

Mikropočítačový modul pro řízení rychlosti vlaků na modelu kolejiště

Microcomputer module for model of train speed control

Stanislav Sehnálek

Bakalářská práce
2010



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Stanislav SEHNÁLEK**

Osobní číslo: **A07091**

Studijní program: **B 3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Téma práce: **Mikropočítačový modul pro řízení rychlosti vlaků na modelu kolejiště**

Zásady pro vypracování:

1. Prostudujte současný stav řízení rychlosti vlaků na modelu kolejiště.
2. Stanovte hardwarové požadavky na řídicí mikropočítač a proveďte výběr vhodného typu.
3. Proveďte hardwarový návrh mikropočítačového modulu s možností přepínání na původní způsob řízení.
4. Realizujte prototyp jednotky a ověřte její funkci.
5. Vytvořte ukázkové programové vybavení pro programovatelný automat SAIA, které bude demonstrovat funkci modulu.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. BURKHARD, Mann. C pro mikrokontroléry. Praha : BEN – technická literatura, 2003. 280 s. ISBN 80-7300-077-6.
2. Freescale Semiconductor. CPU08 Central Processor Unit Reference Manual, 2001. Dostupný z WWW: [http: www.freescale.com](http://www.freescale.com).
3. Freescale Semiconductor. HCS08 Family Reference Manual, Rev.1., 2003. Dostupný z WWW: [http: www.freescale.com](http://www.freescale.com).
4. Freescale Semiconductor. MC9S08QE128 Reference Manual, Rev.2., 2007. Dostupný z WWW: [http: www.freescale.com](http://www.freescale.com).
5. HNILICA, Vladimír. Učební pomůcka pro výuku, vlakové kolejiště . Zlín, 1999. 64 s. Vedoucí bakalářské práce Prof. Ing. Vladimír Vašek, CSc.
6. HNILICA, Vladimír. Počítačové řízení vlakového kolejiště. Zlín, 2002. 68 s. Vedoucí diplomové práce Prof. Ing. Vladimír Vašek, CSc.
7. HRABOVSKÝ, Miroslav, JURÁNEK, Antonín. EAGLE pro začátečníky : Návrhový systém pro plošné spoje. Praha : BEN – technická literatura, 2007. 192 s. ISBN 80-7300-213-2.
8. NOVÁK, Petr. Mobilní roboty – pohony, senzory, řízení. : 1. díl. Praha : BEN – technická literatura, 2005. 256 s. ISBN 80-7300-141-1.

Vedoucí bakalářské práce:

Ing. Petr Dostálek

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

5. března 2010

Termín odevzdání bakalářské práce:

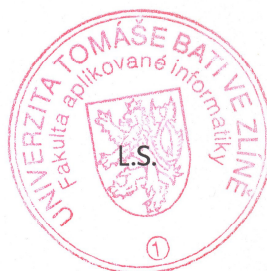
1. června 2010

Ve Zlíně dne 5. března 2010



prof. Ing. Vladimír Vašek, CSc.

děkan



doc. Ing. Ivan Zelinka, Ph.D.

ředitel ústavu

ABSTRAKT

Cílem této bakalářské práce bylo navrhnout a prakticky realizovat mikropočítačový modul pro řízení rychlosti vlaků na modelu kolejiště. Teoretická část popisuje mikropočítač MC9S08QE128 firmy Freescale se zaměřením na jeho základní strukturu a hlavní rozdíly oproti dalším MCU. Dále se zabývá principy PWM modulace a systémy reálného času. Je zde popsán i způsob řízení rychlosti vlaků na kolejišti před úpravou.

V praktické části je popsán návrh částí modulu jako je napájení, LCD, komunikace s PC, propojení s PLC a propojení s kolejištěm. A v neposlední řadě programová část a praktické ověření funkce vytvořeného modulu.

Klíčová slova: PLC, PWM, MCU, Freescale, SCI, BDC, řízení rychlosti

ABSTRACT

The aim of this bachelor thesis was to design and implement microcontroller based control unit for train speed control on rails model. Theoretical part describes microcontroller Freescale MC9S08QE128 with focus on its basic structure and main differences from other microcontrollers. Next it deals with PWM modulation and real-time systems principle. It also describes train speed control method before modification.

The practical part describes the module's part proposal like power, LCD, communication with PC and PLC connection. And finally part is about program implementation and practical verification of created module..

Keywords: PLC, PWM, MCU, Freescale, SCI, BDC, speed control

Rád bych poděkoval mému vedoucímu práce, kterým je Ing. Petr Dostálek, za jeho rady, připomínky a pomoc při řešení této práce.

Když voda naplní džbán, vezme na sebe podobu džbánu. Ale voda přitom není džbán a džbán nemá kromě tvaru nic společného s vodou.

Sir Terry Pratchett, OBE

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému, dostupná k prezenčnímu nahlédnutí; že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití - jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 JEDNOČIPOVÉ MIKROPOČÍTAČE	11
1.1 MIKROPOČÍTAČ MC9S08QE128	11
1.1.1 Centrální procesní jednotka (CPU)	12
1.1.2 Porty	12
1.1.3 Vnitřní registry	13
1.2 BDM.....	16
1.2.1 Komunikace	17
1.2.2 Open Source BDM.....	17
2 PULZNĚ ŠÍŘKOVÁ MODULACE	19
2.1 PRINCIP	19
2.2 DRUHY MODULACE	20
3 REALTIMOVÉ SYSTÉMY	21
3.1 RTMON	23
4 MODEL KOLEJIŠTĚ PRO PODPORU VÝUKY PLC	26
4.1 PLC SAIA PCD2.....	26
4.2 MOŽNÉ ZPŮSOBY PROGRAMOVÁNÍ SAIA PCD2	27
4.3 MODEL KOLEJIŠTĚ.....	29
II PRAKTICKÁ ČÁST	32
5 NÁVRH HARDWARE MODULU	33
5.1 MIKROKONTROLÉR	33
5.2 DISPLEJ	34
5.2.1 LCD modul ATM12864D.....	34
5.2.2 Modifikace zapojení.....	34
5.3 NAPÁJENÍ	35
5.4 PROPOJENÍ MODULU S PLC	36
5.5 PROPOJENÍ MODULU S KOLEJIŠTĚM	37
5.6 KOMUNIKAČNÍ ROZHRANÍ SCI.....	39
5.7 OVLÁDÁNÍ.....	40
5.7.1 PLC mód	41
5.7.2 SCI mód	42
5.7.3 Manuální mód	43
5.7.4 BYPASS mód.....	44
5.7.5 Tabulkový mód	45
5.8 PWM MODULACE.....	45
5.9 DESKY PLOŠNÝCH SPOJŮ	46
5.9.1 Výroba desky plošných spojů	46
6 PROGRAMOVÉ VYBAVENÍ	50

6.1	PŘERUŠOVACÍ RUTINA.....	50
6.2	PROCPLC()	50
6.3	PROCBYP().....	51
6.4	PROCSCI()	51
6.5	PROCDISP()	52
6.6	PROCKEY()	52
6.7	FUNKCE MAIN()	52
6.8	ČTENÍ Z SCI	53
7	PRAKTICKÉ OVĚŘENÍ.....	54
7.1	UKÁZKOVÝ PROGRAM.....	55
	ZÁVĚR	58
	ZÁVĚR V ANGLIČTINĚ.....	59
	SEZNAM POUŽITÉ LITERATURY.....	60
	SEZNAM TABULEK.....	64
	SEZNAM PŘÍLOH.....	65

ÚVOD

V dnešní době se stále častěji setkáváme s problémem řízení výkonu, rychlosti a intenzity osvětlení. V důsledku toho se hledají nová řešení a možnosti, jak toho docílit co nejefektivněji. Jedním z možných řešení je pulzní šířková modulace. Jedná se o princip, kdy se v rychlém cyklu střídá stav zapnuto se stavem zcela vypnuto. Tímto způsobem jsme schopni řídit výkon mnoha zařízení.

Ve výuce programovatelných automatů na Fakultě Aplikované Informatiky se pro výuku používá model vlakového kolejiště. Model s kolejemi zkonstruoval Ing. Vladimír Hnilica jako svou bakalářskou práci. Později se k tomuto tématu vrátil a jako svou diplomovou práci realizoval převodníkovou skříň, aby bylo možné propojit kolejiště s PC vybaveným technologickou kartou. Později se přidalo propojení s PLC SAIA PG2. Praktickou zkušeností z výuky se zjistilo, že PLC není schopno plně řídit modulaci a tím efektivně ovládat rychlost vlaků, aby jezdily plynule. Proto vznikla potřeba dát studentům nástroj, který by se o modulaci postaral. Zároveň, aby byla zachována kompatibilita s kolejištěm a úpravy nebyly moc velké, čímž by vznikaly další problémy k řešení. Jako nejideálnější řešení se jeví externí modul řízený mikrokontrolérem. Takto se bude moci zachovat pinová kompatibilita s PLC a budou se moci rozšířit možnosti ovládání i na PC nebo ruční ovládání.

Tato bakalářská práce řeší hardwarový návrh desky plošných spojů a následné programové vybavení modulátoru. Úvodní kapitola se zabývá teoretickou částí, kde je popsán výběr mikrokontroléru a jeho následný popis. Pak následuje popis jeho centrální procesní jednotky, portů, vnitřních registrů a komunikačního rozhraní BDM. Dále zde jsou vysvětleny základní principy a přednosti PWM modulace. Popsány jsou principy reálných operačních systémů a procesy RTMON. V poslední kapitole teoretické části bude uveden popis modelu vlakového kolejiště před provedením úprav a zapojením PWM modulátoru.

Druhá část, praktická, se zabývá konstrukcí modulátoru a jeho programovým vybavením. Tato část obsahuje dvě hlavní kapitoly - v první z nich se řeší hardwarové aspekty konstrukce. Mezi ně patří realizace zapojení mikrokontroleru a jeho periferních obvodů. Následuje zapojení LCD displeje a jeho propojení s mikropočítačem, řešení napájení a propojení s PLC, respektive s převodníkovou skříní kolejiště. Poslední hlavní kapitola řeší popis obslužného programu, který je zapotřebí k funkci celého modulu.

I. TEORETICKÁ ČÁST

1 JEDNOČIPOVÉ MIKROPOČÍTAČE

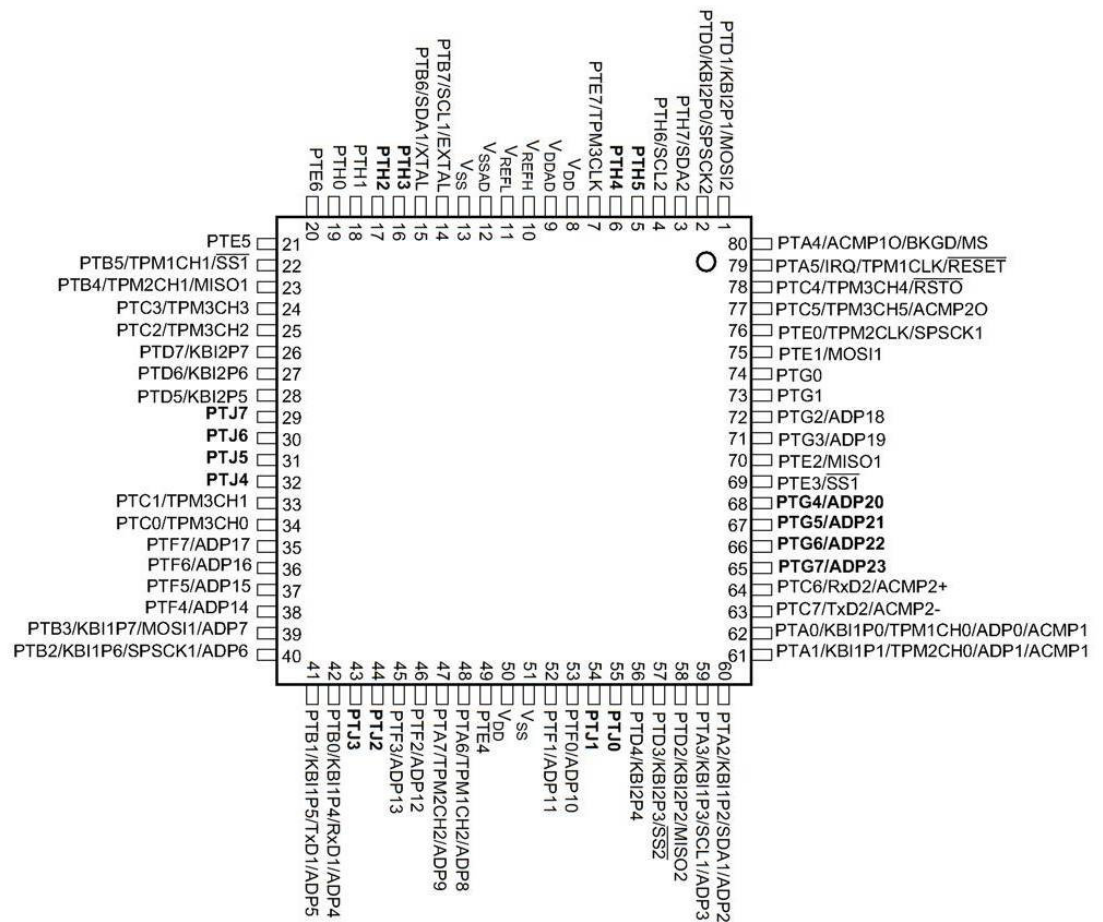
Mikrokontroléry vznikly jako samostatný směr vývoje mikroprocesorů, kdy byly integrovány na jednu desku křemíku všechny části mikropočítače. MCU tedy může obsahovat centrální procesní jednotku, paměť ROM i RAM, digitální vstupy, digitální výstupy, analogové vstupy, analogové výstupy a sériové komunikační rozhraní.

Mikropočítače si našly uplatnění v mnoha sférách lidské činnosti. Lze je najít v embedded zařízeních všude tam, kde je potřeba řídit a kontrolovat. Například výrobní linky, medicínská zařízení, dálkově ovládané aplikace, automobilový průmysl, dětské hračky, automatizace budov, armádní aplikace a kancelářská zařízení.

Na trhu lze nalézt velké množství firem vyrábějících MCU. Proto byl výběr toho nejvhodnějšího mikrokontroléru pro modulátor velice komplikovaný. Mezi požadavky na MCU patří schopnost připojit 10 vstupů od PLC, 10 výstupů na převodníkovou skříň, možnost sériové komunikace, 8 pinů pro LCD display a 4 piny pro tlačítka. Na tyto požadavky postačuje MCU firmy Freescale, konkrétně MC9S08QE128.

1.1 Mikropočítač MC9S08QE128

Tento mikrokontrolér je 8b a patří do rodiny Flexis HCS08. Z toho vyplývá, že má svou 32b variantu, která je pinově, periferně a nástrojově kompatibilní. Díky tomu vývojářům umožňuje migrovat mezi nízkou třídou a vysoce výkonnými integrovanými systémy s mimořádnou lehkostí, rychlostí, efektivností nákladů a provozní efektivností. MC9S08QE128 je založen na jádru HCS08 a jeho 32b varianta MCF51QE128 na ColdFire V1. S08 je postaven na architektuře von Neuman a má instrukční sadu typu CISC. V tomto 80 pinovém mikrokontroleru nalezneme 128KB flash paměti, 9KB RAM paměti, 24 kanálový, analogově digitální převodník, 16b přerušování od klávesnicového rozhraní, 70 I/O portů, 2 sériové komunikační rozhraní, 2 sériové periferní rozhraní a 3 TPM kanály. Význam jednotlivých pinů je vidět na obrázku (Obr. 1). [4]



Obr. 1 Význam pinů u MC9S08QE128CLK

1.1.1 Centrální procesní jednotka (CPU)

CPU má k dispozici 64 KB paměti a speciální jednotku pro případné rozšíření. Taktovací frekvenci lze nastavit od 2 do 50,33 MHz při minimálním napájení 2,4V. Jako zdroj taktovacího signálu může sloužit externí oscilátor v rozmezí 31,25 – 38,4 kHz nebo 1 – 16 MHz. Při použití vnitřního oscilátoru může dojít až k 2% odchylce. Systém Watchdog (COP), který provádí reset systému při jeho zaseknutí, běží na 1 kHz, nebo využívá frekvenci sběrnice. Zaseknutí může nastat v případě chyby hardware nebo software systému. [2]

1.1.2 Porty

MCU disponuje celkem 9 porty po 8 pinech, celkem tedy 72 univerzálních pinů. Navíc také disponuje jedním pinem pouze výstupním a jedním pinem pouze na vstup. Některé tyto vstupně/výstupní piny mají i jiné funkce. Na pinech portu PTD a PTB je umístěno

přerušeni KBI. Piny na portu PTC mohou sloužit k obsluze SCI2. Porty PTB a PTC disponují externím přerušením TPM3.

Pokud není pin sdílen s některou periferií, přejde opět do univerzálního, vstupně/výstupního režimu. Každý port má řídicí registry, které nastavují, jestli bude port vstupní nebo výstupní a jestli se mají aktivovat pull-up rezistory. Po restartu jsou všechny piny nastaveny na vysokou impedanci jako vstupy bez vnitřních pull-up rezistorů.

Jednotlivé porty se značí písmeny A, B, C, D, E, F, G, H, J a bity se značí čísla 0-7, takže přesné určení pinu se zapíše například takto: PTA0.

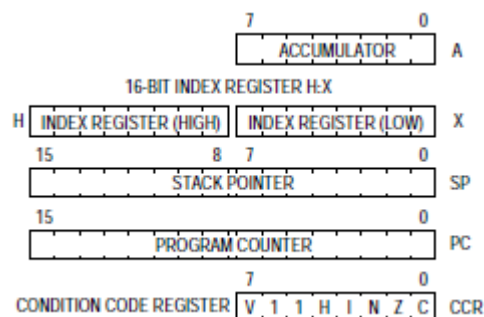
BKGD

Tento pin slouží k ladění na pozadí a je vyveden do BDM, viz kapitola 1.2. Primární funkcí tohoto pinu je obousměrná sériová komunikace aktivních příkazů a dat. Během resetu se tento pin používá k výběru mezi výchozím aktivním režimem na pozadí, nebo startem uživatelského aplikačního programu. Používá se také ke stanovení správné frekvence pro ladění sériové komunikace.

