

STŘEDOŠKOLSKÁ TECHNIKA 2013

Setkání a prezentace prací středoškolských studentů na ČVUT

Návrh systému inteligentního domu

Pavel Mikšík

Brno 2013

Setkání a prezentace prací středoškolských studentů na ČVUT



Návrh systému inteligentního domu

Autor: Pavel Mikšík

**Škola: Střední průmyslová škola elektrotechnická a informačních
technologií, Purkyňova 97, Brno**

Konzultant: Ing. Jaroslav Nesvadba, CSc.

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně, použil jsem pouze podklady (literaturu, SW, atd.) citované v práci a uvedené v příloženém seznamu a postup při zpracování práce je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), v platném znění.

V Brně dne 8. 1. 2013

Podpis:

ANOTACE

Cílem práce bylo sestavit funkční modelový příklad systému řízení inteligentního domu, který by bylo možno ovládat přes program na PC, s velkou možností rozšíření pro další spínací prvky.

OBSAH

Úvod	6
1. Inteligentní dům	6
1.1. Popis návrhu domu	6
1.2. Blokové schéma	7
1.3. Popis jednotky vypínače (slave)	7
1.4. Popis řídicí jednotky (master)	12
1.5. Popis řídicího programu pro PC	17
2. Závěr	20
3. Použitá literatura	21

Úvod

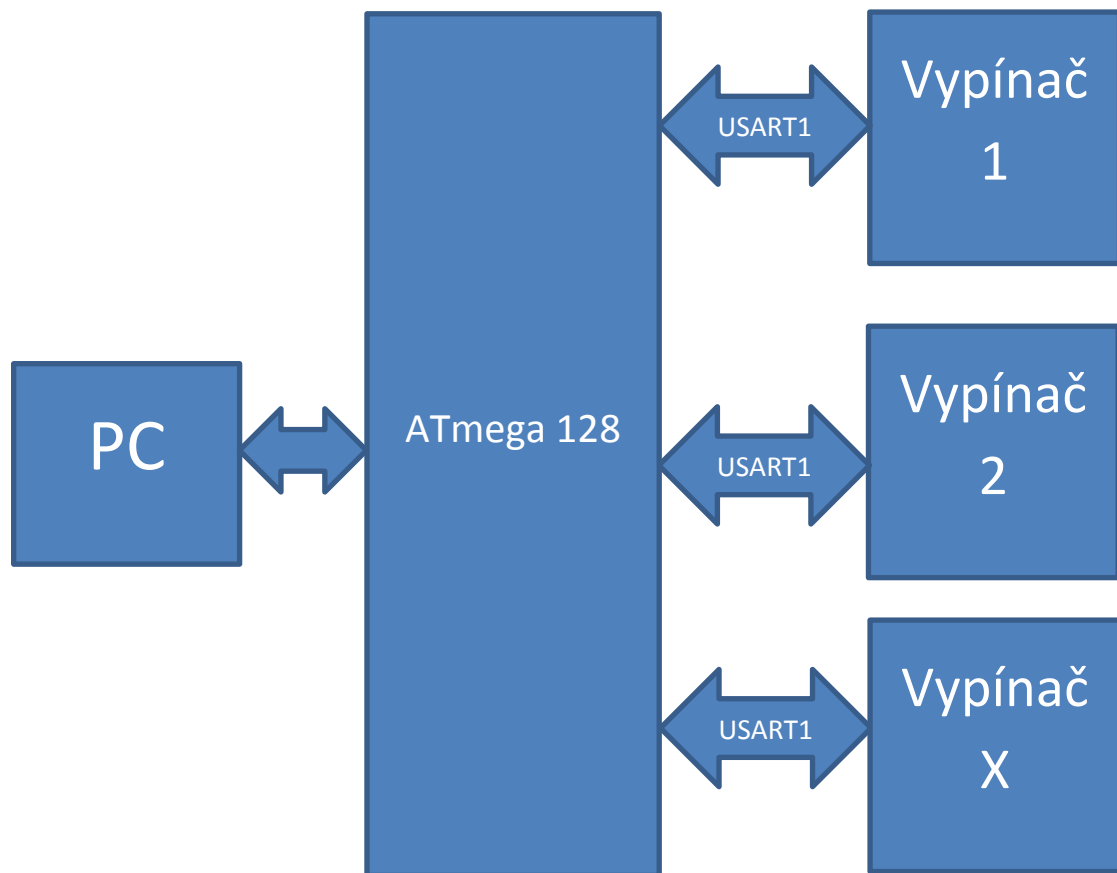
Technika se stala nedílnou součástí našich životů. Čím dál tím více se setkáváme s „chytrými“ zařízeními, jako jsou například chytrý telefon, inteligentní pračka a inteligentní myčka, ale jsou tu i mnohem komplexnější zařízení, třeba tzv. inteligentní dům. U moderních staveb jsou inteligentní systémy první linií obrany před lupiči. Například je možno si nastavit, aby se každý den v přesně nastavenou hodinu po setmění zapnulo osvětlení v kuchyni a o 10 minut na to zhaslo. Takto lze naprogramovat celé osvětlení domu a tento pak budí zdání, jako kdyby tam někdo byl, ale přitom je to jen soubor chytře zadaných příkazů.

1. Inteligentní dům

1.1 Popis návrhu domu

Inteligentní dům má 3 části. První část jsou samotné vypínače, druhou část tvoří řídicí jednotka, která obsluhuje komunikaci po lince RS-485 a třetí částí je řídicí software v PC. Toto členění systému bylo zvoleno z praktického důvodu, protože první část je tvořena tak, že i při výpadku řídicích částí je zaručena funkčnost systému, druhá část zajišťuje správnou komunikaci všech vypínačů, bez nutnosti mít připojené PC

1.2 Blokové schéma



1.3 Popis jednotky vypínače (slave)

Každá jednotka vypínače je navržena tak, aby mohla pracovat nezávisle na ostatních. Jedinou podmínkou je stálý přísun stabilizovaného napětí pro funkčnost vypínače.

Při stavbě na nepájivém poli bylo využito jak klasické DIL verze tak i SMD, pro kterou byla navržena konstrukce převodníku (redukce). Převodník byl využit pouze jako testovací vzorek k ověření funkčnosti zařízení před konečnou stavbou na finální DPS.

Vypínač je konstrukčně navržen tak, aby se vešel do standardizované vypínačové krabice. Tuto kompatibilitu zajišťují tvary plošného spoje. Některé součástky tuto kompatibilitu nezaručují, proto se musí vybírat součástky se sníženou konstrukční

výškou. Z bezpečnostních důvodů je nutno dát pozor na to, že konstrukce bude pracovat se síťovým napětím připojeným na svorkovnici, která je umístěna na plošném spoji. Dále je zde umístěn standardní 10 pinový programovací konektor (SPI), přes který se provádí aktualizace softwaru vypínače. Plošný spoj je většinou navržen v SMD verzi zajišťující úsporu místa, kterého je ve vypínačové krabici málo. Prvky, které v SMD nejsou zahrnuty, jsou převážně prvky výkonové nebo prvky, které se v SMD verzi nehodí do tohoto typu elektrotechniky. Na plošném spoji jsou umístěny 2 drátové propojky, které nijak neomezují funkčnost nebo kvalitu zpracování.

Pro spínání výkonové části bylo využito DC relé s ochranou diodou. Protože maximální proud, který MCU dokáže bezpečně dodat je 20mA a relé potřebuje 40mA, byl zde z tohoto důvodu umístěn zesilovací tranzistor, který zajišťuje správnou funkci relé.

