



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFOMATIKY

FACULTY OF MECHANICAL ENGINEERING

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

NÁVRH A REALIZACE MODELU POSUVNÉ JEDNOTKY

MOVEMENT UNIT DESIGN AND REALIZATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MILAN ŠMARDÁ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. TOMÁŠ MARADA, PH.D

BRNO 2012

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Milan Šmarda

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem c.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh a realizace modelu posuvné jednotky

v anglickém jazyce:

Movement unit design and realization

Stručná charakteristika problematiky úkolu:

Cílem práce je návrh a realizace modelu posuvné jednotky dle předlohy modelu EDU-mod, které se nacházejí v laboratoři programovatelných automatů A1/731a. Model bude realizován mikrokontrolérem ATMEGA. Funkčnost navrženého a realizovaného modelu bude ověřena řízením pomocí PLC.

Cíle bakalářské práce:

1. Seznamte se se stávajícím modelem posuvné jednotky v laboratorii programovatelných automatů.
2. Navrhněte obdobný model s mikrokontrolérem ATMEGA.
3. Navržený model realizujte.
4. Funkčnost realizovaného modelu ověřte řízením z PLC.

Seznam odborné literatury:

- [1] Šmejkal, L., Martinásková, M., PLC a automatizace, Praha: BEN, 1999.
- [2] Firemní materiály o programovatelných automatech fy Siemens S7-200.
- [3] <http://www.robotika.cz>
- [4] <http://www.robotika.sk>

Vedoucí bakalářské práce: Ing. Tomáš Marada, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

V Brně, dne 3.11.2011

L.S.

Ing. Jan Roupec, Ph.D. prof. RNDr.
Ředitel ústavu

Miroslav Doupovec, CSc.
Děkan fakulty

Anotace

Bakalářská práce se zabývá návrhem modelu posuvové hydraulické jednotky podle předlohy téže jednotky od firmy EDU-mot, jež slouží pro výuku řídicích systémů. Je v ní popsán celý vývoj modulu od návrhu přes konstrukci až po výrobu celého řízeného prvku. Tedy návrh základní desky, která se tímto stává univerzální pro jakýkoliv modul, jenž může dosáhnout maximálního počtu osmi vstupů a osmi výstupů. Model je realizován mikrokontrolérem ATmega 16, jenž byl programován pomocí softwaru AVR. Na tento modul byly zhotoveny úlohy v programovacím prostředí STEP7 Micro/Win za pomoci jazyka STL, které dokládají jeho funkčnost.

Anotation

This bachelor work deals with construction of model of hydraulic moving unit. As pattern we used model of EDU-mot company, which is used for especially for education purpose. There is describe all development from scheme to manufacturing of all part that model. We talk about design of motherboard, which is now universal for every model, which has eight inputs and eight outputs. Model is controlled by microcontroller AT mega 16. That microcontroller was programmed by programming environment AVR. And whole model was controlled by program from environment STEP7 Micro/win by language STL.

Klíčová slova

PLC, EDU-mod, Atmel ATmega128, Siemens Simatic S7, STEP7

Key world

PLC, EDU-mod, Atmel ATmega128, Siemens Simatic S7, STEP7

Prohlášení o originalitě

Prohlašuji, že jsem tuto práci vypracoval samostatně bez cizí pomoci.
Pouze za pomoci použité literatury

Bibliografická citace

ŠMARDA, M. *Návrh a realizace modelu posuvné jednotky*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 55 s. Vedoucí bakalářské práce Ing. Tomáš Marada, Ph.D.

Poděkování

Děkuji svému vedoucímu bakalářské práce ing. Tomáši Maradovi, Ph.D. za poskytnuté rady a materiály, jež mi byly velice nápomocny v průběhu celé práce.

Obsah

Zadání bakalářské práce	3
Anotace	5
Prohlášení o originalitě	7
Poděkování	9
Obsah	11
1 Úvod	13
2 Moduly EDU-mod	15
2.1 Určení pomůcky EDU-mod	15
2.2 Vstupní a výstupní signály – připojení	15
2.3 O modulu hydraulické posuvové jednotky	16
3 Návrh desky	17
3.1 Původní zapojení	17
3.2 Prvky funkce zapojení	17
3.2.1 AT mega 16-A	17
3.2.2 Stabilizátor	18
3.3 Komunikace PLC-ATmega	20
3.3.1 TTL logika	21
3.3.2 Napěťové úrovně PLC	21
3.3.3 Výstupy z ATmegy	21
3.3.4 Vstupy do ATmegy	22
3.3.5 Zobrazování pomocí led diod	23
4 Realizace nového návrhu desky	25
4.1 Rýsovací program EAGLE 5.3.0	25
4.2 Schémata	25
4.2.1 Editor schémat	25
4.2.2 Kreslení schématu	26
4.3 Spoje	26
4.3.1 Editor spojů	26
4.3.2 Kreslení spojů	27
4.4 Autorouter	27
5 Výroba univerzální desky	29
5.1 Tisk	29
5.2 Osvět	29
5.3 Vyvolání	30
5.4 Leptání	30
5.5 Dokončovací operace	31
6 Programování modelu	33
6.1 AVR studio	33
6.1.1 Popis vývojového diagramu	34
6.1.2 Popis programu pro ATmega15	35

6.2	AVR ISP programmer	37
6.2.1	UniProg USB	38
6.3	Program proteus	39
6.4	Programovatelné automaty siemens	39
6.5	Programové prostředí micro/win	41
6.5.1	Vlastnosti editoru STL	41
6.5.2	Vlastnosti editoru LAD	42
6.5.3	Vlastnosti editoru FBD	43
6.6	Testovací úlohy	43
6.6.1	úloha 1	44
6.6.2	úloha 2	45
6.6.3	úloha 3	45
6.6.4	úloha 4	47
7	Závěr	49
	Seznam použité literatury	51
	Seznam příloh	53

1 ÚVOD

V dnešní době jsou nejpoužívanějšími zařízeními programovatelné automaty (PLC). Kvůli jejich všestrannosti a mnohým výhodám oproti klasickým PC jsou nasazovány do provozů mnohem častěji. Najdeme je tedy ve výrobních i nevýrobních procesech. Jedná se o průmyslový počítač, který je odolnější proti vnějším vlivům. To podstatné, jak se liší od klasického PC je ovšem to, že provádějí program cyklicky a jejich periferie jsou uzpůsobeny pro napojení na technologické procesy.

Dnešní automatizace se bez nich prakticky neobejde. Proto byla vyvinuta mnohá zařízení na simulaci různých pracovních situací. Jsou to převážně modely EDU-mod, které mě velmi zaujaly.

V mé práci se tedy zabývám návrhem univerzální desky, která by měla pomoci při návrhů dalších modelů.

2 Moduly EDU-mod

Modely EDU-mod jsou souborem modelů simulující technické procesy a jsou určeny především k výukovým účelům. Je na nich učeno řízení logických systémů realizovaných pomocí programovatelných automatů.

Automaty jsou dodávány s řadou učebních textů a dokumentace, která popisuje vývojové prostředí xPRO. Také obsahuje příručku programování PLC Tecomat.



Obr.č.1 Model hydraulické posuvové jednotky

2.1 Určení pomůcky EDU-mod

Modely EDU-mod jsou vyráběny ve dvou řadách, jež se dělí podle napěťové úrovně. Jsou zde modely řady 24V a řady 5V.

- Modely řady 24V je možno použít pro libovolný typ PLC systému. Tady je vidět jejich univerzálnost. Vstupní a výstupní signály má definovány proti zápornému vodiči. Opačnou polaritu signálů je možno řešit přizpůsobovacími členy.
- Modely řady 5V dovolují spojení s logickými automaty realizovanými na bázi stavebnic číslicových IO, programovatelných logických polí (PLD), procesorových obvodů atd.

2.2 Vstupní a výstupní signály – připojení

Vstupní a výstupní signály EDU-mod jsou vyvedeny na dvacetipólový konektor zajišťující propojení plochým kabelem s rozbočovacím modulem s mechanickým provedením shodným s periferiemi EDUtec. Rozbočovač obsahuje čtyři konektory Cannon 9 (2 vstupní a 2 výstupní) pro připojení maximálně 8 vstupních a 8 výstupních binárních signálů z nebo do libovolného systému. Chceme-li tedy použít stačí jej kabelem připojit a do odpovídajících konektorů zapojit řídicí systém.

2.3 O modulu hydraulické posuvové jednotky

Funkce

- Model popisuje posouvání pístu pomocí LED diod, z čehož čtyři mají funkci snímačů polohy a to K1, K2 K3 a K4. Umožňuje plynulou posuv pístu, a to buď rychloposuvem pomocí motorů EM1, EM2 nebo posuvem pracovním za pomocí motoru EM3.

Inicializace

- Po spuštění modelu, popřípadě po jeho restartování se píst nastaví do inicializační polohy, tj. poloha snímače K1.

Hlášení chyby

- Model dokáže detekovat dva druhy chyb. Při jejich detekování nás upozorní červená LED dioda EER.
- První chybové hlášení se objeví při přjetí krajních poloh snímačů polohy K1 a K4 a rozsvítí se červená LED dioda. Model se vrátí do výchozího stavu po stisknutí tlačítka RESET
- Druhé chybové hlášení spočívá ve zpuštění obou motorů současně EM1 a EM2. V tuto chvíli začne červená LED dioda blikat. Po odstranění závady může píst pokračovat v činnosti bez jakéhokoli restartování.

3 Návrh desky

3.1 Původní zapojení

Pro návrh naší univerzální desky byl nejprve prozkoumán původní model posuvové jednotky od firmy EDU-mod. Ten obsahoval samozřejmě jiný procesor a jiné množství vstupů a výstupů. Obsahoval pouze tři vstupy a čtyři výstupy.

Pro naše potřeby to bylo nedostačující. Abychom vybudovali základní desku, která by byla univerzální pro připojení k automatu SIEMENS, potřebovali jsme obsadit osm vstupů a osm výstupů. Tedy nechali jsme stejný systém zapojení elektronických prvků. Pouze jsme jej rozšířili. Toto rozšíření se týkalo především výměně klasických elektrických součástek za součástky SMD, jež mají velmi malé rozměry, zvýšení počtu optočlenů a tranzistorů a v neposlední řadě výměně mikrokontroléru ATmega-8 za mikrokontrolér ATmega-16A.

3.2 Prvky – funkce – zapojení

3.2.1 AT mega 16-A

ATmega16A je 8-bitový mikrokontrolér, založený na rozšířené AVR RISC architektuře. Vykonává veškeré instrukce v reálném čase. Dosahuje průchodnosti blížící se 1 MIPS za MHz, což pomáhá programátorovi optimalizovat spotřebu energie ve srovnání s rychlostí zpracování dat.

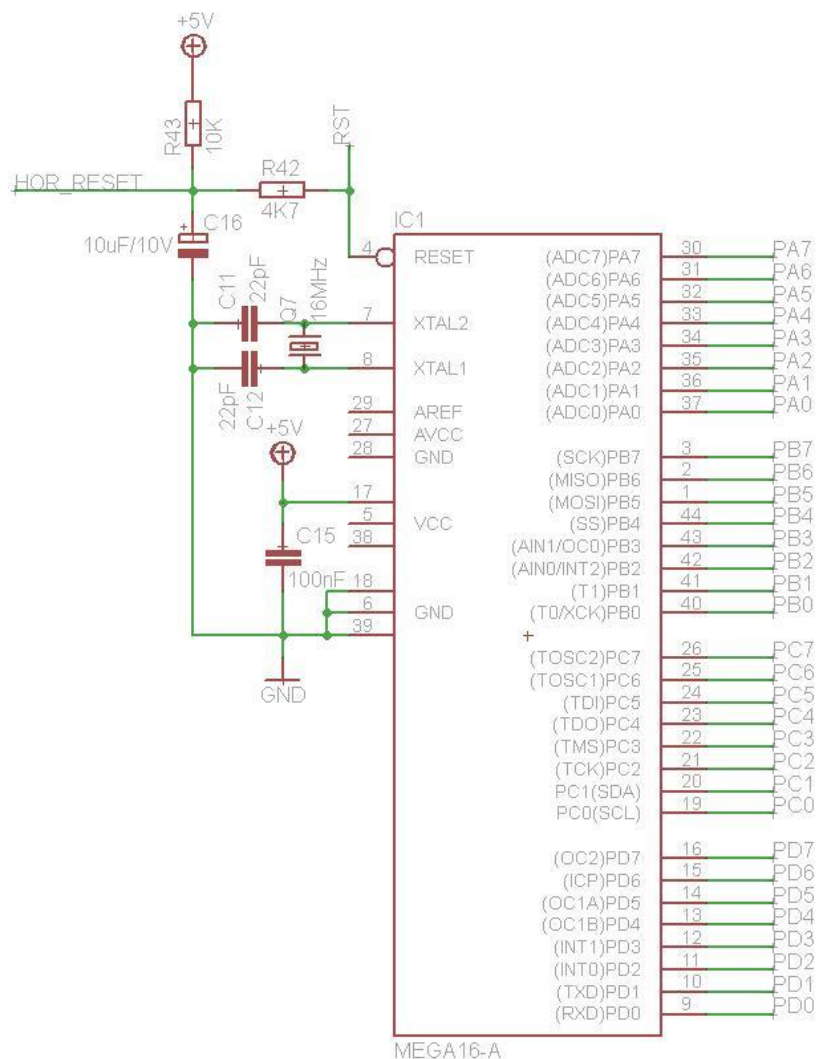
ATmega16A má následující parametry:

- 16K bytů programovatelnou FLASH paměť se schopností čtení při zapisování.
- 512 bytů paměti EEPROM
- 1K byte SRAM
- 32 všeobecných pracovních registrů

Díky tomu, že je počítáno s externím zdrojem, je deska doplněna na o krystal, který je připojen přes kondenzátory o velikosti 22pF na piny XTAL1 a 2.

