

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta - Katedra fyziky

Výuková sestava pro práci s mikrokontroléry AVR Atmel

Diplomová práce

Vedoucí práce: Ing Michal Šerý

Autor: Bc. Miroslav Kubeš

Anotace

Tato diplomová práce se zabývá návrhem a výrobou výukové stavebnice s mikrokontrolérem AVR Atmel. Tato stavebnice obsahuje procesorovou desku a moduly, které se k desce budou připojovat. Popisuje softwarové vybavení potřebné k návrhu a naprogramování mikrokontroléru a technologické postupy pro vytvoření stavebnice. Také v ní najdeme základní teoretické znalosti pro práci s mikrokontrolérem ATmega8535. Stavebnice spolu s diskem, na kterém budou nahrána veškerá schémata, obrázky a programy mnou vytvořené, budou přiloženy k diplomové práci.

Abstract

This thesis deals with design and production of teaching kits with Atmel AVR microcontroller. This kit includes a processor board and modules, which will connect to the board. It describes the equipment needed for software design and programming of microcontroller and technological processes to create a kit. Also, it will find the basic theoretical knowledge to work with the ATmega8535 microcontroller. Kit along with a disc on which are recorded all the diagrams, pictures and programs created by me, will be attached to the thesis.

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně, pouze s použitím pramenů literatury, uvedených v seznamu literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě, v úpravě vzniklé vypuštěním vyznačených částí archivovaných pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

Datum: 3.12.2011

Podpis:

Touto formou bych chtěl poděkovat panu Ing. Michalu Šerýmu za vedení mé diplomové práce, za jeho trpělivost a cenné rady a připomínky, díky kterým jsem vyřešil více problémů týkajících se mé práce.

| | |
|-------------------------------------------------------|-----------|
| 1 Seznam obrázků | 7 |
| 2 Seznam tabulek | 9 |
| 3 Úvod | 10 |
| 4 Společnost Atmel | 11 |
| 5 ATmega8535 | 13 |
| 5.1 Základní parametry ATmega8535 | 13 |
| 5.2 Blokové schéma ATmega8535 | 15 |
| 5.3 Princip funkce mikrokontroléru | 16 |
| 5.4 AVR Centrální procesorová jednotka (CPU) | 16 |
| 3.6 Popis jednotlivých pinů a jejich funcce | 20 |
| 5.7 Charakteristika práce portů | 30 |
| 6 Výběr Softwaru pro návrh schémat a DPS | 33 |
| 6.1 Eagle 5.6.0 | 34 |
| 7 Návrh El. Schémat a DPS | 37 |
| 7.1 Procesorová deska..... | 37 |
| 7.1.1 Použité součástky | 39 |
| 7.1.2 Schéma procesorové desky | 40 |
| 7.1.3 Stabilizační obvod..... | 41 |
| 7.1.4 Rozhraní RS232 | 43 |
| 7.1.5 ISP rozhraní | 44 |
| 7.1.6 Připojení portů | 46 |
| 7.1.7 Resetovací obvod | 47 |
| 7.1.8 Napájení a krystal | 48 |
| 7.1.9 Návrh masky DPS procesorové desky..... | 50 |
| 7.2 Přípojně moduly..... | 54 |
| 7.2.1 Tlačítka | 55 |
| 7.2.2 Ledky | 55 |
| 7.2.3 Modul propojovací deska..... | 57 |
| 7.2.4 Modul LCD | 58 |
| 7.2.5 Moduly Teploměr (I2C, 1-Wire) | 59 |

| | |
|----------------------------------------------------|-----------|
| 8 Výroba plošného spoje..... | 60 |
| 8.1 Výroba DPS fotocestou..... | 61 |
| 8.1.1 Výroba masky | 62 |
| 8.1.2 Osvícení desky | 63 |
| 8.1.3 Zdroje světla..... | 63 |
| 8.1.4 Vyvolání..... | 64 |
| 8.1.5 Leptání | 64 |
| 8.2 Povrchová úprava desky | 65 |
| 8.3 Vrtání a pájení..... | 65 |
| 8.4 DPS Procesorové desky a některých modulů | 66 |
| 9 Programování v Bascomu | 67 |
| 10 Programátory | 71 |
| 11 Závěr | 72 |
| 12 Použité prameny a literatura | 73 |

1 Seznam obrázků

| | |
|-----------------------------------------------------------------------------|----|
| OBR. Č. 1 POROVNÁNÍ PŘÍJMŮ, PŘEVZATO A UPRAVENO Z [1] | 11 |
| OBR. Č. 2 PRODEJ MCU, PŘEVZATO A UPRAVENO Z [1] | 12 |
| OBR. Č. 3 NABÍDKA MCU A DATASHEETY, PŘEVZATO A UPRAVENO Z [3] | 12 |
| OBR. Č. 4 BLOKOVÉ SCHÉMA ATMEGA8535, PŘEVZATO A UPRAVENO Z [4] | 15 |
| OBR. Č. 5 BLOKOVÉ SCHÉMA CPU, PŘEVZATO A UPRAVENO Z [4] | 16 |
| OBR. Č. 6 FLASH PAMĚT, PŘEVZATO A UPRAVENO Z [4] | 17 |
| OBR. Č. 7 PAMĚT SRAM, PŘEVZATO A UPRAVENO Z [4] | 18 |
| OBR. Č. 8 KONFIGURACE PINŮ ATMEGA8535, PŘEVZATO A UPRAVENO Z [4] | 19 |
| OBR. Č. 9 RESETOVACÍ LOGIKA ATMEGA8535, PŘEVZATO A UPRAVENO Z [4] | 27 |
| OBR. Č. 10 MCU START-UP RESET SVÁZÁN S VCC, PŘEVZATO A UPRAVENO Z [4] | 28 |
| OBR. Č. 11 EXTERNÍ RESET V PRŮBĚHU OPERACE, PŘEVZATO A UPRAVENO Z [4] | 28 |
| OBR. Č. 12 BROWN-OUT DETEKCE, PŘEVZATO A UPRAVENO Z [4] | 29 |
| OBR. Č. 13 WATCHDOG RESET, PŘEVZATO A UPRAVENO Z [4] | 29 |
| OBR. Č. 14 ZNÁZORNĚNÍ ZAPOJENÍ OCHRAN PINU, PŘEVZATO A UPRAVENO Z [4] | 30 |
| OBR. Č. 15 OBECNÉ SCHÉMA PINU, PŘEVZATO A UPRAVENO Z [4] | 31 |
| OBR. Č. 16 REGISTRY PORTU A, PŘEVZATO A UPRAVENO Z [4] | 33 |
| OBR. Č. 17 EAGLE 5.6.0 CONTROL PANEL | 34 |
| OBR. Č. 18 ZÁKLADNÍ POPIS PROSTŘEDÍ EAGLE, PŘEVZATO A UPRAVENO Z [6] | 35 |
| OBR. Č. 19 TABULKA SE SOUČÁSTKAMI VYVOLANÁ PŘÍKAZEM ADD | 36 |
| OBR. Č. 20 NASTAVENÍ RASTRU | 36 |
| OBR. Č. 21 ZMĚNA FUSE BITŮ U ATMEGA8535 | 38 |
| OBR. Č. 22 SCHÉMA PROCESOROVÉ DESKY | 40 |
| OBR. Č. 23 STABILIZAČNÍ OBVOD | 41 |
| OBR. Č. 24 STABILIZÁTOR NAPĚTÍ 78S05, PŘEVZATO A UPRAVENO Z [7] | 41 |
| OBR. Č. 25 SCHÉMA OBVODU PRO RS232 | 43 |
| OBR. Č. 26 SCHÉMA PŘIPOJENÍ ISP | 44 |
| OBR. Č. 27 MOŽNOSTI PŘIPOJENÍ ISP, PŘEVZATO A UPRAVENO Z [9] | 45 |
| OBR. Č. 28 PŘIPOJENÍ PORTŮ K ATMEGA8535 | 46 |
| OBR. Č. 29 ZVÝRAZNĚNÍ SPOJENÍ POMOCÍ FUNKCE SHOW | 47 |

| | |
|-------------------------------------------------------------------------------|----|
| OBR. Č. 30 RESETOVACÍ OBVOD MIKROKONTROLÉRU | 47 |
| OBR. Č. 31 NAPÁJENÍ A KRYSTAL..... | 48 |
| OBR. Č. 32 SCHÉMA ZAPOJENÍ TL431, PŘEVZATO A UPRAVENO Z [10]..... | 49 |
| OBR. Č. 33 MODUL EAGLE PRO VYTVOŘENÍ MASKY DPS | 50 |
| OBR. Č. 34 ROZMÍSTĚNÉ SOUČÁSTKY NA PODKLADU | 51 |
| OBR. Č. 35 NASTAVENÍ AUTOROUTERU..... | 52 |
| OBR. Č. 36 AUTOMATICKÉ VYROUTOVÁNÍ..... | 52 |
| OBR. Č. 37 RUČNĚ UPRAVENÉ SPOJE SOUČÁSTEK..... | 53 |
| OBR. Č. 38 KONEČNÁ PODOBA MASKY DPS..... | 54 |
| OBR. Č. 39 SCHÉMA MODULU TLAČÍTKA | 55 |
| OBR. Č. 40 MASKA TLAČÍTEK PRO DPS | 55 |
| OBR. Č. 41 SCHÉMA MODULU LEDKY | 56 |
| OBR. Č. 42 MASKA MODULU LED PRO DPS | 56 |
| OBR. Č. 43 SCHÉMA PROPOJOVACÍ DESKY | 57 |
| OBR. Č. 44 MASKA PROPOJOVACÍ DESKY PRO DPS | 57 |
| OBR. Č. 45 SCHÉMA MODULU LCD | 58 |
| OBR. Č. 46 MASKA LCD MODULU PRO DPS | 58 |
| OBR. Č. 47 MODULY TEPLoměRY | 59 |
| OBR. Č. 48 MASKA TEPLoměRŮ PRO DPS..... | 60 |
| OBR. Č. 49 MASKY A JEJICH KVALITA, PŘEVZATO A UPRAVENO Z [14]..... | 63 |
| OBR. Č. 50 VYVOLÁVÁNÍ A LEPTÁNÍ DESKY, PŘEVZATO A UPRAVENO Z [14]..... | 65 |
| OBR. Č. 51 VYTVOŘENÁ STAVEBNICE, STRANA SOUČÁSTEK | 66 |
| OBR. Č. 52 VYTVOŘENÁ STAVEBNICE, STRANA SPOJE..... | 67 |
| OBR. Č. 53 NASTAVENÍ BASCOMU | 68 |
| OBR. Č. 54 PROGRAM LEDKY | 69 |
| OBR. Č. 55 PROGRAM TLAČÍTKA, LEDKY | 70 |
| OBR. Č. 56 STK500 MICROCONTROLLER STARTER KIT, PŘEVZATO A UPRAVENO Z [17].... | 71 |

2 Seznam tabulek

| | |
|------------------------------------------------------------------------------------|----|
| TAB. Č. 1 ALTERNATIVNÍ FUNKCE PINŮ PORTU B, PŘEVZATO A UPRAVENO Z [4] | 21 |
| TAB. Č. 2 ALTERNATIVNÍ FUNKCE PINŮ PORTU C, PŘEVZATO A UPRAVENO Z [4] | 23 |
| TAB. Č. 3 ALTERNATIVNÍ FUNKCE PINŮ PORTU D, PŘEVZATO A UPRAVENO Z [4] | 25 |
| TAB. Č. 4 CHARAKTERISTIKY RESET PULSŮ, PŘEVZATO A UPRAVENO Z [4] | 30 |
| TAB. Č. 5 SUMA MOŽNÝCH SIGNÁLU NA PINU, PŘEVZATO A UPRAVENO Z [4] | 32 |
| TAB. Č. 6 PROVOZNÍ PODMÍNKY A HODNOTY 78S05, PŘEVZATO A UPRAVENO Z [7] | 42 |
| TAB. Č. 7 ZÁKLADNÍ PARAMETRY A HODNOTY MAX232, PŘEVZATO A UPRAVENO Z [8] | 44 |
| TAB. Č. 8 TEPLOTNÍ KRITÉRIA PRO BEZPEČNOU MONTÁŽ, PŘEVZATO A UPRAVENO Z [15] | 66 |

3 Úvod

Úkol, který řeším ve své diplomové práci, je sestavit výukovou modulární stavebnici pro práci s mikrokontroléry AVR Atmel. Stavebnice obsahuje základní vývojovou desku osazenou mikrokontrolérem z rodiny Atmel a některé moduly, které se budou k stavebnici připojovat. Deska by měla být navržena tak, aby umožňovala variabilní možnosti připojení i různých zařízení vytvořených v budoucnu, ne jen modulů mnou vytvořených a následné naprogramování a programové odladění. K tomu je zapotřebí vybrat vhodný mikrokontrolér z rodiny Atmel, který bude splňovat většinu našich požadavků, seznámit se s jeho základními vlastnostmi a možnostmi využití. Vybrat vhodný software, ve kterém budeme navrhovat schémata procesorové desky, modulů a následně vytvářet masky plošných spojů. Seznámit se s problematikou výroby plošných spojů. Jak takový plošný spoj v odpovídající kvalitě a přesnosti vyrobit v běžných podmínkách. Následné osazení plošného spoje součástkami a oživení desky. Dále je zapotřebí seznámit se s možnostmi programování AVR, zvolit vhodný programovací jazyk a navrhnout a odladit programy pro komunikaci s moduly.

Vytvořená modulová stavebnice může sloužit jako výuková stavebnice, kterou budou žáci používat při práci s mikrokontroléry Atmel. Vytvořené moduly a programy mohou sloužit jako základní či vzorové úlohy, na kterých lze pochopit základy používání mikrokontrolérů Atmel a základy programování. Možností, jak se stavebnicí pracovat a jak Atmel využívat, je celá řada. Např.: rozblikání ledek, čtení teploty, LCD zobrazovač, komunikace s PC přes sériovou linku RS232 atd. Popřípadě žáci mohou vytvářet své vlastní moduly, přičemž musí projít celým procesem návržení a výroby a tím získávají cenné zkušenosti i z praktické části. Vzhledem k tomu, že škála možností jak se stavebnicí pracovat je obrovská, je vhodná pro každého žáka, záleží pouze jaký cíl si zvolí na základě svých možností, znalostí a dovedností.

V diplomové práci se seznámíme s mikrokontroléry z rodiny Atmel a jejich vlastnostmi a použitím. Návrhem a výrobou desky, modulů a programovým vybavením potřebným k realizaci stavebnice.

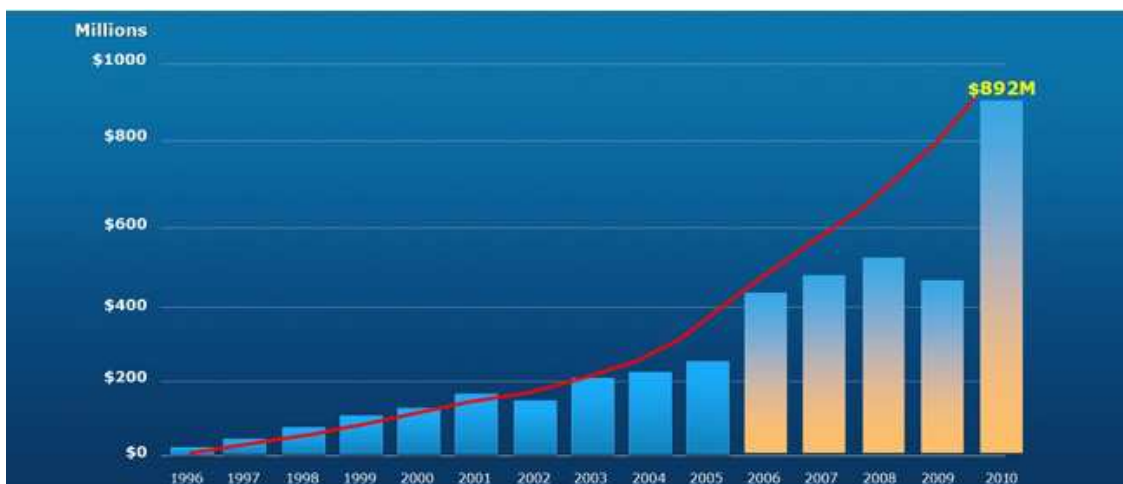
4 Společnost Atmel

Společnost Atmel sídlí v San Jose v Kalifornii a byla založena roku 1984. Má více než 5100 zaměstnanců po celém světě a v současnosti je světovým lídrem v designu a výrobě různých druhů mikroprocesorů, na které se momentálně primárně zaměřuje. Dále také vyrábí kapacitní dotykové obrazovky, vyspělé logické obvody, energeticky nezávislé paměti, wireless moduly, RF komponenty a další. [1]

V neposlední řadě se také věnuje automobilovému průmyslu, kde se zabývá automobilovou elektronikou, jako je návrh a realizace řešení palubních počítačů a jejich komunikace s motorem, různé diagnostické prvky, sítě a osvětlení a také zábavní prvky. Je kladen velký důraz na miniaturizaci, univerzálnost, velmi malou spotřebu energie a schopnost práce při vysokých teplotách a v elektromagnetickém či jinak rušeném prostředí podle požadavků zákazníka a šetrnost k životnímu prostředí. Díky různorodým potřebám automobilového průmyslu je k dispozici široká škála výrobků, od nejjednodušších, až po velice složité integrované obvody s širokou škálou funkcí, rozsáhlými možnostmi připojení, propracovaným rozhraním a velkým zabezpečením. Všechny výrobky splňují přísné standardy automobilového průmyslu, ve kterém má společnost Atmel více než dvacetiletou zkušenost. [2]

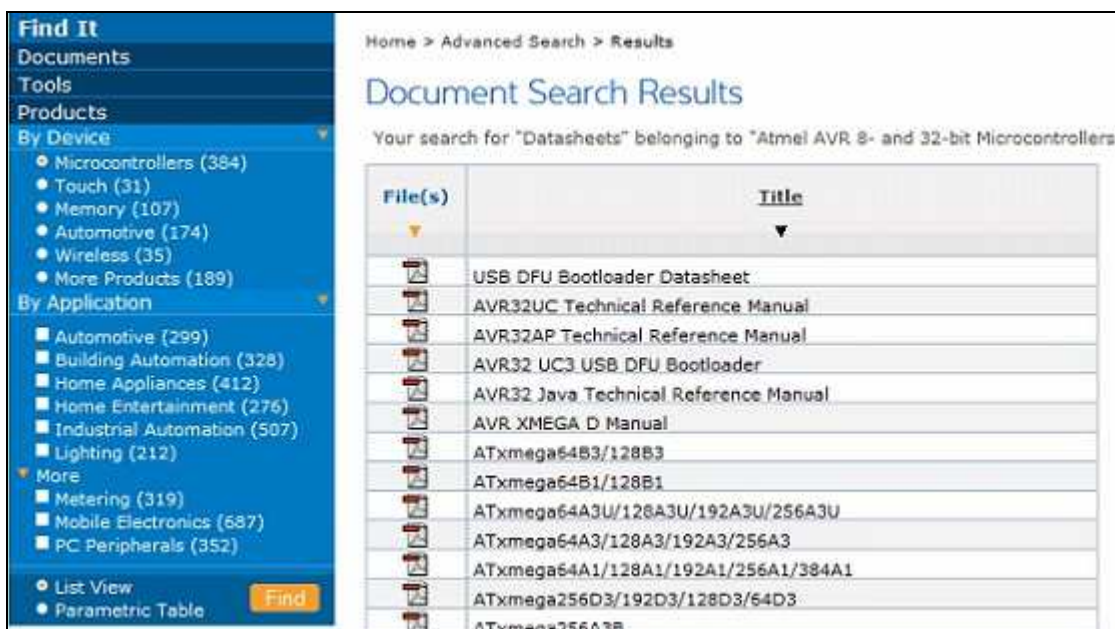


Obr. č. 1 Porovnání příjmů, převzato a upraveno z [1]



Obr. č. 2 Prodej MCU, převzato a upraveno z [1]

Z obrázku č. 1 a 2 můžeme pozorovat jak se po roce 2006 Atmel zaměřuje zejména na mikrokontroléry, ze kterých má společnost v současnosti hlavní příjem. Proto není náhodou, že Atmel nabízí zhruba tři tisíce mikrokontrolérů, které se liší použitím, konstrukcí a parametry. Ke každému mikrokontroléru samozřejmě poskytuje detailní popis (možnosti využití a různé možnosti zapojení, parametry) ve formě datasheetu, který lze stáhnout na stránkách výrobce.



Obr. č. 3 Nabídka MCU a datasheety, převzato a upraveno z [3]

5 ATmega8535

Aby bylo možné vývojovou stavebnici s mikrokontrolérem AVR Atmel vytvořit, musel jsem zvolit z široké škály vhodný mikrokontrolér, který by vyhovoval požadavkům na stavebnici a na přípojné moduly, které mám za úkol vytvořit. Bylo potřeba také zohlednit možné dodatečné moduly, které by popřípadě vytvořil někdo jiný a které by měly jít k mikrokontroléru též bez problémů připojit. Vhodných mikrokontrolérů by asi mohlo být samozřejmě více a díky patici, která bude na desce, se budou moci i popřípadě vyměnit, nicméně jsem vybral mikrokontrolér ATmega8535, který je spolu s ATmega32 hojně rozšířen pro různá domácí či industriální použití. Pouzdra se dělají ve dvou variantách, přičemž jsem vybral klasické obdélníkové DIL40.