BDC sériová komunikace používá vlastního sériového protokolu, poprvé představeného na M68HC12. Tento protokol předpokládá, že hostitel zná komunikační taktovací frekvenci. Veškerá komunikace je iniciována a řízena hostitelem, který řídí hrany signálu na začátku každého přenosu. Snaží-li se hostitel komunikovat s cílovou MCU, která má neznámou BDC taktovací frekvenci, použije se příkaz SYNC. Tímto požádá o časovou synchronizační odezvu signálu, z níž může hostitel určit správnou rychlost komunikace.

1.1.3 Vnitřní registry

CPU disponuje pěti registry, které nejsou součástí paměti [3]. Uspořádání těchto registrů je na obrázku (Obr. 2).



Obr. 2 Vnitřní registry HCS08

Akumulátor (A - Accumulator)

Akumulátor je 8b univerzální registr. Jeden operand se nahraje do ALU, provede se výpočet a výsledek operace se uloží opět do akumulátoru. Akumulátor může být naplněn nebo uložen z/do paměti za použití různých způsobů adresování. Reset nemá vliv na obsah akumulátoru.

Index Registr (H:X)

Tento registr je složen ze dvou 8 bitových registrů H a X, které spolu pracují jako jeden 16 bitový adresní ukazatel, kde H je vyšší bajt adresy a X nižší. Tento index registr používá plnou 16 bitovou adresaci. Avšak kvůli zpětné kompatibilitě se starším procesorem HC05 používají některé instrukce pouze dolní byte X.

Spousta instrukcí používá X jako univerzální registr vedle akumulátoru, může být použit k uchování dat. S registrem X se mohou provádět i další operace jako nulování, inkrementace, dekrementace, rotace, porovnání, negace nebo posun. Instrukce pro přesun dat umožňují provádět přesun do A nebo z A, kde se provedou aritmetické nebo logické operace.

Z důvodu kompatibility s HC05 je během restartu provedeno nastavení registru H na \$00. Registr X však reset nijak neovlivní.

Ukazatel zásobníku (SP – Stack Pointer)

Tento 16 bitový adresový registr ukazuje na další dostupnou pozici v zásobníku. Zásobník je paměť typu LIFO (Last in –First out). Zásobník může být kdekoliv v celém adresovém prostoru, kde se nachází paměť RAM. Jeho velikost může být nastavena na maximální velikost RAM. Zásobník je používán k uložení návratové adresy při volání podprogramu a také uložení stavových registrů při přerušení. Může být také využit pro ukládání lokálních proměnných nebo předávání parametrů funkcím.

Pomocí instrukce AIS můžeme k registru SP přičíst 8 bitové číslo se znaménkem. Toho je často využíváno při alokaci nebo uvolnění lokálních proměnných.

SP je po restartu nastaven na hodnotu \$00FF z důvodů zpětné kompatibility s M68HC05 rodinou. Programově můžeme nastavit adresu zásobníku kdekoliv ve volné RAM.

Programový čítač (PC - Program Counter)

Programový čítač je 16 bitový registr, který obsahuje adresu další instrukce nebo operandu, který se má vykonat. Během běžného vykonávání programu se čítač automaticky inkrementuje na adresu dalšího operandu nebo instrukce, který se má vykonávat. Operace skoku, přerušení, volání podprogramu nebo návratu nahrají do programového čítače jinou adresu než je následující sekvenční.

Během restartu se do programového čítače nahraje vektor na adrese \$FFFE a \$FFFF. Vektor obsahuje adresu první instrukce, která se má po restartu provést.

Stavový registr (CCR - Condition Code Register)

8 bitový stavový registr obsahuje masku přerušení I a pět příznaků, které indikují výsledky právě vykonané instrukce. Bity 5 a 6 jsou permanentně nastaveny na 1.

- V – Two's Complement Overflow Flag Procesor nastaví příznak přetečení, když nastane přetečení dvojkového doplňku. Tento příznak využívají instrukce BGT, BGE, BLE a BLT.

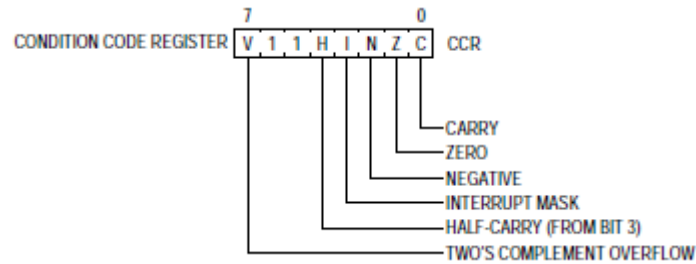
- H – Half-Carry Flag. Tento příznak se nastaví, když dojde k přenosu v akumulátoru z 3 na 4 bit. Příznak se využívá při operacích v BCD kódu. Příkaz DAA podle něj reviduje součet pomocí ADD (standardní binární součet) na součet v kódu BCD.

- I — Interrupt Mask Bit. Pokud je nastavena maska přerušení, všechna maskovaná přerušení jsou zakázána. Přerušení se povolí, když je maska nastavena na 0. Když nastane přerušení, tak se maska přerušení automaticky nastaví na 1, po uložení registru na zásobník.

- N — Negative Flag Procesor nastaví tento příznak, když aritmetická, logická nebo operace přesunu obsahuje záporný výsledek. To znamená, že je nastaven 7 bit. Změnu příznaku také změní prosté načtení nebo uložení 8 nebo 16 bitové hodnoty, pokud je nastaven nejvýznamnější bit na 1.

- Z — Zero Flag Příznak se nastaví, pokud je provedena aritmetická, logická nebo operace přesunu s nulovým výsledkem. I prosté načtení hodnoty čísel \$00 nebo \$0000 má za následek vyvolání tohoto příznaku.

- C — Carry/Borrow Flag se nastaví, pokud při operaci sčítání nastane přenos do vyššího řádu nebo operace odečítání potřebuje výpůjčku.



Obr. 3 Vnitřní struktura registru CCR

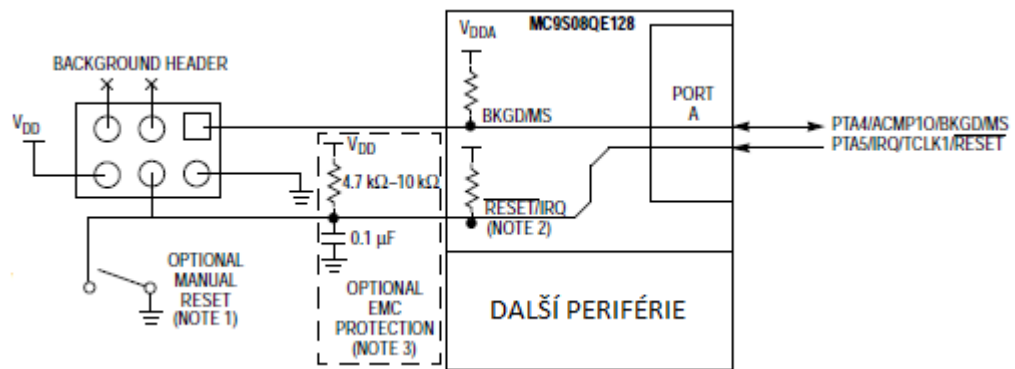
1.2 BDM

Všechny MCU v rodině HCS08 obsahují jednovodičové ladicí rozhraní. Toto rozhraní podporuje programování přímo v zapojení na desce a sofistikované ladění. Na rozdíl od ladění na starších 8bit MCU. Nepoužívá žádné uživatelské paměti nebo místa v paměťové mapě a nesdílí žádné periferie. [3]

BDC příkazy jsou rozděleny do dvou skupin:

- Aktivní příkazy režimu na pozadí požadují, aby bylo MCU v aktivním režimu na pozadí (Uživatelský program není spuštěn). Aktivní příkazy režimu na pozadí umožní, aby mohly být CPU registry čteny nebo zapisovány a umožňují uživateli sledovat instrukce, nebo přejít do uživatelského programu z aktivního režimu na pozadí.
- Ladící příkazy lze spustit kdykoliv, i když je uživatelský program spuštěn. Ladící příkazy umožňují uživateli číst nebo zapisovat do paměťových míst nebo přistupovat do příznakových a ovládacích registrů v rámci ladění MCU.

Typicky je používáno poměrně jednoduché rozhraní k převádění příkazů z hostitelského počítače na příkazy pro vlastní sériové jednopínové rozhraní k ladění systému. V závislosti na zvoleném BDM kitu, může být toto rozhraní připojeno standardním RS-232 sériovým portem, paralelním portem tiskárny nebo nějakým jiným typem komunikace, jako je USB, pro komunikaci mezi hostitelským PC a MCU. Vyvedená propojka se typicky připojuje k cílovému MCU přes zem, pin BKGD, RESET a, v případě potřeby, VDD [4]. Tlačítko připojené k RESET pinu a zemi je užitečné pro znovuzískání kontroly nad MCU nebo k ovládní spuštění cílového systému po permanentním naprogramování. Někdy může být zapojen VDD, aby nemuselo být napájeno MCU externě. Je-li však MCU napájen samostatně, může být BDM kit připojen k běžícímu systému bez vynucení resetu nebo jiného narušení běhu aplikace.



Obr. 4 Zapojení BDC s MC9S08QE128

BDC příkazy

BDC příkazy jsou posílány sériově z hostitelského počítače na BKGD pin cílové HCS08 MCU. Všechny příkazy a data jsou odeslány nejvyšším bitem napřed pomocí vlastního BDC komunikačního protokolu. Aktivní příkazy na pozadí vyžadují, aby bylo MCU v době vysílání v aktivním režimu na pozadí. Zatímco neaktivní příkazy mohou být vydány kdykoli, bez ohledu, zda je cílové MCU v aktivním režimu na pozadí nebo běží uživatelská aplikace.

1.2.1 Komunikace

Sériové rozhraní BDC vyžaduje externí řadič pro generování sestupné hrany na vývodu BKGD k indikaci začátku přenosu. Externí řadič poskytuje tuto sestupnou hranu bez ohledu, zda byla data vyslána nebo přijata. Data jsou přenášena přes bit s nejvyšší hodnotou do 16 BDC hodinových cyklů za bit (nominální rychlost). Rozhraní přeteče, když dojde k 512 BDC hodinovým cyklům.

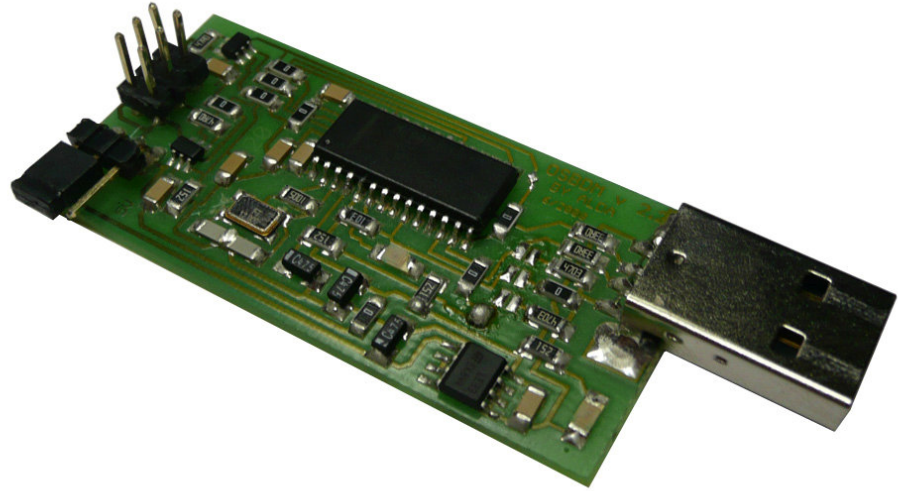
Jakékoliv příkazy BDC, které byly v průběhu, kdy nastalo přetečení, odeslány, jsou zamítnuty bez ovlivnění paměti nebo provozního režimu MCU.

Hodinový CLKS_W bit ve stavovém BDC registru umožňuje uživateli vybrat BDC hodinový kmitočet. Hodinovým BDC zdrojem mohou být buď sběrnice, nebo alternativní BDC hodinový zdroj. Pin BKGD může přijímat vysokou nebo nízkou úroveň signálu, nebo přenášet vysoké nebo nízké úrovně signálu.

1.2.2 Open Source BDM

OSBDM je free alternativa ke komerčním vývojovým deskám, které disponují rozhraním BDM. Na internetu se nachází schéma i s popisem součástek. Bohužel součástky jsou

v našich podmínkách špatně sehnatelné. Právě proto nabízí server HW.cz možnost zakoupení jako stavebnice.



Obr. 5 Open Source BDM

2 PULZNĚ ŠÍŘKOVÁ MODULACE

PWM je velmi účinný způsob, jak poskytovat střední množství elektrické energie mezi stavy zapnuto nebo vypnuto. V minulosti, kdy byl potřebný pouze částečný výkon, například pro motor šicího stroje, dal se použít reostat nacházející se v nožním pedálu šicího stroje. Byl zapojen v sérii s motorem, který upravoval množství proudu, které teče přes motor. To byl neefektivní systém, ale přijatelný, protože celkový výkon byl nízký. Byl to jeden z několika způsobů ovládání výkonu. Byly i jiné způsoby, některé se ještě používají. Jako variabilní autotransformátory, včetně Autrastatu pro divadelní osvětlení a Variac na celkovou úpravu napájení ze sítě. Tato řešení byla docela účinná, ale také poměrně nákladná.

Pojem pracovní cyklus popisuje podíl času na pravidelném intervalu nebo časovém období. Nízký pracovní cyklus odpovídá nízkému výkonu, protože je napájení vypnuto po většinu času. Poměr je vyjádřen v procentech, 100% odpovídá stále zapnuto. PWM pracuje dobře s digitálním ovládáním, které lze z důvodu dvou stavů zapnuto / vypnuto snadno nastavit potřebám pracovního cyklu.

2.1 Princip

Pulzně šířková modulace používá obdélníkové vlny, jejichž šířka impulsu je modulována a má za následek změnu průměrné hodnoty křivky. Pokud uvažíme pulzní průběh $f(t)$ s nízkou hodnotou y_{\min} , vysokou hodnotu y_{\max} a pracovní cyklus D , průměrná hodnota křivky je dána vztahem:

$$\bar{y} = \frac{1}{T} \left(\int_0^T f(t) dt \right) \quad (1)$$

Hodnota pulzní vlny $f(t)$ je pro $y_{\max} = 0 < t < D \cdot T$ a pro $y_{\min} = D \cdot T < t < T$. Z výše uvedeného výrazu se následně stává:

$$\bar{y} = \frac{1}{T} \left(\int_0^{DT} y_{\max} dt + \int_{DT}^T y_{\min} dt \right) = D \cdot y_{\max} + (1 - D) y_{\min} \quad (2)$$

Tento výraz (2) lze zjednodušit v mnoha případech, kdy $y_{\min} = 0$, je:

$$\bar{y} = D \cdot y_{\max} \quad (3)$$

Z toho je zřejmé, že průměrné hodnoty signálu (\bar{y}) jsou přímo úměrné na pracovním cyklu D.

2.2 Druhy modulace

Nejjednodušším způsobem, jak generovat PWM signál, je srovnávací metoda, která vyžaduje pouze pilový nebo trojúhelníkový průběh a komparátor. Jestliže je hodnota referenčního signálu větší než modulační křivky, signál PWM je ve vysokém stavu, jinak je v nízkém stavu.

Delta

V použití delta modulace pro ovládání PWM je výstupní signál integrovaný a výsledek se porovnává s limity, které odpovídají referenčnímu konstantnímu signálu. Pokaždé, když integrál výstupního signálu dosáhne jednoho z limitů, PWM signál změní stav.

Delta-sigma

V delta-sigma modulaci, jako způsobu ovládání PWM, se výstupní signál odečte od referenčního signálu k formě chyby signálu. Tato chyba je integrována a když integrál chyb překračuje meze, výstup změní stav.

Časově dávkovací

Mnoho digitálních obvodů umožňuje generovat PWM signál (např. mnohé mikroprocesory mají PWM výstupy). Normálně používají čítač, který se pravidelně inkrementuje. Buď je připojen přímo, nebo nepřímo k hodinovému obvodu. A resetuje se na konci každého časového cyklu PWM. Když je hodnota čítače větší než referenční hodnota, PWM výstup změní stav z vysoké na nízkou, případně od nejnižší k nejvyšší.

Prostor vektorová modulace

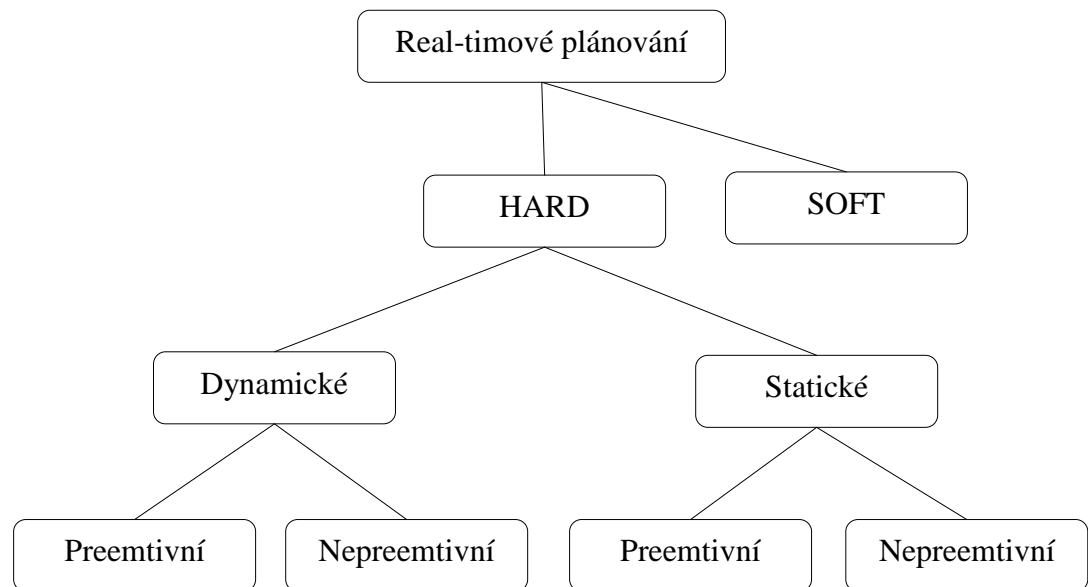
SVM je algoritmus pro kontrolu modulace šířkou pulzu. Používá se pro vytvoření AC křivek, nejčastěji k řízení 3 fázového AC pohonu motorů na různých rychlostech pomocí více Class-D zesilovače. Existují různé varianty SVM, které vedou k různé kvalitě a mají různé výpočetní požadavky. Jednou z aktivních oblastí rozvoje je snížení celkového harmonického zkreslení a vytvoření vlastního rychlého přechodu na tyto algoritmy.