Obvod pro tetování celého mikrokontroléru je realizovaný pomocí kondenzátoru o velikosti 10μF a rezistory s velikostech $R_{43}=10\,000\Omega$ a $R_{42}=4\,700\Omega$.

K portům PD jsou realizovány jako výstupní a jsou dále napojeny na zobrazovací LED diody na horní desce. Porty PC jsou určeny jako vstupní. Zbylé porty PA a PB jsou ponechány pro případné použití a připojování zobrazovacích LED diod.



Obr.č.2 Schéma zapojení ATmegy-16A

3.2.2 Stabilizátor

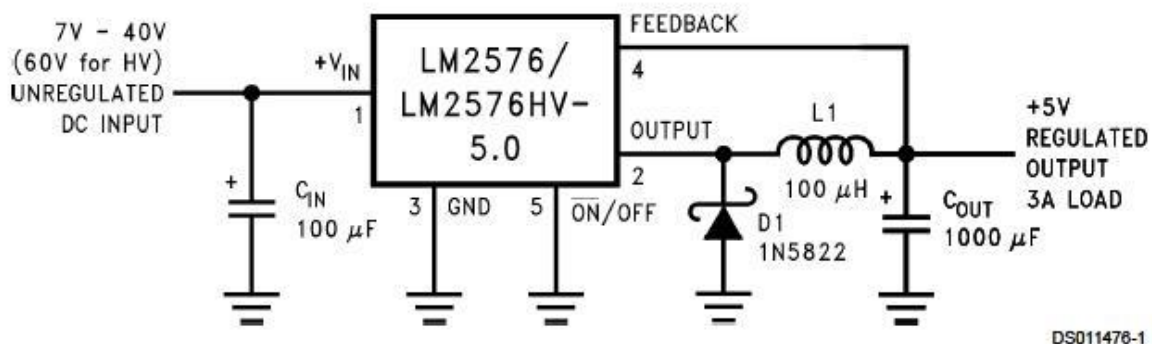
Série regulátorů LM2576 jsou jednolitě integrované obvody. Mohou pracovat až s proudem 3A. Tato zařízení se vyrábí pro fixní výstupní napětí 3,3V, 5V, 12V, 15V a také jako verze s nastavitelným výstupem. Vyžaduje minimální počet externích komponentů. Tyto regulátory jsou velmi jednoduché na používání. Obsahují interní frekvenční kompenzace a fixní frekvenční oscilátor.

Série LM2576 nabízí velmi účinné nahrazení oblíbeného tří-terminálového lineárního regulátoru. Také podstatně zmenšuje velikost chladiče a v některých případech se bez chladiče bez problémů obejde.

Další charakteristikou je přesnost $\pm 4\%$ toleranci výstupního napětí a $\pm 10\%$ toleranci frekvence oscilátoru.

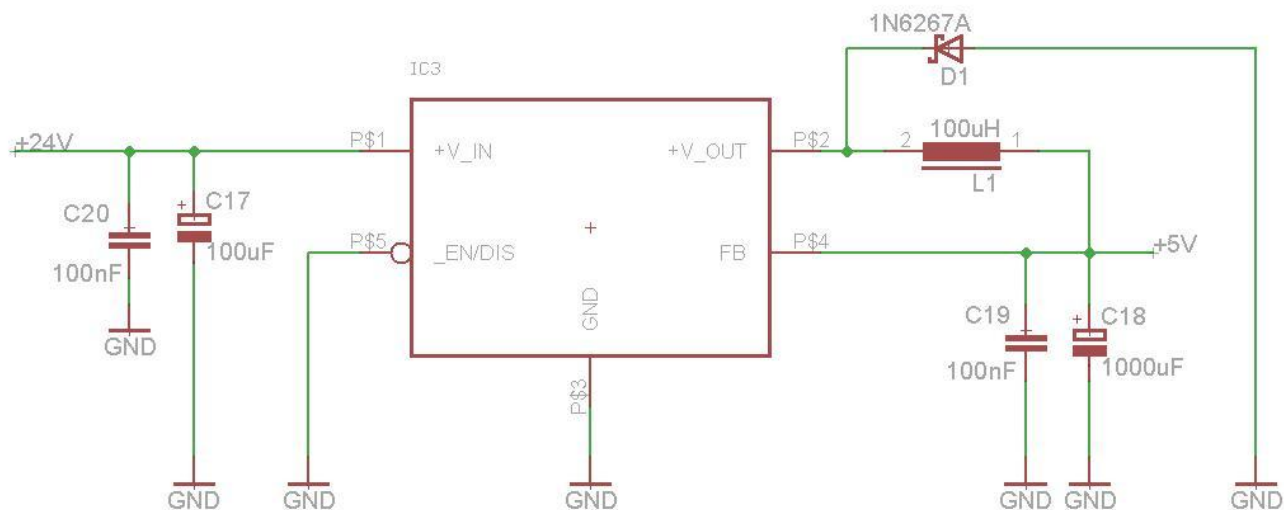
Další vlastnosti:

- 3,3V, 5V, 12V, 15V a verze s nastavitelným výstupem
- Garantován 3A výstupní proud
- Široké rozpětí vstupního napětí do 40V popřípadě až do 60V při verzi HV
- Vyžaduje pouze 4 externí komponenty



Obr.č.3 Původní schéma zapojení stabilizátoru

V našem případě jsme použili téměř stejné schéma, ovšem jsme přidaly navíc dva kondenzátory $C = 100\mu\text{F}$ kvůli plynulosti přechodů.



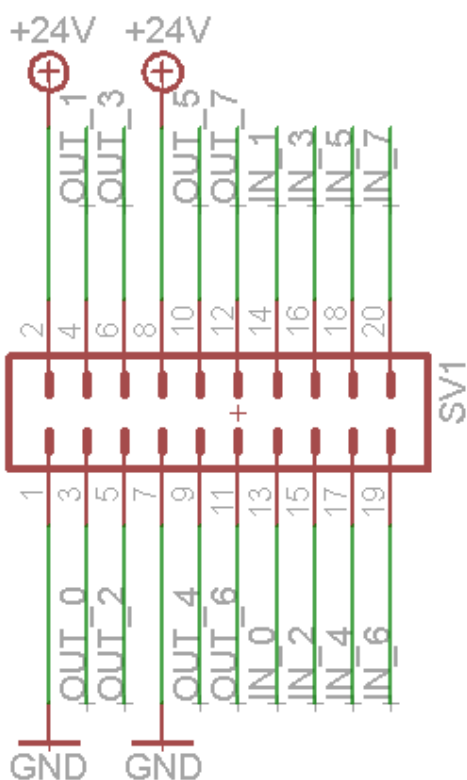
Obr.č.4 Schéma zapojení stabilizátoru

3.3 Komunikace PLC – ATmega

Abychom mohli vytvořit komunikaci mezi mikrokontrolérem /ATmegou, a to na obě strany museli jsme navrhnout vstupní/výstupní obvody tak, aby napětí vstupující do modelu, a rovněž napětí z modelu vystupující mělo odpovídající velikost. Tedy bylo bezpečné pro mikrokontrolér.

Bylo nutno vzít v úvahu informace vstupních/výstupních bran mikrokontroléru a PLC. Tyto byly nalezeny ve svých datasheetech. Podle nichž byly zhotoveny.

Komunikace probíhá stejně jako u klasických modelů EDU-MOD přes dvouřadý 20 pinový konektor. Aby se modely podobaly, co možná nejvíce a jejich univerzalita byla skutečně zachována. Tomuto konektoru jsme nechali stejné zapojení vstupů a výstupů jako původnímu modelu EDU-MOD.



Obr.č.5 Schéma zapojení propojovacího konektoru

Jak je vidět na obrázku č. 5, jsou zde párově připojeny vstupní napětí +24 voltů na porty 2 a 4. Zem – GND je zde zapojena rovněž v páru na porty 1 a 7. Zbytek portů byl použit na připojení vstupů (3 až 6, 9 až 12) a výstupů (13 až 20). Ovšem pro potřeby posuvové jednotky jsou využívány pouze tři vstupní porty a čtyři výstupní porty.

3.3.1 TTL logika

TTL logika je standardem používaným pro implementaci digitálních integrovaných obvodů, vycházejícím z použití technologie bipolárních křemíkových tranzistorů. Tyto obvody používají napájecí napětí od 4,5V po 5,5 V, tedy pro logickou jedničku je napětí 5V a pro logickou nulu 0V. Pro přesnější uvedení logické nuly a jedničky slouží tabulka

TTL logika	Napětí
Logická 0	0V – 0,8V
Logická 1	2V – 5V

Tabulka č.1 Napěťové úrovně TTL logiky

3.3.2 Napěťové úrovně PLC

Programovatelný automat pracuje s různými napěťovými úrovněmi digitálních vstupů a digitálních výstupů. Tyto jsou popsány, pro automaty od firmy SIEMENS, řady Simatic-S7 s CPU 224XP v níže uvedené tabulce. Podrobnější popis se dá ovšem nalézt v technické dokumentaci automatů.

	Digitální vstupy	Digitální výstupy
Logická jednička	15V DC, při 2,5mA 4V DC, při 8mA	L+ minus 0, 4V při max. proudu
Logická nula	5V DC, při 1 mA 1V DC, při 1mA	0,1V DC při zatížení 10000Ω

Tabulka č.2 Napěťové úrovně programovatelného automatu

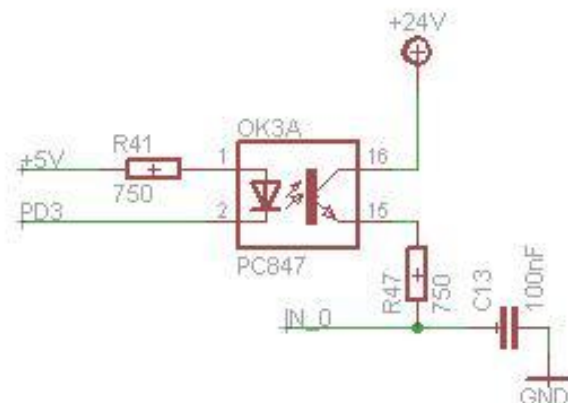
3.3.3 Výstupy z ATmegy

Řízení výstupních signálů je realizováno pomocí jednoduchých optočlenů LTV847. Tento optočlen se skládá ze dvou okruhů zahrnující polovodičové součástky.

Fotoelektrické součástky jsou umístěny v jednom pouzdře a samozřejmě galvanicky odděleny. Je složen z LED diody a fototranzistoru NPN.

Je-li přiveden na vstup optočlenu proud, potřebný k rozsvícení LED diody, fototranzistor se začne otevírat. Díky galvanickému oddělení obou obvodů lze ovládat obvod s vyšším napětím obvodem s napětím nižším a naopak.

V našem případě je na vstupu optočlenu napětí pouhých 5V. To je přes rezistor R41 o velikosti 750Ω vedeno na anodu LED diody. Katoda se spojena s portem PD3. Po otevření tranzistoru je ovládán okruh s napětím 24V. Po otevření fototranzistoru je přivedeno na vstup automatu IN_0 napětí 24V.