5.1 Základní parametry ATmega8535

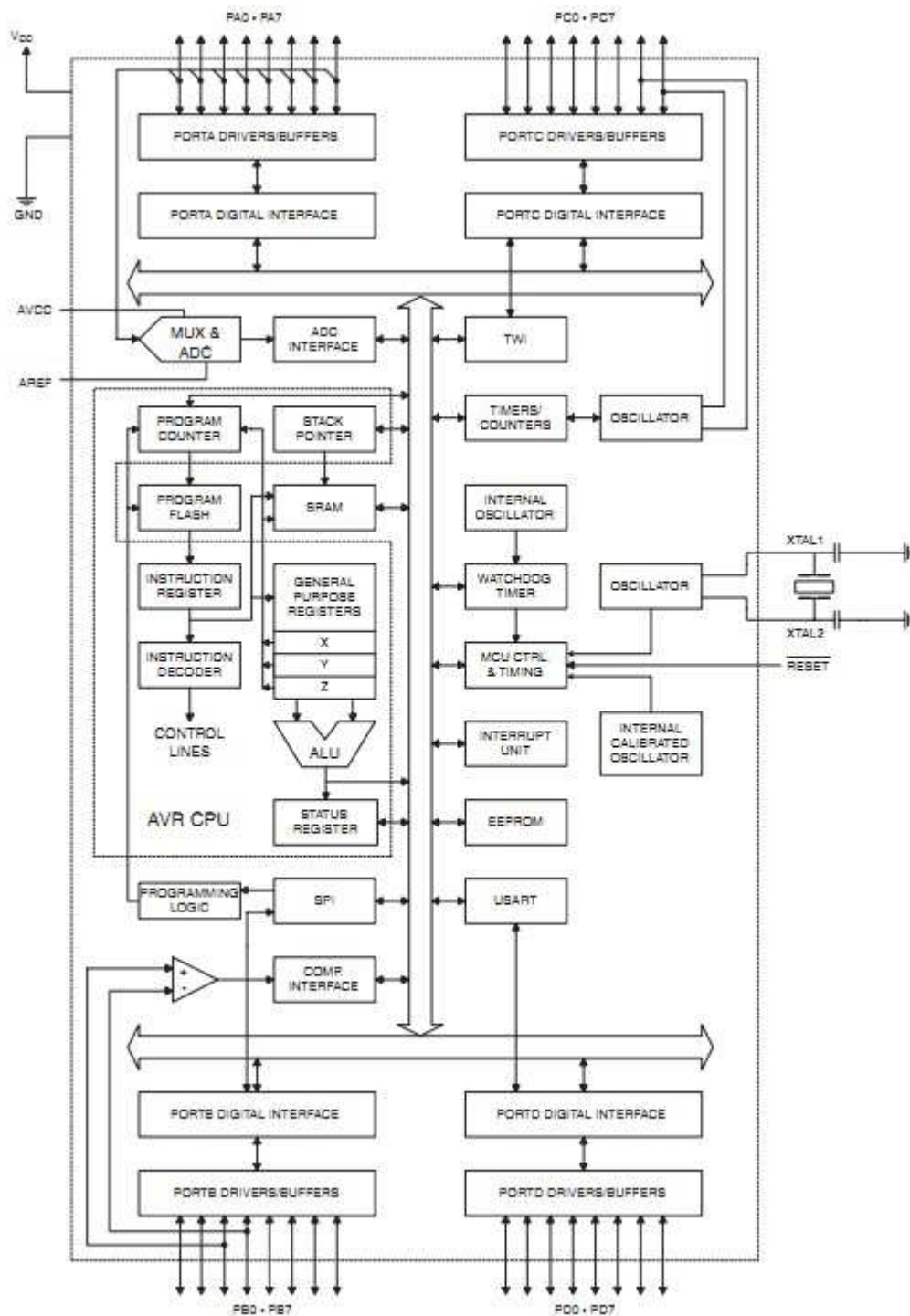
- Výkonný, nízkonapěťový AVR 8-bitový mikrokontrolér
- RISC architektura, může vykonat až 130 instrukcí na jeden takt, obsahuje 32 univerzálních pracovních registrů.
- Permanentní paměti dat a programu, 8K bytů ve vlastní programovatelné flash paměti, která vydrží 10 tisíc zapisovacích a mazacích cyklů, skutečná možnost čtení i když probíhá zápis.
- 512 bytová EEPROM s životností 100 tisíc zapisovacích a mazacích cyklů.
- 512 bytová interní SRAM
- Programovatelné zámky pro ochranu nahraného softwaru.
- Periferní funkce, dva 8 bitové čítače, jeden 16 bitový čítač, sériová linka, SPI interface, Programovatelný watchdog timer s odděleným oscilátorem na chipu...
- Speciální funkce mikrokontroléru, reset, kalibrace interního RC oscilátoru 1MHz, externí a interní zdroje přerušení, 6 sleep modů.
- I/O signály, 32 programovatelných I/O linek, celkem 40 pinů
- Operační napětí od 4,5V - 5,5V ATmega8535, 2,5V - 5,5V ATmega8535L
- Možnost připojení externího oscilátoru až 8MHz pro ATmega8535L a až 16MHz pro ATmega8535 [4]

AVR jádra kombinují bohatou instrukční sadu s dvaatřiceti univerzálními pracovními registry. Všech třicet dva registrů jsou přímo spojeny s aritmeticko-logickou jednotkou (ALU), která k sobě umožňuje přístup dvou nezávislých registrů najednou v jednom hodinovém taktu. Díky pokročilé RISC AVR architektuře využívající CISC instrukce jsou mikrokontroléry desetkrát rychlejší, než klasické CISC mikrokontroléry. [4]

ATmega8535 poskytuje následující funkce: 8K bytovou systémovou programovatelnou flash paměť schopnou skutečného čtení, i když je do ní právě zapisováno. 512 Bytovou EEPROM, 512 bytovou SRAM, 32 hlavních I/O linek, 32 univerzálních pracovních registrů, tři flexibilní časovače/čítače s porovnávacím módem, interní a externí přerušení, sériové programovací rozhraní USART, Two-wire Serial Interface, 10-bit A/D převodník, programovatelný watchdog timer s interním oscilátorem, SPI rozhraní, šest softwarově řízených power saving modů, další časovače/čítače. Power-down system, který zaručí konzistenci registrů, ale zmrazí oscilátor. Vypnutí všech ostatních funkcí mikrokontroléru, dokud nepřijde následující přerušení, nebo hardwarový reset. Power-save mode, kdy je povoleno vykonávání běhu časovače/čítače a dalších funkcí nastavených uživatelem, i když zařízení odpočívá, nebo spí. A/D režim omezení šumu, kdy se zastaví cpu a všechny I/O moduly, včetně asynchronního časovače a ostatních A/D převodníků, za účelem minimalizace chyby převodu. Stand-by mód, kdy běží pouze krystal/oscilátor, když zbytek zařízení spí. Toto umožňuje velice rychlý start kombinovaný s nízkou spotřebou. V rozšířeném Stand-by režimu běží oba hlavní oscilátory spolu s asynchronním časovačem. [4]

Zařízení používá velkokapacitní energeticky nezávislé paměti od společnosti Atmel. In system programing (ISP) využívající serial peripheral interface (SPI) rozhraní. Zařízení lze programovat různě. Například pomocí programátoru AVR STK500, nebo STK600 za pomoci AVR studia, nebo Bascomu, popřípadě jinak. Program se nejprve načte do bufferu počítače, ze kterého se nahraje na programovatelnou flash paměť zařízení, které může vykonávat starý program až do doby, než bude nový program zcela nahrán na flash paměť zařízení. Toto je umožněno díky pokročilé flash paměti schopné čtení i při zápisu. ATmega8535 je "silný" mikrokontrolér, který poskytuje vyrovnaný efektivní výkon za přijatelnou cenu. [4]

5.2 Blokové schéma ATmega8535

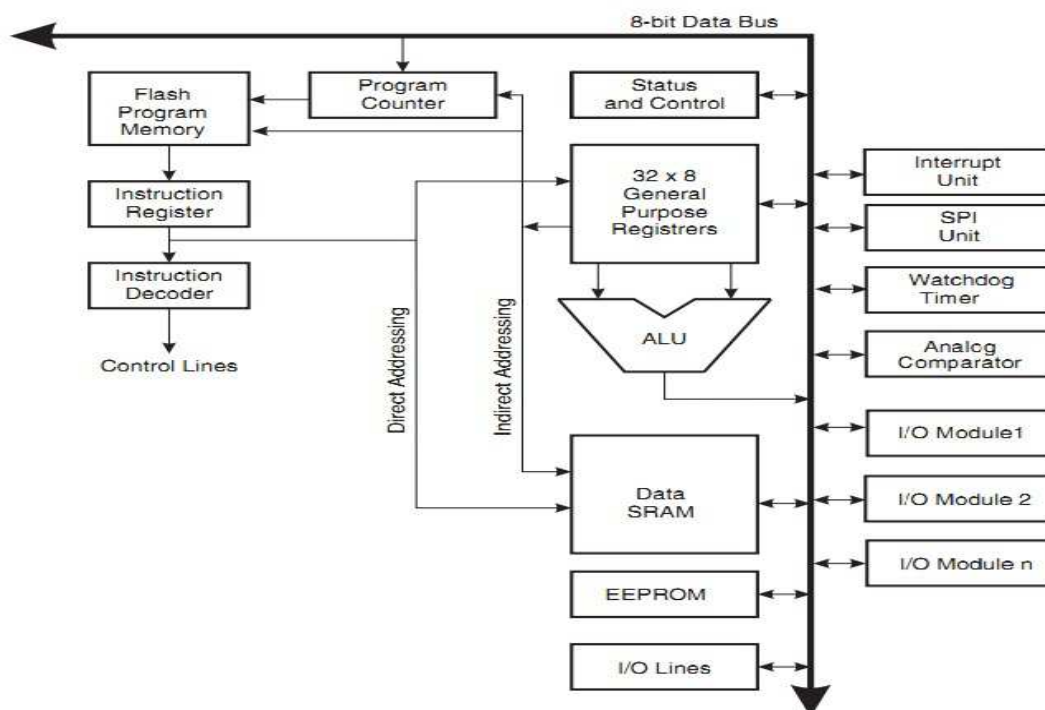


Obr. č. 4 Blokové schéma ATmega8535, převzato a upraveno z [4]

5.3 Princip funkce mikrokontroléru

ATmega8535 je logický obvod, jehož chod a téměř veškerá činnost je řízena programem. Program se ukládá při programování mikrokontroléru do paměti typu EEPROM. Po připojení napětí mikrokontrolér začíná vykonávat program, který je v paměti uložen až do té doby, dokud není odpojen od napětí, popřípadě dokud není program přepsán, nebo jinak přerušen. Při běhu programu čítač programu (PROGRAM COUNTER) vybere instrukci z paměti a přesune jí do dekodéru instrukcí, který instrukci identifikuje a podle typu instrukce se začne provádět příslušná činnost (přesun dat, aritmeticko-logické operace, různé bitové operace, skoky atp.). Paralelně s vykonáváním programu jsou mikrokontrolérem prováděny další operace, které na programu závislé nejsou (kontrola napájení, kontrola zdrojů přerušení, analogový komparátor, analogově digitální převody, sériová linka,...). Samozřejmě, že hodnoty které mikrokontrolér dostane z těchto procesů, mohou ovlivnit chod programu, přerušit ho, ukončit ho, resetovat mikrokontrolér atp.

5.4 AVR Centrální procesorová jednotka (CPU)

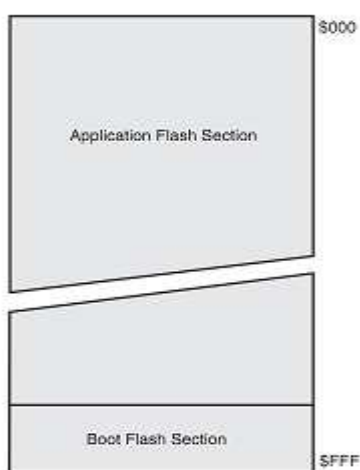


Obr. č. 5 Blokové schéma CPU, převzato a upraveno z [4]

Jádro centrální procesorové jednotky AVR je navrženo za účelem maximalizace výkonu a možnosti paralelně zpracovávat data. AVR používá Harvardskou architekturu s oddělenými paměťmi pro data programu. Instrukce v programové paměti jsou vykonávány po jedné datové lince. V momentě, kdy je jedna instrukce vykonávána, další je již stažena z programové paměti a čeká až bude na řadě. Tento systém řazení a vykonávání instrukcí pracuje stejně každý hodinový takt. [4]

Přístup k třiceti dvou osmibitovým univerzálním registrům je velice rychlý. Přístup k registrům je možný každý hodinový takt. Vždy dva na sobě nezávislé registry přistupují k ALU, která provede dané operace a výsledek je opět uložen zpět v registrech a to vše v jednom hodinovém taktu. Šest z třiceti dvou osmibitových registrů může být použito jako tři šestnácti bitové nepřímé ukazatele registrů pro adresování datového úložiště. Jeden z těchto adresových ukazatelů může být použit také jako adresový ukazatel pro vyhledávání ve Flash paměti (Flash program memory). Tyto tři šestnácti bitové registry jsou značeny ve schématu na obr. č. 4 jako x,y,z registry. [4]

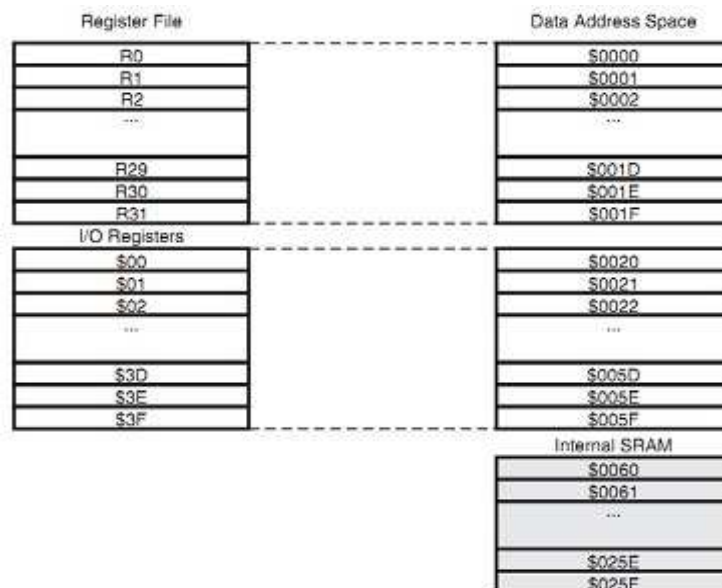
Vykonávání celého programu zaručuje přímé volání adres instrukcí a podmíněné a nepodmíněné skoky, díky kterým může mikrokontrolér ovládat celý adresní prostor. [4]



Programová Flash paměť je rozdělena na dvě sekce. Na bootovací sekci programu a sekci programu pro aplikaci. Obě sekce mají bitový zámeček pro ochranu čtení a zápisu. Instrukce, která chce zapisovat do sekce s aplikací, musí vycházet s bootovací sekce, jinak je tato instrukce zakázána. Velikost flash paměti je 8k bytů a vydrží zhruba 10k přepisů. Z paměti se dá číst i když je do ní právě zapisováno. [4]

Obr. č. 6 Flash paměť, převzato a upraveno z [4]

V průběhu přerušení a volání subrutin jsou vrácené adresy program counterem (PC) ukládány v zásobníku. Zásobník je efektivně přidělen do hlavní paměti SRAM a velikost zásobníku je limitována pouze velikostí SRAM. Na tento zásobník ukazuje a obsluhuje ho Stack Pointer (SP). K datům na SRAM jde lehce přistupovat skrz pět různých adresních módů podporovaných AVR architekturou. Místa v paměti SRAM a obecně ve všech pamětech AVR architektury jsou lineární paměťové mapy, která je rozkouskovaná na dílce a každý dílec má svou vlastní adresu. V paměti SRAM jsou uloženy soubory registrů, I/O data a interní data, do kterých lze ukládat data např. při přerušení atd. Paměť je adresována od nuly, přičemž prvních 95 adres mají zabrány soubory registrů a I/O registry. Dalších 512 adres může být použito pro ukládání dat. [4]

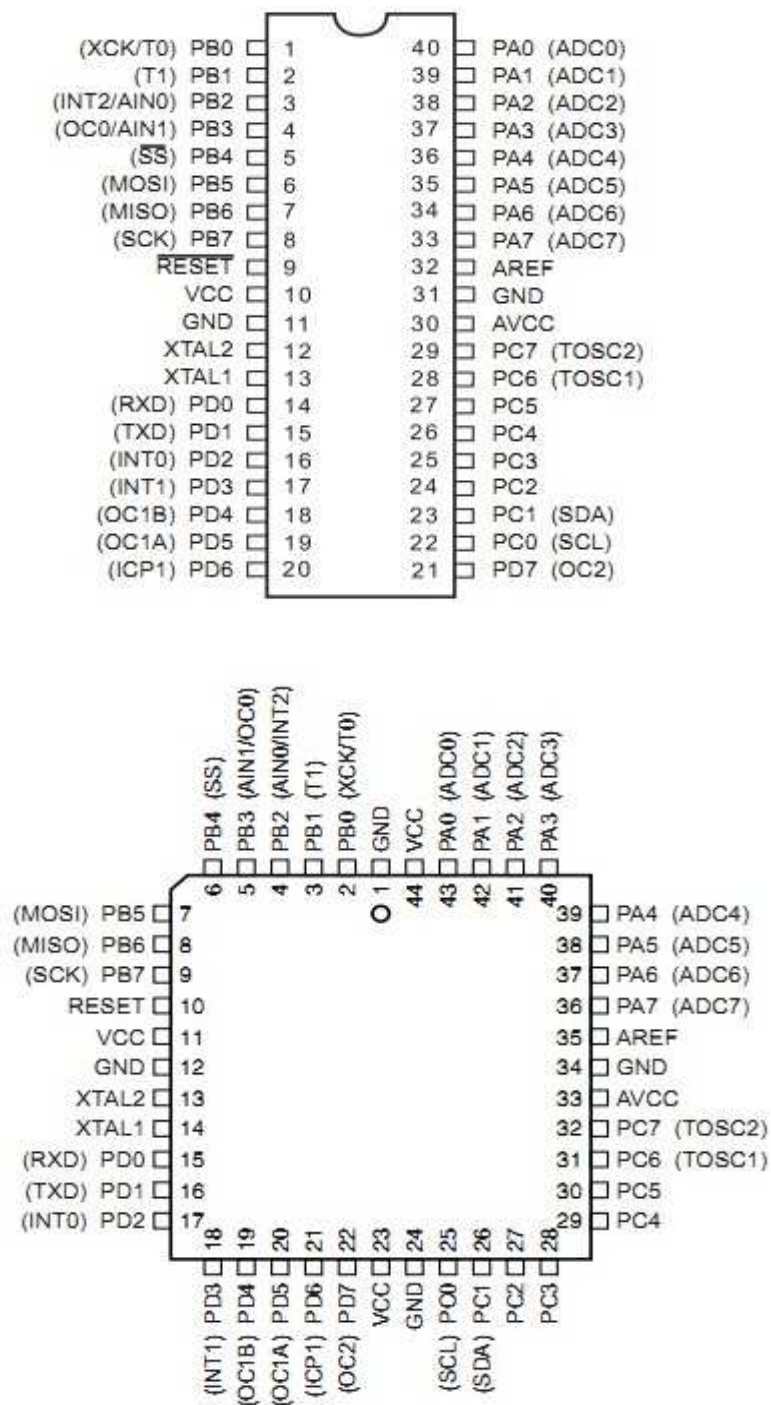


Obr. č. 7 Paměť SRAM, převzato a upraveno z [4]

Díky flexibilnímu modulu přerušení můžeme přerušit chod programu. V tomto modulu je obsažena tabulka s prioritami přerušení, přičemž jsou seřazeny od největší priority po nejmenší. [4]

I/O paměťový prostor (I/O module 1-3) obsahuje šedesát čtyři adres pro periferní funkce CPU. V tomto prostoru jsou obsaženy řídicí registry, SPI a ostatní I/O funkce. K I/O paměti v případě potřeby může být neustálý přímý přístup. Také může být použito globální přerušení, jehož bit je obsažen ve stavovém registru. [4]

5.5 Konfigurace pinů ATmega8535



Obr. č. 8 Konfigurace pinů ATmega8535, převzato a upraveno z [4]

ATmega8535 je 8-bitový CMOS mikrokontroler s nízkou spotřebou energie založený na upravené AVR RISC architektuře, schopný vykonávat instrukce na jeden hodinový takt. ATmega8535 dosahuje průchodnosti blížící se jednomu milionu instrukcí za sekundu (1 MIPS) při taktovací frekvenci 1GHz, což umožňuje kvalitní kompromis mezi spotřebou energie a rychlostí zpracovávání dat. [4]

3.6 Popis jednotlivých pinů a jejich funkce

Vcc - Napájecí napětí 4,5V - 5,5V

AVcc - Napájecí napětí pro A/D převodník na Portu A může být připojen na Vcc přímo, i když se zrovna A/D převodník nepoužívá. Pokud je A/D převodník použitý, přivedeme napájecí napětí z Vcc přes filtr LC. [4]

GND - Zem

XTAL1 - Invertující vstup operačního zesilovače oscilátoru a vstup do obvodu interních hodin. [4]

XTAL2 - Neinvertující výstup operačního zesilovače oscilátoru. [4]

Aref - AREF je referenční napájecí pin pro A/D převodník 2,5V. [4]

Port A (PA7..PA0) - Port A slouží jako analogový vstup do A/D převodníku. Port A slouží také jako 8-bitový obousměrný I/O port, pokud není zrovna používán A/D převodník. Každý pin obsahuje interní pull-up rezistor, které umožňují používat Port A jako I/O. Pokud jsou piny P0 až P7 nastaveny jako výstupní, pull-up rezistory jsou automaticky vypnuty. Pokud jsou piny P0 až P7 nastaveny jako vstupní a je na některý přivedena log.1, stávají se zdrojem proudu a pull-up rezistory jsou automaticky připojeny. Piny Portu A mají tedy tři stavy, pokud počítáme také stav vysoké impedance, který je funkční, i když neběží hodiny mikrokontroléru. [4]

Port B (PB7..PB0) - Port B může být použit jako 8-bitový obousměrný I/O port. Každý pin obsahuje interní pull-up rezistor, které umožňují používat Port B jako I/O. Pokud jsou piny P7 až P7 nastaveny jako výstupní, pull-up rezistory jsou automaticky vypnuty. Pokud jsou piny P0-P7 nastaveny jako vstupní a je na ně přivedená log.1, stávají se zdrojem proudu a pull-up rezistory jsou automaticky připojeny. Piny portu B mají tedy tři stavy, pokud počítáme stav Hi-Z, který je funkční, i když neběží hodiny mikrokontroléru. Dále alternativní funkce pinu P7-P0. [4]

Tab. č. 1 Alternativní funkce pinů portu B, převzato a upraveno z [4]

| Port pin | Alternativní funkce |
|----------|-------------------------------------------------------------------------------|
| PB7 | SCK (Sériové hodiny) Sběrnice SPI |
| PB6 | MISO (Master Input/Slave Output) Sběrnice SPI |
| PB5 | MOSI (Master Output/Slave Input) Sběrnice SPI |
| PB4 | SS (SPI Slave select input) Sběrnice SPI |
| PB3 | AIN1 (Analogový komparátor záporný vstup) OC0 (časovač/čítač0) |
| PB2 | AIN0 (Analogový komparátor kladný vstup) INT2 (Externí přerušení 2 vstupu) |
| PB1 | T1 (časovač/čítač1 vstup externího čítače) |
| PB0 | T0(časovač/čítač0 vstup externího čítače) XCK (USART externí hodiny I/O) |

- **SCK - PortB, Bit7**

SCK: Master Clock výstup, Slave Clock vstup pro kanál SPI. Pokud je SPI nastavena jako Slave, tento pin je nastaven jako vstup bez ohledu na nastavení DDB7. Jestliže je kanál SPI nastaven jako Master směr toku dat, je řízen bitem z registru DDB7. Pokud je pin PB7 nastaven SPI sběrnici jako vstupní, pull-up rezistory mohou být stále řízeny PORTB7. [4]

- **MISO - PortB, Bit6**

MISO: Master Data vstup, Slave Data výstup pro kanál SPI. Jestliže je SPI nastavena jako Master, tento pin je nastaven jako vstup bez ohledu na nastavení DDB6. Jestliže je SPI nastavena jako Slave, je směr dat řízen bitem z registru DDB6. Pokud je pin PB6 nastaven sběrnici jako vstupní, pull-up rezistory mohou být stále řízeny bitem PORTB7. [4]

- **MOSI - PortB, Bit5**

MOSI: Master Data výstup. Slave Data vstup pro kanál SPI. Jestliže je SPI nastavena jako Slave, tento pin je nastaven jako vstupní bez ohledu na nastavení DDB5. Jestliže

je SPI nastavena jako Master, je směr dat řízen bitem z registru DDB5. Pokud je pin PB5 nastaven sběrnici SPI jako vstupní, pull-up rezistory mohou být stále řízeny bitem PORTB5. [4]

- **SS-Port B, Bit 4**

SS: Slave vybere vstup. Pokud je SPI nastavena jako Slave, tento pin je nastaven jako vstupní bez ohledu na nastavení DDB4. Jako Slave, SPI je aktivována pokud je tento pin stažen na low (log.0). Jestliže je SPI nastavena jako Master, směr dat je řízen bitem z registru DDB4. Pokud je pin nastaven SPI jako vstupní, pull-up rezistory mohou být stále řízeny bitem PORTB4. [4]

- **AIN1/OC0-Port B, Bit3**

AIN1: Analogový komparátor, záporný vstup. Konfiguruje pin jako vstup s vypnutým interním pull-up rezistorem, aby nebyla ovlivňována funkce analogového komparátoru.

OC0: Porovnává výstup s požadovaným výstupem. Pin PB3 může sloužit také jako externí výstup pro Timer/Counter0. Jestliže má být PB3 konfigurován jako výstup, bit DDB3 musí být nastaven na log.1, aby byla povolena tato funkce. OC0 pin je také výstup pro PWM mode Timer funkci. (Fázový Modulátor správné šířky impulsu). [4]

- **AIN0/INT2-Port B, Bit2**

AIN0: Analogový komparátor, kladný výstup. Konfiguruje pin jako výstup s vypnutým interním pull-up rezistorem, aby nebyla ovlivňována funkce analogového komparátoru.

INT2: vnější zdroj přerušení. PB2 pin může sloužit také jako externí zdroj přerušení do MCU. [4]

- **T1-PortB, Bit1**

T1: Timer/Counter1 Counter Source. [4]

- **T0/XCK-PortB, Bit0**

T0: Timer/Counter0 Counter Source.

XCK: USART Externí hodiny. Registr směru dat (DDB0) řídí, zda jsou hodiny jako výstup (DDB0 set) nebo vstup (DDB0 cleared). XCK pin je aktivní pouze tehdy jsou-li USART operace v synchronním modu. [4]

Na Port B můžeme připojit například ISP sběrnici (In system programming), která využívá SPI (Serial peripheral interface), což je sériový protokol pro přenos dat. ISP

umožňuje například programování mikrokontroléru přímo na desce, aby jsme při programování nemuseli mikrokontrolér z desky vyjmát a vkládat ho do programátoru. ISP používá pouze piny MISO, MOSI, SCK. Samozřejmě také napájení Vcc, zem GND a RESET.