3 REALTIMOVÉ SYSTÉMY

Real-time systémy (RTS) se stávají všudypřítomné. Mezi typické příklady real-time systémů patří řízení letového provozu, síťové multimediální systémy, řídicí systémy, kosmické systémy a embedded automobilová elektronika. Správnost chování RTS závisí nejen na logickém výsledku výpočtů, ale také na fyzické instanci, která tyto výsledky produkuje. RTS jsou klasifikovány z několika úhlů pohledu, tj. na faktory mimo počítačový systém a faktory uvnitř počítačového systému. Zvláštní důraz je kladen na HARD a SOFT RTS. Zmeškání lhůty uvedené v HARD real-time systémů je katastrofální, ale u SOFT real-time systémů to může vést jen k výrazné ztrátě. Proto je předvídatelnost chování systému nejdůležitější v oblasti týkající se těchto systémů. Předvídatelnosti je často dosaženo buď statickým, nebo dynamickým plánováním plnit své termíny. Statické plánování umožňuje plánování rozhodnutí při kompilaci a je off-line. Dynamické plánování je online a používá plánovací test k určení, zda soubor úkolů zvládne plnit své termíny. [11]

Real-time systém mění svůj stav v závislosti na fyzickém času, např. chemická reakce mění svůj stav i poté, co byl ovládací počítačový systém zastaven. Na základě této vlastnosti lze RTS rozdělit na tyto subsystemy: řízení objektu, real-time počítačový systém a lidský operátor. Real-time počítačový systém musí reagovat na podněty z kontrolovaného objektu v časových intervalech, které jsou diktovány jeho prostředím.

HARD real-time systém musí vykonat současně řadu real-time úkolů takovým způsobem, že po celou dobu jsou jeho kritické úkoly schopny plnit své stanovené termíny. Každá úloha potřebuje výpočetní a datové zdroje pro dokončení práce. Problém plánování se týká přidělování zdrojů na uspokojení časového omezení. Obrázek (Obr. 6) představuje taxonomii real-time plánování algoritmu.



Obr. 6 RTS plánování

HARD real-time plánování se obecně dělí na dva typy: statické a dynamické. Ve statickém plánování jsou plánování rozhodnutí učiněna při kompilaci. Run-time rozvrh je generován off-line, na základě předchozí znalosti úkolu, např. maximální doba realizace, přednost omezení, vzájemné vyloučení omezení a termíny. Takže run-time režie je malá. Naproti tomu - dynamické plánování dělá plánování rozhodnutí za běhu, zvolením jednoho ze stávajícího souboru připravených úkolů. Dynamické plánovače jsou pružné a přizpůsobivé. Ale mohou vzniknout značné režijní náklady. Preemptivní nebo nepreemptivní plánování úkolů je možné ve statickém i dynamickém plánování. V preemptivním plánování bude úkol vykonávající se v současné době předběhnout, při příchodu úkolů s vyšší prioritou. V nepreemptivním plánování se vykonávající úkol nesmí předběhnout až do jeho dokončení. Real-time operační systémy (RTOS) lze použít k poskytování předvídatelné služby aplikacím.

Hodnotící monotónní algoritmus (RMA)

RMA je dynamický preemptivní algoritmus založený na statické prioritě. Míra monotónního algoritmu přiřazuje statické priority na základě období úkolu. Zde je období úkolu doba, po které se opakují úkoly a inverzní období je doba do připraveného stavu. Například úkol s periodou 10ms se opakuje každých 10ms. Úkol s nejkratší dobou dostane nejvyšší prioritou a úkol s nejdelším obdobím dostane nejnižší statickou prioritu. Podle RMA může nezávislý periodický úkol dosáhnout lhůty deadline, pokud je součet využití faktorů u n úkolů dle následujícího vzorce:

$$\sum (C_i / T_i) \leq U(n) = n(2^{1/n} - 1) \quad (4)$$

Kde C_i je nejhorší čas provedení, T_i čas periody, $U(n)$ využití vázané pro n úkolů.

Nejzazší lhůta první (EDF)

EDF algoritmus je optimální dynamický preemptivní algoritmus založený na dynamické prioritě. Po každé významnější události je úkolu s nejbližší deadline přiřazena nejvyšší dynamická priorita. Významnou událostí v systému může být blokování úkolu, vyzvání úkolu, dokončení úkolu. Využití procesoru může dosáhnout až 100%, i když období úkolu není násobkem nejmenšího období. Dispečer funguje stejným způsobem jako dispečer u RMA.

Priorita horní meze protokolu:

Priorita horní meze protokolu se používá k rozvrhu nastavení závislých pravidelných úkolů, které sdílejí zdroje chráněné semaforem. Sdílené zdroje, např. společné datové struktury, jsou používány pro meziprocesové komunikace. Sdílení zdrojů může vést k nespoutané inverzi priorit. Protokoly priority horní meze byly vyvinuty, aby se minimalizovala prioritní inverze a blokování času.

Statické plánovací algoritmy

Ve statickém plánování jsou rozhodnutí učiněna během kompilace. To předpokládá znalost priorit ze všech úkolů a na tomto staví harmonogram. Jakmile je harmonogram sestaven, nemůže být upraven online. Statické plánování se obecně nedoporučuje pro dynamické systémy. Aplikace, jako je řízení procesu, mohou těžit z tohoto plánování, kde jsou data snímačů u všech úkolů známa před kompilací. Neexistuje žádná výslovně statická plánovací technika kromě té, že harmonogram je povinen lhůtu žádosti dodržet podle známé konfigurace systému. Většinou se pojem priority ve statickém plánování nevyskytuje.

3.1 RTMON

Jedná se o speciálně napsaný real-timový operační systém pro MC9S08GB60. Avšak pro potřeby tohoto řešení musel být upraven, aby fungoval na použitém MCU. Tento RTOS je

napsán v jazyce C. Proces je tedy řešen jako funkce jazyka C. Umožňuje preemptivní multitasking.

Maximálně lze použít 10 procesů o celkové velikosti zásobníku 1kB.

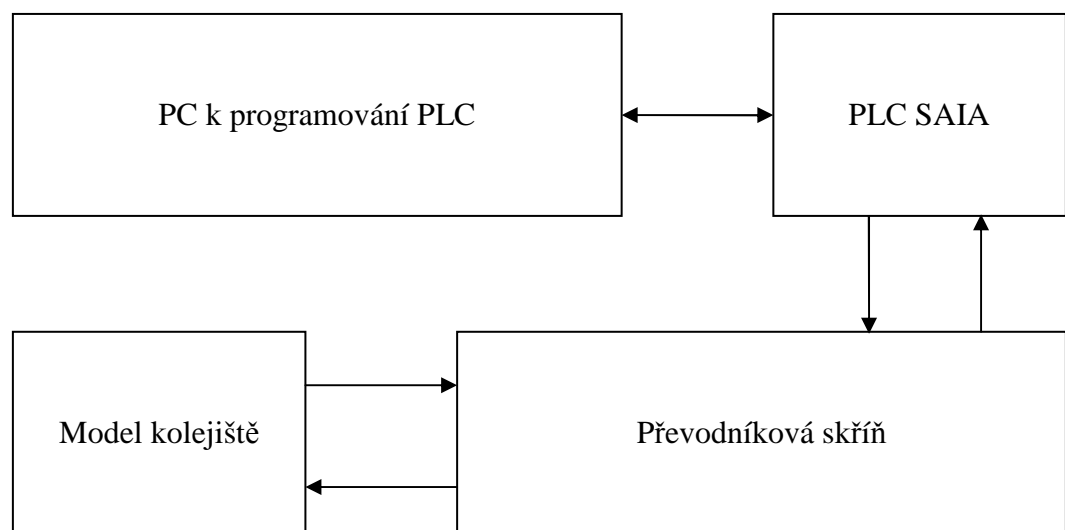
Podrobný přehled funkcí

- char **rtm_init**(IDPROC** init_id);
 - Inicializace systému RTMON. Musí být volána na začátku programu.
- char **rtm_end**(IDPROC* init_id);
 - Ukončení činnosti RTMON. Volá se na konci programu.
- char **rtm_create_p**(const char* pname, unsigned char prio, void(*pfunc)(), int stack_size, IDPROC** proc_id);
 - Vytvoří nový proces.
 - Pname – jméno procesu, nevyužívá se.
 - Prio – priorita procesu, 1 až 254.
 - Pfunc – ukazatel na funkci, která představuje tělo procesu.
 - Stack_size – velikost zásobníku pro proces. Doporučená hodnota 32 Byte a více.
 - Proc_id – proměnná, do které se uloží identifikátor vytvořeného procesu. (IDPROC)
- char **rtm_start_p**(IDPROC* proc_id, int time_to_start, int time_period);
 - Spuštění procesu
 - Proc_id – identifikátor procesu, který se má spustit
 - Time_to_start – počet tiků do startu procesu. 0, pokud se má spustit okamžitě.
 - Time_period – perioda spouštění procesu v ticích. 0, pokud má být proces spuštěn jednorázově.
- char **rtm_delay_p**(IDPROC* proc_id, int time_to_delay);
 - Pozastavení běhu procesu.

- Proc_id – identifikátor procesu, který se má pozastavit.
- Time_to_delay – doba pozastavení v ticích. 0, pokud se má pozastavit na neomezenou dobu (dokud nebude spuštěn jiným procesem).
- char **rtm_continue_p**(IDPROC * proc_id);
 - Pokračování v běhu pozastaveného procesu.

4 MODEL KOLEJIŠTĚ PRO PODPORU VÝUKY PLC

Ve výuce programovatelných automatů na Fakultě Aplikované Informatiky se pro výuku používá model vlakového kolejiště. Model s kolejemi zkonstruoval Ing. Vladimír Hnilica jako svou bakalářskou práci. Později se k tomuto tématu vrátil a jako svou diplomovou práci realizoval převodníkovou skříň, aby bylo možné propojit kolejiště s PC vybaveným technologickou kartou. Časem se přidalo propojení s PLC SAIA PCD2. Praktickou zkušeností z výuky se zjistilo, že PLC není schopno plně řídit modulaci a tím efektivně ovládat rychlost vlaků, aby jezdily plynule.



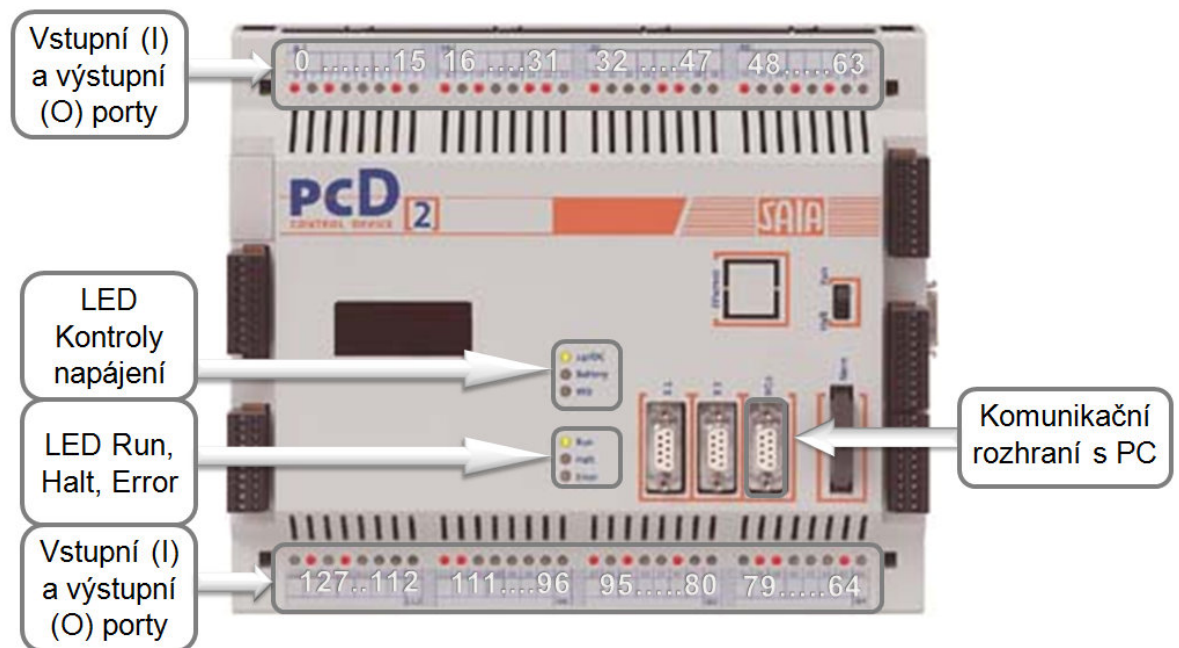
Obr. 7 Blokové schéma současného stavu

Na obrázku (Obr. 7) je zobrazen současný stav propojení ovládacích prvků kolejiště. První blok znázorňuje PC, kterým se programuje SAIA PCD2, také se tímto počítačem mohou monitorovat jednotlivé stavy PLC. PLC je propojeno s převodníkovou skříň, se kterou komunikuje TTL logikou. Tato převodníková skříň zajišťuje odezvu s čidly, osvětlením, výhybkami a semaforem na modelu kolejiště.

4.1 PLC SAIA PCD2

Automaty PCD2 tvoří kombinace operačního systému, CPU, vstupních/výstupních modulů, komunikačních submodulů a programovacích nástrojů. Všechny tyto komponenty byly vyvinuty a jsou vyráběny v Saia-Burgess Controls. Dokonalá znalost všech prvků automatu umožnila firmě úspěšně realizovat systém, který je koncipován jako velmi otevřený a adaptibilní. Páteří systému je základní procesorová jednotka (CPU). K dispozici jsou různá provedení, poskytující široké spektrum funkcí. Do každé základní jednotky PCD2 a rozšíření se libovolně umísťují V/V moduly, kterých je přes 40 typů, ve stejném

provedení existují i 2 typy modemů. Rozšiřovací jednotky PCD2.C150 / C100 poskytují místo pro další 4 nebo 8 V/V modulů (až do 255 V/V). Rozšiřovací základny PCD3.C100 / C110 a PCD3.C200 v kombinaci s PCD2 umožňují napojit až 1023 V/V při použití CPU PCD2.M480.



Obr. 8 PLC SAISA PCD2

4.2 Možné způsoby programování SAIA PCD2

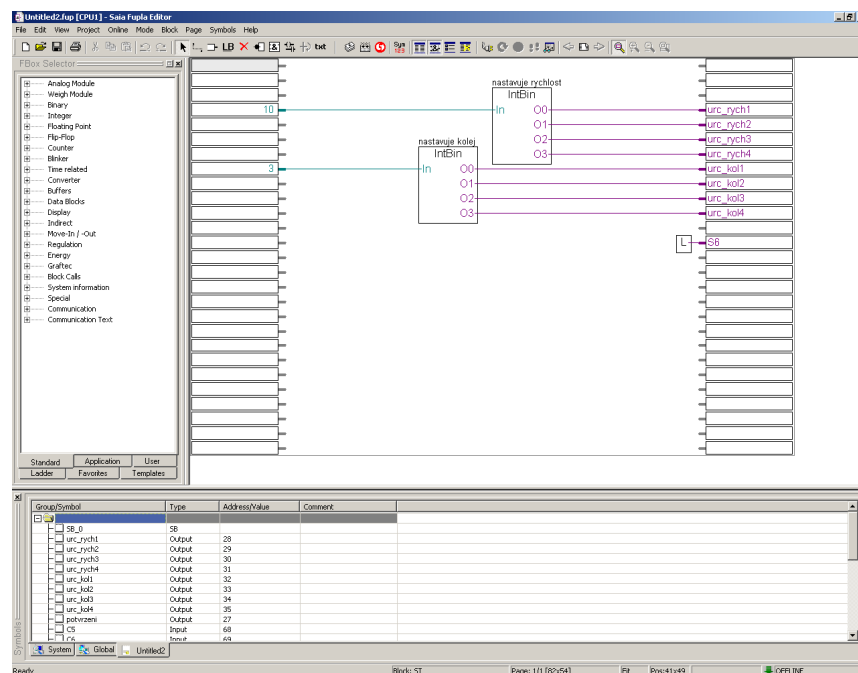
K tvorbě programů pro programovatelné automaty SAIA PCD je výrobcem standardně dodáváno vývojové prostředí PG4. Jedná se o balík nástrojů pro tvorbu jednotlivých rutin v různých programovacích jazycích. Jejich kompilování, správu knihovny takto vzniklých objektů, linkování programů a správu celých projektů. Pomocí PG4 lze on-line připojením k programovatelnému automatu měnit jeho konfiguraci, za chodu měnit hodnoty proměnných a výstupních portů. Navíc lze za chodu monitorovat PLC přímo na obrazovce PC.

AWL – Instruction List (assembler), editor AWL (textový editor)

Toto je klasický způsob programování v assembleru za použití jednotlivých elementárních instrukcí. K zápisu slouží klasický textový editor, řádky se nečíslují, komentáře se oddělují středníkem.