Obr.č.6 Schéma zapojení výstupů jdoucích z mikrokontroléru

Výpočet odporu u optočlenu:

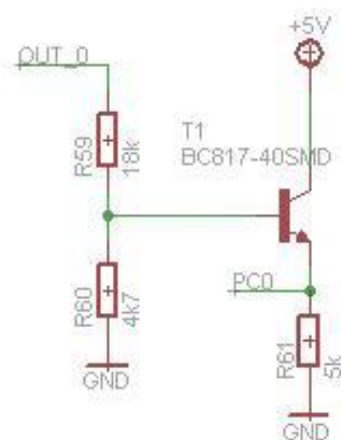
$$I = \frac{U}{R} \Rightarrow R = \frac{U}{I}$$

$$R = \frac{24V}{0,032mA}$$

$$R = 750\Omega$$

3.3.4 Vstupy do ATmegy

Řízení vstupních signálů je realizováno pomocí tranzistorů NPN. Kde z výstupu automatu OUT je přiveden proud přes rezistor 18K. Ten otevře závoru báze a umožní sepnutí napětí v obvodu kolektor-emitor 5V k zemi. Toto napětí projde tranzistorem a vstoupí na port mikrokontroléru AT mega, jenž je označen jako PC.



Obr.č.7 Schéma zapojení vstupů jdoucích do mikrokontroléru

Po samotném návrhu zapojení jsme provedli kontrolní výpočet pro zjištění velikosti odporů pro dosažení optimálního napětí. Tedy z původních 24V jsme potřebovaly udělat 5V. z toho důvodu zde byla použita dělička napětí.

Výpočtem jsme tedy zjistily, že velikosti odporů musejí být $R_1=18000\Omega$ a $R_2=4700\Omega$, aby napětí vstupující do tranzistoru dosahovalo naší požadované hodnoty 5V. Po zjištění hodnot odporů a jim odpovídajícího výstupního napětí jsme byly schopni toto zapojení otestovat. Zkoušení správnosti výpočtu hodnot a zapojení probíhalo na nepájivém poli a to s kladným výsledkem.

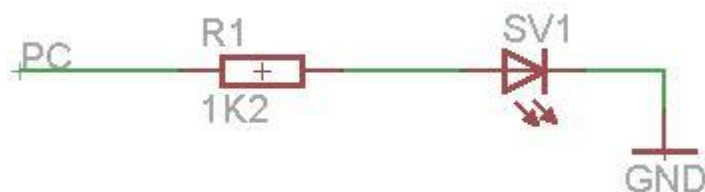
Výpočet pro děličku napětí:

$$\begin{aligned}
 U_{\text{výstupní}} &= U_{\text{vstupní}} \frac{R_2}{R_1 + R_2} = OUT_0 \frac{R_{60}}{R_{59} + R_{60}} = \\
 &= 24V \frac{4\,700\Omega}{18\,000\Omega + 4\,700\Omega} \\
 U_{\text{výstupní}} &= 4,969V
 \end{aligned}$$

3.3.5 Zobrazování pomocí LED diod

Jakožto zobrazovací prvek pro všechny vstupy a výstupy nám vhodně poslouží LED diody. Tyto byly zapojeny z portu AT megy 16 jenž nám dává napětí 5V přes odpor směrem k zemi.

Díky odporu můžeme regulovat intenzitu svitu LED diody. Ovšem to, abychom z diod dostali optimální intenzitu osvětlení, jsme museli vypočítat velikost odporu, přes který prochází proud.



Obr.č.8 Schéma zapojení LED diod

Výpočet velikosti odporu na Pro LED diodu a proud:

$$\begin{aligned}
 R &= \frac{(U_{\text{vstupní}} - U_{LED})}{I_{LED}} = \frac{5V - 2,1V}{20\,mA} = \frac{2,9V}{0,02A} \\
 R &= 145\Omega \\
 I &= \frac{(U_{\text{vstupní}} - U_{LED})}{R} = \frac{5V - 2,1V}{1200\Omega} = 0,0024A
 \end{aligned}$$

Po dosazení do výše uvedeného vzorce nám vyšla hodnota odporu $R = 145\Omega$. Tento výsledek byl tedy přijatelný proto, aby nedošlo k nevratnému poškození naší

LED diody, ovšem před uvedením prvků do provozu bylo toto zapojení a hodnota odporů odzkoušena na nepájivém poli.

Zde bylo zjištěno, že zapojení je v pořádku, ovšem velikost odporu R , je příliš malá. Tedy LED dioda svítila více, než bylo zapotřebí pro naše účely. Z toho důvodu jsme se rozhodli změnit jeho velikost z vypočítaných 140Ω na odhadnutých 1200Ω . Tato markantní změna měla za následek pouze snížení intenzity světla vyvazávaného diodou.

4 Realizace nového návrhu desky

4.1 Rýsovací program EAGLE 5.3.0

Je to program zabývající se návrhem plošných spojů. Název EAGLE zkracuje původní název Easily Applicable Graphical Layout Editor. Program se skládá ze tří hlavních částí

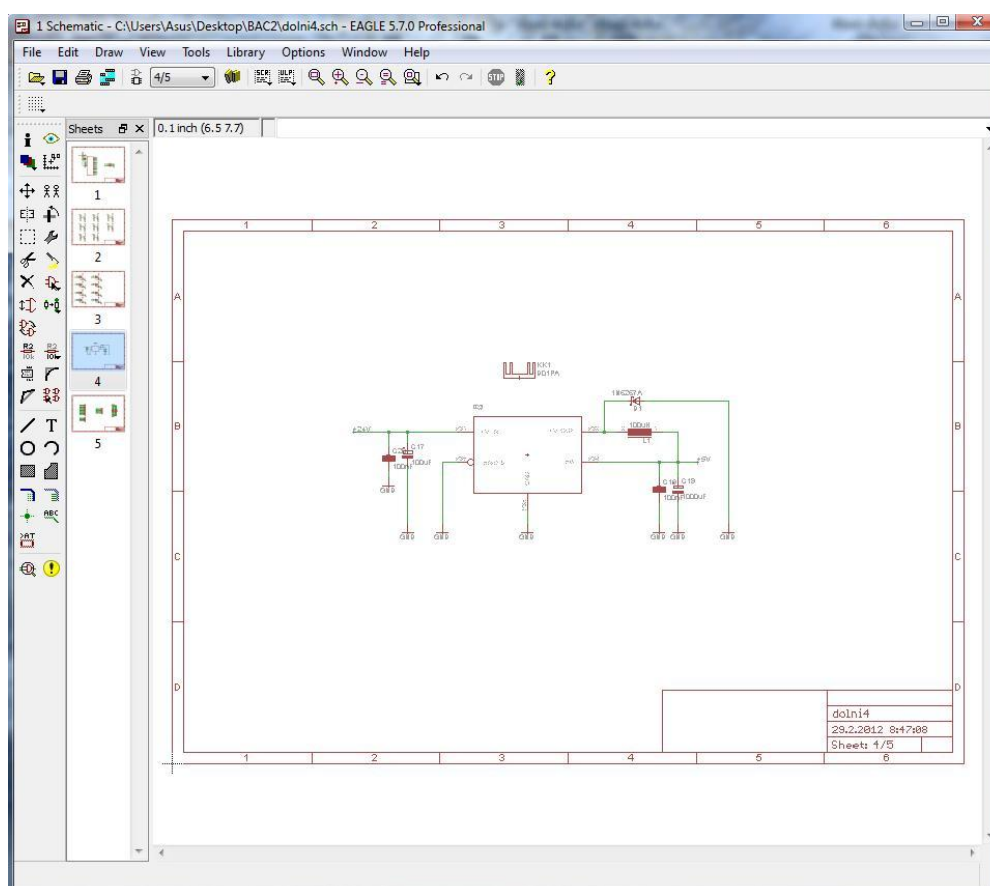
- Editor schémat
- Editor spojů
- Autorouter

Tyto části je ovládají v jednom uživatelském prostředí

4.2 Schémata

4.2.1 Editor schémat

Editor Schémat má možnost použití až 99 listů jednoho schématu s kontrolou elektrických pravidel zapojení. Je zde taktéž možno použití funkce prohazování hradel a pinů. A je opatřen funkcí, která vytvoří desku ze schématu jediným příkazem.



Obr.č.9 Znáznornění editoru schémat EAGLE 5.7.0

4.2.2 Kreslení schématu

Při kreslení schématu je nutno postupovat velice systematicky. A to, nejdříve pomocí nástroje ADD přidat prvky, jež jsou potřeba pro zapojení jedné větve schématu. Potom se prvky mezi sebou propojují. Toto probíhá za pomoci nástroje WIRE.

Ovšem je nutno se ujistit jsou-li spolu opravdu spojeny. V opačném případě se může stát, že jsme prvek s cestou na sebe pouze položily a při transformaci do editoru spojů nebudeme mít tyto cesty vyznačeny. Když spojujeme prvky, jež jsou např. na více stranách schématu, děláme to pomocí popisků.

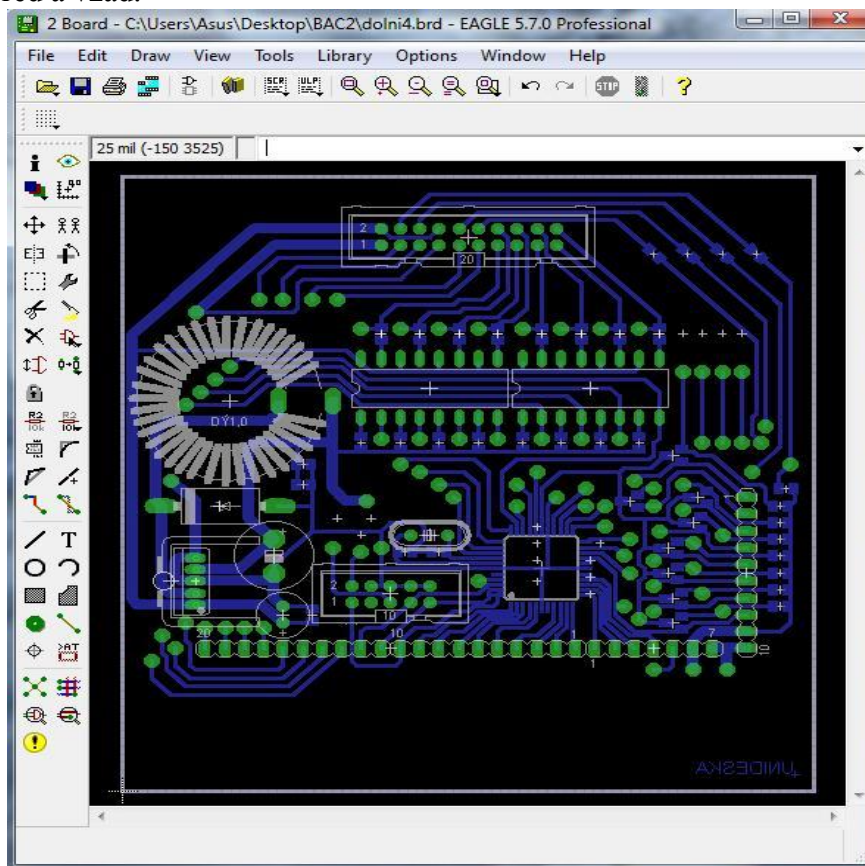
Prostě přiřadíme jeden prvek k druhému. Dále si pomocí funkce LABEL poznačíme hodnoty prvků, popřípadě označíme názvy cest nebo prvků. K tomu nám slouží funkce NAME. Pokud jsme si takto schéma vytvořili, můžeme se vydat do editoru spojů, za pomoci tlačítka SWITCH.

4.3 Spoje

4.3.1 Editor spojů

Editor spojů dokáže vytvořit spoje na desce velké maximálně 1,6 x 1,6 m tedy 64 inch² s rozlišením až 1/10 000 mm (0,1 mikronu) používá 16 signálových vrstev a jeho knihovny zahrnují součástky klasické i SMD součástky.

Umožňuje taktéž, a to považuji za jednu z velkých výhod, vytvoření součástek vlastních. Popřípadě upravení součástek nacházejících se ve stávajících knihovnách. Je zde taktéž možnost kopírování a vkládání součástek. A samozřejmě nejdůležitější funkce vpřed a vzad.



Obr.č.10 Znáznornění editoru spojů EAGLE 5.7.0

4.3.2 Výroba spojů

Vytvoření schéma spojů je snazší, než kreslení schématu, to už z toho důvodu, že máme vše připraveno. Ovšem je pracnější. Po přepnutí z editoru schémat jsme se ocitly před prázdnou deskou a „kupou“ prvků, které je zapotřebí spojit elektrickými cestami.

Je zde také nutno řídit se určitou posloupností tvorby a jistých pravidel. Na začátku je dobré smazat naši automaticky vygenerovanou velikost desky a vytvořit novou dle našich parametrů. Pro nakreslení desky slouží tlačítko WIRE, nastavíme si jiný druh cest, než budeme používat. Nejlépe dimension. Vytvoříme obrys desky a přepneme mřížku pomocí GRID, do hodnot mil ve které jsou malovány prvky z knihoven. Následně si přemístíme prvky do prostoru desky.