Na plošný spoj se zapojuje jakýkoliv vypínač nebo přepínač a přes tento prvek bude téct minimální proud i napětí. Tento vypínač je napojen na řídicí MCU, ze kterého je možné odebírat max. 20mA a 5V.

Na software v MCU je kladena podmínka jednoduchosti a dále je zohledněna přítomnost komunikační sběrnice RS-485.

Popis schématu

Napájecí napětí je přivedeno na usměrňovací diodu, která při případném přepólování zajišťuje, že se komponenty na desce vypínače nepoškodí. Za diodou je napájení rozvedeno na pin napájení převodníku (na linku RS-485) a na řídicí čip Atmega8. V neposlední řadě pak na zbylé součástky (spínací relé, pomocný tranzistor, atd.). Ihned po přivedení napětí k převodníku RS-485 je přivedeno i na pul-up rezistor, který zajišťuje správné fungování jednotky USART v Atmega8 (bez použití tohoto rezistoru by MCU dostávalo neúplné nebo žádné data z převodníku). Dále je napájení přivedeno na pomocný tranzistor. Tento tranzistor zajišťuje správné sepnutí cívky relé. Bez použití tohoto tranzistoru by nebylo možné sepnout relé, protože MCU má maximální spínací proud 20 mA. Dále jsou při vstupu komunikační linky umístěny rezistory, které podle datasheetu obvod vyžaduje pro správnou funkci. Aby plošný spoj neležel přímo na krabici, jsou na jeho okraji umístěny pájecí body, na které se nanese větší množství cínu a tato úprava slouží jako „nožičky“.