FBD/LD – Function Block Diagram / Ladder Diagram (schéma funkčních bloků / FUPLA editor)

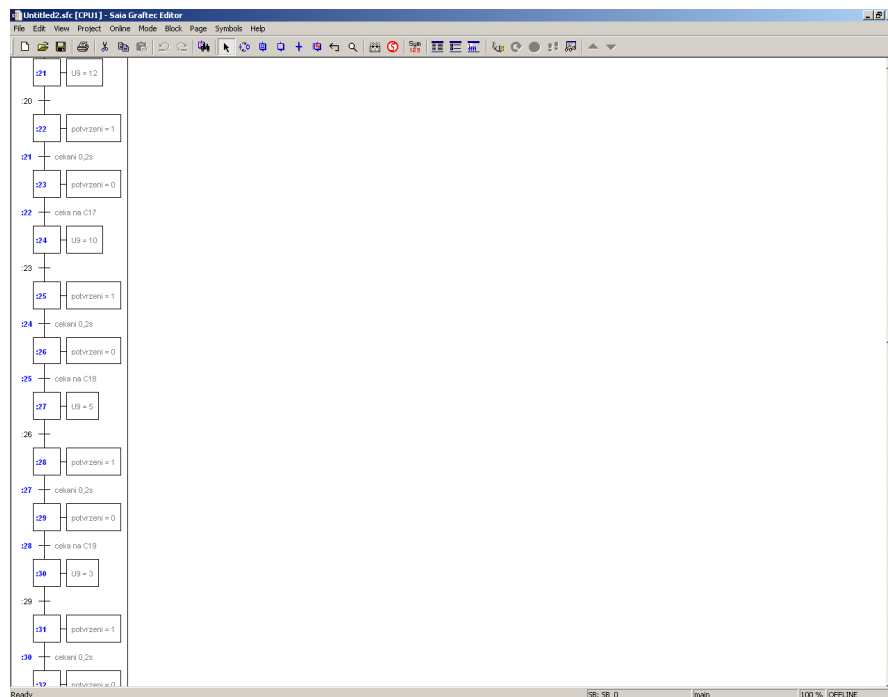
Toto je klasický způsob programování v assembleru za použití jednotlivých elementárních instrukcí. K zápisu slouží klasický textový editor, řádky se nečíslují, komentáře se oddělují středníkem. Jak je patrné z obrázku (Obr. 9), na levé a pravé straně pracovní oblasti je žebříková umísťovací plocha a oblast uprostřed slouží pro vkládání jednotlivých funkcí. Do žebříkové plochy na levém okraji okna se vepisují vstupní data – konstanty, registry, vstupy nebo výstupy a podobně, do pravé sběrnice se zapisují výstupní data přijímající hodnoty z levé sběrnice, avšak zpracované přes nejrůznější funkce umístěné mezi levou a pravou sběrnici. Program je sestaven z funkcí, které jsou zastoupeny obdélníky (mající na levé straně vstupy a na pravé výstupy). Předávání dat mezi jednotlivými funkcemi je realizováno spojnicemi. Výsledný program je tedy vykonáván z levé strany na pravou stranu a shora dolů. Při konečném překladači programu jsou zástupné symboly jednotlivých funkcí nahrazeny makry v logické návaznosti a symbolické proměnné se nahradí konkrétními čísly datových prostředků.



Obr. 9 Okno editoru FUPLA

SFC – Graftec – sekvenční programování

Graftec je grafický jazyk pro zápis sekvenčních algoritmů. Postup provádění programu je dán vývojovým diagramem složeným z procedur kroků (ST) a vyhodnocováním přechodů (TR). Procedury kroků i vyhodnocování přechodů mezi nimi mohou být zapsány buď v jazyce FUPLA, nebo v jazyce IL.



Obr. 10 Okno editoru Graftec

4.3 Model kolejště

Kolejiště je zasazené do imaginární krajiny o rozloze zhruba 2m². Na této ploše je rozmístěno 10 traťových úseků, 2 nádraží s možností osvětlení, most, tunel, 3 železniční přejezdy, návěstidla a množství čidel pro optimální řízení vlaku.



Obr. 11 Model kolejště pohled shora

Následující tabulky zobrazují označení prvků a jejich adres (fyz. vstupů a výstupů).

Tab. 1 Adresy čidel

Označení	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Adresa	I64	I65	I66	I67	I68	I69	I70	I71	I72	I73	I74	I75
Označení	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24
Adresa	I76	I77	I78	I79	I80	I81	I82	I83	I84	I85	I86	I87

Tab. 2 Adresy semaforů

Označení	S1	S2	S3	S4	S5	S6	S11	S12	S13	S14	S15	S16
Adresa	O55	O54	O53	O52	O51	O50	O49	O48	O47	O46	O45	O44

Tab. 3 Adresy výhybek

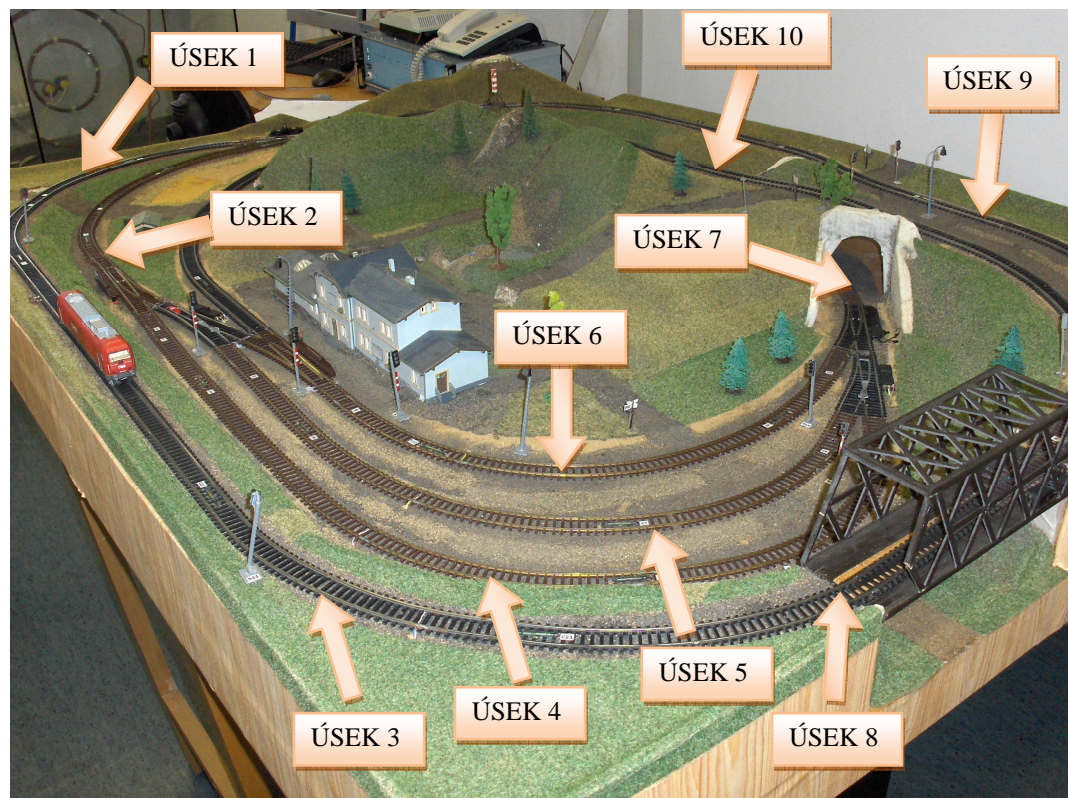
Označení	V1	V2	V3	V4	V5	V6	K1 C,D	K1 A,B
Adresa	O63	O52	O61	O60	O59	O58	O57	O56

Tab. 4 Adresy přejezdů

Označení	P1	P2	P3
Adresa	O36	O37	O38

Tab. 5 Adresy osvětlení

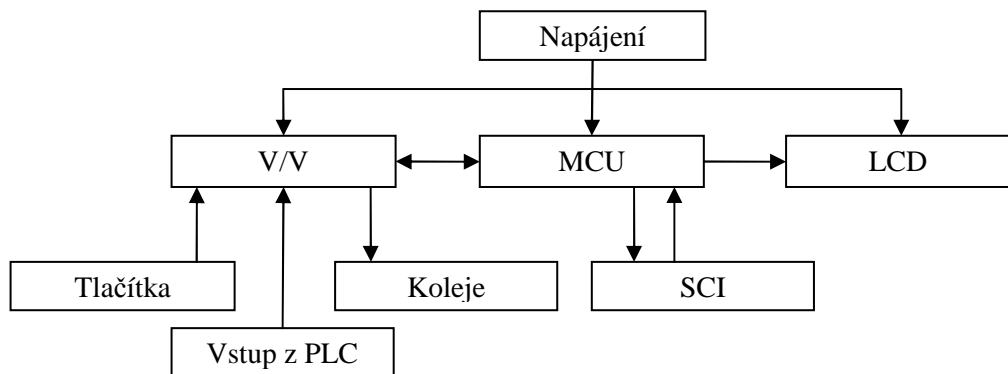
Označení	N1	N2	N3
Adresa	O39	O40	O41



Obr. 12 Boční pohled na kolejiště s vyznačenými úseky

II. PRAKTICKÁ ČÁST

5 NÁVRH HARDWARE MODULU



Obr. 13 Blokové schéma modulu

Jak je vidět z výše uvedeného obrázku (Obr. 14) - modul se skládá z několika částí, které jsou vzájemně propojeny. Základem je napájení, které je rozvedeno k LCD jednotce, MCU a vstupně výstupním periferiím. Blok MCU komunikuje s blokem SCI, který zajišťuje sériovou komunikaci s PC.

Do vstupně výstupního bloku vedou tlačítka pro manuální ovládání. Vstup z PLC blok v sobě zahrnuje opto-elektricky oddělené vstupy z PLC. Výstup z V/V bloku vede do výstupu směřujícího k převodníkové skříni a následně kolejiště. Kompletní schémata zapojení jsou v příloze P I pro řídicí jednotku a P II pro modul displeje.

5.1 Mikrokontrolér

Pro řešení této práce byl použit mikrokontrolér rodiny HCS08 firmy Freescale [4]. Jedná se o 8b řadu, která je však pinově kompatibilní s 32b řadou Flexis V1. Zde použitý MCU je QE128 v 80 pinovém pouzdře, jak je vidět na obrázku (Obr. 15).



Obr. 14 QE128 v 80 pinovém pouzdře

5.2 Displej

5.2.1 LCD modul ATM12864D

LCD modul ATM12864D a veškerá obslužná elektronika byla převzata z bakalářské práce Bc. Jiřího Sedláka [10]. Protože jeho řešení bylo určené do výuky předmětu MPC konkrétně na mikrokontroler MC9S08GB, bylo tedy zapotřebí provést patřičné změny zapojení i obslužného kódu.

LCD modul ATM12864D je grafický displej s rozlišením 128x64 bodů a se zabudovaným LED podsvícením. Díky integrovanému řadiči KS0107B se tento modul vyznačuje nízkou spotřebou, rozhraním pro komunikaci s 8b mikropočítačem a napájením 4.5 až 5.5V.

5.2.2 Modifikace zapojení

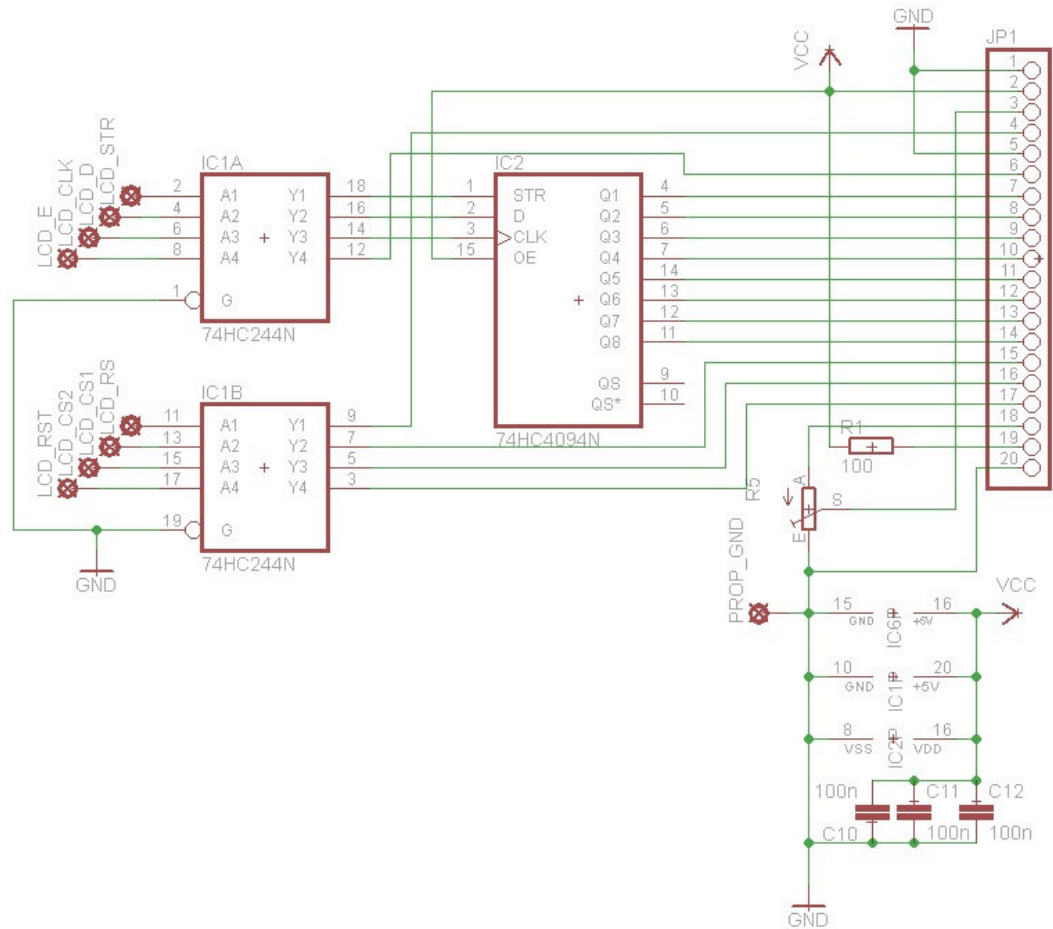
Největší změny doznalo propojení s MPC. Na desce A je z mikropočítače vyvedeno 8 pinů na propojky s deskou B. Těchto 8 pinů vedou z portu PTH a mají funkci, jak ukazuje tabulka (Tab. 7).

Tab. 6 Význam pinů

Číslo pinu	Název portu	Funkce
3	PTH7	RES
4	PTH6	CS1
5	PTH5	CS2
6	PTH4	E
16	PTH3	R/S
17	PTH2	CLK
18	PTH1	D
19	PTH0	STR

K obvodům IC1 a IC2 se přidaly blokovací kondenzátory C11, respektive C12. Také se použil rychlejší CMOS obvod 74HC4094. Také bylo zapotřebí přepsat obslužný program, aby se dalo LCD používat s QE128.

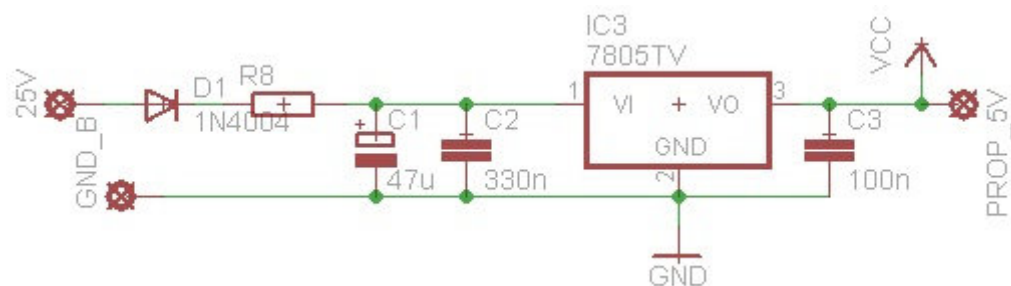
Obslužný kód byl napsán v jazyce Assembler. Proto, aby tento kód fungoval s QE128, stačilo přepsání adresovacích příkazů.



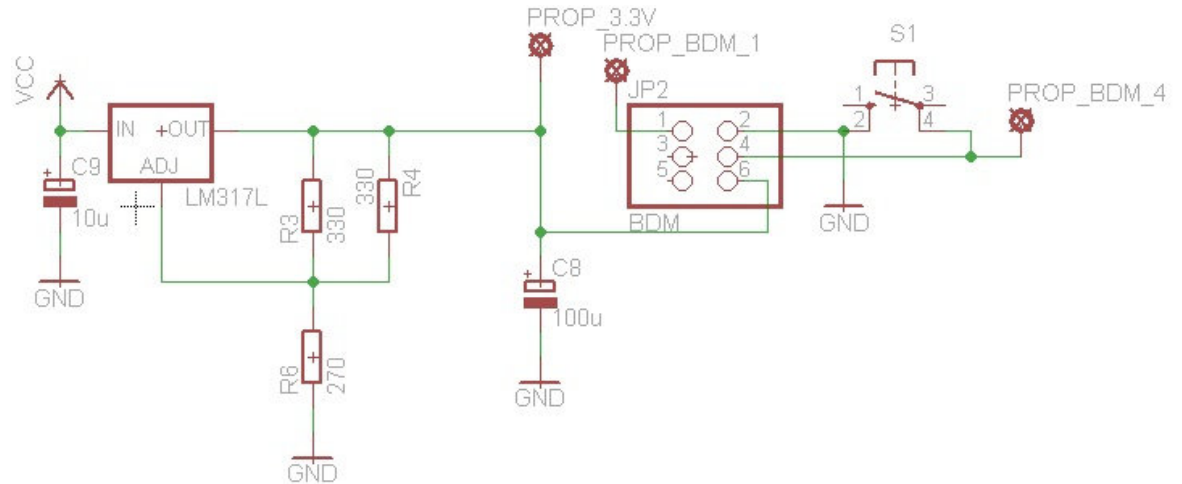
Obr. 15 Schéma zapojení LCD

5.3 Napájení

Přívodní napájení je 24V z PLC nebo 9V z adaptéru. V závislosti, které napájení je zvoleno, se umísťuje předřadný odpor R8. Vůči přepólování je napájení chráněno diodou D1. Pro potřeby LCD, optočlenů a dalších čipů se přívodní napájení reguluje na 5V pomocí IC 7805TV. Při této regulaci je produkováno velké množství tepla a proto je na IC3 ještě pasivní chladič. Mikrokontrolér potřebuje pro svůj chod napětí 3,3V, které se získává z 5V pomocí IC4 LM317L. Toto napětí je propojeno s deskou A pomocí propojky, na které je umístěn MCU. Na desce B je napájen tímto napětím také BDM konektor.