4.4 Autorouter

Využívá možnosti přepojování a znovu vytvoření elektrických cest. Umožňuje propojení a realizování až šestnácti různých signálových vrstev cest. Dále umožňuje strategii propojování nastavitelnou uživatelem pomocí váhových faktorů.

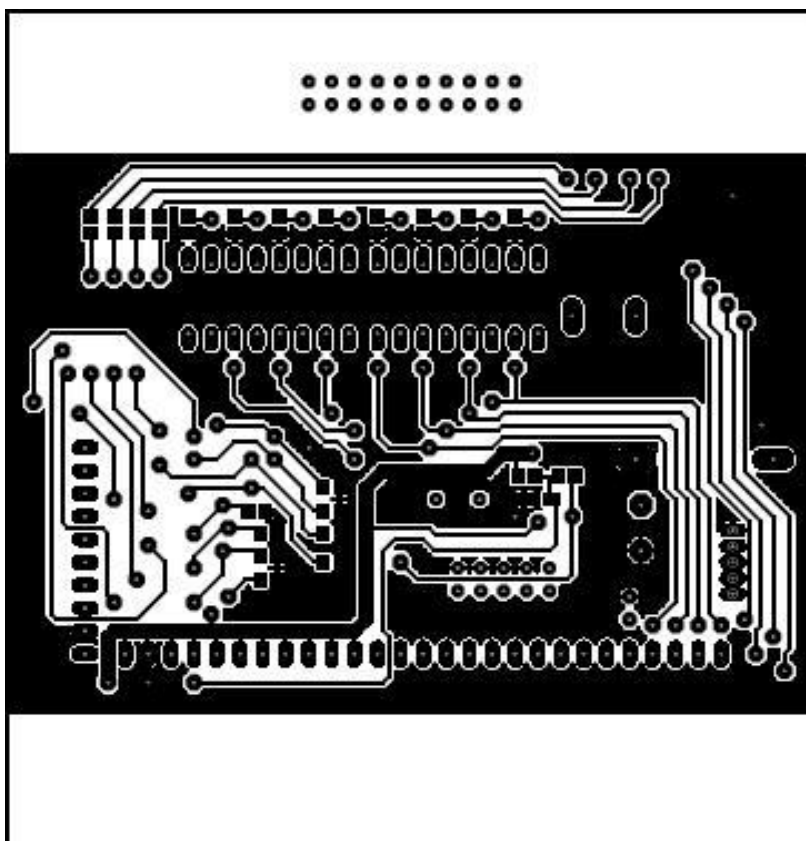
Samozřejmě je zde také možnost automatického propojování cest pomocí stejnojmenného příkazu AUTOROUTER, ovšem z vlastní zkušenosti se nevyplatí tento příkaz využívat. Jeho vhodnost je pouze pro zapojování velmi jednoduchých návrhů plošných spojů. Když se ovšem realizuje složitější schéma, tak nebyly propojeny všechny cesty. Přehlednost velice klesla, přičemž bylo téměř maximální zaplnění plochy určené pro plošné spoje.

5 Výroba univerzální desky

DPS se používá v elektronice pro upevnění a elektrické propojení elektronických součástek. Tyto součástky jsou připevněny k měděné vrstvě na laminátové desce. Elektronické součástky jsou připájeny za svoje koncové vývody, cínovou pájkou k měděné vrstvě.

5.1 Tisk

Nejprve bylo nutno si vytvořit tištěnou fólii, pomocí které by se vysvítilo námi vytvořené schéma zapojení plošného spoje. Tedy jsme si nechali vytvořit tisk našeho zapojení pouze ve velmi syté černé barvě na fólii. Z této fólie jsme vystřížením a spojením vytvořili „kapsu“, do které se vložila deska s fotocitlivou vrstvou patřičných rozměrů.

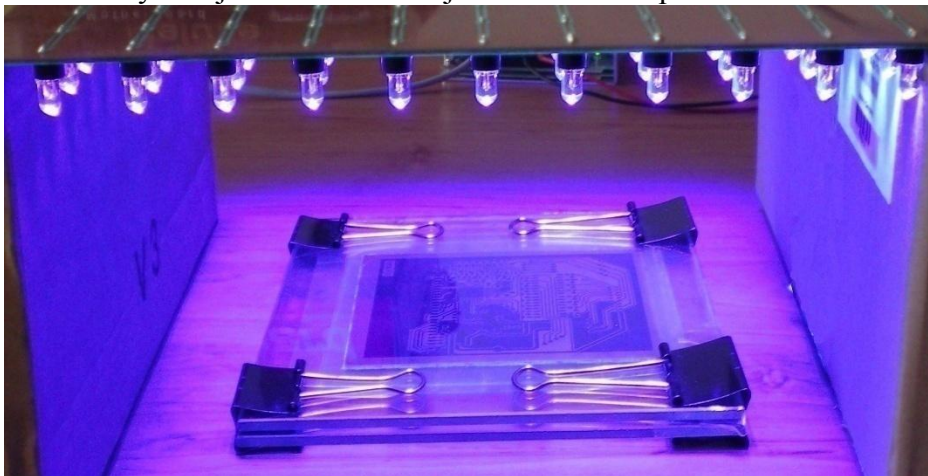


Obr.č.11 Předloha pro tisk na fólii

5.2 Osvit

Samotný osvit byl nejjednodušší část samotného procesu. Pro něj bylo nutno pouze mít připravený panel s diodami, skleněné desky, mezi které se vkládala naše připravená „kapsička“ s deskou a samozřejmě desku. Skleněné desky jsme patřičně očistili, pomocí acetonu, aby na nich nebylo ani stopa po výrazných čmouhách nebo mechanických nečistotách, které by mohly proces osvitu kazit. Mezi ně jsme vložili

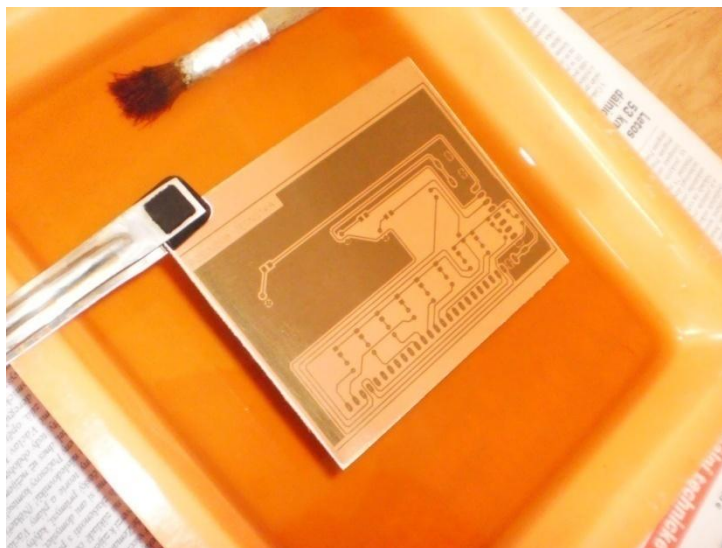
kapsičku s deskou a pečlivě jsme ji zajistili proti případnému posuvu. Poté jsme ji vložili pod světelný zdroj a každou vrstvu jsme osvěcovali přibližně 4-6 minut.



Obr.č.12 Průběh osvitů pod UV světlem

5.3 Vyvolání

Po osvitě přišla na řadu vyvolávací procedura, která měla odstranit lak z osvětlené části DPS. Tento proces se odehrával pomocí vývojky. Vývojka je roztok 1,5% NaOH (Hydroxidu sodného), který byl ještě zředěn na koncentraci přibližně (0,8 až 1,0)%. V této koncentraci sice trvá odplavení vrstvy laku o něco málo déle, ale nehrozí poškození desky. Tedy spíše odplavení veškerého laku bez ohledu na to, zda byl osvětlen.



Obr.č.13 Proces vyvolání osvětlené DPS

5.4 Leptání

Po odplavení laku, tedy vyvolání, přišlo na řadu leptání. Je to proces, při kterém se vytvářejí vodivé cesty. Vyvolaná deska se vloží do chloridu železitého, který z ní odstraní nepotřebnou měď. Tento samotný proces zabírá z celé části výroby DPS

největší časovou náročnost. Při použití nového chloridu železitého, který se před použitím ohřeje zhruba na teplotu 40-60°C se jedna vrstva odleptá přibližně do třiceti minut.

Je-li menší teplota, popřípadě vícekrát použitý chlorid železitý může se tato procedura pohybovat v řádech hodin a výsledek je velmi nekvalitní (např. se na DPS objevují škrábance nebo se jinak znehodnocuje.)



Obr.č.14 Vyleptání elektrických cest na DPS

5.5 Dokončovací operace

Po odleptání mědi se pomocí acetonu odstraní různé nečistoty, které se mohou objevovat na desce. Do takto připravené desky je možno vrtat případné otvory pro elektronické prvky nebo prokovy.

Z vyvrtané desky se pomocí co nejjemnějšího smirkového papíru odstraní výstupky materiálu vzniklé vrtáním děr. Po osmirkování se povrch DPS zase očistil acetonem, nechal chvíli uschnout.

Na takto připravenou desku se nanese vrstva nového laku, který slouží k ochraně proti vnějším vlivům, tedy korozi a mechanickému poškození. Nalakovaná deska se nechala schnout 30 minut. Obrátila a nalakovala se i z druhé strany a nechala se pořádně uschnout – kolem 24 hodin. Po této době už bylo možno bez nějakých problémů začít s pájením. Při pájení sloužila vrstva laku jako pomůcka pro přichytávání SMD prvků, protože po zaschnutí laku se jeví ještě jako mírně lepkavá.

6 Programování modelu

6.1 AVR studio

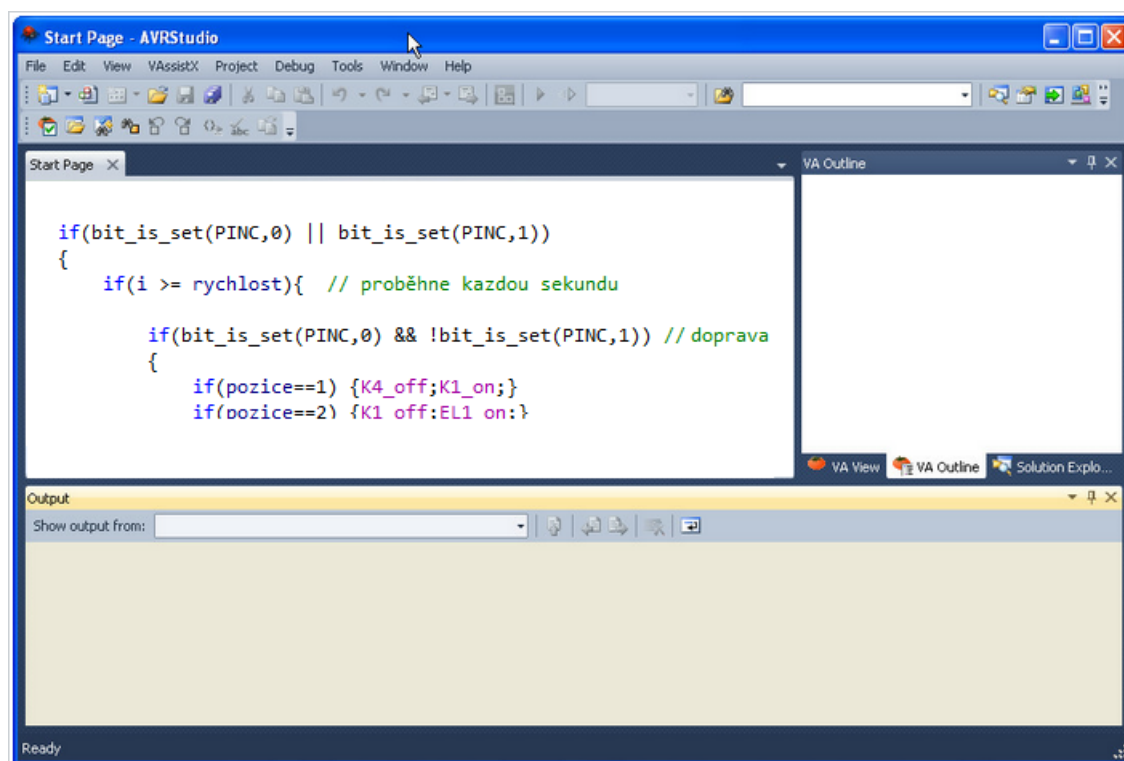
AVR studio od firmy ATMEL je jednotné vývojové prostředí pro tvorbu a ladění aplikací. AVR studio nabízí bezproblémové snadno ovladatelné prostředí k zápisu a ladění vašeho C/C++ popřípadě kódu assembleru.

