

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



BAKALÁŘSKÁ PRÁCE

Řízení modelu železnice

Praha, 2007

Autor: Jiří Kubias

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne _____

podpis

Poděkování

Rád bych na tomto místě poděkoval vedoucímu své práce, Ing. Dušanu Havlíkovi, za vstřícný přístup a vedení v průběhu vzniku této práce. Dále děkuji Ing. Radku Šindelářovi za konzultace, které vedli ke úspěšné přestavbě modelu železnice. V neposlední řadě děkuji rodině za podporu po celou dobu studia.

Abstrakt

Cílem této bakalářské práce je návrh a realizace nového řídicího systému stávajícího modelu železnice, který bude vyhovovat zvýšeným nárokům na řízení modelu z nadřazeného systému. Nová koncepce řídicího systému je založena na jednodušší řídicí struktuře elektroniky, jiném procesoru řídicí jednotky a změněném komunikačním protokolu mezi řídicí jednotkou a nadřazenou aplikací. Jádrem řídicí jednotky je procesor LPC2106 typu ARM7. Komunikace mezi řídicí jednotkou a periferiemi modelu probíhá přes komunikační síť Lo-coNet, což umožnilo zjednodušení struktury řídicí elektroniky. Mezi snímané periferie byly nově zařazeny i přestavníky výhybek. Řízení lokomotiv pomocí DCC standardu bylo ponecháno.

Vzhledem ke změně komunikačního protokolu a změně procesoru, byl vytvořen úplně nový program pro procesor LPC2106 v jazyce C. Pro testování jednotlivých částí modelu byla dále vytvořena aplikace v jazyce C# .NET.

Abstract

Aim of this diploma thesis is design and implementation of new control system for existing railway model, which will meet higher demands for the control of the model from superior system. New control system concept is based upon simpler structure of controlling electronics, different processor in the controlling unit and modified communication protocol between the controlling unit and superior application. Core of the controlling unit is processor LPC2106 from ARM7 family. Communication between the controlling unit and model peripherals takes place across LocoNet communication network, which made the simplification of controlling electronics possible. Point-operating apparatus was newly listed among the controlled peripherals. Controlling of the locomotives using the DCC standard was kept.

Because of the change of communication protocol and replacement of the controlling processor, completely new program for the LPC2106 processor had to be created in C language. Further, for testing of the model was created an application in C# .NET programming language.

Katedra řídicí techniky

Školní rok: 2006/2007

Z a d á n í b a k a l á ř s k é p r á c e

Student: Jiří K u b i a s
Obor: Kybernetika a měření
Název tématu: Řízení modelu železnice

Z á s a d y p r o v y p r a c o v á n í :


1. Seznamte se s modelem železnice v laboratoři K909 na Katedře řídicí techniky.
2. Seznamte se se současnou nabídkou mikroprocesorů.
3. Navrhněte nový způsob řízení modelu železnice, který podstatně zvýší možnosti řízení modelu.
4. Navržený způsob řízení modelu implementujte na daném modelu.
5. Pro ověření funkčnosti vytvořte jednoduchý program ve Vámi zvoleném programovacím jazyku. V textu bakalářské práce pouze uveďte jeho ovládání.

Seznam odborné literatury: Dodá vedoucí práce


Vedoucí bakalářské práce: Ing. Dušan Havlík

Datum zadání bakalářské práce: zimní semestr 2006/07

Termín odevzdání bakalářské práce: 15. 8. 2007


Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




Prof. Ing. Zbyněk Škvor, CSc.
děkan

V Praze, dne 6. 3. 2007

Obsah

Seznam obrázků	ix
Seznam tabulek	xi
1 Úvod	1
2 Model železnice	3
2.1 Stav před rekonstrukcí modelu	3
2.2 Rekonstruovaný model	4
3 Obecné řešení zadaného problému	5
3.1 Řešení zadaného problému	5
3.2 LocoNet	6
3.2.1 Obecný popis	6
3.2.2 Fyzická vrstva	6
3.2.3 Linková vrstva	6
3.3 DCC	7
3.4 Obecný popis komunikace nadřazené aplikace s řídicí jednotkou	8
4 Řídicí elektronika	10
4.1 Konfigurace modelu železnice	10
4.2 Řídicí jednotka	11
4.2.1 Mikroprocesor ARM7 LPC2106	12
4.2.1.1 Stručný popis jádra ARM7	12
4.2.1.2 Vlastnosti mikroprocesoru	13
4.2.1.3 Programování mikroprocesoru	13
4.2.2 Modul MiniARM	14
4.2.3 DCC výstup	14

4.2.4	RS232	15
4.2.5	LocoNet	16
4.3	LocoIO	17
4.4	Snímač polohy výhybky	18
4.5	LocoBuffer	19
4.6	Snímače proudu	20
4.7	Reléová karta	21
4.8	Ovládací panel	21
5	Software	23
5.1	Nastavení mikroprocesoru	23
5.1.1	Přerušovací systém mikroprocesoru	23
5.1.2	Výstupní porty	25
5.1.3	PWM jednotka	26
5.1.4	Sériový kanál	27
5.1.5	Čítače a časovače	30
5.2	Popis programu	31
5.2.1	Popis inicializace mikroprocesoru	31
5.2.2	Popis hlavního programu	32
5.2.3	Obsluha přerušení PWM jednotky	33
5.2.4	Obsluha přerušení UART1 jednotky	33
6	Komunikační protokol	34
6.1	Popis protokolu	34
6.2	Reset jednotky	34
6.3	Průběh komunikace	35
6.4	Příkazy a příklady	36
7	Řídicí aplikace - Locomotives	47
8	Závěr	49
	Literatura	52
A	Schématická zapojení	I
B	Konfigurace modulu LocoIO	V

C	Seznam zkratek	IX
D	Obsah přiloženého CD	X
E	Nákres kolejíště	XI

Seznam obrázků

2.1	Fotografie modelu železnice	4
3.1	Zapojení konektoru pro LocoNet	6
3.2	DCC paket	7
3.3	Blokové schéma připojení dekodéru	8
4.1	Blokové schéma zapojení kolejiště.	11
4.2	Blokové schéma řídicí jednotky.	12
4.3	Blokové schéma jádra ARM7.	13
4.4	Modul MiniArm.	14
4.5	Schéma zapojení výstupu DCC.	15
4.6	Základní zapojení obvodu ICL232.	16
4.7	Schéma zapojení výstupu LocoNet.	17
4.8	Blokové schéma modulu LocoIO.	18
4.9	Blokové schéma snímače polohy výhybky.	19
4.10	Blokové schéma modulu snímače proudu.	20
4.11	Blokové schéma modulu reléové karty.	21
4.12	Ovládací panel modelu kolejiště.	22
5.1	Význam bitů v registrech UART.	29
6.1	Tvorba extend byte.	38
7.1	Aplikace Locomotives	48
A.1	Modul Řídicí jednotky	I
A.2	Modul LocoIO	II
A.3	Modul LocoBuffer	III
A.4	Modul Snímače optické oddělení	IV