Obr. 16 Schéma 5V napájení



Obr. 17 Schéma 3.3V napájení a BDM

5.4 Propojení modulu s PLC

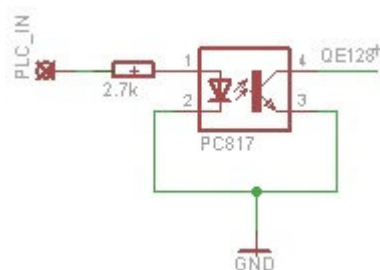
Toto propojení je realizováno na desce A. Na vstupu do krabice je CANNON 25F, která je zapojena podle tabulky (Tab. 8). Propojení s deskou A a mikrokontrolérem je v tabulce (Tab. 9). Aby nedošlo ke zničení MCU chybným zapojením, je PLC odděleno s MCU pomocí optočlenů OK1 – OK10. Na obrázku (Obr. 19) je vzorové zapojení jednoho z 10 optočlenů. [5]

Tab. 7 Propojení PCL s MCU

Číslo pinu	Název portu	Funkce
77	PTC5	BYPASS
76	PTE0	POTVRZENÍ
74	PTG0	URC_RYCH1
73	PTG1	URC_RYCH2
72	PTG2	URC_RYCH3
71	PTG3	URC_RYCH4
68	PTG4	URC_KOL1
67	PTG5	URC_KOL2
66	PTG6	URC_KOL3
65	PTG7	URC_KOL4

Tab. 8 Význam pinů na vstupu CANNON 25F

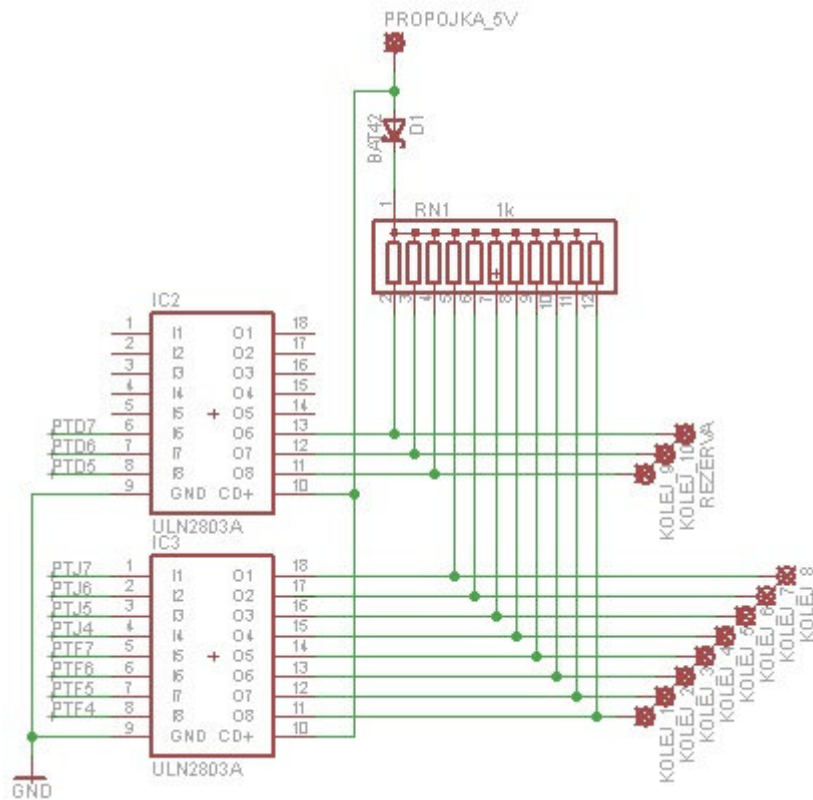
CANNON 25F	Název pinu	Funkce
13	URC_KOL4	horní bit nastavení koleje
12	URC_KOL3	nastavení koleje
11	URC_KOL2	nastavení koleje
10	URC_KOL1	dolní bit nastavení koleje
9	URC_RYCH4	horní bit rychlosti
8	URC_RYCH3	určení rychlosti
7	URC_RYCH2	určení rychlosti
6	URC_RYCH1	dolní bit rychlosti
5	POTVRZENÍ	potvrzovací bit
4	BYPASS	pro BYPASS mod
1	GND	Zemění



Obr. 18 Zapojení optočlenů na vstupu modulátoru

5.5 Propojení modulu s kolejištěm

Model kolejiště je propojen s PWM modulem přes konektory CANON 25F. Výstupy z mikrokontroléru jsou na pinech PTF4 - PTF7, PTJ4 - PTJ7 a PTD5 - PTD7. Tyto piny vedou do IC2 a IC3. Převodníková skříň reaguje na log. 1, proto jsou ještě před výstupem z desky pull-up tranzistory připojeny na 5V. Schéma je zobrazeno v tabulce (Tab. 11). Pro testovací účely byla zkonstruována třetí deska s diodami. Schéma je znázorněno na obrázku (Obr. 20). [6]



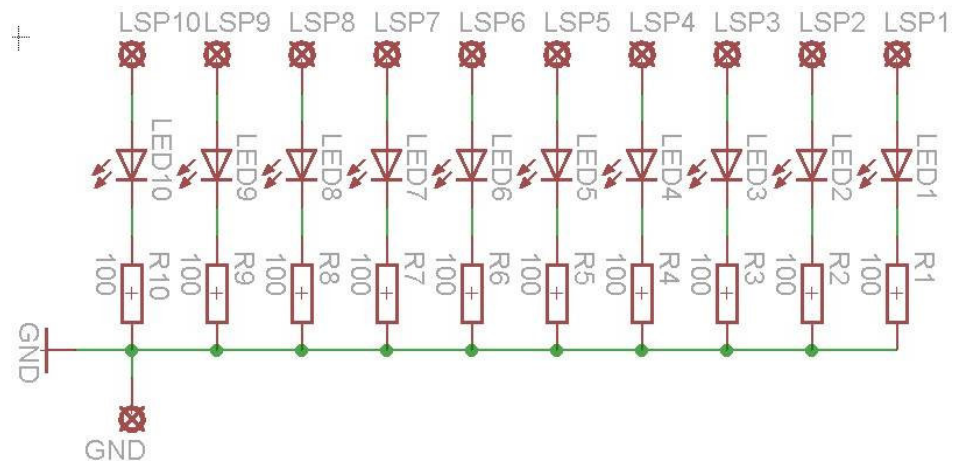
Obr. 19 Schéma zapojení výstupů

Tab. 9 Propojení MCU s kolejištěm

Číslo pinu	Název portu	Funkce
38	PTF4	KOLEJ1
37	PTF5	KOLEJ2
36	PTF6	KOLEJ3
35	PTF7	KOLEJ4
32	PTJ4	KOLEJ5
31	PTJ5	KOLEJ6
30	PTJ6	KOLEJ7
29	PTJ7	KOLEJ8
28	PTD5	KOLEJ9
27	PTD6	KOLEJ10
26	PTD7	REZERVA

Tab. 10 Význam pinů na výstupu

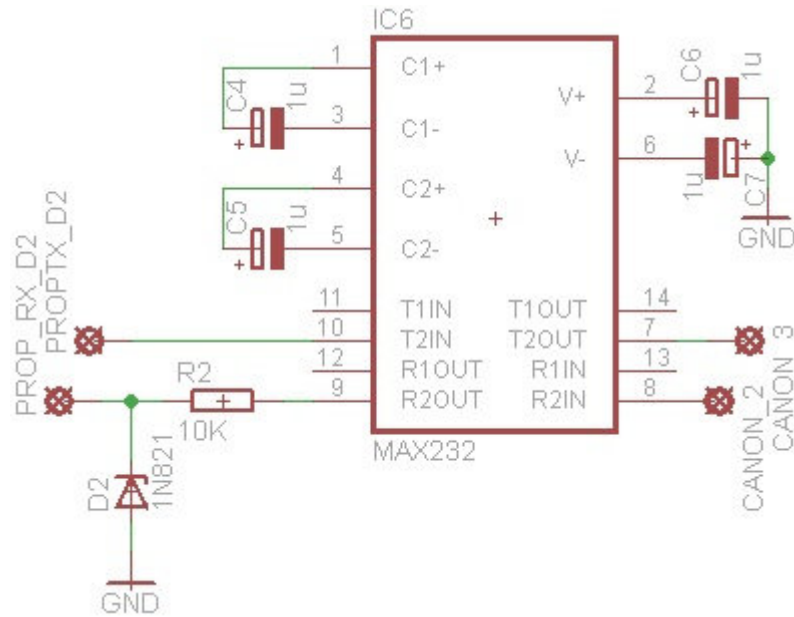
CANNON 25M	Název pinu	Funkce
13	REZERVA	rezervní pin
12	KOLEJ10	vede ke k úseku 10
11	KOLEJ9	vede ke k úseku 9
10	KOLEJ 8	vede ke k úseku 8
9	KOLEJ7	vede ke k úseku 7
8	KOLEJ6	vede ke k úseku 6
7	KOLEJ5	vede ke k úseku 5
6	KOLEJ4	vede ke k úseku 4
5	KOLEJ3	vede ke k úseku 3
4	KOLEJ2	vede ke k úseku 2
3	KOLEJ1	vede ke k úseku 1
1	GND	Zemění



Obr. 20 Schéma testovací desky PWM

5.6 Komunikační rozhraní SCI

Pro komunikaci se sériovou linkou jsou použity piny PTC6 jako RX_D2 a PTC7 jako TX_D2. RX_D2 a TX_D2 jsou použity, protože tyto piny nebyly využity, jako tomu je u RX_D1 a TX_D1. Tyto piny jsou vyvedeny na desku B do IO obvodu MAX232 podle zapojení na obrázku (Obr. 22). Obvod MAX232 se používá jako převodník TTL napěťových úrovní, které generuje MCU na úroveň sériové linky. Obsahuje 2 převodníky TTL → RS232 a 2 převodníky RS232 → TTL.



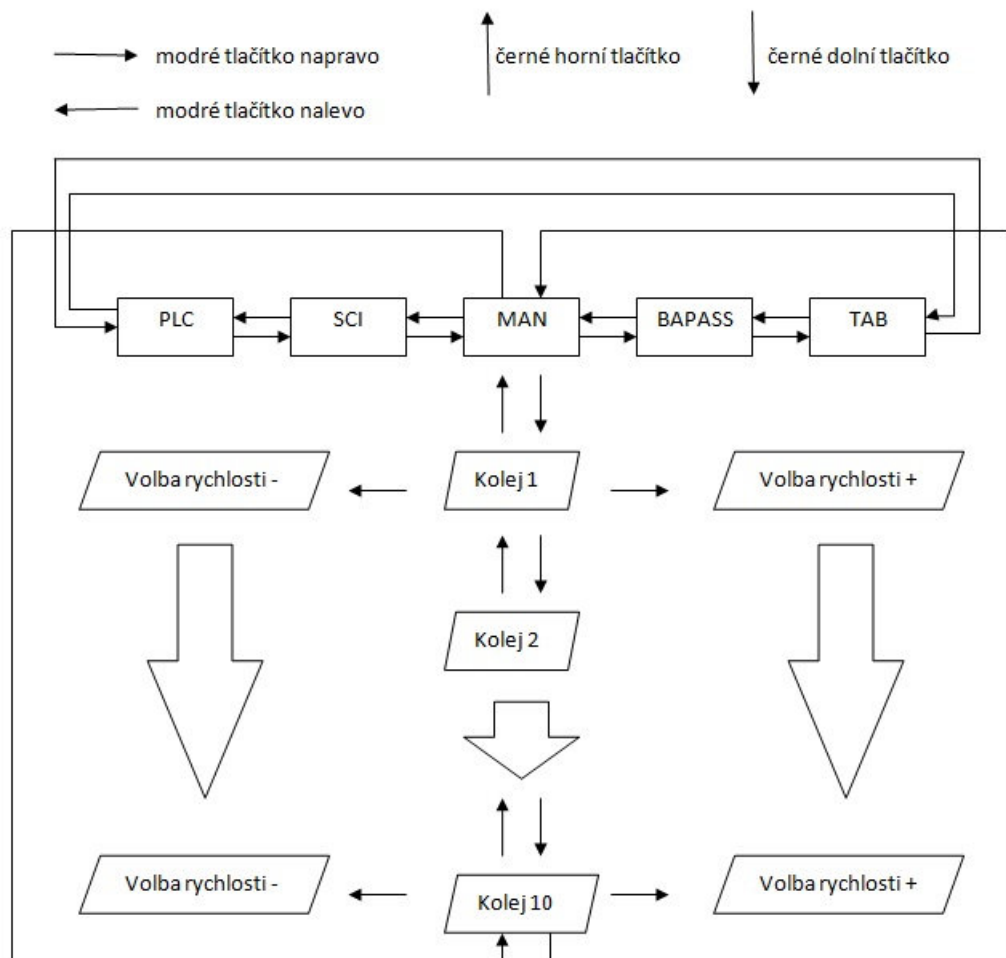
Obr. 21 Zapojení MAX232 pro sériovou komunikaci

5.7 Ovládání

Způsob ovládání zařízení se liší podle aktuálního režimu. Mezi režimy se přepíná pomocí 2 modrých tlačítek, jejichž význam je na obrázku (Obr. 23). Struktura ovládání je na obrázku (Obr. 24). Popis jednotlivých módů je dále.



Obr. 22 Význam tlačítek



Obr. 23 Diagram ovládání

5.7.1 PLC mód

PLC je spojeno s PWM modulátorem pomocí 10 linek, které přenáší signály. Modulátor rozlišuje tyto linky následovně. První 4 linky jsou 4b k určení koleje, další 4 linky jsou 4b k určení rychlosti. Z posledních dvou linek má jeden funkci POTVRZENÍ a druhý je pro zachování kompatibility a je využíván v BYPASS módu. Vodič POTVRZENÍ slouží jako informace pro modul, že má přijmout log. hodnoty na vodičích. Pokud chceme nastavit rychlost na všech kolejích zároveň, stačí, aby byly všechny bity, určené k výběru koleje, na log. 0.

Příklad

Když tedy chceme nastavit kolej 2 na rychlost 8, nastaví se na PLC logické hodnoty dle tabulky (Tab. 12). Poté jen stačí po dobu 0,2s nastavit u PLC výstupu O27 na log. 1 jako potvrzení, že má modulátor načíst hodnoty

Tab. 11 Příklad

PLC IO	Log. hodnota
O28	0
O29	0
O30	0
O31	1
O32	0
O33	1
O34	0
O35	0

5.7.2 SCI mód

Snahou bylo vytvořit ovládání modulu, které by připomínalo příkazovou řádku operačního systému Unix. Proto, když se přepne na ovládání SCI, zobrazí se na terminálu „*pwm modulace* > “. Za tento text už lze psát příkazy. Když se píše na klávesnici, zobrazují se alfanumerické znaky na terminálu. Funguje zde mazání pomocí klávesy BACKSPACE. Jednotlivé příkazy jsou v tabulce (Tab. 13). Při přepnutí do jiného módu se na terminál vypíše „*end PWM modulace*“.

Tab. 12 Příkazy pro terminál

kol X	KOL X
rych Y	RYCH Y
help	HELP
ukaz	UKAZ
tab	TAB

Nastavení terminálu

Pro potřeby komunikace lze použít jakýkoliv terminál s možností připojení na COM port počítače. Aby vše fungovalo, je potřeba nastavit správný COM port, baud rate: 9600, data: 8b.

Výběr koleje

Kolej vybereme pomocí příkazu „*kol X*“ nebo „*KOL X*“. Kde za X dosadíme číslo koleje od 1 po 10. Pokud zadáme špatné číslo, jako třeba 11, tak se předešlá hodnota nezmění. Pokud chceme nastavit rychlost na všech kolejí zároveň, použijeme příkaz „*kol 0*“ nebo „*KOL 0*“.

Výběr rychlosti

Tento příkaz funguje teprve až po vybrání koleje. Rychlost určujeme příkazy „*rych Y*“ či „*RYCH Y*“. Za Y dosadíme převodní číslo podle Tab. 6. Můžeme tedy zadat 0 až 15.

Další příkazy

Pokud na terminál napíšeme „*help*“ nebo „*HELP*“, tak se zobrazí nápověda, jak ukazuje Obr. 7. Příkaz „*ukaz*“ nebo „*UKAZ*“ zobrazí tabulku, jaké rychlosti jsou zvoleny na kolejích. A posledním možným příkazem je „*tab*“ nebo „*TAB*“. Tento příkaz zobrazí převodní tabulku.

```
PHM modulace > help
+-----+
| Napoveda pro SCI komunikaci s PHM modulatorem |
+-----+

KOL X = vyber koleje
RYCH Y = zvoleni rychlosti, musi byt vybrana kolej
TAB    = vypise tabulku rychlosti
UKAZ   = vypise rychlosti na kolejich
X nahrazuj cislem koleje, Y nahrazuj rychlosti dle tabulky
Mazat lze pomoci klavesy BACKSPACE

PHM modulace > █
```

Obr. 24 Ukázka nápovědy na terminálu

5.7.3 Manuální mód

Základní obrazovka manuál módu vypadá, jak je ukázáno na obrázku (Obr. 26). Pohybovat se v něm dá po řádcích pomocí tlačítek nahoru a dolů, podle obrázku (Obr. 27). Struktura ovládání je na obrázku (Obr. 24). Na obrázku (Obr. 27) je ukázka, jak může vypadat obrazovka při manuální změně rychlosti.



