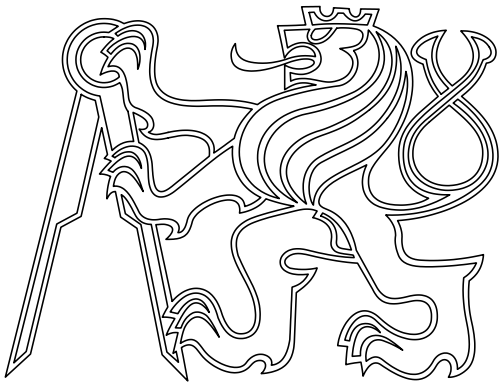
České vysoké učení technické v Praze



Středoškolská technika 2008

Prezentace prací středoškolských studentů a jejich škol na ČVUT v Praze

StreTech 2008

Editoval

Tomáš Mareš, Matej Daniel, Michael Fürbacher, Viktor Kreibich
Radek Šulc, František Tatíček, Karel Vítek
Jan Kudláček, Karel Doubrava, Ctirad Novotný

Praha, 2008

©Tomáš Mareš, Matej Daniel, Michael Fürbacher, Viktor Kreibich, Radek Šulc, František Tatíček, Karel Vítek, Jan Kudláček, Karel Doubrava, Ctirad Novotný
ISBN 978-80-01-04103-1

Obsah

1 Jiří Peroutka: Inteligentní dům 7

2 Martin Vanický: Solární panely v praxi 25

3 Helena Mašková, Vladimíra Mikešová, Iveta Neradová, Lucie Stehlíková: Geoterm 40

4 Lucie Šindelářová: Větrná elektrárna Nová Ves v Horách 52

5 Jakub Horníček, Norman Csiszár: Stirlingův motor 67

6 Jindra Weber, Veronika Bittmannová: Moderní analytické metody v kontrole životního prostředí 72

7 Jaroslava Sukeníková, Michal Šišák: Je kriminalita nezletilých omluvitelná? Upravit zákonem hranici trestní odpovědnosti? (statistické šetření) 102

8 Kristýna Šimoníková, Ivana Vojtová: Moderní metody identifikace biologicky aktivních látek - proteinů 106

9 Alena Vrbová, Ivo Zegzulka: Renderování skleněných předmětů ve 3D scéně 139

10 Martin Švejda: Světelný text na rotační ploše 140

11 Adam Bárta: Zaměření reálného stavu parcel vkatastrálním území Bořislav 146

12 Petr Mundil, Vít Hilenka: Topení biomasou 199

13 Jan Filipovič, Martin Királ: Rekonstrukce Anaburgu v zámecké oboře Horšov, Horšovský Týn 231

14 Umlauf Vojtěch, Trunec Václav, Adam Hák: Atmosférické jevy 232

15 Anna Vogelová, Kateřina Knězáčková: Rozklad a třídění surovin 235

16 Ondřej Tomalík, Jitka Tobišková, Petra Helligerová, Jan Sláma, Marek Draštík: Meteorologie 238

17 Vladimír Burian: Digitální zvukový procesor 240

18 Aneta Hnízdilová: Dřevěné stavby 264

19 Jiří Holeš: Návrh rodinného domu, kde žije tělesně omezený 329

20 Monika Martínková, Ondřej Horký, Michal Trnka, Zdeňka Říhová, Marcela Nováková, Marek Šťastný: Km Beta-Rodinné domy 345

21 Zdeňka Říhová: Novostavba rodinného domu 346

22 Marek Šťastný: Rodinný dům KM Beta 361

23 Michal Trnka: Rodinný dům – Not For Everybody 375

24 Minh Ngoc Mai, Jakub Prokeš: Mini GLOBE Games 402

25 Petr Bezděk: Zelená střecha na panelovém domě 409

26 Vlastimil Hořák, Jan Kalocsányi: Ukázky studentských prací v CAD programech 432

27 Efektivní využití energie 453

28 Jakub Doubrava: Měření elektrické energie odebírané počítačem 459

29 Jan Trejbal: Kostel sv. Mikuláše vLomnici nad Popelkou 472

30 Zuzana Večerníková, Tomáš Kotrbatý: Obchodního domu Ještěd včera, dnes a zítra 473

31 Martin Dolejš, Libor Tomíček, Rosťa Fikáček, Pavel Skořepa: Stirlingův motor 474

32 Ondřej Socha, Jaromír Kult, Filip Kučák: Levitátor 479

33 Lukáš Vacovský: Gotická okna 484

34 Jan Gottlieb, Jaroslav Novák: Automatický zavlažovací systém 489

35 Lukáš Trnka, František Kubát: Telekomunikační systémy – výukový program 491

36 Martin Ko: Identifikace známého zvuku 495

37 Tomáš Podroužek, Zdeněk Bryndač: Vizualizace AutoCAD 2007 502

38 Jan Trauške, Roman Kuběnský: Pohyblivý modul s čidly 504

39 Adam Sedláček, Martin Šedivý: Encyklopedie létání 508

40 Marek Šmolík,Martin Kovářík: Mistři chemie 512

41 Radim Horák: Syntézaa derivace fluorovaných derivátů 3-acyl-4amino-1-arylpyrazolůl 514

42 Jindřich Mašín: Vyhodnocení vlivu skládky na povrchové a podzemní vody 515

43 Petr Tichý, Hana Adámková, Zuzana Kotíková, Kristýna Šimáková: Rak jako bioindikátor čistoty vod 519

44 Jiří Kuřátko: GIRASOL – slunečnice reagující na dopad sluneční energie 522

45 Jiří Hýbek: Projekt umělé inteligence 534

46 Jan Šlajs, David Říha: Inteligentní Elektroinstalace 535

47 Hana Boulová, Josef Gruber, Martin Haken, Jiří Běl, Petr Slavíček: Výuka odborných předmětů strojírenských spodporou CAD/CAM technologií, projektová činnost školy 538

48 Jakub Dubecký: Maculinea alcon – Boršov 540

49 Martin Matuš, Jan Lednický: Studium kmitů matematického kyvadla 541

50 David Bareš, Jakub Kolačný: Casemodding – Red & Black 562

51 PetrPešek, Ondřej Šilhavý: Pneumatické chapadlo manipulátoru 577

52 Jan Pyšvejc, Jakub Malý: Historie Prahy – Pivovarnictví 602

53 PavelŠtěpán, Ondřej Ostárek, Marek Vošahlík: HOMO MUSICUS 612

54 OndřejSýkora, Tomáš Richter: Převod deskové hry LABYRINT do elektronické podoby 619

55 Vojtěch Zavřel: 3D model LEGO – bagr 779

56 LukášKotrba: Grafický editor 806

57 Jan Fikar: Přenos dat pomocí sítě GSM 964

58 Jiří Květoň: Konstrukce prototypu pohybové jednotky robota s krokovými motory 965

59 Pavel Nácal: Čidlo robotnického vidění 967

60 Martin Rendek: Prototyp elektromagnetického děla 970

61 Tomáš Pícha: Model automatické převodovky DSG 972

62 David Fenderl, Miloš Cigánek, Tomáš Blažek: Laserový zaměřovač 984

63 Jakub kříž: Elektromobily 985

64 Milan Daneček: Uhlík a jeho moderní využití 987

65 Josef Chmelař: Využití CAD v technické praxi 1047

66 Michal Benda, Jan Cipro, Jakub Klíma, Miroslav Suk, David Kohout, Jaroslav Černý: Solární vozítka a vozítka poháněná vodíkem 1092

67 Klára Kováčová, Iveta Vildová: Sluneční hodiny – od teorie k praxi 1093

68 Jan Špaček: Lidský chrup, historie a současné trendy 1134

69 Ondřej Perutka: Skladová evidence pro menší firmy 1162

70 Radim Laga: Elektromagnetický urychlovač feromagnetických projektilů 1177

71 Martin Novák, Petr Kunc: Albireo, první český svobodný redakční systém 1179

72 Jakub Petrák, Radek Šembera: Optické pojítko Ronja 1180

73 Jan Hanuš: Digitální teploměr sPIC 1183

74 Jan Kubák: Náhodnéh generátory 1186

75 Oleksandr Kyrychenko: NF výkonový zesilovač svýstupním výkonem 2 x 150 W/8W pro elektroakustické zařízení sobvody LM 4780 1187

76 Marek Slúka: Logická hra SOKOBAN 1192

77 Marie Tomečková: NF výkonový zesilovač 2x200W/4Ω 1195

78 Martin Gába: Školní elektronické hodiny smikrokontrolerem a možností kontroly zvonění 1197

79 Josef Jinek, Radek Štikar: Stirlingův motor – nový zdroj energie 1229

80 David Okurka: Santiniho poutní kostel – virtuální model 1252

81 Ondřej Řezníček: Rangueho vírová trubice 1286

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Inteligentní dům

Jiří Peroutka

Integrovaná Střední Škola Technická, Černolesklá 1997, Benešov,



Obsah

1. Úvod.....

2. Počáteční investice .....

3. Inteligentní dům.....

4. Tepelné čerpadlo.....

- Teplo ze země.....
- Teplo z vody.....

5. Solární panely .....

6. Klasické zdroje.....

7. Velký dokonalý stroj .....

8. Řídící jednotka.....

9. Závěr.....

10. Bibliografie.....

11. Znázornění grafů .....

12. Fotodokumentace .....

Inteligentní dům
Jiří Peroutka

Úvod

Úvodem bych rád použil několik vět, které jsou velice trefné a přečetl jsem si je v jednom nejmenovaném časopise: „Obyvatelé Země spotřebují každou vteřinu 10TW elektrické energie. Tuto energii převádějí na svět zejména fosilní paliva (uhlí, ropa, zemní plyn). Ta však nesvědčí životnímu prostředí a navíc jejich zdroje stále ubývají. O to více je na koni jaderná energie. Existují však i technologie získávání energie z jiných zdrojů, kterým se všeobecně říká alternativní.“

Počáteční investice

Ještě bych Vám chtěl upřesnit přibližnou výši investice do „inteligentního“ domu:

Tepelné čerpadlo	100 000,- Kč
Solární panely	50 000,- Kč
Inteligentní zařízení	80 000,- Kč
Instance + jiný zdroj	70 000,- Kč

- Celá realizace přijde jen na 300 000,- Kč. A když si uvědomíme, že za kotel na tuhé palivo a jeho následnou instalaci bychom zaplatili takřka 80 000,- Kč, náklady nás vyjdou na 220 000,- Kč. Uvážíme-li, že provozní náklady jsou třetinové, realizovat stavbu se opravdu vyplatí, jelikož návratnost se pohybuje okolo pěti let. Pan Brůha takto ročně ušetří až 60 000,- Kč

„Inteligentní“ dům?

„Inteligentní“ dům? Není to jen taková fráze? Jak dům může být inteligentní? – Tyto i mnohé jiné otázky vás mohou napadnout ve spojení „Inteligentní“ dům.

Ale celá věc se má naštěstí trochu jinak. Inteligentní dům je fráze, která se přiděluje objektu, který má příznivý energetický potenciál, nízkou spotřebu energie a zajišťuje celkový tepelný komfort. Co to znamená?

Na střeše jsou solární panely, které mohou ohřívat vodu ve velkém zásobníku, pod zemí jsou kolektory, které vedou k tepelnému čerpadlu (zrovna toto téma je mi nejbližší) a v neposlední řadě kotel, ať už elektrický nebo na tuhé palivo, který vytápí objekt při nejvyšší spotřebě.

Já bych vám rád vysvětlil, jak celé zařízení – pochopitelně jako celek – pracuje. Ale kdybych se měl zabývat každým z mnoha článků, které pohání tento velký a dokonalý stroj, podrobně, tak by tato práce měla zřejmě dvojnásobný rozsah a pro mnohé z vás by byla příliš dlouhá a obsáhlá.

Nyní, když jsem napsal tuto větu, napadla mě jedna otázka. Pro koho tu práci vlastně píši? Jistě je to do soutěže zvané ENERSOL, ale nejen pro soutěž vše sepisuji.

Minulý rok jsem měl možnost absolvovat dvanáct nebo patnáct přednášek na základních školách, kde jsem všem žákům přibližoval, jak funguje tepelné čerpadlo. A tak si na předchozí otázku tak trochu sám odpovím. Tuto práci píši hlavně pro ně. Pro žáky základních škol. Lidé, kteří budou mojí práci mít možnost číst, hodnotit nebo poslouchat, jistě mnoho ví, a tak jim nemusí určité věci přijít zajímavé, ale žákům nebo studentům, kteří o podobných věcech slyšeli maximálně v televizi, může tento projekt přijít zajímavý. Tuto práci tedy píši hlavně těm, kteří jsou v tomto oboru laikové.

Co to všechno znamená? Především to, že tato práce nebude moc náročná a každým tématem se budu zabývat jen v několika málo odstavcích.

Inteligentní dům
Jiří Peroutka

Shrnu tedy věci, které budu popisovat:

1) Tepelné čerpadlo

2) Solární panely

3) Klasický zdroj







Tepelné čerpadlo

Již v devatenáctém století anglický fyzik Lord Krvink objevil princip tepelného čerpadla. Ale jeho využití se začalo používat až ve století dvacátém a hlavně jednadvacátém.

Tepelné čerpadlo je zařízení, které dokáže převádět teplo z nižší teplotní hladiny na hladinu vyšší. Celé zařízení funguje jako obyčejná domácí chladnička. Jde o zařízení, které převádí nižší teplotní potenciál (uvnitř ledničky + 5 - 10°C) na hladinu vyšší (výměník na zadní straně ledničky až + 30°C).

Tepelné čerpadlo pracuje stejně. Převádí venkovní teplo (vzduch v okolí domu, půda v okolí domu nebo spodní voda) na hladinu vyšší.

Existují tedy tři systémy, na základě kterých tepelné čerpadlo pracuje:

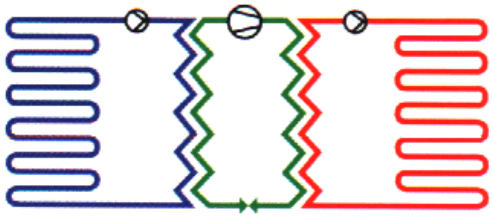
1)Vzduch-voda

2)Voda-voda

3)Země-voda

Jelikož teplo ze vzduchu se zde nedá moc využít, stručně vám popíšu ostatní dvě.

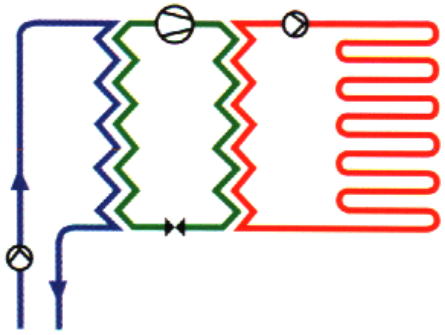
Teplo ze země



Téměř všude je možné využít teplo uložené do povrchových vrstev zemské kůry ze slunečního záření. Vzhledem k velké akumulační schopnosti zemského povrchu je teplota tohoto prostředí prakticky nezávislá na roční době. Přenos tepla ze země zajišťuje nemrznoucí teplonosné medium, které cirkuluje v plastovém potrubí zemního kolektoru. Tento kolektor se zpravidla umísťuje buďto horizontálně zhruba 1,5 až 2 m pod povrch země do ručně

nebo mechanizací provedených výkopů, nebo formou zemních smyček do zemních vrtů hlubokých zpravidla 50 až 120 m. Horizontální zemní kolektory jsou při stejném tepelném výkonu asi o polovinu levnější než zemní vrty, avšak jejich nevýhoda je ta, že jsou poměrně náročné na rozlohu potřebného pozemku.

Teplo z podzemní vody



Tepelná čerpadla pro tyto aplikace jsou stejné konstrukce jako tepelná čerpadla v předchozím případě. Rozdíl je pouze v tom, že přes primární okruh výparníku neprotéká teplonosné médium zemního kolektoru, ale přímo voda buďto z vhodné studny, nebo vodoteče. Podmínkou použití tohoto systému jsou vhodné hydrogeologické podmínky v dané lokalitě a dostatečná vydatnost pramenů. Voda se zpravidla čerpá z jednoho vrtu a po ochlazení v tepelném čerpadle se vpouští do vsakovací jímky popř. vrtu vzdáleného 15 až 20 m. Systémy čerpající teplo přímo z podzemní vody dosahují sice lepších topných faktorů než systémy čerpající teplo ze země, ale vyžadují některé další technické prvky a jejich údržbu (vodní filtry, hlídače zaplavení primárního čerpadla apod.).

Solární panely

Elektrická energie získaná přímou přeměnou slunečního záření je známa již z 19. století. Velký rozvoj aplikace fotovoltaického jevu byl a je závislý na technické úrovni a znalostech především z oblasti fyziky polovodičů. Vlastní fotovoltaické systémy pak představují spojení fotovoltaických součástek do řetězce, na jehož konci jsou elektrické spotřebiče, vykonávaná práce atd. Tímto způsobem využití energie se však zabývat nehodlám. Já se vám pokusím přiblížit přeměnu sluneční energie v teplo.

I když vám to tak nemusí připadat, i v České republice se dá využít sluneční energie

Princip je jednoduchý. Sluneční teplo, které dopadá na plochu solárního panelu, se v absorbéru, který je nejčastěji kovový, přemění na teplo. Toto teplo z absorbérů odebírá teplonosné médium, nejčastěji nemrznoucí směs, které je odvádí do výměníku, kde se teplo uchovává nebo předává k dalšímu použití (vytápění objektu či bazénu, ohřev TUV). Tento jednoduchý princip však není snadné technicky realizovat – absorbér musí pohlcovat dostatečné množství tepla a zároveň ho co nejméně vyzařovat, celý systém musí být velmi dobře izolován, aby nedocházelo ke ztrátám, solární panely musí být zároveň odolné a mít dostatečně dlouhou životnost. V dnešní době však všechny tyto technologické obtíže dokázalo zvládnout již poměrně velké množství výrobců a tak reálnému nasazení solárních panelů jako zdroje energie nic nebrání.



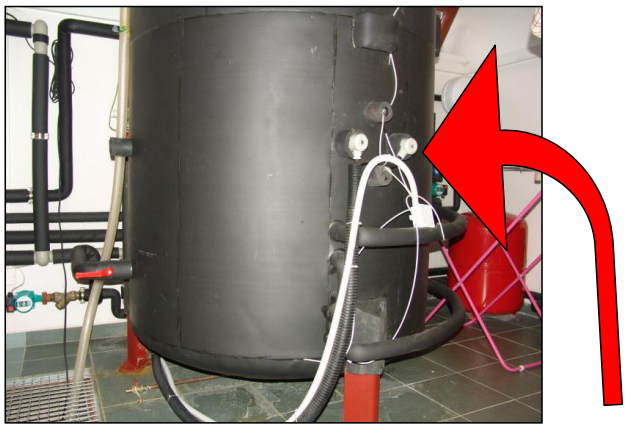
Inteligentní dům
Jiří Peroutka

Klasické zdroje

Nesmíme zapomínat na poslední článek, který je také velmi důležitý. Zajišťuje, aby celé zařízení pracovalo i v případě nejvyšší spotřeby: kotel na tuhé palivo, plyn nebo elektřina. Jelikož pan Brůha používá elektrické články, budu se zabývat jimi.

V našem případě jsou elektrické články nainstalovány na zásobník, kam dodávají teplo, když je potřeba. To jsou většinou chvíle, kdy se využívá plná hladina zásobníku a alternativní zdroje energie zkrátka nestíhají – někdo se koupe, pere pračka, k tomu se celý objekt vytápí a mi zrovna potřebujeme teplou vodu na nádobí. Sepnou se elektrické články a dodají potřebnou energii zásobníku.

Jak jsem již říkal, dá se využít i jiných zdrojů, jelikož řídící jednotce nezáleží, odkud se potřebná energie bere.



Dva elektrické články

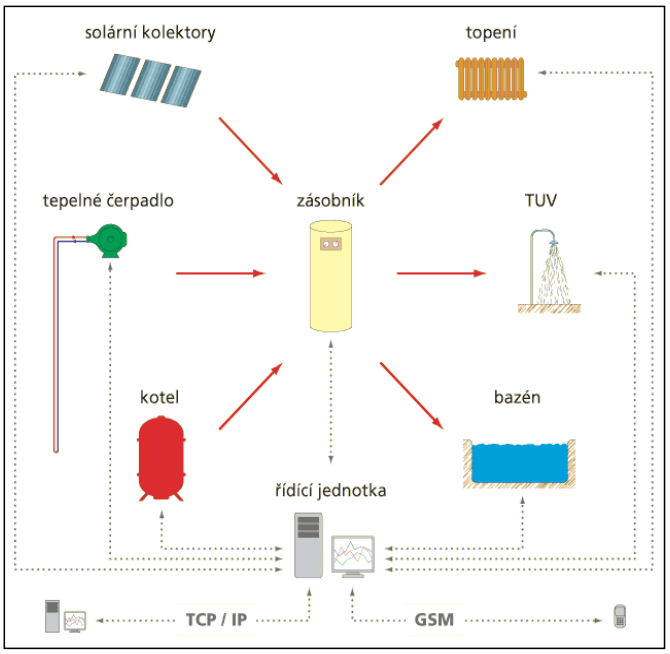
Velký dokonalý stroj

Nyní, když jsem vysvětlil, jak pracují jednotlivé komponenty, pokusím se vám přiblížit, jak přístroj pracuje jako celek. A abych to mohl uskutečnit, musel jsem se vypravit za panem Brůhou.

Tak trochu jsem jel do Nebřichu s pocitem, že se tam dozvím strohé informace, ale pan Brůha mě svými informacemi doslova zavalil a já se pokusím vše, co jsem se dozvěděl, jednoduše vysvětit.

Ve sklepě domu je veliká nádrž, která funguje jako bojler, je v ní topné medium, ke kterému jsou přivedeny tři komponenty. Prvním, a asi nejdůležitějším, je tepelné čerpadlo, které dodává tu největší část energie pro ohřev. Druhým je solární článek, který podobně jako tepelné čerpadlo dodává energii pro ohřev vody v nádrži. A třetím, také velice důležitým prvkem, jsou dva elektrické články. Celek funguje jen díky jedinému zařízení:

Řídící jednotka.



Inteligentní dům
Jiří Peroutka

Řídící jednotka

Asi nejdůležitějším článkem je řídící jednotka, která je pomocí systému čidel a regulačních prvků napojena na všechny ovládané spotřebiče a zdroje tepla. Řídící jednotka neustále vyhodnocuje a zaznamenává signály z čidel a na jejich základě reguluje podle nastaveného programu zdroje a spotřebiče.

Řídící systém rovněž řídí spotřebu tepla. To obnáší hlavně regulaci topení v jednotlivých místnostech objektu, případně regulaci ohřevu bazénu. Cílovou teplotu je možné zvlášť nastavit pro jednotlivé místnosti v libovolných časových režimech tak, aby nikde v objektu nebylo přetopeno nebo naopak příliš chladno. Ze zkušeností s provozem ID vyplývá, že pouhé nasazení inteligentního řízení dokáže ušetřit až 30% spotřeby tepla.



Řídící jednotka celého systému je zcela samostatná a díky tomu bezpečná a spolehlivá – nebezpečí kolapsu v důsledku selhání operačního systému či napadení virem je minimalizováno tím, že se jedná o jednoúčelové zařízení s (relativně) jednoduchým softwarem. Tato řídící jednotka je ovládána přes běžné PC s operačním systémem Windows, na němž běží software ConWin. Ten se stará o grafické znázornění aktuální situace v objektu, dokáže vykreslit grafy všech sledovaných veličin v libovolném časovém období a rovněž umožňuje uživatelsky přívětivým způsobem naplánovat topný režim v objektu. V případě výpadku PC běží samotná řídící jednotka dále podle nastavených programů.

Díky tomu, že ovládání řídící jednotky je realizováno pomocí architektury klient-server přes běžné síťové protokoly (TCP/IP), je možné systém řídit odkudkoliv, kde je k dispozici připojení k internetu a to buď s plným komfortem pomocí ovládacího software ConWin. Systém je rovněž možné ovládat pomocí mobilního telefonu.

Inteligentní dům
Jiří Peroutka

Závěr

Měl jsem možnost o alternativních zdrojích **přednášet na základních školách a zjistil jsem, že většina žáků má o toto téma zájem, ale pro nedostatečný počet hodin a pro nezačlení tohoto tématu v osnovách, se o problematice tohoto tématu nevěnují.**

Pro svoji potřebu, a také pro názornost, **jsem udělal malý průzkum: Jaká je informovanost o obnovitelných a alternativních zdrojích energie? Okolo 90% lidí mi odpovědělo, že vědí, co to znamená a vyjmenovali mi jeden a více zdrojů. O inteligentních domech mi něco dokázalo říct pouhopouhých 9% respondentů. Stejně jako minulý rok jsem se setkal se značnou neochotou a nevolí na mé dotazy odpovídat.**

Doufám, že jste se z této práce poučili tak jako **já a že se každý z nás určitým způsobem zamyslí nad velkým problémem, který tíží a bude tížit nás všechny – znečišťování planety a s tím i související globální oteplování.**

Na závěr bych velice rád poděkoval všem, kteří mi nějakým způsobem pomohli. Jmenovitě; Panové Karel Šebek, Petr Brůha a paní učitelky Jana Křížová a Jitka Kartusová.

Bibliografické údaje

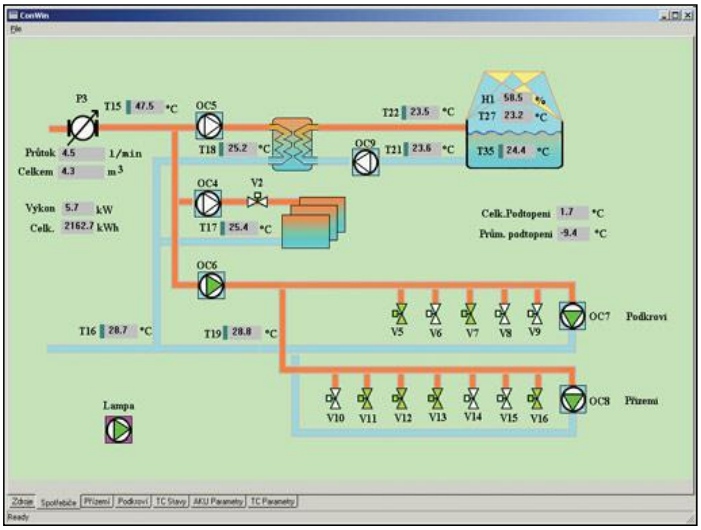
1) www.solary.cz - stránka o inteligentních domech.

2) Osobní rozhovor, poznatky pana Brůhy.

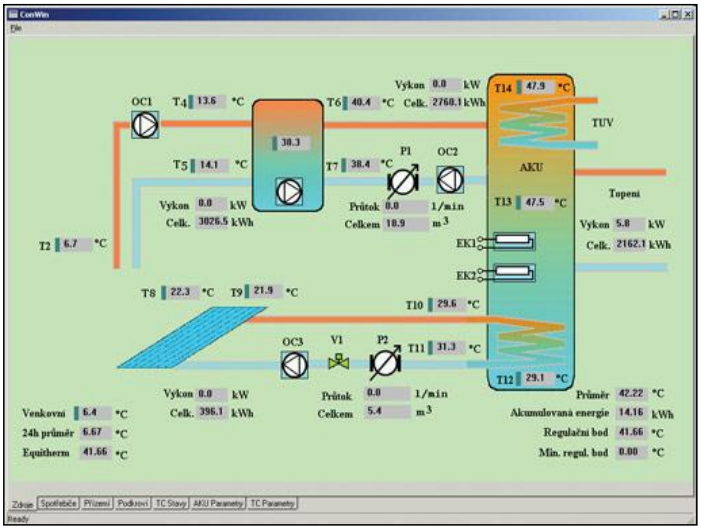
3) Obnovitelné zdroje energie kap 4, 7 ing. Jiří Beránský
Str. 111/ kap. 7.1/ odst. 1
Str. 122/ kap. 4. / odst. 1

Inteligentní dům
Jiří Peroutka

Znázornění Grafů

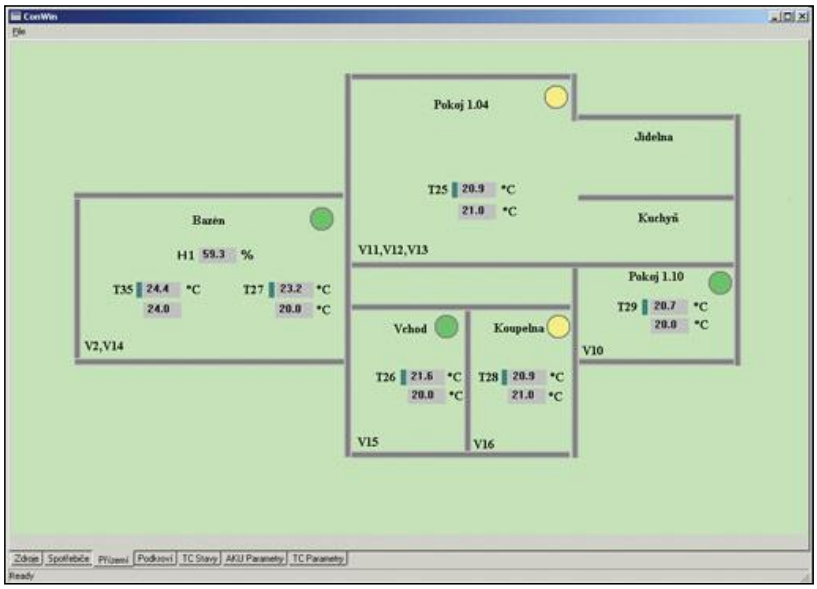


Přehled spotřebičů

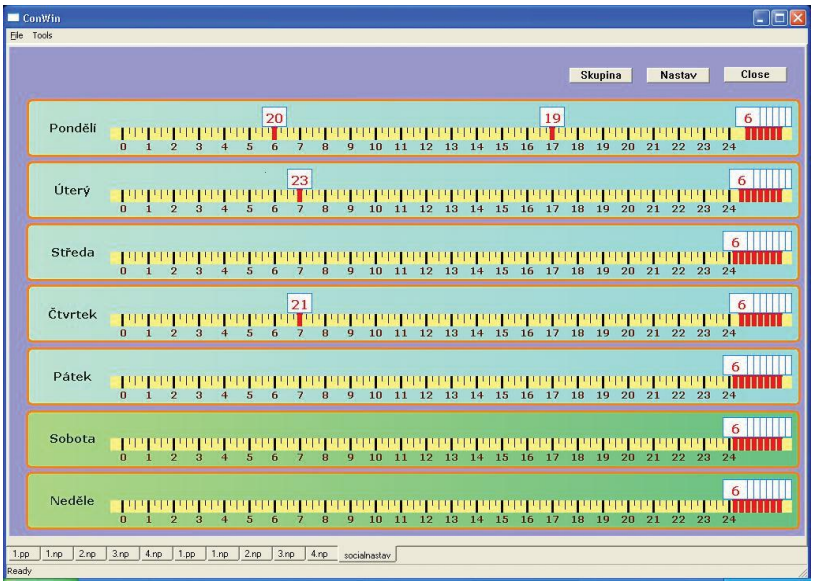


Přehled nastavení zdrojů tepla

Inteligentní dům
Jiří Peroutka



Přehled nastavení teplot



Kalendář nastavení teplot v nejbližším týdnu

Inteligentní dům

Jiří Peroutka



Teplota ve vrtu tepelného čerpadla



Výkon tepelného čerpadla



Teplota v pokoji

Inteligentní dům
Jiří Peroutka

Fotodokumentace



Pohled na celý objekt



Solární panely

Inteligentní dům
Jiří Peroutka

Enersol 08



Tepelné čerpadlo





Zásobník

Inteligentní dům
Jiří Peroutka



Regulační ventil



Pan Brůha



Pan Šebek, já a pan Brůha

Inteligentní dům
Jiří Peroutka

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Martin Vanický

Integrovaná Střední Škola Technická, Černoleská 1997, Benešov,

SOLÁRNÍ PANELY V PRAXI

Jsem studentem Integrované střední školy technické v Benešově u Prahy. Chodím do druhého ročníku maturitního oboru Mechanik Elektronik. Toto téma jsem si vybral také vzhledem k mému hobby kterému se dlouhodobě věnuji.

Stručný obsah

- Základní podstata funkčnosti solárních panelů a jejich složení.
- 2 základní použití v domácnostech
- Využití v praxi, aneb jak se dá využít panel na kopci?
- Klady a zápory
- Poděkování, zdroje informací

Přejděme k tématu.

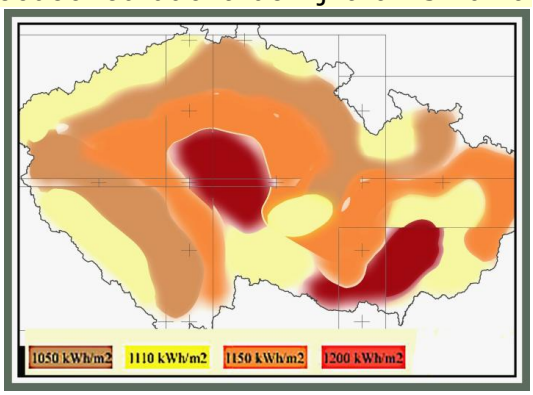
Základní podstata funkčnosti solárních panelů.



Solární kolektory pro svou nejkvalitnější činnost potřebují slunce, ale v našich zeměpisných podmínkám to není úplně ideální, ale přeci zde se celková doba slunečního svitu pohybuje od **1400-1800 h/ročně**, což není hodně ale ani málo. Ze Slunce se každým okamžikem uvolňuje velké množství energie, které je formou elektromagnetického záření předáváno, do kosmu. Záření po **150 milionech km** a **8,3 minutách** dorazí k planetě Zemi. Při průchodu zemskou atmosférou je část záření pohlcena, rozptýlena, odražena nebo vyzářena.
Pohlcené záření Zemí přijímané v bezmračném počasí se nazývá **přímé,** naopak rozptylem přímého záření v mracích a na částečkách v atmosféře vzniká záření **difúzní.**

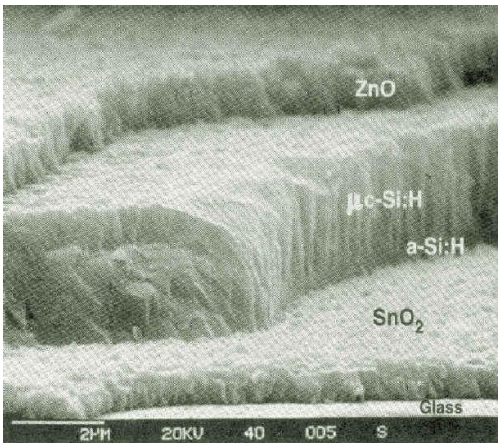
V našich klimatických podmínkách dopadá **ročně na 1 m^2 vodorovné plochy 800-1250 kW-h** zářivé energie. Z toho přibližně 75 % připadá na období od dubna do října a 25 % na zbývající část roku.

Obrázek znázorňující, kde na našem území dopadá nejvíce energie (převedené na kWh/m2).



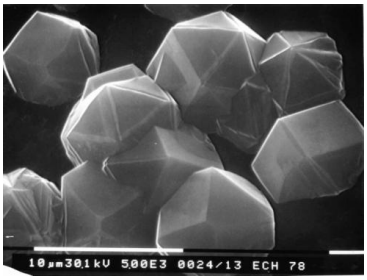
Je zde názorně vidět, že čím jste níže položenému místu tím lepší účinnost beheme roku. (lepší dopad – povětrnostní vlivy)

Fotovoltaický (sluneční, solární) článek je v podstatě **polovodičová dioda**. Jeho základem je tenká křemíková destička s vodivostí typu **P**. Na ní se **při výrobě** vytvoří tenká vrstva polovodiče typu **N**, obě vrstvy jsou odděleny tzv. přechodem **P-N**. Osvětlením článku vznikne v polovodiči vnitřní fotoelektrický jev a v polovodiči se z krystalové mřížky začnou uvolňovat záporné elektrony. Na přechodu P-N se vytvoří elektrické napětí, které dosahuje u křemíkových článků velikosti zhruba 0,5 V. Energie dopadajícího světla se v článku mění na elektrickou energii. Připojíme-li k článku pomocí vodičů spotřebič (například miniaturní elektromotorek), začnou se kladné a záporné náboje vyrovnávat a obvodem začne procházet elektrický proud. Je-li třeba větší napětí nebo proud, zapojují se jednotlivé články sériově či paralelně a sestavují se z nich **fotovoltaické panely**.



Z řady důvodů je pro výrobu **fotovoltaických** (neboli solárních či slunečních) článků velmi vhodným materiálem **křemík (Si)**. Jedná se o druhý nejrozšířenější prvek na Zemi, jeho zpracování pro potřeby mikroelektroniky dosáhlo velmi vysoké úrovně. Různými metodami je dnes možno získat křemíkové krystaly nepředstavitelné čistoty až 99,9 %.



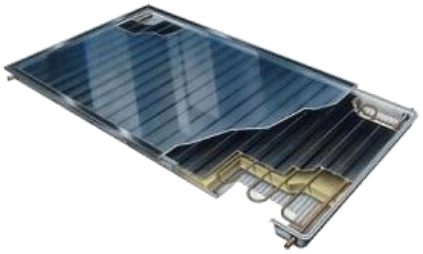


2 základní použití v domácnostech.



Přeměna slunečního záření na elektrickou energii. **Ohřev vody**







Ohřev vody pomocí solárních panelů je v dnešní době více a více rozšířeným trendem. Jde o ohřev bazénu či užitkové vody nebo vody, která se používá v bytech jako pitná. Sluneční energie je zadarmo, záleží však na straně příjmu celého systému panelů, který si zvolíme sami tak, aby se nám stala například denně teplá voda každodenním rytmem.



Z mého osobního pohledu je tento trend používaný více a více, protože pohodlnost dnešních lidí je na takové úrovni, že chtějí po celou dobu léta teplou vodu, která jim bude k dispozici pokaždé, když k bazénu přijdou.
Mnoho lidí tvrdí, že ohřev pomocí solárních panelů je využitelný jenom v létě při velkých teplotách, ovšem jde využívat ohřev i na jaře či na podzim sice účinnost systému padá, ale spousta lidí má bazény schované pod střechou, která jim nabízí teplou vodu s pomocí tohoto systému i v tuto dobu. Bohužel jsem nenašel nikoho ochotného, kdo se této problematice věnuje a poskytne mi informace, které jsou s tímto problémem spojeny. Doufám, že další rozdělení vás zaujme a poskytne něco nového.





Možná někomu obrázek napoví ale zřejmě asi ne.

Tato kapitola převod slunečního záření na elektrickou energii mě zaujala zejména tím, že v mém hobby je si myslím velikým pomocníkem, který se hodí kdekoliv. Moje záliba je radioamatérská činnost, které se věnuji už několik let a zahrnuje to poznání jak krásných západů slunce z kopců, tak i navázání krásných spojení po celém světě. Zabývám se zejména závoděním a jako málokdo v mém věku používám telegrafii na úrovni přes 100 znaků za minutu. Ovšem přejděme k mému projektu.

Můj projekt se zabývá použitím solárního systému při závodech a jeho účinnosti.



Mnohokrát vyrážíme na kopec za účelem nějakého radioamatérského závodu, navazovat spojení či při výskytu inverze nebo různých zvláštních jevů spojené s tímto koníčkem. Zrovna nedávno jsme vyjeli na kopec nedaleko Benešova. Moje místo odkud vysílám je ve středu Benešova, tudíž si nemůžu dovolit jak vysílání větším výkonem, tak i, řekněme si na rovinu, z 300 metrů nad mořem toho moc na velmi krátkých vlnách nejde udělat. Každý ví, že rozhlasová stanice, která je daleká 100km, jde lépe nejvýše položenému přijímači.
Jednou v létě při hodně vysoké teplotě sedím doma a odpočívám říkám si : co kdybych pustil chat, kde se schází radioamatéři z celé Evropy. Pouštím internet a koukám, že se to nějak začíná skládat na sebe v Německu. Skládat znamená - vytváření tzv. ES vrstvy, která se skládá smícháním dvou teplotně rozdílných proudů (teplý a studený), který zapříčiní to, že se vytvoří tzv. DUKT, který je vodivý a pošle VF signál 1000km daleko. Toto vše je stručně řečeno.
Říkám si, bylo by to fajn, kdybych byl někde na kopci, možná že bych něco slyšel. Ovšem po 30minutách začínaly spojení do Španělska dělat stanice z Prahy. V tu dobu jsem s tátou, který unavený přijel z práce, jel na kopec. Cestou volám kamarádovi, který přijel taky. A je to tady. Krásná spojení do Španělska,Francie a Portugalska. Postupně jsme se vystřídali všichni u mikrofonu. Asi po 2hodinách nám přestala dodávat proud baterie, poznal jsem to na rádiu, které má voltmetr. Ovšem z 9 voltů bylo minimum i na transciever, a tak se vypnul. Co teď, tak krásně to chodilo, domů se nám nechtělo. Byl krásný letní den, avšak Jirka, který byl na kopci také dělal spojení, ale jako já a táta slyšel ticho a šumění lehkého vánku. Ovšem Jirka je připraven moderně, ihned vytáhl systém panelů s baterií, která se ihned začala dobíjet při namíření systému kolmo ke slunečnímu záření. Připojil jsem červený a černý vodič k zařízení a hle jedeme dále. Ten den jsem udělal několik desítek pěkných spojení, která mám dneska potvrzená tzv.QSL lístkem.
Jednou o víkendu jsem jel s Jirkou na kopec, jen tak ze srandy, byl takový ten nudný víkend, a tak jsem jel sním. Chtěl otestovat nový systém panelů který pojmenoval „ Butty systém", dnes nevím proč se tak jmenoval. Přijeli jsme na louku poblíž Struhařova a zkoušelo se a dopadlo to nějak takhle.

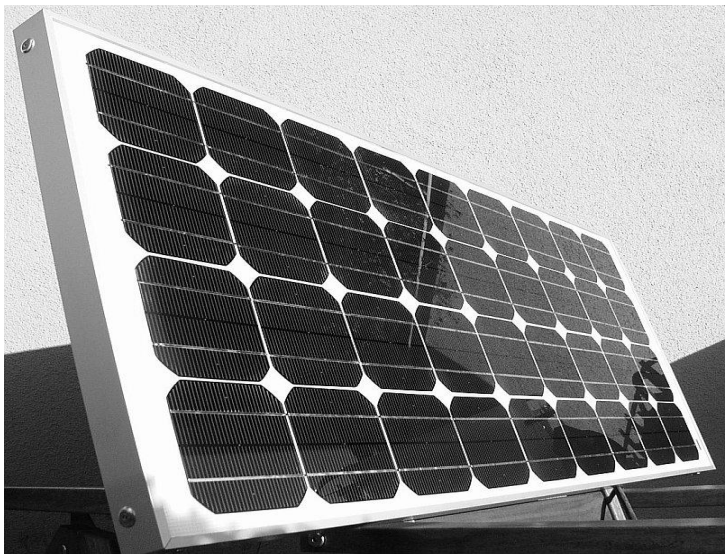
band	70 cm	2 m	20 m
RX	0,59 A	0,56 A	0,58 A
RX + fan	0,8 A	0,8 A	0,8 A
TX 0W out	1,9 A	2,0 A	1,7 A
TX 2W	2,8 A		
TX 5W	3,5 A	3,1 A	3,8 A
TX 10W	4,6 A	3,9 A	5,1 A
TX 15W	5,7 A	4,7 A	5,6 A
TX 20W	7,1 A	5,6 A	6,3 A
TX 50W		8,7 A	10,1 A
TX 100W			14,2 A

Zde je tabulka proudů který na daném pásmu byl odebrán v závislosti na výkonu.

Pro zajímavost – originální tři metry dlouhý napájecí kabel, dodávaný výrobcem společně s rádiem zatížení sedmi ampéry má úbytek 0,4V což při výkonu 20W na 70cm (430MHz) je ztráta jenom na kabelu 3W.



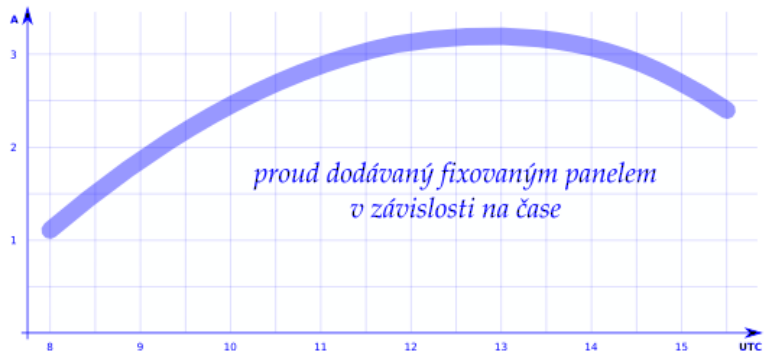
Během červencového odpoledne, generoval 55W panel ještě v sedm hodin odpoledne 2,4 A do umělé zátěže což je poměrně dobré.



Panel je součástí ostrovního systému, který obsahuje regulátor a padesáti ampérhodinový olověný akumulátor.

Vytvořil jsem graf, který znázorňuje přidělaný panel fixním směrem v závislosti na čase.

Je zde vidět maximum slunečního záření okolo 12-13hodiny UTC = SEČ 14 – 15h.



Několik mých *kladů a záporů*

Uveďme si několik kladných faktů, které jsou v solárním použití vhodné za zveřejnění.

Solární panely pracují poměrně automaticky o vše se stará elektronika.

Solární panely jsou ihned připraveny vytvářet elektrickou energii

Jejich údržba je celkem minimální.

I v našich zeměpisných podmínkách se dá celoročně vyrobit Na 1m2 1000kWh.

……..

Jejich původní finanční náklad je ve stovkách tisíc korun.

Nedají se využívat celoročně na plný výkon oproti jižnějším zemím.

Každá rodina si nemůže tento systém koupit z finančních důvodu.

Budoucnost solárních panelů ?



Třeba jednou budeme jezdit nakupovat nebo k moři pomocí fotovoltaických panelů

Z mého vlastního pohledu si myslím, že vše je jenom otázkou času, kdy se solární panely začnou používat v netradičních způsobech. Globální oteplovaní a tání ledovců má za následky veliké problémy, s kterými je každý seznámen ovšem za několik stovek let se může stát cokoliv.

Teď se trošku zasněme, že pojedeme pomocí solárního systému k moři do Chorvatska, nebo v závislosti na soláru bude elektricky funkční celý barák či panelák.

Uvidíme jak to vše příroda zařídí a doufejme, že problémy které lidstvo zasáhnou nás konkrétně trápit nemusí.



KV,VKV,UHF Transciever Yaesu FT 897D

Závěr:

Shodli jsme se na tom, že v praxi využívat tento systém je velmi účinná věc, kterou by měli lidé používat nejenom na výrobu plnohodnotného elektrického proudu. Tato kapitola, kterou jsem se vám snažil shrnout na pár stránek, si myslím, že stojí za úvahu**. Položme si otázku : není vhodné tento systém používat na chalupě či na chatě jako přenosný, než mít nainstalován solární systém celoročně na střeše?** V našich podmínkách není podle mého názoru využívána celoročně výroba elektrického proudu pomocí systémů solárních panelů. Vždyť panel o napětí 0,5v a proudu okolo 2A stojí cca 250kč a myslím si, že není veliký problém tento systém sériově zařadit za sebou. Vše okolo tohoto je otázka budoucnosti a domnívám se, že s pravidelným zdražováním veškerých služeb jednou na tuto, pro většinu neznámou, věc přijde čas.



Chtěl bych poděkovat Jirkovi, bez něj bych na toto téma nikdy nepřišel a nikdy se asi ani nepřihlásil.

Veškeré obrázky jsou upravovány v Zoneru a focené vlastním fotoaparátem.

ZDROJ :
Internet, kamarád Jirka, můj radioamatérský koníček.

Martin Vanický

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

GEOTERM

Helena Mašková, Vladimíra Mikešová, Iveta Neradová, Lucie Stehlíková

Střední zdravotnická škola
Máchova 400, Benešov

Téma, o kterém se nemluví

Na úvod bychom chtěly říci, jak jsme se do této soutěže dostaly. Když jsme se dozvěděly, že se tato soutěž koná, neváhaly jsme a přihlásily se. Jelikož jsme byly s tímto tématem obeznámeny již na základní škole v Neveklově, řekly jsme si, že s našimi znalostmi v oblasti geotermální energie bychom mohly uspět. Doufáme, že se Vám naše práce bude líbit.

Na práci se podíleli:

Helena Mašková



Iveta Neradová



Lucie Stehlíková



Vladimíra Mikešová



Obnovitelný zdroj energie je poněkud nepřesné označení některých vybraných, na Zemi přístupných forem energie, získané primárně především z **jaderných přeměn** v nitru **Slunce**. Těmito reakcemi se přeměňuje sluneční **vodík** (který obnovován není) na **helium** za uvolnění velkého množství **energie**. Definice podle českého zákona o životním prostředí je „Obnovitelné přírodní zdroje mají schopnost se při postupném spotřebovávání částečně nebo úplně obnovovat, a to samy nebo za přispění člověka.“ Ve skutečnosti může být obnovitelnost řady zdrojů považovaných za obnovitelné sporná.

Druhy obnovitelných zdojů energie

Ze **Slunce** je energie předávána na Zemi ve formě **záření.** Energetický příkon ze Slunce je ve vzdálenosti, v níž se nachází Země, přibližně 1300 W/m^2.

- Pokud se tato energie přeměňuje nějakým technickým zařízením (**sluneční kolektor**, **fotovoltaický článek**) přímo, mluvíme obvykle o **sluneční energii**.
- Pokud je tato energie předtím vázána v živých organismech (většinou ve formě sloučenin **uhlíku** - například ve **dřevě**), mluvíme o energii **biomasy**.
- Pokud je tato energie vázána do potenciální energie vody (viz **koloběh vody**), mluvíme o **vodní energii**.
- Pokud se tato energie přemění na kinetickou energii vzdušných mas, mluvíme o **větrné energii**.
- Větrná energie může uvést do pohybu vodu na hladinách oceánů. Tuto energii nazýváme **energií vln**
- Kinetická energie soustavy Země - Měsíc (přeměněná na **energii přílivu**).
- Mezi obnovitelné zdroje se obvykle zařazuje navíc i energie z jaderných reakcí v nitru Země (**geotermální energie**). V některých případech však není vyloučeno vyčerpání „ložisek“ geotermální energie.
- Podle řady názorů patří mezi obnovitelné zdroje energie rovněž **jaderná energie**.
- Za obnovitelné zdroje naopak **není** považována přeměněná biomasa, v níž se sluneční energie akumulovala před delší dobou (**uhlí**, **ropa**, **zemní plyn**). Tyto zdroje jsou souhrnně nazývány **neobnovitelné**.







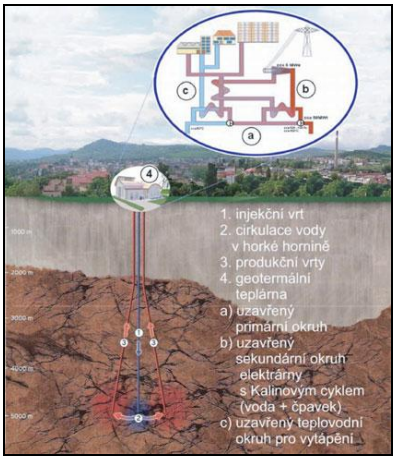
Geotermální energie

Geotermální energie *je projevem tepelné energie zemského jádra, která vzniká rozpadem radioaktivních látek a působením slapových sil. Jejími projevy jsou erupce sopek a gejzírů, horké prameny či parní výrony. Využívá se ve formě tepelné energie (pro vytápění), či pro výrobu elektrické energie v geotermálních elektrárnách. Obvykle se řadí mezi obnovitelné zdroje energie, nemusí to však platit vždy, některé zdroje geotermální energie jsou vyčerpatelné v horizontu desítek let.*

CHARAKTERISTIKA

Geotermální energie je vlastně nejstarší energií na naší planetě Zemi, protože je to energie, kterou získala Země při svém vzniku z mateřské mlhoviny, následnými srážkami kosmických těles a v poslední době je energie částečně generovaná radioaktivním rozpadem některých prvků v zemském tělese.





VYUŽITÍ GEOTERMÁLNÍ ENERGIE

- *Vystavění geotermu vyjde cca. Na 250 000 Kč a z toho stát vracel 60 %.*

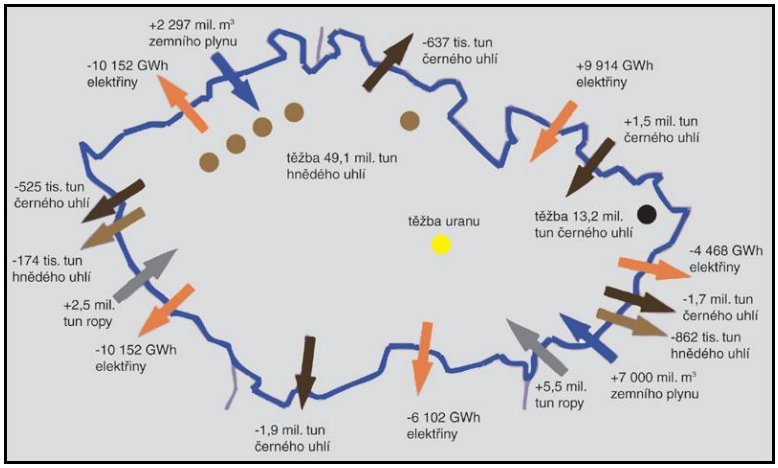
-Tuto energii lze v příznivých podmínkách využívat k vytápění nebo výrobě elektřiny v geotermálních elektrárnách. Takové využití je ale většinou technologicky náročné, protože horká voda z vrtů je obvykle silně mineralizována a zanáší technologická zařízení, což má za následek nutnou častou výměnu potrubí a čištění systému. Navíc je dostatečný tepelný spád obvykle zároveň spojen s geologickou nestabilitou oblasti, v níž se nachází, což klade vysoké nároky na kvalitní stavbu schopnou odolávat zemětřesením.

- *V rozsáhlejším měřítku se tato energie využívá např. na Islandu, kde se využívá pro vyhřívání obytných domů, skleníků, veřejných budov, bazénů, pro vyhřívání chodníků, aby se v zimě nemusely příliš upravovat a dokonce i pro pěstování banánů či jiného jižního ovoce. Další země, které geotermální energii ve větším využívají jsou USA, Velká Británie, Francie, Švýcarsko, Německo a Nový Zéland. Nové studie předpokládají velký rozmach geotermální energie v Austrálii, kde se plánuje stavba soustavy elektráren v tektonicky aktivní oblasti. Pokud se projekt osvědčí, počítá se s podobnými stavbami i na dalších místech Země.*
- *První geotermální elektrárna byla otevřena v Larderello, Itálie už v roce 1904. Celosvětově se produkce pohybovala okolo 8 000 megawattů a na Spojené státy připadalo 2 700 megawattů.*
- *Dnes se využívají tři druhy elektráren - na suchou páru, na mokrou páru a horkovodní (binární). Systém suché páry používá přímo páru získanou ze země na pohon turbíny. Systém mokré páry nechá nejprve horkou vodu přeměnit v páru a ta pak slouží k pohonu turbíny. Horkovodní (binární) systém použije vodu s nízkou teplotou, která předá ve výměníku teplo organické kapalině (např. propan, isobutan a freon) s nižším bodem varu, a teprve její pára pak pohání turbínu.*

VYUŽITÍ V ČESKU

V Česku využívá geotermální energii např. město Ústí nad Labem, kde slouží k vytápění plaveckých bazénů a od května 2006 také k vytápění zoologické zahrady. V Litoměřicích se od listopadu 2006 hloubí zkušební vrt pro geotermální elektrárnu, který by měl skončit v hloubce 2500 m. Pokud budou výsledky měření příznivé, začnou se hloubit další dva vrty - tentokrát již produkční. Tyto vrty mají dosáhnout hloubky až 5000 m. Elektrárna bude založena na metodě HDR, která ještě nebyla ve střední ani východní Evropě použita. Tato metoda spočívá v tom, že se do jednoho vrtu vhání voda, a ze

druhém se čerpá, přičemž se voda v hloubce ohřívá. Jedná se o uzavřený oběh média - vody. V zimě se bude energie využívat především pro vytápění, v létě naopak pro vytváření elektrické energie. Náklady na vybudování vrtů elektrárny mají být kolem 1,11 miliardy Kč, na jejich krytí se bude podílet i EU.



Využití geotermální energie v Neveklově

Základní škola a Obchodní akademie využívají už od svého vzniku jeden z obnovitelných zdrojů energie a tím je geotermální energie. Pod ZŠ a OA je ***28 vrtů****, které jsou* ***150 metrů*** *hluboké, mezi nimiž musí být určitá vzdálenost, aby si nebraly teplo. V celém areálu ZŠ a OA jsou 4 geotermální čerpadla. Zvlášť je pro ZŠ, pro OA, pro tělocvičnu a pro internát.*

Výhodami pro ZŠ je, že topení je ekologické, má výhodnou sazbu s nízkým tarifem a vysoký stupeň kvality.

- ***Nevýhoda*** *je, když vypadne elektrický proud, když se rozbije nějaký kompresor, protože jsou velice drahé opravy, velká spotřeba elektrického proudu.*



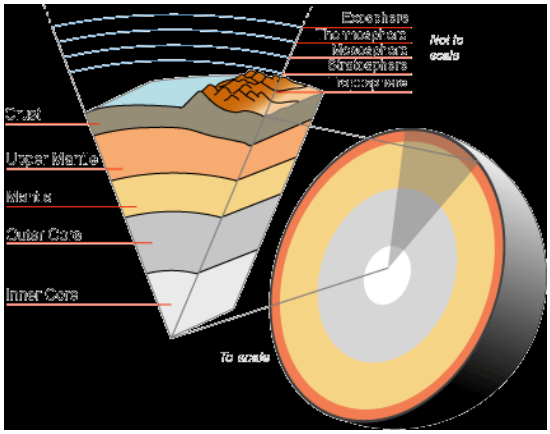
- *Celý tento geoterm funguje tak, že si kompresory vezmou teplo ze země, jde to přes kompresory a ty dokážou vyhřát vodu až na* ***50 °C****.*

- *Geotermální topení vytopí při 0 °C celou ZŠ. Když je chladněji, škola zapíná i plynové vytápění, což je také ekologické, ale dražší.*









Slovo geotermální má původ ve dvou řeckých slovech: ***geo (země)*** a ***therme (teplo),*** a znamená teplota země, a podle toho se tepelná energie Země ještě nazývá i geotermální energie. Teplota ve vnitrozemí je výsledek formování planety z prachu a plynů před více než čtyřmi miliardami let. Radioaktivní rozkládání prvků v horninách kontinuálně regeneruje teplo, a podle toho je geotermální energie obnovitelný zdroj energie. Základní medium, který přenáší teplo z vnitrozemí na povrch je voda nebo pára, a ten komponent se obnovuje tak, že dešťová voda proniká hluboko skrz rozpukliny, tam se zahřívá a cirkuluje zpět k povrchu, kde se objevuje ve formě gejzíru a vroucích pramenů.

Vnější tvrdá kůra Země je hluboká ***5*** až ***50 kilometrů*** a skládá se z hornin. Hmoty z vnitřní vrstvy neustále vycházejí na povrch skrz sopečné jícně a pukliny na dně oceánu. Pod kůrou se nachází plášť a šíří se až do hloubky kolem ***2900 kilometrů***, skládá se ze sloučenin bohatých železem a magnéziem. Zpod všeho toho se nacházejí dvě vrstvy jádra - tekoucí vrstva a tvrdá vrstva v samém jádru planety. Poloměr Země je kolem ***6378 kilometrů***, a nikdo vlastně neví, co se přesně nachází ve vnitrozemí, toto všechno jsou vlastně vědecké předpoklady o vzhledu vnitrozemí planety. Tyto předpoklady se zakládají na pokusech v podmínkách vysokého tlaku a velkých teplot.

Teplota vnější vrstvy Země (kůry) se zvyšuje kolem 17 °C až 30 °C na kilometr hloubky (50 – 87 °F na míly hloubky). Pod kůrou se nachází plášť, který se skládá z částečně rozpuštěných hornin, teplota pláště je mezi 650 a 1250 °C (1200 – 2280 °F). V samém jádru Země by mohla teplota být, podle některých ocenění, mezi 4000 a 7000 °C (7200 – 12600 °F). S ohledem na to, že teplo vždy přechází z teplejších částí na chladnější, teplo z vnitrozemí se přenáší k povrchu a přenos tepla je zásadní hybná síla tektonických desek. Na místech, kde se spojují tektonické desky může dojít k pronikání magmy vrchními vrstvami, magma se pak chladí a tvoří novou vrstvu zemské kůry. Když magma přijde k povrchu, může vytvořit sopky, ale většina zůstává pod povrchem a tvoří ohromné bazény a tady se začíná chladit. Tento proces trvá od 5000 let až milion let. Území, pod kterými se nacházejí takové bazény magmy, mají velký teplotní vzestup, vlastně teplota se zvedá velmi rychle zvětšením hloubky. Taková území jsou velmi výhodná pro využívání geotermální energie.

Potenciál geotermální energie je ohromný, je jí 50 000 krát více, než ostatní energie, která se může dostat z ropy a plynu na celém světě. Geotermální zdroje se nacházejí v širokém spektru hlubin, od mělkých povrchních až více kilometrů hlubokých rezervoárů vroucí vody a páry, která se může přivést na povrch a využít. V přírodě se geotermální energie nejčastěji objevuje ve formě sopek, pramenů vroucí vody a gejzíru. V některých státech se geotermální energie používá už tisícletí ve formě lázní- rekreačního i léčivého koupání.

Ale rozvoj vědy se neomezil jenom na část léčivého využívání geotermální energie, ale také využívání geotermální energie usměrnil k procesu získávání elektrické energie a topení v domácnostech a průmyslových zařízení. Zahřívání budov a využívání geotermální energie v procesu získávání proudu jsou hlavní, ale ne i jediný způsob využívání této energie. Geotermální energie se také může využít i pro jiné cíle jako například ve výrobě papíru, pasterizaci mléka, bazénech na plavaní, v procesu sušení dřeva a vlny, dobytkářství atd. Hlavní nedostatek během využívání geotermální energie je to, že neexistuje mnoho území na

světě, která jsou výjimečně výhodná pro exploataci. Nejvýhodnější jsou území na okrajích tektonických desek, vlastně území velké sopečné a tektonické aktivity.

Největší zdroj geotermální energie:

Riftové údolí v Keni má velké ambice. Rift Valley, velké riftové údolí, které na našich mapách najdete na západě Keni, se má stát největším zdrojem geotermální energie na světě.

Riftové údolí dlouhé *4 000 km* v sobě ukrývá horniny, které jsou stále zahřívány sopečnou činností natolik intenzivně, že by mohly poskytnout *7 000 MW* elektřiny. Při plném využití by se zde získávalo dvakrát víc geotermální energie než na celé planetě. S využitím geotermální energie už má Keňa určité zkušenosti západně od Nairobi, v Olkarském geotermálním poli. Už dvacet let se tu získává *5 %* keňské spotřeby elektřiny. Keňa by chtěla nejprve využívat energie pod sopkou Longonot u Olkarie. Odtud by měla do patnácti let krýt třetinu své potřeby elektřiny. K projektu se chtějí připojit i státy sousedící s Keňou -od Etiopie na severu po Mozambik na jihu. Světová banka a *UN Enviroment Programme* vidí využití této relativně laciné energie jako výtečnou možnost industrializovat tato území, aniž by se zde rozmohlo spalování fosilních paliv a ničila tamní stále ještě bohatá příroda.

Geotermální energie má budoucnost

Pojem "geotermální energie" je u nás většinou používán pro tepelná čerpadla, která využívají teplo půdy nebo velmi mělkých vrtů. Získané teplo je sice levné, ale ochlazení okolní půdy či podzemní vody o několik stupňů (navíc zpravidla v nejchladnějším období roku!) je samozřejmě drastickým zásahem do ŽP, stejně jako je tomu i u čerpadel odebírajících teplo vzduchu.

Snadnou a bezkonflitktní možností využití geotermální energie v ČR jsou některá stará důlní díla. Díky tepelným čerpadlům lze využívat vodu dokonce už od 10°C, neboť půvab dolů spočívá v tom, že jsou schopny teplotu udržet přes celou zimu i v případě, že je odčerpávána a nahrazována chladnější vodou. Ideální jsou příbramské uranové doly (dosahující hloubky až přes 1600 m), kde však bude zatápění ukončeno až za několik let. Možnost využití tepla z téměř stejně hlubokých příbramských polymetalických dolů (které jako první na světě dosáhly hloubky 1 km, a to už v 19. století) byla zkoumána již v 80. letech; poloprovozní zkoušky vytápění Hornického muzea byly z dnešního pohledu v podstatě úspěšné, ale bohužel v té době byla právě dokončena nová teplárna a vzhledem ke špatnému zateplení muzea by výkon stejně nestačil. Nicméně důlní voda už dávno nechladne, tepelná čerpadla i jejich údržba jsou dnes podstatně levnější, zateplování se provádí pořád a navíc se přišlo i na jiné možnosti využití.

V současnosti se využívá teplo důlních vod jen na jedné lokalitě v Jáchymově. V obci Olší na Žďársku se však realizuje další projekt. I v Příbrami se počítá s využitím vody z hloubky "jen" 600 m, takže starých dolů, které přicházejí v úvahu pro podobné využití, je více. Nejnadějnější lokality jsou pravděpodobně na Ostravsku a v Rosicko-oslavanském revíru, beznadějná by však nemusely být ani mnohá místa např. v Krušných horách.

Dopady našeho využívání energie

Znečištění z našeho současného využívání energie zabíjí ročně v Evropě mnoho tisíc lidí, ještě mnohem většímu počtu lidí negativně ovlivňuje zdraví.

Uhelný průmysl například stále více využívá krátkodobých ekonomických výhod povrchové těžby uhlí, zejména když nadcházející liberalizace trhu vede k růstu konkurence. Rozsáhlé oblasti panenské přírody a kvalita života v tisících obcí v celé Evropě jsou devastovány znečištěním vzduchu a vody, prachem, dopravou, výbuchy odstřelů a hlukem strojů v oblastech povrchových dolů.

Ropný a plynárenský průmysl má také vážné dopady na místní životní prostředí. Rybolov, turistický průmysl i volná příroda, to všechno jsou segmenty, které trpí negativními zátěžemi tohoto průmyslu. Čistící operace při ropných haváriích vedou často k ještě dalším škodám na životním prostředí, zpravidla dochází po těchto haváriích k jen částečné nápravě - lze ji vyčíslit na hladině zhruba 10 % - 15 % způsobených škod. Nehody tankerů jsou však příčinou jen asi 6 % z mnoha milionů tun ropy, která ročně vytéká do moře. Miliony tun ropy jsou vypouštěny z ropných plošin, vytékají z podmořských potrubí a z ropovodů, jsou nelegálně vypouštěny z lodí proplachujících si nádrže.

Obavy jsou také z užívání orimulse (suspenze drobných kapének materiálu podobného bitumenu ve vodě) jako paliva, protože jeho i relativně malý únik může způsobit obrovskou škodu mořskému životu a vzhledem ke své vysoké odolnosti a bioakumulativním vlastnostem vede ke kumulaci v potravinovém řetězci.

Na závěr naší práce bychom Vás rádi seznámili s názory lidí, kterých jsme se na geotermální vytápění ptali.

Většina tázaných vůbec nevěděla, o co se jedná. Poté, co jsme jim geoterm přiblížili, nám byly řečeny kladné i záporné názory. Různé a mnohdy velmi zajímavé reakce v nás vyvolaly pocit, že český národ o geotermálním vytápění nemá téměř žádné informace. A to by se určitě mělo změnit. A to hlavně z důvodu, že geotermální energie jako jedna z mála energií nepoškozuje životní prostředí. Ale samozřejmě to má i své negativní stránky.

Pořízení geotermu je velmi nákladné a jen málo lidí by si ho mohlo dovolit. Když stát slibuje, že 60 % vrátí pořizovateli nazpátek, ale na to většina lidí nespoléhá.

Měli bychom se zamyslet nad tím, jak ovlivňujeme negativně životní prostředí a jestli by se to nedalo změnit.

Závěrem chceme říci, že by se o tomto tématu mělo začít více mluvit. Doufáme, že se Vám naše vyprávění líbilo a že Vás vytápění geotermální energií zaujalo stejně jako nás.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

VĚTRNÁ ELEKTRÁRNA NOVÁ VES V HORÁCH

Lucie Šindelářová

Střední zdravotnická škola
Máchova 400, Benešov

Větrné elektrárny - mikro, malé i velké - princip, provedení, regulace

Vedle solárních článků pro potřeby výroby el. energie, jsou k dispozici i další možnosti. Jednou jsou i větrné elektrárny. Pro účely malých výkonů slouží malé a mikroelektrárny malých rozměrů, jednoduché konstrukce a výkony do několika desítek kW. Ráda bych ve své práci stručně popsala funkci, konstrukci, regulaci malých i velkých elektráren.



K místní výrobě elektrické energie, tzn. elektřiny, která není dodána z "centrální" napájecí sítě, mohou sloužit nejen všeobecně známé sluneční články a panely, ale také malé větrné elektrárny. Většina lidí si pod pojmem větrná elektrárna většinou představí často diskutované velké "sloupy" s velkými vrtulemi a s výkony mnoha stovek nebo i tisíců kilowatů. Ty jsou samozřejmě důležité pro globální ekologickou výrobu elektřiny dodávané do elektrické sítě, ale pro malé firmy a lidi jednotlivce až tak zajímavé nejsou. V jejich stínu pak na trhu nalezneme i tzv. větrné mikroelektrárny, které se vyznačují malý rozměry (několik m) a jsou tedy vhodné jako alternativa nebo doplněk solárních panelů pro napájení mobilních nebo v přírodě a krajině osamocených zařízení, jako jsou například měřící stanice či reklamní panely a poutače. O něco větší verze pak mohou v klidu zásobovat elektřinou osamělé sruby, chaty či chalupy, příp. i rodinné domky. Právě hlavně o těchto malých větrných elektrárnách je tato práce.

Výhody a nevýhody napájení větrnou elektrárnou

Výhody:



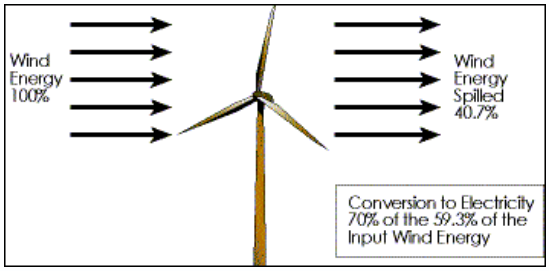
- napájení ve špatně dostupných místech - hory
- relativně velký výkon na rozměry (např. proti solárním článků)
- při zálohování energie je nutná jen omezená doba větru
- jen málo zatěžuje životní prostředí

Nevýhody:

- složitá instalace i v případě mikroelektráren
- použití je na místech s optimálními větrnými podmínkami
- velké elektrárny vzhledem narušují krajinu
- bohužel dost vysoká investice - cena

Princip větrných elektráren

Větrná energie je jedna z forem, do níž se transformuje sluneční záření, neustále dopadající na naši planetu. Vítr je proudění vzduchu, které vzniká tlakovými rozdíly mezi různě zahřátými oblastmi vzduchu v zemské atmosféře. Pokud není uvedeno jinak, rozumí se (i v odborné literatuře) pod pojmem vítr pouze horizontální složka proudění vzduchu. Působením aerodynamických sil na listy vrtule se převádí kinetická energii větru na rotační mechanickou energii rotoru. Ta je poté prostřednictvím generátoru zdrojem elektrické energie (na podobném principu turbogenerátoru např. vodní elektrárny). Prouděním vznikají podél listů velké aerodynamické síly, a proto listy musejí mít speciálně tvarovaný profil, velmi podobný profilu křídel letadla.

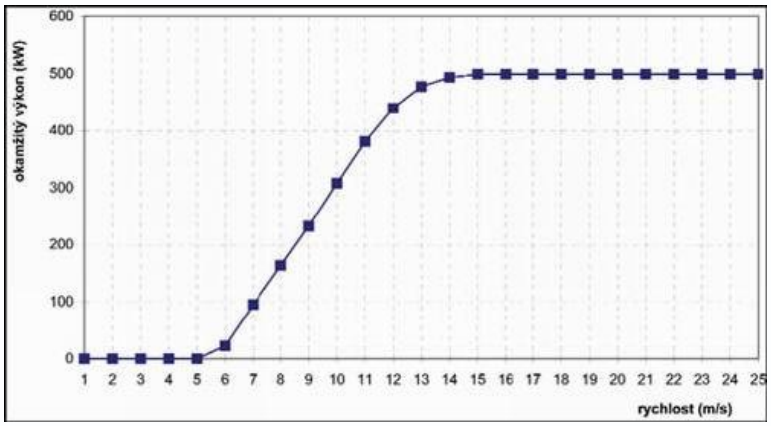


Obr. 1. Větrná elektrárna odnímá kinetickou energii vanoucího větru a převádí ji na energii elektrickou

Se vzrůstající rychlostí vzdušného proudu rostou vztlakové síly s druhou mocninou rychlosti větru a energie vyprodukovaná generátorem s třetí mocninou. Odhad produkovaného výkonu tedy lze stanovit pomocí rovnice

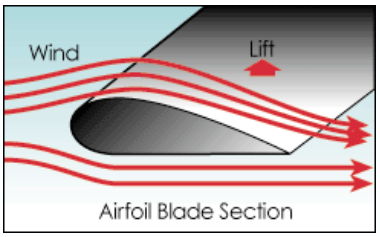
$$P = 0.2 \times v^3 \times D^2,$$

kde P = výkon zařízení, v = rychlost větru, D = průměr vrtule. V praxi však samozřejmě nemůže růst výkon do nekonečna. Existuje vždy konkrétní "výkonový strop", na kterém se již s rostoucí rychlostí proudění výkon neroste. To je zajištěno rychlou automatickou regulaci výkonu rotoru tak, aby se zabránilo mechanickému a elektrickému přetížení větrné elektrárny. Bez ní by lehce mohlo dojít k poškození nebo zničení generátoru a mechanických částí elektrárny.



Obr. 2. Příklad reálné závislosti okamžitého výkonu elektrárny na rychlosti proudění vzduchu

Vrtule je rychloběžný typ větrné elektrárny a má nejvyšší možnou dosažitelnou účinnost ze všech ostatních typů, max. 45%. Listy vrtule ("křídla" elektrárny), kterých zpravidla bývá 1 až 4, jsou vyrobená ze sklolaminátu podle přesně propočítaných profilů. Jejich délka se pohybuje od 30 do 40 metrů, průměr rotoru je tedy 60 až 80 metrů, v poslední době i větší. Při 25 otáčkách/min tak dosahují za provozu konce křídel rychlosti přes 300 km/h! Jejich natáčení kolem vlastní podélné osy je jedním z prvků regulace chodu elektrárny. Fouká-li příliš silný vítr nebo je-li nutné elektrárnu odstavit z jiného důvodu, listy křídel se natočí kolem své osy tak, že vítr rotorem neotočí. Rotor je pak zajištěn brzdou.



Obr. 3. Obtékání vzduchu/větru (wind) okolo listů rotoru elektrárny způsobuje vztlak (Lift) a tím i rotační pohyb rotoru.

Druhy a rozdělení běžných větrných elektráren

Jak již bylo předznamenáno v úvodu, podle velikosti lze větrné elektrárny rozdělit na:

- mikroelektrárny - pro napájení jednotlivých zařízení - nedodávají energii do sítě - výkon do cca 1 kW
- malé elektrárny - pro napájení velkých zařízení nebo stavení - obvykle nedodávají energii do sítě - výkon do cca 15 kW
- středně velké elektrárny - pro napájení několik stavení - obvykle dodávají energii do el. sítě - výkon do cca 100 kW
- velké elektrárny - pro napájení vesnic a měst - vždy dodávají energii do sítě - výkon stovky kW až jednotky MW

Zatímco mikroelektrárny na svém výstupu dávají napětí 12V nebo 24 V a výkon v rozsahu 1 W až cca 1kW a malé elektrárny napětí 230 V (příp. 400 V), s výkonem do cca 15 kW, střední a velké elektrárny již poskytují vyšší napětí, někdy v až řádu kV a výkony až 3 MW na jednu větrnou elektrárnu.



Obr. 4. Typy VT: zleva mikroelektrárna, střední a velké větrné elektrárny

Využitelný potenciál energie větru velmi ovlivňuje typ navržené větrné elektrárny a její výkon. Větrné elektrárny se od sebe liší výtěžností pro určité parametry větru, což vyplývá z konstrukce vrtule/rotoru, typu generátoru a regulace.

Podle aerodynamického principu dělíme větrné elektrárny na:

- Vztlakové s vodorovnou osou otáčení - vítr obtéká lopatky s profilem podobným letecké vrtuli. Na podobném principu pracovaly již historické větrné mlýny, nebo tak pracují větrná kola vodních čerpadel (tzv. americký větrný motor). Při stejném průměru rotoru v zásadě platí nepřímá závislost počtu listů a frekvence otáčení. Moderní elektrárny mají obvykle tři listy, byly však vyvinuty i typy s jediným nebo se dvěma listy.
- Odporové - pracují na vztlakovém principu, kdy existují také elektrárny se svislou osou otáčení, některé pracují na odporovém principu (typ Savonius, jako misky anemometru) nebo na vztlakovém principu (typ Darrieus). Výhodou elektráren se svislou osou pracujících na vztlakovém principu je, že mohou dosahovat vyšší rychlosti otáčení a tím i vyšší účinnosti; není je třeba natáčet do směru převládajícího větru. Elektrárny se svislou osou otáčení se v praxi příliš neuplatnily, neboť u nich dochází k mnohem vyššímu dynamickému namáhání, které značně snižuje jejich životnost. Nevýhodou je malá výška rotoru nad terénem, tj. i menši rychlost větru. V praxi se téměř nepoužívají.

Systém natáčení do směru větru používá několik způsobů:

- Ocasní plocha - hlavně u mikroelektráren, příp. malých elektráren - jednoduché mechanické řešení
- Boční pomocné rotory - malé a střední elektrárny
- Natáčení pomocnými motory - střední a velké elektrárny

K přeměně kinetické energie větru na elektrickou energii slouží generátory. Používají se tyto druhy:

- Stejnosměrné – jsou vhodné pouze pro mikroelektrárny, které produkují stejnosměrné napětí 12 nebo 24 V.
- Asynchronní – produkují střídavý proud a napětí => jsou připojitelné k síti. Nevyžadují složitý připojovací systém - pouze se sledují otáčky a rozhoduje o okamžiku připojení k síti.

- Synchronní – jsou vhodné pro malé, střední i velké větrné elektrárny - mají velkou účinnost. Mnohapólové generátory jsou schopny pracovat s velkým rozsahem rychlostí větru i bez převodovky.

V současné době převládají dva typy regulace výkonu v závislosti na rychlosti větru:

- Regulace Stali (pasivní) - rotor elektrárny má pevné listy a pro regulaci využívá odtržení proudnice vzduchu od listu rotoru při určité rychlosti větru. Po odtrženi dojde ke snížení výkonu. Výhody jsou o něco vyšší výroba elektrické energie při vyšších rychlostech větru s větrnými nárazy a nižší pořizovací náklady. V současné době se používá i aktivní varianta regulace typu Stali, která spočívá v mírném pomalém aktivním natáčení listů v závislosti na okamžitých klimatických podmínkách, např. hustotě vzduchu.
- Regulace Pitch (aktivní) - využívá natáčení celého listu rotoru podle okamžité rychlosti větru tak, aby byl celkový náběh větrného proudu v daném okamžiku optimální (dosažení nejvyšší výroby). Výhodou je vyšší výroba elektrické energie zejména při nižších rychlostech větru, kdy se optimalizace projeví nejvíce. Nevýhodou jsou vyšší pořizovací náklady.

Provedení a funkce jednotlivých typů:

Větrná mikroelektrárna a malá elektrárna

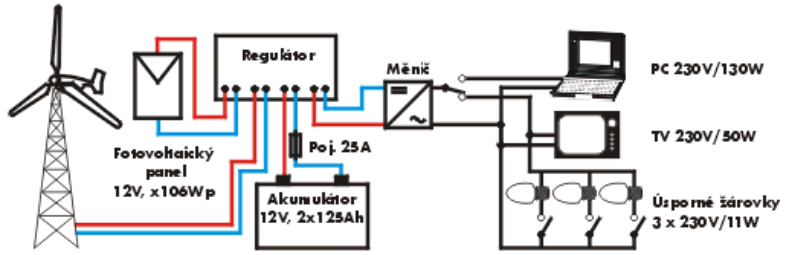
Tyto typy větrných elektráren jsou pro osobní použití nebo malé firmy velmi vhodné. Jsou takovou alternativou k slunečním článkům v místech, kde je dost větrno a naopak méně svítí slunce. Mikroelektrárny s malými výkony (cca 100 W) tak mohou například napájet osvětlení reklamních panelů podél dálnic, aktivní inteligentní dopravní značky, měřiče teploty a hodiny apod. Jejich skládací verze mohou sloužit v přírodě jako mobilní nabíječe akumulátoru, napájení osvětlení, vařiče, malého topení, vysílačky, počítače nebo televizoru.



Obr. 5. Příklad provedení a použití mikroelektráren

Mikroelektrárny, které dosahují výkonů až několik jednotek kW, již mají pevnou instalaci a mohou v klidu napájet chaty. Proč tedy toho nevyužít, když dům stojí na kopci, kde je často stabilní rychlost větru ? Jediným protiargumentem je stále vysoká cena, která se však s rostoucí životností moderních větrných elektráren vrátí. To je lákavá představa, ale co dělat, když zrovna nefouká ? Samozřejmostí je, že se el. energie větrné elektrárny akumuluje do akumulátorů, které pokrývají spotřebu při špičkových zatíženích nebo když nefouká. Navíc je možné ji doplnit o solární články/panely a vhodným řídícím systémem, který přerozdělování výroby el. energie automaticky inteligentně řídí. Tato kombinace je zdá se ideální, protože když nefouká, tak často svítí slunce a obráceně. Na překážku takovému řešení je pak opět hlavně cena, protože tato kombinace je již opravdu dost nákladná. To platí však v současné době, ale s větším rozšířením se dle zákonitostí ekonomiky cena bude automaticky snižovat. Až se ekologie stane módou a módním stylem hlavně

u bohatých lidí, nebude k rozšíření kombinace sluneční a větrné elektrárny na úrovně jednotlivých domácností již nic stát v cestě.

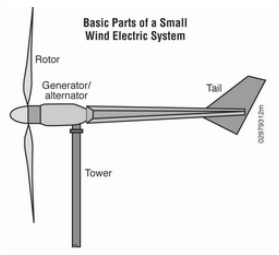


Obr. 6. Princip využití mikroelektrárny pro napájení běžných spotřebičů rodinného domku či chaty

Celosvětově lze již nalézt na poli mikroelektráren docela slušnou nabídku. Český trh však stále trošku pokulhává. Prostě jsme malá republika a síťové napájecí kabely 230 V jsou nataženy prakticky až na výjimky do všech koutů naší vlasti. Proto zde není ani motivace daná nutností, jako je tomu na zapadlých samotách, například amerického venkova. Generátor mikroelektrárny obvykle poskytuje výkon mezi 1 W až cca 1 kW a bateriová napětí 12, případně 24 V.

Tzv. větrná mikroelektrárna je složená z následujících mechanických částí:

- rotoru s listy
- generátoru nebo alternátoru zamontovaného v rámu (frame)
- stabilizačního a natáčecího ocasu (tail)
- upevňovacího podstavce/tyče (tower)

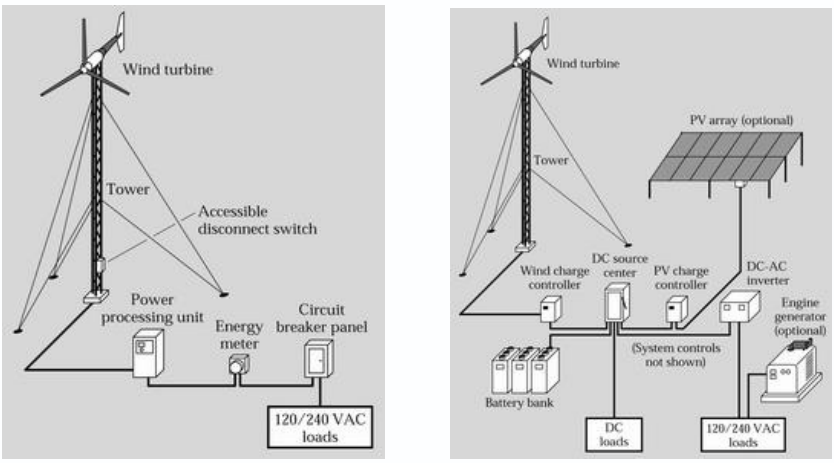


A elektrických částí - regulačního systému:

- řídící jednotky (controller)
- invertorů (inverters)
- akumulátorů/baterií (batteries)

Dnešní mikroelektrárny jsou obvykle v horizontálním provedení s rotorem se dvěma až čtyřmi listy z kompozitového materiálu (skleněná vlákna). Průměr rotoru definuje tzv. "zametací oblast (swept area)" nebo množství zachyceného vzduchu (the

quantity of wind intercepted by the turbine). Ocas (tail) provádí natáčení a stabilizaci rotoru ve větru.



Obr. 13,14. Výroba elektřiny pouze malou větrnou elektrárnou (vlevo) a kombinace větrné elektrárny a solárních panelů (vpravo)

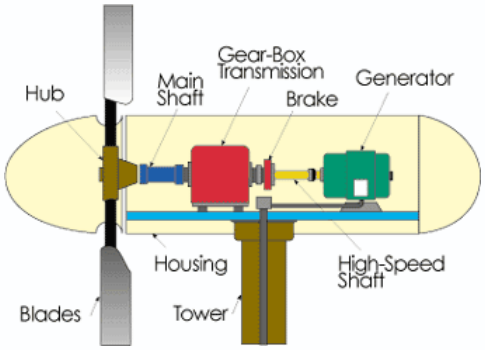
Malé elektrárny



Malé větrné elektrárny již poskytují výkony i mnoho jednotek kW, což již na spotřebu velké chaty nebo běžného rodinného, dobře zatepleného domku stačí. Například výkon od 1 kW již plně postačuje na čerpání vody ze studně a její rozvod do kohoutků v objektu. Tyto elektrárny často vyrábí elektřinu pomocí synchronních generátorů buzených permanentními magnety s výstupním napětím 24 V nebo klasických 230 V, příp. 400V.

Pro správnou volbu je nutné správně spočítat spotřebu a zvolit výkon elektrárny, resp. jejího generátoru/turbíny. Takový typický domek má roční spotřebu cca 9400 kWh za rok, čemuž odpovídá cca 780 kW za měsíc. V závislosti na průměrné rychlosti větru je vhodné volit výkon elektrárny 5 až 10 kW. Výkon 1,5 kW v místě s průměrnou rychlostí větru cca 6 m/s pak za měsíc vygeneruje cca 300 kWh. Z pohledu konstrukce mohou vypadat různě. Zatímco malé elektrárny s výkonem okolo 1 až 5 kW mohou ještě vypadat jako "větší" mikroelektrárny - viz. obrázek 6., konstrukce pro výkony na 10 kW již někdy vypadají jako zmenšeniny těch středních a

velkých, tzn. mají již gondolu vybavenou převodovkou (Gear-Box), brzdou (Brake) a generátorem (Generator) připojené přes hřídel (Main Shaft) na rotor vrtule s listy (Blades). Vše je pak připevněno na sloupu (Tower), kterým vedou výkonové a signálové kabely - viz obrázek 8.



Obr. 16. Detail gondoly/strojovny střední větrné elektrárny

Střední a velké elektrárny



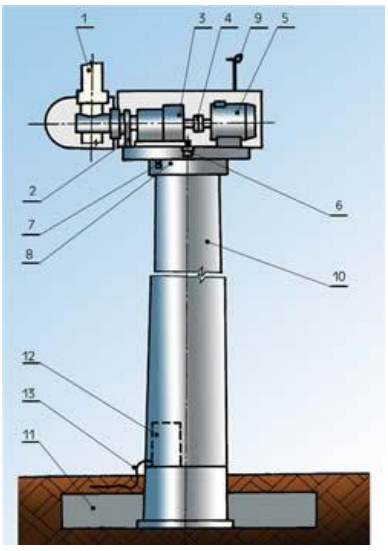
Elektrárny velkých výkonů (300 až 3000 kW) jsou určeny k dodávce energie do veřejné rozvodné sítě. Mají asynchronní nebo synchronní generátor, který dodává střídavý proud o napětí 660 V a vyšších, a tudíž nemohou pracovat jako autonomní zdroje energie. Existují i elektrárny se speciálním mnohapólovým generátorem, který nevyžaduje převodovou skříň. Většina elektráren má konstantní otáčky regulované natáčením listů a proměnným převodovým poměrem převodovky. Některé typy mají i dvě rychlosti otáčení. Některé střední elektrárny mohou však mít i proměnné otáčky podle okamžité rychlostí větru. V praxi se používají většinou větrné elektrárny s horizontální osou rotace a u velké elektrárny mají průměr rotoru až 80 m a věž o výšce více než 80 metrů. Trendem poslední doby je zvětšování výkonu větrných elektráren a zvyšování stožárů. Nejnovější zařízení instalovaná ve světě pracují s generátorem o výkonu až 3 MW, který je na tubusu dosahujícím výšky kolem 100 metrů. Důvodem jsou nižší měrné náklady na výrobu energie a optimální využití lokalit, kterých je omezený počet. K zefektivnění provozu

a snížení nákladů na projektování a výstavbu se velké elektrárny sdružují do skupin (obvykle 5 až 30 elektráren) tzv. větrných farem.

Struktura střední a velké větrné elektrárny je velmi podobná a skládá se z níže uvedených částí. Rozdíl je často jen ve velikosti a dimenzování mechanických částí a pak v provedení gondoly/strojovny a samotné věže. Velké elektrárny mají dutý tubus věže se schody či výtahem a velkou strojovnu - viz obrázek 9.

Popis částí velké větrné elektrárny:

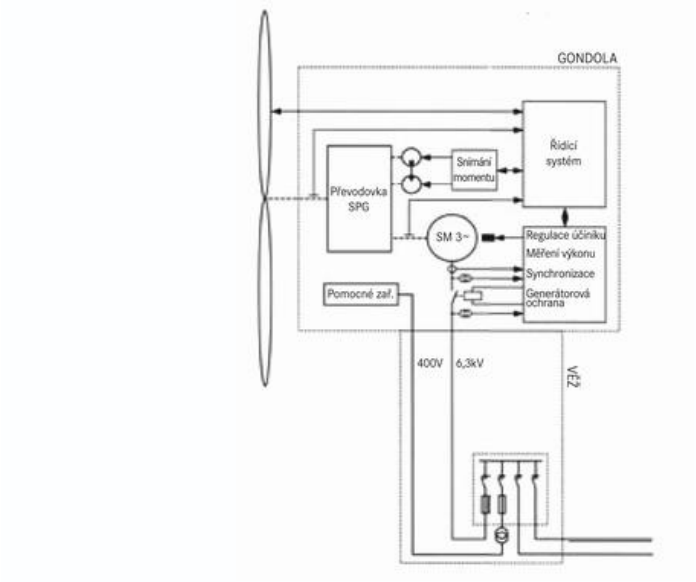
1 - rotor (main shaft) s rotorovou hlavicí (hub) a listy (blades)
2 - brzda rotoru (brake)
3 - planetová převodovka (Gear-Box Transmision)
4 - spojka (cluth)
5 - generátor (Generator)
6 - servo-pohon natáčení strojovny
7 - brzda točny strojovny (Housing)
8 - ložisko točny strojovny
9 - čidla rychlosti a směru větru
10 - několikadílná věž elektrárny (Tower)
11 - betonový armovaný základ elektrárny
12 - elektrorozvaděče silnoproudého a řídícího obvodu
13 - elektrická přípojka



Obr. 18. Struktura velké větrné elektrárny

V dnešních moderních velkých elektrárnách se využívá vyspělé automatické regulace, například sestávající se ze systému individuální regulace natáčení listů rotoru (Individual Pitch Control – IPC) v kombinaci s převodovkou s integrovaným systémem proměnlivého převodového poměru (Super Position Gear – SPG - viz obrázek 10). Toto řešení poskytuje úplnou kontrolu nad přenosem energie získávané z vysoce proměnného prostředí větru a umožňuje výrobu elektrické energie pomocí standardního synchronního generátoru s konstantními otáčkami. Individuální regulace natáčení listů rotoru (IPC) zajišťuje dynamické nastavení optimálního úhlu jednotlivých listů pro maximální účinost získávání větrné energie v samém počátku. Další fáze regulace je prováděna převodovkou s integrovaným systémem proměnlivého převodového poměru (SPG), která umožňuje plynulou změnou převodového poměru udržovat konstantní otáčky synchronního generátoru v celém rozsahu pracovních otáček rotoru větrné elektrárny. Regulace SPG umožňuje provozovat větrnou elektrárnu při otáčkách rotoru určovaných rychlostí větru a zároveň udržovat konstantní výstupní otáčky pro pohon generátoru. Funkce je

zajištěna nastavitelným zdvihem hydraulické jednotky poháněné hřídelí generátoru. V případě, že poryvy větru způsobí nárůst přenášeného kroutícího momentu ve vztahu k daným otáčkám, potom automaticky dojde ke snížení převodového poměru a tím je umožněno rychlejší otáčení rotoru elektrárny v závislosti na síle a rychlosti větru. Toto umožňuje pohlcovat náhlé výkyvy větrné energie akcelerací rotoru na vyšší otáčky a zároveň tak dochází k ochraně celého pohonného soustrojí před destruktivními výkyvy přenášeného kroutícího momentu.



Obr. 19. Příklad blokového schéma regulace velké větrné elektrárny WIKOW

Příklad reálných vlastností - Větrná elektrárna AIRCON 10

Technická data:



- Jmenovitý výkon: 10 kW
- Průměr rotoru: 7,40 m
- Plocha rotoru: 40 m2
- Spouštěcí rychlost: 2.5 m/s
- Nominální rychlost větru: 11 m/s
- Vypínací rychlost větru: 37 m/s
- Kritická rychlost větru: 58 m/s
- Otáčky: 70-160 ot/min
- Regulace otáček: Pitch 1
- Brzdný systém: Regulace přetížení generátoru, 2. brzdný systém: Hydraulická kotoučová brzda
- Alternativní přestavění listů
- Regulace větru: Aktivní řízení počítačem
- Monitorování regulace větru: Mikropočítač
- Generátor: Permanentně nabuzený synchronní měnič IGBT, čtyřkvadrantový s signálovým procesorem DSP
- Nominální výkon: 12.5 kVA
- Nominální napětí: 400 V
- Účiník (power factor - cos q): 0.98
- Minimální harmonické zkreslení
- Dálková regulace výkonu, rozsahu otáček, dálkové monitorování přes telefonní / rádiovou / internetovou síť přes počítačové rozhraní RS232 nebo RS485

Galerie větrných elektráren v přírodě









Nová Ves v Horách

V současnosti stojí v katastru obce Nová Ves v Horách dvě funkční větrné elektrárny a na podzim 2005 byla zahájena výstavba dalších čtyř větrníků (zatím byly postaveny pouze jejich základny). Jedna elektrárna, stojící zde od roku 1994, je nyní mimo provoz a byla již částečně demontována. V roce 2007 byla postavena jedna větrná elektrárna v katastru obce Mníšek (ve správě obce Nová Ves v Horách). Vlastníky všech staveb jsou privátní osoby, nikoliv obec.



Souhlas s výstavbou takových staveb je obvykle podmíněn uzavřením vzájemného smluvního vztahu mezi obcí a investorem. Na základě toho bude investor odvádět pravidelné finanční částky do obecního rozpočtu. Tyto prostředky budou následně použity na rozvoj obce.

V roce 2005 zahájila obec přípravné práce na vlastním projektu výstavby jedné větrné elektrárny.

V souvislosti s projednáváním konceptu územního plánu byla v únoru roku 2006 zdejšímu obecnímu úřadu doručena *petice* proti další výstavbě větrných elektráren. Následně byla obcí vyhlášena anketa mezi občany. Každý obdržel anketní lístek, aby měl možnost se k problematice budoucí výstavby vyjádřit. Většina doručených anketních lístků zněla PRO výstavbu větrníků.

V únoru 2007 zastupitelstvo obce rozhodlo o zařazení do územního plánu pouze těch staveb větrných elektráren, které k tomuto datu měly pravomocné stavební povolení. V říjnu 2007 zastupitelstvo obce svůj kategorický postoj změnilo a přijatým usnesením rozhodlo, že do územního plánu bude větrné elektrárny povoleno zakreslit. Územní plán obce Nová Ves v Horách se ještě zpracovává, probíhají schvalovací procesy. Konečné schválení se předpokládá na počátku roku 2008.

Nová Ves v Horách






Závěr

Cílem této práce bylo obecně přiblížit oblast výroby el. energie prostřednictvím větrných elektráren a ukázat, že mimo v médiích probíraných velkých elektráren, existuje i skupina těch malých. Myslím si, že pro potřeby ekologického nezávislého napájení přístrojů a zařízení, je použití malých elektráren na některých místech docela výhodné. Navíc při kombinaci se solárními články je možné zajistit vcelku spolehlivý zdroj. V současné době sice jde o finančně náročný zdroj (400W mikroelektrárna stojí cca 30 000 Kč), ale patrně v budoucnosti by mohlo jít o více rozšířený způsob výroby elektřiny. V oblasti výstavby velkých větrných elektráren se často vedou diskuse o vlivu na přírodu, kde mohou kvůli vysokému stožáru a velkému průměru rotoru negativně narušit optický reliéf krajiny a hlukem plašit zvěř. U nových typů je však již konstrukce podřízena velmi přísným požadavkům omezení hlučnosti, a to jak mechanické (převodová skříň, generátor) tak aerodynamické (rotor). Téma větrných elektráren tak bude jistě velmi aktivní i v budoucnu. Pro více zajímavých informací odkazuji na články uvedené níže.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

STIRLINGŮV MOTOR - PREZENTACE SOUTĚŽE

Jakub Horníček, Norman Csiszár

SPŠS Betlémská

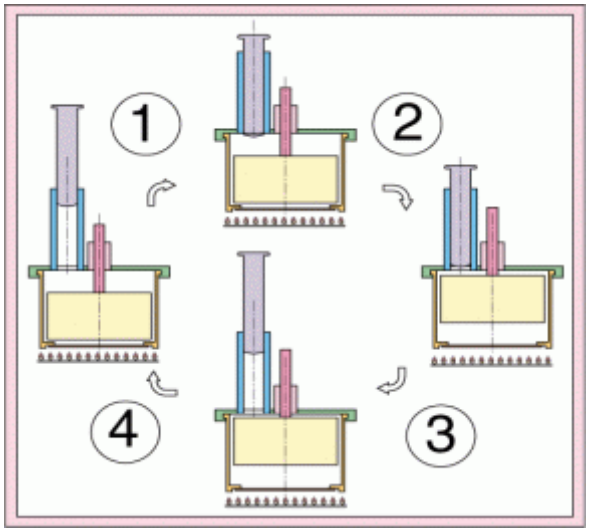
Praha 1, Betlémská 4

Úvod

Soutěže se zúčastnily čtyři pražské průmyslové školy Průběh soutěže je popsán na internetovém blogu **www.betlemska-stirling.blog.cz**

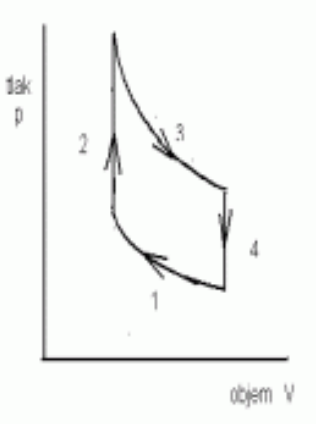
Zadání soutěže „Vyrob si svůj Stirlingův motor“

Jaký to je motor? - Před 170. léty, v době vzniku naší školy, výbušné spalovací motory neexistovaly. Těžkým strojům vládly parní motory, menší výkony byly zastoupeny teplovzdušnými motory, které vynalezl Skot Stirling v roce 1816.



Princip: Uzavřený objem vzduchu (plynu) je na jedné straně motoru ohříván, na druhé ochlazován. Přeháněcí píst přehání vzduch z jedné strany na druhou. Uvnitř motoru vzniká střídavě relativní přetlak a podtlak, které působí na pracovní píst. Ten pohání klikovou hřídel, od níž je odvozen i pohon přehaněče. Čím je větší teplotní rozdíl teplé a studené strany, tím větší je práce jednoho oběhu a tím i výkon motorku.

Teoretický oběh Stirlingova motoru tvoří dvě izotermy a dvě izochory a jeho účinnost je relativně vysoká. Využití tepla na vstupu je však nízké a proto je výkon motorku malý.
V dnešní době s propracovanou konstrukcí, vysokými teplotami a odolnými materiály Stirlingův motor začíná opět nacházet své uplatnění. Je též velmi perspektivní s příkonem obnovitelnými zdroji energie



Pravidla soutěže:
Soutěžící si z obdržených dílů sestaví Stirlingův motorek. Některé součásti ovšem musí být dokončeny. Kterákoliv z nich může být nahrazena jinou - vlastní. Musí být zachovány pouze dva parametry: Průměr komory - D = 100 +/- 2mm a výška komory H = 62 +/- 2 mm. To je zaručeno běžnou plechovkou od konzervy. Základním kriteriem při hodnocení jsou otáčky motoru. Budou měřeny bezdotykovým digitálním otáčkoměrem při stálém plameni plynového hořáku. Doma je možno motor zkoušet na nejmenším plynovém vařiči nastaveném na nejmenší plamen (plamen volte menší, než odpovídá bezpečnostnímu dorazu hořáku).
Při vlastní soutěži budou motorky postupně postaveny nad plamen. Po zahřátí a rozběhu (doba asi 2 min - pro všechny stejná) budou měřeny otáčky po dobu 30 s. Největší otáčky budou zaznamenány. Dále během soutěže budou ohodnoceny provedení a vzhled motorku (design) a také technické "zlepšováky", které student vyrobí.
Soutěž proběhne ve dvou kolech v laboratořích KOM (na plynovém kahanu). První kolo bude v pondělí 26. února od 10.00hod, druhé v pondělí 26. března ve stejnou dobu. (mezi koly bude čas na vylepšování). Hodnoceny budou lepší výsledky z obou kol.
Po soutěži proběhne exhibice v učebně technologie pro žáky školy (termín bude upřesněn) Motorky poběží na vlastních plynových vařičích.

Odměny skvělé - ve třech kategoriích

a) Nejvyšší otáčky
b) Design a provedení
c) Technická vylepšení

POZNÁMKY K PREZENTACI SOUTĚŽE

1) Poznámky k zadání úlohy:
   Zadání úlohy obsahuje výkres sestavení Stirlingova motoru v jeho nejjednodušším provedení.
   Dále součástky, z nichž některé jsou hotové (např. hřídelka) a některé se musí dokončit (např. kliky).
   Naším hlavním úkolem bylo dokončit součásti a sestavit z nich motor, který by se otáčel co nejrychleji.
   Věděli jsme, že hodnoceny budou též provedení a různéá vylepšení. Plechovku od konzervy správných
   rozměrů jsme si museli sehnat sami.

2) Poznámky k provedení úlohy:
   Práci jsme prováděli ve volném čase zejména ve školních dílnách – v zámečně. Drobnosti potom doma.
   Dokončování dílů a sestavování motoru znamenalo – řezat, pilovat, vrtat, zkružovat, ohybat, vyrábět závity, lepit a montovat. Jako pomůcku jsme měli internetový blog , kde jsou zaznamenané fotografie a zkušenosti z loňského ročníku této soutěže.
   Při práci na motorku jsme se snažili nejprve o to, aby motor fungoval a teprve potom jsme dělali úpravy,
   aby pracoval lépe. První motorek se nám k naší veliké radosti rozběhl asi po dvanácti hodinách práce.

3) Poznámky k řešení chladiče:
   Stirlingův motor je teplovzdušný motor pracující na principu rozdílného tlaku chladnějšího a teplejšího vzduchu, který pohybuje pracovním pístem. Přeháněcí píst přitom přehání vzduch z z ohřívané strany na ochlazovanou a naopak. Čím je větší teplotní rozdíl, tím je větší práce oběhu (viz diagram) a tím je více energie na překonání pasivních odporů a tím jsou větší otáčky. Z možností chlazení vodou a vzduchem jsme volili vzduch jako jednoduší. Jako chladiče jsme použili hliníková žebra z chlazení počítače. Žebra nejsou lepená plošně aby se nezvětšoval tepelný odpor, ale jsou přitisknutá a přitmelená po obvodu chladiče.

4) Poznámka k vyvažování:
   Oba klikové mechanismy je dobré vyvážit (i když jsme si zpočátku mysleli, že to bude zbytečné). Zvolili jsme ten nejjednoduší způsob – pomocí magnetu, který se může plynule posouvat na poloměru setrvačníku.
   (moment závaží k ose otáčení musí být stejný, jako moment tíhy klikového mechanismu)
   Takto se dá motor pomocí dvou menších magnetů dobře staticky vyvážit a potom při vyšších otáčkách nevibruje.

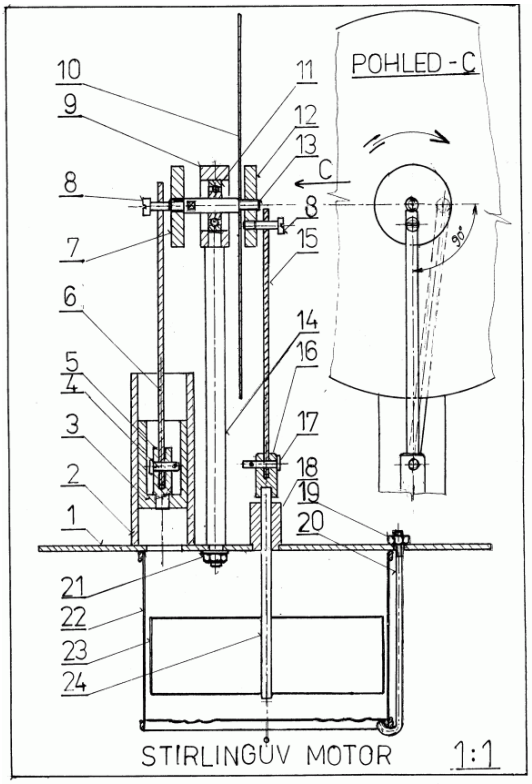
Přínos soutěže

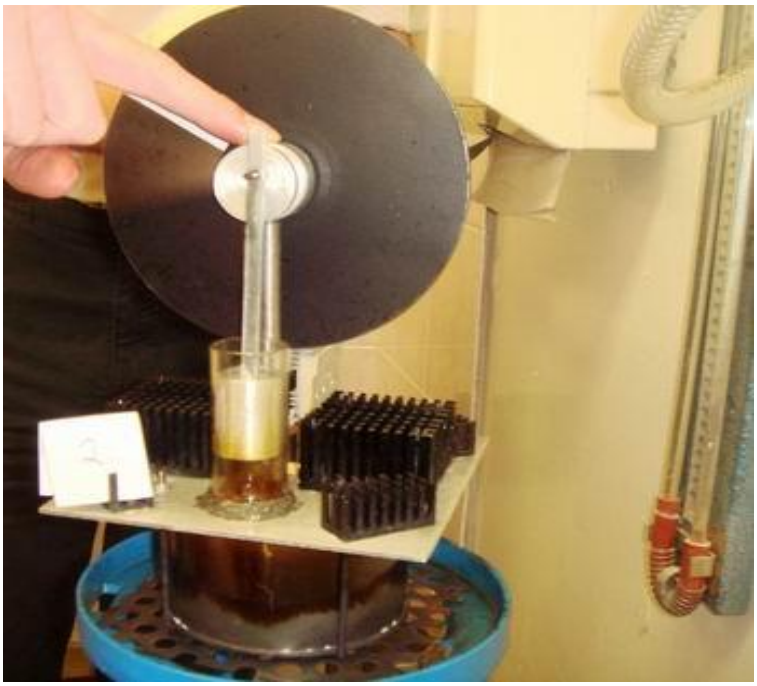
1) Při práci na součástech jsme si uvědomili, jak moc záleží na každém detailu, nic se nedá udělat povrchně, každá chyby se později projeví ve funkci celku.
2) S mechanickými problémy jsme se seznamovali postupně a některé jsme řešili vyrobením některé součásti znovu.
3) Přesvědčili jsme se, že tepelná mechanika má veliký vliv na chod stroje.
4) Uvědomujeme si, že na našich funkčních motorech by bylo stále co vylepšovat.

5) Při vlastní soutěži jsme se seznámili s tím, jak jiní soutěžící řešili stejný problém. A někteří mnohem lépe.
6) Z hlediska našeho učiva ve škole jsme si vlastníma rukama vyrobili
   - dva klikové mechanismy, zkrácený a nezkrácený.
   - uložení hřídele na valivém ložisku
   - setrvačník i s vývažky.
   - klikové a pístní čepy
   - speciální šrouby pro komoru

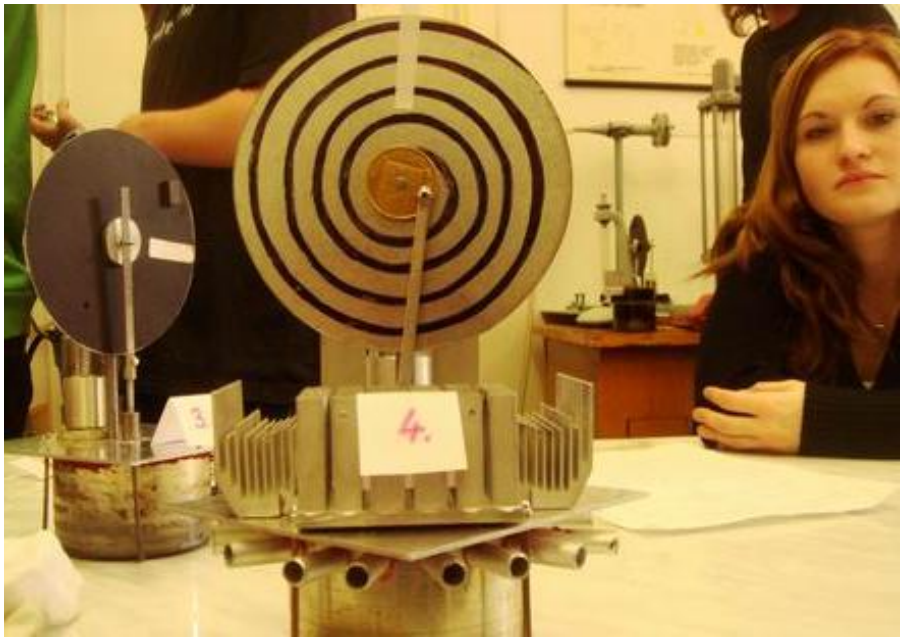
   Největší přínos soutěže byla radost, že jsme schopni vyrobit si nějaký opravdu funkční stroj vlastníma rukama

Stirlingův motor – schéma





Stirlingův motor při rozbíhání nad kahanem – foto



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

MODERNÍ ANALYTICKÉ METODY V KONTROLE ŽIVOTNÍHO PROSTŘEDÍ

Jindra Weber, Veronika Bittmannová

Střední průmyslová škola chemická
Vranovská 65, Brno

Odborní konzultanti:
Mgr. Jiří Machát, Ph.D.
RNDr. Hana Kozáčková

Annotation
In this scientific schoolwork we tried to determine the amount of heavy metals in moss and fungi taken from different localities. The method of Inductively Coupled Plasma Mass Spectrometry ICP-MS was used during our research.We aimed not only at the defining of the amount of heavy metals in moss and fungi from various localities, but also at the defining of the differences of the amount of heavy metals in moss and fungi taken from the same locality.

Anotace
V naši práci, jsme se za použití metody hmotnostní spektrometrie s indukčně vázaným plazmatem ICP-MS rozhodli zkoumat obsah těžkých kovů v mechu a v houbách z různých lokalit. Zaměřili jsme se jak na obsah těžkých kovů v mechu nebo houbě z těchto lokalit, tak rozdíl obsahu těžkých kovů v mechu a houbě v jedné lokalitě.

Poděkování
Chtěli bychom poděkovat panu Mgr. Jiřímu Machátovi Ph.D. za pomoc při praktickém měření a za poskytnutí odborné pomoci při zpracování této práce.
Děkujeme též paní profesorce RNDr. Haně Kozáčkové za provedení korektury vlastního textu práce.

Obsah

1. Úvod a cíl práce ……..

2. Teoretická část ……...
Kovy v životním prostředí……..
2.1.1. Toxické kovy …….
2.1.2. Cesty expozice .....
2.2. Vzduch a půda ……...
2.2.1. Způsob hodnocení kvality a obsahu TK ve vzduchu a půdě ……..
2.3. Biomonitoring ……..
2.4. Mechy a houby ……..
2.4.1. Mechy ……
2.4.2. Houby……..
2.5 Metody stanovení TK ……...
2.6 ICP – MS.....…….
2.6.1. Princip metody……....
2.6.2. Využití metody …….....

3. Experimentální část ……..
3.1. Vzorky a jejich odběr ……..
3.2. Příprava vzorků ……
3.3. Stanovení obsahu TK ve vzorcích ……..

4. Výsledky a diskuze ……..
4.1. Vzorky a jejich zpracování …… .. 1
4.2. Mechy ……..
4.2.1. Výsledky stanovení TK v biomase ……
4.2.2. Srovnání lokalit a jejich oznámkování ……..
4.2.3. Korelace prvků a rozdělení prvků do skupin podle znečištění.....
4.3. Houby ……...
4.3.1. Výsledky stanovení obsahů TK v biomase ……
4.3.2. Korelace prvků ……..

5. Závěr…….

6. Seznam použité literatury a internetových stránek ....……..

1. Úvod a cíl práce

V dnešní době je životní prostředí člověkem opravdu velmi znečišťováno například výfukovými plyny z aut, vypouštěním škodlivých látek z továren do ovzduší nebo řek, atd.

Mezi látky znečišťující životní prostředí patří mimo jiné také těžké kovy, které jsme stanovovali. Těžké kovy jsou všude kolem nás ve vodě, půdě, vzduchu i v potravinách. Zdravotní rizika můžou být pro organismus při vyšší koncentraci než je optimální velmi velká, od bolestí hlavy, přes křeče až po chudokrevnost nebo i smrt. Je proto důležité, aby byl jejich obsah ve složkách životního prostředí kontrolován. Možnými cestami expozice organismu těžkými kovy jsou například vzduch a půda, na které jsme se zaměřili v naší práci.

Možnosti jak kontrolovat hladiny těžkých kovů v těchto matricích jsou různé. Jednou z nich je tzv. biomonitoring, tzn. využití živých organismů, které fungují jako vzorkovače. Analýzou těchto organismů zjistíme míru expozice, odhad velikosti rizika expozice těžkými kovy.

Pro hodnocení kvality ovzduší se využívají mechy, které získávají živiny jen ze srážek, takže obsah kovů v nich vypovídá o znečištění ovzduší. Pro hodnocení kvality půdy lze využít houby, které čerpají živiny z půdy. Pro stanovení obsahu těžkých kovů v těchto organismech se využívají různé metody prvkové analýzy. V současné době je moderní a využívaná metoda multielementární analýzy ICP-MS, kterou jsme využili v naší práci.

2. Teoretická část

2.1. Kovy v životním prostředí

2.1.1. Toxické kovy

Toxické kovy jsou kovy (antropogenního nebo přirozeného původu, avšak ve vyšší koncentraci než je optimální) mající toxické účinky. Mezi toxické kovy jsou řazeny i tzv. **těžké kovy** (všechny kovy o hustotě vyšší než 5.10^3 kg.m^{-3}**,** např. kadmium, rtuť, olovo, železo, kobalt, měď, mangan, molybden, zinek a další). V živých organismech jsou přítomny ve stabilních organických komplexech.

Těžké kovy jsou v malém množství přirozenou součástí půdy. Do životního prostředí se nadměrně dostávají především vinou člověka. Z půdy nebo zdrojů vody pak přecházejí do potravy. Zdravotní problémy mohou působit nejen lidem, ale i zvířatům a rostlinám.

Kadmium je měkký, lehce tavitelný, toxický kov. Je to součást různých slitin a používá se k ochraně jiných kovů před korozí. Kadmium patří mezi několik málo prvků, jejichž vliv na zdravotní stav lidského organizmu je jednoznačně negativní. Postihuje především ledviny a játra. Při inhalační intoxikaci se během několika hodin objeví příznaky podráždění dýchacích cest a plic. Dlouhodobý vliv kadmia na organismus se projevuje také osteoporózou (lidově řídnutí kostí) a anémií (chudokrevnost), zvyšuje se i riziko srdečních a cévních onemocnění.

Rtuť je těžký, toxický, kovový prvek. Slouží jako součást slitin (amalgámů) a jako náplň různých přístrojů (teploměry, barometry). Je jediným kovem, který je za normálních podmínek kapalný. Rtuť patří také mezi prvky, jejichž vliv na zdravotní stav lidského organismu je jednoznačně negativní. Dnes se vyskytují především průmyslové otravy (vdechováním rtuťových par). Z organismu se vylučuje jen velmi pozvolna a obtížně, jeho většina se přitom koncentruje především v ledvinách a v menší míře i v játrech a slezině. Projevy chronické otravy bývají často nespecifické – od studených končetin, vypadávání vlasů, přes zažívací poruchy, různé neurologické a psychické potíže až po závažné stavy jako je například chudokrevnost. *Toxicita* jednotlivých sloučenin je závislá především na jejich rozpustnosti ve vodě. Z tohoto pohledu jsou nejvíce rizikové sloučeniny dvojmocné rtuti Hg^{+2}, které jsou nebo spíše bývaly užívány jako jedy pro hubení hlodavců a jiných zemědělských škůdců. Naopak toxicita samotné elementární rtuti je prakticky nulová, protože jen obtížně vniká do organických tkání. Mnohem škodlivější jsou její páry, které se však do atmosféry dostávají velmi pomalu (bod varu rtuti=356,73 °C).

Olovo je těžký toxický kov, který je znám lidstvu již od starověku. Má velmi nízký bod tání a je dobře kujný a odolný vůči korozi. Toxicita olova je zvláště významná pro dětský organismus. Trvalá expozice dětského

organismu i nízkými dávkami olova je příčinou zpomalení duševního vývoje a nepříznivých změn v chování.V současné době je díky používání olova v rozvodu pitné vody, širokému použití olova při výrobě barev, rozsáhlému použití olova jako aditiva v benzínu a díky jeho ostatnímu využití v průmyslu je olovo všudypřítomným kontaminantem prostředí. Olovo se po vniknutí do organismu ukládá hlavně v kostech a v určitém množství se nachází v krvi. Typickými příznaky otravy olovem jsou bledost obličeje a rtů, zácpa a nechuť k jídlu, kolika, anémie, bolesti hlavy, křeče, chronická nefritida ledvin, poškození mozku a poruchy centrálního mozkového systému. Významné množství je rychle inkorporováno do kostí a erytrocytů (červené krvinky), postupně do všech tkání. Toxický účinek se projevuje při biosyntéze hemu (nebílkovinná složka krevního barviva, hemoglobinu) a následnou anémií (chudokrevnost). Poškozuje renální tubulární buňky tím, že se váže na cystein v bílkovinách a mění jejich terciální strukturu. Váže se též na keratin ve vlasech

2.1.2. Cesty expozice

Existují různé cesty expozice organismů těžkými kovy od vody, kterou pijeme, přes vzduch, který dýcháme až po potraviny, které denně konzumujeme. Proto nemůžeme jednoznačně říci, která expozice je nejnebezpečnější, všechny jsou ale neustále kontrolovány a vyhodnocovány pomocí určených norem a zákonů. Přesto s nimi v dnešní době téměř denně přicházíme do styku (výfukové plyny, barviva, keramické výrobky, potravinářské obaly – například konzervy). U některých vzácnějších prvků se setkáváme spíše s profesní expozicí.

Olovo je hlavním zdrojem expozice znečištěné ovzduší a kontaminovaná potrava. Běžně se sním můžeme setkat v životním prostředí (průmyslové exhalace, výfukové plyny, aditiva v benzínu, barviva, keramické výrobky, potravinářské obaly – konzervy, používání olova v rozvodu pitné vody) olovo je tedy všudypřítomným kontaminantem prostředí.

Kadmium se do organizmu se dostává dvěma cestami – v potravě a dýcháním. Z potravin jsou rizikovým faktorem především vnitřnosti (játra, ledviny) nebo ryby, které byly kontaminovány kadmiem při svém růstu. Rizikové mohou být i zemědělské plodiny, pěstované na kadmiem kontaminované půdě. Vhledem k nízkému bodu varu kadmia se tento prvek poměrně snadno dostává do atmosféry. Je proto nezbytné, aby hutní provozy, které s kadmiem pracují velmi důsledně dbaly o dokonalé čištění plynných exhalací, které z hutí odcházejí. Ohroženi totiž nejsou pouze přímo pracovníci v uvedených provozech, ale i obyvatelstvo v okolí, protože kadmium nasorbované na prachové částice a atmosférický aerosol může být větrem transportováno na značně velké vzdálenosti. Patrně nejohroženější skupinu osob však tvoří kuřáci, jasně vyplývá, že kuřák je kromě běžně uváděné rakoviny plic ohrožen i rakovinou nebo chronickým selháním činnosti ledvin.

Rtuť se do organismu dostává dvěma cestami – v potravě a dýcháním. Z potravin jsou rizikovým faktorem stejně jako u kadmia především vnitřnosti (játra, ledviny) nebo ryby, které byly kontaminovány rtutí při svém růstu. Rizikové mohou být i zemědělské plodiny, pěstované na půdě zamořené rtuťnatými sloučeninami ať již z průmyslových zdrojů nebo nevhodně použitými přípravky k hubení zemědělských škůdců. Zvláště nebezpečné jsou organokovové sloučeniny rtuti, které se mohou snadno dostat do živých tkání a to například i pouhým stykem s pokožkou. Tyto sloučeniny se mohou dostávat do životního prostředí např. rozkladem různých organických sloučenin s obsahem rtuti nebo i metabolickými pochody mikroorganizmů při styku s rtutí. Nejčastějším příkladem je dimethylrtuť, (CH_3)-Hg-(CH_3), kde je jako smrtelná dávka pro dospělého člověka uváděno již 0,1 ml této kapalné substance. . Spory se vedou o dlouhodobém působení amalgámových zubních plomb, které někteří lékaři prohlašují za zcela neškodně a jiní proti nim hystericky zbrojí. Praktické zkušenosti hovoří spíše pro relativní neškodnost těchto zubních výplní, protože v uplynulých desítkách let bylo aplikováno stovky milionů těchto plomb mnoha milionům pacientů a přitom se nesetkáváme s projevy masové chronické otravy rtutí u (především) starších ročníků v populaci. [3,9,11]

2.2. Vzduch a půda

2.2.1. Způsob hodnocení kvality a obsahu TK ve vzduchu a půdě

Vzhledem ke zdravotním rizikům je nutno provádět kontrolu obsahu toxických kovů ve složkách životního prostředí, mezi které patří i atmosféra a půda. Existuje několik přístupů, jak obsah kovů v těchto matricích hodnotit a s tím souvisí také technika vzorkování a metodika stanovení obsahu toxických kovů.

Pro hodnocení obsahu kovů v atmosféře se zpravidla provádí odběr prachových částic, protože toxické kovy se v atmosféře vyskytují převážně ve formě pevných částic, tedy prachu. K odběru se využívají čerpadla, která prosávají definovaný objem vzduchu přes vhodný filtr o dané velikosti pórů. Tím je také dána velikost částic, které jsou na filtru zachyceny. Množství toxických kovů na filtru je potom stanoveno některou z technik prvkové analýzy, například rentgenovou fluorescenční spektrometrií (výhodou je, že není nutno filtr dále upravovat či rozkládat), atomovou absorpční spektrometrií či hmotnostní spektrometrií s indukčně vázaným plazmatem. Tyto dvě techniky však většinou vyžadují převedení zachyceného prachového materiálu do roztoku a je tedy nutno použít vhodný způsob rozkladu. Použitím vhodného zařízení (tzv. kaskádový impaktor) lze rozdělit částice podle velikosti do několika frakcí a poté stanovit obsah kovů v jednotlivých frakcích. Pro běžné hodnocení kvality ovzduší se stanovuje frakce PM10 (případně PM2,5). Jde o částice s velikostí pod 10 (pod 2,5) mikrometrů, které jsou pro lidský organismus nebezpečné, protože se zachycují v plicích. Větší částice se zachytí již v horních dýchacích cestách a jsou vyloučeny s hlenem z organismu.

Technika odběru prachu pomocí čerpadel umožní stanovit koncentraci prachu a obsah toxických kovů v něm pouze v krátkém časovém období během odběru. Nelze tak podchytit období mezi jednotlivými odběry. Pro dlouhodobé sledování se využívají tzv. pasivní vzorkovače, které dlouhodobě vzorkují ovzduší bez použití čerpadla. Další možností je využití biomonitoringu.

Každá půda se vyznačuje určitou pozaďovou hodnotou těžkých kovů, která vyjadřuje jejich přirozený obsah. V důsledku antropogenní činnosti dochází k pronikání těžkých kovů do půdního prostředí a zvyšování jejich hladiny nad úroveň pozadí. Jedná se zejména o produkty průmyslové činnosti, dopravy a energetiky. Za nejzávažnější zdroj kontaminace jsou považovány imise, které zasahují celou biosféru, a přitom dlouhodobě negativně ovlivňují faktory půdní úrodnosti.

Pro hodnocení obsahu toxických prvků v půdě se provádí zpravidla jejich stanovení ve výluhu z půdy. V závislosti na účelu analýzy jsou volena extrakční činidla s různou účinností loužení prvků z půdy. Jako nejslabší činidlo lze použít vodu, která je schopna extrahovat bezprostředně mobilní (pohyblivé) formy prvků v půdě. Tyto mohou být z půdy vyluhovány například deštěm a dostat se tak do podzemních vod. V souvislosti s pěstováním rostlin je důležité znát obsah prvků, které jsou rostlinám dostupné. Pro hodnocení této frakce se provádí extrakce půdy různými činidly, tzv. slabými vyluhovadly, například roztokem $CaCl_2$, NH_4NO_3, EDTA aj. Tato činidla vytěsní z půdy či rozpustí takové formy, které je rostlina schopny svým kořenovým systémem z půdy získat. Vždy jde však jen o modelování dostupnosti rostlinám. Silnějšími vyluhovacími činidly (zředěná HNO_3, lučavka královská) již extrahujeme další frakci, tzv. potenciálně mobilizovatelnou. Úplným rozkladem půdy, včetně silikátů v ní obsažených, lze potom stanovit celkový obsah prvků v půdě. Podobně by mohla být použita metoda rentgenové fluorescenční spektrometrie pro stanovení celkového obsahu prvků v půdě bez rozkladu. Ve výluzích se prvky stanoví některou z běžně používaných metod prvkové analýzy (AAS, ICP-OES, ICP-MS aj.). [12]

Obsah těžkých kovů v půdách je hodnocen limity uvedenými ve vyhlášce č.13/1994 Sb. (As, Sb, Cd, Co, Cr, Cu, Hg, Mo, Ni, Pb, V, Zn) – viz tabulka č. 1. Jde o potenciálně mobilizovatelné koncentrace prvků. I přes použití různých modelových extrakčních činidel nezískáme univerzálně platné hodnoty obsahu tzv. biodostupné frakce. Dostupnost prvků z půdy rostlinám lze hodnotit z obsahu těchto prvků v rostlinách na půdě pěstovaných. Podobně lze využít různé rostliny pro biomonitoring, kdy analýzou vybraných rostlinných druhů sledujeme znečištění půdy, respektive přímo biodostupnou frakci tohoto znečištění.

Obsah rizikových prvků v půdách (mg/kg) podle vyhlášky č.13/1994 Sb.
Výluh 2M HNO_3 (výluh roztokem 2M HNO_3 při poměru půdy k vyluhovadlu 1 : 10) [8]

tab. č. 1:

Prvky	Maximálně přípustné hodnoty (mg/kg)	
	lehké půdy	ostatní půdy

As	4,5	4,5
Be	2,0	2,0
Cd	0,4	1,0
Co	10,0	25,0
Cr	40,0	40,0
Cu	30,0	50,0
Hg	-	-
Mo	5,0	5,0
Ni	15,0	25,0
Pb	50,0	70,0
V	20,0	50,0
Zn	50,0	100,0

2.3. Biomonitoring

V biomonitoringu využíváme živých organismů pro získání informací o stupni kontaminace prostředí, v němž organismus žije, a to chemickou analýzou tohoto organismu. Organismus žije v daném prostředí a tím přichází do styku se znečištěním. Polutanty tak pronikají do organismu a může tam docházet k jejich kumulaci. Obsah zájmových polutantů (prvků) v organismu potom odráží míru znečištění těmito polutanty, či spíše riziko dostupnosti polutantů živým organismům.

Pro biomonitoring kvality ovzduší se s výhodou používají mechy, které díky absenci kořenového systému získávají vodu a živiny pouze ze srážek. Do jejich organismu se tak dostávají všechny látky, které jsou ve srážkách obsaženy, tedy i toxické kovy. Během života mechu se kovy v rostlince kumulují a mech tak může sloužit jako vzorkovač pro sledování kvality ovzduší v dlouhodobém horizontu. V České republice [5] i v Evropě [6, 7] probíhá již několik let monitoring kvality ovzduší s využitím mechů. Tyto práce jsou i dále využity. V ČR byl pro monitoring použit mech Pleurozium schreberi nebo Scleropodium purum na celkem 250 vzorkovacích místech, a to již v několika kampaních po 5 letech. Z výsledků analýzy byly sestaveny mapy znečištění ČR toxickými kovy a také bylo potvrzeno významné ubývání počtu lokalit se znečištěným ovzduším v průběhu let 1990 – 2000.

Znečištění půdy se odráží také v obsahu kovů v rostlinách na ní rostoucích. Kumulací toxických kovů jsou známé také houby. Existují však významné mezidruhové rozdíly ve schopnosti hub kumulovat různé kovy a proto je nutno se zaměřit vždy na jeden druh. V případě mechů jsou mezidruhové rozdíly ve schopnosti kumulace kovů menší. Po odběru jedinců vybraného druhu ze sledované lokality se po úpravě vzorků stanoví obsah prvků ve vzorcích některou z vhodných metod. Srovnáním výsledků lze provést například porovnání míry znečištění na jednotlivých lokalitách mezi sebou. [12]

2.4. Mechy a houby

2.4.1. Mechy

Nadříše: jaderní (*Eucaryota*)
Říše: rostliny (*Plantae*)
Podříše: vyšší rostliny (*Embryobionta*)
Oddělení: Mechy (*Bryophyta*)

Jsou to zelené výtrusové rostliny menšího vzrůstu, preferující vlhká a stinná místa, s výraznou schopností zadržovat vodu. V životním cyklu výrazně převládá haploidní gametofyt = stadium rostlin, které procházejí životním cyklem zvaným rodozměna což je střídání dvou fází – gametofyt (je haploidní, protože má jen jednu sadu chromozomů v jádře) a sporofyt (je diploidní, protože má dvě) v rámci jednoho životního cyklu. Stélka je rozlišena na kořínky, příchytná vlákna (rhizoidy), lodyžku (kauloid) a lístky (fyloidy). mechové rostlinky mají oddělená pohlaví. Pohlavní orgány, vyrůstají na vrcholu lodyžky - samčí pohlavní orgány, pelatky, produkují bičíkaté spermatozoidy, které se ve vlhkém prostředí aktivně pohybují, samičí zárodečníky obsahují velkou nepohyblivou vaječnou buňku. Oplození může proběhnout jen v dostatečně vlhkém prostředí (za deště), kdy se spermatozoidy mohou dostat k vaječným buňkám. Po oplození vznikne diploidní zygota, která se vyvíjí ve sporofyt (který je tvořen nohou, štětem a sporangiem - tobolkou, která produkuje haploidní výtrusy). Ten vyrůstá z vrcholu mateřského gametofytu a je na něm plně závislý.

Příčinou malého vzrůstu mechů je to, že nemají cévní svazky, tudíž je jejich schopnost rozvádět vodu velmi omezená. Nemají ani skutečné kořeny, stonky nebo listy - například kořeny mechů slouží pouze k přichycení a nepřivádějí rostlině z půdy živiny ani vodu, tu přijímají celým povrchem těla a rozvádějí ji vodivými pletivy nebo po snadno smáčivém povrchu rostlin. Způsob, jakým mechy hospodaří s vodou, jim umožňuje využít i velmi malých a horizontálních srážek (mnoho druhů přežívá i dlouhá, velmi suchá období. Jedním z důvodů je, že si dovedou v buňkách vytvořit velkou zásobu vody a během prvních dnů nebo týdnů velkého sucha čerpají z této rezervy. Když už vodu téměř vyčerpaly, upadnou do vegetačního klidu. Listy se stočí, aby se zbylá vlhkost nevypařovala. Celá rostlinka seschne, zhnědne a vypadá, jako by odumřela Jakmile se ale vrátí déšť, mech takřka přes noc zezelená a obživne.). Velký povrch a snadná prostupnost stélky také znamená, že mechy jsou citlivé na kvalitu ovzduší i srážek. Slouží tak jako bioindikátory znečištění.

Mechy najdeme všude, od pobřeží arktických moří přes tropy až k Antarktidě. Některé přežívají úspěšně v pouštích a na skalách, kde na ně praží slunce. Jiné žijí zase v rašeliništích a potocích. Většině mechů se však nejlépe daří na vlhkých, zastíněných místech v oblasti mírného klimatu. V lesích tvoří často husté polštářovité plochy, které dokonale překrývají hnijící pařezy a půdy.

Jednotlivé mechové rostlinky jsou přizpůsobené podmínkám stanoviště svým tvarem a způsobem růstu - to je růstová forma mechu. Existují dvě základní růstové formy a to vzpřímené (orthotropické) mechy a poléhavé, plazivé (plagiotropické) mechy. [1,10]

Mechy mají několik řádů, například:

rašeliníkotvaré – například rašeliník. Rostou na vlhkých místech, kde se z jejich odumírajících stélek v dolní části tvoří bez přístupu vzduchu ložiska rašeliny. Tento rod může ve svých buňkách pojmout velké množství vody, díky čemuž se také rašeliník prodává jako doplněk půdy.
dvouhrotcotvaré – například bělomech sivý, který tvoří kompaktní polštáře ve smrkových lesích. Mají často jednostranně uspořádané lístky.
prutníkotvaré – paprutka nicí, která je nápadná oranžově červeným štětem a nící (převislou) tobolkou, roste v lesích, na pařezech apod. Ostatní druhy jsou převážně horské (v ČR asi 14 druhů) a Vytvářejí nízké, husté polštáře štíhlých rostlinek. [4]

Pro naše zkoumání jsme si vybrali travník Schreberův (*Pleurozium schreberi*). Tento mech je u nás hojně rozšířen, obzvlášť v jehličnatých lesích a ve vřesovištích. Tvoří rozlehlé polštáře větvených rostlinek na okrajích lesů, na travnatých stráních i skalách. Lodyžky bývají až 10 cm dlouhé s prosvítavými lístky se zaokrouhlenou špičkou [1]

2.4.2. Houby

Nadříše: jaderní (*Eukaryota*)
Říše: houby (*Fungi*)
Oddělení: vřeckovýtrusné houby (*Ascomycota*)
stopkovýtrusné houby (*Basidiomycota*)

V užším pojetí jde o stélkaté rostliny různého tvaru a velikostí, bez asimilačních barviv, s heterotrofní výživou, s buněčnou stěnou chitinózní, často s tvorbou plodnic bez zoospor (= rejdivé výtrusy některých řas a hub, jejich tělo je tvořeno jednou buňkou bez buněčné stěny, opatřenou bičíky. Vznikají ve výtrusnicích – zoosporangiích a slouží k nepohlavnímu rozmnožování.). Jsou saprofyty (mikroorganismy využívající organické látky z rozkládajících se těl rostlin a živočichů) a parazity (organismus, který využívá jiné organismy -hostitele- jako zdroj své potravy i jako své stálé nebo dočasné životní prostředí a svému hostiteli tak přímo nebo nepřímo škodí).

Základní stavební jednotkou je vlákno (hyfa), které se může rozlišit v podhoubí (mycelium) a v plodnici. Rozmnožují se buď vegetativně (rozpadem vlákna mycelia), nebo nepohlavními či pohlavními výtrusy.

Systematicky se dělí na 6 tříd. Známo asi několik desítek tisíc druhů. V České republice je zjištěno asi 10.000 druhů.

Četné druhy působí škody na rostlinách, živočiších i člověku tím, že způsobují onemocnění. Mnohé druhy se využívají ve farmaceutickém a chemickém průmyslu. Jedlé druhy slouží jako potravina s malou kalorickou hodnotou nebo jako pochutina. Jsou bohaté na vitamíny a minerální látky. Jedovaté druhy nejsou početné, ale pro obsah prudkých jedů nebezpečné. [2,11]

Pro naše zkoumání jsme si vybrali muchomůrku červenou (*Amanita muscaria* var. *muscaria)*, která roste hojně v červenci až říjnu v listnatých, smíšených a především jehličnatých lesích, někdy společně s hřiby nebo v symbióze s některými stromy - břízou, modřínem, borovicí, smrkem a dubem.

Klobouk má 10 – 20 cm v průměru, v mládí je polokulovitý, pak ploše rozprostřený, ohnivě až oranžově červené, vzácně žluté barvy s bílými až žlutavými strupy, většinou uspořádaných v soustředných kruzích. Klobouk dospělých plodnic je na okraji rýhovaný, lupeny jsou bílé, vysoké a husté. Třeň je bílý s velkým měkkým prstencem, ke klobouku mírně zúžený asi 2 cm tlustý, dole rozšířený v zaobalenou hlízu. Houba roste hlavně koncem léta a na podzim v jehličnatých i listnatých lesech, ale i poblíž jednotlivých stromů mimo les, na mezích, loukách apod. Je mírně jedovatá, nejjedovatější je pokožka klobouku. Obsahuje několik jedovatých látek s účinky na nervovou soustavu a působící zažívací potíže. [2]

2.5. Metody stanovení těžkých kovů

Pro stanovení obsahu kovů ve vzorcích biologických materiálů (mechy, houby a další) lze využít různé metody tzv. prvkové analýzy. Mezi nejrozšířenější metody patří spektroskopické metody, ať již založené na molekulové spektrometrii (spektrofotometrie), nebo atomové spektrometrii. Sem patří například rozšířená atomová absorpční spektrometrie – AAS, atomová fluorescenční spektrometrie – AFS, optická emisní spektrometrie – OES s různými typy excitace, například plamenem, indukčně vázaným plazmatem (ICP-OES), doutnavým výbojem (GD-OES) či další. V poslední době se rozšiřuje moderní metoda hmotnostní spektrometrie s indukčně vázaným plazmatem (ICP-MS), o které je více pojednáno níže.

Z dalších metody pro stanovení obsahu kovů lze jmenovat metody elektrochemické, především polarografie a její různé modifikace, případně separační metody (chromatografie, elektroforéza). U většiny metod je nutno převést vzorek do formy roztoku, takže se využívá některý ze způsobů mineralizace, ať již na suché cestě (spalování v muflové peci) či na mokré cestě (rozklad kyselinou či jejich směsmi). [12]

2.6. ICP – MS

ICP-MS neboli hmotnostní spektrometrie s indukčně vázaným plazmatem je ultrastopová analytická metoda sloužící ke stanovení obsahu stopových množství jednotlivých prvků v analyzovaném vzorku. Tato technika umožňuje analyzovat téměř všechny prvky od lithia po uran s měřícím rozsahem od jednotek ng/l po stovky mg/l. Její masové nasazení nastalo až v posledních přibližně 20 letech, kdy byly úspěšně vyřešeny technické problémy s vytvářením stabilního vakua o tlaku řádově 10^{-3} Pa a udržením konstantních podmínek v argonovém plazmatu

2.6.1. Princip metody

Roztok analytického vzorku je zmlžen a vzniklá mlha je proudem argonu vedena do plazmové hlavice, ve které je za pomoci střídavého vysokofrekvenčního magnetického pole udržováno argonové plazma (= ionizovaný plyn) o teplotě 6 000 – 10 000 K. Za takových podmínek se rozpouštědlo okamžitě odpaří a zanikají chemické vazby v molekulách přítomných sloučenin. Jednotlivé volné atomy ve většině případů

vytvoří jednou kladně nabité ionty Me^+, které jsou dále unášeny do přechodové komory, kde je snížen tlak plynu.

Po průchodu do vstupu k detektoru klesá tlak na řádově 10^{-3} Pa a ionty se systémem elektromagnetických čoček dostávají do kvadrupólového analyzátoru. Zde jsou analyzované ionty vedeny takovým způsobem, aby na detektor dopadly v daném časovém okamžiku pouze ionty se zvolenou hmotností. Dopadem na detektor vzniká velmi slabý elektrický proud, který je následně zesílen a je změřena jeho intenzita. Pomocí výpočetního programu jsou naměřené intenzity signálu převedeny na koncentrační data a výsledkem analýzy jsou údaje o koncentraci měřených prvků v analyzovaném roztoku.

V maximu běžných aplikací se analýza technikou ICP-MS provádí z roztoku vzorku. Proto je nutno pevné vzorky nejprve převést do roztoku vhodným způsobem rozkladu. U biologických vzorků hovoříme o tzv. mineralizaci, kterou provádíme například rozpouštěním vzorku v koncentrované kyselině dusičné za horka. Pro převedení roztoku vzorku do plazmatu je nejprve třeba jej dopravit do zmlžovače, což se provádí za použití peristaltické pumpy. V prvním kanále transportuje roztok vzorku, v druhém kanále roztok vnitřního standardu a třetí může být použit například pro in-line ředění příliš koncentrovaných vzorků.

Vytváření aerosolu vzorku probíhá ve zmlžovači, z něhož vychází jemně dispergovaná „mlha“ kapiček roztoku a je proudem argonu unášena do plazmatu. V těchto zmlžovačích se slabý proud roztoku vstřikuje tenkou tryskou do proudu argonu a v systému několika koncentrických trubic vzniká poměrně velmi dobře definovaný a homogenní aerosol vzorku. Jeho hlavní nevýhodou je možnost zanesení trysky malými nerozpuštěnými částečkami ve vzorku nebo vykrystalovanými solemi při analýze příliš koncentrovaných roztoků. Protože analýzu obsahu jednotlivých iontů ve vzorku lze provést pouze za podmínek poměrně vysokého vakua, je třeba ionty analyzovaných prvků transportovat do tohoto prostředí. Vakuum je získáváno kombinací činnosti několika vakuových pump.

2.6.2. Využití metody

Technika ICP-MS je prakticky využívána všude tam, kde je požadována velmi vysoká citlivost analýz a současně přináší praktické výhody v možnosti velmi rychle a relativně přesně analyzovat velké množství prvků v jednom vzorku. Hlavním omezením jsou zde roztoky o vysoké koncentraci solí (mořská voda, krev…), které způsobují blokování vstupních kónů a mění energetické podmínky v plazmatu a dále tvorba polyatomických částic, která může působit falešné pozitivní signály u analyzovaných prvků.

Problém salinity vzorků lze díky extrémní citlivosti techniky obvykle odstranit vhodným naředěním. V případě polyatomických interferencí je jejich odhalení a eliminace závislá především na zkušenosti analytika, který by měl před analýzou vzorků zcela nového typu provést důkladný průzkum majoritních složek a posoudit, zda některé z nich nebudou příčinou vzniku nečekaných kombinovaných iontů. Moderní přístroje využívají pro eliminaci polyatomických iontů tzv. kolizně-reakční celu. [11, 12]

3. Experimentální část

3.1. Vzorky a jejich odběr

Ke zkoumání jsme použili vzorky z osmi lokalit z Moravy a to z okolí Bobravy, Brno - Štýřice, Ivančic-Rény, Jeseníků-Ramzové, brněnské přehrady, Prostějova, Silůvek a Zbraslavi. Vzorky byly nasbírány koncem září a začátkem října.

Bobrava
vzorky byly odebrány za polojasného počasí v okolí obce Prštice. Mech byl odebrán pomocí nože, v blízkosti rybníka z kmene stromu, houba pak na kraji nedalekého smíšeného lesa.

Brno - Štýřice
vzorky byly odebrány za slunečného počasí v okolí sjezdu z dálnice D1 u ulice Vídeňské zhruba 50 metrů od dálnice. Mech i houba byly odebrány pod stromy pomocí nože.

Ivančice

vzorky byly odebrány za slunečného počasí v parku zvaném Réna, který je osázen listnatými a jehličnatými stromy. Mech byl odebrán pomocí nože z kořenů smrku, muchomůrka poblíž mechu. Místo odběru leží nedaleko silnice druhé třídy a železniční trati.

Jeseníky
vzorky byly odebrány za deštivého počasí v okolí Ramzové, v lese daleko od pole a železniční trati. Mech s houbou byly odebrány bez pomocí nože, pod jehličnatými stromy poblíž potoka.

brněnská přehrada
vzorky byly odebrány za deštivého počasí poblíž hradu Veveří, kousek od méně frekventované silnice, daleko od pole a lesní cesty. Mech byl seškrábnut ze skály, houba odebrána v blízkém lese.

Prostějov
vzorky byly odebrány za slunečného počasí u obce Mostkovice, asi 1,5 km od vodní nádrže Plumlov, daleko od železniční tratě. Mech byl společně s houbou odebrán pomocí nože na kraji listnatého lesa poblíž pole.

Silůvky
vzorky byly odebrány za deštivého počasí poblíž železniční tratě, kudy projíždí denně jak osobní vlaky v intervalech po půl hodině, tak i vlaky nákladní. Mech byl odebrán ze spadlého stromu a houba v nedalekém lese poblíž lesní cesty.

Zbraslav
vzorky byly odebrány v lese nedaleko od silnice daleko od železniční tratě na místě bývalé skládky. Mech byl odebrán pomocí nože z kamene na louce poblíž jehličnatého lesa, ze kterého byla odebrána houba.

tab. č. 2 - přibližné vzdálenosti místa odběru od cest, polí apod.

	vzdálenost od (m)					místo odebrání			
						houba		mech	
	obytné zóny	silnice	lesní cesta	pole	tratě	pod stromem	na louce apod.	pod stromem	na louce apod.
Bobrava	2500	500	20	700	200	ano	ne	ne	ano
Brno	100	50	50	200	-	ano	ne	ano	ne
Ivančice	400	150	20	20	50	ano	ne	ano	ne
Jeseníky	1500	800	200	-	-	ano	ne	ano	ne
přehrada	2500	40	-	400	-	ano	ne	ne	ano
Prostějov	1000	-	20	10	-	ano	ne	ano	ne

Silůvky	100	50	5	5	10	ano	ne	ne	ano
Zbraslav	700	150	10	600	-	ano	ne	ne	ano

3.2. Příprava vzorků

Vzorky mechů jsme po odběru bez úprav usušili na vzduchu při laboratorní teplotě na papíře. Ze suchého mechu jsme keramickými nůžkami odstřihli zhruba horní dva centimetry. Takto upravené vzorky byly uchovávány až do vlastní analýzy v PE sáčku při laboratorní teplotě.

Vzorky hub jsme nakrájeli nerezovým nožem a usušili při laboratorní teplotě. Suché houby jsme pomleli v oscilačním mlýnu MM301 (Retsch, Německo) s mlecími nádobami z karbidu wolframu. Mleté vzorky byly uchovávány až do vlastní analýzy v PE sáčku při laboratorní teplotě.

Oba typy vzorků byly rozkládány v mikrovlnném rozkladném zařízení MWS3+ (Berghof, Německo). 0,25 g vzorku bylo v rozkladné nádobě smícháno se 4 ml koncentrované kyseliny dusičné HNO_3 (p.a., Merck, přečištěná podvarovou destilací v PTFE aparatuře) a 2 ml peroxidu vodíku 30 % H_2O_2 (Analpure, Analytika Praha). Po uzavření víčkem proběhl rozklad při 190 ºC po dobu 10 minut. Po rozkladu byla přebytečná kyselina odpařena na objem mineralizátu 1-2 ml. Mineralizát byl převeden do 25 ml odměrné baňky a doplněn demineralizovanou vodou. Před měřením byl roztok vzorku odstředěn.

3.3. Stanovení obsahu TK ve vzorcích

Stanovení obsahu kovů v připravených vzorcích mechů a hub bylo provedeno metodou hmotnostní spektrometrie s indukčně vázaným plazmatem na přístroji Agilent 7500ce. Pro eliminaci rušících polyatomických iontů byla použita kolizně-realční cela v heliovém módu. V tabulce č. 3 jsou uvedeny analyty, jejich měřené izotopy a použité vnitřní standardy.

Směsný roztok vnitřních standardů byl nasáván peristaltickou pumpu a on-line míchán s roztokem vzorku, nasávaným druhým kanálem stejné pumpy. Směs byla zmlžována proudem argonu ve skleněném koncentrickém zmlžovači MicroMist a aerosol byl po oddělení velkých kapek ve skleněné Scottově mlžné komoře veden injektorem plazmové hlavice do plazmatu. Přes niklové konusy byly vznikající ionty nasávány do vakuové části spektrometru pomocí extrakčních čoček a dále vedeny do kolizně-reakční cely, naplněné heliem. Zde dochází k eliminaci vznikajících polyatomických iontů. Dále jsou ionty rozděleny podle poměru hmotnost/náboj v kvadrupólovém analyzátoru a detekovány na elektronásobičovém detektoru.

Pro kvantitativní vyhodnocení byla použita metoda kalibrační závislosti s vnitřním standardem. Rozsah koncentrací kalibračních roztoků byl volen s ohledem na předpokládané koncentrace analytů ve vzorcích. Kalibrační roztoky byly připraveny ředěním z certifikovaných kalibrační roztoků Astasol (Analytika Praha) a koncentraci $(1{,}000 \pm 0{,}002)\ g.l^{-1}$.

tab. č. 3 - Stanovované analyty, jejich měřené izotopy a použité vnitřní standardy (IS)

Analyt	Al	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	As	Mo	Cd	Sn	Sb	Ba	Pb
Izotop	27	51	52	55	57	59	60	63	66	75	95	111	118	121	137	208
IS	45Sc	72Ge										115In				209Bi

4. Výsledky a diskuze

4.1. Vzorky a jejich zpracování

Lokality byly vybrány podle blízkosti bydliště nás, nebo našich kamarádů, kteří nám byli se sběrem ochotni pomoci. Travník Schreberův a muchomůrku červenou jsme si vybrali proto, že na rozdíl od jiných druhů mechů a hub byly nalezeny ve většině námi vybraných lokalit.

U usušených mechů jsme odstřihli horní 3 cm, protože jsme k analýze potřebovali zelenou, živou část mechu, bez nečistot a odumřelých částí. Mletí hub jsme zvolili, abychom získali homogenní, dobře promísenou směs.

Rozklad vzorků proběhl v mikrovlnném rozkladném zařízení. Jedná se o moderní a rychlý způsob s omezením možností kontaminace a to vzhledem k uzavřenému rozkladnému systému. Dále bylo také potřeba odpařit nadbytečnou kyselinu po rozkladu, neboť přístroj ICP-MS je citlivý na příliš vysoké koncentrace kyselin.

Metoda ICP-MS byla zvolena pro stanovení vybrané řady prvků a je běžně používána při stanovování obsahu těžkých kovů v biologických materiálech na pracovišti RECETOXu.

4.2. Mechy

4.2.1. Výsledky stanovení obsahu TK v biomase

V následující tabulce je uvedena mez detekce, v další je pak uveden obsah stanovených těžkých kovů v mg/kg

tab. č. 4 - meze detekce

Analyt	mg/kg	Analyt	mg/kg
Al	0,2	Zn	0,5
V	0,005	As	0,005
Cr	0,001	Mo	0,002
Mn	0,002	Cd	0,001
Fe	0,1	Sn	0,005
Co	0,001	Sb	0,001
Ni	0,02	Ba	0,01
Cu	0,05	Pb	0,005

tab. č. 5 - obsahy těžkých kovů v mg/kg

	Bobrava	Brno	Ivančice	Jeseníky	přehrada	Prostějov	Silůvky	Zbraslav	ø ČR
Al	932	2180	887	735	982	1210	2030	2572	565
V	1,67	3,48	1,64	1,43	1,70	2,10	3,83	4,10	1,66
Cr	1,33	2,84	1,33	1,22	1,45	1,63	2,59	3,19	2,11
Mn	762	295	595	728	721	359	197	295	520
Fe	557	1460	536	456	563	775	1450	1600	467
Co	0,39	0,70	0,37	0,35	0,44	0,47	0,59	0,88	0,35
Ni	1,64	2,00	1,65	1,34	1,82	1,59	2,04	2,69	2,09
Cu	6,00	5,35	6,35	6,11	8,74	5,81	6,02	5,79	6,62
Zn	35,1	28,2	30,0	39,4	41,3	27,0	30,3	30,9	39,0
As	0,34	0,72	0,36	0,32	0,39	0,42	0,66	0,76	0,32
Mo	0,18	0,14	0,18	0,07	0,16	0,15	0,23	0,17	0,16
Cd	0,24	0,18	0,23	0,19	0,27	0,20	0,17	0,21	0,29
Sn	0,25	0,32	0,26	0,23	0,20	0,22	0,33	0,30	0,207
Sb	0,12	0,07	0,11	0,08	0,09	0,08	0,09	0,05	0,12
Ba	26,3	66,8	33,1	14,6	17,4	32,8	47,2	61,4	21,4
Pb	5,33	5,26	5,30	3,79	3,99	4,05	6,48	5,97	6,72

Všechny naměřené hodnoty obsahů těžkých kovů ve vzorcích jsou nad mezí detekce.

U prvků Al, Fe, Mn byly naměřeny vysoké obsahy, řádově stovky až tisíce mg/kg. V řádu desítek se vyskytují Zn a Ba, v řádu jednotek V, Cr, Ni, Cu, Pb, a v desetinách mg/kg se vyskytují Co, As, Mo, Cd, Sn, Sb.

4.2.2. Srovnání lokalit a jejich oznámkování

Abychom mohli rozhodnout, zda je lokalita znečištěná či nikoli, je třeba srovnat s nějakou hodnotou. Tou by mohl být průměr hodnot všech lokalit. Vzhledem k tomu, že jich ale máme málo, vezmeme raději průměr z ČR z publikace [5]. Tento průměr je uveden v tabulce č. 5. Každá z hodnot obsahu těžkých kovů v mechu z lokality je podělena průměrem a tyto hodnoty jsou uvedeny v tabulce č. 6. Je-li hodnota u prvku menší než 1 znamená to, že je lokalita tímto prvkem znečištěna méně, hodnota větší než 1 znamená, že je lokalita prvkem znečištěna více.

Abychom pak následně mohli říct zda je lokalita čistá, musíme srovnat všechny prvky. Proto oznámkujeme jednotlivé hodnoty opět dle práce [5]. Kritéria pro přidělení známky jsou uvedena v tabulce č. 7. Hodnoty známek, které pro dané prvky plynou jsou uvedeny v tabulce č. 8, průměrné hodnoty známek za lokalitu a dále také za prvek na daných lokalitách jsou uvedeny na obrázcích č. 1 a č. 2.

tab. č. 6 - hodnoty podělené ø ČR

	Bobrava	Brno	Ivančice	Jeseníky	přehrada	Prostějov	Silůvky	Zbraslav	ø
Al	1,65	3,85	1,57	1,3	1,74	2,14	3,59	4,55	2,55
V	1,01	2,1	0,99	0,86	1,02	1,26	2,31	2,47	1,661
Cr	0,63	1,35	0,63	0,58	0,69	0,77	1,23	1,51	0,92
Mn	1,47	0,57	1,15	1,4	1,39	0,69	0,38	0,57	0,95
Fe	1,19	3,13	1,15	0,98	1,21	1,66	3,11	3,41	1,98
Co	1,13	2,01	1,06	1	1,28	1,35	1,68	2,52	1,50
Ni	0,78	0,96	0,79	0,64	0,87	0,76	0,98	1,29	0,88
Cu	0,91	0,81	0,96	0,92	1,32	0,88	0,91	0,88	0,95
Zn	0,9	0,72	0,77	1,01	1,06	0,69	0,78	0,79	0,84
As	1,09	2,28	1,13	1,02	1,12	1,31	2,08	2,39	1,55
Mo	1,14	0,86	1,09	0,41	0,97	0,94	1,42	1,03	0,98
Cd	0,84	0,62	0,79	0,68	0,94	0,69	0,61	0,73	0,74
Sn	1,23	1,53	1,23	1,11	0,95	1,08	1,6	1,45	1,27
Sb	0,99	0,62	0,95	0,68	0,74	0,66	0,8	0,4	0,73
Ba	1,7	3,12	1,55	0,68	0,81	1,53	2,21	2,87	1,81
Pb	0,79	0,78	0,79	0,56	0,59	0,6	0,96	0,89	0,75
ø	1,09	1,58	1,04	0,86	1,04	1,06	1,54	1,73	

Z tabulky plyne, že Al, V, Fe, Co, As, Sn a Ba jsou lokality nadprůměrně znečištěné, zatímco Cr, Mn, Ni, Cu, Zn, Mo,Cd, Sb a Pb znečištěné podprůměrně

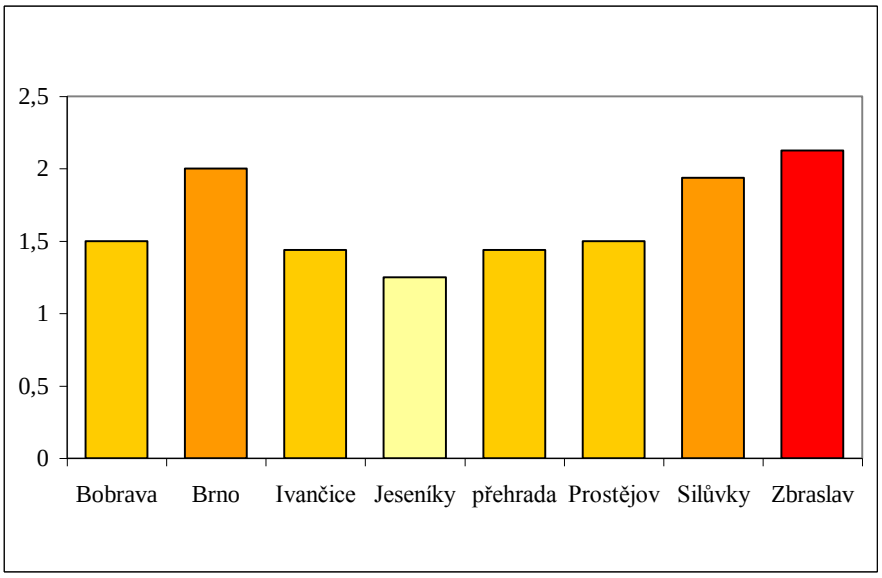
tab. č. 7 – Kritéria pro přidělení známky (mg/kg)

	nízké	střední	vysoké	velmi vysoké
známka	1	2	3	4
Al	<600	600-1200	1200-1600	>1600

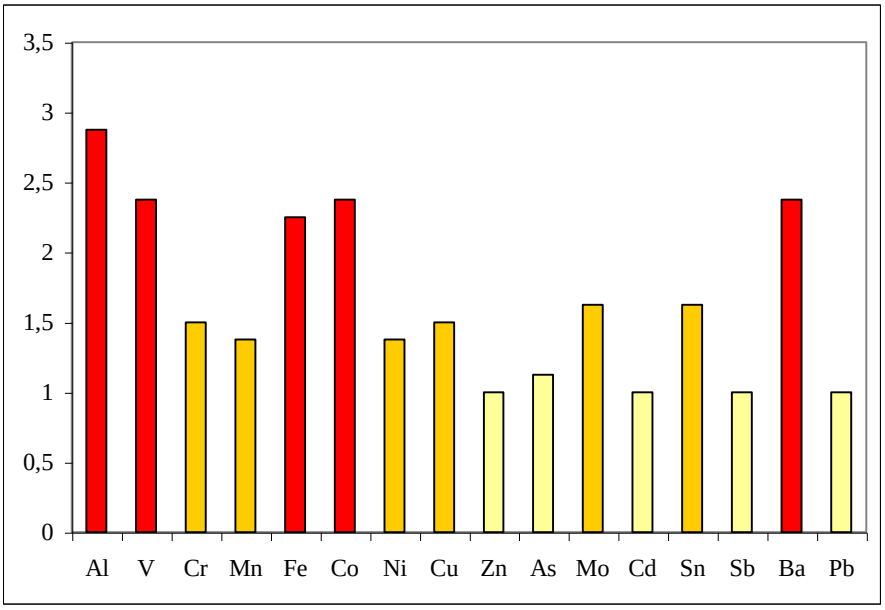
As	<0,4	0,4-0,8	0,8-1,4	>1,4
Ba	<25	25-40	40-50	>50
Cd	<0,3	0,3-0,6	0,6-0,8	>0,8
Co	<0,3	0,-0,6	0,6-0,8	>0,8
Cr	<1,5	1,5-4,0	4,0-6,0	>6,0
Cu	<6	6,-9	9,-12	>12
Fe	<500	500-1250	1250-1750	>1750
Mn	<700	700-1150	1150-1450	>1450
Mo	<0,15	0,15-0,30	0,30-0,40	>0,40
Ni	<2	2,-5	5,-7	>7
Pb	<10	10,-25	25-35	>35
Sb	<0,125	0,125-0,200	0,200-0,250	>0,250
Sn	<0,25	0,25-0,40	0,40-0,50	>0,50
V	<1,5	1,5-3,0	3,0-4,0	>4,0
Zn	<50	50-80	80-100	>100

tab. č. 8 – oznámkování lokalit

	Bobrava	Brno	Ivančice	Jeseníky	přehrada	Prostějov	Silůvky	Zbraslav	ø
Al	2	4	2	2	2	3	4	4	2,88
V	2	3	2	1	2	2	3	4	2,38
Cr	1	2	1	1	1	2	2	2	1,5
Mn	2	1	1	2	2	1	1	1	1,38
Fe	2	3	2	1	2	2	3	3	2,25
Co	2	3	2	2	2	2	2	4	2,38
Ni	1	2	1	1	1	1	2	2	1,38
Cu	1	1	2	2	2	1	2	1	1,5
Zn	1	1	1	1	1	1	1	1	1
As	1	2	1	1	1	1	1	1	1,13
Mo	2	1	1	1	2	2	2	2	1,63
Cd	1	1	1	1	1	1	1	1	1
Sn	2	2	2	1	1	1	2	2	1,63
Sb	1	1	1	1	1	1	1	1	1
Ba	2	4	2	1	1	2	3	4	2,36
Pb	1	1	1	1	1	1	1	1	1
ø	1,5	2	1,44	1,25	1,44	1,5	1,94	2,13	



obr. č. 1 – Graf průměrných známek pro lokality (průměrná známka je vynesena na ose y)



obr. č. 2 – průměrné známky pro prvky (průměrná známka je opět vynesena na ose y)

Podle hodnoty průměrné známky lze rozhodnout, zda je lokalita nízce, středně, vysoce a nebo velmi vysoce znečištěná. K tomuto rozhodnutí slouží meze uvedené v tabulce č. 9 spolu s procentickým údajem o podílu území ČR, spadajícím do této

třídy. Tabulka je převzata z práce [5]. Z obrázku č. 1 plyne, že nejčistší lokalitou se známkou 1,25 jsou Jeseníky. Patří tedy do oblasti nízce znečištěné, stejně jako 66,5 % ČR. Středně znečištěné jsou Bobrava, Ivančice, brněnská přehrada a Prostějov s průměrnou známkou v rozmezí 1,44 – 1,5, tak jako 29,5 % ČR , více znečištěné jsou Brno a Silůvky se známkami 1,94 a 2, podobně jako 3 % území ČR. Velmi vysoce znečištěná je oblast Zbraslav se známkou 2,13. Do stejné třídy znečištění spadá 1 % ČR.

Z obrázku č. 2 plyne, že hlavní prvky, kterými jsou naše lokality znečištěny jsou Al, V, Fe, Co, Ba. Naopak prvky, kterých jsme ve vzorcích naměřili nejméně, jsou Zn, As, Cd, Sb, Pb.

tab. č.9 – kritéria pro rozdělení lokalit podle čistoty

známka	1	2	3	4
kritéria	1,00-1,35	1,35-1,70	1,70-2,05	2,05-2,71
% ČR	66,5	29,5	3	1

4.2.3. Korelace prvků a rozdělení prvků do skupin podle znečištění

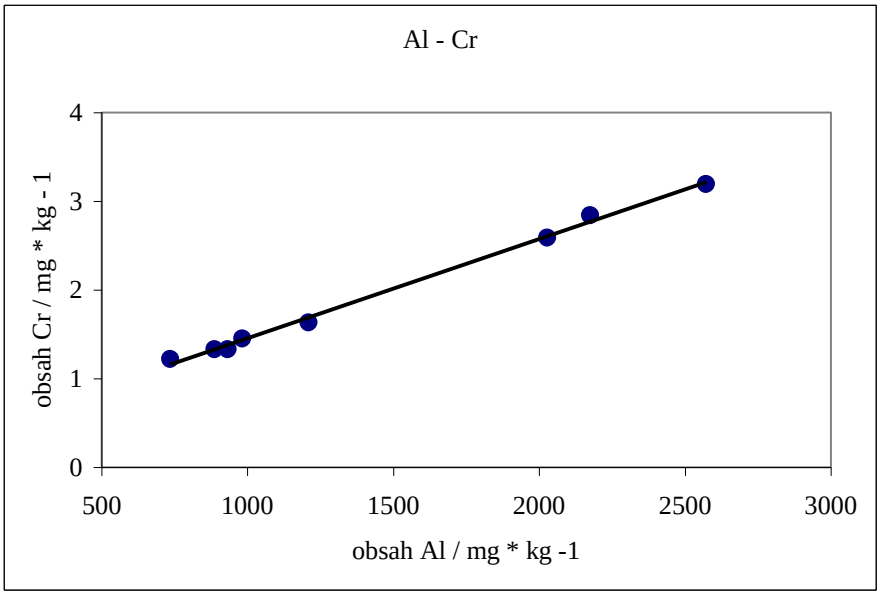
Nápadně podobné znečištění některými prvky ukazuje na možnou korelaci. Proto byly vypočteny korelační koeficienty dle Pearsona. Výsledky jsou uvedeny v tabulce č. 10.

tab. č. 10 - korelační koeficienty

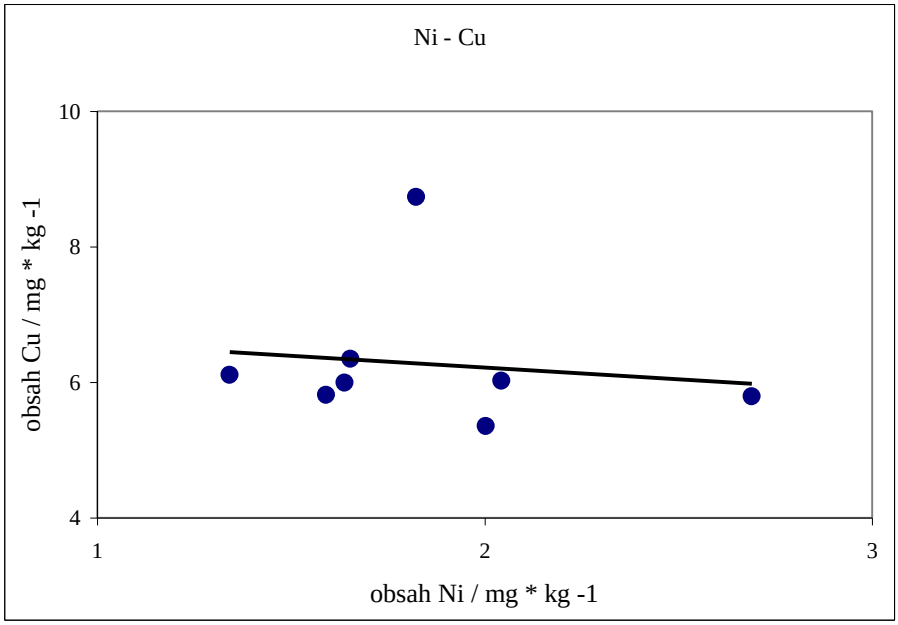
	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	As	Mo	Cd	Sn	Sb	Ba	Pb
Al	0,99	1,00	-0,85	0,99	0,97	0,90	-0,42	-0,53	0,99	0,35	-0,52	0,81	-0,67	0,93	0,69
V		0,98	-0,89	1,00	0,92	0,87	-0,43	-0,53	0,98	0,43	-0,58	0,86	-0,60	0,90	0,75
Cr			-0,85	0,99	0,97	0,89	-0,42	-0,51	1,00	0,31	-0,54	0,82	-0,68	0,93	0,67
Mn				-0,89	-0,75	-0,64	0,52	0,78	-0,85	-0,42	0,73	-0,76	0,52	-0,84	-0,60
Fe					0,93	0,85	-0,46	-0,56	0,99	0,38	-0,60	0,86	-0,62	0,93	0,71
Co						0,94	-0,35	-0,44	0,95	0,24	-0,38	0,68	-0,77	0,89	0,58
Ni							-0,14	-0,32	0,87	0,43	-0,14	0,62	-0,63	0,78	0,67
Cu								0,72	-0,41	0,01	0,78	-0,64	0,19	-0,58	-0,42
Zn									-0,52	0,33	0,61	-0,59	0,16	-0,72	-0,49
As										0,33	-0,55	0,83	-0,65	0,94	0,68
Mo											0,07	0,46	0,32	0,34	0,75
Cd												-0,69	0,36	-0,57	-0,33
Sn													-0,22	0,85	0,88
Sb														-0,52	0,01

Ba	0,70

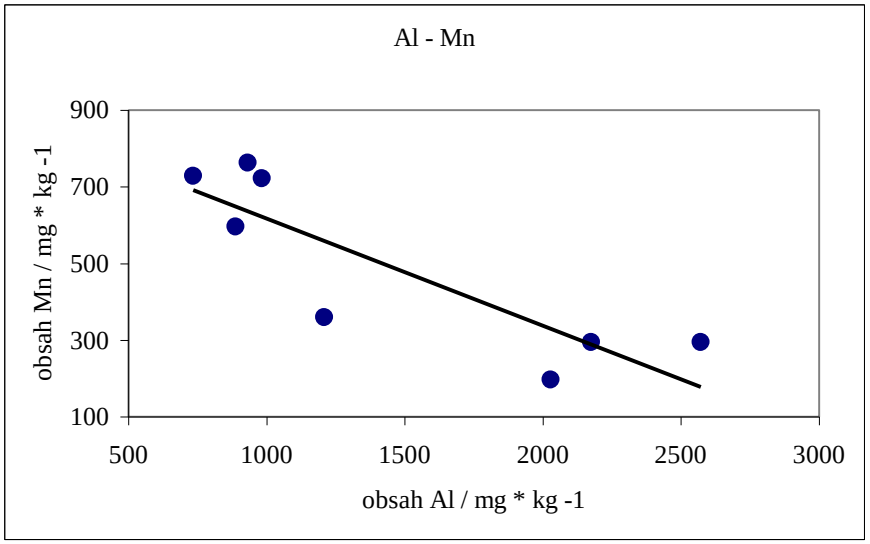
Pokud je koeficient blízko k 1, pak prvky korelují (= jsou na sobě závislé). Je-li kolem 0, pak jsou na sobě nezávislé. Blíží-li se hodnota k -1 je korelace opačná (= čím je víc jednoho prvku, tím je míň druhého). Z tabulky vidět, že jsou prvky, které spolu korelují jako například Al, V, Fe, Co, Ni (vyznačeny hodnoty s koeficientem > 0,85). Naopak jsou také prvky, které nekorelují, například Cu, Zn, Mo, Cd. Zvláštní případ je pak Mn. Následuje ukázka prvků, které spolu korelují, nekorelují a záporně korelují.



obr. č. 3 – graf korelace +1 (Al – Cr)



obr. č. 4 – graf korelace – 0,14 (Ni – Cu)

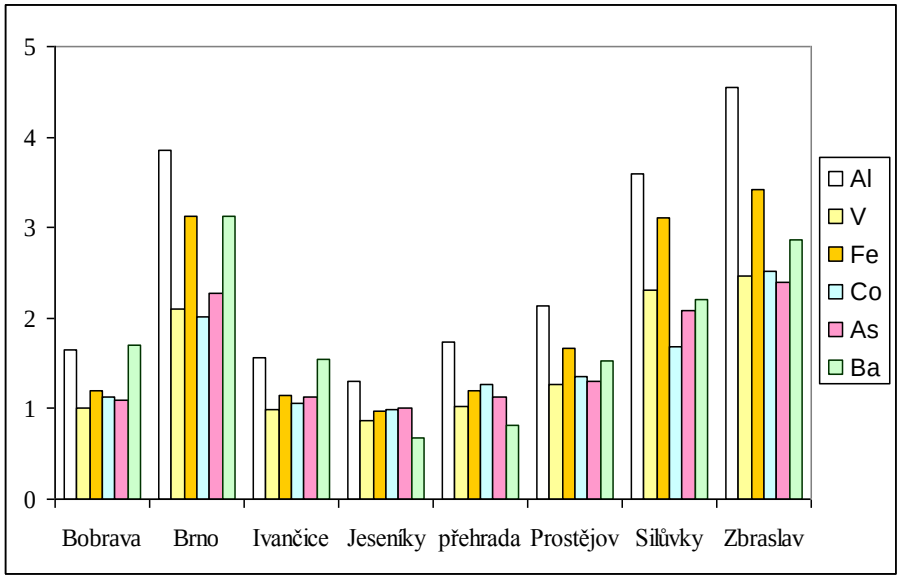


obr. číslo 5 – graf korelace -0,85 (Al – Mn)

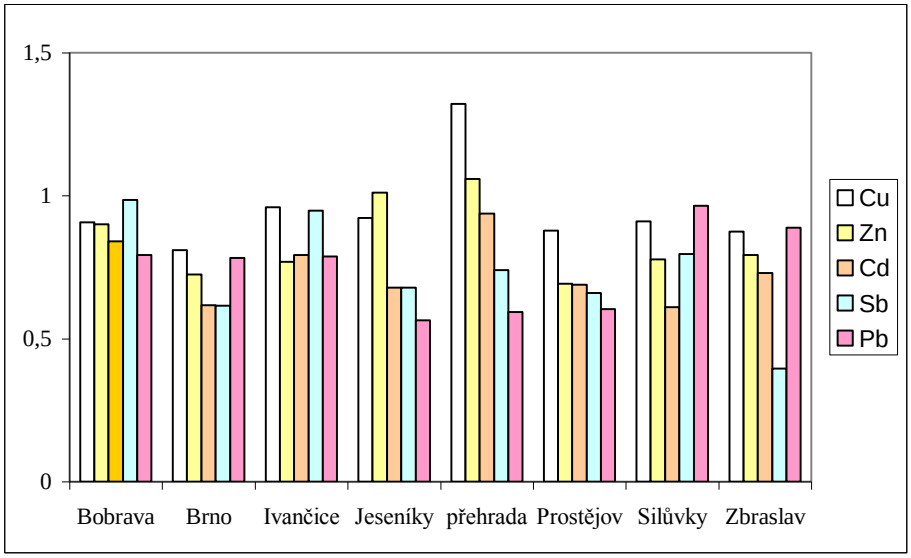
Z korelace se dá usoudit, že prvky, které spolu korelují jsou pravděpodobně ze stejného zdroje znečištění. Ze studia literatury [5] vyplývá, že jsou to zřejmě prvky z půdy, tudíž lokality jsou znečištěné půdním prachem (viz obr. č. 6). Naopak prvky co

nekorelují, patří do skupiny prvků z průmyslu barevných kovů. Těmito prvky lokality znečištěné nejsou (viz obr. č. 7). Zvláštní případ je Mn, kde je záporná korelace. O Mn je totiž známo, že jej uvolňují stromy z listů, a ten pak následně ze stromu padá na vegetaci pod ním. Pokus – analytický výsledek je tedy silně ovlivněn místem vzorkování a nelze jednoznačně posoudit kvalitu ovzduší.

Rozdělení prvků do dvou skupin na korelující a nekorelující (viz obr. č. 6 a č. 7) také ukazuje, že korelujícími prvky jsou lokality většinou znečištěné (průměrná známka nad 1), naopak nekorelujícími znečištěné nejsou (průměrná známka pod 1).



obr. č. 6 – graf vybraných korelujících prvků v lokalitách (průměrná známka je vynesena na ose y)



obr.k č. 7 – graf vybraných nekorelujících prvků na lokalitách (průměrná známka je vynesena na ose y)

4.3. Houby

4.3.1. Výsledky stanovení TK v biomase

V následující tabulce je uveden obsah stanovených těžkých kovů v mg/kg. Obsah kobaltu v houbách není uveden, protože při mletí vzorku dochází ke kontaminaci vzorku kobaltem z materiálu mlecích nádob.

Meze detekce jsou stejné jako u mechů (viz tab. č. 4).

tab. č. 11 – obsahy těžkých kovů v mg/kg

	Bobrava	Brno	Ivančice	Jeseníky	přehrada	Prostějov	Silůvky	Zbraslav	ø
Al	422	408	541	707	40,2	294	326	913	456
V	112	86,7	132	129	60,5	132	145	96,5	111
Cr	0,501	0,652	0,778	0,783	0,265	0,504	0,555	0,732	0,59
Mn	24,1	20,6	14,7	30,3	8,42	38,6	55	19,5	26,4
Fe	184	172	239,7	174,9	22,5	132	148	332	175,8
Ni	5,83	1,27	1,71	6,6	2,05	1,87	1,74	1,55	2,83
Cu	29,2	13	14,2	26,4	15,4	21,6	22,4	22,2	20,6
Zn	160	89,4	115	140	112	165	153	173	138
As	3,54	0,86	1,97	2,62	2,95	2,53	2,23	2,02	2,34
Mo	0,01	0,01	0,03	0,01	0,01	0,01	0,02	0,02	0,02
Cd	11,9	8,03	10,9	11,1	7,4	13,2	13,9	16	11,5
Sn	0,04	0,02	0,05	0,06	0,01	0,04	0,03	0,05	0,04

Sb	0,02	0,01	0,02	0,03	0,07	0,04	0,03	0,02	0,03
Ba	3	3,32	3,25	3,73	0,39	2,79	3,22	7,78	3,44
Pb	0,68	0,34	0,96	0,96	0,1	0,86	0,69	1,12	0,71
ø	63,79	53,61	71,75	82,23	18,15	53,67	58,12	105,70	

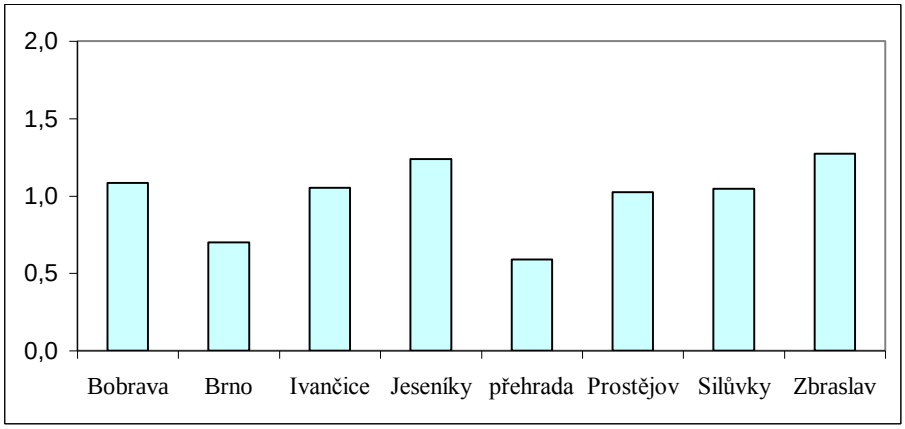
Z tabulky můžeme vyčíst, že nejméně znečištěné vzorky hub jsou z námi vybraných lokalit vzorky z brněnské přehrady, naopak nejvíce znečištěné jsou houby ze Zbraslavi. Překvapilo nás, že Jeseníky, které při stanovení mechů vyšly jako nejméně znečištěné, jsou u hub druhou nejvíce znečištěnou lokalitou. Ale vzhledem k tomu, že nemáme k dispozici podobné hodnoty pro srovnání jako u mechů, nemůžeme říct, jsou-li vzorky hub více či méně znečištěné, než jinde v ČR.

Ze stejného důvodu nemáme ani průměrnou hodnotu pro ČR pro obsah těžkých kovů v houbách a tak jsme podělili naměřené hodnoty průměrem pro dané prvky z našich osmi lokalit. Tyto hodnoty jsou uvedeny v tabulce č. 12.

Tab. č. 12 – hodnoty podělené ø lokalit

	Bobrava	Brno	Ivančice	Jeseníky	přehrada	Prostějov	Silůvky	Zbraslav
Al	0,92	0,89	1,18	1,54	0,09	0,64	0,71	2,05
V	1,00	0,78	1,18	1,15	0,54	1,19	1,29	0,86
Cr	0,85	1,10	1,30	1,32	0,43	0,84	0,93	1,23
Mn	0,91	0,78	0,56	1,15	0,32	1,46	2,08	0,74
Fe	1,05	0,98	1,36	1,00	0,13	0,75	0,84	1,89
Ni	2,06	0,45	0,60	2,33	0,73	0,66	0,62	0,55
Cu	1,42	0,63	0,69	1,28	0,75	1,05	1,09	1,08
Zn	1,16	0,65	0,83	1,02	0,81	1,19	1,10	1,25
As	1,51	0,37	0,84	1,12	1,26	1,08	0,95	0,86
Mo	0,72	0,82	1,87	0,84	0,51	0,84	1,07	1,33
Cd	1,03	0,70	0,94	0,96	0,64	1,14	1,20	1,39
Sn	0,97	0,59	1,36	1,56	0,20	1,09	0,85	1,38
Sb	0,82	0,30	0,76	0,88	2,19	1,40	1,02	0,63
Ba	0,87	0,97	0,95	1,09	0,11	0,81	0,94	2,26
Pb	0,96	0,47	1,35	1,34	0,14	1,21	0,97	1,57
ø	1,08	0,70	1,05	1,24	0,59	1,02	1,04	1,27

Z průměrných hodnot na lokalitě lze tedy rozhodnout, která lokalita (respektive houba z dané lokality) je znečištěna nad- či podprůměrně. Tyto průměrné hodnoty pro lokality jsou uvedeny v obrázku č 8. Odtud je zřejmé již výše zmíněné konstatování, že nejnižší obsah kovů je v houbách z lokality Brněnská přehrada a naopak nejvyšší obsah je v houbě z lokality Zbraslav.



obr. 8: graf průměrných hodnot obsahu těžkých kovů pro lokality

4.3.2. Korelace prvků

Jako u mechů, tak i hub byly vypočteny korelační koeficienty dle Pearsona. Výsledky jsou uvedeny v tabulce č. 13

tab. č. 13 – korelační koeficienty

	V	Cr	Mn	Fe	Ni	Cu	Zn	As	Mo	Cd	Sn	Sb	Ba	Pb
Al	0,25	0,85	-0,04	0,92	0,20	0,31	0,39	-0,24	0,53	0,60	0,82	0,10	0,92	0,81
V		0,48	0,74	0,29	0,20	0,41	0,44	0,05	0,45	0,56	0,66	0,85	0,18	0,67
Cr			0,07	0,80	0,14	0,07	0,05	-0,48	0,68	0,36	0,83	0,17	0,68	0,74
Mn				-0,04	0,04	0,44	0,48	0,02	-0,06	0,52	0,19	0,73	0,07	0,27
Fe					-0,04	0,18	0,37	-0,32	0,70	0,65	0,73	0,13	0,93	0,79
Ni						0,76	0,25	0,62	-0,34	-0,02	0,37	0,18	-0,07	0,19
Cu							0,79	0,68	-0,26	0,56	0,45	0,54	0,26	0,45
Zn								0,51	0,04	0,89	0,49	0,68	0,51	0,65
As									-0,37	0,12	0,01	0,28	-0,32	0,01
Mo										0,39	0,59	0,16	0,48	0,64
Cd											0,62	0,64	0,75	0,80
Sn												0,55	0,65	0,96
Sb													0,13	0,62
Ba														0,74

Narozdíl od mechů, kde bylo významně korelujících párů prvků 30, u hub je korelace prvků spíše výjimečná a opačná korelace není žádná. Opět jde o korelace převážně prvků z půdy (Al, Fe, Cr, Ba), navíc také korelaci Pb-Sn.

5. Závěr

V naší práci jsme sledovali znečištění životního prostředí toxickými kovy na vybraných lokalitách Moravy pomocí biomonitoringu. Pro vyhodnocení znečištění atmosféry jsme použili mech travník Schreberův. Mechy čerpají živiny a vodu jen ze srážek a obsah kovů v nich tedy odráží čistotu ovzduší. Podobně jsme pro hodnocení znečištění půdy použili jeden druh houby – muchomůrku červenou. Pro stanovení obsahu vybraných kovů v těchto biologických materiálech byla použita moderní metoda – hmotnostní spektrometrie s indukčně vázaným plazmatem, která umožňuje stanovení řady prvků během velmi krátkého času.

Lokality na Moravě byly vybrány podle blízkosti našeho bydliště či bydliště spolužáků, kteří nám se sběrem vzorků pomohli. Jednalo se především o jihozápadní okolí Brna, dále Prostějov a horskou lokalitu v Jeseníkách. Na těchto lokalitách byly odebrány vzorky a byly popsány přibližné vzdálenosti cest a významných zdrojů případného znečištění. Po stanovení obsahu kovů ve vzorcích byly výsledky zpracovány podobně jako v dostupných vědeckých článcích o použití biomonitoringu. V případě biomonitoringu znečištění atmosféry byly podle obsahu jednotlivých prvků přiděleny známky 1 až 4, které odpovídají stupni znečištění nízké, střední, vysoké a velmi vysoké. Podle průměrné známky na lokalitě byly lokality rozděleny opět do čtyř tříd se stejným slovním hodnocením. Bylo zjištěno, že v případě znečištění atmosféry jsou nejčistší lokalitou Jeseníky s hodnocením stupně znečištění nízké, podobně jako 66,5 % území České republiky. Středně znečištěnými lokalitami se stupněm znečištění středním jsou lokality Bobrava, Ivančice, brněnská přehrada a Prostějov tak jako 29,5 % území České republiky. Více znečištěné jsou lokality Brno a Silůvky jako jsou znečištěna 3 % ČR a velmi vysoce znečištěná lokalita je Zbraslav, stejně jako 1 % ČR. Znečištěné lokality vykazují vyšší obsah především Al, Fe, V, Co a Ba. Z výsledků je patrná vzájemná souvislost mezi obsahem některých prvků. Tato souvislost byla prokázána korelační analýzou, kdy se projevily významné korelace mezi 30 páry prvků. Vzájemná korelace poukazuje na možnou souvislost znečištění těmito prvky z jednoho zdroje. Tímto zdrojem je zřejmě prach z půdy. Lokality zde spadající do stupně „vysoké“ a „velmi vysoké“ znečištění vykazují vyšší obsahy právě těchto půdních prvků a jsou tedy znečištěny půdním prachem. Naopak prvky, u nichž nebyla

prokázána významná korelace, naše lokality neznečišťují. Jde především o prvky, které se do atmosféry dostávají převážně při zpracování barevných kovů

Jen v náznaku je vidět vliv vzdálenosti nejbližšího možného zdroje znečištění od lokality vzorkování na obsah kovů v mechu. S rostoucí vzdáleností obytné zóny, silnice a lesní cesty od místa vzorkování klesá obsah kovů v mechu. Znečištění ale může přicházet i z větší vzdálenosti, než jsme uvažovali při popisu lokality (tedy řádově nad stovky metrů až kilometry). Pro zhodnocení zdrojů by bylo nutné lépe popsat lokality a brát v úvahu také další parametry, které budou ovlivňovat obsah kovů v mechu, jako je četnost srážek, jejich úhrn, převládající směr větru, průměrná teplota a další přírodní vlivy.

V případě hub nebyly dostupné srovnatelné hodnoty jako pro vyhodnocení znečištění atmosféry. Proto byly lokality jen srovnány mezi sebou. Ukázalo se, že nejnižší obsah kovů byl nalezen v houbě, odebrané u Brněnské přehrady. Nejvyšší obsah kovů byl nalezen opět ve Zbraslavi. Překvapivý byl nepatrný rozdíl mezi obsahem těžkých kovů v houbách ze Zbraslavi a v houbách z Jeseníků, kde bychom v souvislosti s čistou atmosférou očekávali také málo znečištěnou půdu . Na obsah kovů v půdě a potažmo i v houbě má ale také vliv geologické podloží. Nemůžeme v tuto chvíli vyloučit ani vliv techniky odběru a následného zpracování vzorku houby, kdy i malé množství půdy, ulpívající na houbě, může významně ovlivnit výsledek stanovení. Proto by nebylo vhodné z těchto výsledků činit závěry o znečištění půd vlivem činnosti člověka.

Závěrem lze tedy říci, že námi použitá technika biomonitoringu byla úspěšně aplikována alespoň pro sledování znečištění atmosféry vybranými toxickými kovy a získané výsledky korespondují s pracemi uznávaných odborníků v této oblasti. Rozšířením počtu lokalit ve vhodné vzdálenosti od sebe by bylo možno provádět například podrobnější lokální monitoring znečištění atmosféry kovy v rámci kraje či jiného územního celku. Vhodnou analýzou dat lze vysledovat zdroj znečištění a doporučit tak možné kroky k minimalizaci vlivu tohoto zdroje.

6. Seznam použité literatury a internetových stránek

1. Dr. Bruno P. Kremer, Dr. Hermann Muhle: Lišejníky, mechorosty, kapraďorosty; Ikar Praha, spol s.r.o. ve spoluráci s Knižním Klubem 1998
2. V. Garibovová, M. Svrček, J. Baier: Houby; Lidové nakladatelství 1985
3. Heinz Lüllmann, Klaus Mohr, Martin Wehling: Farmakologie a toxikologie; Grada Publishingas, Praha 2004
4. Prof. RNDr. Stanislav Rosypal DrSc. a kol.:Přehled biologie; Státní pedagogické nakladatelství Praha, 1987
5. Ivan Suchar a Julie Sucharová: Current multi-element distribution in forest epigeic moss in the Czech Republic—a survey of the Czech national biomonitoring programme 2000
6. Harry Harmensa, David A. Norris, Georgia R. Koerber,, Alan Buse, Eiliv Steinnes, Ake Rühling: Temporal trends in the concentration of arsenic, chromium, copper, iron, nickel, vanadium and zinc in mosses across Europe between 1990 and 2000
7. Harry Harmens, Alan Buse, Patrick Büker, David Norris, Gina Mills, Bronwen Williams, Brian Reynolds, Trevor W.Ashenden , Ake Rühling and Eivil Steinnes: Heavy Metal Concentrations in European Mosses: 2000/2001 Survey
8. www.biom.cz/legislativa.stm?x=62029 (19. 1. 2008)
9. www.hygiena.gastronews.cz/tezke-kovy-v-zivotnim-prostredi-a-jejich-vliv-na-lidsky-organismus (19. 1. 2008)
10. www.ireferaty.lidovky.cz/321/2840/Mechy (16. 2. 2008)
11. www.wikipedie.cz (15. – 18. 12. 2007)
12. osobní sdělení Mgr. Jiřího Macháta Ph.D.

Středoškolská technika 2008

Setkání a prezentace prací středoškolských studentů na ČVUT

JE KRIMINALITA NEZLETILÝCH OMLUVITELNÁ? UPRAVIT ZÁKONEM HRANICI TRESTNÍ ODPOVĚDNOSTI?

STATISTICKÉ ŠETŘENÍ

Jaroslava Sukeníková, Michal Šišák

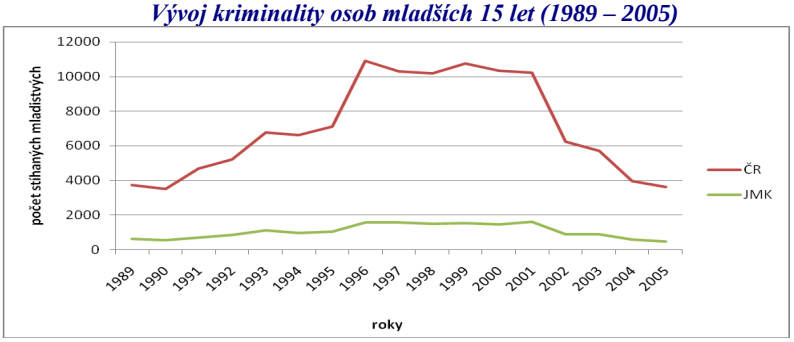
Gymnázium Globe, s. r.o.
Bzenecká 23, 628 000 Brno

Téma této práce jsme si vybrali proto, že jsme se chtěli vrátit k otázce kriminality nezletilých, o které jsme si v naší škole povídali s členy poslanecké sněmovny - D. Filippi a J. Tejcem.

V dnešní době je tato otázka silně medializována. Sdělovací prostředky vnucují myšlenku, že kriminalita nezletilých je čím dál častější, brutálnější a organizovanější, což dokazují pomocí záběrů, na kterých mladiství týrají zvířata, poškozují cizí majetek nebo napadají bezdomovce. Přestože se podle statistik kriminalita nezletilých snižuje, ve všeobecném povědomí převažuje názor, že tomu tak není. Kriminalita nezletilých je velkým sociálním problémem, který vypovídá o stavu dnešní společnosti. Je možno se domnívat, že úpadek morálky společnosti a selhání úlohy rodiny jsou průvodními jevy stoupající kriminality. Je zřejmé, že pokud nejsou dětem odmalička vštěpovány základy chování a práva naší země, je téměř jisté, že děti budou mít blíže k páchání trestné činnosti i v dospělosti.

Ve snaze zabránit růstu kriminality nezletilých došlo k pokusu o novelizaci trestního zákona, který by snižoval hranici trestní odpovědnosti z 15 na 14 let. Tato změna dne 19. 12. 2007 neprošla poslaneckou sněmovnou.

Vývoj kriminality osob mladších 15 let (1989 – 2005)



Uvedené údaje čerpají ze statistik Ministerstva spravedlnosti ČR. Od r. 2001 je zřetelný pokles kriminality osob mladších 15 let.

Jaký může být postih malého kriminálníka? Pokud nezletilý ve věku 12 – 15 let spáchá trestný čin, soud mu může uložit ochranou výchovu. V případě těžkého trestného činu, za který by dospělý člověk mohl být odsouzen k výjimečnému trestu odnětí svobody, soud nezletilému ochranou výchovu uložit musí. Ochranná výchova je však ukládána velmi zřídka. Častěji jsou nezletilí posíláni do ústavní výchovy spolu s dětmi, které trestnou činnost nespáchaly, ale jsou v ústavní péči, protože nemají žádného, kdo by o ně pečoval jako zákonný zástupce. Není správné nechat tyto dvě skupiny nezletilých žít vedle sebe. Děti bez rodičů mohou být negativně ovlivněny ze strany mladistvých, kteří jsou zde na převýchovu. Nejčastější trestnou činností nezletilých je majetková trestná činnost, ze které čerpají finanční prostředky na drogy, alkohol a cigarety. Drobné krádeže jsou však často přehlíženy, zlehčovány a omlouvány nebo nejsou vůbec nahlášeny.

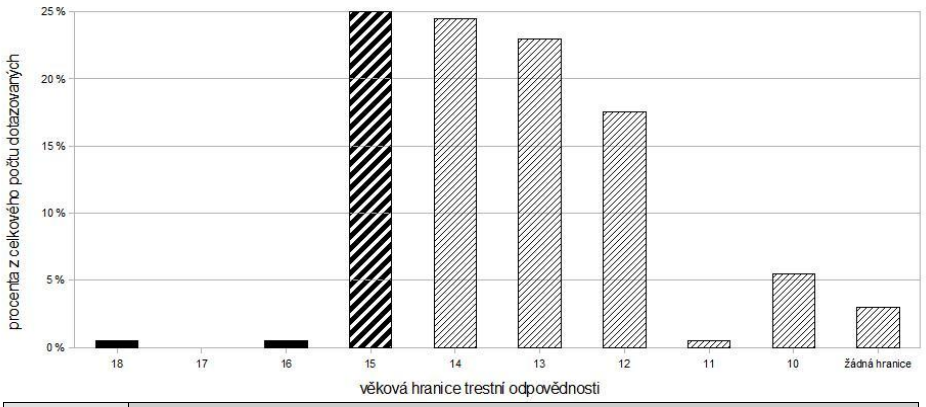
Postatou naší práce bylo provedení průzkumu pohledu občanů na dětskou kriminalitu. Průzkum byl provedený na vzorku 200 osob. Na názory občanů na kriminalitu mladistvých jsme se ptali v okolí našeho bydliště, šlo o obyvatele jihomoravských obcí Tvarožná, Bučovice, Holubice, Šlapanice, Brno – Slatina. Dotazovaným jsme položili dvě jednoduché otázky a dále nás zajímalo, v čem vidí dotazovaní příčinu dětské kriminality. Sledovali jsme také věk a vzdělání respondentů.

VÝSLEDKŮ PRŮZKUMU:

Z odpovědí dotázaných vyplynulo, že jen 10% účastníků ankety by trestnou činnost nezletilých omluvilo, přičemž tento názor zastávají převážně respondenti s ukončeným středoškolským vzděláním (otázka 1). **Většina dotázaných podporuje myšlenku, aby se zákonem snížila věková hranice trestní odpovědnosti pod 15 let** (*viz otázka 2, graf*). **Zákonodárcům bychom doporučili stanovit jako věk trestní odpovědnosti 13 - 14 let.**

Otázka 1:	Je kriminalita nezletilých omluvitelná**?**	
Odpovědi:	**ano**	**10 %**
	ne	**90%**

Otázka 2:	Měla by se upravit zákonem hranice trestní odpovědnosti, která je v současné době 15 let? Na kolik let?	
Odpovědi: (viz také graf)	Hranici si přejí nad 15 let zvýšit 2 lidé z 200 dotázaných	**1%**
	Není nutno posouvat věkovou hranici	**25 %**
	Věkovou hranici je vhodné snížit na 13 - 14 let	**47 %**
	Věkovou hranici je vhodné snížit na 10- 12 let	**24 %**
	Hranici je vhodné úplně zrušit (děti by tak byly za sebe trestně zodpovědné již od narození)	**3%**



Otázka 3:	**V čem tkví příčiny kriminality mladistvých?**	
Odpovědi:	Kriminalita nezletilých je způsobena špatnou výchovou rodičů či selháním rodiny.	21,5%
	Jedinec, včetně nezletilých, nese sám odpovědnost za své činy, protože se takto konat rozhodnul sám.	19,5%
	Mladí lidé mohou být ke kriminalitě naváděni kamarády.	16%
	Nezletilí vědí, že nemohou být postihnuti, proto páchají trestnou činnost.	10,5%
	Kriminalita souvisí se současným stavem romské populace.	7,5%
	Kriminalita je věcí vrozeného charakteru člověka, který se může projevit kdykoliv.	2,5%
	Neví.	22,5%

Dotazovaní v šetření nejvíce zdůraznili vliv rodiny. Zejména u věkových skupin nad 35 let panuje názor, že kriminalita nezletilých je způsobena špatnou výchovou rodičů. Tuto myšlenku naopak nenajdeme u dětí do15 let i skupiny lidí do 25 let se základním vzděláním.

Závěr:

Názor Jaroslavy Sukeníkové, autorky průzkumu (18 let):

Hlavním prostředkem k výchově vyrovnaného a bezproblémového člověka je rodina. Děti z neúplných nebo nefungujících rodin mají ke kriminalitě určitě blíž, než kdyby byly vychovávány v úplném rodinném prostředí. Také rodiče mohou být dětem špatným vzorem. Podobnou úlohu na mladého člověka má i parta kamarádů, kteří ho můžou k trestné činnosti navádět nebo dokonce využívat, třeba právě pro jeho nízký věk. Důležitá je v této oblasti především prevence. Proto je nutná spolupráce školy s rodinou. Určitě je důležité sledovat změny v chování žáků a případné záškoláctví, které signalizuje problém.
I když se většina dotazovaných se klonila k názoru, že by bylo dobré hranici trestní odpovědnosti snížit pod 15 let, nebo alespoň umožnit zákonem, aby ve výjimečných případech mohl být trestně postihnut i nezletilý mladší 15 let, podle mého názoru by se trestní odpovědnost měla ponechat na 15 letech.

Názor Michala Šišáka, druhého z autorů práce (17 let):

Souhlasím s tím, že velký význam na chování člověka má rodina. Když je dítěti kolem 15ti let, tak si říkáme, že už může mít rozum a nedbá se na něj tolik a nevěnuje se mu tolik času. Dítě neví co by, nudí se, začne se potulovat po večerech venku se svými vrstevníky, někdy

hledá zábavu v tom „horším světě“ - začne se řadit do různých spolků nebo gangů. Kdyby rodina byla podpořena společenským míněním v tom, aby dbala na své děti a rodiče s nimi trávili co nejvíce volného času, tak by se podle mě snížila kriminalita. Dalším důležitým faktorem je škola, ta by měla svým studentům umožnit nějaké aktivity na výplň jejich volného času. Dalším problémem je, jaký postoj zastává česká justice. Měla by dávat mnohem tvrdší tresty a nebrat tolik ohled na věk v případě, že dítě někoho zabije nebo udělá těžký zločin. Soudy by si to měly uvědomit a podle provinění dát nezletilé na povinné práce nebo do normální vězeňské cely.
Podle mě, když dospívajícím vyplníme jejich volný čas a budou navštěvovat různé kroužky, tak se kriminalitě v nějakém měřítku musí zamezit. K vandalismu a trestné činnosti nepřispívá jenom alkohol a drogy, ale taky moderní technika, např. počítače, internet a dostupné hazardní herny. U mladých lidí můžeme pozorovat, že ti, co tráví většinu svého času u počítače pak nerozlišují mezi rvačkou nebo střílením ve hře a na ulici. A celkově hlavní chybou podle mě je, velmi tolerantní postoj společnosti k tomuto narůstajícímu problému. Chybou je i malá podpora státu těm, kteří chtějí prevenci zavčas a proto se dospívajícím dětem aktivně věnují.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

MODERNÍ METODY IDENTIFIKACE BIOLOGICKY AKTIVNÍCH LÁTEK – PROTEINŮ

Kristýna Šimoníková , Ivana Vojtová

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA
CHEMICKÁ, BRNO, VRANOVSKÁ 65

KONZULTANTI : RNDr. Hana Kozáčková
Mgr. Ondřej Peš

BRNO 2008

OBSAH

I. ÚVOD

II. TEORETICKÁ ČÁST

1. PROTEOMIKA

1.1. Historie studia proteinů

1.2. Stavební kameny v proteomice

1.2.1. Aminokyseliny

1.2.2. Peptidy

1.2.3. Proteiny

2. SEPARACE V PROTEOMICE

2.1. Elektromigrační techniky

2.1.1. Dvojrozměrná gelová elektroforéza

2.2. Chromatografické techniky

3. HMOTNOSTNÍ SPEKTROMETRIE

3.1. Laserová desorpce a ionizace za účasti matrice

3.2. Průletový analyzátor

3.3. Ionizace elektrosprejem

4. IDENTIFIKACE PROTEINŮ

4.1. Specifické štěpení

4.2. Enzymatické štěpení v gelu a v roztoku

4.3. Peptidové mapování

5. SEZNAM ZKRATEK

III. EXPERIMANTÁLNÍ ČÁST

1. CÍLE PRÁCE

2. POUŽITÉ CHEMIKÁLIE

3. PŘÍSTROJE A VYBAVENÍ

4. EXPERIMENTÁLNÍ PODMÍNKY

5. IDENTIFIKACE MODELOVÝCH PROTEINŮ

5.1. Myoglobin

5.2. Hovězí sérový albumin

5.3. Cytochrom C

IV. DISKUZE A ZÁVĚR

V. POUŽITÁ LITERATURA

I. ÚVOD

V posledních desetiletích vzrůstá potřeba identifikovat v biologickém materiálu jednotlivé bílkoviny; vzniklo tedy nové odvětví biochemie, proteomika, snažící se zjistit proteinovou expresi za daných fyziologických nebo patologických podmínek.

Pro objasnění funkce proteinu je třeba znát jeho primární a sekundární strukturu vedoucí k jeho sbalení.

Metoda peptidového mapování (PMF) patří mezi techniky identifikace proteinů podle jejich primární aminokyselinové struktury.[1, 2, 3] V současnosti se využívá k celé řadě aplikací vyžadující rychlou a úspěšnou metodu identifikace proteinů známé sekvence. [4-7]

Naším úkolem bylo na třech modelových vzorcích proteinů aplikovat metodu PMF a pomocí enzymatického štěpení a hmotnostní spektrometrie MALDI – TOF provést identifikaci.

Teoretická část je zaměřena na nejvýznamnější metody pro identifikaci proteinů, velká pozornost je věnována metodě MALDI – TOF.

II. TEORETICKÁ ČÁST

1. PROTEOMIKA

Termín „proteom“ byl zaveden v roce 1995 Marcelem R. Wilkinsem pro označení veškerého proteinového komplementu kódovaného genomem. Název vznikl z anglické definice „PROTEin completent able to be encoded by a given genOME“ nebo „PROTEin equivalent of a genOME“ [8].

Přestože DNA je nositelem informací o struktuře proteinů, nelze na základě její znalosti objasnit mnohé buněčné procesy. Odhadovaný počet genů člověka je asi kolem 40 – 50 tisíc, zatímco produktů proteinů člověka je odhadován asi na 2 miliony. Studium proteomů je tedy podstatně složitější než studium geonomů, zejména kvůli výskytu více stavebních bloků (20 aminokyselin vs. 4 báze) a absence PCR. Také informace uložená v genomu je relativně stálá, kdežto informace uložená v proteomu se mění s časem. Zatímco lidský genom byl sekvenován během jednoho roku v několika sekvenačních továrních halách na robotických zařízeních a DNA sekvenátorech[9], je pro řešení komplexního zadaní lidské proteomiky nezbytná kombinace širokého spektra metodických přístupů od klasické chemie a biochemie proteinů až po nejnovější techniky automatického zpracování vzorků a identifikace proteinů hmotnostní spektrometrií.[10, 11]

V současnosti je jedním z hlavních cílů proteomiky analýza proteinových komplexů, kdy se využívá současně vysokoúčinných separačních metod jako jsou dvourozměrná elektroforéza, kapalinová chromatografie a hmotnostní spektrometrie’[12]

1.1. Historie studia proteinů

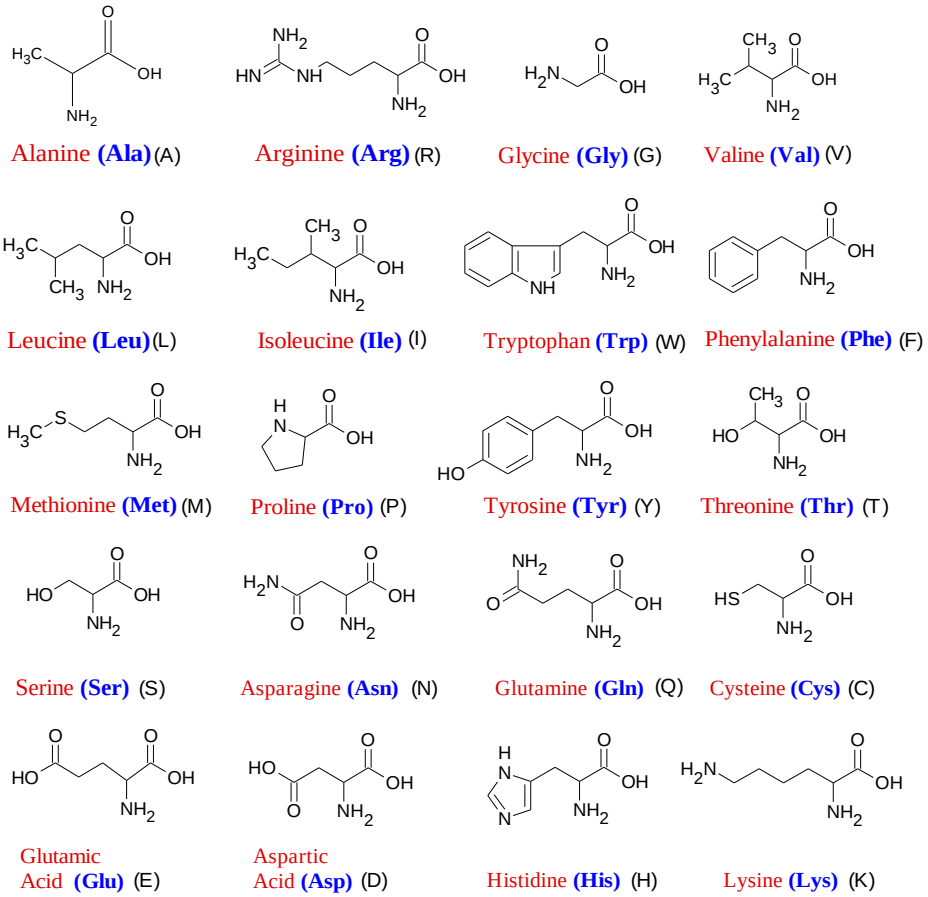
Začátkem 20. století byly proteiny studovány v rámci organické chemie, ale metodami, které nemohly poskytnout dostatek informací o stavbě molekul proteinů a jejich vlastnostech. Proto ztratili organičtí vědci o proteiny na zhruba dalších padesát let zájem.

Proteiny se dále staly středem zájmu fyzikální chemie. V tomto oboru byla sledována jejich hmotnost, tvar a velikost náboje či interakce s prostředím. Byl vytvořen značně zjednodušený model molekuly proteinu – rigidní koule s rovnoměrně rozprostřeným nábojem. Dalším úkolem byla tedy snaha o získání detailních informací o struktuře a funkci proteinů. Výsledky se dostavily až po 2. světové válce, kdy byly vyvinuty nové metody pro analýzu těchto složitých molekul. [13]

1.2. Stavební kameny v proteomice

1.2.1. Aminokyseliny

V přírodě se běžně vyskytuje dvacet aminokyselin uvedených na *Obrázku 1*. Každá aminokyselina obsahuje dvě funkční skupiny navázané na alfa uhlíku; karboxylovou skupinu a aminoskupinu. Dále pak obsahuje alifatický nebo aromatický zbytek, podle jehož charakteru se aminokyseliny dělí do několika skupin (polární, nepolární, kyselé, zásadité). Právě charakter zbytku určuje funkci dané aminokyseliny v polypeptidovém řetězci. Aminokyseliny přítomné v bílkovinách eukaryot se vyskytují zásadně v L-formě. [14, 15]



Obrázek 1: Strukturní vzorce přírodních aminokyselin

1.2.2. Peptidy

Peptidy jsou nízkomolekulární látky tvořené dvěma a více aminokyselinami spojenými peptidovou vazbou (– CO – NH –). Lze je dělit podle počtu aminokyselin v řetězci, podle struktury, případně podle způsobu výroby.

Při molekulové hmotnosti větší než cca deset tisíc hmotnostních jednotek se již hovoří o proteinech.

1.2.3. Proteiny

Proteiny, neboli bílkoviny, jsou základní stavební jednotkou všech živých organismů. Jedná se o přírodní vysokomolekulární sloučeniny, jejichž strukturu lze rozdělit na primární, sekundární, terciární a kvartérní. Chemické a fyzikální vlastnosti aminokyselin a jejich kombinace pak udávají jak prostorové záhyby polypeptidového řetězce, z nichž je „stvořena" konečná podoba proteinu, tak i konečnou funkci proteinu.

Primární strukturu proteinu určuje sekvence aminokyselin v polypeptidovém řetězci. Sekundární struktura proteinu je dána typy interakcí v jednotlivých úsecích polypeptidového řetězce a tvořena dvěma základními tvary; alfa- šroubovicí a beta-skládaným listem, které vzniknou v důsledku tvorby vodíkových vazeb mezi –CO a –NH skupinami peptidových vazeb stočeného polypeptidu. Terciární struktura udává trojrozměrný tvar polypeptidového řetězce a tím i jeho funkci. Hlavním důvodem vzniku terciární struktury je různá chemická povaha postranních aminokyselinových zbytků schopných tvořit nekovalentní vazby. Kvartérní struktura určuje prostorové uspořádání dvou a více polypeptidů v jednom proteinu, které jsou navzájem propojeny vodíkovými můstky a svými postranními řetězci sulfidickými nebo i jinými vazbami.

Podle struktury dělíme proteiny na globulární (albuminy, globuliny, histony) nebo fibrilární (keratin, kolagen, aktin, myozin...).[16]

Podle funkce lze proteiny dělit na stavební, „house-keeping", transportní, speciální funkci zaujímají enzymy.

Enzymy jsou bílkovinné makromolekuly obdařené katalytickými funkcemi, které mohou ovlivňovat chemické přeměny. Nacházejí se ve všech živých systémech a jejich počet se odhaduje na miliardy.

2. SEPARACE V PROTEOMICE

2.1. Elektromigrační techniky

Elektromigrační separační metody využívají dvou elektrokinetických jevů – elektroforézy a elektroosmózy. V prostředí obsahujícím roztok s nabitými částicemi a pevné povrchy stýkající se s roztokem, které mohou nést elektrické náboje (stěny kapiláry, povrchy přítomných částic), se vytvářejí elektrické dvojvrstvy a mezi nimi vzniká určité rovnovážné rozdělení nábojů.

V elektromigračních separačních metodách je na toto prostředí připojeno stejnosměrné elektrické pole, které poruší rovnováhu v rozložení nábojů a vyvolá jejich pohyb. Principem separace složek vzorku je rozdílná rychlost jejich migrace, neboť nabité částice různých složek se v určitém prostředí liší svou elektroforetickou pohyblivostí. [17]

2.1.1. Dvojrozměrná gelová elektroforéza (2DE GE)

Základem této techniky je využití dvou fyzikálně – chemických vlastností proteinů. Tímto způsobem lze separovat velmi komplexní směsi, obsahující tisíce proteinů a porovnávat dva a více vzorků při určování proteinové exprese.

Gelová elektroforéza spočívá v migraci elektricky nabitých částic v elektrickém poli, které je vytvořeno vložením konstantního stejnosměrného napětí mezi dvě elektrody. Slouží k oddělení jednotlivých proteinů ze směsi. Jednotlivé vzorky se vloží na polyakrylamidový gel. V tomto gelu se jednotlivé proteiny pohybují různou rychlostí a podle toho se rozdělí na malá uskupení, kde je velká pravděpodobnost, že se jedná o stejný protein. [18, 19]

2.2. Chromatografické techniky

Chromatografické metody využívají k dělení složek směsi mnohonásobného opakovaného vytváření rovnovážných stavů složek mezi dvěma fyzikálně odlišnými fázemi, fází stacionární (nepohyblivou), a fází mobilní (pohyblivou). Proces separace je charakterizován rovnovážnými stavy vytvářenými na základě různých (kvantitativně i kvalitativně) fyzikálně – chemických interakcí mezi složkou a mobilní fází, složkou a stacionární fází a také mezi mobilní a stacionární fází.

Metoda vysoce účinné kapalinové chromatografie (HPLC – High Performance Liquid Chromatography) je standardní kolonová technika pro separaci proteinů a peptidů s vysokým rozlišením zón. Podstatou separace je rozdílná distribuce analytů ve vzorku mezi stacionární

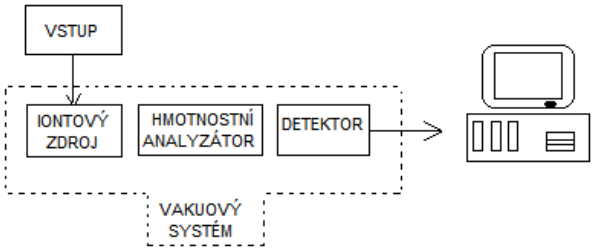
(sorbent) a mobilní fází (eluent). Gelová permeační chromatografie (GPC – Gel Permeation Chromatography, SEC – Size Exclusion Chromatography) využívá mechanického dělení molekul analytů v pórech gelu na základě jejich rozdílné velikosti (molekulové hmotnosti) a tvaru. Využívá se k preparativním separacím, ke stanovení molekulových hmotností, resp. při odstraňování nízkomolekulárních sloučenin (odsolování).

Iontově výměnná chromatografie (IEC – Ion Exchange Chromatography) využívá rozdílné výměnné adsorpce analytů (iontů) na povrchu iontového měniče, neboli ionexu.

3. HMOTNOSTNÍ SPEKTROMETRIE

Je fyzikálně chemická metoda určování hmotností atomů, molekul a jejich částí po jejich převedení na ionty. Hmotnostní spektrometrie převádí vzorek na ionizovanou plynnou fázi a vzniklé ionty separuje podle hodnoty podílu jejich hmotnosti a náboje (*m/z)* . Při ionizaci vzorku se rozlišuje „měkká“ a „tvrdá“ ionizace, kdy při měkké ionizaci nedochází k významné fragmentaci analytu, a proto je vhodnější pro analýzu větších molekul (proteiny, polysacharidy, nukleové kyseliny, atd.).

Mezi tvrdé ionizační metody patří např.: elektronová ionizace (EI), ionizace elektrickým polem (FI) nebo laserová ionizace (LDI). Naproti tomu mezi měkké ionizační techniky patří např.: chemická ionizace (CI), sprejové techniky termosprej (TSI), elektrosprej (ESI), případně ionizace sekundárními ionty (SIMS). [20]



Obrázek 2: Blokové schéma hmotnostního spektrometru

3.1. Laserová desorpce a ionizace za účasti matrice (MALDI)

Je rozšířeným nástrojem pro analýzu přírodních látek i syntetických polymerů. Také za objev této metody obdržel Koichi Tanaka v roce 2002 Nobelovu cenu.[21] Dříve byla používána pouze pro analýzu peptidů a proteinů ale teď se používá i k analýze nízkomolekulárních organických a anorganických látek nebo k analýze nukleových kyselin. Velkou výhodou této metody je rychlost měření a velká citlivost. [20]

Proces MALDI zahrnuje tři kroky:

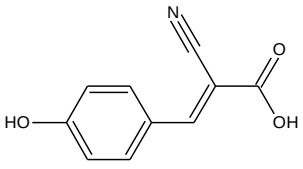
1. Vytvoření směsných krystalů analytu a matrice – v důsledku vysokého přebytku matrice jsou molekuly analytu od sebe izolovány, což vede k vytvoření „pevného roztoku".
2. Excitaci matrice – jakmile je laserový puls zaostřen na místo s krystaly analytu a matrice, většinu laserového záření absorbují chromofory matrice, čímž dochází k její rychlé desorpci. Ta s sebou strhává i molekuly analytu, který se odpaří a ochladí dříve, než se rozloží.
3. Ionizaci analytu – vzhledem ke kyselé povaze matrice je analyt ionizován především přenosem protonu: $MH^+ + ABCD \rightarrow ABCDH^+ + M$ (M je MALDI matrice)

Je možný i přenos kationtu (např. Na^+, K^+), obvykle v případech nedostatečně odsolených vzorků.
Ionty jsou poté pomocí elektrického pole extrahovány do hmotnostního separátoru, kterým je nejčastěji průletový analyzátor (TOF – Time-Of-Flight). Jako matrice se používají zejména organické kyseliny (*Tabulka 1*), které vyhovují podmínce absorpce při vlnové délce použitého laseru (obvykle 337 nm).

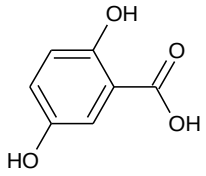
Tabulka 1: Nejčastěji používané typy matric v MALDI MS

Vzorky	Matrice
Peptidy < 10 kDa	α - kyano – 4 – hydroxyskořicová kyselina (CHC)
Proteiny > 10 kDa	2,5 – dihydroxybenzoová kyselina (DHB)
Oligonukleotidy	2,5 – dihydroxybenzoová kyselina (DHB)
Polymery	α - kyano – 4 – hydroxyskořicová kyselina (CHC)

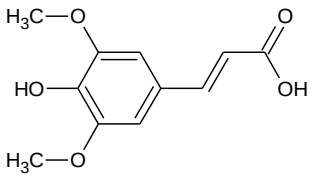
a) CHC



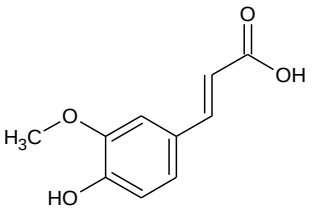
b) DHB



c) CHC



d) MHC

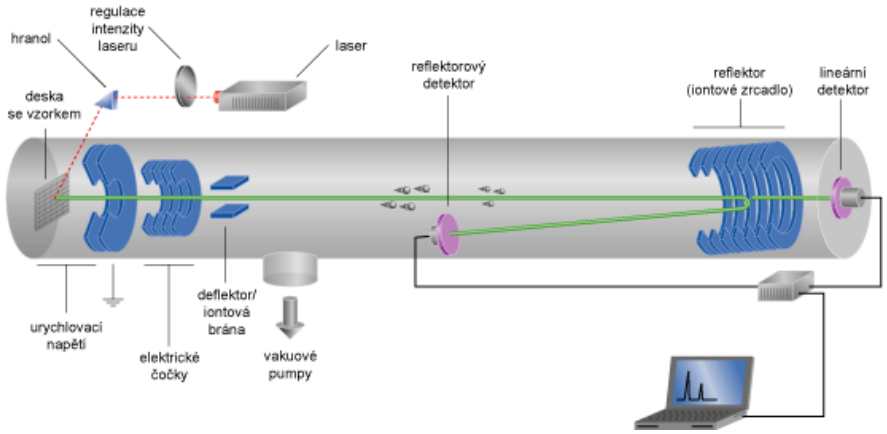


Obrázek 3: Příklady nejpoužívanějších matric

3.2. Průletový analyzátor

Průletový analyzátor (TOF) se využívá k separaci rozdílné rychlosti iontů v závislosti na poměru *m/z*. Jedná se o pulsní hmotnostní analyzátor; nejdříve jsou pulsem extrakčního napětí ionty urychleny na vstupu do analyzátorové trubice, získaná kinetická energie je dána rozdílem elektrického napětí ve zdroji. Hlavní výhodou je schopnost detekce molekul bez teoretického limitu *m/z* a krátká doba záznamu spektra. Vysoká citlivost je dána extrakcí výrazné většiny iontů nacházejících se v iontovém zdroji a jejich transmisí k detektoru. Pro zvýšení rozlišení se využívá dvou konstrukčních prvků, které přispívají k zúžení disperze kinetické energie iontů. Použití tzv. iontového zrcadla, neboli reflektoru, slouží ke kompenzaci různých kinetických energií pro ionty se stejnou hodnotou *m/z.* Ionty s vyšší kinetickou energií proniknou hlouběji do odrazového elektrického pole reflektoru, čímž dojde k jejich opoždění oproti iontům s nižší kinetickou energií a tím i k vyrovnání celkové doby strávené v letové trubici, protože hloubka průniku iontů do elektrického pole reflektoru je přímo úměrná pouze jejich kinetické energii a nezávisí na hodnotě *m/z*. Druhým prvkem pro zvýšení rozlišení je pulsní (zpožděná) extrakce, která koriguje nerovnoměrnou disperzi počátečních kinetických energií iontů. Po laserovém pulsu je aplikováno nulové extrakční napětí po dobu cca 100 až 1000 ns. Během tohoto zpoždění se distribuce iontů v iontovém zdroji změní tak, že ionty o stejném *m/z*, ale s vyšší kinetickou energii se budou nacházet dále od nosiče vzorku než ionty s nižší kinetickou energií. Následnou aplikací extrakčního napětí dojde k vyrovnání distribuce počátečních kinetických energií pro ionty o stejném *m/z*.

Nevýhodou zpoždění extrakce je funkčnost pouze ve vymezeném, předem zvoleném, intervalu *m/z*. [20]



Obrázek 4 :Schéma hmotnostního spektrometru MALDI – TOF s iontovým zrcadlem [22]

3.3. Ionizace elektrosprejem

Ionizace elektrosprejem (ESI) je spolu s MALDI nejpoužívanější technikou měkké ionizace v hmotnostní spektrometrii. Objev ESI se připisuje Malcomovi Dole, který v 60. letech 20. století poprvé demonstroval využití elektrospreje k ionizaci bez poškození molekuly.

O dvacet let později byla publikována práce Johna Fenna, který se svými spolupracovníky v laboratoři úpěšně využil ESI k měkké ionizaci netěkavých, termálně labilních, vysokomolekulárních biologických látek a jejich následné analýze pomocí MS.[23] Tato práce později vedla k udělení Nobelovy ceny za chemii v roce 2002. Jedná se o velmi měkkou ionizaci, kterou lze analyzovat proteiny, peptidy, oligosacharidy a jiné.

4. IDENTIFIKACE PROTEINŮ

4.1. Specifické štěpení

Enzymatické štěpení, neboli trávení, je druhem specifického štěpení využívaného při analýze proteinů. Jedná se o specifické porušení peptidové vazby, v závislosti na použitém enzymu. Produktem je směs peptidů, která vykazuje charakteristický peptidový otisk daného proteinu. Obliba enzymatického štěpení u analýz proteinů tkví v přesunu hmotností z oblasti vysokých hodnot *m/z* do oblasti nízkých hodnot *m/z*. Zde jsou parametry hmotnostních spektrometrů výrazně lepší a izotopová obálka píků je užší.

Užití proteolytických enzymů pro peptidové mapování je velmi časté, nicméně optimální podmínky štěpení jsou obvykle stanoveny individuálně. Existuje celá řada využívaných enzymů (lysin, chymotrypsin, papain, aj.), ale nejčastěji používaným enzymem je trypsin.

Trypsin se řadí mezi skupinu proteolytických enzymů označovaných jako serinové proteasy. Charakterizuje je přítomnost extrémně reaktivního serinového zbytku, který je zodpovědný za jejich enzymovou aktivitu. Struktura trypsinu byla objasněna v roce 1967 rentgenovou krystalografií. Spolu s chymotrypsinem a elastasou patří do skupiny trávicích enzymů, označovaných jako endopeptidasy. Jsou syntetizovány pankreatickými buňkami jako proenzymy, které je potřeba aktivovat prostřednictvím enteropeptidasy za specifického odštěpení N-terminálního hexapeptidu. Tím se změní konformace molekuly, obnaží se aktivní centrum a trypsin, schopný další aktivace sám sebe, tzv. autokatalýzy, se stává aktivním. [24]

4.2. Enzymatické štěpení v gelu a v roztoku

Štěpení v gelu je většinou využíváno po separaci z 2D GE. Ukotvení proteinu v gelu má své výhody – při modifikaci proteinu nebo promývání, ale taky své nevýhody – gel znesnadňuje přístup enzymu k proteinu, což vede ke zvýšené době štěpení a přebytku enzymu.

Při štěpení v roztoku není protein nijak ukotven a díky tomu má enzym k proteinu volný přístup. Autolýza (rozkladné procesy vyvolané vlastními enzymy) trypsinu je zde výrazně nižší. Nevýhodou je fakt, že přidaná činidla zůstávají s analytem v roztoku a může dojít ke kontaminaci analytu, případně znesnadnění MS analýzy. [25]

4.3. Peptidové mapování (PMF)

Analýzou na MALDI – TOF MS je získáno hmotnostní spektrum fragmentů enzymatického štěpení. V případě trávení pomocí trypsinu je získáno spektrum s fragmenty peptidů s hodnotou *m/z* převážně v rozmezí 800 – 3500 Da, které jsou specifické pro daný protein. V mezinárodních databázích se nacházejí sekvence proteinů, které lze pomocí různých komerčních nebo volně dostupných programů teoreticky rozštěpit a porovnat s naměřenými hodnotami, což může vést k identifikaci proteinů. Pro všechny je společný způsob hledání na základě několika parametrů. V každém případě musí být zadány hmotnosti jednotlivých peptidových fragmentů nalezených v hmotnostním spektru a typ použitého enzymu. Dále je potřeba určit toleranci *m/z*, která určuje možnou odchylku hodnoty *m/z* od teoretické hodnoty a výrazně ovlivňuje výsledky hledání. Pokud jsou známy podrobnější informace o štěpeném proteinu, lze zúžit počet kandidátů, např. zadáním rozmezí molekulových hmotností proteinu a nebo p*I* proteinu, možných modifikacích proteinu, typ použitého hmotnostního spektrometru. Dále je možno zadat počet míst, která nebyla rozštěpena; obvykle se udává jedno, maximálně dvě, protože výrazně vzroste počet hmotností, které by mohli odpovídat specifickému štěpu proteinu a tím snížit relevanci výsledků. Výsledkem databázového hledání je seznam kandidátů proteinů, které by mohlo odpovídat původnímu proteinu. Každý výsledek je charakterizován nejméně třemi číselnými parametry, skóre, poměr nalezených *m/z* ku zadaným *m/z* peptidů a procento pokrytí sekvence. Platí, že čím jsou tato čísla vyšší, tím vyšší je pravděpodobnost správné identifikace proteinu. Procentuální pokrytí sekvence proteinu zahrnuje sekvenci nalezených peptidových fragmentů do celkové sekvence proteinu. Jinak řečeno, určuje procento pokrytí všech aminokyselin v proteinu na základě znalostí hmotností jednotlivých specifických peptidů. [22, 26]

Příklady volně přístupných databází pro PMF:

MS – Fit – ProteinProspector [27], MASCOT [28], PeptideSearch [29], aj.

5. SEZNAM ZKRATEK

2DE GE – dvojrozměrná gelová elektroforéza

A – adenin

ACN – acetonitril

ACTH – adrenokortikotropní hormon

Arg – arginin

BSA – hovězí sérový albumin

C – cytosin

CI – chemická ionizace

DHC – skořicová kyselina

EI – elektronová ionizace

ESI – ionizace elektrosprejem

FAB – ionizace urychlenými atomy

FI – ionizace elektrickým polem

G – guanin

GPC, SEC – gelová permeační chromatografie

HPLC – kapalinová chromatografie

CHC – kyselina α-kyano-4-hydroxyskořicová

IEC – iontově výměnná chromatografie

LDI – laserová desorpce/ionizace

Lys – lysin

MALDI – laserová desorpce/ionizace za účasti matrice

MYO - myoglobin

MS – hmotnostní spektrometrie

pI – izoelektrický bod

PMF – peptidové mapování

SC – pokrytí sekvence

SIMS – ionizace sekundárními ionty

T – thymin

TFA – kyselina trifluorooctová

TOF – průletový analyzátor

TSI – ionizace termosprejem

III. EXPERIMENTÁLNÍ ČÁST

1. CÍLE PRÁCE

Cílem této práce bylo seznámit se s základními postupy proteomické analýzy formou rešerše literatury a na jedné z nich, metodě PMF, si prakticky vyzkoušet a osvojit nabyté poznatky.

V teoretické části byly uvedeny některé z postupů, kterými se současná proteomika řídí při analýze proteinů a peptidů.

V praktické části je na metodě peptidového mapování (PMF) demonstrována detailní procedura vhodná pro rychlou identifikaci proteinů známé sekvence pomocí MALDI – TOF.

Na třech modelových proteinech je technika PMF aplikována a výsledky jsou přehledně vyhodnoceny a interpretovány.

2. POUŽITÉ CHEMIKÁLIE

Acetonitril, C_2H_3N, ACN, 99.99% (Scharlau, Německo)

Adrenokortikotropní hormon fragment 18– 39, ACTH, 99 %, M_R = 2464,1 g.mol^{-1} (Sigma-Aldrich, Německo)

Angiotensin lidský, 97 %, M_R = 1295,6 g.mol^{-1} (Sigma-Aldrich, Německo)

Bradykinin, 99 %, M_R = 1059,5 g.mol^{-1} (Sigma-Aldrich, Německo)

Hovězí sérový albumin, BSA, M_R ~ 66 430 g.mol^{-1} (Sigma-Aldrich, Německo)

Hydrogenuhličitan amonný, NH_4HCO_3, 99% (Sigma-Aldrich, Německo)

Kyselina trifluorooctová, CF_3COOH, 99 % (Sigma-Aldrich, Německo)

Kyselina α-kyano-4-hydroxyskořicová, CHC, M_R = 189,2 g.mol^{-1} (Sigma-Aldrich, Německo)

Redestilovaná voda z křemenné aparatury (Heraeus, Německo)

Renin substrát tetradekapeptid, 96 %, M_R = 1758,9 g.mol^{-1} (Sigma-Aldrich, Německo)

Trypsin vepřový, specifická aktivita >5000 jednotek.mg^{-1} (Promega, USA)

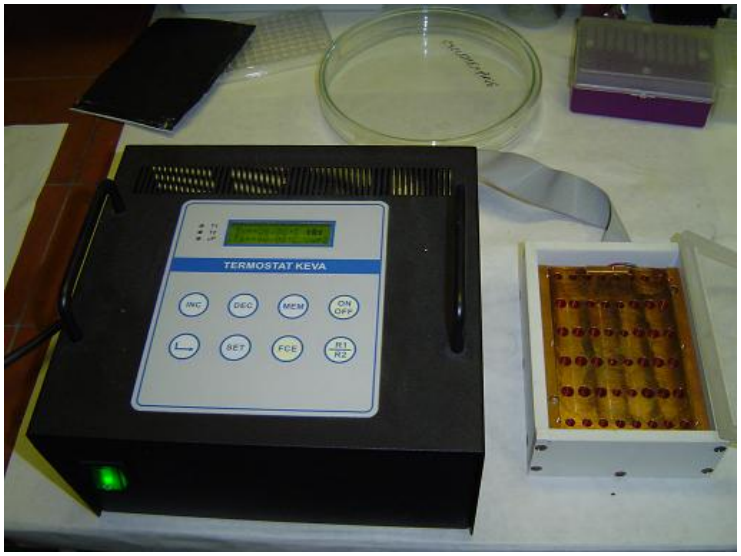
3. PŘÍSTROJE A VYBAVENÍ

Hmotnostní spektrometr MALDI-TOF, Axima CFR, Kratos Analytical Shimadzu Corporation(Velká Británie). Dusíkový laser 337 nm, relativní jednotky výkonu (0 až 180), maximu odpovídá výkon 6 mW při frekvenci 10 Hz. *Software*: Kratos Analytical Launchpad ver. 2. 2. 1. (viz *Obrázek 5*)

Termostat KEVA, Ing. Pavel Krásenský, Katedra analytické chemie PřF MU (Česká Republika), rozsah teplot 30 – 75 °C, umožňuje zahřívání vialek nebo MALDI terčíku.
Ke zpracování výsledků byl použit software MS Excel 2000, Microsoft (USA).
(viz *Obrázek 6*)



Obrázek 5: Foto MALDI TOF MS



Obrázek 6: Foto termostatu KEVA

4. EXPERIMENTÁLNÍ PODMÍNKY

V literatuře existuje celá řada návodů pro enzymatické štěpení proteinů. [30,31,32] Jednotlivé postupy se liší zejména poměrem enzymu k substrátu a výběrem vhodných reakčních podmínek. Přestože se tyto práce věnují převážně problematice štěpení v gelu, je možno aplikovat některé podmínky i pro štěpení v roztoku. Mezi hlavní parametry ovlivňující štěpení patří teplota a doba enzymatického štěpení. Současným trendem je co nejkratší doba analýzy, proto se objevily postupy s poměrně krátkou dobou štěpení za zvýšené teploty. Nevýhodou štěpení za vysoké teploty může být denaturace enzymu, kdy vlivem porušení sekundární a terciární struktury dochází ke ztrátě enzymové aktivity. Trávení může probíhat v roztocích s obsahem organického rozpouštědla i nad 50 %, jde převážně o acetonitril, který napomáhá kratším časům při štěpení a přispívá ke zlepšení kokrystalizace analytu s matricí. Důležitým faktem je p*H*, optimální rozmezí p*H* při štěpení trypsinem je 7 – 9.

V současnosti je enzymatické štěpení jedním z nejvýznamnějších faktorů omezujících meze detekce proteinů pomocí MS, a tím i identifikaci proteinů. Se snižující se koncentrací proteinu se při zachování poměru enzym-substrát doba štěpení neúnosně prodlužuje. Do jisté míry je možné zvýšit koncentraci enzymu, ale vysoký poměr enzym-protein může vést k dominanci produktů autolýzy enzymu ve spektru a jiným komplikacím (nespecifické štěpení, výměnné interakce mezi disulfidickými můstky). Tyto enzymové fragmenty mohou potom rušit i původní signál analytu.

Na základě těchto informací jsme se rozhodli pro následující návrh experimentu. Jako modelové proteiny byly zvoleny BSA, cytochrom C a myoglobin.Teplota byla zvolena 40 ˚C a doba štěpení 2 hodiny. Hmotnostní poměr protein : enzym byl 10:1.

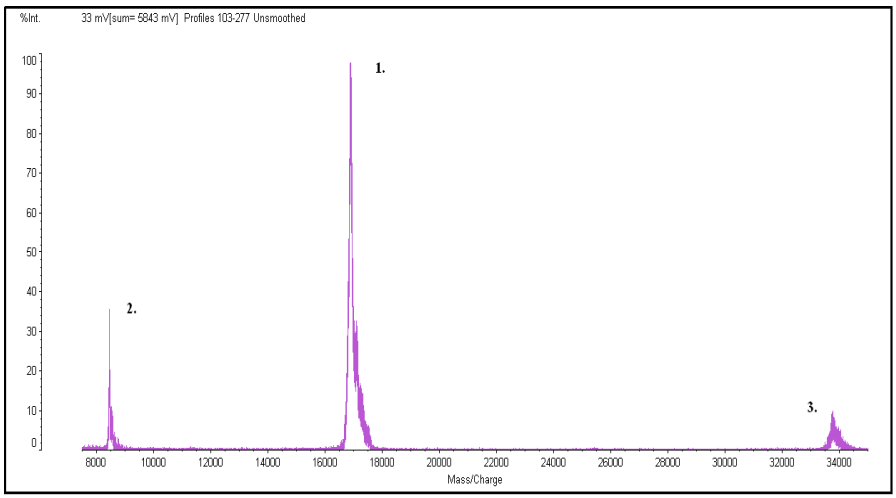
Pro MALDI – TOF MS analýzu intaktních proteinů byl připraven čerstvý roztok skořicové kyseliny (DHC) o koncentraci 10 $mg.ml^{-1}$.

Při trávení byl použit amonný pufr zajišťující alkalické prostředí (pH ~ 9). Pro MALDI-TOF MS analýzu peptidových fragmentů byl připraven čerstvý roztok matrice CHC o koncentraci 10 $mg.ml^{-1}$ v jednom dílu ACN a v jednom dílu 0,2% TFA. Přídavek kyselého roztoku matrice do reakční směsi sloužil také k zastavení reakce.

5. IDENTIFIKACE MODELOVÝCH PROTEINŮ

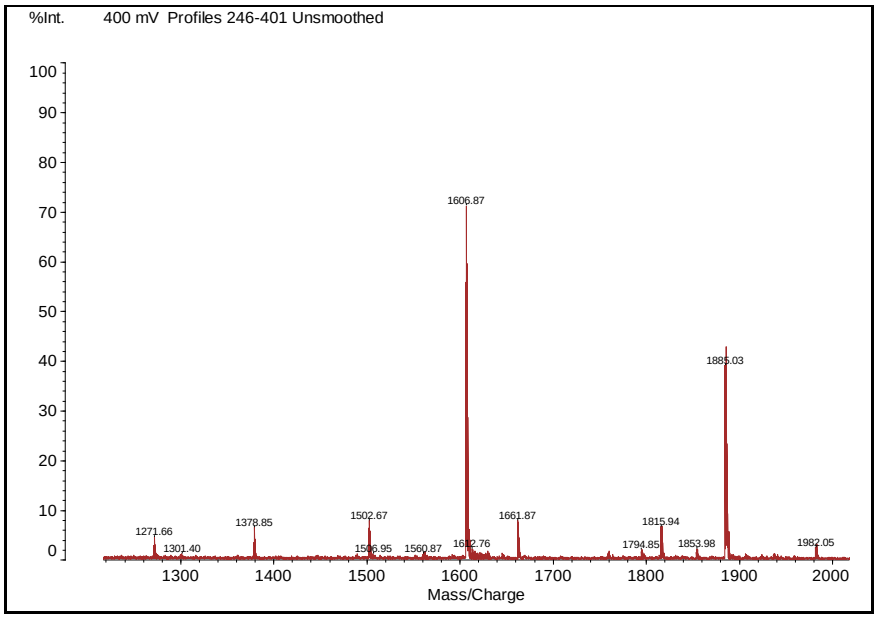
Myoglobin

Myoglobin patří do skupiny transportních proteinů vyskytujících se ve svalech. Relativní molekulová hmotnost je asi 16 700 (*Obrázek 8*). Štěpení myoglobinu o koncentraci 1 mg·ml^{-1} bylo prováděno trypsinem o koncentraci 0,1 mg·ml^{-1} za podmínek uvedených v kapitole 4. Naměřené MS spektrum získaných peptidů je na *Obrázku 9*.



Obrázek 8:MALDI - TOF spektrum intaktního myoglobinu

Pík označený číslem 1 odpovídá částici $[MYO + H]^+$. Pík s číslem 2 náleží částici $[MYO + 2H]^{2+}$ a pík s číslem 3 částici $[(2 \cdot MYO) + H]^+$.



Obrázek 9: MALDI - TOF spektrum peptidových fragmentů myoglobinu

Na *Obrázku 9* je zobrazeno spektrum rozštěpeného proteinu a hodnoty *m/z* signifikantních peptidů.

Tabulka 2: Souhrnné zobrazení odpovídajících peptidů pomocí programu MASCOT

umístění aminokyselin v řetězci	Mr_{exp}	Mr_{teor}	rozdíl Mr (ppm)	missed cleavage	sekvence
2 - 17	1815.9400	1814.8952	21	0	MGLSDGEWQQVLNVWGKV
18 - 32	1606.8700	1605.8475	10	0	KVEADIAGHGQEVLIRL
33 - 43	1271.6600	1270.6557	-2	0	RLFTGHPETLEKF
33 - 46	1661.8700	1660.8461	10	1	RLFTGHPETLEKFDKF
65 - 78	1378.8500	1377.8344	6	0	KHGTVVLTALGGILKK
80 - 97	1982.0500	1981.0493	-3	1	KKGHHEAELKPLAQSHATKH
81 - 97	1853.9800	1852.9544	10	0	KGHHEAELKPLAQSHATKH
104 - 119	1885.0300	1884.0145	4	0	KYLEFISDAIIHVLHSKH
120 - 134	1502.6700	1501.6620	0	0	KHPGDFGADAQGAMTKA

Výsledek hledání v databázi je uveden v *Tabulce 2*. V prvním sloupci tabulky je uveden začátek a konec peptidového fragmentu daný aminokyselinou v polypeptidovém řetězci. V druhém sloupci jsou naměřené relativní molekulové hmotnosti analyzovaných fragmentů, ve třetím sloupci jsou uvedeny teoretické hodnoty relativní hmotnosti peptidů. Ve čtvrtém sloupci je poměrný rozdíl těchto hmotností uvedený v ppm. Pátý sloupec udává počet míst, které mohly být trypsinem nerozštěpeny, a přesto zahrnuty do identifikačního algoritmu. Šestý sloupec pak jen uvádí sekvenci aminokyselin v nalezeném úseku peptidového řetězce.

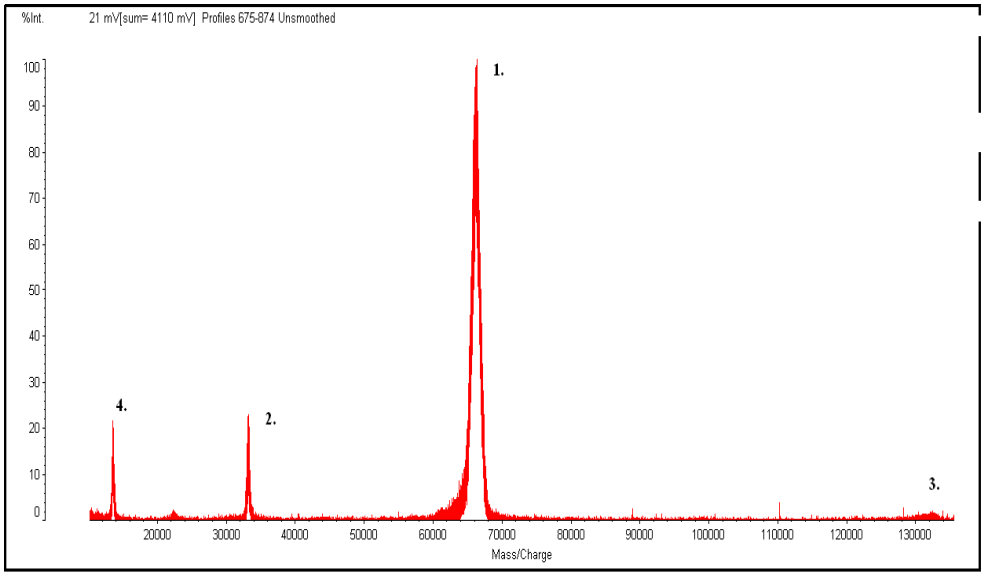
Na *obrázku 10* je uvedena celá sekvence myoglobinu, přičemž červeně vyznačené části patří mezi úspěšně identifikované fragmenty.

M**GLSDGEWQQ VLNVWGKVEA DIAGHGQEVL IRLFTGHPET LEKFDK**FKHL KTEAEMKASE DLKK**HGTVVL TALGGILKK****K GHHEAELKPL AQSHATK**HKI PIK**YLEFISD AIIHVLHSKH PGDFGADAQG AMTK**ALELFR NDIAAKYKEL GFQG

Obrázek 10: Primární struktura myoglobinu, pokrytí sekvence: 70 %

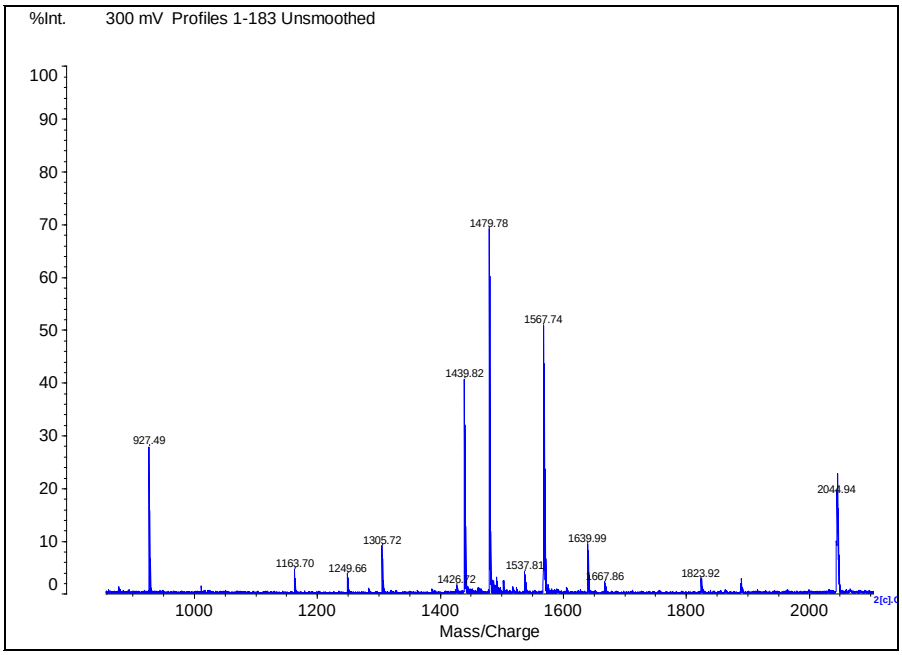
Hovězí sérový albumin

Hovězí sérový albumin (BSA) patří do skupiny sérových proteinů, jehož hlavní funkcí je udržování osmotického tlaku potřebného pro správnou distribuci tělních tekutin mezi nitroceluózními kompartmenty a buněčnou tkání. Jeho relativní molekulová hmotnost je asi 66 000 (viz *Obrázek 11*).Štěpení BSA o koncentraci 1 $mg \cdot ml^{-1}$ bylo prováděno trypsinem o koncentraci 0,1 $mg \cdot ml^{-1}$ za podmínek uvedených v kapitole 4. Naměřené MS spektrum získaných peptidů je na *Obrázku 12*.



Obrázek 11:MALDI - TOF spektrum intaktního BSA

Pík označený číslem 1 odpovídá částici $[BSA + H]^{+}$. Pík s číslem 2 je $[BSA + 2H]^{2+}$, pík s číslem 3 částice $[(2 \cdot BSA) + H]^{+}$ a pík s číslem 4 odpovídá částici $[BSA + 3H]^{3+}$.



Obrázek12: MALDI - TOF spektrum peptidových fragmentů BSA

Na *Obrázku 12* je zobrazeno spektrum rozštěpeného proteinu a hodnoty *m/z* signifikantních peptidů.

Tabulka 3: Souhrnné zobrazení odpovídajících peptidů pomocí programu MASCOT

umístění aminokyselin v řetězci	Mr_{exp}	Mr_{teor}	rozdíl Mr (ppm)	missed cleavage	sekvence
35 – 44	1249.6600	1248.6139	31	1	RFKDLGEEHFKG
66 – 75	1163.7000	1162.6234	60	0	KLVNELTEFAKT
161 – 167	927.4900	926.4861	-4	0	KYLYEIARR
168 – 183	2044.9400	2044.0206	-43	1	RRHPYFYAPELLYYANKY
347 – 359	1567.7400	1566.7354	-2	0	KDAFLGSFLYEYSRR
360 – 371	1439.8200	1438.8045	6	1	RRHPEYAVSVLLRL
402 – 412	1305.7200	1304.7088	3	0	KHLVDEPQNLIKQ
421 – 433	1479.7800	1478.7881	-10	0	KLGEYGFQNALIVRY
437 – 451	1639.9900	1638.9305	32	0	RKVPQVSTPTLVEVSRS
469 – 482	1667.8600	1666.8058	28	0	RMPCTEDYLSLILNRL
483 – 489	841.4500	840.4527	-12	0	RLCVLHEKT
508 – 523	1823.9200	1822.8924	11	0	RRPCFSALTPDETYVPKA

Výsledek hledání v databázi je uveden v *Tabulce 3*. V prvním sloupci tabulky je uveden začátek a konec peptidového fragmentu daný aminokyselinou v polypeptidovém řetězci. V druhém sloupci jsou naměřené relativní molekulové hmotnosti analyzovaných fragmentů, ve třetím sloupci jsou uvedeny teoretické hodnoty relativní hmotnosti peptidů. Ve čtvrtém sloupci je poměrný rozdíl těchto hmotností uvedený v ppm. Pátý sloupec udává počet míst, které mohly být trypsinem nerozštěpeny, a přesto zahrnuty do identifikačního algoritmu.

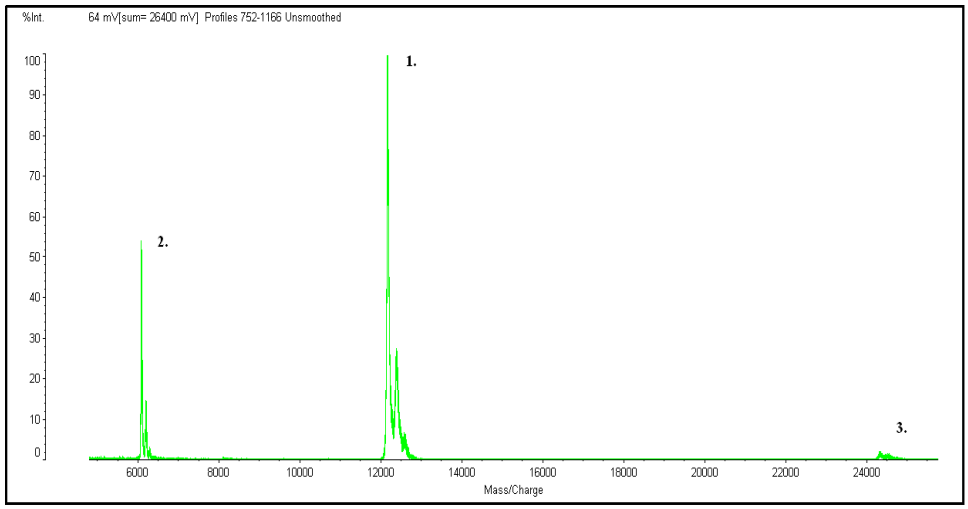
Na *obrázku 13* je uvedena celá sekvence BSA, přičemž červeně vyznačené části patří mezi úspěšně identifikované fragmenty.

MKWVTFISLL LLFSSAYSRG VFRRDTHKSE IAHR**FKDLGE EHFK**GLVLIA
FSQYLQQCPF DEHVK**LVNEL TEFAK**TCVAD ESHAGCEKSL HTLFGDELCK
VASLRETYGD MADCCEKQEP ERNECFLSHK DDSPDLPKLK PDPNTLCDEF
KADEKKFWGK **YLYEIARRHP YFYAPELLYY ANK**YNGVFQE CCQAEDKGAC
LLPKIETMRE KVLASSARQR LRCASIQKFG ERALKAWSVA RLSQKFPKAE
FVEVTKLVTD LTKVHKECCH GDLLECADDR ADLAKYICDN QDTISSKLKE
CCDKPLLEKS HCIAEVEKDA IPENLPPLTA DFAEDKDVCK NYQEAK**DAFL
GSFLYEYSRR HPEYAVSVLL R**LAKEYEATL EECCAKDDPH ACYSTVFDKL
K**HLVDEPQNL IK**QNCDQFEK **LGEYGFQNAL IVR**YTR**KVPQ VSTPTLVEVS
R**SLGKVGTRC CTKPESER**MP CTEDYLSLIL NRLCVLHEK**T PVSEKVTKCC
TESLVNR**RPC FSALTPDETY VPK**AFDEKLF TFHADICTLP DTEKQIKKQT
ALVELLKHKP KATEEQLKTV MENFVAFVDK CCAADDKEAC FAVEGPKLVV
STQTALA

Obrázek 13: Primární struktura BSA, pokrytí sekvence: 23 %

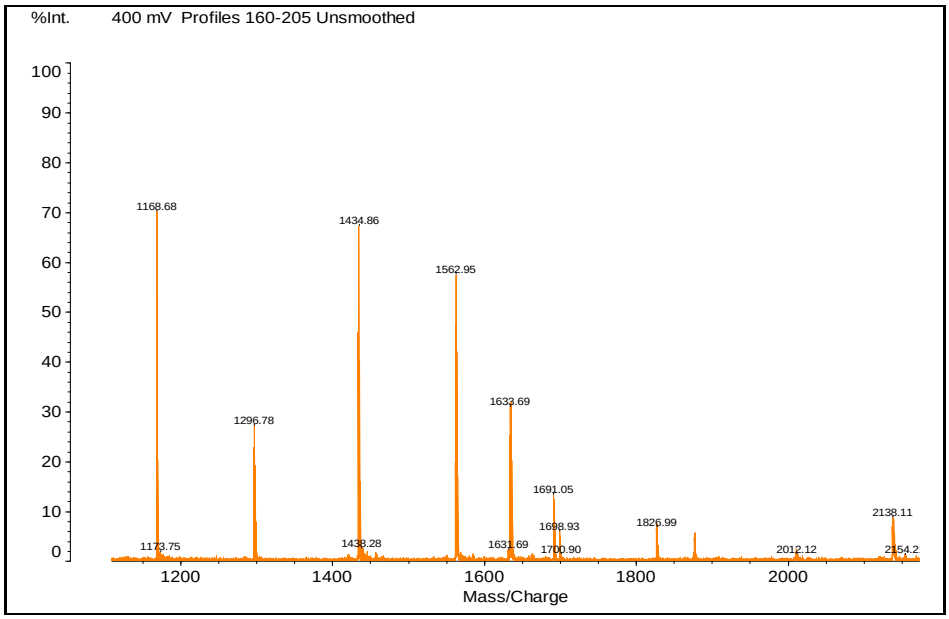
Cytochrom C

Cytochrom C (cyt C) patří mezi transportní proteiny, které jsou součástí mitochondriálního elektronového transportního řetězce zajišťujícího proces dýchání. Jeho relativní molekulová hmotnost je asi 12 400 (viz *Obrázek 14*). Štěpení cyt C o koncentraci 1 $mg \cdot ml^{-1}$ bylo prováděno trypsinem o koncentraci 0,1 $mg \cdot ml^{-1}$ za podmínek uvedených v kapitole 4. Naměřené MS spektrum získaných peptidů je na *Obrázku 15*.



Obrázek 14: MALDI - TOF spektrum intaktního cyt C

Pík označený číslem 1 odpovídá částici $[CYT\ C + H]^+$. Pík s číslem 2 je $[CYT\ C + 2H]^{2+}$ a pík s číslem 3 částice $[(2 \cdot CYT\ C) + H]^+$.



Obrázek 15: MALDI - TOF spektrum peptidových fragmentů cyt C

Na *obrázku 15* je zobrazeno spektrum rozštěpeného proteinu a hodnoty *m/z* signifikantních peptidů.

Tabulka 4: Souhrnné zobrazení odpovídajících peptidů pomocí programu MASCOT

umístění aminokyselin v řetězci	Mr_{exp}	Mr_{teor}	rozdíl Mr (ppm)	missed cleavage	sekvence
29 - 39	1168.6800	1167.6149	50	0	KTGPNLHGLFGRK
29 - 40	1296.7800	1295.7099	49	1	KTGPNLHGLFGRKT
90 - 101	1434.8600	1433.7878	45	2	KGEREDLIAYLKKA
10 - 23	1633.6900	1632.8116	-79	1	KIFVQKCAQCHTVEKG
41 - 56	1698.9300	1697.8009	72	1	KTGQAPGFSYTDANKNKG
40 - 56	1826.9900	1825.8959	48	2	RKTGQAPGFSYTDANKNKG
10 - 26	1875.8400	1874.9495	-62	2	KIFVQKCAQCHTVEKGGKH
57 - 74	2138.1100	2137.0401	29	1	KGITWGEETLMEYLENPKKY

.

Výsledek hledání v databázi je uveden v *Tabulce 4* V prvním sloupci tabulky je uveden začátek a konec peptidového fragmentu daný aminokyselinou v polypeptidovém řetězci. V druhém sloupci jsou naměřené relativní molekulové hmotnosti analyzovaných fragmentů, ve třetím sloupci jsou uvedeny teoretické hodnoty relativní hmotnosti peptidů. Ve čtvrtém sloupci je poměrný rozdíl těchto hmotností uvedený v ppm. Pátý sloupec udává počet míst, které mohly být trypsinem nerozštěpeny, a přesto zahrnuty do identifikačního algoritmu.

Na *obrázku 16* je uvedena celá sekvence cyt C , přičemž červeně vyznačené části patří mezi úspěšně identifikované fragmenty.

MGDVEKGKK**IFVQKCAQCHTVEKGGK**HK**TG PNLHGLFGRK TGQAPGFSYT DANKNKGITW GEETLMEYLE NPKK**YIPGTK MIFAGIKKK**G EREDLIAYLK K**ATNE

Obrázek 16: Primární struktura CYT C, pokrytí sekvence: 71 %

IV. DISKUZE A ZÁVĚR

Tabulka 5 : Souhrnné výsledky

protein	M_r	pokrytí sekvence (%)
myoglobin	16 700	70
BSA	66 000	23
cytochrom C	12 400	71

Podařilo se identifikovat modelové proteiny myoglobin, BSA a cytochrom C metodou peptidového mapování MALDI TOF MS. V Tabulce je pro jednotlivé proteiny uvedena relativní atomová hmotnost M_r a pokrytí sekvence, které udává aminokyselinové pokrytí celého proteinu uvedené v %.

Pro cytochrom C a myoglobin dosahovala hodnota SC okolo 70 %. Nižší pokrytí sekvence u BSA (23 %) je dáno vyšší molekulovou hmotností proteinu a výskytem majoritních fragmentů v oblasti MS spektra, kde již píky nejsou dostatečně rozlišeny. Umístění všech proteinů na prvním místě možných produktů identifikace potvrzuje, že všechny proteiny byly úspěšně identifikovány.

V. POUŽITÁ LITERATURA

[1] Yates, Jr; Speicher, S; Griffin, Pr; Hunkapiller, T.: Analytical Biochemistry, 214; 397 (1993).

[2] Vaňková, H.: Chem. Listy 93, 120 (1999).

[3] James, P; Quadroni, M; Carafoli, E; Gonnet, G.: Biochemical And Biophysical Research Communications, 195; 58 (1993).

[4] Carroll, J; Fearnley, IM; Shannon, RJ; Hirst, J; Walker, JE.: Molecular & Cellular Proteomics, 2; 117 (2003).

[5] Bell, AW; Ward, MA; Blackstock, WP; Freeman, HNM; Choudhary, JS; Lewis, AP; Chotai, D; Fazel, A; Gushue, JN; Paiement, J; Palcy, S; Chevet, E; Lafreniere-Roula, M; Solari, R; Thomas, DY; Rowley, A; Bergeron, JJM.: Journal Of Biological, 276; 5152 (2001).

[6] Lopez, MF; Kristal, BS; Chernokalskaya, E; Lazarev, A; Shestopalov, AI; Bogdanova, A; Robinson, M.: Electrophoresis, 21; 3427 (2000).

[7] Deng, L; Wang, C; Spencer, E; Yang, LY; Braun, A; You, JX; Slaughter, C; Pickart, C; Chen, ZJ.: CELL, 103; 351 (2000).

[8] V. C. Wasinger, S. J. Cordwell, A. Cerpa-Poljak, J. X. Yan, A. A. Gooley, M. R. Wilkins, M. W. Duncan, R. Harris, K. L. Williams, I. Humphery-Smith: Electrophoresis, 16; 1090 (1995).

[9] http://genetika.wz.cz/mapovani.htm (únor 2008).

[10] http://proteom.biomed.cas.cz/proteomics/proteomics.cs.php (březen 2007).

[11] http://www.otevrena-veda.cz/ov/users/Image/default/C1Kurzy/Chemie/28obsil.pdf (únor 2008).

[12] J. C. Venter et al.: Science, 291; 1304 (2001).

[13] Z. Vodrážka, J. Krechl: Bioorganická chemie, SNTL (1991).

[14] http://cs.wikipedia.org/wiki/Aminokyselina (březen 2007).

[15] http://projektalfa.ic.cz/amk.htm (březen 2007).

[16] http://cs.wikipedia.org/wiki/Proteiny (říjen 2007).

[17] http://chemicke-listy.vscht.cz/docs/full/2005_12_886-889.pdf (březen 2007).

[18] http://www.osel.cz/index.php?clanek=241 (březen 2007).

[19] http://biol.lf1.cuni.cz/ucebnice/proteomika.htm (březen 2007).

[20] Havliš J.: Vesmír 78, 448 (1999).

[21] http://koichi-tanaka.navajo.cz/ (březen 2007).

[22] http://biomikro.vscht.cz/maldiman/cz/img/theory/schema_maldi-tof-ms.gif (březen 2007).

[23] M. Yamashita, J. B. Fenn: Journal of Physical Chemistry 88; 4451 (1984).

[24] T. Štosová, J. Havliš, R. Lenobel, M. Šebela: Chem. Listy, 99; 896 (2005).

[25] J. Havliš, H. Thomas, M. Šebela, A. Shevchenko.: Anal. Chem., 75; 1300 (2003).

[26] Pappin, Djc; Hojrup, P; Bleasby, Aj.: Current, 3; 327 (1993).

[27] http://prospector.ucsf.edu/ (březen 2007).

[28] http://www.matrixscience.com/ (březen 2007).

[29] http://www.narrador.embl-heidelberg.de (duben 2007).

[30] Jungblut, PR; Bumann, D; Haas, G; Zimny-Arndt, U; Holland, P; Lamer, S; Siejak, F; Aebischer, A; Meyer, TF.: Molecular Microbiogy, 36; 710 (2000).

[31] Gevaert, K; Vandekerckhove, J.: Electrophoresis, 21; 1145 (2000).

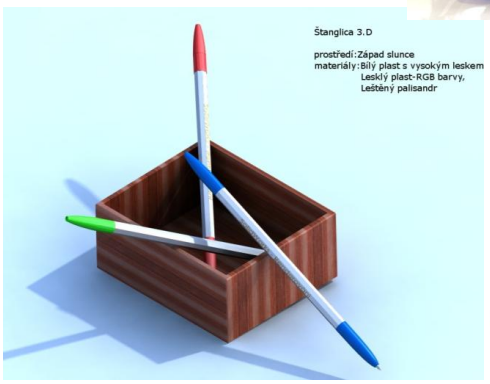
[32] Cottrell, Js.:Peptide Research, 115 (1994).

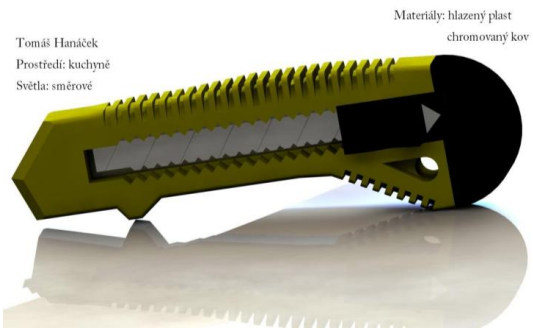
RENDEROVÁNÍ SKLENĚNÝCH PŘEDMĚTŮ VE 3D SCÉNĚ

Alena Vrbová , Ivo Zegzulka
Střední průmyslová škola Bruntál









Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

NEJLEPŠÍ ŠKOLA

MARTIN ŠVEJDA

Střední průmyslová škola strojní a elektrotechnická, Dukelská 13, České Budějovice

PROJEKT K PRAKTICKÉ MATURITNÍ ZKOUŠCE

SVĚTELNÝ TEXT NA ROTAČNÍ PLOŠE

Zadání:

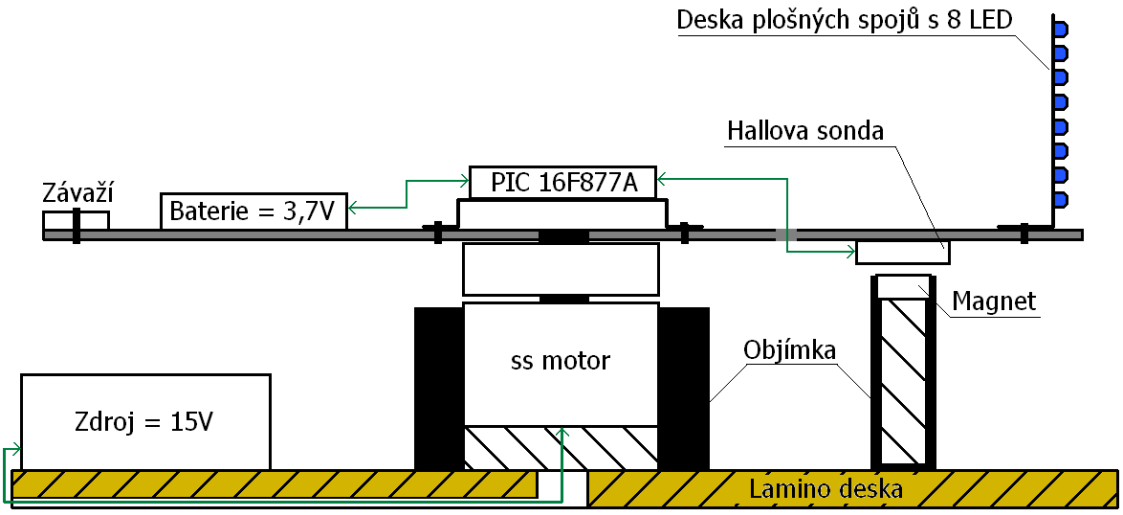
Pomocí 8 LED řízených mikroprocesorem PIC16F877A a motoru sestrojte rotační display, který bude vytvářet nápis, případně jiné grafické prvky v prostoru.

Úvod

Cílem mého projektu bylo sestavit rotační display, který by zobrazoval předem zadaný text, či jiné znaky.

Princip je následující: Řada osmi LED, posazená na vrtuli, rotuje v kruhu. Pomocí mikroprocesoru spínáme a vypínáme v určitých časových okamžicích (zde v každé milisekundě) jednotlivé LED, a podle toho, jaké LED necháme svítit, a jaké zhasneme, vytvoříme například písmeno, číslo či nějaký jiný znak. Nejdůležitější podmínkou, která musí být splněna, je, aby se text vykresloval pokaždé na stejné místo, toho můžeme docílit dvěma způsoby – lehčím a těžším. Lehčí způsob spočívá v tom, že budeme nastavovat ručně časovou konstantu tak, aby nám text stál na místě nebo mírně popojížděl. Nevýhodou tohoto řešení je, že jakákoliv změna otáček motoru vede k pojíždění textu, tedy jeho nečitelnosti. Těžší způsob spočívá ve snímání otáček motoru, které lze provést více způsoby. Já jsem si zvolil ten těžší způsob synchronizace a jako snímač jsem zvolil Hallovu sondu.(viz kapitola Hallova sonda)

Blokové schéma zapojení



POPIS POUŽITÉHO HARDWARU

- Hardwarová část se skládá z komutátorového motoru, původem z videopřehrávače, jehož stator je upevněný v objímce z PVC a následně přidělán k lamino desce.

- Na rotoru je uchycen duralový plátek v podobě vrtule. Na jeho prvním konci je kolmo k plátku přidělaná deska plošných spojů s 8 LED(viz obr.), diodami a 30 ohmovými rezistory.



- Na druhém konci je posazena baterie Lithium Ion 3,7V/650mAh z mobilního telefonu a závaží, sloužící k vyvážení vrtule.

- Přesně uprostřed je přišroubován modul s mikroprocesorem PIC16f877A ze stavebnice MLAB.

- Spod vrtule je upevněna Hallova sonda MH3SS2, která pomocí magnetu zapuštěném v objímce z polypropylenu a staticky upevněném na lamino desce, snímá každou otáčku vrtule.

- Celý rotační display je z bezpečnostních důvodů opevněn plexisklem.

HALLOVA SONDA MH3SS2

Princip Hallovy sondy spočívá v tom, že na povrchu krystalu je umístěný plátek opatřený proudovými vývody a odbočkami pro měření Hallova napětí. Tento element detekuje magnetický tok působící kolmo k povrchu čipu. Signál úměrný velikosti protékajícího proudu a magnetického toku je zpracován v obvodu integrovaném na čipu senzoru.

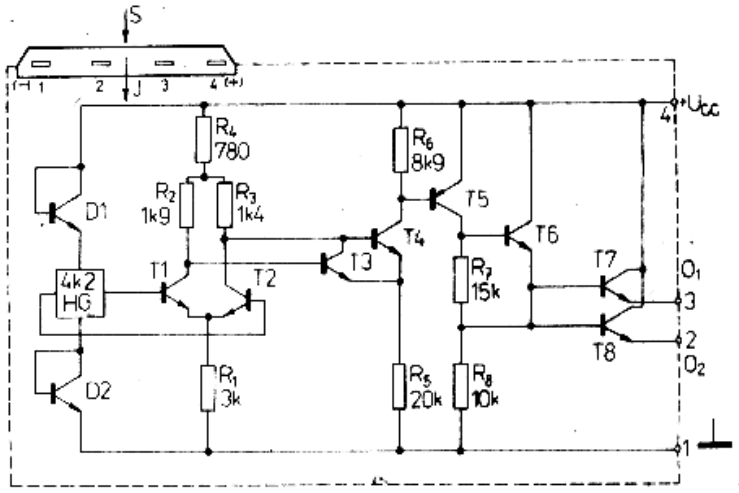
Hallova Sonda MH3SS2 má celkem 4 vývody. Nožičky 1 a 4 slouží k napájení a na nožičce 2 nebo 3 můžeme po umístění Hallovy sondy do magnetického pole naměřit proti zemi napětí. U těchto vývodů není ale na emitoru výstupního tranzistoru rezistor a tak se musí zapojit dodatečně.

Sonda potřebuje ke správné funkci 5V napájecího napětí. Při tomto napětí má výstup číslicový. Jelikož ale používám k napájení obvodu baterii 3,7V, pracuje sonda jako analogová. Pro můj účel to ale nemá význam. Sice Napětí naměřené na sondě při každém průchodu magnetickým polem permanentního magnetu nestačí na logickou 1, ale pomocí A/D převodníku obsaženého v procesoru PIC16F877A ho detekovat lze.

MEZNÍ HODNOTY:

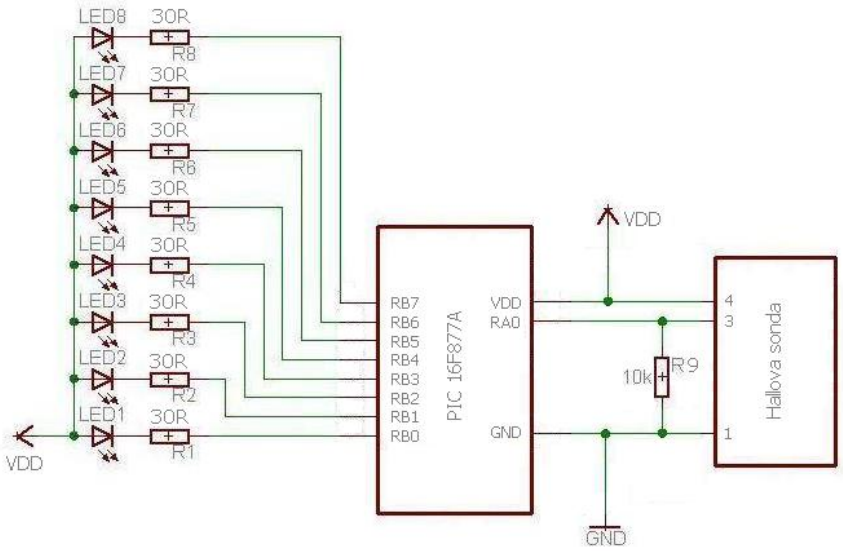
U_{CC}		$5 \pm 0{,}5$	V
$I_{2/1}$	max.	10	mA
$I_{3/1}$	max.	10	mA
$I_{2/1} + I_{3/1}$	max.	20	mA
ϑ_a	min.-max.	0 ... +55	°C
ϑ_{stg}	min.-max.	−55 ... +55	°C

VNITŘNÍ ZAPOJENÍ:



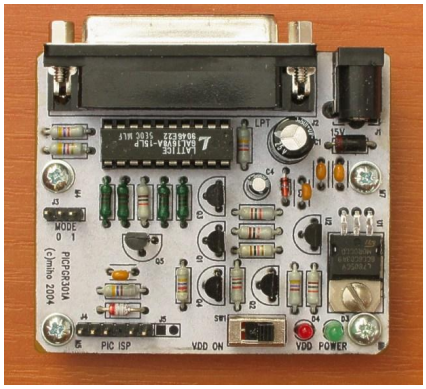
MH3S52

SCHÉMA PROPOJENÍ MIKROPROCESORU S 8 LED A HALLOVOU SONDOU



PROGRAMOVÁNÍ MIKROPROCESORU

Program jsem napsal v jazyce C a použil jsem k tomu program PIC-C Compiler. Do mikroprocesoru jsem ho nahrával pomocí programátoru PICPGR301A, což je malý vývojový programátor firmy MLAB pro programování procesorů PIC firmy MICROCHIP. Umožňuje mimo jiné programované zařízení spustit bez odpojování programátoru a může jej resetovat i napájet, což usnadňuje práci.



pak

PROGRAM

Program podstoupil dlouhou cestu vývoje, než se dobelhal k nynější podobě. Když jsem psal první program, neměl jsem ještě na vrtuli umístěnou Hallovu sondu a tak jsem musel softwarově synchronizovat časovou konstantu. Poté jsem ale obohatil rotační display o Hallovu sondu a snímal jsem každou otáčku motoru. Vždy když se sonda dostane do oblasti magnetického pole, naindukuje se na ní napětí, které mikroprocesor vyhodnotí a na displayi se rozblikají LED, které píšou zadaný text.

Nyní ale už k samotnému programu. Detekce napětí Hallovy sondy je provedena přes A/D převodník, který je definovaný na pin A0. Výstupy LED jsou umístěné na celou bránu B (RB0-RB7), tudíž nemusím pro každou LED psát jiný příkaz a mohu na bránu posílat bitovou informaci. Celý program se mi tím celkově zjednodušil. Jedno písmeno svítí na rotační ploše celkem 6ms; je do něho zahrnuta i mezera která trvá 1ms. Nadefinoval jsem si tedy pro každé písmeno konstantní *pole*, ve kterém je zahrnuto 6 osmibitových hodnot (Pro každou milisekundu). Pole pak vypisuji pomocí cyklu *for*.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Zaměření reálného stavu parcel v katastrálním území Bořislav

Adam Bárta

Středoškolská odborná Střední průmyslová škola Duchcov
Kubicových 2,Ústecký kraj
činnost 2007/2008

Obor 05 – geologie, geografie

Poděkování

Děkuji vedoucímu práce Ing. Jaroslavu Bendovi za odborné rady a konzultace. Rovněž děkuji Ing. Ivě Kloudové za pomoc při celkovém zpracovávání práce a Ing. Jiřímu Gärtnerovi za podpůrné materiály a informace.

Zásady pro vypracování:

a) proveďte stručné shrnutí zadané lokality

b) prohlédněte si zadaný terén a rozhodněte jaké metody měření použijete

c) zaměřte zadanou lokalitu pomocí polygonových pořadů a polární metody

Obsah:

1. Úvod 8
2. Geodetické a mapovací práce 9
   2.1. Historie katastrálních map 1 : 2 880 9
   2.2. Průběh prací v etapách 12
      2.2.1. Rekognoskace terénu a bližší prohlídka zadané lokality 12
      2.2.2. Obstarání potřebných podkladů 13
      2.2.3. Rozvržení prací 14
      2.2.4. Vybudování pevných měřických bodů, jejich stabilizace 15
      2.2.5. Zaměření pevných měřických bodů 15
      2.2.6. Podrobné měření, volba metod a přesnost měření 16
      2.2.7. Výpočetní práce 17
      2.2.8. Zobrazovací práce 17
      2.2.9. Porovnání zaměřených hranic s kopií katastrální mapy 1 : 2 880 18
      2.2.10. Porovnání zaměřené situace s leteckým snímkem Bořislavi 18
3. Měřické vybavení 19
4. Výpočetní programy 20
5. Závěrečné shrnutí 22

Přílohová část:

Příloha č. 1 ..... Žádost obce Bořislav

Příloha č. 2 ..... Stabilizace zhušťovacích bodů

Příloha č. 3 a – 3 c .....Geodetické údaje trigonometrických bodů
Příloha č. 4 a – 4 b .....Geodetické údaje zhušťovacích bodů
Příloha č. 5 a – 5 b .....Geodetické údaje PBPP
Příloha č. 6 .....Náčrt polygonových pořadů
Příloha č. 7 a – 7 i .....Měřické náčrty
Příloha č. 8 a – 8 k ..... Zápisníky podrobného měření
Příloha č. 9 a – 9 d .....Výpočetní protokol polygonových pořadů
Příloha č. 10 .....Výpočetní protokol podrobných bodů 38-56
Příloha č. 11 a – 11d .....Seznam souřadnic
Příloha č. 12 a – 12 b .....Polohopisný náčrt
Příloha č. 13 .....Polohopisný originál
Příloha č. 14 a – 14 b .....Porovnání s katastrální mapou 1 : 2 880
Příloha č. 15 .....Porovnání leteckého snímku s katastrální mapou

Seznam obrázků:

Obr. č. 1 .....Souřadný systém Gusterbergský a Svatoštěpánský
Obr. č. 2 .....Poloha Bořislavi
Obr. č. 3 .....Stabilizace měřického stanoviska
Obr. č. 4 .....Topcon GTS-210

Použité zdroje:

www.amapy.cz
www.zememeric.cz
www.cuzk.cz
www.gepro.cz

Anotace

Práce reaguje na potřeby obce Bořislav, která požádala o spolupráci s cílem získat mapové podklady pro posouzení stávajících vlastnických hranic v chatové oblasti s původní katastrální mapou 1 : 2 880. Výsledek by měl sloužit k vyjasnění sporných hranic v katastru nemovitostí ČR.

Záměr byl splněn a v rámci práce byla pro potřeby obecní správy vyhotovena reálná situace v měřítku 1 : 1 000, provedlo se porovnání se starší mapou 1 : 2 880 a vznikla i montáž zaměřených vlastnických hranic a leteckého snímku.

1. Úvod

Podnětem ke zpracování naší práce byl požadavek obce Bořislav na zaměření reálného stavu jednotlivých parcel staré Bořislavi a porovnání zjištěných údajů s katastrálními mapami obce. Katastrální mapy Bořislavi jsou v zastaralém sáhovém měřítku 1: 2 880. Žádost obce – viz příloha č. 1.

Samotná práce se nám zdála velice zajímavá už tím, že nám v rámci jedné akce dovolila komplexně shrnout a ověřit si praktické i teoretické dovednosti, které jsme v průběhu studia načerpali. Lákalo nás využít ji jako téma vhodné pro SOČ, a to i zejména proto, že nabízela možné srovnání historického vývoje map. Neboť, jak již bylo zmíněno, jako podklad jsme měli k dispozici katastrální mapu v zastaralém měřítku.

První část práce z oddílu, který jsme nazvali jako geodetické a mapovací práce, jsme se proto rozhodli věnovat alespoň stručnému přehledu o vzniku a historii katastrálních map. Další části v oddílu jsou pak zaměřeny na samotný průběh geodetických prací – od rekognoskace terénu až po konečné zobrazovací kancelářské práce. Vzhledem k rozsahu akce jsme si práce vhodně rozdělili do několika etap tak, jak by měly postupně navazovat. Tyto etapy, jejich řešení a porovnání výsledků jsou zahrnuty v části 2.2.

Části 3 a 4 popisují naše možnosti z hlediska technického vybavení, tj. přístroje, s kterými jsme mohli pracovat a geodetický software, který jsme využili pro zpracování dat.

Závěr práce zmiňuje vcelku velmi uspokojivý výsledek našeho snažení, kdy jsme dosáhli pochvaly a ocenění jak ze strany pana starosty obce Bořislav, tak ze strany školy.

Přílohová část obsahuje měřický materiál, se kterým jsme se v průběhu akce pracovali (tj.zápisníky, měřické náčrty, geodetické údaje bodů, apod.). Je zde obsažen i finální výstup naší práce, což je mapa části obce se zachyceným stavem parcel v měřítku 1: 1000.

2. Geodetické a mapovací práce

2.1 Historie katastrálních map 1: 2 880

Rakouský stabilní katastr

Situace v Evropě koncem 18. století byla poznamenána neurovnanými poměry a dopadem napoleonských válek.Vznikala potřeba zvýšeného krytí státních výdajů. Pocit nedokonalé funkce systému výběru pozemkové daně vedla císaře Františka I. k myšlence požádat dvorního radu o úpravu pozemkové daně.

Dvorská komise připravila roku 1816 návrh, který říkal: *„pozemková daň rozdělená a vyměřená má být podle plochy a čistého výnosu“* **.** Po rozpracování a úpravách byl podkladem k vydání patentu císaře Františka I. Patent byl po zkušebním měření v Dolním Rakousku vydán dne 23. 12. 1817.

Stručná forma jeho obsahu:

- katastr bude obsahovat *všechny pozemky* hospodářsky obdělávané i neobdělávané, bez ohledu na panskou nebo poddanskou půdu
- pozemky budou geometricky zaměřeny, zobrazeny, sepsány a popsány
- pozemky budou rozlišeny podle druhu (kultury) a užívání
- provedeno bude vytřídění pozemků do jakostních tříd (bonity), stanoví se čistý výnos z pozemků jako základ pro stanovení výše pozemkové daně.

Císařský patent ve své podstatě představoval souhrn všech zásad nového roztřídění pozemkové daně a současně všeobecné zásady tvorby katastrálních map. Podle tvůrců císařského patentu měl *katastr* představovat *stálý a dokonalý* seznam všech pozemků podrobených dani, s udáním jejich velikosti, plochy a čistého výnosu. Právě pro jeho stálost a dokonalost a domněnku, že bude moci sloužit navždy svému účelu, byl již ve své době nazýván *stabilní katastr***.**

Jejich domněnka se skoro potvrdila. Vždyť ještě dnes používáme v katastru nemovitostí téměř 70 % map, vycházejících z obsahu stabilního katastru a jsou i ve stejném měřítku. Prodělaly pouze svůj vývoj vedoucí mnohdy ke snížení kvality a vypovídací schopnosti jejich původního obsahu.

Aby mapové dílo bylo vyhotoveno ve všech částech země stejným způsobem byla vydána v roce 1824 *měřická instrukce***.** Mapy stabilního katastru byly převážně mapy původní, vyhotovené metodou měřického stolu (grafické mapy) a později, začátkem 20. století, byla velmi malá část map vyhotovena tzv.metodou *trigonometricko-polygonální*, tedy metodou číselnou.

Originální mapy sloužily zpočátku obcím jako evidenční mapy. Tam byly zaznamenávány změny až do doby, kdy byly nahrazeny otiskem originálních map.

Zobrazení:

Snaha o co nejjednodušší způsob zobrazení výsledků měření na mapových listech vedla k použití Cassini-Soldnerova zobrazení, které je charakterizováno jako transverzální válcové zobrazení ekvidistantní v kartografických polednících.

Území dnešní ČR je pokryto dvěma souřadnicovými soustavami, konkrétně pro Čechy soustavou Gusterbergskou a pro Moravu soustavou Svatoštěpánskou. Jejich názvy jsou odvozeny od jmen trigonometrických bodů, do nichž byl vložen počátek souřadnicových systémů.Trigonometrický bod *Gusterberg* leží v Horním Rakousku a trigonometrický bod *Svatý Štěpán* je jednou z věží kostela svatého Štěpána ve Vídni (viz. obr. č.1).

Geodetické základy:

Geodetickým základem pro mapování ve stabilním katastru se stala trigonometrická síť, vybudovaná postupně pro celou rakouskou monarchii a později i pro uherskou část. Na triangulačních pracích se podíleli výhradně vojenští geodeti a poté také byla svěřena činnost civilním úředníkům.

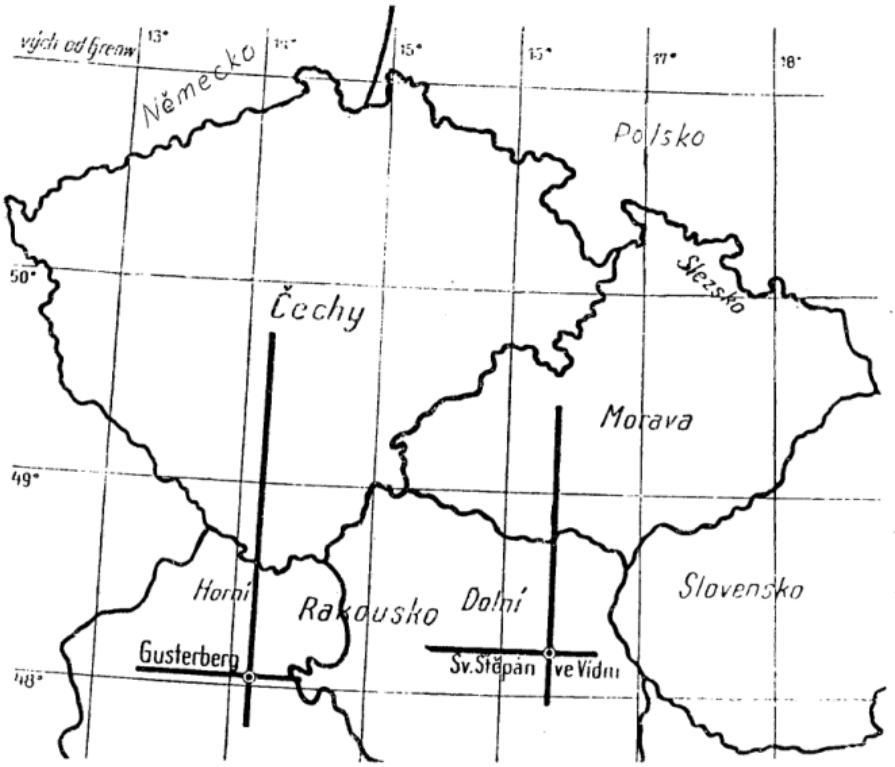
Celkový počet bodů trigonometrické sítě byl uváděn asi 12 590 a prakticky u všech se určovaly jak polohové souřadnice, tak i nadmořské výšky od Jaderského moře. Síť byla připojena na 4 přímo měřené základny, u nichž byla provedena astronomická orientace a určení zeměpisných azimutů.

Vlastní síť se budovala od I. do III. řádu přímým měřením na trigonometrických bodech, kdy délky stran se pohybovaly u I. řádu od 15 do 30 km, u II. řádu od 9 do 15 km a u III. řádu asi od 4 do 9 km.

Souřadnicový systém:

V každém souřadnicovém systému Cassini-Soldnerova zobrazení bylo území rozděleno rovnoběžkami s osami X a Y ve vzdálenostech jedné rakouské míle (1 rakouská míle byla 4 000 sáhů, tj. asi 7 585,94 m).

Pro laika neobvyklé měřítko sekčního listu 1 : 2 880 je odvozeno z tehdy užívané sáhové míry při požadavku, aby se výměra jednoho dolnorakouského jitra (čtverec o straně 40 sáhů), zobrazila na mapě jako jeden čtverečný palec. Velikost rámu sekčního listu je 25 x 20 palců, v metrické míře pak 65,85 x 52,68 cm.



obr. č.1: Souřadný systém Gusterbergský a Svatoštěpánský.

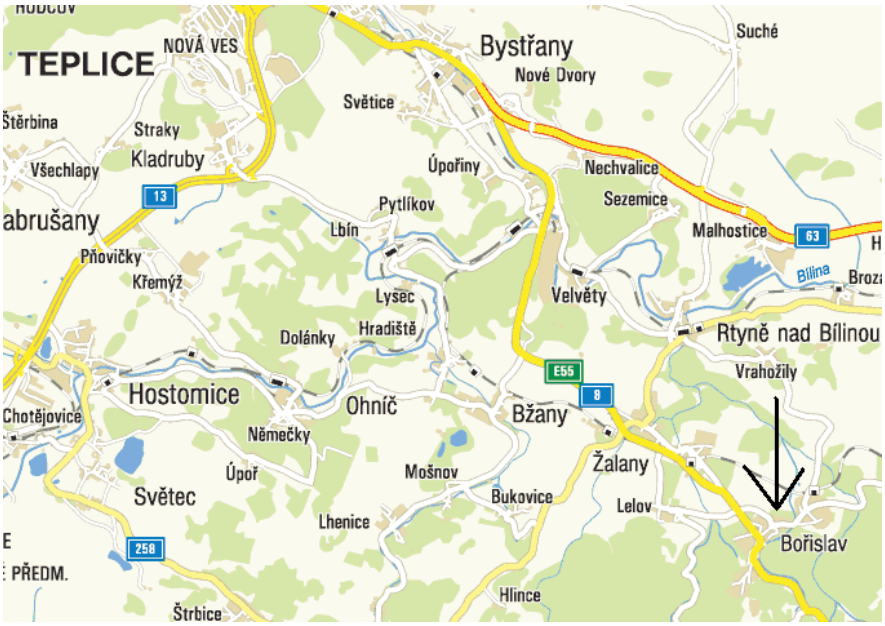
2.2 Průběh prací v etapách

2.2.1 Rekognoskace terénu a bližší prohlídka zadané lokality

Obec Bořislav se nachází v severních Čechách, v oblasti Českého středohoří, přibližně 12 km od Teplic po silnici E 55 směrem na Prahu (viz. obr. č.2).

Na základě prohlídky dané lokality jsme si ujasnili některé záležitosti, které ovlivnily obsah měření. Jednalo se zejména o možnost zaměřovat pouze hranice pozemků (ohraničené ploty), bez zaměřování stavebních objektů, a to z důvodů problematických vstupů na nemovitost (nutný souhlas vlastníka a jeho přítomnost) a z důvodů přírodních překážek (zarostlé zahrádky).

Dalším problémem byl rozsah prací, který přesahoval možnosti studentské práce. Proto bylo zadání zúženo na zaměřování pouze viditelné situace ze silnice procházející zadanou lokalitou v obci Bořislav.



obr. č.2: Poloha Bořislavi

2.2.2 Obstarání potřebných podkladů

Podklady, které bylo nutno použít pro tuto práci, by se daly rozdělit na tři části:

a) podklady od zadavatele, tj.obce Bořislav, která nám poskytla GIS program s uzemním plánem obce (více o GIS v části 4-výpočetní programy).

b) podklady z katastrálního úřadu, kde jsme si obstarali kopie katastrálních map v měřítku 1:1 000. Na katastrálním úřadě nám byly poskytnuty i údaje o měřických bodech 565, 566, 542. Všechny tyto body jsou stabilizované pomocí žulového hranolu s ochrannou tyčí. Ukázka stabilizace – viz. příloha č. 2

c) podklady získané přes internet-z internetových stránek Českého úřadu zeměměřického (www.cuzk.cz), kde jsme si vyhledali další údaje o nejbližších trigonometrických bodech.

Český úřad zeměměřický a katastrální, vede databázové soubory bodů bodového pole (dle zákona 359/92 Sb). Databáze bodových polí v současné době obsahují:

72 000	trigonometrických a zhušťovacích bodů
38 000	přidružených bodů
83 000	nivelačních bodů
400	tíhových bodů

Obsah databáze zhušťovacích bodů nám usnadnil práci. Pomocí jednoduchého grafického vyhledávání jsme na jmenovaném portálu pro okolí obce Bořislav vyhledali nejbližší trigonometrické a zhušťovací body.

Jednalo se o tři *trigonometrické* body:

1) trigonometrický bod č.15, v reálu byl ale zničen a jeho funkci převzal zajišťovací bod (č.bodu 15.1)
2) trigonometrický bod č. 25, byl na hradě Doubravka.Tento bod byl na vrchu věže turistické rozhledny. Při měření nebyl použitý, protože ze zadané lokality nebyl viditelný.

3) trigonometrický bod č. 22, byl věží meteorologické stanice na nejvyšším vrchu Českého středohoří, Milešovce. Tento bod také nebyl použitý ze stejného důvodu jako druhý bod na turistické rozhledně.

Geodetické údaje o trigonometrických bodech viz příloha č. 3 a – 3 c.

Dále byly nalezeny *zhušťovací* body:

1) zhušťovací bod č. 221.1 - pata kříže nad makovicí kostela Bořislav
2) zhušťovací bod č. 212 - komín elektrárny Ledvice.Tento bod nebyl použit při měření z důvodu jeho velké vzdálenosti od našeho zadaného území a také jeho špatné viditelnosti.

Další zhušťovací body v zadané lokalitě už nebyly nalezeny.
Geodetické údaje o zhušťovacích bodech viz příloha č. 4 a – 4 b.

Proto jsme se snažili na katastrálním úřadě zjistit v naší lokalitě co nejvíce bodů PBPP (podrobného bodového pole polohového). Jednalo se o body č. 565 a 566. Geodetické údaje o bodech PBPP viz příloha č. 5 a – 5 b.

Mezi další podklady patřily letecké snímky, které nám poskytala obec Bořislav.

2.2.3 Rozvržení prací

Po zjištění geodetických podkladů jsme se snažili zajistit odpovídající měřickou techniku a doplnit měřickou skupinu o pomocníky, figuranty. S přístrojovým vybavením nám vyšla maximálně vstříc naše škola a s měřením ochotně pomohli dva studenti - Jiří Stara, Jan Špička.

Důležitý byl i časový harmonogram, abychom zadané práce stihli včas. Vše jsme časově rozvrhli takto:

- Zaměření pevných měřických bodů v termínu od 14. 5. 2007 do 21. 5. 2007
- Měření podrobného mapování hranic pozemků v termínu od 22. 5. 2007do 25. 5. 2007
- Kancelářské práce v termínu od 27. 5. 2007 do 30. 5. 2007

Byli jsme rádi, že vše proběhlo bez problémů, i počasí nám přálo a termíny se podařilo dodržet.

2.2.4 Vybudování pevných měřických bodů, jejich stabilizace

Na základě předchozí rekognoskace terénu se navrhlo rozvržení bodového pole tak, aby bylo vidět na předměty měření. Ve vhodných místech jsme pak naše měřické body stabilizovali.

Stabilizace pevných bodů se mohou provádět pomocí hřebů, dřevěných kolíků a dalších pomůcek ke stabilizaci. Protože naše stanoviska měla být vybudována na silnici, rozhodli jsme se ke stabilizaci pomocí nástřelných hřebů, které byly podloženy kovovým víčkem od lahve. Stanoviska se označila výraznou červenou a bílou barvou. Stanoviska se ještě fotograficky zdokumentovala pro snadnější nalezení. Tato stabilizace byla pro naše měření dostačující a byla velmi levná. Stabilizace – viz. obr. č.3.



obr. č.3: Stabilizace měřického stanoviska

2.2.5 Zaměření pevných měřických bodů

K zaměření stanovisek se měl použít oboustranně orientovaný a oboustranně připojený polygonový pořad, ale vzhledem k malému počtu připojovacích bodů, se musel najít jiný způsob určení. Proto se přistoupilo na polygonový pořad vetknutý. Pro tento druh určení stanovisek jsme se rozhodli, protože zadané území se nacházelo přesně mezi body PBPP (podrobného bodového pole) a jedním zhušťovacím bodem, který se zaměřil pouze orientačně k upřesnění měření.

Aby se určila přesně stanoviska celého zadaného území z pevných bodů byly vedeny tři vetknuté polygonové pořady:

1) polygonový pořad č. I začínal na bodě 4001 s orientací na bod PBPP č. 565. Pořad pokračoval přes stabilizovaná stanoviska (č.bodů 4002, 4003, 4004, 4005, 4006) a

končil na bodě 4007 s orientací na bod PBPP č. 542 a ještě s orientací na zhušťovací bod č. 221.1 .

Z bodu 4001 byla zaměřena další podrobná stanoviska (č.bodů 5001, 5002).

Z bodu 4005 byl zaměřeno stanovisko 5008.

2) polygonový pořad č. II začínal na bodě č. 5008 s orientací na bod č. 4005 a končil na bodě č. 5009 s orientací na bod č.bodu 4006.
3) polygonový pořad č. III začínal na bodě č. 4008 s orientací na bod prvního polygonového pořadu č. 4005. Pokračoval přes stabilizovaná stanoviska (č.bodů 4009, 4010, 4011,4012, 4013) a končil opět na počátku prvního polygonového pořadu (č.bodu 4014) s orientací na bod PBPP (č.bodu 565).

Tímto postupem jsme obsáhli celé zadané území a následnými výpočty tak získali souřadnice stabilizovaných bodů v S-JTSK.

Náčrt polygonových viz příloha č. 6

2.2.6 Podrobné měření, volba metod a přesnost měření

Při podrobném mapování jsme mohli volit mezi metodou ortogonální nebo polární. Pro naši situaci se zdála lepší metoda polární, protože jsme mohli použít elektronický dálkoměr, který byl pro práci rychlejší a splňoval dostačující přesnost.

Hlavní výhodou polární metody je její rychlost a při použití el. dálkoměru také možnost vložení naměřených dat do výpočetního programu. Nevýhodou pro naše měření bylo bohužel to, že použitý přístroj TOPCON-210, který vlastní naše škola, nemá možnost ukládat data do paměti, jelikož nám chybí patřičný připojovací kabel k počítači.

Proto se musely všechny naměřené údaje přepisovat do zápisníku podrobného měření. Měřické náčrty viz přílohy č. 7 a -7 i, zápisníky podrobného měření viz příloha č. 8 a - 8 k.

Pro polární metodu platí podmínky pro zaměřování podrobných bodů, které jsou dané normou. Např. pro délky při měření polární metodou platí, že střední chyba délky je M_{xy} = 2cm. Dále se musí zohlednit redukce, teplota, tlak, oprava délek ze zkreslení. Pro měřené směry platí, že orientace musí být minimálně dvě a musí se provádět ve dvou polohách dalekohledu.

Pokud stojíme na stanovisku, u kterého známe souřadnice, stačí jen na jednu orientaci zaměřit, i vzdálenost se směry.

Pokud ale stojíme na bodě, u kterého souřadnice nejsou dány, na tzv. volném stanovisku, musí se měřit na oba dva, vzdálenost a směry. Může také nastat případ, že na bod, který chceme zaměřit, nebude vidět. V tomto případě můžeme využít polární kolmice. Tato kolmice může maximálně dosahovat 30 m.

Požadovaná přesnost pro naše vyhodnocení byla dána kódem kvality 3 (M_{xy} = 0,14 m). Tuto přesnost nám udávají kódy přesnosti pro podrobné mapování katastrální mapy v 1:1 000.

2.2.7 Výpočetní práce

Po zaměření celé lokality následovalo zpracování naměřených údajů. Výpočetní práce byly prováděny v geodetickém programu Groma.

Celkem bylo vypočteno 251 podrobných bodů bodového pole. Celé území bylo zaměřeno z 19 stanovisek. Po celou dobu výpočtu nebyla překročená přípustná odchylka povolená pro práci v katastru.

Výpočetní protokoly polygonových pořadů I – III viz příloha č. 9 a – 9 d.

Ukázka výpočetního protokolu pro podrobné body 38 – 56 viz příloha č. 10. Vzhledem k rozsahu nebyly ostatní protokoly pro podrobné body do přílohy začleněny.

Seznam souřadnic viz příloha č. 11 a – 11 d.

2.2.8 Zobrazovací práce

K zobrazovacím pracem se nemohl použít jen program Groma. Groma umí pouze výpočetní práce a vykreslí jen kontrolní kresbu, se kterou se nedá dále pracovat. Proto se musel najít jiný program, který dokáže splnit naše požadavky na grafické vykreslení. Pro tuto práci jsme se rozhodli použít program Kokeš. Je to výkonný editor rozsáhlých geografických dat, uložených souborově ve výkresech, nejrůznějších rastrových podkladech a geodetických údajích o bodech uložených v seznamech souřadnic. Umožňuje dále použít moduly pro zpracování měření z terénu, geodetické a konstrukční výpočty, nástroje na kontroly a topologické úpravy dat a má i další možnosti.

Hlavní výhodou programu Kokeš je jeho dokonalá grafická kresba.

Nevýhodou programu Kokeš je složitost a náročnost ovládání.

V programu Kokeš byly zobrazeny dvě hlavní části. První část obsahovala měřické náčrty. Druhá část obsahovala polohopisný originál.

Problémem při zpracování v programu Kokeš bylo dokončení kresby hranic. Tento problém nastal z důvodu pouze viditelných hranic pozemků bez hlubšího pronikání měřičů na nemovitost.

Více o programu Groma a Kokeš viz v části 4 - výpočetní programy.

Polohopisný náčrt a polohopisný originál viz přílohy č. 12, 13.

2.2.9 Porovnání zaměřených hranic s kopií katastrální mapy 1:2 880

Po vypočtení a zobrazení všech podrobných bodů, se začal porovnávat výsledek měření se starou katastrální mapou 1 : 2 880 z Rakouského stabilní katastru. Projevily se některé nedostatky, které jsou konkrétně popsány v závěrečném shrnutí. Předmětná katastrální mapa viz příloha č. 14 a - 14 b.

2.2.10 Porovnání zaměřené situace s leteckým snímkem Bořislavi

V závěru se provedlo porovnání s ortofoto mapou ČR. Jednalo se o pokusné porovnání leteckých snímků s výsledky měření.

K tomuto porovnání byl použit program Kokeš. V tomto programu se provedlo otevření ortofoto snímku dané lokality a tento podklad se použil pod hotové hranice parcel. Pomocí funkčních tlačítek v programu se začalo s hranicemi na ortofoto snímku pracovat. Hranice se natáčela tak, aby vystihla skutečnou polohu hranic pozemků. Jak se nám toto srovnání podařilo dokladuje příloha č. 15.

3. Měřické vybavení

Volba měřického vybavení byla velice pestrá.

K našemu měření mohl být použit přístroj BRT 006, který je k dispozici ve školních sbírkách a používal se pro podrobné mapování. Ale protože je tento stroj už velice starý, těžký a neskladný, rozhodli jsme se ho pro toto měření nepoužít.

Dále je v našich sbírkách velké množství teodolitů 010, 015, 020. Ani tyto přístroje jsme se nerozhodli pro měření použít pro jejich nepřesnosti a zdlouhavé měření. K měření by se muselo ještě používat pásmo a to by v některých případech nebylo možné.



obrázek č.5: Topcon GTS-210

Protože jsme chtěli mít měření přesné a rychlé, zapůjčili jsme si elektronický přístroj TOPCON modelové řady GTS-210, který je v současnosti nejlepším strojem na naší škole. Tato totální stanice má dosah až 1200 m a s použitím 3 hranolů dosah až 1600 m.

Přesnost délek je $m_s = 2\ mm + 2\ ppm \cdot sa$.

K měření jsme dále použili 50 m pásmo, stativ, signál a kancelářské potřeby.

Vznik totálních stanic začal s rozvojem elektroniky. Docházelo i k postupnému vývoji elektronických dálkoměrů.

Poznámka:

V sedmdesátých letech byl např. vyvinut Geodimeter model 10 – první elektronický dálkoměr nasazovací na teodolit. Od tohoto okamžiku bylo možné provádět úhlová i délková měření jedním přístrojem. Postupně se upustilo od dálkoměrů s dosahem až 60 km, elektronické dálkoměry se staly běžnými nástroji geodetů. Další integrací vznikly totální stanice, které jsou nyní vybavovány i automatickým cílením pomocí servo motorků a ovládáním přímo z měřeného bodu, přicházejí také družicové metody reprezentované především GPS atd.

4. Výpočetní programy

Groma:

GROMA je geodetický systém pracující v prostředí MS Windows. Systém je určen ke komplexnímu zpracování geodetických dat od surových údajů přenesených z totální stanice až po výsledné seznamy souřadnic, výpočetní protokoly a kontrolní kresbu.

Producent:
GEOLINE, spol. s r.o.
Štěrboholská 44
areál Tesly Karlín
102 00 Praha 10 - Hostivař

Kokeš

Systém KOKEŠ v sobě zahrnuje výkonný editor rozsáhlých geografických dat uložených souborově ve výkresech a nejrůznějších rastrových podkladech a geodetických údajů o bodech uložených v seznamech souřadnic. Dále obsahuje moduly pro zpracování měření z terénu, geodetické a konstrukční výpočty, nástroje na kontroly a topologické úpravy dat a další. Je vhodným nástrojem pro všechny běžné geodetické práce a pro tvorbu a údržbu mapových děl. Pro některé speciální úlohy jsou určeny jeho další nadstavby. Systém KOKEŠ je vybaven vlastním programovacím jazykem, což umožňuje doplnění jeho široké nabídky funkcí podle vlastních potřeb. Všechny operace a výpočty jsou protokolovány a odpovídají požadavkům katastrálních úřadů.

Systém KOKEŠ je datově zcela otevřený, datový formát je podrobně popsán a běžně je používán i textový formát dat. Systém pracuje s daty českého a slovenského katastru (VKM, VGI, VFK) a technické mapy (VTM). Navíc lze používat moduly pro export a import formátu DGN (MicroStation), DXF (AutoCAD) a SHP (ArcView). Data ve formátech DGN, DWG a SHP lze také přímo zobrazit (bez konverze), ale není možné je editovat.

Do systému KOKEŠ lze z totálních stanic importovat údaje z měření v terénu (zápisník). Systém obsahuje veškeré geodetické úlohy měřické i konstrukční. Vyrovnání sítí metodou nejmenších čtverců MNČ včetně výšek je vybaveno grafickým znázorněním výsledků a chyb a intuitivní podporou práce s vícenásobným měřením na stejném stanovisku. Nastavení korekcí a práce s kódem kvality je samozřejmostí. Systém umožňuje automatické zjištění třídy přesnosti mapování porovnáním skupiny identických bodů. K systému lze připojit nejen běžné tablety, digitizéry a planimetry, ale i GPS stanici.

Producent:
GEPRO spol. s r.o
Štefánikova 52
150 00 Praha 5

Map info

MapInfo představuje špičkový software pro vizualizaci a analýzu dat z geografické perspektivy. MapInfo umožňuje odkrytí v datech struktury a trendy, které by jinak zůstaly skryty za řádky a sloupci klasických databází.

S MapInfo můžeme provádět detailní a sofistikované analýzy dat využitím jejich prostorové složky, ať již se jedná o kompletní adresu, pouze PSČ nebo souřadnice naměřené GPS. Mapinfo slouží pro přesné a velmi přehledné podklady!

Použití MapInfo pokud potřebujeme především pro:

- Tvorbu velmi podrobných map pro lepší prezentace
- Odhalení přehledných podkladů v datech, které by byly jen obtížně rozpoznatelné v tabulkách a grafech
- Sofistikovanou analýzu dat
- Geograficky řešit rozmístění poboček, prodejen, skladů
- Řešit dopravní logistické úlohy
- Odhalit přehledně inženýrské sítě,hranice.

MapInfo je celosvětově považováno za jedno z nejvíce „User-friendly“ GIS řešení se snadným intuitivním ovládáním a zároveň velmi širokou funkcionalitou

5. Závěrečné shrnutí

I přes určité omezující faktory (rozsah prací, viditelnost, přístupnost parcel, problematické vlastnické vztahy), bylo vcelku úspěšně provedeno zaměření zadané lokality obce Bořislav. Zaměřená situace byla zobrazena v měřítku 1 : 1 000 v programovém softwaru Kokeš. Při porovnávání současného měření se zastaralou katastrální mapou 1 : 2 880 se plně projevily některé nedostatky:

A) nevhodné sáhové měřítko pro zobrazení detailní situace jednotlivých parcel.

B) aktualizace katastrální mapy, respektive zanedbávání včasné aktualizace průběhu vlastnických hranic.

C) neznatelnost na straně vlastníků při vědomých změnách hranic (např.posuny a změny plotů na vlastnické hranici). Nezajištění a nezaplacení včasné odborné práce geodetů tj. objednání zpracování geometrického plánu jako základního nástroje pro aktualizaci katastrální mapy.

D) nezkušenost geodetů (studentů) v problémech zaměřování vlastnických hranic.Tento problém lze odstranit pouze větší praxí v oboru.

Zaměřená situace se také pokusně porovnala s leteckým snímkem. Toto porovnání odpovídá úrovni znalostí a vybavení, které je k dispozici na SPŠ Duchcov. Jedná se o program Kokeš a jeho softwarové nástroje pro práci s rastry.

Porovnání bylo omezeno chybějícími vlícovacími body na snímku i v měření. Z dříve uvedených důvodu nebyly zaměřeny stavební objekty, které by byly právě vhodné jako vlícovací body pro orientaci kresby a polohového originálu. Výsledek má proto pouze orientační význam.

Při práci jsme byli vedeni maximální snahou o úspěšné dokončení akce v rámci zadaných limit. Jak bylo řečeno, určitá nezkušenost z hlediska měřické praxe se projevila, ale celkový výsledek postačil účelu, pro který jsme byli ze strany pana starosty obce Bořislav osloveni. Projevil celkovou spokojenost s naší prací a zdá se, že přichystal další měřickou „příležitost“ pro další studenty.

Pochvalu jsme obdrželi nejen od pana starosty, ale i ze strany školy za její dobrou reprezentaci.

Následuje: Přílohová část

Příloha č.1

Obec Bořislav
úřad : Bořislav 20, 415 01 Teplice

SPŠ Duchcov
Kubicových 2
Duchcov
419 01 Teplice

Věc: Zaměření vlastnických hranic v Obci Bořislav – chatová osada

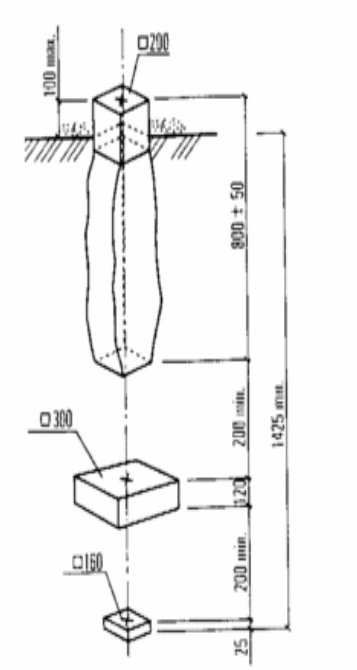
Žádáme o spolupráci při předběžném zjišťování současného stavu vlastnických hranic v Obci Bořislav – chatová osada. Práce má sloužit pro potřebu obecní správy Obce Bořislav. Jedná se o zaměření stávajícího stavu vlastnických hranic pozemků v zadané lokalitě chatové osady v Obci Bořislav. Jde o zaměření pouze vlastnických hranic viditelných z místní komunikace , není nutné zaměřovat vlastnické hranice skryté pohledu z ulice.
Cílem je zjistit současný stav reálných vlastnických hranic pro případné porovnání zjištěného stavu se zákresem v katastrální mapě v měřítku 1: 2880.
Výsledek této práce bude sloužit pro vyjasnění sporných hranic v katastru nemovitostí ČR. Uvítáme zobrazení situace ve vhodném měřítku. případně montáž zaměřených vlastnických hranic a leteckého snímku.

Děkujeme. a s pozdravem

Aleš Navara
starosta

OBECNÍ ÚŘAD BOŘISLAV

Stabilizace zhušťovacích bodů





Příloha č.3 a

GEODETICKÉ ÚDAJE
trigonometrického bodu

Kraj: Ústecký
Okres: Teplice
Obec: Bořislav

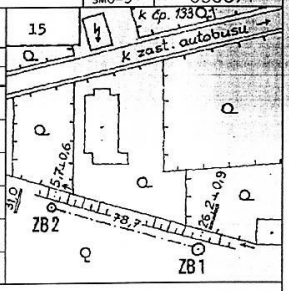
List č.: 1/1
Slav k: 1991

Vytvořena pro web 11.04.2007

TL	0614
ZM-50	02-32
SMO-5	030671

Číslo a název bodu	15	Bořislav, kostel	15

Bod	Druh	Y	X	Nadmořská výška Bpv	vztahuje se na
15	TB	centrum převedeno	na PB		
15.1	ZB1	769972.80	983980.86	371.14	hranol
15.2	ZB2	770048.78	983960.51	369.48	hranol



Orientace na body (ve stupních)

Číslo		Jižník	Délka strany	Číslo		Jižník	Délka strany
15.2	Bod vztažen k 15.1	104 59 29.0	78.656				

Místopisný popis: Věž kostela byla zničena, funkci TB převzal ZBI.

Bod	15		15.1		15.2			
Stab. údaje	0,00	zrušen	0,00	žula 20.20.75	0,00	žula 16.16.90	0,00	
			.87	žula 40.40.10	1.05	žula 30.30.10		
			1.04	sklo 16.16.03	1.20	sklo 16.16.03		
Označ. povrch. značky na boku:			△ v.		△ s. TP j.			
Ochranný znak: (druhy,rok)								
Kat. území: Pozn.zn.: Druh pěz:			Bořislav 722/4		Bořislav 688/3			

Druh a výška signal. stavby nebo nárys trvalého cíle:

Signalizace z roku:

Poznámky:

Zeměměřický úřad 2000

Příloha č.3 b

GEODETICKÉ ÚDAJE
trigonometrického bodu

Kraj: Ústecký
Okres: Teplice
Obec: Teplice

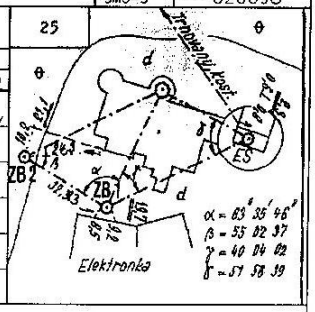
List č.: 1/2
Stav k: 1991

Vytvořeno pro web 11.04.2007

TL	0613
ZM-50	02-32
SMO-5	020698

Číslo a název bodu	25	Hrad Doubravka	25

Bod	Druh	Y	X	Nadmořská výška Bpv	vztahuje se na
25	TB	773744.93	976707.91	422.39	vrch rozhledny
25.2	ZB2	773790.55	976724.52	392.77	hranol
25.5	ZB1	773764.12	976743.06	395.23	čepová značka
25.6	ES1	773716.79	976724.60	414.07	hřebová značka



Orientace na body (ve stupních)

Číslo		Jižník	Délka strany	Číslo		Jižník	Délka strany
25.2		69 59 46.0	48.553		25.5–25.2		32.285
25.5		28 38 09.0	40.044		25.5–25.6		50.798
25.6		300 40 21.0	32.717	29	Orientace z 25.6	311 10 14.0	1198.613

Místopisný popis: Bod je vrch věže turistické rozhledny na vrchu Doubravka jihovýchodně od Teplic.

Bod	25		25.2		25.5		25.6	
Stab. údaje	0,00	vrch věže tur.rozhled.	0,00	žula 16.16.92	0,00	2 mos.čepy zákl. 0.700m	0,00	hřebová niv. značka
			1.03	žula 30.30.10		ram.1.390m 0.3m nad		
			1.31	sklo 16.16.03		terénem		
Označ. povrch. znaky na bodu:								
Ochranný znak: (druhyok)								
Kat. území: Pořadí v druh. pozn.	Teplice-Trnovany 2264/1		Teplice-Trnovany		Teplice-Trnovany 2264/1		Teplice-Trnovany 2264/3	

Druh a výška signal. stavby nebo nárys trvalého cíle:

Signalizace z roku: .....

Poznámky:

Zeměměřický úřad 2000

Příloha č.3 c

GEODETICKÉ ÚDAJE
trigonometrického bodu

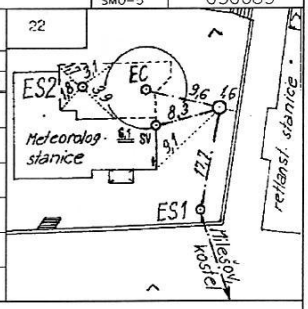
Kraj: Ústecký
Okres: Litoměřice
Obec: Velemín
List č.: 1/2
Stav k: 1991

Vytvořeno pro web 11.04.2007

TL	0614
ZM-50	02-34
SMO-5	030683

Číslo a název bodu	22	Milešovka

Bod	Druh	Y	X	Nadmořská výška Bpv	Nadmořská výška vztahuje se na
22	TB	770096.18	986673.41	836.56	hranol
22.3	EC1	770103.97	986667.73	859.07	vrch tyče
22.4	ES1	770106.83	986687.49	835.90	kónus
22.6	ES2	770108.62	986665.45	847.19	šroub



Orientace na body (ve stupních)

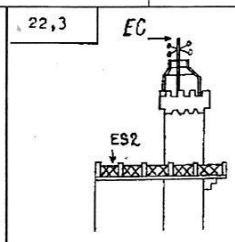
Číslo		Jižník	Délka strany	Číslo		Jižník	Délka strany
22.3		126 06 06.0	9.640				
22.4		37 06 20.4	17.653				
22.6		122 36 50.5	14.769				

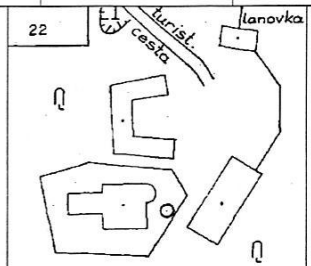
Slovní popis: Bod je na vrcholu nejvyššího kopce Českého středohoří, vedle budovy meteorologické stanice. Je totožný s bodem AGS 20. Body 22,1 a 22,2 zrušeny. Na bodu je zřízeno chráněné území označené výstražnou tabulí se státním znakem sev. od bodu.
* Centrum má další podz.značky.

Bod	22		22.3		22.4		22.6	
Slab. údaje	0,00	žula 29.29.90	0,00	tyč anemometru	0,00	mos. kónus ve zděném	0,00	mos.šroub v cihlovém
	.95	žula 60.60.09				pilíři		sloupu na ochozu věže
	1.10	sklo 16.16.03						
Označ. povrch. značky na bodu:	△ s. TP j.							
Ochranný znak: (druh,rok)								
Kat. území: Pořad.č. druh pozz	Milešov u Lovosic 657/2		Milešov u Lovosic		Milešov u Lovosic		Milešov u Lovosic	

Druh a výška signal. stavby nebo nárys trvalého cíle.

Signalizace z roku:





Poznámky:

Zeměměřický úřad 2000

Příloha č. 4 a

GEODETICKÉ ÚDAJE

zhušťovacího bodu

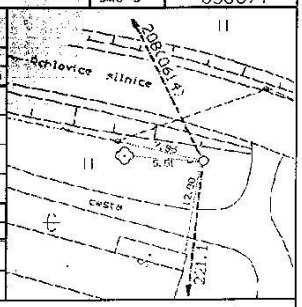
Kraj: Ústecký
Okres: Teplice
Obec: Bořislav
List č.: 1/1
Stav k: 1997

Vytvořeno pro web 11.04.2007

TL	0609
ZM-50	02-32
SMO-5	030671

Číslo a název bodu	221	Bořislav

Bod	Druh	Y	X	Nadmořská výška Bpv	vztahuje se na
221	ZHB	769776.50	983648.37	niv. 324.12	hranol
221.1	EC1	769825.30	983938.40	394.22	pata kříže



Orientace na body (v grádech) :

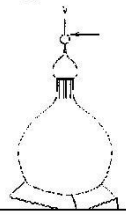
Bod číslo :	Jižník	Délka strany	Bod číslo :	Jižník	Délka strany
221.1	10.6123	294.110			
70614/208	174.48486	2807.149			

Místopisný popis : centr - S od kostela Bořislav, u vjezdu do prasečince. EC - pata kříže nad makovicí kostela Bořislav.

Bod určen : 221 GPS,

Bod	221		221.1					
Číslo údaje	0.00	žula 15x15x71	0.00	pata kříže věž kostela	0.00		0.00	
	.91	žula 30x30x10						
Ochranný znak (druh,rok)	OT-1997							
Katastrální území	Bořislav 656/1		Bořislav st.1					

EC



Bod		221	221.1		
Organizace,rok	Zřízen	1997 KÚ Liberec	1952 GTÚ Praha		
	Určení YX	1997	1952		
	Určení výšky	1997	1952		
	(Pře)Stabilizace	1997			
Rok	Údržba	1997			
	Obnova				

Poznámka : centr - určen GPS, EC (pův. 0809 - 40) převzat bez ověření

GEODETICKE UDAJE O ZhB

Kraj : Ústecký
Okres: Teplice
Obec : Ledvice

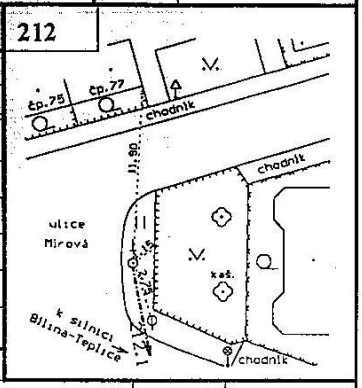
Pořadí č. 1
Platné od 1.1.2004 do

Příloha č. 4b
Příloha č. 4 b

TL	0619
ZM 50	02-32
SMO-5	030520

Číslo a název bodu: **212** **Ledvice východ** **212**

Bod	Druh	Y	X	Nadmořská výška Bpv	vztahuje se na
212	ZhB	780 569.13	981 794.80	211.86	hranol
212.1	EC	780 406.16	982 727.44		není určena



Orientace na body (v grádech)

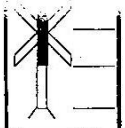
Číslo	Název	Jižník	Délka strany	Číslo	Název	Jižník	Délka strany
212.1 (0619)	EC - /Ledvice komín/	388.9869	946.77				

Místopisný popis:

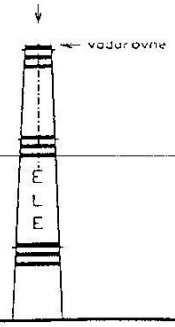
ZhB 212 se nachází na začátku ul. Mírová na V okraji Ledvic, u parčíku. EC 212.1 - bodem je komín elektrárny Ledvice, který se nachází v areálu závodu.