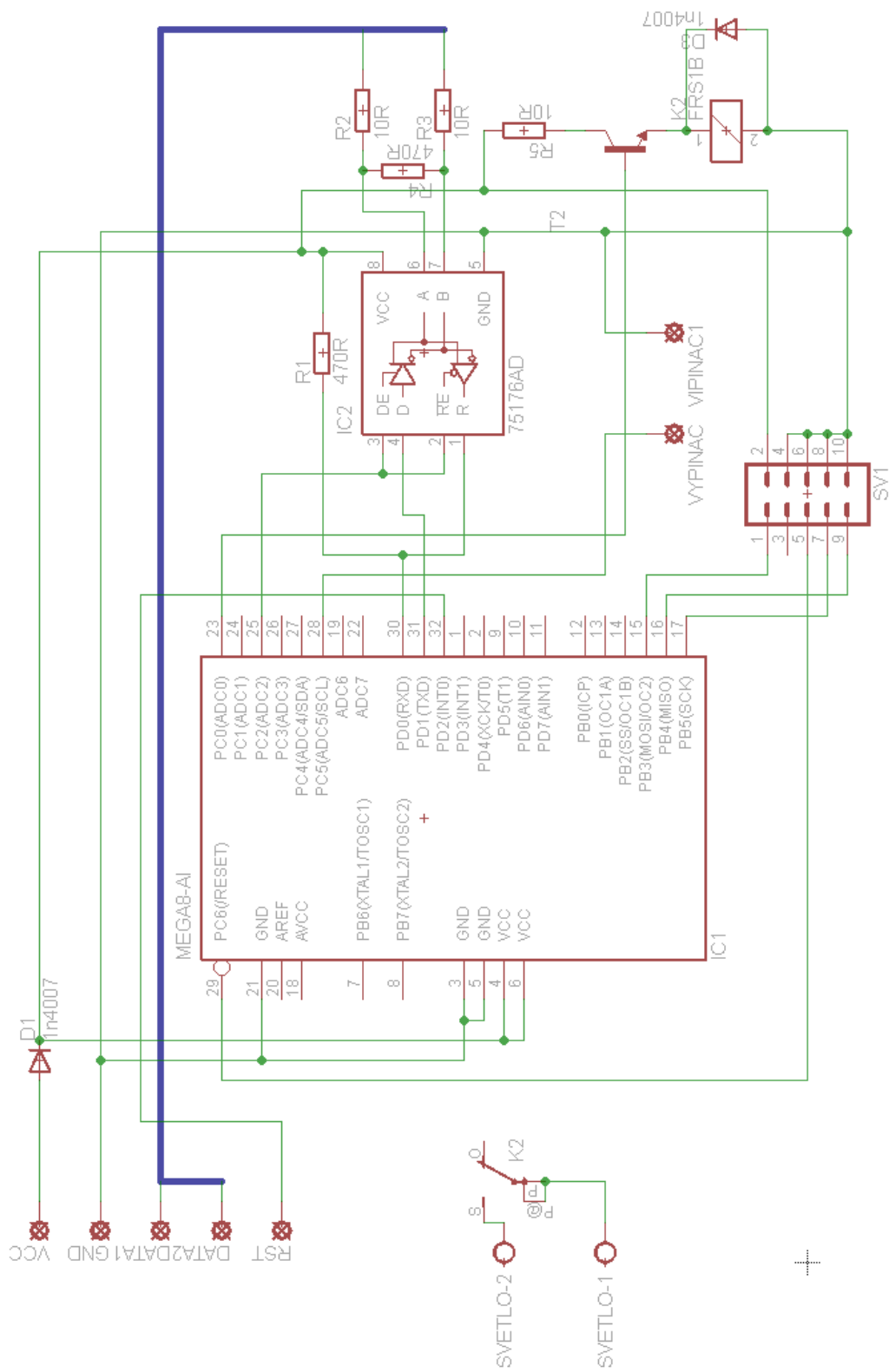


Schéma zapojení vypínače



Foto: modul vypínače z pohľadu súčastek

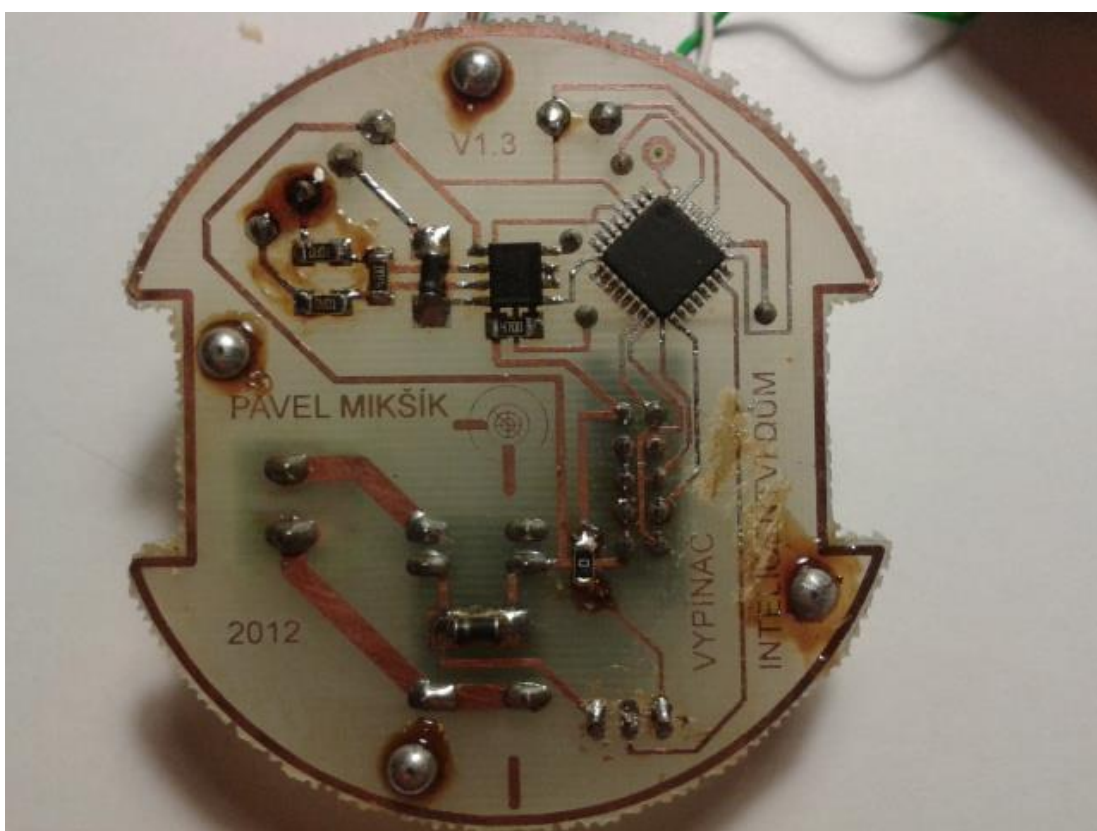
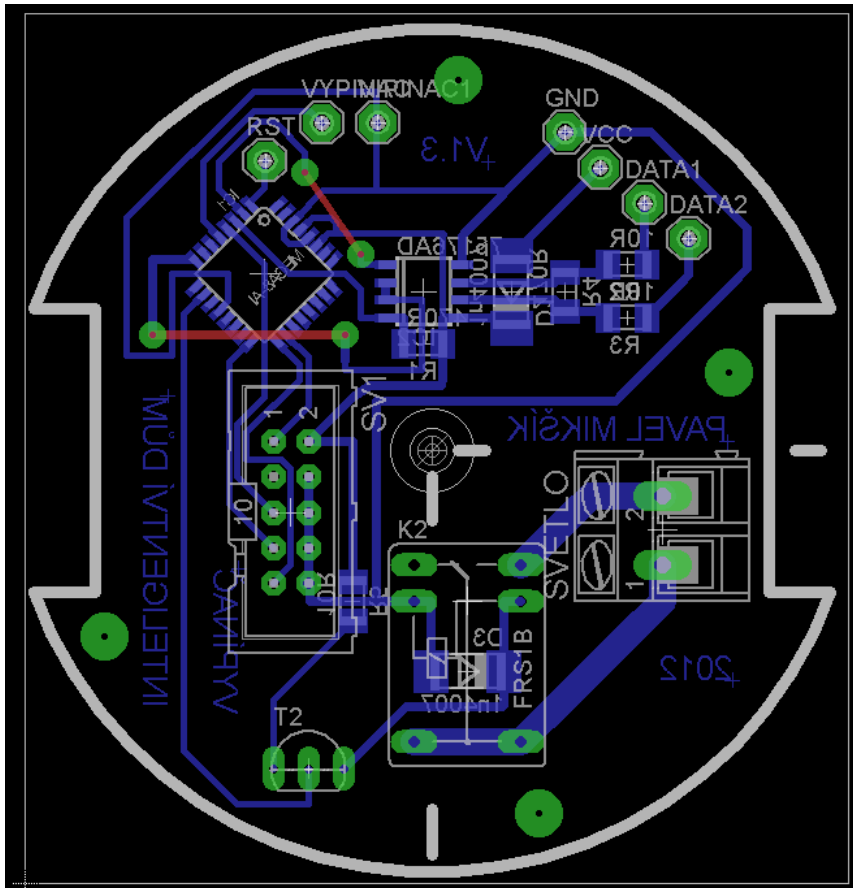


Foto: modul vypínače z pohľadu plošného spoje



Obraz plošného spoje z programu eagle

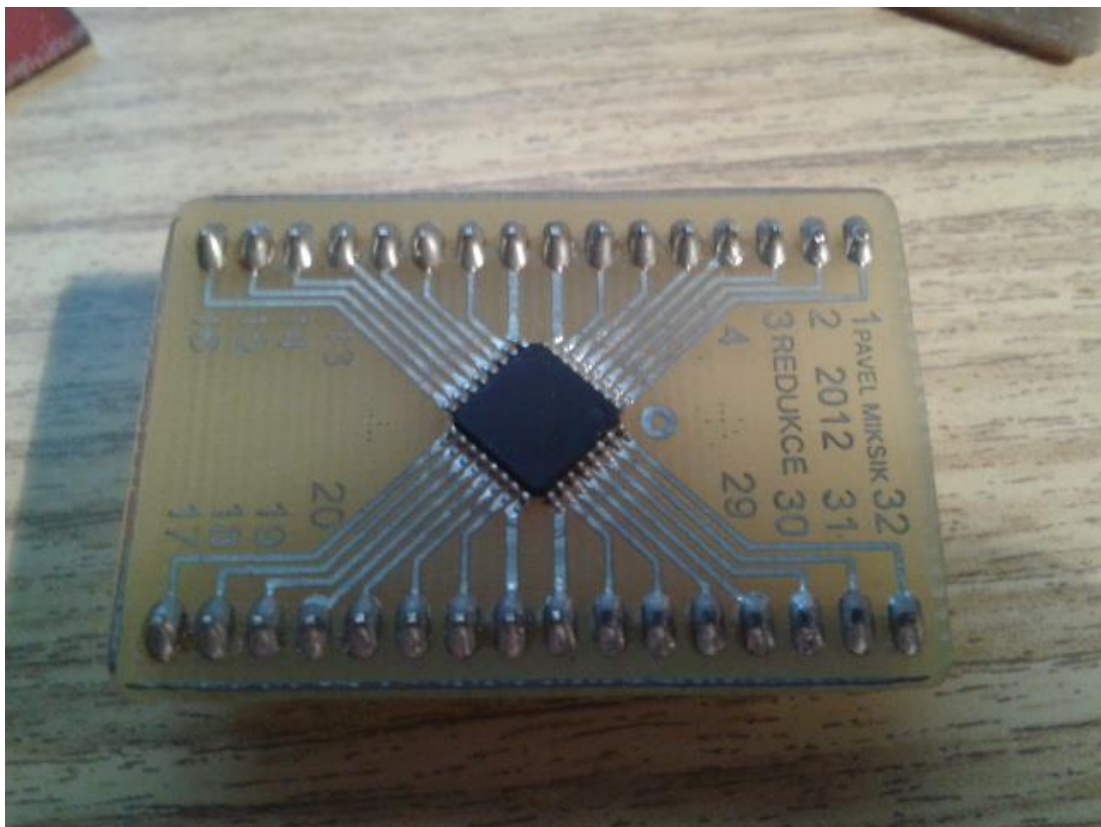


Foto: redukce Atmega8

1.4 Popis řídicí jednotky (master)

Jádrem této části je nepochybně MCU z řady Atmega128, která jako jedna z mála má 2 USART jednotky. V této konstrukci byly využity obě dvě, jedna jako komunikační sběrnice s ostatními vypínači a druhá jako komunikační rozhraní pro PC. Pro napájení řídicího čipu bylo využito napájení 5V místo doporučených 3,3V, které Atmega128 také podporuje. TTL logika ulehčila práci s převodníky na linku RS-485, které používají TTL logiku.

