



Středoškolská technika 2011

Setkání a prezentace prací středoškolských studentů na ČVUT

AVR Testboard 2.0

Karel Babický

SPŠ a VOŠ Písek

Karla Čapka 402, 397 01 Písek

Anotace

Práce zpracovává téma vytvoření vývojové desky s procesory Atmel AVR a vytvoření výukového programu pro programování těchto procesorů s využitím dané desky. Zapojení desky bylo vyzkoušeno na nepájivém poli, poté na prototypové desce vyrobené v amatérských podmínkách a nakonec byla vytvořena tato konkrétní dvouvrstvá deska. Tato deska byla již vyrobena profesionálně a byla ručně pájená. Po realizaci desky byl vymyšlen výukový program, který vyučuje práci s deskou a základy programování procesorů ve vyšším programovacím jazyce C. Jako vývojové prostředí pro programování a pro výukový program bylo použito prostředí CodevisionAVR.

Klíčová slova

AVR Testboard 2.0, vývojová deska, ATmega16, výukový program, Eagle, CodevisionAVR

Obsah

Úvod	1
1 Teoretický rozbor AVR mikrořadičů	2
1.1 Avr procesory obecně	2
1.2 Procesor ATmega16	3
Parametry procesoru	4
Popis pinů procesoru:.....	4
2 Postup vývoje μ P aplikace	4
2.1 Úvod	4
2.2 Konkrétní vývoj aplikací za pomoci vývojové desky	5
3 HW návrh AVR Testboardu 2.0.....	6
3.1 Úvod k návrhu	6
3.2 Popis jednotlivých částí	6
Napájení	6
Hlavní procesor ATmega16	7
LCD zobrazovač	8
Modul Led diod	8
Modul tlačítek	9
Akcelerometr.....	9
Bluetooth modul	11
Programátor	11
Přepínače.....	12
3.3 Návrh plošného spoje.....	13
Vrchní vrstva.....	14
Nepájitvá maska	14
Potisk.....	14
Spodní strana.....	14
3.4 Osazování	16
4 Výukový program pro desku	17
5 Technické parametry.....	19
Závěr.....	20
Seznam citací.....	21

Úvod

Cílem této dlouhodobé maturitní práce bylo vytvořit vývojovou desku pro procesory AVR. Pro tuto desku byl zvolen procesor ATmega16 z důvodu dobrých zkušeností z předchozí práce s tímto procesorem. Vývojová deska měla obsahovat podpůrné periferie k procesoru a měla být jako celek plně funkční bez dalších podpůrných zařízení. Dále bylo úkolem desku otestovat, jestli splňuje dané požadavky a vytvořit k ní výukový program, který by měl studenty naučit základy programování těchto procesorů s využitím dané desky.

Finálním výrobkem je profesionálně vyrobená deska plně funkční se všemi podpůrnými periferiemi s možností bezdrátového přenosu informací a výukový program, který má studenty krok po kroku naučit programovat procesory a nastavovat desku. Jako vývojové prostředí je použit program CodevisionAVR.

1 Teoretický rozbor AVR mikrořadičů

1.1 Avr procesory obecně

Architekturu procesorů AVR vymysleli dva studenti z Norského technického institutu (Norwegian Institute of Technology (NTH)) Alf-Egil Bogen a Vegard Wollan.

AVR procesory jsou RISC procesory postavené na Harvardské architektuře (mají oddělenou paměť pro program a data). Skládají se z třiceti dvou stejných 8bitových registrů, které mohou obsahovat jak data, tak i adresy. Přístup do registrového pole je možný v jednom strojním cyklu. To znamená, že lze v jednom strojním cyklu vykonat jednu aritmeticko-logickou operaci. Oba operandy jsou načteny operace provedena a výsledek zapsán zpět do registrového pole. To vše v jednom strojním cyklu.

Hlavní vlastností RISC procesorů:

- redukováná sada instrukcí
- obsahuje převážně jednoduché instrukce
- délka provádění jedné instrukce je vždy jeden cyklus
- délka (počet bitů) všech instrukcí je stejná
- mikroinstrukce jsou hardwarově implementovány na procesoru
- využívá se zde techniky řetězení instrukcí

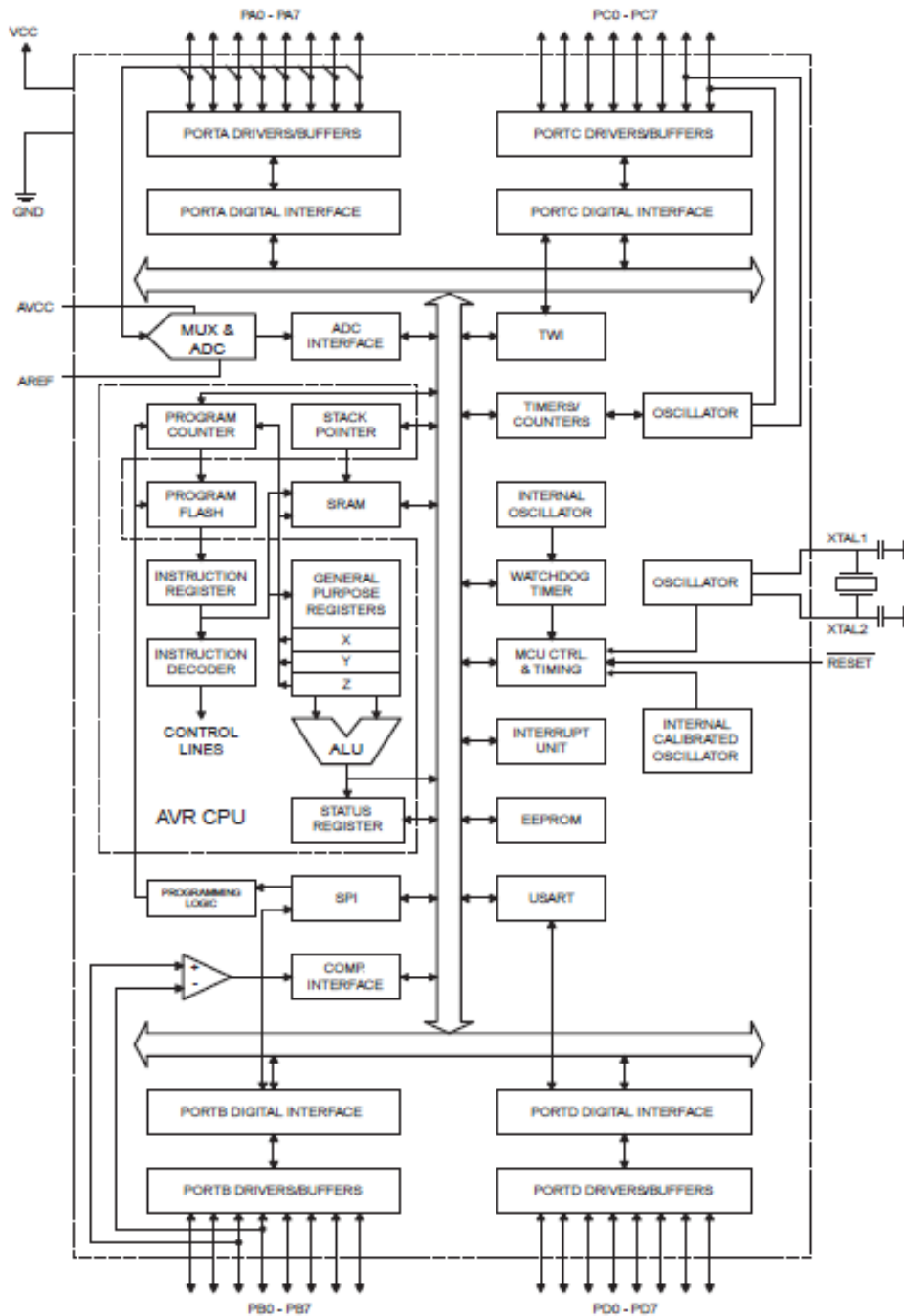
Tato technika dává procesorům AVR velký výpočetní výkon oproti například procesorům řady 8051, které pro některé instrukce potřebují 32 až 40 strojních cyklů navíc pro práci se musí používat akumulátor, který je v procesoru pouze jeden oproti AVR kde se registrové pole dá chápat jako skupina akumulátorů.

Procesory AVR jsou rozděleny do následujících řad:

- AT90
 - řada, která už se dnes nevyrábí, náhradou může být řada ATmega nebo ATtiny
- ATtiny
 - využívají se v jednoduchých a malých elektronických obvodech
 - Paměť flash pro uložení programu 1 – 8 kB
 - Pouzdro 6 – 32 pinů (orientačně)
 - Omezená sada integrovaných rozhraní
- ATmega
 - výkonné mikročipy, mají JTAG rozhraní, větší flash a RAM, více rozhraní.
 - Paměť Flash pro uložení programu 4 – 256 kB
 - V pouzdrech 23 – 100 pinů (orientačně)
 - Rozšířená sada instrukcí (instrukce násobení a instrukce přístupu k větší prg. pam.)
 - Široká sada integrovaných rozhraní
- XMEGA
 - 16–384 kB větší paměť programu
 - v pouzdrech 44–64–100 pinů
 - nová řada s většími možnostmi

1.2 Procesor ATmega16

Na obrázku 1. je znázorněno blokové schéma procesorů architektury AVR konkrétně procesoru ATmega 16, se kterým je realizována i celá vývojová deska. Z blokového schématu můžeme vyčíst, že součástí procesoru není jen samostatná výpočetní jednotka, ale i paměti konkrétně paměť typu flash, E²PROM, obsluha přerušení, dvěma 8bitovými čítači/časovači. K dispozici je dále také jeden 16bit čítač/časovač, analogový komparátor, jednotky USART, SPI, TWI a 10bit A/D převodník.



Obrázek 1.1: Architektura AVR (převzato z [4]).

Parametry procesoru

Na předešlém blokovém schématu je jednoznačně a přehledně znázorněno zapojení jednotlivých bloků v procesoru a přesně viditelná v předchozích odstavcích vysvětlovaná architektura.

V procesoru je kromě bloků čítačů, časovačů, jednotky přerušení, převodníku a jednotek pro komunikaci blok se stavovými a řídicími registry. V tomto bloku se nacházejí registry, ve kterých nastavujeme a testujeme jednotlivé stavy a režimy vnitřních jednotek procesoru. Každá periférie a část procesoru musí být před použitím právně nastavena, aby mohla být používána. Informace o například přerušení se nastavují také do dalších registrů.

Popis pinů procesoru:

- **GND** – uzemnění
- **VCC** – konektor pro připojení napájecího napětí
- **PORTA (PA0-PA7)** – může sloužit jako A/D převodník, nebo jako 8bitový vstupně/výstupní port (pokud není A/D převodník používán)
- **PORTB (PB0-PB7)** – může sloužit jako 8bitový vstupně/výstupní port, používá se pro programování prostřednictvím programátoru STK500 (porty PB5(MOSI), PB6(MISO), PB7(SCK))
- **PORTC (PC0-PC7)** – může sloužit jako 8bitový vstupně/výstupní port, JTAG programátor se připojuje k pinům PC5(TDI), PC3(TMS) a PC2 (TCK)
- **PORTD (PD0-PD7)** – může sloužit jako 8bitový vstupně/výstupní port
- **RESET** – vstup pro reset (schéma zapojení je uvedeno v kapitole základní deska s mikropočítačem)
- **XTAL1** – vstup do invertujícího oscilátoru a vstup do operačního obvodu vnitřních hodin
- **XTAL2** – výstup z invertujícího oscilátoru
- **AVCC** – je napájecí pin pro PORTA a A/D převodník (zpravidla se připojuje na VCC)
- **AREF** – AREF je analogový referenční pin pro A/D převodník

2 Postup vývoje μ P aplikace

2.1 Úvod

Pro vývoj mikroprocesorové aplikace, je zapotřebí tří základních věcí.

- Programovaného procesoru (v tomto případě deska AVR Testboard 2.0)
- Programátoru (v tomto případě přímo na desce)
- Vývojového prostředí

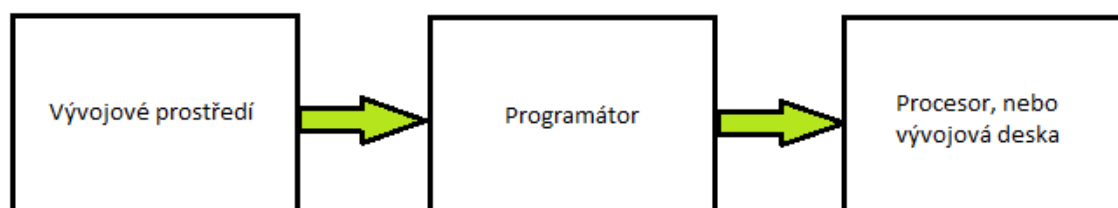
Pro jednoduchost programování je nejlepší použít vyšší programovací jazyk, v tomto případě například C. Programování v C je oproti assembleru velmi přehledné a pro začátečníky znalé alespoň základy jazyka C daleko záživnější než assembler.

2.2 Konkrétní vývoj aplikací za pomoci vývojové desky

Pro vývoj mikroprocesorové aplikace je zapotřebí nejprve konkrétní aplikaci vymyslet určité specifikace a připojit k procesoru dané periferie. U desky AVR Testboard je většina běžných periférií přímo na desce a další periferie je snadné připojit, protože všechny brány procesoru jsou vyvedeny ke konektorům.

Po rozmyšlení konkrétních součástí, které budou použity, se deska správně propojí a nastaví (nejlépe podle výukového programu, který je přiložen v příloze) a může se začít s konkrétním programováním.

Ve výukovém programu byl použit program CodeVisionAVR, který se svým průvodcem pro počáteční nastavení procesoru je vhodný, pro všechny začátečníky, ale dobře poslouží i pokročilým programátorů. Po vytvoření projektu se musí program přeložit do strojového kódu a ten se později nahraje do procesoru pomocí programátoru. Před použitím vývojových prostředí je potřebné vždy program nejprve nastavit, aby později nevznikaly chyby, například nastavit správný port pro programátor, správný protokol programátoru, dále například komunikaci po konzoly nebo použitý typ procesoru.

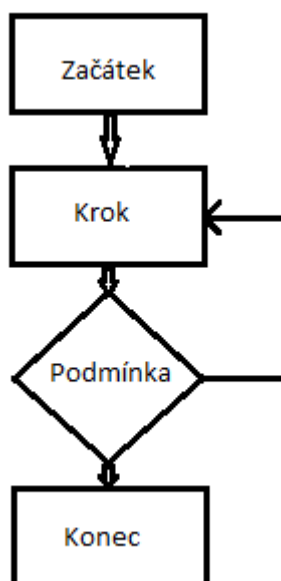


Obrázek 2.1: Cesta programu

Při programování se často dopouští chyb. Jelikož překladač do strojového kódu není nikdy dokonalý, při příliš složitém a špatně napsaném kódu ho špatně přeloží a tím se nevyužije procesor na maximum výkonu. Proto se často do kódu v C vepisují části kódu v assembleru, kterými se program celkově zrychlí a nezatěžuje tolik procesor. Jinak je nejlepší psát co nejkratší kód, ale občas je kvůli tomu nepřehledný.

Při programování je lepší používat vývojové diagramy. Vývojové diagramy často ujasní celkovou funkci programu, jelikož by jí měli zpřehlednit. Pro jednoduché aplikace se často zdá, že vývojový diagram je zbytečný v některých případech to je i pravda, ale když je potřeba se zorientovat v složitém programu velmi pomůže.

Většina studentů si myslí, že vývojový program jediné sťažuje práci, proto není žádný ve výukovém programu použit.