Bod	212		212.1			
Stab. údaje	0,00	žula 16x16x65	0,00	osa komínu	0,00	
	0,91	žula 30x30x10				
Ochran. znak: (ulič. rok, vzdál.)	OT-03-0.75-S					
Kat. území: Parc. č. : Druh poz. :	Ledvice 626/1 ostatní plocha		Ledvice 189/81 zastavěná plocha			
Vlastník: (uživatel) :	Obec Ledvice, Mírová č.p.42, PSČ 417 72		ČEZ a.s., Duhová 1444/2, Praha 4, 140 53			

Druh a výška sign. stavby nebo nárys trvalého cíle:



212.1



Signalizace z roku:

Poznámky:

ZhB 212 je zaměřen metodou GPS. Původní EC 212.1 a přeměřen metodou protínání.

GEODETICKÉ ÚDAJE O BODU PBPP

Údaje katastru nemovitostí

Platnost k: 09.05.2007 12:53

Kat. území : **608254 Bořislav**
Obec : **567469 Bořislav**
Okres : **CZ0426 Teplice**

Příloha č. 5 a

Verze : **1**
Platnost od : **01.04.2001**

Bod	Bod zřídil (jméno, rok)		
565		y	**770184.82**
		x	**984575.80**

SMO-5 : **LITOMĚŘICE 8-2**

Orientace jižník na bod: d ' ; g d cc

nadm. výška (Bpv)

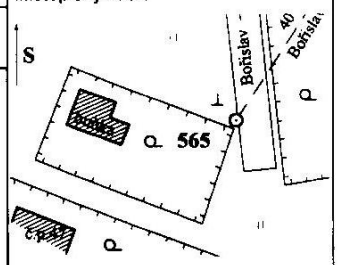
Míst.p., Stab., Zp.určení, Pozn. .

Bodem je žulový hranol s OT na okraji chatové osady nad Bořislaví.
Bodem je žulový hranol s OT.
rajonem
Intergeos s. r. o. Chomutov

WSG souřadnice metodou GPS: Fí, Lambda, Výška

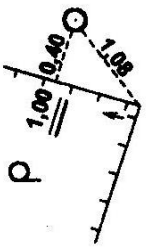
přesnost 3

Místopisný náčrt



Poznámka

Nárys nebo detail



Vyhotovil: KATASTR\LISKAM Katastrální úřad pro Ústecký kraj, Katastrální pracoviště Teplice 09.05.2007 12:5

GEODETICKÉ ÚDAJE O BODU PBPP

Údaje katastru nemovitostí

Platnost k: 09.05.2007 12:46

Příloha č. 5 b

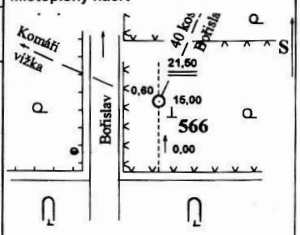
Kat. území : **608254 Bořislav**
Obec : **567469 Bořislav**
Okres : **CZ0426 Teplice**

Verze : **1**
Platnost od : **01.04.2001**

Bod	Bod zřídil (jméno, rok)		
566		y	**769991.30**
		x	**984448.14**

SMO-5
LITOMĚŘICE 7-2

Místopisný náčrt



Orientace jižník na bod: g c cc

nadm. výška (Bpv)

Míst.p., Stab., Zp.určení, Pozn. :
Bodem je žulový hranol s OT v zahradě pod lesem nad Bořislaví.
Bodem je žulový hranol s OT.
GPS
Intergeos s. r. o. Chomutov 1996

WSG souřadnice metodou GPS
Fí
Lambda
Výška

přesnost **3**

Poznámka

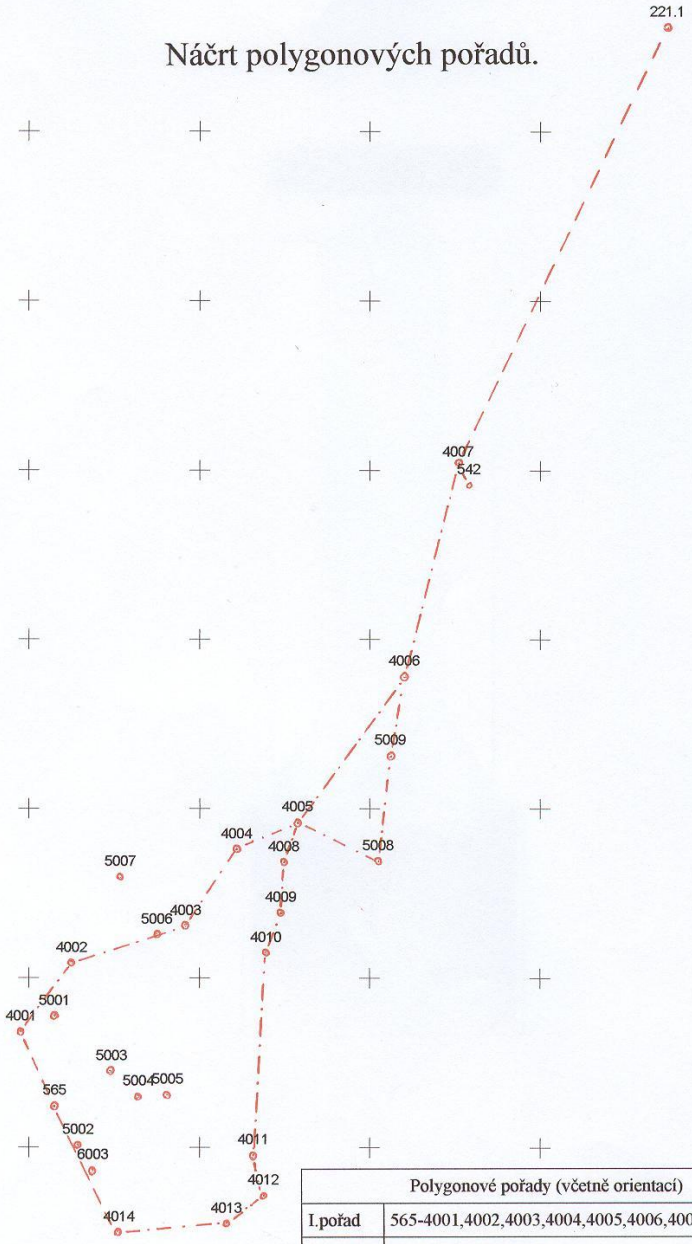
Vyhotovil: KATASTR\LISKAM

Katastrální úřad pro Ústecký kraj, Katastrální pracoviště Teplice

09.05.2007 12:-

Příloha č. 6

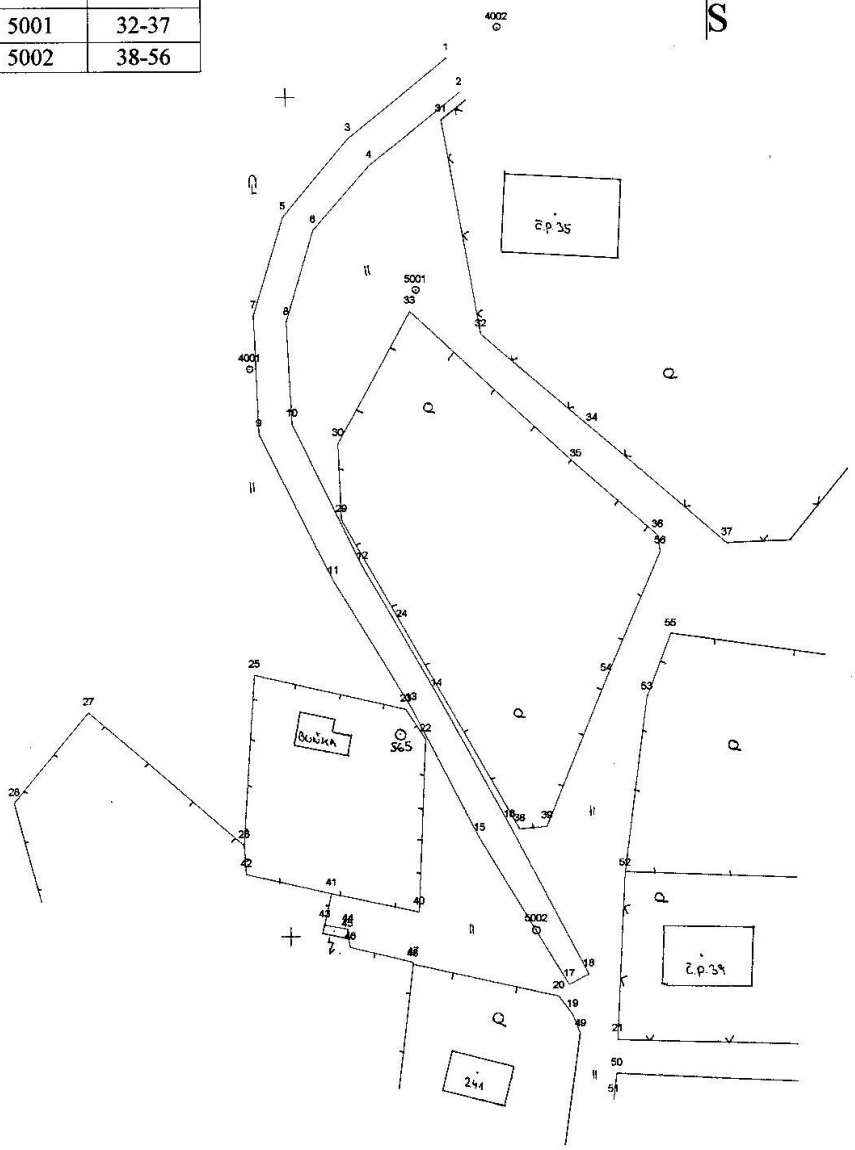
Náčrt polygonových pořadů.



Polygonové pořady (včetně orientací)	
I.pořad	565-4001,4002,4003,4004,4005,4006,4007-542-221.1
II.pořad	4005-5008,5009-4006
III.pořad	4005-4008,4009,4010,4011,4012,4013,4014-565

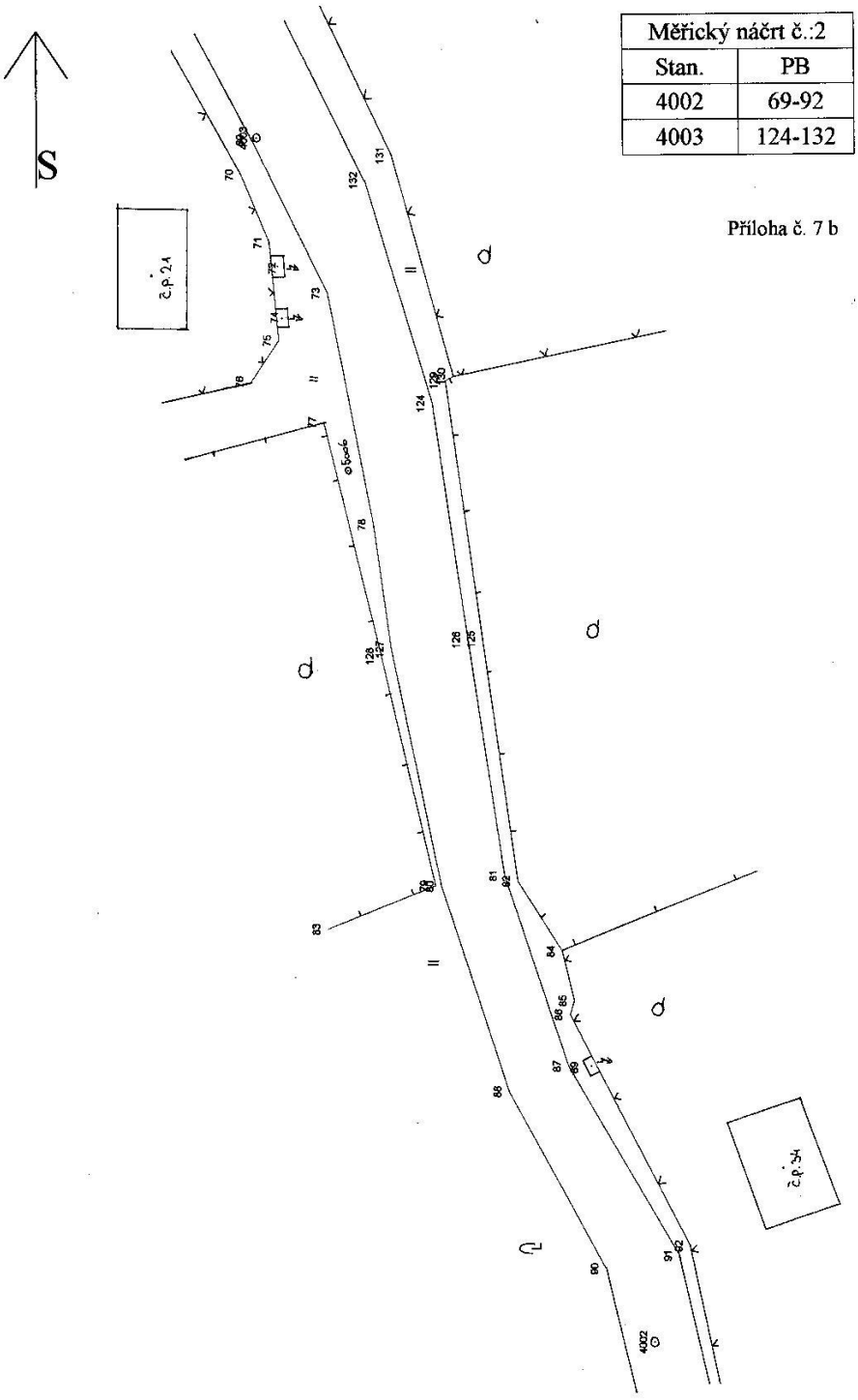
Příloha č. 7 a

Měřický náčrt č.:1	
Stan.	PB
4001	1-31
5001	32-37
5002	38-56



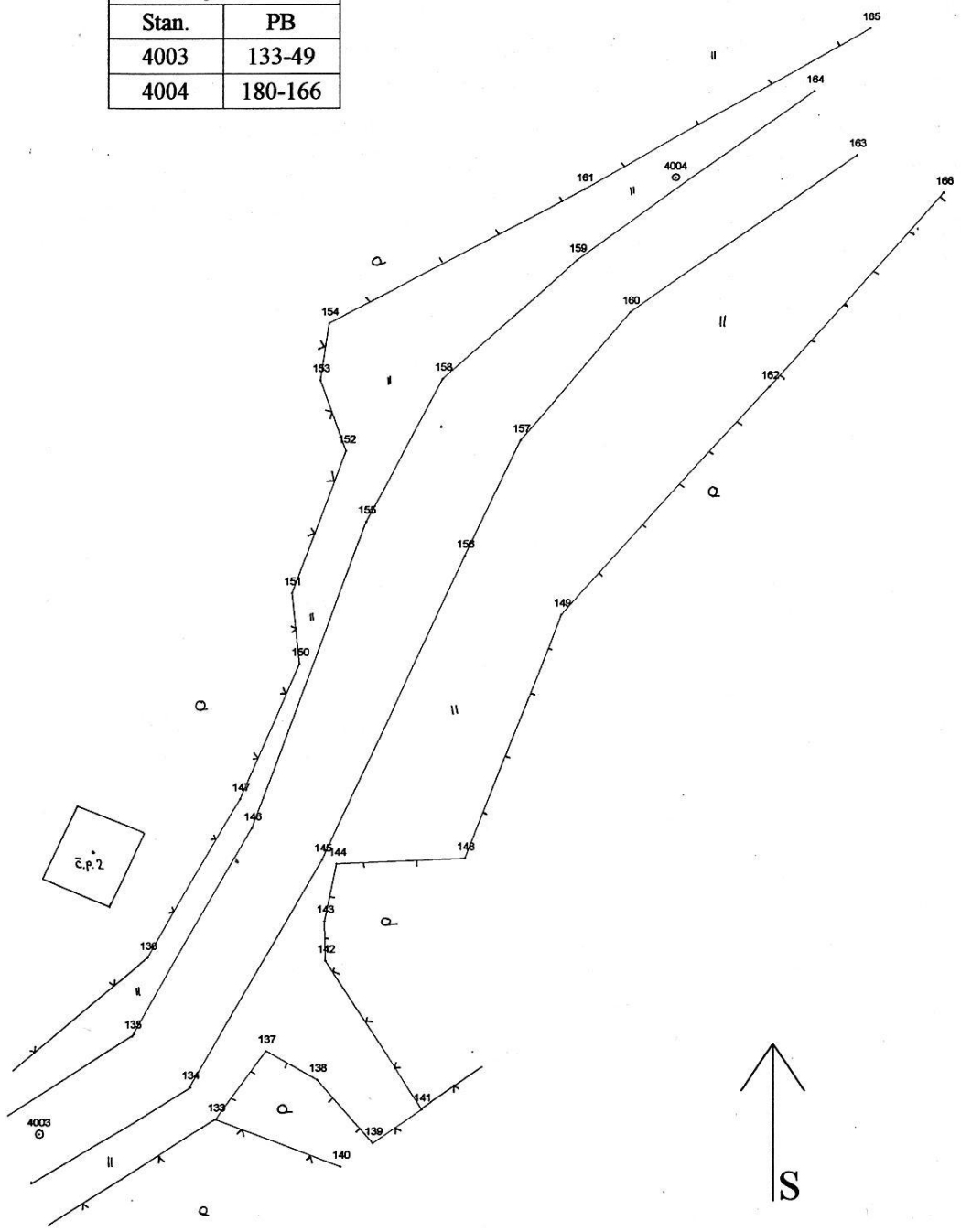
Měřický náčrt č.:2	
Stan.	PB
4002	69-92
4003	124-132

Příloha č. 7 b



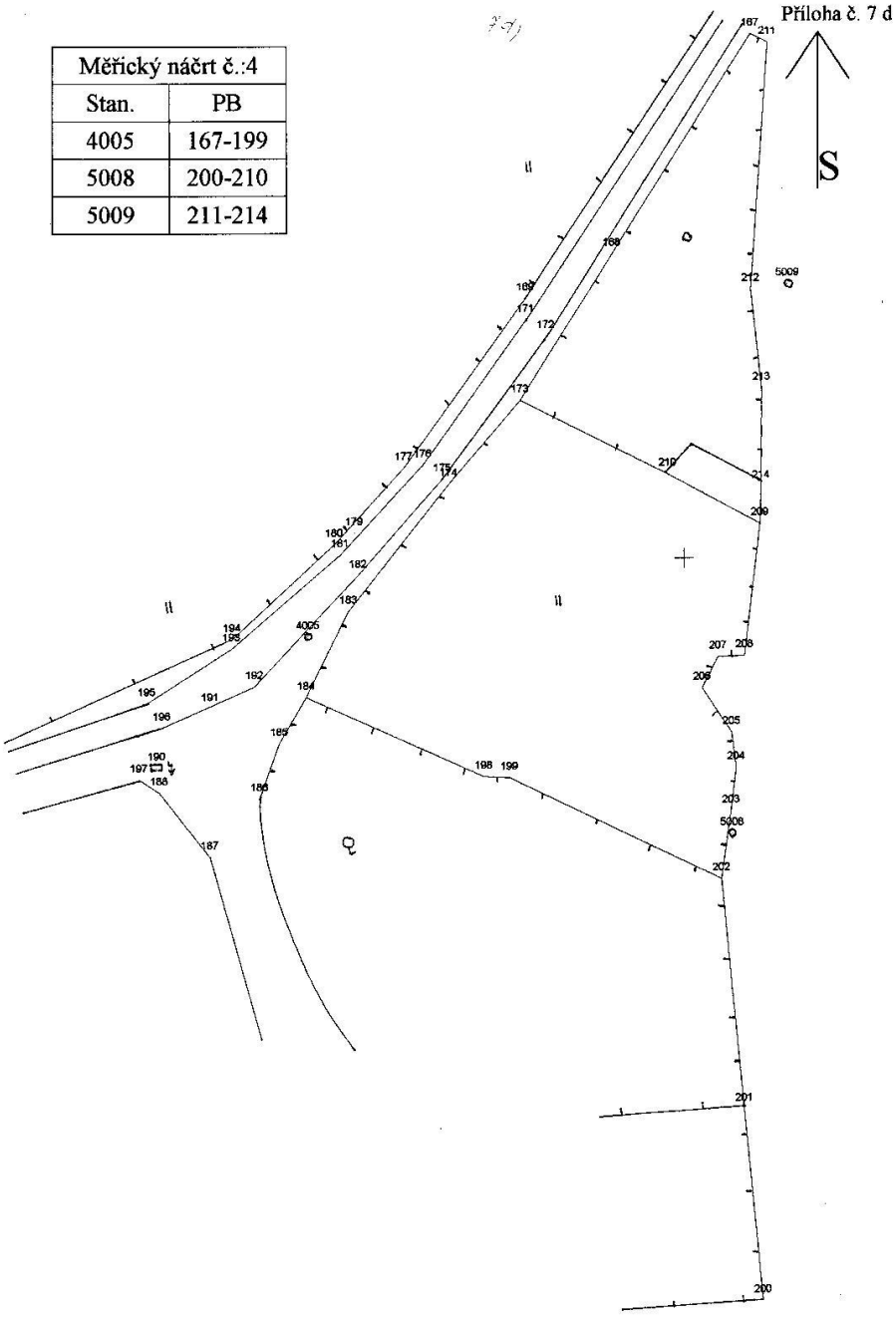
Příloha č. 7 c

Měřický náčrt č.:3	
Stan.	PB
4003	133-49
4004	180-166



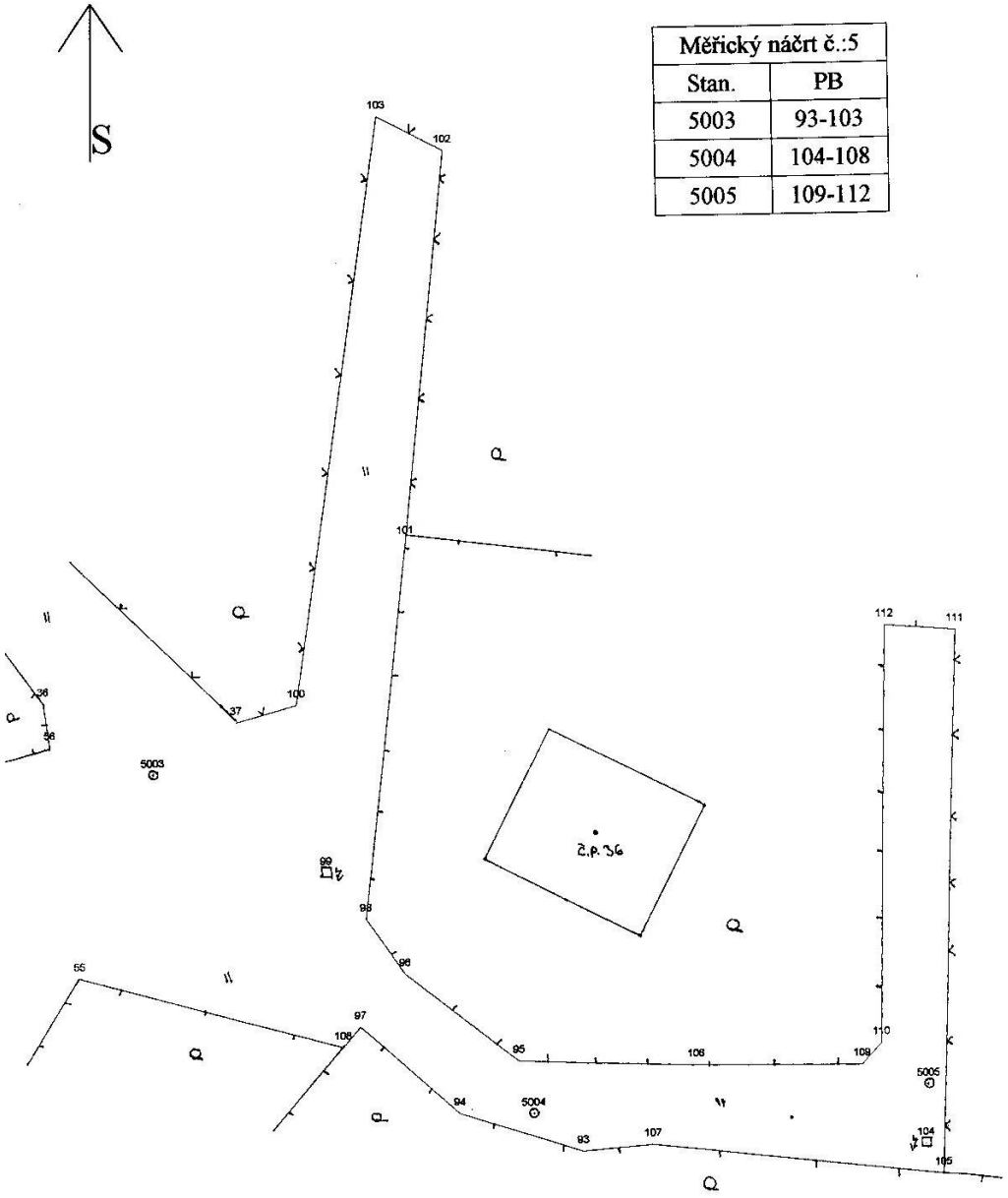
Příloha č. 7 d

Měřický náčrt č.:4	
Stan.	PB
4005	167-199
5008	200-210
5009	211-214



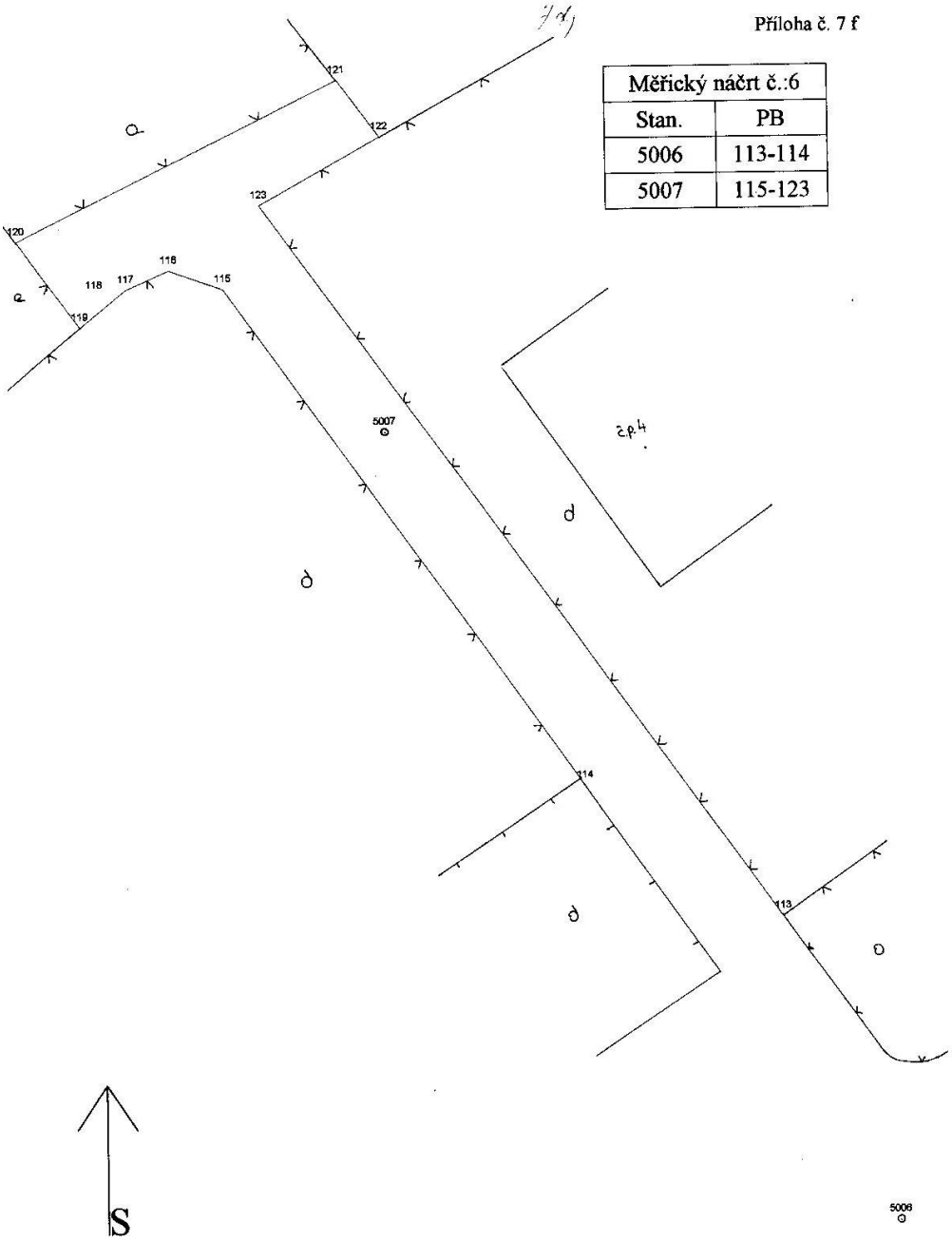
Příloha č. 7 e

Měřický náčrt č.:5	
Stan.	PB
5003	93-103
5004	104-108
5005	109-112



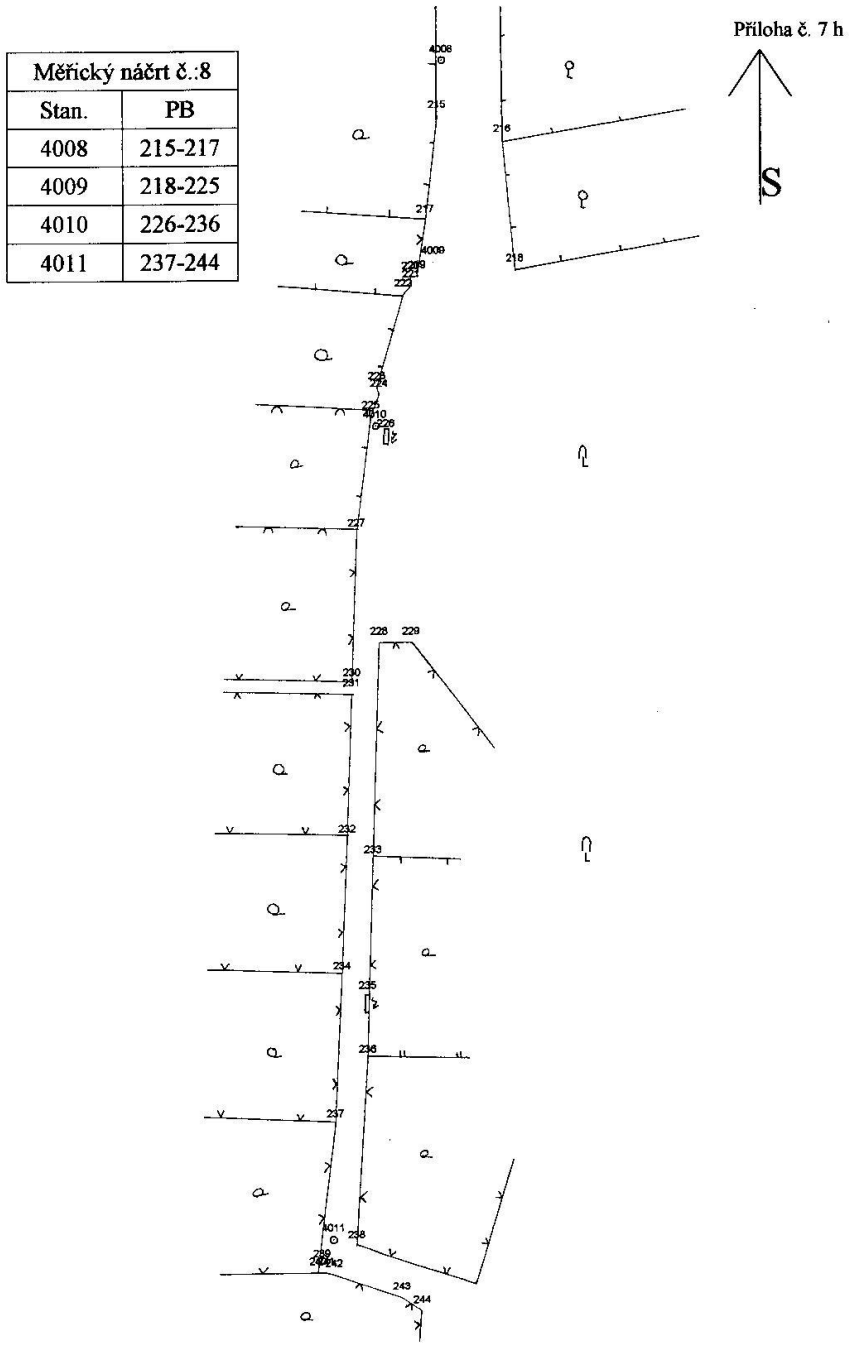
Příloha č. 7 f

Měřický náčrt č.:6	
Stan.	PB
5006	113-114
5007	115-123



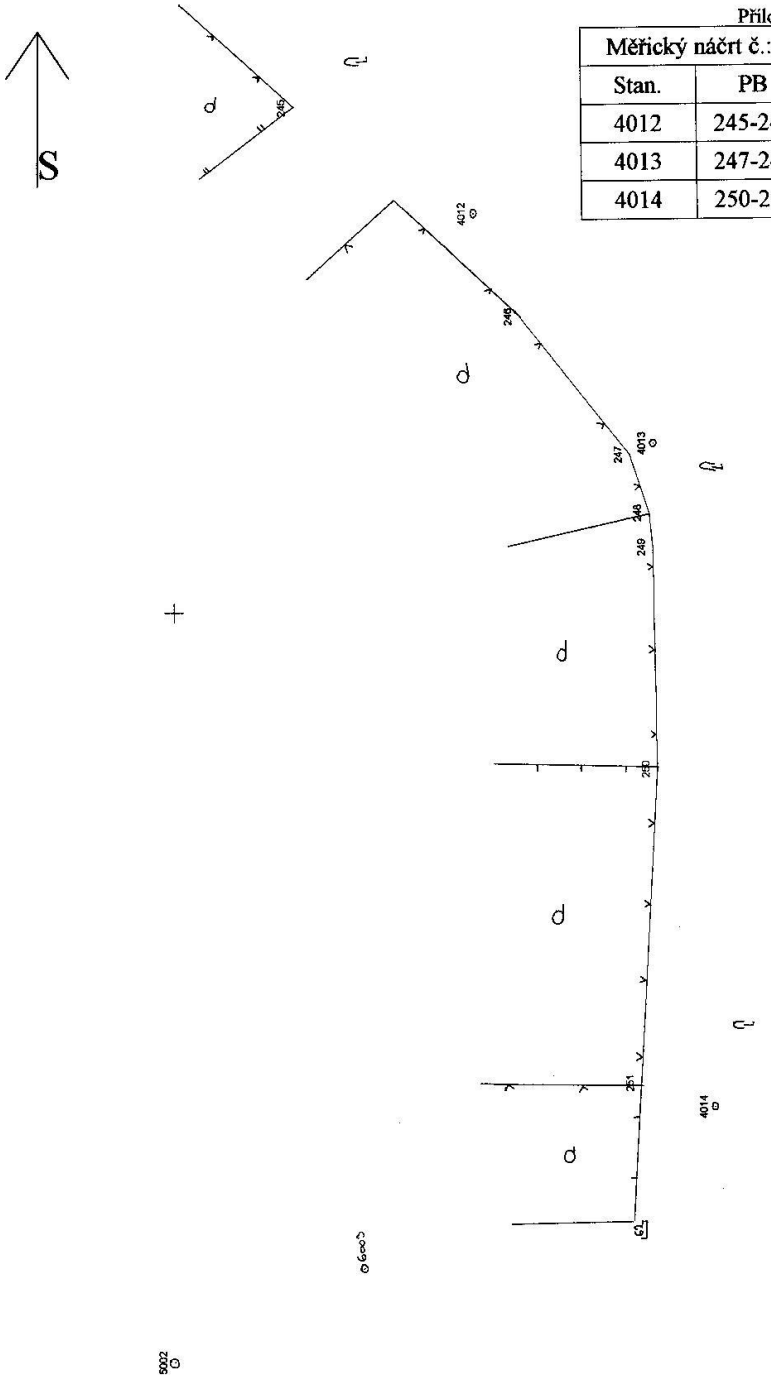
Příloha č. 7 h

Měřický náčrt č.:8	
Stan.	PB
4008	215-217
4009	218-225
4010	226-236
4011	237-244



Příloha č. 7 i

Měřický náčrt č.:9	
Stan.	PB
4012	245-246
4013	247-249
4014	250-251



Příloha č. 8 b

ZÁPISNÍK PODROBNÉHO MĚŘENÍ

Typ úlohy	Číslo k.ú.	Číslo náčrtu	Číslo bodu	Typ vzdálenosti	Staničení		Kolmice					Poznámka
	9	Číslo evidenční jednotky	Číslo trigonom bodu		Vzdálenost	Výška cíle (str.)	Vodorovný úhel	Svislý úhel		Doměrek	Polární kolmice	
1	2			3	4	5	6	7	8	9	10	11
1			4001		.	.	.	.	:	.	.	
			586 5		47.69	.	0.0000	.	:	.	.	
					47.69	.	199.9952	.	:	.	.	
			4002		50.23	.	266.9550	.	:	.	.	
					50.23	.	66.9604	.	:	.	.	
1			1		43.85	.	263.0118	.	:	.	.	
			2		41.30	.	268.3718	.	:	.	.	
			3		29.86	.	252.7370	.	:	.	.	
			4		28.13	.	260.7850	.	:	.	.	
			5		18.54	.	240.9850	.	:	.	.	
			6		18.16	.	254.4926	.	:	.	.	
			7		6.42	.	232.0706	.	:	.	.	
			8		7.07	.	269.4748	.	:	.	.	
			9		7.81	.	18.3442	.	:	.	.	
			10		8.22	.	385.4396	.	:	.	.	
			11		27.17	.	3.6294	.	:	.	.	
			12		27.00	.	394.5980	.	:	.	.	
			13		44.56	.	399.2736	.	:	.	.	
			14		44.48	.	394.3660	.	:	.	.	
			15		62.10	.	398.5066	.	:	.	.	
			16		62.32	.	394.5022	.	:	.	.	
			17		82.39	.	396.9680	.	:	.	.	
			18		82.38	.	394.9210	.	:	.	.	
			19		85.76	.	397.9460	.	:	.	.	
			20		83.07	.	398.3240	.	:	.	.	
			21		90.80	.	395.5040	.	:	.	.	
			22		48.68	.	399.3250	.	:	.	.	
			23		44.47	.	0.2478	.	:	.	.	
			24		35.30	.	393.4790	.	:	.	.	
			25		36.50	.	26.4500	.	:	.	.	
			26		56.80	.	28.1368	.	:	.	.	
			27		45.24	.	55.0868	.	:	.	.	
			28		58.86	.	58.8352	.	:	.	.	
			29		20.93	.	392.8344	.	:	.	.	
			30		13.66	.	371.9516	.	:	.	.	
			31		37.46	.	268.6964	.	:	.	.	
			5001		21.80	.	298.5732	.	:	.	.	
					21.80	.	98.5634	.	:	.	.	
			5002		74.73	.	397.3489	.	:	.	.	
					74.73	.	197.3374	.	:	.	.	

Geodézie č. 4.57 - 1983

Příloha č. 8 c

ZÁPISNÍK PODROBNÉHO MĚŘENÍ

Str. 2

Typ úlohy	Číslo k.ú.	Číslo náčrtu / Číslo evidenční jednotky	Číslo bodu / Číslo trigonom bodu	Typ vzdálenosti	Staničení: Vzdálenost	Výška cíle (str.)	Kolmice: Vodorovný úhel	Svislý úhel		Doměrek	Polární kolmice	Poznámka
1	2			3	4	5	6	7	8	9	10	11
1			5001									
			4001		21.80		83.8090					
					21.80		283.7920					
1			32		9.39		350.2164					
			33		2.68		29.5926					
			34		26.67		354.7630					
			35		28.15		365.1428					
			36		40.96		362.9774					
			37		47.69		355.9570					
			5002									
			4001		74.73		45.1818					
					74.73		245.1724					
1			38		12.11		64.8090					
			39		12.37		81.9934					
			40		14.07		384.2196					
			41		24.65		385.8758					
			42		34.92		386.8634					
			43		25.11		376.0946					
			44		22.42		374.8854					
			45		22.49		373.2188					
			46		22.22		368.7672					
			47		15.22		358.3810					
			48		15.31		357.4410					
			49		13.56		249.9136					
			50		19.62		242.7482					
			51		22.24		248.0518					
			52		12.49		139.0638					
			53		30.66		103.3684					
			54		31.00		92.4610					
			55		38.68		102.2674					
			56		47.48		95.2686					
1			5001		17.06		174.7816					
					17.06		374.7634					
			565		44.26		174.0404					
					44.24		374.0238					
			4001		91.79		177.5938					
					91.77		377.5586					
			5003									volné stan.

Geodézie č. 4.57 - 1983

Příloha č. 8 e

ZÁPISNÍK PODROBNÉHO MĚŘENÍ

Str. 4

Typ úlohy	Číslo k.ú.	Číslo náčrtu	Číslo bodu	Typ vzdálenosti	Staničení		Kolmice					Poznámka
	g	Číslo evidenční jednotky	Číslo trigonom bodu		Vzdálenost	Výška cíle (str.)	Vodorovný úhel	Svislý úhel		Doměrek	Polární kolmice	
1	2			3	4	5	6	7	8	9	10	11
1			91		5.03	.	3.3528	.	:	.	.	
			92		5.73	.	8.0302	.	:	.	.	
					.	.	.	.	:	.	.	
1			5002		.	.	.	.	:	.	.	
			5625		27.18	.	24.5114	.	:	.	.	
					27.19	.	224.5220	.	:	.	.	
			4001		74.74	.	29.1382	.	:	.	.	
					74.75	.	229.1420	.	:	.	.	
			5003		47.92	.	85.2852	.	:	.	.	
					47.91	.	285.2810	.	:	.	.	
						.		.	:	.	.	
			5003		.	.	.	.	:	.	.	
			5004		21.75	.	319.6134	.	:	.	.	
					21.75	.	119.6050	.	:	.	.	
1			93		24.39	.	319.0854	.	:	.	.	
			94		19.59	.	326.4734	.	:	.	.	
			95		19.84	.	315.5012	.	:	.	.	
			96		13.71	.	315.9566	.	:	.	.	
			97		13.94	.	329.5598	.	:	.	.	
			98		10.98	.	311.2770	.	:	.	.	
			99		8.48	.	306.0414	.	:	.	.	
			100		6.80	.	244.2636	.	:	.	.	
			101		14.71	.	225.5914	.	:	.	.	
			102		29.47	.	200.9380	.	:	.	.	
			103		29.75	.	194.0836	.	:	.	.	
						.		.	:	.	.	
			5004			.		.	:	.	.	
			5003		21.76	.	73.2084	.	:	.	.	
					21.76	.	273.1944	.	:	.	.	
			5005		16.89	.	222.1662	.	:	.	.	
					16.89	.	22.1448	.	:	.	.	
1			104		16.78	.	231.7568	.	:	.	.	
			105		17.69	.	236.0858	.	:	.	.	
			106		7.17	.	208.2728	.	:	.	.	
			107		5.19	.	243.0608	.	:	.	.	
			108		8.62	.	47.9416	.	:	.	.	
					.	.	.	.	:	.	.	
			5005		.	.	.	.	:	.	.	
			5004		16.90	.	85.5076	.	:	.	.	
					16.90		285.5044	.	:	.		

Geodézie č. 4.57 - 1983

ZÁPISNÍK PODROBNÉHO MĚŘENÍ

Str. 5

Typ úlohy	Číslo k.ú.	Číslo náčrtu	Číslo bodu	Typ vzdálenosti	Staničení		Kolmice					Poznámka
	9 Číslo evidenční jednotky	Číslo trigonom bodu			Vzdálenost	Výška cíle (str.)	Vodorovný úhel	Svislý úhel		Doměrek	Polární kolmice	
1	2			3	4	5	6	7	8	9	10	11
1		109			3.00	.	108.3800	.	:	.	.	
		110			2.69	.	135.6854	.	:	.	.	
		111			19.47	.	194.1698	.	:	.	.	
		112			19.75	.	184.1428	.	:	.	.	
					.	.	.	.	:	.	.	
		4002			47.49	.	00.00	.	:	.	.	
					47.49	.	199.9924	.	:	.	.	
		4003			15.70	.	204.5442	.	:	.	.	
					15.69	.	1.5370	.	:	.	.	
					.	.	.	.	:	.	.	
		5007			.	.	.	.	:	.	.	
		5006			40.12	.	399.9900	.	:	.	.	
					40.12	.	200.0006	.	:	.	.	
1		115			9.15	.	183.1878	.	:	.	.	
		116			11.43	.	177.9996	.	:	.	.	
		117			12.52	.	169.0138	.	:	.	.	
		118			13.61	.	164.9292	.	:	.	.	
		119			13.65	.	157.8876	.	:	.	.	
		120			17.63	.	167.0930	.	:	.	.	
		121			15.15	.	228.3708	.	:	.	.	
		122			12.61	.	235.8608	.	:	.	.	
		123			11.03	.	204.8446	.	:	.	.	
					.	.	.	.	:	.	.	
		4003			.	.	.	.	:	.	.	
		4004			54.13	.	104.1854	.	:	.	.	
					54.13	.	304.1794	.	:	.	.	
1		124			17.62	.	329.5044	.	:	.	.	
		125			30.45	.	339.8014	.	:	.	.	
		126			30.18	.	341.4216	.	:	.	.	
		127			29.33	.	350.2552	.	:	.	.	
		128			29.56	.	351.6618	.	:	.	.	
		129			17.00	.	324.6022	.	:	.	.	
		130			17.10	.	322.8270	.	:	.	.	
		131			7.59	.	274.4866	.	:	.	.	
		132			6.48	.	290.1512	.	:	.	.	
		133			8.40	.	161.2592	.	:	.	.	
		134			7.52	.	147.3736	.	:	.	.	
		135			6.39	.	115.1628	.	:	.	.	
		136			9.78	.	101.9764	.	:	.	.	
		137			11.42	.	144.3558	.	:	.	.	

Geodézie č. 4.57 - 1983

Příloha č. 8 h

ZÁPISNÍK PODROBNÉHO MĚŘENÍ

Str. 7

Typ úlohy	Číslo k.ú.	Číslo náčrtu	Číslo bodu	Typ vzdálenosti	Staničení		Kolmice					Poznámka
	9	Číslo evidenční jednotky	Číslo trigonom bodu		Vzdálenost	Výška cíle (str.)	Vodorovný úhel	Svislý úhel		Doměrek	Polární kolmice	
1	2			3	4	5	6	7	8	9	10	11
1			4005		.	.	.	.	:	.	.	
			4004		38.95	.	74.5404	.	:	.	.	
					38.95	.	274.5560	.	:	.	.	
			4006		106.85	.	241.6372	.	:	.	.	
					106.85	.	41.6414	.	:	.	.	
1			1667		85.01	.	242.6322	.	:	.	.	
			1668		55.55	.	245.0298	.	:	.	.	
			1669		45.74	.	238.6318	.	:	.	.	
			1670		44.12	.	239.4540	.	:	.	.	
			1671		43.70	.	240.6238	.	:	.	.	
			1672		43.59	.	244.8600	.	:	.	.	
			1673		36.09	.	248.7520	.	:	.	.	
			1674		23.56	.	249.4644	.	:	.	.	
			1675		23.49	.	246.9580	.	:	.	.	
			1676		23.47	.	239.5844	.	:	.	.	
			1677		22.07	.	235.0034	.	:	.	.	
			1678		14.08	.	230.9388	.	:	.	.	
			1679		12.93	.	228.9338	.	:	.	.	
			1680		10.95	.	219.9180	.	:	.	.	
			1681		10.02	.	226.0288	.	:	.	.	
			1682		9.01	.	245.3926	.	:	.	.	
			1683		5.42	.	265.7862	.	:	.	.	
			1684		6.91	.	4.5768	.	:	.	.	
			1685		12.58	.	19.2134	.	:	.	.	
			1686		19.23	.	20.9248	.	:	.	.	
			1687		27.39	.	29.0828	.	:	.	.	
			1688		24.47	.	50.6546	.	:	.	.	
			1689		22.41	.	50.5374	.	:	.	.	
			1690		22.72	.	56.6330	.	:	.	.	
			1691		13.74	.	61.7826	.	:	.	.	
			1692		8.34	.	54.5790	.	:	.	.	
			1693		8.74	.	92.2716	.	:	.	.	
			1694		8.57	.	100.4254	.	:	.	.	
			1695		19.63	.	77.1760	.	:	.	.	
			1696		19.50	.	66.5634	.	:	.	.	
			1697		25.08	.	57.4422	.	:	.	.	
			1698		25.33	.	345.4696	.	:	.	.	
			1699		27.72	.	341.4454	.	:	.	.	
			5008		52.76	.	329.9790	.	:	.	.	
					52.76	.	129.9688	.	:	.	.	

Geodézie č. 4.57 - 1983

ZÁPISNÍK PODROBNÉHO MĚŘENÍ

Str. 8

Typ úlohy	Číslo k.ú.	Číslo náčrtu / Číslo evidenční jednotky	Číslo bodu / Číslo trigonom bodu	Typ vzdálenosti	Staničení / Vzdálenost	Výška cíle (str.)	Kolmice: Vodorovný úhel	Kolmice: Svislý úhel		Doměrek	Polární kolmice	Poznámka
1	2			3	4	5	6	7	8	9	10	11
1			5008		.	.	.	.	:	.	.	
			4005		52.78	.	80.3736	.	:	.	.	
					52.75	.	280.3670	.	:	.	.	
1			200		53.08	.	348.7280	.	:	.	.	
			201		31.30	.	350.2338	.	:	.	.	
			202		5.28	.	367.8420	.	:	.	.	
			203		2.53	.	149.1240	.	:	.	.	
			204		7.50	.	156.6094	.	:	.	.	
			205		11.47	.	152.6374	.	:	.	.	
			206		16.85	.	139.9490	.	:	.	.	
			207		20.14	.	147.9062	.	:	.	.	
			208		20.29	.	157.3886	.	:	.	.	
			209		35.41	.	158.6158	.	:	.	.	
			210		41.45	.	141.6386	.	:	.	.	
					.	.	.	.	:	.	.	
1			5009		.	.	.	.	:	.	.	
			5008		62.76	.	325.4146	.	:	.	.	
					62.76	.	125.4062	.	:	.	.	
			221.1		.	.	140.9698	.	:	.	.	
					.	.	340.9622	.	:	.	.	
			4006		47.52	.	128.9006	.	:	.	.	
					47.53	.	328.9060	.	:	.	.	
1			211		27.53	.	113.9196	.	:	.	.	
			212		4.20	.	9.3840	.	:	.	.	
			213		12.21	.	334.6549	.	:	.	.	
			214		23.14	.	327.3344	.	:	.	.	
					.	.	.	.	:	.	.	
1			4008		.	.	.	.	:	.	.	
			4005		24.95	.	159.4422	.	:	.	.	
					24.95	.	359.4364	.	:	.	.	
			4009		29.63	.	342.0944	.	:	.	.	
					29.62	.	142.0850	.	:	.	.	
1			215		8.25	.	344.4144	.	:	.	.	
			216		14.73	.	297.9810	.	:	.	.	
			217		23.60	.	345.8016	.	:	.	.	
					.	.	.	.	:	.	.	
					.	.	.	.	:	.	.	
					.	.	.	.	:	.	.	
					.	.	.	.	:	.	.	
					.	.	.	.	:	.	.	

Geodézie č. 4.57 - 1983

ZÁPISNÍK PODROBNÉHO MĚŘENÍ

Typ úlohy	Číslo k.ú.				Číslo náčrtu				Číslo bodu				Typ vzdálenosti	Staničení		Kolmice					Poznámka
				9	Číslo evidenční jednotky				Číslo trigonom bodu					Vzdálenost	Výška cíle (str.)	Vodorovný úhel	Svislý úhel		Doměrek	Polární kolmice	
1	2												3	4	5	6	7	8	9	10	11
1									4	0	0	9		.	.	.	.	:	.	.	
									4	0	0	8		29.64	.	3.9654	.	:	.	.	
														29.64	.	203.9760	.	:	.	.	
									4	0	1	0		25.44	.	219.0496	.	:	.	.	
														25.44	.	19.0140	.	:	.	.	
1									2	1	8			11.99	.	107.3438	.	:	.	.	
									2	1	9			3.13	.	255.0640	.	:	.	.	
									2	2	0			3.97	.	264.3288	.	:	.	.	
									2	2	1			4.65	.	249.7228	.	:	.	.	
									2	2	2			6.46	.	249.1028	.	:	.	.	
									2	2	3			20.10	.	228.0850	.	:	.	.	
									2	2	4			20.98	.	225.9174	.	:	.	.	
									2	2	5			24.34	.	225.5742	.	:	.	.	
														.	.	.	.	:	.	.	
1									4	0	1	0		.	.	.	.	:	.	.	
									4	0	0	9		25.44	.	36.6460	.	:	.	.	
														25.43	.	236.6359	.	:	.	.	
									4	0	1	1		119.60	.	218.2104	.	:	.	.	
														119.60	.	18.2042	.	:	.	.	
1									2	2	6			2.07	.	158.8314	.	:	.	.	
									2	2	7			16.16	.	225.6214	.	:	.	.	
									2	2	8			31.86	.	213.9434	.	:	.	.	
									2	2	9			32.17	.	204.8196	.	:	.	.	
									2	3	0			38.09	.	220.6010	.	:	.	.	
									2	3	1			39.62	.	220.4642	.	:	.	.	
									2	3	2			61.10	.	219.2562	.	:	.	.	
									2	3	3			63.92	.	215.3288	.	:	.	.	
									2	3	4			81.12	.	218.7524	.	:	.	.	
									2	3	5			83.86	.	215.8146	.	:	.	.	
									2	3	6			93.21	.	215.7480	.	:	.	.	
														.	.	.	.	:	.	.	
1									4	0	1	1		.	.	.	.	:	.	.	
									4	0	1	0		119.59	.	255.3760	.	:	.	.	
														119.60	.	55.3826	.	:	.	.	
									4	0	1	2		24.19	.	35.6124	.	:	.	.	
														24.20	.	235.6586	.	:	.	.	
1									2	3	7			16.77	.	253.2008	.	:	.	.	
									2	3	8			3.46	.	370.1702	.	:	.	.	
									2	3	9			4.08	.	79.1830	.	:	.	.	
									2	4	0			5.42	.	80.1040	.	:	.	.	

Geodézie č. 4.57 - 1983

ZÁPISNÍK PODROBNÉHO MĚŘENÍ

Str. 10

Typ úlohy	Číslo k.ú.	Číslo náčrtu / Číslo evidenční jednotky	Číslo bodu / Číslo trigonom bodu	Typ vzdálenosti	Staničení: Vzdálenost	Staničení: Výška cíle (str.)	Kolmice: Vodorovný úhel	Kolmice: Svislý úhel		Kolmice: Doměrek	Kolmice: Polární kolmice	Poznámka
1	2			3	4	5	6	7	8	9	10	11
1			241		4.99	.	66.3818	.		.	.	
			242		5.16	.	51.4358	.		.	.	
			243		12.96	.	397.2996	.		.	.	
			244		16.44	.	395.7608	.		.	.	
					.	.	.	.	:	.	.	
1			4012		.	.	.	.	:	.	.	
			4011		24.25	.	234.0314	.	:	.	.	
					24.27	.	34.1024	.	:	.	.	
			4013		27.48	.	108.7300	.	:	.	.	
					27.48	.	308.7424	.	:	.	.	
1			245		19.64	.	284.3672	.	:	.	.	
			246		10.62	.	123.4320	.	:	.	.	
					.	.	.	.	:	.	.	
1			4013		.	.	.	.	:	.	.	
			4012		27.47	.	331.6348	.	:	.	.	
					27.48	.	131.6418	.	:	.	.	
			4014		63.04	.	168.3376	.	:	.	.	
					63.04	.	368.3316	.	:	.	.	
1			247		2.49	.	245.0486	.	:	.	.	
			248		6.64	.	178.1628	.	:	.	.	
			249		9.85	.	174.5466	.	:	.	.	
					.	.	.	.	:	.	.	
1			4014		.	.	.	.	:	.	.	
			4013		63.04	.	314.6224	.	:	.	.	
					63.04	.	114.6026	.	:	.	.	
			5002		56.49	.	191.9676	.	:	.	.	
					56.51	.	391.9754	.	:	.	.	
1			250		32.35	.	309.6718	.	:	.	.	
			251		7.29	.	240.1108	.	:	.	.	
					.	.	.	.	:	.	.	
1			5002		.	.	.	.	:	.	.	
			4014		56.50	.	83.5848	.	:	.	.	
					56.50	.	283.5540	.	:	.	.	
			5665		27.20	.	277.1748	.	:	.	.	
					27.20	.	77.1698	.	:	.	.	
			4001		74.75	.	281.8180	.	:	.	.	
					74.75	.	81.8230	.	:	.	.	
					.	.	.	.	:	.	.	
					.	.	.	.	:	.	.	
					.	.	.	.	:	.	.	

Geodézie č. 4.57 - 1983

Transformační parametry:

Rotace : 226.9950
Měřítko : 0.999725375864 (-27.5 mm/100m)

Bod	I. Y	I. X	II. Y	II. X
4001	770204.43	984532.35	0.00	47.68
4002	770174.91	984491.72	43.64	72.57
4003	770108.14	984469.62	113.61	65.24
4004	770078.17	984424.56	159.47	93.99
4005	770042.66	984408.56	198.43	93.97
4006	769979.70	984322.26	291.35	146.73
4007	769948.05	984198.48	371.15	246.55

Parametry polygonového pořadu:

Typ pořadu : Vetknutý, bez orientace
Délka přadu : 508.08m
Polohová odchylka : -0.12m
Největší / nejmenší délka v pořadu : 127.80m/ 12.07m
Poměr největší / nejmenší délka : 1:10.59
Max. poměr sousedních délek : 1:10.59
Nejmenší vrcholový úhel : 51.0686g

Vypočtené body:

Bod	Y	X
4001	770204.43	984532.35
4002	770174.91	984491.72
4003	770108.14	984469.62
4004	770078.17	984424.56
4005	770042.66	984408.56
4006	769979.70	984322.26
4007	769948.05	984198.48

Test polygonového pořadu:

Polohová odchylka [m]: Skutečná hodnota: -0.12, Mezní hodnota: 0.21
Mezní počet bodů : Skutečná hodnota: 7, Mezní hodnota: 3
Mezní délka pořadu [m]: Skutečná hodnota: 508.08, Mezní hodnota: 1500.00
Mezní délka strany [m]: Skutečná hodnota: 127.80, Mezní hodnota: 400.00
Mezní poměr délek : Skutečná hodnota: 1:10.59, Mezní hodnota: 1:3.00

Mezní odchylky stanovené pro práci v katastru nemovitostí byly dodrženy.
Byl překročen geometrický parametr stanovený pro práci v katastru nemovitostí.

Transformační parametry:

Rotace : -20.7092
Měřítko : 1.000055476521 (5.5 mm/100m)

Bod	I. Y	I. X	II. Y	II. X
4008	770050.63	984432.20	0.00	24.95
4009	770052.20	984461.80	-7.98	53.49
4010	770060.96	984485.68	-7.30	78.92
4011	770068.39	984605.06	-38.41	194.41
4012	770062.45	984628.55	-51.56	214.76
4013	770084.21	984645.33	-36.30	237.62
4014	770147.05	984650.39	21.62	262.50
5002	770170.92	984599.18	60.61	221.60

Parametry polygonového pořadu:

Typ pořadu : Vetknutý, bez orientace
Délka pořadu : 398.08m
Polohová odchylka : 0.01m
Největší / nejmenší délka v pořadu : 119.60m/ 24.23m
Poměr největší / nejmenší délka : 1:4.94
Max. poměr sousedních délek : 1:4.94
Nejmenší vrcholový úhel : 122.6410g

Vypočtené body:

Bod	Y	X
4008	770050.63	984432.20
4009	770052.20	984461.80
4010	770060.96	984485.68
4011	770068.39	984605.06
4012	770062.45	984628.55
4013	770084.21	984645.33
4014	770147.05	984650.39
5002	770170.92	984599.18

Test polygonového pořadu:

Polohová odchylka [m]: Skutečná hodnota: 0.01, Mezní hodnota: 0.20
Mezní počet bodů : Skutečná hodnota: 8, Mezní hodnota: 3
Mezní délka pořadu [m]: Skutečná hodnota: 398.08, Mezní hodnota: 1500.00
Mezní délka strany [m]: Skutečná hodnota: 119.60, Mezní hodnota: 400.00
Mezní poměr délek : Skutečná hodnota: 1:4.94, Mezní hodnota: 1:3.00

Mezní odchylky stanovené pro práci v katastru nemovitostí byly dodrženy.
Byl překročen geometrický parametr stanovený pro práci v katastru nemovitostí.

[1] POLÁRNÍ METODA

Orientace osnovy na bodě 5002:

Bod	Hz	Směrník	V or.	Délka	V délky	V přev.	m0 Red.
4001	45.1771	170.4109	0.0000	74.74	0.02		

Orientační posun : 125.2338g

Test polární metody:

Oprava orientace [g]: Skutečná hodnota: 0.0000, Mezní hodnota: 0.0800
Mezní odchylky stanovené pro práci v katastru nemovitostí byly dodrženy.

Bod	Hz	Délka	Y	X	Popis
38	64.8090	12.11	770172.81	984587.22	
39	81.9934	12.37	770169.52	984586.89	
40	384.2196	14.07	770184.84	984597.10	
41	385.8758	24.65	770195.20	984594.90	
42	386.8634	34.92	770205.21	984592.58	
43	376.0946	25.11	770196.02	984598.66	
44	374.8854	22.42	770193.34	984599.14	
45	373.2188	22.49	770193.40	984599.73	
46	368.7672	22.22	770193.04	984601.27	
47	358.3810	15.22	770185.64	984603.05	
48	357.4410	15.31	770185.67	984603.30	
49	249.9136	13.56	770165.76	984611.72	
50	242.7482	19.62	770161.46	984616.37	
51	248.0518	22.24	770161.86	984619.49	
52	139.0638	12.49	770160.34	984592.54	
53	103.3684	30.66	770157.60	984571.56	
54	92.4610	31.00	770162.41	984569.37	
55	102.2674	38.68	770154.73	984564.05	
56	95.2686	47.48	770155.89	984554.14	

GROMA v. 7.0	SEZNAM SOUŘADNIC				str. 1/4
Předč. Číslo	Y	X	Z	Kv.	Popis
1	770 180.910	984 495.340			
2	770 179.430	984 499.480			
3	770 192.660	984 504.880			
4	770 190.180	984 508.100			
5	770 200.380	984 514.260			
6	770 196.820	984 515.860			
7	770 203.920	984 525.950			
8	770 200.050	984 526.800			
9	770 203.370	984 540.090			
10	770 199.440	984 538.880			
11	770 194.690	984 557.710			
12	770 191.280	984 555.930			
13	770 185.650	984 572.760			
14	770 182.630	984 571.120			
15	770 177.580	984 588.340			
16	770 174.000	984 586.740			
17	770 167.020	984 605.760			
18	770 164.690	984 604.510			
19	770 166.670	984 609.350			
20	770 168.300	984 607.150			
21	770 161.350	984 612.280			
22	770 183.950	984 576.510			
23	770 186.310	984 572.960			
24	770 186.700	984 562.880			
25	770 204.130	984 568.850			
26	770 205.470	984 589.140			
27	770 223.770	984 573.250			
28	770 232.670	984 583.990			
29	770 193.740	984 550.340			
30	770 194.040	984 541.220			
31	770 181.600	984 502.650			
32	770 176.990	984 528.220			
33	770 185.470	984 525.470			
34	770 163.830	984 539.470			
35	770 165.790	984 543.740			
36	770 156.150	984 552.250			
37	770 147.890	984 553.190			
38	770 172.810	984 587.220			
39	770 169.520	984 586.890			
40	770 184.840	984 597.100			
41	770 195.200	984 594.900			
42	770 205.210	984 592.580			
43	770 196.020	984 598.660			
44	770 193.340	984 599.140			
45	770 193.400	984 599.730			
46	770 193.040	984 601.270			
47	770 185.640	984 603.050			
48	770 185.670	984 603.300			
49	770 165.760	984 611.720			
50	770 161.460	984 616.370			
51	770 161.860	984 619.490			
52	770 160.340	984 592.540			
53	770 157.600	984 571.560			
54	770 162.410	984 569.370			
55	770 154.730	984 564.050			
56	770 155.890	984 554.140			
57	770 165.910	984 616.470			
58	770 166.610	984 617.090			
59	770 166.600	984 618.040			
60	770 166.030	984 618.820			
61	770 168.360	984 632.850			
62	770 163.820	984 641.130			
63	770 143.230	984 620.540			
64	770 142.370	984 621.010			
65	770 140.990	984 621.420			
66	770 137.760	984 622.270			
67	770 136.900	984 618.960			
68	770 141.840	984 617.530			
69	770 108.250	984 469.430			
70	770 110.180	984 468.830			
71	770 113.930	984 470.400			
72	770 115.340	984 471.210			
73	770 116.710	984 473.620			
74	770 118.070	984 471.340			
75	770 119.370	984 470.920			
76	770 121.670	984 469.400			
77	770 123.860	984 473.410			
78	770 129.560	984 476.170			

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

TOPENÍ BIOMASOU

Mundil Petr , Hilenka Vít

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA DUCHCOV

OBSAH

1.0. Úvod

2.0. Biomasa

- 2.1. Odpadní biomasa
- 2.2. Výhřevnost biomasy
- 2.3. Sušení biomasy
- 2.4. Spalování biomasy
- 2.5. Palivo z biomasy
  - 2.5.1. Brikety
  - 2.5.2. Pelety
  - 2.5.3. Štěpka
  - 2.5.4. Palivové dřevo

2.5.5. Sláma, obilí

3.0. Kotle na biomasu

3.1. Krby, kamna na dřevo

3.1.1. Krbové vložky

3.1.2. Kamna na dřevo

3.1.3. Kachlová kamna

3.1.4. Teplovodní kotle na zplyňování dřeva

3.1.5. Výrobci kotlů na dřevo

3.2. Automatické kotle na dřevěné pelety

3.2.1. Výrobci kotlů na pelety

3.3. Kotle na spalování dřevní štěpky a pilin

3.3.1. Výrobci kotlů na spalování štěpky a pilin

3.4. Kotle na slámu

3.4.1. Výrobci kotlů na slámu

3.5. Exkurze v EKOEFEKTU s.r.o. v Třebívlicích u Lovosic

4.0. Vytápění biomasou v praxi

4.1. První kotelna na spalování biomasy v ČR

4.2. Další příklady kotelen v ČR

4.2.1. Městys Nová Cerekev

4.2.2. Kotelna v Plané u Mariánských lázní

4.2.3. Kotelna ve Vimperku

4.3. Návštěva kotelny v Třebívlicích

5.0. Závěr

6.0. SEZNAM POUŽITÉ LITERATURY:

7.0. BIOMASA – obnovitelný zdroj energie (Ing. Zdeněk Pastorek, CSc., Ing. Jaroslav Kára, CSc., Ing. Petr Jevič, CSc.)

8.0. ENERGIE Z BIOMASY (Mgr. Karel Murtinger, Ing. Jiří Beranovský, Ph.D.)

9.0.

10.0. Internetové servery a adresy:

11.0. www.tzb-info.cz
12.0. www.ekopaliva.cz
13.0. www.peletky-brikety.cz
14.0. www.wikipedie.cz
15.0. www.biom.cz
16.0. www.enviterm.cz
17.0. www.cmss.cz
18.0. www.krby-blanzek.cz
19.0. www.tts.cz

1.0. Úvod

Zásobování palivy a energiemi je problém, který znepokojuje celou společnost. Všichni vnímáme rostoucí spotřebu energie, rychlí pokles zásob fosilních paliv a negativní dopady na životní prostředí. Jsou to právě obnovitelné zdroje energie, které hrají v energetických koncepcích vyspělých států, hlavně v EU, stále větší roli.

Česká republika se zavázala zvyšovat podíl energie vyrobené z obnovitelných zdrojů na její celkové výrobě. Do roku 2010 by tento podíl měl být 12,5 % a do roku 2020 dokonce 20 %. Česká republika se zavázala k přijetí tzv. uhelné daně, která má znevýhodnit užití právě neobnovitelných zdrojů a podpořit rozvoj zdrojů obnovitelných. Biomasa, jako palivo pro vytápění a ohřev, je nejperspektivnější se rozvíjející oblast alternativních zdrojů energie z hlediska šetrnosti k životnímu prostředí, obnovitelnosti a do budoucna i z hlediska ekonomického.

2.0. Biomasa

Biomasa je souhrn látek tvořících těla organismů, a to jak rostlin, tak živočichů. Tímto pojmem je často označována rostlinná biomasa využitelná pro energetické účely. Biomasa je buď záměrně získávána jako výsledek výrobní činnosti, nebo se jedná o využití odpadů ze zemědělské, potravinářské a lesní výroby, komunálního hospodářství, z údržby krajiny a péče o ni.

Ekologie definuje biomasu jako celkovou hmotu jedinců určitého druhu, skupiny druhů nebo všech druhů společenstva na určité ploše. U rostlin se vyjadřuje v hmotnosti sušiny, u živočichů také v čerstvé hmotnosti (v joulech, dříve i v kaloriích, obsahu uhlíku apod.). U půdních a vodních organismů se vztahuje také k objemu (litr, cm^3, m^3). U rostlin se rozlišuje biomasa podzemní nebo nadzemní, biomasa živá nebo mrtvá (stařina).

Z hlediska způsobu využití lze biomasu rozdělit do dvou skupin:

- Biomasa suchá, s vlhkostí do 40 %, kterou je možné po eventuálním vysoušení spalovat (dřevo, obilní sláma, biomasa jiných energetických rostlin).
- Biomasa vlhká, s vlhkostí nad 40 %, která se využívá zpravidla k výrobě bioplynu (kejda, hnůj nebo kaly z čističek vod).

O výhodách biomasy není pochyb. Dobře se ukládá, má nízký obsah síry, a tedy i oxidu siřičitého ve spalinách, pouhý zlomek v porovnání například s hnědým uhlím. Biomasa je (na rozdíl od fosilních paliv) široce dostupná, není třeba ji dovážet na velké vzdálenosti, využití biomasy podporuje lokální ekonomiku a snižuje závislost státu na dovážených zdrojích energie. Patrně největší nevýhodou je nízká energetická účinnost vzniku biomasy. V relativně hustě zalidněné střední Evropě to znamená, že i v případě, kdy se bude biomasa cíleně a efektivně pěstovat, tak není k dispozici tolik volné půdy, aby se mohla stát hlavním zdrojem energie a náhradou fosilních paliv. Je proto třeba používat biomasu především tam, kde je to nejvýhodnější. Například náhrada kotlů na hnědé uhlí moderními kotli na pyrolýzní spalování dřeva může dramaticky zlepšit ovzduší v nějaké vesnici ležící v inverzní kotlině. Spálení téhož množství dřeva společně s uhlím ve velké, moderní, uhelné elektrárně žádný efekt na čistotu vzduchu nemá.

2.1. Odpadní biomasa

Odpadní biomasa je zde chápána jako biomasa, která již byla člověkem nějak (jinak než energeticky) využita nebo která slouží primárně k jiným účelům, než je produkce energie. Převážně jde o odpady z odvětví, která nějak využívají a zpracovávají biomasu:

- Rostlinné odpady ze zemědělské výroby (řepková, kukuřičná nebo obilná sláma, seno)
- Odpady z údržby krajiny či sadů (prořezy, křoviny a náletové dřeviny) a odpady z údržby travnatých ploch
- Odpady po těžbě dříví (kůra, vršky stromů, větve, šišky, pařezy, kořeny a podobně)
- Odpady z různých dřevozpracujících provozů (odřezky, piliny, hobliny)
- Odpady z potravinářských výrob (cukrovary, jatka, mlékárny, lihovary)
- Některé jinak nevyužité vedlejší produkty z živočišné výroby (hnůj, kejda, zbytky krmiv, odpady z přidružených zpracovatelských kapacit)
- Komunální organické odpady

Spalování komunálního odpadu je v některých případech vlastně také energetické využití odpadní biomasy.

2.2. Výhřevnost biomasy

Výhřevnost suché biomasy je zpravidla v rozmezí 15 – 19 MJ/kg; s rostoucím obsahem pryskyřic nebo olejů výhřevnost roste a naopak výrazně klesá s rostoucím obsahem vody (viz graf).



Graf. : Závislost výhřevnosti na obsahu vody

Výhřevnost zcela zdravého a suchého dřeva je poměrně vysoká: u listnáčů je to 18 MJ, u jehličnanů 19 $MJ*kg^{-1}$. Stejné hodnoty výhřevnosti mají stébelniny, zejména sláma obilovin a traviny. To je asi necelá polovina výhřevnosti ropných paliv. Ve skutečnosti však biomasa vždy obsahuje nejméně 10 % vody, v průměru mají dřevo i štěpka provětrávané a skladované pod střechou asi 30 % vlhkosti. Vlhkost slámy v balících uskladněných v halových skladech nebo v zakrytých stozích dosahuje 14 až 16 %. Při hoření se tato voda odpařuje a tím snižuje základní výhřevnost sušiny biomasy. Je-li teplota spalin za kotlem (výměníkem) vyšší než 101 °C, toto teplo, obsažené ve vodní páře, se nevyužije.

Výhřevnost paliva se však také snižuje s časem, zejména činností mikroorganismů, hub a plísní.

DRUH PALIVA	OBSAH VODY	VÝHŘEVNOST	MĚRNÉ HMOTNOSTI		
	[%]	[MJ/kg]	[kg/m^3]= [kg/plm]	[kg/prm]	[kg/prms]
listnaté dřevo	15	14,605	678	475	278
jehličnaté dřevo	15	15,584	486	340	199
borovice	20	18,4	517	362	212
vrba	20	16,9			
olše	20	16,7			
habr	20	16,7			
akát	20	16,3			
dub	20	15,9	685	480	281
jedle	20	15,9			
jasan	20	15,7			
buk	20	15,5	670	469	275
smrk	20	15,3	455	319	187
bříza	20	15,0			
modřín	20	15,0			
topol	20	12,9			
dřevní štěpka	30	12,18			210
sláma obilovin	10	15,49		120	(balíky)
sláma kukuřice	10	14,40		100	(balíky)
lněné stonky	10	16,90		140	(balíky)
sláma řepky	10	16,00		100	(balíky)

Výhřevnost biomasy. Zdroj: EkoWATT

2.3. Sušení biomasy

Vzhledem k výraznému vlivu vlhkosti na výhřevnost je nevyhnutelné biomasu před spalováním vysušit. Všeobecně se doporučuje snížit vlhkost pod 30%. Za optimální se považuje vlhkost do 20%. Tu lze ještě dosáhnout běžným sušením pod přístřeškem (kusové dřevo se obvykle nechává vyschnout 2 roky). Pro některé účely (například lisování briket nebo peletek) se musí materiál vysušit na podstatně nižší obsah vody, k tomu již nestačí běžné sušení na vzduchu a je nutné použít sušení při zvýšené teplotě.

2.4. Spalování biomasy

Při zahřátí na teplotu přes 200°C dochází k rozkladu a tvorbě hořlavých plynů. Biomasa hoří dlouhým plamenem, což vyžaduje vhodnou konstrukci topeniště. Do plamene se totiž musí přivést dostatečné množství vzduchu a musí dojít k jeho promíšení, aby mohly všechny těkavé složky dokonale vyhořet. K tomu je třeba dostatečná teplota (plamen se nesmí příliš ochladit). V topeništi dochází napřed k odpařování vody, která je vždy v určité míře přítomna. Voda má vysoké výparné teplo, takže její odpařování palivo ochlazuje. Teprve po odpaření vody vzroste teplota a začne docházet k uvolňování

prchavých látek (např. terpenů) a k tepelnému rozkladu (pyrolýze) jednotlivých sloučenin v biomase obsažených. Vzniká směs hořlavých plynů a na roštu zbývá dřevěné uhlí. Uvádí se, že uvolněná prchavá hořlavina v sobě nese přes polovinu energie ve dřevě obsažené. Hořlavé plyny hoří s primárním vzduchem přiváděným pod rošt dlouhým plamenem (primární spalování). Nedojde ale ke spálení všech spalitelných složek, protože k tomu zpravidla není dostatek kyslíku nebo dostatečně vysoká teplota. Pokud je pod rošt přiváděn nadbytek vzduchu, tak se plamen příliš ochladí a část hořlaviny se vyloučí ve formě sazí (uhlík). Když je vzduchu málo, tak zase nemůže dojít k úplnému spálení (oxidaci až na oxid uhličitý). V každém případě to znamená, že ztratíme část energie v palivu obsažené. Je tedy třeba zajistit, aby spalovací komora, v níž hoření probíhá, měla dostatečně vysokou teplotu a malé tepelné ztráty, a do plamene se v určité vzdálenosti nad roštem musí přimíchávat sekundární vzduch, který umožní dohoření nespálených plynů. Tím se uvolní i zbylá energie v palivu obsažená a do komína pak odchází jen oxid uhličitý, vodní pára a dusík. Vzdušný dusík se spalování prakticky neúčastní, pokud je ovšem teplota plamene dostatečně vysoká, dochází v určité míře k reakci dusíku a kyslíku za vzniku směsi oxidů dusíku. Tato reakce je z energetického hlediska nevýznamná, nicméně velmi zásadní je z hlediska emisí (oxidy dusíku jsou považovány za škodlivé emise).

2.5. Palivo z biomasy

Vyrábí se z čisté biomasy a to z pilin, hoblin nebo oprané kůry. Je to nevyčerpatelný zdroj, kterého více než se spotřebuje – na rozdíl od klasických fosilních paliv, jejichž zásoby se snižují a tím jejich cena roste.

Z pohledu budoucnosti topení biomasou, ze které se paliva vyrábějí, nejstarší a také jediný obnovitelný zdroj paliva, který máme k dispozici pro tři tisíciletí (slunce a voda v této oblasti nestačí).

Samostatná výroba biopaliva probíhá tak, že se dřevní odpady nejprve podrtí na jemnou frakci, dále se vysuší na minimální vlhkost a nakonec se slisují do válcových výlisků o vysoké hustotě.

Většina biopaliv z biomasy jsou lisované jen z čisté nekontaminované dřevní biomasy bez použití jakýchkoliv lepidel. Velmi důležitým parametrem všech evropských norem na tyto paliva je vedle výhřevnosti a popelnatosti i hodnota slisování biopaliv, vyjádřená měrnou hmotností. To nesmí být nikdy menší než 1000kg/m^3, tedy slisovaná biopaliva nesmí být lehčí než voda (tzn. kvalitní biopaliva na vodě neplovou). Dobře slisovaná biopaliva vysoké měrné hmotnosti nepřijímají vzdušnou vlhkost čímž se prodlužuje jejich doba skladování a hlavně déle hoří.

Biopaliva i dřevo jsou velkým ekologickým přínosem pro naše životní prostředí. Neobsahují žádná pojiva ani síru, halogeny či těžké kovy. Při jejich spalování se uvolňuje jen tolik oxidu uhlíku, kolik ho stromy spotřebují při svém růstu. To znamená, že se netvoří skleníkové efekty jako z fosilních paliv, které pak způsobují skleníkový efekt.

Biopaliva jsou slisované do malého objemu, takže nezabírají velký prostor. Při skladování je vhodné ukládání na dřevěné podlážky nebo palety v suchém skladu. Voda pevná biopaliva poškozuje a dotykem vody se mohou opět rozpadnout na piliny, ale v suchém prostředí mají prakticky neomezenou skladovatelnost. Obaly používané pro balení biopaliv jsou vyrobeny z polyetylenové fólie, která není ekologická pro spalování, ale je běžně recyklovatelná (tzn. obaly by neměly končit v ohni, ale ve sběrnách pro druhotné zpracování plastů).

Biopaliva se svojí výhřevností 18-19 MJ/kg řadí mezi hnědé a černé uhlí a to díky malé zbytkové vlhkosti v rozmezí 8-10 % (posekané dřevo pod přístřeškem vyschne za dva roky pouze na 15-20 % a dosáhne tím výhřevnosti jen 14,3 MJ/kg).

Velmi nízký je obsah popele. Dřevěná biopaliva mají do 1 % (uhlí až 30 %). Popel z biopaliv rozhodně nepatří do popelnice. Pokud jste nespalovali i obalové folie pak je i váš popel nekontaminovaný. Takový popel obsahuje mimo jiné dusík – N, fosfor – P, draslík – K, vápník – Ca, hořčík – Mg, hydroxid draselný, kysličník křemičitý, kyselinu fosforečnou a důležité stopové prvky. Je to velmi vzácné minerální hnojivo pro vaši zahrádku nebo trávník.

Mezi biopaliva patří:

- Brikety
- Pelety
- Topné dřevo
- Štěpka
- Sláma, obilí

2.5.1. Brikety

Brikety dřevěné pilinové

Pilinové brikety jsou vyrobeny téměř výhradně z odpadních pilin a velmi drobných dřevěných frakcí. Vyznačují se velmi vysokou měrnou hmotností oproti ostatním briketám. Jsou vhodné jak na zatápění, tak do již vytopených domácností k dlouhodobému udržení ohně. Protože surovinou je čisté dřevo, zůstává po spálení jen okolo 0,3 procent popela. Díky minimální zbytkové vlhkosti (do pěti procent) dosahují kůrové brikety výhřevnosti až 19 MJ/kg.



Pilinové brikety od výrobce BIOMAC

Brikety dřevěné hoblinové

Hoblinové brikety jsou vyrobeny z běžného dřevního odpadu.
Jsou vhodné pro vytopení místnosti, i pro udržení tepla.



Hoblinové brikety od výrobce BIOMAC.

Brikety dřevěné kůrové

Hlavní předností kůrových briket je téměř dvojnásobná doba hoření a žhnutí oproti dřevěným briketám. Jsou vhodné do již vytopených domácností k dlouhodobému udržení ohně - je to briketa na noc. Protože surovinou je čistá, ve vodě vypraná kůra, zůstává po spálení jen do pěti procent popela. Díky malé zbytkové vlhkosti (do osmi procent) dosahují kůrové brikety výhřevnosti až 18 MJ/kg.



Kůrové brikety od výrobce BIOMAC.

2.5.2. Pelety

Pelety jsou granule kruhového průřezu o průměru 6-8 mm a délce 20-35 mm, vyrábí se vysokotlakým lisováním dřevěných pilin bez jakýchkoliv chemických příměsí a pojiv. Dřevěné piliny jsou v kontinuální sušárně usušeny na požadovanou vlhkost a poté podrceny na frakci cca 4 mm. Proces sušení a dávkování je řízen automatickou regulací, teplo pro sušení je vyráběno ekokotlem. Topným mediem bývá biomasa. Poté se piliny lisují pod vysokým tlakem pomocí prstencového lisu do finální podoby. Hotové pelety z lisu jsou vynášeny do chladiče, kde jsou chlazeny proudem vzduchu, čímž dojde k jejich vytvrzení.

Dřevěné pelety mají výhřevnost 18-19 MJ/kg, což je v žebříčku paliv řadí na místo mezi hnědým a černým uhlím. Mají však vyšší účinnost spalování a na rozdíl od uhlí, které má po spálení až 30% pevného odpadu, mají dřevěné pelety pouze 0,5-1,5 % odpadu (popela). Popel z tohoto druhu paliva je zároveň výborným zahradním minerálním hnojivem. Ve srovnání s klasickým dřevem, které má při dvouletém skladování vlhkost 15-20 % a tím i nižší výhřevnost – cca 14 MJ/kg, mají dřevěné pelety vlhkost jen 8-10 % a zabezpečují svými vlastnostmi mnohem vyšší životnost spalovacích zařízení.



Pelety

2.5.3. Štěpka

Palivová štěpka představuje strojně zpracované částice ze surových zbytků lesní těžby a dřevin z plantáží v délce zhruba do 50 až 80 mm, v závislosti na použitém strojním zařízení. Další sušení je sice možné, zvyšuje však náklady ovlivňující konečnou prodejní cenu. Se vzrůstajícím obsahem vody se snižuje její kvalita pro energetické účely. Nejhodnotnější je v tomto směru štěpka vznikající jako vedlejší produkt základního dřevozpracujícího průmyslu, která se však většinou používá k jiným účelům (výroba pelet a briket).

Jelikož výhřevnost štěpky silně závisí na její vlhkosti, je velmi účelné ji před spalováním usušit. Rozdíl ve výhřevnosti je dán ztrátou energie na odpaření vody, která je ve vlhké štěpce obsažena. Pro její sušení však jsou i další důvody, např. vhodnější vlastnosti pro hoření v kotli, zachování energetické

hodnoty, vyšší trvanlivost, apod. Uvádí se, že pokud má štěpka obsah vody vyšší než 25–30 %, po určité době, závislé na teplotě, začíná degradovat a plesnivět.

Dosoušení štěpky pod střechou podporuje zkušenost, že jen tam se podstatně snižuje obsah vody a zvyšuje výhřevnost, která je jinak relativně velmi nízká. Obsah vody přirozeným průvanem přes léto dosoušené štěpky pod střechou klesá ze cca 43 % na cca 20 %, kdežto z štěpky skladované venku pouze na cca 30 %. Výhřevnost, jako nejdůležitější faktor paliv se zvyšuje u štěpky skladované pod střechou na cca 15 GJ.t^{-1}, u štěpky venku na 12 GJ.t $^{-1}$.

Výhřevnost energetické štěpky s různým obsahem vody:

Stav štěpky	Obsah vody (%)	Výhřevnost (MJ/kg)
čerstvá	55	7
zavadlá	40	10,5
polosuchá	30	12,2
suchá	20	15

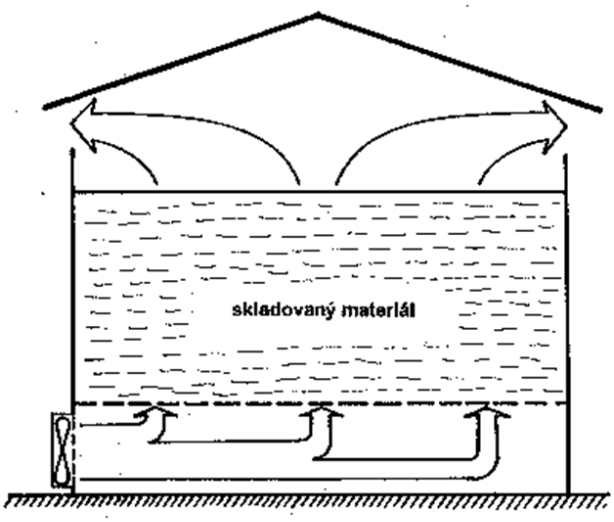
Z procentuálního rozdělení vyprodukované dendromasy pro listnaté a jehličnaté stromy je možno určit množství štěpky z roční produkce dříví vytěženého v ČR.

Poměrné rozdělení vyprodukované dendromasy (%):

	Jehličnany	Listnáče
vršek **větve**	15-25	5 10-15
kmen	60-65	60-65
pařez **kořeny**	10-15	5-10 10-20

Při uvážení zpracovatelnosti vršků a větví dostaneme roční potenciální množství štěpky zhruba 3,8 mil. m^3. Při převaze těžby jehličnanů a průměrné měrné hmotnosti 270 $kg.m^{-3}$ docházíme k množství štěpky asi 1 mil.t. ročně, kterou však z různých důvodů není zatím možno všechnu vytěžit. Rozdíl však mohou uhradit zpracovatelné odpady z pil.

Dosoušení štěpky pod střechou je lepší, začínají se k tomu využívat velkokapacitní seníky, které jsou v důsledku poklesu skotu nevyužité. Výhodou tohoto řešení je možnost použití stávající technologie, především roštů, sušících ventilátorů a manipulačních jeřábů. Tato technologie vyžaduje úpravu. Rošty a jeřáb je třeba přizpůsobit malým rozměrům štěpky tak, aby nedocházelo k jejímu propadávání jak při manipulaci, tak při skladování na roštech. Ventilátory mohou být provozovány pouze v případech velmi vhodných meteorologických podmínek, tj. za vysokých teplot a nízké vlhkosti vzduchu. Rovněž dodávaná množství vzduchu mohou být snížena, např. snížením otáček ventilátorů. Výzkumně se též řeší možnosti přirozeného větrání. Všechny tyto způsoby vedou k minimalizaci spotřeby elektrické energie, která tvoří významnou složku provozních nákladů.



Schematický řez velkokapacitním senikem

2.5.4. Palivové dřevo

Vytápění domácností dřevem se Čechům stále více vyplácí. Vloni se v ČR spálilo něco přes 1,1 milionu metrů krychlových palivového dřeva. Spotřeba se tedy zvýšila o 40 % oproti r. 2002. Topení dřevem je čisté a ekologické. Skladuje se lépe než dřevěné pelety, výhřevností je s peletami nebo dřevěnými briketami plně srovnatelné.

Někdo by mohl namítnout, aby dřevo z lesů nezmizelo rychleji než stačí dorůst, proto existuje několik druhů těžby dřeva:

- Předmýtní úmyslná těžba – jsou tzv. probírky v porostech do 40 let a nad 40 let. Provádí se za účelem dosažení vyšší kvality lesa. Jedná se v podstatě o vyřezání nevhodně rostoucích stromů (tj. slabých) nebo uvolnění kvalitních stromů v hustě rostoucích skupinách.
- Mýtní úmyslná těžba – je těžba zralých lesních porostů, které dosáhly tzv. mýtného věku a mohou se tedy vytěžit. Mýtní těžbou získává majitel lesa kvalitní dřevo.
- Nahodilá těžba – je zpracování stromů napadených škůdci (kůrovec, václavka) či vyvrácených nebo jinak poškozených. Veškeré úmyslné těžby se řídí lesním hospodářským plánem nebo osnovou.

Palivové dřevo se dodává štípané nebo v metrech. Štípané palivové dřevo je výborné pro topení v krbu nebo krbových kamnech. Pro optimální výhřevnost dřeva je dobré mít dřevo proschlé, tj. nejlépe 1-2 roky, proto je dobré si dřevo zajistit už v létě, aby doma dobře doschlo.



Palivové dřevo

2.5.5. Sláma

Sláma na poli je levný zdroj a energetické obilí (např. Triticale) dává v porovnání se vstupy vysoký výnos energie. Vhodný druh energetické plodiny je určován mnoha faktory: druhem půd, způsobem využití a účelem, možností sklizně a dopravy, druhovou skladbou v okolí atp. Předem se musí porovnat náklady na pěstování a na výrobu (spotřebu energie) a výnosy (zisk) energie.

Z bylin jsou zajímavé rostliny produkující cukr, škrob nebo olej. Například brambory, cukrová řepa, slunečnice a zejména řepka (řepkový olej se zpracovává na naftu a mazadla, řepková sláma se využije ke spálení). Řepková sláma má výhřevnost 15-17,5 GJ/t, obilná sláma o něco nižší - 14,0-14,4 GJ/t.

Z víceletých rostlin je známá křídlatka sachalinská (Reynoutria sachalinensis Nakai), která dosahuje vysokých výnosů 30-40 t sušiny z ha. Velmi diskutovanou energetickou rostlinou je sloní tráva (Miscanthus sinensis). Výhodné je pěstování konopí setého (Cannabis sativa L.), neboť nevyžaduje žádné ošetření v průběhu vegetace. V Evropě dosahuje výšky až 4 m a výnosu 6-15 t suché hmoty z hektaru. Konopí je jednoletá rostlina, ale na stanovišti vydrží, pokud se vysemení, mnoho let (odtud např. Konopiště).

Pro vytápění se sláma, seno a podobné traviny lisují do balíků, které se pak přikládají jako celek.

PLODINA/TERMÍN	VÝHŘEVNOST [MJ/kg]	VLHKOST	VÝNOS [t/ha] min.	prům.	opt.
Sláma obilovin (VII-X)	14	15	3	4	5
Sláma řepka (VII)	13,5	17-18	4	5	6
Energetická fytomasa - orná půda (X-XI)	14,5	18	15	20	25
Rychlerostoucí dřeviny - zem. půda (XII-II)	12	25-30	8	10	12
Energetické seno - zem. půda (VI;IX)	12	15	2	5	8
Energetické seno - horské louky (VI;IX)	12	15	2	3	4
Rychlerostoucí dřeviny - antropogenní půda (XII-II)	12	25-30	8	10	12
Jednoleté rostliny - antropogenní půda (X-XI)	14,5	18	15	17,5	20
Energetické rostliny - antropogenní půda (X-XII)	15	18	15	20	25

Orientační klíčová čísla pro výhřevnost, výnosy, dobu sklizně a sklizňovou vlhkost energetické fytomasy. Zdroj: VÚRV

3.0. Kotle na biomasu

Moderní kotle na spalování biomasy se již nepodobají jejich předchůdcům, které známe z kotelen na uhlí. Jsou obvykle vybaveny kvalitní elektronikou a ve funkcích vůbec nezaostávají za kotli plynovými. Výrobci nabízejí řadu doplňků, které dále zlepšují jejich funkce (regulátory, speciální armatury apod.).

Volba kotle na spalování biomasy jako jediného zdroje tepla je závislá na dostupnosti paliva a jeho formě. Do kotle je obvykle nutné ručně doplňovat palivo a péče o vytápění objektu se stává povinností a také nutností některého ze členů domácnosti. Doplňování paliva je nutné asi 3x denně v závislosti na jeho druhu a objemu přikládací komory kotle.

3.1. Krby, kamna na dřevo

Pro přímé spalování bez zplyňování se využívají obvykle krby s vhodnou krbovou vložkou, nebo kamna na dřevo. Jako palivo se téměř výhradně využívá palivové dřevo. Čím kvalitnější dřevo spalujeme, tím méně často přikládáme a více se ohřejeme.

Krby se obvykle konstruují jako teplovzdušné topidlo – krbová vložka ohřívá vzduch, který stoupá do místnosti a funguje jako teplovzdušné vytápění. Celá řada krbových vložek a konstrukce krbového tělesa umožňuje pomocí speciálního teplovzdušného rozvodu (vzduchotechnické potrubí) rozvést teplo do více místností a vytápět tak téměř celý dům.



Krbová stavebnice

3.1.1. Krbové vložky

Rozdělují se na:

- jednoplášťové
- dvouplášťové

Jednoplášťové krbové vložky: Obvykle se jedná o litinovou vložku se žebry na vnějším povrchu pro odvod tepla. Vložka se musí obezdít tak, aby mezi vložkou a obezděním zůstala mezera, kudy proudí ohřívaný vzduch. Obezdění zde nahrazuje „druhý plášť vložky“. Toto obezdění akumuluje část tepla, kterou potom předává do prostoru sáláním a menší část ohřevem vzduchu. Zbytek tepla se předává vzduchem, který je nasáván obvykle pod vložkou, proudí kolem vložky, ohřívá se a vystupuje do místnosti. Zachování požadované mezery mezi obezděním a vložkou je zcela zásadní a musí se dodržet. Plocha otvoru pro nasávání vzduchu a stejně tak výstupní plocha výdechu musí mít minimálně 60 cm2/kW výkonu.

Výhody: akumulace tepla v obezděné části, velká sálavá složka, větší možnost individuální konstrukce, snadná možnost použití pasivního teplovzdušného rozvodu.

Nevýhody: je potřeba věnovat mnohem větší pozornost konstrukci celého tělesa krbu, který tvoří vnější část a použít správné materiály.

Dvouplášťové krbové vložky: Obvykle jde o ocelovou vložku s vnitřním a vnějším pláštěm, mezi nimi je mezera, kudy proudí ohřívaný vzduch. Proti propálení je vložka vyložena žáruvzdorným obkladem. Vnější plášť je relativně chladný a je možné ho přímo obložit například sádrokartonem. Krb této konstrukce především vytápí teplovzdušně, sálavá složka je výrazně menší. Pro uchlazení pláště se obvykle instalují ventilátory, které zajišťují potřebný průtok vzduchu kolem vložky. To zajišťuje její chlazení a optimalizuje množství ohřátého vzduchu.

Výhody: Nucený oběh vzduchu umožňuje krb využít mnohem lépe jako teplovzdušné topidlo, včetně rozvodů teplého vzduchu do vedlejších místností, snazší konstrukce tělesa krbu, rychlejší zátop.

Nevýhody: Menší akumulace tepla, je nutné topit rovnoměrněji; ventilátory se mohou porouchat, což může znamenat velký zásah do krbu při opravě; ventilátory mohou být slyšet; ocelová vložka má menší životnost; při poruše ventilátorů může dojít k poškození vložky přehřátím.

Na trhu jsou také výrobky mající výše popsané nevýhody buď částečně, nebo zcela vyřešeny, jsou ale obvykle výrazně dražší.

Těsnost vložky: Těsnost vložky je zcela zásadním parametrem – záleží na ní účinnost celého spalovacího procesu. V ideálním případě mají krbová dvířka takovou konstrukci, která umožní dokonalé uzavření a utěsnění. Kvalitní utěsnění je žádoucí na bočních stranách a dole. Horní strana dvířek se nechává úmyslně netěsná (pokud výrobce toto nevyřešil jinak).

Při hoření dřeva na roštu se uvolňují hořlavé plynné složky, které stoupají do horní části vložky, kde je přes horní hranu dvířek přisáván spalovací vzduch. Hořlavé plynné složky ve směsi se vzduchem chytí a dojde k dokonalému vyhoření paliva. V horní části krbu je pak vidět typické modré zabarvení plamene, což je právě důsledek hoření plynných složek.

V případě netěsnosti vložky (třeba netěsností dvířek) se do vložky dostává spalovací vzduch nekontrolovaně a plynné složky vyhoří jen částečně a zbytek je odtažen komínem. Z tohoto pohledu mohou být problematické posuvná dvířka.

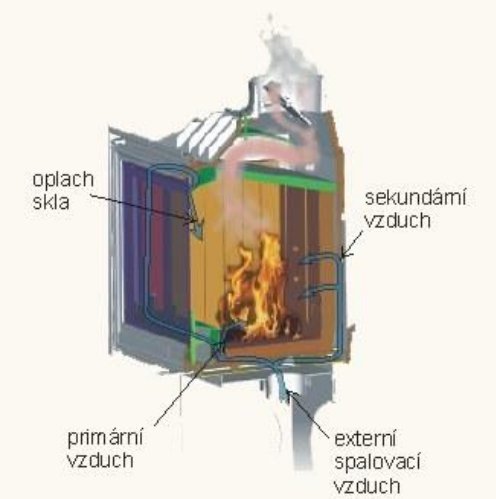


Schéma krbové vložky

3.1.2. Kamna na dřevo

Na trhu je dostatek těchto výrobků. Lze je rozdělit na dvě skupiny:

Klasická kamna na dřevo

Jedná se o kamna, která vytápějí místnost, kde jsou instalovány. Některá jsou zkonstruována jako užitková tj. mají na horní ploše tál, kde je možné vaření jídel a pečení. Základní nevýhodou je to, že není možné obě funkce dokonale oddělit, tj. v letním období vede vaření nebo pečení k nežádoucímu vytápění kuchyně.

Některé výrobky jsou naopak určeny pouze pro vytápění, a to spíše pro občasný pobyt nebo pracovní využití, kde není třeba nějak přesněji regulovat teplotu (rekreační chalupy, skleníky, dílny).

Kamna na dřevo s teplovodním výměníkem

Tato skupina topidel je vybavena teplovodním výměníkem, který lze připojit na etážovou otopnou soustavu a vytápět tak několik místností najednou popřípadě menší nízkoenergetický rodinný domek.

3.1.3. Kachlová kamna

Kachlová kamna jsou dost starý vynález, patrně nejstarší jsou k vidění v Meranu (Itálie) a pocházejí z 15. století.

Oproti krbu mají kachlová kamna četné výhody, mezi ty hlavní patří vyšší účinnost (u moderních kamen dosahuje až 80 %) a schopnost akumulace tepla, což umožní rovnoměrnější vytápění a menší nároky na obsluhu. Slušné účinnosti je u kachlových kamen dosahováno díky tomu, že spaliny jsou do komína vedeny řadou kanálů (tahů), kde odevzdají větší část svého tepla. Také vlastní spalování v topeništi může být účinnější než u klasického krbu, protože topeniště nemá takové tepelné ztráty a lze dobře regulovat přívod primárního i sekundárního vzduchu. V kachlových kamnech lze také vytvořit kanály pro ohřev vzduchu a lze je použít k rozvodu teplého vzduchu do dalších místností domu.



Kachlová kamna

3.1.4. Teplovodní kotle na zplyňování dřeva

Teplovodní kotle na zplyňování dřeva využívají systém spalování za pomoci zplyňování. Jsou velmi dobře regulovatelné (podobně jako plynové kotle prostorovým termostatem) a mohou bez problémů nahradit stávající kotle na tuhá paliva. U těchto zařízení platí dvojnásob, že jejich chod je nejefektivnější v oblasti maximálního výkonu. „Dušení“ kotle v oblasti malého výkonu přináší zhoršení účinnosti a jeho zanášení.

Při instalaci kotle je nutno dodržet další podmínku a to je teplota vratné vody. Ta by neměla klesnout pod 50 - 55 °C. Při nižších teplotách mohou na výměníku kondenzovat agresivní látky, které způsobují jeho korozi. Za kotel se z tohoto důvodu instaluje směšovací armatura, která udržuje požadovanou teplotu zpětného potrubí přimícháváním nahřáté topné vody ze stoupacího potrubí.

Oba tyto negativní jevy lze velmi elegantně vyřešit instalací akumulační nádrže s příslušnou regulací. Kotel pak nevytápí přímo otopnou soustavu, ale akumulační nádrž, z níž si ústřední topení odebírá přes směšovač právě tolik topné vody, kolik je potřeba – podle pokynů regulace. Kotel pracuje v delších časových intervalech a má proběh na plný výkon tj. při nejvyšší účinnosti. Zvýší se tak efektivita chodu kotle, sníží se spotřeba paliva a není třeba tak často přikládat. Při vyhasnutí kotle je také možné dále vytápět dům z akumulační nádrže.

Výhody: Levné palivo; snadná možnost náhrady jiného teplovodního kotle (třeba na hnědé uhlí); vysoká účinnost spalování; čistý provoz; přijatelné pořizovací náklady; výborná regulovatelnost; není nutné tak často přikládat, minimální produkce popela, který lze použít jako hnojivo.

Nevýhody: Je nutné mít sklad palivového dříví a kvalitní komín; kotel má o něco větší rozměry než klasický kotel na hnědé uhlí; kotle mají obvykle zbytečně velký výkon; u novostaveb je téměř nutné použít akumulační nádrž.



Zplyňovací kotel českého výrobce ATMOS (foto EkoWATT)

3.1.5. Výrobci kotlů na dřevo

Agromechanika Lhenice – vyrábí kotle na kusové dřevo s výkony 18,23 a 29 kW.

ATMOS – vyrábí kotle na kusové dřevo ve výkonové řadě 15,18,22,25,32,40,50,75, a 100kW a dodává k nim nádrže o objemu 500,750 a 1 000 litrů.

ATTACK – kotle ATTACK DP s nominálním výkonem 25,35 a 45 kW jsou určeny pro spalování kusového dřeva na principu generátorového zplyňování s použitím odtahového ventilátoru, který odsává spaliny z kotle .

Benekov – vyrábí kotel BENEKOV pelling 27 s výkonem 7-25 kW určený pro spalování polen s délkou až 400 mm a umožňuje také spalovat biomasu o zrnitosti do 25 mm.

Dakon-má ve výrobním programu řadu litinových teplovodních kotlů o výkonech 20,24,28,32 a 36kW.

Verner – vyrábí kotle na kusové dřevo o výkonech 20,25 a 45 kW. Dodává i kotle v provedení s nerezovým kotlovým tělesem .

Viessmann - Vitolig 150 se vyrábí v řadě 18,25,40,60 a 80 kW,jeho výhodou je velká plnící a spalovací komora ze speciální keramiky , která umožňuje používat i u 18kW kotle polena o délce 50 cm.

3.2. Automatické kotle na dřevěné peletky

Automatické kotle na dřevěné pelety či štěpku představují vrchol automatizace vytápění biomasou. Na trhu je dostatek kotlů na dřevěné peletky různých výkonů pro rodinné domy. Kotle jsou vybaveny vlastním zásobníkem – výrobci obvykle nabízejí několik velikostí zásobníků. Transport paliva ze zásobníku je zcela automatický podle pokynů řídící jednotky. Zásobník se snadno naplní palivem, které je dodáváno v pytlích, případně ve větším balení. V sousedním Rakousku je velmi rozšířené dodávat palivo přímo ze speciální cisterny nákladního automobilu. Palivo se pomocí transportní hadice nafouká „přímo do velkého zásobníku s palivem v domě. Z něj je transportováno dopravníkem do kotle.

Automatické doplňování paliva formou peletek představuje perspektivní alternativu ke spalování zemního plynu nebo zkapalněných plynů. Peletky jsou totiž u nových kotlů dopravovány automaticky malými šneky a tím odpadají zvýšené nároky na obsluhu. Zapalování pelet se děje pomocí elektricky žhavené spirály. Vlastní hoření pelet probíhá zpravidla v poměrně malém hořáku. Spalování je poměrně účinné (udává se kolem 90 %). Hořák na pelety lze také vmontovat do existujícího kotle na tuhá paliva, na trhu jsou i kotle umožňující spalovat vedle pelet i další palivo, např. dřevo.



Automatický kotel na pelety se zásobníkem (foto EkoWATT)

3.2.1. Výrobci kotlů na pelety

Atmos – firma nabízí dva kotle na pelety DP 15 (s výkonem 4,5–15 kW) a DP 20 (s výkonem 6–22 kW), které mají jako hlavní palivo dřevěné pelety ø 6-8 mm (bílé pelety) a jako náhradní palivo suché dřevo ø 70-150 mm.

Benekov – firma nabízí kotel Ling 25, umožňující spalování pelet se zrnitostí 8-20 mm nebo hnědého uhlí ořech 2 se zrnitostí 10-25 mm.

Fröling – kotel Fröling Turbomatic je plně automatický bezobslužný kotel na štěpku, pelety nebo kusové dřevo, ve výkonové řadě od 28 do 110 kW s účinností až 92 %. Je vhodný zejména pro vytápění větších objektů.

I.G.B. Holding – nabízí kotel Biostar 12/15 W pro nízkoenergetické domy nebo pro malé prostory

VIESSMANN – kotel na pelety Vitolig 300 s výkonem v rozsahu 2,9 do 25,9 kW má digitální regulaci a ventilátor s plynule nastavitelnými otáčkami a tím umožňuje optimální nastavení na momentální potřebu tepla.

3.3. Kotle na spalování dřevní štěpky a pilin

Štěpka se zpravidla spaluje spíšen větších kotlích zásobující teplem např. zemědělské farmy nebo v obecních výtopnách. Kotle na štěpku zpravidla také umožňují spalování pilin , existují však i kotle určené přímo ke spalování pilin. Pro kotle velkých výkonů je typické , že mají poměrně složité řídící systémy , které zajišťují optimální spalování v závislosti na složení a vlhkosti paliva.

Kotle na štěpku jsou vhodné pro rodinné domky (malého výkonu) se na našem trhu zatím nevyskytují. Například v Rakousku nebo v Německu jsou již běžně k dostání. Na našem trhu je možné sehnat kotle většího výkonu od cca 50 kW, včetně zásobníku paliva a dalšího příslušenství.



Špičkový automatický kotel na dřevěnou štěpku se zásobníkem (foto EkoWATT)

Výhody: Plně automatický chod;snadná možnost náhrady jiného teplovodního kotle (třeba na hnědé uhlí); vysoká účinnost spalování; čistý provoz; plná regulovatelnost; minimální produkce popela, který lze použít jako hnojivo

Nevýhody: Vyšší pořizovací náklady; dražší palivo; je nutné mít sklad paliva a kvalitní komín; kotel má o výrazně větší rozměry než klasický kotel na hnědé uhlí.

3.3.1. Výrobci kotlů na spalování štěpky a pilin

Kohlbach Holding GmbH (Rakousko) – vyrábí kotle v rozsahu výkonů od 300 kW do 10 MW. Kromě instalace do kotelen firma nabízí i tzv. kontejnerové výtopny s výkony od 300 kW do 1,6 MW, které lze převést na místo potřeby a z modulů sestavit a zprovoznit během několika dnů.

VERNER – kotle GOLEM o výkonu od 90 kW do 2,5 MW jsou určeny k vytápění a ohřevu teplé vody nebo k výrobě páry. Mohou spalovat dřevní hmotu ve formě pilin (vlhkost max.35%) , dřevní štěpky nebo zelené lesní štěpky (vlhkost max.50%) o rozměrech do 30x30x60 mm.

STEP Trutnov – teplovodní a parní kotle STEP-KB pro spalování biomasy a dřevní hmoty se dodávají ve výkonech od 0,4 do 5 MW a slouží k ohřevu vody pro vytápění z centrálních kotelen nebo ve velkých budovách.

3.4. Kotle na slámu

S využitím slámy pro energetické účely začali zemědělci. Šlo o spalovací zařízení menších výkonů , kolem 50 až 100 kW, pro potřeby rodinných farem.Tvar spalovací komory byl přizpůsoben balíkům slámy . Původní kotle byly na ruční přikládání, lepší kotle měly zásobník balíků. Pozdější kotle byly ve tvaru velkých balíků s přikládáním traktorovými vidlemi nebo vysokozdvižnými vozíky. V poslední době jsou na trhu kotle nové s řízeným spalováním.

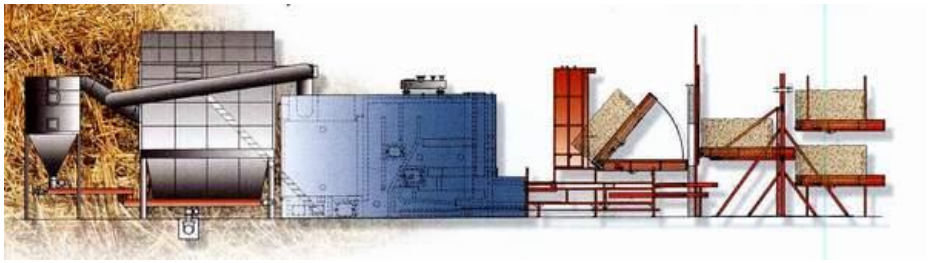


Schéma zařízení pro spalování balíků slámy

3.4.1. Výrobci kotlů na spalování slámy

TTS eko s.r.o. – má ve svém programu kotle na balíky o velikosti šířka 1,2 m, výška 1,2 m, délka 2,2 až 2,5 m a hmotnost balíků 200 až 350 kg.

STEP Trutnov – kotle STEP-KS 175 s výkonem 600 kW jsou určeny na hranaté balíky nebo STEP EKOPALRM s výkonem 0,125 až 600 kW s ručním přikládáním.

VERNER – nabízí automatický kotel A25 na spalování obilí, kukuřice a pelet

LIN-KA Maskinfabrik – vyrábějící průmyslové teplovodní a horkovodní kotle pro spalování slámy o výkonu 100-5000 kW.

Tractant Fabri Josef Novák – firma vyrábí řadu kotlů na spalování biomasy. V nabídce jsou kotle TFS100 a TFS175 na spalování celých balíků slámy.

3.5. Exkurze v EKOEFEKTU s.r.o. v Třebívlicích u Lovosic

Sami jsme navštívili firmu EKOEFEKT s.r.o., která vyrábí kotle na spalování biomasy.

Společnost vznikla v roce 1992 a od počátku se zaměřila na ekologické spalování hnědého uhlí. V prvních letech to byl vývoj a výroba fluidních uhelných kotlů. Firma vyvíjí a vyrábí teplovodní automatické ekologické kotle spalující hnědé uhlí (ekologicky, čistě as komfortní obsluhou) o různých výkonech pro rodinné domy, penziony, průmyslové objekty, pro centrální výtopny, i pro mobilní záložní zdroje. V roce 2001 byly rozšířeny výrobní kapacity o novou společnost Ekoefekt kotle s.r.o. v Třebívlicích u Lovosic. V roce 2008 byla většina výrobních kapacit přesunuta do nových prostor v Třebívlicích. Část technologického vybavení výrobní haly bylo pořízeno za finanční podpory programu SAPARD EU. Společnost využívá systém řízení jakosti ISO 9001.

V roce 1999 vyvinula kotle na spalování biomasy resp. na směsné spalování dřevěné štěpky a hnědého uhlí. Nejvýznamějším projektem v této oblasti byla teplofikace obce Staré- Třebívlice u Lovosic, který získal ocenění v celostátní soutěži Teplárenského sdružení ČR – Projekt roku 2002. V roce 2003 byla uvedena do provozu další kotelna na směsné spalování biomasy a uhlí v obci Kuřivody – Ralsko o výkonu 900 kW. V dalších letech se každoročně realizuje několik projektů centrálního vytápění s použitím kotlů EKOEFEKT BIO. V roce 2007 byl uveden na trh nový kotel EKOEFEKT BIO 23 s retortovým hořákem, který dosahuje účinnosti až 91% a je určen zejména pro rodinné domy.

Zde jsou fotografie z naší návštěvy ve firmě:



Kotel při výrobě



Zásobník na palivo



Kotel před dokončením



Řez kotlem



Sklad vyrobených kotlů

4.0. Vytápění biomasou v praxi

Perspektivní do budoucna se jeví rozvoj využití biomasy pro vytápění a ohřev. Ministerstvo životního prostředí vyhlašuje státní program na podporu úspor energie a využití obnovitelných zdrojů energie. V rámci tohoto programu mohou např. domácnosti získat dotaci na náklady spojené s instalací kotle na biomasu ve výši 50 % z celkových nákladů. Majitelé rodinných domků si musí celý projekt financovat z vlastních zdrojů nebo úvěrem a až zpětně žádat o dotaci z národního programu. Přesto však nemá žadatel záruku, že žádost o dotaci mu bude schválena. Samotné podmínky pro náležité podání žádosti ovšem nejsou nijak drastické. Nejdůležitější je instalovat zařízení doložené certifikátem, resp. protokolem o zkoušce typu dle nařízení vlády č.163/2002 Sb., §5, resp. §7, doložit dodavatelské smlouvy a faktury, příslušné náležitosti ohledně stavebního řízení, případně kolaudační rozhodnutí. V neposlední řadě odborný posudek a v některých případech i energetický audit. V každém případě je nejlepší žádost a její náležitosti konzultovat s příslušným krajským pracovištěm Státní správy životního prostředí, která má příslušné programy na starosti.

4.1. První kotelna na spalování biomasy v ČR

První kotelna na spalování biomasy v České republice je v Dešné. Zkušební provoz byl zahájen 1.10. 1997, předání do trvalého provozu 16.3. 1998. Tuto stavbu realizovali Čeští dodavatelé (Stavcent Jindřichův Hradec a.s.) a byla zde použitá česká technologie (STEP Trutnov a.s., Verner s.r.o. Č. Kostelec). Spalují se dva druhy paliva. Základním palivem jsou všechny druhy obilní a řepkové slámy, doplňkovým palivem je dřevní odpad.

K rozhodnutí zřídit kotelnu vedly obce tyto skutečnosti:

Území obce Deštná má v okrese nejlepší podmínky pro zemědělství a velkou výměru zemědělských ploch, celkem 3088 ha. Nejvíce jsou v místě pěstovány obiloviny (pro krytí energetických potřeb obce postačuje malá část produkce slámy, také místní pila produkuje dřevní odpad.

Celková cena stavby byla 38,5 milionu Kč, kotelna a technická část 21 milionu Kč. Obec dostala dotaci SFŽPČR 40 %, půjčka SFŽPČR byla také 40 % a z vlastních zdrojů financovala 20 %.

Ve zkušebním provozu se vyráběl 1 GJ finančním nákladem 116 Kč.

4.2. Další příklady kotelen na vytápění biomasou v ČR

4.2.1. Městys Nová Cerekev

12.11. 2002 byla uvedena do provozu kotelna se skladem na 3800 prm paliva. V kotelně byl osazen jeden kotel rakouského výrobce značky KOHLBACH o výkonu 2 MW. Bylo osazeno 75 výměníkových stanic, mezi které patří ty největší – obecní úřad, škola, školka, kulturní dům, tři výrobní objekty a tři

devíti bytové domy. Zbytek tvoří rodinné domy. Výměníkové stanice mohou též ohřívat užitkovou vodu.

Prostřednictvím Státního fondu životního prostředí obec obdržela 75 % dotace, 5 % bezúročné půjčky a 20 % financovala obec ze svých zdrojů. Celková částka investice byla 57,5 milionu Kč.

K tomuto druhu topení se obec rozhodla, protože vlastní 450 ha lesa a vlastní obecní pilu, kde vzniká ročně 8 tisíc prm m3 pilin a zhruba stejné množství bočního odpadu.



Kotelna na biomasu v Nové Cerekvi



Kotelna na biomasu v Nové Cerekvi

4.2.2. Kotelna na biomasu v Plané u Mariánských lázní

Dne 16. 3. 2007 byla v Plané u Mariánských lázní otevřena pro veřejnost nová kotelna na biomasu. Ta bude sloužit 2560 obyvatelům města Planá jako zdroj tepla a k výrobě teplé užitkové vody. Na investici ve výši 41,4 milionu Kč se podílely ze 75 % evropské fondy (Operační program Infrastruktura pro životní prostředí), z 15 % samotné město Planá a ze zbylých 10 % pomohl Státní fond životního prostředí. Nová kotelna o celkovém maximálním výkonu 4,8 MW sestává ze tří kotlů. Dosavadní plynový kotel o výkonu 1,4 MW byl rekonstruován a bude sloužit jako záložní. Dva nové kotle o výkonu 1,7 MW spalují biomasu.

4.2.3. Kotelna na biomasu ve Vimperku

6. 2. 2007 ve Vimperku se dočkala slavnostního spuštění nová kotelna na biomasu. Vznikla ve staré uhelné kotelně u čistírny odpadních vod a městské služby stála zhruba 24,5 milionu Kč. Kotelna spalující biomasu, tedy štěpku, vznikla v poměrně krátkém čase ve staré kotelně, která topila uhlím a byl tu záložní plynový kotel. Městské služby její vybudování zaplatily z úvěru, protože dotace se jim získat nepodařilo. Výběrové řízení se uskutečnilo loni v červenci a pak bylo nutné ještě celý projekt zlevnit – cena vzešlá z výběrového řízení byla pro městské služby příliš vysoká. Kotelna má výkon 3,0 MW a je zde použita technologie třebíčské firmy TTS – kotel Vesko-B.

4.3. Návštěva kotelny na biomasu v Třebívlicích

Sami jsme navštívili kotelnu v Třebívlicích v obecní části Staré. Zkušební provoz této kotelny využívající k topení biomasou byl zahájen 21. Ledna 2002. Ve dvou kotlích (kotle CARBOROBOT – B o výkonu 380 kW nyní EKOEFEKT – BIO na dřevěnou štěpku) se spalují štěpky z dřevin smíchané s hnědým uhlím. Tato kotelna původně vytápěla 23 rodinných domů.

Realizace projektu stála 5 milionů korun, z toho zaplatila třebívlická radnice zhruba dvě třetiny a zbytek uhradil státní fond životního prostředí. Na nákup speciálního štěpkovacího zařízení na přípravu biomasy získala obec peníze z fondu Phare. Projekt získal ocenění v celostátní soutěži Teplárenského sdružení v ČR – Projekt roku 2002.

V současnosti je na vytápění napojeno 18 domácností, z nichž se vytápí 14. Měsíční zálohy jsou 1550,- Kč s minimálním doplatkem. Zařízení obsluhuje jeden člověk, který chodí 1x denně doplnit palivo a zkontrolovat stav zařízení. Jednou za 10 dní se provádí celkové vyčištění topné soustavy.

Pořízené fotografie při návštěvě kotelny:



Kotel EKOEFEKT-BIO na dřevěnou štěpku se zásobníkem.



Kotel CARBOROBOT, který se doplňuje ručně.



Rozvody k výše uvedeným kotlům.



Skladování štěpky (paliva).



Odpad z kotelny za dva roky.

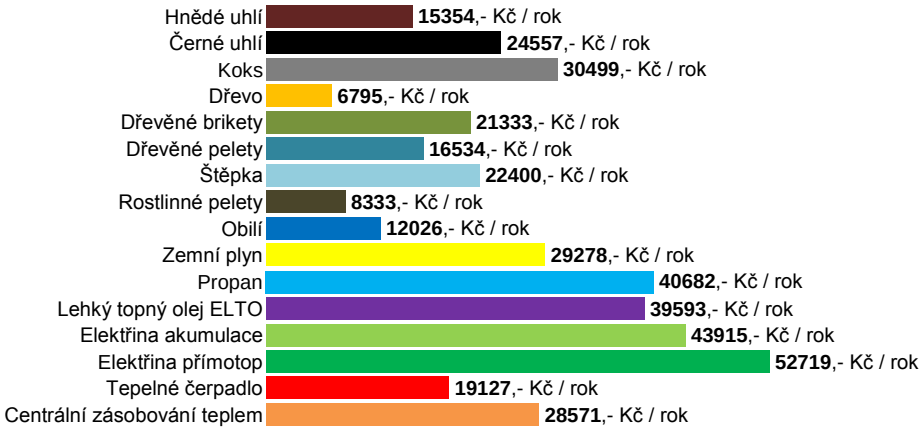
5.0. Závěr

Touto prací jsme chtěli ukázat možnosti a výhody topení biomasou. Rozhodování majitelů nemovitostí pro to, jaké vytápění zvolit není tak úplně jednoduché. Pro rozhodnutí je třeba zvážit různé parametry:

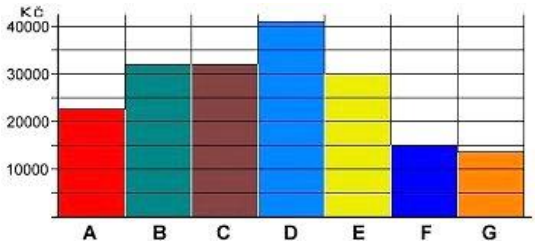
1. Potřebný tepelný výkon – kotle na tuhá paliva se dají regulovat jen do určitého minimálního výkonu, nejmenší výkon mohou dosáhnout topidla na pelety naproti tomu kotle na štěpku nebo na balíky slámy se pro malé výkony vůbec nedají použít.

2. Investiční náklady na zařízení a cena paliva –biomasa není jedinou možností jak vytápět dům (na trhu je elektřina, tepelná čerpadla, zemní plyn nebo uhlí). Významné pro rozhodování jsou celkové roční náklady do nichž je zahrnuta jak cena paliva tak i odpisy (odpovídající část nákladů na koupi a údržbu vytápěcího zařízení). Obecně platí, že pokud je palivo levné, mohou být náklady na zařízení vyšší a naopak.

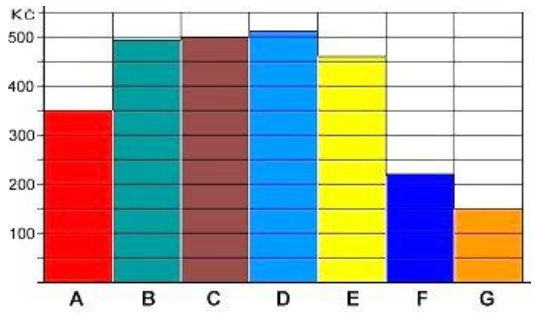
Náklady na vytápění (výpočtová spotřeba tepla = 80 GJ):



A - zemní plyn, B - elektřina, C - lehké topné oleje, D - propan, E - tepelné čerpadlo, F - uhlí, G - dřevo



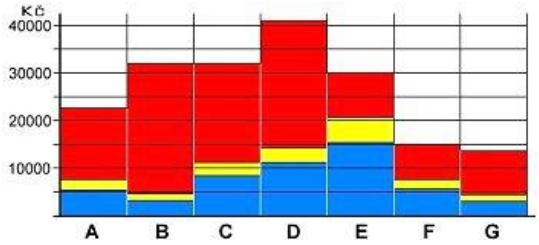
Celkové roční náklady na vytápění



Cena za 1 GJ (278 kWh) tepla podle druhu paliva

Jednotlivé složky celkových ročních nákladů na vytápění
Červená barva - cena paliva
Žlutá barva - provoz a údržba
Modrá barva - investiční složka

Cena za 1 GJ (278 kWh) tepla podle druhu paliva



Jednotlivé složky celkových ročních nákladů na vytápění

3. Dostupnost paliva (zdroje energie) – nejvíce dostupná je zpravidla elektřina. V případě biomasy mohou být v dostupnosti jejich jednotlivých forem poměrně velké lokální rozdíly.

4. Požadovaný komfort a nároky na obsluhu – topení v krbových kamnech vyžaduje poměrně časté přikládání a kontrolu, naproti tomu elektrické nebo plynové vytápění zajistí vysoký stupeň tepelné pohody zcela automaticky.Většina vytápěcích zařízení na biomasu leží někde mezi těmito extrémy.

Zde nabízíme několik možností , jak se dá řešit vytápění různých typů domů :

Nízkoenergetický dům – pro vytápění domu, jehož tepelná ztráta je vypočtena 5-10 kW, přichází v úvahu kamna nebo kotel na pelety. Využijeme jejich hlavní výhody, kterou je dobrá regulace, malé rozměry a minimum práce spojené s obsluhou a obstarání paliva. Vyšší cena pelet nám nijak nevadí, protože roční spotřeba je relativně malá. Zde konkuruje peletám hlavně elektrické vytápění, které má sice větší náklady na energii, ale nižší náklady na vlastní topné zařízení. Také zde lze dobře uplatnit kachlová kamna se vzduchovými kanály pro rozvod tepla do dalších místností.

Běžný rodinný dům – tepelná ztráta mezi 10 a 20kW. Zde se dá také dobře použít kotel na pelety, cena pelet je už důležitá v nákladech na vytápění. Pokud je v místě dostupné palivové dřevo, máme místo na skladování, pak je dobrou volbou kotel na kusové dřevo vybavený akumulační nádrží. Možnou konkurencí je zde tepelné čerpadlo, které má vyšší investiční náklady, ale srovnatelnou nebo nižší cenu za kWh tepla a nevyžaduje žádnou obsluhu.

Větší venkovský dům nebo zemědělská usedlost – tepelná ztráta mezi 20-60 kW v tomto případě je kotel na kusové dřevo dobrou volbou, lze uvažovat i o kotli na spalování štěpky. Další možností je centrální výtopna na štěpku či slámu pro celou vesnici nebo využívání tepla z kogenerační jednotky na bioplyn.

Občas obývaná rekreační chalupa - v tomto domě je důležité po příjezdu do silně vychladlého domu dosáhnout co nejrychleji přijatelné tepelné pohody. Pro tento účel jsou výhodná krbová kamna, která díky významnému podílu sálavého tepla dokážou kompenzovat negativní efekt silně prochladlých stěn. Pro tento typ domů jsou také oblíbená kachlová kamna. Lze použít i topidlo na pelety a pomocí dálkového ovládání nebo programového regulátoru jej využít k temperování domu a k vyhřátí před příjezdem obyvatel.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Annaburg

Jan Filipovič, Martin Királ
Střední odborná škola a Střední odborné učiliště
Littrowa 122, 346 01 Horšovský Týn

Do soutěže pro mladé začínající projektanty jsme se přihlásili za účelem získání nových zkušeností a zdokonalení již získaných znalostí. Jako soutěžní úkol jsme se rozhodli vyprojektovat lovecký zámeček Annaburg, ležící v oboře nedaleko obce Horšov na Domažlicku. Tento zámeček se nacházel v dezolátním stavu.

Z původní stavby zůstaly pouze části opěrných zdí. Z nalezených materiálů se nám podařilo vyprojektovat rekonstrukci, která byla provedena ve spolupráci s památkáři. Na rekonstrukci se finančně podílela naše škola, která čerpala z evropských strukturálních fondů. Ze získaných prostředků probíhá rekonstrukce, kterou vykonávají učni oboru zedník. Složitý krov dodala specializovaná firma, která se postarala o jeho celkovou montáž. Ta proběhla nejprve na specializovaném pracovišti a poté již na samotném místě. Krytinou byly po dohodě s památkáři zvoleny dřevěné došky. Práce dále pokračovaly úpravou terénu vně zámečku. Ten na začátku oprav byl v nejhorším stavu. Nalézaly se zde kořeny stromů ležících v bezprostřední blízkosti objektu. Ty byly odstraněny s ohledem na statiku opěrných zdí. Po upravení již očištěné plochy začaly práce na vlastní podlaze. Ta bude složena z pálených cihel uložených v pískovém loži. V další fázi se budou opravovat štukové omítky a rekonstruovat nástěnné malby, které zde byly objeveny. Poté bude upraven okolní terén a tím bude rekonstrukce dokončena. Studenti podílející se na projektu rekonstrukce se zapojili také do samotných oprav a těší se až bude jejich dílo dokončeno.

V soutěži předvedeme model loveckého zámečku s dokumentací zhotovenou pomocí programu ALPLAN a prezentaci v MS PowerPoint.

tento zrekonstruovaný objekt SE stane plánovaného historicko environmentálního projektu naučné stezky zámeckou oborou žáků školy a poutním místem mnoha turistů navštěvujících Horšovský Týn a okolí.



Středoškolská technika 2008

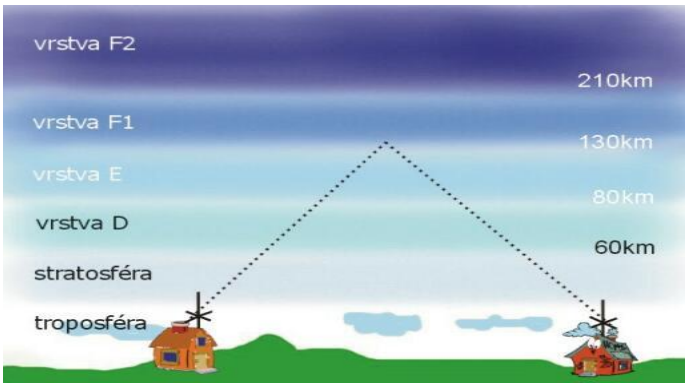
Setkání a prezentace prací
středoškolských studentů na ČVUT

ATMOSFÉRICKÉ JEVY

Vojtěch Umlauf, Václav Trunec, Adam Hák

První soukromé jazykové gymnázium
Brandlova 875, Hradec Králové

Tento projekt pojednává o dějích ve vzdušném obalu Země-atmosféře. Atmosféra je plynný obal tělesa v kosmickém prostoru. Atmosféra se skládá více vrstev. Pro děje způsobující počasí je důležitá spodní vrstva atmosféry, která se nazývá troposféra. V naší zeměpisné šířce je v průměru 11 km silná. V troposféře jsou přibližně tři čtvrtiny hmotnosti celé atmosféry a téměř všechna voda. Odehrávají se v ní meteorologické děje. Vznikají zde mlhy, nejrůznější druhy oblaků, bouřky, atmosférické srážky. Vyměňuje se zde teplo. Neustále zde proudí vzduch a promíchává se.



Obr. 1: Výškové rozložení zemské atmosféry

V práci se zabýváme základními meteorologickými termíny. Pro předpověď počasí je důležitý především tlak vzduchu, který ovlivňuje vznik tlakových níží a výší, dále pak teplota, vlhkost a proudění vzduchu.
Dále jsme se zabývali popisem atmosférických jevů a jejich intenzity. Mezi ně patří hydrometeory - meteory tvořené soustavou vodních částic ve stavu kapalném nebo tuhém, padajících nebo vznášejících se v atmosféře nebo zdvižených větrem z povrchu země, nebo usazených na předmětech na zemi či ve volné atmosféře. Mezi nejznámější hydrometeory patří déšť, sněžení, kroupy, jinovatka, mlha a mnoho dalších.
Mezi další známé meteory patří fotometeory (duha) a elektrometeory (bouřka).

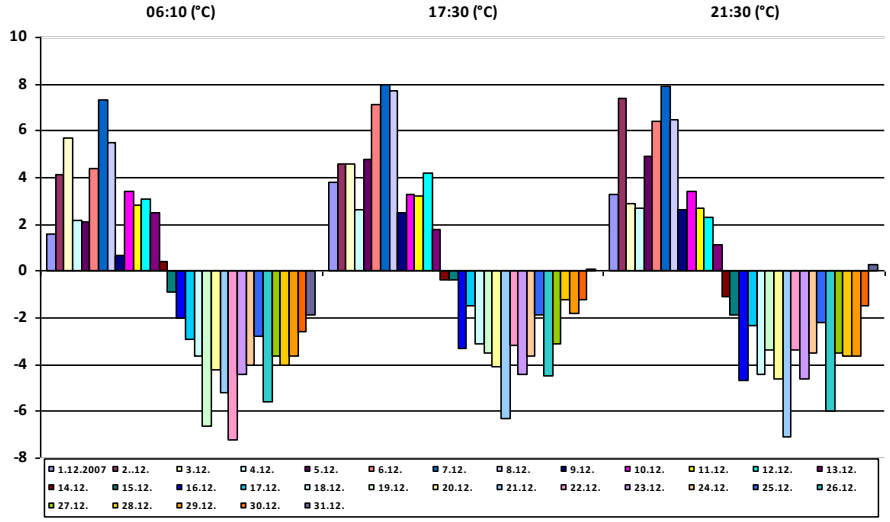
Druhá část práce se zabývá přístroji pro různá meteorologická měření a předpovědí pravděpodobnosti počasí. Mezi základní přístroje patří teploměr, vlhkoměr, tlakoměr (barometr), srážkoměr a anemometr. V práci je popsáno, jak tyto přístroje fungují.

Třetí část tohoto projektu je věnována praktickým údajům vlastního pozorování. V této části práce jsme zaznamenávali teploty vzduchu a stav oblačnosti v prosinci 2007. Získané údaje jsme zaznamenali do tabulky a formou grafu.

Tabulka č.1: Teplota vzduchu a oblačnost zaznamenané v prosinci 2007.

DATUM	06:10 (°C)	17:30 (°C)	21:30 (°C)	OBLAČNOST
1.12.2007	1,6	3,8	3,3	oblačno
2..12.	4,1	4,6	7,4	oblačno,večer déšť
3.12.	5,7	4,6	2,9	zataženo,dešť.přeháňky
4.12.	2,2	2,6	2,7	oblačno,polojasno,déšť
5.12.	2,1	4,8	4,9	oblačno,dešť.přeháňky
6.12.	4,4	7,1	6,4	oblačno,dešť.přeháňky
7.12.	7,3	8,0	7,9	oblačno
8.12.	5,5	7,7	6,5	oblačno
9.12.	0,7	2,5	2,6	déšť,déšť se sněhem
10.12.	3,4	3,3	3,4	oblačno,polojasno,dešť. přeháňky
11.12.	2,8	3,2	2,7	oblačno
12.12.	3,1	4,2	2,3	zataženo
13.12.	2,5	1,8	1,1	polojasno
14.12.	0,4	-0,4	-1,1	zataženo,občasné sněžení
15.12.	-0,9	-0,4	-1,9	oblačno,později jasno
16.12.	-2,0	-3,3	-4,7	oblačno,později jasno
17.12.	-2,9	-1,5	-2,3	oblačno
18.12.	-3,6	-3,1	-4,4	oblačno,večer jasno
19.12.	-6,6	-3,5	-3,4	oblačno
20.12.	-4,2	-4,1	-4,6	zataženo,občasné sněžení
21.12.	-5,2	-6,3	-7,1	zataženo
22.12.	-7,2	-3,2	-3,4	zataženo
23.12.	-4,4	-4,4	-4,6	zataženo
24.12.	-4,0	-3,6	-3,5	zataženo
25.12.	-2,8	-1,9	-2,2	zataženo
26.12.	-5,6	-4,5	-6,0	skoro jasno
27.12.	-3,6	-3,1	-3,5	zataženo
28.12.	-4,0	-1,2	-3,6	polojasno
29.12.	-3,6	-1,8	-3,6	polojasno
30.12.	-2,6	-1,2	-1,5	zataženo,sněžení
31.12.	-1,9	0,1	0,3	zataženo,občasné sněžení

Graf č.1: Teplota vzduchu zaznamenaná v prosinci 2007.



V měření teplot pokračujeme i nadále a naším cílem je zaměřit se na porovnání naměřených data s daty získanými v dřívějších obdobích.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

ROZKLAD SUROVIN A SPRÁVNÉ TŘÍDĚNÍ ODPADŮ

Anna Vogelová, Kateřina Kněžáčková

První soukromé jazykové gymnázium

Brandlova 875, Hradec Králové 3

Cílem našeho projektu bylo postihnout problematiku odpadů. Celou naši planetu pohlcují věci, které jsou zničené nebo věci, které člověk již nepotřebuje. Bohužel se některých nelze tak jednoduše zbavit.

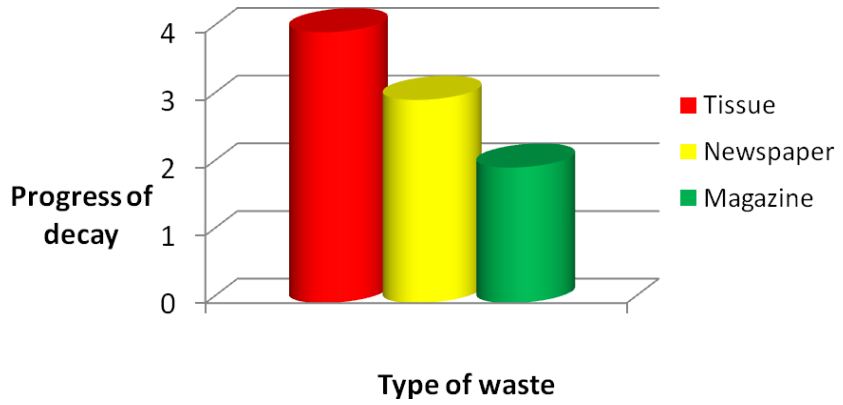
Snažili jsme se vyhledat a shromáždit co nejvíce informací o odpadech obecně jako například: Co můžeme považovat za odpad, jak se odpady dělí, kam by se odpady měli ukládat a jak je správně třídit. Dále jsme se zabývali otázku bioodpadů, likvidace a recyklace odpadů, kde upřesňujeme i druhy odpadů (plast, sklo, papír,…). Náš projekt přibližuje a objasňuje pojmy, jako jsou separace, pyrolýza a spalování odpadů, kompostování, biotechnologie, vitrifikace či skladování.

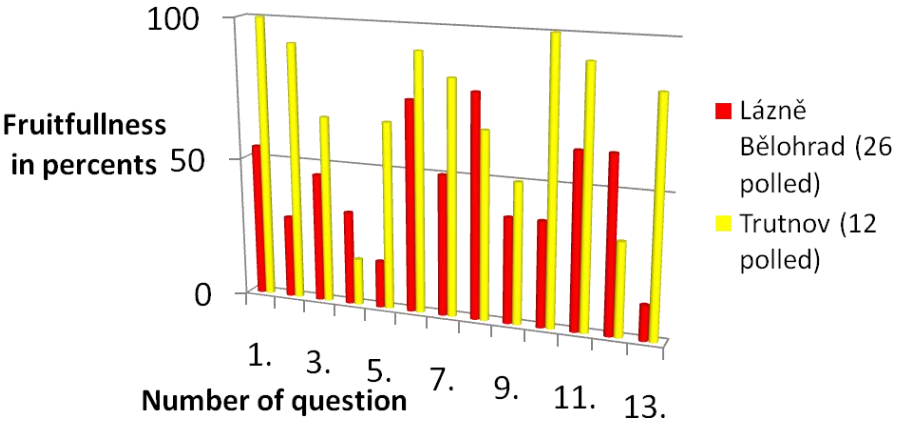
V rámci vlastního výzkumu jsme sestavili dotazník, jenž poukazuje na informovanost problematiky třídění odpadů na školách a mezi občany ČR. Získané výsledky byly zpracovány a data zaznamenány do grafů. Tento dotazník nebyl jediným předmětem vlastního výzkumu. Téměř každý člověk někdy pohodil nějaký odpad do přírody či na ulici, nikoli však do odpadkového koše, proto jsme zkoumali dobu rozkladu několika surovin (např. papírový kapesník, skleněná láhev, jablko) jak za aerobních, tak i anaerobních podmínek. Výzkum započal v říjnu minulého roku a skončil na konci ledna roku 2008. V této době jsme každou surovinu po jednotlivých intervalech kontrolovali a následně fotografovali. Získané informace jsme opět zaznamenali do grafů. V již zmiňovaném dotazníku jsme se ptali na otázky, které jsme sestavili na základě tohoto výzkumu.

Při práci na tomto projektu jsme si ještě více uvědomili, jak důležité je třídit odpad a věnovat se této problematice do hloubky. Pohazování a špatné či žádné třídění odpadů škodí nejen estetickému vzhledu lidmi obývaných míst, ale i životnímu prostředí . V dnešní průmyslově rozvojové době, kdy tají ledovce v důsledku globálního oteplování, narůstá počet ozonových děr a stále více se zhoršuje ovzduší, by člověk měl o přírodu více pečovat a cenit si jí. Během zpracování a vyhodnocování dotazníků, jsme byli velmi často překvapeny. Mnoho lidí nemá o problematice odpadů dostatek informací nebo k ní přistupuje s velkou apatií, což nejsou moc

dobré předpoklady. Proto bychom se všichni měli zamyslet a položit si pár otázek: Zajímá vás tato problematika? Víte, jak se správně odpad třídí a kam se ukládá?

Obr.1: Příklad zpracovaných dat o pokročilosti rozkladu surovin do grafu





Obr. 2: Příklad vyhodnocených dotazníků

Tabulka 1: Dotazník se správnými odpověďmi.

Do you sort waste?	
What color has container for paper?	Blue
How long does it take to analyse plastic?	100-200 years
What does it decay the longest time?	Plastic
How many waste produces inhabitant of Czech Republic on the average?	150 – 200 kg/year
How long does it takes to decays apple?	3-4 months
How many percent inhabitants of Czech Republic sort all the kinds of wastes?	22%
Does wax-paper belongs to container for paper?	No
How many kilograms of paper spend man on the average?	85 kg/year
Do ceramic, porcelain and mirror belong to container for glass?	No
Belong shell of eggs and nuts to container for bio-waste?	Yes
Belongs foam to container for plastics?	No
Is recycling never-ending process?	No

Obr. 3: Příklad pozorované suroviny: jablko v písku.



1st day **31st day** **120th day**

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Meteorologie

Ondřej Tomalík, Jitka Tobišková, Petra Helligerová, Jan Sláma, Marek Draštík

První soukromé jazykové gymnázium v Hradci Králové
Brandlova 875, Hradec Králové

Cílem našeho projektu bylo porovnání údajů naměřených pomocí volně prodejné digitální meteorologické stanice a údajů získaných z jiných zdrojů a naměřených jinými druhy stanic v různých lokalitách České republiky. Především také krátkodobá předpověď počasí.

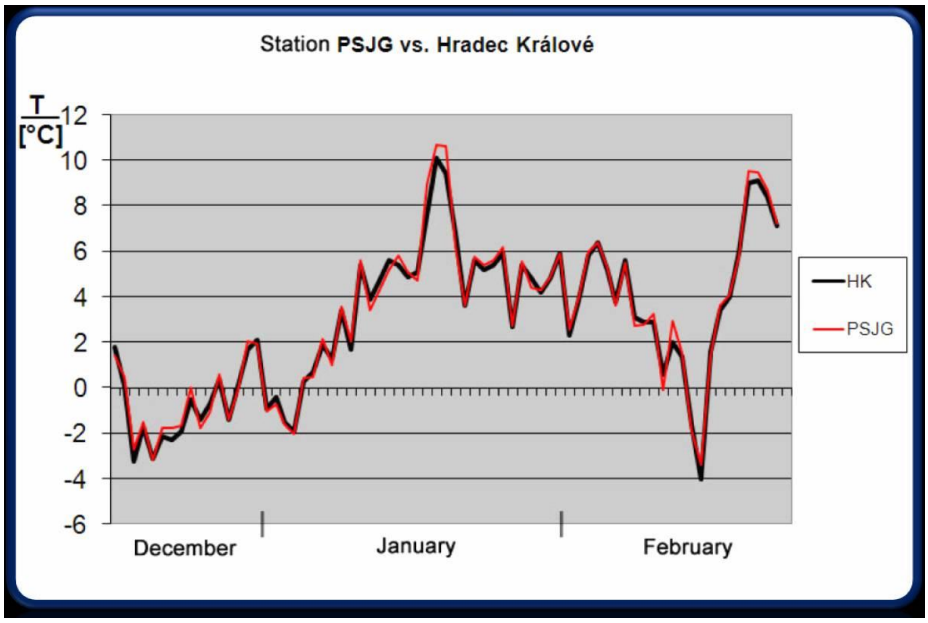
Abychom mohli s touto prací začít, museli jsme nejdříve vybrat a sestrojit meteorologickou stanici. V našem výzkumu nás podpořila škola, která námi vybranou stanici zakoupila.
Pro začátek samotné práce jsme se zkontaktovali s pobočkou ČHMÚ, kde jsme se dozvěděli zajímavé informace ohledně meteorologie a také o jejím vývoji.
Jakmile jsme dokončili práce s výstavbou meteorologické stanice, mohli jsme začít s měřením, a to v polovině prosince loňského roku. Nejprve jsme museli ověřit přesnost měření naší stanice a k tomu nám byly poskytnuty data ze stanice v Hradci Králové. Jakmile jsme zkontrolovali správnou funkčnost, mohli jsme s jistotou pokračovat s měřením.
V porovnávání jsme pokračovali i nadále. Kontaktovali jsme další města s meteorologickými stanicemi. Mohli jsme tak porovnávat především námi naměřené teploty s hodnotami jiných, výše položených lokalit. Čímž jsme dokázali, že teplota klesá se stoupající nadmořskou výškou.
Hlavním cílem našeho projektu byla krátkodobá předpověď počasí podle změny tlaku vzduchu. Naše předpověď se téměř vždy shodovala s nastávajícím počasím. Ze všech těchto dat jsme sestrojili grafy, na nichž lze hodnoty lépe vidět.

Celkově jsme na projektu pracovali od září roku 2007 do února 2008. Za tu dobu jsme mohli ověřit přesnost stanice a potvrdit veškeré předpovědi měřením. V měření nadále pokračujeme a jsme již v kontaktu se Švýcarským institutem meteorologie, který nám poskytl kontakt na školy s meteostanicemi. Věříme, že náš projekt nezůstane nevyužit a v budoucnu někteří studenti naváží na naši práci či použijí naše poznatky ve svých pracích.

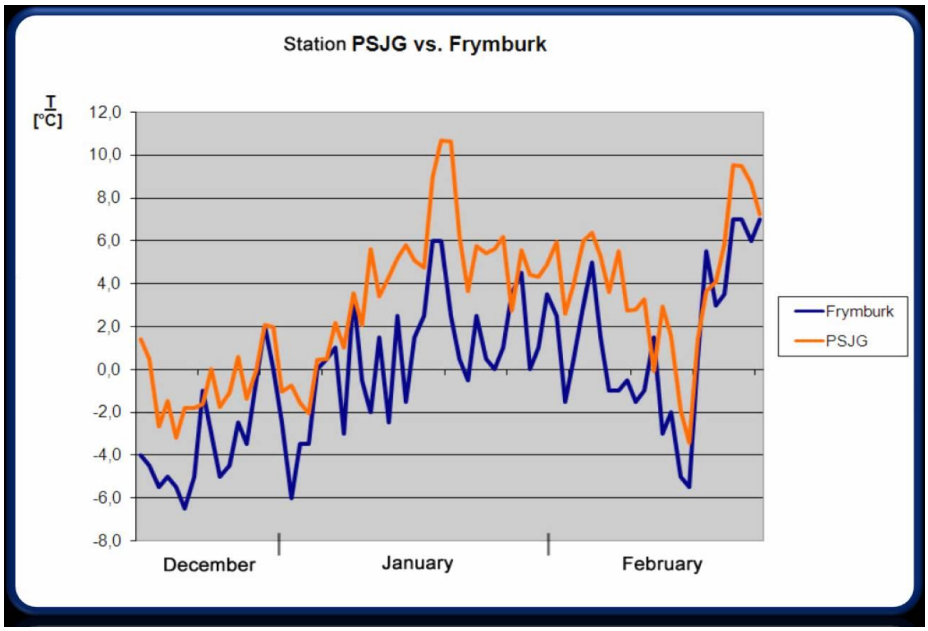


Obr. 1: Stanice umístěná v naší škole

Graf č.1 : Srovnání naměřených hodnot teploty PSJG vs. Hradci Králové



Graf č.2 : Srovnání naměřených hodnot teploty PSJG vs. Nymburk



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

DIGITÁLNÍ ZVUKOVÝ PROCESOR

Vladimír Burian

Gymnázium Jihlava , Jana Masaryka 1, Jihlava

ZADÁNÍ A SPECIFIKACE

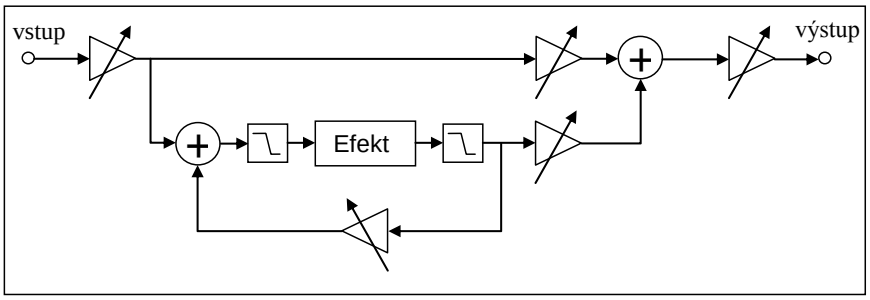
Prvotním cílem pro mě bylo navrhnout, zkonstruovat a zprovoznit přenosné programovatelné zařízení zpracovávající zvuk – přesněji zařízení, které by bylo schopné v reálném čase efektovat zvuk zejména elektronických hudebních nástrojů, např. kytary. Mělo by být obsaženo několik základních efektů, především těch, jejichž podstatou, či rovnou výsadou je relativně delší zpoždění – takové efekty nejsou náročné na numerické výpočty a lze jich těžko dosáhnout čistě analogovou cestou. Samozřejmostí je požadavek na jednoduché ale účinné uživatelské rozhraní.

Zařízení si neklade nárok konkurovat i těm levnějším, masově vyráběným multiefektům, ale mělo by být s přimhouřeným okem v praxi použitelné.

Hlavním cílem samotného zařízení tedy není pouze konečný, na začátku velmi nejistý výsledek, ale především cesta k němu. V každém případě tedy bude sloužit ke vzdělání. S tím souvisí další požadavek: Zařízení (přinejmenším některé jeho části) by mělo mít univerzálnější charakter, možnost rozšíření, připojení dalších periferií a nabídnout možnost osvojení si práce s nimi. Stejně tak posloužit i jako zajímavý základ pro úlohy jiného charakteru, kde by se mohla shodovat řídící jednotka s uživatelským rozhraním.

POPIS KONEČNÉHO ZAŘÍZENÍ

Zvuk prochází zařízením následujícím způsobem:



Obr.1: blokové schéma analogové cesty signálu

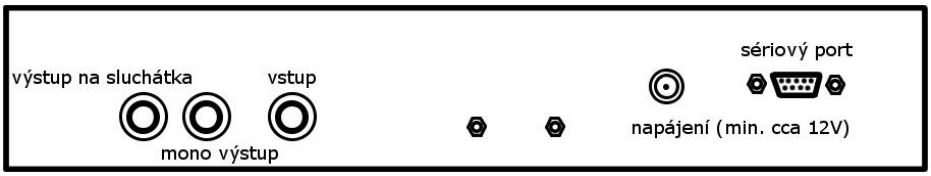
Ihned za vstupem je signál zesílen na potřebnou úroveň nastavenou uživatelem tak, aby byl co nejlépe využit rozsah A/D a D/A převodníků a přitom nedocházelo ke zkreslení – přebuzení

převodníků. Toto nastavení velmi ovlivňuje výslednou kvalitu zvuku a odstup nežádoucích složek signálu vznikajících při převodu.

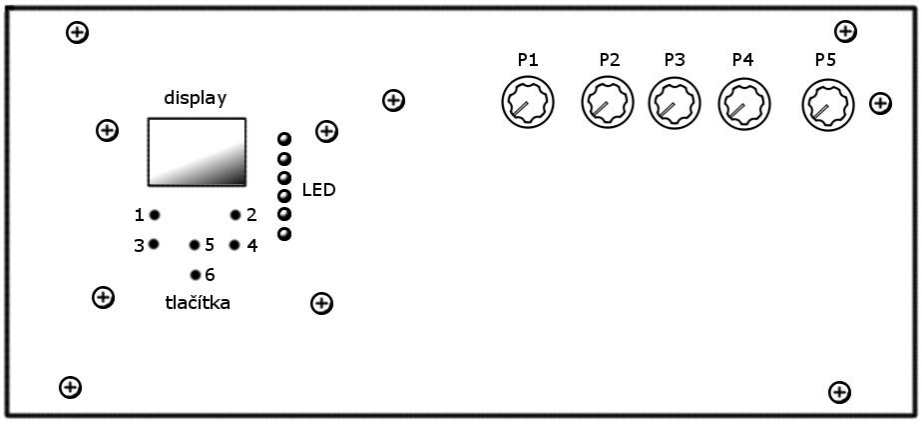
Následně se signál rozděluje do dvou větví. V horní, čisté větvi je neefektovaný signál přiveden k dalšímu zesilovači (prakticky pouze na potenciometr), který určuje jeho úroveň ve výsledném signálu.

Druhá větev se stará o efektovaný signál. Když opomeneme směšovač a následující dolní propust, signál vstoupí do „efektu“ – digitální části přístroje. Následně je „vyčištěn“ další dolní propustí, ze které pokračuje jednak na zesilovač (potenciometr) určující jeho podíl ve výstupu a pak přes jiný zesilovač (potenciometr) zpět na začátek, kde je (již efektovaný signál) přimíchán do signálu teprve vstupujícího do „efektu“. Posledně popsaná část tedy zajišťuje zpětnou vazbu „efektu“, která v praxi slouží k nastavení např. délky ozvěny.

Nakonec jsou oba signály – efektovaný a čistý – smíchány a dále už je pouze řízena celková úroveň výstupního signálu. Na blokovém schématu není zakreslen výkonový výstupní zesilovač.



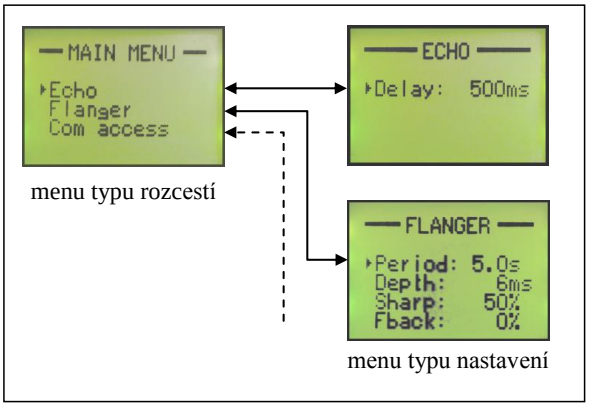
Obr.2: nákres zadního panelu přístroje



Obr.3: pohled na vrchní stranu přístroje

Funkce potenciometrů:

- P1 – zesílení vstupního signálu
- P2 – úroveň čistého signálu
- P3 – úroveň efektovaného signálu
- P4 – zpětná vazba
- P5 – hlasitost



Obr.4: příklad struktury programu

Funkce tlačítek:

- TL1 – menu zpět
- TL2 – menu vpřed
- TL3 – plus
- TL4 – mínus
- TL5 – nahoru
- TL6 – dolu

Rozhraní pro komunikaci programu s uživatelem zahrnuje grafický display, 6 tlačítek a 6 LED diod. Diody slouží primárně k indikaci úrovně signálu se kterým „efekt“ pracuje – což je důležité pro již zmíněné, správné nastavení zesílení vstupního signálu.

Z obr.4 je patné velmi jednoduché, účelné uspořádání nabídek. Základem pro vše je menu, které obsahuje nadpis a jednotlivé položky na zvláštních řádcích. Právě vybraná položka je vlevo indikována šipkou a pro její posun složí tlačítka TL5 a TL6. Položka v rozcestí má funkci jakéhosi odkazu na další menu – rozcestí tedy reaguje také na TL1 a TL2. Pokud se ale dostaneme na menu typu nastavení, znamená to, že zároveň běží jemu odpovídající „efekt“ (programový modul) – položky v tomto případě představují jednotlivé volitelné vlastnosti. Pro jejich změnu slouží TL3 a TL4. TL1 způsobí vypnutí efektu a skok zpět.

Implementovány jsou v současné době pouze dva efekty: echo a flanger.

Základní vlastnosti:

- vzorkovací frekvence: 27 027 – 29 585 Hz
- rozlišení: 12-bitové, mono
- filtry: mezní frekvence 5 kHz; Butterworthovy 4. řádu,
- vstupní citlivost: 160 mV – 1,76 V
- napájení: min. 11 V (ideálně). max. 35 V

POPIS HARDWARU

Vzhledem k požadavku univerzálnosti i kvůli snadnějšímu zprovoznění a odhalování chyb (důvodem je také postupný vznik celku) je zařízení rozděleno do několika samostatných, ucelených modulů (vše je patrno z přiložených schémat):

- řídící a výpočetní jednotka
- modul uživatelského rozhraní
- modul A/D a D/A převodníků („zvuková karta“)
- modul zesilovačů a zpětné vazby
- výkonový sluchátkový zesilovač
- zdroj stabilizovaného napětí

Výkonnost a složitost (jak hardwaru, tak softwaru) se odvíjí od použitého mikropočítače. Dnešní trh nabízí mnoho jednočipových mikropočítačů lišících se počtem I/O pinů, počtem vestavěných obvodů, velikostí pamětí, výpočtovými schopnostmi, rychlostí či rovnou předurčeností k použití v určitých oblastech a tento sortiment se velmi rychle rozšiřuje. Takováto rozmanitost s sebou nese rozličné architektury mikroprocesorů, instrukční sady,

různé vývojové nástroje i programátory k samotnému přenesení kódu do mikropočítače. Vyznat se tak v nabídce a přecházet mezi kategoriemi mikropočítačů není úplně snadné.

Protože mám zkušenosti zejména s 8-mi bitovými mikropočítači (spíše mikrokontroléry) firmy Microchip, vybral jsem jejich PIC16F874A, přestože by se našli mnohem a mnohem vhodnější, třeba přímo z řad DPS (digitálních signálových procesorů), které jsou navrženy pro zpracování signálu obecně.

ŘÍDÍCÍ A VÝPOČETNÍ JEDNOTKA

Některé základní vlastnosti PIC16F874A:

- frekvence 20 MHz – 5 MIPS (milionů operací za vteřinu)
- architektura RISC (redukovaná instrukční sada) – pouze 35 instrukcí
- zásobník na 8 návratových adres
- 4 k words (14 bit) Flash program memory (100 000 cyklů) – 4096 instrukcí
- 192 B SRAM
- 128 B EEPROM (1 000 000 cyklů)
- 33 vstupů/výstupů
- 3x časovač/čítač
- SSP (synchronní sériový port) a I2C
- UART (univerzální synchronní asynchronní přijímač a vysílač)
- 8-kanálový 10-bitový A/D převodník
- 2x PWM (pulzně-šířková modulace) modul

Ostatní moduly jsou k hlavní řídící jednotce připojeny dvěma typy synchronních sběrnic:

1, sériovou sběrnicí SPI – poměrně velmi rychlá a jednoduchá sběrnice využívající 4 signály: data do a z periferie a hodinový signál (společné pro všechny) a adresní signál (jedinečný pro každé zařízení), který určuje, s čím chceme komunikovat (tzv. hardwarová adresace). Tímto odpadá potřeba „zdlouhavého“ odesílání adresy požadované periferie při každém spojení.

2, paralelní sběrnicí – 8 datových vodičů, 2 hodinové signály (pro čtení a pro zápis) a opět pro každé zařízení jedinečný adresní vodič.

Sériová sběrnice umožňuje adresovat 16 zařízení (0 – 15), přičemž první 3 pozice jsou rezervovány pro 3 EEPROM (k uložení např. znakové sady, struktury menu, uvítací bitmapy apod.) Adresní bity jsou negovány – log.0 … aktivní.

Paralelní sběrnice umožňuje adresovat 8 zařízení (0 – 7). Adresní bity zde nejsou negované. Na tuto sběrnici je napevno připojeno 2x32 KB SRAM, které obsadily prví 3 pozice. Adresa 0 má funkci zapsání nižšího bytu adresy pro SRAM (názorně: na datové vodiče pošleme nižší byte adresy pro SRAM a poté !!!změníme!!! adresu paralelní periferie na 0, což vyvolá na odpovídajícím vodiči vzestupnou hranu a adresa pro SRAM je uložena). Adresa 2 odpovídá zapsání vyššího bytu adresy pro SRAM. Adresa 3 už přímo odpovídá samotné SRAM a následně je již možno psát do či číst z předtím určeného místa. Další podrobnosti jsou patrné ze schématu. V této konstrukci nejsou na paralelní sběrnici připojena žádná další zařízení.

Pozn.: SRAM paměti byly získány ze stařičké základní desky PC.

Deska také obsahuje převodník z TTL logiky na linku rs232, což je také sériový (COM) port počítače. Po této lince je možno přeprogramovat mikropočítač bez nutnosti hardwarového programátoru, či ho využít k testování apod. Převádí se pouze základní signály RXD a TXD, avšak na převodníku je volný další pár, který by šel požít např. k hardwarovému řízení komunikace.

Na mikropočítači záměrně zůstaly nepoužity piny RC0 až RC3 (PWM výstupy a vstupy pro čítače) a celý PORTA (piny RA0 – RA5; analogové vstupy).

MODUL UŽIVATELSKÉHO ROZHRANÍ

Funkce z hlediska uživatele již byla popsána s části „Popis konečného zařízení“. Jako display je použit monochromatický grafický display 48x84 bodů z mobilního telefonu Nokia 5110. Jedná se o display firmy Goldentek s řadičem PCD8544. Display je vnitřně rozdělen na 6 řádků – každý elementární sloupec (8 bodů) v řádku odpovídá zapsanému bytu.

Kromě dříve uvedených 6-ti tlačítek je možno připojit k tomuto modulu další 2 tlačítka/přepínače (např. nožní spínač pro zapínání a vypínání efektů apod.).

POPIS SOFTWARU PRO MIKROPOČÍTAČ

Použitý mikropočítač klade na zařízení velká omezení, o to kvalitnější a úspornější musí být program pro něj. Zásadním měřítkem je dosažená vzorkovací frekvence. Abychom pokryli slyšitelné pásmo 20 – 20 000 Hz, potřebovali bychom vzorkovací frekvenci minimálně 40 kHz. Na první pohled by se zdálo, že 5 MIPS je postačujících. To by ale znamenalo zpracovat jeden vzorek signálu ve 125 instrukcích, aniž by mikropočítač dělal cokoli jiného. Bylo by možné této hodnoty dosáhnout ale uživatelské rozhraní jsem nehodlal opomenout.

Všech 35 instrukcí mikropočítače pokryje opravdu velmi základní úkony a uvědomíme-li si, že je 8-bitový, vyvstává mnoho problémů. Za nejsložitější aritmetické operace se dá považovat sčítání a odčítání. Při každé operaci s daty také musí být jedno z čísel umístěno napřed do pracovního registru, tzv. WREG značený W. Sečtení dvou 8-bitových čísel uložených v RAM a zapsání výsledku do RAM na místo třetí tak v ideálním případě zabere 3 instrukce. Obdobná operace s 16-bitovými čísly vyžaduje minimálně 8 instrukcí a přitom není ošetřeno přetečení.

Další nepříjemností je rozdělení programové i datové paměti na banky. To znamená, že nemůžeme jednoduše pracovat s jakýmkoli bytem v RAM nebo odskočit v programu ihned na jakékoli místo. Tomu musí předcházet nastavení správné banky v registrech. Pro úplné pochopení je nezbytné přečtení alespoň části katalogového listu mikropočítače.

V praxi se mi podařilo přiblížit se vzorkovací frekvenci 30 kHz včetně obsloužení uživatele a stupnice znázorňující úroveň signálu. Každé zkrácení programu byť o jednu jedinou instrukci je znatelným úspěchem. Zrychlení by přineslo použití interního A/D převodníku a PWM jako výstupu, avšak za cenu snížení hloubky na 10 bitů.

Program je psán přímo v assembleru s použitím volně dostupného vývojového nástroje MPLAB od Microchipu.

STRUKTURA PROGRAMU (PROJEKTU)

Projekt obsahuje tyto soubory:

- Main.asm – hlavní část programu
- Init.asm – inicializace mikropočítače

- Routines.asm – čekací rutiny
- Hardware_routines.asm – základní rutiny pro SPI, pro práci s RAM a EEPROM …
- COM_access.asm – komunikace mikropočítače s PC po rs232
- Math.asm – násobení a dělení
- Display_modul.asm – základní obsluha displeje, výpis čísel, znaků a řetězců, bitmapy, …
- Effect_Echo.asm – programový modul „Echo“
- Effect_Flanger.asm – programový modul „Flanger“
- DSP_Header.inc – konstanty a definice
- Math_macros.inc – základní operace s 16-bitovými čísly
- 16f874a.lkr – soubor pro linker s rozložením programové paměti

Program lze logicky rozdělit do tří oddílů:

Většinu tvoří podprogramy na té nejnižší úrovni pro komunikaci s hardwarem nebo s PC, pro výstup na displej, matematické operace, odměřování času apod.

Programové moduly – dělají právě to, co se od celého zařízení očekává především. Každý musí mít minimálně 5 podprogramů:

- Inicializace a start modulu
- Zastavení modulu
- Reakce na změnu vlastnosti v menu
- Rutinní podprogram (je volán při každém průchodu hlavní nekonečnou smyčkou)
- Podprogram při přerušení (u efektu je to zpracování jednoho vzorku signálu)

Tyto podprogramy musí být zapsány v tabulkách uvnitř „Main.asm“. Pokud by ale existence nějakého z těchto podprogramů nebyla potřebná, lze do tabulky rovnou zapsat RETURN. No a pokud chceme mít k tomuto modulu přístup jako uživatelé skrze rozhraní, musíme pro něj vytvořit menu strukturu, navázat ji na ostatní menu a celek poté zapsat do externí EEPROM.

Při „spouštění“ programového modulu je ještě před vlastním voláním podprogramu pro start volán podprogram pro změnu vlastnosti postupně pro každou položku – vytvoření vnitřních implicitních údajů na základě implicitních údajů předaných z postupně menu.

Přidávání a odebírání programových modulů by tak nemělo vyžadovat nějak hluboké zásahy do ostatního programového kódu, nicméně částečně vyžaduje jeho znalost.

Ve středu všeho je program „Main.asm“, který se stará především o komunikaci s uživatelem. Stará se o chod menu a spravuje programové moduly (zde jsou to efekty, ale mohlo by jít i o cokoli jiného). Podrobnější funkce by snad měla být patrná z jeho komentářů.

Reprezentace jednotlivých menu mají takovouto strukturu:

```
1B - číslo podprogramu
2B - offset předchozí struktury v ROM
xB - string nadpisu
1B - počet položek této struktury

xB - string 1.položky
4B - Nastavení vlastností 1. položky
xB - string 2.položky
4B - Nastavení vlastností 2. položky
...
--------------------------------------
Vlastnosti položky z menu typu nastavení:
  1B – ItemVal   - počáteční hodnota
  1B – ItemStep  - krok při změně
  1B – ItemMin   - minimum
  1B – ItemMax   - maximum
Vlastnosti položky z menu typu rozcestí:
  2B - Offset dalšího menu
  2B - nepoužito
```

Hlavní menu z obr.4 vypadá v ROM následovně:

```
0550:  00 00 00 09 4D 41 49 4E  ....MAIN
0558:  20 4D 45 4E 55 03 05 20  .MENU..
0560:  45 63 68 6F 00 35 00 00  Echo....
0568:  08 20 46 6C 61 6E 67 65  . Flange
0570:  72 00 51 00 00 0B 20 43  r.Q... C
0578:  6F 6D 20 61 63 63 65 73  om acces
0580:  73 00 A9 00 00           s....
```

Menu typu nastavení pro efekt „Echo“:

```
0580:                 01 FF CB       ...
0588:  04 45 43 48 4F 01 0E 20  .ECHO..
0590:  44 65 6C 61 79 3A 20 20  Delay:
0598:  20 20 30 6D 73 32 01 01    0ms...
05A0:  64                       .
```

Nadpis „ECHO“, jediná položka „ Delay: 0ms“, s výchozí hodnotou 50, měnitelná po kroku 1 v intervalu od 1 do 100.

Pozn.: Data jsou skutečným výpisem z EEPROM a je tudíž také patrné provázání obou menu přes offsety.

EFEKTY

Důležité je, že převodníky jsou připojeny přes sériovou sběrnici a SRAM přes paralelní. Díky tomu se může obojí komunikace provádět zároveň. Oba efekty pracují s RAM jako s nekonečným kruhovým zásobníkem. Každý vzorek v tomto zásobníku obsazuje dva byty. Každý efekt si uchovává aktuální pozici (samozřejmě vždy dělitelnou 2), na kterou zapisuje právě přečtený vzorek. „Delay“ poté udává o kolik bytů zpět se nachází požadovaný zpožděný vzorek („Delay“ je opět dělitelné 2).

Spouštění efektové smyčky (podprogramu) je řízeno přerušením. Důležité je, aby pro hlavní smyčku zbylo alespoň 7 cyklů bez přerušení – tolik cyklů je potřeba, aby mohla pracovat sériová komunikace. V jiném případě program určitě „zamrzne“.

Echo je nejjednodušší efekt, pouze vytváří konstantní zpoždění. Může posloužit jako základ pro další efekty pracující se zpožděním.

Flanger mění lineárně zpoždění signálu v úzkém intervalu (např. 0–5 ms) a změna má periodu v jednotkách sekund. Sám osobě míchá zpožděný a původní signál ve zvoleném poměru. Součtem oněch dvou signálů vzniká nový signál, kde jsou vlivem posunu některé frekvence potlačeny a některé naopak posíleny, a díky změně zpoždění (posunu) se mění i tyto frekvence. Výsledkem je zvuk přirovnávaný ke zvuku stíhačky. Změnou poměru míchání se mění „ostrost“ efektu.

ZÁVĚR

Touto prací jsem zdaleka nepokryl možnosti vytvořeného zařízení. Co se efektů týče, napadlo by mě ještě např. vibrato, zkreslení zvuku podle převodní tabulky uložené v RAM, implementace jednoduchého FIR filtru apod. Hardware je dále připraven na stereo rozšíření pouze propojením několik míst, tuto možnost jsem ale ani nestihl vyzkoušel. Také jsem chtěl zkusit zprovoznit slot na SD/MMC karty (podporují jednoduchý SPI režim), ani na to nedošlo.

Před dalšími pokusy bych ale prvně přeci jenom přešel na jiný mikropočítač, konkrétně na PIC18F452. Tento o řadu vyšší mikropočítač nabízí 75 instrukcí, nemá programovou část rozdělenou do bank, RAM prostor je daleko větší a nejen na všechny řídící registry lze přistupovat okamžitě, disponuje 10 MIPS, vnitřní hardwarovou násobičku (násobí dvě bytová čísla v jednom cyklu !) atd. Tento se jako DSP opravdu používá a jeho vnitřní struktura je pro to uzpůsobena daleko lépe. A především je s PIC16F874A pinově kompatibilní – lze ho přímo zaměnit. Nevyhnutelné je ovšem kompletní přepsání programu – za takovéto situace velká nevýhoda assembleru.

Nápadů na vylepšení je stále dost a dost. Přesto si myslím, že je, za čím se ohlédnout, i když toto téma nevypadá nějak atraktivně a mnoho lidí asi výsledek příliš nezaujme. Alespoň mě ale práce přinesla mnoho nových poznatků, vědomostí a zkušeností – a pokud by inspirovala někoho dalšího, zajisté ho taktéž nějakým způsobem obohatí. V principu se nejedná o nikterak složité zařízení – ovšem při stavbě teprve zjistíte, jaké veškeré problémy se mohou objevit a jak základní věci musíte řešit. To platí víceméně obecně. A taková praktická zkušenost není špatná.

ZDROJOVÉ KÓDY

Pozn.: nejsou kompletní, ale obsahují poznámky; vynechávky označeny (…)

MAIN.ASM

```
;=================================================================
;=================================================================
;--------------------- Hlavní programový kód ---------------------
;=================================================================
; Obsluha uživatele - spravuje se menu, reaguje se na tlačítka...
; Obsluha přerušení a správa podprogramů jednotlivých efektů

;-----------------------------------------------------------------
; Definice
;-----------------------------------------------------------------
    #include "p16f874A.inc"
    __config _BODEN_OFF & _CP_OFF & _PWRTE_ON & _WDT_OFF & _HS_OSC & _DEBUG_OFF &
_CPD_OFF & _LVP_OFF

    #include "DSP_Header.inc"

    EXTERN  COM_BOOT
    EXTERN  INITIALIZATION

    EXTERN  WAIT_x100ms
    EXTERN  INIT_DSP, CLEAR_DSP, STARTUP_SCREEN
    EXTERN  GOTO_XY, CoordX, CoordY
    EXTERN  WRITE_DEC, MAKE_BCD, BCDNum
    EXTERN  WRITE_CHAR_FAST, WRITE_STRING, WRITE_CAPTION
    EXTERN  VUMeter

    EXTERN  ROM_READ_PREPARE, Adr
    EXTERN  SPI_IO

    EXTERN  ECHO_START, ECHO_CHANGE, ECHO_LOOP
    EXTERN  FLANGER_START, FLANGER_CHANGE, FLANGER_LOOP, FLANGER_ROUTINE


    GLOBAL  INTERRUPT_END, EFFECT_LOOP_START
    GLOBAL  IntDelay, ItemVal, MenuItemNum

    GLOBAL  SPIReceive, SPIRecBuf
    GLOBAL  ITEM_WRITE_1, ITEM_WRITE_2
    GLOBAL  CHANGE_LEDS

;-----------------------------------------------------------------
; Proměnné potřebné při obsluze přerušení
;-----------------------------------------------------------------
Inter_var   UDATA
; Záloha důležitých registrů
WregBuf     res 1
StatusBuf   res 1
PCLATHBuf   res 1
SPICSBuf    res 1
; Podle další proměnné se nastavuje interval dalšího přerušení
; Přerušení nastává, pokud časovač TMR0 přeteče přes 255
; "IntDelay" je počáteční hodnota TMR0
; Počet cyklů mezi přerušeními = 256 - "IntDelay" + 5
IntDelay    res 1
; Proměnné využité při kritické sériové komunikaci
SPIReceive  res 1
SPIRecBuf   res 1
```

```
;-----------------------------------------------------------------
; Proměnné k obsluze tlačítek
;-----------------------------------------------------------------
Buttons_var UDATA
BtnNew      res 1   ; Nový stav tlačítek
BtnOld      res 1   ; Starý stav tlačítek
BtnClick    res 1   ; Tlačítka, které byla stisknuta
BtnCount    res 1   ; Počítadlo pro automatické opakování stisku
BtnTmrInit  res 1   ; Počáteční hodnota pro časovač autorepeateru
                    ;   v podstatě není nutné; umožňuje změnu
                    ;   rychlosti automatického opakování

;-----------------------------------------------------------------
; Proměnné potřebné pro chod menu
;-----------------------------------------------------------------
Menu_var    UDATA
MenuAdr     res 2   ; ROM adresa současné menu struktury
MenuOffset  res 2   ; Offset předchozí struktury
MenuSubPrg  res 1   ; Číslo podprogramu náležící k současnému menu
MenuItems   res 1   ; Počet položek v menu
MenuItemNum res 1   ; Číslo právě vybrané položky (od 0)
MenuTemp    res 1   ; Pouze vnitřní proměnná
ItemMax     res 1   ; Maximální hodnota právě vybrané položky
ItemMin     res 1   ; Minimální hodnota právě ...
ItemStep    res 1   ; Změna (krok) hodnoty právě ...
ItemVal     res 1   ; Aktuální hodnota právě ...
Backlight   res 1   ; Podsvícení displeje (0-nesvítí; 1-svítí)
; Proměnné "Item..." se týkají pouze právě vybrané položky.
; Obdobné údaje těch ostatních jsou uloženy za sebou v zásobníku
; ležícím od adresy 0xA0 a přistupuje se k nim skrze ukazatel

;=================================================================
;=================================================================
;---------------------- Prvopočátek programu ---------------------
;=================================================================
RESET       CODE    0x0000
    pagesel INITIALIZATION
    CALL    INITIALIZATION
    CALL    COM_BOOT
    GOTO    PROGRAM

; Provede se nezbytná počáteční inicializace mikropočítače
; Dále je vyvolán pokus o navázání spojení po rs232 lince
; Poté program pokračuje na adrese "PROGRAM"

;=================================================================
;####################### OBSLUHA PŘERUŠENÍ ########################
;=================================================================
; Obsloužení interruptu (bez návratu z efektové smyčky)
; trvá max. 30 cyklů
;      min. 28 cyklů

; Vektor přerušení
INTERRUPT   CODE    0x0004
    MOVWF   WregBuf         ; Uložení pracovního registru W

    BCF     INTCON,TMR0IF   ; Nastavení dalšího přerušení
    MOVF    IntDelay,W
    MOVWF   TMR0

    MOVF    STATUS,W        ; Záloha další kritických registrů
    MOVWF   StatusBuf
    MOVF    SPI_CS,W
    MOVWF   SPICSBuf

    BTFSC   SPIReceive,0    ; Případné dokončení kritické SPI
    GOTO    INTERRUPT_L1    ; komunikace
    MOVF    SSPBUF,W        ; Odeslání 1 bytu trvá 8 cyklů ->
```

```
    MOVWF   SPIRecBuf       ; pokud jsme těsně před přerušením
    BSF     SPIReceive,0    ; začali komunikovat, zde již bude
                            ; komunikace dokončena

INTERRUPT_L1                ; Skok do odpovídající efektové smyčky
    MOVF    PCLATH,W
    MOVWF   PCLATHBuf
    CLRF    PCLATH

    MOVF    MenuSubPrg,W
    ADDWF   PCL,F

    GOTO    INTERRUPT_END   ; Tabulka efektových smyček !!!!!!!!!!
    GOTO    ECHO_LOOP
    GOTO    FLANGER_LOOP
    GOTO    INTERRUPT_END

; Volá se po dokončení efektové smyčky
INTERRUPT_END
    MOVF    PCLATHBuf,W     ; Obnovení kritických registrů
    MOVWF   PCLATH
    MOVF    SPICSBuf,W
    MOVWF   SPI_CS
    MOVF    StatusBuf,W
    MOVWF   STATUS
    MOVF    WregBuf,W
    RETFIE                  ; Návrat z přerušení

;=================================================================
;------------ Inicializace hlavní programové smyčky ------------
;=================================================================
MAIN        CODE    0x0030
PROGRAM
    BSF     SPIReceive,0    ; Inicializace displeje
    CALL    INIT_DSP

    CLRF    Backlight       ; Nastavení a rozsvícení podsvětlení
    BSF     Backlight,0
    CLRF    VUMeter
    CALL    CHANGE_LEDS

    CALL    STARTUP_SCREEN  ; Ukázaní úvodní bitmapy na 1500ms
    MOVLW   .15
    CALL    WAIT_x100ms


    CLRF    MenuSubPrg      ; Načtení prvotní menu struktury
    MOVLW   h'05'           ; (MAIN MENU) z ROM na adrese 0x0550
    MOVWF   MenuAdr
    MOVLW   h'50'
    MOVWF   MenuAdr+1       ; MENU_PREPARE na závěr skočí do
    GOTO    MENU_PREPARE    ; nekonečné smyčky MAIN_MENU

;=================================================================
;-------------------- Hlavní nekonečná smyčka --------------------
;=================================================================
MAIN_MENU
    CALL    SUBPRG_ROUTINE  ; Zavolá rutinní podprogram efektu
    CALL    READ_BUTTONS    ; Přečte stav tlačítek
; Další část reaguje na stisk tlačítek (v podstatě události)
    BTFSC   BtnClick,BTN_F1 ; TL1 - v menu o krok zpět
    GOTO    MENU_BACKWARD

    MOVF    MenuItems,F     ; Pokud menu nemá žádné položky,
    BTFSC   STATUS,Z        ; opaku smyčku, jinak jdi dál
    GOTO    MAIN_MENU

    BTFSC   BtnClick,BTN_DOWN   ; Posun po položkách v menu
```

```
    CALL    MENU_DOWN
    BTFSC   BtnClick,BTN_UP
    CALL    MENU_UP


    MOVF    MenuSubPrg,F        ; Na tlačítka "plus" a "mínus"
    BTFSC   STATUS,Z            ; reaguje pouze pokud je číslo
    GOTO    MAIN_MENU_L1        ; podprogramu větší než 0
    BTFSC   BtnNew,BTN_LEFT     ; (pokud běží efekt)
    CALL    BTN_LEFT_PRESS
    BTFSC   BtnNew,BTN_RIGHT
    CALL    BTN_RIGHT_PRESS
    GOTO    MAIN_MENU

MAIN_MENU_L1
    BTFSC   BtnClick,BTN_F2     ; Pokud je pouze v nabídce, reaguje
    GOTO    MENU_FORWARD        ; na TL2
    GOTO    MAIN_MENU
; Opakování smyčky MAIN_MENU


;=====================================================================
;------------------------- Příprava menu -----------------------------
;=====================================================================
; Každá menu struktura je v ROM reprezentována následovně:
; ----------------------------------------
; 1B - číslo podprogramu
; 2B - offset předchozí struktury v ROM
; xB - string nadpisu
; 1B - počet položek této struktury

; xB - string 1.položky
; 4B - Nastavení vlastností 1. položky
; xB - string 2.položky
; 4B - Nastavení vlastností 2. položky
; ...
; ----------------------------------------
; Vlastnosti položky z menu nastavení:
;   1B - ItemVal
;   1B - ItemStep
;   1B - ItemMin
;   1B - ItemMax
; Vlastnosti položky z rozcestí
;   2B - Offset další struktury
;   2B - ---

; Načte takovouto strukturu uloženou na adrese "MenuAdr"
;   a samozřejmě aktualizuje všechny proměnné "Menu_var".
; Kromě toho také zavolá postupně pro každou položku podprogram
;   pro přepočet vnitřních údajů efektu - inicializace hodnot
MENU_PREPARE
(...)
; MENU PREPARE končí skokem do hlavní smyčky


MENU_BACKWARD
MENU_JUMP
; Načte menu určené offsetem v "MenuOffset"
; Pokud "MenuOffset" nezměníme, znamená to zpět.
(...)

MENU_FORWARD
; Načte následující menu
(...)
```

```
;=====================================================================
;-------------- Rutiny pro práci s položkami v menu --------------
;=====================================================================
MENU_DOWN
; Způsobí vybrání nižší položky včetně zavolání ostatní důležitých
; podprogramů s tím souvisejících
(...)

MENU_UP
; Obdobně způsobí vybrání vyšší položky
(...)

;=====================================================================
ITEM_WRITE_1
; Vypíše na displej hodnotu položky beze změny
(...)

ITEM_WRITE_2
; Vypíše na displej hodnotu položky jako číslo "ItemVal"*10^-1
; Jednotky oddělí od desítek a stovek desetinou čárkou
(...)

;=====================================================================
MENU_ITEM_CH
; Na displeji označí aktuální položku a  do proměnných "Item..."
; načte její vlastnosti
(...)

MENU_ITEM_UNCH
; Na displeji odznačí aktuální položku a její proměnné uloží
; zpět do zásobníku
(...)

MENU_ITEM_POINT
; Nastaví ukazatel na data položky "MenuItemNum" v zásobníku
(...)

;=====================================================================
;----------- Čtení tlačítek a některé jejich události ------------
;=====================================================================
READ_BUTTONS
; Zaktualizuje BtnOld, BtnNew a BtnClick
(...)

; Následující podprogramy reagují na stisk (držení) tlačítek
; "plus" a "mínus" a odpovídajícím způsobem mění hodnotu "ItemVal"
; právě vybrané položky (vlastnosti).
BTN_LEFT_PRESS
(...)

BTN_RIGHT_PRESS
(...)

BTN_LEFT_RIGHT_REPEAT
(...)
```

```
;=================================================================
;############# SPRÁVA PODPROGRAMŮ JEDNOTLIVÝCH EFEKTŮ #############
;=================================================================
; Zde jsou části kódu spravující 4 z 5 podprogramů potřebných pro
; každý efekt. Tyto podprogramy jsou pro:
; - start (inicializace) efektové smyčky
; - konec (zastavení) efektové smyčky
; - aktualizace (přepočet) vnitřních údajů efektu
; - provedení rutinních záležitostí (např. obsluha VU metru)
; Poslední, 5. podprogram (samotná efektová smyčka) je v části
; obsluhující přerušení

SUBPRG_START
; Skočí do startovacího (inicializačního) podprogramu efektu
    pageselw    SUBPRG_START_TABLE
    MOVLW   SUBPRG_START_TABLE
    ADDWF   MenuSubPrg,W
    BTFSC   STATUS,C
    INCF    PCLATH,F
    MOVWF   PCL
SUBPRG_START_TABLE          ;Tabulka startovacích podprogramů
    RETURN
    GOTO    ECHO_START
    GOTO    FLANGER_START
    RETURN


SUBPRG_STOP
; Skočí do ukončovacího podprogramu
; Většinou efekty nevyžadují odlišný přístup a lze použít
; základní podprogram EFFECT_LOOP_STOP
    pageselw    SUBPRG_STOP_TABLE
    MOVLW   SUBPRG_STOP_TABLE
    ADDWF   MenuSubPrg,W
    BTFSC   STATUS,C
    INCF    PCLATH,F
    MOVWF   PCL
SUBPRG_STOP_TABLE           ; Tabulka ukončovacích podprogramů
    RETURN
    GOTO    EFFECT_LOOP_STOP
    GOTO    EFFECT_LOOP_STOP
    RETURN

ITEM_CHANGE
; Je voláno při změně hodnoty položky v menu typu nastavení.
; Nastaví na displeji souřadnici řádku této položky a následně
; skočí do podprogramu odpovídajícího efektu, který přepočte své
; interní údaje.
    MOVF    MenuItemNum,W
    ADDLW   .2
    MOVWF   CoordY
    pageselw    ITEM_CHANGE_TABLE
    MOVLW   ITEM_CHANGE_TABLE
    ADDWF   MenuSubPrg,W
    BTFSC   STATUS,C
    INCF    PCLATH,F
    MOVWF   PCL
ITEM_CHANGE_TABLE           ; Tabulka podprogramů pro aktualizaci
    RETURN                  ; interních údajů
    GOTO    ECHO_CHANGE
    GOTO    FLANGER_CHANGE
    RETURN

SUBPRG_ROUTINE
; Skočí do podprogramu s rutinami odpovídajícího efektu
    pageselw    SUBPRG_ROUTINE_TABLE
    MOVLW   SUBPRG_ROUTINE_TABLE
    ADDWF   MenuSubPrg,W
```

```
    BTFSC   STATUS,C
    INCF    PCLATH,F
    MOVWF   PCL
SUBPRG_ROUTINE_TABLE        ; Tabulka podprogramů s rutinami
    RETURN
    RETURN
    GOTO    FLANGER_ROUTINE
    RETURN


;=====================================================================
;------------ Elementární spuštění, zastavení efektu -------------
;=====================================================================
EFFECT_LOOP_START
; Povolí přerušení od TMR0
    CLRF    TMR0
    BCF     INTCON,TMR0IF
    BSF     INTCON,GIE
    RETURN

EFFECT_LOOP_STOP
; Zakáže přerušení od TMR0 a zhasne VU metr
    BCF     INTCON,GIE
    CLRF    VUMeter
    CALL    CHANGE_LEDS
    RETURN

;=====================================================================
;---------------------------- Ostatní ----------------------------
;=====================================================================
CHANGE_LEDS
; Podle proměnných "VUMeter" a "Backlight" rozsvítí/zhasne
; LED diody a podsvícení.
; Bity 2..7 proměnné "VUMeter" odpovídají jednotlivým LED.
    SET_SPI_ADR LED_ADR
    MOVF    VUMeter,W
    ANDLW   b'11111100'
    IORWF   Backlight,W
    CALL    SPI_IO
    NULL_SPI_ADR
    RETURN

    END
```

EFFECT_ECHO.ASM

```
;=====================================================================
;-------------------------- EFEKT ECHO --------------------------
;=====================================================================

    #include "p16f874a.inc"
    #include "DSP_Header.inc"

    EXTERN  INTERRUPT_END
    EXTERN  EFFECT_LOOP_START
    EXTERN  RAM_CLEAR
    EXTERN  MUL, MulByte, MulWord, MulAns
    EXTERN  ITEM_WRITE_1
    EXTERN  IntDelay, ItemVal

    ; Podprogramy určené k zapsání do tabulek hlavního programu
    GLOBAL  ECHO_START, ECHO_CHANGE, ECHO_LOOP

Effect_var  UDATA_OVR
Pos         res 2   ; Adresa aktuální pozice v RAM
Delay       res 2   ; O kolik bytů před aktuální pozicí
```

```
                    ; se nachází chtěný zpožděný vzorek

SampleIn    res 2   ; vzorek čtený z ADC, poté zapisovaný do RAM
SampleOut   res 2   ; vzorek čtený z RAM, poté zapisovaný na DAC
SAdr        res 2   ; pomocná proměnná

Count       res 1   ; pomocná proměnná

;=================================================================
ECHO    CODE

ECHO_START
; Inicializace a start echa
    CALL    RAM_CLEAR           ; Vyčištění RAM
    CLRF    Pos                 ; Nastavení nulové adresy
    CLRF    Pos+1
    MOVLW   .92                 ; Nystavení přerušení
    MOVWF   IntDelay            ; na 29 585 Hz
    CALL    EFFECT_LOOP_START   ; Zapnutí přerušení
    RETURN

ECHO_CHANGE
; Reaguje na změnu vlastnosti (zpoždění - jeho jediná vlastnost)
    MOVLW   CHAR8           ; novou hodnotu vypíše na displej
    CALL    ITEM_WRITE_1

    MOVF    ItemVal,W       ; Vnitřní proměnná "Delay"
    MOVWF   MulByte         ; je rovna "ItemVal" * 592
    MOVLW   .2
    MOVWF   MulWord
    MOVLW   .80
    MOVWF   MulWord+1
    CALL    MUL
    MOVF    MulAns,W
    MOVWF   Delay
    MOVF    MulAns+1,W
    MOVWF   Delay+1
    RETURN

;=================================================================
ECHO_LOOP                                               ; 123 cyklů
; Zpracování jednoho vzorku
; Přečte vzorek z ADC a zapíše ho na aktuální pozici v RAM.
; Z RAM přečte dřívější vzorek a zapíšeho na DAC.
; Pozn.: instrukce odsazené více vpravo jsou součástí paralelní
;   komunikace s RAM
;-----------------------------------------------------------------
; I. část - čtení z ADC a RAM + částečně zápis do RAM

    MOVF    SPI_CS,W
    IORLW   h'0F'
    ANDLW   ADC_ADR
    MOVWF   SPI_CS
    MOVLW   b'00000001'
    MOVWF   SSPBUF

    ;--------------------
    ; SAdr = Pos - Delay
    ; Adresa, ze které se čte zpožděný signál

        MOVF    Delay,W
        SUBWF   Pos,W
        MOVWF   SAdr

        MOVF    Delay+1,W
        SUBWF   Pos+1,W
        MOVWF   SAdr+1
```

```
    BTFSS   STATUS,C
    DECF    SAdr,F  ;(8)
    NOP             ;(9)
;--------------------

MOVF    SSPBUF,W
BSF     SSPCON,SSPM0
MOVLW   b'10100000'
MOVWF   SSPBUF

;--------------------
; čtení "SampleOut" z RAM
    CLRF    PPI_CS

    MOVF    SAdr+1,W
    MOVWF   PPI_DAT
    INCF    PPI_CS,F
    MOVF    SAdr,W
    MOVWF   PPI_DAT
    INCF    PPI_CS,F

    BSF     STATUS,RP0
    MOVLW   b'11111111'
    MOVWF   TRISD
    BCF     STATUS,RP0

    BCF     PPI_OE
    MOVFW   PPI_DAT
    MOVWF   SampleOut
    BSF     PPI_OE

    BSF     STATUS,RP0
    CLRF    TRISD
    BCF     STATUS,RP0  ;(18)

; čtení "SampleOut+1"
    CLRF    PPI_CS

    INCF    SAdr+1,W
    MOVWF   PPI_DAT

    MOVLW   h'02'
    MOVWF   PPI_CS

    BSF     STATUS,RP0
    MOVLW   b'11111111'
    MOVWF   TRISD
    BCF     STATUS,RP0

    BCF     PPI_OE
    MOVFW   PPI_DAT
    MOVWF   SampleOut+1
    BSF     PPI_OE      ;(31)

    BSF     STATUS,RP0
    CLRF    TRISD
    BCF     STATUS,RP0  ;(34)
;--------------------
MOVF    SSPBUF,W
ANDLW   b'00001111'
MOVWF   SampleIn
MOVLW   b'00000000'
MOVWF   SSPBUF
;--------------------

; zápis "SampleIn" - pozn. již bylo přečteno z ADC
    CLRF    PPI_CS
```

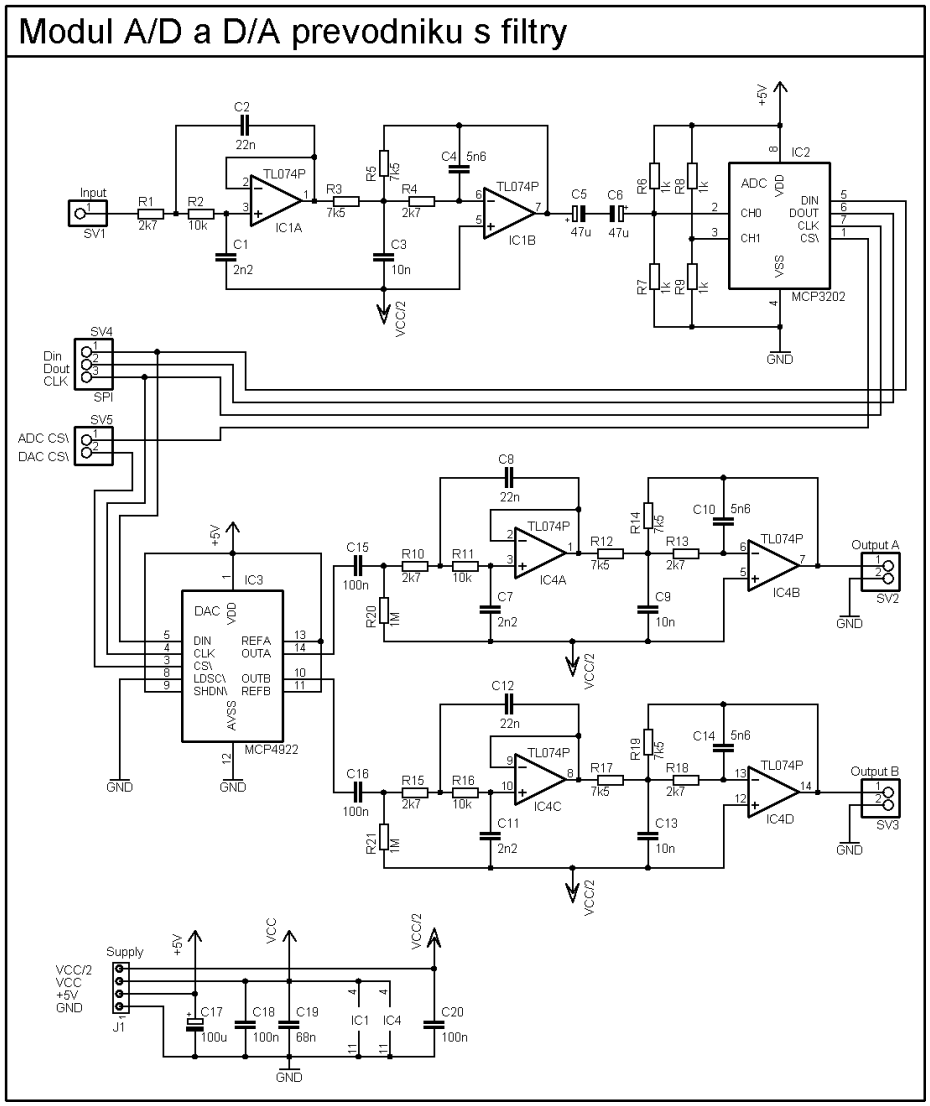
```
        MOVF    Pos+1,W
        MOVWF   PPI_DAT
        INCF    PPI_CS,F

        MOVF    Pos,W
        MOVWF   PPI_DAT
        INCF    PPI_CS,F

        MOVF    SampleIn,W
        MOVWF   PPI_DAT
        BCF     PPI_WE
        BSF     PPI_WE      ;(11)
        SPI_WAIT            ; Počká na dokončení SPI komunikace
    ;--------------------
    MOVF    SSPBUF,W
    MOVWF   SampleIn+1
    INCF    SPI_CS,F

;-----------------------------------------------------------------------
; II. část - zápis do DAC a RAM

    BCF     SSPCON,SSPM0
    MOVFW   SampleOut
    IORLW   b'10110000'
    MOVWF   SSPBUF

    ;--------------------
    ; zápis "SampleIn+1"
        CLRF    PPI_CS

        INCF    Pos+1,W
        MOVWF   PPI_DAT

        MOVLW   h'02'
        MOVWF   PPI_CS

        MOVF    SampleIn+1,W
        MOVWF   PPI_DAT
        BCF     PPI_WE
        BSF     PPI_WE      ;(9)
    ;--------------------

    MOVF    SSPBUF,W
    MOVF    SampleOut+1,W
    MOVWF   SSPBUF

    ;--------------------
    ; Pos := Pos + 2
    ; Posun adresy ukazující do RAM na následující pozici
        MOVLW   h'02'
        ADDWF   Pos+1,F
        BTFSC   STATUS,Z
        INCF    Pos,1
    ;--------------------
        NOP
        NOP
        NOP
        NOP
        NOP
    ;--------------------

    MOVF    SSPBUF,W

;-----------------------------------------------------------------------
; III. část - případné dodatečné výpočty a návrat

    GOTO    INTERRUPT_END
    end
```

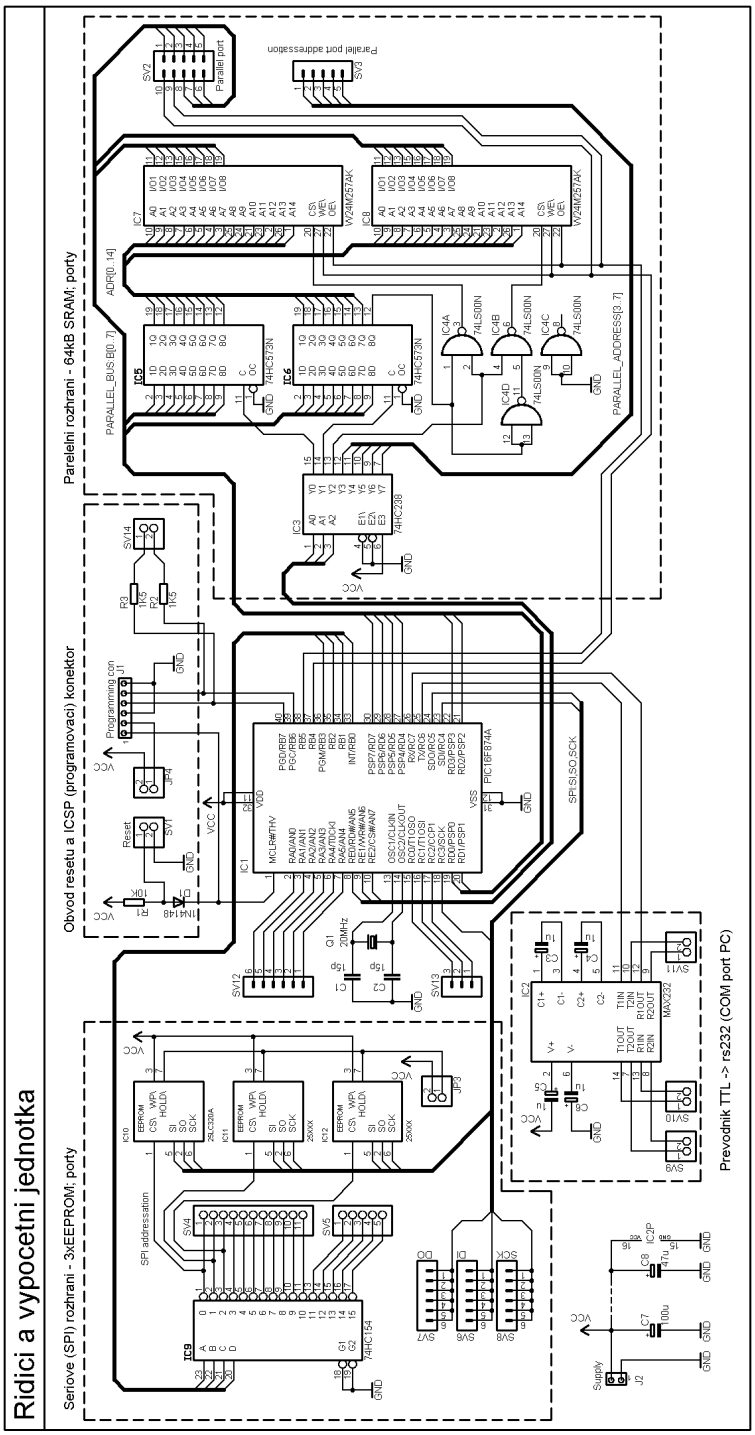
COM_ACCESS.ASM

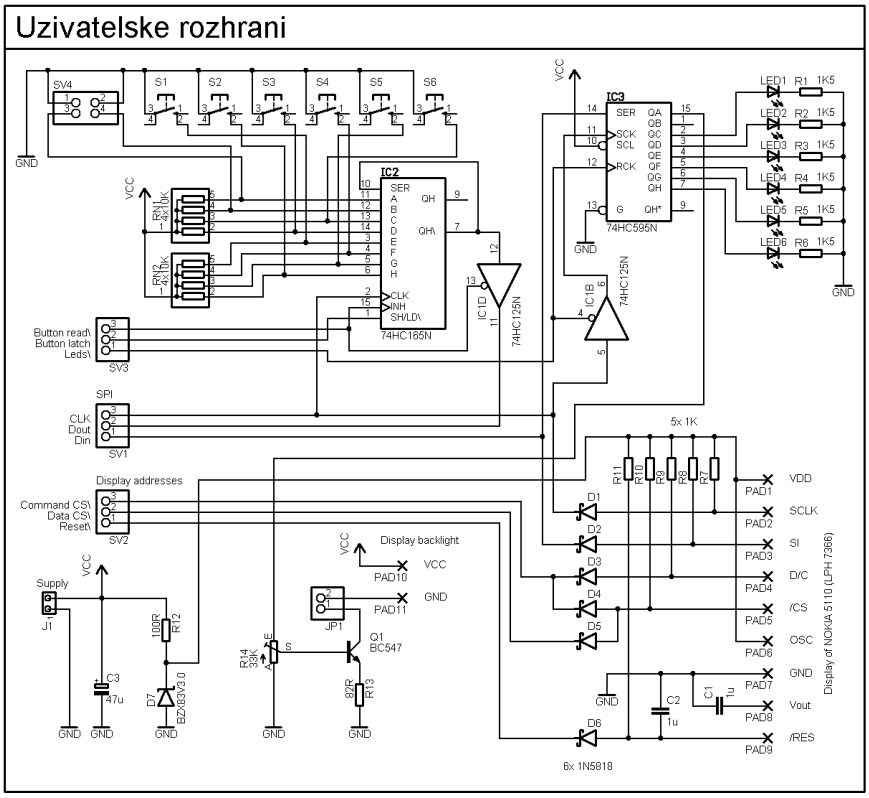
```
;=================================================================
;-------- Komunikace mezi mikropočítačem a PC přes rs232 ---------
;=================================================================
    #include "p16f874a.inc"
    #include "DSP_Header.inc"

    EXTERN  SPI_IO
    EXTERN  WAIT_x10ms
    GLOBAL  COM_SYNC, COM_ACCESS, COM_BOOT

Temp    UDATA_OVR
COM_Command res 1
COM_Dat     res 1

        CBLOCK  0x20
Length:0x01
Count:0x01
Check:0x01
Buf:0x42
        ENDC

;=================================================================
RECEIVE_NEXT_BYTE   MACRO
; Čeká na další byte, pokud nepřijde včas (určeno v podprogramu
; COM_ACCESS_RNB), vyvolá TIMEOUT, jinak vrátí přijatý byte
; skrz WREG.
(...)

;=================================================================
COM_INTERFACE   CODE

COM_SYNC
; Pokus o synchronizaci mikropočítače a PC.
; Vyšle znak "B". Pokud před vypršením timeoutu přijme od PC
; znak "Y", odpoví znakem "O" a nastaví STATUS,Z na 1.
; Pokud vyprší timeout nebo přijme jiný znak než "Y", nastaví
; STATUS,Z na 0.
; Poté se vrátí.
(...)

;=================================================================
COM_BOOT
; Vyvolá pokus o navázání spojení - "COM_SYNC".
; V případě úspěchu přejde do módu naslouchání, jinak se vrátí.
(...)

COM_ACCESS
; Naslouchá počítači a interpretuje jeho příkazy.
; Na první byte četá neomezeně, další se musí vejít do timeoutu.
; První byte - číslo příkazu
; Druhý byte - parametr příkazu
; Pokud je příkaz neplatný, odpoví znakem "E". Jinak příkaz
;   interpretuje, což končí odpovědí:
; "O" - zdárné dokončení
; "L" - vypršel timeout (vymaže se fronta)
; "E" - v případě nějaké chyby (všechny chyby nejsou ošetřeny)
; Dále se celý postu opakuje
; Výjimkou je příkaz TERMINATE - odpoví "T" a program se vrátí
;   ze stavu naslouchání
(...)
```

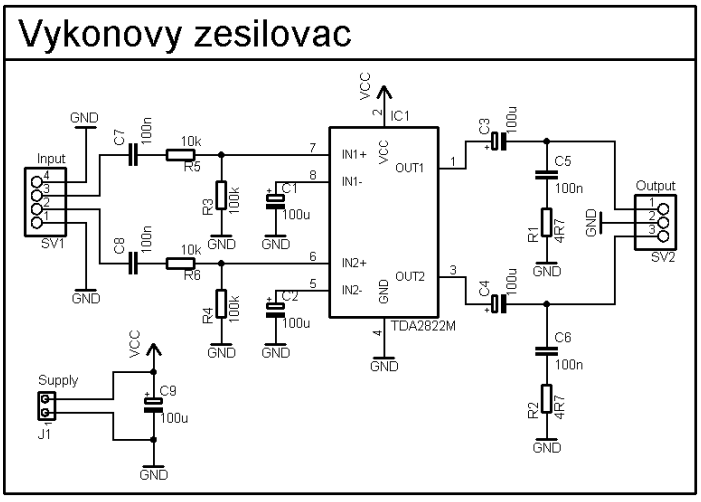
```
COM_ACCESS_TABLE                    ; Tabulka příkazů
    GOTO    COM_ACCESS_TERMINATE    ; Ukončení naslouchání
    GOTO    COM_ACCESS_SPI_CS       ; Adresuje SPI zařízení
    GOTO    COM_ACCESS_SPI_WRITE    ; Odešle po SPI byte
    GOTO    COM_ACCESS_SPI_READ     ; Přijme po SPI byte
    GOTO    FLASH_WRITE
    GOTO    FLASH_READ
    GOTO    COM_ACCESS_RESTART      ; Skočí na úplný začátek
                                    ; programu, ale nevymaže např.
                                    ; zásobník návratových adres.

FLASH_WRITE
; Slouží k nahrání programu do mikropočítače po rs232.
; Paramert je počet WORDů k zapsání (max. 32 words).
; 1. přečte počáteční adresu zápisu programu (word)
; 2. přečte odpovídající počet WORDů nového programu
; 3. přečte kontrolní součet (byte) a zkontroluje ho
; 4. v kladném případě začne zapisovat do programové paměti
; POZOR programovat lze vždy jen čtveřici zarovnaných WORDů
(...)

FLASH_READ
; Slouží ke čtení programové paměti po rs232
; Paramert je počet WORDů k přečtení (max. 255 words).
; 1. přečte počáteční adresu (word)
; 2. přečte kontrolní součet adresy (byte) a zkontroluje ho
; 3. odešle odpovídající počet WORDů programové paměti
; 4. odešle kontrolní součet předtím odeslaných WORDů
(...)

;Pozn.: WORD má tvar ["nižší byte""vyšší byte"]
;Pozn.: Kontrolní součet - je v podstatě součet všech bytů,
;         ze kterého je v jeho 256-kovém zápisu dělán ciferný
;         součet tak dlouho, dokud číslo nemá pouze jednu cifru.
;         př.: 0x0286 -> 0x88
;         př.: 0xFFFF -> 0x01FE -> 0xFF
;         př.: 0x12F861 -> 0x016B -> 0x6C

;=====================================================================

(...)
(...)
(...)

;=====================================================================

COM_ACCESS_RNB                              ;Receive Next Byte
; TimeOut [ms] = (256 - "xxx") * 0.4096
    BCF     STATUS,RP1
    CLRF    TMR1L
    MOVLW   .243            ; "xxx"
    MOVWF   TMR1H
    BCF     PIR1,TMR1IF

    BTFSS   PIR1,RCIF
    GOTO    $+0x03
    BSF     STATUS,Z
    RETURN
    BTFSS   PIR1,TMR1IF
    GOTO    $-0x05
    BCF     STATUS,Z
    RETURN

    END
```

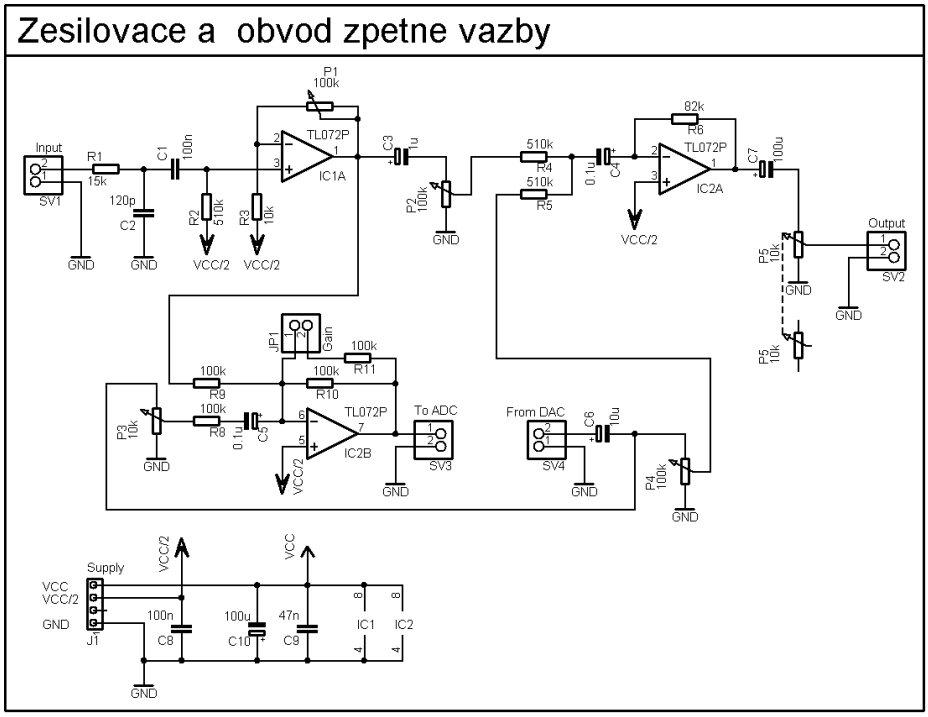
SCHÉMATA

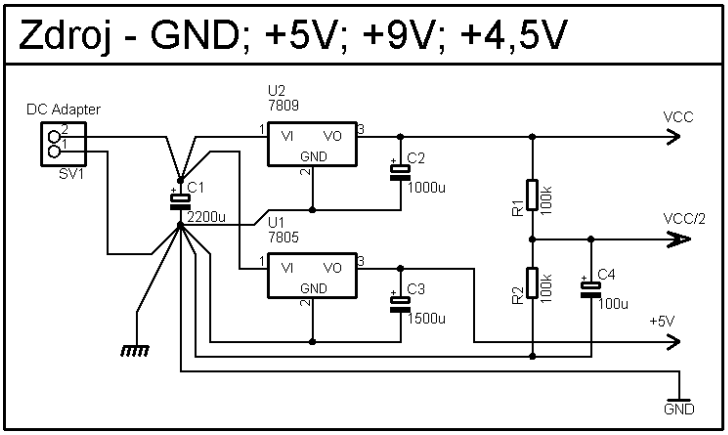

Modul A/D a D/A prevodniku s filtry
Input
SV1
R1
2k7
R2
10k
C2
22n
TL074P
IC1A
R3
7k5
R5
7k5
C4
5n6
R4
2k7
TL074P
IC1B
C5
47u
C6
47u
C1
2n2
C3
10n
VCC/2
+5V
IC2
ADC
VDD
CH0
CH1
VSS
DIN
DOUT
CLK
CS\
MCP3202
R6
1k
R8
1k
R7
1k
R9
1k
GND
SV4
Din
Dout
CLK
SPI
SV5
ADC CS\
DAC CS\
IC3
DAC
VDD
DIN
CLK
CS\
LDSC\
SHDN\
REFA
OUTA
OUTB
REFB
AVSS
MCP4922
C8
22n
TL074P
IC4A
R10
2k7
R11
10k
R12
7k5
R14
7k5
C10
5n6
TL074P
IC4B
R13
2k7
C15
100n
R20
1M
C7
2n2
C9
10n
Output A
SV2
C12
22n
TL074P
IC4C
R15
2k7
R16
10k
R17
7k5
R19
7k5
C14
5n6
TL074P
IC4D
R18
2k7
C16
100n
R21
1M
C11
2n2
C13
10n
Output B
SV3
Supply
VCC/2
VCC
+5V
GND
J1
C17
100u
C18
100n
C19
68n
IC1
IC4
C20
100n


Ridici a vypocetni jednotka
Seriove (SPI) rozhrani - 3xEEPROM; porty
Obvod resetu a ICSP (programovaci) konektor
Parelelni rozhrani - 64kB SRAM; porty
Prevodnik TTL -> rs232 (COM port PC)
Parallel port
Parallel port addressation
SPI addressation
Programming con
Reset
Supply
PARALLEL_BUS.B[0..7]
PARALLEL_ADDRESS[3..7]
ADR[0..14]
74HC154
74HC238
74HC573N
74LS00N
W24M257AK
PIC16F874A
MAX232
SPI.SI,SO,SCK
20MHz
1N4148
10K
1K5
100u
47u
15p
GND
VCC


Uzivatelske rozhrani
SV4
S1
S2
S3
S4
S5
S6
GND
VCC
RN1
4x10k
RN2
4x10k
IC2
74HC165N
IC3
74HC595N
IC1D
IC1B
74HC125N
LED1
LED2
LED3
LED4
LED5
LED6
R1 1K5
R2 1K5
R3 1K5
R4 1K5
R5 1K5
R6 1K5
Button read\
Button latch
Leds\
SV3
SPI
CLK
Dout
Din
SV1
Display addresses
Command CS\
Data CS\
Reset\
SV2
5x 1K
R11
R10
R9
R8
R7
D1
D2
D3
D4
D5
D6
6x 1N5818
PAD1
PAD2
PAD3
PAD4
PAD5
PAD6
PAD7
PAD8
PAD9
VDD
SCLK
SI
D/C
/CS
OSC
GND
Vout
/RES
Display of NOKIA 5110 (LPH 7366)
Supply
J1
C3
47u
100R
R12
D7
BZX83V3.0
R14
R15
Q1
BC547
82R
R13
Display backlight
PAD10
PAD11
JP1
C2
1u
C1
1u


Vykonovy zesilovac
VCC
GND
IC1
TDA2822M
IN1+
IN1-
IN2+
IN2-
OUT1
OUT2
Input
SV1
Output
SV2
C7
100n
C8
100n
10k
R5
R6
R3
100k
R4
100k
C1
100u
C2
100u
C3
100u
C4
100u
C5
100n
C6
100n
R1
4R7
R2
4R7
Supply
J1
C9
100u


Zesilovace a obvod zpetne vazby
P1
100k
Input
SV1
R1
15k
C1
100n
120p
C2
R2
510k
R3
10k
TL072P
IC1A
C3
1u
P2
100k
510k
R4
510k
R5
0.1u
C4
82k
R6
TL072P
IC2A
C7
100u
VCC/2
P5
10k
Output
SV2
JP1
Gain
100k
R11
100k
R9
100k
R10
P3
10k
100k
R8
0.1u
C5
TL072P
IC2B
To ADC
SV3
From DAC
SV4
C6
10u
P4
100k
GND
VCC/2
Supply
VCC
VCC/2
GND
J1
100n
C8
100u
C10
47n
C9
IC1
IC2

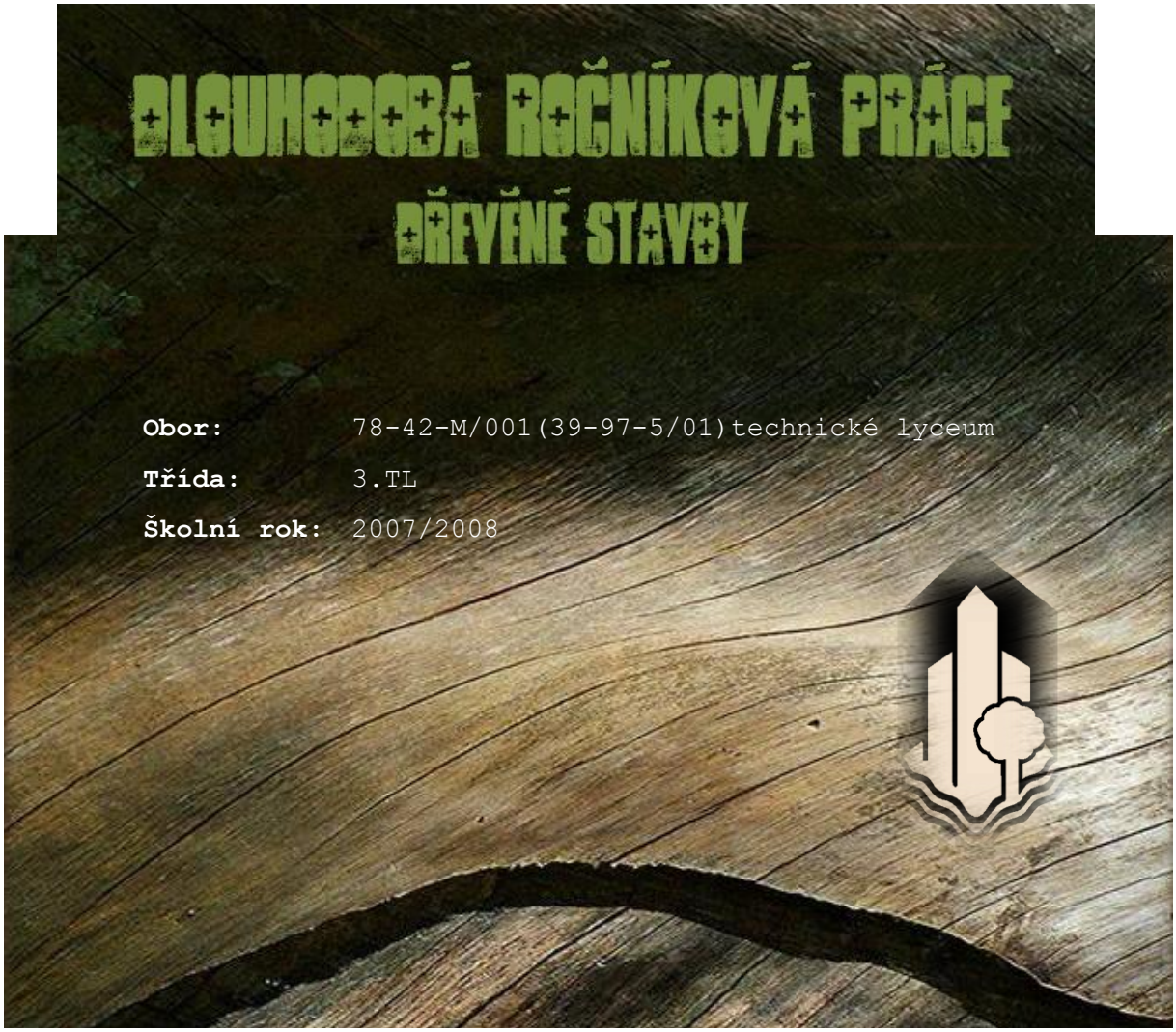

Zdroj - GND; +5V; +9V; +4,5V
DC Adapter
SV1
U2
7809
VI
VO
GND
C2
1000u
C1
2200u
U1
7805
C3
1500u
R1
100k
R2
100k
C4
100u
VCC
VCC/2
+5V
GND

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Aneta Hnízdilová

Střední škola stavební, Jihlava




KAPLE

...

... Milovat dřevo & architekturu

„Prohlašuji, že jsem tuto práci vypracovala samostatně a použila jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

ZADÁNÍ DLOUHODOBÉ PRAKTICKÉ ZKOUŠKY

DŘEVĚNÉ STAVBY

Technická specifikace, osnova, rozsah práce:

- důvod výběru práce
- charakteristika dřev. staveb obecně
- výhody, nevýhody
- konkrétní stavba (sakrální)
- vyhodnocení, názor studenta

Realizace: textová dokumentace svázaná kroužkovou vazbou, Výkresy poskládané ve složce, prezentace v programu PowerPoint na CD nosiči. /konkrétní osnova/

Vedoucí práce: Ing. Jana Svobodová

Konzultant: Petr Procházka

Termín zadání: 27.9.2007
Termín odevzdání: 19.5.2008

OBSAH

str.

I. ÚVODNÍ SLOVO .....

II. HISTORIE .....
II. 1. Historie dřevěných staveb .....
II. 2. Tradice aneb lidová obydlí .....

III. MATERIÁL .....
III. 1. Struktura a vlastnosti .....
III. 2. Rozdělení dřevin .....
III. 3. Rozdělení podle tvaru .....
III. 3. 1. Deskové a hraněné řezivo .....
III. 3. 2. Lepené lamelové dřevo .....
III. 3. 3. Velkoplošné materiály .....

IV. TYPY KONSTRUKCÍ .....

V. CHARAKTERISTIKA .....
V. 1. Charak. dřevěných staveb .....
V. 2. Proč dům ze dřeva .....
V. 3. Srovnání dřeva a železobetonu .....

VI. VÝHODY A NEVÝHODY .....
VI. 1. Výhody .....
VI. 2. Nevýhody .....

VII. PORUCHY .....
VII. 1. Abiotické poruchy .....
VII. 2. Biotické poruchy .....
VII. 3. Proč dochází k poruchám? .....
VII. 4. Dřevokazné houby .....
Přehled dřevokazných hub .....
VII. 5. Dřevokazný hmyz .....
Přehled dřevokazného hmyzu .....

VIII. DŘEVĚNÉ SAKRÁLNÍ STAVBY .....

IX. KAPLE SV. ANTONÍNA .....
IX. 1. Historie kaple .....
IX. 2. Technické část .....
IX. 2. 1. Název a místo stavby .....
IX. 2. 2. Účel stavby .....
IX. 2. 3. Podklady pro projekt .....
IX. 2. 4. Stavebně technické řešení .....

IX. 2. 5. Zemní práce .....
IX. 2. 6. Základová konstrukce .....
IX. 2. 7. Dřevěná konstrukce .....
IX. 2. 8. Vodorovné nosné konstrukce .....
IX. 2. 9. Kce spojující různé úrovně .....
IX. 2. 10. Střešní konstrukce .....
IX. 2. 11. Obvodový plášť .....
IX. 2. 12. Příčky a dělící konstrukce .....
IX. 2. 13. Izolace proti vodě .....
IX. 2. 14. Izolace tepelné .....
IX. 2. 15. Zámečnické konstrukce .....
IX. 2. 16. Lokální vytápění .....
IX. 2. 17. Dešťová kanalizace .....

IX. 3. Architektonická část .....
IX. 3. 1. Urbanistické řešení .....
IX. 3. 2. Arch. a výtvarné řešení .....
IX. 3. 3. Funkční a dispoziční řešení .....
IX. 3. 4. Truhlářské konstrukce .....
IX. 3. 5. Kamenné konstrukce .....
IX. 3. 6. Podlahy dřevěné .....
IX. 3. 7. Podhled .....
IX. 3. 8. Obklady .....
IX. 3. 9. Nátěry .....
IX. 3. 10. Zvon .....
IX. 3. 11. Křížová cesta .....
IX. 3. 12. Osvětlení interiéru kaple .....

X. TVŮRCI KAPLE .....
X. 1. Kuba, Pilař .....
X. 2. Ident. údaje stavebníka .....
X. 3. Ident. údaje zpracovatelů .....
X. 3. 1. Generální projektant .....
X. 3. 2. Projektanti částí .....
X. 4. Identifikační údaje stavby .....

PŘÍLOHA FOPOGALERIE .....
Technická část .....
Architektonická část .....

XI. ZÁVĚR .....

XII. PRAMENY .....

I. ÚVODNÍ SLOVO

Vážení čtenáři,

Otevíráte stránky této dlouhodobé práce, která by Vám měla přiblížit „dřevěné stavby".

Hodně lidí pochybuje o kvalitě a užití dřevostaveb v České republice, a proto dělám tuto práci.

Stačí dočíst do konce. Tato práce není nahodilá, ale myslím, že má nějaký smysl pro druhé i pro mě.

Dovíte se všeobecně o dřevěných stavbách, výhodách, nevýhodách, dřevokazných houbách a hmyzu, ochraně, historii,… O sakrálních stavbách a kapli Sv. Antonína v Černé.

Na konci práce si dovoluji přiložit mé první práce s dřevostavbami.

K práci je též přiloženo CD s prezentací a model kaple v Černé. Publikace je doprovázena fotodokumentací, perokresbami, skicami a naskenovanými částmi výkresů z technické dokumentace.

Dear public,

You are opening page, this long – time works, This work puts near you, wooden building. Many people suspect over quality and use wooden house in czech republic that is why I made this works. This works isn´t haphazard, but I I think that this wokrs has some importance

ÚVOD

Tato práce byla vypracována, aby mohla přiblížit problematiku dřevěných staveb. Představit jejich jedinečnost a dokonalost, ale také upozornit na možné problémy. Když jsem si vybrala toto téma, nebyla jsem si na začátku jistá, jak se s tímto tématem vypořádat. Zvolit formu ankety, zaměřit se na určitou problematiku dřevostaveb? Pak jsem si vlastně uvědomila, že stavby ze dřeva jsou jedinečné ve všech směrech a pojetí práce bylo pro mě ze začátku těžké. Chtěla jsem se věnovat dřevostavbám s využitím panelového systému, ale po hlubším zkoumání jsem zjistila, že se jedná o slisované dřevo, o tzv. OSB desky. A když se na takový domek někdo ve výsledku podívá, tak by ho asi ani nenapadlo, že je ze dřeva. Proto jsem hledala něco konkrétního, kde je dřevo vidět na první pohled. Ale nevybrala jsem si ani tak typologii stavby, jako druh stavby. Blízko mého bydliště, v obci Černá, byla vystavěna v roce 2005 kaple Sv. Antonína. Ta mě motivovala k práci svým moderním a jednoduchým vzhledem. Dozvíte se, jak byla kaple technicky propracována, ale také i architektonicky, včetně interiéru.

Chtěla bych poděkovat konzultantům, obci Černé, hlavně pak panu PaedDr. Vítězslavu Plockovi, za zapůjčený materiál, ochotu Mgr. Oldřicha Hnízdila a pí. Bendové z Černé, kteří mě s kaplí seznámili v terénu.

II. HISTORIE

2.1 Historie dřevěných staveb

Díky dobré opracovatelnosti dřeva, našlo jako stavební materiál uplatnění ve všech historických obdobích. Jak na celkovou stavbu, tak na krovy. Kde bylo dřevo jako materiál dostupné, zachovaly se stavby, krásné a monumentální, ze kterých dnes čerpáme zkušenosti a bereme si je jako vzor pro další moderní stavění. Dokazují to například: Sruby v Kanadě, dřevěné historické staroruské nebo severské konstrukce staveb, dřevěné historické mosty, neobyčejně krásné zdobení střech asijských chrámů, ladné interiéry japonské architektury, kostely a dřevěné stavby ve Skandinávii.

Použití dřeva jako stavebního materiálu se datuje už v době, kdy naši planetu obývali paleolitičtí lovci, kteří bydleli v jeskyních, ale nové výzkumy archeologů poukazují i na jednoduchá dřevěná obydlí. Podrobnější bádání poukazují, že základní způsoby výstavby i vnitřní dispozice vznikaly v mladší době kamenné, základním předělem se považuje nástup domu se svislými stěnami a konstrukčně oddělenou střechou.

Jako stavební materiál bylo využito dřevo a hlína. Půdorysy domů se rozvíjely s uspořádáním osady, které byla oplocena dřevěnou ohradou – palisádou. Další vývoj poukazuje na dřevěný skelet, který byl vyplněn hlínou nebo štípanými fošnami, zvané trhanice. Podle vykopávek lze usuzovat, že měly poschodí, nesené samostatnými sloupy.

V době eneolitu a doby bronzové se stavělo na základových deskách. To sloužilo k dobrému rozložení stavení na rašelinovém podkladu. Tato deska bránila propadnutí. Tvořilo ji několik vrstev ze dřeva. Zespodu malé větve a kůra, na nich rošty z tyčoviny. Podlaha už byla tvořena z kmenů, často ležících kvůli izolaci na příčných polštářích. Domy stály na pilotách – zahrocených kmenech.

Zdokonalení nástrojů ve starší době bronzové, zejména sekery, ovlivnilo lepší opracování dřeva. Z tohoto období jsou známé domy unětické kultury (naleziště v Postoloprtech u Loun) – dlouhé stavby ze tří řad sloupů v pravidelných rozestupech.

Ze střední doby bronzové jsou známé domy o charakteru zemědělských osad s pevnými a velkými dřevěnými domy.

Ve starší době bronzové se objevuje nový způsob stavby a to drážkovou konstrukcí. Pomalu s vývojem drážkové konstrukce se vyvíjela i srubová, stěnová a později hrázděná konstrukce, tyto systémy přetrvaly dodnes.

V dalších obdobích se ale více uplatňovaly a preferovaly materiálové báze kamenné a keramické cihly. Dostupnost stavebního dřeva byla například v horských oblastech velká, a

proto se v těchto oblastech stavělo ze dřeva, jak při výstavbě obytných, ale i sakrálních budov.

Začátkem 20. století se vyvinul v USA nový stavební systém s využitím dřeva jako stavebního materiálu, jednalo se o konstrukci typu „two by four". Znamenalo to přelom ve vývoji úsporných dřevěných konstrukcí a umožnil pokrýt rychle narůstající potřebu ekonomické výstavby a zdravého bydlení. Název je odvozen z dimenze základního konstrukčního prvku o velikosti 2x4 palce (5x10 cm). Tento princip se uplatňuje doposud.

Velký rozmach dřevěných tesařských konstrukcí halových staveb – rámových, příhradových a spřažených dřevo-ocelových konstrukcí – nastal v 19. a 20. století v souvislosti s rozvojem průmyslu. Dalšími průkopnickými konstrukcemi byly: vzpěradlové a věšadlové soustavy, sbíjené nosníky (Stephanův systém), hřebíkové příhradové nosníky, příhradové rámy (Kublerova, Hetzerova-d´Ardantnova soustava), soustavy skruží či spřažené dřevo-ocelové konstrukce nosníků a rámů. Začátkem 19. století se začalo využívat i lamelové lepené dřevo.

2.2 Tradice aneb lidová obydlí ze dřeva

Dřevěné stavební konstrukce měly v Čechách a na Slovensku donedávna bohatou tradici. Lidové stavitelství zaznamenalo postupem času značný vývoj a mnoho nových konstrukcí. Málokterá země se může pochlubit takovým množstvím dochovaných historických objektů ze dřeva na tak malém území v tak vysokém počtu. Zde si uvedeme příklady několika z nich: srubové obytné a hospodářské stavby v Pošumaví, na Českomoravské vrchovině, Valašsku nebo dřevěné kostelíky na východním Slovensku. Základními typy staveb mezi hospodářskými a lidovými obydlí byly dřevěné srubové obytné domy, hrázděné obytné domy, stodoly srubové nebo sloupkové konstrukce, seníky, koliby, hospodářské a technické stavby, jako kovárny, mlýny, sýpky, apod.

III. MATERIÁL

3.1 Struktura a vlastnosti

Pro dřevo je charakteristická vláknitá struktura, anizotropie a nehomogenita. Struktura dřeva předurčuje jeho fyzikální a mechanické vlastnosti.

Anizotropie dřeva nám odhaluje odlišné pevnosti, vlhkosti a ostatních vlastností dřeva. Rovnoběžně s vlákny, tzn. rovnoběžně s podélnou osou kmene, kolmo na vlákna v radiálním směru k letokruhům a kolmo na vlákna v tangenciálním směru. Ve stavební praxi se rozlišuje pouze směr podél vláken a napříč vláken.

Nehomogenita dřeva je způsobena nerovnoměrnou makroskopickou stavbou dřeva s ohledem na šířku letokruhů. To znamená přirozené vady dřeva (např. suky, smolníky, odklon vláken, …). V tomto důsledku má dřevo relativně velké množství vlastností ve srovnání s kovy, plasty …

Při zatížení dřeva ohybem nebo tlakem rovnoběžně s vlákny, je poměrně pevné, dokonce příznivější než ocel. Pevnost dřeva v tahu kolmo na vlákna a ve smyku rovnoběžně s vlákny je nízká.

Dřevo je velmi vhodné pro obrábění, jak ručně, tak i průmyslově. Je to přece jenom organický materiál, který v nevhodných podmínkách rychle podlehne znehodnocení, proto jej musíme správně chránit chemickou ochranou a stavby ze dřeva musí být dobře navrženy. Potom je trvanlivost stavby mimořádně vysoká.

3.2 Rozdělení dřevin

K výrobě dřevěných konstrukcí se používá dřevo vhodných fyzikálních a mechanických vlastností.

Dřeviny si rozdělíme na jehličnaté, listnaté tvrdé a listnaté měkké.

Mezi jehličnany (v domácích podmínkách, tzn. pro Českou republiku), řadíme: Z nejpoužívanějších je to smrk, dále jedle, douglaska, borovice, borovice vejmutovka a modřín.

Z listnatých stromů, pro které použiji název tvrdé, jsou dub zimní i letní, cedr, buk, jasan, javor mléč, horský i babyka, akát, habr, jilm, ořešák vlašský, bříza, hloh, jeřáb i kaštan jedlý. Mezi listnaté tvrdé se řadí i ovocné stromy, jako například švestka, třešeň, hrušeň, jabloň.

Mezi listnaté měkké zahrnujeme dřeviny s měkkou strukturou dřeva, a to lípu, olši, topol, osiku, jírovec a vrbu.

3.3 Rozdělení podle tvaru

3.3.1 Deskové a hraněné řezivo

Řezivo rozdělujeme na deskové (prkna, fošny), hraněné (hranoly), polohraněné (trámy a polštáře).

Deskové řezivo se dělí podle tloušťky. Řezivo tloušťky do 40 mm, prkna a řezivo tloušťky nad 40 mm, fošny.

Hraněné řezivo, tedy hranoly, musí mít šířku větší než 40 mm a a výšku minimálně rovnou šířce.

Polohraněné řezivo zahrnuje dvojstranně řezané řezivo o šířce menší než dvojnásobek tloušťky. Dělí se na polštáře (tloušťka nejvíce 100 mm) a trámy(tloušťka více než 100 mm), jeho nejmenší šířka odpovídá 2/3 tloušťky.

3.3.2 Lepené lamelové dřevo

Toto dřevo je charakteristickou součástí moderních dřevostaveb. Na odvážně a esteticky příjemné věci se dá právě využít tohoto lepeného lamelového dřeva. Jeho předností je, že se dá různě tvarovat. Kvalita lamel a její využití v praxi, např. na trámy, by neměla vlhkost klesnout pod 15%. Lisují se do speciálně navržených tvarů, například pro výrobu obloukových nosných prvků na velké rozpony.

Další využití je na výrobu nosníků s dutým středem a duté lepené dřevěné sloupky.

Dalším produktem je nosník z lisovaných dýhových pásků. Materiál PSL se skládá z přibližně 200 mm dlouhých dýhových pásků o tloušťce několika mm, které se lisují a lepí se fenolickým lepidlem. Používá se k výrobě plnostěnných nosníků. Vylehčené lepené nosníky se vyrábějí podobným způsobem jako plnostěnné. Mají ale krabicový průřez. Používají se ve skladbě stropů.

3.3.3 Velkoplošné materiály

Mezi tyto materiály řadíme překližky, laťovky, biodesky, desky OSB, dřevotřískové desky, dřevocementové desky, dřevovláknité desky, sádrovláknité desky a lamelové dřevo lepené křížem.

IV. TYPY KONSTRUKCÍ

Na konstrukce stěny se uplatňují tyto systémy: srubová stěnová konstrukce, sloupková konstrukce, skeletová konstrukce, panelová konstrukce, hrázděná konstrukce, stěnová soustava z prefabrikovaných desek.

Srubová konstrukce, nosný prvek – stěna se skládá z vodorovně kladených trámů. Na stavbu můžeme použít nehraněné, polohraněné nebo hraněné řezivo (kuláče, polokuláče, hranoly).

Skeletové soustavy, skelet je prostorový nosný systém, tvořen sloupky a vodorovně kladenými nosnými prvky průvlaků. Výplně obvodové stěny a vnitřní nosné příčky jsou nenosné s vyjímkou zabezpečení prostorové tuhosti objektu. Jako nosné prvky se používají plnostěnné průřezy z lepeného lamelového dřeva.

Panelový konstrukční systém je jedním z nejrozšířenějších systémů. Základem je dřevěný rám, který je oplášťovaný vhodným materiálem, jako je třeba dřevotřísková deska, OSB deska, sádrovláknitá deska. Prostor mezi žebry je vyplněn tepelnou a zvukovou izolací.

Hrázděný konstrukční systém, je tvořený z dřevěné kostry, jejíž jednotlivá pole jsou vyplněna nejčastěji cihlovým zdivem.

Stěnové soustavy z prefabrikovaných desek, jsou odvozeny z vyzdívání stěn z velkoformátových cihel, jedná se však o suchý způsob montáže. Na stavbu se používají standardizované a průmyslově vyráběné dřevěné moduly. Základní kus je tvořen z lehkého, dutého materiálu, 5 až 10 kg a rozměrech 300x600mm.

V. CHARAKTERISTIKA

5.1 Charakteristika dřevěných staveb

K přednostem dřeva patří krátká doba výstavby, kterou především ocení investor. Jedná se o to, že dřevostavba je rychleji postavená než cihlový dům. Dřevostavba je také suchou stavbou a nemusí vysychat jako dům z cihel.

Dřevo je lehký stavební materiál, nezatěžuje životní prostředí a má malé energetické nároky. Z ekologického hlediska je to dokonalý stavební materiál. Má také nízké nároky na základovou desku, díky jeho nízké hmotnosti, například v porovnání s keramickým materiálem nebo ocelí.

Velmi dobré mechanické vlastnosti opravňují dřevěné materiály k velmi rozšířenému použití na stavbách. Vysoká odolnost v tlaku, ohybu.

Má výborné tepelně izolační vlastnosti, kmeny se středovým průměrem již 20 cm splňují přísné normy o tepelných ztrátách, tzn., že by zdivo muselo mít šířku 90cm. Proto dobře navržená a postavená dřevostavba vám zajistí úspory energie. Dřevostavba má nejlepší předpoklady k tomu, aby byla navržena jako nízkoenergetická stavba nebo dokonce nulový dům (tzn. velmi nízké náklady na teplo nebo dokonce nulové).

Dřevo stále zůstává živým materiálem i po jeho využití na stavbě. Pohlcuje pachy (např. cigaretový kouř) a částečně pohlcuje prach, absorbuje škodlivé civilizační produkty. Dřevo také udržuje relativní vlhkost a to až 50 – 60%.

Dřevěný stavební materiál příjemně voní a je velmi krásné.

5.2 Proč dům ze dřeva?

Je známé, že v sebehezčích stavbách, ať ocelových, zděných, mrakodrapech, či jen v nízkopodlažních domech, si lidé stěžují na špatnou klimatizaci, cítí se psychicky špatně a výsledkem můžou být i fyzické potíže.

Dřevostavby nabízejí příjemnou atmosféru, která výborně působí na psychiku člověka.

Dřevo je také zcela obnovitelným materiálem v přírodě a nový les vyroste za 30 let.

Dřevo je dobrý zvukový izolátor, který pochycuje zvukové vlny.

V létě je v dřevěné stavbě příjemný chlad a v zimě naopak teplo. Dřevěná stavba je živým materiálem. Má stejné cítění jako člověk. Dřevo se vám přizpůsobí.

Je zde soulad s ekologií a vědomím člověka. Dřevo je vyjádřením filozofie návratu člověka k přírodě. Zahajuje éru

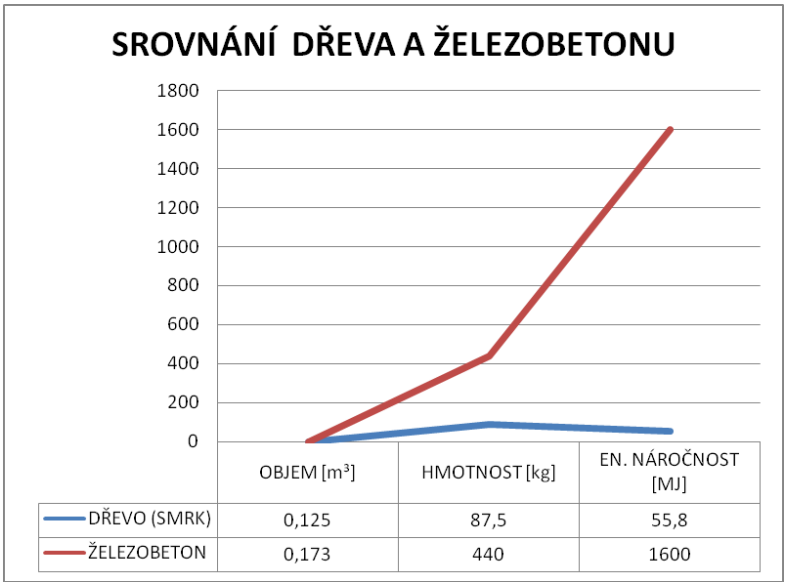
nového životního stylu. Dřevostavby by měly žít v souladu s přírodou.

5.3 Srovnání dřeva a železobetonu

Uděláme si porovnání doplněné grafem. Ke srovnání jsem si vybrala krokev, která je 7,3 m dlouhá. Tuto krokev uděláme ze dřeva a z železobetonu. Určíme si základní rozdíly. Hmotnost krokve vyrobené ze dřeva je 87,5 kg, hmotnost krokve vyrobené z železobetonu je 440 kg. To je rozdíl 5 krokví ze dřeva.

Shrnutí, 5 krokví ze dřeva je hmotnostně stejné jako 1 krokev z železobetonu.

Teď si ještě určíme energetickou náročnost. U dřeva je to 55,8 MJ, při představě, že železobetonová krokev má 28x víc, je to veliký rozdíl. Z toho vyplývá, že krokev z železobetonu má energetickou náročnost 1600 MJ.



	OBJEM [m³]	HMOTNOST [kg]	EN. NÁROČNOST [MJ]
DŘEVO (SMRK)	0,125	87,5	55,8
ŽELEZOBETON	0,173	440	1600

VI. VÝHODY A NEVÝHODY

6.1 Výhody

Jako výhodu staveb ze dřeva bych uvedla účinnost tepelných izolací, proto tyto domy jsou ideální pro stavbu pasivního domu, tzn. dům, kde jsou minimální energetické výdaje.

Další výhodou je malá spotřeba energie na výstavbu. Stavba je suchá, a proto můžeme stavět celý rok. Celkově je energeticky nenáročná, nezatěžuje životní prostředí, v tomto ohledu je uživatelsky sympatická. Zaručuje také malé množství stavebního odpadu.

6.2 Nevýhody

Mezi nevýhody se řadí hořlavost, menší statická odolnost proti přírodním vlivům a katastrofám. Dřevěný materiál je citlivější ke vzniku plísní, napadení hmyzem a dřevokaznými houbami. Těm se dá ale zabránit správnou chemickou ochranou.

VII. PORUCHY

Na zvětrávání, poškození a poruchách dřevěných staveb se podílejí tyto činitelé: abiotické a biotické.

7.1 Abiotické poruchy

Mezi abiotické vlivy patří atmosférické vlivy (sluneční záření, voda, kyslík, prach, imise, vítr,…), agresivní chemikálie, vysoké teploty, sálavé a plamenné energetické zdroje, vyvolávající termickou degradaci a hoření dřeva.

7.2 Biotické poruchy

Mezi biotické poruchy se řadí bakterie, dřevokazné houby, dřevozbarvující houby a plísně, dřevokazný hmyz, ve specifických situacích i ptáci, měkkýši.

7.3 Proč dochází k poruchám?

Dřevo je přírodní materiál, který více nebo méně stárne. Nejvíce se používalo a používá dřevo smrkové nebo jedlové (poměrně málo trvanlivé). Vyšší odolnost mají stavby z dubu, akátu nebo kaštanu. To už věděli staří mistři, proto historické zvonice, spodní trámy, srubové stavby, okna, atd. vydržely až dodnes. Naopak někdy z neznalosti byly použity materiály z netrvanlivého dřeva, jako z buku, topolu nebo lípy. Tyto stavby rychle dožily.

Dalšími znaky poruch jsou zpracovatelko-klimatizační a výrobní vady dřeva, projekční a výrobní chyby a závady, přílišné zatížení konstrukce a jejich prvků. Také pokles mechanických vlastností následkem poškození, ať biotického nebo abiotického nebo přetvoření.

7.4 Dřevokazné houby

Biotičtí škůdci napadají dřevo. Mezi nejznámější patří Dřevomorka domácí (Serpula lacrymans). Napadá listnaté i jehličnaté stromy, nejčastěji se vyskytuje v interiéru, vyhovuje jí vlhkost 30 až 40% a teplota kolem 20 stupňů.

Koniofora sklepní (Coniophora puteana), napadá jehličnany i listnaté stromy, nejčastější výskyt je v interiérech. Optimální vlhkost pro její výskyt je 50 - 90% a teplota 23 stupňů.

Pornatka Vaillantova (Poria vaillanti), napadá jehličnany i listnaté stromy, ale méně. Nejčastěji se vyskytuje v interiérech, ojediněle i v exteriéru. K výskytu potřebuje 35 – 50% vlhkost a teplotu 27 stupňů.

Trámovka plotová (Gloeophyllum sepiarium) a Trámovka jedlová (Gloeophyllum abietinum) napadají pouze jehličnany. Nejčastěji se vyskytují v exteriérech a méně v interiérech. Jejich životní podmínky jsou teplota kolem 30 stupňů a vlhkost kolem 50%.

Houževnatec šupinatý (Lentinus lepideus) a Čechratka sklepní (Paxillus panuoides) tvoří hnědou hnilobu, napadají pouze jehličnaté dřeviny. Nejčastěji se vyskytují v exteriérech, ojediněle i v interiéru. Pro výskyt jim vyhovuje vlhkost mezi 50 až 70% a teploty kolem 27 stupňů.

Outkovka pestrá (Trametes versicolor) a Klanolístka obecná (Schizophyllum commune)jsou houby, které tvoří bílou plíseň, napadají listnaté dřevo, nejčastěji v exteriérech. Optimální teplota pro jejich výskyt je kolem 30 stupňů a vlhkost u Outkovky 60 – 100%, u Klanolístky 50%.

PŘEHLED DŘEVOKAZNÝCH HUB



Dřevomorka domácí



Houževnatec šupinatý



Pornatka Vaillantova



Outkovka pestrá



Trámovka plotová



Klanolístka obecná



Trámovka jedlová



Čechratka sklepní



Koniofora sklepní

7.5 Dřevokazný hmyz

Tesařík krovový (Hylotrupes bajulus) napadá materiály z jehličnanu, použité v interiérech. Životní podmínky pro výskyt tesaříka jsou 29 stupňů a vlhkost 30 až 35%.

Tesařík fialový (Callidium viloacelum) napadá jehličnaté i listnaté dřevo v interiéru i exteriéru. Nejvíce se vyskytuje v podmínkách kolem 25 stupňů a vlhkosti kolem 30%.

Červotoč tečkovaný (Anobium punctatum), napadá jehličnany i listnáče v interiéru. Jeho podmínky pro výskyt jsou 22 stupňů a 26% vlhkost.

Červotoč úmrlčí (Hadrobregmus pertinax)se vyskytuje převážně v jehličnanech, které jsou jako materiál v interiérech, za teploty 25 stupňů a vlhkosti minimálně 20%.

Hrbohlav parketový (Lyctus linearis) napadá listnaté dřevo, především dubový běl, tzn. dubové podlahy. Teplota, při které se drží, je 30 stupňů a vlhkost pouze 12%.

PŘEHLED DŘEVOKAZNÉHO HMYZU



Tesařík krovový



Červotoč tečkovaný



Tesařík fialový



Červotoč umrlčí



Hrbohlav parketový

VIII. DŘEVĚNÉ SAKRÁLNÍ STAVBY

Dřevěné sakrální stavby vznikaly ze srubových konstrukcí. Dochované sakrální stavby ze dřeva se dochovaly již z 15. století. Nejdřív se stavěly malé vesnické modlitebny, kostely, kaple. Podle vlastních představ a inspirací z městské sakrální architektury dokázali dávní tesaři za prostých podmínek vytvořit působivé stavby.

Kostelíky, postavené v gotickém stylu, plně kopírují gotické znaky, půdorysy městských chrámů, ale je na nich znát charakteristické lidové umění.

Evangelické artikulární kostely, tento název je odvozen z tzv. sopronských artikulí, které v roce 1681 poskytly evangelické církvi pravomoci pro výstavbu vlastních sakrálních objektů. Sopronské artikuly ale předepisovaly charakter tehdejších sakrálních staveb. Musely být postavené ze dřeva bez kovových spojů v průběhu jednoho roku, nesměly mít věž, vchod nesměl být přímo z ulice a okna musela být rovná. Byl to zvláštní předepsaný renesanční styl pro evangelické stavby.

Dřevěné konstrukce z východu jsou zejména řecko-katolické a pravoslavné kostely, které měly srubovou konstrukci stěny. Nejčastěji se skládaly ze tří částí. Přední část, nad kterou je konstrukce věže, střední část a svatyně, oddělena bohatě zdobeným ikonostasem. Vnitřní výzdobu tvoří také barevné malby, vzácné sochy a obrazy, které ve spojení s ikonami tvoří mystickou atmosféru.

Dřevěné kostelíky byly přizpůsobené lidskému měřítku, nádherně začleněné do okolní přírodní zeleně. Jejich předností je monumentální, ale jednoduchý vzhled malých rozměrů, to je rozdíl ve srovnání se sakrálními stavbami v městě.

Později byly tvořeny i sloupkovou konstrukcí, zachovaly se např. vstupní portály, které lze také řadit k hodnotným památkám lidové architektury.

IX. KAPLE SV. ANTONÍNA

9.1 Historie kaple

Kaple, zasvěcená sv. Antonínovi, se dříve nacházela na zámečku v Černé. Po restituci zámku původnímu majiteli a následném prodeji soukromému subjektu, musela být kaple odsvěcena a zrušena.





V roce 2001 začala příprava na výstavbu nové kaple. Počátky vyjednávání s Římskokatolickou farností v Měříně a Biskupstvím brněnským vyústily ve výběrové řízení projektanta, které se uskutečnilo v srpnu 2002. Obec požadovala v zadání stavby použití materiálů s dlouhodobou životností, minimální náklady na údržbu a provoz, dokonalou akustiku, respektování záplavového pásma, zeleně a kapacitu kolem 50 sedících návštěvníků. Jedním z rozhodujících faktorů byl i finanční limit nákladů, a to do 3 milionů korun. Ze čtyř přihlášených účastníků byl vybrán ateliér Kuba a Pilař z Brna, se kterými byla uzavřena smlouva o dílo na projektovou dokumentaci. Výsledky posuzoval poradní soubor Biskupství brněnského, který dne 3.4.2003 návrh schválil jako velmi kultivované řešení,

splňující požadavky na současnou výstavbu sakrální stavby. V srpnu téhož roku byl proveden inženýrsko-geologický průzkum pro stavbu kaple v dané lokalitě.

7. července 2004 bylo vydáno stavební povolení. Vzhledem k prováděné plynofikaci obce musela být realizace stavby o jeden rok odložena. V roce 2004 bylo alespoň nakoupeno armovací železo, jehož cena neustále stoupala. Suroviny věnovali místní majitelé lesů a Lesní družstvo obcí Měřín.

V prosinci 2004 byla zveřejněna zpráva o veřejné zakázce na dodavatele stavby, na jejímž základě byla ze tří uchazečů vybrána firma Stylstav, s.r.o., Křižanov. Výstavba byla zahájena v dubnu 2005. Veškeré práce byly dokončeny včas. Kaple byla vysvěcena 10. Června 2006.

Objekt kaple sv. Antonína Paduánského poutá od svého otevření značnou pozornost architektů i laické veřejnosti.

Obec přihlásila stavbu do soutěže Stavba Vysočiny, kde na slavnostním vyhodnocení 29.5.2007 obdržela v kategorii „Stavby občanské vybavenosti" čestné uznání za nezvykle tvarové a materiálové řešení dominantního objektu centra obce. Současně se v hodnocení veřejnosti umístila z 29 přihlášených na druhém místě.

Kaple byla nominována mezi jedenácti vybranými stavbami za ČR na cenu Miese van der Rohe 2007 v Barceloně, což je prestižní evropské udělení za současnou architekturu. Kaple obdržela 2 nominace ze tří.

Kaple zůstala v majetku obce Černá a byla pronajata Římskokatolické farnosti v Měříně.

Výdaje za elektřinu na vytápění a větrání činily za první rok užívání pouhých 25 tisíc korun. To potvrzuje požadované nízké provozní náklady. Stavba byla pořízena za čtyři a půl milionu korun /původní cenu zvýšila změna DPH/. Půl milionu korun stálo vybavení interiéru. Na finančních nákladech se podílely obec, dotace z Programu obnovy venkova Kraje Vysočina a sbírka věřících měřínské farnosti i občanů Černé a Milíkova.

9.2 Technická část

9.2.1 Název a místo stavby

Kaple svatého Antonína, obec Černá, spadající pod Měřín se směrovacím číslem 594 42, parcela č. 716/1 k.ú. Černá.

9.2.2 Účel stavby

Jedná se o sakrální stavbu, přesněji kapli ve tvaru lodě.

9.2.3 Podklady pro projekt

Situace – zaměření návsi. Inženýrsko-geologický a radonový posudek byly provedeny v srpnu roku 2003.

9.2.4 Stavebně technické řešení

Stavba kaple se skládá ze dvou základních konstrukčních částí. Železobetonová konstrukce spodní stavby sestávající ze základové desky tl.300mm, svislých stěn nad základovou deskou tl.300mm a vodorovné podlahové desky tl.250mm.

Celodřevěná konstrukce horní stavby – její konstrukční systém je tvořen soustavou příčně orientovaných dřevěných rámů v osové vzdálenosti 1200 mm, které jsou podélně provázány v ose stavby a ztužujícími prvky v obvodových stěnách. Dřevěné sloupy jsou v patě vetknuty do podlahové železobetonové desky, která monoliticky navazuje na svislé nosné stěny a železobetonovou základovou konstrukci.

9.2.5 Zemní práce

Výkop stavební jámy byl proveden v písčitých hlínách a hlinitých píscích ve třídě rozpojitelnosti č. 2 a kamenitých a štěrkových hlinitých píscích rozpojitelných ve třídě č. 3. Únosnost půdy pod základovou spárou (hlinité písky převážně štěrkovité, třída S4) je Rdt=225 kPa. Svahy byly upraveny v poměru 1:2 a výkopy nad 1,5 m byly paženy. Pro násyp bylo použito materiálu ze stavebního výkopu. Doporučené hutnění bylo po vrstvách 150mm, za mírné vlhkosti zeminy. Úklony svahů

případného násypu bylo doporučeno upravit v poměru 1:2 a násyp zatravnit. Stálá hladina podzemní vody se udržuje ve shodě se spádem terénu v hloubce 2,5m pod současným terénem. Podzemní voda má mírně kyselou reakci (pH=6,4) a mírnou až střední tvrdost. Díky obsahu CO2 a CaCO3 (54,4mg/l) má voda vysoké agresivní účinky na beton. Stavební pozemek je zařazen do kategorie „střední radonový index".

9.2.6 Základová konstrukce

Konstrukce základu je tvořena železobetonovou monolitickou deskou. Deska je provedena z betonu C30/37 XA2 s max. vodním součinitelem w=0,5 a množstvím cementu min. 320kg/m³ za použití superplastifikátoru do betonu pro dobrou dlouhodobou zpracovatelnost betonu (vodotěsný, zhutněný beton). Kvůli agresi spodní vody byla zvětšena krycí vrstva betonu na 50mm. Pracovní postup betonáže byl rozdělen do 3 etap: základová deska, svislé stěny nad základovou deskou a podlahová deska s podestou.

9.2.7 Svislé nosné konstrukce – dřevostavba

Konstrukce je tvořena prostorovou soustavou dřevěných rámů v osové vzdálenosti 1200mm. Dřevěné sloupy, o rozměru 240x120mm, jsou spojeny ocelovými styčníkovými plechy s příčlemi, které kopírují tvar střechy. Příčle jsou vzájemně spojeny dřevěnými příložkami. Celý rám je v koruně jištěn kleštinami (120x60mm). Sloupy jsou v patě vetknuty do podlahové železobetonové desky pomocí ocelových patek a šroubových spojů. Zavětrování stěn je tvořeno vnějším šikmým bedněním a zavětrování střechy je provedeno ondřejovými kříži ve vybraných polích.

9.2.8 Vodorovné nosné konstrukce

Nosná konstrukce podlahy je tvořena železobetonovou deskou monoliticky spojenou se základy.

9.2.9 Konstrukce spojující různé úrovně

Nosná konstrukce nástupní rampy je navržena z ocelových schodnic HEB 160 a příčníků L60x60x6. Nášlapná vrstva je tvořena dřevěnými rošty. Rampa překonává výškový rozdíl 750mm. Na začátku rampy je schodek o výšce 20mm. Sklon rampy je 8,33%, délka 9m. Šířka rampy je navržena 1,3m.

9.2.10 Střešní konstrukce

Sklon střešního pláště je tvořen dřevěnými příložkami o rozměrech 240x60mm, kotvenými do rámových příčlí. Na příložky je provedeno dřevěné deskové bednění.

2.10.11 Obvodový plášť

Je tvořen větranou vícevrstvou konstrukcí. Vrstvení je následující: Začneme z exteriéru - vnější obklad – dřevěné desky, polodrážka, celoplošné bednění – dřevěné desky, větraná vzduchová mezera, tepelná izolace – ORSIL UNI, parotěsná zábrana – Jutafol N 110, vnitřní desky – dřevěné desky, pero a drážka.

9.2.12 Příčky a dělící konstrukce

Vzhledem k tomu, že se jedná o jednoprostorovou stavbu, nejsou ve stavbě navrženy klasické příčky. Prostor sakristie je pouze opticky oddělen vestavěnou celodřevěnou konstrukcí kruhové zástěny do výšky 2960mm.

9.2.13 Izolace proti vodě

Střešní krytina je navržena z titan-zinkového plechu od firmy Rheinzink a byla provedena podle podnikových zásad firmy pro zpracování titan-zinkového materiálu. Pod plechovou krytinou je umístěna strukturovaná dělící vrstva – DELTA TRELA (dodatečné lepení přesahů lepidlem DELTA – FOXX Pren). Izolace spodní stavby je řešena dvěma materiály. Izolační vrstva pod železobetonovou deskou je tvořena fóliovou izolací Sikaplan Tunnel 9,6, která byla přetažena na svislou část základové desky a přivařena do zde zabetonovaného povrchového pásu Sika AR – 24. Další plochy povrchu spodní stavby byly do úrovně

terénu opatřeny ochranným nátěrem Igolflex (min. 3kg/m²). Veškerá hydroizolace spodní stavby je kryta geotextilií.

9.2.14 Izolace tepelné

Tepelná izolace stěny je provedena z ORSIL UNI tl.140mm. Izolace stropu je navržena z ISOVER RIO 18 tl.180mm.Izolace stěn i stropů je překrytá parotěsnou folií Jutafol N 110, z vnitřní strany. Tepelná izolace podlahy je navržena z extrudovaného polystyrenu (Styrodur C 2800) tl.120mm.

9.2.15 Zámečnické konstrukce

Zábradlí vstupní rampy a podesty byly kotveny do ocelové konstrukce vstupní rampy a monolitické konstrukce podesty. Konstrukce zábradlí je kovová z masivních profilů, svařovaná na místě.

9.2.16 Lokální vytápění

S ohledem na způsob využití stavby, frekvence vytápění a potřebu rychlého nárazového vytopení objektu, bylo navrženo teplovzdušné vytápění pomocí vzduchotechnického zařízení. Toto zařízení je umístěno v dutině pod podlahou. Vyústky jsou provedeny jako 50mm široká štěrbina po obvodu interiéru. Nasávání čerstvého vzduchu je v dutině pod zvýšenou podlahou a cirkulační vzduch je nasáván z prostoru podhledu. Nasávací otvory jsou vedeny pod tepelnou izolací v místě přerušení nadbetovávky z lehčeného betonu. Cirkulační vzduch je v místě podlahy veden skrz tepelně izolační vrstvu, potrubí je podloženo 20mm silnou vrstvou XPS.

9.2.17 Dešťová kanalizace

Střešní vpustě jsou zaústěny do svislých svodů vedených uvnitř obvodového pláště. Svislé svody jsou volně vyústěny pod spodní úrovní podlahové železobetonové desky a dešťová voda je zachycena ve vpusti pod svodem. Vpust je zasypána štěrkem. Z vpustí je voda odváděna kanalizačním potrubím o minimální světlosti 125mm zaústěném do sběrné šachty přepadem. Sběrná šachta byla zhotovena ze skruží o průměru 1000mm a zakryta poklopem v úrovni terénu. Ze sběrné šachty je odtok, vyústěný do potoka. Sklon dešťové kanalizace je minimálně 2%. Kanalizační potrubí je z plastu a odtok z šachty v kamenině. Byla vložena i zpětná klapka na odtok a obalení vyústek sítí, bránící vniku hlodavců.

9.3 Architektonická část

9.3.1 Urbanistické řešení

Stavba kaple je situována v centrální poloze na návsi obce Černá. Prostor návsi má podélný charakter severojižního směru, na podélné východní straně návsi protéká od severu k jihu potok Křivý, který na jižní straně pod návsí ústí do Mlýnského rybníka. Po západní podélné straně návsi prochází průjezdná komunikace s autobusovou zastávkou, po protější východní straně návsi je vedena místní obslužná komunikace. Na kratších stranách návsi – na severním a jižním konci – jsou situovány dva mostky obslužné komunikace přes potok Křivý. V centrální ploše návsi je dále umístěna v jihozápadním cípu jednopodlažní stavba hasičské zbrojnice s obecním úřadem. Umístění budovy stávající hasičské zbrojnice a obecního úřadu v jihozápadní části návsi není příliš dobré řešení. Celkový dojem z návsi je touto stavbou předurčen a omezuje celkové výtvarné vyznění stavby kaple. V těsné blízkosti hasičské zbrojnice na severní straně je umístěn kamenný pomník padlým a dále na sever kamenný kříž se dvěma vzrostlými lipami. Podél potoka na východní straně je vysázena řada poměrně mladých stromků (lípy). Samotný tok potoka je částečně zpevněn na sucho skládanými kameny. Přes potok je volně položená ocelová lávka pro pěší provizorního charakteru – bez logické vazby na pěší komunikace. Situování stavby kaple reaguje na tvar návsi a směr protékajícího potoka svojí podélnou orientací severojižním směrem – stavba je situována v blízkosti potoka a souběžně s ním po jeho západní straně. Před kaplí tak vzniká odstupem od průjezdné komunikace předprostor. Požadavkem investora je zvýšit úroveň na 1.np nad úroveň terénu v místě stavby (cca 521,80 m.n.m.). Tento požadavek je v návrhu respektován zvýšeným úrovně 1.np cca 0,8m nad úrovní terénu v místě stavby (+-0,000 = 522,60 m.n.m.). V důsledku tohoto zvýšení úrovně 1.np je vstup do kaple řešen pomocí rampy, která je orientována souběžně s podélnou osou stavby. Rampa má parametry požadované pro vstup tělesně postižených občanů – sklon rampy je 1:12 (8,33%), celková délka 9m. Umístění stavby kaple na návsi se dobře doplňuje se stávajícím kamenným křížem pod lipami. Pomník padlým byl zachovám na původním místě.

9.3.2 Architektonické a výtvarné řešení

Architektonický návrh stavby vychází z urbanistické situace – svojí podélnou nízkou hmotou stavba reaguje na podélný prostor návsi a na potok v těsné blízkosti. Svojí formou i konstrukcí stavba symbolizuje loď – dřevěnou archu. Také z tohoto důvodu byla zvolena celodřevěná konstrukce stavby na betonovém základu. Mírným pozvednutím nad terén je stavba odlehčena – optické spojení se zemí je záměrně potlačeno – tím je také podtržen charakter stavby jako lodi. Vnější plášť stavby tvoří vodorovně ložená prkna na polodrážku. Podobně i vnitřní stěny, podlaha i strop jsou provedeny z podélně kladených prken na pero a drážku.

9.3.3 Funkční a dispoziční řešení

Jedná se o jednopodlažní stavbu, interiér stavby tvoří jeden prostor, který je funkčně členěn pouze vestavěným mobiliářem. Vstup do objektu tvoří ven otevíravé masivní dveře š. 125cm. V centrální části interiéru je prostor s oboustranně umístěnými lavicemi pro sedící – celkem pro 62-64 osob (úroveň +-0,000), v jižní části stavby je situován zvýšený prostor presbytáře (úroveň +0,150m), ve kterém je v centrální poloze umístěn obětní stůl, po stranách sedés a místa pro ministranty a svatostánek. V severní části stavby u vstupu do objektu je umístěna zástěna sakristie kruhového půdorysu, vybavená uzamykatelnou skříní a židlí. Za ní je situováno místo pro zpověď – stolek se dvěma židlemi.

9.3.4 Truhlářské konstrukce

Veškeré interiérové vybavení stavby je provedeno jako truhlářský výrobek. Jedná se zejména o celodřevěnou zástěnu sakristie, celodřevěné lavice. Veškeré truhlářské práce provedla firma p. Vaculy z Měřína.

9.3.5 Kamenné konstrukce

Obětní stůl je z masivní kamenné desky. Oltář je vyroben ze syenitu a v jeho noze jsou uloženy ostatky sv. Zdislavy a Jana Sarkandra. Přístupový chodník ke stavbě je proveden z volně

ložených plochých kamenů rozměrů cca 600/1000mm do pískového lože.

9.3.6 Podlahy dřevěné

Vnitřní podlahová konstrukce je zhotovena z dřevěných fošen na pero a drážku. Tloušťka fošen je 50mm. Fošny byly vybrány pokud možno s ostrými léty na nášlapné ploše. Podlaha je přibita na dřevěné polštáře výšky 70mm, které jsou kryté páskem Miralou.

9.3.7 Podhled

Podhled v interiéru stavby je proveden z prken tl.25mm, mezi prkny je volná spára šířky 5mm – spárami v podhledu je odváděn vzduch z interiéru stavby.

9.3.8 Obklady

Vnější obklad stěn je navržen z prken měkkého dřeva tl.25mm a šířky 150mm (osová vzdálenost 125mm) na polodrážku. Desky mají ze zadní strany vyfrézovány 2 drážky pro zmenšení tvarových rozdílů při bobtnání a sesychání. Jako spojovací prostředek jsou použity vruty se zápustnou hlavou.

Vnitřní obklad stěn je navržen z prken z měkkého dřeva tl.25mm a šířky 150mm na pero a drážku. Desky mají ze zadní strany vyfrézované 2 drážky stejně jako u obkladu vnějšího. Jako spojovací prostředek je použito hřebíků, případně vrutů se zápustnou hlavou – kotvení provedeno skrytě do drenážky.

9.3.9 Nátěry

Veškeré dřevěné prvky byly před zabudováním ošetřeny bezbarvými nezávadnými přípravky proti působení hmyzu, dřevokazných hub a plísní.

Povrchové nátěry dřevených prvků v exteriéru jsou provedeny bezbarvou impregnací proti působení plísní, hub a dřevokazných brouků.

Povrchové nátěry v interiéru jsou provedeny transparentním bezbarvým matným lakem.

Povrchové úpravy ocelových prvků v exteriéru byly provedeny odolným černým matným nátěrem pro venkovní použití.

9.3.10 Zvon

Zvon zavěšený v exteriéru kaple je elektricky poháněn. Zvon zhotovila firma Dytrychů z Brodku u Přerova.

9.3.11 Křížová cesta

V moderním pojetí křížovou cestu ztvárnil Miloš Sláma z Polničky. Použil techniku, zvanou perokresba. V tomto cyklu je 15 obrazů, každý jiné velikosti. Jsou zavěšené na ocelovém nosníku, který je zbarven černě. Výborně se slučuje s obrazy a krásně vyniká v interiéru kaple.

9.3.12 Osvětlení interiéru kaple

Podhled je odsazen od svislé konstrukce, tzn., že se jí nedotýká nebo že s ní není přímo spojen. Tento volný cca 100mm prostor je využit vpravením zářivek po celém obvodu kaple v těchto krajích podhledu. Po jejich rozsvícení tvoří dojem, že se jedná o prostup slunečního světla z venku. Takto osvětlená kaple klade důraz na jednoduchost, ale přesnost a ladnost konstrukce.

V kapli je také okno, a to na západní stranu. Okno má elektricky řízené zatažení černého stínítka. Když je okno odstíněné, sluneční světlo dopadá přímo na oltář, ale v celkovém dojmu narušuje půvab osvětlení umělého.

X. TVŮRCI KAPLE

10.1 Kuba & Pilař

Ze čtyř přihlášených účastníků (na návrh kaple) byl vybrán ateliér Kuba & Pilař z Brna. S nimi byla uzavřena smlouva o dílo. Výsledky práce posuzoval poradní sbor Biskupství brněnského, který 3.dubna 2003 návrh schválil.

Další stavby, navržené tímto ateliérem jsou: Knihovna Filozofické fakulty Masarykovy Univerzity Brno, exteriér i interiér. Obytný soubor Tichá Šárka v Praze, Brno city center na náměstí Svobody v Brně, Fakulta chemicko-technologické Univerzity v Pardubicích, Zastupitelský úřad ČR v Litvě, kašna na náměstí Svobody v Brně a pavilon F na brněnském výstavišti.

10.2 Identifikační údaje stavebníka

Obec Černá
594 42 Měřín
Okres Žďár nad Sázavou

10.3 Identifikační údaje zpracovatelů

10.3.1 Generální projektant

Akad. arch. Ladislav Kuba

10.3.2 Projektanti částí

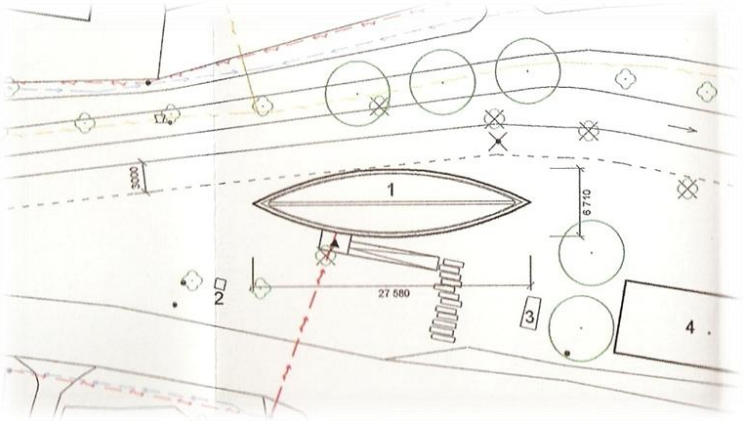
Architektonicko-stavební:	Akad. arch. Ladislav Kuba
Stavební řešení:	Ing. Michal Zajíc
Statika (bet. kce):	Ing. Hradil
Statika (dře. kce):	Ing. Miroslav Bajer, CSc.
Vytápění:	Ing. Dostál
VZT:	Ing. Jiří Hájek
Elektroinstalace:	Bedřich Raida
Požární bezpečnost:	Ing. Lubomír Bauer

Stavební práce provedla firma Stylstav Křižanov.

10.4 Identifikační údaje stavby

Katastrální území č.619809, okres 33714 Žďár nad Sázavou, obec 594438 Černá.
Vlastníkem je obec Černá, pozemek má ve svém vlastnictví, výměra – 8762m² a zastavěná plocha je 121m².

PŘÍLOHA – FOTOGALERIE



LEGENDA:

- stromy stávající
- stromy navrhované
- stromy rušené
- rušený sloup místního rozhlasu
- přípojka NN
- vedení NN
- sdělovací vedení
- vodovod
- ochranné pásmo potoka
- 1 kaple Sv. Antonína
- 2 kamenný kříž
- 3 památník padlých
- 4 obecní úřad, hasičská zbrojnice



Železobetonová konstrukce



Svislá železobetonová konstrukce





Svislá železobetonová konstrukce s ocelovými nosnými prvky





Průduchy (topení)





Vodorovná nosná konstrukce, na které jsou „patky“, do kterých se uchytí trámy





Uchycení trámů do patek, a vztyčení svislé dřevěné konstrukce





Střešní bednění



Vkládání izolace



Pohle na kapli z východu



Opláštění severským smrkem







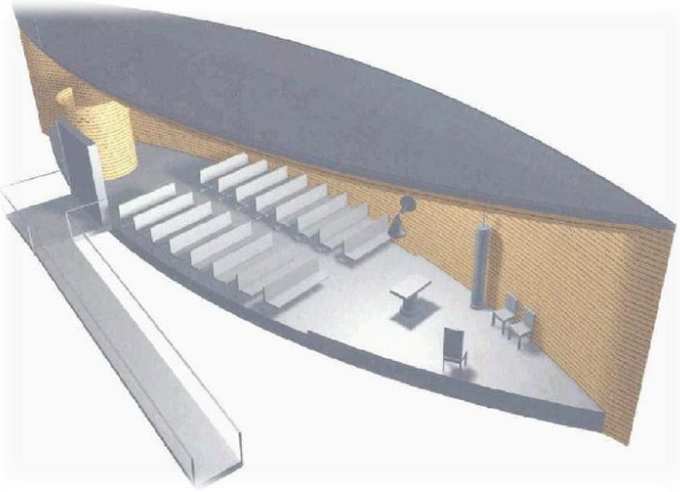


Práce v interiéru – izolace, dřevěný obklad



ARCHITEKTONICKÁ ČÁST







Kaple sv. Antonína v Černé





Okno



Skříň s rozvody



Zvon, umístěn v exteriéru kaple



Interiér, pohled na oltář





Osvětlení v kapli



Truhlářská práce













Zpovědnice



Socha sv. Antonína
(jediná věc, převzatá ze staré kaple)



Syenitový stůl



Kruhová sakristie





Křížová cesta







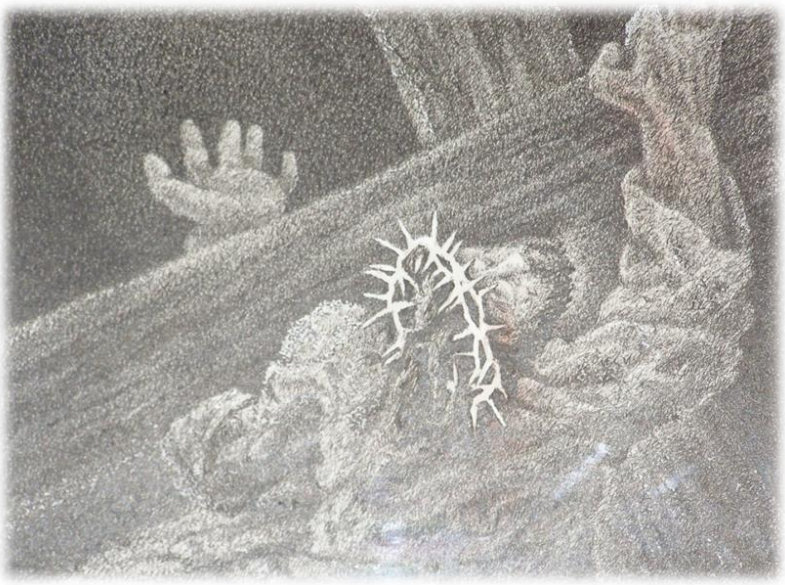






















XI. ZÁVĚR

… na konci jsem zjistila, jak se pojetí tohoto tématu dá široce zpracovat. Teď už se ale moje práce ubírá ke konci. Mým úkolem bylo zpracovat na téma dřevěné stavby tuto práci.

V ní jsem uvedla všeobecnou charakteristiku dřevěných staveb, a blíže jsem se věnovala sakrální novodobé dřevostavbě a to kapli sv. Antonína v obci Černá.

Závěrem a také získáním nových informací pro mě je, že tuto práci mohu předat jako hotový celek doplněný o prezentaci, která z mého pohledu je výrazně graficky pojata a zpracována, dále webové stránky, kde jsem jako hlavní prvek použila „klikací mapu". Moji práci jsem také doplnila o model kaple sv. Antonína.

Tato dlouhodobá práce mě motivovala k dalšímu vzdělávání a získávání nových informací, ze kterých bych v budoucnu měla vycházet.

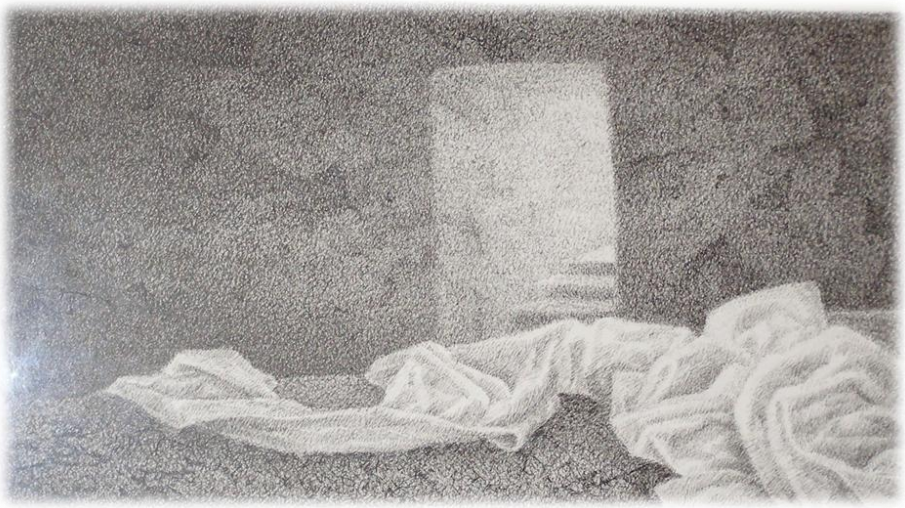
Můj blízký vztah ke dřevu, stavitelství a architektuře je také další aspekt k výběru tohoto témata. Stavby ze dřeva jsou krásné a navrací se zpět k přírodě.

Také jsem se pokoušela navrhnout kapli, nechtěla jsem kopírovat kapli v Černé, ale bohužel se mi to trochu povedlo, i když vlastní iniciativa byla poněkud jiná. Proto také přikládám pár „črtů", pro vaše srovnání.

Touto prací bych taky chtěla vyvrátit myšlenku „dřevostavba=špatné řešení".

CESTA OTEVŘENA DALŠÍMU POZNÁNÍ

… konec



Pozn. Tato perokresba je posledním z cyklu křížové cesty od p. Slámy, proto i já bych s ním chtěla zakončit svoji práci.

XII. PRAMENY

Kompletní projektová dokumentace (kaple sv. Antonína v Černé), akad. arch. Ladislav Kuba, 2003

Černá – Milíkov, Mgr. Oldřich Hnízdil, 2007, 420 str.

Stavíme dům ze dřeva, Martin Růžička, Profi a hobby, GRADA, 2006, 117 str.

Nízkoenergetické domy, Jan Tywoniak, GRADA, 2005, 193 str.

Dřevěné stavby, Jozef Štefko, Ladislav Reinprecht, Petr Kuklík, JAGA, 2006, 204 str.

Stavíme ze dřeva, Václav Hájek, SOBOTÁLES, 1997, 153 str.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Návrh rodinného domu, kde žije tělesně omezený.

Jiří Holeš

Střední škola stavební Jihlava
Žižkova 20, Jihlava

Střední škola stavební Jihlava,
Žižkova 20, 586 01 Jihlava



Dlouhodobá ročníková práce

Návrh rodinného domu, kde žije tělesně omezený.

Obor: 78-42-M/001 (39-97-5/01) technické lyceum

Třída: 3. TL

Školní rok: 2007/2008

Září 2007 Jiří Holeš

Střední škola stavební Jihlava
Žižkova 20, 586 01 JIHLAVA
URL: www.ssstavji.cz
e-mail: jsvobodova@ssstavji.cz
tel. 567 578 560, fax 567 3101 100

Zadání dlouhodobé praktické zkoušky

Jméno a příjmení žáka: Jiří Holeš

Třída: 3. TL

Studijní obor: TECHNICKÉ LYCEUM
Školní rok: **2007/20008**

Téma:

Návrh rodinného domu, kde žije tělesně omezený.

Technická specifikace, osnova, rozsah práce:

důvod výběru tohoto tématu
obecná charakteristika problematiky bezbariérového bydlení
studie rod. domku / půdorysy, řezy, pohledy, měř. 1:50/
konzultace konkrétní realizace – fotografická i technická dokumentace, názor majitele, dodavatele….
připomínky tělesně omezeného
závěrečný názor studenta

Realizace: textová dokumentace svázaná kroužkovou vazbou, výkresy poskládané ve složce, prezentace v programu PowerPoint na CD nosiči. / konkrétní osnova dořešena dle rozsahu práce/

Vedoucí práce: Ing. Jana Svobodová

Konzultant:

Termín zadání: **27. 9. 2007**
Termín odevzdání: **16. 5. 2008** …….
PhDr. Pavel Toman
ředitel školy

Obsah

Anotace práce, 4
Důvod výběru tohoto tématu 5
Obecná charakteristika bezbariérového bydlení 6
Co je to vlastně bariéra? 7
Nejdůležitější úpravy domu/bytu pro omezené 8
Přístup do domu 8
Změny ve vybavení kuchyně 10
Průvodní zpráva 12
I. ARCHITEKTONICKÉ ŘEŠENÍ 12
II. ORIENTACE KE SVĚTOVÝM STRANÁM A DENNÍ OSVĚTLENÍ 12
III. FUNKČNÍ ŘEŠENÍ 12
IV. STAVEBNĚ TECHNICKÉ ŘEŠENÍ 13
- **ZEMNÍ PRÁCE** 13
- **ZÁKLADY** 13
- **SVISLÉ KONSTRUKCE** 13
- **VODOROVNÉ KONSTRUKCE** 13
- **SCHODIŠTĚ** 13
- **ZDVIŽ** 14
- **STŘEŠNÍ KONSTRUKCE** 14

VI. PODLAHY, OBKLADY, DLAŽBY, OMÍTKY (ZATEPLENÍ) 14
OKNA, DVEŘE, ZÁRUBNĚ 14
VIII. VYTÁPĚNÍ 15
IX. TECHNICKÉ VYBAVENÍ, ZDRAVOTECHNIKA, KONSTRUKCE A PRÁCE PSV 15
X. KANALIZACE 15
XI. VODOVOD 15
ZEMNÍ PRÁCE PRO PŘÍPOJKY INŽENÝRSKÝCH SÍTÍ 15
IZOLACE STAVBY PROTI ZEMNÍ VLHKOSTI A RADONU 16
Závěrečné hodnocení studenta 16

Anotace práce,
neboli úvodní slovo

V této práci jsem měl za úkol navrhnout funkční a uživatelný rodinný dům pro tělesně omezeného. Bylo nutno prostudovat příslušné normy a zákony, abych mohl vycházet ze stávajících pravidel, pro bezbariérové stavění.

Poděkování
Chtěl bych poděkovat paní učitelce Svobodové a paní učitelce Lorencové za pomoc s výkresovou částí.
Paní učitelce Svobodové také za všeobecnou pomoc s prací.
Panu Ivo Krajíčkovi za sdělení cenných zkušeností a seznámení s pohledem z „druhé strany".
Děkuji

Prohlášení autora

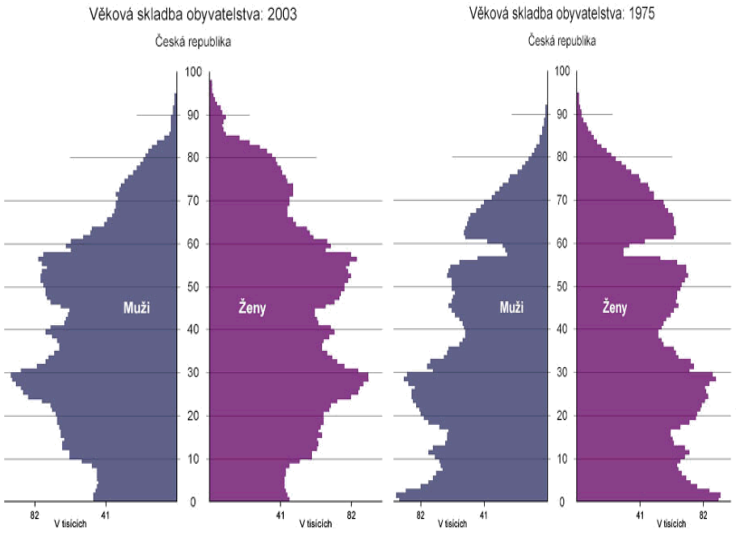
„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.
V Jihlavě dne 16. 5. 2008

……
podpis

Důvod výběru tohoto tématu

K tomu, abych si zvolil toto téma, mě vedlo několik věcí. Asi první byl zájem pochopit tuto problematiku a zjistit o ní více informací a to jak praktických, tak i

teoretických. Pravidla a zásady správného bezbariérového bydlení jsou v současné době stále více přijímána jako určující při stavění s ohledem na budoucnost. A netýká se to pouze lidí na vozíčku, slabozrakých, hluchoslepých a dalších, které jsem nejmenoval, ale každého z nás. Protože i jeden hloupý pád může doposud zcela zdravého člověka upoutat nadosmrti na vozík.
Také ještě přichází v úvahu opomíjený fakt, a to že populace stárne. A pro starého člověka, trpícího těžkými nemocemi může být překonání jednoho jediného schodu obtížným a vyčerpávajícím úkolem. Jako důkaz přikládám dva grafy, populační strom v roce 2003 a 1975.



Obr. č. 1. 1 Pramen: ČSÚ Obr. č. 1. 2

Dalším důvodem bylo získání praxe v kreslení a navrhování staveb obecně. Tato praxe mi bude jistě užitečná na vysoké škole.

Dalším a posledním důvodem bylo, že já sám mám určitou střední formu smyslového postižení, a proto pro mě bylo snadnější se vcítit do pocitů a problémů těchto pohybově, nebo smyslově omezených lidí.

Obecná charakteristika bezbariérového bydlení

Bydlení je jednou ze základních lidských potřeb a také patří mezi základní lidská práva všech různých skupin obyvatelstva. Proto je třeba přistupovat ke každé konkrétní stavbě specificky, podle požadavků investora. Celkový rozsah a rozmanitost v základních funkcích bydlení je dána rozdílnostmi v sociálním složení uživatelů, jejich finanční situací a způsobem životního stylu. Jednou z takových forem je bezbariérové bydlení pro osoby se sníženou schopností pohybu – osob se zdravotním, nebo smyslovým omezením a lidí staršího věku. Jak

jsem uvedl výše, populace stárne a také osoby s různým stupněm nějakého omezení tvoří nezanedbatelnou část populace, a proto je uvědomění si problematiky odpovídajícího bydlení obzvlášť důležité.
Hlavním cílem v této oblasti je vytvářet podmínky pro důstojné a kvalitní bydlení, poskytovat podporu při přizpůsobování bydlení potřebám těchto skupin, zlepšovat podmínky bydlení s důrazem na jeho individuální charakter a zamezit odchodu do různých ústavů a jiných zařízení, kde jsou tito lidé odloučeni od rodin a ztrácejí společenské kontakty.

Společnost by měla vytvářet důstojné podmínky a kvalitní bydlení seniorů, zdravotně a smyslově omezených, jehož výsledkem nebude odsunutí těchto skupin obyvatelstva do domů a zařízení odloučených od společnosti, sociálních kontaktů a rodiny.

Dále je nezbytné zlepšovat podmínky bydlení s důrazem na jeho individuální charakter a požadavky dané charakterem omezení, vyhnout se nepřístupnému, bariérovému prostředí, ale naopak umožnit seniorům a postiženým lidem možnost se přizpůsobit okolnímu prostředí a plně se podílet na běžném každodenním životě bok po boku se zdravými spoluobčany.

Zároveň by měla být věnována pozornost tomu, aby nedocházelo k vytváření nějakých uzavřených bloků pro lidi s jedním typem omezení. Tyto osoby mají plné právo se rozhodnout žít a bydlet v místě, kde si přejí, v místě které plně odpovídá jejich požadavkům a potřebám. Prostředí bytových prostor má důležitý úkol působení na psychiku jeho uživatelů a jim má odpovídat dobré prostorové a dispoziční řešení, dostatečná velikost a potřebný rozsah příslušenství.

V našem, jak zákonném, tak i projektovém prostředí zůstává bezbariérové bydlení stále poněkud nedořešenou otázkou. Při projektování obytných a rodinných domů se často vychází pouze ze situace většinové skupiny společnosti a na budoucnost se příliš nepomýšlí.

Srovnáme-li naše legislativní prostředí například se švýcarským, kde nelze zkolaudovat dům, jenž nesplňuje striktní normy bezbariérového bydlení, uvědomíme si, že u nás jsou v tomto směru stále značné mezery. Významným faktorem při projektování stavby je také skutečnost, že každý z nás se může stát obětí nehody, může onemocnět anebo se k nám v budoucnu může přistěhovat člen rodiny, který je pohybově handicapován.

Co je to vlastně bariéra?

Celou prací prochází slovo bezbariérové. Ale co je vlastně bariéra? Zákon definuje bariéru jako vodorovnou překážku která je vyšší než 20 mm. Ale v praxi je to samozřejmě jinak. Bariéry jsou dvojího druhu. Jsou to bariéry fyzické a psychické.

- bariéry fyzické – to jsou bariéry, které vytvořil projektant, nezodpovědní dělníci, nebo příroda. S jejich překonáváním nemají omezení takové potíže, pokud mají odpovídající vybavení, anebo asistenta, který jim se vším pomůže.
- bariéry psychické – to jsou bariéry, které nejsou na první pohled vidět, a jejich odstranění není tak jednoduché, jako u bariér fyzických. Tyto bariéry mají jak omezení lidé, tak i tzv. „zdravá“ společnost. U omezených může trvat i několik let, než se se svým stavem smíří a začnou žít plnohodnotný život jinak, než byli doposud zvyklí. Bohužel se musejí vyrovnat i s neochotou, podezíravostí a nepříjemností ze strany většinové „zdravé“ populace. Na odstranění těchto bariér musí pracovat celá společnost, včetně státu.

Nejdůležitější úpravy domu/bytu pro omezené

Naprostá většina společnosti si myslí, že bezbariérové úpravy si vyžádají milionové investice. Není tomu tak. Pohybově omezení potřebují úpravy přístupových cest, koupelny a kuchyně, popř. vybudování přístupu do patra.

Přístup do domu

Do domu by mělo být možné projet po rampě, která by měla mít maximální normovaný sklon 1:8, vhodnější je však sklon nižší, nejlépe 1:12. Tato rampa samozřejmě může být vhodně zkombinována se schody, ale to již závisí na konkrétním projektu a na dohodě mezi projektantem a investorem. Rampa se může stát součástí částí stavby a zároveň i zajímavým architektonickým prvkem. Další důležitou věcí je šířka dveří, která musí vycházet ze stanovených norem. Většinou se doporučuje šířka minimálně 900 mm. Stejná šířka by měla být použita i v případě výtahu, popř. zdviže.

Změny ve vybavení koupelny

Jeden z důležitých předpokladů kvalitního života představuje pro handicapované bezpochyby hygiena. Aby mohl handicapovaný člověk bez obtíží provádět hygienu, je třeba přizpůsobit vybavení koupelny tak, aby se zde mohl bez překážek pohybovat. Příkladem špatné prostorové dispozice pro omezeného mohou být koupelny v bytech ze 70. Let.



Vybavení koupelny 1

Parametry umývadla

Umývadla určená pro vozíčkáře a omezené jsou plochá, ukotvená do stěny bez tzv. luxusní nohy a instalují se níže než umývadla klasická. Speciální tvarování přední části spolu s krátkým sifonem usnadňuje lepší zajetí, ovládání baterie a mytí rukou. Důležitá je také páková míchací baterie a co nejdelší délkou páky, instalované buď uprostřed, nebo vpravo, či vlevo, podle potřeb omezeného. Aby bylo možné na vozíku pohodlně odjet, je třeba hloubky nejméně 300 mm od okraje umývadla. Doporučená výška přední hrany umývadla

je 800 mm, jeho minimální hloubka pak 550 mm.

Sprcha, sprchový kout a vana

Sprchový kout musí být opatřen posuvnými dveřmi bez prahu, popř. závěsem s vodící linií nahoře, což vyžaduje 2% sklon podlahy, aby nedocházelo k vytékání vody ze sprchového koutu. Důležitou součástí sprchového koutu jsou rovněž sklápěcí sedátko o rozměru minimálně 400 – 450 mm a opěrná madla. Dno sprchového koutu musí mít protiskluzovou úpravu, kterou jistě uvítají i zdraví členové rodiny.

Standardní vana pro handicapované není vhodná. Aby však nedošlo ke snížení kvality života postiženého, můžeme najít dobré řešení, kterým je zapuštěná nebo polozapuštěná vana, kolem níž jsou upevněna madla. Pro zcela imobilní lidi lze nad vanu instalovat elektrický zvedák na kolejnici.

Toaleta

Sedátko na záchodové míse by nemělo být výše než 480 mm od podlahy, ale i to záleží na schopnostech a míře omezení. Samozřejmostí jsou sklopná madla a prostor alespoň na jedné straně toalety 800 mm. Přední strana sedátka musí být od zadní stěny vzdálena minimálně 700 mm. Vzdálenost mezi mísou a boční stěnou musí dosahovat nejméně 300 mm. Držák toaletního papíru se upevňuje do výšky 800 mm, mýdelník a věšák na ručníky pak do výšky 900 mm.

Změny ve vybavení kuchyně

U kuchyně je nejprve třeba si rozmyslet, jakou měrou ji bude omezený užívat. Zdali pouze sám, nebo i s ostatními členy rodiny. V prvním případě lze kuchyni zcela přizpůsobit potřebám a požadavkům, vyplývající z míry omezení. Pak vzniká zcela originální a jedinečná kuchyň, která se na první pohled ani příliš nemusí lišit od klasické kuchyně pro „zdravé“. Mírnou nevýhodou je vyšší cena kuchyně i komponentů.

Jiné řešení přichází na řadu v případě, že handicapovaný bude kuchyni používat spolu s ostatními členy rodiny a bude se zde tedy střídat větší množství lidí. Pracoviště je třeba zařídit tak, aby splňovalo nároky odpovídající klasickému, běžnému provozu, ale zároveň aby se zde handicapovaný člen rodiny mohl bez problémů pohybovat a vařit i když bude doma sám.

Za těchto okolností můžeme ve všech směrech vyhovující kuchyni vytvořit kombinací zařízení z běžné produkce a kuchyňských segmentů, které jsou speciálně upraveny pro handicapované osoby. Nebudou zde chybět vestavné spotřebiče a níže umístěné horní skříňky. V dalším zařízení kuchyně se velmi dobře uplatní nejrůznější rohové skříňky s otočnými koši, rohové karusely či skříňky s výsuvnými koši. Neméně praktický je rovněž stojan ve formě kovové tyče, která je vedena od stropu a je opatřena několika patry otočných drátěných poliček na koření a drobnější předměty, jež musíme mít stále po ruce.



Úprava kuchyně 1



Úprava kuchyně 2

Speciální kuchyňské díly

Jedná se zejména o spodní zásuvkové skříňky různých šířek a výšek a o skříňky horní, které obsahují vyklápěcí polici, jež se po otevření vysune směrem dolů. Dále jsou to

výsuvné skříňky určené pod dřez, speciálně designované stolky, výsuvná digestoř, zabudovaný odpadkový koš, mělký dřez a snížené pracovní plochy.

Parametry pracovních ploch

Pracovní desky musejí být provedeny v různých výškách, které respektují individuální potřeby postiženého, výšku postavy a eventuální postižení paží. Výšky těchto komponentů se pohybují od 732 mm do 782 mm. Dřez může být instalován výšce 860 mm, případně ve snížené výšce 480 mm.

Funkční návaznost pracovišť

Prostor kuchyně by měl mít minimální velikost 2,4 x 2,5 m. Pro vytvoření odpovídající pracovní sestavy je však vhodnější rozměr 2,4 x 3,2 m, protože varné a mycí centrum pak můžeme umístit přímo vedle sebe. Minimální prostor mezi těmito pracovišti by měl činit 450 mm. K pohodlnému otáčení vozíku je třeba plochy o průměru 1500 mm uprostřed místnosti, k projetí pak nejméně 800 mm.

Pokud prostor kuchyně nedosahuje uvedených rozměrů, nebo je jeho dispozice řešena do L nebo do U, umístíme mycí a varné centrum vedle sebe přes roh. Vždy přitom musíme myslet na možnost čelního zajetí vozíku pod pracovní desky, například pod mycí centrum, aby zde bylo možné pracovat s pohodlně pokrčenýma rukama, bez natahování a předklánění.

Návaznost jednotlivých pracovišť je třeba velmi důkladně rozvrhnout: sporák, dřez a pracovní deska musejí být od sebe vzdáleny tak, aby všude bylo možné dosáhnout z jednoho místa. Pracovní místa pak vedle sebe řadíme podle toho, zda bude kuchyni využívat pravák nebo levák.



Detail prac. prostoru 1

Speciální komponenty a díly

Základním požadavkem na zařízení kuchyně pro rodinu s handicapovaným je dokonale

funkční prostorové uspořádání a vyřešení minimálních rozestupů mezi kuchyňskými centry. Nejdůležitějším kritériem je přitom zachovat dostatek prostoru – na úkor volného místa nikdy neinstalujeme fixní kusy nábytku. Raději upřednostníme různé typy výsuvných desek a polic, sklápěcí stolky či přídavné a jezdící stolky, které lze zasunovat pod pracovní desku. Na našem trhu jsou dostupné rovněž výsuvné kráječe a roboty nebo jiné menší spotřebiče. Veškeré vybavení můžeme po skončení práce zasunout a opět nám vznikne volná plocha.

Průvodní zpráva

Účel stavby: Novostavba leží ve výšce 536 metrů nad mořem – Baltský systém. Rodinný dům, umístěný v okrajové části města, snadno dostupná lokalita.

Zastavěná plocha: 166,4m^2
Obestavěný prostor: 215,5 m^2

I. ARCHITEKTONICKÉ ŘEŠENÍ

Rodinný dům je postaven v okrajové části města, ale snadno dostupné, buď autem, nebo několika linkami MHD. V okolí se nachází další rodinné domy, dětská hřiště, malá samoobsluha a rozlehlý park. Vše je s bezbariérovým přístupem. Parcela je mimo hlavní silniční tahy města. Budova je postavena v nové, vznikající „vilové“ čtvrti.

II. ORIENTACE KE SVĚTOVÝM STRANÁM A DENNÍ OSVĚTLENÍ

U místnosti, které jsou při běžném provozu nejvíce využívány a obývány je zajištěno dostatečné a příjemné proslunění slunečním světlem. Budova je situována tak ke světovým stranám, aby místnosti jako obývací pokoj, ložnice, kuchyně a dětské pokoje byly situovány na JZ, popř. JV.

Méně využívané místnosti, jako schodiště, zdviž, šatna a vstupní prostory jsou situovány směrem na sever.

III. FUNKČNÍ ŘEŠENÍ

- V 1.NP je umístěn bezbariérový vstup do budovy, který je kryt stropem lodžie. Hned ve vstupní chodbě je umístěno schodiště, v jehož zrcadle se nachází zdviž do 2. NP. Ve vstupní chodbě jsou umístěny skříňky na obuv a je z ní přístup do největší místnosti v domě, kombinace obývacího pokoje, kuchyňského koutu, malé spíže a jídelny. Dále je v 1. NP koupelna přizpůsobená potřebám vozíčkáře, včetně dvou

přístupů, jednoho z chodby a druhého z pokoje vozíčkáře. Tento pokoj je prostorný a přizpůsobený potřebám postiženého.

- V 2.NP jsou dva dětské pokoje, ložnice rodičů, šatna a malé WC.
- Střecha je jednoduchá jednoplášťová se spádem 1%.

IV. STAVEBNĚ TECHNICKÉ ŘEŠENÍ

• ZEMNÍ PRÁCE

Výkopy pro základové pásy budou dosahovat hloubky – 1.475 m pro obvodové nosné zdivo a hloubky – 0.875 m pro vnitřní nosné zdivo. Stěny vyhloubených jam budou svislé.

• ZÁKLADY

Pod obvodovým zdivem a vnitřními nosnými zdmi budou probíhat základové pásy s rovnými čely. Materiál beton B20, pod základy štěrkový podsyp.
Základová spára pro obvodové zdivo a sloup sahá do hloubky – 1.275 mm od 1. NP = ± 0,000, pro vnitřní nosné zdivo – 0.475 mm od 1. NP = ± 0.000.
Pro obvodové zdivo je šířka základového pásu 650 mm, pro vnitřní nosnou stěnu – 500 mm, pro sloupovou patku je vyhlouben otvor o rozměrech 700 x 700 mm.

• SVISLÉ KONSTRUKCE

Celý objekt je tvořen z cihel POROTHERM. Obvodové nosné zdivo - 44 P + D a TM malta, vnitřní nosné zdivo - 30 P + D + CB malta.
Příčkové zdivo - 11,5 P + D + CB malta; 8 P + D + CB malta.

• VODOROVNÉ KONSTRUKCE

Strop v objektu se zhotoven z cihel Hurdis a příslušných ocelových konstrukcí.
V objektu najdeme ŽB - RPZ překlady s tepelnou izolací. U velkých světlostí se některé zhotovují až na stavbě. Obvodovým zdivem probíhá věnec – výška 200 mm, ocelové pruty 10216 ø8, izolován tepelnou izolací PPS, zakončeno věncovkou POROTHERM 23,5.

• SCHODIŠTĚ

Schody v interiéru mají železobetonovou schodnici a ve schodišťovém zrcadle je umístěna interiérová zdviž. Celá viditelná část schodiště je obložena hoblovaným, borovým dřevem, včetně nášlapné vrstvy. Podesty leží v polovině světlé výšky podlaží.
Podesta v 1.NP + 1.400

- **ZDVIŽ**

V objektu bude umístěna zdviž od firmy MANUS Prostějov.
Zdviž má tyto technické parametry:

- rychlost plošiny: 0,06 m/s
- nosnost 250 kg
- příkon 800 W
- Elektrické jištění plošiny: 3x10A, 400V

- **STŘEŠNÍ KONSTRUKCE**

Objekt je zakončen plochou, jednoplášťovou střechou Jednotlivé vrstvy: keramzit, filtrační fólie (FILTEK), dilatační vrstva (FILTEK), hydroizolace (FILTEK), dilatační vrstva (FILTEK), tepelná izolace (EPS desky POLYDEK), parotěsná zábrana – (parofol PE 140A), železobeton C 16/25 + ocel. Objekt má pultovou střechu, která je odvodněna na JV straně obvodu střechy odvodňovacím kanálem.
Atika na konci oplechovaná, obíhá okolo celé střechy, kromě JV strany.

VI. PODLAHY, OBKLADY, DLAŽBY, OMÍTKY (ZATEPLENÍ)

Keramické dlaždice značky RAKO - o rozměrech 330 x 330, dále 98 x 198 jsou v koupelně, v kuchyni a na WC.
Jsou lepené na podlahu, flexibilní cementovou lepící maltou značky KIESEL. Proti vlhkosti stěrkové izolace.Podlaha na chodbě je z betonové dlažby 300x 300. Na terasách, speciální protiskluzová betonová dlažba (metropan 1) rozměry 250 x 250.
Keramický obklad vhodně zvolený k barvě podlahy v koupelně a na wc. V kuchyni i na wc je obklad umístěn do výšky 1,5m,opět je nalepován na stěny.
Vnitřní stěny budou dvakrát pačokované. Na malbu jsou použity různé odstíny barvy značky PRIMALEX.

OKNA, DVEŘE, ZÁRUBNĚ

- Použitá okna značky VEKRA mají izolační sklo se zesílenou tepelně izolační schopností Ug = 1,1 W/m2K. Tato skla mají cca 50% úsporu tepelné energie unikající sklem. Zvuková izolace skel asi 38 dB.
  Na objektu jsou namontovány tyto typy oken: otevíravá a sklápěcí.
  - Na oknech jsou nainstalovány elektricky ovládané rolety
- Dveře v interiéru jsou dřevěné, zasklené ze 2/3, od firmy SAPELI.
- Vstupní dveře jsou celodřevěné, otevíravé dovnitř.
- Využití kovových zárubní – TARGUS.

VIII. VYTÁPĚNÍ

Objekt je připojen na městský teplárenský systém. Objekt bude vytápěn pomocí podlahového vytápění

IX. TECHNICKÉ VYBAVENÍ, ZDRAVOTECHNIKA, KONSTRUKCE A PRÁCE PSV

- Zdravotní technika: umývadla ukotvená do stěny s krátkým sifonem zavedeným do stěny, speciální záchodové mísy pro postižené od výrobce – JIKA, mělká vana i sprchový kout. Samozřejmostí jsou sklopná madla z obou přístupových stran. Zabudovány úchyty ve stropní konstrukci pro možné pozdější namontování pojízdného kolejnicového zvedacího systému. Vše v bílé barvě. Nerezové kohoutky s dlouhou pákou SANELA.
- Elektroinstalace jsou navrženy odborníky ze společnosti ZKRAT s.r.o. Rozvodné sítě budou zabudovány ve stěnách objektu. V objektu bude i přípojka na veřejnou síť nízkého napětí.

X. KANALIZACE

- Je navržena a zhotovena odborníkem. Plastové potrubí svádí vodu z celého objektu + drenážní trubky - Durvinil vedeme pod objektem v hloubce asi --- do veřejné betonové kanalizace. Kanalizační trubky Gienger ø 400 mm. Jedná se o kanalizaci jednotnou, splašky jsou sváděny společně s dešťovou vodou do veřejné kanalizace.

XI. VODOVOD

Vlastní vodovodní přípojka na veřejnou vodovodní síť.
Rozvody uvnitř i vně objektu jsou provedeny z trubek PVC.

ZEMNÍ PRÁCE PRO PŘÍPOJKY INŽENÝRSKÝCH SÍTÍ

Strojem budou vytvořeny brázdy pro vedení plynu, vodovodu, elektřiny a tepla. Trubky budou pokládány na pískový podsyp tloušťky 300 mm. Elektrické dráty chráněny plastovými trubkami, ostatní vedení obaleno ochrannou pěnou z PVC tloušťky 15 mm.

IZOLACE STAVBY PROTI ZEMNÍ VLHKOSTI A RADONU

Prováděné testy prokázaly nízkou přítomnost radonu v půdě. Vodorovná i svislá izolace bude provedena izolací ALFOBIT – s hliníkovou nosnou vložkou. Svislá izolace je chráněna geotextilií, plastovou izolací DÖRKEN. Podzemní voda je níže než základy, nehrozí tedy jejich poškození. O odvod přebytečné vody se postarají plastové drenážní trubky položené v blízkosti základových pásů.

Závěrečné hodnocení studenta

V přípravě na tuto práci jsem se dozvěděl mnoho nového o problémech, starostech a ústrcích lidí, kteří jsou nějakým způsobem „jiní“, než většinová část populace. Také jsem zjistil jakým způsobem postupovat při navrhování staveb, jejich prostorového řešení, interiéru a exteriéru. Problémy se vyskytly, ale vždy se je, byť i za pomoci pí.uč. Svobodové, podařilo vyřešit.

Seznam použité literatury

- p. Ivo Krajíček – osobní zkušenosti a názory
- Daniela Filipiová - Projektujeme bez bariér
- Jan Skopec – Bezbariérové řešení staveb
- Stavební zákon
- Vyhláška č. 132/1998 Sb.
- Vyhláška č. 137/1998 Sb.
- Vyhláška č. 369/2001 Sb.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

KM Beta – Rodinné domy

Monika Martinková, Ondřej Horký, Michal Trnka,
Zdeňka Říhová, Marcela Nováková, Marek Šťastný

Střední škola stavební Jihlava
Žižkova 20, Jihlava

Rodinný dům – NOT FOR EVERYBODY (Michal Trnka) - Tento dům je myšlen jako dvoupodlažní, částečně zapuštěný do terénu. V prvním podlaží se nachází hlavní vchod, garáž, technická místnost, šatna a WC. Do druhého podlaží ve tvaru L se dostaneme dvouramenným schodištěm. Zde jsou hlavní místnosti budovy, jako obývací pokoj, kuchyň, ložnice nebo dětský pokoj. Je to jednoduchý dům pro čtyř až pěti člennou rodinu (místo fitcentra možnost dalšího pokoje).

Rodinný dům KM Beta (Marek Šťastný) - Tento dům je myšlený jako dvou podlažní. Druhé podlaží je tvořeno podkrovními místnostmi. V prvním podlaží je kuchyň, technická místnost, WC, jídelna, obývací pokoj, potravinová komora a předsíň. V druhém podlaží se nachází WC, dva dětské pokoje, ložnice a koupelna. K domu náleží dvě parkovací místa. Dům je vhodný pro čtyř člennou rodinu.

Nízkoenergetický rodinný dům (Monika Martinková, Ondřej Horký) – Tento dům je myšlený jako dvoupodlažní, z cihelného zdiva KMB SENDWIX. Je postaven na rovinném povrchu. Dominantním prvkem v tomto domku je prosvětlené schodiště rohovým oknem.

Nízkoenergetický rodinný dům (Zdeňka Říhová, Marcela Nováková) Jedná se o dvoupodlažní nízkoenergetickou vilu umístěnou ve svahu se soustavou pultových střech. Do objektu se vchází v druhém nadzemním podlaží. V prvním nadzemním podlaží se nachází dva dětské pokoje, pokoj pro hosty, ložnice, relaxační kout, šatna, WC a koupelna. V druhém nadzemním podlaží se nachází obývací pokoj s kuchyňským koutem, spíž, úklidová místnost + WC, chodba, garáž, zádveří a prostorná terasa. Dům je určen pro početnou rodinu.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

NOVOSTAVBA RODINNÉHO DOMU

ZDEŇKA ŘÍHOVÁ

Střední škola stavební Jihlava
Žižkova 20
586 01 Jihlava

TECHNICKÁ ZPRÁVA

SEZNAM PŘÍLOH:

- **Technická zpráva**
- **Tabulky PSV**
- **Tabulky HSV**
- **Výkresová část**
- **3D vizualizace**

SEZNAM VÝKRESŮ

Prováděcí projekt 1:50

1) SITUACE
2) VÝKOPY
3) ZÁKLADY
4) PŮDORYS 1. NP
5) PŮDORYS 2. NP
6) STROPY NAD 1. NP
7) STROPY A VÝKRES NADPRAŽOVÝCH PRVKŮ 2. NP
8) PLOCHÁ STŘECHA
9) KROV
10) PODÉLNÝ ŘEZ A-A
11) VÝKRES DETAILŮ PODLAH 1:10
12) DETAILY 1:10

TECHNICKÁ ZPRÁVA

(Stavební část)

SPECIFIKACE STAVBY

lokalita – blíže neurčena
zastavěná plocha – 267,52 m^2
charakteristika stavby- novostavba nízkoenergetického rodinného domu

1.) Základní údaje o stavbě

- **stručný popis stavby**

Jedná se o dvoupodlažní nízkoenergetickou vilu umístěnou ve svahu se soustavou pultových střech. Do objektu se vchází v druhém nadzemním podlaží. V prvním nadzemním podlaží se nachází dva dětské pokoje, pokoj pro hosty, ložnice, relaxační kout, šatna, WC a koupelna. V druhém nadzemním podlaží se nachází obývací pokoj s kuchyňským koutem, spíž, úklidová místnost + WC, chodba, garáž, zádveří a prostorná terasa.

- **využití jednotlivých podlaží**

1. NP- V prvním nadzemním podlaží je tzv. klidová zóna. Jsou zde dva dětské pokoje, pokoj pro hosty, ložnice, WC a koupelna. Jelikož je objekt navržen ve svahu, prostor pod schodištěm je navržen jako odpočinková zóna. V dětských pokojích, pokoji pro hosty a v ložnici je navržena korková podlaha, na chodbě bude keramická dlažba, která bude pod schody překrytá kobercem, v koupelně a na WC je keramická dlažba a stěny jsou obloženy do výšky 1800 mm. Stěny v tomto podlaží se budou malovat PRIMALEXEM v odstínu, který vybere investor.

2. NP- V druhém nadzemním podlaží je obývací pokoj s kuchyňským koutem, odkud je možnost vstupu do spíže. V obývacím pokoji je dřevěná – palubková podlaha, na chodbě, v zádveří, kuchyňském koutě a spíži je navržena keramická dlažba se soklem do výšky 80 mm, v úklidové místnosti + WC je také keramická dlažba a stěny jsou obloženy do výšky 1600 mm, v garáži je betonová mazanina a na terase je kamenná dlažba. Stěny v tomto podlaží se budou malovat PRIMALEXEM v odstínu, který vybere investor.

2.) Charakteristika území, dotčená ochranná pásma, chráněná území, požadavky na demolice, kácení stromů

Stavební pozemek se nachází ve svažitém terénu a je vymezen stávající komunikací. Zeminy na staveništi jsou tvořeny jílovitou hlínou s úlomky břidlic, která přechází do podložních hornin tvořených žulou. Zeminy na staveništi lze zařadit podle ČSN 73 1001 do třídy F2 - jíl štěrkovitý. Vzhledem k úpravám terénu v průběhu předchozí stavební činnosti je nutné volit pro základovou spáru zeminy v původním terénu, v min. hloubce (z hlediska nezámrznosti) 0,8 m pod upraveným terénem.

Na staveništi byl proveden průzkum z hlediska rizika výskytu radonu. A bylo prokázáno, že v tomto místě není výskyt radonu v takové míře, aby ohrožoval lidské zdraví.

3.) Vliv stavby na životní prostředí, případná omezení nebo odstranění těchto vlivů

Stavba rodinného domu neovlivní ani nenaruší životní prostředí. Odpadní vody budou svedeny do veřejné kanalizační sítě. Jejich likvidaci bude zajišťovat investor v souladu s podmínkami životního prostředí. Tuhý domovní odpad je svážen technickými službami na řízenou skládku.
Skladba zemin na staveništi stavbu rodinného domu rovněž neovlivní.

4.) Odolnost a zabezpečení z hlediska požární ochrany a potřeb CO

Stavební objekt bude jako samostatný požární úsek. Řešení rodinného domu je v souladu s požadavky ČSN 73 0833 a ČSN 73 0802. Požární odolnost stavebních konstrukcí je vyšší než požadují uvedené normy.
Požadavky z hlediska CO na tuto stavbu nejsou.

5.) Inženýrské sítě

1) Kanalizace

Odpadní vody z rodinného domu budou odváděny do veřejné kanalizační sítě.

2) Voda

Pro zásobování objektu vodou je navržena vodovodní přípojka, která bude napojena na stávající nápojný bod na hranici pozemku. Vodovodní přípojka je navržena z rPe φ 32 mm.

3) Elektrická energie

Přípojka elektro rodinného domu je ukončena v přípojkové skříni. Nad touto skříní je zároveň umístěna elektroměrová skříň s hlavním domovním jističem. Z této přípojkové skříně bude provedena přípojka zemním kabelem do objektu k podružnému rozvaděči, ze kterého budou napájeny jednotlivé obvody elektrorozvodů.

4) Vytápění

Objekt může být vytápěn solární energií nebo jako možnost příhřevu může být teplovodní plynové vytápění.

POZNÁMKA:

- **PODLAHA 2. NP ± 0,000**
- **VÝŠKOVÉ KÓTY JSOU VZTAŽENY K ± 0,000**
- **ZÁKLADY A PODKLADNÍ BETONY Z BETONU B 30**
- **IZOLACE PROTI ZEMNÍ VLHKOSTI - FOALBIT TL. 1,0 mm, CHRÁNĚNÁ TEPELNOU IZOLACÍ A PŘIZDÍVKOU**

PŘED ZAHÁJENÍM ZEMNÍCH PRACÍ JE INVESTOR POVINEN VYTÝČIT VEŠKERÁ PODZEMNÍ VEDENÍ A ZAŘÍZENÍ JEJICH SPRÁVCI A VIDITELNĚ OZNAČIT JEJICH POLOHU!

TECHNICKÉ ŘEŠENÍ STAVBY

Stavba je navržena ze zdícího systému KMB SENDWIX P na zdící maltu SENDWIX 920 SX. Vnitřní nosné zdivo je z tvárnic SENDWIX 5DF-P. Příčky jsou navrženy z tvárnic pro nenosné zdivo SENDWIX 4DF-LD. Strop je navržen z keramických nosíků POT a MIAKO vložek a poté je celý strop zabetonován betonem B 30.

Střecha je řešena jako jednoduchý dřevěný krov s dřevěným podhledem.

■ Zemní práce

Ornice se vytěží asi v tloušťce 20 cm a uskladní na severovýchod od stavby pro pozdější dorovnání. Zemina je ve třídě těžitelnosti 1, 2 a 3. Dále se budeme řídit výkresem výkopů.

■ Základové konstrukce

Obvodové stěny jsou založeny na základových pasech o výšce 1 m z betonu B 30. Vnitřní nosné stěny jsou také založeny na základových pasech, ty jsou ale vysoké 0,5 m. Podkladní beton je také z betonu B 30 o tl. 0,1 m. Na podkladním betonu je položena hydroizolace FOALBIT, která je vytažená do výšky 0,2 m, a na ní leží tepelná izolace viz. výkres základů. Základová spára bude v hloubce 3,2 což je nezámrzná hloubka. Hladina spodní vody způsob zakládání stavby nijak neovlivní.

■ Svislé konstrukce

Stavba je navržena ze zdícího systému KMB SENDWIX P tl. 400 mm na zdící maltu SENDWIX 920 SX. Vnitřní nosné zdivo je z tvárnic SENDWIX 5DF-P tl. 300 mm. Příčky jsou navrženy z tvárnic pro nenosné zdivo SENDWIX 4DF-LD tl. 115 mm.

■ Vodorovné konstrukce

V objektu se nachází tři vodorovné konstrukce. Je to beton v 1. NP, strop nad 1.NP a strop nad 2.NP. V 1.NP je to podkladní beton s hydroizolací, krycí beton a konstrukce podlahy (viz.výkres podlah). Strop nad 1.NP i 2.NP je navržen z keramických nosíků POT a MIAKO vložek a poté je celý strop zabetonován betonem B 30.

■ Střecha

Na objekt je navržena soustava pultových střech se sklonem 12 a 15 stupňů. Ty jsou tvořeny jednoduchým dřevěným krovem s tepelnou izolací. Krytina je navržena z betonových tašek KM BETA.

■ Úpravy povrchů

a) Podlahy – v prvním nadzemním podlaží je navržena podlaha podle užití místnosti, je tam keramická dlažba+koberec a korek. V druhém nadzemním podlaží je to dřevěná podlaha, keramická dlažba, kamenná dlažba a betonová mazanina. (viz.výkres podlah)

b) Úpravy povrchů vnitřních - vnitřní povrchy jsou tvořené jednovrstvou omítkou, keramickým a dřevěným obkladem. Barevné odstíny budou upřesněny na stavbě s investorem.

c) Úpravy povrchů vnějších – tepelně izolační omítka SUPERTHERM TO, barva podle investora. Celý objekt je lemován do výšky 300 mm lomovým kamenem.

PRÁCE PSV

■ Izolace tepelné

Tepelná izolace pro strop je navržena polystyren XPS TL. 150 mm.

■ Izolace proti zemní vlhkosti a bariéra proti zemnímu plynu

Izolace proti zemní vlhkosti je osazena na podkladním betonu a je tvořena hydroizolační folií FOALBIT v tl. 0,8 mm. Fólie je vytažena na obvodovou stěnu 300 mm nad terén. Tato fólie je zároveň kvalitní bariérou proti průniku zemního vzduchu s obsahem radonu.

Tloušťku izolace z hlediska průniku radonu není nutné posuzovat, neboť v průzkumu nebylo prokázáno takové množství radonu, aby ohrožovalo lidské zdraví. Z hlediska ČSN 73 0601 pak postačí izolace v dostatečném provedení jako hydroizolace.

Izolace stěn a podlahy v sociálních zařízeních je řešena jako izolace proti volně stékající vodě, která je tvořena stěrkou PCI – Lastogum doplněná o těsnící pásky, lepidla a spárovací hmoty.

■ **Výplně otvorů**

Okna a dveře jsou navrženy v dřevěném provedení. Balkónové dveře jsou navrženy dřevěné prosklené.

■ **Práce truhlářské**

Do truhlářských prací je možno zařadit výplně otvorů (okna a dveře). Dále vnitřní schodišťové zábradlí.

■ **Klempířské výrobky**

Veškeré klempířské výroby, tj. střecha, okapy, svody, parapety oken atd. budou vyrobeny z měděného plechu tl. 0,6 mm podle zásad ČSN 73 3610.

■ **Elektroinstalace**

Elektrorozvody jsou řešeny měděnými vodiči uloženými ve stěnách (za sádrokartonovými deskami).

■ **Vytápění**

Objekt bude vytápěn pomocí solárních panelů umístěných na střeše. Možnost příhřevu bude teplovodní plynové vytápění.

■ **Kanalizace**

Vnitřní rozvody kanalizace budou provedeny z PVC spojovaných lepením. Jednotlivá připojovací potrubí jsou rovněž z PVC a jsou napojena na stoupačku nebo přímo na ležatou kanalizaci pod podkladním betonem v nasypané zemině.
Ležatá kanalizace pod podkladním betonem v nasypané zemině včetně přípojky musí být položena před prováděním podkladních betonů.

■ **Voda**

Objekt je napojen samostatnou přípojkou vody z rPe φ 32 mm. Hlavní uzávěr vody a vodoměr jsou umístěny v 1. S v kotelně. Ohřev teplé užitkové vody bude realizován v plynovém ohřívači vody, který je přímou součástí plynového kotle.

BEZPEČNOST PRÁCE

Stavba bude prováděna při dodržování všech předpisů bezpečnosti práce a předpisů hygienických. Nejdůležitějším předpisem pro zajištění bezpečného provádění stavebních prací je vyhláška ČÚBP a ČBÚ a. 324/1990 sb. o bezpečnosti práce a technických zařízeních při stavebních pracích.

Před zahájením stavebních resp. výkopových prací je bezpodmínečně nutné vytýčení všech podzemních inženýrských sítí investorem a viditelné zajištění jejich poloh.