B.1	Zapojení konektoru pro LocoNet	V
B.2	Zapojení konektoru pro LocoNet	VI
E.1	Nákres kolejíště	XI

Seznam tabulek

4.1	Tabulka elektrických vlastností modulu LocoIO	18
4.2	Význam LED diod ovládacího panelu	22
5.1	Význam registrů PWM jednotky	26
5.2	Význam registrů UART0 jednotky	28
6.1	Hodnoty seg pro 8mi bitový mód - zápis	41
6.2	Hodnoty seg pro 4 bitový mód - zápis	42
6.3	Hodnoty seg pro 1 bitový mód - zápis	43
6.4	Hodnoty seg pro 8mi bitový mód - čtení	44
6.5	Hodnoty seg pro 4 bitový mód - čtení	45

Kapitola 1

Úvod

Bakalářská práce se zabývá návrhem nového řídicího systému pro model železnice umístěného na katedře Řídicí techniky na Karlově náměstí v laboratoři K09. Stávající způsob řešení řízení modelu byl nevyhovující pro výuku a bylo nutné navrhnout nový řídicí systém.

První kapitola nás seznamuje s obsahem bakalářské práce s motivem, proč tato bakalářská práce vznikla.

Druhá kapitola se zabývá popisem starého modelu železnice, jeho nedostatků a definuje nové požadavky na řídicí systém modelu železnice.

V kapitole 3. je popsán způsob řešení bakalářské práce a jsou zde popsány jednotlivé použité standardy a normy použité pro řízení modelu železnice.

Kapitola 4. je věnována popisu řídicí elektroniky modelu železnice. Jsou zde uvedeny nové periférie modelu včetně nové řídicí jednotky a úpravy stávajících modulů železnice.

Kapitola 5. se zabývá nastavením mikroprocesoru řídicí jednotky a jeho programem. Kapitola je rozdělena do dvou částí. V první části jsou stručně popsány nastavní nastavení jednotlivých periférií mikroprocesoru včetně jejich základního popisu. V druhé části je popsán program mikroprocesoru s obslužnými podprogramy pro jednotlivé jeho periférie.

V kapitole 6. je popsán komunikační protokol mezi nadřazeným systémem a hlavní řídicí jednotkou modelu železnice. Je zde definován způsob vytváření příkazů a paketů, včetně jejich ukázek.

Kapitola 7. obsahuje popis ovládaní aplikace Locomotives, která je určena pro testování modelu železnice.

Kapitola 8. obsahuje zhodnocení dosažených výsledků a naznačuje možnosti dalšího budoucího rozšíření a zdokonalování modelu železnice.

V této bakalářské práci je předpokládána čtenářova základní znalost obecné architek-

tury mikroprocesorů a základní znalost programovacího jazyka C.

Kapitola 2

Model železnice

2.1 Stav před rekonstrukcí modelu

Model kolejistě se skládá ze tří vzájemně propojených částí: - kolejistě modelu železnice - senzorika a akční elektronika modelu - řídicí jednotka, která řídí model železnice dle pokynů nadřazené aplikace.

Samostatný model železnice obsahuje tři nádraží, 53 oddělených traťových úseků, 15 výhybek, 12 semaforů a 6 rozpojovačů vagonů, které jsou v současné době nefunkční. Senzorika a akční elektronika modelu je složena z 53 proudových snímačů (každý úsek kolejistě má přiřazený jeden proudový snímač), ovladačů semaforů, ovladačů výhybek a posilovače DCC signálu.

V původním stavu byla většina senzorů a akčních členů propojena pomocí paralelní sběrnice k řídicí jednotce. Semaforey byly spojeny s řídicí jednotkou pomocí I2C sběrnice. Vlastní pohyb lokomotivy byl řízen řídicí jednotkou s procesorem PIC16F877 prostřednictvím DCC standardu. Komunikace mezi nadřazenou aplikací a řídicí jednotkou byla zajištěna sériovou linkou RS232 s rychlostí přenosu 9600Bd. Vzhledem k velkému objemu přenášených dat a nízké komunikační rychlosti byla odezva řídicí jednotky vyšší než 300ms.

Výše popsáný model železnice byl v minulých semestrech využíván pro výuku v předmětu „Návrhy automatizovaných zařízení“. V průběhu výuky bylo zjištěno, že parametry řídicí jednotky nevyhovují požadavkům pro výuku. Hlavním problémem byla pomalá odezva řídicí jednotky a problémy s přerušovacím systémem mikroprocesoru, které jsou dány architekturou mikroprocesoru řady PIC16Fxxx.



Obrázek 2.1: Fotografie modelu železnice

2.2 Rekonstruovaný model

Vzhledem k výše uvedeným špatným vlastnostem modelu a k potřebě používat model ve výuce byly stanoveny následující požadavky na rekonstruovaný model.

Odezva řídicí jednotky na příkaz poslaný z nadřazeného systému musí být do 100ms. Řídicí jednotka musí mít dostatečný výpočetní výkon, který by umožňoval bezproblémové rozšíření modelu o další programové komponenty. Dalším požadavkem na model je zvýšení odolnosti vůči chybám, které jsou způsobeny studenty, kteří tento model používají, tj. model železnice se nesmí zablokovat. Posledními požadavky na model železnice jsou snadná rozšiřitelnost o další periferie modelu a zavedení vstupních signálů řídicí jednotky z přestavníku výhybek.

Kapitola 3

Obecné řešení zadaného problému

3.1 Řešení zadaného problému

Po seznámení se s modelem byly identifikovány následující hardwarové a softwarové nedostatky modelu. Prvním nedostatkem bylo rušení signálu v I²C sběrnici modelu železnice. Druhým nedostatkem byla špatná připojitelnost dalšího zařízení na I²C sběrnici nebo na paralelní sběrnici. Mezi další nedostatky modelu patří nedostatečný výpočetní výkon mikroprocesoru, nevektorový přerušovací systém a problémové programování mikroprocesoru.