Obrázek 2.3: Jednoduchý vývojový diagram

3 HW návrh AVR Testboardu 2.0

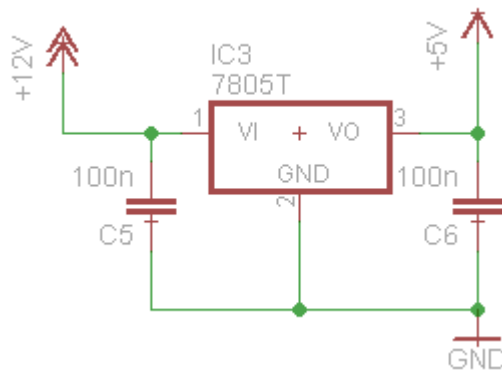
3.1 Úvod k návrhu

V zadání maturitní práce bylo vytvořit testovací desku pro procesory AVR. Zapojení desky vychází ze staré desky AVR Testboard, která byla prezentována minulý rok na školní a krajském kole SOČ a na STRETECHU na ČVUT. V zapojení je jeden zásadní rozdíl a to, že všechny komponenty včetně programátoru jsou na jedné desce oproti AVR Testboardu kde jsou na desce pomocí konektorů připojeny modul bluetooth pro bezdrátovou komunikaci po sériové lince a akcelerometr. Programátor byl jako externí zařízení připojené pomocí speciálního propojovacího kabelu.

3.2 Popis jednotlivých částí

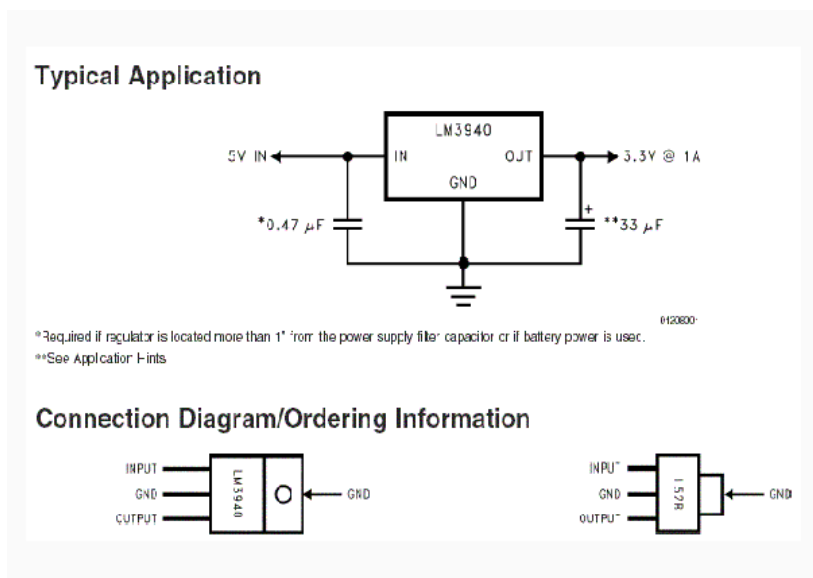
Napájení

Jelikož procesor má 5V logiku a většina periférií používá tuto logiku pro napájení je použit běžný stabilizátor 7805, oproti předešlé desce, kde bylo použito smd pouzdro D2PAK je použito pouzdro TO220 ve kterém tento stabilizátor je schopen dát proud až 1A a dá se lépe uchládit, protože termální můstky a měděná plocha nedostačovala chlazení u SMD verze.



Obrázek 3.1: Stabilizátor 7805

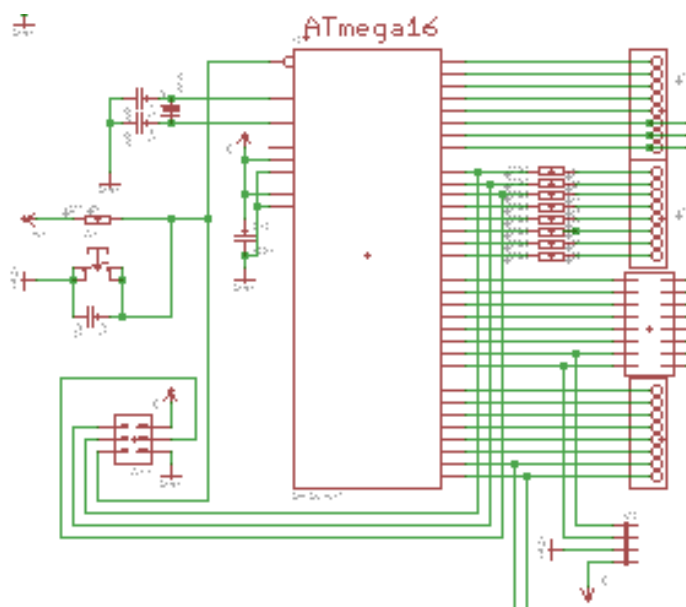
Na desce jsou ale i periferie, které mají 3,3 voltovou logiku a potřebují tím dané napájení. Proto je na desce ještě nízkoztrátový regulátor LM3940, který z 5V z předchozího stabilizátoru vytvoří 3,3V. Stabilizátor slouží primárně k napájení bezdrátového bluetooth modulu a akcelerometru, ale je možné ho použít i pro napájení dalších periférií s daným napájecím napětím.



Obrázek 3.2: Regulátor LM3940 (převzato z [5]).

Hlavní procesor ATmega16

Procesor ATmega16 je zapojen ve standardizovaném zapojení vycházející z datasheetu. V zapojení je pro testovací a vývojové účely i tlačítko pro rychlý reset procesoru a je použit externí krystal o frekvenci 16Mhz. Pro interní převodník je jako referenční napětí použito napájecí napětí. Všechny brány procesoru jsou vyvedeny, aby byli lehce přístupné pro připojení dalších periférií. Z brány B je vyveden programovací konektor, v atmelovém 6pinovém standartu. Za bránou jsou připojeny odrušovací odpory, které jsou zde z důvodu, kdyby na bráně byly připojeny po dobu nějaké periférie, aby neovlivňovali proces programování.



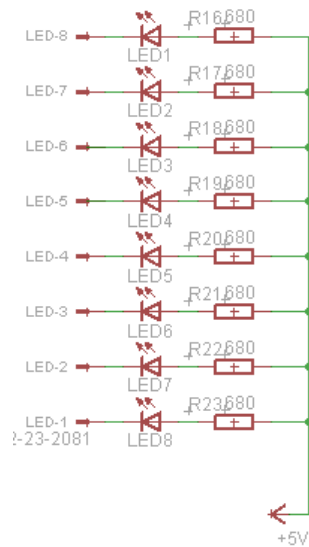
Obrázek 3.3: Zapojení procesoru ATmega16

LCD zobrazovač

V zapojení je použit LCD display se standartním řadičem HD44780. Display je zapojený podle knihovny v CodevisionAVR, aby hned po připojení použití této knihovny nebyl problém s funkčností, jako u minulé verze, kde se problémy vyskytovaly. V zapojení je použit display se 2 řádky o 16 znacích. Byl použit pro jednoduchost v ovládání a také proto, že pro komunikaci potřebujeme jen jednu bránu.

Modul Led diod

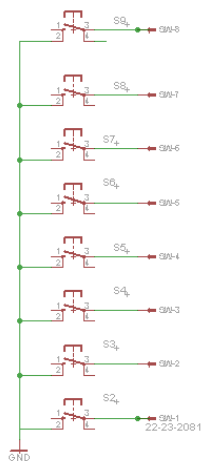
Na desce jsou 2 oddělené moduly. Jeden modul obsahuje 8 led diod, které jsou připojeny na napájení, dají se tedy ovládat logickou nulou procesoru. Propojit s procesorem je můžeme pomocí několika vodičů přímo na bránu procesoru. Počet 8 je dán tím, že brána má 8 výstupů můžeme tak na jednu bránu připojit jen diody, když při vývoji budeme potřebovat indikovat nějakou úroveň.



Obrázek 3.4: Modul led diod

Modul tlačítek

Modul mikrotlačítek je také jako modul led diod oddělenou částí obvodu, spojen je jen se společnou zemí. To znamená, že při stisknutí se daný vývod přizemní (vytvoří Log. 0) a procesor může generovat přerušení. Mikrotlačítka dobře poslouží při ovládání programu nebo v případě grafického menu na LCD. Modul tlačítek se zapojuje stejně jako modul led diod, pomocí vodičů přímo na bránu.

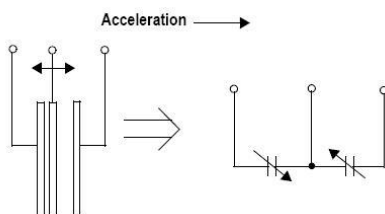


Obrázek 3.5: Modul tlačítek

Akcelerometr

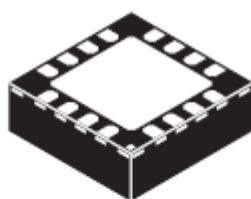
Pro další rozšíření desky byl na desku přidán takzvaný akcelerometr neboli čidlo gravitačního zrychlení. Pomocí tohoto zařízení můžeme měřit náklon desky a podle informací ovládat například modelářské servo. K obsluze akcelerometru je využit interní AD převodník, protože z akcelerometru jsou údaje analogové.

Akcelerometr je vyroben technologií MEMS která umožňuje přímo do pouzdra obvodu implementovat pohyblivé součásti. Tento akcelerometr funguje na principu pohyblivých dielektrik kondenzátorů. To znamená, že při náklonu se změní kapacita princip je vidět na obrázku 3.6



Obrázek 3.6: Princip akcelerometru

V obvodu je použit akcelerometr MMA7260Q od Freescale. Jak už bylo patrné z předchozího odstavce akcelerometr má analogový výstup. Hlavním rysem tohoto akcelerometru je schopnost měřit zrychlení ve všech třech osách (XYZ). Dokáže měřit statické i dynamické zrychlení, neboli statické zrychlení je všudypřítomné tíhové zrychlení (v klidu je ve vodorovné poloze naměřeno 1g v ose Z) a dynamické zrychlení je způsobeno změnou rychlosti pohybu. Výstupní signály všech tří os jsou analogové, proto je pro přesné měření nutné použít AD převodník. Výhodou tohoto akcelerometru je možnost přepínání rozsahů (citlivosti), které jsou čtyři; 1,5g (800mV/g), 2g (600mV/g), 4g (300mV/g), 6g (200mV/g) a dají se na desce pomocí jumperů nastavovat. Dalšími vlastnostmi je nízká spotřeba 500uA, při aktivním Sleep Mode 3uA, napájecí napětí v rozsahu 2,2 – 3,6V. Obvod je bohužel v pouzdru QFN, které má vývody zespod pouzdra a tak je obtížné ho připájet.



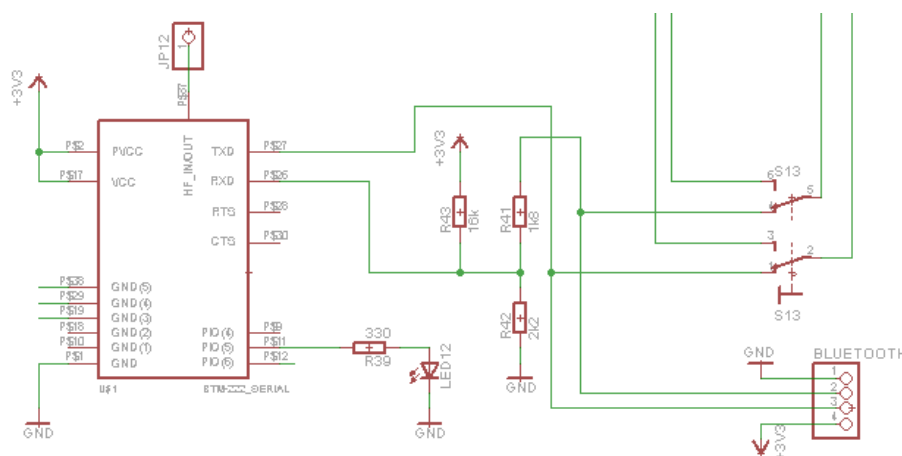
**16 LEAD
QFN
CASE 1622-01**

Obrázek 3.7: Pouzdro QFN

Bluetooth modul

Pro komunikaci s procesorem je možné volit mezi USB komunikací a komunikací přes bezdrátový Bluetooth modul. Počítač, nebo jiné zařízení komunikuje s procesorem nejjednodušeji po sériové lince, tento modul vytvoří virtuální sériovou bezdrátovou linku. Je to jednoduchý a velmi variabilní způsob komunikace.

Na trhu je nabízeno několik typů modulů pro bluetooth komunikaci pro desku byl vybrán modul od firmy Rayson přesněji typ BTM-222. Je to další periférie, která má napájecí napětí 3,3V a tím pádem funguje ve 3V logice. Pro použití s procesorem, který má 5V logiku je signál omezen jednoduše odporovým děličem pro výstupní úroveň z procesoru a pro vstupní úroveň do procesoru je velikost kterou vysílá modul dostačující. Práce s modulem je velmi snadná jednoduše se k němu dá připojit s počítačem, nebo mobilním telefonem a s jednoduchým obslužným softwarem posílat data. Pro spárování je od výroby přednastaveno heslo, které s dalšími nastaveními modulu je možno změnit pomocí vyvedené sériové linky.

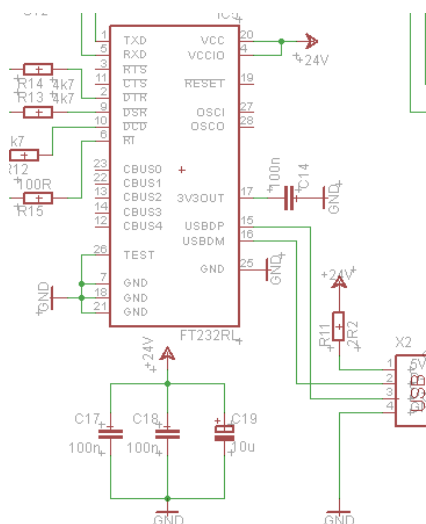


Obrázek 3.8: Zapojení Bluetooth modulu

Programátor

Pro programování procesoru byl vybrán osvědčený programátor, který je sice relativně složitý, ale ukázal se jako velmi spolehlivý. Jde o zapojení AVRUSB500v2, které bylo publikováno na serveru <http://tuxgraphics.org> a ze serveru pochází i použitý firmware v procesoru.

Jelikož jde o celkem složité zapojení, bude vysvětleno ve 2 částech. Protože nové počítače a především notebooky nemají vyveden sériový nebo paralelní port, je potřeba komunikovat přes nejrozšířenější sběrnici USB. Proto je použit dnes již běžný převodník od firmy FTDI FT232RL. Tento převodník vytvoří v počítači virtuální sériovou linku, tento obvod je celkem drahý, ale pro svoji funkčnost potřebuje minimum dalších součástek. Obvod a vlastně celý programátor je napájen přes USB sběrnici. Převodník je použit jak pro programátor tak se dá použít jako převodník na USB sériovou linku pro samotný procesor. Na obrázku 3.9 je ukázané zapojení převodníku.



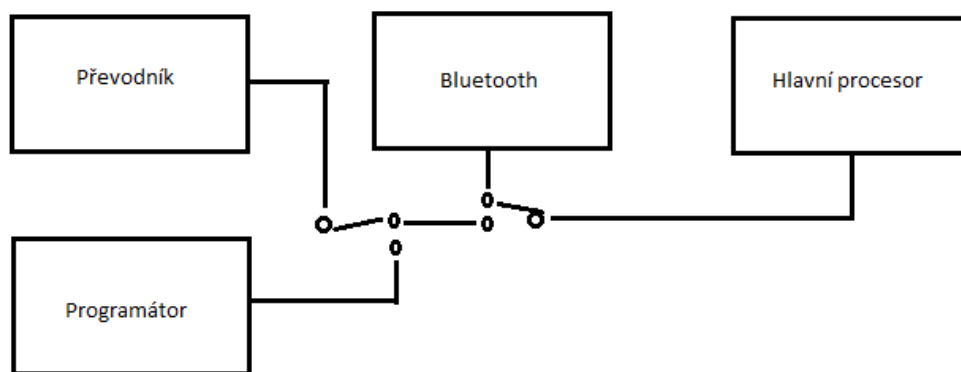
Obrázek 3.9: Zapojení převodníku FR232RL

V druhé části programátoru se nachází také procesor z rodiny ATmega konkrétně ATmega8. Do procesoru je nahraný firmware, který za pomoci převedené sériové linky vytvoří konkrétní signály pro procesor. Po sériové lince posíláme signály standartu STK500 (programátor pro sériovou linku přímo od firmy Atmel), ale jelikož převodník je nedokáže přesně převést, protože oproti normální sériové lince postrádá několik pinů, musí tuto práci vykonat procesor s daným firmwarem. Když připojíme k běžnému USB RS232 převodníku programátor pro sériovou linku, nebude pracovat právě z důvodu chybějících pinů, tedy mimo speciálního módu převodníku.