Jeho největší výhodou, je ovšem to, že je volně dostupné. AVR studio podporuje všechny procesory AVR od 8 – 32-bit AVR.

Samotné programovací prostředí se skládá z textového editoru, okna output a okna outline. Textový editor slouží pro psaní programu. Program je zde automaticky, barevně rozlišen. S možností barevného rozlišení tedy přichází větší přehlednost programu a snazší orientace v delším a složitějším kódu.

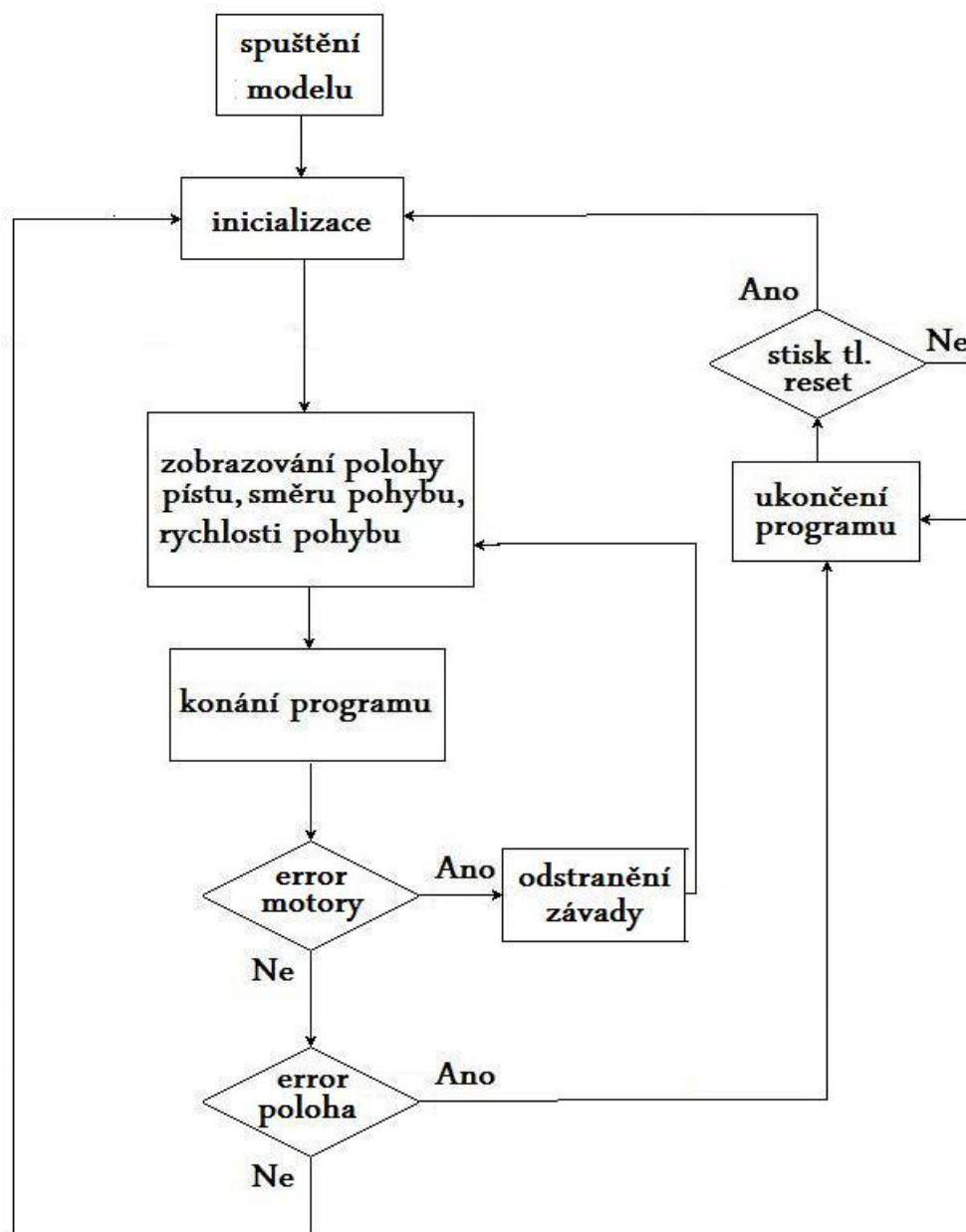
Základní barvou je černá. Touto barvou jsou vyznačovány veškeré hodnoty proměnných a psaného textu, modrou jsou zvýrazněny všechny psané podmínky a v neposlední řadě zelenou jsou označovány popisky k řádkům.

Po dopsání programu máme možnost jej zkontrolovat, zda v něm nejsou chyby, jež by způsobily jeho nefunkčnost. Pomocí funkce build otestujeme, zda je náš program bezchybný a následně je vytvořen stejnojmenný soubor s koncovkou *.hex. tento se následně nahraje do procesoru.



Obr.č.15 Okno programu AVR5.0

6.1.1 Popis vývojového diagramu



Obr.č.16 vývojový diagram řídicího programu

Při samotném spuštění modelu proběhne stav inicializace. Model automaticky rozsvítí koncovou polohu K1, kde budu spočívat do zahájení řídicí činnosti.

Pokud bude spuštěn jeden z motorů EM1 nebo EM2 budou se postupně rozsvěcovat LED diody ve směru pohybu vpravo nebo vlevo, nedojde-li k přjetí jedné z koncových poloh K1 nebo K4.

Budou-li spuštěny oba motory současně, nastane chybový stav. Ovšem pokud bude možno jedem motor odpojit, program bude nadále pokračovat bez nutnosti restartování modelu.

Pokud nastane druhý chybový stav, a to přejetí koncové polohy pístu, celý program se zastaví. Dojde k ukončení programu a rozsvítí se všechny LED diody signalizující polohu pístu a err. Po stisku tlačítka se model znovu nainicializuje.

6.1.2 Popis programu pro ATmega 16

V první řadě bylo nutno zahrnout veškeré knihovny, které budou použity v našem programu toto se dělo pomocí příkazů include.

```
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <string.h>
#include <stdio.h>
#include <stdint.h>
```

Obr.č.17 ukázka příkazu include

Následně jsme definovali veškeré porty, které budeme používat v našem programu. Tedy přiřadit jim označení a definovat kdy budou v logické jedničce popřípadě nule. Toto se dělo pomocí příkazu define.

```
#define ERR_on PORTA &= ~(1 << PA7)
#define ERR_off PORTA |= (1 << PA7)

#define EL1_on PORTB &= ~(1 << PB0)
#define EL1_off PORTB |= (1 << PB0)

#define K1_on PORTD &= ~(1 << PD0)
#define K1_off PORTD |= (1 << PD0)
```

Obr.č.18 ukázka definování jednotlivých portů

Až jsme měli nadefinované veškeré porty, mohli jsme začít se samotným programem. Určit si podmínky, při kterých se co bude dít. Tyto podmínky se dějí pomocí příkazů IF. V mém případě jsem si musel říci, že pokud bude sepnutý pinc0 tak se budou postupně rozsvěcovat a zhasínat signalizační LED diody směrem doprava. Stále se bude přičítat a zvyšovat hodnota aktuální pozice. Pokud se ovšem v určitý okamžik neodstraní logická jednička z tohoto portu, hodnota se dostane až do pozice 11 a vyskytne se chyba. Tedy modul bude signalizovat přejetí a bude muset být restartován. Stejným principem je řešen i pohyb doleva. Ovšem s tím rozdílem, že se zde odčítá od maximální dosažené pozice. (Tedy najedu-li do pozice např. 5 zpětný pohyb bude odečítat z hodnoty pět, až do hodnoty 1 kdy se zase vyskytne chyba).

```

if(bit_is_set(PINC,0) || bit_is_set(PINC,1))
{
    if(i >= rychlost){ // proběhne každou sekundu

        if(bit_is_set(PINC,0) && !bit_is_set(PINC,1)) // doprava
        {
            if(pozice==1) {K4_off;K1_on;}
            if(pozice==2) {K1_off;EL1_on;}
            if(pozice==3) {EL1_off;EL2_on;}
            if(pozice==4) {EL2_off;K2_on;}
            if(pozice==5) {K2_off;EL3_on;}
            if(pozice==6) {EL3_off;EL4_on;}
            if(pozice==7) {EL4_off;K3_on;}
            if(pozice==8) {K3_off;EL5_on;}
            if(pozice==9) {EL5_off;EL6_on;}
            if(pozice==10) {EL6_off;K4_on;}
            if(pozice==11) {ERR_on;err=1;}
            pozice++;
        }
    }
}

```

Obr.č.19 ukázka definování posuvu doprava

Rychlost posuvu se řeší pomocí definování dalších podmínek. V tomto případě pokud máme sepnutý pinc2 a současně pinc0 nebo pinc1 a pinc0 záleží kterým směrem chceme aby se nám signál šířil. Jedeme rychlostí pracovního posuvu. Pokud je zapnut pouze pinc1 nebo pinc2 jedeme rychlostí rychloposuvu.

```

if(bit_is_set(PINC,2) && (bit_is_set(PINC,0) || bit_is_set(PINC,1)))
{
    rychlost = 6000;
}
else
{
    rychlost = 3000;
}

```

Obr.č.20 řešení změny rychlosti posuvu

V hlavní části programu řešíme pouze spuštění obou motorů současně. Tedy je-li přivedena logická jednička na oba piny současně, tedy pinc1 a pinc2. V tomto případě se bude střídavě zapínat a vypínat portA7, LED dioda určena pro signalizaci chyby, dokud nejméně jeden ze signálů nezmizí.

```

if(bit_is_set(PINC,0) && bit_is_set(PINC,1))
{
    ERR_on;
    _delay_ms(200);
    ERR_off;
    _delay_ms(200);
}

```

Obr.č.21 řízení blikání LED err při spuštění obou posuvů současně

6.2 AVR ISP programmer

Tento freewarový program umožňuje programování, čtení a verifikaci FLASH a EEPROM paměti mikrokontrolerů řady AVR firmy Atmel soubory Intel-HEX, které jsou standardně generovány např. vývojovým prostředím [Atmel AVR Studio](#). Umožňuje také programování LOCK bitů a FUSE bitů. Z toho důvodu je tedy mít dobře nastavenou kolonku Fuse bitů, aby nevznikaly komplikace spojené se správností programu.

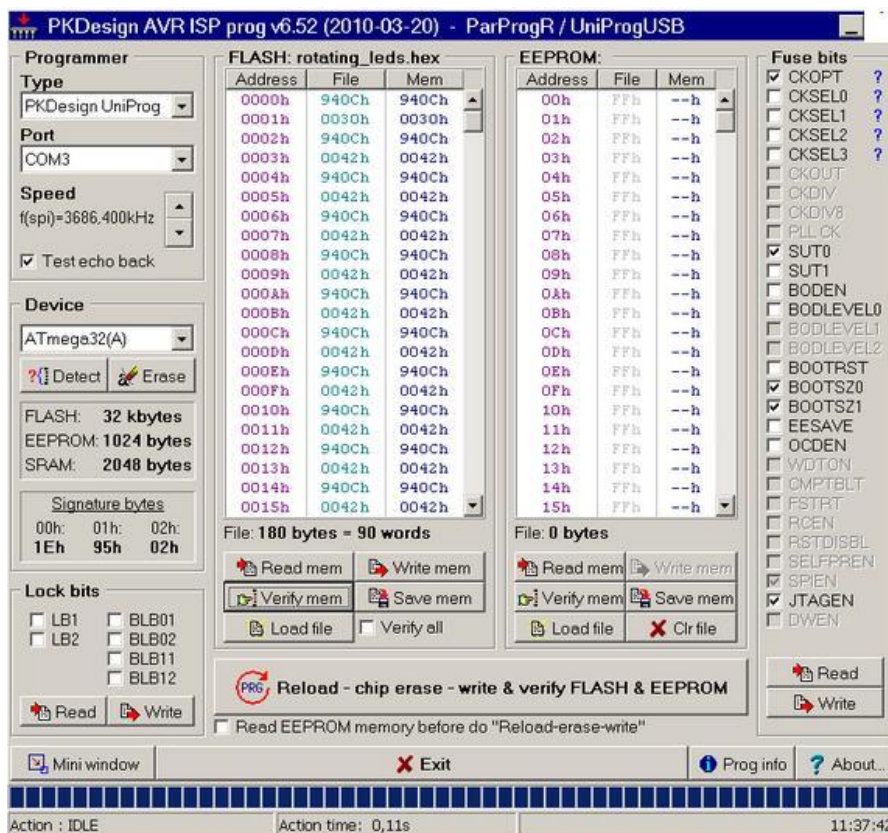
Fuse bits	
☐	CKOPT ?
☒	CKSEL0 ?
☐	CKSEL1 ?
☐	CKSEL2 ?
☐	CKSEL3 ?
☐	CKOUT
☐	CKDIV
☐	CKDIV8
☐	PLL CK
☐	SUT0
☒	SUT1
☐	BODEN
☐	BODLEVEL0
☐	BODLEVEL1
☐	BODLEVEL2
☐	BOTRST
☒	BOOTSZ0
☒	BOOTSZ1
☐	EESAVE
☐	WDTON
☐	CMPTBLT
☐	OCDEN
☐	JTAGEN
☒	SPIEN
☐	SELFPRN
☐	DWEN
☐	HWBE
☐	RSTDISBL
☐	FSTRT
☐	RCEN

Obr.č.22 Nastavení Fuse bitů

Programovací software využívá ke své funkci [UniProg-USB - univerzální programovací kabel](#) a také [Xilinx FPGA, CPLD & Atmel AVR programovací kabel na paralelní port](#) (kompatibilní s Xilinx parallel cable III). Programování se provádí přes ISP rozhraní mikrokontroleru (low voltage serial downloading) tj. pomocí vývodů SCK, MISO, MOSI a RESET\.. Podporovány jsou obvody řady AT90Sxxxx, ATtiny a ATmegaXXXX. Programátor pracuje pod operačními systémy Windows 9x/NT/2000/XP/Vista/7.