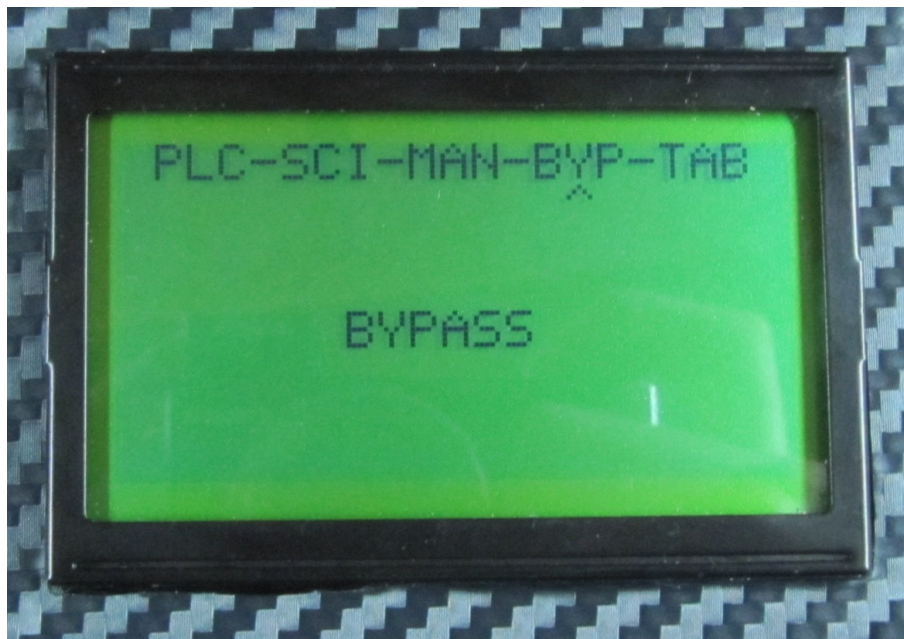
Obr. 25 Manuální mód



Obr. 26 Ukázka změny rychlosti

5.7.4 BYPASS mód

Jak již název napovídá, jedná se o přímé vedení signálu ze vstupu na výstup. Tedy jakýkoliv signál z PLC je poslán přímo na výstup do kolejiště. Díky tomuto je možné provádět PWM modulaci také na PLC. Display v tomto režimu vypadá podle obrázku (Obr. 28).



Obr. 27 Ukázka obrazovky při bypass modu.

5.7.5 Tabulkový mód

Tato položka zobrazuje stejnou převodní tabulku, jaká je v tabulce (Tab. 14).

5.8 PWM modulace

Modulování signálu bylo řešeno jako přerušení od časovače. K přerušení dojde každých 0.238ms. Prescaler nastavuje vstupní frekvenční děličku, f_{source} je frekvence zdroje hodinového signálu, t_{TOF} udává požadovaný čas do přetečení časovače a MODULO vyjadřuje údaj, který zapíšeme do modulo registru. Jedná se o 6b modulaci. Dá se tedy použít $2^6 = 64$ hodnot. Ovládání je možné pomocí PLC, SCI a manuálně tlačítky. Proto má každá možnost své pole typu CHAR, ve kterém je uložen údaj o požadované rychlosti. Díky tomu se při přepnutí do jiného módu ovládání pamatuje předešlý stav. V těchto polích je uložen číselný údaj, který se pak převádí na skutečnou procentní modulaci pomocí dalšího pole. Převod je v tabulce (Tab. 14).

Tab. 13 Převodní tabulka

Tabulková hodnota	Střída [%]	Tabulková hodnota	Střída [%]
0	0	8	67
1	23	9	72
2	31	10	77
3	38	11	81
4	44	12	86
5	50	13	91
6	56	14	95
7	63	15	100

5.9 Desky plošných spojů

Deska plošných spojů byla navržena v programu EAGLE Light. Při práci ve verzi Light jsou jistá omezení, jako je maximální velikost desky 10x8 cm. Proto bylo rozhodnuto, že budou udělané dvě desky a spojí se žebříkovými propojkami.

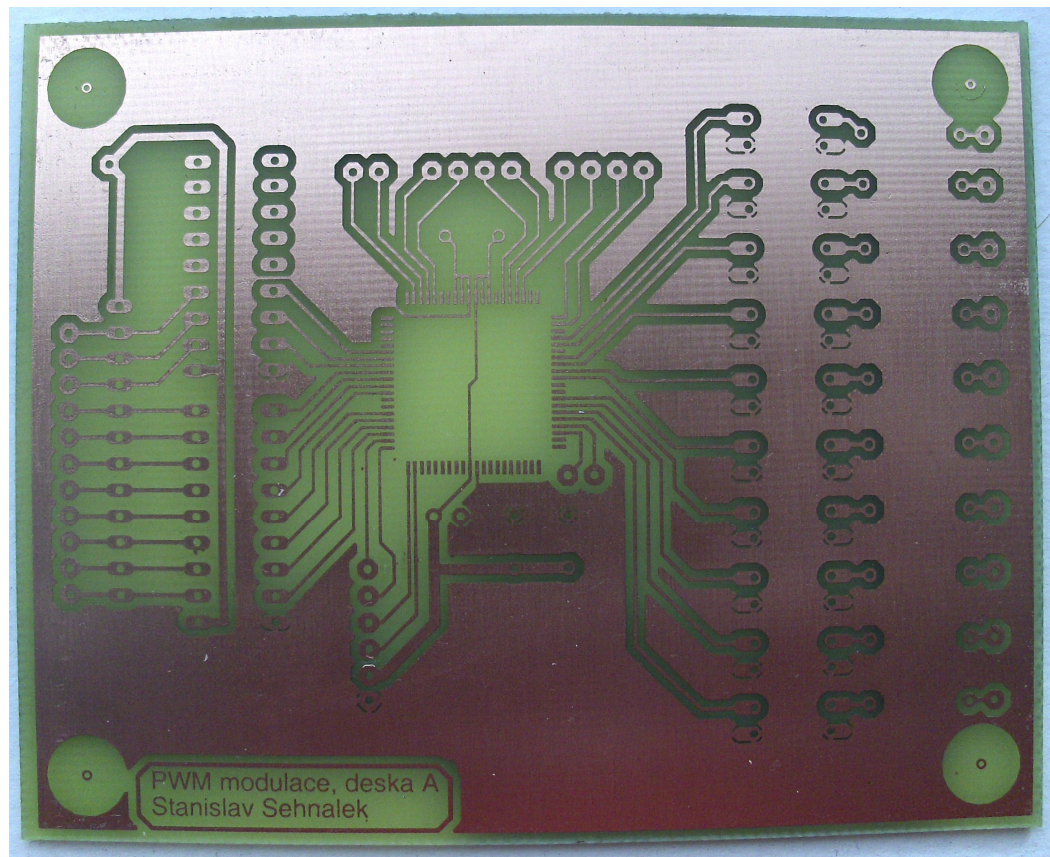
5.9.1 Výroba desky plošných spojů

Při výrobě DPS je nutné postupovat s rozvahou, případná chyba ve výrobním procesu by mohla znamenat zničení desky a tedy nutnost celý proces opakovat. Jako první je potřeba nachystat předlohu vytištěním na pauzovací papír, folii popřípadě kancelářský papír. Protože je potřeba, aby vytištěná předloha byla přesná a s ostrými okraji je vhodné k vytištění použít laserovou tiskárnu. Také je důležité před vytištěním zkontrolovat, že se bude tisknout zrcadlově obrácená předloha.

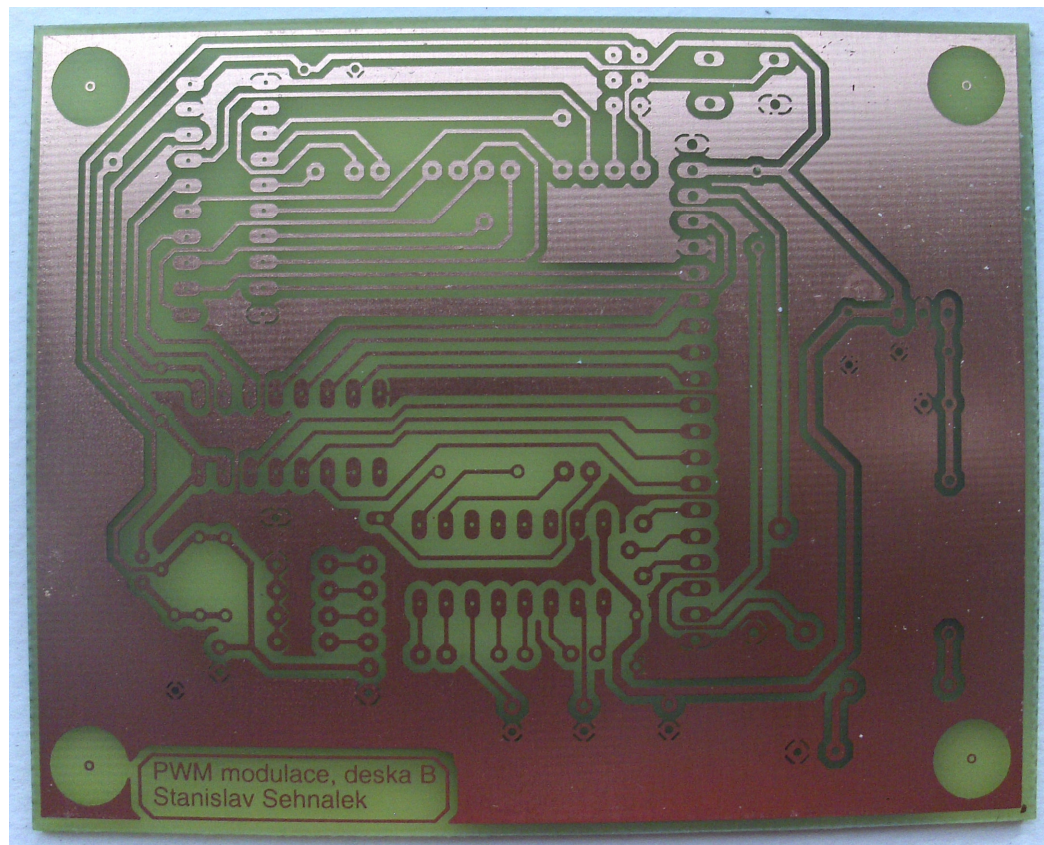
Jako druhý krok je příprava desky pro osvit, respektive žehlení, v závislosti, která metoda se použije. Nejvhodnější je zakoupit Cuprexit s již nanesenou fotocitlivou vrstvou. Před osvitem je potřeba ořezat desku na požadované rozměry, mezi tím můžeme zapnout osvitovou jednotku. K osvětlení se dá deska s předem vytištěnou předlohou a mělo by se zkontrolovat, zda je předloha správně otočená.

Po úspěšném osvícení se deska omyje čistou vodou. Po omytí se namáčí do nádoby s vývojkou, která smyje osvícený lak. Na desce zůstane jen neosvícený lak. Následně jde deska do leptacího roztoku, kde dojde k odstranění mědi na místech, kde není lak. Tento proces trvá kolem 15 min v závislosti na kvalitě leptacího roztoku.

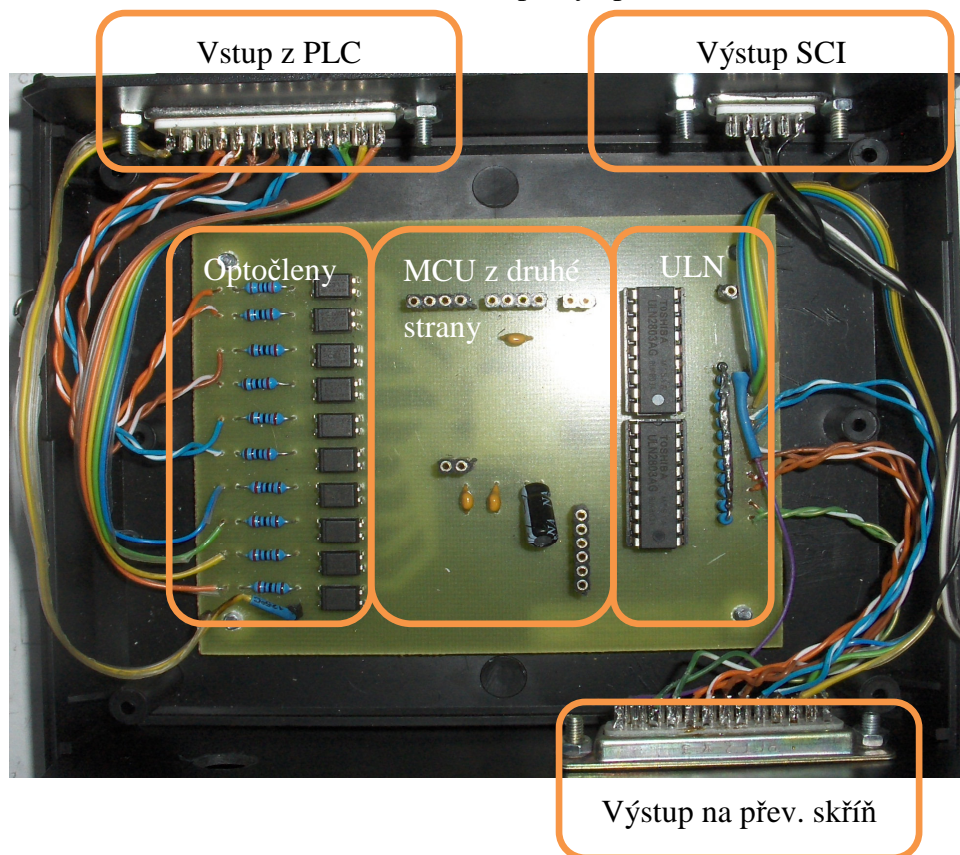
Ve finální fázi odstraníme zbytky roztoku pomocí ředidla. Také je vhodné zkontrolovat kvalitu spojů a případně odstranit nedokonale odleptanou měď. Pro ochranu spojů před korozí potřeeme desku tekutou kalafunou. Před osazením je také potřeba vyvrtat patřičné díry na místa kde budou součástky.



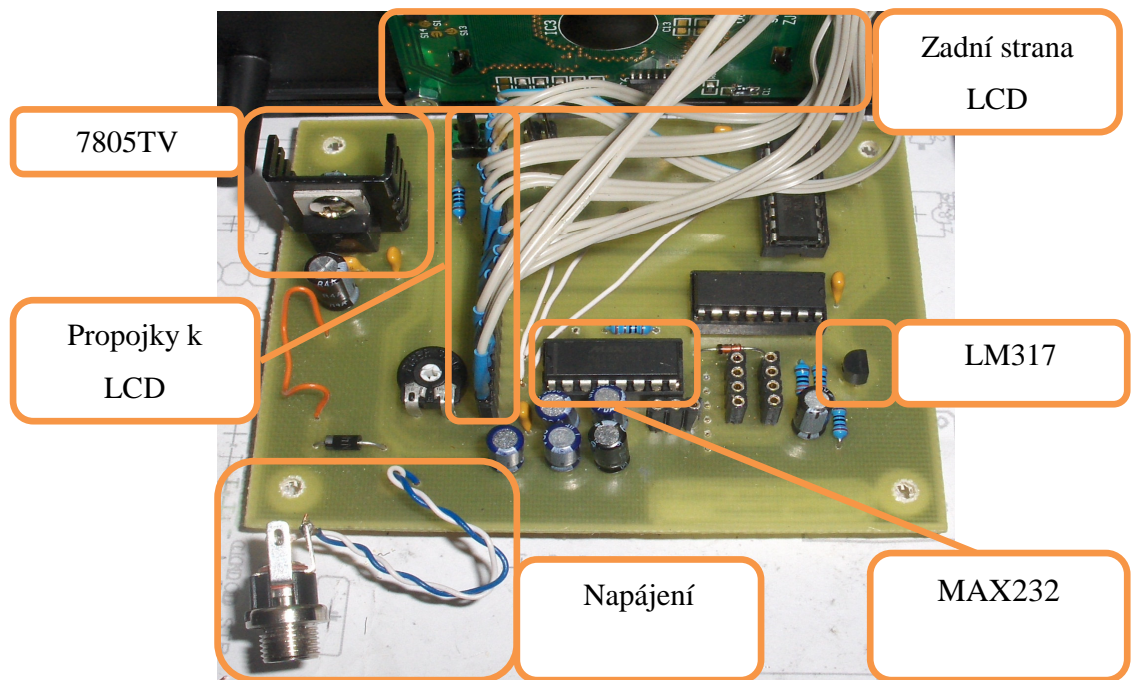
Obr. 28 Deska A po vyleptání



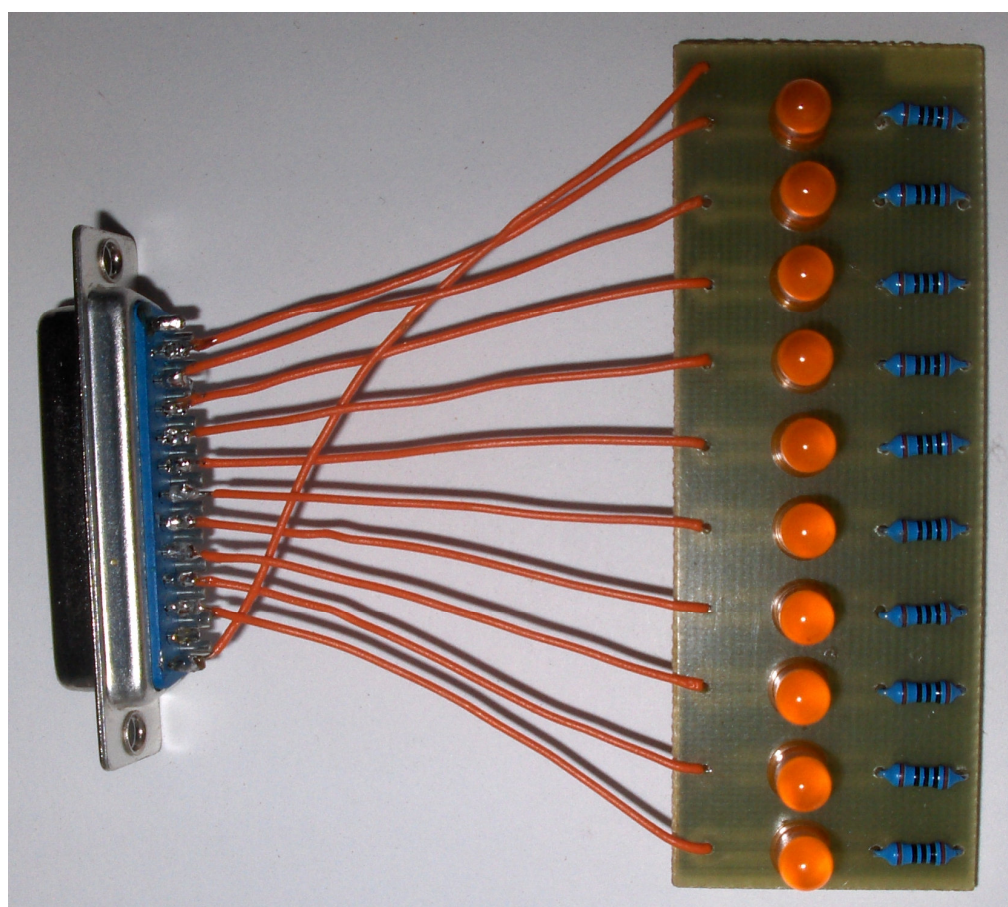
Obr. 29 Deska B po vyleptání



Obr. 30 Osázená deska A



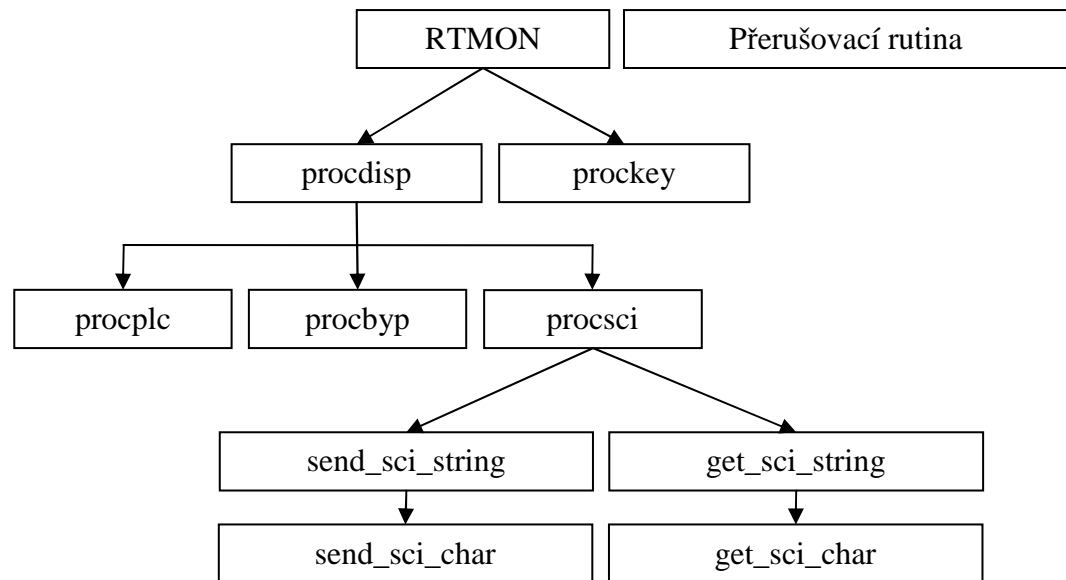
Obr. 31 Osázená deska B



Obr. 32 Testovací deska C s konektorem

6 PROGRAMOVÉ VYBAVENÍ

Program obsluhující MCU je realizován pomocí SOFT real-time systému RTMON, kde procesy odpovídají funkcím jazyka C. Tyto procesy mají svou prioritu a jsou volány buď periodicky, nebo se volají vždy jen jednou, v souladu s potřebou.