Tento programátor má ještě jednu výhodu, jelikož pro jeho zprovoznění není potřeba další programátor. Protože ve zvláštním módu převodníku lze do programátoru nahrát firmware. Tento způsob nahrávání je velmi pomalý a ne na všech počítačích se z důvodu nastavení sériové linky povede firmware správně nahrát.

Přepínače

V zapojení jsou ještě použity 2 páčkové přepínače. Přepínače slouží pro přepínání sériových linek procesoru programátoru Bluetooth modulu a převodníku. Jde vlastně o to, že lze pomocí kombinace přepínačů zvolit celkem 2 zdroje sériové linky a to z USB převodníku a z Bluetooth modulu. Jelikož převodník v programátoru potřebujeme zároveň pro programování a jako sériovou linku pro hlavní procesor je potřeba přepínat mezi těmito dvěma stavy. Zapojení je patrné z blokového schématu na obrázku 3.10.



Obrázek 3.10: Blokové schéma přepínačů

3.3 Návrh plošného spoje

Po návrhu předchozí verze desky bylo vyřešeno několik důležitých problémů. Po dokončení první desky se vyskytlo několik chyb při praktickém používání. Jelikož některé typy součástek jako například stabilizátor, který se trochu přehříval. Taky bylo potřeba změnit umístění některých konektorů, protože byli špatně přístupné nebo na jiných místech než byli potřeba. Upravil se i konektor pro display, který měl trochu jiné rozmístění vývodů, než bylo v knihovně potřeba.

První verze desky byla vyrobena na jednostranném amatérském spoji o velikosti zhruba 11x16 centimetrů. Deska byla osazena většinou SMD součástkami ve větších pouzdrech. Navrzení a vytvoření této desky si při ožívání vyžádalo tedy několik úprav v zapojení, jinak deska celkově je plně funkční a pracuje velmi spolehlivě. Nevýhodou desky byl její nehezky vzhled, daný neprofesionální výrobou, hlavně z hlediska popisků na vrchní straně, které jsou pro správné používání nezbytné.

Na poslední chvíli bylo rozhodnuto o výrobě nové oproti staré již profesionálně vyrobené desce. Starý plošný spoj tedy nevyhovoval, a proto byl postupně navržen nový plošný spoj s úpravami a modulem programátoru přímo na jedné desce. Tento návrh byl oproti první verzi složitější z důvodu zpracování dat pro profesionální výrobu. Profesionálně vyrobené plošné spoje jsou navrhovány jiným stylem, a musí se dodržovat spoustu věcí, na které se normálně nehledí.

Oproti předešlému spoji muselo být navrženo vlastně 5 samostatných vrstev.

- Vrchní vrstva
- Vrchní nepájjivá maska
- Potisk vrchní strany
- Spodní vrstva
- Spodní nepájjivá maska

Každá vrstva má svá specifika v návrhu podle druhu výroby a výrobce.

Z důvodu obvodové složitosti byl navržen oboustranný plošný spoj s prokovy. Návrh takového spoje je odlišný od školních návrhů a realizací. Výroba těchto druhů plošných spojů není ve školních podmínkách možná, a proto byla zadána specializované firmě. To ale neslo spoustu dalších problémů s návrhem.

Vrchní vrstva

Na vrchní vrstvě je netradičně několik SMD součástek, na desce jsou vlastně SMD součástky z obou osazovaných stran. Na vrchní straně jsou jen specifické součástky zde umístěny z daných důvodů. Na vrchní stranu jsou umístěny led diody z jednoduchého důvodu, aby byli vidět. SMD led diody byly použity, protože vývodové diody na předešlé desce se ulamovaly, použitím SMD malých diod se této neřesti zabrání. Dále byl na vrchní stranu umístěn bluetooth modul, protože to je redukce z dalšího plošného spoje a je o trochu vyšší než obyčejné SMD součástky, proto byl umístěn pod LCD display kde by neměl ničemu překážet. Dále je na vrchní straně umístěn akcelerometr, hlavně pro názornou ukázkou. Celou vrchní vrstvu proplétá řada spojů, místo kterých by musely být jinak propojky.

Nepájivá maska

Na každé profesionální desce, je obyčejně zelená nepájivá maska, která chrání spoje a ulehčuje pájení. Deska také lépe vypadá, je přehlednější a dá se na ní nanést další vrstva s potiskem. Při vytváření nepájivé masky se definují u všech plošek pájecí body, kde se maska nebude nacházet. V programu Eagle se maska generuje automaticky, dá se přenastavit jen odestup od plošky.

Potisk

Hlavním důvodem, proč byla vytvořena profesionálně vyrobená deska byly hlavně popisky. Z vývojové desky jsou popisky celkem zásadní částí desky, jelikož je potřebné stále něco přenastavovat a upravovat. Proto byly v návrhu vytvořeny přesné popisky všech portů a nastavovacích přepínačů.

Spodní strana

Na spodní a také hlavní straně jsou umístěny všechny zbylé součástky. Většina součástek na spodní straně jsou pasivní součástky. Všechny rezistory a blokovací kondenzátory jsou na spodní straně umístěné co nejbližší vývodům daných obvodů. Na spodní straně je ještě převodník z USB na RS232, ten je vyráběn v SMD pouzdře s vývody velmi blízko u sebe, proto je obtížné ho správně připájet.

Na plošném spoji je většina součástek v SMD pouzdrech, kvůli celkovému zmenšení celé desky. Všechny pasivní součástky mimo jednoho elektrolytického kondenzátoru jsou v SMD pouzdrech. Při návrhu je třeba dbát na rozmístění součástek a délku spojů.

Celý návrh proběhl v programu Eagle. Jelikož důležité součástky do zapojení neměli v programu vytvořené svoje knihovny, bylo zapotřebí pro tyto součástky do programu vytvořit nové knihovny. Jelikož většina knihoven neodpovídala

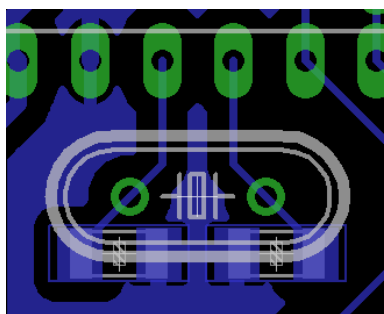
parametrům pro výrobu, museli se součástky upravit. Úprava součástek spočívala hlavně z hlediska velikosti vrtaných děr a pájecích plošek. Pro výrobu se musí vygenerovat také tzv. vrtací data, ty obsahují souřadnice a velikosti vrtáků, kterými se deska má vrtat. Po prvním vygenerování tohoto seznamu musela být předělána velká část knihovny z důvodu velkého počtu nástrojů s jen malými odchylkami. Při výrobě prokovů musela být také zvolena určitá vypočítaná ploška a díra.

Při výrobě prokovů se na desce z obou stran vyleptá plošný spoj a vyvrtají se díry. Jelikož se prokovy vytváří pomocí galvanického pokovování, musí se tedy vnitřní stěny děr zvodivit, aby se tedy mohli pokovit. Při pokovování se na desku na měděnou folii a do pokovovaných děr nanese grafit, který jí zvodiví a může se tedy pokovit. Po pokovení se vlastně zesílí měděná folie a prokoví díry. Jelikož se prokoví všechny díry musí být tedy 2 soubory s dírami, které mají a nemají být prokoveny. Toto je jeden z nejpoužívanějších způsobů výroby prokovů.

Z tohoto postupu je dáno, že díry pro prokovy se po pokovení zúží o určitou sílu měděné vrstvy. Proto si výrobce musí upravit velikosti děr podle toho, jestli jsou nebo nejsou prokovené a u prokovených přidat o sílu nanesené mědi. Protože, údaje o velikosti děr se většinou udávají výsledné po všech úpravách.

Při návrhu je také potřeba dbát na některé důležité detaily. Při umísťování blokovacích kondenzátorů je důležité je umísťovat co nejbližší k vývodům integrovaných obvodů, aby správně plnili svoji funkci.

U krystalů je důležité mít stejnou délku spoje od vývodu procesoru k oběma vývodům krystalu. A z důvodu rušení nesmí být pod krystalem vedený jiný například signálový spoj, aby ho kmitání krystalu nerušilo.



Obrázek 3.11: Délka spojů u krystalů

Pro plošný spoj je také dobré vytvořit rozlitou měděnou plochu spojenou se zemí, protože, se dá použít k několika účelům.

- Jako stínění proti dalším vlivům
- Pro chlazení součástek pomocí termálních můstků
- Vyrovnává povrch a vytváří tak rovnější plochu pro vrstvu popisků

Na tomto plošném spoji je tato plocha z vrchní i spodní strany.

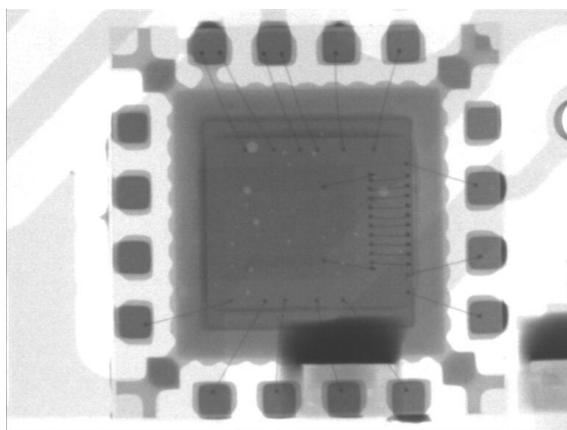
3.4 Osazování

Plošný spoj je převážně navržen, pro součástky pro povrchovou montáž. Některé tyto součástky není možné v domácích ani ve školních podmínkách osazovat, proto jsem se zkontaktoval s firmou Sitronics, u které jsem byl na praxi a na brigádě. Zde mají pro tuto montáž špičkové vybavení.

Při osazování desky jsou tři nejkritičtější součástky:

- Akcelerometr (v QFN pouzdru)
- FT232RL (Pouzdro SSOP)
- Bluetooth modul

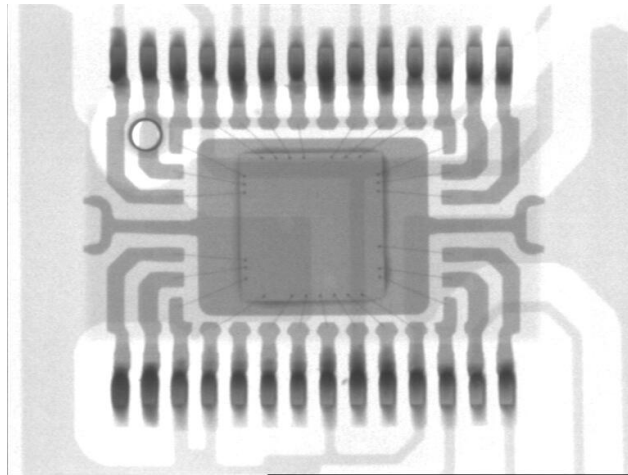
Při osazování akcelerometru je několik způsobů jak ho osadit. Byl zvolen způsob osazení nanesení pájecí pasty (pod lupou na jednotlivé body) poté položení akcelerometru a pomocí speciální topné infra lampy prohřátí součástky a pasty, která vytvořila pevné spojení. Ve firmě je i speciální rentgen pro kontrolu takovýchto součástek, které mají vývody zesponu, proto byl i tento akcelerometr zvlášť zkontrolován.



Obrázek 3.12: Rentgenový snímek akcelerometru

Na obrázku 3.12 je vidět zapájený akcelerometr tmavé plošky přesahující plošky akcelerometru je rozlity cín. Na obrázku je vidět, že mezi ploškami není žádný zkrat.

Další kritickou součástkou je převodník z USB na sériovou linku. Tento převodník se vyrábí v pouzdru SSOP, to je SMD pouzdro s velmi jemnou roztečí vývodu (0,68mm). Z důvodu kvalitního plošného spoje s pozlacenými ploškami nebyl pod lupou sebemenší problém ho připájet.



Obrázek 3.13: Rentgenový snímek FT232RL

Na obrázku 3.13 je vidět rentgenový snímek převodníku. Zajímavé je na něm například, jak jsou vedeny spoje od křemíkového jádra ven k vlastním vývodům. Z obrázku je patrné, že připájený je naprosto dokonale.

Dále ještě tedy kritický bluetooth u kterého není problém s připájením, ale spíše s odpájením. Jelikož v zakoupeném bluetooth modulu byla nejspíš vada, protože správně neplnil svojí funkci, tak bylo potřeba ho vyměnit. Pro výměnu byl použit modul z předešlé verze desky, který plnil svou funkci bezchybně. Z důvodu jiného druhu spojení nešel odsát cín, a proto musel být nahřán celý modul, poté sejmout z desky a deska musela zůstat nepoškozená.

Osazování ostatních součástek už bylo jednoduché, ale velice zdlouhavé, protože musely být perfektně zapájeny.

Oživení desky proběhlo bez větších problémů, tedy mimo vadného bluetooth modulu. Nejprve bylo potřeba proměřit napětí do vstupů a z výstupů stabilizátorů a napětí na patičkách k procesorům a k LCD. Po naměření správných hodnot bylo jasné, že plošný spoj byl bezchybně navržen a vyroben. Následovalo nasazení zbývajících komponentů a obou procesorů do patič.

4 Výukový program pro desku

V zadání maturitní práce bylo vypracovat výukový program. Vypracování tohoto programu bylo potřebné pečlivě promyslet, protože bylo důležité, aby byl lehce pochopitelný pro studenty a zároveň vytvořený zábavnou formou. Protože, pro mnohé studenty je programování mikroprocesorů nezáživné a cílem programu bylo jednoduše a prakticky ukázat jak mikroprocesor ovládat.

5 Technické parametry

- Velikost desky 12x12cm
- Procesor ATmega16
- LCD 2x16 znaků s řadičem, podsvícený
- Analogový akcelerometr MMA7260Q
- 8 tlačítek
- 8 indikačních led diod
- Možnost připojení modelářských serv
- Bluetooth bezdrátový modul pro UART komunikaci
- USB programátor
- USB sériové rozhraní
- ISP programovací konektor
- Možnost pomocí programátoru programovat další procesory AVR
- Externí napájení
- Napájecí napětí 7-36V
 - 5V větev stabilizátor 7805
 - 3V větev regulátor LM3940
- Napájení pomocí USB
- Odebíraný proud okolo 120mA podle konfigurace

Závěr

Předložená práce představuje výslednou verzi návrhu funkční vývojové desky AVR Testboard 2.0. V práci je popsána celková architektura procesorů AVR, která má svá specifika. Poté je popsán celkový návrh desky, ve kterém je jednotlivý popis všech součástí a důvody, proč byli použity a jak jsou zapojeny, popis návrhu dvouvrstvých plošných spojů a popsány konkrétní důležité části při návrhu a výrobě plošných spojů. Popsán postup vývoje mikroprocesorové aplikace a nakonec popsán princip výukového programu.