Zemní práce budou prováděny v nepažených výkopech rýh, což umožňuje soudržná zemina na staveništi. Při místních změnách zemin v podloží je nutné určit další postup prací. Při výkopových pracích je nutno dodržovat ustanovení § 20 vyhláška 324/1990!

Stavební práce budou prováděny podle projektové dokumentace a v souladu s technologickými postupy dodavatele stavby. Veškeré práce budou provedeny ve shodě s platnými ČSN.

Nepředvídané práce a veškeré změny oproti projektové dokumentaci budou řešeny v rámci pravomoci technického dozoru investora a autorského dozoru projektanta stavby.

TABULKY PSV :

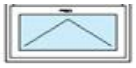
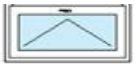
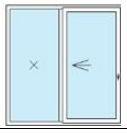
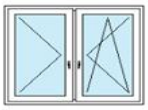
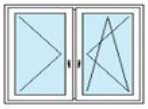
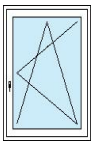
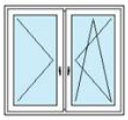
DVEŘE 1. NP

OZN. VE VÝKRESU	SCHEMATICKÉ ZOBRAZENÍ (POPIS)	ROZMĚRY	POČET (KS)	ZASKLENÍ	BAREVNÝ ODSTÍN (NÁTĚR)	POZNÁMKA
L1		800/2000	2	Dle obrázku	Přírodní	
P2		800/2000	2	Dle obrázku	Přírodní	SAPELI
L3		600/2000	1	-	Přírodní	SAPELI
P4		600/2000	1	-	Přírodní	SAPELI
L7		900/2350	2	-	Přírodní	SAPELI
P10		900/2350	1	Dle obrázku	Přírodní	SAPELI

DVEŘE 2. NP

OZN. VE VÝKRESU	SCHEMATICKÉ ZOBRAZENÍ (POPIS)	ROZMĚRY	POČET (KS)	ZASKLENÍ	BAREVNÝ ODSTÍN (NÁTĚR)	POZNÁMKA
1		2400/2200	1	-	Přírodní	
L5		600/1970	2	-	Přírodní	SAPELI
P6		900/1970	1	Dle obrázku	Přírodní	SAPELI
L7		800/1970	1	Dle obrázku	Přírodní	SAPELI
P8		800/1970	1	Dle obrázku	Přírodní	SAPELI

OKNA

OZN. VE VÝKRESU	SCHEMATICKÉ ZOBRAZENÍ (POPIS)	ROZMĚRY	POČET (KS)	ZASKLENÍ	BAREVNÝ ODSTÍN (NÁTĚR)	POZNÁMKA
1		750/800	2	Dle obrázku	Přírodní	VEKRA
2		750/1500	2	Dle obrázku	Přírodní	VEKRA
3		1500/2350	1	Dle obrázku	Přírodní	VEKRA
4		2000/2350	1	Dle obrázku	Přírodní	VEKRA
5		1500/1500	2	Dle obrázku	Přírodní	VEKRA
6		1500/500	2	Dle obrázku	Přírodní	VEKRA
7		1000/1500	4	Dle obrázku	Přírodní	VEKRA
8		2000/1500	2	Dle obrázku	Přírodní	VEKRA

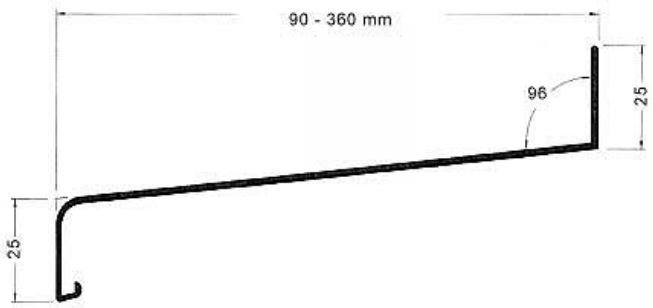
TRUHLÁŘSKÉ PRVKY

OZN.	POPIS	VÝROBCE	MATERIÁL
TR 11	VNITŘNÍ PERAPET	R.K. TRUHLÁŘSTVÍ	SMRK
TR 12	VNITŘNÍ PERAPET	R.K. TRUHLÁŘSTVÍ	SMRK
TR 13	VNITŘNÍ PERAPET	R.K. TRUHLÁŘSTVÍ	SMRK
TR 14	VNITŘNÍ PERAPET	R.K. TRUHLÁŘSTVÍ	SMRK
TR 15	VNITŘNÍ PERAPET	R.K. TRUHLÁŘSTVÍ	SMRK
TR 16	SCHODIŠŤOVÉ ZÁBRADLÍ	R.K. TRUHLÁŘSTVÍ	SMRK

KLEMPÍŘSKÉ PRVKY

OZN.	POPIS	VÝROBCE	MATERIÁL
KL 11	VENKOVNÍ PERAPET	EUROKAN S.R.O	MĚĎ
KL 12	VENKOVNÍ PERAPET	EUROKAN S.R.O	MĚĎ
KL 13	VENKOVNÍ PERAPET	EUROKAN S.R.O	MĚĎ
KL 14	VENKOVNÍ PERAPET	EUROKAN S.R.O	MĚĎ
KL 15	VENKOVNÍ PERAPET	EUROKAN S.R.O	MĚĎ
KL 16	OPLECHOVÁNÍ ATIKY	EUROKAN S.R.O	MĚĎ

DETAIL MĚĎĚNÉHO PARAPETU



TABULKY HSV :

STROP NAD 1.NP

OZN.	POPIS	ROZMĚRY	KUSY	MATERIÁL
N1	KERAMICKÝ NOSNÍK POT	160X175X4500	30	KERAMOBETON
N2	KERAMICKÝ NOSNÍK POT	160X175X5500	13	KERAMOBETON
M1	STROPNÍ VLOŽKA MIAKO	4050X140X225	21	
M2	STROPNÍ VLOŽKA MIAKO	5600X140X225	648	
M3	STRPONÍ VLOŽKA MIAKO	2400X140X225	17	

STROP NAD 2.NP

OZN.	POPIS	ROZMĚRY	KUSY	MATERIÁL
N2	KERAMICKÝ NOSNÍK POT	160X175X4500	9	KERAMOBETON
M2	STROPNÍ VLOŽKA MIAKO	525X190X250	72	
I1	VÁLCOVANÝ PROFIL I 240	106X240X5500	1	OCEL

PŘEKLADY 1.NP

OZN.	POPIS	ROZMĚRY	KUSY	MATERIÁL
P1	8DF 125	240X240X1250	3	VÁPENOPÍSEK
P2	8DF 225	240X240X2250	4	VÁPENOPÍSEK
P3	2DF 125	115X240X1250	6	VÁPENOPÍSEK
P4	2DF 100	115X240X1000	2	VÁPENOPÍSEK

PŘEKLADY 2.NP

OZN.	POPIS	ROZMĚRY	KUSY	MATERIÁL
P1	SENDWIX 8DF 100	240X250X1000	4	VÁPENOPÍSEK
P2	SENDWIX 8DF 175	240X250X1750	6	VÁPENOPÍSEK
P3	SENDWIX 8DF 300	240X250X3000	1	VÁPENOPÍSEK
P4	SENDWIX 8DF 225	240X250X2250	1	VÁPENOPÍSEK
P5	SENDWIX 2DF 125	115X250X1250	2	VÁPENOPÍSEK
P6	SENDWIX 2DF 100	115X250X1000	2	VÁPENOPÍSEK

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

RODINNÝ DŮM KM BETA

Marek Šťastný

STŘEDNÍ ŠKOLA STAVEBNÍ JIHLAVA



Obsah

TEXTOVÁ ČÁST :

- Technická zpráva

VÝKRESOVÁ ČÁST:

A. Architektonicko – stavební část v měřítku 1:100

1. půdorysy

B. prováděcí projekt v měřítku 1: 50

1. situace měřítko 1 : 100
2. výkopy
3. základy
4. půdorys 1.NP
5. půdorys 2.NP
6. technické vybavení 1.NP
7. technické vybavení 2.NP
8. strop
9. střecha – sedlová
10. svislý řez A- A´, B- B´
11. pohledy

TECHNICKÁ ZPRÁVA

RODINNÝ DŮM

Účel stavby: Novostavba leží ve výšce 530 metrů nad mořem – Baltský systém. Rodinný dům umístěn na okraji města.
Zastavěná plocha: 95,34 m^2

I. ARCHITEKTONICKÉ ŘEŠENÍ

Rodinný dům, je postaven na okraji města, snadno dostupný, ať už autem nebo mhd. V okolí najdeme několik dalších rodinných domů. Před parcelou probíhá hlavní silnice. Budova je postavena v nové lokalitě s moderními domy.

II. ORIENTACE KE SVĚTOVÝM STRANÁM A DENNÍ OSVĚTLENÍ

Místnosti, které budou nejčastěji využívány jsou dostatečně osvětleny a osluněny na JV jsou umístěny – jídelna, kuchyň,dětský pokoj a ložnice na JZ je umístěn obývací pokoj, dětský pokoj.Oproti tomu méně používané prostory jako –WC ,koupelna technická místnost,potravinová komora a předsíň jsou situovány na SV,SZ.

III. FUNKČNÍ ŘEŠENÍ

- V 1.NP je umístěn hlavní vchod (na SZ budovy), tím se dostaneme do předsíně, ze které můžeme jít na WC nebo do jídelny a jejich přilehlých místností tzn. Obývací pokoj, kuchyň,potravinová komora a schodiště. Schodiště probíhá celou budovou.
- V 2.NP jsou umístěny dětské pokoje, ložnice, koupelna a samostatné WC.

IV. STAVEBNĚ TECHNICKÉ ŘEŠENÍ

ZEMNÍ PRÁCE

Terén kolem objektu bude snížen pouze o 50 mm.Výkopy pro základové pásy budou dosahovat hloubky –1,350 m od 1.NP = ± 0,000, pro schody před domem do hloubky -1,200m a pro schodiště v domě do hloubky -1,200m stěny vyhloubených jam budou svislé, půdorys je ve tvaru obdélníku, výkopy se jim musí přizpůsobit. Zeminu budou těžit rýpadla, nákladní auta ji odvezou.

ZÁKLADY

Pod obvodovým zdivem budou probíhat základové pásy s rovnými čely, pod základy štěrkový podsyp. Základová spára sahá do hloubky –1,250 m od 1.NP = ± 0,000. Velikosti a natočení základu závisí na šířce stěn a úhlu, který spolu obvodové stěny svírají (90° a 135°). Pro obvodové zdivo je šířka základového pásu 550 mm.

SVISLÉ KONSTRUKCE

Celý objekt je tvořen ze zdiva KM BETA. Obvodové zdivo SENDWIX P, příčkové zdivo SENDWIX 2DF-D VŠE NA MALTU 920 SX.

VODOROVNÉ KONSTRUKCE

Všechny stropy jsou zhotoveny z keramického stropu MIAKO. V objektu najdeme keramické překlady s tepelnou izolací. Obvodovým zdivem probíhá věnec – výška 300 mm, ocelový pruty 10216 ø8, izolován tepelnou izolací PPS, zakončeno věncovkou POROTHERM 23,5.

SCHODIŠTĚ

Všechny interiérové schody mají železobetonovou schodnici. Na jednotlivých stupních je jako nášlapná vrstva dřevěný obklad.

V. STŘEŠNÍ KONSTRUKCE

Dům je zakončen sedlovou a střechou, která je pokryta betonovými taškami KM Beta – barva višňová.

VI. PODLAHY, OBKLADY, DLAŽBY, OMÍTKY (ZATEPLENÍ)

- Keramické dlaždice značky RAKO - jsou převážně v koupelně, ve spižírně, na WC a technické místnosti. Lepené na podlahu, flexibilní cementovou lepící maltou značky KIESEL. Proti vlhkosti stěrkové izolace.
- Keramický obklad vhodně zvolený k barvě podlahy v koupelně a na WC. V kuchyni, na WC i v koupelně je obklad umístěn do výšky 1,5m, opět je nalepován na stěny.
- Na chodbě, v kuchyni, jídelně a obývacím pokoji je plovoucí podlaha.
- V ostatních místnostech jsou koberce s různými odstíny červené barvy.
- Vnitřní stěny budou dvakrát pačokované. Na malbu jsou použity různé odstíny barvy značky PRIMALEX.

VII. OKNA, DVEŘE, ZÁRUBNĚ

- Použitá okna značky OKNA (znackovaokna.cz) mají izolační troj sklo U = 0,5 W/m^{2}K. Zvuková izolace skel asi Rw32dB. Na objektu najdeme tyto typy oken : otevíravá .
- Dveře dřevěné, částečně zasklené od různých firem.
- Využití kovových zárubní – TARGUS.

VIII. VYTÁPĚNÍ

Objekt je vytápěn elektrickým proudem a tepelným podlahovým vytápěním. El.kotel se nachází v technické místnosti.Značka kotle Electra8.

IX. TECHNICKÉ VYBAVENÍ, ZDRAVOTECHNIKA, KONSTRUKCE A PRÁCE PSV

- Zdravotní technika: umývadla, záchodové mísy, rohová vana. Vše v bílé barvě. Nerezové kohoutky SANELA.
- V podlaze je umístěno podlahové vytápění od firmy STORC (storc.cz) Elektroinstalace navrženy odborníky ze společnosti ELEKTROS spol. s r.o. Rozvodné

  sítě jsou zabudovány ve stěnách objektu. Přípojka na veřejnou síť nízkého napětí.

X. KANALIZACE

Je navržena a zhotovena odborníky z firmy VODOINSTAL. Plastové potrubí svádí vodu z celého objektu
+ drenážní trubky - Durvinil vedeme pod objektem v hloubce asi 1m do veřejné betonové kanalizace. Kanalizační trubky Gienger ø 400 mm. Jedná se o kanalizaci jednotnou, splašky jsou sváděny společně s dešťovou vodou do veřejné kanalizace.

XI. VODOVOD

Vlastní vodovodní přípojka na veřejnou vodovodní síť. Rozvody uvnitř i vně objektu jsou provedeny z trubek PVC.

XII. ZEMNÍ PRÁCE PRO PŘÍPOJKY INŽENÝRSKÝCH SÍTÍ

Strojem budou vytvořeny brázdy pro vedení plynu, vodovodu, elektřiny. Trubky budou pokládány na pískový podsyp tloušťky 300 mm. Elektrické dráty chráněny plastovými trubkami.

XIII. IZOLACE STAVBY PROTI ZEMNÍ VLHKOSTI A RADONU

Prováděné testy prokázali nízkou přítomnost radonu v půdě. Za vodorovnou i svislou izolaci jsme zvolili izolaci ALFOBIT – s hliníkovou nosnou vložkou. Svislá izolace je chráněna geotextilií, plastovou izolací DÖRKEN. Podzemní voda je níže než základy, nehrozí tedy jejich poškození. O odvod přebytečné vody se postarají plastové drenážní trubky položené v blízkosti základových pasů .

PLASTOVÉ VÝROBKY

OZNAČENÍ	ZOBRAZENÍ	POPIS (OKNA MAJÍ ATIPICKÝ TVAR)	ROZMĚRY V MM (Š V)	POČET KS			KOVÁNÍ;	ZASKLENÍ	BARVA
				PODLAŽÍ		CELKEM			
				1.NP	2.NP				
1		OKNO PLASTOVÉ, OTVÍRÁVÉ, BEZ VENTILACE	1300 x 1375	1	0	1	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
2		OKNO PLASTOVÉ, OTVÍRÁVÉ, BEZ VENTILACE	2390 x 2250	2	0	2	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
3		OKNO PLASTOVÉ, OTVÍRÁVÉ, S VENTILACÍ	1000 x 1375	2	0	2	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
4		OKNO PLASTOVÉ, OTVÍRÁVÉ, S VENTILACÍ	800 x 1375	1	0	1	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
5		OKNO PLASTOVÉ, OTVÍRÁVÉ, S VENTILACÍ	1375 x 800	3	0	3	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
6		OKNO PLASTOVÉ, OTVÍRÁVÉ, S VENTILACÍ	1200 x 1500	1	0	1	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
7	ZKOSENÉ	OKNO PLASTOVÉ, OTVÍRÁVÉ, S VENTILACÍ	1000 x 1100	0	1	1	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
8	ZKOSENÉ	OKNO PLASTOVÉ, OTVÍRÁVÉ, S VENTILACÍ	1400 x 1100	0	1	4	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
9		STŘEŠNÍ OKNO PLASTOVÉ, OTVÍRÁVÉ, S VENTILACÍ	600 x 1576	0	2	5	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
10		STŘEŠNÍ OKNO PLASTOVÉ, JEDNOKŘÍDLÉ S VENTILACÍ	900 x 1222	0	4	4	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
11		STŘEŠNÍ OKNO PLASTOVÉ, JEDNOKŘÍDLÉ S VENTILACÍ	500 x 1050	0	2	2	CELOOBVODOVÉ KOVÁNÍ SIGENIE - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	RAL 1002
CELKEM VÝROBKŮ						**26**			

DŘEVĚNÉ VÝROBKY

OZNAČENÍ	POPIS	ROZMĚRY V MM (Š V)	ZÁRUBEŇ	POČET KS			BARVA
				PODLAŽÍ		CELKEM	
				1.NP	2.NP		
1	DVEŘNÍ KŘÍDLO JEDNODUCHÉ OTOČNÉ PRAVÉ	900 x 1970	KOVOVÁ	1	0	1	ZÁKLADOVÁ BARVA RAL 3002
2	DVEŘNÍ KŘÍDLO JEDNODUCHÉ OTOČNÉ PRAVÉ	600 x 1970	KOVOVÁ	1	0	1	ZÁKLADOVÁ BARVA RAL 3002
3	DVEŘNÍ KŘÍDLO JEDNODUCHÉ OTOČNÉ LEVÉ	800 x 1970	KOVOVÁ	1	3	4	ZÁKLADOVÁ BARVA RAL 3002
4	DVEŘNÍ KŘÍDLO JEDNODUCHÉ OTOČNÉ PRAVÉ	800 x 1970	KOVOVÁ	1	1	2	ZÁKLADOVÁ BARVA RAL 3002
5	DVEŘNÍ KŘÍDLO DVOJTE OTOČNÉ	1800 x 1970	KOVOVÁ	1	0	1	ZÁKLADOVÁ BARVA RAL 3002
6	DVEŘNÍ KŘÍDLO JEDNODUCHÉ OTOČNÉ PRAVÉ	800 x 2300	KOVOVÁ	1	0	1	RAL 1002
7	DVEŘNÍ KŘÍDLO JEDNODUCHÉ OTOČNÉ PRAVÉ	750 x 1970	KOVOVÁ	0	1	1	ZÁKLADOVÁ BARVA RAL 3002
CELKEM VÝROBKŮ						**11**	

VÝPIS PREFABRIKÁTŮ A OCELOVÝCH NOSNÍKŮ

PROFILOVANÉ OCELOVÉ I NOSNÍKY

POPIS PRVKU	POČET KS CELKEM	ROZMĚRY PRVKŮ (D Š V)
I č. 180	4	4087 x 90 x 180
I č. 180	3	4487 x 90 x 180
I č. 180	2	6700 x 90x 180
I č. 180	2	4750 x 90x 180
I č.200	10	5256 x 90x 200
CELKEM VÝROBKŮ		**9**

ŽELEZOBETONOVÉ PŘEKLADY

OZNAČENÍ	POPIS PRVKU	POČET KS CELKEM	ROZMĚRY PRVKŮ (D Š V)
b	PŘEKLAD SENDWIX 2DF	4	1250 x 115 x 240
c	PŘEKLAD SENDWIX 2DF	8	1000 x 115 x 240
d	PŘEKLAD SENDWIX 2DF	4	1500 x 115 x 240
e	PŘEKLAD SENDWIX 2DF	5	2750 x 115 x 240
CELKEM PRVKŮ			**21**

Vizualizace

Pohled na vstupní dveře a na parkoviště



Pohled na WC, potravinovou komoru,kuchyň a technickou místnost



Pohled na terasu a bazén.



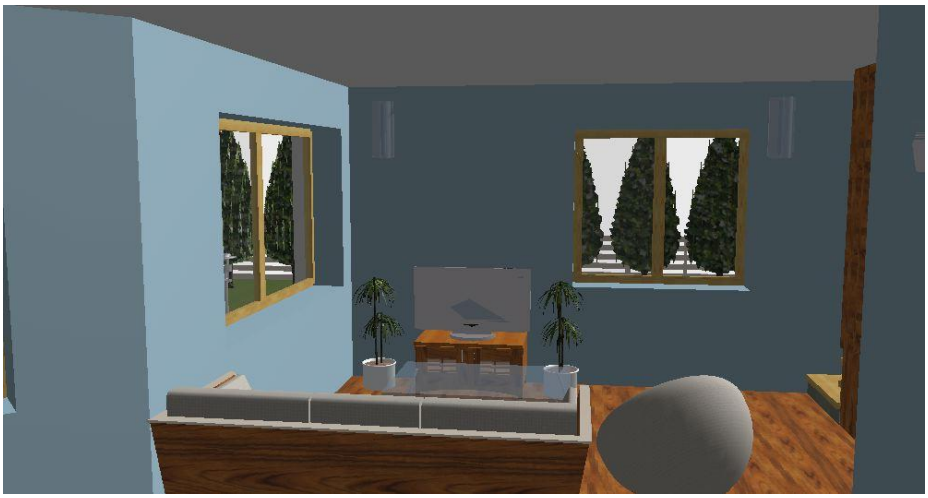
Pohled na kuchyň.



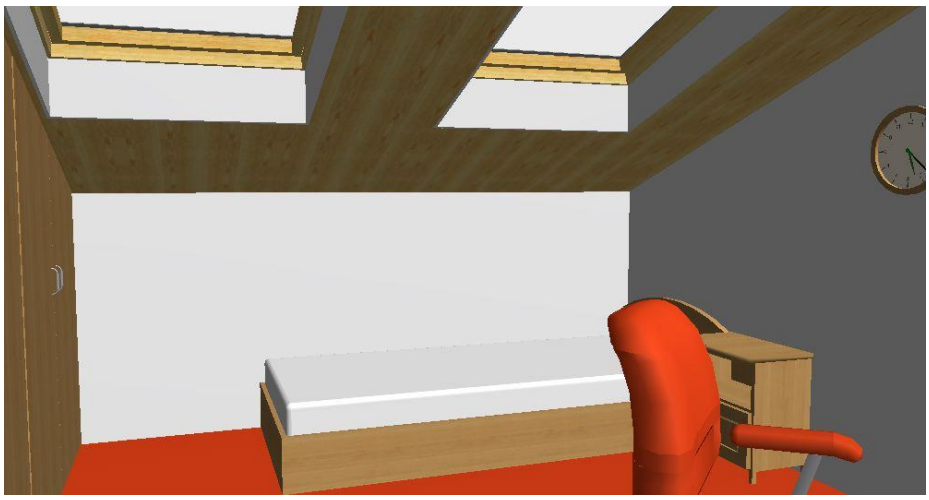
Pohled na knihovnu.



Pohled na obývací pokoj.



Pohled na dětský pokoj.



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

RODINNÝ DŮM – NOT FOR EVERBODY

MICHAL TRNKA

STŘEDNÍ ŠKOLA STAVEBNÍ A UČILIŠTĚ V JIHLAVĚ



Datum : 8. 4. 2008

OBSAH

TEXTOVÁ ČÁST :

- Technická zpráva
- Tabulky PSV
- Výpis prefabrikovaných prvků
- Skladby podlah a střešních kcí
- Tepelné posudky podlahy a stěn
- Výpočet základů
- Přílohy
- Vizualizace

VÝKRESOVÁ ČÁST:

A. architektonicko – stavební část 1:100
    1. půdorysy, řez, pohledy

B. prováděcí projekt v měřítku 1: 50
    1. situace měřítko 1 : 100
    2. výkopy
    3. základy
    4. půdorys 1.NP
    5. půdorys 2.NP
    6. technické vybavení 1.NP
    7. technické vybavení 2.NP
    8. strop nad 1.NP
    9. strop nad 2.NP
    10. střecha
    11. svislý řez A –A´
    12. svislý řez B – B´
    13. pohled SZ
    14. pohled JZ
    15. pohled JV
    16. pohled SV

TECHNICKÁ ZPRÁVA

RODINNÝ DŮM

NOT FOR EVERYBODY

Účel stavby: Dům se nachází ve výšce 530 metrů nad mořem – Baltský systém. Je umístěn na velké ploše, v blízkosti které se nenacházejí žádné stavby.

Zastavěná plocha: 192,38 m^2
Obestavěný prostor: 940,87 m^3

I. ARCHITEKTONICKÉ ŘEŠENÍ:

Rodinný domek je postaven na místě nepatrně vzdáleném od města. Je snadno dostupný autem. V nedaleké blízkosti objektu se nachází základní občanská vybavenost – škola, dětské hřiště a supermarket. Před parcelou probíhá silnice, která je připojena na místní hlavní komunikaci. Z důvodů zadání střešní krytiny KM Beta je u tohoto domu navržena pultová střecha.

II. ORIENTRACE KE SVĚTOVÝM STRANÁM A DENNÍ OSVĚTLENÍ

Místnosti jako je obývací pokoj a pracovna jsou směřovány na SZ. Dětský pokoj a ložnice rodičů jsou směřovány na V až SV.
Ostatní místnosti jsou rozmístěny mezi těmito pokoji.

III. FUNKČÍ ŘEŠENÍ

- 1.NP – Hlavním vchodem se dostaneme do zádveří, kde se nachází vchod do šatny, na samostatné WC a do garáže. Posuvnými dveřmi vcházíme do obývacího pokoje s kuchyňským koutem a jídelnou.
- 2.NP – Do tohoto podlaží se dostaneme dvouramenným schodištěm. Po levé straně se nachází pracovna, po pravé obývací pokoj s kuch. koutem. Pokud projdeme obývacím pokojem dostáváme se do chodby, ze které se dostaneme do ostatních místností tohoto podlaží. Což jsou ložnice dětí a rodičů nebo posilovna.

IV. STAVEBNĚ TECHNICKÉ ŘEŠEÍ

- ZEMNÍ PRÁCE

Terén kolem objektu je snížen o 100mm. Výkopy budou dosahovat hloubky -1,405 od 1.NP = +0,000. Stěny výkopových jam budou svislé a přizpůsobeny tvaru budovy. Zemina bude těžena rypadly. Díky dobré kvalitě bude použita při výstavbě jiného objektu. Hlavní figury jímají 193,29 m^3, vedlejší 63,43 m^3.

-ZÁKLADY

Pod obvodovými zdmi budou probíhat základové pásy s rovnými čely šířky 700mm. Základy budou tvořeny betonem C16/20, pod kterými bude štěrkový hutněný podsyp. Základová spára sahá do hloubky -1,255 od 1.NP = +0,000. Základové pásy svírají úhel 90°.

- SVISLÉ KONSTRUKCE

Obvodové zdivo je ze sendvičového zdiva KM Beta SENDWIX L a příčky tloušťky 100mm – vápenopískové zdivo 4DF-D. Skloněná a zadní zeď u garážového stání je vyzděna z vápenopískových cihel SENDWIX 16DF-LD

- VODOROVNÉ KONSTRUKCE

Strop nad 1.NP a 2.NP je zhotoven z válcovaných I profilů výšky 200mm a keramickými vložkami HURDIS 2 + 2pa. Je tepelně odizolován a zmonolitněn betonem C16/20. Celková tloušťka obou stropů i s konstrukcí podlahy je 300mm. V objektu se nachází překlady SENDWIX 2DF různých délek viz výkresy stropů. Po celém obvodu objektu probíhá železobetonový věnec výšky 300mm schován za tepelnou izolací a lícovou cihlou NF.

- SCHODIŠTĚ

Všechny schody jsou tvořeny železobetonovou konstrukcí. Nášlapnou vrstvu tvoří dřevěný obklad. Podesta leží v polovině konstrukční výšky tj. v 1,450 m.

V. STŘEŠNÍ KONSTRUKCE

Objekt je zakončen pultovou nezateplenou střechou.
Jednotlivé vrstvy: Střešní krytina KMB Beta elegant višňová, prkenné pobití, pojistná folie (BETAFOL) a dřevěné krokve. Pro vstup na půdu bude v prostoru posilovny vytvořen vlez s dřevěným žebříkem.

VI. PODLAHY, OBKLADY, DLAŽBY A OMÍTKY

- V koupelně a na WC v 1.NP jsou podlahy a obklady různých druhů a rozměrů značky RAKO = výkresy vybavení, lepené na podlahu, flexibilní cementovou lepící maltou značky KIESEL. Proti vlhkosti stěrkové izolace.

- V místnostech, mimo dětský pokoj a posilovnu, kde je koberec, garáže a garážové stání kde je betonová mazanina a spíže kde se nachází marmoleum, jsou všude plovoucí podlahy různých druhů od firmy VaP nábytek.

- Vnitřní stěny budou dvakrát pačokované. Na malbu jsou použity různé odstíny barvy značky PRIMALEX.

VII. OKNA, DVEŘE A ZÁRUBNĚ

- V celé budově budou použita okna s izolačním dvojsklem značky VEKRA v barvě perleťová zlatá RAL 1036. Tato skla mají cca 50% úsporu tepelné energie unikající sklem. Na objektu najdeme otvíravá, vyklápěcí a neotevíratelná okna.
- Dveře jsou v celém objektu od firmy SAPELI – plné nebo částečně zasklené.
- U všech dveří jsou použity ocelové zárubně – TARGUS

VIII. VYTÁPĚNÍ

Celý dům bude vytápěn elektrickým proudem. Elektrický kotel se nachází v technické místnosti v 1.NP

IX. TECHNICKÉ VYBAVENÍ, ZDREVOTNÍ TECHNIKA, KONSTRUKCE A PRÁCE PSV

- Zdravotní technika: umyvadla, WC a vana, vše značky JIKA v bílé barvě.

- Klempířské práce: v objektu jsou navržena litinová otopná tělesa VIADRUS.

- Všechny elektroinstalace jsou navrženy odborníky. Rozvodové sítě budou vedeny ve stěnách objektu.

X. KANALIZACE

Je navržena a zhotovena odborník. Plastové potrubí svádí vodu z celého objektu
+ drenážní trubky - Durvinil vedeme pod objektem do veřejné betonové kanalizace.
Kanalizační trubky Gienger ø 400 mm. Jedná se o kanalizaci jednotnou, splašky jsou sváděny společně s dešťovou vodou do veřejné kanalizace.

XI. VODOVOD

Objekt má vlastní přípojku na vodovodní kanalizaci viz. výkres situace. Uvnitř i vně budovy budou vodovodní rozvody z trubek pvc.

XII. ZEMNÍ PRÁCE PRO PŘÍPOJKY INŽENÝRSKÝCH SÍTÍ

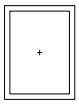
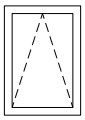
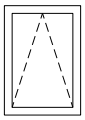
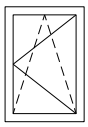
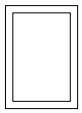
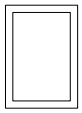
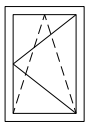
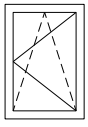
Pro vedení plynu, vodovodu a elektřiny budou strojem vytvořeny brázdy. Trubky budou pokládány na pískový podsyp tloušťky 300 mm a elektrické dráty budou chráněny plastovými trubkami.

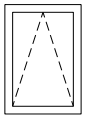
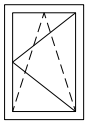
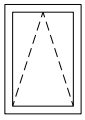
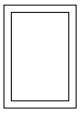
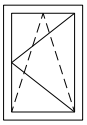
XIII. IZOLACE STAVBY PROTI ZEMNÍ VLHKOSTI A RADONU

. Jako vodorovná a svislá izolace bude použita izolace značky ALFOBIT s hliníkovou nosnou vložkou. Svislá izolace je chráněna geotextilií a nopovanou fólií značky JUNOP. Díky tomu, že je podzemní voda níže, než základy, nehrozí jejich poškození. V případě přebytečné vody jsou okolo celého objektu vybudovány plastové drenážní trubky, které ji odvedou.

TABULKY VÝROBKŮ PSV

PLASTOVÉ VÝROBKY

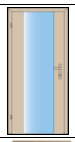
OZNAČENÍ	ZOBRAZENÍ	POPIS	ROZMĚRY V MM (ŠxV)	POČET KUSŮ			KOVÁNÍ	ZASKLENÍ	BARVA
				PODLAŽÍ		CELKEM			
				1.NP	2.NP				
O1		OKNO PLASTOVÉ NEOTVÍRAVÉ, BEZ VENTILACE	500x4400	1	0	1	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O2		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	500x1970	1	0	1	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O3		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	1500x500	1	0	1	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O4		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	500x2000	0	3	3	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O5		OKNO PLASTOVÉ NEOTVÍRAVÉ, BEZ VENTILACE	3700x2300	0	1	1	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O6		OKNO PLASTOVÉ NEOTVÍRAVÉ, BEZ VENTILACE	3000x2300	0	2	2	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O8		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	500x500	0	1	1	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O9		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	2500x500	0	2	2	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	

O10		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	2000x500	0	3	3	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O11		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	900x2300	0	1	1	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O12		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	2000x2300	0	1	1	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O13		OKNO PLASTOVÉ NEOTVÍRAVÉ, BEZ VENTILACE	3100x2300	0	2	2	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
O14		OKNO PLASTOVÉ JEDNOKŘÍDLÉ S VENTILACÍ	1000x2300	0	3	3	CELOOBVODOVÉ KOVÁNÍ ROTO NT - BARVA STŘÍBRNÁ	TEPELNĚ IZOLAČNÍ DVOJSKLA	
CELKEM VÝROBKŮ						**21**			

OCELOVÉ VÝROBKY

OZNAČENÍ	POPIS	ROZMĚRY V MM (ŠxV)	POČET KUSŮ			BARVA
			PODLAŽÍ		CELKEM	
			1.NP	2.NP		
K1	OCELOVÉ ZÁRUBNĚ	800x1970	4	6	10	PERLEŤOVÁ ZLATÁ – RAL 1036
K2	OCELOVÉ ZÁRUBNĚ	600x1970	1	3	4	PERLEŤOVÁ ZLATÁ – RAL 1036
CELKEM VÝROBKŮ					**14**	

TRUHLÁŘSKÉ VÝROBKY

ZOBARZENÍ	POPIS	ROZMĚRY V MM	POČET KUSŮ				KOVÁNÍ	ZASKLENÍ	BARVA
			PODLAŽÍ			CELKEM			
			1.PP	1.NP	2.NP				
	DVEŘE JEDNOKŘÍDLÉ, VNĚJŠÍ	800x1970	0	1	0	1	KLIKA – MT 00 LIGHT-R	MLÉČNÉ SKLO	WENGE
	DVEŘE JEDNOKŘÍDLÉ, VNITŘNÍ	800x1970	3	1	5	9	KLIKA – ENTERO-R,H	MLÉČNÉ SKLO	ZEBRANO
	DVEŘE JEDNOKŘÍDLÉ, VNITŘNÍ	600x1970	1	1	0	2	KLIKA – ENTERO-R,H	-	ZEBRANO
CELKEM VÝROBKŮ						**13**			

VÝPIS PREFABRIKÁTŮ A OCELOVÝCH NOSNÍKŮ

PROFILOVANÉ OCELOVÉ NOSNÍKY

OZN.	POPIS PRVKU	POČET KUSŮ	ROZMĚRY V MM (ŠxDxV)
1	I č.200	5	90x6800x200
2	I č.200	3	90x1200x200
3	I č.200	2	90x5150x200
4	I č.200	21	90x6400x200
5	I č.200	2	90x5450x200
6	I č.200	9	90x4500x200
7	I č.200	2	90x6675x200
CELKEM VÝROBKŮ		44	

ŽELEZOBETONOVÉ PŘEKLADY

OZN.	POPIS PRVKU	POČET KUSŮ	ROZMĚRY V MM (ŠxDxV)
Q	PŘEKLAD	1	240x3300x250
W	PŘEKLAD	1	240x5000x250
G	PŘEKLAD	1	240x7300x250
S	PŘEKLAD	1	240x7500x250
CELKEM VÝROBKŮ		4	

SKLADBY PODLAH A STŘEŠNÍCH KONSTRUKCÍ

SKLADBY PODLAH

- VIZ VÝKRES 1.NP

P5 – KERAMICKÁ DLAŽBA
BETON Z KERAMZITU 100mm
HYDROIZOLACE ALFOBIT
BETON Z KERAMZITU 100mm
ŠTĚRKODRŤ

P6 – BETONOVÁ MAZANINA
BETON Z KERAMZITU 100mm
HYDROIZOLACE ALFOBIT
BETON Z KERAMZITU 100mm
ŠTĚRKODRŤ

P7 – PLOVOUCÍ PODLAHA
BETON Z KERAMZITU 100mm
HYDROIZOLACE ALFOBIT
BETON Z KERAMZITU 100mm
ŠTĚRKODRŤ

P8 – BETONOVÁ MAZANINA (SKLON 1%)
BETON Z KERAMZITU 100mm
HYDROIZOLACE ALFOBIT
BETON Z KERAMZITU 100mm
ŠTĚRKODRŤ

- VIZ VÝKRES 2.NP

P0 – PARKETOVÁ PODLAHA
BETON Z KERAMZITU 100mm
HYDROIZOLACE ALFOBIT
BETON Z KERAMZITU 100mm
ŠTĚRKODRŤ

P1 – KERAMICKÁ DLAŽBA
BETON Z KERAMZITU 100mm
HYDROIZOLACE ALFOBIT
BETON Z KERAMZITU 100mm
ŠTĚRKODRŤ

P2 – MARMOLEUM
BETON Z KERAMZITU 50mm
STROP – HURDIS 2

P3 – PLOVOUCÍ PODLAHA
BETON Z KERAMZITU 50mm
STROP – HURDIS 2

P9 – PLOVOUCÍ PODLAHA
BETON Z KERAMZITU 100mm
HYDROIZOLACE ALFOBIT
BETON Z KERAMZITU 100mm
ŠTĚRKODRŤ

SKLADBA PULTOVÉ STŘECHY

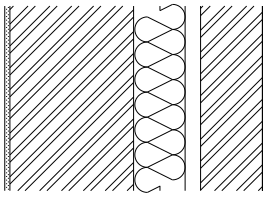
- VIZ. VÝKRES PULTOVÉ STŘECHY

- STŘEŠNÍ KRYTINA KMB Beta – ELEGANT VIŠŇOVÁ
- PRKENNÉ POBITÍ
- POJISTNÁ IZOLACE BETAFOL
- KROKEV

VÝPOČET TEPELNÝCH PROSTUPŮ

VENKOVNÍ STĚNA

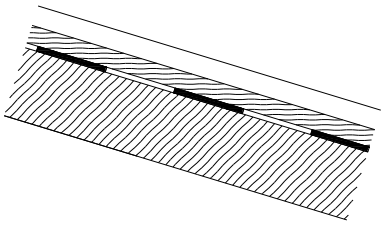
SENDVIČOVÉ ZDIVO KMB SENDWIX L



Konstrukce	Tloušťka celkem	Tepelně technické parametry			Difusně technické parametry				typové označení
typové označení	mm	U	R	Tsi	Gk	Gv	Gv - Gk	Bilance	
		W/(m²K)	(m²K)/W	°C	Zkondenzované	Vypařené			
					g/m²	g/m²	g/m²	+ / -	
L 2410	495/451	0,338	2,791	19,48					L 2410
L 2910	545/501	0,331	2,849	19,51					L 2910
L 2412	520/476	0,289	3,291	19,70					L 2412
L 2912	570/526	0,284	3,349	19,72					L 2912
L 2414	540/496	0,253	3,791	19,86					L 2414
L 2914	590/546	0,249	3,849	19,88					L 2914
L 2416	560/516	0,224	4,291	19,99	Nedochází ke kondenzaci vodních par				L 2416
L 2916	610/566	0,221	4,349	20,00					L 2916
L 2418	580/536	0,202	4,791	20,09					L 2418
L 2918	630/586	0,199	4,849	20,10					L 2918
L 2420	600/556	0,183	5,291	20,18					L 2420
L 2920	650/606	0,181	5,349	20,18					L 2920

Nosná stěna vyhovuje, nebude docházet k srážení vlhkosti na povrchu kce.

PULTOVÁ STŘECHA

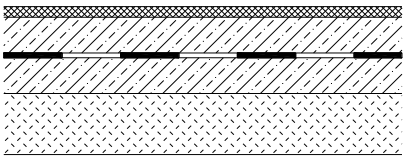


U střešní konstrukce tvořené rovným izolovaným stropem a šikmou či strmou střechou se její podstřešní prostor může považovat za tepelně stejnorodou vrstvu s tepelným odporem:

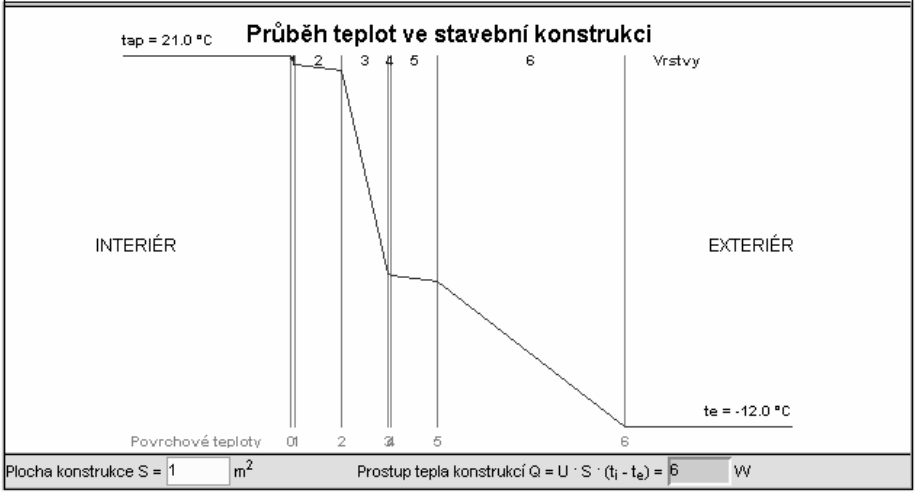
0.06 m^2K/W	pro střechu se skládanou netěsněnou krytinou, bez prkenného podbití
0.20 m^2K/W	pro střechu se skládanou těsněnou krytinou, s prkenným či podobným podbitím
0.30 m^2K/W	pro střechu se skládanou těsněnou krytinou, s prkenným či podobným podbitím a vnitřním lícem střechy opatřeným materiálem s nízkou emisivitou (hliník a pod.)

Střešní kce vyhoví, nebude docházet k srážení vlhkosti na povrchu kce.

PODLAHA 1.NP



				Vnitřní výpočtová teplota místnosti (podle ČSN 06 0210:1994)	t_i =	20	°C	???
				Výpočtová teplota vnitřního vzduchu (dle ČSN 73 0540 se pro obytné budovy volí $t_{ap} = t_i + 1$)	t_{ap} =	21	°C	???
✓				Odpor při přestupu tepla na vnitřní straně konstrukce	R_{si} =	???	$t_{si,0}$ = 21	°C ???
		Materiál	d [m]	λ [W/mK]				
interiér	1.	Koberec	0.008	0.065	R_1 = 0.123	m²K/W	$t_{si,1}$ = 20.24	°C ???
	2.	Beton hutný	0.100	1.23	R_2 = 0.081	m²K/W	$t_{si,2}$ = 19.74	°C ???
	3.	Pěnový polystyren extrudovaný - EXP	0.1	0.034	R_3 = 2.941	m²K/W	$t_{si,3}$ = 1.62	°C ???
	4.	Foalbit	0.005	0.2	R_4 = 0.025	m²K/W	$t_{si,4}$ = 1.47	°C ???
exteriér	5.	Beton hutný	0.100	1.23	R_5 = 0.081	m²K/W	$t_{si,5}$ = 0.97	°C ???
	6.	Křemelina	0.400	0.19	R_6 = 2.105	m²K/W	$t_{si,6}$ = -12	°C ???
			Σd = 0.713	m	R_N = 5.36	m²K/W ???		
✓				Odpor při přestupu tepla na vnější straně konstrukce	R_{se} =	???	t_e = -12	°C ???
Součinitel prostupu tepla U = 0.19			W/m²K	Tepelný odpor konstrukce	R_T = 5.36	m²K/W ???		



Plocha konstrukce S = 1 m² Prostup tepla konstrukcí Q = U · S · (t_i - t_e) = 6 W

Konstrukce vyhovuje. Podlaha vyhovuje, nebude docházet k srážlivosti vlhka na povrchu kce.

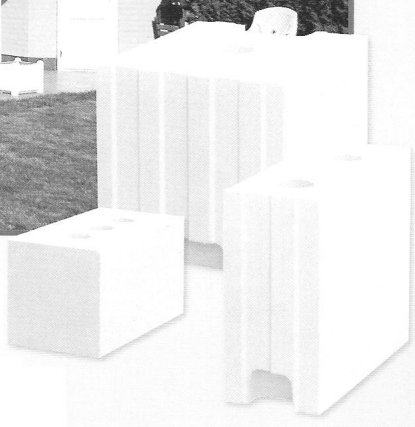
PŘÍLOHY

KM BETA

Technická příručka

ZDICÍ SYSTÉM SENDWIX





KM BETA

Technická příručka

BETONOVÁ STŘEŠNÍ KRYTINA



VIZUALIZACE

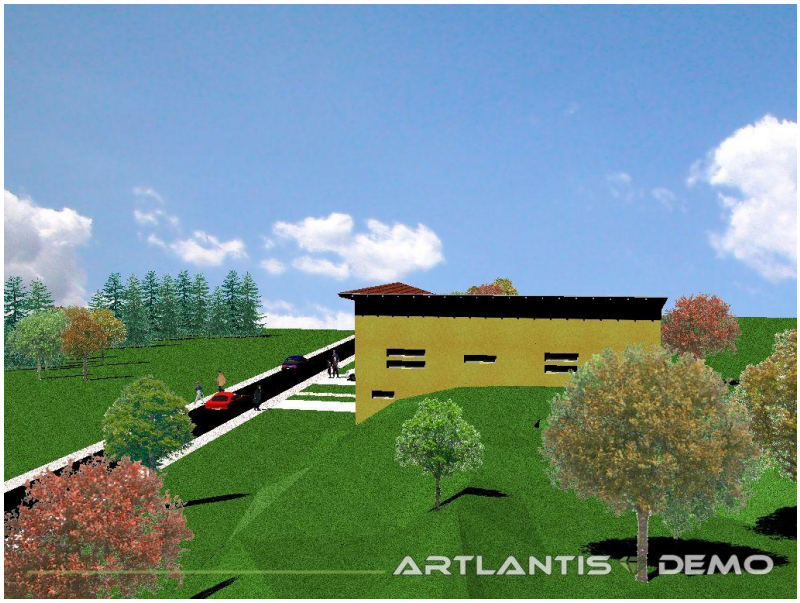

ARTLANTIS DEMO

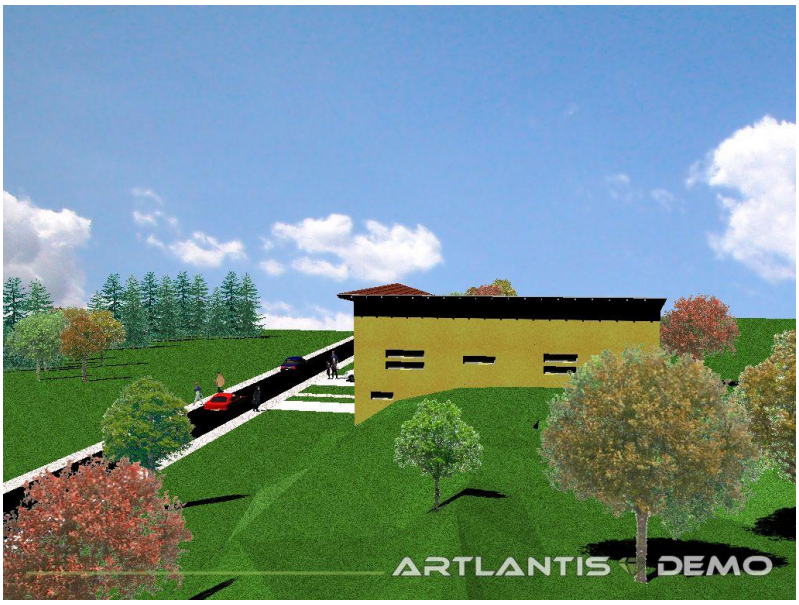

ARTLANTIS DEMO


ARTLANTIS DEMO


ARTLANTIS DEMO


ARTLANTIS DEMO


ARTLANTIS DEMO


ARTLANTIS DEMO

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Mini GLOBE Games
"Pošli to dál"

Minh Ngoc Mai, Jakub Prokeš

Gymnázium, Kadaň, 5. května 620, příspěvková organizace
5. května 620, 432 01 Kadaň

Mini GLOBE Games je ekologická soutěž pro žáky základních škol okresu Chomutov. Jako vlastní projekt byla soutěž připravena a uskutečněna týmem studentů Semináře z ekologie 2. ročníku Gymnázia Kadaň. Studenti navrhli náměty úkolů, připravili pomůcky, promysleli organizaci a rozpočet, zajistili pozvání škol, sponzorské peníze a dary, program a občerstvení pro účastníky, propagaci a vyhodnocení. S pomocí spolužáků také soutěž uskutečnili. Soutěž byla financována z dotace MŠMT Program rozvoje EVVO na školách v roce 2008.

Hlavní myšlenkou soutěže bylo vzbudit zájem žáků o životní prostředí, ve kterém žijí. Soutěž se uskutečnila 15. května 2008 a zúčastnilo se jí 23 tříčlenných týmů ze 13 základních škol a víceletých gymnázií okresu Chomutov. Soutěžilo se ve 2 kategoriích (4. - 5.tř., 6. – 9. tř.). Žáci v terénu v okolí místní cyklostezky postupně řešili úkoly na 10 stanovištích:

Gymnázium Kadaň

1. poznávání stop zvířat
2. vlastnosti vody v řece Ohři (teplota, pH, průhlednost)
3. pavouk (hra na týmovou spolupráci)
4. určování mraků a pokryvnosti oblohy
5. Kimova hra (na vizuální paměť)
6. symboly turistických map
7. meteorologická měření
8. určování dřevin
9. třídění odpadů
10. ohrožené druhy zvířat

Soutěž nebyla jen vědomostní, ale hodnotila se také schopnost týmové spolupráce. Mini GLOBE Games na naší škole proběhly poprvé, ohlas účastníků byl příznivý . Rádi bychom, aby se soutěž stala tradicí a opakovala se každoročně..



Stanoviště č. 10 - poznání druhů ohrožených zvířat



Stanoviště č. 2 – hydrologická měření



Stanoviště č. 9 - třídění odpadu

Ukázky zadání úkolů:

Stanoviště č.3 Pavouk

Kategorie: mladší	Číslo skupiny:	Body:	Spolupráce:

Dostali jste míček a kroužek, na kterém je uvázáno 6 provázků . Vaším úkolem je položit míček na kroužek, rozdělit si provázky (každý dva) a projít určenou trasou tak, aby provázky byly stále napnuté a míček nespadl. Za úkol můžete získat maximálně 10 bodů. Za každý pád vám jeden bod bude odečten. Na úkol máte 10 minut.

Stanoviště č.4 Mraky

Zadání: Přiřaďte názvy mraků k obrázkům a zařaďte do výškových pásem.

Názvy mraků (lístečky k přiřazování) :

Cirrocumulus
Cirrus
Altocumulus
Stratus
Cumulonimbus

Potřeby: zalaminované fotografie mraků (15 ks), 2x pruh modré látky 1x1,5m, papíry se zadáním, papíry s tabulkami a lístečky s názvy mraků.

Příklad fotografie mraků



Stanoviště č. 6 Symboly turistických map

Starší

Soutěžící se budou pohybovat po vyznačené trase, na které budou umístěny obrázky s popisky a čísly. Na stanovišti bude soutěžícím družstvům přidělen papír s tabulkou a se symboly, které budou mít za úkol přiřadit do tabulky. Psací potřeby budou mít soutěžící svoje nebo budou přiděleny na stanovišti (záleží na pozdější domluvě). Po dokončení úkolu bude papír s vyřešeným úkolem vrácen na stanoviště, kde bude vyhodnocen.

Zadání: K číslům v tabulce přiřaďte čísla jednotlivých symbolů, které odpovídají obrázkům a nápisům na cestě.

Symbol							
Číslo symbolu	1	2	3	4	5	6	7
Symbol							
Číslo symbolu	8	9	10	11	12	13	14
Symbol							
Číslo symbolu	15	16	17	18	19	20	21

Číslo objektu	1.	2.	3.	4.	5.	6.	7.
Číslo symbolu							



Stanoviště č. 7: meteorologické měření

Starší

Kategorie: starší	Číslo skupiny:	Počet bodů:	Spolupráce:

Úkoly:

1. V modelu meteorologické budky odečti co nejpřesněji
   - minimální, okamžitou a maximální teplotu
   - objem dešťových srážek v ml

Minimální teplota	
Okamžitá teplota	
Maximální teplota	
Objem dešťových srážek v ml	

2. Do tabulky doplň k uvedeným průměrným teplotám srážky a název popisovaného ekosystému(biomu)

Průměrné roční teploty	Množství srážek v roce	Název ekosystému
-5°C až -15°C		
5°C až 20°C		
18°C až 30°C		
5°C až -5°C		
-4°C až 30°C		

Množství vody v roce	dostatek	občasný nedostatek	nedostatek

Ekosystém (biom)	step	tundra	tajga	poušť	tropický prales

Stanoviště č. 9 Třídění odpadu

Před sebou máte deset druhů odpadků. Vaším úkolem je správně je roztřídit do připravených popelnic. Za každý správně umístěný obal získáte 1 bod.

č.1	typ obalu	body
1:	igelitový sáček	
2.	noviny	
3.	plastový kelímek	
4.	ohryzek od jablka	
5.	plechovka od barvy	
6.	krabice od mléka	
7.	pet láhev	
8.	obal od čokolády	
9.	sklenice	
10.	krabice z lepenky	

Středoškolská technika 2008

**Setkání a prezentace prací
středoškolských studentů na ČVUT**

ZELENÁ STŘECHA NA PANELOVÉM DOMĚ

Petr Bezděk

Střední průmyslová škola stavební a obchodní akademie
Komenského562, Kadaň

Prohlašuji, že jsem práci vypracoval samostatně a že jsem použil pouze literaturu a prameny uvedené v seznamu

V Kadani 6. března 2008

OBSAH

ÚVOD……...4
CO JE VEGETAČNÍ STŘECHA……5
SEZNÁMENÍ S PROBLÉMEM……..6
VÝHODY A NEVÝHODY……7
FUNKCE A ÚČINKY ZELENÝCH STŘECH……..10
STATICKÉ ŘEŠENÍ ……12
SOUVRSTVÍ VEGETAČNÍ STŘECHY ……..15
DALŠÍ TECHNICKÉ ŘEŠENÍ……..18
ANKETA……20
VIZUALIZACE ……22
ZÁVĚR ……..26
POUŽITÁ LITERATURA A PODĚKOVÁNÍ ……27

ÚVOD

Jsem student Střední průmyslové školy stavební a Obchodní akademie v Kadani. Jako téma, pro mou středoškolskou odbornou činnost, jsem si vybral vegetační střechy umístěné na panelových domech.

Tuto tématiku jsme si vybral, protože pocházím ze severních Čech, kde je jak je všem známo, velice rozšířený těžební, energetický a chemický průmysl. Provoz elektráren, těžebních dolů a továren přináší mnoho negativních vlivů na životní prostředí v tomto regionu.

V loňském ročníku SOČ jsem se zúčastnil školního, okresního, krajského a dále jsem postoupil do republikového kola s prací týkajících se zelených střech umístěných na panelových domech, kterou jsem nazval Zelené oázy uprostřed šedi panelových sídlišť.

V letošním ročníku bych se chtěl zaměřit přímo na konkrétní objekt (panelový dům) a vytvořit na něm zelenou střechu - prozatím jen „na papíře‘‘. Dále se více zaměřit na únosnost střešní konstrukce a více propracovat samotnou skladbu zelené střechy.



Obr.1 Na tento panelový dům navrhuji zelenou střechu.

CO JE VEGETAČNÍ STŘECHA

Vegetační střecha neboli zelená střecha je netypické řešení samotného pláště střechy. Vegetační střecha nahrazuje klasickou krytinu (střešní tašky, plech, asfaltové folie) za vegetaci (trávy, keře, byliny). Takže vegetační střechu si můžeme představit, jako zahradu nebo park na střeše budovy a je jedno, jestli se jedná o obytný, administrativní, průmyslový nebo skladovací objekt. Zelenou střechu dnes můžeme navrhovat téměř na jakoukoliv střešní konstrukci a může se jednat o novostavbu tak i o adaptaci.

Zelená střecha není vůbec žádnou novinkou. Vznikaly již za dávných dob na střechách egyptských paláců, chrámových stavbách i na střechách obydlí prostých lidí. Nejznámější zelená střecha se v minulosti objevila na visutých zahradách královny Semiramis v Babylonu.

Dnes můžeme navrhovat zelené střechy na ploché, i šikmé střechy. Rozlišujeme dva základní druhy vegetačních střech a to extenzivní ozelenění a intenzivní ozelenění.

Extenzivní vegetační střecha je jednodušší na svojí konstrukční skladbu. Plní pouze estetickou a ekologickou funkci. Je nepochozí tzn., že na ní není zajištěn přístup osob. Samotná vrstva substrátu je malá a je vhodná pro pěstování nenáročných rostlin (trávy, rozchodníky, skalničky)

Intenzivní vegetační střecha je naopak náročnější na svou konstrukční skladbu. Vrstva substrátu může dosahovat i jednoho metru. Musíme zajistit bezpečný přístup na střechu, protože tento typ ozelenění kromě estetické a ekologické funkce plní také funkci zahrady nebo parku. Na této střeše můžeme pěstovat velké spektrum rostlin, keřů i stromů.




Obr.1 a 2 Příklady již realizovaných vegetačních střech.

SEZNÁMENÍ S PROBLÉMEM

Vegetační střecha je konstrukce, která má hodně výhod, ale v dnešní době jí na budovách vidíme jen zřídka. Investory od výstavby zelených střech odrazují vyšší pořizovací náklady a řekl bych i malá informovanost, protože ve finále je zelená střecha velice ekonomický a ekologický prvek stavby. Především slouží jako tepelný, protipožární a zvukový izolant objektu, její životnost při správném technickém provedení je téměř nekonečná a nepotřebuje další údržbu.

V mé práci se snažím zelenou střechu navrhnout na střechy panelových domů a tím zlepšit životní prostředí v dnešních zašedlých sídlištích. Zelené plochy na dnešních sídlištích vytlačují parkovací plochy a komerční objekty. Kde tedy mají trávit čas maminky s dětmi, důchodci a obyčejní lidé. Jediné volné plochy na sídlištích, které se dají využít jsou pouze ploché střechy.

Bez problému si můžeme představit, že na střeše místo zašedlého asfaltového pásu bude malý privátní park pro obyvatele domu. Park, kde se mohou lidé scházet, popíjet kávu, hrát hry, vychutnávat čerstvý vzduch a to vše ve výšce s krásným výhledem na okolí. Ve výšce nad shonem dnešní doby.



Obr.3 Pohled na šedé střechy panelových domů.

VÝHODY A NEVÝHODY

VÝHODY

Tepelně izolační účinek: Přispívají k lepší tepelně technické stabilitě v letním či zimním období v objektu. V létě má střecha ochlazující účinek díky stínu vegetace, vypařováním vody a odrazem paprsků od vegetace. V zemině vzniká malé množství tepla které v zimě přispívá k tomu, že zemina nepromrzne a slouží jako tepelná izolace.

Čištění vzduchu: Rostliny mohou filtrovat částice prachu a nečistot ze vzduchu.Částice se zachytí na povrchu lístků a déšť je pak spláchne do substrátu. Rostliny mohou rovněž absorbovat plynné škodliviny a aerosoly.
Snižování prachu: Zeleň výrazně snižuje zahřátí střešní plochy, tím se snižuje víření prachu v městských aglomeracích.

Regulace vlhkosti: Při deštích substrát získává vodu, která se drží v hydroakumulační vrstvě, ze které při velkých teplotách vegetace čerpá vláhu a vypařováním se vyrovnává vlhkost vzduchu.

Estetika: Na rozdíl od střechy se štěrkopískovým nebo asfaltovým povrchem působí střecha porostlá trávou a bylinami svou přírodní krásou příznivě na lidskou psychiku.

Životnost: Trvanlivost ozeleněné střechy při správné volbě střešního pláště a odborném provedení spojů a detailů je prakticky neomezena.Střecha je rovněž plně chráněna před UV zářením a mechanickým poškozením

Zvětšují zelené plochy ve městě: Jak je známo, ve městech není moc zelené plochy, díky vegetačním střechám se množství zelných ploch zvětšuje.

Působí jako zvuková izolace: Vegetační vrstva se substrátem působí jako výborná zvuková izolace.

Jsou téměř nehořlavé: Zemina je nehořlavý materiál, a tak chrání celou střešní konstrukci před případným požárem.

Zpomalují odtok dešťové vody: Dešťová voda se zadržuje ve vegetaci. Odtéká minimum vody, zadržená voda se později odpaří, čímž se snižuje zatížení veřejné kanalizace.

Šíří aromatickou vůni: Díky rostlinám zlepšují kvalitu okolního kyslíku a příjemnou vůni rostlin.

Možnost získání finanční dotace: Některá města v Evropské Unii udělují dotace na financování zelených střech.

Umožňují lepší výhled a vytváří životní prostor pro hmyz

NEVÝHODY

Údržba: Vybudovaná vegetační střecha vyžaduje nutnost údržby zeleně zejména u intenzivních střech

Statické posouzení: U střech s vyšším vegetačním souvrstvím je nutné statické posouzení, případně zesílení nosné střešní konstrukce.

Bezpečnost: Z hlediska pobytu lidí na střeše je nutné zabezpečit proti možnému pádu zábradlím apod.

Finance: Při jejich realizaci je nutno vynaložit vyšší finanční prostředky v důsledku vyšší pracnosti, zařízení vegetačního souvrství a výsadby zeleně.

FUNKCE A ÚČINKY ZELENÝCH STŘECH

Vlivem koncentrace budov a dopravy je život v našich městech nezdravý. Auta a topná zařízení spotřebovávají množství kyslíku a produkují nadbytek škodlivin. Obrovské betonové a asfaltové plochy vedou k přehřívání klimatu ve městě a způsobují, že teplý vzduch zvedá ze země částice nečistot a škodlivin a víří je do všech stran. V létě lze ve večerních hodinách v centru velkoměsta naměřit teplotu vzduchu o 4 – 11°C vyšší než v okrajových čtvrtí. Ozeleněním střech se nezdravé klima zásadně může změnit: vzduch se vyčistí, víření prachu se výrazně sníží a zatěžující kolísání teplot se výrazně vyrovná. Pro dosažení zdravějšího klimatu ve městě by pravděpodobně stačilo dodatečně ozelenit 10 - 20% střešních ploch ve městě. Když se zamyslíte, jaké budovy tvoří naše města, zjistíte, že nejčastěji vyskytující se objekty jsou právě panelové domy, jejichž střechy by se daly využít pro zlepšení klimatu a psychické pohody obyvatel. (Česká Republika byla v průzkumu označena za zemi, kde je největší koncentrace panelových domů, které se zde začaly stavět již v 70. letech minulého století). Města se zabývají nedostatkem parkovacích ploch, ale zapomínají na nedostatek zeleně, o kterou nás stále připravují. Náš projekt řeší nedostatek zelených ploch, relaxačních míst v panelových sídlištích, zlepšení klimatu a ekologicko-ekonomické výstavby.

1. Ekonomická funkce
Zelená střecha slouží jako ochrana bytů pod ní. Zejména před nadměrným ohříváním v létě a ochlazovaní v zimě. Tím, že eliminujeme tyto teplotní výkyvy, prodlužujeme životnost krytiny a snižujeme nároky na její údržbu. Avšak pořizovací náklady jsou vyšší. V zahraničí je dokonce možné získat finanční dotaci na takovéto střechy a to zejména v hustě zastavěných oblastech.

2. Ekologická funkce
Rozšíření zelených střech a to zejména na panelových domech, kterých v České republice je velké množství, může přispět k lepšímu životnímu prostředí. Mizející plochy zeleně, které bývají vystřídány betonem a asfaltem, se mohou nahradit zelení na střechách. Právě zelených ploch je v dnešních sídlištích poměrně málo. Zelené střechy na panelových domech by se podílely na zlepšení vlastností městského prostředí a to obohacováním vzduchu kyslíkem a vázáním oxidu uhličitého. Podle výzkumů plocha 25 m2 listnaté zeleně vyprodukuje za den tolik kyslíku, kolik spotřebuje člověk za stejný čas dýcháním.

3. Estetická funkce
Pokud bydlíte uprostřed panelákového sídliště a k tomu ve vyšším patře, první co uvidíte, když se podíváte z okna, budou další paneláky a zejména šedé asfaltové střechy a možná někde v povzdálí zahlédnete trochu zeleně. Plochá střecha se stává pohledově pátou fasádou budovy. Proto stejně jako svislá fasáda i plochá střecha by měla plnit určitá estetická kritéria. Proč právě nepoužít zelenou střechu, která vnese trochu přírody do dané lokality a stane se zelenou oázou v zašedlých sídlištích.

4. Relaxační a odpočinková funkce
Zelená střecha na panelovém domě se rázem stane jakýmsi privátním parkem pro obyvatele daného domu. Na střechu je možné umístit nejrůznější společenské hry, nadzemní bazény a třeba i saunu. Pro děti je zde možné vybudovat pískoviště, klouzačky a prolézačky.



Obr.4 a 5 Fotomontáž. Zelené střechy na všech administrativních budovách v centrech měst Ameriky.

TECHNICKÉ ŘEŠENÍ

STATICKÉ ŘEŠENÍ

Základní problém, který nastal u mé vize je, zda konstrukční systém starý 30 let unese konstrukci zelené střechy.

Nevěděl jsem si rady, a proto jsem kontaktoval statika-projektanta Ing. Vratislava Steinera. Pan inženýr má velkou praxi s výstavbou a projektováním panelových domů. Spočítaly jsme přitížení konstrukce a skladby zelené střechy.
Celkové přitížení je 388,75 Kg / m2

1. Nahodilé zatížení + zatížení od sněhu	305,40 Kg/m2
2. Parotěsná vrstva (Vedagard sk plus)	0,1 Kg/m2
3. Spojovací vrstva (Vedatex adhaesiv)	0,1 Kg/m2
4. Tepelně izolační vrstva	12,00 Kg/m2
5. Ochranná vrstva (Vedatop SU)	0,10 Kg/m2
6. Hydroizolační vrstva (Vedaflor TGF 200)	1,80 Kg/m2
7. Vrstva proti prorůstání kořenů (Vedaflor WS-I)	0,61 Kg/m2
8. Drenážní a filtrační vrstva (Vedaflor DSM 1502)	1,64 Kg/m2
9. Vegetační vrstva	20,00 Kg/m2
10. Keře, trávníky, rostliny	2,00 Kg/m2
CELKEM:	388,75 Kg/m2

Tab.1 Přitížení konstrukce zelenou střechou

Po konzultaci s panem Steinerem jsme došli k závěru, že základy panelových domů jsou předimenzované, a že jsou schopny unést dvakrát tolik zatížení, co přenášejí do základové spáry nyní, to znamená, že základová konstrukce přitížení vyhoví.

Stejně tak příčný stěnový systém je v pořádku a bez problému dokáže přenést zatížení do základů.

Problém, který zde nastal, je, že samotná deska střešní konstrukce nevyhoví, to nám zkomplikovalo situaci a znatelné zvýšení vstupních finančních nákladů. Zatížení skladbou střechy musíme převést do příčného stěnového systému, který je nosný.

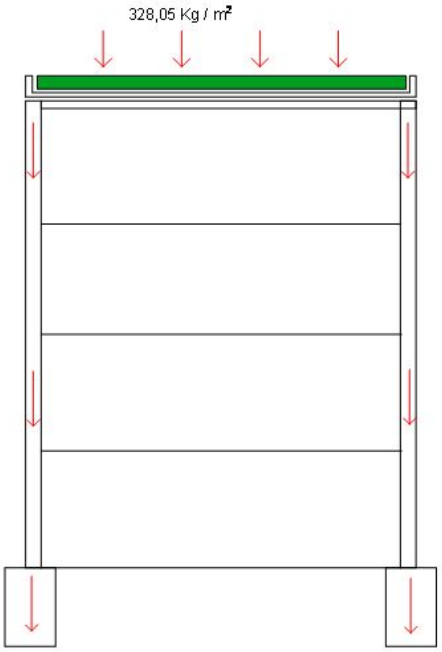
Pro nás to znamená, že musíme odstranit původní střešní konstrukci, nahradit jí roznášecím systémem**,** který roznese zatížení do již zmiňovaného stěnového systému a bude zajištěno spolupůsobení s nosným stěnovým systémem.

Roznášecí systém si můžeme představit ve formě určitého roštu, nebo vany, na které bude vybudována skladba zelené střechy. Dále bude zatížení přecházet ze stěnového systému do základů a dále do zeminy.

V mé práci neřeším dopodrobna přenášení zatížením střechy do konstrukce. Ale abychom se toho problému jednoduše nezbavil, navrhl jsem dvě možná řešení jak této problém vyřešit.

1.Jako první možnost, která je méně radikální je zesílení panelů žebírkovými trámy. Trámy by se betonovaly rovnoměrně vedle sebe na panely v podélném směru. Do původního panelu by se navrtaly ocelové trny a na nich by se vybetonoval trám. Ocelové trny by zajistili spolupůsobení. Betonové trámy by zajistili zpevnění konstrukce. Veškeré rozměry, druh betonu, výztuž atp. je záležitost odborného statika a není to také součást mé práce.

2. Druhá možnost je, že by se kompletně vyměnili střešní panely za panely takové, které dokáží přenést zatížení od skladby zelené střechy. Ovšem tato metoda by citelně poznamenala život obyvatel daného panelového domu. Další nevýhodou je velká finanční náročnost tohoto úkonu.



Obr.6 Na obrázku je znázorněno kudy prochází zatížení od skladby zelené střechy.

Skladba vegetační střechy zatěžuje původní stropní panely zpevněné trámy. Zatížení od celkové střešní konstrukce dále pokračuje přes nosný příčný systém a dále pokračuje do základové konstrukce.
Jak už bylo řečeno nosný stěnový systém a základy přitížení od vegetační střechy vyhoví.

Samotný konstrukční systém panelových domů je zcela v pořádku, je praktický, výstavba je rychlá. Jeho handicapem je, že v minulosti byl takzvaně zneužit předchozím režimem. Není města v ČR, kde by nestál nějaký panelový dům. Byty v něm jsou poměrně dobře dispozičně řešeny, jsou prostorné, prosvětlené. Bohužel dnešní stav panelových domu je takový, že nevyhovuje dnešním normám a standardům. Panelové domy bez rekonstrukce jsou morálně zastaralé a nevyhovující dnešnímu životnímu stylu. Ovšem po celkové rekonstrukci (fasády, oken, jader, TZB , střech atd.) se ze zastaralých bytů okamžitě stávají byty nové generace. A právě tyto bytové panelové domy by díky zelené střeše mohly získat ještě větší exkluzivitu. Cena nemovitosti by stoupla, majitel by mohl požadovat větší nájemné, byty by se lépe prodávali.

SOUVSRVÍ VEGETAČNÍ STŘECHY

V současnosti jsou na našem trhu firmy, které nabízejí kompletní soustavu vrstev pro zelené střechy (Fatrafol, Akroplan, Vaiplan,) Já jsem zvolil pro svou práci produkty od firmy VEDAG. Technologie a produkty těchto firem jsou již ozkoušené a splňují současné normy. Proto není třeba vymýšlet vlastní skladbu střechy.



Obr. 7 Skladba zelené střechy.

1. Nosná střešní konstrukce
2. Parotěsná vrstva (Vedagard sk plus)
3. Spojovací vrstva (Vedatex adhaesiv)
4. Tepelně izolační vrstva
5. Ochranná vrstva (Vedatop SU)
6. Hydroizolační vrstva (Vedaflor TGF 200)
7. Vrstva proti prorůstání kořenů (Vedaflor WS-I)
8. Drenážní a filtrační vrstva (Vedaflor DSM 1502)
9. Vegetační vrstva
10. Keře, trávníky, rostliny

1. Nosná střešní konstrukce:

Jak již bylo řečeno, nosná střešní konstrukce musí být staticky dimenzována na zatížení vegetační střechou. Bohužel střešní panely na panelovém domě toto zatížení nejsou schopny přenášet. Tudíž se musí navrhnout nový roznášecí systém.

2. Parotěsná vrstva - VEDAG SK- PLUS

Parotěsná vrstva chrání skladbu před případnou vlhkostí z objektu. Zejména tepelně izolační vrstvu. Parotěsná vrstva dále plní funkci pojistné hydroizolace.
Materiál Vedag sk – plus je ideální pro lepení tepelné izolace.

3. Spojovací vrstva - Vertex – adhaesiv

Jedná se o asfaltové lepidlo pro lepení tepelné izolace. Toto lepidlo vykazuje vysokou pevnost lepení. Vysoká je i jeho počáteční pevnost a tím je i vysoká bezpečnost proti sání větru. Lepidlo zůstává funkční i za nepříznivých povětrnostních podmínek.

4. Tepelně izolační vrstva – FOAMGLAS

Pěnové sklo FOAMGLAS tvoří velmi pevný podklad pod hydroizolaci, minimalizuje riziko jejího poškození a umožňuje velké přitížení. Další předností kompaktní skladby s izolací z pěnového skla je velmi vysoká pevnost v tlaku a nestlačitelnost tepelně-izolační vrstvy.

5. Ochranná vrstva - VEDATOP SU

Jedná se o elastomerobitumenový podkladní pás. Tato vrstva se klade na tepelnou izolaci jako ochrana před ožehnutím plamenem při pokládce vrchní vrstvy. Dále slouží jako ohnivzdorná izolace objektu. Tento materiál je vysoce odolný vůči povětrnostním vlivům, žáru ohně a přetržení.

6. Hydroizolační vrstva - Vedaflor TGF 200

Jedná se o odolný materiál na bázi plastu, který plní funkci separační. Dále zde slouží jako parotěsná folie. Hydroizolační vrstva je velmi důležitá součást střešní skladby. Zabraňuje vnikání vody do objektu. Jsou tudíž na ni kladeny velké požadavky. Materiál Vedaflor TFG 200 je speciálně navržená folie určená přímo do skladeb vegetačních střech. Jsou zde kladeny velké požadavky na kvalitu provádění, přesnost práce a spojování jednotlivých plastových pásů. Po položení této folie by se měla provést záplavová zkouška.

7. Vrstva proti prorůstání kořenů VEDAFLOR WS

Plastový natavovací pás s měděnou vložkou. Povrch tohoto materiálu je opatřen jemným minerálním posypem. Zaručuje dlouhodobou ochranu stavby proti prorůstání kořenů.

8. Drenážní a filtrační vrstva

Plastová smyčková rohož, která plní funkci drenážní a ochranou, je vyrobena z vysoce kvalitních materiálů. Rohož je opatřena z obou stran zpevněnou rohoží, která zde plní funkci filtru. To znamená, že chrání samotnou skladbu střechy a odpadní vpusti před zanášením a usazováním jemnými částicemi z půdy způsobené závlahami a dešti.

9. Vegetační vrstva – Zahradní substrát Vedaflor E

Jedná se o speciálně vyvinutý zahradní substrát určený pro zelené střechy. Tento substrát obsahuje směs minerálních substrátů a hovězího kompostu. Je vysoce kvalitní s vysokou drenážní a retenční schopností. Je určen pro extenzivní ozelenění. Je ohleduplný k životnímu prostředí, protože se vyrábí recyklací. Vytváří zdravé a živné prostředí pro rostlinstvo.

Vegetace

Jedná se o výchozí prvek skladby střechy, bez které by zelená střecha nebyla zelenou střechou. Do vegetační vrstvy sázíme rostliny, keře trávník. Podle libosti a přání uživatele. Volíme méně náročné rostliny s menším kořenovým systémem.

DALŠÍ TECHNICKÉ ŘEŠENÍ

Únosnost střešní konstrukce:

Únosnost střešní konstrukce je nejdůležitějším požadavkem pro realizaci ozeleňování střechy. Tento požadavek vyřeší statik výpočtem a posouzením, kterým zjistí únosnost konstrukce. A dále navrhne řešení jak konstrukci zesílit.

Přístup na střechu:

Přístup na střechu lze řešit přistavěním schodišťového ramene v posledním patře a vyjmutím jednoho střešního panelu. Kvůli navýšení výšky střešní skladby nelze použít stejné schodišťové rameno, které jsou již v panelových domech. Jako nejlepší řešení je vystavění lehké ocelové konstrukce s kombinací s dřevěnými stupni. Schodiště může být řešeno individuálně. V mém návrhu se jedná o schodiště ve tvaru L. Schodišťový prostor musí být zastřešen a izolován proti dešti, větru apod.

Další varianta je prodloužit výtah až na úroveň střechy, ale je to finančně velmi náročné ovšem velice efektivní. Pokud vezmeme v úvahu, že současný stav výtahů v panelových domech už nevyhovuje dnešním normám a je nutná jejich výměna za moderní a bezpečné výtahy. V tomto případě by bylo velmi výhodné provést realizaci zelené střechy a zároveň výměnu výtahu, který byl vyveden až na ozeleněnou střechu.

Bezpečnost:

Protože náš projekt počítá s tím, že na střeše bude volný pohyb lidí, je zapotřebí zajistit jejich bezpečnost. Po celém obvodu střechy musí být bezpečné zábradlí vysoké minimálně 1100 mm. Na střeše by měli být umístěny dobře dosažitelné hasicí přístroje. Veškeré elektrické rozvody, větrací komínky apod. musí být zabezpečeny tak, aby neohrožovaly lidské zdraví a samy nebyly poškozovány. Toto zabezpečení by nemělo nabourávat estetický ráz střechy.

Dále by se provoz a užívání střechy měl řídit příslušným bezpečnostním řádem. Například: zákaz pohybu dětí bez dozoru rodičů, zákaz venčení psů, zákaz rozdělávání ohně a podobně.

Předpokládám, že na střechu budou mít přístup pouze obyvatelé daného domu. Vstupní dveře na střechu budou zamykatelné. Obyvatelé domu budou mít k dispozici klíč od těchto dveří. Podobně jako u ostatních společných prostor jako jsou kočárkárny, sklepy, sušárny atp.

Užívání střechy bude monitorovat a zaznamenávat, dnes již tak populární, bezpečnostní kamerový systém. Při případném vandalismu a ničení vybavení „zahrady‘‘ by byl použit záznam k usvědčení vandala.

Atika:

Atiky nám poslouží jako zábrana sesouvání zeminy a následnému znečištění fasády. Ne však všechny ploché střechy panelových domů mají dostatečně vysoké atiky, a proto je potřeba je zvýšit podle typu ozelenění (intenzivní, extenzivní). Výška atiky musí přesahovat nad úroveň zeminy.

Osvětlení:

Aby střešní zahrada mohla být bezpečně využívána i ve večerních hodinách, je zapotřebí jí řádně osvětlit. Osvětlení lze řešit například zahradními lampami.

Hromosvod:

Hromosvody se zde nebudou řešit jako klasické ocelové sítě, ale použijeme aktivních hromosvodů.

Přesuny stavebních hmot:

Na přepravu materiálu bude potřeba jeřábů nebo stavebních výtahů a přepravníků, záleží na výšce budovy.

Zavlažování:

Nejběžnější a nejjednodušší je zálivka shora, její významnou výhodou je přirozenost a účinnost obdobná jako déšť v přírodě. Technicky náročnější je spodní zálivka, která má však výhodu dlouhodobějším zajištění vody, případně i s rozpouštěnými živinami pro rostliny. Tato spodní zálivka je zajištěna pomocí kapacích trubek, kde jsou rozestupy kapání po 30 cm.

Spádování:

I vegetační střecha musí být v určitém spádu, ideální je 1,5 - 2% , který je vytvořen pomocí spodních vrstev. Vegetační vrstva je v rovině. V našem případě využijeme tepelnou izolaci jako spádovací prvek .

Odpady a výpustě:

Odpady pro odtok dešťové vody se umisťují do drenážní vrstvy. Měly by být náležitě ochráněny proti ucpání a vyplavování drenážní vrstvy. Tohoto dosáhneme instalací filtračních hlavic na vpusť, které zabrání vnikání nežádoucích předmětů do odpadních trub. Střešních odpadů musí být na střeše dostatečný počet.

ANKETA

Během mého zpracovávání tématu jsem dospěl k problému, že plocha střechy není zas tak velká, aby se na ní vešlo vše, co do běžného parku nebo zahrady. Rozhodl jsem se, že mezi nájemníky panelového domu, na který navrhuji zelenou střechu, udělám anketu.

Někteří nájemníci neměli zájem vůbec odpovídat na otázky. Ale většině se nápad líbil a zajímali se o různé informace.

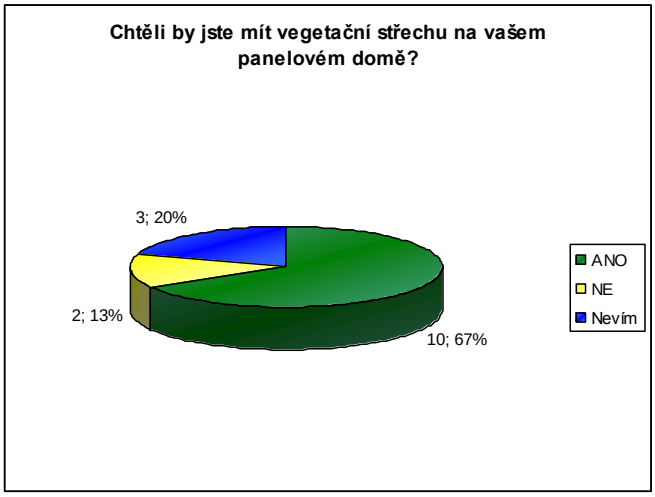
Anketa probíhala tak, že jsem nejprve obyvatelům vysvětlil co znamená pojem zelená střecha, když nevěděli tak, jsem jim pojem objasnil. Dále jsem jim položil tři otázky:

1. Důležité je vědět, jestli jsou obyvatelé spokojeni se zelení v okolí.



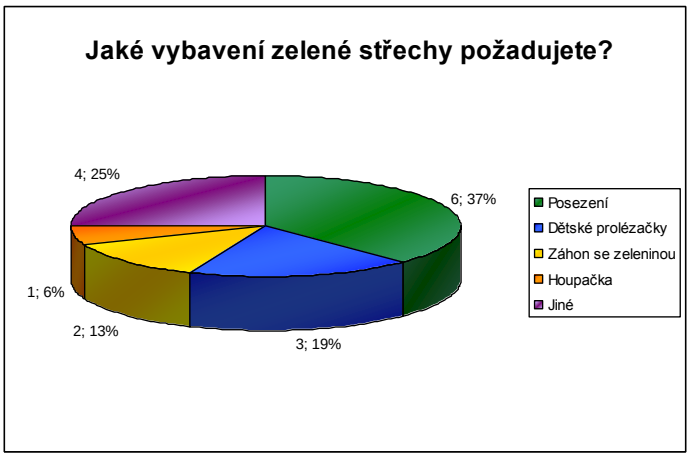
Graf č. 1

2. Neméně důležité je vědět, zda by obyvatelé měli zájem o realizaci vegetační střechy na střeše jejich panelového domu.



Graf č. 2

3. Po objasnění co zelená střecha je, a že by bylo možné jí instalovat přímo na jejich panelový dům jsem, se zeptal, co by si na takové zahradě přáli za vybavení:



Graf č. 3

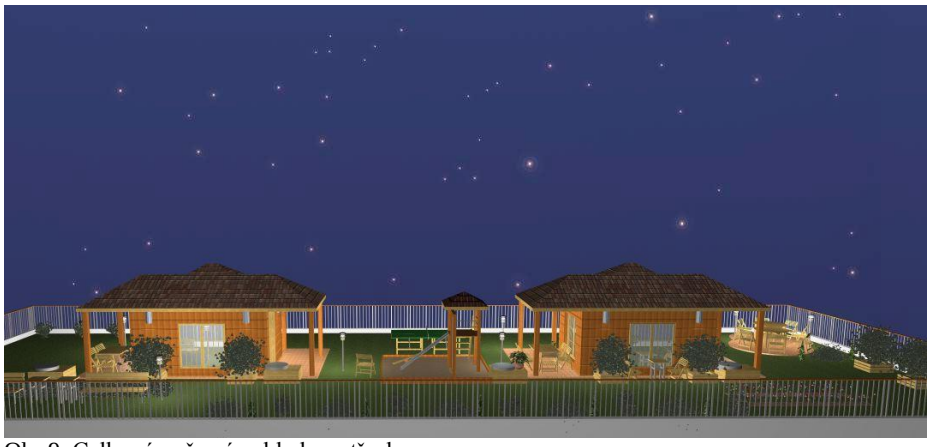
Obyvatelé dále odpovídali, že by si přáli na zelené střeše gril, pískoviště, trampolínu, stolní tenis nebo záhon s květinami a viz. položka ostatní.

VIZUALIZACE:

Také jsem připravil několik 3D vizualizacích jak by taková zelená střecha mohla vypadat. Tyto vizualizace jsou navrhovány na panelový dům v Kadani , Na podlesí č.p. 1463 a 1464.



Obr.8 Celkový pohled na střechu.



Obr.9 Celkový večerní pohled na střechu.



Obr.10 Detailní pohled na záhony a zastřešený vstup.



Obr.11 Zastřešený vstup a lavička k posezení. Odpadkový koš na odpadky.



Obr.12 Večerní pohled na posezení v rohu střechy.



Obr.13 Večerní pohled na stolní tenis a dětské hřiště.



Obr.14 Pohled na dětské hřiště s pískovištěm, prolézačkou, klouzačkou a houpačkou.



Obr.15 Pohled na posezení v rohu, zastřešený vstup a okolí.

ZÁVĚR:

Cílem mé práce bylo zaměřit se na konkrétní panelový dům na, který jsem se pokusil navrhnout vegetační střechu. Dále upozornit na některé technické a konstrukční nedostatky, problémy a přiblížit výhody vegetačních střech. Pokusil jsem se také zaměřit na problematiku panelových domů u nás a pomoci jim zvýšit jejich popularitu díky zeleným střechám.

První problém, na který jsem se zaměřil bylo, zda původní střešní konstrukce unese samotnou vegetační střechu. Po konzultaci s odborným statikem jsme došli k závěru, že se musí provést dodatečné zpevnění střešní konstrukce nebo navrhnout roznášecí systém.

Fakt, že původní střešní konstrukce panelových domů obecně neunese vegetační střechu, zkomplikovalo situaci a nejspíš by se tento problém projevil zvýšením vstupních nákladů. Ovšem žijeme v moderní době, kde je možné téměř vše, obzvlášť ve stavebnictví. Myslím, že tato závažná komplikace, ovšem však řešitelná, by neměla odradit od vize zřízení zelené střechy na střešní konstrukci panelového domu.

Také jsem se musel zabývat přístupem na střechu a bezpečností obyvatel na střeše.

Vyřešit přístup na střechu není žádný závažný problém. Dnešní technologie nám umožňují zajistit bezpečnou, lehkou a estetickou schodišťovou konstrukci.

Bezpečný pobyt na zelené střeše bude zajištěn zábradlím po celém obvodě. Pokud se uživatelé budou chovat zodpovědně a nenechají své děti bez dozoru, nemělo by dojít k žádné ujmě na zdraví, zejména spadnutí ze střechy.

Dále proč zelené střechy navrhovat je fakt, že lidé si na sídlištích nemají kde odpočinout, relaxačních ploch je málo. A opět jediné volné nevyužité plochy jsou střechy. Ovšem problém nastává, jak vybavit takovou zelenou střechu pro obyvatele domu.

Proto jsem uspořádal anketu v jednom vybraném panelovém domě. Díky ní jsem se dozvěděl, jaké potřeby mají obyvatelé a jak by chtěli trávit volný čas ve svém „parku na střeše‘‘ Podle ankety jsem přizpůsobil můj návrh vegetační střechy, tak aby tam každý našel to, co potřebuje.

Zelená střecha je obecně velmi ekologická konstrukce. Nejen, že životní prostředí neznečišťuje, zelená střecha mu dokonce pomáhá s jeho obnovou a čištěním vzduchu. V dnešní době plné globálního oteplování je, myslím zelená střecha příjemný a osvěžující prvek. Když si představíme, že by na každém panelovém domě v České Republice byla zelená střecha, určitě by to zanechalo pozitivní dopad na životní prostředí a kvalitu vzduchu ve městech. Konečný součet plochy střech panelových domů je obrovský. Kdybychom je všechny ozelenily, města by získala nové zelené plochy v řádech tisíců až stovek tisíců metrů čtverečních.

Použitá literatura a materiály:

Časopis stavební obzor
Technické listy Vedag
Zelené střechy – Genom Minke
ERA 21. 1/2001
Projektová dokumentace panelového domu v Kadani

Poděkování:

Poděkování za pomoc při vypracování mé práce patří:

Ing. Milanu Parpelovi
Ing. Miroslavu Pacholíkovi
Ing. Vratislavovi Steinerovi
Stavebnímu úřadu v Kadani
Obyvatelům panelového domu v Kadani, Na podlesí č.p. 1463 a 1464

VOŠ, SOŠ a SOU Kopřivnice, příspěvková organizace

Ukázky studentských prací v programech Mechanical Desktop, Inventor, Catia



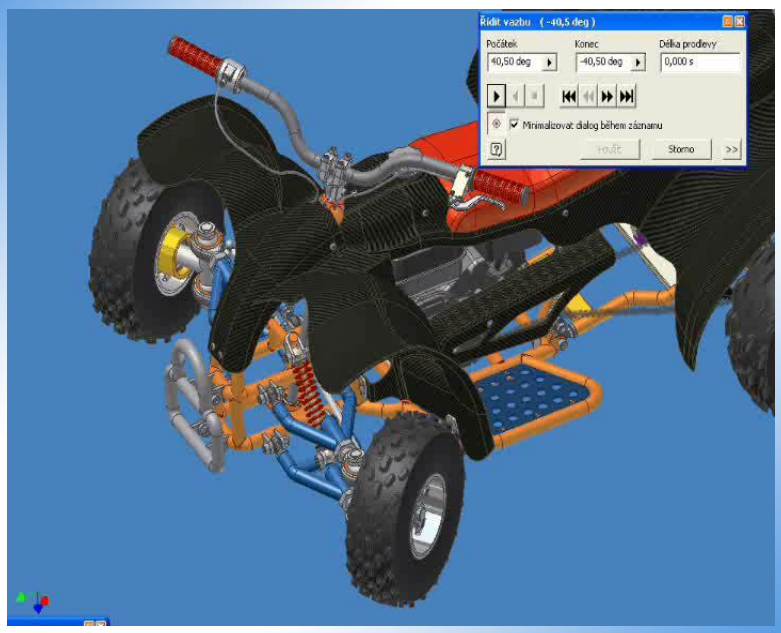
TERÉNNÍ ČTYŘKOLKA

Hlavní části:

- Rám
- Nápravy
- Řízení
- Tlumiče
- Brzdy
- Pohon

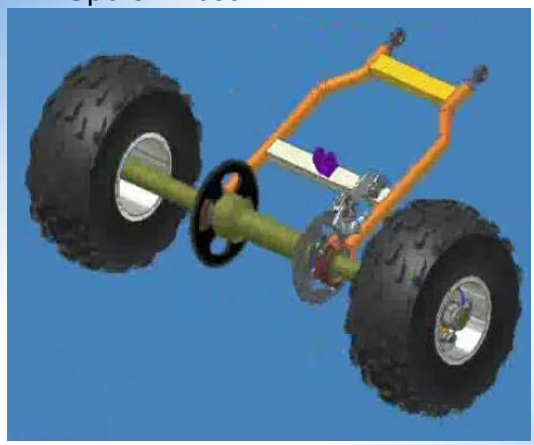


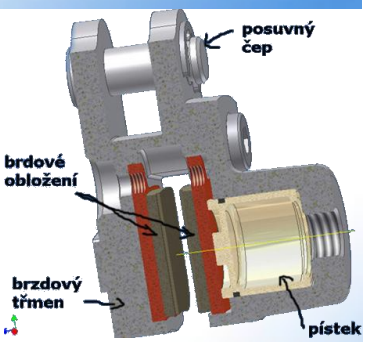
Řízení

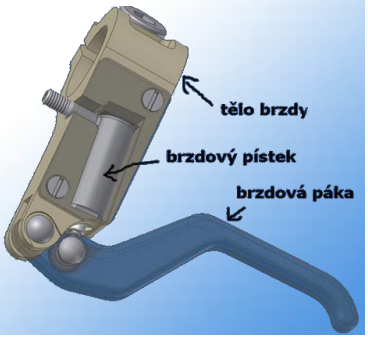


Brzdy

- Bezpečnostní prvek
- Hydraulické kotoučové brzdy
- Spolehlivost





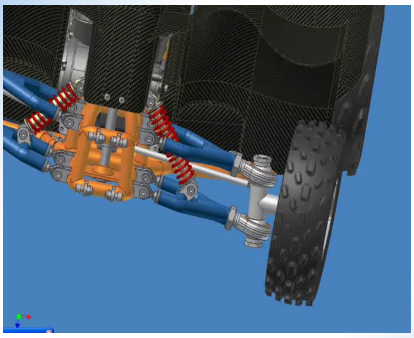


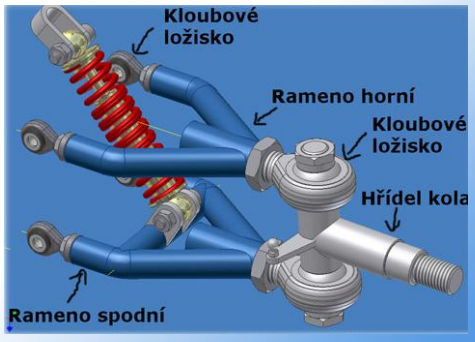
Nápravy

Přední náprava:

•Konstrukční řešení-trojúhelníková ramena

•Možnost nastavovat geometrii-odklon, záklon, příklon

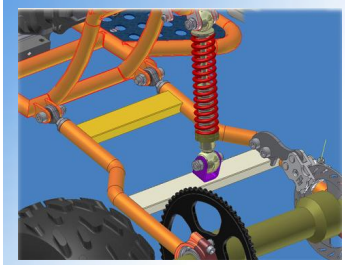


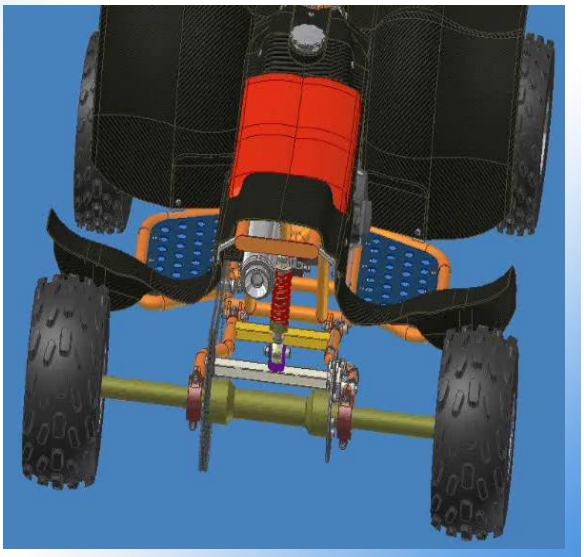


Zadní náprava:

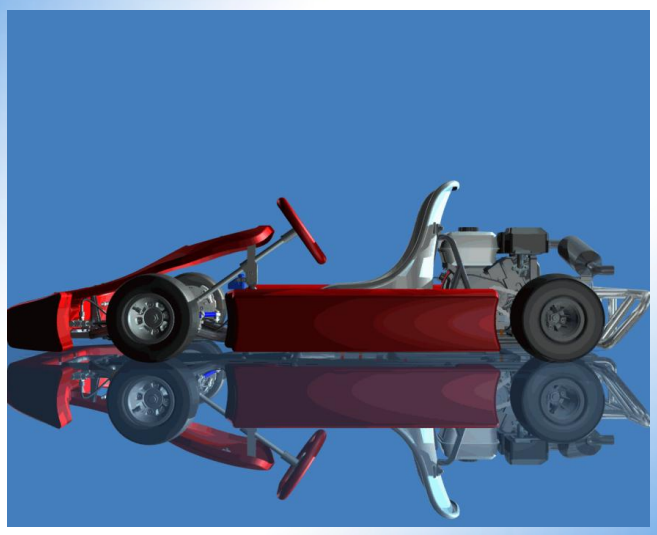
Trubkový svařenec

Centrální brzda

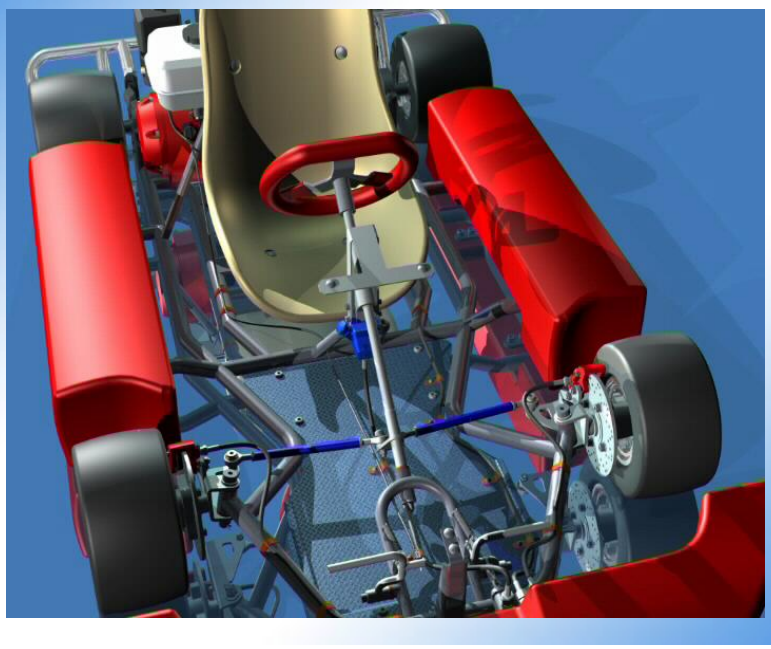




3D model závodní motokáry



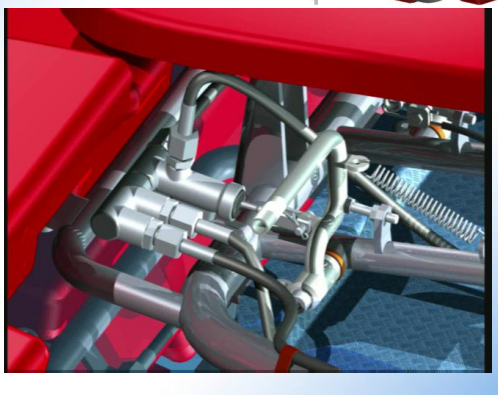
ŘÍZENÍ MOTOKÁRY



Brzdová soustava

- Hydraulický jednookruhový systém
- Přední brzda – plovoucí třmen (jedna pohyblivá destička)
- Zadní brzda – čtyř válečková dvou čelisťová brzda
- Rozvod zajištěn hydraulickými hadicemi

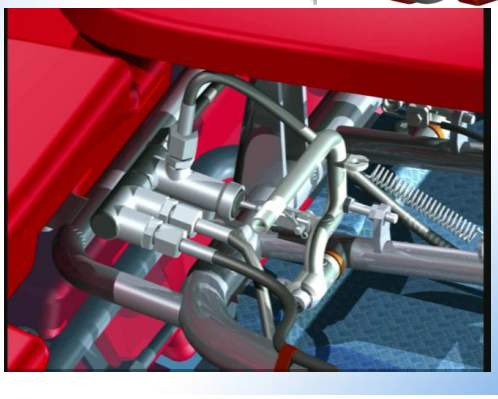




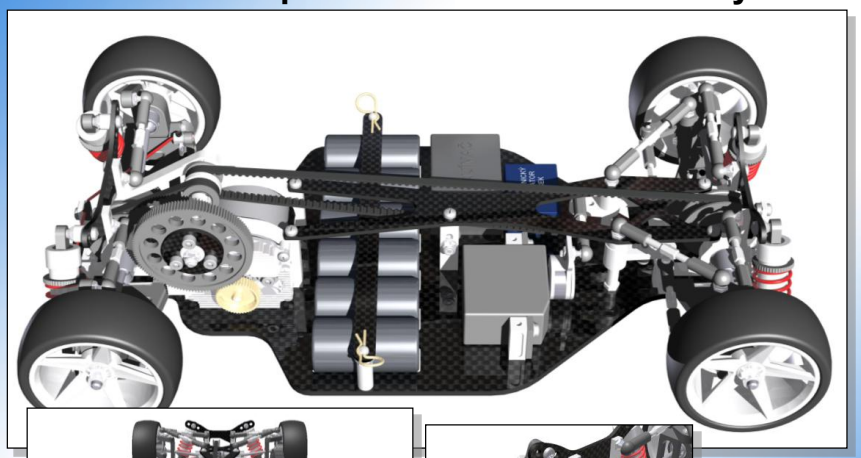
Brzdová soustava

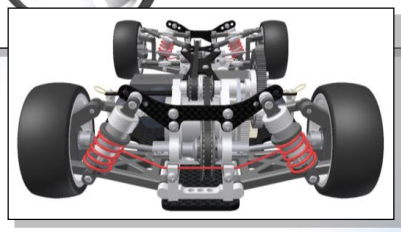
- Hydraulický jednookruhový systém
- Přední brzda – plovoucí třmen (jedna pohyblivá destička)
- Zadní brzda – čtyř válečková dvou čelisťová brzda
- Rozvod zajištěn hydraulickými hadicemi

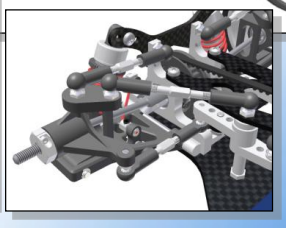




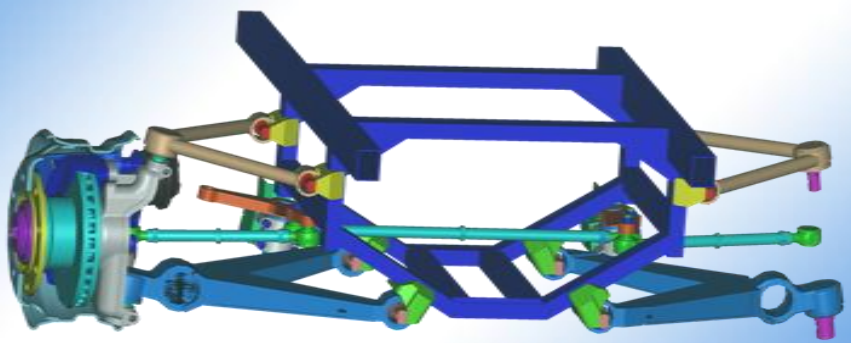
RC model podvozku závodní motokáry



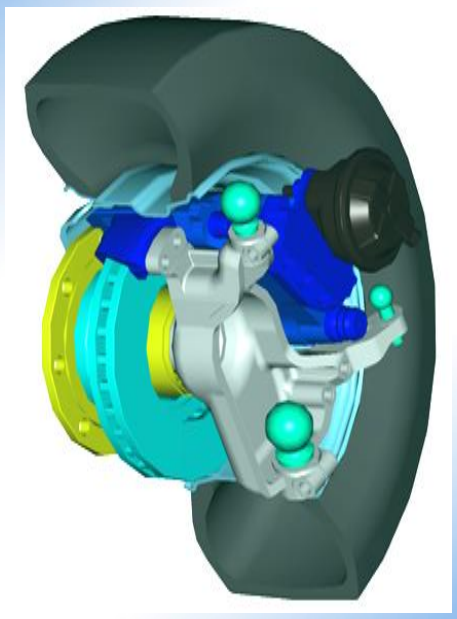




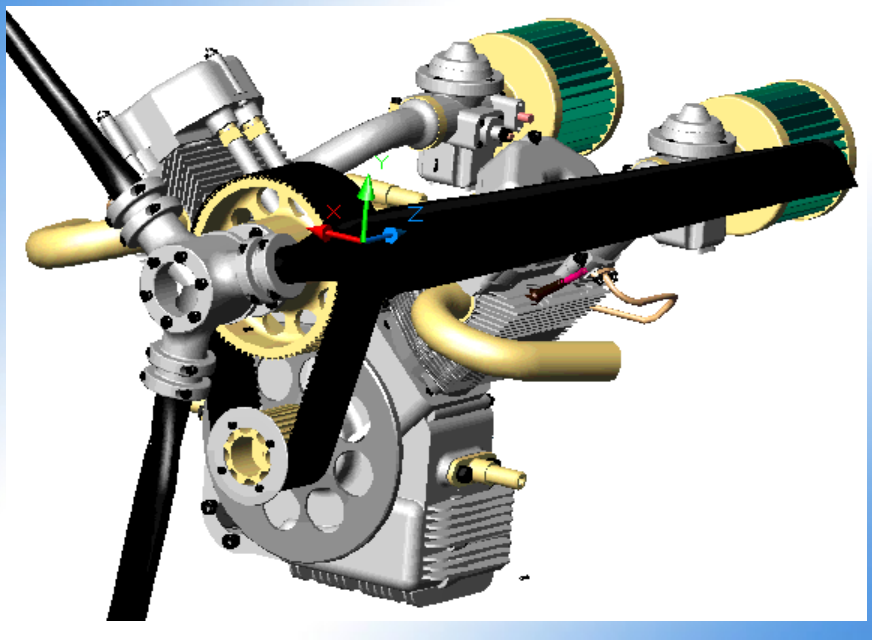
Návrh těhlice pro zabudování brzdy do přední nápravy autobusu SRO



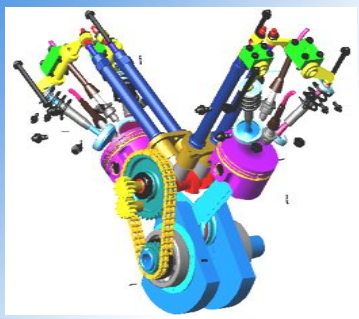
Detail náboje kola

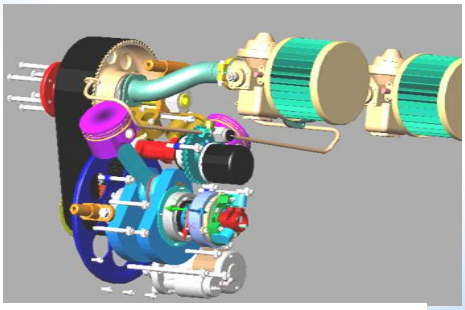


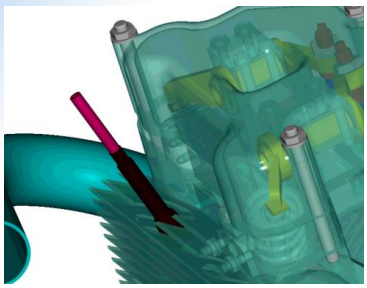
Prototyp leteckého motoru

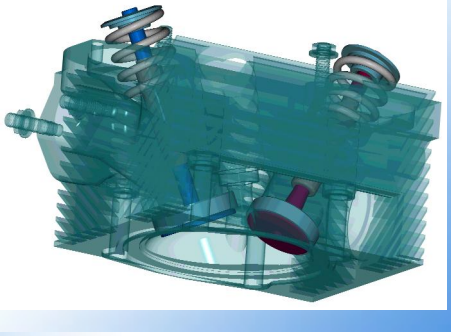


Detaily motoru

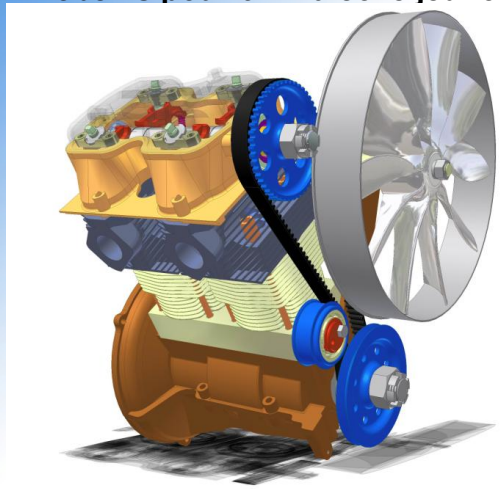






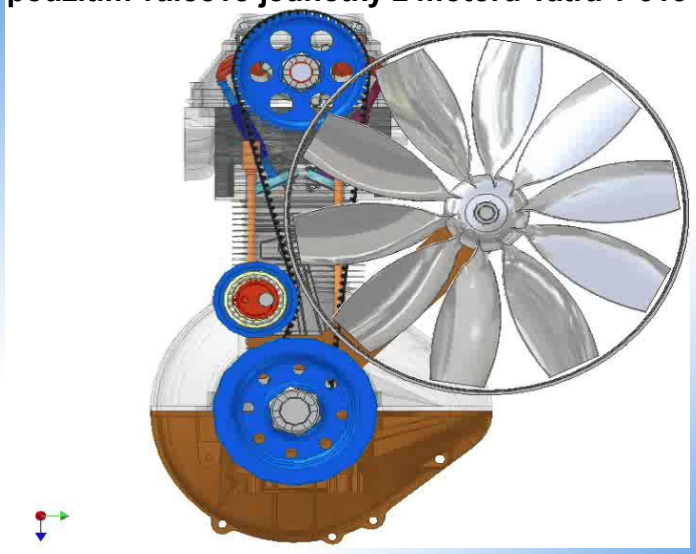


Přestavba motoru Trabant T601 na čtyřdobý pracovní oběh s použitím válcové jednotky z motoru Tatra T613





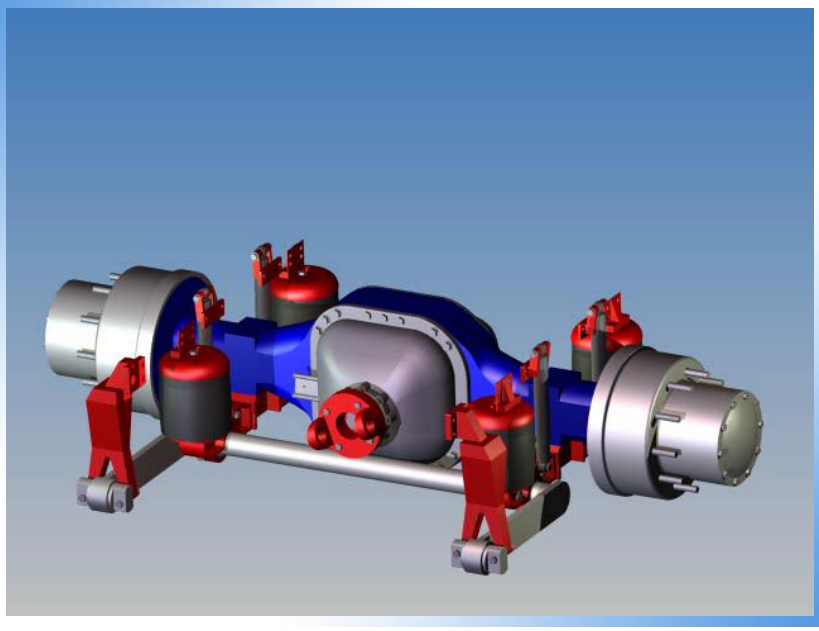
Přestavba motoru Trabant T601 na čtyřdobý pracovní oběh s použitím válcové jednotky z motoru Tatra T 613



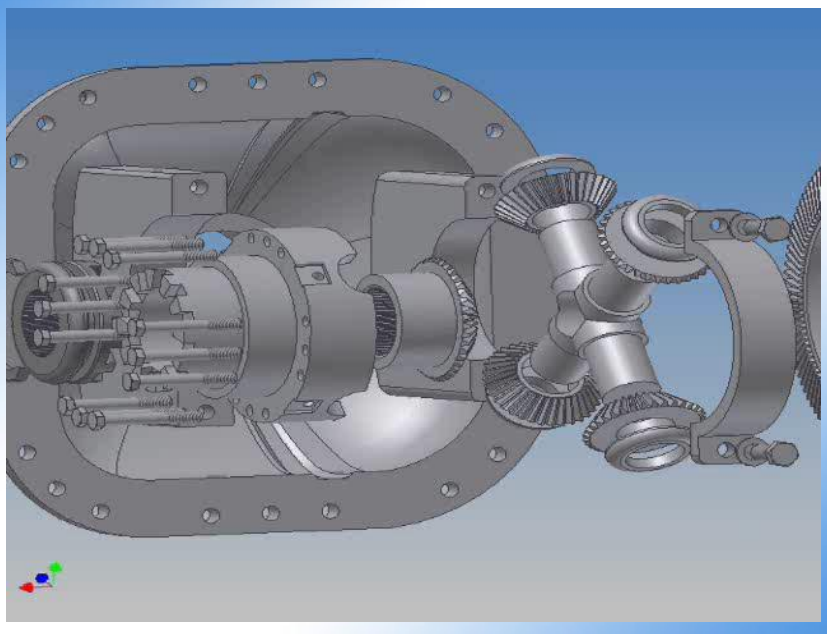
Návrh vznětového polomotoru – funkce



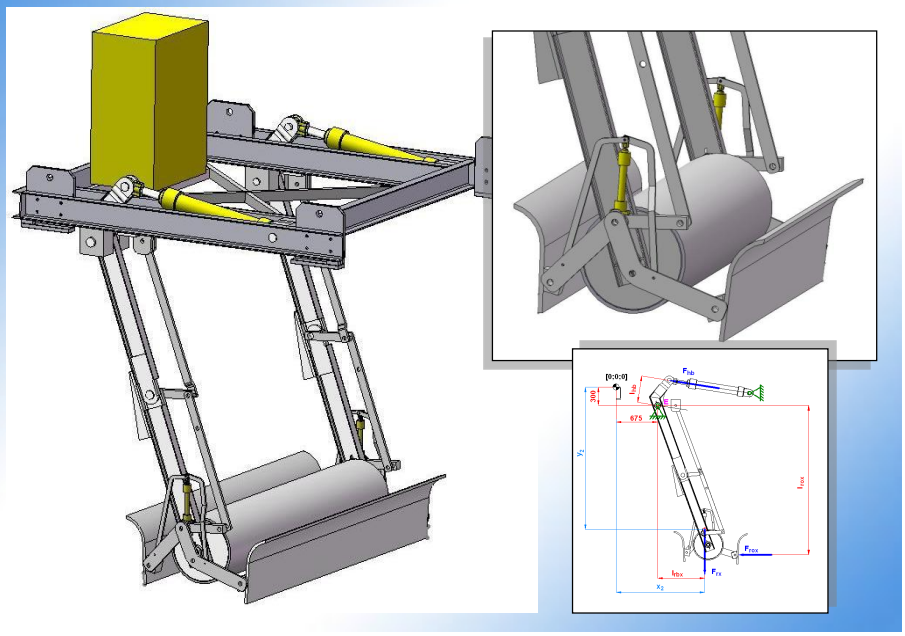
Návrh hnací nápravy nákladního automobilu-funkce



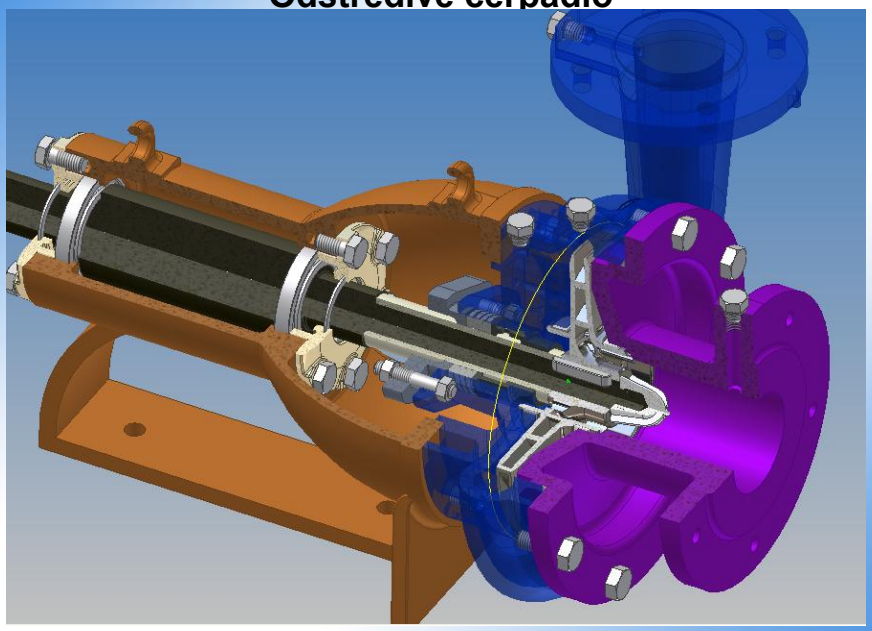
Návrh hnací nápravy nákladního automobilu- montáž

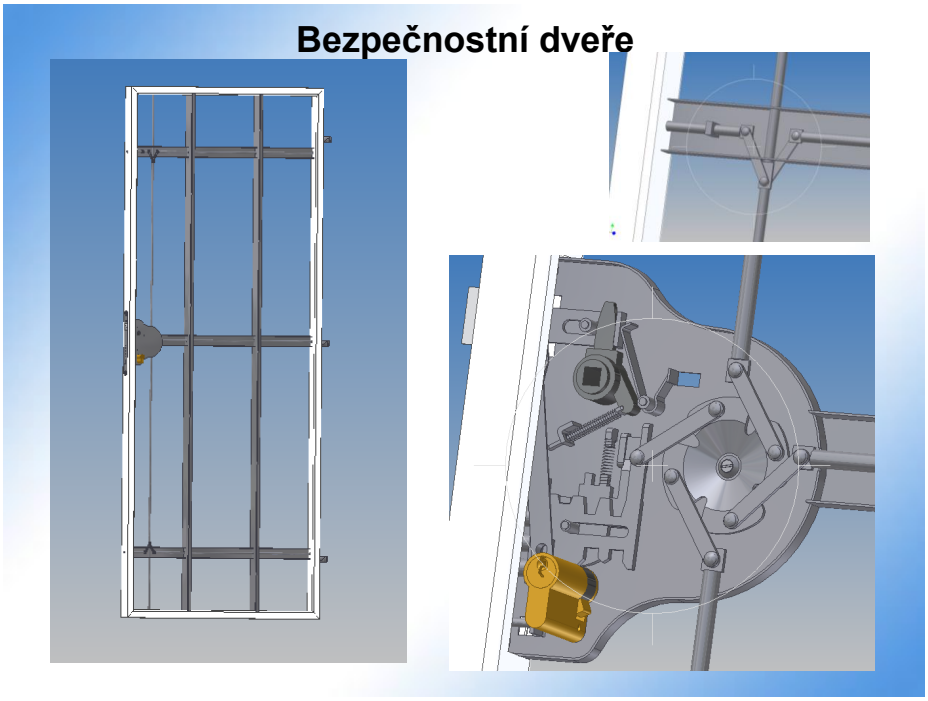


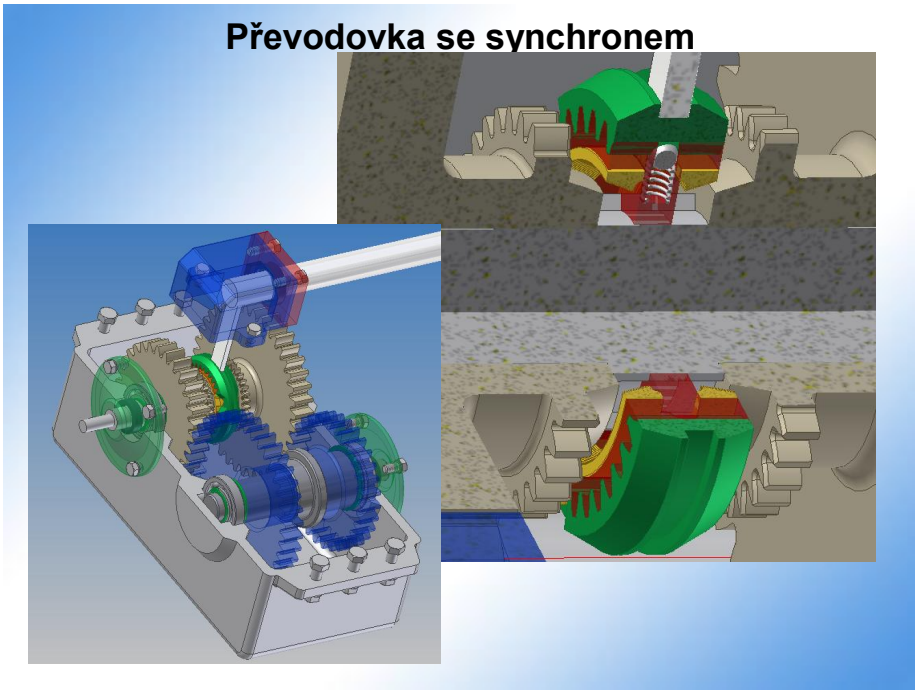
Rozhrnovací zařízení



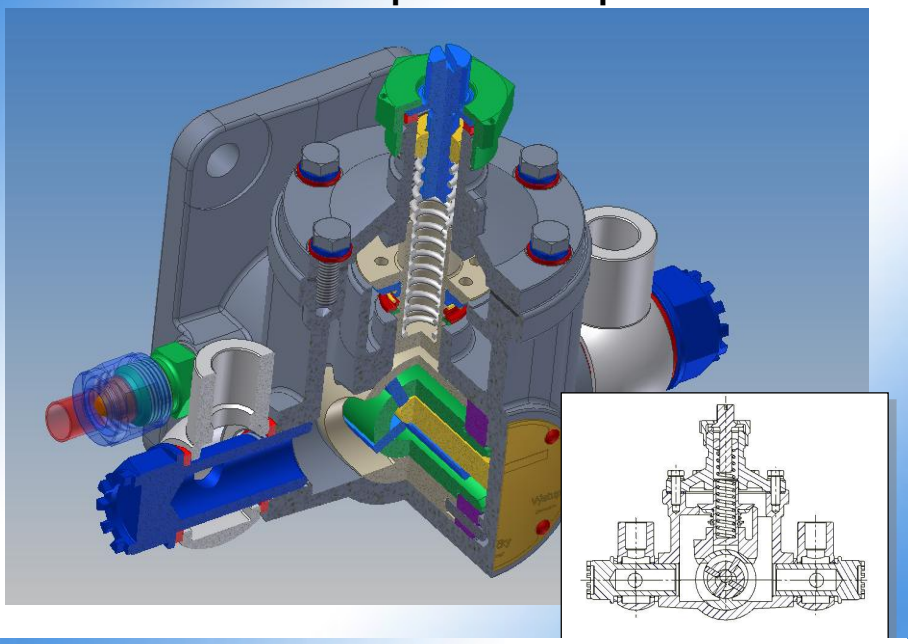
Odstředivé čerpadlo




Bezpečnostní dveře


Převodovka se synchronem

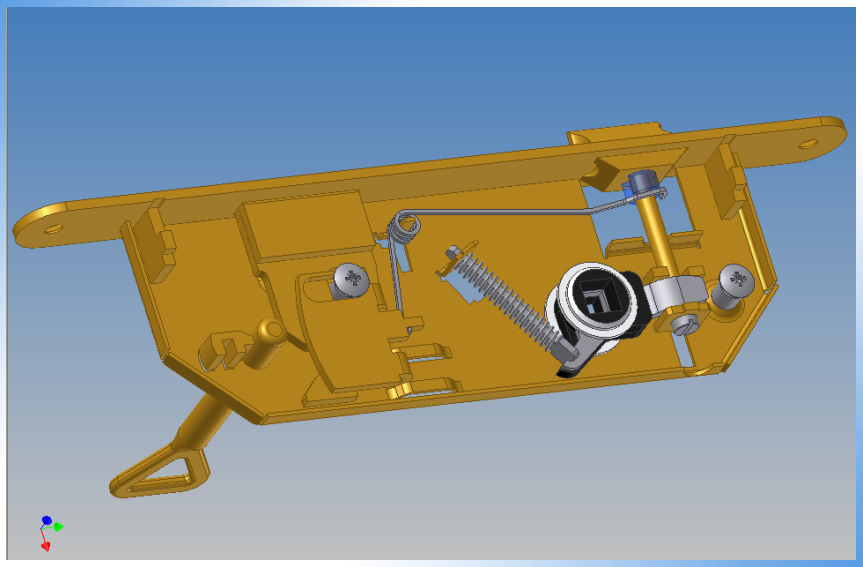
Nízkotlaké palivové čerpadlo



Pedál pro bubeníka



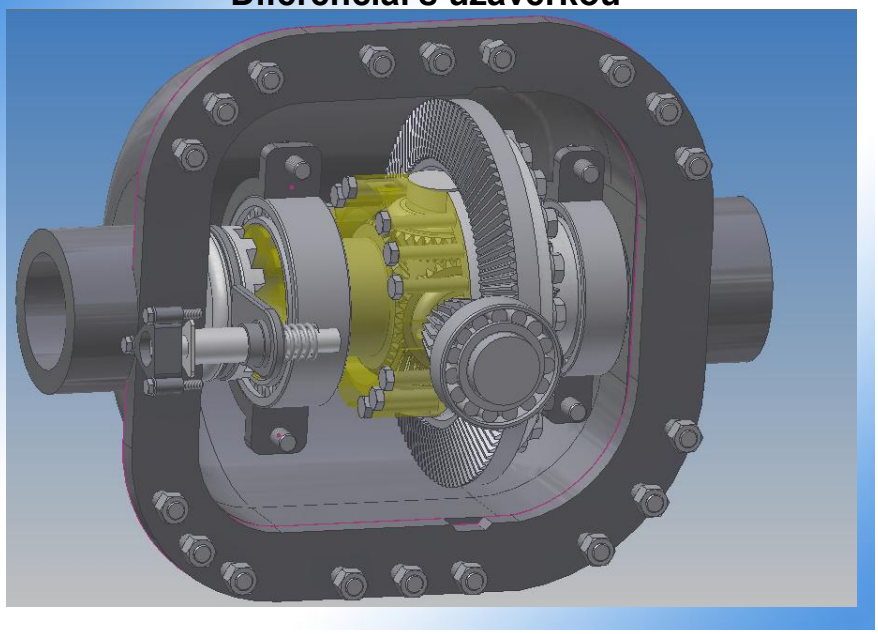
Zadlabávací zámek



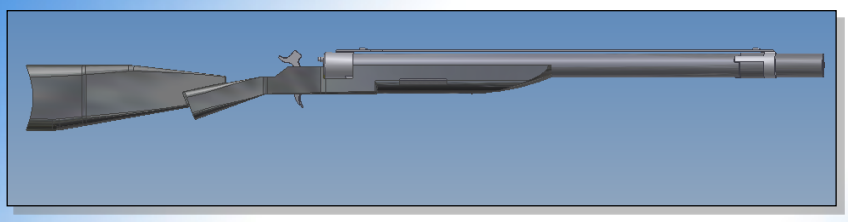
Pistole 9mm

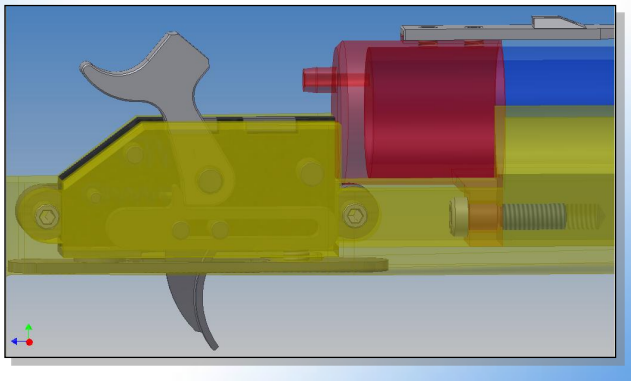


Diferenciál s uzávěrkou



Perkusní brokovnice

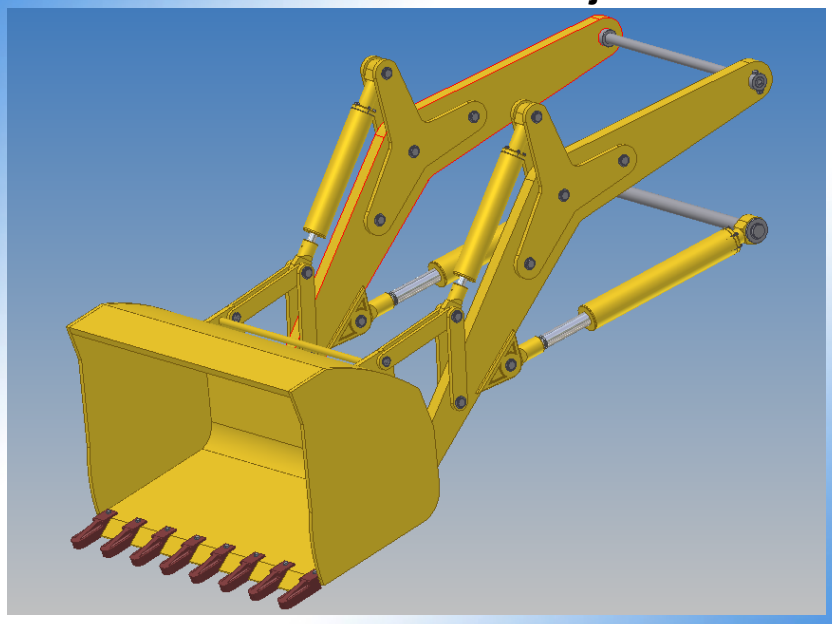




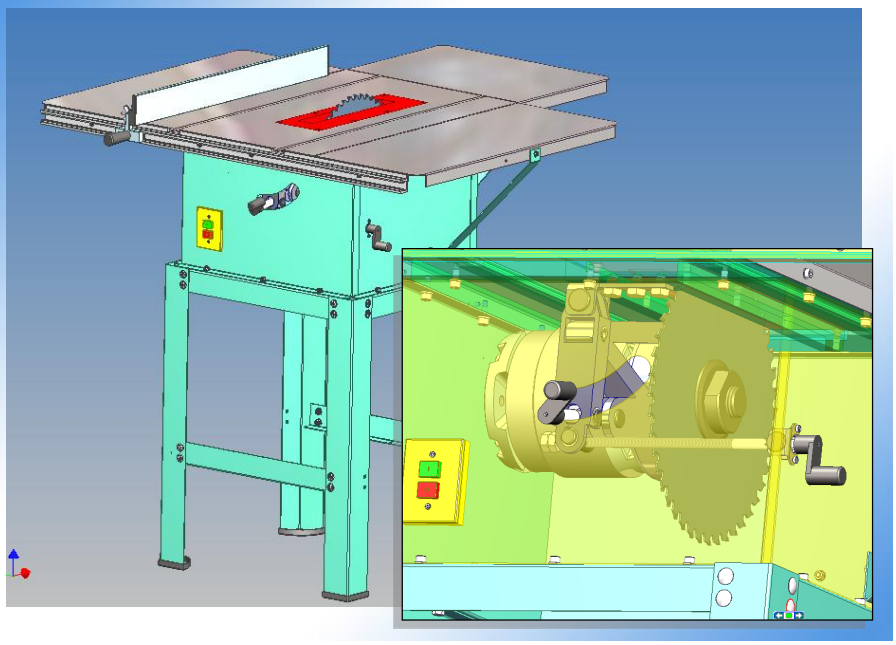
Stavebního stroj



Radlice stavebního stroje

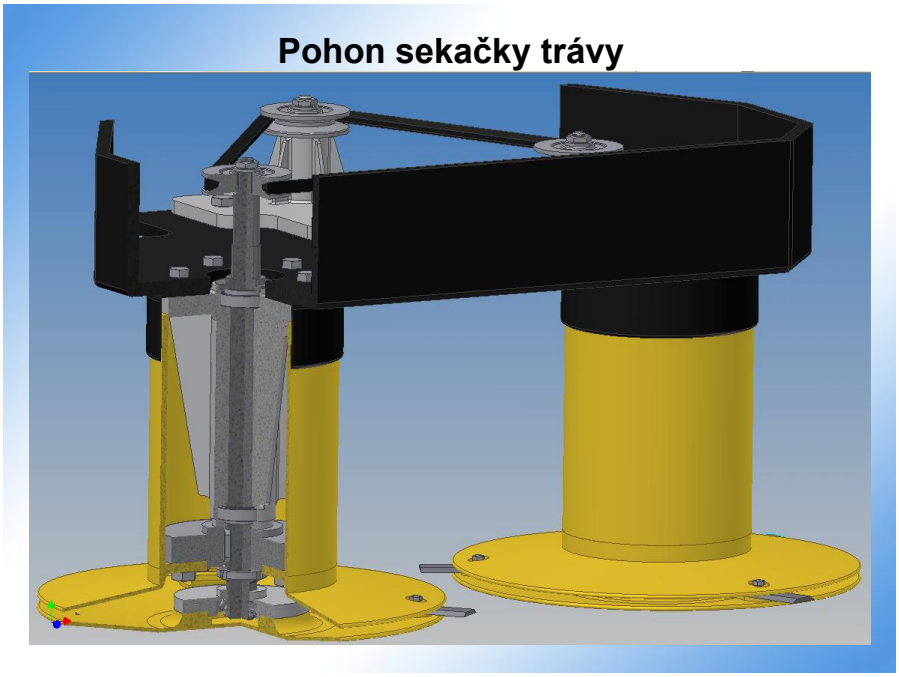


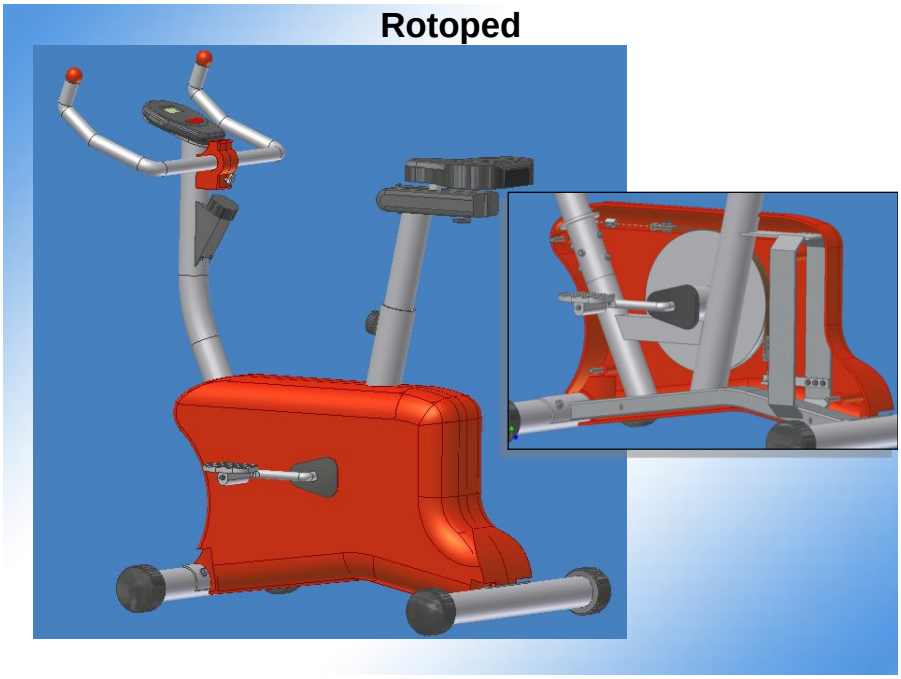
Nastavitelná stolařská cirkulárka



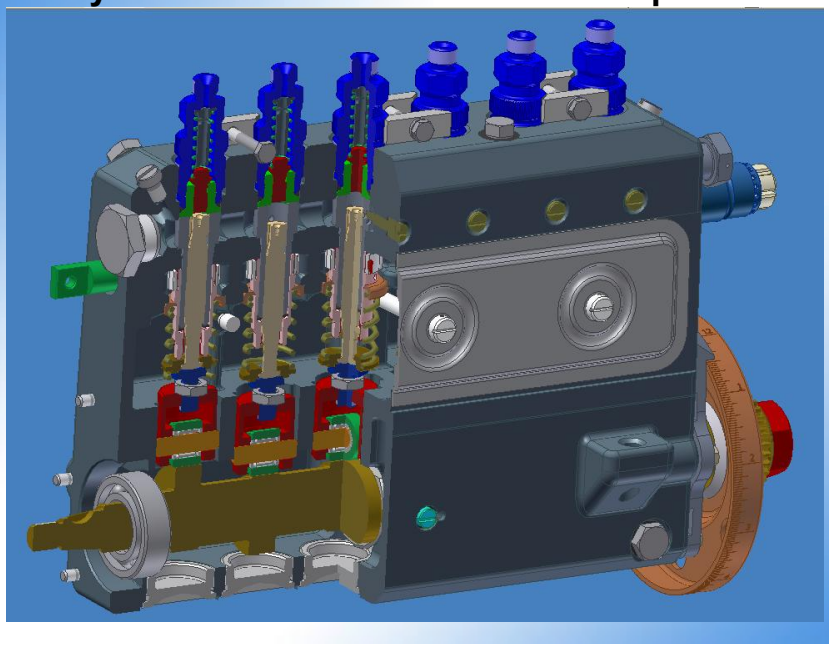
Šlapací autíčko



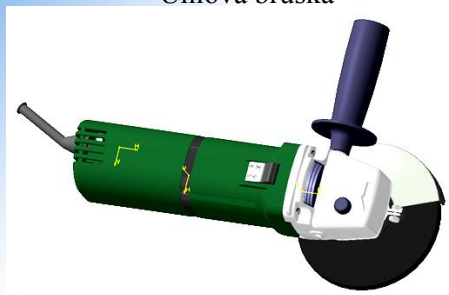

Pohon sekačky trávy

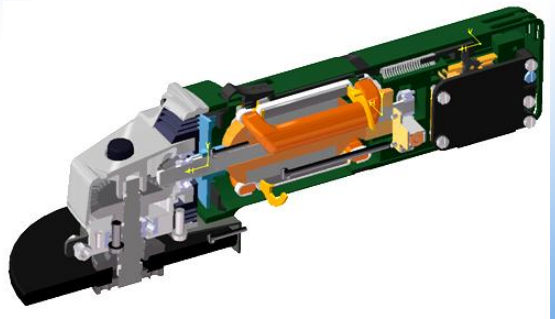

Rotoped

Vysokotlaké řadové vstřikovací čerpadlo



Úhlová bruska





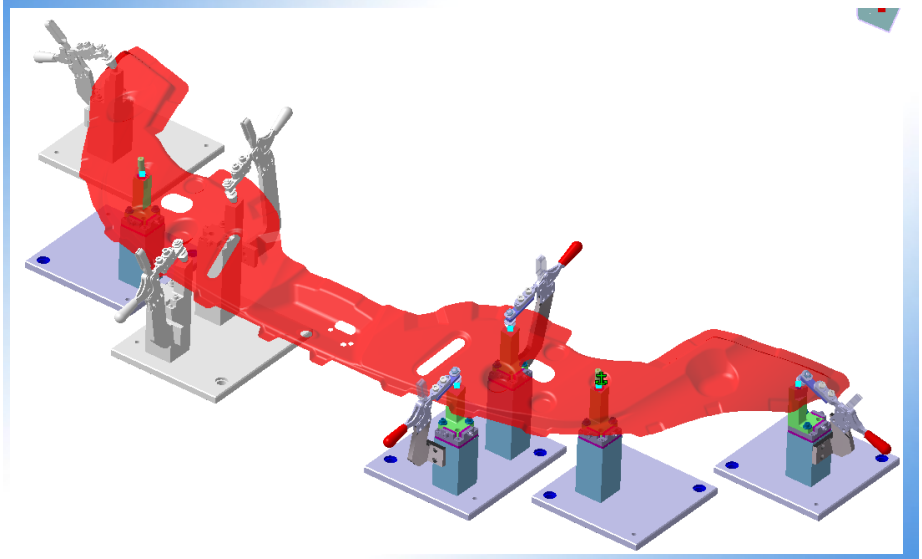
Návrh designu pákové baterie



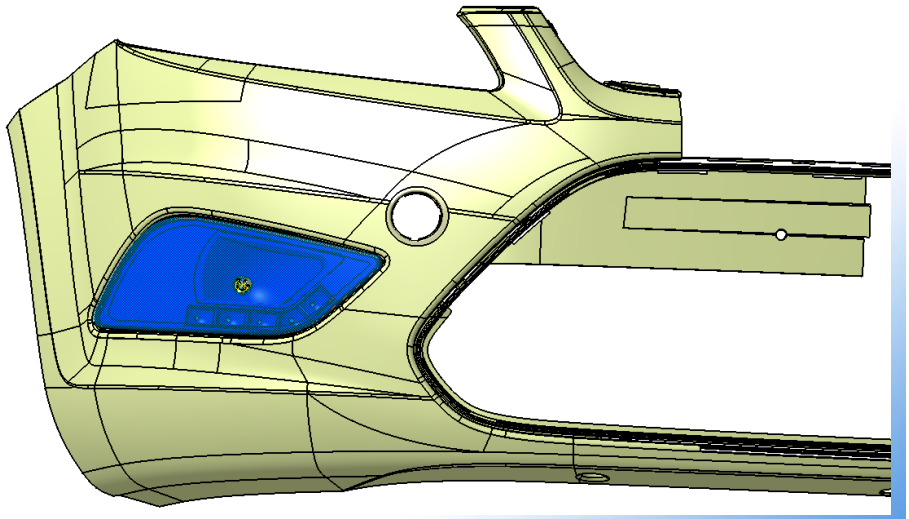
Jízdní kolo typu Downhill



Návrh konstrukce měřícího přípravku výlisku karoserie v SW Catia v5.



Vývoj mlhového světla s aplikací LED jako denní světlo



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

EFEKTIVNÍ VYUŽITÍ ENERGIE

Kolektiv třídy 3A

Vyšší odborná škola a střední průmyslová škola elektrotechnická Fr. Křižíka
Na Příkopě 16, Praha 1

Po hodinách společných diskusí, týmové práce i individuálně psaných příspěvků se čím dál zřetelněji ukazuje, že se téma úvahy trefilo do černého. Na hodně věcech se neshodneme a někde se i obyčejně míjíme, ale otázka efektivního využití energie zajímá každého z nás.

Fyzikové nám říkají, že energie je schopnost konat práci, že je obsažena v každé hmotě a že v této chvíli umíme získat jen její nepatrnou část. O tom, jak jsme se dostali do tohoto bodu, podávají svědectví dějiny naší civilizace, statistiky a prognózy vypovídají, že spotřeba energie stoupá a že tento trend nelze zastavit. V těchto souvislostech se pak stále více hovoří o kvalitách lidského života, o životním prostředí, o vědeckotechnickém rozvoji, i o kolapsu.

Jaká řešení připadají v úvahu?

Podle informací EIA (Energy Information Administration) využil svět v roce 1980 celkově 283 quadrilionů Btu energie, o dvacet let později už to bylo 400 quadrilionů Btu, v roce 2010 bychom se podle předpokládaného vývoje měli dostat na 511 quadrilionů Btu a v roce 2030 dokonce na 702 quadrilionů Btu spotřeby energie. Když vyjdeme ze stavu v roce 1980 a uvažujeme o předpovědi na rok 2030, pak by se v průběhu padesáti let zvýšila roční spotřeba energie téměř dvaapůlkrát!

Tento proces je ve stejném období provázen opačně směřující křivkou, která sleduje světové zásoby fosilních zdrojů – uhlí, ropy a zemního plynu. To je nebezpečné i proto, že tyto suroviny jsou potřebné nejen k výrobě elektrické nebo tepelné energie, ale také pro chemický průmysl.

Někteří z nás se domnívají, že jako můstek do doby, než bude vyvinuta řízená nukleární fúze, mohou sloužit uranové elektrárny. Jejich nebezpečnost je možná spíše zveličována. Radioaktivní uran se přece rozpadá sám, takže výhodnější je ho spálit. Tolik diskutované vyhořelé palivo není odpad, ale druhotná surovina. A řízená nukleární fúze? O řízené termonukleární reakci nebo o termojaderné syntéze se uvažuje už půl století – od chvíle, kdy byla shozena vodíková bomba. Omezenost surovinami by nebyla výrazná a bylo by i méně radioaktivního odpadu. Problém s dostatkem energie by byl vyřešen pro řadu generací.

Tady však mnozí z nás projevují obavu z toho, že bychom tuto cestu zvyšování energie nezvládli eticky. Lidé jsou různí a dělají chyby. Podařilo by se dostatečně ochránit takové systémy před touhou po moci, před zlobou, agresí, chamtivostí nebo náhodou?

Budoucnost většina třídy vidí spíše v přírodních obnovitelných zdrojích: slunci, vodě a větru.Vždyť jejich energie je základem života! (Koneckonců i energie z fosilních paliv pochází ze slunečních paprsků.) Podle dostupných informací máme dojem, že skutečně účinné využití těchto zdrojů je teprve před námi.Chystá se obrovský převrat v životě civilizace; solární panely, větrné mlýny či vodní kaskády jakoby byly jen jeho drobným předznamenáním.

Lidé se však zároveň obávají narušení ekologické rovnováhy, nedostatku financí, zahrávání si s ohněm. Takový projekt na využívání Golfského proudu by mohl vést k ochlazení severní Evropy, přerušení větrných proudů ochlazuje nebo naopak otepluje planetu, na solární panely nejsou peníze ...

A pak – nevedla by jakási jistota v zabezpečení energie lidstvo do zkázy? Kolikanásobně by se zvýšil objem odpadků, jejichž ukládání už dnes začíná měnit krajinu? Podle zvyšující se frekvence mikrotenových sáčků v korunách stromů u silnice si můžeme být téměř jisti, že se blížíme ke skládce. Sílící zápach z čerstvého pohoří už jen dál naši domněnku potvrzuje.

Podporu konzumu je cítit ze všech stran. Do mezireklamních šotů, kterým se pořád ještě říká televizní zpravodajství, se pomalu vejdou jen rozvádějící se modelky, fotbalisté či politici a hrdinou dne je zloděj, který připravil banku o půl miliardy. A jakoby dnešnímu světu ani tak nešlo o efektivní využívání energie, ale o to, aby se na energii dalo co nejvíce vydělat.

Na druhou stranu však samotná snaha uspět na trhu není nic špatného a možná může v určitém bodě vést i ke změně.

Když sledujeme statisticky znázorněný přehled vývoje spotřeby energie v rozmezí prvních tří desetiletí jednadvacátého století, okamžitě nás upoutá rozdíl mezi nárůstem spotřeby energie v hospodářsky vyspělých zemích a ve zbytku světa. Ještě v roce 2004 byla podle EIA vyšší spotřeba energie v hospodářsky vyspělých zemích. Rozdíl činil zhruba 25 quadrilionů Btu. Ale podle téhož grafu už v roce 2010 se výše spotřeby v zemích OECD začíná oproti ostatnímu světu pomalu snižovat (asi o pět quadrilionů Btu), a tento trend – podle prognózy - pokračuje stále větším rozdílem (v roce 2030 už by to mělo být téměř sto quadrilionů Btu).

Ekonomické tlaky, energetická závislost, vyčerpání zásob a životní prostředí nutí svět práce ke změnám přístupů a podněcují další vědeckotechnický rozvoj. Lídři světové energetiky pomáhají zákazníkům zvyšovat jejich operační výkonnost, zlepšovat spolehlivost sítí a zvyšovat produktivitu za současné úspory energie a snížení dopadu na životní prostředí.Podniky hledají cesty jak snížit ztráty v průběhu energetického řetězce a věnují pozornost kvalitě svých procesů.

A my navzdory mediálnímu humbuku a vlastní chtivosti, ale i v důsledku zdražování, přece jen začínáme uvažovat o šetření.

Šetřit se přece dá mnoha způsoby. Co kdybychom místo sedění u počítače nebo u televize raději vzali do ruky knížku nebo šli ven s přáteli? Co kdybychom zbytečně nesvítili přes den, a nebo jen v místnosti, kde se právě nacházíme? Proč necháváme řadu spotřebičů běžet naprázdno?

Efektivní využívání elektrické energie by se v celé republice zlepšilo, kdyby lidé byli řádně poučeni o stand-by režimech, nebo kdyby se tyto režimy přestaly využívat u elektronických spotřebičů. O stand-by režimech se začalo hovořit teprve nedávno a o jejich zákeřnosti toho lidé většinou moc nevědí. Aby ušetřili, někteří lidé na venkově pálí raději co jim přijde pod ruku ve starých kamnech, přitom ale mají doma televizi, video, rádio, set top box, počítač v úsporném režimu, mikrovlnnou troubu a podobně, o kterých si myslí, že jsou vypnuté, a ty je zatím ve stand-by režimu připravují každou minutu o peníze.

V mnoha domácnostech je běžné, že jsou radiátory prakticky zakryté, často z tzv. estetických důvodů, teplo uniká nedostatečně utěsněnými okny a, i když se dnes začínají zateplovat i „paneláky“, pořád je ještě většina domů bez tepelné izolace.

Efektivní využívání energie nemusí být něco kolosálního. Když si člověk víc všímá, potěší ho i obyčejná tlačítka na dveřích v metru: Dveře se otevřou, jen když je skutečně třeba a někdo jimi chce projít.

Moderní technologie spojené s efektivním využitím energie pronikají i do automobilového průmyslu. Tzv. hybridní motory využívají spalovacího a elektrického motoru najednou, a tak se zvyšuje účinnost. Známá je snaha zavést motory na vodíkový pohon, na auta se dokonce instalují solární panely. Uvidíme, co všechno ještě přijde. Přestože se ozývá hlas, že místo autem můžeme jezdit na kole, v zimě se doma více oblékat, aby se nemuselo příliš topit, a v noci si svítit svíčkami, abychom nespotřebovávali elektrickou energii, je jasné, že jde o provokaci. Téměř nikdo se nechce vzdát svého pohodlí. Stačí, když budou nové vynálezy při nižší spotřebě energie podávat mnohem větší výkony.

Mysleme tedy na úspěch, ne na katastrofu.

Jestliže platí, že energie je obsažena v každé hmotě,že jeden kilogram hmoty v sobě soustřeďuje asi dvacet pět miliard kilowatthodin energie a že ji neumíme získat, a když ano, tak jen nepatrnou část, potom svět a život je tajemství.

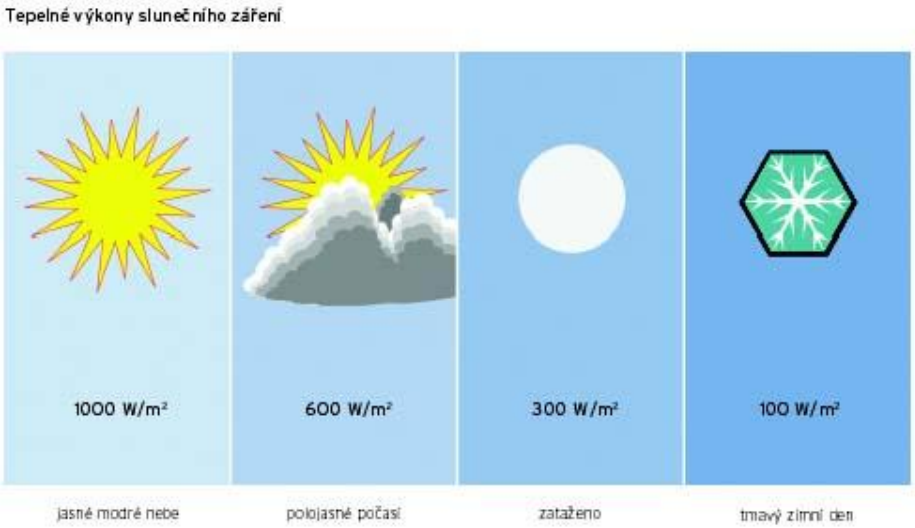
Získávání energie, její efektivní využití a spotřebu chápeme samozřejmě v souvislostech našeho života. Nutně se pak musíme ptát, kde se bere lidská energie a jak ji efektivně využít, abychom si to všechno nepokazili. Nabízíme ve zkratce příběh, který popsal jeden náš spolužák:

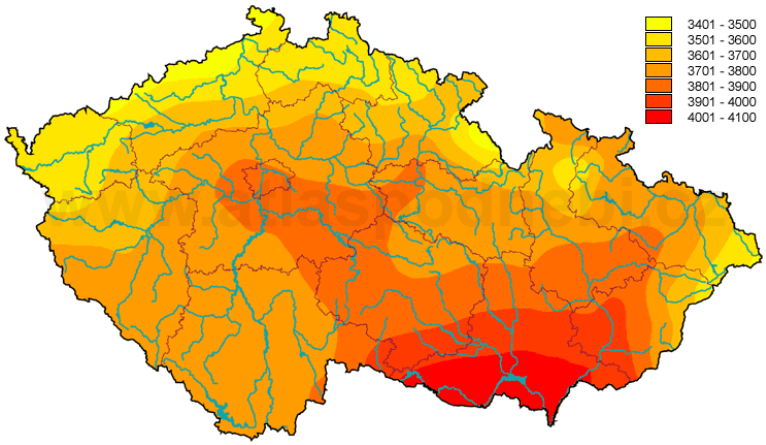
„Naše kočka onemocněla po aplikaci kortikoidové injekce, která vyvolala u zvířete cukrovku. Zamýšlel jsem se nad tím, co mohu udělat, abych pomohl. Uvědomil jsem si, že pokud se dá předat energie z člověka na člověka, že by mohlo jít předat energii i z člověka na zvíře. Nejprve jsem část psychické energie věnoval nastudování příslušné literatury, abych se dozvěděl něco bližšího o cukrovce. Fyzickou energii jsem věnoval přepravě zvířete na veterinární kliniku, kde probíhala léčba infuzemi, protože zvíře nežralo a nepilo. Seděl jsem tam s kočkou a s někým z rodiny a snažil se rozdělit energii mezi zvíře a blízkého člověka. Měl jsem snahu předat svou energii, aby došlo u zvířete ke zlepšení zdravotního stavu a u blízkého člověka k překonání této složité situace. Ale na mysli mi vyvstala otázka: „Kolik energie mám, abych mohl pomoci a měl dost síly poprat se sám s tímto problémem?“ Vždy, když jsme dorazili domů, cítil jsem se značně vyčerpaně a dobíjel jsem energii u počítačové hry, která mi pomáhala na chvíli zapomenout na realitu. Stále jsem si kladl otázku: „Existuje přenos lidské energie?“ Ale protože jsem byl svědkem toho, jak mamka přenášením energie pomáhala kamarádce překonat krizovou životní situaci, věřil jsem a věřím, že něco takového

je možné. Snad se mi podařilo naší kočce předat alespoň nějakou energii, protože nastalo malé zlepšení. Nevím, nakolik se uzdraví, ale budu se snažit jí pomoci. Ale jak můžu dočerpávat energii? Mám dostatek zdrojů? Domnívám se, že u každého člověka to funguje jinak. Já dočerpávám fyzickou energii cvičením a psychickou energii poslechem hudby a počítačovou hrou. Pořád přemýšlím o tom, jestli je předávání energie reálné. Může lidstvo a příroda sobě takto pomáhat? Domnívám se, že kdyby v každém bylo něco pozitivního, co by předal alespoň jednomu živému tvorovi, nemohlo by být na světě tolik násilí a smutku."

Možná, že čtenáře této úvahy překvapí, že hlavním předmětem zájmu se v závěrečné fázi našich diskusí stala energie lidská. Nejsme samurajové a neumíme efektivně využít každý kousek energie, která se skrývá v našem těle, nemáme ani jejich vytrénovanou mysl. Docela obyčejně, ale zároveň vážně, jsme mluvili o potřebě odpočinku, o hospodárnosti a trpělivosti, i o síle úsměvu. A také jsme se ptali, zda i pro nás platí, že v jednom kilogramu našeho těla je soustředěno dvacet pět miliard kilowatthodin energie?

Tabulka č. 1: Hodnoty výkonu dopadajících paprsků na zem, za různého počasí





Obr. 1: Dopadající energie(v Joulech) na 1 m^2 plochy zemského povrchu v ČR



Obr. 2: Takto vypadají nyní instalované fotovoltaické panely na střechách budov.

Středoškolská technika 2008

**Setkání a prezentace prací
středoškolských studentů na ČVUT**

MĚŘENÍ SPOTŘEBY ELEKTRICKÉ ENERGIE ODEBÍRANÉ POČÍTAČEM

Jakub Doubrava, Marek Růžička, Tomáš Luňák

Vyšší odborná a Střední průmyslová škola elektrotechnická Fr. Křižíka
Na Příkopě 16, Praha 1

Úvod

Při prohlížení různých internetových stránek jsme narazili na plno odporujících se informací o tom, jaká je spotřeba počítačového zdroje, nebo celé počítačové sestavy. Zajímalo nás, jestli dokážeme například u 550 W zdroje využít jeho plnou funkčnost, nebo jestli je tato hodnota nevyužitá. Zkoumali jsme počítače při různých stavech a měřili je jak vypnuté, tak i v úsporném režimu,při startu operačního systému Windows, nebo při velkém zatížení různými programy. Aby jsme mohli porovnávat výsledky, měřili jsme 3 různé počítače, z nichž jeden byl staršího modelu, druhý o několik let novější a třetí byl obyčejný Notebook.

Při tomto měření jsme objasnili mnoho nepravd a fám o tom, jaká je doopravdy spotřeba počítače a pomůže vám možná objasnit to, že výběr zdroje není o tom, aby za jeho názvem bylo co největší číslo. Dokonce i vysoce výkonné počítače mají při plném zatížení spotřebu maximálně 250 W.

Cíl projektu

Cílem projektu bylo, ověřit nakolik údaje o parametrech – tj. přechodovém ději při zapnutí, náběhu operačního systému a provozu vybraných programů odpovídají údajům o spotřebě a časových průbězích napětí a proudu uváděným v literatuře a na internetu. Projekt jsme rozčlenili do několika částí:

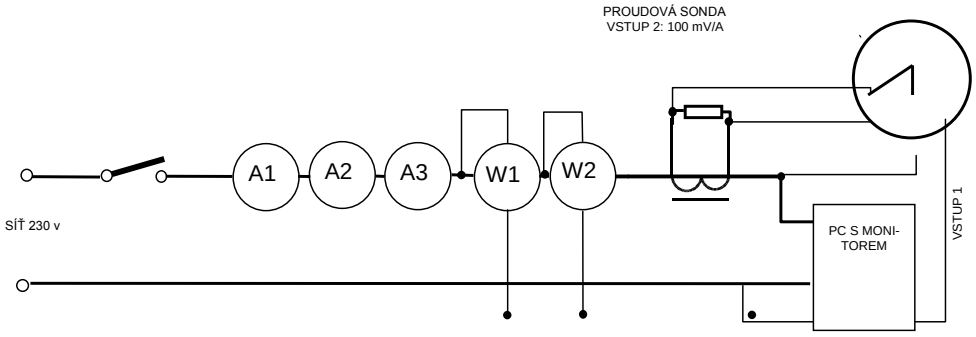
- Přechodový děj při zapnutí počítače se zjištěním proudového nárazu a poklesu napětí
- Výkonové parametry při rozběhu operačního systému
- Nároky na zdroj při chodu typických programů a výběr nejnáročnějšího programu z hlediska zatížení zdroje
- Parametry úsporného režimu
- Nároky napájecího systému notebooku
- Ekonomické vyhodnocení vhodného režimu provozu počítačů

Dále nás také zajímala ekonomická stránka spotřeby počítače a počítali jsme kolik stojí provoz jednoho počítače, nebo i celé počítačové učebny.

Použité přístroje a postup měření

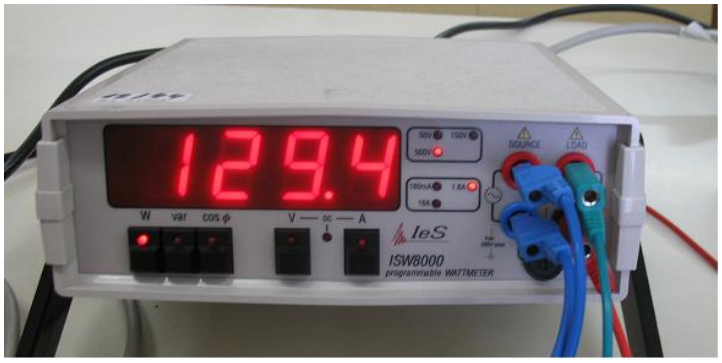
Během tohoto měření jsme využívali prostor i vybavení laboratoří na naší škole, a za odborného dohledu profesorů Ing.Jaroslava Sládečka, CSc. a Ing. Richarda Poula jsme byli vedeni k správnému zapojení a následovanému měření všech tří počítačů.

Za pomoci přístrojů, které nám byly zapůjčeny jsme provedli měření a objasnili neznámá fakta.
Použili jsme tyto přístroje (viz následující schéma):





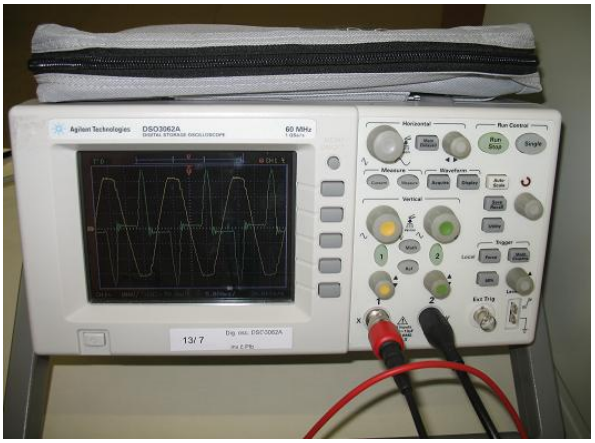
A1 – Ampérmetr Metrix MX035C feromagnetický systém; s usměrňovačem; č.uložení: 11/7 izolační pevnost 2; inventární číslo: PlbIII/23



W1 – Digitální Wattmetr ISW8000 ; výrobce IeS 230 V ~ 50 Hz
Rozsahy: 50 , 150 , 500 V: 160mA, 1,6A, 16A



A2 – Ampérmetr Metra magnetoelektrický systém; rozsahy: 0,2; 1; 5 A ; č.uložení: 11/1; střídavý proud; izolační pevnost 2



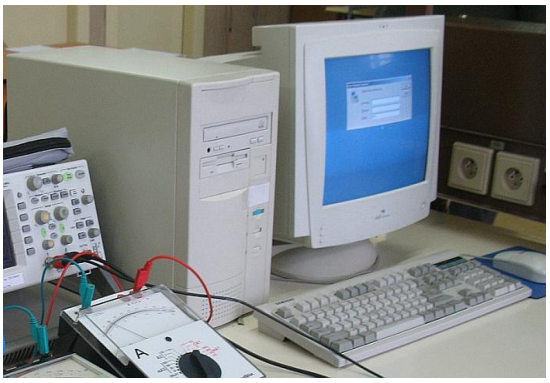
Osciloskop Agilent Technologies DSO3062 Digital Storage Oscilloscope 60MHz Inv.Č 13\7

Pro všechna měření bylo použito v zásadě stejného schéma s tím, že nebyly použity vždy všechny přístroje.

Měření bylo provedeno na 3 kusech PC:

PC1:

CPU: 300 MHz
RAM: 64MB Kingston
HDD: 2,5 Gb
Zdroj: Kingston KC-250W
Vstup: 115/230V 4/2A 50-60Hz
Dc výstup: 5V; 12V; -5V; -12V
25A; 10A; 0,5A; 0,5A
Grafická karta: S3 ViRGE-DX\GX PCI
Zvuková karta: Integrovaná
OS: Windows 98
LAN: 10-100MB/s
CD-Rom: ATAPI 52 max
Monitor: ADI Micro scan 15´´



PC2

CPU:Pentium Model 3 2806 Mhz
RAM:256Mhz
HDD:60 Gb
Zdroj:EC model 300X1(PFC)
S/N D01152512 ATX 2-03
Vstup:230V~ ; 3,5A ; 50Hz
Dc výstup:
+3,3V; +5V; +12V; -5V; -12V; -5V SB
14A; 30A 10A; 0,5A; 0,8A; 2A
Grafická karta:Gigabyte Radeon 9200
Zvuková karta:Realtek AC´97 audio
OS:Windows XP Professional
LAN: 10-100MB/s
Monitor:Yusmart 198 YP-A 19´´



Notebook:

Acer TravelMate 2203LC
CPU:Intel Celeron D335
2,8GHz; 256kB L2 cache,
533MHz FSB
RAM:256MB DDR
HDD:40GB
Základní deska:
Zdroj:100-240V~ ; 50-60Hz ;
1,5A
Liteon 135W ; Output 19V=7,1A
Model DA-1131-08
Grafická karta:ATI mobility
radeon 9000 IGP
Zvuková karta:
OS: Windows XP Profesional
LAN: 10-100MB/s
CD-Rom:DVD/CD -RW Combo
Monitor:15" XGA TFT LCD



Přechodový děj při zapnutí počítače se zjištěním proudového nárazu a poklesu napětí

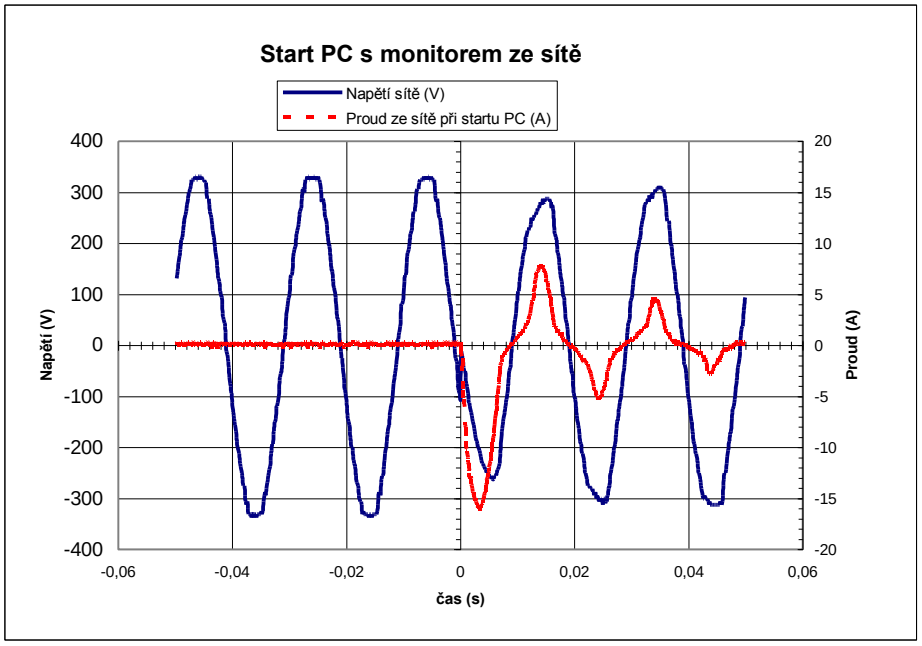
Ke zkoumání přechodového jevu při zapnutí počítače nás přivedla vlastní zkušenost z učeben výpočetní techniky. Jestliže zapneme současně 15 nebo i více počítačů v učebně tak se nám většinou podaří způsobit výpadek jističe učebny. Není to jenom naše zkušenost, jev je známý všude kde je větší množství počítačů. Z vyprávění o zkušenostech s tímto jevem od pracovníků v bankovnictví. zdravotnictví apod… jsme zjistili že touto problematikou se téměř nikdo seriózně nezabývá a je často používáno svérázných řešení.

Protože jsme budoucí elektrotechnici, víme že správné dimenzování jištění rozvodů tento problém může redukovat, nebo i úspěšně odstranit. Potřebujeme pro něj však nějaké podklady. Ke shromáždění podkladů bude sloužit i tento projekt. Samotná metodika projektování jištění není náplní této práce.

Postup měření přechodového jevu

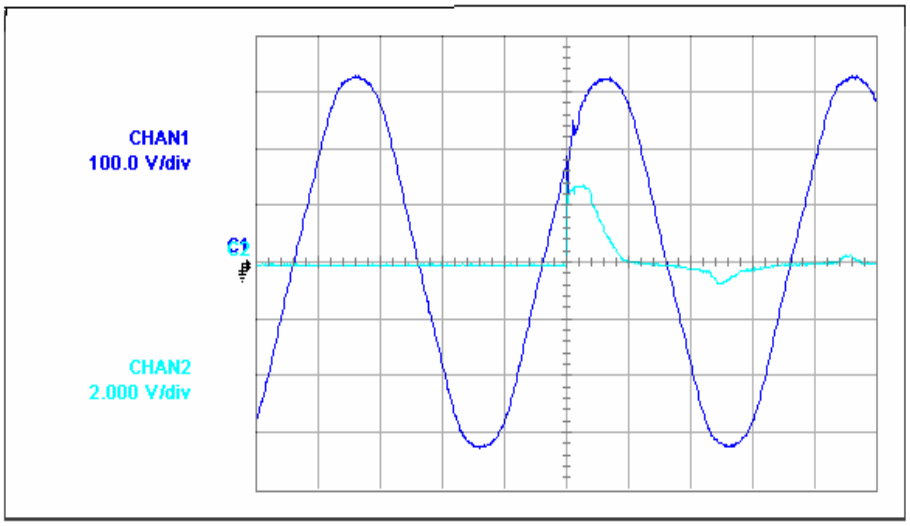
Zapojili jsme přístroje podle schématu a nastavili digitální osciloskop, propojený komunikační linkou se sériovým rozhraním USB na počítač pro sběr dat. K ovládání nám sloužil program HP scope dodávaný výrobcem osciloskopu. Snímané průběhy bylo možno ukládat ve dvou formátech. Tabulka zpracovatelná v tabulkovém kalkulátoru(Excel). Druhou možností je bitmapová kopie obrazovky osciloskopu která se dá použít pouze jako obrázek. Obrázek pořízený metodou bitmapové kopie obrazovky má lepší grafickou

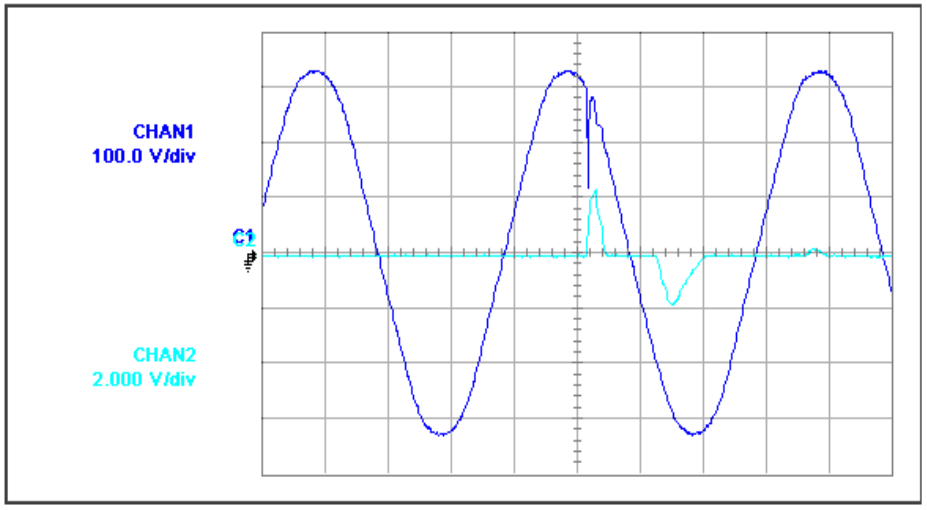
podobu která zjednodušuje odečítání hodnot, metoda tabulkového kalkulátoru je vhodnější pro přesnější a podrobnější měření a sledování složitějších průběhů. Počítače jsme postupně uváděli do stavů pro které jsme měření chtěli provést a snímali data v obou dostupných metodách. Přechodový jev u notebooku jsme nemohli zkoumat, protože pracuje z baterie která je podle toho jak moc je nabitá nebo vybitá podstatně určujícím článkem pro determinaci parametrů přechodového jevu zapnutí-start jejího dobíjení.



Výsledky měření přechodového jevu:

Na obrázku je časový průběh přechodového jevu zapnutí počítače PC1.Z průběhu je zřejmé, že jev zapnutí počítače odpovídá přechodovému ději nabíjení kondenzátoru. Špička proudu snímaná proudovou Hallovou sondou je 16A má proudová špička amplitudu 1,8A. Kromě vlivu jevu nabíjení kondenzátoru se může projevit u klasického monitoru i pomocný demagnetizační puls. To se však u našeho měření vůbec neprojevilo.





Další dva průběhy přechodových dějů na síť jsou získány měřením na počítači č. 2.
U obou je patrný obdobný časový průběh proudu.

Měření bylo provedeno na měkčí síti takže začátek nabíjení kondenzátoru odpovídá skokovému úbytku napětí napájecí sítě.
Špička proudu dosahuje hodnoty až 24A (proudová sonda má konstantu 100mV/A.)

Výkonové parametry při rozběhu operačního systému

Následující údaje byli získány v energeticky nejnáročnějším režimu při rozběhu operačního systému. Obdobných výsledků jsme dosáhli i při poslechu hudby přes program Windows Media Player a také při chodu testu antivirového programu AVG.

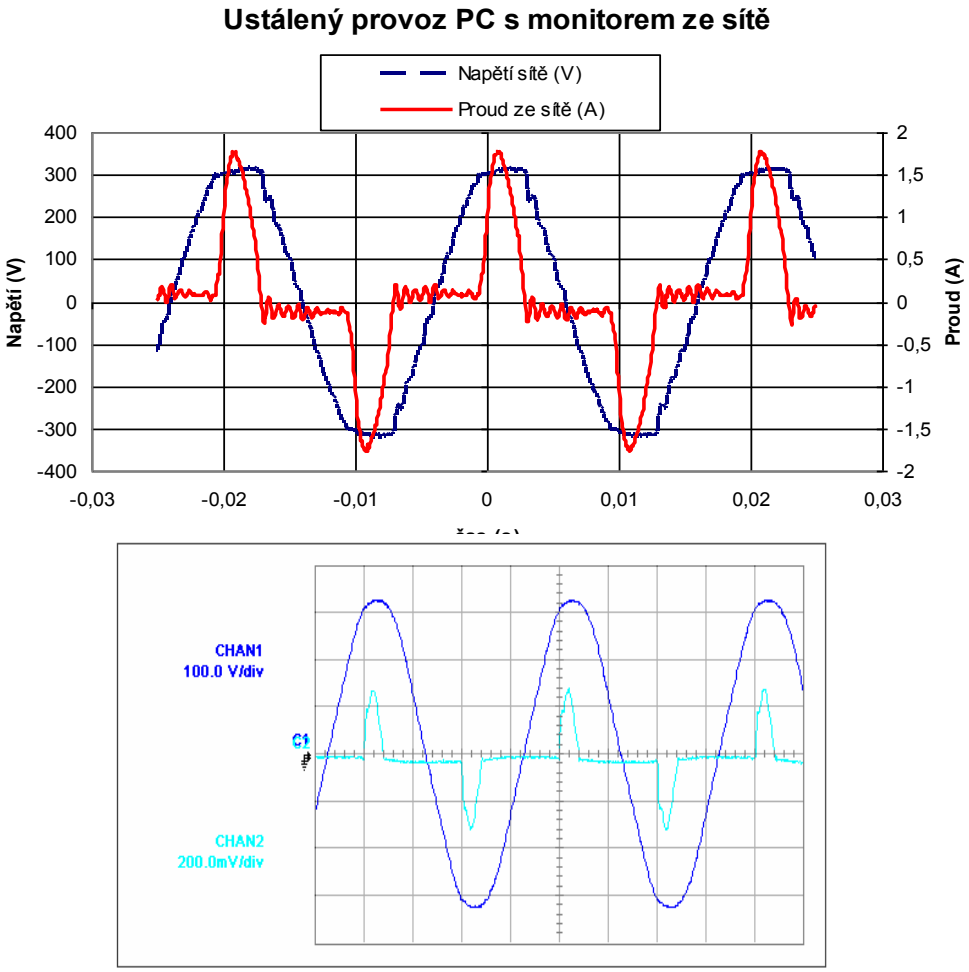
	P (W)	**Q (var)**	**cos φ**	**U (V)**	**I1 (A)**
Pc1	130	-260	-	234	0,97
Pc2	127,8	2	0,71	227,2	0,689
Notebook	62	-19,5	0,71	234,3	0,297

Údaje uvedené v tabulce pocházejí z digitálního wattmetru.
Pro porovnání vhodnosti použití ampérmetru bylo provedeno srovnání jejich údajů které je uvedené v následující tabulce

PC1 s monitorem ADI MICROSCAN		
	U (V)	230
Digitální Wattmetr ISW 8001 [W2]	*P* (W)	128
Elektrodynamický wattmetr METRA [W1]	*P* (W)	140
Ampérmetr METRIX [A1]	*I* (A)	0,54
Elektromagnetický A METRA [A2]	*I* (A)	0,83
Elektromagnetický A METRA starší typ [A3]	*I* (A)	0,90
Zdánlivý výkon z 0,83 A [údaj A2]	*S* (VA)	190,90
Zdánlivý výkon z 0,90 A [údaj A3]	*S* (VA)	207,00
Účiník z 0,83 A [údaj A2]	□ □ = *P/S*	0,73
Účiník z 0,90 A [údaj A3]	□ □ = *P/S*	0,68

Je patrné že magnetoelektrický přístroj METRIX ukazuje podstatně odlišnou hodnotu proudu což je způsobeno tím že ve skutečnosti měří střední a nikoliv efektivní hodnotu.

Digitální wattmetr a elektrodynamický měří přibližně stejně, chyba měření je však patě dost velká. K tabulce přísluší následující časové průběhy.



První průběh platí pro počítač č. 1, druhý z průběhů pro PC č. 2.

Zajímavé je že ačkoliv u počítače č. 2 je deklarováno aktivní kompenzaci účiníku (PFC) na obrázku se to nijak neprojevuje. Podle údajů z literatury to může být způsobeno malým

zatížením zdroje kde tato kompenzace ještě nefunguje. Udává se hodnota prahu činnosti 50 až 60% jmenovitého výkonu, které zde nedosahujeme.

U notebooku jsme průběh pouze monitorovali. Časový průběh vypadá stejně, ale velikost proudu je silně závislá na stavu nabití baterie. V podstatě byli nejvyšší hodnoty stejné jako při režimu nabíjení baterie při vypnutém notebooku, což bude uvedeno dále.

Parametry úsporného režimu

	P (W)	Q (var)	cos φ	U (V)	I1 (A)
PC1	-	-	-	-	-
PC2	8,6	-13,58	0,29	228	0,079
Notebook	16,5	-5,8	0,16	234	0,08

Výsledky testování počítačů v úsporném režimu jsou uvedené v této tabulce. Počítač PC 1 nemá možnost úsporného režimu vzhledem k nainstalovanému operačnímu systému Windows 98.

Zajímavé je , že oba počítače vykazují kapacitní složku výkonu (jalový výkon Q je záporný) .

Nároky napájecího systému notebooku

P(W)	Q(var)	cosφ	U(V)	I1 (A)	P (W)
61 - 63	-39	0,7	234,6	0,290 - 0,310	poslech hudby
65 - 75	-19	0,76	234,6	0,33 - 0,395	surfování po internetu
14,16	-5,9	0,15	235	142,8	při vypnutém stavu - dobíjení baterie
62	-19,5	0,71	234,3	0,297	proudový náraz při zapnutí - start windows

Při měření notebooku se nám nepodařilo oddělit změny v odběru proudu ze sítě při jednotlivých režimech v plné míře viz. Tabulka, protože stav nabití baterie byl do jisté míry proměnný a v čase se měnil. Z časových důvodů nebylo možné změřit notebook vždy s plně nabitou baterií.

Ekonomické vyhodnocení vhodného režimu provozu počítačů

	1 hodina	8 hodin	1 den	1 měsíc	1 rok
počet kWh	0,13	1,04	3,12	93,6	1123,2
cena za energii	0,52	4,16	12,48	374,4	4492,8

Ekonomické vyhodnocení je velice zajímavé, z tabulky je patrno, že trvalý provoz počítače, tak jak řada lidí praktikuje na doporučení počítačových expertů, je velkou zátěží pro naše finance.

Ekonomické vyhodnocení úsporného režimu provozu počítačů

	1 hodina	8 hodin	1 den	1 měsíc	1 rok
počet kWh	0,009	0,072	0,216	6,48	77,76
cena za energii	0,036	0,288	0,864	25,92	311,04

Ekonomické náklady na provoz počítače v úsporném režimu jsou přibližně 12x menší než v plné zátěži.

Závěr

V tomto projektu jsme zjistili jakou má doopravdy spotřebu obyčejný počítač i notebook. Zjistili jsme za jakých okolností má počítač největší spotřebu a kdy má spotřebu naopak nejmenší a kolik nás jejich provoz stojí za různá časová období. Podle našich měření jsme zjistili, že počítač má v zatíženém stavu spotřebu jako 3 až 4 čtyřicetiwattové žárovky. Z toho plyne, že by lidé neměli počítače vůbec zapínat…

Použitá literatura

HAMMER, Miloš, KRATOCHVÍL, Petr. *Měniče elektrické energie (interaktivní studijní text).* Vyd. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství Ústav výrobních strojů, systémů a robotiky Odbor elektrotechniky, 2004
Dostupné též na World Wide Web: <www.uvssr.fme.vutbr.cz/opory/menice/>

Internetové stránky VOŠ a SPŠE Františka Křižíka, Praha 2006. *Studijní materiály Elektronika*.
Dostupné na World Wide Web: < http://81.0.251.93/tab09.htm>

PAVLÍČEK, Jiří. *EDUCON (Výukový systém - interaktivní studijní text). 2005*. Západočeská univerzita v Plzni. Zpracováno v rámci diplomové práce na Katedře elektromechaniky a výkonové elektroniky FEL ZČU v Plzni.
Dostupné na World Wide Web < http://educon.zcu.cz/search.php?rstext=all-phpRS-all>
Text je autorem průběžně rozšiřován.

BEZDĚK, Miloslav. *Elektronika I.*
1. vydání. České Budějovice: Kopp, 2002. 272 s. ISBN 80-7232-171-4

SILVER, H. W. *Experimenty z elektroniky – 8. Spínané regulátory.* Radioamatér č. 3/2005, s. 8 až 10.
Vyd. Praha: Český radioklub, prostřednictvím společnosti Cassiopea Consulting, a. s. ISSN 1212-9100.
Dostupné též na World Wide Web: < http://www.radioamater.cz/cisla/pdf/2005-3.pdf/>

TICHÝ, Milan. *Elektronika.* 2002.
Dostupné na World Wide Web <http://lucy.troja.mff.cuni.cz/~tichy>

Středoškolská technika 2008

**Setkání a prezentace prací
středoškolských studentů na ČVUT**

KOSTEL sv. MIKULÁŠE – LOMNICE n. POP.

Jan Trejbal

SPŠ Stavební Liberec
Sokolovské náměstí 14, 460 31 Liberec 1

Cílem práce je přehledně popsat stavební objekt a sestavit osnovu základních informací.

Práce je tvořena chronologicky od historie objektu po jeho současnost. Mapuje jeho historii, nosná data, vývoj a rekonstrukce. Popisuje vnější i vnitřní architektonické části objektu. V práci jsou též zmíněna jména, která mají s objektem úzkou spojitost.

Práce umožňuje široké veřejnosti se seznámit jak se vzhledem, tak i s teoretickými fakty daného objektu. Dovoluje zájemcům nahlédnout do zákonitostí stavební architektury. Stručně naznačuje účel stavby. Lepšímu porozumění popisovaného objektu pomáhá obsáhlá fotodokumentace, která zachycuje potřebné detaily a umožňuje tak vytvořit si náhled na danou problematiku.

Byla použita data z webových stránek města, ve kterém se objekt nachází, dále znalosti lidí, kteří jsou obeznámeni s problematikou dané stavby.

Práce má přínos pro daný region v oblasti kultury a turistického ruchu. Z hlediska ucelenosti informací se jedná o jedinou práci svého druhu.

Pohled na průčelní a boční část kostela



Středoškolská technika 2008

Setkání a prezentace prací středoškolských studentů na ČVUT

OBCHODNÍ DŮM JEŠTĚD VČERA, DNES A ZÍTRA

Zuzana Večerníková, Tomáš Kotrbatý

Střední průmyslová škola stavební
Sokolovské náměstí 14, Liberec

Cílem práce je vytvořit přehledný popis vývoje obchodního domu Ještěd a přiblížit jej široké veřejnosti.

Popis je řazen chronologicky od historie objektu až po jeho budoucí osud. Obsahuje historii místa, průběh stavby objektu, jeho využití, probíhající demolici a plánovanou výstavbu.

Práce se dále snaží přiblížit názory laické i odborné veřejnosti na daný problém.

Informace byly čerpány jak z odborných knih, časopisů a internetových stránek, tak i z libereckého archivu.



Středoškolská technika 2008

**Setkání a prezentace prací
středoškolských studentů na ČVUT**

Martin Dolejš, Libor Tomíček, Rosťa Fikáček a Pavel Skořepa

**VYŠŠÍ ODBORNÁ ŠKOLA A STŘEDNÍ PRŮMYSLOVÁ ŠKOLA DOPRAVNÍ
MASNÁ 18, 110 00 PRAHA 1**

Stirlingův motor

Naše škola pod vedením ředitele Ing. Jana Kadeřábka má již více než padesátiletou tradici ve výchově odborníků, kteří se uplatňují jak ve všech druzích dopravy, tak i v oborech, které s dopravou více méně souvisejí.
Mnoho našich žáků, kteří složili maturitní zkoušku na střední odborné škole, pokračuje ve studiu na zdejší vyšší odborné škole nebo ve studiu na vysokých školách.
Škola je dobře vybavena audiovizuální, prezentační a výpočetní technikou.
Odborné učebny jsou vybaveny tak, aby řádně plnily svůj účel a aby žáci měli přístup k ověření teoretických poznatků z praxe.

Žáci třídy S2 – jmenovitě
Martin Dolejš, Libor Tomíček, Rosťa Fikáček a Pavel Skořepa,
se na soutěž připravovali ve výuce praxe, ale hlavně ve svém volném čase.

Výroba našeho motoru

Náš motor jsme začali vyrábět na podzim 2007 pro soutěž
„Vyrob si svůj Stirlingův motor“ pořádanou SPŠS Betlémská.

Podmínkou soutěže byly stanovené rozměry spodní ohřívané komory, které jsou zajištěny obyčejnou plechovkou od konzervy. Naším úkolem bylo vymyslet technická vylepšení, se kterými by naše motory dosáhly nejvyšších výkonů.

Soutěže se zúčastnilo 17 soutěžících ze škol: SPŠS Betlémská, SPŠ dopravní Masná, SPŠ Třebešín a SPŠS Novoborská.

Stirlingův Motor

Historie

První Stirlingův motor byl vynalezen roku 1816 Robertem Stirlingem. Později roku 1818 postavil motor s výkonem 2 hp na čerpání vody v kamenolomu.

V 19. století byly tyto motory velmi populární hlavně v USA. Stirlingův motor byl používán na statcích, v dolech, na železnicích, na pohon navijáků
při čerpání vody. Kvalita a výkon motoru je v tomto století ovlivněn hlavně dostupnými materiály. Proto byl na začátku 20. století vytlačen spalovacími motory a elektromotory.

Princip

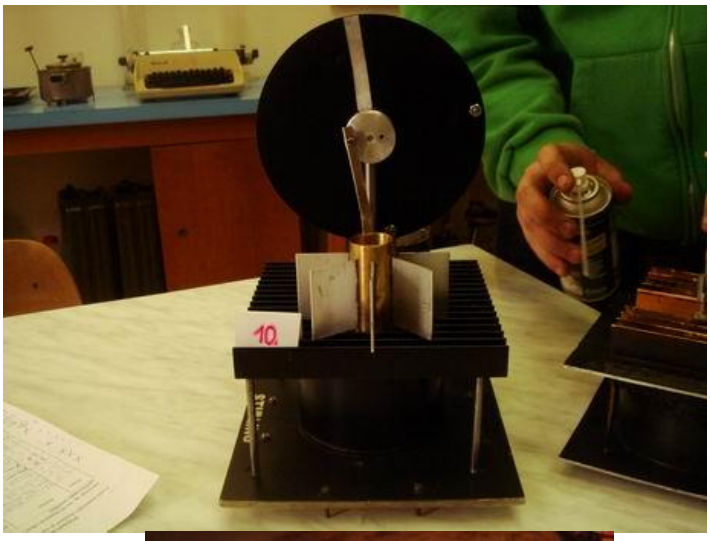
- Stirlingův motor má dva pracovní prostory, mezi kterými může volně proudit plyn.
- Uzavřený objem vzduchu (plynu) je na jedné straně motoru ohříván,
  na druhé ochlazován.

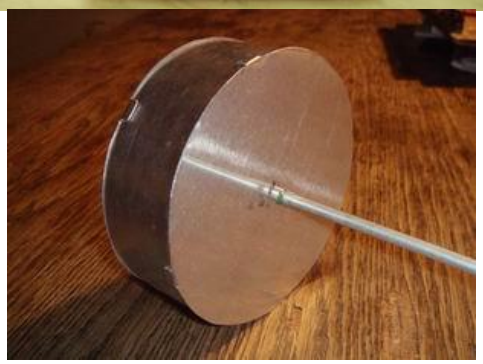
- V motoru se vytváří střídavě relativní přetlak a podtlak, které působí na pracovní píst.
- Tento píst pohání klikovou hřídel, od níž je odvozen také pohon přehaněče.
- Čím je větší teplotní rozdíl teplé a studené strany, tím větší je práce jednoho oběhu a tím také výkon motoru.
- Tohoto motoru existuje mnoho modifikací.
    - Písty mohou být v samostatných válcích.
    - Písty mohou být v jednom válci společném, kdy jeden z pístů pracuje v dvojčinném režimu.

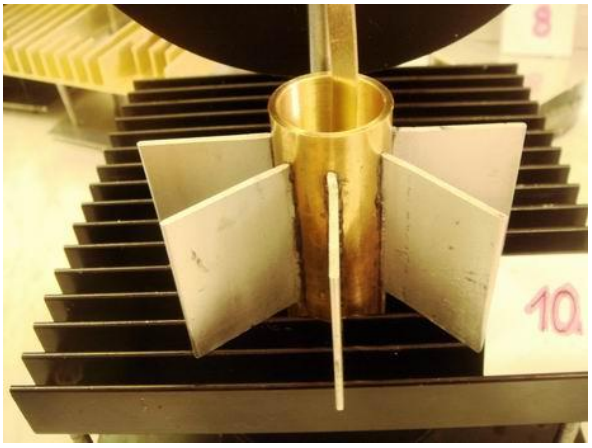
Pracovní cyklus

1.) Oba písty se pohybují společně, expandující zahřátý plyn v horkém prostoru koná práci.
2.) Řídící píst začíná vytlačovat plyn z horkého do studeného prostoru. Celkový objem se nemění, není tedy konána práce.
3.) Pracovní píst začíná stlačovat plyn ve studeném prostoru. Tlak ochlazením klesne, proto je vykonávaná práce menší, než při expanzi.
4.) Stlačený studený plyn proniká do horkého prostoru, kde po zahřátí začne expandovat.

Autor: Martin Dolejš



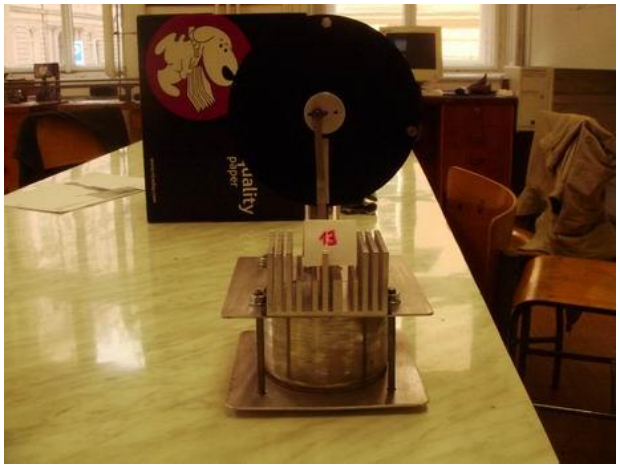




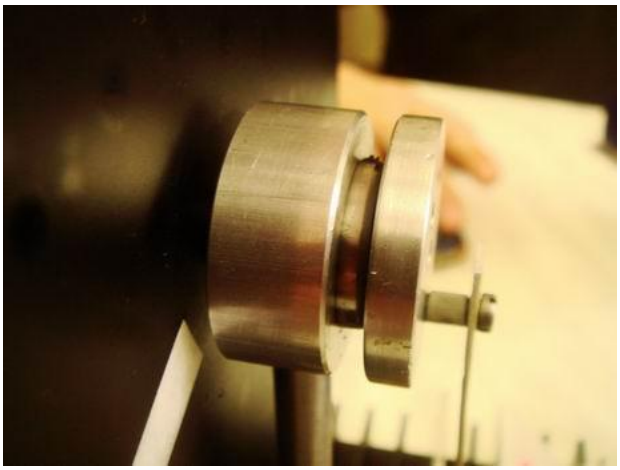
Autor: Rosťa Fikáček



Autor: Pavel Skořepa









Hodnotící komise: Ing. Toman, Jan Haken

Závěr:

Soutěže se zúčastnilo 17 soutěžících ze škol: SPŠS Betlémská, SPŠ dopravní Masná, SPŠ Třebešín a SPŠS Novoborská.

Na této soutěži jsme uspěli v následujících kategoriích:

1. kategorie - NEJVYŠŠÍ RYCHLOST - jsme získali 1. a 2. místo
2. kategorie - TECHNICKÁ VYLEPŠENÍ - jsme získali 1. a 3. místo
3. v kategorii VZHLED jsme obdrželi 1. místo.

Tabulka výsledků.

Start. Č.	Jméno	Škola	1.rozběh	2.rozběh	Otáčky	Design	Tech. vylepšení
2	Jakub Horníček	Betlémská	325	300		2	
3	Honza Chlad	Betlémská	217	223			
4	Norman Cziszár	Betlémská	0	274		3	2
8	Josef Turek	Betlémská	139	0			
9	Jarda Mach a kol.	Prosek	434	455	3		
10	Martin Dolejš	Masná	270	251		1	1
11	Libor Tomíček	Masná	0	580	1		3
12	Rosťa Fikáček	Masná	418	445			
13	Pavel Skořepa	Masná	538	506	2		
14	Martin Chochola	Třebešín	186	136			
15	Honza Fiedler	Třebešín	38	272			
17	Jarda Chochola	Třebešín	381	285			

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

LEVITÁTOR

Ondřej Socha, Jaromír Kult, Filip Kučák
Střední průmyslová škola,Ostrava-Vítkovice, příspěvková organizace

Konzultovali učitelé:
Ing. Leoš Petřkovský
Ing. Sokolová Vladislava
Ing. Leo Večeřa
Výroba konstrukce-učitelé v dílnách:
Alois Lasák
Antonín Dombek

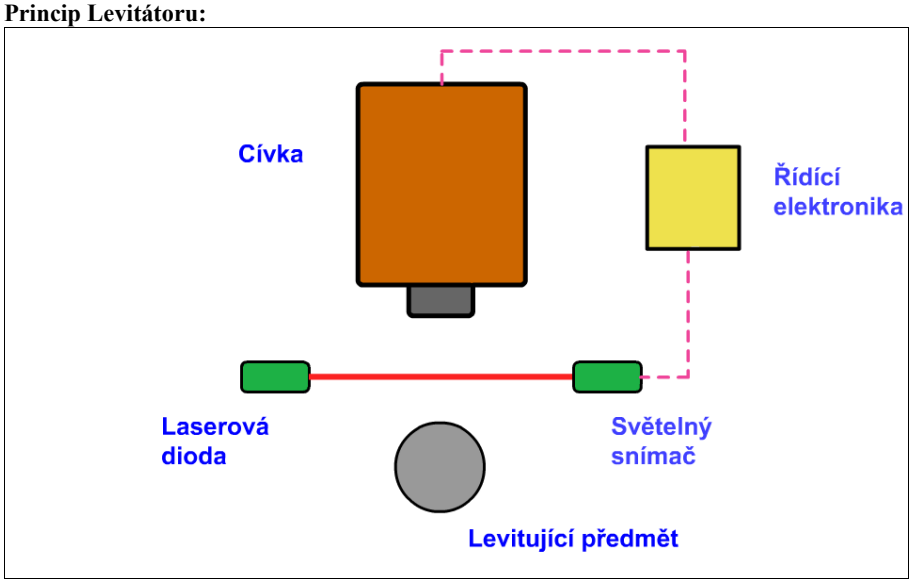


Anotace:

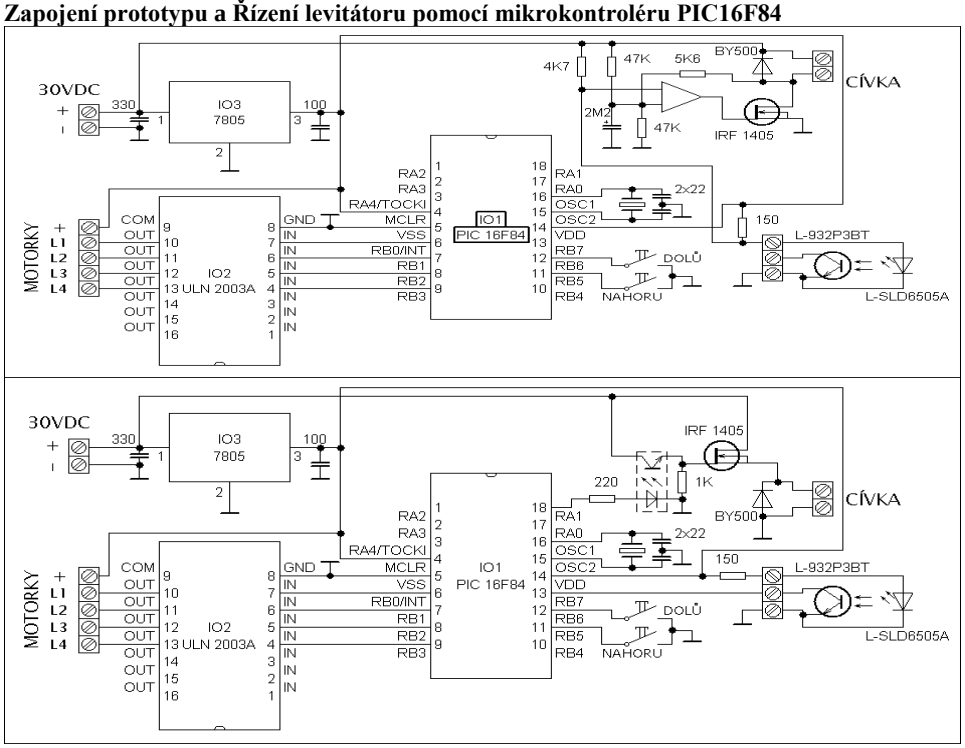
Naše zařízení dokáže působením silného magnetického pole udržet ve volném prostoru feromagnetické látky. Velikost proudu a tím magnetické indukce je řízena pomocí zpětné vazby. Světelná závora zajistí napětí v závislosti na ploše jejího překrytí,operační zesilovač jej zesílí a porovná s napětím na cívce. Velikost protékajícího proudu řídí výkonový unipolární tranzistor MOSFET. Cívka je chlazena ventilátorem, jehož druhá funkce je, že předmět nejen levituje, ale také rotuje kolem svislé osy.

Levitátor má různé možnosti využití ,kromě hračky pro pobavení,jej využíváme jako učební pomůcku pro demonstraci silového působení magnetického pole,vlivu proudu protékajícího cívkou a dalších jevů v elektrotechnice,elektronice a automatizaci. Pracujeme na nové verzi,která bude řízena mikrořadičem PIC,s možností posuvu levitujícího předmětu a bude schopna pracovat s těžšími předměty.

Princip Levitátoru:

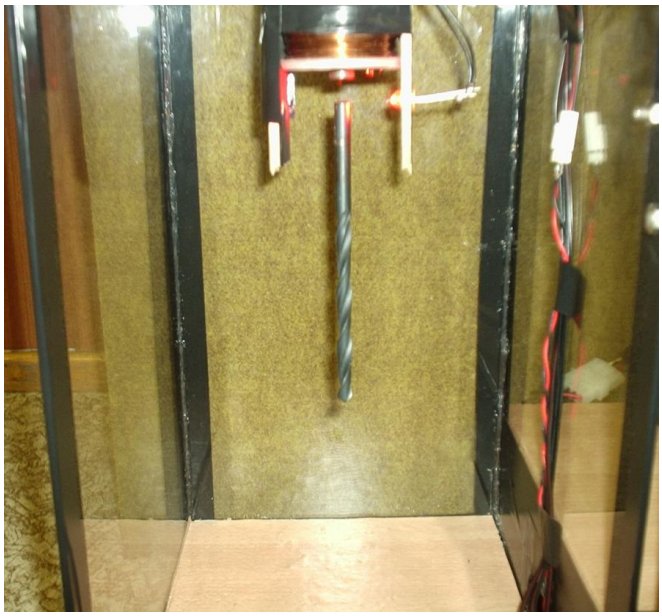


Zapojení prototypu a Řízení levitátoru pomocí mikrokontroléru PIC16F84

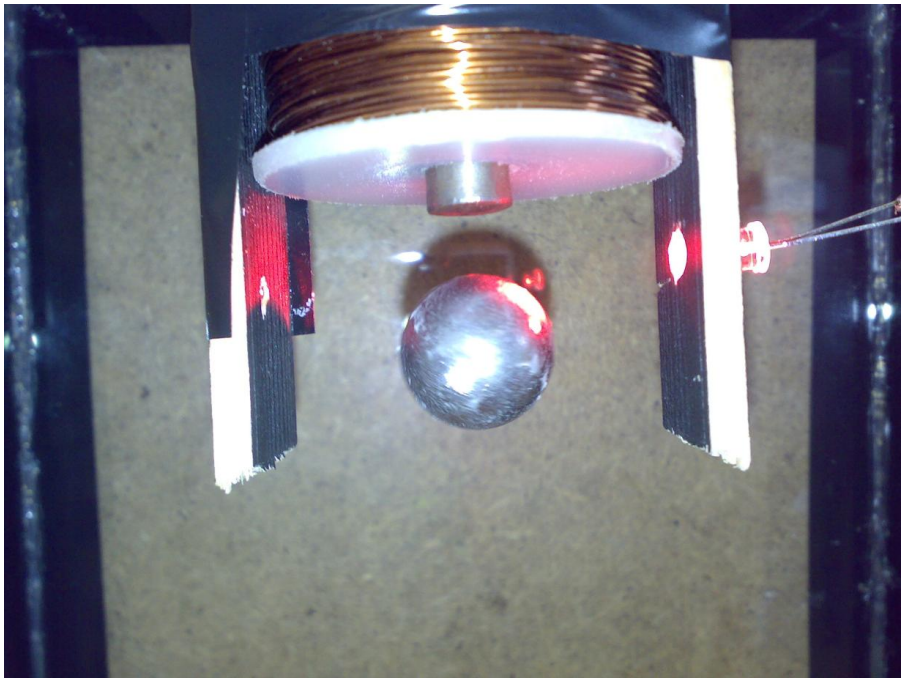


Levitátor slouží například:

- Jako školní pomůcka k demonstraci magnetických jevů
- Pomocné zařízení pro lakování předmětu ze všech stran najednou bez otáčení a nutnosti kontaktního úchopu
- Ve větším provedení např. jako halový jeřáb pro přesun a manipulaci předmětu opět bezkontaktně např. z důvodu zachování kvalitního povrchu předmětu











Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Gotická okna

Lukáš Vacovský

SPŠST Panská
Panská 3, Praha 1

Vacovský, Lukáš: Gotická okna, Praha 2008

Práce, zaměřená na konstrukce gotických oken, je rozdělena do dvou hlavních částí: Základní konstrukce a Příklady gotických oken. Obě části obsahují postupy konstrukcí, výkresy zpracované pomocí programu AutoCAD 2007 a jsou obohaceny o fotodokumentaci a výkladový text k budovám, ze kterých pochází zkonstruovaná okna. Práce tvoří sbírku konstrukcí.

Jakým způsobem je gotická kružba konstruována? Ačkoliv by se očekávalo, že odpověď na tuto otázku můžeme nalézt ve většině knihoven, vždyť se jedná o jeden ze základních architektonických prvků gotiky, který můžeme najít ve většině českých měst, skutečnost je jiná. Publikace o gotice se především zajímají o to, co okno zobrazuje, ale již se nezajímají o geometrickou konstrukci. Při pátrání jsem nenarazil na žádnou mě přístupnou publikaci, která by se daným tématem zabývala. Veškeré informace, na které jsem narazil byly roztroušené do všech možných zdrojů a mnohé byly jen kusé a pouze inspirativní. Tato neúplnost a roztroušenost informací mě vedla k myšlence vytvoření práce, která by shrnovala informace, které jsem získal nebo odvodil.

Práce je rozdělena do dvou základních částí. První část, zaměřenou na základní konstrukce, jsem rozdělil do tří podkapitol pojednávajících o tvaru oblouku, kružbě vyplňující lomený oblouk a rozetách. Každá kapitola obsahuje text popisující daný prvek, obrazovou dokumentaci a postup konstrukce. Druhá část se zabývá konstrukcemi existujících oken.

Okna do druhé části jsem se snažil vybírat tak, aby se nějakým způsobem vztahovala k podkapitolám první časti. K podkapitole o tvarech oblouků jsem vybral kružby z Dóžecího paláce a paláce Ca'd'Oro. Myslím si, že Benátská gotika je nejlepší přiklad, na kterém lze ukázat, že gotické oblouky se vyvinuly i do dalších tvarů, než je tvar lomený. Podkapitolu o kružbách v lomených obloucích reprezentují okna z chrámu sv. Víta a Týnského kostela. Tyto okna též zastupují vrcholnou a pozdní gotiku. Podkapitolu o rozetách reprezentují Staronová synagoga, Notre Dame v Paříži a Saint Chapelle. Na těchto příkladech je také vidět vývoj rozet od rané gotiky po pozdní.

Na následujících stranách je uvedena ukázka části práce.

4.6 Sainte Chapelle

Sainte Chapelle se nachází v historickém srdci Paříže na ostrově Ile de la Cité v komplexu budov, které tvořili v období gotiky královský palác. Kapli dal postavit král Ludvík IX. v letech 1242 - 1248, jeden z posledních velkých křižáckých králů, jako místo pro uložení nejvýznamnějších křesťanských relikvií. Mezi tyto relikvie patřila Kristova trnová koruna, Obrázek z Edessi nebo kus Pravého kříže. Pro srovnání: nákup relikvií stál Ludvíka 135 000 livrů, zatímco stavba kaple pouhých 40 000 livrů.

Kaple byla postavena na obdelníkovém půdorysu a skládá se ze dvou částí. Dolní nižší a temnější část sloužila jako pařížský kostel pro všechny obyvatele paláce, nad touto kaplí byla postavena samotná královská kaple, která je dnes považována za nejvýznamnější pozdně gotickou stavbu na světě. Královská kaple je obrovský prostor osvětlený velkými okny, kterými proniká do interiéru magicky barevné světlo. Na místě oltáře stojí velký baldachýn, kde bývaly vystaveny relikvie.

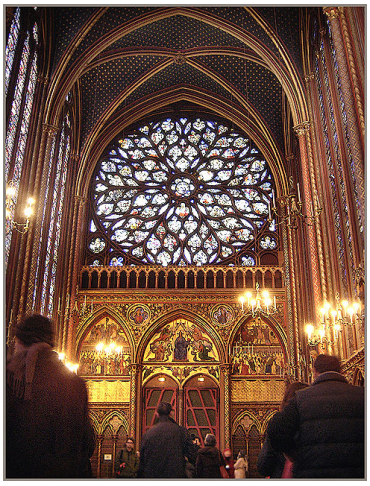
Stavitel kaple je neznámý, nejčastěji se zmiňuje jméno Pierre de Montreuil, který se podílel na stavbě opatství Saint Denis a dokončil fasadu Notre Damu.

Kaple byla výrazně poničena za Francouzské revoluce, kdy byla zničena věž, odstraněn baldachýn, ztraceny relikvie a mnoho dalšího. Rekonstrukce proběhla roku 1855 a od roku 1862 je považována za národní památku. *(foto – www.sacred-destinations.com)*





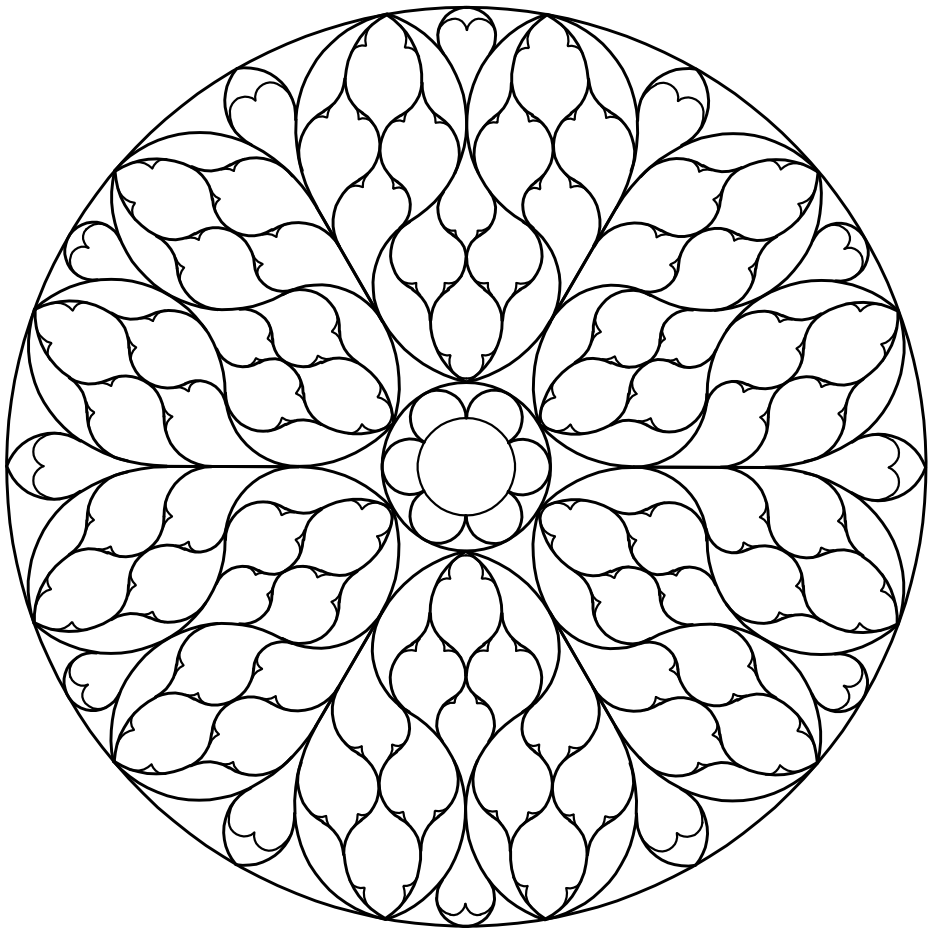


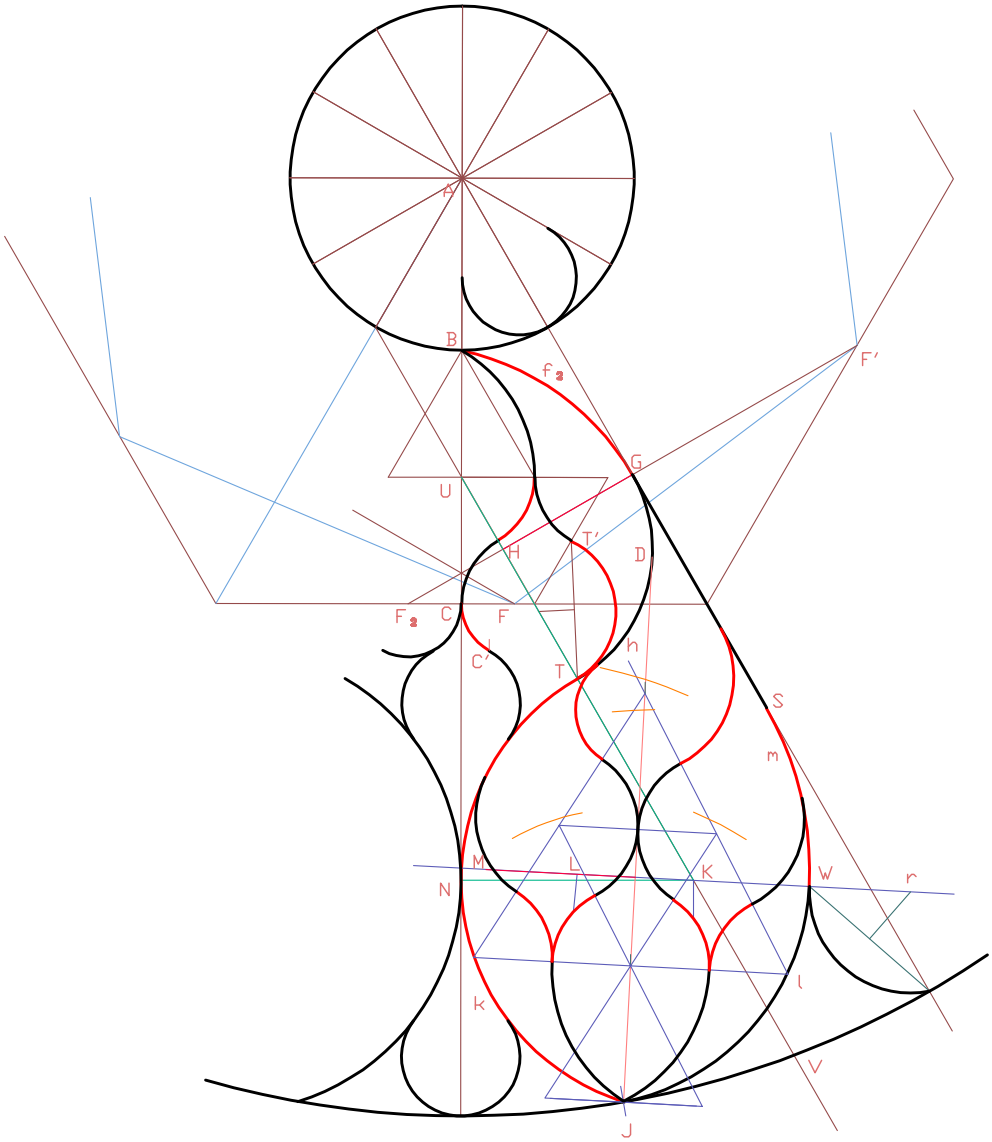


4.6.1 Postup konstrukce rozety na Saint Chapelle

1) Šestiúhelníku, jehož kružnice vepsaná má poloměr AC.
2) F; F_2 Leží na stranách šestiúhelníku ve stejné vzdálenosti od C. Tuto vzdálenost jsem určil libovolně.
3) F' je bod F analogicky narýsovaný na sousední straně šestiúhelníku.
4) G je střed úsečky $F'F_2$
5) f_2 tvoří oblouk BG a má střed F_2
6) U je střed úsečky BC. Z bodu U vedeme přímku rovnoběžnou s AG *(dále UV)*. Bod H, který tvoří střed h tvořící oblouk GT, vznikne průnikem úsečky F_2G a přímky UV.
7) K sestrojení plaménkové výplně vznikajícího laloku jsem použil rovnostranné trojúhelníky, jak je znázorněno na konstrukci. Vyjímku tvoří oblouk TT', který jsem sestrojil pomocí Bodů T'; T a přímky UV na které leží střed oblouku a oblouk C'C, jehož střed je F.
8) k tvořící oblouk TJ má střed K na UV a je tečná oblouku GT a přímce AC.
9) J vznikne na k ve třetinové vzdálenosti mezi přímkami AC a AG
10) Úsečka DJ je tečna oblouku GT (h) a tvoří pomyslnou osu plaménkové výplně druhého laloku.
11) Zrcadlením K podle DJ vznikne L, který je středem l tvořící oblouk JW
12) Bod M leží na přímce KL ve vzdálenosti odpovídající délce úsečky GH od průsečíku přímky KL s úsečkou DJ. Tuto vzdálenost jsem získal, když jsem se snažil najít vztah mezi bodem M a zbytkem konstrukce. Bod M je středem oblouku SW (m). Který je tečný k přímce AG.
13) Plaménková výplň druhého laloku je sestrojena pomocí rovnostranných trojúhelníků, jejichž výška je rovna ¼ |DJ|. Jejich konstrukci není nutno zvlášť popisovat. Jediný problém nastává u oblouků, které jsou tečné k obloukům ohraničujících prostor laloku, v těchto případech jsem byl nucen sestrojit křivku, na které leží středy všech tečných kružnic.

Rozeta Saint Chapelle




A
B
f₂
F′
G
U
T′
H
D
F₂
C
F
C′
h
T
S
m
M
L
K
W
r
N
l
k
V
J

Středoškolská technika 2008

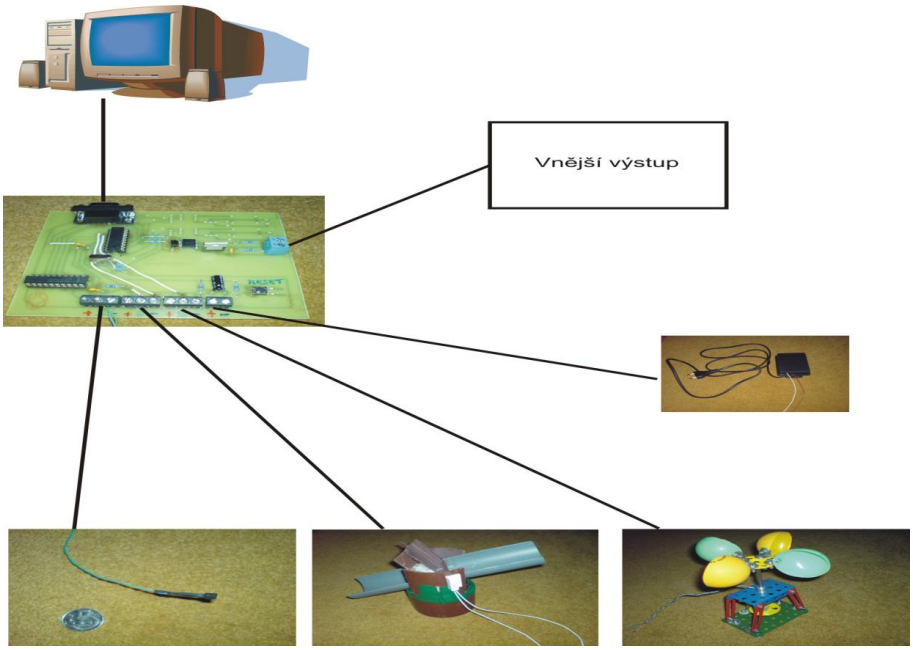
Setkání a prezentace prací
středoškolských studentů na ČVU

Automatický zavlažovací systém

Jan Gottlieb, Jaroslav Novák

Střední škola sdělovací techniky
Panská 3, Praha 1

Důvodem vývoje tohoto zařízení bylo vytvoření automatického zavlažovacího systému pro zahrádkáře, kteří nemají dostatek volného času k péči o své zahrádky. Celé zařízení je sestaveno z hlavního mikrořadiče a třech čidel sbírajících informace jako je teplota, rychlost větru a počet srážek. Poté se získaná data zpracovávají v mikrořadiči, kde se po vyhodnocení rozhodne, zda se započne se zavlažováním zahrady. Každé čidlo je navrženo tak, aby vyhovovalo našim požadavkům. Původní idea byla vytvořit každé čidlo s vlastním mikrořadičem, toto řešení jsme zavrhli a začali pracovat na tom, že vše bude zpracovávat jeden hlavní mikrořadič. Velkou výhodou oproti komerčním výrobkům, je konfigurace zařízení pomocí běžného PC, kde můžeme v námi vyvinutém programu nastavit pro jakou teplotu, sílu větru a srážky se určitý zavlažovací okruh zapne. Tato komunikace je zařízena pomocí sériového portu (COM) portu z PC a UARTu v PICu.



Obr. 1: blokové schéma

Celé zařízení je založeno na principu, že je nejdříve naprogramováno přes PC, pro jaké teploty, sílu větru a srážky se začne zalévat. Komunikace je docílena vytvořením sériového přenosu dat mezi PC a procesorem ve výrobku, kde se přes UART poslaná dana uloží do paměti. Poté se pouze čeká na příjem dat od čidel. Přijatá data z každého čidla se uloží do paměti a pak porovnávají. Když je vše vyhodnoceno, dle výsledku se rozhodne, zda-li se nějaké ventily otevřou či nějaké zavřou. Určitou dobu se vyčká a proces kontroly počasí se zopakuje.

Obsluhování čidel je zařizováno pomocí mikrořadiče Pic (tzv. integrační přírůstková metoda).

Popis jednotlivých částí

Větrné čidlo

Je řešeno jako větrník s optickou bránou. Jakmile zafouká vítr, roztočí se osa, která má v dolní části okruží s otvorem, skrz který se spíná optická brána. Ta je tvořena infračervenou diodou a foto-tranzistorem, který obvod spíná. Jako výstupní data dostáváme impulzy díky kterým jsme schopni určit rychlost větru.

Srážkové čidlo

Je řešeno jako překlápějící se člunek s vodou, kde je zaznamenáváno počet překlopení člunku. Podle počtu překlopení se určí počet srážek a jejich počet se poté posílá do PICa.
K indikaci překlopení jsme použili relé spínané magnetem.

Tepelné čidlo

U tohoto čidla jsme využili digitálního teploměru (DS18S20), který posílá v devíti bitech hodnotu teploty s přesností na půl stupně Celsia.

Napájecí část

Celá napájecí část je zapojena přes transformátor do elektrické sítě. Na primární cívku transformátoru se ze sítě naindukuje napětí ~230V, na sekundární cívce se magnetickými toky naindukuje napětí ~10V. Toto napětí prochází do diodového můstku, který ze záporné složky sinusového napětí s maximální hodnotou 22V udělá kladnou složku. Výsledkem je kladný signál, který kondenzátor C1 vyhladí na 17V a ten pak pokračuje do již kondenzátory ošetřeného stabilizátoru napětí. Stabilizátor funguje v rozmezí 7 – 35V, proto jsme museli signál ošetřit silným kondenzátorem, aby signál pod tuto hladinu neklesl. Na výstupu se poté objeví stabilizované napětí +5V, které se rozvádí do součástek na plošném spoji.

Hlavní část

Jedná se o nejdůležitější část práce, samotné vyhodnocování zpracování a také otevírání ventilů zavlažovacího systému. Proto se v této části nachází hlavní mikrořadič PIC 16F648A, převodník UART/TTL - MAX 3235, stejně tak i samotné ovládání ventilů. K tomuto obvodu jsou hlavně připojeny všechna čidla. Jinak se na tomto obvodu také nachází okruh, pomocí kterého lze PIC programovat přes PC klidně i za běhu programu.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

VÝUKOVÝ PROGRAM – TELEKOMUNIKAČNÍ SYSTÉMY

František Kubát, Lukáš Trnka

SPŠ Sdělovací techniky
Panská 856 / 3, Praha 1

Tento výukový program byl vytvořen jako doplněk pro výuku předmětu Telekomunikační systémy a snaží se naplno využít možností interaktivní tabule. Program se skládá z několika prezentací, z nichž každá slouží pro výuku jedné kapitoly. Jednotlivé prezentace jsou tvořeny s důrazem na přehlednost, názornost a obsažení všech hlavních bodů dané kapitoly.

Zvolené programy:

Microsoft Office:

Microsoft Office je sada kancelářských programů, která obsahuje především program Word pro psaní dokumentů, Powerpoint pro vytváření prezentací, Excel pro vytváření tabulek a další. Tato sada nám postačila k většině úkonů spojených s naší prací.

Microsoft Powerpoint:

Naším hlavním pracovním nástrojem byl program Microsoft Powerpoint, který je přímo určen pro vytváření prezentací a bezproblémově spolupracuje s interaktivními tabulemi umístěnými ve škole. Umožňuje psaní a upravování textu, poměrně velké možnosti tvorby obrázků a vytváření komplexních animací. Je tedy pro náš účel naprosto postačující.

Další programy:

Ze začátku jsme pro tvorbu obrázků používali program CorelDraw. Ten sice nabízí velké možnosti pro grafické úpravy. V rámci naší práce jsme ale většinu těchto možností nepotřebovali. Naopak jsme naráželi na problémy vyplývající z užití programu CorelDraw. Vzniklé obrázky jsme museli exportovat z vektorového formátu do bitmapového a až teprve potom jsme se mohli podívat, jak bude výsledek vypadat v samotné prezentaci. Případné úpravy byly komplikované, a proto jsme raději využili vektorových grafických nástrojů sady Office.

Pro některé úpravy jsme využívali programu Microsoft Malování, který je běžnou součástí operačního systému Windows. Malování nám umožnilo především ořezávání nepotřebných částí naskenovaných obrázků či obrázků z Internetu.

Tvorba prezentací:

Na úvodním snímku každé kapitoly jsme pro podklad zvolili přechod tvořený kombinací barvy vybrané pro dané téma a bílé barvy. Pruh vedoucí vodorovně přes stránku je

tvořen opět přechodem barev. Po spuštění je tento snímek animován, a po chvíli plynule přejde na další stranu kde začíná běžný snímek s danou látkou.

Základem každého běžného snímku je bílá stránka s barevným pruhem v záhlaví. Na tomto pruhu v pravém horním rohu obrazovky je zkratka nadpisu kapitoly, např. EWSD, v levém horním rohu se může nalézat podnadpis určité části. Na tento snímek se poté umísťuje veškerý text a obrázky

Volba fontu použitého v prezentacích:

Prakticky ve všech prezentacích jsme použili písmo Arial, které je k nalezení téměř na všech počítačích. Tím tedy v podstatě odpadá riziko špatného zobrazování na počítačích s jinou nabídkou fontů. Zároveň je to jednoduché a dobře čitelné písmo, takže prezentace by měly být dobře čitelné i ze zadních lavic ve třídě.

Zvolená velikost písma:

Běžný text je v prezentacích psán velikostí 18. U popisků ve schématech a obrázcích jsme používali různé velikosti písma, v závislosti na prostoru, který nám dané schéma poskytovalo. Vždy jsme se ale snažili použít co největší písmo a držet se v okolí velikosti 18.

Snažili jsme se, aby byl text tak veliký, aby byl zřetelný i ze zadních lavic, ale zároveň tak malý, aby se na stránku vešlo dostatečné množství informací. Jako nejvhodnější velikost při použití písma Arial nám vyšla velikost 18.

Hypertextové odkazy v prezentacích:

Pro snadnější navigaci a orientaci v programu jsme přidali i několik hypertextových odkazů, které bude možno využít i na interaktivní tabuli.

Všechny stránky ve všech prezentacích obsahují *zkratku* na obsah dané kapitoly. V levém dolním rohu stránky (znázorněno šipkou na obr.1) je její číslo, které slouží zároveň jako odkaz na obsah. Stejně tak obsah je vybaven hypertextovými odkazy, kterými se můžeme dostat na požadovanou část kapitoly.

Obr. 1: Modul ASM z kapitoly S12



Některé objekty v obrázcích obsahují hypertextový odkaz na jinou stránku, kde je tento objekt rozebírán podrobněji. Takovéto objekty jsou nejčastěji k nalezení v kapitolách EWSD a S12.

Animace:

Statický obrázek vysvětlující fungování nějakého systému nebo principu bývá pro nezasvěcenou osobu často nedostatečný pro úplné pochopení. Proto je v naší prezentaci několik animací, které by měly napomoci k porozumění.

Jednodušší animace se spustí hned po přechodu na stránku a nebo po kliknutí. Pak proběhne celá animace.

Některé animace, jako třeba výměnu signalizačních zpráv mezi ústřednami, jsme se rozhodli udělat tak, že po kliknutí se spustí jen část animace a po dalším kliknutí se spustí další část. Tak vzniká prostor pro vyučujícího, aby popsal právě proběhlý děj a teprve potom pokračoval další částí.

Kapitoly:

Pobočkové ústředny:

První kapitolou jsou Pobočkové ústředny. Jedná se o malé ústředny pro soukromé firemní použití, které dovolují vytvořit si vlastní firemní telefonní síť. Výhodou použití pobočkových ústředen je ušetření nákladů na vnitrofiremní volání . Hovory které neprocházejí přes státní ústřednu se netarifikují. Provozovatel ústředny (majitel firmy) může kontrolovat a případně omezovat odchozí a příchozí volání podle vlastních potřeb, nezávisle na poskytovateli státních linek. Počet těchto linek připojených k pobočkové ústředně závisí na množství hovorů do veřejné sítě.

EWSD:

Tato kapitola je věnována systému EWSD, což je Digitální elektronický spojovací systém vyráběný firmou Siemens. Je používán v mnoha zemích světa včetně České republiky. V českých telefonních sítích je nasazen společně se systémem Alcatel 1000S12. Systém EWSD byl vybudován v jižní polovině republiky.

Systém je sestaven ze standardizovaných modulů, což umožňuje snadnou a rychlou výstavbu i pozdější rozšiřování. Díky standardizovaným rozhraním každého modulu, je možné postupně vybavovat telefonní ústředny novými technologiemi bez nutnosti provádění nějakých zásadních zásahů do provozu telefonní sítě.

1000S12:

Ústředny 1000 S12 od firmy Alcatel SEL se zakládají na systému plně distribuovaného řízení. Celá ústředna má hlavní část nazývanou digitální spojovací pole (DSN), ke kterému se připojují jednotlivé moduly ústředny. Každý modul má svoje vlastní řízení, takže pokud se nějaký modul porouchá, ovlivní to pouze k němu připojené periferie. Ústředna i v tomto případě pokračuje v provozu bez ovlivnění. Moduly se dají vyměňovat za provozu, lze je tedy kdykoliv jednoduše vyměnit.

Signalizace:

Signalizace je soubor informací, které jsou zodpovědné například za sestavování, udržování a rušení telekomunikačního spojení.

Plesiosynchronní digitální hierarchie:

PDH slouží jako přenosové médium, do kterého se dají sdružovat kanály PCM. Dovoluje přenos většího objemu dat a vyššími rychlostmi než základní PCM. Hlavními součástmi jsou multiplexor a demultiplexor. Multiplexor sdružuje vždy 4 signály nižšího řádu do signálu vyššího řádu. Poté jsou data přenášena a na druhé straně se postupným demultiplexováním dostaneme zpět k signálům nižšího řádu. Přenášené signály nemají pevný

časový vztah oproti signálu vyššího řádu, proto je vyčleněna určitá rezerva pro odchylky přenosových rychlostí. Nevýhoda PDH je nutnost pokaždé demultiplexovat a zase znova multiplexovat, pokud si chceme z PDH „vybrat“ konkrétní signál. Problémem při častém multiplexovíní a demultiplexování je degradace signálu. Později začal být tento systém nahrazován systémem SDH, který tuto nevýhodu odstraňuje.

Synchronní digitální hierarchie:

SDH vzniklo jako následovník PDH, které již přestávalo stačit požadavkům na objemy přenášených dat. Oproti PDH má výhodu ve vyšší rychlosti, ale hlavně v tom, že pokud je třeba z celého datového toku vybrat malou část, nemusí se celý demultiplexovat a pak zase multiplexovat (ušetření nákladů a místa na potřebnou techniku). Základní transportní modul je rámec STM-1, skládá se ze záhlaví a informačního pole VC-4 ve kterém jsou obsažená samotná přenášená data. Nejnižší stupeň SDH dokáže přenášet například PDH 4. řádu. Nevýhodou SDH je nestandardizovaná technika, takže výrobky různých firem nejsou mezi sebou kompatibilní.

Na závěr:

Naší práci byla již od počátku přikládána důvěra a i přes různé problémy s kterými jsme se potýkali jsme ji zdárně dokončili. Na zkoušku byly prezentace několikrát použity vyučujícím i námi samotnými a vzbudily dobré ohlasy, tudíž se dá očekávat že budou při výuce používány a splní tak svůj účel. Je možné, že v budoucnu dojde k nějakým dalším úpravám a vylepšením.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

IDENTIFIKACE ZNÁMÉHO ZVUKU

Martin Koś

SPŠST Panská
Panská 3, Praha 1

Úvod

Začátkem bych vás chtěl seznámit s mojí dlouhodobou maturitní prací. Jak již napovídá zadání práce jedná se o zařízení pro identifikaci známého zvuku. Zařízení by mělo bezpečně rozeznat přicházející signál a příslušně na něj reagovat. Pro matematický výpočet vstupního signálu v DSP bylo zapotřebí ho nejprve převést do digitální podoby. O to se stará 12-ti bitový analog-digitální převodník procesoru dsPIC30F3012. Po převedení analogového signálu na digitální je potřeba výsledek spočítat pomocí Goertzelova algoritmu což je speciální případ Fourierovy transformace. Fouerierova transformace se zabývá vyjádřením časově závislého signálu pomocí harmonických signálů, tj. funkcí sin a cos, obecně tedy funkce komplexní exponenciály. Slouží pro převod signálů z časové oblasti do oblasti frekvenční. Goertzelův algoritmus potom z tohoto spektra získává pouze jednu určitou frekvenci (viz. kapitola Goertzelův algoritmus). Jelikož je přístroj zamýšlený jako pomůcka pro sluchově postižené při sportovních aktivitách v tělocvičně, zařízením reagujícím na zvuk je fotografický blesk, jehož záblesk (pouze teoreticky, protože zařízení zatím nebylo použito v tělocvičně) by se měl odrazit od lesklé plochy parket, a tak dát neslyšícím příkaz k zastavení.

Teoretický rozbor

Zde bych se pozastavil nad problematikou jednotlivých částí přístroje. Jako první jsem narazil na problém jak PIC naprogramovat. Nelíbila se mi moc myšlenka, že bych musel procesor pro každé naprogramování aktualizace programu vytahovat z patice vložit do programovacího zařízení a opět z tohoto zařízení procesor vyjmout a vložit zpět do mé desky se zapojením. Proto po pár poradách s mým třídním profesorem, který zároveň plnil funkci mého dohlížejícího jsme se shodli na tom, že by se nám hodil program, který by se nahrál do procesoru a umožňoval nám programování za chodu, což po jepřečtení datasheetu možné. Takovýto program jsme opravdu našli a do dnešní doby si ho nemohu vynachválit. Umožňuje programovat procesor na desce, která byla samozřejmě pro toto zkušební. U finální verze jsem upustil od tohoto typu programování. Neviděl jsem potřebu aby závěrečná deska se zapojením byla vybavena sériovým portem, jež by zde nenacházel využití. Program se jmenuje TinyBootloader a umožňuje programování procesoru za chodu. Stručně řečeno, program na začátku zkontroluje zda-li je procesor připojen k sériovému portu počítače a zda-li se přes něj něco posílá. Pokud právě přijímá program, nahrává do procesoru a opět kontroluje připojení (dá se nastavit, v našem případě nastaveno na 3s) , teď již nic neposíláme program tedy čeká určený čas a spouští naprogramované instrukce.

Programování

Dále jsem narazil na problém jakým jazykem procesor programovat. Jelikož jsem si byl vědom svých znalostí assembleru, které nebyly zvlášť převratné rozhodl jsem se vstoupit tak trochu do neznámé vody a pokusit se procesor programovat procesor pomocí jazyka “C”. Postupem času jsem přicházel na myšlenku, že by možná I výuka tohoto jazyka na naší škole nebyla zbytečná. Jelikož programování, je to jen můj osobní názor, se stalo daleko přehlednější a díky produktům od firmy Microchip Mplab a Mplab C30 (což je verze s překladačem pro jazyk “C”) to nebyl žádný problém. Bohužel i v této sekci jsem narazil na problém týkající se A/D převodu. A/D převod jsem měl hotový, ale i po několikátém nahrání na procesor nefungoval. Po jisté chvíli jsem si byl prakticky jistý, že chyba nemůže být na mé straně a po chvíli bádání jsem narazil na to, že TinyBootloader A/D převod z jakéhosi pro mě nepochopitelného důvodu zakazuje. Řešení bylo tedy velice jednoduché napsat instrukci pro povolení A/D převodu a vše bylo ihned v pořádku. Dále bylo zapotřebí napsat vzorec pro Goertzelův algoritmus.

Goertzelův Algoritmus = problematika

Goertzelův algoritmus nebylo ani tak obtížné do procesoru napsat, jako tento program v něm zprovoznit. Tento algoritmus je sám o sobě velice přesný, používá se například v DTMF technologii, ale i tak se mi to zdálo jako velice dobré řešení. Program jsem napsal do procesoru, ale později jsem zjistil, že pro mé užití je až moc přesný. Po zobrazení zvuku píšťalky na osciloskopu jsem viděl, že námi rozpoznávaná frekvence

bude cca v rozmezí od 2400Hz až do 2600Hz, ale Goertzlův algoritmus pracoval pouze s přesnými frekvencemi. Pro ověření jsem vypracoval tabulku v Excelu abych zjistil, jak tento algoritmus ve skutečnosti funguje (viz sekce Goertzelův algoritmus). Po tomto kroku jsem zjistil, že mnou zkoumaný algoritmus má toleranci v řádu ±5Hz a mnou testovaná frekvence byla ±100Hz. Proto v programu byla nutná část pro průměrování několika hodnot, abychom dostali vůbec nějaký výsledek. Dále potom následovalo odečítání hodnot, které v závěru měli rozhodnout jestli se nejprve LED dioda (a později blesk) rozsvítí nebo ne. K tomu mi pomohl opět program TinyBootloader, který je vybaven sériovou komunikací s počítačem a dokáže odesílat data na procesor nebo naopak je z něj přijímat. Musel jsem si tedy stoupnout do pokoje, hvízdat na píšťalku a zjišťovat jaký je dostávaný výsledek. Zde se jednalo o koeficient algoritmu, který byl v dekadickém zápisu. Výsledek stačilo potom pouze přepsat do procesoru.

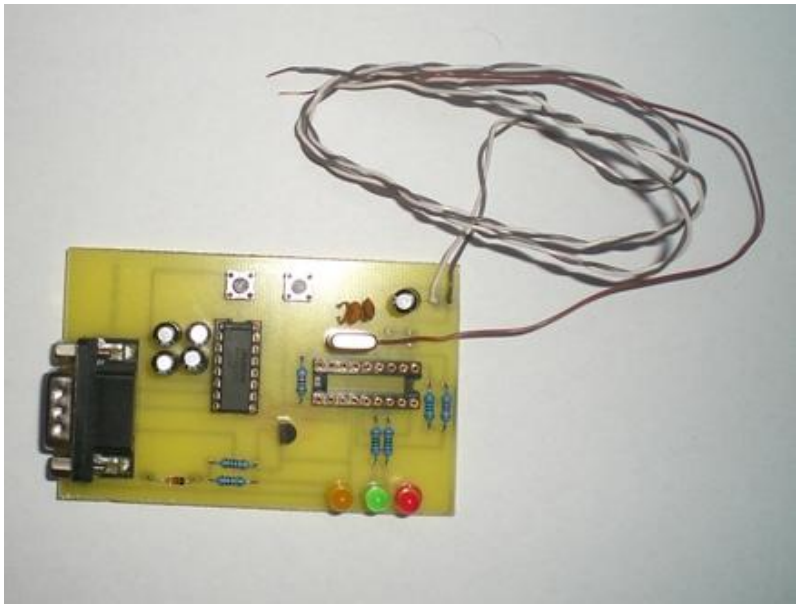
Konstrukce

V konstrukční části uvedu zkušenosti s pájením, tvorbou krabičky, leptáním. Jak je vidět dle schémat, využil jsem konstrukce obou možností, normální a SMD. Leptání nebylo žádný problém. Využíval jsem leptání fotocitlivou cestou, kdy plošný spoj po osvícení se nechá vyvolat v roztoku a potom vyleptat. Pro letování obvyklou metodou není ani tak co dodat, zkušební deska pracovala prakticky okamžitě a spoje byly kvalitní. Obtížnější bylo pájení SMD součástek. Jako minimální velikost pouzdra pro ruční pájení s mikropájkou bych doporučoval pouzdra velikosti 1206, i tak je to již velice obtížné a vyžaduje to velice velkou dávku trpělivosti a klidu. Dalším problémem byla neznalost pouzder typu SMD. Po zkušenostech z klasických součástek jsem byl si nebyl vědom mnoha chyb kterých jsem se dopouštěl. Nevěděl jsem jak poznat integrovaný obvod (kde je pin 1 a kde poslední pin). To jsem bohužel zjistil až ve skutečnosti, kdy jsem první zapojení s SMD součástkami pokazil. Nejen, že jsem vybral špatné pouzdro zesilovače, ale ještě jsem ho otočil o 180®, takže jsem obvod zároveň spálil. Konstrukce krabičky je poměrně jednoduchá a tak během ní nenastaly žádné problémy.

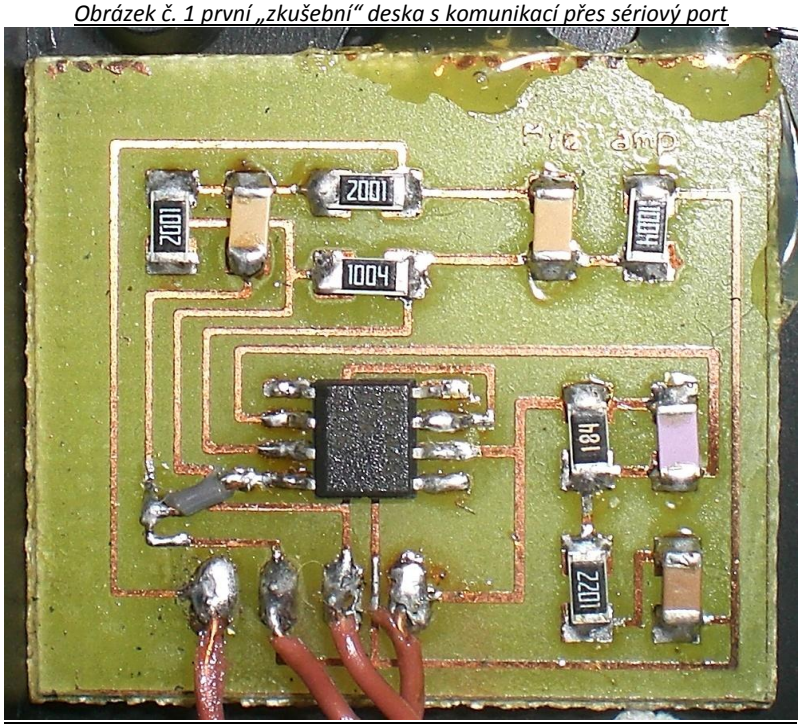
Mikrofon

Důležitým článkem je také zároveň mikrofon. Použil jsem jednoduchý elektretový mikrofon, který bez zesílení je prakticky nepoužitelný. Nyní tedy budu vycházet z parametrů po zesílení, protože vstupy A/D převodníku potřebují pro chod vstup minimálně 3,5 – 5,5V. Po zesílení dostáváme z mikrofonu analogový signál. Po změření na osciloskopu bylo vidět že rozkmit je mezi -2,5V – 2,5V, což pro vstup na piny A/D převodu krásně stačí.

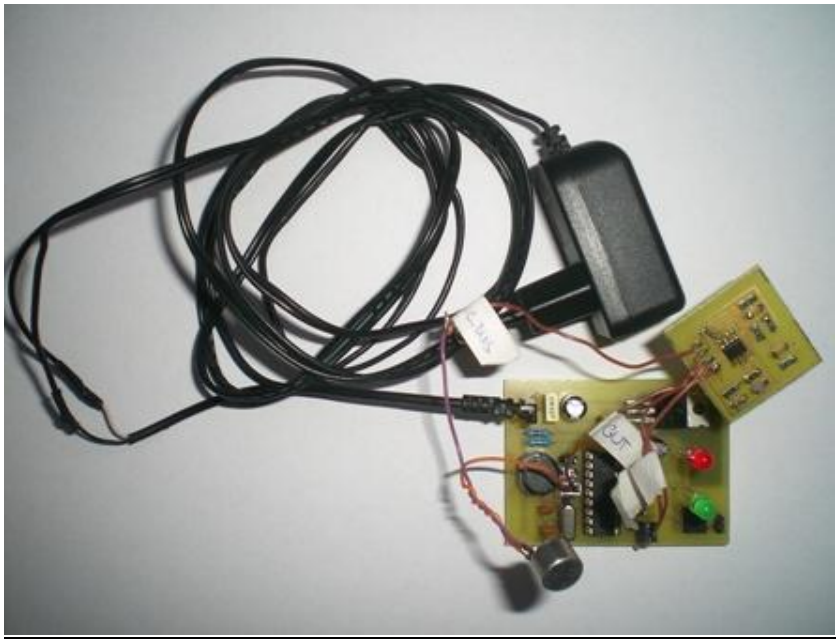
Fotodokumentace



Obrázek č. 1 první „zkušební“ deska s komunikací přes sériový port



Obrázek č.2 mikrofonní zesilovač(SMD konstrukce)



Obrázek č.3 funkční finální zapojení = bez krabičky



Obrázek č.4 uspořádání v krabičce



Obrázek č.5 provedení v krabičce

Tabulka tvořená pro 1000 vzorků(zde vypsáno pouze prvních a posledních 25)

n	Xn	s	Sprev	Sprev2	power	šum
1	11,51181	11,51181	0	0	0	0,765772
2	-0,65279	-2,09845	11,51181	0	132,5218	0,300274
3	-7,86535	-19,1136	-2,09845	11,51181	133,8916	0,978763
4	4,35068	8,849441	-19,1136	-2,09845	374,7714	0,931891
5	10,38865	28,39096	8,849441	-19,1136	422,4022	0,439041
6	-1,90267	-14,3175	28,39096	8,849441	915,9106	0,889287
7	-7,76983	-34,3628	-14,3175	28,39096	959,9897	0,639218
8	5,35661	23,9894	-34,3628	-14,3175	1447,576	0,269537
9	10,09113	41,4413	23,9894	-34,3628	1652,771	0,823926
10	-4,6647	-33,8583	41,4413	23,9894	2417,72	0,606578
11	-7,41506	-44,6044	-33,8583	41,4413	2687,562	0,145035
12	8,399285	47,85909	-44,6044	-33,8583	3325,596	0,776907
13	7,868623	46,46283	47,85909	-44,6044	4011,964	0,511576
14	-6,05992	-59,7539	46,46283	47,85909	4728,538	0,822604
15	-4,58449	-43,5434	-59,7539	46,46283	5380,664	0,64668
16	8,62433	73,84642	-43,5434	-59,7539	5793,297	0,090525
17	5,846443	40,1161	73,84642	-43,5434	6945,51	0,514453
18	-8,60272	-87,487	40,1161	73,84642	7434,621	0,222777
19	-2,09168	-31,2211	-87,487	40,1161	8822,526	0,794783
20	9,523275	100,931	-31,2211	-87,487	8971,74	0,006355
21	3,142135	21,68819	100,931	-31,2211	10766,1	0,327618
22	-9,18758	-112,842	21,68819	100,931	10932,34	0,317648
23	0,344307	-7,17304	-112,842	21,68819	12896,4	0,79882
24	10,46873	124,2117	-7,17304	-112,842	12886,47	0,244232
25	1,935333	-6,49026	124,2117	-7,17304	15368,12	0,967666
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
976	-9,26248	-9,26248	0	0	0	0,358893
977	1,774453	2,937645	-9,26248	0	85,79356	0,260561
978	10,92764	19,82121	2,937645	-9,26248	91,00628	0,552386
979	-1,29089	-6,71771	19,82121	2,937645	408,8226	0,598003
980	-8,03217	-27,0098	-6,71771	19,82121	421,2866	0,739198
981	4,79979	14,90941	-27,0098	-6,71771	797,441	0,559272
982	10,10924	35,24667	14,90941	-27,0098	901,2466	0,530486
983	-4,73645	-24,0722	35,24667	14,90941	1530,612	0,040545
984	-7,93457	-40,1582	-24,0722	35,24667	1715,246	0,254352
985	7,704364	36,81965	-40,1582	-24,0722	2313,552	0,913256
986	9,066602	44,60099	36,81965	-40,1582	2782,685	0,680735
987	-5,30659	-47,7273	44,60099	36,81965	3551,163	0,769443
988	-5,98357	-44,5909	-47,7273	44,60099	3999,818	0,430949
989	8,564139	61,89119	-44,5909	-47,7273	4533,505	0,429503
990	6,326631	43,14519	61,89119	-44,5909	5472,292	0,224389
991	-7,78948	-75,0989	43,14519	61,89119	6027,366	0,326897
992	-3,84255	-37,5567	-75,0989	43,14519	7094,449	0,487494
993	9,565568	89,38087	-37,5567	-75,0989	7404,55	0,258649
994	4,226043	30,55825	89,38087	-37,5567	8977,892	0,272399
995	-9,04841	-102,267	30,55825	89,38087	9265,749	0,23108
996	-2,12669	-19,8422	-102,267	30,55825	10999,85	0,180104
997	11,59346	116,3521	-19,8422	-102,267	11107,04	0,885296
998	1,322522	6,55307	116,3521	-19,8422	13641,59	0,034595
999	-9,08603	-126,261	6,55307	116,3521	13676,5	0,447121
1000	0,549509	9,852432	-126,261	6,55307	15880,89	0,274755

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Vizualizace AutoCAD 2007

Tomáš Podroužek, Zdeněk Bryndač

SPŠ sdělovací techniky
Panská 3, Praha 1

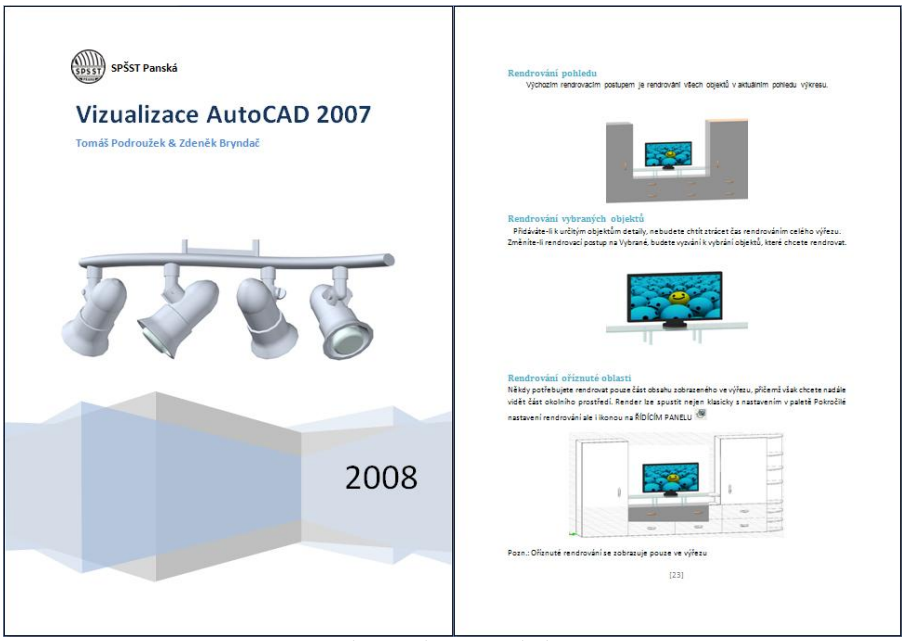
Tato práce vznikla jako dlouhodobá maturitní práce na SPŠST Panská. Cílem naší práce je seznámit uživatele a budoucí studenty naší školy s problematikou vizualizací v programu AutoCAD 2007. Součástí práce je učebnice v tištěné podobě, elektronická učebnice a 3D model bytu, který byl námi, autory vytvořen v AutoCADu 2007 a je použit pro názorné ukázky jednotlivých částí vizualizace.



Obr. 1: 3D byt



Obr. 2: Elektronická učebnice



Obr. 3: Tištěná učebnice

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

POHYBLIVÝ MODUL S ČIDLY

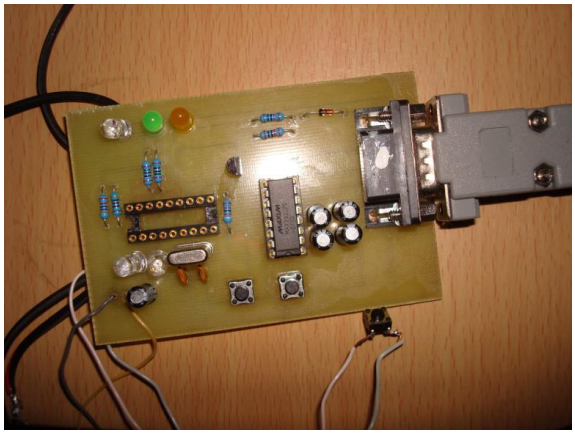
Jan Trauške, Roman Kuběnský
Střední průmyslová škola sdělovací techniky Panská
Panská 3, Praha 1

1. Úvod

Jedná se o malé vozítko na dvou kolech, které má za úkol rozpoznat překážku a vyhnout se jí. To je zajištěno pomocí dvou taktilních čidel, která jsou upevněna v přední části vozítka. Po setkání s překážkou čidlo tuto situaci nahlásí mikrokontroléru. Jako mikrokontrolér jsme použili dsPic 30F3012 od firmy Microchip, do kterého jsme naprogramovali kód, který zařídí, aby se vozítko vyhnulo překážce. Mikroprocesor jsme programovali programovacím jazykem C. DsPic pomocí můstkového budiče SN754410 řídí činnost motorů.

2. Programování

Na vložení kódu do dsPica jsme si vyrobili programovací zařízení, které zároveň sloužilo jako jednoduché testovací zařízení. Jako vývojové prostředí jsme použili program MPlab od firmy Microchip. Při programování jsme chod programu průběžně simulovali na led diodách, které jsme připevnili na programovací zařízení a vše bez problémů fungovalo. Do mikrokontroléru dsPic jsme nejprve museli ve školních dílnách nahrát firmware Bootloader, který později umožňoval pokračovat v programování doma. Jako software pro vkládání kódu do dsPica jsme vybrali program TinyBootloader, který je jednoduchý a zároveň výkonný.
Programovací zařízení se skládá se z osmnácti pinové patice na dsPica, integrovaného odvodu MAX232, který slouží pro přenos dat s PC do dsPica pomocí 4 žilového kabelu, krystalu s frekvencí 7,37Mhz, dvou tlačítek a tří led diod. Podrobné schéma je na obrázku č.2. Více informací najdete na stránkách ing. Šerých. Navíc jsme si pro simulaci připájeli na výstupní porty dsPica dvě led diody a na vstupní port jedno tlačítko. Při simulaci jsme použili tlačítka, která v našem případě nahrazovala čidla, a diody, které simulovaly chod motorů.



Programovací zařízení, obr.1

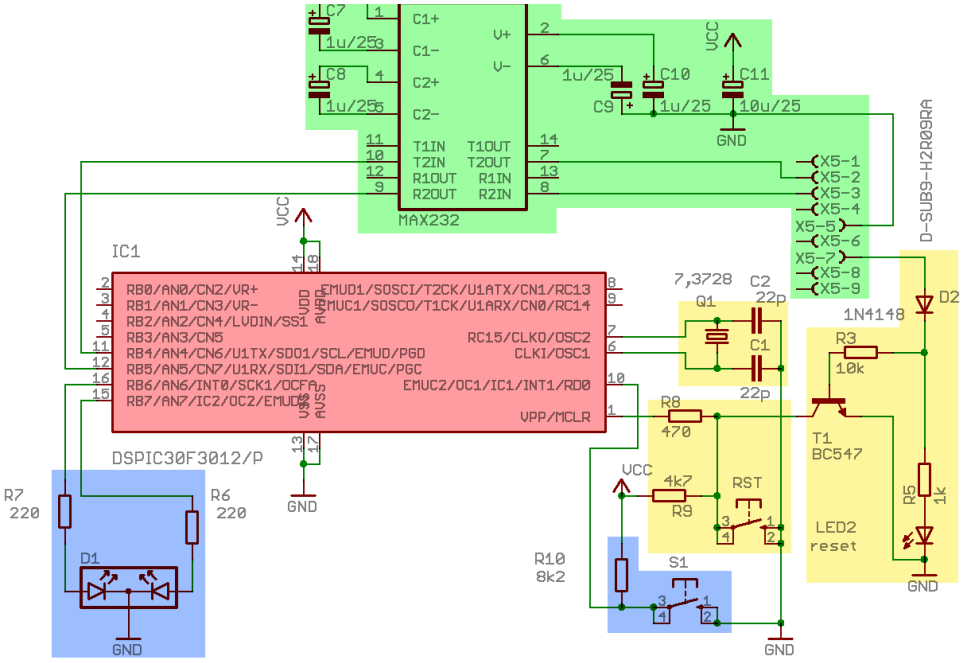


Schéma zapojení programovacího zařízení, obr.2

3.Konstrukce

Jako kostru pro naše vozítko jsme použili plastovou krabičku o rozměrech 110x80x30 mm. Do té bylo vyvrtáno několik děr pro umístění čidel, motorků, osvětlení, napájení, plošného spoje a tlačítka pro vypnutí a zapnutí.
Jako čidla slouží dva měděné dráty, připojené na vstupní porty dsPica a provlečené skrz uzemněné kovové spirálky.
Dále jsme použili motorky s převodovkou GM9, připájeli je k můstkovému budiči SN754410 a připevnily na ně dvě plastická kolečka. (Je důležité si dát pozor na výběr motorků, protože jak jsme sami zjistili, na některé typy nelze kolečka namontovat).
Plošný spoj jsme si navrhli v programu Eagle, desku vytiskli na průhlednou folii, osvítili ultrafialovým světlem a vyleptali. Po vyvrtání děr jsme plošný spoj natřeli kalafunovým potěrem a připájeli všechny součástky dle schématu obr.3. Desku jsme poté přišroubovali k plastové krabičce. Po přivedení napájení a připojení k motorkům jsme provedli první test. Při provádění testu se program zacyklil, načež jsme zjistili, že příčinou je rušení z motorů. Tento problém jsme řešili připájením několika kondenzátorů, a stabilizátoru LF00A. Napájení je tvořeno čtyřmi alkalickými bateriemi zapojenými do série.

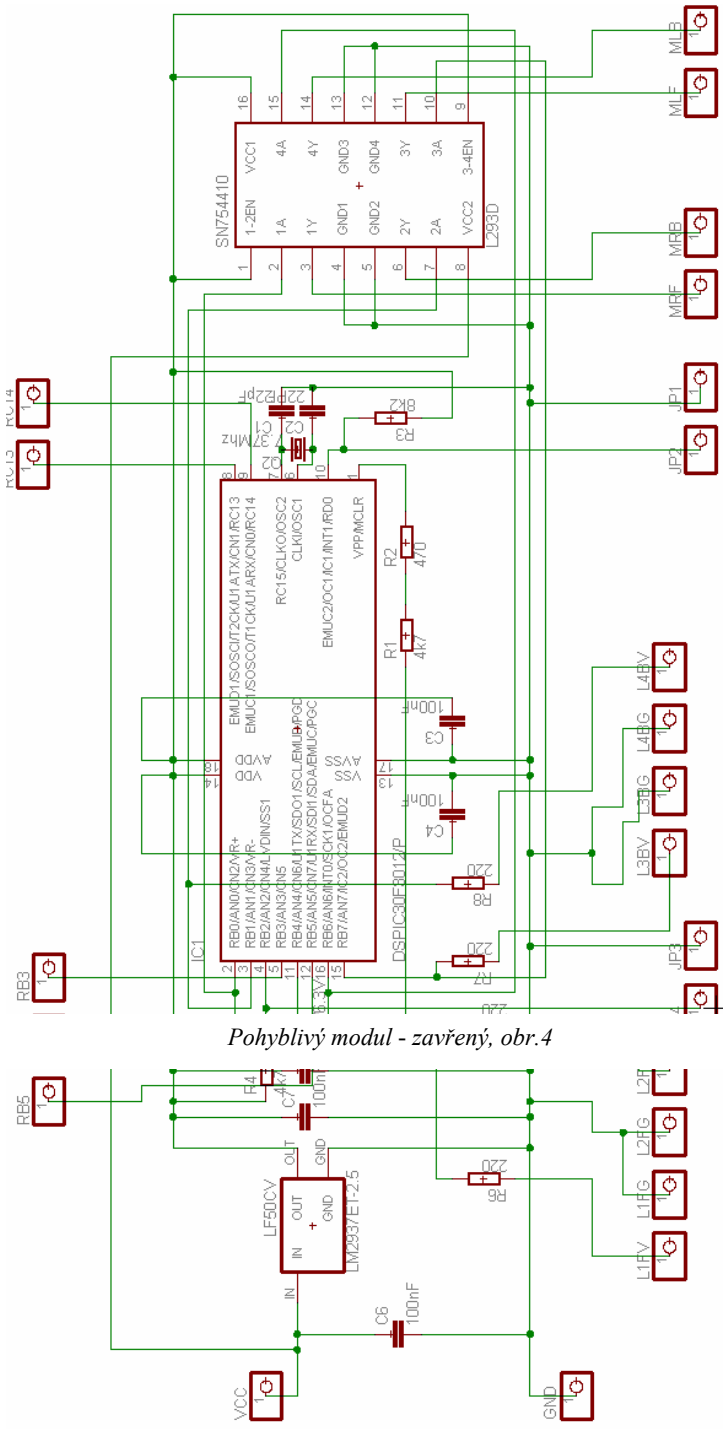


Schéma zapojení – obr.3

Pohyblivý modul - zavřený, obr.4

4. Použité součástky:

dsPIC30F3012
Mikrokontrolér dsPIC 30F3012 má 18pinů. Shodli jsme se, že to pro naši práci stačí s tím, že pokud by jsme v průběhu práce dostali ještě nějaké nápady, zbude nám pro jejich realizaci ještě několik pinů volných.

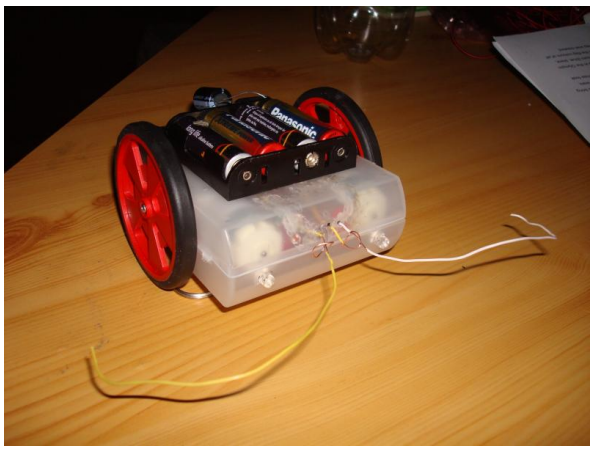
Můstkový budič SN754410
Budič SN754410 je čtyřnásobný vysokoproudový můstkový budič, který dokáže obousměrné řízení motorů do proudu až 1A a při napětí od 4.5V do 36V
Zařízení je navrženo k řízení indukčních součástek jako jsou relé, elektromagnety, stejnosměrné a bipolární krokové motory a jiné vysokoproudové/vysokonapěťové součástky. Ovládací prvky jsou aktivovány po dvou. Ovládání 1 a 2 je aktivovováno pomoci enable (EN) 1, 2 a ovládání 3, 4 pomocí enable 3, 4. Pokud je na enable vysoké napětí (H), příslušné ovládací prvky jsou povoleny a jejich výstupy jsou aktivní. Pokud je na enable nízké napětí (L), ovládání je vypnuté a výstupy jsou neaktivní. Oddělené napájení VCC1 slouží pro logické vstupy zapojení k minimalizaci plýtvání výkonu součástky. Napájení VCC2 je používáno pro výstupní obvody.

Stabilizátor LF33CV

Stejnosměrný motor GM9
Převodovka 143 : 1
Rychlost otáčení: 70 ot/min při 5 V
Napájecí napětí: maximálně 6 V
Odběr proudu: 58 mA bez zátěže, 670 mA při 0 ot/min
Točivý moment: 0,3 Nm (při 0 ot/min)
Hřídel: průměr 7 mm, dvojité zploštění
Rozměr: 70 x 27 x 23 mm
Hmotnost: 32 g

4. Závěr

Práce na vozítku byla pro nás velice zajímavá. Tak jak v průběhu programování a konstruování docházelo k různým nečekaným událostem, odhalovali jsme do největších podrobností všechna možná nepředvídatelná úskalí. Asi největším problémem bylo rušení motorů a indukce, která se nám tvořila na čidlech (čidla fungovala jako antény).



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

ENCYKLOPEDIE LÉTÁNÍ

Adam Sedláček, Martin Šedivý

Střední průmyslová škola sdělovací techniky
Panská 3, Praha 1

Tato práce vznikla jako dlouhodobá maturitní práce vznikla jako multimediální interaktivní počítačový program za účelem shrnutí fyzikálních principů létání a umožnit tak široké veřejnosti seznámení s leteckou problematikou a jejími fyzikálními principy.

Aby bylo možno dosáhnout cílů, které si práce kladla, museli se nejprve autoři, kteří se aktivně zajímají o letectví, detailněji seznámit s danou problematikou. To však nebyl jednoduchý úkol, protože současná vědecko-populární literatura, která se týká letectví, nabízí většinou pouze základní informace o fungování letadel a dále se pak zabývá pouze konkrétními letadly, ale nenabízí hlubší pohled do fyzikálních příčin létání a problematikou s letadly a létáním úzce spojenou. Ani v odborné literatuře není situace o mnoho lepší. Je v ní sice mnohem více informací než v literatuře vědecko-populární, avšak tyto informace jsou na tak odborné úrovni, že člověk, který nepracuje přímo v oboru, má problémy s pochopením i relativně jednoduchých principů. Z tohoto důvodu byl i tento druh literatury pro potřeby zpracování dlouhodobé maturitní práce Encyklopedie létání nevhodný. Proto se museli autoři poohlédnout po jiných zdrojích potřebných informací. Nakonec byly jako nejdůvěryhodnější a nejvhodnější zdroje informací vybrány anglická verze internetové encyklopedie Wikipedia, internetové stránky amerického úřadu pro letectví a kosmonautiku a publikace Encyklopedie letadel. Informace získané z těchto zdrojů byly následně přeloženy z anglického jazyka a převedeny do srozumitelné češtiny.

Provedení a grafické ztvárnění práce:

Dlouhodobá maturitní práce Encyklopedie létání byla vytvořena jako interaktivní multimediální počítačový program a to především z důvodu neustále se rozvíjející výpočetní techniky, která je již v dnešní době hojně rozšířená a stává se nejběžnějším nástrojem pro získávání informací. Dále pak také z důvodu snadné použitelnosti této dlouhodobé maturitní práce při výuce fyziky díky projekční technice, která se nachází ve většině učeben SPŠST.

Počítačová aplikace Encyklopedie létání byla vytvořena pomocí licencovaného programu Adobe Flash CS3. K volbě tohoto programu vedlo několik následujících důvodů. Zaprvé to byly vlastnosti vhodné pro zpracování této dlouhodobé maturitní práce a to především vzhledem k možnosti vytvoření efektního grafického prostředí aplikace. V neposlední řadě byla pro autory podstatná funkce přehrávání vloženého videa.

Aplikace Encyklopedie létání byla vytvořena tzv. framovou metodou, tzn., že jednotlivé kapitoly jsou utvořeny jako skupina několika po sobě jdoucích snímků, mezi kterými se pomocí tlačítek vpřed a vzad přepíná.

Grafické prostředí aplikace bylo uzpůsobeno pro co nejpřehlednější orientaci v programu, aby i zájemci, kteří nejsou počítačovými znalci, mohli tuto aplikaci bez problémů využívat. Menu aplikace je vytvořeno jako kokpit letadla, kde jednotlivé přístroje jsou tlačítka, při jejichž zmáčknutí dojte k přesunu na požadovanou kapitolu. V jednotlivých kapitolách je pak

uživatelské prostředí řešeno klasicky, kdy po levé straně jsou umístěna tlačítka pro přechod mezi jednotlivými částmi kapitoly. V každé části kapitoly se pak nachází text vhodně doplněný názornými 3D animacemi, případně obrázky, které se významnou měrou podílejí na objasnění dané problematiky.

Obsah práce:

Obsahem této dlouhodobé maturitní práce je popis a vysvětlení základních fyzikálních principů příčin létání a problematiky s létáním a s letadly spojenými. Zároveň pak také přiblížení historie tohoto technického odvětví, které je nedílnou součástí fungování dnešní ekonomiky a ozbrojených sil.

V kapitole Historie létání jsou vylíčeny dějiny letectví, které začaly již před dávnou dobou představami, že by člověk mohl létat jako pták. Jako prvním se podařilo splnit tento sen bratřím Wrightovým, kteří dne 17. prosince 1903 poprvé vzlétli se svým letadlem těžším než vzduch. Dále jsou v této kapitole popsány další mezní úspěchy, kterých dosáhli piloti se svými letadly, ať to byl Louis Blériot, který jako první přeletěl kanál La Manche nebo Charles Lindberg se svým strojem Spirit of Saint Louis, který jako první překonal Atlantský oceán. Kapitola se také zabývá současnými letadly, které se díky své výjimečnosti zapsaly do letecké historie.

Kapitola Jak letadlo funguje se zabývá fyzikálními příčinami létání a objasňuje tak, jak je možné, že letadlo létá. Je zde popsána vztlaková síla jako jedna z hlavních příčin letu letadla, dále pak ovládání letadla, funkce křídel a nakonec i princip fungování leteckých motorů. Všechna témata dané kapitoly jsou doprovázeny názornými 3D animacemi, díky nímž je pochopení daných principů podstatně jednodušší.

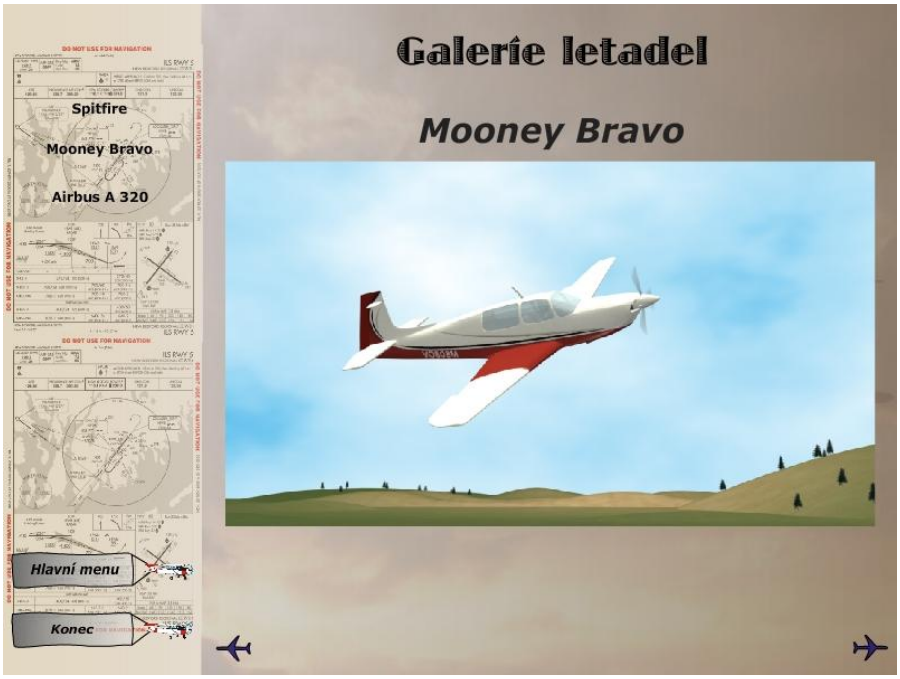
Další částí Encyklopedie létání je kapitola Letecké přístroje, která popisuje princip měření veličin, které jsou pro bezpečný a kontrolovaný let naprosto nezbytné. Také seznamuje s leteckými přístroji jako takovými.

Kapitola Letecká navigace seznamuje s technikami používanými k orientaci pilota ve vzduchu, které jsou nepostradatelné v letecké dopravě. Konkrétně se jedná o GPS navigaci, která dnes vytlačuje dříve hojně používanou navigaci rádiovou.

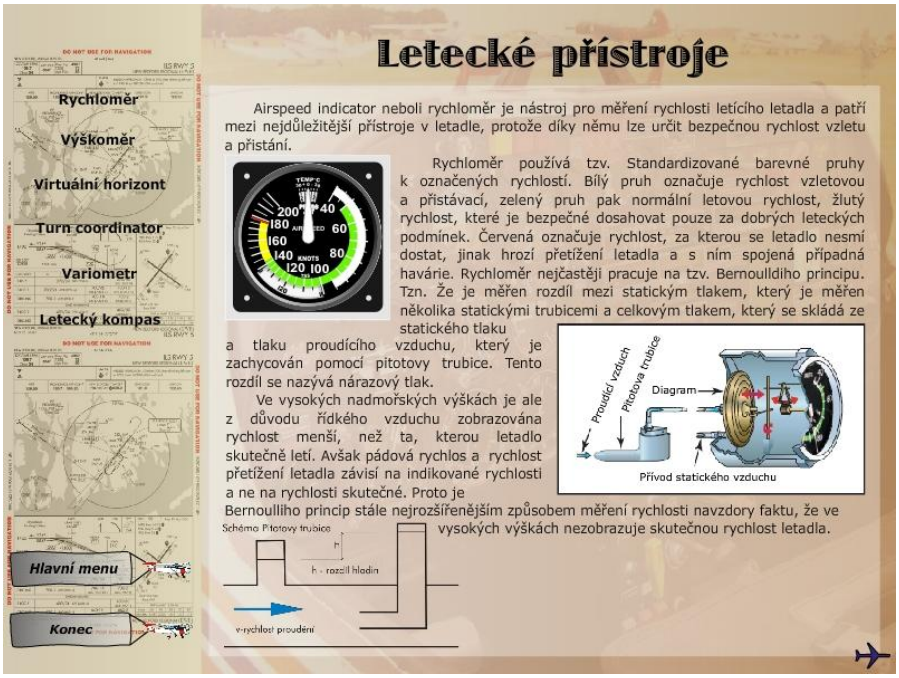
Poslední kapitolou dlouhodobé maturitní práce Encyklopedie létání je Galerie letadel, ve které jsou obrázky-rendery letadel, které jsme modelovali pro použití v 3D animacích.



Obrázek 1: Jak letadlo funguje-Ovládání letadla.



Obrázek 2: Galerie letadel-Mooney Bravo.



Obrázek 3: Letecké přístroje-Rychloměr.

Středoškolská technika 2008

Setkání a prezentace prací středoškolských studentů na ČVUT

MISTŘI CHEMIE

Marek Šmolík, Martin Kovářík

SPŠST Panská 3
Panská 856/3, Praha 1

Tato práce je učební pomůcka pro studenty středních škol. Jejím výsledkem by měla být interaktivní aplikace pomáhající s výukou chemie. Její hlavní výhody jsou vizualizace učení a zábavný přístup. Výstupem práce je CD se soubory hry (a veškerými podklady), která je dotvořena do verze 0.8A.

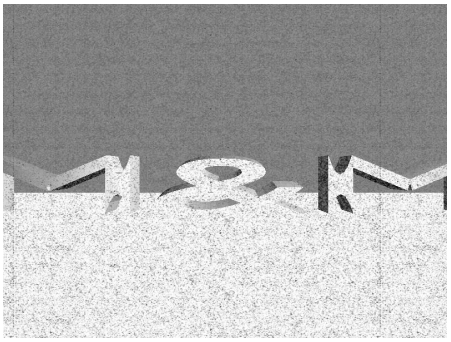
Příběh

Hráč se dostane do role Jana, hocha z malé vesnice uprostřed chemiků-plného kraje. Janova maminka hráči sdělí příběh o jejich vesnici. Jsou to chudí farmáři a už farmaří celá léta. Celá vesnice je závislá na jejich malém poli a to přestalo nedávno z neznámých důvodů nést úrodu. Nikdo ve vesnici není schopen záhadu objasnit a tak poprosí svého syna Jana, aby se vydal do světa, hledat pomoc u velkých Mistrů Chemie. Jan si bude (hráčovou rukou pochopitelně) moci sám zvolit, jestli půjde pomoci své vesnici, nebo ne. Potom co se rozhodne, že půjde, je matkou poslán za starším mužem na konci jejich vesnice. Ten muž se jmenuje Antoine a hráč se až na samém konci hry dozví, že jde o Antoina Laurenta Lavoisiera, který mu vysvětlí, jaký problém vlastně má jejich vesnice. (Tuto část už ale verze 0.8A neobsahuje). Antoine vezme Jana na povoz a odjíždí společně za profesorem Bacarem. Je to vážený chemik v jejich kraji a jako jediný ze všech je ochoten se s Janem, jakožto s člověkem „neznalým chemie“ bavit. Antoine se s Janem rozloučí u Bacarova domu a odjíždí vstříc městu Výdov nedaleko. Jan přivítán milým chemikem vstupuje do světa chemie a ještě ani netuší, jak daleká cesta ho vlastně čeká.

Zpracování hry

Vizuálně se hra nemůže rovnat s komerční elitou v herním světě, ale na možnosti studentů je projekt na celkem vysoké grafické úrovni.

Pro představu je zde přiloženo několik ukázkových záběrů ze hry.



Logo Teamu M&M



Logo Blenderu



Úvodní scéna hry



Hlavní Menu



Obrázek ze hry



Další obrázek ze hry

Středoškolská technika 2008

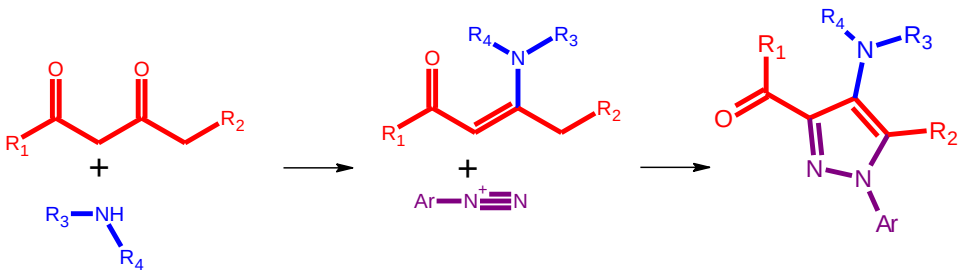
Setkání a prezentace prací středoškolských studentů na ČVUT

SYNTÉZA A CHARAKTERIZACE FLUOROVANÝCH DERIVÁTŮ 3-ACYL-4-AMINO-1-ARYLPYRAZOLŮ

Radim Horák

Střední průmyslová škola chemická
Na Třísle 135, Pardubice

Náplní práce byla syntéza 5-substituovaných 3-acyl-4-amino-1-arylpyrazolů obsahujících atomy fluoru v různých polohách a jejich charakterizace pomocí elementární analýzy, fyzikálních konstant, ^{13}C a ^{1}H NMR spektroskopie. Látky byly připraveny nově objevenou metodou (objevila skupina VLADIMÍRA MACHÁČKA na Univerzitě Pardubice), která vychází z β-enaminonů a arendiazonium-tetrafluoroborátů. Tato syntéza probíhá za normálního tlaku a při laboratorní teplotě. Její největší výhoda je však práce bez toxických derivátů hydrazinu, které jsou dosud asi nejvíce využívány pro syntézu pyrazolových skeletů.
Substituované pyrazoly jsou využívány v mnoha odvětvích jako biologicky aktivní látky (pesticidy, léčiva) nebo například barviva, katalyzátory.
V rámci této práce bylo připraveno 8 nových látek. Z toho pět vysoce substituovaných pyrazolů s potenciální biologickou aktivitou a tři enaminony. V jednom případě bylo strukturně modifikováno léčivo antipyrin (již se však nepoužívá).
Tento postup přípravy pyrazolů bude publikován v mezinárodním vědeckém časopise.



Obr. 1: Syntetické schéma

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Vyhodnocení vlivu skládky odpadů na povrchové a podzemní vody

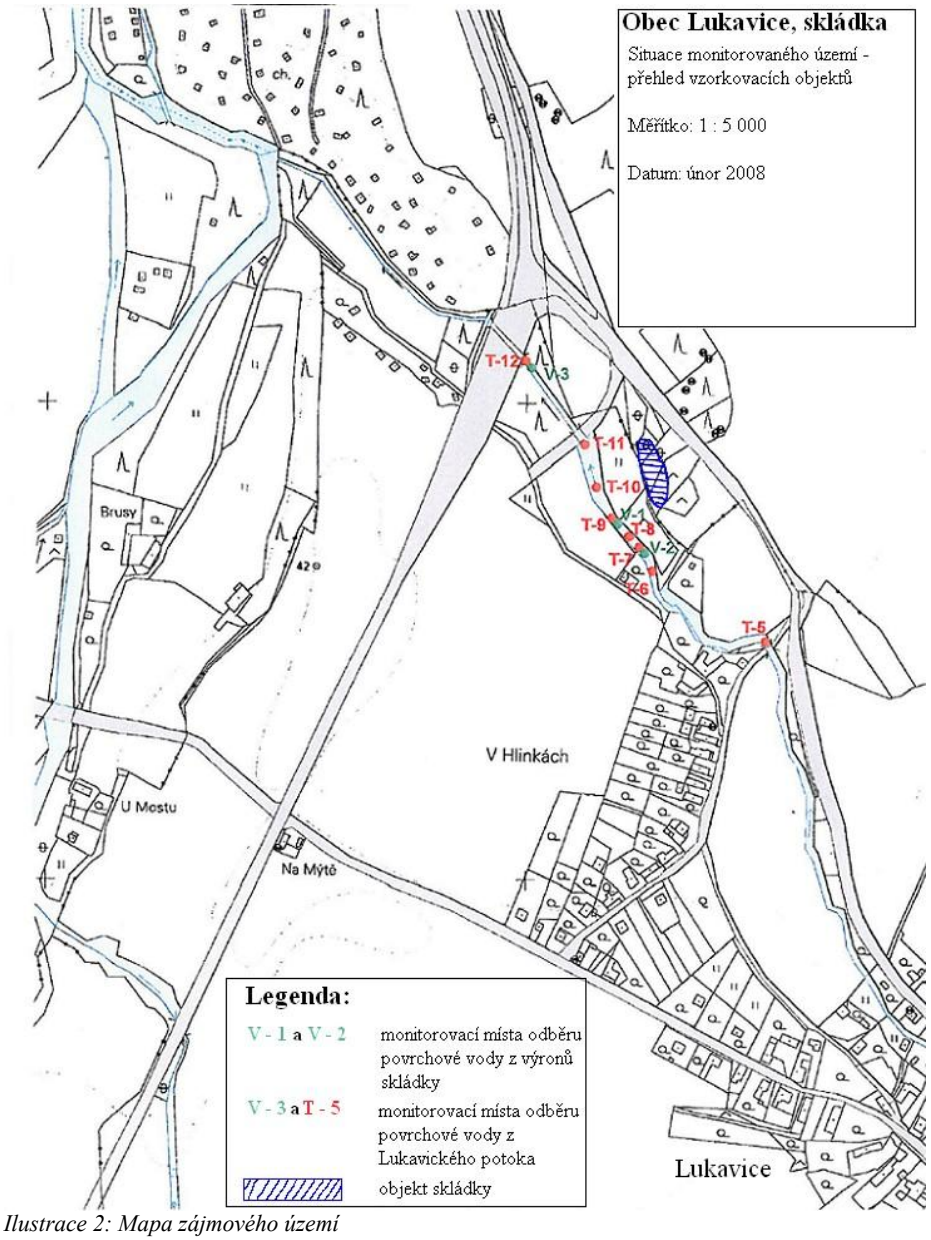
Jindřich Mašín

Střední průmyslová škola chemická Pardubice
Na Třísle 135, Pardubice

Cílem práce je monitorovat povrchové vody vytékající ze skládky odpadů na lokalitě Lukavice a vody v recipientu Lukavický potok v kvalitativních parametrech nepolární extrahovatelné látky (dále jen NEL) a kovy – olovo, rtuť, kadmium, měď, arsen a zinek. Návazně na získané údaje o znečištění povrchových vod a převzatých údajů o znečištění podzemních vod vyhodnotit vliv skládky na šíření kontaminace do okolí skládky.



Ilustrace 1: Pohled na těleso skládky a lukavický potok



Ilustrace 2: Mapa zájmového území

Příklad výsledeku rozboru vody ze skládkového výronu

Zkouška	Hodnota	Jednotka	Vzorek ze dne
Cd	0,0009	[mg.l^{-1}]	23.1.2007
Cd	0,00544	[mg.l^{-1}]	20.2.2007
Cd	0,004	[mg.l^{-1}]	4.4.2007
Cd	0,0046	[mg.l^{-1}]	3.5.2007
Cd	0,0091	[mg.l^{-1}]	6.6.2007
Cd	0,0025	[mg.l^{-1}]	29.6.2007
Cd	0,0029	[mg.l^{-1}]	21.8.2007
Cd	0,0015	[mg.l^{-1}]	4.10.2007
Cd	0,0028	[mg.l^{-1}]	29.11.2007

Červeně jsou vyznačeny nadlimitní hodnoty obsahu Cd.

Závěr práce

Odběrem vzorků, jejich analýzou, vyhodnocením výsledků a jejich porovnáním s platnou legislativou bylo zjištěno, že lukavická skládka je trvalou ekologickou zátěží.

Měření pH na místě ukázalo, že vody vytékající z těla skládky jsou extrémně kyselé. Z výsledků měření před a za skládkou také vyplývá, že se voda po průtoku územím skládky okyselí tak, že nesplňuje imisní limit pH.

Imisní limity byly překročeny i u NEL a TK, v některých případech několikanásobně. V případě TK a NEL už však situace není tak jasná jako u pH. V jistých obdobích byly totiž koncentrace některých znečišťujících látek vyšší v toku nad skládkou než pod ní. Nad příčinou tohoto jevu lze jen spekulovat – nejpravděpodobnější příčinou je sorpce sledovaných látek na materiál tvořící koryto potoka a dále průběžná sedimentace a zpětné uvolňování sedimentů v závislosti na oscilaci pH a průtoků v Lukavickém potoce.

Za celou dobu monitorování skládky nebylo zjištěno překročení imisního limitu As a Hg. Z hlediska překročení imisních limitů bylo dominující Cd. To překročilo imisní limity ve výronu V 2 takřka ve všech případech, v bodu odběru V 3 asi v polovině případů. Z hlediska kvantity byla ve sledovaných vzorcích nejvíce obsažena Cu a Zn. Tyto kovy přesáhly imisní limity ve výronech ve většině případů. Cu nesplňovala asi v polovině případů imisní limity ani v odběrovém bodě V 3. Zn však v tomto místě imisní limity nesplňoval pouze ve dvou případech. Ve výronech byl významným polutantem také Ni, který nesplňoval imisní limity asi v polovině případů. V odběrovém bodě V 3 však nebyl jeho imisní limit překročen ani jednou. Imisní limity olova byly překročeny jen v několika málo případech, v odběrovém bodě V 3 pouze jednou. Přestože výskyt NEL ve výronech nebyl zvlášť vysoký, byly imisní limity NEL v odběrovém bodě V 3 překročeny ve ve většině případů. Toto je dáno také vysokým obsahem NEL před vtokem na území skládky (před odběrovým bodem T 5). pH vod z výronů nesplňuje imisní limit v naprosté většině případů. Voda z těchto výronů ve většině případů okyselí vodu v potoce tak, že nesplňuje imisní limity v odběrovém bodě V 3.

I když jsou v kapitole výsledky uvedeny srážkové úhrny naměřené v hydrometeorologické stanici v Seči, nelze je přímo dávat do souvislosti s množstvím znečišťujících látek v Lukavickém potoku nebo ve skládkových výronech, přestože srážky jsou pravděpodobně nejdůležitějším faktorem ovlivňujícím množství znečišťujících látek a pH povrchových vod Lukavické skládky.

Tato práce poskytuje pouze velmi omezený přehled informací o dopadech ukládání odpadů na Lukavické skládce. Na základě poznatků z této práce a dříve provedených

průzkumů by se nechalo navázat např. sledováním výskytu a koncentrace znečišťujících látek v podzemních vodách, půdě, organismech, etc.

Analýza rizik zpracovaná v roce 2006 odbornou firmou. doporučuje skládku před její rekultivací sanovat. Výsledky této práce potvrdily nutnost sanace skládky, neboť bezprostředně ovlivňuje jakost povrchových vod. Je však nutno mít na zřeteli, že kvalitu povrchové vody v potoce významným způsobem ovlivňují i důlní vody vytékající ze starých dolů v obci Lukavice. Aktuální kvalita podzemních vod není v této práci zahrnuta, protože výsledky jejího monitorování nemohou být k datu vypracování SOČ uveřejněny z důvodů probíhajícího výběrového řízení společnosti, která bude pověřena sanací skládky.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

RAK JAKO BIOINDIKÁTOR ČISTOTY VOD

Petr Tichý, Hana Adámková, Zuzana Kotíková, Kristýna Šimáková

Střední průmyslová škola chemická Pardubice
Na Třísle 135

ŠKOLNÍ MEZINÁRODNÍ PROJEKT – COMENIUS

Cíl projektu:

- Monitorování výskytu raků a dalších bioindikátorů čistoty vod
- Sledování stavu životního prostředí vybraných lokalit v okolí Pardubic
- Sledování čistoty našeho životního prostředí se stavem přírody v zemích našich partnerů
- Chemické analýzy vod ve školních laboratořích
- Tvorba výukových programů a jejich využití při hodinách ekologie a biologie
- Tvorba prezentačních CD
- Zdokonalování jazykových dovedností a seznámení se školstvím a s kulturním dědictvím zemí našich partnerů



V současné době si lidé na naší planetě stále více uvědomují svou zodpovědnost za zachování života na Zemi a udržení stability životního prostředí. Negativních faktorů , kterými člověk zasahuje do přírody, je velké množství a bylo by velkou chybou je neustále přehlížet. Člověk by měl včas reagovat na neobvyklé změny, které nám příroda v současné době přináší .

Naše škola vypracovala projekt , který se zabývá stavem vodních ekosystémů a životním prostředím, nejen v našem blízkém okolí , ale také v zemích našich partnerů . Na projektu spolupracujeme se zahraničními partnery z Finska , Itálie a Maďarska.Vodní ekosystémy patří mezi nejohroženější oblasti životního prostředí a voda je nezbytnou součástí života živých organismů.Toxické látky nebo těžké kovy, které se do vodních nádrží a toků dostávají, zasahují do potravních řetězců a negativně ovlivňují mnoho živočišných a rostlinných druhů včetně člověka. Řeky, jezera, vodní nádrže, ledovce a mokřady obsahují sladkou vodu, jejíž množství na Zemi představuje nízké procentuální zastoupení. Člověk je schopný tuto vodu čistit, upravovat ji, ale také velmi znečišťovat.

V našem projektu tedy sledujeme právě tento ekosystém, stav vodního prostředí pro bioindikátory, ukazatele její čistoty, a podmínky pro život dalších organismů. Symboly čistoty vod se nám staly ohrožené druhy raků, jejichž životní prostředí sledujeme a budeme sledovat po dobu tří let práce na projektu. Součástí práce na projektu jsou také chemické analýzy odebraných vzorků vody z různých lokalit, které provádíme ve školních laboratořích. Společně s našimi partnery provádíme analýzu a hodnotíme výsledky, které budou charakterizovat stav životního prostředí v severní, střední a jižní Evropě. Rádi bychom poukázali na aktuální negativní faktory působící na vodní biotopy a seznámili veřejnost s možnostmi jejich řešení vedoucích ke zlepšení kvality čistoty vod.

Práce na projektu byla zahájena v září roku 2006 a bude probíhat do června roku 2009. První schůzka se uskutečnila v Pardubicích ve dnech 16.–19. října 2006. Cílem první schůzky bylo seznámit účastníky projektu s výstupy a produkty, které by měly být výstupním materiálem. Součástí první schůzky byla návštěva zajímavých lokalit, kde jsme zaznamenali výskyt chráněných raků a dalších vodních bioindikátorů.Vybrané lokality nadále sledujeme a výsledky budou zahrnuty v závěrečném dokumentu. Ve škole jsme vytvořili tři studentské skupiny, které provádějí ve školních laboratořích chemické analýzy, kde sledují výskyt minerálních prvků, dusičnanů, biologickou a chemickou spotřebu kyslíku a speciálními přístroji zaznamenávají přímo na lokalitách pH, salinitu a teplotu vody v různých ročních obdobích.

Úzce spolupracujeme s panem Dolanským, zoologem Východočeského muzea, který nás průběžně informuje o lokalitách s možným výskytem raků. Monitorovací mapy a listy jsme získali od paní Moniky Štambergové z Agentury ochrany přírody a krajiny v Praze, která se specializuje na výskyt raků v České republice.

Výsledky naší práce z prvního roku jsme prezentovali na druhé mezinárodní schůzce, která se uskutečnila ve dnech 5.–11. května 2007 ve Finsku v regionu Isojoki. Na této schůzce jsme mohli představit zajímavé lokality našeho blízkého okolí a porovnat je se stavem životního prostředí v okolí Isojoki. Na schůzce jsme se zabývali způsoby ochrany přírody a vodních toků a negativními faktory, které ovlivňují výskyt vodních bioindikátorů. V terénu jsme prováděli základní měření kvality vodních toků a jezer, které můžeme porovnávat s našimi výsledky. V průběhu schůzky jsme s našimi partnery navštívili národní parky, které se nacházejí v okolí Isojoki a poznávali jsme zajímavé živočichy a rašeliništní společenstva různých vzácných druhů rostlin. Měli jsme možnost sledovat postupné změny krajiny způsobené ustupujícím ledovcem.

První rok práce na projektu byla velmi zajímavá a podněcující k další činnosti. Velmi aktivní byli naši studenti, jejich učitelé a také naši zahraniční partneři. Výstupním materiálem byly naše prezentace a výukový program, který lze použít při hodinách biologie a ekologie. Výsledkem bylo schválení projektu a práce tak může pokračovat také v dalších letech.

15.–18.října jsme jako koordinátoři projektu uspořádali další, již třetí partnerskou schůzku, jejíž hlavním tématem bylo představit formou prezentací národní parky naší země a zemí našich partnerů. Učitelé a studenti z partnerských zemí také informovali o průběhu projektu ve svých školách. Součástí programu této schůzky byla i návštěva čističky odpadních vod u města Skutče a návštěvy zatopených lomů a toků s výskytem vodních bioindikátorů.

V květnu 2008 proběhne čtvrtá mezinárodní schůzka, která se uskuteční v Itálii na Sicílii, ve městě Messina. Schůzky se zúčastní čtyři ze studentů, kteří se aktivně podílí na monitorování výskytu bioindikátorů a velmi pečlivě plní zadané úkoly. Hlavním tématem schůzky je sledování negativních faktorů působících na vodní bioindikátory čistoty vod. Všichni partneři si postupně připravují materiály a tvoří prezentace právě k tomuto tématu. V plánu je také setkání účastníků projektu v Maďarsku.

Vzhledem k mezinárodní účasti na projektu je pracovním jazykem angličtina, studenti připravují v anglickém jazyce prezentace, připravují a překládají potřebné materiály. Do projektu se tak zapojují studenti různých ročníků, učitelé biologie, chemie i jazyků. Na mezinárodních schůzkách zazní i slova ve finštině, maďarštině a italštině. Angličtina přestává být pouhým vyučovacím předmětem, ale stává se nástrojem komunikace a porozumění.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

GIRASOL - slunečnice reagující na dopad sluneční energie

Autor:
Jiří Kuřátko
SPŠE a VOŠ, Karla IV. 13531 69 Pardubice, 3. ročník

Konzultant práce:
Ing. Ladislav Štěpánek
(SPŠE a VOŠ, Pardubice)

Pardubice, 2008

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací."

Zadání písemné práce

- Navrhněte a sestrojte zařízení z fotovoltaických článků reagujících na změnu pohybu a intenzity slunečního záření
- Problém vyřešte softwarově včetně mechanické konstrukce

Anotace

Hlavní částí této práce je solární panel uchycený na pohyblivém kloubu, který se otáčí za zdrojem světla. Girasol navíc měří i některé fyzikální veličiny jako například :

- Teplotu
- Výkon solárních článků
- Tlak

Naměřené hodnoty dále graficky zpracovává a vyhodnocuje na displeji počítače nebo graficky na papíru.

Obsah

Kapitola
1. Úvod
1.1 Předmluva
1.2 Vznik a vývoj solárního článku
1.3 Funkce solárního článku
2. Technický popis navrhovaného zařízení
2.1 Popis navrhovaného řešení
2.2 Technické parametry
2.3. Blokové schema
2.4 Schema zapojení
2.5 Funkce a popis zapojení
2.6 Použité integrované obvody
2.6.1 Mikroprocesor PIC 18f8720.
2.6.2 Převodník TLV2548
2.6.3 Tlakový senzor MPX4115
2.6.4 Převodník USB – RS232 FT232BM
2.6.5 konvektor LT1173 je DC/DC
3. Závěr
4. Seznam použité literatury
5. Přílohy

1. Úvod

Předmluva

Lidstvo na celé Zemi spotřebuje každou vteřinu 10 TW (terrawattů), což je 10 000 000 000 000 wattů. Z čeho tuto energii vlastně získáváme? Po zralé úvaze dojde většina z nás zřejmě k názoru, že převážně z fosilních paliv. Ale dnes je už jasné, že tyto zdroje energie nemají žádnou budoucnost. Za prvé jsou nečisté, drahé, znečišťují životní prostředí a za druhé jich velmi rychle ubývá. Dnešním tempem vyčerpáme ložiska fosilních paliv za velmi krátkou dobu. Už je opravdu na čase poohlédnout se po jiném zdroji energie. Dnes jsou již vyvinuté technologie získávání energie z jiných zdrojů, kterým se obecně říká alternativní zdroje energie.

Jeden z těchto zdrojů se nachází přímo nad námi a vidíme ho takřka každý den. Je to Slunce. A tímto alternativním zdrojem energie bych se v této práci chtěl zabývat.

Slunce je základem všeho života na Zemi. Teplo a energie ze Slunce umožnily vznik a vývoj života. Sluneční paprsky dávají 20 000krát více energie než potřebujeme. Kdybychom tuto energii uměli využít, byly by všechny naše energetické problémy vyřešeny. I když sluneční energie byla lidstvu dostupná už v pravěku, dosud jsme se ji nenaučili efektivně využívat jako jiné zdroje. Vezměme např. solární nebo-li fotovoltaické články sloužící k přímé výrobě elektrické energie ze slunečních paprsků. Jejich účinnost se běžně pohybuje kolem 12 - 15 %. Porovnáme-li tuto účinnost s využitím vodní energie, pak jsme na tom podobně jako Římané a jejich vodní kola před 2 000 lety. . .

1.2 Vznik a vývoj solárního článku

Historie solárního článku se začala datovat rokem 1839, kdy francouzský experimentální fyzik Edmund Becquerel při pokusech s 2 kovovými elektrodami umístěnými v elektrovodivém roztoku zjistil, že při osvícení zařízení vzrostlo na elektrodách napětí: fotovoltaický efekt byl na světě. V roce 1877 byl objeven fotovoltaický efekt na selenu (W. G. Adams a R. E. Day) a vyroben první článek.

Důležitým krokem v historii byl objev způsobu růstu monokrystalu křemíku polským vědcem Czochralským v roce 1918. Přestože byl fotovoltaický efekt postupně objeven i u jiných prvků (sirník kadmia, oxid mědi), křemík se ukázal jako nejvýhodnější. Za vynálezce křemíkového solárního článku bývá označován američan Russel Ohl (1941). Patent na "převaděč solární energie" dostali však 5. března 1954 D. M. Chapin, C. S. Fuller a G. L. Pearson, kteří o měsíc později předvedli křemíkové solární články s účinností 4,5 % a později 6 %.

Další vývoj solárních článků urychlilo rozhodnutí použít je jako zdroj energie pro družice Země. Účinnost článků postupně stoupá, v roce 1958 dosáhla 9 %. To už je postačující na to, aby první umělá družice Spojených Států (Vanguard I), vypuštěná 17. března téhož roku, mohla být takovýmito články napájena (0,1 W, rozměry cca 100 cm^2, družice pracovala 8 let). Explorer VI,vypuštěný o rok později, měl už solárních článků 9 600 (1 x 2 cm). V roce 1960 se účinnost článků vyhoupla na "neuvěřitelných" 14 %. A tak první telekomunikační družice, legendární Telstar, mohla být zásobena zdrojem o 15 W.

I když se v této době solární články již vyrábějí komerčně, vzhledem k jejich závratné ceně se používají jen výjimečně. Příkladem může být instalace 242 W panelu pro napájení majáku v Japonsku, v té době (1963) největšího na světě.

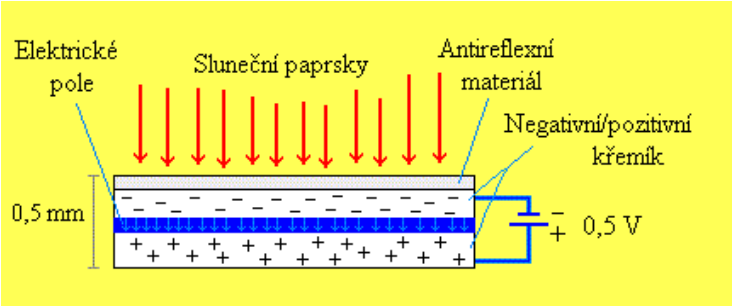
K výraznému zlevnění solárních článků nepřispěla ani naftová krize v 70. letech, ačkoliv ostatním alternativním zdrojům energie (využití větru, ohřev vody) pomohla. Také účinnost

běžných článků zůstala na 12 - 15 %. V Japonsku sice byly vyvinuty články s účinností přesahující 30 %, jejich výrobní cena je však příliš vysoká.

A budoucnost? Pro člověka snad levná a snadno použitelná fotovoltaická fólie, pro lidstvo modernizace myšlenek Nikoly Tesly o bezdrátové přenášení elektrické energie: vždyť nad zemskou atmosférou dopadá na každý čtvereční metr 10x více energie než na zemi.

1.3 Funkce solárního článku

Jak už bylo uvedeno, využívají tzv. fotovoutaického jevu. Je to jev, při kterém se v látce působením světla (fotonů) uvolňují elektrony. Tento jev může nastat v některých polovodičích. Fotovoltaický článek je nejčastěji tvořen tenkou destičkou (0,5 mm) nařezanou z monokrystalu křemíku (dnes se používá i levnější polykrystalický materiál). Každá strana destičky je obohacena atomy vhodných prvků tak, aby jedna byla kladná a druhá záporná. Když na destičku dopadnou fotony, uvolňují se záporné elektrony a po nich zbývají kladně nabyté "díry". Přiložíme-li na obě strany elektrody a spojíme je drátem, začne protékat elektrický proud. Jeden cm^2 dává proud kolem 12 mW. Jeden m^2 tak může dát až 150 W stejnosměrného proudu. Solární články můžeme zapojovat, jako každé jiné, buď za sebou (sériově), abychom dosáhli potřebného napětí (na jednom článku je asi 0,5 V), nebo vedle sebe (paralelně), abychom získali větší proud.



Princip solárního článku

2. Technický popis navrhovaného zařízení

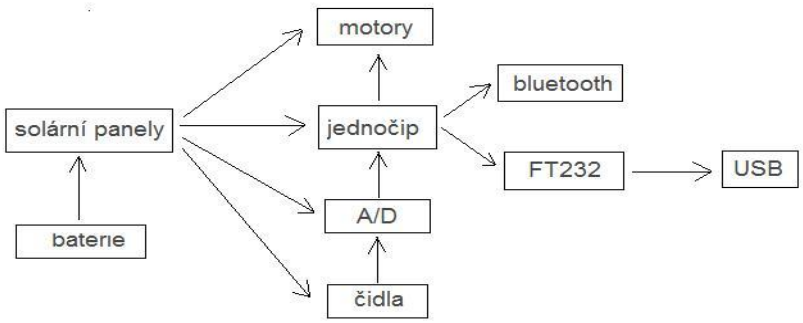
2.1 Popis navrhovaného řešení

Můj výrobek využívá k získání energie fotovoltaický články. Ty se pro získání maximální energie otáčí ke směru světla. Vzhledem k miniaturizaci se toto zařízení nedá efektivně využít, protože získaná energie je velmi malá (při denním osvětlení cca. 900mW). Navíc téměř jedna třetina získané energie je využita na provoz řídící elektroniky a čidel. Ta jsou řešena tak, že se dají pro snížení spotřeby vypnout.

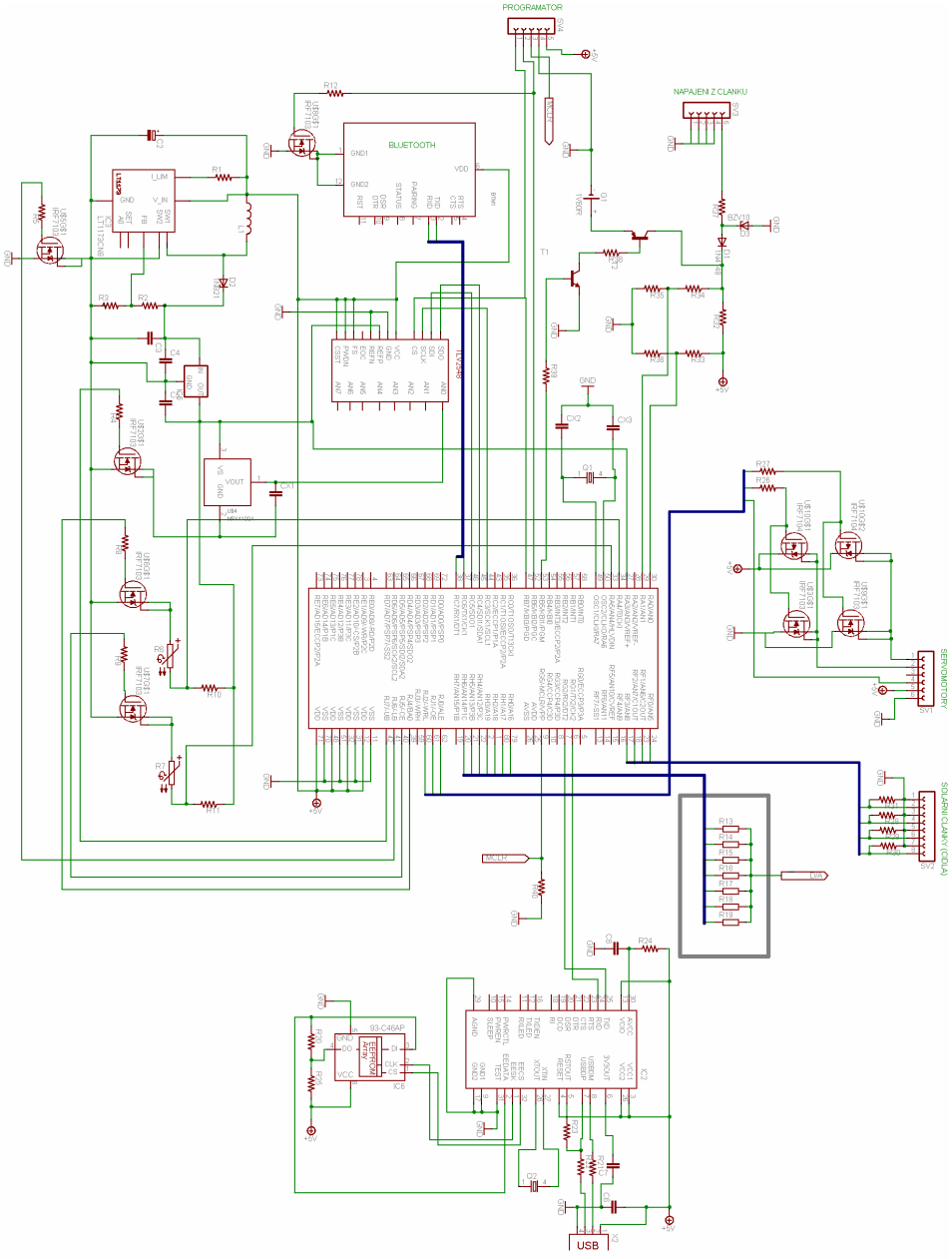
2.2 Technické parametry

Napětí na článcích	4-5 V (podle osvětlení)
Odběr řídící elektroniky	30 mA (při zapnutí všech čidel)
Teoretický výkon článku	2 W
Skutečný maximální výkon	900 mW
Rozsah měření teploty	0-50 C°
Rozsah měření tlaku	15-115 kPa
Rozměry solárních článků	210x155 mm

2.3 Blokové zapojení



2.4 Schéma zapojení



2.5 Funkce a popis zapojení

Výstupní napětí ze solárních článků je přiváděno na rezistor, který spolu se Zenerovou diodou tvoří stabilizátor napětí a úbytek napětí na rezistoru je použito k výpočtu výkonu solárních článků. V zapojení je použita baterie, která napájí elektroniku při nedostatku osvětlení. Zapínání a vypínaní baterie je automatické. Slunečnice pracuje i bez záložní baterie. Jelikož při provozu ve venkovním prostředí solární kolektory sledují chod slunce až do západu a při východu slunce by se už kolektory nedokázaly natočit.

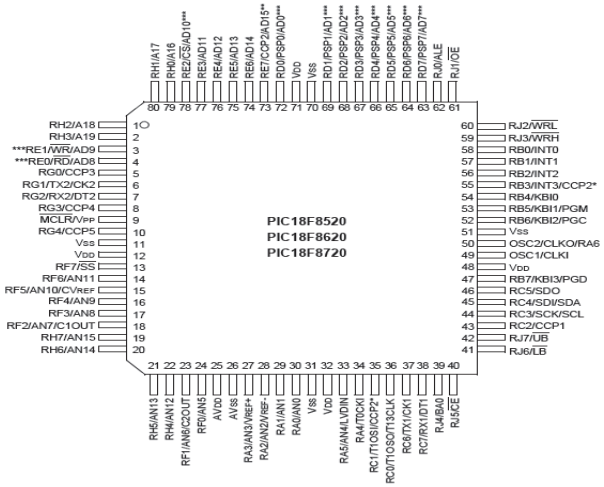
Mosfety IRF7104 a IRF7103 s prahovým napětím cca 2V pro proud cca 5A zapojené do můstku slouží jako zesilovač proudu pro motor, který otáčí slunečnicí. DC/DC konvektor tvořený obvodem LT1173 zvyšující napětí ze solárních článků na hodnotu cca 5V, které je potom stabilizováno stabilizátorem 78L02 na hodnotu 2,5V. Toto napětí slouží jako referenční napětí pro A/D převodníky. Přesnost stabilizátoru je velmi důležitá protože už při změně napětí o 4,9 mV dojde na snímači tlaku ke změně 1hPa. Snímač tlaku MPX 4115 se dá pomoci Mosfetů IRF7103 pro snížení spotřeby zapínat a vypínat. Snímač teploty je tvořen termistorem zapojeného do děliče napětí a přiváděného na vstup A/D převodníku.

2.6 Použité integrované obvody

2.6.1 Mikroprocesor PIC 18f8720.

Tento obvod byl vybrán ¨s ohledem na velký počet A/D vstupů. Vyrábí se v 64 a 80 pinovém provedení v pouzdru TQFP. I když by pro toto zapojení stačilo 64 pinové provedení, bylo použito 80 pinové provedení s ohledem na lepší dostupnost. Mikroprocesor disponuje velkou flash pamětí 128kB, integrovanou eeprom pamětí 1kB, 10 bit A/D převodníkem, SPI, I2C a RS232(UART), který je pro komunikaci s PC důležitý. Jednočip je taktován krystalem 20MHz. Jeho cena je kolem 400kč.

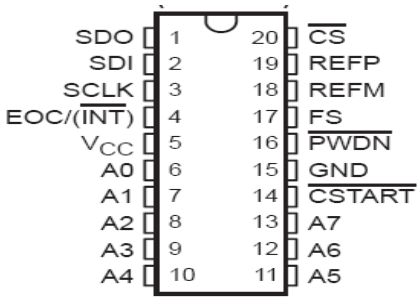
Pin Diagrams:



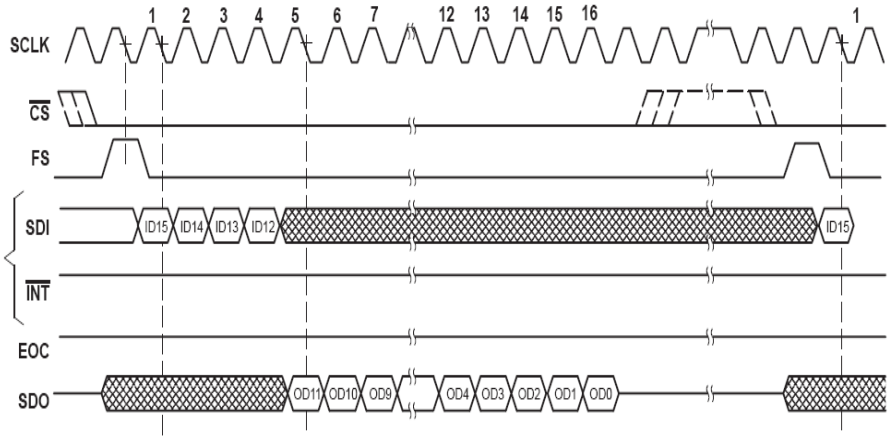
2.6.2 TLV2548

je to 12-bitový převodník komunikující s jednočipem po sběrnice SPI a slouží k vyhodnocování výstupního napětí na senzoru tlaku. Napájecí napětí muže být v rozsahu 3.3 až 5V.Odběr při napětí 3.3V je 1mA. Rychlost převodu napětí do číslicové podoby zvládne za 3.86µs. Obvod byl použit v pouzdře TSSOP pro plošnou montáž

Pin Diagrams:



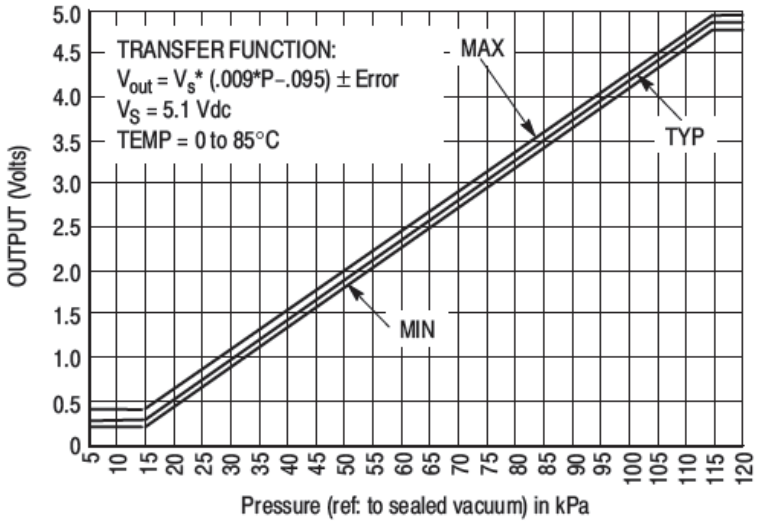
Grafický diagram čtení dat z A/D



2.6.3 Tlakový senzor MPX4115

poskytuje kalibrované výstupní napětí v závislosti na okolním atmosférickém tlaku. Výstupní napětí je dáno tímto vztahem: Uout=Us (0,009P – 0,095). Us je napájecí napětí a P je tlak v kPa. Podle katalogových údajů je citlivost tlakového čidla 4,9 mV/hPa.

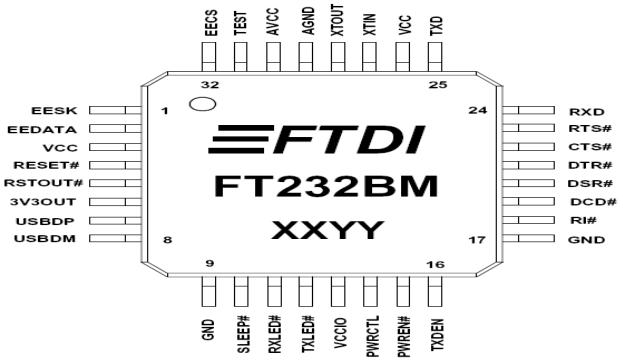
Závislost výstupního napětí na tlaku



2.6.4 Převodník FT232BM

je převodník USB → RS232 s přenosovou rychlostí 300 Bd až 3 MBd. K dispozici je i plně hardwarové řízení přenosu - signály RTS, CTS, DTR, DSR, DCD a RI. V obvodu je zabudována dvouportová vyrovnávací paměť o velikosti 128 B ve směru od PC k aplikaci a 384 B ve směru k PC. Převodníky podporují protokol USB 1.1 (resp. USB 2.0 s rychlostí 12 Mbit/s), možnost připojení externí EEPROM obsahující uživatelské sériové číslo nebo identifikační řetězec.

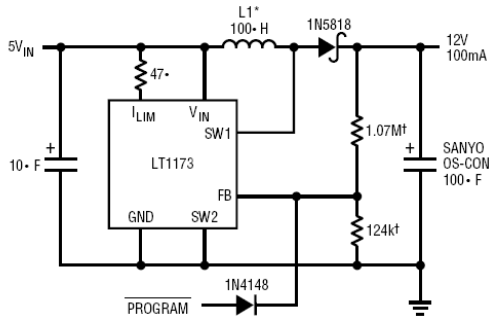
Pin Diagrams:



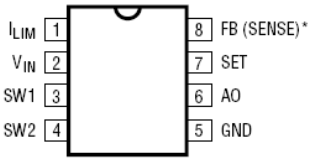
2.6.5 Konvektor LT1173

je DC/DC konvektor. Vstupní napětí se může pohybovat v rozsahu 2-12V a výstupní napětí se dá regulovat pomocí změnou napěťového děliče a indukčností cívky v rozsahu 5-12V.

Schéma měniče uvedené v datasheetu:



Pin Diagrams:

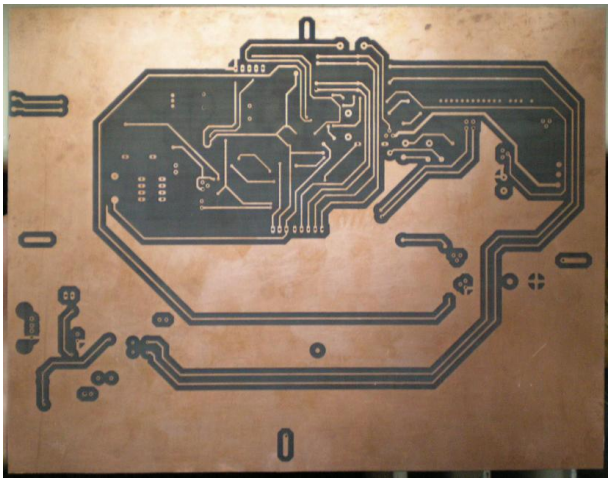


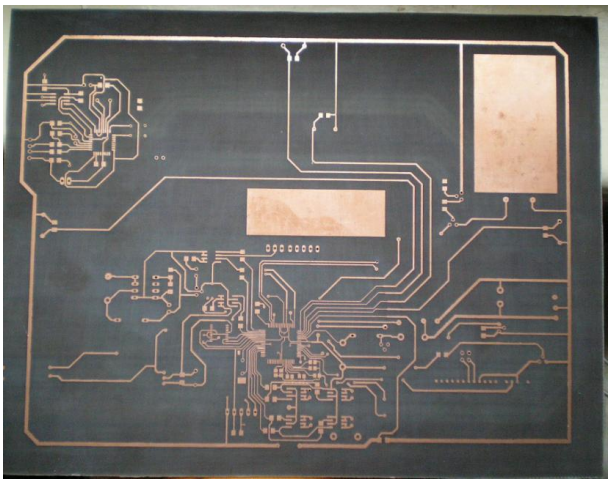
3. Závěr

Chtěl jsem se zdokonalit v programování jednočipu a jelikož se zajímám o fotovoltaické články, zvolil jsem pro tuto práci toto řešení. Mohu zde využít poznatky z programování, fyziky a mechaniky a regulační techniky. Výsledkem mé práce je skutečně model, který reaguje na světelné záření a snímá fyzikální veličiny a tyto zároveň dále umožňuje graficky zpracovávat a vyhodnocovat.

5. Přílohy

Desky plošných spojů





Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

N.I.K.I. - Projekt umělé inteligence

Jiří Hýbek

SPŠ a VOŠ Písek
Karla Čapka 402, Písek

Znáte ze sci-fi filmů situace, kdy lidé komunikují s počítačem přirozenou řečí? Už brzy to může být realita.

N.I.K.I. je program, který s vámi komunikuje prostřednictvím chatu. Vyhodnocuje vstup na základě slovníku a paměti, tedy zkušeností, které v průběhu času získal(a). Jedná se o simulaci chování člověka – dívky, které zahrnuje i napodobení emocí. Verze 1.0 dostupná na internetu umí vyprávět pár vtipů, počítat a i posílat zprávy známým uživatelům. Do budoucna je v plánu propojení s rozpoznáváním hlasu a syntézou řeči, takže se opravdu může stát, že si budete s počítačem přirozeně povídat.

Tento projekt může mít využití u osobních počítačů. Usnadní vám vyhledávání informací na internetu a celkově zpříjemní práci. Pokud se budete nudit, popovídejte si.

Projekt naleznete na adrese:

http://ai.bluezone.name/

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Inteligentní Elektroinstalace

Jan Šlajs, David Říha

SPŠ a VOŠ Písek
Karla Čapka 402, Písek

Základní konstrukce

Kde si myslíte že končí zlámané lavice? Ano, tento model je postaven z upravených školních lavic složených do základní konstrukce. Příčka je dřevotřísková a nábytek je dřevo polepené dýhou. Podlaha v obývacím pokoji je koberec a v kuchyni jsou to lepené a spárované keramické dlaždičky.

Celý dům řídí systém Ego-n, je sestaven z modulů a snímačů, které jsou mezi sebou propojeny sběrnicí. Osvětlení zajišťují tři 3W LED (dvě bílé a jedna modrá) a zářivka nad linkou. Zapnutí zásuvkových okruhů indikují LED na zásuvkách. Žaluzie jsou skutečné - mechanismus je kupovaný, ale upravený na míru modelu, pohyb žaluzií zajišťuje servomotor Belimo, převody jsou částečně provedeny z LEGA.


obr. 1

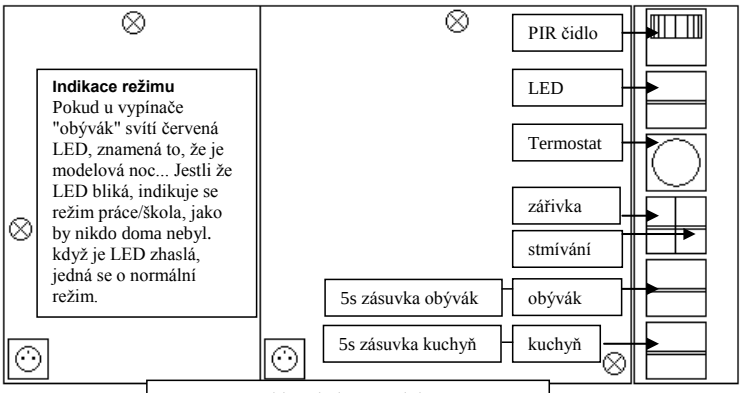

obr. 2


obr. 3

Ego-n?

- Komunikační modul - pro komunikaci s PC, PDA a dalšími zařízeními po LAN síti
- Řídící modul - řídí celý systém
- Logické funkce - tvoří podstatnou část inteligence systému - tvoří ho dvou vstupové bloky AND, OR, XOR s volitelnou negací vstupů i výstupů, zde řeší některé hardwarové nedostatky, spouští automatické režimy osvětlení a hlídá okolí systému pomocí PIR čidla.
- Stmívací modul - řídí stmívání hlavních světel a umožňuje i blikání
- Spínací modul - spíná okruhy světel
- Modul žaluziový - spíná běh motoru žaluzií a okruhy zásuvek
- Modul bin. vstupů - slouží pro vnitřní fotočidlo, a pro koncové snímače žaluzií
- Modul napájecí - napájí elektroniku modulů po sběrnici
- Dálkové ovládání - slouží jako sada bezdrátových snímačů - spínají vše!
- Snímače tlačítkové - slouží jako nevšední vypínače

Každý prvek systému se dá komfortněji naprogramovat přes software Ego-n Asistent.