Pro komunikaci s PC bylo využito externí přerušení při příjmu na USART jednotce. Každý znak, který přijde, se ukládá do proměnné typu char pole o maximální velikosti 128 znaků i s ukončovacími, které tvoří vždy dvojznak („\n“ nebo „\r“ a „\0“). Při práci v programu je k hodnotám přistupováno ne jako k cece, ale jako k jednotlivým proměnným. Ukončovací znaky využíváme jenom při komunikaci s ostatními periferiemi.

Při návrhu zařízení vyvstal závažný problém. Atmega128 se vyrábí pouze v SMD verzi, což znemožňovalo její zapojení do standartního nepájivého pole. Proto byla navržena konstrukce převodníku. Na tuto redukci je Atmega128 připájena ručně pomocí mikropájky se speciálním hrotem. Jsou zde vyvedeny všechny piny a očíslovány každý svým číslem (čísla pinů jsou shodná s datasheetem). Dále jsou zde 4 otvory, dva na kraji DPS, tyto mají sloužit pro případné připevnění redukce k nějaké podložce a dva ve střední části.

Popis schématu

Schéma a i konstrukce se skládá z 3 hlavních částí 1) převodník na USB, 2) RS-485 modul a 3) samotný řídicí čip. USB převodník je realizován integrovaným obvodem od firmy FTDI. Tento čip nese sériové označení ftr-232. Pro správnou funkci stačilo k tomuto obvodu připojit pouze jednu externí součástku a to kondenzátor, který jak většina součástek je také v SMD verzi. Hodnota kondenzátoru byla udána v originálním datasheetu $c = 100\text{nF}$. Tento čip je připojen na napájení a dále od něj vedou pouze dva vodiče TX a RX. Přes tyto vodiče probíhá vlastní komunikace s hlavním řídicím čipem

Atmega128. Na tomto čipu běží hlavní program, který se stará nejen o komunikaci s PC, ale i o linku RS-485. Čip je v tzv. master modu, neboli řídicím módu. Program pro tento čip byl napsán v jazyce C. Čip má jako jeden z mála 2 jednotky USART a je tedy ideální pro toto zapojení. Dále je k hlavnímu čipu připojen programovací konektor, kterým se provádí samotné naprogramování.

Popis DPS

O přívodní kabely se stará svorkovnice, která se současně postará i o kabely, vycházející z tohoto plošného spoje. Dále je tu konektor USB, který zabezpečuje připojení ke kontrolnímu počítači, na kterém běží ovládací software. Pro programování čipu na desce je zde 10 pinový programovací konektor, který slouží pouze tomuto účelu.