Závěrem bych chtěl poděkovat svému vedoucímu práce Mgr. Milanovi Janouškovi, za konzultace a kontrolu maturitní práce, dále panu Ing. Pavlu Bořkovcovi za konzultace a kontrolu dat pro výrobu plošného spoje ve specializované firmě.

Seznam citací

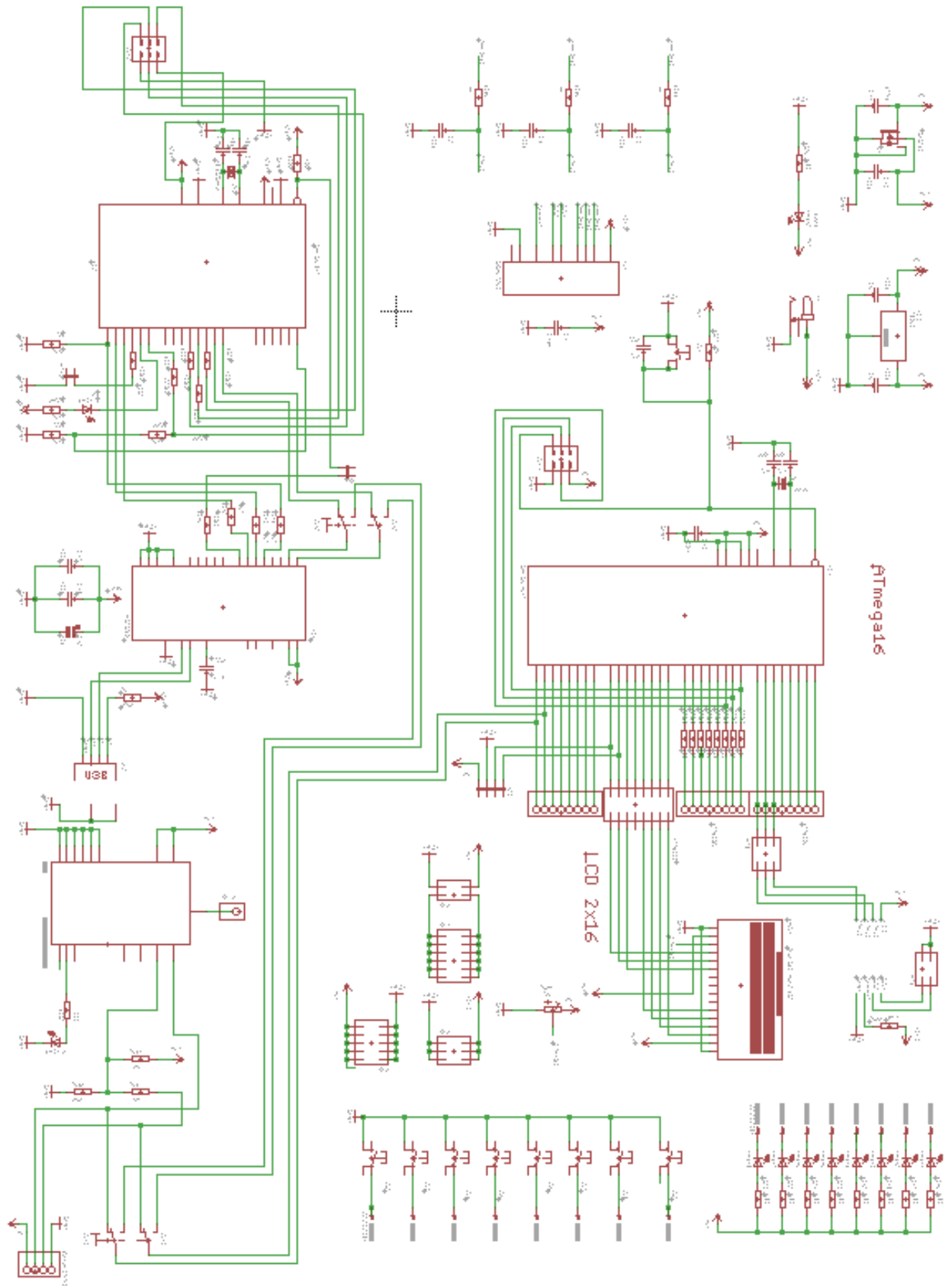
- [1] VÁŇA, Vladimír . *Mikrokontroléry Atmel AVR - programování v jazyce C*. Praha : BEN, 2003. 216 s. ISBN 80-7300-102-0.
- [2] MATOUŠEK, David. *Práce s mikrokontroléry Atmel AVR ATmega16*. Praha : BEN, 2006. 320 s. ISBN 80-7300-174-8.
- [3] AVR. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 1. 3. 2007, 11:54, last modified on 28. 2. 2011, [cit. 2011-03-27]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/AVR>>.
- [4] *datasheet mikrokontroleru ATMEGA16*. Atmel, 344 stran. Dostupné z WWW [29.3.2011]: <http://www.atmel.com/dyn/resources/prod_documents/doc8154.pdf>.
- [5] *datasheet LM3940*. National Semiconductor, 12 stran. Dostupné z WWW [29.3.2011]: <<http://www.national.com/ds/LM/LM3940.pdf>>.

Seznam příloh

- Schéma zapojení
- Návrh plošného spoje
 - Vrchní vrstva
 - Spodní vrstva
 - Rozložení součástek na vrchní vrstvě
 - Rozložení součástek na spodní vrstvě
- Seznam součástek
- Fotografie výrobku
 - Foto neosazené desky
 - Foto osazené desky
 - Rentgenové snímky
- Výukový program
- Na přiloženém cd
 - Vygenerovaná Gerber data
 - Zdrojové kódy k výukovému programu