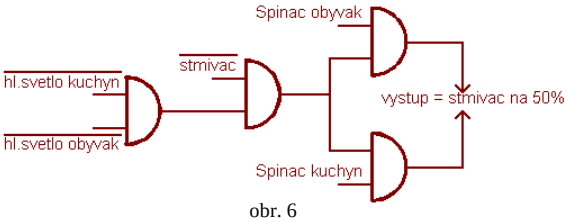
obr. 4 - návod k ovládání modelu

Ego-n Asistent a Konfigurátor

Základem vyšší úrovně programování systému Egon je software Ego-n Asistent, pro jeho funkčnost musí být Ego-n připojen k PC přes Ego-n IP-Konfigurátor. V programu Ego-n Asistent lze nastavit téměř cokoliv. Tento software je pravidelně aktualizován. Pro systém je možno navrhnout vizualizaci a být s ním v kontaktu například přes PC, PDA, nebo mobilní telefon.



obr. 5



obr. 6

Scénické režimy

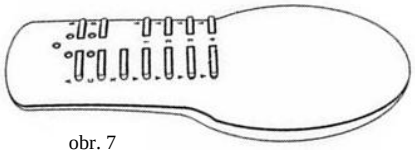
Scénický režim je ve své podstatě pouze nastavení světelných podmínek, avšak lze do režimu přidat prakticky cokoliv, lze zapnout zásuvky, nebo nastavit systém tak, aby se choval jako zabezpečovací zařízení. Další výhodou jsou režimy ovládané časově, umožňují spouštění režimu v konkrétní dobu a v nastavenou dobu opět režim vypnou.

Automatické režimy

Zde naplno vyniknou logické bloky, jelikož jsou automatické režimy spínané tlačítky, je třeba vytvořit program, který zajistí spuštění aut. režimu až po stisknutí. Programy se tvoří z logických bloků AND, OR, XOR.

Dálkové ovládání

Inteligentní elektroinstalace mají mnoho možností pro vstupní jednotky, tlačítka, binární vstupy apod. Pro větší komfort a pohodlí se vyplatí pořídit dálkové ovládání.



obr. 7

obývací pokoj:		
A:	tlačítko:	funkce:
1	Zapnout	zap. Hlavní světlo
	Vypnout	vyp. Hlavní světlo
2	Zapnout	zap. Modrá LED
	Vypnout	vyp. Modrá LED
3	Zapnout	zap. Zásuvka
	Vypnout	vyp. Zásuvka
4	Zapnout	rozsvícení - stmívač
	Vypnout	stmívání - stmívač

kuchyň		
B:	tlačítko:	funkce:
1	Zapnout	zap. Hlavní světlo
	Vypnout	vyp. Hlavní světlo
2	Zapnout	zap. Zářivka
	Vypnout	vyp. Zářivka
3	Zapnout	zap. Zásuvka
	Vypnout	vyp. Zásuvka
4	Zapnout	rozsvícení - stmívač
	Vypnout	stmívání - stmívač

Automatické režimy a žaluzie		
C:	tlačítko:	funkce:
1	Zapnout	zap. 1. Aut. režim
	Vypnout	vyp. 1. Aut. režim
2	Zapnout	zap. 2. Aut. režim
	Vypnout	vyp. 2. Aut. režim
3	Zapnout	zap. Hlídání
	Vypnout	vyp. Hlídání
4	Zapnout	žaluzie otevřít
	Vypnout	žaluzie zavřít

Scény a žaluzie		
D:	tlačítko:	funkce:
1	Zapnout	zap. 1. Power
	Vypnout	
2	Zapnout	zap. 2.Noc
	Vypnout	
3	Zapnout	žaluzie pootevřít
	Vypnout	žaluzie pozavřít
4	Zapnout	žaluzie stop
	Vypnout	

tab.1 - přehled funkcí jednotlivých tlačítek DO.

WWW:

http://www.ego-n.cz/

http://www. sps-pi.cz/

Pro školní potřeby vytvořili studenti Šlajs a Říha

Poděkování:
p.u. Paulovi, podniku ABB s. r. o., Elektro-Praga, p.u. Havlanovi, p.u. Petříkovi, p.u. Karvanovi, p.u. Komrskovi, spolužákům z D4.S - T. Pokornému a Z. Havelkovi. Dále mnoho dalším lidem, bez jejichž pomoci by tento projekt těžko vznikl.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Výuka odborných předmětů strojírenských s podporou CAD/CAM technologií, projektová činnost školy

Hana Boulová, Josef Gruber, Martin Haken, Jiří Běl, Petr Slavíček

Střední průmyslová škola strojnická, Plzeň
Klatovská 109, Plzeň

Moderní technika se neobejde bez podpory informačních technologií. Budoucí kvalifikovaní odborníci musí být už v průběhu vzdělávání dobře seznámeni s počítačovou podporou konstruování a výroby. To vyžaduje nemalé finanční zdroje, jimiž nemohou na střední odborné škole být pouze provozní prostředky. Střední průmyslová škola strojnická, Plzeň využila v minulých letech např. možnosti získat prostředky v rámci Státní informační podpory ve vzdělávání (dále SIPVZ). Tento příspěvek se věnuje dvěma vybraným projektům realizovaným v letech 2005-2007 právě v rámci SIPVZ.

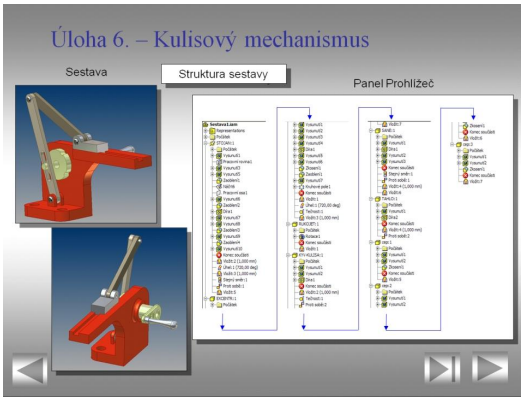


Obr. 1: CNC frézka v Centru praktického vyučování

Vyučující SPŠ strojnické, Plzeň vypracovali dvě metodické pomůcky, které mají za cíl nejen usnadnit vzdělávání žákům, ale především zefektivnit práci učitele. V současnosti se předpokládá, že každý učitel odborného strojírenského předmětu ovládá na určité úrovni nějaký prostředek počítačové podpory své oblasti. Pozice učitele je však jiná, než pozice profesionála ve firmě. Učitel musí program nejen zvládnout, ale také vedle dalších aprobačních předmětů rychle a efektivně předat znalosti žákům. Učebnic pro výuku příslušných programů je dostatek, ale chybí metodická podpora – jak efektivně učit.

Významnou podporou s takřka synergickým efektem je i následná možnost výměny zkušeností a vzájemné sdílení postupů, úloh a metod.

První z těchto metodických pomůcek se zabývá podporou výuky CAD (běžně užívaná zkratka pro počítačovou podporu konstruování) systémů. Ve třech částech je pozornost věnována postupně 2D kreslení, klasickému 3D modelování a parametrickému modelování. Přestože jsou postupy demonstrovány s využitím softwarových produktů firmy Autodesk (AutoCAD, Mechanical Desktop a Inventor), může pomůcka pomoci i ve výuce s podporou jiného softwaru – hlavní náplní je metodika, nikoli výuka práce s programem.



Obr. 2: Ukázka úvodu k jedné úloze z prezentace k Metodické příručce pro výuku CAD systémů na SOŠ a SOU

Druhá představovaná metodická pomůcka má za cíl podpořit nasazení CAM (počítačová podpora výroby) systémů do výuky. Navazuje na předchozí pomůcku a na příkladech využívajících CAD/CAM systému AlphaCAM prezentuje obecně platné metodické postupy použitelné při výuce strojního programování CNC obráběcích strojů.

Výuka, jejíž součástí je oblast CAD/CAM, se těší zájmu žáků, protože nahrazuje tradiční méně produktivní a zdlouhavé techniky, např. klasické kreslení a pauzování originálů. Místo toho nastupuje kreativní navrhování součástí, jejich propojování v celky, variantní konstruování, optimalizace, animace výsledků činnosti. Znalost základů číslicového řízení a programování CNC strojů je spojovacím článkem mezi teoretickým stanovením výrobních a provozních podmínek strojů (včetně vydatné počítačové podpory) a vlastní výrobou. AlphaCAM je CAD/CAM systém pro programování CNC strojů. Byl vytvořen firmou Licom Systems Limited of Coventry, Anglie. Je modulární. Moduly jsou dostupné v různých úrovních (Basic, Standard nebo Advanced) .Ač má AlphaCAM verze pro frézování, obrysové frézování, 3D frézování, 5D frézování, soustružení, řezání laserem, řezání plamenem, drátořez, prostřihování, vysekávání a řezání kamene, z důvodu omezené hodinové dotace bylo na SPŠ strojnické využito pouze frézování a soustružení.

V rámci předmětů informační a komunikační technologie a stavba a provoz strojů byla vymodelována součást a vygenerována výkresová dokumentace. V předmětu programování NC strojů byl vytvořen NC program a v rámci praxe byla součást vyrobena.
Obě zmíněné metodické pomůcky jsou na Střední průmyslové škole strojnické, Plzeň k dispozici v elektronické podobě, v případě zájmu se můžete obrátit na autory článku (e-mailové adresy: gruber@spstrplz.cz, boulova@spstrplz.cz).

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

MACULINEA ALCON – BORŠOV

Jakub Dubecký

Gymnázium Jiřího z Poděbrad
Studentská 166, Poděbrady 290 01

Maculinea alcon, jde o eurosibiřský druh modráska, který je v posledních letech nejen v ČR na pokraji vyhynutí. Tento motýl je monofágní, vázající se na hořec hořepník. K vývoji housenek jsou zapotřebí mravenci rodu Myrmica. Na území ČR obývá vlhké bezkolencové louky s živnou rostlinou. Motýl má jednu generaci.



Smyslem práce je monitoring tohoto druhu, a následné použití dat při změně v managementu území a zjištění, v jakém stavu se populace nachází. Současná populace čítá cca 50 kusů každým rokem. Modrásek se opět vrátil i na chráněné území, kde původně vyhynul. Jeho rozšíření se podařilo z nedaleké obhospodařované louky, kde přežívalo pár posledních jedinců.

Středoškolská technika 2008

Setkání a prezentace prací středoškolských studentů na ČVUT

STUDIUM KMITŮ MATEMATICKÉHO KYVADLA

Martin Matuš, Jan Lednický

Gymnázium Jiřího z Poděbrad

Studentská 166, Poděbrady

1. Úvod

1.1. Anotace - vymezení řešeného problému

V první části práce charakterizujeme matematické a fyzikální kyvadlo, zabýváme se jejich historií a stručně zmiňujeme možnosti jejich využití. Dále se zabýváme matematickým kyvadlem, specifikujeme síly na něj působící a uvádíme pro něj platné pohybové rovnice. Numerickými metodami potom tyto rovnice řešíme – zvláště pak pro kmity neharmonické – a dosažené výsledky podáváme ve formě tabulek, rovnic a grafů.

V závěru pak vše stručně sumarizujeme, hodnotíme, zabýváme se přesností našeho výpočtu a diskutujeme výsledky, k nimž jsme došli. Nedílnou součástí této práce je příloha v podobě souboru v tabulkovém procesoru, pomocí něhož jsme všechny výpočty provedli, rovněž tak seznam použitých zkratek a literatury.

1.2. Matematické kyvadlo

Matematické kyvadlo je matematickým modelem kyvadla. U matematického kyvadla se zkoumá pouze hmotný bod zavěšený na tenkém vláknu (v našem případě tuhém závěsu) zanedbatelné hmotnosti; zanedbává se odpor vzduchu při pohybu kyvadla i tření v závěsu a gravitační pole se považuje za homogenní. Matematické kyvadlo je mechanický oscilátor, tedy zařízení, které po dodání počáteční energie volně kmitá bez vnějšího působení. Při malých výchylkách je průběh tohoto kmitání harmonický, lze jej tedy vyjádřit pomocí funkce sinus.

1.3. Fyzikální kyvadlo

Fyzikální kyvadlo (též fyzické kyvadlo) je tuhé těleso otočné bez tření kolem vodorovné osy, jež neprochází těžištěm. Od matematického kyvadla se liší tím, že těleso není považováno za hmotný bod.

1.4. Trocha historie

První zaznamenané využití kyvadla pochází z dob čínské dynastie Chan (206 př. n. l. – 220 n. l.) v seismometru vědce a vynálezce Zheng Henga. Jeho účelem bylo aktivovat sérii páček poté, co bylo rozkmitáno dalekým zemětřesením. Tím se nastartoval proces, na jehož konci byl míček, který vypadl z otvoru v přístroji do jedné z nádob vespod. Tím se dalo určit, jakým směrem leží epicentrum zemětřesení.

Egyptský vědec Ibn Yunus popsal kyvadlo v 10. století n. l. Někdy se uvádí, že ho použil k měření času, ale to se považuje za chybnou interpretaci anglického historika Edwarda Bernarda.

Také Galileo Galilei mezi svými mnohými objevy mnohokrát pozoroval a popisoval vlastnosti kyvadla. Jeho zájem o kyvadla byl patrně vzbuzen pozorováním lustru v katedrále v Pise. Své studium kyvadla započal roku 1602. Zjistil, že doba periody kmitu je nezávislá na hmotnosti zavěšeného předmětu. Jako praktické využití kyvadla navrhl Galileo metronom, tedy pomůcku pro studenty hudby, a možné použití v hodinách.

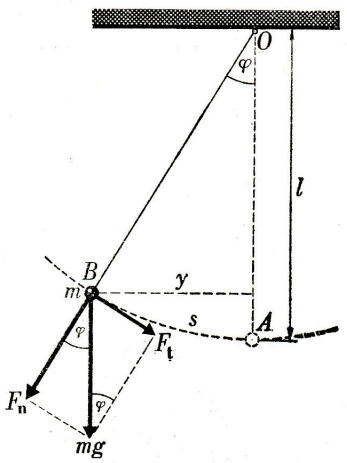
Možná na základě Galileových návrhů nechal Christiaan Huygens patentovat mechanické hodiny, které využívaly kyvadlo jako regulaci pohybu. To se stalo v roce 1656. Tyto hodiny pak byly mnohem přesnější než ty doposud užívané.

2. Pohyb matematického kyvadla

2.1. Harmonicky kmitající matematické kyvadlo

Matematické kyvadlo je hmotný bod kývající se na pevné tyčce zanedbatelné hmotnosti. Rozklad tíhové síly Fg = mg při vychýlení kyvadla délky l o úhel φ je na obrázku. Složka F_n se v každé poloze ruší tahovou silou tyčky, složka F_t míří proti výchylce φ a vyvolává kmitavý pohyb. (obr. XV-15a) Pro složku F_t, která uvádí těleso do pohybu, pak můžeme psát

$$F_t = mg \cdot \sin\varphi \qquad (1)$$



XV-15a. Matematické kyvadlo

Uvažujme malou úhlovou výchylku φ < 5°, pro něž lze sinus úhlu nahradit s dostatečnou přesností přímo úhlem samým vyjádřeným ovšem v obloukové míře, tj. v radiánech

$$\sin\varphi \approx \varphi$$

Použitím tohoto faktu pro F_t dostáváme:

$$F_t = mg \cdot \sin\varphi = mg \cdot \varphi = k \cdot \varphi \qquad (2)$$

... tzn. síla F_t je přímo úměrná **φ,** kmity jsou tedy harmonické, neboli úhel **φ** (okamžitá výchylka) závisí na čase prostřednictvím funkce sinus.

Uvážíme-li, že úhel φ lze vyjádřit pomocí délky oblouku kružnice s a délky kyvadla l jako

$$\varphi = \frac{s}{l} \tag{3}$$

můžeme rovnici (2) vyjádřit ve tvaru

$$F_t = mg.\frac{s}{l} = \frac{mg}{l} \cdot s = k \cdot s = k \cdot y \tag{4}a\ (5)$$

$$kde\ k = \frac{mg}{l}$$

Úhlová rychlost kmitání je pak dána vztahem

$$\omega = \sqrt{\frac{k}{m}} = \sqrt{\frac{g}{l}} \tag{6}$$

Periodu T, resp. frekvenci f, pak již odvodíme snadno

$$T = \frac{2\pi}{\omega} = 2\pi \cdot \sqrt{\frac{l}{g}}$$

$$f = \frac{1}{T} = \frac{1}{2\pi} \cdot \sqrt{\frac{g}{l}}$$

Rovnici harmonických kmitů matematického kyvadla, které odpovídá graf č. 1, pak můžeme – za předpokladu, že hmotný bod v čase 0 s právě prochází rovnovážnou polohou, – napsat pomocí amplitud s_m, φ_m ve tvaru

$$s = s_m \cdot \sin(\omega t)$$

resp. (9) *a* (10)

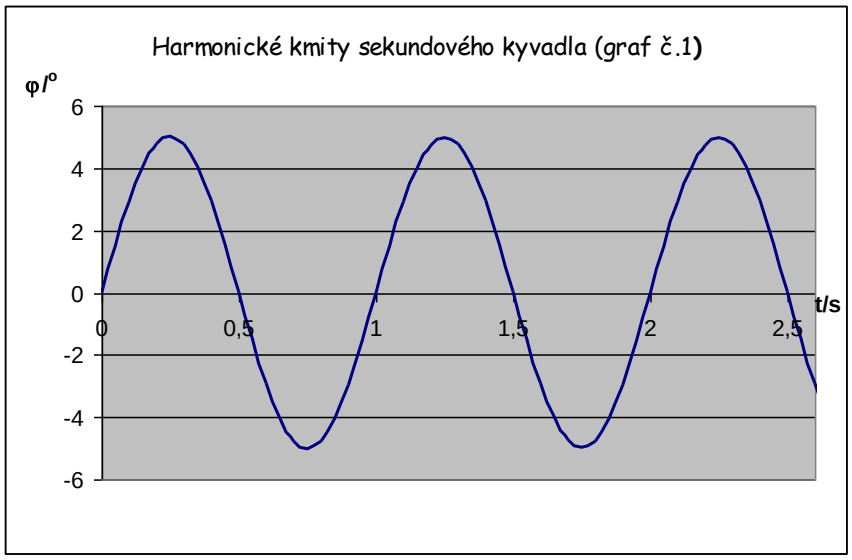
$$\varphi = \varphi_m \cdot \sin(\omega t)$$

Doba kmitu T (jakož i ω a f) závisí pouze na délce l kyvadla a na tíhovém zrychlení g; nezávisí na hmotnosti m kyvadla. Poloviční doba kmitu τ = T/2 se nazývá *doba kyvu*. Tzv. *sekundové kyvadlo* má dobu kyvu τ = 1 s. Pro jeho délku evidentně platí

(11)

$$l = \frac{g}{g_2}$$

$$l = \frac{\pi^2}{\pi^2}$$



2.2. Neharmonické kmity matematického kyvadla

Jestliže vychýlíme matematické kyvadlo o velké úhly (větší než 5°) a uvolníme, začne vykonávat kmitavý pohyb, který již není harmonický. V této kapitole se pak budeme snažit takový pohyb matematicky popsat.

Jak již bylo v předchozí kapitole zmíněno, kmitavý pohyb matematického kyvadla je způsoben silou **F**, která závisí na okamžité výchylce vztahem (1), tj.

$$F = mg \cdot \sin\varphi$$

Každou sílu ovšem můžeme vyjádřit také pomocí 2. Newtonova pohybového zákona, v němž vystupuje zrychlení a

$$F = m \cdot a \qquad (12)$$

Porovnáním rovnic (1) a (12) získáme závislost zrychlení a na okamžité výchylce **φ**, což je úhel, který svírá matematické kyvadlo se svislým směrem.

$$\begin{aligned} m \cdot a &= mg \cdot \sin\varphi \\ a &= g \cdot \sin\varphi \end{aligned} \qquad (13)$$

Pro jednoduchost přejdeme od zrychlení posuvného pohybu a ke zrychlení úhlovému **ε**, mezi nimiž platí vztah

$$\varepsilon = \frac{a}{l} = \frac{g}{l} \cdot \sin\varphi \qquad (14)$$

Poznámka:

Uvážíme-li, že pro malé výchylky **φ** je sin**φ** ≈ **φ**, můžeme vyjádřit úhlové zrychlení harmonicky kmitajícího kyvadla takto

$$\varepsilon = \frac{g}{l} \cdot \varphi \qquad (15)$$

Naším úkolem nyní bude s využitím známé závislosti ε = f(φ) zjistit jakým způsobem závisí pro obě kyvadla – jak harmonicky, tak neharmonicky kmitající – okamžitá výchylka **φ** na čase. Protože tomu odpovídající diferenciální rovnice (**φ´´**=k.**φ**, resp**. φ´´** = k.sin**φ)** nejsou pro středoškoláka řešitelné, budeme k problému přistupovat numericky s použitím příslušného software; v našem případě tabulkového procesoru Microsoft Excel.

2.3. Numerické řešení pohybu neharmonického kyvadla

Problém matematického kyvadla řeší excelovský soubor „Kyvadlo“. Do listu „Zadání a výsledky“ nejdříve zadáme konstanty, s nimiž bude program pracovat, tzn. tíhové zrychlení g, délka kyvadla l, časový krok, po něž se bude počítat nová poloha kyvadla Δt, a amplitudy výchylky φ_m, tj. maximální odchylka od svislého směru. Situace vypadá takto

g/ms^{-2}	9,81
l/m	1
dt/s	0,01

□$_m$/^{o}	160,0
T/s	2,0

Vlastní výpočty jsou realizovány v listu „Výpočty“, kde nalezneme tabulka s tímto záhlavím

.

t/s	□ □$_1$ /rad	□ □$_2$ /rad	□$_1$/rad.s^{-2}	□$_2$/rad.s^{-3}	□$_1$/rad.s^{-1}	□$_2$/rad.s^{-2}

čas

okamžitá výchylka harmonického oscilátoru

úhlová rychlost neharmonického oscilátoru

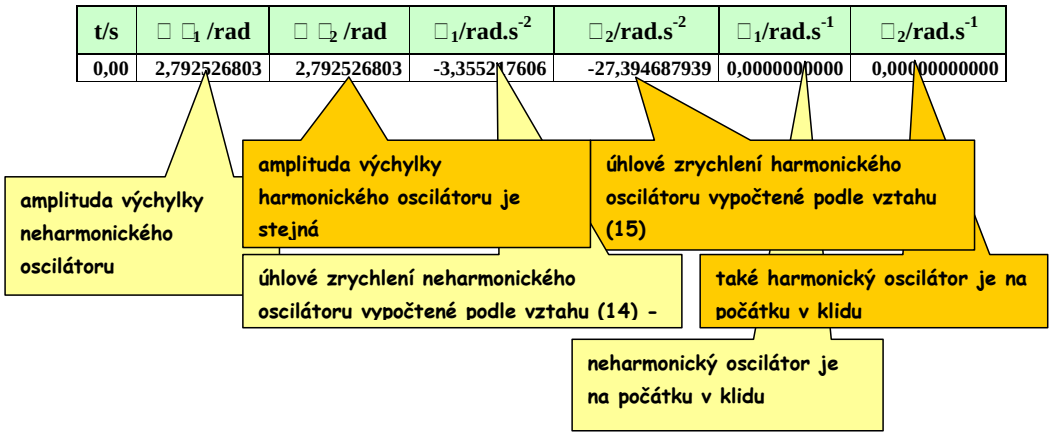
okamžitá výchylka neharmonického oscilátoru

úhlové zrychlení neharmonického oscilátoru

úhlová rychlost harmonického oscilátoru

úhlové zrychlení harmonického oscilátoru

V prvním řádku tabulky popíšeme výchozí stav oscilátoru, tzn. počáteční podmínky v čase 0 s. Program pak vypočte podle vztahu (14) a (15) příslušná úhlová zrychlení.



Nyní necháme uplynout velmi krátký čas, přičemž program na druhém řádku počítá, jak se dané veličiny změnily. Jeli časový interval Δt velmi malý, např. 0,01 s, můžeme po tuto dobu považovat zrychlení ε_1 a ε_2 za konstantní. Změnu úhlové rychlosti tedy můžeme vyjádřit jako

$$\Delta\omega_1 = \varepsilon_1 \cdot \Delta t$$
$$\Delta\omega_2 = \varepsilon_2 \cdot \Delta t \qquad (16)$$

A právě o tyto hodnoty se zvětší hodnoty ω_1 a ω_2 v 6. a 7. sloupci.

Protože časový interval Δt je velmi malý, můžeme po tuto dobu považovat úhlovou rychlost ω_1 a ω_2 obou kmitavých pohybů přibližně za konstantní. Změnu úhlu $\Delta\varphi_1$ a $\Delta\varphi_2$ pak mžeme psát jako

$$\Delta\varphi_1 = \omega_1 \cdot \Delta t \qquad (17)$$
$$\Delta\varphi_2 = \omega_2 \cdot \Delta t$$

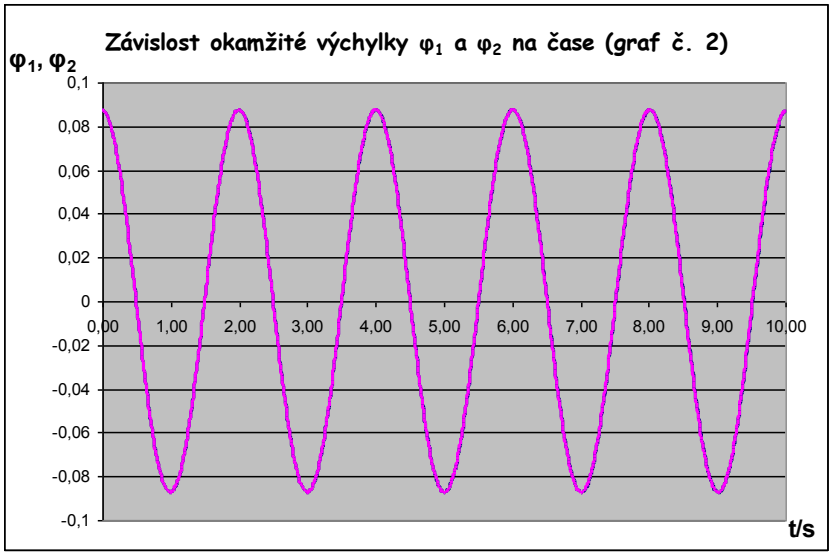
A o tyto hodnoty zvětšíme hodnoty φ_1 a φ_2 v 2. a 3. sloupci tabulky a vypočteme nová zrychlení □1 a □2. Dostáváme tak popis situace v čase 0,01 s.

t/s	□ □1 /rad	□ □2 /rad	□1/rad.s-2	□2/rad.s-2	□1/rad.s-1	□2/rad.s-1
0,00	2,792526803	2,792526803	-3,355217606	-27,394687939	0,0000000000	0,00000000000

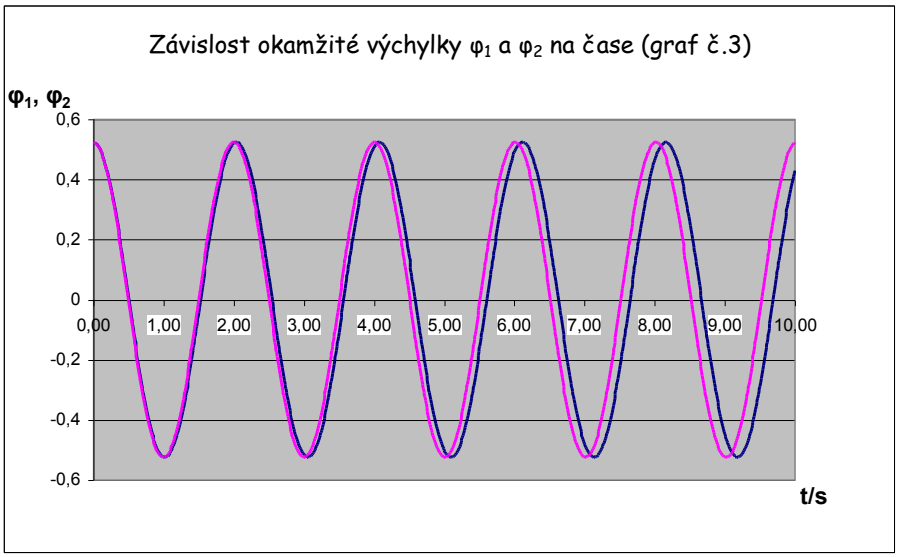
0,01	**2,792191281**	**2,789787334**	**-3,358310386**	**-27,367813750**	**-0,03355217606**	**-0,273946879**

Tento postup pak můžeme neomezeně dlouho opakovat. Musíme si však být vědomi toho, že každý krok je zatížen drobnou chybou, která postupně narůstá., až výsledky nedopovídají realitě. V tabulce v listě „Výpočty“ jsou tak numericky vyjádřeny závislosti $\varphi_1 = f(t)$, $\varphi_2 = f(t)$, $\omega_1 = f(t)$, $\omega_2 = f(t)$, $\varepsilon_1 = f(t)$ a $\varepsilon_2 = f(t)$. Příslušné grafy jsou na předcházejícím listě „Zadání a výsledky“.

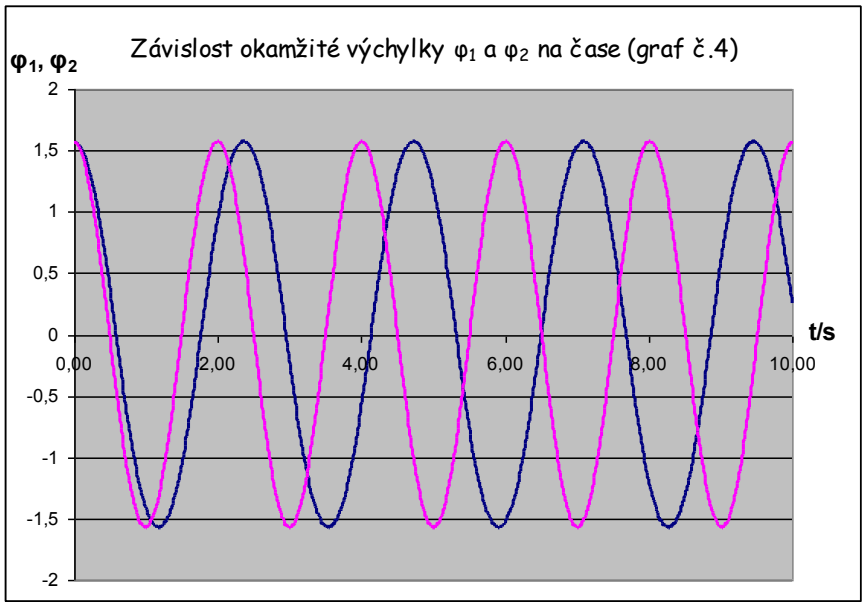
Měníme-li nyní amplitudu výchylky □ □$_m$, dostáváme grafy jak kmitů harmonických **(fialově),** tak neharmonických **(modře).** Podle očekávání při hodnotě 5° nemůžeme obě křivky odlišit (graf č. 2).



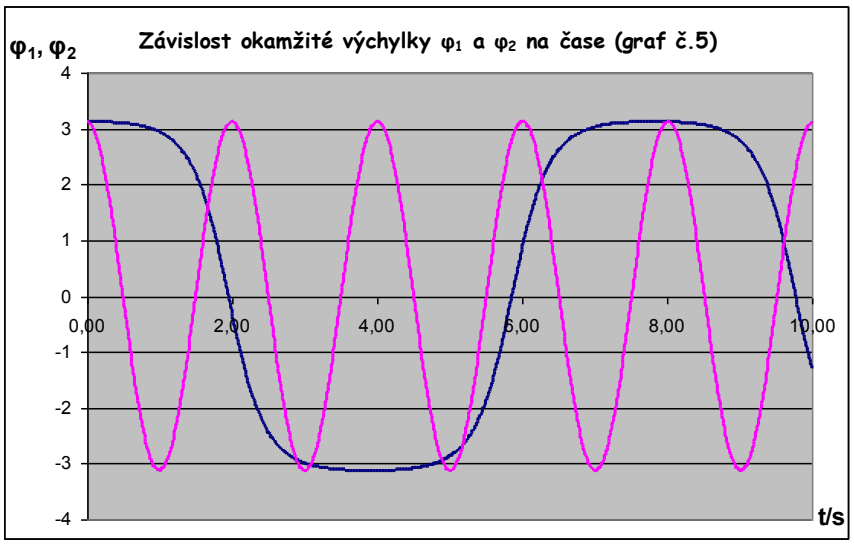
Když matematické kyvadlo odchýlíme o 30° a pustíme, stále ještě bude kmitat přibližně harmonicky. Jeho perioda však je již výrazně větší než odpovídá hodnotě vypočtené podle vztahu (8), tj. 2,01 s. (graf č. 3).



Při $\varphi_{im} = 90^{o}$ kmitá matematické kyvadlo zdánlivě stále harmonicky; perioda je však již výrazně vyšší oproti hodnotě teoretické (graf č. 4).



Výrazně neharmonické kmity pak vykonávají kyvadla, jejichž rozkmit je opravdu extrémní. Zde se jedná o hodnotu 179° (graf č. 5).

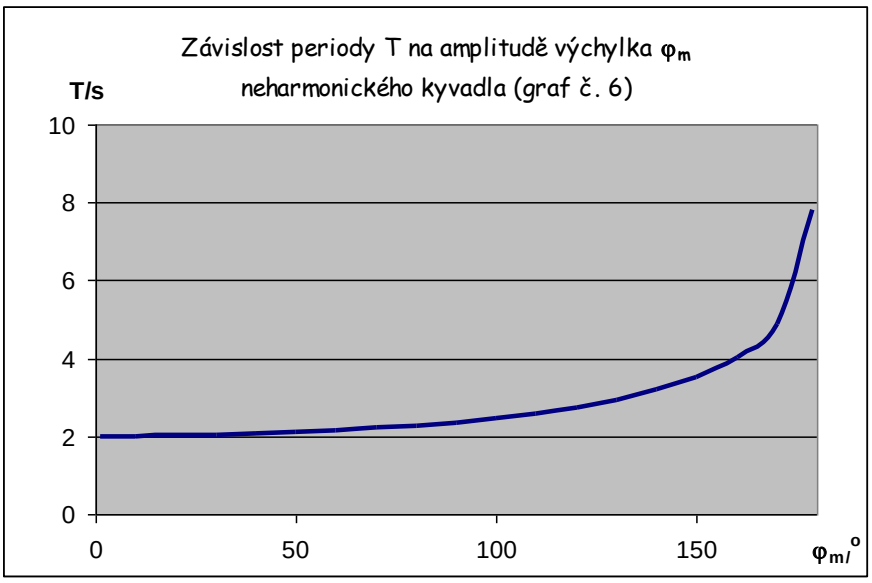


2.4. Perioda kmitů neharmonicky kmitajícího kyvadla

U neharmonicky kmitajícího matematického kyvadla není tedy perioda konstantní (tj. 2,01 s), ale záleží na amplitudě výchylky. Opakovaným použitím našeho programu získáme následující závislost T=f(φ_m) (tabulka č. 1)

č.	φ_m/°	T/s	č.	φ_m/°	T/s
1	1	2,01	12	110	2,59
2	10	2,01	13	120	2,74
3	20	2,02	14	130	2,94
4	30	2,04	15	140	3,20
5	40	2,06	16	150	3,53
6	50	2,10	17	160	4,02
7	60	2,15	18	170	4,89
8	70	2,21	19	179	7,82
9	80	2,28			
10	90	2,36			
11	100	2,47			

Znázorníme–li funkci T = f(φ_m) graficky, získáme graf č. 6



2.5. Aproximace rovnice neharmonických kmitů kyvadla Fourierovou řadou

V tabulce v listu „Výpočty“ tedy máme funkci φ = f(t) pro $\varphi_m = 179^{\circ}$ s odpovídající periodou asi T = 7,82 s v podobě uspořádaných dvojic [t, φ]. Nyní se pokusíme napsat její rovnici. Využijeme toho, že každou lichou periodickou funkci φ můžeme v intervalu ⟨0; T⟩ napsat jako Fourierovu řadu ve tvaru:

$$\varphi(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cdot \cos(n\omega t); \quad \omega = \frac{2\pi}{T}$$

kde

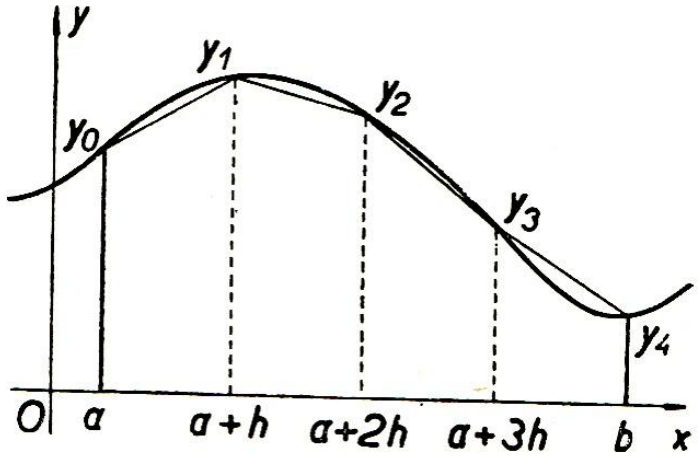
$$a_n = \frac{2}{T}\int_0^T \varphi(t) \cdot \cos(n\omega t)dt \qquad (18)$$

Musíme tedy počítat tyto určité integrály

$$a_0 = \frac{2}{T}\int_0^T \varphi(t)dt; \quad a_1 = \frac{2}{T}\int_0^T \varphi(t)\cdot\cos(\omega t)dt; \tag{19}$$

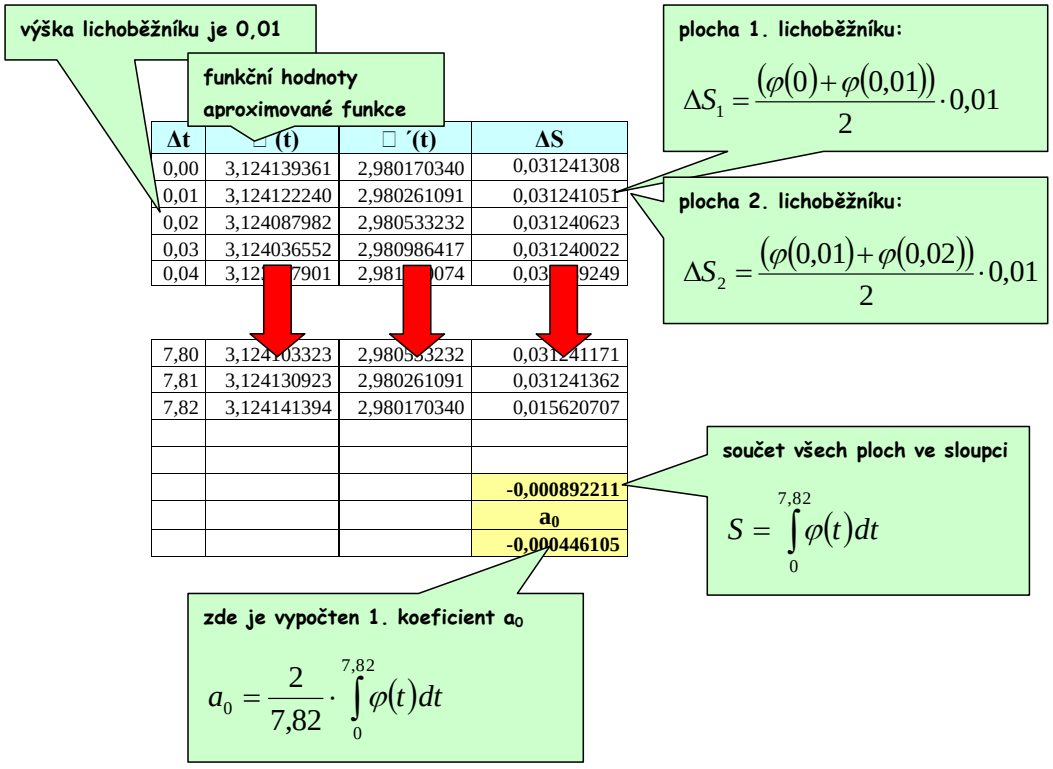
$$a_2 = \frac{2}{T}\int_0^T \varphi(t)\cdot\cos(2\omega t)dt;\ a_3 = \frac{2}{T}\int_0^T \varphi(t)\cdot\cos(3\omega t)dt;\ \ldots$$

Využijeme toho, že určitý integrál v mezích od **a** do **b** lze interpretovat jako obsah plochy S pod grafem v tomto intervalu. K přibližnému výpočtu pak použijeme lichoběžníkovou metodu, která spočívá v tom, že si plochu pod grafem rozdělíme na co možná největší počet lichoběžníků, určíme plochu každého z nich a pak tyto plochy sečteme. Výpočet je tím přesnější, čím je počet lichoběžníků větší. (Obr. 13.8)



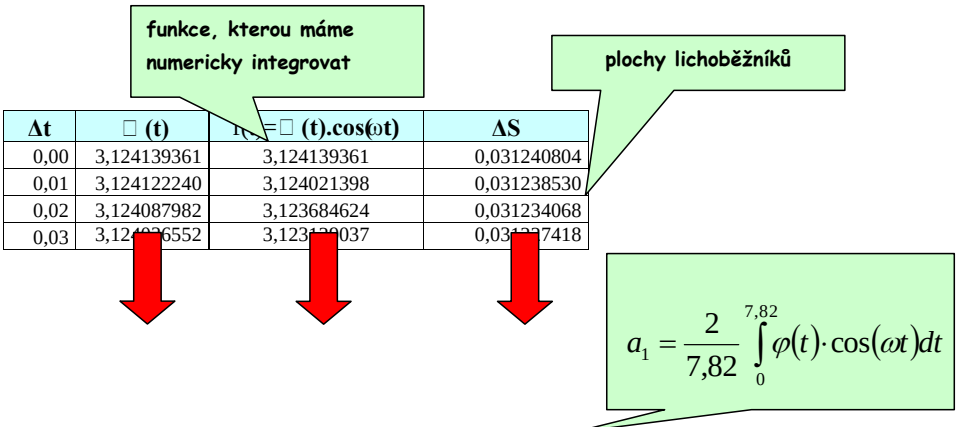
Obr. 13.8.

Výpočet je na listu „Aproximace“, kde najdeme tabulku, která obsahuje toto:



Δt	□ (t)	□ ´(t)	ΔS
0,00	3,124139361	2,980170340	0,031241308
0,01	3,124122240	2,980261091	0,031241051
0,02	3,124087982	2,980533232	0,031240623
0,03	3,124036552	2,980986417	0,031240022
0,04	3,123967901	2,981620074	0,031239249
7,80	3,124103323	2,980533232	0,031241171
7,81	3,124130923	2,980261091	0,031241362
7,82	3,124141394	2,980170340	0,015620707
			-0,000892211
			a_0
			-0,000446105

Obdobně postupujeme při výpočtu koeficientu a_1 (ω = **0,803476382 rad.s^{-1}**).



Δt	□ (t)	□(t).cos(ωt)	ΔS
0,00	3,124139361	3,124139361	0,031240804
0,01	3,124122240	3,124021398	0,031238530
0,02	3,124087982	3,123684624	0,031234068
0,03	3,124036552	3,123130037	0,031227418

7,80	3,124103323	3,123699963	0,031238650
7,81	3,124130923	3,124030081	0,031240857
7,82	3,124141394	3,124141394	0,015620707
			a_1
			3,699870689

...a také a_2, a_3, a_4, a_5, a_6, a a_7.

Δt	**□ (t)**	**f(t)=□ (t).cos(2ωt)**	**ΔS**	**f(t)=□ (t).cos(3ωt)**	**ΔS**
0,00	3,124139361	3,124139361	0,031239291	3,124139361	0,031236770
0,01	3,124122240	3,123718878	0,031230968	3,123214700	0,031218365
0,02	3,124087982	3,122474656	0,031214408	3,120458389	0,031181652
0,03	3,124036552	3,120407019	0,031189618	3,115872080	0,031126653

7,79	3,124058566	3,120429008	0,031214595	3,115894037	0,031181839
7,80	3,124103323	3,122489989	0,031231088	3,120473712	0,031218485
7,81	3,124130923	3,123727560	0,031239345	3,123223381	0,031236824
7,82	3,124141394	3,124141394	0,015620707	3,124141394	0,015620707
			a_2		**a_3**
			0,007676852		**-0,726931095**

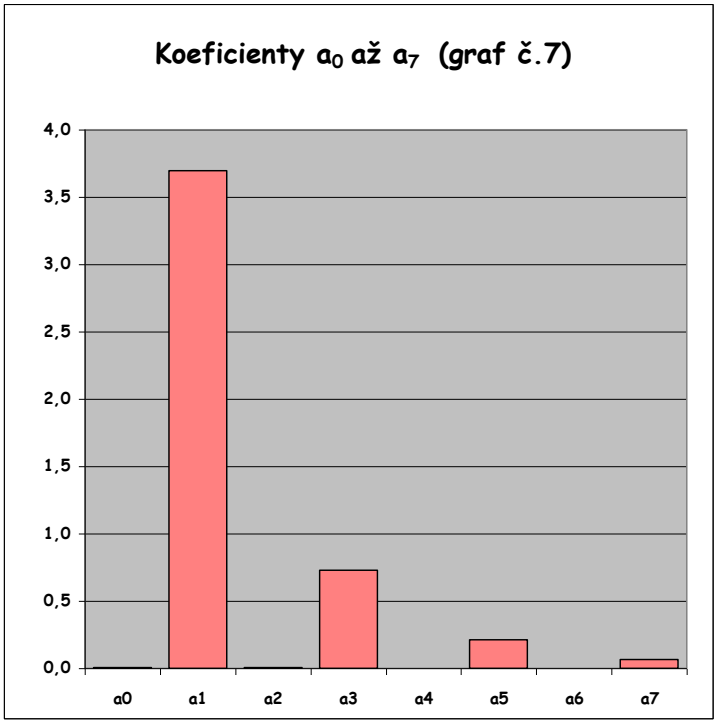
$$a_2 = \frac{2}{7{,}82}\int_0^{7{,}82} \varphi(t)\cdot\cos(2\omega t)dt$$

$$a_3 = \frac{2}{7{,}82}\int_0^{7{,}82} \varphi(t)\cdot\cos(3\omega t)dt$$

Výsledky našeho snažení jsou obsaženy v této tabulce (tabulka č. 2), jenž byla vytvořena na následujícím listu „Frekvenční analýza“.

$\lvert a_0 \rvert$	**0,0048873**	**0,1%**	zanedbatelné
a_1	**3,6958756**	**100,0%**	
a_2	**0,0036818**	**0,1%**	zanedbatelné
$\lvert a_3 \rvert$	**0,7309262**	**19,8%**	
$\lvert a_4 \rvert$	**0,0018728**	**0,1%**	zanedbatelné
a_5	**0,2096489**	**5,7%**	
a_6	**0,0008773**	**0,0%**	zanedbatelné
$\lvert a_7 \rvert$	**0,0678353**	**1,8%**	

A zdá se, že sudé koeficienty a_0, a_2, a_4, a a_6 budeme moci opravdu zanedbat, což vyplývá i z následujícího grafu č. 7



Zanedbáme-li tedy tyto koeficienty, můžeme napsat funkci vyjadřující závislost okamžité výchylka na čase (tzn. $\varphi' = f(t)$) pro $\varphi_m = 179^o$ pomocí Fourierovy řady takto:

$$\varphi'(t) = 3{,}696 \cdot \cos(\omega t) - 0{,}731 \cdot \cos(3\omega t) + 0{,}210 \cdot \cos(5\omega t) + \\ +0{,}068 \cdot \cos(7\omega t) + \ldots \quad \omega = 0{,}803\ rad \cdot s^{-1} \qquad (20)$$

… a to bylo naším cílem!

Poznámka:

Pokud bychom řešili kmity matematického kyvadla s amplitudou výchylky φ_m menší než 179°, prováděli bychom výpočty ve stejných excelovských tabulkách ovšem s tím, že bychom je museli zdola omezit na periodu.

2.6. Spektrum kmitů matematického kyvadla

Rovnici (20) můžeme pomocí známých goniometrických vztahů mezi siny a cosiny

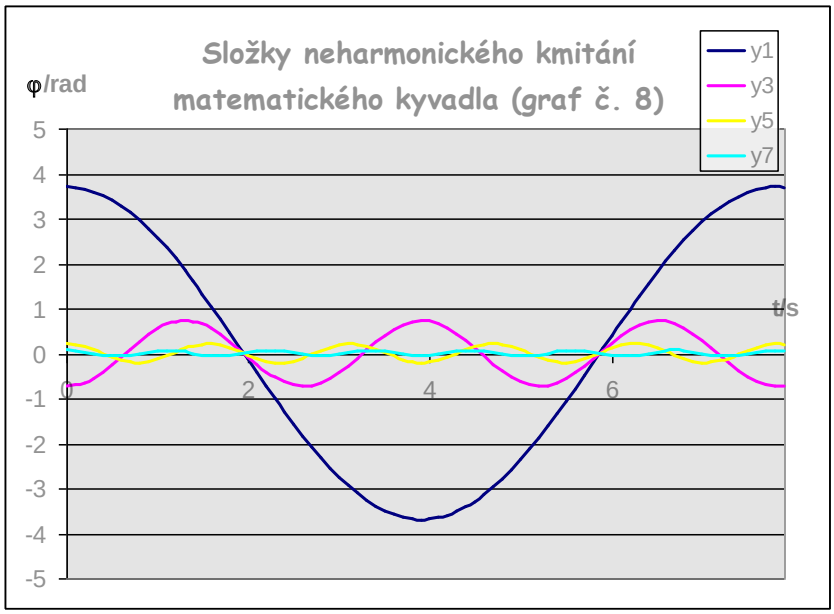
$$\cos x = \sin\left(x + \frac{\pi}{2}\right) \quad (21)$$

$$-\cos x = \sin\left(x - \frac{\pi}{2}\right) \quad (22)$$

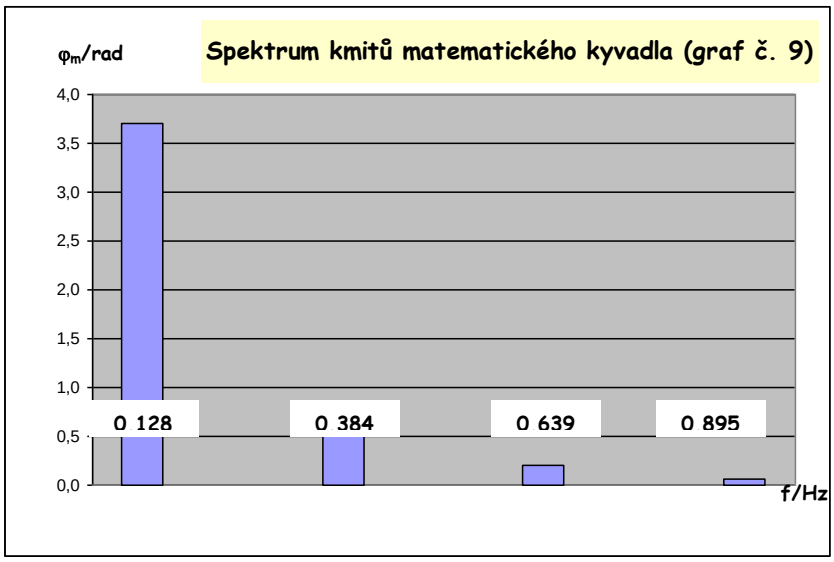
přepsat na tvar :

$$\varphi'(t) = 3{,}696 \cdot \sin\left(\omega t + \frac{\pi}{2}\right) + 0{,}731 \cdot \sin\left(3\omega t - \frac{\pi}{2}\right) + 0{,}210 \cdot \sin\left(5\omega t + \frac{\pi}{2}\right) + \\ + 0{,}068 \cdot \sin\left(7\omega t + \frac{\pi}{2}\right) + \ldots \; ; \; \omega = 0{,}803 \; rad \cdot s^{-1} \quad (23)$$

Na neharmonické kmity matematického kyvadla se tedy můžeme dívat jako na kmity složené a to z kmitů harmonických, jejichž amplitudy výchylky odpovídají absolutní hodnotě koeficientů Fourierovy řady. Příslušný graf č. 8 byl vytvořen opět v listu „Frekvenční analýza“.



Ve spektru složeného kmitání matematického kyvadla se tedy nachází jistá nejnižší základní frekvence (f_1= 0,128 Hz) a pak její liché násobky (f_3 = 0,384 Hz, f_5 = 0,639 Hz, f_7 = 0,895 Hz, ... graf č. 9)



3.

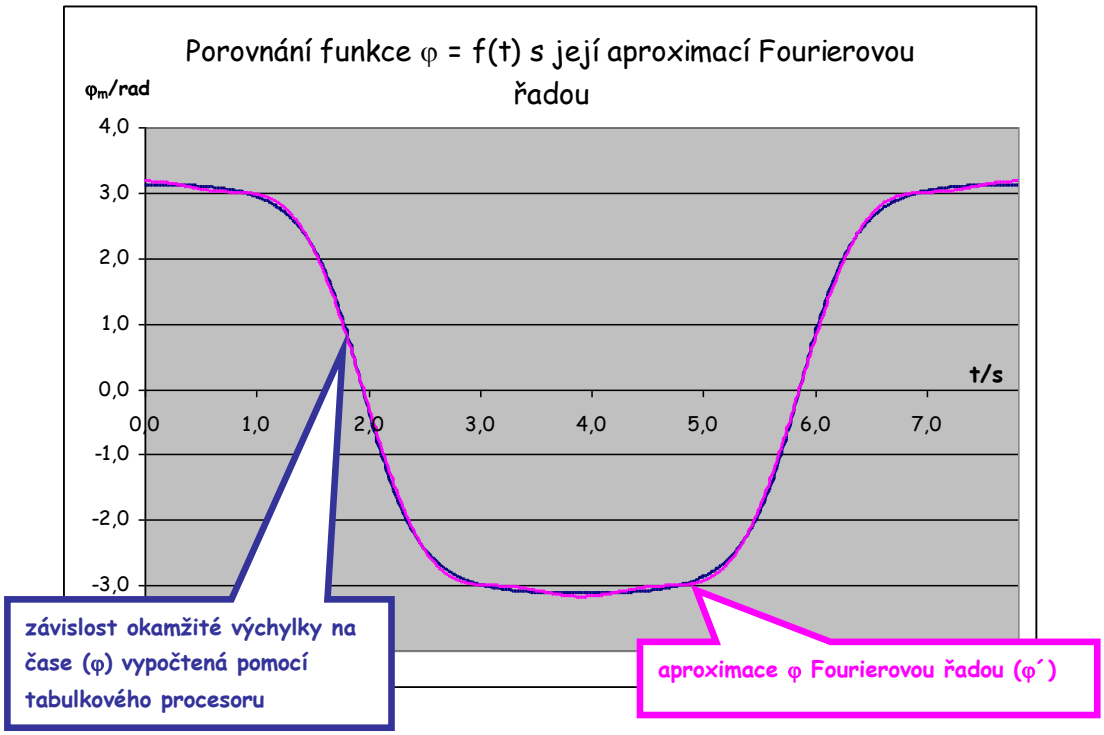
Diskuze

Výše uvedené vyjádření závislosti odchylky matematického kyvadla na čase rovnicí (20) by nemuselo být dostatečně přesné, protože jsme z Fourierovy řady použili jenom první čtyři nenulové členy. Abychom přesnost posoudili, musíme srovnat grafy funkce původní (φ) a funkce aproximované (φ´). K tomu je určena tato část tabulky v listu v listu „Aproximace“.

t	□ (t)	□ ´(t)
0,00	3,124139361	2,980170340
0,01	3,124122240	2,980261091
0,02	3,124087982	2,980533232
0,03	3,124036552	2,980986417
0,04	3,123967901	2,981620074
0,05	3,123881961	2,982433401

7,81	3,124136	2,980261091
7,82	3,124141394	2,980170340

... na základě níž sestrojí tabulkový procesor graf č. 10



Na základě tohoto grafu pak můžeme říci, že se nám úkol popsat kmity matematického kyvadla pro amplitudu 179° podařilo splnit, neboť obě křivky velmi dobře korespondují. Rovnice (23) je tedy dostatečně přesná na to, aby byla pohybovou rovnicí těchto kmitů. V opačném případě by ovšem nic nebránilo v tom, abychom ve výpočtech pokračovali a obdobným způsobem neurčili i další členy Fourierovy řady.

4. Závěr

V této práci se nám numerickými metodami podařilo potvrdit, že matematické kyvadla kmitá pro malé výchylky opravdu harmonicky a že platí rovnice (7) a (8) pro frekvenci a periodu.

Dále jsme zkoumali pohyb matematického kyvadla pro amplitudu výchylky větší než 5°, přičemž jsme určili grafy závislosti okamžité výchylka na čase. Zjistili jsme, že doba kmitu neharmonicky kmitajícího kyvadla je větší než harmonicky kmitajícího kyvadla a sestrojili jsme graf závislosti periody na amplitudě výchylky

Nakonec jsme numerickými metodami studovali neharmonické kmity kyvadla s extrémní amplitudou výchylky 179°, přičemž se nám rovnici tohoto pohybu podařilo vyjádřit pomocí členů Fourierovy řady.

Došli jsme také k závěru, že na kmity neharmonického kyvadla se můžeme dívat jako na kmity složené; tzn. kyvadlo kmitá s jakousi „základní frekvencí“ f a pak s jejími lichými násobky 3f, 5f, 7f, … (obdoba vyšších harmonických tónů v akustice), jejichž amplituda však velmi rychle klesá.

Naše práce je zvláště přínosná v tom, že numerická metoda řešení pohybových rovnic je obecně použitelná i na řešení pohybu jakéhokoliv tělesa za předpokladu, že dokážeme specifikovat jakým způsobem závisí síla na těleso působící na nějakém parametru, kterým může být např. vzdálenost, souřadnice, rychlost aj. Není tedy nutné řešit problém obecně a sestavovat často složité diferenciální rovnice, jejichž řešení by pohyb popisovalo. Místo toho zadáváme do tabulkového procesoru pouze počáteční hodnoty veličin a vztahy mezi nimi, které jsou pak tak jednoduché, že je dokáže pochopit i středoškolák. Program potom neřeší diferenciální rovnice, ale počítá polohu pohybujícího se tělesa „bod po bodu“, přičemž výpočetní technika dobře zvládá i tento obrovský objem numerických výpočtů.

Samozřejmě naše práce není vyčerpávající, neboť ve Fourierovu řadu jsme např. rozvedli pouze kmity matematického kyvadla s amplitudou 179°. Bylo by možno podrobně zkoumat i kmity s amplitudou menší, i když v tomto případě bychom s velkou pravděpodobností zjistili, že amplitudy vyšších harmonických frekvencí velmi rychle konvergují k nule a že kmity složené tedy přecházejí v kmity jednoduché – harmonické.

Stálo by také za úvahu zabývat se i kmity fyzikálního kyvadla. Protože však i v tomto případě je jeho úhlové zrychlení přímo úměrné $\sin(\varphi)$, nelze kromě jiné frekvence očekávat žádné zásadně odlišné výsledky. Dále by nebylo v zásadě nemožné zahrnout do celkové bilance sil i síly odporové ($F \approx v$, $F \approx v^2$) a zkoumat tak kmity tlumené, resp. kmity v nehomogenním gravitačním poli.

Na základě toho se domníváme, že naše práce je přístupná pro každého středoškoláka se zájmem o matematiku, fyziku a výpočetní techniku a že mu rozšíří jeho poznatky o matematickém kyvadle a o harmonickém a neharmonickém kmitání.

5. Seznam použitých zkratek

m … hmotnost
g … tíhové zrychlení
φ … odchylka závěsu kyvadla od svislého směru (okamžitá výchylka vyjádřená v radiánech)
φ_m … amplituda výchylky
s … délka oblouku kružnice
l … délka kyvadla
F … síla
y … vzdálenost kmitajícího hmotného bodu od svislého směru (okamžitá výchylka vyjádřená v metrech)
ω … úhlová frekvence
f … frekvence
T … perioda
a … zrychlení
ε … úhlové zrychlení

τ … poloviční perioda

6. Přílohy

Soubor „Kyvadlo příloha“ v tabulkovém procesoru Microsoft Excel je pouze v elektronické formě, protože je velmi rozsáhlý (cca 1000 řádků).

7. Použité zdroje

Urgošík B.: Fyzika. SNTL Praha 1981
Rektorys K. a kol.: Přehled užité matematiky. SNTL Praha 1981
Hlavička A.. a kol.: Fyzika pro pedagogické fakulty. SPN Praha 1981

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

CASEMODDING „RED & BLACK“

David Bareš, Jakub Kolačný
SSPŠ (Smíchovská střední průmyslová škola)
Preslova 25, Praha 5 - Smíchov

1. Obsah

1. Obsah ... - -
2. Seznam obrázků ... - -
3. Seznam příloh ... - -
4. Úvod ... - -
5. Teorie ... - -
5.1. Casemodding můžeme dělit na různá odvětví: ... - -
5.1.1. Optický casemodding ... - -
5.1.2. Chlazení ... - -
5.1.3. Přetaktování ... - -
5.2. Z hlediska casemoddingu se počítačové skříně dělí na: ... - -
5.2.1. „Sériovky“ ... - -
5.2.2. „HomeMade case“ ... - -
5.2.3. „Zmoddingované sériovky“ ... - -
6. Popis práce ... - -
6.1. Zadání ... - -
6.2. Návrhy ... - -
6.3. Měření ... - -
6.4. Řezání, vrtání a broušení ... - -
6.5. Tmelování ... - -
6.6. Barvení ... - -
6.7. Kompletace ... - -
6.8. Závěr ... - -

2. Seznam obrázků

Obrázek 1 - Kostra skříně + přední maska ... - -
Obrázek 2 - Prostředí 3DS STUDIO MAX ... - -
Obrázek 3 - Měření součásti ... - -
Obrázek 4 - Přímá bruska ... - -
Obrázek 5 - Přední maska při tmelování ... - -
Obrázek 6 - Nanášení barvy sprejem ... - -
Obrázek 7 - Vše pohromadě ... - -
Obrázek 8 - Hotový case ... - -

3. Seznam příloh

7. Přílohy ... - -
7.1. Tabulka porovnání cen ventilátorů ... - -
7.2. Tabulka porovnání cen Cold Cathode ... - -
7.3. Anglická stránka projektu ... - -
7.4. Návrhy ... - -
7.5. Leták CZ ... - -
7.6. Leták ENG ... - -

7.7. Zadání.....
7.8. Prezentace..... - -

4. Úvod

Casemodding, nebo-li úprava počítačové skříně v sobě skrývá spoustu možností a je hlavně založen na tvořivosti a fantazii každého, kdo se do nějakého takového projektu pustí. Jedná se většinou o změnu vzhledu, ale pozadu nezůstává ani celková přestavba a rozmístění komponentů uvnitř skříně. Možností na úpravy je opravdu hodně. Od nepatrných změn, jako je přidání jednoho ventilátoru pro lepší chlazení a když je osvětlen barevnou LED diodou tak i pro lepší vizuální efekt, až po složité úpravy kdy na celé skříni nezůstane nit suchá. Vyřezávání bočnice skříně a změna barvy s nasvícením různobarevných světel, vytváří krásný požitek pro naše oči. Je li počítač navíc chlazen sadou ventilátorů, tak je to nejen krásný, ale hlavně také velice účinný case.

5. Teorie

Casemodding je spojení dvou anglických slov: ***case***, neboli počítačová skříň, a ***modding***, neboli přeměna, úprava. Takže jde o upravování počítačových skříní. Podstatou casemoddingu je odlišit svoji počítačovou skříň od jiné, takže každá case je po úpravách jiná a naprosto jedinečná, neboť každý člověk je jiná individualita a tudíž má i jiný vkus. Tím vznikají naprosto jedinečná díla. Samotným upraveným skříním se říká ***casemody*** a člověk, který casemod vytváří je ***casemoder***. Často se lze setkat i s označením ***PC tuning***. Samotný tuning ale znamená především úpravy osobních aut. Proto tedy casemodding.

5.1. Casemodding můžeme dělit na různá odvětví:

5.1.1. Optický casemodding

Přidávání různých doplňků do case, upravování toho, co se od výrobce nelíbilo, změny velikostí oken v bočnici nebo vyřezání svého vlastního, změny barev, různé malůvky, okna z průhledných nebo průsvitných materiálů, nejznámější a často přeceňovanou úpravou jsou všelijaká svítidla od obyčejných LED po UV trubice. Zde se totiž uplatní rčení že: „Někdy míň, je víc“

5.1.2. Chlazení

Zlepšené odvádění teplého vzduchu ze skříně nejčastěji přidáváním ***fanů*** (ventilátorů), což je taky nejčastější chyba, víc větráků neznamená lepší větrání… Modeři se uchylují k přeuspořádání komponent uvnitř case, efektivnějšímu umístění větráků, nahrazení malých vysokootáčkových za větší nízkootáčkové nebo za stejné rozměry ovšem kvalitnější s garancí menší hlučnosti. Krokem do jiné oblasti, ale o to zajímavější, je vodní chlazení - ***vodník***. Toto chlazení je ve většině případech o dost tišší a také hezčí než chlazení ventilátory. Nevýhodou je vyšší pořizovací cena.

5.1.3. Přetaktování

Zvyšování frekvence procesoru a další „ladění“ komponent v počítači. Při přetaktování však může dojít k poškození nebo zničení dané komponenty (modernější komponenty se proti tomu většinou umí dobře bránit). Většinou se komponenty přehřívají a jejich základní chlazení nestíhá odebírat přebytečné teplo. Tím může dojít

ke spálení a tím k nepoužitelnosti komponentu. Zde platí, že kdo nemá vůbec ponětí o tom, jak by se mělo přetaktovávat, tak by se do toho sám neměl pouštět.

5.2. Z hlediska casemoddingu se počítačové skříně dělí na:

5.2.1. „Sériovky“

Skříně bez moddingových doplňků, běžně prodávané a vyráběné ve velkém. Casemodér je sám dodatečně upraví podle svých představ. Nejvíce používané skříně pro casemodding.

5.2.2. „HomeMade case“

Skříň od základů postavená casemodérem, nejčastěji z nějakého netradičního materiálu. Každý kus je jedinečný. Zde se naplno projeví potenciál casemodera, a tak se můžeme těšit na zcela nevšední výtvory.

5.2.3. „Zmoddingované sériovky“

Case, které jsou již od výrobce upravené a vybavené různými doplňky pro casemodding. Většinou již nepotřebují další úpravy.

Jako zdroj některých informací posloužil server **wikipedia.org** a my mu tímto mu děkujeme.

6. Popis práce

6.1. Zadání

Ve škole jsme dostali poměrně dost starý a ošklivý ***bigtower.*** To je case skoro 2x větší než je normální case. Nejdřív se nám vůbec nelíbil, protože to bylo něco naprosto strašného. Velká hromada plechu, která sotva držela pohromadě a vypadala, že se dřív než ji stihneme upravit rozpadne a nezbude z ní nic než kopa smetí. To nás ale přimělo k velkému odhodlání, že z takovéhle vykopávky uděláme skvostný case, který bude chloubou školy.

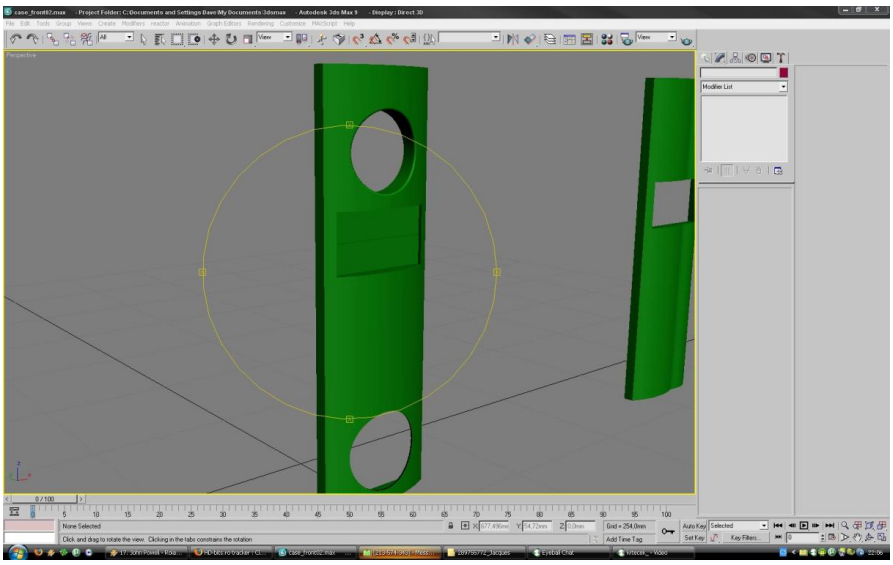


Obrázek 1 - Kostra skříně + přední maska

6.2. Návrhy

Ze všeho nejdřív jsme museli vymyslet, jak bude náš budoucí case vypadat. Návrhů a nápadů bylo mnoho, a tak jsme z nich vybrali 3 nejlepší, a ty si vytvořili v programu **3DS STUDIO MAX.** Tento program se používá pro modelování různých věcí, budov, přírody, ale i postav a různých animací a simulací pohybů. Tyto modely nám posloužili k rozhodnutí který z návrhů zhotovíme. Poté jsme si museli obstarat materiál a pracovní pomůcky. V tomto nám dost pomohla naše škola, která nám na nákup poskytla potřebné finance. Když bylo

vše nakoupeno, mohli jsme konečně v naší dílně, kterou jsme si vytvořili ze sklepa, vrhnout do manuální práce. Vše jsme museli vzít systematicky popořadě.



Obrázek 2 - Prostředí 3DS STUDIO MAX

6.3. Měření

Nejdříve jsme si museli vše naměřit. Různých pomocných čar a bodů bylo požehnaně, a tak jsme důležité body a hrany vyznačili černým nebo zeleným lihovým fixem. Jelikož se zlomilo dlouhé pravítko, museli jme vzít za vděk rolovacímu, skládacímu a krejčovskému metru. Později se našlo využití i pro posuvné měřítko Dlouhé čáry jsme rýsovali podle hrany plexiskla. Vše se ale povedlo naměřit a označit a tak jsme mohli postoupit na další fázi.



Obrázek 3 - Měření součásti

6.4. Řezání, vrtání a broušení

Vesměs na vše jsme použili přímou brusku, do které jsme nasazovali nástavce podle druhu práce, kterou jsme vykonávali. Na řezání jsme používali brusné kotouče, které se jako horký nůž máslem zařezávali do tvrdého plechu. Bohužel, občas se stalo, že se nám kotouček zlomil. Na řezání se spotřebovalo nespočet kotoučků. Při vrtání jsme použili frézu, jelikož vrták, který byl v sadě, se nám podařilo brzy zničit. A nakonec broušení si vyžádalo

služby brusných válečku. Po provrtání menších otvorů frézou jsme použili většího vrtáku, který jsme měli zasazen do elektrického šroubováku. Otvory po fréze byli totiž na šrouby malé, a tak potřebovaly rozšířit. Vše jsme pak ještě přebrousili smirkovým papírem, aby nebyly ostré hrany. Nejobtížnější část bylo spodní otevírací okno. Museli jsme ho nejdřív vyříznout z celé bočnice, poté vyříznout další menší okno uvnitř a přitom dbát na tloušťku hran, aby se nezlomilo, ale aby také zbylo dost míst na plexisklo, protože v okně je ještě zasazeno madlo na otevírání. Aby se dvířka otvírala přidělali jsme je na panty, které nám dali při instalaci pořádně zabrat. Vše muselo být naprosto přesné, aby to krásně sedělo. Nepotřebné kusy bočnice jsme odřezali a panty se krásně vešly. Aby se dvířka neotvírala samovolně přidělali jsme na spod bočnice magnety. Na řadu přišla přední maska skříně.



Obrázek 4 - Přímá bruska

6.5. Tmelování

Jednou z největších a nejtěžších změn bylo bezpochyby kitování (tmelování) přední masky. Podle vybraného návrhu jsme měli za úkol vyrovnat povrch a z jemné vlnky tak vytvořit mohutný rádius. Aby toho nebylo málo, v horní i spodní části bylo potřeba udělat dvě kruhové díry o průměru 11cm. U spodní části jsme začali vyřezáním díry pro ventilátor. Poté jsme uřízli kousek novodurové trubky a kitem ji přidělali k masce. Nakonec jsme postupným nanášením kitu vytvořili přesný rádius od kraje ke kraji. Horní část jsme začali vyřezáním menších rádiusů do 1. a 4. záslepky na mechaniky, přiložili jsme kousek duralu tak, aby pokrýval všechny horní 4 pozice na mechaniky a kitem s troškou silikonu jsme dural napevno přidělali. Nakonec jsme do samotného duralu vyřízli otvor o daném průměru a vše pečlivě zakitovali a vyhladili případné nerovnosti. Pro dosáhnutí ještě jemnějšího povrchu jsme celou část přestříkli plničem, který se postaral o menší povrchové nerovnosti. Posledním krokem byla základová barva, v našem případě matně černá.



Obrázek 5 - Přední maska při tmelování

6.6. Barvení

Jakmile bylo vše připraveno, na řadu přišlo barvení. Vše jsme ještě jednou pečlivě přebrousili smirkovým papírem, aby pozdější nástřik držel na materiálu, a začali jsme nanášet barvu. Všechny části, které má náš case se krásně vyrýsovali. V novém kabátu nyní vypadá mladě a svěže. Sice nás výpary ze sprejů s barvou pomalu posouvaly do mdlob, ale s přestávkami jsme úspěšně nastříkali celý case. Po zaschnutí barvy jsme se vrhli na konečně práce.



Obrázek 6 - Nanášení barvy sprejem

6.7. Kompletace

Začali jsme opět k sobě šrouby přidělávat plexiskla a bočnici, a dvířka krásně zapadla do pantů. Umístili jsme 2 rudé katody, které dodají šmrnc a záři. Přišroubovali všechny komponenty počítače. Do pozic zasadili a šrouby upevnili ventilátory a nasadili přední masku. Přišroubovali jsme bočnici ke kostře a bylo hotovo.



Obrázek 7 - Vše pohromadě

6.8. Závěr

Z počítačové skříně, kterou jsme dostali na začátku projektu jsme vytvořili case, který má nejen skvělý design a krásné křivky, ale také skvěle chladí komponenty jemu svěřené. Jsem rádi, že nám bylo umožněno věnovat se casemoddingu, protože nás to baví, a tak jsme se chtěli podělit s našimi zkušenostmi i s ostatními.



Obrázek 8 - Hotový case

7. Přílohy

7.1. Tabulka porovnání cen ventilátorů

Noctua NF-S12-800		
Prodejce	**Cena (včetně DPH)**	**Koupeno**
Czc.cz	392,--Kč	ANO
Alz.cz	415,--Kč	NE
Lan-Shop.cz	401,--Kč	NE

(ceny jsou uvedeny k datu 9.4.2008)

7.2. Tabulka porovnání cen Cold Cathode

Revoltec Cold Cathode Twin-Set Red, 30cm		
Prodejce	**Cena (včetně DPH)**	**Koupeno**
Czc.cz	189,--Kč	ANO
Eo.cz	976,--Kč	NE
Obchodnet.cz	298,--Kč	NE

(ceny jsou uvedeny k datu 9.4.2008)

7.3. Anglická stránka projektu

English Summary - Casemodding

Our task is casemodding. That means to improve a computer case to be the most attractive and beautiful thing in your room. Your normal case usually looks like a grey box with no beautiful design. We have made it better. We have changed the visual aspect and air circulation.

We have got a bigtower to modify. The bigtower is two times bigger than a normal case. We wanted to change the ugly white colour and add 3 ventilators to make better air circulation and 2 red diodes. The needed steps were changing the colour inside and outside, the placement of the component and upgrading the side and the front board.



It was necessary to cut, sharpen, paint and cement and to transport the holder of hard drive disk to the other place for better cooling and to make position for our new 3 ventilators. Other needed techniques were cutting into side and placing plexiglass, fixing 2 red cathode diodes with 4 pin molex connector to reach a nice red shine from our case. At last we have made completely new look of the front side. It is SECRET ☺

The most difficult part is the modification of the front side. It includes many changes and it is time consuming. We hope it will be the nicest case you have ever seen.

7.4. Návrhy







7.5. Leták CZ


WWW
PUPPET MODIFICATIONS
COM
Co pro Vás můžeme udělat...
úprava omezená pouze fantazií
3D vizualizace Vašeho nápadu
kompletní odhlučnení PC
a to vše za...
použití nejlepších materiálů
velmi krátkou dobu
skvělou cenu
...už jste na cestě k nám?
Puppet Modifications
Preslova 25
Praha 5 - Smíchov
150 21
Tel.: 257 320 533
Fax: 257 316 416

7.6. Leták ENG


WWW
PUPPET MODIFICATIONS
COM
What we can do for You...
mod limited only by imagination
3D visualization of Your idea
full noise elimination of PC
and all of that with...
using the best quality materials
for very short term
and for friendly cost
...are You already coming?
Puppet Modifications
Preslova 25
Praha 5 - Smíchov
150 21
Tel.: 257 320 533
Fax: 257 316 416

7.7. Prezentace

Casemodding

Projekt 2007/2008

- **Made by:**
  David Bareš
  &
  Jakub Kolačný
- **Garant:**
  Lukáš Bayer

Proč to děláme?

1. Máme zájem o vytvoření svého vlastního vzhledu
2. Chceme case připravit pro výkonnou sestavu
3. Manuální i grafická práce nás baví

Ukázka několika návrhů

- Vizualizace jsme dělali v 3ds Max Studiu 9
- Poté jsme vybrali jeden a zrealizovali ho
- Výsledek posuďte sami

Case I.



Case II.



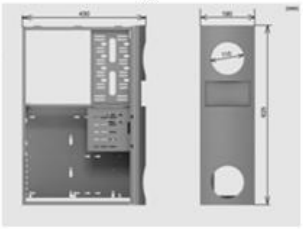
Case III.



Realizace návrhu č. III.

- Realizace včetně návrhu trvala cca 90 dní
- Výroba probíhala ve sklepních prostorech
- Na vytvoření padlo 8 brusných kotoučů, 3 vrtáky, moc kapek krve a mnoho nadávek

Rozměry, aby se vědělo kam se to vejde



Fotky, aneb jak to vznikalo

- Na následujících snímcích uvidíte, že to byla zábava, ale i strádání



Bylo potřeba vše změřit a nakreslit



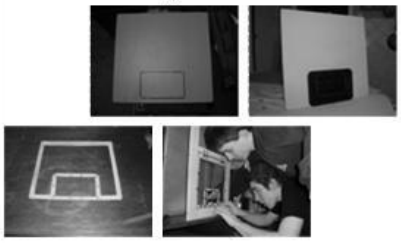
Vyřezávání oken



Přebytečná konstrukce musí pryč



Tvorba spodních dvířek



Předek mluví za vše…



Jde se stříkat



Kompletace



Menší bilance

- Spotřeba materiálu: 3 plechovky polyester. tmelu, 4 spreje (celkem 2 litry barvy), troška silikonu, kousek duralu, novodur. trubka a zbruba 2 m^2 smirk. papíru
- Vybavení: 3x ventilátor Noctua 120mm, 2x červená katoda 30cm
- Celková cena: 3000,-- Kč

K čemu nám to bylo dobré?

- Víme, že nářadí nám není cizí
- Umíme si vytvořit unikátní case
- Naučili jsme se, že spolupráce je hlavně kompromis

Děkujeme garantovi projektu za pomoc, škole za finanční prostředky a svému okolí za trpělivost po dobu trvání projektu☺

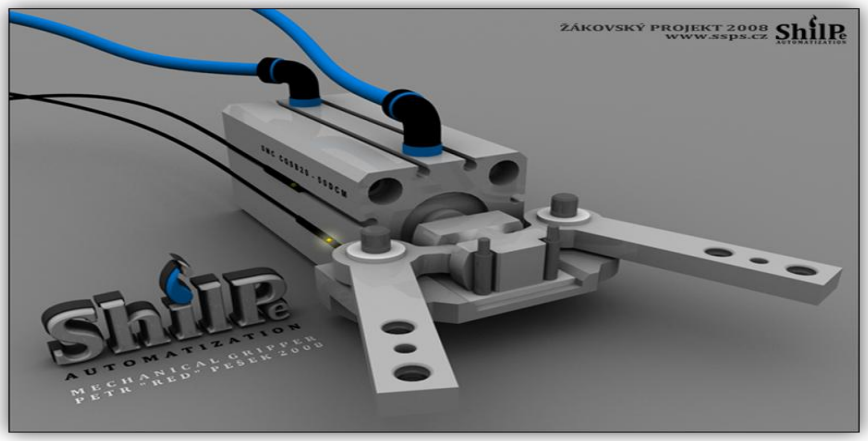
Děkujeme za pozornost

Středoškolská technika 2008

Setkání a prezentace prací středoškolských studentů na ČVUT

PNEUMATICKÉ CHAPADLO MANIPULÁTORU

Petr Pešek, Ondřej Šilhavý
SSPŠ (Smíchovská střední průmyslová škola)
Preslova 25, Praha 5 - Smíchov



KAPITOLA 1 Obsah

KAPITOLA 2 Úvod do projektu

Zadání:

Vytvořte konstrukci dvouprstého chapadla pro manipulátor s použitím pneumatického motoru, který vám bude k dispozici. Upínací síla je dána výkonem pneu-motoru.

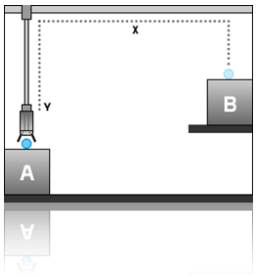
Téma:

Tvorba technické dokumentace potřebné pro výrobu zařízení

Požadavky:

Navrhnout 3 varianty řešení – funkce, nákresy, výhody / nevýhody
Výpočty
Výkres sestavy
Výrobní výkresy
Technologický postup
Vzduchové schéma + ovládací program
Vypracovat jazykovou část
Vytvořit prezentaci výrobku

KAPITOLA 3 Analýza chapadel



Chapadlo je systémový výrobek pro montážní a manipulační techniku. S použitím manipulačních jednotek, na které může být chapadlo připevněno, docílíme menší manipulační linky, která bude z bodu „A“ přenášet objekty do bodu „B“

viz. Obrázek

PODMÍNKY KONTRUKCE:

Dvojčinný pohon s pístem
Samostředící funkce
Oboustranný úchop (vnitřní / vnější)
Snímače polohy pístu – integrace čidel do bočních drážek
Možnost nastavení vnějších palců pro uchopování různých předmětů

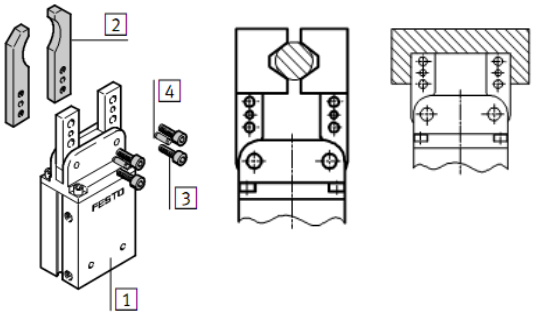
DRUHY DVOUPRSTÝCH CHAPADEL:

Úhlová chapadla
Radiální chapadla
Paralelní chapadla

VNĚJŠÍ PALCE

Možnosti upevnění vnějších palců chapadla (vlastní výroba)

1 úhlové chapadlo
2 vnější palec chapadla
3 upevňovací šrouby
4 středicí kolíčky



KAPITOLA 4 Návrhy variant řešení

Varianta č.1 – ÚHLOVÉ CHAPADLO

Všeobecné údaje

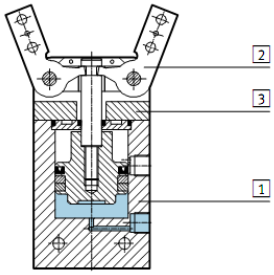
- dvojčinný pohon s pístem
- samočinně se středí
- různé směry úchopu:
  - vnější/vnitřní
- mnohostranná díky vnějším adaptibilním palcům chapadla
- velké množství možností adaptace pro pohony
- konstantní moment úchopu v celém rozsahu
- úhel rozevření 40°
- pevné vnitřní škrcení
- čidla:
  - adaptabilní čidla pro malá chapadla
  - u středních a velkých chapadel lze integrovat čidla

Obecné technické údaje						
velikost		10	16	25	32	40
konstrukce		páka				
způsob činnosti		dvojčinný pohon				
funkce úchopu		úhlová				
počet čelistí		2				
úhel rozevření	[°]	40				
připojení pneumatiky		M3		M5	G⅛	
opakovatelná přesnost[1]	[mm]	≤ 0,04				
max. přesnost při výměně	[mm]	0,2				
max. pracovní frekvence	[Hz]	4				
snímání poloh		čidly na válce (objednávají se zvlášť)				
upevnění		vnitřním závitem a středicím otvorem				

1) Rozptyl koncových poloh při stálých okolních podmínkách při 100 po sobě následujících zdvizích ve směru pohybu čelistí chapadla.

Provozní a okolní podmínky		
min. provozní tlak	[bar]	2
max. provozní tlak	[bar]	8
provozní médium		filtrovaný stlačený vzduch, mazaný nebo nemazaný
teplota okolí	[°C]	+5 ... +60
odolnost korozi KBK[1]		2

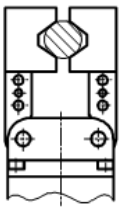
funkční řez

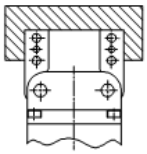


čelisti

Bočnice

Tělo





velikost	10	16	25	32	40
celkový moment úchopu					
rozevření	25	120	360	680	965
sevření	22	106	320	600	880

Varianta č.2 – PARALELNÍ CHAPADLO

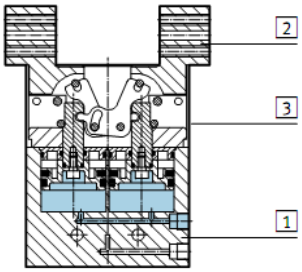
Všeobecné údaje

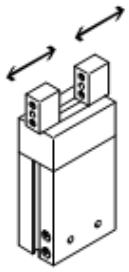
- dvojčinný pohon s pístem
- samočinně se středí
- různé směry úchopu:
  - vnější/vnitřní
- mnohostranné díky možnosti adaptace palců
- velké množství možností adaptace pro pohony
- silný úchop v malém prostoru
- max. opakovatelná přesnost
- zajištění síly úchopu
- pevné vnitřní škrcení
- s ochranným krytem proti prachu pro použití v prostředí s nečistotami (stupeň krytí IP54)
- snímače:
  - adaptabilní čidla u malých chapadel
  - u středních a velkých chapadel lze čidla integrovat

Obecné technické údaje							
velikost		6	10	16	20	25	35
konstrukce		šikmá plocha	páka				
způsob činnosti		dvojčinný pohon					
funkce úchopu		paralelní					
počet čelistí		2					
max. tíha externího palce chapadla[1]	[N]	0,1	0,2	0,4	0,6	0,8	1,2
zdvih každé čelisti	[mm]	2	3	5	6,5	7,5	12,5
připojení pneumatiky		M3			M5	G1/8	
opakovatelná přesnost[2]	[mm]	≤ 0,04					
max. přesnost při výměně	[mm]	0,2					
max. pracovní frekvence	[Hz]	4					
snímání poloh		čidly (objednávají se zvlášť)					
upevnění		vnitřním závitem a středicí dutinkou					
		–	s průchozími otvory a středicí dutinkou				

Provozní a okolní podmínky								
velikost			6	10	16	20	25	35
min. provozní tlak	HGP-...-A/-B	[bar]	2					
	HGP-...-G...	[bar]	5					
max. provozní tlak		[bar]	8					
provozní médium			filtrovaný stlačený vzduch, mazaný nebo nemazaný					
teplota okolí		[°C]	+5 ... +60					
odolnost korozi KBK[1]			2	1				

funkční řez





Varianta č.3 – RADIÁLNÍ CHAPADLO

Všeobecné údaje

- dvojčinný pohon s pístem
- samočinně se středí
- různé směry úchopu:
  - vnější/vnitřní
- mnohostranná díky vnějším adaptibilním palcům
- velké množství možností adaptace pro pohony
- konstantní moment úchopu v celém rozsahu
- úhel rozevření 180°
- pevné vnitřní škrcení
- čidla:
  - adaptabilní čidlo pro malá chapadla
  - u středních a velkých chapadel lze integrovat čidla

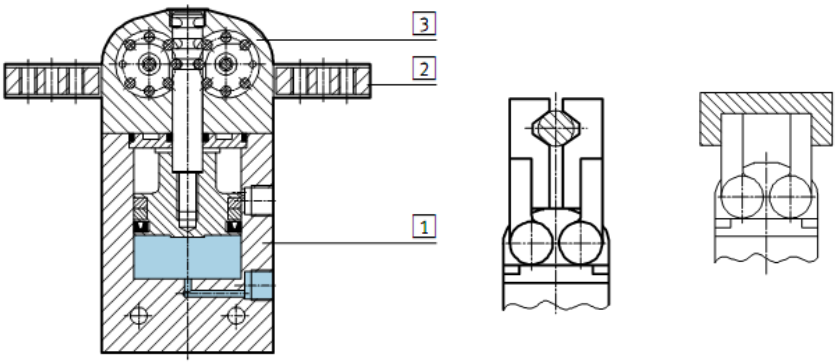
Obecné technické údaje									
velikost		10		16		25		32	40
konstrukce		ozubený řemen/pastorek							
způsob činnosti		dvojčinný pohon							
funkce úchopu		radiální							
počet čelistí		2							
úhel rozevření	[°]	180							
připojení pneumatiky		M3				M5		G1/8	
opakovatelná přesnost[1]	[mm]	≤ 0,1							
max. přesnost při výměně	[mm]	0,2							
max. pracovní frekvence	[Hz]	4							
snímání poloh		čidly na válce (objednávají se zvlášť)							
upevnění		vnitřním závitem a středicím otvorem							

1) Rozptyl koncových poloh při stálých okolních podmínkách při 100 po sobě následujících zdvizích ve směru pohybu čelistí chapadla

Provozní a okolní podmínky		
min. provozní tlak	[bar]	2
max. provozní tlak	[bar]	8
provozní médium		filtrovaný stlačený vzduch, mazaný nebo nemazaný
teplota okolí	[°C]	+5 ... +60
odolnost korozi KBK[1]		2

1) Třída odolnosti korozi 2 dle normy Festo 940 070:
konstrukční díly s mírnějšími nároky na odolnost korozi. Vnější viditelné části s požadavky především na vzhled povrchu, který je vystaven přímému kontaktu s okolní pro průmysl běžnou atmosférou, respektive látkami, jako jsou chladicí látky a maziva

funkční řez



KAPITOLA 5 Hodnocení variant

VLASTNOSTI	ÚHLOVÉ CHAPADLO	PARALELNÍ CHAPADLO	RADIÁLNÍ CHAPADLO
MANIPULACE	2	1	1
NÁROČNOST KONSTRUKCE	1	5	4
KINEMATIKA	1	3	2
CENOVÁ DOSTUPNOST	2	3	4
DESIGN	3	1	3
VYUŽITELNOST	1	1	1
Ø HODNOCENÍ	1,7	2,3	2,7

LEGENDA
1 – VYNIKAJÍCÍ
2 – VELMI DOBRÉ
3 – DOBRÉ
4 – DOSTAČUJÍCÍ
5 – NEVYHOVUJÍCÍ

ZVOLENÍ VARIANTY
Volíme variantu úhlového chapadla pro její jednoduchou kinematiku a ne příliš náročnou konstrukci. V kombinaci s naším dostupným motorem bude přímo ideální pro tvorbu chapadla. Celý mechanismus bude umístěn přímo na motoru a bude, sním spojen šrouby. Velikosti komponentů budou muset být dimenzovány tak, aby chapadlo vypadalo elegantně a zároveň plnilo svou funkci. Jako hlavní konstrukční materiál použijeme superdural a konstrukční ocel 11 500

VOLÍME VARIANTU č.1 -

KAPITOLA 6 Pohon

Dostupný typ pohonu:
Pneumatický lineární pohon řady kompaktních válců třídy CQS od firmy **SMC**.

CQSB20-50DCM

CQS = označení pro kompaktní válec
B = označení pro montáž válce – průchozí otvory / montážní závity
20 = průměr pístu (mm)
-
50 = zdvih (mm)
D = dvojčinný
CM = elastomerové tlumení a vnější závit na pístnici
Tento motor bude zakomponován přímo do sestavy chapadla a stane se jeho pohonnou jednotkou.
Maximální provozní tlak tohoto motoru je 1 MPa (10 bar)



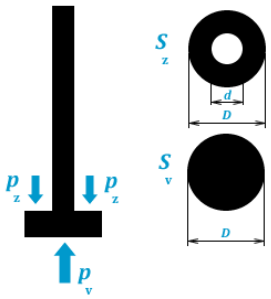
Přehled použitelných snímačů polohy/ podrobné informace naleznete v kapitole 10 - Snímače polohy pro pneumatické pohony

Typ snímače	Zvláštní funkce	Provedení	Indikace sepnutí	Počet vodičů (výstup)	Napětí DC	Napětí DC	Napětí AC	Snímač: Poloha kabelu kolmo k ose	Snímač: Poloha kabelu v ose	Délka kabelu* [m] 0,5 (–)	3 (L)	5 (Z)	Konektor M8/M12
jazýčkový	–	s kabelem	ne	2 vodiče	24V	5V, 12V	max. 100V	D-A90V	D-A90	●	●	–	–
			ano			12V	100V	D-A93V	D-A93	●	●	●	●
				3 vodiče (NPN)	–	5V	–	D-A96V	D-A96	●	●	–	–
elektronický	–	s kabelem	ano	3 vodiče (NPN)	24V	12V	–	D-M9NV	D-M9N	●	●	–	–
				3 vodiče (PNP)				D-M9PV	D-M9P	●	●	●	●
				2 vodiče				D-M9BV	D-M9B	●	●	–	●
	Signalizace stavu (dvoubarevná)			3 vodiče (NPN)				D-F9NWV	D-F9NW	●	●	●	–
				3 vodiče (PNP)				D-F9PWV	D-F9PW	●	●	●	●
				2 vodiče				D-F9BWV	D-F9BW	●	●	●	–

* Délka přívodního kabelu: 0,5 m ..... – např. D-A93
3 m ..... L např. D-A93L
5 m ..... Z např. D-A93Z

KAPITOLA 7 Kinematika mechanismu
ROZEVÍRÁNÍ ČELISTÍ = VYSOUVÁNÍ MOTORU

Výpočet plochy pístu



$$S_v = \frac{\pi D^2}{4} \qquad S_v = \frac{\pi 20^2}{4}$$

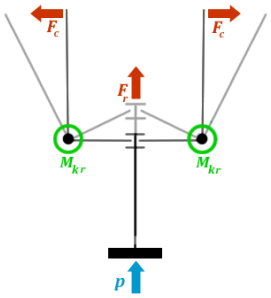
Výsledek $S_v = 314 mm^2$

Výpočet rozevírací síly

$$F_r = p \cdot S \qquad F_r = 0{,}6 \cdot 314$$

Výsledek $F_r = 188{,}4 N$

Výpočet kroutícího (rozevíracího) momentu, x = 15 mm



$$M_{kr} = F \cdot x \qquad M_{kr} = 188{,}4 \cdot 20$$

Výsledek $M_{kr} = 2826 Nmm$ => **282,6 Ncm**

Síla pro vnitřní úchop, y = 50 mm

$$F_c = \frac{M_{kr}}{y} \qquad F_c = \frac{2826}{50}$$

Výsledek $F_c = 56{,}52 N$

1 N = 0,102 kg/mm2 => 1 čelist při rozevírání vykoná sílu 5,77 $^{kg}/_{mm^2}$

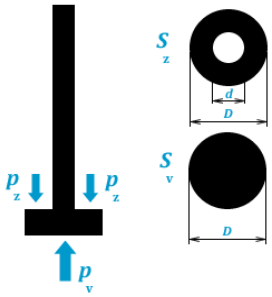
Moment rozevírání čelistí: 282,6 Ncm
Rozevírací síla na jedné čelisti: 56,52 N = 5,77 $^{kg}/_{mm^2}$

KAPITOLA 8 Kinematika mechanismu
SVÍRÁNÍ ČELISTÍ = ZASOUVÁNÍ MOTORU

Výpočet plochy pístu

$$S_z = S_v - \frac{\pi d^2}{4} \qquad S_z = 314 - \frac{\pi 10^2}{4} \qquad S_z = 314 - 78{,}5$$

Výsledek plochy pro zasouvání $S_z = 235{,}5 mm^2$

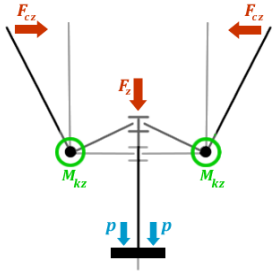


Výpočet svírací síly

$$F_z = p \cdot S \qquad F_z = 0{,}6 \cdot 235{,}5$$

Výsledek $F_z = 141{,}3N$

Výpočet F_{zy}, složky síly F_z



Víme, že úhel rozevření mezi čelistmi má být 40°,
Tedy 20° od čelisti k ose pístu. Složka ***F*py** je
Rovnoběžná s ramenem ***y*** a kolmá na rameno ***x***.
***F*z** je rovnoběžná s osou pístu.
Platí tedy že mezi složkou ***F*z** a ***F*zy** je úhel 20°

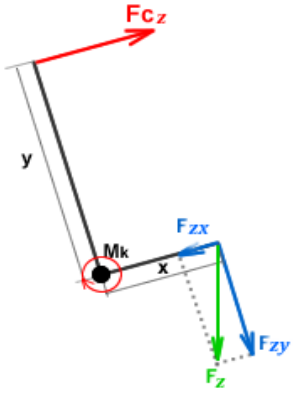
$$F_{zy} = F_z \cdot \cos\alpha \qquad F_r = 141{,}3 \cdot \cos 20$$

Výsledek $F_{ZY} = 132{,}78N$

Výpočet kroutícího (svíracího) momentu, ***x*** = 15 mm

$$M_{kz} = F_{zy} \cdot x \qquad M_{kz} = 132{,}78 \cdot 15$$

Výsledek $M_{kz} = 1991{,}7Nmm$ => **199,17Ncm**



Síla pro vnější úchop, ***y*** = 50 mm

$$F_{cz} = \frac{M_{kz}}{y} \qquad F_{cz} = \frac{1991{,}7}{50}$$

Výsledek $F_c = 39{,}834N$

1 N = 0,102 kg/mm2 => 1 čelist při rozevírání vykoná sílu 4,1 $^{kg}/_{mm}$

Moment svírání čelistí: 199,12 Ncm
Svírací síla na jedné čelisti: 39,384 N = 4,1 kg/mm2

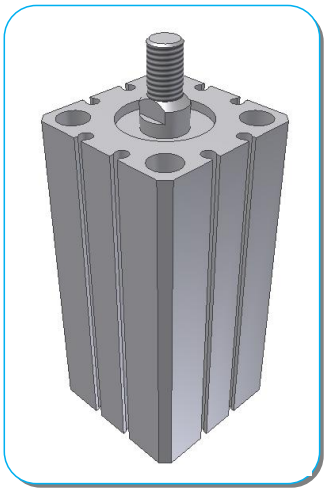
VIZ ANIAMCE : kinematika.avi

KAPITOLA 9 Navržení konstrukce

Při navrhování konstrukce jsme vycházeli z již hotového výrobku firmy FESTO – úhlového chapadla HGW

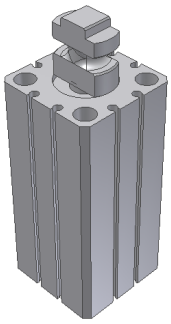
Kinematika pohybu čelistí je již vyřešen, základní rozměry čelistí (x = 15 mm y = 50 mm) jsou navrženy.

MOTOR



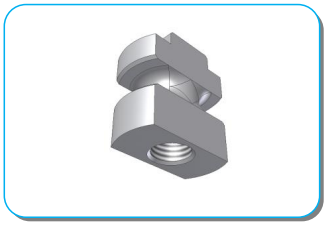
Pří výrobě celého produktu byl motor klíčivým vodítkem. Dimenzování ostatních komponentů se řídilo jeho rozměry.

Vysouvání pístnice muselo být omezeno tak aby čelisti mezi sebou svírali úhel 40° ± 5°. Musel být omezen pracovní zdvih motoru.



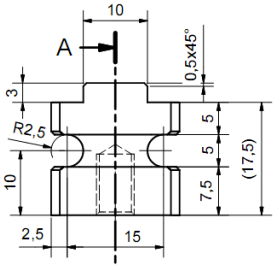
Celá konstrukce se skládá z následujících částí:
MOTOR
- o KÁMEN
- o ČELISTI
- o BOČNICE
- o ČEPY
- o TEFLONOVÉ VODÍCÍ LIŠTY
- o TEFLONOVÉ KROUŽKY
- o DORAZY
- o PLASTOVÁ KRYTKA
- o ŠROUBY

KÁMEN



Tento prvek přeměňuje lineární pohyb pístnice na rotační pohyb čelistí kolem čepů (momentů). Je zastavován dorazovými příčkami. Kámen má v sobě závit a je našroubován na vrch pístnice a díky tomu že má motor rotační stupeň volnosti pístu a pístnice, tak se může kámen nastavit do kterékoliv polohy.

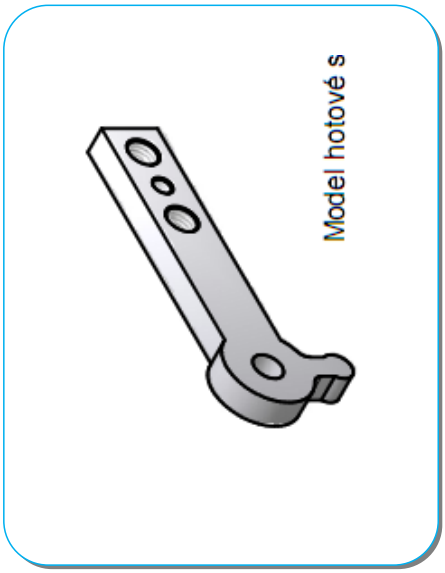
Materiál:



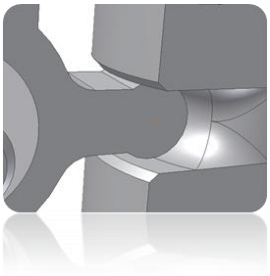
SUPERDURAL - označovaný jako Al-Cu4-Mg1 je to dural legovaný 1,5% hořčíku a 4% mědi a 1% manganu, tyto legující prvky mu dávají pevnost nad 550 MPa

Počet: 1ks

ČELIST

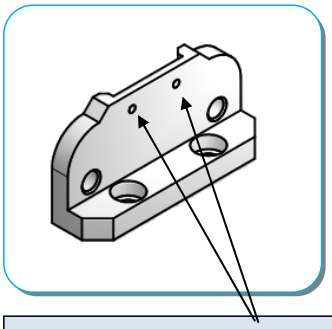


Asi nejsložitější tvar na výrobu
Důležité je zakončení ve styku s kamenem. Když se kámen pohybuje nahoru či dolu, tak vzniká tření mezi kamenem a zakončením čelisti.



Materiál:
Ocel 11 500 – konstrukční ocel
Počet: 2ks koeficient tření – 0,35 (dural/ocel)

BOČNICE



Bočnice byla zkonstruována tak aby pomocí teflonových vodících lišt vymezila kamenu rotační stupeň volnosti kolem osy pístnice. Její rozměr je dimenzován, tak aby byla zarovnána s motorem, a spojena sním šrouby. Potřebovali jsme docílit otáčení čelisti kolem pevného bodu a zároveň spolehlivé pojištění. Za normálních podmínek, by jsme volili klasický čep s tolerancí h11, kolem kterého by se čelist otáčela a čep by byl vsazen do díry a pojištěn pojistným kroužkem. Vytvoříme vlastní čep s přechodným uložením při styku s bočnicemi - **N7/h6.** Je to pevné uložení s nepatrnou vůlí nebo malým přesahem. Montáž uložení lisováním malou silou. Dorazy pojistíme stejnou tolerancí (Výpočet pevnostní podmínky pod obrázkem). Avšak na čelistech musíme docílit uložení s vůlí, aby se mohli otáčet. Zvolíme uložení - **G7/h6** což je Točné uložení s velmi malou vůlí pro přesné vedení hřídele. Konstrukce tak bude vypadat elegantně a splní pevnostní účely.

Materiál:
SUPERDURAL

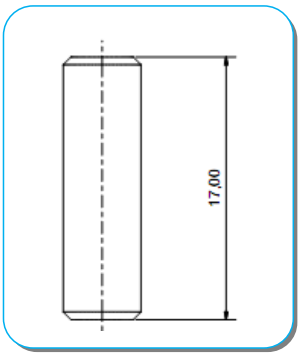
Počet: 2ks

Míjivé zatížení na tlak:
Pro slitiny hliníku: 50 až 90 Mpa
síla 190 N působící na plochu 4mm^2

$\sigma_K = F/S \leq \sigma_{DK}$
$\sigma_K = 190/4 \leq 50 \div 90$

$\sigma_K = 47,5 \leq 50 \div 90$ => vyhovuje

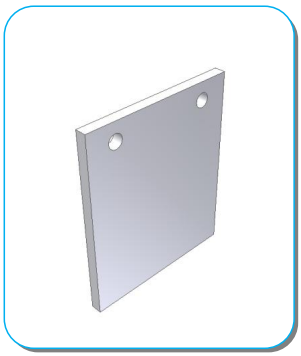
ČEPY



Vlastní čep s tolerancí **h6**
Počet: 2ks
Materiál:
Ocel 11 500

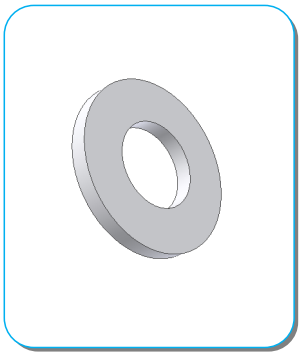
PŘED SESTAVENÍM – POMAZAT GRAFITEM STŘEDNÍ ČÁST

TEFLONOVÉ VODÍCÍ LIŠTY



Teflon má perfektní kluznou vlastnost, a proto jsme z něho vyrobili vodící lišty kamene. Jednak budou vymezovat vůli mezi kamenem a bočnicí, ale také budou minimalizovat třeni a odpor kamene při pohybu.
Počet: 2ks
Materiál:
Teflon

TEFLONOVÉ KROUŽKY



Stejná vlastnost jako u vodících líšt – vymezují vůle mezi bočnicí a čelistí – minimalizují tření při otáčení čelistí.
Počet: 2ks
Materiál:
Teflon

DORAZOVÉ PŘÍČKY

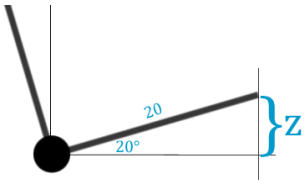


Dorazová příčka slouží k zastavení kamene v úseku, kdy čelisti dosáhnou vzájemný úhel cca 40°. Budou uloženy mezi bočnicemi se stejnou tolerancí jako čepy. Tolerance bočních vybrání – **h6**

Materiál:

Ocel 11 500

Dále bylo potřeba spočítat potřebný zdvih pístu, aby mezi čelistmi vznikl úhel 40° ± 5°



$$\sin\alpha = \frac{Z}{x}$$

$$Z = \sin\alpha \,.\, x$$

$$Z = \sin 20 \,.\, 15$$

$$Z = 5{,}13 \text{ mm} = 5 \text{ mm}$$

Dorazové příčky musíme umístit 5 mm nad dorazovou plochu kamene – dbát při konstrukci bočnice.

ŠROUBY

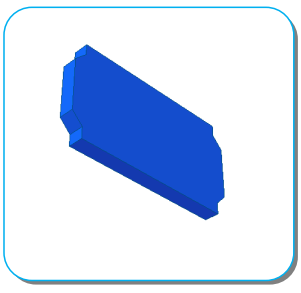


Upevnění bočnic k motoru provedeme šrouby s vnitřním šestihranem
– **ČSN 02 1143**

Rozměry: **ŠROUB M5 X 30 ČSN 02 1143**

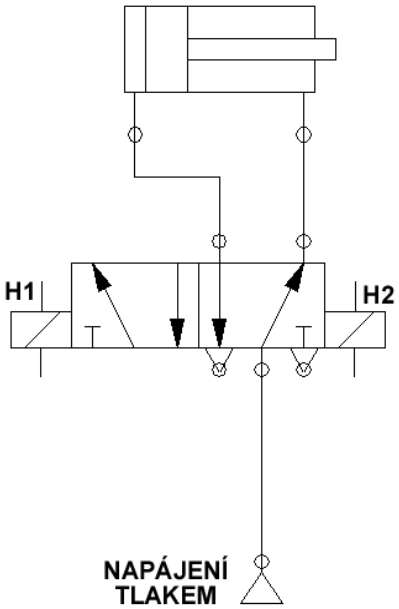
Počet: 4ks
Nemusíme počítat pevnostní podmínky, je logické že 4 x M5 šrouby bez problémů udrží konstrukci i s přenášeným objektem.

PLASTOVÁ KRYTKA



Slouží jako dekorativní prostředek, ale také proti dostání prachu do mechanismu chapadla.

KAPITOLA 10 Pneumatické schéma



Vzduchové schéma

- Lineární motor
- Bistabilní rozvaděč – ventily 5/2
- Vstupy H1 a H2

OVLÁDACÍ PROGRAM - FST

STEP 1; ROZEVŘENÍ ČELISTÍ

IF	S1	
THEN	SET	H1
	RESET	H2

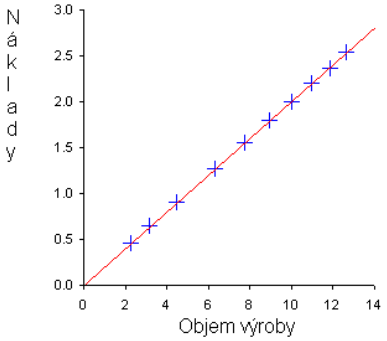
STEP 2; SEVŘENÍ ČELISTÍ

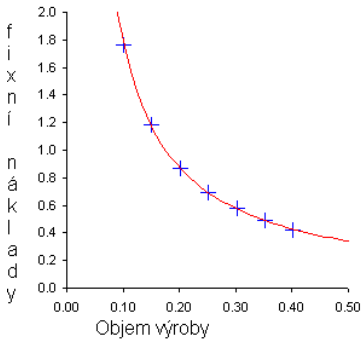
IF	S2	
THEN	SET	H2
	RESET	H1

S1 = tlačítko pro operaci H1
S2 = tlačítko pro operaci H2
H1= vstup vzduchu pro rozevření čelistí
H2= vstup vzduchu pro sevření čelistí

KAPITOLA 11 EKONOMIKA PROJEKTU

Projekt je navrhován pro kusovou výrobu, neboť u mechanických chapadel není předpokládán masový odběr. Konečná cena se odvíjí od částky, kterou si účtuje firma zabývající se přesným CNC obráběním (např. firma CNC Řezáč), dále od nákladů na montáž chapadla dohromady a popř. dovoz. Pokud by se uvažovalo o sériové výrobě, cena by se odvíjela od fixních nákladů, které zahrnují pronájem výrobní haly, měsíční cenu za elektřinu, plat zaměstnanců atd.
Sériová výroba by se vyplatila při situaci, kdy by se firma **ShilPe** zabývala více než jedním produktem.
Využití fixních nákladů na jednotku při sériové výrobě je znázorněno v grafu nalevo, pružné náklady na jednotku se vyvíjí dle grafu napravo





NÁKLADY PŘI KUSOVÉ VÝROBĚ

- Objednávka přesného vysoustružení a vyfrézování součástí chapadla od firmy CNC Řezáč sro. - Cca 10 000,--Kč
- Náklady na dopravu součástí z Plzně do Prahy, kde sídlí **ShilPe** 500,-- Kč
- Plat montážního dělníka, který se odvíjí od doby nutné k sestavení a otestování chapadla 100,-- Kč/h.
- Cena za dodávku chapadla do cílové firmy 0 – 1000,-- Kč

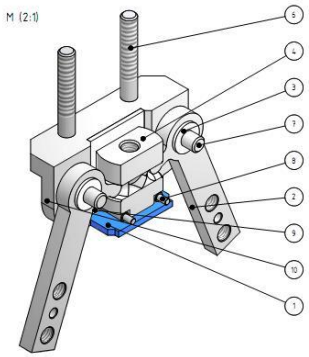
Celkové náklady cca 12000,-- Kč

Vzhledem k možným komplikacím při výrobě a částečnému zisku by se prodejní cena pohybovala kolem 18 000,-- Kč což pro firmy, kterým je náš produkt směřován není vysoká částka.

ShilPe chapadlo je výrobní pomůcka, která má vysokou životnost a svou funkcí slouží ke zjednodušení a zrychlení výrobních procesů, čímž šetří jak peníze, tak čas.

KAPITOLA 12 ENGLISH SUMMARY

- The aim of the project was to design a manipulator with two mechanic fingers, driven by a pneumatic engine from SMC company
- We also have to draw a technical drawing of whole manipulator and every part of it, make some basic economic calculations and make a presentation.
- Our manipulator works on the basis of two fingers rotating around one solid point.
- Construction is really easy, even when it is compared with some other types of manipulators from Festo company.
- It works just with one working piston.
- Fingers can be opened up to 40 degrees.
- The hardest thing on the whole projekt, was to figure out how to place so many parts on such a small place.
- Our manipulator is ideal for moving objects from place A to place B, not for welding or cutting operation.
- We have been co-working with Festo company, and our gripper was originally inspired by festo manipulator.
- These two fingers can have lot of extensions, it depends on the form of using.
- Since the first economic calculation, it has looked that our manipulator will be cheaper than the original one.

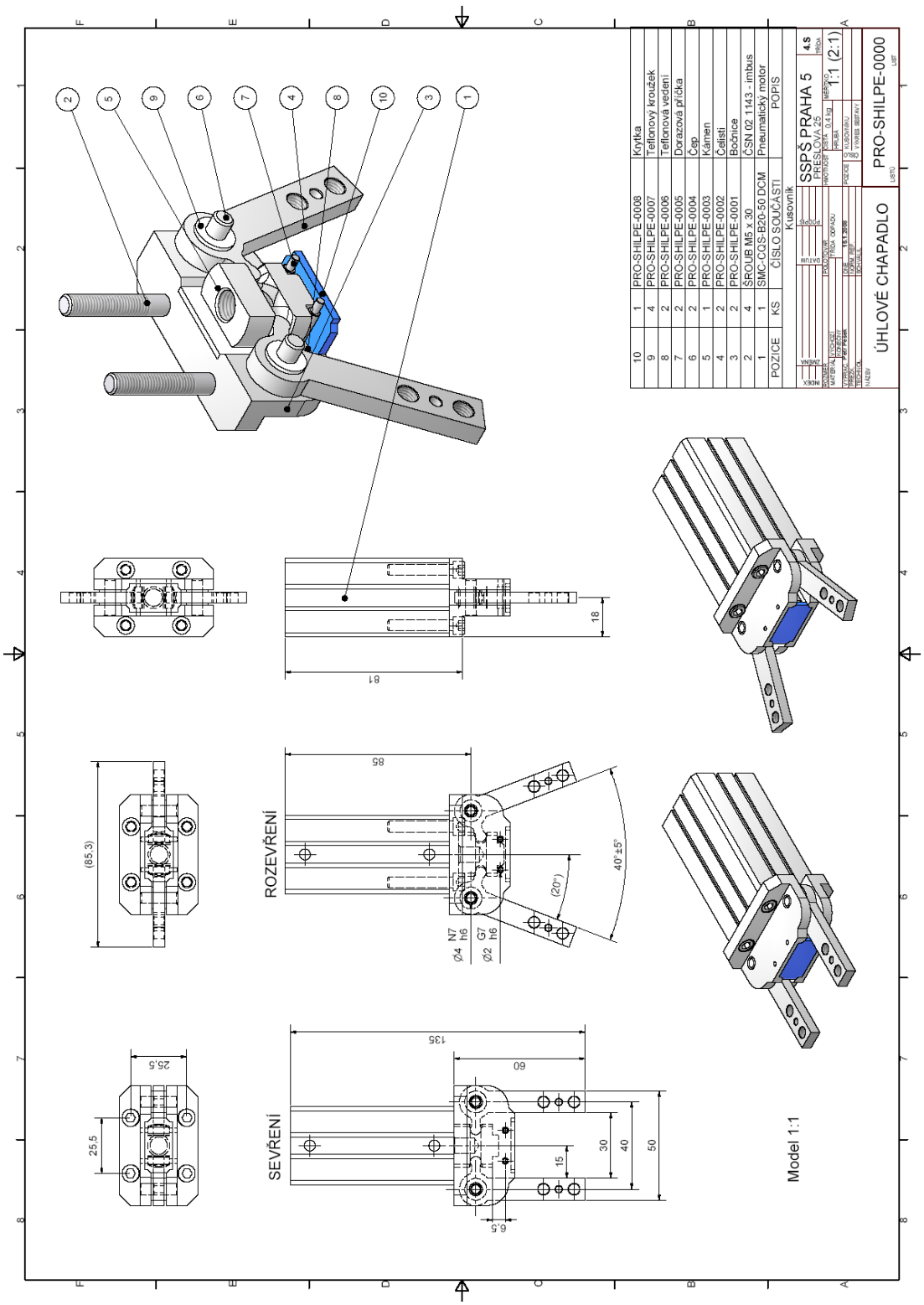


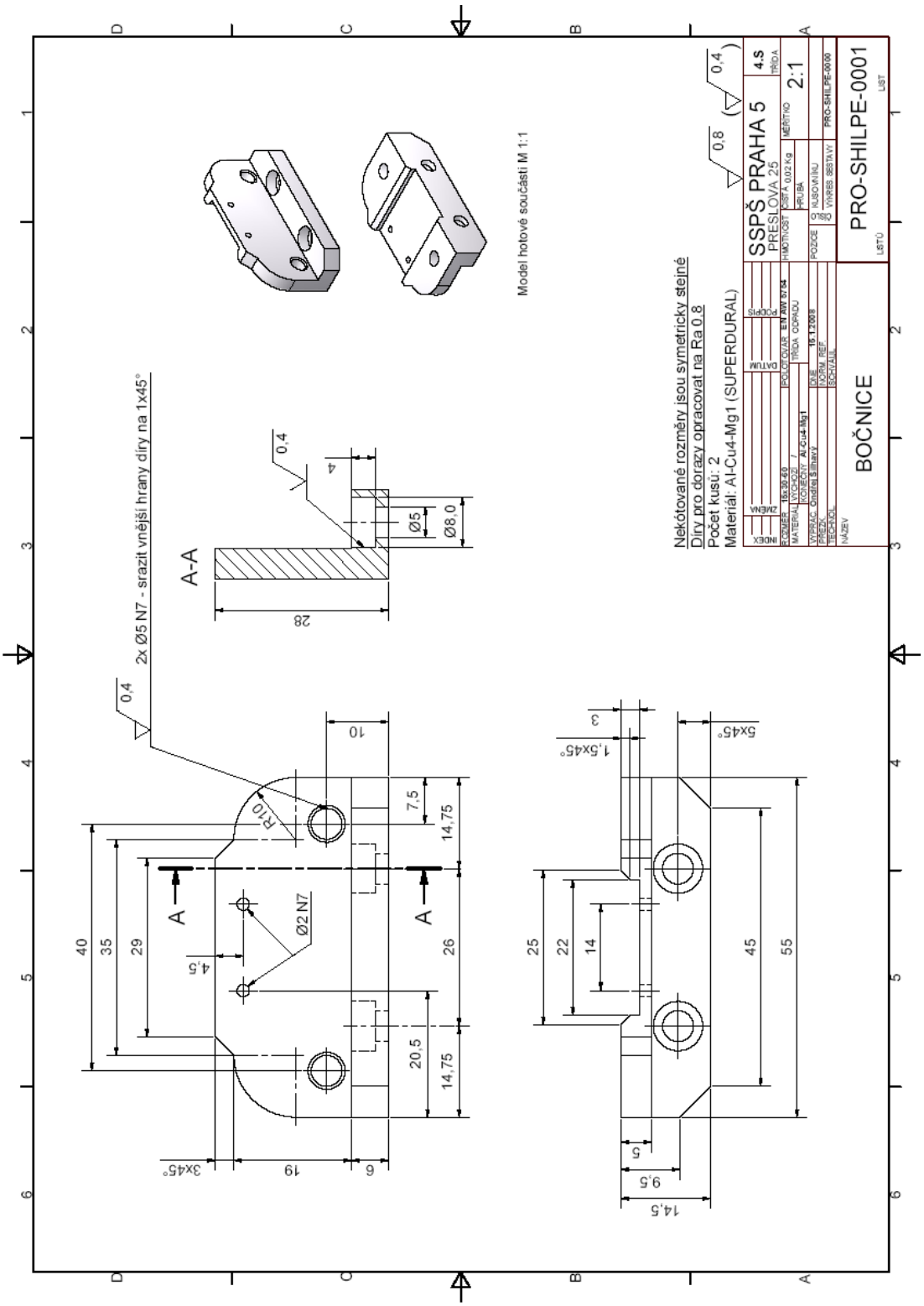
PARTS OF MECHANICAL GRIPPER

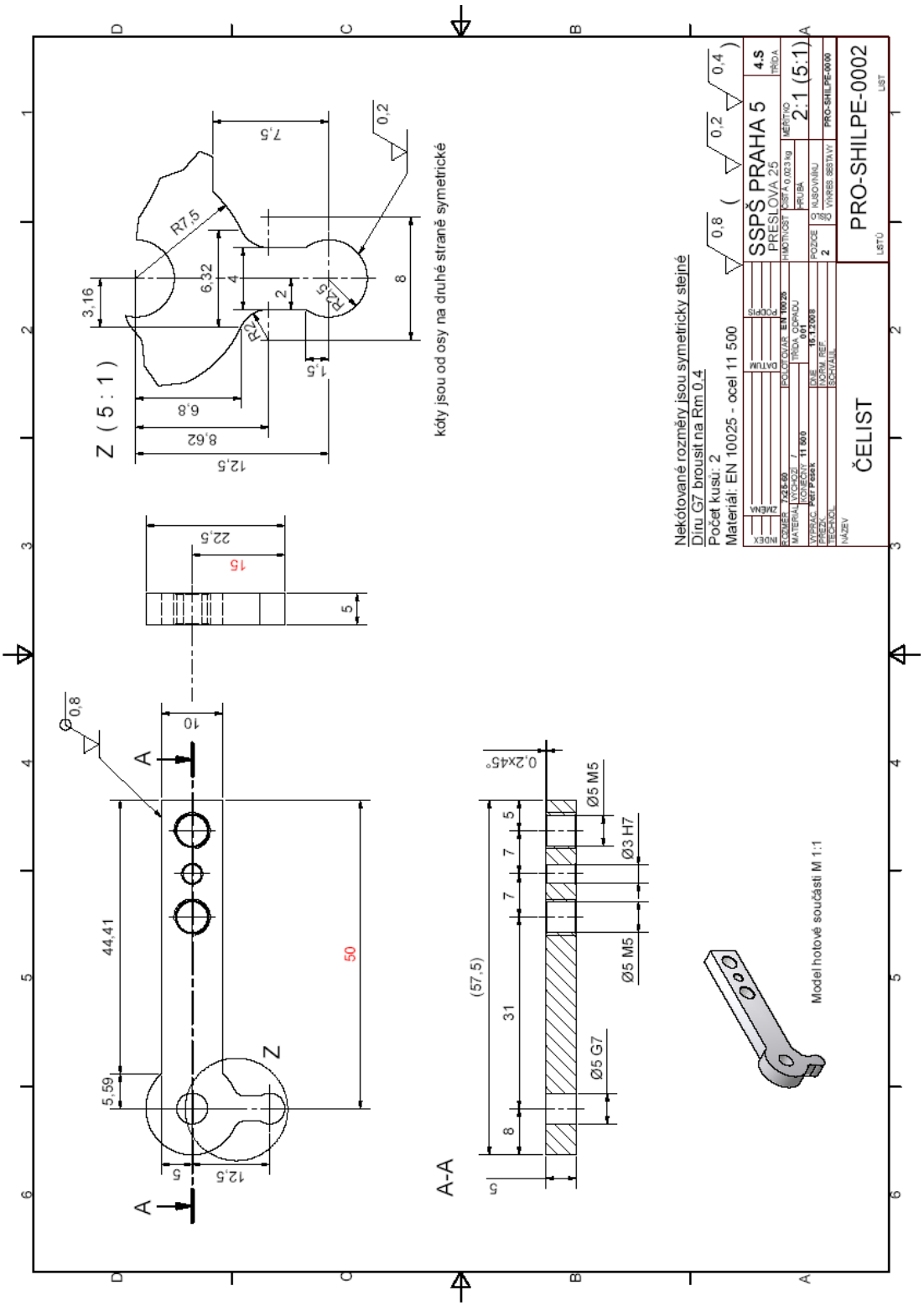
6 - Male screw
4 - Piston head
3 - Teflon underlay
7 - Peg
8 - Stick
2 - Universal Finger
9 - Teflon desk
10 - Side wall
1 - Covert

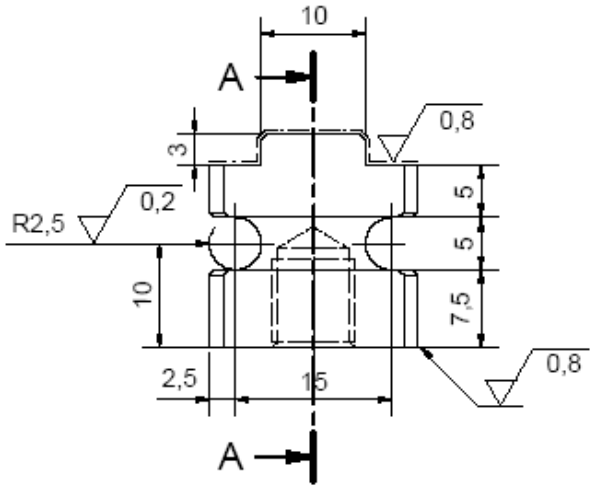
KAPITOLA 13 ZÁVĚR

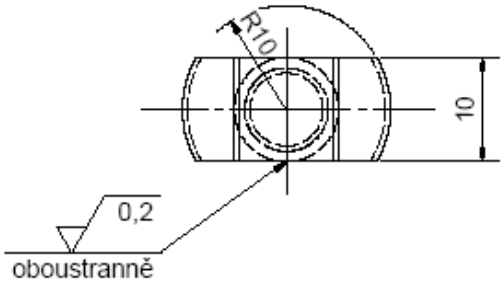
Na závěr, bychom rádi uvedli, že chapadlo může být vyrobeno podle výkresové dokumentace, za kterou osobně zodpovídáme. Technologické postupy jsou pouze orientační, předpokládáme, že celá výroba proběhne na NC strojích. Výrobek doporučujeme používat nejen jako učební pomůcku při automatizaci či mechanice, ale i v lehkém průmyslu pro přenášení drobných objektů. S celkovou prací jsme velice spokojeni a doufáme, že v nás zanechala poznatky, které v budoucí praxi využijeme. Touto cestou bychom chtěli poděkovat našemu garantovi a profesoru automatizace panu inženýru Vladislavu Štolbovi, za to, že nás seznámil se základy automatizační technologie i s produkty světových automatizačních firem. Děkujeme mu za trpělivost, ochotu a konzultování našich problému s projektem. Dále bychom chtěli poděkovat ostatním profesorům, zejména panu ing. Františku Humhalovi, za to že na nás neustále tlačil a díky němu máme základní znalosti z oborů strojní technologie a technického kreslení. Děkujeme naší třídní profesorce paní ing. Zuzaně Cimlerové, za zasvěcení a základní znalosti z předmětu stavby a provozu strojů, které jsme v našem projektu plně využili. Děkujeme paní ing. Anně Kabelové, za uvedení do oboru mechaniky, díky které jsme byli schopni vypočítat důležité hodnoty pro výrobu celého chapadla. V neposlední řadě bychom chtěli poděkovat paní ing. Eleně Chládkové, za seznámení s ekonomikou podniku a řešení projektu z finančního hlediska. Všem těmto i jiným profesorům velice děkujeme a doufáme, že jako poslední čistokrevní strojaři budeme tuto školku v budoucí praxi reprezentovat nejlépe, jak to půjde.

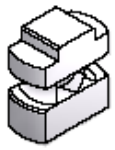

10 1 PRO-SHILPE-0008 Krytka
9 4 PRO-SHILPE-0007 Teflonový kroužek
8 2 PRO-SHILPE-0006 Teflonová vedení
7 2 PRO-SHILPE-0005 Dorazová příčka
6 2 PRO-SHILPE-0004 Čep
5 1 PRO-SHILPE-0003 Kámen
4 2 PRO-SHILPE-0002 Čelisti
3 2 PRO-SHILPE-0001 Bočnice
2 4 ŠROUB M5 x 30 ČSN 02 1143 - imbus
1 1 SMC-CQS-B20-50 DCM Pneumatický motor
POZICE KS ČÍSLO SOUČÁSTI POPIS
Kusovník
SSPŠ PRAHA 5
PŘESLOVA 25
4.S
1:1 (2:1)
ÚHLOVÉ CHAPADLO
PRO-SHILPE-0000
ROZEVŘENÍ
SEVŘENÍ
Model 1:1


Model hotové součásti M 1:1
2x Ø5 N7 - srazit vnější hrany díry na 1x45°
A-A
Ø2 N7
R10
Nekótované rozměry jsou symetricky stejné
Díry pro dorazy opracovat na Ra 0,8
Počet kusů: 2
Materiál: Al-Cu4-Mg1 (SUPERDURAL)
SSPŠ PRAHA 5
PRESLOVA 25
4.S
2:1
BOČNICE
PRO-SHILPE-0001


Nekótované rozměry jsou symetricky stejné
Díru G7 brousit na Rm 0,4
Počet kusů: 2
Materiál: EN 10025 - ocel 11 500
SSPŠ PRAHA 5
PRESLOVA 25
4.S
2:1 (5:1)
PRO-SHILPE-0002
ČELIST
Z (5 : 1)
kóty jsou od osy na druhé straně symetrické
A-A
Model hotové součásti M 1:1





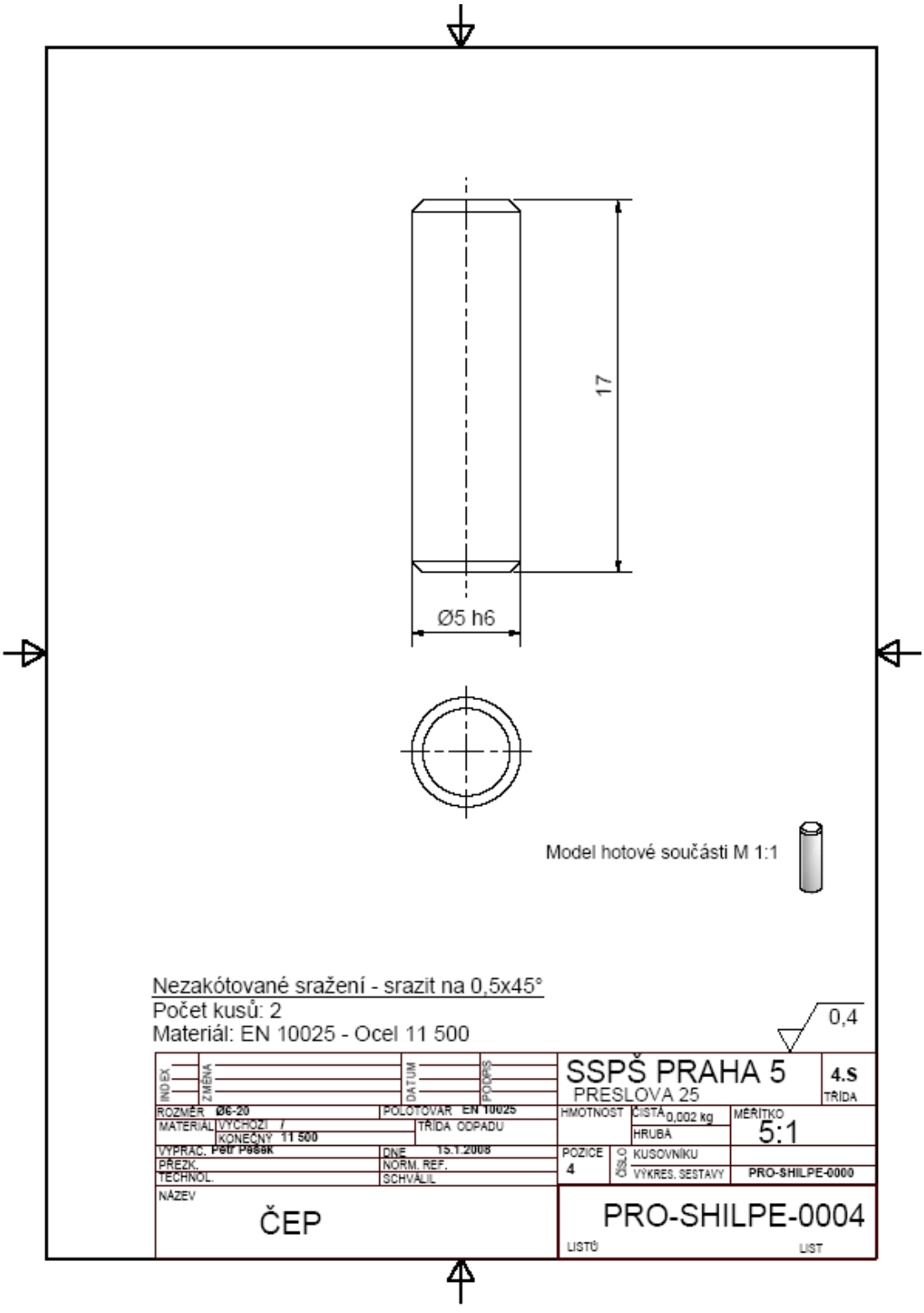


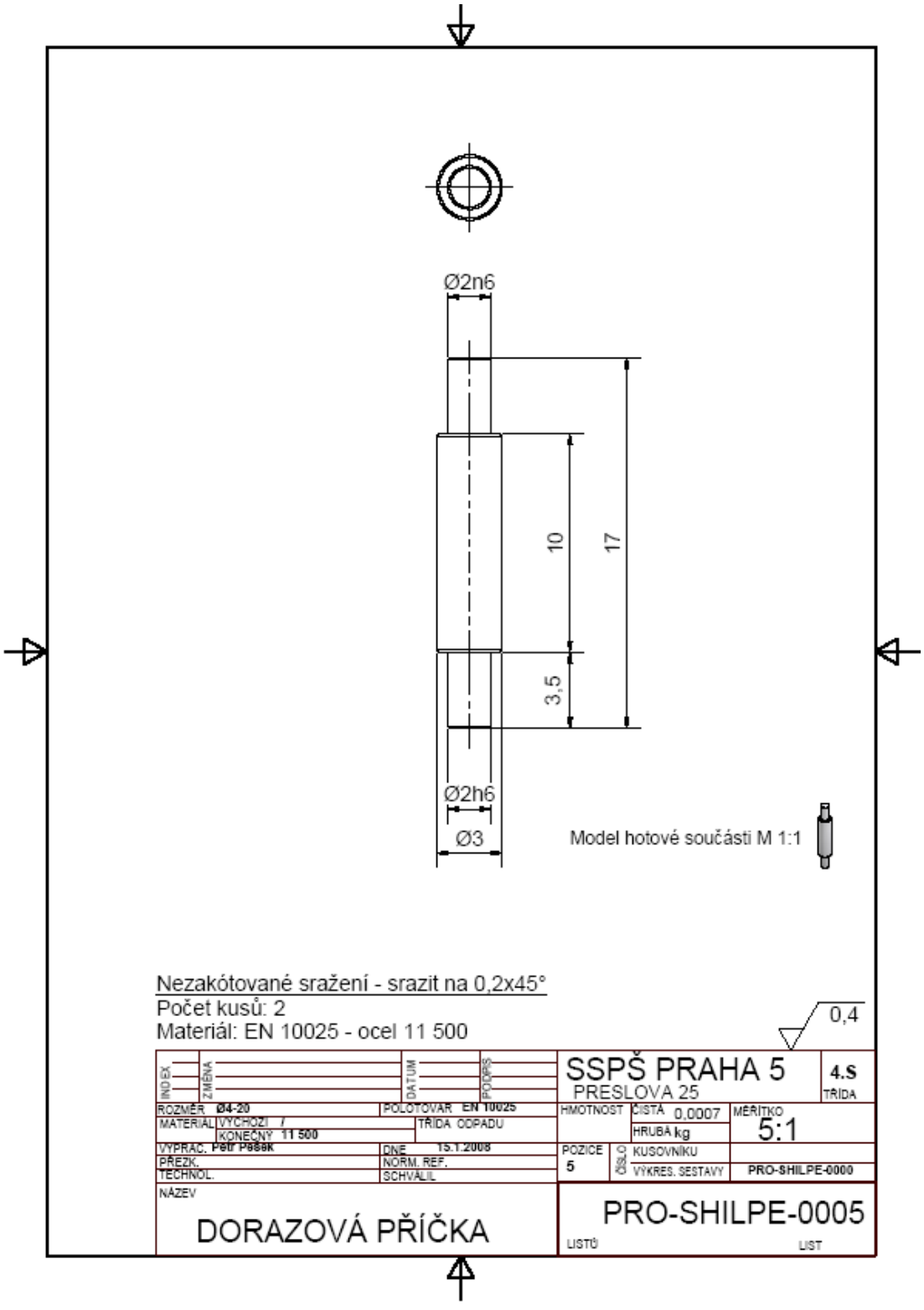
Model hotové součásti M 1:1

Nezakótované sražení - srazit na 0,5x45°
Počet kusů: 1
Materiál: Al-Cu4-Mg1 (SUPERDURAL)

1,6 (0,2 0,8)

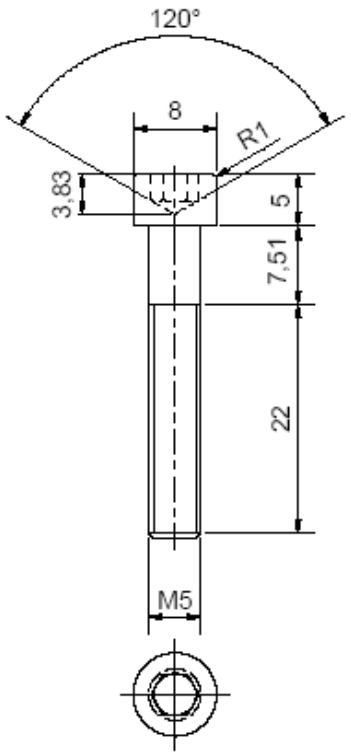
INDEX	ZMĚNA	DATUM	PODPIS	SSPŠ PRAHA 5 PRESLOVA 25	4.S TŘÍDA
ROZMĚR Ø25-100		POLOTOVAR EN AW 5754		HMOTNOST	ČISTÁ 0,007 kg / HRUBÁ	MĚŘÍTKO 2:1
MATERIÁL	VÝCHOZÍ / KONEČNÝ Al-Cu4-Mg1	TŘÍDA ODPADU				
VYPRAC. Petr Pešek		DNE 15.1.2008		POZICE 3	ČÍSLO KUSOVNÍKU	
PŘEZK.		NORM. REF.			VÝKRES. SESTAVY	PRO-SHILPE-0000
TECHNOL.		SCHVÁLIL				
NÁZEV KÁMEN				PRO-SHILPE-0003		
				LISTŮ	LIST	

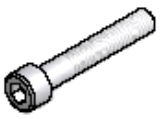

17
Ø5 h6
Model hotové součásti M 1:1
Nezakótované sražení - srazit na 0,5x45°
Počet kusů: 2
Materiál: EN 10025 - Ocel 11 500
0,4
IND.EX
ZMĚNA
DATUM
PODPIS
SSPŠ PRAHA 5
PRESLOVA 25
4.S
TŘÍDA
ROZMĚR Ø6-20
POLOTOVAR EN 10025
MATERIÁL VÝCHOZÍ /
KONEČNÝ 11 500
TŘÍDA ODPADU
HMOTNOST
ČISTÁ 0,002 kg
HRUBÁ
MĚŘÍTKO
5:1
VYPRAC. Petr Pešek
DNE 15.1.2008
PŘEZK.
NORM. REF.
TECHNOL.
SCHVÁLIL
POZICE
4
ČÍSLO
KUSOVNÍKU
VÝKRES SESTAVY
PRO-SHILPE-0000
NÁZEV
ČEP
PRO-SHILPE-0004
LISTŮ
LIST


Ø2n6
10
17
3,5
Ø2h6
Ø3
Model hotové součásti M 1:1
Nezakótované sražení - srazit na 0,2x45°
Počet kusů: 2
Materiál: EN 10025 - ocel 11 500
0,4
INDEX
ZMĚNA
DATUM
POPIS
SSPŠ PRAHA 5
PRESLOVA 25
4.S
TŘÍDA
ROZMĚR Ø4-20
POLOTOVAR EN 10025
MATERIÁL VÝCHOZÍ /
KONEČNÝ 11 500
TŘÍDA ODPADU
HMOTNOST
ČISTÁ 0,0007
HRUBÁ kg
MĚŘÍTKO
5:1
VYPRAC. Petr Pešek
DNE 15.1.2008
PŘEZK.
NORM. REF.
TECHNOL.
SCHVÁLIL
POZICE
5
ČÍSLO
KUSOVNÍKU
VÝKRES. SESTAVY
PRO-SHILPE-0000
NÁZEV
DORAZOVÁ PŘÍČKA
PRO-SHILPE-0005
LISTŮ
LIST

Upravení šroubu - skrácení hlavy o 1 mm (odsoustružení od spodní části)

ostatní rozměry podle normy - ŠROUB M5 X 30 ČSN 02 1143

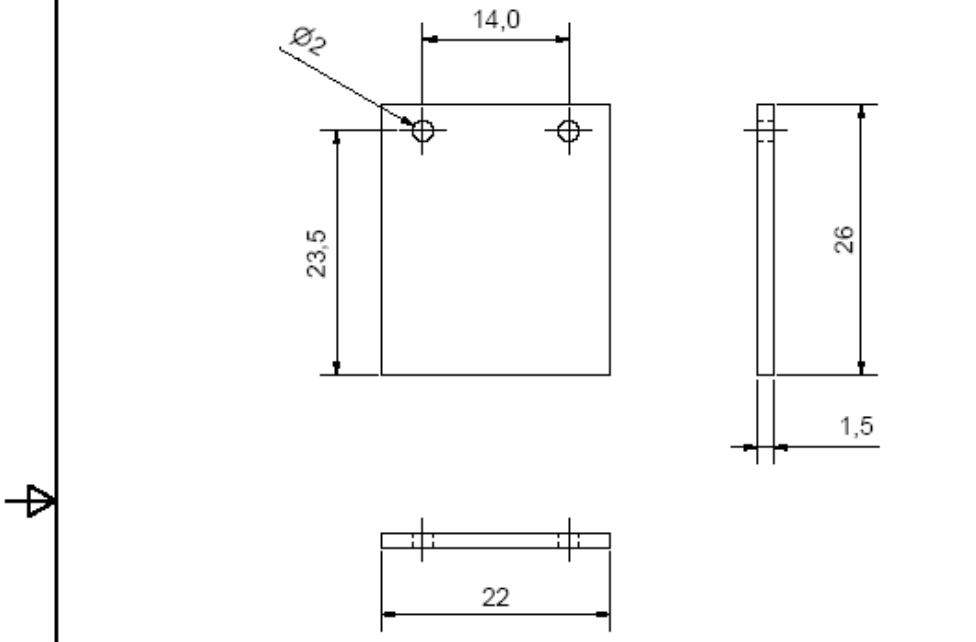


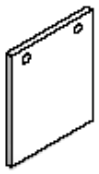


Model součásti M1:1

Počet kusů: 4

INDEX	ZMĚNA	DATUM	PODPIS	SSPŠ PRAHA 5 PRESLOVA 25			4.S TŘÍDA
ROZMĚR		POLOTOVAR ČSN 02 1143		HMOTNOST	ČISTÁ	MĚŘÍTKO	
MATERIÁL	VÝCHOZÍ / KONEČNÝ	TŘÍDA ODPADU /			HRUBÁ	2:1	
VYPRAC. Petr Pešek		DNE 15.1.2008		POZICE 6	ČÍSLO KUSOVNÍKU		
PŘEZK.		NORM. REF.			VÝKRES. SESTAVY	PRO-SHILPE-0000	
TECHNOL.		SCHVÁLIL					
NÁZEV ŠROUB M5 X 30				PRO-SHILPE-0006			
				LISTŮ		LIST	





Počet kusů: 2

Model součásti M 1:1

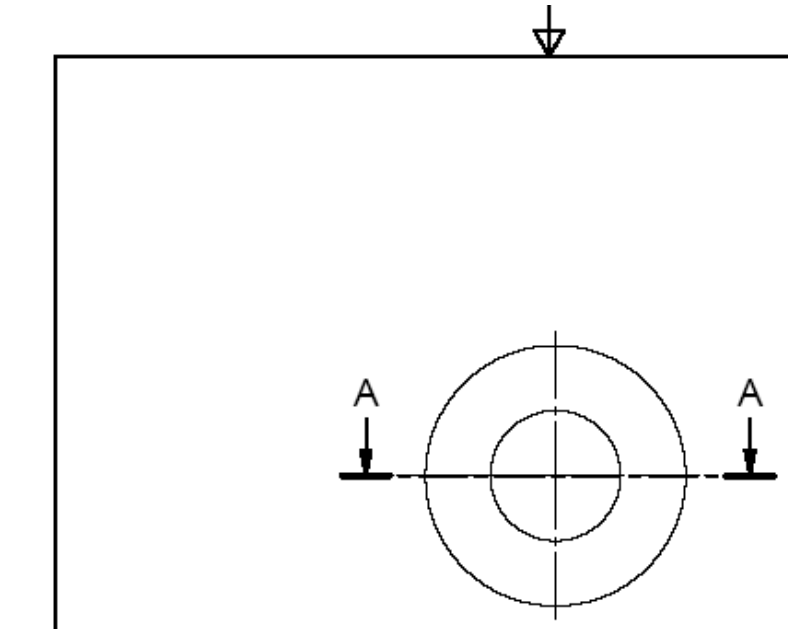
INDEX	ZMĚNA	DATUM	PODPIS	SSPŠ PRAHA 5 PRESLOVA 25	4.S TŘÍDA
ROZMĚR 1,5x23-25		POLOTOVAR		HMOTNOST	ČISTÁ HRUBÁ	MĚŘÍTKO 2:1
MATERIÁL	VÝCHOZÍ TEFLON KONEČNÝ TEFLON	TŘÍDA ODPADU				
VYPRAC. Petr Pešek		DNE 15.1.2008		POZICE 7	ČÍSLO KUSOVNÍKU VÝKRES. SESTAVY	PRO-SHILPE-0000
PŘEZK.		NORM. REF.				
TECHNOL.		SCHVÁLIL				

NÁZEV

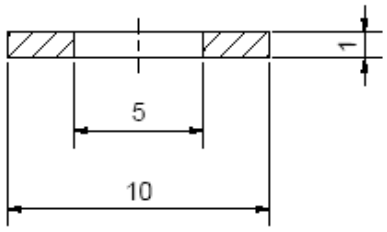
TEFLONOVÉ VEDENÍ

PRO-SHILPE-0007

LISTŮ LIST



A-A

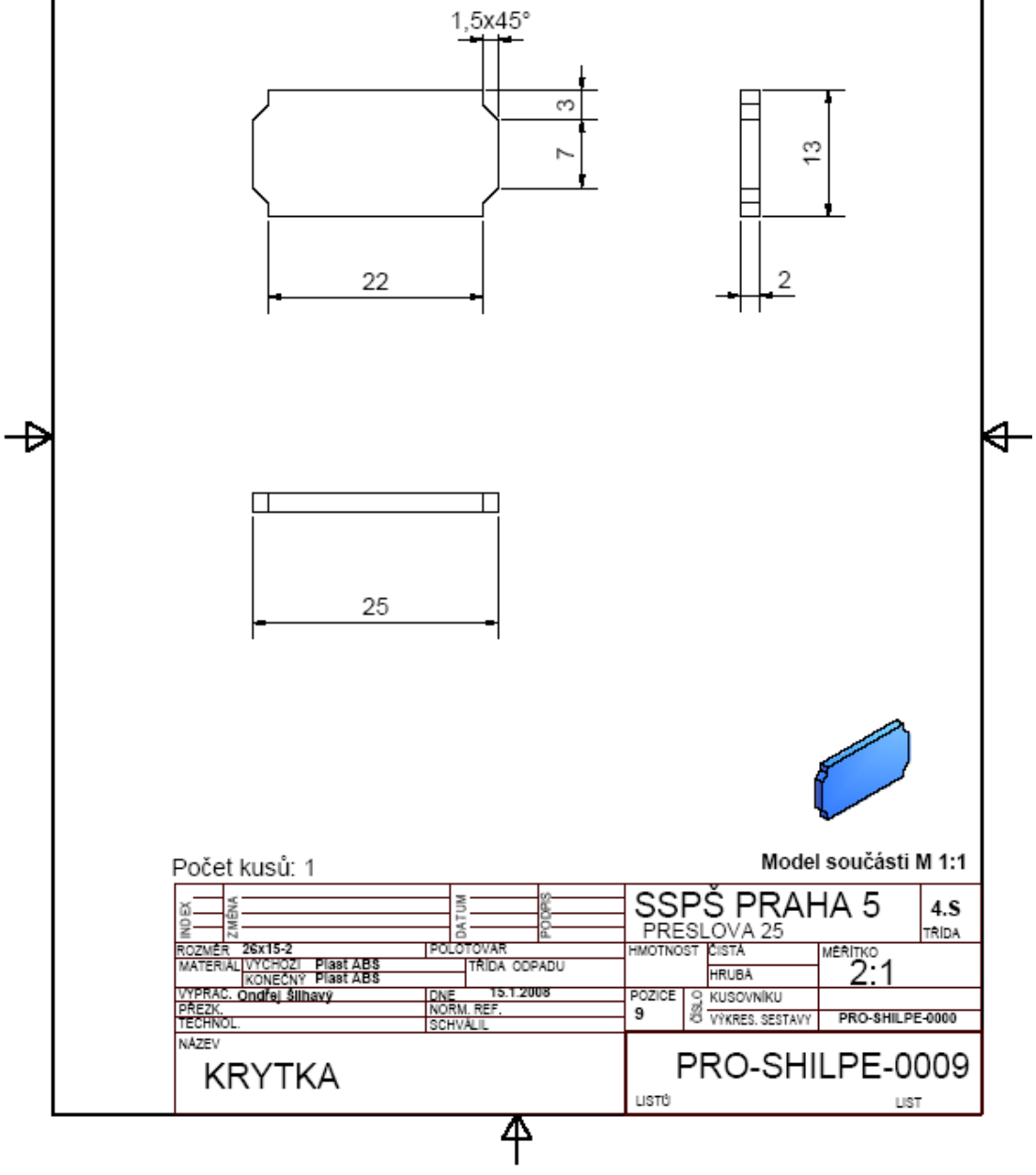




Počet kusů: 4

Model součásti M 1:1

IND.EX / ZMĚNA / DATUM / PODPIS		SSPŠ PRAHA 5 PRESLOVA 25	4.S TŘÍDA
ROZMĚR 15x15-1	POLOTOVAR	HMOTNOST	ČISTÁ / HRUBÁ
MATERIÁL VÝCHOZÍ TEFLON KONEČNÝ TEFLON	TŘÍDA ODPADU		MĚŘÍTKO 5:1
VYPRAC. Petr Pešek	DNE 15.1.2008	POZICE 8	OSS.Č. KUSOVNÍKU
PŘEZK.	NORM. REF.		VÝKRES. SESTAVY PRO-SHILPE-0000
TECHNOL.	SCHVÁLIL		
NÁZEV TEFLONOVÝ KROUŽEK		PRO-SHILPE-0008	
		LISTŮ	LIST



Počet kusů: 1

Model součásti M 1:1

INDEX / ZMĚNA / DATUM / PODPIS			SSPŠ PRAHA 5 PRESLOVA 25		4.S TŘÍDA
ROZMĚR 26x15-2	POLOTOVAR		HMOTNOST	ČISTÁ	MĚŘÍTKO 2:1
MATERIÁL VÝCHOZÍ Plast ABS KONEČNÝ Plast ABS	TŘÍDA ODPADU			HRUBÁ	
VYPRAC. Ondřej Šilhavý	DNE 15.1.2008		POZICE 9	ČÍSLO KUSOVNÍKU	
PŘEZK.	NORM. REF.			VÝKRES. SESTAVY	PRO-SHILPE-0000
TECHNOL.	SCHVÁLIL				
NÁZEV KRYTKA			PRO-SHILPE-0009		
			LISTŮ		LIST

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

HISTORIE PRAHY - PIVOVARNICTVÍ

Jan Pyšvejc, Jakub Malý

SSPŠ (Smíchovská střední průmyslová škola)
Preslova 25, Praha 5 - Smíchov

Zadání:

Vytvořte multimediální klip na dané téma.
Zpracujte jej do formátu video DVD.

Téma:

Historie Prahy – Pivovarnictví

Obsah:

Vytvořte scénář (popis natáčení) videoklipu.
Natočte digitální videokamerou záběry dle vašeho scénáře.
Natočené záběry převeďte do počítače pro zpracování.
V programu ULEAD MEDIA STUDIO PRO sestříhejte vaše záběry.
Doplňte videoklip úvodními a závěrečnými titulky
Dle potřeby doplňte zvuk do pozadí (komentář, hudba)
Z hotového videoklipu vytvořte video DVD

1 Obsah

1 Obsah

2 Úvod - Historie Prahy – Pivovarnictví
1.2 Projekt obsahuje:

3 Scénář
1.3 Historie Prahy – Pivovarnictví
3.1.1 Ext. - Město - Den - Úvod
3.1.2 Ext. - Město - Noc - Úvod
3.1.3 Ext. - Pivovar - Den
3.1.4 Int. - Pivovar - Den
3.1.5 Int. - Pivovar - Den
3.1.6 Int. - Pivovar - Den
3.1.7 Int. - Pivovar - Den
3.1.8 Int. - Pivovar - Noc - Závěr
3.1.9 Int. - Pivovar - Noc - Závěr

4 Obrázky
1.4 Založení nového projektu.
2.4 Editace a střih videa.
3.4 Práce s videem, vkládání přechodů (Transition) snímků.
4.4 Konvertování videa do finální podoby.

5 Závěr

Historie Prahy - pivovarnictví

6 Přílohy
1.6 Prague history and brewery U Fleků
2.6 Prezentace

2 Úvod - Historie Prahy – Pivovarnictví

Tento projekt referuje historii Prahy se zaměřením na pražské pivovarnictví, konkrétněji na pivovar U Fleků. Všechny informace a rozhovory, které lze shlédnout v tomto projektu, jsme usilovně získávali, shromažďovali a následně upravovali do finální podoby. Hlavním cílem našeho projektu je tvorba videa pro prezentaci.

1.2 Projekt obsahuje:

1) **Video**
   a. Intro
   b. Praha
   c. Cesta ke Flekům
   d. Pivovar U Fleků
   e. Závěrečné titulky
2) **Dokumentace** – popis práce na projektu
3) **Prezentace** – základní údaje projektu

3 Scénář

1.3 Historie Prahy – Pivovarnictví

Dokument spojený s prohlídkou nejstaršího pražského pivovaru „U Fleků“

3.1.1 Ext. - Město - Den - Úvod

Prohlídka Prahou. Natočení a zachycení pražských památek, turistického a všedního ruchu.

- Kamera přibližuje pohled na centrum Prahy
- Záběry projíždějících aut a kolemjdoucích lidí

3.1.2 Ext. - Město - Noc - Úvod

Procházka a popis cesty k pivovaru.

- Záběry a pohled z první osoby
- Možný záběr na mapu města s vyznačenou trasou

3.1.3 Ext. - Pivovar - Den

Dokumentace exteriéru pivovaru. Zachycení vstupních vrat apod.

- Kamera zachycuje pivovar ze všech stran

3.1.4 Int. - Pivovar - Den

Vstup do pivovaru. Dokumentace interiéru.

- Kamera pomalu najíždí hlavní branou

3.1.5 Int. - Pivovar - Den

Prohlídka venkovní zahrádky a ostatních částí interiéru pivovaru.

- Záběry všech venkovních částí
- Možné záběry výzdoby

3.1.6 Int. - Pivovar - Den

Rozhovory s vybranými osobami (vedení). Možný náhled do kuchyně. Vyzvídání specialit apod.

Jan Pyšvejc, Jakub Malý

3.1.7 Int. - Pivovar - Den

Prohlídka sklepení pivovaru. Zachycení varných kotlů a přípravy piva. Prohlídka skladiště.

- Kamera se pomalu přibližuje k varným kotlům
- Detailní záběry na sudy s pivem a ostatní dekoraci

3.1.8 Int. - Pivovar - Noc - Závěr

Zachycení celkového ruchu v pivovaru, záběry obsluhy, specialit apod.

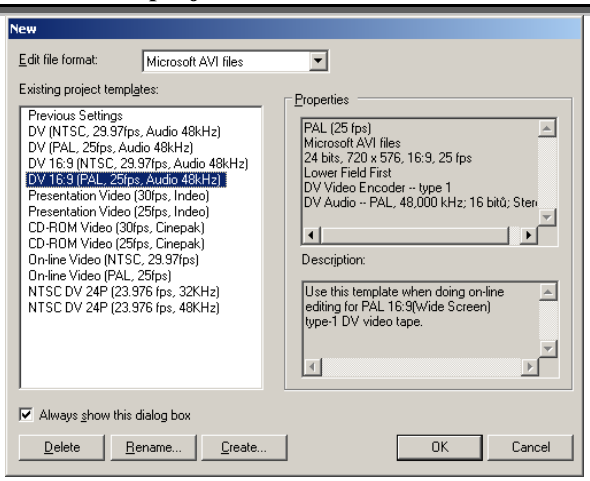
- Možné popovídání s hosty pivovaru
- záběry projíždějících aut a kolemjdoucích lidí

3.1.9 Int. - Pivovar - Noc - Závěr

Ochutnávka piva.

4 Obrázky

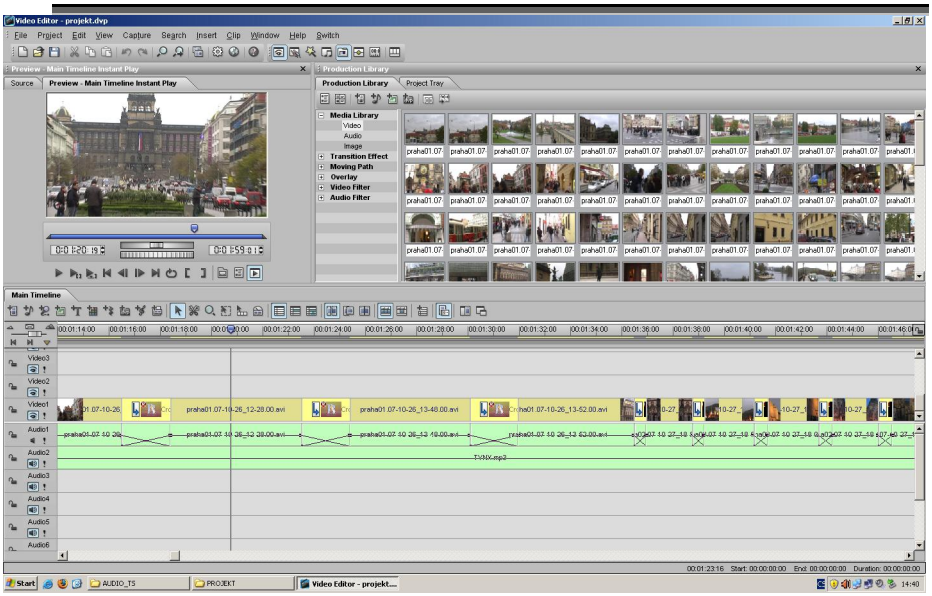
1.4 Založení nového projektu.



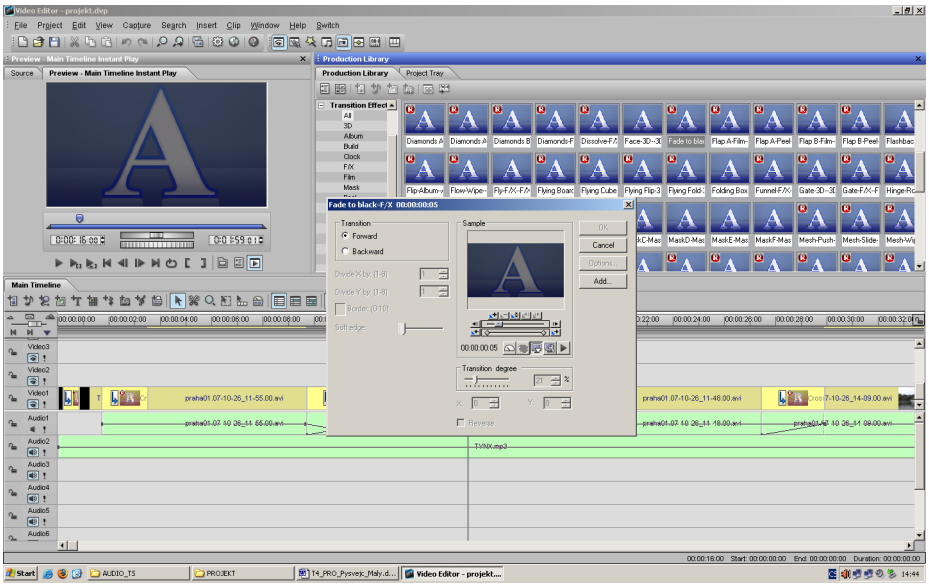
Jan Pyšvejc, Jakub Malý

Historie Prahy - pivovarnictví

2.4 Editace a střih videa.

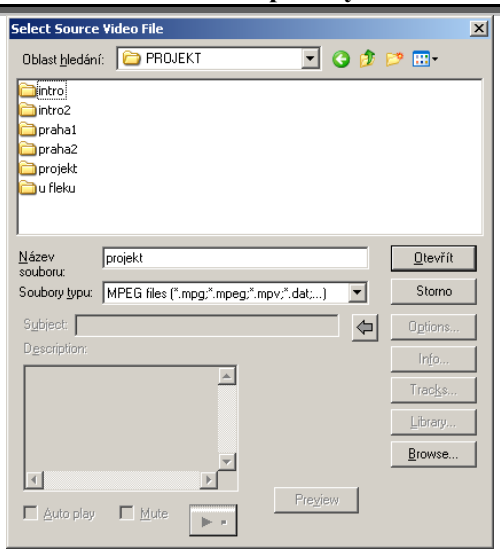


3.4 Práce s videem, vkládání přechodů (Transition) snímků.



Jan Pyšvejc, Jakub Malý

4.4 Konvertování videa do finální podoby.



5 Závěr

Projekt jsme úspěšně dokončili. Natáčení a střih videa nám dal mnoho nových zkušeností. Natáčení nás bavilo a bez větších problémů jsme projekt úspěšně dokončili.

6 Přílohy

1.6 Prague history and brewery U Fleků

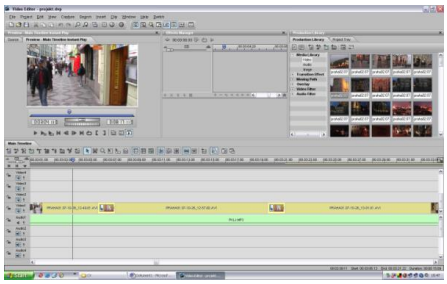
This project refers to history of Prague. We were also engaged in brewery U Fleků. We must have carried out a lot of questions in this project. We must have recorded a lot of information about Prague and brewery U Fleků. These records served to create documentation and presentation about these topics. It was difficult to collect all these information.
The main goal was making a video for presentation about Prague and brewery U Fleků.
This project consists of:

1. Video, which consists of:
   Introduction
   Part about Prague
   Part about brewery U Fleků
   Result
2. Documentation - it is manual (scenario), which characterises our work in this project
3. Presentation - it shows our work other people

Jan Pyšvejc, Jakub Malý

Historie Prahy - pivovarnictví





We have made records in Prague and brewery U Fleků. For it we must have walked in Prague and recorded our historical sights and modern buildings, we must have reserved some time that we were spending in brewery. It wasn't easy to find free time in which we were allowed to make some recordings in the brewery. We cut records that become parts of our final video. We got together this parts and our project was on the world.
Finally it we have made some documentation like this and we have created some presentation for showing project.
This project people can use like a presentation for showing or some advertising.

Jan Pyšvejc, Jakub Malý

Historie Prahy - pivovarnictví

2.6 Prezentace





Jan Pyšvejc, Jakub Malý

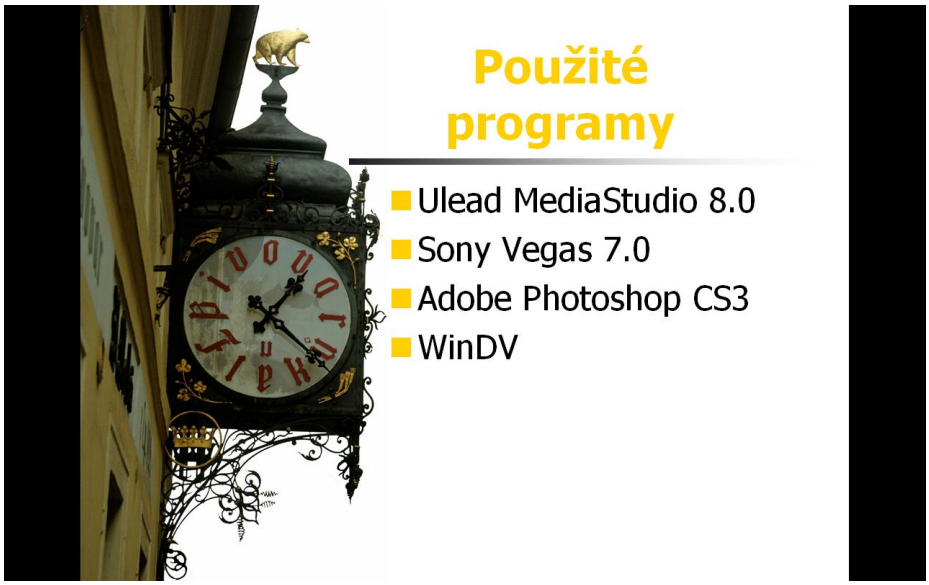
Historie Prahy - pivovarnictví

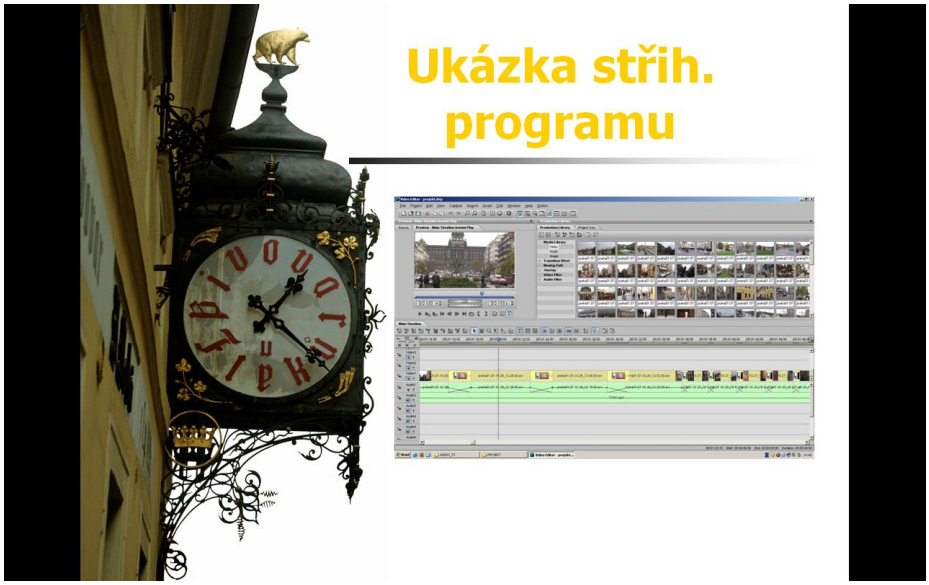




Jan Pyšvejc, Jakub Malý

Historie Prahy - pivovarnictví





Jan Pyšvejc, Jakub Malý

Historie Prahy - pivovarnictví



Jan Pyšvejc, Jakub Malý

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

HOMO MUSICUS

Pavel Štěpán,Ondřej Ostarek, Marek Vošahlík

SSPŠ (Smíchovská střední průmyslová škola)
Preslova 25, Praha 5 - Smíchov

1 Obsah

1 Obsah
2 Úvod
3 Vytyčené cíle
3.1. Cíl 1
3.2. Cíl 2
4 „Pomocníci“
5 Vlastní práce
5.1. Střih
5.2. Vložení nového klipu na časovou osu
5.3. Přechody hudby a videa
5.4. Změna kontrastu a jasu
5.5. Synchronizace titulků s obrazem
5.6. Dodání vnějších zdrojů hudby
6 Místa natáčení

2 Úvod

Film se zabývá zvláště undergroundovou a amatérskou rockovou scénou a mladými lidmi kteří ji tvoří.. Klíčový prvek je studio, skrze které naši reportéři provázejí diváka životem kapely – zkouškami a koncerty. Další část filmu je věnována rozhovorům a názorům veřejnosti.
Tento projekt jsme si z volili, protože se velmi zajímáme o amatérskou hudební scénu.



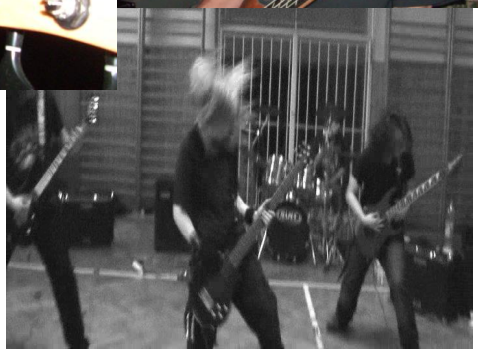
3 Vytyčené cíle

3.1. Cíl 1

Snažili jsme se zachytit rozpoložení a názory mladých hudebníků během zkoušek, před koncerty a po koncertech..







3.2. Cíl 2

Ptali jsme se veřejnosti, jakou muziku poslouchá a kde..





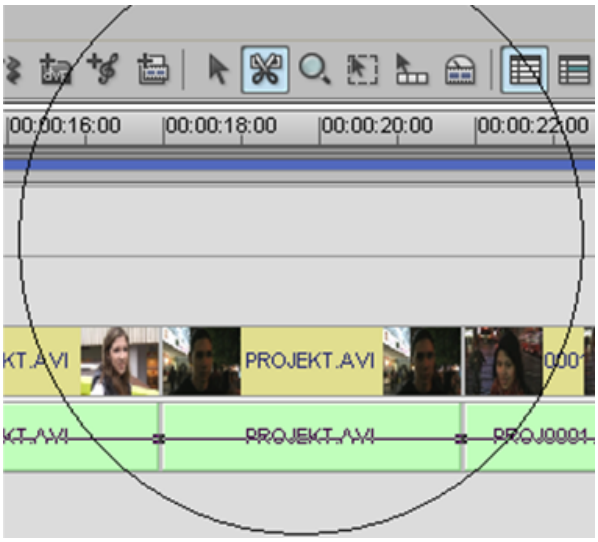


4 „Pomocníci“

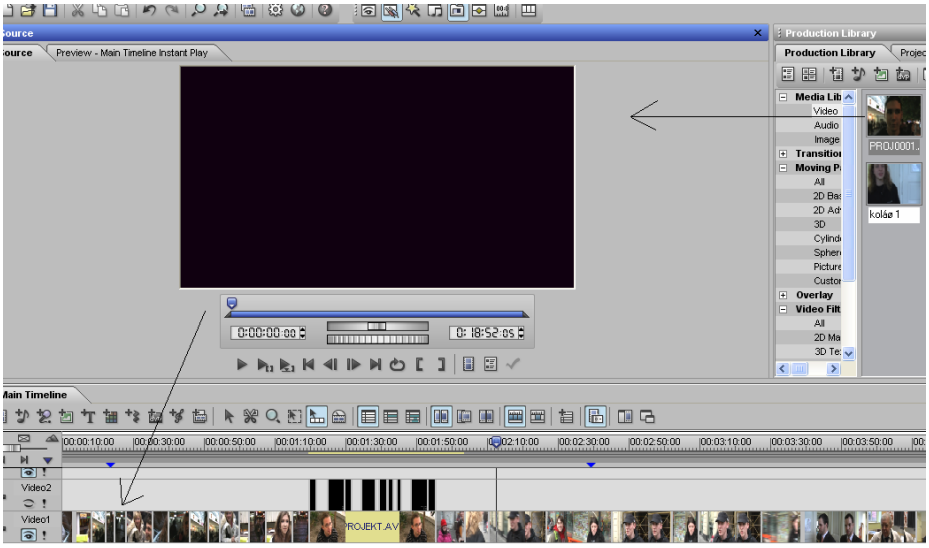
- Kamera Panasonic GS500
- PC Intel Core Duo 2,4 ghz
- Ulead Media Studio 8

5 Vlastní práce

5.1. Střih



5.2. Vložení nového klipu na časovou osu



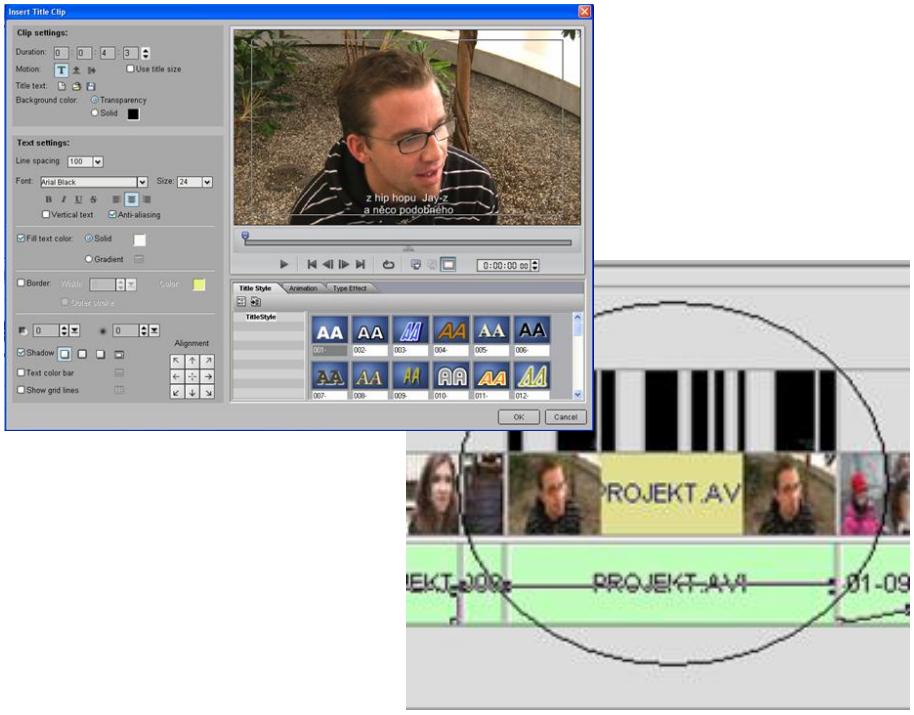
5.3. Přechody hudby a videa



5.4. Změna kontrastu a jasu



5.5. Synchronizace titulků s obrazem



5.6. Dodání vnějších zdrojů hudby



6 Místa natáčení

- Ulice Prahy
- Zkušebny
- Tělocvična Obchodní akademie
- SSPŠ





Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

PŘEVOD DESKOVÉ HRY "LABYRINT" DO ELEKTRONICKÉ PODOBY

Ondřej Sýkora, Tomáš Richter

SSPŠ (Smíchovská střední průmyslová škola)
Preslova 25, Praha 5 - Smíchov

OBSAH:

Obsah:
Seznam obrázků
Seznam Příloh
1. Úvod
2. Teorie
   2.1 Počítačová hra
   2.2 Hry hrané na jednom počítači
   2.3 Programování a algoritmy
   2.4 Programovací jazyk C#
   2.5 Umělá inteligence
   2.6 Path-finding
   2.7 2D počítačová grafika
   2.8 Rendering
3. Popis
   3.1 Pravidla
   3.2 Spuštění
   3.3 Nová hra
   3.4 Hrací plocha
   3.5 Pravidla
   3.6 Autoři
4. Závěr … 13

SEZNAM OBRÁZKŮ

Obrázek 1 - Hlavní menu
Obrázek 2 - Nová hra
Obrázek 3 - Hrací plocha
Obrázek 4 - Detail hrací plochy
Obrázek 5 - Pravidla
Obrázek 6 - Autoři
Obrázek 7 - Český leták
Obrázek 8 - Anglický leták
Obrázek 9 - Hlavní menu
Obrázek 10 - Nová hra
Obrázek 11 - Hrací plocha
Obrázek 12 - Pravidla
Obrázek 13 - Autoři
Obrázek 14 - Úvod
Obrázek 15 – Labyrint
Obrázek 16 - Schémata
Obrázek 17 - O nás
Obrázek 18 - Ondřej Sýkora, Ondřej Sýkora, Tomáš Richter
Obrázek 19 - Ondřej Sýkora
Obrázek 20 - O projektu
Obrázek 21 – Úvod
Obrázek 22 - Seznámení s tématem
Obrázek 23 - Hra Labyrint
Obrázek 24 - Charakteristické rysy
Obrázek 25 – Uživatelské rozhraní
Obrázek 26 - Hrací plocha
Obrázek 27 - Technologické okénko
Obrázek 28 - O projektu
Obrázek 29 - Doprovodné elementy
Obrázek 30 - Autoři
Obrázek 31 – Booklet
Obrázek 32 - Polep CD

SEZNAM PŘÍLOH

Příloha A Leták .....
A1 Česky .....
A2 Anglicky .....
Příloha B Labyrint .....
B1 Hlavní menu .....
B2 Nová hra .....
B3 Hrací plocha .....
B4 Pravidla .....
B5 Autoři .....
Příloha C Zdrojový kód .....
C1 AI .....
C2 Figurka .....
C3 Graficky objekt .....
C4 Hra .....
C5 Hrac .....
C6 HraciForm .....
C7 HraciForm-design .....
C8 HraciForm-udalosti .....
C9 HraciKarta .....
C10 Kamen .....
C11 Menu .....
C12 Rozliseni .....
C13 Startup .....
C14 Zdroj .....
Příloha D Webové stránky .....
D1 Úvod .....
D2 Labyrint .....
D3 O nás .....
D4 O projektu .....
Příloha E Prezentace .....
E1 Úvod .....
E2 Seznámení s tématem .....
E3 Hra Labyrint .....
E4 Charakteristické rysy .....
E5 Uživatelské rozhraní .....
E6 Hrací plocha .....
E7 Technologické okénko .....
E8 O projektu .....
E9 Doprovodné elementy .....
E10 Autoři .....
Příloha F Booklet & Polep CD .....
F1 Booklet .....
F2 Polep CD .....

1. ÚVOD

Rádi bychom Vás úvodem seznámili s naším maturitním projektem, počítačovou hrou Labyrint. Jedná se o adaptaci rodinné deskové hry německé společnosti Ravensburger, kterou jsme vyvíjeli pod taktovkou paní profesorky Ing. Olgy Kaiferové, za níž jsme na začátku školního roku s nápadem realizovat tento projekt přišli, přičemž bylo nutné vypracovat návrh, jak by měla hra nejen vypadat, ale především fungovat.

Jakmile jsme dostali zelenou, začali jsme na projektu ihned pracovat. Zpočátku jsme vytvořili grafický interface, abychom mohli hru následně programovat. Vzhledem k tomu, že naše nároky na hru byly více než vysoké, většinu volného času jsme strávili právě u vývoje, neboť z každé maličkosti se nakonec vyklubal nezměrný problém, díky čemuž jsme museli v návrhu udělat nemalé škrty.

Jelikož hra samotná je založena na rádoby interaktivním prostředí, bylo nezbytné naprogramovat vlastní vykreslovací technologii, která jde ruku v ruce například s plynulými animacemi. Než jsme však dosáhli touženého výsledku, první dva měsíce jsme usilovně přemýšleli a vytvářeli zkušební verze programu, načež jsme teprve v listopadu začali pracovat na samotných funkcích hry. Těmi jsou například umělá inteligence, chytré vyhledávání cesty nebo hra jednoho hráče, což jsou natolik komplexní záležitosti, že jsme byli často myšlenkami namísto ve škole u počítače.

V průběhu vývoje jsme rovněž museli pracovat na podpůrných elementech, které samotný maturitní projekt doprovází. Jmenovat můžeme kupříkladu webové stránky, leták či prezentaci. Leták byl navržen především tak, aby oslovil a donutil osloveného pátrat po dalších informacích na internetových stránkách naší společnosti, které rovněž vsází na jednoduchost a přehlednost.

2. TEORIE

2.1 POČÍTAČOVÁ HRA

Počítačová hra je provozovaná prostřednictvím počítače, dnes obvykle na PC či Macintosh. Jedná se o druh softwaru vydávaného na médiu (CD nebo DVD), které se vkládá do příslušné mechaniky. V současnosti je rovněž populární i distribuce přes internet. Počítačová hra slouží k zábavě, uvolnění a relaxaci. Moderní počítačové hry mívají vysoké hardwarové požadavky, přičemž vyžadují zejména ovládací zařízení, grafickou kartu, zvukovou kartu a jiné hardwarové komponenty pro lepší zážitek ze hry. Vývoj počítačové hry bývá velmi zdlouhavý a nezřídka se stává, že trvá i několik let. Hry většinou svými moderními technologiemi a herním enginem předvádí možnosti současného hardwarového pokroku.

Každá počítačová hra se skládá z virtuálního světa nebo prostředí, do kterého může hráč pomocí komponent připojených k počítači, jako je například myš, klávesnice nebo joypad, vstoupit a nějakým způsobem ovlivňovat dění virtuálního prostředí. To znamená, že k interakci dochází na základě zásahu hráče do spuštěného programu hry, který byl pro takový zásah vyvíjen. Obvykle se jedná o určitý úkol, který musí hráč splnit v nějakém časovém limitu, porazit počítačový program v simulaci nějakého boje, vyhrát závodní simulaci nebo dostát úplně jiného cíle odvislého od námětu a žánru i záměru vývojářů.

Herní žánry se dnes nejobecněji kategorizují do žánrových skupin, přičemž u některých titulů dochází k prolínání žánrů nebo vytváření specifických podžánrů. Žánry se pak liší způsobem zásahu hráče do hry a nabízenými prostředky programu k interakci.

2.2 HRY HRANÉ NA JEDNOM POČÍTAČI

Na jednom počítači může hrát až několik hráčů najednou, a zasahovat tak do dění na obrazovce, přičemž počet hráčů, kteří zasahují do hry, je obvykle limitován samotnou hrou nebo množstvím ovládacích zařízení, která jsou k dispozici. Tento způsob hraní však dnes nebývá u komerčních titulů k vidění, a to zejména pro značné nepohodlí několika hráčů tísnících se u jedné obrazovky. Zůstává tak doménou hlavně starších arkád.

Taktéž existuje mód Hotseat. Ten je stále využíván zejména u her hraných po kolech, především pak u her strategických nebo logických. Hráči se střídají pouze na jednom počítači, kdy se délka hraní může neúměrně protahovat, což bývá nejčastěji ošetřeno časovým limitem. Je zároveň na nich, zda sledují tahy protivníka či ne.

Posledním způsobem, jak hrát hru ve více hráčích na jednom počítači, je Split Screen. Ten je k vidění jak u starších her, které ještě nepodporovaly hraní po síti, tak u současných konzolových her. Spočívá v rozdělení obrazovky do několika částí podle počtu hráčů. Každý hráč pak vidí na obrazovce svojí část hry. Tento mód dokáže být až překvapivě zábavný, přesto je však často nepřehledný.

2.3 PROGRAMOVÁNÍ A ALGORITMY

Programování je činnost, která zahrnuje tvorbu algoritmu a programu. Program je zápis algoritmu ve zvoleném programovacím jazyce. Algoritmus je přesný návod či postup, kterým lze vyřešit daný typ úlohy. Pojem algoritmu se nejčastěji objevuje při programování, kdy se jím myslí teoretický princip řešení problému (oproti přesnému zápisu v konkrétním programovacím jazyce). Obecně se ale algoritmus může objevit v jakémkoli jiném vědeckém odvětví. Jako jistý druh algoritmu se může chápat i například kuchyňský recept. V užším smyslu se slovem algoritmus rozumí pouze takové postupy, které splňují některé silnější požadavky.

Je třeba poznamenat, že abstraktní kritérium konečnosti je pro praktické použití ještě příliš slabé. V praxi je třeba zajistit, aby algoritmus skončil „v rozumném“ čase. Za rozumný čas lze v praxi považovat takový čas, který nám umožní smysluplně využít výsledek.

Např. existuje jednoduchý algoritmus, který dokáže určit, zda v dané šachové pozici může hráč na tahu vynutit vítězství a zároveň dokáže určit nejlepší možný tah. Tento algoritmus se však nedá použít, protože by na svou činnost potřeboval ohromné množství času, jakkoli je toto množství konečné. Mimoto by takový algoritmus spotřeboval ohromné množství paměti, což je další praktický zřetel, který se uplatňuje při volbě algoritmu. I když průměrná počítačová paměť stále narůstá, pro některé algoritmy jí nebude nikdy dost.

Pro vyčíslení výpočetní složitosti algoritmů v závislosti na velikosti vstupních dat se používá asymptotický zápis závislosti výpočetního času na rozsahu úlohy (typicky na počtu vstupních údajů). Například *O(log N)* znamená, že počet kroků algoritmu závisí logaritmicky na velikosti vstupních dat. Pokud u takového algoritmu zdvojnásobíme rozsah vstupních údajů, doba výpočtu se zvýší o jednu jednotku času, pokud bude vstupních dat čtyřikrát více, doba výpočtu se prodlouží o dvě jednotky času, a tak dále. To je třeba případ nalezení jednoho prvku o určité hodnotě v seznamu prvků seřazeném podle hodnoty (např. nalezení jména v telefonním seznamu).

2.4 PROGRAMOVACÍ JAZYK C#

Jde o vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework, později schválený standardizačními komisemi ECMA a ISO. Microsoft založil C# na jazycích C++ a Java. Je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi. C# se využívá hlavně k tvorbě

databázových programů, webových aplikací, webových služeb, formulářových aplikací ve Windows a jiných.

C# je jednoduchý, moderní a mnohoúčelový programovací jazyk. Jeho implementace poskytuje podporu pro principy softwarového inženýrství, jako jsou hlídání hranic polí, detekce použití neinicializovaných proměnných a automatický garbage collector. Důležité jsou také jeho vlastnosti, tedy robustnost, trvanlivost a programátorská produktivita. Jazyk je vhodný pro vývoj softwarových komponent distribuovaných v různých prostředích. Přenositelnost zdrojového kódu je velmi důležitá, obzvláště pro ty programátory, kteří jsou obeznámeni s C a C++. Mezinárodní podpora je rovněž velice důležitá.

2.5 UMĚLÁ INTELIGENCE

Umělá inteligence (UI) je obor informatiky, zabývající se tvorbou strojů vykazujících známky inteligentního chování. Definice pojmu „inteligentní chování“ je stále předmětem diskuse, nejčastěji se jako etalon inteligence užívá lidský rozum. Umělá inteligence je v dnešní době využívaná v podobě AI (Artificial Inteligency), jejíž definice je na první pohled vskutku jednoduchá, když použijeme definici inteligence a doplníme, že inteligenci budeme nazývat umělou, pokud je výtvorem jiné inteligence. AI je považována za primitivní formu umělé inteligence.

V případě AI se jedná o první pokusy o autonomně rozhodující systémy, které se dokonce ještě ani nemusí umět učit. Sem patří neuronové sítě, většina agentových a multiagentových systémů, expertní systémy, systémy na dokazování hypotéz a všechny ostatní algoritmy neschopné přesáhnout rámec vlastního naprogramování.

2.6 PATH-FINDING

Důležitou částí umělé inteligence realtimových počítačových her je hledání cest. Dnešní standard prohledávacích algoritmů, algoritmus A*, ve své základní verzi počítá celou cestu najednou dopředu. To však klade na rozsáhlejších mapách nebo při mnoha simultánně hledajících jednotkách vysoké nároky na výpočetní čas, navíc se podstatná část nalezené cesty nemusí použít kvůli dynamicky se měnícímu světu. Vznikají proto různé přístupy k tomu, jak daný problém lépe rozprostřít v čase a jak lépe reagovat na proměnlivost světa.

2.7 2D POČÍTAČOVÁ GRAFIKA

Použití obrazovek CRT považujeme za první pokrok v počítačové grafice vůbec. Jsou dva základní možné přístupy ke 2D grafice, tedy vektorová a rastrová grafika. Vektorová grafika ukládá přesná geometrická data, topologii a styl, jako například souřadnice bodů, propojení mezi body, které tvoří úsečky, vybarvení, tloušťku a možné vyplnění tvarů. Většina vektorových grafických systémů umožňuje použít další standardní tvary jako kružnice, čtverce a další. Ve většině případů je zapotřebí vektorový obraz převést na obraz rastrový, aby jej bylo možné zobrazit. Rastrová grafika je pravidelná síť pixelů, organizovaná jako dvourozměrná matice bodů. Každý pixel nese specifické informace, například o jasu, barvě, průhlednosti bodu, nebo kombinaci těchto hodnot. Obrázek v rastrové grafice má omezené rozlišení, které se udává počtem řádek a sloupců. Standardní monitory zobrazují rastrový obraz s rozlišením o velikosti 1280x1024 pixelů (tato hodnota udává počet sloupců x počet řádek pixelů). Dnes se často kombinuje rastrová a vektorová grafika v souborových formátech, jako je PDF či SWF.

2.8 RENDERING

Rendering je tvorba obrazu na základě počítačového modelu. Rendering obsahuje v závislosti na softwaru mnoho parametrů a nastavení, kterými lze ovlivnit konečný vzhled scény. Jde o odvětví počítačové grafiky. Zabývá se tvorbou obrazů, napodobující reálný svět. Je to zvláštní způsob vizualizace dat. Data jsou parametry popisující reálný nebo imaginární svět s jeho objekty a jejich vlastnostmi.

Zdroj: *encyklopedie WikipediA*

3. POPIS

Labyrint je počítačová adaptace stejnojmenné deskové hry německé společnosti Ravensburger. Labyrint může hrát až čtveřice hráčů. Devizou hry je především umělá inteligence pro hru proti počítači, interaktivní interface a v neposlední řadě přívětivé grafické zpracování.

3.1 PRAVIDLA

Každý hráč má za úkol najít v labyrintu cestu k jednotlivým artefaktům, které mu byly na začátku hry losováním přiděleny. Ovšem pozor, cesty se neustále posouvají a mění svůj tvar. Vyhrává ten, kdo první najde všechny artefakty.

Na počátku hry se kartičky s chodbami náhodně promíchají a následně rozprostřou po hrací ploše takovým způsobem, aby vznikl náhodný vzor chodeb. Vždy je jedna kartička navíc, neboť se používá během hry k zasouvání do řad. Všech 24 tajných artefaktů se zamíchá a rozdá rovnoměrně všem hráčům. Táhnout začíná hráč, který byl hrou určen jako první. Poté je pořadí hráčů určeno směrem otáčení hodinových ručiček. Na pořadí má rovněž vliv zvolená barva figurky, jejíž startovní pozice na hracím poli je neměnná.

Jakmile přijde na hráče řada, hra mu ukáže jeho cíl, ke kterému se musí dostat. Aby hráč došel k artefaktu, musí postupně měnit chodby bludiště vsouváním kartičky a potom postupovat svou figurkou. Zasunutí je na rozdíl od posunutí figurkou povinné. Pokud tak hráč neučiní, hra za něj náhodně zasune do libovolné řady, kdy již nemá možnost táhnout figurkou.

Zasunutí do aktivních řad je vskutku prosté. Hráč vsune po straně přebytečnou hrací kartičku do hracího pole tak, aby se na protější straně opět vysunula jedna hrací kartička. Tato kartička zůstává na kraji hracího plánu ležet tak dlouho, dokud není další hráč na tahu, který nesmí kartičku vsunout na stejném místě, kde byla právě vysunuta. To znamená, že se nesmí takto zrušit tah předchozího hráče. Místa, kde je možné vsunout kartičky do hracího pole, jsou označena šipkou.

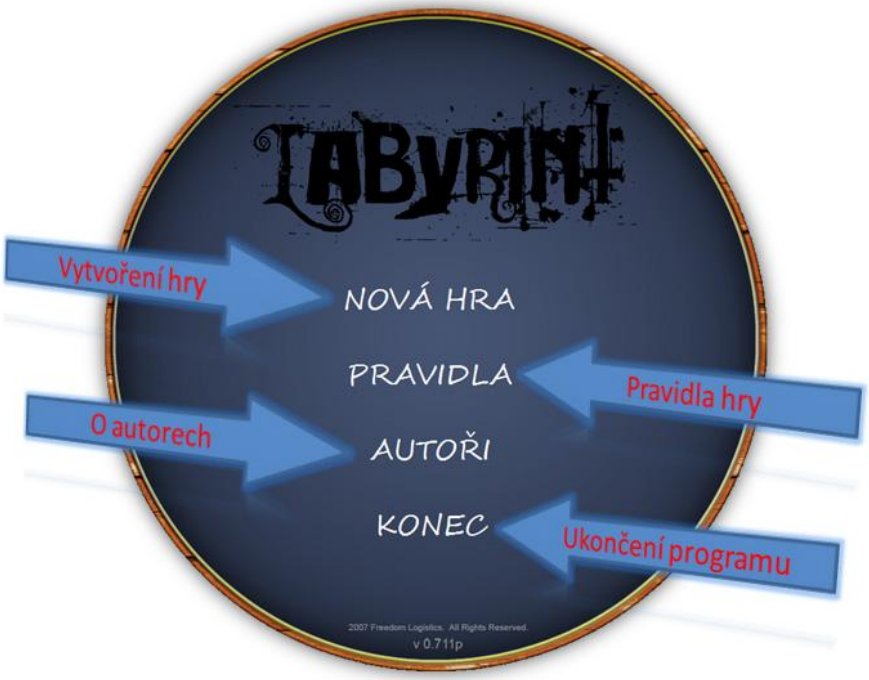
V zápětí na to, když hráč posunul chodby, postupuje figurkou. Hráč se smí přesunout na jakoukoliv kartičku bludiště, která je s výchozím bodem spojen nepřerušenou chodbou. Hráč nemusí figurkou postoupit, může tak zůstat na místě. Když hráč nemůže přímou cestou dojít k artefaktu, projde figurkou tak daleko, aby měl pokud možno výhodnou pozici pro příští tah. Pokud hráč došel k cíli, artefakt se přemístí do seznamu nalezených artefaktů. Jakmile na hráče přijde opět řada, hra mu odkryje další tajemnost, kterou musí najít.

Jakmile hráč najde všechny svoje tajemné artefakty, musí se dostat s figurkou opět k výchozímu bodu v rohu hracího pole. Vítězí hráč, který jako první najde všechny artefakty a dojde s figurkou na výchozí bod.

3.2 SPUŠTĚNÍ

Při spuštění programu se na ploše objeví hlavní menu (viz. Obrázek 1), kde se nachází hned čtyři možnosti, jak zasáhnout do hry. Tlačítko „Nová hra“ slouží pro vytvoření a následné zahájení nové hry. V pravidlech se každý hráč může podrobně seznámit se zákonitostmi a funkcionalitou hry Labyrint, přičemž pro ukončení hry slouží nejníže situované tlačítko „Konec“. Informace o autorech hry pak najde hráč pod tlačítkem „Autoři“.

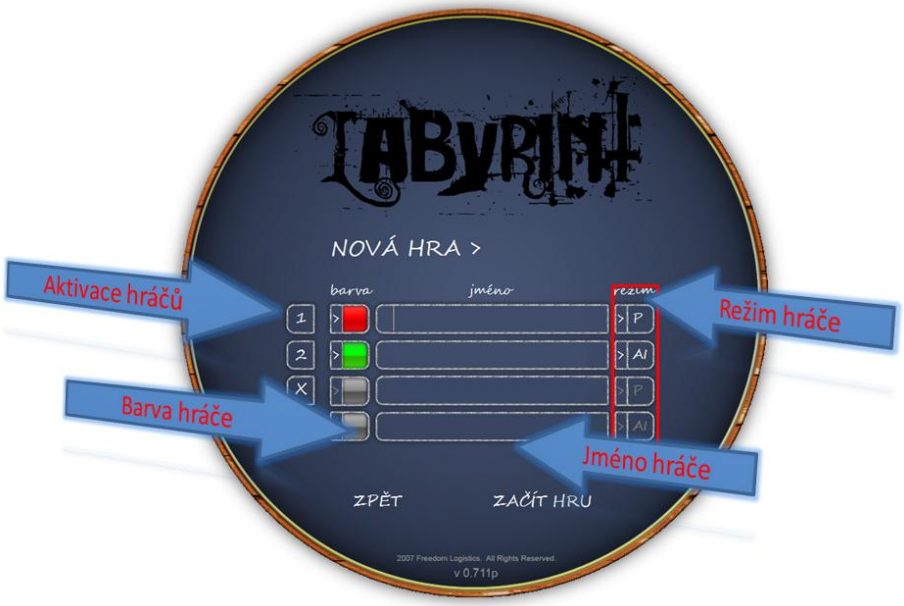
Obrázek 1 - Hlavní menu



3.3 NOVÁ HRA

Obrazovka „Nová hra“ slouží pro vytvoření a následné zahájení nové hry (viz. Obrázek 2). Nachází se zde intuitivní tabulka, která slouží pro jednoduché přidávání hráčů do hry, respektive odebírání hráčů ze hry, což lze provést v prvním případě kliknutím na křížek v řádku se zašedlou barvou, určující neaktivovanou kolonku. V druhém případě stačí stejným způsobem kliknout na číslo již aktivované kolonky.

Hráč si zde může vybrat vlastní barvu a napsat do kolonky, určené pro zadání jména, vlastní nick. Jméno musí být vždy vyplněné a nesmí mít duplicitní charakter. Posledním nastavením je „režim“, který určuje, zda-li se jedná o živého hráče nebo umělou inteligenci. Pro návrat zpět je možné využít stejnojmenného tlačítka. Hra se zahájí po kliknutí na tlačítko „začít hru“.



Obrázek 2 - Nová hra

3.4 HRACÍ PLOCHA

Hrací plocha je nejdůležitější obrazovkou hry, kde se celá hra odehrává. K dispozici je zde hned několik informačních panelů (viz. Obrázek 3). První z nich je „Hledaný artefakt“, který obsahuje informaci o aktuálně hledaném artefaktu daného hráče v podobě grafického vyobrazení. Stejným způsobem jsou prezentovány již nalezené artefakty, mezi nimiž se lze pomocí šipek přepínat. Seznam nalezených artefaktů se flexibilně mění podle vybraného hráče.

Zcela poslední panel obsahuje informace o hráči, přičemž se zde nachází údaje o počtu tahů a nalezených artefaktů. Mezi informacemi o hráčích lze rovněž přepínat šipkami. Jakmile přijde řada na dalšího hráče, informační tabule se ihned vyplní hodnotami o aktuálním hráči.

V horní části hrací plochy se nachází klasická tlačítka pro návrat do menu, minimalizaci okna a ukončení programu. V případě, že jde o přerušení hry, Labyrint se hráče zeptá, zda-li chce danou akci opravdu provést. Dolní část nabízí časový limit a tlačítko, které nejenže signalizuje barvu aktuálního hráče, ale také slouží jako tlačítko pro ukončení tahu po zasunutí hrací kartičky, když se hráč rozhodne nevyužít svůj tah figurkou, což jde taktéž provést klávesou „N“.



Obrázek 3 - Hrací plocha

Na hrací ploše se vždy nachází kartička, která slouží k následnému zasunutí. Uchopení kartičky se provádí klikem levého tlačítka myši. Klikem pravého tlačítka myši lze pak otáčet kartičkou ve směru otáčení hodinových ručiček. Zasunutí kartičky do aktivní řady je opět svěřeno levému tlačítku myši, stejně tak jako při tahu figurkou, kdy je nutné kliknout na určitou část cesty, kam se potřebuje hráč dostat.

Aktivní řady jsou označený šipkami a výchozí pozice se nachází v rozích hracího pole. Jakmile některý z hráčů vyhraje, Labyrint se zeptá zbylých hráčů, zda-li chtějí pokračovat ve hře.



Obrázek 4 - Detail hrací plochy

3.5 PRAVIDLA

V pravidlech se každý hráč může podrobně seznámit se zákonitostmi a funkcionalitou hry Labyrint. V nabídce je možno listovat pomocí šípek (viz. Obrázek 5). Pro návrat zpět je zde určeno stejnojmenného tlačítko.



Obrázek 5 - Pravidla

3.6 AUTOŘI

Obrazovka „Autoři“ je stejně jako nabídka „Pravidla“ zcela informativní. Uživatel se zde může dozvědět základní informace o tvůrcích hry (viz. Obrázek 6). Pro návrat zpět je zde určeno stejnojmenného tlačítka.



Obrázek 6 - Autoři

4. ZÁVĚR

Labyrint je komplexní hra, jejíž vývoj byl vzhledem k nulovým znalostem na poli vývoje počítačových her nesmírně složitý a časově náročný. Potýkali jsme se s mnoha problémy, ať už jde o vykreslování, plynulost nebo umělou inteligenci. Nebylo dne, kdy jsme neřešili nějaký oříšek, který se zpočátku zdál být naprostou banalitou. Labyrint je založen na vysokoúrovňovém objektově orientovaném programovacím jazyce C#, s nímž se programátor Ondřej Sýkora

teprve seznamoval, což rozhodně nepřispělo k samotnému vývoji, stejně jako blížící se maturita. Avšak byla to jediná cesta, kterou jsme se mohli vydat.

Přestože je na Labyrintu stále co vylepšovat, můžeme říct, že jsme vytvořili hru, která je nejen zábavná, ale také nutí hráče namáhat své mozkové závity. Rovněž jsme vymysleli a naprogramovali umělou inteligenci, která již v rané fázi vývoje porazila své stvořitele, což považujeme za obrovský úspěch. Hra disponuje příjemným grafickým prostředím, intuitivním ovládáním, za čímž stojí grafik a designér Ondřej Sýkora, Ondřej Sýkora, Tomáš Richter.

Co se týče celého projektu, snažili jsme neodbýt žádný element s ním spojený. Vše jsme položili na základech jednoduchosti a přehlednosti. Projekt nás donutil k spolupráci a potřebné komunikaci, přičemž jsme vše společně konzultovali. Většinu znalostí, které jsme v průběhu vývoje užili, jsme nabyli vlastní iniciativou doma, což umocňuje naše veskrze pozitivní pocity z finální verze hry, i když je Labyrint mnohými lidmi nedoceněn.

Ondřej Sýkora, Tomáš Richter

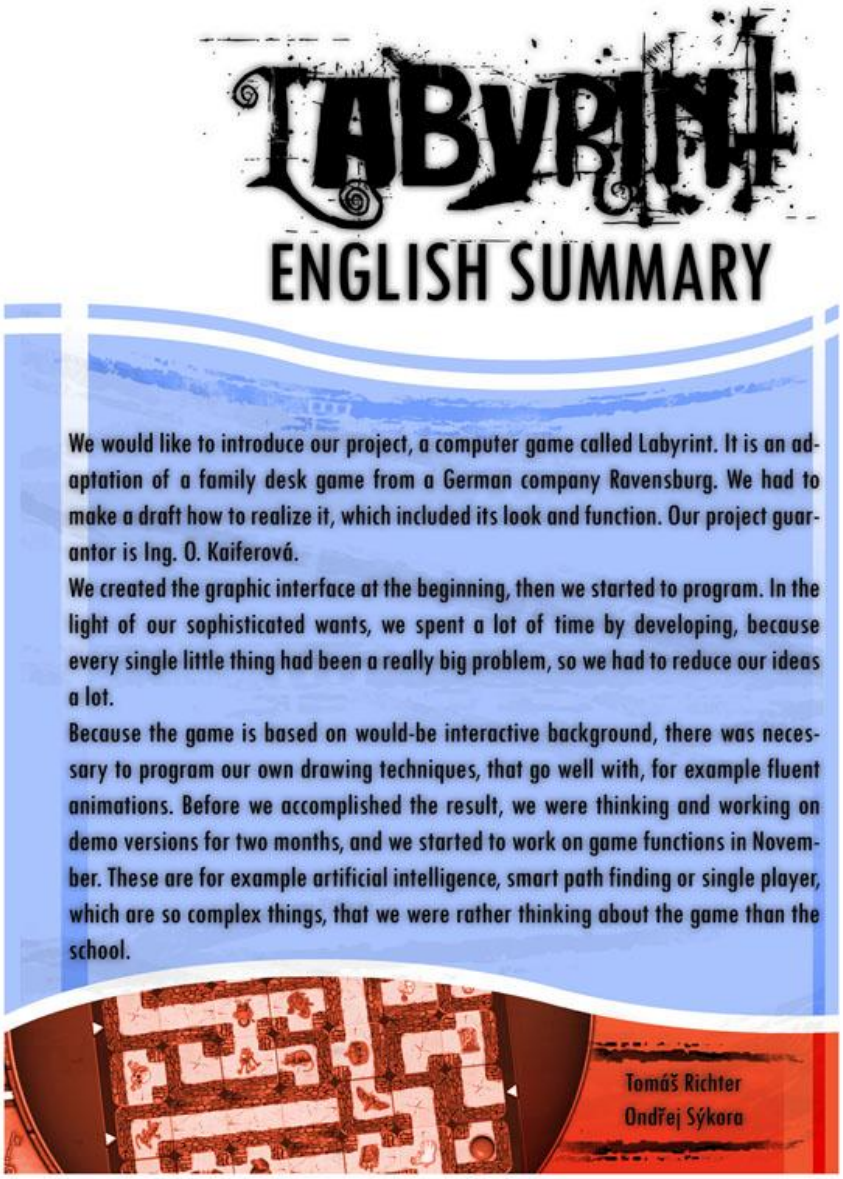
Příloha A LETÁK

A1 ČESKY



Obrázek 7 - Český leták

A2 ANGLICKY



Obrázek 8 - Anglický leták

Příloha B LABYRINT

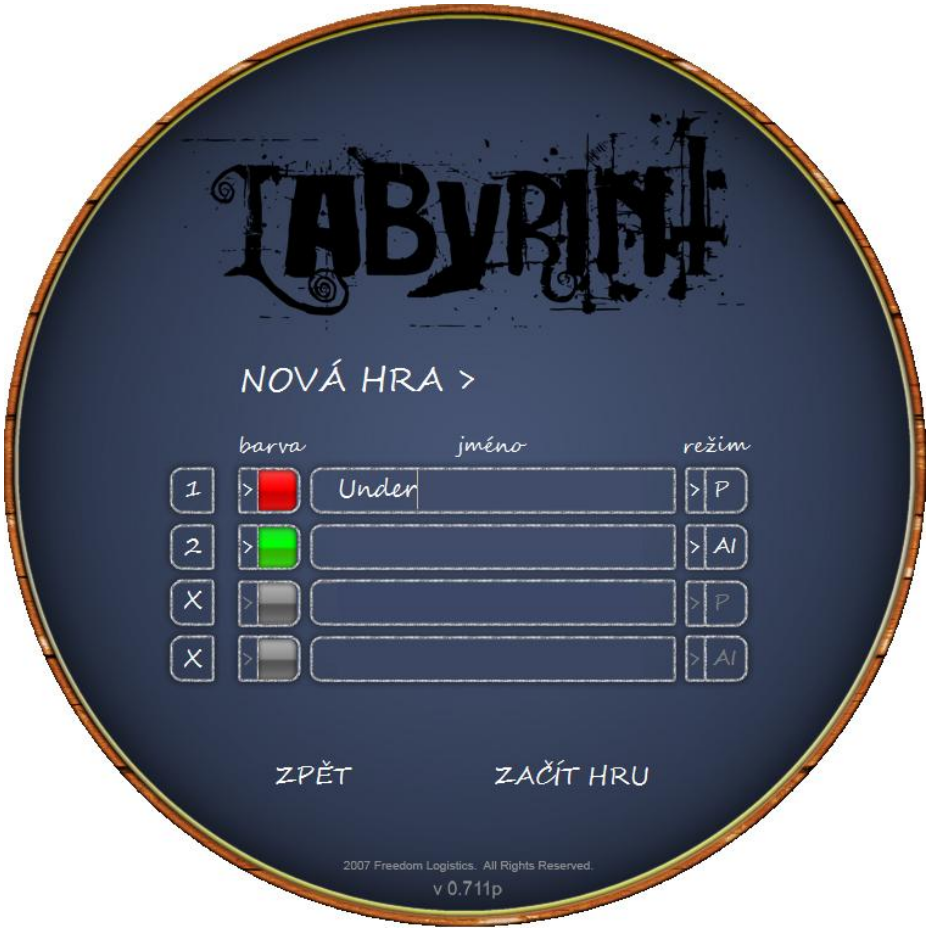
B1 HLAVNÍ MENU



Obrázek 9 - Hlavní menu

B2 NOVÁ HRA

Obrázek 10 - Nová hra



B3 HRACÍ PLOCHA

Obrázek 11 - Hrací plocha



B4 PRAVIDLA

Obrázek 12 - Pravidla



B5 AUTOŘI

Obrázek 13 - Autoři

Příloha C ZDROJOVÝ KÓD

C1 AI

```
using System;
using System.Drawing;
using System.Threading;
using System.Windows.Forms;

namespace Labyrint
{
   static class AI
   {
      private static Kamen[,] pracovniPole;
      private static Hra aktualniHra;
      private static HraciForm cilovyForm;
      private static Kamen posunovaci;
      private static Kamen[,] docasnePole;
      private static Random generNahodne = new Random();//trida pro generovani nahodnych cisel
      private static Point[] Cesta;
      private static Kamen aktualniKamen;
      private static System.Timers.Timer casovac = new System.Timers.Timer();
      private static Figurka[] proPosunuti = null;
      private static int pocetPosunuti = 0;

      //konstruktor
      static AI()
      {
         pracovniPole = null;
         casovac.Interval = 40;
         casovac.Elapsed += new System.Timers.ElapsedEventHandler(casovac_Elapsed);
      }
      //--------vlastnosti----
      public static Kamen[,] PracovniPole//nastavi pracovni pole, podle aktualni situace ve hre--------mozna nebude potreba
      {
         get
         {
            return pracovniPole;
         }
         set
         {
            Kamen[,] tmp = value;
            //vytvorim nove pole a do nej prekopiruju jednotlive hodnoty.
            pracovniPole = new Kamen[7, 7];
            for (int i = 0; i < 7; i++)
            {
               for (int j = 0; j < 7; j++)
               {
                  pracovniPole[i, j] = new Kamen();
                  pracovniPole[i, j].Artefakt = tmp[i, j].Artefakt;
                  bool[] hodnota=tmp[i,j].Cesty;
                  pracovniPole[i, j].Cesty = new bool[hodnota.Length];
                  for (int x = 0; x < hodnota.Length; x++)
                  {
                     if (hodnota[x] == true)
                     {
                        pracovniPole[i, j].Cesty[x] = true;
                     }
                     else
                     {
                        pracovniPole[i, j].Cesty[x] = false;
                     }
                  }
                  pracovniPole[i, j].JesteProhledat = tmp[i, j].JesteProhledat;
                  pracovniPole[i, j].Jmeno = tmp[i, j].Jmeno;

                  hodnota = tmp[i, j].PouzitelneSmery;
                  pracovniPole[i, j].PouzitelneSmery = new bool[hodnota.Length];
                  for (int x = 0; x < hodnota.Length; x++)
                  {
                     if (hodnota[x] == true)
```

```
                    {
                        pracovniPole[i, j].PouzitelneSmery[x] = true;
                    }
                    else
                    {
                        pracovniPole[i, j].PouzitelneSmery[x] = false;
                    }
                }

                pracovniPole[i, j].Pozice = tmp[i, j].Pozice;

                pracovniPole[i, j].PropojeneKameny = tmp[i, j].PropojeneKameny;
                pracovniPole[i, j].Rozmery = tmp[i, j].Rozmery;
                pracovniPole[i, j].TypKarty = tmp[i, j].TypKarty;
                pracovniPole[i, j].Uhel = tmp[i, j].Uhel;
                pracovniPole[i, j].VPracovnimPoli = tmp[i, j].VPracovnimPoli;
            }
        }
        aktualniKamen = pracovniPole[aktualniHra.AktualniHrac.Figurka.AktualniKamen.VPracovnimPoli.X,
aktualniHra.AktualniHrac.Figurka.AktualniKamen.VPracovnimPoli.Y];
    }
}
public static Kamen Posunovaci
{
    set
    {
        Kamen tmp = value;
        posunovaci = new Kamen();
        posunovaci.Artefakt = tmp.Artefakt;
        bool[] hodnota = tmp.Cesty;
        posunovaci.Cesty = new bool[hodnota.Length];
        for (int x = 0; x < hodnota.Length; x++)
        {
            if (hodnota[x] == true)
            {
                posunovaci.Cesty[x] = true;
            }
            else
            {
                posunovaci.Cesty[x] = false;
            }
        }
        posunovaci.JesteProhledat = tmp.JesteProhledat;
        posunovaci.Jmeno = tmp.Jmeno;

        hodnota = tmp.PouzitelneSmery;
        if (hodnota != null)
        {
            posunovaci.PouzitelneSmery = new bool[hodnota.Length];
            for (int x = 0; x < hodnota.Length; x++)
            {
                if (hodnota[x] == true)
                {
                    posunovaci.PouzitelneSmery[x] = true;
                }
                else
                {
                    posunovaci.PouzitelneSmery[x] = false;
                }
            }
        }

        posunovaci.Pozice = tmp.Pozice;

        posunovaci.PropojeneKameny = tmp.PropojeneKameny;
        posunovaci.Rozmery = tmp.Rozmery;
        posunovaci.TypKarty = tmp.TypKarty;
        posunovaci.Uhel = tmp.Uhel;
        posunovaci.VPracovnimPoli = tmp.VPracovnimPoli;
    }
}
public static Hra AktualniHra
{
    get
    {
        return aktualniHra;
```

```
            }
            set
            {
                aktualniHra = value;
            }
        }//nastavi odkaz na aktualni hru
        public static HraciForm CilovyForm
        {
            get
            {
                return cilovyForm;
            }
            set
            {
                cilovyForm = value;
            }
        }

        //---------metody------
        private static int NahodneCislo(int Od, int Do)//generuje nahodne cislo v zadanem rozsahu
        {
            int nahodneCislo = generNahodne.Next(Od, Do);
            return nahodneCislo;
        }
        //vrati souradnice ciloveho kamene
        private static VariantaZasunuti FindBestPath(int PosledniRadekSloupec, int PosledniSmer, int PosledniCisloRadku, Kamen
VsunovanaKarta,Kamen[,] PracovniPole)//na zaklade posunuti v pameti vyhleda nejoptimalnejsi cestu pro umelou inteligenci
        {
            //figurka stoji na nejakem kameni, od ktereho se bude vychazet
            //
            //vyhleda cestu posunutim do vsech moznych smeru------celkem 11 v jednom tahu
            //nejdrice potrebuju vzit cil, kam se cesta bude hledat


            //nejdrive posunu v pameti pro jeden smer
            //vyhledam cestu
            //najdu konce cesty a zmerim vzdalenost k hledanemu artefaktu
            //ulozim cislo cesty, jeji souradnice i koncovy kamen, na ktery bych se mel s figurkou presunout
            //jdu na dalsi cestu

            Kamen posunovaci = VsunovanaKarta;
            pracovniPole = PracovniPole;

            int radeksloupec = 0;
            int cislo = 0;
            int smer = 0;
            Point cil = new Point(0,0) ;//cilovy kamen
            bool cestaNalezena = false;
            VariantaZasunuti[] poleVariant = new VariantaZasunuti[0];
            VariantaZasunuti varianta = null;

            for (int zasunuti = 0; zasunuti < 12; zasunuti++)//projdu vsechna mozna zasunuti
            {
                //prochazim po smeru hodinovych rucicek
                //hodnoty pro ruznych smeru pro hledani
                //radekSloupec - 1=radek, 2=sloupec
                //smer - jestlize se jedna o radek: 1=doleva, 2=doprava
                //       jestlize se jedna o sloupec: 1=dolu, 2=nahoru
                switch (zasunuti)
                {
                    case 0:
                        radeksloupec = 2;
                        smer = 1;
                        cislo = 1;
                        break;
                    case 1:
                        radeksloupec = 2;
                        smer = 1;
                        cislo = 3;
                        break;
                    case 2:
                        radeksloupec = 2;
                        smer = 1;
                        cislo = 5;
                        break;
                    case 3:
```

```
                radeksloupec = 1;
                smer = 1;
                cislo = 1;
                break;
            case 4:
                radeksloupec = 1;
                smer = 1;
                cislo = 3;
                break;
            case 5:
                radeksloupec = 1;
                smer = 1;
                cislo = 5;
                break;
            case 6:
                radeksloupec = 2;
                smer = 2;
                cislo = 5;
                break;
            case 7:
                radeksloupec = 2;
                smer = 2;
                cislo = 3;
                break;
            case 8:
                radeksloupec = 2;
                smer = 2;
                cislo = 1;
                break;
            case 9:
                radeksloupec = 1;
                smer = 2;
                cislo = 5;
                break;
            case 10:
                radeksloupec = 1;
                smer = 2;
                cislo = 3;
                break;
            case 11:
                radeksloupec = 1;
                smer = 2;
                cislo = 1;
                break;
        }

        if (PosledniRadekSloupec == radeksloupec && PosledniCisloRadku == cislo && PosledniSmer != smer)//jedna se o stejne
zasunuti jako predtim, cili neplatne
        {
            continue;
        }
        else
        {
            //u kazdeho zasunuti jsou 4 moznosti natoceni kamene, takze je vyskousim vsechny
            for (int natoceni = 0; natoceni < 4; natoceni++)
            {
                //natocim posunovaci kamen
                posunovaci.Uhel += 90f;
                //zasunu
                docasnePole = Posunout(radeksloupec, smer, cislo, posunovaci);//cilovyForm.Posunovaci); //posunovaci);//vrati nove
pracovni pole, ve kterem muzu vyhledat cestu
                //aktualizuju cesty na plose
                AktualizujPropojeniCest(docasnePole);
                //vyhledam cestu
                //jako cil uvedu souradnice hledaneho artefaktu
                bool nalezeno = false;
                for (int i = 0; i < 7; i++)
                {
                    for (int j = 0; j < 7; j++)
                    {
                        if (aktualniHra.AktualniHrac.HledanaKarta.Artefakt == docasnePole[i, j].Artefakt)//kdyz je hledany mimo hraci
plochu, cil je nastaven na predchozi hodnotu
                        {
                            cil = new Point(i, j);
                            nalezeno = true;
                            break;
```

```
                        }
                    }
                    if (nalezeno) { break; }
                }
                if (nalezeno == false) { break; }//cil nebyl v poli nalezen, tudiz je mimo a nema cenu pokracovat v danem zasunuti
                VytvorCestu(cil, docasnePole);//pokusi se vytvorit cestu k hledanemu artefaktu
                if (Cesta.Length != 0)//nejaka cesta byla vytvorena/nalezena
                {
                    varianta = new VariantaZasunuti(radeksloupec, cislo, smer, cil, posunovaci.Uhel);
                    cestaNalezena = true;
                    break;//ukonci cyklus
                }
                else//cesta nebyla vytvorena, tudiz hledany artefakt nebyl nalezen na vznikle ceste
                {
                    //urci optimalni kamen pro posunuti
                    //cilovy kamen ulozi do pole pro optimalni kameny-bude jich vice

                    //----------urceni optimalniho kamene jako cilovy----------//
                    Point souradnice = NajdiCestu(docasnePole);

                    //----------konec------------//

                    cestaNalezena = false;
                    if (poleVariant.Length == 0)
                    {
                        poleVariant = new VariantaZasunuti[1];
                        poleVariant[0] = new VariantaZasunuti(radeksloupec, cislo, smer, souradnice, posunovaci.Uhel);
                    }
                    else
                    {
                        VariantaZasunuti[] tmp = poleVariant;
                        poleVariant = new VariantaZasunuti[poleVariant.Length + 1];
                        tmp.CopyTo(poleVariant, 0);
                        poleVariant[poleVariant.Length - 1] = new VariantaZasunuti(radeksloupec, cislo, smer, souradnice,
posunovaci.Uhel);
                    }
                }
                //dalsi iterace cyklu natoceni
                //jednotlivym kamenum v hracim poli nastavim vlastnost VPracovnimPoli na stejnou hosnotu jakou meli pred vstupem do
cyklu
                //je to dulezite pro dalsi iteraci cyklu
                for (int i = 0; i < 7; i++)
                {
                    for (int j = 0; j < 7; j++)
                    {
                        pracovniPole[i, j].VPracovnimPoli = aktualniHra.Kameny[i, j].VPracovnimPoli;
                    }
                }
                aktualniKamen = pracovniPole[aktualniHra.AktualniHrac.Figurka.AktualniKamen.VPracovnimPoli.X,
aktualniHra.AktualniHrac.Figurka.AktualniKamen.VPracovnimPoli.Y];
            }
            //konec cyklu natoceni
        }
        //dalsi iterace cyklu zasunuti
        for (int i = 0; i < 7; i++)
        {
            for (int j = 0; j < 7; j++)
            {
                pracovniPole[i, j].VPracovnimPoli = aktualniHra.Kameny[i, j].VPracovnimPoli;
            }
        }
        aktualniKamen = pracovniPole[aktualniHra.AktualniHrac.Figurka.AktualniKamen.VPracovnimPoli.X,
aktualniHra.AktualniHrac.Figurka.AktualniKamen.VPracovnimPoli.Y];
        if (cestaNalezena)
        {
            break;
        }
    }
    //konec cyklu zasunuti

    if (cestaNalezena)//cesta byla nalezena jeste v cyklu a ten byl nasilne ukoncen
    {
        return varianta;
    }
    else//cesta nebyla nelezena ani po vsech zasunutich, takze prichazi na radu urceni optimalni varianty
    {
```

```
        //projde pole variant a urci, ktera je nejlepsi
        //varianta = poleVariant[5];
        varianta = UrciNejlepsiVariantu(poleVariant);
        return varianta;
    }
}

//-------imaginarne si posune v pameti hraci plochu----------
//vrati nove vytvorene pracovni pole
private static Kamen[,] Posunout(int radekSloupec, int smer, int cisloRadkuSloupce, Kamen vsunovanaKarta)
{
    //radekSloupec - 1=radek, 2=sloupec
    //smer - jestlize se jedna o radek: 1=doleva, 2=doprava
    //       jestlize se jedna o sloupec: 1=dolu, 2=nahoru
    //zjistim o jaky typ posunuti se vlastne jedna
    Kamen tmpVsunovana = null;
    Kamen[,] PracovniPoleOtisk = new Kamen[7, 7];
    //prekopiruje pracovni pole do jineho pole a tim vytvori kopii usporadani, odkazuje se porad na stejne kameny.
    for (int i = 0; i <= 6; i++)
    {
        for (int j = 0; j <= 6; j++)
        {
            PracovniPoleOtisk[i, j] = pracovniPole[i, j];
        }
    }

    switch (radekSloupec)
    {
        case 1://radek
            switch (smer)
            {
                case 1://doleva
                    for (int j = cisloRadkuSloupce; j == cisloRadkuSloupce; j++)//radek je porad stejny
                    {
                        for (int i = 0; i <= 7; i++)//sloupce se meni
                        {
                            if (i == 0)
                            {
                                tmpVsunovana = PracovniPoleOtisk[j, i];//karta, ktera se vysunuje ven z hraciho pole
                            }
                            else if (i == 7)//karta, ktera se vsunuje do hraciho pole
                            {
                                //ulozeni v poli
                                PracovniPoleOtisk[j, i - 1] = vsunovanaKarta;
                                PracovniPoleOtisk[j, i - 1].VPracovnimPoli = new Point(j, i - 1);
                            }
                            else//karty, ktere se posunuji vne hraciho pole
                            {
                                //ulozeni v poli
                                PracovniPoleOtisk[j, i - 1] = pracovniPole[j, i];
                                PracovniPoleOtisk[j, i - 1].VPracovnimPoli = new Point(j, i - 1);
                            }
                        }
                    }
                    break;
                case 2://doprava
                    for (int j = cisloRadkuSloupce; j == cisloRadkuSloupce; j++)//radek je porad stejny
                    {
                        for (int i = 6; i >= -1; i--)//sloupce se meni
                        {
                            if (i == 6)
                            {
                                tmpVsunovana = pracovniPole[j, i];//karta, ktera se vysunuje ven z hraciho pole
                            }
                            else if (i == -1)//karta, ktera se vsunuje do hraciho pole
                            {
                                //ulozeni v poli
                                PracovniPoleOtisk[j, i + 1] = vsunovanaKarta;
                                PracovniPoleOtisk[j, i + 1].VPracovnimPoli = new Point(j, i + 1);
                            }
                            else//karty, ktere se posunuji vne hraciho pole
                            {
                                //ulozeni v poli
                                PracovniPoleOtisk[j, i + 1] = pracovniPole[j, i];
                                PracovniPoleOtisk[j, i + 1].VPracovnimPoli = new Point(j, i + 1);
                            }
```

```
                        }
                    }
                    break;
                }
                break;
            case 2://sloupec
                switch (smer)
                {
                    case 1://dolu
                        for (int i = 6; i >= -1; i--)
                        {
                            for (int j = cisloRadkuSloupce; j == cisloRadkuSloupce; j++)
                            {
                                if (i == 6)
                                {
                                    tmpVsunovana = pracovniPole[i, j];//karta, ktera se vysunuje ven z hraciho pole
                                }
                                else if (i == -1)//karta, ktera se vsunuje do hraciho pole
                                {
                                    //ulozeni v poli
                                    PracovniPoleOtisk[i + 1, j] = vsunovanaKarta;
                                    PracovniPoleOtisk[i + 1, j].VPracovnimPoli = new Point(i + 1, j);
                                }
                                else//karty, ktere se posunuji vne hraciho pole
                                {
                                    //ulozeni v poli
                                    PracovniPoleOtisk[i + 1, j] = pracovniPole[i, j];
                                    PracovniPoleOtisk[i + 1, j].VPracovnimPoli = new Point(i + 1, j);
                                }
                            }
                        }
                        break;
                    case 2://nahoru
                        for (int i = 0; i <= 7; i++)
                        {
                            for (int j = cisloRadkuSloupce; j == cisloRadkuSloupce; j++)
                            {
                                if (i == 0)
                                {
                                    tmpVsunovana = pracovniPole[i, j];//karta, ktera se vysunuje ven z hraciho pole
                                }
                                else if (i == 7)//karta, ktera se vsunuje do hraciho pole
                                {
                                    //zmenim v poli
                                    PracovniPoleOtisk[i - 1, j] = vsunovanaKarta;
                                    PracovniPoleOtisk[i - 1, j].VPracovnimPoli = new Point(i - 1, j);
                                }
                                else//karty, ktere se posunuji vne hraciho pole
                                {
                                    //zmenim v poli
                                    PracovniPoleOtisk[i - 1, j] = pracovniPole[i, j];
                                    PracovniPoleOtisk[i - 1, j].VPracovnimPoli = new Point(i - 1, j);
                                }
                            }
                        }
                        break;
                }
                break;
            }
            //zkontroluju, zda na tmpVsunovana(kamen, ktery se vysunul ven) nebyla figurka, kdyz jo, tak nastavim aktualni kamen jako ten
naproti
            if (tmpVsunovana == aktualniKamen)//figurka byla vysunuta ven
            {
                if (tmpVsunovana.VPracovnimPoli == new Point(1, 0))//na levem okraji na druhem radku
                {
                    aktualniKamen = PracovniPoleOtisk[1, 6];
                }
                else if (tmpVsunovana.VPracovnimPoli == new Point(3, 0))//na levem okraji na ctvrtem radku
                {
                    aktualniKamen = PracovniPoleOtisk[3, 6];
                }
                else if (tmpVsunovana.VPracovnimPoli == new Point(5, 0))//na levem okraji na sestem radku
                {
                    aktualniKamen = PracovniPoleOtisk[5, 6];
                }
                else if (tmpVsunovana.VPracovnimPoli == new Point(1, 6))//na pravem okraji na druhem radku
```

```
        {
            aktualniKamen = PracovniPoleOtisk[1, 0];
        }
        else if (tmpVsunovana.VPracovnimPoli == new Point(3, 6))//na pravem okraji na ctvrtem radku
        {
            aktualniKamen = PracovniPoleOtisk[3, 0];
        }
        else if (tmpVsunovana.VPracovnimPoli == new Point(5, 6))//na pravem okraji na sestem radku
        {
            aktualniKamen = PracovniPoleOtisk[5, 0];
        }
        else if (tmpVsunovana.VPracovnimPoli == new Point(0, 1))//na hornim okraji na druhem sloupci
        {
            aktualniKamen = PracovniPoleOtisk[6, 1];
        }
        else if (tmpVsunovana.VPracovnimPoli == new Point(0, 3))//na hornim okraji na ctvrtem sloupci
        {
            aktualniKamen = PracovniPoleOtisk[6, 3];
        }
        else if (tmpVsunovana.VPracovnimPoli == new Point(0, 5))//na hornim okraji na sestem sloupci
        {
            aktualniKamen = PracovniPoleOtisk[6, 5];
        }
        else if (tmpVsunovana.VPracovnimPoli == new Point(6, 1))//na dolnim okraji na druhem sloupci
        {
            aktualniKamen = PracovniPoleOtisk[0, 1];
        }
        else if (tmpVsunovana.VPracovnimPoli == new Point(6, 3))//na dolnim okraji na ctvrtem sloupci
        {
            aktualniKamen = PracovniPoleOtisk[0, 3];
        }
        else if (tmpVsunovana.VPracovnimPoli == new Point(6, 5))//na dolnim okraji na sestem sloupci
        {
            aktualniKamen = PracovniPoleOtisk[0, 5];
        }
    }
    return PracovniPoleOtisk;
}

//posunuje po hraci plose posuvny kamen
public static void PosunPoPlose(int RadekSloupec, int Cislo, int Smer, HraciForm Formular)
{
    Point pozice = Formular.Posunovaci.Pozice;
    //najdu ji v souradicich
    int indexVSouradnicich = -1;
    for (int i = 0; i < Formular.pocatecniPovolene.Length; i++)
    {
        if (pozice == Formular.pocatecniPovolene[i])
        {
            indexVSouradnicich = i;
            break;
        }
    }
    //radekSloupec - 1=radek, 2=sloupec
    //smer - jestlize se jedna o radek: 1=doleva, 2=doprava
    //       jestlize se jedna o sloupec: 1=dolu, 2=nahoru
    int indexKamZasunout = -1;
    switch (RadekSloupec)
    {
        case 1://radek
            switch (Smer)
            {
                case 1://doleva
                    switch (Cislo)
                    {
                        case 1:
                            indexKamZasunout = 3;
                            break;
                        case 3:
                            indexKamZasunout = 4;
                            break;
                        case 5:
                            indexKamZasunout = 5;
                            break;
                    }
                    break;
```

```
                    case 2://doprava
                        switch (Cislo)
                        {
                            case 1:
                                indexKamZasunout = 11;
                                break;
                            case 3:
                                indexKamZasunout = 10;
                                break;
                            case 5:
                                indexKamZasunout = 9;
                                break;
                        }
                        break;
                }
                break;
            case 2://sloupec
                switch (Smer)
                {
                    case 1://dolu
                        switch (Cislo)
                        {
                            case 1:
                                indexKamZasunout = 0;
                                break;
                            case 3:
                                indexKamZasunout = 1;
                                break;
                            case 5:
                                indexKamZasunout = 2;
                                break;
                        }
                        break;
                    case 2://nahoru
                        switch (Cislo)
                        {
                            case 1:
                                indexKamZasunout = 8;
                                break;
                            case 3:
                                indexKamZasunout = 7;
                                break;
                            case 5:
                                indexKamZasunout = 6;
                                break;
                        }
                        break;
                }
                break;
        }
        if (indexKamZasunout == indexVSouradnicich)//nachazim se na souradnicich, kam mam kamen posunout, takze se posouvat ani
nemusim
        {
        }
        else//musim najit cestu, po ktere posunu kamen
        {
            //vytvorim si specialni pole indexu, ve kterem budu hledat nejkratsi moznou cestu k cilove souradnici
            int[] poleIndexu = new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 };
            int[] cesta = null;//pro uladani cest, maximalne 2

            //nejdrive budu prochazet smerem doprava
            int poziceVPoli = -1;
            for (int i = 0; i < poleIndexu.Length; i++)//zjitim kde v poli indexu se nachazim
            {
                if (poleIndexu[i] == indexVSouradnicich)
                {
                    poziceVPoli = i;
                    break;
                }
            }
            //hledam od dalsiho indexu
            for (int i = poziceVPoli + 1; i < poleIndexu.Length; i++)
            {
                //pridam index do cesty
                if (cesta == null)
                {
```

```
                    cesta = new int[1];
                    cesta[0] = poleIndexu[i];
                }
                else
                {
                    int[] tmp = cesta;
                    cesta = new int[cesta.Length + 1];
                    tmp.CopyTo(cesta, 0);
                    cesta[cesta.Length - 1] = poleIndexu[i];
                }

                if (poleIndexu[i] == indexKamZasunout)//uz jsem dorazil k cilovemu indexu
                {
                    break;
                }
            }
            //
            //
            //
            if (cesta.Length > 6)//neni urcite nejkratsi
            {
                //ted projdu pole indexu doleva, ale od nove pozice, najdu totiz hledany index znova od predesleho
                for (int i = poziceVPoli + 1; i < poleIndexu.Length; i++)//zjitim kde v poli indexu se nachazim
                {
                    if (poleIndexu[i] == indexVSouradnicich)
                    {
                        poziceVPoli = i;
                        break;
                    }
                }
                //vytvorim druhou cestu
                bool prvniPruchod = true;
                for (int i = poziceVPoli - 1; i >= 0; i--)
                {
                    //pridam index do cesty
                    if (prvniPruchod)
                    {
                        cesta = new int[1];
                        cesta[0] = poleIndexu[i];
                        prvniPruchod = false;
                    }
                    else
                    {
                        int[] tmp = cesta;
                        cesta = new int[cesta.Length + 1];
                        tmp.CopyTo(cesta, 0);
                        cesta[cesta.Length - 1] = poleIndexu[i];
                    }

                    if (poleIndexu[i] == indexKamZasunout)//uz jsem dorazil k cilovemu indexu
                    {
                        break;
                    }
                }
                //nasel jsem cestu a ted budu postupne vykreslovat na formular
                for (int i = 0; i < cesta.Length; i++)
                {
                    Formular.Posunovaci.Pozice = Formular.pocatecniPovolene[cesta[i]];
                    Formular.Refresh();
                }
            }

        }

        private static void AktualizujPropojeniCest(Kamen[,] kameny)
        {
            //postupne prochazim jednotlive kameny v pracovnim poli, kontroluju, zda nejsou na okraji a potom nastavuju jejich vlastnost
propojene cesty
            //vysledkem metody je, ze kazdy kamen bude mit nastaveny smery, ve kterych je propojen se sousednim kamenem
            Kamen praveVybrany;
            int pocetCest;
            bool[] smery;

            //prochazim radky
            for (int y = 0; y < 7; y++)//prochazim y souradnice, takze po radcich
```

```
{
   //prochazim sloupce
   for (int x = 0; x < 7; x++)
   {
      //vemu kamen a vytvorim mu pouzitelne smery, zatim prazdne
      praveVybrany = kameny[y, x];
      praveVybrany.PouzitelneSmery = new bool[4];

      //zjistim, kolik cest danny kamen obsahuje-krizovatka ma 3, zatacka i rovina 2
      pocetCest = praveVybrany.Cesty.Length;

      //zjistim smery, do kterych tyto cesty smeruji-urcuje se na zaklade natoceni
      smery = praveVybrany.Cesty;

      //zkontroluju, zda dany smer nemiri ven z hraci plochy, tudiz jestli je danny smer pouzitelny pro spojeni cesty
      //postupne projdu vsechny smery, i ty, ktere kamen vubec nema implementovane
      for (int i = 0; i < 4; i++)
      //0-doprava;1-dolu;2-doleva;3-nahoru
      {
         //zjistim, jestli kamen ma cestu smerujici timto smerem
         //kdyz je true, tak ji ma, kdyz false, tak ji nema
         if (smery[i])//kamen ma cestu, smerujici timto smerem
         {
            //na zaklade toho, o jaky smer se jedna rozhodnu, zda bude pouzitelny pro pruchodnost nebo ne
            switch (i)//smer
            {
               case 0://doprava
                  if (x == 6)//pozicne se nachazim na pravem okraji
                  {
                     praveVybrany.PouzitelneSmery[i] = false;
                  }
                  else
                  {
                     praveVybrany.PouzitelneSmery[i] = true;
                  }
                  break;
               case 1://dolu
                  if (y == 6)//nachazim se dole
                  {
                     praveVybrany.PouzitelneSmery[i] = false;
                  }
                  else
                  {
                     praveVybrany.PouzitelneSmery[i] = true;
                  }
                  break;
               case 2://doleva
                  if (x == 0)
                  {
                     praveVybrany.PouzitelneSmery[i] = false;
                  }
                  else
                  {
                     praveVybrany.PouzitelneSmery[i] = true;
                  }
                  break;
               case 3://nahoru
                  if (y == 0)
                  {
                     praveVybrany.PouzitelneSmery[i] = false;
                  }
                  else
                  {
                     praveVybrany.PouzitelneSmery[i] = true;
                  }
                  break;
            }

         }
         else//danny smer neobsahuje, takze mu ho rovnou nastavim jako nepouzitelny
         {
            praveVybrany.PouzitelneSmery[i] = false;
         }
      }
   }
}
```

```
            //vsechny kameny ted vi, ktere smery maji opravdu pouzitelne.
        //jestlize ma cestu doprava a nachazi se na levem okraji, je cesta nepouzitelna

        //jednotlivym kamenum prirazuju souradnice kamenu, se kterymi maji pruchodne cesty
        //vezmu kamen, projdu jeho smery a u kazdeho smeru prohledam sousedni kamen-ten musi mit smer opacny, aby se jednalo o
pruchodnou cestu

        //prochazim radky
        for (int y = 0; y < 7; y++)
        {
            //sloupce
            for (int x = 0; x < 7; x++)
            {
                Point[] souradnice = null;
                //vemu kamen
                praveVybrany = kameny[y, x];
                int iks = 0, ypsylon = 0, smer = 0;

                //postupne projdu vsechny smery, vcetne tech, ktere kamen nema inmplementovane, ale to nemuzu predem vedet
                for (int i = 0; i < 4; i++)//vsechny smery
                {
                    //zjistim, zda ma kamen danny smer nastaven na true, tedy povolen
                    if (praveVybrany.PouzitelneSmery[i])//je povolen
                    {
                        //na zaklade smeru rozhodnu ktery vedlejsi kamen mam kontrolovat
                        switch (i)
                        {
                            //kdyz ma smer doprava, znamena to, ze bude prohledava kamen napravo, zda nema cestu, ktera smeruje nalevo
                            case 0://doprava
                                smer = 2;
                                iks = x + 1;
                                ypsylon = y;
                                break;
                            case 1://dolu
                                smer = 3;
                                iks = x;
                                ypsylon = y + 1;
                                break;
                            case 2://doleva
                                smer = 0;
                                iks = x - 1;
                                ypsylon = y;
                                break;
                            case 3://nahoru
                                smer = 1;
                                iks = x;
                                ypsylon = y - 1;
                                break;
                        }

                        //zkontroluje, zda kamen vedle ma cestu v pozadovanem smeru
                        if (kameny[ypsylon, iks].Cesty[smer])//kamen tuto cestu obsahuje
                        {
                            //do pole souradnic pridam souradnice kamene, ktery sdili cestu s timto kamenem, tedy maji propojenou cestu
                            //hledam prvni smer, tudiz souradnice jeste nejsou zaplneny
                            if (souradnice == null)
                            {
                                souradnice = new Point[1];
                                souradnice[0] = new Point(ypsylon, iks);
                            }
                            else
                            {
                                Point[] tmp = new Point[souradnice.Length];
                                tmp = souradnice;
                                souradnice = new Point[souradnice.Length + 1];
                                tmp.CopyTo(souradnice, 0);
                                souradnice[souradnice.Length - 1] = new Point(ypsylon, iks);
                            }
                        }
                    }
                }
                //u kamene jsem prosel vsechny smery a zjistil, ve kterych ma propojenou cestu s ostatnimi kameny
                //tyto souradnice nyni zkopiruju do jeho vlastnosti
                praveVybrany.PropojeneKameny = souradnice;
            }
```

```
            }
            //prosel jsem vsechny kameny v na plose
        }

        private static bool VytvorCestu(Point cil, Kamen[,] kameny)
        {
            try
            {
                Point[] buffer = new Point[0];//buffer pro ukladani krizovatek
                //ulozim kamen, na kterem stoji aktualni hrac
                Kamen pocatecniKamen = aktualniKamen;
                //cestu vytvorim prochazenim jednotlivych propojenych kamenu, ktere implementuje kazdy kamen. Kdyz zadny nema, ma
hodnoty null
                int pocetPropojeni;
                Point[] cesty = new Point[0];
                Kamen predchoziKarta = null;//souradnice karty, ze ktere jsem dorazil na aktualni kartu-vyhledam  v pracovnim poli a mam
kamen
                Kamen aktualniKarta = pocatecniKamen;
                int jesteProhledat;//pocet cest k prohledani na dannem kamnu
                if (pocatecniKamen.PropojeneKameny != null)
                {
                    jesteProhledat = 1;//pocetPropojeni;
                }
                else
                {
                    jesteProhledat = 0;
                }
                //prochazim kamen po kameni a zapisuju do pole pro cestu
                int index;//slouzi pro vyber cesty z pole cest kamene
                int rozmery = cesty.Length;
                //---------------zde je nekde chyba, tuhne to-----------------
                while (jesteProhledat != 0)
                {
                    if (aktualniKarta.JesteProhledat == null)//kdyz je prazdna, tak hledam poprve
                    {
                        aktualniKarta.JesteProhledat = aktualniKarta.PropojeneKameny;//pro zacatek se zde ulozi vsechny kameny, se kterymi je
propojen
                        if (predchoziKarta != null)//to znamena, ze jdu z prvni karticky-vychozi pozice, jinak tato situace nemuze nastat
                        {
                            //postupne porovnavam minule souradnice se souradnicemi v poli JesteProhledat, a kdyz je najdu, tak je odstranim z
pole
                            for (int i = 0; i < aktualniKarta.JesteProhledat.Length; i++)
                            {
                                if (predchoziKarta.VPracovnimPoli.X == aktualniKarta.JesteProhledat[i].X && predchoziKarta.VPracovnimPoli.Y
== aktualniKarta.JesteProhledat[i].Y)
                                {
                                    Point[] tmp = aktualniKarta.JesteProhledat;//docasne ulozim mimo
                                    aktualniKarta.JesteProhledat = new Point[aktualniKarta.JesteProhledat.Length - 1];//zmensim
                                    //nakopiruju zpatky
                                    //ten minuly kamen je na souradnicich "i", takze zpatky nakopiruju vsechny krome teto
                                    int indexPole = 0;
                                    for (int j = 0; j < tmp.Length; j++)
                                    {
                                        if (j != i)
                                        {
                                            aktualniKarta.JesteProhledat[indexPole] = tmp[j];
                                            indexPole += 1;
                                        }
                                    }
                                }
                            }
                        }
                        jesteProhledat = aktualniKarta.JesteProhledat.Length;
                    }
                    else//prohledavam kameny z bufferu
                    {
                        jesteProhledat = aktualniKarta.JesteProhledat.Length;
                    }
                    //nasledne najdu tu, ze ktere jsem prisel a tu z pole odstranim
                    int prvkuNaRozmer;
                    switch (jesteProhledat)
                    {
                        case 0://konec cesty
                            //prohledat buffer, kdyz je prazdny, tak ukoncit. Jinak zvolit jako aktualni kamen ten v bufferu
                            if (buffer.Length == 0)
                            {
```

```
                    Point[] temp = cesty;
                    //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                    prvkuNaRozmer = temp.Length;//temp.GetLength(1);
                    //rozmery + 1 pak upravit zpatky na rozmery
                    cesty = new Point[prvkuNaRozmer + 1];
                    temp.CopyTo(cesty, 0);
                    cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                    jesteProhledat = 0;
                }
                else
                {
                    Point[] temp = cesty;
                    //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                    prvkuNaRozmer = temp.Length;//temp.GetLength(1);
                    //rozmery + 1 pak upravit zpatky na rozmery
                    cesty = new Point[prvkuNaRozmer + 1];
                    temp.CopyTo(cesty, 0);
                    cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                    jesteProhledat = 1;//jeste se bude hledat dale v bufferu
                    //nastavim posledni kartu z bufferu jako aktualni a pak ji z bufferu odeberu
                    Point souradniceKarty = buffer[buffer.Length - 1];//posledni karta
                    Kamen tmp = kameny[souradniceKarty.X, souradniceKarty.Y];//vytahnu ji z pracovniho pole
                    aktualniKarta = tmp;
                    //odeberu ji z bufferu
                    Point[] tmpBuffer = buffer;
                    buffer = new Point[buffer.Length - 1];//zmensim o jednu
                    for (int i = 0; i < buffer.Length; i++)
                    {
                        buffer[i] = tmpBuffer[i];
                    }
                }
                break;
            case 1://----------bezny kamen----------

                Point[] tempor = cesty;
                //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                prvkuNaRozmer = tempor.Length;
                cesty = new Point[prvkuNaRozmer + 1];
                //maji stejne prvku jako ty nad ni, jenom nejsou vsechny vyuzity
                //----------zjistit aktualni cestu
                //prekopiruju zpatky
                tempor.CopyTo(cesty, 0);
                cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                Point dalsiKarta = aktualniKarta.JesteProhledat[0];
                aktualniKarta.JesteProhledat = new Point[0];// null;
                predchoziKarta = aktualniKarta;
                aktualniKarta = kameny[dalsiKarta.X, dalsiKarta.Y];
                break;
            case 2://----------krizovatka----------
            case 3:
                //prida se do bufferu pro opetovne prohledani, protoze ma na sobe cestu, ktera se neprohleda
                Point[] tmpbuffer = buffer;
                buffer = new Point[buffer.Length + 1];
                tmpbuffer.CopyTo(buffer, 0);
                buffer[buffer.Length - 1] = aktualniKarta.VPracovnimPoli;
                //vybrany kamen odeberu z prohledavani pro zpetny pruchod, vyhledam ho hned
                Point[] tmp2 = null;
                index = NahodneCislo(0, aktualniKarta.JesteProhledat.Length);
                for (int i = 0; i < aktualniKarta.JesteProhledat.Length; i++)
                {
                    //Console.WriteLine("Case 2 - for i");
                    if (aktualniKarta.JesteProhledat[index].X == aktualniKarta.JesteProhledat[i].X &&
aktualniKarta.JesteProhledat[index].Y == aktualniKarta.JesteProhledat[i].Y)
                    {
                        tmp2 = aktualniKarta.JesteProhledat;//docasne ulozim mimo
                        aktualniKarta.JesteProhledat = new Point[aktualniKarta.JesteProhledat.Length - 1];//zmensim
                        //nakopituju zpatky
                        //ten minuly kamen je na souradnicich "i"
                        int indexPole = 0;
                        for (int j = 0; j < tmp2.Length; j++)
                        {
                            if (j != i)
                            {
                                aktualniKarta.JesteProhledat[indexPole] = tmp2[j];
                                indexPole += 1;
                            }
```

```
                        }
                    }
                }
                //pridam souradnice vybrane karty do cesty
                Point[] tmp3 = cesty;
                prvkuNaRozmer = tmp3.Length;
                cesty = new Point[prvkuNaRozmer + 1];
                tmp3.CopyTo(cesty, 0);
                cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                //mozna prohodit indexy
                predchoziKarta = aktualniKarta;
                int iks = tmp2[index].X;
                int yps = tmp2[index].Y;
                aktualniKarta = kameny[iks, yps];//z pracovniho pole urcim jako aktualni kamen ten, ktery se vylosoval z jeste
nehledanych cest
                if (aktualniKarta.PropojeneKameny != null)
                {
                    pocetPropojeni = aktualniKarta.PropojeneKameny.Length;
                }
                else
                {
                    pocetPropojeni = 0;
                }
                jesteProhledat = pocetPropojeni;
                break;
            }
        }
        //kartam musim nastavit vlastnosti na hodnoty pred zapocenim hledanim, abych mohl stejne hledat i priste

        for (int h = 0; h < 7; h++)
        {

            for (int g = 0; g < 7; g++)
            {
                kameny[h, g].JesteProhledat = null;
            }

        }

        //------------Cesta nalezena uspesne, ted jeste rozpoznat hlavni----------
        //ze vznikle cesty, ktera obsahuje serazeny zapis kamenu tak, jak se prochazelo pri hledani, vytvorim jednu jedinou cestu, po
ktere pujde figurka
        //staci mi zjistene propojene vsechny mozne cesty, ktere lze nalezt, ty pak dale budu zpracovavat-najdu konce cest, a podle
nich budu hledat
        //nejkratsi cestu k artefaktu
        //z cesty musim najit jejich konce
        //definice konce: u beznych cest se jedna o kameny, kter maji pouze jednu spojenou cestu
        //u krizovatek jsou to kameny, ktere nemaji spojeny vsechny 3 cesty, tedy maji spojene 2 cesty
        //to muzu vyresit uz pri samotnem hledani cesty, nebo zde projit vsechny kameny na ceste a urcit je jako koncove body
        Point[] pracovniCesty = cesty;

        bool hotovo = false;
        Point[] docasna = new Point[0];

        //prochazim jednotlive kameny pracovni cesty a zjistuji, zda nejaky neodpovida cili
        for (int i = 0; i < pracovniCesty.Length; i++)
        {
            if (pracovniCesty[i] != cil)//souradnice nenalezena
            {
                hotovo = true;
            }
            else//nalezena
            {
                hotovo = false;
                //ulozim cestu souradnice az po prave nalezenou do docasne cesty
                //projdu puvodni cestu az po prave nalezenou sourad
                docasna = new Point[i + 1];

                for (int j = 0; j <= i; j++)
                {
                    docasna[j] = pracovniCesty[j];
                }

                //---------po tud funguje spravne----------
                //v docasne ceste najdu misto, kde se nachazi stejna souradnice. Kdyz nic nenajdu, tak je cesta hotova
                //zjistim, ktera souradnice se v poli nachazi 2x
```

```
                while (hotovo == false)
                {
                    int pozicePrvniho = -1, poziceDruheho = 0;
                    bool ukonceno = false;

                    for (int y = 0; y < docasna.Length; y++)
                    {
                        for (int z = y + 1; z < docasna.Length; z++)
                        {
                            if (docasna[y] == docasna[z])//nalezeno neco 2x
                            {
                                pozicePrvniho = y;
                                poziceDruheho = z;
                                ukonceno = true;
                                break;
                            }
                        }
                        if (ukonceno) { break; }
                    }
                    if (pozicePrvniho != -1)//nalezeno neco 2x
                    {
                        Point[] tmpDocasna = docasna;
                        Point[] uschovna = new Point[tmpDocasna.Length - poziceDruheho + pozicePrvniho];
                        int uschovnaIndex = 0;
                        //prekopiruju data az do prvniho nalezeneho
                        for (int k = 0; k <= pozicePrvniho; k++)
                        {
                            uschovna[uschovnaIndex] = tmpDocasna[k];
                            uschovnaIndex += 1;
                        }
                        //dokopiruju data, ktera jsou za druhym nalezenym
                        for (int l = poziceDruheho + 1; l < tmpDocasna.Length; l++)
                        {
                            uschovna[uschovnaIndex] = tmpDocasna[l];
                            uschovnaIndex += 1;
                        }
                        docasna = uschovna;
                    }
                    else//v ceste neni zadna souradnice 2x
                    {
                        hotovo = true;
                    }
                }
                break;
            }
        }
        if (docasna == null)
        {
            throw new Exception();//skoci na catch
        }
        else
        {
            Cesta = docasna;
            return true;
        }
    }
    catch
    {
        MessageBox.Show("Pri hledana cesty doslo k chybe");
        return false;
    }
}

private static Point NajdiCestu(Kamen[,] kameny)
{
    Point[] buffer = new Point[0];//buffer pro ukladani krizovatek
    //ulozim kamen, na kterem stoji aktualni hrac
    Kamen pocatecniKamen = aktualniKamen;//aktualniHra.AktualniHrac.Figurka.AktualniKamen;
    //cestu vytvorim prochazenim jednotlivych propojenych kamenu, ktere implementuje kazdy kamen. Kdyz zadny nema, ma
hodnoty null
    int pocetPropojeni;Point[] cesty = new Point[0];
    //Point[] propojeneKameny;
    Kamen predchoziKarta = null;//souradnice karty, ze ktere jsem dorazil na aktualni kartu-vyhledam  v pracovnim poli a mam
kamen
    Kamen aktualniKarta = pocatecniKamen;
```

```
            int jesteProhledat;//pocet cest k prohledani na dannem kamnu
            if (pocatecniKamen.PropojeneKameny != null)
            {
               jesteProhledat = 1;//pocetPropojeni;
            }
            else
            {
               jesteProhledat = 0;
            }
            //prochazim kamen po kameni a zapisuju do pole pro cestu
            int index;//slouzi pro vyber cesty z pole cest kamene
            int rozmery = cesty.Length;
            //----------------zde je nekde chyba, tuhne to-----------------
            while (jesteProhledat != 0)
            {
               if (aktualniKarta.JesteProhledat == null)//kdyz je prazdna, tak hledam poprve
               {
                  aktualniKarta.JesteProhledat = aktualniKarta.PropojeneKameny;//pro zacatek se zde ulozi vsechny kameny, se kterymi je
propojen
                  if (predchoziKarta != null)//to znamena, ze jdu z prvni karticky-vychozi pozice, jinak tato situace nemuze nastat
                  {
                     //postupne porovnavam minule souradnice se souradnicemi v poli JesteProhledat, a kdyz je najdu, tak je odstranim z pole
                     for (int i = 0; i < aktualniKarta.JesteProhledat.Length; i++)
                     {
                        if (predchoziKarta.VPracovnimPoli.X == aktualniKarta.JesteProhledat[i].X && predchoziKarta.VPracovnimPoli.Y ==
aktualniKarta.JesteProhledat[i].Y)
                        {
                           Point[] tmp = aktualniKarta.JesteProhledat;//docasne ulozim mimo
                           aktualniKarta.JesteProhledat = new Point[aktualniKarta.JesteProhledat.Length - 1];//zmensim
                           //nakopiruju zpatky
                           //ten minuly kamen je na souradnicich "i", takze zpatky nakopiruju vsechny krome teto
                           int indexPole = 0;
                           for (int j = 0; j < tmp.Length; j++)
                           {
                              if (j != i)
                              {
                                 aktualniKarta.JesteProhledat[indexPole] = tmp[j];
                                 indexPole += 1;
                              }
                           }
                        }
                     }
                  }
                  jesteProhledat = aktualniKarta.JesteProhledat.Length;
               }
               else//prohledavam kameny z bufferu
               {
                  jesteProhledat = aktualniKarta.JesteProhledat.Length;
               }
               //nasledne najdu tu, ze ktere jsem prisel a tu z pole odstranim
               int prvkuNaRozmer;
               switch (jesteProhledat)
               {
                  case 0://konec cesty
                     //prohledat buffer, kdyz je prazdny, tak ukoncit. Jinak zvolit jako aktualni kamen ten v bufferu
                     if (buffer.Length == 0)
                     {
                        Point[] temp = cesty;
                        //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                        prvkuNaRozmer = temp.Length;//temp.GetLength(1);
                        //rozmery + 1 pak upravit zpatky na rozmery
                        cesty = new Point[prvkuNaRozmer + 1];
                        temp.CopyTo(cesty, 0);
                        cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                        jesteProhledat = 0;
                     }
                     else
                     {
                        Point[] temp = cesty;
                        //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                        prvkuNaRozmer = temp.Length;//temp.GetLength(1);
                        //rozmery + 1 pak upravit zpatky na rozmery
                        cesty = new Point[prvkuNaRozmer + 1];
                        temp.CopyTo(cesty, 0);
                        cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                        jesteProhledat = 1;//jeste se bude hledat dale v bufferu
```

```
                    //nastavim posledni kartu z bufferu jako aktualni a pak ji z bufferu odeberu
                    Point souradniceKarty = buffer[buffer.Length - 1];//posledni karta
                    Kamen tmp = kameny[souradniceKarty.X, souradniceKarty.Y];//vytahnu ji z pracovniho pole
                    aktualniKarta = tmp;
                    //odeberu ji z bufferu
                    Point[] tmpBuffer = buffer;
                    buffer = new Point[buffer.Length - 1];//zmensim o jednu
                    for (int i = 0; i < buffer.Length; i++)
                    {
                        buffer[i] = tmpBuffer[i];
                    }

                }
                break;
            case 1://----------bezny kamen----------

                Point[] tempor = cesty;
                //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                prvkuNaRozmer = tempor.Length;
                cesty = new Point[prvkuNaRozmer + 1];
                //maji stejne prvku jako ty nad ni, jenom nejsou vsechny vyuzity
                //----------zjistit aktualni cestu
                //prekopiruju zpatky
                tempor.CopyTo(cesty, 0);
                cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                Point dalsiKarta = aktualniKarta.JesteProhledat[0];
                aktualniKarta.JesteProhledat = new Point[0];// null;
                predchoziKarta = aktualniKarta;
                aktualniKarta = kameny[dalsiKarta.X, dalsiKarta.Y];
                break;
            case 2://----------krizovatka----------
            case 3:
                //prida se do bufferu pro opetovne prohledani, protoze ma na sobe cestu, ktera se neprohleda
                Point[] tmpbuffer = buffer;
                buffer = new Point[buffer.Length + 1];
                tmpbuffer.CopyTo(buffer, 0);
                buffer[buffer.Length - 1] = aktualniKarta.VPracovnimPoli;
                //vybrany kamen odeberu z prohledavani pro zpetny pruchod, vyhledam ho hned
                Point[] tmp2 = null;
                index = NahodneCislo(0, aktualniKarta.JesteProhledat.Length);
                for (int i = 0; i < aktualniKarta.JesteProhledat.Length; i++)
                {
                    //Console.WriteLine("Case 2 - for i");
                    if (aktualniKarta.JesteProhledat[index].X == aktualniKarta.JesteProhledat[i].X &&
aktualniKarta.JesteProhledat[index].Y == aktualniKarta.JesteProhledat[i].Y)
                    {
                        tmp2 = aktualniKarta.JesteProhledat;//docasne ulozim mimo
                        aktualniKarta.JesteProhledat = new Point[aktualniKarta.JesteProhledat.Length - 1];//zmensim
                        //nakopituju zpatky
                        //ten minuly kamen je na souradnicich "i"
                        int indexPole = 0;
                        for (int j = 0; j < tmp2.Length; j++)
                        {
                            if (j != i)
                            {
                                aktualniKarta.JesteProhledat[indexPole] = tmp2[j];
                                indexPole += 1;
                            }
                        }
                    }
                }
                //pridam souradnice vybrane karty do cesty
                Point[] tmp3 = cesty;
                prvkuNaRozmer = tmp3.Length;
                cesty = new Point[prvkuNaRozmer + 1];
                tmp3.CopyTo(cesty, 0);
                cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                //mozna prohodit indexy
                predchoziKarta = aktualniKarta;
                int iks = tmp2[index].X;
                int yps = tmp2[index].Y;
                aktualniKarta = kameny[iks, yps];//z pracovniho pole urcim jako aktualni kamen ten, ktery se vylosoval z jeste
nehledanych cest
                if (aktualniKarta.PropojeneKameny != null)
                {
                    pocetPropojeni = aktualniKarta.PropojeneKameny.Length;
```

```
                    }
                    else
                    {
                        pocetPropojeni = 0;
                    }
                    jesteProhledat = pocetPropojeni;
                    break;
                }
            }
            //kartam musim nastavit vlastnosti na hodnoty pred zapocenim hledanim, abych mohl stejne hledat i priste

            for (int h = 0; h < 7; h++)
            {

                for (int g = 0; g < 7; g++)
                {
                    kameny[h, g].JesteProhledat = null;
                }
            }
            Point souradnice;
            if (cesty.Length != 0)
            {
                souradnice = cesty[cesty.Length - 1];
            }
            else
            {
                souradnice = aktualniKamen.VPracovnimPoli;
            }
            return souradnice;
        }

        private static void PosunFigurku(HraciForm KamKreslit, Kamen[,] kameny)
        {
            Kamen kamen1 = null;
            Figurka aktualniFigurka = aktualniHra.AktualniHrac.Figurka;
            int posunutiX, posunutiY;
            Point stred1;

            //posune figurku
            if (Cesta.Length > 1)//bude se posouvat figurka
            {
                aktualniHra.casovac.Stop();
            }
            for (int policko = 1; policko < Cesta.Length; policko++)
            {
                kamen1 = kameny[Cesta[policko].X, Cesta[policko].Y];
                //vypocitam u kazdeho kamene jeho stred, ktery bude slouzit jako pocatecni a koncovy bod
                stred1 = new Point(kamen1.Pozice.X + kamen1.Rozmery.Width / 2, kamen1.Pozice.Y + kamen1.Rozmery.Height /
2);//stred kamene
                stred1 = new Point(stred1.X - aktualniFigurka.Rozmery.Width / 2, stred1.Y - aktualniFigurka.Rozmery.Height / 2);
                posunutiX = (aktualniFigurka.Pozice.X - stred1.X) / 10;
                posunutiY = (aktualniFigurka.Pozice.Y - stred1.Y) / 10;
                for (int i = 0; i < 10; i++)
                {
                    aktualniFigurka.Pozice = new Point(aktualniFigurka.Pozice.X - posunutiX, aktualniFigurka.Pozice.Y - posunutiY);
                    KamKreslit.Refresh();
                }
                if (cilovyForm.dodatecnyPosun > 0)
                {
                    if (posunutiX > 0)//doprava
                    {
                        posunutiX = cilovyForm.dodatecnyPosun;
                    }
                    else if (posunutiX < 0)//doleva
                    {
                        posunutiX = cilovyForm.dodatecnyPosun * (-1);
                    }
                    else if (posunutiY > 0)//dolu
                    {
                        posunutiY = cilovyForm.dodatecnyPosun;
                    }
                    else if (posunutiY < 0)//nahory
                    {
                        posunutiY = cilovyForm.dodatecnyPosun * (-1);
                    }
                    aktualniFigurka.Pozice = new Point(aktualniFigurka.Pozice.X - posunutiX, aktualniFigurka.Pozice.Y - posunutiY);
```

```
                    cilovyForm.Refresh();
                }
            }

            //neaktivni figurka musi mit stejnou pozici jako ta aktivni
            aktualniHra.AktualniHrac.NeaktivniFigurka.Pozice = aktualniHra.AktualniHrac.Figurka.Pozice;
            //na konci posunuti urcim cilovy kamen jako ten aktualni, na kterem fugurka prave stoji
            if (kamen1 != null)//znamena to, ze cyklus neprobehl, tudiz nebyla nalezena zadna cesta, takze stoji na stejnem miste
            {
                aktualniFigurka.AktualniKamen = kamen1;
                aktualniHra.AktualniHrac.NeaktivniFigurka.AktualniKamen = aktualniHra.AktualniHrac.Figurka.AktualniKamen;
                //kontrolovat, zda se nenachazi v domecku


                //zkontroluje zda se nenachazim na hledanem artefaktu
                if (aktualniFigurka.AktualniKamen.Artefakt == aktualniHra.AktualniHrac.HledanaKarta.Artefakt)//nasel jsem artefakt
                {
                    //zkontroluju, zda se nejedna o posledni artefakt, ktery hrac hledal.
                    //kdyz ano, nastav konec hry
                    aktualniHra.KartaNalezena(aktualniHra.AktualniHrac.HledanaKarta, aktualniHra.PoziceNalezeneKarty);
                    if (aktualniHra.hrajeSe == false)
                    {
                        MessageBox.Show(aktualniHra.AktualniHrac.Nick + " vyhrál!");
                    }
                    else
                    {
                        aktualniHra.DalsiHrac();
                        //prehodit prvky formu a refreshnout
                        KamKreslit.PrvkyNaFormu = aktualniHra.prvkyNaFormu;
                        KamKreslit.Refresh();
                    }
                }
                else//jakmile hrac posunul figurku, hraje dalsi hrac
                {
                    aktualniHra.DalsiHrac();
                    KamKreslit.PrvkyNaFormu = aktualniHra.prvkyNaFormu;
                    KamKreslit.Refresh();
                }

                KamKreslit.hracPosunul = false;//hrac muze posunovat kamenem
            }
            else//cesta nebyla nalezena, stoji na stejnem miste
            {
                aktualniHra.DalsiHrac();
                KamKreslit.PrvkyNaFormu = aktualniHra.prvkyNaFormu;
                KamKreslit.Refresh();
                KamKreslit.hracPosunul = false;
            }
        }

        private static VariantaZasunuti UrciNejlepsiVariantu(VariantaZasunuti[] varianty)
        {
            Random nahodne = new Random();
            VariantaZasunuti varianta = varianty[nahodne.Next(0, varianty.Length)];
            return varianta;
        }

        public static void PosunZasun()
        {
            VariantaZasunuti varianta = FindBestPath(cilovyForm.predchoziSloupecRadek, cilovyForm.predchoziSmer,
cilovyForm.predchoziCislo, posunovaci, pracovniPole);// cilovyForm.Posunovaci, aktualniHra.Kameny);// cilovyForm.Posunovaci,
pracovniPole);//vrati souradnice kamene, kam se presune figurka
            PosunPoPlose(varianta.RadekSloupec, varianta.Cislo, varianta.Smer, cilovyForm);//posune posunovaci kamen po plose
            cilovyForm.Posunovaci.Uhel = varianta.Uhel;
            cilovyForm.Posunout(varianta.RadekSloupec, varianta.Smer, varianta.Cislo, cilovyForm.Posunovaci); //posune v pameti
            cilovyForm.predchoziCislo = varianta.Cislo;
            cilovyForm.predchoziSloupecRadek = varianta.RadekSloupec;
            cilovyForm.predchoziSmer = varianta.Smer;
            //-------pro pripad, ze by se vysunula z hraci plochy
            for (int hrac = 0; hrac < aktualniHra.Hraci.Length; hrac++)
            {
                if (aktualniHra.Hraci[hrac].Figurka.AktualniKamen == cilovyForm.Posunovaci)//figurka stoji na kameni, ktery je mimo hraci
plochu
                {
                    //Posunivaci si pamatuje souradnice v pracovnim poli. Pred posunutim na techto souradnicich stal, ale kdyz byl posunut
                    //tyto hodnoty se mu neprepsaly,tudiz jsou porad stejne, i kdyz uz na nich nestoji
```

```
                if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(1, 0))//na levem okraji na druhem radku
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[1, 6];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(3, 0))//na levem okraji na ctvrtem radku
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[3, 6];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(5, 0))//na levem okraji na sestem radku
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[5, 6];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(1, 6))//na pravem okraji na druhem radku
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[1, 0];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(3, 6))//na pravem okraji na ctvrtem radku
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[3, 0];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(5, 6))//na pravem okraji na sestem radku
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[5, 0];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(0, 1))//na hornim okraji na druhem sloupci
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[6, 1];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(0, 3))//na hornim okraji na ctvrtem sloupci
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[6, 3];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(0, 5))//na hornim okraji na sestem sloupci
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[6, 5];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(6, 1))//na dolnim okraji na druhem sloupci
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[0, 1];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(6, 3))//na dolnim okraji na ctvrtem sloupci
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[0, 3];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(6, 5))//na dolnim okraji na sestem sloupci
                {
                    aktualniHra.Hraci[hrac].Figurka.AktualniKamen = cilovyForm.PracovniPole[0, 5];
                }
            }
            if (aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen == cilovyForm.Posunovaci)//figurka stoji na kameni, ktery je
mimo hraci plochu
            {
                //Posunivaci si pamatuje souradnice v pracovnim poli. Pred posunutim na techto souradnicich stal, ale kdyz byl posunut
                //tyto hodnoty se mu neprepsaly,tudiz jsou porad stejne, i kdyz uz na nich nestoji
                if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(1, 0))//na levem okraji na druhem radku
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[1, 6];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(3, 0))//na levem okraji na ctvrtem radku
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[3, 6];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(5, 0))//na levem okraji na sestem radku
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[5, 6];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(1, 6))//na pravem okraji na druhem radku
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[1, 0];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(3, 6))//na pravem okraji na ctvrtem radku
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[3, 0];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(5, 6))//na pravem okraji na sestem radku
                {
```

```
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[5, 0];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(0, 1))//na hornim okraji na druhem sloupci
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[6, 1];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(0, 3))//na hornim okraji na ctvrtem sloupci
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[6, 3];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(0, 5))//na hornim okraji na sestem sloupci
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[6, 5];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(6, 1))//na dolnim okraji na druhem sloupci
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[0, 1];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(6, 3))//na dolnim okraji na ctvrtem sloupci
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[0, 3];
                }
                else if (cilovyForm.Posunovaci.VPracovnimPoli == new Point(6, 5))//na dolnim okraji na sestem sloupci
                {
                    aktualniHra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = cilovyForm.PracovniPole[0, 5];
                }
            }
        }
        //----------------//
        aktualniKamen = aktualniHra.AktualniHrac.Figurka.AktualniKamen;
        aktualniHra.Kameny = cilovyForm.PracovniPole;
        //----
        aktualniHra.AktualizujPropojeniCest();
        VytvorCestu(varianta.SouradCile, aktualniHra.Kameny);//tato metoda je pouzivana zaroven i pri hledani optimalni cesty
        //--
        //v pameti uz ma figurka nastaven aktualni kamen na opacnou stranu, tedy pokud stala na kraji.
        //v nasledujicim kodu ovsem hledam v poli, ve kterem je obsazen i kamen, ktery se vsunuje.
        //takze figurka ma aktualni kamen nastaven uz spravne, ale tady pracuju s jejimi fyzickymi souradnicemi, takze ji posunu s
kartickou, na ktere stoji
        //ovsem jenom jako, protoze ve skutecnosti menim jenom jeji souradnice.

        //u aktualniho hrace budu pracovat s aktivni figurkou, u ostatnich s neaktivni
        for (int i = 0; i < aktualniHra.Hraci.Length; i++)//projdu vsechny hrace
        {
            for (int j = 0; j < cilovyForm.poleKaret.Length; j++)//u kazdeho z nich prohledam vsechny kameny, ktere se prave budou
posouvat
            {
                if (aktualniHra.Hraci[i] == aktualniHra.AktualniHrac)//aktualni hrac
                {
                    if (aktualniHra.Hraci[i].Figurka.AktualniKamen == cilovyForm.poleKaret[j])//figurka stoji na kamenu, ktery se bude
posouvat
                    {
                        //ulozim ji do pole pro posunuti figurky
                        if (proPosunuti == null)
                        {
                            proPosunuti = new Figurka[1];
                            proPosunuti[0] = aktualniHra.Hraci[i].Figurka;
                        }
                        else
                        {
                            Figurka[] tmp = proPosunuti;
                            proPosunuti = new Figurka[proPosunuti.Length + 1];
                            tmp.CopyTo(proPosunuti, 0);
                            proPosunuti[proPosunuti.Length - 1] = aktualniHra.Hraci[i].Figurka;
                        }
                        break;
                    }
                }
                else//neaktivni hrac, takze priradim tu neaktivni figurku
                {
                    if (aktualniHra.Hraci[i].NeaktivniFigurka.AktualniKamen == cilovyForm.poleKaret[j])//figurka stoji na kamenu, ktery se
bude posouvat
                    {
                        //ulozim ji do pole pro posunuti figurky
                        if (proPosunuti == null)
                        {
```

```
                        proPosunuti = new Figurka[1];
                        proPosunuti[0] = aktualniHra.Hraci[i].NeaktivniFigurka;
                    }
                    else
                    {
                        Figurka[] tmp = proPosunuti;
                        proPosunuti = new Figurka[proPosunuti.Length + 1];
                        tmp.CopyTo(proPosunuti, 0);
                        proPosunuti[proPosunuti.Length - 1] = aktualniHra.Hraci[i].NeaktivniFigurka;
                    }
                    break;
                }
            }
        }
    }
    for (int i = 0; i < 10; i++)
    {
        cilovyForm.pohni(proPosunuti);
        cilovyForm.Refresh();
    }
    if (proPosunuti != null)
    {
        for (int b = 0; b < proPosunuti.Length; b++)
        {
            proPosunuti[b].Pozice = new Point((proPosunuti[b].AktualniKamen.Pozice.X +
proPosunuti[b].AktualniKamen.Rozmery.Width / 2) - proPosunuti[b].Rozmery.Width / 2, (proPosunuti[b].AktualniKamen.Pozice.Y +
proPosunuti[b].AktualniKamen.Rozmery.Height / 2) - proPosunuti[b].Rozmery.Height / 2);
            cilovyForm.Refresh();
            Thread.Sleep(10);
        }
    }
    proPosunuti = null;
    PosunFigurku(cilovyForm, aktualniHra.Kameny);
}

static void casovac_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
{
    if (pocetPosunuti < 10)
    {
        ThreadPool.QueueUserWorkItem(new WaitCallback(cilovyForm.ProcesVeVlakne4));
    }
    else
    {
        casovac.Stop();
        pocetPosunuti = 0;
        if (proPosunuti != null)
        {
            for (int b = 0; b < proPosunuti.Length; b++)
            {
                proPosunuti[b].Pozice = new Point((proPosunuti[b].AktualniKamen.Pozice.X +
proPosunuti[b].AktualniKamen.Rozmery.Width / 2) - proPosunuti[b].Rozmery.Width / 2, (proPosunuti[b].AktualniKamen.Pozice.Y +
proPosunuti[b].AktualniKamen.Rozmery.Height / 2) - proPosunuti[b].Rozmery.Height / 2);
                ThreadPool.QueueUserWorkItem(new WaitCallback(cilovyForm.RefreshFormu));
            }
        }
        proPosunuti = null;
        ThreadPool.QueueUserWorkItem(new WaitCallback(cilovyForm.PosunoutFigurkuVAI));
    }
}
public static void PosunoutKameny()
{
    pocetPosunuti += 1;
    cilovyForm.pohni(proPosunuti);
    cilovyForm.Refresh();
}
public static void PosunoutFigurku()
{
    PosunFigurku(cilovyForm, aktualniHra.Kameny);
}

}

class VariantaZasunuti
{
    private int radekSloupec, cislo, smer;
    private Point souradCile;
```

```
        private float uhel;

        public VariantaZasunuti(int RadekSloupec, int Cislo, int Smer, Point SouradCile,float Uhel)
        {
            this.radekSloupec = RadekSloupec;
            this.cislo = Cislo;
            this.smer = Smer;
            this.souradCile = SouradCile;
            this.uhel = Uhel;
        }

        public int RadekSloupec
        {
            get
            {
                return radekSloupec;
            }
        }
        public int Cislo
        {
            get
            {
                return cislo;
            }
        }
        public int Smer
        {
            get
            {
                return smer;
            }
        }
        public Point SouradCile
        {
            get
            {
                return souradCile;
            }
        }
        public float Uhel
        {
            get
            {
                return uhel;
            }
        }
    }
}
```

C2 FIGURKA

```
using System;
using System.Drawing;

namespace Labyrint
{
    public class Figurka:GrafickyObjekt
    {
        private Kamen aktualniKamen;
        public Figurka(Image Obrazek)
        {
            this.obrazek = Obrazek; //ulozeni obrazku do interni promenne
            pozice = new Point(0, 0);//defaultni pozice je v levem rohu formulare
            this.uhel = 0f;//uhel natoceni;
            rozmery = new Size(0, 0);
            rect = new Rectangle(pozice.X, pozice.Y, rozmery.Width, rozmery.Height);
            jmeno = "";
        }

        //---------vlastnosti----------
        //nastavuje nebo vraci kamen, na kterem se figurka nachazi
        public Kamen AktualniKamen
        {
            get
```

```
            {
                return aktualniKamen;
            }
            set
            {
                aktualniKamen = value;
            }
        }
    }
}
```

C3 GRAFICKY OBJEKT

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;

namespace Labyrint
{
    public delegate void KliknutiHandler(object volajici);//deklarace verejneho delegata, cili zastupce metody
    public delegate void PrejetiNadHandler(object volajici);//pro MouseOver
    public delegate void StisknutoHandler(object volajici);//pro MouseDown
    public delegate void UvolnenoHandler(object volajici);//pro MouseUp
    public delegate void OpusteniHandler(object volajici);//pro MouseLeave

    public class GrafickyObjekt
    {
        protected Image obrazek;
        protected Point pozice;
        protected float uhel;
        protected Size rozmery;
        protected Rectangle rect;
        protected String jmeno;
        public int zmenaUhlu = 0;


        //----------konstruktor--------------------
        public GrafickyObjekt(Image Obrazek)//konstruktor objektu
        {
            this.obrazek = new Bitmap(Obrazek); //ulozeni obrazku do interni promenne
            pozice = new Point(0, 0);//defaultni pozice je v levem rohu formulare
            this.uhel = 0f;//uhel natoceni;
            rozmery = new Size(0, 0);
            rect = new Rectangle(pozice.X, pozice.Y, rozmery.Width, rozmery.Height);
            jmeno = "";
        }
        public GrafickyObjekt()//kdyz v odvozene tride nevolam konstruktor predka, tak se automaticky zavola tento. Pote muzu v dedene
tride pouzit vlastni konstruktor
        {
            obrazek = null;
        }

        //----------vlastnosti----------
        public Image Obrazek
        {
            get
            {
                return obrazek;
            }
            set
            {
                this.obrazek = value;
            }
        }
        public Point Pozice
        {
            get
            {
                return pozice;
            }
            set
            {
                pozice = value;
                //rect = new Rectangle(pozice.X, pozice.Y, rozmery.Width, rozmery.Height);
```

```
    }
}
public virtual float Uhel
{
    get
    {
        return uhel;
    }
    set
    {
        uhel = value;
        switch ((int)uhel)
        {
            case 0:
                obrazek.RotateFlip(RotateFlipType.RotateNoneFlipNone);
                break;
            case 90:
                obrazek.RotateFlip(RotateFlipType.Rotate90FlipNone);
                break;
            case 180:
                obrazek.RotateFlip(RotateFlipType.Rotate180FlipNone);
                break;
            case 270:
                obrazek.RotateFlip(RotateFlipType.Rotate270FlipNone);
                break;
        }
        zmenaUhlu += 1;
    }
}
public virtual Size Rozmery
{
    get
    {
        return rozmery;
    }
    set
    {
        rozmery = value;
        //rect = new Rectangle(pozice.X, pozice.Y, rozmery.Width, rozmery.Height);
    }
}
public String Jmeno
{
    get
    {
        return jmeno;
    }
    set
    {
        jmeno = value;
    }
}

//----------metody----------------

//vykresleni na formular - obrazek na dannou pozici
public virtual void Vykresli(Graphics gr)
{
    if (this.obrazek != null)
    {
        gr.DrawImage(obrazek, pozice);
    }
}

//----------udalost----------
//deklarace udalosti, ktera musi byt stejneho formatu jako delegat
public event KliknutiHandler Kliknuto;
public event PrejetiNadHandler PrejetoMysi;
public event StisknutoHandler StisknutoMysi;
public event UvolnenoHandler UvolnenoMysi;
public event OpusteniHandler OpusteniMysi;
//tato metoda spusti udalost
public void SpoustecUdalosti(object volajici)
{
    if (Kliknuto != null)
    {
```

```
            Kliknuto(volajici);
        }
    }
    public void SpoustecUdalostiPrejetoMysi(object volajici)
    {
        if (PrejetoMysi != null)
        {
            PrejetoMysi(volajici);
        }
    }
    public void SpoustecUdalostiStisknutoMysi(object volajici)
    {
        if(StisknutoMysi!=null)
        {
            StisknutoMysi(volajici);
        }
    }
    public void SpoustecUdalostiUvolnenoMysi(object volajici)
    {
        if (UvolnenoMysi != null)
        {
            UvolnenoMysi(volajici);
        }
    }
    public void SpoustecUdalostiOpusteniMysi(object volajici)
    {
        if (OpusteniMysi != null)
        {
            OpusteniMysi(volajici);
        }
    }
  }
}
```

C4 HRA

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.Timers;
using System.Threading;

namespace Labyrint
{
  class Hra
  {
    private int hracu;
    private int pocetTahu;//urcuje celkovy pocet tahu ve hre
    private TypHry typHry;//1=single>>hrac proti pc; 2=multi>>vice hracu; 3=lan>>vice hracu po siti
    private Hrac[] hraci;
    private Hrac aktualniHrac;
    private HraciKarta[] karty;
    private HraciKarta hledanaKarta;
    private Figurka[] figurky;
    private Kamen[,] kameny;//pracovni pole
    private int casNaTah;//v sekundach
    public GrafickyObjekt[] prvkyNaFormu;
    private Random nahodneCislo;
    private Point poziceNalezeneKarty;
    public bool hrajeSe = true;//znaci, ze se jeste hraje a hra nebyla ukoncena
    private HraciForm formular;
    public System.Timers.Timer casovac;

    //Kameny slouzi jako pracovnipole
    public Hra(HraciKarta[] Karty, TypHry TypHry, Kamen[,] Kameny, GrafickyObjekt[] PrvkyNaFormu,HraciForm Formular)
    {
        if (Karty == null || Kameny == null || PrvkyNaFormu == null)
        {
        }
        else
        {
            hraci = new Hrac[0];
            typHry = TypHry;//default
            karty = Karty;
            kameny = Kameny;
```

```
        prvkyNaFormu = PrvkyNaFormu;
        pocetTahu = 0;
        casNaTah = 120;
        nahodneCislo = new Random();
        formular = Formular;
        casovac = new System.Timers.Timer();
    }
}

//--------vlastnosti-------
//vrati hrace, kteri jsou ve hre
public Hrac[] Hraci
{
    get
    {
        return hraci;
    }
}
public TypHry HerniTyp
{
    get
    {
        return typHry;
    }
    set
    {
        typHry = value;
    }
}
public enum TypHry
{
    single = 1,
    multi = 2,
    lan = 3
}
public Point[] Cesta;
public Kamen[,] Kameny
{
    get
    {
        return kameny;
    }
    set
    {
        if (value != null)
        {
            kameny = value;
        }
    }
}
public Hrac AktualniHrac
{
    get
    {
        return aktualniHrac;
    }
    set
    {
        aktualniHrac = value;
    }
}
public Point PoziceNalezeneKarty
{
    set
    {
        poziceNalezeneKarty = value;
    }
    get
    {
        return poziceNalezeneKarty;
    }
}

//---------metody-----------
//----------prida hrace do hry----------
```

```
        public void PridejHrace(string[] Nicks, Image[] Figurka,bool[] AI ,Size Velikost)//podle poctu prezdivek vypocita pocet hracu a
prida je do hry
        {
            hracu = Nicks.Length;//pocet hracu
            //prochazim jednotlive hrace
            while (hraci.Length < hracu)
            {
                //soucasne hrace ulozim mimo
                Hrac[] tmp = new Hrac[hraci.Length];
                hraci.CopyTo(tmp, 0);
                //pridam hrace
                hraci = new Hrac[hraci.Length + 1];
                tmp.CopyTo(hraci, 0);//hrace z docasneho uloziste prekopiruju zpet
                hraci[hraci.Length - 1] = new Hrac();//na posledni pozici pridam noveho hrace
                //z prezdivek, ktere jsem ziskal v parametru metody, vyberu tu, ktera odpovida poradi hrace
                hraci[hraci.Length - 1].Nick = Nicks[hraci.Length - 1];
                hraci[hraci.Length - 1].Ai = AI[hraci.Length - 1];
            }
            //vsichni hraci jsou pridani do hry
            //kazdemu hraci rozdam artefakty, ktere bude hledat
            RozdejKarty(hracu, karty);

            //kazdemu hraci nastavim jeho hledanou kartu na prvni v balicku vsech karet
            //projdu postupne vsechny hrace
            for (int i = 0; i < hracu; i++)
            {
                //jako hledanou kartu mu nastavim tu, ktera je prvni v jeho balicku hracich karet-artefaktu
                this.hraci[i].HledanaKarta = this.hraci[i].HraciKarty[0];

                //do hledanych karet zkopiruju vsechny, protoze na zacatku jsou hledany opravdu vsechny
                this.hraci[i].Hledane = new HraciKarta[this.hraci[i].HraciKarty.Length];
                this.hraci[i].HraciKarty.CopyTo(this.hraci[i].Hledane, 0);

                //z pocatku zadna karta nemohla byt nalezene, proto vytvorim prazdne pole
                this.hraci[i].Nalezene = new HraciKarta[0];
            }
            //hraci maji nastaveny hledane karty a take pole pro ukladani nalezenych karet

            //vytvori pole pro nove figurky - podle poctu hracu
            figurky = new Figurka[hracu];//kazdy hrac ma svou figurku

            //postupne projdu vsechny hrace
            //u kazdeho hrace vytvori novou figurku
            //podle barvy figurky urci, na jakem kameni bude figurka stat
            for (int i = 0; i < hracu; i++)
            {
                //vytvorim figurku a priradim ji obrazek, ktery jsem ziskal jako parametr metody
                //poradi obrazku odpovida poradi hracu
                figurky[i] = new Figurka(Figurka[i]);

                //nastavim ji jmeno a velikost pro vykresleni, kterou jsme ziskal na zaklade rozliseni
                figurky[i].Jmeno = "Figurka" + (i + 1).ToString();//Figurka1, Figurka2....
                figurky[i].Rozmery = Velikost;

                //na zaklade barvy figurky rozhodnu, kde bude umistena na hraci plose
                //urcim jeji kamen, na kterem se bude nachazet
                //ke kazdemu hraci priradim tlacitko a neaktivni figurku
                if (figurky[i].Obrazek == Zdroj.Postavicka_cervena)
                {
                    figurky[i].AktualniKamen = kameny[0, 0];

                    Zdroj.TlacitkoCerveny = new Bitmap((Image)Zdroj.TlacitkoCerveny, formular.umisteni.DalsiHracRozmery);
                    hraci[i].Tlacitko =new GrafickyObjekt((Image)Zdroj.TlacitkoCerveny);
                    hraci[i].Tlacitko.Pozice = formular.umisteni.DalsiHrac;
                    hraci[i].Tlacitko.Kliknuto+= new KliknutiHandler(formular.dalsiHrac_Kliknuto);
                    hraci[i].Tlacitko.Jmeno = "dalsiHrac";
                    hraci[i].Tlacitko.Rozmery = new Size(30, 30);

                    Zdroj.PostavickaCervenaNeaktivni=new
Bitmap((Image)Zdroj.PostavickaCervenaNeaktivni,formular.umisteni.velikostFigurky);
                    hraci[i].NeaktivniFigurka = new Figurka((Image)Zdroj.PostavickaCervenaNeaktivni);

                }
                else if (figurky[i].Obrazek == Zdroj.Postavicka_zluta)
                {
                    figurky[i].AktualniKamen = kameny[0, 6];
```

```
                Zdroj.TlacitkoZluty = new Bitmap((Image)Zdroj.TlacitkoZluty, formular.umisteni.DalsiHracRozmery);
                hraci[i].Tlacitko = new GrafickyObjekt((Image)Zdroj.TlacitkoZluty);
                hraci[i].Tlacitko.Pozice = formular.umisteni.DalsiHrac;
                hraci[i].Tlacitko.Jmeno = "dalsiHrac";
                hraci[i].Tlacitko.Kliknuto += new KliknutiHandler(formular.dalsiHrac_Kliknuto);
                hraci[i].Tlacitko.Rozmery = new Size(30, 30);

                Zdroj.PostavickaZlutaNeaktivni = new Bitmap((Image)Zdroj.PostavickaZlutaNeaktivni,
formular.umisteni.velikostFigurky);
                hraci[i].NeaktivniFigurka = new Figurka((Image)Zdroj.PostavickaZlutaNeaktivni);
            }
            else if (figurky[i].Obrazek == Zdroj.Postavicka_zelena)
            {
                figurky[i].AktualniKamen = kameny[6, 0];
                Zdroj.TlacitkoZeleny = new Bitmap((Image)Zdroj.TlacitkoZeleny, formular.umisteni.DalsiHracRozmery);
                hraci[i].Tlacitko = new GrafickyObjekt((Image)Zdroj.TlacitkoZeleny);
                hraci[i].Tlacitko.Pozice = formular.umisteni.DalsiHrac;
                hraci[i].Tlacitko.Jmeno = "dalsiHrac";
                hraci[i].Tlacitko.Kliknuto += new KliknutiHandler(formular.dalsiHrac_Kliknuto);
                hraci[i].Tlacitko.Rozmery = new Size(30, 30);

                Zdroj.PostavickaZelenaNeaktivni = new Bitmap((Image)Zdroj.PostavickaZelenaNeaktivni,
formular.umisteni.velikostFigurky);
                hraci[i].NeaktivniFigurka = new Figurka((Image)Zdroj.PostavickaZelenaNeaktivni);
            }
            else if (figurky[i].Obrazek == Zdroj.Postavicka_modra)
            {
                figurky[i].AktualniKamen = kameny[6, 6];
                Zdroj.TlacitkoModry = new Bitmap((Image)Zdroj.TlacitkoModry, formular.umisteni.DalsiHracRozmery);
                hraci[i].Tlacitko = new GrafickyObjekt((Image)Zdroj.TlacitkoModry);
                hraci[i].Tlacitko.Jmeno = "dalsiHrac";
                hraci[i].Tlacitko.Kliknuto += new KliknutiHandler(formular.dalsiHrac_Kliknuto);
                hraci[i].Tlacitko.Pozice = formular.umisteni.DalsiHrac;
                hraci[i].Tlacitko.Rozmery = new Size(30, 30);

                Zdroj.PostavickaModraNeaktivni = new Bitmap((Image)Zdroj.PostavickaModraNeaktivni,
formular.umisteni.velikostFigurky);
                hraci[i].NeaktivniFigurka = new Figurka((Image)Zdroj.PostavickaModraNeaktivni);
            }
            hraci[i].NeaktivniFigurka.AktualniKamen = figurky[i].AktualniKamen;
            hraci[i].NeaktivniFigurka.Jmeno = figurky[i].Jmeno;
            hraci[i].NeaktivniFigurka.Rozmery = Velikost;

            hraci[i].Figurka = figurky[i];
            //na posledni pozici v poli hledanych artefaktu pridam obrazek posledni hledane pozice, a to je "chlivek"

            HraciKarta[] tempor = new HraciKarta[this.hraci[i].Hledane.Length];
            this.hraci[i].Hledane.CopyTo(tempor, 0);
            this.hraci[i].Hledane = new HraciKarta[this.hraci[i].Hledane.Length + 1];
            tempor.CopyTo(this.hraci[i].Hledane, 0);
            //na zaklade barvy figurky priradim spravny obrazel
            if (hraci[i].Figurka.Obrazek == Zdroj.Postavicka_cervena)
            {
                this.hraci[i].Hledane[this.hraci[i].Hledane.Length - 1] = formular.hraciKartaCervena;
            }
            else if (hraci[i].Figurka.Obrazek == Zdroj.Postavicka_modra)
            {
                this.hraci[i].Hledane[this.hraci[i].Hledane.Length - 1] = formular.hraciKartaModra;
            }
            else if (hraci[i].Figurka.Obrazek == Zdroj.Postavicka_zelena)
            {
                this.hraci[i].Hledane[this.hraci[i].Hledane.Length - 1] = formular.hraciKartaZelena;
            }
            else if (hraci[i].Figurka.Obrazek == Zdroj.Postavicka_zluta)
            {
                this.hraci[i].Hledane[this.hraci[i].Hledane.Length - 1] = formular.hraciKartaZluta;
            }

            //podle kamene, na kterem se nachazi, urcim souradnice
            Kamen aktKamen = figurky[i].AktualniKamen;

            //jeji souradnice pro vykresleni urcim jako pozici kamene na kterem stoji + sirku/2=x souradnice, y se dela obdobne
            figurky[i].Pozice = new Point((aktKamen.Pozice.X + aktKamen.Rozmery.Width / 2) - figurky[i].Rozmery.Width / 2,
(aktKamen.Pozice.Y + aktKamen.Rozmery.Height / 2) - figurky[i].Rozmery.Height / 2);
            //hotovou figurku priradim hraci
            hraci[i].NeaktivniFigurka.Pozice = figurky[i].Pozice;
```

```
            figurky[i].Obrazek = (Image)new Bitmap(Figurka[i], Velikost);

        }
        //hraci maji vytvoreny a prirazeny figurky

        //rozlosuju poradi hracu pro hru
        RozlosujPoradi(hraci);

        //aktualnimu hraci nastavim aktivni figurku a vsem ostatnim tu neaktivni
        for (int i = 0; i < hraci.Length; i++)
        {
            //vlozim do pole pro vykresleni
            GrafickyObjekt[] tmp = new GrafickyObjekt[prvkyNaFormu.Length];
            prvkyNaFormu.CopyTo(tmp, 0);
            prvkyNaFormu = new GrafickyObjekt[prvkyNaFormu.Length + 1];
            tmp.CopyTo(prvkyNaFormu, 0);
            if (hraci[i] == aktualniHrac)
            {
                prvkyNaFormu[prvkyNaFormu.Length - 1] = (GrafickyObjekt)aktualniHrac.Figurka;
            }
            else
            {
                prvkyNaFormu[prvkyNaFormu.Length - 1] = (GrafickyObjekt)hraci[i].NeaktivniFigurka;
            }
        }


        //uz znam aktualniho hrace, takze mu nastavim tlacitko k vykresleni
        //nejdrive ho ale najdu v poli pro vykreslovani
        for (int i = 0; i < prvkyNaFormu.Length; i++)
        {
            if (prvkyNaFormu[i] != null)
            {
                if (prvkyNaFormu[i].Jmeno != null)
                {
                    if (prvkyNaFormu[i].Jmeno == "dalsiHrac")//nasel jsem ho
                    {
                        prvkyNaFormu[i] = aktualniHrac.Tlacitko;
                        break;
                    }
                }
            }
        }

        //

        //v poli pro vykreslovani hned za figurkama nasleduji artefakty aktualniho hrace-hledany a posledni nalezeny
        //proto do tohoto pole pridam dve nova mista
        GrafickyObjekt[] temp = new GrafickyObjekt[prvkyNaFormu.Length];
        prvkyNaFormu.CopyTo(temp, 0);
        prvkyNaFormu = new GrafickyObjekt[prvkyNaFormu.Length + 2];
        temp.CopyTo(prvkyNaFormu, 0);
        //predposledni pozice zustane vyhrazena pro pozdejsi zapis nalezene karty
        //na posledni zapisu hledanou kartu aktualni hrace, ktery byl vylosovan vyse
        if (aktualniHrac.Ai)
        {
            prvkyNaFormu[prvkyNaFormu.Length - 1] = formular.otaznik;
        }
        else
        {
            prvkyNaFormu[prvkyNaFormu.Length - 1] = (GrafickyObjekt)this.aktualniHrac.HledanaKarta;
        }
    }
    //rozda hraci karty hracum
    private void RozdejKarty(int Hracu, HraciKarta[] HraciKarty)
    {
        //zamichani hracich karet-artefaktu
        int pocetKaret = 24;//celkovy pocet karet ve hre
        int hracu = Hracu;//pocet hracu ve hre
        int karetNaHrace = pocetKaret / hracu;//pocet karet na jednoho hrace
        HraciKarta[] hraciKarty = new HraciKarta[HraciKarty.Length];
        Random fromTo = new Random();//nahodna cisla pro vyber karet, ktere se budou prohazovat
        HraciKarta[] tmp = new HraciKarta[1];
        HraciKarty.CopyTo(hraciKarty, 0);//prekopiruju do soukromeho pole karet pro dalsi praci s nimi
        for (int i = 0; i < 100; i++)//zamicha se celkem 100x
```

```
            {
                int z = fromTo.Next(0, hraciKarty.Length - 1);//odkud budu brat prvni kartu
                tmp[0] = hraciKarty[z];//prvni vylosovana karta se ulozi do docasneho uloziste
                int copyDo = fromTo.Next(0, hraciKarty.Length - 1);//misto, kam se nakopiruje prvni vybrana karta
                hraciKarty[z] = hraciKarty[copyDo];//na prvni vybrane misto se nakopiruje druhe vybrane
                hraciKarty[copyDo] = tmp[0];//na druhe vybrane misto se nakopiruje prvni vybrane, cimz se dokonci prohozeni
            }
            int x = 0;
            //zamichane karty priradim jednotlivym hracum
            for (int i = 0; i < hracu; i++)//prochazim postupne hrace
            {
                int velikostKaret = this.hraci[i].HraciKarty.Length;
                for (int j = 0; j < karetNaHrace; j++)//prochazim postupne zamichane karty
                {
                    HraciKarta[] temp = new HraciKarta[velikostKaret];
                    this.hraci[i].HraciKarty.CopyTo(temp, 0);//puvodni prekopiruju do docasneho
                    this.hraci[i].HraciKarty = new HraciKarta[velikostKaret + 1];
                    temp.CopyTo(this.hraci[i].HraciKarty, 0);//z docasneho prekopiruju nazpet do jiz zvetseneho pole
                    velikostKaret = this.hraci[i].HraciKarty.Length;
                    this.hraci[i].HraciKarty[velikostKaret - 1] = hraciKarty[x];
                    x += 1;//pocita celkove pruchody
                }
            }
        }
        //zavolam ji tehdy, kdyz se karta nalezne, ona zaridi upravu poli s kartama u sebe a u hrace
        public void KartaNalezena(HraciKarta NalezenaKarta, Point poziceNalezene)//poziceNalezene udava pozici pro vykresleni na
formu
        {
            try
            {
                //zjistim, kde v poli hracu se hrac nachazi a pak upravim hledane a nalezene karty
                for (int i = 0; i < this.hracu; i++)
                {
                    if (this.hraci[i] == this.aktualniHrac)//nasel jsem ho
                    {
                        if (this.hraci[i].Hledane.Length == 1)//hrac hledal posledni kartu, a kdyz byla spustena tato cast kodu, znamena to, ze uz
ji nasel
                        {
                            this.hrajeSe = false;
                            return;
                        }
                        else//hrac nasel kartu, ale neni posledni
                        {
                            //upravim pole hledanych
                            //aktualne hledana karta je na 0, pote je tam presunuta nasledujici karta
                            HraciKarta[] tmp = new HraciKarta[this.hraci[i].Hledane.Length];//docasne pole o stejne velikosti jako hledane
                            tmp = this.hraci[i].Hledane;
                            this.hraci[i].Hledane = new HraciKarta[this.hraci[i].Hledane.Length - 1];//pole hledanych zmensim, protoze se jedna
karta jiz nasla
                            //z docasneho data prekopiruju zpet do puvodniho, ale zmenseneho. Zacinam kopirovat
                            Array.Copy(tmp, 1, this.hraci[i].Hledane, 0, this.hraci[i].Hledane.Length);
                            this.hraci[i].HledanaKarta = this.hraci[i].Hledane[0];
                            //od jednicky, protoze na nule je prave nalezene karta, kterou nepotrebujeme

                            //upravim pole nalezenych
                            //nalezenou pridam na konec
                            if (this.hraci[i].Nalezene.Length == 0)//nalezena prvni karta
                            {
                                this.hraci[i].Nalezene = new HraciKarta[1];
                                this.hraci[i].Nalezene[0] = NalezenaKarta;
                                this.hraci[i].Nalezene[0].Pozice = poziceNalezene;
                            }
                            else//uz jich je nalezeno vice
                            {
                                tmp = this.hraci[i].Nalezene;
                                this.hraci[i].Nalezene = new HraciKarta[this.hraci[i].Nalezene.Length + 1];//nalezenych je o jednu vice
                                tmp.CopyTo(this.hraci[i].Nalezene, 0);//prekopiruju zpet
                                this.hraci[i].Nalezene[this.hraci[i].Nalezene.Length - 1] = NalezenaKarta;//jako posledni pridam prave nalezenou
kartu
                                this.hraci[i].Nalezene[this.hraci[i].Nalezene.Length - 1].Pozice = poziceNalezene;
                            }

                            return;
                        }
                    }
                }
```

```
            }
            catch
            {
               MessageBox.Show("v metode KartaNalezena() nastala chyba");
            }
         }
         //rozlosuje poradi hracu
         private void RozlosujPoradi(Hrac[] Hraci)
         {
            //urcim prvniho hrace, dale se hraje podle hodinovych rucicek
            //tzn., ze hrace v poli seradim ve spravnem poradi
            int pocethracu = Hraci.Length;
            Random nahodneCislo = new Random();
            //vyberu hrace, ktery bude zacinat
            int cisloHrace = nahodneCislo.Next(0, pocethracu);

            this.aktualniHrac = Hraci[cisloHrace];

            //hrace musim preskupit tak, aby odpovidali dannemu poradi
            Hrac[] tmp = new Hrac[Hraci.Length];
            int docasnyIndex = 0;
            tmp[docasnyIndex] = aktualniHrac;//na 1. miste je vybrany hrac

            Point[] sourad = new Point[4] { new Point(0, 0), new Point(0, 6), new Point(6, 6), new Point(6, 0) };
            int index = 0;
            for (int j = 0; j < sourad.Length; j++)//prochazim souradnice
            {
               if (this.aktualniHrac.Figurka.AktualniKamen.VPracovnimPoli.X == sourad[j].X &&
this.aktualniHrac.Figurka.AktualniKamen.VPracovnimPoli.Y == sourad[j].Y)//ma stejne souradnice jako v poli
               {
                  index = j;//index v poli
               }
            }
            //projde zbyle souradnice a prohleda hrace, jestli nektere ze souradnic nemaji. Kdyz jo, tak je spravne seradi
            for (int i = 0; i < sourad.Length - 1; i++)//staci 3 cykly
            {
               index += 1;//posunu se v poli dale
               index = index % sourad.Length;//skutecny index
               //najdu hrace, jehoz figurka stoji na dannych souradnicich
               for (int j = 0; j < pocethracu; j++)
               {
                  if (Hraci[j].Figurka.AktualniKamen.VPracovnimPoli.X == sourad[index].X &&
Hraci[j].Figurka.AktualniKamen.VPracovnimPoli.Y == sourad[index].Y)//ve hre je hrac, ktery ma na dannych souradnicich umistenou
figurku
                  {
                     docasnyIndex += 1;
                     tmp[docasnyIndex] = Hraci[j];
                  }
               }
            }
            tmp.CopyTo(Hraci, 0);
         }
         //pohyb figurkou
         //prehodi aktualniho hrace a zobrazi ho
         public void DalsiHrac()
         {
            //aktualniho hrace najdu v poli hracu a pak se posunu o jednoho hrace dale, ten se stane aktualnim

            //neaktivni figurka musi mit stejnou pozici jako ta aktivni
            aktualniHrac.NeaktivniFigurka.Pozice = aktualniHrac.Figurka.Pozice;
            aktualniHrac.NeaktivniFigurka.AktualniKamen = aktualniHrac.Figurka.AktualniKamen;

            int indexHrace = 0;
            for (int i = 0; i < hraci.Length; i++)
            {
               if (this.aktualniHrac == hraci[i])//nasel jsem aktualniho hrace
               {
                  indexHrace = i;
                  break;
               }
            }
            indexHrace += 1;//posunu se na dalsiho hrace
            indexHrace = indexHrace % hraci.Length;//skutecny index, kdyby po posunuti presahoval pole, presune se zase na zacatek
            this.aktualniHrac = hraci[indexHrace];

            aktualniHrac.Figurka.Pozice = aktualniHrac.NeaktivniFigurka.Pozice;
```

```
            aktualniHrac.Figurka.AktualniKamen = aktualniHrac.NeaktivniFigurka.AktualniKamen;

            //zobrazim hledany artefakt
            //pole pro vykreslovani prvku na formu obsahuje 1 hraci plochu + 50 kamenu + pocet hracu*figurky + 1 hledany artefakt + 1
nalezeny artefakt + cas=54+hracu
            //vsechny prvky jsou v poli jiz obsazeny-nalezena karticka je predposledni a hledana je posledni
            if (aktualniHrac.Nalezene.Length != 0)//uz byla nejaka karta nalezena
            {
               prvkyNaFormu[prvkyNaFormu.Length - 2] = aktualniHrac.Nalezene[aktualniHrac.Nalezene.Length - 1];//jako nalezenou
budu prezentovat tu posledni nalezenou
            }
            else//v pripade, ze nejaka karta byla nalezena predchazejicim hracem, tak v poli pro vykresleni nalezenych je artefakt.
            //Jenze tento hrac jeste nic nenasel, a tudiz musim nastavit v poli null, aby se nevykresloval artefakt predesleho hrace
            {
               prvkyNaFormu[prvkyNaFormu.Length - 2] = null;
            }
            if (aktualniHrac.Ai)
            {
               //nactu otaznik
               prvkyNaFormu[prvkyNaFormu.Length - 1] = formular.otaznik;
            }
            else
            {
               prvkyNaFormu[prvkyNaFormu.Length - 1] = aktualniHrac.HledanaKarta;
            }

            //nastaveni hodnot pro info-----nejdrive nastavit souradnice, protoze pri nastaveni obsahu se automaticky refreshne
            formular.hracTitulek.TextovyObsah = this.aktualniHrac.Nick;
            formular.nalezenoKolik.TextovyObsah = this.aktualniHrac.Nalezene.Length.ToString() + " z " +
this.aktualniHrac.HraciKarty.Length.ToString();
            formular.pocetTahu.TextovyObsah = aktualniHrac.PocetTahu.ToString();
            formular.prohledaniNalezenych.TextovyObsah = this.aktualniHrac.Nalezene.Length.ToString() + " z " +
this.aktualniHrac.Nalezene.Length.ToString();

            formular.umisteni.Vycentruj();

            //nastavim tlacitko dle aktualniho hrace
            //nejdrive ho najdu v poli pro vykreslovani
            for (int i = 0; i < formular.PrvkyNaFormu.Length; i++)
            {
               if (formular.PrvkyNaFormu[i] != null)
               {
                  if (formular.PrvkyNaFormu[i].Jmeno != null)
                  {
                     if (formular.PrvkyNaFormu[i].Jmeno == "dalsiHrac")//nasel jsem ho
                     {
                        formular.PrvkyNaFormu[i] = aktualniHrac.Tlacitko; //.Obrazek = (Image)new Bitmap(aktualniHrac.Tlacitko.Obrazek,
formular.umisteni.DalsiHracRozmery);
                        break;
                     }
                  }
               }
            }

            //aktualnimu hraci nastavim aktivni figurku, ostatnim neaktivni
            for(int i=0;i<formular.PrvkyNaFormu.Length;i++)
            {
               if (formular.PrvkyNaFormu[i] != null)
               {
                  if (formular.PrvkyNaFormu[i].Jmeno != null)
                  {
                     //aktivni i neaktivni figurky maji stejne jmeno, vyuziju toho
                     if (formular.PrvkyNaFormu[i] is Figurka)//jedna se o nejakou figurku
                     {
                        if (formular.PrvkyNaFormu[i].Jmeno == aktualniHrac.NeaktivniFigurka.Jmeno)//aktualniho hrace figurka
                        {
                           formular.PrvkyNaFormu[i] = aktualniHrac.Figurka;
                        }
                        else
                        {
                           //zjistim jmeno figurky v poli pro vykreslovani a porovnam ho s neaktivni figurkou. Kdyz se shoduje, prohodim je
                           for (int j = 0; j < hraci.Length; j++)
                           {
                              if (hraci[j].NeaktivniFigurka.Jmeno == formular.PrvkyNaFormu[i].Jmeno)
                              {
                                 formular.PrvkyNaFormu[i] = hraci[j].NeaktivniFigurka;
```

```
                        break;
                    }
                }
            }
        }
    }
    casNaTah = 120;
    int minuty, vteriny;
    minuty = (int)(casNaTah / 60);
    vteriny = casNaTah - minuty * 60;
    string text = minuty.ToString() + ":" + ((vteriny == 0) ? "0" + vteriny.ToString() : vteriny.ToString());
    formular.cas.TextovyObsah = text;
    casovac.Start();
    formular.Refresh();
    //zmenit tak, aby se nevytvorila zacyklena smycka jako ted
    if (aktualniHrac.Ai)//hraje pocitac
    {
        NaTahuJePocitac();
    }

}
//posune figurku na misto, kam hrac kliknul mysi
public void PosunFigurku(HraciForm KamKreslit)
{
    try
    {
        if (KamKreslit.hracPosunul)//hrac uz zasunul kartu
        {
            HraciForm kamKreslit = (HraciForm)KamKreslit;
            Kamen kamen1 = null;
            Figurka aktualniFigurka = aktualniHrac.Figurka;
            int posunutiX, posunutiY;
            Point stred1;

            //posune figurku
            if (this.Cesta.Length > 1)//bude se posouvat figurka
            {
                casovac.Stop();//zastavi se odpocitavani casu na tah
            }
            for (int policko = 1; policko < this.Cesta.Length; policko++)
            {
                kamen1 = kameny[this.Cesta[policko].X, this.Cesta[policko].Y];
                //vypocitam u kazdeho kamene jeho stred, ktery bude slouzit jako pocatecni a koncovy bod
                stred1 = new Point(kamen1.Pozice.X + kamen1.Rozmery.Width / 2, kamen1.Pozice.Y + kamen1.Rozmery.Height /
2);//stred kamene
                stred1 = new Point(stred1.X - aktualniFigurka.Rozmery.Width / 2, stred1.Y - aktualniFigurka.Rozmery.Height / 2);
                posunutiX = (aktualniFigurka.Pozice.X - stred1.X) / 10;
                posunutiY = (aktualniFigurka.Pozice.Y - stred1.Y) / 10;

                for (int i = 0; i < 10; i++)
                {
                    aktualniFigurka.Pozice = new Point(aktualniFigurka.Pozice.X - posunutiX, aktualniFigurka.Pozice.Y - posunutiY);
                    kamKreslit.Refresh();
                }
                if (formular.dodatecnyPosun > 0)
                {
                    if (posunutiX > 0)//doprava
                    {
                        posunutiX = formular.dodatecnyPosun;
                    }
                    else if (posunutiX < 0)//doleva
                    {
                        posunutiX = formular.dodatecnyPosun * (-1);
                    }
                    else if (posunutiY > 0)//dolu
                    {
                        posunutiY = formular.dodatecnyPosun;
                    }
                    else if (posunutiY < 0)//nahory
                    {
                        posunutiY = formular.dodatecnyPosun * (-1);
                    }
                    aktualniFigurka.Pozice = new Point(aktualniFigurka.Pozice.X - posunutiX, aktualniFigurka.Pozice.Y - posunutiY);
                    kamKreslit.Refresh();
```

```
                    }
                }

                //neaktivni figurka musi mit stejnou pozici jako ta aktivni
                aktualniHrac.NeaktivniFigurka.Pozice = aktualniHrac.Figurka.Pozice;
                //na konci posunuti urcim cilovy kamen jako ten aktualni, na kterem fugurka prave stoji
                if (kamen1 != null)//znamena to, ze cyklus probehl, tudiz byla nalezena nejaka cesta
                {
                    aktualniFigurka.AktualniKamen = kamen1;
                    aktualniHrac.NeaktivniFigurka.AktualniKamen = aktualniHrac.Figurka.AktualniKamen;

                    //zkontroluje zda se nenachazim na hledanem artefaktu
                    if (aktualniFigurka.AktualniKamen.Artefakt == aktualniHrac.HledanaKarta.Artefakt)//nasel jsem artefakt
                    {
                        //zkontroluju, zda se nejedna o posledni artefakt, ktery hrac hledal.
                        //kdyz ano, nastav konec hry
                        this.KartaNalezena(aktualniHrac.HledanaKarta, poziceNalezeneKarty);
                        if (this.hrajeSe==false)
                        {
                            MessageBox.Show(this.AktualniHrac.Nick + " vyhrál!");
                        }
                        else
                        {
                            this.DalsiHrac();
                            //prehodit prvky formu a refreshnout
                            kamKreslit.PrvkyNaFormu = this.prvkyNaFormu;
                            kamKreslit.Refresh();
                        }
                    }
                    else//jakmile hrac posunul figurku, hraje dalsi hrac
                    {
                        this.DalsiHrac();
                        kamKreslit.PrvkyNaFormu = this.prvkyNaFormu;
                        kamKreslit.Refresh();
                    }

                    KamKreslit.hracPosunul = false;//dalsi hrac muze posunovat kamenem
                }
                else//cesta nebyla nalezena
                {
                    KamKreslit.hracPosunul = true;
                }
            }
        }
        catch
        {
            MessageBox.Show("V metode PosunFigurku nastala chyba. Metoda se nachazi ve Hre.");
        }
    }
    //zacne hru
    public void ZacniHru()
    {
        //otestuju, zda aktualni hrac ma jiz nejakou nalezenou kartu
        if (aktualniHrac.Nalezene.Length != 0)
        {
            prvkyNaFormu[prvkyNaFormu.Length - 2] = aktualniHrac.Nalezene[aktualniHrac.Nalezene.Length - 1];//jako nalezenou
budu prezentovat tu posledni nalezenou
        }
        //jako posledni prvek pro vykresleni nastavim hledanou kartu
        if (aktualniHrac.Ai)
        {
            prvkyNaFormu[prvkyNaFormu.Length - 1] = formular.otaznik;
        }
        else
        {
            prvkyNaFormu[prvkyNaFormu.Length - 1] = aktualniHrac.HledanaKarta;
        }

    }
    //projde vsechny sousedni kameny a zjisti, jestli s nimi nema propojenou cestu, vysledky ulozi do pole propojeneKameny
    public void AktualizujPropojeniCest()
    {
        //postupne prochazim jednotlive kameny v pracovnim poli, kontroluju, zda nejsou na okraji a potom nastavuju jejich vlastnost
propojene cesty
        //vysledkem metody je, ze kazdy kamen bude mit nastaveny smery, ve kterych je propojen se sousednim kamenem
        Kamen praveVybrany;
```

```
int pocetCest;
bool[] smery;

//prochazim radky
for (int y = 0; y < 7; y++)//prochazim y souradnice, takze po radcich
{
   //prochazim sloupce
   for (int x = 0; x < 7; x++)
   {
      //vemu kamen a vytvorim mu pouzitelne smery, zatim prazdne
      praveVybrany = kameny[y, x];
      praveVybrany.PouzitelneSmery = new bool[4];

      //zjistim, kolik cest danny kamen obsahuje-krizovatka ma 3, zatacka i rovina 2
      pocetCest = praveVybrany.Cesty.Length;

      //zjistim smery, do kterych tyto cesty smeruji-urcuje se na zaklade natoceni
      smery = praveVybrany.Cesty;

      //zkontroluju, zda dany smer nemiri ven z hraci plochy, tudiz jestli je danny smer pouzitelny pro spojeni cesty
      //postupne projdu vsechny smery, i ty, ktere kamen vubec nema implementovane
      for (int i = 0; i < 4; i++)
      //0-doprava;1-dolu;2-doleva;3-nahoru
      {
         //zjistim, jestli kamen ma cestu smerujici timto smerem
         //kdyz je true, tak ji ma, kdyz false, tak ji nema
         if (smery[i])//kamen ma cestu, smerujici timto smerem
         {
            //na zaklade toho, o jaky smer se jedna rozhodnu, zda bude pouzitelny pro pruchodnost nebo ne
            switch (i)//smer
            {
               case 0://doprava
                  if (x == 6)//pozicne se nachazim na pravem okraji
                  {
                     praveVybrany.PouzitelneSmery[i] = false;
                  }
                  else
                  {
                     praveVybrany.PouzitelneSmery[i] = true;
                  }
                  break;
               case 1://dolu
                  if (y == 6)//nachazim se dole
                  {
                     praveVybrany.PouzitelneSmery[i] = false;
                  }
                  else
                  {
                     praveVybrany.PouzitelneSmery[i] = true;
                  }
                  break;
               case 2://doleva
                  if (x == 0)
                  {
                     praveVybrany.PouzitelneSmery[i] = false;
                  }
                  else
                  {
                     praveVybrany.PouzitelneSmery[i] = true;
                  }
                  break;
               case 3://nahoru
                  if (y == 0)
                  {
                     praveVybrany.PouzitelneSmery[i] = false;
                  }
                  else
                  {
                     praveVybrany.PouzitelneSmery[i] = true;
                  }
                  break;
            }

         }
         else//danny smer neobsahuje, takze mu ho rovnou nastavim jako nepouzitelny
         {
```

```
                    praveVybrany.PouzitelneSmery[i] = false;
                }
            }
        }
    }
    //vsechny kameny ted vi, ktere smery maji opravdu pouzitelne.
    //jestlize ma cestu doprava a nachazi se na levem okraji, je cesta nepouzitelna

    //jednotlivym kamenum prirazuju souradnice kamenu, se kterymi maji pruchodne cesty
    //vezmu kamen, projdu jeho smery a u kazdeho smeru prohledam sousedni kamen-ten musi mit smer opacny, aby se jednalo o pruchodnou cestu

    //prochazim radky
    for (int y = 0; y < 7; y++)
    {
        //sloupce
        for (int x = 0; x < 7; x++)
        {
            Point[] souradnice = null;
            //vemu kamen
            praveVybrany = kameny[y, x];
            int iks = 0, ypsylon = 0, smer = 0;

            //postupne projdu vsechny smery, vcetne tech, ktere kamen nema inmplementovane, ale to nemuzu predem vedet
            for (int i = 0; i < 4; i++)//vsechny smery
            {
                //zjistim, zda ma kamen danny smer nastaven na true, tedy povolen
                if (praveVybrany.PouzitelneSmery[i])//je povolen
                {
                    //na zaklade smeru rozhodnu ktery vedlejsi kamen mam kontrolovat
                    switch (i)
                    {
                        //kdyz ma smer doprava, znamena to, ze bude prohledava kamen napravo, zda nema cestu, ktera smeruje nalevo
                        case 0://doprava
                            smer = 2;
                            iks = x + 1;
                            ypsylon = y;
                            break;
                        case 1://dolu
                            smer = 3;
                            iks = x;
                            ypsylon = y + 1;
                            break;
                        case 2://doleva
                            smer = 0;
                            iks = x - 1;
                            ypsylon = y;
                            break;
                        case 3://nahoru
                            smer = 1;
                            iks = x;
                            ypsylon = y - 1;
                            break;
                    }

                    //zkontroluje, zda kamen vedle ma cestu v pozadovanem smeru
                    if (kameny[ypsylon, iks].Cesty[smer])//kamen tuto cestu obsahuje
                    {
                        //do pole souradnic pridam souradnice kamene, ktery sdili cestu s timto kamenem, tedy maji propojenou cestu
                        //hledam prvni smer, tudiz souradnice jeste nejsou zaplneny
                        if (souradnice == null)
                        {
                            souradnice = new Point[1];
                            souradnice[0] = new Point(ypsylon, iks);
                        }
                        else
                        {
                            Point[] tmp = new Point[souradnice.Length];
                            tmp = souradnice;
                            souradnice = new Point[souradnice.Length + 1];
                            tmp.CopyTo(souradnice, 0);
                            souradnice[souradnice.Length - 1] = new Point(ypsylon, iks);
                        }
                    }
                }
            }
```

```
                    //u kamene jsem prosel vsechny smery a zjistil, ve kterych ma propojenou cestu s ostatnimi kameny
                    //tyto souradnice nyni zkopiruju do jeho vlastnosti
                    praveVybrany.PropojeneKameny = souradnice;
                }
            }
            //prosel jsem vsechny kameny v na plose
        }
        //ze souradnic kamenu, ktere maji z kamenem projene cesty zjistim cestu
        public bool VytvorCestu(Point cil)
        {
            try
            {
                Point[] buffer = new Point[0];//buffer pro ukladani krizovatek
                //ulozim kamen, na kterem stoji aktualni hrac
                Kamen pocatecniKamen = aktualniHrac.Figurka.AktualniKamen;
                //cestu vytvorim prochazenim jednotlivych propojenych kamenu, ktere implementuje kazdy kamen. Kdyz zadny nema, ma
hodnoty null
                int pocetPropojeni;
                Point[] cesty = new Point[0];
                //Point[] propojeneKameny;
                Kamen predchoziKarta = null;//souradnice karty, ze ktere jsem dorazil na aktualni kartu-vyhledam  v pracovnim poli a mam
kamen
                Kamen aktualniKarta = pocatecniKamen;
                int jesteProhledat;//pocet cest k prohledani na dannem kamnu
                if (pocatecniKamen.PropojeneKameny != null)
                {
                    jesteProhledat = 1;//pocetPropojeni;
                }
                else
                {
                    jesteProhledat = 0;
                }
                //prochazim kamen po kameni a zapisuju do pole pro cestu
                int index;//slouzi pro vyber cesty z pole cest kamene
                int rozmery = cesty.Length;
                //---------------tady jsou chyby-----------------
                while (jesteProhledat != 0)
                {
                    if (aktualniKarta.JesteProhledat == null)//kdyz je prazdna, tak hledam poprve
                    {
                        aktualniKarta.JesteProhledat = aktualniKarta.PropojeneKameny;//pro zacatek se zde ulozi vsechny kameny, se kterymi je
propojen
                        if (predchoziKarta != null)//to znamena, ze jdu z prvni karticky-vychozi pozice, jinak tato situace nemuze nastat
                        {
                            //postupne porovnavam minule souradnice se souradnicemi v poli JesteProhledat, a kdyz je najdu, tak je odstranim z
pole
                            for (int i = 0; i < aktualniKarta.JesteProhledat.Length; i++)
                            {
                                if (predchoziKarta.VPracovnimPoli.X == aktualniKarta.JesteProhledat[i].X && predchoziKarta.VPracovnimPoli.Y
== aktualniKarta.JesteProhledat[i].Y)
                                {
                                    Point[] tmp = aktualniKarta.JesteProhledat;//docasne ulozim mimo
                                    aktualniKarta.JesteProhledat = new Point[aktualniKarta.JesteProhledat.Length - 1];//zmensim
                                    //nakopiruju zpatky
                                    //ten minuly kamen je na souradnicich "i", takze zpatky nakopiruju vsechny krome teto
                                    int indexPole = 0;
                                    for (int j = 0; j < tmp.Length; j++)
                                    {
                                        if (j != i)
                                        {
                                            aktualniKarta.JesteProhledat[indexPole] = tmp[j];
                                            indexPole += 1;
                                        }
                                    }
                                }
                            }
                        }
                        jesteProhledat = aktualniKarta.JesteProhledat.Length;
                    }
                    else//prohledavam kameny z bufferu
                    {
                        jesteProhledat = aktualniKarta.JesteProhledat.Length;
                    }
                    //nasledne najdu tu, ze ktere jsem prisel a tu z pole odstranim
                    int prvkuNaRozmer;
                    switch (jesteProhledat)
```

```
                {
                    case 0://konec cesty
                        //prohledat buffer, kdyz je prazdny, tak ukoncit. Jinak zvolit jako aktualni kamen ten v bufferu
                        if (buffer.Length == 0)
                        {
                            Point[] temp = cesty;
                            //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                            prvkuNaRozmer = temp.Length;
                            //rozmery + 1 pak upravit zpatky na rozmery
                            cesty = new Point[prvkuNaRozmer + 1];
                            temp.CopyTo(cesty, 0);
                            cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                            jesteProhledat = 0;

                        }
                        else
                        {
                            Point[] temp = cesty;
                            //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                            prvkuNaRozmer = temp.Length;//temp.GetLength(1);
                            //rozmery + 1 pak upravit zpatky na rozmery
                            cesty = new Point[prvkuNaRozmer + 1];
                            temp.CopyTo(cesty, 0);
                            cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                            jesteProhledat = 1;//jeste se bude hledat dale v bufferu
                            //nastavim posledni kartu z bufferu jako aktualni a pak ji z bufferu odeberu
                            Point souradniceKarty = buffer[buffer.Length - 1];//posledni karta
                            Kamen tmp = kameny[souradniceKarty.X, souradniceKarty.Y];//vytahnu ji z pracovniho pole
                            aktualniKarta = tmp;
                            //odeberu ji z bufferu
                            Point[] tmpBuffer = buffer;
                            buffer = new Point[buffer.Length - 1];//zmensim o jednu
                            for (int i = 0; i < buffer.Length; i++)
                            {
                                buffer[i] = tmpBuffer[i];
                            }
                        }
                        break;
                    case 1://----------bezny kamen----------

                        Point[] tempor = cesty;
                        //zjistim rozmery pole a take kolik ma kazdy rozmer prvku
                        prvkuNaRozmer = tempor.Length;
                        cesty = new Point[prvkuNaRozmer + 1];
                        //maji stejne prvku jako ty nad ni, jenom nejsou vsechny vyuzity
                        //----------zjistit aktualni cestu
                        //prekopiruju zpatky
                        tempor.CopyTo(cesty, 0);
                        cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                        Point dalsiKarta = aktualniKarta.JesteProhledat[0];
                        aktualniKarta.JesteProhledat = new Point[0];// null;
                        predchoziKarta = aktualniKarta;
                        aktualniKarta = kameny[dalsiKarta.X, dalsiKarta.Y];
                        break;
                    case 2://----------krizovatka----------
                    case 3:
                        //prida se do bufferu pro opetovne prohledani, protoze ma na sobe cestu, ktera se neprohleda
                        Point[] tmpbuffer = buffer;
                        buffer = new Point[buffer.Length + 1];
                        tmpbuffer.CopyTo(buffer, 0);
                        buffer[buffer.Length - 1] = aktualniKarta.VPracovnimPoli;
                        //vybrany kamen odeberu z prohledavani pro zpetny pruchod, vyhledam ho hned
                        Point[] tmp2 = null;
                        index = nahodneCislo.Next(0, aktualniKarta.JesteProhledat.Length);
                        for (int i = 0; i < aktualniKarta.JesteProhledat.Length; i++)
                        {
                            if (aktualniKarta.JesteProhledat[index].X == aktualniKarta.JesteProhledat[i].X &&
aktualniKarta.JesteProhledat[index].Y == aktualniKarta.JesteProhledat[i].Y)
                            {
                                tmp2 = aktualniKarta.JesteProhledat;//docasne ulozim mimo
                                aktualniKarta.JesteProhledat = new Point[aktualniKarta.JesteProhledat.Length - 1];//zmensim
                                //nakopituju zpatky
                                //ten minuly kamen je na souradnicich "i"
                                int indexPole = 0;
                                for (int j = 0; j < tmp2.Length; j++)
                                {
```

```
                                if (j != i)
                                {
                                    aktualniKarta.JesteProhledat[indexPole] = tmp2[j];
                                    indexPole += 1;
                                }
                            }
                        }
                    }
                    //pridam souradnice vybrane karty do cesty
                    Point[] tmp3 = cesty;
                    prvkuNaRozmer = tmp3.Length;
                    cesty = new Point[prvkuNaRozmer + 1];
                    tmp3.CopyTo(cesty, 0);
                    cesty[prvkuNaRozmer] = aktualniKarta.VPracovnimPoli;
                    //mozna prohodit indexy
                    predchoziKarta = aktualniKarta;
                    int iks = tmp2[index].X;
                    int yps = tmp2[index].Y;
                    aktualniKarta = kameny[iks, yps];//z pracovniho pole urcim jako aktualni kamen ten, ktery se vylosoval z jeste
nehledanych cest
                    if (aktualniKarta.PropojeneKameny != null)
                    {
                        pocetPropojeni = aktualniKarta.PropojeneKameny.Length;
                    }
                    else
                    {
                        pocetPropojeni = 0;
                    }
                    jesteProhledat = pocetPropojeni;
                    break;
                }
            }
            //------------konec chybneho bloku------------
            //kartam musim nastavit vlastnosti na hodnoty pred zapocenim hledanim, abych mohl stejne hledat i priste

            for (int h = 0; h < 7; h++)
            {

                for (int g = 0; g < 7; g++)
                {
                    kameny[h, g].JesteProhledat = null;
                }

            }

            //------------Cesta nalezena uspesne, ted jeste rozpoznat hlavni----------
            //ze vznikle cesty, ktera obsahuje serazeny zapis kamenu tak, jak se prochazelo pri hledani, vytvorim jednu jedinou cestu, po
ktere pujde figurka
            if (this.aktualniHrac.Ai)//hrac je pocitac
            {
                //staci mi zjistene propojene vsechny mozne cesty, ktere lze nalezt, ty pak dale budu zpracovavat-najdu konce cest, a podle
nich budu hledat
                //nejkratsi cestu k artefaktu
                //z cesty musim najit jejich konce
                //definice konce: u beznych cest se jedna o kameny, kter maji pouze jednu spojenou cestu
                //u krizovatek jsou to kameny, ktere nemaji spojeny vsechny 3 cesty, tedy maji spojene 2 cesty
                //to muzu vyresit uz pri samotnem hledani cesty, nebo zde projit vsechny kameny na ceste a urcit je jako koncove body
                Point[] pracovniCesta = cesty;//cesta, ktera vznikla
                return false;

            }
            else//hrac je clovek
            {
                Point[] pracovniCesty = cesty;
                bool hotovo = false;
                Point[] docasna = new Point[0];

                //prochazim jednotlive kameny pracovni cesty a zjistuji, zda nejaky neodpovida cili
                for (int i = 0; i < pracovniCesty.Length; i++)
                {
                    if (pracovniCesty[i] != cil)//souradnice nenalezena
                    {
                        hotovo = true;
                    }
                    else//nalezena
                    {
```

```
                hotovo = false;
                //ulozim cestu souradnice az po prave nalezenou do docasne cesty
                //projdu puvodni cestu az po prave nalezenou sourad
                docasna = new Point[i + 1];

                for (int j = 0; j <= i; j++)
                {
                   docasna[j] = pracovniCesty[j];
                }

                //---------po tud funguje spravne----------
                //v docasne ceste najdu misto, kde se nachazi stejna souradnice. Kdyz nic nenajdu, tak je cesta hotova
                //zjistim, ktera souradnice se v poli nachazi 2x

                while (hotovo == false)
                {
                   int pozicePrvniho = -1, poziceDruheho = 0;
                   bool ukonceno = false;

                   for (int y = 0; y < docasna.Length; y++)
                   {
                      for (int z = y + 1; z < docasna.Length; z++)
                      {
                         if (docasna[y] == docasna[z])//nalezeno neco 2x
                         {
                            pozicePrvniho = y;
                            poziceDruheho = z;
                            ukonceno = true;
                            break;
                         }
                      }
                      if (ukonceno) { break; }
                   }
                   if (pozicePrvniho != -1)//nalezeno neco 2x
                   {
                      Point[] tmpDocasna = docasna;
                      Point[] uschovna = new Point[tmpDocasna.Length - poziceDruheho + pozicePrvniho];
                      int uschovnaIndex = 0;
                      //prekopiruju data az do prvniho nalezeneho
                      for (int k = 0; k <= pozicePrvniho; k++)
                      {
                         uschovna[uschovnaIndex] = tmpDocasna[k];
                         uschovnaIndex += 1;
                      }
                      //dokopiruju data, ktera jsou za druhym nalezenym
                      for (int l = poziceDruheho + 1; l < tmpDocasna.Length; l++)
                      {
                         uschovna[uschovnaIndex] = tmpDocasna[l];
                         uschovnaIndex += 1;
                      }
                      docasna = uschovna;
                   }
                   else//v ceste neni zadna souradnice 2x
                   {
                      hotovo = true;
                   }
                }
                break;
             }
          }
          if (docasna == null)
          {
             throw new Exception("Cesta nebyLa vytvorena");//skoci na catch
             //return false;
          }
          else
          {
             this.Cesta = docasna;// cesty;
             return true;
          }
       }
    }
    catch
    {
       MessageBox.Show("Pri hledana cesty doslo k chybe");
       return false;
```

```
            }
        }

        //provadi ukony, ktere provadi pocitac, kdyz je na tahu
        public void NaTahuJePocitac()
        {
            casNaTah = 120;
            AI.AktualniHra = this;
            AI.CilovyForm = formular;
            AI.PracovniPole = formular.PracovniPole;
            AI.Posunovaci = formular.Posunovaci;
            formular.hracPosunul = true;
            AI.PosunZasun();
        }
        //udpocitavani casu, ktery zbyva do startu hry-pouze pokud zacina pocitac
        public void odpocitavaniStartu(object sender, ElapsedEventArgs e)
        {
            //z casovace odeberu tuto udalost, aby zustala jenom jedna pro obsluhu odpocitavani
            casovac.Stop();
            casovac.Elapsed -= odpocitavaniStartu;

            //zacnu odpocitavat cas hraci
            casovac.Elapsed += new System.Timers.ElapsedEventHandler(casovac_Elapsed);
            casovac.Interval = 1000;
            casovac.Start();

            ThreadPool.QueueUserWorkItem(new WaitCallback(formular.ProcesVeVlakne));
        }
        public void casovac_Elapsed(object sender, ElapsedEventArgs e)
        {
            ThreadPool.QueueUserWorkItem(new WaitCallback(formular.ProcesVeVlakne2));
        }

        //odpocitava cas, ktery ma hrac urcen na tah, a kdyz uplyne, nahodne zasune
        public void OdpocetCasu()
        {
            //odpocitavam cas a graficky ho zobrazuji
            casNaTah -= 1;
            int minuty, vteriny;
            minuty = (int)(casNaTah / 60);
            vteriny = casNaTah - minuty * 60;
            string text = minuty.ToString() + ":" + ((vteriny < 10) ? "0" + vteriny.ToString() : vteriny.ToString());
            formular.cas.TextovyObsah = text;
            formular.Refresh();

            //kdyz uplyne cas, nahodne zasunu kamen a prepnu na dalsiho hrace
            //muze se ale stat, ze hrac zasunul kamen, a pouze neposunul figurkou, v tom pripade uz znova zasunovat kamen nebudu
            if (casNaTah == 0)
            {
                if (!formular.hracPosunul)//hrac ani nezasunul kamen
                {
                    //nahodny kamen
                    Random nahodne = new Random();
                    int radeksloupec = nahodne.Next(1, 3);
                    int smer = nahodne.Next(1, 3);
                    int[] cisla = new int[] { 1, 3, 5 };
                    int pozice = nahodne.Next(0, 3);
                    int cislo = cisla[pozice];
                    //posune po plose
                    AI.PosunPoPlose(radeksloupec, cislo, smer, formular);
                    //posune se v pameti
                    formular.Posunout(radeksloupec, smer, cislo, formular.Posunovaci);

                    formular.predchoziCislo = cislo;
                    formular.predchoziSloupecRadek = radeksloupec;
                    formular.predchoziSmer = smer;

                    for (int hrac = 0; hrac < Hraci.Length; hrac++)
                    {
                        if (Hraci[hrac].Figurka.AktualniKamen == formular.Posunovaci)//figurka stoji na kameni, ktery je mimo hraci plochu
                        {
                            //Posunovaci si pamatuje souradnice v pracovnim poli. Pred posunutim na techto souradnicich stal, ale kdyz byl
posunut
                            //tyto hodnoty se mu neprepsaly,tudiz jsou porad stejne, i kdyz uz na nich nestoji
                            if (formular.Posunovaci.VPracovnimPoli == new Point(1, 0))//na levem okraji na druhem radku
                            {
```

```
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[1, 6];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(3, 0))//na levem okraji na ctvrtem radku
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[3, 6];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(5, 0))//na levem okraji na sestem radku
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[5, 6];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(1, 6))//na pravem okraji na druhem radku
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[1, 0];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(3, 6))//na pravem okraji na ctvrtem radku
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[3, 0];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(5, 6))//na pravem okraji na sestem radku
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[5, 0];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(0, 1))//na hornim okraji na druhem sloupci
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[6, 1];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(0, 3))//na hornim okraji na ctvrtem sloupci
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[6, 3];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(0, 5))//na hornim okraji na sestem sloupci
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[6, 5];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(6, 1))//na dolnim okraji na druhem sloupci
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[0, 1];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(6, 3))//na dolnim okraji na ctvrtem sloupci
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[0, 3];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(6, 5))//na dolnim okraji na sestem sloupci
                {
                    Hraci[hrac].Figurka.AktualniKamen = formular.PracovniPole[0, 5];
                }
            }
            if (Hraci[hrac].NeaktivniFigurka.AktualniKamen == formular.Posunovaci)//figurka stoji na kameni, ktery je mimo hraci
plochu
            {
                //Posunovaci si pamatuje souradnice v pracovnim poli. Pred posunutim na techto souradnicich stal, ale kdyz byl
posunut
                //tyto hodnoty se mu neprepsaly,tudiz jsou porad stejne, i kdyz uz na nich nestoji
                if (formular.Posunovaci.VPracovnimPoli == new Point(1, 0))//na levem okraji na druhem radku
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[1, 6];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(3, 0))//na levem okraji na ctvrtem radku
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[3, 6];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(5, 0))//na levem okraji na sestem radku
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[5, 6];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(1, 6))//na pravem okraji na druhem radku
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[1, 0];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(3, 6))//na pravem okraji na ctvrtem radku
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[3, 0];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(5, 6))//na pravem okraji na sestem radku
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[5, 0];
```

```
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(0, 1))//na hornim okraji na druhem sloupci
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[6, 1];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(0, 3))//na hornim okraji na ctvrtem sloupci
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[6, 3];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(0, 5))//na hornim okraji na sestem sloupci
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[6, 5];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(6, 1))//na dolnim okraji na druhem sloupci
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[0, 1];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(6, 3))//na dolnim okraji na ctvrtem sloupci
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[0, 3];
                }
                else if (formular.Posunovaci.VPracovnimPoli == new Point(6, 5))//na dolnim okraji na sestem sloupci
                {
                    Hraci[hrac].NeaktivniFigurka.AktualniKamen = formular.PracovniPole[0, 5];
                }
            }
        }

        Kameny = formular.PracovniPole;
        AktualizujPropojeniCest();

        Figurka[] proPosunuti = null;

        for (int i = 0; i < Hraci.Length; i++)//projdu vsechny hrace
        {
            for (int j = 0; j < formular.poleKaret.Length; j++)//u kazdeho z nich prohledam vsechny kameny, ktere se prave budou
posouvat
            {
                if (Hraci[i] == AktualniHrac)//aktualni hrac
                {
                    if (Hraci[i].Figurka.AktualniKamen == formular.poleKaret[j])//figurka stoji na kamenu, ktery se bude posouvat
                    {
                        //ulozim ji do pole pro posunuti figurky
                        if (proPosunuti == null)
                        {
                            proPosunuti = new Figurka[1];
                            proPosunuti[0] = Hraci[i].Figurka;
                        }
                        else
                        {
                            Figurka[] tmp = proPosunuti;
                            proPosunuti = new Figurka[proPosunuti.Length + 1];
                            tmp.CopyTo(proPosunuti, 0);
                            proPosunuti[proPosunuti.Length - 1] = Hraci[i].Figurka;
                        }
                        break;
                    }
                }
                else//neaktivni hrac, takze priradim tu neaktivni figurku
                {
                    if (Hraci[i].NeaktivniFigurka.AktualniKamen == formular.poleKaret[j])//figurka stoji na kamenu, ktery se bude
posouvat
                    {
                        //ulozim ji do pole pro posunuti figurky
                        if (proPosunuti == null)
                        {
                            proPosunuti = new Figurka[1];
                            proPosunuti[0] = Hraci[i].NeaktivniFigurka;
                        }
                        else
                        {
                            Figurka[] tmp = proPosunuti;
                            proPosunuti = new Figurka[proPosunuti.Length + 1];
                            tmp.CopyTo(proPosunuti, 0);
                            proPosunuti[proPosunuti.Length - 1] = Hraci[i].NeaktivniFigurka;
                        }
```

```
                            break;
                        }
                    }
                }
            }

            for (int a = 0; a < 10; a++)
            {
                formular.pohni(proPosunuti);//predat figurky, ktere stoji na dannem pruhu / vybrat variantu zasunuti, kde nejsou zadne
figurky
                formular.Refresh();
            }
            formular.hracPosunul = false;
            DalsiHrac();
            formular.PrvkyNaFormu = this.prvkyNaFormu;
            formular.Refresh();
        }
        else//hrac zasunul kamen,ale neposunul figurkou
        {
            formular.hracPosunul = false;
            DalsiHrac();
            formular.PrvkyNaFormu = this.prvkyNaFormu;
            formular.Refresh();
        }
    }
  }
}
```