Obr. 33 Schéma funkcí modulu

6.1 Přerušovací rutina

Tato funkce se stará o PWM modulaci. Jedná se o klasickou přerušovací rutinu MCU volanou každých 2,3ms.

První se kontroluje, zda je aktivní mód, ve kterém má kolejiště fungovat. Následně kontroluje, zda je počítadlo rovno 64 přerušením. Pokud ano, tak překopíruje obsah pole s rychlostmi do pomocného pole. Zároveň se nuluje počítadlo přerušení.

Poslední částí této rutiny je uvádět jednotlivé piny, odpovídající úsekům kolejiště, do log. 1 nebo log. 0 v závislosti na požadované modulaci.

6.2 procplc()

Jedná se o RTMON proces, který odpovídá PLC módu. Je spouštěn $10 \cdot \text{tik}$, tedy každých 30ms. Jako první přečte obsah registru PTG, který rotuje o 4 bity doprava. Tento údaj se uloží do pomocné proměnné a určuje kolej. Následně se přečte znovu PTG, ale tentokrát pomocí funkce AND a bitové masky %00001111 se vyberou jen poslední 4 bity. Tyto bity se uloží do pomocné proměnné rychlost a odpovídají požadované rychlosti na úseku

kolejiště. Pak už jen následuje vyhodnocení, které buď nastaví rychlost na vybranou kolej, nebo pokud byla vybrána kolej 0, tak se na všechny koleje nastaví stejná rychlost.

6.3 procbyb()

Tento proces slouží pro BYPPAS mód. Hlavním úkolem zde je posílat vstup na výstup.

6.4 procsci()

V tomto procesu je řešena sériová komunikace pomocí COM portu. Z tohoto důvodu není volána periodicky, ale vždy jen jednou, pokud je vybrán SCI mód. Jako první provede výpis „*PWM modulace >* „ na naslouchající terminál pomocí funkce `send_sci_string()`. Tato funkce zajišťuje vypisování na sériovou linku sekvencí znaků. Poté se volá funkce `get_sci_string()`, která vrací zadané znaky. O této funkci se podrobněji píše v kapitole 6.8. Kód následně porovnává přijaté znaky s možnostmi, nebo vypíše, že byla zadána špatná sekvence znaků.

Podle sekvence znaků následují tyto možnosti kódu:

HELP

Na uživatelský terminál se vypíše nápověda pomocí již zmiňované funkce `send_sci_string()`. A opět sekvence „*PWM modulace >* „.

KOL

Tato část kódu načte zadanou hodnotu a uloží ji do pomocné proměnné `SCI_kol`. Pokud uživatel zadal neplatnou hodnotu koleje, vypíše se varování

RYCH

Provede uložení číselné hodnoty do pomocné proměnné `SCI_rych`. Jestli uživatel zadal neplatnou hodnotu rychlosti, vypíše se upozornění.

Když byla hodnota správná, tak se pomocí pomocných proměnných `SCI_kol` a `SCI_rych` nastaví příslušná rychlost na úsek kolejiště.

TAB

Tato sekvence slouží k vypsání převodní tabulky rychlostí podle tabulky (Tab. 14). Za vypsanou tabulkou následuje odeslání sekvence „*PWM modulace >* „ na terminál.

UKAZ

Toto je poslední z možností, kterou může uživatel zadat. Vypisuje tabulku, jaké rychlosti jsou na jednotlivých úsecích kolejiště.

6.5 procdisp()

Tento proces RTMONu slouží jako obsluha LCD displeje. V závislosti jaký mód je vybrán, zobrazuje tabulky nastavených rychlostí na kolejišti. Pokud je vybrán BYPPAS mód, tak jen vypisuje informaci o tom, že je vybrán. U TAB módu se vypisuje převodní tabulka.

K výpisu na displej slouží funkce DISP_STRING(), která je řešena assemblerem.

6.6 prockey()

K obsluze tlačítek slouží tento periodicky volaný proces. Při každém průchodu se kontroluje, zda bylo zmáčknuto tlačítko. Pokud ano, tak v závislosti na tom, který se provede konkrétní kód.

Kód níže provádí výběr módu, proto se počítadlo cyklicky inkrementuje po pěti položkách.

```
1 if(tla4==0 && radek == 0)// pokud je zmačknuto tlačítko tla4
2 {
3   vyber++;                // inkrementuje se počítadlo
4   if(vyber == 6)         // pokud počítadlo dosáhne stanovené výše
5     vyber=1;             // tak se natavuje na 1
6   zmena[vyber] = 1;     // provádí se přepnutí módu
7   zmena[0]=0;          // nastaví se příznak změny
8   rtm_delay_p(pk,20);   // čekací proces RTMON k ošetření zákmitů
9 }
```

6.7 Funkce main()

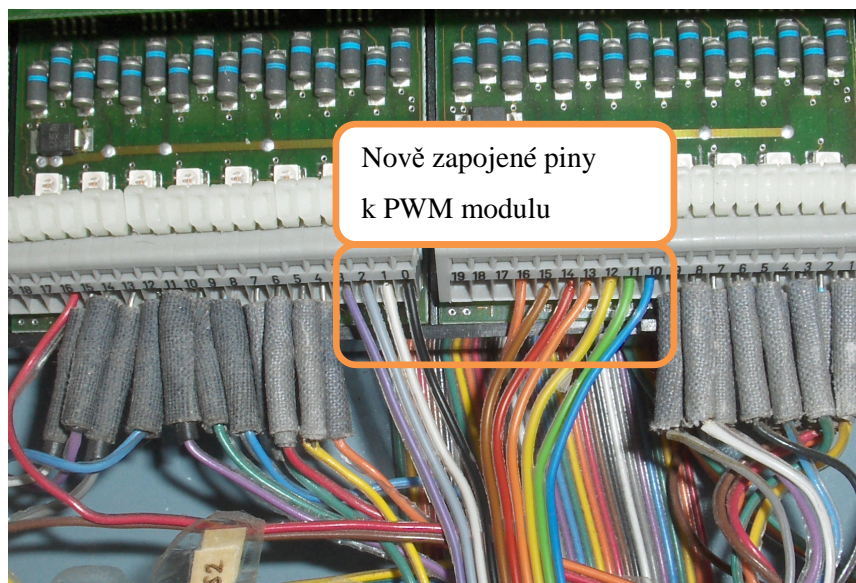
Hlavní funkce, ve které dochází k nastavení potřebných registrů a RTMON. Jako první se nastavuje hodinový generátor pro správnou frekvenci. Následuje nastavení časovače TMP2 pro modulaci. Jednotlivé porty se dále nastavují takto: PTD, PTF, PTJ jako vystup a u PTB, PTC, PTE, PTG se zapínají pull-up. Před inicializací RTMON se nastavuje baud rate na 52 a registry pro SCI komunikaci.

6.8 Čtení z SCI

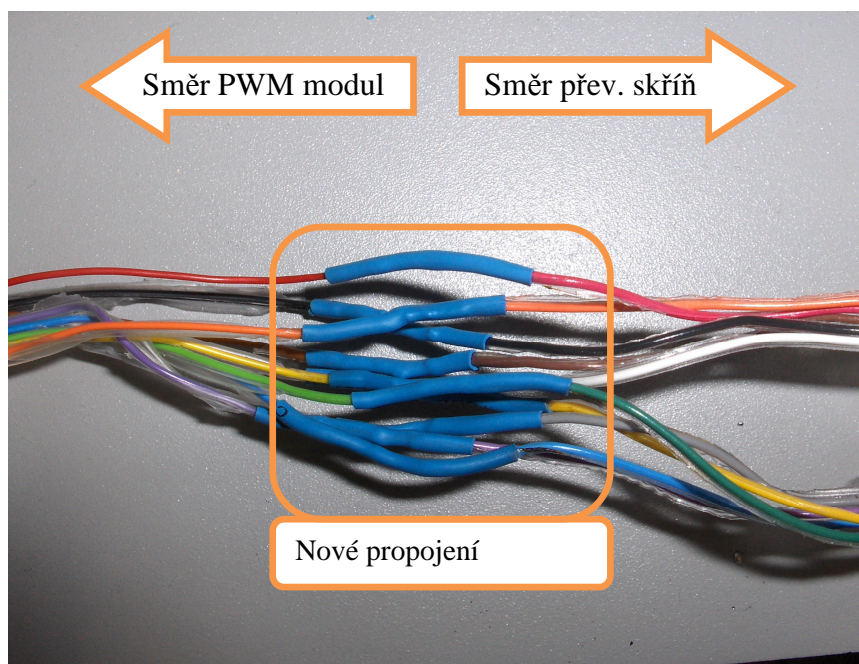
Funkce je nazvána `get_sci_string(char *string)`. Jedná se o funkci, která zajišťuje čtení znaků ze sériového rozhraní. Jakmile je zavolána, tak periodicky čte znaky pomocí `get_sci_char()`, dokud není načten znak „\N“, což odpovídá klávese ENTER, nebo není načteno 30 znaků. Je zde i řešeno mazání znaků pomocí klávesy BACKSPACE.

7 PRAKTICKÉ OVĚŘENÍ

Aby bylo možno integrovat modul do stávajícího řešení, bylo potřeba provést několik úprav zapojení PLC a převodníkové skříně. Musely se odpojit piny převodníkové skříně od PLC a připojit nové piny, jak ukazuje obrázek (Obr. 34). Také bylo potřeba propojit staré piny, které byly napojeny na PLC s piny od konektoru modulu. Toto propojení je znázorněno na obrázku (Obr. 35).

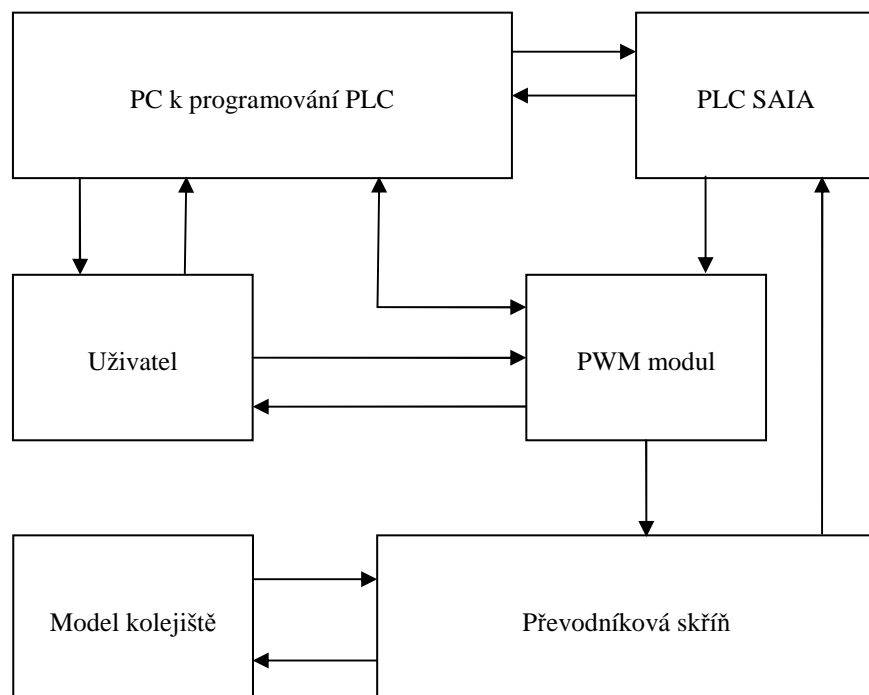


Obr. 34 Nové zapojení pinů k modulu



Obr. 35 Propojení pinů z PLC na převodníkovou skříně

Na obrázku (Obr. 36) je blokové schéma znázorňující propojení PWM modulu s kolejištěm. Jak znázorňuje šipka vedoucí z bloku - uživatel může modul ovládat ručně a zpětná šipka představuje zobrazení rychlostí na LCD displeji. Blok PC je s modulem spojen sériovou linkou, kde na základě příkazů probíhá obousměrná komunikace. Blok PLC je s modulem spojen 10 pinovým kabelem, což umožňuje volit rychlosti na kolejích nebo v BYPASS módu přímým ovládním rychlostí na úsecích kolejiště. PWM modul je s převodníkovou skříní spojen také 10 pinovým kabelem, kde každý pin znamená PWM modulaci pro jednotlivý úsek.



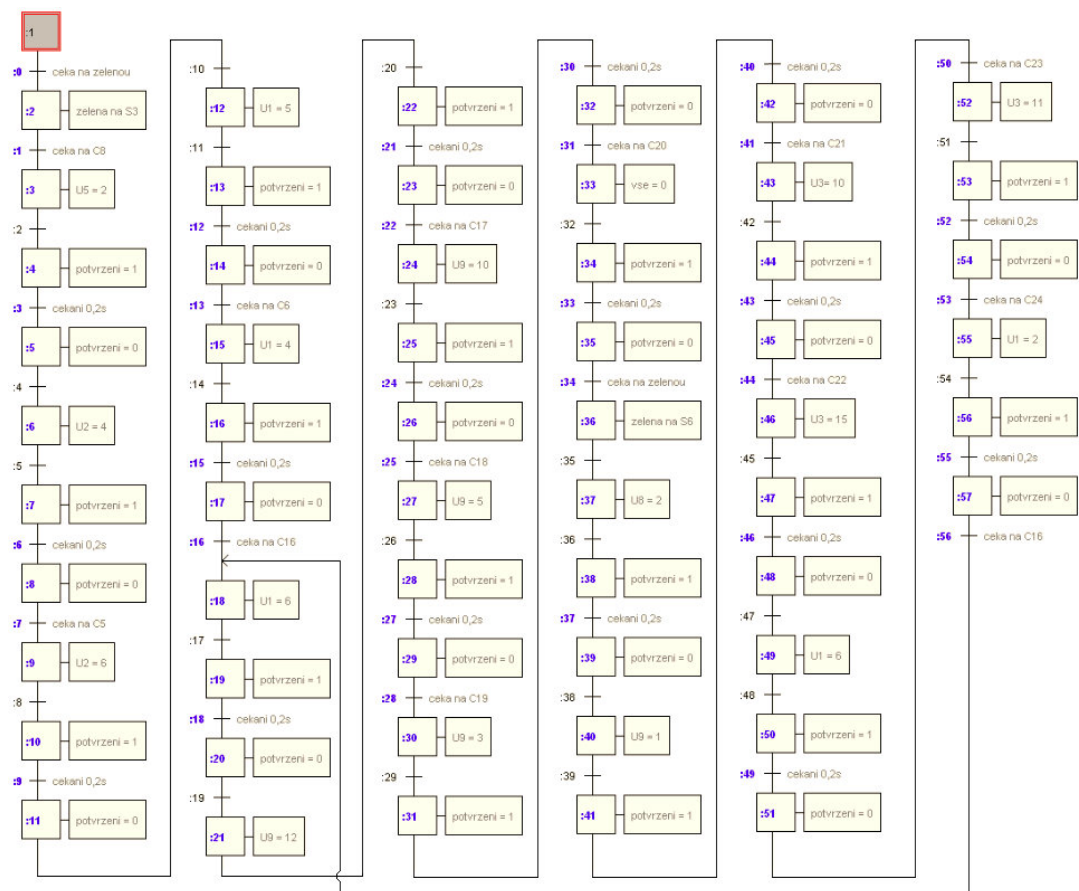
Obr. 36 Blokové schéma zapojení kolejiště s modulem

7.1 Ukázkový program

Pro názornou ukázkou funkčnosti zapojení PLC s PWM modulem a převodníkovou skříní je napsán řídicí program do SAIA PCD2. Je napsán v Graftek editoru pomocí FUP bloků. Celá sekvence bloků je znázorněna na obrázku (Obr. 36).