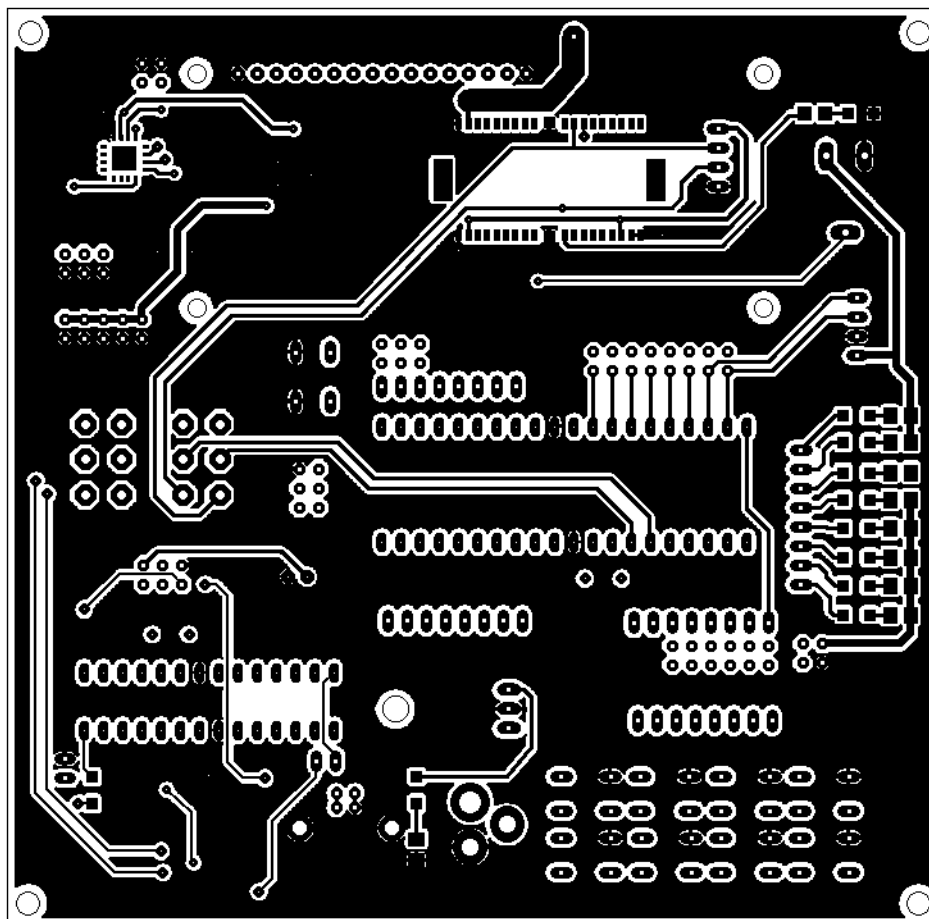
Přílohy

Schéma zapojení

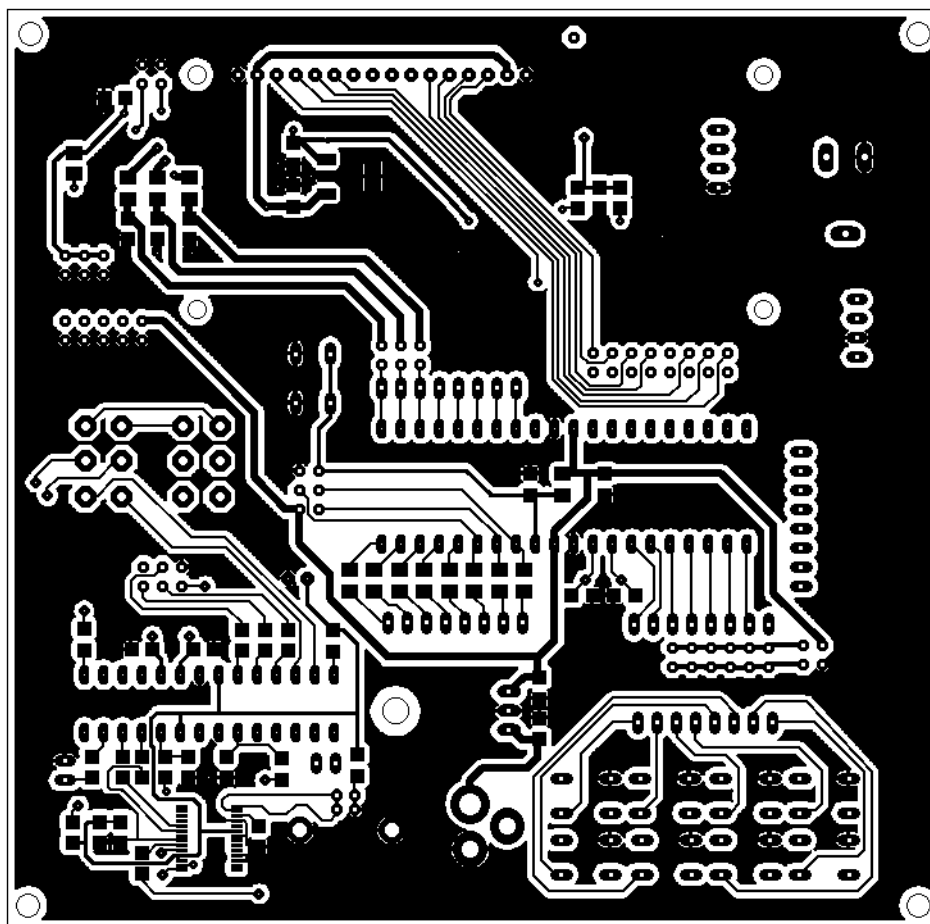


Návrh plošného spoje

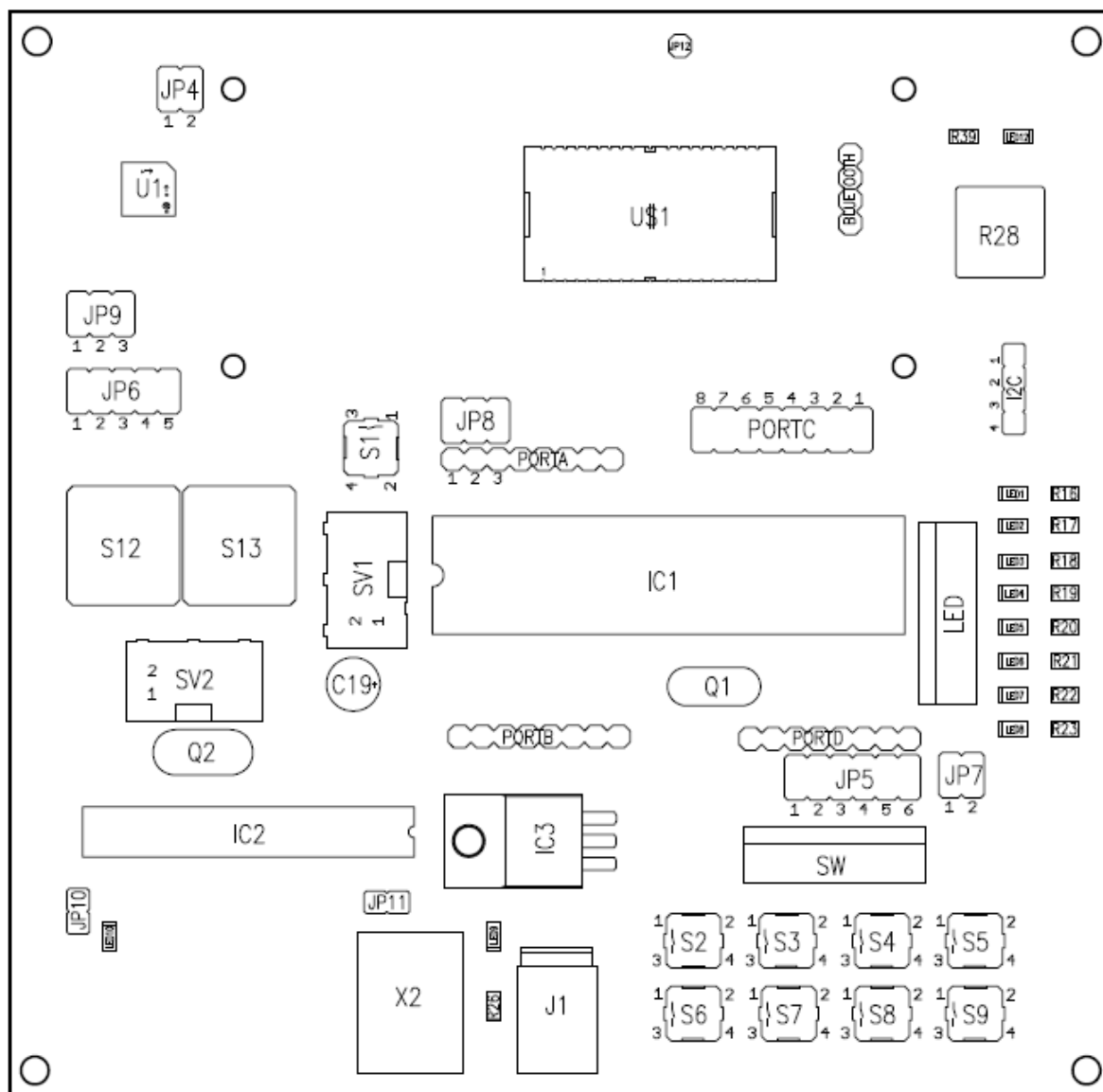
Vrchní vrstva



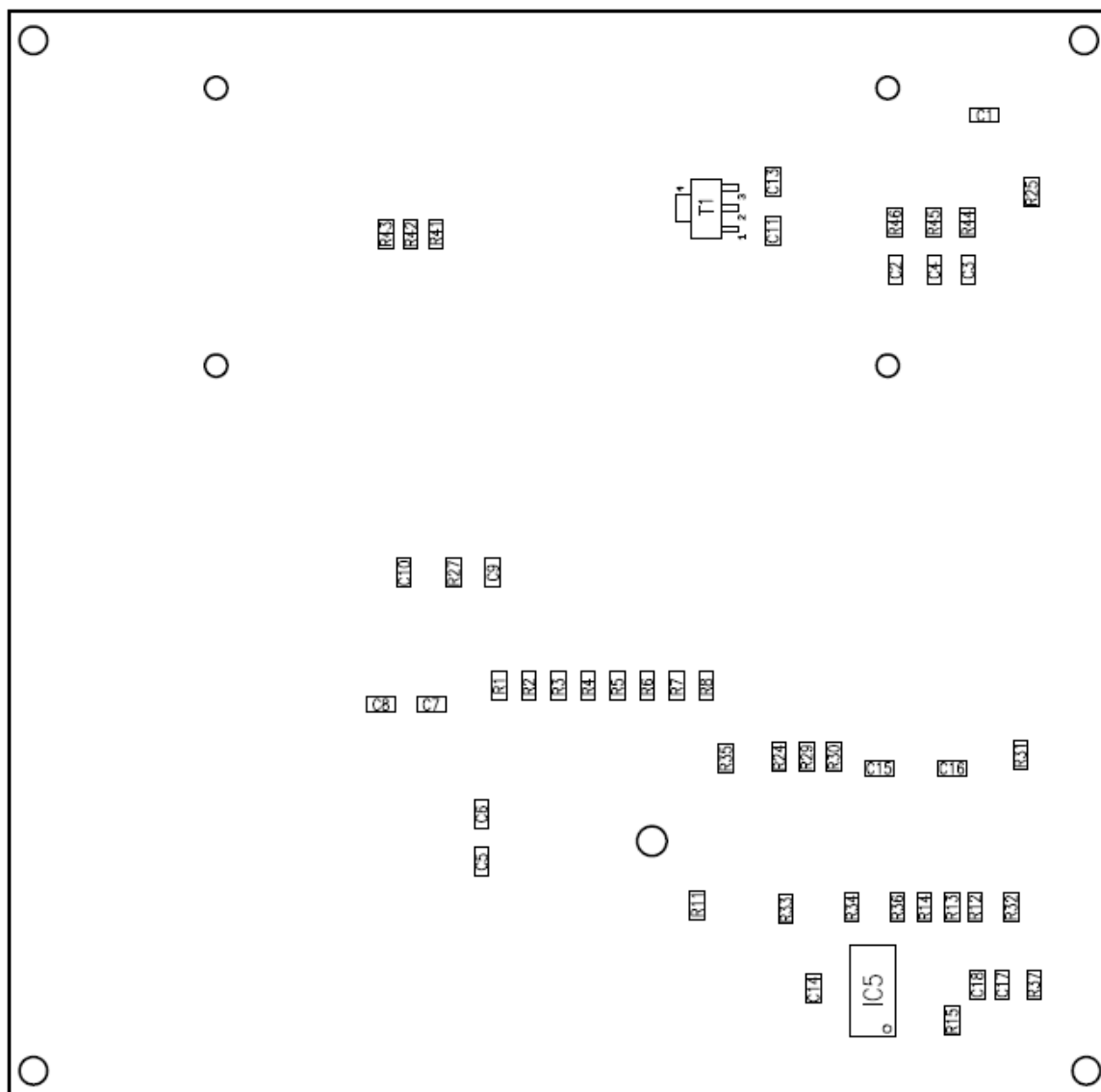
Spodní vrstva



Rozložení součástek vrchní strana



Rozložení součástek spodní strana



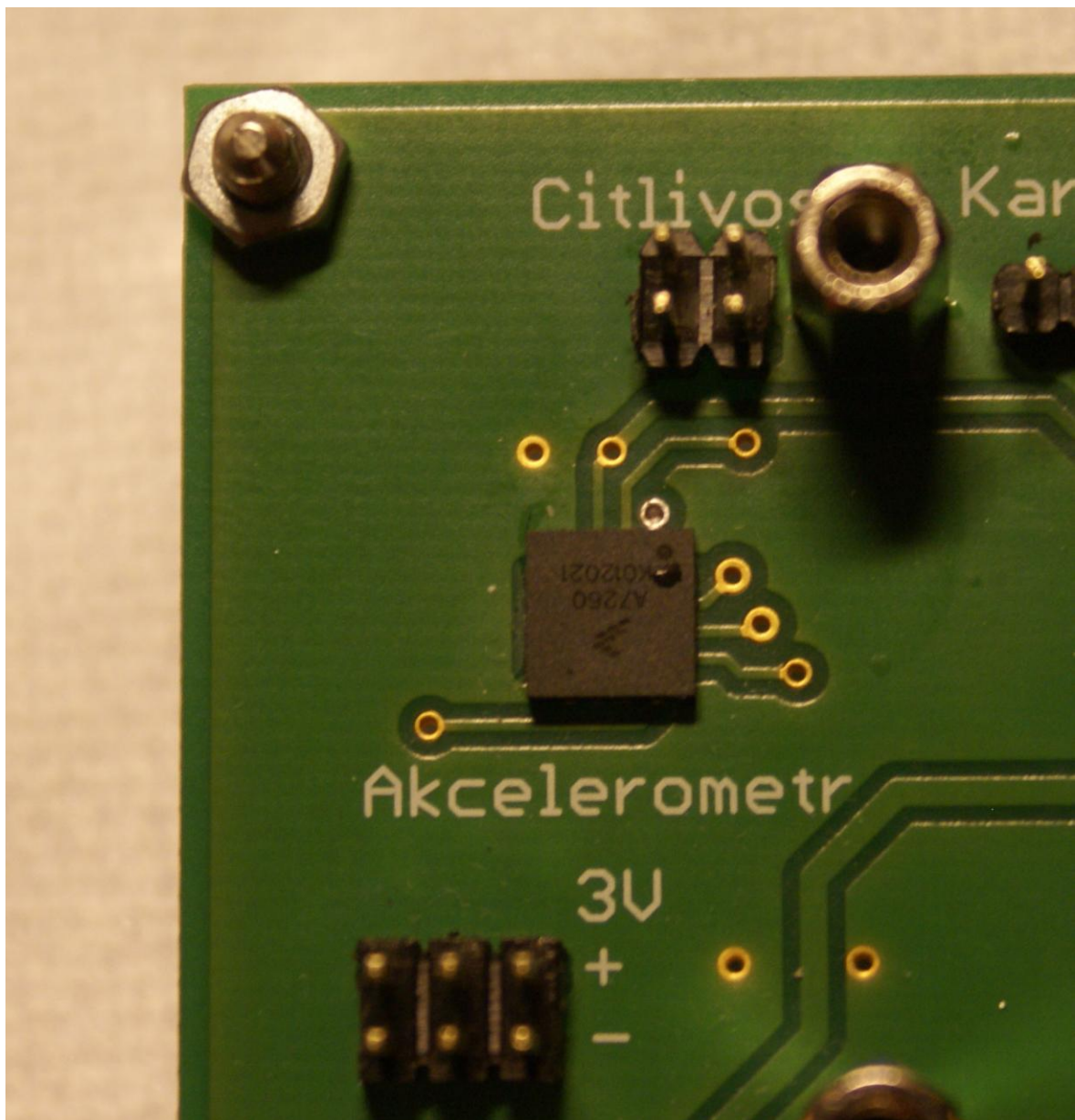
Seznam součástí

Položka	Počet	Označení	Hodnota
1	1	BLUETOOTH	4pin konektor
2	11	C1, C2, C3, C4, C5, C6, C9, C10, C14, C17, C18	100n
3	2	C7, C8	22p
4	1	C11	470n
5	1	C13	100n
6	2	C15, C16	22p
7	1	C19	10u
8	1	DIS1	LCD 16x2 znaky
9	1	I2C	4pin konektor
10	1	IC1	ATmega16
11	1	IC2	ATmega8
12	1	IC3	LM7805
13	1	IC5	FT232RL
14	1	J1	Power jack konektor
15	2	JP4, JP5	2x2pin jumper
16	1	JP6	2x5pin jumper
17	1	JP7	2x6pin jumper
18	2	JP8, JP9	2x3pin jumper
19	2	JP10, JP11	1x2pin jumper
20	1	JP12	1x1pin
21	2	LED, SW	8pin konektor
22	8	LED1, LED2, LED3, LED4, LED5, LED6, LED7, LED8	Modrá led
23	3	LED9, LED10, LED12	Zelená led
24	3	PORTA, PORTB, PORTD	8pin konektor
25	1	PORTC	2x8pin konektor
26	1	Q1	16Mhz krystal
27	1	Q3	18,432Mhz krystal
28	8	R1, R2, R3, R4, R5, R6, R7, R8	240R
29	1	R11	2R2
30	4	R12, R13, R14, R25	4K7
31	3	R15, R24, R32	100R
32	10	R16, R17, R18, R19, R20, R21, R22, R23, R26, R37	680R
33	4	R27, R44, R45, R46	1K
34	1	R28	Trimr 10K
35	2	R29, R31	220R
36	1	R33	220K
37	1	R34	100K
38	2	R35, R36	10K
39	1	R39	330R
40	1	R41	1K8
41	1	R42	2K2
42	1	R43	16K
43	9	S1, S2, S3, S4, S5, S6, S7, S8, S9	mikrotlačítka

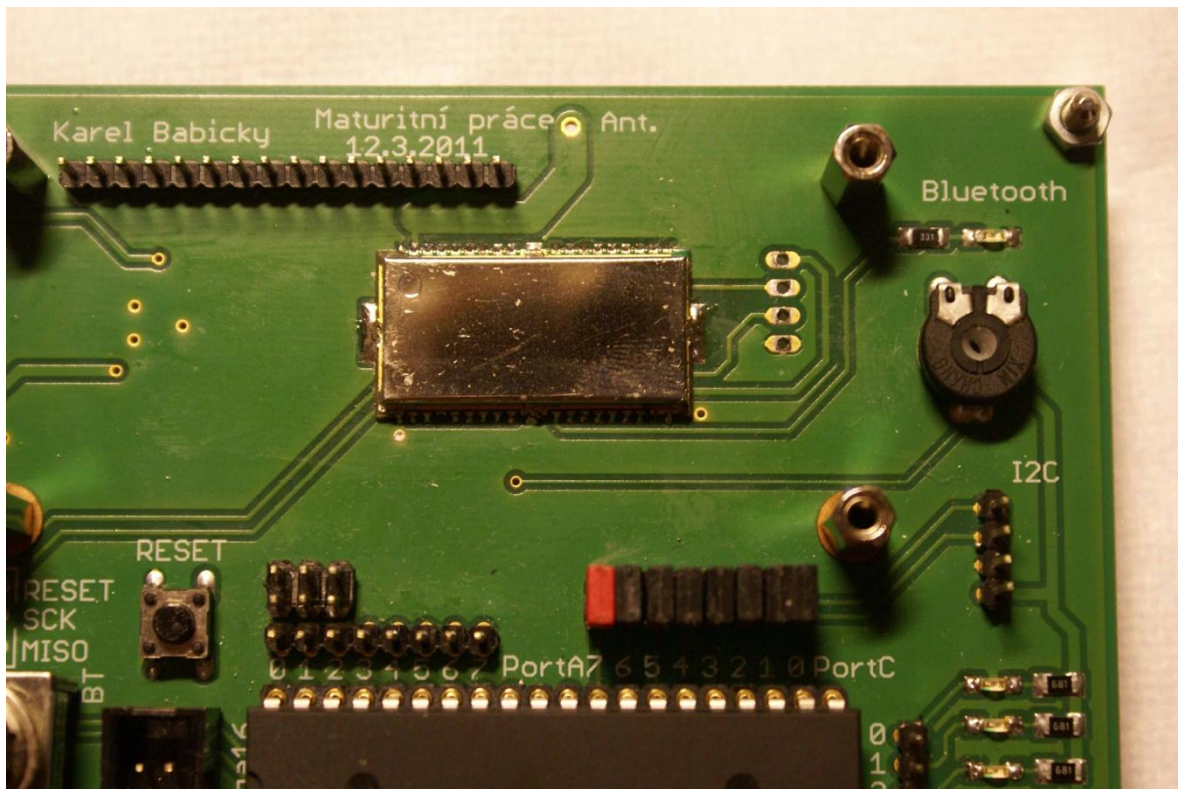
44	2	S12, S13	Páčkové spínače
45	2	SV1, SV2	Programovací konektor
46	1	T1	LM3940
47	1	U\$1	BTM-222
48	1	U1	MMA7260Q
49	1	X2	USB B konektor
50	1	PCB	TESTBOARD 2.0