Port C (PC7..PC0) Port C může být použit jako 8-bitový obousměrný I/O port. Každý pin obsahuje interní pull-up rezistor, které umožňují používat Port C jako I/O. Pokud jsou piny P0-P7 nastaveny jako vstupní a je na ně přivedená log.1, stávají se zdrojem proudu a pull-up rezistory jsou automaticky připojeny. Piny portu B mají tedy tři stavy, pokud počítáme stav vysoké impedance, který je funkční, i když neběží hodiny mikrokontroléru. [4]

Port C, respektive některé jeho piny, mají i jiné využití a poskytují další možnosti připojení jiných rozhraní a zařízení. Alternativní funkce pinů Portu C jsou zobrazeny v následující tabulce. [4]

Tab. č. 2 Alternativní funkce pinů portu C, převzato a upraveno z [4]

| Port Pin | Alternativní Funkce |
|----------|------------------------------------------|
| PC7 | TOSC2 (Časovač oscilátor Pin 2) |
| PC6 | TOSC1 (Časovač oscilátor Pin1) |
| PC1 | SDA (Two-wire Sériová sběrnice Data I/O) |
| PC0 | SCL (Two-wire Sériová sběrnice Hodiny) |

- **TOSC2 - Port C, Bit7**

TOSC2: Timer Oscillator pin 2: Jestliže je AS2 bit v ASSR registru nastaven na log.1, je zapnut Timer/Counter2, pin PC7 je odpojen od portu a stává se invertujícím výstupem zesilovače oscilátoru. V tomto módu je krystal oscilátoru připojen na pin a pin nemůže být používán jako I/O. [4]

- **TOSC1 - Port C, Bit6**

TOSC1: Timer Oscilator pin1: Jestliže je AS2 bit v ASSR registru nastaven na log.1, je zapnut Timer/Counter2, pin PC6 je odpojen od portu a stává se neinvertujícím vstupem

zesilovače oscilátoru. V tomto módu je krystal oscilátoru připojen na pin a pin nemůže být používán jako I/O. [4]

- **SDA - Port C, Bit1**

SDA: Two-wire Serial Interface Data. Jestliže je TWEN bit v TWCR registru nastaven na log.1, je zapnut Two-wire Serial Interface. Pin PC1 je odpojen od portu a stává se vstupně výstupním I/O sériovým rozhraním pro Two-wire Serial Interface. V tomto módu je aktivován na pinu filtr, který potlačuje vstupní signály kratší než 50ns. Pokud je pin používán jako Two-wire Serial interface, pull-up rezistory mohou být stále řízeny PORTC1 bitem. [4]

- **SCL PortC, Bit0**

SCL: Two-wire Serial Interface Clock: Jestliže je TWEN bit v TWCR registru nastaven na log.1, je zapnut Two-wire Serial Interface. Pin PC0 je odpojen od portu a stává se vstupně výstupním I/O Serial Clock pinem pro Two-wire Serial Interface. V tomto módu je aktivován na pinu filtr, který potlačuje vstupní signály kratší než 50ns. Pokud je pin používán jako Two-wire Serial interface, pull-up rezistory mohou být stále řízeny PORTC0 bitem. [4]

K tomuto portu můžeme připojit například 1-Wire nebo I2C sběrnice, které jsou v dnešní době velice populární jak pro běžné uživatele, tak např. pro automobilový průmysl z důvodu komunikace po velice malém počtu vodičů. Např. u 1-Wire po jednom, po kterém běží jak data tak hodiny, což je vykoupeno do jisté míry složitějším komunikačním algoritmem než u I2C, který je z mého pohledu jednodušší, nicméně používá pro komunikaci o 1 vodič více pro hodiny atd. Po těchto sběrnících komunikuje veliké množství zařízení, např. teplotní čidla, mikrokontroléry, LCD, paměti a další.

Port D (PD7..PD0) Port D může být použit jako 8-bitový obousměrný I/O port. Každý pin obsahuje interní pull-up rezistor, které umožňují používat Port C jako I/O. Pokud jsou piny P0-P7 nastaveny jako vstupní a je na ně přivedená log.1, stávají se zdrojem proudu a pull-up rezistory jsou automaticky připojeny. Piny portu D mají tedy tři stavy, pokud počítáme stav vysoké impedance, který je funkční, i když neběží hodiny mikrokontroléru. [4]

Port D má i jiné využití a poskytuje další možnosti připojení jiných rozhraní a zařízení. Alternativní funkce pinů Portu D jsou zobrazeny v následující tabulce. [4]

Tab. č. 3 Alternativní funkce pinů portu D, převzato a upraveno z [4]

| Port Pin | Alternativní funkce |
|----------|----------------------------------|
| PD7 | OC2 (časovač/čítač2) |
| PD6 | ICP1 (časovač/čítač1 vstup) |
| PD5 | OC1A (časovač/čítač1) |
| PD4 | OC1B (časovač/čítač) |
| PD3 | INT1 (externí přerušení 1 vstup) |
| PD2 | INT0 (externí přerušení 0 vstup) |
| PD1 | TXD (USART výstup) |
| PD0 | RXD (USART vstup) |

- **OC2 - Port D, Bit7**

OC2: Timer/Counter2. Výstup porovnává s požadovanou hodnotou na výstupu. Pin PD7 může sloužit jako externí výstup pro Timer/Counter2. Pin bude konfigurován jako výstup pro tuto funkci, pokud bit DDD7 bude nastaven jako log.1. Pin PD7 může sloužit také jako PWM Timer funkce. (Fázový modulátor správné šířky impulzu). [4]

- **ICP1 - Port D, Bit6**

ICP1: Input Capture Pin: Pin PD6 může zabrat pin jako vstup pro Timer/Counter1 [4]

- **OC1A - Port D, Bit5**

OC1A: Timer/Counter1. Výstup porovnává s požadovanou hodnotou A na výstupu. Pin PD5 může sloužit také jako externí výstup pro Timer/Counter1. Pin je konfigurován pro tuto funkci jako výstup, pokud je bit DDD5 nastaven jako log.1. Pin PD5 může sloužit také jako PWM Timer funkce. (Fázový modulátor správné šířky impulzu). [4]

- **OC1B - Port D, Bit4**

OC1B: Timer/Counter1. Výstup porovnává s požadovanou hodnotou B na výstupu. Pin PD4 může sloužit také jako externí výstup pro Timer/Counter1. Pin je konfigurován pro tuto funkci jako výstup, pokud je bit DDD4 nastaven jako log.1. Pin PD4 může sloužit také jako PWM Timer funkce. (Fázový modulátor správné šířky impulzu). [4]

- **INT1 - Port D, Bit3**

INT1: External Interrupt Source 1. Pin PD3 může sloužit jako vnější zdroj přerušení. [4]

- **NIT0 - Port D, Bit2**

INT0: External Interrupt Source 0. Pin PD2 může sloužit jako vnější zdroj přerušení. [4]

- **TXD - Port D, Bit1**

TXD: Transmit Data. (výstupní data z mikrokontroléru po USART) Jestliže je zapnut přenos dat po USART, tento pin je nastaven jako výstupní bez ohledu na hodnotu bitu DDD1. [4]

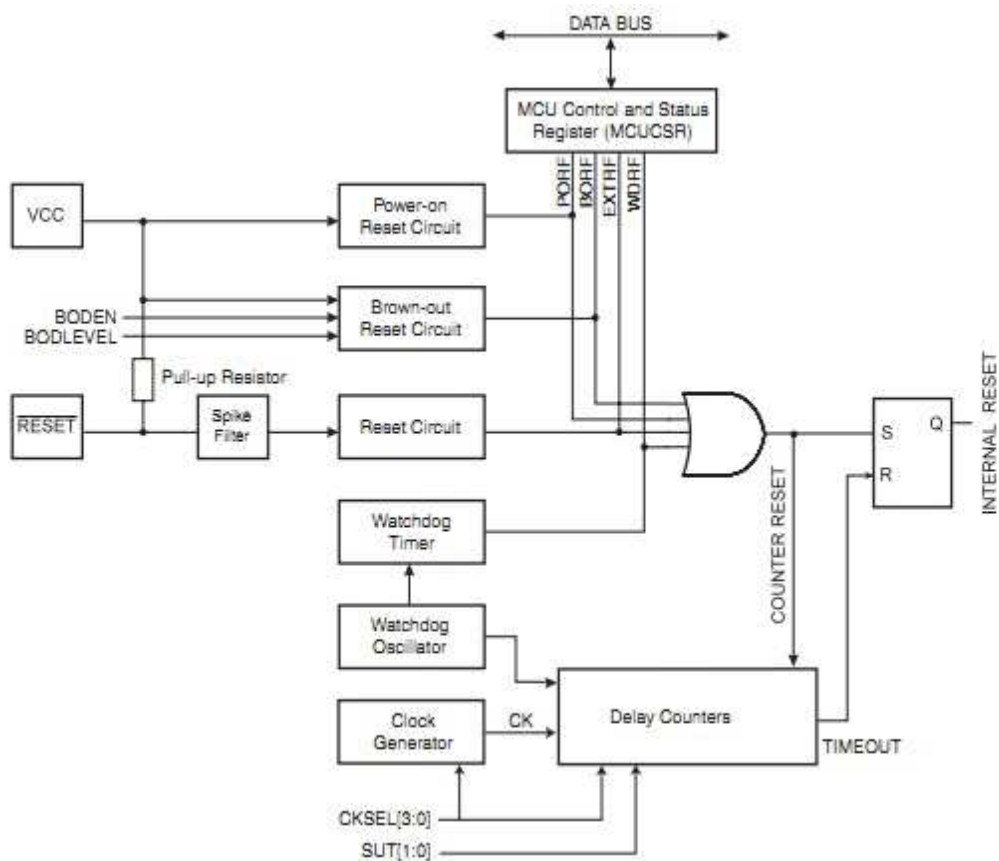
- **RXD - Port D, Bit0**

RXD: Receive Data. (Vstupní data do mikrokontroléru po USART) Jestliže je zapnut přenos dat po USART, tento pin je nastaven jako vstupní, bez ohledu na hodnotu bitu DDD0. Pokud je tento pin nastaven jako vstupní, pull-up rezistor může být stále řízen bitem PORTD0. [4]

Na tento pin můžeme připojit například sériovou linku RS232 nebo LPT, pokud potřebujeme mikrokontrolér řídit přímo z PC. Těmito rozhraními jsou vybaveny starší typy PC, nebo můžeme použít redukci USB to RS232, kterou lze zakoupit i s příslušnými ovladači pro jednotlivé typy windows. Popřípadě lze připojit rovnou USB, čímž odpadá použití redukce, ale řešení je poněkud složitější. Proto je stále hojně používaná sériová linka RS232.

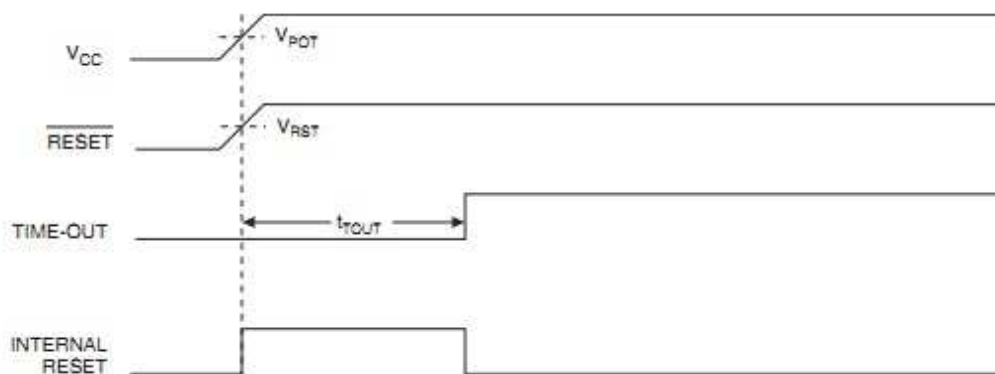
Reset - Vstup pro reset impuls. Aby byl impuls za reset impuls považován, musí být generován po určitý časový okamžik, musí mít určitou minimální délku a velikost působícího napětí musí být v daném rozsahu.. U kratších impulsů není garantováno, že vyvolají reset. Po reset pulzu jsou I/O registry nastaveny na danou inicializační hodnotu. Program se začne opět vykonávat od místa, ve kterém byl přerušen. Porty jsou nastaveny do stavu vysoké impedance. Reset pulsů je několik druhů. [4]

- Power-on reset
- External reset
- Watchdog reset
- Brown-out reset



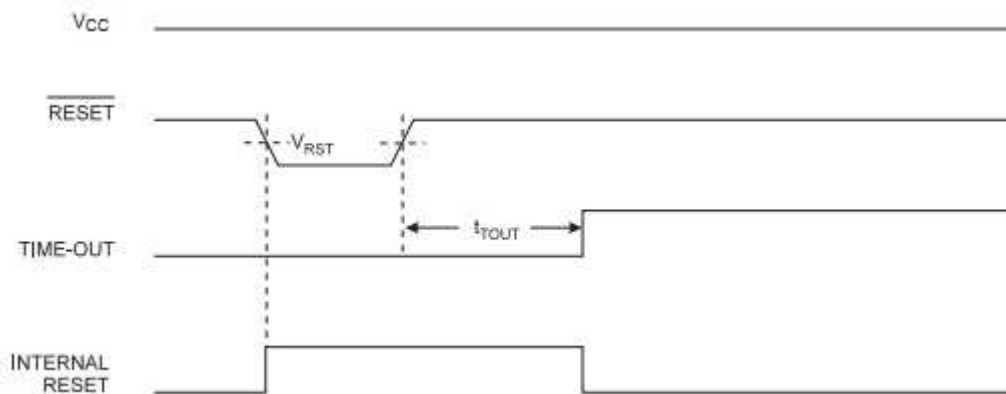
Obr. č. 9 Resetovací logika ATmega8535, převzato a upraveno z [4]

1. Power on reset (POR) - Puls je generován detekčním obvodem na chipu. Napěťové úrovně, potřebné pro reset puls, jsou zobrazeny v tabulce č. 4. Power on reset je aktivován, jestliže je napájecí napětí V_{cc} menší, než požadovaná úroveň napětí power on resetu. POR obvod spustí Start-up reset při připojení napětí, nebo když zjistí kolísání, nebo selhání napájecího napětí. [4]



Obr. č. 10 MCU Start-up RESET svázán s Vcc, převzato a upraveno z [4]

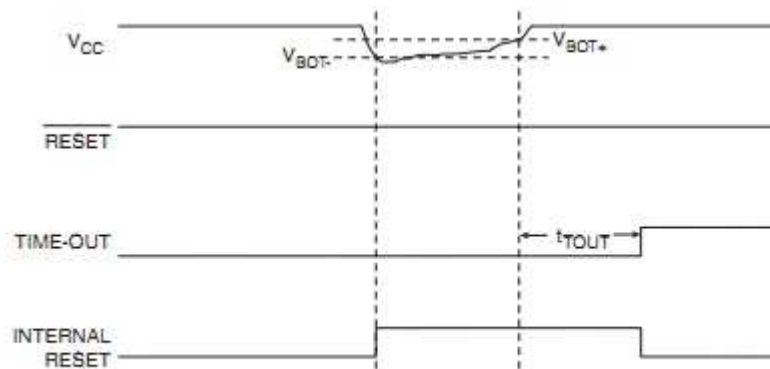
2. Externí reset - je generován i velmi malým napětím působícím na reset pin. Reset pulz, který má větší šířku impulzu než minimální - viz tabulka č. 4 - generuje reset, i když nejsou momentálně funkční hodiny. Kratší impulsy negarantují reset. Když reset signál dosáhne Reset Treshold Voltage (V_{RST}), zpožďovací obvod (delay counter) spustí MCU poté co vyprší Time-out perioda (t_{TOUT}). [4]



Obr. č. 11 Externí reset v průběhu operace, převzato a upraveno z [4]

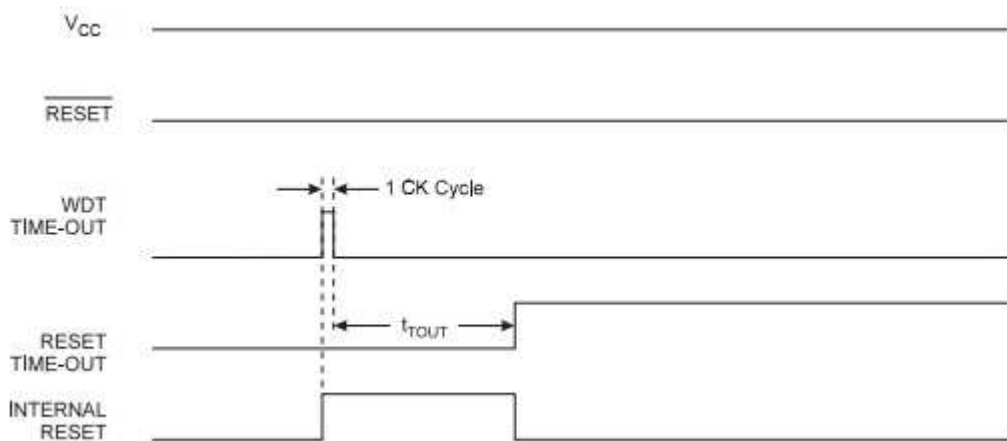
3. Brown-out Detection - ATmega8535 má na chipu zabudovaný obvod, tzv. Brown-out detektor (BOD). Tento obvod monitoruje napájecí napětí V_{cc} v průběhu chodu programu a porovnává V_{cc} se zvoleným BOD napětím (BODLEVEL). BODLEVEL je nastaven buď jako log.1, nebo jako log.0 viz tabulka č.4. Brown-out detekce může být zapnuta nebo vypnuta. Pokud máme BOD detekci zapnutou a V_{cc} klesne pod požadovanou hodnotu (V_{bot} - v obr. č 12), je okamžitě aktivován Brown-out

reset. Pokud opět V_{CC} vzroste nad hodnotu (V_{BOT+} v obr. č. 12), zpožd'ovací obvod (delay counter) spustí MCU po vypršení TIME-OUT (t_{TOUT}) periody. BOD obvod detekuje pokles V_{CC} pouze tehdy, když napětí, které kleslo pod V_{CC} , zůstane pod jeho úrovní delší dobu, než je označená v tabulce č. 4. [4]



Obr. č. 12 Brown-out detekce, převzato a upraveno z [4]

4. Watchdog Reset - Watchdog, neboli hlídací pes, může generovat reset impuls také. Po dobu, která je mu nastavena pouze čeká a odpočítává čas. Jakmile čas, který byl nastaven uplyne, watchdog nastaví WDT TIMEOUT linku do log.1, kterou vzápětí stáhne do log.0, od toho času po uplynutí (t_{TOUT}) zpožd'ovací obvod spustí MCU. [4]



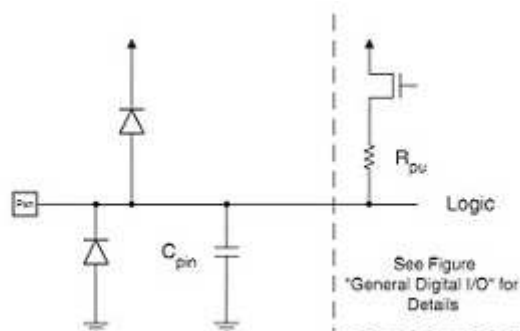
Obr. č. 13 Watchdog reset, převzato a upraveno z [4]

Tab. č. 4 Charakteristiky reset pulsů, převzato a upraveno z [4]

| Symb. | Parametr | Nastavení | Min. | Typ. | Max. | Jedn. |
|----------|--------------------------------------|------------|------|------|------|-------|
| Vpot | Power-on RESET | | | 1,4 | 2,3 | V |
| Vrst | Externí RESET | | 0,2 | | 0,9 | V |
| Time RST | Min. šířka pulsu Externího RESETU | | 1,5 | | | μs |
| Vbot | Brown-out RESET | BODLEVEL=1 | 2,5 | 2,7 | 2,9 | V |
| | | BODLEVEL=0 | 3,6 | 4,0 | 4,9 | V |
| Time BOD | Brown-out RESET | BODLEVEL=1 | | 2 | | μs |
| | Det. doba poklesu Vcc | BODLEVEL=0 | | 2 | | μs |

5.7 Charakteristika práce portů

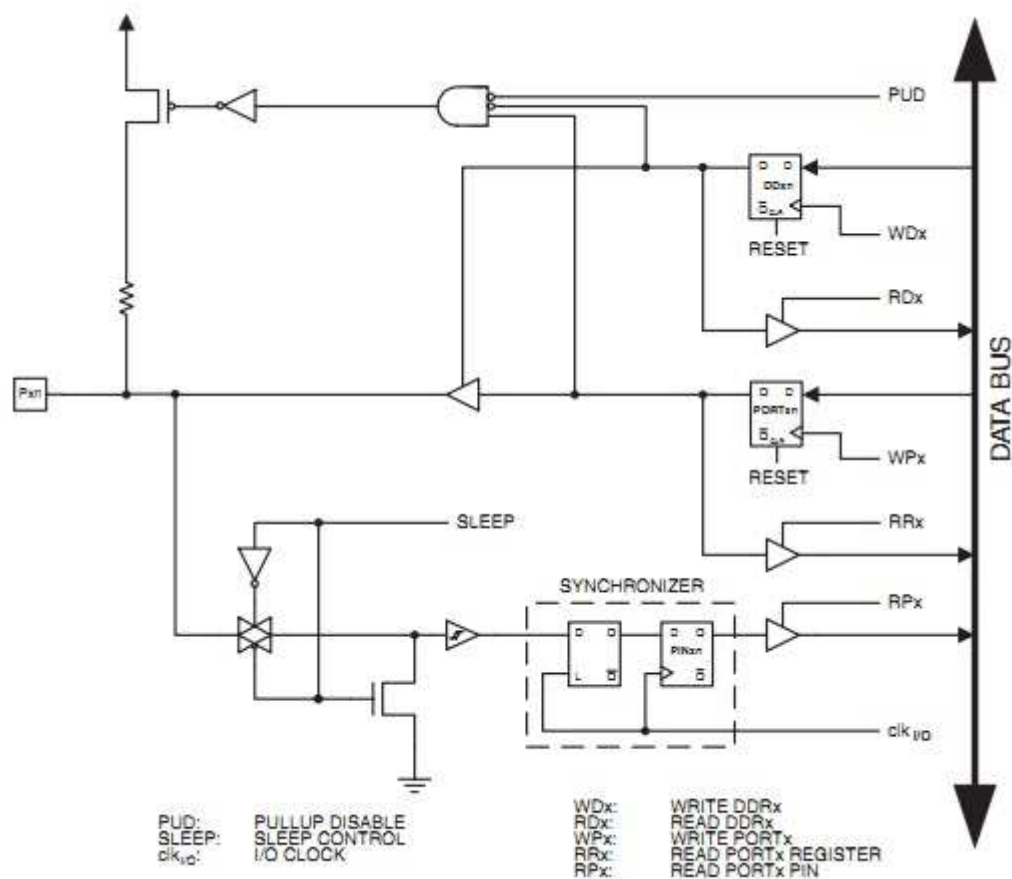
Všem AVR portům můžeme měnit funkci pro čtení a zápis, jestliže jsou používány jako hlavní digitální I/O porty. Tím je myšleno, že směr každého pinu portu (jestli je vstupní, nebo výstupní) se dá měnit bez toho, aby bylo nutné měnit směr ostatních pinů portu. Pokud je pin portu konfigurován jako výstup, pull-up rezistory jsou odpojeny, pokud je konfigurován jako vstup, pull-up rezistory mohou být připojeny nebo odpojeny. Mikrokontrolér poskytuje silný signál na výstupu pinu i na to, aby vybudil ledky, nebo display přímo bez použití dalšího budiče. Pull-up rezistory mají proměnný odpor od 20-50kΩ. Všechny piny jsou také opatřeny ochrannými diodami, např. proti napěťovým špičkám atd... [4]



Obr. č. 14 Znárodnění zapojení ochran pinu, převzato a upraveno z [4]

Na obrázku č. 14 vidíme schéma pinu Pxn. "X", označuje o jaký port se jedná (A, B, C, D) a n označuje o jaký se jedná pin (7-0) na daném portu. Proto pokud používáme registry, nebo definujeme bity v programu, musíme používat přesnou formu. PORTB3 například znamená, že se jedná o PORT B a o jeho 3. pin, obecně PORTxn. [4]

Pro každý port jsou v paměti lokalizovány tři registry. Data Register - PORTx, Data Direction Register - DDRx, a Port Inputs Pins - PINx. Data Register a Data Direction Register jsou typu Read/Write, zatímco Port Input Pin je pouze read. Kromě toho pull-up rezistory odpojuje PUD bit v SFIOR registru, který odpojí všechny rezistory na všech pinech, na všech portech, když je nastaven. [4]