Vzhledem k výše uvedeným nedostatkům bylo třeba výrazně změnit koncepci řídicí jednotky, program pro řídicí mikroprocesor a komunikační protokol. Jádrem řídicí jednotky se stal mikroprocesor ARM7 LPC2106, který má integrovaný dynamický vektorovaný přerušovací řadič a je dostatečně výkonný pro řízení modelu železnice a obsahuje dostatečné množství komunikačních periférií pro další rozšíření. Paralelní sběrnice a sběrnice I²C byly zrušeny a na radu Ing. Radka Šindeláře nahrazeny sítí LocoNet. Tato sběrnice se používá v moderních železničních modelech k jejich řízení. Pro řízení lokomotiv byl ponechán standard DCC, který již byl v modelu implementován.

Použití sběrnice LocoNet v modelu železnice vedlo k nutnosti upravit stávající části, anebo vytvořit nové části řídicí elektroniky, tak aby bylo možné všechny části řídicí elektroniky připojit k síti LocoNet.

Současný model oproti modelu minulému byl doplněn snímači polohy výhybky, dle požadavků na rekonstrukci modelu železnice.

Konkrétní hardwarové úpravy stávajícího zařízení a vytvoření nových zařízení je popsáno v kapitole 4. Popis vytvořeného softwarové vybavení modelu je v kapitole 5.

Kapitola 6 se zabývá komunikačním protokolem mezi nadřazenou aplikací a řídicí jednotkou.

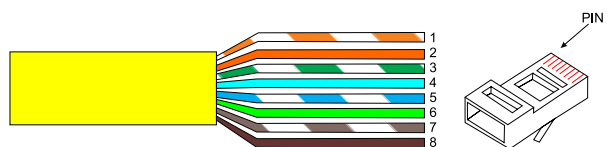
3.2 LocoNet

3.2.1 Obecný popis

LocoNet je standard pro elektronické řízení modelu železnice, který byl vyvinut firmou DIGITRAX® [1] pro řízení jejich modelů železnice. Sít' LocoNet je založena na principu multi-master komunikace, což znamená, že v síti může být více řídicích jednotek. Například moduly LocoIO mohou být konfigurovány jako slave, master nebo slave/master. Detailní popis LocoNet je uveden v [2].

3.2.2 Fyzická vrstva

Jedná se o asynchronní RS232 half-duplex protokol s rychlostí 16,66KBaud $\pm 1.5\%$, což umožňuje využít rychlost 16,457KBaud standardně generovaným UARTem mikroprocesorů. Přístup k síti je tzv. multiple access - k jednomu datovému kabelu přistupují všechny zařízení. Pro zvýšení odolnosti proti rušení musí mít log. 1 hodnotu mezi 12V až 24V, log. 0 musí mít úroveň 0V, rozhodovací úroveň na straně přijímače jsou 4V. Norma sítě LocoNET definuje připojení zařízení pomocí konektorů RJ12, ale vzhledem k obtížnější dostupnosti lze použít konektory RJ45, které nejsou v normě definovány.



Obrázek 3.1: Zapojení konektoru pro LocoNet

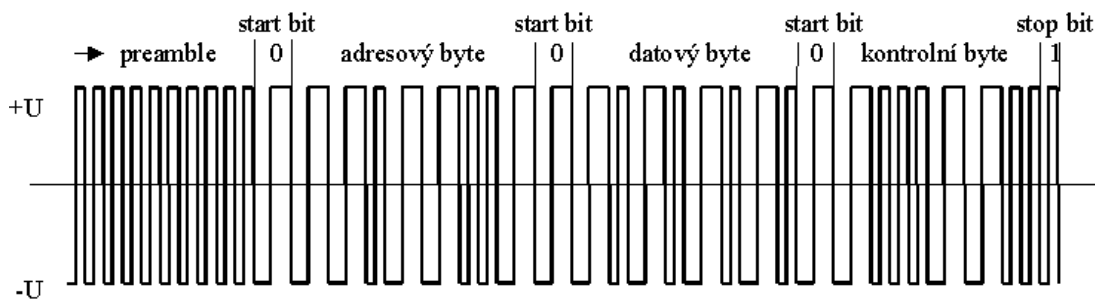
3.2.3 Linková vrstva

Pro přístup ke sběrnici je využito CSMA/CD techniky, která zaručuje přenos dat i v případě kolize na síti. Pokud vysílací jednotka detekuje kolizi na síti, musí přestat vysílat a musí

vyčkat dobu cca 20bitů (cca 1,2ms) + náhodnou dobu, aby nezačali všechny jednotky vysílat zároveň. Pokud chce jednotka vyslat více paketů, musí po vyslání či chybě na síti vyčkat dobu 20bitů. Po 25 pokusech o vyslání paketu je paket vyhodnocen jako nedoručený a následně zahozen. Pokud je na síti připojena řídicí jednotka, nemusí vyčkávat dobu 20bitů a po té může začít opět vysílat. Vysílané pakety jsou zabezpečeny kontrolním součtem a mohou obsahovat log. 1 na MSB pouze v prvním (příkaz) a posledním byte (kontrolní součet).

3.3 DCC

DCC standard je rozhraní pro řízení modelů lokomotiv definované normou sdružení NMRA [4]. Při použití DCC řízení není nezbytně nutné rozdělit kolejiště na několik elektricky oddělených částí, protože tato norma umožňuje nezávislé řízení více lokomotiv v jednom kolejovém úseku. DCC je založeno na jednosměrném vysílání modulovaného paketu lokomotivám nebo perifériím, které jsou připojeny k napájecím kolejím. Pakety jsou rozděleny na tři základní typy. První je paket určený pro řízení lokomotiv, druhý je určený pro ovládání připojených periférií a třetí typ je tzv. IDLE paket, který se vysílá v případě, že není vysílán žádný příkaz. Pokud byl řídicí jednotkou do kolejiště vyslán paket prvního typu, lokomotiva ho svým dekodérem přijme a přečte. V případě, že byl paket určen pro ni, začne jej okamžitě vykonávat, v opačném případě je paket zahozen. Obdobně by se zachovalo jakékoliv zařízení připojené k napájecím kolejnicím.