Fotografie výrobku

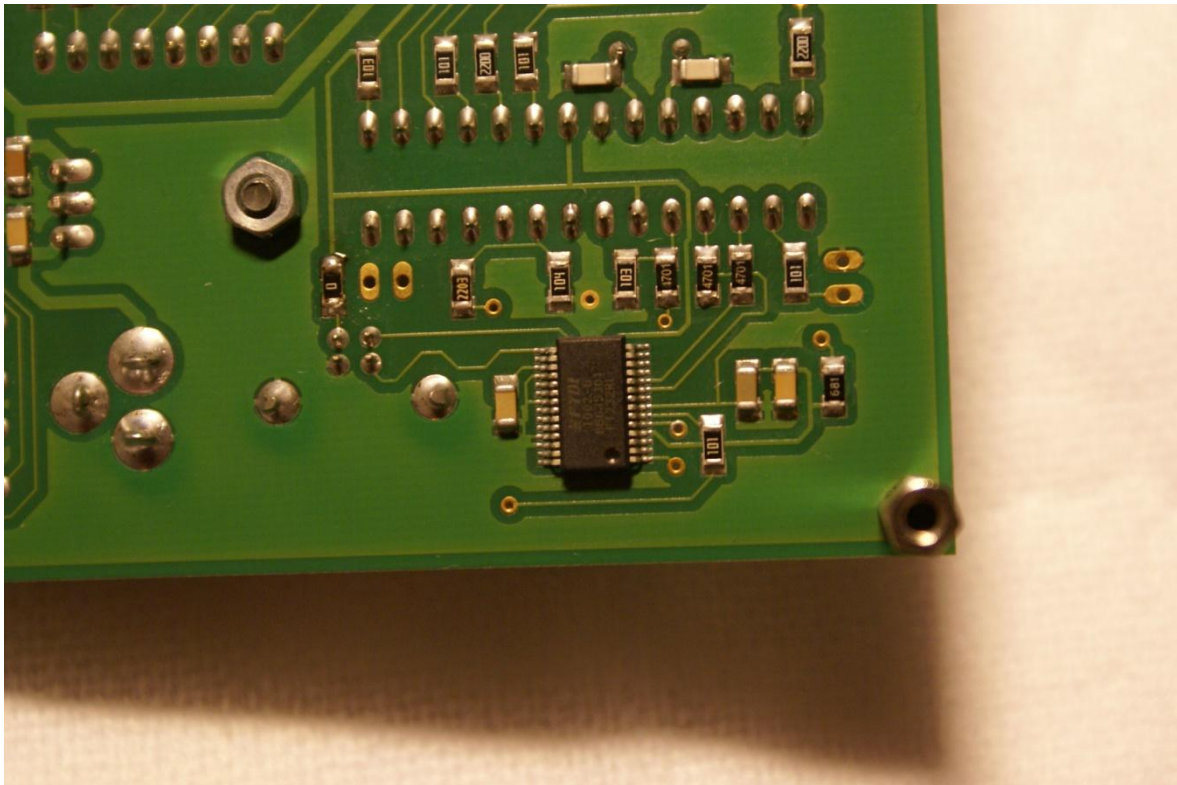
Akcelerometr



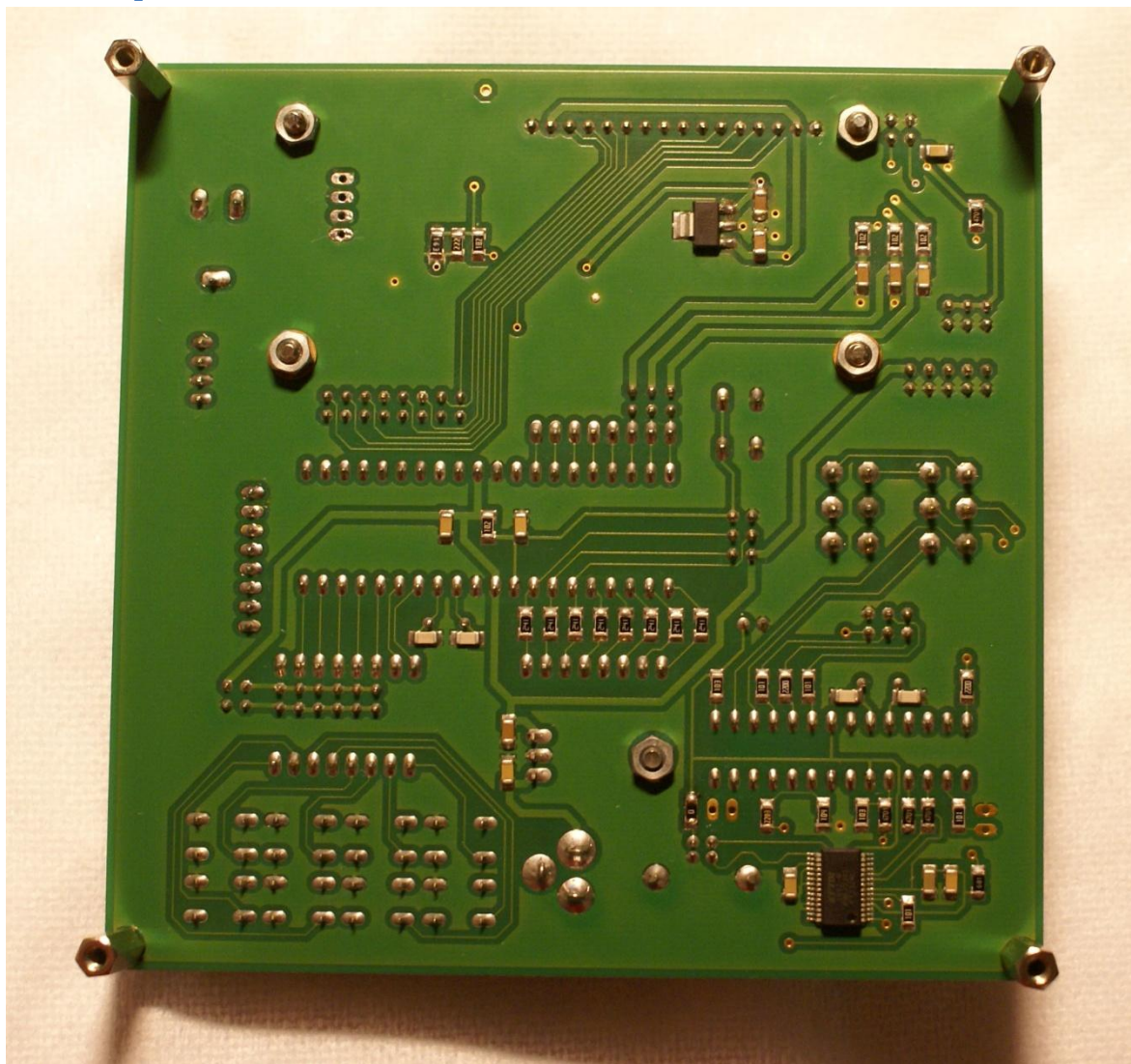
BT modul



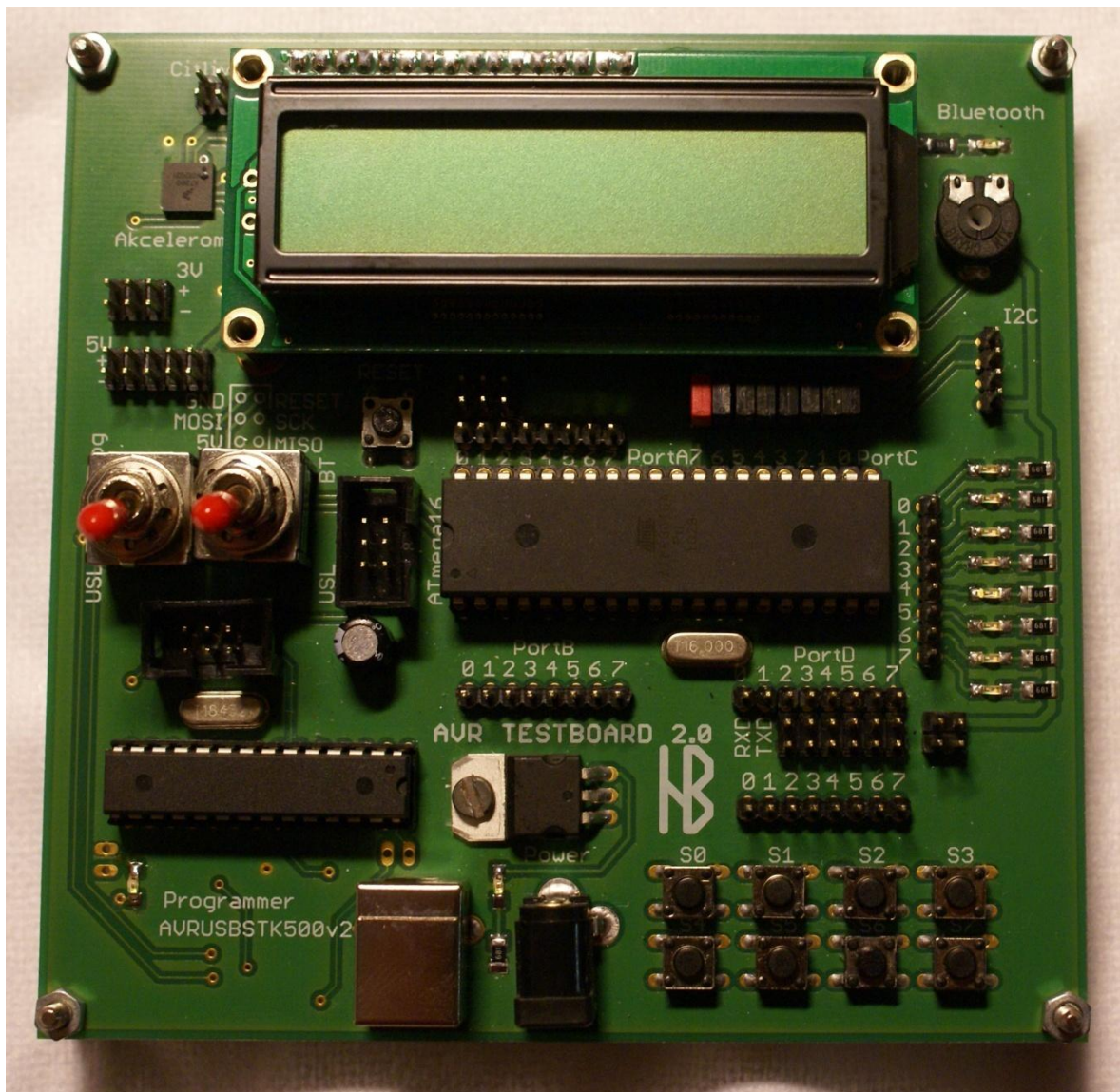
Převodník FT232RL



Pohled spodní strana

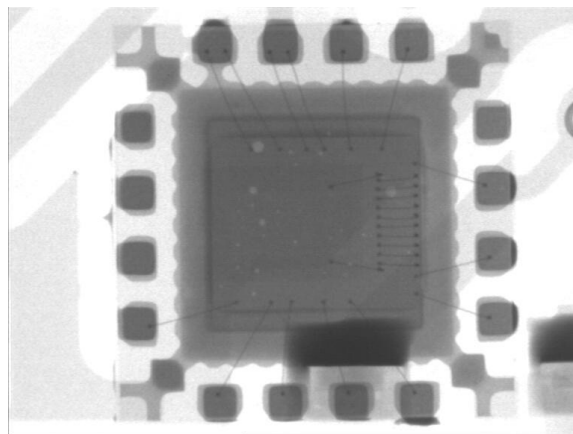


Pohled vrchní strana

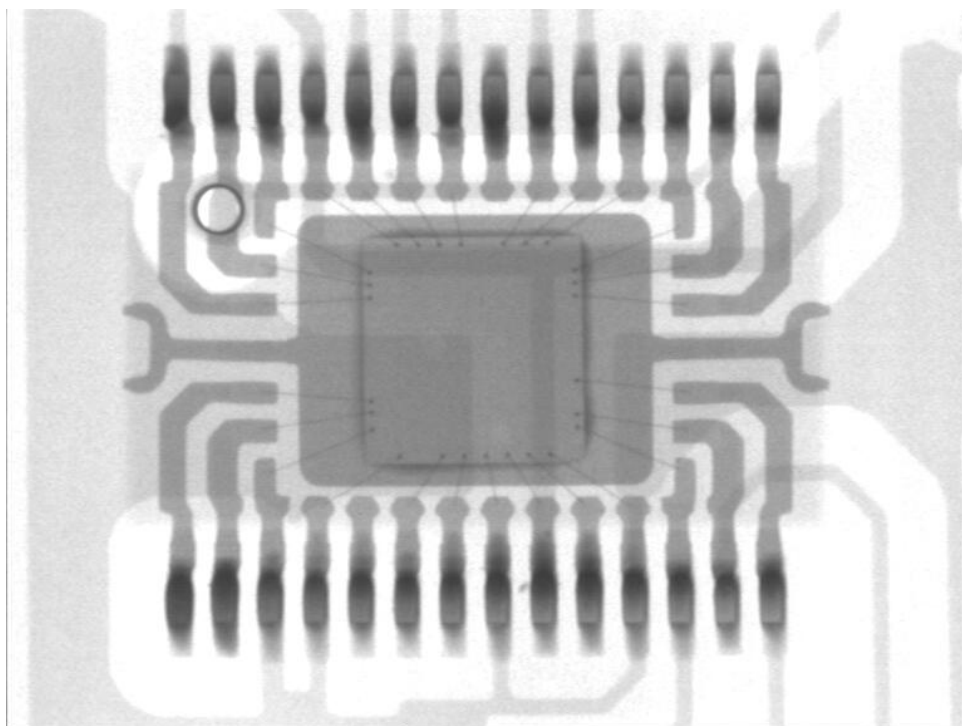


Rentgenové snímky

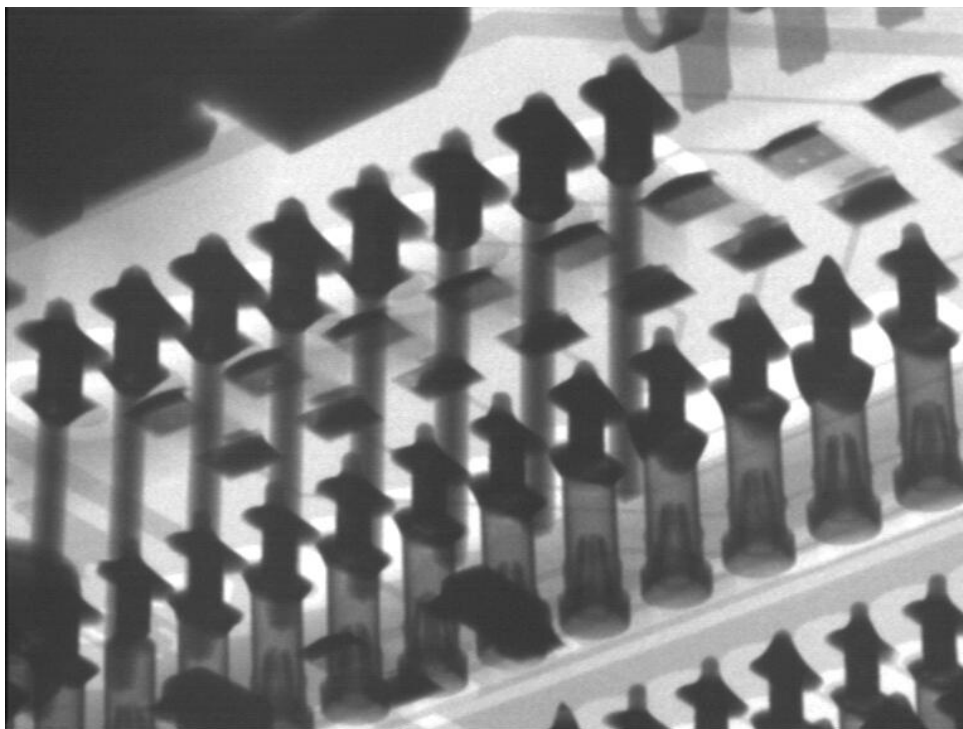
Akcelerometr



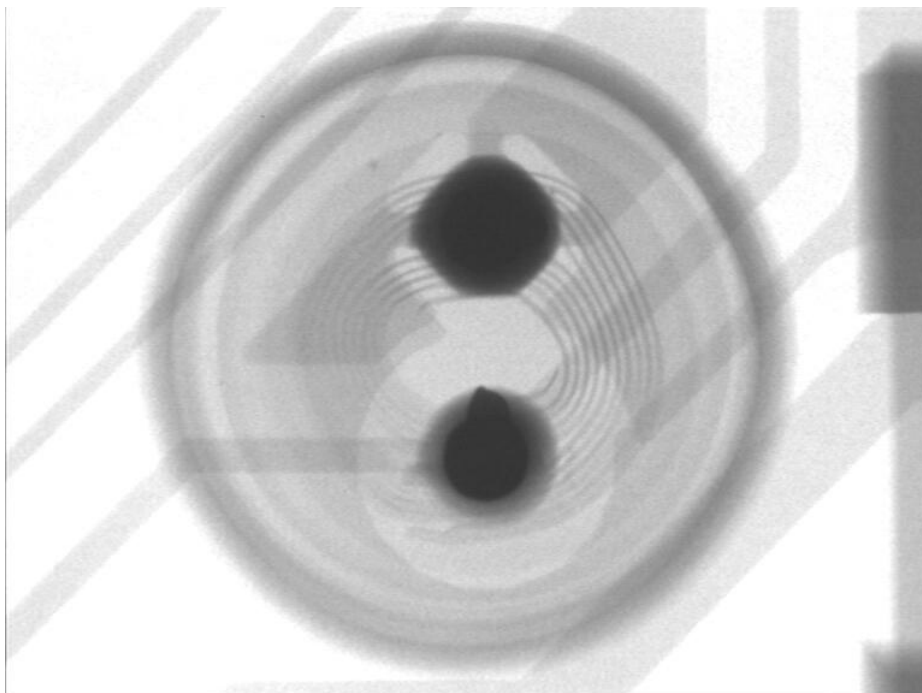
Převodník FT232RL



Prokovy

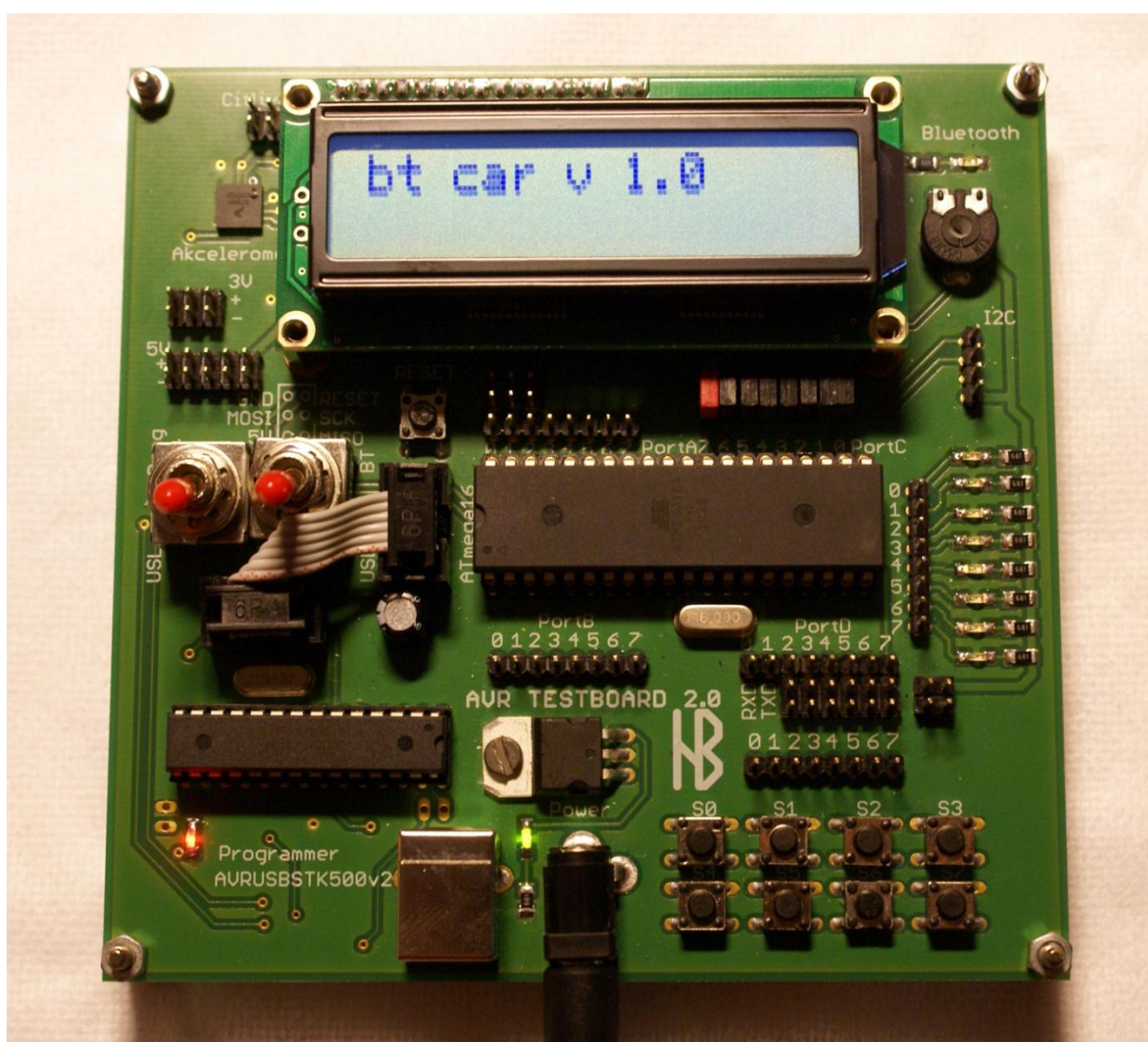


Elektrolytický kondenzátor



Výukový program pro desku AVR Testboard 2.0

Karel Babický



Úvodní slovo

Tento výukový program vznikl jako součást dlouhodobé maturitní práce pro školní rok 2010/2011. Cílem tohoto programu je nastínit základy programování mikrokontrolerů Atmel AVR konkrétně procesoru ATmega16 a otestovat komponenty na vývojové desce. Tento program je jako doplňující součást k vývojové desce AVR Testboard 2.0.

Obsah

Úvodní slovo.....	2
1. AVR Testboard 2.0.....	3
1.1 Úvod k desce	3
1.2 Codevision AVR	5
Instalace programu	5
Vytvoření projektu	5
Vytvoření prvního programu.....	7
2. Ukázkové programy.....	9
2.1 Běžící světlo	9
2.2 UART komunikace	10
2.3 Ovládání modelářského serva	11
Použitá literatura.....	12

1. AVR Testboard 2.0

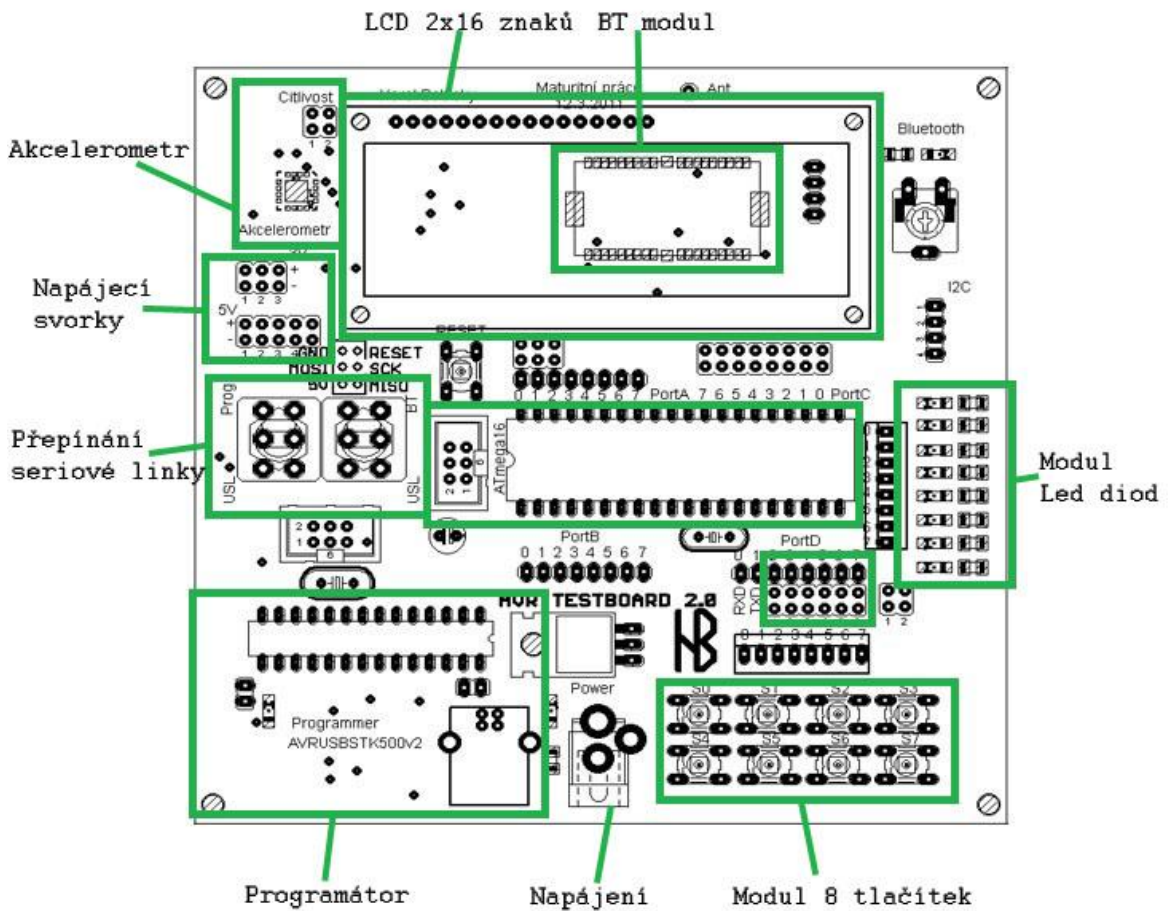
1.1 Úvod k desce

Srdcem desky AVR Testboard 2.0, je procesor ATmega16 (je možno použít i ATmega32). Deska mimo procesoru obsahuje spoustu periférií včetně celého programátoru s USB rozhraním. V procesoru není žádný bootloader jako u konkurenčních desek, ale na desce je plnohodnotný programátor s vlastním procesorem. Tím pádem můžeme za pomoci desky i programovat jiné procesory stejné řady.

Deska obsahuje spoustu periférií kromě modulů led diod a tlačítek, je přímo na desce integrovaný 3osý akcelerometr a bezdrátový bluetooth modul. Jako zobrazovací jednotka je použit LCD display se standartním řadičem.

Parametry desky :

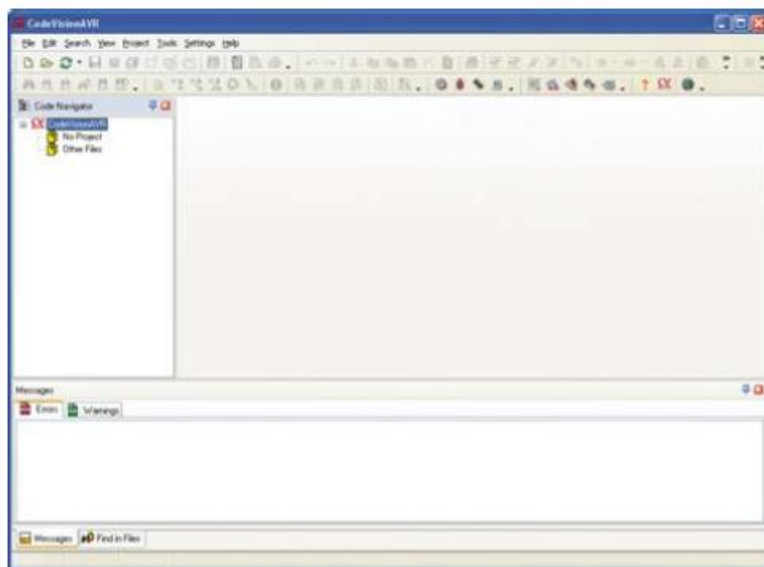
- Procesor ATmega16
- LCD 2x16 znaků s řadičem, podsvícený
- Analogový akcelerometr MMA7260Q
- 8 tlačítek
- 8 indikačních led diod
- Možnost připojení modelářských serv
- Bluetooth bezdrátový modul pro UART komunikaci
- USB programátor
- USB sériové rozhraní
- ISP programovací konektor
- Možnost pomocí programátoru programovat další procesory ATmega
- Externí napájení
- Napájecí napětí 7-18V
 - o 5V větev stabilizátor 7805
 - o 3V větev regulátor LM3940
- Napájení pomocí USB