Program po spuštění automaticky detekuje připojený programovací kabel i typ připojeného AVR mikrokontroleru. Pro zjištění MCU si program nastaví potřebné parametry (velikost FLASH a EEPROM, typy FUSE bitů a LOCK bitů) a zobrazí základní informace o tomto obvodu (velikosti pamětí a přečtené identifikační byty – signature bytes). Průběh každé operace je přehledně indikován progress barem. Zobrazuje se také aktuální čas, čas od začátku operace a typ operace.

Po dokončení libovolné operace čtení či zápisu dat z/do mikrokontroleru se provede jeho reset a programovací kabel se uvede do stavu, kdy veškeré výstupy jsou ve stavu vysoké impedance. Není tedy nutné programovací kabel od obvodu odpojovat, což je pro ladění aplikace důležité. [13]



Obr.č.23 AVR ISP programmer

6.2.1 UniProg-USB

Tento programátor má velice univerzální použití. Umožňuje vysokorychlostní programování AVR přes ISP rozhraní. Programování a ladění AVR přes JTAG rozhraní z AVR studia. Připojuje se jednoduše přes USB port. Je plnohodnotně galvanicky oddělen. Napájení má 3,3V -5,0V(z připojené aplikace.) [13]

Další specifikace:

- Programovací konektor typu MLW10 - kompatibilní s konektory základových desek systému MVS PKDesign.
- Automatické spuštění ISP/JTAG firmware na základě detekce typu připojeného kabelu (s/bez redukce v konektoru MLW10).
- Velmi malé rozměry – obal je typu „redukce CAN9-CAN9“.
- Součástí balení je USB kabel k PC a propojovací kabel PFL10-PFL10.



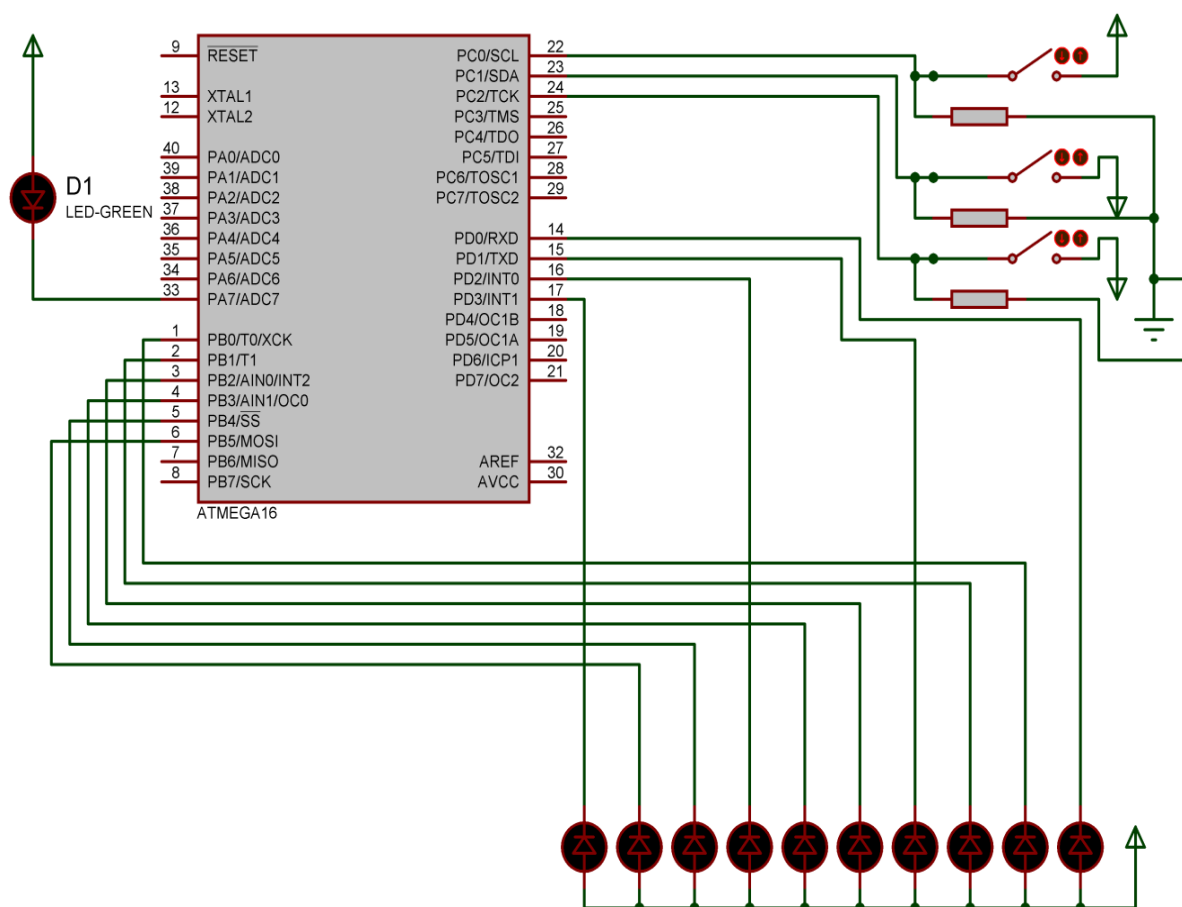
Obr.č.24 UniProg-USB

6.3 Program Proteus

Po vytvoření programu v AVR studiu bylo nutno jej otestovat zda-li vůbec funguje. Tedy buď jej přímo nahrát do našeho mikrokontroléru nebo jednodušší cestou ověřit jeho funkčnost v příhodném simulačním systému.

K tomuto účelu nám dobře posloužil program Proteus 7 Professional. Je to program pro simulaci elektronických obvodů.

Vytvořili jsme si zde tedy systém obsahující stejné parametry. Tedy vybrali mikrokontrolér ATmega16 a na příslušné piny jsme zapojili vstupní a výstupní signalizační prvky (LED diody a tlačítka) abychom si mohli vyzkoušet funkčnost.



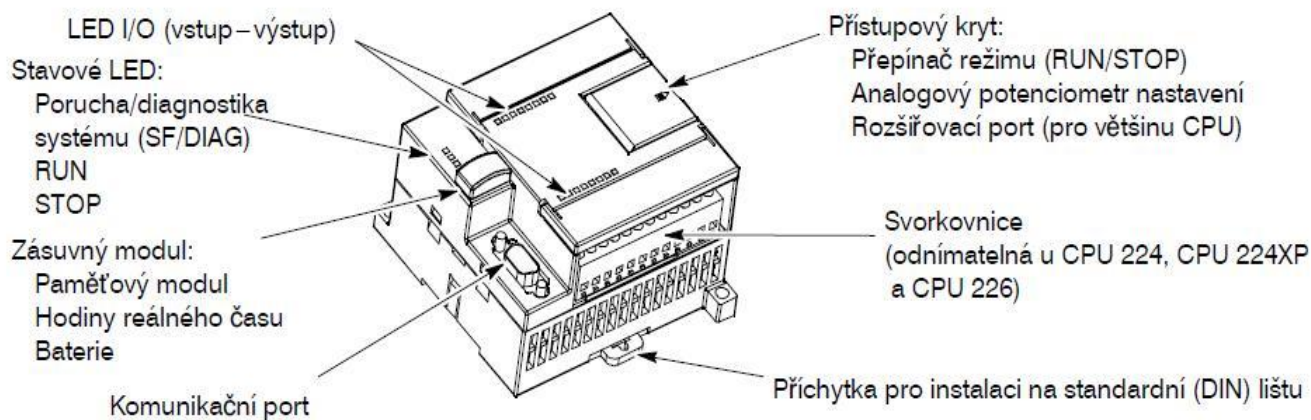
Obr.č.25 Schéma ATmega 16 v prostředí Proteus 7 professional

6.4 Programovatelné automaty SIEMENS

SIMATIC S7-200 je řada malých programovatelných automatů (micro PLC) určených k řízení v různých automatizačních aplikacích.

Kompaktní design, nízká cena a výkonné instrukce se v řadě s7-200 spojují tak, aby byl celý systém nejen jednoduchý, ale i výkonný. [3]

Zařízení S7-200 monitoruje vstupy a řídí výstupy pomocí uživatelského programu, který může obsahovat Booleovy logiku, počítání, časování, složité matematické operace a komunikaci s jinými inteligentními zařízeními. Kompaktní design, flexibilní konfigurace a výkonný instrukční soubor jsou důvody, proč je zařízení S7-200 optimálním řešením pro řízení široké škály aplikací. [2]



Obr.č.26 Programovatelný automat SIEMENS

V našem případě se jedná o Siemens simatic S7 CPU 224 XP, tato jednotka má následující parametry:

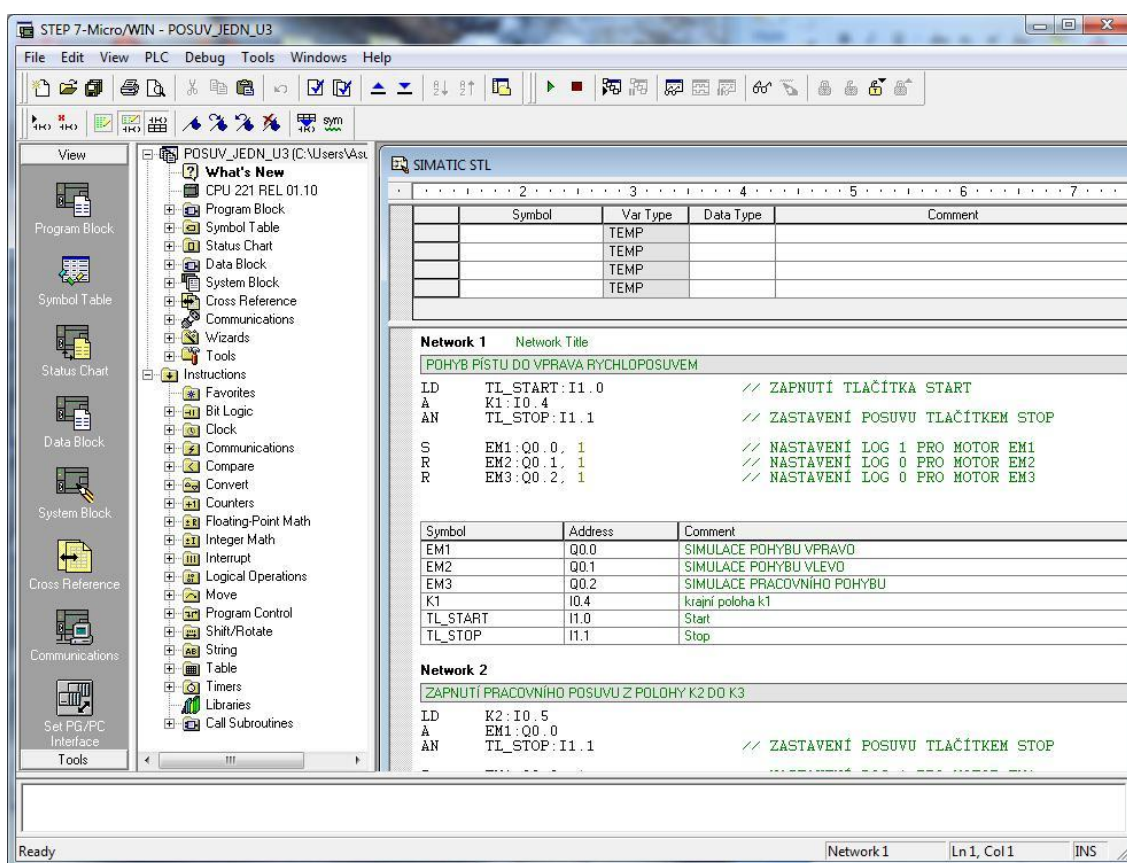
- Paměť pro program
 - S editací v režimu RUN 12286 bytů
 - Bez editace v režimu RUN 16384 bytů
- Paměť pro data 10240 bytů
- Počet vstupů
 - Digitální 14 vstupů
 - Analogové 2 vstupy
- Počet výstupů
 - Digitální 10 výstupů
 - Analogové 1 výstup
- Rozšiřovací moduly 7 modulů
- Vysokorychlostní čítače
 - Jednofázové 4 při 30kHz
2 při 200 kHz
 - dvoufázové 3 při 20 kHz
1 při 100 kHz
- Komunikační porty 2xRS-485
- Rychlost booleovských instrukcí

6.5 Programové prostředí Micro/Win

Programový balík STEP 7 Micro/Win je prostředí pro vytváření, editaci a monitorování logiky nutné k řízení aplikace. Obsahuje tři programové editory pro komfort a efektivitu při vytváření řídicího programu pro aplikaci.