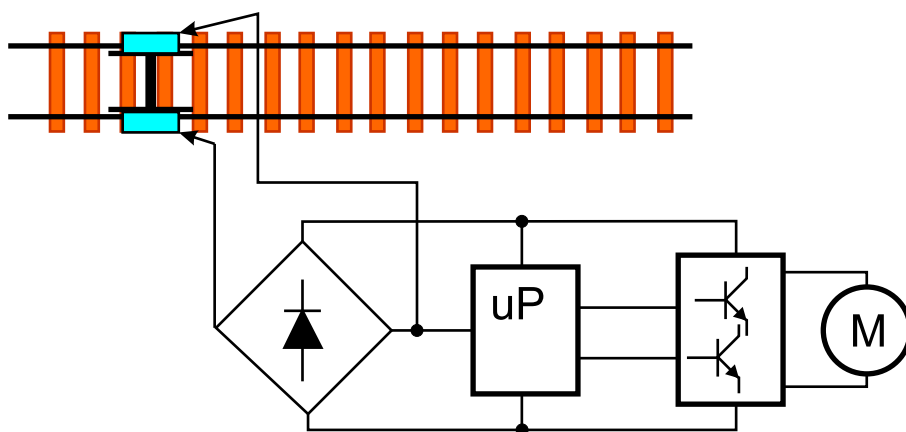


Obrázek 3.2: DCC paket

DCC paket je složen z preamble, adresy zařízení, dat a kontrolního součtu přenášených dat. Preamble je hlavička paketu určená k synchronizaci všech přijímačů připojených

k napájecí kolejnici. Tato hlavička paketu se musí skládat z minimálně 11 po sobě jdoucích log. 1. Pokud se tato adresa skládá z 8 po sobě jdoucích log. „0“ jedná se to tzv. broadcast. Dále následují data přenášená do přijímacího zařízení. Tato data jsou vždy oddělena log.0. Jako poslední se vyše kontrolní součet přenášených dat.

Obsah paketu je definován normou NMRA 9.2.1[6], je složen z řady posloupností log. „0“ a „1“. Kód log. 1 je definován jako obdélníkový signál o periodě 112us. Logická hodnota 0 je definována obdobě jako log. 1 s tím rozdílem, že délka jednotlivých částí signálu musí trvat od 100us do 10ms, díky čemuž je možné v modelu řídit jednu analogovou lokomotivu. Napěťové úrovně těchto signálů v modelu železnice jsou v rozmezí -12V až +12V.



Obrázek 3.3: Blokové schéma připojení dekodéru

Hlavní výhodou tohoto řízení je možnost řízení pohybu velkého počtu lokomotiv na jednom traťovém úseku nezávisle na sobě. Dále je jednodušší řízení a jednodušší výkonová elektronika. Nevýhodou je nutnost mít digitální řídicí jednotku na straně vysílače, tak i na straně přijímače (lokomotiv, ovládacích periferií ...).

3.4 Obecný popis komunikace nadřazené aplikace s řídicí jednotkou

Komunikace mezi nadřazeným systémem a řídicí jednotkou modelu se provádí pomocí datově zabezpečeného komunikačního protokolu, jehož zprávy jsou tvořeny definovanými

pakety. Každý paket může být složen z jednoho nebo více příkazů a musí být ukončen ukončovacím příznakem. Jednotlivé příkazy musí obsahovat označení typu příkazu, data a zabezpečovací byte. Při komunikaci mezi nadřazenou jednotkou musí být dodržen komunikační time-out, který je volitelný na ovládacím panelu mezi 10s a 100ms. Po uplynutí tohoto času je model kolejiště nastaven do výchozího stavu. Tomuto stavu odpovídá červená barva na všech semaforech, všechny výhybky jsou nastaveny na vedlejší kolej a lokomotivy jsou zastaveny.

Po přijetí paketu od nadřazeného systému řídicí jednotka nejdříve zkontroluje jeho zabezpečení a pošle zpět potvrzení o přijetí paketu. Posílané potvrzení obsahuje počet platných příkazů, které byly obsaženy v přijatém paketu. Poté začne jednotlivé příkazy postupně zpracovávat a vykonávat. Pokud je v paketu žádost o informace stavu kolejiště, řídicí jednotka odešle paket s požadovanými informacemi. Po dokončení všech operací souvisejících s přijatým paketem zašle řídicí jednotka nadřazenému systému zprávu o dokončení všech požadovaných operací. Dokud řídicí jednotka nepošle tuto zprávu, jsou všechny došlé pakety zrušeny bez jakéhokoliv potvrzení.

Řídicí jednotka je schopna v rámci řídicího protokolu informovat nadřazený systém o chybách, výpadcích a nepřítomnosti LocoIO modulů.