- pomocí páčkových přepínačů se dá nastavit použití sériové linky. Prvním přepínačem se přepíná mez polohami prog (připojený programátor) a USL (USB sériová linka). Druhým přepínačem se připíná zdroj sériové linky k procesoru: BT (bluetooth), nebo USL (USB Sériová linka). Aby fungovala USL musí být oba přepínače v poloze USL.
- U portu D je možné připojit 5 modelářský serv, napájení pro ně je možné přivést z desky, ale pro jejich velký odběr je lepší připojit externí napájení. Interní napájení se připojuje pomocí dvojice jumperů vedle portu D a externí se přivádí přímo na piny jumperů.

1.2 Codevision AVR

Pro vývoj programů je pro začátečníky nejlepší program CodevisionAVR od firmy HP infotech. Tento program ve verzi evaluation je k dispozici zdarma k nekomerčním účelům a je limitován délkou zdrojového kódu a omezením použití některých knihoven.



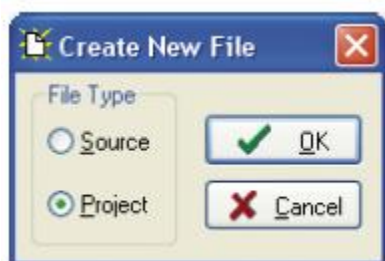
Obrázek 1.1: Vývojové prostředí CodevisionAVR

Instalace programu

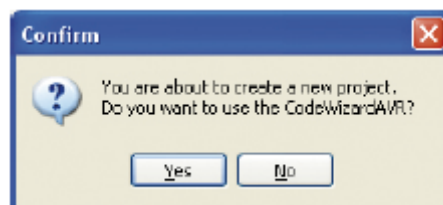
Stáhneme nejnovější verzi programu ze stránek <http://www.hpinfotech.ro/> stáhneme si verzi „CodeVisionAVR Evaluation“. Program se standardně nainstaluje na zvolený disk do adresáře „cvavr2“. Na instalaci a spuštění programu je třeba mít pro operační systémy Windows® 2000, XP, Vista nebo Windows 7 přístupová práva **Administrator**. Pro spuštění již nainstalovaného programu pod systémy Windows® 2000 nebo XP stačí práva pro **Power User**.

Vytvoření projektu

Pro vytvoření prvního programu. Zvolíme **File->New (Ctrl+N)**, na dotaz zda vytvořit „Source“ (soubor) nebo „Project“ (projekt) odpovíme Project (obr. 1.2.) a dále „Yes“ na použití průvodce (CodeWizzardAVR)– obr. 1.3.

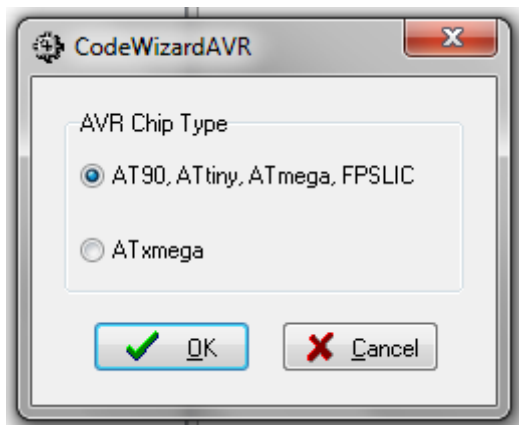


Obrázek 1.2: Volba v CodevisionAVR

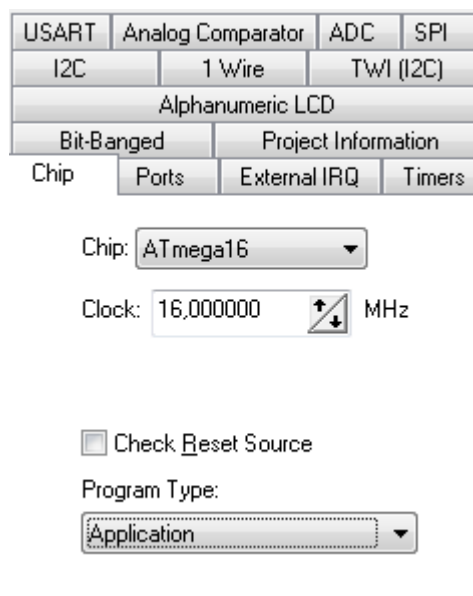


Obrázek 1.3: Volba v CodevisionAVR

Poté zvolíme ok (obr. 1.4). Poté se otevře průvodce nastavením projektu kde zvolíme typ procesoru. Na desce AVR Testoboard 2.0 je procesor ATmega 16 taktovaný na frekvenci 16Mhz (obr. 1.5).

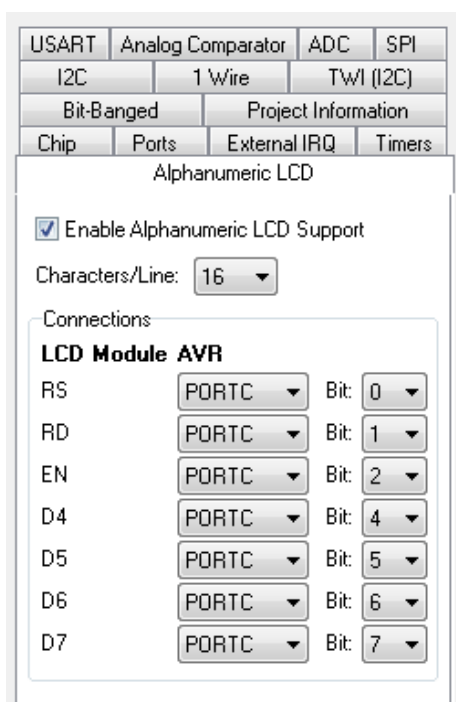


Obrázek 1.4: Volba série procesorů

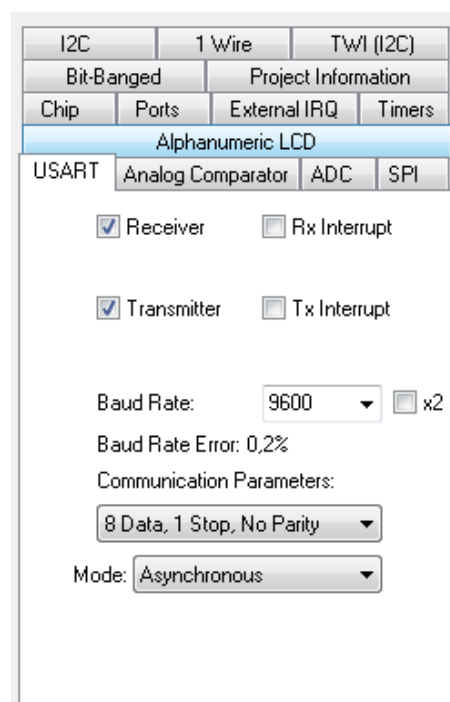


Obrázek 1.5: Volba typu procesoru

Pro správnou funkci procesoru můžeme v průvodce donastavit na jakých portech chceme mít výstup a jestli chceme použít knihovny pro lcd a mnoho dalšího. My si nastavíme, že máme na port C připojený lcd zobrazovač (obr. 1.6) a že máme připojenou seriovou linku pro komunikaci (obr. 1.7). UART seriovou linku prozatím nastavíme bez externího přerušení (změna nastavení UBRRL=0x33;).