V situaci, kdy si nejste jisti a potřebujete poradit je vývojové prostředí vybaveno rozsáhlým systémem online nápovědy a CD dokumentací, která obsahuje elektronickou verzi tohoto manuálu, typy pro aplikace a další užitečné a další užitečné informace.

Okno projektu nabízí příjemné pracovní prostředí pro vytváření řídicího programu. Nástrojové lišty obsahují tlačítka pro zkratky k často používaným příkazům menu. Strom s instrukcemi zobrazuje všechny objekty projektu a instrukce pro tvorbu řídicího programu. Jednotlivé instrukce můžete do programu vložit pomocí metody drag & drop. Tato metoda umožňuje rychlejší práci v prostředí. [3]



Obr.č.27 Programové prostředí Step7 Micro/Win

6.5.1 Vlastnosti editoru STL

Editor STL zobrazuje program jako znakově orientovaný jazyk. Umožňuje vytvářet řídicí programy vkládáním textových instrukcí. Taktéž umožňuje tvorbu programů, které by pomocí jazyků LAD popřípadě FBD nešly vytvořit. To je kvůli tomu, že jej neprogramujete graficky, tedy nepodléháte omezení grafických editorů.

Tato znakově orientovaná koncepce je velmi podobná programování ve strojovém kódu. S7 – 200 provádí každou instrukci v pořadí určeném od shora dolů.

Potom se opakuje cyklicky. Pro řešení řídicí logiky používá jazyk STL logický zásobník.

```

Network 1   Network Title
POHYB PÍSTU DO VPRAVA RYCHLOPOSUVEM

LD    TL_START:I1.0          // ZAPNUTÍ TLAČÍTKA START
A     K1:I0.4

S     EM1:Q0.0, 1            // NASTAVENÍ LOG 1 PRO MOTOR EM1
R     EM2:Q0.1, 1            // NASTAVENÍ LOG 0 PRO MOTOR EM2
R     EM3:Q0.2, 1            // NASTAVENÍ LOG 0 PRO MOTOR EM3

```

Obr.č.28 Ukázka programovacího jazyka STL

- Vhodnější pro zkušené programátory
- Umožňuje řešit problémy, s nimiž si nedovedou poradit jazyky FBD nebo LAD
- Lze používat pouze s instrukčním souborem SIMATIC
- Lze jej vždy použít na zobrazování nebo editaci programu vytvořeného v jazyku FBD nebo LAD, ale naopak to není možné v některých případech.

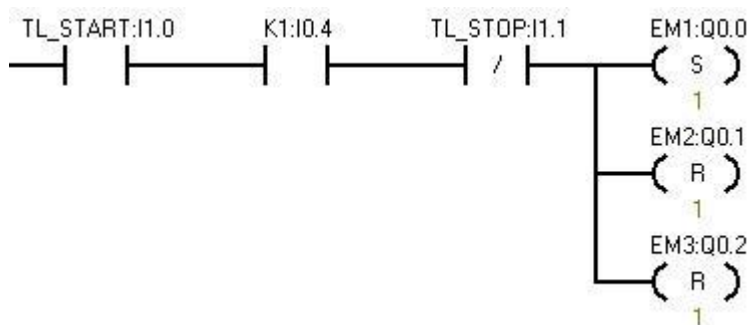
6.5.2 Vlastnosti editoru LAD

Editor LAD zobrazuje program grafickou formou podobnou elektrickým schémátům. Programy v kontaktním schématu umožňují simulovat tok elektrického proudu z elektrického zdroje přes řadu logických vstupních podmínek, které následně aktivují výstupní logické podmínky.

Program LAD obsahuje levou lištu, která je pod napětím. Jsou-li kontakty sepnuté, umožňují to energie k dalším prvkům. Oproti tomu rozpojené kontakty tok energie blokují.

Logika je rozdělena do několika spojitých sítí, NETWORKŮ. Řídicí jednotka vykonává program vždy v jednom network z levé lišty směrem vpravo. Potom teprve přistoupí k následujícímu networku. Instrukce jsou představovány grafickými symboly. Mají dvě základní formy.

Kontakty symbolizují logické vstupní podmínky, jako jsou spínače, tlačítka nebo vnitřní podmínky. Cívky obvykle představují logické výstupní obvody, jako jsou žárovky startéry motorů nebo přechodová relé (vnitřní výstupní podmínky). Jestliže se volí editor LAD je nutno brát v úvahu některé okolnosti.



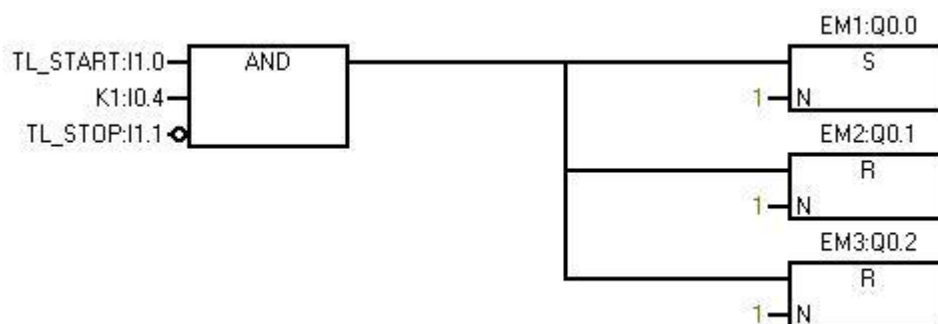
Obr.č.29 Ukázka programovacího jazyka LAD

- Kontaktní schémata snadno mohou používat programátoři začátečníci
- Grafické zobrazení se snadno chápe a je populární na celém světě
- Editor lad je možné použít s instrukčním souborem SIMATIC i IEC 1131-3
- Vždy můžete použít editor STL pro zobrazení programu vytvořeného v editorem SIMATIC LAD

6.5.3 Vlastnosti editoru FBD

Editor FBD zobrazuje program v grafické formě, která připomíná klasická logická schémata. Neobsahuje jak kontakty, tak ani cívky, jež se nalézají v editoru LAD, ale ekvivalentní instrukce. Tyto se objevují jako blokové instrukce. Editor FBD nepoužívá pojem levých a pravých napájecích lišt. Termín „signálový tok“ se tedy využívá pro vyjádření toku řízení přes logické bloky.

Vstup a výstup signálového toku je možno přímo přiřadit operandu. Logika je odvozena od spojení těchto bloků. Tedy výstup jedné instrukce je možno použít pro aktivaci instrukce jiné. Tato koncepce umožňuje vyřešit mnoho různých logických problémů. Jestliže se volí editor FBD, je nutno vzít v potaz některé okolnosti.



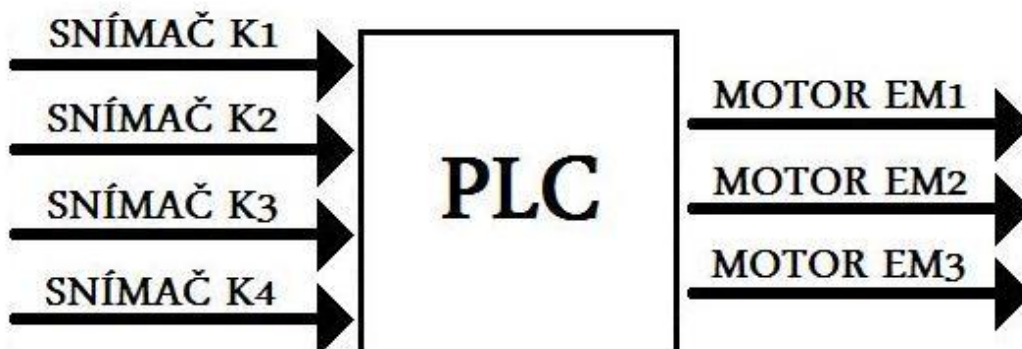
Obr.č.30 Ukázka programovacího jazyka FBD

- Styl zobrazení pomocí grafických členů umožní dobře sledovat tok programu
- Editor FBD je možné použít s instrukčním souborem SIMATIC i IEC 1131-3
- Pro zobrazení programu vytvořeného editorem SIMATIC FBD můžete vždy použít editor STL

6.6 Testovací úlohy

Vypracované úlohy přikládám k práci na CD. Jsou to úlohy testovací, které uživateli umožní postupně se seznámit s modelem posuvové jednotky a vyzkoušet většinu jejích možností. Jsou zde ukázány jak ty nejjednodušší úlohy, kde se pouze vyzkouší rozjetí modulu. Tak i úlohy, kde se testují veškeré funkce posuvové jednotky. Tyto úlohy jsou řešeny pomocí zde již zmíněného prostředí Step 7Micro/Win.

Pro každou úlohu lze využít dané schéma:



Obr.č.31 Schéma vstupů a výstupů PLC

Zde je vzor tabulky symbolů, která byla vytvořena v programu Step7 pro zjednodušení a převážně pro zpřehlednění programu. Využívá zde přiřazení slovních vyjádření na místo označování vstupů a výstupů písmeny I a Q

Symbol Table				
		Symbol	Address	Comment
1		K1	I0.4	krajní poloha k1
2		K2	I0.5	poloha k2
3		K3	I0.6	poloha k3
4		K4	I0.7	krajní poloha k4
5				
6				
7		TL_START	I1.0	Start
8		TL_STOP	I1.1	Stop
9				
10		EM1	Q0.0	SIMULACE POHYBU VPRAVO
11		EM2	Q0.1	SIMULACE POHYBU VLEVO
12		EM3	Q0.2	SIMULACE PRACOVNÍHO POHYBU
13				

Obr.č.32 Vzor tabulky symbolů pro PCL

6.6.1 Úloha 1.

Zde se pouze seznámíme s funkcí rychloposuvu a dosažení krajních poloh modelu.

Zadání: Po zapnutí tlačítka START se spustí simulace motoru EM1. Rychloposuvem se dostaneme do polohy K4 odkud se píst vrátí do polohy K1 taktéž rychloposuvem. Zde počká na opětovné zmáčknutí tlačítka start.

6.6.2 Úloha 2.

Tato úloha má pouze upravené zadání předchozí úlohy. Je zde ukázáno využití čítačů a tedy pozdržení procesu o krátký okamžik.

Zadání: Po zapnutí tlačítka START se spustí simulace motoru EM1. Rychloposuvem se dostaneme do polohy K4. Zde vyčká čtyři vteřiny a potom se vrátí rychloposuvem do polohy K1 kde vyčká do opětovného zmáčknutí tlačítka START.

6.6.3 Úloha 3.

V této úloze si již ověříme funkci pracovního posuvu.

Zadání: Po zapnutí tlačítka START se začne píst posouvat rychloposuvem do polohy K2, odkud se začne pohybovat posuvem pracovním až do polohy K3. Z polohy K3 se přesune rychloposuvem do krajní polohy K4. Po dosažení polohy K4 se vrátí do polohy K1 kde vyčká na opětovné sepnutí tlačítka START. Po sepnutí tlačítka STOP se modul zastaví. Po jeho vypnutí bude pokračovat v pohybu.

Řešení úlohy 3:

Pro názornost a alespoň malé seznámení se s tím jak by měly vypadat programy Pro step 7 Micro/Win jsem se rozhodl ukázat řešení jedno z příkladů. V této úloze, kvůli její jednoduchosti byly použity pouze čtyři networky.

V networku 1 se načte tlačítko START a současně pokud bude píst spočívat v poloze K1 bude nastavena simulace motoru EM1 na logickou jedničku, motory EM2 a EM3 budou resetovány (nastaveny na logickou nulu). Toto způsobí pohyb pístu vpravo rychloposuvem.