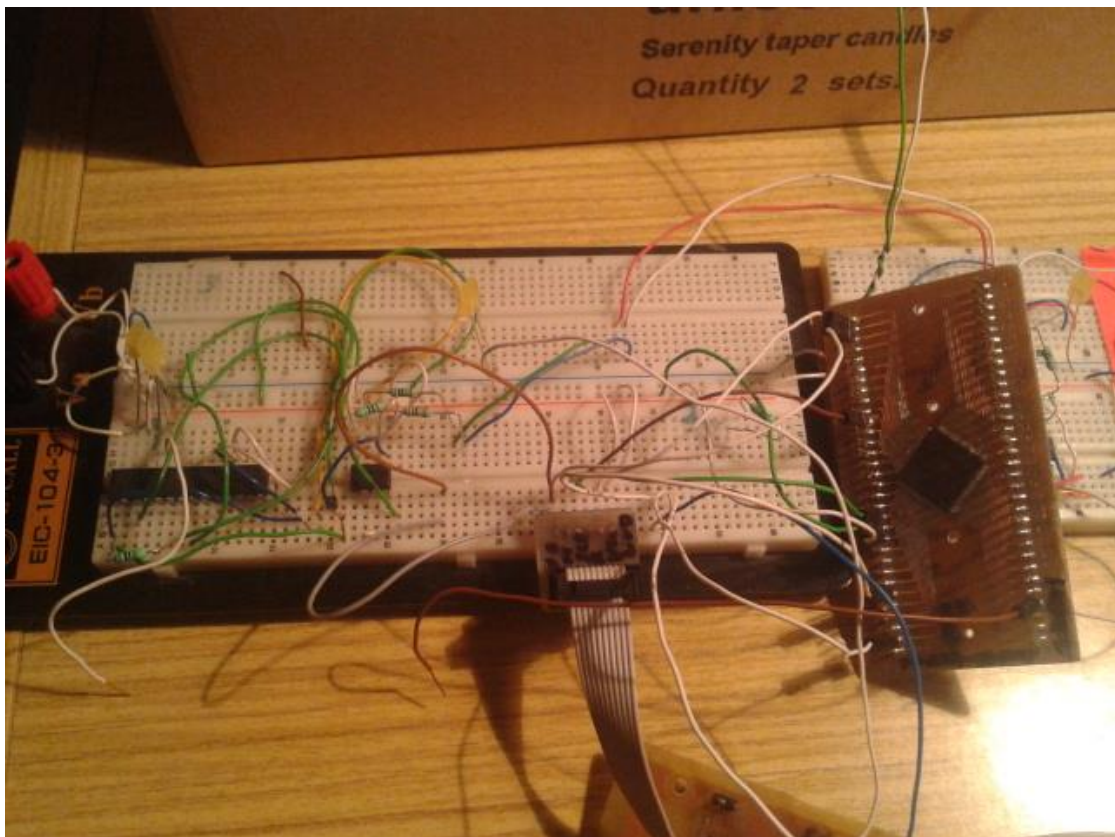


Foto: konstrukce řídicího modulu ve vývoji

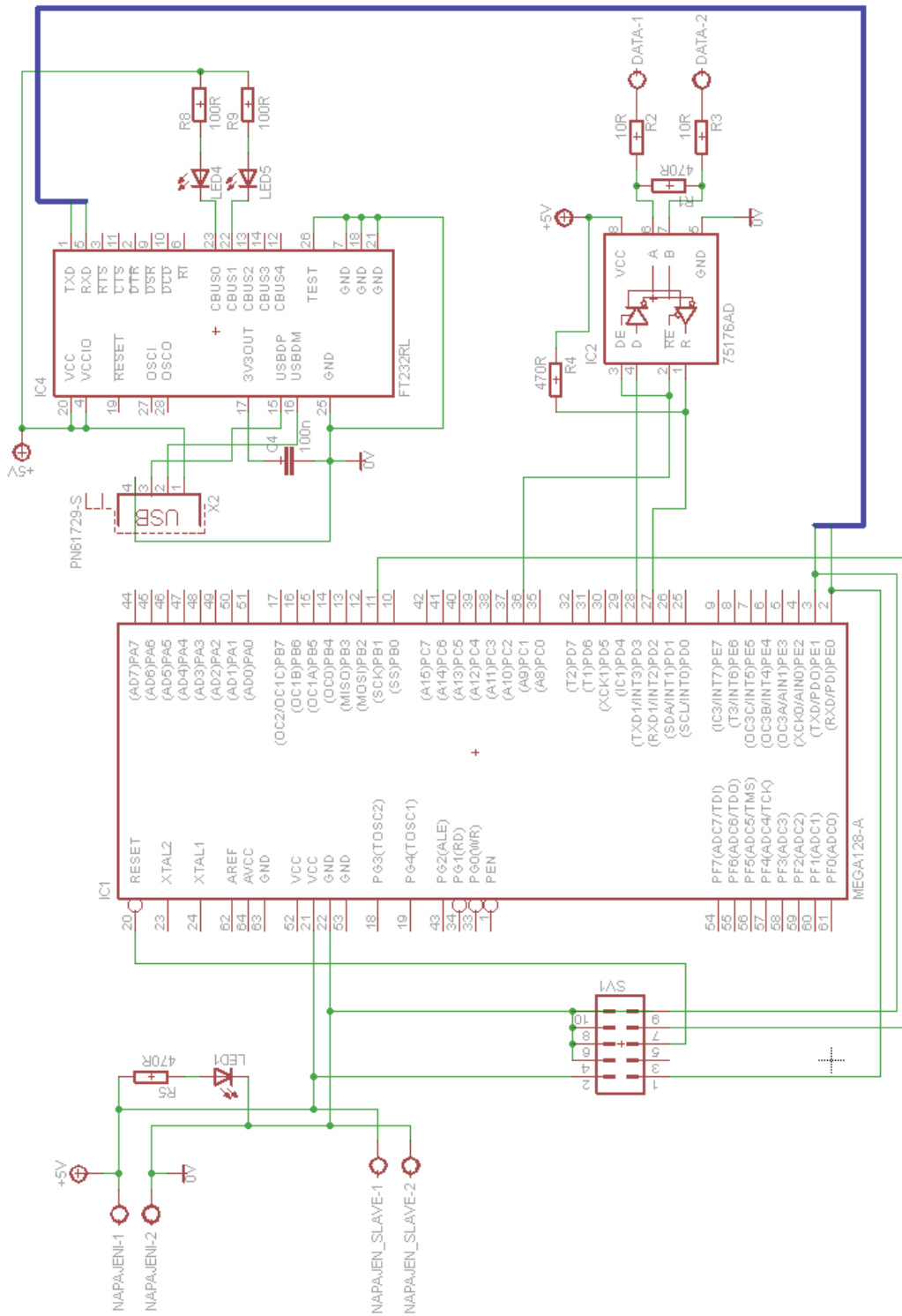


Schéma řídicího modulu

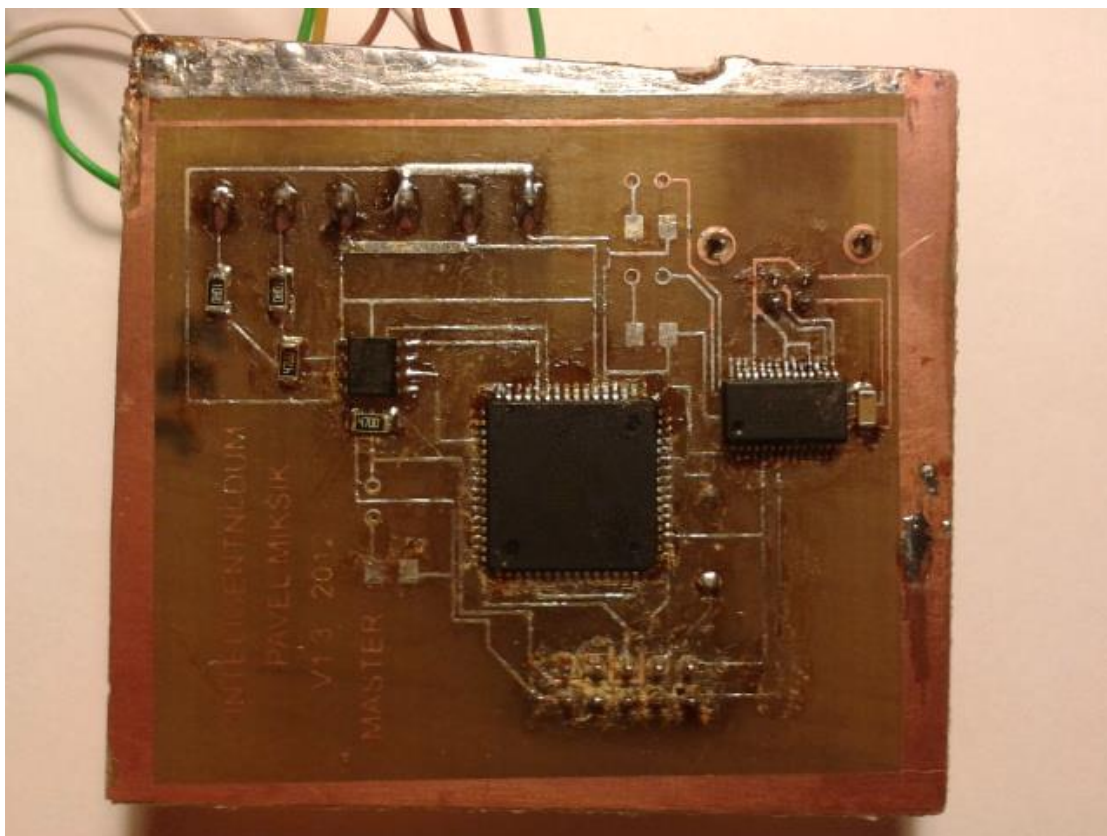


Foto: řídicí modul z pohledu plošného spoje

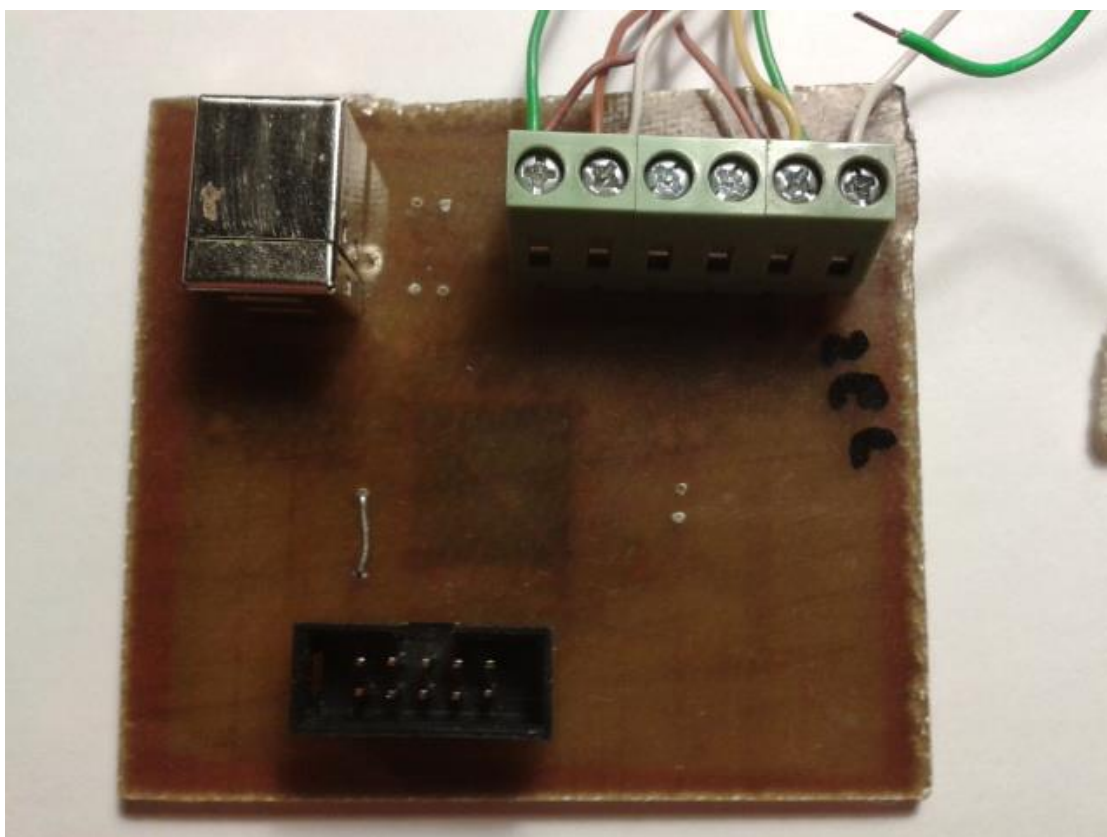


Foto: řídicí modul z pohledu součástek

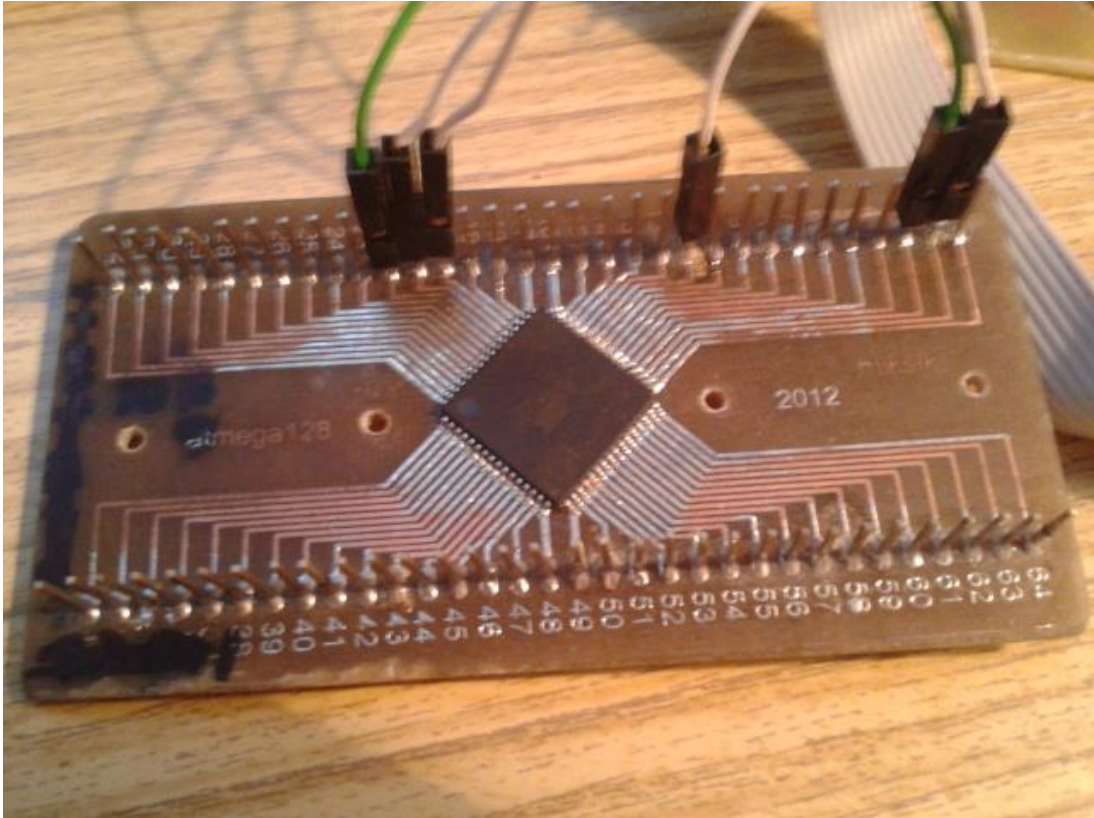
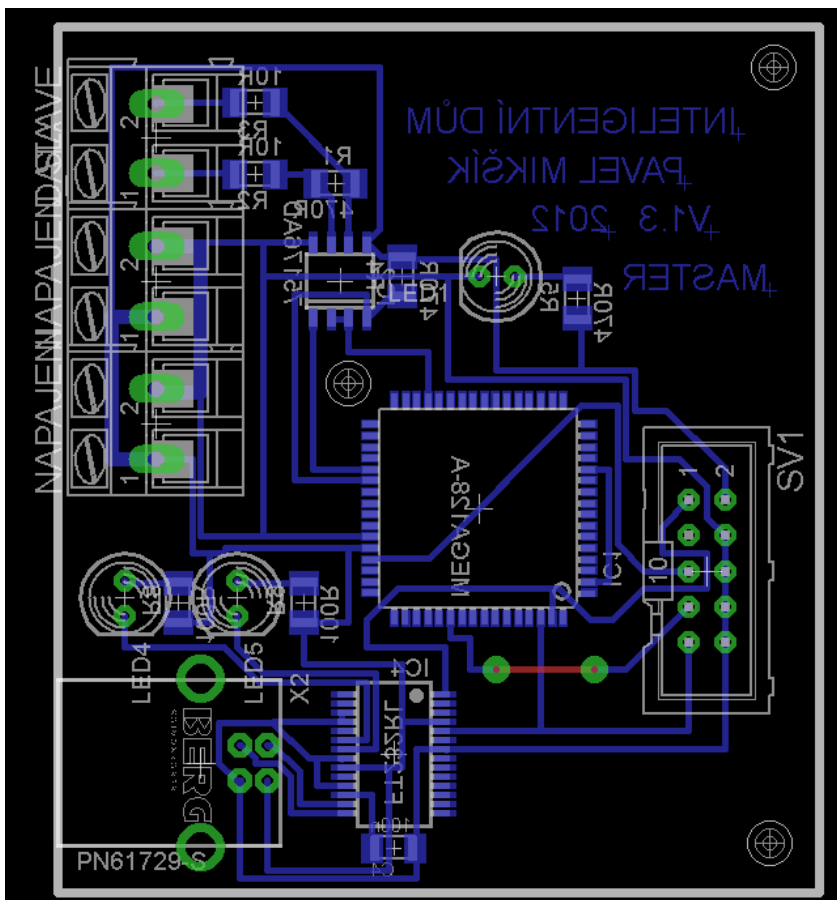


Foto: redukce Atmega128



Obraz plošného spoje masteru

1.5 Popis řídicího programu pro PC

Pro naprogramování řídicí části pro osobní počítače byl použit programovací jazyk C sharp a vývojové prostředí Visual studio 2013. Ovládací program je přizpůsoben pro prezentaci zařízení a jen nepatrnou změnou se dá rozšířit o N dalších vypínačů. V programu jsou využity základní programovací komponenty, např. button label, atd. Pro komunikaci PC a masteru je využita komponenta serial port, která zprostředkovává přenos informací na námi zadaný port. Je nutno dodržet nastavenou komunikační rychlost (19200 Bd). Tato rychlost je nastavena i na masteru jako komunikační rychlost.