C5 HRAC

```
using System;

namespace Labyrint
{
    class Hrac
    {
        private HraciKarta[] hraciKarty;
        private HraciKarta[] nalezene;
        private HraciKarta[] hledane;
        private HraciKarta hledanaKarta;
        private string nick;
        private Kamen posunovaci;//kamen, kterym hrac posunuje
        private Figurka figurka;
        private Figurka neaktivniFigurka;
        private int pocetTahu;//urcuje pocet tahu hrace
        private bool ai;
        private GrafickyObjekt tlacitko;
        //private int casNaTah;//v sekundach
        //---------konstruktor-----------
        public Hrac()
        {
            nick = "";
            hraciKarty = new HraciKarta[0];
            pocetTahu = 0;
            ai = false;

        }
        //------------vlastnosti------------
        public HraciKarta[] HraciKarty
        {
            get
            {

                return hraciKarty;
            }
            set
            {

                this.hraciKarty = value;
            }
        }
        public HraciKarta HledanaKarta
        {
```

```
    get
    {
        return hledanaKarta;
    }
    set
    {
        hledanaKarta = value;
    }
}
public String Nick
{
    get
    {
        return nick;
    }
    set
    {
        nick = value;
    }
}
public Kamen Posunovaci
{
    get
    {

        return posunovaci;
    }
    set
    {

        posunovaci = value;
    }
}
public HraciKarta[] Nalezene
{
    get
    {

        return nalezene;
    }
    set
    {

        nalezene = value;
    }
}
public HraciKarta[] Hledane
{
    get
    {

        return hledane;
    }
    set
    {

        hledane = value;
    }
}
public Figurka Figurka
{
    get
    {
        return figurka;
    }
    set
    {
        figurka = value;
    }
}
public Figurka NeaktivniFigurka
{
    get
    {
      return neaktivniFigurka;
    }
```

```
            set
            {
               neaktivniFigurka = value;
            }
         }
         public int PocetTahu
         {
            get
            {

                  return pocetTahu;
            }
            set
            {

                  pocetTahu = value;
            }
         }
         public bool Ai
         {
            get
            {
               return ai;
            }
            set
            {
               ai = value;
            }
         }
         public GrafickyObjekt Tlacitko
         {
            get
            {
               return tlacitko;
            }
            set
            {
               tlacitko = value;
            }
```

C6 HRACIFORM

```
using System;
using System.Windows.Forms;
using System.Drawing;

namespace Labyrint
{
   public partial class HraciForm : Form
   {
      //-----------
      Random generNahodne;//trida pro generovani nahodnych cisel
      public Kamen Posunovaci;//ulozeni kamene, ktery prave slouzi k posunovani ostatnich kamenu
      public Kamen[,] PracovniPole = new Kamen[7, 7];//slouzi jako otisk hraci plochy - pro programove ucely, vzdy, kdyz nastane
zmena, tak se aktualizuje
      public Kamen[] poleKaret = new Kamen[8];//zde se ukladaji karty pri posunovani pro posouvani
      Point[,] poleHodnot = new Point[2, 8];//
      int RAdekSloupec, SMer, pocetPosunuti = 0;
      public int predchoziSloupecRadek=0, predchoziSmer=0, predchoziCislo=0;
      Point[] povoleneSourad = new Point[4];//souradnice, ve kterych se smim vyskytovat pri presunovani posunovaciho kamene po
hraci plose
      int cyklusPosunutiProbehl = 0;//pocet probehnuti cyklu pro posunuti kamenu


      //souradnice pro dane rozliseni
      int[] poleSouradnicX;
      int[] poleSouradnicY;
      //velikost posunu pri posunuti na formu
      float velikostPosunu, velikostPohybu;
      Point souradMinimalizace, souradZavrit, souradDalsiHrac, souradCas, souradHledanyTitulek;


      int vyskaKamene, sirkaKamene;
      public int dodatecnyPosun;
```

```
        Hra hra;
        public Rozliseni umisteni;
        bool ukoncitProgram = false;//rika, zda mam ukoncit program - hodnota se nastavi na true, jestlize nastane napriklad nejaka chyba
nebo tak neco
        //---------konstruktor----------
        public HraciForm()//konstruktor
        {
            try
            {
                generNahodne = new Random();//pro generovani nahodnych cisel
                //nastavi vse potrebne pro spusteni formulare a pro dalsi praci na nem, vcetne nastaveni dat
                NactiObjekty();
                //uzpusobi jednotlive velikosti prvku, ktere se budou vykreslovat
                PrizpusobRozliseni();
                //hraci kameny nahodne rozmisti na form a vrati jednu posunovaci karticku
                Kamen[] naPlochu = this.VyberNaPlochu(out this.Posunovaci);
                this.NahodneRozmistit(naPlochu);
            }
            catch
            {
            }
        }

        //------------------metody--------------------
        protected override void OnPaint(PaintEventArgs e)//metoda pro vykreslovani formulare a veci na nem
        {
            for (int i = 0; i < prvkyNaFormu.Length; i++)
            {
                if (prvkyNaFormu[i] != null)
                {
                    prvkyNaFormu[i].Vykresli(e.Graphics);
                }
            }
            kontrols[0].Show();
        }

        //veskere rozmery hry prizpusobi rozmerum monitoru
        private void PrizpusobRozliseni()
        {
            Rectangle rozliseni = Screen.PrimaryScreen.Bounds;
            Rectangle pracivniPlocha = Screen.PrimaryScreen.WorkingArea;

            this.StartPosition = FormStartPosition.Manual;
            //1,6 je sirokouhly monitor s pomerem 16:10
            PomerStran pomer = PomerStran.klasicke;
            float pomerStran = (float)rozliseni.Width / (float)rozliseni.Height;
            const float sestnactKuDeviti=1.77777778f;
            const float sestnactKuDesiti=1.6f;
            const float patnactKuDeviti=1.66666667f;
            const float ctyriKuTrem=1.33333337f;
            const float petKuCtyrem=1.25f;

            if (pomerStran == sestnactKuDeviti || pomerStran == sestnactKuDesiti || pomerStran == patnactKuDeviti)
            {
                pomer = PomerStran.sirokouhle;
            }
            else if (pomerStran == ctyriKuTrem)
            {
                pomer = PomerStran.klasicke;
            }
            else if (pomerStran == petKuCtyrem)
            {
                pomer = PomerStran.petKuCtyrem;
            }
            else
            {
                MessageBox.Show("Litujeme, nemáte nastaveno podporované rozlišení. Tyto jsou 1280x800, 1680x1050.");
                ukoncitProgram = true;
            }
            //na zaklade uhlosti monitoru, tj. zda je sirokouhly ci ne, priradim jednotliva rozliseni
            switch (pomer)
            {
                //sirokouhle
                case PomerStran.sirokouhle:
                    if (rozliseni.Width == 1280 && rozliseni.Height == 800)
```

```
            {
                //rozmery formu
                this.Width = 908;
                this.Height = 760;
                //umisteni na stred
                int zleva = (pracivniPlocha.Width - this.Width) / 2;
                int shora = (pracivniPlocha.Height - this.Height) / 2;
                //umistim na stred
                this.DesktopLocation = new Point(zleva, shora);
                //nastaveni rozmeru vsem prvkum, ktere se budou vykreslovat na form
                for (int i = 0; i <= prvkyNaFormu.Length - 1; i++)
                {
                    if (prvkyNaFormu[i].Jmeno == "Plocha")
                    {
                        prvkyNaFormu[i].Rozmery = new Size(908, 760);
                        prvkyNaFormu[i].Obrazek = (Image)Zdroj.Plocha1280x800;
                    }
                    else if (prvkyNaFormu[i].Jmeno.Contains("Kamen"))
                    {
                        prvkyNaFormu[i].Rozmery = new Size(70, 70);
                        Image obr = prvkyNaFormu[i].Obrazek;
                        prvkyNaFormu[i].Obrazek = (Image)new Bitmap(obr, new Size(70, 70));
                        obr.Dispose();
                    }
                }
                //pri startu programu zjistit rozliseni a na jeho zaklade se rozhodne, ktere obrazky si natahne z disku do pameti-ve tride
Zdroj
                poleSouradnicX = new int[] { 137, 207, 277, 347, 417, 487, 557 };
                poleSouradnicY = new int[] { 136, 206, 276, 346, 416, 486, 556 };
                velikostPosunu = 7;
                vyskaKamene = 70;
                sirkaKamene = 70;

                umisteni = new DvanactSetXOsmSet(this.CreateGraphics(), hracTitulek, nalezenoKolik, pocetTahu,
prohledaniNalezenych);
            }
            else if (rozliseni.Width == 1680 && rozliseni.Height == 1050)
            {
                //rozmery formu
                this.Width = 908;
                this.Height = 760;
                //umisteni na stred
                int zleva = (pracivniPlocha.Width - this.Width) / 2;
                int shora = (pracivniPlocha.Height - this.Height) / 2;
                //umistim na stred
                this.DesktopLocation = new Point(zleva, shora);
                //nastaveni rozmeru vsem prvkum, ktere se budou vykreslovat na form
                for (int i = 0; i <= prvkyNaFormu.Length - 1; i++)
                {
                    if (prvkyNaFormu[i].Jmeno == "Plocha")
                    {
                        prvkyNaFormu[i].Rozmery = new Size(908, 760);
                        prvkyNaFormu[i].Obrazek = (Image)Zdroj.Plocha1280x800;
                    }
                    else if (prvkyNaFormu[i].Jmeno.Contains("Kamen"))
                    {
                        prvkyNaFormu[i].Rozmery = new Size(70, 70);
                        Image obr = prvkyNaFormu[i].Obrazek;
                        prvkyNaFormu[i].Obrazek = (Image)new Bitmap(obr, new Size(70, 70));
                        obr.Dispose();
                    }
                }
                //pri startu programu zjistit rozliseni a na jeho zaklade se rozhodne, ktere obrazky si natahne z disku do pameti-ve tride
Zdroj
                poleSouradnicX = new int[] { 137, 207, 277, 347, 417, 487, 557 };
                poleSouradnicY = new int[] { 136, 206, 276, 346, 416, 486, 556 };
                velikostPosunu = 7;
                vyskaKamene = 70;
                sirkaKamene = 70;

                umisteni = new DvanactSetXOsmSet(this.CreateGraphics(), hracTitulek, nalezenoKolik, pocetTahu,
prohledaniNalezenych);
            }
            else//defaultni rozmery pro nepodporovane sirokouhle rozliseni
            {
                //rozmery formu
```

```
                this.Width = 908;
                this.Height = 760;
                //umisteni na stred
                int zleva = (pracivniPlocha.Width - this.Width) / 2;
                int shora = (pracivniPlocha.Height - this.Height) / 2;
                //umistim na stred
                this.DesktopLocation = new Point(zleva, shora);
                //nastaveni rozmeru vsem prvkum, ktere se budou vykreslovat na form
                for (int i = 0; i <= prvkyNaFormu.Length - 1; i++)
                {
                   if (prvkyNaFormu[i].Jmeno == "Plocha")
                   {
                      prvkyNaFormu[i].Rozmery = new Size(908, 760);
                      prvkyNaFormu[i].Obrazek = (Image)Zdroj.Plocha1280x800;
                   }
                   else if (prvkyNaFormu[i].Jmeno.Contains("Kamen"))
                   {
                      prvkyNaFormu[i].Rozmery = new Size(70, 70);
                      Image obr = prvkyNaFormu[i].Obrazek;
                      prvkyNaFormu[i].Obrazek = (Image)new Bitmap(obr, new Size(70, 70));
                      obr.Dispose();
                   }
                }
                //pri startu programu zjistit rozliseni a na jeho zaklade se rozhodne, ktere obrazky si natahne z disku do pameti-ve tride
Zdroj
                poleSouradnicX = new int[] { 137, 207, 277, 347, 417, 487, 557 };
                poleSouradnicY = new int[] { 136, 206, 276, 346, 416, 486, 556 };
                velikostPosunu = 7;
                vyskaKamene = 70;
                sirkaKamene = 70;

                umisteni = new DvanactSetXOsmSet(this.CreateGraphics(), hracTitulek, nalezenoKolik, pocetTahu,
prohledaniNalezenych);
             }
             break;

          //normalni
          case PomerStran.klasicke:
             if (rozliseni.Width == 1024 && rozliseni.Height == 768)
             {
                //rozmery formu
                this.Width = 873;
                this.Height = 730;
                //umisteni na stred
                int zleva = (pracivniPlocha.Width - this.Width) / 2;
                int shora = (pracivniPlocha.Height - this.Height) / 2;
                //umistim na stred
                this.DesktopLocation = new Point(zleva, shora);
                //nastaveni rozmeru vsem prvkum, ktere se budou vykreslovat na form
                for (int i = 0; i <= prvkyNaFormu.Length - 1; i++)
                {
                   if (prvkyNaFormu[i].Jmeno == "Plocha")
                   {
                      prvkyNaFormu[i].Rozmery = new Size(873, 730);
                      prvkyNaFormu[i].Obrazek = (Image)Zdroj.Plocha1024x768;
                   }
                   else if (prvkyNaFormu[i].Jmeno.Contains("Kamen"))
                   {
                      prvkyNaFormu[i].Rozmery = new Size(67, 67);
                      Image obr = prvkyNaFormu[i].Obrazek;
                      prvkyNaFormu[i].Obrazek = (Image)new Bitmap(obr, new Size(67, 67));
                      obr.Dispose();
                   }
                }
                //pri startu programu zjistit rozliseni a na jeho zaklade se rozhodne, ktere obrazky si natahne z disku do pameti-ve tride
Zdroj
                poleSouradnicX = new int[] { 132, 199, 266, 333, 400, 467, 534 };
                poleSouradnicY = new int[] { 131, 198, 265, 332, 399, 466, 533 };
                velikostPosunu = 6.7f;
                vyskaKamene = 67;
                sirkaKamene = 67;

                umisteni = new TisicDvacetCtyriXSedmsetSedesatOsm(this.CreateGraphics(), hracTitulek, nalezenoKolik, pocetTahu,
prohledaniNalezenych);

             }
```

```
                break;
            case PomerStran.petKuCtyrem:
                if (rozliseni.Width == 1280 && rozliseni.Height == 1024)
                {
                    //rozmery formu
                    this.Width = 873;
                    this.Height = 730;
                    //umisteni na stred
                    int zleva = (pracivniPlocha.Width - this.Width) / 2;
                    int shora = (pracivniPlocha.Height - this.Height) / 2;
                    //umistim na stred
                    this.DesktopLocation = new Point(zleva, shora);
                    //nastaveni rozmeru vsem prvkum, ktere se budou vykreslovat na form
                    for (int i = 0; i <= prvkyNaFormu.Length - 1; i++)
                    {
                        if (prvkyNaFormu[i].Jmeno == "Plocha")
                        {
                            prvkyNaFormu[i].Rozmery = new Size(873, 730);
                            prvkyNaFormu[i].Obrazek = (Image)Zdroj.Plocha1024x768;
                        }
                        else if (prvkyNaFormu[i].Jmeno.Contains("Kamen"))
                        {
                            prvkyNaFormu[i].Rozmery = new Size(67, 67);
                            Image obr = prvkyNaFormu[i].Obrazek;
                            prvkyNaFormu[i].Obrazek = (Image)new Bitmap(obr, new Size(67, 67));
                            obr.Dispose();
                        }
                    }
                    //pri startu programu zjistit rozliseni a na jeho zaklade se rozhodne, ktere obrazky si natahne z disku do pameti-ve tride
Zdroj
                    poleSouradnicX = new int[] { 132, 199, 266, 333, 400, 467, 534 };
                    poleSouradnicY = new int[] { 131, 198, 265, 332, 399, 466, 533 };
                    velikostPosunu = 6.7f;
                    vyskaKamene = 67;
                    sirkaKamene = 67;

                    umisteni = new DvanactSetXOsmSet(this.CreateGraphics(), hracTitulek, nalezenoKolik, pocetTahu,
prohledaniNalezenych);

                }

                break;
            }
            if (!ukoncitProgram)
            {
                //aplikovani jednotlivych rozmeru dle rozliseni
                velikostKarty = umisteni.velikostKarty;
                poziceKarty = umisteni.poziceKarty;
                poziceKartyNalezene = umisteni.poziceKartyNalezene;
                velikostFigurky = umisteni.velikostFigurky;
                minimalizace.Pozice = umisteni.Minimalizace;
                zavreni.Pozice = umisteni.Zavrit;
                dalsiHrac.Pozice = umisteni.DalsiHrac;
                souradCas = umisteni.Cas;
                souradHledanyTitulek = umisteni.HledanyTitulek;

                nalezenTitulek.Pozice = umisteni.NalezenTitulek;
                nalezenoTitulek.Pozice = umisteni.NalezenoTitulek;
                pocetTahuTitulek.Pozice = umisteni.PocetTahuTitulek;
                sipkaLevaNick.Pozice = umisteni.SipkaLevaNick;
                sipkaPravaNick.Pozice = umisteni.SipkaPravaNick;
                sipkaLevaNalezeno.Pozice = umisteni.SipkaLevaNalezeno;
                sipkaPravaNalezeno.Pozice = umisteni.SipkaPravaNalezeno;

                zavreni.Obrazek = (Image)new Bitmap((Image)Zdroj.KrizekBily, umisteni.ZavritRozmery);
                minimalizace.Obrazek = (Image)new Bitmap((Image)Zdroj.MinimalizaceBila, umisteni.MinimalizaceRozmery);
                sipkaLevaNick.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaBila, umisteni.SipkaLevaNickRozmery);
                sipkaPravaNick.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaBila, umisteni.SipkaPravaNickRozmery);
                sipkaLevaNalezeno.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaBila, umisteni.SipkaLevaNalezenoRozmery);
                sipkaPravaNalezeno.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaBila, umisteni.SipkaPravaNalezenoRozmery);

                pocatecniPovolene = new Point[] { new Point(poleSouradnicX[1], poleSouradnicY[0] - vyskaKamene), new
Point(poleSouradnicX[3], poleSouradnicY[0] - vyskaKamene), new Point(poleSouradnicX[5], poleSouradnicY[0] - vyskaKamene),
                new Point(poleSouradnicX[6] + sirkaKamene, poleSouradnicY[1]), new Point(poleSouradnicX[6] + sirkaKamene,
poleSouradnicY[3]), new Point(poleSouradnicX[6] + sirkaKamene, poleSouradnicY[5]),
```

```
            new Point(poleSouradnicX[5], poleSouradnicY[6] + vyskaKamene),new Point(poleSouradnicX[3], poleSouradnicY[6] +
vyskaKamene), new Point(poleSouradnicX[1], poleSouradnicY[6] + vyskaKamene),
            new Point(poleSouradnicX[0] - sirkaKamene, poleSouradnicY[5]), new Point(poleSouradnicX[0] - sirkaKamene,
poleSouradnicY[3]), new Point(poleSouradnicX[0] - sirkaKamene, poleSouradnicY[1]) };
            koncovePovolene = new Point[] { new Point(poleSouradnicX[1] + sirkaKamene, poleSouradnicY[0]), new
Point(poleSouradnicX[3] + sirkaKamene, poleSouradnicY[0]), new Point(poleSouradnicX[5] + sirkaKamene, poleSouradnicY[0]),
            new Point(poleSouradnicX[6] + 2 * sirkaKamene, poleSouradnicY[1] + vyskaKamene), new Point(poleSouradnicX[6] + 2 *
sirkaKamene, poleSouradnicY[3] + vyskaKamene), new Point(poleSouradnicX[6] + 2 * sirkaKamene, poleSouradnicY[5] + vyskaKamene),
            new Point(poleSouradnicX[5] + sirkaKamene, poleSouradnicY[6] + 2 * vyskaKamene), new Point(poleSouradnicX[3] +
sirkaKamene, poleSouradnicY[6] + 2 * vyskaKamene), new Point(poleSouradnicX[1] + sirkaKamene, poleSouradnicY[6] + 2 *
vyskaKamene),
            new Point(poleSouradnicX[0], poleSouradnicY[5] + vyskaKamene), new Point(poleSouradnicX[0], poleSouradnicY[3] +
vyskaKamene),new Point(poleSouradnicX[0], poleSouradnicY[1] + vyskaKamene) };


            //aplikuje rozmery a souradnice na hraci karty-artefakty
            for (int i = 0; i < hraciKarty.Length; i++)
            {
               hraciKarty[i].Rozmery = velikostKarty;
               hraciKarty[i].Pozice = poziceKarty;
               Bitmap obr = new Bitmap(hraciKarty[i].Obrazek, velikostKarty);
               hraciKarty[i].Obrazek = (Image)obr;
            }
            cas.Pozice = souradCas;
            hledanyTitulek.Pozice = souradHledanyTitulek;

            for (int i = 0; i < hraciKartyVychozi.Length; i++)
            {
               hraciKartyVychozi[i].Rozmery = velikostKarty;
               hraciKartyVychozi[i].Pozice = poziceKarty;
               Bitmap obr = new Bitmap(hraciKartyVychozi[i].Obrazek, velikostKarty);
               hraciKartyVychozi[i].Obrazek = (Image)obr;
            }
         }
         else
         {
            throw new Exception();
         }
      }

      //nastaveni souradnic kamenum

      private int NahodneCislo(int Od, int Do)//generuje nahodne cislo v zadanem rozsahu
      {
         int nahodneCislo = generNahodne.Next(Od, Do);
         return nahodneCislo;
      }

      //vybira 49 karticek, ktere budou vykresleny na hraci desku, zatimco jedna karta zbude a bude pouzita jako posunovaci

      private Kamen[] VyberNaPlochu(out Kamen posunovaci)
      {
         try
         {
            //pole s kartami, ktere maji pevne nastavene souradnice a pri hre s nimi nelze pohybovat
            Kamen[] pevna = new Kamen[] {kamen1,null,kamen3,null,kamen5,null,
            kamen7,null,null,null,null,null,null,null,
            kamen15,null,kamen17,null,kamen19,null,kamen21,null,null,null,null,null,null,null,kamen29,null,kamen31,null,
            kamen33,null,kamen35,null,null,null,null,null,null,null,kamen43,null,kamen45,null,kamen47,null,kamen49};

            //pole s kartami, ktere jsou pohyblive a pri hre se posunuji
            //z tohoto pole se vybere 33 kamenu, ktere se umisti na formular, jedna, ktera zbyde, se umisti jako posunovaci
            Kamen[] posuvna = new
Kamen[]{kamen2,kamen4,kamen6,kamen8,kamen9,kamen10,kamen11,kamen12,kamen13,kamen14,kamen16,kamen18,kamen20,

kamen22,kamen23,kamen24,kamen25,kamen26,kamen27,kamen28,kamen30,kamen32,kamen34,kamen36,kamen37,kamen38,kamen39,kam
en40,kamen41,kamen42,
            kamen44,kamen46,kamen48,kamen50};

            //nahodne vybere z pole posuvnych karticku, ktera bude slouzit k posouvani
            int indexVPosuvna = NahodneCislo(0, posuvna.Length);//od 0 do 33. Je tam do 34, ale tu nikdy nevybere
            posunovaci = posuvna[indexVPosuvna];
            posunovaci.Pozice = new Point(poleSouradnicX[3], poleSouradnicY[0] - kamen1.Rozmery.Height);
            posunovaci.Uhel = 90;

            int index = 0;//pocita hodnoty null
```

```
            //prochazi pole pevnych karet a kontroluje zda nema hodnotu null, jestli ano, zkontroluje, zda v poli s posuvnymi kartami
aktualne nepracuje s
            //posunovaci kartou, kdyz ne, tak ji priradi do pole s pevnymi kartami. Toto pole je pak vraceno k dalsimu zpracovani
            for (int i = 0; i < pevna.Length; i++)
            {
               if (pevna[i] == null)
               {
                  if (posuvna[index]!=posunovaci)//(posuvna[index].Jmeno != posunovaci.Jmeno)
                  {
                     pevna[i] = posuvna[index];
                     index += 1;
                  }
                  else
                  {
                     index += 1;
                     pevna[i] = posuvna[index];
                     index += 1;
                  }

               }
            }
            //vrati pole pevnych karet
            return pevna;
         }
         catch
         {
            MessageBox.Show("Omlouváme se, v programu došlo k chybě kvůli náhodnému výběru hracích kamenů do hry");
            posunovaci = null;
            return null;
         }

      }

      //karticky rozmisti po hraci plose
      private bool NahodneRozmistit(Kamen[] Pictures)
      {
         try
         {
            if (Pictures != null)
            {
               //zamichane souradnice
               int[] zamichaneX = new int[49];
               int[] zamichaneY = new int[49];
               //pevne souradnice jsou oznaceny hodnotou nula, pohyblive svymi souradnicemi
               int[] tmpX = new int[49];
               int[] tmpY = new int[49];
               //pouze pohyblive souradnice
               int[] tmpX2 = new int[33];
               int[] tmpY2 = new int[33];
               //pouze pevne souradnice
               int[] tmpXpevne = new int[16];
               int[] tmpYpevne = new int[16];
               //uhly pevnych policek
               float[] uhelPevny = new float[] { 270F, 180F, 180F, 0F, 90F, 90F, 180F, 270F, 90F, 0F, 270F, 270F, 180F, 0F, 0F, 90F };
               float[] uhel = new float[] { 0F, 90F, 180F, 270F };
               //odkazy na pictureboxy umistim do pole stejne jako do kolekce, budu mit pristup ke vsem atributum


               //pomocne promenne pro operace v cyklech
               int a = 0, b = 0, r = 0;
               int souradniceY = poleSouradnicY.Length - 1, souradniceX = poleSouradnicX.Length - 1;

               //vytvorim 3 pole, kde budou umisteny ruzne druhy souradnic
               for (int y = 0; y <= souradniceY; y++)//souradnice y
               {
                  for (int x = 0; x <= souradniceX; x++)//souradnice x
                  {
                     if ((y % 2 == 0 && x % 2 == 0) || (y == 0 && x == 0) || (y == 0 && x % 2 == 0) || (x == 0 && y % 2 == 0))//jedna se
o pevne policko
                     {
                        tmpX[a] = 0;//do pole pro oboji souradnice misto pevne souradnice zapisu nulu
                        tmpY[a] = 0;

                        tmpYpevne[r] = poleSouradnicY[y];//ulozeni pevnych souradnic
                        tmpXpevne[r] = poleSouradnicX[x];
```

```
            //vyuziju cyklu a rovnou priradim uhel
            Pictures[a].Uhel = uhelPevny[r];

            r += 1;
         }
         else//pohyblive souradnice
         {
            tmpX[a] = poleSouradnicX[x];//doplneni pole o pohyblive souradnice
            tmpY[a] = poleSouradnicY[y];

            tmpX2[b] = poleSouradnicX[x];//ulozeni pohyblivych souradnic
            tmpY2[b] = poleSouradnicY[y];

            //rovnou priradim uhel posuvnemu policku
            Pictures[a].Uhel = uhel[NahodneCislo(0, uhel.Length - 1)];

            b += 1;
         }
         a += 1;
      }
   }

   //generovani nahodnych cisel-funguji jako indexy pro pole
   //prohozeni souradnic-zamichani
   int cislo1, cislo2, tmp;

   for (int k = 0; k < 100; k++)//melo by to stacit
   {

      //vygenerovani nahodnych cisel
      cislo1 = NahodneCislo(0, tmpX2.Length);
      cislo2 = NahodneCislo(0, tmpX2.Length);
      //samotne prehazovani
      tmp = tmpX2[cislo1];//X ulozim docasne mimo
      tmpX2[cislo1] = tmpX2[cislo2];
      tmpX2[cislo2] = tmp;

      tmp = tmpY2[cislo1];//Y ulozim docasne mimo
      tmpY2[cislo1] = tmpY2[cislo2];
      tmpY2[cislo2] = tmp;
   }

   //zkopirovani prehozenych hodnot pole zpet do puvodniho pole
   int l = 0, n = 0;
   int tmpYKonec = tmpY.Length - 1;

   for (int m = 0; m <= tmpYKonec; m++)
   {
      if (tmpX[m] != 0)//nejdedna se o pevne policko
      {
         tmpX[m] = tmpX2[l];//zamichane zkopiruje do puvodniho pole
         tmpY[m] = tmpY2[l];
         l += 1;
      }
      else//pevne soradnice
      {
         tmpX[m] = tmpXpevne[n];//doplni od pole z pole pro pevne souradnice
         tmpY[m] = tmpYpevne[n];

         n += 1;
      }
   }

   //zkopiruju hodnoty do zvlatniho pole
   tmpX.CopyTo(zamichaneX, 0);
   tmpY.CopyTo(zamichaneY, 0);

   //rozmisti jednotlive karty po hraci plose
   int konec = zamichaneX.Length;

   for (int index = 0; index <= konec - 1; index++)
   {
      Pictures[index].Pozice = new Point(zamichaneX[index], zamichaneY[index]);
   }
   //ulozim si hraci karty na hraci plose do matice, kde s nimi budu nadale pracovat
```

```
            int pom = 0;

            for (int s = 0; s <= 6; s++)//radky
            {
               for (int t = 0; t <= 6; t++)//sloupce
               {
                  int x = 0, y = 0;

                  for (int i = 0; i <= poleSouradnicX.Length - 1; i++)//hleda souradnice v poli souradnic
                  {
                     if (Pictures[pom].Pozice.X == poleSouradnicX[i])
                     {
                        y = i;
                     }

                     if (Pictures[pom].Pozice.Y == poleSouradnicY[i])
                     {
                        x = i;
                     }
                  }
                  PracovniPole[x, y] = Pictures[pom];
                  //kamen si bude pamatovat souradnici v pracovnim poli
                  PracovniPole[x, y].VPracovnimPoli = new Point(x, y);
                  pom += 1;
               }
            }
            return true;
         }
         else
         {
            return false;
         }
      }
      catch
      {
         return false;
      }
   }


   //posune sloupec nebo radek v pameti
   public void Posunout(int radekSloupec, int smer, int cisloRadkuSloupce, Kamen vsunovanaKarta)
   {
      //radekSloupec - 1=radek, 2=sloupec
      //smer - jestlize se jedna o radek: 1=doleva, 2=doprava
      //       jestlize se jedna o sloupec: 1=dolu, 2=nahoru
      //zjistim o jaky typ posunuti se vlastne jedna
      Kamen tmpVsunovana = null;
      Point[,] PracovniPoleOtisk = new Point[7, 7];

      //ulozeni potrebnych pozic do extra pole, protoze dochazi k prepisu pri posunovani
      for (int i = 0; i <= 6; i++)//cislo prevest na PracovniPoleOtisk.Lenght
      {
         for (int j = 0; j <= 6; j++)
         {
            PracovniPoleOtisk[i, j] = PracovniPole[i, j].Pozice;
         }
      }


      switch (radekSloupec)
      {
         case 1://radek
            switch (smer)
            {
               case 1://doleva
                  for (int j = cisloRadkuSloupce; j == cisloRadkuSloupce; j++)//radek je porad stejny
                  {
                     for (int i = 0; i <= 7; i++)//sloupce se meni
                     {
                        if (i == 0)
                        {
                           poleKaret[i] = PracovniPole[j, i];

                           tmpVsunovana = PracovniPole[j, i];//karta, ktera se vysunuje ven z hraciho pole
                        }
```

```
                else if (i == 7)//karta, ktera se vsunuje do hraciho pole
                {
                    poleKaret[i] = vsunovanaKarta;

                    //ulozeni v poli
                    PracovniPole[j, i - 1] = vsunovanaKarta;
                    PracovniPole[j, i - 1].VPracovnimPoli = new Point(j, i - 1);
                    Posunovaci = tmpVsunovana;
                }
                else//karty, ktere se posunuji vne hraciho pole
                {
                    poleKaret[i] = PracovniPole[j, i];//prave posunova karta

                    //ulozeni v poli
                    PracovniPole[j, i - 1] = PracovniPole[j, i];
                    PracovniPole[j, i - 1].VPracovnimPoli = new Point(j, i - 1);
                }
            }
        }
        break;
    case 2://doprava
        for (int j = cisloRadkuSloupce; j == cisloRadkuSloupce; j++)//radek je porad stejny
        {
            for (int i = 6; i >= -1; i--)//sloupce se meni
            {
                if (i == 6)
                {
                    poleKaret[i + 1] = PracovniPole[j, i];
                    tmpVsunovana = PracovniPole[j, i];//karta, ktera se vysunuje ven z hraciho pole
                }
                else if (i == -1)//karta, ktera se vsunuje do hraciho pole
                {
                    poleKaret[i + 1] = vsunovanaKarta;

                    //ulozeni v poli
                    PracovniPole[j, i + 1] = vsunovanaKarta;
                    PracovniPole[j, i + 1].VPracovnimPoli = new Point(j, i + 1);
                    Posunovaci = tmpVsunovana;
                }
                else//karty, ktere se posunuji vne hraciho pole
                {
                    poleKaret[i + 1] = PracovniPole[j, i];//prave posunova karta

                    //ulozeni v poli
                    PracovniPole[j, i + 1] = PracovniPole[j, i];
                    PracovniPole[j, i + 1].VPracovnimPoli = new Point(j, i + 1);
                }
            }
        }
        break;
    }
    break;
case 2://sloupec
    switch (smer)
    {
        case 1://dolu
            for (int i = 6; i >= -1; i--)
            {
                for (int j = cisloRadkuSloupce; j == cisloRadkuSloupce; j++)
                {
                    if (i == 6)
                    {
                        tmpVsunovana = PracovniPole[i, j];//karta, ktera se vysunuje ven z hraciho pole
                        poleKaret[i + 1] = PracovniPole[i, j];

                    }
                    else if (i == -1)//karta, ktera se vsunuje do hraciho pole
                    {
                        poleKaret[i + 1] = vsunovanaKarta;

                        //ulozeni v poli
                        PracovniPole[i + 1, j] = vsunovanaKarta;
                        PracovniPole[i + 1, j].VPracovnimPoli = new Point(i + 1, j);
                        Posunovaci = tmpVsunovana;
                    }
                    else//karty, ktere se posunuji vne hraciho pole
```

```
                    {
                        poleKaret[i + 1] = PracovniPole[i, j];//prave posunova karta

                        //ulozeni v poli
                        PracovniPole[i + 1, j] = PracovniPole[i, j];
                        PracovniPole[i + 1, j].VPracovnimPoli = new Point(i + 1, j);
                    }
                }
            }
            break;
        case 2://nahoru
            for (int i = 0; i <= 7; i++)
            {
                for (int j = cisloRadkuSloupce; j == cisloRadkuSloupce; j++)
                {
                    if (i == 0)
                    {
                        tmpVsunovana = PracovniPole[i, j];//karta, ktera se vysunuje ven z hraciho pole
                        poleKaret[i] = PracovniPole[i, j];
                    }
                    else if (i == 7)//karta, ktera se vsunuje do hraciho pole
                    {
                        poleKaret[i] = vsunovanaKarta;

                        //zmenim v poli
                        PracovniPole[i - 1, j] = vsunovanaKarta;
                        PracovniPole[i - 1, j].VPracovnimPoli = new Point(i - 1, j);
                        Posunovaci = tmpVsunovana;
                    }
                    else//karty, ktere se posunuji vne hraciho pole
                    {
                        poleKaret[i] = PracovniPole[i, j];//prave posunova karta
                        //zmenim v poli
                        PracovniPole[i - 1, j] = PracovniPole[i, j];
                        PracovniPole[i - 1, j].VPracovnimPoli = new Point(i - 1, j);
                    }
                }
            }
            break;
    }
    break;
}
RAdekSloupec = radekSloupec;
SMer = smer;
this.hra.AktualniHrac.PocetTahu += 1;
}

//posune na formu
public void pohni(Figurka[] proPosunuti)
{
    int tmpPosun = (int)velikostPosunu;
    dodatecnyPosun = (int)Math.Round(((velikostPosunu - tmpPosun) * 10));
    int Posunu = tmpPosun;

cyklusPosunutiProbehl += 1;
if (dodatecnyPosun == 0)
{
    switch (RAdekSloupec)
    {
        case 1://radek
            switch (SMer)
            {
                case 1://doleva
                    for (int i = 0; i <= 7; i++)//posunuje se 8 kamenu, 7 v plose a 1 posunovaci
                    {
                        poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X - (int)Posunu, poleKaret[i].Pozice.Y);
                    }
                    if (proPosunuti != null)
                    {
                        for (int i = 0; i < proPosunuti.Length; i++)
                        {
                            proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X - (int)Posunu, proPosunuti[i].Pozice.Y);
                        }
                    }
                    break;
                case 2://doprava
```

```
                for (int i = 0; i <= 7; i++)
                {
                    poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X + (int)Posunu, poleKaret[i].Pozice.Y);
                }
                if (proPosunuti != null)
                {
                    for (int i = 0; i < proPosunuti.Length; i++)
                    {
                        proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X + (int)Posunu, proPosunuti[i].Pozice.Y);
                    }
                }
                break;
            }
            break;
        case 2://sloupec
            switch (SMer)
            {
                case 1://dolu
                    for (int i = 0; i <= 7; i++)
                    {
                        poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X, poleKaret[i].Pozice.Y + (int)Posunu);
                    }
                    if (proPosunuti != null)
                    {
                        for (int i = 0; i < proPosunuti.Length; i++)
                        {
                            proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X, proPosunuti[i].Pozice.Y + (int)Posunu);
                        }
                    }
                    break;
                case 2://nahoru
                    for (int i = 0; i <= 7; i++)
                    {
                        poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X, poleKaret[i].Pozice.Y - (int)Posunu);
                    }
                    if (proPosunuti != null)
                    {
                        for (int i = 0; i < proPosunuti.Length; i++)
                        {
                            proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X, proPosunuti[i].Pozice.Y - (int)Posunu);
                        }
                    }
                    break;
            }
            break;
    }
}
else
{
    switch (RAdekSloupec)
    {
        case 1://radek
            switch (SMer)
            {
                case 1://doleva
                    for (int i = 0; i <= 7; i++)//posunuje se 8 kamenu, 7 v plose a 1 posunovaci
                    {
                        poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X - (int)Posunu, poleKaret[i].Pozice.Y);
                    }
                    if (proPosunuti != null)
                    {
                        for (int i = 0; i < proPosunuti.Length; i++)
                        {
                            proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X - (int)Posunu, proPosunuti[i].Pozice.Y);
                        }
                    }
                    if (cyklusPosunutiProbehl == 10)
                    {
                        for (int i = 0; i <= 7; i++)//posunuje se 8 kamenu, 7 v plose a 1 posunovaci
                        {
                            poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X - dodatecnyPosun, poleKaret[i].Pozice.Y);
                        }
                        if (proPosunuti != null)
                        {
                            for (int i = 0; i < proPosunuti.Length; i++)
                            {
```

```
                    proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X - dodatecnyPosun, proPosunuti[i].Pozice.Y);
                }
            }
        }
        break;
    case 2://doprava
        for (int i = 0; i <= 7; i++)
        {
            poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X + (int)Posunu, poleKaret[i].Pozice.Y);
        }

        if (proPosunuti != null)
        {
            for (int i = 0; i < proPosunuti.Length; i++)
            {
                proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X + (int)Posunu, proPosunuti[i].Pozice.Y);
            }
        }
        if (cyklusPosunutiProbehl == 10)
        {
            for (int i = 0; i <= 7; i++)//posunuje se 8 kamenu, 7 v plose a 1 posunovaci
            {
                poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X + dodatecnyPosun, poleKaret[i].Pozice.Y);
            }
            if (proPosunuti != null)
            {
                for (int i = 0; i < proPosunuti.Length; i++)
                {
                    proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X + dodatecnyPosun, proPosunuti[i].Pozice.Y);
                }
            }
        }
        break;
    }
    break;
case 2://sloupec
    switch (SMer)
    {
        case 1://dolu
            for (int i = 0; i <= 7; i++)
            {
                poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X, poleKaret[i].Pozice.Y + (int)Posunu);
            }
            if (proPosunuti != null)
            {
                for (int i = 0; i < proPosunuti.Length; i++)
                {
                    proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X, proPosunuti[i].Pozice.Y + (int)Posunu);
                }
            }
            if (cyklusPosunutiProbehl == 10)
            {
                for (int i = 0; i <= 7; i++)
                {
                    poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X, poleKaret[i].Pozice.Y + dodatecnyPosun);
                }
                if (proPosunuti != null)
                {
                    for (int i = 0; i < proPosunuti.Length; i++)
                    {
                        proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X, proPosunuti[i].Pozice.Y + dodatecnyPosun);
                    }
                }
            }
            break;
        case 2://nahoru
            for (int i = 0; i <= 7; i++)
            {
                poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X, poleKaret[i].Pozice.Y - (int)Posunu);
            }
            if (proPosunuti != null)
            {
                for (int i = 0; i < proPosunuti.Length; i++)
                {
                    proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X, proPosunuti[i].Pozice.Y - (int)Posunu);
                }
```

```
                        }
                        if (cyklusPosunutiProbehl == 10)
                        {
                            for (int i = 0; i <= 7; i++)
                            {
                                poleKaret[i].Pozice = new Point(poleKaret[i].Pozice.X, poleKaret[i].Pozice.Y - dodatecnyPosun);
                            }
                            if (proPosunuti != null)
                            {
                                for (int i = 0; i < proPosunuti.Length; i++)
                                {
                                    proPosunuti[i].Pozice = new Point(proPosunuti[i].Pozice.X, proPosunuti[i].Pozice.Y - dodatecnyPosun);
                                }
                            }
                        }
                        break;
                    }
                    break;
                }
            }
            if (cyklusPosunutiProbehl == 10) { cyklusPosunutiProbehl = 0; }
        }
    }


    public enum PomerStran
    {
        sirokouhle = 1,
        klasicke,
        petKuCtyrem
    }
}
```

C7 HRACIFORM-DESIGN

```
using System;
using System.Windows.Forms;
using System.Drawing;
using System.Drawing.Text;
using System.Timers;

namespace Labyrint
{
    public partial class HraciForm
    {
        GrafickyObjekt[] prvkyNaFormu = new GrafickyObjekt[0];
        HraciKarta[] hraciKarty = new HraciKarta[0];
        Kamen kamen1, kamen2, kamen3, kamen4, kamen5, kamen6, kamen7, kamen8, kamen9, kamen10, kamen11, kamen12, kamen13,
kamen14, kamen15, kamen16, kamen17, kamen18,
            kamen19, kamen20, kamen21, kamen22, kamen23, kamen24, kamen25, kamen26, kamen27, kamen28, kamen29, kamen30,
kamen31, kamen32, kamen33, kamen34, kamen35,
            kamen36, kamen37, kamen38, kamen39, kamen40, kamen41, kamen42, kamen43, kamen44, kamen45, kamen46, kamen47,
kamen48, kamen49, kamen50;
        GrafickyObjekt plocha;
        public GrafickyObjekt minimalizace, zavreni, dalsiHrac, sipkaLevaNick, sipkaPravaNick, sipkaLevaNalezeno,
sipkaPravaNalezeno;
        public Text cas, hledanyTitulek, nalezenTitulek, nalezenoTitulek, pocetTahuTitulek, hracTitulek, nalezenoKolik, pocetTahu,
prohledaniNalezenych;
        HraciKarta karta1, karta2, karta3, karta4, karta5, karta6, karta7, karta8, karta9, karta10, karta11, karta12, karta13, karta14, karta15,
            karta16, karta17, karta18, karta19, karta20, karta21, karta22, karta23, karta24;
        Size velikostKarty;
        Point poziceKarty;
        Point poziceFigurky;
        Point poziceKartyNalezene;//nastavuje se pri prizpusobovani rozliseni-udava pozici pro vykresleni artefaktu
        Size velikostFigurky;
        Label label = new Label();
        Control[] kontrols = new Control[1];
        PrivateFontCollection sadaFontu = new PrivateFontCollection();
        HraciKarta[] hraciKartyVychozi;
        public HraciKarta hraciKartaModra, hraciKartaCervena, hraciKartaZluta, hraciKartaZelena, otaznik;

        public GrafickyObjekt[] PrvkyNaFormu
        {
            get
```

```
            {
                return prvkyNaFormu;
            }
            set
            {
                prvkyNaFormu = value;
            }
        }

        private void NactiObjekty()
        {
            try
            {
                sadaFontu.AddFontFile("./Resources/fonty/segoesc.ttf");
            }
            catch (System.IO.FileNotFoundException e)
            {
                MessageBox.Show("Potřebný font nebyl v tomto počítači nalezen! Nainstalujte si, prosím, font s názvem Segoe Script. Poté
spusťte program znovu");
            }

            //vytvoreni objektu, ktere budou na formu
            //----------plocha----------
            plocha = new GrafickyObjekt();//((Image)Zdroj.Plocha);
            plocha.Jmeno = "Plocha";
            //---------kamen1----------
            kamen1 = new Kamen((Image)Zdroj.Pozice_cervena);
            kamen1.Jmeno = "Kamen1";
            kamen1.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
            kamen1.TypKarty = typKarty.zatacka;
            kamen1.Artefakt = "Cervena";

            //---------kamen2----------
            kamen2 = new Kamen((Image)Zdroj.Zatacka);
            kamen2.Jmeno = "Kamen2";
            kamen2.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
            kamen2.TypKarty = typKarty.zatacka;

            //---------kamen3----------
            kamen3 = new Kamen((Image)Zdroj.Kniha);
            kamen3.Jmeno = "Kamen3";
            kamen3.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
            kamen3.TypKarty = typKarty.krizovatka;
            kamen3.Artefakt = "Kniha";

            //---------kamen4----------
            kamen4 = new Kamen((Image)Zdroj.Zatacka);
            kamen4.Jmeno = "Kamen4";
            kamen4.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
            kamen4.TypKarty = typKarty.zatacka;

            //---------kamen5----------
            kamen5 = new Kamen((Image)Zdroj.Měšec);
            kamen5.Jmeno = "Kamen5";
            kamen5.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
            kamen5.TypKarty = typKarty.krizovatka;
            kamen5.Artefakt = "Mesec";

            //---------kamen6----------
            kamen6 = new Kamen((Image)Zdroj.Zatacka);
            kamen6.Jmeno = "Kamen6";
            kamen6.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
            kamen6.TypKarty = typKarty.zatacka;

            //---------kamen7----------
            kamen7 = new Kamen((Image)Zdroj.Pozice_zluta);
            kamen7.Jmeno = "Kamen7";
            kamen7.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
            kamen7.TypKarty = typKarty.zatacka;
            kamen7.Artefakt = "Zluta";

            //---------kamen8----------
            kamen8 = new Kamen((Image)Zdroj.Zatacka);
            kamen8.Jmeno = "Kamen8";
            kamen8.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
            kamen8.TypKarty = typKarty.zatacka;
```

```
//---------kamen9----------
kamen9 = new Kamen((Image)Zdroj.Zatacka);
kamen9.Jmeno = "Kamen9";
kamen9.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen9.TypKarty = typKarty.zatacka;

//---------kamen10----------
kamen10 = new Kamen((Image)Zdroj.Zatacka);
kamen10.Jmeno = "Kamen10";
kamen10.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen10.TypKarty = typKarty.zatacka;

//---------kamen11----------
kamen11 = new Kamen((Image)Zdroj.Zatacka);
kamen11.Jmeno = "Kamen11";
kamen11.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen11.TypKarty = typKarty.zatacka;

//---------kamen12----------
kamen12 = new Kamen((Image)Zdroj.Zatacka);
kamen12.Jmeno = "Kamen12";
kamen12.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen12.TypKarty = typKarty.zatacka;

//---------kamen13----------
kamen13 = new Kamen((Image)Zdroj.Zatacka);
kamen13.Jmeno = "Kamen13";
kamen13.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen13.TypKarty = typKarty.zatacka;

//---------kamen14----------
kamen14 = new Kamen((Image)Zdroj.Zatacka);
kamen14.Jmeno = "Kamen14";
kamen14.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen14.TypKarty = typKarty.zatacka;

//---------kamen15----------
kamen15 = new Kamen((Image)Zdroj.Mapa);
kamen15.Jmeno = "Kamen15";
kamen15.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen15.TypKarty = typKarty.krizovatka;
kamen15.Artefakt = "Mapa";

//---------kamen16----------
kamen16 = new Kamen((Image)Zdroj.Rovina);
kamen16.Jmeno = "Kamen16";
kamen16.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen16.TypKarty = typKarty.rovina;

//---------kamen17----------
kamen17 = new Kamen((Image)Zdroj.Koruna);
kamen17.Jmeno = "Kamen17";
kamen17.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen17.TypKarty = typKarty.krizovatka;
kamen17.Artefakt = "Koruna";

//---------kamen18----------
kamen18 = new Kamen((Image)Zdroj.Rovina);
kamen18.Jmeno = "Kamen18";
kamen18.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen18.TypKarty = typKarty.rovina;

//---------kamen19----------
kamen19 = new Kamen((Image)Zdroj.Klice);
kamen19.Jmeno = "Kamen19";
kamen19.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen19.TypKarty = typKarty.krizovatka;
kamen19.Artefakt = "Klice";

//---------kamen20----------
kamen20 = new Kamen((Image)Zdroj.Rovina);
kamen20.Jmeno = "Kamen20";
kamen20.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen20.TypKarty = typKarty.rovina;
```

```
//---------kamen21----------
kamen21 = new Kamen((Image)Zdroj.Lebka);
kamen21.Jmeno = "Kamen21";
kamen21.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen21.TypKarty = typKarty.krizovatka;
kamen21.Artefakt = "Lebka";

//---------kamen22----------
kamen22 = new Kamen((Image)Zdroj.Rovina);
kamen22.Jmeno = "Kamen22";
kamen22.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen22.TypKarty = typKarty.rovina;

//---------kamen23----------
kamen23 = new Kamen((Image)Zdroj.Rovina);
kamen23.Jmeno = "Kamen23";
kamen23.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen23.TypKarty = typKarty.rovina;

//---------kamen24----------
kamen24 = new Kamen((Image)Zdroj.Rovina);
kamen24.Jmeno = "Kamen24";
kamen24.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen24.TypKarty = typKarty.rovina;

//---------kamen25----------
kamen25 = new Kamen((Image)Zdroj.Rovina);
kamen25.Jmeno = "Kamen25";
kamen25.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen25.TypKarty = typKarty.rovina;

//---------kamen26----------
kamen26 = new Kamen((Image)Zdroj.Rovina);
kamen26.Jmeno = "Kamen26";
kamen26.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen26.TypKarty = typKarty.rovina;

//---------kamen27----------
kamen27 = new Kamen((Image)Zdroj.Rovina);
kamen27.Jmeno = "Kamen27";
kamen27.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen27.TypKarty = typKarty.rovina;

//---------kamen28----------
kamen28 = new Kamen((Image)Zdroj.Rovina);
kamen28.Jmeno = "Kamen28";
kamen28.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen28.TypKarty = typKarty.rovina;

//---------kamen29----------
kamen29 = new Kamen((Image)Zdroj.Prsten);
kamen29.Jmeno = "Kamen29";
kamen29.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen29.TypKarty = typKarty.krizovatka;
kamen29.Artefakt = "Prsten";

//---------kamen30----------
kamen30 = new Kamen((Image)Zdroj.Rovina);
kamen30.Jmeno = "Kamen30";
kamen30.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen30.TypKarty = typKarty.rovina;

//---------kamen31----------
kamen31 = new Kamen((Image)Zdroj.Truhla);
kamen31.Jmeno = "Kamen31";
kamen31.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen31.TypKarty = typKarty.krizovatka;
kamen31.Artefakt = "Truhla";

//---------kamen32----------
kamen32 = new Kamen((Image)Zdroj.Krysa);
kamen32.Jmeno = "Kamen32";
kamen32.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen32.TypKarty = typKarty.zatacka;
kamen32.Artefakt = "Krysa";
```

```
//---------kamen33----------
kamen33 = new Kamen((Image)Zdroj.Diamand);
kamen33.Jmeno = "Kamen33";
kamen33.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen33.TypKarty = typKarty.krizovatka;
kamen33.Artefakt = "Diamand";

//---------kamen34----------
kamen34 = new Kamen((Image)Zdroj.Chrobak);
kamen34.Jmeno = "Kamen34";
kamen34.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen34.TypKarty = typKarty.zatacka;
kamen34.Artefakt = "Chrobak";

//---------kamen35----------
kamen35 = new Kamen((Image)Zdroj.Meč);
kamen35.Jmeno = "Kamen35";
kamen35.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen35.TypKarty = typKarty.krizovatka;
kamen35.Artefakt = "Mec";

//---------kamen36----------
kamen36 = new Kamen((Image)Zdroj.Pavouk);
kamen36.Jmeno = "Kamen36";
kamen36.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen36.TypKarty = typKarty.zatacka;
kamen36.Artefakt = "Pavouk";

//---------kamen37----------
kamen37 = new Kamen((Image)Zdroj.Kubec);
kamen37.Jmeno = "Kamen37";
kamen37.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen37.TypKarty = typKarty.krizovatka;
kamen37.Artefakt = "Kubec";

//---------kamen38----------
kamen38 = new Kamen((Image)Zdroj.Netopyr);
kamen38.Jmeno = "Kamen38";
kamen38.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen38.TypKarty = typKarty.krizovatka;
kamen38.Artefakt = "Netopyr";

//---------kamen39----------
kamen39 = new Kamen((Image)Zdroj.Mlok);
kamen39.Jmeno = "Kamen39";
kamen39.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen39.TypKarty = typKarty.zatacka;
kamen39.Artefakt = "Mlok";

//---------kamen40----------
kamen40 = new Kamen((Image)Zdroj.Mura);
kamen40.Jmeno = "Kamen40";
kamen40.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen40.TypKarty = typKarty.zatacka;
kamen40.Artefakt = "Mura";

//---------kamen41----------
kamen41 = new Kamen((Image)Zdroj.Vila);
kamen41.Jmeno = "Kamen41";
kamen41.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen41.TypKarty = typKarty.krizovatka;
kamen41.Artefakt = "Vila";

//---------kamen42----------
kamen42 = new Kamen((Image)Zdroj.Sova);
kamen42.Jmeno = "Kamen42";
kamen42.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen42.TypKarty = typKarty.zatacka;
kamen42.Artefakt = "Sova";

//---------kamen43----------
kamen43 = new Kamen((Image)Zdroj.Pozice_zelena);
kamen43.Jmeno = "Kamen43";
kamen43.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen43.TypKarty = typKarty.zatacka;
kamen43.Artefakt = "Zelena";
```

```
//---------kamen44----------
kamen44 = new Kamen((Image)Zdroj.Drak);
kamen44.Jmeno = "Kamen44";
kamen44.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen44.TypKarty = typKarty.krizovatka;
kamen44.Artefakt = "Drak";

//---------kamen45----------
kamen45 = new Kamen((Image)Zdroj.Svícen);
kamen45.Jmeno = "Kamen45";
kamen45.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen45.TypKarty = typKarty.krizovatka;
kamen45.Artefakt = "Svicen";

//---------kamen46----------
kamen46 = new Kamen((Image)Zdroj.Duch1);
kamen46.Jmeno = "Kamen46";
kamen46.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen46.TypKarty = typKarty.krizovatka;
kamen46.Artefakt = "Duch1";

//---------kamen47----------
kamen47 = new Kamen((Image)Zdroj.Helma);
kamen47.Jmeno = "Kamen47";
kamen47.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen47.TypKarty = typKarty.krizovatka;
kamen47.Artefakt = "Helma";

//---------kamen48----------
kamen48 = new Kamen((Image)Zdroj.Duch2);
kamen48.Jmeno = "Kamen48";
kamen48.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen48.TypKarty = typKarty.krizovatka;
kamen48.Artefakt = "Duch2";

//---------kamen49----------
kamen49 = new Kamen((Image)Zdroj.Pozice_modra);
kamen49.Jmeno = "Kamen49";
kamen49.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen49.TypKarty = typKarty.zatacka;
kamen49.Artefakt = "Modra";

//---------kamen50----------
kamen50 = new Kamen((Image)Zdroj.Rovina);
kamen50.Jmeno = "Kamen50";
kamen50.Kliknuto += new KliknutiHandler(kamen_Kliknuto);
kamen50.TypKarty = typKarty.rovina;

//------karta1-----------
karta1 = new HraciKarta((Image)Zdroj.Karta_diamand);
karta1.Jmeno = "Karta1";
karta1.Pozice = poziceKarty;
karta1.Rozmery = velikostKarty;
karta1.Artefakt = "Diamand";
//-----karta2--------
karta2 = new HraciKarta((Image)Zdroj.Karta_drak);
karta2.Jmeno = "Karta2";
karta2.Pozice = poziceKarty;
karta2.Rozmery = velikostKarty;
karta2.Artefakt = "Drak";
//-----karta3--------
karta3 = new HraciKarta((Image)Zdroj.Karta_duch_01);
karta3.Jmeno = "Karta3";
karta3.Pozice = poziceKarty;
karta3.Rozmery = velikostKarty;
karta3.Artefakt = "Duch1";
//-----karta4--------
karta4 = new HraciKarta((Image)Zdroj.Karta_duch_02);
karta4.Jmeno = "Karta4";
karta4.Pozice = poziceKarty;
karta4.Rozmery = velikostKarty;
karta4.Artefakt = "Duch2";
//-----karta5--------
karta5 = new HraciKarta((Image)Zdroj.Karta_helma);
karta5.Jmeno = "Karta5";
```

```
karta5.Pozice = poziceKarty;
karta5.Rozmery = velikostKarty;
karta5.Artefakt = "Helma";
//-----karta6--------
karta6 = new HraciKarta((Image)Zdroj.Karta_chrobak);
karta6.Jmeno = "Karta6";
karta6.Pozice = poziceKarty;
karta6.Rozmery = velikostKarty;
karta6.Artefakt = "Chrobak";
//-----karta7--------
karta7 = new HraciKarta((Image)Zdroj.Karta_klice);
karta7.Jmeno = "Karta7";
karta7.Pozice = poziceKarty;
karta7.Rozmery = velikostKarty;
karta7.Artefakt = "Klice";
//-----karta8--------
karta8 = new HraciKarta((Image)Zdroj.Karta_kniha);
karta8.Jmeno = "Karta8";
karta8.Pozice = poziceKarty;
karta8.Rozmery = velikostKarty;
karta8.Artefakt = "Kniha";
//-----karta9--------
karta9 = new HraciKarta((Image)Zdroj.Karta_koruna);
karta9.Jmeno = "Karta9";
karta9.Pozice = poziceKarty;
karta9.Rozmery = velikostKarty;
karta9.Artefakt = "Koruna";
//-----karta10--------
karta10 = new HraciKarta((Image)Zdroj.Karta_krysa);
karta10.Jmeno = "Karta10";
karta10.Pozice = poziceKarty;
karta10.Rozmery = velikostKarty;
karta10.Artefakt = "Krysa";
//-----karta11--------
karta11 = new HraciKarta((Image)Zdroj.Karta_kubec);
karta11.Jmeno = "Karta11";
karta11.Pozice = poziceKarty;
karta11.Rozmery = velikostKarty;
karta11.Artefakt = "Kubec";
//-----karta12--------
karta12 = new HraciKarta((Image)Zdroj.Karta_lebka);
karta12.Jmeno = "Karta12";
karta12.Pozice = poziceKarty;
karta12.Rozmery = velikostKarty;
karta12.Artefakt = "Lebka";
//-----karta13--------
karta13 = new HraciKarta((Image)Zdroj.Karta_mapa);
karta13.Jmeno = "Karta13";
karta13.Pozice = poziceKarty;
karta13.Rozmery = velikostKarty;
karta13.Artefakt = "Mapa";
//-----karta14--------
karta14 = new HraciKarta((Image)Zdroj.Karta_mec);
karta14.Jmeno = "Karta14";
karta14.Pozice = poziceKarty;
karta14.Rozmery = velikostKarty;
karta14.Artefakt = "Mec";
//-----karta15--------
karta15 = new HraciKarta((Image)Zdroj.Karta_mesec);
karta15.Jmeno = "Karta15";
karta15.Pozice = poziceKarty;
karta15.Rozmery = velikostKarty;
karta15.Artefakt = "Mesec";
//-----karta16--------
karta16 = new HraciKarta((Image)Zdroj.Karta_mlok);
karta16.Jmeno = "Karta16";
karta16.Pozice = poziceKarty;
karta16.Rozmery = velikostKarty;
karta16.Artefakt = "Mlok";
//-----karta17--------
karta17 = new HraciKarta((Image)Zdroj.Karta_mura);
karta17.Jmeno = "Karta17";
karta17.Pozice = poziceKarty;
karta17.Rozmery = velikostKarty;
karta17.Artefakt = "Mura";
//-----karta18--------
```

```
karta18 = new HraciKarta((Image)Zdroj.Karta_netopyr);
karta18.Jmeno = "Karta18";
karta18.Pozice = poziceKarty;
karta18.Rozmery = velikostKarty;
karta18.Artefakt = "Netopyr";
//-----karta19--------
karta19 = new HraciKarta((Image)Zdroj.Karta_pavouk);
karta19.Jmeno = "Karta19";
karta19.Pozice = poziceKarty;
karta19.Rozmery = velikostKarty;
karta19.Artefakt = "Pavouk";
//-----karta20--------
karta20 = new HraciKarta((Image)Zdroj.Karta_prsten);
karta20.Jmeno = "Karta20";
karta20.Pozice = poziceKarty;
karta20.Rozmery = velikostKarty;
karta20.Artefakt = "Prsten";
//-----karta21--------
karta21 = new HraciKarta((Image)Zdroj.Karta_sova);
karta21.Jmeno = "Karta21";
karta21.Pozice = poziceKarty;
karta21.Rozmery = velikostKarty;
karta21.Artefakt = "Sova";
//-----karta22--------
karta22 = new HraciKarta((Image)Zdroj.Karta_svicen);
karta22.Jmeno = "Karta22";
karta22.Pozice = poziceKarty;
karta22.Rozmery = velikostKarty;
karta22.Artefakt = "Svicen";
//-----karta23--------
karta23 = new HraciKarta((Image)Zdroj.Karta_truhla);
karta23.Jmeno = "Karta23";
karta23.Pozice = poziceKarty;
karta23.Rozmery = velikostKarty;
karta23.Artefakt = "Truhla";
//-----karta24--------
karta24 = new HraciKarta((Image)Zdroj.Karta_vila);
karta24.Jmeno = "Karta24";
karta24.Pozice = poziceKarty;
karta24.Rozmery = velikostKarty;
karta24.Artefakt = "Vila";

hraciKartaCervena = new HraciKarta((Image)Zdroj.Karta_Pozice_red);
hraciKartaCervena.Jmeno = "Vychozi cervena";
hraciKartaCervena.Pozice = poziceKarty;
hraciKartaCervena.Rozmery = velikostKarty;
hraciKartaCervena.Artefakt = "Cervena";

hraciKartaModra = new HraciKarta((Image)Zdroj.Karta_Pozice_blue);
hraciKartaModra.Jmeno = "Vychozi modra";
hraciKartaModra.Pozice = poziceKarty;
hraciKartaModra.Rozmery = velikostKarty;
hraciKartaModra.Artefakt = "Modra";

hraciKartaZluta = new HraciKarta((Image)Zdroj.Karta_Pozice_yellow);
hraciKartaZluta.Jmeno = "Vychozi zluta";
hraciKartaZluta.Pozice = poziceKarty;
hraciKartaZluta.Rozmery = velikostKarty;
hraciKartaZluta.Artefakt = "Zluta";

hraciKartaZelena = new HraciKarta((Image)Zdroj.Karta_Pozice_green);
hraciKartaZelena.Jmeno = "Vychozi zelena";
hraciKartaZelena.Pozice = poziceKarty;
hraciKartaZelena.Rozmery = velikostKarty;
hraciKartaZelena.Artefakt = "Zelena";

otaznik = new HraciKarta((Image)Zdroj.otaznik);
otaznik.Jmeno = "Otaznik";
otaznik.Pozice = poziceKarty;
otaznik.Rozmery = velikostKarty;
otaznik.Artefakt = "Otaznik";

hraciKartyVychozi = new HraciKarta[] { hraciKartaCervena, hraciKartaModra, hraciKartaZelena, hraciKartaZluta, otaznik };
```

```
            //--------formular----------
            this.Text = "Labyrint";
            this.DoubleBuffered = true;//pri vice obrazcich k vykresleni, je nejprve vykresluje do pameti a az vysledny obrazek vykresli na
form - 1x
            this.BackColor = Color.FromArgb(24, 255, 0);
            this.TransparencyKey = Color.FromArgb(24, 255, 0);
            this.FormBorderStyle = FormBorderStyle.None;
            this.KeyDown += new KeyEventHandler(HraciForm_KeyDown);
            this.Click += new EventHandler(HraciForm_Click);
            this.MouseMove += new MouseEventHandler(HraciForm_MouseMove);
            this.MouseDown += new MouseEventHandler(HraciForm_MouseDown);
            this.Load += new EventHandler(HraciForm_Load);
            this.MouseUp += new MouseEventHandler(HraciForm_MouseUp);

            //-------minimalizace----
            minimalizace = new GrafickyObjekt((Image)Zdroj.MinimalizaceBila);
            minimalizace.Jmeno = "minimalizace";
            minimalizace.StisknutoMysi += new StisknutoHandler(minimalizace_StisknutoMysi);
            minimalizace.OpusteniMysi += new OpusteniHandler(minimalizace_OpusteniMysi);
            minimalizace.PrejetoMysi += new PrejetiNadHandler(minimalizace_PrejetoMysi);
            minimalizace.UvolnenoMysi += new UvolnenoHandler(minimalizace_UvolnenoMysi);
            minimalizace.Pozice = souradMinimalizace;
            minimalizace.Rozmery = new Size(30, 30);
            //-------zavreni--------
            zavreni = new GrafickyObjekt((Image)Zdroj.KrizekBily);
            zavreni.Jmeno = "zavreni";
            zavreni.PrejetoMysi += new PrejetiNadHandler(zavreni_PrejetoMysi);
            zavreni.StisknutoMysi += new StisknutoHandler(zavreni_StisknutoMysi);
            zavreni.OpusteniMysi += new OpusteniHandler(zavreni_OpusteniMysi);
            zavreni.UvolnenoMysi += new UvolnenoHandler(zavreni_UvolnenoMysi);
            zavreni.Rozmery = new Size(30, 30);
            //-------dalsi hrac----
            dalsiHrac = new GrafickyObjekt();//
            dalsiHrac.Jmeno = "dalsiHrac";
            dalsiHrac.Rozmery = new Size(30, 30);
            //------cas-------
            //souradnice nacitam pri nacteni formulare v udalosti PrizpusobRozliseni
            cas = new Text(this);
            cas.FontFamily = sadaFontu.Families[0];
            cas.Jmeno = "cas";
            cas.FontSize = 20f;
            cas.TextovyObsah = "2:00";
            //-----titulky pro hrace, jako info
            hledanyTitulek = new Text(this);
            hledanyTitulek.FontFamily = sadaFontu.Families[0];
            hledanyTitulek.Jmeno = "hledanyTitulek";
            hledanyTitulek.FontSize = 12f;
            hledanyTitulek.TextovyObsah = "Hledaný artefakt";

            nalezenTitulek = new Text(this);
            nalezenTitulek.FontFamily = sadaFontu.Families[0];
            nalezenTitulek.Jmeno = "nalezenTitulek";
            nalezenTitulek.FontSize = 11f;
            nalezenTitulek.TextovyObsah = "Nalezený artefakt";

            nalezenoTitulek = new Text(this);
            nalezenoTitulek.FontFamily = sadaFontu.Families[0];
            nalezenoTitulek.Jmeno = "nalezenoTitulek";
            nalezenoTitulek.FontSize = 10f;
            nalezenoTitulek.TextovyObsah = "Nalezeno";

            pocetTahuTitulek = new Text(this);
            pocetTahuTitulek.FontFamily = sadaFontu.Families[0];
            pocetTahuTitulek.Jmeno = "pocetTahuTitulek";
            pocetTahuTitulek.FontSize = 10f;
            pocetTahuTitulek.TextovyObsah = "Počet tahů";


            hracTitulek = new Text(this);
            hracTitulek.FontFamily = sadaFontu.Families[0];
            hracTitulek.Jmeno = "hracTitulek";
            hracTitulek.FontSize = 12f;
            hracTitulek.TextovyObsah = "forstec";//omezeno na 10 znaku

            nalezenoKolik = new Text(this);
```

```
nalezenoKolik.FontFamily = sadaFontu.Families[0];
nalezenoKolik.Jmeno = "nalezenoKolik";
nalezenoKolik.FontSize = 10f;
nalezenoKolik.TextovyObsah = "8 z 1";

pocetTahu = new Text(this);
pocetTahu.FontFamily = sadaFontu.Families[0];
pocetTahu.Jmeno = "pocetTahu";
pocetTahu.FontSize = 10f;
pocetTahu.TextovyObsah = "100";

sipkaLevaNick = new GrafickyObjekt((Image)Zdroj.LevaSipkaBila);
sipkaLevaNick.Jmeno = "sipkaLevaNick";
sipkaLevaNick.Rozmery = new Size(20, 20);
sipkaLevaNick.OpusteniMysi += new OpusteniHandler(sipkaLevaNick_OpusteniMysi);
sipkaLevaNick.PrejetoMysi += new PrejetiNadHandler(sipkaLevaNick_PrejetoMysi);
sipkaLevaNick.StisknutoMysi += new StisknutoHandler(sipkaLevaNick_StisknutoMysi);
sipkaLevaNick.UvolnenoMysi += new UvolnenoHandler(sipkaLevaNick_UvolnenoMysi);

sipkaPravaNick = new GrafickyObjekt((Image)Zdroj.PravaSipkaBila);
sipkaPravaNick.Jmeno = "sipkaPravaNick";
sipkaPravaNick.Rozmery = new Size(20, 20);
sipkaPravaNick.OpusteniMysi += new OpusteniHandler(sipkaPravaNick_OpusteniMysi);
sipkaPravaNick.PrejetoMysi += new PrejetiNadHandler(sipkaPravaNick_PrejetoMysi);
sipkaPravaNick.StisknutoMysi += new StisknutoHandler(sipkaPravaNick_StisknutoMysi);
sipkaPravaNick.UvolnenoMysi += new UvolnenoHandler(sipkaPravaNick_UvolnenoMysi);

sipkaLevaNalezeno = new GrafickyObjekt((Image)Zdroj.LevaSipkaBila);
sipkaLevaNalezeno.Jmeno = "sipkaLevaNalezeno";
sipkaLevaNalezeno.Rozmery = new Size(20, 20);
sipkaLevaNalezeno.OpusteniMysi += new OpusteniHandler(sipkaLevaNalezeno_OpusteniMysi);
sipkaLevaNalezeno.PrejetoMysi += new PrejetiNadHandler(sipkaLevaNalezeno_PrejetoMysi);
sipkaLevaNalezeno.StisknutoMysi += new StisknutoHandler(sipkaLevaNalezeno_StisknutoMysi);
sipkaLevaNalezeno.UvolnenoMysi += new UvolnenoHandler(sipkaLevaNalezeno_UvolnenoMysi);

sipkaPravaNalezeno = new GrafickyObjekt((Image)Zdroj.PravaSipkaBila);
sipkaPravaNalezeno.Jmeno = "sipkaPravaNalezeno";
sipkaPravaNalezeno.Rozmery = new Size(20, 20);
sipkaPravaNalezeno.OpusteniMysi += new OpusteniHandler(sipkaPravaNalezeno_OpusteniMysi);
sipkaPravaNalezeno.PrejetoMysi += new PrejetiNadHandler(sipkaPravaNalezeno_PrejetoMysi);
sipkaPravaNalezeno.StisknutoMysi += new StisknutoHandler(sipkaPravaNalezeno_StisknutoMysi);
sipkaPravaNalezeno.UvolnenoMysi += new UvolnenoHandler(sipkaPravaNalezeno_UvolnenoMysi);

prohledaniNalezenych = new Text(this);
prohledaniNalezenych.FontFamily = sadaFontu.Families[0];
prohledaniNalezenych.Jmeno = "prohledaniNalezenych";
prohledaniNalezenych.FontSize = 10f;
prohledaniNalezenych.TextovyObsah = "12/12";

//---------prirazeni do pole pro vykresleni----------
prvkyNaFormu = new GrafickyObjekt[67];
prvkyNaFormu[0] = plocha;
prvkyNaFormu[1] = kamen1;//vyuziti polymorfismu v praxi :-))
prvkyNaFormu[2] = kamen2;
prvkyNaFormu[3] = kamen3;
prvkyNaFormu[4] = kamen4;
prvkyNaFormu[5] = kamen5;
prvkyNaFormu[6] = kamen6;
prvkyNaFormu[7] = kamen7;
prvkyNaFormu[8] = kamen8;
prvkyNaFormu[9] = kamen9;
prvkyNaFormu[10] = kamen10;
prvkyNaFormu[11] = kamen11;
prvkyNaFormu[12] = kamen12;
prvkyNaFormu[13] = kamen13;
prvkyNaFormu[14] = kamen14;
prvkyNaFormu[15] = kamen15;
prvkyNaFormu[16] = kamen16;
prvkyNaFormu[17] = kamen17;
prvkyNaFormu[18] = kamen18;
prvkyNaFormu[19] = kamen19;
prvkyNaFormu[20] = kamen20;
prvkyNaFormu[21] = kamen21;
prvkyNaFormu[22] = kamen22;
prvkyNaFormu[23] = kamen23;
prvkyNaFormu[24] = kamen24;
```

```
    prvkyNaFormu[25] = kamen25;
    prvkyNaFormu[26] = kamen26;
    prvkyNaFormu[27] = kamen27;
    prvkyNaFormu[28] = kamen28;
    prvkyNaFormu[29] = kamen29;
    prvkyNaFormu[30] = kamen30;
    prvkyNaFormu[31] = kamen31;
    prvkyNaFormu[32] = kamen32;
    prvkyNaFormu[33] = kamen33;
    prvkyNaFormu[34] = kamen34;
    prvkyNaFormu[35] = kamen35;
    prvkyNaFormu[36] = kamen36;
    prvkyNaFormu[37] = kamen37;
    prvkyNaFormu[38] = kamen38;
    prvkyNaFormu[39] = kamen39;
    prvkyNaFormu[40] = kamen40;
    prvkyNaFormu[41] = kamen41;
    prvkyNaFormu[42] = kamen42;
    prvkyNaFormu[43] = kamen43;
    prvkyNaFormu[44] = kamen44;
    prvkyNaFormu[45] = kamen45;
    prvkyNaFormu[46] = kamen46;
    prvkyNaFormu[47] = kamen47;
    prvkyNaFormu[48] = kamen48;
    prvkyNaFormu[49] = kamen49;
    prvkyNaFormu[50] = kamen50;
    prvkyNaFormu[51] = cas;
    prvkyNaFormu[52] = hledanyTitulek;
    prvkyNaFormu[53] = nalezenTitulek;
    prvkyNaFormu[54] = nalezenoTitulek;
    prvkyNaFormu[55] = nalezenoKolik;
    prvkyNaFormu[56] = pocetTahuTitulek;
    prvkyNaFormu[57] = hracTitulek;
    prvkyNaFormu[58] = pocetTahu;
    prvkyNaFormu[59] = minimalizace;
    prvkyNaFormu[60] = zavreni;
    prvkyNaFormu[61] = dalsiHrac;
    prvkyNaFormu[62] = sipkaLevaNick;
    prvkyNaFormu[63] = sipkaPravaNick;
    prvkyNaFormu[64] = sipkaPravaNalezeno;
    prvkyNaFormu[65] = sipkaLevaNalezeno;
    prvkyNaFormu[66] = prohledaniNalezenych;
    kontrols[0] = label;

    //-------prirazeni do pole hracich karet----------
    hraciKarty = new HraciKarta[24];
    hraciKarty[0] = karta1;
    hraciKarty[1] = karta2;
    hraciKarty[2] = karta3;
    hraciKarty[3] = karta4;
    hraciKarty[4] = karta5;
    hraciKarty[5] = karta6;
    hraciKarty[6] = karta7;
    hraciKarty[7] = karta8;
    hraciKarty[8] = karta9;
    hraciKarty[9] = karta10;
    hraciKarty[10] = karta11;
    hraciKarty[11] = karta12;
    hraciKarty[12] = karta13;
    hraciKarty[13] = karta14;
    hraciKarty[14] = karta15;
    hraciKarty[15] = karta16;
    hraciKarty[16] = karta17;
    hraciKarty[17] = karta18;
    hraciKarty[18] = karta19;
    hraciKarty[19] = karta20;
    hraciKarty[20] = karta21;
    hraciKarty[21] = karta22;
    hraciKarty[22] = karta23;
    hraciKarty[23] = karta24;

    casovac.Interval = 40;
    casovac.Elapsed += new ElapsedEventHandler(casovac_Elapsed);
}
```

```
}

public class Text : GrafickyObjekt
{
    public Text(Form Form)
        : base()
    {
        this.cisloFontu = inter.CisloFontu;
        this.grafika = Form.CreateGraphics();
        this.form = Form;
        this.fontSize = 14f;
        this.barva = Brushes.White;
    }
    private string text = "";
    private float fontSize;
    public float FontSize
    {
        get
        {
            return fontSize;
        }
        set
        {
            fontSize = value;
            this.font = new Font(fontFamily, fontSize);
        }
    }
    public string TextovyObsah
    {
        get
        {
            return text;
        }
        set
        {
            text = value;
        }
    }
    private Font font;
    public Font Font
    {
        get
        {
            return font;
        }
    }
    private FontFamily fontFamily;
    public FontFamily FontFamily
    {
        get
        {
            return fontFamily;
        }
        set
        {
            fontFamily = value;
            font = new Font(fontFamily, fontSize);
        }
    }
    private int cisloFontu;
    private Graphics grafika;
    private Form form;
    private Brush barva;
    public Brush Barva
    {
        get
        {
            return barva;
        }
        set
        {
            barva = value;
        }
    }
    public override void Vykresli(Graphics gr)
```

```
            {
                //vykreslim text
                gr.DrawString(this.text, font, barva, this.pozice);
            }
        }
    }
```

C8 HRACIFORM-UDALOSTI

```
    using System;
    using System.Windows.Forms;
    using System.Drawing;
    using System.Timers;
    using System.Threading;

    namespace Labyrint
    {
        delegate void VolajiciDelegat();
        partial class HraciForm
        {
            private bool stisknutoLeve = false;
            private MouseButtons tlacitkoMysi;//udava, jake tlacitko bylo stisknuto
            public bool hracPosunul = false;//hrac jeste nezasunul - indikuje, zda hrac jiz posunul kamen. Nejdrive se MUSI posunout, pote az
posunout figurku

            //slouzi pro ukladani povolenych rozmezi souradnic pro pohyb posuvneho kamene po formu
            public Point[] pocatecniPovolene;
            public Point[] koncovePovolene;
            Figurka[] proPosunuti = null;
            System.Timers.Timer casovac = new System.Timers.Timer();

            //zjisteni kliknuti na kamen
            //pracuje s: prvkyNaFormu, tlacitkoMysi
            void HraciForm_Click(object sender, EventArgs e)
            {
                //klikani je povoleno pouze hraci v dobe, kdy hraje, nebo kdyz hra skoncila.
                if ((hra.hrajeSe && hra.AktualniHrac.Ai==false) || hra.hrajeSe==false)
                {
                    MouseEventArgs udalostMysi = (MouseEventArgs)e;
                    //projde vsechny prvky, ktere jsou vykresleny na formu, vcetne plochy

                    for (int i = 0; i < this.prvkyNaFormu.Length; i++)
                    {

                        if (this.prvkyNaFormu[i] != null)
                        {
                            if (this.prvkyNaFormu[i].Jmeno == "Plocha")//kontroluje se plocha, takze ji preskocim
                            {
                            }
                            //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
                            else if (udalostMysi.Location.X >= this.prvkyNaFormu[i].Pozice.X && udalostMysi.Location.X <=
this.prvkyNaFormu[i].Pozice.X + this.prvkyNaFormu[i].Rozmery.Width &&
                                udalostMysi.Location.Y >= this.prvkyNaFormu[i].Pozice.Y && udalostMysi.Location.Y <=
this.prvkyNaFormu[i].Pozice.Y + this.prvkyNaFormu[i].Rozmery.Height)
                            {
                                //spusti udalost danneho objektu

                                this.tlacitkoMysi = udalostMysi.Button;
                                this.prvkyNaFormu[i].SpoustecUdalosti(prvkyNaFormu[i]);
                                break;
                            }
                        }

                    }
                }
            }


            //obsluha udalosti
            //pracuje s: Volajici, Posunovaci, tlacitkoMysi, poleSouradnicX, poleSouradnicY, predchoziSloupecRadek, predchoziCislo,
predchoziSmer
            //pracovniPole, hra.AktualniHrac.Figurka.AktualniKamen.VPracovnimPoli
            //Posunout(), hra.kameny, hra.AktualizujPropojeniCest(),
```

```
        void kamen_Kliknuto(object Volajici)
        {
            if (hra.hrajeSe)
            {
                Kamen volajici = (Kamen)Volajici;

                if (volajici == this.Posunovaci)//jedna se o posunovaci kartu
                {

                    if (this.tlacitkoMysi == MouseButtons.Left)//na posunovaci kamen jsem kliknul levym tlacitkem mysi, cili ho bud chci
aktivovat nebo zasunout
                    {
                        if (this.stisknutoLeve == true)//tlacitko bylo uz predtim zmacknuto, takze ho chci zasunout
                        {

                            this.stisknutoLeve = false;//karta se deaktivuje

                            //posunout cely radek nebo sloupec karet
                            //jestlize se X souradnice karticky shoduji s PoleSouradnicX, tak se jedna o posunuti sloupce, protoze se nachazi ve
sloupci, podle toho, o jake souradnice se jedna, poznam, o jaky sloupec se jedna
                            //souradnice jsou preden dabe, protoze mohou nabyvat jenom predem definovanych hodnot
                            //ulozeni aktualnich soradnic
                            int x = Posunovaci.Pozice.X;
                            int y = Posunovaci.Pozice.Y;
                            int radeksloupec = 0, smer = 0, cislo = 0;//potrebna data pro posunuti, ktera musim ziskat

                            for (int i = 0; i < this.poleSouradnicX.Length; i++)//projdu iksove pole
                            {

                                //pevne sloupce a radky nemusim testovat, protoze se posouvat nebudou, tudiz nemusim znat data pro posunovani
                                if (i != 0 && i != 2 && i != 4 && i != 6)//kdyz se nejedna o pevna policka
                                {
                                    //kamen chci zasunout do spravneho radku nebo sloupce
                                    //kdyz je kamen nalevo nebo napravo, tak ma X souradnice mensi nebo vetsi nez krajni souradnice plochy
                                    //stejne je to dole a nahore s Y
                                    if (this.poleSouradnicX[i] == x)//x souradnice se nachazi v poli souradnic, takze je pod nebo nad hraci plochou
                                    {
                                        //vyhodit z cyklu
                                        radeksloupec = 2;//sloupec
                                        cislo = i;//cislo sloupce pro posun
                                        if (y > this.poleSouradnicY[this.poleSouradnicY.Length - 1])//nachazi se dole
                                        {
                                            smer = 2;//posune nahoru
                                        }
                                        else
                                        {
                                            smer = 1;//posune dolu
                                        }

                                        if (this.predchoziSloupecRadek == radeksloupec && this.predchoziCislo == cislo && this.predchoziSmer !=
smer)//snazi se zasunout na stejne misto
                                        {
                                            radeksloupec = 0;
                                        }
                                        else
                                        {
                                            this.predchoziSloupecRadek = radeksloupec;
                                            this.predchoziCislo = cislo;
                                            this.predchoziSmer = smer;
                                        }
                                        break;
                                    }
                                    else if (this.poleSouradnicY[i] == y)//y souradnice se nachazi v poli souradnic, takze je vlevo nebo vpravo od
hraci plochy
                                    {

                                        //vyhodit z cyklu
                                        radeksloupec = 1;//radek
                                        cislo = i;//cislo radku pro posun
                                        if (x > this.poleSouradnicX[this.poleSouradnicX.Length - 1])//nachazi se napravo
                                        {
                                            smer = 1;//doleva
                                        }
                                        else
                                        {
                                            smer = 2;//doprava
```

```
                        }

                        if (this.predchoziSloupecRadek == radeksloupec && this.predchoziCislo == cislo && this.predchoziSmer !=
smer)
                        {
                            radeksloupec = 0;
                        }
                        else
                        {
                            this.predchoziSloupecRadek = radeksloupec;
                            this.predchoziCislo = cislo;
                            this.predchoziSmer = smer;
                        }
                        break;
                    }
                }
            }

            //posunu
            if (radeksloupec != 0)//znamena to, ze uzivatel se chystal zasunout karicku skutecne do hraci plochy. Kdyby byla nula,
tak se souradnice v polich nenasly, tudiz kliknuti bylo mimo povoleny rozsah
            {
                if (hracPosunul == false)//hrac jeste nezasunul
                {
                    hracPosunul = true;//hrac uz zasunul

                    //posune se v pameti
                    this.Posunout(radeksloupec, smer, cislo, Posunovaci);

                    //kdyz figurka hrace stoji na kameni, ktery se stal posunovacim, znamena to, ze je mimo hraci plochu
                    //proto musim figurku prehodit na opacnou stranu plochy
                    //u kazdeho hrace zkontroluju, zda figurka nestoji na posunovacim kameni
                    //kdyz ano, prehodim ji, kdyz ne, nic neudelam
                    for (int hrac = 0; hrac < hra.Hraci.Length; hrac++)
                    {
                        if (hra.Hraci[hrac].Figurka.AktualniKamen == Posunovaci)//figurka stoji na kameni, ktery je mimo hraci
plochu
                        {
                            //Posunuvaci si pamatuje souradnice v pracovnim poli. Pred posunutim na techto souradnicich stal, ale kdyz
byl posunut
                            //tyto hodnoty se mu neprepsaly,tudiz jsou porad stejne, i kdyz uz na nich nestoji
                            if (Posunovaci.VPracovnimPoli == new Point(1, 0))//na levem okraji na druhem radku
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[1, 6];
                            }
                            else if (Posunovaci.VPracovnimPoli == new Point(3, 0))//na levem okraji na ctvrtem radku
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[3, 6];
                            }
                            else if (Posunovaci.VPracovnimPoli == new Point(5, 0))//na levem okraji na sestem radku
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[5, 6];
                            }
                            else if (Posunovaci.VPracovnimPoli == new Point(1, 6))//na pravem okraji na druhem radku
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[1, 0];
                            }
                            else if (Posunovaci.VPracovnimPoli == new Point(3, 6))//na pravem okraji na ctvrtem radku
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[3, 0];
                            }
                            else if (Posunovaci.VPracovnimPoli == new Point(5, 6))//na pravem okraji na sestem radku
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[5, 0];
                            }
                            else if (Posunovaci.VPracovnimPoli == new Point(0, 1))//na hornim okraji na druhem sloupci
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[6, 1];
                            }
                            else if (Posunovaci.VPracovnimPoli == new Point(0, 3))//na hornim okraji na ctvrtem sloupci
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[6, 3];
                            }
                            else if (Posunovaci.VPracovnimPoli == new Point(0, 5))//na hornim okraji na sestem sloupci
                            {
                                hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[6, 5];
```

```
                }
                else if (Posunovaci.VPracovnimPoli == new Point(6, 1))//na dolnim okraji na druhem sloupci
                {
                    hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[0, 1];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(6, 3))//na dolnim okraji na ctvrtem sloupci
                {
                    hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[0, 3];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(6, 5))//na dolnim okraji na sestem sloupci
                {
                    hra.Hraci[hrac].Figurka.AktualniKamen = PracovniPole[0, 5];
                }
            }

            if (hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen == Posunovaci)//figurka stoji na kameni, ktery je mimo
hraci plochu
            {
                //Posunovaci si pamatuje souradnice v pracovnim poli. Pred posunutim na techto souradnicich stal, ale kdyz
byl posunut
                //tyto hodnoty se mu neprepsaly,tudiz jsou porad stejne, i kdyz uz na nich nestoji
                if (Posunovaci.VPracovnimPoli == new Point(1, 0))//na levem okraji na druhem radku
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[1, 6];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(3, 0))//na levem okraji na ctvrtem radku
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[3, 6];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(5, 0))//na levem okraji na sestem radku
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[5, 6];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(1, 6))//na pravem okraji na druhem radku
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[1, 0];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(3, 6))//na pravem okraji na ctvrtem radku
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[3, 0];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(5, 6))//na pravem okraji na sestem radku
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[5, 0];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(0, 1))//na hornim okraji na druhem sloupci
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[6, 1];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(0, 3))//na hornim okraji na ctvrtem sloupci
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[6, 3];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(0, 5))//na hornim okraji na sestem sloupci
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[6, 5];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(6, 1))//na dolnim okraji na druhem sloupci
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[0, 1];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(6, 3))//na dolnim okraji na ctvrtem sloupci
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[0, 3];
                }
                else if (Posunovaci.VPracovnimPoli == new Point(6, 5))//na dolnim okraji na sestem sloupci
                {
                    hra.Hraci[hrac].NeaktivniFigurka.AktualniKamen = PracovniPole[0, 5];
                }
            }
        }

        //aktualizovane pracovni pole prekopiruju do hry
        hra.Kameny = PracovniPole;
        hra.AktualizujPropojeniCest();
```

```
                        //pred posunutim kamenu musim zjistit, zda se na nekterem z nich nenachazi nejaka figurka. Proto musim projit
vsechny figurky a zjistit kamen, na kterem stoji
                        //figurku, jejiz kamen se shoduje s prave posunovanym, ulozim do pole figurek, ktere budou posunovany-muze
jich byt vice
                        //kdyz se figurka po posunuti vyskytne vne hraci plochy, musi se presunout na opacny konec hraci plochy
                        //v pameti uz ma figurka nastaven aktualni kamen na opacnou stranu, tedy pokud stala na kraji.
                        //v nasledujicim kodu ovsem hledam v poli, ve kterem je obsazen i kamen, ktery se vsunuje.
                        //takze figurka ma aktualni kamen nastaven uz spravne, ale tady pracuju s jejimi fyzickymi souradnicemi, takze ji
posunu s kartickou, na ktere stoji
                        //ovsem jenom jako, protoze ve skutecnosti menim jenom jeji souradnice.
                        //je to nahoda jak blazen, ze to facha!!!

                        //u aktualniho hrace budu pracovat s aktivni figurkou, u ostatnich s neaktivni
                        for (int i = 0; i < hra.Hraci.Length; i++)//projdu vsechny hrace
                        {
                           for (int j = 0; j < poleKaret.Length; j++)//u kazdeho z nich prohledam vsechny kameny, ktere se prave budou
posouvat
                           {
                              if (hra.Hraci[i] == hra.AktualniHrac)//aktualni hrac
                              {
                                 if (hra.Hraci[i].Figurka.AktualniKamen == poleKaret[j])//figurka stoji na kamenu, ktery se bude posouvat
                                 {
                                    //ulozim ji do pole pro posunuti figurky
                                    if (proPosunuti == null)
                                    {
                                       proPosunuti = new Figurka[1];
                                       proPosunuti[0] = hra.Hraci[i].Figurka;
                                    }
                                    else
                                    {
                                       Figurka[] tmp = proPosunuti;
                                       proPosunuti = new Figurka[proPosunuti.Length + 1];
                                       tmp.CopyTo(proPosunuti, 0);
                                       proPosunuti[proPosunuti.Length - 1] = hra.Hraci[i].Figurka;
                                    }
                                    break;
                                 }
                              }
                              else//neaktivni hrac, takze priradim tu neaktivni figurku
                              {
                                 if (hra.Hraci[i].NeaktivniFigurka.AktualniKamen == poleKaret[j])//figurka stoji na kamenu, ktery se bude
posouvat
                                 {
                                    //ulozim ji do pole pro posunuti figurky
                                    if (proPosunuti == null)
                                    {
                                       proPosunuti = new Figurka[1];
                                       proPosunuti[0] = hra.Hraci[i].NeaktivniFigurka;
                                    }
                                    else
                                    {
                                       Figurka[] tmp = proPosunuti;
                                       proPosunuti = new Figurka[proPosunuti.Length + 1];
                                       tmp.CopyTo(proPosunuti, 0);
                                       proPosunuti[proPosunuti.Length - 1] = hra.Hraci[i].NeaktivniFigurka;
                                    }
                                    break;
                                 }
                              }
                           }
                        }
                        for (int i = 0; i < 10; i++)
                        {
                           pocetPosunuti += 1;
                           pohni(proPosunuti);
                           this.Refresh();
                        }
                        //vse jsem prave posunul tak, jak ma byt. Ovsem figurka, ktera se vysunula ven z hraci plochy, zustala na
vysunutem kameni
                        //takze ji musim presunout na jeji aktualni kamen.
                        if (proPosunuti != null)
                        {
                           for (int b = 0; b < proPosunuti.Length; b++)
                           {
```

```
                                proPosunuti[b].Pozice = new Point((proPosunuti[b].AktualniKamen.Pozice.X +
proPosunuti[b].AktualniKamen.Rozmery.Width / 2) - proPosunuti[b].Rozmery.Width / 2, (proPosunuti[b].AktualniKamen.Pozice.Y +
proPosunuti[b].AktualniKamen.Rozmery.Height / 2) - proPosunuti[b].Rozmery.Height / 2);
                                this.Refresh();
                                Thread.Sleep(10);
                            }
                        }
                        proPosunuti = null;
                    }
                }
            }
            else//tlacitko je stisknuto poprve
            {
                stisknutoLeve = true;//karta se aktivuje pro pohyb
            }

        }
        else if (tlacitkoMysi == MouseButtons.Right)//na posuvnou kartu jsem kliknul pravym, takze ho chci natocit
        {
            if (stisknutoLeve == true)//karta je aktivovana, tudiz az ted muzu menit uhel
            {
                volajici.Uhel += 90;
                this.Refresh();
            }
        }

    }
    else//klikl jsem dovnitr hraci plochy
    {

        if (hra.VytvorCestu(volajici.VPracovnimPoli))
        {
            hra.PosunFigurku(this);
        }
        else
        {
            MessageBox.Show("Cestu se nepodarilo vyhledat");
        }
    }
  }
}

void casovac_Elapsed(object sender, ElapsedEventArgs e)
{
  if (pocetPosunuti < 10)
  {
    ThreadPool.QueueUserWorkItem(new WaitCallback(ProcesVeVlakne3));
  }
  else
  {
    casovac.Stop();
    pocetPosunuti = 0;
    //vse jsem prave posunul tak, jak ma byt. Ovsem figurka, ktera se vysunula ven z hraci plochy, zustala na vysunutem kameni
    //takze ji musim presunout na jeji aktualni kamen.
    if (proPosunuti != null)
    {
      for (int b = 0; b < proPosunuti.Length; b++)
      {
        proPosunuti[b].Pozice = new Point((proPosunuti[b].AktualniKamen.Pozice.X +
proPosunuti[b].AktualniKamen.Rozmery.Width / 2) - proPosunuti[b].Rozmery.Width / 2, (proPosunuti[b].AktualniKamen.Pozice.Y +
proPosunuti[b].AktualniKamen.Rozmery.Height / 2) - proPosunuti[b].Rozmery.Height / 2);
        ThreadPool.QueueUserWorkItem(new WaitCallback(RefreshFormu));
      }
    }
    proPosunuti = null;
  }
}


//pohyb po formu
GrafickyObjekt predeslyNajetyObjekt = null;
void HraciForm_MouseMove(object sender, MouseEventArgs e)
{
  //kdyz je karta aktivovana, muze hrac posouvat s kamenem po plose, kdyz aktivovana neni, muze prejizdet pres ruzne prvky na
formu
  Point poziceMysi = e.Location;
```

```
            if (stisknutoLeve == true)
            {
               //karta je aktivovana a tudiz muzu s ni posouvat
               for (int i = 0; i < 12; i++)
               {
                  if (poziceMysi.X >= pocatecniPovolene[i].X && poziceMysi.Y >= pocatecniPovolene[i].Y && poziceMysi.X <=
koncovePovolene[i].X && poziceMysi.Y <= koncovePovolene[i].Y)
                  {
                     Posunovaci.Pozice = pocatecniPovolene[i];
                     this.Refresh();
                     break;
                  }
               }

            }
            else
            {
               MouseEventArgs udalostMysi = e;
               for (int i = 0; i < this.prvkyNaFormu.Length; i++)
               {

                  if (this.prvkyNaFormu[i] != null)
                  {
                     if (this.prvkyNaFormu[i].Jmeno == "Plocha")//kontroluje se plocha, takze ji preskocim
                     {
                     }
                     //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
                     else if (udalostMysi.Location.X >= this.prvkyNaFormu[i].Pozice.X && udalostMysi.Location.X <=
this.prvkyNaFormu[i].Pozice.X + this.prvkyNaFormu[i].Rozmery.Width &&
                        udalostMysi.Location.Y >= this.prvkyNaFormu[i].Pozice.Y && udalostMysi.Location.Y <=
this.prvkyNaFormu[i].Pozice.Y + this.prvkyNaFormu[i].Rozmery.Height)
                     {
                        //spusti udalost danneho objektu
                        this.prvkyNaFormu[i].SpoustecUdalostiPrejetoMysi(this.prvkyNaFormu[i]) ;
                        return;
                     }
                  }

               }
               //kdyz kod dosel az sem, znamena to,ze se nachazim na plose
               if (predeslyNajetyObjekt != null)
               {
                  predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
                  predeslyNajetyObjekt = null;
               }
            }
         }

         private bool mysDole = false;
         void HraciForm_MouseDown(object sender, MouseEventArgs e)
         {
            mysDole = true;
            //klikani je povoleno pouze hraci v dobe, kdy hraje, nebo kdyz hra skoncila.
            if ((hra.hrajeSe && hra.AktualniHrac.Ai == false) || hra.hrajeSe == false)
            {
               MouseEventArgs udalostMysi = (MouseEventArgs)e;
               //projde vsechny prvky, ktere jsou vykresleny na formu, vcetne plochy

               for (int i = 0; i < this.prvkyNaFormu.Length; i++)
               {

                  if (this.prvkyNaFormu[i] != null)
                  {
                     if (this.prvkyNaFormu[i].Jmeno == "Plocha")//kontroluje se plocha, takze ji preskocim
                     {
                     }
                     //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
                     else if (udalostMysi.Location.X >= this.prvkyNaFormu[i].Pozice.X && udalostMysi.Location.X <=
this.prvkyNaFormu[i].Pozice.X + this.prvkyNaFormu[i].Rozmery.Width &&
                        udalostMysi.Location.Y >= this.prvkyNaFormu[i].Pozice.Y && udalostMysi.Location.Y <=
this.prvkyNaFormu[i].Pozice.Y + this.prvkyNaFormu[i].Rozmery.Height)
                     {
                        //spusti udalost danneho objektu
```

```
                    this.tlacitkoMysi = udalostMysi.Button;
                    this.prvkyNaFormu[i].SpoustecUdalostiStisknutoMysi(prvkyNaFormu[i]);
                    break;
                }
            }

        }
    }
}

void HraciForm_MouseUp(object sender, MouseEventArgs e)
{
    mysDole = false;
    //klikani je povoleno pouze hraci v dobe, kdy hraje, nebo kdyz hra skoncila.
    if ((hra.hrajeSe && hra.AktualniHrac.Ai == false) || hra.hrajeSe == false)
    {
        MouseEventArgs udalostMysi = (MouseEventArgs)e;
        //projde vsechny prvky, ktere jsou vykresleny na formu, vcetne plochy

        for (int i = 0; i < this.prvkyNaFormu.Length; i++)
        {

            if (this.prvkyNaFormu[i] != null)
            {
                if (this.prvkyNaFormu[i].Jmeno == "Plocha")//kontroluje se plocha, takze ji preskocim
                {
                }
                //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
                else if (udalostMysi.Location.X >= this.prvkyNaFormu[i].Pozice.X && udalostMysi.Location.X <=
this.prvkyNaFormu[i].Pozice.X + this.prvkyNaFormu[i].Rozmery.Width &&
                    udalostMysi.Location.Y >= this.prvkyNaFormu[i].Pozice.Y && udalostMysi.Location.Y <=
this.prvkyNaFormu[i].Pozice.Y + this.prvkyNaFormu[i].Rozmery.Height)
                {
                    //spusti udalost danneho objektu

                    this.tlacitkoMysi = udalostMysi.Button;
                    this.prvkyNaFormu[i].SpoustecUdalostiUvolnenoMysi(prvkyNaFormu[i]);
                    break;
                }
            }

        }
    }
}

public void ProcesVeVlakne(object stateInfo)
{
    VolajiciDelegat delegat = new VolajiciDelegat(hra.NaTahuJePocitac);
    BeginInvoke(delegat);
}
public void ProcesVeVlakne2(object stateInfo)
{
    VolajiciDelegat delegat = new VolajiciDelegat(hra.OdpocetCasu);
    BeginInvoke(delegat);
}
public void ProcesVeVlakne3(object stateInfo)
{
    VolajiciDelegat delegat = new VolajiciDelegat(PosunKameny);
    BeginInvoke(delegat);
}
public void ProcesVeVlakne4(object statInfo)
{
    VolajiciDelegat delegat = new VolajiciDelegat(AI.PosunoutKameny);
    BeginInvoke(delegat);
}
public void RefreshFormu(object stateInfo)
{
    VolajiciDelegat delegat = new VolajiciDelegat(refresh);
    BeginInvoke(delegat);
}
public void PosunoutFigurkuVAI(object stateInfo)
{
    VolajiciDelegat delegat = new VolajiciDelegat(AI.PosunoutFigurku);
    BeginInvoke(delegat);
}
```

```
private void refresh()
{
   this.Refresh();
}


public void dalsiHrac_Kliknuto(object volajici)
{
   //zkontrolovat, zda hrac zasunul kamen, jinak nepovolovat
   if (hracPosunul)//uz zasunul kamen
   {
      //prepnout na dalsiho hrace
      hra.DalsiHrac();
      hracPosunul = false;
      this.prvkyNaFormu = hra.prvkyNaFormu;
      this.Refresh();
   }
}


void HraciForm_Load(object sender, EventArgs e)
{
   if (ukoncitProgram)
   {
      Application.Exit();
   }
   else
   {
      Image[] barvy = null;
      for (int i = 0; i < inter.Barvy.Length; i++)
      {
         Image obrazek = null;
         if (inter.Barvy[i] == Color.Red)
         {
            obrazek = Zdroj.Postavicka_cervena;
         }
         if (inter.Barvy[i] == Color.Green)
         {
            obrazek = Zdroj.Postavicka_zelena;
         }
         if (inter.Barvy[i] == Color.Blue)
         {
            obrazek = Zdroj.Postavicka_modra;
         }
         if (inter.Barvy[i] == Color.Yellow)
         {
            obrazek = Zdroj.Postavicka_zluta;
         }

         if (barvy == null)
         {
            barvy = new Image[1];
            barvy[0] = obrazek;
         }
         else
         {
            Image[] tmp = barvy;
            barvy = new Image[tmp.Length + 1];
            tmp.CopyTo(barvy, 0);
            barvy[barvy.Length - 1] = obrazek;
         }
      }

      hra = new Hra(this.hraciKarty, Hra.TypHry.multi, PracovniPole, prvkyNaFormu, this);//vytvori novou hru
      hra.PridejHrace(inter.Nicks, barvy, inter.Ai, velikostFigurky);
      //hre predam pozici, na kterou se budoy vykreslovat nalezene artefakty
      hra.PoziceNalezeneKarty = poziceKartyNalezene;

      //do pole pro vykreslovani pridam veci, ktere byly pri vytvareni hry zmeneny nebo pridany
      this.prvkyNaFormu = hra.prvkyNaFormu;//musim to prekopirovat zpatky
      //aktualizuju propojeni jednotlivych kamenu mezi sebou
      hra.AktualizujPropojeniCest();
      //nastavi vse potrebne pro zacatek hry
      hra.ZacniHru();
      //-----udelat to tak, aby se to pole volalo primo. Asi lepe navrhnout!!!!!
      this.hracTitulek.TextovyObsah = this.hra.AktualniHrac.Nick;
```

```
            this.prohledaniNalezenych.TextovyObsah = "0 z 0";
            this.nalezenoKolik.TextovyObsah = this.hra.AktualniHrac.Nalezene.Length.ToString() + " z " +
this.hra.AktualniHrac.HraciKarty.Length.ToString();
            this.pocetTahu.TextovyObsah = this.hra.AktualniHrac.PocetTahu.ToString();
            this.umisteni.Vycentruj();
            this.Refresh();

            //kdyz zacina pocitac, pocka nekolik sekund a pote zacne hrat
            if (hra.AktualniHrac.Ai)//zacina pocitac
            {
               hra.casovac.Elapsed += new ElapsedEventHandler(hra.odpocitavaniStartu);
               hra.casovac.Interval = 3000;
               hra.casovac.Start();
            }
            else
            {
               //zacnu odpocitavat cas hraci
               hra.casovac.Elapsed += new System.Timers.ElapsedEventHandler(hra.casovac_Elapsed);
               hra.casovac.Interval = 1000;
               hra.casovac.Start();
            }
         }
      }

      void HraciForm_KeyDown(object sender, KeyEventArgs e)
      {
         if (e.KeyCode == Keys.N)//dalsi hrac
         {
            hracPosunul = false;
            hra.DalsiHrac();
            this.prvkyNaFormu = hra.prvkyNaFormu;
            this.Refresh();
         }
      }

      bool krizekZeleny = false;
      void zavreni_PrejetoMysi(object volajici)
      {
         if (!krizekZeleny)
         {
            //pri prejeti mysi nactu zeleny krizek, ktery bude signalizovat aktivitu
            GrafickyObjekt zavreni = (GrafickyObjekt)volajici;
            zavreni.Obrazek = (Image)new Bitmap((Image)Zdroj.KrizekZeleny, umisteni.ZavritRozmery);
            krizekZeleny = true;
            if (predeslyNajetyObjekt == null)//prechazim s plochy
            {
               predeslyNajetyObjekt = zavreni;
            }
            else//predtim jsem byl na jinem objektu
            {
               predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
               predeslyNajetyObjekt = zavreni;
            }
            this.Refresh();
         }
      }
      void zavreni_StisknutoMysi(object volajici)
      {
         //pri kliknuti nactu cerveny krizek, ktery bude signalizovat zavreni
         GrafickyObjekt zavreni = (GrafickyObjekt)volajici;
         zavreni.Obrazek = (Image)new Bitmap((Image)Zdroj.KrizekCerveny, umisteni.ZavritRozmery);
         this.Refresh();
      }
      void zavreni_OpusteniMysi(object volajici)
      {
         zavreni.Obrazek = (Image)new Bitmap((Image)Zdroj.KrizekBily, umisteni.ZavritRozmery);
         krizekZeleny = false;
         this.Refresh();
      }

      void minimalizace_OpusteniMysi(object volajici)
      {
         minimalizace.Obrazek = (Image)new Bitmap((Image)Zdroj.MinimalizaceBila, umisteni.MinimalizaceRozmery);
         minimalizaceZelena = false;
         this.Refresh();
      }
```

```
void minimalizace_StisknutoMysi(object volajici)
{
   GrafickyObjekt minimalizace = (GrafickyObjekt)volajici;
   minimalizace.Obrazek = (Image)new Bitmap((Image)Zdroj.MinimalizaceCervena, umisteni.MinimalizaceRozmery);
   this.Refresh();
}

bool minimalizaceZelena = false;
void minimalizace_PrejetoMysi(object volajici)
{
   if (!minimalizaceZelena)
   {
      //pri prejeti mysi nactu zeleny krizek, ktery bude signalizovat aktivitu
      GrafickyObjekt minimalizace = (GrafickyObjekt)volajici;
      minimalizace.Obrazek = (Image)new Bitmap((Image)Zdroj.MinimalizaceZelena, umisteni.MinimalizaceRozmery);
      minimalizaceZelena = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = minimalizace;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = minimalizace;
      }
      this.Refresh();
   }
}


void sipkaLevaNick_StisknutoMysi(object volajici)
{
   GrafickyObjekt objekt = (GrafickyObjekt)volajici;
   objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaCervena, umisteni.SipkaLevaNickRozmery);
   this.Refresh();
}

bool sipkaLevaNickZelena = false;
void sipkaLevaNick_PrejetoMysi(object volajici)
{
   if (!sipkaLevaNickZelena)
   {
      //pri prejeti mysi nactu zeleny krizek, ktery bude signalizovat aktivitu
      GrafickyObjekt objekt = (GrafickyObjekt)volajici;
      objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaZelena, umisteni.SipkaLevaNickRozmery);
      sipkaLevaNickZelena = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = objekt;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = objekt;
      }
      this.Refresh();
   }
}

void sipkaLevaNick_OpusteniMysi(object volajici)
{
   GrafickyObjekt objekt = (GrafickyObjekt)volajici;
   objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaBila, umisteni.SipkaLevaNickRozmery);
   sipkaLevaNickZelena = false;
   this.Refresh();
}

void sipkaPravaNick_StisknutoMysi(object volajici)
{
   GrafickyObjekt objekt = (GrafickyObjekt)volajici;
   objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaCervena, umisteni.SipkaPravaNickRozmery);
   this.Refresh();
}

bool sipkaPravaNickZelena = false;
```

```
void sipkaPravaNick_PrejetoMysi(object volajici)
{
   if (!sipkaPravaNickZelena)
   {
      //pri prejeti mysi nactu zeleny krizek, ktery bude signalizovat aktivitu
      GrafickyObjekt objekt = (GrafickyObjekt)volajici;
      objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaZelena, umisteni.SipkaPravaNickRozmery);
      sipkaPravaNickZelena = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = objekt;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = objekt;
      }
      this.Refresh();
   }
}

void sipkaPravaNick_OpusteniMysi(object volajici)
{
   GrafickyObjekt objekt = (GrafickyObjekt)volajici;
   objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaBila, umisteni.SipkaPravaNickRozmery);
   sipkaPravaNickZelena = false;
   this.Refresh();
}

void sipkaPravaNalezeno_StisknutoMysi(object volajici)
{
   GrafickyObjekt objekt = (GrafickyObjekt)volajici;
   objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaCervena, umisteni.SipkaPravaNalezenoRozmery);
   this.Refresh();
}

bool sipkaPravaNalezenoZelena = false;
void sipkaPravaNalezeno_PrejetoMysi(object volajici)
{
   if (!sipkaPravaNalezenoZelena)
   {
      //pri prejeti mysi nactu zeleny krizek, ktery bude signalizovat aktivitu
      GrafickyObjekt objekt = (GrafickyObjekt)volajici;
      objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaZelena, umisteni.SipkaPravaNalezenoRozmery);
      sipkaPravaNalezenoZelena = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = objekt;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = objekt;
      }
      this.Refresh();
   }
}

void sipkaPravaNalezeno_OpusteniMysi(object volajici)
{
   GrafickyObjekt objekt = (GrafickyObjekt)volajici;
   objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaBila, umisteni.SipkaPravaNalezenoRozmery);
   sipkaPravaNalezenoZelena= false;
   this.Refresh();
}

void sipkaLevaNalezeno_StisknutoMysi(object volajici)
{
   GrafickyObjekt objekt = (GrafickyObjekt)volajici;
   objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaCervena, umisteni.SipkaLevaNalezenoRozmery);
   this.Refresh();
}

bool sipkaLevaNalezenoZelena = false;
void sipkaLevaNalezeno_PrejetoMysi(object volajici)
{
```

```
            if (!sipkaLevaNalezenoZelena)
            {
               //pri prejeti mysi nactu zeleny krizek, ktery bude signalizovat aktivitu
               GrafickyObjekt objekt = (GrafickyObjekt)volajici;
               objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaZelena, umisteni.SipkaLevaNalezenoRozmery);
               sipkaLevaNalezenoZelena = true;
               if (predeslyNajetyObjekt == null)//prechazim s plochy
               {
                  predeslyNajetyObjekt = objekt;
               }
               else//predtim jsem byl na jinem objektu
               {
                  predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
                  predeslyNajetyObjekt = objekt;
               }
               this.Refresh();
            }
         }

         void sipkaLevaNalezeno_OpusteniMysi(object volajici)
         {
            GrafickyObjekt objekt = (GrafickyObjekt)volajici;
            objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaBila, umisteni.SipkaLevaNalezenoRozmery);
            sipkaLevaNalezenoZelena = false;
            this.Refresh();
         }

         void sipkaLevaNick_UvolnenoMysi(object volajici)
         {
            GrafickyObjekt objekt = (GrafickyObjekt)volajici;
            objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaZelena, umisteni.SipkaLevaNickRozmery);

            //zjistim zda se nenachazim na zacatku v poli hracu, a kdyz ne, tak posunu na predchoziho hrace
            if (this.hracTitulek.TextovyObsah == this.hra.Hraci[0].Nick)
            {
            }
            else
            {
               //zjistim, kde v poli hracu se nachazi ten aktualni zobrazeny
               for (int i = 0; i < this.hra.Hraci.Length; i++)
               {
                  if (this.hra.Hraci[i].Nick == this.hracTitulek.TextovyObsah)//nasel jsem ho
                  {
                     this.hracTitulek.TextovyObsah = this.hra.Hraci[i - 1].Nick;
                     this.nalezenoKolik.TextovyObsah = this.hra.Hraci[i - 1].Nalezene.Length.ToString() + " z " + this.hra.Hraci[i -
1].HraciKarty.Length.ToString();
                     this.pocetTahu.TextovyObsah = this.hra.Hraci[i - 1].PocetTahu.ToString();
                     this.prohledaniNalezenych.TextovyObsah = this.hra.Hraci[i - 1].Nalezene.Length.ToString() + " z " + this.hra.Hraci[i -
1].Nalezene.Length.ToString();
                     if (this.hra.Hraci[i - 1].Nalezene.Length != 0)
                     {
                        this.prvkyNaFormu[prvkyNaFormu.Length - 2] = this.hra.Hraci[i - 1].Nalezene[hra.Hraci[i - 1].Nalezene.Length -
1];//zobrazeni posledni karty
                     }
                     else
                     {
                        this.prvkyNaFormu[prvkyNaFormu.Length - 2] = null;
                     }

                     this.umisteni.Vycentruj();
                     break;
                  }
               }
            }
            this.Refresh();
         }

         void sipkaPravaNalezeno_UvolnenoMysi(object volajici)
         {

            GrafickyObjekt objekt = (GrafickyObjekt)volajici;
            objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaZelena, umisteni.SipkaPravaNalezenoRozmery);

            //posunuju se dale v poli nalezenych artefaktu
            //defaultne jsem na konci
            //zjistim pozici, na ktere se v poli aktualne nachazim a posunu se dale, maximalne do konce
```

```
            //tato karta je ulozena v poli prvkyNaFormu na predposledni pozici

            //kdyz uz byla nejaka karta nalezena, zkontroluju, zda neni zobrazena posledni karta. Kdyz ano, nic neprovedu, v opacnem
pripade posunu dale
            //zjistim, ktereho hrace informace jsou prave zobrazovany
            Hrac hrac = null;
            for (int j = 0; j < this.hra.Hraci.Length; j++)
            {
               if (this.hra.Hraci[j].Nick == this.hracTitulek.TextovyObsah)//nasel jsem ho
               {
                  hrac = this.hra.Hraci[j];
               }
            }
            if (hrac.Nalezene.Length != 0)
            {
               if ((HraciKarta)this.prvkyNaFormu[prvkyNaFormu.Length - 2] == hrac.Nalezene[hrac.Nalezene.Length - 1])
               {
               }
               else
               {
                  //zjistim, ktera karta je prave zobrazena
                  for (int i = 0; i < hrac.Nalezene.Length; i++)
                  {
                     if (hrac.Nalezene[i] == (HraciKarta)this.prvkyNaFormu[prvkyNaFormu.Length - 2])//nasel ji
                     {
                        this.prvkyNaFormu[prvkyNaFormu.Length - 2] = hrac.Nalezene[i + 1];
                        this.prohledaniNalezenych.TextovyObsah = (i + 2).ToString() + " z " + hrac.Nalezene.Length.ToString();
                        this.umisteni.Vycentruj();
                        break;
                     }
                  }
               }
            }
            this.Refresh();
         }

         void sipkaLevaNalezeno_UvolnenoMysi(object volajici)
         {

            GrafickyObjekt objekt = (GrafickyObjekt)volajici;
            objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.LevaSipkaZelena, umisteni.SipkaLevaNalezenoRozmery);

            //zjistim, zda neni zobrazena prvni karta v poli, a kdyz ne, tak zobrazim predeslou kartu v poli
            Hrac hrac = null;
            for (int j = 0; j < this.hra.Hraci.Length; j++)
            {
               if (this.hra.Hraci[j].Nick == this.hracTitulek.TextovyObsah)//nasel jsem ho
               {
                  hrac = this.hra.Hraci[j];
               }
            }
            if (hrac.Nalezene.Length != 0)
            {
               if ((HraciKarta)this.prvkyNaFormu[prvkyNaFormu.Length - 2] == hrac.Nalezene[0])
               {
               }
               else
               {
                  //zjistim, ktera karta je prave zobrazena
                  for (int i = 0; i < hrac.Nalezene.Length; i++)
                  {
                     if (hrac.Nalezene[i] == (HraciKarta)this.prvkyNaFormu[prvkyNaFormu.Length - 2])
                     {
                        this.prvkyNaFormu[prvkyNaFormu.Length - 2] = hrac.Nalezene[i - 1];
                        this.prohledaniNalezenych.TextovyObsah = i.ToString() + " z " + hrac.Nalezene.Length.ToString();
                        this.umisteni.Vycentruj();
                        break;
                     }
                  }
               }
            }
            this.Refresh();
         }

         void sipkaPravaNick_UvolnenoMysi(object volajici)
         {
```

```
            GrafickyObjekt objekt = (GrafickyObjekt)volajici;
            objekt.Obrazek = (Image)new Bitmap((Image)Zdroj.PravaSipkaZelena, umisteni.SipkaPravaNickRozmery);

            //zjistim zda se nenachazim na konci v poli hracu, a kdyz ne, tak posunu na dalsiho hrace
            if (this.hracTitulek.TextovyObsah == this.hra.Hraci[hra.Hraci.Length - 1].Nick)
            {
            }
            else
            {
               //zjistim, kde v poli hracu se nachazi ten aktualni zobrazeny
               for (int i = 0; i < this.hra.Hraci.Length; i++)
               {
                  if (this.hra.Hraci[i].Nick == this.hracTitulek.TextovyObsah)//nasel jsem ho
                  {
                     this.hracTitulek.TextovyObsah = this.hra.Hraci[i + 1].Nick;
                     this.nalezenoKolik.TextovyObsah = this.hra.Hraci[i + 1].Nalezene.Length.ToString() + " z " + this.hra.Hraci[i +
1].HraciKarty.Length.ToString();
                     this.pocetTahu.TextovyObsah = this.hra.Hraci[i + 1].PocetTahu.ToString();
                     this.prohledaniNalezenych.TextovyObsah = this.hra.Hraci[i + 1].Nalezene.Length.ToString() + " z " + this.hra.Hraci[i +
1].Nalezene.Length.ToString();
                     if (this.hra.Hraci[i + 1].Nalezene.Length != 0)
                     {
                        this.prvkyNaFormu[prvkyNaFormu.Length - 2] = this.hra.Hraci[i + 1].Nalezene[hra.Hraci[i + 1].Nalezene.Length -
1];//zobrazeni posledni karty
                     }
                     else
                     {
                        this.prvkyNaFormu[prvkyNaFormu.Length - 2] = null;
                     }

                     this.umisteni.Vycentruj();

                     break;
                  }
               }
            }
            this.Refresh();
         }

         void zavreni_UvolnenoMysi(object volajici)
         {


            //pri kliknuti nactu cerveny krizek, ktery bude signalizovat zavreni
            GrafickyObjekt zavreni = (GrafickyObjekt)volajici;
            zavreni.Obrazek = (Image)new Bitmap((Image)Zdroj.KrizekZeleny, umisteni.ZavritRozmery);
            this.Refresh();

            //pred zavrenim se ujistim, zda uzivatel chce ukoncit hru, kdyz ne, nastavim krizek zpet na puvodni barvu
            DialogResult vysledek = MessageBox.Show("Opravdu chcete zavřít aplikaci?", "Ukončení", MessageBoxButtons.YesNo,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);
            if (vysledek == DialogResult.Yes)
            {
               Application.Exit();
            }
            else
            {
               zavreni.Obrazek = (Image)new Bitmap((Image)Zdroj.KrizekBily, umisteni.ZavritRozmery);
               this.Refresh();
            }
         }

         void minimalizace_UvolnenoMysi(object volajici)
         {
            GrafickyObjekt minimalizace = (GrafickyObjekt)volajici;
            minimalizace.Obrazek = (Image)new Bitmap((Image)Zdroj.MinimalizaceZelena, umisteni.MinimalizaceRozmery);
            this.Refresh();
            this.WindowState = FormWindowState.Minimized;
         }

         private void PosunKameny()
         {
               pocetPosunuti += 1;
               pohni(proPosunuti);
               this.Refresh();
         }
```

```
    }
}
```

C9 HRACIKARTA

```
using System;
using System.Drawing;

namespace Labyrint
{
    public class HraciKarta:GrafickyObjekt
    {
        private string artefakt = "";
        public HraciKarta(Image Obrazek)
            :base(Obrazek)
        {
        }
        public string Artefakt
        {
            get
            {
                return artefakt;
            }
            set
            {
                artefakt = value;
            }
        }
    }
}
```

C10KAMEN

```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace Labyrint
{
    public enum typKarty
    {
        rovina = 1,
        zatacka = 2,
        krizovatka = 3
    }
    public class Kamen : GrafickyObjekt
    {
        public Kamen(Image Obrazek)
            : base(Obrazek)//NOvemu kamenu predam obrazek, ktery se presune do zakladadni tridy, a tam provede potrebne inicializacni
operace
        {
        }
        public Kamen() :
            base()
        {
        }
        private Point vPracovnimPoli;
        private typKarty typkarty;
        private bool[] cesty;
        private Point[] propojeneKameny;//pole souradnic v pracovnim poli, ktere urcuji karticky, se kterymi ma propojenou cestu
        private bool[] pouzitelneSmery;
        private Point[] jesteProhledat;
        private string artefakt = "";
        public string Artefakt
        {
            get
            {
                return artefakt;
            }
            set
            {
                artefakt = value;
            }
```

```
}
//vrati nebo nastavi pozici v pracovnim poli, neboli otisku hraci plochy
public Point VPracovnimPoli
{
   get
   {
            return vPracovnimPoli;
   }
   set
   {
            vPracovnimPoli = value;
   }
}
public typKarty TypKarty
{
   get
   {
            return typkarty;
   }
   set
   {
            typkarty = value;
   }
}
public override float Uhel
{
   get
   {
            return uhel;
   }
   set
   {
      //pri zmene uhlu se automaticky upravi pruchodnost cest
      //chybny blok
      /*int otoceni = (int)value - (int)uhel;
      if (otoceni < 0)
      {
         otoceni = otoceni * (-1);
      }
      uhel = value % 360;//predavam i uhel 360 stupnu, coz je nesmysl, takze ho prevedu na nulu*/
      //konec chybneho bloku

      //oprava
      /*if (value == 0)
      {
         value = 360;
      }*/
         int otoceni = (int)value - (int)uhel;
      if (otoceni < 0)
      {
         //otoceni = otoceni * (-1);//kdyz je -90, tak je mysleno 270; kdyz -270, tak 90
         //oprava
         switch (otoceni)
         {
            case -90:
               otoceni = 270;
               break;
            case -270:
               otoceni = 90;
               break;
            default:
               otoceni = otoceni * (-1);
               break;
         }
         //
      }
      uhel = value % 360;

      //konec opravy

      if (this.obrazek != null)
      {
         switch (otoceni)//natoci se o tolik stupnu. At ma uhel jakykoli, toto natoceni je o x stupnu vuci soucasnemu stavu
         {
            case 0:
               obrazek.RotateFlip(RotateFlipType.RotateNoneFlipNone);
```

```
                    //MessageBox.Show("value=" + value.ToString() + "; uhel=" + uhel.ToString() + "; Natoceno o 0°");
                    break;
                case 90:
                    this.obrazek.RotateFlip(RotateFlipType.Rotate90FlipNone);
                    //MessageBox.Show("value=" + value.ToString() + "; uhel=" + uhel.ToString() + "; Natoceno o 90°");
                    break;
                case 180:
                    obrazek.RotateFlip(RotateFlipType.Rotate180FlipNone);
                    //MessageBox.Show("value=" + value.ToString() + "; uhel=" + uhel.ToString() + "; Natoceno o 180°");
                    break;
                case 270:
                    obrazek.RotateFlip(RotateFlipType.Rotate270FlipNone);
                    //MessageBox.Show("value=" + value.ToString() + "; uhel=" + uhel.ToString() + "; Natoceno o 270°");
                    break;
            }
        }
        zmenaUhlu += 1;
        this.AktualizujSmeryCest();
    }
}
//vrati smery, ktere kamen implementuje
public bool[] Cesty
{
    get
    {
        return cesty;
    }
    set
    {
        cesty = value;
    }
}
//vrati nebo nastavi souradnice propojenych kamenu
public Point[] PropojeneKameny
{
    get
    {
        return propojeneKameny;
    }
    set
    {
        propojeneKameny = value;
    }
}
//smery, ktere jsou pouzitelne pro samotnou hru, tedy uz problehla kontrola na okraje a podobne
public bool[] PouzitelneSmery
{
    get
    {
            return pouzitelneSmery;
    }
    set
    {
        pouzitelneSmery = value;
    }
}
//rika, jestli se pri hledani cesty ma jeste hledat od tohoto kamene dale. Jedna se o krizovatky, ktere maji tri smery:1. ze ktereho
jsem prisel
//2. do ktereho smeruju a 3. ke kteremu se jeste musim vratit, a to mi indikuje tato promenna
public Point[] JesteProhledat
{
    get
    {
            return jesteProhledat;
    }
    set
    {
        jesteProhledat = value;
    }
}
//-------metody----------
//na zaklade uhlu natoceni a typu kamene urci smery, do kterych miri cesty, neurcuje vsak jejich pruchodnost!!!
public void AktualizujSmeryCest()
{
    //cesty>>>0-doprava;1-dolu;2-doleva;3-nahoru
    if(cesty==null)
```

```
{
   cesty =new bool[4];
}//pri prvnim pruchodu nastavi velikost pole
switch (typkarty)
{
   case typKarty.krizovatka:
      switch ((int)uhel)
      {
         case 0:
            cesty[0] = true;
            cesty[1] = false;
            cesty[2] = true;
            cesty[3] = true;
            break;
         case 90:
            cesty[0] = true;
            cesty[1] = true;
            cesty[2] = false;
            cesty[3] = true;
            break;
         case 180:
            cesty[0] = true;
            cesty[1] = true;
            cesty[2] = true;
            cesty[3] = false;
            break;
         case 270:
            cesty[0] = false;
            cesty[1] = true;
            cesty[2] = true;
            cesty[3] = true;
            break;
      }
      break;
   case typKarty.rovina:
      switch ((int)uhel)
      {
         case 0:
            cesty[0] = true;
            cesty[1] = false;
            cesty[2] = true;
            cesty[3] = false;
            break;
         case 90:
            cesty[0] = false;
            cesty[1] = true;
            cesty[2] = false;
            cesty[3] = true;
            break;
         case 180:
            cesty[0] = true;
            cesty[1] = false;
            cesty[2] = true;
            cesty[3] = false;
            break;
         case 270:
            cesty[0] = false;
            cesty[1] = true;
            cesty[2] = false;
            cesty[3] = true;
            break;
      }
      break;
   case typKarty.zatacka:
      switch ((int)uhel)
      {
         case 0:
            cesty[0] = false;
            cesty[1] = true;
            cesty[2] = true;
            cesty[3] = false;
            break;
         case 90:
            cesty[0] = false;
            cesty[1] = false;
            cesty[2] = true;
```

```
                    cesty[3] = true;
                    break;
                case 180:
                    cesty[0] = true;
                    cesty[1] = false;
                    cesty[2] = false;
                    cesty[3] = true;
                    break;
                case 270:
                    cesty[0] = true;
                    cesty[1] = true;
                    cesty[2] = false;
                    cesty[3] = false;
                    break;
            }
            break;
        }
    }
  }
}
```

C11MENU

```
using System;
using System.Windows.Forms;
using System.Drawing;
using System.Threading;
using System.Drawing.Text;

namespace Labyrint
{
  enum TYPmenu
  {
    uvodni = 0,
    hraci = 1,
    pravidla=2
  }
  class UvodniMenu : Form
  {

    GrafickyObjekt menuHraci, menuUvod;
    GrafickyObjekt[] naFormu, naFormuUvod, naFormuPravidla;
    private TYPmenu typMenu;
    Text zpet, zacitHru;
    TextBox hrac1, hrac2, hrac3, hrac4;
    Text cislo1, cislo2, cislo3, cislo4;
    Text sipka1, sipka2, sipka3, sipka4, sipka5, sipka6, sipka7, sipka8;
    Text nadBarva, nadJmeno, nadRezim, nadNovaHra;
    Text rezim1, rezim2, rezim3, rezim4;
    TextBox[] nicky = new TextBox[4];
    Text[] poleCisel = new Text[4];
    Barva[] poleBarev = new Barva[8];
    Text[] poleRezimu = new Text[4];
    Barva Barva1, Barva2, Barva3, Barva4, Barva5, Barva6, Barva7, Barva8;
    Point[] poleSouradnic;
    Size rozmeryBarvy;
    public PrivateFontCollection sadaFontu = new PrivateFontCollection();
    GrafickyObjekt predeslyNajetyObjekt;

    Barva[] dostupneBarvy;
    bool ukoncit = false;//indikuje, zda mam ukoncit program treba kvuli nejake chybe...
    bool uvodniNacteno = false, hraciNacteno = false;


    //Label pravidla, autori, konec, novaHra;
    Text pravidla, autori, konec, novaHra;
    RozliseniMenu umisteni;

    //promenne pro pravidla
    Text pravidlaTitulek;
    MultiLineText pravidlaText;
    GrafickyObjekt pravidlaSipkaDolu, pravidlaSipkaNahoru;
    string pravidlaObsah;
    //-----vlastnosti----
```

```
        public TYPmenu TypMenu
        {
            get
            {
                return typMenu;
            }
            set
            {
                typMenu = value;
            }
        }

        public UvodniMenu(TYPmenu TypMenu)
        {
            this.DoubleBuffered = true;//pri vice obrazcich k vykresleni, je nejprve vykresluje do pameti a az vysledny obrazek vykresli na
form - 1x
            this.BackColor = Color.FromArgb(24, 255, 0);
            this.TransparencyKey = Color.FromArgb(24, 255, 0);
            this.Click += new EventHandler(UvodniMenu_Click);
            this.MouseMove += new MouseEventHandler(UvodniMenu_MouseMove);
            SetStyle(ControlStyles.UserPaint, true);

            this.typMenu = TypMenu;
            //nactu font
                try
                {
                    sadaFontu.AddFontFile("./Resources/fonty/segoesc.ttf");
                }
                catch (System.IO.FileNotFoundException e)
                {
                    MessageBox.Show("Potřebný font nebyl v tomto počítači nalezen! Nainstalujte si, prosím, font s názvem Segoe Script. Poté
spusťte program znovu");
                    ukoncit = true;
                }

            NactiObjety();
            PrizpusobRozliseni();
        }

        void UvodniMenu_MouseMove(object sender, MouseEventArgs e)
        {
            //na zaklade zobrazeneho menu budu nacitat udalosti objektu

            MouseEventArgs udalostMysi = (MouseEventArgs)e;
            //projde vsechny prvky, ktere jsou vykresleny na formu, vcetne plochy
            if (TypMenu == TYPmenu.uvodni)
            {
                for (int i = 0; i < this.naFormuUvod.Length; i++)
                {

                    if (this.naFormuUvod[i] != null)
                    {
                        if (this.naFormuUvod[i].Jmeno == "MenuUvod")//kontroluje se plocha, takze ji preskocim
                        {
                        }
                        //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
                        else if (udalostMysi.Location.X >= this.naFormuUvod[i].Pozice.X && udalostMysi.Location.X <=
this.naFormuUvod[i].Pozice.X + this.naFormuUvod[i].Rozmery.Width &&
                            udalostMysi.Location.Y >= this.naFormuUvod[i].Pozice.Y && udalostMysi.Location.Y <=
this.naFormuUvod[i].Pozice.Y + this.naFormuUvod[i].Rozmery.Height)
                        {
                            //spusti udalost danneho objektu
                            this.naFormuUvod[i].SpoustecUdalostiPrejetoMysi(naFormuUvod[i]);
                            return;
                        }
                    }

                }
                //kdyz kod dosel az sem, znamena to,ze se nachazim na plose
                if (predeslyNajetyObjekt != null)
                {
                    predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
                    predeslyNajetyObjekt = null;
                }
            }
```

```
            else if (typMenu == TYPmenu.hraci)
            {
               for (int i = 0; i < this.naFormu.Length; i++)
               {

                  if (this.naFormu[i] != null)
                  {
                     if (this.naFormu[i].Jmeno == "MenuHraci")//kontroluje se plocha, takze ji preskocim
                     {
                     }
                     //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
                     else if (udalostMysi.Location.X >= this.naFormu[i].Pozice.X && udalostMysi.Location.X <= this.naFormu[i].Pozice.X +
this.naFormu[i].Rozmery.Width &&
                        udalostMysi.Location.Y >= this.naFormu[i].Pozice.Y && udalostMysi.Location.Y <= this.naFormu[i].Pozice.Y +
this.naFormu[i].Rozmery.Height)
                     {
                        //spusti udalost danneho objektu
                        this.naFormu[i].SpoustecUdalostiPrejetoMysi(naFormu[i]);
                        return;
                     }
                  }

               }
               //kdyz kod dosel az sem, znamena to,ze se nachazim na plose
               if (predeslyNajetyObjekt != null)
               {
                  predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
                  predeslyNajetyObjekt = null;
               }
            }
         }

         void UvodniMenu_Click(object sender, EventArgs e)
         {
            //na zaklade zobrazeneho menu budu nacitat udalosti objektu

            MouseEventArgs udalostMysi = (MouseEventArgs)e;
            //projde vsechny prvky, ktere jsou vykresleny na formu, vcetne plochy
            if (TypMenu == TYPmenu.uvodni)
            {
               for (int i = 0; i < this.naFormuUvod.Length; i++)
               {

                  if (this.naFormuUvod[i] != null)
                  {
                     if (this.naFormuUvod[i].Jmeno == "MenuUvod")//kontroluje se plocha, takze ji preskocim
                     {
                     }
                     //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
                     else if (udalostMysi.Location.X >= this.naFormuUvod[i].Pozice.X && udalostMysi.Location.X <=
this.naFormuUvod[i].Pozice.X + this.naFormuUvod[i].Rozmery.Width &&
                        udalostMysi.Location.Y >= this.naFormuUvod[i].Pozice.Y && udalostMysi.Location.Y <=
this.naFormuUvod[i].Pozice.Y + this.naFormuUvod[i].Rozmery.Height)
                     {
                        //spusti udalost danneho objektu
                        this.naFormuUvod[i].SpoustecUdalosti(naFormuUvod[i]);
                        break;
                     }
                  }

               }
            }
            else if (typMenu == TYPmenu.hraci)
            {
               for (int i = 0; i < this.naFormu.Length; i++)
               {

                  if (this.naFormu[i] != null)
                  {
                     if (this.naFormu[i].Jmeno == "MenuHraci")//kontroluje se plocha, takze ji preskocim
                     {
                     }
                     //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
```

```
                    else if (udalostMysi.Location.X >= this.naFormu[i].Pozice.X && udalostMysi.Location.X <= this.naFormu[i].Pozice.X +
this.naFormu[i].Rozmery.Width &&
                        udalostMysi.Location.Y >= this.naFormu[i].Pozice.Y && udalostMysi.Location.Y <= this.naFormu[i].Pozice.Y +
this.naFormu[i].Rozmery.Height)
                    {
                        //spusti udalost danneho objektu
                        this.naFormu[i].SpoustecUdalosti(naFormu[i]);
                        break;
                    }
                }

            }
        }
        else if(typMenu==TYPmenu.pravidla)
        {
            for (int i = 0; i < this.naFormuPravidla.Length; i++)
            {

                if (this.naFormuPravidla[i] != null)
                {
                    if (this.naFormuPravidla[i].Jmeno == "MenuUvod")//kontroluje se plocha, takze ji preskocim
                    {
                    }
                    //kontroluje pozice a zjistuje, zda je kliknuto na nejaky kamen pomoci intervalu souradnic. V poli je i plocha, takze se
nesmi jmenovat "Plocha"
                    else if (udalostMysi.Location.X >= this.naFormuPravidla[i].Pozice.X && udalostMysi.Location.X <=
this.naFormuPravidla[i].Pozice.X + this.naFormuPravidla[i].Rozmery.Width &&
                        udalostMysi.Location.Y >= this.naFormuPravidla[i].Pozice.Y && udalostMysi.Location.Y <=
this.naFormuPravidla[i].Pozice.Y + this.naFormuPravidla[i].Rozmery.Height)
                    {
                        //spusti udalost danneho objektu
                        this.naFormuPravidla[i].SpoustecUdalostiStisknutoMysi(naFormuPravidla[i]);
                        break;
                    }
                }

            }
        }
    }

    private void NactiObjety()
    {
        try
        {
            this.FormBorderStyle = FormBorderStyle.None;
            switch ((int)typMenu)
            {
                case (int)TYPmenu.uvodni:
                    this.Text = "Labyrint";
                    this.Controls.Clear();
                    if (!uvodniNacteno)
                    {
                        //----menuUvod----
                        menuUvod = new GrafickyObjekt();
                        menuUvod.Jmeno = "MenuUvod";

                        this.Load += new EventHandler(UvodniMenu_Load);

                        novaHra = new Text(this);
                        novaHra.FontFamily = sadaFontu.Families[0];
                        novaHra.FontSize = 24f;
                        novaHra.Jmeno = "novaHra";
                        novaHra.TextovyObsah = "NOVÁ HRA";
                        novaHra.Kliknuto += new KliknutiHandler(novaHra_Kliknuto);
                        novaHra.Rozmery = (this.CreateGraphics().MeasureString(novaHra.TextovyObsah, novaHra.Font)).ToSize();
                        novaHra.PrejetoMysi += new PrejetiNadHandler(novaHra_PrejetoMysi);
                        novaHra.OpusteniMysi += new OpusteniHandler(novaHra_OpusteniMysi);

                        pravidla = new Text(this);
                        pravidla.FontFamily = sadaFontu.Families[0];
                        pravidla.FontSize = 24f;
                        pravidla.Jmeno = " pravidla";
                        pravidla.TextovyObsah = "PRAVIDLA";
                        pravidla.Kliknuto += new KliknutiHandler(pravidla_Kliknuto);
                        pravidla.Rozmery = (this.CreateGraphics().MeasureString(pravidla.TextovyObsah, pravidla.Font)).ToSize();
                        pravidla.PrejetoMysi += new PrejetiNadHandler(pravidla_PrejetoMysi);
```

```
            pravidla.OpusteniMysi += new OpusteniHandler(pravidla_OpusteniMysi);

            autori = new Text(this);
            autori.FontFamily = sadaFontu.Families[0];
            autori.FontSize = 24f;
            autori.Jmeno = "autori";
            autori.TextovyObsah = "AUTOŘI";
            autori.Kliknuto += new KliknutiHandler(autori_Kliknuto);
            autori.Rozmery = (this.CreateGraphics().MeasureString(autori.TextovyObsah, autori.Font)).ToSize();
            autori.PrejetoMysi += new PrejetiNadHandler(autori_PrejetoMysi);
            autori.OpusteniMysi += new OpusteniHandler(autori_OpusteniMysi);

            konec = new Text(this);
            konec.FontFamily = sadaFontu.Families[0];
            konec.FontSize = 24f;
            konec.Jmeno = "konec";
            konec.TextovyObsah = "KONEC";
            konec.Kliknuto += new KliknutiHandler(konec_Kliknuto);
            konec.Rozmery = (this.CreateGraphics().MeasureString(konec.TextovyObsah, konec.Font)).ToSize();
            konec.PrejetoMysi += new PrejetiNadHandler(konec_PrejetoMysi);
            konec.OpusteniMysi += new OpusteniHandler(konec_OpusteniMysi);

            naFormuUvod = new GrafickyObjekt[5];
            naFormuUvod[0] = menuUvod;
            naFormuUvod[1] = novaHra;
            naFormuUvod[2] = pravidla;
            naFormuUvod[3] = autori;
            naFormuUvod[4] = konec;

            uvodniNacteno = true;
        }
        break;
    case (int)TYPmenu.hraci:
        this.Controls.Add(hrac1);
        this.Controls.Add(hrac2);
        this.Controls.Add(hrac3);
        this.Controls.Add(hrac4);
        if (!hraciNacteno)
        {
            Graphics gr = this.CreateGraphics();
            //-------menuHraci------
            //slouzi pro zadavana hracu do hry
            menuHraci = new GrafickyObjekt();//((Image)Zdroj.menu_hraci);
            menuHraci.Obrazek = umisteni.ObrazekMenuHraci;
            menuHraci.Rozmery = umisteni.ObrazekMenuHraciRozmery;
            menuHraci.Jmeno = "MenuHraci";
            //----formular-----
            this.Text = "Nová hra";

            //-------textboxy--------

            //-----hrac1----
            hrac1 = new TextBox();
            hrac1.Location = umisteni.Hrac1;
            hrac1.Width = 250;
            hrac1.Font = new Font(sadaFontu.Families[0], 15);
            hrac1.ForeColor = Color.White;
            hrac1.BackColor = Color.FromArgb(64, 76, 105);
            hrac1.BorderStyle = BorderStyle.None;
            hrac1.TabIndex = 1;
            this.Controls.Add(hrac1);

            //-----hrac2----
            hrac2 = new TextBox();
            hrac2.Location = umisteni.Hrac2;
            hrac2.Width = 250;
            hrac2.Font = new Font(sadaFontu.Families[0], 15);
            hrac2.ForeColor = Color.White;
            hrac2.BackColor = Color.FromArgb(64, 76, 105);
            hrac2.BorderStyle = BorderStyle.None;
            hrac2.TabIndex = 2;
            this.Controls.Add(hrac2);

            //-----hrac3----
            hrac3 = new TextBox();
            hrac3.Location = umisteni.Hrac3;
```

```
hrac3.Width = 250;
hrac3.Font = new Font(sadaFontu.Families[0], 15);
hrac3.ForeColor = Color.White;
hrac3.BackColor = Color.FromArgb(59, 71, 102);
hrac3.TabIndex = 3;
hrac3.BorderStyle = BorderStyle.None;
this.Controls.Add(hrac3);

//-----hrac4----
hrac4 = new TextBox();
hrac4.Location = umisteni.Hrac4;
hrac4.Width = 250;
hrac4.Font = new Font(sadaFontu.Families[0], 15);
hrac4.ForeColor = Color.White;
hrac4.BackColor = Color.FromArgb(58, 70, 98);
hrac4.TabIndex = 4;
hrac4.BorderStyle = BorderStyle.None;
this.Controls.Add(hrac4);

//----pridani do pole----
nicky[0] = hrac1;
nicky[1] = hrac2;
nicky[2] = hrac3;
nicky[3] = hrac4;

//---------labely s cisly---------

cislo1 = new Text(this);
cislo1.FontFamily = sadaFontu.Families[0];
cislo1.Pozice = umisteni.Cislo1;
cislo1.FontSize = 15f;
cislo1.TextovyObsah = "1";
cislo1.Jmeno = "cislo1";
cislo1.Rozmery = (gr.MeasureString(cislo1.TextovyObsah, cislo1.Font)).ToSize();

//----cislo2----
cislo2 = new Text(this);
cislo2.FontFamily = sadaFontu.Families[0];
cislo2.Pozice = umisteni.Cislo2;
cislo2.TextovyObsah = "2";
cislo2.FontSize = 15f;
cislo2.Jmeno = "cislo2";
cislo2.Rozmery = (gr.MeasureString(cislo2.TextovyObsah, cislo2.Font)).ToSize();
cislo2.Kliknuto += new KliknutiHandler(cislo2_Click);
//----cislo3----
cislo3 = new Text(this);
cislo3.FontFamily = sadaFontu.Families[0];
cislo3.Pozice = umisteni.Cislo3;
cislo3.TextovyObsah = "X";
cislo3.FontSize = 15f;
cislo3.Jmeno = "cislo3";
cislo3.Rozmery = (gr.MeasureString(cislo3.TextovyObsah, cislo3.Font)).ToSize();
cislo3.Kliknuto += new KliknutiHandler(cislo3_Click);
//----cislo4----
cislo4 = new Text(this);
cislo4.FontFamily = sadaFontu.Families[0];
cislo4.Pozice = umisteni.Cislo4;
cislo4.TextovyObsah = "X";
cislo4.FontSize = 15f;
cislo4.Jmeno = "cislo4";
cislo4.Rozmery = (gr.MeasureString(cislo4.TextovyObsah, cislo4.Font)).ToSize();
cislo4.Kliknuto += new KliknutiHandler(cislo4_Click);

poleCisel[0] = cislo1;
poleCisel[1] = cislo2;
poleCisel[2] = cislo3;
poleCisel[3] = cislo4;
//---------labely jako tlacitka---------

//----zpet----

zpet = new Text(this);
zpet.FontFamily = sadaFontu.Families[0];
zpet.Pozice = umisteni.Zpet;
zpet.TextovyObsah = "ZPĚT";
zpet.FontSize = 15f;
```

```
zpet.Jmeno = "zpet";
zpet.Rozmery = (gr.MeasureString(zpet.TextovyObsah, zpet.Font)).ToSize();
zpet.Kliknuto += new KliknutiHandler(zpet_Click);
zpet.PrejetoMysi += new PrejetiNadHandler(zpet_PrejetoMysi);
zpet.OpusteniMysi += new OpusteniHandler(zpet_OpusteniMysi);
//----zacitHru----
zacitHru = new Text(this);
zacitHru.FontFamily = sadaFontu.Families[0];
zacitHru.Pozice = umisteni.ZacitHru;
zacitHru.TextovyObsah = "ZAČÍT HRU";
zacitHru.FontSize = 15f;
zacitHru.Jmeno = "zacitHru";
zacitHru.Kliknuto += new KliknutiHandler(zacitHru_Click);
zacitHru.Rozmery = (gr.MeasureString(zacitHru.TextovyObsah, zacitHru.Font)).ToSize();
zacitHru.PrejetoMysi += new PrejetiNadHandler(zacitHru_PrejetoMysi);
zacitHru.OpusteniMysi += new OpusteniHandler(zacitHru_OpusteniMysi);

//---------labely jako sipky----------

//----sipka1----
sipka1 = new Text(this);
sipka1.FontFamily = sadaFontu.Families[0];
sipka1.Pozice = umisteni.Sipka1;
sipka1.TextovyObsah = ">";
sipka1.FontSize = 13f;
sipka1.Kliknuto += new KliknutiHandler(sipka1_Click);
sipka1.Rozmery = (gr.MeasureString(sipka1.TextovyObsah, sipka1.Font)).ToSize();
//----sipka2----
sipka2 = new Text(this);
sipka2.FontFamily = sadaFontu.Families[0];
sipka2.Pozice = umisteni.Sipka2;
sipka2.Kliknuto += new KliknutiHandler(sipka2_Click);
sipka2.FontSize = 13f;
sipka2.TextovyObsah = ">";
sipka2.Rozmery = (gr.MeasureString(sipka2.TextovyObsah, sipka2.Font)).ToSize();
//----sipka3----
sipka3 = new Text(this);
sipka3.FontFamily = sadaFontu.Families[0];
sipka3.Pozice = umisteni.Sipka3;
sipka3.TextovyObsah = ">";
sipka3.FontSize = 13f;
sipka3.Barva = Brushes.Gray;
sipka3.Kliknuto += new KliknutiHandler(sipka3_Click);
sipka3.Rozmery = (gr.MeasureString(sipka3.TextovyObsah, sipka3.Font)).ToSize();
//----sipka4----
sipka4 = new Text(this);
sipka4.FontFamily = sadaFontu.Families[0];
sipka4.Pozice = umisteni.Sipka4;
sipka4.TextovyObsah = ">";
sipka4.FontSize = 13f;
sipka4.Barva = Brushes.Gray;
sipka4.Kliknuto += new KliknutiHandler(sipka4_Click);
sipka4.Rozmery = (gr.MeasureString(sipka4.TextovyObsah, sipka4.Font)).ToSize();
//----sipka5----------tady zacina vyber druhu hrace nebo AI
sipka5 = new Text(this);
sipka5.FontFamily = sadaFontu.Families[0];
sipka5.Pozice = umisteni.Sipka5;
sipka5.TextovyObsah = ">";
sipka5.FontSize = 13f;
sipka5.Kliknuto += new KliknutiHandler(sipka5_Click);
sipka5.Rozmery = (gr.MeasureString(sipka5.TextovyObsah, sipka5.Font)).ToSize();
//----sipka6----
sipka6 = new Text(this);
sipka6.FontFamily = sadaFontu.Families[0];
sipka6.Pozice = umisteni.Sipka6;
sipka6.TextovyObsah = ">";
sipka6.FontSize = 13f;
sipka6.Kliknuto += new KliknutiHandler(sipka6_Click);
sipka6.Rozmery = (gr.MeasureString(sipka6.TextovyObsah, sipka6.Font)).ToSize();
//----sipka7----
sipka7 = new Text(this);
sipka7.FontFamily = sadaFontu.Families[0];
sipka7.Pozice = umisteni.Sipka7;
sipka7.TextovyObsah = ">";
sipka7.FontSize = 13f;
sipka7.Barva = Brushes.Gray;
```

```
sipka7.Kliknuto += new KliknutiHandler(sipka7_Click);
sipka7.Rozmery = (gr.MeasureString(sipka7.TextovyObsah, sipka7.Font)).ToSize();
//----sipka8----
sipka8 = new Text(this);
sipka8.FontFamily = sadaFontu.Families[0];
sipka8.Pozice = umisteni.Sipka8;
sipka8.TextovyObsah = ">";
sipka8.FontSize = 13f;
sipka8.Barva = Brushes.Gray;
sipka8.Kliknuto += new KliknutiHandler(sipka8_Click);
sipka8.Rozmery = (gr.MeasureString(sipka8.TextovyObsah, sipka8.Font)).ToSize();

//------------rezimy------------
//---rezim1----
rezim1 = new Text(this);
rezim1.FontFamily = sadaFontu.Families[0];
rezim1.Pozice = umisteni.Rezim1;
rezim1.TextovyObsah = "P";
rezim1.FontSize = 13f;
rezim1.Rozmery = (gr.MeasureString(rezim1.TextovyObsah, rezim1.Font)).ToSize();
//---rezim2----
rezim2 = new Text(this);
rezim2.FontFamily = sadaFontu.Families[0];
rezim2.Pozice = umisteni.Rezim2;
rezim2.TextovyObsah = "AI";
rezim2.FontSize = 13f;
rezim2.Rozmery = (gr.MeasureString(rezim2.TextovyObsah, rezim2.Font)).ToSize();
//---rezim3----
rezim3 = new Text(this);
rezim3.FontFamily = sadaFontu.Families[0];
rezim3.Pozice = umisteni.Rezim3;
rezim3.TextovyObsah = "P";
rezim3.FontSize = 13f;
rezim3.Barva = Brushes.Gray;
rezim3.Rozmery = (gr.MeasureString(rezim3.TextovyObsah, rezim3.Font)).ToSize();
//---rezim4----
rezim4 = new Text(this);
rezim4.FontFamily = sadaFontu.Families[0];
rezim4.Pozice = umisteni.Rezim4;
rezim4.TextovyObsah = "AI";
rezim4.FontSize = 13f;
rezim4.Barva = Brushes.Gray;
rezim4.Rozmery = (gr.MeasureString(rezim4.TextovyObsah, rezim4.Font)).ToSize();

poleRezimu[0] = rezim1;
poleRezimu[1] = rezim2;
poleRezimu[2] = rezim3;
poleRezimu[3] = rezim4;


//-------------naspisy-----------
//--------nadBarva----------
nadBarva = new Text(this);
nadBarva.FontFamily = sadaFontu.Families[0];
nadBarva.Pozice = umisteni.NadBarva;
nadBarva.TextovyObsah = "barva";
nadBarva.FontSize = 13f;
nadBarva.Rozmery = (gr.MeasureString(nadBarva.TextovyObsah, nadBarva.Font)).ToSize();
//------nadJmeno-------
nadJmeno = new Text(this);
nadJmeno.FontFamily = sadaFontu.Families[0];
nadJmeno.Pozice = umisteni.NadJmeno;
nadJmeno.TextovyObsah = "jméno";
nadJmeno.FontSize = 13f;
nadJmeno.Rozmery = (gr.MeasureString(nadJmeno.TextovyObsah, nadJmeno.Font)).ToSize();
//------nadRezim-------
nadRezim = new Text(this);
nadRezim.FontFamily = sadaFontu.Families[0];
nadRezim.Pozice = umisteni.NadRezim;
nadRezim.TextovyObsah = "režim";
nadRezim.FontSize = 13f;
nadRezim.Rozmery = (gr.MeasureString(nadRezim.TextovyObsah, nadRezim.Font)).ToSize();
//------nadNovaHra-------
nadNovaHra = new Text(this);
nadNovaHra.FontFamily = sadaFontu.Families[0];
nadNovaHra.Pozice = umisteni.NadNovaHra;
```

```
nadNovaHra.TextovyObsah = "NOVÁ HRA >";
nadNovaHra.FontSize = 20f;
nadNovaHra.Rozmery = (gr.MeasureString(nadNovaHra.TextovyObsah, nadNovaHra.Font)).ToSize();

hrac4.Enabled = false;
hrac3.Enabled = false;

//-------------barvy------------
poleSouradnic = umisteni.poleSouradnic;
rozmeryBarvy = umisteni.rozmeryBarvy;

Barva1 = new Barva((Image)Zdroj.barva_cerverna, "cervena");
Barva1.Pozice = poleSouradnic[0];
Barva1.Rozmery = rozmeryBarvy;
Barva1.Obrazek = new Bitmap((Image)Zdroj.barva_cerverna, rozmeryBarvy);
Barva2 = new Barva((Image)Zdroj.barva_zelena, "zelena");
Barva2.Pozice = poleSouradnic[1];
Barva2.Rozmery = rozmeryBarvy;
Barva2.Obrazek = new Bitmap((Image)Zdroj.barva_zelena, rozmeryBarvy);
Barva3 = new Barva((Image)Zdroj.barva_modra, "modra");
Barva3.Pozice = new Point(0, 0);
Barva3.Rozmery = rozmeryBarvy;
Barva3.Obrazek = new Bitmap((Image)Zdroj.barva_modra, rozmeryBarvy);
Barva4 = new Barva((Image)Zdroj.barva_zluta, "zluta");
Barva4.Pozice = new Point(0, 0);
Barva4.Rozmery = rozmeryBarvy;
Barva4.Obrazek = new Bitmap((Image)Zdroj.barva_zluta, rozmeryBarvy);
Barva5 = new Barva((Image)Zdroj.barva_cerna, "cerna");
Barva5.Pozice = new Point(0, 0);
Barva5.Rozmery = rozmeryBarvy;
Barva5.Obrazek = new Bitmap((Image)Zdroj.barva_cerna, rozmeryBarvy);
Barva6 = new Barva((Image)Zdroj.barva_cerna, "cerna");
Barva6.Pozice = poleSouradnic[2];
Barva6.Rozmery = rozmeryBarvy;
Barva6.Obrazek = new Bitmap((Image)Zdroj.barva_cerna, rozmeryBarvy);
Barva7 = new Barva((Image)Zdroj.barva_cerna, "cerna");
Barva7.Pozice = poleSouradnic[3];
Barva7.Rozmery = rozmeryBarvy;
Barva7.Obrazek = new Bitmap((Image)Zdroj.barva_cerna, rozmeryBarvy);
Barva8 = new Barva("prazdna");
Barva8.Rozmery = rozmeryBarvy;
Barva8.Obrazek = new Bitmap((Image)Zdroj.barva_cerna, rozmeryBarvy);

poleBarev[0] = Barva1;
poleBarev[1] = Barva2;
poleBarev[2] = Barva3;
poleBarev[3] = Barva4;
poleBarev[4] = Barva5;//cerna
poleBarev[5] = Barva6;//cerna
poleBarev[6] = Barva7;//cerna
poleBarev[7] = Barva8;//zadna



dostupneBarvy = new Barva[2];
dostupneBarvy[0] = Barva3;//modra
dostupneBarvy[1] = Barva4;//zluta

naFormu = new GrafickyObjekt[27];
naFormu[0] = menuHraci;
naFormu[1] = zacitHru;
naFormu[2] = zpet;
naFormu[3] = cislo1;
naFormu[4] = cislo2;
naFormu[5] = cislo3;
naFormu[6] = cislo4;
naFormu[7] = sipka1;
naFormu[8] = sipka2;
naFormu[9] = sipka3;
naFormu[10] = sipka4;
naFormu[11] = sipka5;
naFormu[12] = sipka6;
naFormu[13] = sipka7;
naFormu[14] = sipka8;
naFormu[15] = rezim1;
naFormu[16] = rezim2;
```

```
                naFormu[17] = rezim3;
                naFormu[18] = rezim4;
                naFormu[19] = nadBarva;
                naFormu[20] = nadJmeno;
                naFormu[21] = nadNovaHra;
                naFormu[22] = nadRezim;
                naFormu[23] = Barva1;
                naFormu[24] = Barva2;
                naFormu[25] = Barva6;
                naFormu[26] = Barva7;

                Barva1.Zobrazena = true;
                Barva2.Zobrazena = true;
                Barva6.Zobrazena = true;
                Barva7.Zobrazena = true;
                hraciNacteno = true;
            }

            break;

        case (int)TYPmenu.pravidla:
            //na obrazovce musim zobrazit vzdy jen tu cast pravidel, ktera se tam rozmerove vejde, tzn na zaklade rozliseni monitoru
a velikosti pisma spocitat pocet pismen, ktera zobrazim.
            //dale je dulezite, aby znaky byly zarovnany

            Graphics Gr = this.CreateGraphics();
            //titulek
            pravidlaTitulek = new Text(this);
            pravidlaTitulek.FontFamily = sadaFontu.Families[0];
            pravidlaTitulek.Pozice = umisteni.Pravidla;
            pravidlaTitulek.TextovyObsah = "PRAVIDLA";
            pravidlaTitulek.FontSize = 14f;
            pravidlaTitulek.Rozmery = (Gr.MeasureString(pravidlaTitulek.TextovyObsah, pravidlaTitulek.Font)).ToSize();

            //sipky pro posouvani textu
            pravidlaSipkaNahoru = new GrafickyObjekt((Image)new Bitmap((Image)Zdroj.SipkaNahoruBila,
umisteni.PravidlaSipkaNahoruRozmery));
            pravidlaSipkaNahoru.Jmeno = "pravidlaSipkaNahoru";
            pravidlaSipkaNahoru.Pozice = umisteni.PravidlaSipkaNahoru;
            pravidlaSipkaNahoru.Rozmery = umisteni.PravidlaSipkaNahoruRozmery;
            pravidlaSipkaNahoru.OpusteniMysi += new OpusteniHandler(pravidlaSipkaNahoru_OpusteniMysi);
            pravidlaSipkaNahoru.PrejetoMysi += new PrejetiNadHandler(pravidlaSipkaNahoru_PrejetoMysi);
            pravidlaSipkaNahoru.StisknutoMysi += new StisknutoHandler(pravidlaSipkaNahoru_StisknutoMysi);
            pravidlaSipkaNahoru.UvolnenoMysi += new UvolnenoHandler(pravidlaSipkaNahoru_UvolnenoMysi);

            pravidlaSipkaDolu = new GrafickyObjekt((Image)new Bitmap((Image)Zdroj.SipkaDoluBila,
umisteni.PravidlaSipkaDoluRozmery));
            pravidlaSipkaDolu.Jmeno = "pravidlaSipkaDolu";
            pravidlaSipkaDolu.Pozice = umisteni.PravidlaSipkaDolu;
            pravidlaSipkaDolu.Rozmery = umisteni.PravidlaSipkaDoluRozmery;
            pravidlaSipkaDolu.OpusteniMysi += new OpusteniHandler(pravidlaSipkaDolu_OpusteniMysi);
            pravidlaSipkaDolu.PrejetoMysi += new PrejetiNadHandler(pravidlaSipkaDolu_PrejetoMysi);
            pravidlaSipkaDolu.StisknutoMysi += new StisknutoHandler(pravidlaSipkaDolu_StisknutoMysi);
            pravidlaSipkaDolu.UvolnenoMysi += new UvolnenoHandler(pravidlaSipkaDolu_UvolnenoMysi);

            //pravidla samotna
            pravidlaText = new MultiLineText(this);
            pravidlaText.FontFamily = sadaFontu.Families[0];
            pravidlaText.Pozice = umisteni.PravidlaText;
            pravidlaText.FontSize = 13f;
            pravidlaText.Rozmery = new Size(umisteni.PravidlaTextHranice.Koncove.X -
umisteni.PravidlaTextHranice.Pocatecni.X, umisteni.PravidlaTextHranice.Koncove.Y - umisteni.PravidlaTextHranice.Pocatecni.Y);


            pravidlaObsah = " Každý hráč má za úkol najít v labyrintu nejlepší cestu k jednotlivým artefaktům, které si na začátku
hry vylosoval. Ovšem pozor, cesty se neustále posouvají a mění svůj tvar. Vyhrává ten, kdo první sebere všechny artefakty. Na počátku hry
se kartičky s chodbami náhodně promíchají a následně rozprostřou po hrací ploše takovým způsobem, aby vznikl náhodný vzor chodeb.
Vždy je jedna kartička navíc, neboť se používá během hry k zasouvání do řad. Všech 24 tajných artefaktů se zamíchá a rozdá rovnoměrně
všem hráčům. Táhnout začíná hráč, který byl hrou určen jako první. Poté je pořadí hráčů určeno směrem otáčení hodinových ručiček. Na
pořadí má rovněž vliv zvolená barva figurky, jejíž startovní pozice na hracím poli je neměnná. Jakmile přijde na hráče řada, hra mu ukáže
jeho cíl, ke kterému se musí dostat. Aby hráč došel k artefaktu, musí postupně měnit chodby bludiště vsouváním kartičky a potom
postupovat svou figurkou. Zasunutí je na rozdíl od posunutí figurkou povinné. Pokud tak hráč neučiní, hra za něj náhodně zasune do
libovolné řady, kdy již nemá možnost táhnout figurkou. Zasunutí do aktivních řad je vskutku prosté. Hráč vsune po straně přebytečnou hrací
kartičku do hracího pole tak, aby se na protější straně opět vysunula jedna hrací kartička. Tato kartička zůstává na kraji hracího plánu ležet
tak dlouho, dokud není další hráč na tahu, který nesmí kartičku vsunout na stejném místě, kde byla právě vysunuta. To znamená, že se nesmí
takto zrušit tah předchozího hráče. Místa, kde je možné vsunout kartičky do hracího pole, jsou označena šipkou. V zápětí na to, když hráč
```

posunul chodby, postupuje figurkou. Hráč se smí přesunout na jakoukoliv kartičku bludiště, která je s výchozím bodem spojen nepřerušenou chodbou. Hráč nemusí figurkou postoupit, může tak zůstat na místě. Když hráč nemůže přímou cestou dojít k artefaktu, projde figurkou tak daleko, aby měl pokud možno výhodnou pozici pro příští tah. Pokud hráč došel k cíli, artefakt se přemístí do seznamu najitých artefaktů. Jakmile na hráče přijde opět řada, hra mu odkryje další tajemnost, kterou musí najít. Jakmile hráč najde všechny svoje tajemné artefakty, musí se dostat s figurkou opět k výchozímu bodu v rohu hracího pole. Vítězí hráč, který jako první najde všechny artefakty a dojde s figurkou na výchozí bod.";

```
                pravidlaText.TextovyObsah = pravidlaObsah;


                //pridam do pole pro vykreslovani
                naFormuPravidla = new GrafickyObjekt[5];
                naFormuPravidla[0] = menuUvod;
                naFormuPravidla[1] = pravidlaTitulek;
                naFormuPravidla[2] = pravidlaSipkaNahoru;
                naFormuPravidla[3] = pravidlaSipkaDolu;
                naFormuPravidla[4] = pravidlaText;
                break;
            }
        }
        catch (System.ArgumentNullException)
        {
            MessageBox.Show("null");
        }
        catch (System.ArgumentOutOfRangeException)
        {
            MessageBox.Show("out of range");
        }
        catch
        {
            MessageBox.Show("chyba v nacitani grafiky");
            ukoncit = true;
        }
    }

    void pravidlaSipkaDolu_UvolnenoMysi(object volajici)
    {
        throw new NotImplementedException();
    }

    void pravidlaSipkaDolu_StisknutoMysi(object volajici)
    {
        pravidlaText.PosunObsah(MultiLineText.Smer.dolu);
        this.Refresh();
    }

    void pravidlaSipkaDolu_PrejetoMysi(object volajici)
    {
        throw new NotImplementedException();
    }

    void pravidlaSipkaDolu_OpusteniMysi(object volajici)
    {
        throw new NotImplementedException();
    }

    void pravidlaSipkaNahoru_UvolnenoMysi(object volajici)
    {
        throw new NotImplementedException();
    }

    void pravidlaSipkaNahoru_StisknutoMysi(object volajici)
    {
        pravidlaText.PosunObsah(MultiLineText.Smer.nahoru);
        this.Refresh();
    }

    void pravidlaSipkaNahoru_PrejetoMysi(object volajici)
    {
        throw new NotImplementedException();
    }

    void pravidlaSipkaNahoru_OpusteniMysi(object volajici)
    {
        throw new NotImplementedException();
    }
```

```
void zacitHru_OpusteniMysi(object volajici)
{
   zacitHru.Barva = Brushes.White;
   zacitHruBarevna= false;
   this.Refresh();
}

bool zacitHruBarevna = false;
void zacitHru_PrejetoMysi(object volajici)
{
   if (!zacitHruBarevna)
   {
      SolidBrush brush = new SolidBrush(Color.Gray);//(Color.FromArgb(10, 215, 0));
      zacitHru.Barva = (Brush)brush;
      zacitHruBarevna = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = zacitHru;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = zacitHru;
      }
      this.Refresh();
   }
}

void zpet_OpusteniMysi(object volajici)
{
   zpet.Barva = Brushes.White;
   zpetBarevna = false;
   this.Refresh();
}

bool zpetBarevna = false;
void zpet_PrejetoMysi(object volajici)
{
   if (!zpetBarevna)
   {
      SolidBrush brush = new SolidBrush(Color.Gray);
      zpet.Barva = (Brush)brush;
     zpetBarevna = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = zpet;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = zpet;
      }
      this.Refresh();
   }
}

void konec_OpusteniMysi(object volajici)
{
   konec.Barva = Brushes.White;
   konecBarevna = false;
   this.Refresh();
}

void autori_OpusteniMysi(object volajici)
{
   autori.Barva = Brushes.White;
   autoriBarevna = false;
   this.Refresh();
}

void pravidla_OpusteniMysi(object volajici)
{
   pravidla.Barva = Brushes.White;
   pravidlaBarevna = false;
   this.Refresh();
}
```

```
void novaHra_OpusteniMysi(object volajici)
{
   novaHra.Barva = Brushes.White;
   novaHraZelena = false;
   this.Refresh();
}

bool konecBarevna = false;
void konec_PrejetoMysi(object volajici)
{
   if (!konecBarevna)
   {
      SolidBrush brush = new SolidBrush(Color.Gray);
      konec.Barva = (Brush)brush;
      konecBarevna = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = konec;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = konec;
      }
      this.Refresh();
   }
}

bool autoriBarevna = false;
void autori_PrejetoMysi(object volajici)
{
   if (!autoriBarevna)
   {
      SolidBrush brush = new SolidBrush(Color.Gray);
      autori.Barva = (Brush)brush;
      autoriBarevna = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = autori;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = autori;
      }
      this.Refresh();
   }
}

bool pravidlaBarevna = false;
void pravidla_PrejetoMysi(object volajici)
{
   if (!pravidlaBarevna)
   {
      SolidBrush brush = new SolidBrush(Color.Gray);
      pravidla.Barva = (Brush)brush;
      pravidlaBarevna = true;
      if (predeslyNajetyObjekt == null)//prechazim s plochy
      {
         predeslyNajetyObjekt = pravidla;
      }
      else//predtim jsem byl na jinem objektu
      {
         predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
         predeslyNajetyObjekt = pravidla;
      }
      this.Refresh();
   }
}

bool novaHraZelena = false;
void novaHra_PrejetoMysi(object volajici)
{
   if (!novaHraZelena)
   {
```

```
            SolidBrush brush = new SolidBrush(Color.Gray);
            novaHra.Barva = (Brush)brush;
            novaHraZelena = true;
            if (predeslyNajetyObjekt == null)//prechazim s plochy
            {
               predeslyNajetyObjekt = novaHra;
            }
            else//predtim jsem byl na jinem objektu
            {
               predeslyNajetyObjekt.SpoustecUdalostiOpusteniMysi(predeslyNajetyObjekt);
               predeslyNajetyObjekt = novaHra;
            }
            this.Refresh();
         }
      }

      private void PrizpusobRozliseni()
      {
         //zjistim aktualni rozliseni displeje a nastavim pozici a velikost pro zobrazeni
         Rectangle rozliseni = Screen.PrimaryScreen.Bounds;
         Rectangle pracivniPlocha = Screen.PrimaryScreen.WorkingArea;

         this.StartPosition = FormStartPosition.Manual;

         //1,6 je sirokouhly monitor s pomerem 16:10
         PomerStran pomer = PomerStran.klasicke;
         float pomerStran = (float)rozliseni.Width / (float)rozliseni.Height;
         const float sestnactKuDeviti = 1.77777778f;
         const float sestnactKuDesiti = 1.6f;
         const float patnactKuDeviti = 1.66666667f;
         const float ctyriKuTrem = 1.33333337f;
         const float petKuCtyrem = 1.25f;
         //MessageBox.Show("Tvoje rozliseni je " + rozliseni.Width.ToString() + " x " + rozliseni.Height.ToString() + " a pomer stran je
" + pomerStran.ToString());

         if (pomerStran == sestnactKuDeviti || pomerStran == sestnactKuDesiti || pomerStran == patnactKuDeviti)
         {
            pomer = PomerStran.sirokouhle;
         }
         else if (pomerStran == ctyriKuTrem)
         {
            pomer = PomerStran.klasicke;
         }
         else if (pomerStran == petKuCtyrem)
         {
            pomer = PomerStran.petKuCtyrem;
         }
         else
         {
            MessageBox.Show("Litujeme, nemáte nastaveno podporované rozlišení. Tyto jsou 1280x800, 1680x1050.");
            ukoncit = true;
         }

         if (!ukoncit)
         {
            //na zaklade uhlosti monitoru, tj. zda je sirokouhly ci ne, priradim jednotliva rozliseni
            switch (pomer)
            {
               //sirokouhle
               case PomerStran.sirokouhle:
                  if (rozliseni.Width == 1280 && rozliseni.Height == 800)
                  {
                     this.Size = new Size(908, 760);
                     //umisteni na stred
                     int zleva = (pracivniPlocha.Width - this.Width) / 2;
                     int shora = (pracivniPlocha.Height - this.Height) / 2;
                     //umistim na stred
                     this.DesktopLocation = new Point(zleva, shora);

                     naFormuUvod[0].Obrazek = new Bitmap((Image)Zdroj.menu1280x800); //((Image)Zdroj.menu, new Size(908, 760));
                     naFormuUvod[0].Rozmery = new Size(908, 760);

                     umisteni = new MenuDvanactSetXOsmSet(this.CreateGraphics(), novaHra, pravidla, autori, konec);

                     umisteni.Vycentruj();
                  }
```

```
            else if (rozliseni.Width == 1680 && rozliseni.Height == 1050)
            {
               this.Size = new Size(908, 760);
               //umisteni na stred
               int zleva = (pracivniPlocha.Width - this.Width) / 2;
               int shora = (pracivniPlocha.Height - this.Height) / 2;
               //umistim na stred
               this.DesktopLocation = new Point(zleva, shora);

               naFormuUvod[0].Obrazek = new Bitmap((Image)Zdroj.menu1280x800);//((Image)Zdroj.menu, new Size(908, 760));
               naFormuUvod[0].Rozmery = new Size(908, 760);

               umisteni = new MenuDvanactSetXOsmSet(this.CreateGraphics(), novaHra, pravidla, autori, konec);
               umisteni.Vycentruj();
            }
            else//defaultni rozmery pro nepodporovane sirokouhle rozliseni
            {
               this.Size = new Size(908, 760);
               //umisteni na stred
               int zleva = (pracivniPlocha.Width - this.Width) / 2;
               int shora = (pracivniPlocha.Height - this.Height) / 2;
               //umistim na stred
               this.DesktopLocation = new Point(zleva, shora);

               naFormuUvod[0].Obrazek = new Bitmap((Image)Zdroj.menu1280x800);
               naFormuUvod[0].Rozmery = new Size(908, 760);

               umisteni = new MenuDvanactSetXOsmSet(this.CreateGraphics(), novaHra, pravidla, autori, konec);
               umisteni.Vycentruj();
            }
            break;
         case PomerStran.klasicke:
            if (rozliseni.Width == 1024 && rozliseni.Height == 768)
            {
               this.Size = new Size(873, 730);
               //umisteni na stred
               int zleva = (pracivniPlocha.Width - this.Width) / 2;
               int shora = (pracivniPlocha.Height - this.Height) / 2;
               //umistim na stred
               this.DesktopLocation = new Point(zleva, shora);

               naFormuUvod[0].Obrazek = new Bitmap((Image)Zdroj.menu1024x768);
               naFormuUvod[0].Rozmery = new Size(873, 730);

               umisteni = new MenuTisicDvacetCtyriXSedmsetSedesatOsm(this.CreateGraphics(), novaHra, pravidla, autori, konec);
               umisteni.Vycentruj();
            }
            break;
         case PomerStran.petKuCtyrem:
            if (rozliseni.Width == 1280 && rozliseni.Height == 1024)
            {
               this.Size = new Size(873, 730);
               //umisteni na stred
               int zleva = (pracivniPlocha.Width - this.Width) / 2;
               int shora = (pracivniPlocha.Height - this.Height) / 2;
               //umistim na stred
               this.DesktopLocation = new Point(zleva, shora);

               naFormuUvod[0].Obrazek = new Bitmap((Image)Zdroj.menu1024x768);
               naFormuUvod[0].Rozmery = new Size(873, 730);

               umisteni = new MenuTisicDvacetCtyriXSedmsetSedesatOsm(this.CreateGraphics(), novaHra, pravidla, autori, konec);
               umisteni.Vycentruj();
            }

            break;
      }
   }

}

void novaHra_Kliknuto(object volajici)
{
   this.typMenu = TYPmenu.hraci;
   this.NactiObjety();
   this.Refresh();
```

```
}

void UvodniMenu_Load(object sender, EventArgs e)
{
   if (ukoncit) { Application.Exit(); }
}

void sipka8_Click(object volajici)
{
   if (cislo4.TextovyObsah != "X")
   {
      if (rezim4.TextovyObsah == "AI")
      {
         rezim4.TextovyObsah = "P";
      }
      else
      {
         rezim4.TextovyObsah = "AI";
      }
      this.Refresh();
   }
}

void sipka7_Click(object volajici)
{
   if (cislo3.TextovyObsah != "X")
   {
      if (rezim3.TextovyObsah == "AI")
      {
         rezim3.TextovyObsah = "P";
      }
      else
      {
         rezim3.TextovyObsah = "AI";
      }
      this.Refresh();
   }
}

void sipka6_Click(object volajici)
{
   if (cislo2.TextovyObsah != "X")
   {
      if (rezim2.TextovyObsah == "AI")
      {
         rezim2.TextovyObsah = "P";
      }
      else
      {
         rezim2.TextovyObsah = "AI";
      }
      this.Refresh();
   }
}

void sipka5_Click(object volajici)
{
   if (cislo1.TextovyObsah != "X")
   {
      if (rezim1.TextovyObsah == "AI")
      {
         rezim1.TextovyObsah = "P";//hrac
      }
      else//umela inteligence
      {
         rezim1.TextovyObsah = "AI";
      }
      this.Refresh();
   }
}

void sipka4_Click(object volajici)
{
   if (cislo4.TextovyObsah != "X")
   {
      for (int i = 0; i < poleBarev.Length; i++)
```

```
        {
            if (poleBarev[i].Pozice == poleSouradnic[3])
            {
                ZobrazDalsiBarvu(poleBarev[i], 4);
                this.Refresh();
                break;
            }
        }
    }
}

void sipka3_Click(object volajici)
{
    if (cislo3.TextovyObsah != "X")
    {
        for (int i = 0; i < poleBarev.Length; i++)
        {
            if (poleBarev[i].Pozice == poleSouradnic[2])
            {
                ZobrazDalsiBarvu(poleBarev[i], 3);
                this.Refresh();
                break;
            }
        }
    }
}

void sipka2_Click(object volajici)
{
    if (cislo2.TextovyObsah != "X")
    {
        for (int i = 0; i < poleBarev.Length; i++)
        {
            if (poleBarev[i].Pozice == poleSouradnic[1])
            {
                ZobrazDalsiBarvu(poleBarev[i], 2);
                this.Refresh();
                break;
            }
        }
    }
}

void sipka1_Click(object volajici)
{
    if (cislo1.TextovyObsah != "X")
    {
        //vyhledam barvu, ktera se tu nachazi
        for (int i = 0; i < poleBarev.Length; i++)
        {
            if (poleBarev[i].Pozice == poleSouradnic[0])
            {
                ZobrazDalsiBarvu(poleBarev[i], 1);
                this.Refresh();
                break;
            }
        }
    }
}

void cislo4_Click(object volajici)
{
    //kliknul jsem na cislo, takze musim zobrazit krizek
    if (cislo4.TextovyObsah == "4")
    {
        if (cislo3.TextovyObsah == "3")
        {
            for (int i = 0; i < poleBarev.Length; i++)
            {
                if (poleBarev[i].Pozice == poleSouradnic[3])
                {
                    zacernit(poleBarev[i], 4);
                    break;
                }
            }
            cislo4.TextovyObsah = "X";
```

```
            sipka4.Barva = Brushes.Gray;
            hrac4.Enabled = false;
            sipka8.Barva = Brushes.Gray;
            rezim4.Barva = Brushes.Gray;
            this.Refresh();
        }
    }
    else if (cislo4.TextovyObsah == "X")
    {
        if (cislo3.TextovyObsah == "3")
        {
            cislo4.TextovyObsah = "4";
            sipka4.Barva = Brushes.White;
            hrac4.Enabled = true;
            sipka8.Barva = Brushes.White;
            rezim4.Barva = Brushes.White;
            for (int i = 0; i < poleBarev.Length; i++)
            {
                if (poleBarev[i].Pozice == poleSouradnic[3])
                {
                    ZobrazDalsiBarvu(poleBarev[i], 4);
                    break;
                }
            }
        }
    }
}

void cislo3_Click(object volajici)
{
    //kliknul jsem na cislo, takze musim zobrazit krizek
    if (cislo3.TextovyObsah == "3")
    {
        if (cislo4.TextovyObsah == "X")
        {
            for (int i = 0; i < poleBarev.Length; i++)
            {
                if (poleBarev[i].Pozice == poleSouradnic[2])
                {
                    zacernit(poleBarev[i], 3);
                    break;
                }
            }
            cislo3.TextovyObsah = "X";
            sipka3.Barva = Brushes.Gray;
            hrac3.Enabled = false;
            sipka7.Barva = Brushes.Gray;
            rezim3.Barva = Brushes.Gray;
            this.Refresh();
        }
    }
    else if (cislo3.TextovyObsah == "X")
    {
        if (cislo2.TextovyObsah == "2")
        {
            cislo3.TextovyObsah = "3";
            sipka3.Barva = Brushes.White;
            hrac3.Enabled = true;
            sipka7.Barva = Brushes.White;
            rezim3.Barva = Brushes.White;
            for (int i = 0; i < poleBarev.Length; i++)
            {
                if (poleBarev[i].Pozice == poleSouradnic[2])
                {
                    ZobrazDalsiBarvu(poleBarev[i], 3);
                    break;
                }
            }
        }
    }
}

void cislo2_Click(object volajici)
{
    //kliknul jsem na cislo, takze musim zobrazit krizek
    if (cislo2.TextovyObsah == "2")
```

```
            {
                if (cislo3.TextovyObsah == "X")
                {
                    for (int i = 0; i < poleBarev.Length; i++)
                    {
                        if (poleBarev[i].Pozice == poleSouradnic[1])
                        {
                            zacernit(poleBarev[i], 2);
                            break;
                        }
                    }
                    cislo2.TextovyObsah = "X";
                    sipka2.Barva = Brushes.Gray;
                    hrac2.Enabled = false;
                    sipka6.Barva = Brushes.Gray;
                    rezim2.Barva = Brushes.Gray;
                    this.Refresh();
                }
            }
            else if (cislo2.TextovyObsah == "X")
            {
                cislo2.TextovyObsah = "2";
                sipka2.Barva = Brushes.White;
                hrac2.Enabled = true;
                sipka6.Barva = Brushes.White;
                rezim2.Barva = Brushes.White;
                for (int i = 0; i < poleBarev.Length; i++)
                {
                    if (poleBarev[i].Pozice == poleSouradnic[1])
                    {
                        ZobrazDalsiBarvu(poleBarev[i], 2);
                        break;
                    }
                }
            }
        }


        void konec_Kliknuto(object volajici)
        {
            Application.Exit();
        }

        void autori_Kliknuto(object volajici)
        {
            MessageBox.Show("Sykric");
        }

        void pravidla_Kliknuto(object volajici)
        {
            this.typMenu = TYPmenu.pravidla;
            this.NactiObjety();
            this.Refresh();
        }

        void zacitHru_Click(object volajici)
        {
            //uzivatel chce zacit hru
            //projdu prvky a zjistim, kolik jich uzivatel vyplnil

            //jelikoz hodnoty ukladam do staticke tridy, hodnoty tam zustanou porad, takze se muze stat, ze tuto metodu spoustim nekolikrat
a proto hodnoty radeji nastavim na null
            inter.Ai = null;
            inter.Barvy = null;
            inter.Nicks = null;
            //projdu vsechny, ale kontrolovat budu jen ty aktivni
            for (int i = 0; i < 4; i++)
            {
                if (poleCisel[i].TextovyObsah == (i + 1).ToString())//rada je funkcni
                {
                    //kontrola, zda je spravne vyplnen nick
                    if (nicky[i].Text == "")
                    {
                        MessageBox.Show("U hráče číslo " + (i + 1).ToString() + " není vyplněn nick!", "Chybějící nick",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                        return;
```

```
                }
                //delka nesmi presahovat 10 znaku
                if (nicky[i].Text.Length > 10)
                {
                    MessageBox.Show("Maximální délka nicku je 10 znaků! Upravte, prosím, nick u hráče číslo " + (i + 1).ToString(), "Příliš
dlouhý nick", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    return;

                }
                //nick je v poradku
                //kontrola probehla uspesne, takze nyni prekopiruju nick a k nemu prirazeny obrazek do docasneho uloziste dat
                if (inter.Nicks == null)
                {
                    inter.Nicks = new string[1];
                    inter.Nicks[0] = nicky[i].Text;
                }
                else
                {
                    String[] tmp = inter.Nicks;
                    inter.Nicks = new string[inter.Nicks.Length + 1];
                    tmp.CopyTo(inter.Nicks, 0);
                    inter.Nicks[inter.Nicks.Length - 1] = nicky[i].Text;
                }
                //podle radku zjistim souradnice, ktere by barva mela mit, a podle nic najdu tu spravnou barvu a pak zkontroluju, jestli je
viditelna...pro jistotu
                Color barva = Color.Transparent;
                for (int k = 0; k < poleBarev.Length; k++)
                {
                    if (poleBarev[k].Pozice == poleSouradnic[i])
                    {
                        if (poleBarev[k].Zobrazena)
                        {
                            switch (poleBarev[k].Popis)
                            {
                                case "cervena":
                                    barva = Color.Red;
                                    break;
                                case "zelena":
                                    barva = Color.Green;
                                    break;
                                case "modra":
                                    barva = Color.Blue;
                                    break;
                                case "zluta":
                                    barva = Color.Yellow;
                                    break;
                            }
                            break;
                        }
                    }
                }
                if (inter.Barvy == null)
                {
                    inter.Barvy = new Color[1];
                    inter.Barvy[0] = barva;
                }
                else
                {
                    Color[] tmp = inter.Barvy;
                    inter.Barvy = new Color[inter.Barvy.Length + 1];
                    tmp.CopyTo(inter.Barvy, 0);
                    inter.Barvy[inter.Barvy.Length - 1] = barva;
                }

                if (inter.Ai == null)
                {
                    inter.Ai = new bool[1];
                    switch (poleRezimu[i].TextovyObsah)
                    {
                        case "P":
                            inter.Ai[0] = false;
                            break;
                        case "AI":
                            inter.Ai[0] = true;
                            break;
                    }
```

```
                }
                else
                {
                    bool[] tmp = inter.Ai;
                    inter.Ai = new bool[inter.Ai.Length + 1];
                    tmp.CopyTo(inter.Ai, 0);
                    switch (poleRezimu[i].TextovyObsah)
                    {
                        case "P":
                            inter.Ai[inter.Ai.Length - 1] = false;
                            break;
                        case "AI":
                            inter.Ai[inter.Ai.Length - 1] = true;
                            break;
                    }
                }

            }
        }
        //minimalni pocet hracu je 2
        if (inter.Nicks.Length < 2)
        {
            MessageBox.Show("Minimální počet hráčů je 2! Přidejte do hry alespoň jednoho hráče.", "Málo hráču ve hře",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }
        //nicky jsou nyni ulozene, ale musim zkontrolovat, zda nejsou nejake stejne, nesmi se totiz opakovat
        //zkontroluju kazdy s kazdym-nejjednodussi reseni
        for (int j = 0; j < inter.Nicks.Length; j++)
        {
            for (int k = j + 1; k < inter.Nicks.Length; k++)
            {
                if (inter.Nicks[j] == inter.Nicks[k])
                {
                    MessageBox.Show("Zadali jste stejné nicky!", "Chyba nicků", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    return;
                }
            }
        }



        VstupniBodProgramu.vlakno.Priority = ThreadPriority.Normal;
        VstupniBodProgramu.vlakno.Start();
        Application.ExitThread();
        this.Dispose();
    }

    void zpet_Click(object volajici)
    {
        //uzivatel se chce vratit na uvodni menu
        //spustim nove vlakno, ve kterem pobezi form
        this.typMenu = TYPmenu.uvodni;
        this.NactiObjety();
        this.Refresh();
    }

    protected override void OnPaint(PaintEventArgs e)
    {
        if (typMenu == TYPmenu.hraci)
        {
            for (int i = 0; i < naFormu.Length; i++)
            {
                if (naFormu[i] != null)
                {
                    naFormu[i].Vykresli(e.Graphics);
                }
            }
        }
        else if(typMenu==TYPmenu.uvodni)
        {
            for (int i = 0; i < naFormuUvod.Length; i++)
            {
                naFormuUvod[i].Vykresli(e.Graphics);
            }
        }
```

```
    else if (typMenu == TYPmenu.pravidla)
    {
        for (int i = 0; i < naFormuPravidla.Length; i++)
        {
            naFormuPravidla[i].Vykresli(e.Graphics);
        }
    }
}

private void ZobrazDalsiBarvu(Barva barva, int sender)
{
    if (dostupneBarvy.Length != 0)
    {
        if (barva.Popis != "cerna")
        {
            Barva tmp = barva;//ulozim soucasnou barvu
            tmp.Pozice = new Point(0, 0);
            tmp.Zobrazena = false;
            dostupneBarvy[0].Pozice = poleSouradnic[sender - 1];//hned te prvni jsem priradil souradnice pro vykresleni
            dostupneBarvy[0].Zobrazena = true;
            //odeberu ji z pole pro vykresleni a nahradim jej noou barvou
            for (int i = 0; i < naFormu.Length; i++)
            {
                if (naFormu[i] == tmp)
                {
                    naFormu[i] = dostupneBarvy[0];
                }
            }

            //vsechny barvy posunu o jednu dopredu
            for (int i = 0; i < dostupneBarvy.Length - 1; i++)
            {
                dostupneBarvy[i] = dostupneBarvy[i + 1];
            }
            dostupneBarvy[dostupneBarvy.Length - 1] = tmp;//na poslendni misto pridam tu, ktera byla predtim zobrazena

        }
        else//byla zablokovana, tedy cerna
        {
            dostupneBarvy[0].Pozice = poleSouradnic[sender - 1];//zobrazim hned tu prvni povolenou
            dostupneBarvy[0].Zobrazena = true;
            barva.Pozice = new Point(0, 0);
            barva.Zobrazena = false;
            for (int i = 0; i < naFormu.Length; i++)
            {
                if (naFormu[i] == barva)
                {
                    naFormu[i] = dostupneBarvy[0];
                }
            }
            //vsechny barvy posunu o jednu dopredu
            Barva[] tmp = dostupneBarvy;
            dostupneBarvy = new Barva[dostupneBarvy.Length - 1];
            for (int i = 0; i < dostupneBarvy.Length; i++)
            {
                dostupneBarvy[i] = tmp[i + 1];
            }
        }
        this.Refresh();
    }

}

private void zacernit(Barva barva, int sender)
{
    //barvu, ktera tady je, ulozim jako dostupnou a pak ji vypnu a zobrazim cernou
    Barva aktualni = barva;
    aktualni.Pozice = new Point(0, 0);
    aktualni.Zobrazena = false;
    Barva[] tmp = dostupneBarvy;
    dostupneBarvy = new Barva[tmp.Length + 1];
    for (int i = 0; i < tmp.Length; i++)
    {
        dostupneBarvy[i + 1] = tmp[i];
    }
    dostupneBarvy[0] = aktualni;//na prvni pozici ulozim tu, ktera tady prave byla
```

```
            for (int i = 0; i < naFormu.Length; i++)
            {
                if (naFormu[i] == aktualni)
                {
                    naFormu[i] = poleBarev[sender + 2];
                    poleBarev[sender + 2].Zobrazena = true;
                    naFormu[i].Pozice = poleSouradnic[sender - 1];
                }
            }
            this.Refresh();
        }
    }


    //trida pro barvy
    class Barva : GrafickyObjekt
    {
        public Barva(string Popis)
            : base()
        {
            this.popis = Popis;
        }
        public Barva(Image Obrazek, string Popis)
            : base(Obrazek)
        {
            this.popis = Popis;
            this.zobrazena = false;
        }

        private string popis;
        public string Popis
        {
            get
            {
                return popis;
            }
        }
        private bool zobrazena;
        public bool Zobrazena
        {
            get
            {
                return zobrazena;
            }
            set
            {
                zobrazena = value;
            }
        }
    }

    //trida pro zobrazovani viceradkoveho textu
    class MultiLineText : GrafickyObjekt
    {
        public MultiLineText(Form Form)
            : base()
        {
            this.cisloFontu = inter.CisloFontu;
            this.grafika = Form.CreateGraphics();
            this.form = Form;
            this.fontSize = 14f;
            this.barva = Brushes.White;
        }
        private float fontSize;
        public float FontSize
        {
            get
            {
                return fontSize;
            }
            set
            {
                fontSize = value;
                this.font = new Font(fontFamily, fontSize);
                viditelneRadky = this.rozmery.Height / grafika.MeasureString("A", this.font).ToSize().Height;
            }
```

```
}
private string text = "";
public string TextovyObsah
{
   get
   {
      return text;
   }
   set
   {
      text = value;
      text = text.ToUpper();
      //text rozlozim na jednotliva slova, ktera ulozim do pole
      //pote zjistim sirku jednotlivych slov a hodnoty ulozim taky do pole

      string[] slova = new string[0];
      Size[] delky = new Size[0];
      bool posledni = false;
      int pozice, predeslaPozice = 0, pocetZnaku = 0;
      char[] tmp = new char[0];
      while (posledni == false)
      {
         pozice = text.IndexOf(" ", predeslaPozice + 1);

         //kdyz nebyla nalezena mezera, tak se nachazim na poslednim slove
         if (pozice == -1)
         {
            pocetZnaku = text.Length - predeslaPozice;
            posledni = true;
         }
         else
         {
            pocetZnaku = pozice - predeslaPozice;
         }
         tmp = new char[pocetZnaku];
         text.CopyTo(predeslaPozice, tmp, 0, pocetZnaku);

         string[] tmpSlova = new string[slova.Length];
         slova.CopyTo(tmpSlova, 0);
         slova = new string[slova.Length + 1];
         tmpSlova.CopyTo(slova, 0);

         //prevod znaku na retezec
         string slovo = "";
         for (int i = 0; i < tmp.Length; i++)
         {
            slovo += tmp[i].ToString();
         }

         slova[slova.Length - 1] = slovo;
         predeslaPozice = pozice;

         //zjistim delku slova a ulozim ji
         Size[] tmpDelky = new Size[delky.Length];
         delky.CopyTo(tmpDelky, 0);
         delky = new Size[delky.Length + 1];
         tmpDelky.CopyTo(delky, 0);

         Size delka = grafika.MeasureString(slovo, this.Font).ToSize();
         delky[delky.Length - 1] = delka;

      }

      //zjistim, kolik slov se vejde na radek

      int sirkaRadku = this.rozmery.Width;
      int vyskaRadku = this.rozmery.Height;
      int pocetRadku = vyskaRadku / delky[0].Height;//tolik se jich vejde na zobrazeni


      int index = 0, minulyIndex = 0;
      bool posledniSlovo = false;

      while (posledniSlovo == false)
      {
         if (index == slova.Length)
```

```
                {
                    posledniSlovo = true;
                }
                else
                {
                    int volnyProstor = sirkaRadku;

                    Radek[] temp = new Radek[radky.Length];
                    radky.CopyTo(temp, 0);
                    radky = new Radek[radky.Length + 1];
                    temp.CopyTo(radky, 0);
                    radky[radky.Length - 1] = new Radek();
                    radky[radky.Length - 1].FontFamily = this.fontFamily;

                    //vypocitam pozici radku
                    //prvni radek ma pozici nahore, kazdy dalsi ma pozici pod predchozim + vyska
                    Point souradnice;
                    if (radky.Length == 1)
                    {
                        souradnice = this.pozice;
                    }
                    else
                    {
                        souradnice = new Point(radky[radky.Length - 2].Pozice.X, radky[radky.Length - 2].Pozice.Y + delky[0].Height);
                    }
                    radky[radky.Length - 1].Pozice = souradnice;
                    radky[radky.Length - 1].FontSize = this.fontSize;

                    while (volnyProstor > 0)
                    {
                        if (index == slova.Length)
                        {
                            break;
                        }
                        volnyProstor -= delky[index].Width;
                        index += 1;
                    }
                    string obsah = "";

                    for (int i = minulyIndex; i < index; i++)
                    {
                        obsah += slova[i];
                    }
                    radky[radky.Length - 1].Text = obsah;
                    minulyIndex = index;
                }
            }
            viditelnostOd = 0;
            //kdyz je lichy pocet radku, tak pridam jeden na konec, aby byl sudy pocet
            if (radky.Length % 2 == 1)
            {
                Radek[] temp = new Radek[radky.Length];
                radky.CopyTo(temp, 0);
                radky = new Radek[radky.Length + 1];
                temp.CopyTo(radky, 0);
                radky[radky.Length - 1] = new Radek();
                radky[radky.Length - 1].FontFamily = this.fontFamily;
                radky[radky.Length - 1].Pozice = new Point(radky[radky.Length - 2].Pozice.X, radky[radky.Length - 2].Pozice.Y +
delky[0].Height);
                radky[radky.Length - 1].FontSize = this.fontSize;
                radky[radky.Length - 1].Text = "";
            }
        }
    }
    private Font font;
    public Font Font
    {
        get
        {
            return font;
        }
    }
    private FontFamily fontFamily;
    public FontFamily FontFamily
    {
        get
```

```
    {
        return fontFamily;
    }
    set
    {
        fontFamily = value;
        font = new Font(fontFamily, fontSize);
        viditelneRadky = rozmery.Height / grafika.MeasureString("A", font).ToSize().Height;
        viditelnostDo = viditelneRadky - 1;
    }
}
private int cisloFontu;
private Graphics grafika;
private Form form;
private Brush barva;
public Brush Barva
{
    get
    {
        return barva;
    }
    set
    {
        barva = value;
    }
}
public override Size Rozmery
{
    get
    {
        return base.Rozmery;
    }
    set
    {
        base.Rozmery = value;
        viditelneRadky = rozmery.Height / grafika.MeasureString("A", font).ToSize().Height;
        viditelnostDo = viditelneRadky;
    }
}

private Radek[] radky = new Radek[0];
private int viditelneRadky;
public override void Vykresli(Graphics gr)
{
    //projde radky, ktere obsahuje a postupne je vykresli
    for (int i = viditelnostOd; i < viditelnostDo; i++)
    {
        radky[i].Vykresli(gr);
    }
}

private int viditelnostOd=0, viditelnostDo=0;
//posunovani textu
public void PosunObsah(Smer smerSipky)
{
    switch (smerSipky)
    {
        case Smer.dolu:
            //text se posouva nahoru o 2 radky
            if (viditelnostDo < radky.Length-1)
            {
                viditelnostOd += 2;
                viditelnostDo += 2;
                for (int i = viditelnostDo; i >= viditelnostOd; i--)
                {
                    if (i == radky.Length)
                    {
                        continue;
                    }
                    radky[i].Pozice = radky[i - 2].Pozice;
                }
            }
            break;
        case Smer.nahoru:
            //text se posouva dolu o 2 radky
            if (viditelnostOd >= 2)
```

```
            {
                viditelnostOd -= 2;
                viditelnostDo -= 2;
                for (int i = viditelnostOd; i <= viditelnostDo; i++)
                {
                    if (i+2 == radky.Length)
                    {
                        continue;
                    }
                    radky[i].Pozice = radky[i + 2].Pozice;
                }
            }
            break;
        }
    }

    public enum Smer
    {
        nahoru = 1,
        dolu = 2
    }
    private  class Radek : GrafickyObjekt
    {
        public Radek()
            : base()
        {
            this.fontSize = 14f;
            this.barva = Brushes.White;
        }

        private string text;
        public string Text
        {
            get
            {
                return text;
            }
            set
            {
                text = value;
            }
        }
        private Font font;
        public Font Font
        {
            get
            {
                return font;
            }
        }
        private Brush barva;
        public Brush Barva
        {
            get
            {
                return barva;
            }
            set
            {
                barva = value;
            }
        }
        private float fontSize;
        public float FontSize
        {
            get
            {
                return fontSize;
            }
            set
            {
                fontSize = value;
                this.font = new Font(fontFamily, fontSize);
            }
        }
        private FontFamily fontFamily;
```

```
        public FontFamily FontFamily
        {
            get
            {
                return fontFamily;
            }
            set
            {
                fontFamily = value;
                font = new Font(fontFamily, fontSize);
            }
        }
        public override void Vykresli(Graphics gr)
        {
            gr.DrawString(this.text, font, barva, this.pozice);
        }
    }
  }
}
```

C12ROZLISENI

```
using System;
using System.Windows.Forms;
using System.Drawing;

namespace Labyrint
{
  public class DvanactSetXOsmSet:Rozliseni
  {
    private Graphics gr;
    private Ohraniceni ohrHracTitulek, ohrNalezenoKolik, ohrPocetTahu, ohrProhledaniNalezenych;
    private Text hracTitulek, nalezenoKolik, pocetTahu, prohledaniNalezenych;
    public DvanactSetXOsmSet(Graphics Gr, Text HracTitulek, Text NalezenoKolik, Text PocetTahu, Text ProhledaniNalezenych)
    {
      this.gr = Gr;
      //defaultni hodnoty pro titulky, respektive
      this.ohrHracTitulek = new Ohraniceni(new Point(780, 320), new Point(870, 320));
      this.ohrNalezenoKolik = new Ohraniceni(new Point(835, 370), new Point(895, 370));
      this.ohrPocetTahu = new Ohraniceni(new Point(850, 400), new Point(893, 400));
      this.ohrProhledaniNalezenych = new Ohraniceni(new Point(770, 465), new Point(860, 465));

      this.hracTitulek = HracTitulek;
      this.nalezenoKolik = NalezenoKolik;
      this.pocetTahu = PocetTahu;
      this.prohledaniNalezenych = ProhledaniNalezenych;


      HledanyTitulek = new Point(680, 130);
      velikostKarty = new Size(130, 125);
      poziceKarty = new Point(720, 150);
      poziceKartyNalezene = new Point(720, 480);
      velikostFigurky = new Size(40, 40);
      Minimalizace = new Point(348, 25);
      Zavrit = new Point(378, 25);
      DalsiHrac = new Point(335, 706);
      Cas = new Point(380, 700);
      NalezenTitulek = new Point(680, 605);
      NalezenoTitulek = new Point(760, 370);
      PocetTahuTitulek = new Point(760, 400);

      DalsiHracRozmery = new Size(30, 30);
      ZavritRozmery = new Size(30, 30);
      MinimalizaceRozmery = new Size(30, 30);
      SipkaLevaNickRozmery = new Size(20, 20);
      SipkaLevaNick = new Point(760, 322);
      SipkaPravaNickRozmery = new Size(20, 20);
      SipkaPravaNick = new Point(870, 322);
      SipkaLevaNalezenoRozmery = new Size(20, 20);
      SipkaLevaNalezeno = new Point(750, 465);
      SipkaPravaNalezenoRozmery = new Size(20, 20);
      SipkaPravaNalezeno = new Point(860, 465);
```

```
            Vycentruj();
        }
        public override void Vycentruj()
        {
            //aktualizuje umisteni na zaklade zmeny textu
            //na zaklade delky textu provedu vycentrovani v ohraniceni
            SizeF delkaTextu = gr.MeasureString(hracTitulek.TextovyObsah, hracTitulek.Font);
            int delkaOhraniceni = this.ohrHracTitulek.Koncove.X - this.ohrHracTitulek.Pocatecni.X;
            int odsazeni = (delkaOhraniceni - (int)delkaTextu.Width) / 2;
            hracTitulek.Pozice = new Point(ohrHracTitulek.Pocatecni.X + odsazeni, ohrHracTitulek.Pocatecni.Y);

            //tady bude zarovnani k pravemu okraji
            delkaTextu = gr.MeasureString(nalezenoKolik.TextovyObsah, nalezenoKolik.Font);
            nalezenoKolik.Pozice = new Point(ohrNalezenoKolik.Koncove.X - (int)delkaTextu.Width, ohrNalezenoKolik.Pocatecni.Y);

            delkaTextu = gr.MeasureString(pocetTahu.TextovyObsah, pocetTahu.Font);
            pocetTahu.Pozice = new Point(ohrPocetTahu.Koncove.X - (int)delkaTextu.Width, ohrPocetTahu.Pocatecni.Y);

            delkaTextu = gr.MeasureString(prohledaniNalezenych.TextovyObsah, prohledaniNalezenych.Font);
            delkaOhraniceni = this.ohrProhledaniNalezenych.Koncove.X - this.ohrProhledaniNalezenych.Pocatecni.X;
            odsazeni = (delkaOhraniceni - (int)delkaTextu.Width) / 2;
            prohledaniNalezenych.Pozice = new Point(ohrProhledaniNalezenych.Pocatecni.X + odsazeni,
ohrProhledaniNalezenych.Pocatecni.Y);


        }

    }

    public class TisicDvacetCtyriXSedmsetSedesatOsm : Rozliseni
    {
        private Graphics gr;
        private Ohraniceni ohrHracTitulek, ohrNalezenoKolik, ohrPocetTahu, ohrProhledaniNalezenych;
        private Text hracTitulek, nalezenoKolik, pocetTahu, prohledaniNalezenych;
        public TisicDvacetCtyriXSedmsetSedesatOsm(Graphics Gr, Text HracTitulek, Text NalezenoKolik, Text PocetTahu, Text
ProhledaniNalezenych)
        {
            this.gr = Gr;
            //defaultni hodnoty pro titulky, respektive
            this.ohrHracTitulek = new Ohraniceni(new Point(730, 310), new Point(870, 310));
            this.ohrNalezenoKolik = new Ohraniceni(new Point(780, 370), new Point(863, 370));
            this.ohrPocetTahu = new Ohraniceni(new Point(820, 400), new Point(863, 400));
            this.ohrProhledaniNalezenych = new Ohraniceni(new Point(745, 450), new Point(835, 450));

            this.hracTitulek = HracTitulek;
            this.nalezenoKolik = NalezenoKolik;
            this.pocetTahu = PocetTahu;
            this.prohledaniNalezenych = ProhledaniNalezenych;


            HledanyTitulek = new Point(656, 127);
            velikostKarty = new Size(125, 120);
            poziceKarty = new Point(700, 150);
            poziceKartyNalezene = new Point(700, 480);
            velikostFigurky = new Size(40, 40);
            Minimalizace = new Point(335, 25);
            Zavrit = new Point(365, 25);
            DalsiHrac = new Point(335, 675);
            Cas = new Point(380, 670);
            NalezenTitulek = new Point(650, 585);
            NalezenoTitulek = new Point(730, 370);
            PocetTahuTitulek = new Point(728, 400);

            DalsiHracRozmery = new Size(30, 30);
            ZavritRozmery = new Size(30, 30);
            MinimalizaceRozmery = new Size(30, 30);
            SipkaLevaNickRozmery = new Size(20, 20);
            SipkaLevaNick = new Point(730, 312);
            SipkaPravaNickRozmery = new Size(20, 20);
            SipkaPravaNick = new Point(845, 312);
            SipkaLevaNalezenoRozmery = new Size(20, 20);
            SipkaLevaNalezeno = new Point(720, 450);
            SipkaPravaNalezenoRozmery = new Size(20, 20);
            SipkaPravaNalezeno = new Point(833, 450);
```

```
            Vycentruj();
        }
        public override void Vycentruj()
        {
            //aktualizuje umisteni na zaklade zmeny textu
            //na zaklade delky textu provedu vycentrovani v ohraniceni
            SizeF delkaTextu = gr.MeasureString(hracTitulek.TextovyObsah, hracTitulek.Font);
            int delkaOhraniceni = this.ohrHracTitulek.Koncove.X - this.ohrHracTitulek.Pocatecni.X;
            int odsazeni = (delkaOhraniceni - (int)delkaTextu.Width) / 2;
            hracTitulek.Pozice = new Point(ohrHracTitulek.Pocatecni.X + odsazeni, ohrHracTitulek.Pocatecni.Y);

            //tady bude zarovnani k pravemu okraji
            delkaTextu = gr.MeasureString(nalezenoKolik.TextovyObsah, nalezenoKolik.Font);
            nalezenoKolik.Pozice = new Point(ohrNalezenoKolik.Koncove.X - (int)delkaTextu.Width, ohrNalezenoKolik.Pocatecni.Y);

            delkaTextu = gr.MeasureString(pocetTahu.TextovyObsah, pocetTahu.Font);
            pocetTahu.Pozice = new Point(ohrPocetTahu.Koncove.X - (int)delkaTextu.Width, ohrPocetTahu.Pocatecni.Y);

            delkaTextu = gr.MeasureString(prohledaniNalezenych.TextovyObsah, prohledaniNalezenych.Font);
            delkaOhraniceni = this.ohrProhledaniNalezenych.Koncove.X - this.ohrProhledaniNalezenych.Pocatecni.X;
            odsazeni = (delkaOhraniceni - (int)delkaTextu.Width) / 2;
            prohledaniNalezenych.Pozice = new Point(ohrProhledaniNalezenych.Pocatecni.X + odsazeni,
ohrProhledaniNalezenych.Pocatecni.Y);


        }

    }


    public class MenuDvanactSetXOsmSet : RozliseniMenu
    {
        Graphics gr;
        private Size plocha = new Size(760,760);//(750, 760);//sirku udavam pouze tu viditelnou cast, cely obrazek je sirsi
        Text novaHra, pravidla, autori, konec;
        Ohraniceni blok;
        public MenuDvanactSetXOsmSet(Graphics gr, Text novaHra, Text pravidla, Text autori, Text konec)
        {
            this.gr = gr;
            this.novaHra = novaHra;
            this.pravidla = pravidla;
            this.autori = autori;
            this.konec = konec;
            blok = new Ohraniceni(new Point(0, 310), new Point(0, 570 + konec.Rozmery.Height));

            Hrac1 = new Point(275, 384);
            Hrac2 = new Point(275, 431);
            Hrac3 = new Point(275, 476);
            Hrac4 = new Point(275, 524);
            Cislo1 = new Point(146, 384);
            Cislo2 = new Point(146, 431);
            Cislo3 = new Point(146, 476);
            Cislo4 = new Point(146, 524);
            Zpet = new Point(220, 620);
            ZacitHru = new Point(400, 620);
            Sipka1 = new Point(193, 386);
            Sipka2 = new Point(193, 433);
            Sipka3 = new Point(193, 479);
            Sipka4 = new Point(193, 525);
            Sipka5 = new Point(560, 386);
            Sipka6 = new Point(560, 433);
            Sipka7 = new Point(560, 479);
            Sipka8 = new Point(560, 525);
            Rezim1 = new Point(580, 386);
            Rezim2 = new Point(580, 433);
            Rezim3 = new Point(580, 479);
            Rezim4 = new Point(580, 525);
            NadBarva = new Point(190, 350);
            NadJmeno = new Point(370, 350);
            NadRezim = new Point(555, 350);
            NadNovaHra = new Point(190, 290);
            Pravidla = new Point(100, 200);
            PravidlaSipkaDolu = new Point(555, 650);
            PravidlaSipkaNahoru = new Point(555, 300);
            PravidlaText = new Point(170, 300);
            PravidlaTextHranice = new Ohraniceni(new Point(170, 300), new Point(510, 650));
```

```
            PravidlaSipkaDoluRozmery = new Size(20, 20);
            PravidlaSipkaNahoruRozmery = new Size(20, 20);
            ObrazekMenuHraci = new Bitmap((Image)Zdroj.menu_hraci1280x800);
            ObrazekMenuHraciRozmery = new Size(908, 760);
            poleSouradnic = new Point[4] { new Point(210, 384), new Point(210, 431), new Point(210, 477), new Point(210, 523) };
            rozmeryBarvy = new Size(31, 33);
        }

        public override void Vycentruj()
        {
            int velikost, odsazeniX, odsazeniY;
            int mezeraY;

            mezeraY = ((blok.Koncove.Y - blok.Pocatecni.Y) - (novaHra.Rozmery.Height + pravidla.Rozmery.Height +
autori.Rozmery.Height + konec.Rozmery.Height)) / 3;


            velikost = novaHra.Rozmery.Width;
            odsazeniX = (plocha.Width - velikost) / 2;

            odsazeniY = blok.Pocatecni.Y;

            novaHra.Pozice = new Point(odsazeniX, odsazeniY);

            velikost = pravidla.Rozmery.Width;
            odsazeniX = (plocha.Width - velikost) / 2;
            odsazeniY = odsazeniY + novaHra.Rozmery.Height + mezeraY;

            pravidla.Pozice = new Point(odsazeniX, odsazeniY);

            velikost = autori.Rozmery.Width;
            odsazeniX = (plocha.Width - velikost) / 2;
            odsazeniY = odsazeniY + autori.Rozmery.Height + mezeraY;

            autori.Pozice = new Point(odsazeniX, odsazeniY);

            velikost = konec.Rozmery.Width;
            odsazeniX = (plocha.Width - velikost) / 2;
            odsazeniY = odsazeniY + konec.Rozmery.Height + mezeraY;

            konec.Pozice = new Point(odsazeniX, odsazeniY);
        }
    }

    public class MenuTisicDvacetCtyriXSedmsetSedesatOsm : RozliseniMenu
    {
        Graphics gr;
        private Size plocha = new Size(760, 760);//(750, 760);//sirku udavam pouze tu viditelnou cast, cely obrazek je sirsi
        Text novaHra, pravidla, autori, konec;
        Ohraniceni blok;
        public MenuTisicDvacetCtyriXSedmsetSedesatOsm(Graphics gr, Text novaHra, Text pravidla, Text autori, Text konec)
        {
            this.gr = gr;
            this.novaHra = novaHra;
            this.pravidla = pravidla;
            this.autori = autori;
            this.konec = konec;
            blok = new Ohraniceni(new Point(0, 310), new Point(0, 570 + konec.Rozmery.Height));

            Hrac1 = new Point(260, 369);
            Hrac2 = new Point(260, 414);
            Hrac3 = new Point(260, 458);
            Hrac4 = new Point(260, 502);
            Cislo1 = new Point(140, 369);
            Cislo2 = new Point(140, 414);
            Cislo3 = new Point(140, 458);
            Cislo4 = new Point(140, 502);
            Zpet = new Point(220, 620);
            ZacitHru = new Point(400, 620);
            Sipka1 = new Point(187, 370);
            Sipka2 = new Point(187, 415);
            Sipka3 = new Point(187, 459);
            Sipka4 = new Point(187, 503);
            Sipka5 = new Point(539, 370);
            Sipka6 = new Point(539, 415);
            Sipka7 = new Point(539, 459);
```

```
            Sipka8 = new Point(539, 503);
            Rezim1 = new Point(559, 369);
            Rezim2 = new Point(559, 414);
            Rezim3 = new Point(559, 458);
            Rezim4 = new Point(559, 502);
            NadBarva = new Point(183, 335);
            NadJmeno = new Point(360, 335);
            NadRezim = new Point(538, 335);
            NadNovaHra = new Point(183, 280);
            Pravidla = new Point(100, 200);
            PravidlaSipkaDolu = new Point(555, 620);
            PravidlaSipkaNahoru = new Point(555, 300);
            PravidlaText = new Point(170, 300);
            PravidlaTextHranice = new Ohraniceni(new Point(170, 300), new Point(510, 650));
            PravidlaSipkaDoluRozmery = new Size(20, 20);
            PravidlaSipkaNahoruRozmery = new Size(20, 20);
            ObrazekMenuHraci = new Bitmap((Image)Zdroj.menu_hraci1024x768);
            ObrazekMenuHraciRozmery= new Size(873, 730);
            poleSouradnic = new Point[4] { new Point(203, 369), new Point(203, 414), new Point(203, 458), new Point(203, 502) };
            rozmeryBarvy = new Size(30, 32);
        }

        public override void Vycentruj()
        {
            int velikost, odsazeniX, odsazeniY;
            int mezeraY;

            mezeraY = ((blok.Koncove.Y - blok.Pocatecni.Y) - (novaHra.Rozmery.Height + pravidla.Rozmery.Height +
autori.Rozmery.Height + konec.Rozmery.Height)) / 3;


            velikost = novaHra.Rozmery.Width;
            odsazeniX = (plocha.Width - velikost) / 2;

            odsazeniY = blok.Pocatecni.Y;

            novaHra.Pozice = new Point(odsazeniX, odsazeniY);

            velikost = pravidla.Rozmery.Width;
            odsazeniX = (plocha.Width - velikost) / 2;
            odsazeniY = odsazeniY + novaHra.Rozmery.Height + mezeraY;

            pravidla.Pozice = new Point(odsazeniX, odsazeniY);

            velikost = autori.Rozmery.Width;
            odsazeniX = (plocha.Width - velikost) / 2;
            odsazeniY = odsazeniY + autori.Rozmery.Height + mezeraY;

            autori.Pozice = new Point(odsazeniX, odsazeniY);

            velikost = konec.Rozmery.Width;
            odsazeniX = (plocha.Width - velikost) / 2;
            odsazeniY = odsazeniY + konec.Rozmery.Height + mezeraY;

            konec.Pozice = new Point(odsazeniX, odsazeniY);
        }
    }


    public abstract class RozliseniMenu
    {

        public abstract void Vycentruj();

        public Point Hrac1;
        public Point Hrac2;
        public Point Hrac3;
        public Point Hrac4;
        public Point Cislo1;
        public Point Cislo2;
        public Point Cislo3;
        public Point Cislo4;
        public Point Zpet;
        public Point ZacitHru;
        public Point Sipka1;
        public Point Sipka2;
```

```
    public Point Sipka3;
    public Point Sipka4;
    public Point Sipka5;
    public Point Sipka6;
    public Point Sipka7;
    public Point Sipka8;
    public Point Rezim1;
    public Point Rezim2;
    public Point Rezim3;
    public Point Rezim4;
    public Point NadBarva;
    public Point NadJmeno;
    public Point NadRezim;
    public Point NadNovaHra;
    public Point PravidlaText;
    public Point PravidlaSipkaDolu;
    public Point PravidlaSipkaNahoru;
    public Point Pravidla;
    public Ohraniceni PravidlaTextHranice;
    public Size PravidlaSipkaDoluRozmery;
    public Size PravidlaSipkaNahoruRozmery;
    public Bitmap ObrazekMenuHraci;
    public Size ObrazekMenuHraciRozmery;
    //slouzi jako souradnice pro barvy do menu
    public Point[] poleSouradnic;
    public Size rozmeryBarvy;
}

public abstract class Rozliseni
{
    public abstract void Vycentruj();

    public Point HledanyTitulek;
    public Size velikostKarty;
    public Point poziceKarty;
    public Point poziceKartyNalezene;
    public Size velikostFigurky;
    public Point Minimalizace;
    public Point Zavrit;
    public Point DalsiHrac;
    public Point Cas;
    public Point NalezenTitulek;
    public Point NalezenoTitulek;
    public Point PocetTahuTitulek;

    public Size DalsiHracRozmery;
    public Size ZavritRozmery;
    public Size MinimalizaceRozmery;
    public Size SipkaLevaNickRozmery;
    public Point SipkaLevaNick;
    public Size SipkaPravaNickRozmery;
    public Point SipkaPravaNick;
    public Size SipkaLevaNalezenoRozmery;
    public Point SipkaLevaNalezeno;
    public Size SipkaPravaNalezenoRozmery;
    public Point SipkaPravaNalezeno;
}

public struct Ohraniceni
{
    Point pocatecni;
    Point koncove;
    public Ohraniceni(Point Pocatecni, Point Koncove)
    {
        this.pocatecni = Pocatecni;
        this.koncove = Koncove;
    }
    public Point Pocatecni
    {
        get
        {
            return pocatecni;
        }
    }
    public Point Koncove
    {
```

```
            get
            {
                return koncove;
            }
        }
    }
}
```

C13STARTUP

```
using System;
using System.Windows.Forms;
using System.Threading;
using System.Drawing;

namespace Labyrint
{
    //trida slouzici pro uchovani dat z menu do hry
    static class inter
    {
        private static string[] nicks;
        private static Color[] barvy;
        private static bool[] ai;
        private static int cisloFontu;
        public static int CisloFontu
        {
            get
            {
                return cisloFontu;
            }
            set
            {
                cisloFontu = value;
            }
        }
        public static string[] Nicks
        {
            get
            {
                return nicks;
            }
            set
            {
                nicks = value;
            }
        }
        public static Color[] Barvy
        {
            get
            {
                return barvy;
            }
            set
            {
                barvy = value;
            }
        }
        public static bool[] Ai
        {
            get
            {
                return ai;
            }
            set
            {
                ai = value;
            }
        }
    }


    static class VstupniBodProgramu
    {
```

```
        static ThreadStart zastupce = new ThreadStart(formular);
        public static Thread vlakno = new Thread(zastupce);//nove vlakno programu

        public static void Main()
        {
          Application.EnableVisualStyles();
          //pro zacatek urcim, ze bude zobrazeno uvodi menu, trochu krkolomne
          UvodniMenu menu = new UvodniMenu(TYPmenu.uvodni);
          //nacist samostatny formular, kde bude jako pozadi menu. Pri zacatku hry form vymazat a prenest nastaveni do formu hry
          Application.Run(menu);
        }


        private static void formular()
        {
          Application.Run(new HraciForm());
        }
      }
    }
```

C14ZDROJ

```
    using System;
    using System.Drawing;
    using System.Drawing.Text;

    namespace Labyrint
    {
      public static class Zdroj
      {
        public static Bitmap Diamand = new Bitmap("./Resources/Diamand.png");
        public static Bitmap Helma = new Bitmap("./Resources/Helma.png");
        public static Bitmap Klice = new Bitmap("./Resources/Klíče.png");
        public static Bitmap Plocha1280x800 = new Bitmap("./Resources/plochy/1280_800_hraciplocha.png"); //new
Bitmap("./Resources/Plocha1280x800.png");
        public static Bitmap Kniha = new Bitmap("./Resources/Kniha.png");
        public static Bitmap Koruna = new Bitmap("./Resources/Koruna.png");
        public static Bitmap Lebka = new Bitmap("./Resources/Lebka.png");
        public static Bitmap Mapa = new Bitmap("./Resources/Mapa.png");
        public static Bitmap Meč = new Bitmap("./Resources/Meč.png");
        public static Bitmap Měšec = new Bitmap("./Resources/Měšec.png");
        public static Bitmap Prsten = new Bitmap("./Resources/Prsten.png");
        public static Bitmap Svícen = new Bitmap("./Resources/Svícen.png");
        public static Bitmap Truhla = new Bitmap("./Resources/Truhla.png");
        public static Bitmap Pozice_modra = new Bitmap("./Resources/Position_blue.png");
        public static Bitmap Pozice_zelena = new Bitmap("./Resources/Position_green.png");
        public static Bitmap Pozice_cervena = new Bitmap("./Resources/Position_red.png");
        public static Bitmap Pozice_zluta = new Bitmap("./Resources/Position_yellow.png");
        public static Bitmap Drak = new Bitmap("./Resources/Drak.png");
        public static Bitmap Duch1 = new Bitmap("./Resources/Duch_01.png");
        public static Bitmap Duch2 = new Bitmap("./Resources/Duch_02.png");
        public static Bitmap Chrobak = new Bitmap("./Resources/Chrobák.png");
        public static Bitmap Krysa = new Bitmap("./Resources/Krysa.png");
        public static Bitmap Kubec = new Bitmap("./Resources/Kubec.png");
        public static Bitmap Mlok = new Bitmap("./Resources/Mlok.png");
        public static Bitmap Mura = new Bitmap("./Resources/Můra.png");
        public static Bitmap Netopyr = new Bitmap("./Resources/Netopýr.png");
        public static Bitmap Pavouk = new Bitmap("./Resources/Pavouk.png");
        public static Bitmap Sova = new Bitmap("./Resources/Sova.png");
        public static Bitmap Vila = new Bitmap("./Resources/Víla.png");
        public static Bitmap Rovina = new Bitmap("./Resources/Rovina.png");
        public static Bitmap Zatacka = new Bitmap("./Resources/Zatáčka.png");
        public static Bitmap Postavicka_cervena = new Bitmap("./Resources/Person_red.png");
        public static Bitmap Postavicka_zelena = new Bitmap("./Resources/Person_green.png");
        public static Bitmap Postavicka_modra = new Bitmap("./Resources/Person_blue.png");
        public static Bitmap Postavicka_zluta = new Bitmap("./Resources/Person_yellow.png");
        public static Bitmap Karta_diamand = new Bitmap("./Resources/Artefakty/Diamand.png");
        public static Bitmap Karta_drak = new Bitmap("./Resources/Artefakty/Drak.png");
        public static Bitmap Karta_duch_01 = new Bitmap("./Resources/Artefakty/Duch_01.png");
        public static Bitmap Karta_duch_02 = new Bitmap("./Resources/Artefakty/Duch_02.png");
        public static Bitmap Karta_helma = new Bitmap("./Resources/Artefakty/Helma.png");
        public static Bitmap Karta_chrobak = new Bitmap("./Resources/Artefakty/Chrobák.png");
        public static Bitmap Karta_klice = new Bitmap("./Resources/Artefakty/Klíče.png");
```

```
        public static Bitmap Karta_kniha = new Bitmap("./Resources/Artefakty/Kniha.png");
        public static Bitmap Karta_koruna = new Bitmap("./Resources/Artefakty/Koruna.png");
        public static Bitmap Karta_krysa = new Bitmap("./Resources/Artefakty/Krysa.png");
        public static Bitmap Karta_kubec = new Bitmap("./Resources/Artefakty/Kubec.png");
        public static Bitmap Karta_lebka = new Bitmap("./Resources/Artefakty/Lebka.png");
        public static Bitmap Karta_mapa = new Bitmap("./Resources/Artefakty/Mapa.png");
        public static Bitmap Karta_mec = new Bitmap("./Resources/Artefakty/Meč.png");
        public static Bitmap Karta_mesec = new Bitmap("./Resources/Artefakty/Měšec.png");
        public static Bitmap Karta_mlok = new Bitmap("./Resources/Artefakty/Mlok.png");
        public static Bitmap Karta_mura = new Bitmap("./Resources/Artefakty/Můra.png");
        public static Bitmap Karta_netopyr = new Bitmap("./Resources/Artefakty/Netopýr.png");
        public static Bitmap Karta_pavouk = new Bitmap("./Resources/Artefakty/Pavouk.png");
        public static Bitmap Karta_prsten = new Bitmap("./Resources/Artefakty/Prsten.png");
        public static Bitmap Karta_sova = new Bitmap("./Resources/Artefakty/Sova.png");
        public static Bitmap Karta_truhla = new Bitmap("./Resources/Artefakty/Truhla.png");
        public static Bitmap Karta_vila = new Bitmap("./Resources/Artefakty/Víla.png");
        public static Bitmap Karta_svicen = new Bitmap("./Resources/Artefakty/Svícen.png");
        public static Bitmap Karta_Pozice_blue = new Bitmap("./Resources/Pozice_blue.png");
        public static Bitmap Karta_Pozice_red = new Bitmap("./Resources/Pozice_red.png");
        public static Bitmap Karta_Pozice_green = new Bitmap("./Resources/Pozice_green.png");
        public static Bitmap Karta_Pozice_yellow = new Bitmap("./Resources/Pozice_yellow.png");

        public static Bitmap menu_hraci1280x800 = new Bitmap("./Resources/plochy/1280_800_hracmenu.png");
        public static Bitmap menu1280x800 = new Bitmap("./Resources/plochy/1280_800_hlavnimenu.png");
        public static Bitmap menu1024x768 = new Bitmap("./Resources/plochy/1024_768_hlavnimenu.png");
        public static Bitmap menu_hraci1024x768 = new Bitmap("./Resources/plochy/1024_768_hracmenu.png");
        public static Bitmap Plocha1024x768 = new Bitmap("./Resources/plochy/1024_768_hraciplocha.png");
        public static Bitmap barva_cerverna = new Bitmap("./Resources/Color_red.png");
        public static Bitmap barva_modra = new Bitmap("./Resources/Color_blue.png");
        public static Bitmap barva_zelena = new Bitmap("./Resources/Color_green.png");
        public static Bitmap barva_zluta = new Bitmap("./Resources/Color_yellow.png");
        public static Bitmap barva_cerna = new Bitmap("./Resources/Color_grey.png");
        public static Bitmap otaznik = new Bitmap("./Resources/Otaznik.png");
        //----------tlacitka---------
        public static Bitmap TlacitkoModry = new Bitmap("./Resources/tlacitka/blue_01.png");
        public static Bitmap TlacitkoZeleny = new Bitmap("./Resources/tlacitka/green_01.png");
        public static Bitmap TlacitkoCerveny = new Bitmap("./Resources/tlacitka/red_01.png");
        public static Bitmap TlacitkoZluty = new Bitmap("./Resources/tlacitka/yellow.png");
        public static Bitmap KrizekBily = new Bitmap("./Resources/tlacitka/ix_01.png");
        public static Bitmap KrizekCerveny = new Bitmap("./Resources/tlacitka/ix_02.png");
        public static Bitmap KrizekZeleny = new Bitmap("./Resources/tlacitka/ix_03.png");
        public static Bitmap KlicBily = new Bitmap("./Resources/tlacitka/key_01.png");
        public static Bitmap KlicCerevny = new Bitmap("./Resources/tlacitka/key_02.png");
        public static Bitmap KlicZeleny = new Bitmap("./Resources/tlacitka/key_03.png");
        public static Bitmap LevaSipkaBila = new Bitmap("./Resources/tlacitka/left_01.png");
        public static Bitmap LevaSipkaCervena = new Bitmap("./Resources/tlacitka/left_02.png");
        public static Bitmap LevaSipkaZelena = new Bitmap("./Resources/tlacitka/left_03.png");
        public static Bitmap PravaSipkaBila = new Bitmap("./Resources/tlacitka/right_01.png");
        public static Bitmap PravaSipkaCervena = new Bitmap("./Resources/tlacitka/right_02.png");
        public static Bitmap PravaSipkaZelena = new Bitmap("./Resources/tlacitka/right_03.png");
        public static Bitmap MinimalizaceBila = new Bitmap("./Resources/tlacitka/minimalizace_01.png");
        public static Bitmap MinimalizaceCervena = new Bitmap("./Resources/tlacitka/minimalizace_02.png");
        public static Bitmap MinimalizaceZelena = new Bitmap("./Resources/tlacitka/minimalizace_03.png");
        public static Bitmap PostavickaModraNeaktivni = new Bitmap("./Resources/tlacitka/person_blue_02.png");
        public static Bitmap PostavickaZelenaNeaktivni = new Bitmap("./Resources/tlacitka/person_green_02.png");
        public static Bitmap PostavickaCervenaNeaktivni = new Bitmap("./Resources/tlacitka/person_red_02.png");
        public static Bitmap PostavickaZlutaNeaktivni = new Bitmap("./Resources/tlacitka/person_yellow_02.png");
        public static Bitmap SipkaDoluBila = new Bitmap("./Resources/tlacitka/down_01.png");
        public static Bitmap SipkaDoluZelena = new Bitmap("./Resources/tlacitka/down_03.png");
        public static Bitmap SipkaDoluCervena = new Bitmap("./Resources/tlacitka/down_02.png");
        public static Bitmap SipkaNahoruBila = new Bitmap("./Resources/tlacitka/top_01.png");
        public static Bitmap SipkaNahoruZelena = new Bitmap("./Resources/tlacitka/top_03.png");
        public static Bitmap SipkaNahoruCervena = new Bitmap("./Resources/tlacitka/top_02.png");
    }
}
```

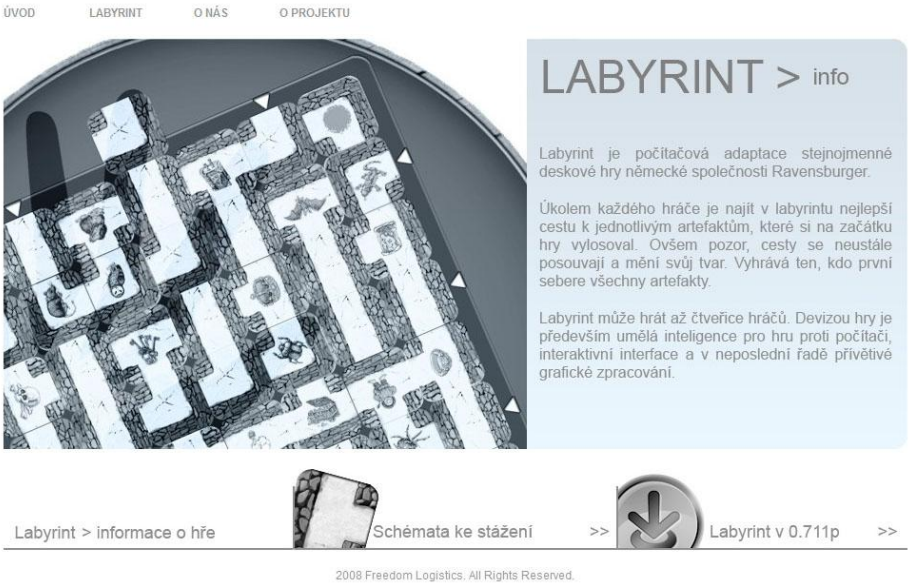
Příloha D WEBOVÉ STRÁNKY

D1 ÚVOD



Obrázek 14 - Úvod

D2 LABYRINT

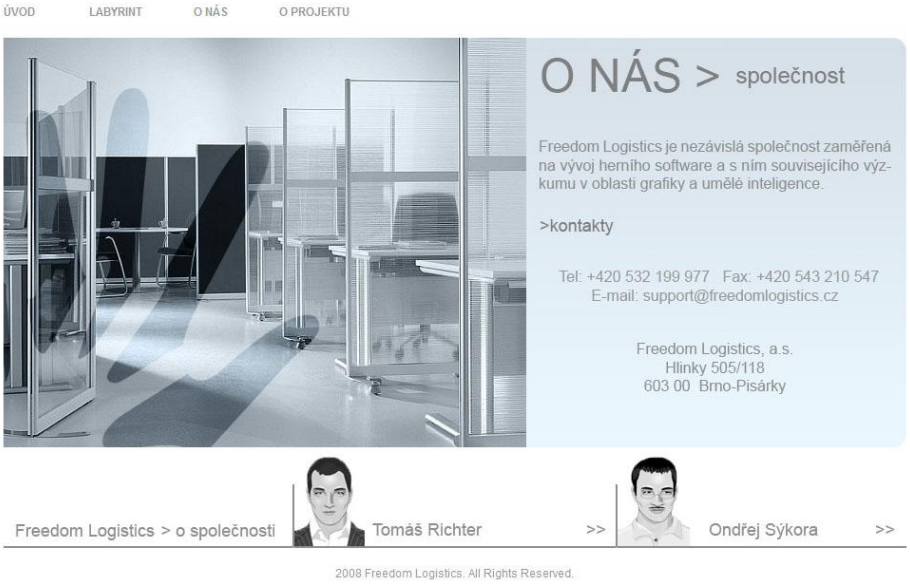


Obrázek 15 – Labyrint

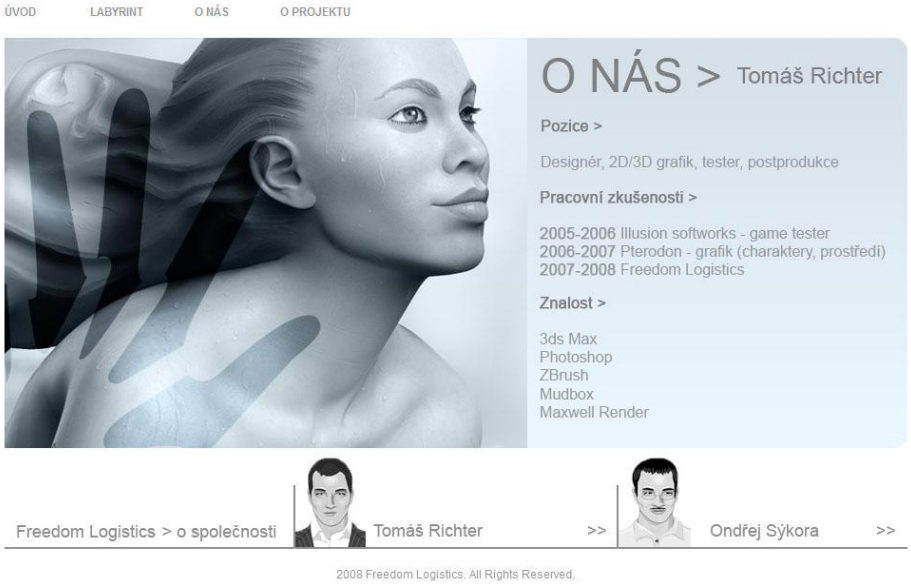


Obrázek 16 - Schémata

D3 O NÁS



Obrázek 17 - O nás



Obrázek 18 - Ondřej Sýkora, Ondřej Sýkora, Tomáš Richter



Obrázek 19 - Ondřej Sýkora

D4 O PROJEKTU



Obrázek 20 - O projektu

Příloha E PREZENTACE

E1 ÚVOD



Obrázek 21 – Úvod

E2 SEZNÁMENÍ S TÉMATEM



Obrázek 22 - Seznámení s tématem

E3 HRA LABYRINT



Obrázek 23 - Hra Labyrint

E4 CHARAKTERISTICKÉ RYSY



Obrázek 24 - Charakteristické rysy

E5 UŽIVATELSKÉ ROZHRANÍ



Obrázek 25 – Uživatelské rozhraní

E6 HRACÍ PLOCHA

Obrázek 26 - Hrací plocha

E7 TECHNOLOGICKÉ OKÉNKO



Obrázek 27 - Technologické okénko

E8 O PROJEKTU

Obrázek 28 - O projektu

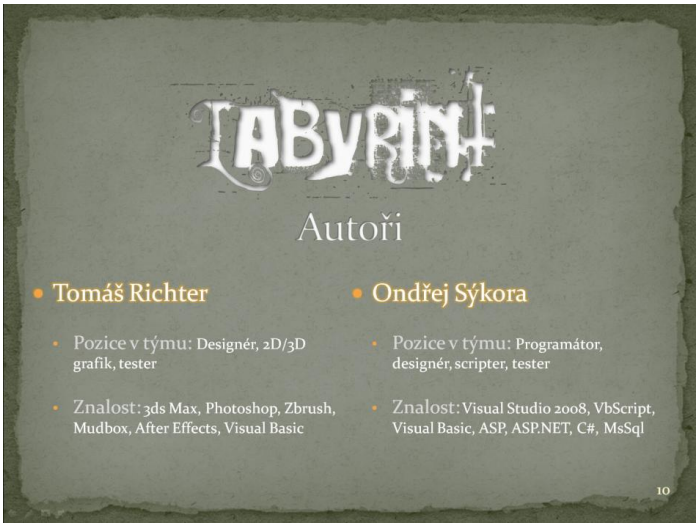


E9 DOPROVODNÉ ELEMENTY

Obrázek 29 - Doprovodné elementy

E10 AUTOŘI

Obrázek 30 - Autoři



Příloha F BOOKLET & POLEP CD

F1 BOOKLET



Obrázek 31 – Booklet

F2 POLEP CD



Obrázek 32 - Polep CD

Středoškolská technika 2008

Setkání a prezentace prací středoškolských studentů na ČVUT

3D MODEL LEGO - BAGR

VOJTĚCH ZAVŘEL

SSPŠ (SMÍCHOVSKÁ STŘEDNÍ PRŮMYSLOVÁ ŠKOLA)
PRESLOVA 25, PRAHA 5 - SMÍCHOV

Obsah:

Obsah: .....
Seznam obrázků .....
Seznam Příloh .....
1 Úvod, anotace úlohy .....
2 Technologie vytváření součástek .....
2.1 Vysunutí .....
2.1.1 Příklad vysunutí .....
2.2 .....
2.3 Rotace .....
2.3.1 Příklad rotace .....
2.4 Pole .....
2.4.1 Příklad pole .....
2.5 Zrcadlení .....
2.5.1 Příklad zrcadlení .....
2.6 Jiné .....
2.6.1 Příklad použití spirály .....
2.6.2 Příklad použití tažení .....
3 Technologie vytváření sestavy .....
3.1 Vazba proti sobě .....
3.2 Úhlová vazba .....
3.3 Vazba vložit .....
3.4 Kombinace vazeb .....
3.5 Pohybové vazby .....
4 Použité programy .....
4.1 Autodesk Inventor .....
4.2 3DStudio Max .....
4.3 Adobe Photoshop .....
4.4 Macromedia Flash .....
5 Scénář WWW stránek .....
5.1 Popis jednotlivých částí WWW stránek .....
5.1.1 Tvorba VRML modelu + HTML stránky modelu .....
5.1.2 Grafický návrh .....
5.1.3 Tvorba menu .....
5.1.4 Tvorba struktury a oddělení .....
5.1.5 Tvorba galerie .....
5.1.6 Kontrola animace + publikace .....

6 Závěr, návrh možných řešení úlohy .....

Seznam obrázků

obr. 1. dialogové okno vysunutí.....
obr. 2. Náčrt hrabla .....
obr. 3. četnost funkce vytažení .....
obr. 4. Dialogové okno rotace .....
obr. 5. Rotace pneumatiky.....
obr. 6. Obdélníkové pole .....
obr. 7. Kruhové pole .....
obr. 8. zrcadlení .....
obr. 9. Použití spirály .....
obr. 10. Použití tažení .....
obr. 11. Dialogové okno vazeb .....
obr. 12. Vazba proti sobě .....
obr. 13. Vazba vložit.....
obr. 14. Kombinace vazeb .....
obr. 15. Dialogové okno pohybových vazeb .....
obr. 16. Pohybová vazba .....
obr. 17. Výkres .....
obr. 18. Logo inventoru (z images.google.com).....
obr. 19. Logo 3DSMax 2008 (z images.google.com).....
obr. 20. Nové logo Adobe Photoshop (z images.google.com).....
obr. 21. Logo nové verze Adobe Flash (z images.google.com).....
obr. 22. VRML model po vygenerování .....
obr. 23. Render č.1 - čistý .....
obr. 24. Render č.2 – drátový model.....
obr. 25. Složení obou renderů v celek .....
obr. 26. Přidělení odkazu tlačítku v menu .....
obr. 27. Rozložení časové osy .....
obr. 28. Vrstva galerie + vrstva mlhy – časová osa.....
obr. 29. Prohození bitmapy.....
obr. 30. Debug filmu.....
obr. 31. Možnost složení vytvořených součástek (scan z letáku).....
obr. 32. Náš model v propagačním letáku (scan z letáku).....

Seznam Příloh

Příloha A Letáky .....
A1 Česky.....
A2 Anglicky .....
Příloha B Seznam součástek.....

1 Úvod, anotace úlohy

Model byl dříve součástí kolekce dánské firmy LEGO, konkrétně serie Technic. Jako takový má 60 různých součástek, přičemž některé se v modelu objevují víckrát. Nejdříve si musíme součástky oměřit a pak převést v měřítku do Inventoru. Tak postupně vzniknou všechny

součástky a sestavy, až nakonec bude bagr kompletní. Následně vytvoříme vazby tak, že bagr bude reagovat na pohyb určité komponenty pohybem jiné komponenty. Např. zatáčení zadních kol, zdviháním hrabla atp.
Až bude toto hotovo, vytvoříme prezentaci v programu Adobe Flash, kterou následně vložíme do webové stránky. K té bude připojen 3D VRML model a video, na kterém jsou demonstrovány všechny funkce bagru.

2 Technologie vytváření součástek

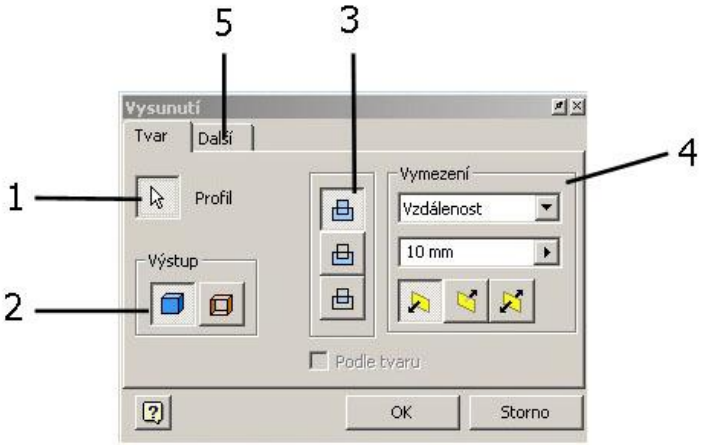
Při výstavbě součástek jsem potřebné rozměry získával pomocí digitálního posuvného měřítka od firmy Powerfix. Kvůli velkému počtu součástek není možno popisovat každou součástku zvlášť.Rozdělíme si tedy součástky dle způsobu tvorby do tří kategorií a na pár příkladech demonstrujeme vytvoření zbylých součástí.

1.kategorie vysunutím
2.kategorie rotací
3.kategorie pomocí polí
4.kategorie zrcadlením
5.kategorie jiné

Program Autodesk Inventor umožňuje různé způsoby tvorby komponentů.My se však budeme zabývat pouze těmi které jsme použili ve stavění modelu Bagru.

2.1 Vysunutí

Pomocí příkazu vysunutí jsem modeloval velkou část komponentů.Aby jste mohli použít funkce vysunutí je potřeba mít alespoň jeden viditelný neadaptivní náčrt.Jestli že máme tento náčrt,přes *okno nástrojů* nebo přes klávesovou zkratku *E* vyvoláme dialogové okno funkce.



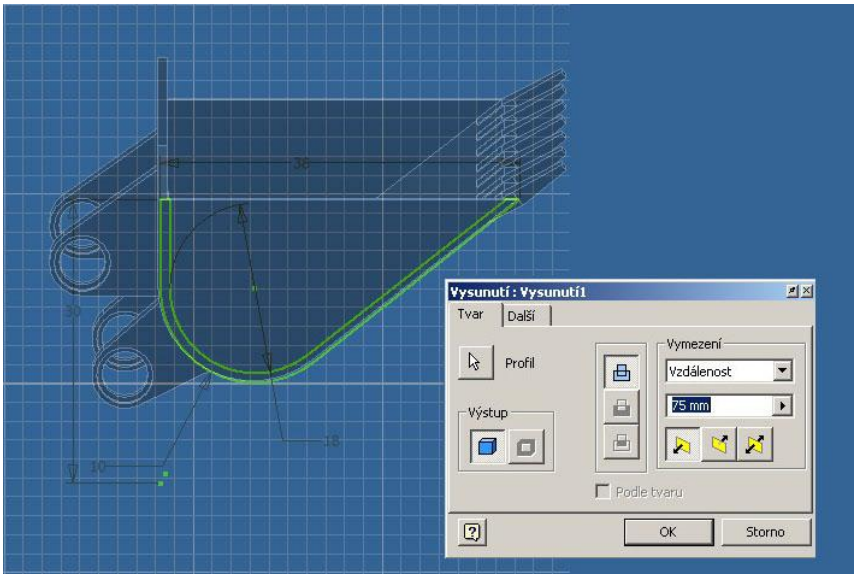
obr. 1. dialogové okno vysunutí

V dialogovém okně nastavíme *profil (1)* výběrem náčrtu.Dále si v nabídce *výstup (2).*vybereme jestli chceme vytvořit plochu nebo těleso.

Musíme si také zvolit, jestli chceme vysunutím náčrtu *připojit,odříznout nebo udělat rozdíl(3)* k modelované součásti. V nabídce *vymezení (4)* můžeme určit vysunutí *vzdáleností, od do* nějaké plochy, nebo u odříznutí skrz *vše.* Také zde zvolíme jakým směrem se má náčrt vysunout.V záložce *další(5)* je například možnost zúžení.

2.1.1 Příklad vysunutí

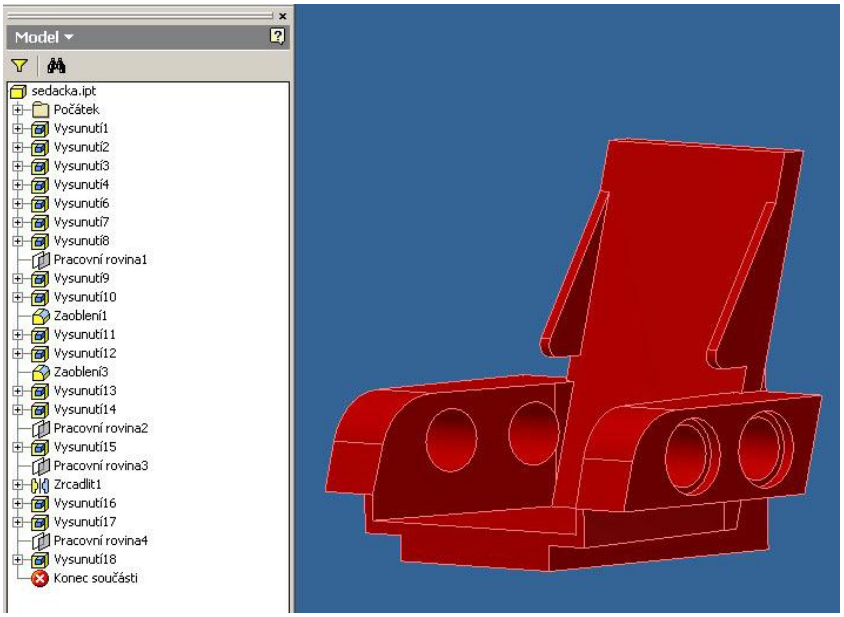
Jako první příklad jsem vybral jednu z nejdůležitějších součástí našeho modelu.Komponent který bagr dělá bagrem tj.hráblo. Zde se jako u většiny dílů hojně používá funkce vysunutí.



obr. 2. Náčrt hrabla

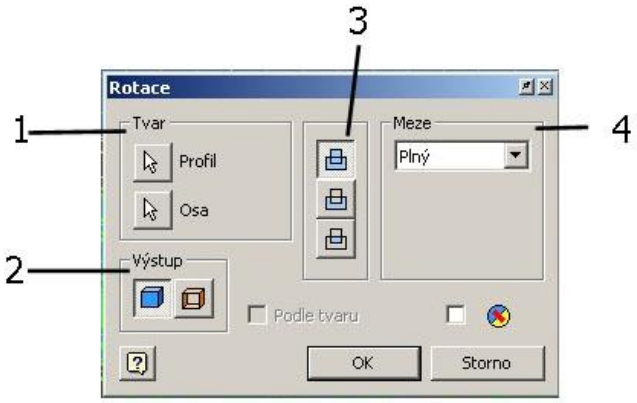
Na obrázku snadno vidíme náčrt, který byl použit jako základ celého hrábla. Postupným přidáváním dalších náčrtů vymodelujeme pomocí *vysunutí* celé hráblo

V druhém příkladu bych rád ukázal četnost s jakou jsem používal funkci vysunutí.Například k vymodelování sedátka jsem použil tuto funkci 18-krát.*Vysunutí* se dá kombinovat i s dalšími modelovacími prvky jako třeba *pole*, nebo *zrcadlení.*



***obr. 3. četnost funkce vytažení* Rotace**

Většinu rotačních součástí jsem dělal právě pomocí funkce rotace.K této funkci potřebujeme stejně jako u vytažení jeden viditelný neadaptivní náčrt, který navíc obsahuje osu.Dialogové okno rotace vyvoláme přes *okno nástrojů* nebo přes klávesovou zkratku *R.*



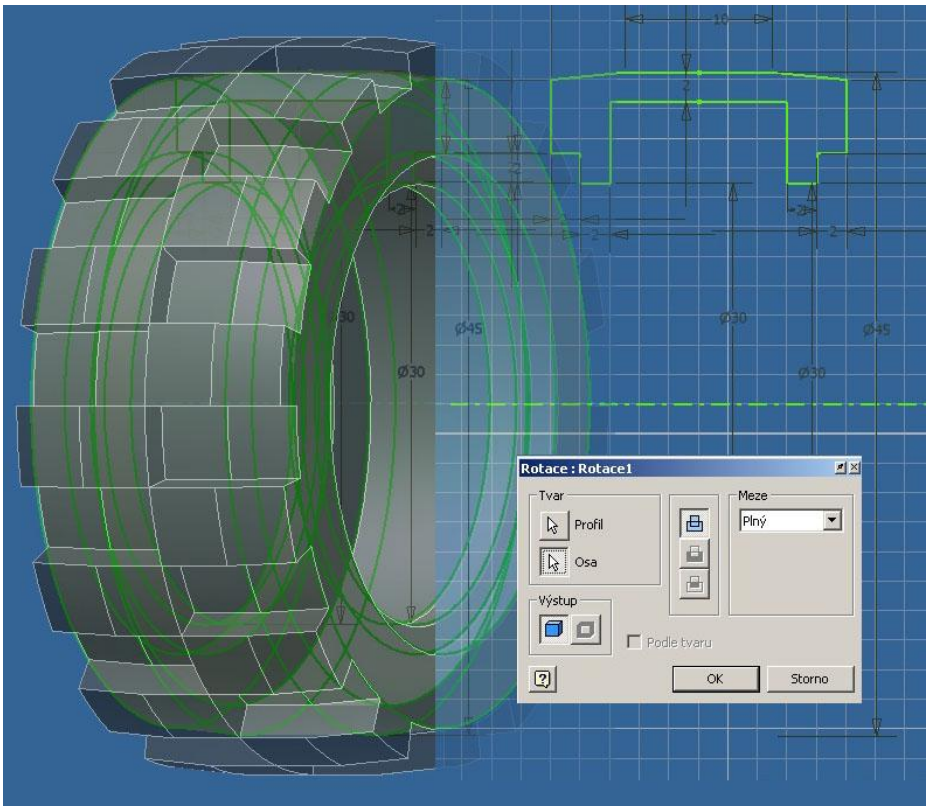
obr. 4. Dialogové okno rotace

Nabídce *Tvar (1)* připojíme profil (náčrt) a určíme osu rotace.Opět můžeme v nabídce *výstup (2)* nastavit jestli budeme tvořit těleso nebo plochu. Stejně tak rozhodneme jestli chceme vysunutím náčrtu *připojit,odříznout nebo udělat rozdíl(3)* k modelované součásti.Jediná

rozdílná nabídka je nabídka *meze (4)*, kde nastavujeme jestli budeme rotovat plný úhel 360°nebo jiný námi zvolený úhel.

2.2.1 Příklad rotace

Na obrázku si ukážeme příklad rotace pneumatiky.Nejprve jsem si udělal osu, následovně jsem nakreslil a okótoval náčrt a nechal orotovat plný úhel.Vzorek pneumatiky je tvořen kruhovým polem malých dílků. Jeden dílek vzorku je také orotován s tím rozdílem že jsem zvolil přesný úhel rotace.



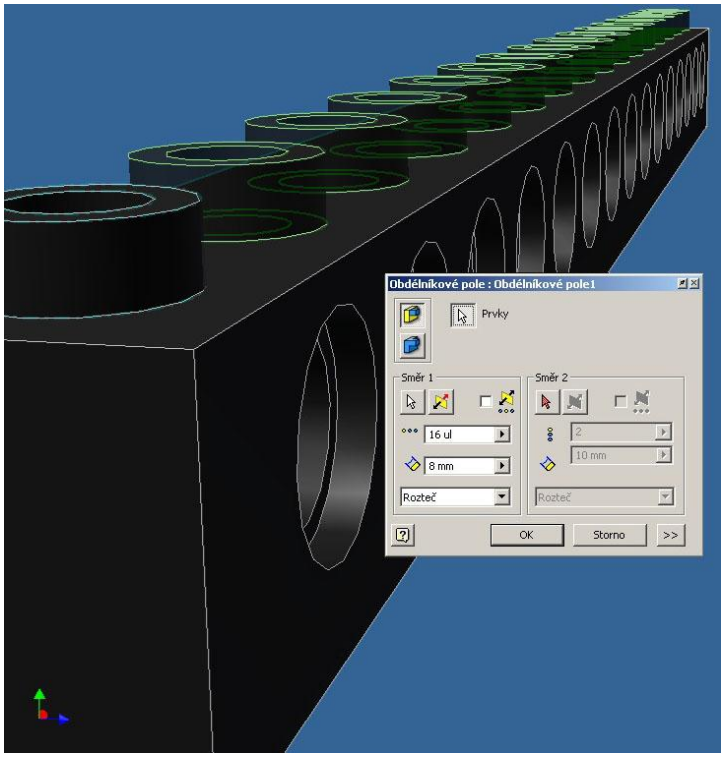
obr. 5. Rotace pneumatiky

2.3 Pole

Nástroj pole nám slouží ke kopírování opakujících se prvků na součásti.Existují dva druhy polí. První druh se nazývá *obdélníkové pole* a druhý se nazývá *kruhové pole*. Jak z názvů vyplývá, v prvním případě budeme kopírovat prvky do obdélníku. Po výběru prvku nastavujeme směr 1 případně i směr 2, pokud chceme mít pole víceřadé.Dále je potřeba určit počet kopírovaných prvků a rozteč mezi nimi. U kruhového pole postupujeme obdobně. Nejprve vybereme prvky a osu rotace. Pak nastavíme počet kopií a úhel v jakém rozmezí chceme pole mít.

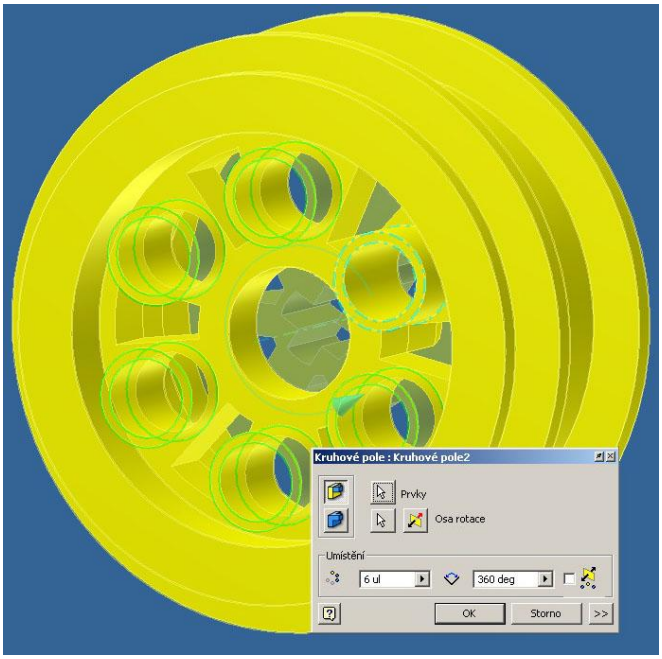
2.3.1 Příklad pole

Za příklad obdélníkového pole jsem vybral součástku známou asi každému, kdo se někdy setkal s legem technic. Je to klasický dírkovaný lego trámek.Základ trámku je tvořen vytažením. Zbytek trámu jsem vymodeloval právě pomocí obdélníkového pole.



obr. 6. Obdélníkové pole

Příklad kruhového pole jsem zvolil součást ráfku, kde je kruhového pole využito k vybrání středu ráfku.

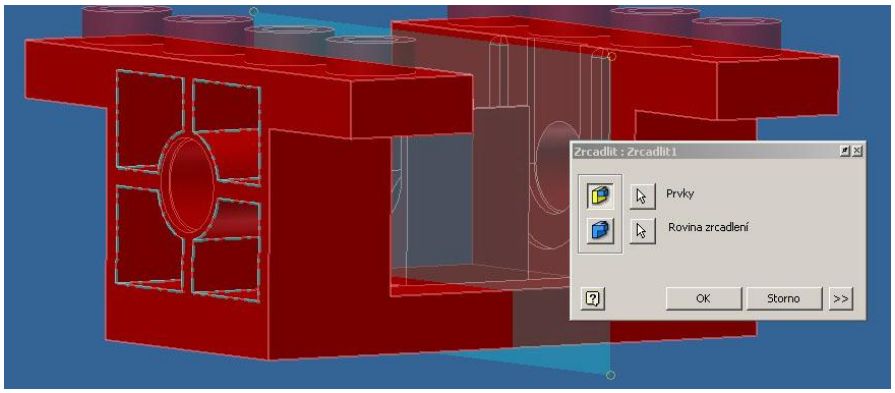


obr. 7. Kruhové pole

2.4 Zrcadlení

Pokud máme některou součástku symetrickou můžeme použít funkci *zrcadelní*.K použití potřebujeme pouze jednu *pracovní rovinu* ,kterou nazýváme rovinou zrcadlení.Po spuštění dialogového okna jednoduše vybereme prvky a rovinu zrcadlení.

2.4.1 Příklad zrcadlení



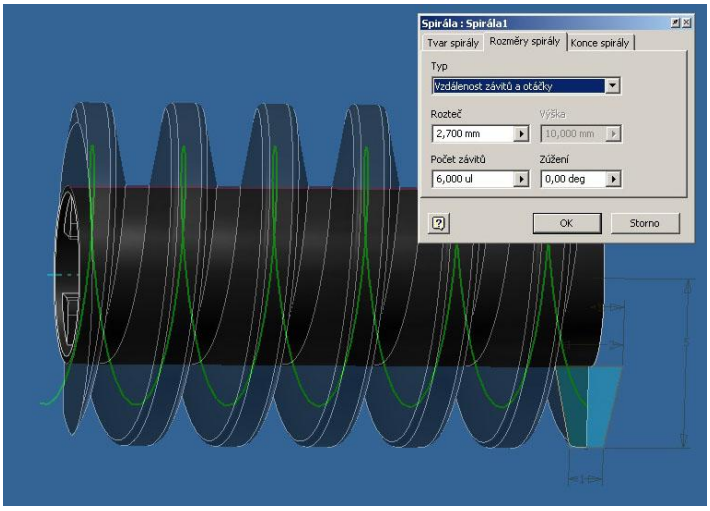
obr. 8. zrcadlení

2.5 Jiné

Jak už jsem výše napsal způsobů tvorby je mnoho. Ani na našem bagru nejsou zdaleka použity všechny.Kategorii jiné bych se teda rád zabýval zbylými způsoby tvorby o kterých jsem se ještě nezmínil. Jsou dva a to je funkce Spirála a funkce Tažení.Nejlépe si to vysvětlíme přímo na příkladech.

2.5.1 Příklad použití spirály

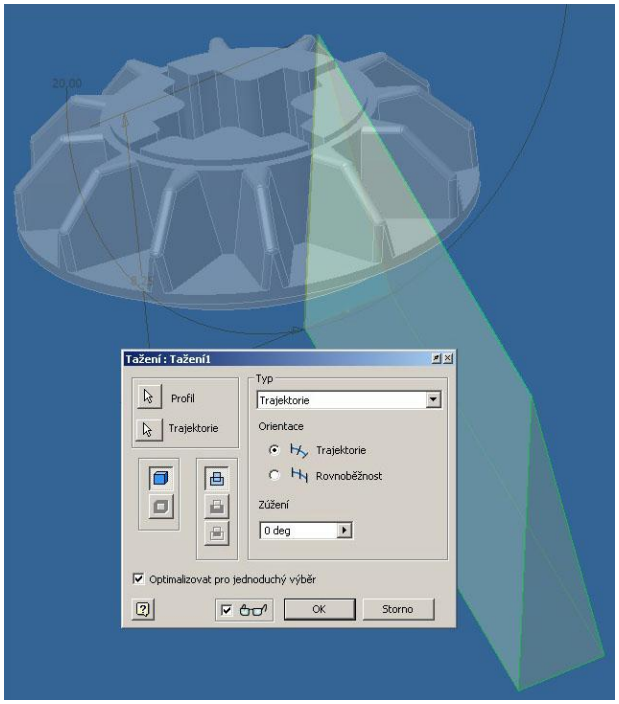
Funkce spirála byla použita pro tvorbu šneku ve šnekovém převodu.Po otevření dialogového okna spirály, stejně jako u rotace, přiřadíme profil(náčrt) a osu.Od rotace nám v základním nastavení přibyla pouze nabídka směru rotace.V dalších záložkách musíme přesněji specifikovat rozměry spirály jako například rozteč,počet závitu, typ a další.



obr. 9. Použití spirály

2.5.2 Příklad použití tažení

Tažení jsem použil u axiálně ozubeného kola,které je součástí mechanismu otáčení kol.K tažení potřebujeme opět námi známí profil a k němu navíc trajektorii, kterou se nám má vybraný profil táhnout.Tento složitější způsob tvorby jsem použil proto, protože ozubení se nám zužuje z více stran.Finálního tvaru zubu jsem docílil pomocí odebrání vysunutím a zaoblením hran.



obr. 10. Použití tažení

Na těchto pár příkladech jsem chtěl vysvětlit základní technologie modelování součástek.Zbylé součástky se dělali obdobným způsobem jako v uvedených příkladech.

3 Technologie vytváření sestavy

Po vytvoření všech součástek přichází jejich sestavení ve výsledný model.Při spojování sestavy jsem postupoval pomocí přiloženého Lego plánku a stejně jako v plánku jsem dělil sestavu do kroků.Některé součástky (například ráfek a pneumatiku) jsem si připravil do tzv. podsestav (kolo), kterou jsem připojil k finální sestavě.Tyto podsestavy se při připojování k sestavě chovají jako jedna součást.K spojování součástí je použito vazeb.Inventor těchto vazeb nabízí hned několik.Pojďme se na ně podívat z blízka.



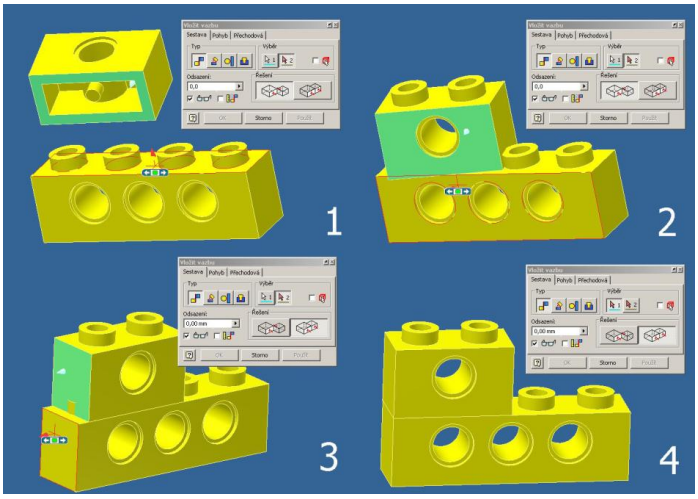
obr. 11. Dialogové okno vazeb

V dialogovém okně vazeb určujeme *typ vazby (1)*. To znamená jestli chceme vazbu *proti sobě*, vazbu tvořenou *úhlem*, *tečnou vazbu* nebo součásti do sebe *vložit*.Podle toho jakou vazbu zvolíme se nám mění i způsob *řešení(3)*.*Výběrem(4)* vybíráme plochu nebo hranu kterou se mají vázat vybrané součásti. Ve volbě *odsazení(2)* zadáváme vzdálenost mezi vázanými komponenty.V záložce *Pohyb* se nám ukrývají pohybové vazby, na které se budeme soustředit později.

Teď si zde popíšeme jednotlivé vazby a ukážeme si kde byly v modelu použity.

3.1 Vazba proti sobě

Tato vazba se nejčastěji používala k spojení dvou lego kostek k sobě. Na obrázku je vidět že je zde použita hned několikrát.Dále jsou na obrázku vidět plochy, které jsou k sobě vázány



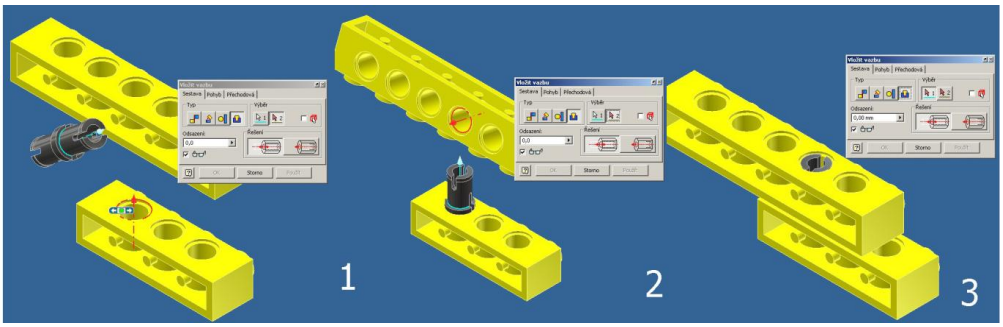
obr. 12. Vazba proti sobě

3.2 Úhlová vazba

Úhlová vazba je v modelu použita jen zřídka.Většinou je užita tam kde jsem potřeboval vazbou řídit nějaký pohyb.Přesto že je tato vazba použita málo má veliký význam. Díky možnosti nahrát řízenou vazbu, je vytvořeno celé prezentační video.

3.3 Vazba vložit

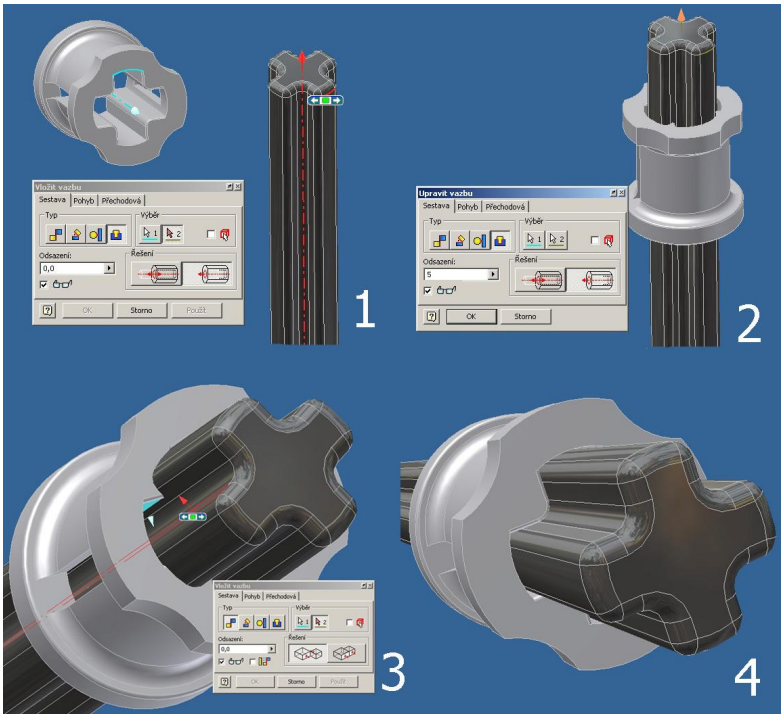
Vazba vložit se nejlépe hodila k čepovým spojům které se na bagru také hojně vyskytují.K vložení čepu se tato vazba skvěle hodí.



obr. 13. Vazba vložit

3.4 Kombinace vazeb

Vazby se mezi sebou samozřejmě mohou kombinovat.Musíme ale dbát na to aby se nám vazby mezi sebou nekřížily nebo si neodporovaly.V takovém případě nás inventor upozorní že takovou vazbu nemůže vytvořit.Další často používaná vazba ve stavění bagru,nasazení lego matky na lať, nám kombinování vazeb objasní více.



obr. 14. Kombinace vazeb

Jak vidíme na obrázku v kroku 1 a 2 je použita vazba vložit s odsazením 5 mm. V kroku 3 je užito vazby proti sobě. Ve čtvrtém kroku je vidět konečný výsledek.

Jak jste si možná všimli, vynechali jsme vazbu tečně. Tuto vazbu jsme vynechali úmyslně, protože jsme na našem modelu nenašli její uplatnění.

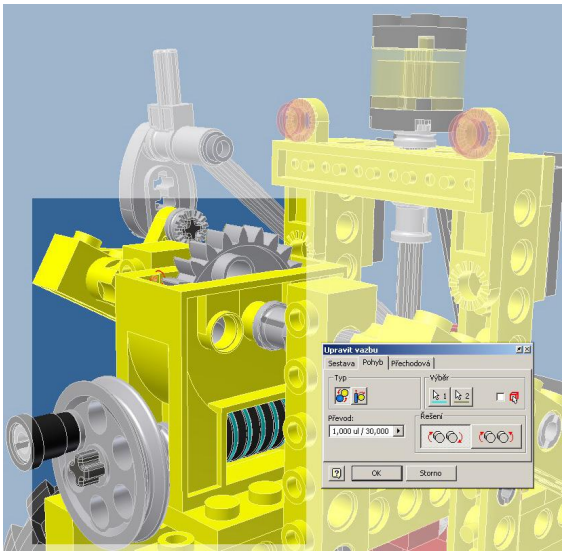
3.5 Pohybové vazby

Jak sem již avizoval budeme se zabývat i pohybovými vazbami. Tyto vazby se dělali až na konec jako taková „perlička". Aby se vše pohybovalo jako ve skutečnosti,musel jsem propojit ozubená kola mechanismu právě pohybovou vazbou.Tyto vazby jsou použity jak u zatáčení bagru, tak u zvedání lžíce pomocí šnekového soukolí.



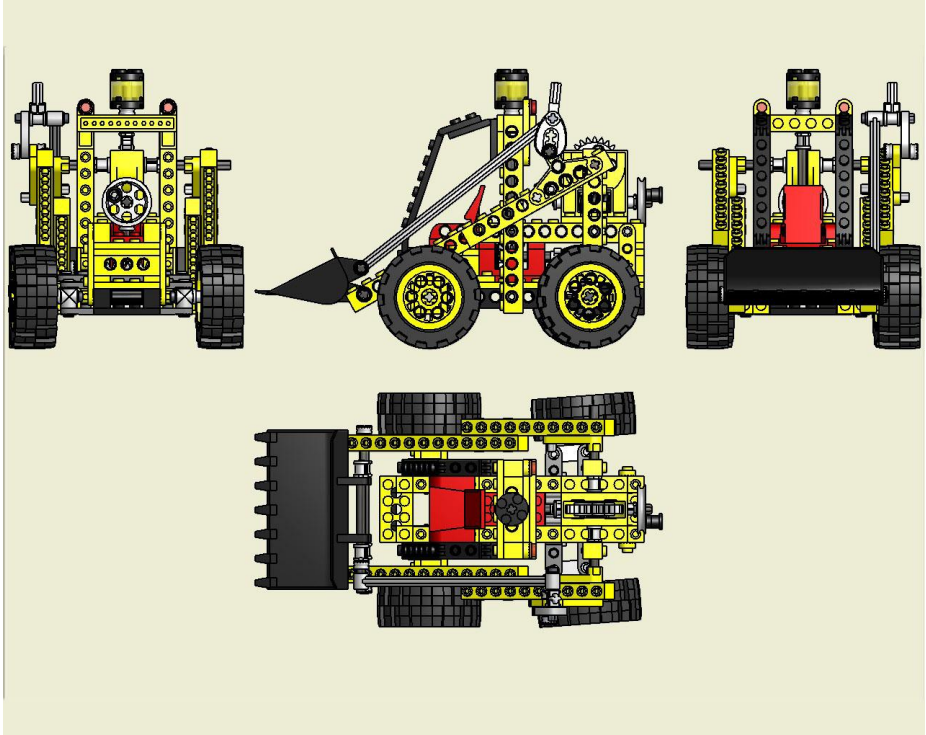
obr. 15. Dialogové okno pohybových vazeb

V dialogovém okně můžeme jako první nastavit *typ(1)* a to jestli chceme pouze rotovat obě součásti nebo rotací posunovat nějaký komponent.*Výběr(3)* funguje stejně jako u předchozích vazeb. Nastavením hodnoty *převod(2)* nastavujeme jaký převodní poměr bude mezi dvěma vybranými komponenty.*Řešení(4)* nám pouze určuje v jakém směru na sebe budou součástky reagovat.



obr. 16. Pohybová vazba

Závěrem, vazby jsem se snažil tvořit tak, aby co nejvíce odpovídaly vazbám, které jsou ve skutečnosti.Jelikož inventor nemá žádnou možnost kontrolovat jestli je vazba logicky v pořádku, bylo to občas dost obtížné. Nakonec se však vše podařilo spojit tak jak by mělo a bagr má stejné možnosti jako ve skutečnosti.



obr. 17. Výkres

4 Použité programy

Následující texty jsou z části www.xanadu.cz a z www.cs.wikipedia.org .

4.1 Autodesk Inventor

Autodesk Inventor je světově nejprodávanější CAD aplikace pro strojírenskou 3D konstrukci - výkonná aplikace od Autodesku, výrobce nejrozšířenější CAD aplikace - AutoCADu. Ve světě bylo dosud (1/2008) instalováno přes 775.000 licencí Inventoru (každých 5 minut přejde někdo na Inventor). Firma XANADU je autorem českých lokalizací Inventoru a dodavatelem Inventoru i podnikových řešení postavených na této CAD aplikaci.



obr. 18. Logo inventoru (z images.google.com)

Autodesk Inventor vám umožní:

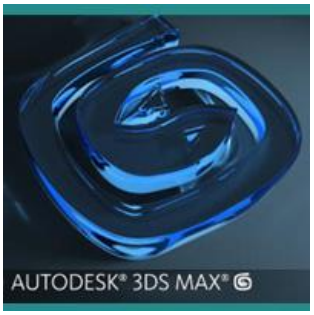
- být produktivní již od prvního dne jeho použití
- pracovat s obzvláště vysokým výkonem při rozsáhlých sestavách
- vytvářet náčrty s prvotním řešením funkčnosti a následným řešením tvaru
- používat existující DWG výkresy AutoCADu
- intuitivně vytvářet a modifikovat konstrukční sestavy
- zachytit, sdílet a opakovaně využívat konstrukční znalosti
- pracovat paralelně v širším týmu konstruktérů
- používat konstrukční údaje o výrobku kdykoliv a kdekoliv
- pracovat s rozhraním, které sleduje vaše pracovní postupy
- vytvářet 2D výkresovou dokumentaci rychleji a přesněji než pomocí 2D CAD nástrojů

4.2 3DStudio Max

Autodesk 3DS MAX pochází z programu 3D Studio vyvíjeném Yost Group pod záštitou Autodesku, který ke konci 90. let dosáhl verze „Release 4". 3DS MAX je jeho značně vylepšenou verzí, přizpůsobenou pro multitaskingové operační systémy, využívající všech jejich výhod. 3DS MAX je tedy určen pro operační systémy Microsoft Windows, 32bitové i nativní 64bitové verze.

3ds max obsahuje několik technologií pro rendering, včetně radiosity a global illumination, mj. i stínovač *Mental Ray*. Vestavěn je skriptovací jazyk *MaxScript*. Další funkce lze přidávat pomocí API rozhraní a na něm postavených plugin modulů.

3ds max podporuje řadu souborových formátů - kromě nativních souborů .MAX jsou to i CAD formáty .DWG a DXF. Vestavěný modul *Character Studio* umožňuje snadnou tvorbu pohybů postav a tvorů, využívá inverzní kinematiky. Vestavěný modul *Reactor* je nástroj pro fyzikální reakce těles - simuluje reakce pevných a měkkých těles, textilií, gravitační síly, atd.



obr. 19. Logo 3DSMax 2008 (z images.google.com)

4.3 Adobe Photoshop

Adobe Photoshop je bitmapový grafický editor pro tvorbu a úpravy bitmapové grafiky (např. fotografií) vytvořený firmou Adobe Systems. První verze (1.0) vyšla v únoru roku 1990 pro Mac OS a v současné době (k 15. září 2007) je k dispozici desátá verze, prodávaná pod označením *Creative Suite 3* (CS3), která byla vydána v dubnu 2007. V ten samý měsíc vyšla i rozšířená verze Extended, nabízející všechny funkce jako verze standardní, avšak s novými nástroji pro oblast vědy, 3D a videa. Původně je Photoshop dílem bratrů Thomase a Johna Knolla, kteří na vývoji začali pracovat již v roce 1987. Jedním z nejvýznamnějších bodů byl vznik verze pro operační systém Microsoft Windows v roce 1996 (verze 4.0). Označení Creative Suite používané u nových verzí vyjadřuje fakt, že je Photoshop integrován se skupinou dalších grafických programů firmy Adobe (Adobe's Creative Suite), kam patří mimo jiné Adobe Illustrator či Adobe InDesign.



obr. 20. Nové logo Adobe Photoshop (z images.google.com)

4.4 Macromedia Flash

Flash je grafický vektorový program, momentálně ve vlastnictví společnosti Adobe (dříve Macromedia). Používá se především pro tvorbu (převážně internetových) interaktivních animací, prezentací a her. Rozšíření Flashe na internetu pomohla malá velikost výsledných souborů, protože se uchovávají ve vektorovém formátu, a proto ve většině případů vytlačily flashové bannery ty klasické, dříve používané ve formátu GIF.

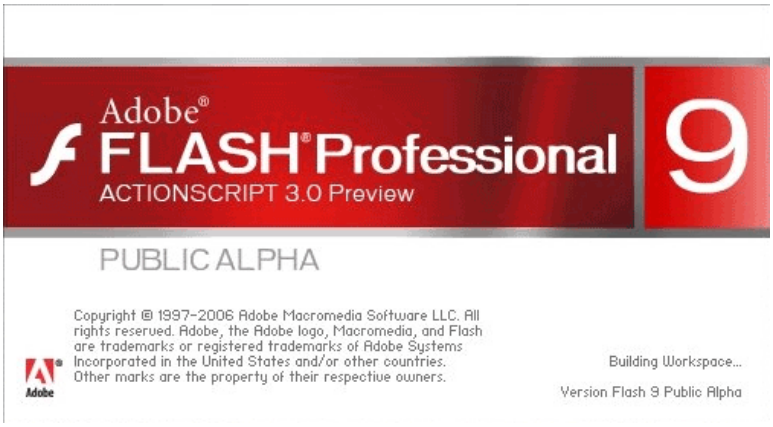
Flash má také vlastní implementovaný programovací jazyk ActionScript, který slouží k rozvinutí všech možností interaktivní animace a vývoji robustních aplikací, v aktuálních verzích je ActionScript poměrně vyspělý objektově orientovaný programovací jazyk. Poslední oficiální verzí jazyku ActionScript je verze **3.0**.

Flash Preview umožní publikovat v nově uvolněném Flash Player 9 a využít také AS 3.0. Flash Professional 9 ActionScript 3.0 Preview je jen plánované uvolnění „pre-release" před vydáním finální verze Flash 9 v roce 2007. Majitelé Flash 8 mohou aplikaci stáhnout z [1]

Flash exportuje soubory do dvou základních formátů:

.swf – v tomto formátu má soubor malou velikost, může být přehrávaný ve webovém prohlížeči, ale k jeho běhu je nutný přehrávač – Adobe Flash Player, který je volně ke stažení na stránkách společnosti Adobe

.exe – formát určený pro spouštění ve Windows (není nutný další přehrávač), soubor má větší velikost. Říká se mu také projektor, jelikož k má tento soubor v sobě implementovaný FlashPlayer.



obr. 21. Logo nové verze Adobe Flash (z images.google.com)

Vedle flashových banerů je největší použití flashe u her a u animací. Flash hry a animace jsou velmi populární mezi návštěvníky internetu. Výhodou takových her je malá velikost souboru a možnost hraní přímo v okně prohlížeče, bez nutnosti stahování. Existuje nesčíselné množství her a jejich variant. V mnoha případech jsou flashové hry používány jako forma reklamy výrobku, firmy nebo například filmu.

5 Scénář WWW stránek

Používat strukturované nadpisy – různé úrovně. Funguje jako manuál. 5-10 stránek. Včetně obrázků… Obrázky číslované a popsané.

Stránky jsou udělané jako čistě prezentační. Nejsou použity žádné databáze ani redakční systémy. Pro jejich tvorbu byl z velké části použit Macromedia Flash 8.0 a při tvorbě interaktivního VRML modelu pak 3DS Max 2008. Pro grafické návrhy Adobe Photoshop 7CS.

5.1 Popis jednotlivých částí WWW stránek

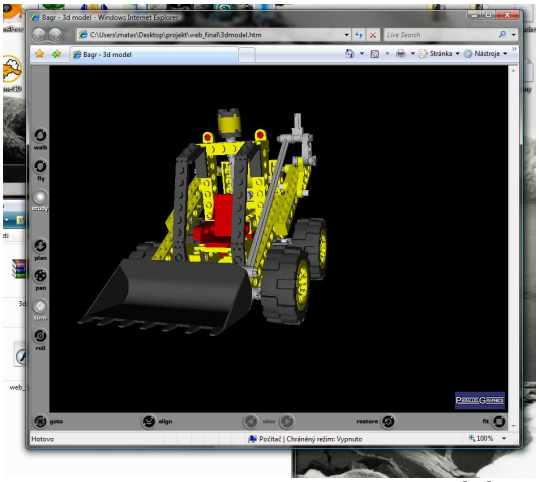
Před začátkem jsem si na papír rozvrhl strukturu stránky dle jednotlivých políček tak, aby člověk neztratil přehled, až poté jsem začal tvořit na PC.

5.1.1 Tvorba VRML modelu + HTML stránky modelu

Model VRML byl vytvořen pomocí převodu původních souborů z Inventoru do prostředí 3DS Max. Tam jsem použil možnost přímého exportu modelu do VRML. Následné implementování do stránky a její formátování podle potřeb modelu (tj. těsné obklopení) je definováno zčásti Java skriptem, zčásti standardně v kódu.
Použitý JavaScript upravoval velikost okna prohlížeče po spustění stránky.

```
<script language="JavaScript">
<!—
window.resizeTo(h,w)
-->
</script>
```

V kódu pak platí, že hodnota h je šířka a w výška.

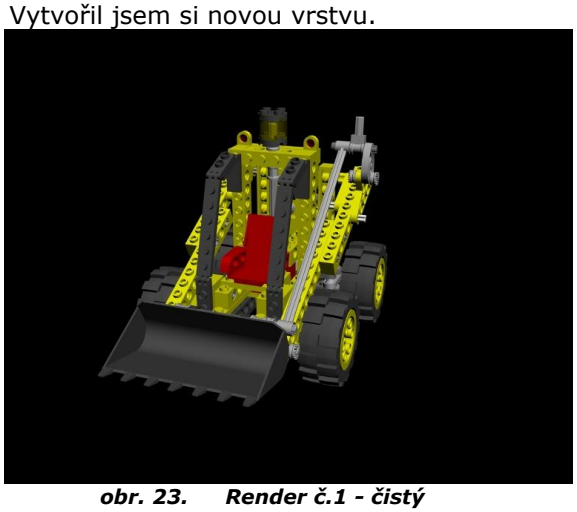


obr. 22. VRML model po vygenerování

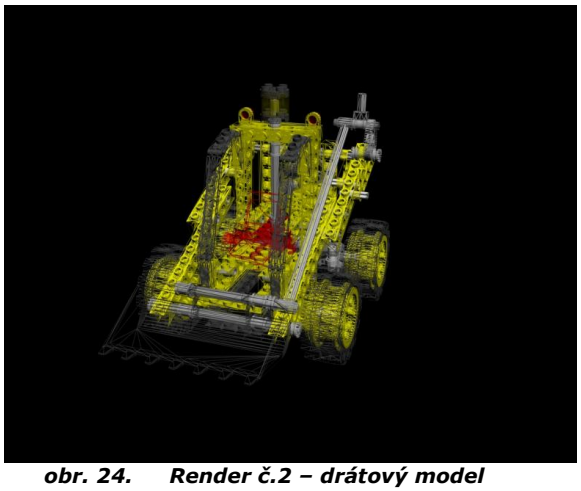
5.1.2 Grafický návrh

Návrh jsem začal tvořit již v 3DS Max. Využil jsem různé možnosti renderování finálního modelu k tvorbě dvou základních renderů – plného a drátového modelu. Poté jsem pokračoval v Adobe Photoshop. Tam jsem

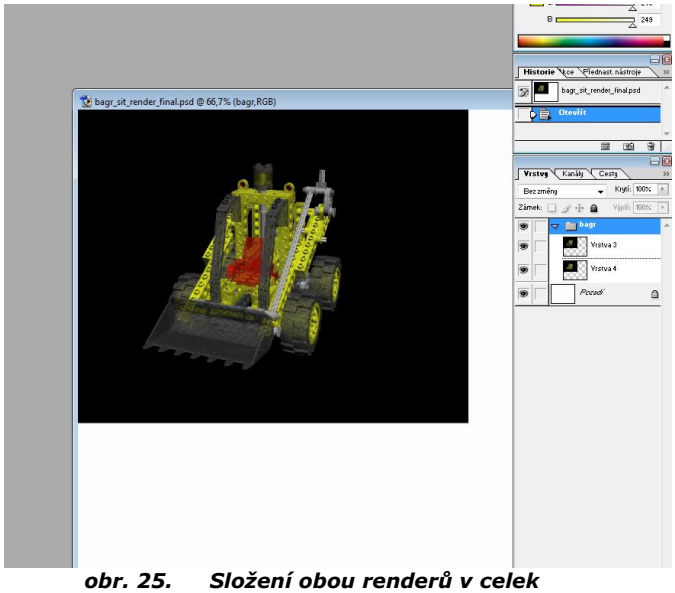
tyto dva obrázky pomocí průhlednosti vrstev složil do obrazu vyhovujícímu mé představě. Ten jsem exportoval do formátu JPEG, který je více vyhovující použití ve Flashi, i když standardní *.psd by Flash načetl také bez problémů, čili spíše kvůli velikosti souboru. Ve flashi jsem si vytvořil nový soubor, definoval velikost na 1280x1024. V tom jsem nacrtl základní rozvržení a postupně vyplnil barvou a následně i obrázkem bagru. Poté jsem jen definoval zbylé barvy a vše uzamkl, aby při další práci ,,nepřekáželo''. Vytvořil jsem si novou vrstvu.



obr. 23. Render č.1 - čistý



obr. 24. Render č.2 – drátový model



obr. 25. Složení obou renderů v celek

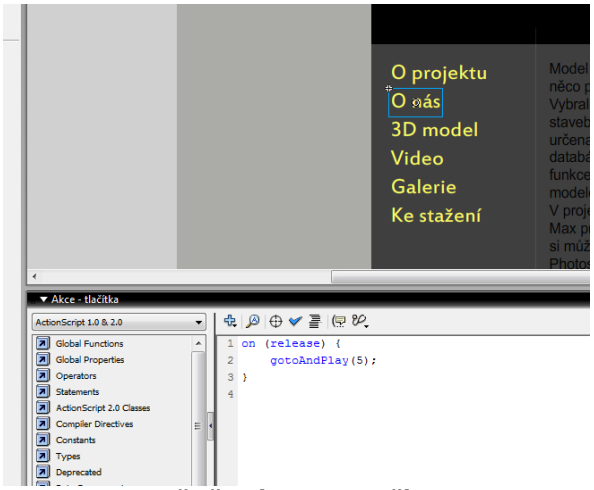
5.1.3 Tvorba menu

Do nové vrstvy jsem na určená místa vložil textová pole s popisky, kam budou budoucí odkazy odkazovat. Následně jsem všechna pole umístil tak, aby nebyla jen tak rozházená. Poté byla převedena použitím převedení na symbol – tlačítko, čímž se otevřely další možnosti práce s nimi jako s objekty. První, co jsem udělal, bylo že jsem proklikl tlačítko a zobrazila se mi další časová osa s možnostmi určení míst, na kterých bude tlačítko odkazovat, jestli změní barvu po kliknutí a jestli ji změní i po najetí. Zvolil jsem pouze možnost po najetí a upravil vložením klíčového snímku hodnotu barvy. Stejný postup jsem aplikoval na zbylých tlačítkách.
Poté jsem použil ActionScript – jazyk Flashe. Nadefinoval jsem kód tak, aby po tisku tlačítka odkazoval na určité políčko, které odpovídalo cíli.
Kód, který jsem při tom použil byl nejjednodušší možností.

```
on (release) {
        gotoAndPlay(x);
}
```

V kódu pak hodnota x znamená políčko animace, na kterou se po stisku tlačítka s takovým skriptem přesunete.
Pouze na volbu 3D model jsem použil

```
on (release) {
     getURL("abcd.html", "ramec");
}
```



obr. 26. Přidělení odkazu tlačítku v menu

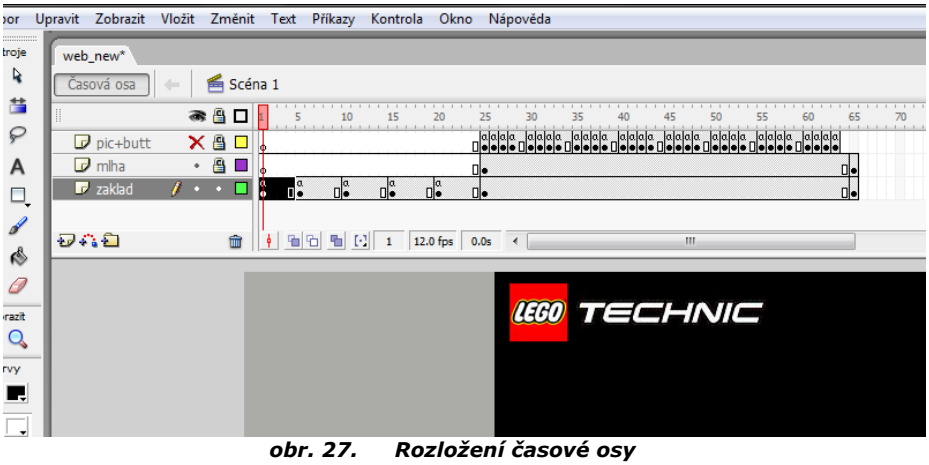
tady se dá zaměnit stejně jako u předchozích skriptů hodnota release např. na press nebo nějakou jinou dle použití. Abc.html nám určuje co chceme otevřít, použít se to dá nejen na stránky, ale i na soubory. Když chceme, aby nám odkaz fungoval i v HTML stránce po vložení *.swf souboru, musíme zapsat celou cestu – C://Documents an…./abcd.htm nebo cestu na web – http://www.xa..../abcd.htm . „RAMEC" nám pak určuje kam chceme stránku otevřít (např. _blank pro otevření nového okna).

5.1.4 Tvorba struktury a oddělení

Když bylo hotové menu, bylo potřeba převést strukturu z papíru do digitální formy. Čili jsem vytvořil 5 základních klíčových snímků, do kterých měly být vloženy obsahy jednotlivých částí. Na každý snímek jsem aplikoval skript

```
stop();
```

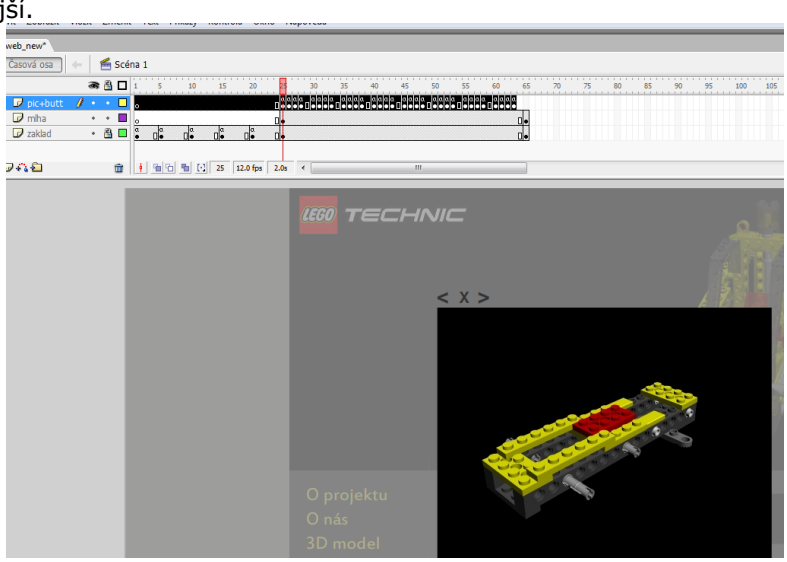
ten nám zastavuj animaci na každém snímku, kde je tento skript vložen. Čili můžeme odkazovat na jednotlivé snímky a prohlížet je než se přesuneme na jiné. Jednotlivá oddělení jsem potom naplnil textem – O projektu, textem a předem upravenou fotkou – O nás, videem – Video, textem s odkazy – ke stažení. Do galerie jsem importoval zmenšeniny prvních obrázků a přiřadil jim odkazy na další část – galerii.



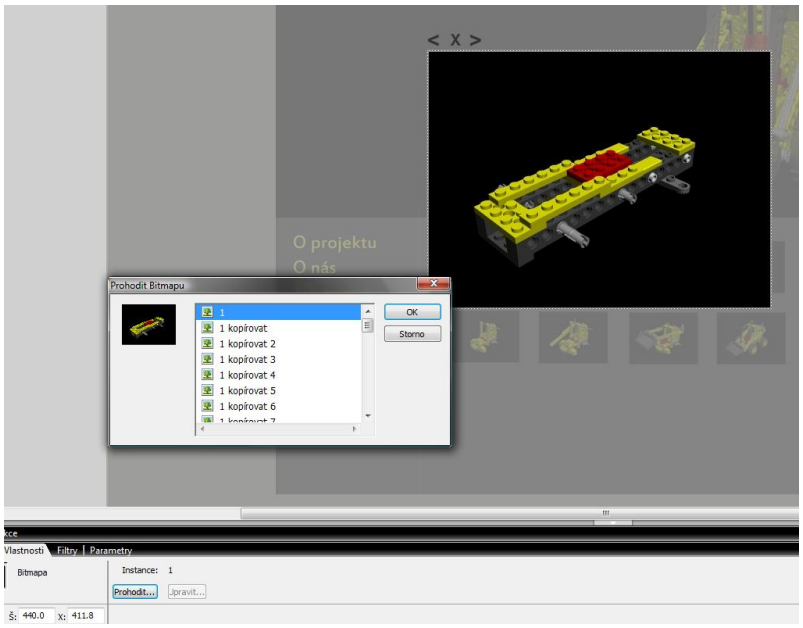
obr. 27. Rozložení časové osy

5.1.5 Tvorba galerie

Galerie byla časově nejnáročnější. Na začátku jsem vytvořil dvě vrstvy – ,,mlha" a ,,obrazky + jejich ovladani". Mlhu jsem hned na začátek oddělil klíčovými snímky tak, aby jejich vzdálenost přesahovala nutnou délku časové osy pro galerii. Poté jsem určil její průhlednost tak,aby vypadala jako vhodné pozadí pro obrázky. Následně jsem na první klíčový snímek galerie importoval první obrázek a poté přidal ovládací prvky galerie – dopředu,vzad a zpět do galerie. Těm jsem přiřadil hodnoty, které jsem pak musel u dopředu a vzad měnit vzhledem k číslu snímku. Následně jsem rozkopíroval tuto verzi po celé galerii a postupně měnil obrázky díky obsahové vlastnosti každé bitmapy – tak bere Flash obrázky. Poté jsem upravil hodnoty všech odkazů a prozkoušel funkčnost galerie. Občas se stává, že Flash vrací jinak formátované snímky a to je pak rychlejší vrátit se zpět a vše udělat znovu, neboť úprava je mnohdy složitější.



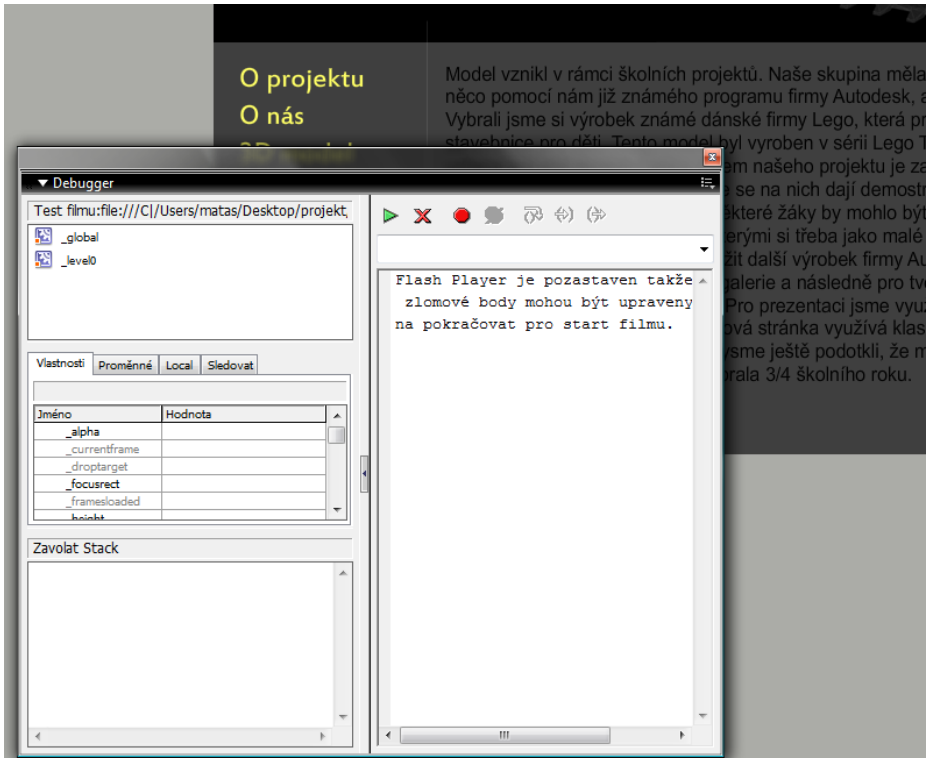
obr. 28. Vrstva galerie + vrstva mlhy – časová osa



obr. 29. Prohození bitmapy

5.1.6 Kontrola animace + publikace

Po skončení práce na stránce jsem postupně zkontroloval funkčnost všech prvků, snímků a formátování. Po odstranění pár chyb jsem mohl považovat stránku za dokončenou. Ještě jsem nastavil volby exportu a nastavil jejich parametry – formát *.swf , hlavně pak kompresní a kvalitativní volby. Nato jsem celou animaci vyexportoval. Do HTML jsem animaci nevkládal vzhledem k odkazům, které by musely obsahovat celé cesty a to je pro naše účely nežádoucí. Pro případnou stránku bych změnil hodnoty odkazů a znovu vyexportoval. Celá animace tedy byla hotova a spustitelna pod verzí Flash Playeru 8 a novější. Tento program je volně ke stažení, stačí jen zadat heslo do vyhledávače, v případě, že prohlížeč přehrávač nemá, program se ho sám zeptá na stažení pluginu a po potvrzení si ho sám stáhne a nainstaluje.



obr. 30. Debug filmu

6 Závěr, návrh možných řešení úlohy

Po půlročním snažení se projekt podařilo zrealizovat. Bagr je kompletně sestaven, součástky na sebe reagují, prezentace je hotova. Na projektu jsme si znovu ověřili velmi dobré vlastnosti užitých programů a práci s nimi. Je samozřejmě možné v nich vymodelovat nekonečné množství jiných součástí, přesto myslíme, že tato byla naprosto ideální.
Po nejdelší fázi modelování jsme součásti spojili vazbami, v galerii v naší prezentaci je dobře vidět, jak tento proces probíhal. Poté jsme celý model převedli do 3D studia kvůli možnosti renderování a širším prezentačním využitím.
Nakonec jsme zpracovali prezentaci, která obsahuje vše co nezbytné.
Určitě by se našla spousta programů, ve kterých by se toto vše dalo vytvořit, nicméně tyto jsou nejrozšířenější a myslíme, že jejich ovládnutím se člověk stane více ,,univerzálním", neboť na jejich principu pracuje více aplikací.
Závěrem můžeme jen doufat, že náš projekt osloví a že třeba již příštím rokem najde své následovníky.

Příloha A Letáky
A1 Česky

Software

Model jsme vytvořili v profesionálním CAD softwaru Inventor 11. Tato aplikace je schopna pracovat v 2D i 3D. Za pomoci jednoduchých pomůcek jsme schopni vytvořit velmi složité součástky a spoje. Pro prezentaci jsme použili velmi známé aplikace Adobe Photoshop a Adobe Flash. Tyto aplikace jsou uživatelsky nenáročné a používá je mnoho designerů po celém světě.



obr. 31. Možnost složení vytvořených součástek (scan z letáku)

Model

Model, který jsme vytvořili, je product firmy Lego, která vyrábí užásnou a velmi originální stavebnici pro děti i dospělé. Konkrétně náš model je Lego Technic číslo 8235. 8235 obsahuje zajímavé komponenty jako ozubené kolo, pneumatiky, lžíci rypadla. To je důvod proč jsme zvolili tento výrobek.

Účel

Doufáme, že náš projekt započne budování databáze Lego komponentů, takže za pár let bude škola schopna složit v Inventoru mnoho výrobků firmy Lego. Také studenti se mohou naučit, jak pracovat s aplikací, protože některé součástky jdou modelovat těžce a jsou na to potřeba znalosti.

Závěrem bychom chtěli poděkovat ing.Honsovi za spolupráci a pomoc.

A2 Anglicky

Software

Our model is created in professional CAD software called Inventor 11. This application is able to work in 2D or 3D. With simply clever tools we are able to create very complicated components and assembly. For presentation we have used well known applications Adobe Photoshop and Adobe Flash. These applications are user-friendly and they are used by many designers all over the world.



obr. 32. Náš model v propagačním letáku (scan z letáku)

Model

The model we have made is product of Lego, company that produce wonderful and very original brick-box for adults and children. Our model is Lego Technic 8235. 8235 contains a lot of amazing components like cogwheels, tires, excavator bucket. That is why we have chosen this product.

Purpose

We hope that our project will start building a library of Lego components , so in a few years school will be able to put together a lot of Lego products in Inventor. Also students can learn how to work in this application, because some of these parts are difficult to model and a great deal of knowledge is needed to create them.

In conclusion we would like to thank our guarantor ing.Honsa for his assistance.