Network 1	Network Title
POHYB PÍSTU DO VPRAVA RYCHLOPOSUVEM	
LD	TL_START : I1.0 // ZAPNUTÍ TLAČÍTKA START
A	K1 : I0.4
S	EM1 : Q0.0, 1 // NASTAVENÍ LOG 1 PRO MOTOR EM1
R	EM2 : Q0.1, 1 // NASTAVENÍ LOG 0 PRO MOTOR EM2
R	EM3 : Q0.2, 1 // NASTAVENÍ LOG 0 PRO MOTOR EM3

Obr.č.33 Network 1- ukázka programu PLC

V networku 2 se načte nejprve poloha pístu K2 a pokud je současně spuštěna simulace motoru EM1 nastaví se motor EM1 a EM3 na logickou jedničku a resetuje se motor EM2. Tedy zůstane stále v logické nule.

Network 3**POHYB PÍSTU Z K3 DO K4 RYCHLOPOSUVEM**

```

LD      K3:I0.6
A      EM1:Q0.0

S      EM1:Q0.0, 1          // NASTAVENÍ LOG 1 PRO MOTOR EM1
R      EM2:Q0.1, 1          // NASTAVENÍ LOG 0 PRO MOTOR EM2
R      EM3:Q0.2, 1          // NASTAVENÍ LOG 0 PRO MOTOR EM2

```

Obr.č.34 Network 2- ukázka programu PLC

V networku 3 se načte poloha pístu K3 a pokud je současně zapnutý motor EM1 nastaví se simulace motoru EM1 na logickou jedničku, EM2 a EM3 se vyresetují. Tedy nastaví se na logickou nulu a píst bude pokračovat v pohybu vpravo pomocí rychloposuvu.

Network 2**ZAPNUTÍ PRACOVNÍHO POSUVU Z POLOHY K2 DO K3**

```

LD      K2:I0.5
A      EM1:Q0.0

S      EM1:Q0.0, 1          // NASTAVENÍ LOG 1 PRO MOTOR EM1
S      EM3:Q0.2, 1          // NASTAVENÍ LOG 1 PRO MOTOR EM3
R      EM2:Q0.1, 1          // NASTAVENÍ LOG 0 PRO MOTOR EM2

```

Obr.č.35 Network 3- ukázka programu PLC

V posledním networku se řeší pouze návrat pístu z koncové polohy K4 do výchozí polohy K1. Jako první se zjistí, zda je píst už v poloze K4. Je-li tato podmínka splněna píst se bude rychloposuvem vracet do polohy K1. Tedy nastaví se motor EM2 na logickou jedničku a motory EM1, EM3 budou v logické nule. Po dosažení krajní polohy K1 bude píst opět vyčkávat na sepnutí tlačítka START a program se bude opakovat.

Network 4**POHYB PÍSTU VLEVO RYCHLOPOSUVEM**

```

LD      K4:I0.7
|

S      EM2:Q0.1, 1          // NASTAVENÍ LOG 1 PRO MOTOR EM2
R      EM1:Q0.0, 1          // NASTAVENÍ LOG 0 PRO MOTOR EM1
R      EM3:Q0.2, 1          // NASTAVENÍ LOG 0 PRO MOTOR EM3

```

Obr.č.36 Network 4- ukázka programu PLC

6.6.4 Úloha 4.

V úloze 4 se již prolínají veškeré funkce, jež jsou dosažitelní pro tento modul. Tedy rychloposuv, pracovní posuv a rovněž zpoždění realizované pomocí čítačů.

Zadání: Po zapnutí tlačítka START se začne píst posouvat rychloposuvem do polohy K2, zde vyčká po dobu dvou vteřin. Po uplynutí této doby se začne pohybovat posuvem pracovním až do polohy K3, kde vyčká další dvě vteřiny. Z polohy K3 se přesune rychloposuvem do krajní polohy K4. Po dosažení polohy K4 se vrátí do polohy K1 kde vyčká na opětovné sepnutí tlačítka START. Po sepnutí tlačítka STOP se modul zastaví. Po jeho vypnutí bude pokračovat v pohybu.

Samozřejmě k tomuto modelu může být vytvořen nespočet jiných testovacích úloh a jejich kombinací. Pro ukázání a vyzkoušení všech funkcí modulu bohatě postačí výše zmíněné čtyři úlohy, jež jsou vypracovány a přiloženy na kompaktním disku.

7 Závěr

Naším úkolem bylo seznámit se s modelem posuvné hydraulické jednotky a realizovat jej pomocí mikrokontroléru ATmega. Funkčnost realizovaného modelu měla být ověřena pomocí programovatelného automatu PLC a simulačních úloh.

V první řadě jsem se seznámil s funkcí a vlastnostmi provozovaného modelu a potom jsem vytvořil univerzální základní desku, na kterou je možno připojit jakoukoliv zobrazovací desku. V tomto případě posuvovou jednotku. Návrh desky je k nalezení v kapitole tři a výroba desky je v kapitole pět.

Posléze jsem vytvořil program pro mikrokontrolér ATmega, aby zajišťoval správnou funkci modelu.

Dále jsem vytvořil základní testovací úlohy pro PLC, na kterých byla odzkoušena funkčnost modelu. Tyto úlohy jsou k nalezení v šesté kapitole.

Zadané cíle se nám podařilo splnit a tedy je možno tento model zařadit do výuky pro práci s programovatelnými automaty.

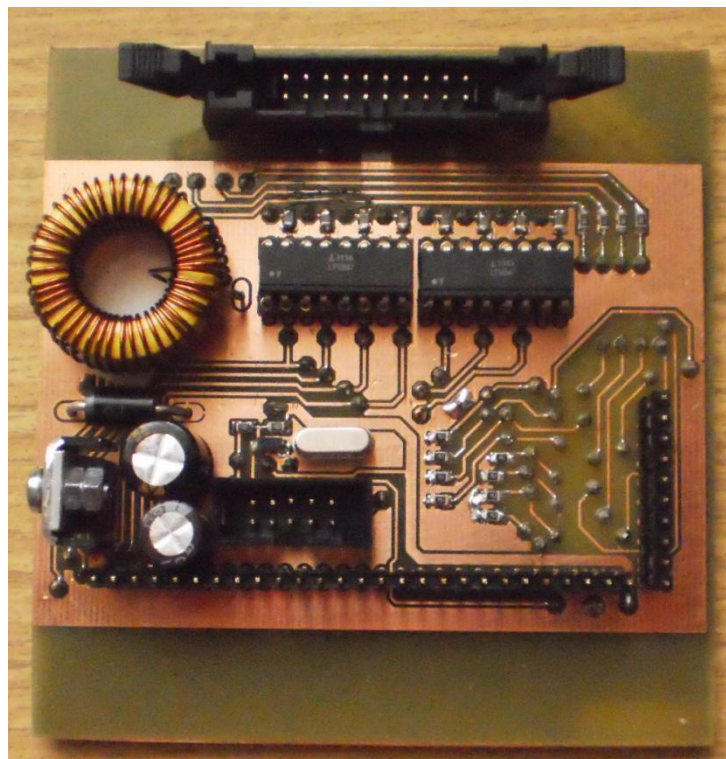
Seznam použité literatury

- [1] Modely procesů EDU-mod ing. Luděk Kohout [cit. 2012-05-23]
- [2] SIEMENS – SIMATIC Programovatelný automat S7-200 systémový manuál [cit. 2012-05-23]
- [3] Řídicí systém static S7-200 www1.siemens.cz [cit. 2012-05-23]
- [4] Dokumentace k mikrokontroléru ATmega16A [cit. 2012-05-23]
- [5] wikipedia free encyklopedia TTL logika [cit. 2012-05-23]
Dostupné z [http://cs.wikipedia.org/wiki/TTL_\(logika\)](http://cs.wikipedia.org/wiki/TTL_(logika))
- [6] <http://Atmel.com>
- [7] <http://avr.hw.cz>
- [8] výroba plošných spojů [cit. 2012-05-23]
Dostupné z <http://www.elweb.cz>
- [9] Práce s programem Eagle, [cit. 2012-05-23]
Dostupné z <http://www.eagle.cz/>
- [10] výukové modely [cit. 2012-05-23] dostupné z <http://www.edumat.cz/>
- [11] DOLEČEK, Jaroslav. Moderní učebnice elektroniky 2. 1. české. [s.l.] : BEN, 2010. 208 s. ISBN 978-80-7300-287-9.
- [12] LM2576 : Datasheet [online]. 2004 [cit. 2012-05-23]. National Semiconductor. Dostupné z WWW: <www.national.com/ds/LM/LM2576.pdf>
- [13] Programátory a konektory [cit. 2012-05-23]
Dostupné z: www.pk-design.net/HtmlCz/ProgCables.html

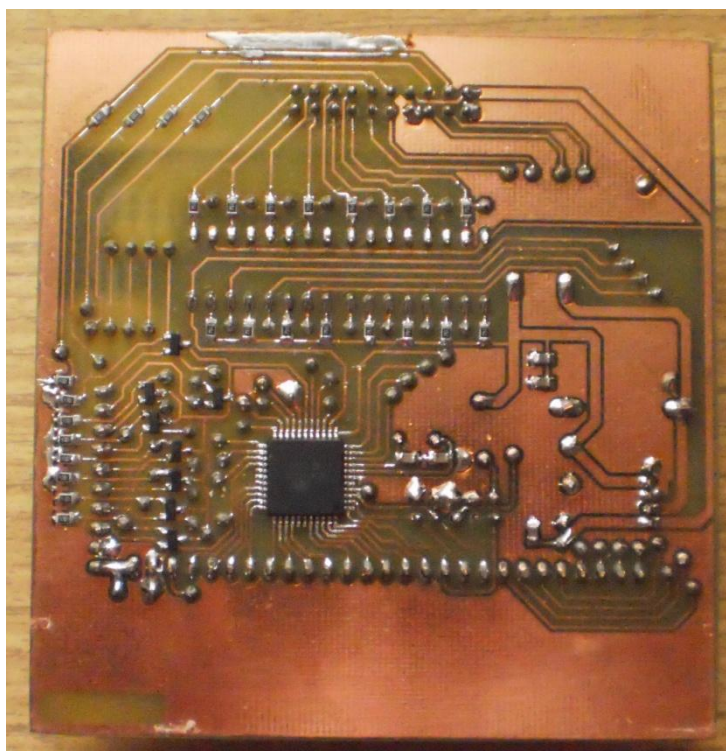
Seznam příloh

- Příloha 1: Základní deska (vrchní strana)
- Příloha 2: Základní deska (spodní strana)
- Příloha 3: Zobrazovací deska
- Příloha 4: Modul posuvové hydraulické jednotky

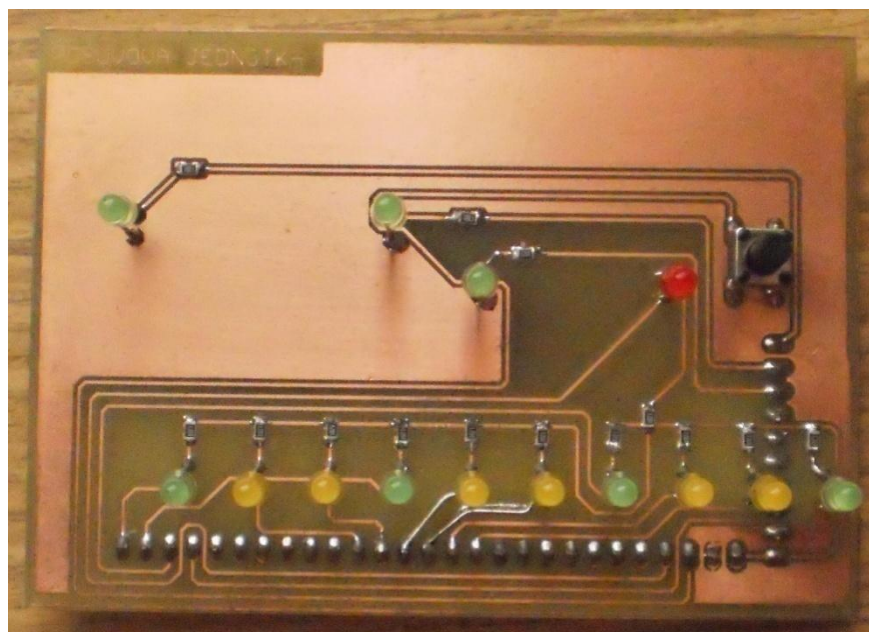
Příloha 1: Základní deska (vrchní strana)



Příloha 2: Základní deska (spodní strana)



Příloha 3: Zobrazovací deska



Příloha 4: Modul posuvové hydraulické jednotky