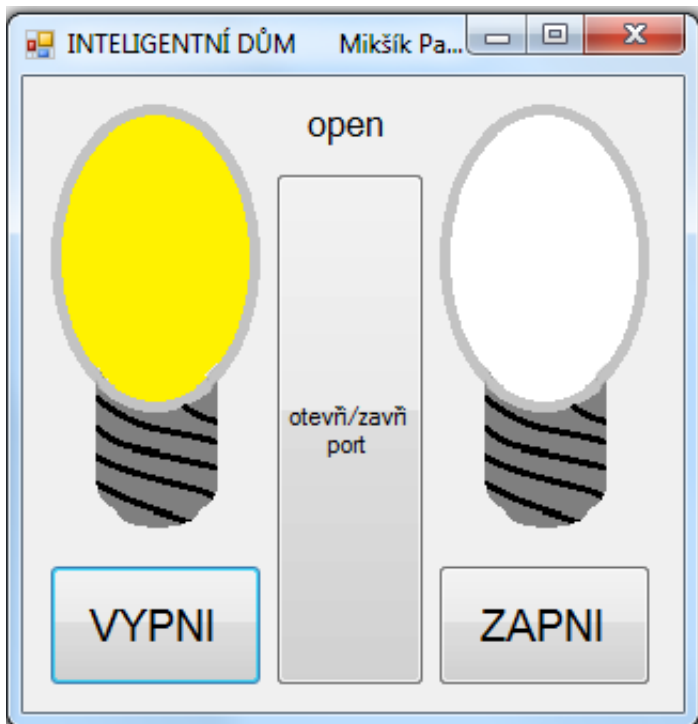
Při psaní programu nastal problém se zjišťováním přijatých dat, protože serial port v C sharp běží na svém vlastním vlákne a v samostatné metodě. Tato metoda nechtěla vracet žádné proměnné do hlavního kódu. Tento problém byl vyřešen tím, že namísto složitěho předávání proměnných mezi vlákny zde byly využity globální proměnné, ke kterým může metoda svobodně přistupovat z kterékoliv části kódu bez jakýchkoli potíží. Dále nastal problém z vypisováním přijatých dat. Program totiž vyčkával na uživatele a až poté kontroloval, jestli se přijaté proměnné shodují s grafickým vyjádřením (GUI). Proto zde byl využit další prvek a to timer, který byl nastaven tak, že po každých 500ms vyvolá přerušování a aktualizuje GUI.

Při programování řídicího programu byla využita možnost, kterou C sharp nabízí, a to zobrazení konzole, která byla při programování velice nápomocná, protože umožňovala opakovaný zápis přijatých hodnot, což zjednodušilo práci při ladění programu.

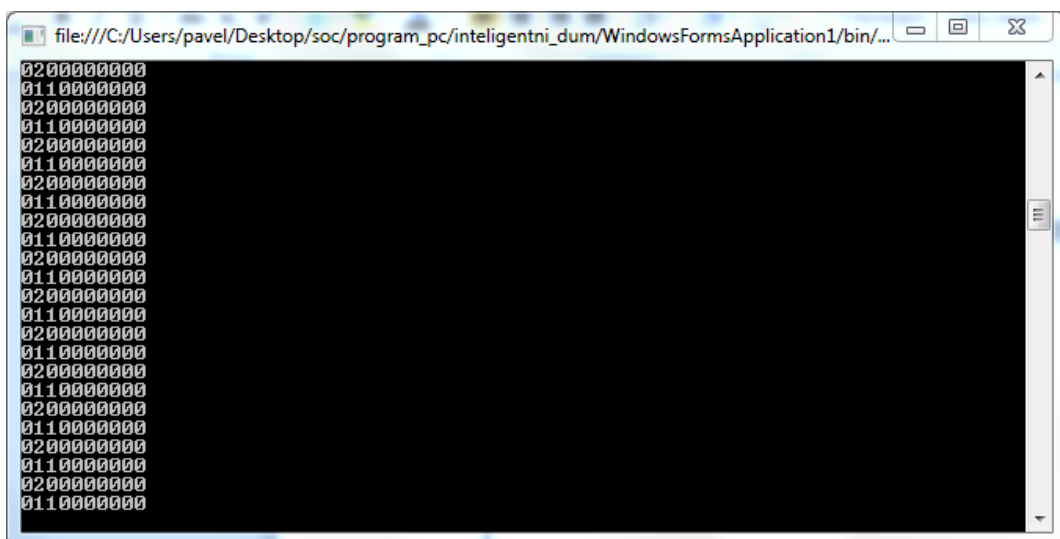
Popis programu

Program je sepsán tak, že po spuštění programu je nejprve nutné otevřít komunikační port. Tento port se spojí s master modulem a každých 1500ms vyšle dotaz na 1 čidlo. Posléze dostane odpověď, tato se uloží do globální proměnné a dále je zpracována příslušnými metodami. Byly vytvořeny dva symboly žárovky, jeden svítící a druhý zhaslý. Tyto obrázky fungují tak, že pokud žárovka svítí, je pro obrázek zhaslé žárovky nastavena vlastnost visible false, což znamená, že obrázek je skryt a pro obrázek svítící

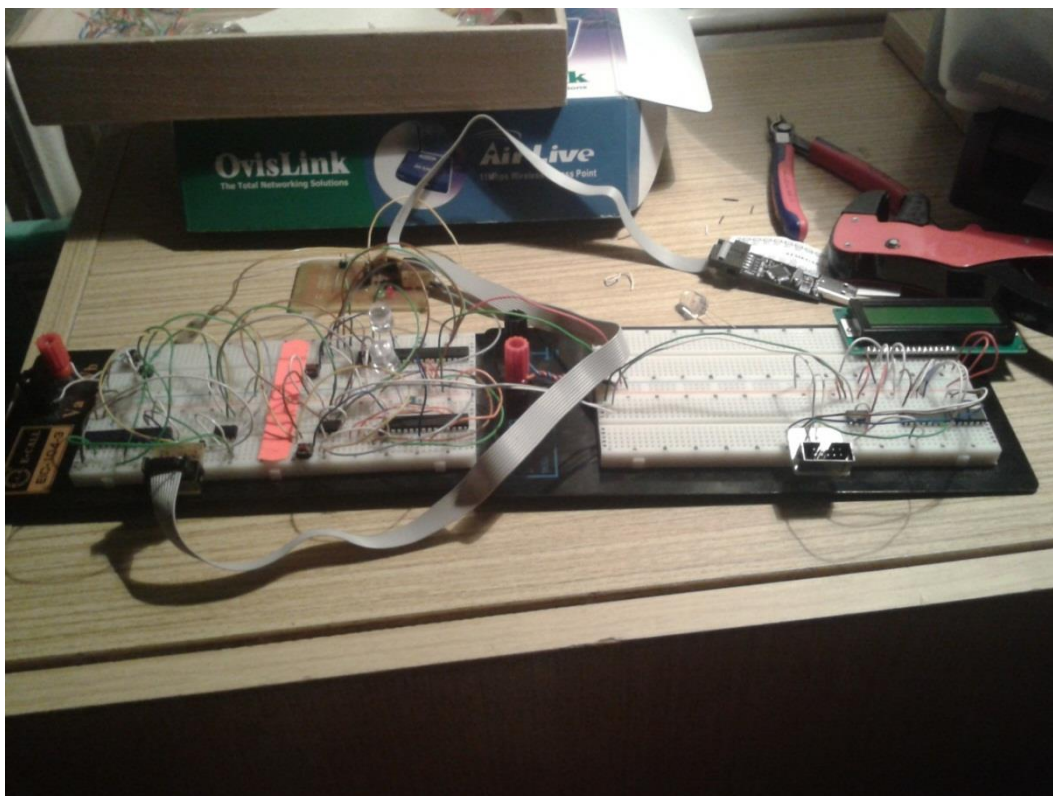
žárovky je nastavena vlastnost visible true (obrázek je vidět). Tyto obrázky jsou položeny na sobě a rychlost programu zajistí, že změna stavu bude pouhým okem nezaznamatelná.