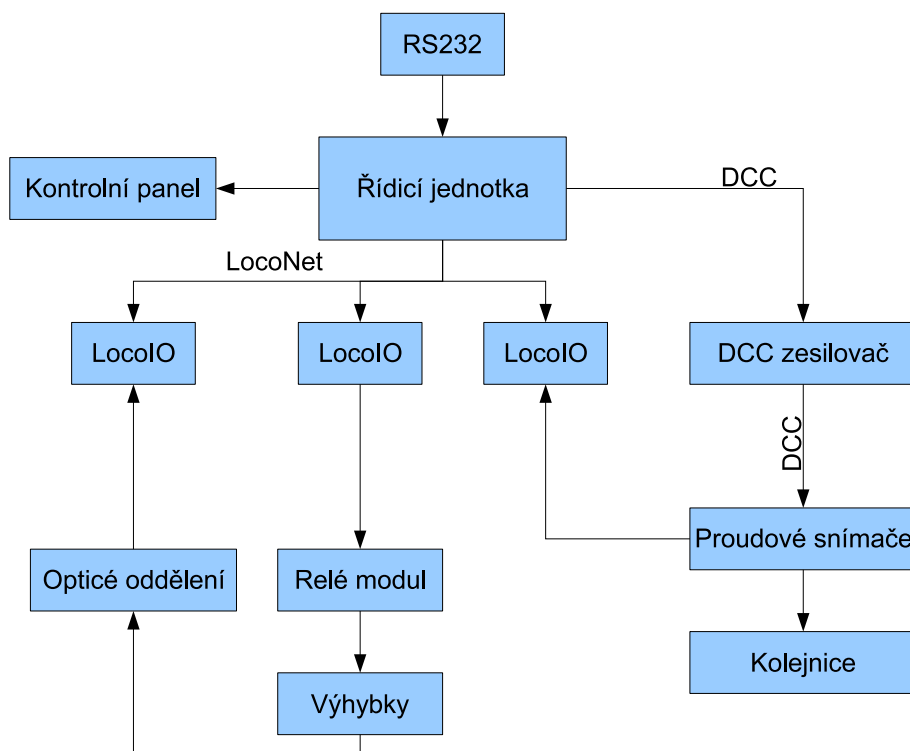
Kapitola 4

Řídicí elektronika

4.1 Konfigurace modelu železnice

V současné době model železnice obsahuje hlavní řídicí jednotku s řídicím mikroprocesorem ARM7 LPC2106, 10 LocoIO modulů (celkem 160 vstupů/výstupů) zajišťující snímání dat z kolejiště a řízení akčních členů. Všechny tyto jednotky jsou spojeny sítí LocoNet s hlavní řídicí jednotkou. Model železnice dále obsahuje 15 přestavníků výhybek, které jsou ovládány ze dvou reléových karet připojených k LocoIO modulům. Poslední ovládanou částí modelu je 12 semaforů, které jsou ovládány přímo LocoIO Moduly. Aby bylo možné snímat pozici lokomotiv na kolejišti je ke každému z 53 traťových úseků připojen jeden proudový snímač. V modelu je umístěno 7 modulů po osmi proudových snímačích. Výstupy těchto snímačů jsou připojeny přímo k modulům LocoIO. Model nově obsahuje zpětnou vazbu od přestavníků výhybek, která je zajištěna pomocí snímačů přestavení připojených k LocoIO modulu.

DCC signál z řídicí jednotky je před vstupem do kolejiště výkonově posílen a napětově upraven dle DCC standardu DCC zesilovačem. Z DCC zesilovače je zesílený DCC signál přiveden do proudových snímačů a z nich veden do jednotlivých úseků kolejiště.



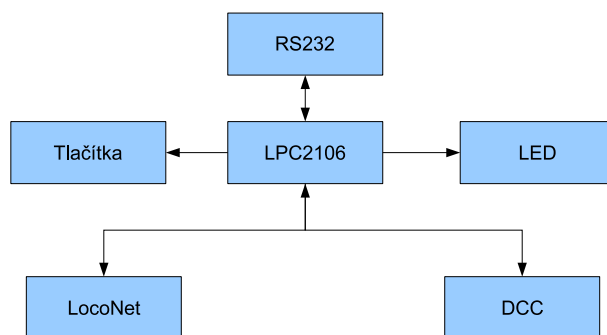
Obrázek 4.1: Blokové schéma zapojení kolejiště.

Pro přestavbu modelu bylo tedy nutné vyrobit novou řídicí jednotku, 10 modulů LocoIO a snímače přestavení výhybky. Dále byly upraveny proudové snímače traťových úseků a reléová karta. V modelu kolejiště zůstal nezměněný modul DCC zesilovače, který zajišťuje zesílení DCC signálu z řídicí jednotky a napájí jím celé kolejiště.

4.2 Řídicí jednotka

Řídicí jednotka se skládá z mikroprocesorového modulu MiniArm s mikroprocesorem LPC2106 s jádrem ARM7 od firmy Philips, napájecích zdrojů, které jsou určeny pro napájení samostatné řídicí jednotky a sběrnice LocoNet. Dále obsahuje sériovou linku RS232 pro komunikaci s nadřazenou jednotkou, komunikační rozhraní LocoNet a výstup pro DCC řízení. K indikaci napájení je na DPS umístěna červená LED dioda, pro hardwarový reset řídicí jednotky je na DPS umístěno mikrotlačítko. Schéma řídicí DPS je uvedeno v příloze A na str. 1. Rozsah napájecího napětí řídicí jednotky je 11V - 16V a celkový odběr proudu řídicí jednotky by neměl přesáhnout 0,5A. Na polaritě a průběhu

napájecího napětí nezáleží.

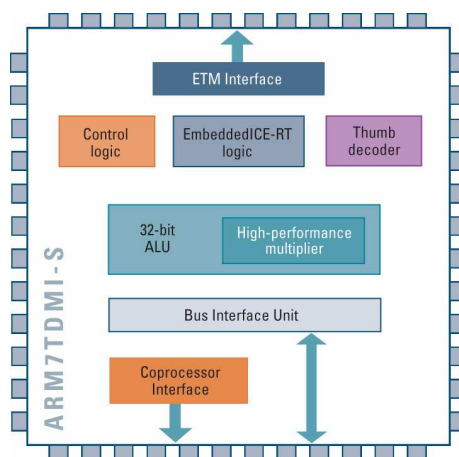


Obrázek 4.2: Blokové schéma řídicí jednotky.

4.2.1 Mikroprocesor ARM7 LPC2106

4.2.1.1 Stručný popis jádra ARM7

Jádro ARM7 (Acron RISC Machines) je moderní 32bitové RISC jádro mikroprocesoru, které je optimalizované pro nízkopříkonové, ale zároveň výpočetně výkonné procesory. Toto jádro bylo vyvinuto firmou ACRON [8], která jej vlastní a prodává tuto architekturu jiným výrobcům mikroprocesoru. Vzhledem k této skutečnosti je možné mezi mikroprocesory od různých výrobců přenášet kódy výpočetních funkcí bez jakýchkoliv úprav. Protože je jádro ARM7 chráněno HW patenty, jsou konkrétní informace o jeho architektuře velmi těžce dosažitelné.



Obrázek 4.3: Blokové schéma jádra ARM7.

4.2.1.2 Vlastnosti mikroprocesoru

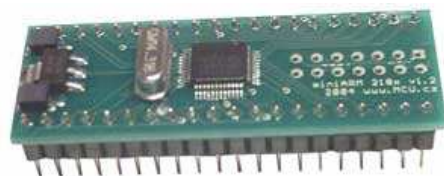
Mezi hlavní vlastnosti mikroprocesoru patří 3,3V napájení s +5V tolerantními vstupy, přímo adresovatelná 64kB SRAM a 128kB FLASH paměť. Mikroprocesor umožňuje ISP a IAP programování pomocí RS232 rozhraní. Pro komunikaci s okolím je mikroprocesor vybaven dvěma RS232 porty v CMOS úrovních, I2C sběrnici a SPI sběrnici. Mezi další periferie mikroprocesoru patří výkonná 32bitová PWM jednotka se šesti programovatelnými výstupy, dva 32bitové čítače/časovače, PLL jednotka s možností taktování mikroprocesoru až na 60MHz a obsahuje řadič přerušení s programovatelnou prioritou a adresou.