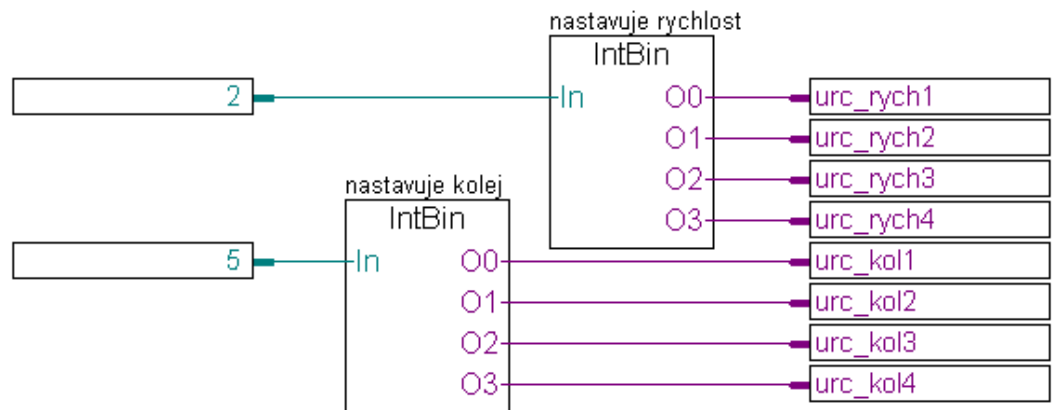
Vlak stojí na čidle C8, kde čeká na zelenou na semaforu S3. Po naskočení zelené se vlak rozjede rychlostí 2, což odpovídá 31% modulaci. Po najetí na úsek dva jede rychlostí 4, která odpovídá 44% modulace. Po přejetí čidla C5 se rychlost změní na 6, než najede na úsek 1, kde je rychlost 5, jenž odpovídá 50% modulace. Při přejetí přes čidlo C6 se rychlost mění na 4. Jakmile přejede čidlo C16, přehodí se výhybka, aby po objetí kolejiště pokračoval znovu dokola. Také se mění rychlost na 6, než najede na úsek 9, kde je rychlost

12, která odpovídá 86% modulaci. Při přejetí čidla C17 se nastavuje rychlost 10. Po projetí přes čidlo C18 se rychlost snižuje na 5 a při průjezdu čidla C19 je rychlost 3. Toto zpomalení je důležité, protože na čidle C20 dochází k zastavení a čeká se na zelenou, která naskočí zhruba po 4s. Následně se rozjede rychlostí 1 a po najetí na úsek 8 jede rychlostí 2, až po najetí na úsek 3, kde se rychlost mění na 10. Na čidle C22 se zrychluje na maximální rychlost - tedy 15, což je 100% modulace. Přejetím přes čidlo C23 se nastavuje rychlost 11 až do okamžiku, než najede lokomotiva na kolej 1, kde je rychlost 6. Po najetí na čidlo 24 se nastavuje rychlost 2 a celá sekvence se opakuje od čidla C16.



Obr. 37 Posloupnost bloků ukázkového programu

FUP blok pro nastavení rychlosti je znázorněn na obrázku (Obr. 38). Jedná se o dva bloky, které převádí integer číslo na binární. V tomto případě se nastavuje úsek 5 na rychlost 2.



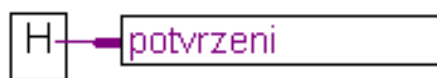
Obr. 38 FUP blok nastavení rychlosti

Potvrzení přejetí přes určité čidlo slouží obrázek (Obr. 39). Jelikož jsou čidla inverzní, je také potřeba negovat vstup na blok ETR (End Of Transition).



Obr. 39 Potvrzení přejetí přes čidlo

Pro potvrzení je potřeba poslat na pin POTVRZENI tedy O27 log. 1. Toho je docíleno dle obrázku (Obr. 40). Následně před vysláním log. 0 na O27 je potřeba počkat 0,2s. Tento čekací blok je znázorněn na obrázku (Obr. 41). Jde o Graftec blok Timer, který se přepne do log. 1 po uběhnutí vstupu krát 1/10s.



Obr. 40 Log. 1 na O27



Obr. 41 Čekací blok

ZÁVĚR

Hlavní cílem této bakalářské práce bylo navrhnout a vytvořit mikropočítačový modul pro řízení rychlosti vlaků na modelu kolejiště, které slouží pro podporu výuky předmětu „Programovatelné automaty“. Základním požadavkem na modul byla možnost jeho ovládání pomocí programovatelného automatu SAIA prostřednictvím standardních digitálních výstupů nebo sériovou linkou a v případě potřeby i manuálně bez připojeného PLC.

Bakalářská práce řeší hardwarový návrh desky plošných spojů a následné programové vybavení modulátoru. Úvodní kapitola se zabývá teoretickou částí, kde je popsán výběr mikrokontroléru a jeho následný popis. Následuje popis jeho centrální procesní jednotky, portů, vnitřních registrů a komunikačního rozhraní BDM. Dále jsou zde vysvětleny základní principy a přednosti PWM modulace, principy real-timových systémů a procesy RTMON. V poslední kapitole teoretické části je uveden popis modelu vlakového kolejiště před provedením úprav a zapojením PWM modulátoru.

Druhá část, tedy praktická, se zabývá konstrukcí modulátoru a jeho programovým vybavením. Tato část obsahuje dvě hlavní kapitoly, v první z nich jsou řešeny hardwarové aspekty konstrukce. Mezi ně patří realizace zapojení mikrokontroléru a jeho periferních obvodů. Dále pak zapojení LCD displeje a jeho propojení s mikropočítačem, řešení napájení a propojení s PLC, respektive s převodníkovou skříní kolejiště. Poslední kapitola řeší popis obslužného programu, který je zapotřebí k funkci celého modulu.

Je vytvořen prototyp modulátoru a následně ověřena jeho funkčnost s kolejištěm. Rovněž je napsán zkušební program pro PLC SAIA, který demonstruje funkčnost zapojení a modulu.

Díličí výsledky práce byly prezentovány na studentské soutěži FREESCALE TECHNOLOGY APPLICATION 2010 ve spolupráci s UTB ve Zlíně, kterou pořádala pobočka firmy Freescale v Rožnově pod Radhoštěm. Prezentovaný modul se umístil na 4 místě, což dokládá jeho funkčnost a spolehlivost.

ZÁVĚR V ANGLIČTINĚ

The main objective of this bachelor thesis was to design and create a pulse width modulator for teaching programmable logical controllers. So it can be controlled by PLC SAIA, a serial line of PC and manually.

This work deals with the hardware design of circuit board and subsequent modulator software. Introductory chapter deals with the theoretical part, which describes the selection of a microcontroller and its subsequent description. Furthermore, a description of its central process units, ports, registers and internal communication interface BDM. Then, there are explained basic principles and advantages of the PWM modulation. Describes the principles of real-time systems and processes RTMON. The last theoretical chapter is a description of the model train track before the modifications and the involvement of the PWM modulator.

The second part, a practical, deals with the construction of the modulator and its software. This section contains two main chapters, the first of these aspects are deal with hardware design. This includes the implementation of the involvement of the microcontroller and its peripheral circuits. Furthermore, the involvement of the LCD and its connection with the microprocessor, networking and power solutions with PLC, respectively transition box. The last chapter deals with the description of a utility, which was required for the function module.

Prototype modulator was developed and subsequently reviewed its operation with the railway model. Also, the test program was written for the PLC SAIA, which demonstrates the module functionality and integration.

This module competed in FREESCALE TECHNOLOGY APPLICATION 2010, hosted by a subsidiary of Freescale in Roznov pod Radhostem. At this competition, he placed the 4 place, which demonstrates the functionality and reliability of this module.

SEZNAM POUŽITÉ LITERATURY

- [1] BURKHARD, Mann. C pro mikrokontroléry. Praha: BEN - technická literatura, 2003. 280 s. ISBN 80-7300-077-6.
- [2] Freescale Semiconductor. CPU08 Central Processor Unit Reference Manual., 2001. Dostupný z WWW: <http://www.freescale.com>.
- [3] Freescale Semiconductor. HCS08 Family Reference Manual, Rev.1., 2003. Dostupný z WWW: <http://www.freescale.com>.
- [4] Freescale Semiconductor. MC9S08QE128 Reference Manual, Rev.2., 2007. Dostupný z WWW: <http://www.freescale.com>.
- [5] HNILICA, Vladimír. Učební pomůcka pro výuku, vlakové kolejiště. Zlín, 1999. 64 s. Vedoucí bakalářské práce Prof. Ing. Vladimír Vašek, CSc.
- [6] HNILICA, Vladimír. Počítačové řízení vlakového kolejiště. Zlín, 2002. 68 s. Vedoucí diplomové práce Prof. Ing. Vladimír Vašek, CSc.
- [7] HRABOVSKÝ, Miroslav, JURÁNEK, Antonín. EAGLE pro začátečníky: Návrhový systém pro plošné spoje. Praha: BEN - technická literatura, 2007. 192 s. ISBN 80-7300-213-2.
- [8] NOVÁK, Petr. Mobilní roboty - pohony, senzory, řízení. : 1. díl. Praha: BEN - technická literatura, 2005. 256 s. ISBN 80-7300-141-1.
- [9] VÁŇA V.: Začínáme s mikrokontroléry Motorola HC08 Nitron. Praha: BEN – technická literatura, 2003. 96 s. ISBN 80-7300-124-1.
- [10] SEDLÁK, Jiří. Výukový modul pro předmět mikropočítače: grafický LCD displej. Zlín, 2008. 65 s. Vedoucí bakalářské práce Ing. Petr Dostálek.
- [11] Juvva, Kanaka. Dependable Embedded Systems. Získáno 18.5.2010, z Real-Time Systems: http://www.ece.cmu.edu/~koopman/des_s99/real_time/

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ALU	Arithmetic Logic Unit (aritmeticko logická jednotka)
BDC	Background Debug Controller (ladění a monitorování chodu programu)
BDM	Background Debug Mode (ladící a monitorovací mód)
COP	Computer Operating Properly (obvod pro hlídání správného běhu programu)
CPU	Central Procesing Unit (centrální procesní jednotka)
Flash	nevolatilní elektricky programovatelná paměť s libovolným přístupem
I/O	Input/Output (vstupně/výstupní)
LCD	Liquid crystal display (displej z tekutých krystalů)
MCU	Microcontroller Unit (mikrokontrolér)
MSB	Most Significant Bite (bit s nejvyšší hodnotou v binárním vyjádření čísla)
OS	Operating System (operační systém)
PLC	Programable Logical Controler (programovatelný automat)
PWM	Pulse-wave modulation (pulzní šířková modulace)
RAM	Random-Access Memory (paměť s libovolným přístupem)
RTS	Real-time system (real-timový systém)
RTOS	Real-time operating system (real-time operační systém)
SCI	Seriál Comunication Interface (sériové komunikační rozhraní)
SPI	Seriál Perifeal Interface (sériové periferní rozhraní)
TPM	Časový modul
TTL	Transistor-Transistor-Logic (tranzistorově-tranzistorová logika)
USB	Universal serial bus (univerzální sériové rozhraní)

SEZNAM OBRÁZKŮ

Obr. 1 Význam pinů u MC9S08QE128CLK.....	12
Obr. 2 Vnitřní registry HCS08.....	13
Obr. 3 Vnitřní struktura registru CCR	16
Obr. 4 Zapojení BDC s MC9S08QE128	17
Obr. 5 Open Source BDM	18
Obr. 6 RTS plánování	22
Obr. 7 Blokové schéma současného stavu.....	26
Obr. 8 PLC SAISA PCD2.....	27
Obr. 9 Okno editoru FUPla.....	28
Obr. 10 Okno editoru Graftec	29
Obr. 11 Model kolejiště pohled shora.....	29
Obr. 12 Boční pohled na kolejiště s vyznačenými úseky	31
Obr. 13 Blokové schéma modulu	33
Obr. 14 QE128 v 80 pinovém pouzdře	33
Obr. 15 Schéma zapojení LCD	35
Obr. 16 Schéma 5V napájení	35
Obr. 17 Schéma 3.3V napájení a BDM	36
Obr. 18 Zapojení optočlenů na vstupu modulátoru	37
Obr. 19 Schéma zapojení výstupů	38
Obr. 20 Schéma testovací desky PWM	39
Obr. 21 Zapojení MAX232 pro sériovou komunikaci.....	40
Obr. 22 Význam tlačítek	40
Obr. 23 Diagram ovládní	41
Obr. 24 Ukázka nápovědy na terminálu	43
Obr. 25 Manuální mód.....	44
Obr. 26 Ukázka změny rychlosti	44
Obr. 27 Ukázka obrazovky při bypass modu.....	45
Obr. 28 Deska A po vyleptání	47
Obr. 29 Deska B po vyleptání.....	48
Obr. 30 Osázená deska A.....	48
Obr. 31 Osázená deska B	49
Obr. 32 Testovací deska C s konektorem	49

Obr. 33 Schéma funkcí modulu	50
Obr. 34 Nové zapojení pinů k modulu.....	54
Obr. 35 Propojení pinů z PLC na převodníkovou skříň	54
Obr. 36 Blokové schéma zapojení kolejiště s modulem.....	55
Obr. 37 Posloupnost bloků ukázkového programu.....	56
Obr. 38 FUP blok nastavení rychlosti.....	57
Obr. 39 Potvrzení přejetí přes čidlo	57
Obr. 40 Log. 1 na O27	57
Obr. 41 Čekací blok	57

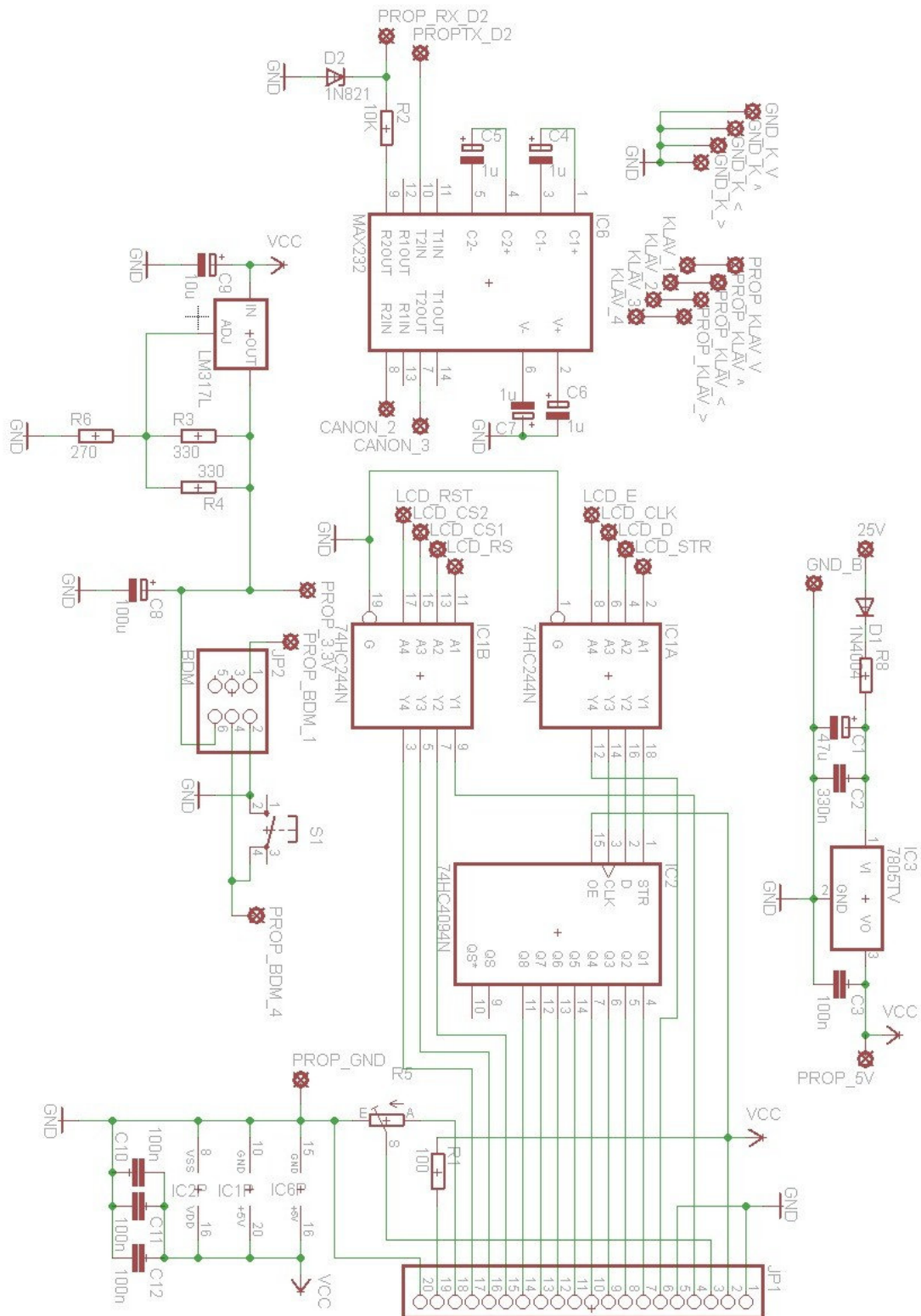
SEZNAM TABULEK

Tab. 1 Adresy čidel	30
Tab. 2 Adresy semaforů	30
Tab. 3 Adresy výhybek	30
Tab. 4 Adresy přejezdů	30
Tab. 5 Adresy osvětlení	30
Tab. 6 Význam pinů.....	34
Tab. 7 Propojení PCL s MCU.....	36
Tab. 8 Význam pinů na vstupu CANNON 25F.....	37
Tab. 9 Propojení MCU s kolejištěm	38
Tab. 10 Význam pinů na výstupu	39
Tab. 11 Příklad.....	42
Tab. 12 Příkazy pro terminál	42
Tab. 13 Převodní tabulka	46

SEZNAM PŘÍLOH

- P I Schéma zapojení řídicí jednotky
- P II Schéma zapojení modulu displeje
- P III Osazovací plán desek A, B
- P IV Podklad pro výrobu desek A, B v měřítku 1:1

PŘÍLOHA P II: SCHÉMA ZAPOJENÍ MODULU DISPLEJE



PŘÍLOHA P IV: PODKLAD PRO VÝROBU DESEK A, B V MĚŘÍTKU

1:1