Příloha B Seznam součástek

Odkaz

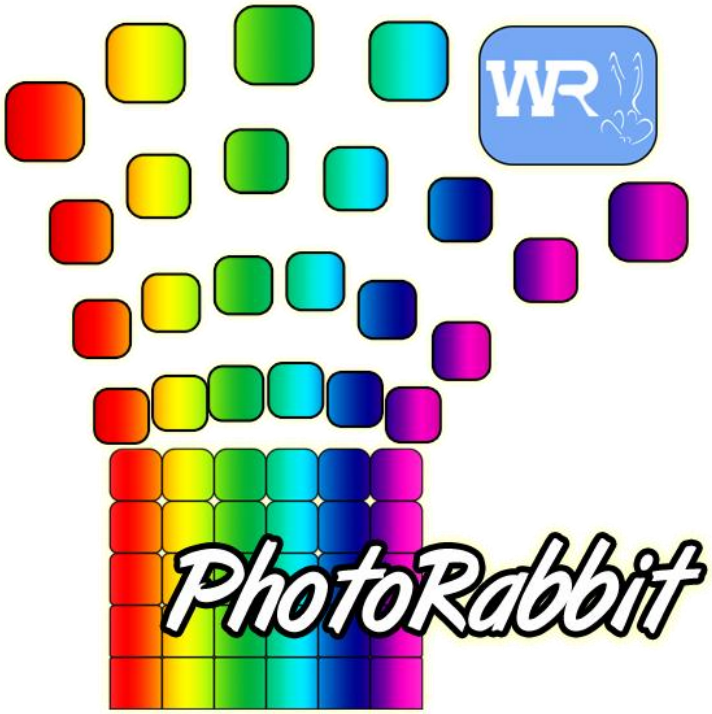
Středoškolská technika 2008

**Setkání a prezentace prací
středoškolských studentů na ČVUT**

GRAFICKÝ EDITOR

Lukáš Kotrba

**SSPŠ (Smíchovská střední průmyslová škola)
Preslova 25, Praha 5 - Smíchov**



Obsah:

1. Úvod
2. Teorie
   2.1. Bitmapová grafika (rastrová)
   2.2. MS Visual Basic
   2.3. Štětec
   2.4. Kapátko
   2.5. Barevný model RGB
   2.6. Barevný model HSB (HSV)
   2.7. Aditivní míchání barev
   2.8. Subtraktivní míchání barev
   2.9. Hexadecimální zápis barvy
3. Manuál
   3.1. Seznámení s WR PhotoRabbitem
      3.1.1. Co je co ve WR PhotoRabbitu
   3.2. Základní ovládání
      3.2.1. Práce se souborem
      3.2.2. Základní nástroje
   3.3. Pokročilé filtry
4. Závěr

Seznam obrázků

Obr. 1 – Srovnání vektorové (vlevo) a rastrové (vpravo) grafiky .....
Obr. 2 – Různé druhy nastavení vyhlazení a tlaku (stopa 100 px) .....
Obr. 3 – Ukázky funkcí kapátka (zvětšení 400%) .....
Obr. 4 – Ukázka jednotkové krychle barevného prostoru RGB .....
Obr. 5 – Různé způsoby zobrazení barevného modelu HSB (válcová a kuželová) .....
Obr. 6 – Ukázka aditivního míchání barev .....
Obr. 7 – Subtraktivní míchání barev .....
Obr. 8 – Pracovní plocha .....
Obr. 9 – Roletová nabídka .....
Obr. 10 – Panel nástrojů .....
Obr. 11 – Dialogové okno pro otevření souboru .....
Obr. 12 – Dialogové okno nového souboru .....
Obr. 13 – Nástroj Výběr .....
Obr. 14 – Různé druhy nastavení vyhlazení a tlaku (stopa 100 px) .....
Obr. 15 – Panel nabídek Text .....
Obr. 16 – Nástroj Pravítko .....
Obr. 17 – Ukázky funkcí kapátka (zvětšení 400%) .....
Obr. 18 – Dialogové okno pro výběr barvy .....
Obr. 19 – Filtr Pixelizace .....
Obr. 20 – Filtr Pointilizace .....
Obr. 21 – Filtr Solarizace .....
Obr. 22 – Leták i
Obr. 23 – Booklet ii
Obr. 24 – Potisk CD iii
Obr. 25 – Internetová prezentace – Hlavní stránka iv
Obr. 26 – Internetová prezentace – O projektu iv
Obr. 27 – Internetová prezentace – Fotogalerie v
Obr. 28 – Internetová prezentace – Downloads v
Obr. 29 – Internetová prezentace – About vi
Obr. 30 – Prezentace MS PowerPoint – Úvodní obrazovka vii
Obr. 31 – Prezentace MS PowerPoint – O firmě vii
Obr. 32 – Prezentace MS PowerPoint – Seznámení s tématem viii
Obr. 33 – Prezentace MS PowerPoint – Úvodní obrazovka viii
Obr. 34 – Prezentace MS PowerPoint – Filtr – Pixelizace ix
Obr. 35 – Prezentace MS PowerPoint – Filtr – Pointilizace ix
Obr. 36 – Prezentace MS PowerPoint – Filtr – Solarizace x
Obr. 37 – Prezentace MS PowerPoint – Filtr – Karikatura x
Obr. 38 – Prezentace MS PowerPoint – Filtr – Dualizace xi
Obr. 39 – Prezentace MS PowerPoint – Jak to funguje??? xi
Obr. 40 – Prezentace MS PowerPoint – Ukázky nástrojů xii
Obr. 41 – Prezentace MS PowerPoint – Střípky z projektu xii
Obr. 42 – Prezentace MS PowerPoint – Doprovodné produkty xiii
Obr. 43 – Prezentace MS PowerPoint – Závěr xiii
Obr. 44 – Program – O aplikaci WR PhotoRabbit – frmAbout xiv
Obr. 45 – Program – Dialogové okno pro výběr barvy – frmBar xiv
Obr. 46 – Program – Dialogové okno filtrů – frmFil xv
Obr. 47 – Program – Dialogové okno uložení do JPEG – frmJPEG xv

Obr. 48 – Program – Panel nástrojů – frmNas xvi
Obr. 49 – Program – Formulář dokumentu – frmOb xvi
Obr. 50 – Program – Paleta barev – frmPal xvii
Obr. 51 – Program – Posuvník – frmPos xvii
Obr. 52 – Program – Výběr tloušťky stopy – frmSte xvii
Obr. 53 – Program – Nový soubor – frmVel xvii
Obr. 54 – Program – Hlavní formulář – mdiMain xviii
Obr. 55 – Program – Splash screen – frmSplashScreen xix

Seznam příloh

Příloha A – Leták i
Příloha B – Booklet ii
Příloha C – Potisk CD iii
Příloha D – Internetová prezentace iv
D.1. Hlavní stránka iv
D.2. O projektu iv
D.3. Fotogalerie v
D.4. Downloads v
D.5. About vi
Příloha E – Prezentace MS PowerPoint vii
E.1. Úvodní obrazovka vii
E.2. O firmě vii
E.3. Seznámení s tématem viii
E.4. Úvodní obrazovka viii
E.5. Filtr – Pixelizace ix
E.6. Filtr – Pointilizace ix
E.7. Filtr – Solarizace x
E.8. Filtr – Karikatura x
E.9. Filtr – Dualizace xi
E.10. Jak to funguje??? xi
E.11. Ukázky nástrojů xii
E.12. Střípky z projektu xii
E.13. Doprovodné produkty xiii
E.14. Závěr xiii
Příloha F – Program xiv
F.1. O aplikaci WR PhotoRabbit – frmAbout xiv
F.2. Dialogové okno pro výběr barvy – frmBar xiv
F.3. Dialogové okno filtrů – frmFil xv
F.4. Dialogové okno uložení do JPEG – frmJPEG xv
F.5. Panel nástrojů – frmNas xvi
F.6. Formulář dokumentu – frmOb xvi
F.7. Paleta barev – frmPal xvii
F.8. Posuvník – frmPos xvii
F.9. Výběr tloušťky stopy – frmSte xvii
F.10. Nový soubor – frmVel xvii
F.11. Hlavní formulář – mdiMain xviii
F.12. Splash screen – frmSplashScreen xix
Příloha G – Zdrojový kód xx

G.1. O aplikaci WR PhotoRabbit – frmAbout xx
G.2. Dialogové okno pro výběr barvy – frmBar xx
G.3. Dialogové okno filtrů – frmFil xxix
G.4. Dialogové okno uložení do JPEG – frmJPEG xxxvii
G.5. Panel nástrojů – frmNas xxxix
G.6. Formulář dokumentu – frmOb xlii
G.7. Paleta barev – frmPal xciii
G.8. Posuvník – frmPos xcvi
G.9. Splash Screen – frmSplashScreen xcvii
G.10. Výběr tloušťky stopy – frmSte c
G.11. Nový soubor – frmVel cii
G.12. Hlavní formulář – mdiMain cv
G.13. Modul Barvy – Barvy cxiv
G.14. Modul Doplnení – Doplněni cxvii
G.15. Modul File Open – FileOpen cxviii
G.16. Modul INI – Ini cxxiii
G.17. Modul Intel JPEG Library – IntelJPEGLibrary cxxiii
G.18. Modul Malování – Malovani cxxv
G.19. Modul Proměnné – promene cxxvii
G.20. Modul Spash Screen – SplashScreen cxxviii
G.21. Vlase modul DIB Section – cDIBSection cxxix
G.22. Vlase modul GIF – GIF cxxxii

1. Úvod

WR PhotoRabbit je český rastrový grafický editor pro nejširší použití, právě takový jaký by neměl chybět v žádném počítači. Svým jednoduchým a přehledným ovládáním na principu rozumné dostatečnosti je určeno především pro běžné uživatele se základní znalostí Windows bez nároků na další znalosti v oblasti grafické tvorby. Pokryje vaše požadavky od tvorby jednoduchého nápisu až po složité ilustrace. Využijete jej k tvorbě působivých propagačních tiskovin a letáků, vizitek, cenovek, hlavičkového papíru, plánků, nápisů, pohlednic, blahopřání atd. Nabízí i možnost tvorby grafiky pro internetové prezentace.

Základním impulzem pro započetí vývoje se bezpochyby stal můj postup do čtvrtého ročníku SSPŠ, jehož součástí je celoroční práce na vybraném projektu. Jelikož jsem už dlouho snil o tom, že udělám vlastní grafický editor, tak když jsem se na začátku roku rozhodoval, co budu dělat, WR PhotoRabbit byl pro mě jasná volba.

V průběhu celého vývoje bylo zajímavé sledovat, jak se vše mění přímo pod mýma rukama. Věci, které se zpočátku zdáli být banální se postupně přeměňovali na větší a větší oříšky které bylo potřeba rozlousknout. Z řad nevětších je třeba jmenovat vlastní textový „engine“, který byl potřeba pro interaktivní zadávání textu, jaké známe třeba z věhlasného Adobe Photoshop.

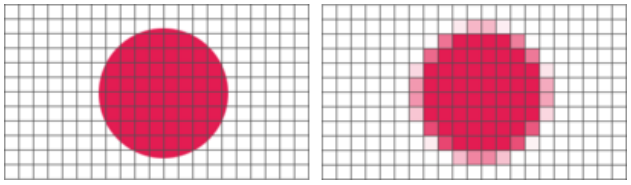
Součástí celého projektu je krom samotného programu také propagační leták, internetová prezentace, prezentace v MS PowerPoint, booklet a potisk CD.

2. Teorie

2.1. Bitmapová grafika (rastrová)

Bitmapová grafika (rastrová grafika) je jeden ze dvou základních způsobů, jakým počítače ukládají a zpracovávají obrazové informace. Spolu s vektorovou grafikou představují dva základní způsoby ukládání obrázků.

V bitmapové grafice je celý obrázek popsán pomocí jednotlivých barevných bodů (pixelů). Body jsou uspořádány do mřížky. Každý bod má určen svou přesnou polohu a barvu (např. RGB). Tento způsob popisu obrázků používá např. televize nebo digitální fotoaparát. Kvalitu záznamu obrázku ovlivňuje především rozlišení a barevná hloubka.



Obr. 1 – Srovnání vektorové (vlevo) a rastrové (vpravo) grafiky

2.2. MS Visual Basic

MS Visual Basic je následovník rodiny programovacích jazyků BASIC, navrženém v roce 1963 Johnem G. Kemeny a Thomasem E. Kurtzem. Název BASIC je zkratkou anglických slov Beginner's All–purpose Symbolic Instruction Code.

Zásadní výhodou tohoto jazyka je jeho jednoduchost a snadná pochopitelnost pro začátečníky a velmi rychlý návrh aplikací zejména pro jednoduché databázové aplikace.

2.3. Štětec

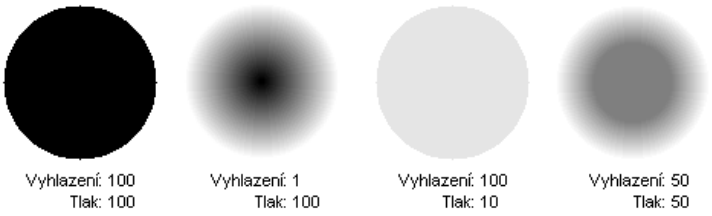
Štětci, jak je chápán v tomto programu, byly dány čtyři vlastnosti – velikost stopy, vyhlazení, tlak a kontinuální kreslení. Štětec samotný funguje tak, že při stisknutí myši, kdekoli na plátně, se vybere čtverec o stranách stejných, jako je průměr štětce. Poté se projíždí každý jednotlivý pixel v daném čtverci a je – li jeho vzdálenost od středu menší nebo rovna průměru stopy tak se pixel přebarví v závislosti na velikosti tlaku a vyhlazení.

Velikost stopy asi není třeba představovat, jedná se o velikost průměru kruhové stopy, která bude vykreslena na plátno.

Vyhlazení je vlastnost specifická právě pro štětec. Procentuelně určuje, jaká část vykreslené stopy bude vyhlazena směrem od okraje ke středu.

Tlak představuje pomyslný tlak na štětec, který se projevuje průhledností. Hodnota se opět uvádí v procentech.

Při rychlém pohybu kurzorem po plátně dochází k tomu, že jsou zaznamenávány pouze polohy v určitých intervalech, tato nevýhoda je odstraněna pomocí vlastnosti „kontinuální kreslení“ kde se dopočítává zbytek trasy a je tak zajištěna plynulá čára. Tuto vlastnost lze z prezentačních důvodů vypnout, v ostré verzi by neměla jít vypnout.



Obr. 2 – Různé druhy nastavení vyhlazení a tlaku (stopa 100 px)

2.4. Kapátko

Kapátko je jednoduchý nástroj zjišťující barvu z libovolného bodu plátna. Zajímavější jsou jeho další funkce – Průměr 3×3 a Průměr 5×5. Tady se načte barva každého bodu ze čtverce o stranách 3×3 (5×5) a zprůměrňuje.



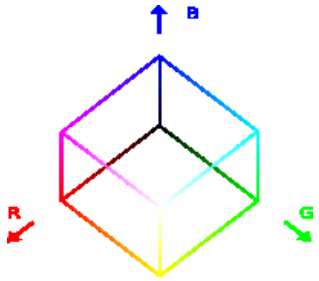
Obr. 3 – Ukázky funkcí kapátka (zvětšení 400%)

2.5. Barevný model RGB

Barevný model **RGB** neboli červená – zelená – modrá je aditivní způsob míchání barev používaný ve všech monitorech a projektorech (jde o míchání vyzařovaného světla), tudíž nepotřebuje vnější světlo.

Základní složkou tohoto modelu jsou barvy **R**ed (červená), **G**reen (zelená) a **B**lue (modrá). Současný výskyt všech tří barev se projevuje barvou bílou, jejich absence naopak černou barvou. Každá složka nabývá hodnot od 0 do 255 (8 bitů), přičemž 255 znamená 100% intenzity barvy a 0 naopak 0% intenzity

Model RGB je možné zobrazit jako krychli, ve které každá ze stran udává intenzitu (mohutnost) každé barevné složky. Potom libovolný bod se souřadnicemi (R, G, B) v této krychli udává hodnotu své výsledné barvy.



Obr. 4 – Ukázka jednotkové krychle barevného prostoru RGB

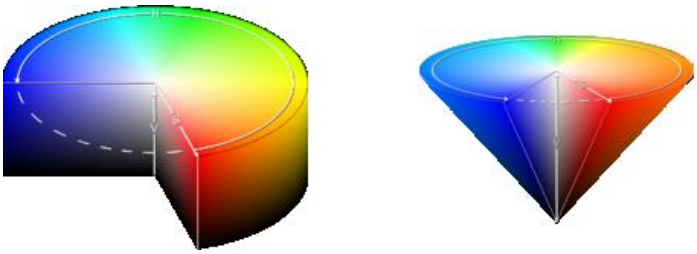
2.6. Barevný model HSB (HSV)

HSB (**H**ue, **S**aturation, **B**rightness), je barevný model, který vytvořil v roce 1978 Alvy Ray Smith. Tento barevný model nejvíce odpovídá lidskému vnímání barev. Sestávající ze tří složek (nejsou to základní barvy), u nichž je nutno hlídat hodnoty (možné nesmyslné kombinace).

Hue – barevný tón, převládající. Neboli odstín – barva odražená nebo procházející objektem. Měří se jako poloha na standardním barevném kole (0° až 360°). Obecně se odstín označuje názvem barvy.

Saturation – sytost barvy, příměs jiné barvy. Někdy též chroma, síla nebo čistota barvy, představuje množství šedi v poměru k odstínu, měří se v procentech od 0% (šedá) do 100% (plně sytá barva). Na barevném kole vzrůstá sytost od středu k okrajům. Např. červená s 50% sytostí bude růžová.

Brightness – hodnota jasu, množství bílého světla. Relativní světlost nebo tmavost barvy. Jas vyjadřuje, kolik světla barva odráží, dalo by se také říct přidávání černé do základní barvy.



Obr. 5 – Různé způsoby zobrazení barevného modelu HSB (válcová a kuželová)

2.7. Aditivní míchání barev

Aditivní míchání barev je takový způsob míchání barev, kdy se jednotlivé složky barev sčítají a vytváří světlo větší intenzity. Výsledná intenzita se rovná součtu intenzit jednotlivých složek. Tento způsob se používá převážně u barevného modelu RGB.

Aditivní míchání barev odpovídá vzájemnému prolínání tří barevných kuželů světla ze tří reflektorů na bílém plátně. Každý reflektor má filtr odpovídající základní barvě.

Část plátna, která je osvětlená rovnoměrně všemi třemi reflektory, je bílá. Když smícháme jen dvě barvy světla, např. červené a zelené, dostaneme barvu žlutou. Budeme-li clonou měnit poměr intenzity obou světel, dostaneme různé barevné odstíny mezi těmito barvami.

Modrá a zelená barva ve stejném poměru dávají azurovou barvu, červená a modrá dávají barvu purpurovou.

Smícháním dvou základních barev vznikne třetí, základní barva, která je barvou komplementární (doplňkovou).



Obr. 6 – Ukázka aditivního míchání barev

2.8. Subtraktivní míchání barev

Subtraktivní míchání barev je způsob míchání barev, kdy se s každou další přidanou barvou ubírá část původního světla. Pokud například skládáme na sebe barevné filtry nebo mícháme pigmentové barvy, mícháme je subtraktivní metodou.

Světlo prochází jednotlivými barevnými vrstvami a je stále více pohlcováno. Výsledná barva se skládá z vlnových délek, které zbudou po odrazu nebo průchodu filtrem.

Základní barvy subtraktivního míchání jsou komplementární (doplňkové) k základním barvám při jejich aditivním míchání. Jsou to tedy barvy: žlutá, purpurová a azurová.

Smícháním modrozelené (azurové) a žluté barvy vznikne barva zelená, žluté a purpurové barva červená a purpurové a modrozelené barva modrá. Smícháním všech tří základních barev dostaneme barvu černou.



Obr. 7 – Subtraktivní míchání barev

2.9. Hexadecimální zápis barvy

Hexadecimální (dále jen hex) zápis barvy je jedno číslo o šesti pozicích, jeho maximální hodnota je tedy 16777215 dec. Každá dvojice v hex tvaru reprezentuje jednotlivé barevné složky RGB – FF0000 hex tedy znamená (255,0,0) RGB. Přepočet z hex na jednotlivé RGB složky je jednoduchý – jsou zde dvě možnosti.

Můžeme hex zápis rozdělit na dvojice a pak přepočítat na decimální hodnoty. Například číslo ABCDEF hex – AB h = 171 d, CD h = 205 d a EF h = 239 d, takže RGB (171,205,239).

Druhý způsob je převést celé číslo na decimální zápis a později rozdělit. Např. ABCDEF h = 11259375 d, B = 11259375 mod 256 = 239, G = (11259375 / 256) mod 256 = 205 a R = 11259375 / 65536 = 171 z toho plyne, že RGB (171,205,239).

3. Manuál

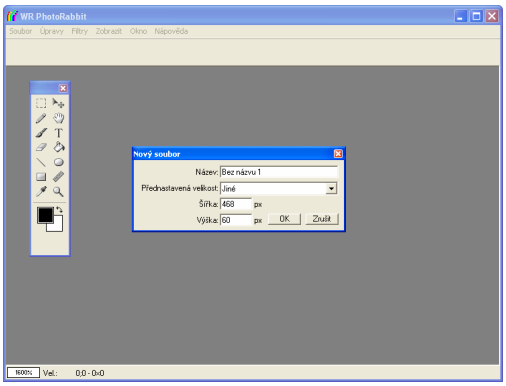
3.1. Seznámení s WR PhotoRabbitem

První část uživatelské příručky je „Seznámení s WR PhotoRabbitem“, která vám pomůže se seznámit se se základními schopnostmi a ovládáním WR PhotoRabbit, a to tou nejrychlejší a nejnázornější formou, jaká vůbec existuje. V této první části se seznámíte s prostředím a ovládáním programu.

3.1.1. Co je co ve WR PhotoRabbitu

Pracovní prostředí WR PhotoRabbita

Po spuštění WR PhotoRabbita se zobrazí obdobná obrazovka, jakou vidíte na obrázku.



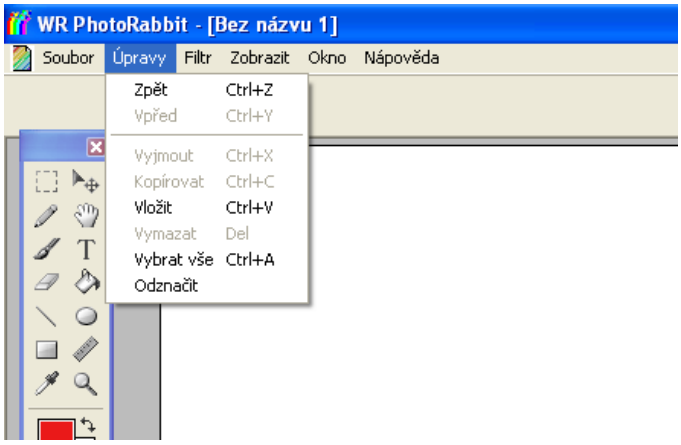
Obr. 8 – Pracovní plocha

V titulním řádku (modrý pruh nahoře) vidíte název dokumentu, který je právě otevřený. Těsně pod ním je roletová nabídka a pod ní lišta voleb jednotlivých nástrojů. Dole se nachází stavový řádek s doprovodnými informacemi. Pracovní plocha je vpravo a dole olemována posuvníky, je–li jich třeba. A asi nejvýraznější je panel nástrojů v samostatném okně.

Roletové nabídky

Roletové nabídky (menu) jsou již samozřejmou součástí snad každého programu, který pod Windows pracuje. Na pracovní ploše totiž zabírá minimum místa a přitom poměrně snadno a přehledně zpřístupní uživateli mnoho funkcí, které daný program umožňuje.

Klepnutím myši na položku roletové nabídky, popř. stiskem klávesy Alt a příslušného podtrženého písmene, rozbalíte seznam příkazů, z nějž si vybíráte opět myší. U některých z nich je i klávesová zkratka, časem si ty nejčastěji používané jistě osvojíte. U některých je vpravo šipka, která symbolizuje přítomnost podnabídky, tu rozbalíte klepnutím na tuto položku, nebo jen najetím a setrváním nad ní. Jiné položky fungují jako přepínač, jsou – li zapnuty, je vlevo symbol zaškrtnutí.



Obr. 9 – Roletová nabídka

Panel nástrojů

K ovládání většiny funkcí slouží panel nástrojů, umístěný v samostatném okně – jak vidíte na obrázku.



Obr. 10 – Panel nástrojů

Většina tlačítek na tomto panelu slouží jako přepínače, jsou-li zamáčknuté, znamená to stav „zapnuto“. V dolní části je vidět barva popředí i pozadí, funkce ostatních tlačítek si vysvětlíme později.

Posuvníky

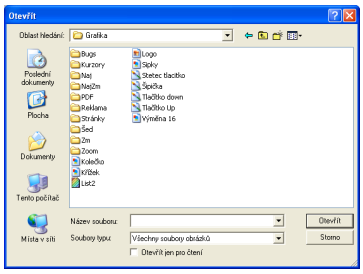
Posuvníky jsou nezbytnou součástí pohybu v oknech i textových polích. I ve WR PhotoRabbitu lemují zleva a zdola okno pracovní plochy a slouží k pohybu po dokumentu, který se při standardním zvětšení nevejde do okna.

3.2. Základní ovládání

3.2.1. Práce se souborem

Otevření souboru

V programu WR PhotoRabbit lze otevírat rastrové soubory typu GIF, JPEG, TIFF, ICO a BMP. Otevření provedete kliknutím v nabídce **Soubor → Otevřít** (Ctrl + O), následně se před vámi zobrazí dialogové okno pro otevření souboru.



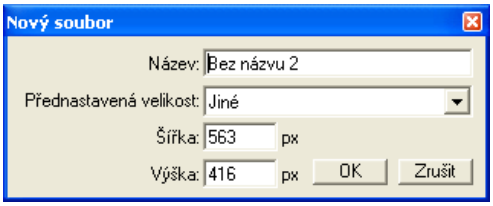
Obr. 11 – Dialogové okno pro otevření souboru

Uložení souboru

Uložení obrázku je možné ve formátech BMP, GIF a JPEG. Uložení provedete v nabídce **Soubor → Uložit** (Ctrl + S), tím uložíte otevřený soubor na jeho původní umístění. Příkazem **Soubor → Uložit jako** uložíte soubor na místo, které si zvolíte pomocí dialogového okna pro uložení souboru.

Vytvoření nového souboru

Pro otevření nového souboru zvolte **Soubor → Nový** (Ctrl + N). Nyní se před vámi objeví dialogové okno nového souboru



Obr. 12 – Dialogové okno nového souboru

Do pole **Název,** můžete napsat název nového souboru, který se vám bude zobrazovat v titulku okna. **Pole Přednastavené** velikosti obsahuje několik nejpoužívanějších velikostí, a zrychluje tím otevření normalizovaných formátů. Do polí **Šířka** a **Výška** se udávají rozměry obrázku v pixelech.

3.2.2. Základní nástroje

Výběr

Výběr určité části obrázku se provádí takzvaným ohraničováním. To spočívá v tom, že tahem myši stanovíte obdélníkové pole, s jehož obsahem pak můžete pracovat odděleně (např. používat filtry, kreslit, mazat apod.). Odznačení provedete opětovným kliknutím kamkoli na obrázek, nebo pomocí nabídky **Úpravy → Odznačit**. Vše vyberete v nabídce **Úpravy → Vybrat vše** (Ctrl + A).



Obr. 13 – Nástroj Výběr

Přesun

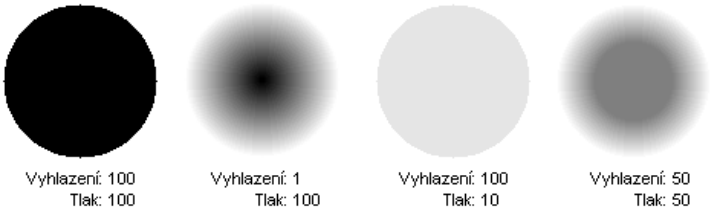
Pokud máte vybranou nějakou část obrázku, můžete jen přesunout pomocí nástroje přesun. Přesouvání probíhá stisknutím a přesunem. Jste – li spokojeni s polohou přesunu, stačí uvolnit tlačítko myši.

Tužka

Nástroj tužka kreslí souvislou plnou čáru bez vyhlazení. Kreslení se provádí stiskem a tahem myši.

Štětec

Štětec na rozdíl od tužky kreslí souvislou plnou čáru s volitelným vyhlazením a průhledností. Ukázky jednotlivého nastavení můžete vidět na obrázku níže.



Obr. 14 – Různé druhy nastavení vyhlazení a tlaku (stopa 100 px)

Text

Za pomoci nástroje text můžete zadávat text v několika variantách.



Obr. 15 – Panel nabídek Text

První nabídka umožňuje podobně jako v textových editorech vybrat typ písma, druhá velikost a třetí řez písma. Čtvrtá umožňuje přidat písmu podtržení, přeškrtnutí nebo obojí najednou. Předposlední nabídka umožňuje psát text s obrysem, bez obrysu nebo pouze obrys,

a poslední typ obrysu (plný, čárkovaný, tečkovaný, čerchovaný nebo dvojitě čerchovaný), tyto vlastnosti lze však přiřadit pouze, je–li tloušťka obrysu rovna 1. Čtvereček před šířkou obrysu značí barvu, jakou je obrys vykreslen. Poslední dvě tlačítka slouží pro zrušení resp. potvrzení napsaného textu.

Guma

Guma umožňuje mazat nakreslený obrázek barvou pozadí.

Plechovka barvy

Plechovka barvy je chytrý nástroj, který umožňuje vyplnit sousedící body stejné barvy barvou jinou.

Čára

Nástroj **Čára** dovoluje uživateli nakreslit rovnou čáru, zadáním počátečního a koncového bodu. Speciální funkcí je kreslení se stisknutou klávesou **Shift**, poté se čára kreslí s krokem 45°. Jediným atributem Čáry je její **Tloušťka**, ta se udává v pixelech.

Elipsa

Nástrojem **Elipsa** se kreslí souvislá elipsa. Kreslení probíhá zadáním dvou protějších rohů. Při stisknuté klávese **Shift** se nakreslí kružnice. Je-li stisknutá i kláves **Ctrl**, kreslení probíhá od středu.

Obdélník

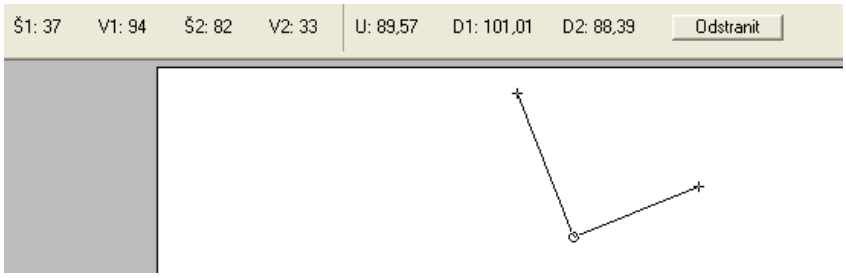
Obdélník podobně jako elipsa se kreslí pomocí dvou protilehlých rohů, v případě stisknutého **Shift,** se kreslí pravidelný čtverec a při stisknutém **Ctrl** se opět kreslí od středu.

Pravítko

Pravítko je nástroj, který neslouží k upravování samotného obrázku, nicméně ke změření vzdálenosti a úhlu dvou bodů či tří

Měření probíhá následujícím způsobem. Stisknutím a tažením určíte první dva body, v průběhu zadávání můžete sledovat v panelu nabídek aktuální šířku, výšku, délku a úhel od nulové osy. Při stisknutém Ctrl můžete z libovolného konce vést druhou úsečku, opět se vám ukazuje aktuální šířka, výška, délka a kladný úhel, který tyto dvě úsečky svírají.

Tyto úsečky zůstávají viditelné pouze při aktivním nástroji Pravítko, chcete-li tyto úsečky odstranit, klikněte na tlačítko **Odstranit** v panelu nabídek.



Obr. 16 – Nástroj Pravítko

Kapátko

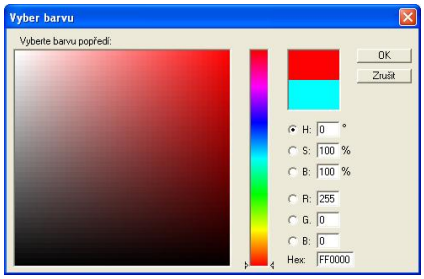
Nástroj **Kapátko** je jednoduchý nástroj, který načte barvu z libovolného bodu kdekoli na obrázku. U tohoto nástroje máte volbu tří způsobů načítání barvy: barva z **jednoho bodu**, ze čtverce o stranách **3×3** nebo **5×5** pixelů. Jak jednotlivá nastavení vypadají ve výsledku, se můžete podívat na obrázku níže.



Obr. 17 – Ukázky funkcí kapátka (zvětšení 400%)

Barva popředí a pozadí

Na panelu nástrojů se také nachází ikony barev popředí a pozadí. Kliknutím na libovolnou z nich se vám otevře dialogové okno pro výběr barev, kde lze vybrat barvu ručně, nebo zadáním ve formátu HSB, RGB nebo hexadecimálním zápisu.



Obr. 18 – Dialogové okno pro výběr barvy

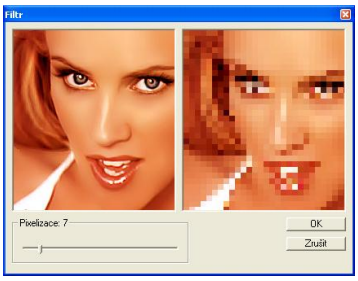
Prohození barev

Posledním prvkem na panelu nástrojů je oboustranná šipka, sloužící k rychlému prohození barev popředí a pozadí.

3.3. Pokročilé filtry

Pixelizace

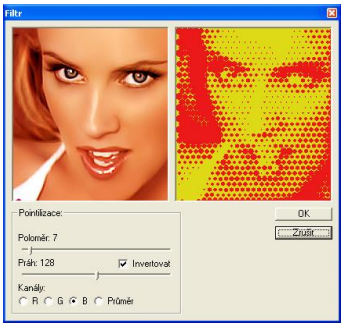
Pixelizace je jednoduchý filtr známí především z médií, kde se používá pro utajení identity osob na fotografiích. Tento filtr má jen jeden parametr, který značí, jakou měrou bude obrázek pixelizován.



Obr. 19 – Filtr Pixelizace

Pointilizace

Pointilizace je nástroj který převádí obrázek do množiny bodů s různým průměrem tak, aby základní rysy obrázku zůstaly zachovány.

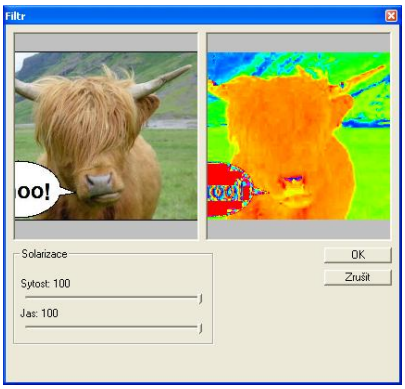


Obr. 20 – Filtr Pointilizace

Tento filtr má několik nastavení. **Poloměr** slouží k nastavení maximálního poloměru bodů a tím také hustotu bodů – čím více, tím větší body ale také menší počet bodů. **Práh** značí mezní hodnotu od které je bod vykreslen. Políčko **Invertovat** přehodí práh – čím méně barvy tím větší kolečko. **Kanál** určuje, dle jaké barevné složky se body určují.

Solarizace

Solarizace je filtr, který vyvolává dojem termovize, jeho parametry jsou **Jas** a **Sytost**, které udávají výsledný jas a sytost obrázku.



Obr. 21 – Filtr Solarizace

4. Závěr

Od samého počátku kdy jsem se pln elánu řítil ze školy domů, jen abych mohl dále pracovat a tvořit na svém programu uplynulo již mnoho dní, a ačkoli nadšení do celé práce místy ochabovalo, jsem celý projekt dotáhl do zdárného konce. Zároveň se zakončením práce a započetím hledání chyb a nepřesnosti jsem začal neustále více obdivovat rozsáhlé vývojové týmy, které vyvíjí programy takového kalibru jako můj oblíbený Adobe Photoshop. Neboť sám sem si vyzkoušel jaké je to pracovat na „velkém“ projektu, a jaká je práce se později v celém kódu vyznat.

Naštěstí jsem díky své prozíravosti tvořil už od začátku každou proceduru s důrazem na její pozdější rozvoj, a implementování každé další součásti bylo mnohdy jednodušší, než jsem očekával. Vzhledem k těmto okolnostem se mnohde vyskytují větvené podmínky, které mají přiřazenou funkci pouze u jedné větve, jelikož z nedostatku času jsem nemohl dále pokračovat na jejich rozvoji, a byl jsem nucen se zaměřit především na důležitější věci.

Celou práci navíc usnadňoval fakt, že když už scházela motivace do psaní kódu, mohl jsem svůj nevyřáděný elán vrhnout do jiných doprovodných produktů jako leták, booklet, potisk CD a další. Mnohé z těchto produktů také nesou známky toho, že byly vyráběny při částečném odporu či nechuti pokračovat v projektu, a tak se zde dají nelézt věci dosud nevídané. Navzdory všem těmto komplikacím jsem celý projekt se vším všudy ukončil v požadovaném čase a s požadovanou – ne-li vyšší – kvalitou.

Závěrem bych rád zdůraznil, že celá práce byla pro mě zábavou, kterou jsem dělal ve většině případů s chutí a radostí, to však nesmí zastřít fakt, že byla nesmírně namáhavá a zdlouhavá. Obzvláště v začátcích kdy vše utíkalo pomalu a celý základ se teprve rodil, byla práce nejtěžší, neboť každá změna neměla kýžený efekt a mnohdy se jednalo pouze o procedury běžící na pozadí. Postupem času se však začala každá změna více a více projevovat což samozřejmě podporovalo i mé sebevědomí, které mě hnalo neustále kupředu.

Příloha A – Leták



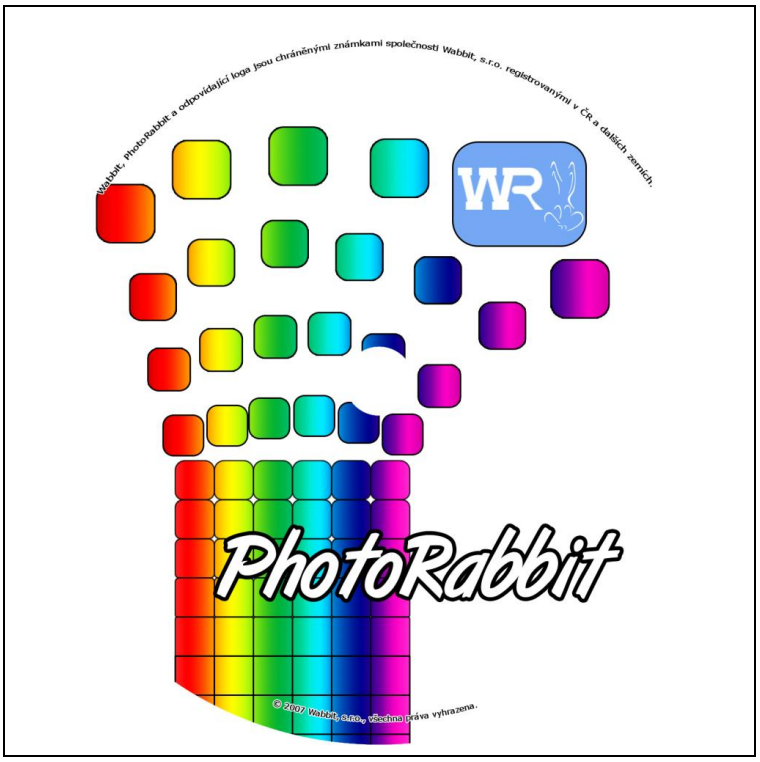
Obr. 22 – Leták

Příloha B – Booklet



Obr. 23 – Booklet

Příloha C – Potisk CD



Obr. 24 – Potisk CD

Příloha D – Internetová prezentace

D.1. Hlavní stránka



Obr. 25 – Internetová prezentace – Hlavní stránka

D.2. O projektu



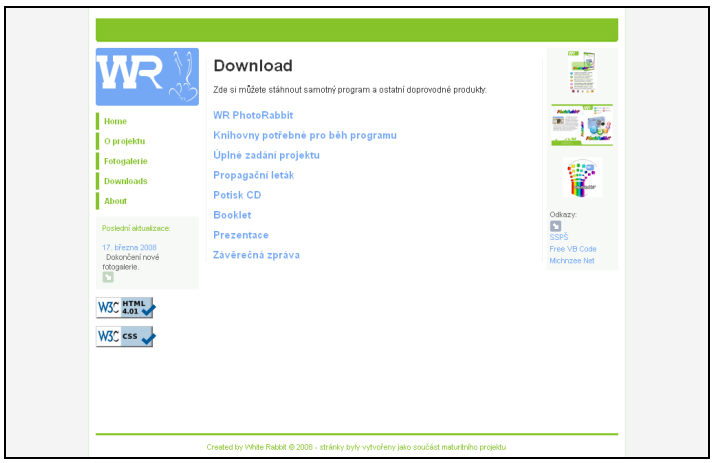
Obr. 26 – Internetová prezentace – O projektu

D.3. Fotogalerie



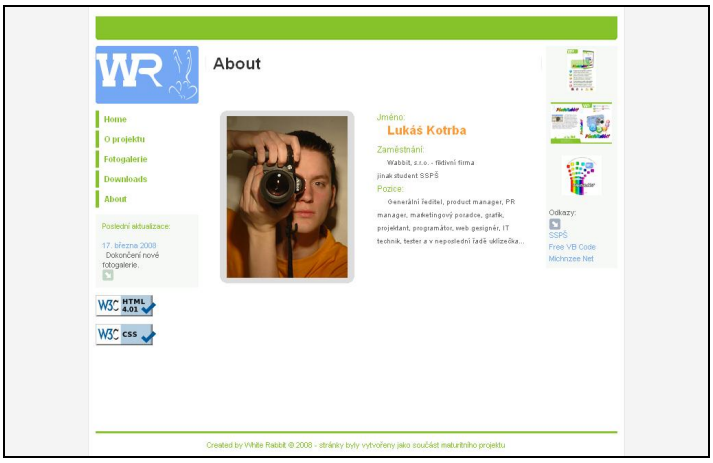
Obr. 27 – Internetová prezentace – Fotogalerie

D.4. Downloads



Obr. 28 – Internetová prezentace – Downloads

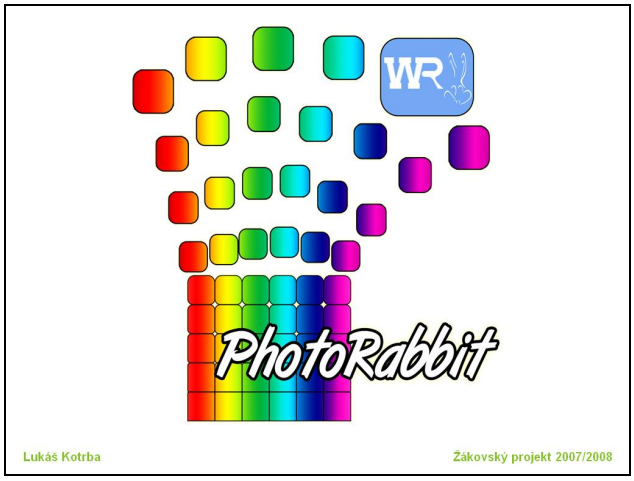
D.5. About



Obr. 29 – Internetová prezentace – About

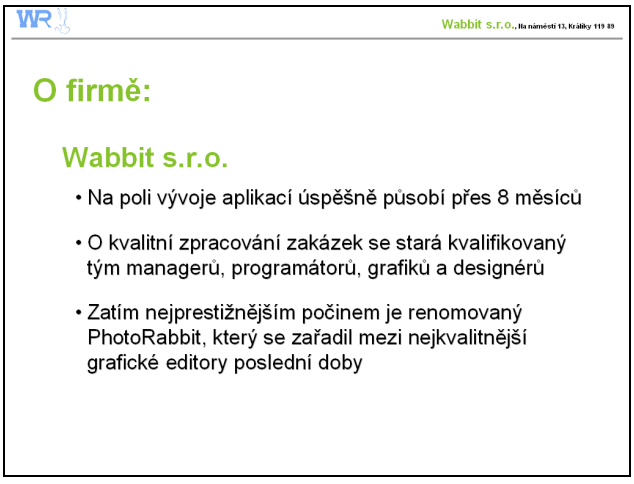
Příloha E – Prezentace MS PowerPoint

E.1. Úvodní obrazovka



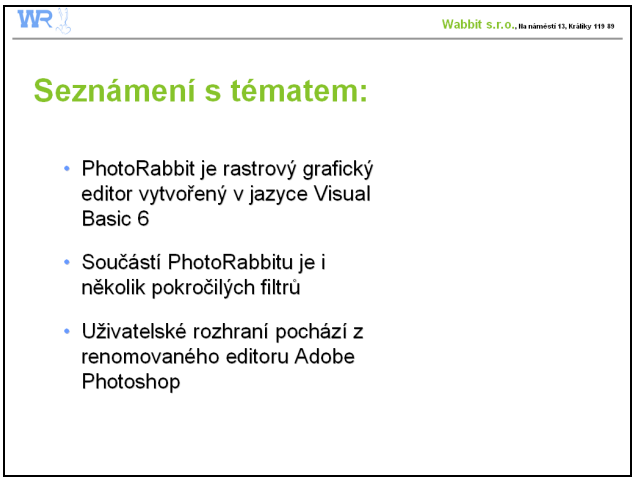
Obr. 30 – Prezentace MS PowerPoint – Úvodní obrazovka

E.2. O firmě



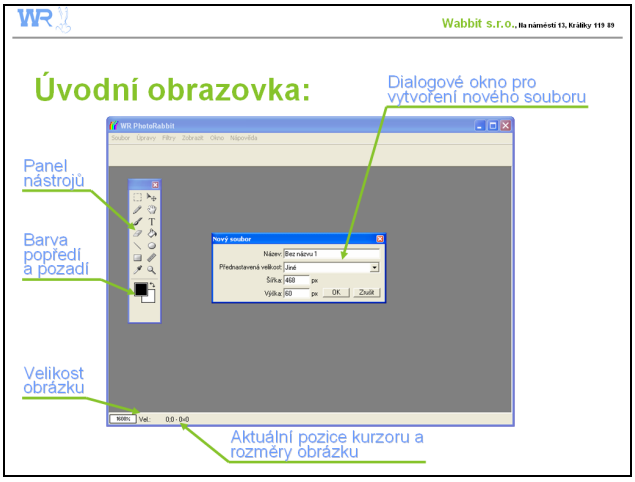
Obr. 31 – Prezentace MS PowerPoint – O firmě

E.3. Seznámení s tématem



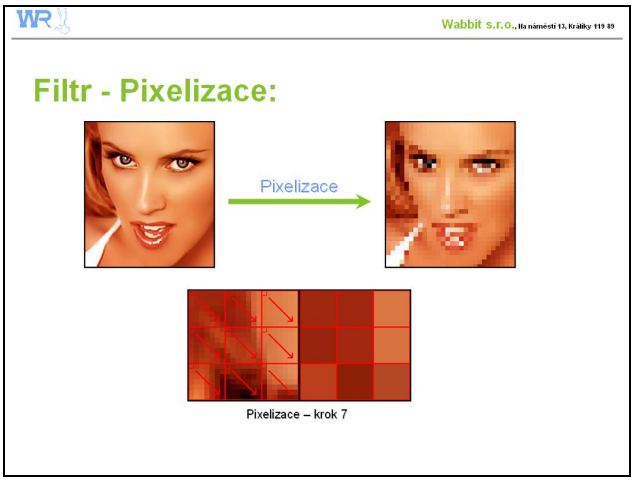
Obr. 32 – Prezentace MS PowerPoint – Seznámení s tématem

E.4. Úvodní obrazovka



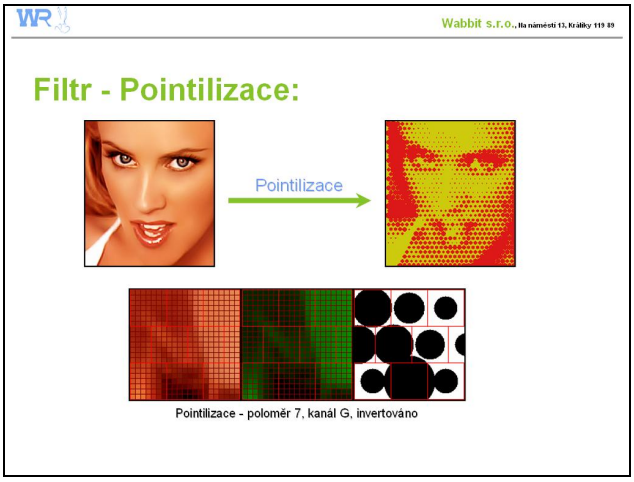
Obr. 33 – Prezentace MS PowerPoint – Úvodní obrazovka

E.5. Filtr – Pixelizace



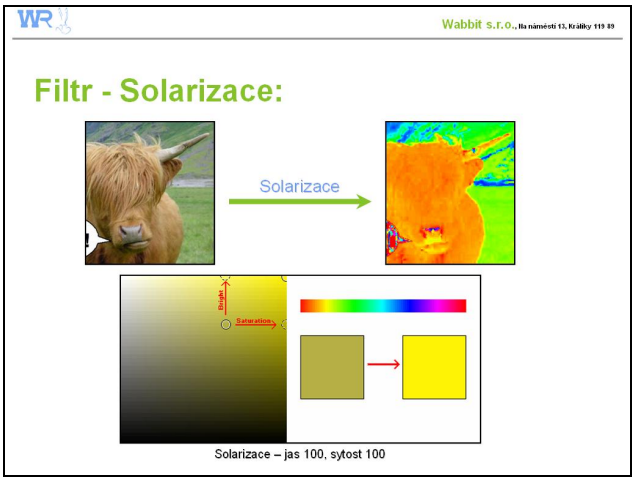
Obr. 34 – Prezentace MS PowerPoint – Filtr – Pixelizace

E.6. Filtr – Pointilizace



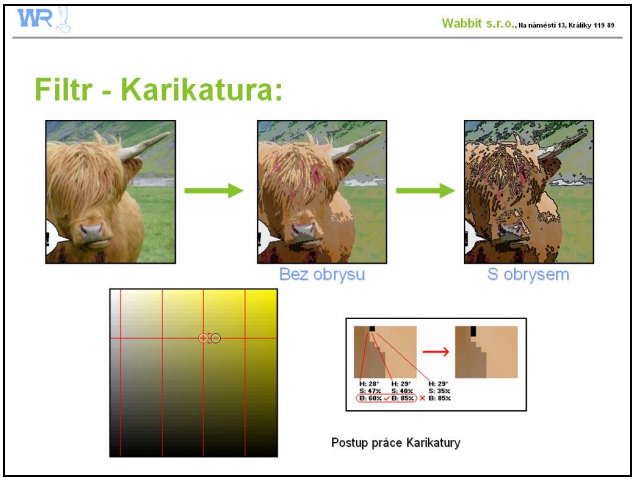
Obr. 35 – Prezentace MS PowerPoint – Filtr – Pointilizace

E.7. Filtr – Solarizace



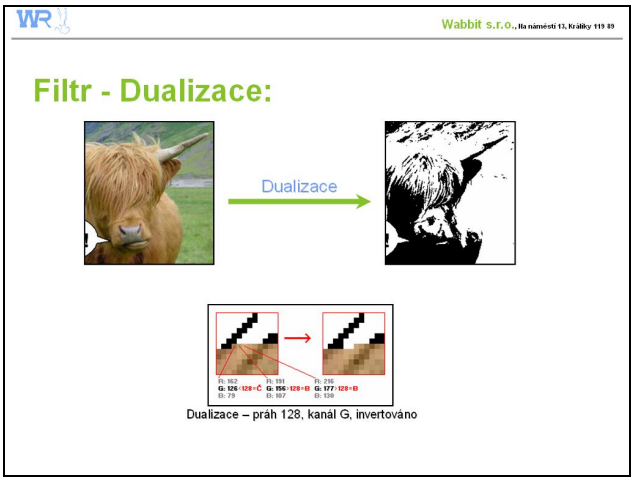
Obr. 36 – Prezentace MS PowerPoint – Filtr – Solarizace

E.8. Filtr – Karikatura



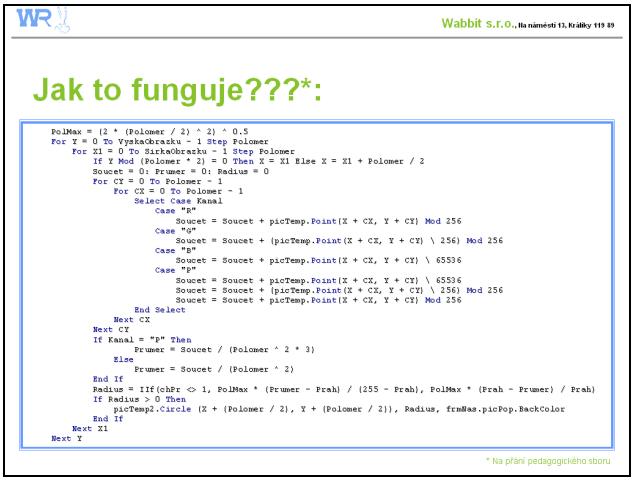
Obr. 37 – Prezentace MS PowerPoint – Filtr – Karikatura

E.9. Filtr – Dualizace



Obr. 38 – Prezentace MS PowerPoint – Filtr – Dualizace

E.10. Jak to funguje???



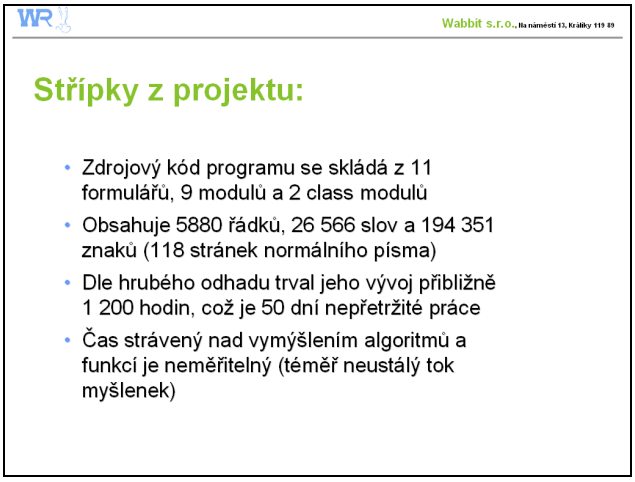
Obr. 39 – Prezentace MS PowerPoint – Jak to funguje???

E.11. Ukázky nástrojů



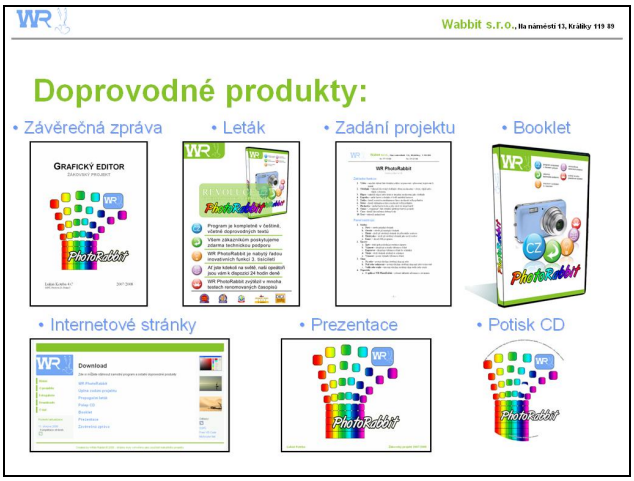
Obr. 40 – Prezentace MS PowerPoint – Ukázky nástrojů

E.12. Střípky z projektu



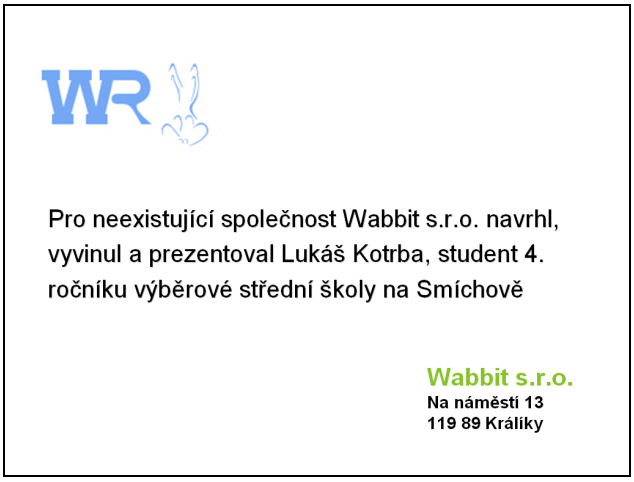
Obr. 41 – Prezentace MS PowerPoint – Střípky z projektu

E.13. Doprovodné produkty



Obr. 42 – Prezentace MS PowerPoint – Doprovodné produkty

E.14. Závěr



Obr. 43 – Prezentace MS PowerPoint – Závěr

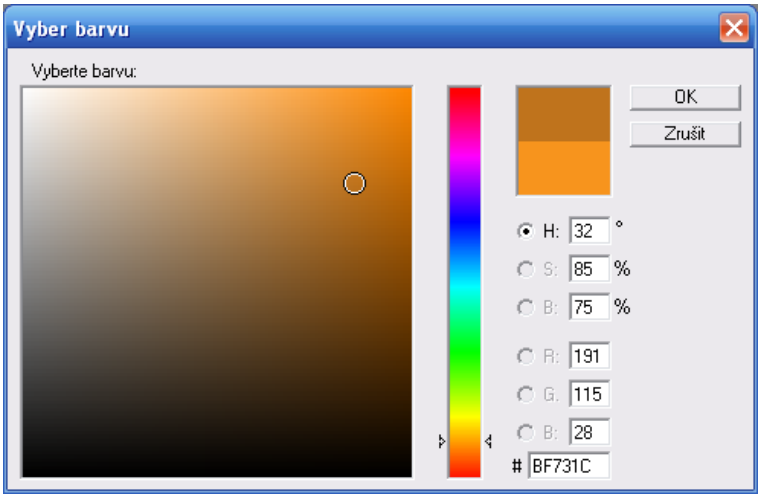
Příloha F – Program

F.1. O aplikaci WR PhotoRabbit – frmAbout



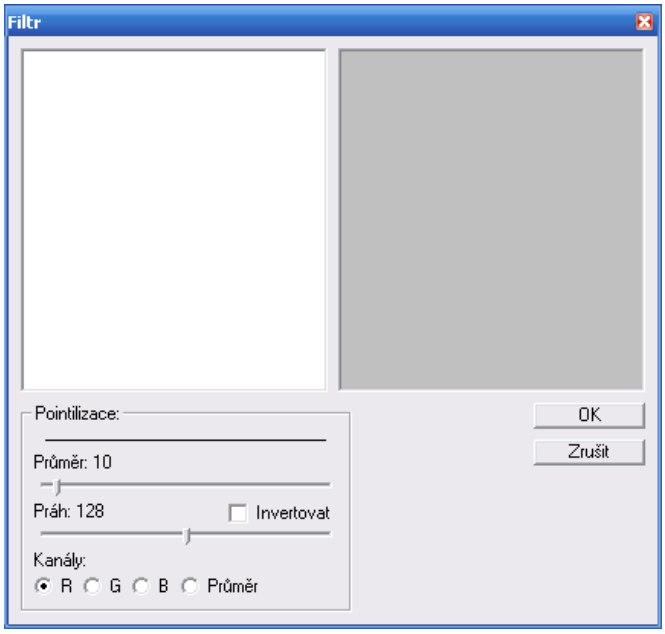
Obr. 44 – Program – O aplikaci WR PhotoRabbit – frmAbout

F.2. Dialogové okno pro výběr barvy – frmBar



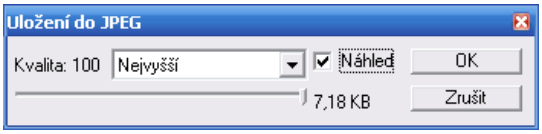
Obr. 45 – Program – Dialogové okno pro výběr barvy – frmBar

F.3. Dialogové okno filtrů – frmFil



Obr. 46 – Program – Dialogové okno filtrů – frmFil

F.4. Dialogové okno uložení do JPEG – frmJPEG



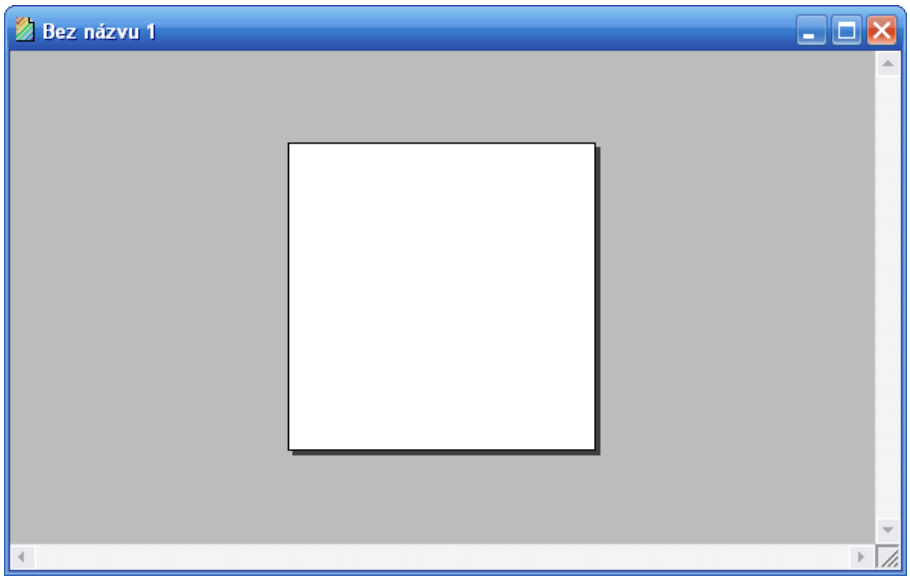
Obr. 47 – Program – Dialogové okno uložení do JPEG – frmJPEG

F.5. Panel nástrojů – frmNas



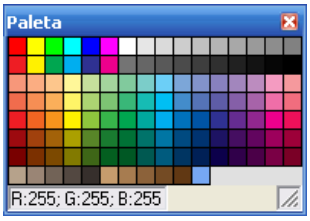
Obr. 48 – Program – Panel nástrojů – frmNas

F.6. Formulář dokumentu – frmOb



Obr. 49 – Program – Formulář dokumentu – frmOb

F.7. Paleta barev – frmPal



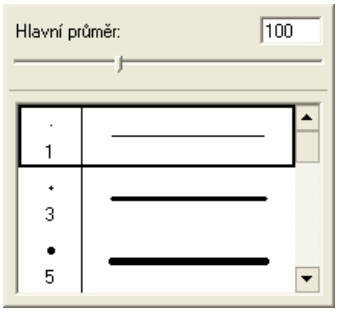
Obr. 50 – Program – Paleta barev – frmPal

F.8. Posuvník – frmPos



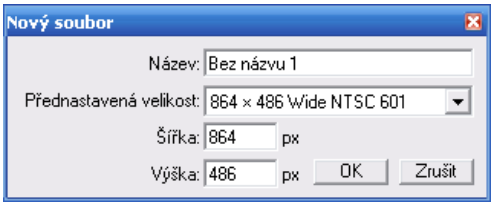
Obr. 51 – Program – Posuvník – frmPos

F.9. Výběr tloušťky stopy – frmSte



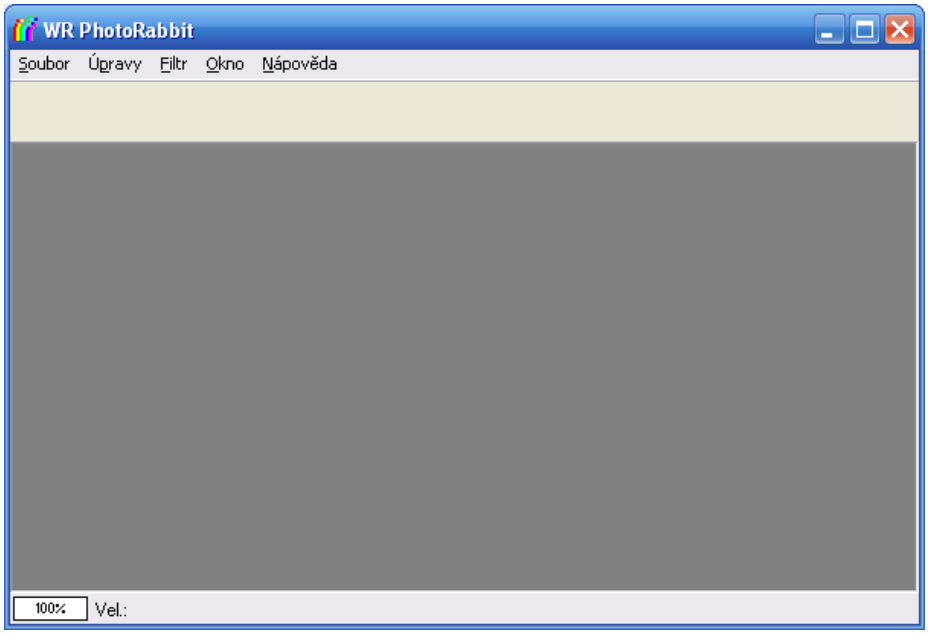
Obr. 52 – Program – Výběr tloušťky stopy – frmSte

F.10. Nový soubor – frmVel



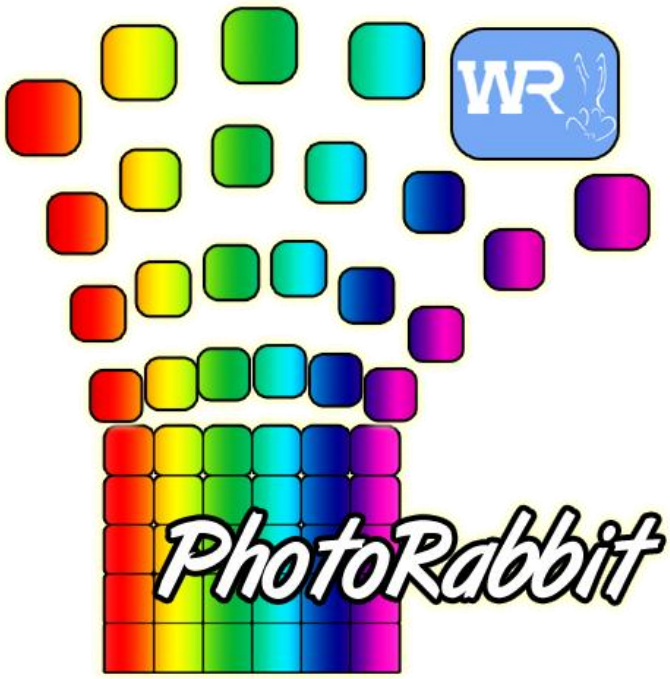
Obr. 53 – Program – Nový soubor – frmVel

F.11. Hlavní formulář – mdiMain



Obr. 54 – Program – Hlavní formulář – mdiMain

F.12. Splash screen – frmSplashScreen



Obr. 55 – Program – Splash screen – frmSplashScreen

Příloha G – Zdrojový kód

G.1. O aplikaci WR PhotoRabbit – frmAbout

```
Option Explicit

Private Sub cmdOK_Click()
    Me.Hide
End Sub

Private Sub Form_Load()
    Me.Caption = "O aplikaci WR PhotoRabbit - " & App.Major & "." &
App.Minor & "." & App.Revision
End Sub
```

G.2. Dialogové okno pro výběr barvy – frmBar

```
Option Explicit

Private Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal X
As Long, ByVal Y As Long) As Long
Private Declare Function GetR Lib "draw.dll" (ByVal nColor As Long) As
Integer
Private Declare Function GetG Lib "draw.dll" (ByVal nColor As Long) As
Integer
Private Declare Function GetB Lib "draw.dll" (ByVal nColor As Long) As
Integer

Dim Hod(4) As String
Dim BarR, BarG, BarB, I As Integer
Dim PrevY
Dim Typ
Dim Aktiv

Private Sub cmdOK_Click()
    Select Case TypBar
        Case 0
            frmNas.picPoz.BackColor = RGB(Int(txR.Text), Int(txG.Text),
Int(txB.Text))
        Case 1
            frmNas.picPop.BackColor = RGB(Int(txR.Text), Int(txG.Text),
Int(txB.Text))
        Case 2
            mdiMain.picObr.BackColor = RGB(Int(txR.Text), Int(txG.Text),
Int(txB.Text))
        Case 3
            frmPal.picPal(PalIndex).BackColor = RGB(Int(txR.Text),
Int(txG.Text), Int(txB.Text))
    End Select
    frmBar.Hide
End Sub

Private Sub cmdZrus_Click()
    Me.Hide
```

```
End Sub

Private Sub imgCur_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then
        Aktiv = 0
        picCtve_MouseMove Button, Shift, imgCur.Left + (X / 15), imgCur.Top
+ (Y / 15)
    End If
End Sub

Private Sub imgCur_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then
        Aktiv = 0
        picCtve_MouseMove Button, Shift, imgCur.Left + (X / 15), imgCur.Top
+ (Y / 15)
    End If
End Sub

Private Sub opTyp_Click(Index As Integer)
    Typ = Index
    Vybarvi
End Sub

Private Sub picCar_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim HSB, Bar() As String

    Aktiv = 0
    If Button = vbLeftButton Then
        Select Case Typ
            Case 0
                If picCar.Top + Y - 2 > picCar.Top + picCar.Height - 7 Then
                    picSip.Top = picCar.Top + picCar.Height - 7
                ElseIf picCar.Top + Y - 2 < picCar.Top - 2 Then
                    picSip.Top = picCar.Top - 2
                Else
                    picSip.Top = picCar.Top + Y - 2
                End If
                BarvaH picSip.Top - 23, 1
                HSB      =      HSBtoRGB(Int(txH.Text),      Int(txS.Text),
Int(txBr.Text), "")
                Bar = Split(HSB, ";")
                picBar.BackColor     =     RGB(Int(Bar(0)),     Int(Bar(1)),
Int(Bar(2)))
                txR.Text = Int(Bar(0)): txG.Text = Int(Bar(1)): txB.Text =
Int(Bar(2))
        End Select
    End If
End Sub

Private Sub picCar_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim HSB, Bar() As String

    Aktiv = 0
    If Button = vbLeftButton Then
        Select Case Typ
```

```
            Case 0
                If picCar.Top + Y - 2 > picCar.Top + picCar.Height - 7 Then
                    picSip.Top = picCar.Top + picCar.Height - 7
                ElseIf picCar.Top + Y - 2 < picCar.Top - 2 Then
                    picSip.Top = picCar.Top - 2
                Else
                    picSip.Top = picCar.Top + Y - 2
                End If
                BarvaH picSip.Top - 23, 0
                HSB      =      HSBtoRGB(Int(txH.Text),      Int(txS.Text),
Int(txBr.Text), "")
                Bar = Split(HSB, ";")
                picBar.BackColor     =     RGB(Int(Bar(0)),     Int(Bar(1)),
Int(Bar(2)))
                txR.Text = Int(Bar(0)): txG.Text = Int(Bar(1)): txB.Text =
Int(Bar(2))
        End Select
    End If
End Sub

Private Sub picCtve_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim Bar() As String, HSB, NaPix

    NaPix = 359 / 255
    Aktiv = 0

    If X < 0 Then X = 0
    If Y < 0 Then Y = 0
    If X > 255 Then X = 255
    If Y > 255 Then Y = 255
    If Button = vbLeftButton Then
        imgCur.Left = X - 8: imgCur.Top = Y - 8
        txS.Text = Int(100 / 255 * X): txBr.Text = Int(100 - (100 / 255 *
Y))
        HSB = HSBtoRGB(Int(txH.Text), Int(txS.Text), Int(txBr.Text), "")
        Bar = Split(HSB, ";")
        picBar.BackColor = RGB(Int(Bar(0)), Int(Bar(1)), Int(Bar(2)))
        txR.Text  =  Int(Bar(0)):  txG.Text  =  Int(Bar(1)):  txB.Text  =
Int(Bar(2))
    End If
End Sub

Private Sub picCtve_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim Bar() As String, HSB, NaPix

    NaPix = 359 / 255

    If X < 0 Then X = 0
    If Y < 0 Then Y = 0
    If X > 255 Then X = 255
    If Y > 255 Then Y = 255
    If Button = vbLeftButton Then
        Aktiv = 0
        imgCur.Left = X - 8: imgCur.Top = Y - 8
        txS.Text = Int(100 / 255 * X): txBr.Text = Int(100 - (100 / 255 *
Y))
        HSB = HSBtoRGB(Int(txH.Text), Int(txS.Text), Int(txBr.Text), "")
```

```
        Bar = Split(HSB, ";")
        picBar.BackColor = RGB(Int(Bar(0)), Int(Bar(1)), Int(Bar(2)))
        txR.Text = Int(Bar(0)): txG.Text = Int(Bar(1)): txB.Text =
Int(Bar(2))
        txHex.Text = PreDec(Int(txR.Text), Int(txG.Text), Int(txB.Text))
    End If
End Sub

Private Sub picSip_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then
        If picSip.Top + Y - PrevY < 23 Then picSip.Top = 23: GoTo Mimo
        If picSip.Top + Y - PrevY > 275 Then picSip.Top = 275: GoTo Mimo
        picSip.Top = picSip.Top + Y - PrevY
Mimo:
        BarvaH picSip.Top - 23, 1
        PrevY = PrevY
        Exit Sub
    End If

    PrevY = Y
End Sub

Private Sub picSip_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then
        Select Case Typ
            Case 0
                BarvaH picSip.Top - 23, 0
        End Select
    End If
End Sub

Private Sub Vybarvi()
    Select Case Typ
        Case 0
            CarH
            BarvaH picSip.Top - 23, 0
        Case 1
            'CarS
    End Select
End Sub

Private Sub CarH()
Dim Bar() As String, HSB, NaPix
    NaPix = 359 / 255

    For I = 0 To 255 Step 3

        HSB = HSBtoRGB(Int(NaPix * I), 100, 100, "")
        Bar = Split(HSB, ";")

        picCar.Line (0, 255 - I)-(20, 255 - I + 3), RGB(Int(Bar(0)),
Int(Bar(1)), Int(Bar(2))), BF
    Next I
End Sub

Private Sub BarvaH(Pozice As Single, Pohyb As Single)
    Dim Vys, Bar() As String, BarY, BarX
```

```
    Dim R, G, B
    If Pohyb = 1 And Pozice Mod 10 <> 0 Then GoTo ZjH
    picCtve.Cls

    For BarY = 0 To 255 Step 6
        For BarX = 0 To 255 Step 6
            Vys = HSBtoRGB(Int((253 - Pozice) * 1.4118), Int(100 / 255 *
BarX), Int(100 - (100 / 255 * BarY)), BarX & ";" & BarY)
            Bar = Split(Vys, ";")
            R = Int(Bar(0))
            G = Int(Bar(1))
            B = Int(Bar(2))
            picCtve.Line (BarX, BarY)-(BarX + 5, BarY + 5), RGB(R, G, B),
BF
        Next BarX
    Next BarY
ZjH:
    txH.Text = Int((253 - Pozice) * 1.4118)
End Sub

Public Sub Nove()
    Dim Hexa

    Aktiv = 0
    Select Case TypBar
        Case 0
            txR.Text = GetR(frmNas.picPoz.BackColor)
            txG.Text = GetG(frmNas.picPoz.BackColor)
            txB.Text = GetB(frmNas.picPoz.BackColor)
            Hexa = Hex(frmNas.picPoz.BackColor)
            picPuv.BackColor = frmNas.picPoz.BackColor
        Case 1
            txR.Text = GetR(frmNas.picPop.BackColor)
            txG.Text = GetG(frmNas.picPop.BackColor)
            txB.Text = GetB(frmNas.picPop.BackColor)
            Hexa = Hex(frmNas.picPop.BackColor)
            picPuv.BackColor = frmNas.picPop.BackColor
        Case 2
            txR.Text = GetR(mdiMain.picObr.BackColor)
            txG.Text = GetG(mdiMain.picObr.BackColor)
            txB.Text = GetB(mdiMain.picObr.BackColor)
            Hexa = Hex(mdiMain.picObr.BackColor)
            picPuv.BackColor = mdiMain.picObr.BackColor
        Case 3
            txR.Text = GetR(frmPal.picPal(PalIndex).BackColor)
            txG.Text = GetG(frmPal.picPal(PalIndex).BackColor)
            txB.Text = GetB(frmPal.picPal(PalIndex).BackColor)
            Hexa = Hex(frmPal.picPal(PalIndex).BackColor)
            picPuv.BackColor = frmPal.picPal(PalIndex).BackColor
    End Select
    If Len(Hexa) <> 6 Then
        Hexa = String(6 - Len(Hexa), "0") & Hexa
    End If
    txHex.Text = Right(Hexa, 2) & Mid(Hexa, 3, 2) & Left(Hexa, 2)
    Vybarvi
    DoHSB
End Sub

Private Sub txB_GotFocus()
```

```
    Aktiv = txB.TabIndex - 10
End Sub

Private Sub txB_Change()
    If txB.Text = "" Then Exit Sub
    If Int(txB.Text) > 255 Then MsgBox "Hodnota musí být mezi 0 a 255!",
vbCritical + vbOKOnly, "Chyba": Exit Sub
    picBar.BackColor = RGB(Val(txR.Text), Val(txG.Text), Val(txB.Text))
    If Aktiv = txB.TabIndex - 10 Then
        txHex.Text = PreDec(Val(txR.Text), Val(txG.Text), Val(txB.Text))
        DoHSB
    End If
End Sub

Private Sub txB_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub

Private Sub txB_LostFocus()
    If txB.Text = "" Then
        MsgBox "Textové pole je prázdné!" & Chr(13) & "Hodnota musí být
mezi 0 a 255!", vbCritical + vbOKOnly, "Chyba"
        txB.SetFocus
    End If
End Sub

Private Sub txBr_GotFocus()
    Aktiv = txBr.TabIndex - 10
End Sub

Private Sub txBr_Change()
Dim HSB, Bar() As String

    If txBr.Text = "" Then Exit Sub
    If Len(txBr.Text) = 3 And Val(txBr.Text) > 100 Then MsgBox "Hodnota
musí být mezi 0 a 100%!", vbCritical + vbOKOnly, "Chyba": Exit Sub
    If Aktiv = txBr.TabIndex - 10 Then
        HSB = HSBtoRGB(Int(txH.Text), Int(txS.Text), Int(txBr.Text), "")
        Bar = Split(HSB, ";")
        picBar.BackColor = RGB(Int(Bar(0)), Int(Bar(1)), Int(Bar(2)))
        txR.Text = Int(Bar(0)): txG.Text = Int(Bar(1)): txB.Text =
Int(Bar(2))
        picSip.Top = 255 - (255 / 360 * Int(txH.Text)) + 21
        imgCur.Left = (2.55 * Int(txS.Text)) - 7: imgCur.Top = 255 - (2.55
* Int(txBr.Text)) - 7
        txHex.Text = PreDec(Val(txR.Text), Val(txG.Text), Val(txB.Text))
    End If
End Sub

Private Sub txBr_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub

Private Sub txBr_LostFocus()
    If txBr.Text = "" Then
        MsgBox "Textové pole je prázdné!" & Chr(13) & "Hodnota musí být
mezi 0 a 100%!", vbCritical + vbOKOnly, "Chyba"
```

```
        txBr.SetFocus
    End If
End Sub

Private Sub txG_GotFocus()
    Aktiv = txG.TabIndex - 10
End Sub

Private Sub txG_Change()
    If txG.Text = "" Then Exit Sub
    If Len(txG.Text) = 3 And Val(txG.Text) > 255 Then MsgBox "Hodnota musí
být mezi 0 a 255!", vbCritical + vbOKOnly, "Chyba": Exit Sub
    picBar.BackColor = RGB(Val(txR.Text), Val(txG.Text), Val(txB.Text))
    If Aktiv = txG.TabIndex - 10 Then
        txHex.Text = PreDec(Val(txR.Text), Val(txG.Text), Val(txB.Text))
        DoHSB
    End If
End Sub

Private Sub txG_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub

Private Sub txG_LostFocus()
    If txG.Text = "" Then
        MsgBox "Textové pole je prázdné!" & Chr(13) & "Hodnota musí být
mezi 0 a 255!", vbCritical + vbOKOnly, "Chyba"
        txG.SetFocus
    End If
End Sub

Private Sub txH_GotFocus()
    Aktiv = txH.TabIndex - 10
End Sub

Private Sub txH_Change()
Dim HSB, Bar() As String

    If txH.Text = "" Then Exit Sub
    If Len(txH.Text) = 3 And Val(txH.Text) > 360 Then MsgBox "Hodnota musí
být mezi 0 a 360°!", vbCritical + vbOKOnly, "Chyba": Exit Sub
    If Aktiv = txH.TabIndex - 10 Then
        HSB = HSBtoRGB(Int(txH.Text), Int(txS.Text), Int(txBr.Text), "")
        Bar = Split(HSB, ";")
        picBar.BackColor = RGB(Int(Bar(0)), Int(Bar(1)), Int(Bar(2)))
        txR.Text  =  Int(Bar(0)):  txG.Text  =  Int(Bar(1)):  txB.Text  =
Int(Bar(2))
        picSip.Top = 255 - (255 / 360 * Int(txH.Text)) + 21
        BarvaH picSip.Top - 23, 0
        imgCur.Left = (2.55 * Int(txS.Text)) - 7: imgCur.Top = 255 - (2.55
* Int(txBr.Text)) - 7
        txHex.Text = PreDec(Val(txR.Text), Val(txG.Text), Val(txB.Text))
    End If
End Sub

Private Sub txH_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
```

```
End Sub

Private Sub txH_LostFocus()
    If txH.Text = "" Then
        MsgBox "Textové pole je prázdné!" & Chr(13) & "Hodnota musí být
mezi 0 a 255!", vbCritical + vbOKOnly, "Chyba"
        txH.SetFocus
    End If
End Sub

Private Sub txHex_GotFocus()
    Aktiv = txHex.TabIndex - 10
End Sub

Private Sub txHex_Change()
    Dim Puv As String
    Dim Hexa As String
    Dim PolRGB() As String

    If txHex.Text = "" Then Exit Sub
    If Aktiv = txHex.TabIndex - 10 Then
        Hexa = txHex.Text
        Puv = txHex.Text

        If Len(Hexa) < 6 Then
            Hexa = String(6 - Len(Hexa), "0") & Hexa
        End If

        Hexa = PreHex(Hexa)
        PolRGB = Split(Hexa, ",")
        txR.Text = PolRGB(0)
        txG.Text = PolRGB(1)
        txB.Text = PolRGB(2)
        picBar.BackColor = RGB(Int(txR.Text), Int(txG.Text), Int(txB.Text))
        txHex.Text = Puv
        DoHSB
    End If
End Sub

Private Sub txHex_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii < 71 And KeyAscii > 64 Then Exit Sub
    If KeyAscii < 103 And KeyAscii > 96 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub

Private Sub txHex_LostFocus()
    If txHex.Text = "" Then
        MsgBox "Textové pole je prázdné!" & Chr(13) & "Hodnota musí být
mezi 0 a 255!", vbCritical + vbOKOnly, "Chyba"
        txHex.SetFocus
    End If
End Sub

Private Sub txR_GotFocus()
    Aktiv = txR.TabIndex - 10
End Sub

Private Sub txR_Change()
```

```
    If txR.Text = "" Then Exit Sub
    If Len(txR.Text) = 3 And Val(txR.Text) > 255 Then MsgBox "Hodnota musí
být mezi 0 a 255!", vbCritical + vbOKOnly, "Chyba": Exit Sub
    picBar.BackColor = RGB(Val(txR.Text), Val(txG.Text), Val(txB.Text))
    If Aktiv = txR.TabIndex - 10 Then
        txHex.Text = PreDec(Val(txR.Text), Val(txG.Text), Val(txB.Text))
        DoHSB
    End If
End Sub

Private Sub txR_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub

Private Sub txR_LostFocus()
    If txR.Text = "" Then
        MsgBox "Textové pole je prázdné!" & Chr(13) & "Hodnota musí být
mezi 0 a 255!", vbCritical + vbOKOnly, "Chyba"
        txR.SetFocus
    End If
End Sub

Private Sub txS_GotFocus()
    Aktiv = txS.TabIndex - 10
End Sub

Private Sub txS_Change()
Dim HSB, Bar() As String

    If txS.Text = "" Then Exit Sub
    If Len(txS.Text) = 3 And Val(txS.Text) > 100 Then MsgBox "Hodnota musí
být mezi 0 a 100%!", vbCritical + vbOKOnly, "Chyba": Exit Sub
    If Aktiv = txS.TabIndex - 10 Then
        HSB = HSBtoRGB(Int(txH.Text), Int(txS.Text), Int(txBr.Text), "")
        Bar = Split(HSB, ";")
        picBar.BackColor = RGB(Int(Bar(0)), Int(Bar(1)), Int(Bar(2)))
        txR.Text  =  Int(Bar(0)):  txG.Text  =  Int(Bar(1)):  txB.Text  =
Int(Bar(2))
        picSip.Top = 255 - (255 / 360 * Int(txH.Text)) + 21
        imgCur.Left = (2.55 * Int(txS.Text)) - 7: imgCur.Top = 255 - (2.55
* Int(txBr.Text)) - 7
        txHex.Text = PreDec(Val(txR.Text), Val(txG.Text), Val(txB.Text))
    End If
End Sub

Private Sub txS_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub

Private Sub DoHSB()
Dim HSB, Bar() As String

    HSB = RGBtoHSB(Val(txR.Text), Val(txG.Text), Val(txB.Text))
    Bar = Split(HSB, ";")
    txH.Text = Int(Bar(0))
    picSip.Top = 255 - (Int(255 / 360 * Bar(0))) + 21
    txS.Text = Int(Bar(1))
```

```
    txBr.Text = Int(Bar(2))
    imgCur.Left = (2.55 * Int(Bar(1))) - 7: imgCur.Top = 255 - (2.55 *
Int(Bar(2))) - 7
    Vybarvi
End Sub

Private Sub txS_LostFocus()
    If txS.Text = "" Then
        MsgBox "Textové pole je prázdné!" & Chr(13) & "Hodnota musí být
mezi 0 a 255!", vbCritical + vbOKOnly, "Chyba"
        txS.SetFocus
    End If
End Sub
```

G.3. Dialogové okno filtrů – frmFil

```
Option Explicit

Dim ID, Typ, Vyber As Boolean, Vbehu As Boolean
Dim HoldBitmap As Long, HoldDC As Long, bRegion As Boolean
Dim ObX, ObY, MX, MY
Dim SolKanal, DuaKanal

Private Sub Pointilizace()
Dim StartX, StartY, X, X1, Y, CX, CY, Sou, Pru, Pol, PolMax

    If Vbehu = True Then Exit Sub
    lnPoiSt.Visible = True: Vbehu = True

    lbPoiPol.Caption = "Poloměr: " & slPoiPol.Value
    picTemp2.FillColor = frmNas.picPop.BackColor

    If picRes.Left < 0 Then StartX = -picRes.Left \ slPoiPol.Value Else
StartX = "M"
    If picRes.Top < 0 Then StartY = -picRes.Top \ slPoiPol.Value Else
StartY = "M"

    picDest.Cls: picTemp.Cls

    If picRes.Picture = 0 Then Exit Sub

    If StartX = "M" And StartY = "M" Then
            picTemp.Width = picRes.ScaleWidth + 2: picTemp.Height =
picRes.ScaleHeight + 2
            picTemp2.Width = picRes.ScaleWidth + 2: picTemp2.Height =
picRes.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , 0, 0,
picRes.ScaleWidth, picRes.ScaleHeight
        ElseIf StartX = "M" Then
            picTemp.Width = picRes.ScaleWidth + 2: picTemp.Height =
picResRam.ScaleHeight + 2
            picTemp2.Width = picRes.ScaleWidth + 2: picTemp2.Height =
picResRam.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , 0, StartY *
slPoiPol.Value, picRes.ScaleWidth, picResRam.ScaleHeight
        ElseIf StartY = "M" Then
```

```
            picTemp.Width = picResRam.ScaleWidth + 2: picTemp.Height =
picRes.ScaleHeight + 2
            picTemp2.Width = picResRam.ScaleWidth + 2: picTemp2.Height =
picRes.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , StartX *
slPoiPol.Value, 0, picResRam.ScaleWidth, picRes.ScaleHeight
        Else
            picTemp.Width = picResRam.ScaleWidth + 2: picTemp.Height =
picResRam.ScaleHeight + 2
            picTemp2.Width = picResRam.ScaleWidth + 2: picTemp2.Height =
picResRam.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , StartX *
slPoiPol.Value,     StartY     *     slPoiPol,     picResRam.ScaleWidth,
picResRam.ScaleHeight
    End If
    picTemp2.Picture   =   picTemp.Picture:   picTemp2.Line   (0,   0)-
(picTemp2.ScaleWidth, picTemp2.ScaleHeight), frmNas.picPoz.BackColor, BF

    PolMax = (2 * (slPoiPol.Value / 2) ^ 2) ^ 0.5
    For Y = 0 To picTemp.ScaleHeight - 1 Step slPoiPol.Value
        For X1 = 0 To picTemp.ScaleWidth - 1 Step slPoiPol.Value
            If Y Mod (slPoiPol.Value * 2) = 0 Then X = X1 Else X = X1 +
slPoiPol.Value / 2
            Sou = 0: Pru = 0: Pol = 0
            For CY = 0 To slPoiPol.Value - 1
                For CX = 0 To slPoiPol.Value - 1
                    Select Case SolKanal
                        Case "R"
                            Sou = Sou + picTemp.Point(X + CX, Y + CY) Mod
256
                        Case "G"
                            Sou = Sou + (picTemp.Point(X + CX, Y + CY) \
256) Mod 256
                        Case "B"
                            Sou = Sou + picTemp.Point(X + CX, Y + CY) \
65536
                        Case "P"
                            Sou = Sou + picTemp.Point(X + CX, Y + CY) \
65536
                            Sou = Sou + (picTemp.Point(X + CX, Y + CY) \
256) Mod 256
                            Sou = Sou + picTemp.Point(X + CX, Y + CY) Mod
256
                    End Select
                Next CX
            Next CY
            If SolKanal = "P" Then
                    Pru = Sou / (slPoiPol.Value ^ 2 * 3)
                Else
                    Pru = Sou / (slPoiPol.Value ^ 2)
            End If
            Pol = IIf(chPr <> 1, PolMax * (Pru - slPoiPra.Value) / (255 -
slPoiPra.Value), PolMax * (slPoiPra.Value - Pru) / slPoiPra.Value)
            If Pol > 0 Then
                picTemp2.Circle   (X   +   (slPoiPol.Value   /   2),   Y   +
(slPoiPol.Value / 2)), Pol, frmNas.picPop.BackColor
            End If
        Next X1
```

```
        If Y Mod (5 * slPoiPol.Value) = 0 Then DoEvents: lnPoiSt.X2 =
lnPoiSt.X1 + Y * 2760 / (picTemp2.ScaleHeight - 1)
    Next Y
    picTemp2.Picture = picTemp2.image

    picDest.PaintPicture picTemp2.Picture, picDest.ScaleWidth / 2 -
picTemp2.ScaleWidth / 2, picDest.ScaleHeight / 2 - picTemp2.ScaleHeight / 2

    lnPoiSt.Visible = False: Vbehu = False
End Sub


Private Sub cmdOK_Click()
Dim X, Y

    Select Case Typ
        Case "pix"
            Document(ID).Pixelizace slPix.Value, Vyber
        Case "poi"
            Document(ID).Pointilizace    slPoiPol.Value,    slPoiPra.Value,
SolKanal, chPr.Value, Vyber
        Case "sol"
            Document(ID).Solarizace slSolS.Value, slSolB.Value, Vyber
        Case "dua"
            Document(ID).Dualizace  slDuaPr.Value,  DuaKanal,  chDua.Value,
Vyber
    End Select
    Me.Hide
End Sub

Private Sub cmdZru_Click()
    Me.Hide
End Sub

Private Sub Form_Load()
    SolKanal = "R"
    DuaKanal = "R"
End Sub

Public Sub ShowMe(IID, Druh, Vyb As Boolean)
    ID = IID
    Typ = Druh
    Vyber = Vyb
    picRes.Picture  =  IIf(Vyber  <>  True,  Document(ID).picOb.Picture,
Document(ID).picTemp.Picture)
    ObX  =  picTemp.ScaleWidth  /  2  +  picRes.ScaleWidth  /  2:  ObY  =
picTemp.ScaleHeight / 2 + picRes.ScaleHeight / 2
    ObX = picRes.Width / 2 - picResRam.ScaleWidth / 2: ObY = picRes.Height
/ 2 - picResRam.ScaleHeight / 2
    picRes.Left = -ObX: picRes.Top = -ObY
    Hidni
    Select Case Typ
        Case "pix"
            frPix.Visible = True
        Case "poi"
            frPoi.Visible = True
        Case "sol"
            frSol.Visible = True
        Case "dua"
```

```
            frDua.Visible = True
    End Select
    Me.Show 1, mdiMain
End Sub

Private Sub chDua_Click()
    Dualizace
End Sub

Private Sub chPr_Click()
    Pointilizace
End Sub

Private Sub opDuaB_Click()
    DuaKanal = "B"
    Dualizace
End Sub

Private Sub opDuaG_Click()
    DuaKanal = "G"
    Dualizace
End Sub

Private Sub opDuaP_Click()
    DuaKanal = "P"
    Dualizace
End Sub

Private Sub opDuaR_Click()
    DuaKanal = "R"
    Dualizace
End Sub

Private Sub opPoiB_Click()
    SolKanal = "B"
    Pointilizace
End Sub

Private Sub opPoiG_Click()
    SolKanal = "G"
    Pointilizace
End Sub

Private Sub opPoiPr_Click()
    SolKanal = "P"
    Pointilizace
End Sub

Private Sub opPoiR_Click()
    SolKanal = "R"
    Pointilizace
End Sub

Private Sub picRes_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then MX = X: MY = Y
End Sub
```

```
Private Sub picRes_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then
        If picResRam.ScaleWidth < picRes.Width Then
            If (picRes.Left + X - MX) > 0 Or picRes.Width + (picRes.Left +
X - MX) < picResRam.ScaleWidth Then GoTo Dal
            picRes.Left = picRes.Left + X - MX
        End If
Dal:
        If picResRam.ScaleHeight < picRes.Height Then
            If (picRes.Top + Y - MY) > 0 Or picRes.Height + (picRes.Top + Y
- MY) < picResRam.ScaleHeight Then Exit Sub
            picRes.Top = picRes.Top + Y - MY
        End If
    End If
End Sub

Private Sub picRes_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then
        Select Case Typ
            Case "pix"
                slPix_Change
            Case "poi"
                Pointilizace
            Case "sol"
                Solarizace
            Case "dua"
                Dualizace
        End Select
    End If
End Sub

Private Sub slDuaPr_Change()
    Dualizace
End Sub

Private Sub slDuaPr_Scroll()
    lbDuaPr.Caption = "Práh: " & slDuaPr.Value
End Sub

Private Sub slPix_Change()
Dim StartX, StartY, X, Y

    If Vbehu = True Then Exit Sub
    Vbehu = True
    frPix.Caption = "Pixelizace: " & slPix.Value

    If picRes.Left < 0 Then StartX = -picRes.Left \ slPix.Value Else StartX
= "M"
    If picRes.Top < 0 Then StartY = -picRes.Top \ slPix.Value Else StartY =
"M"

    picDest.Cls: picTemp.Cls

    If picRes.Picture = 0 Then Exit Sub

    If StartX = "M" And StartY = "M" Then
```

```
            picTemp.Width = picRes.ScaleWidth + 2: picTemp.Height =
picRes.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , 0, 0,
picRes.ScaleWidth, picRes.ScaleHeight
        ElseIf StartX = "M" Then
            picTemp.Width = picRes.ScaleWidth + 2: picTemp.Height =
picResRam.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , 0, StartY *
slPix.Value, picRes.ScaleWidth, picResRam.ScaleHeight
        ElseIf StartY = "M" Then
            picTemp.Width = picResRam.ScaleWidth + 2: picTemp.Height =
picRes.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , StartX *
slPix.Value, 0, picResRam.ScaleWidth, picRes.ScaleHeight
        Else
            picTemp.Width = picResRam.ScaleWidth + 2: picTemp.Height =
picResRam.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , StartX *
slPix.Value, StartY * slPix, picResRam.ScaleWidth, picResRam.ScaleHeight
    End If

    lnPixSt.Visible = True
    For Y = 0 To picTemp.ScaleHeight - 1 Step slPix.Value
        For X = 0 To picTemp.ScaleWidth - 1 Step slPix.Value
            picTemp.Line (X, Y)-(X + slPix.Value - 1, Y + slPix.Value - 1),
picTemp.Point(X, Y), BF
        Next X
        If Y Mod (5 * slPix.Value) = 0 Then DoEvents: lnPixSt.X2 =
lnPixSt.X1 + 2760 / (picTemp.ScaleHeight - 1) * Y
    Next Y
    picTemp.Picture = picTemp.image

    picDest.PaintPicture picTemp.Picture, picDest.ScaleWidth / 2 -
picTemp.ScaleWidth / 2, picDest.ScaleHeight / 2 - picTemp.ScaleHeight / 2

    lnPixSt.Visible = False: Vbehu = False
End Sub

Private Sub Hidni()
    frPix.Visible = False
    frPoi.Visible = False
    frSol.Visible = False
    frDua.Visible = False
End Sub

Private Sub slPoiPol_Change()
    Pointilizace
End Sub


Private Sub slPoiPol_Scroll()
    lbPoiPol.Caption = "Průměr: " & slPoiPol.Value
End Sub

Private Sub slPoiPra_Change()
    Pointilizace
End Sub

Private Sub slPoiPra_Scroll()
```

```
    lbPoiPra.Caption = "Práh: " & slPoiPra.Value
End Sub

Private Sub slSolB_Change()
    Solarizace
End Sub

Private Sub slSolB_Scroll()
    lbSolB.Caption = "Jas: " & slSolB.Value
End Sub

Private Sub slSolS_Change()
    Solarizace
End Sub

Private Sub slSolS_Scroll()
    lbSolS.Caption = "Sytost: " & slSolS.Value
End Sub

Private Sub Solarizace()
Dim StartX, StartY, X, Y, SolRGB, R As Integer, G As Integer, B As Integer,
HSB, Bar() As String

    If Vbehu = True Then Exit Sub
    lnSolSt.Visible = True: Vbehu = True

    If picRes.Left < 0 Then StartX = -picRes.Left \ slPix.Value Else StartX
= "M"
    If picRes.Top < 0 Then StartY = -picRes.Top \ slPix.Value Else StartY =
"M"

    picDest.Cls: picTemp.Cls

    If picRes.Picture = 0 Then Exit Sub

    If StartX = "M" And StartY = "M" Then
            picTemp.Width  =  picRes.ScaleWidth  +  2:  picTemp.Height  =
picRes.ScaleHeight + 2
            picTemp.PaintPicture  picRes.Picture,  0,  0,  ,  ,  0,  0,
picRes.ScaleWidth, picRes.ScaleHeight
        ElseIf StartX = "M" Then
            picTemp.Width  =  picRes.ScaleWidth  +  2:  picTemp.Height  =
picResRam.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , 0, StartY *
slPix.Value, picRes.ScaleWidth, picResRam.ScaleHeight
        ElseIf StartY = "M" Then
            picTemp.Width  =  picResRam.ScaleWidth  +  2:  picTemp.Height  =
picRes.ScaleHeight + 2
            picTemp.PaintPicture  picRes.Picture,  0,  0,  ,  ,  StartX  *
slPix.Value, 0, picResRam.ScaleWidth, picRes.ScaleHeight
        Else
            picTemp.Width  =  picResRam.ScaleWidth  +  2:  picTemp.Height  =
picResRam.ScaleHeight + 2
            picTemp.PaintPicture  picRes.Picture,  0,  0,  ,  ,  StartX  *
slPix.Value, StartY * slPix, picResRam.ScaleWidth, picResRam.ScaleHeight
    End If

    For Y = 0 To picTemp.ScaleHeight - 1
        For X = 0 To picTemp.ScaleWidth - 1
```

```
            SolRGB = picTemp.Point(X, Y)
            R = SolRGB Mod 256
            G = (SolRGB \ 256) Mod 256
            B = SolRGB \ 65536
            HSB = RGBtoHSB(R, G, B)
            Bar = Split(HSB, ";")
            Bar(1) = Int(slSolS.Value): Bar(2) = Int(slSolB.Value)
            SolRGB = HSBtoRGB(Int(Bar(0)), Int(Bar(1)), Int(Bar(2)), "")
            Bar = Split(SolRGB, ";")
            picTemp.ForeColor = RGB(Int(Bar(0)), Int(Bar(1)), Int(Bar(2)))
            picTemp.PSet (X, Y)
        Next X
        If Y Mod 10 = 0 Then DoEvents: lnSolSt.X2 = lnSolSt.X1 + 2760 /
(picTemp.ScaleHeight - 1) * Y
    Next Y
    picTemp.Picture = picTemp.image

    picDest.PaintPicture picTemp.Picture, picDest.ScaleWidth / 2 -
picTemp.ScaleWidth / 2, picDest.ScaleHeight / 2 - picTemp.ScaleHeight / 2

    lnSolSt.Visible = False: Vbehu = False
End Sub

Private Sub Dualizace()
Dim StartX, StartY, X, Y, DuaSou

    If Vbehu = True Then Exit Sub
    lnDuaSt.Visible = True: Vbehu = True

    If picRes.Left < 0 Then StartX = -picRes.Left \ slPix.Value Else StartX
= "M"
    If picRes.Top < 0 Then StartY = -picRes.Top \ slPix.Value Else StartY =
"M"

    picDest.Cls: picTemp.Cls

    If picRes.Picture = 0 Then Exit Sub

    If StartX = "M" And StartY = "M" Then
            picTemp.Width = picRes.ScaleWidth + 2: picTemp.Height =
picRes.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , 0, 0,
picRes.ScaleWidth, picRes.ScaleHeight
        ElseIf StartX = "M" Then
            picTemp.Width = picRes.ScaleWidth + 2: picTemp.Height =
picResRam.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , 0, StartY *
slPix.Value, picRes.ScaleWidth, picResRam.ScaleHeight
        ElseIf StartY = "M" Then
            picTemp.Width = picResRam.ScaleWidth + 2: picTemp.Height =
picRes.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , StartX *
slPix.Value, 0, picResRam.ScaleWidth, picRes.ScaleHeight
        Else
            picTemp.Width = picResRam.ScaleWidth + 2: picTemp.Height =
picResRam.ScaleHeight + 2
            picTemp.PaintPicture picRes.Picture, 0, 0, , , StartX *
slPix.Value, StartY * slPix, picResRam.ScaleWidth, picResRam.ScaleHeight
    End If
```

```
    For Y = 0 To picTemp.ScaleHeight - 1
        For X = 0 To picTemp.ScaleWidth - 1
            Select Case DuaKanal
                Case "R"
                    DuaSou = picTemp.Point(X, Y) Mod 256
                Case "G"
                    DuaSou = (picTemp.Point(X, Y) \ 256) Mod 256
                Case "B"
                    DuaSou = picTemp.Point(X, Y) \ 65536
                Case "P"
                    DuaSou = picTemp.Point(X, Y) Mod 256
                    DuaSou = DuaSou + (picTemp.Point(X, Y) \ 256) Mod 256
                    DuaSou = DuaSou + picTemp.Point(X, Y) \ 65536
                    DuaSou = DuaSou \ 3
            End Select
            If chDua.Value = 0 Then
                    If DuaSou > slDuaPr.Value Then
                            picTemp.ForeColor = frmNas.picPop.BackColor
                            picTemp.PSet (X, Y)
                        Else
                            picTemp.ForeColor = frmNas.picPoz.BackColor
                            picTemp.PSet (X, Y)
                    End If
                Else
                    If DuaSou < slDuaPr.Value Then
                            picTemp.ForeColor = frmNas.picPop.BackColor
                            picTemp.PSet (X, Y)
                        Else
                            picTemp.ForeColor = frmNas.picPoz.BackColor
                            picTemp.PSet (X, Y)
                    End If
            End If
        Next X
        If Y Mod 10 = 0 Then DoEvents: lnDuaSt.X2 = lnDuaSt.X1 + 2760 /
(picTemp.ScaleHeight - 1) * Y
    Next Y
    picTemp.Picture = picTemp.image

    picDest.PaintPicture  picTemp.Picture,  picDest.ScaleWidth  /  2  -
picTemp.ScaleWidth / 2, picDest.ScaleHeight / 2 - picTemp.ScaleHeight / 2

    lnDuaSt.Visible = False: Vbehu = False
End Sub
```

G.4. Dialogové okno uložení do JPEG – frmJPEG

```
Option Explicit

Dim cDIB As New cDIBSection
Dim Cesta, Index

Private Sub cbKval_Click()
    slKval.Value = cbKval.ItemData(cbKval.ListIndex)
    lbKval.Caption = "Kvalita: " & slKval.Value
End Sub
```

```
Private Sub cmdOK_Click()
    If Not SaveJPG(cDIB, Cesta, slKval.Value) Then
        MsgBox "Chyba při ukládání obrázku do: '" & Cesta & "'",
vbExclamation, "Chyba"
    End If
    If Document(Index).picJPG.Visible = True Then
        Document(Index).picJPG.Visible = False
        Kill App.Path & "\data\temp.jpg"
    End If
    Me.Hide
End Sub

Private Sub cmdZru_Click()
    If Document(Index).picJPG.Visible = True Then
        Document(Index).picJPG.Visible = False
        Kill App.Path & "\data\temp.jpg"
    End If
    Me.Hide
End Sub

Private Sub Form_Load()
    cbKval.ListIndex = 4
End Sub

Private Sub chNah_Click()
    If chNah.Value = 1 Then
            Zkus
        Else
            Document(Index).picJPG.Visible = False
    End If
End Sub

Private Sub slKval_Change()
    If chNah.Value = 1 Then Zkus
End Sub

Private Sub slKval_Scroll()
    lbKval.Caption = "Kvalita: " & slKval.Value
End Sub

Public Sub ShowMe(sCesta, sIndex)
    Cesta = sCesta: Index = sIndex
    cDIB.CreateFromPicture Document(Index).picOb.Picture
    Me.Show 1, mdiMain
End Sub

Private Sub Zkus()
Dim Vel

    If Not SaveJPG(cDIB, App.Path & "\data\temp.jpg", slKval.Value) Then
        MsgBox "Chyba při vytváření náhledu", vbExclamation, "Chyba"
    End If
    With Document(Index).picJPG
        .Picture = LoadPicture(App.Path & "\data\temp.jpg")
        .Top = -1
        .Left = -1
        .Visible = True
    End With
```

```
    Vel = FileLen(App.Path & "\data\temp.jpg")
    If Vel < 1024 Then
        lbVel.Caption = Vel & " B"
    ElseIf Vel >= 1024 And Vel < 1048576 Then
        lbVel.Caption = (Vel * 100 \ 1024) / 100 & " KB"
    ElseIf Vel >= 1048576 Then
        lbVel.Caption = (Vel * 100 \ 1048576) / 100 & " MB"
    End If
End Sub
```

G.5. Panel nástrojů – frmNas

```
Option Explicit

Dim Barva
Dim Zmac(18) As Boolean

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = True: Exit Sub
End Sub

Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = False: Exit Sub
End Sub

Private Sub Form_Load()
Dim I

    For I = 1 To 18
        Zmac(I) = False
        picNas(I).Picture = imSed.ListImages(I).Picture
    Next I
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim I

    For I = 1 To 18
        If Zmac(I) = False Then picNas(I).Picture =
imSed.ListImages(I).Picture Else picNas(I).Picture =
imZm.ListImages(I).Picture
    Next I
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Cancel = 1
End Sub

Private Sub picBar_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = True: Exit Sub
End Sub

Private Sub picBar_KeyUp(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = False: Exit Sub
End Sub
```

```
Private Sub picNas_Click(Index As Integer)
Dim I
    If mdiMain.ActiveForm Is Nothing Then Exit Sub

    ZrusProm
    Skryj
    For I = 1 To 18
        picNas(I).Picture = imSed.ListImages(I).Picture
        Zmac(I) = False
    Next I
    picNas(Index).Picture = imNajZm.ListImages(Index).Picture
    Zmac(Index) = True
    mdiMain.ActiveForm.Pryc

    Select Case Index
        Case 1
            Vyber = True
        Case 2
            Presouvej = True
        Case 4
            Rucickuj = True
        Case 5
            Maluj = True
            mdiMain.frSto.Visible = True
            mdiMain.chG.Visible = False
            mdiMain.frSt.Visible = False
        Case 6
            Stetcuj = True
            mdiMain.frSto.Visible = True
            mdiMain.chG.Visible = False
            mdiMain.frSt.Visible = True
        Case 7
            Textuj = True
            mdiMain.frText.Visible = True
        Case 8
            Gumuj = True
            mdiMain.frSto.Visible = True
            mdiMain.chG.Visible = True
            mdiMain.frSt.Visible = False
        Case 9
            Vyplnuj = True
            mdiMain.frSto.Visible = False
            mdiMain.chG.Visible = False
            mdiMain.frSt.Visible = False
        Case 11
            Caruj = True
            mdiMain.picTyp(1).Visible = False
            mdiMain.picTyp(2).Visible = False
            For I = 0 To 3
                mdiMain.lin(I).Visible = False
            Next I
            mdiMain.chPop.Visible = False
            mdiMain.frTl.Visible = True
        Case 12
            Elipsuj = True
            mdiMain.picTyp(1).Visible = True
            mdiMain.picTyp(2).Visible = True
            For I = 0 To 3
                mdiMain.lin(I).Visible = True
```

```
            Next I
            mdiMain.chPop.Visible = True
            mdiMain.picTyp(1).Picture = mdiMain.imSed.ListImages(1).Picture
            mdiMain.picTyp(2).Picture = mdiMain.imSed.ListImages(2).Picture
            If    Plno    =    True    Then    mdiMain.picTyp(1).Picture    =
mdiMain.imZm.ListImages(1).Picture    Else    mdiMain.picTyp(2).Picture    =
mdiMain.imZm.ListImages(2).Picture
            mdiMain.frTl.Visible = True
        Case 13
            Obdelnikuj = True
            mdiMain.picTyp(1).Visible = True
            mdiMain.picTyp(2).Visible = True
            For I = 0 To 3
                mdiMain.lin(I).Visible = True
            Next I
            mdiMain.chPop.Visible = True
            mdiMain.picTyp(1).Picture = mdiMain.imSed.ListImages(3).Picture
            mdiMain.picTyp(2).Picture = mdiMain.imSed.ListImages(4).Picture
            If    Plno    =    True    Then    mdiMain.picTyp(1).Picture    =
mdiMain.imZm.ListImages(3).Picture    Else    mdiMain.picTyp(2).Picture    =
mdiMain.imZm.ListImages(4).Picture
            mdiMain.frTl.Visible = True
        Case 14
            Mer = True
            mdiMain.frMer.Visible = True
            For I = 0 To UBound(Document)
                Document(I).ZobMer
            Next I
        Case 15
            Kapatko = True
            TypKap = mdiMain.cbVz.ListIndex
            mdiMain.frVzo.Visible = True
        Case 16
            Zoomuj = True
        Case Else
            MsgBox "Ještě není naprogramováno", vbInformation, "Nedostatek
času"
    End Select
End Sub

Private Sub picNas_KeyDown(Index As Integer, KeyCode As Integer, Shift As
Integer)
    If KeyCode = vbKeyControl Then klCtrl = True: Exit Sub
End Sub

Private Sub picNas_KeyUp(Index As Integer, KeyCode As Integer, Shift As
Integer)
    If KeyCode = vbKeyControl Then klCtrl = False: Exit Sub
End Sub

Private Sub picNas_MouseMove(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
Dim I As Integer

    For I = 1 To 18
        If  Zmac(I)  =  True  And  I  =  Index  Then  picNas(I).Picture  =
imNajZm.ListImages(I).Picture
        If  Zmac(I)  =  True  And  I  <>  Index  Then  picNas(I).Picture  =
imZm.ListImages(I).Picture
```

```
        If Zmac(I) = False And I = Index Then picNas(I).Picture =
imNaj.ListImages(I).Picture
        If Zmac(I) = False And I <> Index Then picNas(I).Picture =
imSed.ListImages(I).Picture
    Next I
End Sub

Private Sub picPop_Click()
    TypBar = 1
    frmBar.Nove
    frmBar.Show 1
End Sub

Private Sub picPop_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = True: Exit Sub
End Sub

Private Sub picPop_KeyUp(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = False: Exit Sub
End Sub

Private Sub picPoz_Click()
    TypBar = 0
    frmBar.Nove
    frmBar.Show 1
End Sub

Private Sub picPoz_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = True: Exit Sub
End Sub

Private Sub picPoz_KeyUp(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = False: Exit Sub
End Sub

Private Sub picVym_Click()
    Barva = picPop.BackColor
    picPop.BackColor = picPoz.BackColor
    picPoz.BackColor = Barva
End Sub

Private Sub picVym_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = True: Exit Sub
End Sub

Private Sub picVym_KeyUp(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyControl Then klCtrl = False: Exit Sub
End Sub
```

G.6. Formulář dokumentu – frmOb

```
Option Explicit

Private Declare Function CreateSolidBrush Lib "gdi32" (ByVal crColor As
Long) As Long
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal
hObject As Long) As Long
```

```
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long)
As Long
Private Declare Function BeginPath Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function EndPath Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function TextOut Lib "gdi32" Alias "TextOutA" (ByVal hdc As
Long, ByVal X As Long, ByVal Y As Long, ByVal lpString As String, ByVal
nCount As Long) As Long
Private Declare Function StrokePath Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function StrokeAndFillPath Lib "gdi32" (ByVal hdc As Long)
As Long

Dim I
Dim StartX, StartY, HiX, HiY, LoX, LoY, Endx, EndY
Const Pi = 3.14159265
Dim ColUndo As New Collection
Dim ColRedo As New Collection
Dim Undo
Dim X1 As Single, Y1 As Single
Dim Hist(), HistInd()
Dim klShift As Boolean, Vzato As Boolean
Dim TuX, TuY
Dim GuX, GuY, GuBar
Dim StX, StY
Dim tStartX, tStartY, Text(), Akt, AktZ, Rad
Dim SelSt, SelEn
Dim Vlozeno As Boolean
Dim CarM(1) As Pravitko, CisM As Single, KonM As Single, RohM As Single,
AkceM As Single
Dim TeVem As Boolean, TeVzato As Boolean, Ozn As Boolean, Prv As Boolean,
TeX, TeY

Private Sub Form_Activate()
Dim I
Dim Sir, Vys

    mdiMain.lbPoz.Caption = "0;0 - " & picOb.ScaleWidth & "×" &
picOb.ScaleHeight
    If FState(Me.Tag).Velikost < 1024 Then
        mdiMain.lbVel.Caption = "Vel.: " & FState(Me.Tag).Velikost & " B"
    ElseIf FState(Me.Tag).Velikost >= 1024 And FState(Me.Tag).Velikost <
1048576 Then
        mdiMain.lbVel.Caption = "Vel.: " & Int(FState(Me.Tag).Velikost *
100 / 1024) / 100 & " KB"
    ElseIf FState(Me.Tag).Velikost >= 1048576 Then
        mdiMain.lbVel.Caption = "Vel.: " & Int(FState(Me.Tag).Velikost *
100 / 1048576) / 100 & " MB"
    End If
    mdiMain.txZo.Text = FState(Me.Tag).Zoom & "%"

    For I = 0 To 1
        Sir = Abs(CarM(I).X1 - CarM(I).X2)
        Vys = Abs(CarM(I).Y1 - CarM(I).Y2)
        mdiMain.lbS(I).Caption = "Š" & I + 1 & ": " & Int(Sir)
        mdiMain.lbV(I).Caption = "V" & I + 1 & ": " & Int(Vys)
        mdiMain.lbD(I).Caption = "D" & I + 1 & ": " & Int(((Sir ^ 2) + (Vys
^ 2)) ^ (0.5) * 100) / 100
    Next I
End Sub
```

```
Private Sub Form_Deactivate()
    Pryc
End Sub

Private Sub Form_Load()
    ReDim Hist(0): ReDim HistInd(0)
End Sub

Private Sub Form_LostFocus()
    Pryc
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
On Error GoTo Eror

    If FState(Me.Tag).Zmena = True Then
        Select Case MsgBox("Chcete uložit změny v dokumentu " & Me.Caption
& "?", vbQuestion + vbYesNoCancel, "Uložit změny?")
            Case vbYes
                If FState(Me.Tag).Cesta = "Novy" Then
                    mdiMain.CMD.Flags     =     cdlOFNOverwritePrompt     +
cdlOFNNoReadOnlyReturn
                    mdiMain.CMD.CancelError = True
                    mdiMain.CMD.Filter    =    "JPG    (*.jpg)|*.jpg|Bitmapa
(*.bmp)|*.bmp|Gif (*.gif)|*.gif|Průhledný Gif (*.gif)|*.gif"
                    mdiMain.CMD.FilterIndex = FState(Me.Tag).Typ
                    mdiMain.CMD.filename = Me.Caption
                    mdiMain.CMD.ShowSave
                    Ulozit  mdiMain.CMD.filename,  mdiMain.CMD.FilterIndex,
Me.Tag
                Else
                    Ulozit FState(Me.Tag).Cesta, FState(Me.Tag).Typ, Me.Tag
                End If
            Case vbCancel
                Exit Sub
        End Select
    End If

    mdiMain.lbVel = "Vel.: ": mdiMain.lbPoz.Caption = ""
    FState(Me.Tag).Smaz = True
    FState(Me.Tag).Cesta = "Novy"
    FState(Me.Tag).Zmena = False
    SmazHist
Exit Sub
Eror:
    If Err = 32755 Then Exit Sub
End Sub

Private Sub Form_Resize()
    If Me.ScaleHeight < 20 Or Me.ScaleHeight = 20 Then Exit Sub
    If Me.ScaleWidth < 20 Or Me.ScaleWidth = 20 Then Exit Sub
    scbHor.Top = Me.ScaleHeight - 17
    scbHor.Width = Me.ScaleWidth - 17
    scbVer.Height = Me.ScaleHeight - 17
    scbVer.Left = Me.ScaleWidth - 17
    picRes.Top = Me.ScaleHeight - 17
    picRes.Left = Me.ScaleWidth - 17

    If FState(Me.Tag).Zoom = 100 Then
```

```
        If picOb.Height > Me.ScaleHeight - 17 Then
                scbVer.Max = picOb.Height - Me.ScaleHeight + 17
                scbVer.SmallChange = Int(Me.ScaleHeight / 20)
                scbVer.Enabled = True
                scbVer_Change
            Else
                picOb.Top = Int(((Me.ScaleHeight - 17) / 2) - (picOb.Height
/ 2))
                scbVer.Enabled = False
        End If

        If picOb.Width > Me.ScaleWidth - 17 Then
                scbHor.Max = picOb.Width - Me.ScaleWidth + 17
                scbHor.SmallChange = Int(Me.ScaleWidth / 20)
                scbHor.Enabled = True
                scbHor_Change
            Else
                picOb.Left = Int(((Me.ScaleWidth - 17) / 2) - (picOb.Width
/ 2))
                scbHor.Enabled = False
        End If

        picStin.Top = picOb.Top + 3
        picStin.Left = picOb.Left + 3
        picStin.Height = picOb.Height
        picStin.Width = picOb.Width
    Else
        If picZoom.Height > Me.ScaleHeight - 17 Then
                scbVer.Max = picZoom.Height - Me.ScaleHeight + 17
                scbVer.SmallChange = Int(Me.ScaleHeight / 20)
                scbVer.Enabled = True
                scbVer_Change
            Else
                picZoom.Top   =   Int(((Me.ScaleHeight   -   17)   /   2)   -
(picZoom.Height / 2))
                scbVer.Enabled = False
        End If

        If picZoom.Width > Me.ScaleWidth - 17 Then
                scbHor.Max = picZoom.Width - Me.ScaleWidth + 17
                scbHor.SmallChange = Int(Me.ScaleWidth / 20)
                scbHor.Enabled = True
                scbHor_Change
            Else
                picZoom.Left   =   Int(((Me.ScaleWidth   -   17)   /   2)   -
(picZoom.Width / 2))
                scbHor.Enabled = False
        End If

        picStin.Top = picZoom.Top + 3
        picStin.Left = picZoom.Left + 3
        picStin.Height = picZoom.Height
        picStin.Width = picZoom.Width
    End If
End Sub

Private Sub imM_MouseDown(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
```

```
    picOb_MouseDown   Button,   Shift,   (X   /   Screen.TwipsPerPixelX)   +
imM(Index).Left, (Y / Screen.TwipsPerPixelY) + imM(Index).Top
End Sub

Private Sub imM_MouseMove(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
    picOb_MouseMove   Button,   Shift,   (X   /   Screen.TwipsPerPixelX)   +
imM(Index).Left, (Y / Screen.TwipsPerPixelY) + imM(Index).Top
End Sub

Private Sub imM_MouseUp(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
    picOb_MouseUp   Button,   Shift,   (X   /   Screen.TwipsPerPixelX)   +
imM(Index).Left, (Y / Screen.TwipsPerPixelY) + imM(Index).Top
End Sub

Private Sub mnuFilJas_Click()
End Sub

Private Sub mnuFilDua_Click()
    frmFil.ShowMe Me.Tag, "dua", shVyb.Visible
End Sub

Private Sub mnuFilKar_Click()
    If MsgBox("Vytáhnout obrysy???", vbQuestion + vbYesNo, "Karikatura") =
vbYes Then
            Karikatura shVyb.Visible, True
        Else
            Karikatura shVyb.Visible, False
    End If
    'frmFil.ShowMe Me.Tag, "kar", shVyb.Visible
End Sub

Private Sub mnuFilPix_Click()
    frmFil.ShowMe Me.Tag, "pix", shVyb.Visible
End Sub

Private Sub mnuFilPoi_Click()
    frmFil.ShowMe Me.Tag, "poi", shVyb.Visible
End Sub

Private Sub mnuFilSol_Click()
    frmFil.ShowMe Me.Tag, "sol", shVyb.Visible
End Sub

Private Sub mnuNapO_Click()
    frmAbout.Show 1, mdiMain
End Sub

Private Sub mnuOkNa_Click()
    mdiMain.Arrange 0
End Sub

Private Sub mnuOkSv_Click()
    mdiMain.Arrange 2
End Sub

Private Sub mnuOkVo_Click()
    mdiMain.Arrange 1
```

```
End Sub

Private Sub mnuSouKon_Click()
    mdiMain.Zavrit
End Sub

Private Sub mnuSouNov_Click()
    NovyProc
End Sub

Private Sub mnuSouOt_Click()
    Otevrit
End Sub

Private Sub mnuSouUl_Click()
On Error GoTo Eror
Dim UlTemp

    If FState(Me.Tag).Cesta = "Novy" Then
        mdiMain.CMD.Flags = cdlOFNOverwritePrompt + cdlOFNNoReadOnlyReturn
        mdiMain.CMD.CancelError = True
        mdiMain.CMD.Filter  =  "JPG  (*.jpg)|*.jpg|Bitmapa  (*.bmp)|*.bmp|Gif
(*.gif)|*.gif|Průhledný Gif (*.gif)|*.gif"
        mdiMain.CMD.FilterIndex = FState(Me.Tag).Typ
        mdiMain.CMD.filename = Me.Caption
        mdiMain.CMD.ShowSave
        Ulozit mdiMain.CMD.filename, mdiMain.CMD.FilterIndex, Me.Tag
        UlTemp = InStrRev(mdiMain.CMD.FileTitle, ".")
        Me.Caption = Left(mdiMain.CMD.FileTitle, UlTemp - 1)
        FState(Me.Tag).Typ = mdiMain.CMD.FilterIndex
        FState(Me.Tag).Cesta = mdiMain.CMD.filename
    Else
        Ulozit FState(Me.Tag).Cesta, FState(Me.Tag).Typ, Me.Tag
    End If
Exit Sub
Eror:
    If Err = 32755 Then Exit Sub
End Sub

Private Sub mnuSouUlJ_Click()
On Error GoTo Eror
Dim UlTemp

    If FState(Me.Tag).Cesta = "Novy" Then
        mdiMain.CMD.filename = Me.Caption
    Else
        mdiMain.CMD.filename = FState(Me.Tag).Cesta
    End If

    mdiMain.CMD.Flags = cdlOFNOverwritePrompt + cdlOFNNoReadOnlyReturn
    mdiMain.CMD.CancelError = True
    mdiMain.CMD.Filter   =   "JPG   (*.jpg)|*.jpg|Bitmapa   (*.bmp)|*.bmp|Gif
(*.gif)|*.gif|Průhledný Gif (*.gif)|*.gif"
    mdiMain.CMD.FilterIndex = FState(Me.Tag).Typ
    mdiMain.CMD.ShowSave
    Ulozit mdiMain.CMD.filename, mdiMain.CMD.FilterIndex, Me.Tag
    UlTemp = InStrRev(mdiMain.CMD.FileTitle, ".")
    Me.Caption = Left(mdiMain.CMD.FileTitle, UlTemp - 1)
    FState(Me.Tag).Typ = mdiMain.CMD.FilterIndex
```

```
    FState(Me.Tag).Cesta = mdiMain.CMD.filename
Exit Sub
Eror:
    If Err = 32755 Then Exit Sub
End Sub

Private Sub mnuSouZav_Click()
    Unload Me
End Sub

Private Sub mnuUp_Click()
    If Clipboard.GetFormat(vbCFBitmap) = True Then
        mnuUpVl.Enabled = True
    Else
        mnuUpVl.Enabled = False
    End If
    mnuUpOd.Enabled = shVyb.Visible
End Sub

Private Sub mnuUpKop_Click()
    If shVyb.Visible = True Then
        Clipboard.Clear
        Clipboard.SetData picClip.Clip, vbCFBitmap
    End If
End Sub

Private Sub mnuUpOd_Click()
    If Vzato = True Or Vlozeno = True Then
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Přesun": HistInd(Undo + 1) = 2
        UpdZpet
    End If
    Presun(Me.Tag) = False
    shVyb.Visible = False
    shVybAll.Visible = False
    Vzato = False
    mnuUpKop.Enabled = False
    mnuUpVyj.Enabled = False
    mnuUpVym.Enabled = False
End Sub

Private Sub mnuUpVl_Click()
    If shSel.Visible = True Then mnuUpOd_Click
    If Clipboard.GetFormat(vbCFBitmap) = True Then
        picTemp.Picture = Clipboard.GetData(vbCFBitmap)
        Presun(Me.Tag) = True
        Vzato = False
        shVyb.Left = -1: shVyb.Top = -1
        shVyb.Width    =    picTemp.ScaleWidth    +    2:    shVyb.Height    =
picTemp.ScaleHeight + 2
        shVyb.Visible = True
        picOb.PaintPicture picTemp.Picture, 0, 0
        Vlozeno = True
    End If
End Sub

Private Sub mnuUpVp_Click()
    Vpred
End Sub
```

```
Private Sub mnuUpVse_Click()
    mnuUpKop.Enabled = True
    mnuUpVyj.Enabled = True
    mnuUpVym.Enabled = True
    shVyb.Visible = True
    shVyb.Left = -1: shVyb.Top = -1
    shVyb.Width = picOb.ScaleWidth + 2: shVyb.Height = picOb.ScaleHeight +
2
    shVybAll.Visible = True
    shVybAll.Left = 0: shVybAll.Top = 0
    shVybAll.Width = picOb.ScaleWidth: shVybAll.Height = picOb.ScaleHeight
    picClip.Picture = picOb.Picture
    picClip.ClipX = 0: picClip.ClipY = 0
    picClip.ClipWidth    =    picOb.ScaleWidth:    picClip.ClipHeight    =
picOb.ScaleHeight
    picTemp.Picture = picClip.Clip
    Presun(Me.Tag) = True
    Vzato = False
End Sub

Private Sub mnuUpVyj_Click()
    If shVyb.Visible = True Then
        Clipboard.Clear
        Clipboard.SetData picClip.Clip, vbCFBitmap

        If Vzato = False Then
            Smaz True
            FState(Int(Me.Tag)).Zmena = True
            shVyb.Visible = False
            shVybAll.Visible = False
            Presun(Me.Tag) = False
        Else
            picOb.Cls
            shVyb.Visible = False
        End If
    End If
End Sub

Private Sub mnuUpVym_Click()
    If shVyb.Visible = True Then
        If Vzato = False Then
            Smaz True
            FState(Int(Me.Tag)).Zmena = True
            shVyb.Visible = False
            shVybAll.Visible = False
            Presun(Me.Tag) = False
        Else
            picOb.Cls
            shVyb.Visible = False
        End If
    End If
End Sub

Private Sub mnuUpZp_Click()
    Zpet
End Sub

Private Sub mnuZobBar_Click()
```

```
    mdiMain.Zobrazit 1
End Sub

Private Sub mnuZobNas_Click()
    mdiMain.Zobrazit 0
End Sub

Private Sub picOb_GotFocus()
    If Textuji(Me.Tag) = True Then tmrKur.Enabled = True
End Sub

Private Sub picOb_KeyDown(KeyCode As Integer, Shift As Integer)
Dim I

    If Textuji(Me.Tag) = True And tmrKur.Enabled = True Then
        If KeyCode = 17 Or KeyCode = 16 Then Exit Sub
        lnKur.Visible = True: tmrKur.Enabled = False
'======Ctrl Označení============
        If Shift = 3 Then
            If shSel.Visible = False Then
                shSel.Visible = True
                shSel.Width = 0: shSel.Height = picOb.TextHeight("X")
                SelSt = AktZ: SelEn = AktZ
                shSel.Left  =  tStartX  +  picOb.TextWidth(Left(Text(Akt),
AktZ)): shSel.Top = tStartY + picOb.TextHeight("X") * Akt
            End If
            Select Case KeyCode
                Case vbKeyLeft
                    If Not AktZ = 0 Then
                        Najdi (True)
                    End If
                Case vbKeyRight
                    If Not AktZ = Len(Text(Akt)) Then
                        Najdi (False)
                    End If
            End Select
            SelEn = AktZ
            If SelEn > SelSt Then
                shSel.Left  =  tStartX  +  picOb.TextWidth(Left(Text(Akt),
SelSt))
                shSel.Width  =  picOb.TextWidth(Left(Text(Akt),  SelEn))  -
picOb.TextWidth(Left(Text(Akt), SelSt))
            Else
                shSel.Left  =  tStartX  +  picOb.TextWidth(Left(Text(Akt),
SelEn))
                shSel.Width  =  picOb.TextWidth(Left(Text(Akt),  SelSt))  -
picOb.TextWidth(Left(Text(Akt), SelEn))
            End If
        End If

'======Označení================
        If Shift = 1 Then
            If shSel.Visible = False Then
                shSel.Visible = True
                shSel.Width = 0: shSel.Height = picOb.TextHeight("X")
                SelSt = AktZ: SelEn = AktZ
                shSel.Left  =  tStartX  +  picOb.TextWidth(Left(Text(Akt),
AktZ)): shSel.Top = tStartY + picOb.TextHeight("X") * Akt
            End If
```

```
            Select Case KeyCode
                Case vbKeyLeft
                    If Not AktZ = 0 Then
                        AktZ = AktZ - 1
                    End If
                Case vbKeyRight
                    If Not AktZ = Len(Text(Akt)) Then
                        AktZ = AktZ + 1
                    End If
            End Select
            SelEn = AktZ
            If SelEn > SelSt Then
                shSel.Left  =  tStartX  +  picOb.TextWidth(Left(Text(Akt),
SelSt))
                shSel.Width  =  picOb.TextWidth(Left(Text(Akt),  SelEn))  -
picOb.TextWidth(Left(Text(Akt), SelSt))
            Else
                shSel.Left  =  tStartX  +  picOb.TextWidth(Left(Text(Akt),
SelEn))
                shSel.Width  =  picOb.TextWidth(Left(Text(Akt),  SelSt))  -
picOb.TextWidth(Left(Text(Akt), SelEn))
            End If
        End If

'======Ctrl Posun===============
        If Shift = 2 Then
            Select Case KeyCode
                Case vbKeyLeft
                    If Not AktZ = 0 Then
                        Najdi (True)
                    ElseIf Not Akt = 0 Then
                        Akt = Akt - 1
                        AktZ = Len(Text(Akt))
                    End If
                    shSel.Visible = False
                Case vbKeyRight
                    If Not AktZ = Len(Text(Akt)) Then
                        Najdi (False)
                    ElseIf Not Akt = Rad Then
                        Akt = Akt + 1
                        AktZ = 0
                    End If
                    shSel.Visible = False
                Case vbKeyDelete
                    If shSel.Visible = True Then
                        If SelEn > SelSt Then
                            Text(Akt)    =    Left(Text(Akt),    SelSt)    &
Right(Text(Akt), Len(Text(Akt)) - SelEn)
                            AktZ = SelSt
                        Else
                            Text(Akt)    =    Left(Text(Akt),    SelEn)    &
Right(Text(Akt), Len(Text(Akt)) - SelSt)
                            AktZ = SelEn
                        End If
                        shSel.Visible = False
                        GoTo Tu
                    End If
                    If Not AktZ = Len(Text(Akt)) Then
```

```
                        SelSt = AktZ
                        Najdi (False)
                        SelEn = AktZ
                        Text(Akt)     =     Left(Text(Akt),     SelSt)     &
Right(Text(Akt), Len(Text(Akt)) - SelEn)
                        AktZ = SelSt
                    End If
                    If AktZ = Len(Text(Akt)) And Not Akt = Rad Then
                        Text(Akt) = Text(Akt) & Text(Akt + 1)
                        For I = Akt + 1 To Rad - 1
                            Text(I) = Text(I + 1)
                        Next I
                        Rad = Rad - 1
                        ReDim Preserve Text(Rad)
                    End If
            End Select
        End If

'======Posun====================
        If Shift = 0 Then
            Select Case KeyCode
                Case vbKeyLeft
                    If Not AktZ = 0 Then
                        AktZ = AktZ - 1
                    ElseIf Not Akt = 0 Then
                        Akt = Akt - 1
                        AktZ = Len(Text(Akt))
                    End If
                    shSel.Visible = False
                Case vbKeyRight
                    If Not AktZ = Len(Text(Akt)) Then
                        AktZ = AktZ + 1
                    ElseIf Not Akt = Rad Then
                        Akt = Akt + 1
                        AktZ = 0
                    End If
                    shSel.Visible = False
                Case vbKeyDown
                    If Not Akt = Rad Then Akt = Akt + 1
                    If AktZ > Len(Text(Akt)) Then AktZ = Len(Text(Akt))
                    shSel.Visible = False
                Case vbKeyUp
                    If Not Akt = 0 Then Akt = Akt - 1
                    If AktZ > Len(Text(Akt)) Then AktZ = Len(Text(Akt))
                    shSel.Visible = False
                Case vbKeyDelete
                    If shSel.Visible = True Then
                        If SelEn > SelSt Then
                            Text(Akt)    =    Left(Text(Akt),    SelSt)    &
Right(Text(Akt), Len(Text(Akt)) - SelEn)
                            AktZ = SelSt
                        Else
                            Text(Akt)    =    Left(Text(Akt),    SelEn)    &
Right(Text(Akt), Len(Text(Akt)) - SelSt)
                            AktZ = SelEn
                        End If
                        shSel.Visible = False
                        GoTo Tu
                    End If
```

```
                    If Not AktZ = Len(Text(Akt)) Then
                        Text(Akt)     =     Left(Text(Akt),     AktZ)     &
Right(Text(Akt), Len(Text(Akt)) - AktZ - 1)
                    End If
                    If AktZ = Len(Text(Akt)) And Not Akt = Rad Then
                        Text(Akt) = Text(Akt) & Text(Akt + 1)
                        For I = Akt + 1 To Rad - 1
                            Text(I) = Text(I + 1)
                        Next I
                        Rad = Rad - 1
                        ReDim Preserve Text(Rad)
                    End If
            End Select
        End If
Tu:
        lnKur.X1   =   tStartX   +   picOb.TextWidth(Left(Text(Akt),   AktZ)):
lnKur.X2 = lnKur.X1
        lnKur.Y1   =   tStartY   +   Akt   *   picOb.TextHeight("X"):   lnKur.Y2   =
lnKur.Y1 + picOb.TextHeight("X")
        Textni
    End If
    If KeyCode = vbKeyControl Then klCtrl = True: Exit Sub
    If KeyCode = vbKeyShift Then klShift = True: Exit Sub
End Sub

Private Sub picOb_KeyPress(KeyAscii As Integer)
Dim I, Temp

    If Textuji(Me.Tag) = False Then Exit Sub
    lnKur.Visible = True
    If shSel.Visible = True Then
        If SelEn > SelSt Then
            Text(Akt)    =    Left(Text(Akt),    SelSt)    &    Right(Text(Akt),
Len(Text(Akt)) - SelEn)
            AktZ = SelSt
        Else
            Text(Akt)    =    Left(Text(Akt),    SelEn)    &    Right(Text(Akt),
Len(Text(Akt)) - SelSt)
            AktZ = SelEn
        End If
    End If
    Select Case KeyAscii
        Case 8 'Bspc
            If shSel.Visible = True Then GoTo Tu
            If Not AktZ = 0 Then
                Text(Akt)  =  Left(Text(Akt),  AktZ  -  1)  &  Right(Text(Akt),
Len(Text(Akt)) - AktZ)
                AktZ = AktZ - 1
            End If
            If AktZ = 0 And Not Akt = 0 Then
                Temp = Len(Text(Akt - 1))
                Text(Akt - 1) = Text(Akt - 1) & Text(Akt)
                For I = Akt To Rad - 1
                    Text(I) = Text(I + 1)
                Next I
                Rad = Rad - 1
                ReDim Preserve Text(Rad)
                Akt = Akt - 1
                AktZ = Temp
```

```
            End If
        Case 13 'Ent
            Rad = Rad + 1
            ReDim Preserve Text(Rad)
            Akt = Akt + 1
            If Not Akt = Rad Then
                For I = Rad To Akt + 1 Step -1
                    Text(I) = Text(I - 1)
                    Text(I - 1) = ""
                Next I
            End If
            If Not AktZ = 0 Then
                Text(Akt) = Right(Text(Akt - 1), Len(Text(Akt - 1)) - AktZ)
                Text(Akt - 1) = Left(Text(Akt - 1), AktZ)
            End If
            AktZ = 0
        Case 127 'Ctrl + Bspc
            If shSel.Visible = True Then GoTo Tu
            If Not AktZ = 0 Then
                SelEn = AktZ
                Najdi (True)
                SelSt = AktZ
                Text(Akt) = Left(Text(Akt), SelSt) & Right(Text(Akt),
Len(Text(Akt)) - SelEn)
                AktZ = SelSt
            End If
            If AktZ = 0 And Not Akt = 0 Then
                Temp = Len(Text(Akt - 1))
                Text(Akt - 1) = Text(Akt - 1) & Text(Akt)
                For I = Akt To Rad - 1
                    Text(I) = Text(I + 1)
                Next I
                Rad = Rad - 1
                ReDim Preserve Text(Rad)
                Akt = Akt - 1
                AktZ = Temp
            End If
        Case Else
            Text(Akt) = Left(Text(Akt), AktZ) & Chr(KeyAscii) &
Right(Text(Akt), Len(Text(Akt)) - AktZ)
            AktZ = AktZ + 1
    End Select
Tu:
    shSel.Visible = False
    lnKur.X1 = tStartX + picOb.TextWidth(Left(Text(Akt), AktZ)): lnKur.X2 =
lnKur.X1
    lnKur.Y1 = tStartY + Akt * picOb.TextHeight("X"): lnKur.Y2 = lnKur.Y1 +
picOb.TextHeight("X")
    Textni
End Sub

Private Sub picOb_KeyUp(KeyCode As Integer, Shift As Integer)
    If shSel.Visible = False And Textuji(Me.Tag) = True Then tmrKur.Enabled
= True
    If KeyCode = vbKeyControl Then klCtrl = False: Exit Sub
    If KeyCode = vbKeyShift Then klShift = False: Exit Sub
End Sub

Private Sub picOb_LostFocus()
```

```
    If tmrKur.Enabled = True Then tmrKur.Enabled = False
End Sub

Private Sub picOb_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim Temp

'======Text====================
    If Textuji(Me.Tag) = True Then
        If Button = vbLeftButton And TeVem = True Then
            TeVzato = True
        End If
        If (Y > shTe.Top + 4 And Y < shTe.Top + shTe.Height - 4 And X >
shTe.Left + 4 And X < shTe.Left + shTe.Width - 4) Then
            tmrKur.Enabled = False
            Mys X, Y
            Ozn = True: Prv = True
        End If
        Exit Sub
    End If

    If Textuj = True And Textuji(Me.Tag) = False Then
        If shVyb.Visible = True Then
            mnuUpOd_Click
        End If
        tStartX = X: tStartY = Y
        Textuji(Me.Tag) = True
        picOb.FontName = mdiMain.cbFont.Text
        picOb.FontSize = mdiMain.cbVel.Text
        picOb.DrawWidth = Int(mdiMain.txSir.Text)
        Select Case mdiMain.cbSpec.ListIndex
            Case 0
                picOb.FontUnderline = False: picOb.FontStrikethru = False
            Case 1
                picOb.FontUnderline = True: picOb.FontStrikethru = False
            Case 2
                picOb.FontUnderline = False: picOb.FontStrikethru = True
            Case 3
                picOb.FontUnderline = True: picOb.FontStrikethru = True
        End Select
        Select Case mdiMain.cbTyp.ListIndex
            Case 0
                picOb.FontItalic = False: picOb.FontBold = False
            Case 1
                picOb.FontItalic = True: picOb.FontBold = False
            Case 2
                picOb.FontItalic = False: picOb.FontBold = True
            Case 3
                picOb.FontItalic = True: picOb.FontBold = True
        End Select
        Select Case mdiMain.cbObr.ListIndex
            Case 0
                picOb.DrawStyle = 5
                JenObrys = False
            Case 1
                picOb.DrawStyle = mdiMain.cbCar.ListIndex
                JenObrys = False
            Case 2
                picOb.DrawStyle = mdiMain.cbCar.ListIndex
```

```
                JenObrys = True
        End Select
        picOb.DrawWidth = Val(mdiMain.txSir.Text)
        If Val(mdiMain.txSir.Text) > 1 Then mdiMain.cbCar.Enabled = False
        ReDim Text(0): Akt = 0: Rad = 0: AktZ = 0
        lnKur.X1 = tStartX: lnKur.X2 = lnKur.X1
        lnKur.Y1 = tStartY: lnKur.Y2 = tStartY + picOb.TextHeight("X")
        lnKur.Visible = True
        tmrKur.Enabled = True
        shTe.Visible = True
        Textni
        Exit Sub
    End If

'======Lupa====================
    'If Zoomuj = True Then
    '    FState(Me.Tag).Zoom = 200
    '    Zoom
    'End If

'======Pravítko================
    If Mer = True And Button = vbLeftButton Then
        If AkceM = 2 Then 'vytvoření druhé
            If X > imM(0).Left - 3 * (picOb.ScaleHeight / (picOb.Height -
2)) And X < imM(0).Left + 10 And Y > imM(0).Top - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)) And Y < imM(0).Top + 10 Then
                If (imM(0).Left + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).X1 And imM(0).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y1) Or (imM(0).Left + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).X2 And imM(0).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y2) Then
                    If imM(0).Left + 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) = lnM(0).X2 And imM(0).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y2 Then
                        Temp = lnM(0).X1: lnM(0).X1 = lnM(0).X2: lnM(0).X2
= Temp: Temp = lnM(0).Y1: lnM(0).Y1 = lnM(0).Y2: lnM(0).Y2 = Temp
                    End If
                    CisM = 1
                    lnM(1).X1 = lnM(0).X1: lnM(1).Y1 = lnM(0).Y1: lnM(1).X2
= lnM(0).X1: lnM(1).Y2 = lnM(0).Y1
                    lnM(1).Visible = True
                    shM.Left = lnM(0).X1 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): shM.Top = lnM(0).Y1 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
                    shM.Visible = True
                    imM(0).Left = lnM(0).X2 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): imM(0).Top = lnM(0).Y2 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
                    CarM(0).X1 = lnM(0).X1: CarM(0).Y1 = lnM(0).Y1:
CarM(0).X2 = lnM(0).X2: CarM(0).Y2 = lnM(0).Y2
                    CarM(1).X1 = lnM(0).X1: CarM(1).Y1 = lnM(0).Y1:
CarM(1).X2 = lnM(0).X1: CarM(1).Y2 = lnM(0).Y1
                    CarM(1).Visible = True
                Else
                    If imM(0).Left + 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) = lnM(1).X2 And imM(0).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(1).Y2 Then
                        Temp = lnM(1).X1: lnM(1).X1 = lnM(1).X2: lnM(1).X2
= Temp: Temp = lnM(1).Y1: lnM(1).Y1 = lnM(1).Y2: lnM(1).Y2 = Temp
```

```
                    End If
                    CisM = 0
                    lnM(0).X1 = lnM(1).X1: lnM(0).Y1 = lnM(1).Y1: lnM(0).X2
= lnM(1).X1: lnM(0).Y2 = lnM(1).Y1
                    lnM(0).Visible = True
                    shM.Left = lnM(1).X1 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): shM.Top = lnM(1).Y1 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
                    shM.Visible = True
                    imM(0).Left = lnM(1).X2 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): imM(0).Top = lnM(1).Y2 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
                    CarM(1).X1 = lnM(1).X1: CarM(1).Y1 = lnM(1).Y1:
CarM(1).X2 = lnM(1).X2: CarM(1).Y2 = lnM(1).Y2
                    CarM(0).X1 = lnM(1).X1: CarM(0).Y1 = lnM(1).Y1:
CarM(0).X2 = lnM(1).X1: CarM(0).Y2 = lnM(1).Y1
                    CarM(0).Visible = True
                End If
                RohM = 1
            ElseIf X > imM(1).Left - 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) And X < imM(1).Left + 10 * (picOb.ScaleHeight / (picOb.Height - 2))
And Y > imM(1).Top - 3 * (picOb.ScaleHeight / (picOb.Height - 2)) And Y <
imM(1).Top + 10 * (picOb.ScaleHeight / (picOb.Height - 2)) Then
                If (imM(1).Left + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).X1 And imM(1).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y1) Or (imM(1).Left + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).X2 And imM(1).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y2) Then
                    CisM = 1
                    If imM(1).Left + 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) = lnM(0).X2 And imM(1).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y2 Then
                        Temp = lnM(0).X1: lnM(0).X1 = lnM(0).X2: lnM(0).X2
= Temp: Temp = lnM(0).Y1: lnM(0).Y1 = lnM(0).Y2: lnM(0).Y2 = Temp
                    End If
                    lnM(1).X1 = lnM(0).X1: lnM(1).Y1 = lnM(0).Y1: lnM(1).X2
= lnM(0).X1: lnM(1).Y2 = lnM(0).Y1
                    lnM(1).Visible = True
                    shM.Left = lnM(0).X1 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): shM.Top = lnM(0).Y1 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
                    imM(0).Left = lnM(0).X2 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): imM(0).Top = lnM(0).Y2 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
                    shM.Visible = True
                    CarM(0).X1 = lnM(0).X1: CarM(0).Y1 = lnM(0).Y1:
CarM(0).X2 = lnM(0).X2: CarM(0).Y2 = lnM(0).Y2
                    CarM(1).X1 = lnM(0).X1: CarM(1).Y1 = lnM(0).Y1:
CarM(1).X2 = lnM(0).X1: CarM(1).Y2 = lnM(0).Y1
                    CarM(1).Visible = True
                Else
                    CisM = 0
                    If imM(1).Left + 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) = lnM(1).X2 And imM(1).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(1).Y2 Then
                        Temp = lnM(1).X1: lnM(1).X1 = lnM(1).X2: lnM(1).X2
= Temp: Temp = lnM(1).Y1: lnM(1).Y1 = lnM(1).Y2: lnM(1).Y2 = Temp
                    End If
```

```
                    lnM(0).X1 = lnM(1).X1: lnM(0).Y1 = lnM(1).Y1: lnM(0).X2
= lnM(1).X1: lnM(0).Y2 = lnM(1).Y1
                    lnM(0).Visible = True
                    shM.Left = lnM(1).X1 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): shM.Top = lnM(1).Y1 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
                    shM.Visible = True
                    imM(0).Left = lnM(1).X2 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): imM(0).Top = lnM(1).Y2 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
                    CarM(1).X1 = lnM(1).X1: CarM(1).Y1 = lnM(1).Y1:
CarM(1).X2 = lnM(1).X2: CarM(1).Y2 = lnM(1).Y2
                    CarM(0).X1 = lnM(1).X1: CarM(0).Y1 = lnM(1).Y1:
CarM(0).X2 = lnM(1).X1: CarM(0).Y2 = lnM(1).Y1
                    CarM(0).Visible = True
                End If
                RohM = 1
            End If
            Merim = True
        ElseIf AkceM = 1 Then 'editace vytvořené
            If X > imM(0).Left - 3 * (picOb.ScaleHeight / (picOb.Height -
2)) And X < imM(0).Left + 10 * (picOb.ScaleHeight / (picOb.Height - 2)) And
Y > imM(0).Top - 3 * (picOb.ScaleHeight / (picOb.Height - 2)) And Y <
imM(0).Top + 10 * (picOb.ScaleHeight / (picOb.Height - 2)) Then
                If (imM(0).Left + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).X1 And imM(0).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y1) Or (imM(0).Left + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).X2 And imM(0).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y2) Then
                    CisM = 0
                    If imM(0).Left + 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) = lnM(0).X2 And imM(0).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y2 Then
                        KonM = 1
                    Else
                        KonM = 0
                    End If
                Else
                    CisM = 1
                    If imM(0).Left + 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) = lnM(1).X2 And imM(0).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(1).Y2 Then
                        KonM = 1
                    Else
                        KonM = 0
                    End If
                End If
                RohM = 0
            ElseIf X > imM(1).Left - 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) And X < imM(1).Left + 10 * (picOb.ScaleHeight / (picOb.Height - 2))
And Y > imM(1).Top - 3 * (picOb.ScaleHeight / (picOb.Height - 2)) And Y <
imM(1).Top + 10 * (picOb.ScaleHeight / (picOb.Height - 2)) Then
                If (imM(1).Left + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).X1 And imM(1).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y1) Or (imM(1).Left + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).X2 And imM(1).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y2) Then
                    CisM = 0
```

```
                    If imM(1).Left + 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) = lnM(0).X2 And imM(1).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(0).Y2 Then
                        KonM = 1
                    Else
                        KonM = 0
                    End If
                Else
                    CisM = 1
                    If imM(1).Left + 3 * (picOb.ScaleHeight / (picOb.Height
- 2)) = lnM(1).X2 And imM(1).Top + 3 * (picOb.ScaleHeight / (picOb.Height -
2)) = lnM(1).Y2 Then
                        KonM = 1
                    Else
                        KonM = 0
                    End If
                End If
                RohM = 1
            End If
            Merim = True
        ElseIf AkceM = 0 Then 'vytvoření nové
            imM(0).Left = Int(X) + 0.49 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): imM(0).Top = Int(Y) + 0.49 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
            imM(1).Left = Int(X) + 0.49 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): imM(1).Top = Int(Y) + 0.49 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2))
            lnM(0).X1 = Int(X) + 0.49: lnM(0).Y1 = Int(Y) + 0.49: lnM(0).X2
= Int(X) + 0.49: lnM(0).Y2 = Int(Y) + 0.49
            CarM(0).X1 = Int(X) + 0.49: CarM(0).Y1 = Int(Y) + 0.49:
CarM(0).X2 = Int(X) + 0.49: CarM(0).Y2 = Int(Y) + 0.49
            imM(0).Visible = True: imM(1).Visible = True
            lnM(0).Visible = True
            CisM = 0: RohM = 1
            CarM(0).Visible = True
            mdiMain.lbS(0).Caption = "Š1: 0": mdiMain.lbV(0).Caption = "V1:
0": mdiMain.lbD(0).Caption = "D1: 0": mdiMain.lbU.Caption = "U: 0"
            lnM(1).Visible = False
            CarM(1).Visible = False
            shM.Visible = False
            KonM = 1
            Merim = True
        End If
    End If

'======Plechovka barvy==========
    If Vyplnuj = True And Button = vbLeftButton Then
        If shVyb.Visible = True Then
            NastTemp
            picTemp.FillStyle = 0
            picTemp.FillColor = frmNas.picPop.BackColor
            ExtFloodFill picTemp.hdc, X - picTemp.Left, Y - picTemp.Top,
picTemp.Point(X - picTemp.Left, Y - picTemp.Top), FLOODFILLSURFACE
            Vykresli
        Else
            picOb.FillStyle = 0
            picOb.FillColor = frmNas.picPop.BackColor
            ExtFloodFill    picOb.hdc,    X,    Y,    picOb.Point(X,    Y),
FLOODFILLSURFACE
```

```
        End If
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Plechovka barvy": HistInd(Undo + 1) = 9
        UpdZpet
    End If

'======Štetec==================
    If Stetcuj = True And Button = vbLeftButton Then
        Stetcuji = True
        X1 = X: Y1 = Y
        If shVyb.Visible = True Then
            NastTemp
            picTemp.DrawWidth = Int(frmSte.txTl.Text)
            picTemp.ForeColor = frmNas.picPop.BackColor
            If klShift = True Then
                Vypocti Int(StX - picTemp.Left), Int(StY - picTemp.Top),
Int(X - picTemp.Left), Int(Y - picTemp.Top), 1
            Else
                Vypocti Int(X1 - picTemp.Left), Int(Y1 - picTemp.Top),
Int(X - picTemp.Left), Int(Y - picTemp.Top), 1
            End If
        Else
            picOb.DrawWidth = Int(frmSte.txTl.Text)
            picOb.ForeColor = frmNas.picPop.BackColor
            If klShift = True Then
                Vypocti Int(StX), Int(StY), Int(X), Int(Y), 0
            Else
                Vypocti Int(X1), Int(Y1), Int(X), Int(Y), 0
            End If
        End If
        Exit Sub
    End If

'======Guma====================
    If Gumuj = True And Button = vbLeftButton Then
        Gumuji = True
        picOb.DrawStyle = 0
        X1 = X: Y1 = Y
        If mdiMain.chG.Value = 1 Then GuBar = frmNas.picPoz.BackColor Else
GuBar = vbWhite
        If shVyb.Visible = True Then
            NastTemp
            picTemp.DrawWidth = Int(frmSte.txTl.Text)
            If klShift = True Then
                picTemp.Line (GuX - picTemp.Left, GuY - picTemp.Top)-(X -
picTemp.Left, Y - picTemp.Top), GuBar
            Else
                picTemp.Line (X1 - picTemp.Left, Y1 - picTemp.Top)-(X -
picTemp.Left, Y - picTemp.Top), GuBar
            End If
        Else
            picOb.DrawWidth = Int(frmSte.txTl.Text)
            If klShift = True Then
                picOb.Line (GuX, GuY)-(X, Y), GuBar
            Else
                picOb.Line (X1, Y1)-(X, Y), GuBar
            End If
        End If
        Exit Sub
```

```
    End If

'======Obdélník=================
    If Obdelnikuj = True And Button = vbLeftButton Then
        Obdelnikuji = True
        picOb.DrawStyle = 0
        X1 = X: Y1 = Y
        If Plno = True Then picOb.FillStyle = 0: picTemp.FillStyle = 0 Else
picOb.FillStyle = 1: picTemp.FillStyle = 1
        If    mdiMain.chPop.Value    =    1    Then    picOb.FillColor    =
frmNas.picPop.BackColor: picTemp.FillColor = frmNas.picPop.BackColor Else
picOb.FillColor    =    frmNas.picPoz.BackColor:    picTemp.FillColor    =
frmNas.picPoz.BackColor
        If shVyb.Visible = True Then
            NastTemp
            picTemp.DrawWidth = Int(mdiMain.txTl.Text)
        Else
            picOb.DrawWidth = Int(mdiMain.txTl.Text)
        End If
        Exit Sub
    End If

'======Elipsa==================
    If Elipsuj = True And Button = vbLeftButton Then
        Elipsuji = True
        picOb.DrawStyle = 0
        X1 = X: Y1 = Y
        If Plno = True Then picOb.FillStyle = 0: picTemp.FillStyle = 0 Else
picOb.FillStyle = 1: picTemp.FillStyle = 1
        If    mdiMain.chPop.Value    =    1    Then    picOb.FillColor    =
frmNas.picPop.BackColor: picTemp.FillColor = frmNas.picPop.BackColor Else
picOb.FillColor    =    frmNas.picPoz.BackColor:    picTemp.FillColor    =
frmNas.picPoz.BackColor
        If shVyb.Visible = True Then
            NastTemp
            picTemp.DrawWidth = Int(mdiMain.txTl.Text)
        Else
            picOb.DrawWidth = Int(mdiMain.txTl.Text)
        End If
        Exit Sub
    End If

'======Cára====================
    If Caruj = True And Button = vbLeftButton Then
        picOb.DrawStyle = 0
        Caruji = True
        X1 = X: Y1 = Y
        If shVyb.Visible = True Then
            NastTemp
            picTemp.DrawWidth = Int(mdiMain.txTl.Text)
        Else
            picOb.DrawWidth = Int(mdiMain.txTl.Text)
        End If
        Exit Sub
    End If

'======Tuzka====================
    If Maluj = True And Button = vbLeftButton Then
        Maluji = True
```

```
        picOb.DrawStyle = 0
        X1 = X: Y1 = Y
        If shVyb.Visible = True Then
            NastTemp
            picTemp.DrawWidth = Int(frmSte.txTl.Text)
            If klShift = True Then picTemp.Line (TuX - picTemp.Left, TuY -
picTemp.Top)-(X - picTemp.Left, Y - picTemp.Top), frmNas.picPop.BackColor
            picTemp.Line (X1 - picTemp.Left, Y1 - picTemp.Top)-(X -
picTemp.Left, Y - picTemp.Top), frmNas.picPop.BackColor
        Else
            picOb.DrawWidth = Int(frmSte.txTl.Text)
            If klShift = True Then picOb.Line (TuX, TuY)-(X, Y),
frmNas.picPop.BackColor
            picOb.Line (X1, Y1)-(X, Y), frmNas.picPop.BackColor
        End If
        Exit Sub
    End If

'======Presun==================
    If Presouvej = True And Presun(Me.Tag) = True And Button = vbLeftButton
Then
        shVybAll.Visible = False
        Presouvam = True
        StartX = X - shVyb.Left: StartY = Y - shVyb.Top
        If klCtrl = False And Vzato = False And Vlozeno = False Then
            Smaz False
            picOb.Cls
            picOb.PaintPicture picTemp.Picture, X - StartX + 1, Y - StartY
+ 1
            shVyb.Left = X - StartX: shVyb.Top = Y - StartY
            Vzato = True
        ElseIf klCtrl = True And Vzato = True Then
            picOb.Cls
            picOb.PaintPicture picTemp.Picture, X - StartX + 1, Y - StartY
+ 1
            shVyb.Left = X - StartX: shVyb.Top = Y - StartY
            ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
            Hist(Undo + 1) = "Přesun": HistInd(Undo + 1) = 2
            UpdZpet
        End If
        Vzato = True
        Exit Sub
    End If

'======Výber==================
    If Vyber = True And Button = vbLeftButton Then
        mnuUpOd_Click
        Vlozeno = False
        Vybiram = True
        HiX = X: HiY = Y
        LoX = X: LoY = Y
        shVyb.Width = 1
        shVyb.Height = 1
        shZVyb.Width = 1
        shZVyb.Height = 1
        mnuUpKop.Enabled = True
        mnuUpVyj.Enabled = True
        mnuUpVym.Enabled = True
        Exit Sub
```

```
    End If
End Sub

Private Sub picOb_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim Pomer, Tloustka
Dim Sir As Single, Vys As Single, Uhel, Uhel2

    shSte.Visible = False
    shZSte.Visible = False
    mdiMain.lbPoz.Caption = X + 1 & ";" & Y + 1 & " - " & picOb.ScaleWidth
& "×" & picOb.ScaleHeight

'======Text====================
    If Textuji(Me.Tag) = True Then
        If Y > shTe.Top + 4 And Y < shTe.Top + shTe.Height - 4 And X >
shTe.Left + 4 And X < shTe.Left + shTe.Width - 4 Then
            picOb.MouseIcon = imIco.ListImages(4).Picture
            TeVem = False
        Else
            picOb.MouseIcon = imIco.ListImages(1).Picture
            TeVem = True
        End If

        If TeVzato = True Then
            picOb.Cls
            tmrKur.Enabled = False
            tStartX = tStartX + X - TeX: tStartY = tStartY + Y - TeY
            shSel.Left = shSel.Left + X - TeX: shSel.Top = shSel.Top + Y -
TeY
            Textni
        End If

        If Ozn = True And Prv = True Then
            shSel.Visible = True
            shSel.Width = 0: shSel.Height = picOb.TextHeight("X")
            SelSt = AktZ: SelEn = AktZ
            shSel.Left = tStartX + picOb.TextWidth(Left(Text(Akt), AktZ)):
shSel.Top = tStartY + picOb.TextHeight("X") * Akt
            tmrKur.Enabled = False
            Prv = False
        End If

        If Ozn = True Then
            Mys X, Y
            SelEn = AktZ
            If SelEn > SelSt Then
                shSel.Left = tStartX + picOb.TextWidth(Left(Text(Akt),
SelSt))
                shSel.Width = picOb.TextWidth(Left(Text(Akt), SelEn)) -
picOb.TextWidth(Left(Text(Akt), SelSt))
            Else
                shSel.Left = tStartX + picOb.TextWidth(Left(Text(Akt),
SelEn))
                shSel.Width = picOb.TextWidth(Left(Text(Akt), SelSt)) -
picOb.TextWidth(Left(Text(Akt), SelEn))
            End If
        End If
        TeX = X: TeY = Y
```

```
        Exit Sub
    End If

    If Textuj = True Then
        GetKeyboardState kbArray
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picOb.MouseIcon =
imIco.ListImages(10).Picture Else picOb.MouseIcon =
imIco.ListImages(4).Picture
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picZoom.MouseIcon =
imIco.ListImages(10).Picture Else picZoom.MouseIcon =
imIco.ListImages(4).Picture
        Exit Sub
    End If

'======Pravítko================
    If Mer = True Then
        If lnM(0).Visible = True And ((X > imM(0).Left - 3 *
(picOb.ScaleHeight / (picOb.Height - 2)) And X < imM(0).Left + 10 *
(picOb.ScaleHeight / (picOb.Height - 2)) And Y > imM(0).Top - 3 *
(picOb.ScaleHeight / (picOb.Height - 2)) And Y < imM(0).Top + 10 *
(picOb.ScaleHeight / (picOb.Height - 2))) Or (X > imM(1).Left - 3 *
(picOb.ScaleHeight / (picOb.Height - 2)) And X < imM(1).Left + 10 *
(picOb.ScaleHeight / (picOb.Height - 2)) And Y > imM(1).Top - 3 *
(picOb.ScaleHeight / (picOb.Height - 2)) And Y < imM(1).Top + 10 *
(picOb.ScaleHeight / (picOb.Height - 2)))) Then
            If klCtrl = True And Merim = False Then
                picOb.MouseIcon = imIco.ListImages(15).Picture
                picZoom.MouseIcon = imIco.ListImages(15).Picture
                AkceM = 2
            Else
                picOb.MouseIcon = imIco.ListImages(14).Picture
                picZoom.MouseIcon = imIco.ListImages(14).Picture
                AkceM = 1
            End If
        ElseIf Merim = False Then
            picOb.MouseIcon = imIco.ListImages(7).Picture
            picZoom.MouseIcon = imIco.ListImages(7).Picture
            AkceM = 0
        End If
    End If

    If Merim = True Then
        imM(RohM).Left = Int(X) + 0.49 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): imM(RohM).Top = Int(Y) + 0.49 - 3 * (picOb.ScaleHeight
/ (picOb.Height - 2))
        If KonM = 1 Then
            lnM(CisM).X2 = Int(X) + 0.49: lnM(CisM).Y2 = Int(Y) + 0.49
            CarM(CisM).X2 = Int(X) + 0.49: CarM(CisM).Y2 = Int(Y) + 0.49
        Else
            lnM(CisM).X1 = Int(X) + 0.49: lnM(CisM).Y1 = Int(Y) + 0.49
            CarM(CisM).X1 = Int(X) + 0.49: CarM(CisM).Y1 = Int(Y) + 0.49
        End If

        Sir = Abs(lnM(CisM).X1 - lnM(CisM).X2)
        Vys = Abs(lnM(CisM).Y1 - lnM(CisM).Y2)
        mdiMain.lbS(CisM).Caption = "Š" & CisM + 1 & ": " & Int(Sir)
        mdiMain.lbV(CisM).Caption = "V" & CisM + 1 & ": " & Int(Vys)
        mdiMain.lbD(CisM).Caption = "D" & CisM + 1 & ": " & Int(((Sir ^ 2)
+ (Vys ^ 2)) ^ (0.5) * 100) / 100
```

```
        If CarM(0).Visible = True And CarM(1).Visible = False Then
            Sir = CarM(0).X2 - CarM(0).X1
            Vys = CarM(0).Y1 - CarM(0).Y2
            If Sir = 0 Then
                mdiMain.lbU.Caption = "U: " & 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    mdiMain.lbU.Caption = "U: " & Int(Atn(Vys / Sir) * 180
* 100 / Pi) / 100
                Else
                    mdiMain.lbU.Caption = "U: " & (180 + Int(Atn(Vys / Sir)
* 180 * 100 / Pi) / 100 * PlusMinus(Vys)) * PlusMinus(Vys)
                End If
            End If
        ElseIf CarM(0).Visible = True And CarM(1).Visible = True Then
            Sir = CarM(0).X2 - CarM(0).X1
            Vys = CarM(0).Y1 - CarM(0).Y2
            If Sir = 0 Then
                Uhel = 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    Uhel = Int(Atn(Vys / Sir) * 180 * 100 / Pi) / 100
                Else
                    Uhel = (180 + Int(Atn(Vys / Sir) * 180 * 100 / Pi) /
100 * PlusMinus(Vys)) * PlusMinus(Vys)
                End If
            End If
            If Not Uhel = Abs(Uhel) Then Uhel = 360 + Uhel

            Sir = CarM(1).X2 - CarM(1).X1
            Vys = CarM(1).Y1 - CarM(1).Y2
            If Sir = 0 Then
                Uhel2 = 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    Uhel2 = Int(Atn(Vys / Sir) * 180 * 100 / Pi) / 100
                Else
                    Uhel2 = (180 + Int(Atn(Vys / Sir) * 180 * 100 / Pi) /
100 * PlusMinus(Vys)) * PlusMinus(Vys)
                End If
            End If
            If Not Uhel2 = Abs(Uhel2) Then Uhel2 = 360 + Uhel2

            If Uhel < Uhel2 Then
                mdiMain.lbU.Caption = "U: " & Uhel2 - Uhel
            ElseIf Uhel2 < Uhel Then
                mdiMain.lbU.Caption = "U: " & Uhel - Uhel2
            Else
                mdiMain.lbU.Caption = "U: 0"
            End If
        End If
        Exit Sub
    End If

'======Plechovka barvy==========
    If Vyplnuj = True Then
        GetKeyboardState kbArray
```

```
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picOb.MouseIcon =
imIco.ListImages(10).Picture         Else         picOb.MouseIcon         =
imIco.ListImages(5).Picture
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picZoom.MouseIcon =
imIco.ListImages(10).Picture        Else        picZoom.MouseIcon        =
imIco.ListImages(5).Picture
        Exit Sub
    End If

'======Štetec==================
    If Stetcuji = True Then
        If shVyb.Visible = True Then
            If mdiMain.chKo.Value = 1 Then
                Vypocti Int(X1 - picTemp.Left), Int(Y1 - picTemp.Top),
Int(X - picTemp.Left), Int(Y - picTemp.Top), 1
            Else
                NakSte picTemp.hdc, X - picTemp.Left, Y - picTemp.Top,
Int(frmSte.txTl.Text / 2)
            End If
        Else
            If mdiMain.chKo.Value = 1 Then
                Vypocti Int(X1), Int(Y1), Int(X), Int(Y), 0
            Else
                NakSte picOb.hdc, X, Y, Int(frmSte.txTl.Text / 2)
            End If
        End If
        X1 = X: Y1 = Y
    End If
    If Stetcuj = True Then
        GetKeyboardState kbArray
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picOb.MouseIcon =
imIco.ListImages(10).Picture         Else         picOb.MouseIcon         =
imIco.ListImages(11).Picture
        shSte.Height = frmSte.txTl.Text: shSte.Width = frmSte.txTl.Text
        shSte.Left = X - Int(shSte.Height / 2): shSte.Top = Y -
Int(shSte.Height / 2)
        shSte.Visible = True
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picZoom.MouseIcon =
imIco.ListImages(10).Picture        Else        picZoom.MouseIcon        =
imIco.ListImages(11).Picture
        shZSte.Height = frmSte.txTl.Text: shZSte.Width = frmSte.txTl.Text
        shZSte.Left = X - Int(shZSte.Height / 2): shZSte.Top = Y -
Int(shZSte.Height / 2)
        shZSte.Visible = True
        Exit Sub
    End If

'======Guma====================
    If Gumuji = True Then
        If shVyb.Visible = True Then
            picTemp.Line (X1 - picTemp.Left, Y1 - picTemp.Top)-(X -
picTemp.Left, Y - picTemp.Top), GuBar
        Else
            picOb.Line (X1, Y1)-(X, Y), GuBar
        End If
        X1 = X: Y1 = Y
    End If
    If Gumuj = True Then
        GetKeyboardState kbArray
```

```
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picOb.MouseIcon =
imIco.ListImages(10).Picture         Else         picOb.MouseIcon         =
imIco.ListImages(11).Picture
        shSte.Height = frmSte.txTl.Text: shSte.Width = frmSte.txTl.Text
        shSte.Left = X - Int(shSte.Height / 2): shSte.Top = Y -
Int(shSte.Height / 2)
        shSte.Visible = True
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picZoom.MouseIcon =
imIco.ListImages(10).Picture        Else        picZoom.MouseIcon        =
imIco.ListImages(11).Picture
        shZSte.Height = frmSte.txTl.Text: shZSte.Width = frmSte.txTl.Text
        shZSte.Left = X - Int(shZSte.Height / 2): shZSte.Top = Y -
Int(shZSte.Height / 2)
        shZSte.Visible = True
        Exit Sub
    End If

'======Obdelnik=================
    If Obdelnikuji = True Then
        Tloustka = Int(mdiMain.txTl.Text)
        Sir = Abs(X - X1): Vys = Abs(Y - Y1)
        If shVyb.Visible = True Then
            picTemp.Cls
            If klShift = True Then
                If klCtrl = False Then
                    picTemp.Line (X1 - Sir + Tloustka / 2 - picTemp.Left,
Y1 - Vys + Tloustka / 2 - picTemp.Top)-(X1 + Sir - Tloustka / 2 -
picTemp.Left,   Y1   +   Vys   -   Tloustka   /   2   -   picTemp.Top),
frmNas.picPop.BackColor, B
                Else
                    picTemp.Line (X1 - Min(Sir, Vys) + Tloustka / 2 -
picTemp.Left, Y1 - Min(Sir, Vys) + Tloustka / 2 - picTemp.Top)-(X1 +
Min(Sir, Vys) - Tloustka / 2 - picTemp.Left, Y1 + Min(Sir, Vys) - Tloustka
/ 2 - picTemp.Top), frmNas.picPop.BackColor, B
                End If
            Else
                If X = X1 Then X = X1 + 1
                If Y = Y1 Then Y = Y1 + 1
                If klCtrl = False Then
                    picTemp.Line (X1 + (Tloustka / 2 * Abs(X - X1) / (X -
X1)) - picTemp.Left, Y1 + (Tloustka / 2 * Abs(Y - Y1) / (Y - Y1)) -
picTemp.Top)-(X - (Tloustka / 2 * Abs(X - X1) / (X - X1)) - picTemp.Left, Y
-  (Tloustka  /  2  *  Abs(Y  -  Y1)  /  (Y  -  Y1))  -  picTemp.Top),
frmNas.picPop.BackColor, B
                Else
                    picTemp.Line (X1 + ((Min(Sir, Vys) - Tloustka / 2) *
Abs(X - X1) / (X - X1)) - picTemp.Left, Y1 + ((Min(Sir, Vys) - Tloustka /
2) * Abs(Y - Y1) / (Y - Y1)) - picTemp.Top)-(X1 + (Tloustka / 2 * Abs(X -
X1) / (X - X1)) - picTemp.Left, Y1 + (Tloustka / 2 * Abs(Y - Y1) / (Y -
Y1)) - picTemp.Top), frmNas.picPop.BackColor, B
                End If
            End If
        Else
            picOb.Cls
            If klShift = True Then
                If klCtrl = False Then
                    picOb.Line (X1 - Sir + Tloustka / 2, Y1 - Vys +
Tloustka / 2)-(X1 + Sir - Tloustka / 2, Y1 + Vys - Tloustka / 2),
frmNas.picPop.BackColor, B
```

```
                Else
                    picOb.Line (X1 - Min(Sir, Vys) + Tloustka / 2, Y1 -
Min(Sir, Vys) + Tloustka / 2)-(X1 + Min(Sir, Vys) - Tloustka / 2, Y1 +
Min(Sir, Vys) - Tloustka / 2), frmNas.picPop.BackColor, B
                End If
            Else
                If X = X1 Then X = X1 + 1
                If Y = Y1 Then Y = Y1 + 1
                If klCtrl = False Then
                    picOb.Line (X1 + (Tloustka / 2 * Abs(X - X1) / (X -
X1)), Y1 + (Tloustka / 2 * Abs(Y - Y1) / (Y - Y1)))-(X - (Tloustka / 2 *
Abs(X - X1) / (X - X1)), Y - (Tloustka / 2 * Abs(Y - Y1) / (Y - Y1))),
frmNas.picPop.BackColor, B
                Else
                    picOb.Line (X1 + ((Min(Sir, Vys) - Tloustka / 2) *
Abs(X - X1) / (X - X1)), Y1 + ((Min(Sir, Vys) - Tloustka / 2) * Abs(Y - Y1)
/ (Y - Y1)))-(X1 + (Tloustka / 2 * Abs(X - X1) / (X - X1)), Y1 + (Tloustka
/ 2 * Abs(Y - Y1) / (Y - Y1))), frmNas.picPop.BackColor, B
                End If
            End If
        End If
        Exit Sub
    End If
    If Obdelnikuj = True Then
        GetKeyboardState kbArray
        If  kbArray.kbByte(vbKeyCapital)  <>  0  Then  picOb.MouseIcon  =
imIco.ListImages(10).Picture         Else         picOb.MouseIcon         =
imIco.ListImages(11).Picture
        If  kbArray.kbByte(vbKeyCapital)  <>  0  Then  picZoom.MouseIcon  =
imIco.ListImages(10).Picture        Else        picZoom.MouseIcon        =
imIco.ListImages(11).Picture
        Exit Sub
    End If

'======Elipsa==================
    If Elipsuji = True Then
        Tloustka = Int(mdiMain.txTl.Text)
        If shVyb.Visible = True Then
            picTemp.Cls
            If klShift = True Then
                If X - X1 = 0 Then X = X1 + 1
                If Y - Y1 = 0 Then Y = Y1 + 1
                If Abs(X - X1) < Tloustka / 2 + 1 Then X = X1 + (Tloustka /
2 + 1) * Abs(X - X1) / (X - X1)
                If Abs(Y - Y1) < Tloustka / 2 + 1 Then Y = Y1 + (Tloustka /
2 + 1) * Abs(Y - Y1) / (Y - Y1)
                If klCtrl = False Then
                    If Abs(X - X1) > Abs(Y - Y1) Then
                        Pomer = Abs(Y - Y1) / Abs(X - X1)
                        picTemp.Circle    (X1    -    picTemp.Left,    Y1    -
picTemp.Top),  Abs(X  -  X1)  -  Tloustka  /  2,  frmNas.picPop.BackColor,  ,  ,
Pomer
                    Else
                        Pomer = Abs(Y - Y1) / Abs(X - X1)
                        picTemp.Circle    (X1    -    picTemp.Left,    Y1    -
picTemp.Top),  Abs(Y  -  Y1)  -  Tloustka  /  2,  frmNas.picPop.BackColor,  ,  ,
Pomer
                    End If
                Else
```

```
                    If Abs(X - X1) > Abs(Y - Y1) Then
                        picTemp.Circle    (X1    -    picTemp.Left,    Y1    -
picTemp.Top), Abs(Y - Y1) - Tloustka / 2, frmNas.picPop.BackColor
                    Else
                        picTemp.Circle    (X1    -    picTemp.Left,    Y1    -
picTemp.Top), Abs(X - X1) - Tloustka / 2, frmNas.picPop.BackColor
                    End If
                End If
            Else
                If X = X1 Then X = X1 + 1
                If Y = Y1 Then Y = Y1 + 1
                If Abs(X - X1) < Tloustka + 1 Then X = X1 + (Tloustka + 1)
* Abs(X - X1) / (X - X1)
                If Abs(Y - Y1) < Tloustka + 1 Then Y = Y1 + (Tloustka + 1)
* Abs(Y - Y1) / (Y - Y1)
                If klCtrl = False Then
                    If Abs(X - X1) > Abs(Y - Y1) Then
                        Pomer = Abs(Y - Y1) / Abs(X - X1)
                        picTemp.Circle  ((X1  +  X)  /  2  -  picTemp.Left  +
(Tloustka  /  4  *  Abs(X  -  X1)  /  (X  -  X1)),  (Y1  +  Y)  /  2  -  picTemp.Top  +
(Tloustka / 4 * Abs(Y - Y1) / (Y - Y1))), Abs(X - X1) / 2 - Tloustka / 2,
frmNas.picPop.BackColor, , , Pomer
                    Else
                        Pomer = Abs(Y - Y1) / Abs(X - X1)
                        picTemp.Circle  ((X1  +  X)  /  2  -  picTemp.Left  +
(Tloustka  /  4  *  Abs(X  -  X1)  /  (X  -  X1)),  (Y1  +  Y)  /  2  -  picTemp.Top  +
(Tloustka / 4 * Abs(Y - Y1) / (Y - Y1))), Abs(Y - Y1) / 2 - Tloustka / 2,
frmNas.picPop.BackColor, , , Pomer
                    End If
                Else
                    If Abs(X - X1) > Abs(Y - Y1) Then
                        If (Abs(X - X1) = X - X1 And Abs(Y - Y1) = Y - Y1)
Or (Abs(X - X1) <> X - X1 And Abs(Y - Y1) <> Y - Y1) Then
                            picTemp.Circle   (X1   +   (Y   -   Y1)   /   2   -
picTemp.Left + (Tloustka / 4 * Abs(X - X1) / (X - X1)), Y1 + (Y - Y1) / 2 -
picTemp.Top + (Tloustka / 4 * Abs(Y - Y1) / (Y - Y1))), Abs(Y - Y1) / 2 -
Tloustka / 2, frmNas.picPop.BackColor
                        Else
                            picTemp.Circle   (X1   -   (Y   -   Y1)   /   2   -
picTemp.Left + (Tloustka / 4 * Abs(X - X1) / (X - X1)), Y1 + (Y - Y1) / 2 -
picTemp.Top + (Tloustka / 4 * Abs(Y - Y1) / (Y - Y1))), Abs(Y - Y1) / 2 -
Tloustka / 2, frmNas.picPop.BackColor
                        End If
                    Else
                        If (Abs(X - X1) = X - X1 And Abs(Y - Y1) = Y - Y1)
Or (Abs(X - X1) <> X - X1 And Abs(Y - Y1) <> Y - Y1) Then
                            picTemp.Circle   (X1   +   (X   -   X1)   /   2   -
picTemp.Left + (Tloustka / 4 * Abs(X - X1) / (X - X1)), Y1 + (X - X1) / 2 -
picTemp.Top + (Tloustka / 4 * Abs(Y - Y1) / (Y - Y1))), Abs(X - X1) / 2 -
Tloustka / 2, frmNas.picPop.BackColor
                        Else
                            picTemp.Circle   (X1   +   (X   -   X1)   /   2   -
picTemp.Left + (Tloustka / 4 * Abs(X - X1) / (X - X1)), Y1 - (X - X1) / 2 -
picTemp.Top + (Tloustka / 4 * Abs(Y - Y1) / (Y - Y1))), Abs(X - X1) / 2 -
Tloustka / 2, frmNas.picPop.BackColor
                        End If
                    End If
                End If
            End If
```

```
        Else
            picOb.Cls
            If klShift = True Then
                If X - X1 = 0 Then X = X1 + 1
                If Y - Y1 = 0 Then Y = Y1 + 1
                If Abs(X - X1) < Tloustka / 2 + 1 Then X = X1 + (Tloustka /
2 + 1) * Abs(X - X1) / (X - X1)
                If Abs(Y - Y1) < Tloustka / 2 + 1 Then Y = Y1 + (Tloustka /
2 + 1) * Abs(Y - Y1) / (Y - Y1)
                If klCtrl = False Then
                    If Abs(X - X1) > Abs(Y - Y1) Then
                        Pomer = Abs(Y - Y1) / Abs(X - X1)
                        picOb.Circle (X1, Y1), Abs(X - X1) - Tloustka / 2,
frmNas.picPop.BackColor, , , Pomer
                    Else
                        Pomer = Abs(Y - Y1) / Abs(X - X1)
                        picOb.Circle (X1, Y1), Abs(Y - Y1) - Tloustka / 2,
frmNas.picPop.BackColor, , , Pomer
                    End If
                Else
                    If Abs(X - X1) > Abs(Y - Y1) Then
                        picOb.Circle (X1, Y1), Abs(Y - Y1) - Tloustka / 2,
frmNas.picPop.BackColor
                    Else
                        picOb.Circle (X1, Y1), Abs(X - X1) - Tloustka / 2,
frmNas.picPop.BackColor
                    End If
                End If
            Else
                If X = X1 Then X = X1 + 1
                If Y = Y1 Then Y = Y1 + 1
                If Abs(X - X1) < Tloustka + 1 Then X = X1 + (Tloustka + 1)
* Abs(X - X1) / (X - X1)
                If Abs(Y - Y1) < Tloustka + 1 Then Y = Y1 + (Tloustka + 1)
* Abs(Y - Y1) / (Y - Y1)
                If klCtrl = False Then
                    If Abs(X - X1) > Abs(Y - Y1) Then
                        Pomer = Abs(Y - Y1) / Abs(X - X1)
                        picOb.Circle ((X1 + X) / 2 + (Tloustka / 4 * Abs(X
- X1) / (X - X1)), (Y1 + Y) / 2 + (Tloustka / 4 * Abs(Y - Y1) / (Y - Y1))),
Abs(X - X1) / 2 - Tloustka / 2, frmNas.picPop.BackColor, , , Pomer
                    Else
                        Pomer = Abs(Y - Y1) / Abs(X - X1)
                        picOb.Circle ((X1 + X) / 2 + (Tloustka / 4 * Abs(X
- X1) / (X - X1)), (Y1 + Y) / 2 + (Tloustka / 4 * Abs(Y - Y1) / (Y - Y1))),
Abs(Y - Y1) / 2 - Tloustka / 2, frmNas.picPop.BackColor, , , Pomer
                    End If
                Else
                    If Abs(X - X1) > Abs(Y - Y1) Then
                        If (Abs(X - X1) = X - X1 And Abs(Y - Y1) = Y - Y1)
Or (Abs(X - X1) <> X - X1 And Abs(Y - Y1) <> Y - Y1) Then
                            picOb.Circle (X1 + (Y - Y1) / 2 + (Tloustka / 4
* Abs(X - X1) / (X - X1)), Y1 + (Y - Y1) / 2 + (Tloustka / 4 * Abs(Y - Y1)
/ (Y - Y1))), Abs(Y - Y1) / 2 - Tloustka / 2, frmNas.picPop.BackColor
                        Else
                            picOb.Circle (X1 - (Y - Y1) / 2 + (Tloustka / 4
* Abs(X - X1) / (X - X1)), Y1 + (Y - Y1) / 2 + (Tloustka / 4 * Abs(Y - Y1)
/ (Y - Y1))), Abs(Y - Y1) / 2 - Tloustka / 2, frmNas.picPop.BackColor
                        End If
```

```
                    Else
                        If (Abs(X - X1) = X - X1 And Abs(Y - Y1) = Y - Y1)
Or (Abs(X - X1) <> X - X1 And Abs(Y - Y1) <> Y - Y1) Then
                            picOb.Circle (X1 + (X - X1) / 2 + (Tloustka / 4
* Abs(X - X1) / (X - X1)), Y1 + (X - X1) / 2 + (Tloustka / 4 * Abs(Y - Y1)
/ (Y - Y1))), Abs(X - X1) / 2 - Tloustka / 2, frmNas.picPop.BackColor
                        Else
                            picOb.Circle (X1 + (X - X1) / 2 + (Tloustka / 4
* Abs(X - X1) / (X - X1)), Y1 - (X - X1) / 2 + (Tloustka / 4 * Abs(Y - Y1)
/ (Y - Y1))), Abs(X - X1) / 2 - Tloustka / 2, frmNas.picPop.BackColor
                        End If
                    End If
                End If
            End If
        End If
        Exit Sub
    End If
    If Elipsuj = True Then
        GetKeyboardState kbArray
        If  kbArray.kbByte(vbKeyCapital)  <>  0  Then  picOb.MouseIcon  =
imIco.ListImages(10).Picture         Else         picOb.MouseIcon         =
imIco.ListImages(11).Picture
        If  kbArray.kbByte(vbKeyCapital)  <>  0  Then  picZoom.MouseIcon  =
imIco.ListImages(10).Picture        Else        picZoom.MouseIcon        =
imIco.ListImages(11).Picture
        Exit Sub
    End If

'======Cára=====================
    If Caruji = True Then
        If shVyb.Visible = True Then
            picTemp.Cls
            If Shift = 1 Then
                CaraXY X, Y, X1, Y1, 1
            Else
                picTemp.Line (X1 - picTemp.Left, Y1 - picTemp.Top)-(X -
picTemp.Left, Y - picTemp.Top), frmNas.picPop.BackColor
            End If
        Else
            picOb.Cls
            If Shift = 1 Then
                CaraXY X, Y, X1, Y1, 0
            Else
                picOb.Line (X1, Y1)-(X, Y), frmNas.picPop.BackColor
            End If
        End If
        Exit Sub
    End If
    If Caruj = True Then
        GetKeyboardState kbArray
        If  kbArray.kbByte(vbKeyCapital)  <>  0  Then  picOb.MouseIcon  =
imIco.ListImages(10).Picture         Else         picOb.MouseIcon         =
imIco.ListImages(11).Picture
        If  kbArray.kbByte(vbKeyCapital)  <>  0  Then  picZoom.MouseIcon  =
imIco.ListImages(10).Picture        Else        picZoom.MouseIcon        =
imIco.ListImages(11).Picture
        Exit Sub
    End If
```

```
'======Tužka====================
    If Maluji = True Then
        If shVyb.Visible = True Then
            picTemp.Line (X1 - picTemp.Left, Y1 - picTemp.Top)-(X -
picTemp.Left, Y - picTemp.Top), frmNas.picPop.BackColor
        Else
            picOb.Line (X1, Y1)-(X, Y), frmNas.picPop.BackColor
        End If
        X1 = X: Y1 = Y
    End If
    If Maluj = True Then
        GetKeyboardState kbArray
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picOb.MouseIcon =
imIco.ListImages(10).Picture Else picOb.MouseIcon =
imIco.ListImages(11).Picture
        shSte.Height = frmSte.txTl.Text: shSte.Width = frmSte.txTl.Text
        shSte.Left = X - Int(shSte.Height / 2): shSte.Top = Y -
Int(shSte.Height / 2)
        shSte.Visible = True
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picZoom.MouseIcon =
imIco.ListImages(10).Picture Else picZoom.MouseIcon =
imIco.ListImages(11).Picture
        shZSte.Height = frmSte.txTl.Text: shZSte.Width = frmSte.txTl.Text
        shZSte.Left = X - Int(shZSte.Height / 2): shZSte.Top = Y -
Int(shZSte.Height / 2)
        shZSte.Visible = True
        Exit Sub
    End If

'======Kapátko==================
    Dim Barva

    If Kapatko = True And Button = vbLeftButton Then
            If X > picOb.ScaleWidth Or X < 0 Or Y > picOb.ScaleHeight Or Y
< 0 Then Exit Sub
            Select Case TypKap
                Case 0
                    Barva = GetPixel(picOb.hdc, X, Y)
                Case 1
                    Barva = Kap3(X, Y)
                Case 2
                    Barva = Kap5(X, Y)
                Case 3
                    MsgBox "Chyba výběru typu kapátka"
            End Select
            frmNas.picPop.BackColor = Barva
        Exit Sub
    End If

    If Kapatko = True Then
        GetKeyboardState kbArray
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picOb.MouseIcon =
imIco.ListImages(10).Picture Else picOb.MouseIcon =
imIco.ListImages(8).Picture
        If kbArray.kbByte(vbKeyCapital) <> 0 Then picZoom.MouseIcon =
imIco.ListImages(10).Picture Else picZoom.MouseIcon =
imIco.ListImages(8).Picture
    End If
```

```
'======Presun====================
    If Presouvej = True Then
       picOb.MouseIcon = imIco.ListImages(1).Picture
       picZoom.MouseIcon = imIco.ListImages(1).Picture
    End If

    If Presouvam = True And shVyb.Visible = True Then
        picOb.Cls
        picOb.PaintPicture picTemp.Picture, X - StartX + 1, Y - StartY + 1
        shVyb.Left = X - StartX: shVyb.Top = Y - StartY
        Exit Sub
    End If

'======Výber====================
    If Vyber = True Then
        GetKeyboardState kbArray
        If  kbArray.kbByte(vbKeyCapital)  <>  0  Then  picOb.MouseIcon  =
imIco.ListImages(10).Picture         Else         picOb.MouseIcon         =
imIco.ListImages(11).Picture
        If  kbArray.kbByte(vbKeyCapital)  <>  0  Then  picZoom.MouseIcon  =
imIco.ListImages(10).Picture        Else        picZoom.MouseIcon        =
imIco.ListImages(11).Picture
    End If

    If Vybiram = True Then
        HiX = X: HiY = Y
        If HiX < 0 Then HiX = 0
        If HiY < 0 Then HiY = 0
        If HiX > picOb.ScaleWidth - 1 Then HiX = picOb.ScaleWidth - 1
        If HiY > picOb.ScaleHeight - 1 Then HiY = picOb.ScaleHeight - 1
        If LoX < 0 Then LoX = 0
        If LoY < 0 Then LoY = 0
        If LoX > picOb.ScaleWidth - 1 Then LoX = picOb.ScaleWidth - 1
        If LoY > picOb.ScaleHeight - 1 Then LoY = picOb.ScaleHeight - 1

        shVyb.Width = Abs(HiX - LoX) + 1
        shVyb.Height = Abs(HiY - LoY) + 1
        shVyb.Visible = True

        If HiX > LoX And HiY > LoY Then
            shVyb.Top = LoY: shVyb.Left = LoX
        End If
        If HiX > LoX And HiY < LoY Then
            shVyb.Top = HiY: shVyb.Left = LoX
        End If
        If HiX < LoX And HiY < LoY Then
            shVyb.Top = HiY: shVyb.Left = HiX
        End If
        If HiX < LoX And HiY > LoY Then
            shVyb.Top = LoY: shVyb.Left = HiX
        End If
        shZVyb.Left = shVyb.Left: shZVyb.Top = shVyb.Top
        shZVyb.Height = shVyb.Height: shZVyb.Width = shVyb.Width
        shZVyb.Visible = True
        Exit Sub
    End If
End Sub
```

```
Private Sub picOb_MouseUp(Button As Integer, Shift As Integer, X As Single,
Y As Single)
Dim I
Dim Vys, Sir, Uhel, Uhel2

'======Text====================
    If Textuji(Me.Tag) = True Then
        If TeVzato = True Then TeVzato = False
        If Ozn = True Then Ozn = False
        If Prv = True Then shSel.Visible = False
        If shSel.Visible = False Then tmrKur.Enabled = True
    End If

'======Pravítko=================
    If Merim = True Then
        Merim = False
        imM(RohM).Left = Int(X) + 0.49 - 3 * (picOb.ScaleHeight /
(picOb.Height - 2)): imM(RohM).Top = Int(Y) + 0.49 - 3 * (picOb.ScaleHeight
/ (picOb.Height - 2))
        If KonM = 1 Then
            lnM(CisM).X2 = Int(X) + 0.49: lnM(CisM).Y2 = Int(Y) + 0.49
            CarM(CisM).X2 = Int(X) + 0.49: CarM(CisM).Y2 = Int(Y) + 0.49
        Else
            lnM(CisM).X1 = Int(X) + 0.49: lnM(CisM).Y1 = Int(Y) + 0.49
            CarM(CisM).X1 = Int(X) + 0.49: CarM(CisM).Y1 = Int(Y) + 0.49
        End If

        Sir = Abs(lnM(CisM).X1 - lnM(CisM).X2)
        Vys = Abs(lnM(CisM).Y1 - lnM(CisM).Y2)
        mdiMain.lbS(CisM).Caption = "Š" & CisM + 1 & ": " & Int(Sir)
        mdiMain.lbV(CisM).Caption = "V" & CisM + 1 & ": " & Int(Vys)
        mdiMain.lbD(CisM).Caption = "D" & CisM + 1 & ": " & Int(((Sir ^ 2)
+ (Vys ^ 2)) ^ (0.5) * 100) / 100

        If CarM(0).Visible = True And CarM(1).Visible = False Then
            Sir = CarM(0).X2 - CarM(0).X1
            Vys = CarM(0).Y1 - CarM(0).Y2
            If Sir = 0 Then
                mdiMain.lbU.Caption = "U: " & 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    mdiMain.lbU.Caption = "U: " & Int(Atn(Vys / Sir) * 180
* 100 / Pi) / 100
                Else
                    mdiMain.lbU.Caption = "U: " & (180 + Int(Atn(Vys / Sir)
* 180 * 100 / Pi) / 100 * PlusMinus(Vys)) * PlusMinus(Vys)
                End If
            End If
        ElseIf CarM(0).Visible = True And CarM(1).Visible = True Then
            Sir = CarM(0).X2 - CarM(0).X1
            Vys = CarM(0).Y1 - CarM(0).Y2
            If Sir = 0 Then
                Uhel = 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    Uhel = Int(Atn(Vys / Sir) * 180 * 100 / Pi) / 100
                Else
                    Uhel = (180 + Int(Atn(Vys / Sir) * 180 * 100 / Pi) /
100 * PlusMinus(Vys)) * PlusMinus(Vys)
```

```
                End If
            End If
            If Not Uhel = Abs(Uhel) Then Uhel = 360 + Uhel

            Sir = CarM(1).X2 - CarM(1).X1
            Vys = CarM(1).Y1 - CarM(1).Y2
            If Sir = 0 Then
                Uhel2 = 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    Uhel2 = Int(Atn(Vys / Sir) * 180 * 100 / Pi) / 100
                Else
                    Uhel2 = (180 + Int(Atn(Vys / Sir) * 180 * 100 / Pi) /
100 * PlusMinus(Vys)) * PlusMinus(Vys)
                End If
            End If
            If Not Uhel2 = Abs(Uhel2) Then Uhel2 = 360 + Uhel2

            If Uhel < Uhel2 Then
                mdiMain.lbU.Caption = "U: " & Uhel2 - Uhel
            ElseIf Uhel2 < Uhel Then
                mdiMain.lbU.Caption = "U: " & Uhel - Uhel2
            Else
                mdiMain.lbU.Caption = "U: 0"
            End If
        End If
        Exit Sub
    End If

'======Štetec==================
    If Stetcuji = True Then
        If shVyb.Visible = True Then
            Vykresli
        End If
        Stetcuji = False
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Štětec": HistInd(Undo + 1) = 6
        UpdZpet
        StX = X: StY = Y
        Exit Sub
    End If

    If Stetcuj = True And Button = vbRightButton Then
        frmSte.Left = mdiMain.Left + (X * Screen.TwipsPerPixelX) +
(picOb.Left * Screen.TwipsPerPixelX) + frmOb.Left
        frmSte.Top = mdiMain.Top + (Y * Screen.TwipsPerPixelY) + (picOb.Top
* Screen.TwipsPerPixelY) + mdiMain.Height - mdiMain.ScaleHeight + frmOb.Top
        frmSte.Show 0, mdiMain
    End If

'======Guma====================
    If Gumuji = True Then
        If shVyb.Visible = True Then
            Vykresli
        End If
        Gumuji = False
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Guma": HistInd(Undo + 1) = 8
        UpdZpet
```

```
        GuX = X: GuY = Y
        Exit Sub
    End If

    If Gumuj = True And Button = vbRightButton Then
        frmSte.Left  =  mdiMain.Left  +  (X  *  Screen.TwipsPerPixelX)  +
(picOb.Left * Screen.TwipsPerPixelX) + frmOb.Left
        frmSte.Top = mdiMain.Top + (Y * Screen.TwipsPerPixelY) + (picOb.Top
* Screen.TwipsPerPixelY) + mdiMain.Height - mdiMain.ScaleHeight + frmOb.Top
        frmSte.Show 0, mdiMain
    End If

'======Obdelnik=================
    If Obdelnikuji = True Then
        If shVyb.Visible = True Then
            Vykresli
        End If
        Obdelnikuji = False
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Obdélník": HistInd(Undo + 1) = 13
        UpdZpet
        Exit Sub
    End If

'======Elipsa==================
    If Elipsuji = True Then
        If shVyb.Visible = True Then
            Vykresli
        End If
        Elipsuji = False
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Elipsa": HistInd(Undo + 1) = 12
        UpdZpet
        Exit Sub
    End If

'======Cára====================
    If Caruji = True Then
        If shVyb.Visible = True Then
            Vykresli
        End If
        Caruji = False
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Čára": HistInd(Undo + 1) = 11
        UpdZpet
        Exit Sub
    End If

'======Tuzka==================
    If Maluji = True Then
        If shVyb.Visible = True Then
            Vykresli
        End If
        Maluji = False
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Tužka": HistInd(Undo + 1) = 5
        UpdZpet
        TuX = X: TuY = Y
        Exit Sub
```

```
    End If

    If Maluj = True And Button = vbRightButton Then
        frmSte.Left = mdiMain.Left + (X * Screen.TwipsPerPixelX) +
(picOb.Left * Screen.TwipsPerPixelX) + frmOb.Left
        frmSte.Top = mdiMain.Top + (Y * Screen.TwipsPerPixelY) + (picOb.Top
* Screen.TwipsPerPixelY) + mdiMain.Height - mdiMain.ScaleHeight + frmOb.Top
        frmSte.Show 0, mdiMain
    End If

'======Kapátko==================
    Dim Barva

    If Kapatko = True And Button = vbLeftButton Then
        If X > picOb.ScaleWidth Or X < 0 Or Y > picOb.ScaleHeight Or Y < 0
Then Exit Sub
            Select Case TypKap
                Case 0
                    Barva = GetPixel(picOb.hdc, X, Y)
                Case 1
                    Barva = Kap3(X, Y)
                Case 2
                    Barva = Kap5(X, Y)
                Case 3
                    MsgBox "Chyba výběru typu kapátka"
            End Select
            frmNas.picPop.BackColor = Barva
        Exit Sub
        Exit Sub
    End If

'======Presun==================
    If Presouvam = True Then
        Presouvam = False
        Exit Sub
    End If

'======Výber==================
    If Vybiram = True Then
        Vybiram = False
        If shVyb.Width < 3 And shVyb.Height < 3 Then mnuUpOd_Click: Exit
Sub
        If shVyb.Left < 0 Then shVyb.Left = 0
        If shVyb.Top < 0 Then shVyb.Top = 0
        If shVyb.Left + shVyb.Width > picOb.ScaleWidth Then shVyb.Width =
picOb.ScaleWidth - shVyb.Left - 2
        If shVyb.Top + shVyb.Height > picOb.ScaleHeight Then shVyb.Height =
picOb.ScaleHeight - shVyb.Top - 2
        shVyb.Left = shVyb.Left - 1: shVyb.Top = shVyb.Top - 1
        shVyb.Width = shVyb.Width + 2: shVyb.Height = shVyb.Height + 2
        shZVyb.Left = shZVyb.Left: shZVyb.Top = shZVyb.Top
        shZVyb.Width = shZVyb.Width: shZVyb.Height = shZVyb.Height
        picClip.Picture = picOb.Picture
        picClip.ClipX = shVyb.Left + 1: picClip.ClipY = shVyb.Top + 1
        picClip.ClipWidth = shVyb.Width - 2: picClip.ClipHeight =
shVyb.Height - 2
        picTemp.Picture = picClip.Clip
        Presun(Me.Tag) = True
        Vzato = False
```

```
        Exit Sub
    End If
End Sub

Private Sub picZoom_KeyDown(KeyCode As Integer, Shift As Integer)
    picOb_KeyDown KeyCode, Shift
End Sub

Private Sub picZoom_KeyUp(KeyCode As Integer, Shift As Integer)
    picOb_KeyUp KeyCode, Shift
End Sub

Private Sub picZoom_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    picOb_MouseDown Button, Shift, Int(X), Int(Y)
    picZoom.PaintPicture    picOb.image,    0,    0,    picZoom.ScaleWidth,
picZoom.ScaleHeight
End Sub

Private Sub picZoom_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    picOb_MouseMove Button, Shift, Int(X), Int(Y)
    picZoom.PaintPicture    picOb.image,    0,    0,    picZoom.ScaleWidth,
picZoom.ScaleHeight
End Sub

Private Sub picZoom_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    picOb_MouseUp Button, Shift, Int(X), Int(Y)
    picZoom.PaintPicture    picOb.image,    0,    0,    picZoom.ScaleWidth,
picZoom.ScaleHeight
End Sub

Private Sub scbHor_Change()
    If FState(Me.Tag).Zoom = 100 Then
        picOb.Left = -scbHor.Value
        picStin.Left = picOb.Left + 3
    Else
        picZoom.Left = -scbHor.Value
        picStin.Left = picZoom.Left + 3
    End If
End Sub

Private Sub scbHor_Scroll()
    If FState(Me.Tag).Zoom = 100 Then
        picOb.Left = -scbHor.Value
        picStin.Left = picOb.Left + 3
    Else
        picZoom.Left = -scbHor.Value
        picStin.Left = picZoom.Left + 3
    End If
End Sub

Private Sub scbVer_Change()
    If FState(Me.Tag).Zoom = 100 Then
        picOb.Top = -scbVer.Value
        picStin.Top = picOb.Top + 3
    Else
        picZoom.Top = -scbVer.Value
```

```
        picStin.Top = picZoom.Top + 3
    End If
End Sub

Private Sub scbVer_Scroll()
    If FState(Me.Tag).Zoom = 100 Then
        picOb.Top = -scbVer.Value
        picStin.Top = picOb.Top + 3
    Else
        picZoom.Top = -scbVer.Value
        picStin.Top = picZoom.Top + 3
    End If
End Sub

Public Sub UpdZpet()
    Undo = Undo + 1
    picOb.Picture = picOb.image
    picOb.Cls
    ColUndo.Add picOb.Picture, CStr(Undo)
    FState(Int(Me.Tag)).Zmena = True
    mnuUpZp.Enabled = True
    mnuUpVp.Enabled = False
    For I = ColRedo.count To 1 Step -1
        ColRedo.Remove I
    Next I
End Sub

Public Sub Pryc()
    Presouvam = False: Vybiram = False
End Sub

Private Sub NastaVyb()
    On Error GoTo errChk

    picTemp.Picture = picClip.Clip
    picTemp.ScaleWidth = picClip.Width
    picTemp.Width = picTemp.ScaleWidth
    picTemp.ScaleHeight = picClip.Height
    picTemp.Height = picTemp.ScaleHeight
    picTemp.Left = shVyb.Left
    picTemp.Top = shVyb.Top
    picTemp.Visible = True
    DoEvents
    Exit Sub
errChk:
    MsgBox "In setupSelect: " & Err.Number & " " & Err.Description
    Resume Next
End Sub

Private Sub Smaz(DatDoHist As Boolean)
    Dim Mys

    Mys = picOb.MousePointer
    If Mys = 15 Then Mys = 2
    picOb.MousePointer = 11 'hodiny
    picDelete.Width = shVyb.Width
    picDelete.Height = shVyb.Height
    picDelete.BackColor = frmNas.picPoz.BackColor
```

```
    picDelete.Line  (0,  0)-(picDelete.ScaleWidth,  picDelete.ScaleHeight),
picDelete.BackColor, BF
    picDelete.Picture = picDelete.image
    picOb.PaintPicture picDelete.Picture, shVyb.Left + 1, shVyb.Top + 1
    If DatDoHist = True Then
        ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
        Hist(Undo + 1) = "Odstranění": HistInd(Undo + 1) = 19
        UpdZpet
    End If
    picOb.MousePointer = Mys
End Sub

Private Sub Vykresli()
    picTemp.Picture = picTemp.image
    picOb.PaintPicture picTemp.Picture, shVyb.Left + 1, shVyb.Top + 1
    picTemp.Visible = False
End Sub

Private Sub NastTemp()
    picTemp.Width = shVyb.Width - 2
    picTemp.ScaleWidth = picTemp.Width
    picTemp.Height = shVyb.Height - 2
    picTemp.ScaleHeight = picTemp.Height
    picTemp.Left = shVyb.Left + 1
    picTemp.Top = shVyb.Top + 1
    picTemp.Visible = True
End Sub

Private Function Kap3(X As Single, Y As Single)
Dim Re, Gr, Bl, Bar
Dim Xx, Yy

    For Xx = -1 To 1
        For Yy = -1 To 1
            Bar = GetPixel(picOb.hdc, X + Xx, Y + Yy)
            Re = Re + Int(Bar Mod 256)
            Gr = Gr + Int(Int(Bar / 256) Mod 256)
            Bl = Bl + Int(Bar / 256 / 256)
        Next Yy
    Next Xx

    Kap3 = RGB(Int(Re / 9), Int(Gr / 9), Int(Bl / 9))
End Function

Private Function Kap5(X As Single, Y As Single)
Dim Re, Gr, Bl, Bar
Dim Xx, Yy

    For Xx = -2 To 2
        For Yy = -2 To 2
            Bar = GetPixel(picOb.hdc, X + Xx, Y + Yy)
            Re = Re + Int(Bar Mod 256)
            Gr = Gr + Int(Int(Bar / 256) Mod 256)
            Bl = Bl + Int(Bar / 256 / 256)
        Next Yy
    Next Xx

    Kap5 = RGB(Int(Re / 25), Int(Gr / 25), Int(Bl / 25))
End Function
```

```
Private Sub CaraXY(X As Single, Y As Single, X1 As Single, Y1 As Single,
Vyb As Integer)
Dim DelX, DelY

    DelX = X - X1
    DelY = Y - Y1

    If Abs(DelY) > Abs(DelX) Then
        If Abs(DelY) / 2 < Abs(DelX) Then
            If (Abs(DelX) = DelX And Abs(DelY) = DelY) Or (Abs(DelX) <>
DelX And Abs(DelY) <> DelY) Then
                X = Y - Y1 + X1
            Else
                X = X1 - Y + Y1
            End If
        Else
            X = X1
        End If
    Else
        If Abs(DelX) / 2 < Abs(DelY) Then
            If (Abs(DelX) = DelX And Abs(DelY) = DelY) Or (Abs(DelX) <>
DelX And Abs(DelY) <> DelY) Then
                Y = X - X1 + Y1
            Else
                Y = Y1 - X + X1
            End If
        Else
            Y = Y1
        End If
    End If

    If Vyb = 1 Then
        picTemp.Line (X1, Y1)-(X, Y), frmNas.picPop.BackColor
    Else
        picOb.Line (X1, Y1)-(X, Y), frmNas.picPop.BackColor
    End If
End Sub

Private Function Min(Unu As Single, Du As Single)
    If Unu > Du Then Min = Du Else Min = Unu
End Function

Private Function Max(Unu As Single, Du As Single)
    If Unu > Du Then Max = Du Else Max = Unu
End Function

Private Sub Vypocti(X1 As Integer, Y1 As Integer, X2 As Integer, Y2 As
Integer, Temp As Single)
Dim Sir, Vys, Del, KrX, KrY, Stepp, I

    Sir = X2 - X1: Vys = Y2 - Y1
    If Sir = 0 Then Sir = 1
    If Vys = 0 Then Vys = 1
    Del = Int((Sir ^ 2 + Vys ^ 2) ^ (1 / 2))
    KrX = Sir / Del: KrY = Vys / Del

    Stepp = Int(frmSte.txTl.Text * 0.22)
    If Stepp < 1 Then Stepp = 1
```

```
    If Temp = 1 Then
        For I = 1 To Del Step Stepp
            NakSte  picTemp.hdc,  X1  +  KrX  *  I,  Y1  +  KrY  *  I,
Int(frmSte.txTl.Text / 2)
        Next I
    Else
        For I = 1 To Del Step Stepp
            NakSte  picOb.hdc,  X1  +  KrX  *  I,  Y1  +  KrY  *  I,
Int(frmSte.txTl.Text / 2)
        Next I
    End If
End Sub

Public Sub ZobMer()
Dim Sir, Vys, Uhel, Uhel2
    If CarM(0).Visible = False Then Exit Sub
    lnM(0).Visible = True: lnM(0).X1 = CarM(0).X1: lnM(0).X2 = CarM(0).X2:
lnM(0).Y1 = CarM(0).Y1: lnM(0).Y2 = CarM(0).Y2
    imM(0).Left = CarM(0).X1 - 3: imM(0).Top = CarM(0).Y1 - 3
    imM(1).Left = CarM(0).X2 - 3: imM(1).Top = CarM(0).Y2 - 3
    lnM(0).Visible = True: imM(0).Visible = True: imM(1).Visible = True

    If CarM(1).Visible = False Then GoTo Po
    lnM(1).Visible = True: lnM(1).X1 = CarM(1).X1: lnM(1).X2 = CarM(1).X2:
lnM(1).Y1 = CarM(1).Y1: lnM(1).Y2 = CarM(1).Y2
    imM(0).Left = lnM(1).X2 - 3: imM(0).Top = lnM(1).Y2 - 3
    shM.Left = lnM(1).X1 - 3: shM.Top = lnM(1).Y1 - 3: shM.Visible = True
Po:
        If CarM(0).Visible = True And CarM(1).Visible = False Then
            Sir = CarM(0).X2 - CarM(0).X1
            Vys = CarM(0).Y1 - CarM(0).Y2
            If Sir = 0 Then
                mdiMain.lbU.Caption = "U: " & 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    mdiMain.lbU.Caption = "U: " & Int(Atn(Vys / Sir) * 180
* 100 / Pi) / 100
                Else
                    mdiMain.lbU.Caption = "U: " & (180 + Int(Atn(Vys / Sir)
* 180 * 100 / Pi) / 100 * PlusMinus(Vys)) * PlusMinus(Vys)
                End If
            End If
        ElseIf CarM(0).Visible = True And CarM(1).Visible = True Then
            Sir = CarM(0).X2 - CarM(0).X1
            Vys = CarM(0).Y1 - CarM(0).Y2
            If Sir = 0 Then
                Uhel = 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    Uhel = Int(Atn(Vys / Sir) * 180 * 100 / Pi) / 100
                Else
                    Uhel = (180 + Int(Atn(Vys / Sir) * 180 * 100 / Pi) /
100 * PlusMinus(Vys)) * PlusMinus(Vys)
                End If
            End If
            If Not Uhel = Abs(Uhel) Then Uhel = 360 + Uhel

            Sir = CarM(1).X2 - CarM(1).X1
```

```
            Vys = CarM(1).Y1 - CarM(1).Y2
            If Sir = 0 Then
                Uhel2 = 90 * PlusMinus(Vys)
            Else
                If Sir > 0 Then
                    Uhel2 = Int(Atn(Vys / Sir) * 180 * 100 / Pi) / 100
                Else
                    Uhel2 = (180 + Int(Atn(Vys / Sir) * 180 * 100 / Pi) /
100 * PlusMinus(Vys)) * PlusMinus(Vys)
                End If
            End If
            If Not Uhel2 = Abs(Uhel2) Then Uhel2 = 360 + Uhel2

            If Uhel < Uhel2 Then
                mdiMain.lbU.Caption = "U: " & Uhel2 - Uhel
            ElseIf Uhel2 < Uhel Then
                mdiMain.lbU.Caption = "U: " & Uhel - Uhel2
            Else
                mdiMain.lbU.Caption = "U: 0"
            End If
        End If
End Sub

Public Sub HideMer()
Dim I

    For I = 0 To 1
        imM(I).Visible = False
        lnM(I).Visible = False
    Next I
    shM.Visible = False
End Sub

Public Sub SmazMer()
Dim I

    For I = 0 To 1
        imM(I).Visible = False
        lnM(I).Visible = False
        CarM(I).Visible = False
    Next I
    shM.Visible = False
End Sub

Private Function PlusMinus(Cislo)
    If Cislo = 0 Then PlusMinus = 1: Exit Function
    PlusMinus = Cislo / Abs(Cislo)
End Function

Public Sub Zoom()
Dim Temp
    'viz Nicka
End Sub

Public Sub DeZoom()
    picZoom.Visible = False
    picOb.Visible = True
    Form_Resize
End Sub
```

```
Public Sub Textni()
Dim I
Dim hBrush As Long, oldBrush As Long
    picOb.Cls
    Okolo
    lnKur.X1 = tStartX + picOb.TextWidth(Left(Text(Akt), AktZ)): lnKur.X2 =
lnKur.X1
    lnKur.Y1 = tStartY + Akt * picOb.TextHeight("X"): lnKur.Y2 = lnKur.Y1 +
picOb.TextHeight("X")
    If shSel.Visible = True Then
        If SelEn > SelSt Then
            shSel.Left = tStartX + picOb.TextWidth(Left(Text(Akt), SelSt))
            shSel.Width   =   picOb.TextWidth(Left(Text(Akt),   SelEn))   -
picOb.TextWidth(Left(Text(Akt), SelSt))
        Else
            shSel.Left = tStartX + picOb.TextWidth(Left(Text(Akt), SelEn))
            shSel.Width   =   picOb.TextWidth(Left(Text(Akt),   SelSt))   -
picOb.TextWidth(Left(Text(Akt), SelEn))
        End If
        shSel.Height = picOb.TextHeight("X")
    End If
    'Vytvoří nový brush
    hBrush = CreateSolidBrush(frmNas.picPop.BackColor)
    'Překreslí starý brush novým
    oldBrush = SelectObject(picOb.hdc, hBrush)
    'Nastaví barvu popředí na barvu obrysu
    picOb.ForeColor = mdiMain.picObr.BackColor
    'Začne novou cestu
    BeginPath picOb.hdc
    For I = 0 To Rad
        TextOut picOb.hdc, tStartX, tStartY + I * picOb.TextHeight("X"),
Text(I), Len(Text(I))
    Next I
    EndPath picOb.hdc
    'stroke the outline of the path by using the current pen and fill its
interior by using the current brush
    If JenObrys = True Then
        StrokePath picOb.hdc 'Pouze obtáhne cestu
    Else
        StrokeAndFillPath picOb.hdc 'Obtáhne a vyplní cestu
    End If
    'Přepíše nový brush zpět na starý
    SelectObject picOb.hdc, oldBrush
    'Smaže nový brush
    DeleteObject hBrush
    picOb.ForeColor = frmNas.picPop.BackColor
End Sub

Private Sub tmrKur_Timer()
    lnKur.X1 = tStartX + picOb.TextWidth(Left(Text(Akt), AktZ)): lnKur.X2 =
lnKur.X1
    lnKur.Y1 = tStartY + Akt * picOb.TextHeight("X"): lnKur.Y2 = lnKur.Y1 +
picOb.TextHeight("X")
    If lnKur.Visible = True Then lnKur.Visible = False Else lnKur.Visible =
True
End Sub

Private Sub Najdi(Doleva As Boolean)
```

```
Dim Temp, Bylo, Je

    If Doleva = True Then
        Temp = Mid(Text(Akt), AktZ, 1)
        AktZ = AktZ - 1
    Else
        Temp = Mid(Text(Akt), AktZ + 1, 1)
        AktZ = AktZ + 1
    End If
    Select Case Temp
        Case " "
            Bylo = 0
        Case ",", ".", "?", "!", "+", "-", "*", "/", "\", ":", ";", "_",
"§", "'", "(", ")", "=", "%", "°", "|", ">", "<", "&", "#", "@", "{", "}",
"[", "]"
            Bylo = 1
        Case Else
            Bylo = 2
    End Select

    Do
        If Doleva = True Then
            If AktZ = 0 Then Exit Sub
            Temp = Mid(Text(Akt), AktZ, 1)
            AktZ = AktZ - 1
        Else
            If AktZ = Len(Text(Akt)) Then Exit Sub
            Temp = Mid(Text(Akt), AktZ + 1, 1)
            AktZ = AktZ + 1
        End If

        Select Case Temp
            Case " "
                Je = 0
            Case ",", ".", "?", "!", "+", "-", "*", "/", "\", ":", ";",
"_", "§", "'", "(", ")", "=", "%", "°", "|", ">", "<", "&", "#", "@", "{",
"}", "[", "]"
                Je = 1
            Case Else
                Je = 2
        End Select

        If Not Bylo = Je Then
            If Doleva = True Then
                If Bylo = 0 And Je = 2 Then Bylo = 2: GoTo Dal
                If Bylo = 0 And Je = 1 Then Bylo = 1: GoTo Dal
            Else
                If Bylo = 2 And Je = 0 Then Bylo = 0: GoTo Dal
                If Bylo = 1 And Je = 0 Then Bylo = 0: GoTo Dal
            End If
            If Doleva = False Then AktZ = AktZ - 1 Else AktZ = AktZ + 1
            Exit Sub
        End If
Dal:
    Loop
End Sub

Private Sub Okolo()
Dim Del, I, Por
```

```
    shTe.Left = tStartX - 5: shTe.Top = tStartY - 5
    shTe.Height = picOb.TextHeight("X") * (Rad + 1) + 10
    Del = 0
    For I = 0 To Rad
        If picOb.TextWidth(Text(I)) > Del Then Por = I: Del =
picOb.TextWidth(Text(I))
    Next I
    shTe.Width = picOb.TextWidth(Text(Por)) + 10
End Sub

Private Sub Mys(X As Single, Y As Single)
Dim R, Z
    For R = 0 To Rad
        If Y > tStartY + picOb.TextHeight("X") * R And Y < tStartY +
picOb.TextHeight("X") * (R + 1) Then
            If X > tStartX And X < tStartX + picOb.TextWidth(Left(Text(R),
1)) / 2 Then
                Akt = R: AktZ = Z
                lnKur.X1 = tStartX + picOb.TextWidth(Left(Text(Akt),
AktZ)): lnKur.X2 = lnKur.X1
                lnKur.Y1 = tStartY + Akt * picOb.TextHeight("X"): lnKur.Y2
= lnKur.Y1 + picOb.TextHeight("X")
                lnKur.Visible = True
                Exit Sub
            End If
            For Z = 1 To Len(Text(R))
                If X > tStartX + (picOb.TextWidth(Left(Text(R), Z)) +
picOb.TextWidth(Left(Text(R), Z - 1))) / 2 And X < tStartX +
(picOb.TextWidth(Left(Text(R), Z)) + picOb.TextWidth(Left(Text(R), Z + 1)))
/ 2 Then
                    Akt = R: AktZ = Z
                    lnKur.X1 = tStartX + picOb.TextWidth(Left(Text(Akt),
AktZ)): lnKur.X2 = lnKur.X1
                    lnKur.Y1 = tStartY + Akt * picOb.TextHeight("X"):
lnKur.Y2 = lnKur.Y1 + picOb.TextHeight("X")
                    lnKur.Visible = True
                    Exit Sub
                End If
            Next Z
        End If
    Next R
End Sub

Public Sub DeTex()
    tmrKur.Enabled = False
    lnKur.Visible = False
    shSel.Visible = False
    shTe.Visible = False
    TeVzato = False
    Ozn = False
    Prv = False
End Sub

Public Sub PrvZpet()
    Undo = Undo + 1
    ColUndo.Add picOb.Picture, CStr(Undo)
End Sub
```

```
Private Sub Zpet()
    ColRedo.Add ColUndo.Item(ColUndo.count)
    ColUndo.Remove ColUndo.count
    picOb.Picture = ColUndo.Item(ColUndo.count)
    picOb.Refresh
    If ColUndo.count = 1 Then mnuUpZp.Enabled = False
    mnuUpVp.Enabled = True
End Sub

Private Sub Vpred()
    ColUndo.Add ColRedo.Item(ColRedo.count)
    picOb.Picture = ColRedo.Item(ColRedo.count)
    ColRedo.Remove ColRedo.count
    picOb.Refresh
    If ColRedo.count = 0 Then mnuUpVp.Enabled = False
    mnuUpZp.Enabled = True
End Sub

Public Sub MenuHist()

End Sub

Public Sub Pixelizace(Hod, Vyber As Boolean)
Dim Xpix, Ypix
    If Vyber = False Then
            frmTemp.picTemp.Picture = picOb.Picture
            frmTemp.picTemp.Height     =     picOb.ScaleHeight     +     2:
frmTemp.picTemp.Width = picOb.ScaleWidth + 2
        Else
            frmTemp.picTemp.Picture = picTemp.Picture
            frmTemp.picTemp.Height     =     picTemp.ScaleHeight     +     2:
frmTemp.picTemp.Width = picTemp.ScaleWidth + 2
    End If
    mdiMain.pb.Max          =          frmTemp.picTemp.ScaleWidth          *
frmTemp.picTemp.ScaleHeight
    mdiMain.pb.Value = 1: mdiMain.pb.Visible = True
    mdiMain.lbPb.Caption = "Pixelizace: ": mdiMain.lbPb.Visible = True

    For Ypix = 0 To frmTemp.picTemp.ScaleHeight - 1 Step Hod
        For Xpix = 0 To frmTemp.picTemp.ScaleWidth - 1 Step Hod
            frmTemp.picTemp.Line (Xpix, Ypix)-(Xpix + Hod - 1, Ypix + Hod -
1), frmTemp.picTemp.Point(Xpix, Ypix), BF
        Next Xpix
        If  Ypix  Mod  Hod  *  10  Then  DoEvents:  mdiMain.pb.Value  =  Ypix  *
frmTemp.picTemp.ScaleWidth
    Next Ypix

    frmTemp.picTemp.Picture = frmTemp.picTemp.image
    If Vyber = False Then
            picOb.PaintPicture frmTemp.picTemp.Picture, 0, 0
        Else
            picTemp.Picture = frmTemp.picTemp.Picture
            picOb.PaintPicture picTemp.Picture, shVyb.Left + 1, shVyb.Top +
1
    End If
    ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
    Hist(Undo + 1) = "Pixelizace": HistInd(Undo + 1) = 20
    UpdZpet
    mdiMain.pb.Visible = False: mdiMain.lbPb.Visible = False
```

```
End Sub

Public Sub Pointilizace(Polomer, Prah, Kanal, Invertovat, Vyber)
Dim PoiX, PoiX1, PoiY, PoiCX, PoiCY, PoiSou, PoiPru, PoiPol, PoiPolMax

    If Vyber = False Then
            frmTemp.picTemp.Picture = picOb.Picture
            frmTemp.picTemp.Height     =     picOb.ScaleHeight     +     2:
frmTemp.picTemp.Width = picOb.ScaleWidth + 2
        Else
            frmTemp.picTemp.Picture = picTemp.Picture
            frmTemp.picTemp.Height     =     picTemp.ScaleHeight     +     2:
frmTemp.picTemp.Width = picTemp.ScaleWidth + 2
    End If
    frmTemp.picTemp2.Picture = frmTemp.picTemp.Picture
    mdiMain.pb.Max          =          frmTemp.picTemp.ScaleHeight          *
frmTemp.picTemp.ScaleWidth
    mdiMain.pb.Value = 1: mdiMain.pb.Visible = True
    mdiMain.lbPb.Caption = "Pointilizace: ": mdiMain.lbPb.Visible = True
    frmTemp.picTemp2.FillColor = frmNas.picPop.BackColor

    frmTemp.picTemp2.Line    (0,    0)-(frmTemp.picTemp2.ScaleWidth    -    1,
frmTemp.picTemp2.ScaleHeight - 1), frmNas.picPoz.BackColor, BF
    PoiPolMax = (2 * (Polomer / 2) ^ 2) ^ 0.5
    For PoiY = 0 To frmTemp.picTemp.ScaleHeight - 1 Step Polomer
        For PoiX1 = 0 To frmTemp.picTemp.ScaleWidth - 1 Step Polomer
            If PoiY Mod (Polomer * 2) = 0 Then PoiX = PoiX1 Else PoiX =
PoiX1 + Polomer / 2
            PoiSou = 0: PoiPru = 0: PoiPol = 0
            For PoiCY = 0 To Polomer - 1
                For PoiCX = 0 To Polomer - 1
                    Select Case Kanal
                        Case "R"
                            PoiSou = PoiSou + frmTemp.picTemp.Point(PoiX +
PoiCX, PoiY + PoiCY) Mod 256
                        Case "G"
                            PoiSou = PoiSou + (frmTemp.picTemp.Point(PoiX +
PoiCX, PoiY + PoiCY) \ 256) Mod 256
                        Case "B"
                            PoiSou = PoiSou + frmTemp.picTemp.Point(PoiX +
PoiCX, PoiY + PoiCY) \ 65536
                        Case "P"
                            PoiSou = PoiSou + frmTemp.picTemp.Point(PoiX +
PoiCX, PoiY + PoiCY) \ 65536
                            PoiSou = PoiSou + (frmTemp.picTemp.Point(PoiX +
PoiCX, PoiY + PoiCY) \ 256) Mod 256
                            PoiSou = PoiSou + frmTemp.picTemp.Point(PoiX +
PoiCX, PoiY + PoiCY) Mod 256
                    End Select
                Next PoiCX
            Next PoiCY
            If Kanal = "P" Then
                    PoiPru = PoiSou / (Polomer ^ 2 * 3)
                Else
                    PoiPru = PoiSou / (Polomer ^ 2)
            End If
            PoiPol = IIf(Invertovat <> 1, PoiPolMax * (PoiPru - Prah) /
(255 - Prah), PoiPolMax * (Prah - PoiPru) / Prah)
            If PoiPol > 0 Then
```

```
                frmTemp.picTemp2.Circle (PoiX + (Polomer / 2), PoiY +
(Polomer / 2)), PoiPol, frmNas.picPop.BackColor
            End If
        Next PoiX1
        If PoiY Mod (5 * Polomer) = 0 Then DoEvents: mdiMain.pb.Value =
(PoiY + 1) * frmTemp.picTemp.ScaleWidth
    Next PoiY

    frmTemp.picTemp2.Picture = frmTemp.picTemp2.image
    If Vyber = False Then
            picOb.PaintPicture frmTemp.picTemp2.Picture, 0, 0
        Else
            picTemp.Picture = frmTemp.picTemp2.Picture
            picOb.PaintPicture picTemp.Picture, shVyb.Left + 1, shVyb.Top +
1
    End If
    ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
    Hist(Undo + 1) = "Pointilizace": HistInd(Undo + 1) = 21
    UpdZpet
    mdiMain.pb.Visible = False: mdiMain.lbPb.Visible = False
End Sub

Public Sub Solarizace(Sytost, Jas, Vyber)
Dim SolX, SolY, SolRGB, SolR, SolG, SolB, SolHSB, SolBar() As String
    If Vyber = False Then
            frmTemp.picTemp.Picture = picOb.Picture
            frmTemp.picTemp.Height = picOb.ScaleHeight + 2:
frmTemp.picTemp.Width = picOb.ScaleWidth + 2
        Else
            frmTemp.picTemp.Picture = picTemp.Picture
            frmTemp.picTemp.Height = picTemp.ScaleHeight + 2:
frmTemp.picTemp.Width = picTemp.ScaleWidth + 2
    End If
    mdiMain.pb.Max = frmTemp.picTemp.ScaleWidth *
frmTemp.picTemp.ScaleHeight
    mdiMain.pb.Value = 1: mdiMain.pb.Visible = True
    mdiMain.lbPb.Caption = "Solarizace: ": mdiMain.lbPb.Visible = True

    For SolY = 0 To frmTemp.picTemp.ScaleHeight - 1
        For SolX = 0 To frmTemp.picTemp.ScaleWidth - 1
            SolRGB = frmTemp.picTemp.Point(SolX, SolY)
            SolR = SolRGB Mod 256
            SolG = (SolRGB \ 256) Mod 256
            SolB = SolRGB \ 65536
            SolHSB = RGBtoHSB(Int(SolR), Int(SolG), Int(SolB))
            SolBar = Split(SolHSB, ";")
            SolBar(1) = 100: SolBar(2) = 100
            SolRGB = HSBtoRGB(Int(SolBar(0)), Int(SolBar(1)),
Int(SolBar(2)), "")
            SolBar = Split(SolRGB, ";")
            frmTemp.picTemp.ForeColor = RGB(Int(SolBar(0)), Int(SolBar(1)),
Int(SolBar(2)))
            frmTemp.picTemp.PSet (SolX, SolY)
        Next SolX
        If SolY Mod 50 Then DoEvents: mdiMain.pb.Value = SolY *
frmTemp.picTemp.ScaleWidth
    Next SolY

    frmTemp.picTemp.Picture = frmTemp.picTemp.image
```

```
    If Vyber = False Then
            picOb.PaintPicture frmTemp.picTemp.Picture, 0, 0
        Else
            picTemp.Picture = frmTemp.picTemp.Picture
            picOb.PaintPicture picTemp.Picture, shVyb.Left + 1, shVyb.Top +
1
    End If
    ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
    Hist(Undo + 1) = "Solarizace": HistInd(Undo + 1) = 22
    UpdZpet
    mdiMain.pb.Visible = False: mdiMain.lbPb.Visible = False
End Sub

Public Sub SmazHist()
    mnuUpZp.Enabled = False: mnuUpVp.Enabled = False
    For I = ColRedo.count To 1 Step -1
        ColRedo.Remove I
    Next I
    For I = ColUndo.count To 1 Step -1
        ColUndo.Remove I
    Next I
    Undo = 0
    ReDim Hist(0): ReDim HistInd(0)
End Sub

Public Sub Karikatura(Vyber, Obrys As Boolean)
Dim KarX, KarY, KarRGB, KarR, KarG, KarB, KarHSB, KarBar() As String, KarH
    If Vyber = False Then
            frmTemp.picTemp.Picture = picOb.Picture
            frmTemp.picTemp.Height     =     picOb.ScaleHeight     +     2:
frmTemp.picTemp.Width = picOb.ScaleWidth + 2
        Else
            frmTemp.picTemp.Picture = picTemp.Picture
            frmTemp.picTemp.Height     =     picTemp.ScaleHeight     +     2:
frmTemp.picTemp.Width = picTemp.ScaleWidth + 2
    End If
    mdiMain.pb.Max          =          frmTemp.picTemp.ScaleWidth          *
frmTemp.picTemp.ScaleHeight
    mdiMain.pb.Value = 1: mdiMain.pb.Visible = True
    mdiMain.lbPb.Caption = "Karikatura: ": mdiMain.lbPb.Visible = True

    For KarY = 0 To frmTemp.picTemp.ScaleHeight - 1
        For KarX = 0 To frmTemp.picTemp.ScaleWidth - 1
            KarRGB = frmTemp.picTemp.Point(KarX, KarY)
            KarR = KarRGB Mod 256
            KarG = (KarRGB \ 256) Mod 256
            KarB = KarRGB \ 65536
            KarHSB = RGBtoHSB(Int(KarR), Int(KarG), Int(KarB))
            KarBar = Split(KarHSB, ";")
            'KarBar(0) = (KarBar(0) \ 30) * 30
            If KarBar(0) > 359 Then KarBar(0) = KarBar(0) - 360
            If KarBar(2) > 92 Then
                    KarBar(2) = 92
                Else
                    KarBar(2) = ((KarBar(2)) \ 25) * 25 + 10
            End If
            KarRGB      =      HSBtoRGB(Int(KarBar(0)),      Int(KarBar(1)),
Int(KarBar(2)), "")
            KarBar = Split(KarRGB, ";")
```

```
            frmTemp.picTemp.ForeColor     =     RGB(KarBar(0),     KarBar(1),
KarBar(2))
            frmTemp.picTemp.PSet (KarX, KarY)
            If Obrys = True Then
                If KarX > 3 Then
                    KarRGB = frmTemp.picTemp.Point(KarX - 2, KarY)
                    KarR = KarRGB Mod 256
                    KarG = (KarRGB \ 256) Mod 256
                    KarB = KarRGB \ 65536
                    KarHSB = RGBtoHSB(Int(KarR), Int(KarG), Int(KarB))
                    KarBar = Split(KarHSB, ";")
                    KarH = KarBar(2)
                    KarRGB = frmTemp.picTemp.Point(KarX, KarY)
                    KarR = KarRGB Mod 256
                    KarG = (KarRGB \ 256) Mod 256
                    KarB = KarRGB \ 65536
                    KarHSB = RGBtoHSB(Int(KarR), Int(KarG), Int(KarB))
                    KarBar = Split(KarHSB, ";")
                    frmTemp.picTemp.ForeColor = vbBlack
                    If     Not     KarBar(2)     =     KarH     And     Not
frmTemp.picTemp.Point(KarX - 2, KarY) = 0 Then frmTemp.picTemp.PSet (KarX -
1, KarY)
                End If
                If KarY > 3 Then
                    KarRGB = frmTemp.picTemp.Point(KarX, KarY - 2)
                    KarR = KarRGB Mod 256
                    KarG = (KarRGB \ 256) Mod 256
                    KarB = KarRGB \ 65536
                    KarHSB = RGBtoHSB(Int(KarR), Int(KarG), Int(KarB))
                    KarBar = Split(KarHSB, ";")
                    KarH = KarBar(2)
                    KarRGB = frmTemp.picTemp.Point(KarX, KarY)
                    KarR = KarRGB Mod 256
                    KarG = (KarRGB \ 256) Mod 256
                    KarB = KarRGB \ 65536
                    KarHSB = RGBtoHSB(Int(KarR), Int(KarG), Int(KarB))
                    KarBar = Split(KarHSB, ";")
                    frmTemp.picTemp.ForeColor = vbBlack
                    If     Not     KarBar(2)     =     KarH     And     Not
frmTemp.picTemp.Point(KarX, KarY - 2) = 0 Then frmTemp.picTemp.PSet (KarX,
KarY - 1)
                End If
            End If
        Next KarX
        If  KarY  Mod  10  =  1  Then  DoEvents:  mdiMain.pb.Value  =  KarY  *
frmTemp.picTemp.ScaleWidth
    Next KarY

    frmTemp.picTemp.Picture = frmTemp.picTemp.image
    If Vyber = False Then
            picOb.PaintPicture frmTemp.picTemp.Picture, 0, 0
        Else
            picTemp.Picture = frmTemp.picTemp.Picture
            picOb.PaintPicture picTemp.Picture, shVyb.Left + 1, shVyb.Top +
1
    End If
    ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
    Hist(Undo + 1) = "Karikatura": HistInd(Undo + 1) = 22
    UpdZpet
```

```
    mdiMain.pb.Visible = False: mdiMain.lbPb.Visible = False
End Sub

Public Sub Dualizace(Prah, Kanal, Invertovat, Vyber)
Dim DuaX, DuaY, DuaSou

    If Vyber = False Then
            frmTemp.picTemp.Picture = picOb.Picture
            frmTemp.picTemp.Height     =     picOb.ScaleHeight     +     2:
frmTemp.picTemp.Width = picOb.ScaleWidth + 2
        Else
            frmTemp.picTemp.Picture = picTemp.Picture
            frmTemp.picTemp.Height     =     picTemp.ScaleHeight     +     2:
frmTemp.picTemp.Width = picTemp.ScaleWidth + 2
    End If
    mdiMain.pb.Max          =          frmTemp.picTemp.ScaleHeight          *
frmTemp.picTemp.ScaleWidth
    mdiMain.pb.Value = 1: mdiMain.pb.Visible = True
    mdiMain.lbPb.Caption = "Dualizace: ": mdiMain.lbPb.Visible = True

    For DuaY = 0 To frmTemp.picTemp.ScaleHeight - 1
        For DuaX = 0 To frmTemp.picTemp.ScaleWidth - 1
            Select Case Kanal
                Case "R"
                    DuaSou = frmTemp.picTemp.Point(DuaX, DuaY) Mod 256
                Case "G"
                    DuaSou = (frmTemp.picTemp.Point(DuaX, DuaY) \ 256) Mod
256
                Case "B"
                    DuaSou = frmTemp.picTemp.Point(DuaX, DuaY) \ 65536
                Case "P"
                    DuaSou = frmTemp.picTemp.Point(DuaX, DuaY) Mod 256
                    DuaSou = DuaSou + (frmTemp.picTemp.Point(DuaX, DuaY) \
256) Mod 256
                    DuaSou = DuaSou + frmTemp.picTemp.Point(DuaX, DuaY) \
65536
                    DuaSou = DuaSou \ 3
            End Select
            If Invertovat = 1 Then
                    If DuaSou < Prah Then
                            frmTemp.picTemp.ForeColor                    =
frmNas.picPop.BackColor
                            frmTemp.picTemp.PSet (DuaX, DuaY)
                        Else
                            frmTemp.picTemp.ForeColor                    =
frmNas.picPoz.BackColor
                            frmTemp.picTemp.PSet (DuaX, DuaY)
                    End If
                Else
                    If DuaSou > Prah Then
                            frmTemp.picTemp.ForeColor                    =
frmNas.picPop.BackColor
                            frmTemp.picTemp.PSet (DuaX, DuaY)
                        Else
                            frmTemp.picTemp.ForeColor                    =
frmNas.picPoz.BackColor
                            frmTemp.picTemp.PSet (DuaX, DuaY)
                    End If
            End If
```

```
        Next DuaX
        If DuaY Mod 10 = 1 Then DoEvents: mdiMain.pb.Value = DuaY *
frmTemp.picTemp.ScaleWidth
    Next DuaY

    frmTemp.picTemp.Picture = frmTemp.picTemp.image
    If Vyber = False Then
            picOb.PaintPicture frmTemp.picTemp.Picture, 0, 0
        Else
            picTemp.Picture = frmTemp.picTemp.Picture
            picOb.PaintPicture picTemp.Picture, shVyb.Left + 1, shVyb.Top +
1
    End If
    ReDim Preserve Hist(Undo + 1): ReDim Preserve HistInd(Undo + 1)
    Hist(Undo + 1) = "Dualizace": HistInd(Undo + 1) = 21
    UpdZpet
    mdiMain.pb.Visible = False: mdiMain.lbPb.Visible = False
End Sub
```

G.7. Paleta barev – frmPal

```
Option Explicit

Dim Zob As Integer, Poc As Integer
Dim ResX, ResY

Private Sub Form_Load()
Dim I, Hod As String, iniPal, Bar() As String

    Hod = String(50, 0)
    iniPal = App.Path & "\data\palet.wr"
    Zob = 0
    ZIni "palet", "poc", -1, Hod, 5000, iniPal
    Poc = Int(Hod)
    If Poc = -1 Then Exit Sub Else picPal(0).Visible = True
    For I = 0 To Poc
        ZIni "palet", "bar" & I, "nic", Hod, 5000, iniPal
        If Not Left(Hod, 3) = "nic" Then
                Novy
                Bar = Split(Hod, ";")
                picPal(I).BackColor = RGB(Bar(0), Bar(1), Bar(2))
                ZIni "palet", "pop" & I, "", Hod, 5000, iniPal
                picPal(I).ToolTipText = Hod
                Zob = Zob + 1
        End If
    Next I
    Zob = Zob - 1
    Resize
End Sub

Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    If Button = vbRightButton Then
        mnuPopUp.Enabled = False
        mnuPopSm.Enabled = False
        Me.PopupMenu mnuPop
    End If
```

```
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Cancel = 1
End Sub

Private Sub imgRes_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then ResX = X: ResY = Y
End Sub

Private Sub imgRes_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = vbLeftButton Then
        If Not Me.Width + X - ResX < 2985 Then Me.Width = Me.Width + X -
ResX Else Me.Width = 2985
        Resize
        If Not Me.Height + Y - ResY < 810 Then Me.Height = Me.Height + Y -
ResY Else Me.Height = 810
        picList.Width = Me.ScaleWidth
        picList.Top = Me.ScaleHeight - picList.Height
        picRes.Left = picList.ScaleWidth - picRes.Width
    End If
End Sub

Private Sub mnuPopNo_Click()
    Zob = Zob + 1
    Novy
    Resize
    PalIndex = Zob
    TypBar = 3
    frmBar.Nove
    frmBar.Show 1
End Sub

Private Sub mnuPopPr_Click()
    txZm.Text = ""
    txZm.Top = picPal(PalIndex).Top + picPal(PalIndex).Height
    If Not picPal(PalIndex).Left + txZm.Width > frmPal.ScaleWidth Then
txZm.Left = picPal(PalIndex).Left Else txZm.Left = frmPal.ScaleWidth -
txZm.Width
    txZm.Text = picPal(PalIndex).ToolTipText
    txZm.SelStart = 0: txZm.SelLength = Len(txZm.Text)
    txZm.Visible = True
    txZm.SetFocus
End Sub

Private Sub mnuPopSm_Click()
Dim I

    If MsgBox("Opravdu chcete smazat položku z palety?", vbQuestion +
vbYesNo, "Smazat?") = vbNo Then Exit Sub

    For I = PalIndex To Zob - 1
        picPal(I).BackColor = picPal(I + 1).BackColor
    Next I
    picPal(Zob).Visible = False: Zob = Zob - 1
End Sub
```

```
Private Sub mnuPopUp_Click()
    TypBar = 3
    frmBar.Nove
    frmBar.Show 1
End Sub

Private Sub picPal_MouseMove(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
Dim Bar, R, G, B

    Bar = picPal(Index).BackColor
    R = Bar Mod 256
    G = (Bar \ 256) Mod 256
    B = Bar \ 65536
    lbRGB.Caption = "R:" & R & "; G:" & G & "; B:" & B
End Sub

Private Sub picPal_MouseUp(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
    PalIndex = Index
    If Button = vbRightButton Then
        mnuPopUp.Enabled = True
        mnuPopSm.Enabled = True
        Me.PopupMenu mnuPop
    ElseIf Button = vbLeftButton Then
        frmNas.picPop.BackColor = picPal(Index).BackColor
    End If
End Sub

Private Sub Resize()
Dim PocX, I

    PocX = Me.ScaleWidth \ 12
    For I = 0 To Zob
        picPal(I).Left = (I Mod PocX) * 12
        picPal(I).Top = (I \ PocX) * 12
    Next I
End Sub

Private Sub Novy()
If Zob = 0 Then Exit Sub
    Load picPal(Zob)
    picPal(Zob).Visible = True
End Sub

Public Sub Konci()
Dim I, iniPal, BarRGB, R, G, B

    iniPal = App.Path & "\data\palet.wr"
    Open iniPal For Output As #1
    Close
    DoIni "palet", "poc", Zob, iniPal
    For I = 0 To Zob
        BarRGB = picPal(I).BackColor
        R = BarRGB Mod 256
        G = (BarRGB \ 256) Mod 256
        B = BarRGB \ 65536
        DoIni "palet", "bar" & I, R & ";" & G & ";" & B, iniPal
        DoIni "palet", "pop" & I, picPal(I).ToolTipText, iniPal
```

```
    Next I
End Sub

Private Sub txZm_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then txZm.Visible = False
    picPal(PalIndex).ToolTipText = txZm.Text
End Sub

Private Sub txZm_LostFocus()
    txZm.Visible = False
    picPal(PalIndex).ToolTipText = txZm.Text
End Sub
```

G.8. Posuvník – frmPos

```
Option Explicit

Dim vTyp

Public Sub Nast(Typ As Single)
    Select Case Typ
        Case 0
            vTyp = 0
            slHod.Max = 100
            slHod.Value = Int(mdiMain.txVyh.Text)
        Case 1
            vTyp = 1
            slHod.Max = 100
            slHod.Value = Int(mdiMain.txTla.Text)
        Case 2
            vTyp = 2
            slHod.Max = 150
            If Int(mdiMain.txSir.Text) < 101 Then
                slHod.Value = Int(mdiMain.txSir.Text)
            ElseIf Int(mdiMain.txSir.Text) < 301 Then
                slHod.Value = 100 + (Int(mdiMain.txSir.Text) - 100) / 4
            Else
                slHod.Value = 150
            End If
    End Select
End Sub

Private Sub Form_Deactivate()
    frmPos.Hide
End Sub

Private Sub slHod_Scroll()
    Select Case vTyp
        Case 0
            mdiMain.txVyh.Text = slHod.Value
        Case 1
            mdiMain.txTla.Text = slHod.Value
        Case 2
            If slHod.Value < 101 Then
                mdiMain.txSir.Text = slHod.Value
            Else
                mdiMain.txSir.Text = 100 + (slHod.Value - 100) * 4
```

```
            End If
    End Select


End Sub
```

G.9. Splash Screen – frmSplashScreen

```
Option Explicit

Private Const ULW_OPAQUE = &H4
Private Const ULW_COLORKEY = &H1
Private Const ULW_ALPHA = &H2
Private Const BI_RGB As Long = 0&
Private Const DIB_RGB_COLORS As Long = 0
Private Const AC_SRC_ALPHA As Long = &H1
Private Const AC_SRC_OVER = &H0
Private Const WS_EX_LAYERED = &H80000
Private Const GWL_STYLE As Long = -16
Private Const GWL_EXSTYLE As Long = -20
Private Const HWND_TOPMOST As Long = -1
Private Const SWP_NOMOVE As Long = &H2
Private Const SWP_NOSIZE As Long = &H1

Private Type BLENDFUNCTION
    BlendOp As Byte
    BlendFlags As Byte
    SourceConstantAlpha As Byte
    AlphaFormat As Byte
End Type

Private Type Size
    CX As Long
    CY As Long
End Type

Private Type POINTAPI
    X As Long
    Y As Long
End Type

Private Type RGBQUAD
    rgbBlue As Byte
    rgbGreen As Byte
    rgbRed As Byte
    rgbReserved As Byte
End Type

Private Type BITMAPINFOHEADER
    biSize As Long
    biWidth As Long
    biHeight As Long
    biPlanes As Integer
    biBitCount As Integer
    biCompression As Long
    biSizeImage As Long
    biXPelsPerMeter As Long
```

```
    biYPelsPerMeter As Long
    biClrUsed As Long
    biClrImportant As Long
End Type

Private Type BITMAPINFO
    bmiHeader As BITMAPINFOHEADER
    bmiColors As RGBQUAD
End Type

Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long)
As Long
Private Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA"
(ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
Private Declare Function UpdateLayeredWindow Lib "user32.dll" (ByVal hwnd
As Long, ByVal hdcDst As Long, pptDst As Any, psize As Any, ByVal hdcSrc As
Long, pptSrc As Any, ByVal crKey As Long, ByRef pblend As BLENDFUNCTION,
ByVal dwFlags As Long) As Long
Private Declare Function CreateDIBSection Lib "gdi32.dll" (ByVal hdc As
Long, pBitmapInfo As BITMAPINFO, ByVal un As Long, ByRef lplpVoid As Any,
ByVal handle As Long, ByVal dw As Long) As Long
Private Declare Function CreateCompatibleDC Lib "gdi32.dll" (ByVal hdc As
Long) As Long
Private Declare Function SelectObject Lib "gdi32.dll" (ByVal hdc As Long,
ByVal hObject As Long) As Long
Private Declare Function DeleteDC Lib "gdi32.dll" (ByVal hdc As Long) As
Long
Private Declare Function SetWindowPos Lib "user32.dll" (ByVal hwnd As Long,
ByVal hWndInsertAfter As Long, ByVal X As Long, ByVal Y As Long, ByVal CX
As Long, ByVal CY As Long, ByVal wFlags As Long) As Long
Private Declare Function GetWindowLong Lib "user32.dll" Alias
"GetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long

Dim mDC As Long  ' Memory hDC
Dim mainBitmap As Long ' Memory Bitmap
Dim blendFunc32bpp As BLENDFUNCTION
Dim token As Long ' Needed to close GDI+
Dim oldBitmap As Long
Dim Cas

Private Sub Form_Initialize()
   ' Start up GDI+
   Dim GpInput As GdiplusStartupInput
   GpInput.GdiplusVersion = 1
   If GdiplusStartup(token, GpInput) <> 0 Then
     MsgBox "Error loading GDI+!", vbCritical
     Unload Me
   End If
   MakeTrans (App.Path & "\data\SplashScreen.png")
End Sub

Private Sub Form_Unload(Cancel As Integer)
    ' Cleanup everything
    Call GdiplusShutdown(token)
    SelectObject mDC, oldBitmap
    DeleteObject mainBitmap
    DeleteObject oldBitmap
    DeleteDC mDC
End Sub
```

```
Private Function MakeTrans(pngPath As String) As Boolean
   Dim tempBI As BITMAPINFO
   Dim tempBlend As BLENDFUNCTION      ' Used to specify what kind of blend
we want to perform
   Dim lngHeight As Long, lngWidth As Long
   Dim curWinLong As Long
   Dim img As Long
   Dim graphics As Long
   Dim winSize As Size
   Dim srcPoint As POINTAPI

   With tempBI.bmiHeader
      .biSize = Len(tempBI.bmiHeader)
      .biBitCount = 32    ' Each pixel is 32 bit's wide
      .biHeight = Me.ScaleHeight  ' Height of the form
      .biWidth = Me.ScaleWidth    ' Width of the form
      .biPlanes = 1   ' Always set to 1
      .biSizeImage = .biWidth * .biHeight * (.biBitCount / 8) ' This is the
number of bytes that the bitmap takes up. It is equal to the
Width*Height*ByteCount (bitCount/8)
   End With
   mDC = CreateCompatibleDC(Me.hdc)
   mainBitmap = CreateDIBSection(mDC, tempBI, DIB_RGB_COLORS, ByVal 0, 0,
0)
   oldBitmap = SelectObject(mDC, mainBitmap)     ' Select the new bitmap,
track the old that was selected

   ' GDI Initializations
   Call GdipCreateFromHDC(mDC, graphics)
   Call GdipLoadImageFromFile(StrConv(pngPath, vbUnicode), img)  ' Load Png
   Call GdipGetImageHeight(img, lngHeight)
   Call GdipGetImageWidth(img, lngWidth)
   Call GdipDrawImageRect(graphics, img, 0, 0, lngWidth, lngHeight)

   ' Change windows extended style to be used by updatelayeredwindow
   curWinLong = GetWindowLong(Me.hwnd, GWL_EXSTYLE)
    ' Accidently did This line below which flipped entire form, it's neat
so I left it in
    ' Comment out the line above and uncomment line below.
    'curWinLong = GetWindowLong(Me.hwnd, GWL_STYLE)
   SetWindowLong Me.hwnd, GWL_EXSTYLE, curWinLong Or WS_EX_LAYERED

    ' Make the window a top-most window so we can always see the cool stuff
   SetWindowPos Me.hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE Or SWP_NOSIZE

   ' Needed for updateLayeredWindow call
   srcPoint.X = 0
   srcPoint.Y = 0
   winSize.CX = Me.ScaleWidth
   winSize.CY = Me.ScaleHeight

   With blendFunc32bpp
      .AlphaFormat = AC_SRC_ALPHA ' 32 bit
      .BlendFlags = 0
      .BlendOp = AC_SRC_OVER
      .SourceConstantAlpha = 255
   End With
```

```
   Call GdipDisposeImage(img)
   Call GdipDeleteGraphics(graphics)
   Call  UpdateLayeredWindow(Me.hwnd,  Me.hdc,  ByVal  0&,  winSize,  mDC,
srcPoint, 0, blendFunc32bpp, ULW_ALPHA)
End Function

Private Sub t_Timer()
    Cas = Cas + 1
    If Cas = 200 Then
        mdiMain.Show
        Unload frmSplashScreen
    End If
End Sub
```

G.10. Výběr tloušťky stopy – frmSte

```
Option Explicit

Dim Poc, iniSte, Pocet As String

Private Sub Form_Deactivate()
    frmSte.Hide
End Sub

Private Sub Form_Load()
Dim I, Hod As String

    Poc = 1
    txTl.Text = slTl.Value
    iniSte = App.Path & "\data\brush.wr"
    Pocet = String(50, 0)
    ZIni "seznam", "poc", "Nic", Pocet, 5000, iniSte
    If IsNumeric(Pocet) = False Then Exit Sub

    Hod = String(50, 0)
    ZIni "seznam", 1, "0", Hod, 5000, iniSte
    lbSir(0).Caption = Hod
    If Int(Hod) > 20 Then
        picPol(0).Circle (21, 13), 10
    Else
        picPol(0).Circle (21, 13), Int(Hod / 2)
    End If
    If Int(Hod) > 33 Then lnPr(0).BorderWidth = 33 Else lnPr(0).BorderWidth
= Int(Hod)
    If Int(Hod) = 1 Then picPol(0).PSet (21, 13)
    If Int(Hod) = 2 Then
        picPol(0).Cls
        picPol(0).Line (20, 12)-(21, 13), , BF
    End If

    For I = 2 To Int(Pocet)
        Poc = I - 1
        Novy
        ZIni "seznam", I, "0", Hod, 5000, iniSte
        lbSir(I - 1).Caption = Hod
        If Int(Hod) > 33 Then lnPr(I - 1).BorderWidth = 33 Else lnPr(I -
1).BorderWidth = Int(Hod)
```

```
            If Int(Hod) > 20 Then
                picPol(I - 1).Circle (21, 13), 10
            Else
                picPol(I - 1).Circle (21, 13), Int(Hod / 2)
            End If
            If Int(Hod) = 1 Then picPol(I - 1).PSet (21, 13)
            If Int(Hod) = 2 Then
                picPol(I - 1).Cls
                picPol(I - 1).Line (20, 12)-(21, 13), , BF
            End If
        Next I

        If Int(Pocet) < 4 Then
            scPol.Max = 1
            scPol.Enabled = False
        Else
            scPol.Max = Int(Pocet - 2)
            scPol.Enabled = True
        End If
    End Sub

    Private Sub picPol_Click(Index As Integer)
    Dim I

        For I = 0 To picPol.count - 1
            shVyb(I).Visible = False
        Next I
        shVyb(Index).Visible = True
        txTl.Text = lbSir(Index).Caption
    End Sub

    Private Sub scPol_Change()
        If scPol.Value = scPol.Max Then
            frPol.Top = -((scPol.Max - 1) * 615)
        Else
            frPol.Top = -(scPol.Value * 615)
        End If
    End Sub

    Private Sub scPol_Scroll()
        If scPol.Value = scPol.Max Then
            frPol.Top = -((scPol.Max - 1) * 615)
        Else
            frPol.Top = -(scPol.Value * 615)
        End If
    End Sub

    Private Sub slTl_Scroll()
        txTl.Text = slTl.Value
    End Sub

    Private Sub Novy()
        Load picPol(Poc)
        Set picPol(Poc).Container = frPol
        picPol(Poc).Top = Poc * 615
        frPol.Height = (Poc + 1) * 615
        picPol(Poc).Visible = True
        Load ln(Poc)
        Set ln(Poc).Container = picPol(Poc)
```

```
    With ln(Poc)
        .X1 = 41: .X2 = 41
        .Y1 = 0: .Y2 = 41
    End With
    ln(Poc).Visible = True
    Load lbSir(Poc)
    Set lbSir(Poc).Container = picPol(Poc)
    lbSir(Poc).Top = 24
    lbSir(Poc).Visible = True
    Load shVyb(Poc)
    Set shVyb(Poc).Container = picPol(Poc)
    shVyb(Poc).ZOrder 0
    shVyb(Poc).Top = 1: shVyb(Poc).Left = 1
    Load lnPr(Poc)
    Set lnPr(Poc).Container = picPol(Poc)
    With lnPr(Poc)
        .X1 = 61: .X2 = 161
        .Y1 = 20: .Y2 = 20
        .Visible = True
    End With
End Sub

Private Sub txTl_Change()
    If txTl = "" Then Exit Sub

    mdiMain.picSir.Cls
    If Int(txTl.Text) > 24 Then
        mdiMain.sh.Shape = 3
        mdiMain.sh.Left = 0: mdiMain.sh.Top = 0
        mdiMain.sh.Height = 25: mdiMain.sh.Width = 25
    Else
        If Int(txTl.Text) < 3 Then mdiMain.sh.Shape = 1 Else
mdiMain.sh.Shape = 3
        mdiMain.sh.Left = 13 - Int(txTl.Text) / 2: mdiMain.sh.Top = 13 -
Int(txTl.Text) / 2
        mdiMain.sh.Height = Int(txTl.Text): mdiMain.sh.Width =
Int(txTl.Text)
    End If
    If Int(txTl.Text) = 1 Then mdiMain.picSir.PSet (12, 12)
    mdiMain.lbSir.Caption = txTl.Text
    slTl.Value = Int(txTl.Text)
End Sub

Private Sub txTl_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0
End Sub
```

G.11. Nový soubor – frmVel

```
Option Explicit

Private Sub cbVel_Click()
    Select Case cbVel.ListIndex
        Case 0
            txSi.Text = 640
            txVy.Text = 480
```

```
        Case 1
            txSi.Text = 800
            txVy.Text = 600
        Case 2
            txSi.Text = 1024
            txVy.Text = 768
        Case 3
            txSi.Text = 468
            txVy.Text = 60
        Case 5
            txSi.Text = 720
            txVy.Text = 540
        Case 6
            txSi.Text = 720
            txVy.Text = 534
        Case 7
            txSi.Text = 864
            txVy.Text = 486
        Case 8
            txSi.Text = 864
            txVy.Text = 480
        Case 9
            txSi.Text = 768
            txVy.Text = 576
        Case 10
            txSi.Text = 1024
            txVy.Text = 576
        Case 11
            txSi.Text = 1280
            txVy.Text = 720
        Case 12
            txSi.Text = 1920
            txVy.Text = 1080
        Case 4, 13
            cbVel.ListIndex = 14
    End Select
End Sub

Private Sub cbVel_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then cmdOK_Click: Exit Sub
End Sub

Private Sub cmdOK_Click()
Dim FSize
    frmVel.Hide
    PoslX = Int(txSi.Text): PoslY = Int(txVy.Text)
    Document(fIndex).Caption = txNaz.Text
    Document(fIndex).picOb.Height = Int(txVy.Text) + 2
    Document(fIndex).picOb.Width = Int(txSi.Text) + 2
    Document(fIndex).picOb.Picture = Document(fIndex).picOb.image
    Document(fIndex).picZoom.Height = Int(txVy.Text) + 2
    Document(fIndex).picZoom.Width = Int(txSi.Text) + 2
    Document(fIndex).picZoom.Picture = Document(fIndex).picOb.Picture
    Document(fIndex).picStin.Height = Int(txVy.Text) + 2
    Document(fIndex).picStin.Width = Int(txSi.Text) + 2
    Document(fIndex).PrvZpet
    Document(fIndex).Show
    PoslY = Int(txVy.Text)
    PoslX = Int(txSi.Text)
```

```
    SavePicture Document(fIndex).picOb.Picture, App.Path & "\TempPic.bmp"
    FSize = FileLen(App.Path & "\TempPic.bmp")
    If FSize < 1024 Then
        mdiMain.lbVel.Caption = "Vel.: " & FSize & " B"
    ElseIf FSize >= 1024 And FSize < 1048576 Then
        mdiMain.lbVel.Caption = "Vel.: " & Int(FSize * 100 / 1024) / 100 &
" KB"
    ElseIf FSize >= 1048576 Then
        mdiMain.lbVel.Caption = "Vel.: " & Int(FSize * 100 / 1048576) / 100
& " MB"
    End If
    FState(fIndex).Velikost = FSize
    Kill App.Path & "\TempPic.bmp"
End Sub

Private Sub cmdZru_Click()
    frmVel.Hide
    FState(fIndex).Smaz = True
End Sub

Private Sub Form_Activate()
    txNaz.SetFocus
    If Clipboard.GetFormat(vbCFBitmap) = True Then
        picTemp.Picture = Clipboard.GetData(vbCFBitmap)
        txSi.Text = picTemp.ScaleWidth
        txVy.Text = picTemp.ScaleHeight
    End If
    If Clipboard.GetFormat(vbCFBitmap) = True Then
            frmTemp.picTemp.Picture = Clipboard.GetData(vbCFBitmap)
            txSi.Text = frmTemp.picTemp.ScaleWidth
            txVy.Text = frmTemp.picTemp.ScaleHeight
        Else
            txSi.Text = PoslX
            txVy.Text = PoslY
    End If
End Sub

Private Sub txNaz_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then cmdOK_Click: Exit Sub
End Sub

Private Sub txSi_KeyDown(KeyCode As Integer, Shift As Integer)
    If Shift = 1 Then Exit Sub
    cbVel.ListIndex = 14
End Sub

Private Sub txSi_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then cmdOK_Click: Exit Sub
    If KeyAscii = 8 Or KeyAscii = 9 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub

Private Sub txVy_KeyDown(KeyCode As Integer, Shift As Integer)
    If Shift = 1 Then Exit Sub
    cbVel.ListIndex = 14
End Sub

Private Sub txVy_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then cmdOK_Click: Exit Sub
```

```
    If KeyAscii = 8 Or KeyAscii = 9 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub
```

G.12. Hlavní formulář – mdiMain

```
Option Explicit

Dim iniMDI
Dim Zmac(2) As Boolean

Private Sub cbCar_Click()
    If mdiMain.ActiveForm Is Nothing Or cbObr.ListIndex = 0 Then Exit Sub
    If Textuji(mdiMain.ActiveForm.Tag) = False Then Exit Sub
    mdiMain.ActiveForm.picOb.DrawStyle = cbCar.ListIndex
    mdiMain.ActiveForm.picOb.DrawWidth = Val(txSir.Text)
    txSir_Change
    mdiMain.ActiveForm.Textni
End Sub

Private Sub cbFont_Click()
    If mdiMain.ActiveForm Is Nothing Then Exit Sub
    If Textuji(mdiMain.ActiveForm.Tag) = False Then Exit Sub
    mdiMain.ActiveForm.picOb.FontName = cbFont.Text
    mdiMain.ActiveForm.Textni
End Sub

Private Sub cbObr_Click()
    If mdiMain.ActiveForm Is Nothing Then Exit Sub
    If Textuji(mdiMain.ActiveForm.Tag) = False Then Exit Sub
    Select Case cbObr.ListIndex
        Case 0
            cbCar.Enabled = False
            mdiMain.ActiveForm.picOb.DrawStyle = 5
            JenObrys = False
        Case 1
            If Int(txSir.Text) = 1 Then cbCar.Enabled = True Else
cbCar.Enabled = False
            mdiMain.ActiveForm.picOb.DrawStyle = cbCar.ListIndex
            JenObrys = False
        Case 2
            If Int(txSir.Text) = 1 Then cbCar.Enabled = True Else
cbCar.Enabled = False
            mdiMain.ActiveForm.picOb.DrawStyle = cbCar.ListIndex
            JenObrys = True
    End Select
    mdiMain.ActiveForm.picOb.DrawWidth = Val(txSir.Text)
    txSir_Change
    mdiMain.ActiveForm.Textni
End Sub

Private Sub cbSpec_Click()
    If mdiMain.ActiveForm Is Nothing Then Exit Sub
    If Textuji(mdiMain.ActiveForm.Tag) = False Then Exit Sub
    Select Case cbSpec.ListIndex
        Case 0
```

```
            mdiMain.ActiveForm.picOb.FontUnderline          =          False:
mdiMain.ActiveForm.picOb.FontStrikethru = False
        Case 1
            mdiMain.ActiveForm.picOb.FontUnderline          =           True:
mdiMain.ActiveForm.picOb.FontStrikethru = False
        Case 2
            mdiMain.ActiveForm.picOb.FontUnderline          =          False:
mdiMain.ActiveForm.picOb.FontStrikethru = True
        Case 3
            mdiMain.ActiveForm.picOb.FontUnderline          =           True:
mdiMain.ActiveForm.picOb.FontStrikethru = True
    End Select
    mdiMain.ActiveForm.Textni
End Sub

Private Sub cbTyp_Click()
    If mdiMain.ActiveForm Is Nothing Then Exit Sub
    If Textuji(mdiMain.ActiveForm.Tag) = False Then Exit Sub
    Select Case cbTyp.ListIndex
        Case 0
            mdiMain.ActiveForm.picOb.FontBold           =           False:
mdiMain.ActiveForm.picOb.FontItalic = False
        Case 1
            mdiMain.ActiveForm.picOb.FontBold           =           True:
mdiMain.ActiveForm.picOb.FontItalic = False
        Case 2
            mdiMain.ActiveForm.picOb.FontBold           =           False:
mdiMain.ActiveForm.picOb.FontItalic = True
        Case 3
            mdiMain.ActiveForm.picOb.FontBold           =           True:
mdiMain.ActiveForm.picOb.FontItalic = True
    End Select
    mdiMain.ActiveForm.Textni
End Sub

Private Sub cbVel_Click()
    If mdiMain.ActiveForm Is Nothing Then Exit Sub
    If Textuji(mdiMain.ActiveForm.Tag) = False Then Exit Sub
    mdiMain.ActiveForm.picOb.FontSize = cbVel.Text
    mdiMain.ActiveForm.Textni
End Sub

Private Sub cbVel_Change()
    If cbVel.Text = "" Or cbVel.Text < 1 Then Exit Sub
    If mdiMain.ActiveForm Is Nothing Then Exit Sub
    If Textuji(mdiMain.ActiveForm.Tag) = False Then Exit Sub
    mdiMain.ActiveForm.picOb.FontSize = cbVel.Text
    mdiMain.ActiveForm.Textni
End Sub

Private Sub cbVel_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Or KeyAscii = 9 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0: Beep
End Sub

Private Sub cbVz_Click()
    TypKap = cbVz.ListIndex
End Sub
```

```
Private Sub cmdOd_Click()
Dim I

    For I = 0 To 1
        lbS(I).Caption = "Š" & I + 1 & ": "
        lbV(I).Caption = "V" & I + 1 & ": "
        lbD(I).Caption = "D" & I + 1 & ": "
    Next I
    lbU.Caption = "U: "
    Document(mdiMain.ActiveForm.Tag).SmazMer
End Sub

Private Sub frText_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    picTKon.Picture = imSed.ListImages(5).Picture
    picTPo.Picture = imSed.ListImages(6).Picture
End Sub

Private Sub frTl_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim Tp

    If Elipsuj = True Then Tp = 0 Else Tp = 2
    If Plno = True Then
        picTyp(1).Picture = imZm.ListImages(1 + Tp).Picture
        picTyp(2).Picture = imSed.ListImages(2 + Tp).Picture
    Else
        picTyp(1).Picture = imSed.ListImages(1 + Tp).Picture
        picTyp(2).Picture = imZm.ListImages(2 + Tp).Picture
    End If
End Sub

Private Sub lbSt_Click()
    frmSte.Left = Me.Left + 720
    frmSte.Top = Me.Top + 1380
    frmSte.Show 0, mdiMain
End Sub

Private Sub lbTla_Click()
    frmPos.Left = Me.Left + 2560
    frmPos.Top = Me.Top + 1380
    frmPos.Nast (1)
    frmPos.Show 0, mdiMain
End Sub

Private Sub lbTSir_Click()
    frmPos.Left = Me.Left + 11000
    frmPos.Top = Me.Top + 1380
    frmPos.Nast (2)
    frmPos.Show 0, mdiMain
End Sub

Private Sub lbVel_Change()
    lbPoz.Left = lbVel.Left + lbVel.Width + 20
End Sub

Private Sub lbVyh_Click()
    frmPos.Left = Me.Left + 1500
    frmPos.Top = Me.Top + 1380
```

```
    frmPos.Nast (0)
    frmPos.Show 0, mdiMain
End Sub

Private Sub MDIForm_Click()
    Me.SetFocus
End Sub

Private Sub MDIForm_DblClick()
    Otevrit
End Sub

Private Sub MDIForm_Load()
Dim I
    iniMDI = App.Path & "\data\setup.wr"

    ReDim Document(0)
    ReDim FState(0)
    mdiMain.Caption = "WR PhotoRabbit"
    picSir.PSet (12, 12)
    cbVz.Text = cbVz.List(0)
    PoslX = 200: PoslY = 400
    For I = 0 To 1
        lbS(I).Caption = "Š" & I + 1 & ": "
        lbV(I).Caption = "V" & I + 1 & ": "
        lbD(I).Caption = "D" & I + 1 & ": "
    Next I
    lbU.Caption = "U: "
    frmPal.Show 0, mdiMain

    Nacti
    Pisma
End Sub

Private Sub MDIForm_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Zavrit
End Sub

Private Sub MDIForm_Resize()
    ln.X2 = Me.ScaleWidth
End Sub

Private Sub mnuNapO_Click()
    frmAbout.Show 1, mdiMain
End Sub

Private Sub mnuSouKon_Click()
    Zavrit
End Sub

Private Sub mnuSouNov_Click()
    NovyProc
End Sub

Private Sub mnuSouOt_Click()
    Otevrit
End Sub

Private Sub mnuZoPa_Click()
```

```
    Zobrazit 0
End Sub

Public Sub Nacti()
Dim Hod As String

    mdiMain.Visible = True
    Hod = String(50, 0)
    ZIni "mdi", "x", 120, Hod, 5000, iniMDI
    mdiMain.Left = Hod
    ZIni "nas", "x", mdiMain.Left + 360, Hod, 5000, iniMDI
    frmNas.Left = Hod
    ZIni "nas", "y", mdiMain.Left + 360, Hod, 5000, iniMDI
    frmNas.Top = Hod
    ZIni "barva", "pop", RGB(0, 0, 0), Hod, 5000, iniMDI
    frmNas.picPop.BackColor = Hod
    ZIni "barva", "poz", RGB(255, 255, 255), Hod, 5000, iniMDI
    frmNas.picPoz.BackColor = Hod
    ZIni "ost", "ste", 5, Hod, 5000, iniMDI
    frmSte.txTl.Text = Hod
    ZIni "ost", "tlak", 100, Hod, 5000, iniMDI
    txTla.Text = Hod
    ZIni "ost", "vyhl", 1, Hod, 5000, iniMDI
    txVyh.Text = Hod
    ZIni "ost", "kont", 1, Hod, 5000, iniMDI
    chKo.Value = Hod
    ZIni "ost", "x", 130, Hod, 5000, iniMDI
    PoslX = Hod
    ZIni "ost", "y", 260, Hod, 5000, iniMDI
    PoslY = Hod
    frmNas.Show 0, mdiMain
    mdiMain.SetFocus
End Sub

Private Sub picLista_Resize()
    picHist.Left = picLista.ScaleWidth - picHist.Width - 5
End Sub

Private Sub picObr_Click()
    TypBar = 2
    frmBar.Nove
    frmBar.Show 1
    If mdiMain.ActiveForm Is Nothing Or Textuji(mdiMain.ActiveForm.Tag) =
False Then Exit Sub
    mdiMain.ActiveForm.picOb.ForeColor = picObr.BackColor
    mdiMain.ActiveForm.Textni
End Sub

Private Sub picSt_Click()
    lbSt_Click
End Sub

Private Sub picStav_Resize()
    pb.Left = picStav.ScaleWidth - pb.Width - 2
    lbPb.Left = pb.Left - lbPb.Width
End Sub

Private Sub picTKon_Click()
    With mdiMain.ActiveForm
```

```
        .picOb.Cls
        .shTe.Visible = False
        .shSel.Visible = False
        .tmrKur.Enabled = False
        .lnKur.Visible = False
    End With
    Textuji(mdiMain.ActiveForm.Tag) = False
End Sub

Private Sub picTKon_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    picTKon.Picture = imNajZm.ListImages(5).Picture
End Sub

Private Sub picTKon_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
     If Button = 0 Then picTKon.Picture = imNaj.ListImages(5).Picture
     picTPo.Picture = imSed.ListImages(6).Picture
End Sub

Private Sub picTKon_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    picTKon.Picture = imSed.ListImages(5).Picture
End Sub

Private Sub picTla_Click()
    lbTla_Click
End Sub

Private Sub picTPo_Click()
    With mdiMain.ActiveForm
        .UpdZpet
        .picOb.Cls
        .shTe.Visible = False
        .shSel.Visible = False
        .tmrKur.Enabled = False
        .lnKur.Visible = False
    End With
    Textuji(mdiMain.ActiveForm.Tag) = False
End Sub

Private Sub picTPo_MouseDown(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    picTPo.Picture = imNajZm.ListImages(6).Picture
End Sub

Private Sub picTPo_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    If Button = 0 Then picTPo.Picture = imNaj.ListImages(6).Picture
    picTKon.Picture = imSed.ListImages(5).Picture
End Sub

Private Sub picTPo_MouseUp(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    picTPo.Picture = imSed.ListImages(6).Picture
End Sub

Private Sub picTSir_Click()
    lbTSir_Click
```

```
End Sub

Private Sub picTyp_Click(Index As Integer)
Dim Tp

    If Elipsuj = True Then Tp = 0 Else Tp = 2
    picTyp(1).Picture = imSed.ListImages(1 + Tp).Picture
    picTyp(2).Picture = imSed.ListImages(1 + Tp).Picture
    picTyp(Index).Picture = imNajZm.ListImages(Index + Tp).Picture
    If Index = 1 Then Plno = True Else Plno = False
End Sub

Private Sub picTyp_MouseMove(Index As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
Dim I, Tp

    If Elipsuj = True Then Tp = 0 Else Tp = 2
    If Plno = True Then Zmac(1) = True: Zmac(2) = False Else Zmac(1) =
False: Zmac(2) = True

    For I = 1 To 2
        If Zmac(I) = True Then
            If Index = I Then
                picTyp(I).Picture = imNajZm.ListImages(I + Tp).Picture
            Else
                picTyp(I).Picture = imZm.ListImages(I + Tp).Picture
            End If
        Else
            If Index = I Then
                picTyp(I).Picture = imNaj.ListImages(I + Tp).Picture
            Else
                picTyp(I).Picture = imSed.ListImages(I + Tp).Picture
            End If
        End If
    Next I
End Sub

Private Sub picVyh_Click()
    lbVyh_Click
End Sub


Private Sub txSir_Change()
    If txSir.Text = "" Then Exit Sub
    If Int(txSir.Text) = 1 Then cbCar.Enabled = True Else cbCar.Enabled =
False
    If mdiMain.ActiveForm Is Nothing Then Exit Sub
    If Textuji(mdiMain.ActiveForm.Tag) = False Then Exit Sub
    If mdiMain.ActiveForm Is Nothing Or Textuji(mdiMain.ActiveForm.Tag) =
False Or txSir.Text = "" Then Exit Sub
    mdiMain.ActiveForm.picOb.DrawWidth = Int(txSir.Text)
    mdiMain.ActiveForm.Textni
End Sub

Private Sub txSir_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0
End Sub
```

```
Private Sub txTl_Change()
    If txTl.Text = "" Then Exit Sub
    If txTl.Text < 1 Then MsgBox "Hodnota musí být větší než 0.",
vbCritical, "Chyba": txTl.Text = 1
End Sub

Private Sub txTl_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0
End Sub

Private Sub txTla_Change()
    If txTla.Text = "" Then Exit Sub
    If txTla.Text < 1 Then MsgBox "Hodnota musí být větší než 0.",
vbCritical, "Chyba": txTla.Text = 1
End Sub

Private Sub txTla_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0
End Sub

Private Sub txVyh_Change()
    If txVyh.Text = "" Then Exit Sub
    If txVyh.Text > 100 Then MsgBox "Hodnota musí být mezi 1 a 100.",
vbCritical, "Chyba": txVyh.Text = 100
    If txVyh.Text < 1 Then MsgBox "Hodnota musí být mezi 1 a 100.",
vbCritical, "Chyba": txVyh.Text = 1
End Sub

Private Sub txVyh_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If KeyAscii > 57 Or KeyAscii < 48 Then KeyAscii = 0
End Sub

Public Sub Zobrazit(Co As Single)
    Select Case Co
        Case 0
            frmNas.Show 0, mdiMain
        Case 1
            frmPal.Show 0, mdiMain
    End Select
End Sub

Public Sub Zavrit()
Dim I

On Error GoTo Eror

    For I = 0 To UBound(Document)
        If FState(I).Zmena = True And FState(I).Smaz = False Then
            Select Case MsgBox("Chcete uložit změny v dokumentu " &
Document(I).Caption & "?", vbQuestion + vbYesNoCancel, "Uložit změny?")
                Case vbYes
                    If FState(I).Cesta = "Novy" Then
                        mdiMain.CMD.Flags = cdlOFNOverwritePrompt +
cdlOFNNoReadOnlyReturn
                        mdiMain.CMD.CancelError = True
```

```
                        mdiMain.CMD.Filter   =   "JPG   (*.jpg)|*.jpg|Bitmapa
(*.bmp)|*.bmp|Gif (*.gif)|*.gif|Průhledný Gif (*.gif)|*.gif"
                        mdiMain.CMD.FilterIndex = FState(I).Typ
                        mdiMain.CMD.filename = Document(I).Caption
                        mdiMain.CMD.ShowSave
                        Ulozit                          mdiMain.CMD.filename,
mdiMain.CMD.FilterIndex, I
                    Else
                        Ulozit FState(I).Cesta, FState(I).Typ, I
                    End If
                Case vbCancel
                    Exit Sub
            End Select
        End If
    Next I

    frmPal.Konci
    DoIni "mdi", "x", mdiMain.Left, iniMDI
    DoIni "mdi", "y", mdiMain.Top, iniMDI
    DoIni "mdi", "wid", mdiMain.Width, iniMDI
    DoIni "mdi", "hei", mdiMain.Height, iniMDI
    DoIni "mdi", "sta", mdiMain.WindowState, iniMDI
    DoIni "nas", "x", frmNas.Left, iniMDI
    DoIni "nas", "y", frmNas.Top, iniMDI
    DoIni "barva", "pop", frmNas.picPop.BackColor, iniMDI
    DoIni "barva", "poz", frmNas.picPoz.BackColor, iniMDI
    DoIni "ost", "ste", frmSte.txTl.Text, iniMDI
    DoIni "ost", "tlak", txTla.Text, iniMDI
    DoIni "ost", "vyhl", txVyh.Text, iniMDI
    DoIni "ost", "kont", chKo.Value, iniMDI
    DoIni "ost", "x", PoslX, iniMDI
    DoIni "ost", "y", PoslY, iniMDI
    End
Exit Sub
Eror:
    If Err = 32755 Then Exit Sub
End Sub

Private Sub txZo_Click()
    txZo.SelStart = 0: txZo.SelLength = Len(txZo.Text) - 1
End Sub

Private Sub txZo_KeyPress(KeyAscii As Integer)
Dim Zoom

    If KeyAscii = 13 Then
        KeyAscii = 0
        If  txZo.Text  =  ""  Or  txZo.Text  =  "%"  Then  txZo.Text  =
FState(mdiMain.ActiveForm.Tag).Zoom
        If Not Right(txZo.Text, 1) = "%" Then txZo.Text = txZo.Text & "%"
        mdiMain.SetFocus
        Zoom = Int(Left(txZo.Text, Len(txZo.Text) - 1))
        If   Zoom   <   5   Or   Zoom   >   3200   Then   txZo.Text   =
FState(mdiMain.ActiveForm.Tag).Zoom & "%": Exit Sub
        FState(mdiMain.ActiveForm.Tag).Zoom = Zoom
        mdiMain.ActiveForm.Zoom
        Exit Sub
    End If
    If KeyAscii = 8 Then Exit Sub
```

```
    If KeyAscii < 48 Or KeyAscii > 57 Then KeyAscii = 0
End Sub

Private Sub txZo_LostFocus()
    If   txZo.Text   =   ""   Or   txZo.Text   =   "%"   Then   txZo.Text   =
FState(mdiMain.ActiveForm.Tag).Zoom
    If Not Right(txZo.Text, 1) = "%" Then txZo.Text = txZo.Text & "%"
End Sub

Private Sub Pisma()
Dim I
    Me.Show
    cbTyp.ListIndex = 0
    cbSpec.ListIndex = 0
    cbCar.ListIndex = 0
    cbObr.ListIndex = 0
    txSir.Text = 3
    pb.Max = Screen.FontCount
    pb.Min = 1
    For I = 0 To Screen.FontCount - 1
        cbFont.AddItem Screen.Fonts(I)
        cbFont.ListIndex = 0
        If I Mod 10 = 0 Then
            pb.Value = I + 1
            DoEvents
        End If
    Next I
    pb.Visible = False: lbPb.Visible = False
End Sub
```

G.13. Modul Barvy – Barvy

```
Option Explicit

Public Function FindMax(Unu As Integer, Du As Integer, Tri As Integer)
Dim Max As Integer

    Max = Unu
    If Du > Max Then Max = Du
    If Tri > Max Then Max = Tri

    FindMax = Max
End Function

Public Function FindMin(Unu As Integer, Du As Integer, Tri As Integer)
Dim Min As Integer

    Min = Unu
    If Du < Min Then Min = Du
    If Tri < Min Then Min = Tri

    FindMin = Min
End Function


Public Function RGBtoHSB(R As Integer, G As Integer, B As Integer)
Dim Max, Min, H, s, Br
```

```
    Max = FindMax(R, G, B)
    Min = FindMin(R, G, B)

    If Max > 255 Or Min < 0 Then RGBtoHSB = "0;0;0": Exit Function

    If Max = Min Then H = 0: GoTo PoH:
    If Max = R Then
        If G > B Or G = B Then
            H = 60 * (G - B) / (Max - Min)
        Else
            H = (60 * (G - B) / (Max - Min)) + 360
        End If
    ElseIf Max = G Then
        H = (60 * (B - R) / (Max - Min)) + 120
    ElseIf Max = B Then
        H = (60 * (R - G) / (Max - Min)) + 240
    End If
PoH:

    If Max = 0 Then
        s = 0
    Else
        s = (Max - Min) / Max * 100
    End If

    Br = Max * 100 / 255

    RGBtoHSB = H & ";" & s & ";" & Br
End Function

Public Function HSBtoRGB(H As Integer, s As Integer, B As Integer, I As
String)
Dim F, P, Q, t, Hi, Sa, Br

    If H > 359 Or H < 0 Or s > 100 Or s < 0 Or B > 100 Or B < 0 Then
HSBtoRGB = "0;0;0": Exit Function

    s = s
    B = B

    Hi = Int(H / 60) Mod 6
    F = (H / 60) - Hi
    P = B * (100 - s) * 0.0255
    Q = B * (100 - F * s) * 0.0255
    t = B * (100 - (1 - F) * s) * 0.0255
    Br = B * 2.55

    Select Case Hi
        Case 0
            HSBtoRGB = Br & ";" & t & ";" & P
        Case 1
            HSBtoRGB = Q & ";" & Br & ";" & P
        Case 2
            HSBtoRGB = P & ";" & Br & ";" & t
        Case 3
            HSBtoRGB = P & ";" & Q & ";" & Br
        Case 4
            HSBtoRGB = t & ";" & P & ";" & Br
```

```
            Case 5
                HSBtoRGB = Br & ";" & P & ";" & Q
        End Select

        If Br > 0.5 And I <> "" Then
            s = B
        End If
    End Function

    Public Function PreHex(HexCislo As String)
        Dim HR, HG, HB
        Dim R1 As String, R2 As String, G1 As String, G2 As String, B1 As
    String, B2 As String
        Dim R, G, B

        R1 = Mid(HexCislo, 1, 1)
        R2 = Mid(HexCislo, 2, 1)
        G1 = Mid(HexCislo, 3, 1)
        G2 = Mid(HexCislo, 4, 1)
        B1 = Mid(HexCislo, 5, 1)
        B2 = Mid(HexCislo, 6, 1)

        If IsNumeric(R1) <> True Then
                R = 16 * Int(DeHexni(R1))
            Else
                R = 16 * Int(R1)
        End If
        If IsNumeric(R2) <> True Then
                R = R + Int(DeHexni(R2))
            Else
                R = R + Int(R2)
        End If

        If IsNumeric(G1) <> True Then
                G = 16 * Int(DeHexni(G1))
            Else
                G = 16 * Int(G1)
        End If
        If IsNumeric(G2) <> True Then
                G = G + Int(DeHexni(G2))
            Else
                G = G + Int(G2)
        End If

        If IsNumeric(B1) <> True Then
                B = 16 * Int(DeHexni(B1))
            Else
                B = 16 * Int(B1)
        End If
        If IsNumeric(B2) <> True Then
                B = B + Int(DeHexni(B2))
            Else
                B = B + Int(B2)
        End If

        PreHex = R & "," & G & "," & B
    End Function

    Private Function DeHexni(Cislo As String)
```

```
    Select Case Cislo
        Case "A", "a"
            DeHexni = 10
        Case "B", "b"
            DeHexni = 11
        Case "C", "c"
            DeHexni = 12
        Case "D", "d"
            DeHexni = 13
        Case "E", "e"
            DeHexni = 14
        Case "F", "f"
            DeHexni = 15
    End Select
End Function

Public Function PreDec(Red As Integer, Green As Integer, Blue As Integer)
    Dim R, G, B

    R = Hex(Red)
    If Len(R) <> 2 Then R = String(2 - Len(R), "0") & R
    G = Hex(Green)
    If Len(G) <> 2 Then G = String(2 - Len(G), "0") & G
    B = Hex(Blue)
    If Len(B) <> 2 Then B = String(2 - Len(B), "0") & B
    PreDec = R & G & B
End Function
```

G.14. Modul Doplnení – Doplněni

```
Option Explicit

Public Declare Function GetKeyboardState Lib "user32" (pbKeyState As
KeyboardBytes) As Long

Public Type KeyboardBytes
    kbByte(255) As Byte
End Type

Public kbArray As KeyboardBytes
Public TypBar, PalIndex
Public JenObrys As Boolean

Public Sub Skryj()
Dim I

    mdiMain.frSto.Visible = False
    mdiMain.frVzo.Visible = False
    mdiMain.frTl.Visible = False
    mdiMain.frMer.Visible = False
    mdiMain.frText.Visible = False
    For I = 0 To UBound(Document)
        Document(I).HideMer
    Next I
    For I = 0 To UBound(Document)
        Document(I).DeTex
        Textuji(I) = False
```

```
    Next I
End Sub
```

G.15. Modul File Open – FileOpen

```
Option Explicit

Type FormState
    Smaz As Boolean
    Zmena As Boolean
    Cesta As String
    Typ As Integer
    Velikost As Long
    Zoom As Integer
End Type

Public FState()  As FormState   ' Pole uzivatelsky definovane promenne
Public Document() As New frmOb  ' Pole child formularu
Public fIndex As Integer
Public PoslX, PoslY
Public PoslCesta

Sub NovyProc()
    'Najdi nejnizsi volny index a vytvor novy document
    fIndex = VolnyIndex()
    Document(fIndex).Hide
    Document(fIndex).Tag = fIndex
    FState(fIndex).Typ = 2
    FState(fIndex).Zoom = 100
    Presun(fIndex) = False
    Textuji(fIndex) = False
    If Clipboard.GetFormat(vbCFBitmap) = True Then
            frmTemp.picTemp.Picture = Clipboard.GetData(vbCFBitmap)
            frmVel.txSi.Text = frmTemp.picTemp.ScaleWidth
            frmVel.txVy.Text = frmTemp.picTemp.ScaleHeight
        Else
            frmVel.txSi.Text = PoslX
            frmVel.txVy.Text = PoslY
    End If
    frmVel.txNaz = "Bez názvu " & fIndex
    frmVel.cbVel.ListIndex = 14
    frmVel.Show 1, mdiMain
End Sub

Function VolnyIndex() As Integer
Dim I As Integer
Dim Polata As Integer

    Polata = UBound(Document) 'Najde nejvyssi index

    ' Cyklus projde pole documentu. Kdyz jeden byl smazan, vrati jeho index
    For I = 1 To Polata
        If FState(I).Smaz Then
            VolnyIndex = I
            FState(I).Smaz = False
            FState(I).Cesta = "Novy"
            Exit Function
```

```
        End If
    Next I

    ' Kdyz nebyl zadny smazan, rozsiri pole o jednu
    ReDim Preserve Document(Polata + 1)
    ReDim Preserve FState(Polata + 1)
    ReDim Preserve Presun(Polata + 1)
    ReDim Preserve Textuji(Polata + 1)
    FState(Polata + 1).Cesta = "Novy"
    VolnyIndex = UBound(Document)
End Function

Sub Otevrit()
Dim Polata As Integer, I As Integer

    On Error GoTo Eror
    mdiMain.CMD.filename = ""
    mdiMain.CMD.Filter       =       "Všechny       soubory       obrázků
|*.bmp;*.jpg;*.jpeg;*.jpe;*.jfif;*.gif;*.tif;*.tiff;*.ico|Rastrové  soubory
(*.bmp)|*.bmp|JPEG
(*.jpg;*.jpeg;*.jpe;*.jfif)|*.jpg;*.jpeg;*.jpe;*.jfif|GIF
(*.gif)|*.gif|TIFF   (*.tif;*.tiff)|*.tif;*.tiff|ICO   (*.ico)|*.ico|Všechny
soubory|*.*"
    mdiMain.CMD.ShowOpen

    If LCase(Right(mdiMain.CMD.filename, 4)) = ".gif" Then setGifOptions

    Otevri False
Exit Sub
Eror:
    If Err = 32755 Then Exit Sub
End Sub

Public Sub setGifOptions()
Dim setOpt, ret1, ret2
Dim hFile         As Long
Dim sImgHeader    As String
Dim sFileHeader   As String
Dim sBuff         As String
Dim sPicsBuff     As String
Dim nImgCount     As Long
Dim I             As Long
Dim j             As Long
Dim Finished As Boolean, GifType As String
Dim sGifMagic As String
On Error GoTo ErrHandler
    'Načíst GIF to proměné buffer
    hFile = FreeFile
    Open mdiMain.CMD.filename For Binary Access Read As hFile
    'Vyplní  buffer  netisknutelnými  znaky  se  stejnou  délkou  jako  má
otevíraný GIF
    'Přečte celou hlavičku GIFu a všechny snímky do sBuff
        sBuff = String(LOF(hFile), Chr(0))
        Get #hFile, , sBuff
    Close #hFile
    'Hlavička a snímky GIFu (kromě posledního snímku)
    'Končí na null(0), Vykřičník(33) a u s kroužkem(249)
    sGifMagic = Chr(0) & Chr(33) & Chr(249)
    ret1 = InStr(1, sBuff, sGifMagic)
```

```
    If ret1 = 0 Then Exit Sub                           'EXIT SUB
    GifType = Mid(sBuff, 1, ret1 + 2)
    ret2 = InStr(ret1 + 1, sBuff, sGifMagic)
    If ret2 = 0 Then Exit Sub                           'EXIT SUB
    setOpt = MsgBox("Tento GIF je animovaný." & Chr(13) & "Chcete otevřít
kařdý snímek zvlášť?", vbYesNo)
    If setOpt = vbNo Then Exit Sub                      'EXIT SUB
        '====Frame 1 will be loaded back in the main routine. Will now load
Frame 2 to last frame.
        'set pointer ahead 3 bytes from the
        'end of the gif magic number
        I = ret2 + 3
        'create a temporary file in the current directory
        hFile = FreeFile
        Open App.Path & "\tempvbDraw.gif" For Binary As hFile
            'split out each frame of the gif, and
            'write each frame to the temporary file.
            'Then load a frmImage
            nImgCount = 1
            Do
                'increment counter
                nImgCount = nImgCount + 1
                'locate next frame end
                j = InStr(I, sBuff, sGifMagic) + 3 '3 is for 3 char in
sGifMagic
                'Right here is where trouble occurs because there is no
sGifMagic
                'after last image. It is either EOF or Comment starting
with
                'null (0) exclamation(33) and funny p(254).
                '==========
                If j = 3 Then
                    'check for comment
                    j = InStr(I, sBuff, Chr$(0) & Chr$(33) & Chr$(254)) + 3
                    Finished = True
                    'If j > 3 Then
                    'MsgBox "Found Funny p Comment j=" & j & " i= " & i & "
nImgCount: " & nImgCount & " Len(sBuff) " & Len(sBuff)
                    'End If
                    'no comment
                    If j = 3 Then
                        j = Len(sBuff) '- 1
                        'MsgBox "NO Comment j=" & j & " nImgCount: " &
nImgCount
                        Finished = True
                        'MsgBox "Finished = " & Finished
                    End If
                End If
                '=========
                'another check
                If j > Len(sGifMagic) Then
                    'pad an output string, fill with the
                    'frame info, and write to disk. A header
                    'needs to be added as well, to assure
                    'LoadPicture recognizes it as a gif.
                    sPicsBuff = Mid(sBuff, I, (j - 3) - (I))
                    'MsgBox Len(sPicsBuff)
                    Put #hFile, 1, GifType & sPicsBuff 'sPicsBuff 'GifType
& sPicsBuff
```

```
                    'assign the data.
                    If nImgCount > 1 Then
                        'if this is the second or later frame
                        'load the picture into a new frmImage.
                        '===============
                        Otevri True
                        '================
                    End If
                    'update pointer
                    I = j
                End If
                'when the j = Instr() command above returns 0,
                '3 is added, so if j = 3 there was no more
                'data in the header. We're done.
                If Finished Then
                    Finished = False
                    j = 3
                End If
            Loop Until j = 3
            'close and nuke the temp file
        Close #hFile
   Kill App.Path & "\tempvbDraw.gif"
Exit Sub
ErrHandler:
    MsgBox "Error No. " & Err.Number & " when reading file. Image Count " &
nImgCount, vbCritical
   'Resume Next
    On Error GoTo 0
End Sub

Private Sub Otevri(Anim As Boolean)
Dim FSize, Temp
Dim Polata As Integer, I

    If Anim = False Then
        FSize = FileLen(mdiMain.CMD.filename)
    Else
        FSize = FileLen(App.Path & "\TempPic.gif")
    End If

    If FSize < 1024 Then
        mdiMain.lbVel.Caption = "Vel.: " & FSize & " B"
    ElseIf FSize >= 1024 And FSize < 1048576 Then
        mdiMain.lbVel.Caption = "Vel.: " & Int(FSize * 100 / 1024) / 100 &
" KB"
    ElseIf FSize >= 1048576 Then
        mdiMain.lbVel.Caption = "Vel.: " & Int(FSize * 100 / 1048576) / 100
& " MB"
    End If

    If Anim = False Then
        Polata = UBound(Document)
        For I = 1 To Polata
            If FState(I).Cesta = mdiMain.CMD.filename And FState(I).Smaz =
False Then
                Document(I).SetFocus
                Exit Sub
            End If
        Next I
```

```
        frmTemp.picTemp.AutoSize = True
        frmTemp.picTemp.Picture = LoadPicture(mdiMain.CMD.filename)
        Temp = InStrRev(mdiMain.CMD.filename, "\")
        PoslCesta = Left(mdiMain.CMD.filename, Temp)
        Temp = InStrRev(mdiMain.CMD.FileTitle, ".")

        fIndex = VolnyIndex()
        Document(fIndex).Hide
        Document(fIndex).Tag = fIndex
        Document(fIndex).Caption = Left(mdiMain.CMD.FileTitle, Temp - 1)
        FState(fIndex).Cesta = mdiMain.CMD.filename
        FState(fIndex).Zoom = 100
        Presun(fIndex) = False
        Textuji(fIndex) = False
        Temp  =  Right(mdiMain.CMD.FileTitle,  Len(mdiMain.CMD.FileTitle)  -
Temp)
        Select Case LCase(Temp)
            Case "jpg", "jpeg", "jpe", "jfif"
                FState(fIndex).Typ = 1
            Case "bmp", "ico"
                FState(fIndex).Typ = 2
            Case "gif", "tif", "tiff"
                FState(fIndex).Typ = 3
        End Select

        Document(fIndex).picOb.Height = frmTemp.picTemp.Height
        Document(fIndex).picOb.Width = frmTemp.picTemp.Width
        Document(fIndex).picOb.Picture = LoadPicture(mdiMain.CMD.filename)
        Document(fIndex).picZoom.Height = frmTemp.picTemp.Height
        Document(fIndex).picZoom.Width = frmTemp.picTemp.Width
        Document(fIndex).picZoom.Picture = Document(fIndex).picOb.Picture
        Document(fIndex).picStin.Height = frmTemp.picTemp.Height
        Document(fIndex).picStin.Width = frmTemp.picTemp.Width
        Document(fIndex).Show
        Document(fIndex).PrvZpet
        FState(fIndex).Velikost = FSize
    Else
        frmTemp.picTemp.AutoSize = True
        frmTemp.picTemp.Picture = LoadPicture(App.Path & "\tempvbDraw.gif")

        fIndex = VolnyIndex()
        Document(fIndex).Hide
        Document(fIndex).Tag = fIndex
        Document(fIndex).Caption = "Anim " & fIndex
        FState(fIndex).Cesta = "Novy"
        FState(fIndex).Zoom = 100
        Presun(fIndex) = False
        FState(fIndex).Typ = 3

        Document(fIndex).picOb.Height = frmTemp.picTemp.Height
        Document(fIndex).picOb.Width = frmTemp.picTemp.Width
        Document(fIndex).picOb.Picture     =     LoadPicture(App.Path     &
"\tempvbDraw.gif")
        Document(fIndex).picStin.Height = frmTemp.picTemp.Height
        Document(fIndex).picStin.Width = frmTemp.picTemp.Width
        Document(fIndex).Show
        Document(fIndex).PrvZpet
        FState(fIndex).Velikost = FSize
```

```
    End If
End Sub

Public Sub Ulozit(Cesta, Typ, Index)
Dim cGIF As GIF
If Cesta = "" Then MsgBox "Musíte zadat cestu!!!", vbCritical, "Chyba":
Exit Sub

    Set cGIF = New GIF
    Select Case Typ
        Case 1 'JPEG
            frmJPEG.ShowMe Cesta, Index
        Case 2
            SavePicture Document(Index).picOb.Picture, Cesta
            FState(Index).Cesta = Cesta
            FState(Index).Typ = Typ
            FState(Index).Zmena = False
        Case 3 'GIF
            cGIF.SaveGIF        Document(Index).picOb.Picture,        Cesta,
Document(Index).picOb.hdc
        Case 4 'Transparent GIF
            cGIF.SaveGIF        Document(Index).picOb.Picture,        Cesta,
Document(Index).picOb.hdc, True, frmNas.picPoz.BackColor
    End Select
    Set cGIF = Nothing
    FState(Index).Zmena = False
End Sub
```

G.16. Modul INI – Ini

```
Public iniCesta

Declare Function ZIni Lib "kernel32" Alias "GetPrivateProfileStringA"
(ByVal Sekce As String, ByVal Klic As String, ByVal Default As String,
ByVal Hodnota As String, ByVal Velikost As Long, ByVal Cesta As String) As
Long

Declare Function DoIni Lib "kernel32" Alias "WritePrivateProfileStringA"
(ByVal Sekce As String, ByVal Klic As String, ByVal Hodnota As String,
ByVal Cesta As String) As Long
```

G.17. Modul Intel JPEG Library – IntelJPEGLibrary

```
Option Explicit
'
================================================================================
=======
' Requires:    cDIBSectionmod.cls
'              ijl15.dll (Intel)
'
' An interface to Intel's IJL (Intel JPG Library) for use in VB.
' http://developer.intel.com/software/products/perflib/ijl/index.htm
'
'
================================================================================
=======
```

```
Private Enum IJLERR
  IJL_OK = 0
End Enum

Private Enum IJLIOTYPE
    ''// Write an entire JFIF bit stream.
  IJL_JFILE_WRITEWHOLEIMAGE = 8&
End Enum
Private Type JPEG_CORE_PROPERTIES_VB
  UseJPEGPROPERTIES As Long                      '// default = 0

  '// DIB specific I/O data specifiers.
  DIBBytes As Long ';                   '// default = NULL 4
  DIBWidth As Long ';                   '// default = 0 8
  DIBHeight As Long ';                  '// default = 0 12
  DIBPadBytes As Long ';                '// default = 0 16
  DIBChannels As Long ';                '// default = 3 20
  DIBColor As Long ';                   '// default = IJL_BGR 24
  DIBSubsampling As Long  ';             '// default = IJL_NONE 28

  '// JPEG specific I/O data specifiers.
  JPGFile As Long 'LPTSTR                JPGFile;                  32    '//
default = NULL
  JPGBytes As Long ';                   '// default = NULL 36
  JPGSizeBytes As Long ';               '// default = 0 40
  JPGWidth As Long ';                   '// default = 0 44
  JPGHeight As Long ';                  '// default = 0 48
  JPGChannels As Long ';                '// default = 3
  JPGColor As Long           ';                  '// default = IJL_YCBCR
  JPGSubsampling As Long  ';             '// default = IJL_411
  JPGThumbWidth As Long ' ;              '// default = 0
  JPGThumbHeight As Long ';             '// default = 0

  '// JPEG conversion properties.
  cconversion_reqd As Long ';           '// default = TRUE
  upsampling_reqd As Long ';            '// default = TRUE
  jquality As Long ';                          '// default = 75.   90 is my
preferred quality setting.

  '// Low-level properties - 20,000 bytes.  If the whole structure
  ' is written out then VB fails with an obscure error message
  ' "Too Many Local Variables" !
  ' These all default if they are not otherwise specified so there
  ' is no trouble.
  jprops(0 To 19999) As Byte

End Type

Private Declare Function ijlInit Lib "ijl15.dll" (jcprops As Any) As Long
Private Declare Function ijlFree Lib "ijl15.dll" (jcprops As Any) As Long
Private Declare Function ijlWrite Lib "ijl15.dll" (jcprops As Any, ByVal
ioType As Long) As Long
Private Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory"
(lpvDest As Any, lpvSource As Any, ByVal cbCopy As Long)

Public Function SaveJPG(ByRef cDIB As cDIBSection, ByVal sFile As String,
Optional ByVal lQuality As Long = 90) As Boolean
Dim tJ As JPEG_CORE_PROPERTIES_VB
```

```
Dim bFile() As Byte
Dim lptr As Long
Dim lR As Long

   lR = ijlInit(tJ)
   If lR = IJL_OK Then
      ' Set up the DIB information:
      ' Store DIBWidth:
      tJ.DIBWidth = cDIB.Width
      ' Store DIBHeight:
      tJ.DIBHeight = -cDIB.Height
      ' Store DIBBytes (pointer to uncompressed JPG data):
      tJ.DIBBytes = cDIB.DIBSectionBitsPtr
      ' Very important: tell IJL how many bytes extra there
      ' are on each DIB scan line to pad to 32 bit boundaries:
      tJ.DIBPadBytes = cDIB.BytesPerScanLine - cDIB.Width * 3

      ' Set up the JPEG information:

      ' Store JPGFile:
      bFile = StrConv(sFile, vbFromUnicode)
      ReDim Preserve bFile(0 To UBound(bFile) + 1) As Byte
      bFile(UBound(bFile)) = 0
      lptr = VarPtr(bFile(0))
      CopyMemory tJ.JPGFile, lptr, 4
      ' Store JPGWidth:
      tJ.JPGWidth = cDIB.Width
      ' .. & JPGHeight member values:
      tJ.JPGHeight = cDIB.Height
      ' Set the quality/compression to save:
      tJ.jquality = lQuality
      ' Write the image:
      lR = ijlWrite(tJ, IJL_JFILE_WRITEWHOLEIMAGE)
      If lR = IJL_OK Then
         SaveJPG = True
      Else
         ' Throw error
         MsgBox "Chyba při ukládání JPG", vbExclamation
      End If
      ' Ensure we have freed memory:
      ijlFree tJ
   Else
      ' Throw error:
      MsgBox "Chyba při inicializaci IJL: " & lR, vbExclamation
   End If
End Function
```

G.18. Modul Malování – Malovani

```
Option Explicit

Public Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal X As
Long, ByVal Y As Long) As Long
Public Declare Function SetPixelV Lib "gdi32" (ByVal hdc As Long, ByVal X
As Long, ByVal Y As Long, ByVal crColor As Long) As Long
```

```
Public Declare Function ExtFloodFill Lib "gdi32.dll" (ByVal hdc As Long,
ByVal nXStart As Long, ByVal nYStart As Long, ByVal crColor As Long, ByVal
fuFillType As Long) As Long
Public Const FLOODFILLBORDER = 0
Public Const FLOODFILLSURFACE = 1

Public Sub NakSte(hDcCil As Long, X As Single, Y As Single, Radius As Long)
Dim CX, CY
Dim TempColor As Long

    ReDim Hot(-Radius To Radius, -Radius To Radius)

    For CY = -Radius To Radius
        For CX = -Radius To Radius
            If (CX ^ 2 + CY ^ 2) ^ (1 / 2) <= Radius Then
                TempColor = GetPixel(hDcCil, CX + X, CY + Y)
                SetPixelV hDcCil, CX + X, CY + Y, SteBar(TempColor, Radius,
(CX ^ 2 + CY ^ 2) ^ (1 / 2), mdiMain.txVyh.Text, mdiMain.txTla.Text)
            End If
        Next CX
    Next CY
End Sub

Private Function SteBar(Barva As Long, Rad As Long, Poloha As Long,
Vyhlazeni As Long, Tlak As Long)
Dim Red1, Green1, Blue1
Dim Red2, Green2, Blue2
Dim VysR, VysG, VysB
Dim RDif, GDif, BDif
Dim VyhRad

    If Rad = 0 Then Rad = 1
    VyhRad = Rad / 100 * (Vyhlazeni + 1)

    Blue1 = frmNas.picPop.BackColor \ 65536
    Green1 = (frmNas.picPop.BackColor - Blue1 * 65536) \ 256
    Red1 = frmNas.picPop.BackColor - Blue1 * 65536 - Green1 * 256

    Blue2 = Barva \ 65536
    Green2 = (Barva - Blue2 * 65536) \ 256
    Red2 = Barva - Blue2 * 65536 - Green2 * 256

    RDif = Red1 - Red2
    GDif = Green1 - Green2
    BDif = Blue1 - Blue2

    If Poloha < VyhRad Then
        VysR = Red2 + (RDif * Tlak / 100)
        VysG = Green2 + (GDif * Tlak / 100)
        VysB = Blue2 + (BDif * Tlak / 100)
        If VysR < 0 Then VysR = 0
        If VysR > 255 Then VysR = 255
        If VysG < 0 Then VysG = 0
        If VysG > 255 Then VysG = 255
        If VysB < 0 Then VysB = 0
        If VysB > 255 Then VysB = 255
        SteBar = RGB(Int(VysR), Int(VysG), Int(VysB))
    Else
```

```
        VysR = Red2 + RDif * Tlak / 100 * (1 - (1 / (Rad - VyhRad) *
(Poloha - VyhRad)))
        VysG = Green2 + GDif * Tlak / 100 * (1 - (1 / (Rad - VyhRad) *
(Poloha - VyhRad)))
        VysB = Blue2 + BDif * Tlak / 100 * (1 - (1 / (Rad - VyhRad) *
(Poloha - VyhRad)))
        If VysR < 0 Then VysR = 0
        If VysR > 255 Then VysR = 255
        If VysG < 0 Then VysG = 0
        If VysG > 255 Then VysG = 255
        If VysB < 0 Then VysB = 0
        If VysB > 255 Then VysB = 255
        SteBar = RGB(Int(VysR), Int(VysG), Int(VysB))
    End If
End Function

Public Sub UserBrush(hDcCil As Long, X As Single, Y As Single, Obrazek As
Integer, Radius As Long)

End Sub
```

G.19. Modul Proměnné – promene

```
Option Explicit

Public klCtrl As Boolean
Public Vyber As Boolean, Vybiram As Boolean
Type Pravitko
    X1 As Single
    X2 As Single
    Y1 As Single
    Y2 As Single
    Visible As Boolean
End Type

Public Presouvej As Boolean, Presouvam As Boolean, Presun() As Boolean
Public Kapatko As Boolean, TypKap As Integer
Public Maluj As Boolean, Maluji As Boolean
Public Stetcuj As Boolean, Stetcuji As Boolean
Public Caruj As Boolean, Caruji As Boolean
Public Plno As Boolean
Public Elipsuj As Boolean, Elipsuji As Boolean
Public Obdelnikuj As Boolean, Obdelnikuji As Boolean
Public Gumuj As Boolean, Gumuji As Boolean
Public Vyplnuj As Boolean, Vyplnuji As Boolean
Public Zoomuj As Boolean, Zoomuji As Boolean
Public Mer As Boolean, Merim As Boolean
Public Rucickuj As Boolean, Rucickuji As Boolean
Public Textuj As Boolean, Textuji() As Boolean

Public Sub ZrusProm()
    Vyber = False: Vybiram = False
    Presouvej = False: Presouvam = False
    Kapatko = False
    Maluj = False: Maluji = False
    Stetcuj = False: Stetcuji = False
    Caruj = False: Caruji = False
```

```
    Elipsuj = False: Elipsuji = False
    Obdelnikuj = False: Obdelnikuji = False
    Gumuj = False: Gumuji = False
    Vyplnuj = False: Vyplnuji = False
    Zoomuj = False: Zoomuji = False
    Mer = False: Merim = False
    Rucickuj = False: Rucickuji = False
    Textuj = False
End Sub
```

G.20. Modul Spash Screen – SplashScreen

```
Option Explicit

Public Declare Function GdipCreateFromHDC Lib "gdiplus" (ByVal hdc As Long,
graphics As Long) As GpStatus
Public Declare Function GdipDeleteGraphics Lib "gdiplus" (ByVal graphics As
Long) As GpStatus
Public Declare Function GdipDrawImageRect Lib "gdiplus" (ByVal graphics As
Long, ByVal image As Long, ByVal X As Single, ByVal Y As Single, ByVal
Width As Single, ByVal Height As Single) As GpStatus
Public Declare Function GdipLoadImageFromFile Lib "gdiplus" (ByVal filename
As String, image As Long) As GpStatus
Public Declare Function GdipGetImageWidth Lib "gdiplus" (ByVal image As
Long, Width As Long) As GpStatus
Public Declare Function GdipGetImageHeight Lib "gdiplus" (ByVal image As
Long, Height As Long) As GpStatus
Public Declare Function GdipDisposeImage Lib "gdiplus" (ByVal image As
Long) As GpStatus

Public Type GdiplusStartupInput
   GdiplusVersion As Long                  ' Must be 1 for GDI+ v1.0, the
current version as of this writing.
   DebugEventCallback As Long           ' Ignored on free builds
   SuppressBackgroundThread As Long      ' FALSE unless you're prepared to
call
                                        ' the hook/unhook functions properly
   SuppressExternalCodecs As Long       ' FALSE unless you want GDI+ only to
use
                                        ' its internal image codecs.
End Type


Public Declare Function GdiplusStartup Lib "gdiplus" (token As Long,
inputbuf As GdiplusStartupInput, Optional ByVal outputbuf As Long = 0) As
GpStatus
Public Declare Sub GdiplusShutdown Lib "gdiplus" (ByVal token As Long)

Public Enum GpStatus   ' aka Status
   Ok = 0
   GenericError = 1
   InvalidParameter = 2
   OutOfMemory = 3
   ObjectBusy = 4
   InsufficientBuffer = 5
   NotImplemented = 6
   Win32Error = 7
```

```
   WrongState = 8
   Aborted = 9
   FileNotFound = 10
   ValueOverflow = 11
   AccessDenied = 12
   UnknownImageFormat = 13
   FontFamilyNotFound = 14
   FontStyleNotFound = 15
   NotTrueTypeFont = 16
   UnsupportedGdiplusVersion = 17
   GdiplusNotInitialized = 18
   PropertyNotFound = 19
   PropertyNotSupported = 20
End Enum
```

G.21. Vlase modul DIB Section – cDIBSection

```
Option Explicit
'
================================================================================
=======
' Requires:    mIJLmod.cls
'              ijl15.dll (Intel)
'
================================================================================
=======

Private Type RGBQUAD
    rgbBlue As Byte
    rgbGreen As Byte
    rgbRed As Byte
    rgbReserved As Byte
End Type
Private Type BITMAPINFOHEADER '40 bytes
    biSize As Long
    biWidth As Long
    biHeight As Long
    biPlanes As Integer
    biBitCount As Integer
    biCompression As Long
    biSizeImage As Long
    biXPelsPerMeter As Long
    biYPelsPerMeter As Long
    biClrUsed As Long
    biClrImportant As Long
End Type
Private Type BITMAPINFO
    bmiHeader As BITMAPINFOHEADER
    bmiColors As RGBQUAD
End Type
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long)
As Long
Private Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long

Private Declare Function GetDesktopWindow Lib "user32" () As Long
Private Declare Function CreateDIBSection Lib "gdi32" _
    (ByVal hdc As Long, _
```

```
    pBitmapInfo As BITMAPINFO, _
    ByVal un As Long, _
    lplpVoid As Long, _
    ByVal handle As Long, _
    ByVal dw As Long) As Long
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X
As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long,
ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop
As Long) As Long
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal
hObject As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long)
As Long
Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Const BI_RGB = 0&
Private Const BI_RLE4 = 2&
Private Const BI_RLE8 = 1&
Private Const DIB_RGB_COLORS = 0 '  color table in RGBs
Private Type BITMAP
    bmType As Long
    bmWidth As Long
    bmHeight As Long
    bmWidthBytes As Long
    bmPlanes As Integer
    bmBitsPixel As Integer
    bmBits As Long
End Type
Private Declare Function GetObjectAPI Lib "gdi32" Alias "GetObjectA" (ByVal
hObject As Long, ByVal nCount As Long, lpObject As Any) As Long

' Handle to the current DIBSection:
Private m_hDIb As Long
' Handle to the old bitmap in the DC, for clear up:
Private m_hBmpOld As Long
' Handle to the Device context holding the DIBSection:
Private m_hDC As Long
' Address of memory pointing to the DIBSection's bits:
Private m_lPtr As Long
' Type containing the Bitmap information:
Private m_tBI As BITMAPINFO
Public Property Get BytesPerScanLine() As Long
    ' Scans must align on dword boundaries:
    BytesPerScanLine = (m_tBI.bmiHeader.biWidth * 3 + 3) And &HFFFFFFFC
End Property
Public Property Get Width() As Long
    Width = m_tBI.bmiHeader.biWidth
End Property
Public Property Get Height() As Long
    Height = m_tBI.bmiHeader.biHeight
End Property
Public Sub LoadPictureBlt( _
        ByVal lhDC As Long, _
        Optional ByVal lSrcLeft As Long = 0, _
        Optional ByVal lSrcTop As Long = 0, _
        Optional ByVal lSrcWidth As Long = -1, _
        Optional ByVal lSrcHeight As Long = -1, _
        Optional ByVal eRop As RasterOpConstants = vbSrcCopy _
    )
    If lSrcWidth < 0 Then lSrcWidth = m_tBI.bmiHeader.biWidth
```

```
    If lSrcHeight < 0 Then lSrcHeight = m_tBI.bmiHeader.biHeight
    BitBlt m_hDC, 0, 0, lSrcWidth, lSrcHeight, lhDC, lSrcLeft, lSrcTop,
eRop
End Sub
Public Property Get DIBSectionBitsPtr() As Long
    DIBSectionBitsPtr = m_lPtr
End Property
Public Sub ClearUp()
    If (m_hDC <> 0) Then
        If (m_hDIb <> 0) Then
            SelectObject m_hDC, m_hBmpOld
            DeleteObject m_hDIb
        End If
        DeleteObject m_hDC
    End If
    m_hDC = 0: m_hDIb = 0: m_hBmpOld = 0: m_lPtr = 0
End Sub
Public Function CreateFromPicture( _
        ByRef picThis As StdPicture _
    )
Dim lhDC As Long
Dim lhDCDesktop As Long
Dim lhBmpOld As Long
Dim tBMP As BITMAP

    GetObjectAPI picThis.handle, Len(tBMP), tBMP
    If (Create(tBMP.bmWidth, tBMP.bmHeight)) Then
        lhDCDesktop = GetDC(GetDesktopWindow())
        If (lhDCDesktop <> 0) Then
            lhDC = CreateCompatibleDC(lhDCDesktop)
            DeleteDC lhDCDesktop
            If (lhDC <> 0) Then
                lhBmpOld = SelectObject(lhDC, picThis.handle)
                LoadPictureBlt lhDC
                SelectObject lhDC, lhBmpOld
                DeleteObject lhDC
            End If
        End If
    End If
End Function
Public Function CreateDIB( _
        ByVal lhDC As Long, _
        ByVal lWidth As Long, _
        ByVal lHeight As Long, _
        ByRef hDib As Long _
    ) As Boolean
    With m_tBI.bmiHeader
        .biSize = Len(m_tBI.bmiHeader)
        .biWidth = lWidth
        .biHeight = lHeight
        .biPlanes = 1
        .biBitCount = 24
        .biCompression = BI_RGB
        .biSizeImage = BytesPerScanLine * .biHeight
    End With
    hDib = CreateDIBSection( _
            lhDC, _
            m_tBI, _
            DIB_RGB_COLORS, _
```

```
            m_lPtr, _
            0, 0)
    CreateDIB = (hDib <> 0)
End Function
Public Function Create( _
        ByVal lWidth As Long, _
        ByVal lHeight As Long _
    ) As Boolean
    ClearUp
    m_hDC = CreateCompatibleDC(0)
    If (m_hDC <> 0) Then
        If (CreateDIB(m_hDC, lWidth, lHeight, m_hDIb)) Then
            m_hBmpOld = SelectObject(m_hDC, m_hDIb)
            Create = True
        Else
            DeleteObject m_hDC
            m_hDC = 0
        End If
    End If
End Function
```

G.22. Vlase modul GIF – GIF

```
' Class for Saving VB StdPicture object (BMP) in GIF format
'
' Written by Arkadiy Olovyannikov (ark@fesma.ru)
'
' This software is FREEWARE. You may use it as you see fit for
' your own projects but you may not re-sell the original or the
' source code.
'
' No warranty express or implied, is given as to the use of this
' program. Use at your own risk.

' This sample was written for education purposes. 'GIF' and
' 'Graphics Interchange Format' are trademarks of Compuserve,
' Inc., an H&R  Block Company.

Option Explicit

'============BITMAP STAFF=======================
Private Type RGBTRIPLE
     rgbRed As Byte
     rgbGreen As Byte
     rgbBlue As Byte
End Type

Private Type RGBQUAD
     rgbBlue As Byte
     rgbGreen As Byte
     rgbRed As Byte
     rgbReserved As Byte
End Type

Private Type BITMAPINFOHEADER '40 bytes
     biSize As Long
     biWidth As Long
```

```
     biHeight As Long
     biPlanes As Integer
     biBitCount As Integer
     biCompression As Long
     biSizeImage As Long
     biXPelsPerMeter As Long
     biYPelsPerMeter As Long
     biClrUsed As Long
     biClrImportant As Long
End Type

Private Type BITMAPINFO256
     bmiHeader As BITMAPINFOHEADER
     bmiColors(0 To 255) As RGBQUAD
End Type

Private Type BITMAP '14 bytes
   bmType As Long
   bmWidth As Long
   bmHeight As Long
   bmWidthBytes As Long
   bmPlanes As Integer
   bmBitsPixel As Integer
   bmBits As Long
End Type

Private Const BI_RGB = 0&

Private Declare Function CreateDCAsNull Lib "gdi32" Alias "CreateDCA"
(ByVal lpDriverName As String, lpDeviceName As Any, lpOutput As Any,
lpInitData As Any) As Long
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long)
As Long
Private Declare Function GetObjectAPI Lib "gdi32" Alias "GetObjectA" (ByVal
hObject As Long, ByVal nCount As Long, lpObject As Any) As Long
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal
hObject As Long) As Long
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X
As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long,
ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop
As Long) As Long
Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long)
As Long
Private Declare Function GetDIBits Lib "gdi32" (ByVal aHDC As Long, ByVal
hBitmap As Long, ByVal nStartScan As Long, ByVal nNumScans As Long, lpBits
As Any, lpBI As BITMAPINFO256, ByVal wUsage As Long) As Long
Private Declare Function CreateDIBSection256 Lib "gdi32" Alias
"CreateDIBSection" (ByVal hdc As Long, pBitmapInfo As BITMAPINFO256, ByVal
un As Long, lplpVoid As Long, ByVal handle As Long, ByVal dw As Long) As
Long
Private Const DIB_RGB_COLORS = 0

Private Declare Function GetFileAttributes Lib "kernel32" Alias
"GetFileAttributesA" (ByVal lpFileName As String) As Long
'============================GIF STAFF================

Private Type GifScreenDescriptor
     logical_screen_width As Integer
```

```
     logical_screen_height As Integer
     Flags As Byte
     background_color_index As Byte
     pixel_aspect_ratio As Byte
End Type

Private Type GifImageDescriptor
     Left As Integer
     Top As Integer
     Width As Integer
     Height As Integer
     Format As Byte 'ImageFormat
End Type
'========Added by Wolfgang Goetz for transparent GIFs=====
Private Type CONTROLBLOCK '(April 8., 2002 --> Wolfgang Goetz)
    Blocksize As Byte
    Flags As Byte
    Delay As Integer
    TransParent_Color As Byte
    Terminator As Byte
End Type
Private Const GIF89a = "GIF89a"
Private Const CtrlIntro As Byte = &H21
Private Const CtrlLabel As Byte = &HF9
'=========================================================
Const GIF87a = "GIF87a"

Const GifTerminator As Byte = &H3B
Const ImageSeparator As Byte = &H2C
Const CHAR_BIT = 8
Const CodeSize As Byte = 9
Const ClearCode = 256
Const EndCode  As Integer = 257
Const FirstCode = 258
Const LastCode As Integer = 511
Const MAX_CODE = LastCode - FirstCode

Private colTable As New Collection
Private fn As Integer
Private gifPalette(0 To 255) As RGBTRIPLE
Private bit_position As Integer
Private code_count As Integer
Private data_buffer(255) As Byte
Private aPower2(31) As Long
Private picWidth As Long, picHeight As Long
Private IsBusy As Boolean
Public Event Progress(ByVal Percents As Integer)

Public Function SaveGIF(ByVal pic As StdPicture, ByVal sFileName As String,
_
       Optional hdc As Long = 0, Optional UseTrans As Boolean = False, _
       Optional ByVal TransColor As Long = 0) As Boolean
   If IsBusy Then Exit Function
   Dim scr As GifScreenDescriptor, im As GifImageDescriptor
   Dim bi As BITMAPINFO256, bm As BITMAP
   Dim hDCScn As Long, OldObj As Long, Src_hDc As Long
   Dim hDib256 As Long, hDC256 As Long, OldObj256 As Long
   Dim buf() As Byte, data As Byte, TransIndex As Byte
   Dim I As Long, j As Long, clr As Long
```

```
   Dim bFound As Boolean
   Dim intCode As Integer, nCount  As Integer
   Dim sPrefix As String, sByte As String
   Dim tempPic As StdPicture
   IsBusy = True
'get image size and allocate buffer memory
   Call GetObjectAPI(pic, Len(bm), bm)
   picWidth = bm.bmWidth
   picHeight = bm.bmHeight
   ReDim buf(CLng(((picWidth + 3) \ 4) * 4), picHeight) As Byte
'Prepare DC for paintings
   hDCScn = CreateDCAsNull("DISPLAY", ByVal 0&, ByVal 0&, ByVal 0&)
   hDC256 = CreateCompatibleDC(hDCScn)
   If hdc = 0 Then
      Src_hDc = CreateCompatibleDC(hDCScn)
      OldObj = SelectObject(Src_hDc, pic)
   Else
      Src_hDc = hdc
   End If
   DeleteDC hDCScn

'Since GIF works only with 256 colors, reduce color depth to 256
'This sample use simpliest HalfTone palette to reduce color depth
'If you want advanced color manipulation with web-safe palettes or
'optimal palette with the specified number of colors using octree
'quantisation, visit http://vbaccelerator.com/codelib/gfx/octree.htm

   If bm.bmBitsPixel <> 8 Then hDib256 = CreateDib256(hDC256, bi)
   If hDib256 <> 0 Then
      OldObj256 = SelectObject(hDC256, hDib256)
      Call  BitBlt(hDC256,  0,  0,  picWidth,  picHeight,  Src_hDc,  0,  0,
vbSrcCopy)
      For I = 0 To picHeight - 1
          Call GetDIBits(hDC256, hDib256, I, 1, buf(0, picHeight - I), bi,
0)
      Next
   Else
      With bi.bmiHeader
          .biSize = Len(bi.bmiHeader)
          .biWidth = picWidth
          .biHeight = picHeight
          .biPlanes = 1
          .biBitCount = 8
          .biCompression = BI_RGB
      End With
      For I = 0 To picHeight - 1
          Call GetDIBits(Src_hDc, pic, I, 1, buf(0, picHeight - I), bi, 0)
      Next
   End If
'Fill gif file info
   For I = 0 To 255
       gifPalette(I).rgbBlue = bi.bmiColors(I).rgbBlue
       gifPalette(I).rgbGreen = bi.bmiColors(I).rgbGreen
       gifPalette(I).rgbRed = bi.bmiColors(I).rgbRed
       If Not bFound Then
          clr     =     RGB(gifPalette(I).rgbRed,     gifPalette(I).rgbGreen,
gifPalette(I).rgbBlue)
          If clr = TransColor Then
             TransIndex = I: bFound = True
```

```
            End If
         End If
      Next
'    If TransColor = 0 Then TransIndex = 0
      scr.background_color_index = 0
      scr.Flags = &HF7 '256-color gif with global color map
      scr.pixel_aspect_ratio = 0

      im.Format = &H7 'GlobalNonInterlaced
      im.Height = picHeight
      im.Width = picWidth

      If FileExists(sFileName) Then Kill sFileName

      fn = FreeFile
      Open sFileName For Binary As fn
'Write GIF header and header info
        If UseTrans = True Then '(April 8., 2002 --> Wolfgang Goetz)
           Put #fn, , GIF89a
        Else
           Put #fn, , GIF87a
        End If
        Put #fn, , scr
        Put #fn, , gifPalette
 '(April 8., 2002 --> Wolfgang Goetz)
        If UseTrans = True Then
           Put #fn, , CtrlIntro
           Put #fn, , CtrlLabel
           Dim cb As CONTROLBLOCK
           cb.Blocksize = 4 'Always 4
           cb.Flags = 9 'Packed = 00001001 (If Bit 0 = 1: Use Transparency)
           cb.Delay = 0
           cb.TransParent_Color = TransIndex
           cb.Terminator = 0 'Always 0
           Put #fn, , cb
        End If
        Put #fn, , ImageSeparator
        Put #fn, , im
        data = CodeSize - 1
        Put #fn, , data
        data_buffer(0) = 0
        bit_position = CHAR_BIT
'Process pixels data using LZW/GIF compression
        For I = 1 To picHeight
            Reinitialize
            sPrefix = ""
            intCode = buf(0, I)
            On Error Resume Next
            For j = 1 To picWidth - 1
                sByte = MyFormat(buf(j, I))
                sPrefix = sPrefix & sByte
                intCode = colTable(sPrefix)
                If Err <> 0 Then 'Prefix wasn't in collection - save it and
output code
                   nCount = colTable.count
                   If nCount = MAX_CODE Then Reinitialize
                    colTable.Add nCount + FirstCode, sPrefix
                    OutputBits intCode, CodeSize
                    sPrefix = sByte
```

```
                intCode = buf(j, I)
                Err.Clear
            End If
        Next
        OutputBits intCode, CodeSize
        If I Mod 10 = 0 Then
           RaiseEvent Progress(I * 100 / picHeight)
           DoEvents
        End If
     Next
     OutputCode (EndCode)
     For I = 0 To data_buffer(0)
         Put #fn, , data_buffer(I)
     Next
     data = 0
     Put #fn, , data
     Put #fn, , GifTerminator
   Close fn
   Erase buf
   If hdc = 0 Then
      SelectObject Src_hDc, OldObj
      DeleteDC Src_hDc
   End If
   SelectObject hDC256, OldObj256
   DeleteObject hDib256
   DeleteDC hDC256
   SaveGIF = True
   IsBusy = False
End Function

Private Sub OutputBits(Value As Integer, count As Integer)
   Dim I As Integer, bit As Integer
   Do While I < count
      If bit_position = CHAR_BIT Then
         If data_buffer(0) = 255 Then
            Put #fn, , data_buffer
            data_buffer(0) = 1
         Else
            data_buffer(0) = data_buffer(0) + 1
         End If
         data_buffer(data_buffer(0)) = 0
         bit_position = 0
       End If
       bit = Sgn(Power2(I) And Value)
       If bit > 0 Then data_buffer(data_buffer(0)) = Power2(bit_position)
Or data_buffer(data_buffer(0))
       I = I + 1: bit_position = bit_position + 1
   Loop
End Sub

Private Sub OutputCode(code As Integer)
   code_count = code_count + 1
   If code_count > LastCode Then
      code_count = FirstCode
      Call OutputBits(ClearCode, CodeSize)
      ClearTable
    End If
    Call OutputBits(code, CodeSize)
End Sub
```

```
Private Sub ClearTable()
   Set colTable = Nothing
   Set colTable = New Collection
End Sub

Private Sub Reinitialize()
   ClearTable
   Call OutputBits(ClearCode, CodeSize)
End Sub

Private Function FileExists(ByVal strPathName As String) As Boolean
   Dim af As Long
   af = GetFileAttributes(strPathName)
   FileExists = (af <> -1)
End Function

Private Function Power2(ByVal I As Integer) As Long
    If aPower2(0) = 0 Then
       aPower2(0) = &H1&
       aPower2(1) = &H2&
       aPower2(2) = &H4&
       aPower2(3) = &H8&
       aPower2(4) = &H10&
       aPower2(5) = &H20&
       aPower2(6) = &H40&
       aPower2(7) = &H80&
       aPower2(8) = &H100&
       aPower2(9) = &H200&
       aPower2(10) = &H400&
       aPower2(11) = &H800&
       aPower2(12) = &H1000&
       aPower2(13) = &H2000&
       aPower2(14) = &H4000&
       aPower2(15) = &H8000&
       aPower2(16) = &H10000
       aPower2(17) = &H20000
       aPower2(18) = &H40000
       aPower2(19) = &H80000
       aPower2(20) = &H100000
       aPower2(21) = &H200000
       aPower2(22) = &H400000
       aPower2(23) = &H800000
       aPower2(24) = &H1000000
       aPower2(25) = &H2000000
       aPower2(26) = &H4000000
       aPower2(27) = &H8000000
       aPower2(28) = &H10000000
       aPower2(29) = &H20000000
       aPower2(30) = &H40000000
       aPower2(31) = &H80000000
    End If
    Power2 = aPower2(I)
End Function

Private Function MyFormat(ByVal s As String) As String
   MyFormat = Right$("00" & s, 3)
End Function
```

```
Private Function CreateDib256(ByVal h_Dc As Long, bi As BITMAPINFO256) As
Long
   Dim lScanSize As Long
   Dim lptr As Long, lIndex As Long
   Dim R As Long, G As Long, B As Long
   Dim rA As Long, gA As Long, bA As Long
   With bi.bmiHeader
       .biSize = Len(bi.bmiHeader)
       .biWidth = picWidth
       .biHeight = picHeight
       .biPlanes = 1
       .biBitCount = 8
       .biCompression = BI_RGB
       lScanSize = (picWidth + picWidth Mod 4)
       .biSizeImage = lScanSize * picHeight
   End With
   ' Halftone 256 colour palette
   For B = 0 To &H100 Step &H40
      If B = &H100 Then
         bA = B - 1
      Else
         bA = B
      End If
      For G = 0 To &H100 Step &H40
         If G = &H100 Then
            gA = G - 1
         Else
            gA = G
         End If
         For R = 0 To &H100 Step &H40
            If R = &H100 Then
               rA = R - 1
            Else
               rA = R
            End If
            With bi.bmiColors(lIndex)
               .rgbRed = rA: .rgbGreen = gA: .rgbBlue = bA
            End With
            lIndex = lIndex + 1
         Next R
      Next G
   Next B
   CreateDib256 = CreateDIBSection256(h_Dc, bi, DIB_RGB_COLORS, lptr, 0, 0)
End Function
```

Vydala společnost Wabbit s.r.o. v roce 2008

Copyright © 2008 Lukáš Kotrba

Obálku a grafickou úpravu navrhl Lukáš Kotrba
Sazba Wabbit s.r.o., Praha
Tisk Wabbit s.r.o., Praha

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Přenos dat pomocí sítě GSM

Jan FIKAR

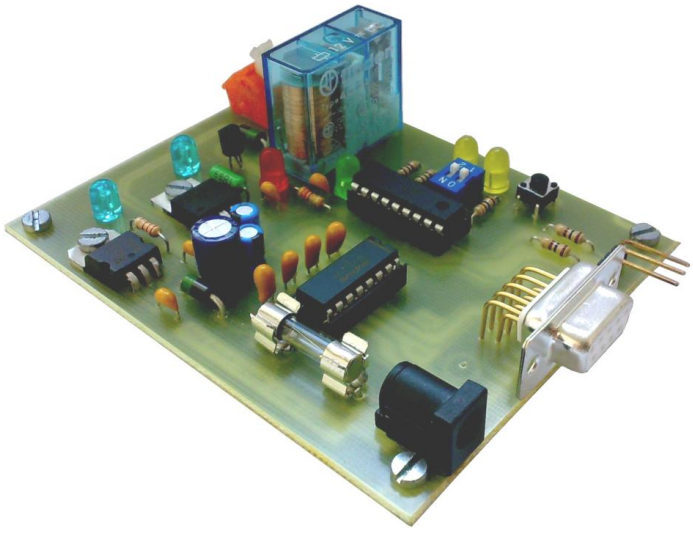
VYŠŠÍ ODBORNÁ ŠKOLA, STŘEDNÍ ŠKOLA, CENTRUM ODBORNÉ PŘÍPRAVY,
Budějovická 421, 391 02 Sezimovo Ústí

Práce se zabývá využitím bezdrátového přenosu dat pomocí sítě GSM. Ta je dnes roz-šířená téměř po celém světě a využívají ji i mobilní telefony, např. k přenosu hovorů.