Obr. č. 15 Obecné schéma pinu, převzato a upraveno z [4]

Bit DDxn v DDRx registru vybírá směr tohoto pinu. Jestliže je DDxn zapsán v registru jako log. 1, Pxn je nastaven jako výstupní, Pokud je DDxn zapsán v registru jako log. 0 potom je pin nastaven jako vstupní. [4]

Jestliže je PORTxn zapsán v registru jako log. 1 a pin je konfigurován jako vstupní, pull-up rezistory jsou připojeny. Pro vypnutí pull-up rezistorů musí být pin zapsán v registru jako log. 0, nebo musí být pin konfigurován jako výstupní. Piny portů mají tři stavy, pokud počítáme reset pulz, který může nastat i když hodiny zrovna neběží. [4]

Jestliže je PORTxn zapsán jako log. 1, když je pin konfigurován jako výstupní, na výstupu bude log. 1. Pokud PORTxn bude zapsán v registru jako log. 0 a pin bude konfigurován jako výstupní, na výstupu bude log. 0. [4]

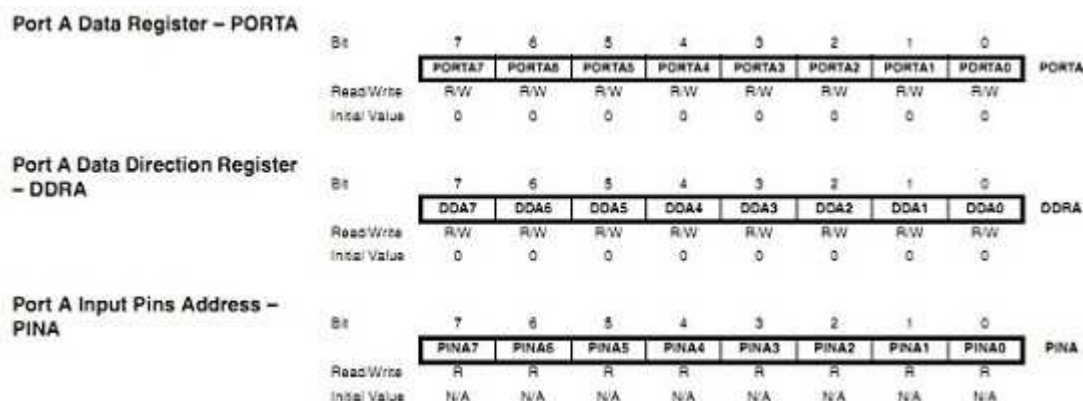
PUD bit v SFIOR registru, tento bit, pokud je nastaven na log. 1, odpojuje, jak už bylo řečeno, všechny pull-up rezistory na všech pinech všech portů. Proto může nastat ještě stav vysoké impedance a to tehdy, pokud je DDxn = 0 (pin nastaven jako vstup) PORTxn = 0 (Pull-up rezistory odpojeny), tato kombinace by neměla nikdy nastat, nezávisle na PUD bitu je na pinu stav vysoké impedance. Pokud je DDxn = 0 (vstupní režim) a PORTxn = 1 (pull-up rezistor připojen) a PUD bit nastaven v log. 1 (odpojuje všechny pull-up rezistory) nastává stejný případ jako předchozí. Na pinu je stav vysoké impedance. [4]

Stavy jednotlivých pinů můžeme shrnout v následující tabulce č. 5

Tab. č. 5 Suma možných signálu na pinu, převzato a upraveno z [4]

| DDxn | PORTxn | PUD | I/O | Pull-up | Result |
|------|--------|-----|--------|---------|-------------|
| 0 | 0 | X | Input | No | Hi-Z |
| 0 | 1 | 0 | Input | Yes | Source of I |
| 0 | 1 | 1 | Input | No | Hi-Z |
| 1 | 0 | X | Output | No | 0(Sink) |
| 1 | 1 | X | Output | No | 1(Source) |

Pro úplné pochopení fungování jednotlivých pinů portů a pochopení značení jednotlivých registrů, které piny portů používají, přikládám obrázek s registry pro port A.



Obr. č. 16 Registry portu A, převzato a upraveno z [4]

6 Výběr Softwaru pro návrh schémat a DPS

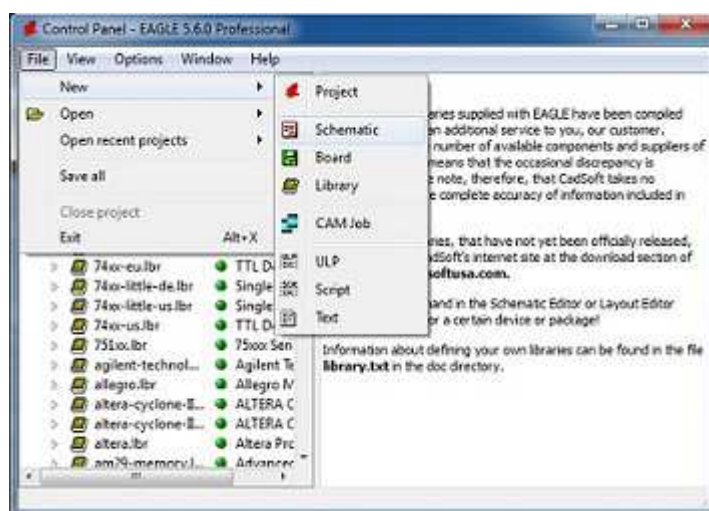
Pro návržení elektrického schématu výukové stavebnice s mikrokontrolérem AVR ATmega8535 bylo nutné vybrat vhodný software, který by umožňoval nejenom návrh schématu jako takového, ale v lepším případě i vytvoření plošného spoje. Také je nutné, aby měl daný software aktualizovaný seznam součástek, a aby jeho součástí byly mikrokontroléry AVR a jiné součástky, které budeme potřebovat používat. Softwarů, které by tohle splňovaly, je samozřejmě více například: Formica, Pads Logick 2005, Eagle, někdo navrhuje desky i v Autocadu atd. V Autocadu docela umím a nakreslil jsem v něm už více výkresů, ale i přes to jsem ho hned vyloučil, protože to není software primárně určený pro kreslení elektrických schémat a výrobu plošných spojů a myslím, že by to byla větší práce, než se naučit s jiným softwarem, který je na to vhodnější. Ze zbylé nabídky programů jsou vesměs všechny výborné, liší se různými věcmi, které pro mě nebyly až tak podstatné. Nejlepší by byl asi software Pads logick, což je vynikající nástroj na tvorbu schémat a DPS, dokonce dokáže simulovat dané obvody, řekl bych že je to nástroj na profesionální úrovni, nicméně stojí dost peněz a není poskytován ve free verzi. Na druhou stranu Eagle je poskytován pro běžného uživatele ve free verzi, která dovoluje použít desku o velikosti standardní eurodesky

160x100 mm, zhruba asi 200 součástek na výkres a routování plošného spoje je dovoleno ve free verzi dvouvrstvě. Eagle má také velikou podporu součástek. V klasické instalaci je takřka všechno na co si vzpomenete a pokud by náhodou některá součástka chyběla, dají se stáhnout knihovny, které seznam součástek rozšiřují. Myslím si ale, že pro běžné schémata to není potřebné, už takhle pro uživatele, který nemá katalog součástek v hlavě, bude těžké se v detailním výčtu součástek orientovat a zvolit ty správné s roztečemi, které potřebujeme. Toto je velice důležité, protože po vyleptání a vyvrtání desky by nám některé součástky zvláště ty, které mají více nohou, nemusely do desky jít.

6.1 Eagle 5.6.0

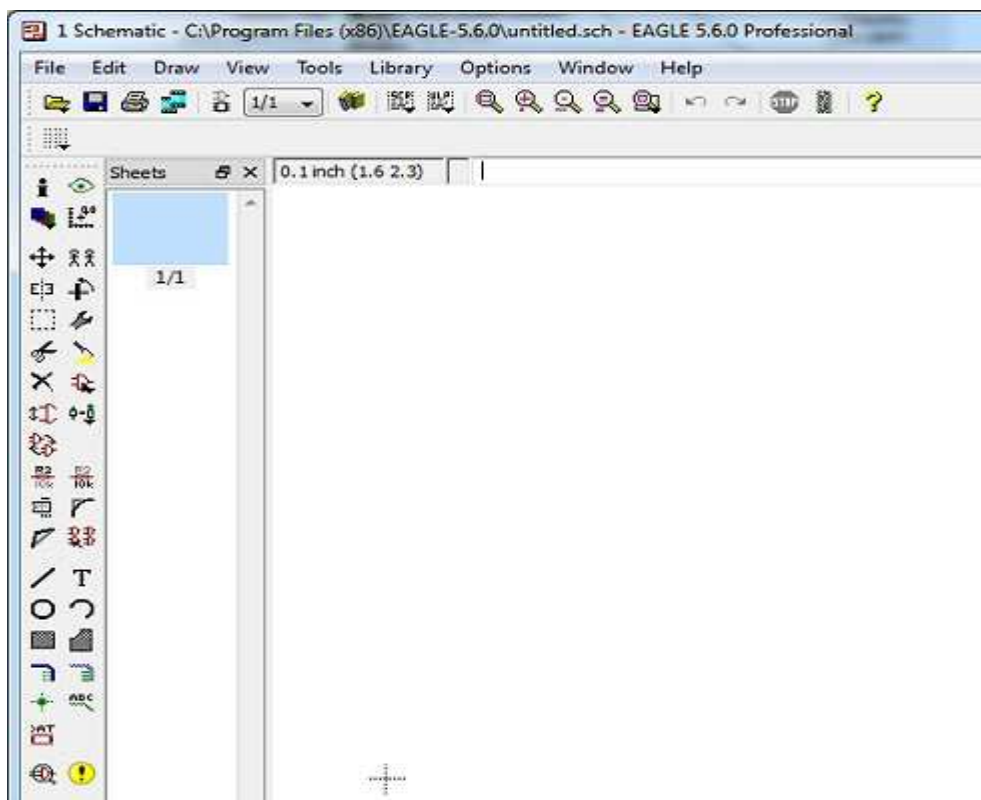
Eagle je ucelený systém programů pro návrh desek plošných spojů, jde o jeden z tak zvaných eCAD programů. Postupem času, jak se Eagle vyvíjel a byl z počátku projektem hlavně nadšených amatérů, v dnešní době představuje ucelený systém, který postačí potřebám zručného konstruktéra v oblasti elektroniky. [5]

Pár slov o filozofii a koncepci Eagle. Při práci musí být neustále spuštěn Control panel, který zajišťuje a hlídá konzistenci mezi schématem a deskou, přenáší změny ze schématu do desky, zajišťuje otevírání knihoven, obsluhuje schránku atd. V tomto prostředí zakládáme jednotlivé projekty, můžeme je doplňovat o popisy, přejmenovávat kopírovat, přesouvat atd. [5]



Obr. č. 17 Eagle 5.6.0 control panel

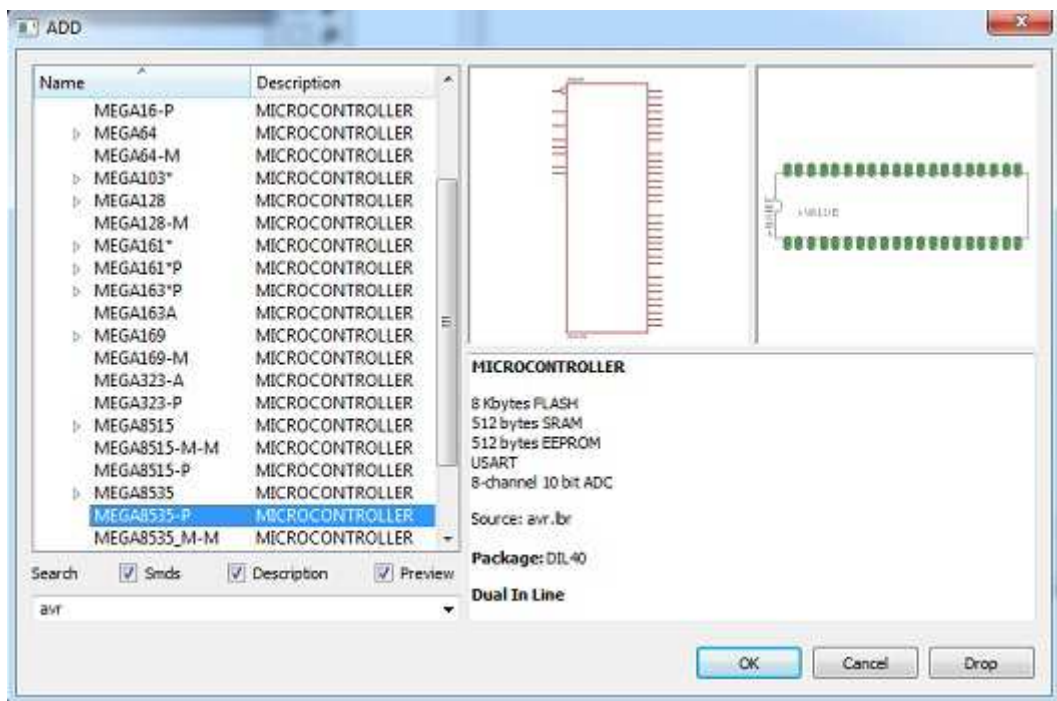
Eagle používá grafické intuitivní rozhraní podobné wordu a jiným, proto je práce s programem příjemná a není příliš složitá. V rámci editoru lze vybírat příkazy několika způsoby. Nejčastěji to bude kliknutím na odpovídající ikonu v panelu nástrojů, to však samozřejmě není jediná cesta. Instrukci lze vybrat dále jak z roletového menu, tak můžeme napsat příkaz, což později, když dostaneme program do ruky, asi nejrychlejší a nepohodlnější cesta. [5]



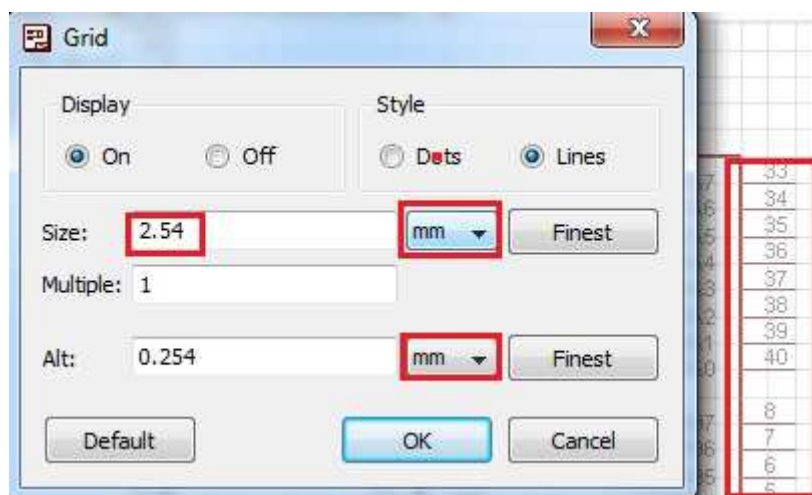
Obr. č. 18 základní popis prostředí Eagle, převzato a upraveno z [6]

Nejdůležitější příkaz, který budeme asi nejvíce a hlavně používat je příkaz "add", který se píše do příkazového řádku. Viz obrázek č. 18. Tento příkaz vyvolá tabulku se součástkami Obr. č. 19, ze které vybereme potřebné součástky v nejrůznějších pouzdrech a provedeních a přetáhneme je na pracovní plochu. Jednotlivé součástky se spojují vodiči wire. Obr. č. 18 (čára). Pokud vytváříme schéma s mikrokontroléry, za zmínku stojí používání sběrnice bus vis obrázek č. 18 (sběrnice), díky které si ušetříme mnoho času a schéma se stává přehlednější díky menšímu počtu vodičů. Důležité ovšem je v tomto případě neustále kontrolovat, jestli přivádíme

vstupní vodiče do sběrnice na správné výstupní vodiče ze sběrnice, jinak by nám neseděly "cesty" při routování desky. Také je důležité kontrolovat připojení jednotlivých součástek k obvodu, protože se někdy stává, že součástka nemá jeden konec připojen, i když jsme přesvědčeni že má. Tyto kontroly nám ušetří ve výsledku mnoho času, protože nebudeme muset zdlouhavě hledat chyby ve schématu.



Obr. č. 19 Tabulka se součástkami vyvolaná příkazem add



Obr. č. 20 Nastavení rastru

Poslední důležitá připomínka je nastavení rastru. Viz obrázek č. 20 Zde především nastavujeme jednotky, které používáme, dále zobrazení mřížky, do které součástky vkládáme a rozteč bodů, který je definován parametrem size. Velice jemný rastr by zhoršoval viditelnost součástek a přehlednost schématu a hodně hrubý rastr by postrádal využitelnost, defaultní nastavení, které se většinou používá je běžná rozteč nožiček chipů 2,54 mm. [5]

K tomu, abychom mohli začít vytvářet schémata, detailnější popis programu nepotřebujeme. Já osobně dávám přednost tomu, že si zjistím základní informace k tomu, abych mohl začít s programem pracovat a v průběhu práce řeším vniklé problémy na které narazím, popřípadě dohledávám řešení. Nepotřebuji přečíst knihu o Eaglu, abych s nim začal pracovat, stačí ji mít k ruce jako příručku.

7 Návrh El. Schémat a DPS

Pro splnění diplomové práce bylo za potřebí navrhnout stovebnici s mikrokontrolérem AVR Atmel, která bude obsahovat procesorovou desku a přídatné moduly, které se k desce budou připojovat. Byl kladen důraz na to, aby bylo možné v budoucnu k procesorové desce připojit defakto libovolné zařízení, které po naprogramování bude s deskou komunikovat. Proto byla potřeba vymyslet, jak desku navrhnout, aby tento hlavní požadavek splňovala.

7.1 Procesorová deska

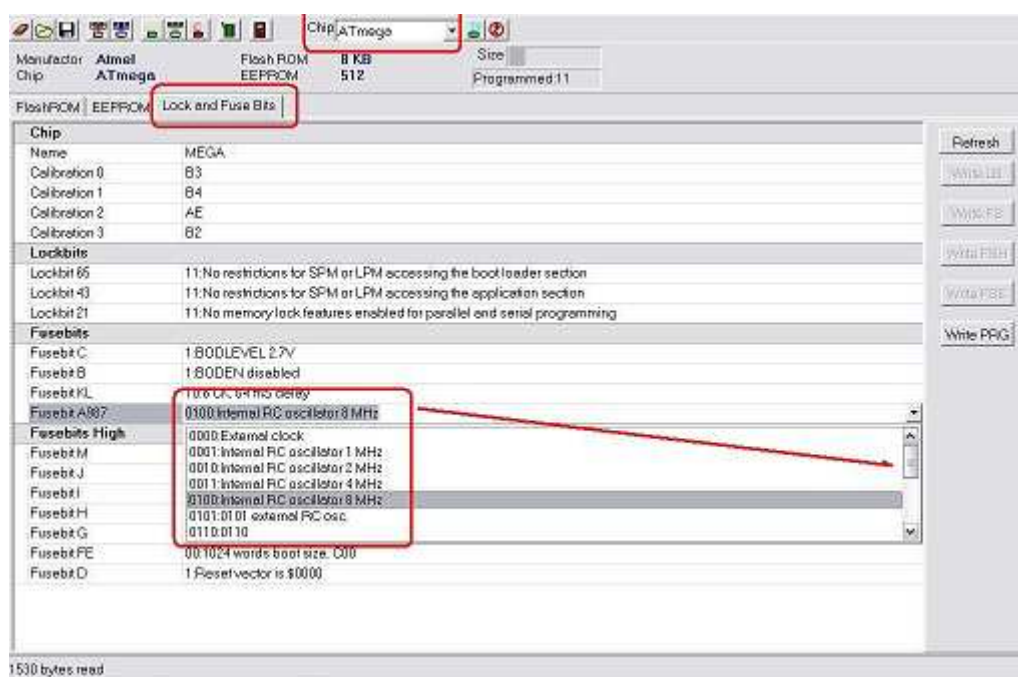
Jako první jsem se rozhodl vyřešit nejdůležitější a nejtěžší problém co se hardwaru týká. Navrhnout procesorovou desku, která bude využívat většiny I/O možností mikrokontroléru ATmega8535. Proto deska musí mít vývody na všechny čtyři I/O porty mikrokontroléru, musí obsahovat možnost připojení ISP (in system programming) sběrnice, která využívá SPI (serial peripheral interface) což mikrokontrolér ATmega8535 podporuje, také by bylo dobré, kdyby deska obsahovala možnost připojení RS232 pro komunikaci s PC.

Mimoto deska obsahuje také stabilizátor napájecího napětí na 5V, který je možné v případě, že máme regulovatelný kvalitní zdroj odpojit a napájet desku rovnou ze zdroje. Pokud zdroj není k dispozici, obvod se stabilizátorem napětí na 5V je možné

připojit a napájet se bude např. z 9-12V stejnosměrného trafo adaptéru, který je možné koupit v jakékoliv elektro prodejně.

Pro externí reset desky je na desce implementován resetovací obvod s tlačítkem, díky kterému je možné desku bezproblémově a rychle resetovat stiskem tlačítka.

Deska obsahuje svůj vůči mikrokontroléru externí 10MHz krystal, který je možné uvést v případě potřeby do chodu tak, že se přímo v mikrokontroléru nastaví ve Fuse bitech používat externí krystal, který je defaultně zakázán a mikrokontrolér využívá svůj interní 1MHz krystal, který nám ale na ukázkové programy postačí.



Obr. č. 21 Změna Fuse bitů u ATmega8535

Jak je z obrázku č. 21 vidět, ve Fuse bitech se toho dá nastavovat mnohem více, ne jen oscilátor, například napětí BODLEVEL, což je úroveň Brown-out Detector, který je používán pro Brown -out reset. Tento reset je popsán podrobně na straně 20-21. Ovšem při nastavování Fuse bitů musíme být opatrní, protože si při nedbalém manipulování bity můžeme práci místo ulehčení i ztížit, ne-li mikrokontrolér zničit. Například mě nedošlo, že když nastavím externí oscilátor u mikrokontroléru, jeho interní je odpojen a nebude pracovat. Což by nebyl takový problém, pokud bych chtěl

mikrokontrolér jednou naprogramovat a nechat ho konat program. Jenže v mém případě, když ho programuji několikrát v časovém horizontu pár minut, je docela na obtíž spolu s mikrokontrolérem do programátoru neustále zapojovat externí oscilátor a neustále přepojovat propojky. Bez oscilátoru samozřejmě mikrokontrolér naprogramovat nejde. V momentě, když jsem nastavil zpět interní oscilátor ve Fuse bitech, už mikrokontrolér nekomunikoval vůbec.