Okno programu (grafický výstup programu)



Výstup dat z konzole



Průběh konstrukce: vývojová část

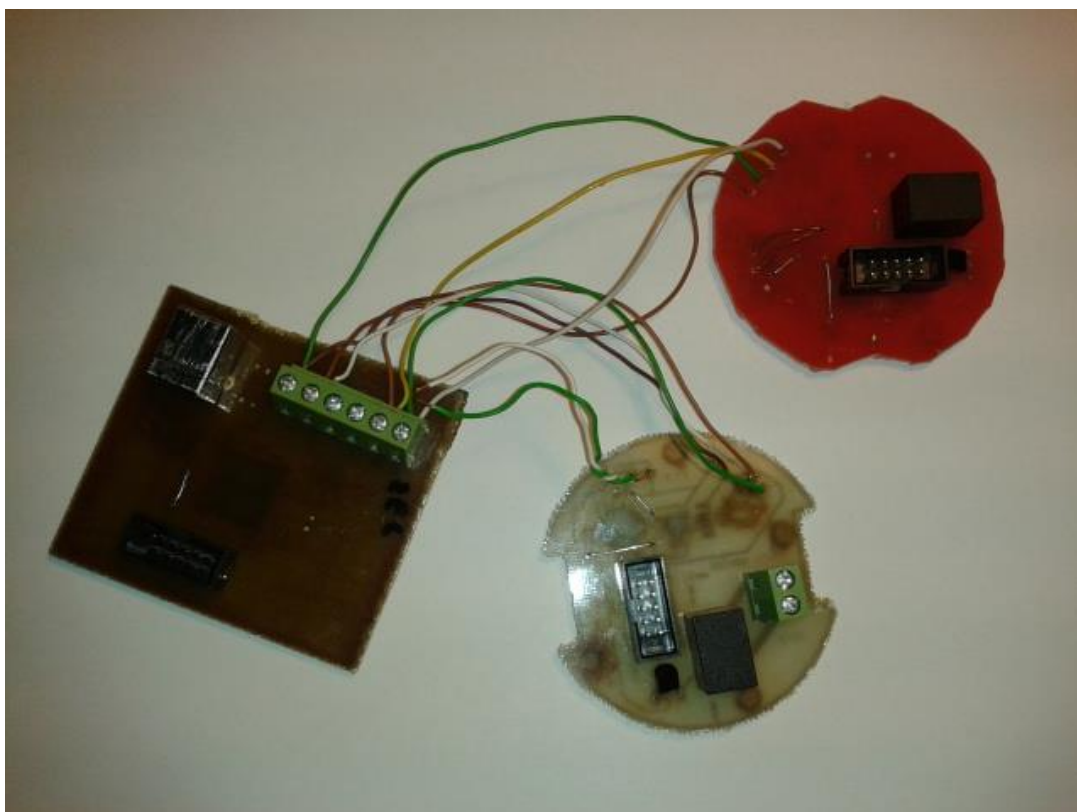


Foto: dokončené zařízení z vrchního pohledu

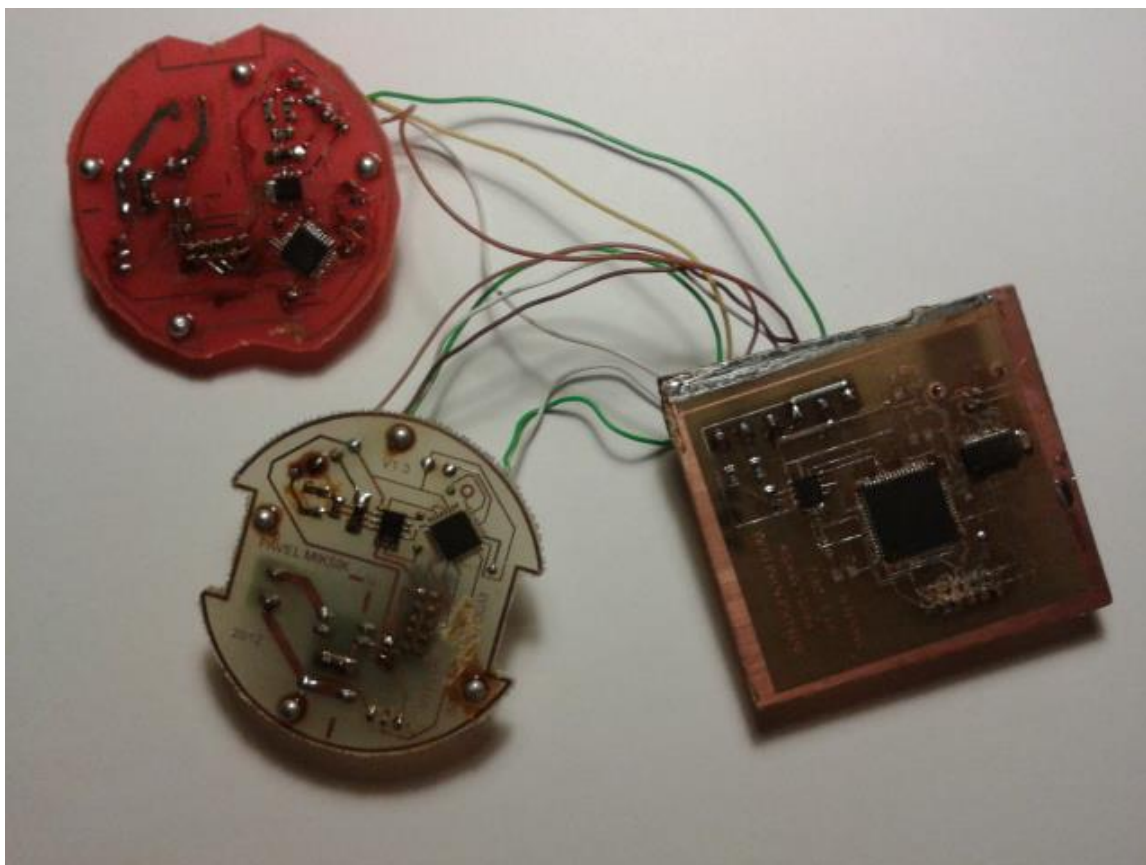


Foto: dokončené zařízení z vrchního pohledu

2. Závěr

Cílem práce bylo sestavit funkční model inteligentního ovládní světla a také se zdokonalit v elektrotechnice a programovacích jazycích C a C Sharp. To se mi, podle mého názoru, povedlo. Pro programování v těchto jazycích jsem využíval nejnovější programovací prostředí (C AVR studio, C Sharp visual studio), dále jsem se úspěšně zdokonalil v návrhovém systému plošných spojů eagle a v samotné výrobě plošných spojů, které jsem si vyráběl doma.

3. Použitá literatura

Internet:

<http://www.avrfreaks.net>

<http://pandatron.cz/>

<http://cs.wikipedia.org/wiki/RS-485>

<http://www.hw.cz/teorie-a-praxe/dokumentace/rs-485-422.html>

Knihy:

LÁNÍČEK, R. *Elektronika – obvody – součástky – děje*. Praha: Nakladatelství BEN – technická literatura, 1998, ISBN 80-86056-25-2

Datasheety:

ftt-232 - http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

bc547 - http://www.datasheetcatalog.org/datasheet/philips/BC546_547_3.pdf

Atmega128 - <http://www.atmel.com/Images/doc2467.pdf>

Atmega8 - http://wvshare.com/datasheet/ATMEL_PDF/ATmega8.PDF