4.2.1.3 Programování mikroprocesoru

Software byl vyvíjen ve vývojovém prostředí Keil uVision3 [10] s GNU gccARM331 kompilér. K nahrání programu do mikroprocesoru byla použita aplikace LPC2000, která nahrává program do mikroprocesoru pomocí sériového rozhraní RS232. Software Keil uVision3 umožňuje integraci této aplikace a je ji tedy možné spouštět přímo z programovacího rozhraní.

4.2.2 Modul MiniARM

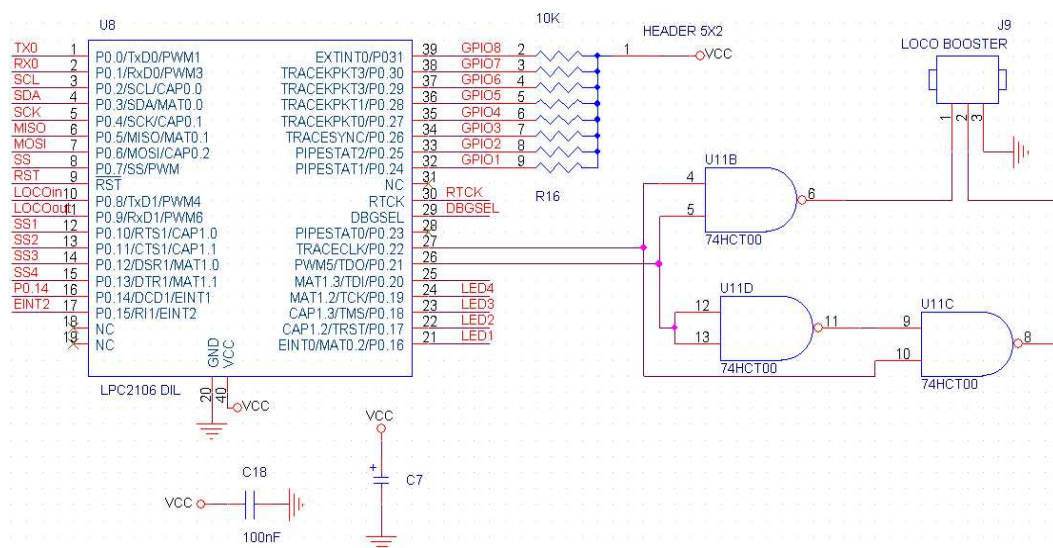
Modul MiniArm je zakoupený [5] malý embeded modul s mikroprocesorem LPC2106, který obsahuje veškeré nutné součástky k funkci mikroprocesoru. Zároveň jsou všechny jeho piny vyvedeny na 40ti pinovou patici což zjednodušuje návrh DPS.



Obrázek 4.4: Modul MiniArm.

4.2.3 DCC výstup

DCC signál je generován mikroprocesorem na výstupním pinu výstupu P0.21 (pátý kanál PWM jednotky). Tento pin je připojen na integrovaný obvod 74HCT00, který povoluje přenos DCC signálu na vstup zesilovače. Povolování nebo zakazování DCC signálu se provádí na základě signálu z pinu P0.22. Pokud je pin P0.22 v log. „1“ je výstup DCC signálu připojen na DCC zesilovač v opačném případě jsou na vstupy DCC zesilovače přivedené úrovně log „1“ a DCC zesilovač odpojí svůj výstup od traťových úseků. Další funkcí obvodu 74HCT00 je invertování DCC signálu, protože DCC zesilovač vyžaduje dva vzájemně invertované logické signály.

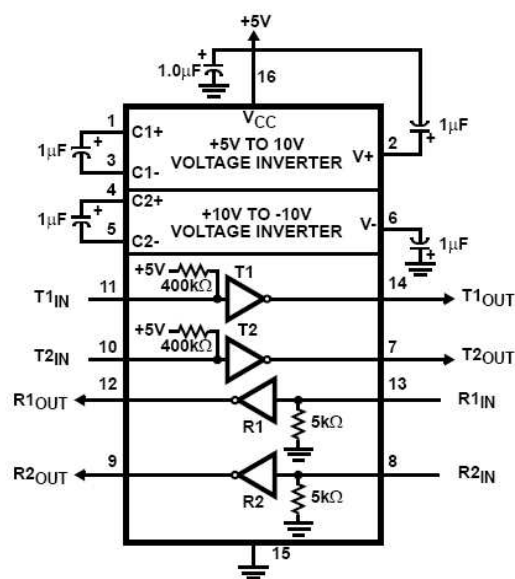


Obrázek 4.5: Schéma zapojení výstupu DCC.

4.2.4 RS232

Ke komunikaci s nadřazenou aplikací a programovací aplikací je využita sériová linka RS232. K převodu napěťových úrovní signálů RXD a TXD z TTL na RS232 je použit obvod ICL232-SMD v základním zapojení (obr. 4.6) dle katalogového listu tohoto obvodu.

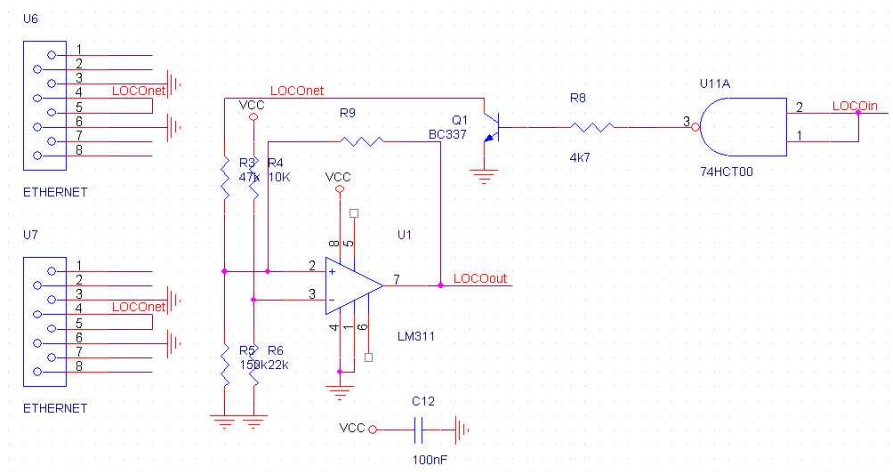
Pro programovací rozhraní jsou upraveny napěťové úrovně pinů DTR a RTS linky RS232, které jsou odpojitelné pomocí propojek J7 a J8. Řízením pinu RTS a propojením propojky J7 je možné řídicí jednotku resetovat. Pomocí pinu DTR a propojením propojky J8 je možné uvést mikroprocesor do programovacího stavu. Pokud je ke konektoru RS232 řídicí jednotky připojen sériový kabel pouze s linkami RXD a TXD je nutné rozpojit propojku J7 a J8, aby nebyla řídicí jednotka ve stavu kdy činnost mikroprocesoru je zastavena.