7.1.1 Použité součástky

Kondenzátory:

C1, 10 μ F 2x
C2, 100nF 4x
C3, 470 μ F 1x
C4, 47 μ F 1x
C5, 33pF 2x
C6, 1 μ F 4x

Rezistory:

R1, 10K Ω 1x
R2, 100 Ω 1x

Cívky:

L1, 100 μ H 1x

Konektory:

con-phoenix-254 7x
con-ml-10pin 6x
con-avr-isp6 2x

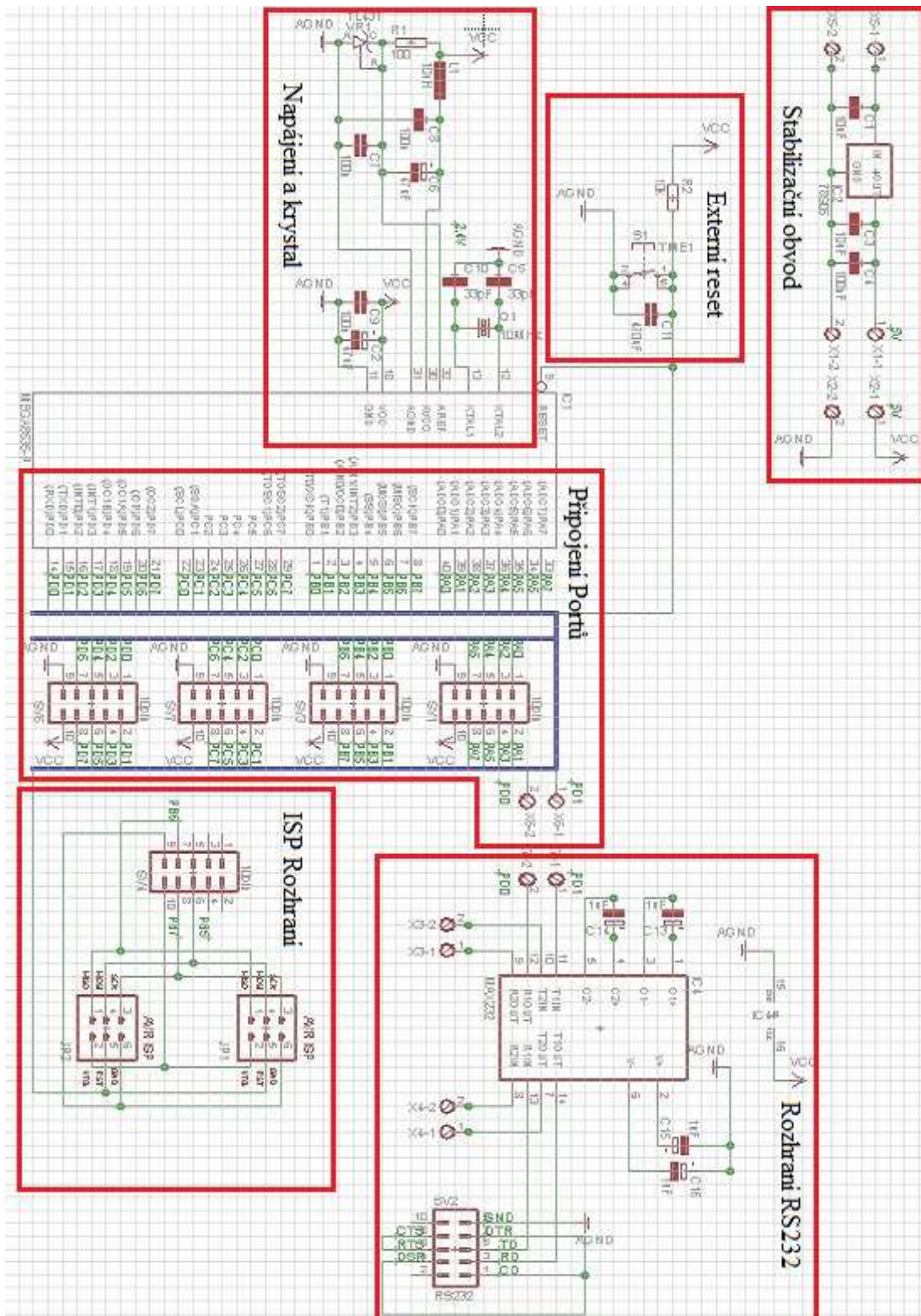
Integrované Obvody:

IC1: ATmega8535 1x, Package DIL40
IC2: 78S05 1x, Package 78XXS
IC3: MAX232 1x, Package DIL16
IC4: V-reg TL431CLP 1x, Package TO92CLP

Ostatní:

Krystal Q1: 10MHz 1x, Tlačítko S1: Switch-misc TME1 1x

7.1.2 Schéma procesorové desky

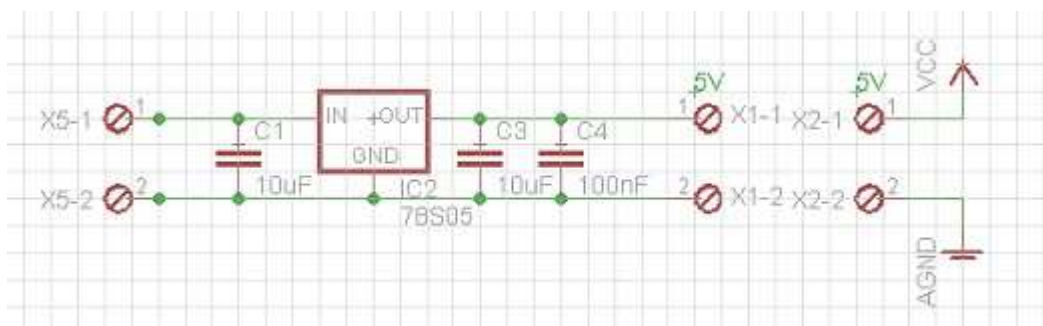


Obr. č. 22 Schéma procesorové desky

Na obrázku č. 22 vidíme schéma procesorové desky, které jsem kvůli přehlednosti a lepší možnosti popsání rozdělil do jednotlivých bloků, které bude lepší a jednodušší popsat samostatně.

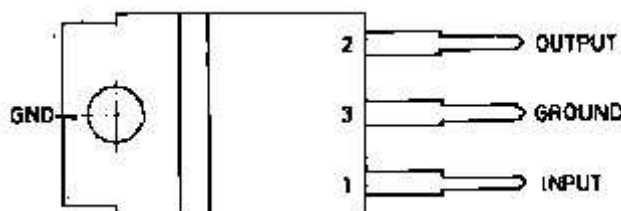
Samotné schéma bylo několikrát upravováno a opravováno, než vznikla tato finální podoba, která by už měla být bez chyb. Samotné navržení a nakreslení schématu trvalo poměrně dlouhou dobu, z důvodu toho, že jsem před tím nikdy s programem Eagle nepracoval, takže jsem se setkával s překážkami které jsem se musel naučit řešit. Také bylo za potřebí se naučit využívat pokročilejší funkcí Eagle, kvůli tomu aby výsledné schéma bylo na pohled elegantnější i když funkčně stejné jako předchozí. Mám namysli hlavně používání sběrnic atd.

7.1.3 Stabilizační obvod



Obr. č. 23 Stabilizační obvod

Stabilizační obvod je na desce implementován z důvodu napájení. Stabilizační obvod obsahuje integrovaný obvod, stabilizátor napětí 78S05 značen ve schématu jako IC2.



Obr. č. 24 Stabilizátor napětí 78S05, převzato a upraveno z [7]

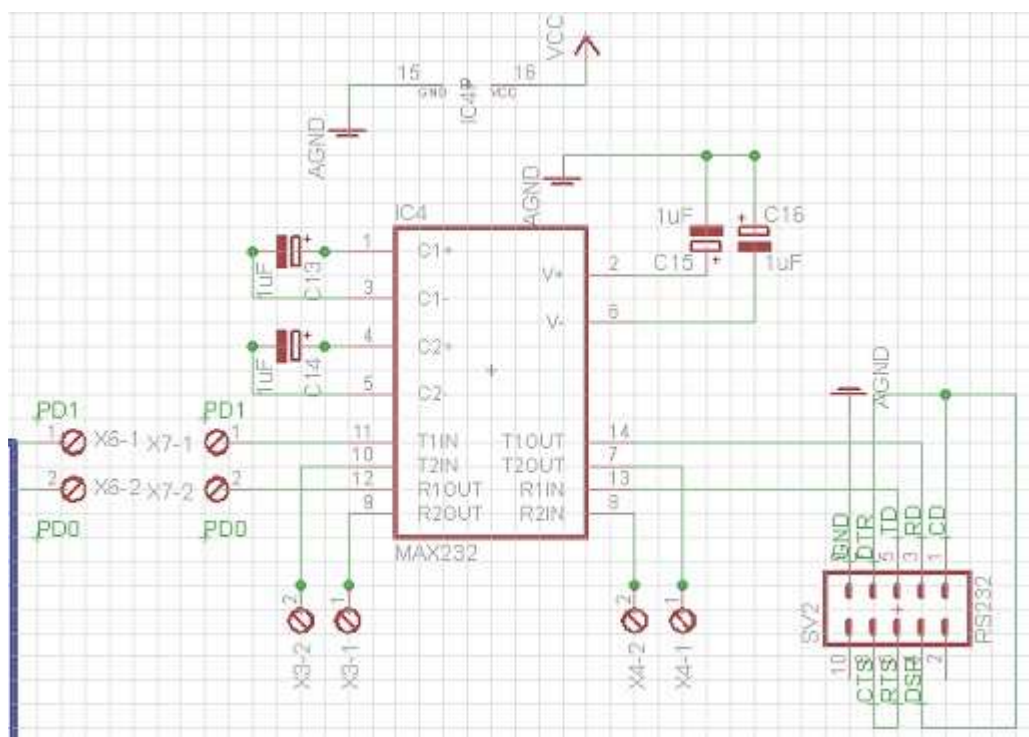
78S05 je pozitivní napěťový regulátor, dostupný v pouzdře TO-220 a TO-3. Integrovaný obvod se charakterizuje vlastnostmi a provozními podmínkami, za kterých je garantován jeho správný chod. Tyto hodnoty jsou popsány v tabulce č. 6. 78S05. Má teplotní a proudovou pojistku, což ho činí při správném zapojení téměř nezničitelným. Je tu také možnost přišroubování chladiče. 78S05 je hojně rozšířen v mnoha zařízeních či obvodech s velkými nároky na přesnost a kvalitu. [7]

Tab. č. 6 Provozní podmínky a hodnoty 78S05, převzato a upraveno z [7]

| Symbol | Parametr | Popis | Min. | Typ. | Max. | Jedn. |
|------------------|------------------|---------------------------------------|------|------|------|-------|
| V _{out} | Výstupní napětí | V _{in} =10V T=-55°C až 150°C | 4,75 | 5 | 5,25 | V |
| I _{out} | Výstupní proud | | | | 2 | A |
| T | Provozní teplota | -55 až 150 | | | | °C |
| V _i | Vstupní napětí | Stejnoseměrné (DC) | 5 | 9-12 | 18 | V |

Stabilizační obvod též obsahuje filtrační kondenzátory. Tyto kondenzátory jsou použity z důvodu možného kolísání vstupního napětí do stabilizátoru a kondenzátory na vstupu jsou použity z důvodu možného kolísání zátěže stabilizátoru. Tím ulehčíme stabilizátoru kompenzace těchto případných kolísání. Hodnoty kondenzátorů jsou stanoveny v datasheetu pro IC 78S05. Stabilizační obvod také obsahuje dvojice konektorů X5-1, na který je přiveden kladný pól ze zdroje a X5-2 na který je přiveden záporný pól ze zdroje. Jako zdroj může být použit stejnosměrný adaptér např. 9-12V, který lze zakoupit. V případě použití adaptéru musíme propojit konektory X1-1 s X2-1 a X1-2 s X2-2. V případě, že budeme používat kvalitnější stabilizovaný regulovatelný zdroj, můžeme stabilizační obvod úplně odstavit a přivést napájení 5V rovnou na konektor X2 a to konkrétně kladný pól na X2-1 a záporný pól na X2-2.

7.1.4 Rozhraní RS232



Obr. č. 25 Schéma obvodu pro RS232

ATmega8535 podporuje na portu D sériový přenos informací, čehož využívá například RS232. Tuto sériovou linku můžeme spojit s PC buď přímo, nebo v případě absence redukce USB to RS232. Tato linka je výhodná zejména potřebujeme-li ATmega8535 řídit přímo programem z PC, napsaném například ve Visual Basicu atd., proto by byla škoda, kdyby deska tuto linku postrádala. Obvod pro RS232 obsahuje integrovaný obvod MAX 232 označený na schématu jako IC4, který se stará o komunikaci mezi PC a ATmega8535. Integrovaný obvod požaduje zapojení kondenzátorů mezi vývody C1+ a C1-, C2+ a C2- V+ a zemí, V- a zemí. Hodnoty těchto kondenzátorů pro různá napájecí napětí jsou uvedeny v datasheetu a slouží jako filtrace kolísání toho napětí.

Jak můžeme ze schématu zobrazeného na obrázku č. 25 vidět MAX 232, může obsluhovat dvě sériové linky. Tuto možnost nevyužijeme a stačí nám pouze vývod pro jednu. Z toho důvodu vývody T2IN, R2OUT, T2OUT, R2IN jsou přivedeny na konektory, kdyby je bylo potřeba použít, můžeme je připojit vodičem. Konektor pro

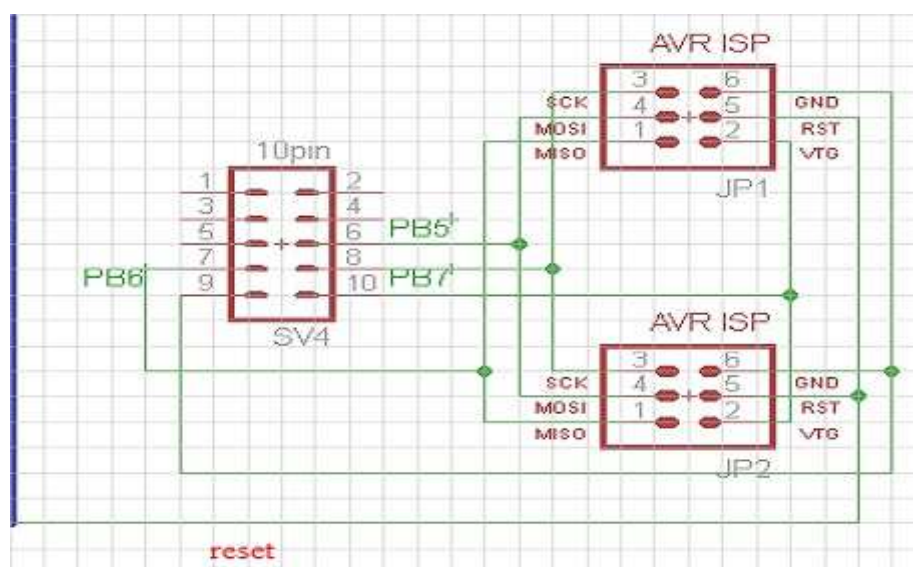
RS232 je z MAX 232 vyveden v podobě konektoru ML10, jehož piny jsou popsány ve schématu. Pro spojení s klasickým konektorem RS232 poslouží připravený kabel, který bude tyto dva konektory spojovat.

Pokud chceme využívat rozhraní RS232, potřebujeme spojit na desce konektory X6-1 s X7-1 a X6-2 s X7-2 kvůli přivedení signálu z ATmega8535 z pinu PD1(TXD Transfer data) na vstup MAX232 (T1IN) a signálu z MAX 232 (R1OUT) na ATmega8535 na pin RXD (Recieve data). Když je používáno RS232, doporučuji na PORTD nepřipojovat žádné zařízení a vybrat raději port jiný z důvodu přítomnosti signálů na konektoru SV6 (Obr. č. 22), konkrétně na pinu 2 a pinu 3, díky komunikaci ATmega8535 a MAX 232. Ostatní piny jsou v případě potřeby k dispozici.

Tab. č. 7 Základní parametry a hodnoty MAX232, převzato a upraveno z [8]

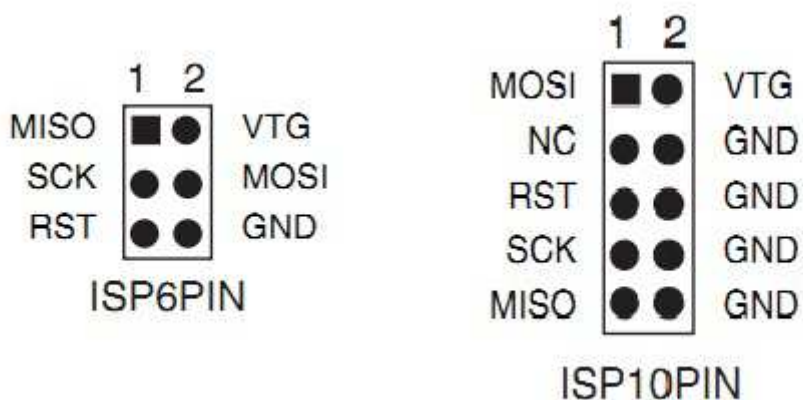
| Symbol | Parametr | popis | Min. | Typ. | Max. | Jedn. |
|--------|------------------|------------------|------|------|------|-------|
| Vcc | Napájecí napětí | | 4,7 | 5 | 5,9 | V |
| T | Provozní teplota | | | 0 | 70 | °C |
| C1 | Kondenzátor C1 | Při Vcc 5V ± 10% | | 1 | | μF |
| C2 | Kondenzátor C2 | Při Vcc 5V ± 10% | | 1 | | μF |
| V+,V- | CV+, CV- | Při Vcc 5V ± 10% | | 1 | | μF |

7.1.5 ISP rozhraní



Obr. č. 26 Schéma připojení ISP

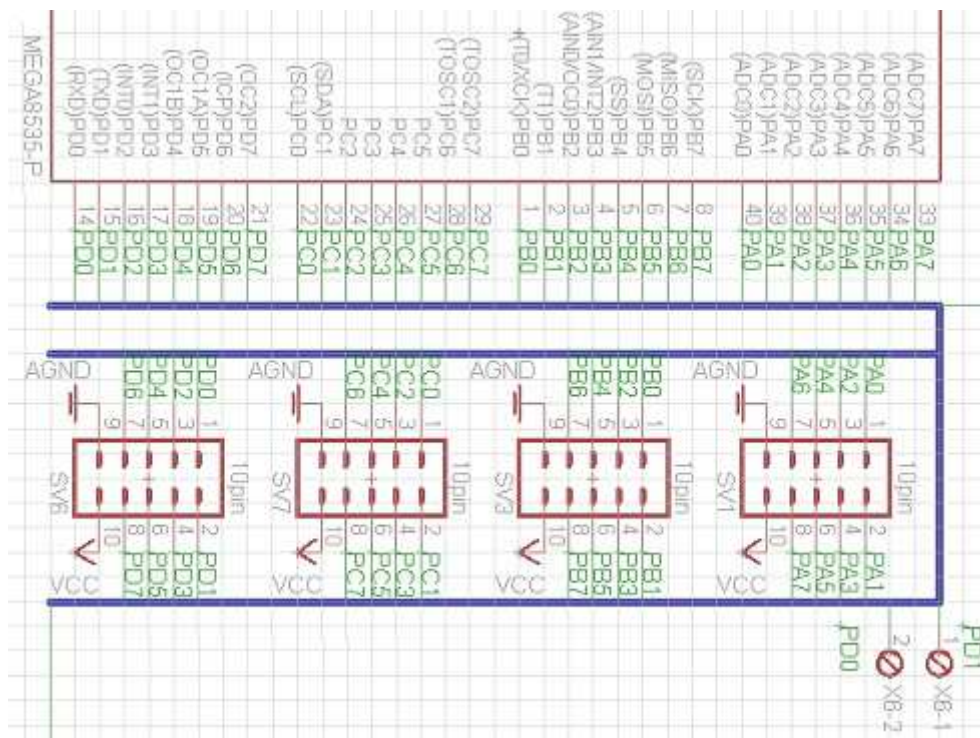
ISP, nebo-li in system programin, je druh nízkonapětového programování mikrokontroléru. ISP využívá SPI (serial peripheral interface), který podporuje ATmega8535 na portu B. ISP využívá pouze několika linek znázorněných na schématu na obrázku č. 26. Konkrétně SCK (hodiny), MOSI (master out slave in), MISO (master in slave out), RST (reset), GND (zem), VTG (napájení 5V). ISP se dá připojit samozřejmě také 10pinovým konektorem CON-ML10. [4]



Obr. č. 27 možnosti připojení ISP, převzato a upraveno z [9]

Pro připojení ISP jsem použil konektory CON-ML6, tedy šesti pinové. Na desce jsou použity dva, které jsou spolu paralelně propojeny z důvodu, že bychom chtěli najednou programovat například dva mikrokontroléry. Do konektoru JP1 (obrázek č. 26) připojíme programátor, který umožňuje externí ISP programování, například jsou dnes k dispozici programátory pro ISP do USB, který by se dal velice dobře použít. Pokud budeme chtít programovat mikrokontrolér přímo na desce, musíme propojit konektor SV4 (obrázek č. 26) s portem B mikrokontroléru konektor SV3 (obrázek č. 22). Pro paralelní programování druhého mikrokontroléru připojíme zařízení na konektor JP2 (obrázek č. 26).

7.1.6 Připojení portů



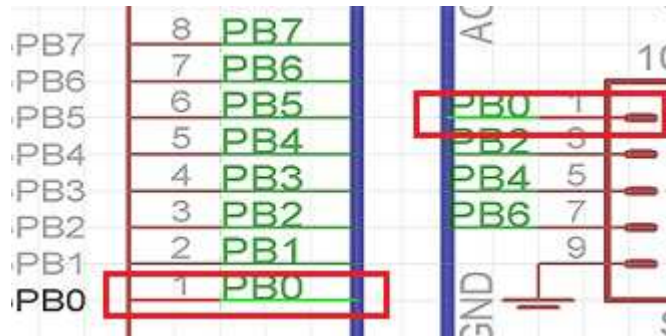
Obr. č. 28 Připojení portů k ATmega8535

Na obrázku č. 27 je vidět řešení připojení jednotlivých portů. Jak už bylo řečeno v popisu pinů mikrokontroléru ATmega8535, tento mikrokontrolér má čtyři porty po osmi pinech. K připojování těchto portů k různému zařízení se velice dobře hodí konektory CON-ML10.

Na každý konektor je přivedeno osm pinů daného portu, zem a napájení. Zem a napájení proto, aby zařízení, které se k portům připojuje (jedná se většinou o jednoduché obvody s různými integrovanými obvody s napájením +5V a nižším), nemusely být dodatečně napájeny. Vcc a GND pro dané zařízení jsou přivedeny přímo deseti žilovým kabelem, což je velice jednoduché a efektivní. Napájení je připojeno v momentě zapojení daného modulu na příslušný port. Kam jsou zapojeny jednotlivé piny do konektoru, vidíme na obrázku č. 27.

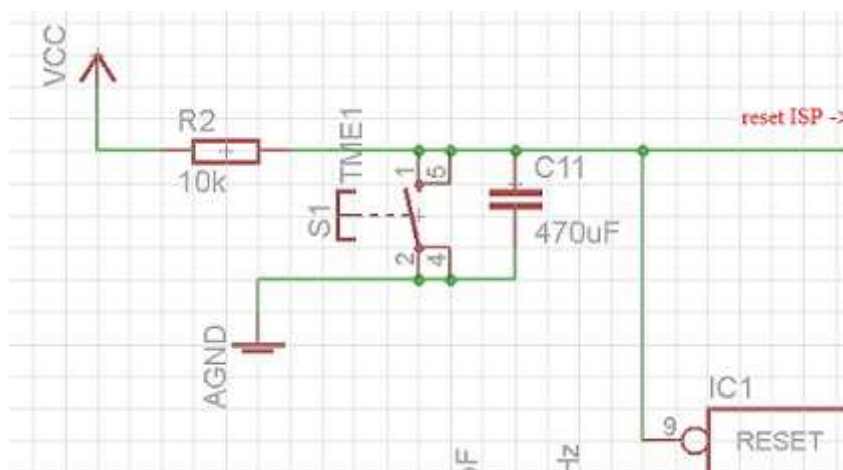
Na schématu na obrázku č. 27 vidíme řešení připojení pomocí funkce Eagle sběrnice (tlustá modrá čára). Toto řešení je na pohled elegantnější, protože se při

pohledu na schéma neztrácíme ve vysokém množství vodičů, které by byly jinak ve schématu obsaženy. Toto zapojení je ale zdouhavější, protože se musí přesně pojmenovat pin mikrokontroléru a pin konektoru, které se na sebe mají připojit, což také vede k větší pravděpodobnosti udělat chybu. Proto je důležité neustále kontrolovat spojení pomocí funkce Show.