Obrázek 1.6: Nastavení LCD



Obrázek 1.7: Nastavení UART

Pomocí **File->save** uložíme toto konkrétní nastavení pro další projekty. Pro vygenerování projektu už zvolíme jen **program->Generate, save and exit**. Otevře se nám vygenerovaný projekt a zdrojový kód s přednastavenými parametry pro porty, UART a LCD. Průvodce také vloží do hlavičky potřebné knihovny pro obsluhu lcd a ostatních součástí procesoru.

Vytvoření prvního programu

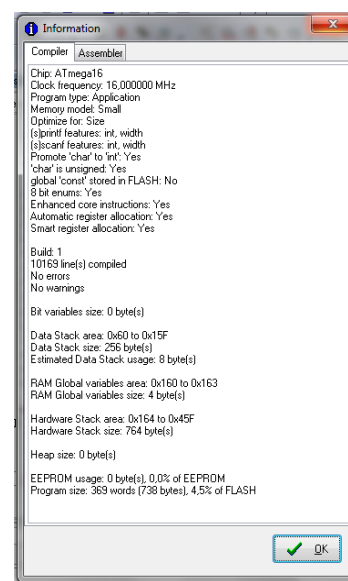
Ve vytvořeném projektu přejdeme do spodní části kde je od průvodce vytvořená nekonečná smyčka. Pro první program se pokusíme vypsát text na lcd (obr. 1.8)

```
145 // Alphanumeric LCD initialization
146 // Connections specified in the
147 // Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
148 // RS - PORTC Bit 0
149 // RD - PORTC Bit 1
150 // EN - PORTC Bit 2
151 // D4 - PORTC Bit 4
152 // D5 - PORTC Bit 5
153 // D6 - PORTC Bit 6
154 // D7 - PORTC Bit 7
155 // Characters/line: 16
156 lcd_init(16);
157 lcd_gotoxy(0,0);
158 lcd_putsf("AVR Testboard 2.0");
159
160 while (1)
161 {
162     // Place your code here
163 }
164
165 }
```

Obrázek 1.8: Vypsání textu na LCD

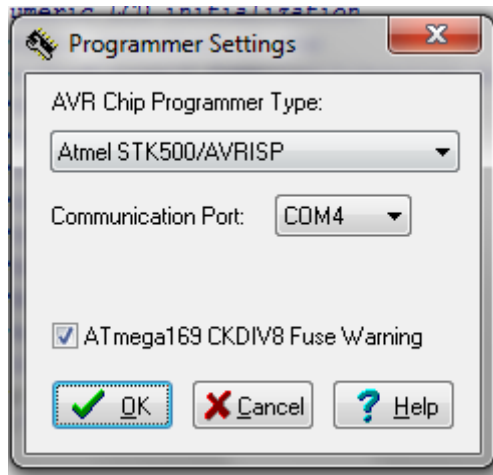
Místo AVR Testboard 2.0 si každý může mezi uvozovky napsat cokoliv, co se mu líbí ☺. Po napsání těchto 2 příkazů přejdeme nahore na **Projekt->Build ALL** a jestli nemáme v zápisu chybu, měla by se objevit tabulka (obr. 1.9). Potvrdíme ok a ve složce s projektem jsme vytvořili hex soubor, který můžeme nahrát přímo do procesoru. Jelikož součástí vývojového prostředí CodeVisionAVR je i integrovaný programátor, můžeme hned přímo z programu nahrát firmware do procesoru.

Pro správnou funkci programátoru ho musíme v programu nejprve nakonfigurovat. Připojíme desku přes USB k počítači a čekáme, až se nainstaluje převodník (operační systém Windows Vista a Seven mají ovladač již integrovaný, pro Windows XP je přímo na cd se zdrojovými kódy k příkladům programu). Po nainstalování převodníku musíme zjistit k jakému COM portu se převodník nakonfiguroval. To se dá zjistit ve správci zařízení pod porty COM LPT.



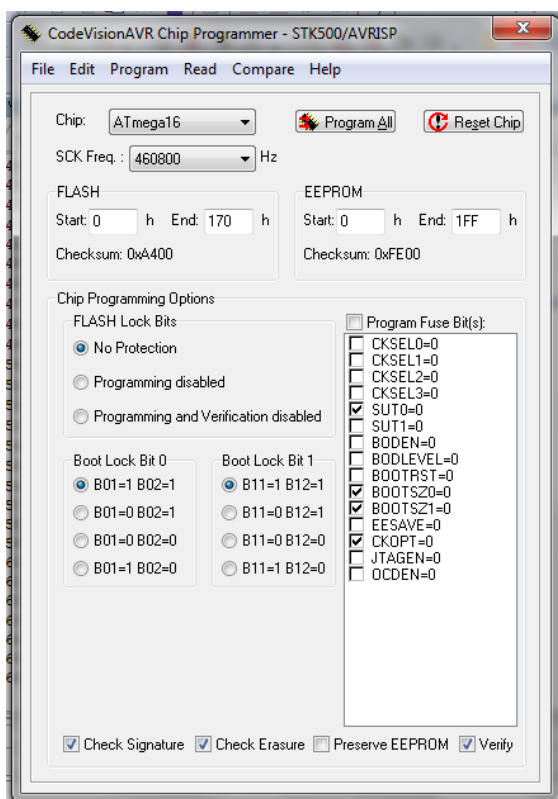
Obrázek 1.9: Po Kompilaci

Po zjištění portu jdeme zpátky do CodeVision a zvolíme **Settings->Programmer** a nastavíme číslo portu a typ programátoru na STK500 (obr. 1.10)

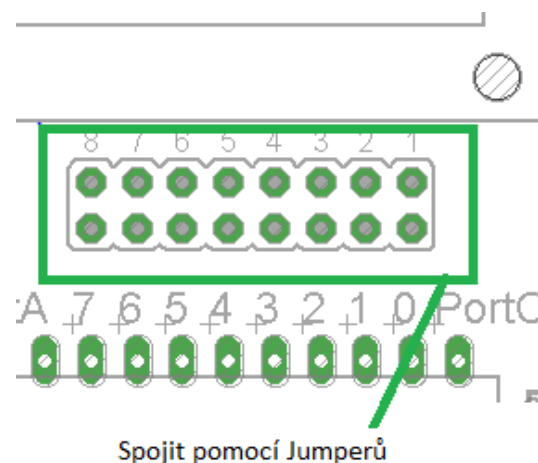


Obrázek 1.10: Nastavení portu

Teď můžeme přejít k vlastnímu programování. Zvolíme **Tools->Chip Programmer**. Nastavíme typ procesoru na ATmega16. Zvolíme **File->Load FLASH** a ve složce Exe je uložený náš program pro procesor, následně klikneme na Program ALL a u EEPROM zvolíme NO. A jestli je vše správně nastaveno a zapojeno měl by být náš první program nahraný v procesoru. Po zapnutí desky by se měl na LCD vypsát text, který jsme si zvolili (pokud je správně propojen port s LCD (obr. 1.13)).



Obrázek 1.12: Nastavení portu



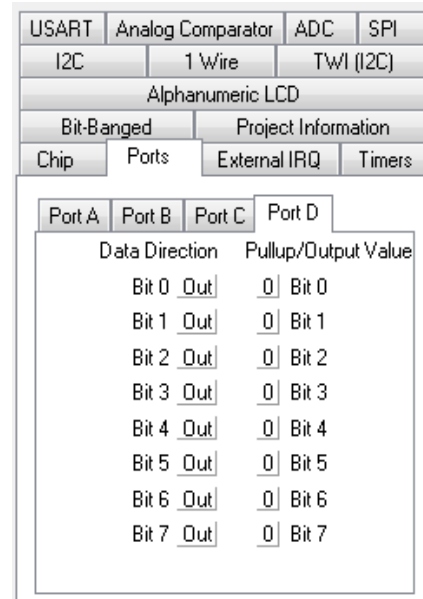
Obrázek 1.13: Propojení LCD

2. Ukázkové programy

2.1 Běžící světlo

Jako jednoduchý program pro začátek se hodí například program pro běžící světlo. Vytvoříme si nový projekt s předchozími nastaveními, jen přidáme nastavení vstupně výstupních portů (obr. 2.1). Jako výstupní port si zvolíme například port B, který je na desce blízko led diod. Celý port propojíme s modulem led diod pomocí propojovacích kablíků.

Po dokončení propojení a nastavení projektu, můžeme začít programovat. V projektu přejdeme do předvytvořené nekonečné smyčky. Kde se pokusíme jednoduše naprogramovat běžící světlo, pomocí jednoduchého příkazu pro bitový posun v registru portu a generovat zpoždění pomocí knihovny. Vedle „`#include <mega16.h>`“ přidáme ještě „`#include <delay.h>`“.



Obrázek 2.1: Nastavení portů

```
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0xFF;
DDRB=0xFF;
```

Obrázek 2.2: Nastavení portů v programu

Ještě musíme předem deklarovat proměnnou pod „`// Declare your global variables here`“ vložíme **unsigned char c**; tím v c dopředu deklaruje použítanou proměnnou.

```
c = 0b00000001; //nastavení brány
while (1)
{
    // prostor pro kod
    for(i=0;i<7;i++){
        c = c<<1; //pro bytový posun na druhou stranu použijeme ">>"
        PORTB = ~c; //po posunutí o jeden bit přepošleme do registru portu
        delay_ms(300); //nastavíme časovou prodlevu 300ms mezi dalším cyklem
    }
    c = 0b00000001; //znovunastavení brány
}
}
```

Obrázek 2.2: Nastavení portů v programu

2.2 UART komunikace

Jako další příklad si vyzkoušíme UART sériovou komunikaci mezi deskou a počítačem. Budeme posílat text pomocí konzole v sériové lince a vypisovat na LCD. Vytvoříme si projekt s vloženou knihovnou pro LCD a povolenou obousměrnou UART komunikací (změna nastavení UBRRL=0x33;). Na desce propojíme LCD a zvolíme druh sériové komunikace např. přes bluetooth. Nejprve musíme zase přednastavit proměnné (obr. 2.3).

```
#include <mega16.h>

// Alphanumeric LCD Module functions
#include <alcd.h>

// Standard Input/Output functions
#include <stdio.h>
|
#include <delay.h>

// Declare your global variables here

} void main(void)
|
{
// Declare your local variables here
unsigned char a;
unsigned int i,j;
```

Obrázek 2.3:Nastavení knihoven a deklarace proměnných

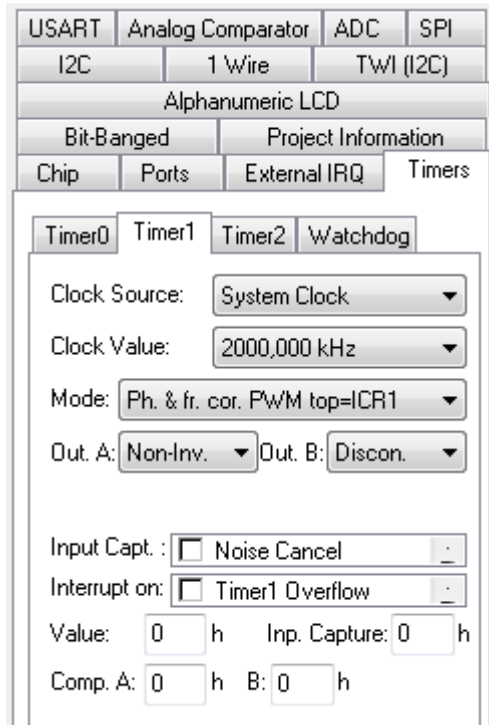
```
printf("pripojeno"); //pošle do konzole jestli se deska připojila

while (1)
{
// Place your code here
for(i=0;i<2;i++) // 2 řádkové lcd
{
for(j=0;j<16;j++) // 16 znaků na řádce
{
a = getchar(); //přijme znak ze konzole po ser lince
delay_ms(10);
lcd_gotoxy(j,i); //nastaví pozici na lcd
lcd_putchar(a); //pošle zpět znak na lcd
delay_ms(50);
putchar( a); //vrátí znak zpět do konzole
}
}
}
}
```

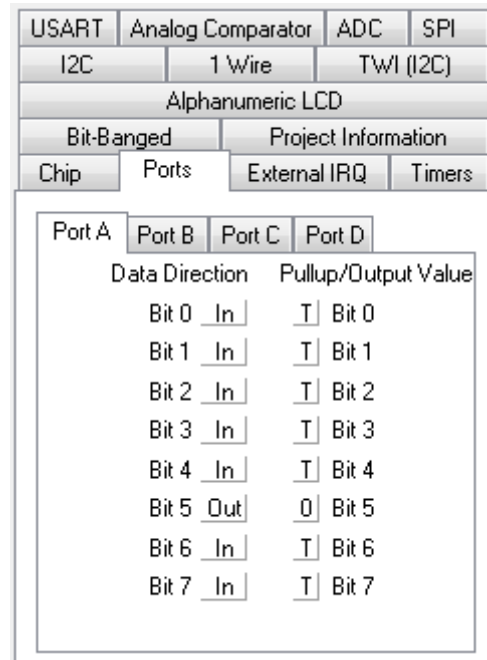
Obrázek 2.4: Program pro komunikaci po sériové lince

2.3 Ovládání modelářského serva

Modelářské servo se ovládá pomocí pulzní šířkové modulace konkrétně pulzy o frekvenci 50hz a délce od zhruba 800 do 1800ms podle typu modelářského serva. Pro generování toho signálu použijeme 16 bitový čítač procesoru.



Obrázek 2.5: Nastavení čítače



Obrázek 2.6: Nastavení výstupního portu

Podle obrázků 2.5 a 2.6 nastavíme v průvodci čítač a výstupní porty a vygenerujeme projekt. Poté budeme muset ještě upravit části nastavení kódu podle obrázku 2.7 a do záhlaví programu vložit knihovnu zpoždění `#include <delay.h>`. V programu přenastavíme hodnoty registrů v čítači podle následujícího obrázku 2.7. Registr OCR1H a OCR1L nastavíme na 2000 = 4E20_h a registr OCR1A nastavíme na počáteční hodnotu 1000ms (vyšle pulz o šířce 1000ms. Dále už budeme jen generovat zpoždění a přenastavovat délku pulzu v registru OCR1A podle obrázku 2.8

```
TCCR1A=0x80;
TCCR1B=0x12;
TCNT1H=0x00;
TCNT1L=0x00;

ICR1H=0x4e;
ICR1L=0x20;

OCR1A=1000;

OCR1BH=0x00;
OCR1BL=0x00;
```

Obrázek 2.7: Nastavení čítače

```
while (1)
{
    // Place your code here
    OCR1A = 800;
    delay_ms(1500);

    OCR1A = 2400;
    delay_ms(1500);
};
```

Obrázek 2.8: Hlavní program

Použitá literatura

- [1] VÁŇA, Vladimír . *Mikrokontroléry Atmel AVR - programování v jazyce C*. Praha : BEN, 2003. 216 s. ISBN 80-7300-102-0.