Obrázek 4.6: Základní zapojení obvodu ICL232.

4.2.5 LocoNet

Řízení sítě LocoNet je zajištěno pomocí jednotky UART1 v mikroprocesoru LPC2106. Vzhledem k tomu, že síť loconet je v zapojení half-duplex, je linka TXD invertována a přivedena na spínací tranzistor BC337, který zajišťuje spínání sběrnice LocoNet do log. „0“. Linka RXD je buzena z operačního zesilovače LM311 zapojeného jako komparátor. Vstup operačního zesilovače je přiveden na sběrnici LocoNet.



Obrázek 4.7: Schéma zapojení výstupu LocoNet.

Z principu sběrnice LocoNet je nutné, aby byla tato sběrnice buzena externím zdrojem proudu. Řídicí jednotka je opatřena odpojitelným 20mA zdrojem proudu, který je tvořen tranzistorem BC547. Pomocí propojky J4 je možné připojit zdroj proudu k síti LocoNet. Schéma napájecího zdroje bylo převzato z jednotky LocoBuffer.

4.3 LocoIO

LocoIO je univerzální modul vstupu a výstupů pro síť LocoNet vyvinutý panem De-loofem [3]. Tento modul je využit pro obsluhu periférií modelu železnice. Obsahuje 16 vstupně/výstupních vývodů, které mohou být nakonfigurovány jako vstup, výstup, výstup pro čtyřsměrný zobrazovač a pulzující výstup. Každému vývodu musí být udělena adresa v rozsahu 1 až 1024. Ve speciálních případech adresy mohou být duplikované.

Všechny vstupy jsou řízeny TTL úrovněmi v 5V úrovních. Vstupy na LocoIO jsou zapojeny jako otevřený kolektor a jsou opatřeny pull-up rezistory o hodnotě 10k Ohm. Velkou výhodou vstupů je to, že při jejich změně stavu vysílají řídicí jednotce nebo jinému LocoIO informaci o změně vstupů. Všechny vstupy na LocoIO modulu lze přechít najednou.

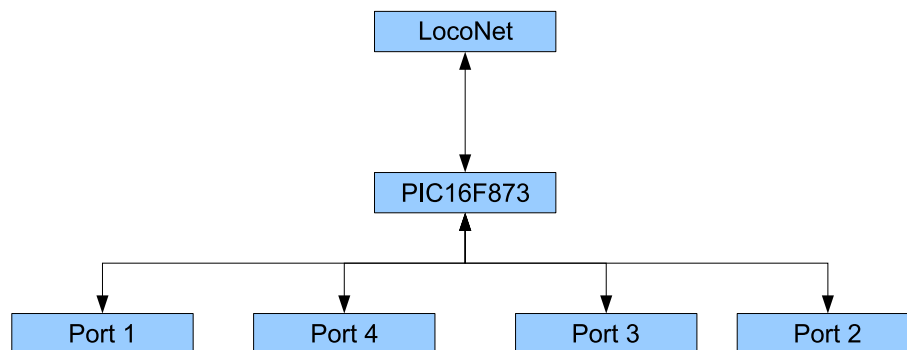
V případě, že vstupně/výstupní vývod LocoIO je nakonfigurován jako výstup, tak může spínat do log „0“ a log „1“ úroveň TTL 20mA při 5V a zároveň nesmí být překročen maximální přípustný součtový proud všech výstupů odpovídající 200mA. Výstupy mohou

přijímat rozkazy z řídicí jednotky nebo z jiných LocoIO vstupů. Výstupy mohou být nastaveny hromadně nebo jednotlivě.

Pro zjednodušení orientace v LocoIO jednotkách, jsou tyto celé jednotky konfigurovány jako vstupní nebo výstupní. Ke konfiguraci LocoIO modulů je nutné použít modul LocoBuffer a softwarovou aplikaci locoHDL. Aplikace je velmi dobře zdokumentována na stránkách HDL [3], v příloze [B] je návod na konfiguraci modulu.

Parametr	Hodnota
Vstupní napětí	8V - 16V
U_{out} log. „1“ výstupu	5V
U_{out} log. „0“ výstupu	0V
I_{max} výstupu	20mA
I_{max} celkový součet	200mA
U vstupu	5V
I_{vst} vstupu při log „0“	0,5mA

Tabulka 4.1: Tabulka elektrických vlastností modulu LocoIO



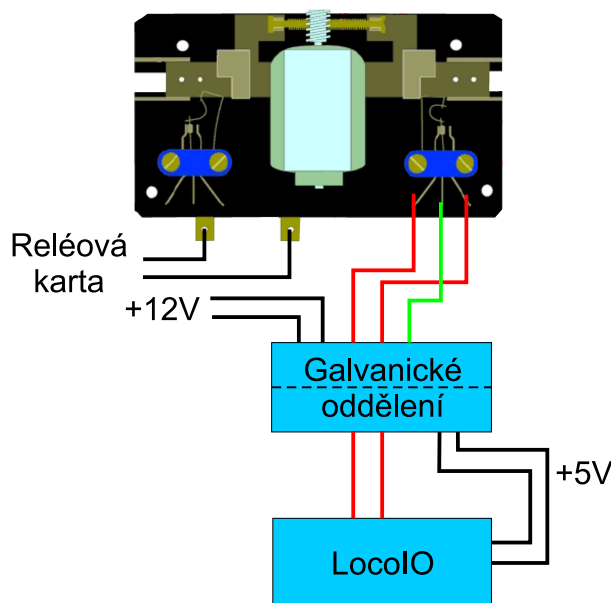
Obrázek 4.8: Blokové schéma modulu LocoIO.

4.4 Snímač polohy výhybky

Snímač polohy výhybky je složen ze dvou částí, které jsou mezi sebou elektricky spojeny. První část je mechanický přepínač (obr. 4.9), který je pevně spojený s přestavníkem

výhybky. Druhá část je tvořena optočlenem, který galvanicky odděluje první část snímače od jednotky LocoIO, ke které je snímač připojen.