Obr. č. 29 Zvýraznění spojení pomocí funkce Show

7.1.7 Resetovací obvod

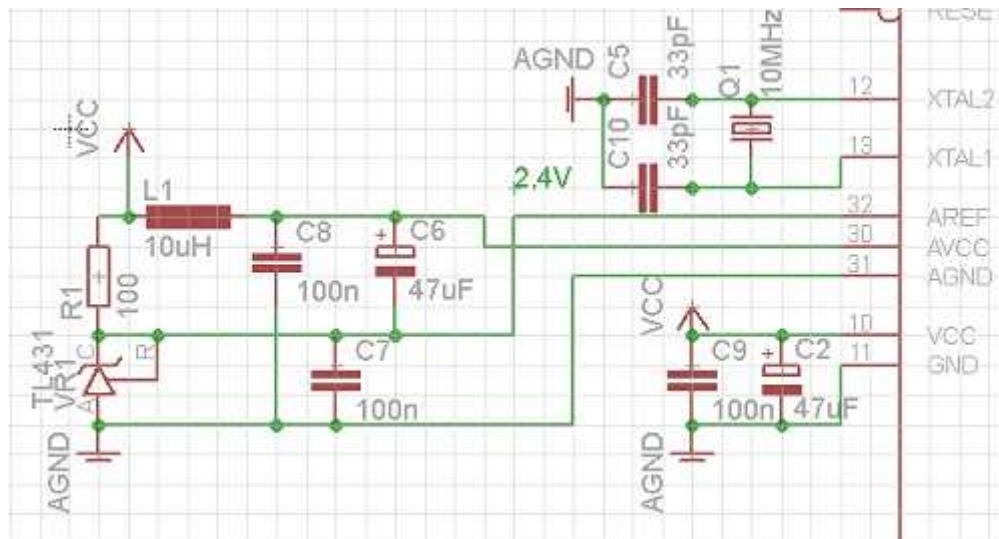


Obr. č. 30 Resetovací obvod mikrokontroléru

Na obrázku č. 30 je vidět externí resetovací obvod, pro resetování a znovu spuštění mikrokontroléru. Resetovací obvod obsahuje rezistor 10K Ω , tlačítko, a filtrovací kondenzátor. Na pin RESET mikrokontroléru působí nepřetržitě napětí. Pokud chceme mikrokontrolér resetovat, zmáčkne tlačítko, díky čemuž uzemníme Reset pin, spustíme interní reset (interní reset jde do "high"). Po povolení tlačítka začne

zpoždovací odvod odpočítávat časovou konstantu (t_{tout}), po které jde interní reset do "low" a je znovu spuštěn MCU. (obrázek č. 11 a podmínky pro externí reset)

7.1.8 Napájení a krystal



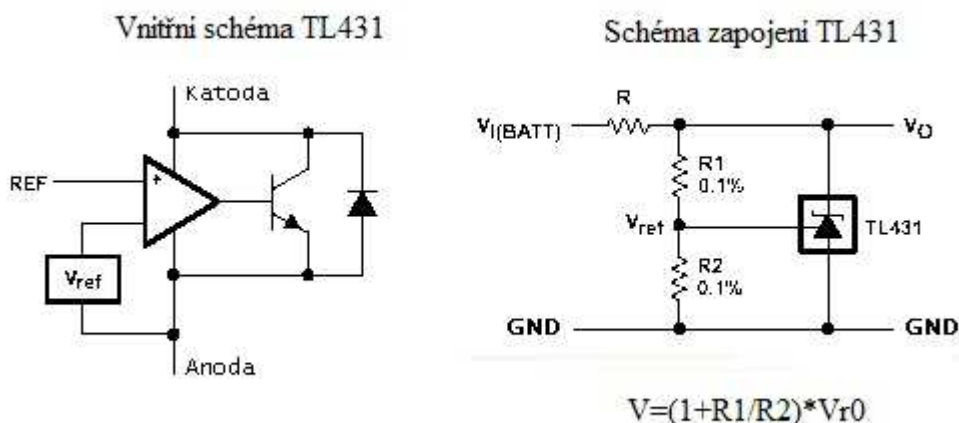
Obr. č. 31 Napájení a krystal

Na obrázku č. 31 vidíme schéma zapojení napájení mikrokontroléru, zapojení externího krystalu a napájení A/D převodníku na portu A (AVCC) a referenční napájení ostatních A/D převodníků (AREF).

Jak už bylo řečeno (v popisu jednotlivých pinů mikrokontroléru), pokud budeme chtít někdy v budoucnu využívat A/D převodník na portu A, je dobré tento A/D převodník (AVCC) zapojit k napájecímu napětí (VCC) přes LC filtr, který má následující funkce: tlumivka L prodlužuje dobu nabíjení kondenzátoru a tím omezuje proudové nárazy, výstupní napětí dosahuje potřebné "tvrdosti" a potlačuje zvlnění, které by mohlo znamenat v tomto případě nepřesnost A/D převodníku. Výstupní napětí z LC filtru téměř nemění svou velikost (nekolísá) ani při náhlých změnách zatížení. Hodnoty těchto součástek jsou zaznamenány v datasheetu. [4]

Na pin AREF přivádíme referenční napětí pro A/D převodníky. Toto napětí má mít hodnotu 2,5V. Pro vyřešení tohoto napájení jsem použil integrovaný obvod TL431. Jedná se o tří svorkový napěťový regulátor s nastavitelným výstupním napětím, se zaručenou teplotní stabilitou a malým výstupním odporem. Vnitřní reference (nejmenší možný krok, tedy nejmenší možné výstupní napětí) $V_{r0} = 2,5V$. Výstupní napětí může

být nastaveno v rozsahu 2,5V-36V. Výstupní napětí se nastavuje dvěma rezistory. Celé zapojení pak potřebuje ještě sériový srážecí rezistor. Schéma zapojení (obrázek č. 32). Obvod lze použít také jako přesnou referenci pro zdroje napětí s větším výstupním proudem a spínané zdroje. Strmá spínací charakteristika z něj vytváří například ideální přepěťový spínač. Vlastnosti obvodu jsou, stálý teplotní koeficient, výstupní odpor typicky 0,2Ω, referenčním vstupem teče proud 0,2μA-0,4μA. Příčný proud 1mA-100mA zaručuje spolehlivou stabilizaci výstupního napětí. Povolený ztrátový výkon se liší podle pouzdra a v závislosti na teplotě (800mW-1500mW při okolní teplotě 25°C). [10]



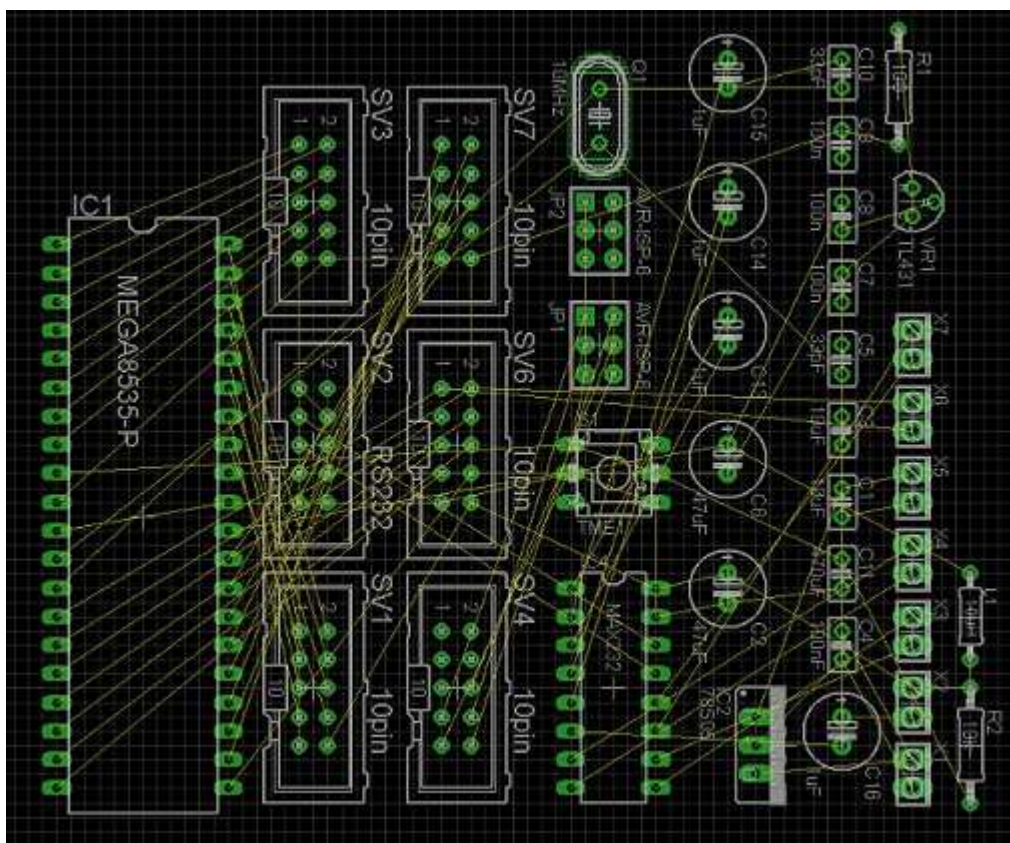
Obr. č. 32 Schéma zapojení TL431, převzato a upraveno z [10]

Dále na schématu na obrázku č. 31 vidíme zapojení externího krystalu na piny mikrokontroléru XTAL1 a XTAL2. Pro správný a přesný chod krystalu je nutno použít dvou parazitních kondenzátorů C10 a C5 v rozmezí 20pF-33pF dle zapojení na obrázku č. 31. [4]

Mezi vstupy VCC a GND je nutno zapojit dvojici filtračních kondenzátorů C9 a C2, přičemž alespoň kondenzátor C2 by měl být na desce osazen co nejbližší u pinů mikrokontroléru, kvůli minimalizaci ztrát. [4]

7.1.9 Návrh masky DPS procesorové desky

V momentě, kdy máme v programu Eagle navržené elektrické schéma desky, můžeme začít s vytvářením masky pro DPS. Tuto masku lze elegantně vytvořit také v jednom z modulů Eagle, který je dokonce aktualizován přes Eagle control panel, který neustále běží na pozadí, pokud s programem Eagle pracujeme. Nicméně doporučuji vytvářet masku pro DPS až v momentě, kdy máme finálně vytvořené schéma, protože aktualizace součástek a spojů v modulu pro návrh DPS nemusí být vždy přínosem. Spíše naopak to vede k neustálému přesouvání součástek a úprav cest tak, aby se nekřížily. Tedy to přidělová práci. Spíše v tom vidím využití v momentě, kdy ve schématu objevíme chybu, nebo ho musíme rozšířit či náhodně upravit.



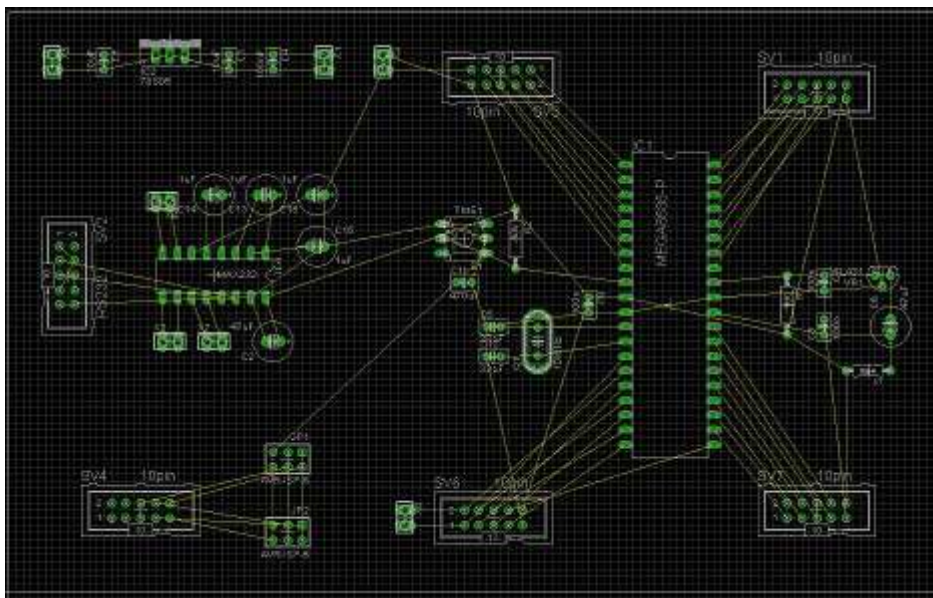
Obr. č. 33 Modul Eagle pro vytvoření masky DPS

Na obrázku č. 33 vidíme modul pro vytvoření masky DPS. Tento modul zpustíme funkcí board. Po vytvoření souboru board program Eagle přenesou jednotlivé součástky na podklad a vytvoří k nim spoje podle schématu (žluté čáry). Na uživateli je,

aby tyto součástky rozmístil na plochu o velikosti desky, na kterou se bude plošný spoj vytvářet. Většinou se používá eurodeska o velikosti 160mm x100mm, pro kterou lze vložit rámeček o dané velikosti do Eaglu už při návrhu schématu.

Součástky přesouváme na určená místa. Rozmístění součástek je vhodné volit rovnoměrně po celé ploše rámečku. Zajistíme tak přehlednost schématu. Nejdříve se mnohdy vyplatí umístit "centrální součástku" a okolo ní ostatní. Je vhodné zapnout pokládací raster GRID. Součástky pokládáme symetricky, vyrovnané v horizontální či vertikální poloze. [6]

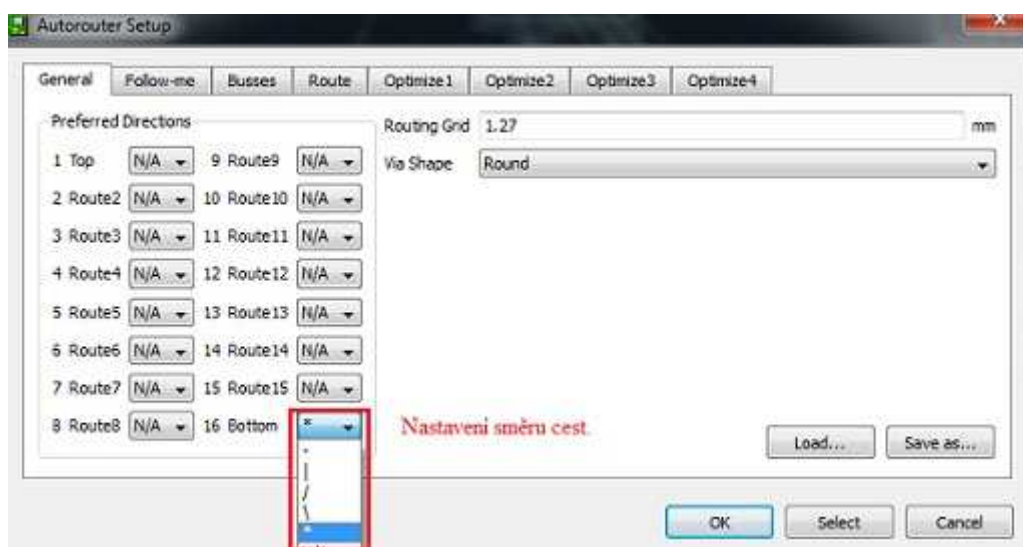
Při větším počtu součástek zabere rozmístění poměrně dlouhou dobu. Vybrat polohu součástek tak, aby při jednovrstvém DPS bylo obsaženo co nejméně propojek a zároveň, aby rozvržení součástek dávalo jistý smysl, není vždy jednoduché.



Obr. č. 34 Rozmístěné součástky na podkladu

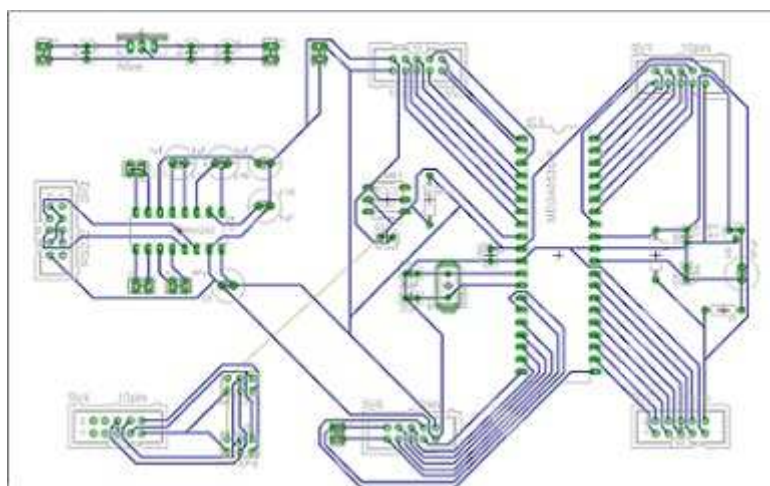
Na obrázku č. 34 můžeme vidět první rozmístění součástek na podkladu o velikosti desky. Samozřejmě, že toto rozmístění nemusí být konečné. Upravovat a doladovat pozice součástek je možné kdykoliv. Nyní máme součástky rozmístěny a připraveny na routování. Routování je proces, při kterém vytváříme vodivé cesty, nebo spoje jednotlivých součástek tak, aby se nekřížily cesty, které nemají. Routování můžeme provádět sami, nebo lze použít funkci Eagle "autorouter", který nám navrhne jednotlivé spoje, podle našeho nastavení, které pak ručně doladíme do finální podoby.

Při samotném routování se stává, že rozmístění součástek, které se zdálo být na první pohled "dobré", je stěží uskutečnitelné, nebo neuskutečnitelné vůbec. Proto rozmístit součástky napoprvé při vyšším počtu tak, aby byly kvalitně vyrouterovány, je skoro nemožné. Proto je dobré desku rozdělit do jednotlivých bloků a ladit rozmístění součástek v jednotlivých blocích, když daný blok odpovídá požadavkům, necháme ho a ladíme další. Samozřejmě čím větší zkušenosti člověk má, tím lépe odhaduje pozice a dokáže předvídat problémy, které se naskytou.



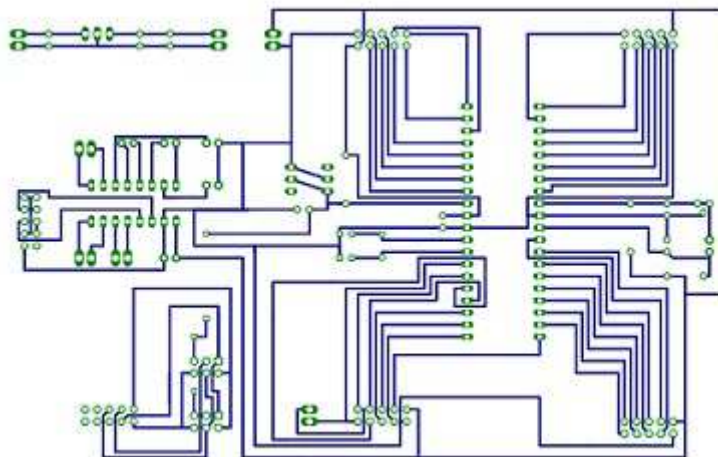
Obr. č. 35 Nastavení autorouteru

Na obrázku č.35 vidíme nastavení autorouteru, kde se dá nastavit mnoho věcí, například počet vrstev, směr cest v jednotlivých vrstvách optimalizace vrstev atd.



Obr. č. 36 Automatické vyrouterování

Na obrázku č. 36 vidíme jak program Eagle automaticky vyroutoval spoje mezi součástkami, což je výborné protože za nás provedl ideální výpočet cest tak, aby nastal co nejmenší počet konfliktů a spoje byly co nejkratší, ale jak je vidět z obrázku, je zapotřebí tyto spoje upravit ručně.

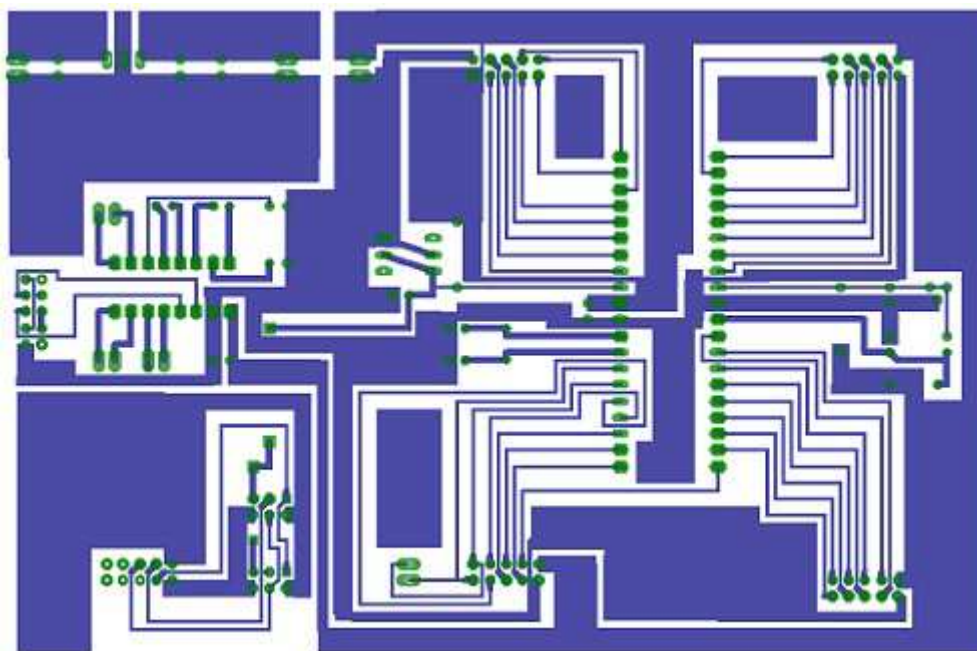


Obr. č. 37 Ručně upravené spoje součástek

Na obrázku č. 37 vidíme ručně upravené spoje součástek, které už připomínají masku plošného spoje, nicméně je potřeba ještě vytáhnout napájení a zem, popřípadě kdyby jsme desku řešili dvouvrstvě, zem by mohla být celá jedna vrstva a na druhé by byly jenom cesty součástek a napájení. To by asi bylo nejideálnější řešení. Deska by nevyzařovala, nevíbrovala a nerušila okolí, zem by byla ideálně přivedena všude tam, kde má být pouze vyvrtáním příslušného otvoru v daném místě.

Kromě vytažení napájení a země je potřeba také zvýraznit jednotlivé pady (na obrázku č. 37), zelené plošky určené pro připájení součástek. To je velice důležité, protože by se mohlo stát, že při zahlubování plošného spoje by se některé pady poškodily a nešlo by součástku kvalitně připájet. Je třeba volit kompromis mezi tloušťkou padu a místem kolem něj, aby zase nedošlo ke spojení s cestou či jiným padem. Tam, kde je možnost pad zvýraznit více, není problém, ale například u konektorů, kde jsou jednotlivé pady poměrně blízko u sebe, by mohlo dojít ke spojení. Zvýraznění padů v Eaglu mi nepřišlo příliš dobře řešené pro běžného uživatele. Je tu možnost pad zvětšit o předem stanovenou velikost, přičemž větší pad bude umístěn na

menší přesně se středovou souměrností. Nelze ho posunout do strany či jinak upravit. Toto může být problém právě u konektorů, kde je více místa pro zvýraznění - například na vnějším obvodu, ale méně místa mezi piny. Proto lze masku ještě upravit například v Corel Draw, nebo v malování, ale musíme dbát zvýšené opatrnosti aby grafický program neupravil výslednou velikost masky (aby rozteče mezi piny zůstaly nezměněny), což je velice důležité, jinak by se mohlo stát, že po vyvolání desky nebudou sedět rozteče součástek s pady, a tedy součástky nepůjdou napájet.

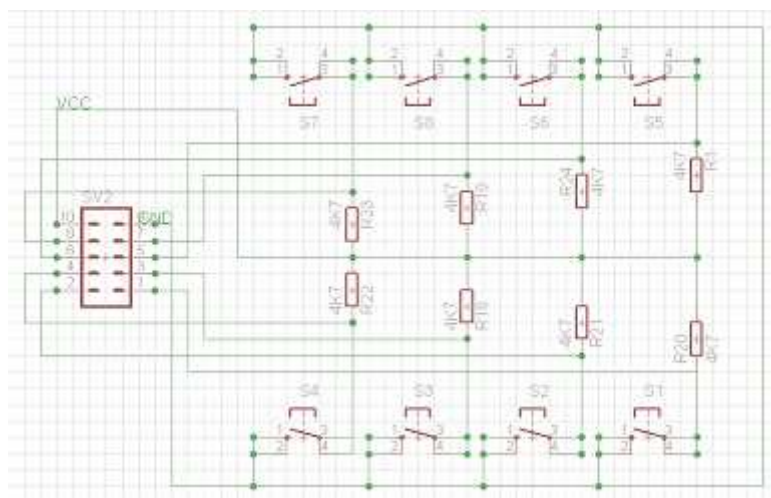


Obr. č. 38 Konečná podoba masky DPS

7.2 Přípojné moduly

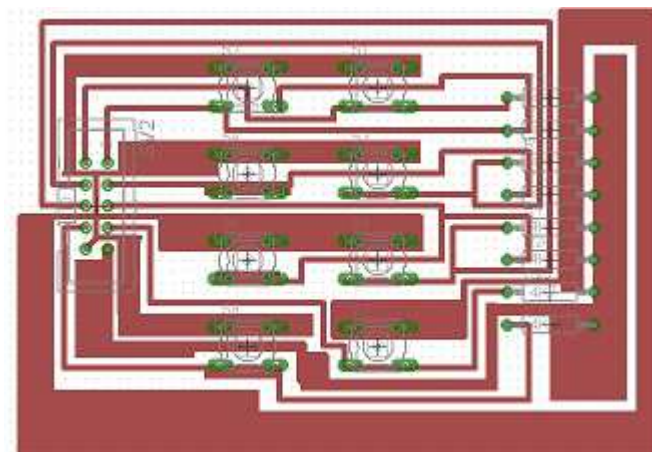
Součástí stavebnice jsou také přípojné moduly, které se k procesorové desce připojují pomocí deseti žilových kabelů k příslušným portům mikrokontroléru. Těchto modulů může být celá řada. Vybral jsem jednak takové, u kterých je možné jednoduše ověřit funkčnost desky a také takové, na kterých je možné dobře vysvětlit základní principy programování AVR mikrokontroléru v příslušném programovacím prostředí. Samozřejmě je tu možnost vytvoření dalších modulů, které se dají bez problému k desce připojit, bude-li potřeba. Moduly se dají kombinovat mezi sebou.