Autor si jako hlavní cíl stanovil postavit zařízení umožňující přenos dat přes stávající robotickou síť RoboCOP, která je ovládaná povelově a je používaná v laboratoři mikro-procesorové techniky na VOŠ, SŠ, COP Sezimovo Ústí.

Prvním výstupem práce je GSM převaděč vyrobený podle návrhu firmy Flajzar, s.r.o. připojený k PC pomocí sériové linky RS-232. Ten umožní sledovat stav mobilního telefonu a také jej ovládat z pracovního prostředí Windows (např. odesílaní SMS, zápis telefon-ních čísel na SIM kartu v telefonu, apod.).
Druhým výstupem projektu je přijímací GSM brána s jednočipovým mikroprocesorem vlastní konstrukce a k tomuto obvodu připojený druhý mobilní telefon a jeho napájení. To-to zařízení lze řídit GSM převaděčem bezdrátově tak, že se z PC odesílají pro jednočipový mikroprocesor potřebná data.

Kompletní sestava obou desek se bude moci využít při výuce jako názorná pomůcka a bude začleněna do stávající školní sítě RoboCOP.



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Komstrukce prototypu pohybové jednotky robota s krokovými motory

Jiří Květoň

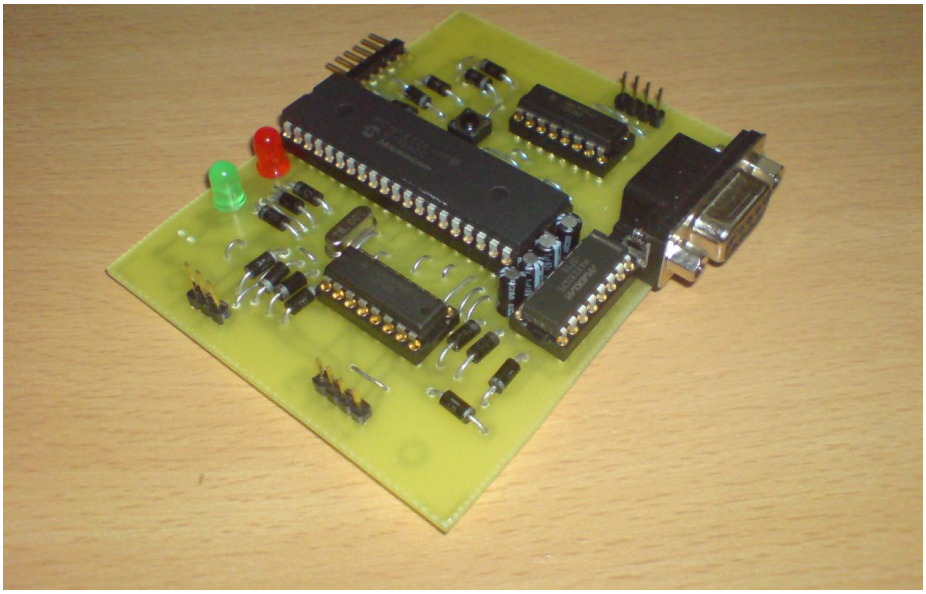
VYŠŠÍ ODBORNÁ ŠKOLA, STŘEDNÍ ŠKOLA, CENTRUM ODBORNÉ PŘÍPRAVY
Budějovická 421, 391 02 Sezimovo Ústí

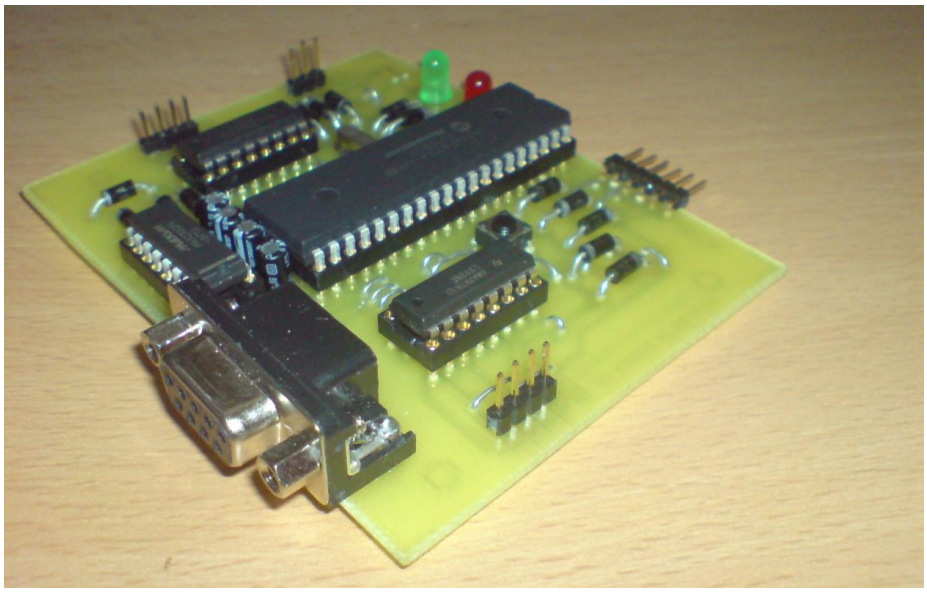
Hlavním úkolem práce je popsat možnosti využití a řízení krokových motorů, navrhnout řídící systém pro pohonnou jednotku robota s krokovými motory a ověřit její funkčnost na vytvořeném prototypu podvozku s dvojicí krokových motorů.

Práce se zabývá problematikou pohybu robota pomocí kolového podvozku. Pro pohon kol lze použít několik typů motorů. Jedním z nich jsou krokové motory. Výhodou krokových motorů je definovaný úhel pootočení v jednom pracovním kroku. Hlavní nevýhoda spočívá ve složitějším řízení vyžadujícím mikroprocesorový řídící systém.

Dalším úkolem řídícího softwaru pro řídící mikropočítač je komunikace s nadřízeným systémem. Předpokládá se, že řídící jednotka bude přijímat povely po sériové lince v souladu se standardem stávající povelové ovládací robotické sítě používané v laboratoři mikroprocesorové techniky na VOŠ,SŠ – COP v Sezimově Ústí pod názvem Robo-COP. Jednotka zpětně odesílá nadřazenému řídícímu systému hlášení o svém stavu.

Vytvořený prototyp pohonné jednotky lze použít při konstrukci robotů s kolovým podvozkem. Řídící jednotku lze využít k ovládání i jiných typů bipolárních krokových motorů s proudovým odběrem do 1A na každý motor.





Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

ČIDLO ROBOTICKÉHO VIDĚNÍ

Pavel NÁCAL

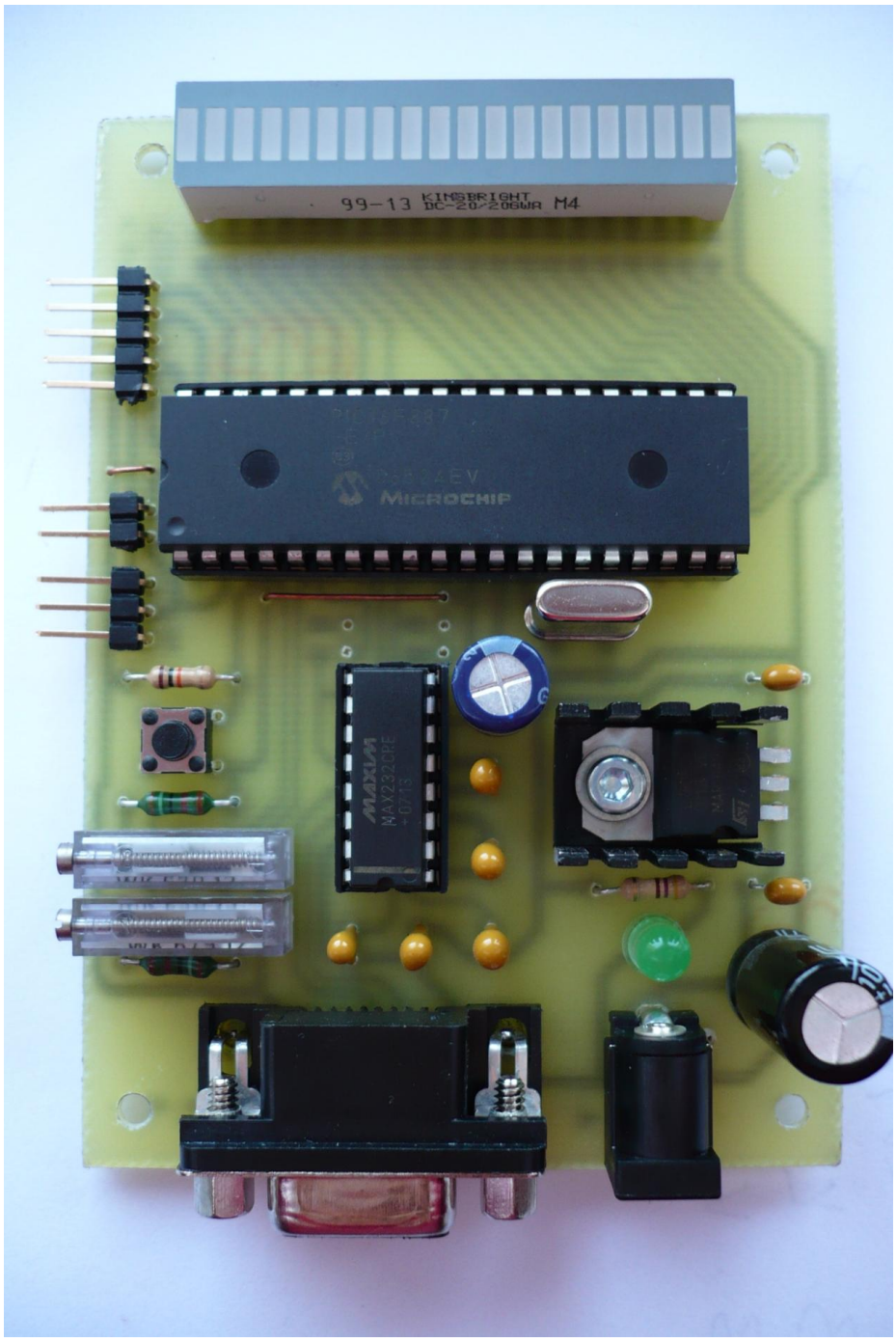
VYŠŠÍ ODBORNÁ ŠKOLA, STŘEDNÍ ŠKOLA, CENTRUM ODBORNÉ PŘÍPRAVY,
Budějovická 421, 391 02 Sezimovo Ústí

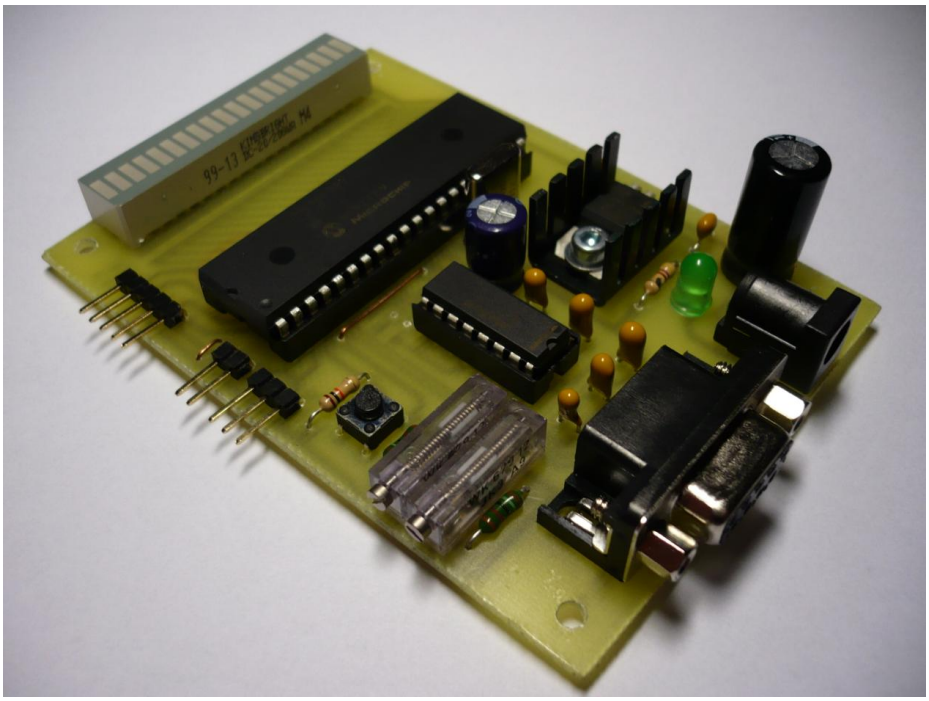
Práce se zabývá problematikou robotického vidění. Základem je využití malé a levné CCD kamery v ceně cca 800,-. Čidlo robotického vidění může zlepšit schopnost orientace robota v prostoru, a doplnit tak dotyková čidla, ultrazvuková čidla, infra čidla a další. Na VOŠ SŠ COP Sezimovo Ústí jsou v laboratořích mikroprocesorové techniky zpracovávány celkem tři různé roboty, které by bylo vhodné vybavit robotickým viděním. Jedná se o podvozek robota se dvěma krokovými motory, na kterém pracovala skupina studentů oboru Elektrotechnika v letech 2002-2004. Dále je to robot Minisumo z roku 2006 od studenta ET Lukáše Průchy a Robotická ruka z roku 2006 od studenta ET Milana Dudy.

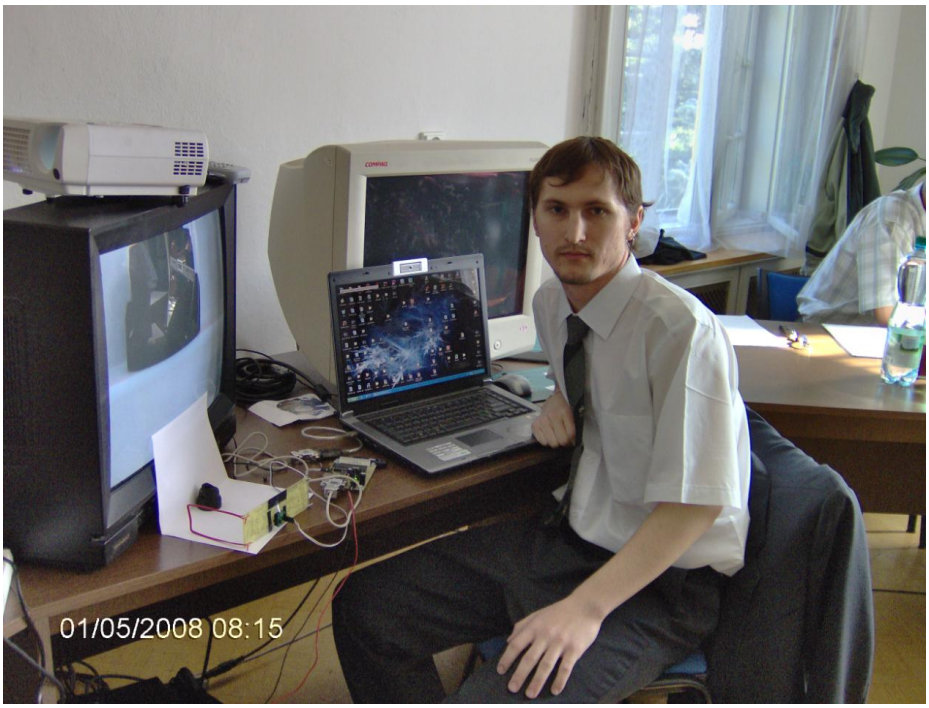
Hlavním úkolem je navrhnout a zkonstruovat HW čidlo umožňující realizaci robotického vidění s CCD kamerou. Čidlo je ovládáno jednočipovým mikropočítačem PIC firmy Microchip. Řídící jednočipový mikropočítač zajišťuje detekci videosignálu z výstupu CCD kamery, komunikuje s nadřízeným systémem po sériové lince RS 232 C a indikuje zjištěnou informaci na sloupci 20 LED. Čidlo je realizováno na jedné desce plošných spojů a jeho rozměry jsou minimalizovány s ohledem na předpokládané použití polohovací jednotkou nebo s některým z robotů.

Dalším úkolem práce je navrhnout komunikační protokol pro zapojení do stávající povelové robotické ovládací sítě, která je v laboratořích mikroprocesorové techniky provozována pod názvem Robo-COP. Hlavní činností řídícího SW pro jednočipový mikropočítač je sejmutí 20 obrazových bodů podle požadavku nadřízeného systému.

Čidlo je možné použít pro navádění pohybujících se robotů sledujících čáru. Po změně řídícího SW lze čidlo použít na další druhy vidění, jako je rozpoznávání tvarů, sledování obrysu, hledání překážek, sledování pohybujícího se předmětu atd…


99-13 KINGBRIGHT
DC-20/20GWA M4
MICROCHIP
MAXIM
MAX232CPE




01/05/2008 08:15

Středoškolská technika 2008

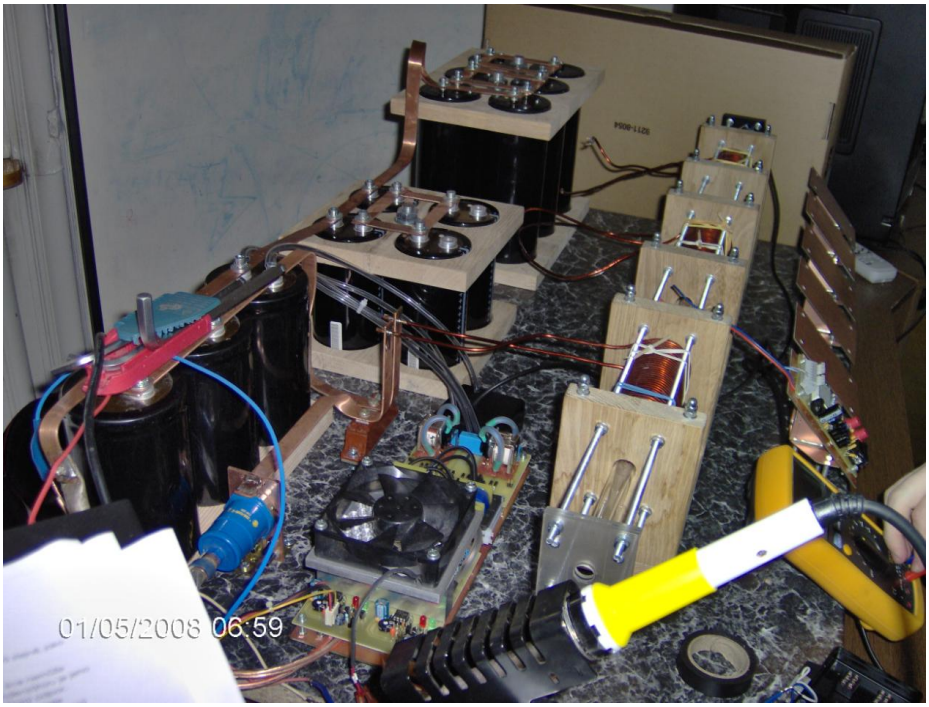
Setkání a prezentace prací
středoškolských studentů na ČVUT

Prototyp elektromagnetického děla

Martin RENDEK

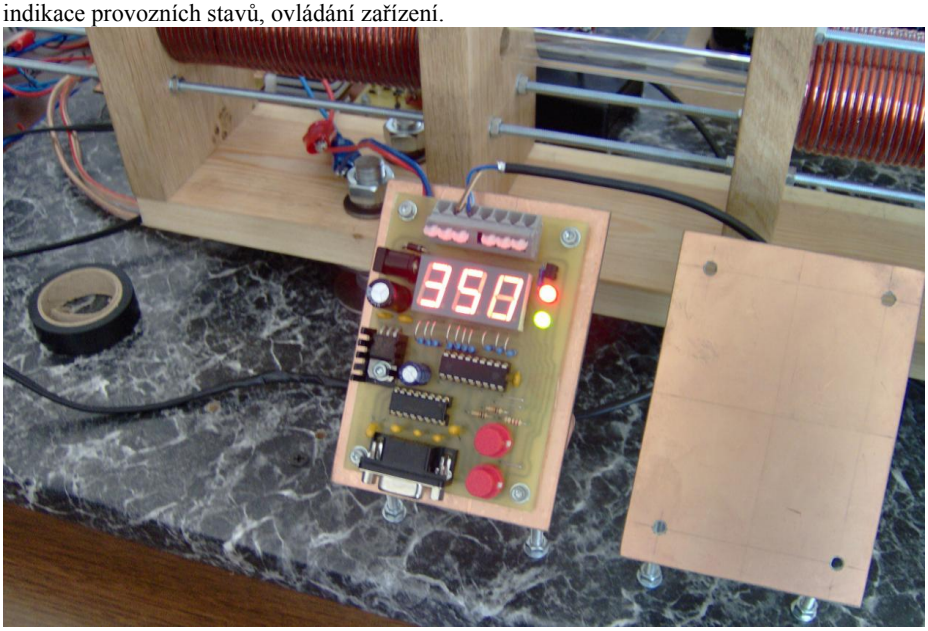
VYŠŠÍ ODBORNÁ ŠKOLA, STŘEDNÍ ŠKOLA, CENTRUM ODBORNÉ PŘÍPRAVY,
Budějovická 421, 391 02 Sezimovo Ústí

Práce se zabývá zkoumáním přeměny energie elektromagnetického pole na energii kinetickou. Nosnou myšlenkou práce je konstrukce elektromagnetického děla, kde k urychlení projektilu slouží výlučně energie magnetických polí generovaných soustavou urychlovacích cívek. Kromě teoretického popisu principu, výpočtů a návrhů se práce zabývá popularizací problematiky pro účely výukové pomůcky. Hlavním úkolem práce je analyzovat možnosti pro urychlení kovového projektilu na principu elektromagnetické indukce. Současně je nutné navrhnout a zkonstruovat funkční vzorek zařízení. Prototyp se stává z jednotky napájecího zdroje, kondenzátorových baterií, skleněné hlavně se třemi souose uloženými cívkami, polovodičového spínače a mikroprocesorového řídícího systému s čidly.



Dalšími úkoly je návrh a konstrukce vlastního řídícího systému s jednočipovými mikropočítači PIC a tvorba SW pro tento systém. Primární funkcí řídícího systému je precizní

časování spínání proudů do urychlovacích cívek v závislosti na poloze projektilu v hlavni. Sekundárními funkcemi jsou regulace velikosti napětí na kondenzátorových bateriích, indikace provozních stavů, ovládání zařízení.



Zařízení umožňuje vystřelovat kovové projektily různou rychlostí v závislosti na napětích kondenzátorových baterií. Je možné ho použít na demonstraci děla. Napětí všech kondenzátorových baterií lze měřit stejně jako úsťovou rychlost projektilu. Dále je možné na něm studovat vlastnosti přeměny energie elektromagnetické energie na kinetickou. Komerčnímu použití brání cena a hmotnost jednotlivých kondenzátorů, cena a technické provedení spínacích prvků.



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

MODEL AUTOMATICKÉ PŘEVODOVKY DSG

Autor:
Tomáš Pícha

Vyšší odborná škola, Střední škola, COP
Budějovická 421
391 02 Sezimovo Ústí

Anotace

Tématem práce je automatická převodovka DSG. Téma je zpracováno z několika pohledů. Jednak se autor zabývá problémem vývoje tohoto zařízení, dále jej zpracovává do podoby základního počítačového modelu.
Jako základní informační zdroj je využíván hlavně internet, ale také odborná literatura k použitým odborným softwarům.
Vybrané sestavy a podsestavy jsou řešeny z pohledu dynamiky proto, aby se v závěrečné práci dal vypracovat reálný 3D model pohybující se automatické převodovky s popiskami a s možností vysvětlit princip činnosti v reálu.
Na závěr autor přináší porovnání cíle projektu s dosaženými výsledky. Nechybí ani doporučení pro další využití práce ve škole.

Anotation

The main topic of my work is an automatic gearbox. A approached the problem from several points of view. Above all I del at with the porblem of development of such a machine. Besides other I transformed it into the basic
computer model.¨
As a basic source of informationin in a preparatory phase I mainly used not used not only the internet but also technical literature about appropriate swoftware. The seleced compositions are solved form the dinamic view to create real 3D model of moving automatic gearbox conteiningnotes and possibility of the princip explanation.
In the end I compared goals
of the project with reached results. There Is offered a recommandation how to use this work as a school requisite too.

1/ ÚVOD

1.1. Téma a širší vymezení problematiky

Tématem práce je vytvořit počítačový model automatické převodovky DSG a provést animaci funkce.

V širším vymezení problematiky se práce zabývá nejen samotným vznikem a vývojem automatické převodovky, ale snaží se také vysvětlit způsob řazení a funkci zařízení.

Vymezení základních pojmů a klíčových slov s kterými projekt operuje:
Převodovka – zařízení měnící počet vstupních otáček a krouticí moment či smysl.

Seznam zkratek s kterými práce operuje:
COP – Centrum odborné přípravy;
DSG - Direct-Shift Gearbox;
Převodová skříň – zařízení pro převod točivého momentu, změnu jeho strany otáčení a počtu otáček;
Kinematika – část mechaniky, která se zabývá klasifikací a popisem různých druhů pohybu, ale nezabývá se jeho příčinami;
Maya- profesionální program pro 3D grafiku americké firmy Autodesk Inc.;
OS – operační systém;
PC – (Personal Computer) osobní počítač;
HDRI – (High Dynamic Range Image) snímek ve vysokém rozlišení;
Blog – dikuzní formum na internetu.

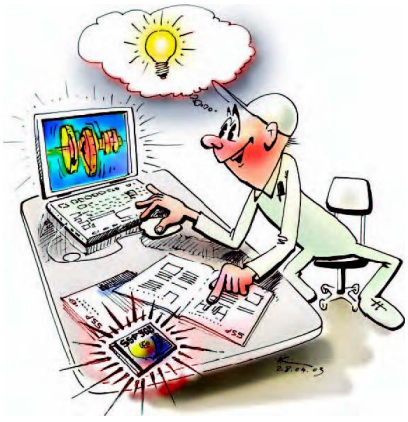
1.2. Cíl projektu

Cílem projektu je vytvořit počítačový 3D model automatické převodovky DSG následně ho rozhýbat, pomocí popisků a animace vysvětlit způsob řazení a zhodnotit její klady a zápory.

2/ TEORETICKÁ ČÁST

2.1.Historie problému

Klasické mechanické řazení s řadicí pákou a nutností citlivě ovládat spojkový pedál u mnoha vozidel nevyhovoval. Byla žádána převodovka, která bude fungovat samočinně. V USA se v osobních automobilech vyskytuje již snad od padesátých let. Řidiči v USA dnes využívají převážně auta s automatickou převodovkou.



V Evropě mají osobní automobily stále přímočinné mechanické převodovky. Automatické převodovky se ale od sedmdesátých let ve velké míře uplatňují u městských autobusů a u motorových vozů na železnici.
Evropské automobilky a zákazníci byli v otázce zda řadit ručně nebo nechat řadit automobil vždy poněkud skeptičtí. Přesto je německý koncern ZF jedním z největších výrobců automatických převodovek na světě. Letos uvedl na trh druhou generaci šestistupňové automatické převodovky a od roku 1965 má na kontě 10 milionů samočinných převodových ústrojí. V roce 1965 začínala ZF také s BMW, tehdy však šlo o třístupňovou automatickou převodovku pro model BMW 2000C, později také pro Peugeot, Alfu Romeo, Maserati, Lancii a Citroën. V roce 1982 byla představena čtyřstupňová automatická převodovka ZF s elektrohydraulickou řídicí jednotkou, v roce 1992 následovalo pětistupňové automatické ústrojí ZF s adaptivní logikou řazení.

Druhy automatických převodovek:

- hydrodynamická - základním prvkem je „pružné“ spojení přes hydrodynamický měnič (viz výše)
- hydrodynamická s uzamykáním - umožňuje uzamčení hydrodynamického měniče, tj. pevné propojení čerpadla a turbíny (po uzamčení má vozidlo mírně nižší spotřebu, mírně vyšší výkon a nelze přeřadit rychl. stupeň). Uzamčení je provedeno automaticky programem v závislosti na jízdních podmínkách a ovladačích (plyn, rychlostní stupeň …).
- hydromechanická - je poněkud pozměněná obdoba hydrodynamické, kterou používaly hlavně železniční vozidla (pravděpodobně tam není přímo HD měnič, ale nějaká jednodušší a robustnější obdoba - přesně nevím).
- mechanická automatická - je normální mechanická převodovka se suchou spojkou (jako u manuálního řazení), s tím rozdílem, že řidič spojku a řadící páku vůbec nemá a nepoužívá, řazení se provádí samočinně elektronicky a řídí jej počítač (je u některých kamionů a nově bude u dodávek značky Iveco). Elektronika zvládne přeřazení rychleji, než člověk, tudíž přerušení tahu motoru je jen na okamžik.
- dvouspojková automatická (DSG) - je normální mechanická převodovka se dvěma spojkami (zvlášť pro sudé a liché stupně), v každém okamžiku jsou zařazeny dva rychlostní stupně a samotné přeřazení představuje vypnutí první spojky a zapnutí druhé spojky. Obvykle s možností přímého řazení (obdoba sekvečního řazení).
Spojky i převodovky jsou ovládány hydraulicky. Za to zodpovídá mechatronika zabudovaná do převodovky DSG.
Pojem "mechatronika" se skládá ze dvou slov: mechanika a elektronika. A mechatronika je rovněž stejně koncipována: elektronická řídicí jednotka převodovky, různé senzory a hydraulická řídicí jednotka vytvářejí kompaktní jednotku. Řídicí jednotka využívá mimo jiné informace týkající se otáček motoru, rychlosti jízdy, polohy plynového pedálu a jízdního programu k tomu, aby byl zvolen optimální rychlostní stupeň a aby byl určen moment, který je pro přeřazení nejvhodnější. Hydraulická řídicí jednotka provádí pokyny týkající se řazení jako komplexní soubor jednotlivých akcí, které jsou co nejpřesněji koordinovány.

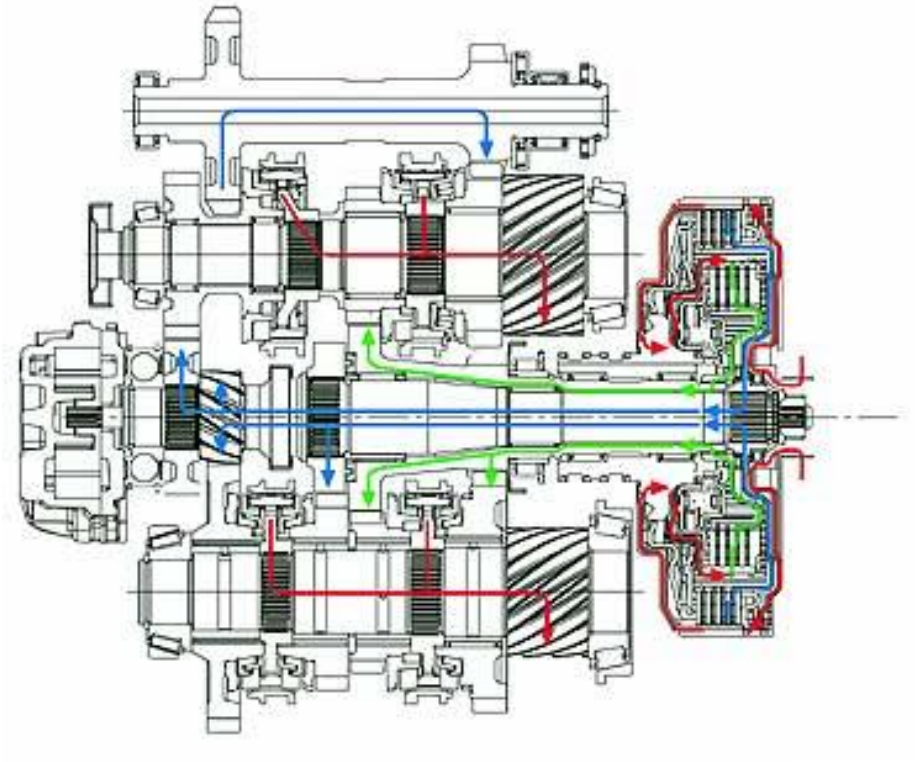
Pojmy:

- manuální řazení - řidič volí konkrétní rychlostní stupně (pokud chce přeřadit, musí přesunout páku z polohy „rychlostí stupeň 3“ na „r. s. 4“, přičemž používá spojku) - většina Evorpských automobilů.
- sekvenční řazení - řidič ovládá tlačítka nebo páčku pouze „nadřadit“ a „podřadit“ (pokud chce přeřadit, pouze zmáčkne tlačítko „nadřadit“, spojku nepoužívá) - některé závodní automobily, dosud i Formule 1.
- automatické - řidič se o řazení vůbec nestará, systémy vozidla si samy rozhodují, kdy budou a jak řadit - městské autobusy, automobily v USA, v Česku jen luxusní limuzíny; v Evropě jsou již automatické převodovky standardem, ať klasické (hydrodynamické) nebo poloautomatické (klasická manuální převodovka, ale řazení provádí počítač, řidič nemá spojku.

== Převodovky ==

Praga 2M70 - nejrozšířenější česká hydrodyn. převodovka, v autobusech ŠM 11, Karosa B 731, nejběžnějších motořáčcích řady 810. vyráběla se od poč. 70-tých let. Na konci 80-tých let již značně zastaralá a nespolehlivá. V motorových vozech na železnici slouží dodnes.
Voith - nejrozšířenější výrobce moderních hydrodyn. převodovek do současných autobusů

ZF - druhý nejrozšířenější po Voithu
Automatická převodovka slouží k zjednodušení řízení automobilu. Má i velký vliv na bezpečnost provozu, protože řidič se může plně věnovat volantu. Automatické převodovky jsou zpravidla čtyř nebo pětistupňové. Řidič může volit mezi jízdou vpřed, vzad a dále má k dispozici první a druhý rychlostí stupeň pro využití brzdného účinku motoru. Některé automatické převodovky jsou vybaveny i tzv. sportovním a zimním režimem. První jmenovaný přikazuje převodovce, aby řadila dříve nižší rychlostní stupně a vyšší později.
Dnes však již existují systémy jako například AGS (Adaptive Getriebesteuerung) pro přizpůsobivé řízení převodovky, kdy automatický systém rozhoduje o vhodném přeřazení podle jízdní situace, jízdního stylu řidiče a podmínek na cestě (BMW).



2.2. Analýza informačních zdrojů vybraných do projektu

Jelikož je tento vynález (princip řazení) poměrně mladý, používá se pouze dva roky, tak byly informační zdroje značně omezené. Veškeré informace byly čerpány z internetu mnohdy i zdlouhavý pročítáním velkého množství blogů. Modely byly tvořeny pouze z velmi nepřehledných a informačně nedostačujících obrázků. Veškeré hledání informací bylo velmi složité a zdlouhavé.

3/ VÝZKUMNÁ /PRAKTICKÁ/ ČÁST

3.1. Rozbor vlastního zadání – praktického problému

Koncept praktického řešení – empirického výzkumu se opírá zejména o následující východiska:

- Na středních školách stále chybí praktické příklady a didaktické pomůcky pro mezipředmětové vazby.

- Důležité je zejména hned po vysvětlení nebo během něj názorná ukázka chodu zařízení, což se u velkých strojů provádí těžko.V tomto případě přichází na řadu grafické ukázky a animace.

- Automatická převodovka je ve své podstatě mechatronický výrobek z automobilního systému a jedná se tedy o synergii několika vědních oborů – mechaniku, elektroniku, informatiku.

- Vybraná problematika je tedy při hlubším zkoumání velice složitá a rozsáhlá, proto bude nezbytné po teoretické přípravě vybrat si zúžený, v daném čase zvládnutelný, praktický problém a ten vyřešit. Práce se tedy nebude zabývat složitými výpočty hřídelí a ozubených kol, ale pouze věrným zobrazením pohybů a dílů automatické převodovky.

- Práce si klade za cíl vysvětlit s pomocí počítače základní způsob řazení nové generace převodovek a současně vytvořit animovaný 3D model.

- Vlastní řešení dokáže přiblížit některé ne tak docela pochopitelné způsoby převodování rychlostí v převodovce.

3.2. Stanovení postupných kroků řešení /dílčích cílů/

Žákovský projekt si pro praktickou část stanovil tyto dílčí cíle:

- Z prozkoumaných internetových zdrojů vůbec pochopit způsob přenosu kroutícího momentu a způsob řazení. Tyto podklady dále zpracovat do hrubého nástinu situace.

- Zhotovit 3D modely v programech používaných ve strojařském průmyslu a následně je převést do čistě grafických programů.

- Vytvořit funkční sestavy jednotlivých částí převodovky a nastavit správné parametry vizualizace.

- Vytvořit vhodné animace a videa s cílem co nejlépe simulovat postup řazení a vůbec celý chod zařízení.

Videa pak následně vhodně sestříhat, vytvořit ucelený materiál a dokončit didaktickou část.

3.3. Použitá metodika

Jaké metody byly použity?

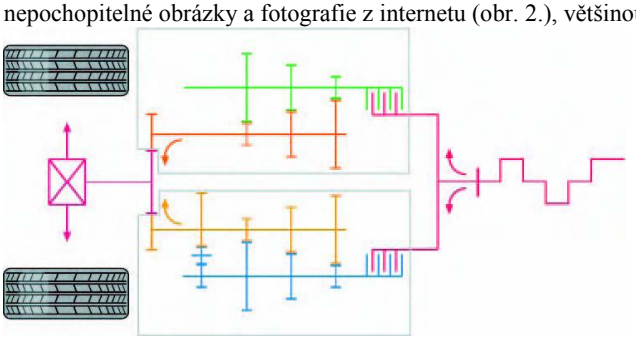
Praktická část problému byla založena ryze na problémové metodě. Postupným zpracovávání projektu jsem však dospěl k metodě heuristické a situační. Při vytváření animací se práce ubírala přes postupná řešení k zpětnému ověřování. Teoretická část byla tvořena čistě poznatky z zčásti praktické.
Jak byly zaznamenávány získané informace a výstupy praktické činnosti?
Záznam výstupů v průběhu práce vždy odpovídal dané činnosti. Protože se jednalo především o obrazové informace, tak šlo o převod obrazů do elektronické podoby prostřednictvím vhodného softwaru. Pro dokumentační účely byly snímány i obrazovky počítače a vkládány do textu projektu. Jako výstupy projektu pak slouží velké množství informací, které jsou jednak v tištěné podobě jako ukázky problematiky této práce a dále uloženy na nosiči CD.

3.4. Popis použitých technických zařízení

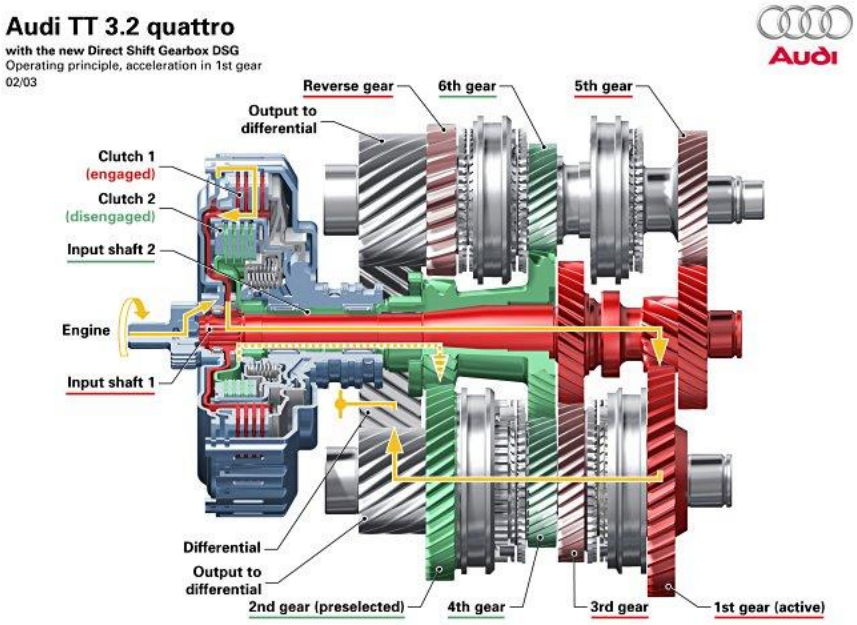
- počítač Asus A6Rseries CP-Celeron-M440, memory-1024, garfický akcelerátor-ATIReadon Express 200M s běžným software MS Windows XP v průběhu celého projektu;
- software Maya 2008 pro tvoření veškeré praktické části projektu;
- software SolidEdge 20 pro modely hnacích hřídelí.

3.5. Výsledky projektu – vlastní řešení problému

Základním materiálem byly ze začátku pouze schématické nákresy (obr. 1.), později mnohdy nepochopitelné obrázky a fotografie z internetu (obr. 2.), většinou s cizojazyčnými popisky.

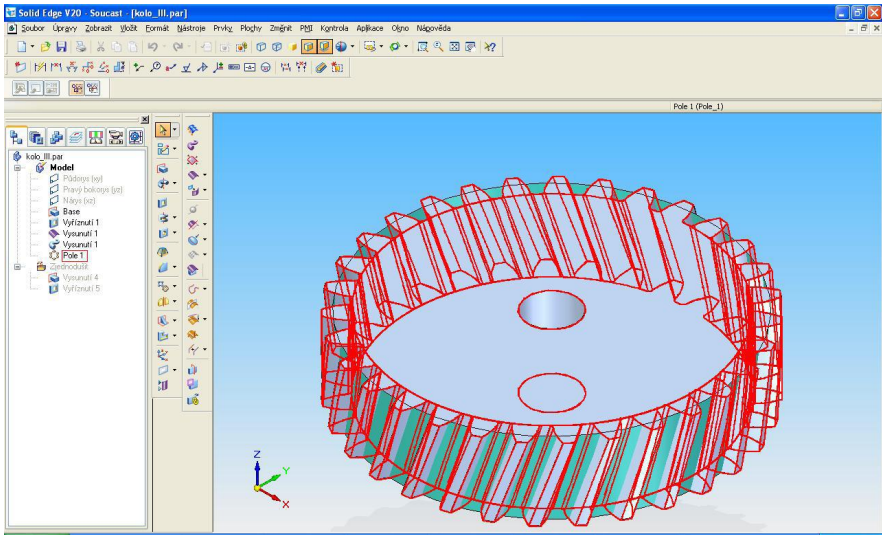


obr. 1. schématický nákres převodovky



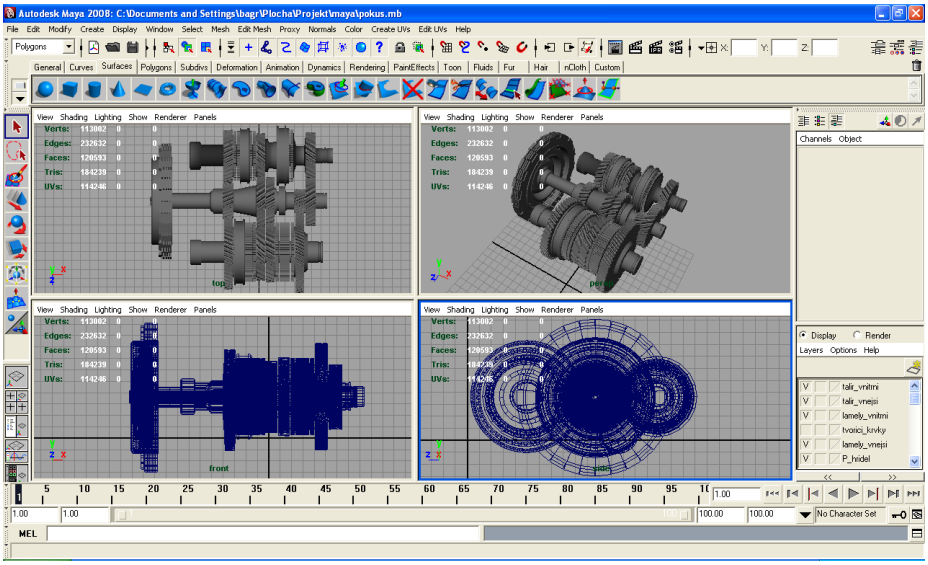
obr. 2. obrázek uspořádání součástí

Podle nich jsem se snažil vytvořit nejdříve modely součástí v strojařském programu (obr .3.) Solid Edge, které jsem následně převáděl do grafického programu Maya.



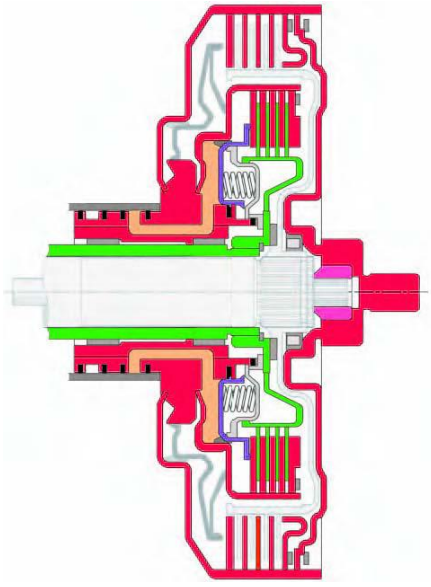
obr.3. vymodelované kolo v Solid Edge před grafickou úpravou

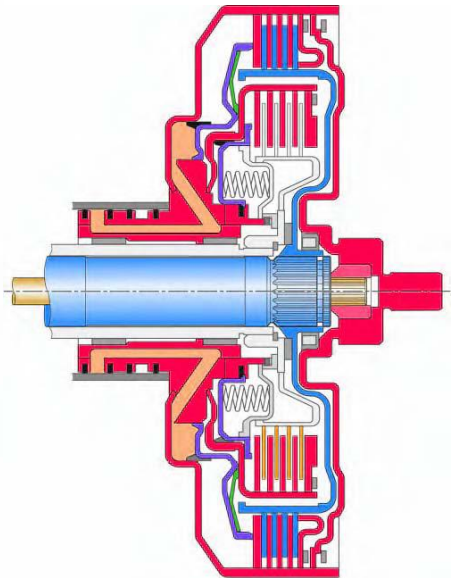
Tuto taktiku jsem po několika dnech vzdal. Byla zdlouhavá a hlavně velice náročná na hardware. Proto jsem všechny součásti začal modelovat rovnou v software Maya (obr. 4.). Práce se tím velice
usnadnila, odpadlo převádění do různých formátů a následné upravování.

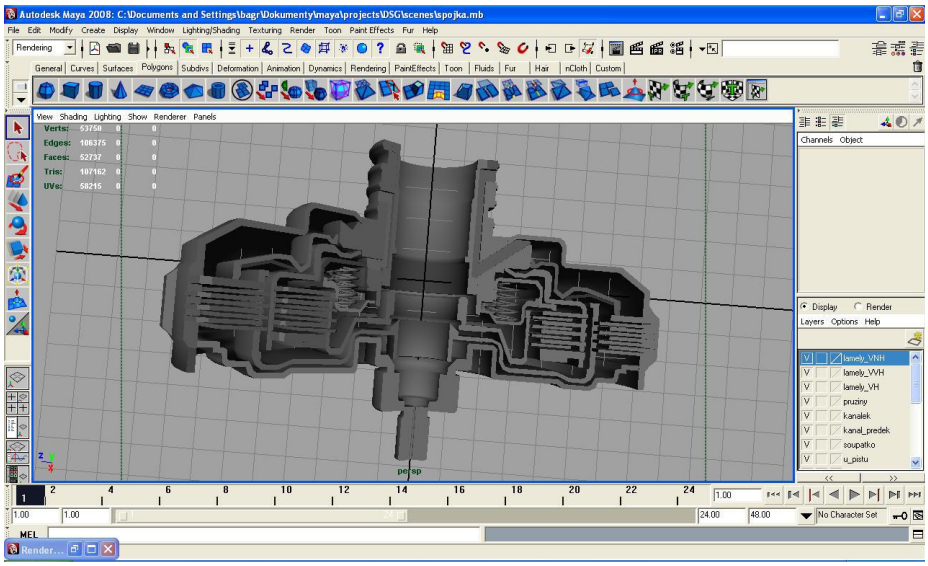


obr. 4. hrubý model převodovky – Maya

Samostatným problémem byla spojka, která je sama o sobě velice složitá a z dostupných informací (obr.5. a 6.) je možná pouze dedukce její činnosti. Chtěl jsem z ní znázornit jen lamely, ale protože byla vyvinuta specielně pro tento typ převodovky tak jsem jí po dlouhém bádání zapojil do projektu také (obr. 7.).

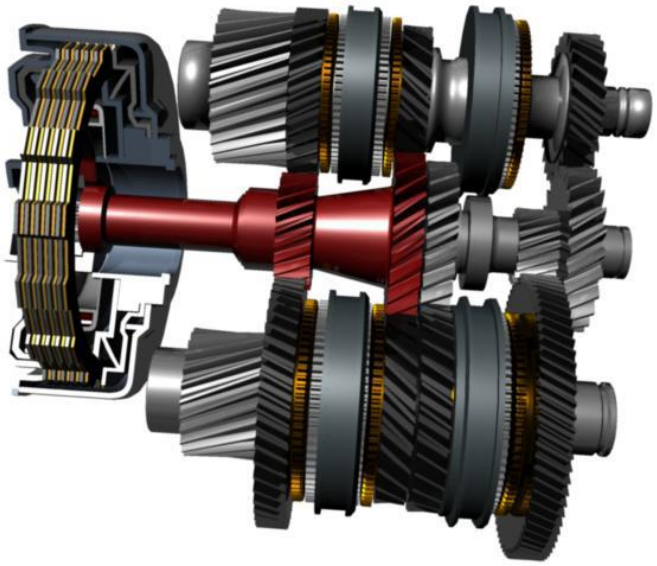

obr. 5. spojka – v záběru vnější hřídel


obr. 6. spojka – v záběru vnitřní hřídel

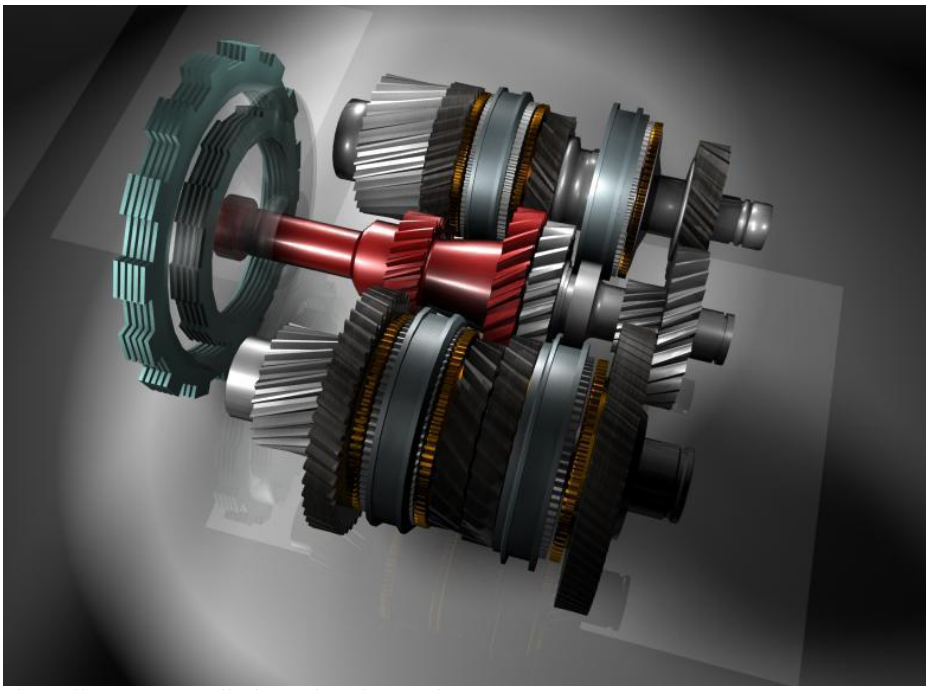


obr. 7. 3D model spojky – Maya

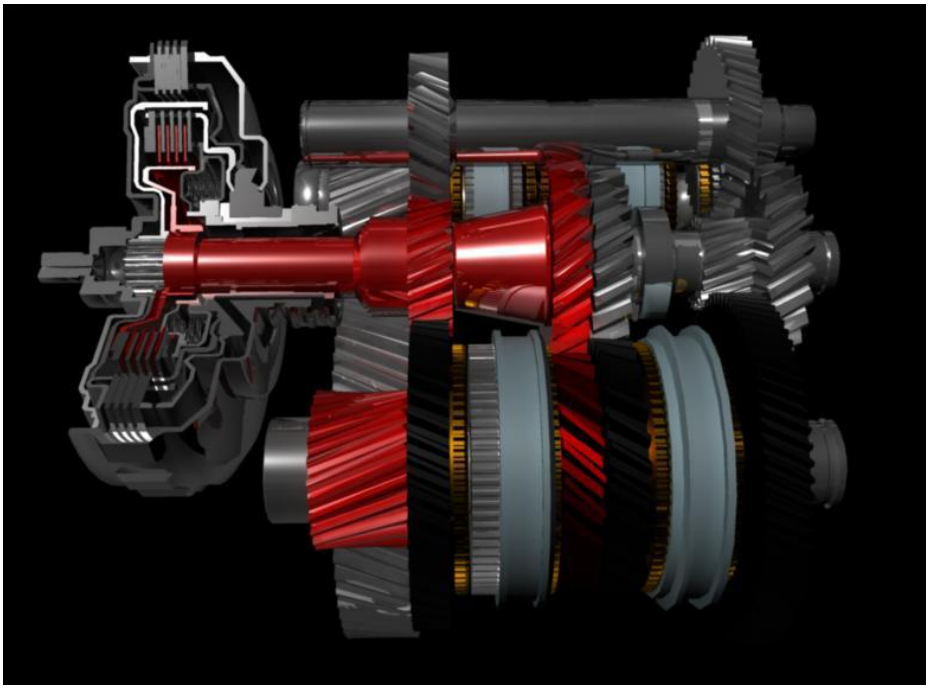
Potom následovalo texturování (obr.8.) a svícení. Zkoušel jsem několik možností, nakonec se jako nejlepší ukázala volby HDRI svícení(obr. 9. a 10.), je nejjednodušší a zároveň nejoptimálnější pro následný popis děje.



obr. 8. ještě nenasvícená převodovka

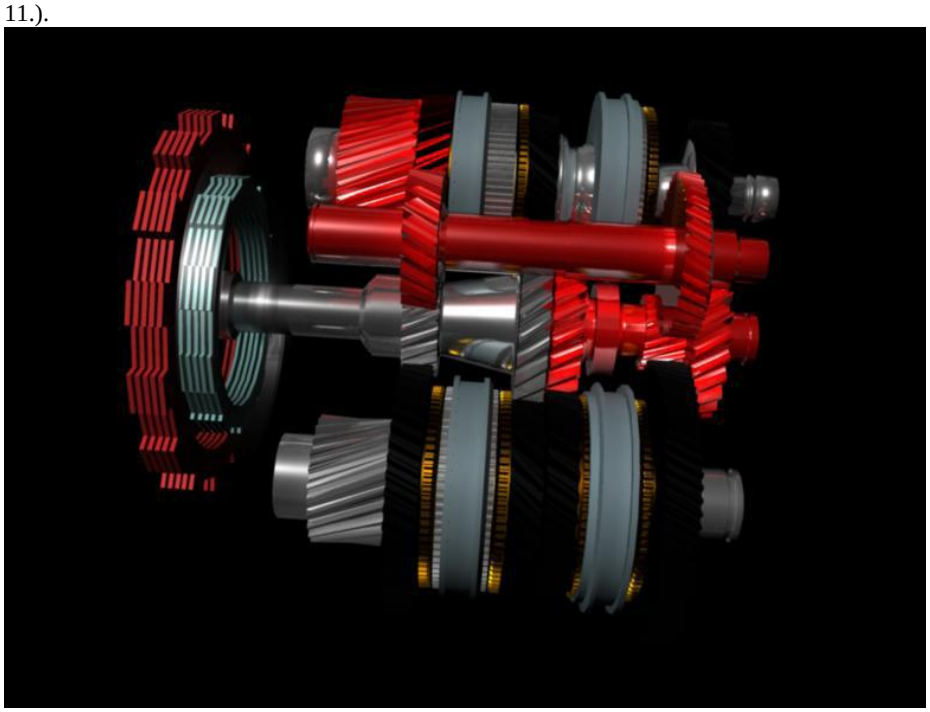


obr. 9. již ze stíny a odlesky – obyčejná světla



obr. 10. HDRI svícení – nedotexturovaná spojka

Následným zanimováním sestřiháním videa a zhotovením prezentace je projekt u konce (obr. 11.).



obr. 11. střih z animace – zpátečka

4/ ZÁVĚR

4.1. Shrnutí a zhodnocení výsledků

Zadáním projektu bylo vytvořit funkční model automatické převodovky DSG a pomocí animace předvést její činnost. Popisky navíc vysvětlit princip řazení. Dá se konstatovat, že projekt toto zadání naplnil, i když ne beze zbytku. Animaci jako výstup projektu pokládá autor za poměrně zdařilou.
O zásadnější teoretické závěry se práce nesnaží, neboť rozsah i hloubka zpracovávaných údajů pro ně není dostačující. Hlubší závěry by vyžadovaly více času a především využití daleko náročnějšího vědeckého aparátu z oblasti nově koncipovaného oboru MECHATRONIKA, který je synergií mnoha vědních oborů, především mechaniky (jemné mechaniky), elektrotechniky (elektroniky a senzoriky), informatiky.

4.2. Doporučení pro další využití a praxi

Zadáním bylo vytvořit didaktickou pomůcku. To bylo ve své podstatě splněno. Vytvořená animace by se dala použít v předmětu stavba strojů, konstrukční cvičení a jistě i v mechanice.
Úkolem bylo popsat způsob řazení, což bylo úspěšně splněno, ale tato převodovka skýtá daleko větší možnost na práci. Celá je řízena počítačovým čipem, který by byl určitě skvělým předmětem zkoumání. Dále je tam celá hydraulická soustava která by s jistotou mohla být příštím maturitním projektem.

5/ SEZNAM LITERATURY

Pro praktickou část
Riddell, D. Maya 5. Brno : Computer press , 2004. ISBN 80-7226-956-9
Dostupné na < http://auto.howstuffworks.com/dual-clutch-transmission.htm >
Dostupné na < http://www.kfz-tech.de/GetrDSG.htm >
Historie převodovky
Dostupné na < http://renault19f16.blogspot.com/2006/04/audi-a3-sportback-20-tdi-stronic-dsg.html>
Dostupné na < http://www.skoda-auto.sk/skoda_magazin/?2807>

6/ PŘÍLOHY

DVD – obrazová, vizualizační a animační část (JPG, AVI)

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Laserový zaměřovač

David Fenderl, Miloš Cigánek, Tomáš Blažek
Integrovaná střední škola technická a ekonomická, **Sokolov**

Za cíl této práce jsme si určili najít ve strojní výrobě něco, co je ekonomicky a pracovně náročné a následně to přepracovat tak, aby to zásadně ulehčilo situaci. Po konzultaci ve firmě Chemopetrol Pentar, kde nás pracovníci řádně provedli, nás už přesný cíl napadl. Pokud máme např. v chemické továrně tlakovou nádobu, je samozřejmé, že musí mít přívod nějaké kapaliny, nebo plynu. Právě to zajišťuje trubkový rozvod, to je přeci jednoduché. Nicméně u výroby tohoto průniku trubky s tlakovou nádobou už to tak jednoduché není. Tlaková nádoba může mít totiž různý tvar, od válce s klenutým víkem až po rotační kužel a jiné tvary. Trubka také nemusí být zasazena do nádoby kolmo, ale v různých úhlech a polohách.

Výroba těchto průniků jde řešit několika způsoby, ale ani jeden z nich není úsporný a nenákladný. Díra pro trubku jde vyrobit např. na horizontální fréze, to není příliš levné, ani rychlé a nejlépe to zrovna také nejde. Pak se také můžeme obrátit na konstruktéra, který nám může pomocí výpočtů navrhnout speciální šablonu, která se na danou nádobu přiloží a poté se obkreslí. Ta je ovšem skoro na každou nádobu jiná, a tak se užije třeba párkrát za několik let. Levné to také není, jednoduché už vůbec ne. V tom to právě spočívá, pro výrobu díry se musí dělat výpočty a jiné složité věci. Ještě tu existuje speciální matematický program, který nám vše vypočítá, ale o jeho ceně se raději nebudu vůbec zmiňovat. Když se otvor určí a orýsuje, tak se hořákem vypálí. V těch jednodušších případech průniku jsou lidé na dílně schopni si otvor orýsovat a hořákem vypálit, takto ručně je to nejrychlejší a nejlépe proveditelné.

My jsme se snažili vymyslet jednoduchou, levnou a časově úspornou rýsovací pomůcku. Výsledkem mnoha hodin strávených u rysů a technologických postupů a dalších prací je naše nynější konečné dílo. Chtěli bychom zdůraznit velký vděk pracovníkům firmy Chemopetrol Pentar ZBA, Horní Slavkov, bez jejichž spolupráce by tento projekt nemohl vzniknout. Dále bychom tímto chtěli vyjádřit velký vděk všem pracovníkům naší školy, kteří nám poskytli potřebné konzultace a odborné rady. Na této práci jsme si museli projít vším, co strojaře provází v jeho práci, a to od návrhu, přes technickou dokumentaci, technologické postupy, skoro až po samotnou výrobu. Zřejmě totiž bude vyroben prvotní model projektu, ale to je teprve v jednání. Zkusit si, jak to chodí v běžné praxi bylo pro nás nesmírné plus.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Elektromobily

Jakub Kříž
Integrovaná střední škola technická a ekonomická, **Sokolov**

Práce se zabývá elektromobily od jejich historie sahající až do poloviny 19. století, kdy trhaly světové rychlostní rekordy, čistě a tiše vozily lidi. Osvětluje důvody úpadku počátkem dvacátého století a jejich vzestupu v podobě EV1 koncem devadesátých let minulého století. Popisuje ohromnou manipulaci ze stran států a ropných společností, likvidaci pokročilých a existujících technologií v nadnárodních zájmech, lehkost s jakou jsou lidé manipulováni a lhostejnost s jakou k tomu přistupují. Je zde nastíněn současný stav v České republice a státech, které pro elektromobilismus něco dělají. Dále jsou zde rozebrány technické a konstrukční vlastnosti, motory, akumulátory, pojmy příliš neznámé v běžném moto-provozu a srovnány klady a zápory elektromobilů. Téže je zde seznam automobilek vyrábějící elektromobily a několik zajímavých je zde podrobněji popsáno a fotograficky doloženo.

Dalším důležitým bodem je snaha o vybudování nabíjecí stanice pro elektromobily v Sokolově, jakožto poměrně dobře situovaným místem z hlediska chybějící infrastruktury, která je celorepublikově budována velmi zřídka a téměř výhradně ze soukromých zdrojů jakožto snaze na oproštění se od závislosti na ropě, objektivnímu ulehčení přírodě i dopravě a často i jen ušetřit peníze. Umožní tak cestování přes ČR do Německa a naopak. V sousedních státech je elektromobilismus na daleko vyšší úrovni než u nás. Jedná se o vybudování poměrně nenákladné stanice nevyžadující žádnou obsluhu, zabírající dvě parkovací místa a možnost a bezprostřednost využívat elektrickou energii pro dopravu auty, e-koly a invalidních vozíků. Je zpracován seznam doprovodných aktivit možných dělat ve městě během doby nabíjení euta.

Přiložena je i mapka nabíjecích stanic, kontakt na společnost zajišťující dovoz eut a jejich bezproblémovou homologaci na Českých silnicích, jelikož znají již kličky v nehostinném silničním zákoníku, který bere čistou energii jako vetřelce. Výstavba této stanice je však v nedohlednu, jelikož městská rada není ve svých názorech jednotná, objektivně nejsou přesvědčeni o ekonomických příjmech a masového využití, staví se k tomu spíše jako o nadšenecký výkřik. S takovýmto přístupem však naše město asi nikdy nebude

zelené, jak město rádo hlásá. Na tomto místě zatím mé úsilí končí, však nekončím, nýbrž pokračuji.

Mapa nabíjecích stanic v České Republice





Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

UHLÍK A JEHO MODERNÍ VYUŽITÍ

Milan Daneček
Střední Průmyslová Škola - Nové Město nad Metují

Školní rok: 2007 – 2008

Datum: 2. 6. 2008

Konzultant: Mgr. Hana Gajdušková

Obsah

Úvod *2*

1. Uhlík jako prvek *3*

1.1. Historie uhlíku **3**

1.2. Výskyt **5**

1.3. Chemické a fyzikální vlastnosti .....

1.4. Významné sloučeniny .....

1.5. Nanotechnologie .....

2. Uhlíkové materiály .....

2.1. Úvod .....

2.2. Rozčlenění uhlíkových materiálů .....

2.3. Získávání uhlíkových materiálů .....

2.4. Černá keramika .....

2.5. Neokeramika .....

2.6. Uhlíková vlákna .....

2.7. Ohebný grafit .....

3. Strukturální, chemické a biologické vlastnosti uhlíkových vrstev na povrchu polymerů .....

Úvod

Pro moji ročníkovou práci jsem si vybral moderní uhlíkové technologie. O toto téma se zajímám již delší dobu. Uhlíkové materiály zažívají v současné době dynamický rozvoj. Podle mého názoru má uhlík a uhlíkové materiály velkou budoucnost.

Ve své práci se chci postupně zaměřit na obecně známé skutečnosti o uhlíku. V první části budu hovořit o jeho nejznámějších modifikacích a základních fyzikálně-chemických vlastnostech. V závěru první části bych se chtěl zaměřit na nejnovější objevy týkající se nových vlastností a modifikací uhlíku a využitím atomárního uhlíku v průmyslových odvětvích.

Ve druhé části se již budu zabývat složitějšími a zajímavějšími uhlíkovými materiály a technologiemi jejich výroby. Tyto materiály jsou běžně využívány a průmyslově vyráběny. Nalézají uplatnění v širokém spektru průmyslových odvětví (například trysky balistických střel).

Ve třetí části se společně podíváme na praktický příklad využití. Bude jím využití tenkých uhlíkových vrstev na povrchu polymerů. Takto upravené polymery mají největší využití v medicíně. A položme si otázku: Co je v lidském životě důležitější než zdraví?...

1. Uhlík jako prvek

1.1. Historie uhlíku

Uhlík využívají lidé již několik tisíciletí, avšak netušili, že je to právě uhlík, který se později projeví jako jeden z nejdůležitějších prvků. Mezinárodní název uhlíku "carbon" je odvozen od latinského carbo, čímž Římané označovali dřevěné uhlí.

V minulosti využívali lidé uhlík poněkud skrytě. Již v pravěku, lidé přišli na to, jak dokáže být malý černý kamínek užitečný. Jaká se v něm skrývá síla. No ano, mám na mysli uhlí. Lidově by se dalo říci, že uhlík je vlastně známý odjakživa, proto je velice těžké určit datum jeho objevení. Stejně těžké je určit datum jeho podrobnějšího popsání, protože vědci objevují stále nové modifikace a vlastnosti, je to až neuvěřitelné, kolik zajímavých vlastností skýtá prvek uhlík.

Lidé se naučili velice rychle využívat možností, které jim uhlík poskytoval. Nacházeli v přírodě úlomky kamenů, které na svou velikost měly velice malou hmotnost a navíc se daly bez větších obtíží zapálit. Ze začátku lidé využívali uhlí jako zdroj tepla, nebo k pečení uloveného masa. Později se naučili stavět pece, tavit železo a jiné kovy. Během času se význam uhlí zvětšoval. Uhlí se stalo klíčovou surovinou pozdějších dob.

V 19. století bylo uhlí nejdůležitějším palivem světa. Poháněl parní stroje, které stály na prahu průmyslové revoluce. I dnes se ještě uhlí spotřebovává ve velkém množství. Většina se ho spaluje v tepelných elektrárnách. Vyrábí elektřinu, která pokrývá velkou část světové spotřeby energií. Uhlí slouží ovšem také jako základní surovina pro výrobu železa a oceli. V dnešní době uhlík nachází uplatnění díky svým revolučním a velice zajímavým vlastnostem v mnoha oblastech lidské činnosti. Mnozí vědci ho právem nazývají prvkem budoucnosti.

Dalo by se říci, že uhlík je prvkem paradoxů. Je vodič jako kovy, ale může být i izolátorem jako keramika. Je propustný jako keramika, avšak může být i nepropustný jako skla. Ačkoli má mazací schopnosti , může být mechanicky odolnější než kovy, může také lépe odolávat teplotním šokům, dokonce lépe než keramika. Uhlík umožňuje výrobu kompozitních materiálů a může být užíván také jako textilie ve formě tkaniny. Může být vyráběn ve tvaru bloků o jednotkové hmotnosti několika tun , ale také jako vlákno o průměru několika mikrometrů, anebo jako ohebné pásky o tloušťce několika desetin milimetru.

Jeho vlastnosti se dají velmi dobře upravovat v souvislosti s rozdílným procesy při jeho zpracování, to nám dovoluje velice dobře přizpůsobit materiál specifickým požadavkům v daném oboru použití. Uhlíkové materiály jsou využívány prakticky ve všech průmyslových odvětvích a pro mnohá z nich jsou zcela nezbytnými. Zároveň však nachází uplatnění jako biomateriál v medicíně.

Dalo by se říci, že uhlík je starý a přesto velice moderní materiál. Úplně prvními uhlíkovými výrobky byly pravděpodobně tuhy do tužek, které byly zhotovovány v 16. století z bloků přírodního grafitu. Uhlíkové materiály jak je známe dnes začaly být využívány ve druhé polovině minulého století. Masivně se využívaly kupříkladu v elektrických obloukových lampách nebo pro elektrometalurgii. Z tohoto pohledu je uhlík starý materiál.

S odstupem doby se uhlík pro své mimořádné vlastnosti stával stále důležitějším prvkem. Již v této době nacházel uplatnění především v revolučních vynálezech a technologiích. Stal se tedy jakýmsi pomocníkem průmyslového rozvoje. Počátkem padesátých let minulého století byla završena důležitá etapa. Uhlík se stal nedílnou součástí výroby jaderné energie v grafitových reaktorech chlazených plynem. Toto období je pro uhlíkové materiály důležité zejména proto, že vyvolalo důležitou vlnu výzkumů a vzbudilo zájem vědeckého světa o tento materiál. Dalším impulzem pro výzkum uhlíkových materiálů nastal během 60. let současně s rozvojem letectví a kosmonautiky.

1.2. Výskyt

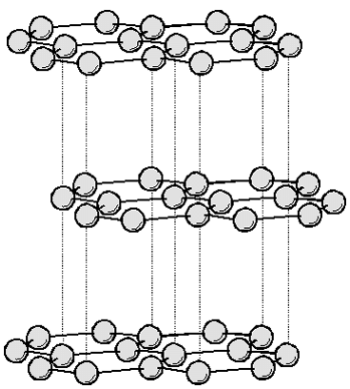
Při svém 0,087%[15] zastoupení je až na 13. místě v pořadí nejčastěji se vyskytujících prvků v zemské kůře. Celkový obsah uhlíku se odhaduje na 4112 tun. Elementární uhlík se vyskytuje ve dvou základních podobách a dalších třech, které jsou otázkou několika posledních desetiletí. Celkem jich je tedy 5. V té základní řadě to jsou grafit a diamant, tyto modifikace jsou opačných vlastností, což souvisí s uspořádáním jejich atomů C.

Grafit

Grafit je elementární formou uhlíku. Elementární znamená, že se skládá pouze z atomů uhlíku. Je to černošedý, kovově lesklý nerost s velmi nízkou tvrdostí (tvrdost 1). Je šupinkatý, lupenitý nebo zrnitý. Vyskytuje se v pegmatitových žilách, rulách, xylitech i svorech. Vzniká převážně přeměnou uhlí. Tento proces se nazývá prouhelnění.

Díky své návštěvě v českokrumlovském grafitovém dole jsem se mohl na vlastní oči přesvědčit, jak vznikal grafit a v jakých horninách se nachází. Jihočeské grafity jsou jedny z nejkvalitnějších na světě, avšak jejich ložiska nejsou schopna uspokojit tuzemské potřeby průmyslu, natož zahraniční. Každý vytěžený grafit má svou specifickou strukturu odlišnou podle místa naleziště. Významná naleziště jsou například ve střední Africe nebo Jižní Americe

Grafit neboli tuha (starší název) je tvořen uhlíkem krystalizujícím v šesterečné soustavě a patří mezi nejměkčí známé nerosty. Této vlastnosti se využívá např. při výrobě tužek, kde mletá tuha tvoří základní složku tyčinky určené pro psaní a kreslení. Jak jsem již zmiňoval, grafit krystalizuje v šesterečné soustavě, tyto základní vrstvy jsou na obr.1. dobře patrné. Mezi uhlíky v základních vrstvách jsou poměrně silné vazby, avšak mezi jednotlivými vrstvami jsou vazby několikanásobně slabší, to způsobuje tzv. stíratelnost tuhy. Tvrdost grafitu je 1. Je velmi dobře vodivý.



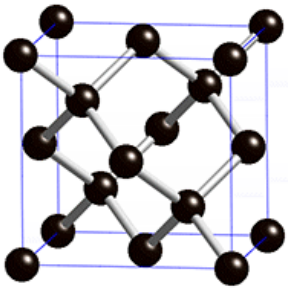
Obr. 1. Krystalická mřížka grafitu

Syntetický grafit se získává rozkladem některých kovových karbidů, jako např. karbidu křemíku, nebo srážením uhlíku rozpuštěného v roztaveném kovu nebo slitině, jako je železo nebo ferosilicium. V takovém případě získáme vločky grafitu, nazývané Kish-grafit. Jejich velikost je velmi blízká velikosti vloček přírodního grafitu.

Polykrystalický syntetický grafit je průmyslově vyráběn teplotním procesem za vysoké teploty (přibližně 3000°C) z grafitovatelných uhlíkatých sloučenin. Průmyslovou výrobou lze získat výrobky z polykrystalického grafitu, jejichž hmotnost přesahuje jednu tunu. Hustota grafitu je 2,3 g/cm^3.

Diamant

Diamant se svou tvrdostí řadí mezi nejtvrdší minerály vyskytující se na Zemi. Jakožto nejtvrdší nerost v Mohsově stupnici tvrdosti má hodnotu 10, je 140 x tvrdší než korund. Jedná se o krystalickou formu uhlíku. Tvoří hlavně jednotlivé krystaly oktaedrického, dodekaedrického nebo krychlového vzhledu. Diamant vzniká v zemské kůře za vysokých tlaků v ultrabazických vyvřelinách - kimberlitech.



Obr. 2. Jednotková buňka diamantové krystalové soustavy

Diamant se jako jediný drahý kámen vyskytuje ve všech barevných formách, nejčastěji je však bílý. Bývá šedý, neprůhledný-bort, karbonádo-bezbarvý, dokonale štěpný podle osmistěnu., nejvyšší tvrdost má na plochách štěpného osmistěnu (což umožňuje broušení diamantu diamantovým práškem). Jeho hustota se pohybuje v rozmezí 3,52-3,6 g/cm³. Má nejvyšší index lomu světla ze všech minerálů - přes 2,4 a nejvyšší tepelnou vodivost ze všech látek vůbec.

Diamant se nachází v mnoha barevných modifikacích. Barva diamantu může být způsobena chemickou nečistotou nebo strukturálními kazy v krystalové mřížce. Např.: tmavě modré zabarvení je způsobeno stopovými prvky bóru. (viz obr. 3.) Bílé diamanty s odstínem žluté jsou většinou levnější, zatímco růžové nebo modré diamanty můžou být velmi ceněny. Vybroušený diamant se nazývá briliant.



Obr. 3. Barevná modifikace diamantu

Diamanty se průmyslově těží v ohromných povrchových dolech. Nejznámější naleziště nalezneme v Indii (oblast Golgonda), dále je diamant hojně nacházen v Brazílii (stát Minas Gerais), Jižní Africe a Rusku. V Česku byl nalezen 2 x (Dlažkovice, Chrášťany).

Umělé diamanty se vyrábí dvěmi metodami. Jeden okruh metod je HPHT a další způsob se nazývá CVD. Při HPHT se diamanty tvoří za vysokých teplot a tlaků, kdežto CVD je metoda "kondenzace" par na

vhodném podkladu (známá mimo jiné z výroby nanotrubek), kterým je zde přímo diamant. Největší diamant byl vyroben vylepšenou metodou CVD. Vědci nám neprozradí, jakým způsobem metodu vylepšili, nelze tudíž říct, v čem spočívá její novost. Běžně je směs vodíku s metanem bombardována nabitými částicemi nebo přímo plazmou. To vyvolá složitou chemickou reakci a následně déšť uhlíku dopadající na podklad, kde ztvrdne ve formě diamantu

Tímto způsobem se získávají diamanty ve tvaru tenkých destiček. Tyto umělé diamanty mají použití např. v optice, elektronice a v jiných speciálních zařízeních.

Při udržování teploty a tlaku na nezbytné velmi vysoké úrovni, lze získat drahokamové diamanty o hmotnosti jednoho či více karátů.

Fosilní paliva

Mezi fosilní paliva se řadí uhlí, ropa a zemní plyn. Z fosilních paliv se získává energie (elektřina, teplo, světlo). Produkty z fosilních paliv jsou velice důležité pro rozvoj průmyslu a dokonce i pro život. Bez těchto paliv by se lidstvo nedostalo na takovou úroveň, na jaké se dnes nachází. Mnoho lidí by bez fosilních paliv v této době nedokázalo žít. Samozřejmě všechno má své klady a zápory a stejně tak i fosilní paliva. O výhodách jsem se již zmínil. A zápory? I těch je více. Používáním (produktů) těchto paliv se nemůžeme ubránit znečišťování životního prostředí, hlavně ovzduší, skleníkovému efektu, znečišťování řek, moří a oceánů, vlastně veškerého vodstva na zemi a zároveň s moři a oceány i pobřeží, čímž se zabíjejí různé organismy jak na zemi, tak i pod vodou.

- **Uhlí**

Uhlí je hořlavá hornina, která vznikla v průběhu desítek až stovek milionů let složitými procesy. Většina světových zásob uhlí se začala tvořit v období karbonu. Toto období začalo před 360 milióny lety a skončilo před 286 milióny let. Uhlí vzniklo z tropických pravěkých rostlin, rostoucích v bažinách.

Trouchnivěním, rašeliněním (za vzniku metanu - tzv. bahenního plynu) a pozvolným uhelnatěním rostlinných látek v močálech za nedokonalého přístupu vzduchu při vlhkém a teplém podnebí pozbývaly tyto látky vodík a kyslík, vytvářely uhlík, ztrácely rostlinnou strukturu a stávaly se tmavší.

Kvalita uhlí jako paliva závisí především na obsahu uhlíku, proto se uhlí dá rozdělit do několika skupin.

Druhy uhlí

1) **lignit** – Označuje se jako hnědé uhlí, je nejméně kvalitní druh uhlí. Užívá se výhradně pro parní výrobu elektřiny. Leštěné bylo využíváno jako ozdobný kámen od doby železné. Obsahuje 40-60% uhlíku.
2) **hnědo-černé** - Jeho vlastnosti spadají mezi hnědé a černé uhlí. Využívá se výhradně pro parní výrobu elektřiny.
3) **černé uhlí** - Má vysokou hustotu, jeho barva je obvykle černá až hnědočerná s přesně známými vlastnostmi. Obsahuje 75% až 95% uhlíku.
4) **antracit** - Nejkvalitnější uhlí. Používá se na vytápění. Má největší procento uhlíku až 94% a nejméně prchavých látek má.

Další produkty úzce související s uhlím

Koks je pevný produkt vyráběný z nízkopopelového, nízkosirného černého uhlí, ze kterého jsou odstraněny prchavé složky v peci s omezeným přístupem kyslíku při teplotách kolem 1000 °C. Koks je téměř čistý uhlík. Používá se hlavně při výrobě surového železa ve vysokých pecích a jako palivo. Koks se používá jako palivo a jako redukční činidlo např. ve vysoké peci. Koks z uhlí je šedý, tvrdý a pórovitý a má výhřevnost 29,6 MJ/kg. Petrolejový koks je pevný zbytek získaný rafinací ropy, obsahuje ale příliš mnoho nečistot na to, aby mohl být využíván v metalurgických technologiích také jako palivo.

Saze vznikají zvláště při spalování na uhlík bohatých organických paliv za nedostatku kyslíku. Hromadí se v komínech, automobilových výfucích (hlavně automobilů se vznětovým motorem) a dalších površích vystavených kouři. Saze byly využívány jako černé barvivo v barvách a inkoustech. Dnes už se ještě široce užívají v tiskařských inkoustech, tonerech pro xerografii, laserových tiskárnách, a v chemickém průmyslu. Jsou také využívány jako potravinářské barvivo, například lékořicové bonbóny. Nejdůležitějším způsobem využití je vulkanizace kaučukových pneumatik. Toto použití spotřebuje 85% na trhu uplatněných sazí.

Volbou vhodné technologie lze vyrobit saze odlišných vlastností. Sazí se vyrábí několik druhů, jejich názvy souvisí s technologickým způsobem výroby. Např.: acetylenové saze se připravují pyrolýzou acetylenu podle rovnice:

$$n\ HC \equiv CH \rightarrow C_{2n}\ (\text{amorfní uhlík}) + n\ H_2$$

- **Zemní plyn**

Zemní plyn je přírodní směs plynných uhlovodíků s převažujícím podílem metanu CH_4 a proměnlivým množstvím neuhlovodíkových plynů (zejména inertních plynů). Z vyšších uhlovodíků zemní plyny obsahují hlavně nasycené uhlovodíky, které jsou za normálních podmínek plynné - etan, propan a butan. V některých ložiscích obsahují zemní plyny i uhlovodíky, které jsou za normálních podmínek kapalné (od pentanu výše), které se při úpravě oddělují jako plynový kondenzát. Jejich směs se nazývá gazolín nebo přírodní benzín.

Zemní plyn se velmi často vyskytuje spolu s ropou (naftový zemní plyn – obsahuje větší procento metanu 97%) nebo s uhlím (karbonský zemní plyn – obsahuje nižší procento metanu 92%).

Těžený (přírodní) zemní plyn se podle složení dělí do čtyř základních skupin:

1) **zemní plyn suchý (chudý) -** obsahuje vysoké procento metanu (95 - 98%) a nepatrné množství vyšších uhlovodíků
2) **zemní plyn vlhký (bohatý)** – kromě metanu obsahuje vyšší podíl vyšších uhlovodíků
3) **zemní plyn kyselý** - je plyn s vysokým obsahem sulfanu (H_2S), který se v úpravárenských závodech před dodávkou zemního plynu do distribučního systému odstraňuje
4) **zemní plyn s vyšším obsahem inertů** - jedná se hlavně o oxid uhličitý a dusík

- **Ropa**

Ropa byla známa pod různými názvy vyžívána již od starověku. Podle řeckých i římských historiků v 7. století př. Kr. Asyřané a později i Peršané ji těžili ze studní a rozdělovali na různé frakce. Rozlišovali dokonce ropu světlou od tmavé. Světlé říkali "nafata", což značilo "prosakující kapalina". Číňané ropu destilovali snad od 11. století před Kristem. Už v té době prý uměli vrtat do hloubek až 1000 metrů.

Ropa (též surová nafta, zemní olej) je hnědá až nazelenalá hořlavá kapalina tvořená směsí uhlovodíků, především alkanů a aromatických uhlovodíků.

Nachází se ve svrchních vrstvách zemské kůry – nejčastěji v oblasti kontinentálních šelfů. Je základní surovinou petrochemického průmyslu. Naleziště ropy jsou pod nepropustnými vrstvami, v hloubkách až 8 km pod zemským povrchem. Ropa při těžbě buď vyvěrá pod tlakem, nebo je čerpána.

Koloběh uhlíku

Uhlík je spolu s kyslíkem a vodíkem základním stavebním prvkem organických látek. Koloběh tohoto hlavního biogenního prvku je jedním z klíčových cyklů podmiňujících život na Zemi. Je přítomen především jako oxid uhličitý CO_2 v atmosféře, v rozpustných uhličitanech ve vodě a také ve formě uhličitanů (zejména $CaCO_3$) ve vrstvách vápenců. Do vody se CO_2 dostává srážkami; výměna CO_2 mezi vodou a ovzduším se děje difúzí ve směru koncentračního spádu. CO_2 uniká do ovzduší také z uhličitanů, např. při zvětrávání vápenců. Do koloběhu oxidu uhličitého zasáhl také člověk spalováním fosilních paliv (uhlí, nafty) a zvýšil koncentraci CO_2 v atmosféře již zhruba o 30 % jeho původního množství. Část je obsažena v tělech žijících organismů, v odumřelých tělech v podobě humusu, ve fosilních palivech.Vzdušný CO_2 je v procesu fotosyntézy zdrojem uhlíku pro zelené rostliny. Část je ho zpět vydýchána, část je spotřebována ke stavbě biomasy. Rostlinná těla se stávají potravou býložravců, pak masožravců. Z odumřelých těl se uhlík dostává do půdy, kde je za účasti rozkradačů rozložen na jednoduché organické látky. Z půdy je CO_2 uvolněn zpět do atmosféry.

1.3. Chemické a fyzikální vlastnosti

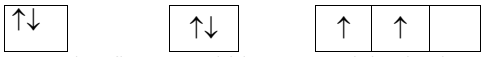
Uhlík je nekovový prvek s atomovým číslem 6. Relativní atomová hmotnost je 12,01. Teplota tání kolísá kolem 3500°C (za normálního tlaku). Teplota varu byla stanovena na
4027 °C (4300 K)[10]. Uhlík se rozpouští jen v roztaveném železe, niklu, kobaltu a v platinových kovech.

Elektronová konfigurace v anorganických sloučeninách

$_{6}C \quad 1s^2 \quad 2s^2 \quad 2p^2$

nebo jiné znázornění pomocí rámečků

$_{6}C$ 1s 2s 2p

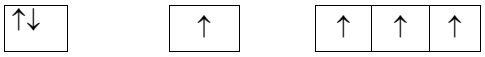


Elektronová konfigurace uhlíku v organických sloučeninách je odlišná. Uhlík je v tzv. excitovaném stavu, jeden elektron přeskočí na dráhu energeticky vyšší

$^{*}_{6}C \quad 1s^1 \quad 2s^2 \quad 2p^3$

pomocí rámečků

$^{*}_{6}C$ 1s 2s 2p



tím se uhlík stává v organických sloučeninách čtyřvazným.

Elementární uhlík se vyskytuje ve dvou základních podobách a dalších třech, které jsou otázkou několika posledních desetiletí. Celkem jich je tedy 5.

V té základní řadě to jsou grafit a diamant. Obě tyto modifikace jsou opačných vlastností. Nejvíce se toto tvrzení dotýká jejich tvrdostí a s tím úzce souvisejícím uspořádáním jejich krystalové mřížky.

V další řadě můžeme uvést fullereny, nanovlákna a novinku posledních dvou let uhlíkovou nanopěnu. Podrobněji se této druhé řadě budu věnovat v další části mé práce.

S kyslíkem se uhlík nedokonale spaluje na oxid uhelnatý $2C + O_2 \rightarrow 2CO$ nebo dokonale na oxid uhličitý $C + O_2 \rightarrow CO_2$, přičemž z 1g uhlíku vznikne okolo 33kJ tepla. Rozpouštěním oxidu uhličitého ve vodě vzniká kyselina uhličitá $CO_2 + H_2O \rightarrow H_2CO_3$.

Jako katalyzátor zastává uhlík významnou úlohu při přeměně vodíku na helium[16], přičemž se jedná o cyklus více složitějších reakcí. Schopnost uhlíku tvořit miliony

sloučenin se stala podmínkou existence života na Zemi. Touto schopností uhlíku se zabývá organická chemie.

S uhlíkem a jeho vlastnostmi je úzce spjatá tzv. Avogadrova konstanta. Avogadrova konstanta vyjadřuje počet částic v jednotkovém látkovém množství (v 1 molu). Je definována jako počet atomů v 0.012 kg izotopu uhlíku C_6^{12}, tedy stabilního izotopu, který obsahuje v jádře šest protonů a šest neutronů. Je pojmenována po italském fyzikovi Avogadrovi a její hodnota je $6{,}022 \cdot 10^{23}\ mol^{-1}$.

1.4. Významné sloučeniny

- **Oxid uhelnatý**

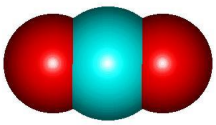
Oxid uhelnatý je bezbarvý plyn bez chuti a zápachu, lehčí než vzduch, jedovatý. Ve vodě je málo rozpustný. Je obsažen ve svítiplynu, v generátorovém a v kychtovém plynu (vysoké pece), dále ve výfukových plynech spalovacích motorů. Má silně redukční vlastnosti, které se využívají především v procesech vysokých pecí. V přírodě je přítomen v nepatrném množství v atmosféře, kde vzniká především světelným rozkladem oxidu uhličitého působením ultrafialového záření, dále jako produkt nedokonalého spalování fosilních paliv i biomasy. Je také obsažen v sopečných plynech.

Oxid uhelnatý hoří modrým plamenem a spaluje se na oxid uhličitý

Oxid uhelnatý je velice jedovatý. Jeho jedovatost se způsobena silnou reaktivností s hemoglobinem (červené krevní barvivo), s nímž vyváří karboxyhemoglobin, čímž znemožňuje přenos kyslíku v podobě oxyhemoglobinu z plic do tkání. Vazba oxidu uhelnatého na hemoglobin je přibližně dvousetkrát silnější než kyslíku a proto jeho odstranění z krve trvá mnoho hodin až dní. Toto je podstatou jednoho ze škodlivých vlivů kouření.

- **Oxid uhličitý**

Oxid uhličitý je bezbarvý plyn bez chuti a zápachu, který je tvořen jedním atomem uhlíku a dvěma atomy kyslíku (viz obr. 4.). Při ochlazení pod –80°C plynný CO_2 mění své skupenství za vzniku tuhé látky (desublimuje), tato látka se nazývá suchý led. Průmyslově se vyrábí frakční destilací zkapalněného vzduchu. Teplota tání -56,7°C (při tlaku 0,5MPa), teplota varu 20°C. Oxid uhličitý má asi 1,5 x vyšší hustotu než vzduch, proto má ve vyšších koncentracích (např. přirozené vývěry ze země) tendenci hromadit se při zemi. Vyplňuje také podzemní prostory, jako jsou studny a sklepy. Oxid uhličitý je nedýchatelný a ve vyšších koncentracích může způsobit ztrátu vědomí a smrt.



Obr. 4. Molekula CO_2

Oxid uhličitý se využívá k nejrůznějším průmyslovým účelům, to se týká jeho využití v plynném i tuhém (a v menší míře kapalném) skupenství.

Všichni dobře znají použití CO_2 při sycení limonád, minerálních vod a dalších nápojů. Suchý led se často používá v potravinářství, ale i v jiných odvětvích jako chladící médium. Inertní vlastnosti oxidu uhličitého umožňují jeho použití jako ochranného plynu při svařování. Často také představuje náplň hasicích přístrojů, zejména používaných pro hašení elektrických zařízení.

- **Kyanovodík**

Chemická značka je HCN. Je neobyčejně jedovatý. Roku1782 jej objevil švédský lékárník Carl W. Scheele. Je to bezbarvá kapalina. Teplota varu je 25,7°C, teplota tuhnutí je -13,4°C. Ve vodném roztoku je velmi slabou kyselinou. V přírodě se vyskytuje v hořkých mandlích a v jádrech některého ovoce (hořkomandlový olej). Jeho soli se nazývají kyanidy. Ve vodě dobře rozpustné a velmi jedovaté. Př.: kyanid draselný, KCN (tzv. cyankáli), a sodný NaCN. Obzvláště tento se používá při získávání zlata kyanidovým způsobem, na galvanické lázně a k získávání drahých kovů.

- **Kyselina uhličitá**

Vzniká rozpouštěním oxidu uhličitého ve vodě. Od kyseliny uhličité, kterou není možné z roztoku izolovat, se odvozují dvě řady solí:

Hydrogenuhličitany, pro které je charakteristický **hydrogenuhličitanový** anion HCO_3^-

Uhličitany, obsahující anion **uhličitanový** CO_3^{2-}.

Nejznámějšími uhličitany jsou například uhličitan sodný Na_2CO_3, který známe pod názvem soda; uhličitan draselný K_2CO_3 neboli potaš a uhličitan vápenatý $CaCO_3$, který se v přírodě vyskytuje ve formě vápence a používá se k výrobě páleného vápna.

- **Vápenec**

Uhličitan vápenatý, $CaCO_3$ je bílá krystalická látka. V přírodě se vyskytuje ve formě vápence. Termickým rozkladem (zahříváním) se uhličitan vápenatý rozkládá za vzniku oxidu vápenatého a oxidu uhličitého: $CaCO_3 \rightarrow CaO + CO_2$

Popsaný proces probíhá ve vápenkách a bývá označován jako pálení vápna. Pro oxid vápenatý se užívá i název pálené vápno. Pro hydroxid vápenatý, který vzniká jeho reakcí s vodou, se užívá označení hašené vápno.

- **Karbidy**

Karbid křemíku a nitrid křemíku, karbid boru a nitrid boru jsou nejdůležitějšími zástupci tzv. **neoxidové keramiky**. Neoxidová keramika je chemicky velmi stálá za vysokých teplot. Výjimkou je špatná odolnost proti oxidační atmosféře, v ostatních směrech je jejich chemická odolnost vynikající. Materiály obsahující sloučeniny křemíku mohou být do 1600°C použity i v oxidační atmosféře, díky vytvářené nereaktivní vrstvě SiO_2 na jejich povrchu. Pro technické použití jsou zvláště důležité jejich mechanické vlastnosti, vysoká pevnost a tvrdost, ve kterých převyšují vlastnosti oxidové keramiky, zejména při teplotách nad 1000°C. V mé práci se budu dále zabývat jen částí neoxidové keramiky, a sice karbidem křemíku a karbidem bóru, protože jsou nejpoužívanější. Tyto materiály mají v poslední době stále větší využití díky jejich vysoké teplotní stálosti a nereaktivnosti.

Karbid křemíku

Technické využití pro SiC bylo původně jako vynikající brusivo, díky jeho tvrdosti (9,5 leží mezi tvrdostí diamantu a Al_2O_3 korundu) a také zvláštní lámavosti při které vznikají velmi ostré řezné hrany.

Budoucnost keramiky z karbidu křemíku se očekává v jejím použití při výrobě teplotně vysoce namáhaných částí strojů a zařízení, jako v plynových turbínách a vznětových motorech. Díly mohou bez chlazení pracovat až do teploty 1400°C a jsou o 60% lehčí než vysokoteplotní slitiny z kovů.

Karbid bóru

Karbid bóru se vyrábí rozkladem oxidu boritého uhlíkem v elektrické odporové peci při 2400°C. Jemný disperzní prach karbidu boru se získává reakcí oxidu boritého s hořčíkem nebo hliníkem za přítomnosti uhlíku. Tento se zpracovává na keramické výrobky, které nacházejí použití při výrobě pancéřových desek a ochranných štítech bojových letadel, nebo v jaderných elektrárnách jako stínicí materiál pro neutrony.

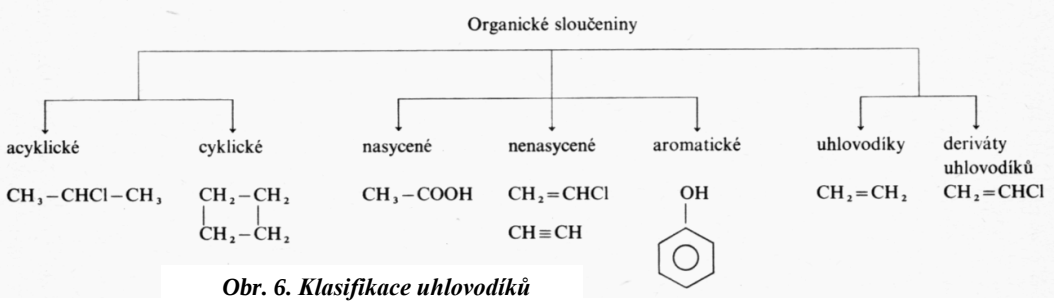
Zajímavost: karbid boru a borid berylia byly zjištěny jako nejúčinnější materiály v neprůstřelných vestách.

Tyto materiály jsou známé již delší dobu, avšak přesto se dají považovat za moderní, protože jak nám předchozí odstavce naznačují mají stále nezastupitelné uplatnění v moderních technologiích a progresivních průmyslových odvětvích.

- **Uhlovodíky**

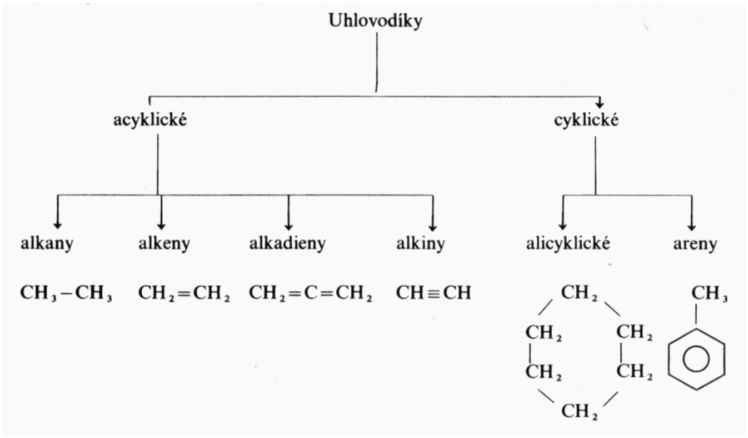
Organické sloučeniny dělíme do dvou velkých skupin. První skupina jsou **uhlovodíky**, ty obsahují jen uhlík a vodík. Druhá skupina jsou **deriváty uhlovodíků**, které obsahují kromě atomů uhlíku a vodíku i atomy jiných prvků. Deriváty vznikají spojením uhlovodíkového zbytku a funkční skupiny.

Uhlovodíky se dělí podle různých kritérií, pro názornost nám poslouží následující obrázky (Obr. 5., 6.).



Obr. 6. Klasifikace uhlovodíků

Obr. 5. Klasifikace organických sloučenin



Alkany

Jsou to nasycené uhlovodíky – obsahují jen jednoduché vazby, $C_1 - C_4$ jsou plyny, $C_5 - C_{17}$ – jsou kapaliny, $C_{17}>$ - jsou tuhé látky. Fyzikální vlastnosti – teplota tání a varu stoupá se stoupajícím počtem atomů uhlíku. Alkany s nevětveným řetězcem vyšší teploty varu a nižší tání než alkany s větveným řetězcem (Van der Waalsovy síly). Do této skupiny patří například methan, ethan,…

Alkeny

Alkeny jsou nenasycené uhlovodíky – mezi atomy uhlíku se vyskytuje dvojná vazba (C=C). Vazba dvojná je v porovnání s vazbou jednoduchou kratší ale pevnější. Fyzikální vlastnosti jsou podobné alkanům, mohou však mít všechna tři skupenství, jsou bezbarvé. Většinou bez aroma. S rostoucím počtem uhlíků stoupá teplota varu a bod tání. Nejvýznamnějším zástupcem je ethen.

Alkadieny

Molekula dienů obsahuje dvě dvojné vazby. Příkladem může být 1,3 butadien.

Alkyny

Alkyny jsou nenasycené uhlovodíky. Mezi atomy uhlíku se vyskytuje trojná vazba. Nejznámějším zástupcem je ethyn (acetylen), ten je plyn, ostatní jsou kapaliny nebo pevné látky. Jsou velmi reaktivní.

Areny

Jsou to uhlovodíky, které obsahují nejméně jeden aromatický cyklus, tzv. benzenové jádro. Například benzen, naftalen, toluen. Jsou to většinou kapaliny nebo pevné látky. Kapaliny jsou často těkavé, mají charakteristické aroma. Pevné látky většinou sublimují

- **Deriváty uhlovodíků**

Deriváty uhlovodíků jsou organické sloučeniny odvozené od uhlovodíků náhradou jednoho nebo více vodíkových atomů v jejich molekule nějakou charakteristickou skupinou. Tato skupina může být i vícevazná, takže je na ní potom navázáno více uhlovodíkových zbytků. Příkladem jsou halogenderiváty (atom vodíku nahrazen halogenem), kyslíkaté deriváty (alkoholy, aldehydy, ketony, ethery, karboxylové kyseliny…), dusíkaté deriváty (nitrosloučeniny, aminy…), sirné deriváty. Deriváty vyskytující se v rostlinných a živočišných organismech nazýváme složitější organické sloučeniny. Mezi ně patří sacharidy, tuky, bílkoviny, enzymy,vitaminy, hormony,…

Tyto poslední dvě podkapitoly (uhlovodíky a deriváty uhlovodíků) patří mezi velmi rozsáhlá témata, nám však slouží pouze jako přehled pro doplnění představy o rozličných sloučeninách uhlíku. Popis jejich vlastností a aplikací by vystačil na dvě takovéto práce.

Celá kapitola významných sloučenin by nám měla přispět k tomu, abychom si všichni dokázali udělat obrázek o tom, kde všude můžeme uhlík najít.

Opět se nám potvrzují věty vyřčené v úvodní části mé práce a sice, že uhlík je prvkem mnoha tváří a vlastností. Zdaleka však ještě nekončíme. Touto kapitolou jsme

překlenuli již známá fakta o uhlíku a nyní se budeme již zabývat novými a revolučními vlastnostmi uhlíku.

1.5. Nanotechnologie

Samotné slovo nanotechnologie se dá pochopit třemi odlišnými způsoby. Každý, kdo alespoň někdy slyšel nějaký příběh ze science fiction, si pod slovem nanotechnologie představí mrňavé robůtky, kterým se říká nanoboti, kteří jsou schopni sami se vyvíjet a reprodukovat. Tento způsob však ještě dlouhou dobu zůstane jen vidinou budoucnosti a námětem nových vědeckofantastických filmů.

Další výskyt slova nanotechnologie můžeme zaznamenat v materiálovém inženýrství. Dochází k přípravě materiálů s novými vlastnostmi, které vznikají změnou molekulární struktury již známých materiálů. Tato technologie již není jen vizí budoucnosti, vznikly už nanotechnologické materiály, které se používají v leteckém a astronautickém průmyslu, ve stavebnictví i lékařství. Vznikly již také první samoopravitelné materiály podobné polymerům.

Poslední význam slova, i když není nejméně důležitý, je molekulární nanotechnologie (zkratka MNT). Tento obor kombinuje poznatky chemie, mechaniky a biotechnologie ke vzniku nových strojů s miniaturními rozměry. Molekulární nanotechnologie vyrábí zvláštním postupem známé sloučeniny a prvky ve velikostí atomů, čímž tyto látky získávají zcela nové vlastnosti (z téměř nereaktivního zlata se v nanovelikostech stává velice reaktivní prvek).

Nejslibnějším prvkem v nanotechnologii je uhlík. Uhlík se na Zemi vyskytuje ve dvou základních modifikacích. První je diamant, ve kterém je uhlík čtyřvazný a vytváří velice pevnou strukturu. Další je tuha (grafit), ve které je uhlík třívazný a vytváří šestiuhlíkovou mřížku, kde působí slabší vazby.

Fullereny

Fullereny jsou látky složené z kulových molekul uhlíku tvořených jakousi dutou klecí z mnoha atomů. Jméno získaly podle amerického architekta a vynálezce Richarda Buckministera Fullera, který se proslavil podobnými stavebními konstrukcemi, avšak nikoliv jejich objevem! V roce 1966 byla teoreticky předpovězena jejich existence Davidem Jonesem. První, šedesátiatomový fulleren byl syntetizován a objeven v roce 1985 britským vědcem Haroldem Krotem, který za tento objev dostal Nobelovu cenu za chemii.

U sférických fullerenových molekul lze například ovlivňovat vlastnosti tím, že je uvnitř uvězněn atom jiného prvku. Tak lze změnit třeba hustotu, tepelnou kapacitu,

teplotu tání a varu. Prvek uzamčený v molekule fullerenu s ním nevytváří žádné chemické vazby, hovoříme tedy stále o molekule složené pouze z atomů uhlíku.

Počet šestiúhelníků pak může být v podstatě libovolný. Skupina fullerenů tedy začíná od nejméně stabilního benjamínka C_{20}. C_{20} vytváří pravidelný dvanáctistěn, jehož stěny jsou pětiúhelníky. Fullerenové molekuly nemusí být nutně zcela kulaté, jako např. C_{70}, který má tvar rugbyového míče.

Zdaleka nejrozšířenějším a také nejstabilnějším fullerenem je C_{60}, který má stejné uspořádání jako švy na fotbalovém míči. Pravidelně se střídají pěti- a šestiúhelníky. Průměr této téměř dokonalé koule je asi 1 nm. Vyskytují se i mnohem větší molekuly, např. C_{240} nebo C_{540}.

Fullereny se v malém množství vyskytují i v přírodě. A to v sazích, v uhlí anebo hojněji jako pozůstatky po úderu blesků do skal. V laboratoři je dnes možné fulleren C_{60} produkovat s padesátiprocentním ziskem. Pro výrobu je používán obloukový výboj mezi uhlíkovými elektrodami. Méně čistý C_{60} (s příměsí C_{70}) však stojí v internetových obchodech s chemikáliemi podstatně více než zlato.

- **Vlastnosti, využití**

Fullereny s trochou fantazie nejen že vypadají jako fotbalový míč, ale mají i některé vlastnosti společné s fotbalovým míčem. Jsou velmi pružné, při stlačení se deformují a při uvolnění se vrátí do původního tvaru, velmi rychle rotují. Tyto molekuly mají velice slibnou budoucnost, mohou se používat ve filtrech, na které se budou vázat škodlivé látky, mohou se použít ve velice výkonných mikroskopech, které umožní vidět povrchy molekul, jako mazadla a dokonce prokazují i supravodivé vlastnosti. Molekuly fullerenů se vyskytují v přírodě, vyskytují se například i v plameni svíčky (jejich spalování tvoří žluté světlo plamene).

Pokud do nich uzavřeme nějakou molekulu, je s obalem sice nerozlučně spjata, ale není na něj ani chemicky vázána, ani nemůže reagovat s vnějším světem. To je např. C_{60} s uvězněnými atomy kovů, např. $La@C_{60}$ (pomocí @ se označuje, že atom je uvnitř fullerenu a netvoří s ním chemickou vazbu) nebo třeba $He@C_{60}$. Do větších fullerenů je možné umístit i víceatomové molekuly. Naskýtá se i možnost takto přenášet do organizmu molekuly léčiv nebo ukládat do fullerenů radioaktivní atomy.

Nejrůznější atomy mohou být navázány na povrch fullerenů. Příkladem může být $C_{60}F_{60}$, se kterým se počítá jako s perspektivním mazadlem, protože tato „ježatá“ molekula připomíná molekulární valivé ložisko.

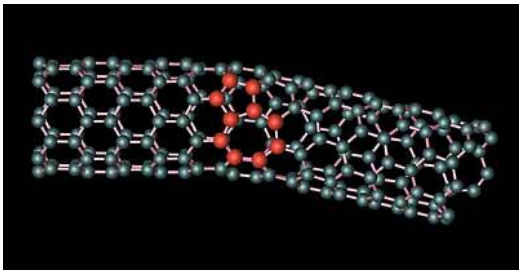
V kombinaci s alkalickými kovy u nich byla pozorována supravodivost. Např. K_3C_{60}, Rb_3C_{60}, Cs_3C_{60} vykazují supravodivost při nízkých teplotách až do ~30 K, to je mnohem vyšší teplota než u ostatních molekulárních supravodičů (~10 K).

Jiné možné využití fullerenů je v ochranných sklech. Při intenzivním osvětlení totiž C_{60} zvyšuje absorpci a tím udržuje množství prošlého světla na nízké hodnotě (tento jev se nazývá „optical limiting“).

Vysokým tlakem je možné přeměnit C_{60} na diamant, a to i při pokojové teplotě. Kvalita takto připravených diamantových vrstev je mnohem lepší než u dosud používaných metod. Možnost vytvářet ochranné vrstvy na různých součástkách může mít velký ekonomický význam.

Nanotrubičky

Další nanomodifikace uhlíku, může tvořit molekuly, které svým tvarem připomínají tubu od zubní pasty (viz Obr. 7). Mohou vzniknout, pokud uhlíkem prochází elektrický proud. Když se přidá trochu kobaltu, niklu nebo železa (katalyzují reakci) vznikají prázdné nanotrubyičky, jejichž povrch je tvořen jedinou vrstvou atomů. Takové vzhledem ke svému průměru několika až několika desítek nm a tvaru byly příhodně pojmenovány jako nanotubes čili nanotrubičky. Tyto nanotrubičky mohou být nesmírně dlouhé, jsou pevné a velice stabilní.



Obr. 7. Část nanotrubičky pod elektronovým mikroskopem

Průměr nanotrubiček může nabývat mnoha hodnot. V jejich průřezu totiž nemusí být celočíselný počet atomů uhlíku. To je dáno tím, že se můžou uhlíky navázat nikoli kruhově, ale v nejrůznější míře spirálně, tedy na uhlík, který je o kroužek dál. Tím se mírně zmenší průměr.

Délka nanotrubiček se běžně uvádí v řádech mikrometrů. V nedávné době byl ale objeven způsob, jak vyrábět nanotrubičky skoro neomezené délky. Konce bývají většinou kulatě uzavřeny polovinou Fullerova míče. Při přípravě fullerenů v elektrickém oblouku mezi uhlíkovými elektrodami byly ještě také nalezeny tenké uhlíkové jehličky o průměru několika nanometrů a délky několika mikrometrů. Po podrobném zkoumání pod

elektronovým mikroskopem vyšlo najevo, že se jehličky skládají z nanotrubiček různého průměru vložených do sebe.

- **Vlastnosti, využití**

Elektrická vodivost

Elektrickou vodivost nanotrubiček lze podstatně zvýšit, jestliže se nanotrubičky vyplní atomy kovů (nebo přechodných kovů). Je možné připravit tzv. nanodrátky (nanowire) – nanotrubičky vyplněné chromem, niklem či gadoliniem. V budoucnu budou mít patrně využití i ve stále se zmenšujících elektronických obvodech.

Nanoteploměr

V americkém Státním ústavu pro výzkum materiálů se podařilo vytvořit sice nepraktický, ale zaručeně nejmenší teploměr na světě. Tvoří ho uhlíková nanotrubička o průměru 75 nanometrů, asi 10 mikrometrů dlouhá a naplněná galliem. Gallium je podobně jako rtuť kapalné již za poměrně nízkých teplot. Výzkumníci zjistili, že v rozmezí teplot přibližně od 50 do 500 °C hladina gallia lineárně stoupá či klesá s teplotou. Nevýhodou teploměru je skutečnost, že k odečtu teploty by byl zapotřebí elektoronový mikroskop

Kvantové síto

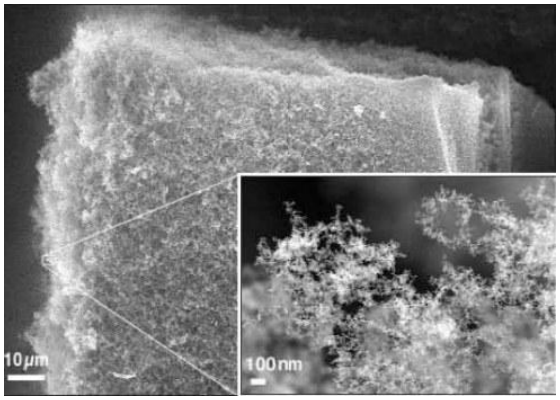
Matrice uhlíkových nanotrubiček by mohla sloužit jako kvantové síto, schopné oddělit lehký izotop vodíku ^{1}H a těžký izotop vodíku – tritium ^{3}H. Chování obou izotopů v nanotrubici se totiž liší. Může za to jejich rozdílná hmota. V podstatě by se dalo říci, že lehčí izotop se (právě pro svoji nižší hmotu) tak silně „vzpírá“, že z nanotrubičky unikne.

Není to tedy tak, že 1H je prostě lehčí a proto projde stěnou, jak by se dalo předpokládat. Jen je u něj větší pravděpodobnost, že se náhle „objeví“ mimo trubičku.

Velká délka

Pomocí postupného růstu v „živném roztoku“ je možné vytvářet nanotrubičky prakticky neomezené délky. Trubičky pak rostou ve svazcích – tvoří tedy vlákna, jak je vidět na snímku z elektronového mikroskopu. Rekord v jejich délce přesáhl 100 m.

Nanopěna



Obr. 8. Snímek uhlíkové nanopěny pod elektronovým mikroskopem

Novou, poslední známou formu uhlíku se podařilo vytvořit týmu fyziků z australské národní laboratoře v Canbeře poté, co vystavili uhlíkový terčík v argonové atmosféře působení výkonného laserového pulsního systému. Mikrostruktura, jež se vytvořila po zahřátí na teplotu 10 000 °C, připomíná jakési vzájemně pospojované sítě uhlíkových trubiček, 5 nm dlouhých (viz Obr. 8). Vědci tuto formu pevné látky nazvali nanopěnou. John Giapintzakis z univerzity na Krétě podrobil nanopěnu pečlivému zkoumání elektronovým mikroskopem, přičemž dospěl k závěru, že novou strukturu můžeme považovat za pátou formu uhlíku.

Na výročním setkání Americké fyzikální společnosti 22. března 2004 v Montrealu Giapintzakis oznámil, že vzorky připravené z uhlíkové nanopěny mohou vykazovat feromagnetické vlastnosti, což je u látky z čistého uhlíku překvapivé. Všechny ostatní známé formy uhlíku se totiž až dosud chovaly jako nemagnetické. Efekt za pokojové teploty vymizí po několika hodinách, ale může se udržet mnohem déle za nízkých teplot. Podle Giapintzakise by bylo možné novou látku využít například v medicíně, když by se po vstříknutí látky do krevního oběhu mohl zviditelnit průtok krve tkání metodou jaderné magnetické rezonance.

David Tomanek z Michiganské státní univerzity, který se zkoumáním uhlíkové pěny také zabývá, zase předpovídá, že by látka mohla posloužit při léčbě rakoviny. Po vstříknutí látky do nádoru by totiž bylo možné jej zničit lokálním zvýšením teploty po pohlcení infračerveného záření nanopěnou, zatímco okolní, zdravá tkáň by zůstala nepoškozena. Vlastnosti nové látky se zatím intenzivně zkoumají v laboratořích, ale lze očekávat, že se nové aplikace objeví brzy.

Shrnutí

Již dnes se nanotechnologie využívá i k výrobě spotřebního zboží, např. tenisové rakety a míčky, nepromokavé lyžařské bundy, vyživující krémy, digitální kamery, sluneční brýle nebo opalovací krémy.

V budoucnosti se počítá s využitím nanotechnologií ve všech oblastech. Lákavé je zejména využití v počítačové technice, kde by nanotechnologie mohly nahradit polovodiče, což by znamenalo podstatně menší rozměry počítačů. Tato technologie se také možná bude používat ke zlepšení komunikace mezi uživateli a přístroji, např. ovládání hlasem nebo holografická podoba dat.

Zajímavou vizí je i použití nanotechnologií v lékařství. Organismus často odmítá transplantované orgány a také klouby. Což způsobuje problém při nahrazování kloubů endoprotézami. Vědci již pracují na úpravě titanových povrchů protéz tak, aby se na nich osteoblasty (buňky odpovědné za tvorbu kostí) dobře uchytily. Na titanový povrch se váží molekuly trojúhelníkového tvaru, které jsou tvořeny guaninem, cytosinem a vybranými látkami. Kombinací molekul trojúhelníkového tvaru s nepokrytými místy vzniká šesterečná struktura, do které se osteoblasty dobře navazují.

Výroba ani výzkum nanotechnologií dosud nejsou v žádném státě nijak omezeny nebo dokonce zakázány. Možná by měly být, protože nikdo dosud pořádně neví, jak se nanočástice chovají v živých organismech. Ví se zatím to, že se nedají zachytit žádným filtrem, snadno procházejí buněčnými stěnami a dokáží zasáhnout i DNA. Jeden z prvních pokusů provedla doktorka Eva Oberdörster z dallaské university, která nasypala malé množství fullerenů do akvária s rybičkami. Po dvou dnech se projevilo poškození mozku, které připomínalo poškození způsobené Alzheimerovou chorobou. Jiné experimenty prokázaly poškození plic při inhalaci vzduchu obsahujícího fullereny a nanotrubičky.

2. Uhlíkové materiály

2.1. Úvod

Uhlík sám o sobě je prvek mnoha tváří a mnoha rozličných vlastností. O této skutečnosti jsem se měl možnost přesvědčit na vlastní kůži při svých exkurzích. Podobné je to i s materiály, které jsou složeny pouze z uhlíku. Prvek uhlík zůstává jedinou složkou uhlíkových materiálů. Velká rozmanitost jeho struktury a textury nám poskytuje širokou paletu vlastností uhlíkových materiálů. Uhlík je z tohoto pohledu velice výjimečným materiálem.

Jedinečné vlastnosti uhlíkových materiálů jsou důvodem jejich nenahraditelnosti, protože lidstvo nezná žádný jiný materiál odolávající teplotám nad 2000°C, aniž by se jeho mechanické vlastnosti nezhoršovaly. V případě uhlíku naopak dochází dokonce ke zlepšení mechanické odolnosti s teplotou 2500°C.

Během posledních 25 let došlo k rozvoji nových uhlíkových materiálů. Jedná se především o uhlíková vlákna , skelný uhlík, pyrolytický uhlík, kompozity uhlík-uhlík, pružný grafit. V tomto pohledu je uhlík nový materiál. Zároveň s těmito objevy byly tradiční materiály, které můžeme obecně shrnout pod název „černá keramika‘‘ , neustále vylepšovány a byly jim dodávány nové vlastnosti a tak je můžeme bez obav také zařadit mezi nové uhlíkové materiály.

Jednoznačně tedy nelze rozdělit uhlíkové materiály na staré a nové. Již Edison používal jako vlákna žárovek uhlíková vlákna pokrytá pyrolytickým uhlíkem. Naprosto zřejmé je to, že uhlíkové materiály se řadí mezi jedny z nejmodernějších materiálů, které lidstvo zná. Svého postavení dosáhly především díky velice unikátním a nenahraditelným vlastnostem. Jsou vyhledávány nejen tradičními průmyslovými odvětvími, ale také špičkovými technickými oblastmi, jako jsou elektronika, oblast jaderného výzkumu, kosmický průmysl a lékařství.

2.2. Rozčlenění uhlíkových materiálů

Jak jsem již naznačil výše uhlíkové materiály se dají rozčlenit především podle vlastností jednotlivých forem a jejich následnou aplikací. Přizpůsobení materiálu požadavkům uživatele závisí na pochopení jednoduchého vztahu mezi třemi spolu souvisejícími stupni: výroba – vlastnosti - použití materiálu.

Ačkoliv vlastnosti uhlíkových materiálů mohou být velice rozdílné, jsou všechny odvozeny pouze od základních vlastností prvku uhlíku.

Materiály, kterými se budeme nadále zabývat si tedy rozdělíme na černou keramiku, neokeramiku, uhlíková vlákna a ohebný grafit.

Toto rozdělení má výhodu v tom, že odpovídá různým způsobům výroby materiálů a zároveň je spojuje do skupin v závislosti na jejich vývoji během historie.

2.3. Získávání uhlíkových materiálů

Uhlíkové materiály

Uhlíkové materiály jsou definovány Francouzskou společností pro studium uhlíku(GFEC) jako libovolná pevná forma uhlíku černé barvy, která ovšem může obsahovat ve stopovém množství další atomy jiných prvků. Tato definice vylučuje ze skupiny uhlí, protože obsahuje podstatně vyšší podíl cizích atomů.

Všechny uhlíkové materiály jsou získávány pyrolýzou organických sloučenin, s výjimkou grafitu vyskytujícího se v přírodním stavu. Pyrolýzou se tedy rozumí termický rozklad organických sloučenin, jako jsou uhlí, ropa, smoly, polymery, uhlovodíky za současné nepřítomnosti dalších chemických činitelů (např.: oxidační činidla). Pyrolýza může pobíhat za postupného zvyšování teploty od 200 do 1000°C a může eventuálně pokračovat až do teploty 1600°C (kalcinace) nebo může probíhat přímo vysokoteplotní cestou. Pyrolýza provedená přímo za vysoké teploty vede k rozptýleným uhlíkovým agregátům – uhlíkovým černím nebo ke kompaktním povlakům – k pyrolytickému uhlíku.

V kondenzovaném stavu (kapalném či pevném) vede pyrolýza k produkci koksů nebo karbonizátů, podle toho, probíhá-li měknutí, či nikoli a prochází-li materiál kapalným stavem.

Dojde-li k měknutí, proces se nazývá koksování a vede ke koksům a dalším uhlíkovým produktům získaným při teplotě 1000°C. V opačném případě je proces nazýván karbonizací, která vede k produkci dřevěného uhlí a aktivního uhlí, eventuálně k produkci skelného uhlíku a uhlíkových vláken.

Poslední stadium pyrolýzy spočívá ve vytěsnění zbytkových heteroatomů, zastoupených nejčastěji vodíkem (ačkoli ve skutečnosti dochází k zachování malého množství heteroatomů při velmi vysokých teplotách). Tato fáze může probíhat jako dokončení koksování, nebo v rámci látky bohaté na uhlík, jako je například antracit. V tom případě je nazývána kalcinací. Rozsah teplot aplikovaných v procesu kalcinace se pohybuje v rozmezí 1200 až 1600°C. Pokud dochází k dalšímu zvyšování teploty nad 2000°C, začíná fáze grafitace, během níž dochází k nárůstu velikosti krystalitů a jejich přeskupení. Tímto způsobem lze připravit polykrystalický grafit a modifikovat vlastnosti dalších uhlíkových materiálů, jako je pyrolytický uhlík nebo uhlíková vlákna. Z výchozích organických látek lze získat:

- nedefinované produkty sloužící jako výchozí suroviny: kalcinovaný antracit, černouhelný koks, černouhelný smolný koks, ropný koks, saze
- kompaktní, dobře definované uhlíkové produkty, které jsou přestaviteli „uhlíkových materiálů" : polykrystalický uhlík a grafit, pyrolytický uhlík, uhlíková vlákna, kompozity uhlík-uhlík, skelný uhlík

Z přírodního grafitu lze získat mimo jiné exfoliovaný grafit.

Zmíněné materiály jsme rozdělili do 4 skupin

- černá keramika
- neokeramika
- uhlíková vlákna
- ohebný grafit

Pyrolýza a grafitace jsou stěženími procesy při výrobě uhlíkových materiálů. Podmiňují strukturu a texturu materiálu, a tím i jeho vlastnosti.

Grafitace

Tento termín označuje termální technologický proces, při němž uhlíkové sloučeniny přecházejí do struktury grafitu.

V aromatických lamelách vytvořených během pyrolýzy jsou již atomy uhlíku uspořádány do hexagonálního, opakujícího se motivu krystalu grafitu. Karbonizací nebo koksováním se tyto lamely přemění na uhlíkové lamely s hexagonálním motivem a stanou se součástí shluků, vedoucích k vytvoření krystalitů. Tyto krystality mají před grafitací délku jen několika nanometrů. Vzdálenost mezi paralelními rovinami je vyšší než v případě monokrystalů (asi 0,345nm oproti 0,335nm) a uspořádání grafenových vrstev kolem svislé osy není dokonalé a připomíná sbírání karet bez jejich dokonalého urovnání. V podstatě jsou paralelní uspořádání a dokonalost rovin omezené a v rámci rovin existují defekty (např.: vakance apod…).

V průběhu grafitace dojde k redukci defektů, nárůstu velikosti krystalitů, snížení mezirovinné vzdálenosti a nárůstu trojrozměrného uspořádání rovin. Tento průběh lze zaznamenat v případě grafitovatelných uhlíkatých sloučenin od teploty 2000°C a je nejvýznamnější v oblasti teplot 2500 – 3000°C. Úroveň této přeměny závisí na dosažené teplotě a především na předchozí orientaci krystalitů, kterou získává uhlíkový materiál od počátku pyrolýzy. Existují tzv. grafitovatelné a negrafitovatelné uhlíkaté sloučeniny.

- **Grafitovatelná uhlíková sloučenina**

Schopnost uhlíkaté sloučeniny přeměnit se na strukturu grafitu je předurčena již v průběhu pyrolýzy. Grafitovatelné uhlíkaté sloučeniny jsou takové, ve kterých mají krystality vytvořené během pyrolýzy tutéž orientaci.

Během grafitace je odstraněna drtivá většina defektů, průměrná velikost krystalitů vzroste a při teplotě 3000°C dosáhne několika desítek až set nm. Mezirovinná vzdálenost klesne až na hodnotu 0,355nm, která odpovídá hodnotě dokonalého krystalu. Fyzikální vlastnosti budou značně změněny: například po grafitaci při teplotě 3000°C elektrická vodivost uhlíkového materiálu získaného pyrolýzou při 1100°C vzroste až na desetinásobek, tepelná vodivost 10-20x. Průběh grafitace lze sledovat pomocí rentgenové difrakce. Tato metoda umožňuje měření nárustu krystalitů a poklesu mezirovinné vzdálenosti.

- **Negrafitovatelný uhlík**

Negrafitovatelné uhlíkové sloučeniny jsou výsledkem karbonizace bez přechodu na kapalnou fázi. U tohoto druhu uhlíkových sloučenin je nežádoucí mezirovinná vzdálenost (0,34 až 0,345 namísto 0,355 nm). Nedokonalé uspořádání lamel s hexagonálním motivem, vzniklé během pyrolýzy, neumožňuje vytvoření delších řetězců a jejich vzájemné propojení. Vytvoří se pouze grafitické články skládající se pouze z několika vrstev. Vznik těchto článků zamezí vzniku grafitové struktury. Tento proces je nezvratný a to i výrazným zvýšením teploty. Tento problém se týká zejména materiálů, jejichž výchozí látky jsou bohaté na kyslík či síru.

2.4. Černá keramika

Několik slov k historii

První výrobky s označením černá keramika byly vyrobeny v polovině 19. stol. Z počátku se hledal materiál především pro výrobu elektrod do elektrických článků, ale do 19. století se vyřezávaly z dřevěného uhlí. Až v roce 1840 připravil Foucault elektrody pro obloukové výboje z retortových sazí (tvrdý uhlíkový nános tvořící se v horní části komínů plynárenských podniků).

K velké revoluci došlo při vynálezu dynama objeveného Gramem v roce 1970. Tento vynález umožnil rychlý rozvoj průmyslových aplikací elektřiny a v první řadě osvětlení pomocí elektrického oblouku. Tento rozmach umožnil vznik nového průmyslového odvětví – výroba obloukových uhlíků.

V roce 1876 byl patentován objev výroby obloukových uhlíků ze směsi mletého kalcinovaného koksu a lampových sazí, spojených cukrem a tvarovaných před pyrolýzou vytlačováním hydraulickým lisem.

Tento princip výroby uhlíkových materiálů objevený v roce 1876 se používá dodnes. Během let s vývojem průmyslu docházelo ke zlepšení technologií a přísad pro výrobu černé keramiky. Stupňující potřeby průmyslu hnaly technologii černé keramiky silně kupředu. K potřebě výroby anod a elektrod, např. pro elektrolytickou výrobu chlóru, přibývá v roce 1882 výroba uhlíkových kartáčů do motorů a dynam.

S objevem grafitace Firardem a Streetem ze společnosti LE CARBON se otevřela uhlíkovým materiálů brána do 20. století. Grafitace, tedy transformace uhlíku v grafit otevírá velmi široké pole působnosti syntetického grafitu.

Ve 20. století byly uhlíkové materiály postupně využívány z počátku ve strojírenství, např. ložiska, zarážky, kulové čepy. Ve 40. letech 20. století se prudce rozvíjí aplikace uhlíkových materiálů v oblasti chemické technologie. V roce 1942 dochází k použití jako nukleárního materiálu v prvním atomovém reaktoru, později i v grafitových reaktorech chlazených plynem, využívaných při výrobě tepla a elektřiny.

Od 50. let 20. století lze uhlíkovou keramiku považovat za moderní materiál, který přispívá k rozvoji nových průmyslových odvětví: elektroeroze, elektronika letectví a astronautika.

Výroba černé keramiky

Postup výroby černé keramiky je jednoduchý a je takřka totožný s obecným způsobem výroby keramiky: příprava zelené hmoty, vytvarování, vytvrzení vypálením. Přičemž výpal může být ještě následován vysokoteplotním přepracováním.

Ve skutečnosti je postup výroby složitější. Pro naše potřeby nám ovšem postačí tento jednoduchý model. Úpravou jednotlivých kroků např. změna teploty nebo výběrem různých surovin pro výrobu zelené hmoty lze docílit požadovaných vlastností pro konkrétní použití.

Příprava výchozí zelené hmoty závisí na použití materiálu, ale také na rozměrech požadovaného výrobku a na technologických možnostech. Aditiva se přidávají jen v malém množství, zlepšují a usnadňují technologický postup výroby. Jsou jimi například mazadla usnadňující tvarování anebo inhibitory rozpínání často pozorovaného během grafitace

Jako plniva se používají ropné a černouhelné koksy, antracit, uhlíkové černě, přírodní a syntetický grafit. Plniva se vybírají dle požadovaných vlastností výrobku. Použitím některého z plniv anebo jejich kombinací můžeme dosáhnout různých vlastností. Plnivo představuje 85% hmoty vyráběného výrobku.

Dalším technologickým krokem je výběr pojidla. Jako pojidla se používají syntetické pryskyřice cukry, melasy a smoly. Běžně užívaná pojidla jsou dehty získané při destilaci černého uhlí. Pojidlo musí snadno smáčet a vést k pevnému propojení zrn.

Dále se směs plniva a pojidla důkladně promísí a vytvaruje do podoby konečného výrobku. Složitou soustavou míchaček, drtičů a tvarovacích lisů postupně dospějeme až k výrobku určeném pro spékání. Tyto produkty si pro pochopení můžeme představit v analogii s klasickou keramikou jako výrobky, které opouštějí hrnčířský kruh.

Spékání spočívá v pyrolýze pojidla za účelem získání vypáleného uhlíkového artefaktu, který takřka neobsahuje vodík. Jde o kritické stádium během něhož může docházet k deformaci výrobků a kdy jsou výrobky vystaveny výrazným tlakům, které mohou způsobit jejich popraskání a rozbití. Pro spékání se používají různé druhy pecí s různými technologiemi ohřevu např. komorové pece s otočnými hořáky nebo pece s pojízdnou nístějí.

Dalším technologickým postupem je primární impregnace. Uvolňování těkavých látek během pyrolýzy pojidla způsobuje vznik porozity. Ta se mění podle sestavy, tvarování, podmínek spékání a samozřejmě v závislosti na podílu pojidla a jeho složení. Porozita může dosáhnout až 30% objemu výrobku. Princip primární impregnace je

jednoduchý. Výrobek je ponořen do organického činidla a následně pyrolyzován. Tím se vytvoří skelet uvnitř pórů a uzavře je. Impregnační činidlo musí mít vysokou výnosnost uhlíku během koksování. Nejčastěji používaná impregnační činidla jsou černouhelné smoly.

Dalším technologickým krokem je grafitace. Aby mohlo dojít k vývoji vlastností spojených s grafitací, je třeba, aby byl materiál určený pro grafitaci pyrolyzován za teploty vyšší než 2500°C, často až 3000°C. Tento proces probíhá v termálních indukčních pecích, které jsou opět založeny na různých principech, které umožňují modifikace vlastností pro konečné využití produktů.

Sekundární impregnace je předposledním technologickým krokem. Charakteristiky uhlíkových materiálů získaných procesy popsanými v předchozích odstavcích mohou být vylepšovány doplňkovými operacemi. Opět se jedná o nanášení různých látek na povrch výrobku.

Impregnovat můžeme:

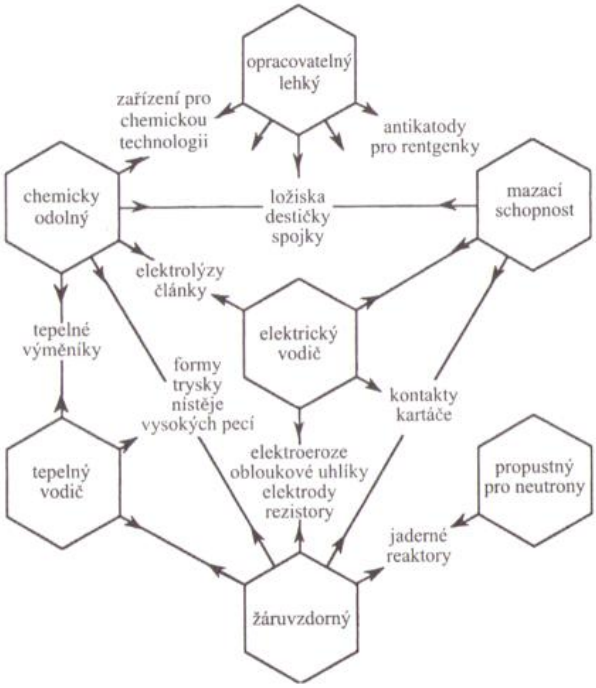
- Pryskyřicemi – výrobky jsou nepropustné pro kapaliny a plyny a mají význam pro chemický průmysl
- Kovy – výrobek je mechanicky odolnější, více vodivý. Nejčastěji se používá stříbro, měď a olovo. Materiály se využívají jako elektrické kontakty, např. sběrače proudu na kolejnicích apod.
- Skly – materiály jsou odolnější proti oxidaci a erozi
- Pyrolytickým uhlíkem – vzniká kompozit typu uhlík-uhlík

Pomocí těchto operací lze značně vylepšit kvalitu materiálu nebo vyrobit materiály nové.

Posledním technologickým krokem je opracování. Výrobky se tvarují podle forem, a proto se jejich povrch musí zabrousit, aby získal přesné rozměry. Pro broušení se používají velmi tvrdé materiály, např. karbid wolframu nebo diamant.

Zvládnutí pracovního postupu výroby černé keramiky umožňuje modifikovat její vlastnosti v širokém rozsahu.

Podle vlastností lze černou keramiku rozdělit do několika skupin. Od těchto vlastností se také odvíjí její aplikace (viz Obr. 9).



Obr. 9. Vztahy mezi vlastnostmi a použitím černé keramiky

Žáruvzdorný materiál

Obecně je žáruvzdorný materiál definován jako materiál s vysokou teplotou tání. Uhlík netaje ani neměkne, tyto vlastnosti ho nesporně zvýhodňují ve srovnání s dalšími běžnými žáruvzdornými materiály na bázi Al_2O_3, křemíku a dokonce i se žáruvzdornými slitinami. Navíc mechanické vlastnosti se nezhoršují s rostoucí teplotou, ale až do teploty 2500°C se zlepšují. Uhlíkové materiály mají oproti žáruvzdorným materiálů tyto výhody:

- velmi dobrá odolnost vůči teplotním šokům
- dobrá tepelná vodivost
- vynikající rozměrová stabilita
- jsou nesmáčivé, skly a struskami
- nezpůsobují kontaminaci kovy
- jsou lehké a snadno se obrábějí

Z těchto tvrzení tedy vyplývá, že uhlíkové materiály jsou nejlepšími žáruvzdornými materiály. Jejich jedinou slabinou, ostatně společnou i s jinými žáruvzdornými materiály, je jejich chováni v oxidační atmosféře. Průmyslový grafit na vzduchu oxiduje při teplotách od 500°C. V případě eliminace některých příměsí působících jako katalyzátory lze oxidaci výrazně omezit.

Použití uhlíkových materiálů související s jejich žáruvzdorností je neobyčejně početné. Aplikaci uhlíkových materiálů si proto rozdělíme do 3 skupin vymezených rozdílnými technicko–ekonomickými nároky.

Použití v oblasti zpracovatelského průmyslu

Použití uhlíku ve zpracovatelských odvětvích jakožto žáruvzdorného materiálu je opět velmi široké. Pro příklad si uvedeme dvě možnosti.

Formy pro:

- odlévání součástek z roztavených žáruvzdorných látek pro stavbu sklářských pecí
- tvarování sklářských forem pro flakony a láhve
- přípravu forem pro tvarování pod tlakem, pod tlakem je možno slévat ocelové ingoty, výhodou slévání do forem sestavených z grafitových desek je rychlé zchlazení a úhledný povrch výrobku
- stlačování za horka v práškové metalurgii, používá se při výrobě součástí z tvrdých karbidů a keramiky, také pro výrobu diamantových součástí, diamantový prášek je smíšen s kovovým pojidlem a stlačován

za velmi vysoké teploty – materiál pro diamantové korunky velmi namáhaných vrtáků

Průvlaky pro kontinuální slévání

Jedná se o průvlečné desky pro výrobu drátů z mědi nebo z oceli. Průvlačné desky musí umožňovat uvolňování tepla při protahování tak, aby mohl kov tryskou vystupovat v pevném stavu.

Speciální aplikace

Ablace je narušení povrchu materiálu plynem za vysoké teploty a při velké rychlosti proudění. Grafity odolávají ablaci velmi dobře a z tohoto důvodu slouží jako povlaky trysek balistických střel. V současnosti vytlačují v oblasti trysek balistických střel černou keramiku kompozity typu uhlík-uhlík. Dochází tedy k nahrazování uhlíkové keramiky neokeramikou. Při procházení odborných periodik jsem zjistil, že tato skutečnost se dotýká i kosmického výzkumu. Neokeramika se používá v podobě kachlí k pokrývání přední části raketoplánů, vytváří tak žáruovzdornou vrstvu při průletu atmosférou.

Žáruvzdorný materiál a vodič

Černá keramika představuje vodiče, jejichž odpor měřený při laboratorní teplotě je výrazně vyšší ve srovnání s kovy. Odpor kovu vzrůstá s rostoucí teplotou a odpor černé keramiky zůstává stálý.

Uhlíkové materiály jako jediné prakticky a ekonomicky splňují požadavky elektrického oblouku a černá keramika nachází díky kombinaci vodivosti a žáruvzdorného charakteru mnohá využití, v nichž je nenahraditelná.

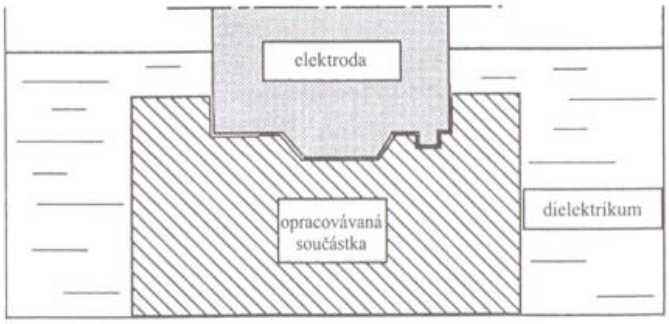
Používají se jako - elektrody do obloukových pecí
- obloukové uhlíky pro osvětlení a pro vypalování drážek
- elektrody pro spektrografii
- odpory pro odporové pece pracující za velmi vysoké teploty
- susceptory pro indukční pece

- **Elektrody pro opracování elektroerozí**

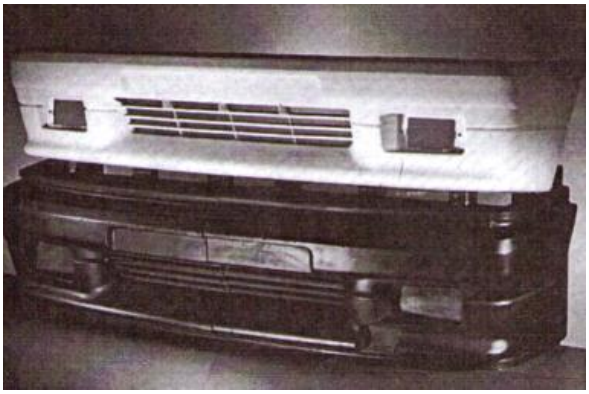
Elektrický oblouk umožňuje přeměnu elektrické energie na světlo nebo teplo. Je tedy zdrojem koncentrovaného tepla vysoké tepelné úrovně.

Opracování elektroerozí spočívá ve vývoji série jisker mezi elektrodou a opracovávanou součástkou. Opracovaná součástka je ponořena do dielektrika a vlivem generátoru výboje je probitím dielektrika vytvořena jiskra (viz. Obr. 10). Tyto výboje

jsou velmi krátké (kratší než 0,001 sekundy) a způsobují mikrotavení nebo mikrovypařování kovu, který tvoří opracovávanou součástku. Opracování elektroerozí bylo vyvinuto před několika desetiletími Lazarenkem v bývalém Sovětském svazu. Umožňuje levně získávat součástky složitých tvarů z velmi tvrdých slitin, jsou to zejména složité součástky pro obrábění (viz. Obr. 11), například: kovací zápustky, formy pro sklářství, formy pro výrobu plastů a tlakové tavení…



Obr. 10. Schéma opracování elektroerozí



Obr. 11. Elektroda pro opracování elektroerozí čelní nárazník auta (bílý) v horní části fotografie vychází z formy opracované pomocí grafitové elektrody

Třecí materiál

Tření je odpor působící proti vzájemnému pohybu dvou povrchů. Jde o známý jev, který je ve skutečnosti velmi složitý.

Na úrovni krystalové mřížky zůstávají oba povrchy hrubé i po jejich značném vyleštění. Takovéto povrchy se dotýkají pouze vrcholky svých mikroskopických nerovností. Tyto vrcholky jsou nejdříve plasticky deformovány později dochází na kontaktních bodech k tavení, zakalení a tvorbě slitin. Všechny tyto jevy jsou však nežádoucí.

Pro eliminaci těchto problémů se používají různé technologie. Je to například vytvoření tenké vrstvy modifikující povahu povrchu. Tyto vrstvy jsou většinou tvořeny oxidy, to je případ ocelí, litiny a bronzu. Pokud tato vrstva zmizí například ve vakuu (podmínky, existující v oblasti kosmického výzkumu nebo při jaderných aplikacích atd) dojde k zadření.

Krystalická struktura grafitu umožňuje rovinám, které ji vytvářejí, vzájemně po sobě klouzat (ve směru stěpnosti) a vyžadují jen slabé síly pro překonání van der Waalsových sil, které způsobují vzájemnou vazbu těchto rovin. Takové slabé síly mohou být ještě oslabeny skutečností, že grafit má schopnost tvořit interkalační sloučeniny, které oslabují van der Waalsovy síly. Tření mohou například napomoci vodní pára, stříbro nebo některé halogenidy.

Grafit dobře přilne k povrchům, se kterými je v kontaktu. Vzhledem ke snadné štěpnosti vytváří povrchovou vrstvu, která je tvořena uhlíkem a oxidy kovů a vodou, ke kterým grafit přiléhá. Grafit je tedy samomazná pevná látka a představuje vynikající třecí materiál. Těchto vlastností se nejvíce využívá při výrobě kartáčků pro elektrická zařízení.

- **Kartáče pro otočná zařízení**

Ve všech otočných elektrických zařízeních, motorech, generátorech, musí být proud veden od otáčivých vodičů do elektrického obvodu nebo naopak, kontakt je zajištěn malými součástkami, které se třou o tyto otočné



Obr. 12. Různé typy kartáčů

vodiče. Tyto součástky se nazývají kartáče (viz Obr. 12). Kartáče z uhlíkové hmoty mají hned několik výhod: dobrá vodivost, samomazné schopnosti umožňující tření bez poškození kovové plochy…

Uhlíkové kartáče jsou součástí systému: držák kartáče/kartáč/kolektor a třecí dráha vytvořená na kolektoru označovaná jako patina (složena ze složité směsi uhlíku, oxidů kovů a vody).

Materiál odolávající korozi

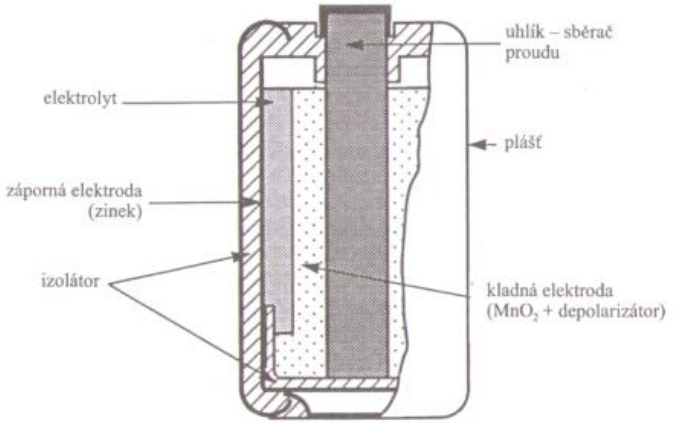
Uhlíkové materiály jsou za běžné teploty téměř inertní (s výjimkou silných velmi oxidačních kyselin). I za zvýšené teploty odolávají rozpouštědlům, chloru, fluoru, roztaveným solím, a zejména roztaveným fluoridům a chloridům.

Tyto dobré vlastnosti spolu s mechanickými a teplotními vlastnostmi způsobují, že uhlíkové materiály jsou zcela mimořádnými pro použití v chemickém průmyslu a v kombinaci s elektrickou vodivostí jsou pozoruhodným materiálem pro elektrochemii.

V současné době se syntetický grafit využívá v chemickém průmyslu pro tepelné výměníky, jednotky pro syntézu kyseliny chlorovodíkové, čerpadla, kolony, výparníky. V elektrochemii pro výrobu hliníku, hořčíku, sodíku, lithia a fluoru termoelektrolýzou, dále pro ochranu katod a elektrické palivové články.

- **Elektrické články**

Použití uhlíkových materiálů, coby elektrod v elektrických článcích, bylo vůbec prvním použitím těchto článků (viz Obr. 13). Od té doby došlo k významnému rozvoji jejich použití. Souvisí to se spotřebou elektrických uhlíků pro výrobu baterií s oxidem manganičitým.



Objem výroby těchto uhlíků dosáhl v 60. letech několika desítek tisíc tun, ale od té doby je na poklesu vzhledem k významnému úspěchu alkalických baterií, které

neužívají uhlíky, nýbrž rozptýlené formy uhlíku.

Obr. 13. Suchý článek s MnO_2

Materiál pro jadernou energetiku

Vzhledem k nízké atomové hmotnosti je uhlík stejně jako vodík, deuterium, nebo berylium dobrý moderátor neutronů. Při srážce neutronu s atomy lehkého prvku mu neutrony předají energii, ačkoli by při srážce s těžkým atomem byly odraženy bez výrazného zpomalení.

Uhlík má navíc nízký účinný průřez pro záchyt tepelných neutronů. To znamená, že pravděpodobnost zachycení neutronu prolétajícího v blízkosti atomu uhlíku je velmi nízká. Hovorově tedy můžeme říci, že uhlík je propustný pro neutrony. Zmíněné vlastnosti předurčují syntetický grafit pro použití v jaderné energetice. Tyto vlastnosti umožnily jeho použití jako moderátoru v prvním jaderném reaktoru postaveném během druhé světové války v Chicagu.

Další velkou výhodou je, že grafit jako strukturní materiál není přeměněn na radioaktivní odpad. Podle mého názoru se zajisté dočkáme ještě spousty dalších vlastností, které uhlík předurčí pro využití v jaderné energetice.

Uhlíkový materiál je většinou používán pro soubor jeho vlastností a nikoliv pro jeho jedinou specifickou vlastnost. Souhrn vlastností uhlíkových materiálů, umožňuje pochopit jejich všeobecné použití v jaderné oblasti:

- jako moderátor (zpomaluje při jaderných reakcích neutrony), jádra a opláštění v grafitových reaktorech chlazených plynem
- jako reflektoru v reaktorech s těžkou vodou
- pro neutronovou ochranu reaktorů s rychlými neutrony, chlazených sodíkem
- jako ochranné součástky první stěny kontejneru plazmatu v reaktorech jaderné fůze
- jako kontejnery pro skladování radioaktivních odpadů

2.5. Neokeramika

Termín neokeramika označuje nové materiály, které se snaží vyhovět velmi přísným podmínkám jejich použití, vyžadují vysoké nároky na termomechanickou odolnost, dokonalou rozměrovou stabilitu a odolnost v agresivních prostředích.

Rozdíl mezi neokeramikou a černou keramikou vychází z odlišného způsobu jejich výroby. Materiály spadající pod název černá keramika, jsou získávány pyrolýzou plynných uhlovodíků.

Tyto materiály mají značný význam v oblasti kosmického průmyslu.

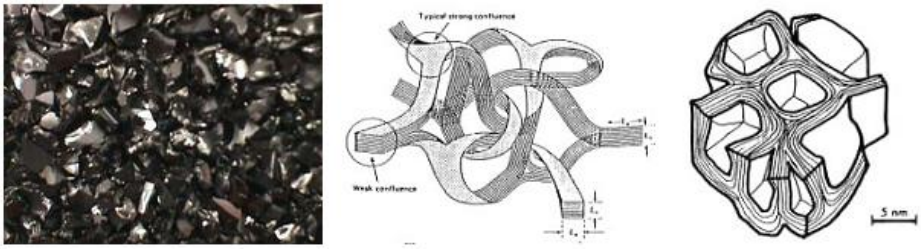
Skelný uhlík

Skelný uhlík získal své jméno podle svého charakteristického lasturnatého lomu a lesklého vzhledu, který je velmi podobný sklu. Je nepropustný vůči kapalinám a plynům. Těmito vlastnostmi však analogie se sklem končí.

Skelný uhlík se získává karbonizací polymerů, jako je celulóza nebo teplotně tvrditelné pryskyřice (viz Obr. 14). K pyrolýze dochází v pevném stavu, a tak jsou vytvářené grafénové vrstvy neuspořádané a nemohou narůstat. Skelný uhlík tedy není grafitovatelný a jeho vlastnosti se během zahřívání na vysokou teplotu mění jen nepatrně. Jednotlivé vrstvy se neuspořádávají do paralelních rovin o více něž dvou nebo třech vrstvách, sklený uhlík je tak tvrdý a izotropní. Tloušťka současných výrobků ze skelného uhlíku může dosahovat řádově jen několik milimetrů.

Opracování skelného uhlíku je složité, protože je velmi tvrdý, proto je definitivní tvar výrobků vytvořen před jejich karbonizací.

Skelný uhlík se využívá například na kelímky pro pěstování monokrystalů, báze harddisků, formy pro optické čočky, reflektory pro velmi výkonné lampy, protézy vnitřního ucha, srdeční chlopně.



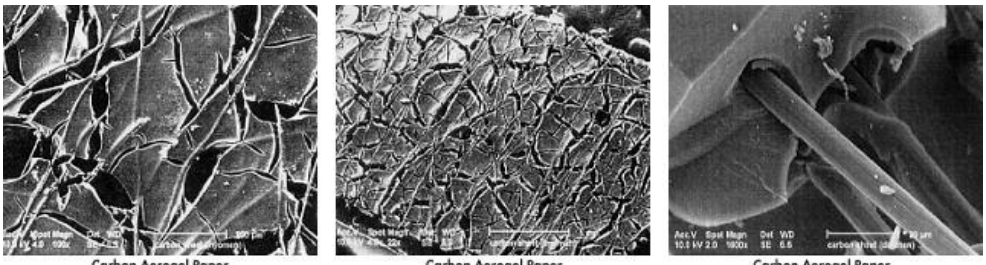
Obr. 14. Skelný uhlík – snímek a modely jeho struktury

Pyrolytický uhlík

Pyrolytický uhlík (pyrolytic carbon, pyrocarbon, pyrolytic graphite) je monolitický materiál získaný chemickým rozkladem těkavých uhlovodíkových sloučenin na podložce o rozsahu teplot 727 - 1827 °C. Pyrolytický grafit má vysoký stupeň přednostní orientace C-os kolmých k povrchu podložky a je připravován nad teplotou 1827 °C. Je prakticky nepropustný pro plyny, tepelná a elektrická vodivost závisí na stupni orientace vrstev a je vyšší ve směru grafitových rovin. Průmyslové využití je v oblasti rezistorů, povlaků pro jaderná paliva, v raketové technice jako vystýlka trysek, povlaky grafitových lodiček, kelímků, kyvet a elektrod v analytické chemii.

Uhlíkové aerogely

Uhlíkové aerogely (carbon aerogels) jsou vyráběny karbonizací aerogelů připravených z formaldehydových pryskyřic (viz Obr. 15). Vyznačují se nízkou měrnou hmotností, supernízkou tepelnou vodivostí, vysokým měrným povrchem a zajímavými elektrickými vlastnostmi. Využití nacházejí nejen jako tepelné isolace, ale především v dobíjecích bateriích a palivových článcích a jako nosiče katalyzátorů.



Obr. 15. Uhlíkový aerogel ve formě papíru

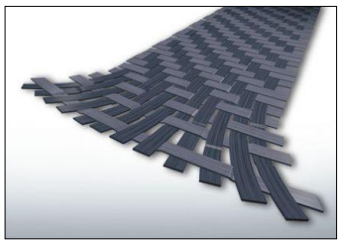
2.6. Uhlíková vlákna

Uhlíková vlákna nemohou být vyráběna stejným způsobem jako vlákna kovová, skelná, křemenná nebo vlákna polymerů, vzhledem k tomu, že uhlík taje a je tažný až za velmi vysokých teplot a je dokonale odolný vůči rozpouštědlům.

Jako ostatní uhlíkové materiály jsou i uhlíková vlákna výsledkem kontrolované pyrolýzy organických prekurzorů. Prekurzor sám o sobě musí být ve formě vlákna. Karbonizace, je-li prováděna na netajícím prekurzoru, uchovává jeho tvar. To umožňuje získávat nejen vlákna, ale z vláken také látky nebo uhlíkové plsti nebo textilie. Tkaniny, plsti a izotropní vlákna, získaná tímto způsobem mají jen průměrné mechanické vlastnosti, avšak nacházejí již řadu použití. Přesto se v běžné mluvě označení uhlíková vlákna vztahuje na vlákna s vysokými mechanickými parametry, jejich odolnost překračuje odolnost nejlepších ocelí a jejich výhodou je čtyřikrát nižší hustota. Tato vlákna jsou užívána pro výrobu kompozitních materiálů, široce používaných v oblasti leteckého průmyslu a průmyslu pro kosmický výzkum. Dále pro výrobu některých sportovních potřeb (hokejky, tenisové rakety) nebo součástek pro různá průmyslová odvětví. Každý svazek je tvořen z 1 000 až 10 000 vláken o průměru 5 – 12 mikrometrů, jsou tedy mnohem tenčí než lidský vlas.

Izotropní vlákna

Tato vlákna mají průměrné mechanické vlastnosti. Jsou získávána tkaním smoly (bez předchozí specifické úpravy) v proudu teplého vzduchu. Takto prováděna oxidace způsobuje jejich netavitelnost, což umožňuje jejich karbonizaci, aniž by došlo k deformaci nebo spojení mezi vlákny.



Obr. 16. Příklad tkaniny utkané z uhlíkových pásek

Za podobných podmínek lze získat tkaniny nebo plsti (viz Obr. 16), v nichž vlákna zůstanou izotropní.

Jak jsem se již zmínil izotropní vlákna mají průměrné mechanické vlastnosti ve srovnání s vlákny s vysokými parametry. Nelze je tedy použít pro výrobu výjimečných kompozitních materiálů, lze je naproti tomu užít jako posílení cementových nebo betonových součástí, pro stavbu domů a mostů.

Vlákna s vysokými parametry

Vlákna s vysokými parametry jsou v podstatě tuhá a odolná vlákna, tedy s vysokým modulem pružnosti a vysokou odolností v tahu. Jsou charakteristická svým vysokým stupněm anizotropie. Na obrázku je porovnána pevnost a tuhost uhlíkových vláken a některých jiných materiálů. Hodnota 100 byla přiřazena uhlíkovým vláknům a další materiály byly umístěny s relativní hodnotou (přičemž platí, že hodnota 100 odpovídá 5 000MPa pro pevnost a 500GPa, pokud jde o modul pružnosti uhlíkových vláken, ačkoli mohou být získána uhlíková vlákna s odolností proti přetržení až 7000MPa nebo hodnotou 800GPa pro modul pružnosti).

V odborných zdrojích jsem se dočetl, že pro tyto výborné vlastnosti jsou tato vlákna využívána v aeronautice, ve vesmíru a všude tam, kde jsou požadovány jak vysoká tuhost, tak i značná lehkost.

Kompozity vlákno-pryskyřice

Výjimečná kombinace vlastností nabízená uhlíkovými vlákny, která jim zaručuje první místo při použití jako zesilující struktury v kompozitech o vysokých užitných vlastnostech, umožnila neobyčejný rozvoj těchto materiálů, a to od 70. let minulého století až do současnosti.

Odolnost vůči přetržení, tuhost, odolnost vůči únavě a korozi jsou u kompozitů několikanásobně větší než u oceli a lehkých slitin, přičemž jejich specifická hmotnost je dvakrát až pětkrát nižší. Jejich koeficient roztažnosti je mimo jiné zanedbatelný a nelze ho srovnávat s koeficienty roztažnosti kovů.

Nejdůležitější pryskyřice pro výrobu jsou epoxidové pryskyřice, které mají výbornou adhezi k uhlíkovým vláknům. Jsou charakterizovány slabým smrštěním během tvrzení a výbornou odolností vůči vnějším vlivům. Pro využití za vyšších teplot (nad 200°C) jsou používány polyimidové pryskyřice.

Tyto kompozity nacházejí uplatnění v leteckém a kosmickém průmyslu. Nemalou měrou se uplatňují i v dalších průmyslových odvětvích, včetně výroby potřeb pro sport a zábavu, nemluvě při tom o využití v lékařství.

Kompozity s matricí sklo, kov nebo keramika

Křehkost monolitové struktury nebo skel omezuje jejich použití při realizaci velkorozměrových struktur nebo objektů, které musejí odolávat otřesům a zvýšenému napětí. Jejich vyztužení uhlíkovými vlákny vede k uskutečnění dávných požadavků: využití jejich výborné odolnosti vůči vysokým teplotám a korozi a zároveň zajistit

odolnost při šíření prasklin, což vylučuje tvorbu puklin s katastrofickými následky. Také pevnost kovů a jejich tuhost mohou být vylepšeny jejich vyztužením uhlíkovými vlákny.

Výroba těchto substrátů může být prováděna infiltrací substrátu uhlíkových vláken v kapalné nebo plynné fázi. Pokud to bod tání matrice umožňuje, tato součást může být vstřikována do předem připravené formy s vlákny. Pokud hodnota tání omezuje aplikaci v kapalném stavu, postupuje se při výrobě chemickou kondenzací z plynné fáze nebo slinutím.

Kompozity uhlík-uhlík

„Kdo dokáže více, dokáže i méně.“ Toto úsloví vysvětluje, že kompozity uhlík-uhlík našly dosud mnohá použití, zatímco kompozity s kovovou nebo keramickou matricí zatím na své využití čekají. Žádný jiný materiál než uhlík nedokáže udržet své mechanické vlastnosti nad teplotu vyšší jak 1 500°C, zatímco kompozity uhlík-uhlík je dále zlepšují až do teploty 2500°C.

Kompozity uhlík-uhlík kombinují výhody 100% uhlíkového materiálu s výjimečným odporem a pevností, kterou mu dodává mikrostruktura vláken. Kompozit uhlík-uhlík složený z uhlíkové matrice a z výztuže z uhlíkových vláken je tedy charakterizován tím, že je složen pouze z jediného prvku a sice z uhlíku.

Výroba těchto kompozitů spočívá ve včlenění uhlíku do prostoru mezi vlákny substrátu tak, že získáme materiál s požadovanou hustotou, aniž bychom vyvolali nadměrné vnitřní napětí a poškodili tak vlákna. Tohoto můžeme docílit například impregnací kapalnými pryskyřicemi, nebo smolami s následným vypálením.

Pro výrobu kompozitů uhlík-uhlík můžeme volit celou řadu technologií, které nám umožňují poměrně jednoduše modifikovat vlastnosti výsledného kompozitu. Kompozity uhlík-uhlík kombinují lehkost s překvapivými termomechanickými vlastnostmi (vysoká pevnost, tuhost, houževnatost, slabý součinitel teplotní roztažnosti, dobrá tepelná vodivost, dobrá odolnost vůči teplotním šokům). Vedle lehkosti a termomechanických vlastností mají tyto kompozity také celou škálu vlastností jako jiné uhlíkové materiály. Jsou zejména biokompatibilní a mají výhodné třecí a otěrové vlastnosti.

Avšak i pro tyto uhlíkové materiály platí, že jejich slabinou je odolnost za vysokých teplot v oxidační atmosféře. Tento nedostatek se dá omezit nebo zcela odstranit vytvářením povlaků, nebo lépe impregnací, jejichž výsledkem je vyplnění porozity:

- povlaky vzácných kovů, získaných chemickou depozicí z plynné fáze

- křemíkováním, spočívá v transformaci určité povrchové vrstvy v karbid křemíku
- impregnací sloučeninami bóru, fosforu nebo tetraetlortosilikáty, což vede k vyplnění porozity a mikroskopických trhlinek oxidem křemičitým

- **Termomechanické vlastnosti**

Žáruvzdornost a velmi vysoká hodnota skupenského vypařovacího tepla dělají z kompozitů uhlík-uhlík nejlepší známý materiál odolný vůči ablaci a umožňují jejich použití jak v prostředí horkých plynů, tak v oblasti návratu nadzvukových letadel do atmosféry.

Souhrn těchto vlastností dělá z kompozitu uhlík-uhlík nejvhodnější materiál pro průmyslové využití v oblasti aeronautiky. Tyto materiály se používají pro nosy a náběžné hrany křídel raketoplánů a pro součásti raket a balistických střel.

- **Biokompatibilita**

Kompozity uhlík-uhlík jsou vynikajícím biomateriálem, vzhledem k tomu, že uhlík je součástí molekulární stavby lidského těla. Jsou odolnější než kosti a jejich výhodou je ve srovnání s kovy to, že nenarušují růst kostní tkáně mladého člověka. V kardiovaskulární chirurgii je jejich biokompatibilita ještě důležitější a nezpůsobuje žádnou negativní komplexní interakci s krví, jako je tomu v případě použití kovů. Tuto skutečnost potvrzuje i praktický příklad v závěru práce.

Chování uhlíkového materiálu přesahuje samotnou chemickou kompatibilitu. Skutečně se spojuje s tkáněmi. V kostních protézách a v náhradách vazů je postupně jeho porozita vyplněna kostní nebo vazovou tkání. Použití uhlíkových materiálů v lékařství je tedy významné a pravděpodobně se bude dále rozšiřovat.

V současné době se používají jako:

- kostní protézy – rameno, koleno, kyčel, vnitřní ucho, zubní implantáty
- svalové protézy – srdeční chlopně, umělá srdce

- **Třecí a otěrové vlastnosti**

Tyto vlastnosti jsou podobné jako v případě černé keramiky. Samozřejmě i využití je velmi obdobné jen s tím rozdílem, že kompozity uhlík-uhlík mají daleko lepší mechanické vlastnosti než černá keramika. Používají se zejména jako brzdové disky a čelisti brzd pro vojenská, osobní letadla a vysokorychlostní vlaky. Účinnost takovýchto

brzdných systémů je čtyřikrát vyšší ve srovnání s kovovými disky a navíc umožňuje značné odlehčení.

2.7. Ohebný grafit

Jak jsem si již uvedl, grafit je charakterizován existencí velmi silných vazeb v grafenových vrstvách, kde je opakován hexagonální motiv benzenu a slabými silami mezi těmito rovinami. Toto uspořádání vysvětluje možnost vstupu chemické sloučeniny mezi tyto dvě grafenové vrstvy, aniž by došlo k narušení lamelární struktury. Činidlo způsobí pouze rozestoupení rovin a vytvoří něco, co se podobá plátu upečeného listového těsta. Vstup činidla mezi grafénové vrstvy je reakcí interkalárního typu.

Interkalační sloučeniny mohou být získávány z grafitu působením kapalin, par nebo elektrolýzou. Tyto sloučeniny objevil v polovině 19. století pan Brodie. Tyto sloučeniny po zahřátí výrazně zvětší svůj objem. Jedná se zejména o sloučeniny s alkáliemi, s chloridy, a s fluoridy kovů. S výrobou těchto uhlíkových materiálů jsem měl možnost se seznámit ve společnosti Graphite Týn nad Vltavou. Jedná se o grafit, který nese mezi jednotlivými grafénovými vrstvami atom jiného kovu, například sodíku nebo lithia. Tyto grafity mají tu vlastnost, že pokud se zahřejí na teplotu vyšší než 80°C, roztáhnou se jako harmonika. Této specifické vlastnosti se využívá především v protipožární ochraně.

Interkalační sloučeniny lze rozdělit do dvou skupin. V první vznikají silné kovalentní vazby, ve druhé jsou vazby mezi grafénovými rovinami slabě ovlivněny.

Sloučeniny se silnými vazbami

Jde o sloučeniny vznikající interkalací kyslíku nebo fluoru. Vzniklé silné vazby mobilizují volné elektrony grafitu a získané sloučeniny jsou tedy elektrickými izolátory. Sloučeniny vzniklé interkalací fluoru mají mazací vlastnosti přechodné mezi grafitem a teflonem a jsou tedy používány jako mazadla. Používají se také pro výrobu miniaturních elektrických článků s vysokou energetickou kapacitou.

Oxid grafitu je pevná látka. Její barva se mění v závislosti na poměru uhlíku a kyslíku. Oxid grafitu je připravován v kyselém silně oxidačním prostředí z rozetřeného přírodního grafitu.

Oxid grafitu je charakterizován schopností rozptylovat se ve vodě a za vzniku thixotropních gelů. Tyto viskózní gely umožňují získávání membrán polopropustných pro vodní páru. Lze z nich také připravovat povlaky grafitovatelné za nízkých teplot.

Sloučeniny se slabými vazbami

Jedná se o nejpočetnější skupinu interkalačních sloučenin. V těchto sloučeninách se střídají grafénové vrstvy s vrstvami interkalovaných molekul. Vzdálenost mezi

grafénovými vrstvami vzrůstá v závislosti na povaze použitého interkalárního činidla. Interkalace způsobí nárůst nosičů elektrického náboje, což dále vede ke zvýšení elektrické vodivosti. Ta může vzrůst 10-100x ve srovnání s původním grafitem.

Některé interkalační sloučeniny mají elektrickou vodivost blízkou vodivosti mědi. Grafit, který je obyčejně charakterizován jako polokov, se stává skutečným kovem. Provedením interkalace gravitovaných uhlíkových vláken, jejichž mechanická odolnost je nám dobře známa, by bylo možné získat lehké a odolné vodiče. Lze si představit realizaci vedení vysokého napětí se snížením počtu sloupů nebo mnohem lépe manipulovatelných podmořských kabelů. Interkalační sloučeniny jsou poměrně nestabilní a musely by být chráněny před vnějším prostředím.

Exfoliovaný grafit

Oxidy grafitu se při rychlém zahřívání rozkládají formou objemných sazí. Získaný černý materiál obsahuje ještě značný podíl kyslíku. Z tohoto materiálu můžeme připravit komplexy, jejichž rozklad vede k čistému stlačitelnému grafitu. Tyto komplexy jsou získávány interkalací přírodního grafitu pomocí činidel jako je například kyselina sírová. Získaný komplex je následně promýván a sušen za nízké teploty.

Rychlým zahřátím dojde k exfoliaci komplexu se značným nárůstem objemu. Průmyslově je interkalovaný grafit vháněn do plamene hořáku (za takových podmínek jsou interkalované molekuly eliminovány v plynném stavu. Rychlé zahřátí vede k vytvoření „černého sněhu‘‘, který se zvolna ukládá v dekantační komoře. Tato látka má velmi nízkou specifickou hmotnost. Tento proces představuje první krok při výrobě ohebného grafitu.

Komplexy, vyráběné za podobných podmínek, expandují za teplot 200 až 250°C s rychlostí, která je tím vyšší, čím vyšší je teplota. Vytvářejí kompaktní, nereaktivní pěnu, která umožňuje hašení hořících kovů. Zejména sodíku, hořčíku, hliníku, uranu, titanu, pro něž je použití vody z pochopitelných důvodů nemožné, podobně jako použití ostatních sloučenin v běžných hasících přístrojích.

Vytvořená pěna je lehká a umožňuje izolaci povrchu kovu od okolní atmosféry s komplexní hmotností desetkrát nižší ve srovnání s ostatními látkami užívanými pro hašení hořících kovů (karbonáty a chloridy alkálií)

Exfoliovaný grafit má mimo jiné výhodu v tom, že neznečišťuje kov a v možnosti jeho zpětného získání jednoduchým odstraněním grafitu s povrchovou vrstvou.

Ohebný grafit

Přibližně před 30-ti lety bylo objeveno, že jemné vločky, jimiž je tvořen exfoliovaný grafit, jsou stlačitelné a umožňují získat list čistého grafitu.

V praxi je prováděno stlačení nebo válcování grafitu. Tato mechanická technologie prováděná například za nízkého tlaku uspořádá jemné vločky exfoliovaného grafitu kolmo k působící síle a vede ke vzniku silně anizotropních výrobků, které jsou distribuovány ve formě destiček nebo listů. Ty mohou být svinovány do rolí a pásek (viz. Obr. 17).

Ohebný grafit má obecné vlastnosti uhlíkových materiálů. Navíc je jeho výhodou nepropustnost i přes jeho nízkou specifickou hmotnost (1,1 g/cm^3) a velmi snadné použití.



Obr. 17. Role ohebného grafitu

- **Ideální materiál pro těsnění**

Vzhledem k tomu, že ohebný grafit nestárne a neochabuje a vzhledem k chemické odolnosti a mazacím vlastnostem ho lze výhodně použít jako těsnící materiál v korozivním prostředí, v rotujících systémech (čerpadla, ventily, míchačky…). Může být užíván za extrémních teplot, a to jak velmi vysokých, tak velmi nízkých, také v potravinářství a v rozvodech pitné vody. Velmi dlouhá životnost a vysoká těsnost vzniklých výstelek jsou argumenty pro jeho značné využití v jaderných elektrárnách. V rozřezané formě je používán jako spojka přírub a pro těsnění válců motorů a sběračů výfuků.

Poslední jmenované využití grafitu se dále rozvíjí. Při výzkumu, jak zlepšit aerodynamické tvary automobilu, jsou konstruktéři nuceni ukládat motory šikmo a naležato. Výfuk však vyžaduje ohyb a v důsledku toho krátký kulový čep. Azbest používaný pro zajištění těsnění tohoto čepu (nehledě na způsobené nebezpečné znečištění) se zničí po několika desítkách tisíc kilometrů. Ohebný grafit vyztužený kovovou textilií vydrží po celou dobu životnost vozu a je zcela neškodný.

- **Žáruvzdorný a nepropustný materiál**

Ohebný grafit je žáruvzdorný a nepropustný, navíc odráží záření, přijatelně vede elektřinu a je snadno tvarovatelný. Soubor těchto vlastností vysvětluje jeho široké použití

při konstrukci pecí, při konstrukci jednoduchých nebo složených clon, pro ochranu izolací a uzavírání speciálních atmosfér.

Není smáčen skly ani kovy a může být tedy užíván pro ochranu forem a pro manipulaci s horkými výrobky ve formách. Jeho naprostou výhodou je to, že na rozdíl od azbestu není škodlivý.

Některé pevné kovové chloridy reagují s plynným amoniakem za vzniku komplexů. Reakce je exotermní a reverzibilní. Rozklad komplexu je tedy endotermní a vyvíjí chlad. Tento princip je používán při realizaci chemických tepelných čerpadel. Avšak výměna tepla v objemu chloridů je slabá, jakož i kontakt mezi pevnou látkou a plynem.

Vše je jinak pokud jsou bloky vyrobeny ze slabě stlačeného exfoliovaného grafitu, který může být napuštěn chloridy. Takové bloky umožňují velmi dobrý přenos tepla ve směru kolmém k ose stlačení a porozita umožňuje velmi dobrou difůzi reaktivního plynu. Na tomto principu byla zkonstruována tepelná čerpadla malých rozměrů. Počítá se nejprve s jejich využitím při vyvíjení chladu nebo v klimatizačních systémech, ale uvažuje se i o dalších možnostech.

3. Strukturální, chemické a biologické vlastnosti uhlíkových vrstev na povrchu polymerů

Tato část by měla být ukázkou konkrétního využití uhlíku v praxi.

Využití uhlíku, jak z celé mé práce vyplývá, je značné. Když jsem hledal oblast, o které bych se chtěl zmínit podrobněji, zaujala mne asi nejvíce oblast medicíny, kde se jedná o bezprostřední přínos pro člověka.

Zjistil jsem si, že velmi zajímavým výzkumem se kromě jiných zabývá např. i VŠCHT Praha, Ústav inženýrství pevných látek, konkrétně prof. ing. Václav Švorčík, DrSc. Vědecký výzkum tohoto ústavu je velmi široký. Nejvíce mne však zaujal výzkum biokompatibility polymerů. Ústav inženýrství pevných látek se tímto výzkumem zabývá již několik let.

Biokompatibilita polymerů je obor zabývající se snášenlivostí polymerů biologickou hmotou (lidské buňky, rostliny). Velmi mne potěšilo, že poté, co jsem prof. ing. Václava Švorčíka, DrSc oslovil a požádal o možnost konzultací, případně možnost seznámit se s výzkumem, vyšel mi ochotně vstříc a do celé problematiky jsem byl zasvěcen. O jaký převratný objev se jedná?

Budeme se zabývat vlastnostmi uhlíkových vrstev na povrchu polymerů.

Polymery, ve kterých je přítomen dostatek uhlíkatých struktur, jsou příznivě vnímány živými buňkami. Uhlík má tu výhodu, že není toxický a je tedy vhodný pro použití v medicíně.

Jedním z nejpoužívanějších polymerů v medicíně je polyethylen (PE). Jeho využití je velmi široké, avšak spíše jako pomocný materiál – nádoby apod. Polyethylen je také vhodný k modifikaci jak plazmou (zahřátí povrchu PE – dojde k vytvoření uhlíkových struktur na povrchu materiálu), tak pro nanášení uhlíku jinými metodami (napařování, aj.). Takto upravený polyethylen se používá jako substrát pro kožní implantáty. Dalšími často užívaným polymery v medicíně jsou například teflon (protézy sluchového ústrojí a mimo jiné i jako kapalina snižující tření v kloubech), nebo jiné polymery, PET (pletené cévní náhrady), PA (srdeční chlopně) , polykarbonáty (náhrady tvrdých tkání).

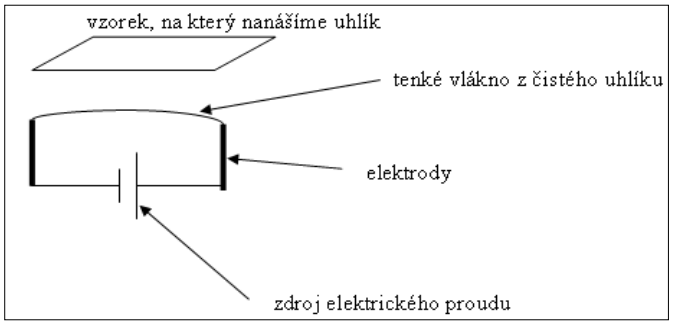
Pro nanášení uhlíku na povrch polymeru se používají různé technologie. Vrstvy uhlíku na polymeru můžeme docílit například zahříváním polymeru bez přístupu vzduchu. Touto technologií se také připravuje čistý uhlík

Dalšími metodami jsou napařování a naprašování. Tyto metody patří mezi PVD metody (“ physical vapour deposition ”).

Naprašování je proces, při kterém je povrch odprašovaného materiálu bombardován urychlenými ionty inertního plynu a tím rozprašován do evakuovaného prostoru (5.10^{-2} – 5.10^{-5} Pa). Inertním plynem obvykle bývá argon, který vytváří při ionizaci kladné ionty. Na terč je přivedeno záporné napětí 500 - 5000 V. Při srážkách se projevuje také elasticita srážek dopadajících částic a atomů. Atomy jsou naprašovány na daný substrát, přičemž dochází zároveň ke změnám chemických vazeb na substrátu a vzniku vazeb nových.

Napařování je specifickou metodou nanášení uhlíkové vrstvy na povrch polymeru. Tato reakce probíhá ve vakuu. Následující jednoduché schéma (viz Obr. 18) nám znázorňuje průběh napařování.

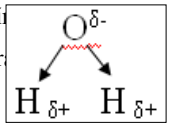
Při napařování umístíme tenkou uhlíkovou nit mezi kladnou a zápornou elektrodu a připojíme silný zdroj elektrického proudu. Proud, který probíhá uhlíkovým vláknem jej rozžhaví a uhlíkové vlákno praskne, uhlík se odpaří a naváže se na vzorek polymeru. Množství nanášeného uhlíku se dá ovlivnit tloušťkou uhlíkové niti a vzdáleností vlákno – polymer. Vše probíhá opět ve vakuu. V tomto případě se jedná o metodu flash, nikoliv obecně o napařování.



Obr. 18. Schéma napařování

Při napařování dochází k zahřátí uhlíku a k jeho odpařování. Při tomto procesu vznikají volné radikály. Jedná se o velmi agresivní částice, které mají velmi silnou snahu přitáhnout k sobě jiný prvek a vytvořit s ním silnou vazbu. V tomto případě to bývá nejčastěji kyslík. Právě uhlíkové struktury s přítomností kyslíku mají významný vliv na kultivaci buněk.

Buňky také mají velmi rády, „něco mezi‘‘ tzv. polárním a nepolární Jako příklad polární kapaliny si můžeme uvést vodu (viz Obr. 19), kter

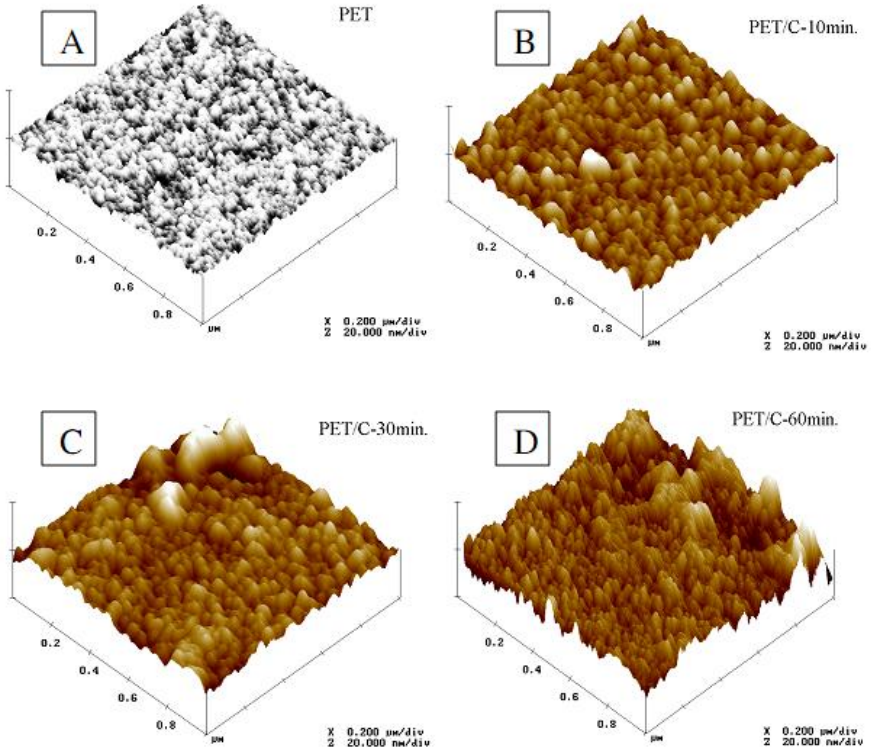


Obr. 19. Polarita vody

parciálního kladného a záporného náboje. Polární vazbu charakterizuje dipól**.** Velikost dipólu je závislá na součinu délky vazby a parciálního náboje.

Pokud kápneme kapku vody na nějaký povrch a tento povrch bude polární, kapka se rovnoměrně rozlije, pokud polární nebude, kapka zůstane celistvá a nerozlije se. Jako příklad si můžeme uvést suchou tkaninu. Pokud na ni nakápneme vodu, bude okamžik trvat, než se polární vazby natočí směrem ven ke, v ten moment se kapka vsákne. Podobně to funguje i u buněk. Pokud naneseme na povrch polymeru potaženého uhlíkem buňky, mnohem lépe přilnou, když bude povrch polární. Popisované reakce jsou ve skutečnosti mnohem složitější, záleží například i na velikosti náboje jednotlivých atomů vrstvy, morfologii vrstvy, druhu buněk apod., pro nás ale nemají větší význam.

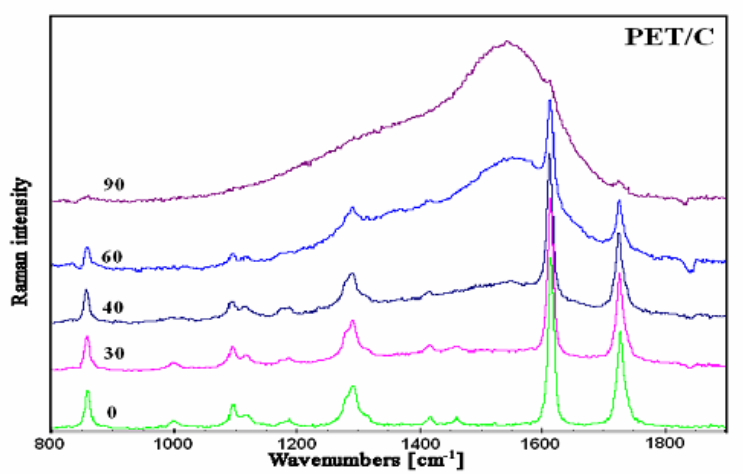
Názornou ukázku toho, jak může vypadat povrch pokrytý vrstvou uhlíku, nám poskytuje Obr. č. 20. Jeho dílčí snímky A-D se vztahují k jiné technologii nanášení uhlíku na povrch polymeru a sice k chemical vapour deposition. Na snímku A vidíme původní povrch polymeru, zatímco na snímku D je povrch zcela pokryt uhlíkem. Tyto snímky byly pořízeny mikroskopem atomárních sil. U metody napařování, kterou se zabýváme, by povrch vypadal téměř stejně, proto nám tyto snímky mohou sloužit pro



Obr. 20. Specifický povrch polymeru během nanášení uhlíku

přibližnou představu.

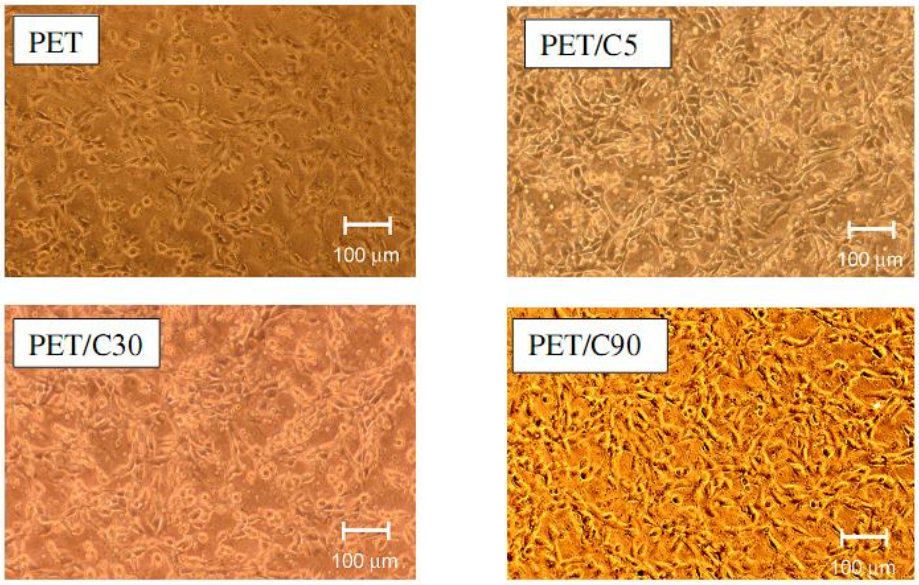
Na Obr. č. 21 vidíme tzv. Ramanovu spektroskopii, která nám slouží pro získání podrobnějších informací o struktuře zkoumaného povrchu. Ramanova spektroskopie nám ukazuje přírůstek množství uhlíku za určitý čas. Zelená křivka s označením 0 představuje základní polymer, v tomto případě je to PET. Další křivky nám ukazují postupný nárůst množství uhlíku na povrchu polymeru. Jednotlivé píky na křivkách představují atomy uhlíku, které jsou součástí základního řetězce polymeru. U poslední křivky s označením 90 vidíme, že uhlík zcela překryl původní polymer. Tento graf se opět vztahuje k CVD metodě a slouží nám tedy jako ilustrace. Spektroskopie napařeného povrchu by vypadala obdobně.



Obr. 21. Ramanova spektroskopie

Nanášení tenkých uhlíkových vrstev má velký význam v medicíně. Jak jsem se již zmiňoval, tak polyethylen pokrytý vrstvou uhlíkatých struktur je velmi dobře přijímán kožními buňkami. Pro tuto jeho vlastnost se používá ke kultivaci různých typů buněk. Dříve se pro tyto účely používala textilní vlákna, ale buňky se nemnožily příliš rychle a v potřebném množství, proto vědci zkoumali, jaký povrch by buňkám vyhovoval více než textilní vlákna. Jejich zrak se upřel na polymery pokryté tenkými vrstvami uhlíku. Na obr. č. 22 vidíme povrch polymeru pokrytý nakultivovanými buňkami. Popis v levém horním rohu opět znázorňuje dobu nanášení uhlíku. Opět se jednalo o CVD metodu, obrázky pro napařování by vypadaly obdobně.

Buňky jsou mimo jiné citlivé na množství napařeného uhlíku. Pokud je množství napařeného uhlíku příliš velké, buňky se nekultivují s takovou úspěšností. Je to dáno zejména velkou změnou morfologie povrchu. Povrch je velmi členitý a buňky k němu obtížně přilnou (viz. obr. 3 D). Musíme tedy zvolit ideální množství napařeného uhlíku. U CVD metody to lze ovlivnit dobou depozice a u námi popisovaného napařování tloušťkou uhlíkové niti a velikostí proudu a vzdáleností vlákno-polymer. Tento parametr uhlíkových vrstev je předmětem intenzivního zkoumání, protože každému druhu buněk



vyhovují jiné parametry.

Obr. 22. Růst buněk na povrchu polymeru

Jako praktický příklad využití polymerů s vrstvou uhlíku můžeme uvést kultivaci kožních buněk. Lékaři jsou schopni na takto připraveném povrchu vypěstovat kožní štěp, který se poté aplikuje na tkáň poškozenou například popálením, obr. č. 23, 24 a 25. Na obrázcích č. 23., 24 jsou uvedeny textilní kožní štěpy (substráty).



Obr. 23. Ilustrační obrázek kožního štěpu



Obr. 24. Přiložení autokultivátu



Obr. 25. Rozsáhlý trofický defekt zahojený autotransplantací kůže

Tato jedinečná vlastnost uhlíku nám znovu dokazuje, že uhlík je prvkem mnoha tváří a mnoha rozličných vlastností. Uhlík nám poskytuje stále nové a nové možnosti svého využití. Je to téměř neuvěřitelné, ale jako prvek je uhlík známý snad od pradávna, avšak vědci stále objevují nové modifikace a využití. Podle mého názoru se s ním v budoucnu budeme setkávat mnohem častěji a to ve fascinujících technologiích třetího tisíciletí.

Závěr

Pro svoji ročníkovou práci jsem si vybral moderní uhlíkové technologie. Tato práce mne obohatila jednoznačně všemi směry. Při své práci jsem se seznámil nejen s teorií o uhlíku a uhlíkových materiálů, ale chtěl jsem se seznámit také bezprostředně s praxí. V rámci své práce se mi naskytla možnost navštívit různé typy společností zabývající se činnostmi blízkými mé ročníkové práci.

Jako první jsem navštívil grafitový důl u Českého Krumlova. Díky vstřícnému přístupu průvodce jsem se dozvěděl spoustu užitečných informací o vzniku grafitové rudy a o historii jihočeských grafitů. Zde se mi podařilo získat několik vzorků čistého amorfního grafitu.

Při dalších návštěvách Jižních Čech jsem navštívil společnost Graphite Týn nad Vltavou, která se zabývá zpracováním surového grafitu a přípravou polotovarů pro další zpracování.

Ve společnosti jsem se seznámil jednak s výrobou polotovarů pro všechny sektory průmyslu, ale také s finálním využitím těchto polotovarů. Zde mne nejvíce zaujala výroba takzvaných expandovatelných grafitů.

Nemalým přínosem pro mne byla spolupráce s Vysokou školou chemicko-technologickou v Praze. Naše spolupráce se týkala především poslední části mé práce a sice praktického příkladu využití. Tato zkušenost pro mne byla opět významným přínosem, neboť jsem se seznámil s prací ve špičkově vybavených laboratořích.

Díky těmto exkurzím jsem poznal mnoho kvalitních oborníků, kteří mi poskytli odpovědi na veškeré mé otázky a nejasnosti. Veškeré zkušenosti a poznatky se výrazně promítly do mé práce. Díky těmto aspektům se má práce obohatila a doufám, že se stane přínosem nejen pro mne, ale také pro ty, kteří touží proniknout do tajů zdánlivě známých a všedních skutečností o uhlíku. Každý je toho názoru, že uhlík je dokonale známý a že již nemá co nabídnout, ale opak je pravdou a má práce je toho důkazem.

Práce mi pomohla také při volbě směru mého dalšího studia. Této problematice bych se rád věnoval i při mém dalším studiu na vysoké škole a dále jej rozvíjel a tím si připravil budou-cí základ jednou své diplomové práce.

Uhlíkové materiály zažívají v současné době dynamický rozvoj. Podle mého názoru má uhlík a uhlíkové materiály velkou budoucnost. Uhlík stál na prahu již několika klíčových vývojových etap (průmyslová revoluce) lidstva a nepochybuji o tom, že se v této pozici ocitne ještě několikrát.

Poděkování

Děkuji za pomoc při řešení problémů spjatých s moji ročníkovou prací.

V první řadě mému školnímu konzultantovi Mgr. Haně Gajduškové, která mi pomáhala při řešení problémů týkajících se obsahové formy a logického uspořádání mé práce.

Můj další významný dík patří Vysoké škole chemicko-technologické v Praze konkrétně Ústavu inženýrství pevných látek, jmenovitě panu Prof. Ing. Václavu Švorčíkovi, DrSc, za jeho významnou pomoc při zpracovávání praktického příkladu využití uhlíku. Jeho rady se pro mne a mou práci staly opravdu velmi cenným přínosem.

Na úplný závěr bych chtěl poděkovat za spolupráci společnosti Graphite Týn, spol. s r. o., za poskytnutí cenných materiálů a rad týkajících se formy mé práce.

Seznam použitých zkratek

Obr.	Obrázek
HPHT	High pressure high temperature
CVD	Chemical vapor deposition
Tzv.	Tak zvaný
Např.	Například
MNT	Molecular nanotechnology
Viz.	Videlicet – lze vidět
Apod.	A podobně
Č.	Číslo
PE	Polyethylen
Aj.	A jiné

Použitá literatura

1. http://cs.wikibooks.org/wiki/Fullereny_a_nanotrubičky
2. http://www.aldebaran.cz/bulletin/2004_25_uhl.html
3. http://www.petrasek.info/fyzika-pevnych-latek/diamanty-uz-nejsou-nejtvrdsi.html
4. http://www.scienceworld.cz
5. http://www.veda.cz/detail.jsp?articleId=9351
6. http://www.scienceworld.cz/sw.nsf/0/3C0818C7706B9B93C1256E970048FF0A?OpenDocument&cast=1
7. http://www.zvedavec.org/pocitace_840.htm
8. http://www.seminarky.cz
9. http://www.maturita.cz
10. http://cs.wikipedia.org/wiki/Uhl%C3%ADk
11. http://www.navajo.cz
12. Uhlíkové materiály, André Legendre, nakladatelství Informatorium rv. 2001
13. 21. století, září 2003
14. Propagační materiály společnosti Graphite Týn, Týn nad Vltavou
15. Jednosvazková encyklopedie Universum A-Z, r. v. 2004
16. http://www.jergym.hiedu.cz/~canovm/orbital/vesmir/vesmir.html



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

VYUŽITÍ CAD V TECHNICKÉ PRAXI

Josef Chmelař
Střední Průmyslová Škola - Nové Město nad Metují

Anotace

- práce je zaměřená na výuku 3D modelování v programu SolidWorks
- Jsou zde představeny systémy Autodesk Inventor a SolidWorks
- Popisuje tvorbu 3D modelu pásového bagru v systému SolidWorks
- Ukazuje možnost využití 3D dat při tvorbě prototypů, které jsou součástí práce
- Popisuje využití redakčního systému PmWiki při tvorbě výukové pomůcky

Prohlášení

Prohlašuji tímto, že jsem ročníkovou práci vypracoval samostatně pod vedením Ing. Josefa Nymše a uvedl v seznamu informačních zdrojů veškerou použitou literaturu a další informační zdroje včetně internetu.

Obsah

Úvod .....

1. Výhody 3D CAD modelování v praxi a historie .....

1.1. Vývoj a historie CAD systémů .....

1.2. Výhody konstruování na PC ve 3D .....

2. 3D systémy Autodesk Inventor 11 a SolidWorks 2007 .....

2.1. Autodesk Inventor Professional 11 .....

2.2. SolidWorks Office Premium 2007 .....

3. Modelování v programu SolidWorks .....

3.1. Projekt: Bagr .....

3.2. Návaznost 3D modelu .....

4. Výuková pomůcka – online: info.spsnome.cz .....

Závěr .....

Poděkování .....

Použité informační zdroje .....

-

Úvod

Téma využití CAD v technické praxi jsem si vybral na základě mého zájmu o tuto problematiku. Mám již nějaké zkušenosti s modelovacími programy Autodesk Inventor a 3ds Max, které bych chtěl zajisté v této práci využít a rozšířit.

V první části práce stručně vysvětlím vývoj grafických 3D systémů a také se budu věnovat výhodám modelovacích programů ve výrobním procesu výrobku oproti klasickým výkresům 2D. Nastíním jejich možnosti vizualizace, simulace nejrůznějších stavů výrobku apod. Pak se zaměřím na dva z těchto programů a to Autodesk Inventor 11 a SolidWorks 2007 budu se snažit udělat stručný přehled hlavních funkcí u každého z těchto programů doplněný názornými obrázky. Chtěl bych v této kapitole také okrajově obsáhnout hardwarové požadavky těchto programů, jako jsou vhodné typy procesorů či specializované grafické karty. V hlavní části této práce se zaměřím na program SolidWorks 2007, kde je mým konečným cílem vytvoření výukových prezentací pro školní portál www.info.spsnome.cz. Pokusím se také navázat kontakt s dealerem systému SolidWorks Top-Tech s.r.o. pro odbornou pomoc a řešení otázek.

Během zpracovávání této práce budu čerpat informace z odborných časopisů, které mám k dispozici, pokusím se sehnat si nějakou další literaturu spojenou s tímto tématem.

Jsem velmi rád, že mohu zpracovávat takovouto práci, protože si myslím, že tím nabudu spoustu zkušeností, které se mi budou hodit nejen v dalším studiu, ale i v životě či v zaměstnání.

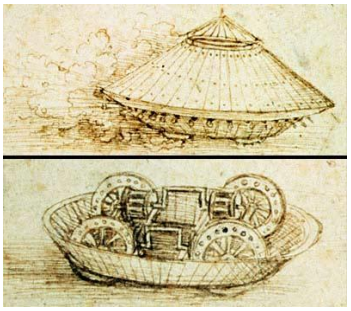
-

1. Výhody 3D CAD modelování v praxi a historie

1.1. Vývoj a historie CAD systémů



Lidé již od nepaměti cítili nutnost zaznamenávat svoje myšlenky a nápady někam, kde by si to mohli přečíst zpětně nejen oni, ale i další lidé. V pravěku si lidé zaznamenávali svoje nápady úhlem na stěny jeskyní, staří Egypťané používali k zaznamenávání dláto a hliněné destičky, které byli záhy vyměněny za poněkud praktičtější papyrus či zadělanou kůži.



Ve středověku a renesanci se můžeme setkat s důmyslnými návrhy nejrůznějších strojů či opevnění jak můžeme vidět třeba v díle Leonarda da Vinciho, které byli již zaznamenáváni na papír.

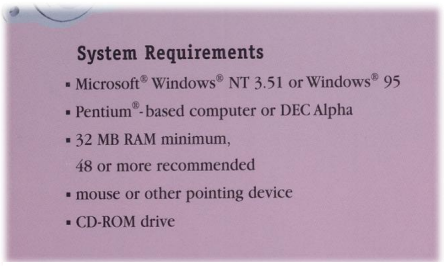
Další velkou změnu přinesl nástup elektroniky v 60. letech 20. století. S postupným rozvojem počítačů se začali i někteří lidé zajímat o to, jak by šel využít potenciál počítače k navrhování v průmyslu. Prvním takovým průkopníkem byl Dr. Patrick J. Hanratty, který je přezdíván i jako otec CAD (Computer Aided Design) aplikací, vytvořil první komerční NC programovací systém Pronto a jako zaměstnanec automobilky General Motors stál u vývoje prvního interaktivního grafického systému pro strojírenství DAC (Design Automated by Computer).

Vzhledem k tomu, že tato práce je zaměřená na produkty firmy Autodesk a Dassault Systems, měl bych také uvést, jak je to s historií těchto dvou programů.

Šestnáctičlenná skupina počítačových fandů, pod vedením Johna Walkera založila firmu Autodesk. Rozhodli vyvinout komerční CAD, který by cenou nepřesáhl tisíc dolarů a byl široce použitelný v nejrůznějších odvětvích průmyslu. To se jim také povedlo a v roce 1982 představili na veletrhu Comdex první komerční CAD pro osobní počítače-AutoCAD. Tento software se stal za čtvrt století své existence jedním z nejznámějších CAD programů. V Dnešní době firma Autodesk disponuje širokou škálou software řešení pro počítačově podporované navrhování, a to ať už jde o strojírenství, stavebnictví, design, infrastruktury apod.

-

Co se týká konkrétně programu SolidWorks, jeho historie sahá do roku 1995, kdy byl na trh uveden první nativní 3D CAD systém pro Windows a byl od samého počátku koncipován jako 3D modelář. Program byl navržen pro co nejjednodušší pochopení a intuitivní práci. SolidWorks je patentován mimo jiné také za pomůcku Feature Manager, což je v podstatě historie modelování, kde si můžu jakýkoli krok, který jsem v minulosti na modelu udělal, zpětně editovat. Tuto součást využívá většina moderních CAD systémů. V roce 1997 odkoupila firmu SolidWorks firma Dassault Systems, která stojí mimo jiné za vývojem programu CATIA. O úspěchu SolidWorks již svědčí za více než desetiletou éru života více než 700.000 spokojených uživatelů.



trocha nostalgie na obrázku: minimální požadavky první verze SolidWorks 95

1.2. Výhody konstruování na PC ve 3D

Hlavní výhodou konstruhování v moderních programech na PC je jistě jejich rychlost, přesnost a provázanost s dalšími etapami vývoje navrhovaného výrobku.

V minulosti, když měl konstruktér navrhnout nějaký stroj, zabralo zbytečně spoustu času kreslení celé konstrukce na papír. Musel se stále spoléhat na svoji přesnost, a když někde v procesu konstruhování udělal chybu, většinou se na ni přišlo až při výrobě, a to firma již ztrácela nemalé prostředky na jejím odstranění a také čas vývoje.

Dnes si již konstruktér za pomocí počítače dokáže nejen že celý stroj navrhnout, ale může si ho i vyzkoušet, jak mu bude fungovat a zda se v jeho návrhu nevyskytly nějaké chyby. Zapojení počítačů do procesu vývoje a prototypu přineslo velké zrychlení celé výroby. Konstruktér zkonstruhuje výrobek na PC ve 3D a ihned z něho může udělat 2D dokumentaci nutnou na dílně k výrobě a sestavování součástí. Další nedílnou výhodou je vytváření prezentací výrobků na PC. V dnešní době již také velmi záleží na designu výroku, tak efektní fotorealistická vizualizace může velmi pomoci marketingu a obchodníkům prezentovat výrobek dříve, než je opravdu vyroben.

Nedávno jsem diskutoval na prezentaci programu SolidWorks 2008 s konstruktéry jedné firmy z Pardubic. Ptal jsem se jich na to, jaké CAD systémy používají při konstruhování. Dostal jsem odpověď, že používají 2D řešení AutoCAD, ale že tam mají také na několika stanicích 3D CAD SolidWorks, který moc nevyužívají, protože v něm nikdo v

-

podstatě neumí efektivně pracovat. Docela mě tento fakt zarazil, protože z vlastní zkušenosti vím, jaký je rozdíl v produktivitě práce pokud pracujete na 2D CADu nebo 3D CADu.

Navrhování ve 3D, neboli trojrozměrném prostoru, vám nabízí úplně nové možnosti pohledu na výrobek, některé jsem již zmiňoval výše, další budou jistě zmíněny v průběhu mé práce. Jednak vidíte kompletně model tohoto výrobku, se kterým si můžeme otáčet, také vytváření celé výkresové 2D dokumentace je tam opravdu snadné. Většina dnešních 3D CADů umí dokonce generovat výkresy včetně popisků automaticky. Stačí jen zadat co na výkrese požaduji a z které strany. Mezi další velmi vyhledávané funkce patří dnes pevnostní analýza modelů, nebo knihovna standardů – šroubů, ložisek, matic, profilů…

-

2. 3D systémy Autodesk Inventor 11 a SolidWorks 2007

V této kapitole bych chtěl představit 3D systémy pro podporu konstruování, a to Autodesk Inventor Professional 11 a SolidWorks Premium 2007. Na počátku bych chtěl říci, že není mým cílem tyto dva systémy přímo srovnávat, protože v dnešní době si myslím, že jsou oba na velmi vysoké úrovni a kolem každého je utvořena komunita lidí, kterým vyhovuje k práci právě ten jejich oblíbený program. S nástupem nové produktové řady „2008“ od společnosti Autodesk dostane i Inventor nativní podporu 64bitového rozhraní procesorů, takže bude moci využívat mnohem více paměti RAM. Toto napomůže konstruktérům v navrhování mnohem komplexnějších sestav, než to bylo možné doposud a tím v podstatě dožene jeden z posledních nedostatků oproti programu SolidWorks.

2.1. Autodesk Inventor Professional 11

Ceník - orientační ceny produktu u dealera Xanadu a.s.

Autodesk Inventor Suite 2008 CZ – 170.435 Kč

Autodesk Inventor Professional 2008 CZ – 255.625 Kč

Obecné základní informace

Produktová řada Autodesk Inventor je navazujícím článkem v evoluci vývoje konstrukčních systémů AutoCAD. Autodesk Inventor 11 je dodáván na jednom DVD společně Mechanical Desktop 2007 a Autodesk Vault, který slouží ke správě dat a knihoven normalizovaných částí do programů. Při koupi Inventoru se můžeme rozhodnout mezi dvěma verzemi a to Autodesk Inventor Professional a Autodesk Inventor Series. První jmenovaný je základní verze Inventoru – Series obsahující navíc ještě oborově zaměřené moduly (Import elektrotechnických schémat, Rozvody drátů a kabeláží, Potrubí a potrubní systémy, Analýza metodou konečných prvků a dynamická analýza). Měl bych také dodat, že verze Professional je dražší než její základní verze Series.

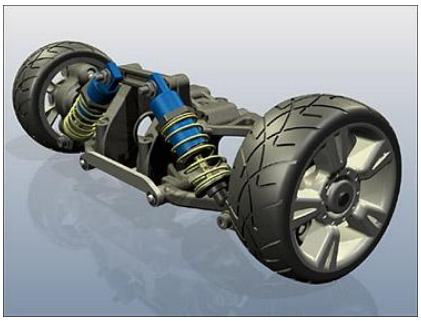
Potřebuji funkce:	Inventor Suite (Series)	Inventor Professional for Routed Systems	Inventor Professional for Simulation	Inventor Professional
Autodesk Inventor	✓	✓	✓	✓
AutoCAD Mechanical	✓	✓	✓	✓
Návrh kabelů a kabelových svazků, IDF	-	✓	-	✓
Návrh rour, potrubí a hadic	-	✓	-	✓
Analýza zatížení (MKP)	-	-	✓	✓
Dynamické simulace	-	-	✓	✓
Autodesk Vault (PLM)	✓	✓	✓	✓

-

Inventor je navržen tak, aby umožnil co nejlehčí přechod uživatelů z 2D CAD řešení AutoCAD Mechanical na Inventor. Programová filozofie je taková, že se pracuje se sestavami a dílci, kde spojením více dílců vznikne sestava, ve které můžeme provádět např. dynamické simulace, kde si ověřujeme, jak se bude výrobek chovat v sestavě apod. Od verze 10 popř. 11 se v Inventoru objevují i nejrůznější moduly pro usnadnění rozšíření práce návrháře:

Inventor studio

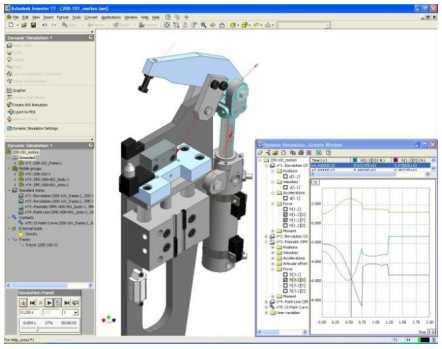
Tento modul je určen k rychlým a intuitivním vizualizacím vytvořených výrobků. Z mých osobních zkušeností je to výhoda, protože konstruktér pak nemusí svoje návrhy složitě převádět a exportovat do přenosných formátů a pak je importovat do speciálních vizualizačních programů jako je Artlatnis (www.artlantis.com) nebo 3DS Max (www.autodesk.com/3dsmax). Je pravda, že tyto programy nabízejí mnohem více možností nastavení a vizualizace z nich mají rozhodně vyšší kvalitu, ale Inventor studio je právě navrženo pro jednoduché a intuitivní ovládání, bez potřeby dalšího exportu dat nebo složitého nastavování. Po přepnutí do modulu Inventor studia se okamžitě uživateli nabídne řada nástrojů pro animaci. Uživatel si nadefinuje rovinu, která tvoří jakoby zem, poté si nadefinuje kameru, osvětlení, které tvoří 3 základní světla: směrové světlo, bodové světlo a reflektor. U každého lze pomocí jednoduchých sliderů měnit nejrůznější parametry jako jas, barvu apod. Animace komponent je tu také velice intuitivní, stačí si vybrat jednu z možností, co chceme vlastně animovat: zda komponenty, vazby, útlum či polohovou reprezentaci, kdy nám zůstávají tzv. ghosty (animovaná komponenta zůstává v určitých polohách průhledná, aby byla vidět trasa, po které se pohybovala). Co se týká materiálových stylů, je tu v knihovně předdefinováno několik desítek materiálů, včetně textur, odlesků a průhlednosti (pokud jde o skla), ale není žádný problém pro uživatele si je upravit dle svých představ a uložit do knihovny pro další použití v jiných projektech.

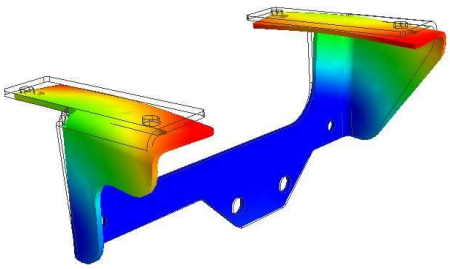


-

Kinematická a dynamická simulace

Tento modul umožňuje návrhářům simulovat a analyzovat chování sestav z hlediska funkčnosti a výsledky ihned využít např. pro metodu konečných prvků. Práce s tímto modulem je velmi jednoduchá, přehledná a uživatelsky příjemná. Přesto vyžaduje po návrháři alespoň základní znalosti problematiky simulace. Do modulu dynamická simulace se dá z prostředí modelování přepnout stejně jako u modulu Inventor Studio klepnutím v nabídce Aplikace. Po vstoupení do tohoto modulu si uživatel nejprve musí nadefinovat pohybové vazby. K dispozici je 30 definovaných spojení komponentů. Poté je potřeba nadefinovat síly a momenty a nesmíme opomenout aktivování gravitace. Nakonec uživatel zadá vstupní a výstupní podmínky simulace, nastaví čas a krok výpočtu a spustí analýzu. Zároveň s výpočtem je ihned prováděna animace sestavy i s výslednými reakcemi v určených místech. Celý průběh animace lze zaznamenat do souboru *.avi či *.wmv . U tohoto modulu musíme počítat s delším časem výpočtu u složitějších simulací, to vše však závisí na použitém hardwaru. Nechybí zde ani přehledné prostředí pro tvorbu grafů, do kterých si lze nechat vykreslit polohu, rychlost, zrychlení, síly a momenty. Vypočítané hodnoty je možné samozřejmě převést do MS Excelu – v takovémto případě se MS Excel automaticky otevře a okamžitě vygeneruje příslušný graf a sestaví tabulku hodnot.

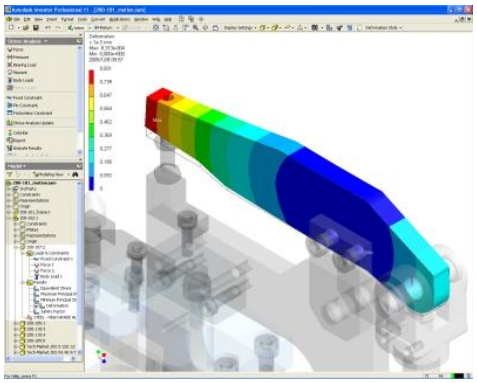




-

Analýza Metodou Konečných prvků

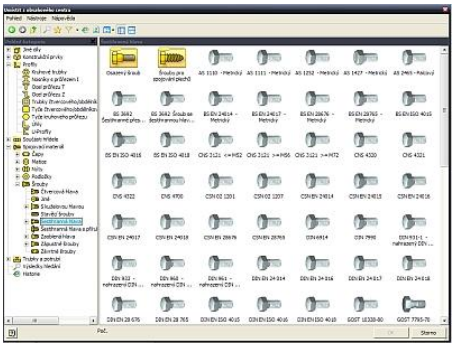
Modul je určen k řešení analýzy metodou konečných prvků u jednotlivých součástí. Dílci se přiřadí materiál a následně přímo v prostředí Inventoru určit podpory a zatížení. Podpora může být fixní určením plochy na modelu s možností posunu v jednotlivých osách. Zatížení může být definováno samostatnou silou, či momentem působícími na vybranou plochu, případně působící ve směru definovaným složkami základních os: x,y,z. Další možností je simulování tlaku rozloženého na ploše, zatížení válcových ploch od ložisek či případně definování gravitačního zrychlení, což umožňuje počítat s dynamikou stroje působící na analyzovanou součást. Podpory a zatížení jsou zobrazeny ve stromové struktuře prohlížeče Inventoru, pomocí které je možné je odstranit, či modifikovat. Výsledkem řešení je analýza napětí, deformace, faktoru bezpečnosti. Znázornění výsledků je provedeno barevnou mapou přímo na součásti, s ostrým, nebo vyhlazeným přechodem barev. Přiřazení barev rozsahům hodnot lze modifikovat jednoduše posunem hranice barvy ve stupnici barev. Zapnout lze pochopitelně i zobrazení sítě, či označení maxima a minima hledaného řešení na součásti. Výsledky lze exportovat jako report ve formátu HTML a přiložit k technické zprávě. Pro podrobnější řešení, pak lze využít export celého řešení do systému Ansys, který disponuje pokročilejšími nástroji pro analýzu, než modul Inventoru, který je prioritně určen pro potřeby konstruktéra jako návrhový a optimalizační nástroj.



-

Obsahové centrum

Dalším modul, který velice zefektivňuje a urychluje práci návrháře je obsahové centrum. Můžeme říci, že je to taková sbírka normalizovaných dílů, které si v případě potřeby lze z Obsahového centra vyzvednout. Při koupi Inventoru jsou k dispozici knihovny základních norem běžně používaných ve světě jako ISO, DIN, PARKER, ANSI a další. Osobně používám pro správu knihoven aplikaci Autodesk Vault, kde mám v podstatě odpojené nepotřebné knihovny, pro urychlení chodu a vyhledávání. Do Obsahového centra se dostaneme v modulu Sestava kliknutím v hlavním panelu na tlačítko Obsahové centrum. Otevře se nám tabulka, kde si můžeme vybrat ze spousty kategorií normalizovaných částí. Uvedu příklad na šroubu. V Inventoru jsem si vytvořil součást dírou se závitem pro šroub. Potřebuji vložit šroub, tedy vstoupím do Obsahového centra, najdu si kategorii spojovací materiál → šrouby → pak si vyberu, s jakou hlavou chci ten šroub použít a jednoduše si jej zvolím dle požadované normy. Jsou dvě možnosti jak šroub vložit. Tak buď, víme rozměr díry a tak si z tabulky, která nám vyskočí, po vybrání šroubu vyberu rozměry, nebo mám ještě možnost funkce *Autodrop.* Ta spočívá v tom, že po vybrání šroubu mně program vyzve k ukázání na díru se závitem, do které chci šroub vložit, a program už sám vygeneruje nejvhodnější šroub. Na uživateli je už jenom zeditování jeho délky, či případně vytvoření šroubového spoje, na který tam je také speciální funkce, která nám velmi urychluje vytváření spojů mezi součástmi.



-

Plechy

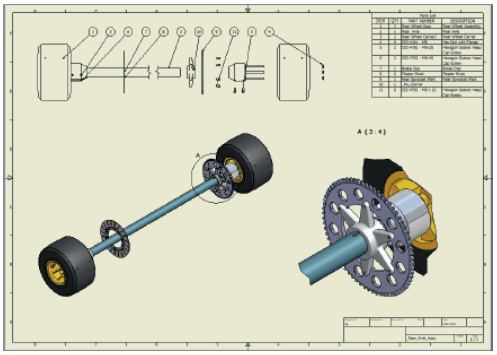
Autodesk Inventor podporuje také modelování součástí z plechů. Jde vlastně o další modul, kde si vytvořím základní plochu, zadám tloušťku a materiál. Pak plochu mohu pomocí nejrůznějších funkcí dále dotvářet. Lze přidávat lem, obrubu, nejrůzněji ohýbat, vyřezávat apod. Hlavní výhodou tohoto modulu je to, že nakonec mohu udělat z plechu rozvin a vložit jej do výkresové dokumentace, podle které ho pak při výrobě dokáží plech přesně zhotovit.

Svařované sestavy

Další nedílnou součástí tohoto programu je vytváření svařovaných sestav. Spojovací prvek je svár. Uživatel má k dispozici velkou škálu nástrojů pro práci a vytváření svařovacích spojů. Tyto spoje se pak při přenesení do výkresové dokumentace zobrazí jako sváry.

Výkresová dokumentace

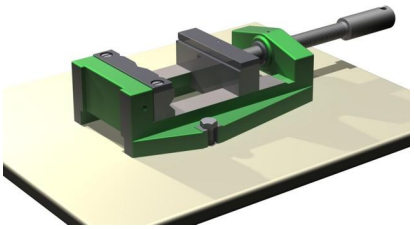
K čemu by nám bylo, kdybychom si navrhli složitý model, měli perfektně nasimulované všechny stavy výrobku, a nebyla by žádná výkresová dokumentace.
V Autodesk Inventoru tuto část zvládli přímo výborně. Modul pro vytváření výkresové dokumentace je podle mého názoru opět velice uživatelsky přívětivý. Nabízí široké možnosti ve vytváření výkresů. Tyto výkresy jsou přímo provázány s modelem, takže veškeré dodatečné úpravy na modelu se promítnou do výkresu. Vytvoření výkresu spočívá v několika málo základních operacích. Nejdříve si otevřeme prázdný výkres, poté vložíme do výkresu určitý základní pohled na dílec, a z něho následně vytváříme ostatní (promítnuté) pohledy. Nakonec už nám zbývá pouze výkres okótovat, přidat další nezbytné poznámky a máme hotovo. Snadná je také generace kusovníku se všemi díly sestavy.



-

Praktické zkušenosti

Se systémem Inventor mám již dlouhodobé praktické zkušenosti, které jsem nabil v průběhu let od svého otce. Na střední škole jsme měli výuku Inventoru ve čtvrtém ročníku, kde jsme zpracovávali školní projekt – model svěráku dle zadaných parametrů. Získali jsme možnost nechat si své zkušenosti na základě tohoto projektu certifikovat od společnosti Autodesk, čehož jsem využil. Dále jsem se během studia také účastnil mezinárodní soutěže Autodesk Academia Design, kde jsem v roce 2008 obsadil osmé místo.



Shrnutí

Dle mého názoru je Autodesk Inventor výborný program pro navrhování nových součástí. Je postaven na jádře ACIS, obsahuje téměř všechny nezbytné funkce pro navrhování. Kdybych tu měl vypisovat všechny moduly, které obsahuje, zřejmě by to dalo na samostatnou ročníkovou práci, a to zde ani není mým cílem. Na závěr představení tohoto programu ještě mohu dodat hardwarové nároky udávané společností Autodesk: na modelování sestav do 1000 částí by vám podle udaných parametrů mělo stačit 1GB RAM nebo více, 128MB Grafická karta podporující OPENGL nebo Direct3D a alespoň jeden procesor 2+ GHz. Na samotnou instalaci na disku je doporučeno mít alespoň 3,5GB volného místa. Zatím pro Autodesk Inventor 11 jsou oficiálně podporované systémy Windows 2000 nebo Windows XP Professional 32bit. Osobně jsem měl nějaký čas nainstalovaný systém Inventor na Windows XP-64bit, dle Autodesku, jako operační systém na kterém vám Autodesk Inventor 11 poběží, ale není pro něj nativní podpora využití 64bitové sběrnice. Podle vlastních zkušeností však mohu dodat, že mi Inventor běží rychleji na Windows XP Professional-32bit, a to i když jsem měl v době testování operační paměť 2GB RAM.

-

2.2. SolidWorks Office Premium 2007

Úvod

SolidWorks je parametrický modelář postavený na jádře Parasolid. SolidWorks byl od počátku vyvíjen nativně pro operační systém Windows, takže dokáže v dnešní době využít naplno všech jeho možností, které nabízí.

SolidWorks je programem společnosti SolidWorks Corporation, která byla založena roku 1993. Dva roky po jejím založení byla vypuštěna první verze SolidWorks 95, za kterou stáli lidé jako Michael Payne – zakladatel PTC (ProEngineer), Jon Hirschtick – Computervision(CADDS 5), Vic Leventhal – Cad Solutions. Tito lidé se snažili o využití svých bohatých zkušeností z oblasti "velkých modelářů" a jejich skloubení s výhodami, které nabízí operační systém Windows a to vše při zachování přijatelných cenových podmínek odpovídajícím cenové hladině CAD systémů střední třídy.

Výsledkem je snadno ovladatelný 3D systém, který je dnes velmi rozšířený. Nevyžaduje dlouhou dobu na přivyknutí a ovládá-li uživatel Windows, není většinou problém orientace v SolidWorks.

SolidWorks přistupuje k uživateli jako ke tvůrci určitého výrobku a usnadňuje mu realizaci jeho myšlenek. Tvorba modelu probíhá ve 3D prostoru, protože konstruktér nebo designér by měl přemýšlet o funkci výrobku, jeho vlastnostech a vzhledu, nikoli o tom, jak výrobek vypadá v jednotlivých pohledech nebo řezech. Tento problém za uživatele řeší systém.

SolidWorks je dodáván hned v několika verzích, takže si i menší konstrukční kanceláře, nepotřebující složité analýzy výrobků mohou vybrat.

Ceník - ceny produktu u dealera Top-Tech s.r.o.

SolidWorks 2008 – 198.000Kč

SolidWorks 2008 Office Professional – 237.000Kč

SolidWorks 2008 Office Premium – 285.000Kč

-

Verze

Jak jsem již zmiňoval výše, SolidWorks je možno koupit ve třech verzích, které se liší ve vybavení modulů a především cenou. Jak je vidět na přiloženém obrázku základní verze SolidWorks obsahuje pouze Samotný program s modulem pro analýzu integrity konstrukcí a modulem pro simulaci plnění plastových výlisků. Verze Office Professional obsahuje nástroje pro vizualizace, komunikaci a obecně ke zvýšení produktivity práce. A poslední nejvyšší verze konstrukčního programu Solid Works – Office Premium - obsahuje ještě dodatečně nástroje pro další podrobnější analýzy výrobků atd. V další části práce se budeme podrobněji věnovat jednotlivým modulům programu SolidWorks.

	SolidWorks	SolidWorks Office Professional	SolidWorks Office Premium
SolidWorks 3D CAD Software	✓	✓	✓
DWGeditor	✓	✓	✓
COSMOSXpress	✓	✓	✓
MoldflowXpress	✓	✓	✓
eDrawings Professional		✓	✓
SolidWorks Utilities		✓	✓
SolidWorks Animator		✓	✓
Toolbox		✓	✓
FeatureWorks		✓	✓
PhotoWorks		✓	✓
SolidWorks Task Scheduler		✓	✓
3D Instant Website		✓	✓
SolidWorks Design Checker		✓	✓
PDMWorks Workgroup		✓	✓
SolidWorks Routing			✓
COSMOSWorks Designer			✓
COSMOSMotion			✓
ScanTo3D			✓

✓ = included in that product

-

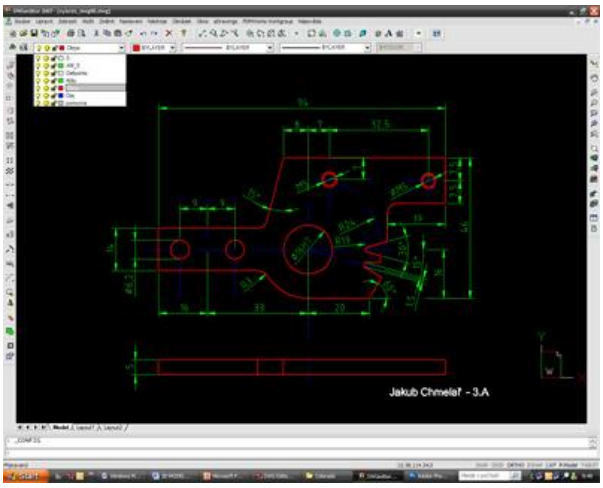
SolidWorks 3D CAD

V programu SolidWorks modelujeme ve dvou základních úrovních. První úrovní je díl. Díl si vytvoříme, můžeme ho editovat, případně na něm provádět nejrůznější analýzy. Můžeme u něho také měnit materiál, barvu a přidávat textury. Další úrovní je sestava, kterou tvoří více dílů pospojovaných do sebe. Jednotlivé díly jsou v sestavě pospojovány vazbami, které si může uživatel jednoduše nadefinovat. Samozřejmě na úrovni sestav je možné vkládat i podsestavy. Poslední úrovní je tvorba výkresové dokumentace. Ta má dnes i přes 3D revoluci svoji nezastupitelnou funkci v procesu výroby.

SolidWorks nabízí velmi pěkné a přehledné uživatelské rozhraní, které si každý může nakonfigurovat podle svého, přidáním některého z mnoha nabízených plovoucích menu, která jsou i nadále konfigurovatelná!

DWGeditor

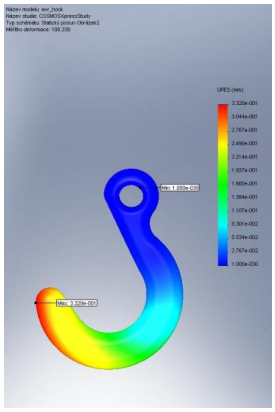
DWGeditor je velmi zajímavý program, který umožňuje uživatelům používajícím Solidworks vytvářet a editovat nativní soubory výkresů *.dwg, které jsou soubory AutoCADu. Uživatelské rozhraní programu je velmi podobné programu AutoCAD. Program pracuje s hladinami a tloušťkami čar. Zjistil jsem, že nabízí téměř všechny základní funkce programu AutoCAD.



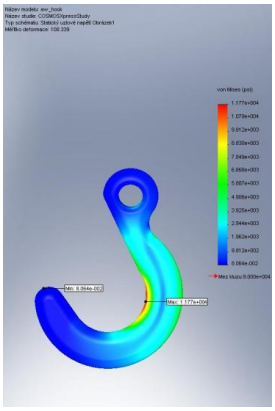
-

COSMOSXpress

COSMOSXpress je program pro analýzu pevnosti navržených výrobků. Práce s tímto modulem je velice jednoduchá, je však opět, jako u Inventoru potřeba alespoň trošku rozumět tomu, co dělám. Uživatel má k dispozici celou škálu různých materiálů, se kterými může pracovat. CosmosXpress umožňuje simulovat zatížení nebo tlak na součást a je schopen vypočítat a pak názorně zobrazit pomocí animace rozložení napětí na modelu či posun modelu včetně stupnice apod. Z takto vypočtené simulace můžeme automaticky vytvořit video ve formátu *.avi, zprávu v HTML verzi, nebo můžeme celou simulaci uložit do souboru eDrawings, který bude popsán níže.



Rozložení posunu při zatížení



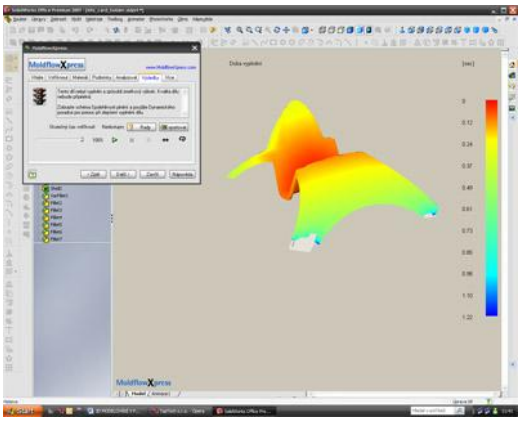
Rozložení napětí při zatížení

MoldflowXpress

MoldflowXpress je modul pro simulaci plnění plastových forem. Používá se jako první krok k otestování vyrobitelnosti výlisku přímo v prostředí programu SolidWorks. Uživatel má k dispozici databázi více než 20 druhů základních používaných plastových materiálů. MoldflowXpress dokáže po vypočtení analýzy vytvořit animaci plnění formy plastem. Výsledná data je možné opět exportovat do programu eDrawings. Použití MoldflowXpress je velmi jednoduché, uživatel vytvoří výlisek, zapne si program MoldflowXpress, ve kterém stačí zadat místo vtoku (MoldflowXpress umí pracovat pouze s jedním místem vtoku, tohoto omezení se lze však zbavit zakoupením Moldflow Plastics Advisers, což je jakýsi rozšiřující modul pro MoldflowXpress), poté materiál z knihovny. Nakonec podmínky jako teplota vstřikovaného plastu či teplotu formování a stisknout tlačítko spustit. Program vygeneruje podrobný model, kde je barevně odlišená doba natékání a upozorní na problémy, jaké by se

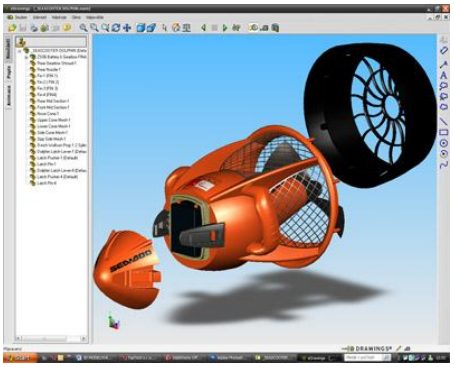
-

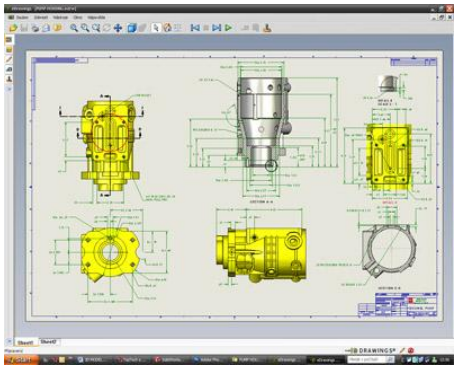
mohly vyskytnout při vstřikování „naostro“(např. vznik studených spojů). Také navrhne řešení. Na přiloženém obrázku je vidět, že forma není zcela vyplněna. Program mi v tomto případě navrhl rozšířit tloušťku stěny či změnit místo vstřikování.



eDrawings Professional

Již dříve zmiňovaný eDrawings Professional slouží ke komunikaci konstruktéra/designéra se zákazníkem, který si může prohlédnout výrobek ještě v digitální formě a nemusí vůbec mít žádný CAD systém. Program eDrawings je navržen tak, aby maximálně skomprimoval model vytvořený v programu Solidworks, a bylo jej možné zaslat např. elektronickou poštou. eDrawings vytvoří jeden malý soubor s příponou *.exe nebo *.html, který si zákazník otevře a může si prohlížet 3D model, rozložit jej na jednotlivé díly a s těmi pak manipulovat. Umí samozřejmě přidávat k modelu poznámky, což výrazně urychluje komunikaci. Je také možné do eDrawings uložit 2D výkres.

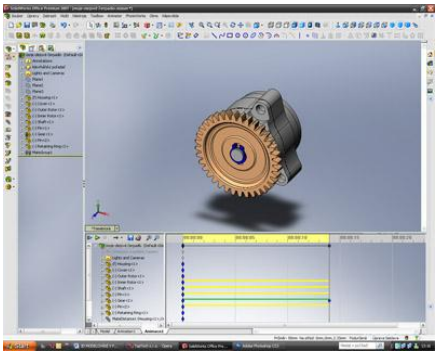


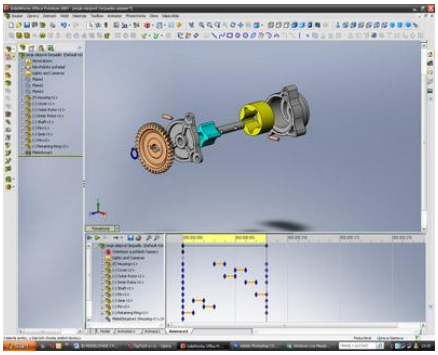


-

SolidWorks Animátor

SolidWorks Animátor umožňuje uživateli vytvářet animace svých návrhů a poté je ukládat do souborů *.avi. SolidWorks Animátor je plně integrován do prostředí SolidWorks, takže jej lze kdykoli během práce vyvolat jako lištu a zase naopak zavřít. Animování můžeme provádět dvěma způsoby a to pomocí Průvodce animací, který umožní vytváření jednoduchých animací, jako rotace objektu kolem své osy či rozpad objektu, nebo můžeme vytvářet složitější - propracovanější animace, pomocí metody klíčování. Tato metoda spočívá v ručním definování pohybu a dráhy jednotlivých komponent, kde si na časové ose před každým pohybem zadáme délku pohybu pro daný komponent a poté jej animujeme. Tato metoda je sice složitější, ale jdou pomocí ní vytvářet opravdu působivé animace. Co se týče dalších možností při animování, tak nesmíme opomenout možnost postupného slábnutí viditelnosti komponenty až úplného zmizení, nebo změna v zobrazení, kde může postupně přejít do drátového zobrazení objektu. Také mě velmi zaujala možnost postupné změny barvy objektu, která by se dala využít při animaci zahřívání nějaké součástky, kdy by se barva postupně měnila z defaultní barvy kovu do červena.





-

PhotoWorks

PhotoWorks je špičkový nástroj pro vytváření fotorealistických vizualizací 3D CAD modelů, které výrazně zvyšují účinek nabídek a prezentací. PhotoWorks je opět plně integrován do prostředí SolidWorks, takže se do něj lze přepnout prakticky jedním kliknutím tlačítka na myši. Velice mile mě překvapila široká možnost v nastavování různých parametrů, vizualizace. Opět je k dispozici rozsáhlá databáze materiálů, které může konstruktér použít při vizualizaci. Je také nesmírnou výhodou, že si uživatel může každý materiál sám ještě dotvářet, a to přidávání odrazivosti, nastavování indexu lomu a další. Máme také k dispozici velké množství přednastavených scén, do kterých můžeme svoje návrhy





umísťovat.

SolidWorks Toolbox

Solidworks Toolbox je jakousi knihovou normalizovaných součástí, které se běžně používají v průmyslu po celém světě. Modul Toolbox je navržen, tak aby byl co nejjednodušší na použití. Stačí si otevřít knihovnu, vybrat příslušnou součástku, zadat parametry a přetáhnout myší do modelu, kde se pomocí „chytrých“ vazeb sama umístí, nebo ji samozřejmě můžeme umístit ručně. Je také možno použít tzv. inteligentních šroubů, kdy si přednastavíme, že určité typy šroubů se budou automaticky bez zadávání dalších parametrů vkládat do určitého typu děr. Trošku mě překvapilo, že přímo v programu SolidWorks nejsou knihovny s českou normou ČSN, avšak po konzultaci s prodávajícím dealerem Top-Tech s.r.o. jsem se dozvěděl, že oni každému, kdo si koupí produkt SolidWorks Premium, dávají zdarma strojírenskou nadstavbu TDS-TECHNIK (www.tds-technik.cz), která obsahuje velmi rozsáhlou databázi normalizovaných dílů norem ČSN EN ISO, a umí i mnoho dalších věcí, jakou jsou strojírenské pevnostní výpočty apod. Samostatně pak se její cena pohybuje okolo 19.000Kč

-

PDMWorks

PDMWorks je modul řešící správu dat v projektech. Umožňuje plnou kontrolu nad historií vytvořených dílů, zachovává historii, takže si pak každý může dohledat kdo, kdy a jakou změnu provedl na tom kterém dílu. PDMWorks disponuje také funkcemi pro rychlé vyhledávání a přístup, takže si lze zkontrolovat, jestli podobná součástka již není vytvořena třeba pro jiný projekt a pak ji použít v projektu stávajícím.

COSMOSWorks

COSMOSWorks můžeme přirovnat ke kompletní sbírce nástrojů pro simulaci nejrůznějších stavů výrobku. COSMOSWorks toho umí opravdu hodně – od běžné analýzy deformace výrobku až po analýzu působení teploty na výrobek. Celý modul COSMOSWorks můžeme rozdělit na 3 části, které se liší v komplexnosti a možnostech simulací – Designer, Professional a Advanced Professional. Poslední jmenovaný již dokáže vytvořit velmi složité analýzy nelineárního zatížení, nebo simulace výrobků, které jsou vytvořeny z kompozitních materiálů. Výsledky analýzy se zobrazí pomocí názorných barevných přechodů se stupnicí, takže i opravdový laik pozná, kde je výrobek asi nejvíce namáhán. Práce s tímto modulem opět vyžaduje alespoň základní znalosti uživatele v oboru mechaniky. Co jsem si zkoušel nejrůznější cvičení podle výukových tutoriálů v tomto modulu, tak mohu říci že práce v něm je vesměs velice snadná a člověk si velice rychle osvojí postupy, které jsou velice obdobné – určení o jakou analýzu jde →vybrání materiálu→ určení pevných bodů → zadání místa působení síly či jiného namáhajícího elementu. Pak již stačí pouze nechat počítač spočítat výsledky. Celý systém analýzy funguje na principu sítě, na kterou si celý výrobek program rozebere. Konstruktér si může určit, jak moc složitou tuto síť chce. Avšak musíme brát v potaz, že čím složitější síť tím déle nám budou trvat všechny kalkulace.

COSMOSFloWorks

Modul, který mě velice zaujal, avšak bohužel jsem neměl možnost ho více poznat je COSMOSFloWorks. Ten slouží k simulaci proudění kapalin a plynů. Je to velice užitečný nástroj, pomocí kterého lze nasimulovat například proudění vzduchu kolem křídla letadla či turbulence v proudění například v počítačové skříni. Také můžeme sledovat změny teploty závisle na okolních podmínkách (ventilátor, či naopak těleso vyzařující nějaké teplo). Celý modul obsahuje opravdu mnoho různých velice zajímavých nástrojů, které usnadňují konstruktérům celý proces navrhování. Ovládání je velice jednoduché, uživatel je veden

-

pomocí průvodce, kde je na všechny potřebné vstupní informace dotazován. Velkou výhodou je opět úplná integrace do prostředí SolidWorks.

COSMOSMotion

Velice populárním nástrojem mnoha konstruktérů je modul zvaný COSMOSMotion. Ten umožňuje sledovat nejrůznější chování celých sestav v pohybu. Pomáhá konstruktérům pochopit chování jejich sestav v pohybu a určování jejich vlastnostní. Tento modul je standardně obsažen v COSMOSWorks Professional, a COSMOSWorks Advanced Professional.

Závěr

Podle mého názoru je program SolidWorks velmi výkonným pomocníkem každého konstruktéra či designéra. Velmi se mi líbí na tomto programu intuitivní práce. Než jsem se pustil do psaní této části práce, chtěl jsem se s programem seznámit. Pomohly mi v tom zabudované online výukové lekce, které s vámi podrobně projdou nejdůležitější funkce programu. Osobně, pochopení a naučení se s programem, mně trvalo přibližně 2 dny, což si myslím, že je opravdu krátká doba pro takto rozsáhlý program. Pro zvýšení svých zkušeností s programem jsem se pustil do modelování pásového bagru. Na tomto projektu bych chtěl názorně dále během této práce předvést různé možnosti programu SolidWorks a postupy.

Co se týče hardwarové náročnosti programu, musím říci, že je na tom o poznání lépe než výše zmiňovaný Autodesk Inventor. Pro plynulý chod, jak uvádí společnost SolidWorks Corporation, je nutné mít alespoň 512MB RAM a procesor s 32bit architekturou. Doporučená velikost paměti je však 2GB RAM či pro velké sestavy (nad 10 tisíc dílů) 6GB RAM, ale k tomu je však zapotřebí procesor s 64bit architekturou, kvůli omezeným možnostem práce s pamětí, jakými disponují 32bit procesory a to je hodnota přibližně 4096 MB. Z grafických karet je doporučeno vlastnit profesionální workstationovou OpenGL kartu. Mě však naprosto plynule běží i na mé GeForce 6800GS, což je spíše herní grafická karta. Měl bych se také tedy ještě zmínit jak je to s podporou 64bit systémů Windows. SolidWorks verze 2007 už je dodáván jako plně nativní pro Windows XP 64bit, a měl by po instalaci Service Packu být schopen i plné funkčnosti na Windows Vista (v době psaní této části práce v prodeji cca 3 měsíce).

-

3. Modelování v programu SolidWorks

V kapitole Modelování v programu SolidWorks chci ukázat, za pomocí názorných obrázků a animací, základní principy práce během modelování projektu. Není mým záměrem zde ukazovat postup u každé jednotlivé součástky, bylo by to příliš náročné a nemělo by to význam.

3.1. Projekt: Bagr

Jako praktickou část své ročníkové práce jsem se rozhodl vytvořit model pásového bagru. Zřejmě se mnoho lidí bude ptát, proč jsem si právě vybral pásový bagr. Je to především proto, že jeden takový pracuje v kamenolomu nedaleko místa, kde bydlím. Dalším argumentem je jeho tvarová příznivost, úměrná k mým znalostem tohoto programu.

Samotný proces vytváření virtuálního modelu bagru zahrnuje několik částí:

- Příprava projektu – výběr předlohy, získávání co největšího počtu podkladových materiálů a informací, fotky, cíl projektu
- Příprava k modelování – určení detailnosti a měřítka, rozvržení sestav, vytvoření základních skic s rozměry
- Modelování – modelování dílů a skládání sestav
- Vizualizace – vizualizace modelu, rendering
- Vytvoření prezentace, další kroky
- Vytvoření reálného modelu pomocí 3D tisku Rapid prototyping

Příprava projektu

Co se týče přípravy projektu, když jsem měl vybráno, co budu modelovat, musel jsem o dotyčném stroji zjistit co nejvíce informací. Když jsem si zjistil přesný typ stroje, usedl jsem k počítači a použil mocný nástroj vyhledávání – Google. Vzhledem k tomu, že se jednalo o bagr české výroby DH 621, nenašel jsem o něm příliš informací, natož nějakou dokumentaci či rozměry. Rozhodl jsem se, že si alespoň ty základní rozměry změřím sám a že si pořídím rozsáhlou foto dokumentaci. – viz ukázky referenčních fotografií.

-

Příprava modelování

Další částí je příprava k modelování. Zde jsem se musel především rozhodnout, jak moc detailní model požaduji. Vzhledem k tomu, že se jedná pouze o pohledový model, který není určen pro další použití např. jako předloha pro výrobu, stačilo vymodelovat pouze vnější části modelu.

Dalším krokem bylo rozvržení modelu do jednotlivých sestav. Řekl jsem si, že si tímto model nebudu příliš komplikovat a tak jsem si vystačil pouze s několika sestavami hlavních částí – Podvozek, rameno a tělo. Některé tyto části ještě obsahují menší podsestavy.

Závěrečným krokem bylo vytvoření papírové skici s rozměry. Několik základních rozměrů jsem si změřil sám na bagru a další jsem si musel v průběhu modelování poměrem spočítat ze získaných fotografií. Naposledy jsem si ještě určil měřítko - 1:1.

Modelování

Když jsem usoudil, že mám veškeré materiály, které jsem potřeboval k zahájení 3D modelování, pustil jsem se do první základní části – podvozku, poté následovalo modelování samotného těla bagru, rameno a naposledy detaily. Ještě bych měl dodat, že veškeré normalizované součástky jako jsou šrouby a matice, jsem vytvářel pouze jako pohledové modely a ne součásti z *ToolBoxu* – knihovny normalizovaných součástí.

- **Podvozek**

První částí bylo vytvoření podvozku. Ten jsem začal vytvořením skici a jejím vysunutím do prostoru. Jako vysunutí jsem volil souměrné – tj. prvek se vysune od skici rovnoměrně na obě strany. Bylo to kvůli tomu, abych mohl případně dále využít středové základní roviny. Při vytváření skici jsem se držel několika zásad, které mi pak dále velice zjednodušili práci:

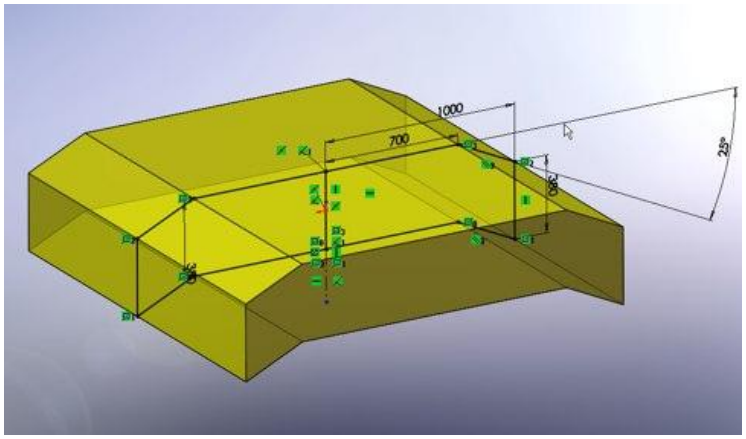
- Umístění skici do středu souřadného systému – možnost využití referenčních rovin
- Zafixování skici do středu systému – zabráním tím spoustě problémům při editování skici a při zavazbování modelu v sestavách
- Použití co nejméně kót ve skice avšak používat vazby skici – jednoduší editace, lepší přehlednost

Vzhledem k tomu, že se jednalo o osově souměrný díl, použil jsem ve skice funkci zrcadlení, která zrcadlí vybrané entity (jednotlivé prvky skici – úsečka, kružnice atd.)

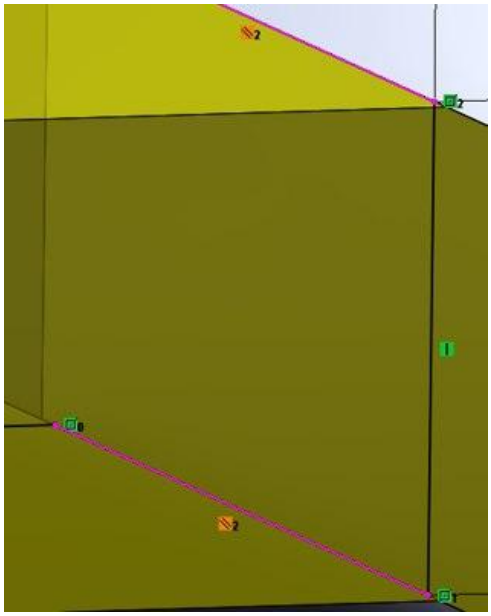
-

s vazbou na ty originální, takže při editaci jedné strany se zedituje i ta druhá. Velmi důležité je i během vytváření skici používat konstrukční čáry, které nám také velmi zjednoduší práci.

Na programu SolidWorks se mi opravdu líbí, že docela přehledně ukazuje i vazby každé čáry ve skice, takže ihned na najetí na vazbu se mi znázorní se kterými dalšími prvky je mnou zvolaná čára zavazbena.



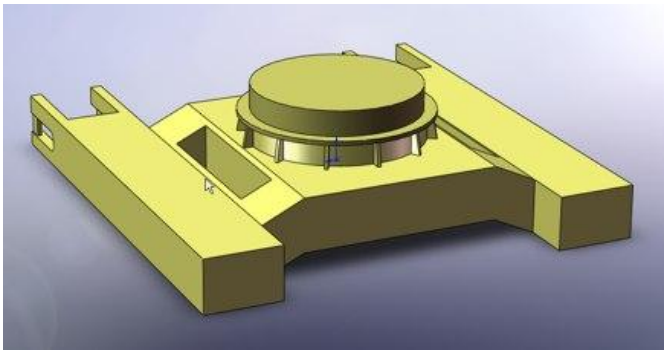
Náčrt a vysunutí



-

Zobrazení vazeb v náčrtu

Za pomocí výše popsaných postupů jsem dokončil celý díl podvozku a mohl jsem se pustit do další části.

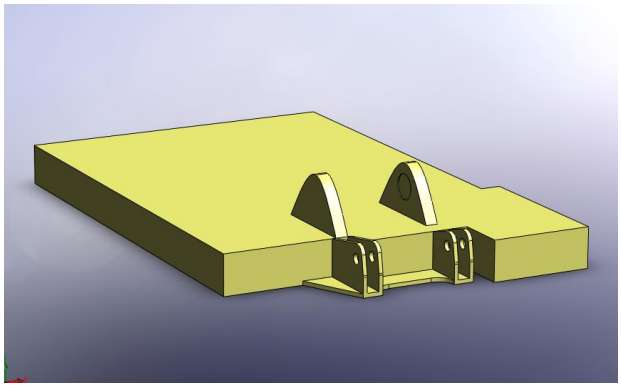


Obrázek součásti podvozek

- **Tělo Bagru**

Tělo bagru je tvořeno několika hlavními částmi – základna, motorová část, závaží, kabina a také spoustou menších detailů jako jsou chladící vstupy, panty apod. Jak jsem již zmiňoval na začátku, v první fázi modelování jsem se soustředil na základní tvar.

Nejprve jsem začal modelovat základnu. Jde v podstatě o vytvarovanou desku, na kterou budu dále usazovat další části. Viz. Obr dole.



-

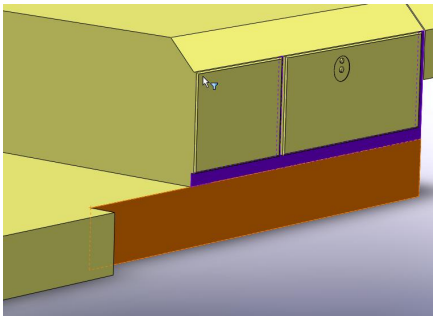
Tvar a rozměry této základny byly jedny z mála, které jsem měl přesně změřené podle skutečného bagru. Zachování proporcí bylo v tomto případě velice důležité, protože jsem potřeboval dosáhnout co nejvěrohodnějšího tvaru.

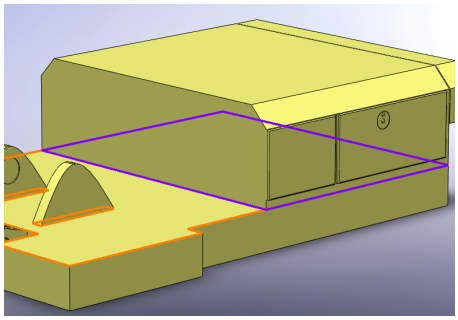
Další částí bylo vytvoření bloku motoru. Celý náčrt tohoto dílu jsem začal již v sestavě a velmi se mi tu nabízelo použití provázanosti rozměrů (tzv. adaptivity) mezi základní deskou a blokem motoru. Abych správně vysvětlil tento pojem. V podstatě promítnu rozměry, v tomto případě šířku základní desky, do nového náčrtu a použiji ji při modelování bloku motoru. Pokud bych v budoucnu změnil šířku základní desky, změní se i šířka bloku motoru. Tato funkce má velký význam především při konstruování, kde na sebe musí přímo navazovat více součástí a jsou na sobě závislé rozměry. Editace jedné součásti změní i příslušný rozměr u další součásti. Podobným způsobem jsem modeloval i zbylé součásti těla bagru a mohl jsem se pustit do jejich umísťování.

Umístění součástí do požadované polohy se provádí použitím vazeb. V programu SolidWorks máme hned 6 standardních vazeb a několik dalších v kategoriích strojní vazby a upřesňující vazby.

K zavazbení bloku motoru k základní desce jsem použil základní vazbu sjednocení. Jde v podstatě o slepení dvou rovin k sobě. V obrázku vlevo je vidět použití zarovnané sjednocení, a na druhém obrázku je vidět opačné sjednocení.

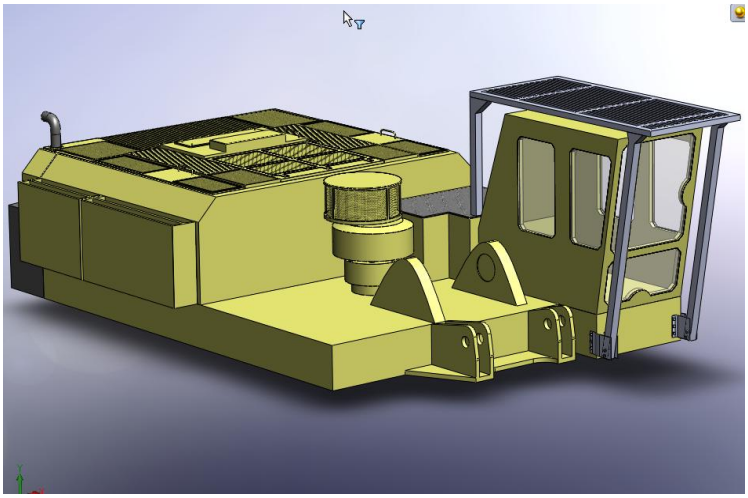
-

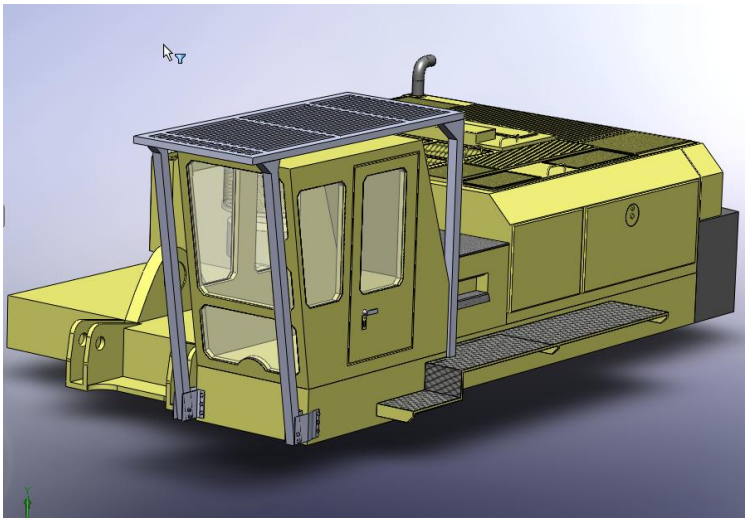




-

Postupy popsanými výše jsem dotvořil zbylé části těla bagru. U všech jsem použil pouze základní funkce modelování a umísťování. Na obrázcích níže je vidět dokončená sestava těla bagru





-

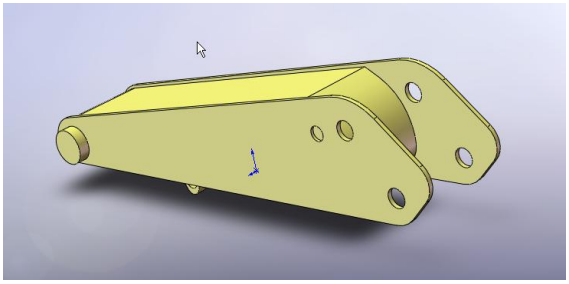
- **Rameno**

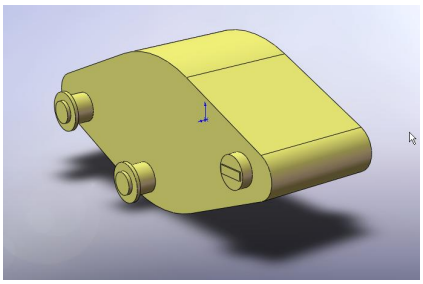
Nedílnou součástí bagru je jeho rameno a radlice. Tento prvek se mi obzvláště líbí, protože obsahuje několik technicky velice zajímavých komponent, jako jsou hydraulické válce s písty, čepy a klouby a pákové mechanismy.

Rameno jsem začal tvořit od jeho největší části a postupoval jsem směrem k jeho konci. Opět jsem měl k dispozici několik fotek, které mi výrazně pomohly při modelování. Využíval jsem, zejména při vytváření děr pro čepy, funkci adaptivity, aby mi přesně odpovídaly vzdálenosti a průměry otvorů, respektive čepů.





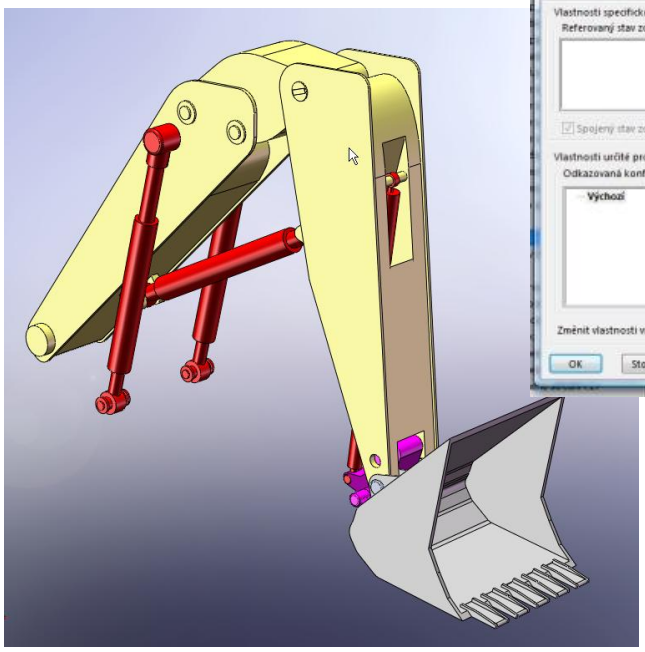


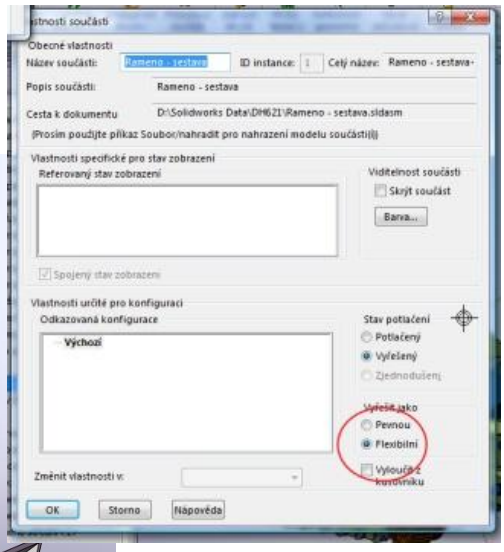


-

Když jsem dokončil modelování základních součástí ramene, využil jsem pro sestavování vazby soustřednosti, které jsem aplikoval na průměry otvorů a příslušných čepů. Abych součásti umístil i osově, využil jsem vazby sjednocení mezi dvěma plochami, které se mají dotýkat.

Docela velký problém mně způsobovalo zavazbení hlavního zvedacího pístu. Bylo to proto, že rameno je tvořeno jednou sestavou a mělo být uchyceno k sestavě tělo ve 3 místech – v hlavním čepu ramene a ve dvou pístech. Řešení tohoto problému spočívalo, jak jsem se dozvěděl od mé externí konzultantky paní Rybkové, v tom že celou sestavu ramene převedu na flexibilní v panelu vlastnosti sestavy. Poté již mohu upevnit celou sestavu přesně tam, kde potřebuji.





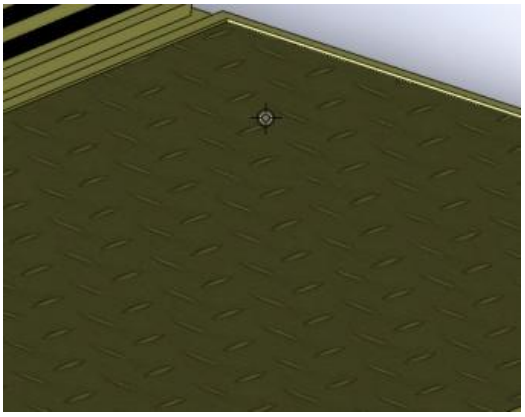
- **Pohledové detaily**

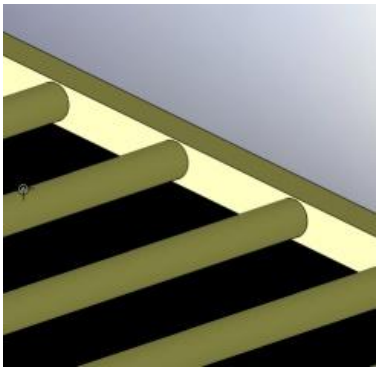
Jak je známo, když vytváříme model, tak to, co z něho dělá opravdu téměř dokonalou kopii reálného předmětu, jsou detaily. Zde mám, namysli například chladiče motoru, různé servisní vstupy a viditelné šrouby. V průběhu modelování jsem se rozmýšlel, jakým způsobem budu tyto detaily dělat. Nakonec jsem se rozhodl pro pouze povrchové výstupky, které by imitovaly funkční otvory či chladiče.

-

Jak je vidět na obrázku, tak ty části, kde je v reálu vidět do motoru, jsem vytvořil pomocí vyhloubení otvoru do kostky, aby vznikla jakási vana, z jejíchž okrajů jsem vysunul kulaté tyče. Abych docílil efektu hloubky, tak jsem „dno“ té vany obarvil na černo.

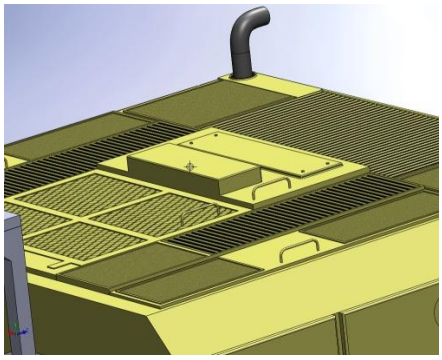
Ostatní detaily jako jsou třeba stupačky, které jsou ze slzičkového plechu, jsem imitoval za pomocí textury tohoto plechu, kterou jsem si vzal z knihovny materiálů.





A jak vypadá výsledek po dokončení všech detailů se můžete přesvědčit na obrázcích pod textem – srovnání fotografie s virtuálním modelem.

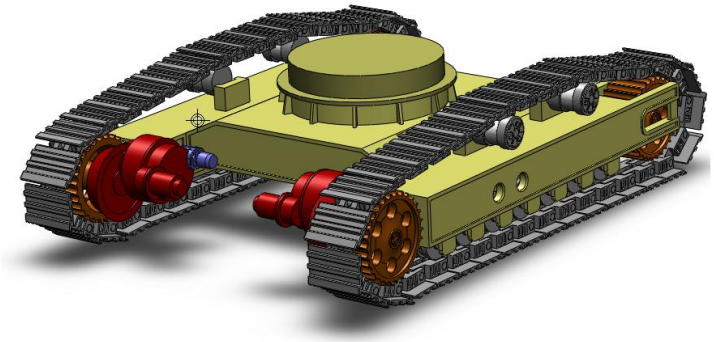




-

Pásy

Velkou výzvou pro mě bylo vytvoření pásů. Jedná se o spoustu do sebe zapadajících článků, které se musí v jednom místě spolu potkat. Začal jsem vytvořením jednoho článku pásu. Dále jsem si v sestavě podvozku nakreslil křivku, po které tento pás bude vést, a nakonec jsem začal tyto pásy vkládat. První článek jsem umístil tečnou vazbou spodní část článku ke křivce a střední pomocnou rovinou jsem si článek umístil tak aby vyhovoval mým představám. Další články jsem postupně vazbil jeden na druhý vazbami na čepy, tečně ke křivce a střední rovinou na rovinu toho předchozího pásu. Tímto způsobem jsem pokračoval, až jsem měl hotový pás na jedné straně. Druhou stranu jsem pro ulehčení práce vytvořil zrcadlením již hotové strany.

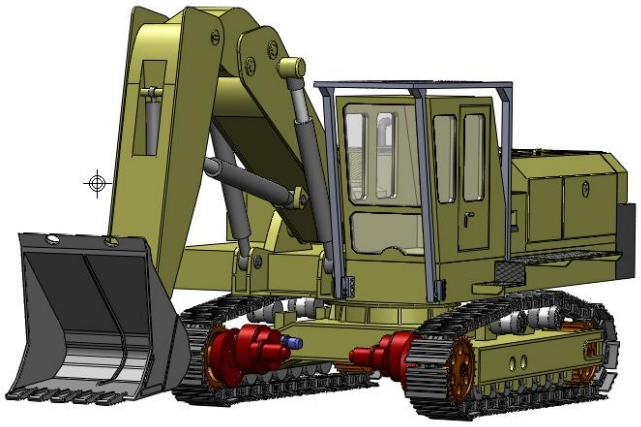


Finalizace 3D modelu bagru

Když jsem měl všechny potřebné díly hotové, mohl jsem tedy konečně složit vše dohromady. Většinu dílů jsem modeloval v sestavách, které jsem spojoval již „za chodu“, takže mi pouze zbývalo složit tyto dílčí sestavy dohromady. Postupoval jsem tedy od podvozku, ke kterému jsem připojil sestavu těla se sestavou ramene, kterou, jak je již popsáno výše, jsem převedl na flexibilní.

Nyní bych tedy měl mít celý model bagru hotový. Finální model je vidět níže na obrázku. Vzhledem k tomu že název mé ročníkové práce je využití CAD v technické praxi, budu se dále zabývat praktickým využitím takovýchto digitálních 3D modelů.

-



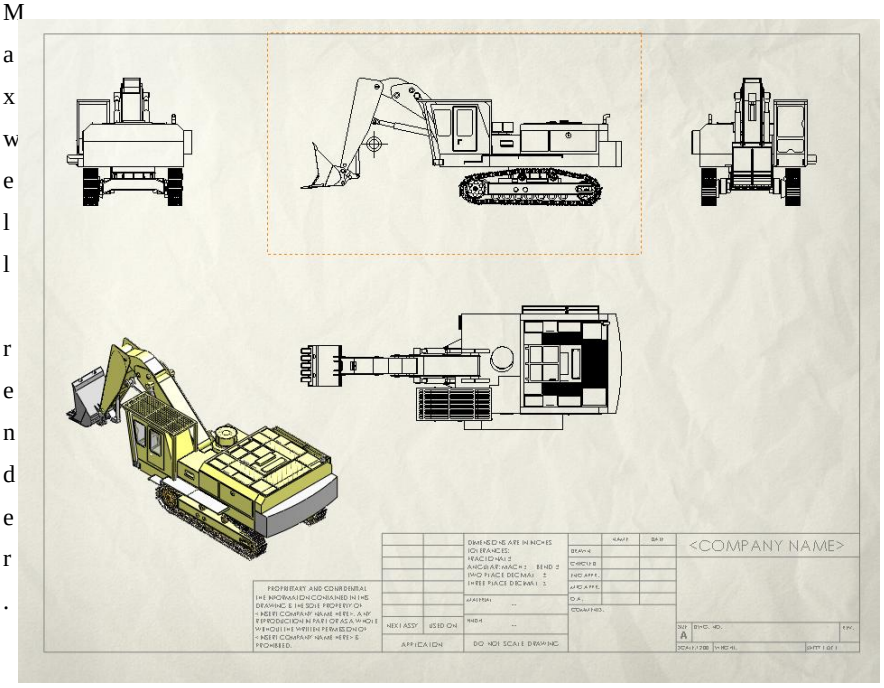


-

3.2. Návaznost 3D modelu

Model bagru je hotov a teď z něho můžeme vycházet v dalších krocích. Z vytvořeného modelu můžeme snadno vygenerovat výkresy jakékoli součástky či sestavy pouhým stisknutím tlačítka. Samozřejmostí je provázanost výkres-model, takže pokud cokoli změníme ve výkrese, změní se to i v modelu a naopak.

Mezi další kroky můžeme například zařadit mechanické a dynamické analýzy modelu, export dat pro programy, které umí generovat programy pro obráběcí NC a CNC či pro programy na vytváření forem plastových výlisků. Jedním takovým krokem, o který jsem se zajímal více, je možnost vytvoření realistických vizualizací z modelu. Samozřejmě spousta programů, jak jsem již výše popisoval, obsahuje modul pro vizualizaci výrobků. Musím však říci, že ač mají tyto vizualizace vysokou kvalitu, některé další, specializované programy (3ds Max, Studio VIZ, Maya aj.), dokáží vytvořit daleko realističtější vizualizace. Je to především proto, že mají pokročilejší nastavování materiálů (mám na mysli možnost nastavení desítek parametrů) a také velmi kvalitní renderovací systémy založené na vypočítávání fyzikálních vlastností světla jakou jsou Mental ray, Brazil, NVIDIA Gelato či M



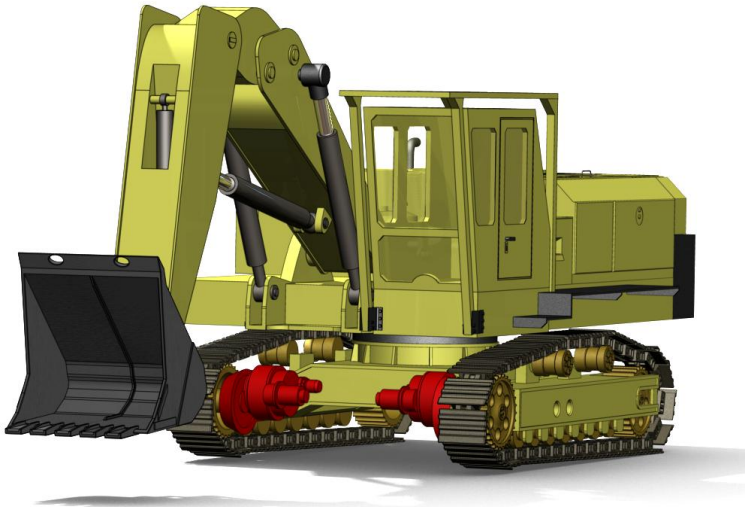
a
x
w
e
l
l

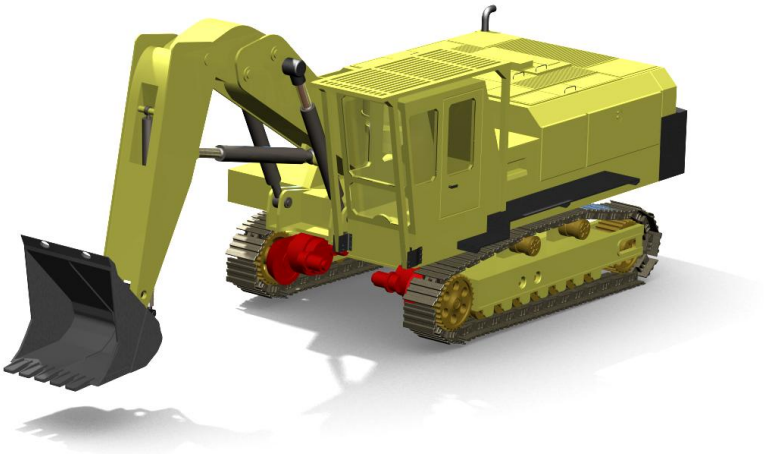
r
e
n
d
e
r
.

-

Vizualizace

Vizualizace z programu SolidWorks





-



Vizualizace z programu 3ds Max

-

Tvorba prototypů

- **3D tisk – Rapid Prototyping**

Nedávno jsem se setkal s velmi zajímavou technologií tvorby funkčních prototypů, která velmi výrazně zkracuje čas potřebný k zahájení výroby prvních „ostrých“ verzí výrobku. Nazývá se Rapid prototyping neboli 3D tisk. Existuje více principů, na kterých tyto 3D tiskárny fungují.

Osobně jsem se seznámil s tiskárnou od Izraelské firmy Objet, která pracuje na principu postupného nanášení plastové hmoty po malých vrstvách a jejich ustálení pomocí UV lampy. Tato metoda dosahuje přesnosti 0,2mm a síla jedné nanášené vrstvy je 0,16mm. Na místa, kde není žádný materiál, se kvůli zpevnění přidává tzv. podpůrný materiál, který se na konci procesu musí mechanicky odstranit. Díky tomu můžeme vytvořit například celý, již spojený řetěz složený z jednotlivých článků, mezi kterými je právě tento materiál. Po jeho odstranění můžeme jednotlivými články řetězu pohybovat.

Export dat probíhá z CADu, ve kterém jsme výrobek vytvořili, do formátu *.stl nebo *.step, který je schopný řídící program 3D tiskárny načíst a zpracovat.

Myslím, že využívat tuto metodu na školách by bylo pro studenty velice přínosné, protože by pro ně celý proces neskončil jen vytvořením 3D modelu na počítači, ale mohli by si ho i vytisknout. To by mohlo být dle mého názoru dobrým motivačním prvkem k výuce CAD na školách.

Problémem však je otázka prostředků. Nákup 3D tiskárny je stále ještě otázkou téměř milionu korun, a také její provoz není příliš levnou záležitostí. Stálo by také zauvažovat o získání těchto prostředků od státu formou dotací nebo z evropských fondů.

- **Frézování z umělé pryskyřice**

Další metoda, kterou bych chtěl uvést, je NC obrábění frézováním umělé pryskyřice. Tuto metodu jsem ve své práci použil k vytvoření reálného modelu ramene bagru. Velmi ochotně mi s tím pomohl pan Otakar Pěnička, Dvůr Králové nad Labem, který pomocí NC gravírovacího frézovacího stroje zhotovil po předání 3D dat modely.

Tato metoda však vyžaduje ještě následné dotvoření modelů povrchovými úpravami, avšak při vytváření větších modelů se finančně vyplatí více než 3D tisk

Součásti vyrobené oběma metodami jsem ponechal bez povrchové úpravy (přebroušení, tmelení a nástřik) pro názornost technologie výroby.

-

4. Výuková pomůcka – online: info.spsnome.cz

Dalším cílem mé ročníkové práce, bylo zpracování online výukové pomůcky pro výuku CAD s programem SolidWorks a její vystavení na stránkách info.spsnome.cz . Rozhodl jsem se podrobně rozpracovat sestavu ramene.

Online: info.spsnome.cz/pmwiki.php/CAD/Solidw00

Systém PmWiki

K vytvoření jsem použil redakční systém PmWiki, který mi byl doporučen mým konzultantem. Již po několika minutách používání jsem si ho velmi oblíbil. Bylo to především pro jeho jednoduchost a přístup přímo z webového prohlížeče. Pokud jsem chtěl do připravené stránky vepisovat text, jednoduše jsem dvakrát poklepal myší do stránky a po zadání příslušného hesla jsem se dostal do editovacího módu.

Formátování textu probíhá pomocí přednastavených značek. Ty nejčastěji používané (tučné písmo, odrážky, vodorovná čára a další) jsou znázorněny na liště nad editovacím oknem, o dalších se můžeme dozvědět ve velice pěkně napsaném manuálu. Do textů se také dají pomocí příslušných příkazů vkládat různé multimediální soubory jako obrázky, zvuky či animace. Když máme připravené místo pro daný soubor, jednoduše klepneme na odkaz pro tento soubor, systém nás sám navede do módu pro nahrávání obrázku, kde jej vybereme z disku a nahrajeme.

Dle mého názoru jsou redakční systémy tohoto druhu velmi vhodné k tvoření výukových pomůcek. K jejich ovládání není potřeba znalost HTML jazyka, takže se s nimi naučí pracovat téměř každý. Velmi příznivé je také to, že studenti k nim mohou přistupovat i z domova, takže si mohou případně více osvojovat postupy, které byly demonstrovány během výuky.

-

Výuková pomůcka

Výuková pomůcka nepopisuje příliš podrobně postupy modelování. Je určena do školní výuky, kde před posluchači stojí zkušený instruktor či učitel, který všechny vede. Obsahuje také kromě popsaných postupů názorné obrázky, popisující řešení konkrétních situací.

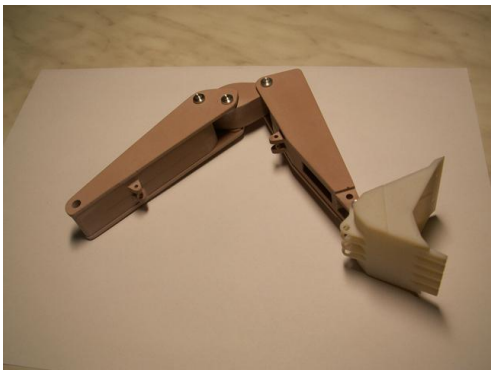
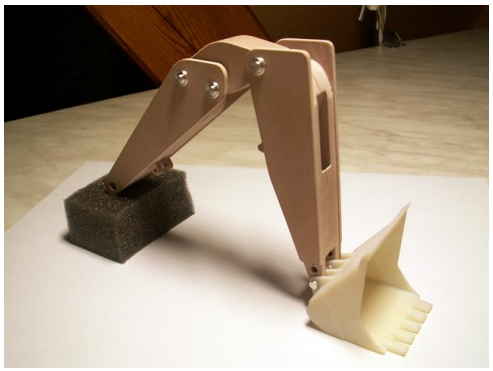
Na hlavní stránce je možno najít několik informací o autorovi práce, zajímavé ukázky z celé práce a také jakýsi rozcestník s odkazy na jednotlivé části pomůcky, které jsou chronologicky seřazeny podle postupu modelování. V jednotlivých krocích jsou při řešení obtížných situací obrázky, které si může čtenář vyvolat klepnutím na hypertextový odkaz u vyznačeného slova. Pro návrat pak stačí znovu klepnout na obrázek. Při dokončení dílu se není nutné vracet na úvodní stránku, ale klepnutím na odkaz v horní části stránky se automaticky dostaneme k další části. Dále na hlavní stránce nalezneme cíle projektu a také odkazy na zajímavé ukázky z práce.

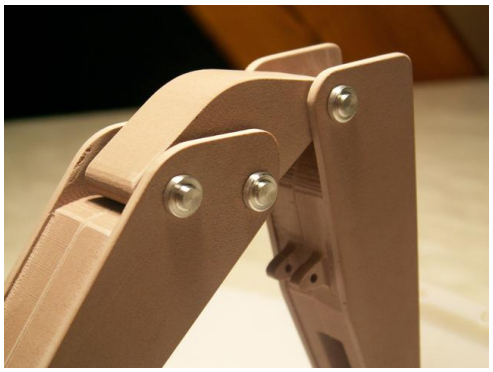
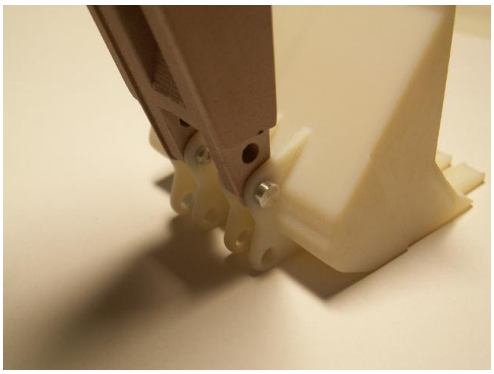
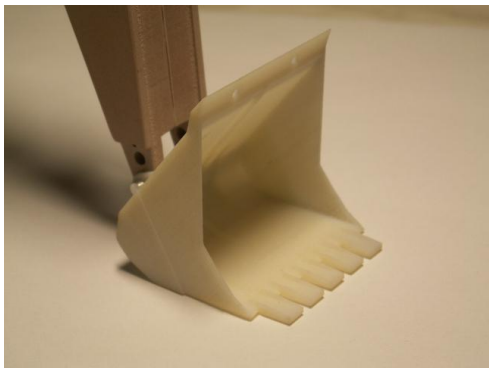
Z vlastní zkušenosti vím, že nejlépe se učím, pokud k tomu mám motivaci. Proto jsem do výukové pomůcky zařadil ukázku modelu celého ramene, na které můžeme názorně demonstrovat, k čemu se vlastně učíme modelovat v CAD systémech. Záměrně byl model nechán udělat dvěma možnostmi vytváření prototypů, aby si studenti mohli co nejširší představu o tom, co se dá z 3D dat vytvořit.

Samozřejmě pokud by nastal ideální případ a škola by vlastnila 3D tiskárnu či NC stroj s gravírovací frézkou, mohl by si každý student nechat vytvořit vlastní model.

-

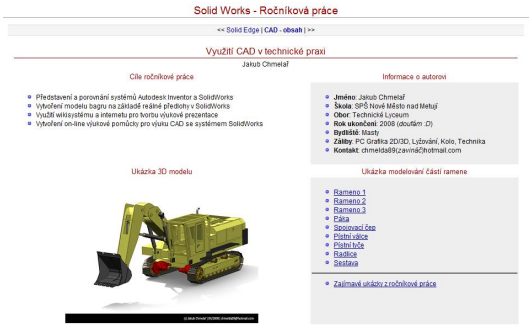
Fotografie modelu ramena vytvořeného z 3D modelu podle pomůcky

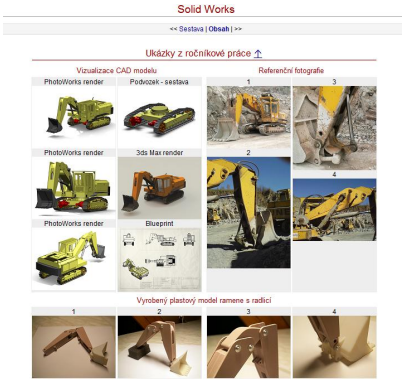



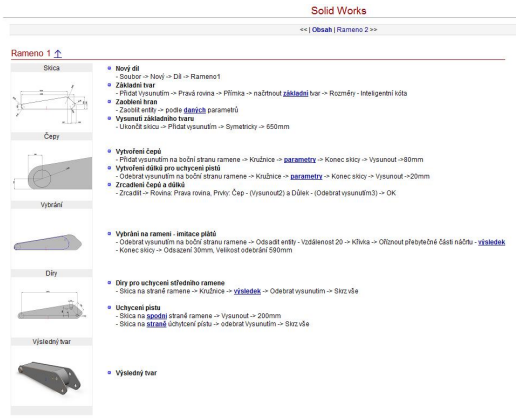




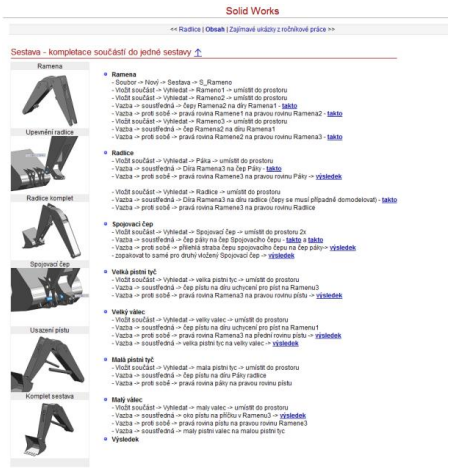
-

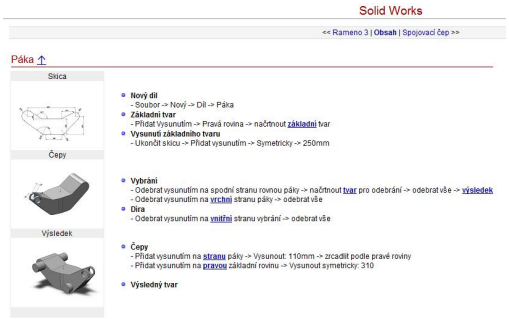
Ukázky z výukové pomůcky

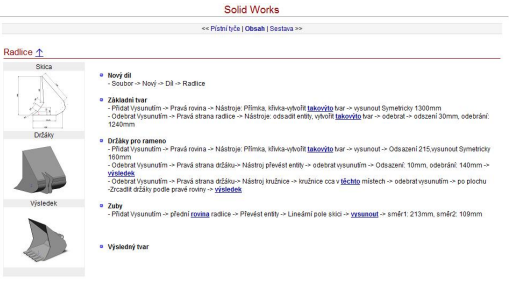












Závěr

V průběhu psaní této práce jsem měl možnost poznat špičkový CAD systém SolidWorks, ve verzi od 2006 až po verzi 2008, a porovnat jej s konkurenčním systémem Inventor v. 2008. Musím říci, že oba systémy mě velmi oslovily ať už svými funkcemi či uživatelským prostředím. Mnoho lidí se mě zřejmě bude ptát, který systém je podle mě tedy lepší. Osobně musím říci, že jsem u obou z nich našel funkce a možnosti nastavení, které jsem u druhého postrádal apod., takže nemohu, s určitostí říci tento je nejlepší. Pokud by se mě však někdo zeptal, který z těchto systémů bych si koupil, odpověděl bych mu že SolidWorks. Velmi mě na něm zaujali možné simulace a analýzy výrobků, které je možno provádět přímo v prostředí programu a také celkově velmi uživatelsky přívětivé prostředí.

Před začátkem psaní této práce jsem si vytyčil nějaké cíle, kterých jsem chtěl dosáhnout. Především to bylo vytvoření modelu pásového bagru podle reálné předlohy. Souviselo s tím spoustu problémů, které jsem musel řešit. Musím zdůraznit, že na začátku psaní práce jsem s programem SolidWorks neměl vůbec žádné zkušenosti, takže jsem musel, zvládnout jeho ovládání. Další výzvou pro mě bylo samotné modelování sestavy bagru, kde jsem si musel většinu jeho rozměrů dopočítávat z dostupných fotek, kterých jsem si opatřil a seznámit se také s konstrukcí bagru.

V další části mé ročníkové jsem vytvářel výukovou pomůcku pro školní web info.spsnome.cz.

Měl jsem také možnost zúčastnit několika seminářů a školení zaměřených na 3D modelování. Měl jsem možnost shlédnout oficiální prezentaci nové verze programu SolidWorks 2008, také jsem se zúčastnil Autodesk fóra 2007 v Praze, kde byla české veřejnosti představena produktová řada 2008 společnosti Autodesk. Při těchto setkání jsem poznal spoustu nových specialistů a dozvěděl jsem se mnoho užitečných informací.

Určitě jsem velmi rád, že jsem mohl napsat tuto práci, protože jsem si zkusil jaké to je, sbírat informace a řešit problémy pro nějaký projekt, o kterém jsem na začátku nevěděl téměř nic. Jsem si jistým, že na tuto práci budu moci navázat při studiu na vysoké škole či případně alespoň využít získané zkušenosti.

-

Poděkování

Na závěr bych chtěl poděkovat několika lidem, kteří mi s prací pomáhali a se kterými jsem mohl konzultovat všechny své nápady a řešit problémy.

První je především můj hlavní konzultant, vedoucí projektu, učitel, Ing. Josef Nymš, který mi po celou dobu projektu radil s otázkami, jakým směrem bych měl směřovat svojí práci.

Další mé poděkování patří společnosti TOP-TECH s.r.o. a především paní Radce Rybkové, která mi zařídila vypůjčení programu SolidWorks a pomáhala mi při řešení problémů týkajících se ovládání a používání tohoto programu.

Nesmím také opomenout zmínit společnost TECNOTRADE obráběcí stroje s.r.o. která mi prostřednictvím pana Obůrka zhotovila model radlice metodou 3D tisku - RAPID PROTOTYPING.

Samozřejmě, poděkování patří celé mé rodině, která mě v psaní práce po celou dobu podporovala a především otci, který mě pomáhal při výběru tématu práce a se kterým jsem vzhledem k jeho dlouholeté technické praxi konzultoval technické záležitosti při modelování bagru.

-

Použité informační zdroje

Časopisy Computer Design

Informační a propagační materiály společnosti Dassault Systems na program SolidWorks

Informační a propagační materiály společnosti Autodesk na program Inventor

Prezentační materiály společnosti TOP-TECH spol. s.r.o. (PowerPointové prezentace)

Použité obrázky a informace, kromě vlastní tvorby, byly použity z webů:

www.solidworks.com

www.autodeskclub.cz

www.xanadu.cz

www.manufacturing.cadalyst.com

www.students.autodesk.com

www.autodesk.com/inventor

www.pmwiki.org

-

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

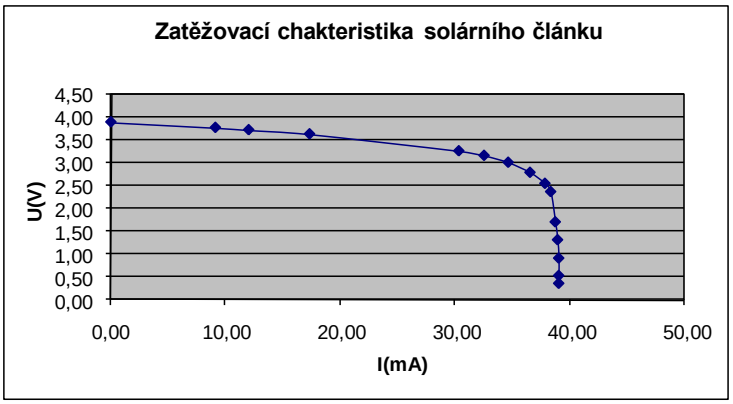
Solární vozítka a vozítko poháněné vodíkem.

Michal Benda, Jan Cipro, Jakub Klíma, Miroslav Suk,
David Kohout, Jaroslav Černý

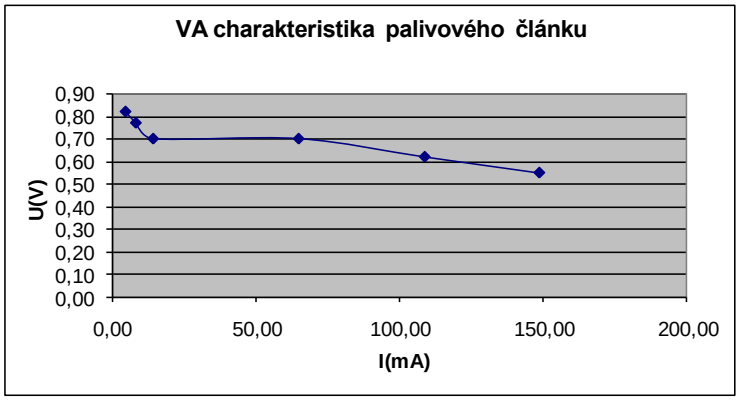
Střední škola technická hl.m.Prahy, Radlická 115, 158 00 Praha 5

Naši studenti se pravidelně zúčastňují soutěže Napájení Sluncem se solárními vozítky. Jeden typ vozítka je přímo poháněn energií, získanou ze solárních panelů, motorek je se solárními panely spojen přímo. To vyžaduje vyšší intenzitu osvětlení a při slabším osvětlení již je energie nedostatečná pro trvalý chod motorku.

Druhý typ vozítka je vybaven jednoduchou elektronickou regulací. Energie se akumuluje v kondenzátoru a motorek je zapnut až při určité úrovni napětí a při poklesu napětí opět odpojen. Na obrázku je uvedena naměřená zatěžovací charakteristika solárního článku.



V letošním roce studenti realizovali stavebnici vozítka s palivovým článkem. Vodík pro palivový článek se získává rozkladem vody. Palivový článek dodává napětí cca 500mV a proud maximálně 500mA. Pro napájení motorku je vozítko vybaveno elektronickým měničem. Vzhledem k nízkému napětí palivového článku, musí být pro napájení elektroniky použit ještě pomocný měnič. Voltampérová charakteristika palivového článku je uvedena na obrázku.



Středoškolská technika 2008

Setkání a prezentace prací středoškolských studentů na ČVUT

SLUNEČNÍ HODINY – OD TEORIE K PRAXI

Klára Kováčová, Iveta Vildová

Gymnázium Pierra de Coubertina
Náměstí Františka Křižíka 860, Tábor 39001

Obsah:

Úvod.....
Historie slunečních hodin.....
Obrazová příloha k historii.....
Princip slunečních hodin.....
Obrazová příloha k principu slunečních hodin.....
Rovníkové hodiny.....
Polární hodiny.....
Vícenásobné sluneční hodiny.....
Obrazová příloha vícenásobných hodin.....
Jak vytvořit sluneční hodiny.....
Jak poznat gnómonicky chybné hodiny?.....
„Nej“ slunečních hodin v ČR.....
Vybrané lokality slunečních hodin Táborska.....
Vila Třeboňka – Budějovická 987, Tábor.....
Obrazová příloha k Vile Třeboňka.....
Kostel sv. Václava - Planá nad Lužnicí.....
Obrazová příloha Kostel sv. Václava .....
Náměstí Tomáše Bati – Sezimovo Ústí.....
Obrazová příloha k náměstí Tomáše Bati.....
Větrovy 38.....
Obrazová příloha Větrovy 38.....
Návrhy modelů slunečních hodin.....
Model č.1.....
Model č.2.....
Model č.3.....
Model č.4.....
Model č.5.....
Závěr.....
Zdroje.....

Úvod

Sluneční hodiny byly tématem naší seminární práce z fyziky. Rozdělili jsme se do dvojic až trojic a každá si vybrala jedny hodiny na Táborsku. Tyto lokality jsme navštívili, vyhledali co nejvíce informací a zpracovali.

Dále naši třídu čekalo vypracování několika modelů slunečních hodin, které by bylo možné umístit v areálu školy, což se nám (doufáme úspěšně) podařilo. Vyvrcholením naší práce by mělo být reálné zhotovení slunečních hodin v naší škole.

Této práci jsme všichni věnovali obrovské úsilí, a proto nás napadlo, že by nebylo špatné s tímto bezesporu zajímavým námětem seznámit širší veřejnost. Již nám vyšel článek v regionálních novinách Táborsko a nyní jsme se rozhodli zkusit zúročit naší práci v účasti v Středoškolské odborné činnosti.

Historie slunečních hodin

Pro měření času člověk nejprve využíval to, co měl kolem sebe — dalo by se říci, že k sledování toku času spoutával čtyři živly přírody (vodu – kapalinové hodiny, oheň – ohňové knotové či doutnákové hodiny, zemi – pískové přesýpací hodiny, vzduch – resp. promítnutí pohybu nebeských těles při sestavování slunečních hodin). Tyto nejstarší časoměrné přístroje souhrnně označujeme jako elementární. Počátky jejich používání můžeme na základě konkrétních archeologických nálezů a dochovaných písemných zpráv zařadit do doby přibližně 2000 let př. n. l. S prvními zdokonaleními mechanických kolečkových hodin, asi ve 13. století n. l., končí éra těchto elementárních přístrojů, i když se s jejich používáním setkáváme i v dnešní době.

Objev slunečních hodin spočívá v uvědomění souvislosti mezi délkou či směrem vrženého stínu a polohou Slunce na obloze. Nejstarším typem slunečních hodin byl gnómon (svislý obelisk) se stupnicí na zemi (obr. 1). Délka stínu určovala denní hodinu. Hodinovou stupnicí tedy mohly být kružnice soustředné okolo paty gnómonu.

S využíváním gnómonu k měření času, ale i ke kultovním účelům, se setkáváme již ve starověkém Egyptě, a to přibližně v 15. století př. n. l. (viz obr. 2). Pro měření času slunečními hodinami tu byly zvláště příznivé klimatické podmínky. Obelisky, stojící zpravidla před vchody chrámů, sloužily Egypťanům především k uctívání boha Slunce; úlohu gnómonu však mohly plnit i prosté pylony na veřejných prostranstvích.

Zkoumání pohybu nebeských těles, spojené s náboženskými obřady, provázelo kultury civilizací po celém světě — Egypťany, Babyloňany, Číňany, Maye, Aztéky, Inky a další. Hrálo zásadní roli při tvorbě kalendářů. Astronomické přístroje používané pro tyto účely měly různý tvar. S uskupeními kamenných monolitů, které se vyznačují astronomickou orientací, se setkáváme i na území dnešní Evropy — známé je Stonehenge ve Velké Británii (obr.3).

Gnómony nebyly zcela přesnými časoměrnými pomůckami, protože délky jejich stínu závisejí i na roční době. Ale zejména v oblastech bližších k rovníku dovolovaly měřit čas s přijatelnou chybou. Od Egypťanů se znalost využívání slunečních hodin přenesla do starověkého Řecka. Původní ostrý hrot gnómonu nahradili Řekové malým kruhovým otvorem, tzv. slunečním okem, které vrhalo na stupnici malý světelný bod.

Vývojově mladším typem slunečních hodin byly horizontální duté polokulové hodiny – skafé. Nejstarší nálezy pocházejí ze starověkého Řecka a jsou datovány do 7. století př. n. l. Vnitřní plocha polokoule je průmětem nebeské klenby, s rovníkovou čarou, s dvěma čárami slunovratů a s dvanáctihodinovou časovou stupnicí (obr. 4). Hrot ukazatele musí být ve středu skafé.

Řekové znali ještě vertikální sluneční hodiny, které umisťovali na veřejných budovách. Jako příklad lze uvést hodiny na svislých stěnách Věže větrů v Athénách, pocházející z prvního století př. n. l. Jde o osmibokou budovu přesně orientovanou do základních světových stran, kde na každé stěně byly umístěny sluneční hodiny.

Epochu moderních slunečních hodin otevřel významný objev spočívající v nastavení stínového ukazatele do směru zemské osy (datování tohoto objevu je však nejisté; v Evropě existují písemné zprávy o šikmém ukazateli až ze 14. století n. l.). Touto jednoduchou úpravou se dosáhlo stavu, kdy se stín ukazatele, nyní nazývaného polos, rovnoměrně otáčel kolem ukazatele každou hodinu o 15°. Na číselníku se pak začal vyznačovat čas hodinami, které během celého roku trvaly stejně dlouhou dobu.

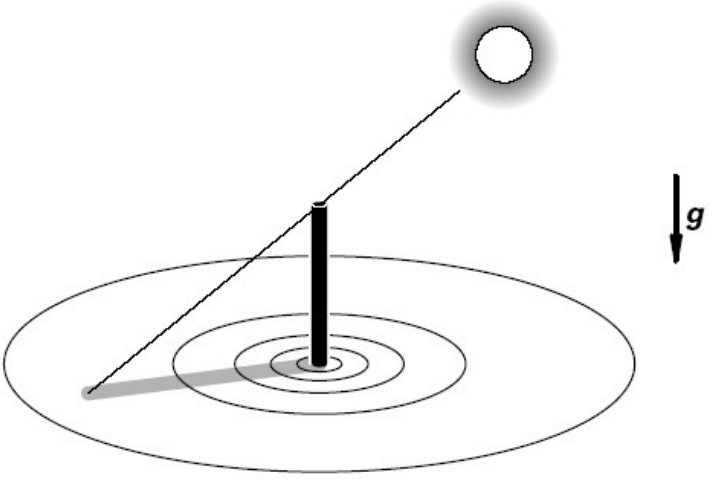
Šíření slunečních hodin mezi různými civilizacemi probíhalo mnohem pomaleji než u jiných vynálezů. Zřejmě to je způsobeno tím, že sluneční hodiny není možné jednoduše přenést z místa na místo (vzhledem k závislosti pohybu Slunce po obloze na zeměpisné šířce),

ale je nutné přenést jejich princip a pochopení.

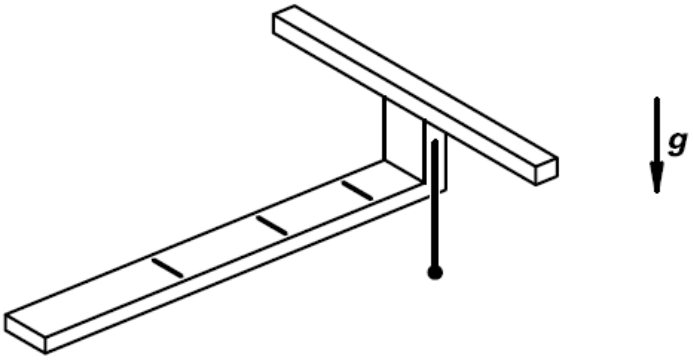
Z pohledu historického vývoje slunečních hodin je třeba zmínit i přenosné sluneční hodiny. Jejich počátek spadá do starověku – prvním typem byly hodiny sestrojené na principu gnómonu, následovaly hodiny využívající výše popsaný princip skafé. Z dalších jmenujme např. rovníkové, sloupkové, horizontální, poutnické, „selské“, Lambertovy, prstencové, diptychové nebo analematické. Důležitým vylepšením přenosných hodin bylo jejich spojení s kompasem.

Během tisíciletého vývoje se objevilo mnoho velmi rozmanitých tvarů a typů slunečních hodin. Dnes jsou zdaleka nejrozšířenější nástěnné hodiny s polosem.

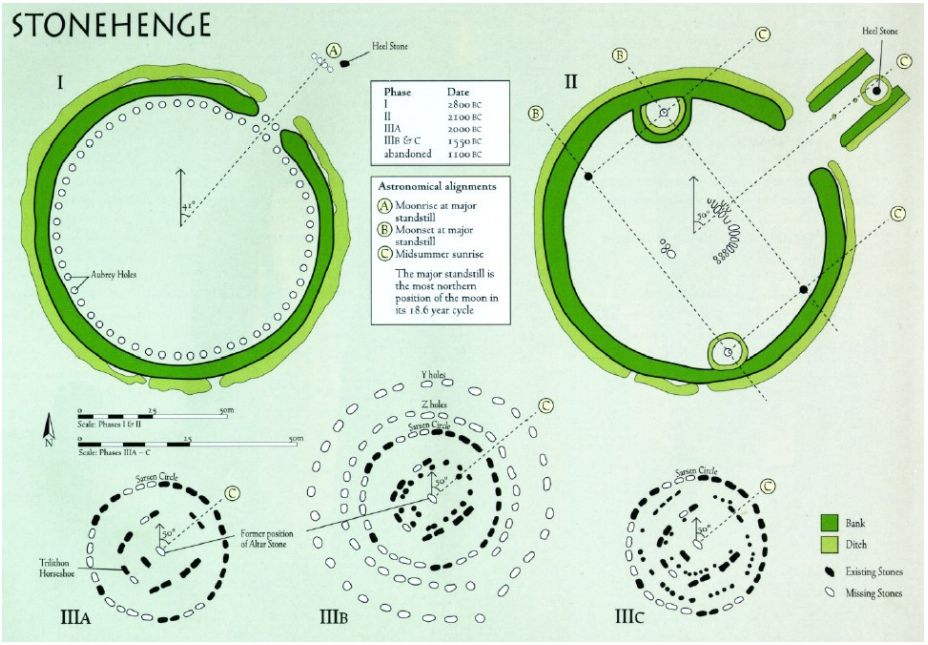
Obrazová příloha k historii



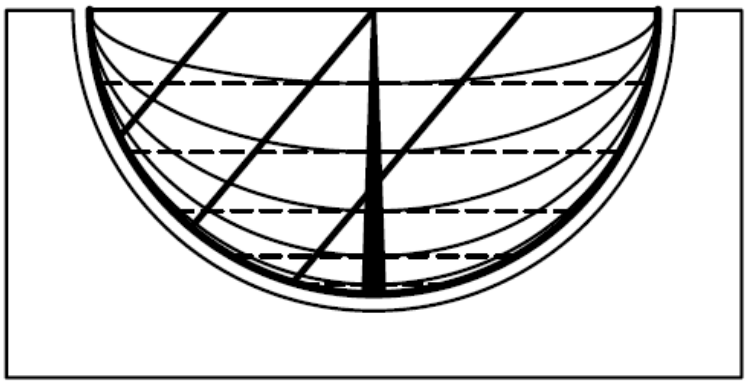
Obr. 1 – Princip gnómonu, tedy měření času podle délky vrženého stínu. Ukazatelem slunečních hodin je zde svislá tyč, číselníkem je vodorovný povrch země, s nakreslenými soustřednými kružnicemi.



Obr. 2 – Staroegyptské sluneční hodiny ve tvaru „T“, které také pracují s délkou stínu. Byly vybaveny olovnicí pro ustavení do vodorovné polohy. Dopoledne byly hodiny orientovány na východ, odpoledne je bylo nutné otočit na západ. Pak stín horního hranolu ukazoval na stupnici denní dobu.



Obr. 3 – Sonehenge založil lid s rýhovanou keramikou okolo roku 2800 př. n. l. jako chrám zasvěcený Měsíci. Jeho hlavní osa s astronomickým azimutem 221° ukazovala na nejsevernější východ Měsíce během cyklu dlouhého 18,6 roku. Kolem roku 2100 př. n. l. lid zvoncových pohárů přeměnil Stonehenge na sluneční a měsíční chrám. Dva nové hlavní směry stavby byly nejsevernější západ Měsíce a (nejsevernější) východ Slunce za letního slunovratu s azimuty 140° a 230°. Na zeměpisné šířce Stonehenge svírají tyto směry pravý úhel, což bylo zřejmě pro stavitele důležité. Mezi 2000 až 1550 lety př. n. l. došlo k dalšímu rozvoji Stonehenge, a to vztyčováním obrovských opracovaných monolitů, které jsou dnes pro stavbu charakteristické. Nejprve vznikl Sarsenský kruh, Trilithonová podkova a pak menší oblouky; předchozí astronomické orientace zůstaly zachovány. Desítky kamenných bloků vysokých až 5 m a vážících až 40 tun byly dopravovány 360 km dlouhou cestou po souši a po vodě z lomu v Mynydd Preseli v severním Pembrokeshire. Okolo roku 1100 př. n. l. bylo Stonehenge náhle opuštěno.



Obr. 4 – Skafé (rovinný řez středem polokoule) s vyznačeným rovníkem a obratníky (čáry tlusté), hodinovými kružnicemi (tenké) a výškovými kružnicemi (čárkované). Takové uspořádání dělilo „bílý“ den na 12 nestejně dlouhých (temporálních) hodin.



Obr. 5 – Přenosné sluneční hodiny diptychového typu, vyrobené v norimberské dílně Conrada Karnera v roce 1622. Jsou vybaveny kompasem, na němž je vyznačena tehdy platná magnetická deklinace 5°, a pěti hodinovými stupnicemi se škálami (4–12–8) příslušnými pěti různým hodnotám zeměpisné šířky (42°, 45°, 48°, 51°, 54°). Na vodorovné destičce jsou ještě malé sluneční hodiny ukazující babylonský a italský čas. Na svislé destičce nahoře vidíme číselník s temporálními hodinami, sedmi datovými křivkami, kterým přísluší symboly znamení zvěrokruhu a délky trvání dne a noci. Níže se nachází pět otvorů pod sebou; do jednoho z nich, podle zeměpisné šířky stanoviště, je potřeba zasunout šikmý vláknový ukazatel. Hodiny jsou doplněny tabulkou zeměpisných šířek významných evropských měst. Na vnější straně bychom mohli vidět ještě větrnou růžici s 32 směry a kalendářní data juliánského i řehořského kalendáře.



Obr. 6 – Kopie prstencových výškových slunečních hodin pruských benediktinů z roku 1721. Před čtením času je potřeba otvor posunout podle aktuálního měsíce (viz zkratky J, F, M, A, M, J a J, A, S, O, N, D), a pak volně visící hodiny otočit směrem k Slunci (vnitřní strana prstence musí být přitom celá ve stínu).

Princip slunečních hodin

Ke snazšímu porozumění činnosti slunečních hodin si připomeneme základní poznatky z astronomie a zeměpisu. Země vykonává několik pohybů, z nichž nejpatrnější jsou ve svých důsledcích dva: Země se jednou za den otočí kolem své osy a jednou za rok oběhne Slunce (obr. 7).

Rotace Země kolem osy je příčinou zdánlivého pohybu Slunce od východu k západu. Roční oběh kolem Slunce má za následek, že zdánlivá denní dráha Slunce po obloze se během roku mění (obr. 8, 9).

Dráha Země není kruhová, ale eliptická; Slunce leží v jednom ohnisku elipsy (tj. 1. Keplerův zákon). Vzdálenost Země od Slunce se tedy v průběhu roku mění. Bod na dráze, ve kterém je Země nejblíže, nazýváme přísluní (perihel); je-li Země nejdál, mluvíme o odsluní (afelu). Pohyb Země kolem Slunce je nerovnoměrný, jak o tom svědčí 2. Keplerův zákon. Pro pozorovatele na Zemi se Slunce pohybuje vzhledem k hvězdné obloze nejrychleji, je-li v perihelu, a nejpomaleji, je-li v afelu. Časový interval od jarní (21. 3.) do podzimní rovnodennosti (23. 9.) tak trvá 186 dní, kdežto druhá „polovina“ roku jen 179 dní (tj. rozdíl celého týdne).

Zemská osa není k rovině zemské dráhy kolmá, ale je od kolmice odkloněna přibližně o 23,5°.

Obě uvedené skutečnosti mají za následek, že zdánlivý pohyb Slunce po obloze je nerovnoměrný a úsek pro 1 hodinu na slunečních hodinách mění v průběhu roku svoji délku. Hodnota rozdílu mezi časem udávaným slunečními hodinami (pravý sluneční čas) a časem rovnoměrně plynoucím (střední sluneční čas) se nazývá časová rovnice (obr. 10).

Čtyřikrát do roka (zhruba 15. dubna, 14. června, 31. srpna a 24. prosince) je časová rovnice rovna nule. Tehdy jsou časové údaje slunečních hodin a rovnoměrný čas našich hodinek shodné. Kladné hodnoty (zhruba 15. 4. až 14. 6. a 31. 8. až 24. 12.) znamenají, že čas měřený slunečními hodinami předbíhá čas našich hodinek, a naopak záporné hodnoty (14. 6. až 31. 8. a 24. 12. až 15. 4.) znamenají, že se údaj slunečních hodin zpožďuje.

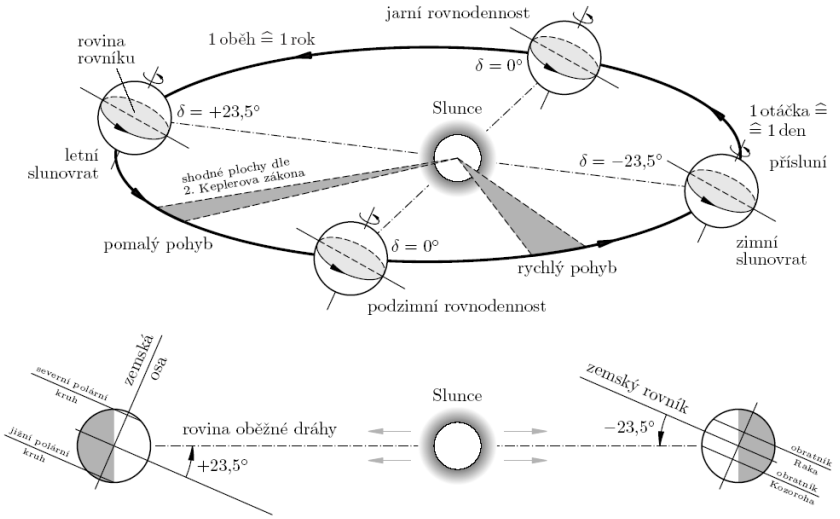
Ještě existuje jeden důvod, proč se časový údaj čtený ze slunečních hodin liší od údaje přesných mechanických nebo atomových hodin (občanského času). Vzniká tím, že sluneční hodiny zpravidla udávají pravý sluneční čas místního poledníku. Občanský čas naopak vychází z pásmového času, tj. času odpovídajícího střednímu poledníku daného pásma (u nás je to poledník 15° východně od Greenwiche). Jednotlivé okamžiky (východ, pravé sluneční poledne či západ Slunce) na stanovištích východně od tohoto poledníku nastávají dříve (např. v Českém Těšíně o 13 minut), na stanovištích západně od něj nastávají později (v Aši o 11 minut).

Prakticky všechny hodiny na pevných stanovištích u nás pracují na principu hodinového úhlu.

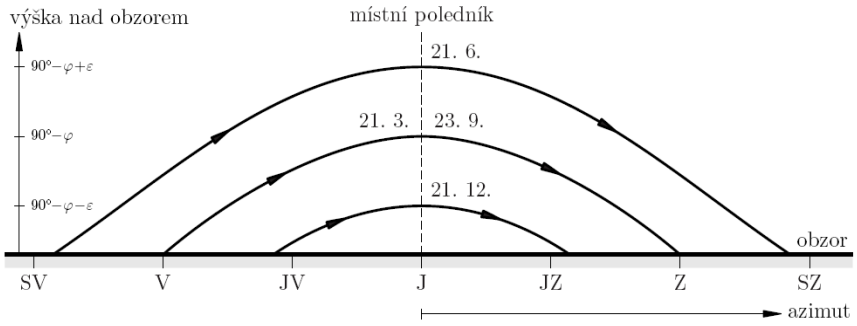
Střední slunce zdánlivě oběhne zeměkouli rovnoměrně za jeden sluneční den, tedy 24 hodin (jedna otočka Země o 360° vzhledem ke hvězdám trvá kratší dobu — 1 hvězdný den = 23 h 56 min 4 s). Světovou sféru lze pomyslně rozdělit na 24 stejných dílu hodinovými hlavními kružnicemi, které procházejí přes póly. Protože plnému úhlu 360° odpovídá 24 hodin, připadá na 1 hodinu úhel 15°.

Je-li stínový ukazatel slunečních hodin rovnoběžný se zemskou osou, otáčí se slunce zdánlivě okolo ukazatele vždy o 15° za jednu hodinu.

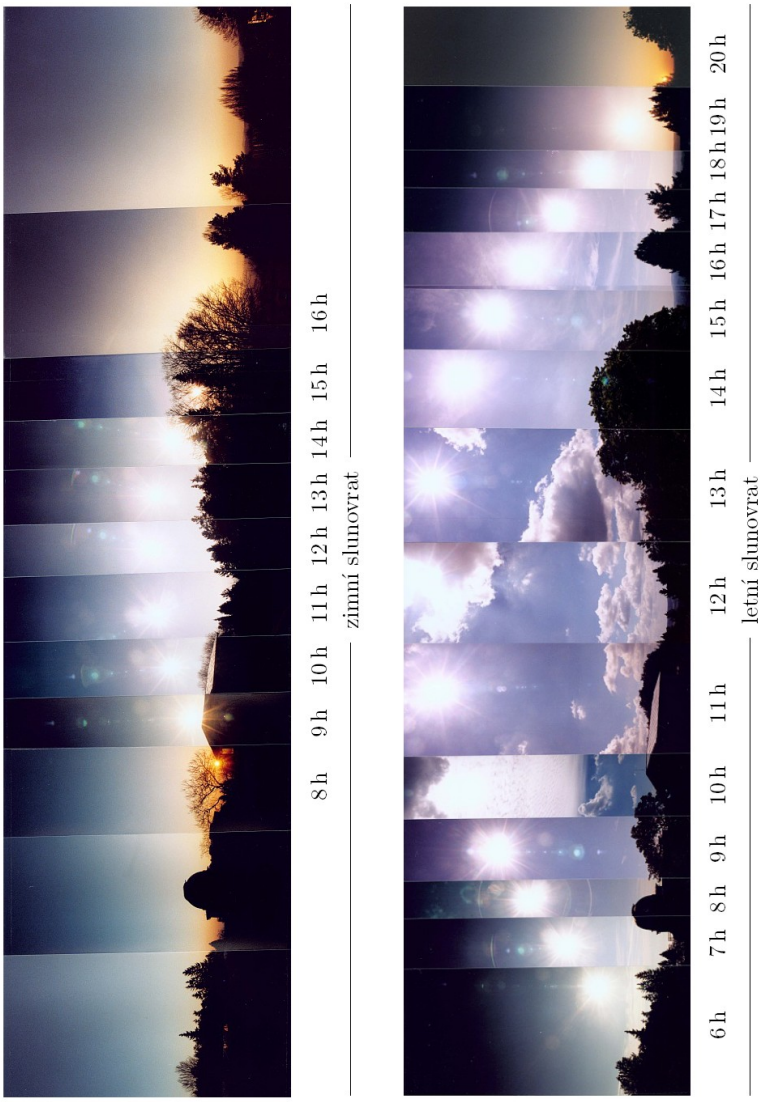
Obrazová příloha k principu slunečních hodin



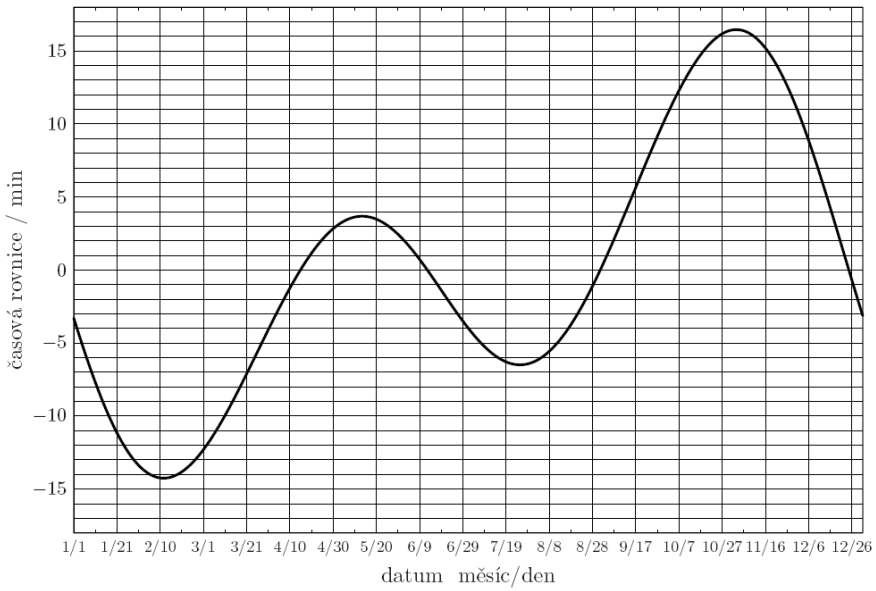
Obr. 7 – Dva základní pohyby Země: otáčení okolo osy a oběh kolem Slunce.



Obr. 8 – Denní dráha Slunce po obloze v různých ročních obdobích. Pohled k jihu na severní polokouli. Úhel φ označuje zeměpisnou šířku stanoviště, ε = 23,5° je úhel sevřený rovinou rovníku a rovinou oběžné dráhy Země.



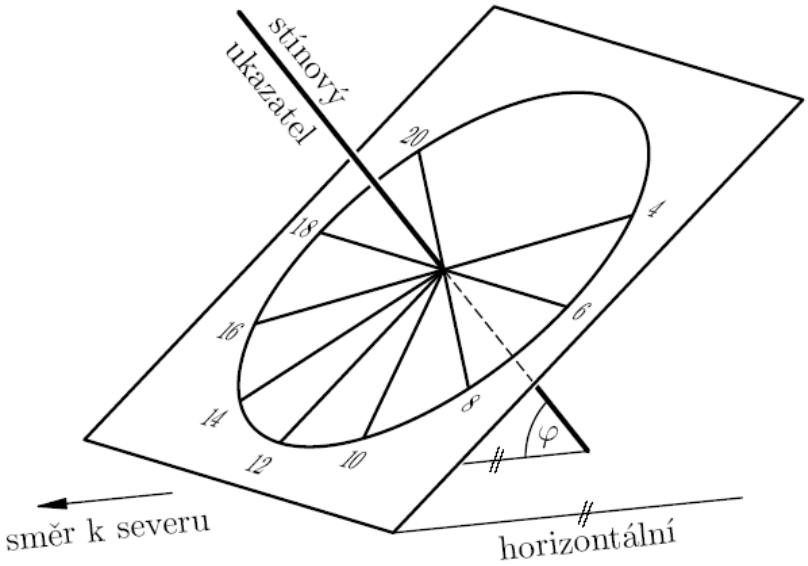
Obr. 9 – Složené fotografie pohybu Slunce během dne: (a) 9. prosince 2001, při deklinaci Slunce δ = −22° 50‘; (b) 21. až 27. června 2001, δ = +23° 20‘. Použitý přístroj Nikon F–801s s objektivem Nikkor 20 mm, f/16, materiál Konica VX 200 ASA (pro zimní snímek) a Fuji 100 ASA (pro letní), stanoviště Ondřejov, Astronomický ústav AV ČR. Foto Aleš Kolář (2001).



Obr. 10 – Časová rovnice v roce 2000. Hodnotu E časové rovnice pro příslušné datum musíme odečíst od času čteného na slunečních hodinách, abychom z místního pravého slunečního času zjistili místní střední sluneční čas.

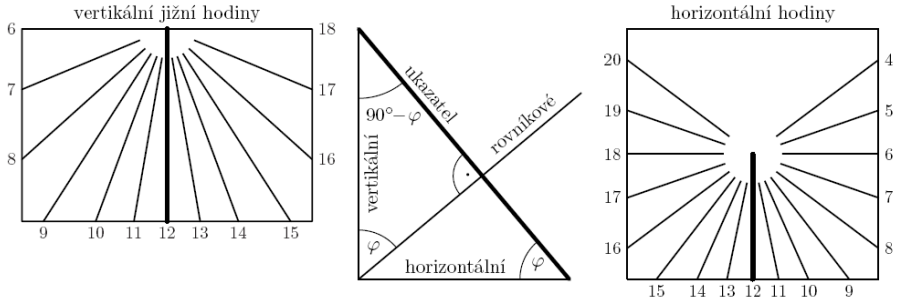
Rovníkové hodiny

Nejlépe je princip patrný na rovníkových hodinách. Svůj název dostaly proto, že rovina číselníku je rovnoběžná s rovinou rovníku. Stínový ukazatel rovnoběžný se zemskou osou je k takovému číselníku kolmý. Hodinové čáry jsou rozmístěny po uvedených 15°. Toto rozmístění je vidět na obr. 11.