7.2.1 Tlačítka



Obr. č. 39 Schéma modulu tlačítka

Modul tlačítka umožňuje indikaci, nebo aktivaci libovolných procesů. Pokud je tlačítko x stisknuto, generuje vstupní signál do pinů portu na který je modul připojen. Schéma tohoto modulu vidíme na obrázku č. 39. Je to velice jednoduchý obvod. Stiskem tlačítka je uzavřen daný okruh a okruhem začne protýkat proud.

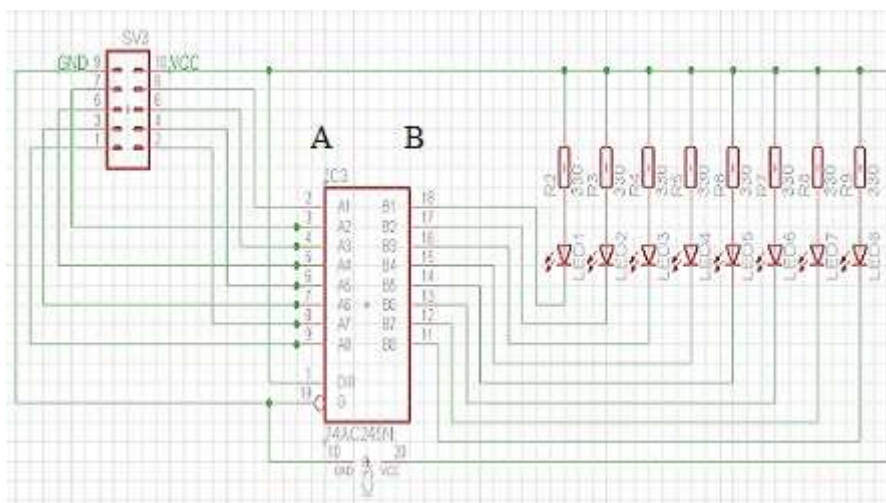


Obr. č. 40 Maska tlačítek pro DPS

7.2.2 Ledky

Pomocí modulu ledky můžeme detekovat delší výstupní signály z jednotlivých pinů libovolného portu. Pokud je ledka rozsvícena, pin je nastaven jako výstupní a je na

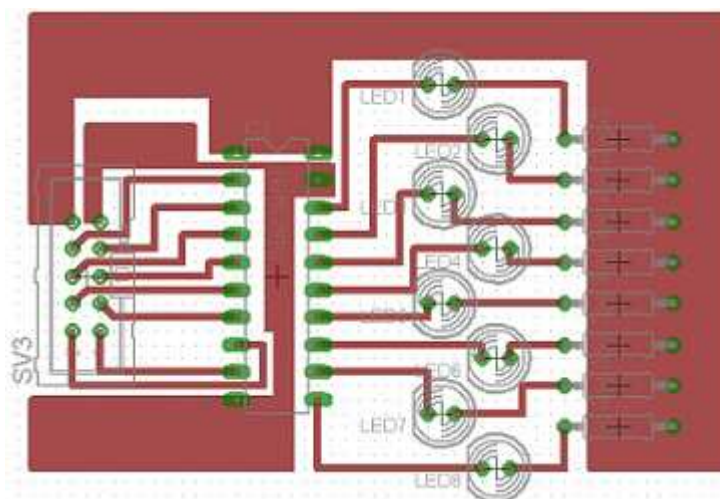
něm log. 1. Tento modul lze dobře kombinovat s modulem tlačítka. Např. pokud stisknu tlačítko na pinu 1 portu A, rozsvítím ledku na pinu 1 portu B atd.



Obr. č. 41 Schéma modulu ledky

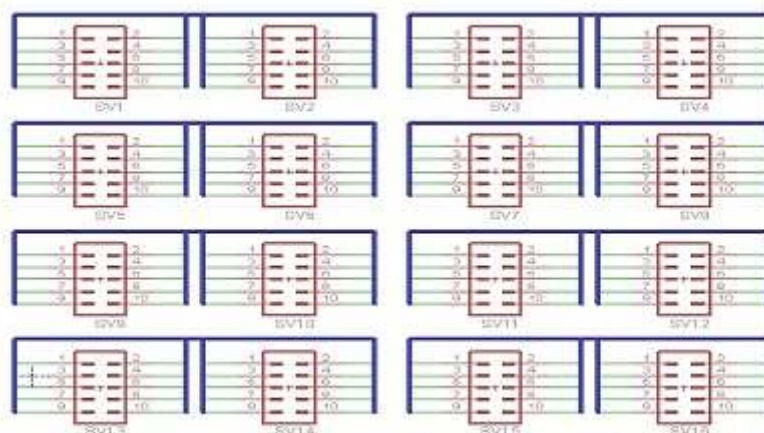
Na obrázku č. 41 vidíme schéma modulu ledky. Toto schéma obsahuje pro každý pin ledky a rezistor, navíc je v modulu obsažen také budič sběrnice 74AC245N, i když ATmega8535 přímo na vybuzení ledky na svém výstupu budič nepotřebuje, nicméně je lepší budič použít.

74AC245N má napájecí napětí od 2V-6V, provozní teplota je -40°C do 85°C , Nízký vstupní proud $0,1\mu\text{A}$ při 25°C , výstupní proud 24mA při 25°C , velká mechanická i elektrická odolnost. Směr buzení je od A do B. [11]



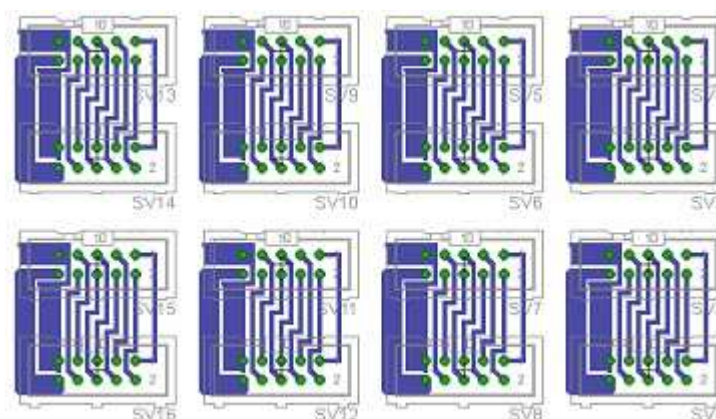
Obr. č. 42 Maska modulu led pro DPS

7.2.3 Modul propojovací deska



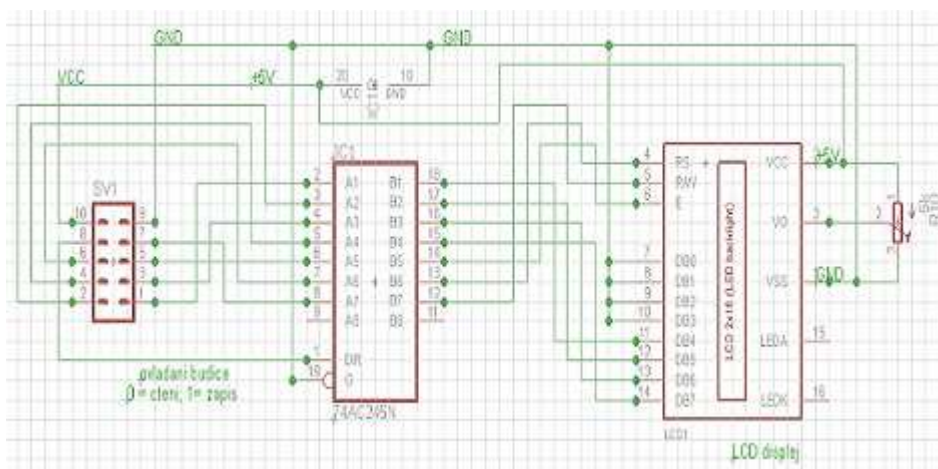
Obr. č. 43 Schéma propojovací desky

Propojovací deska na obrázku č. 43 obsahuje šestnáct konektorů, čtyři pro každý port. Piny například konektoru SV1 a SV2 jsou zapojeny do série, stejně tak piny konektoru SV3 a SV4. Pokud budeme chtít připojit zařízení k desce, kterému nevyhovuje připojení pinů mikrokontroléru na konektor daného portu a z nějakého důvodu potřebujeme piny libovolně přeházet, připojíme konektor portu do konektoru SV1 deseti žilovým kabelem. Z konektoru SV2 můžeme libovolně přeházet signály pinů na konektor SV3 jednožilovými kabely a zařízení připojíme konektorem SV4 opět klasickým deseti žilovým kabelem. Deska je tedy velice jednoduchá, v jistých situacích může být ale velice užitečná a ušetří nám výrobu unikátních kabelů pro každé zařízení.



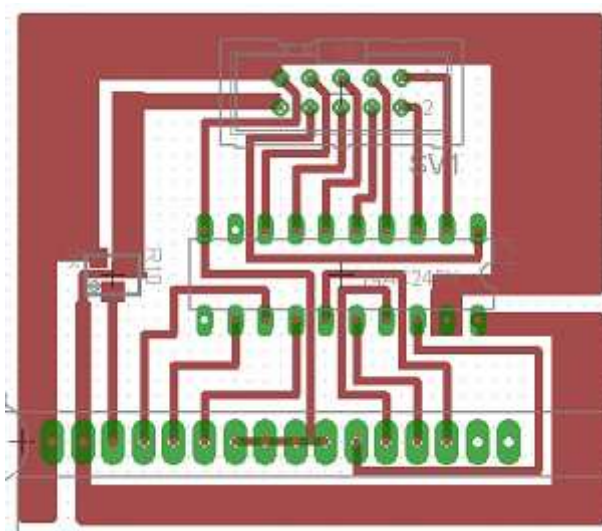
Obr. č. 44 Maska propojovací desky pro DPS

7.2.4 Modul LCD



Obr. č. 45 Schéma modulu LCD

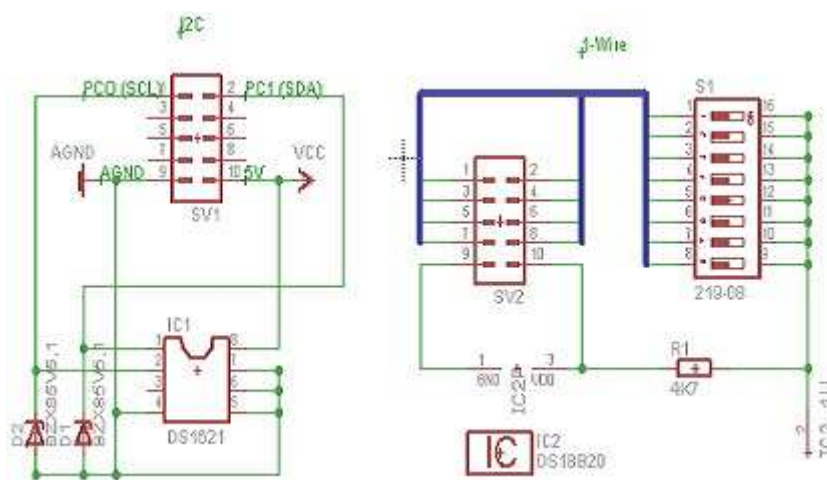
Na obrázku č. 43 je znázorněno schéma zapojení LCD modulu. LCD modul stejně jako led modul používá IC74AC245N, k buzení sběrnic. Tento integrovaný obvod a jeho základní vlastnosti jsou popsány v modulu LED. Toto schéma obsahuje také trimr R10, kterým se nastavuje jas LCD. Modul je navržen pro LCD o velikosti 2x16.



Obr. č. 46 Maska LCD modulu pro DPS

7.2.5 Moduly Teploměr (I2C, 1-Wire)

Pro komunikaci po sběrnicích I2C, 1-Wire poslouží například moduly teploměry. Pro tyto moduly se perfektně hodí integrované obvody DS1621 pro komunikaci po I2C sběrnici a DS18B20 pro komunikaci po 1-Wire sběrnici. ATmega8535 umožňuje připojení zařízení, které tyto sběrnice používají. Jak už bylo napsáno v popisu jednotlivých pinů, I2C a 1-Wire zařízení připojujeme standardně na PORTC. U 1-Wire by teoreticky mělo být jedno, na jaký port zařízení připojíme.



Obr. č. 47 Moduly teploměry

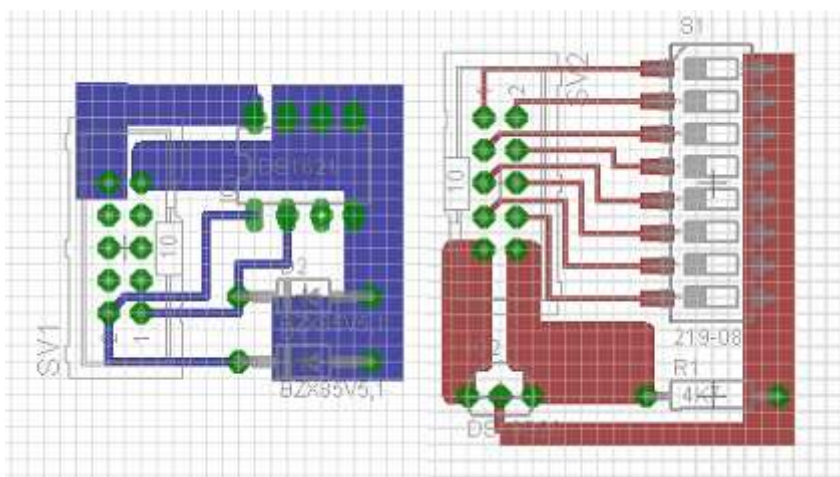
Jak můžeme vidět na obrázku č. 45 I2C teploměr využívá čidla DS1621 a piny daných portů (PC0 SCL, PC1 SDA), napájení a zem. Komunikace probíhá po datové lince, která je řízena hodinami. Naopak 1-Wire čidlo DS18B20 využívá pro komunikaci pouze jednu linku, která je využívána jak pro data, tak pro hodiny. U 1-Wire teploměru vidíme zapojen přepínač, který slouží pro zapnutí daného pinu, po kterém probíhá komunikace. Ostatní jsou vypnuty.

DS1621 nepotřebuje k měření teploty žádné další externí komponenty. Rozsah měření je -55°C - $+125^{\circ}\text{C}$. Přesnost měření je $0,5^{\circ}\text{C}$. Napájecí napětí je 2,7V-5,5V, Převod teploty na digitální 8bitové slovo je kratší než 1s. Muže být řízen termostatem, pokud to umožňuje aplikace. [12]

Popis pinů: Pin1-SDA (serial data I/O), Pin2-SCL (Sériové hodiny), Pin3-Tout (výstupní signál pro termostat), Pin4-GND (zem), Pin5,6,7- A2, A1, A0 (adresní vstupy čipu), Pin8-Vcc (napájení). [12]

DS18B20 potřebuje pro komunikaci pouze jeden pin. Každé zařízení má unikátní 64-bitový kód, zapsaný v paměti ROM na čipu. Možnost připojení a adresace více zařízení, po jedné lince. Pro měření teploty nepotřebuje žádné další externí zařízení. Napájecí napětí je od 3V-5,5V. Rozsah měření od -55°C - 125°C . Přesnost měření je $0,5^{\circ}\text{C}$. Rozlišení teploměru je od 9 do 12 bitů. Při rozlišení 12 bitů převede teplotu na bitové slovo za maximální dobu 750ms. [13]

Popis pinů: Pin1-GND (zem), Pin2-DQ (data), Pin3-Vcc (napájení). [13]



Obr. č. 48 Maska teploměrů pro DPS

8 Výroba plošného spoje

Pokud máme vytvořené masky plošných spojů, můžeme se vrhnout na výrobu plošných spojů jako takovou. V této části bych se chtěl věnovat amatérské výrobě plošných spojů foto cestou. Tedy technologií výroby plošných spojů, kterou jsem vyráběl plošné spoje ke své diplomové práci já a kterou vyrábí plošné spoje mnoho dalších amatérských výrobců. Samozřejmě, že se tato technika, která se liší od profesionální výroby plošných spojů, respektive technologie výroby plošných spojů velkých firem a masová výroba plošných spojů, která musí splňovat nejrůznější normy a nároky na čistotu, přesnost a také ochranu životního prostředí, je úplně jiná. Tyto postupy v běžném prostředí pro běžného uživatele nejsou realizovatelné. Proto je tu

rozebírat nebudu, určitě by to vydalo na samostatnou diplomovou práci a zaměřím se na to, jak se dá velice přesný a kvalitní plošný spoj vyrobit například v domácím prostředí.

8.1 Výroba DPS fotocestou

Na výrobu plošného spoje používám desku, která se dá pořídit v obchodě s elektronikou a elektro součástkami běžně o rozměrech 160mm x 100mm. Deska je z materiálu typu kuprextit, což je sklolaminát odolný při mechanickém i teplotním namáhání, nebo kuprexcart, což je lisovaný tvrzený papír. Deska jako taková se prodává buď s jednou stranou plátovanou měděnou fólií, která se hodí pro jednovrstvé plošné spoje. Pro složitější zapojení lze vytvořit v domácím prostředí i dvouvrstvý plošný spoj, na který budeme potřebovat desku plátovanou měděnou fólií z obou stran. Tloušťka měděné fólie je většinou 35 μ m. Měděná fólie se vyrábí tak, že na elektricky vodivý buben se galvanicky nanáší měď, která je pak z bubnu stržena.

Velikost desky lze samozřejmě upravit dle požadavků ideálně pákovými nůžkami na plech, popřípadě v krajní situaci pilkou na železo, nicméně tento postup velice ničí hranu desky, na pohled to nevypadá potom příliš pěkně a musí o to více deska zabrousit.

Za zmínku stojí, že deska se prodává už většinou povrchově řádně upravena a nalakována. Může se stát, ale že deska není nalakována fotocitlivou vrstvou, kterou budeme později osvětlovat přes masku. V takovém případě si můžeme desku nalakovat sami. Požívá se k tomu přípravek positiv 20, který nanese štětcem, nebo v lepším případě stříkací pistolí, ve velice tenké vrstvě. Deska musí být ale velice řádně odmaštěna a očištěna čistící pastou na pískové bázi, nebo silným odmašťovacím přípravkem, pokud možno přípravek by neměl obsahovat aktivní chlór, aby jinak nereagoval s měděnou fólií. Nanášení fotocitlivé vrstvy je nejlepší za pokojové teploty 20°C, aby docházelo k rovnoměrnému rozlívání laku. Positiv20 je minimálně citlivý na běžné žárovkové světlo takže můžeme pracovat za běžného osvětlení. Po nanesení laku se musí vytvrdit, což trvá zhruba 24 hodin při 20°C, nebo 15 minut při 70°C [14]

Doporučuji koupit desku s nanesenou fotocitlivou emulzí, vybírejme ale obchody, kde jim deska dlouho neleží, a obchod má větší obrat, tím se vyhneme

problému, že deska bude stará. U takových desek reaguje fotocitlivá vrstva minimálně a nebo také vůbec při osvětování s UV zářením.

8.1.1 Výroba masky

Předloha by měla mít následující vlastnosti: vysoký kontrast, dostatečné rozlišení, tvarovou stálost a přesnost, motiv vytištěný zrcadlově tak, aby strana tisku přiléhala na měděnou fólii, čímž vznikne ostřejší kresba. [14]

Masky můžeme vytvořit různými způsoby. Jednotlivé metody a způsoby mají své výhody i nevýhody.

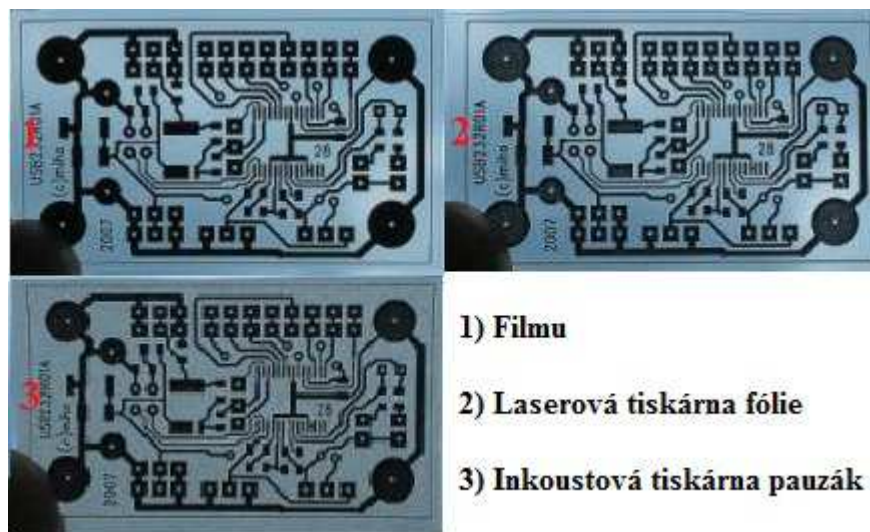
Maska vysvícená na film: Jednoznačně nejlepší a nejpresnější. Jedná se o poloprofesionální výrobu plošných spojů. Filmy určené pro masky plošných spojů mají specifické vlastnosti, jako je minimální deformace, vysoká tvarová a kontrastní stálost atd. Nevýhoda je, že si film nevytvoříme doma. Můžeme si ho nechat vytvořit v profesionálním grafickém studiu. Cena je kolem 100Kč za A4, což není levná záležitost. Vyplatí se v momentě, když budeme vyrábět více desek, nebo pokud víme, že daná maska je v konečném stádiu a nebude delší dobu upravována. [14]

Maska vytištěná laserovou tiskárnou na fólii: Velice snadná výroba a malé náklady. Lze vytisknout na kterékoliv laserové tiskárně, ovšem celkem závažná nevýhoda je ta, že u levnějších tiskáren dochází ke špatnému kontrastu z důvodu nedostatečného krytí.

Maska vytištěná inkoustovou tiskárnou na fólii: Obvykle lepší krytí než u laserové tiskárny. Velice závislé na kvalitě použitého inkoustu a na kvalitě tiskárny. Nevýhodou je velice drahá fólie do inkoustové tiskárny a náklady na inkoust. [14]

Maska vytištěná na pauzovací papír: Z pravidla lepší krytí než při tisku na fólii, i náklady na pauzovací papír jsou menší. Nevýhoda je ta, že skrz pauzovací papír neproniká tolik UV záření jako přes fólii, takže je vyvolání delší a při použití starší desky nemusí k vyvolání dojít. [14]

Samozřejmě, že je možné s trochou trpělivosti a snažení dané masky perfektně vyretušovat pomocí lihové fixy. Většina domácích výrobců volí většinou méně nákladné metody vytvoření masky a poté retušují. Musím ale říci z vlastní zkušenosti, že je nemilé, když po vyretušování celé masky ujede ruka na poslední cestě.



Obr. č. 49 Masky a jejich kvalita, převzato a upraveno z [14]

8.1.2 Osvícení desky

Při osvětlování položíme plošný spoj na tenký molitan, pak následuje deska kuprextitu s fotoemulzí, na to položíme masku s předlohou a navrch položíme ploché sklo. Molitan zajistí rovnoměrné přitisknutí kuprextitu ke sklu. [14]

Když zapneme UV zářič, maska zabrání pronikání záření na místa, která nemají být osvětlena (vodivé cesty) a naopak propustí záření na místa, které mají být osvětlena. Na těchto místech dojde k rozpuštění a narušení fotocitlivé vrstvy. Deska se dále dá vyvolat.

8.1.3 Zdroje světla

Výrobce pro fotoemulzi pozitiv 20 uvádí největší citlivost v rozsahu 340nm - 420nm (UV-A). V praxi se ukazuje, že typická expozice pro různé zdroje světla bývá v jednotkách až desítkách minut ve vzdálenosti zdroje světla od desky cca 20cm. Expozici je třeba vyzkoušet pro konkrétní proces (výbojka, sklo, emulze, předloha). Delší expozice nevedí, pokud je předloha dostatečně kontrastní. [14]

Co se školního procesu týká, masku jsem měl vytištěnou na laserové tiskárně na fólii. Kontrast nebyl extra kvalitní. S použitou výbojkou, která je na katedře fyziky, jsem osvětloval desku zhruba 3-4 minuty, což bohatě dostačovalo akorát na to, aby nebyla rozrušena fotocitlivá vrstva, kde být nemá.