Obr. 11 – Rovníkové sluneční hodiny. Rovina číselníku (rovníku) je k vodorovné rovině skloněna pod úhlem 90° φ, kde φ označuje zeměpisnou šířku stanoviště. Hodinové úsečky jsou zde značeny po 2 hodinách (30°).

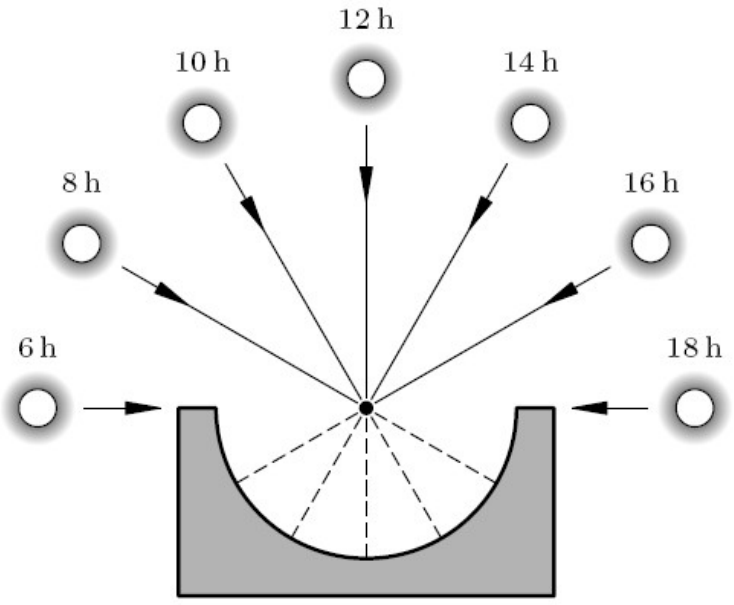
Číselníky svislých nebo vodorovných hodin získáme jako kolmé průměty číselníku rovníkových hodin. Vztah mezi nimi naznačuje obr. 13.



Obr. 13 – Vztah rovníkových, vertikálních a horizontálních hodin; prostřední obrázek je pohled „z boku“, tj. od západu.

Polární hodiny

V případě polárních hodin jsou hodinové přímky rovnoběžné se zemskou osou, a tedy i se sebou navzájem. Je-li číselníkem vnitrní plocha válce (v případě polárních válcových hodin, obr. 12), jsou na něm rozmístěny opět rovnoměrně po 15°.



Obr. 12 – Polární válcové sluneční hodiny. Pohled je ve směru ukazatele, tj. ve směru zemské osy, od severního pólu, kolmo na rovinu rovníku.

Vícenásobné sluneční hodiny

Vícenásobné sluneční hodiny vzniknou bud vícenásobným použitím hodin stejného typu, nebo kombinacemi hodin různého typu (vodorovných, svislých, polárních, rovníkových) na jednom stanovišti nebo tělese.

Číselník mohou mít na různých druzích ploch (rovinných, válcových, kulových, obecných). Mohou být na vypouklých plochách nebo v dutinách. Jejich vzájemnou kombinací lze vytvořit vícenásobných slunečních hodin nesmírné množství.

Vzhledem k tomu, že na pevných stanovištích se nejčastěji vyskytují svislé hodiny, lze vícenásobné sluneční hodiny spatřit nejčastěji na zdech budov, zpravidla ve dvojicích. Pak bývají označovány jako párové nebo rohové. Můžeme je nalézt na rozích budov nebo na vnitřních zdech (dvorany, atria). Podle toho, jak byla budova orientována při stavbě, mohou být orientovány do základních světových stran nebo vůči nim natočeny.

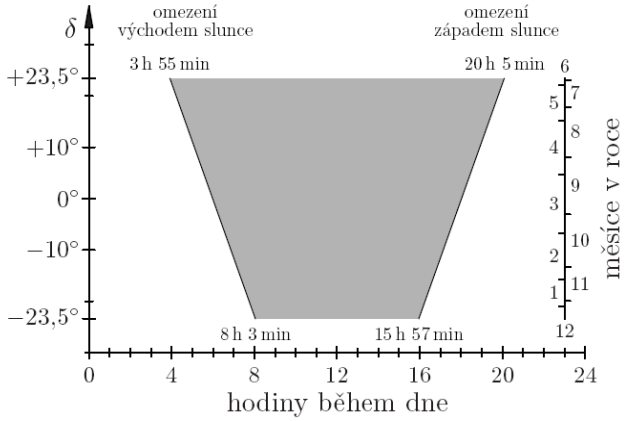
Činnost vodorovných slunečních hodin na volném prostranství není ničím omezena. Jak je patrné z obr. 14, rozsah jejich pracovní činnosti během dne je vymezen okamžiky východu (levá mezní čára) a západu slunce (pravá mez).

U svislých hodin tomu tak není. Hodiny umístěné na zdi fungují jen tehdy, svítí-li slunce na stěnu. Například jižní zeď je osvětlena, svítí-li slunce z východního až západního směru. V pravé poledne (místního času) svítí slunce vždy z jižního směru. Pro ostatní hodiny se úhel, měřený ve vodorovné rovině, pod kterým slunce svítí na zeď (azimut slunce), mění v závislosti na roční době (deklinaci slunce). Proto se rozsah činnosti nástěnných hodin mění v závislosti na ročním období. Rozsah činnosti hodin během dne (na zeměpisné šířce 50°) v závislosti na orientaci hodin vůči světovým stranám ukazují obr. 15 až 17, alespoň pro základní orientace.

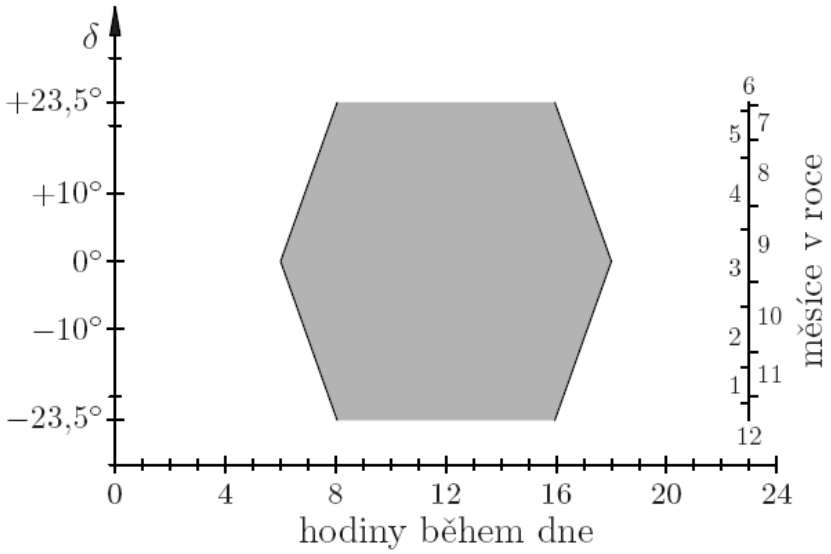
Kromě již známých mezí (východ a západ slunce) jsou zde omezení daná azimutem příslušné stěny. V závislosti na orientaci zdi se poloha mezí pro okamžiky, kdy na stěnu s příslušným azimutem svítí slunce, mění. Jaký by byl rozsah činnosti hodin na zdi orientované k jihozápadu, je patrné z obr. 18.

Jedním z důvodů použití vícenásobných nástěnných hodin je to, že ve slunných dnech „pracují“ vždy alespoň jedny z nich.

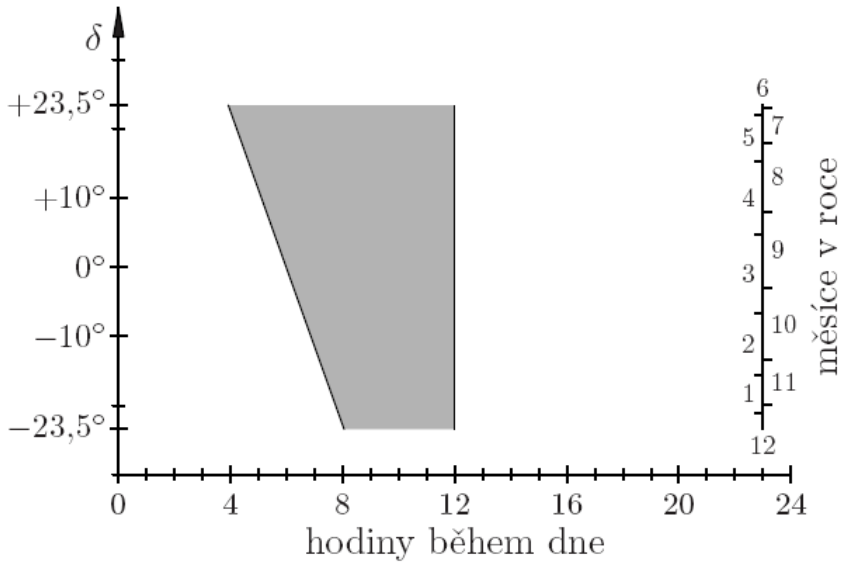
Obrazová příloha vícenásobných hodin



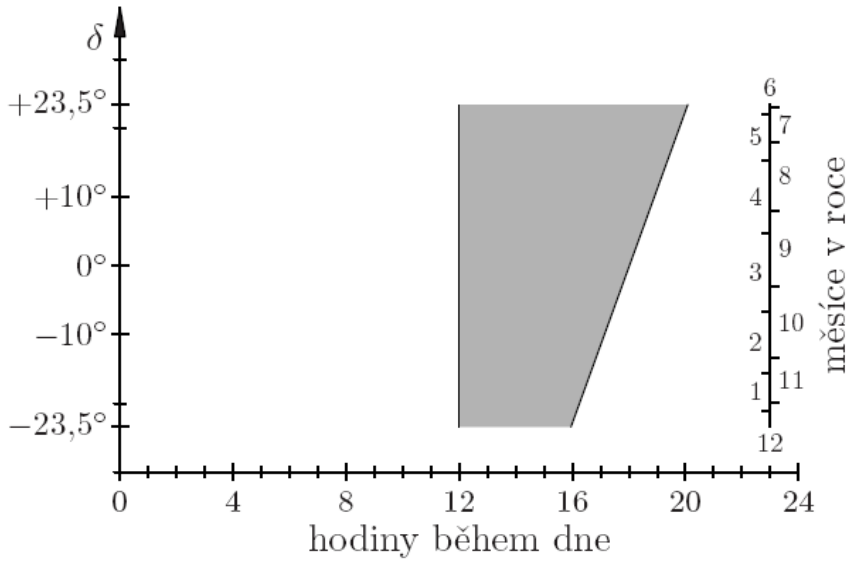
Obr. 14 – Rozsah činnosti vodorovných hodin. Zeměpisná šířka stanoviště je 50° s.z.š. Písmeno δ značí deklinaci slunce.



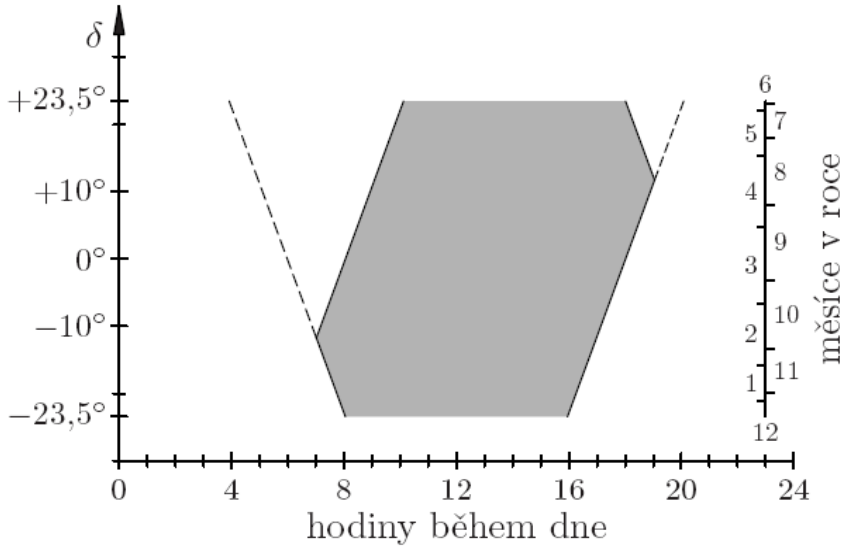
Obr. 15 – Rozsah činnosti jižních svislých hodin.



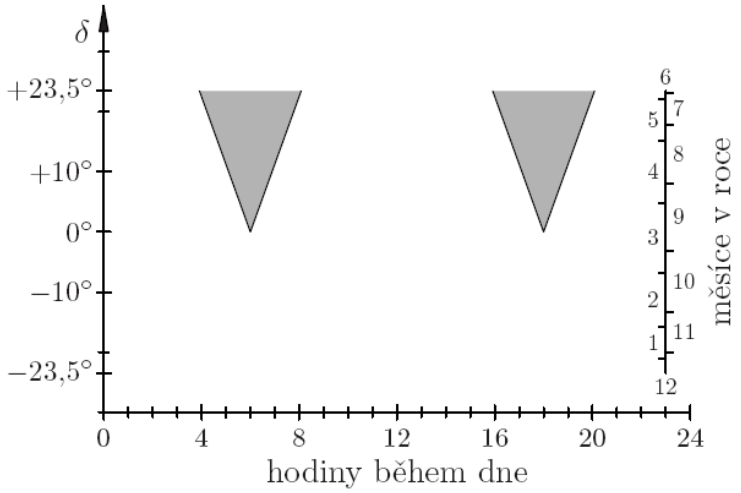
Obr. 16 – Rozsah činnosti východních hodin.



Obr. 17 – Rozsah činnosti západních hodin.



Obr. 18 – Rozsah činnosti jihozápadních hodin.



Obr. 19 – Rozsah denní činnosti severních slunečních hodin. δ označuje deklinaci Slunce. Protože lze severní sluneční hodiny využít během roku jen poměrně zřídka, vyskytují se spíše pro kompletnost tam, kde jsou sluneční hodiny orientované k ostatním světovým stranám.

Jak vytvořit sluneční hodiny

Jednoduché sluneční hodiny můžeme konstruovat během slunečného dne tak, že nejprve správně upevníme ukazatel („tyčku“), a poté v určité okamžiky vyznačíme hodinové rysky podle stínu vrženého ukazatelem.

Začneme tím, že z podrobnější mapy zjistíme zeměpisnou délku λ a šířku φ pro naše stanoviště. Spočteme rozdíl D (v minutách) mezi středoevropským časem (SEČ) a pravým místím slunečním časem (PMSČ) podle vztahu

$$D = \text{SEČ} - \text{PMSČ} = (15 - [\lambda]^\circ) \cdot 4\ \text{min} - E\ ,$$

kde výraz ([λ]° −15) je roven rozdílu zeměpisných délek středoevropského pásmového poledníku a našeho stanoviště, vyjádřený ve stupních a jejich desetinách, a E označuje hodnotu „časové rovnice“, kterou pro příslušné datum zjistíme z tabulky 1.

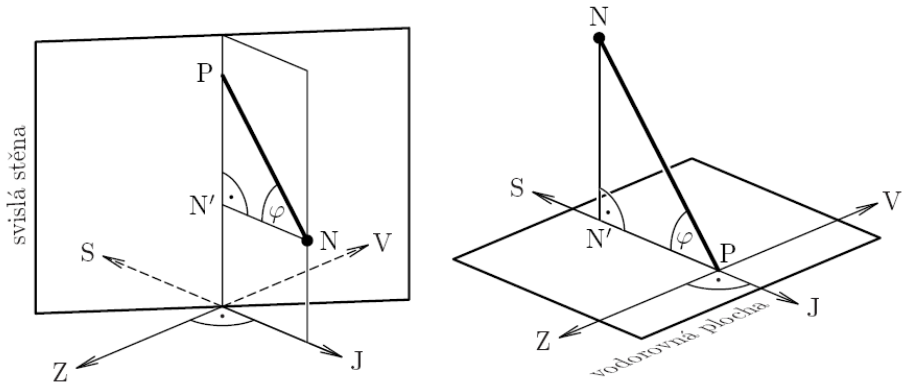
Tab. 1 – Odchylky od pravého slunečního času

Datum	E/min	Datum	E/min
1. leden	−3	1. červenec	−3
15. leden	−9	15. červenec	−6
1. únor	−14	1. srpen	−6
15. únor	−14	15. srpen	−4
1. březen	−13	1. září	0
15. březen	−9	15. září	+5
1. duben	−4	1. říjen	+10
15. duben	0	15. říjen	+14
1. květen	+3	1. listopad	+16
15. květen	+4	15. listopad	+15
1. červen	+2	1. prosinec	+11
15. červen	0	15. prosinec	+5

Hodnoty pro jiná data, než která jsou v tabulce uvedena, vypočítáme interpolací ze dvou nejbližších dat.

Vytyčíme směr sever–jih takto: upravíme si vodorovnou plochu a upevníme do ní svislou tyč. V okamžiku, kdy naše mechanické hodiny ukazují čas 12 h + D, směřuje stín tyče k severu. (Pro vytyčení směru není vhodné použití kompasu.)

Nyní můžeme ustavit samotný ukazatel slunečních hodin — musí ležet ve svislé severo–jižní rovině a zároveň svírat s vodorovnou rovinou úhel φ rovný zeměpisné šířce stanoviště (viz obr. 20).



Obr. 20 – Nákres polohy ukazatele (a) na svislých a (b) na vodorovných slunečních hodinách.

Nakonec vyznačíme hodinové rysky přesně podle stínu vrženého ukazatelem. Rysku příslušnou 13. hodině narýsujeme tehdy, když mechanické hodiny ukazují 13 h+D; stejný posun o D uplatníme i pro ostatní hodinové rysky. Rysky udělejme dostatečně dlouhé – uvažme přitom, že v danou hodinu v zimním období je Slunce mnohem níž nad obzorem než v letním období, a stín ukazatele bude proto různě dlouhý.

Pokud bychom na plochu číselníku zaznamenali trajektorii konce stínu ukazatele, získali bychom datovou křivku pro aktuální datum.

Jak poznat gnómonicky chybné hodiny?

Jak rozpoznáme vertikální nástěnné sluneční hodiny, které nemohou ukazovat celoročně správný čas, protože porušují některé z gnómonických zásad: Začínající konstruktéři nebo restaurátoři hodin by se chyb měli samozřejmě také vyvarovat. Na druhou stranu, rozpoznáme-li gnómonicky chybné hodiny, vůbec to neznamená, že jsou „ošklivé“ – stále mohou sloužit jako ozdoba budovy či prostranství a symbolizovat autorův zájem o přírodní vědy.

Gnómonicky chybné hodiny jsou takové, které vykazují jednu nebo více následujících závad:

1. Šikmý ukazatel není rovnoběžný se zemskou osou.
2. Dvanáctka není na svislici spuštěné z paty šikmého ukazatele (polosu).
3. Pata polosu a rysky pro 6. a 18. hodinu neleží na jedné přímce.
4. Hodinové úsečky se po prodloužení neprotínají v jednom bodě — patě polosu.
5. Datové křivky nemají tvar hyperbol (resp. přímek pro rovnodennosti).

„Nej“ slunečních hodin v ČR

Největší koncentrace hodin na jednom místě:
pražské Klementinum (16, další existovaly v minulosti)
Zlatá Koruna (10)
Kájov (7)
Strahovský klášter (6)
Plasy (6)

Největší počet v kraji:
385 (Jihočeský)

Největší počet v okrese:
115 (Praha-město)
105 (České Budějovice)

Nejstarší:
Telč, rok 1450 (?)

Nejvýše nad zemí umístěné:
Šternberk (35 m)

Nejpřesnější:
polední štěrbinové (vláknové) hodiny v astronomické věži Klementina. Chyba určení pravého poledne je jen několik sekund.

Vybrané lokality slunečních hodin Táborska

Vila Třeboňka – Budějovická 987, Tábor

Zeměpisné souřadnice těchto hodin jsou 14° 40' 34.4" v.z.d., 49° 24' 38.49" s.z.š. Jsou to hodiny svislého typu. Jejich ukazatel je šikmý (polos), typ číselníku poloorlojní. Rozsah těchto hodin je: VIII – XII – II, X – XII – IV.

Majitelem budovy, na které se tyto sluneční hodiny nacházejí, je pan Včelák. Vila Třeboňka prošla v roce 2004 celkovou rekonstrukcí (hodiny jsou staré pouhé tři roky), a právě při té příležitosti se pan Včelák rozhodl umístit na zeď svého domu sluneční hodiny – prý, aby se měly děti, když chodí do školy a ze školy, na co dívat a aby věděly, kolik je hodin.

Konstrukci navrhovala a popsala akademická malířka a grafička – paní Marie Michaela Šechtlová.

Od ní jsme se dozvěděly, že „instalaci“ hodin podle jejích návrhů realizoval pan Slavík. Jsou to její první sluneční hodiny – ale určitě by si chtěla podobný projekt zopakovat.

Dělala také celé barevné řešení fasády, aby s ní hodiny ladily.

Paní Šechtlová musela nejdříve sehnat plány, objekt vyfotografovat, sladit celkovou kompozici – hodiny by měly být tvarově jednoduché, aby se konstruktéři lépe trefili.

Poloha zdi, na které se hodiny nacházejí však bohužel není ideální – nejsou přesně na jih.

Tyč na hodinách se musí natočit podle sklonu zemské osy. Konstruktéři však nebyli úplně přesní, a tak hodiny po většinu roku nejdou správně. Ve skutečnosti ale většina slunečních hodin funguje jen určitou roční dobu.

Obrazová příloha k Vile Třeboňka



Obr. 21 – Celkový pohled na hodiny.



Obr. 22 – Detailní pohled na hodiny

Kostel sv. Václava - Planá nad Lužnicí

Sluneční hodiny jsou umístěny na zadní stěně kostela sv. Václava v Plané nad Lužnicí, která se nachází přibližně 8 kilometrů jihojihovýchodně od Tábora. Zeměpisná poloha hodin je udána zeměpisnou šířkou: 49° 21' 6,11" a zeměpisnou délkou: 14° 42' 14,81".

Původně gotický kostel sv. Václava byl založen roku 1357 a je považován za nejstarší památku v Plané nad Lužnicí. Sluneční hodiny na jeho stěně však byly zřízeny až v roce 1701. Kostel byl v letech 1666 a 1796 přestavován, naposledy byl restaurován po povodních v roce 2002 (sluneční hodiny však zůstaly nedotčeny).

Sluneční hodiny jsou výtvarně zajímavé. Jsou postaveny na fresce zobrazující svatého Václava, který v jedné ruce drží prapor (který zároveň tvoří číselník hodin) a v druhé ruce má štít se znakem orlice. Nad freskou je vyveden latinský nápis: Magne Dux Gechice Inrola Coilice.

Obrazová příloha Kostel sv. Václava



Obr. 23 – Kostel sv. Václava



Obr. 24 – Detail slunečních hodin

Náměstí Tomáše Bati – Sezimovo Ústí

Sluneční hodiny tvoří vpodstatě celé náměstí Tomáše. Jsou to největší sluneční horizontální hodiny v České republice s ciferníkem 20x21m a sklonem 3% k SZ, byly vybudovány v r. 2005. Nacházejí se ve 421,34 m.n.m. a ukazují čas odpovídající poloze místního poledníku 14°41' východní zeměpisné délky a 49°24' severní šířky.

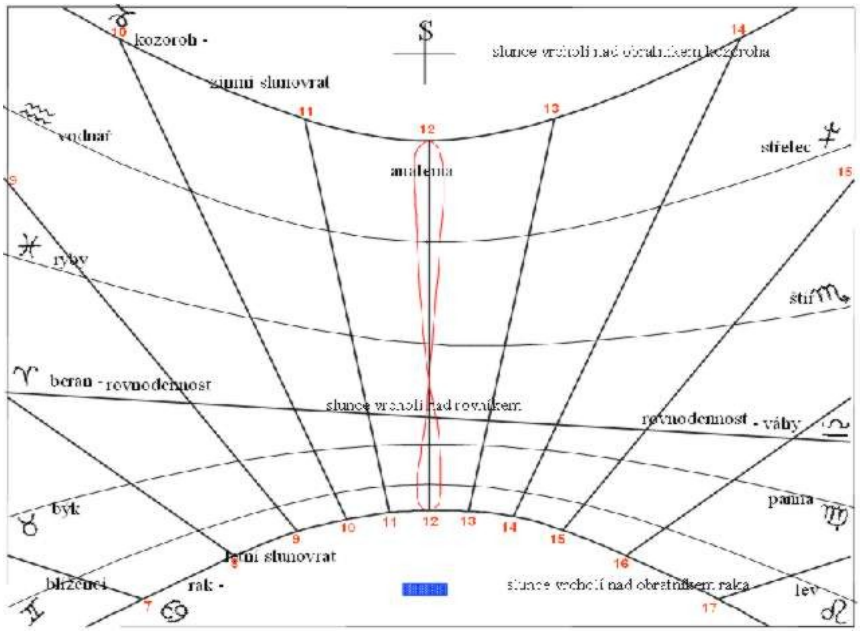
Ukazují pravý sluneční čas, ke kterému je nutno přičíst v letních měsících 1 hodinu s ohledem na letní čas a časovou odchylku způsobenou nerovnoměrností oběžné dráhy Země kolem Slunce, dráhou ekliptiky, tato odchylka není v každou roční dobu shodná a je vyjádřena grafem (obr. 10) a tabulkou (tab. 1), nebo analemou, což je skutečná poloha Slunce, která vyjadřuje rozdíl mezi pravým a středním slunečním časem.

Čas ukazuje stín Žulové koule, kterou drží v levé ruce ve výšce 3,9 m socha Strážce času.

Obrazová příloha k náměstí Tomáše Bati



Obr. 25 – Socha Strážce času při její instalaci



Obr. 26 – Rozkreslená podoba náměstí (modrý obdélníček představuje sochu)

Větrovy 38

Sluneční hodiny se nacházejí na Větrovech (okres Tábor, Větrovy 38) a jsou umístěny na stěně rodinného domu rodiny Hronových (poloha: 14° 38' 29.2" v. d., 49° 23' 22.79" s. š).

Autorem hodin je pan Hron a jeho kamarád pan Rybář, kteří sestrojili hodiny v roce 2001. Hodiny samotné byly slavnostně instalovány až o rok později. Jejich renesanční tvar obdivují lidé z celého okolí a v roce 2003 dokonce vyhrály soutěž o nejlepší amatérsky sestrojené sluneční hodiny pod názvem „Zlatá svislice“.

Hodiny můžeme zařadit mezi tzv. nástěnné s tyčovým ukazatelem. Stěna s hodinami je orientována na jih. Číselník hodin je symetrický podle polední čáry, která leží na svislici. Číselník začíná 8. hodinou raní a končí 4. hodinou odpolední. Polos v podobě železné tyče je umístěn kolmo ke stěně.

Tyto sluneční hodiny pracují na principu hodinového úhlu. Pomineme-li letní čas, je časový údaj jednoznačný bez ohledu na roční dobu.

Obrazová příloha Větrovy 38



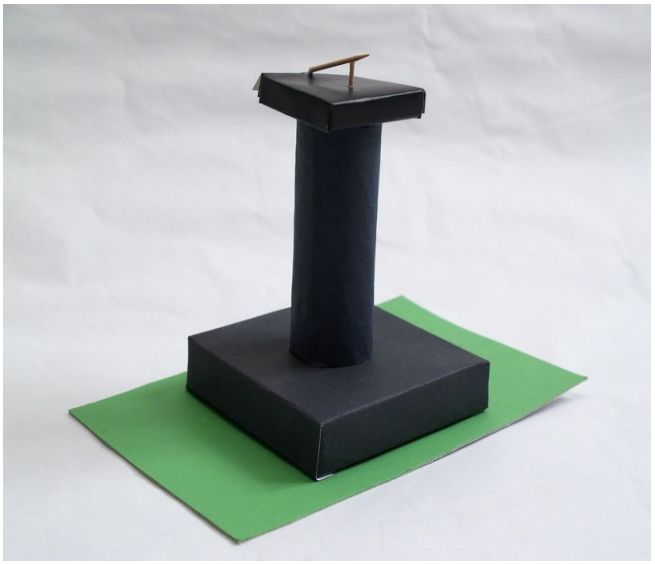
Obr. 27 – Detail slunečních hodin

Návrhy modelů slunečních hodin

Model č.1

• Kamenný podstavec s kovovým polosem by se měl nacházet v areálu školy – tj. v přístupném dvoře.

• Rozměry – 20 x 60 x 60 cm podstavec, 15 x 100 cm válec, 10 x 40 x 40 cm deska



Obr. 28 – Model číslo 1.

Model č.2

• Umístění na čelní stěně budovy školy, pro správnou funkčnost je třeba hodiny mírně odklonit od této stěny.

• Ciferník bude nakreslen přímo na zeď, ručička z nerezu. Design kapesních hodinek bude zachován.

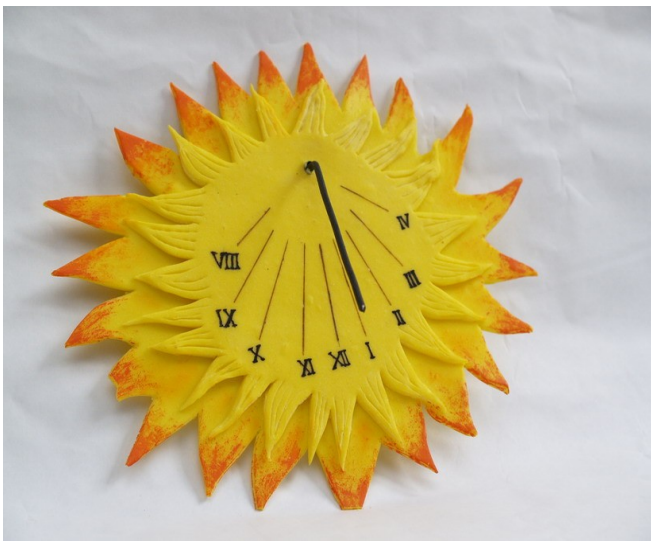


Obr. 29 – Model číslo 2.

Model č.3

• Poloha těchto hodin by měla být čelní stěna budovy školy orientovaná na jih, nebo i zadní stěna mířící na sever směrem k rybníku Jordánu.

• Materiál hodin by pravděpodobně měla být hlína nebo modurit.

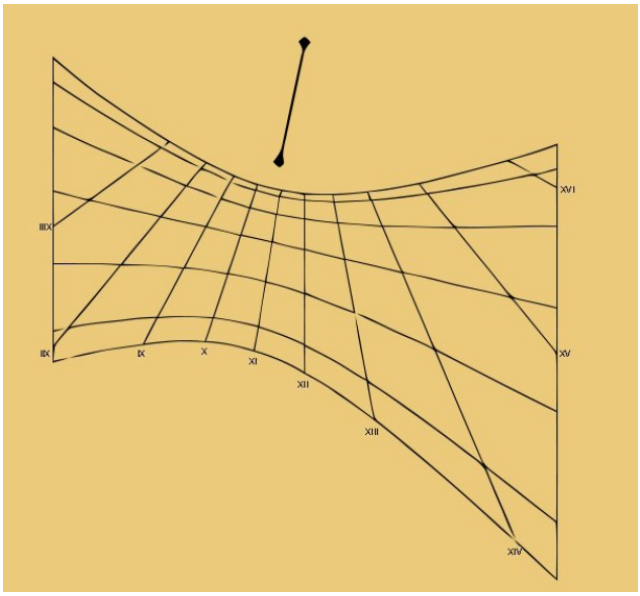


Obr. 30 – Model číslo 3.

Model č.4

• Pro tento model se opět nejlépe hodí čelní nebo pravá stěna budovy školy.

• Hodiny by byly řešeny jako nástěnná malba.



Obr. 31 – Model číslo 4.

Model č.5

• Podoba dopravní značky, nebo mandaly, průměr číselníku modelu je 8.45 cm. V reálu by byl předpokládaný průměr měl být 84.5 cm.

• Místo pro tento typ hodin je nejlepší u hlavního vchodu do budovy.



Obr. 32 – Model číslo 5.

Závěr

Práce na tomto projektu byla velmi zajímavá. Měli jsme možnost porovnat, jak by sluneční hodiny měli vypadat a jaké jsou ve skutečnosti. I když v dnešní době budování slunečních hodin upadá, stále ještě nezaniklo.

Také jsme zjistili, jak snadno se dá udělat při konstrukci chyba, která znemožní praktické využití hodin, a zůstane jen požitek estetický. Není se tedy čemu divit, že hodiny pocházející z poměrně dávné minulosti neplní svou původní funkci. Ovšem musíme uznat, že tento nedostatek plně kompenzují svou originalitou a výtvarnou krásou, která se jen tak nevidí.

Práce na návrzích byla zábavná a procvičili jsme si na ní svou zručnost a to, jestli jsme opravdu schopní hodiny zrealizovat. Máme velkou šanci vytvořit opravdu funkční sluneční hodiny bez chyb, před kterými jsme díky tomuto projektu varováni.

A ještě jeden pozitivní aspekt. Od této doby se všude, kam přijedeme, pozorně díváme na všechny stěny a objevujeme mnoho slunečních hodin. Dozajista jsme dnes více schopní je ocenit než kdy dříve, když víme, jak složité je je vytvořit a zkonstruovat.

Zdroje

Mapa ČR – místa slunečních hodin:
http://gismaster.liberix.cz/cgi-bin/mapserv?map=%2Fvar%2Fwww%2Flocalhost%2Fhtdocs%2Fslunhod%2Fslunhod.map&layer=%C4%8CR_plocha

Typy a princip slunečních hodin:
http://slunecni-hodiny.webzdarma.cz/

Sluneční hodiny méně obvyklých typů:
http://slunecnihodiny.wz.cz/galerie/gsh_p1/galerie-p1.htm

Sluneční hodiny na pevných stanovištích (Čechy, Morava, Slezsko a Slovensko) – editoři: Miroslav Brož, Miloš Nosek, Jan Trebichavský, Drahomíra Pecinová; nakladatelství: Academia
http://astro.mff.cuni.cz/mira/sh/sh.php



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

LIDSKÝ CHRUP HISTORIE A SOUČASNÉ TRENDY

Vypracoval: Jan Špaček
Garant: Michal Špaček
Trojské gymnázium Svatopluka Čecha

Prohlášení

Prohlašuji, že jsem tuto písemnou ročníkovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím s tím, aby práce byla zpřístupněna veřejnosti ke studijním účelům.

Poděkování

Děkuji svému garantovi Michalovi Špačkovi, studentovi 3. lékařské fakulty Univerzity Karlovy v Praze, za odbornou konzultaci. Dále bych chtěl poděkovat Mgr. Martinovi Markovi za informace týkající se struktury a povinných položek ročníkové práce.

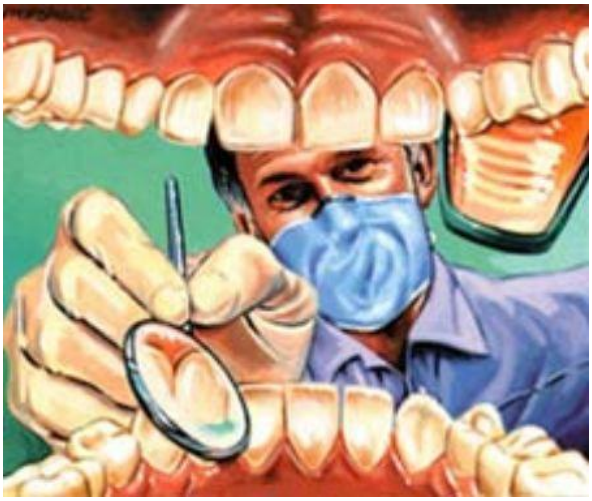
Obsah

1 Úvod .....
2 Stomatologie .....
2.1 Obory stomatologie .....
2.2 Morfologie zubů .....
2.2.1 obr. stavba zubu .....
2.3 Časový průběh dentice .....
2.3.1 tabulka: časový průběh dentice .....
2.3.2 tabulka: růst zubů .....
3 Vznik zubního kazu .....
3.1.1 obr. vznik zubního kazu .....
3.1 Rozdělení a ošetřování zubního kazu .....
3.3 Následky zubního kazu .....
3.4 Prevence zubního kazu .....
4 Úprava výživy .....
5 Péče o chrup .....
5.1 Základní pomůcky ústní hygieny .....
6 Zuby jako zázračný aparát lidského těla .....
6.1 Vznik skloviny a zuboviny .....
6.2 Péče o mléčné zuby .....
7 Zajímavosti .....
7.1 Sladkosti a zubní kaz aneb nejlepší chrup mají Skandinávci .....
7.1.1 obr. sladkosti .....
7.2 Kde se vzala zubní pasta ? .....
7.2.1 obr. 1. zubní pasta .....
7.3 Kde se vzal zubní kartáček ? .....
7.4 Žvýkačka na zuby ? .....
8 Parodontóza – atrofie dásní .....
8.1.1 obr. vznik parodontózy .....
9 Chybné postavení zubů .....
10 Literatura .....
11 Závěr .....
12 Obrazová příloha .....
12.1 obr. fixační rovnátko .....
12.2 obr. upevnění fixačního rovnátka .....
12.3 obr. rentgen lidského chrupu .....
12.4 obr. souhrnný přehled zubů v lidském těle .....
12.5 obr. řez zubem .....

1 Úvod

Mou inspirací na téma ročníkové práce bylo osobní zaměření do budoucnosti. Zajímám se o přírodní vědy a rád bych se jim do budoucna věnoval. Stavba lidského chrupu, jeho vývoj a především péče o něj se na první pohled zdá již zcela prozkoumaná a probádaná. Ale i nyní v 21. století se setkáváme s novinkami ve stomatologii.

Ve své ročníkové práci se zaměřuji na lidský chrup z hlediska odborného. Chci čtenářům přiblížit jak obecné informace, tak i tzv. „perličky“. Cílem práce není chrlit na čtenáře plno nesrozumitelných pojmů, ale seznámit je s péčí o chrup, jeho nemocemi a zásadami ústní hygieny.



2 Stomatologie

Stomatologie je lékařský obor, který se zabírá prevencí, diagnostikou a léčením chorob a úrazy zubů, dutiny ústní a všech orgánů a tkání souvisejících.

Zaobírá se kromě nemocí zubů a dutiny ústní také choroby čelistních kostí, slinných žláz, obličejových mízních uzlin, čelistního kloubu a žvýkacího svalstva.

2.1 Obory stomatologie

- Záchovná stomatologie – ošetřování zubního kazu a jeho komplikací. Součástí je endodoncie, která se věnuje ošetřování dřeňové dutiny a kořenových kanálků zubů.

- Protetická stomatologie – nahrazuje ztracené části zubu, ztráty celého zubu nebo skupin zubů a ztráty celého chrupu.

- Chirurgická stomatologie – tento obor se dělí na dentoalveolární a maxilofacální.
  Dentoalveolární chirurgie zahrnuje extrakci zubu, chirurgické ošetření, zánětlivých komplikací zubního kazu, léčení úrazů zubů, kostí i měkkých tkání, resekce kořenu zubu, chirurgické odstraňování menších cyst a řada dalších výkonů. Maxilofacální chirurgie se zaobírá operací velkých cyst, slinných žláz a ošetřování kolemčelistních zánětů.

- Paradontologie – se věnuje chorobám paradontu a ústní sliznice.

- Dětská stomatologie – veškerá prevence o mladou generaci do 18 ti let.

- Orotodoncie – se zabírá prevencí, diagnostikou a léčením nepravidelností mezičelistních vztahů a nepravidelného postavení jednotlivých zubů.

- Preventivní stomatologie – prevence stomatologických onemocnění – zejména kazu a parodontopatií, ale i ortodontických anomálií, onkologických onemocnění a úrazů.

- Stomatologická rentgenologie – se zabírá různými speciálními vyšetřeními.

- Forensní stomatologie – úzce spolupracuje se soudním lékařstvím a zabývá se především identifikací osob podle chrupu.

- Gerontostomatologie – se věnuje specifické problematice stomatologické péče o nejstarší generaci. (Šimůnek; 1997)

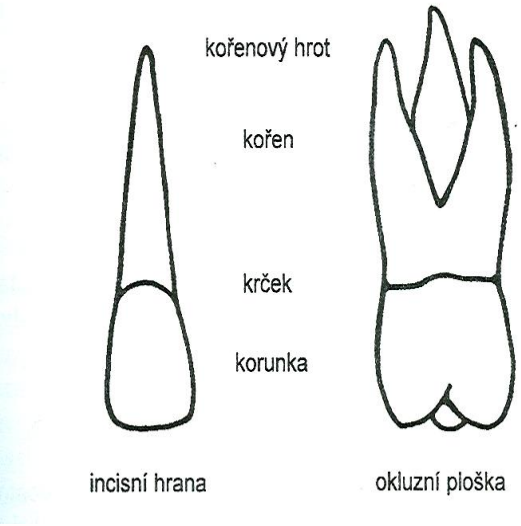
2.2 Morfologie zubů

Na jednotlivých zubech rozlišujeme korunku, krček a kořen. Podle tvaru korunky se dělí zuby na jednotlivé typy: *řezáky, špičáky, premorály* (třenové zuby) a *moláry* (stoličky). Korunka řezáků je tvarována do řezací hrany, u špičáků do hrotu, u premolárů a molárů má žvýkací plošku s hrbolky a rýhami. Řezáky a špičáky a premorály mají po jednom kořeni, s výjimkou horního prvního premorálu, která stejně jako dolní morály má dva kořeny. Horní moláry mají kořeny tři. Další moláry (zuby moudrosti) mohou mít nepravidelný počet kořenů.

☞ *Zub se skládá ze skloviny, zuboviny, cementu, zubní dřeně a ozubice.* ☜

Zubní sklovina je nejtvrdší tkání v lidském těle. Kryje celý povrch korunky zubu. Hlavní součástí skloviny zubu jsou krystalky hydroxiapatitu uspořádané do sloupců – prizmat, které probíhají paprskovitě od zuboviny povrchu zubu. Ve sklovině jsou přítomna také malá množství uhličitanů, hořčíků a fluóru.

2.2.1 obr. stavba zubu



Zubovina (dentin) obsahuje anorganické látky (72%) a látky organické (28%) a svým složením se podobá kosti. Organické látky jsou zastoupeny převážně kolagenními vazivovými vlákny a mukopolysacharidy, které tvoří základní substanci. V ní jsou uloženy krystalky hydroxiapatitu.

Zubní dřeň (pulpa) je tkáň uzavřená v dřeňové dutině a v kořenovém kanálku. Zubní dřeň je rosolovitá tkán obsahující fibrocyty a fibroblasty a malé množství tenkých kolagenních vláken. Buňky dřeně sousedící s dentinem se nazývají odontoblasty a mají schopnost vytvářet dentin. Četně cévy krevní a lymfatické vstupují do dřeně hrotovým otvorem, stejně jako nerovová vlákna.

Zubní cement pokrývá dentin kořenové části zubu. Cement se spojí skladbou rovněž podobá kosti. Do cementu se upínají kolagenní vlákna ozubice.

Ozubice (periodontium) spojuje zub s kostí zubního lůžka. Tento závěsný aparát zubu je tvořen systémem vazivových vláken, které se na jedné straně upínají do zubního cementu, na druhé do stěny kostního zubního lůžka a v oblasti krčku také do měkkých tkání dásně.

Alveoální výběžky horní a dolní čelisti se zubními lůžky (alveoly) mají na povrchu tenkou vrstvu kompaktní kosti a uvnitř je trámčitá spongiózní kost. I vnitřní povrch zubního lůžka je tvořen tenkou vrstvou kompakty. Jednotlivé zuby jsou odděleny přepážkami.

Dáseň (gingiva) je stejně jako ostatní sliznice dutiny ústní kryte vrstevnatým dláždicovým epitelem. Dáseň je pevně přirostlá k orkaji alveolárního výběžku a tvoří tzv. připojenou gingivu.

2.3 Časový průběh dentice

Dentice se definuje jako soubor zubů společných vlastností. Rozeznáváme dentici dočasnou a stálou. Po určitou dobu má dítě v dutině ústní zuby dočasné i stálé – smíšenou dentici. Dočasnou dentici tvoří 20 zubů, dentici stálou 28 – 36 zubů.

Prořezávání dočasných zubů probíhá od 6.měsíce do 24. – 30. měsíce věku dítěte v následujícím pořadí.

2.3.1 tabulka: časový průběh dentice

První dočasné řezáky	6. – 8. měsíc
Druhé dočasné řezáky	8. – 12. měsíc
První dočasné moláry	12. – 16. měsíc
Dočasné špičáky	16. – 20. měsíc
Druhé dočasné moláry	20. – 24. měsíc

Dítě by mělo mít prořezány všechny dočasné zuby do 24. měsíce věku, za ještě fyziologickou hranici se však považuje až 30. měsíc věku dítěte.

Stálé zuby prořezávají do dutiny ústní od 6 do 13 let věku dítěte. Výjimku tvoří třetí moláry, které se objevují v dutině ústní nejčastěji až po 18. roce věku.

Pořadí prořezávání stálých zubů:

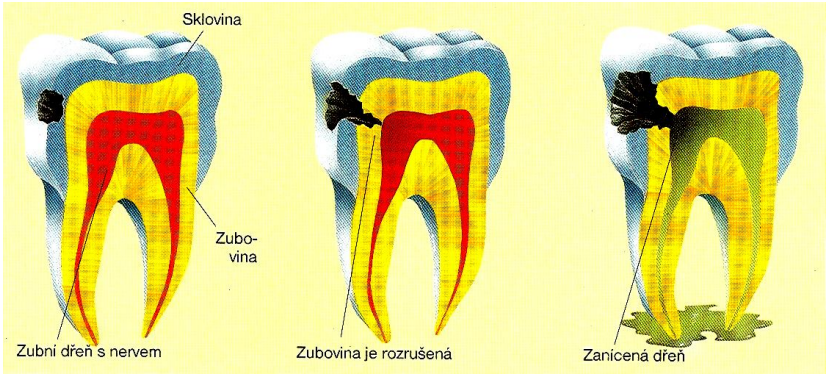
2.3.2 tabulka: růst zubů

První moláry	6. rok
První řezáky	6. – 7. rok
Druhé řezáky	7. – 8. rok
První premoláry	9. – 11. rok
Špičáky	10. – 11. rok
Druhé premoláry	10. – 12. rok
Druhé moláry	12. – 13. rok

3 Vznik zubního kazu

Zubní kaz je nejrozšířenější onemocnění zubů. Definuje se jako lokalizovaný patologický proces mikrobakteriálního původu, který postihuje tvrdé zubní tkáně. Podle současných názorů vzniká zubní kaz interakcí mezi třemi základními faktory: vnímavou zubní tkání, ústními bateriemi a dietou. Za čtvrtý faktor považujeme čas, protože zubní kaz se klinicky projevuje až po určité době. (Merglová; 2000)

3.1.1 obr. vznik zubního kazu



3.1 Rozdělení a ošetřování zubního kazu

Zubní kaz můžeme dělit podle postižení jednotlivých zubních tkání, podle časového průběhu, podle rozsahu defektu v tvrdých zubních tkáních a vztahu je dřeni a podle charakteru šíření kazivé destrukce.

Podle postižení tvrdých zubních tkání rozlišujeme:

- Kaz skloviny
- Kaz dentinu
- Kaz cementu

Podle časového průběhu zubního kazu rozeznáváme:

- Akutní kaz – vyznačuje se rychlým průběhem, ztrátou většího množství tvrdých zubních tkání, rychlým ohrožením zubní dřeně a postižením více zubů.

- Chronický kaz – má pomalý průběh, šíří se spíše do šířky než k zubní dřeni a postižené tvrdé zubní tkáně bývají tmavě pigmentované.

Podle rozsahu defektu v tvrdých zubních tkání rozlišujeme kazy:

- Kaz povrchový – zasahuje sklovinu a dentic pouze v těsné blízkosti skloviny
- Kaz středně hluboký – probíhá ve střední vrstvě dentinu
- Kaz blízký dřeni – spodina kazu je v blízkosti dřeně



3.3 Následky zubního kazu

Mezi následky neošetřeného nebo chybně ošetřeného zubního kazu řadíme překrvení a zánět dásně, nekrózu (odumření) dřeně, popř. zánět okostice, záněty lymfatických uzlin až po zánět čelistních kostí.

3.4 Prevence zubního kazu

Fluorizace pitné vody: Fluorizaci doporučuje Světová zdravotnická organizace jako jednu z výhodných metod prevence zubního kazu. Odhaduje se, že fluorizace pitné vody snižuje kazivost chrupu o 40 – 50 %. Výhodou této metody je, že nevyžaduje přímou účast zdravotníků a náklady na fluorizaci vody jsou poměrně nízké. I přes uvedené výhody se fluorizace pitné vody v současné době v České republice neprovádí.

Fluoritové tablety: Tyto tablety ordinuje stomatolog nebo pediatr dítěti nejpozději v 6 měsících věku. Podávají se až do doby prořezání druhého stálého moláru.

Fluoridovaná sůl: Fluorizace soli představuje další možnost v prevenci zubního kazu, fluoridovaná sůl se používá pouze k dosolování jídel.

Minerální vody: Přírodním zdrojem fluoru jsou i minerální vody. Minerální vody se rozdělují na přírodní minerální vody, přírodní léčivé, přírodní minerální vody stolní. Pro prevenci proti zubnímu kazu lze doporučit přírodní minerální vody stolní. Stolní minerální vody jsou totiž vhodné pro prevenci zubního kazu jak pro jednotlivce, tak i pro skupinové použití v dětských kolektivech.

Fluoridované zubní pasty: jsou v současné době nejrozšířenějším a nejvýznamnějším prostředkem v prevenci zubního kazu. V zubních pastách jsou nejčastěji fluoridy obsaženy ve formě fluoridu sodného. Koncentrace fluoridu se v zubních pastách liší. Nejnižší je v zubních pastách určených pro malé děti, naopak vysoký podíl fluoridu nalezneme v tzv. terapeutických, určené pro mladistvé a dospělé jedince.

Ústní vody se sloučeninami fluoru: Ústní vody obsahují většinou fluorid sodný. Používáním ústních vod je pouze doplněk ústní hygieny, který nenahrazuje mechanické odstraňování zubního plaku kartáčkem.

Fluoridované roztoky a laky: Fluoridované roztoky a laky jsou určeny pro použití ve stomatologické ordinaci. U dětí se aplikují 2krát až 4krát ročně na vyčištění povrchů zubů.

Žvýkačky s fluoridy: Fluoridy se mohou místně aplikovat také ve formě žvýkací gumy s fluoridy. Tato metoda je výhodná zejména pro děti školní věku a dorostového věku. (Koukal; 2007)

4 Úprava výživy

Správná výživa je důležitou součástí prevence zubního kazu. Požadavky stomatologů na správnou výživu nejsou v rozporu s doporučením pediatrů a dietologů, protože špatné stravovací návyky, které způsobují zubní kaz, mohou být příčinou obezity, arteriosklerózy a dalších nemocí.

Úprava výživy vzhledem k prevenci zubního kazu je důležitá v každém věku, ale u dětí zvláště. Špatně stravovací návyky fixované v dětství se v dospělosti velice těžce mění.

Správná výživa je významná zejména u kojenců a batolat. Nejvhodnější způsob výživy v kojeneckém věku je kojení. Pokud dítě dostává umělou výživu, neměla by být přislazována. Okolo 6. měsíce věku začínají dítěti prořezávat první dočasné zuby. Pokud dítě dostává dudlík namočený do medu nebo cukru, může se velice rychle vytvořit při absenci ústní hygieny zubní kaz. Stejné nebezpečí představují i sladké nápoje podávaná v kojenecké láhvi často během dne a zejména před usnutím a v noci. Takto vzniklému kazu, který postihuje hlavně horní dočasné řezáky a způsobuje jejich rychlou destrukci, se říká „medový kaz“ nebo „kaz z kojenecké láhve“. S tím to problémem se setkáváme i v batolecím věku, protože pití z láhve je obvyklé i u dětí této věkové skupiny.

Další nebezpečí pro vznik kazu představují u dětí různé cukrovinky, slazené nápoje, ale i bramborové lupínky a slané tyčinky podávané často během dne. Naopak za prospěšné se považuje žvýkání žvýkaček bez cukru po každém jídle.

Časté konzumování sladkostí, sladkých nápojů a jídel z tepelně upravených škrobů ve spojitosti s nedostatečnou ústní hygienou představuje velké nebezpečí pro vznik zubního kazu, a to již u velmi malých dětí. O správné výživě ve vztahu k zubnímu kazu by proto měli být informování nejen rodiče, prarodiče, ale i učitelé a vychovatelé.

5 Péče o chrup

5.1 Základní pomůcky ústní hygieny

- Zubní kartáček
- Zubní pasta
- Zubní nit

Zubní kartáčky

Jsou nejběžnější pomůckou domácí péče o chrup. Za nejvhodnější zubní kartáček se v současné době považuje kartáček s krátkou pracovní částí, která usnadňuje manipulaci v ústech. Štětinky zubního kartáčku mají být vyrobeny ze syntetických vláken se zaoblenými konci. Doporučuje se používat zubní kartáček se středně tvrdými štětinami, u malých dětí jsou vhodné měkké zubní kartáčky. Při pravidelném čištění tvrdými kartáčky nevhodnou technikou je nebezpečí poškození tvrdých zubních tkání. Doporučená doba výměny kartáčku jsou 2 měsíce.

Zubní pasty

Zubní pasty usnadňují mechanické odstranění plaku. Kromě již uvedených fluoridů obsahují léčebné prostředky, abraziva, pojiva a plnidla, chuťová korigencia a vonné látky.

Zubní nit

Zubní nit se používá k odstraňování dentálního plaku z mezizubních prostorů. Používání zubní nitě lze doporučit až po důkladném nácviku v ordinaci zubního lékaře.

Pro udržování dokonalé hygieny dutiny ústní se používají i další speciální hygienické pomůcky jako jsou mezizubní kartáčky, jednosvazkové kartáčky a tzv. superfossy. Tyto hygienické pomůcky jsou vhodné například pro pacienty s fixními Ortodontickými aparáty, s fixními protetickými náhradami, implantáty nebo paradontologickými problémy.

6 Zuby jako zázračný aparát lidského těla

Zuby jsou poslední dobou odborníky považovány jako za zázrak přírody. Nejenom proto, jak jsem již podotkl v předešlých odstavců, je sklovina nejtvrdší část těla, je tvrdší než kost, ale na rozdíl od kosti nedorůstá. Víme také, že jimi můžeme oslňovat kolemjdoucí nebo naopak nám mohou způsobit nevyslovitelnou bolest. V následujících odstavcích se ponoříme hlouběji pod povrch zubů.

Jak již víme, zub je dutý útvar, který se liší podílem jednotlivých složek, a tedy i vlastnostmi. Jeho nejnápadnější a jedinou viditelnou částí je korunka, kterou tvoří vrstva skloviny (email). Vytvářejí ji specializované buňky – ameoblasty, které posléze zanikají. Zajímavostí a také vysvětlením, proč je sklovina nejtvrdším materiálem lidského těla je poměr anorganických a organických látek, který je 97 : 3.

Pod sklovinou se nachází zubovina (dentin), kterou vytvořily odontoblasty, které nezanikají. Zubovinu tvoří ze dvou třetin minerály, zbytek je organická hmota, což jsou nejrůznější proteiny a bílkoviny, nakonec i voda. Je proto měkčí než sklovina, a tím i méně odolná proti mechanickému poškození včetně útokům bakterií v lidských ústech. Představuje vlastně „kostru“ zubu, protože z části korunkové přechází v krčkovou a v kořen, kterým je zub zakotven v čelisti.

Společné pro korunkovou, krčkovou a kořenovou část zubu je dřeňová dutina, vyplněná u zdravého zubu zubní dření – řídkým rosolovitým vazivem, v němž jsou nervy a cévy, zajišťující zásobování zubu potřebnými živinami a ochrannými látkami. Díky nervu cítíme teplo a chlad. Zuby jsou v kosti uchyceny závěsným aparátem zvaný parodont nebo – li ozubice. (Koukal; 2007)

☞ *Zub může mít více než jeden kořen.* ☜

☞ *Nejpevněji jsou zakotveny stoličky, které při skusu vyvíjejí největší sílu.* ☜

6.1 Vznik skloviny a zuboviny

Vznik skloviny a zuboviny začíná v době, kdy specializované buňky nejdříve z bílkovin v kosti vytvoří měkkou formičku. Do ní se z krve při vytvrzování ukládají minerály – jejich soli: vápník, fosfor, fluor. Jejich množství a vzájemný poměr rozhodují o odolnosti skloviny. Jako první mineralizuje korunka první stálé stoličky, úplně naposled zuby moudrosti.

Zuby *jsou tvořeny dvěma různorodými složkami – Anorganická a Organická složka*

Anorganická složka – tvořená apatitem výrazně převažuje. Na vzniku apatitu se podílí vápník, fosfor, fluor, kyslík a vodík. Apatit svou vnitřní strukturou připomíná včelí plasty. Místo voskových stěn zde však nalezneme organickou hmotu. (Dr. Smith 1997)

Organická složka – představují zejména proteiny (bílkoviny)

6.2 Péče o mléčné zuby

Omylem platí domněnka, že mléčné zuby není třeba čistit, protože po čase stejně postupně jeden za druhým vypadnou. Ale od počátku na ně působí vnější

vlivy. Jsou však náchylnější k zubnímu kazu, vzhledem k měkčí sklovině, která se rychleji odírá. Stav tohoto dočasného chrupu navíc výrazně ovlivňuje, jak bude posléze vypadat trvalý chrup. Každý mléčný zub totiž trvale drží místo svému následníkovi.

Zajímavostí je, že vývoj dětského chrupu narušují již od narození nejrůznější zlozvyky jsou je cumlání palce, dudlíku v době, kdy je dítě již „zubaté“. Také okusování nehtů může způsobit křivost zubů, která se pak musí řešit rovnátky.

Léčením různých nepravidelností zubů a čelistí se zabývá lékařský obor zvaný ortodoncie. V současné době mají v České republice problém s touto problematikou nepravidelnosti problém 60 – 80 % obyvatel.

7 Zajímavosti

7.1 Sladkosti a zubní kaz aneb nejlepší chrup mají Skandinávci

Neradostnou roli v dětské ústní dutině sehrává cukr. Sladkost vytvoří v ústech velmi agresivní látky, z čehož se převelice radují bakterie zubního plaku, které se po vydatné dávce cukru rozmnoží. Kaz nejdříve zasáhne plošky předních zoubků, někdy postupuje po obvodu celého krčku, což nezřídka končí ulomením celého zubu. Mléčnému chrupu však neškodí jen sladkosti a sladké nápoje. Mezi další nepřátelé patří i umělé i přírodní kyseliny a chemikálie v potravinách. Příkladem jsou například nejrůznější ovocné šťávy, džusy. Nejlepší nám tedy zůstává čistá voda. (Koukal; 2007)

7.1.1 obr. sladkosti



Zdravé první zuby v pěti letech má u nás podle posledních průzkumů jen 41,6 % dětí. Jejich malí vrstevníci ve Francii, Velké Británii či Nizozemsku se mohou pochlubit 55,5 %, zatímco malí Skandinávci se mohou pochlubit prvenstvím se „zdravým úsměvem“ až v 70%.

7.2 Kde se vzala zubní pasta ?

Egypt: už ve starém Egyptě se objevila první předchůdkyně zubní pasty. Avšak rozdíl od dnešní zubní pasty je zřetelný už tím, že ve starém Egyptě měla podobu prášku, smíchaný s popelem z kopyt dobytka, myrhy, rozemletých vaječných skořápek a pemzy.

Arabové: si zase dřevěnými tyčinkami roztírali „zubní pastu“ složenou z popele, jeleního parohu, mastixu, soli, kamence a myrhy. Tato směs údajně zvyšovala i tvrdost zubů.

Peršané: měli prášek z jeleního paroží, pálených hlemýždích ulit a šklebí či z pálené sádry.

Anglie: v Anglii se mohli vyznavači ústní hygieny rozhodnout mezi čistícím práškem v keramických miskách a pastou.

7.2.1 obr. 1. zubní pasta



1. zubní pasta, která „vylezla“ z tuby, jak ji známe v dnešní podobě, se datuje k roku 1850 díky americkému stomatologovi Washingtonovi Wentworth Sheffieldovi.
(Koukal; 2007)

7.3 Kde se vzal zubní kartáček ?

Staleté záznamy připomínají, že Arabové si již před několika tisíci lety čistili zuby dřevěnou tyčkou ve tvaru štětečku. Převrat představoval nápad zchudlého francouzského šlechtice Josefa de la Maire. Ten se ve 20. letech 19. století v Paříži živil jako zubní lékař, aniž měl potřebného vzdělání. Klientům většinou zuby trhal. Časem ho to však unavovalo, a tak tvrdil, že nejlepší je ústa vyplachovat čistou vodou a zuby čistit měkkým kartáčkem. Tento osvícenec později vydal knihu s názvem „Dámský zubař“, k níž každý kupující dostal zubní kartáček.



7.4 Žvýkačka na zuby ?



Vedle pravidelného čištění zubům pomáhá i vhodná žvýkačka (samozřejmě bez cukru). Opět ve starém Egyptě si před 3500 roky osvěžovali dech kuličkami z melounu, myrhy a kadidla.

Starověcí Řekové žvýkali pryskyřici z kůry keře Pistacia lentiscus. Jedině mastixem si desinfikovaly ústa i Řekyně. Z historie se nám poskytuje záznam o „prababičce žvýkačce“, který se objevil na jihu Švédska při vykopávkách. V Americe zase indiáni žvýkali mízu z jehličnanů, hlavně smrku či jedle.

8 Parodontóza – atrofie dásní

Tuto nemoc se snažím ve své práci zdůraznit už pouze pro to, že se jedná o jednu z nejvíce se vyskytujících onemocnění chrupu.

Co to vlastně parodontóza je?

- Začíná většinou zánětem dásní. Když se toto onemocnění včas neléčí, šíří se až dolů do úzkých dásňových chobotů, které se nalézají mezi dásněmi a krčky zubů. Když zánět přejde na povrch kořene a okolní čelist, ve které je zub pomocí vláknité, pojivé tkáně ukotven, začne se zub uvolňovat.

Jaké jsou příčiny parodontózy?

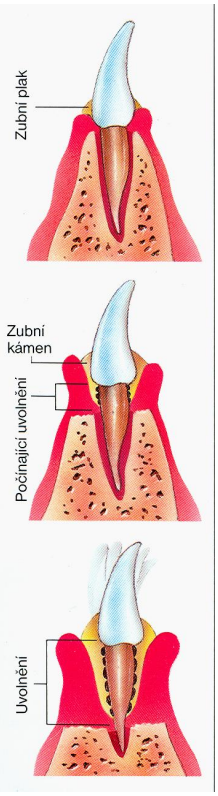
- Parodontóza je zavíněna nedostatečnou péčí o zuby. Když se zuby pravidelně nečistí, tvoří se na povrchu zubů povlak ze zbytků jídla a odumřelých buněk ústní sliznice.
  Tomuto povlaku se říká *zubní plak* – je to ideální půda pro množení bakterií a hub. A opět na jejich základě se tvoří nejen zubní kámen, ale jsou také příčinou vzniku zánětu dásní.

Léčba parodontózy …

- Nejdůležitější prevencí je pečlivá pravidelná zubní hygiena, která znemožní přečnívání bakterií v prostorech mezi zuby a dásní.

Příznaky:

- Tvorba zubního kamene na povrchu zubu
- Dáseň při čištění zubů bolí a krvácí
- Dásňové choboty
- Uvolnění zubů



☞ *Ihned během objevení prvních příznaků nám nezbývá nic jiného než ihned vyhledat lékařskou pomoc.* ☜

Průběh nemoci:

Povlak zubů, který se denně znovu vytváří, je měkký a dá se lehce odstranit zubním kartáčkem. Pokud se však denně neodstraňuje, stává se rychle tvrdým a pevným. Vytvoří se zubní kámen, na kterém se mohou lehce usadit bakterie a houby. Po nějaké době se zanítí i dáseň a dostane temně červenou barvu. Pokud zánět dále postupuje, vniká do dásňových chobotů a dochází k nevyhnutelným následkům – uvolnění zubů.

8.1.1 obr. vznik parodontózy

Po seznámení s touto zákeřnou chorobou se zeptejme sami sebe: „Jak se s ní dokážeme vypořádat?, jak odolat?“

Poskytnu Vám pár rad:

- Malou konzumací sladkých pokrmů (nejlépe vůbec)
- Alespoň dvakrát denně si vyčistěme zuby, obzvláště na noc je to důležité
- Mezery mezi jednotlivými zuby čistíme zubní nití nebo alespoň pročistíme párátkem.
- Pravidelně navštěvujeme zubního lékaře !!!

(Dr. Smith; 1993)

Poslední dobou zažila stomatologie tzv. „bum“ v oblasti zarovnávání a srovnávání zubů. V následujících odstavcích se tudíž budu zabývat chybným postavením zubů.

9 Chybné postavení zubů

Co to znamená?

- Chybná postavení zubů patří k velké skupině anomálií tvaru a funkcí chrupu. Lékař rozeznává mezi prostým chybným postavením chrupu či jednotlivého zubu a chybným postavením celého skusu.
- U chybného postavení zubů nejsou postaveny jednotlivé zuby nebo skupiny zubů v ideálním oblouku čelisti. Při vadném postavením skusu se jedná o vzájemné posunutí horní a dolní čelisti. V naprosté většině těchto případů se ovšem jedná nejčastěji o kombinaci obou.

Příčiny:

- Chybná postavení zubů vznikají většinou v dětství. Častou příčinou je cumlání palce a kousání rtů a jazyka. K chybnému postavení zubů vede také příliš dlouhodobé používání dudlíku nebo kojenecké láhve. Vývoj chrupu je také řízen dědičnými faktory.

- Chybná postavení zubů se mohou projevovat u dětí již v mléčném chrupu, častěji se však objevují až v chrupu trvalém. Právě při vývoji druhých zubů dochází v čelistech k jejich značnému posunování a stlačování.

Léčba:

- Chybné postavení zubů se léčí individuálně pomocí zhotovených strojků nebo svorek. V dětském věku se většinou používají snímatelné aparáty (nasazovací rovnátka).

Co můžeme udělat sami?

- Rodiče musí sledovat, jak se zuby jejich dětí vyvíjejí. Známkami chybného postavení jsou nepravidelnosti v zubním oblouku, velké mezery nebo naopak přílišné vzájemné naléhání zubů a také poruchy skusu.

Rodiče by vskutku měli dohlížet na své děti, aby jim jejich chrup dobře rostl. Již od útlého věku je důležité sledovat, aby zuby rostly tak, jak mají a při jakékoli komplikaci ihned vyhledat lékařskou pomoc. V mé ročníkové práci jsem byl sám překvapen, jak je pouhý jeden zub důležitý. Jakou má stavbu a jak je odolný.

Vždyť i pouhé bolení zubů je nepředstavitelné trápení. Celkově bych mohl zhodnotit, že jakékoli poranění hlavy či dokonce dutiny ústní je velice bolestivé. Osobně se znám s mně blízkými lidmi, kteří parodontózu mají. Do konce života mají jisté to, že je zuby už nikdy nepřestanou bolet. Díky této nemoci si bohužel někteří již pozdě uvědomí, jak důležité je o naše zuby dbát již od útlého věku. Neztotožňuji se s názorem mnoha lidí, že si chrup když tak v nouzi nahradí umělým. Toto řešení je konečné! Mysleme na to …

10 literatura

Merglová, Vlasta; a kol., *Stomatologie pro studující bakalářských oborů lékařské fakulty*, Praha, Karolinum, 2000

Šimůnek, Antonín; a kol., *Stomatologie pro posluchače všeobecného lékařství*, Praha, Karolinum, 1997

Dr. Smith, Tony, *Rodinná encyklopedie zdraví*, Praha, The British Medical Association, 1993

Kolektiv autorů, *Domácí lékař od A do Z – Rádce pro zdraví*, svazek 2, část 4 – Trávicí orgány, Praha, International Masters Publisher s. r. o.

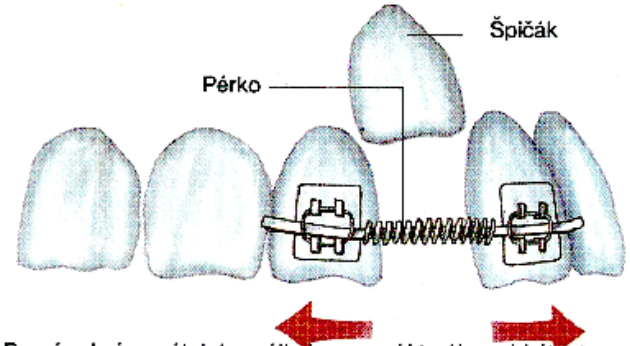
Koukal, Milan, časopis 21. století, *Porostou nám někdy třetí zuby?*, 2007, ročník 2, strana 68 - 76

11 Závěr

Je mnoho pohledů, jak se dívat na oblast stomatologie. Některé jsou více populární až zavádějící, jiné příliš suchopárné, postavené na ryze vědeckých základech. Ve svém přístupu jsem vycházel ze seriozních vědeckých poznatků a zároveň se snažil volit formu, která by byla srozumitelná pro základní orientaci. Snažil jsem se použít příklady snadno zapamatovatelná pro širší skupinu čtenářů, kteří nemusí mít dokonalé znalosti o tom, co pro ně vlastně zuby znamenají a jak s nimi zacházet, aby jim co nejdéle vydržely zdravé.

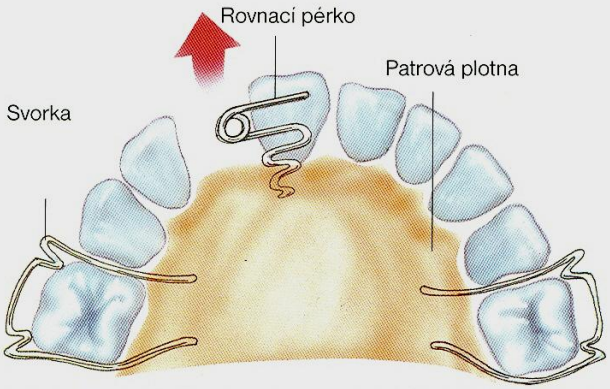
12 Obrazová příloha

12.1 obr. fixační rovnátko

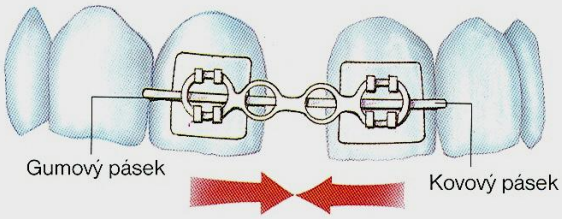


Pevný zubní aparátek (rovnátko), pomocí kterého má být vytvořen prostor pro špičák. Pérko odtlačuje zuby od sebe.

12.2 obr. upevnění fixačního rovnátka

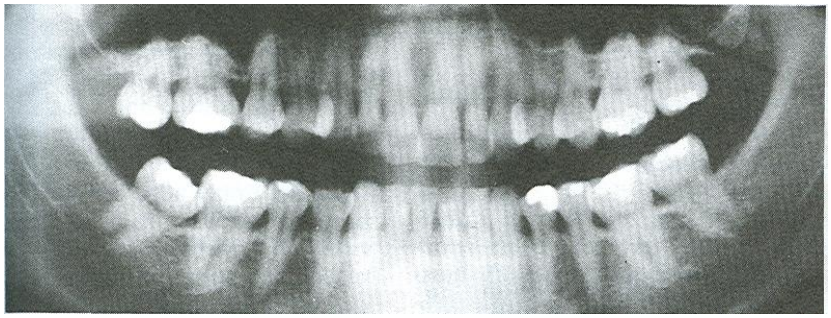


Snímací rovnací strojek, který je upevněn pomocí svorek. Pérko srovnává řezák.

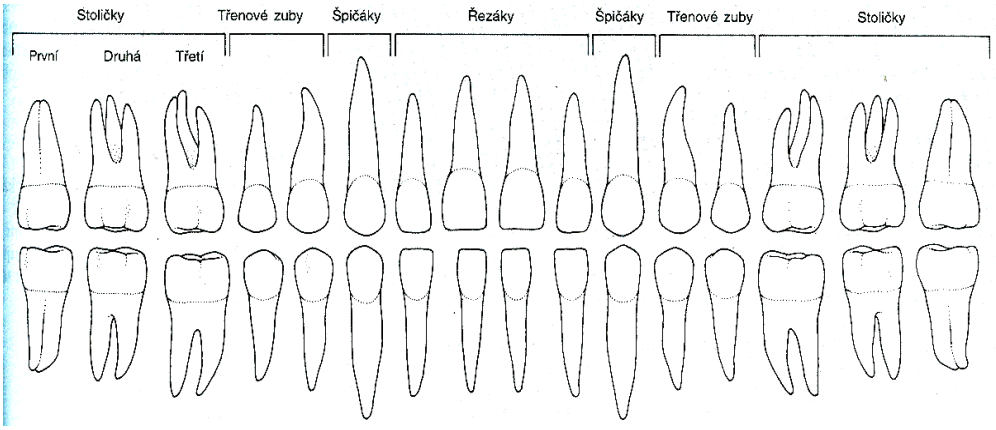


Tímto zařízením se stahují mezizubní prostory. Gumový pásek stahuje zuby k sobě, zatímco kovový pásek pečuje o jejich správné postavení.

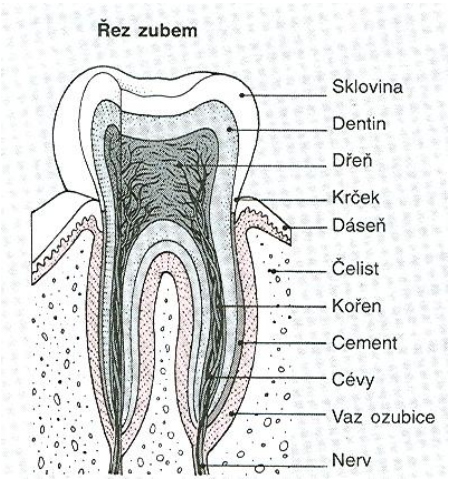
12.3 obr. rentgen lidského chrupu



12.4 obr. souhrnný přehled zubů v lidském těle



12.5 obr. řez zubem





Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

SKLADOVÁ EVIDENCE PRO MENŠÍ FIRMY

Ondřej Perutka

Střední škola průmyslová a hotelová Uherské Hradiště
Kollárova 617, Uherské Hradiště

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechny použité zdroje a literaturu.

Dále nesouhlasím s tím, aby s výsledky mé práce (ať už se jedná o program samotný, tento text nebo prezentaci) bylo nakládáno jakýmkoliv způsobem v rozporu s mými autorskými právy. Toto dílo je chráněno autorským zákonem.

V Uherském Hradišti dne 13. května 2008

Anotace

Ondřej Perutka, Skladová evidence pro menší firmy. Práce STRETECH.
Uherské Hradiště 2008

Tato práce popisuje uživatelské rozhraní, funkce a technické pozadí vývoje programu pro vedení skladového hospodářství. Je zde také uvedeno jakým směrem se bude pravděpodobně ubírat budoucí vývoj a jaké funkce bude program po svém dokončení obsahovat (výčet samozřejmě není konečný, protože vývoj jde neustále dopředu).

Obsah

1. Úvod 5
2. Seznámení s uživatelským rozhraním 6
3. Funkce programu 7
3.1. Skladové hospodářství 7
3.1.1. Vytváření skladové karty 8
3.1.2. Editace skladové karty 9
3.1.3. Mazání skladové karty 9
3.1.4. Příjem / výdej 9
3.2. Základní účetní operace 10
3.3. Zpracování podkladů pro přiznání DPH 10
3.4. Zjištění stavu skladu 11
3.5. Propojení s registrační pokladnou 12
3.6. Tisk čárových kódů 12
3.7. Zálohy a obnova dat + import / export 12
4. Technické pozadí 13
4.1. Systémové požadavky 13
4.2. Programová struktura a budoucí vývoj 13
5. Závěr 16
Literatura 17

1. Úvod

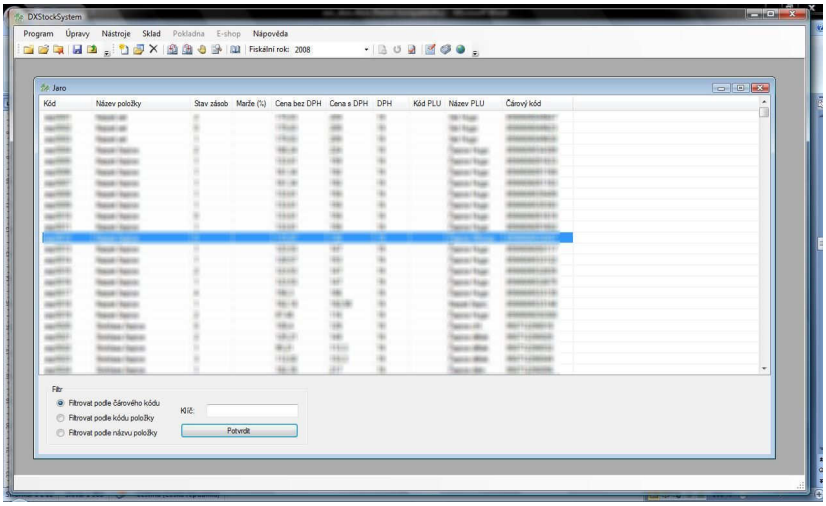
Proč vlastně skladový program? Na tuto myšlenku mě asi před rokem přivedla absence softwaru, který by byl postaven na otevřené architektuře[1] a byl schopný spolupracovat i s dalšími produkty jako jsou e-shopy nebo pokladny a zároveň by byl z uživatelského hlediska jednoduchý na pochopení, na českém trhu. Samozřejmě, že lze najít velké množství podobných programů, ale tyto programy mají spoustu nevýhod (počínaje přehnanou složitostí uživatelského rozhraní, až po archaické technické zpracování v prostředí konsole). Zkrátka bylo potřeba vytvořit takový program, který bude mít moderně zpracované, jednoduché a intuitivní uživatelské prostředí, širokou škálu funkcí, relativně nízké hardwarové nároky a svými možnostmi neurazí ani profesionály.

Pozn.: V textu se objevují obrázky z programu, které se mohou zdát na první pohled rozmazané, ve skutečnosti jsou rozmazány jen určité části, které obsahují důvěrná data. (Jedná se totiž o snímky ze skutečného skladu a zákazník si nepřál, aby tato data pronikla na veřejnost.)

[1] Tím nemíním OpenSource, ale program navržený takovým způsobem, aby se dokázal přizpůsobit růstu firmy.

2. Seznámení s uživatelským rozhraním

Pro tvorbu uživatelského rozhraní jsem se nechal inspirovat programovým produktem Microsoft® Office® (ne Office® 2007 - ten využívá nové rozhraní Silverlight®). Tento „vzhled“ je v dnešní době nejvíce zaužívaný, proto se domnívám, že se i uživatelé se základními znalostmi počítačů dovedou v programu rychle zorientovat, aniž by četli nějaký návod.



Obrázek 1 - Vzhled programu

Kostru programu tvoří základní okno s rozhraním MDI (Multiple Document Interface), které obsahuje hlavní menu a nástrojovou lištu. Po otevření některého skladu z databáze se všechny nabídky podstatně rozšíří a otevře se okno se seznamem skladových karet. V dolní části se zobrazuje tzv. filtr, který umožňuje různým způsobem filtrovat zobrazované položky. Nenechte se zmást jednoduchostí nebo zdánlivou nepotřebností této „pomůcky“, jedná se o neocenitelného pomocníka, který umožní uživateli bleskové vyhledávání a orientaci ve velkém množství položek. Navíc tento filtr neslouží pouze pro hlavní okno, jak by se mohlo na první pohled zdát, ale nastavení filtru se využívá i v dalších dialogových oknech, což velmi zrychluje a ulehčuje práci s celým programem.

3. Funkce programu

Doposud jsem se zmiňoval o „skladovém programu“, nicméně tento program má o něco širší záběr. Zde je přehled funkcí:

- Skladové hospodářství (evidence skladových karet, příjem, výdej)
- Základní účetní operace (fakturace)
- Zpracování podkladů pro přiznání DPH
- Zjištění stavu skladu (přehledová inventura, uzávěrka skladu na konci účetního období)
- Přehledný archív všech dokumentů
- Adresář
- Propojení s registrační pokladnou (typ EURO 2000TE Alpha)
- Tisk čárových kódů
- Ochrana heslem
- Zálohování a obnova dat
- Export adresáře a skladových karet

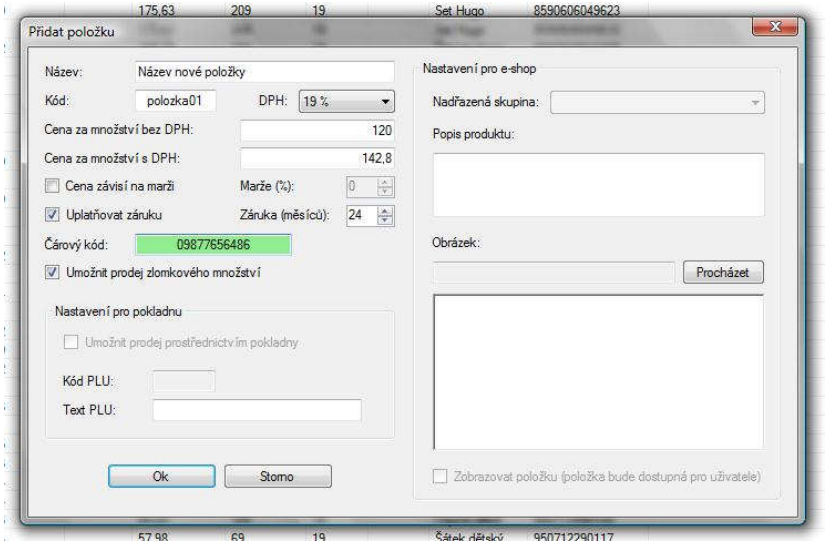
Podrobněji zde popíšu některé z nich.

3.1. Skladové hospodářství

Funkce, pro kterou byl program primárně vyvíjen. Umožňuje snadnou administrativu velkého množství položek. Podívejme se tedy blíže na „život“ jedné takové položky.

3.1.1. Tvorba skladové karty

Každou skladovou kartu lze vytvořit jednoduchým způsobem. Po kliknutí na „Vytvořit položku“ v menu „Úpravy“ se zobrazí dialogové okno, ve kterém jsou zobrazena různá textová pole, ne všechna však musí být vyplněna. Povinným údajem je pouze kód položky, její cena (v případě, že není použita marže) a název PLU (v případě, že bude umožněno prodávat položku prostřednictvím pokladny), ostatní údaje vyplňuje uživatel dle vlastního uvážení.



Obrázek 2 - Dialogové okno pro tvorbu skladové karty

V případě, že se uživatel rozhodne používat čárový kód, je toto textové pole vybaveno schopností „inteligentně“ rozpoznat typ a správnost zadaného čárového kódu, což se projeví v podbarvení textového pole. Zde je význam barev:

Bílá	Zobrazuje se v případě, že není zadán žádný kód nebo zadaný kód není rozpoznán.
Červená	Zobrazuje se v případě, že je v řetězci nějaká chyba.
Zelená	Zobrazuje se, když je zadán čárový kód typu UPCA bez kontrolní číslice (kontrolní číslice se automaticky dopočítá).
Oranžová	Zobrazuje se, když je zadán čárový kód typu UPCA se správnou kontrolní číslicí.
Modrá	Zobrazuje se, když je zadán čárový kód typu EAN13 (u tohoto typu neprobíhá ověření kontrolní číslice).

Tabulka 1 - Význam podbarvení pole pro zadání čárového kódu

3.1.2. Editace skladové karty

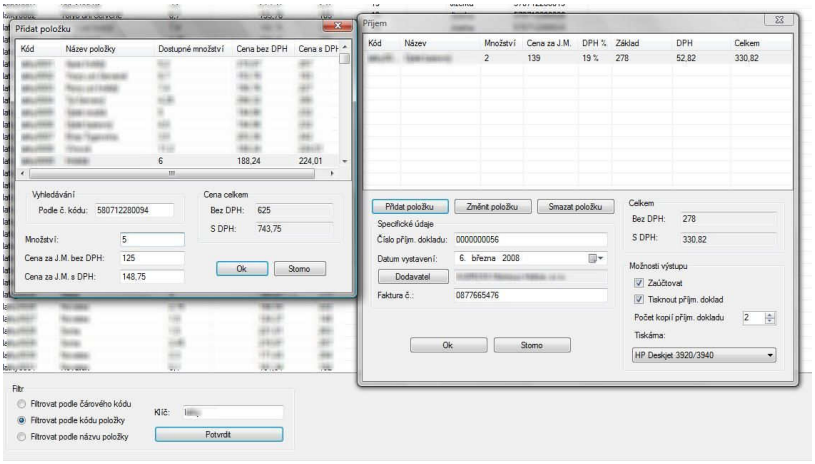
Editace karty je ve své podstatě prakticky stejným úkonem jako její tvorba, nicméně je z pochopitelných důvodů zakázáno editovat kód položky a pro přehlednost a pohodlí uživatele je zde zobrazena i poslední nákupní cena.

3.1.3. Mazání skladové karty

Pokud je potřeba nějakou skladovou kartu vymazat, není nic jednoduššího, než označit danou položku a stisknout klávesu DELETE. Samozřejmě, že nejde položku vymazat jen tak, jedinou podmínkou je, že množství této položky na skladě musí být rovno nule.

3.1.4. Příjem / výdej

Příjem i výdej položek je v programu realizován navzájem velice podobnými dialogy, které tvoří seznam položek, číslo dokumentu, možnosti tisku a zaúčtování a některé další údaje a nastavení. Přidávání a editace položek příjemky (resp. výdejky) je realizována samostatnými dialogovými okny, ve kterých si uživatel vybere ze seznamu skladových karet a doplní příslušné údaje (jako je množství, cena, …) v těchto dialogových oknech je (jak již bylo výše zmíněno) použit filtr položek pro jednodušší orientaci a navíc je možné položky vyhledávat pomocí čárového kódu (buď opsáním kódu, nebo naskenováním čtečkou čárových kódů).



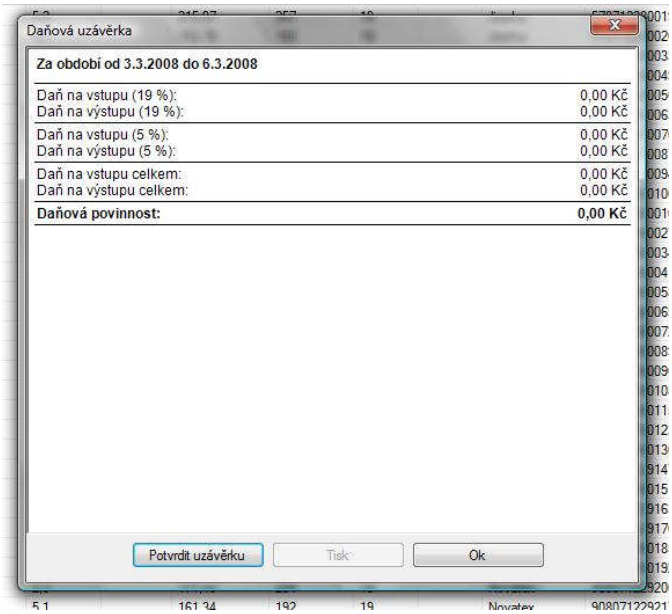
Obrázek 3 - Tvorba příjemky

3.2. Základní účetní operace

Tato funkce byla do programu implementována z toho důvodu, že uživatel, který používá skladový program, potřebuje také čas od času vystavit nějakou tu fakturu, ale protože pro malé firmy je zpravidla zbytečné pořizovat program pro skladové hospodářství a navíc ještě další program pro fakturace a účetnictví, obsahuje tento program i tuto mužnost. Současně s tvorbou faktury je také možnost nechat si automaticky vygenerovat dodací list. Dialogové okno pro tvorbu faktury má prakticky stejný vzhled jako dialogy pro příjem a výdej, je zde pouze několik nastavení navíc.

3.3. Zpracování podkladů pro přiznání DPH

Ze stejného důvodu, z jakého je v programu implementována funkce fakturace, je zde i možnost zpracování podkladů pro přiznání DPH. Pokud je v nastavení programu zatržena možnost „Plátce DPH“. Je uživateli umožněno nechat si programem spočítat daňovou povinnost se všemi dalšími náležitostmi od poslední „Uzávěrky DPH“ - tak je totiž v programu nazvána tato funkce.



Obrázek 4 - Ukázka zpracování podkladů pro přiznání DPH

3.4. Zjištění stavu skladu

Stav skladu je možné zjistit hned dvěma způsoby tím prvním je roční uzávěrka skladu, tím druhům a neméně důležitým je přehledová inventura. Každá ze dvou jmenovaných funkcí na první pohled dělá jednu a tutéž věc - zjišťuje stav skladu, ale realita je poněkud jiná. Přehledová inventura počítá se všemi zaúčtovanými pohyby za všechny účetní období (reflektuje skutečné fyzické množství položek na skladě), naopak roční uzávěrka skladu počítá s pohyby, které byly zaúčtovány pouze do daného účetního období včetně. Pokud má uživatel aktivní pouze jedno účetní období, tak je výsledek obou funkcí shodný, ale může dojít k takovému případu (a zpravidla k tomu dochází), že bude mít uživatel aktivní dvě po sobě jdoucí účetní období. Roční uzávěrka slouží k „zakonzervování“ a zjištění stavu staršího z obou období.

3.5. Propojení s registrační pokladnou

Program umožňuje propojení s registrační pokladnou typu EURO 2000TE Alpha, kterou vyrábí slovenská firma ELCOM s.r.o. Spojení s počítačem je realizováno pomocí sériového portu RS-232. Po fyzickém připojení pokladny pak stačí jenom upravit v programu nastavení pokladny a pokladna je okamžitě připravená k použití. Samozřejmostí je zaúčtování veškerých pohybů provedených pomocí pokladny do skladu (děje se tak prostřednictvím denní uzávěrky pokladny).

3.6. Tisk čárových kódů

Jako jeden z mála programů ve své kategorii podporuje tento program také tisk čárových kódů typu UPC-A na prakticky jakékoliv papírové „médium“ (od obyčejného papíru až po samolepící štítky). Stačí k tomu mít jakoukoliv laserovou nebo thermo tiskárnu. Tisk pomocí inkoustové tiskárny je také možný, ale musí být na fotopapír a navíc ještě ve vysokém rozlišení, jinak je vytištěný kód prakticky nečitelný.

Pro tisk čárového kódu musí mít uživatel ve skladu nějakou položku se zadaným čárovým kódem typu UPC-A. V dialogovém okně pro tisk čárových kódů si pak uživatel zvolí položku, zadá počet kódů k vytisknutí, nastaví parametry stránky a v náhledu označí pole, kde je možné kód vytisknout. Pak již zbývá jediné – stisknout tlačítko „Tisk“.

3.7. Zálohy a obnova dat + import / export

Každý se zcela určitě setkal s tím, že technika umí zradit v ten nejnevhodnější okamžik, a proto zde existuje funkce zálohy a obnovy dat. Data je možné zálohovat nebo obnovit pouze ve speciálním formátu. (Je to z důvodů co nejmenší velikosti zálohy.)

Funkce exportu a importu dat je v programu z poněkud odlišného důvodu. Slouží především pro přenos skladových karet a adresáře mezi různými sklady.

4. Technické pozadí

V této části je věnována pozornost nejrůznějším technickým aspektům, které běžný uživatel programu téměř vůbec nevnímá, ale přesto jsou důležité a je potřeba se o nich alespoň částečně zmínit.

4.1. Systémové požadavky

Program je primárně vyvíjen v prostředí .NET verze 3.5 v jazyku C#, tudíž je do jisté míry závislý na platformě Microsoft® Windows®, ale postupem času dojde i k portaci do prostředí GNU/Linux. Zde je přehled systémových nároků:

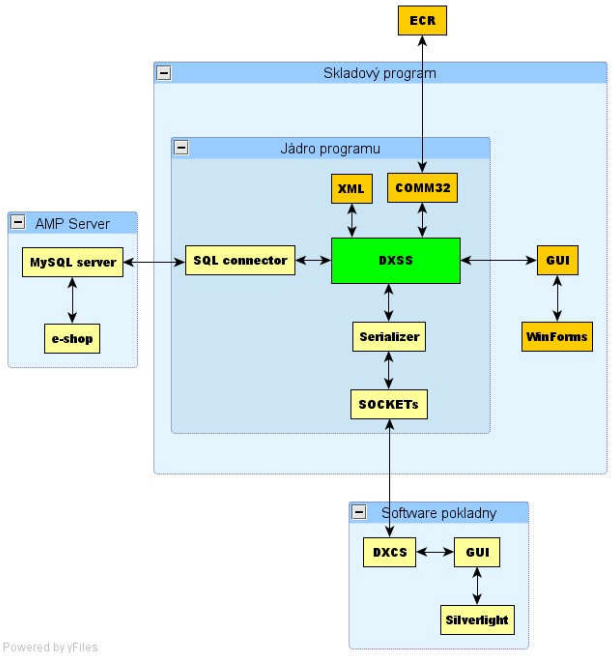
- Operační systém Microsoft® Windows® XP nebo vyšší
- Procesor Intel Pentium 4 – 2GHz nebo ekvivalentní vyšší
- Operační paměť alespoň o velikosti 512MB
- 500 MB volného místa na disku

Jak je vidět nejedná se o nijak zvlášť náročný program, tudíž s jeho během nebude mít problém žádný průměrný kancelářský počítač.

K systémovým požadavkům dále patří i volný port RS-232 (v případě, že uživatel požaduje propojení s pokladnou) nebo jakékoliv rozhraní pro připojení čtečky čárového kódu (pokud má být použita).

4.2. Programová struktura a budoucí vývoj

Programová struktura je pro slovní popis poněkud složitější, proto se ji alespoň z části pokusím nastínit následujícím diagramem.



Obrázek 5 - Funkční schéma programu

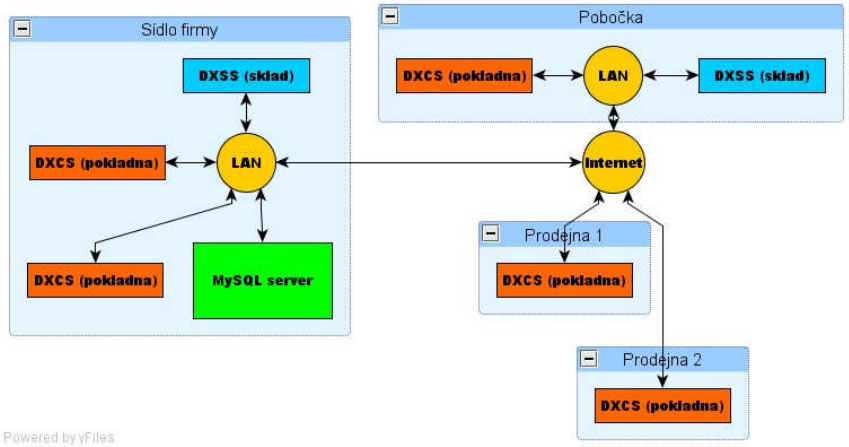
Vysvětlení zkratek

DXSS	Označení vlastních datových struktur programu (tzv. vnitřní jádro)
COMM32	Název komunikační knihovny pro komunikaci s pokladnou
ECR	Označení pokladny zavedené výrobcem (z angl. EURO Cash Register)
AMP Server	Zkratka pro Apache, MySQL, PHP server
DXCS[2]	Označení jádra pokladního softwaru

Tabulka 2 - význam zkratek použitých v diagramu

Tento diagram nepopisuje pouze aktuální vývojový stupeň programu, ale částečně také nastiňuje směr, kterým se bude rozvíjet budoucí vývoj (zelená + tmavě žlutá barva označuje již hotové části programu, světle žlutá označuje části programu, které jsou ve vývoji). Momentálně program dokáže zpracovávat data „pouze“ z registrační pokladny a z vlastních aktivit a ukládá je do databáze ve formátu XML. Později se úloha hlavní databáze přesune na MySQL server (přičemž schopnost ukládat data ve formátu XML zůstane zachována). Tato architektura dovolí propojení teoreticky neomezeného množství skladových a pokladních (neplatí pro ECR) terminálů a umožní tak centralizovanou správu skladu i pro větší firmy. Modelovou situaci ukazuje následující obrázek.

[2] Neplést s registrační pokladnou. DXCS je vlastní pokladní systém, který je prozatím ve fázi vývoje.

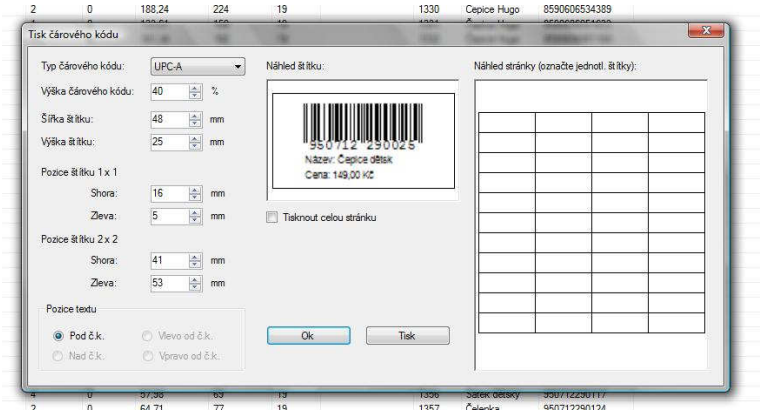


Obrázek 6 - Modelová situace použití programu

Tato modelová situace velmi pěkně ukazuje, jak lze v tomto případě využít výhod internetu. Tato pomyslná firma by přitom ani nemusela vynaložit nějaké extrémně vysoké náklady na zřízení podobného systému.

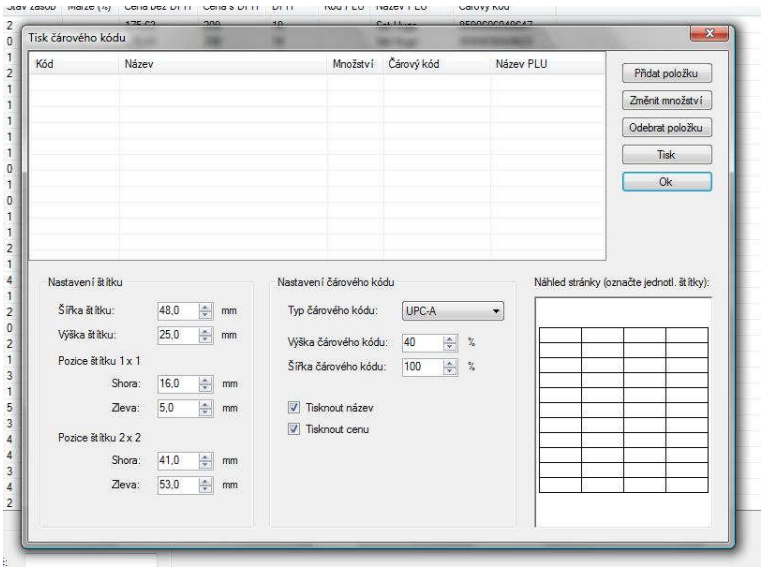
5. Závěr

Na vývoji tohoto programu pracuju již druhým rokem, neustále se učím nové technologie, získávám nové poznatky a zkušenosti, které se snažím nějakým způsobem zužitkovat a aplikovat je v tomto programu. Kupředu mě tlačí i vědomí toho, že má snaha není zbytečná, a že už teď se může tento program pyšnit ostrým nasazením v praxi, což mi velice pomáhá zlepšovat, zrychlovat a zefektivňovat práci s tímto programem. Zde je na ukázku malý příklad pokroku v průběhu vývoje.



Obrázek 7 - Původní vzhled dialogu pro tisk čárových kódů

Toto dialogové okno bylo na přání zákazníka od základu přepracováno. Důvodem byla nemožnost hromadného tisku čárových kódů u předchozí verze. Nyní již stačí jenom vložit do seznamu položky, ke kterým chce uživatel vytisknout čárový kód, zadat jejich počet a program se o zbytek postará.



Obrázek 8 - Ten samý dialog po přepracování

Literatura

1. PETR PUŠ, *Poznáváme C# a Microsoft .NET* [online] [cit. 6. Března 2007] dostupné na internetu <http://poznavame-c-msnet.wz.cz/>
2. ELCOM SPOL. S R.O., *Dokumentace a technická podpora komunikační knihovny k EURO pokladnám* [online] [cit. 6. března 2007] dostupné na internetu <http://elcom.sk/cz/podpora_cz.php>

Středoškolská technika 2008

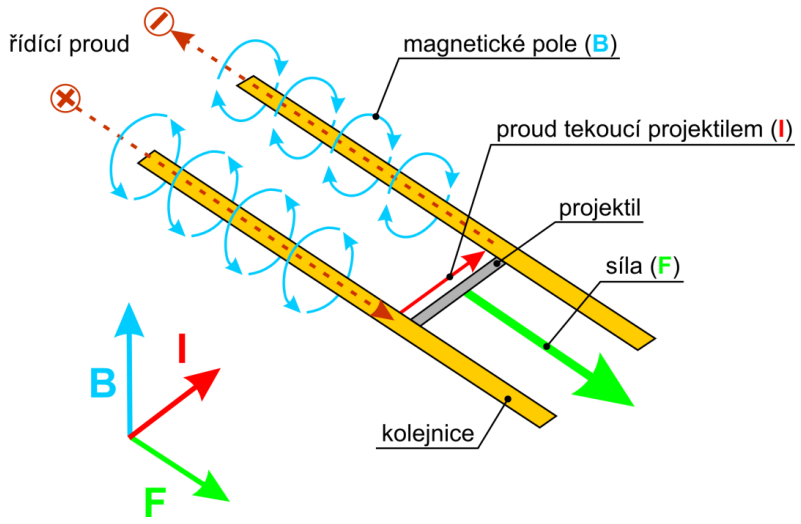
Setkání a prezentace prací
středoškolských studentů na ČVUT

Elektromagnetický urychlovač feromagnetických projektilů

Radim Laga

Gymnázium J.A. Komenského a Jazyková škola s právem státní jazykové zkoušky Uherský Brod
Komenského 169, Uherský Brod

Elektromagnetické urychlovače můžeme primárně rozdělit do dvou skupin. První typ je kolejnicový urychlovač, který využívá vodiče přímo jako projektilu. Celý systém tvoří jakýsi tři-čtvrtě závit, kde kolejnice po stranách jsou pevně upevněny a uprostřed se pohybuje vodivý projektil. Tento je pak magnetickým polem vytlačován ven z kolejnic obrovskou rychlostí. Základní zobrazení tohoto zařízení je na obr. 1. Silnou nevýhodou tohoto zařízení je značné opalování vnitřních stěn kolejnic a tím i malá životnost.



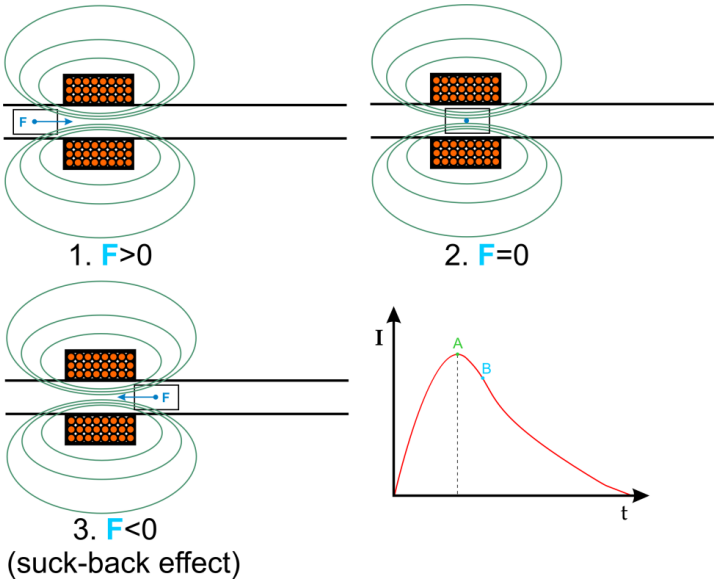
Obr. 1: Principiální znázornění kolejnicového urychlovače (railgun)

Pokud uvažujeme situaci, že v celém zařízení jsou vektor proudu a magnetické indukce na sebe vzájemně kolmé, pak výsledná síla **F** bude na tyto dva vektory taktéž kolmá. Protože uvažujeme, že všechny veličiny jsou na sebe kolmé, lze vztah pro velikost síly F zapsat jako:

$$F = B\,I\,l$$

Kde B je velikost magnetické indukce, I velikost proudu a l je aktivní délka projektilu.

Druhým typem je cívkový urychlovač neboli coilgun. Je to soustava cívky a magnetického projektilu, nejčastěji ocelového. Projektil je v počáteční poloze vsunut na okraj cívky, do které se pak pustí velké množství energie. Krátkodobé magnetické pole o obrovské intenzitě (ve špičce okolo 9T) přitáhne projektil dovnitř cívky. Ten pak už dál letí svou vlastní setrvačností. Pokud ovšem magnetické pole uvnitř cívky přetrvává, tak je projektil zpátky vtahován dovnitř, resp. brzděn. Jednotlivé fáze znázorňuje obr. 2.



Obr. 2: Znázornění fází cívkového urychlovače (coilgun)

Naskýtají se ovšem otázky: tloušťka drátu, délka cívky, počet závitů…
Na tyto otázky lze jen částečně odpovědět. Např. průměr drátu se musí volit tak, aby měl při více závitech malý odpor ale aby zároveň vzdálenost mezi vrchními vrstvami cívky a středem hlavně byla co nejmenší, protože magnetická indukce slábne s druhou mocninou vzdálenosti:

$$\mathbf{B}(r) = -\frac{1}{4\pi\epsilon_0 c^2}\int \frac{I\mathbf{e}\times d\mathbf{s}}{r^2}$$

Naneštěstí celý systém je analyticky neřešitelný, protože se zde musí uvažovat změna indukčnosti cívky při průletu projektilu a změna jeho nasycení vzhledem k hysterezí křivce použitého materiálu.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Albireo, první český svobodný redakční systém

Martin Novák, Petr Kunc

Gymnázium Uničov
Gymnazijní 257, Uničov

Naším cílem bylo vytvořit redakční systém (serverový software určený k usnadnění tvorby a správy internetových stránek) šířený pod svobodnou licencí, který bude opravdu jednoduše ovladatelný pro každého.
V mnoha částech používáme vlastní originální řešení a popíráme dlouho zaběhnuté trendy ve vývoji redakčních systémů. Klasická instalace po uživateli vyžaduje přenos desítek až stovek souborů a orientaci ve složité adresářové struktuře. Albireo má jediný instalační soubor velký jako prázdný dokument v MS Word. Celý instalační proces probíhá v plně interaktivním webovém rozhraní a je tak jednoduchý, že jej zvládne každý i bez technických znalostí. S adresářovou strukturou nepracujete vůbec.
Kompletní správa webu následně probíhá také už jen v protředí vašeho webového prohlížeče, ať už děláte cokoli. S výhodou využíváme technologie Webu 2.0, které nám umožňují vytvořit impresi desktopových aplikací. Proto se soubory pracujete ve správci složek fungujícím na principu klasického průzkumníka z vašeho operačního systému. Nepřicházíte ani o dialogová okna.
Psaní nových článků pro web po vás nebude vyžadovat ani znalost základů HTML, je připraveno WYSIWYG rozhraní, které se používá podobně jako MS Word. Dokonce fungují i známé klávesové zkratky.
Svoboda Albirea daná licencí GNU/AGPL představuje v České republice v oblasti redakčních systému naprostý unikát. Kompletně celý software je možné legálně šířit, kopírovat i upravovat zdarma a bez limitů. Nečekejte žádnou zradu ani v podobě reklamy. Všichni uživatelé mají prakticky stejná práva jako samotní autoři.
Vychudnejte si pocit luxusu zdarma a bez limitů.



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Optické pojítko RONJA

Jakub Petrák, Radek Šembera
Střední průmyslová škola elektrotechnická,
V Úžlabině 320, Praha 10

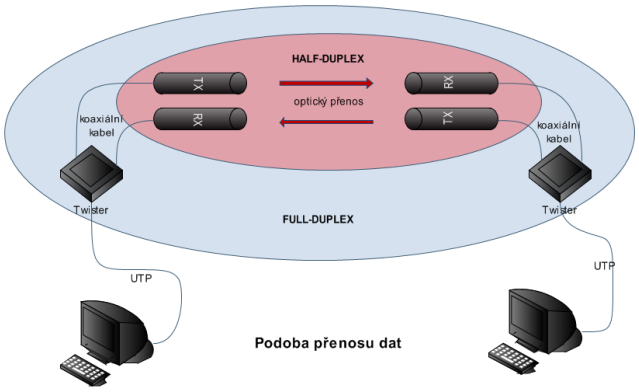
Ronja (Reasonable Optical Near Joint Access)

RONJA, nebo-li optické pojítko, je zařízení sloužící k propojení dvou míst bezdrátovou optickou cestou, pracující ve viditelném světelném spektru. Každá strana je opatřena přijímačem (Receiver) a vysílačem (Transmitter). Z vnějšího pohledu můžeme přenos označit duplexním (přijímání i vysílání dat může probíhat ve stejný okamžik). Vysílač je nejčastěji realizovaný prostřednictvím LED diody, jejíž maximální přenosová frekvence pravoúhlého průběhu dosahuje hodnoty okolo 10 MHz/s, což odpovídá přenosové rychlosti dat 10 Mb/s. Pro zvýšení přenosové rychlosti lze teoreticky použít laserové diody, s jejíž pomocí lze uskutečnit přenos dat dosahující hodnoty až 100 Mb/s. Přijímač je realizován nejčastěji fotodiodou.

Zásadní výhodou pojítka oproti alternativám (např. WiFi) je možnost realizace spoje až na vzdálenost 1 500m v případě použití laserové diody. Další výhodou, kterou RONJA disponuje, je využití úzce směrového světelného paprsku při přenosu dat a tím i zamezení téměř jakékoliv možnosti odposlechu vysílaného signálu. Oproti mikrovlnkou rušitelné WiFi je RONJA prakticky 100% odolná proti rušení vnějších vlivů, které úzce nesouvisí s extrémním zhoršením propustnosti prostředí mezi dvěma pojítky.

Jedna z nevýhod optického pojítka spočívá v nutnosti realizace přímé viditelnosti mezi oběma stanicemi, která může být zhoršena výskytem intenzivních dešťových a sněhových srážek či hustou mlhou. Z tohoto důvodu je nanejvýše vhodné realizovat, vedle optické cesty, alternativní datový spoj pracující pouze při výpadku pojítka.

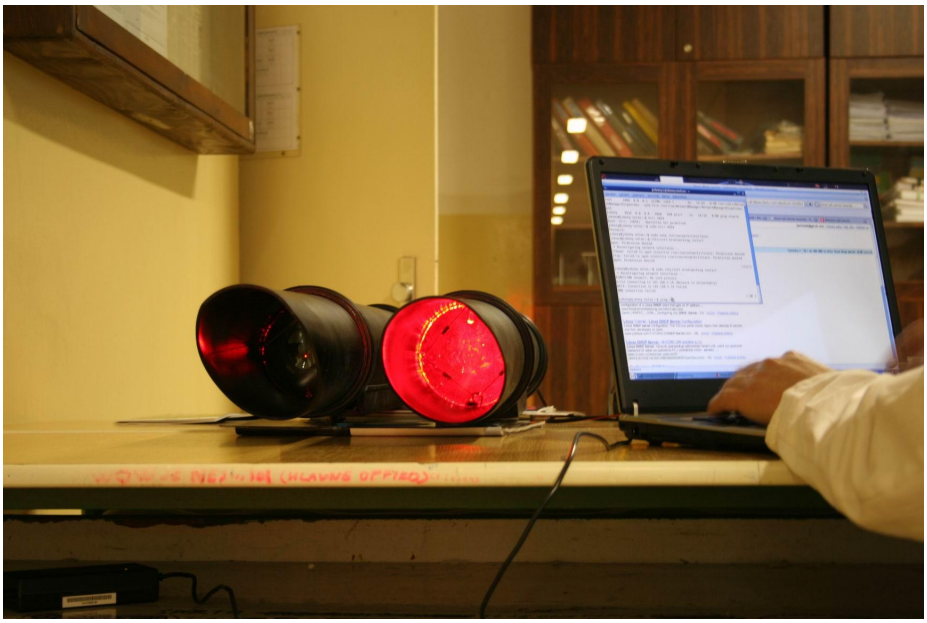
Pořizovací cena jednoho páru přijímač, vysílač a twister se pohybuje v řádu několika tisíc korun.



Obr. 1: Schématické znázornění přenosu



Obr. 2: Realizované spojení



Obr. 3: Pohled do trubic

Středoškolská technika 2008

Setkání a prezentace prací středoškolských studentů na ČVUT

DIGITÁLNÍ TEPLOMĚR S PIC

Jan Hanuš

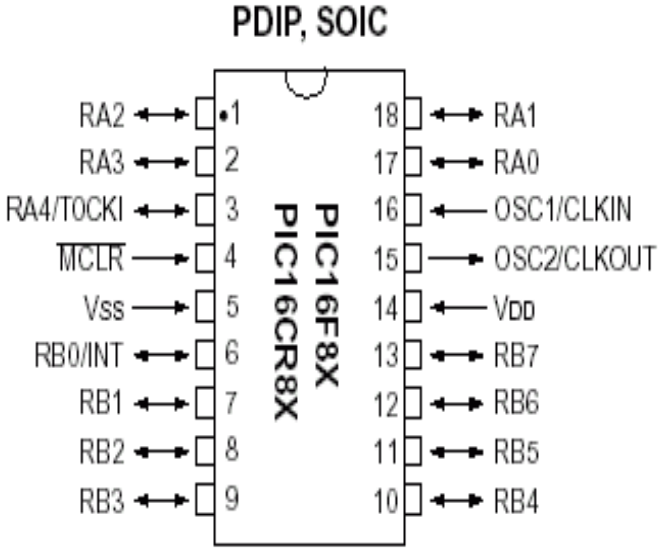
Střední průmyslová škola elektrotechnická,
V Úžlabině 320, Praha 10

Práce se zabývá zhotovením digitálního teploměru s mikročipem PIC16F84. Pomocí přepínače lze na LED displeji zobrazit aktuální teplotu, max. nebo min. teplotu za posledních 6 hodin. Celý obvod je napájen 9V baterií.

Mikroprocesor PIC 16F84

Všechny procesory PIC jsou typu RISC, tedy procesory s omezenou instrukční sadou. To znamená, že je k dispozici pouze několik málo příkazů (35), jejichž zpracování trvá pouze jeden (maximálně dva) strojové takty a v paměti zabírají vždy po jednom slově. Jeden strojový takt je rychlost, se kterou procesor zpracovává příkazy a odvíjí se od kmitočtu oscilátoru jeho vydělením 4. Je-li například použit krystal 4MHz (jako v našem případě), můžeme si lehce spočíst, že jeden takt trvá 1us. Během tohoto jednoho taktu procesor provede jednu instrukci a zároveň si připraví tu následující. Většina instrukcí proto trvá přesně jeden takt. Pouze instrukce, které naruší plynulý běh programu (skokové instrukce) trvají dva takty, protože procesor nevyužije připravenou instrukci a musí načítat další.

Rozložení vývodů procesoru



Popis vývodů

RA0 - RA4 - vývody portu A (5)
RB0 - RB7 - vývody portu A (8)
MCLR - reset, stačí připojit přímo na +Ucc (procesor se resetuje sám)
Vss - zem napájecího napětí
Vdd - plus napájecího napětí (2 - 6 V)
OSC1, OSC2 - vývody na připojení oscilátoru

Popis postupu zhotovení

- návrh tištěného spoje v programu Sprint Layout
- vyleptání desky plošných spojů (dále jen DPS) fotocestou o rozměrech 75x130 mm
- vyvrtání a osazení DPS
- ověření funkčnosti
- odstranění problémů při nefunkčnosti
- mechanická úprava

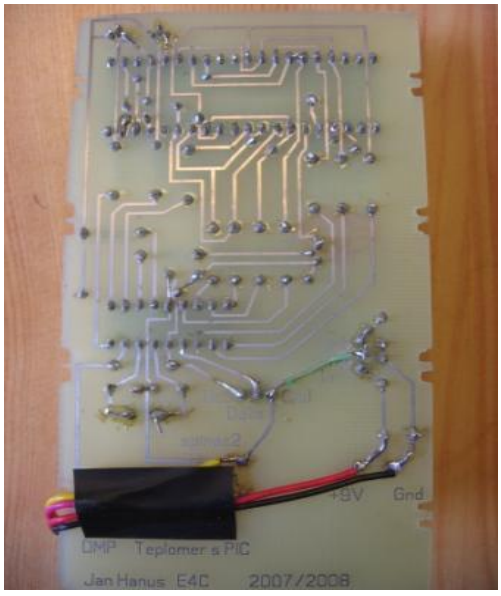
Popis činnosti

Obvod je napájen 9V baterií. Stabilizátor však napětí stabilizuje na 5V. Hlavní částí celého obvodu je mikroprocesor PIC 16F84. Tomu zajišťuje hodinový kmitočet obvod: Q- C1- C2. Jako snímač teploty je použito čidlo SMT 160-30, které komunikuje s mikroprocesorem pomocí pulzně šířkové modulace. V praxi to znamená, že snímaná teplota se mění na frekvenci a při každé další změně teploty dochází ke změně šířky pulsu nosného signálu. Tranzistory spínají kladné napětí a rozsvěcují jednotlivé číslice LED displeje. Jsou tři, protože chceme rozsvítit 3 číslice. Na dvou jedno a půl místných displejích se teplota do 19,9 stupňů zobrazuje na jedno desetinné místo a od 20 stupňů výše se teplota zaokrouhlí na jednotky. Na výstupech mikroprocesoru (1, 2, 6, 7, 8, 9, 10, 18) se zobrazují jednotlivé segmenty čísla (A - H).

Spínač přepíná aktuální teplotu na - min. teplotu za posledních 6 hodin
- max. teplotu za posledních 6 hodin
(Počítá se od doby zapnutí, tj. od připojení napájení.)



Obr. 1 : Osazená DPS (vrchní strana) o rozměrech 75 x 130 mm



Obr. 2 : Spodní strana osazené DPS

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

NÁHODNÉ GENERÁTORY

Jan Kubák

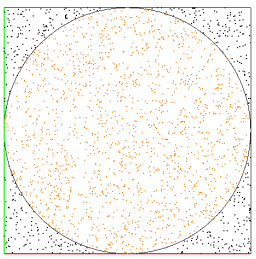
Střední průmyslová škola elektrotechnická,
V Úžlabině 320, Praha 10

Kladli jste si někdy otázku, jak je možné v počítači vygenerovat náhodné číslo? Počítač je přeci logický stroj pracující s výrokovou (binární) logikou. Existují pouze dvě možnosti, pravda, či nepravda. S těmito stavy počítač pracuje pomocí logických kombinací, jejichž výsledkem je opět pravda, či nepravda, přičemž vždy musí být známý vstup. Jak je tedy možné z takto pracujícího stroje „vytěžit“ náhodné číslo? Řešením je pseudonáhodný generátor.

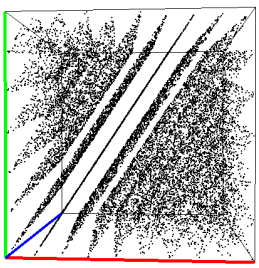
Pseudonáhodný generátor je program, který generuje pseudonáhodná čísla na základě vzorce pro množinu v rekurzivním tvaru. ($a_{n+1}=a_n B+C$). Kouzlo se skrývá právě ve skladbě tohoto vzorce, dle kterého se vyvíjí celá množina pseudonáhodných čísel. Dalším faktorem je počáteční hodnota posloupnosti, tzv. seed pseudonáhodného generátoru. Seed je často opravdu náhodný - random seed. V praxi se na PC získává např. systémovým časem, počtem proběhnutých cyklů procesoru od spuštění procesu, intervaly mezi stisky kláves uživatelem či aplikací připraveného algoritmu na pohyb myši na obrazovce.

Opravdu náhodné číslo lze získat díky skutečně náhodnému generátoru, ty jsou také teoreticky popsány v mé maturitní práci. Práce ale hlavně obsahuje jednotlivé typy pseudonáhodných generátorů, jejich popis, programové provedení a test. To je realizováno pomocí programu, který umožňuje otestování jejich výstupů pomocí zobrazení vygenerovaných čísel jako bodů v ploše či v prostoru. V grafickém výstupu jsou patrné různé nedokonalosti a kazy. Program dále testuje rovnoměrnost rozložení vygenerovaných bodů v prostoru a v ploše. Pro ilustraci si uveďme několik ukázek:

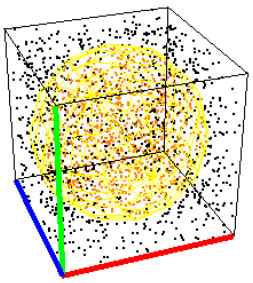
Obr 1.



Obr 2.



Obr 3.



- Obr 1.: Zřetelně špatný typ pseudonáhodného generátoru RANDU.
- Obr 2.: Test rozložení vygenerovaných bodů v ploše; generátor používaný Pascalem má chybu 0,92%.
- Obr 3.: Test rozložení vygenerovaných bodů v prostoru; generátor používaný Pascalem má chybu 3,29%.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

NF výkonový zesilovač 2x150W/8Ω s obvody LM4780

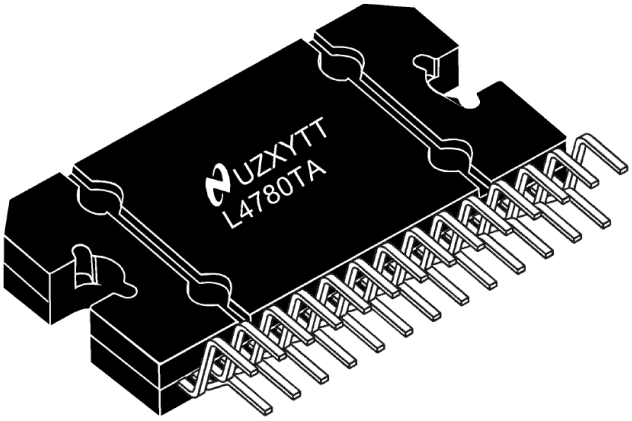
Oleksandr Kyrychenko

Střední průmyslová škola elektrotechnická,
V Úžlabině 320, Praha 10

Obvody LM4780TA zapojené do můstku, použité v této konstrukci stereofonního zesilovače, dosahují základních parametrů: 75W na kanál, pro 4Ω zátěž s maximálním zkreslením 0,5% THD v kmitočtovém rozsahu 20 – 20000Hz. Protože zesilovač je konstruován v můstkovém zapojení, dosahuje výstupní výkon hodnoty 120 – 150W.

Obvod LM4780TA je plně chráněn proti tepelnému a proudovému přetížení vnitřním obvodem, nazvaným od výrobce obvodu – „SPike". Obvod „SPike" dále dynamicky optimalizuje bezpečnou pracovní oblast koncových výkonových tranzistorů obvodu LM4780TA, navíc funkce „SPike" hlídá přepětí a podpětí na jednotlivých vstupech obvodu, přetížení a zkrat na výstupu, zkrat proti napájecí větvi a zemi (GND), teplotní přetížení a impulsní dynamické přetížení obvodu. Všechny tyto funkce obvodu „SPike" zařazuje výkonový integrovaný zesilovač LM4780TA mezi špičkové zesilovače sestavené pouze z diskrétních součástek.

I když obvod LM4780TA není zatím tak příliš známý, jako jsou obvody řady TDA, je od výrobce obvodu firmy National Semiconductor předurčen pro velice kvalitní NF aplikace. V samotném katalogovém listu stojí, že tento obvod je určen pro HIGH-END a HD TV techniku. Cena tohoto obvodu na našem trhu je přibližně 250,-Kč, ovšem cena skutečně odpovídá špičkovým parametrům, jaké obvod LM4780TA dosahuje.



Obr. 1: Obvod LM4780

Technické parametry LM4780:

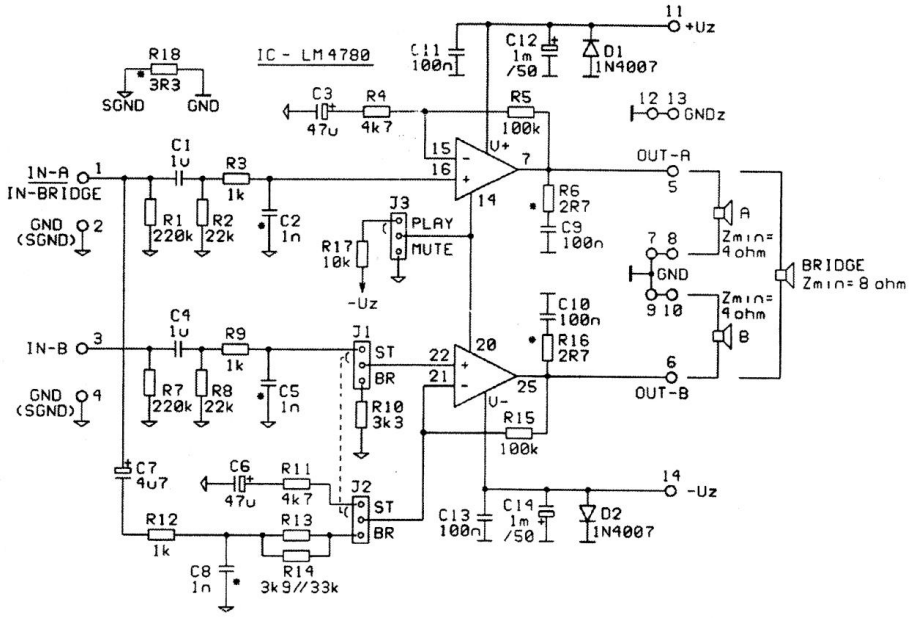
Max. napájecí napětí:	± 20V až ± 42V
Osazení:	2 x LM4780
Výstupní výkon:	4 x 75W (4Ω) 2 x 150W (8Ω)
Proudový odběr:	7.2A (4 x 75W)
Kmitočtová charakteristika:	10Hz až 100kHz
Zkreslení:	0.02% (1kHz)
Klidový odběr:	240mA (±35V)
Vstupní citlivost:	1000mV
Minimální zatěžovací impedance:	4Ω (stereo), 8Ω (můstek)
Vstupní impedance:	22kΩ (stereo), 4kΩ (můstek)

Tento projekt se skládá ze tří části:

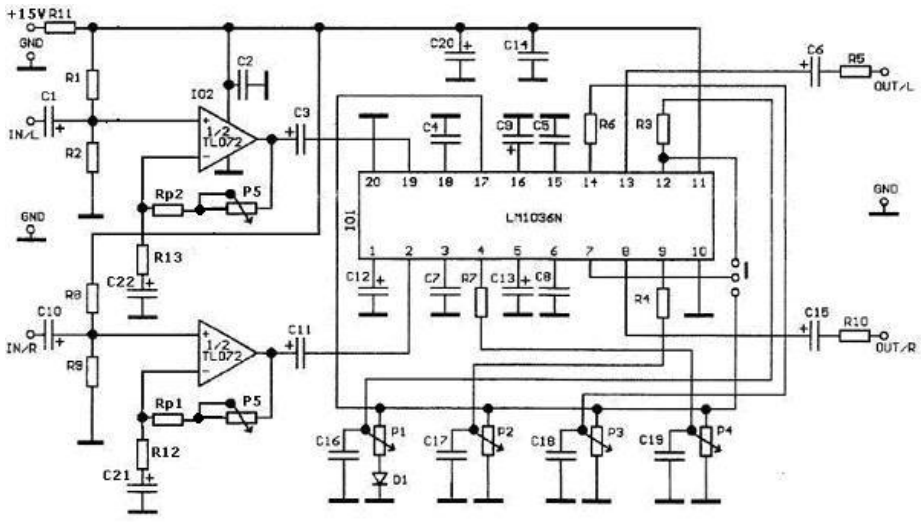
První část je již ze zmiňovaného koncového zesilovače LM4780. Na obr. 2 vidíme jeho zapojení, kde si můžeme všimnout funkce MUTE a také přepínání mezi STEREO a BRIDGE.

Druhou část tvoří korekční předzesilovač ve třídě Hi-Fi. Tento korekční předzesilovač umožňuje regulovat hlasitost, stereováhu, hloubky a výšky. Kromě toho si můžeme zapnout fyziologickou regulaci hlasitosti – zvýraznění basů a výšek při malé hlasitosti reprodukce. Na obr. 3 je jeho schéma zapojení.

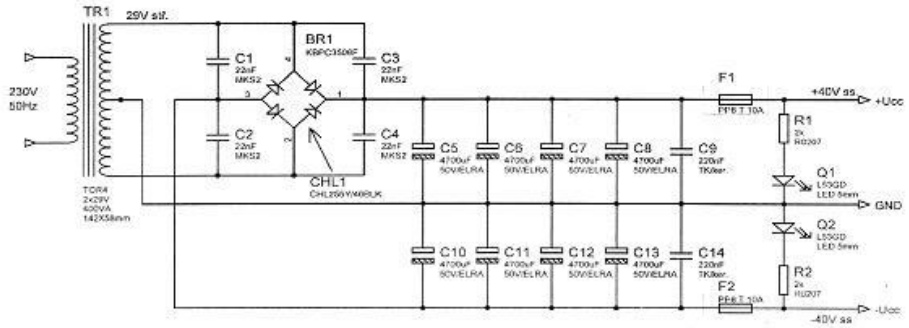
Třetí část potřebujeme pro napájení celého zesilovače. Pokud očekáváme od zesilovače výstupní výkon 2x150W, musíme navrhnout zdroj s výkonem alespoň 400W. V této konstrukci je použit toroidní transformátor o výkonu 400VA, se sekundárním napětím 2x29V/2x6,9A. Za transformátorem je zařazen usměrňovací můstek KBPC3506F (600V; 35A). Za diodovým můstkem je zařazena filtrační kapacita vytvořená z elektrolytických kondenzátorů. Doporučuje se na každý protékající ampér kapacita 1000µF, ovšem pro spolehlivou výkonovou rezervu zesilovače je doporučeno tuto kapacitu zdvojnásobit. Pro lepší vlastnosti filtrační „baterie“ je vhodné rozložit kapacitu do čtveřice kondenzátorů s hodnotou 4x4700µF pro každou napájecí větev, snížíme tím vnitřní odpor filtrační „baterie“, která dokáže do zesilovače dodávat daleko větší impulsní proudy (pro basové pasáže velké plus). Celé zapojení napájecího zdroje je na obr. 4.



Obr. 2: Schéma zapojení koncového zesilovače

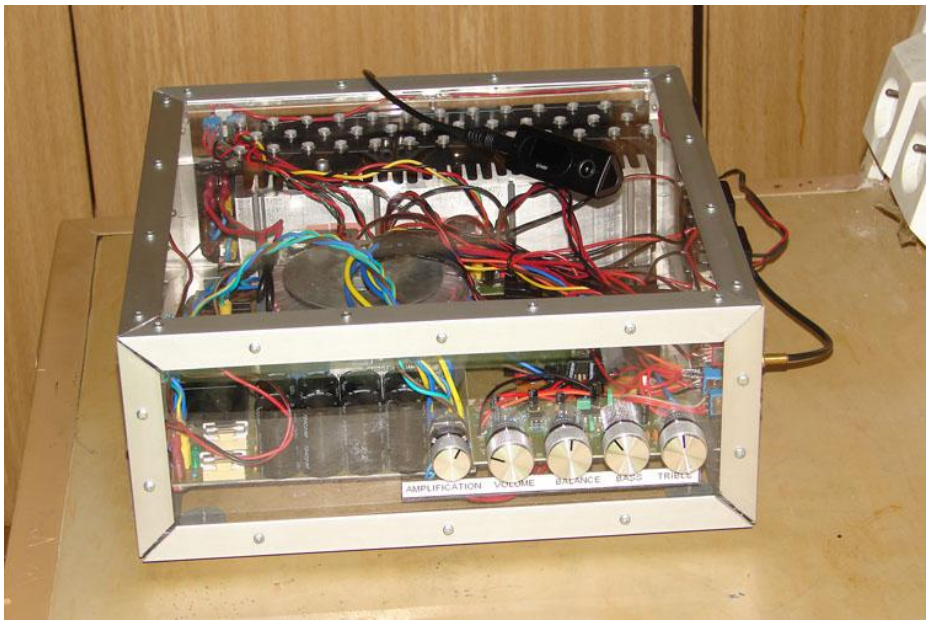


Obr. 3: Korekční předzesilovač

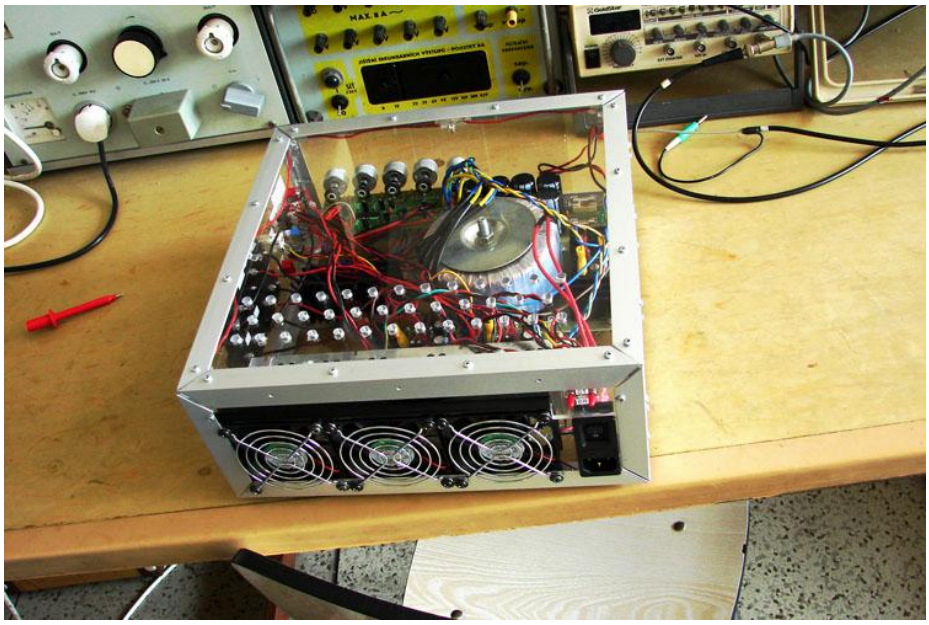


Obr. 4: Napájecí zdroj

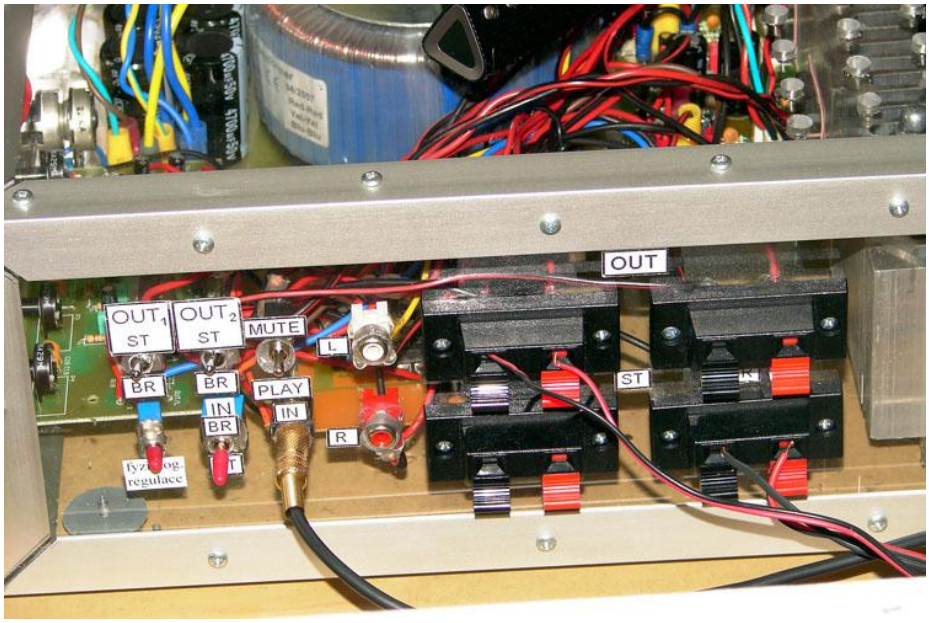
Fotografie finální podoby zesilovače:



Obr. 5: Korekce zesilovače



Obr. 6: Aktivní chlazení a přívod napětí ze sítě



Obr. 7: In/Out zesilovače

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Logická hra Sokoban

Marek Slúka

Střední průmyslová škola elektrotechnická,
V Úžlabině 320, Praha 10

První hru Sokoban vytvořil v roce 1980 Japonec Hiroyuki Imabayashi v rámci soutěže vyhlášené vedením Takarazukých skladů pro své zaměstnance, kdy měli soutěžící za úkol vytvořit motivující software pro volné chvíle zaměstnanců. Vítězem se stala na první pohled nenápadná logická hra „Sokoban“. Její název v českém překladu znamená skladník a cílem hry bylo přesunout bedny ve skladišti na určená místa. Vydána byla tato hra až o dva roky později společností Thinking Rabbit, jejímž ředitelem byl právě Imabayashi. Do Evropy se Sokoban dostal roku 1984, kdy byl vydán společností Spectrum Holobyte. Od té doby vznikla celá řada her na toto téma, ať už některé zmutované verze, nebo klasické.



Obr. 1: Náhled na uvítací obrazovku

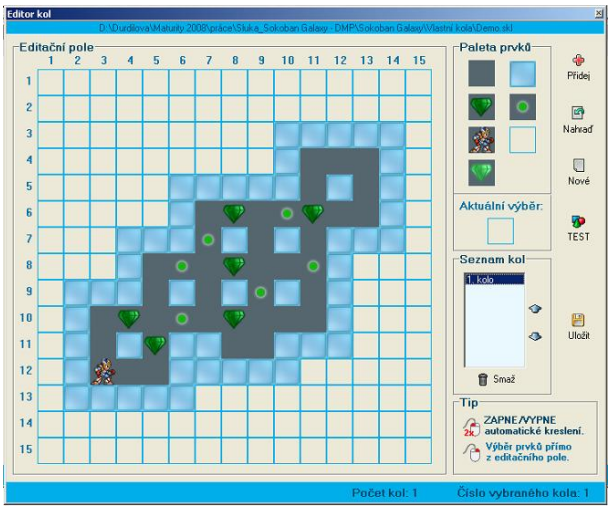
Po spuštění mé aplikace se objeví uvítací obrazovka a poté je již možné začít hru hrát. V základním souboru je uloženo 28 kol s různým stupněm obtížnosti, mezi kterými je možno vybírat. Během hry se jednak počítá počet posunů hrací postavičkou, jednak čas. Nejlepší

dosažené výsledky v každém kole jsou ukládány do souboru na disku k informacím o daném kole. Hra je obohacena o zvukové efekty a jednoduchou animaci.



Obr. 2: Náhled na hrací plán (hlavní okno programu)

Pomocí vestavěného editoru má hráč možnost stávající soubor s koly upravovat nebo si vytvořit vlastní soubor s vlastními koly.



Obr. 3: Editor kol

Svoji aplikaci jsem naprogramoval v programovacím jazyce Object Pascal ve vývojovém prostředí Delphi 7 Personal a využil jsem výhod objektového programování. Aplikace se skládá z několika souborů se zdrojovými texty:

Soubor	**Popis**
sokoban.dpr	zdrojový text programu
sokoban_F_Hlavni.pas	unita hlavního okna aplikace
sokoban_F_Editor.pas	unita editoru kol
sokoban_F_Uvod.pas	unita úvodního okna aplikace.
sokoban_F_Nastaveni.pas	unita okna s nastavením
sokoban_F_NovyRek.pas	unita okna, které se zobrazuje při dosažení rekordu kola.
sokoban_F_HratVK.pas	unita okno pro možnost hraní vlastních kol
sokoban_F_OProgramu.pas	unita informační okno „O programu“
sokoban_F_StartLogo.pas	unita startovacího okna, které se zobrazuje na samém počátku spouštění aplikace.
Sokoban.res	soubor se zdroji (např. ikona aplikace)
DialogZpravy.pas	pomocná unita sloužící pro zobrazování dialogů zpráv
Data.dat	soubor obsahující informace o kolech
Nastaveni.dat	soubor se základním nastavením aplikace

V aplikaci jsou dále využívány soubory s nápovědou uložené ve složce „nápověda“, soubory se zvuky ve formátu „wav“ uložené ve složce „zvuky“ a soubory s vlastními koly jsou implicitně ukládány do složky „vlastní kola“.

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

NF výkonový zesilovač 2x200W/4Ω

Marie Tomečková

Střední průmyslová škola elektrotechnická,
V Úžlabině 320, Praha 10

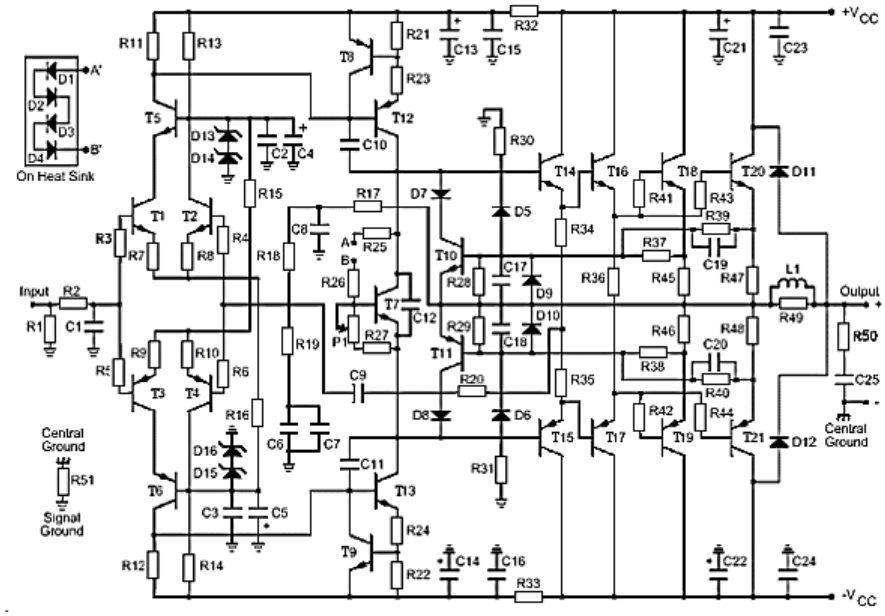
Při pořádání konferencí a různých firemních setkání svých přátel jsem zjistila potřebu kvalitního ozvučení uvedených prostor. Proto jsem si pro dlouhodobou praktickou maturitní práci zvolila konstrukci výkonového zesilovače s výkonem okolo 2x200W. S přihlédnutím k tomu, že v uvedených případech je kladen důraz na kvalitu zvuku, zvolila jsem v současné době ne příliš často používané zapojení pro jeho vyšší pracnost a cenu.

Jedná se o dvojici výkonových stupňů s řešením z diskrétních součástek. Zesilovač je moderně koncipován, zcela symetrický, podobně jako současné operační zesilovače. Používá rychlé a spolehlivé polovodičové součástky. Jako budicí a výkonové tranzistory jsou zde použity tranzistory bipolárního typu ON Semiconductor MJ 15003 (NPN) a MJ 15004 (PNP) umístěné přímo na chladiči. Koncový zesilovač je středně složitý a jeho parametry jsou velmi dobré. Především úspěšně konkuruje rychlostí přeběhu, výkonovou šířkou pásma a zkreslením signálu. Vstupní citlivost zesilovače pro plné vybuzení je standardní, 1,5 V, a lze ji v menším rozmezí měnit změnou odporu rezistoru R19 ve smyčce záporné zpětné vazby bez nebezpečí vzniku oscilací. Teplotní stabilita klidových proudů je zajištěna regulací klidového proudu pomocí teplotní změny napětí na diodách umístěných v otvorech chladiče s dobrým teplotním kontaktem. Oddělení od sítě nn je řešeno převodním transformátorem v toroidním provedení, a to jak z důvodu nežádoucího vyzařování, tak i většího výkonu při úspoře váhy. Byl použit transformátor s napětím sekundárního vinutí 2x 45V s výkonem 500VA a filtrační kondenzátory 10 000 μF /100V pro malé zvlnění napájecího napětí.

Tab. 1: Technické parametry zesilovače

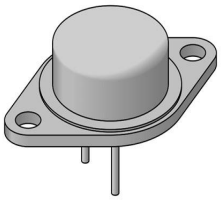
Parametr	Hodnota
Napájecí napětí :	± 45V (max. 50V)
Výstupní výkon :	200 W /4 Ω
Vstupní citlivost :	1,5 V pro P=200W/4 Ω
THD:	při f = 20 Hz.....0,007 % při f = 1 kHz.....0,02% při f = 20 kHz.....0,3%
Výkonová šířka pásma:	5 Hz – 25 kHz (v toler. pásmu -3dB)

Obr. 1: Schéma zapojení koncového stupně



Tab. 2: Koncové tranzistory MJ 15003 (MJ 15004) v pouzdře TO3 (T_{18} - T_{21})

Parametr	Hodnota	Jednotka
Collector-Emitter Voltage	140 (-140)	V
Collector-Base Voltage	140 (-140)	V
Emitter-Base Voltage	5 (-5)	V
Collector Current-Continuous	20 (-20)	A
Total Power Dissipation	250	W
Temperature Range	-65 to + 200	°C





Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Střední průmyslová škola elektrotechnická
a Zařízení pro další vzdělávání pedagogických pracovníků v Žatci

Martin Gába

Školní elektronické hodiny s mikrokontrolerem a možností kontroly zvonění

Obsah:

Obsah -
Anotace -

Teoretická příprava -
- **Hardware-**
  - **Mikroprocesor 8051**
    - **Časování Mikroprocesoru**
    - **Nulování Mikroprocesoru**
    - **Vstupně - výstupní porty**
  - **LCD displej MC1602E-SYL/H**
- **Software-**
  - **Programování mikropočítače**

Vlastní konstrukce-
- **Hardware-**
  - **Mikroprocesor8052**
  - **Časování mikroprocesoru**
  - **Nulování mikroprocesoru**
  - **Zapojení jednotlivých portů**
  - **Programátor**
  - **Výstup signálu pro zvonění**
  - **Připojení LCD displeje**
  - **Napájení 230/12V**
  - **Ovládání LCD displeje**
- **Software-**
  - **Nastavení timeru T2**
  - **Pole párů zvonění**
  - **Funkce přičítání času**
  - **Výpis na LCD displej**
  - **Funkce stisknutí tlačítka**
  - **Funkce stisků tlačítek**
  - **Funkce pro vyhledávání zvonění**

Přílohy:
- **Příloha č.1 - Kompletní program**
- **Příloha č.2 - Návrh ze strany součástek**
- **Příloha č.3 - Návrh ze strany kontaktů**
- **Příloha č.4 - Tlačítka ze strany součástek**
- **Příloha č.5 - Tlačítka ze strany kontaktů**
- **Příloha č.6 - Fotodokumentace**
- **Příloha č.7 - Ekonomický rozbor**

Použitá literatura

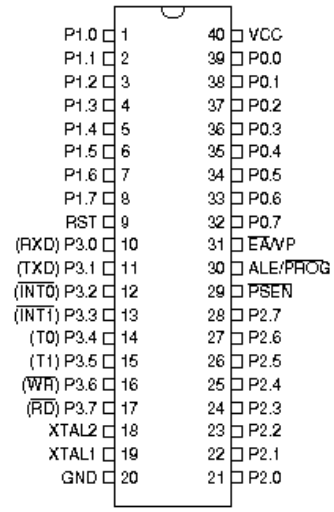
Anotace:

Cílem této práce je sestrojení funkčních školních elektronických hodin za použití mikrokontroleru typu 8051 a prohloubení znalostí této problematiky. V první části se práce zabývá teoretickou přípravou použitých součástek a jejich nastavením. A ve druhé části vlastní tvorbou a konstrukcí celého elektronického zařízení, napájení a ovládání elektronických hodin.

Tuto práci jsme zpracovávali v dvojici. Spolupracovníkem byl Miroslav Jandík. Při práci jsem se zabýval jak hardwarovou tak softwarovou záležitostí.

Zvláště bych chtěl poděkovat za pomoc při programování a oživování projektu panu Miroslavu Mikušovi.

Rozmístění pinů:

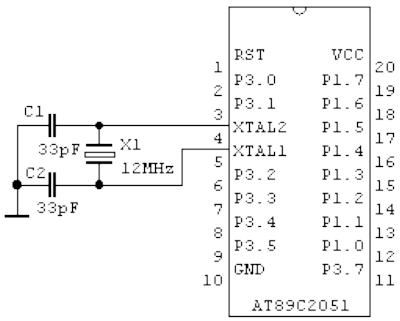


Časování Mikroprocesoru

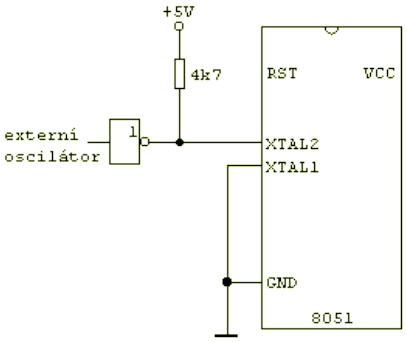
Mikroprocesor ke své činnosti a správné funkci potřebuje nějaký zdroj hodinových (taktovacích) impulzů o určité frekvenci (f_{osc}), které využívájí vnitřní obvody mikroprocesoru a od kterého je pak odvozena délka trvání strojového cyklu, a tedy i doba vykonávání instrukcí. K taktování (někdy se říká také synchronizaci) mikroprocesoru může využít buď vnitřní obvodu oscilátoru, který zabudován přímo v čipu mikroprocesoru nebo k taktování můžeme použít externí obvod oscilátoru. V prvním případě musíme k vývodům XTAL1 a XTAL2 připojit krystal dle obr.1, v druhém případě vývod XTAL1 uzemníme a na vývod XTAL2 přes hradlo TTL připojíme výstup externího obvodu oscilátoru - viz obr.2. U druhého případu je nutné uvést, že u každého typu mikroprocesoru se může způsob připojení signálu z externího oscilátoru lišit. Zapojení dle obr.2 platí pro původní verzi fy Intel v provedení HMOS. V praxi se nejvíce používá zapojení s vnitřním oscilátorem - s připojením krystalu. Když chcete použít druhý případ - externí oscilátor - můžete k tomu využít integrované oscilátory v pouzdrech DIL8 nebo DIL16 např. od firmy MOTOROLA. Pokud tedy k taktování mikroprocesoru použijeme první případ, platí jednoduchá rovnice: frekvence použitého krystalu = frekvence oscilátoru (f_{osc}). V druhém případě je situace jasná: frekvence externího oscilátoru = frekvence oscilátoru (f_{osc}). Obvod oscilátoru je ve vnitřním blokovém schématu mikroprocesoru pojmenován jako Generátor hodin.

Zdrojem taktovacích impulzů může být buď piezokeramická jednotka připojená k mikroprocesoru nebo jiný externí zdroj taktovacích impulzů. Piezokeramickou jednotku představuje krystal připojený přes vývody XTALl a XTAL2 - viz obr.1. V případě použití vnějšího (externího) zdroje taktovacích impulzů, je signál z tohoto zdroje připojen na vývod XTAL2 při současném připojení vývodu XTAL1 na zem - viz obr.2. Frekvence taktovacího signálu určuje frekvenci oscilátoru (kterou dále v našem povídání označujeme f_{osc}), platí jednoduchý vztah - frekvence použitého krystalu (taktovacího signálu) = frekvence oscilátoru (f_{osc}). Obvod, který zpracovává signál od zdroje taktovacích impulzů a rozvádí jej k dalším vnitřním obvodům mikroprocesoru, je ve vnitřním blokovém schématu mikroprocesoru pojmenován jako Generátor hodin.

Obr.1 Taktování vnitřním oscilátorem - připojení krystalu



Obr.2 Taktování externím oscilátorem - připojení vnějšího zdroje

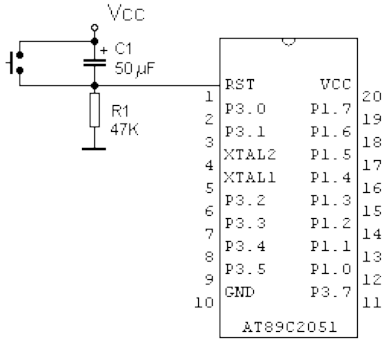


Nulování Mikroprocesoru

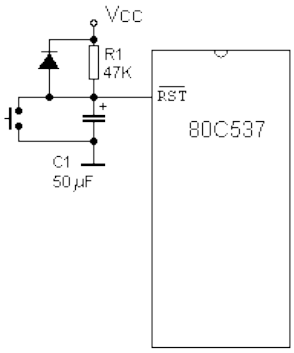
Na inicializaci mikrokontroléru je k dispozici vstup RST (reset), na který je připojen interní Schmittův obvod. Proto lze na tento vstup připojit i spojitě se měnící napětí, např. napětí na kondenzátoru při automatické inicializaci po připojení napájecího napětí. Pro verze CMOS lze rezistor vynechat,protože vstup RST má vnitřní zatěžovací rezistor. Tehdy musí být hodnota předřazené kapacity snížena na úroveň asi 6uF. Klasický mikroprocesor 8051 a jeho klony např. ATMEL AT89C51 a AT89C2051 až AT89C4051 mají resetovací úroveň log. 1. Na tomto místě je třeba upozornit na to, že řada nástupců 8051 má aktivní úroveň nulovacího impulzu opačnou (log.0), jak bývá u většiny procesorů obvyklé. Pro vysvětlení se zde budu zabývat jen RESET-em klasického procesoru 8051. Podrobnosti o aktivní úrovni signálu RESET naleznete v dokumentaci ke každému typu procesoru. Reset mikrokonlroléru proběhne, bude-li signál RST=1 alespoň po dobu dvou strojových cyklů (24 period hodinových impulzů oscilátoru). Samozřejmě pokud oscilátor běží. V době inicializace se nastaví ALE a PSEN do vstupního stavu. Vnitřní inicializace se vykoná v době druhého strojového cyklu, ve které je RST=1 a opakuje se pokud RST=1.Obsah interní paměti RAM se resetem neovlivní a po zapnutí napájecího napětí je v ní náhodný obsah (v aplikacích, kde na tom záleží je ji třeba vynulovat programově po resetu).Po připojení napájení drží obvod vstup RST ve vysoké úrovni po dobu,která je závislá na hodnotě připojené kapacity a rychlostí náběhu napájecího napětí. Abychom si byli jisti,že RESET bude platný, musí být RESET držen ve vysoké úrovni déle než dva strojové cykly.

Přivedení napájecího napětí bez vykonání platného RESETu způsobí, že CPU mikrokontroléru začne vykonávat program z náhodného místa v paměti programu, protože programový čítač nebyl spolehlivě vynulován a jeho obsah byl náhodný. Uvedené jednoduché nulivací obvody nejsou vhodné pro systémy se zálohovanou pamětí dat RAM nebo pamětí EEPROM, protože v případě krátkodobého poklesu napájecího napětí nevytvářejí nulovací impuls a procesor pracující mimo tolerance napájecího napětí udaného výrobcem může snadno jakýmkoliv způsobem přepsat zálohovaná data.Pro takovéto aplikace je třeba používat nulovací obvod nebo nulovací generátor s obvodem WATCHDOG.

Obr.4 Jednoduchý resetovací obvod pro RESET po přivedení napájecího napětí,aktivní úroveň RST=1.



Obr.5 Jednoduchý resetovací obvod pro RESET po přivedení napájecího napětí,aktivní úroveň RST=0.



Vstupně - výstupní porty

Vstupně/výstupní porty nám umožňují připojit k mikroprocesoru cokoli dalšího (tranzistory, LED, tlačítka,...), co pracuje s logickými signály (tj.dvoustavovými - log.0 a log.1). Vstupně / výstupní (v/v) porty jsou slučitelné s TTL/CMOS logikou, což nám tak udává napěťové úrovně pro log.0 a 1. U standardní verze 8051 máme k dispozici 4 v/v porty, z nichž každý má po 8 pinech. Tyto čtyři porty jsou označovány P0, P1, P2 a P3. Pokud chceme označit konkrétní pin určitého portu, pak označení vypadá celkem logicky takto - např. P1.4 , kde 1 před des.tečkou určuje port a 4 za tečkou je číslo konkrétního pinu příslušejícího k portu 1. Místo slova *port* se někdy můžete setkat s označením *brána*. Z toho, že se bavíme o vstupně/výstupních portech vyplývá, že tyto porty (všechny čtyři) jsou obousměrné - můžeme tedy jednotlivé piny všech portů v rámci jednoho programu používat jako vstupní i jako výstupní (viz dále). Porty hardwarově sestávají ze vstupních vyrovnávacích pamětí, klopných obvodů typu D.

Charakteristika portů

Port 0 jako jediný nemá ve výstupním budiči zvyšovací odpor - jde tedy o obvod s otevřeným kolektorem. Pokud jej chceme použít jako obecné vstupy/výstupy, musíme k pinům připojit externí odpory, jejichž druhý konec připojíme na +Ucc. Port 0 se také může využívat při styku s vnější pamětí - popis najdete v odstavci věnovaném portu 2.
Port 1 nemá žádnou alternativní funkci, díky tomu jej můžeme bez nějakého omezení libovolně používat.
Port 2 se kromě své normální funkce může využívat spolu s portem 0 ke komunikaci s externí pamětí. K tomuto účelu se využívají jen části obou portů - výstupní budiče portu 0 a 2 a vstupní vyrovnávací paměť portu 0. Na portu 0 je časově přepínán výstup nižšího bytu adresy pro externí paměť s datovým bytem, který je zapisován nebo čten z paměti. Port 2 pak vysílá vyšší byte adresy. Tato adresa je tedy 16-bitová, abychom mohli adresovat až 64kB paměti. Port 2 jinak pracuje jako normální port.
Port 3 - všechny piny portu 3 jsou vícefunkční. Kromě standardní funkce navíc tento port poskytuje své piny pro potřeby jiným vnitřním obvodům mikroprocesoru, seznam pinů a jim příslušné alternativní funkce zobrazuje následující tabulka:

Pin	Alternativní funkce
P3.O	RXD (serial input)
P3.1	TXD (serial output)
P3.2	INTO (external interrupt)
P3.3	INT1 (external interrupt)
P3.4	TO (Timer/Counter O external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	WR (external Data Memory write strobe)
P3.7	RD (external Data Memory read strobe)

LCD displej MC1602E-SYL/H

Teoretické informace staženy z: http://www.cmail.cz/doveda/index.htm



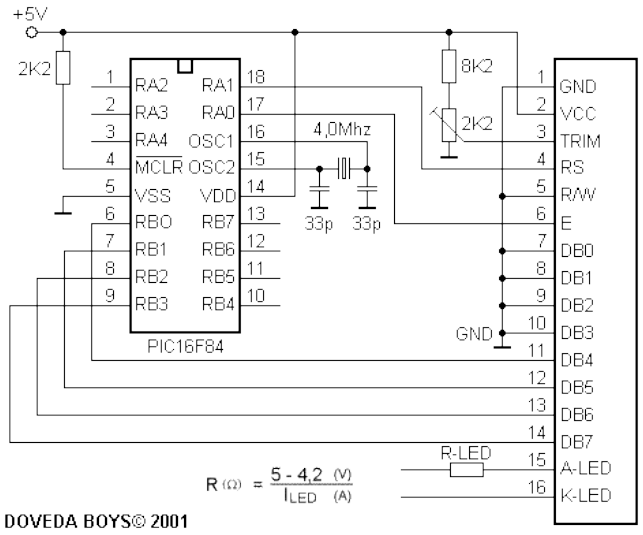
Komunikace s modulem

Pro komunikaci s displejem je potřeba minimálně 6 vodičů až maximálně 11 vodičů a napájení +5V. Před zapojením s procesorem si musíme zvolit datovou komunikaci. Komunikace s modulem muže být po 8-bitové sběrnici (DB0-DB7) nebo 4-bitové sběrnici (DB4-DB7).

Po zapnutí procesoru a displeje je nutné provést základní inicializaci displeje. Nastaví se počet bitů datové komunikace (8 nebo 4 bity), směr psaní znaků, posun řádku, pozice kurzoru a blikání kurzoru. Po této inicializaci je displej připraven na komunikaci.

8-bitová sběrnice (DB0-DB7). Pošle se 8 bitů a zapíšou se Enablem

4-bitová sběrnice (DB4-DB7). Tuto komunikaci je vhodné použít při nedostatku vývodů na procesoru. Komunikace bude 2x pomalejší, protože se data posílají nadvakrát. Nejdříve se pošlou vyšší 4 bity a zapíšou se Enablem a pak nižší 4 bity a zapíšou se Enablem. Nižší 4 bity (DB0-DB3) na displeji je potřeba spojit se zemí. Zapojení displeje s procesorem je na obrázku.



vývod standard	**název**	**popis**
1	Vss	napájení GND
2	Vdd	napájení +5V
3	Vo	kontrast 0V .. 5V (typ. 0,8V)
4	RS	Register Select (0=instrukce, 1=data)
5	R/W	Read / Write (0=zápis, 1=čtení)
6	E	Enable (1, **hrana z 1 -> do 0**)
7	DB0	Data Bus 0
8	DB1	Data Bus 1
9	DB2	Data Bus 2
10	DB3	Data Bus 3
11	DB4	Data Bus 4
12	DB5	Data Bus 5
13	DB6	Data Bus 6
14	DB7	Data Bus 7

15	A	podsvícení - anoda LED (jen některé typy)
16	K	podsvícení - katoda LED (jen některé typy)

Software

Mikropočítač byl naprogramován v jazyce C.
Text byl použit ze stránky: http://cs.wikipedia.org/wiki/C_(programovac%C3%AD_jazyk)

C je programovací jazyk, který vyvinuli Ken Thompson a Dennis Ritchie pro potřeby operačního systému Unix. V současné době je to jeden z nejpopulárnějších jazyků, zřejmě nejčastější pro psaní systémového softwaru, ale velmi rozšířený i pro aplikace.

C je nízkoúrovňový, kompilovaný, relativně minimalistický programovací jazyk. Je dostatečně mocný na většinu systémového programování (ovladače a jádro OS) a zbytek lze dořešit tzv. inline assemblerem, tedy metodou zápisu assembleru přímo do kódu. Zdrojový kód C je přitom mnohem čitelnější než assembler, je jednodušší ho zapsat a navíc je snáze přenositelný na jiné architektury. Proto jsou často operační systémy, překladače, knihovny a interpretry vysokoúrovňových jazyků implementovány právě v C.

Ukládání dat je v C řešeno třemi základními způsoby: statickou alokací paměti (při překladu), automatickou alokací paměti na zásobníku a dynamickou alokací na haldě (heap) pomocí knihovních funkcí. Jazyk disponuje jen minimální abstrakcí nad alokací: s pamětí se pracuje přes datový typ zvaný ukazatel, který drží odkaz na paměťový prostor daného typu proměnné, ale je na něm možné provádět aritmetické operace (tyto operace ale neoperují s ukazateli přímo na úrovni jednotlivých bajtů, nýbrž přihlíží k velikosti datového typu, na který ukazují - existují ale také ukazatele typu void *, které mohou odkazovat na jakýkoliv typ dat uložený v paměti.). Ukazatelé tedy existují nezávisle na proměnných, na které odkazují, a je na odpovědnosti programátora, aby neukazovaly na paměť nealokovanou.

Ukazatele jsou velmi mocným nástrojem, protože C jazyk povoluje ukazatele nejen na data, ale i na funkce. Současně jsou ukazatele z hlediska přenositelnosti a rizika zhroucení programu při jejich nesprávném použití Achillovou patou jazyka. Na druhou stranu, programátorovi je dána plná zodpovědnost za alokace paměti, není zde tedy závislost na automatickém dealokátoru paměti (garbage collector), jehož činnost nemá v jazycích vyšší úrovně šanci ovlivnit.

Jazyky Java a C#, oba odvozené od C, používají méně univerzální způsob odkazování alokovaných proměnných, který snižuje pravděpodobnost chyby v programu. Jazyk C++, původně rozšíření jazyka C, si ovšem ukazatele zachoval.

Mnoho dalších moderních programovacích jazyků přebírá způsob zápisu (neboli syntaxi) z jazyka C. Patří mezi ně například zmíněná Java či Perl a PHP.

Vlastní konstrukce zařízení

Hardware

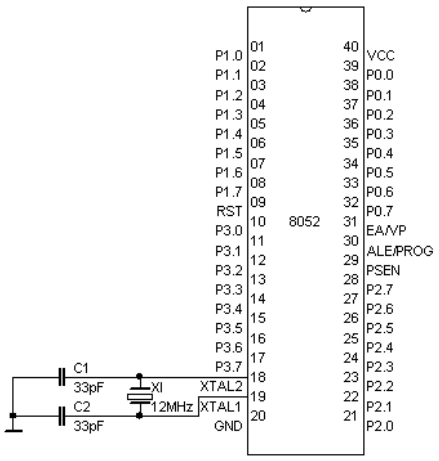
Kompletní fotodokumentace z tvorby mého zapojení najdete v příloze č.6, schémata jsou uvedeny v příloze č.2, 3, 4, 5 a ekonomický rozbor je uveden v příloze č.7.

Mikroprocesor 8052

K tomuto projektu byl použit Mikroprocesor 8052, který má stejné vlastnosti jako mikroprocesor 8051 (vlastnosti uvedeny výše) s nepatrným rozdílem. Mikroprocesor 8052 má přidán ještě jeden časovač T2 a ještě několik rozšiřující funkcí.

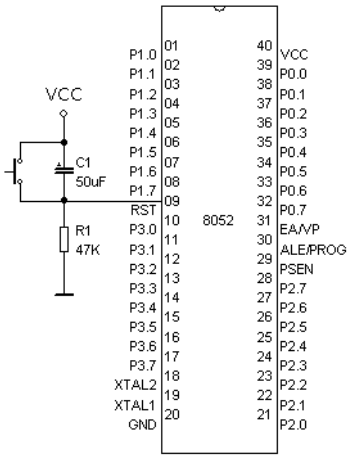
Časování Mikroprocesoru

K taktování bylo použito zapojení pomocí vnitřního oscilátoru - krystalu o frekvenci 12MHz. Krystal připojujeme podle přiloženého obrázku k vývodům XTAL1 a XTAL2. Paralelním připojením kondenzátorů signál vyhlazujeme. Vlastní nastavení pro mikroprocesor je popsáno v kapitole software.



Nulování Mikroprocesoru (resetovací obvod)

Resetovaní obvod připojujeme na pin RST. Reset mikrokonlroléru proběhne, bude-li signál RST=1 alespoň po dobu dvou strojových cyklů. Samozřejmě pokud oscilátor běží. Stisknutím tlačítka se celý mikroprocesor restartuje. Tato funkce je důležitá po připojení do sítě, aby se celý program reloudoval na začátek.



Zapojení jednotlivých portů

V projektu byli použity porty P1 a P2. Porty P1 byly použity pro ovládací prvky (tlačítka), jako výstup signálu pro zvonění a pro programování mikroprocesoru. Porty P2 byli použity pro komunikaci s LCD displejem.

Porty P1:

OVLADANI:
P1.1 tlacitko - mode
P1.2 tlacitko - ++
P1.3 tlacitko - --
P1.4 tlacitko - zvoneni

VYSTUPY:
P1.4 vystup zvonek

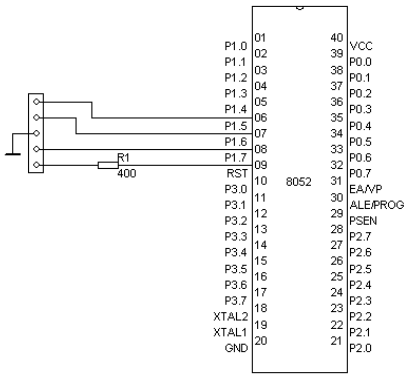
PROGRAMÁTOR:
P1.5
P1.6
P1.7

Porty P2:

LCD:
P2.0 - RS
P2.1 - RW
P2.2 - E
P2.4 - D4
P2.5 - D5
P2.6 - D6
P2.7 - D7

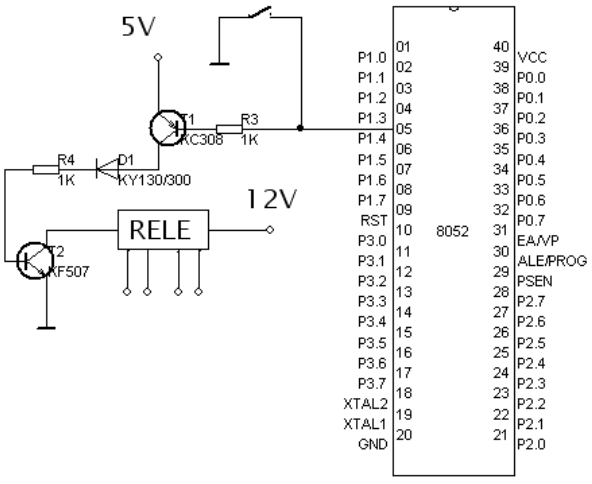
Programátor

Programátor slouží pro připojení k PC paralelním portem. Pro programování mikropočítače stačí jen jeho připojení na napájení 5V= a připojením na zem.



Výstup signálu pro zvonění

Obvod určený ke spínání rele je sestrojen pomocí tranzistorů zapojených jako spínače. Přivedením napětí neboli log 1 na tranzistor T1 dojde k sepnutí tranzistoru tedy sepneme 5 V. Signál sepne tranzistor T2 který spíná 24V obvod pro spínání rele. Zatížitelnost svorek relátka je 11A. Zde je také vyřešeno zapojení pro manuální zvonění, sepnutím tlačítka propojíme obvod se zemí a dojde k následnému spínání tranzistorů (popsáno výše).

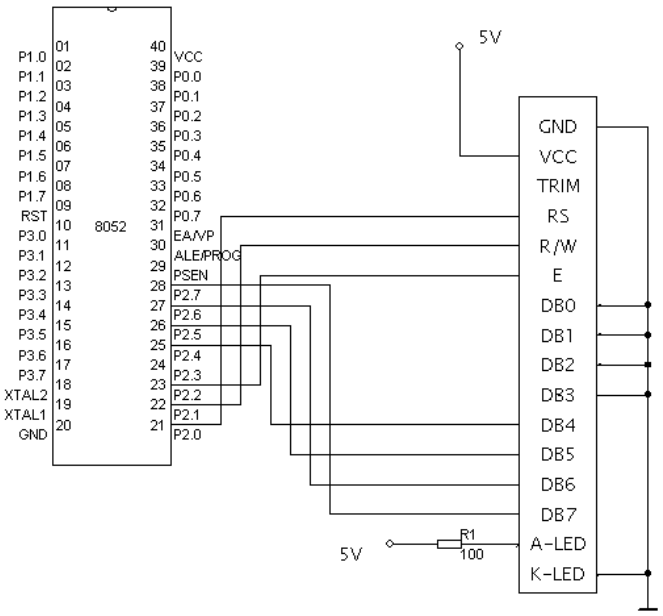


Připojení LCD displeje k Mikropočítači

Pro komunikaci s LCD displejem byl použit port P.2. Rozdělení jednotlivých pinů je následující:
- P2.0 –RS
- P2.1 – RW
- P2.2 – E
- P2.3 – volný
- P2.4 – D4
- P2.5 – D5
- P2.6 – D6
- P2.7 – D7

Dále musíme na LCD displej přivést napájení a zapojit země. Displej má funkci LED podsvícení proto musíme na anodu přivést přes rezistor napětí a na katodu zem. Úroveň intenzity podsvícení určuje velikost Rled, my použily velikost tohoto rezistoru 100ohm.

Zapojení LCD displeje

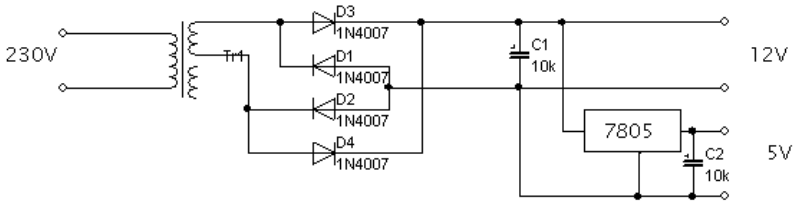


Napájení 230/12V

Zařízení je z bezpečnostních důvodů uzemněno a opatřeno ochranou pojistkou. Celé zařízení je napájeno síťovým napětím 230V, které je transformováno na 2*12V přes transformátor se dvěma okruhy. Pro zařízení bylo použito pouze jednoho okruhu. Napětí 12V bylo usměrněno

pomocí Grettzova můstku a vyhlazeno pomocí elektrolytického kondenzátoru. Toto napětí je potřebné ke spínání relé. Z tohoto napětí vyvedeme vývody a připojíme integrovaný stabilizátor 7805 který stabilizuje napětí 12V na 5V které využívá ke své funkci mikropočítač. Toto napětí dále vyhlazujeme elektrolytickým kondenzátorem. Celé zařízení musí mít (pro správnou funkci) společnou kostru obou napětí. Viz obrázek.

Zapojení zdroje



Ovládání LCD displeje

LCD displej je ovládán třemi tlačítky (čtvrté tlačítko má funkci manuálního zvonění). První tlačítko má funkci MODE (mody nastavování). Jednotlivými stisky tohoto tlačítka se přepínáme mezi jednotlivými mody: ŠKOLNÍ HODINY
NASTAVENÍ SEKUND
NASTAVENÍ MINUT
NASTAVENÍ HODIN
NASTAVENÍ ZVONĚNÍ
DELKA ZVONĚNI XX
NÁVRAT – ŠKONÍ HODINY

Druhé tlačítko má funkci přičítání. Tedy stiskem tlačítka přičítáme (sekundy, minuty, hodiny, zvonění zap/vyp, délka zvonění) v jednotlivých módech.

Třetí tlačítko má funkci odečítání. Tedy stiskem tlačítka odečítáme (sekundy, minuty, hodiny, zvonění zap/vyp, délka zvonění) v jednotlivých módech.

Nastavení času:
Nastavení času probíhá v modu (nastavení sekund, minut, hodin). Nastavujeme tlačítky 2 a 3 (přičítání a odečítáním).

Nastavení zvonění:
V tomto modu nastavujeme zvonění zapnuto nebo vypnuto. Na displeji je označeno : * zapnuto, - vypnuto.

Délka zvonění:
V tomto modu nastavujeme délku zvonění. Nastavení provádíme tlačítky 2 a 3. délku je možné nastavit od 5s do 59s.

Software

Kompletní ukázku programu naleznete v příloze č. 1

Hlavní funkce - Nastavení timeru T2

Mikropočítač 8052 obsahuje timer T2. Timery lze také použít jako časovače. V hardwarové části jsem použil 12 MHz krystal. Integrovaný obvod dělí takt krystalu 12, kvůli 16 bitovému čítači. Tímto je dána frekvence potřebná pro přetečení $2*10_{16} - 1 = 65535$ v hexadecimální soustavě 0xFFFF . Dělením tedy získáme frekvenci 1MHz který má délku 1us.Instrukce má 12 částí ve kterých se vykonává. Proto dělíme 12ti pokud potřebujeme přetéci právě po 1 ms. Ze vztahu 1ms = 1000us proto odečteme z frekvence 1000 (0xFFFF-1000).
T2 režim autoreloud – automaticky nastaví při přerušení obsah čítače T2 na hodnotu uloženou v RCAPH + RCAPL. Nastaví se do T2H + T2L (v programu).
Musíme dávat pozor aby přerušení netrvalo déle jak 1ms. Aby stihlo obsloužit přerušení než se zavolá další.

Pole párů časů zvonění

V tomto poli jsou uvedeny časy zvonění. Program nejdříve kontroluje po každém přetečení mikrokontroleru první číslici ve sloupci (hodiny) a pokud dojde ke schodě s aktuálním časem hodin zkontroluje druhý sloupec (minuty). Pokud dojde ke schodě je zapnuto zvonění. Pokud nedojde ke schodě pokračuje v kontrolování hodin.

```
unsigned char pole [] = {
   7,  0,
   7, 30,
   7, 55,
   8,  0,
   8, 45,
   8, 50,
   9, 35,
   9, 50,
  10, 35,
  10, 40,
  11, 25,
  11, 30,
  12, 15,
  12, 20,
  13,  5,
  13, 15,
  14,  0,
  14, 20,
  15,  5,
  15, 15,
  16,  0,
  17,  0,
  18,  0,
  255,255,
};
```

Funkce přičítání času

Mikrokontroler je nastaven aby přetekl každou milisekundu. V této funkci bylo použito podmínek, milisekundy přičítají a když dosáhnou hodnoty 1000 milisekund přičte se jedna

sekunda a milisekundy se vynulují. Takto přičítá a nuluje až k hodinám, až hodiny dosáhnou 24 hodin také se vynulují a vše se opakuje znovu.

```
 if ( ++ms == 1000 ) {
 ms = 0;
 refresh = 1;
  if ( ++sec == 60 ) {
    sec = 0;
    if ( ++min == 60 ) {
      min = 0;
      if ( ++hour == 24 ) {
        hour = 0;
      }
    }
  }
```

Výpisy na LCD displej

Pro výpis textu na displeji musíme nejdříve určit polohou výpisu, tu určujeme pomocí příkazu lcd_com. Poté můžeme na toto místo posílat informace pomocí příkazu lcd_puts a lcd_data. Knihovna pro ovládání LCD byla převzata z codelibrary ze stránek: http://www.8052.com/codelib.phtm "connect lcd in 4 bit mode"

```
while (1) {
    if ( refresh ) {
      lcd_com(0x80 /* adresa */ );

      lcd_data(hour/10 + '0');
      lcd_data(hour%10 + '0');

      lcd_data(':');

      lcd_data(min/10 + '0');
      lcd_data(min%10 + '0');

      lcd_data(':');

      lcd_data(sec/10 + '0');
      lcd_data(sec%10 + '0');

            lcd_com(0x80+0x0F);
            if (zvoneni)
                lcd_data('*');
            else lcd_data('-');

      lcd_com(0x80+0x40);
      switch ( nastaveni ) {
        case 1:  lcd_puts("Nastaveni sekund") ; break ;
        case 2:  lcd_puts("Nastaveni minut ") ; break ;
        case 3:  lcd_puts("Nastaveni hodin ") ; break ;
        case 4:  lcd_puts("Nastaveni zvon  ") ; break ;
        case 5:  lcd_puts("Delka zvoneni ") ; lcd_data(delka_zvoneni/10 + '0');
                                    lcd_data(delka_zvoneni%10 + '0');
                                    break ;
        default: lcd_puts("skolni hodiny   ") ; break ;
      }
```

```
    refresh = 0;
  }}}
```

Funkce stisknutí tlačítka

V programu jsou naprogramována tři tlačítka, čtvrté naprogramováno být nemusí protože slouží k funkci manuálního zvonění. Toto tlačítko spojuje zvonící obvod se zemí zařízení.

```
if ( !P1_1 ) {
   if (tlacitka1 == sw_on) stisknuto_tl1();
   tlacitka1++;
   if (tlacitka1 == sw_reload) tlacitka1=sw_on;
 } else {
  tlacitka1=0;
 }
```

Funkce stisků tlačítek

Tato část programu nastavuje hlavní funkce zařízení,1 slouží pro nastavování všech dostupných funkcí které jsme naprogramovali skrze tlačítka.

```
void stisknuto_tl1() {
 if ( ++nastaveni == (MAX_NASTAVENI + 1) ) {
  nastaveni = 0;
 }
 refresh =1;
}

void stisknuto_tl2() {

 switch (nastaveni) {
   case 1:
     if (++sec  == 60)  sec = 0;
     break;

   case 2:
     if (++min  == 60)  min = 0;
     break;

   case 3:
     if (++hour == 24) hour = 0;
     break;

   case 4:
     zvoneni = !zvoneni;
     break;

   case 5:
     if (delka_zvoneni < 59) delka_zvoneni++ ;
     break;
 }

 refresh =1;
}

void stisknuto_tl3() {
 switch (nastaveni) {
   case 1:
       if (sec--  == 0)  sec = 59;
     break;
```

```
   case 2:
       if (min--  == 0)  min = 59;
       break;

   case 3:
       if (hour-- == 0) hour = 23;
       break;

   case 4:
       zvoneni = !zvoneni;
       break;

   case 5:
     if (delka_zvoneni > 5) delka_zvoneni-- ;
     break;

  }
}
```

Funkce pro vyhledávání zvonění

Tato funkce prochází pole s časy zvonění (viz více pole párů zvonění) pokud nalezne schody v obou případech dochází ke zvonění.

```
 unsigned char i ;
       for ( i = 0 ; pole[i] != 255 ; i = i+2 ) {
        if ( pole[i] == hour && pole[i+1] == min && sec < delka_zvoneni && zvoneni ) return 1;

       }
       return 0;
}
```

Příloha č.1

Kompletní program v jazyce C

```
#include <at89x52.h>

// LCD

#define _RS P2_0
#define _RW P2_1
#define _E P2_2
#define LCDPORT P2

#include "lcd.h"

#define sw_on (20)
#define sw_reload (350 + sw_on)

// LCD:
// P2.0 - RS
// P2.1 - RW
// P2.2 - E
// P2.3 - ---
// P2.4 - D4
// P2.5 - D5
// P2.6 - D6
// P2.7 - D7

// PROGRAMOVANI:
// P1.5
// P1.6
// P1.7
// RST

// OVLADANI
// P1.4 tlacitko - zvoneni
// P1.1 tlacitko - mode
// P1.2 tlacitko - ++
// P1.3 tlacitko - --

// VYSTUPY
// P1.4 vystup zvonek

volatile unsigned int ms = 0;
volatile unsigned char sec = 0;
volatile unsigned char min = 0;
volatile unsigned char hour = 0;
volatile unsigned char delka_zvoneni = 10; // delka zvoneni
volatile unsigned char zvoneni = 1;  // vyp zvoneni
volatile unsigned int tlacitka1 = 0;
volatile unsigned int tlacitka2 = 0;
volatile unsigned int tlacitka3 = 0;
```

```
volatile bit refresh = 1 ;


unsigned char nastaveni = 0;
// nastaveni: 0 - zobrazeni casu
//            1 - nastaveni sekundy
//            2 - nastaveni minuty
//            3 - nastaveni hodiny
//            4 - zvoneni zap/vyp
//            5- delka zvoneni

#define MAX_NASTAVENI 5

void test_tlacitek();
void stisknuto_tl1();
void stisknuto_tl2();
void stisknuto_tl3();
bit je_zvoneni(void);

// POLE PARU CASU ZVONENI    pole[0]==7   pole[1]==0

unsigned char pole [] = {
   7,  0,
   7, 30,
   7, 55,
   8,  0,
   8, 45,
   8, 50,
   9, 35,
   9, 50,
  10, 35,
  10, 40,
  11, 25,
  11, 30,
  12, 15,
  12, 20,
  13,  5,
  13, 15,
  14,  0,
  14, 20,
  15,  5,
  15, 15,
  16,  0,
  17,  0,
  18,  0,
  255,255,
};

// ZAKLADNI NASTAVENI
void timer2_isr (void) __interrupt (TF2_VECTOR)
{
  EXF2 = 0; // pro jistotu vynulovani
  TF2 = 0;

  if ( ++ms == 1000 ) {
```

```
  ms = 0;
  refresh = 1;
   if ( ++sec == 60 ) {
     sec = 0;
     if ( ++min == 60 ) {
       min = 0;
       if ( ++hour == 24 ) {
         hour = 0;
       }
     }
   }

   // mam zvonit?
   // ....
   P1_4 = !je_zvoneni();
  }

  test_tlacitek();

}


void main(void)
{
  EA=1; // povolení vsech preruseni
  ET2=1;   // povoleni preruseni od casovace 2

  // nastaveni T2 reload (znovunacteni hodnoty)
  TCLK=0;
  RCLK=0;
  CP_RL2=0;

  // 12MHz  65536-1000 => 1ms
  RCAP2H=0xFC ;
  RCAP2L=0x18 ;

  TL2=0xFC;
  TH2=0x18;
  TR2=1;
  lcd_init();
  lcd_com(12);



   while (1) {
    if ( refresh ) {
      lcd_com(0x80 /* adresa */ );

      lcd_data(hour/10 + '0');
      lcd_data(hour%10 + '0');

      lcd_data(':');

      lcd_data(min/10 + '0');
```

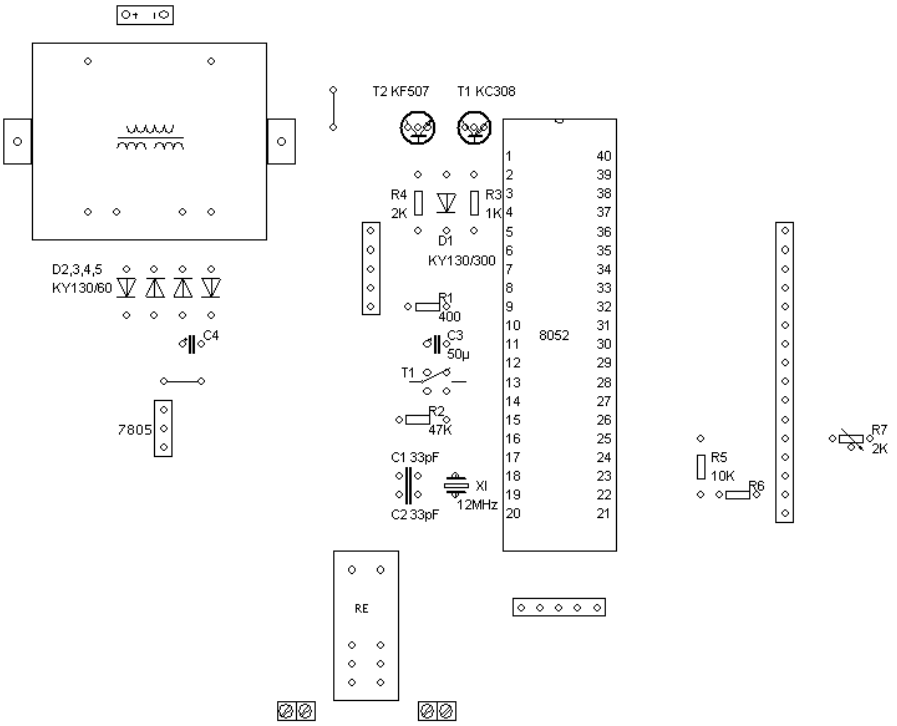
```
      lcd_data(min%10 + '0');

      lcd_data(':');

      lcd_data(sec/10 + '0');
      lcd_data(sec%10 + '0');

           lcd_com(0x80+0x0F);
           if (zvoneni)
               lcd_data('*');
           else lcd_data('-');

      lcd_com(0x80+0x40);
      switch ( nastaveni ) {
        case 1:  lcd_puts("Nastaveni sekund") ; break ;
        case 2:  lcd_puts("Nastaveni minut ") ; break ;
        case 3:  lcd_puts("Nastaveni hodin ") ; break ;
        case 4:  lcd_puts("Nastaveni zvon  ") ; break ;
        case 5:  lcd_puts("Delka zvoneni ") ; lcd_data(delka_zvoneni/10 + '0');
                                    lcd_data(delka_zvoneni%10 + '0');
                                    break ;
        default: lcd_puts("skolni hodiny   ") ; break ;
      }



      refresh = 0;
    }
  }


}



void test_tlacitek(void) {


// tlacitko 1
  if ( !P1_1 ) {
    if (tlacitka1 == sw_on) stisknuto_tl1();
    tlacitka1++;
    if (tlacitka1 == sw_reload) tlacitka1=sw_on;
  } else {
    tlacitka1=0;
  }

// tlacitko 2
  if ( !P1_2 ) {
    if (tlacitka2 == sw_on) stisknuto_tl2();
    tlacitka2++;
    if (tlacitka2 == sw_reload) tlacitka2=sw_on;
  } else {
    tlacitka2=0;
  }
```

```
// tlacitko 3
  if ( !P1_3 ) {
    if (tlacitka3 == sw_on) stisknuto_tl3();
    tlacitka3++;
    if (tlacitka3 == sw_reload) tlacitka3=sw_on;
  } else {
    tlacitka3=0;
  }

}

void stisknuto_tl1() {
  if ( ++nastaveni == (MAX_NASTAVENI + 1) ) {
    nastaveni = 0;
  }
  refresh =1;
}

void stisknuto_tl2() {

  switch (nastaveni) {
    case 1:
      if (++sec  == 60)  sec = 0;
      break;

    case 2:
      if (++min  == 60)  min = 0;
      break;

    case 3:
      if (++hour == 24) hour = 0;
      break;

    case 4:
      zvoneni = !zvoneni;
      break;

    case 5:
      if (delka_zvoneni < 59) delka_zvoneni++ ;
      break;
  }

  refresh =1;
}

void stisknuto_tl3() {
  switch (nastaveni) {
    case 1:
        if (sec--  == 0)  sec = 59;
      break;

    case 2:
        if (min--  == 0)  min = 59;
        break;
```

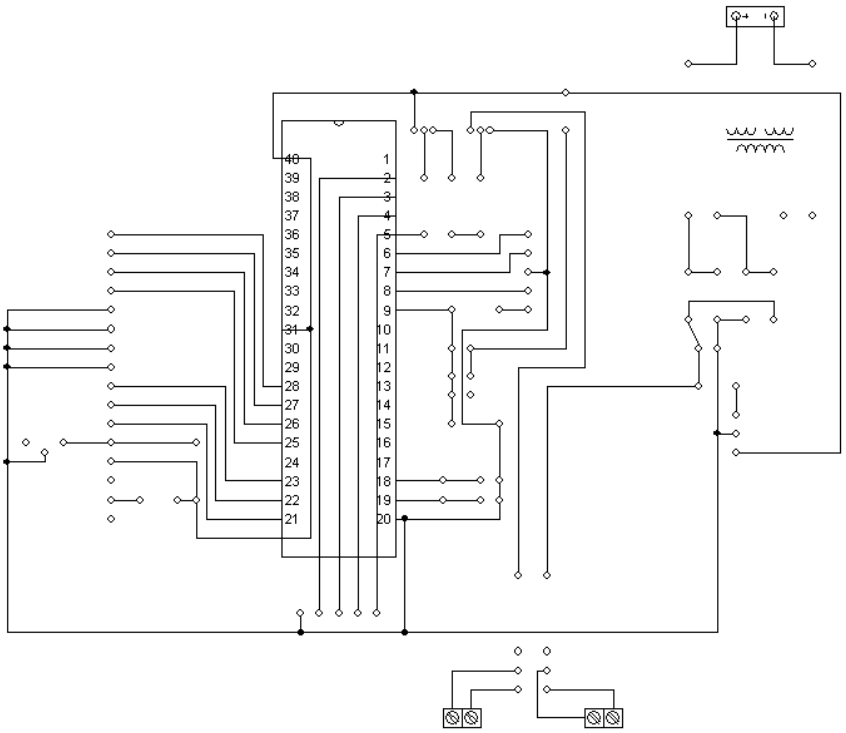
```
   case 3:
        if (hour-- == 0) hour = 23;
        break;

   case 4:
        zvoneni = !zvoneni;
        break;

   case 5:
      if (delka_zvoneni > 5) delka_zvoneni-- ;
      break;

  }
}


bit je_zvoneni(void) {
 // fce pro vyhledavani zvoneni
     unsigned char i ;
       for ( i = 0 ; pole[i] != 255 ; i = i+2 ) {
        if ( pole[i] == hour && pole[i+1] == min && sec < delka_zvoneni && zvoneni ) return 1;

       }
       return 0;
}
```

Příloha č. 2

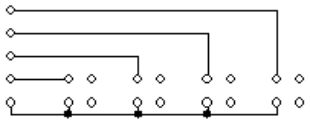
Plošný spoj - rozmístění součástek



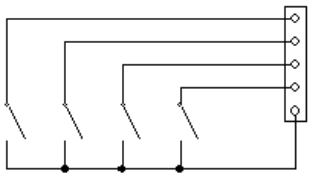
Příloha č. 3
Plošný spoj



Příloha č. 4
Tlačítka ze strany plošného spoje



Příloha č. 5
Tlačítka - zapojení ze strany součástek



Teoretická příprava:

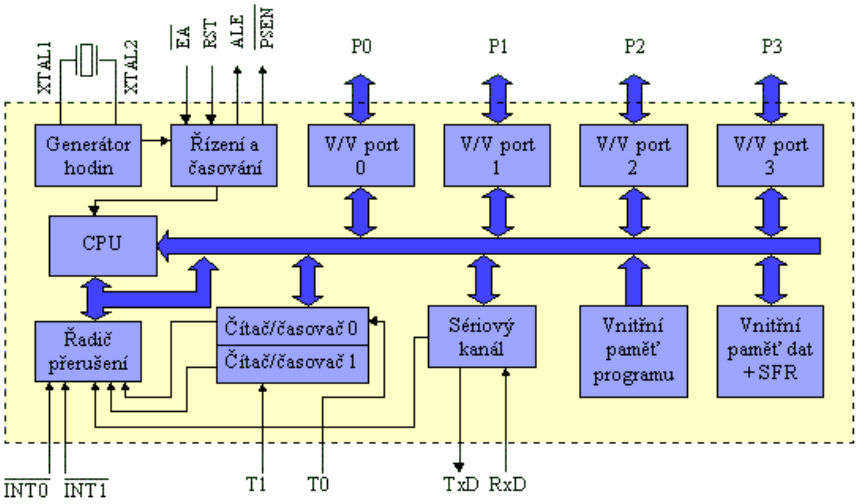
Teoretická část se zabývá teoretickou přípravou a nastavením hlavních častí přístroje a dalšími možnostmi řešení.

Hardware

Mikroprocesor 8051

Teoretické informace staženy z: http://www.dhservis.cz/

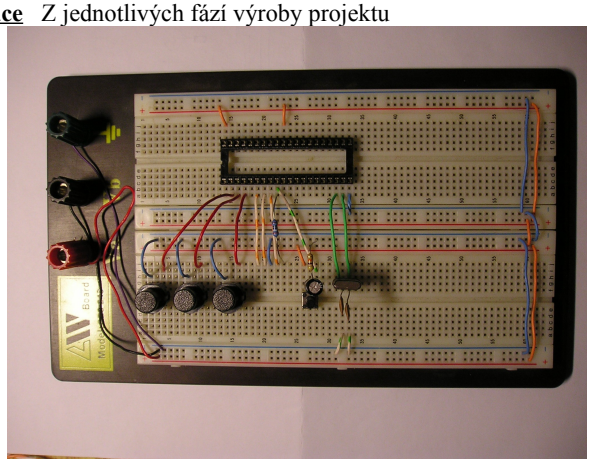
Mikroprocesor 8051 je 8-bitový jednočipový mikroprocesor s harwardskou strukturou, u které je oddělena programová a datová paměť. Vnitřní struktura mikroprocesoru je zobrazena na obr.1. Procesor je schopen samostané činnosti po připojení vnějšího piezokeramického rezonátoru (krystalu) na vývody XTAL1 a XTAL2, napájecího napětí 5V ss na vývod Ucc a připojení nulovacího obvodu na vývod RST. Pokud nepoužíváme vnější paměť programu, je ještě nutné nastavit vstup EA (neg.) do log.1.

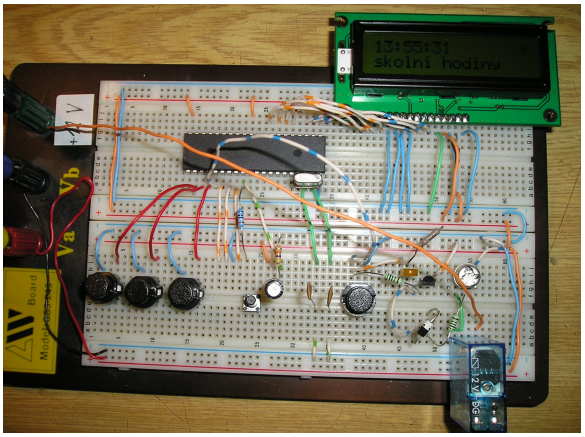


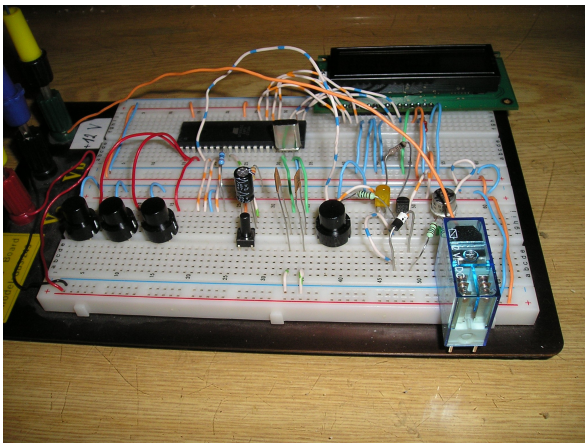
Mikroprocesor tvoří centrální procesorová jednotka (CPU), jejíž podstatnou částí je aritmeticko-logická jednotka. Ta umožňuje pracovat s jednotlivými bity paměti, vykonávat instrukce programu atd. Centrální procesorová jednotka je vnitřní 8-bitovou společnou sběrnicí (v obr.1 světle zelená plná šipka) propojena s pamětí programu a pamětí dat. Vnitřní paměť programu o velikosti 4kB může být typu ROM (8051), EPROM (8751) nebo mikroprocesor nemusí mít žádnou vnitřní pamět programu (8031). Vnitřní paměť dat je typu RAM o velikosti 128 bytů. Ke společné sběrnici jsou dále připojeny 4 vstupně/výstupní porty P0 až P3, které umožňují styk mikroprocesoru s vnějšími periferiemi.

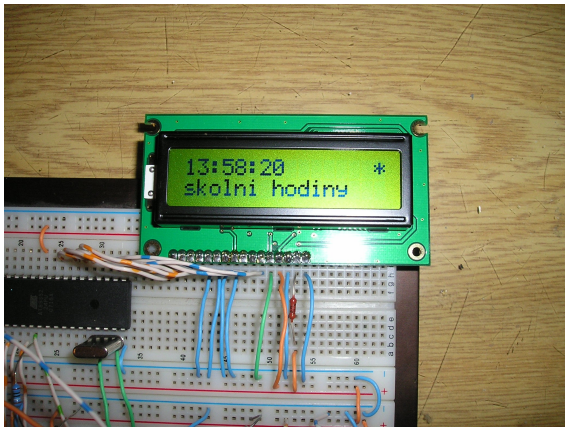
Příloha č. 6

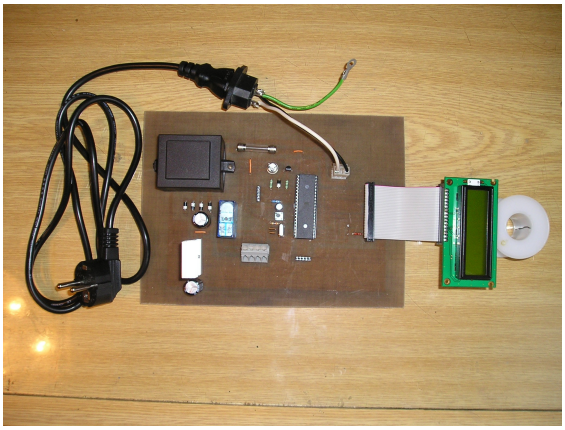
Fotodokumentace Z jednotlivých fází výroby projektu








13:58:20 *
skolni hodiny





Příloha č.7

Ekonomický rozbor

Náklady na materiál:	nákup elektromateriálu	522kč
	Mikropočítač	25kč
	LCD Displej	125kč
	Destička měděná	80kč
	Elektromateriál	200kč
	Práce	-

		952kč

Literatura:

http://www.dhservis.cz/ (teoretická část Mikroprocesoru 8051)
http://www.cmail.cz/doveda/index.htm (teoretická část LCD displeje)
http://cs.wikipedia.org/wiki/C_(programovac%C3%AD_jazyk) (teoretická část softwaru)
http://www.8052.com/ (stažen základ volného programu pro komunikaci s displejem)

Velice děkuji za pomoc při zpracovávání projektu Miroslavu Mikušovi a dalším učitelům.

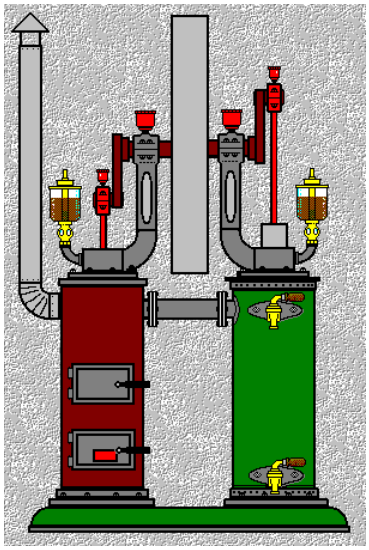
Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Stirlingův motor - „nový“ zdroj energie

Josef Jinek, Radek Štikar

Vyšší odborná škola a Střední průmyslová škola
Studentská ulice 1, Žďár nad Sázavou



.

Prohlašujeme tímto, že jsme soutěžní práci vypracovali samostatně pod vedením **Ing. Milana Řehoře** a dalších konzultantů uvedených v práci a uvedli v seznamu literatury veškerou použitou literaturu a další informační zdroje včetně Internetu.

Na úvod bychom chtěli poděkovat všem, kteří nám s prací pomáhali. Jsou to:

Dr. Ing. Josef Příhoda, ředitel VOŠ a SPŠ Žďár nad Sázavou, metodické vedení práce
Ing. Milan Řehoř, odborný učitel VOŠ a SPŠ Žďár nad Sázavou, oblast Stirlingova motoru
RNDr. Ladislav Jež, učitel matematiky a deskriptivní geometrie, pomoc při návrhu matematického modelu paraboly
Ing. Jaroslav Kletečka, učitel CAD, 3D model paraboly
Ing. Jaromír Tulis, učitel CAM, výroba formy na CNC frézce
Josef Štěpina, dílenský učitel, pomoc s výrobou kompletu zařízení
Vlastimil Nedopil, údržbář, pomoc s výrobou paraboly

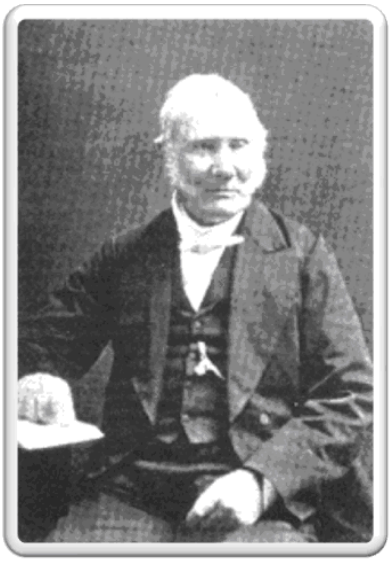
1. Úvod

Studujeme 3. ročník VOŠ a SPŠ ve Žďáře nad Sázavou, obor TZB (technická zařízení budov). O soutěži SOČ 2008 jsme se dozvěděli od našeho vyučujícího. Tato soutěž je pro nás oba novou a vítanou zkušeností jak se seznámit s novými věcmi, dostat alespoň trochu do povědomí, že naše planeta je také živá a my se musíme snažit ji co nejvíce ochraňovat a nebo k její ochraně co nejvíce přispívat. Důležitým aspektem je i to, že neobnovitelné zdroje energie docházejí nezadržitelně rychle a je potřeba je něčím nahradit, např. použitím Stirlingova motoru s ohřevem z obnovitelného zdroje energie. I když náš obor Technická zařízení budov se přímo na motory a jejich užívání nevztahuje, i v našem oboru se najdou oblasti využití těchto znalostí, např. čerpadlo nezávislé na dodávce elektrického proudu.

Stirlingův motor je velmi zajímavý **druh motoru s vnějším spalováním**, který v současnosti nalézá uplatnění jako **jeden z ekologických zdrojů energie.** Jeho přednosti jsou popsány v dalším textu.

2. Stirlingův motor z historického pohledu

Důvod vzniku Stirlingůva motoru je ten, že parní motory (v té době už velice známé) trpěly dlouhým nahříváním a rizikem exploze. První použitelný teplovzdušný motor sestrojil Sir George Cayley kolem roku 1807. Motor si ale nenechal patentovat a později je znám spíše jako průkopník letectví.



Obrázek 1 Robert Stirling

Skotský duchovní **Robert Stirling** je vynálezce, po němž se dnes teplovzdušné motory nazývají. Důvodem, proč se touto problematikou začal zabývat, bylo to, že chtěl ochránit své ovečky před explozí parního kotle. Robert Stirling není ani tak úplně první vynálezce samotného motoru, ale regenerátoru. Je to zařízení, které značně zefektivnilo provoz motoru. Motor si nechal patentovat **roku 1816.** Stirling podal patent pod názvem „Heat Economiser". Již v roce 1818 postavil motor s výkonem 2 koňských sil pro čerpání vody v kamenolomu. Jako konstruktér nezahálel a už o pár let později, v letech 1827 a 1840, si nechal patentovat zdokonalení svého vynálezu. Robert Stirling pracoval s teplovzdušnými motory celý život. Na jeho počest nesou motory jeho jméno ve svém názvu.

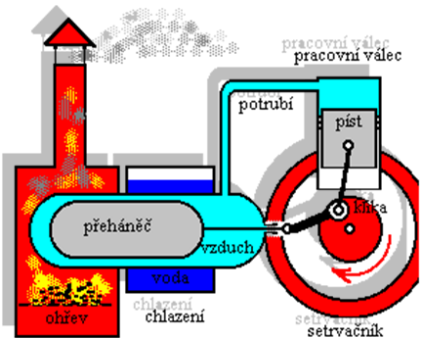
V průběhu 19. století a na počátku 20. století se objevovaly nejrůznější aplikace Stirlingových motorů. Malé Stirlingovy motory poháněly **zubařské vrtačky**, domácí **ventilátory**, **šicí stroje** atd. Velké typy byly používány k **pohonu navijáků** a v dalších průmyslových aplikacích. Až později byly vytlačeny spalovacími motory.

3. Co to vlastně Stirlingův motor je a jak funguje

Jednoduchý Stirlingův motor se skládá z těchto základních částí:

- Ohřívač (je to zdroj tepla potřebný k ohřevu vzduchu v pracovní komoře, ve které běhá přeháněč)

- Přeháněč (těleso válcovitého tvaru, které je připevněno ke klice se díky rozdílnému tlaku teplého a studeného vzduchu pohybuje ve válcovité nádobě viz. obrázek zleva doprava a zpět)



Obrázek 2 Schéma motoru

- Pracovní komora válcovitého tvaru se vzduchem (obsahuje určité množství vzduchu nebo jiného plynu pro lepší vlastnosti se užívá např. vodík, helium a pohybuje se v něm přeháněč)

- Chladič (je to zdroj chladu potřebný k ochlazení vzduchu v pracovní komoře, ve které běhá přeháněč)

- Potrubí (je to spojovací potrubí mezi pracovním válcem a pracovní komorou)

- Pracovní válec (pohybuje se v něm píst, který je v něm uložen natěsno)

- Píst (je připevněn ke klice a pohybuje se v pracovním válci)

- Klika (je v jednom bodě připevněna k setrvačníku, se kterým se pohybuje a spojuje píst a přeháněč se setrvačníkem)

- Setrvačník (má na sobě upevněnou kliku a otáčí se viz. obr. 2)

4. Stručný popis chodu Stirlingova motoru

Nejprve uvedeme do chodu **ohřívač** (kahan, hořák, kotel, atp.), který ohřeje část vzduchu ve válcovité nádobě. Téměř veškerý vzduch ve válci obteče **přeháněč** a nashromáždí se vlevo. Protože je tato část válce vyhřívána, ohřeje se i vzduch. Vzduch má snahu se rozpínat. Tím se zvýší jeho tlak. Vzduch o tomto vyšším tlaku je potrubím veden do pracovního válce, kde tlačí na píst a přes **klikový mechanismus** roztáčí setrvačník. Tím koná práci. Na stejnou kliku je napojena ojnice pohybující přeháněčem. Ještě dříve, než píst pracovního válce dokončí zdvih a ocitne se ve spodní úvrati, přesune se přeháněč v ležatém válci vlevo. Tím odtud vytlačí horký vzduch. Ten obteče přeháněč doprava do studené části válce. Tam se ochladí, jeho tlak výrazně poklesne i pod úroveň okolního atmosférického tlaku. V celém válci nastane podtlak. Podtlak se potrubím přenese do pracovního válce a "nasaje" píst zpět do válce. Tím se ovšem opět koná práce a roztáčí setrvačník. Jeden pracovní cyklus je dokončen. Setrvačník se však otáčí dál a přesouvá přeháněč opět doprava. Tím se studený vzduch dostává opět vlevo do ohřáté komory, ohřeje se a zvýší svůj tlak. To se stále opakuje a motor se trvale otáčí.



Obrázek 3 Příklad průmyslově vyráběného motoru

5. Stirlingův motor a využití v současnosti

Moderní Stirlingův motor se vyznačuje **dobrou účinností**, **spolehlivostí**, **tichým chodem** a **nižšími emisemi škodlivých plynů**. Hlavní výhodou je skutečnost, že tento motor **může pracovat s nejrůznějšími zdroji tepla** počínaje **sluneční energií** a konče libovolným **fosilním palivem** a **biomasou**. Elektrická účinnost těchto motorů se pohybuje v rozpětí **25 až 33 %**, což je účinnost u motorů o výkonu 1 až 25 kW velmi dobrá. Velké možnosti se Stirlingovu motoru nabízejí v oblasti využití biomasy. Protože zde spaliny nepřicházejí do styku s pohyblivými částmi motoru, nehrozí nebezpečí zadehtování v případě použití plynu získaného zplyněním biopaliv v generátoru s pevným ložem. Bez komplikovaného čištění surového plynu proto mohou být použity zplyňovací generátory tohoto typu v malých kompaktních kogeneračních jednotkách. Velmi zajímavé jsou mikrojednotky vyvinuté americkou firmou Sunpower, vybavené Stirlingovými motory s volnými písty o elektrickém výkonu 1 kW až 25 kW a již provozně vyzkoušené.



Obrázek 4 Celkový pohled na motor

Využití tohoto motoru

Začneme tím, co je nejblíže našemu oboru Technická zařízení budov a to je vytápění a protože Češi jsou národ chatařů, zvolili jsme jako příklad firmu AllTech s.r.o., která se zabývá problematikou realizace technologií vytápění. Ve své nabídce má plynový kotel a generátor na výrobu elektrické energie v jedné jednotce. Je určena pro tepelné provozy s nepřetržitou potřebou dodávky tepelné energie. Jako vedlejší produkt je při plnohodnotném vytápění (nebo chlazení) v objektu zajištěno částečné, nebo úplné pokrytí spotřeby elektrické energie.

Jednotka je vhodná pro: - chaty, chalupy

- malé podniky
- hotely, bazény a lázně

6. Využití Stirlingova motoru ve světě

USA - SES (Stirling Energy Systems)



Obrázek 5 Solární jednotka (USA)

Společnost SES vyvinula solární jednotku pro výrobu elektrické energie s elektrickým výkonem **25 kW** a se špičkovou účinností **29,4%.** Koncentrátorem sluneční energie je parabolické **zrcadlo** o průměru 11,37 metrů. Použitý Stirlingův motor vychází z typu V4-95 Kockums. Pracovním plynem je vodík s tlakem až 20 MPa. Celý projekt je realizován za spolupráce firem United Stirling, Kockums, Volvo a Boeing. Několik těchto zařízení bylo podrobeno **provozním zkouškám v reálném provozu téměř 20 let** a v současné době je připravována sériová výroba. SES totiž uzavřela kontrakt na dodávku jednotek pro jednu solární elektrárnu s výkonem 500 MW s možností rozšíření až na 850 MW a druhou podobnou s výkonem 300 MW s možností rozšíření na 900 MW. Obě elektrárny by měly být postupně realizovány do roku 2025 v poušti Mojave a v Imperial Walley.



Obrázek 6 Solární elektrárna (USA)

7. Průzkum informovanosti veřejnosti o Stirlingově motoru

Po důkladném seznámení s principem práce motoru, s jeho výhodami a nevýhodami a s jeho využitím, jsme se rozhodli udělat **průzkum**, který by ukázal **jak veřejnost zná tento netradiční typ motoru**. Sestavili jsme **anketní dotazník s celkem 18 otázkami**. Celkem jsme získali **532 vyplněných dotazníků**. Přikládáme podrobné výsledky, **rozbory** odpovědí na jednotlivé otázky i vlastní **volně psané názory** respondentů.. Z nich je patrno, že **veřejnost** (i technicky vzdělaná) **Stirlingův motor téměř nezná**, ale po pochopení jeho principu a zejména výhod **má zájem o případný nákup a využívání**. To i koresponduje se snahou některých zahraničních i tuzemských výrobců dodávat Stirlingův motor na trh. Tyto motory by mohly sloužit jako **kogenerační jednotky** (výroba elektřiny a tepla), pro pohon elektrických generátorů nebo přímo i strojů. Hlavní výhoda je ta, že mohou využívat odpadní teplo, obnovitelné zdroje tepla (spalování bioplynu, biomasy, solární záření atd.).

Výsledky ankety:

Dotazník s celkovými výsledky

VOŠ SPŠ Žďár nad Sázavou

Anonymní dotazník - Stirlingův motor

Vyplněný dotazník prosím odevzdejte učiteli nebo na sekretariátu školy. Děkujeme

V některých otázkách je používaná zkratka "OZE":
OZE - zkratka pro **O**bnovitelné **Z**droje **E**nergie:
sluneční (solární), vodní, větrná, geotermální, biomasa, …

1)
Vaše pohlaví: 1 – muž 2 – žena

76%	24%			
1	2	402	mužů	celkem
		130	žen	532 lidí

2)
Váš věk: 19 let - průměrný věk respondentů

3)
Třída: (např. 2.A, 1.BV, …)

Pokud nejste studentem VOŠ a SPŠ, napište, kterou školu navštěvujete nebo Vaše pracovní zařazení:

471 studentů SPŠ 22 studentů VOŠ 27 zaměstnanců 12 ostatní

4)
Jste pro to, aby byly více využívány OZE než v současné době?

	1	2	3	4	5	
1 -určitě **ano,** je to velice důležité pro lidstvo						
2 -myslím, že **ano**	239	206	71	7	5	lidí
3 -**nevím**, nezajímám se o to	45	39	13	1	1	%
4 -spíš **ne**, myslím, že již jsou OZE využívány v dostatečné míře						
5 -určitě **ne**, jejich využívání spíš komplikuje život						

5)
Myslíte si, že do roku 2010 je možné v ČR vyrábět 8 % (skoro 1/10) elektrické energie z OZE?

	1	2	3	4	5	
1 -určitě **ano**						
2 -asi **ano**	39	133	222	106	28	lidí
3 -**nevím**, nedovedu odhadnout	7	25	42	20	5	%
4 -spíš **ne**						
5 -určitě **ne**						

6)
Které skutečnosti nejvíce brání rozmachu použití OZE (v domácnostech, v průmyslu, …) ?

	1	2	3	4	5	
1 -malá podpora státu (dotace, …)						
2 -nezájem veřejnosti	124	166	107	122	9	lidí
3 -vliv velkých energetických společností	23	31	20	23	2	%
4 -vysoká prvotní investice						
5 -závist okolí (sousedů, …)						

7)
Slyšel(a) jste někdy pojem Stirlingův motor?

		1	2	3	
1 -**ne**, slyším (vidím) to poprvé					
2 -**ano**, myslím, že jsem to někde již zaslechl(a)	kde:	287	99	142	lidí
3 -**ano**, slyšel(a) jsem o tom		54	19	27	%

8)

Víte, co to je Stirlingův motor?

	1	2	3	4	5	
1 -vůbec **nevím**						
2 -spíš **nevím**, slyšel(a) jsem jen název	265	92	61	57	53	lidí
3 -myslím, že **vím**, co to je	50	17	12	11	10	%
4 -**ano**, vím co to je						
5 -**ano**, vím co to je, a znám jeho princip (asi bych dovedl(a) i vysvětlit)						

9)

Tipněte si, kde byl vynalezen první Stirlingův motor, pokud víte, že se jmenuje po svém vynálezci:

	1	2	3	4	5	
1 -Německo						
2 -Švýcarsko	210	186	99	24	9	lidí
3 -Skotsko	40	35	19	5	2	%
4 -Česká republika						
5 -Maďarsko						

10)

Tipněte si, v kterém století byl tento motor sestrojen poprvé:

	1	2	3	4		
1 -18. století						
2 -19. století	54	252	198	24		lidí
3 -20. století	10	48	38	5		%
4 -21. století						

11)

Věděl(a) byste nebo dokážete si tipnout, co využívá Stirlingův motor pro svoji činnost?

	1	2	3	4	5	
1 -vodní energii						
2 -spalování známých paliv (benzin, nafta, plyn, …)	75	59	54	301	39	lidí
3 -větrnou energii	14	11	10	57	7	%
4 -rozdíl teplot (ohřev a ochlazení)						
5 -rozdíl světla (světlo a tma)						

12)

Zajímalo by Vás zařízení velikosti řádově plynového sporáku, které by bylo nezávislé na vnějších energetických sítích (plyn, elektřina) a přitom by zajišťovalo teplo a elektrickou energii pro Vaši domácnost?

	1	2	3	4	5	
1 -**ano**, to by mě velice zajímalo						
2 -**ano**, zajímalo by mě to	158	212	95	42	21	lidí
3 -zajímá mě to jen jako informace pro rozšíření znalostí, jinak **ne**						
4 -**ne**, asi by mě to nezajímalo	30	40	18	8	4	%
5 -**ne**, určitě by mě to nezajímalo						

13)

Víte, že v kraji Vysočina existuje firma, která se zabývá vývojem Stirlingova motoru pro využití v domácnosti jako zdroje tepla a elektřiny?

	1	2	3	4	5	
1 -**ne**, to nevím a nezajímá mě to						
2 -**ne**, to nevím a zajímalo by mě to	171	318	27	9	3	lidí
3 -**ano**, myslím, že jsem něco o tom slyšel(a)	32	60	5	2	1	%
4 -**ano**, vím						
5 -**ano**, vím a zajímal(a) jsem se již o to						

14)

Proč myslíte, že se o Stirlingově motoru neučí na základních školách?
(Ve fyzice se na ZŠ učí principy spalovacích motorů a parní stroje.)

	1	2	3	4	5	
1 -je asi složitý na pochopení						
2 -je to bezvýznamný stroj	237	36	81	105	68	lidí
3 -je to záměr (jakýkoli)	45	7	15	20	13	%
4 -zapomnělo se na něj při sestavování učebních osnov						
5 -má to jiný důvod. Napište jaký:						

15)

Není zarážející, že při dnešním stupni vývoje techniky a materiálů není princip Stirlingova motoru více využíván? (Pro Vaši informaci - pro pohon Stirling. motoru lze využít např. spalování biomasy, sluneční energie, …)

	1	2	3	4	5	
1 -**ano**, je to zarážející						
2 -**ano**, je to zarážející, ale asi to není tak významné						
3 -**nevím**, nedovedu posoudit	158	76	245	41	8	lidí
4 -**ne**, protože je to stroj s nízkou účinností	30	14	46	8	2	%
5 -má to jiný důvod. Napište jaký:						

16)

Myslíte si, že by bylo možné jednoduchý Stirlingův motor postavit v domácích podmínkách?

	1	2	3	4	5	
1 -**ne**, to určitě není možné						
2 -spíš **ne**	30	245	126	76	50	lidí
3 -**ano**, ale musela by k tomu být k dispozici dobře vybavená dílna						
4 -**ano**, lze to, ale je třeba veliké dovednosti	6	46	24	14	9	%
5 -**ano**, jednoduchý stroj lze bez problémů postavit						

17)

Zapůsobil na Vás dotazník tak, že se teď budete více zajímat o to, co je to Stirlingův motor?

	1	2	3	4	5	
1 -určitě **ne**, vůbec mě to nezajímá						
2 -spíš **ne**						
3 -asi **ano** (alespoň pro rozšíření znalostí, …)	75	172	181	69	30	lidí
4 -**ano**, je to zajímavé	14	33	34	13	6	%
5 -určitě **ano**, velice mě to zajímá						

18)

Co si myslíte o Obnovitelných Zdrojích Energie (OZE)? (jakýkoli vlastní názor, či jiné doplnění):

Názory a odpovědi zpracovány v další tabulce

Přehled skupin zúčastněných respondentů

Počty respondentů

	ročník / obor	1.	2.	3.	4.	celkem
SPŠ	A	18	32	30	10	90
	B	27	27	24	23	101
	C	29	20	12	22	83
	D	22	22	23	15	82
	E	28	32	31	24	115
	celkem	124	133	120	94	471

	ročník / obor	1.	2.	3.	celkem
VOŠ	AV	4	0	0	4
	BV	0	18	0	18
	celkem	4	18	0	22

		počet
VOŠ a SPŠ	studenti SPŠ	471
	studenti VOŠ	22
	zaměstnanci	27
ostatní		12
celkem všech respondentů		532

Otázka č.18: vlastní názory dotázaných

- Měly by se **používat co nejdřív**, ale někteří lidé by si museli sáhnout hluboko do kapsy.

- Při dnešní spotřebě energie není možné získat větší potencionální podíl OZE na energetickém trhu

- OZE ano, ale bez základních zdrojů nemohou ve vyspělé společnosti stačit. Nutná je v budoucnu **termojaderná syntéza.**

- V budoucnosti by mělo dojít **k jejich širšímu využití**. Veřejnost by o nich měla být **více informována**. Stát by měl přispět vyššími dotacemi.

- Oni jsou pouze tzv. obnovitelné - **spalovat biomasu** a vyrábět z potravin palivo **je zločin** vůči hladem umírajícím. Slunce a vítr nejsou vždy a všude atd., např. na Marsu bychom s tím těžko uspěli.

- Nemůže se stavět velká ekonomika na tom jak zafouká **vítr.**

- Preferuji spíše **sluneční energii. Vodní a větrná spíše "škodí".**

- Je důležité se jimi zabývat, ale také se zabývat **úsporou energie.**

- Jsou ekologičtější, plně podporuji. Na svém domě využívám **solární panely.**

Naše názory k anketě

Vzhledem k tomu, že ze všech stran nás „straší“ globální oteplování, roztávání ledovců a podobné jevy, tak si myslíme, že je nutné začít **vážně využívat OZE. Stirlingův motor je jistě jedna z variant, jak energie neobnovitelné nahradit, ale určitě ne jediná**. Práce na anketě byla velmi zajímavá, protože jsme se dozvěděli mnoho názorů na zadané téma. Jako další část práce jsme se rozhodli pokračovat ve **vývoji sestavy**, která by využívala Stirlingův motor.

8. Výhody a nevýhody Stirlingova motoru

Výhody Stirlingova motoru

Pracuje **tiše, nemá choulostivé rozvody** a obtížně vyrobitelné součásti, je **nenáročný na obsluhu**, je **hospodárný** a při vhodné volbě paliva může být velice **ekologický** (plyn, biomasa), dokonce je možné vyloučit spalování a použít jiného zdroje tepla (**sluneční**

záření). Může využít chladící vodu k ústřednímu vytápění, **nezatěžuje životní prostředí**, má velmi **malé opotřebení**. Další jeho předností je **levná výroba.**

Nevýhody Stirlingova motoru

Pokud je k provozu užíváno odpadní tepelné energie nebo sluneční, je na nich **závislý.** To znamená, že poběží jen v případě, že tyto energie jsou momentálně dostupné. V případě **chlazení vzduchem** potřebujeme **velkou chladící plochu**. Je poměrně **málo výkonný** na jednotku hmotnosti, **obtížně regulovatelný** a pouze **pomaluběžný**. Vyžaduje velmi **intenzivní chlazení**, ideální je vodou.

9. Vývoj sestavy pro využití Stirlingova motoru

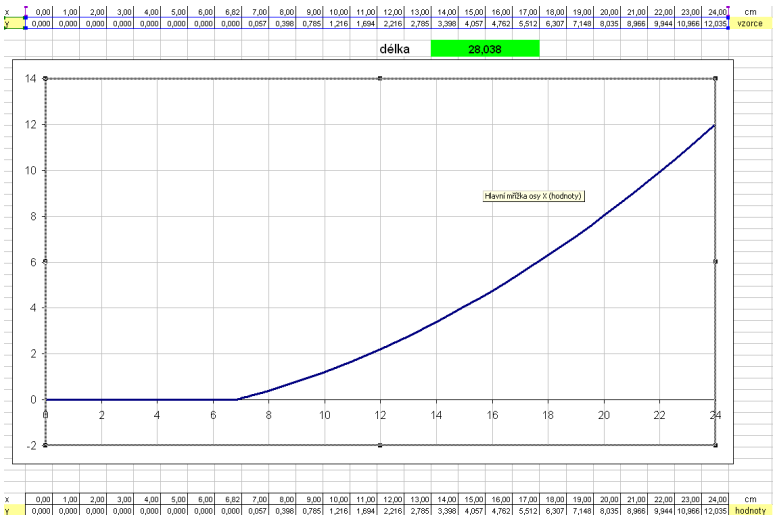
V této části projektu jsme se rozhodli vyrobit následující sestavu, která by využívala sluneční záření k pohonu Stirlingova motoru. Zařízení se skládá z následujících částí:

1. **Stirlingův motor (SM)**
2. **parabola**
3. **vodní čerpadlo, nebo elektrický generátor**

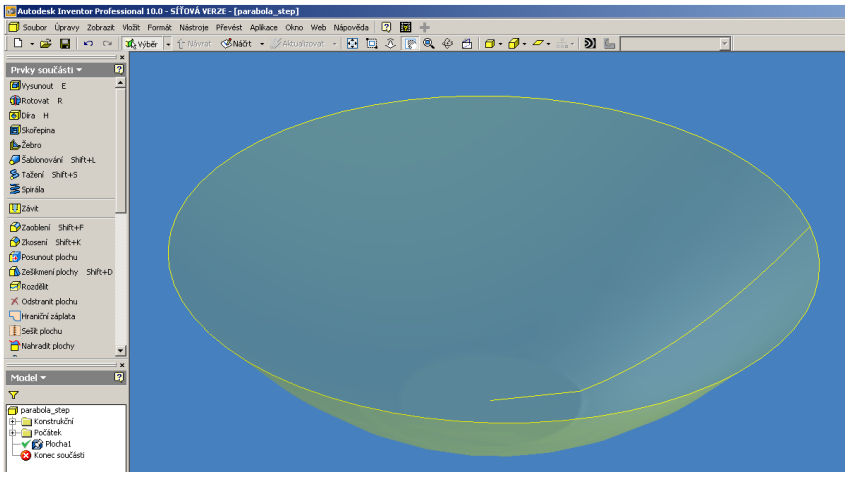
Škola vlastní celkem asi **6 SM**, některé byly zakoupeny, jiné vyrobeny ve školních dílnách. Pro náš účel jsme zvolili zakoupený Stirlingův motor, který má oba písty (pracovní i přeháněcí) v jedné ose. Zahřívání motoru bude soustředěním slunečního záření parabolickým zrcadlem (tedy obdoba zařízení viz obr. 5 a 6).

Model paraboly lze zakoupit, ale tato parabola nemá zarovnané dno a dochází k ohřevu části motoru, která slouží k chlazení a tím se snižuje výkon SM. Proto jsme se rozhodli s pomocí našich školních dílen a kovotlačitelské firmy **vyrobit parabolu**, která nám bude vyhovovat a bude mít větší celkový účinek.

Nejdříve jsme v Excelu **spočítali souřadnice** bodů X a Y paraboly (viz obr. 7). Na základě těchto výsledků jsme v 3D modeláři Invertoru vytvořili **3D virtuální model paraboly** (viz obr. 8), který byl následně vyroben **ze dřeva na CNC frézce** ve školních dílnách (viz obr. 9). Tento dřevěný model byl použit při výrobě **paraboly v kovotlačitelské firmě**, kde na se podle něho na kovotlačitelském soustruhu natvarovala parabola z hliníkového plechu tloušťky 2 mm. Ta se následně **vyleštila** do zrcadlového lesku, vyvrtal se středový otvor pro Stirlingův motor a další otvory pro uchycení SM (viz obr. 11).



Obrázek 7 Matematický model paraboly



Obrázek 8 Virtuální model paraboly v Inventoru



Obrázek 9 Model paraboly ze dřeva



Obrázek 10 Parabola



Obrázek 11 Stirlingův motor

Vyrobená a vyleštěná parabola se smontovala se Stirlingovým motorem (viz obr. 11) a umístila na **polohovací** (natáčecí) **zařízení**. Toto zařízení se používá pro nastavení slunečních fotovoltaických panelů kolmo na sluneční paprsky a pro sledování Slunce při jeho pohybu po obloze. Na noc a při oblačném počasí je plocha panelů vodorovná. Takové zařízení (viz obr. 12) je tedy i vhodné pro nastavování naší paraboly vzhledem k poloze Slunce.



O



Pro demonstraci čerpání vody
nádrže do druhé. Na obrázku 13 j



Obrázek 13 Čerpadla

Aby se zvýšila intenzita chlazení motoru, **bude čerpaná voda proudit okolo chladících žeber motoru.** K tomu jsme **navrhli obtékání motoru a jeho utěsnění** v chlazené oblasti. Navržený kryt je nyní vyráběn, viz obrázek 14 (bílá část).



Obrázek 14 Stirlingův motor s nasazenou částí pro chlazení

Teoretický příkon zařízení.

Sluneční konstanta (solární konstanta) je tok sluneční energie procházející plochou 1 m², kolmou na směr paprsků, za 1 s ve střední vzdálenosti Země od Slunce měřený mimo zemskou atmosféru. Konstanta zahrnuje celé spektrum slunečního záření, nejen viditelné světlo. Tato konstanta má hodnotu asi **1360 W/m².** Konkrétní množství sluneční energie dopadající v daném místě a čase na povrch je taky ovlivněno stavem atmosféry (nečistoty, počasí) a zeměpisnou šířkou**.** V literatuře [2] se uvádí, že asi 35% energie se odráží od atmosféry a 20% je absorpcí pohlceno. Zbývá tedy 45%, což je **612 W/m²**. Naše parabola má velký průměr 480 mm a vnitřní rovinnou plochu (viz obrázky 8 až 11) o průměru 140 mm. Účinná plocha mezikruží je 0,1655 m². Po vynásobení přibližně spočítaným tokem zářivé energie dostáváme **max. teoretický příkon 101 W**.

10. Shrnutí

Sluneční záření je koncentrováno vyrobenou **parabolou** na horkou část **Stirlingova motoru** (termočlánkem naměřena teplota až 200°C). Stirlingův motor pohání připojené **čerpadlo**, které čerpá vodu a tuto vodu používá také ke chlazení studené části Stirlingova motoru. Celková účinnost zařízení se dá spočítat jako poměr práce vykonané při čerpání vody k dodávané energií dopadajícího slunečního záření na parabolu.

Popsané zařízení bude umístěno na **polohovací zařízení SF2**, které se automaticky otáčí za Sluncem. Celá sestava tedy demonstruje následující řetězec:

Energie slunečního záření – fokusace parabolou – ohřev Stirlingova motoru – práce čerpadla

Zařízení je příkladem **čerpací jednotky na vodu poháněné slunečním zářením.** Tato jednotka může najít uplatnění **v zemích s dlouhou dobou slunečního svitu a s potřebou čerpání vody například k zavlažování.**

11. Závěr a diskuse

Podařilo se zkompletovat **demonstrační čerpací jednotku** na sluneční záření s využitím Stirlingova motoru. Jednotka muže sloužit jako **učební pomůcka** v environmentálním vzdělávání a výchově.

Jako pokračování tohoto projektu navrhujeme:

1) **zdokonalit vlastní Stirlingův motor** do podoby průmyslově vyráběného typu. Zde je možné využít například některý z dvouválcových vyráběných kompresorů a doplnit ho regenerátorem tepla.
2) zajistit produktivní **výrobu parabolických koncentrátorů slunečního záření**
3) navrhnout **využití průmyslově vyráběného čerpadla** s malým vlastním odporem (malé ztráty třením)
4) navrhnout **konstrukční řešení** na propojení komponent ad 1) až ad 3) s polohovacím zařízením SF2
5) celé takto průmyslově vyvinuté zařízení **odzkoušet**, **změřit** jeho celkovou **účinnost** a **provozuschopnost**
6) **provést ekonomický rozbor** nasazení této čerpací jednotky

Protože do konce studia na střední škole nám zbývá více než jeden rok, budeme se navrženými činnostmi zabývat a pokusíme se je dovést do skutečné průmyslového nasazení.

Tato práce nám umožnila poznat velmi zajímavý a málo známý motor a jeho aplikace. Při práci na projektu jsme se naučili nejen obhajovat a zdůvodňovat technická řešení, ale také spojovat si dílčí školní znalosti do celku. Myslíme si, že práce na projektech je důležitou součástí školního vzdělávání a každý žák a student by se měl zapojit do aspoň jednoho takového projektu.

12. Seznam použité literatury

[1] DARLINGTON, Roy., STRONG, Keith. STIRLING AND HOT AIR ENGINES. The Crowood Press, Ramsbury, 2005.

ISBN 1-86126-688-X

[2] KAMINSKÝ, Jaroslav., VRTEK, Mojmír. Obnovitelné zdroje energie, skriptum VŠB – TU Ostrava, Fakulta strojní, 1998

ISBN 80-7078-445-8
[3] VAN ARSDELL, Brent H. AROUND THE WORLD BY STIRLING ENGINE
Američan Stirling Copany, 2006
ISBN 0-9713918-0-7
[4] Internetové zdroje
www.stirlingengine.com
www.wikipedia.com
www.stirling.gov.uk
www.stirling.cz

Příloha č. 1

Přehled Stirlingových motorů na škole

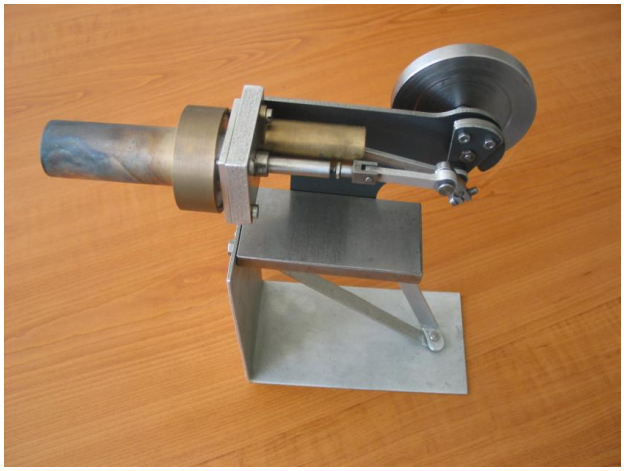
Stirlingův nízkoteplotní motor – učební pomůcka



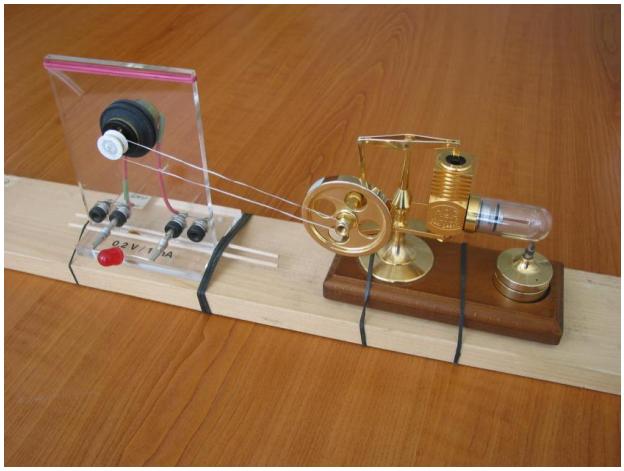
Stirlingův termoakustický motor – učební pomůcka



Stirlingův motor – vyroben ve školních dílnách



Stirlingův motor s demonstrací výroby elektrické energie



Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Santiniho poutní kostel – virtuální model

David Okurka

Vyšší odborná škola a Střední průmyslová škola
Studentská ulice 1, Žďár nad Sázavou



Prohlašuji tímto, že jsem soutěžní práci vypracoval samostatně pod vedením **Dr. Ing. Josef Příhody** a dalších konzultantů uvedených v práci a uvedl v seznamu literatury veškerou použitou literaturu a další informační zdroje včetně Internetu.

Na úvod bych chtěl poděkovat všem, kteří mi s prací pomáhali. Jsou to:

Dr. Ing. Josef Příhoda, ředitel VOŠ a SPŠ Žďár nad Sázavou, metodické vedení práce
Ing. Arch. Martin Zezula, SANTIS a. s., architectural design studio Žďár nad Sázavou, konzultace v oblasti konstruování
Martin Kocmánek, projektant SANTIS a. s., architectural design studio Žďár nad Sázavou, konzultace v oblasti konstruování
Zdenek Musil, projektant SANTIS a. s., architectural design studio Žďár nad Sázavou, konzultace v oblasti konstruování

Úvod

Jsem studentem 4. ročníku oboru Technické lyceum. Z nabízených témat dlouhodobé praktické maturitní zkoušky jsem si vybral téma Santiniho poutní kostel na Zelené hoře ve Žďáře nad Sázavou. V zadání byl požadavek nejen na historický popis památky, ale také na vytvoření digitálního modelu kostelu. Byl jsem si vědom, že tento projekt bude velmi náročný, ale přesto mne zadání oslovilo.

Historie kostelu

Poutní kostel Sv. Jana Nepomuckého na Zelené hoře - historie

Poutní kostel Svatého Jana Nepomuckého na Zelené hoře (obr. 1) u města Žďár nad Sázavou patří mezi nejvýznamnější stavby a ukazuje vyvrcholení tvorby geniálního architekta a barokního stavitele Jana Blažeje Santiniho-Aichla, příslušníka již třetí generace italské umělecké rodiny usazené od 17.století v Praze.

Kostel byl postaven ve slohu barokní gotiky. Stavba poutního kostela probíhala v letech 1719–1722 na náklad Václava Vejmluvy, který byl v letech 1705-1738 opatem zdejšího cisterciáckého kláštera.K tomuto rozhodnutí jistě přispěl vztah žďárského kostela k místu, kde Jan Nepomucký vyrůstal. Původní mateřské sídlo cisterciáckého řádu, odkud mniši v roce 1251 do Žďáru přišli bylo v letech 1144-1420 právě na Zelené hoře u Nepomuku. Smyslem stavby bylo především oslavit Jana Nepomuckého, jako mocného patrona a světce. V době vzniku stál na travnatém vršku, který Václav Vejmluva pojmenoval právě Zelená hora (dříve se nazýval Černý les nebo Strmá hora) podle vrchu u Nepomuku. Do nedávné doby byla stavba obklopena vysokým borovým lesem (obr.2), který však byl odstraněn (obr.3), aby bylo na kostel tak jako původně vidět i z velké dálky.

Stavba byla zahájena v srpnu roku 1719, kdy byl v hrobě blahoslaveného Jana Nepomuckého v katedrále sv. Víta nalezen jeho domněle neporušený jazyk a kdy byla zahájena příprava na jeho kanonizaci. V květnu 1720 se konala slavnost položení základního kamene. Kostel byl vysvěcen 27.září1722 a stal se první velkou svatyní, zasvěcenou sv. Janu Nepomuckému. Dostavba ambitů spadala do doby o málo pozdější.

Centrální stavba kostela i ambitů je jednotně projektovanou a budovanou stavbou. Tvarosloví stavby je velmi minimalistické a nesmírně účinné. Spojuje barokní a gotické prvky, čímž jasně odkazuje k době, v níž žil, působil a byl umučen Jan Nepomucký. Zatímco v gotické

době hledaly duše Boha v popírání tělesnosti a neustálým pohledem vzhůru, v baroku hledí Boha postihnout skrze pohyb, který dal světu.

Chrám je postaven na půdorysu ve tvaru pěti cípé hvězdy (obr.4). Jeho konstrukce kostela vychází z geometrie kruhu, přičemž mnohonásobně opakuje číslo pět jako odkaz na pět hvězd Jana Nepomuckého, které se podle legendy objevily nad tělem mrtvého světce a dále pak na pět písmen V ve stavebníkově jméně (WEJMLVWA). Dobře tak ukazuje způsob, kterým Santini obvykle konstruuje své stavby prakticky pouze za použití kružítka, přičemž buduje celou stavbu pomocí úseků kružnic, jejichž poloměrem jsou zpravidla násobky základního modulu stavby. Používá přitom určitých číslic, které mají ikonografický význam. V jeho stavbách se proto často objevují čísla tři, pět, sedm, dvanáct atp. Na zelené hoře je to vedle čísla pět ještě číslo tři, které je poukazem na Boží trojici a potom ve formě hvězd číslo šest, které je zase odkazem mariánským. Mj. Protože Nepomuk byl chápán i jako obzvláštní ctitel Panny Marie. Po obvodu kostela se tak střídá pět hrotitých kaplí s trojúhelníkovým půdorysem a pět kaplí oválných. Do celého areálu vede pět vchodů, uvnitř chrámu je pět kaplí a pět oltářů.Od příchodu prvních cisterciáků uplynulo v té době pět století, pět písmen má i latinské TACUI – mlčel jsem a dále již zmiňované jméno Václav Wejmluva s 5 V. To navazuje na legendu, podle které světec neprozradil zpovědní tajemství královny Žofie, manželky krále Václava IV, a proto ho král nechal nejprve krutě mučit a potom svrhnout do Vltavy. Dnes však víme, že smrt Janova měla především politické pozadí kvůli jeho sporům se samotným králem.

Celý kostel přitom svou konstrukcí připomene řez chórem gotické katedrály a jeho opěrným systémem. Centralita kostela pak buduje vertikálu, jako určitý symbolický poukaz. Tato vertikála totiž nutně táhne pohled návštěvníků vzhůru k nebi.

Ve vnitřním prostoru kostela jsou oválné obvodové kaple spojeny s ústředním válcovým prostorem užšími lomenými oblouky. Ústřední prostor je zaklenut kopulí s lunetami, která je nesena deseti pilíři. V prvním patře jsou tribuny odpovídající přízemním kaplím a ve druhém patře je vedena kolem paty klenby desetidílná galerie. Štuková dekorace se omezuje na motivy protínajících se a přesekávaných žeber připomínajících gotické síťové klenby.

Ve vrcholku kopule je umístěn velký jazyk, symbol sv. Jana Nepomuckého, obklopený kruhem plamenů, z něhož šlehají paprsky. Jedná se o symbol vítězné zbraně-meče mučedníka a protějšek hvězdy, zdůrazněné půdorysem svatyně

Tak jako na jiných stavbách, použil i zde Santini k budování prostoru světla a působivých kontrastů.

Vychází přitom z barokní interpretace světla, jako symbolu Boží přítomnosti. Hovoříme proto o „sakrálním světle“. Použití gotických oken symbolizujících opět meč nebo ruce sepnuté k modlitbě, půdorysné formy hrotů hvězd, gotizující štukové žebroví, ve kterých je možno spatřovat symbolickou narážku na spojení se středověkým mateřským klášterem na Zelené hoře v Nepomuku. Množství oken poskytuje dostatek světla, které barokně rozšiřuje celý prostor a dává možnost vyniknout sochařské výzdobě, pojaté jednoznačně v barokním charakteru.Díky zajímavému umístění oken není v celém kostele tmavý kout. Okna mají tři různé tvary. Okna ve tvaru jazyka jsou zakomponována u lucernových kaplí nad vstupními vestibuly. Připomínají meč v pochvě a jazykem- mečem dosáhl světec svého duchovního vítězství. Okna ve tvaru biskupské mitry se jako opatský odznak a součást erbu vztahují ke stavebníkovi. Okna v podobě sférického rovnostranného trojúhelníku symbolizují Nejsvětější Trojici. Světelné poměry interiéru kostela působí zvláštní paradox. Zatímco centrální prostor kostela je osvětlen pouze zprostředkovaně, jsou boční „ochozové“ části kostela hojně zality světlem. Toto světlo proniká do centrálního prostoru kostela lomenými oblouky umístěnými v místech, kde by správně měly být umístěny nosné pilíře. Tento paradox, ještě podtržený štukovou výzdobou která vytváří paradoxní vztah mezi stěnou a klenbou na této stěně spočívající má na návštěvníky působit dojmem, že stěna je pouhou skořápkou a stavba je vlastně budována a nesena světlem. Celý chrám pak je vlastně chápán také jako jakýsi relikviář, v němž je uložena relikvie jazyka svatého Jana Nepomuckého.

Hlavní oltář (obr.5) na východní straně je umístěn do vysoké arkády. Vrchol oltáře sahá až k zábradlí galerie druhého patra. Na hlavním oltáři je socha světce, stojícího na zeměkouli a opticky stoupajícího vzhůru. Pět osmicípých cisterciáckých hvězd představuje 5 kontinentů, na kterých bylo šířeno křesťanství. Zeměkoule(nebeská klenba) ozdobená pěti hvězdami je nesena třemi anděly, kompozičně rozmístěnými do tvaru písmene V. Na kouli stojí postava sv. Jana Nepomuckého. Tato plastika je dílem Řehoře Thenyho. Jeho prací jsou i reliéfy na nosítkách pro poutní stříbrnou sochu sv. Jana Nepomuckého z roku 1729 od pražského zlatníka J. Diesbacha, která se žel po roce 1784 ztratila. Volba sochařů, kteří mají blízko k dílně Matyáše Bernarda Brauna (Thény byl Braunovým tovaryšem) zde nebyla náhodná, neboť Santini mnohokrát spolupracoval na svých realizacích právě s Braunem, jehož sochařský jazyk mu zjevně velmi vyhovoval.

Hvězdy se nesčetněkrát objevují na exteriéru i v interiéru celého poutního areálu. Kromě osmicípých hvězd najdeme v kostele i hvězdy šesticípé – svatojánské a deseti a dvanácticípé – mariánské. Řezby pěti andělů na hlavním oltáři (které svým počtem opět poukazují k Janu Nepomuckému) a čtyři evangelisté jsou prací chrudimského sochaře Jana Pavla Čechpauera z

let 1725–1727. Od chrámové lodi je oltář oddělen ručně kovanou trojbokou mříží. Dvě bílé münsterberské orlice jakoby nadnášely svými křídly oltářní stůl. Mezi nimi je znak zakladatele kláštera Bočka z Olbřan. Oltáře po stranách kostela jsou zasvěceny čtyřem evangelistům. Matoušovi s knihou (Obr.6), Markovi se lvem (obr.7), Lukášovi s býkem (obr.8) a Janovi s orlem (obr.9). Všechny oltáře jsou vyrobeny z lipového dřeva a mají mramorovou imitaci. Autorem sochařské výzdoby chrámu je Jan Pavel Čechpauer (1670 – 1726) z Chrudimi, dokončoval ji Ignác Rohrbach (1691 – 1747), který po jeho smrti převzal dílnu. Provádějícím stavitelem byl Donát Marazzi a truhlářské práce zhotovil Jan Ludvík Pešinský z Hradce Králové.

Kolem kostela byl pak vybudován na půdoryse složeném z deseti úseků kružnic (deseticípá hvězda) prstenec ambitů (obr.10), který je pročleněn pěti pětiúhelnými kaplemi a pěti branami. Byl navržen architektem současně s chrámem, k jeho dokončení však došlo až v r. 1769. Při právě probíhající opravě byly nalezeny základní kameny, vymezující polohy oblouků, což dokazuje, že jejich zaměření bylo provedeno již při zakládání kostela. Stanové střechy nad branami nesly alegorické sochy, tři z nich se zachovaly. Kdysi bohatá ozdoba štukových obrazců a žeber na klenbách se dochovala jen v mizivém rozsahu.

Střechy kaplí původně vrcholily pěti pylony, které opět poukazovaly na význam světla a symbolizovaly věčnost. Tyto ambity s kaplemi sloužily pro poutnické modlitby i pro ochranu poutníků před nepřízní počasí. Nejen samotný kostel, ale i tento prstenec ambitů svým architektonickým řešením jasně dokládá Santiniho obrovskou architektonickou a tvůrčí potenci.

Období největší slávy zažil kostel na Zelené hoře spolu s klášterem v roce 1735, kdy bylo slaveno pětisté výročí založení kláštera. Bohužel však již o dva roky později klášter vyhořel a rok poté umírá opat Václav Vejmluva. Katastrofu dovršil další zhoubný požár v roce 1784, kterému padl za oběť nejen konventní kostel, ale také kostel sv. Jana Nepomuckého na Zelené hoře. V říjnu roku 1784 byl pak císařem Josefem II. zrušen klášter, včetně poutního kostela. Vnitřní inventář kostela byl přenesen do okolních farností. Teprve za působení bývalého klášterního mnicha Bonifáce Procházky začala obnova kostela, která s přestávkami trvala až do roku 1830. Za tehdejšího správce farnosti, vlasteneckého buditele P.Matěje Sychry bylo dosaženo znovuobnovení kostela, které souviselo se založením hřbitova ve vnitřním areálu. Na hřbitově se již v dnešní době nepohřbívá, aby vnitřní prostor ambitů získal původní barokní podobu. Výjimečné hodnoty tohoto poutního kostela vedly k tomu, že v roce 1994 byl zapsán jako první solitérní stavba v českých zemích do seznamu světového kulturního dědictví UNESCO.



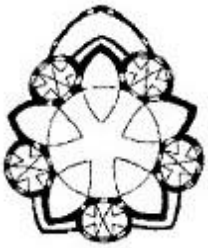
Obr.1-Poutní kostel sv.Jana Nepomuckého ve Žďáře nad Sázavou



Obr.2-Poutní kostel před vykácením lesa



Obr.3-Poutní kostel po vykácení lesa



Obr.4-Půdorys ve tvaru 5-cípé hvězdy



Obr.5-Hlavní oltář



Obr.6-Evangelista Matouš s knihou



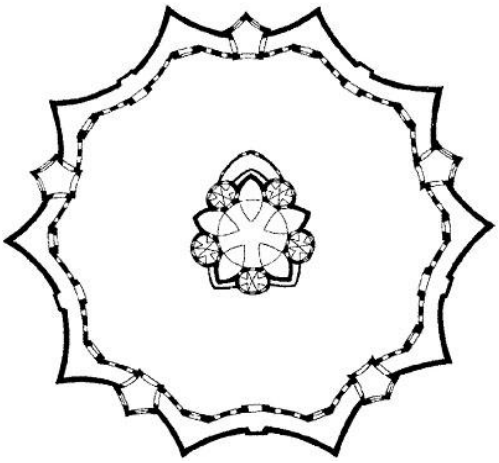
Obr.7-Evangelista Marek se lvem



Obr.8-Evangelista Lukáš s býkem



Obr.9-Evangelista Jan s orlem



Obr.10-Půdorys kostela s prstencem ambitů

Významné osobnosti poutního kostelu

Cisterciácký řád



Zakladatel kláštera Bořek z Olbřan

V klášterech nejstarší řehole, založené sv. Benediktem, se koncem jedenáctého století objevila snaha po reformaci a návratu k prostému řeholnímu životu. Pod vedením sv. Bernarda odešla část mnichů z francouzského kláštera v Molesmes do údolí zvaného Clairvaux, aby tam vybudovala středisko nového života. Z Cistercia, jak se pak toto místo přejmenovalo, odcházeli bratří do jiných pustých krajů, aby je vzdělali a budovali nové kláštery. Ty byly vždy spojeny s mateřským klášterem a vznikla tak mimořádná společenství celoevropsky provázaných řádových středisek. Koncem dvanáctého století se již jednalo o více než pět set nových klášterů. Členové řádu se ve smyslu hesla Ora et labora (Modli se a pracuj) zabývali duchovním a hospodářským povznesením svěřených míst.

Jan Blažej Santini Aichl (1677 – 1723)



Architekt Jan Blažej Santiny Aichl

viz.samostatná kapitola v dalším textu.

Václav Vejmluva (1670-1738)



Václav Vejmluva

Narodil se jako syn klášterního sládka na Starém Brně. V devatenácti letech se stal členem řeholního bratrstva ve Žďáře. Po studiích v Praze byl v roce 1705 zvolen opatem a začal naplno rozvíjet svoji cestu k životnímu cíli. Tou byla obnova a rozkvět svěřeného kláštera. Po vyřešení hospodářské situace získal ke spolupráci architekta Jana Santiniho Aichla. Spolu projektovali přestavbu celého areálu a především novostavbu poutního kostela sv. Jana Nepomuckého na Zelené hoře. Se stejným úsilím se opat také věnoval rozvoji hospodářství a řešení sociálních problémů. Díky úspěchům v této oblasti byl povýšen na císařského radu. V r.1735 bylo jeho úsilí završeno oslavami pětisetletého trvání kláštera. Vejmluvův triumf třicetileté práce však trval pouze krátký čas. Devátého května 1737 zničil požár celé dílo a do roka umírá také sám Vejmluva.

Jindřich zvaný Řezbář (1242 – po r. 1300)

Syn kameníka Ekwarda, pracujícího na stavbě žďárského kláštera. Stal se klerikem, ale ve svých dvaceti pěti letech z kláštera utíká. Zpět se kajícně vrací až v roce 1294. Vedle řezbářských prací pro kostel se především věnoval sepsání historie založení žďárského kláštera ve dvaceti kapitolách, latinsky veršované kroniky CRONICA DOMUS SARENSIS.

Jan Nepomucký (1340 – 1393)



Poutní obrázek sv. Jana Nepomuckého

Jan Nepomucký je světec, který bývá uctíván jako mučedník zpovědního tajemství. Legenda vypráví, že nechtěl vyzradit zpovědní tajemství manželky krále Václava IV., královny Žofie a proto tedy musel zemřít. Pravda je však jiná. Po studiích byl Jan jako velmi nadaný kněz pověřen samotným arcibiskupem Janem z Jenštejna úkolem generálního vikáře. Jan svůj úkol plnil velmi svědomitě a poctivě a to právě nevyhovovalo králi Václavu IV., který chtěl do kláštera v Kladrubech dosadit jednoho ze svých oblíbenců. Tak se dostal Jan s králem do sporu. Byl uvězněn a krutě mučen. Nakonec ho spoutaného a s roubíkem v ústech shodili z Karlova mostu do Vltavy.

Řehoř Theny (1695 –1759)

Sochař, pocházející z jižních Tyrol se po vyučení a cestách po Evropě stal pomocníkem v dílně sochaře Matyáše Brauna. V roce 1729 ho opat Vejmluva přijímá do žďárské klášterní dílny a svěřuje mu výzdobu dostavěného konventního kostela. Po šťastných chvílích stavby a narození prvního syna však přichází tragédie. Sochařova žena Marie Klára umírá a umělec vtiskuje její podobu na věčné časy do tváře cherubína na hlavním oltáři. Toto pouto a osobní lidský vztah s opatem Vejmluvou spojilo Thenyho s klášterem na dlouhých třicet let. Vytvořil zde řadu mistrovských plastik, které snesou srovnání s největšími díly své doby. Na sklonku života se Theny vrací znovu do Žďáru, aby napravil ničivé následky požáru.

Otto Steinbach (1751 – 1825)

Poslední žďárský opat, spisovatel, historik a osvícenec. Po požáru v roce 1784 se sám zasadil u Josefa II o zrušení vlastního kláštera. Dobrodružná a nestálá povaha ho přivedla daleko od svěřeného údělu, a tak umírá zapomenut v německém Bonnu.

P. Matěj Sychra (1776 – 1830)

Po působení v Jimramově se stal správcem farnosti v Zámku Žďáře. Spisovatel a hudebník, který se zasloužil o opravu kostela na Zelené hoře. Obnovení bohoslužeb bylo podmíněno zřízením hřbitova uvnitř areálu. Na opravu organizoval sbírky a věnoval celou svoji pozůstalost.



Původní Santiniho nákres kostelu



Kardinál František Dietrichstein

Jan Blažej Santini-Aichel (3. února 1677 – 7. prosince 1723)

Jan Blažej Santini-Aichel (obr.11) se narodil 3.února 1677 v Praze. Byl to významný český architekt italského původu, který se proslavil svým jedinečným stylem nazývaným barokní gotika.

Narodil se na den sv. Blažeje jako nejstarší syn do vážené rodiny pražského kameníka Santina Aichela a druhého dne byl pokřtěn v chrámu sv. Víta jako Jan Blažej Aichl. Narodil se s tělesnou vadou – byl na část těla ochrnutý a chromý, což mu bránilo úspěšně pokračovat v otcově kariéře a převzít po něm kamenickou dílnu. Kamenictví se přesto vyučil (stejně jako bratr František), ale studoval také malířství, pravděpodobně u Kristiána Schrödera. Po vyučení se okolo roku 1696 vydal sbírat zkušenosti obvyklou vandrovní cestou. Prošel Rakousko a v Itálii dorazil až do Říma, kde měl možnost se seznámit s díly tesinského Francesca Borominiho, radikálního architekta považovaného římskými konzervativními současníky za „blázna“. V Itálii také přebral do svého jména otcovo jméno Santini, stejně tak učinil i bratr František.

V roce 1700 už Santini samostatně stavěl, projektoval a měl vlastní peníze, tzn. byl členem některého ze stavitelských cechů a vlastnil stavební firmu. V činnosti navazoval na architekta Jan Baptista Matheye, po jehož smrti některé projekty přebral a dokončil, přebral i Matheyův okruh stavebníků a částečně navázal i na jeho styl. Santinimu se dařilo a v roce 1705 koupil Valkounský dům (čp. 211) v Nerudově ulici za 3000 zlatých v hotovosti. O dva roky později koupil i sousední dům U zlaté číše (čp. 212) a oba domy spojil.

Po Schröderově smrti si Santini roku 1707 vzal za ženu jeho dceru Veroniku Alžbětu. S Veronikou měl čtyři děti, ale všichni tři synové – Jan Norbert Lukáš (* 1707), Josef Rudolf Felix Řehoř (* 1708) a František Ignác (* 1710) – zemřeli na souchotiny ještě jako děti a zbyla jen dcera Anna Veronika (* 1713).

Když v roce 1720 manželka Veronika umřela, oženil se s jihočeskou šlechtičnou Antonií Ignatií Chřepickou z Modliškovic a získal tak povýšení do šlechtického stavu. S ní měl v roce 1721 dceru Janu Ludmilu a v roce své smrti syna Jana Ignáce Rochuse. Kmotry všech Santiniho dětí se stali Santiniho mecenáši z řad vysoké šlechty.
Jan Blažej Santini-Aichel zemřel 7.prosince 1723 ve svých nedožitých 47 letech.

Příbuzenstvo:

Otec-Santiny Aichel

Bratr-František Santiny-Aichel

Děd-Antonín Aichel
Tchán-Kristián Shröder

Santiniho dílo

- kostel Nanebevzetí Panny Marie v Sedlci u Kutné Hory (obr.12)
- kaple Jména Panny Marie, Mladotice (1708–1710) (obr.13)
- dostavba sýpky, Klášter Plasy (obr.14)
- konvent, Klášter Plasy (obr.15)
- proboštství s poutním kostelem Zvěstování Panny Marie zvané Mariánská Týnice (obr.16)
- klášterní kostel Nanebevzetí P. Marie, Klášter Kladruby (obr.17)
- proboštství kláštera Údolí blahoslavené Panny Marie, Pohled (1714) (obr.18)
- poutní kostel sv. Jana Nepomuckého na Zelené hoře, Žďár nad Sázavou (obr.19)
- zámek Karlova Koruna, Chlumec nad Cidlinou (1721–1723) (obr.20)
- kostel sv. Libora, Jesenec (obr.21)
- kaple sv. Anny, Panenské Břežany (obr.22)
- kostel sv. Václava, Zvole (obr.23)
- poutní kostel Narození Panny Marie, Křtiny (obr.24)
- Thunovský palác (sídlo Poslanecké sněmovny Parlamentu ČR), Malá Strana, Praha (obr.25)
- přestavba kláštera v Želivi (obr.26)
- panská hospoda v Ostrově nad Oslavou na půdorysu ve tvaru W (obr.27)
- kostel na půdorysu želvy v Obyčtově (obr.28)
- kostel sv. Jiří v Pivíně (obr.29)
- Zámek Kalec (obr.30)

Mnoho podob Santiniho jména

Sám Santini, ale i doba a mnohonárodnostní evropské prostředí jsou zdrojem mnoha podob jeho jména. Setkat se můžeme s tvary a přepisy: Jan Blažej Santini, Jan Santini Aichl, Giovanni Santini, Jan Blažej Santini-Eichel, Giovanni Santini-Aichl, Johann Blasius Santini-Aichel, Giovanni Blasius Santini či Jan Blažej Santini-Aüchel.

Citát

Mojmír Horyna: „Santiniho Zelená Hora je báseň… Zatímco stavby dvacátého století jsou jenom slogany.“



Obr.11-Jan Blažej Santini-Aichel



Obr.12-kostel Nanebevzetí Panny Marie v Sedlci u Kutné Hory



Obr.13-kaple Jména Panny Marie, Mladotice



Obr.14- dostavba sýpky, Klášter Plasy



Obr.15- konvent, Klášter Plasy



Obr.16- proboštství s poutním kostelem Zvěstování Panny Marie zvané Mariánská Týnice



Obr.17- klášterní kostel Nanebevzetí Panny Marie, Klášter Kladruby



Obr.18- proboštství kláštera Údolí blahoslavené Panny Marie, Pohled



Obr.19-poutní kostel sv. Jana Nepomuckého na Zelené hoře, Žďár nad Sázavou



Obr.20- zámek Karlova Koruna, Chlumec nad Cidlinou (1721–1723)



Obr.21- kostel sv. Libora, Jesenec



Obr.22- kaple sv. Anny, Panenské Břežany



Obr.23- kostel sv. Václava, Zvole



Obr.24- poutní kostel Narození Panny Marie, Křtiny



Obr.25-Thunovský palác (sídlo Poslanecké sněmovny Parlamentu ČR), Praha



Obr.26- přestavba kláštera v Želivi



Obr.27- panská hospoda v Ostrově nad Oslavou půdorysu ve tvaru W



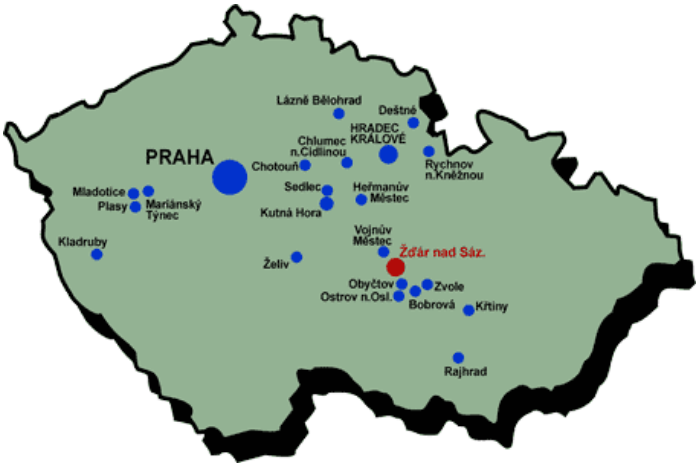
Obr.28- kostel na půdorysu želvy v Obyčtově



Obr.29-kostel sv.Jiří v Pivíně



Obr.30-Zámek Kalec



Mapa Santiniho díla

Barokní gotika

Barokní gotika nebo též gotizující baroko je jedinečný architektonický sloh, který spojuje dynamické baroko s výrazovými prostředky gotiky. Jejím tvůrcem a nejvýraznějším představitelem je český architekt Jan Blažej Santini-Aichel.

Po konci třicetileté války se české země pozvolna vzpamatovávaly z válečných útrap včetně starých klášterů. Základní obnovu klášterních panství následuje ke konci 17. století obnova a rozvoj samotných opatství. Některé kláštery při rekonstrukci svých chrámů žádají zachování kontinuity se středověkými stavbami, aby proti novým protireformačním řádům demonstrovaly vlastní starobylost. V jiných případech je nutné se starými stavbami počítat pro omezené zdroje. Pro architekta, který s pražskou gotickou Katedrálou sv. Víta vyrůstal (jeho otec i děd na výstavbě Pražského hradu pracovali), který gotiku nepovažuje za starý a překonaný módní trend, ale způsob konstrukce, a který je ze svých vandrovních cest okouzlen náznaky gotiky v barokních stavbách, jsou taková přání požehnáním.

V první třetině 18. století tak v Čechách vzniká řada historizujících staveb. Přesto nelze mluvit o pouhém kopírování nalezených původních vzorů – jsou rozšiřovány o prvky zahraniční pozdní gotiky, dále rozvíjeny a kombinovány s prvky odpovídající současnosti. Mezi nejkrásnější ukázky patří klášterní kostel v Želivi, přestavba kostela Nanebevzetí P. Marie v kladrubském klášteře, obnova chrámu Nanebevzetí Panny Marie v Sedlci u Kutné Hory a především novostavba kostela sv. Jana Nepomuckého na Zelené hoře u Žďáru nad Sázavou.

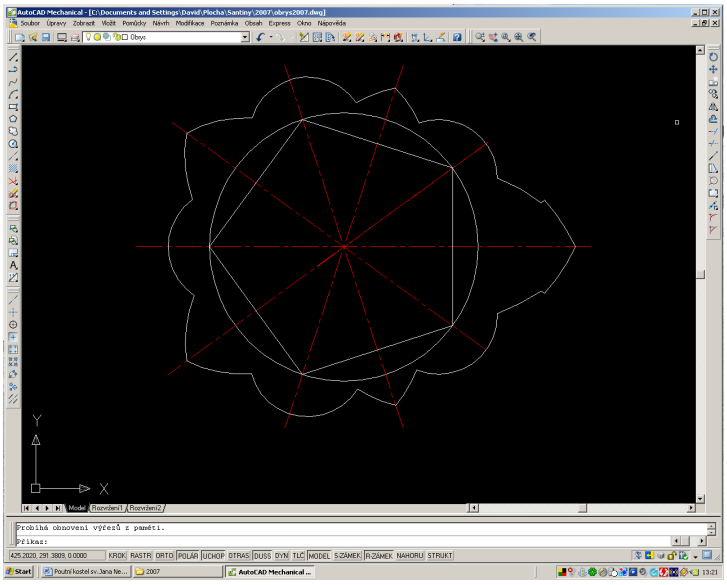
Vytvoření 3D modelu kostela

Dalším mým úkolem bylo vytvořit 3D model Santiniho poutního chrámu. K tomuto projetu jsem dostal od vedoucího práce kopie náčrtů Santiniho poutního kostela, kde ovšem nebyly žádné kóty, takže všechny potřebné rozměry jsem určoval pomocí měřítek uvedených na každém náčrtu (náčrty byly provedeny pro žádost na rekonstrukci památky). Přeměřování těchto parametrů trvalo poměrně dlouhou dobu, protože kostel má mnoho nepravidelných a složitých tvarů. Dále jsem obdržel program ArchiCAD 10, což je program na 3D modelování budov a různých stavebních pomůcek. Rozhodl jsem se, že nejlepším způsobem by mohlo být nejprve vynešení půdorysu a jednotlivých pater v programu AutoCAD 2007, protože tento strojařský program je přesnější a na rýsování 2D lepší než právě ArchiCAD. V AutoCADu jsem si narýsoval i různé křivky (střechy, okna, dveře, atd…), které jsou velmi významné pro tuto budovu. Velkou výhodou CAD programů je to, že se mezi sebou dají jednotlivé produkty různě překopírovávat a tak nebyl problém v přenesení půdorysu do programu ArchiCAD.
V tomto programu jsem nikdy předtím nepracoval, a proto bylo velmi důležité nejprve se řádně seznámit s jednotlivými funkcemi programu. K tomu mi sloužila knížka Názorný průvodce ArchiCADem. Zde jsem se dočetl různé pracovní rady, jak nejlépe využívat všechny funkce programu. Následně jsem se pomalu dostával do problému mého zadání. Pomocí funkce **spojit** jsem si vynesl půdorys do 3D prostředí a začal s vynášením zdí. To bylo asi to nejjednodušší na 3D modelování. Do tohoto problému jsem proniknul velmi rychle. Následně jsem modeloval dveře a okna. To už bylo složitější, protože otvory kostela jsou velmi specifické a tudíž nebyly v žádné z knihoven. Obrys otvorů jsem narýsoval ve 2D a následně převedl do 3D. Vytvořil jsem si knihovnu se speciálními otvory a tam všechny moje rysy převedené na 3D modely otvorů vkládal. Následně jsem musel hledat osy všech otvorů, aby okna a dveře byly umístěné na správných místech. Pořád jsem musel ověřovat, jestli jednotlivé rozměry sedí s celkovými měřítky. Když jsem otvory vnesl do celkového modelu, pro lepší design jsem za otvory vkládal další zdi, aby se zde vykreslily různé římsy, které udávají lepší celkový vzhled. Možná si říkáte, jak jsem vytvářel zdi, když na kostele jsou různé střechy a stříšky. Ovšem program ArchiCAD má velkou výhodu v tom, že zdi se dají ořezávat střechami. Křivky střech jsem si vytvářel v programu AutoCAD a následně převáděl zpět do 3D modelování. Zde se tato křivka pomocí několika nástrojů změnila v 3D model střechy. Takto jsem postupoval u všech střech a stříšek. To bylo asi to nejvíce složité na tomto modelu, protože na takto složité tvary není tento program přizpůsobený. Když se

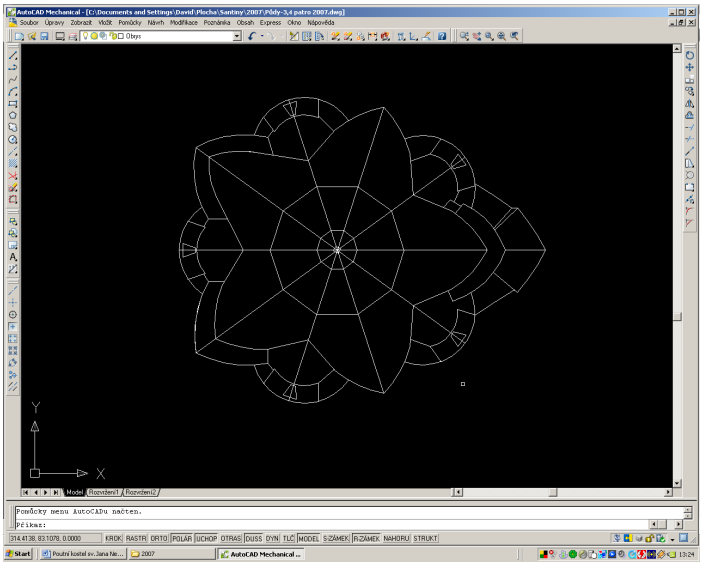
dokončily střechy, tak se zadal příkaz ořezat střechou a kostel postupně dostával konečný tvar. Následně se už pouze dodělávaly různé části potřebné pro zdokonalování hlavně designu.

Protože vynést 3D model kostela sv. Jana Nepomuckého nebylo zrovna jednoduché, domlouval jsem si různé konzultace na firmě PKS ve Žďáře nad Sázavou. Zde se mi věnovali Martin Kocmánek a Zdeněk Musil. Když se dostávalo na složitější věci, které v oboru stavitelství nejsou k vidění, věnoval mi pozornost hlavně Ing.Arch. Martin Zezula, bez jehož rad a drobných úprav bych asi těžko dosáhl takovéhoto výsledku, jako je vymodelování Santiniho poutního chrámu. Tyto konzultace byly pro mne velmi důležité, a proto bych chtěl vyzvednout práci těchto pánů, kteří jsou opravdovými specialisty ve svém oboru. Pro mne byla tato práce velkým přínosem, jelikož jsem se poměrně dobře naučil modelovat v programu ArchiCAD, poznal jsem nové modelovací prostředí a poznal také něco jiného, než jen kreslení ve strojařských programech. Nyní ovšem nevím, jestli se s tímto programem ještě někdy setkám, protože v budoucnu bych se chtěl dále zabývat spíše strojařskými programy jako AutoCAD, Autodesk Inventor, atd. (odpovídá mému oboru Technické lyceum, zaměření strojírenství). Přesto jsem velmi rád, že jsem se podrobně naučil pracovat v ArchiCADu. Celá práce mi zabrala odhadem desítky hodin volného času.

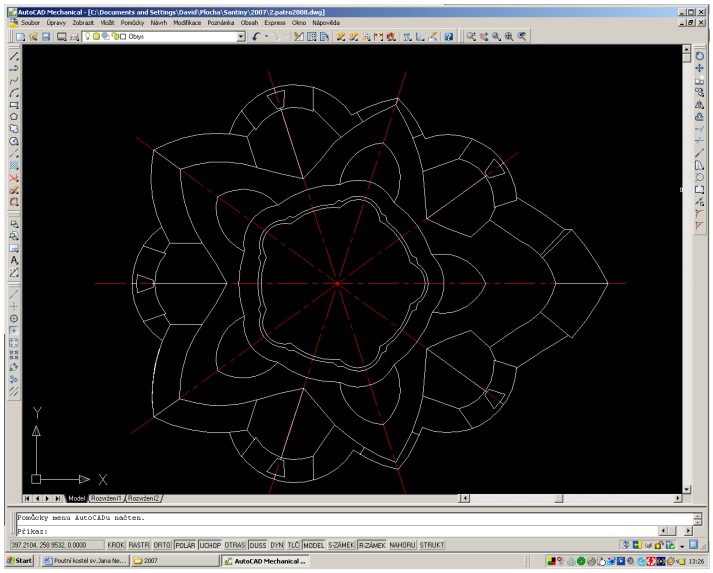
Pracovní výkresy



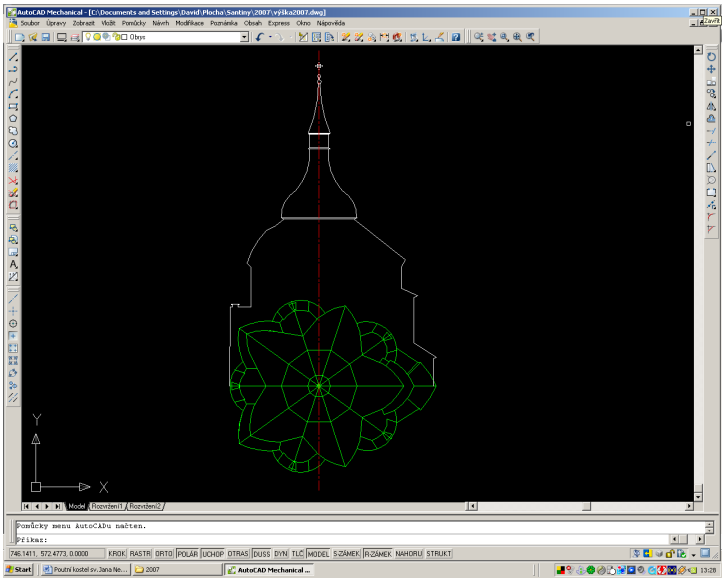
Půdorys kostelu, kde je vidět nejen obrys, ale i pětiúhelník, který je charakteristický pro tuto stavbu.



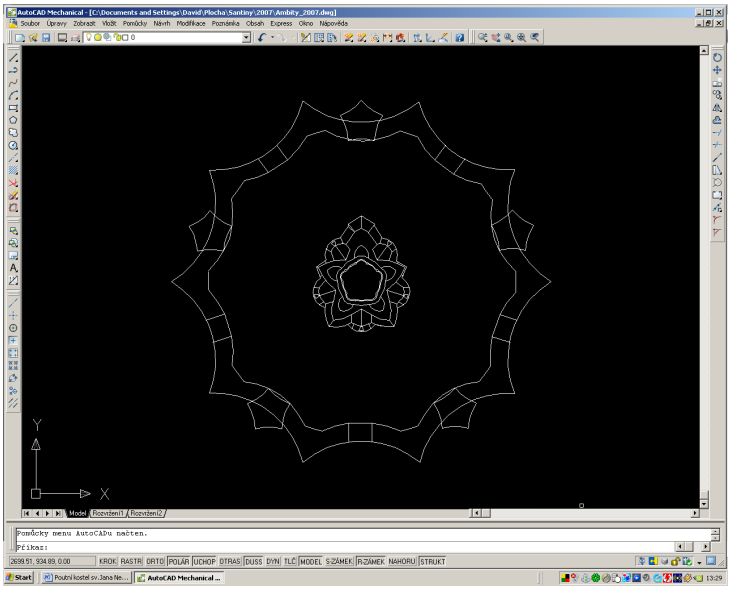
Výkres půdy, 3. a 4. patra. Tento výkres je důležitý pro tvorbu střech.



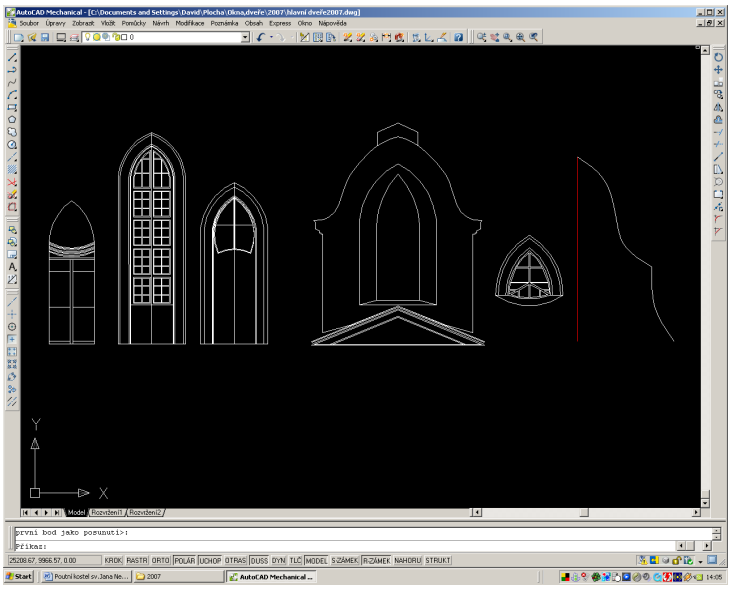
2.patro kostela. Tento výkres byl nejdůležitější pro celou tvorbu kostelu, protože je zde vidět nejen obrys, ale jsou zde vidět i místa, kde na sebe navazují jednotlivé střechy.



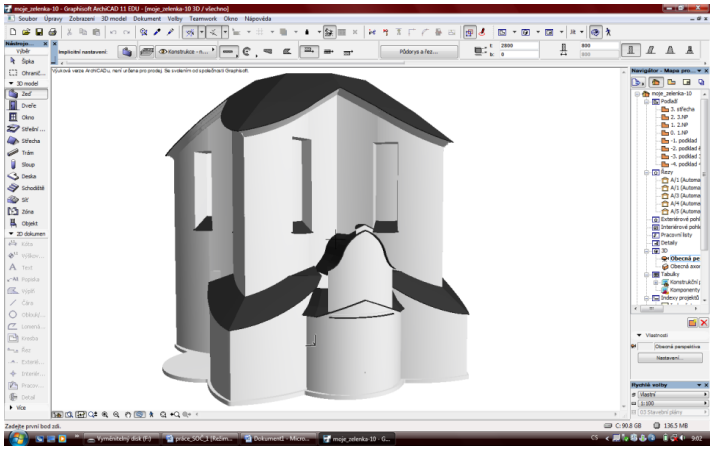
Tento výkres je pro samotný 3D model také velmi důležitý, protože je v něm určen poměr půdorys : výška.



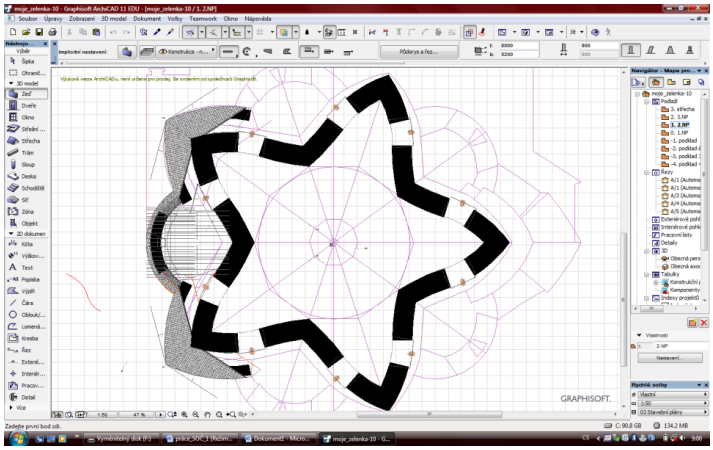
Zde je kromě půdorysu vykreslen ještě prstenec ambitů, který obklopuje poutní kostel.



Kreslení dveří a oken, která jsem později vkládal do knihoven 3D modeláře.



Rozpracovaný virtuální model 3D kostela.



Půdorys v ArchiCADu.

Závěr a diskuse výsledků

Návrh uplatnění vzniklého produktu ve škole

V této části bych chtěl rozebrat, jak může být vzniklý produkt užitečný ve škole. Myslím si, že něco tak skvostného, jako je Santiniho poutní chrám by neměl být jen tak zapomínán a studentům by měl být neustále připomínán jako jedna z nejvýznamnějších dominant nejen Žďáru nad Sázavou, ale i celé České republiky.

3D model kostela by mohl být užitečný například ve výuce zeměpisu. Ať už se probírá kraj Vysočina nebo přímo Žďár nad Sázavou, žádnému učiteli se nechce se studenty chodit na Zelenou horu, aby si tuto památku prohlédli pěkně zblízka a nevnímají, že někteří studenti Poutní kostel ani pořádně neviděli. Takhle si mohou alespoň zblízka prohlédnout všechny typické rysy kostela. Některé tyto charakteristické rysy ani vidět ze země nemůžeme a málo komu z nás se poštěstí vidět toto Santiniho dílo například z letadla. Takhle si studenti prohlédnou model a už ví, jak to vlastně všechno vypadá.

Dále by se dal 3D model využít například ve výuce CAD systémů, aby student viděl v reálném pohledu, co všechno lze v těchto moderních programech vytvořit. Studenti by alespoň dostali inspiraci do další práce a viděli by i něco složitějšího, než se vyučuje v normálních hodinách a co mají studenti šanci si v hodinách vyzkoušet. Třeba by to někoho začalo zajímat a začal by se více zaujímat touto problematikou. Také na modelu může vyučující demonstrovat některé konstrukční postupy, např. vysunování, tvoření otvorů a ořezávání plochami.

Dobrou učební pomůckou by Santiniho poutní chrám mohl mít při vyučování architektury v oboru stavebnictví, protože zde se také Santiniho dílo probírá a studenti by měli mít větší přehled o jeho tvorbě. V tomto oboru by studenty měl zaujmout hlavně půdorys, ve kterém je poutní kostel postaven a pak různé detaily, které se v architektuře probírají.

Dále by mohl 3D model sloužit k výuce dějepisu. Zde se rozebírají jednotlivá období a umělecké směry. Když se zastavíme u období tzv. Barokní gotiky, co jiného by mělo tuto kapitolu tvořit, než Jan Blažej Santini a jeho dílo.

Když jsou na naši škole Dny otevřených dveří, tak se ukazují různé produkty, co studenti ve škole vytvářejí. Budoucí studenti mohou uvidět, že studenti na naší škole jsou schopni vytvářet takovéto výrobky a musí jim být jasné, že tato škola je opravdu dobrá a je schopna připravit studenty na budoucí povolání v technických oborech. Musí být všem zřejmé, že někdo, kdo by neuměl konstruovat v moderních softwarech, asi by těžko něco takového dokázal vytvořit.

Myslím si, že učitelé by 3D model Santiniho poutního chrámu měli ve výuce začít ve ve větší míře využívat, protože jak jste se mohli dozvědět v předchozím textu, předmětů, kde by se 3D model dal využít, je opravdu dost a dost. Studenti by měli vidět dominantu našeho kraje a města nejen například při cestě autobusem do školy, ale i přímo ve školních předmětech.

Vytvořený model bude vystaven na školních web stránkách www.spszr.cz a o projektu bude napsán článek do odborné literatury a umístěn na technický web www.designtech.cz.

Model bude nejen skutečně virtuální, ale vygenerovaná data ve formátu STL budou předána k 3D tisku na školní tiskárnu. Tento 3D tisk z plastu ABS bude povrchově upraven a vystaven ve škole.

Použití v komerční oblasti

Model kostela může sloužit k mnoha účelům. Jak už jsem se zmiňoval v této práci, může být využíván například k vyučování některých školních předmětů. Mimo to by mohl mít velké uplatnění i v komerční oblasti.

Mohl by být využíván jako model k propagaci města a cestovního ruchu v naší Chráněné krajinné oblasti Žďárské vrchy (CHKO). Z tohoto důvodu bude model (virtuální i skutečný z 3D tisku) i předán panu starostovi k další propagaci.

Když se zamyslíme nad dalším využitím 3D modelu kostela, tak každého musí napadnout, že by bylo dobré používat tento model na různých výstavách, protože ho zde uvidí asi nejvíce lidí ať už z blízkého nebo dalekého okolí.

Dále by se dal 3D model velmi dobře využít k výuce budoucích stavařů a architektů, pro které by měl být model velmi dobrou inspirací pro budoucí práci. Nad čím jiným by měli žasnout údivem než nad skvostným dílem Santiniho kostela? Myslím si, že ve zdejším okolí nic lepšího z architektonického hlediska nenajdete.

Pro naši školu by mohl mít model také velmi významnou cenu. Mohl by sloužit k velmi dobré propagaci naší školy a k prezentaci dovedností našich studentů.

Tento model by mohl být nafocen do různých katalogů, kde by mohla proběhnout porovnání- model : realita.

Model by mohl vést k obohacení některých internetových stránek, ať už naší školy nebo i stránek města Žďár nad Sázavou.

Seznam použité literatury

[1] Kolektiv autorů. Zelená hora u Žďáru nad Sázavou (Příspěvky k dějinám a obnově poutního místa), Státní ústav památkové péče, 1997

[2] BARTH, Fritz. Ein Baumeister des Barock in Böhmen, Santini, Hatje Cantz Verlag GmbH+C, 2004, ISBN 3-7757-1468-5

[3] HORYNA, Mojmír. J. B. Santini-Aichel – Život a dílo. Karolinum, Praha, 1998, ISBN 80-7184-664-3

[4] HORYNA, Mojmír. Santini - Stavby J.B.Santiniho na Žďársku, rezident, 1999

[5] KOTRBA, Viktor. Česká barokní gotika – dílo Jana Santiniho-Aichla, Academia, Praha, 1976

[6] SEDLÁK, Jan. Jan Blažej Santini – setkání baroku s gotikou, Vyšehrad, Praha,1987

[7] SEDLÁK, Jan. Jan Santini Aichl – Dědictví (Kostel sv.Jana Nepomuckého ve Žďáře nad Sázavou), Amos, Brno, 1995

[8] SEDLÁK, Jan. Žďár nad Sázavou (Areál státního zámku), Čelákovice, 1979

[9] Novinové články: Životopis v KDO BYL KDO v našich dějinách do roku 1918,

Po Santiniho stopách na Žďársku (Novinky.cz)

Internetové zdroje:

www.wikipedia.org

www.hrady.cz

www.zamekzdar.cz

Středoškolská technika 2008

Setkání a prezentace prací
středoškolských studentů na ČVUT

Rangueho vírová trubice

Ondřej Řezníček

Vyšší odborná škola a Střední průmyslová škola
Studentská ulice 1, Žďár nad Sázavou



.

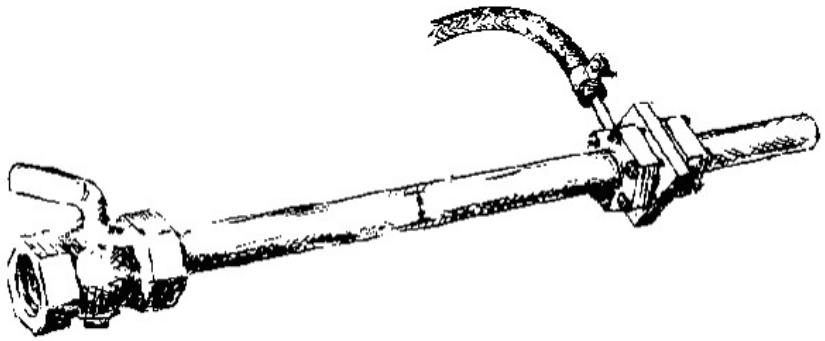
Prohlašuji tímto, že jsem soutěžní práci vypracoval samostatně pod vedením **Dr. Ing. Josef Příhody** a dalších konzultantů uvedených v práci a uvedl v seznamu literatury veškerou použitou literaturu a další informační zdroje včetně Internetu.

Na úvod bych chtěl poděkovat všem, kteří mi s prací pomáhali. Jsou to:
Dr. Ing. Josef Příhoda, ředitel VOŠ a SPŠ Žďár nad Sázavou, metodické vedení práce
Ing. Karel Bittner, Humusoft s. r. o. Praha, simulace proudění a přenosu tepla
Ing. Petr Fořt, učitel CAD, VOŠ a SPŠ Žďár nad Sázavou, CAD konzultace
Josef Štěpina, dílenský učitel, VOŠ a SPŠ Žďár nad Sázavou, experimentální ověřování

1. Úvod

Za téma své dlouhodobé maturitní práce jsem si vybral problematiku vírové trubice, která řeší specifické průmyslové problémy zvláště při chlazení za použití stlačeného vzduchu jako jediného zdroje energie. Vírová trubice je **velmi jednoduché zařízení**, které nemá žádné pohyblivé součástky a **nepotřebuje žádnou údržbu**. Proud stlačeného vzduchu se v trubici rozděluje na teplou a studenou složku. **Není známo žádné jednoduché ani složité fyzikální vysvětlení tohoto jevu** a proto mne toto zařízení velmi zaujalo. Nelze ale zpochybnit, že **trubice je funkční** a v praxi **se používá k chlazení** ve vhodných aplikacích.

Vírová trubice pracuje na principu vírového efektu, který v roce 1928 jako první zpozoroval a popsal francouzský fyzik **Joseph Georges Ranque**. Všiml si totiž, že při odčerpávání vzduchu z vývěvy na jednom výstupu dochází k výfuku proudu vzduchu o podstatně menší a na druhém o podstatně větší teplotě, než byla teplota vzduchu uvnitř vývěvy před začátkem odčerpávání.



Obr. 1 Pohled na vírovou trubici

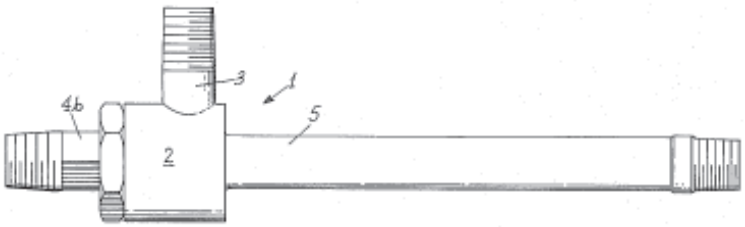
Začal se tedy tímto jevem zabývat a po dvou letech práce předvedl svým kolegům první vírovou trubici, která však trpěla několika nedostatky a to zejména velmi malým chladícím efektem a velkými náklady na provoz. I přesto založil firmu a začal s výrobou vírových trubic, ale kvůli nedostatku zakázek jeho podnik zkrachoval. Svoji celoživotní práci shrnul ve svém díle Experiences sur la detente avec production simultanees dun echappement dair chaud et dun echappement dair froid.

Poté se na vírovou trubici zapomnělo, dokud se jí a zvláště vírovým efektem ve čtyřicátých letech v USA nezačal zabývat fyzik **Rudolf Hilsch**, který své poznatky shrnul ve svém díle The use of the expansion of gases in a centrifugal field as cooling process. Zde se snažil objasnit vznik studeného proudu vzduchu, to se mu ovšem nepovedlo, ale svojí prací se mu podařilo zvýšit zájem odborníků o tento problém. Od té doby se tímto problémem zabývalo mnoho vědeckých týmu z celého světa, nikomu se však dodnes tento problém nepodařilo uspokojivě vysvětlit.

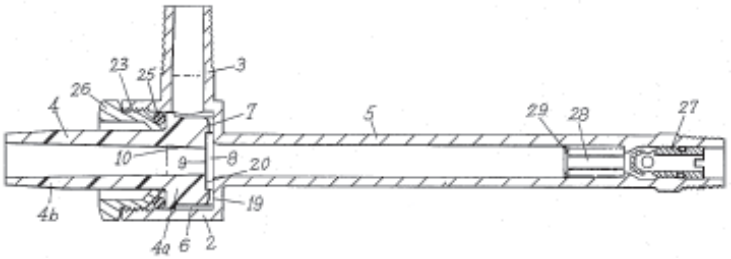
V roce 1961 vznikla v městečku Cincinnati (Ohio,USA) firma **Vortec**, která jako první úspěšně zavedla vírovou trubici do praxe. Nyní je firma Vortec vlastníkem mnoha patentů na aplikace vírových trubic a také největším dodavatelem vírových trubic a jejich doplňků.

2. Vírová trubice

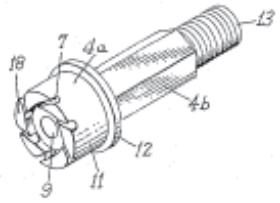
V této kapitole bude popsán pohyb vzduchu uvnitř trubice a zároveň základní konstrukční prvky vírové trubice, která se svým vnitřním uspořádáním velice podobá vírové trubici, na které jsem provedl ověření funkčnosti vyráběných vírových trubic. Autorem této trubice je Američan Charles D. Fulton, který si ji patentoval v roce 1965.



Obr. 2 Boční pohled na vírovou trubici (A)



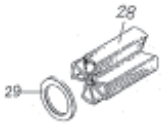
Obr. 3. Řez vírovou trubicí (B)



Obr. 4 Detail generátoru a studeného výfuku vírové trubice (C)



Obr. 5 Přední pohled na generátor vírové trubice (D)



Obr. 6 Detail řídícího ventilu (E)

Hlavními součástmi vírové trubice (1) je tělo (2), přívodní ventil (3), generátor (4a), studený výfuk (4b) a teplý výfuk (5). Filtrovaný vzduch stlačený na hodnotu 0,6MPa se přivádí pomocí přívodního ventilu (3) do kruhové komory (6), která se nachází mezi stěnou těla vírové trubice (2) a generátorem (4a), zde vzduch začne rotovat, odtud je vzduch pod stále stejným přívodním tlakem rovnoměrně vstřikován do trysek (7), které jsou umístěny po obvodu generátoru (4a) a jsou orientovány tečně k vnitřnímu komoře generátoru (8), tento detail je zobrazen na obrázku D. Vzduch tedy vstupuje do trysek, kde expanduje a získává velmi vysokou rychlost a vstupuje do vnitřní komory generátoru (8), kde vzduch díky odstředivým silám začne rotovat po vnitřní straně komory generátoru (8).

Rotující proud vzduchu opouští komoru generátoru (8) a postupuje teplým výfukem (5), tato volba je důsledkem toho, že otvor ve vnitřní komoře generátoru (8) pro teplý výfuk (5) je větší než otvor do studeného výfuku (9). Rotující vzduch tedy postupuje dále po stěnách teplého výfuku (5) až narazí na řídící ventil (27, 28, 29), který je umístěn na jejím konci. Zde nastavení řídícího ventilu (27, 28, 29) určí množství vzduchu, které odejde teplým koncem vírové trubice. Hodnota tlaku tohoto vzduchu je menší než na přívodu, ale je stále větší než okolní atmosférický tlak. Teplota tohoto vzduchu je větší než na vstupu, ale teplotní rozdíl je velmi malý, proto se v praxi moc teplá frakce vírové trubice nepoužívá. Dalším důvodem je také to, že existuje mnoho výkonnějších a také úspornějších prostředků k průmyslovému ohřevu.

Zbylý vzduch je řídícím ventilem (27, 28, 29) tlačen do středu vírové trubice, kde se vzduch rotačním pohybem začne středem trubice pohybovat teplým výfukem zpět do vnitřní komory generátoru, kde prochází otvorem pro studený výfuk (9) do studeného výfuku (4b), kde vírovou trubici opouští. Studený proud vzduchu, který trubici opouští, má už hodnotu tlaku atmosférického a teplota tohoto proudu odpovídá nastavení množství výfuku teplého vzduchu na řídícím ventilu (27, 28, 29).

Příčinami existence dvou oddělených rotujících proudů, jeden uvnitř druhého, je to, že díky odstředivým silám působících na vnější proud vzduchu se vytvoří ideální cesta pro vnitřní proud vzduchu, a také rozdíl tlaků mezi řídícím ventilem a studeným výfukem. Důležité také je, že oba dva proudy vzduchu rotují ve stejném směru a také se stejnou úhlovou rychlostí.

V následující tabulce jsou uvedeny **parametry vyráběných modelů** vírových trubic firmou Vortec, která vyvinula několik standardních velikostí vírových trubic pro zajištění různého poměru množství vzduchu a chladící nebo ohřívací schopnosti.

Model	Spotřeba stlač. vzdu. při 6,9bar (l/min)	Pokles teploty (°C)	Kapacita tepla (kcal/hod.)	Spotřeba stlač. vzdu. při 5,5bar (l/min)	Pokles teploty (°C)	Kapacita tepla (kcal/hod.)
106-2-H	57	34	25	47	36	21
106-4-H	113	44	64	93	41	48
106-8-H	227	45	101	188	42	82
208/208SS-11-H	312	47	161	258	45	126
208/208SS-15-H	425	47	227	351	46	192
208/208SS-25-H	708	37	378	585	37	285
308-35-H	992	42	668	820	40	542
328-50-H	1416	44	756	1170	36	544
328-75-H	2125	47	1134	1755	44	950
328-100-H	2833	43	1512	2340	41	1247
610 CAG	425/708	55,6	227/378			
608 Mini CAG	227					
424 Thread Guard	113	-12 abs.				
750*/760	227		101			
740*/730	425		225			
790*/780	708		378			
770**	708		378			
795*/785	991		625			
7970*/7980	1981		1250			
797*/787	708		378			
797SS*/787SS	708		378			
796-6 VorCool	708		378			
22815	170-425	+/- 33°C				
22825	453-708	+/- 33°C				
22835	538-962	+/- 33°C				
220	170-340	+/- 33°C				

Množství tepla je při teplotě stlačeného vzduchu 21°C a rosném bodu –40°C.
* Model obsahuje termostat
** Model obsahuje jednoduché řízení teploty bez seřiditelného termostatu

3. Teplotní rozdělení

Vzduch v teplé trubici je v neustálém pohybu. Vnější proud vzduchu se pohybuje k teplému konci vírové trubice a vnitřní proud vzduchu se pohybuje ke konci studenému. Oba proudy vzduchu rotují ve stejném směru a také se stejnou úhlovou rychlostí. To je důsledkem intenzivní turbulencí na hranici mezi proudy, která spojuje oba proudy do jedné masy tak dalece, jak se dotýkají jejich rotační pohyby.

Vnitřní proud vzduchu je případem tzv. nuceného víru, který odlišujeme od víru volného, ve kterém rotační pohyb nezávisí na nějakém vnějším vlivu odlišným od zachování momentu hybnosti. V našem případě způsobuje vnější teplý proud vzduchu rotaci vnitřního studeného vzduchu konstantní úhlovou frekvencí.

Pro lepší vysvětlení pojmů nuceného víru a volného víru, si můžeme představit příklad. Vír vznikající při vypouštění vody z vany je příkladem volného víru. Jak se voda pohybuje dovnitř, její rychlost rotace vzrůstá, aby se zachoval moment hybnosti. Lineární rychlost každé částice ve víru je nepřímo úměrná jejímu poloměru. Tak se ve volném víru při pohybu částice z poloměru jedné jednotky na poloměr poloviční lineární rychlost částice zdvojnásobuje. V nuceném víru s konstantní úhlovou rychlostí se lineární rychlost zmenší při pohybu částice z poloměru jedné jednotky na polovinu jednotky na polovinu.

Z toho plyne, že u volném víru částice vstupuje do dráhy s rychlostí čtyřikrát větší než u víru nuceného. Kinetická energie je úměrná čtverci lineární rychlosti, takže částice opouštějící dráhu v nuceném víru má 1/16 hodnoty kinetické energie částice ve volném víru. Nyní přichází na řadu tajemství vírové trubice, které spočívá v tom, kde se zbylých 15/16 celkové kinetické energie ztrácí.

Energie odcházející z vnitřního proudu vzduchu ve formě tepla je předávána vnějšímu proudu. Nyní je velmi důležité si uvědomit to, že množství vzduchu ve vnějším proudu je vždy větší než v proudu vnitřním, protože část teplého vzduchu odchází ventilem na teplém konci vírové trubice. Jestliže je tedy potom množství tepla předaného studeným proudem stejné jako množství tepla získaného vnějším proudem, musí být pokles teploty vnitřního proudu větší než teplotní zisk vnějšího proudu, protože je jeho množství menší.

4. Rovnice teplotní rovnováhy

Rovnice teplotní rovnováhy je základní rovnicí vírové trubice, vychází ze základního poznatku, že energie odevzdaná studeným vzduchem ve vírové trubici se objeví ve vzduchu teplém.

Rovnice má tvar:

$$CF \cdot (T_i - T_c - JT) = (100 - CF) \cdot (T_h - T_i + JT)$$

$CF[\%]$ = studená frakce

$T_i[°F]$ = vstupní teplota

$T_c[°F]$ = teplota studeného vzduchu

$T_h[°F]$ = teplota teplého vzduchu

$JT = 4°F$ (při vstupním tlaku 100 PSIG), Joule- Thomasova teplotní korekce

(PSIG „Pound per Square Inch Gauge"- PSI relativního tlaku vůči okolnímu atmosférického tlaku,1 psi = 6 894,757 Pa)

Pro výpočet studené frakce lze tuto rovnici upravit do tvaru:

$$CF = \frac{T_h - T_i + 4}{T_h - T_c} \cdot 100$$

která je pro její výpočet vhodnější.

U vírových trubic se také snažíme vypočítat výkon chlazení a ohřívání v BTUH, které vypočteme pomocí těchto vzorců:

pro chlazení:

$$BTUH = 1{,}0746 \cdot CF_{Mc} \cdot (T_i - T_c)$$

pro ohřívání:

$$BTUH = 1{,}0746 \cdot CF_{Mh} \cdot (T_h - T_i)$$

CF_{Mt} = celkový průtok

CF_{Mc} = průtok studeného vzduchu $CF_{Mc} = CF_{Mt} \cdot CF$

CF_{Mh} = průtok horkého vzduchu $CF_{Mh} = CF_{Mt} \cdot (100 - CF)$

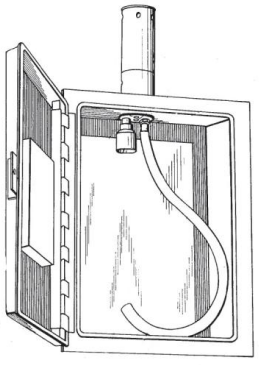
5. Využití vírové trubice

Vírová trubice je velmi jednoduché zařízení, proto se její konstrukce od doby svého vzniku moc nezměnila. Podařilo se však zvětšit její chladící efekt, čímž se podařilo odstranit hlavní příčinu jejího počátečního neúspěchu. Ovšem v porovnání s ostatními prostředky průmyslového chlazení je její výkon stále malý. To vede k jejímu nasazení k **řešení specifických průmyslových problémů, zvláště při chlazení**, kde nelze z různých důvodů využít jiné výkonnější přístroje.

Z konstrukčního řešení vírové trubice totiž vyplývá několik výhod a to především její vysoká spolehlivost, která je důsledkem toho, že neobsahuje **žádné pohyblivé součásti**, a také její **okamžitý náběh**. Dále můžeme zmínit její **nízkou pořizovací cenu**, **ekologický provoz**, **snadnou ovladatelnost**, práci **bez rizika vzniku jisker** a její **nízkou hmotnost**. V praxi se kvůli těmto výhodám vírové trubice využívá **více než 1000 aplikací.**

V následujícím seznamu je uvedeno **několik příkladu využití vírové trubice v praxi:**

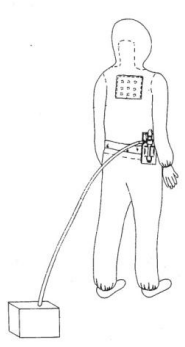
- Nejrozšířenější aplikací vírové trubice v praxi je její využití při **chlazení skříní** elektronického řízení automatických výrobních linek, což vede k omezení chybných elektronických signálů, které mohou mít za následek přerušení celého výrobního procesu. Hlavní výhodou této aplikace vírové trubice je především snížení nákladu na údržbu. Na následujícím obrázku je tato aplikace zobrazena.



- Druhou nejrozšířenější aplikací vírové trubice je její nasazení při **chlazení nástrojů** při obrábění či tváření (soustružení, vrtání, frézování, broušení, vysekávání, děrování, řezání atd.)

plastů, gumy, dřeva atd. U obrábění kovů se využívá pouze ve speciálních případech, protože nedosahuje dobrých výsledků jako klasické chlazení olejem a jinými řeznými kapalinami.

- Dalším příkladem využití vírové trubice je **chlazení ochranných oděvů** a helem pracovníků pracujících v prostředí, kde jsou vystaveni velkým teplotám. Využívá se především její malé hmotnosti a malých rozměrů, takže pracovníka při práci nijak nezatěžuje. Pracovníci jsou ovšem omezeni délkou přívodu stlačeného vzduchu, což omezuje jejich dosah.



- Jedním z příkladů využití teplé složky vzduchu vírové trubice je její aplikace při **sušení inkoustů** při tisku na speciální etikety. Hlavní výhodou aplikace vírové trubice oproti elektronickým ohřívačům při tisku na etiketu je to, že se elektronické ohřívače velmi těžko regulují, což má za následek zvrásnění materiálu etikety, zatímco vírová trubice vytváří mírný proud teplého vzduchu pro sušení inkoustů kontrolovaným způsobem, tedy bez nežádoucích deformací materiálu.
- Další možností využití vírové trubice je její aplikace na **ohřívání materiálů před lepením** epoxidovými lepidly. Epoxidová lepidla nebo také povlaky se totiž při kontaktu s chladnou plochou předčasně vytvrzují, což vede k nežádoucímu snížení účinnosti lepení.
- Vírových trubic se také využívá k **udržení teploty par benzínu** při testování. To probíhá tak, že se benzín ohřívá, aby došlo k uvolnění par a udržení konstantní teploty, aby se zabránilo zkondenzování par benzínu. Teplý konec vírové trubice je proto nasměrován na potrubí, kterým proudí páry benzínu. Hlavní výhodou vírové trubice je to, že pracuje bez rizika vzniku jisker a tím tedy také bez rizika vznícení par benzínu.

- Vírová trubice se také uplatnila ve firmě na výrobu vyfukovaných plastových nádob. Docházelo totiž k problémům s lepením nálepek na nádobu před ochlazením nádoby. Nádoba se totiž při ochlazování smršťuje a tím dochází k svraštění nálepky. Problém byl vyřešen využitím vírové trubice, která foukala studený vzduch do nádoby a umožnila její **rychlé ochlazení ještě před lepením nálepek**.

Další příklady využití vírové trubice jsou dále uvedeny v tomto seznamu výčtem:

- chlazení **nástrojů** při obrábění kovů, plastů, gumy a dřeva
- chlazení **nožů** při řezání papíru a látek
- chlazení **jehel** u průmyslových šicích strojů
- vysoušení **vzorků plynů** pro laboratorní experimenty
- **zkoušení** termostatů, přístrojů a elektroniky
- **testování** teplotních senzorů, kalibrace jističů
- chlazení **forem** při vstřikování plastů, ve sklárnách
- chlazení **při svařování** fólií, broušení čoček
- chlazení svařovacích **hořáků** (náhrada chlazení vodou)
- chlazení **při navařování** kovů plazmou
- chlazení horkého roztaveného **lepidla** při vytváření knižní vazby, skládání kartónů, a výrobě nálepek
- chlazení **nástrojů** při vysekávání/prostřihování žvýkací gumy
- chlazení roztaveného **lepidla** na nálepkách zabraňuje slepení a zvyšuje produktivitu výroby cigaret
- chlazení roztaveného rubu **koberce**
- chlazení **u sítotisku** v textilním průmyslu
- chlazení **taveniny** při povlakování papíru zabraňuje vrásnění papíru vlivem tepla
- chlazení při vkládání **fólie** v nábytkářském průmyslu
- chlazení **při tvarování teplem** v nábytkářském průmyslu
- chlazení **zápustky** při vysekávání těsnění z kartónu nebo fólie
- chlazení **ostří** při rozřezávání obálek
- chlazení **pily** na řezání papírových krabic
- chlazení lepicí vrstvy kovových **nálepek** na tuby pro rtěnky
- chlazení roztaveného **švu ve formě** - zabrání nalepení na řezací nůž

- chlazení a sušení **pásku** při ořezávání knih
- vysoušení a chlazení **vzorku** chlóru - chlazený vzduch proudí spirálou výměníku tepla
- chlazení **mýdla** v tunelu před lisováním - zabraňuje nalepení na lisovadla
- ohřívání - roztažení **tlačítka** pro aerosolové nádobky pro snazší instalaci
- chlazení **hadice** před operací vyztužování - zvýší produktivitu práce, kvalitu výroby, nahradí chlazení dusíkem, které způsobuje zkřehnutí hadice

6. Experimentální ověření

Škola vlastní několik druhů Rangueho trubice. Dvě trubice byly zakoupeny (cena asi 12 000 Kč), další čtyři trubice byly ve škole vyrobeny. Pro praktické ověření funkčnosti trubice jsem zvolil jednu ze zakoupených trubic. Trubici jsem uchytil do ručního svěráku podle následující fotografie:



Obr. 7 Uchycení trubice

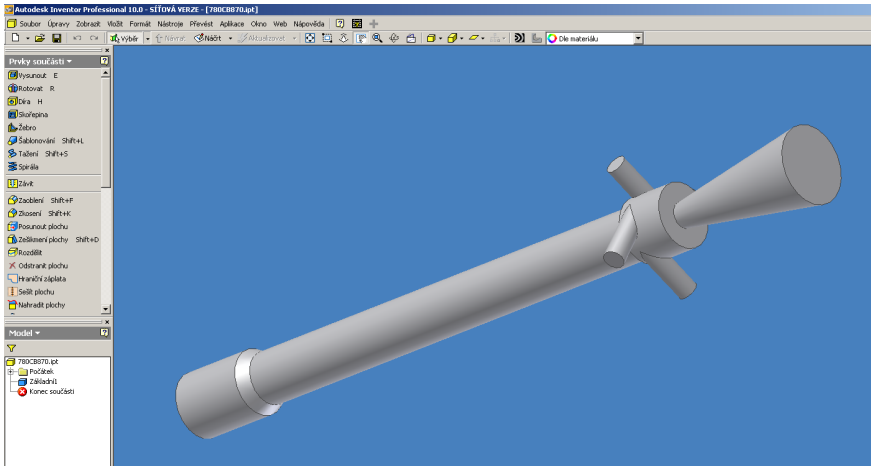
Stlačený vzduch potřebný k činnosti trubice byl veden hadicí od dvouválcového dílenského kompresoru. Tlak tohoto kompresoru nebyl požadovaných 0,6 MPa (6 atmosfér), ale mnohem nižší (**0,2 MPa**). Proto také dosažená minimální teplota na studeném konci trubice byla **9 °C**. Teplota byla změřena digitálním teploměrem (termočlánkem s digitálním výstupem). Vstupní teplota stlačeného vzduchu byla asi **22 °C**. I takto zjištěný pokles teploty svědčí o správné činnosti trubice. Předchozí měření v kompresorovně a. s. ŽĎAS Žďár nad Sázavou bylo založeno na skutečném požadovaném vstupním tlaku **0,6 MPa** (tvrdý zdroj) a

vstupní teplotě **20 °C**. Při tomto měření se dosáhlo snížení teploty až na **- 30°C**. Vyšší hladina hluku („hvízdání“ trubice) se může potlačit dodávaným tlumičem hluku.

7. Simulace činnosti trubice

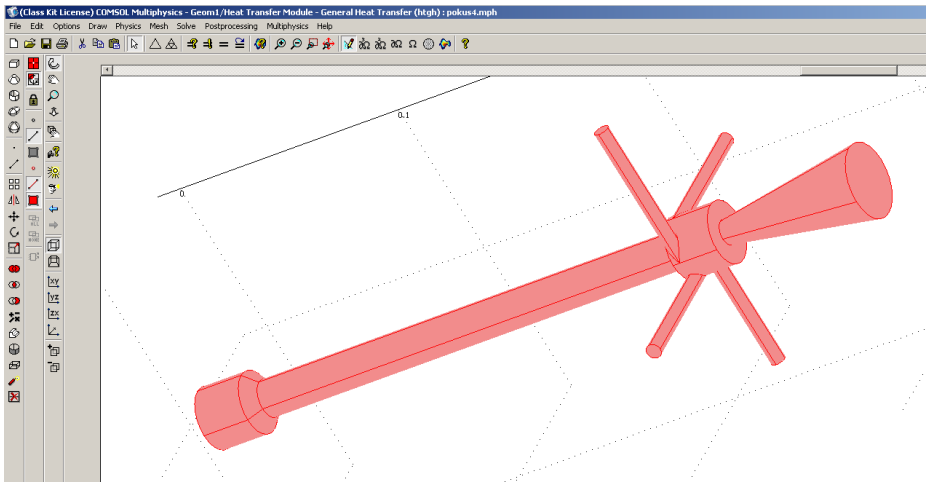
Na škole je vyučován program **MKP** (MKP - Metoda konečných prvků, anglicky FEM – Finite Element Method) švédské firmy COMSOL. Program se nazývá **COMSOL Multiphysics 3.2** a umožňuje výpočty MKP pro mnoho oblastí fyziky. V trubici nastává turbulentní proudění a zcela jistě i přenos tepla prouděním a vedením.

Z uvedených důvodů jsem program COMSOL zkusil aplikovat i na Rangueho trubici. Vytvořil jsem 3D virtuální model vnitřku trubice (vzduchu) v programu **Inventor**, viz obrázek.



Obr. 8 Model vnitřku trubice v Inventoru

Tento model jsem potřeboval importovat do grafického prostředí programu COMSOL, ale nastaly komplikace s tímto přenosem. Proto jsem grafická data vytvořil raději přímo v programu COMSOL, viz následující obrázek.

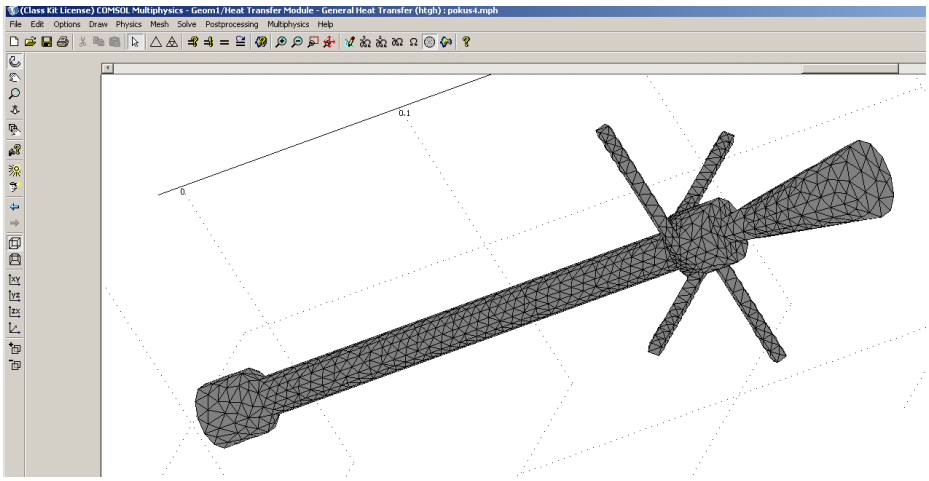


Obr. 9 Model vnitřku trubice v COMSOLu

Tato geometrická data se v programu použijí pro řešení které má vždy následující postup:

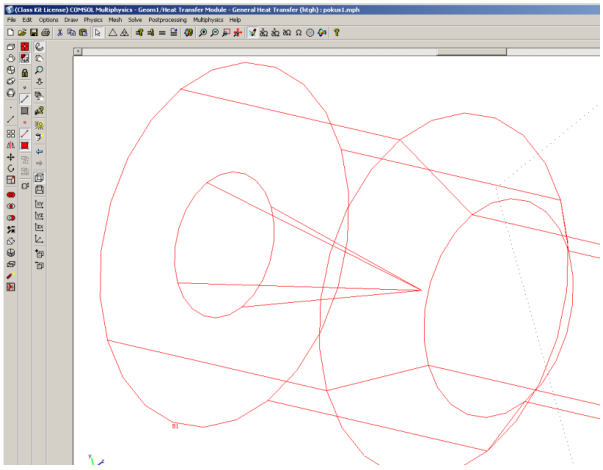
- **vytvoření geometrie úlohy**
- **vygenerování sítě konečných elementů**
- **zadání vlastností domény (vlastnosti materiálu)**
- **zadání okrajových podmínek (např. tlak a teplota vzduchu na vstupu…)**
- **zvolení řešiče**
- **vlastní výpočet**
- **prohlížení a diskuse výsledků (postprocessing)**

Podle uvedeného postupu jsme nechal vygenerovat síť konečných elementů, výsledek je na dalším obrázku.



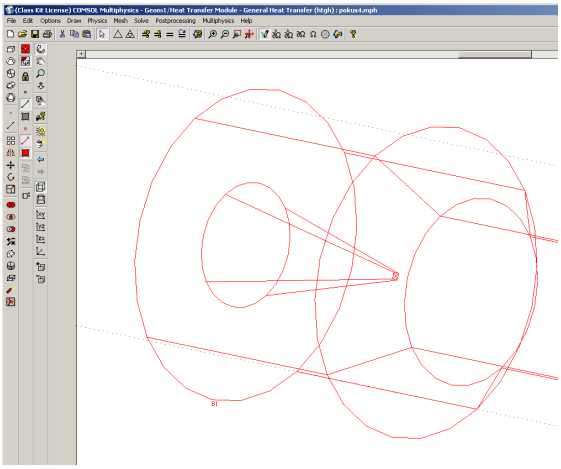
Obr. 10 Síť modelu

Protože škola nemá zakoupené moduly pro turbulentní proudění a 3D úloha je velmi náročná na výpočty, předal jsem **geometrická data k výpočtu do firmy Humusoft Praha**. Z firmy mi bylo sděleno, že v oblasti škrtícího ventilu je nekorektní generování geometrických elementů, pro které výpočet kolabuje. Je to zřejmě v důsledku ostré špičky škrtícího ventilu.



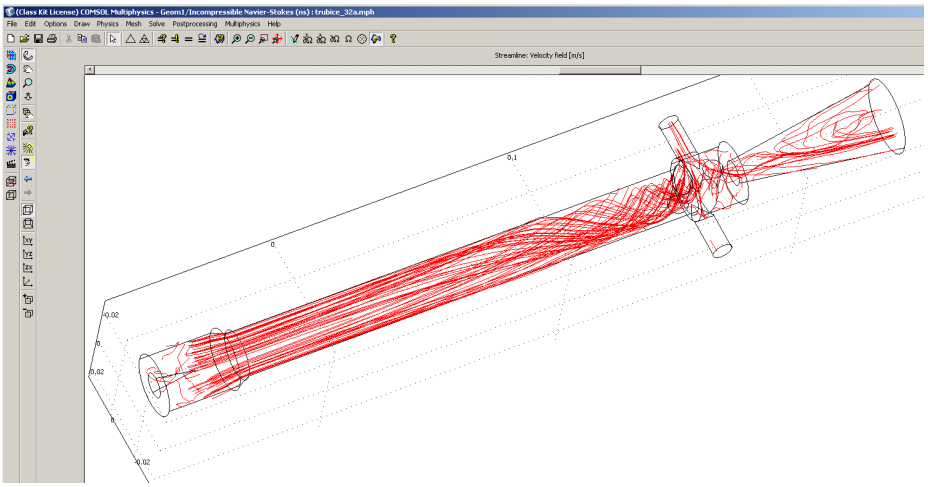
Obr. 11 Ostrá špička kuželu

Z tohoto důvodu jsem udělal **dvě další varianty modelu** s úpravou ostré špičky. Jedna je s komolým kuželem, další se zaoblenou špičkou kuželu. První varianta je na dalším obrázku.



Obr. 12 Komolý kužel

Pro omezené možnosti výpočtů na školní verzi programu COMSOL jsem provedl aspoň **simulaci proudění vzduchu v trubici**. Toto proudění je ale spočteno pouze pro malé rychlosti a bez přenosu tepla. Ale aspoň jsem se seznámil s novým softwarem, který se v technické praxi používá stále častěji. Výsledek tohoto výpočtu je na následujícím obrázku.



Obr. 13 Proudění v trubici

V současnosti jsou geometrická data předána firmě Humusoft **k dalšímu výpočtu proudění a přenosu tepla**. Tento výpočet nemůžeme provádět na naší škole, protože nemáme zakoupené již uvedené moduly a také náš hardware nestačí na obsáhlé matematické výpočty (firma má k dispozici dvoujádrové PC s operační pamětí několika GB). **Firma také bude konzultovat případné další problémy s dodavatelem programu COMSOL ve Švédsku**.

8. Závěr a diskuse výsledků

Práce mi umožnila se seznámit s velmi zajímavým zařízením, které slouží pro chladící účely. Nastudoval jsem **mnoho aplikací** Rangueho trubice a seznámil jsem se s některými **výpočty**. V další části práce jsem provedl **experimentální ověření** činnosti trubice. Poslední částí byla snaha o **simulaci proudění a přenosu tepla pomocí MKP**. Zde jsem se naučil pracovat s grafickými formáty a s vlastním programem COMSOL. Pokud v blízké budoucnosti obdržím výsledky z firmy Humusoft, budu moci udělat jeden z následujících závěrů:

1) výpočty **potvrdí rozložení teplot** na studeném a horkém konci trubice
2) výpočty **rozložení teplot nepotvrdí**

V případě varianty ad 2) jsem přesvědčen, že uvedený program COMSOL neobsahuje potřebné rovnice popisující fyzikální jev v Rangueho trubici (většinou parciální diferenciální rovnice). Přesto ale tato trubice pracuje, používá se v praxi a její činnost je mnohokrát potvrzena**. Trubice zkrátka čeká na vyřešení fyzikální záhady.**

Předpokládám, že při další obhajobě SOČ budu schopen dodat další výsledky.

9. Seznam použité literatury

Ke zpracovávanému tématu jsem nesehnal žádnou knižní publikaci. Informace jsem čerpal z firemních materiálů firmy VORTEC a z následujících internetových zdrojů:

[1] http://www.visi.com/~darus/hilsch/

[2] http://www.southstreet.freeserve.co.uk/rhvtmatl/index.htm

[3] http://www1.vsb.cz/ke/vyuka/CHZ&TC/CHLADICI1.pdf

[4] http://www.lontech.cz/virove-trubice.html

[5] www.ksp.vslib.cz/cz/kpt/obsah/ lidi/behalek/Termoplastyvpraxi.pdf

[6] http://en.wikipedia.org/wiki/Vortex_tube

[7] http://www.airtxinternational.com/how_vortex_tubes_work.php

[8] http://www.vortec.com/vortex_tubes.php

[9] www.andrews-space.com/images/videos/PAPERS/Pub-ExperimentalResultsOfAVortexTubeAirSeparator(200307).pdf

[10] http://www.google.com/patents?id=AZNSAAAAEBAJ&dq=vortex+tube

[11] http://www.google.com/patents?id=L7w7AAAAEBAJ&dq=vortex+tube

[12] http://www.google.com/patents?id=skFuAAAAEBAJ&dq=vortex+tube

[13] http://www.google.com/patents?id=QMY4AAAAEBAJ&dq=vortex+tube

[14] http://www.google.com/patents?id=tb1YAAAAEBAJ&dq=Ranque