Jako zdroj světla se většinou používají různé druhy výbojek či zářivek. Výbojku nelze připojit rovnou na síť, protože napětí na síti je mnohem větší než napětí potřebné k dosažení výboje. Výbojku připojíme přes trafo nebo tlumivku či elektronický předřadník nebo jinak. Nutno podotknout, že oči máte jen jedny, takže se do žádného zdroje UV záření nedívejte přímo, zejména pozor na výkonné rtuťové výbojky bez luminoforu (čiré baňky). [14]

Druhy výbojek: rtuťová výbojka 125W (horské sluníčko), výbojka 12W do stolní lampy, výbojka pouličního osvětlení (rtuťová, vysokotlaká 250W nebo 400W), nebo v dnešní době například UV led panel výhodný zejména pro nízké napájecí napětí 24V atd. [14]

8.1.4 Vyvolání

Vyvolání provádíme v roztoku NaOH (hydroxid sodný, louh sodný) při koncentraci zhruba 10g/l. Louh vychytává ze vzduchu vodu a oxid uhličitý, proto je nezbytné uchovávat jej v neprodyšně uzavřených nádobách. Roztok nalijeme do misky větší než deska kterou chceme vyvolat. Desku umístíme do misky a potopíme fotoemulzi nahoru. S miskou provádíme krouživé pohyby a omýváme roztokem desku. Vyvoláváme při mírném umělém osvětlení. Doba trvání zhruba 2 minuty. Louh je silnou žíravinou, velice lehce se s ním dá zničit oblečení v případě pocákání atd. [14]

8.1.5 Leptání

Leptání můžeme provádět například v chloridu železitém, kyselině chlorovodíkové, kyselině dusičné. Nejbezpečnější je používat chlorid železitý, protože není nijak zvlášť nebezpečný, kromě toho, že zanechává na oblečení tmavé černé fleky, které nejdou vyprat. Důležité je, aby chlorid železitý nebyl zásaditý a nenapadal fotoemulzi a nedocházelo k podleptání míst, které leptat nechceme. Proto můžeme přidat asi 10ml kyseliny chlorovodíkové na 1l chloridu železitého, tím vytvoříme roztok absolutně vhodný pro leptání desky. [14]

Způsobů leptání je více, nejpoužívanější je však vodorovné leptání, kdy desku položíme na hladinu daného leptacího roztoku tak, aby plavala na této hladině. Je nutné mít povrch desky čistý a pokládání na hladinu provádět tak, abychom nevytvořili

bublinky mezi deskou a roztokem. V čerstvém chloridu leptání trvá asi 10 minut. V průběhu leptání vidíme na vrchní straně desky, jak začíná prosvítat motiv, to je znamení, že se leptání chýlí ke konci.



Obr. č. 50 Vyvolávání a leptání desky, převzato a upraveno z [14]

8.2 Povrchová úprava desky

Po vyleptání desky desku omyjeme od zbytků chloridu železitého a můžeme začít s povrchovou úpravou. Máme několik možností, jak desku povrchově upravit. Buď desku potřeme pájitelným lakem, nebo desku plošně pocínujeme, popřípadě potřeme roztokem rozmíchané kalafuny a lihu. Tato povrchová úprava zabraňuje oxidaci plošného spoje a zlepšuje pájení součástek na desku, tím docílíme i lepších spojů mezi součástkou a deskou a nebudou nám vznikat tzv. studené spoje, kdy se součástka jeví připájená, ale není tomu tak.

8.3 Vrtání a pájení

Vrtání provádíme na stojanové vrtačce, či vrtačce ruční, která ale vyžaduje větší zručnost. Vrtat bychom měli samozřejmě kolmo. Otáčky volíme zhruba kolem 6000ot/min; je vyzkoušené, že běžné vrtáky se při vyšších otáčkách zničí během pár děr, za to profesionální vrtáky vyžadují vyšší otáčky, nicméně jsou křehké a hodně se lámou, zejména pokud nepoužíváme stojanovou vrtačku. Nejdříve bychom měli díry předvrtat jedním tenčím vrtákem zhruba 0,7mm-0,8mm a pak dle potřeby některé díry převrtat na větší průměr. Důležité je, abychom měli při vrtání vrták upnutý ve sklíčidle tak, aby neházel. [14]

Co se pájení týče, to samo o sobě vyžaduje zkušenosti. Bez tréninku to nejde. Žádný popis nebo návod moc nepomůže. Velice dobré je chvíli pracovat pod dohledem zkušenějšího. Největší chybou začátečníků je to, že nepoužívají tavidlo, například kalafunu a pájejí příliš vysokou teplotou, protože se jim cín nechytá na desku a nohy součástek tak, jak by chtěli. Proto dochází k přehřívání součástek a k jejich zničení. Při vyšší teplotě dochází také k oxidaci kovových povrchů. V průběhu chlazení spoje by se s ním nemělo hýbat a jinak ho mechanicky namáhat. [14]

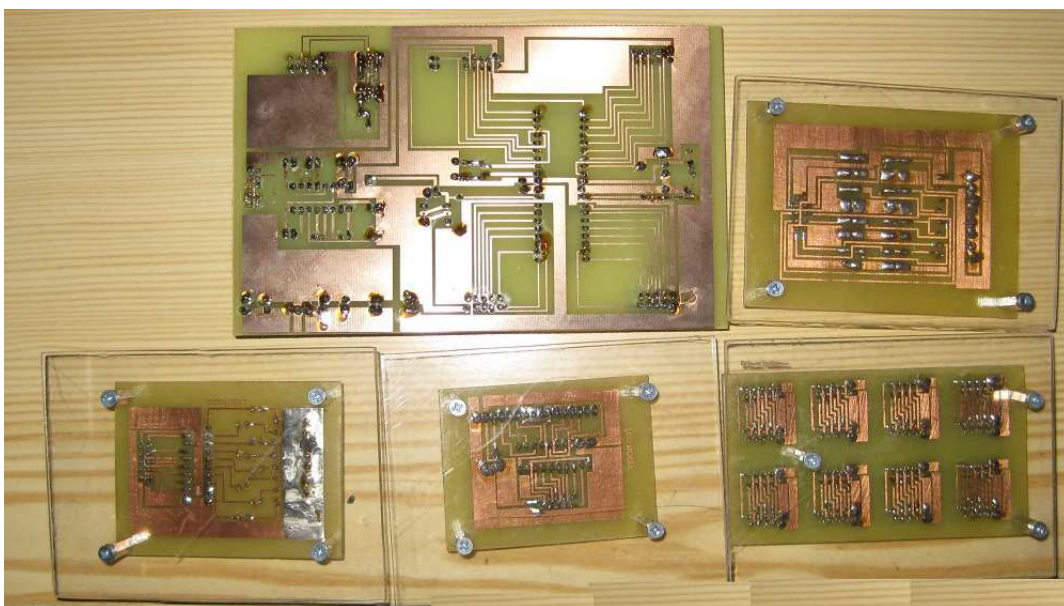
Tab. č. 8 Teplotní kritéria pro bezpečnou montáž, převzato a upraveno z [15]

| Položka | Teplotní kritéria |
|----------------------------------------|-------------------|
| Teplota uvnitř součástky (max) | 220°C |
| Teplota uvnitř citlivé součástky (max) | 150°C |
| Rychlost ohřevu citlivých součástek | 2 - 5°C /s |
| Teplota pájecího spoje | 220°C po 2-4s |
| Teplota pájecího hrotu (max) | 385°C |

8.4 DPS Procesorové desky a některých modulů



Obr. č. 51 Vytvořená stovebnice, strana součástek



Obr. č. 52 Vytvořená stavebnice, strana spoje

9. Programování v Bascomu

Mikrokontrolér budeme programovat v programovacím prostředí Bascom-AVR, který je navržen přímo pro programování mikrokontrolérů AVR. Bascom-AVR obsahuje kromě programovacího prostředí i kompilér, který rozpozná různé chyby v napsaném programu a automaticky uživatele na chyby upozorní, tak simulátor, ve kterém můžeme daný program, pokud je opatřen direktivou \$sim, odsimulovat a zjistit, co program provádí krok po kroku. Toto je velice dobré, pokud je někde v námi napsaném programu chyba a nejsme si jisti kde. Bascom-AVR využívá programovací jazyk BASIC. Tento programovací jazyk je velice vhodný pro začínajícího programátora v oblasti mikrokontrolérů. Práce s tímto programovacím jazykem je mnohonásobně jednodušší než například s assemblerem, protože používá ustálené rutiny, které nemusíme pokaždé znovu psát od začátku. Samozřejmě vše má své výhody a nevýhody. Bascom používá své předdefinované rutiny a knihovny, které jsou předdefinovány výrobcem a které nám velice usnadňují a zlehčují samotný zápis kódu. Toto zlehčení si ale vybere daň jinou a sice tu, že po zkompilování je výsledný soubor, který chceme nahrát do procesoru, zhruba o 20 % větší, než čistý kód v assembleru. Bascom využívá zhruba kolem tří set předdefinovaných rutin, přičemž i pro různé

složitější aplikace si vystačíme s pár desítkami rutin. Bascom umí pracovat se sériovou linkou, LCD, klávesnicí, sběrnicemi atd.

Největší úskalí při používání Bascomu je jeho nastavení. Je potřeba Bascom správně nastavit tak, aby komunikoval s našim mikrokontrolérem, tedy právně nastavit kompilér pro náš mikrokontrolér, popřípadě používáme-li redukci SR232 to USB, tak potřebujeme ještě nastavit port COM, po kterém naše redukce komunikuje. Port COM přiděluje windows automaticky po nainstalování ovladačů k redukci. Také je potřeba v Bascomu nastavit cestu k souboru STK500, který je ve složce s AVR studiem, které musíme mít nainstalované spolu s Bascomem. STK500 je soubor, který využívá programátor STK500 prodáváný společností Atmel AVR při programování mikrokontroléru. Pokud používáme jiný programátor, nastavíme soubor jiný, podle našeho programátoru. Bez těchto nastavení Bascom nebude pracovat správně. Správné nastavení výše popsaných příkladů provedeme v nabídce OPTION -> COMPILER -> CHIP.

The image shows three screenshots of the BASCOM-AVR Options dialog box, illustrating the configuration steps for the compiler, communication, and programmer.

- Top Left Screenshot (Nastavení Chipu):** Shows the 'Compiler' tab with the 'Chip' dropdown menu open, highlighting 'm8535.dat'. Other options include 'FlashROM', 'SRAM', and 'EEPROM'.
- Top Right Screenshot (Nastavení COM portu):** Shows the 'Communication' tab with 'COM3' selected in the 'COM port' dropdown, '57600' in the 'Baudrate' dropdown, and 'None' in the 'Parity' dropdown.
- Bottom Left Screenshot (Nastavení programátoru):** Shows the 'Programmer' tab with 'STK500 (requires stk500.exe)' selected in the 'Programmer' dropdown, '3' in the 'COM port' dropdown, and the path 'C:\Program Files (x86)\Atmel\AVR Tools\STK500' in the 'STK500 EXE' field.

1.) Pokud se mikrokontrolér neohlásí při inicializaci je pravděpodobně špatně nastaven druh chipu.
2.) COM port pro naši redukci RS232 to USB zjistíme ve zprávci zařízení ve Windows.
3.) Pro nastavení programátoru lze použít i interní soubory Bascomu, které taky obsahují "native driver STK500", potom nenastavujeme cestu. Jistější je ale použít STK500 z AVR Studia.

Obr. č. 53 Nastavení Bascomu

Pokud máme programovací prostředí nastaveno, můžeme začít se psáním základních jednoduchých programů pro ujasnění základních funkcí Bascomu a abychom vyzkoušeli jak Bascom komunikuje s chipem. Samozřejmě musíme mít naši redukci RS232 to USB zapojenou v PC, ze kterého budeme Chip programovat, konektor RS232 musí být zapojen v programátoru STK500 a chip musí být osazen ve správném slotu programátoru STK500. Propojky programátoru musí být zapojeny podle manuálu k STK500 a programátor musí být připojen k napájení a zapnut.

```
Ren pro práci v simulátoru povol direktivu sia
Ren -----
Ren $sin
Ren -----
Ren definice chipu a rychlosti krystalu
$regfile = 'm8535.dat'
$crystal = 1000000
Ren nastavení vstupních a výstupních portu
Config PORTA = Output
Ren při obrácené logice nastavit portx = 255
Ren přejmenování proměných pro lepší přehlednost
1 Alias On
0 Alias Off
Ren HLAVNI PROGRAM
Ren *****
Do
PORTA = Off
PORTA.0 = On
PORTA.2 = On
PORTA.4 = On
PORTA.6 = On
Wait = 2
PORTA = Off
PORTA.1 = On
PORTA.3 = On
PORTA.5 = On
PORTA.7 = On
Wait = 2
Loop
Ren konec programu
End
```

Na obrázku č. 54 vidíme jednoduchý program pro modul ledky. \$regfile udává, jaký je používaný druh mikrokontroléru a je nastaven automaticky po nastavení chipu obr. č. 53. \$crystal definuje krystal momentálně používáme interní oscilátor, jehož rychlost je 1MHz. Config PORTA nastavuje PORTA jako výstupní. Lze také nastavit jako výstupní pouze některé piny, popřípadě nastavit některé výstupní a některé jako vstupní podle potřeby. Dále vidíme přejmenování 1 a 0 na On a Off pro lepší přehlednost kódu.

Obr. č. 54 Program ledky

Pokud by docházelo k tomu, že příkaz Off by rozsvěcoval ledky, je potřeba nastavit obrácenou logiku příkazem PORTA = 255. Dále následuje tělo samotného programu. Jedná se o jednoduchou smyčku, která se neustále opakuje. V první části jsou vypnuty všechny ledky, z žádného pinu není vysílán signál, piny jsou v log. 0. Vzápětí jsou zvednuty sudé piny do log.1. Rozsvítí se ledky připojené k těmto pinům. Následuje pauza 2 sekundy. Poté jsou všechny piny portu A opět nastaveny do log.0 sudé ledky zhasnou. Poté se zvednou liché piny do log. 1. Rozsvítí se ledky připojené k

těmto pinům. Následuje pauza 2 sekundy. Tento proces se neustále opakuje. Tedy modul ledky bude blikat podle popisu. Po napsání programu zmáčkneme tlačítko F7, které daný program zkompile a vytvoří soubor pro mikrokontrolér. Tlačítkem F4 se program nahraje na chip. Můžeme vyjmout chip a vložit ho do vyrobené desky a připojit na PORTA modul ledky. Po připojení napájení by měl modul blikat.

```
Ren pro práci v simulátoru povol direktivu sia.
Ren -----
Ren $sia
Ren -----

Ren definice chipu a rychlosti krystalu
$regfile = 'm8535.dat'
$crystal = 1000000

Ren nastavení vstupních a výstupních portu
Config PORTA = Output
Config PORTC = Input

Ren při obrácené logice nastavit portx = 255

Ren přejmenování proměných pro lepší přehlednost
1 Alias On
0 Alias Off
Tlacitkol Alias PORTC.0

Ren HLAVNI PROGRAM
Ren *****
Do
  If Tlacitkol = On Then
    Waitms = 200
    If Tlacitkol = On Then
      PORTA = 1
    End If
  Else
    PORTA = Off
    PORTA.0 = On
    PORTA.2 = On
    PORTA.4 = On
    PORTA.6 = On

    Waitms = 500

    PORTA = Off
    PORTA.1 = On
    PORTA.3 = On
    PORTA.5 = On
    PORTA.7 = On

    Wait = 500
  End If
Loop
End
```

Na obrázku č. 55 vidíme možnou ukázkou programu kombinující modul ledky s modulem tlačítka. Tento program se od programu čistě pro ledky liší tím, že musíme kromě výstupního portu definovat i port vstupní, tedy port, na který připojíme modul tlačítka. V našem případě to je PORTC.

Program pracuje následovně: pokud je stisknuté tlačítko, na pinu 0 portu C je detekován signál, program počká 200ms kvůli zákmitu tlačítka, popřípadě pouhého neúmyslného zavadění o tlačítko. Pokud je tlačítko zmáčknuto i po 200 milisekundách rozsvítí se všechny ledky na na portu A, celý portA je v log. 1.

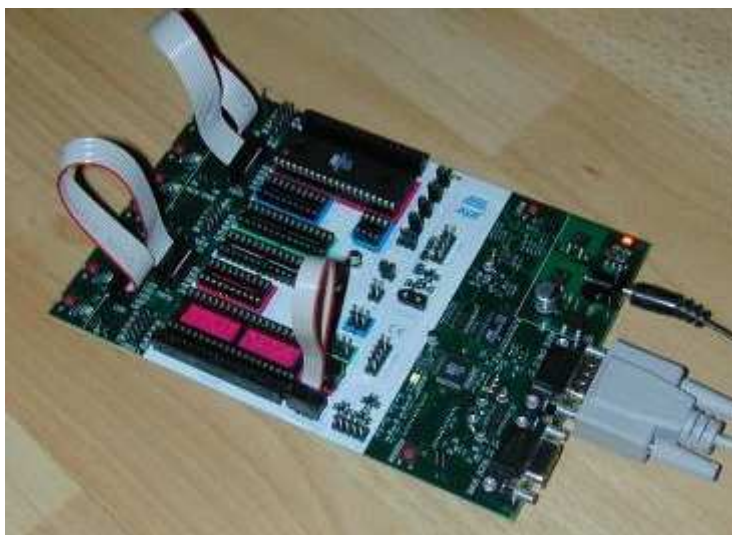
Obr. č. 55 Program Tlačítka, ledky

Pokud tlačítko zmáčknuté není, proběhne program ledky, tedy střídání blikání lichých a sudých ledek. Tento proces se neustále opakuje, protože je v programu smyčka. Po zkompileování a nahrání programu na chip umístíme chip do vyrobené desky a připojíme potřebné moduly. Modul tlačítka na vstupní port C a modul ledky na výstupní port A. Po přivedení napájení se začne vykonávat program.

Samozřejmě, že variací daného programu může být mnohem více, složitějších či jednodušších, principiálně se budou dané příkazy opakovat.

10 Programátory

Mnou používaný programátor, který jsem měl zapůjčený od pana Ing. Michala Šerýho, byl STK500 flash microcontroller starter kit vyráběný firmou Atmel. K tomuto programátoru je dostupný manuál, ve kterém jsou popsány podrobně jeho funkce. Pro spojení s PC používá sériovou linku RS232. Chip se podle druhu umísťuje do příslušných slotů na desce. Je také potřeba propojit konektory, kterých STK500 používá při programování s příslušnou patičí. Tyto patice jsou pro lepší orientaci rozlišeny barevně. STK500 také umožňuje IPS programování, používání externího krystalu atd. Má opravdu mnoho funkcí včetně svého modulu led a modulu s tlačítky. Napájecí napětí je od 10V do 15V, z důvodů možnosti paralelního a sériového vysokonapěťového programování AVR zařízení. STK 500 má také integrovanou 2Mbit flash paměť na desce. [16]



Obr. č. 56 STK500 microcontroller starter kit, převzato a upraveno z [17]

Programátorů je samozřejmě více druhů, například momentálně má k dispozici Atmel novější model STK600, nebo se začínají rozmáhat ISP programátory do USB, přičemž jejich cena je zhruba 500Kč. Tyto USB programátory dostačují pro naprogramování chipu, ale neposkytují takové možnosti jako STK500 či STK600, které jsou ale mnohem dražší.

11 Závěr

Na závěr bych shrnul mé snažení a práci na výukové stavebnici s mikrokontrolérem AVR Atmel asi takto. Ke splnění cílů diplomové práce bylo potřeba prohloubení mých teoretických a praktických znalostí a dovedností. Některé věci, jako programování v Bascomu, či podrobnější práce s programem Eagle, byly úplně nové. Většina mé práce mě velice bavila, ať už z teoretické, či z praktické části. Pro mě nejzajímavější činnost, kromě prohlubování znalostí o mikrokontroléru, byl návrh a realizace desky a modulů. Nejméně zajímavé pro mně bylo samotné programování, protože vím, že programování nebude nikdy můj koníček. Nicméně musím říct, že je velice příjemný pocit, když po veškerém úsilí jako je návrh, výroba, ožívování, programování atd. stavebnice funguje, byť například pouze s jednoduchým programem, který rozblíká ledky.

Myslím, že se vytvoření stavebnice povedlo v rámci mých znalostí a dovedností dobře. Určitě by se našly věci, které bych teď řešil jinak, což je dáno získáváním zkušeností v průběhu řešení problému a zpětné reflexe výsledků. I přes to si myslím, že stavebnice splňuje požadavky, které na ní byly kladeny. Nevím přesně, kolik času mi vytvoření stavebnice zabralo, můžu říct že to byla práce zdlouhavá a leckdy časově náročná. Čas od času jsem se ocitl ve slepé uličce a nad vyřešením daného problému jsem dlouho přemýšlel, popřípadě chodil pro rady za panem Ing. Michalem Šerým. Stavebnice jako taková bude přiložena k diplomové práci s diskem, který bude obsahovat všechny obrázky, schémata a masky plošných spojů, tak mnou vytvořené programy.

Co se obrázků týká, všechny, které jsem převzal a upravil jsou náležitě ocitovány stejně jako odstavce v textu. Pokud citace u obrázku není, obrázek jsem vytvořil sám.

Jsem rád, že jsem pracoval právě na této diplomové práci, protože si myslím, že se mi získané znalosti a dovednosti budou v budoucnu hodit a použiji je.

12 Použité prameny a literatura

- [1] Atmel Corporation. <http://www2.atmel.com/About/corporate/factsheet.aspx> (accessed Dec 6, 2011).
- [2] Atmel Corporation.
http://www.atmel.com/products/overview_automotive.asp?source=cms&category_id=154&source=global_nav (accessed Dec 6, 2011).
- [3] Atmel Corporation.
http://www.atmel.com/dyn/general/document_search_results.asp?docType=Datasheet&prodFamily=607&searchhome.x=24&searchhome.y=10 (accessed Dec 6, 2011).
- [4] *Datasheet ATmega8535(L)* [online]; Atmel corporation, 2006;
http://www.atmel.com/dyn/resources/prod_documents/doc2502.pdf (accessed Dec 6, 2011).
- [5] Plíva, Z. *Eagle prakticky: řešení problému při běžné práci*, 2.nd ed.; BEN - technická literatura: Praha, 2007.
- [6] Hrabovský, M.; Juránek, A. *Eagle pro začátečníky*, 2.nd ed.; BEN - technická literatura: Praha, 2007.
- [7] *Datasheet 78S05* [online]; SGS-Thomson microelectronic,
<http://html.alldatasheet.com/htmlpdf/22621/STMICROELECTRONICS/L78S05C/1621/1/L78S05C.html> (accessed Dec 14, 2011).
- [8] *Datasheet MAX220-MAX249*, 11th ed. [online]; MAXIM, 2003.
<http://www.datasheetcatalog.org/datasheet/maxim/MAX220-MAX249.pdf> (accessed Dec 14, 2011).
- [9] *Datasheet STK500* [online]; ATMEL, 2003.
http://www.atmel.com/dyn/resources/prod_documents/doc1925.pdf (accessed Dec 15, 2011).
- [10] Dresler, T. Hardware. http://www.hw.cz/docs/stabiliky/tl431_2.html (accessed Dec 17, 2011).
- [11] *Datasheet KK74AC245N* [online]; Kodenshi,
<http://pdf1.alldatasheet.com/datasheet-pdf/view/203708/KODENSHI/KK74AC245N.html> (accessed Dec 17, 2011).

- [12] *Datasheet DS1621* [online]; Maxim, Dallas semiconductor, 2003. <http://docs-europe.electrocomponents.com/webdocs/07c1/0900766b807c18b7.pdf> (accessed Dec 18, 2011).
- [13] *Datasheet DS18B20* [online]; Maxim, Dallas semiconductor, 2008. <http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf> (accessed Dec 18, 2011).
- [14] Kákona, J.; Lafata, J.; Horkel, M. výroba plošných spojů. MLAB. http://www.mlab.cz/Articles/HowTo/How_to_make_PCB/DOC/HTML/How_to_make_PCB.cs.html (accessed Dec 18, 2011).
- [15] Starý, J.; Kahle, P.; Švandra, J. *Plošné spoje a povrchová montáž*; PC-DIR Real: Brno, 1999.
- [16] *STK500 User Guide* [online]; Atmel, 2003. http://www.atmel.com/dyn/resources/prod_documents/doc1925.pdf (accessed Dec 19, 2011).
- [17] Wikiaalto <https://wiki.aalto.fi/download/attachments/16221372/stk500.jpg?version=1&modificationDate=1200662509000> (accessed Dec 19, 2011).