Důvodem galvanického oddělení první části snímače od modulu LocoIO je skutečnost, že při přestavování výhybky motorovým přestavníkem z jedné pozice do druhé dochází vlivem magnetického pole od rotoru motoru přestavníku k indukci rušivého napětí do vodičů snímače, a tím k znehodnocení kvality signálu snímače. Vstupní část galvanického oddělení zajišťuje 12V napájení a snímání stavů mechanického přepínače. Výstupní část je napájena 5V přímo z jednotky LocoIO, ke které jsou připojeny výstupy galvanického oddělení. Na obrázku níže je náčrt zapojení jímáče přestavníku.



Obrázek 4.9: Blokové schéma snímače polohy výhybky.

4.5 LocoBuffer

LocoBuffer slouží výhradně ke konfiguraci LocoIO modulů a pro sběr dat ze sítě LocoNet při debuggingu řídicí jednotky. Tento modul byl navržen panem Deloofem [3]. LocoBuffer zajišťuje konverzi dat mezi sítí LocoNet a PC, se kterým komunikuje pomocí linky RS232 při komunikační rychlosti 57600Bd. Aby bylo možné LocoBuffer spojit s aplikací pro nastavení jednotek LocoIO je nutné propojit tyto propojky: konektor J7 piny 2 - 3, konektor J5 piny 2 - 3 a konektor J6 piny 2 - 3. LocoBuffer může být nastaven i do kompatibilního

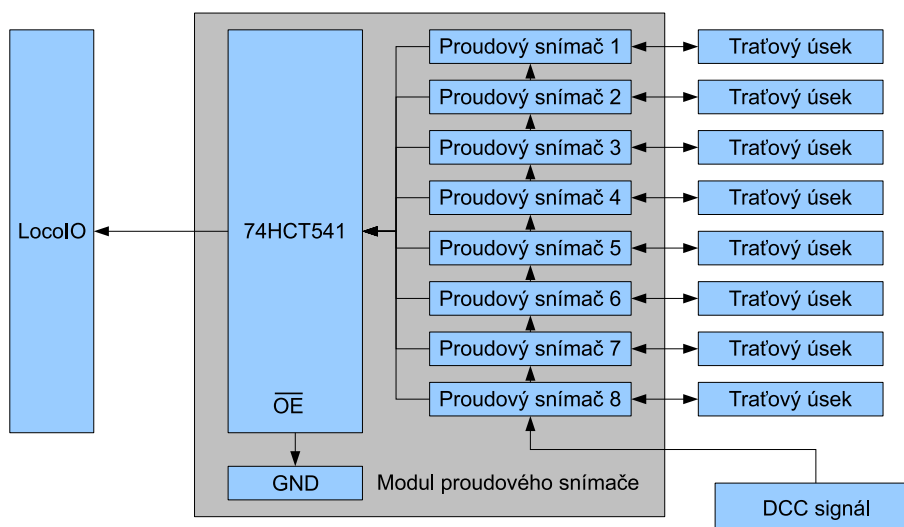
módu MS100, který není v kolejišti využit.

4.6 Snímače proudu

Moduly proudových snímačů jsou jedním z nejdůležitějších prvků zpětné vazby v modelu železnice. Poskytují informaci o tom, který úsek kolejiště je obsazen a který není.

Každý sledovaný úsek kolejiště má svoji vnější kolej elektricky izolovanou od sousedního úseku. Každá tato vnější kolej je individuálně napájena z DCC zesilovače přes snímač proudu. Pokud např. vjede lokomotiva do hlídaného úseku, začne z daného úseku odebírat elektrický proud, což má za následek sepnutí proudového snímače.

V modelu je umístěno celkem 7 modulů obsahujících celkem 56 snímačů. Každý modul obsahuje celkem 8 proudových snímačů. Výstupní část těchto modulů tvoří 8mi bitový sběrnice obvod 74HCT541, který zajišťoval při správné adresaci modulu jeho připojení k paralelní sběrnici. K připojení modulů snímačů k modulu LocoIO byl tento modul upraven tak, že vstup \overline{OE} obvodu 74HCT541 je trvale uzemněn a jednotlivé výstupy obvodu 74HCT541 jsou přímo připojeny na vstupy LocoIO jednotky, která signál potom odesílá řídicí jednotce.

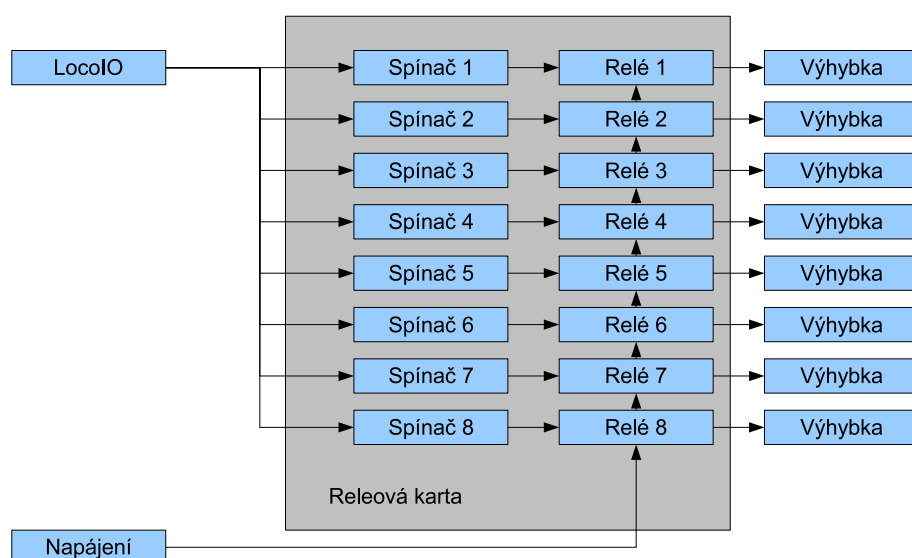


Obrázek 4.10: Blokové schéma modulu snímače proudu.

4.7 Reléová karta

Reléová karta je principiálně velmi jednoduchá. Skládá se ze dvou částí: galvanického oddělení od vstupů a spínacího relé. Vstupní signál z LocoIO jednotky je přiveden přes galvanické oddělení na cívku relé. Relé svým sepnutím či rozepnutím kontaktů pak přivede buď kladnou nebo zápornou polaritu napětí na vstup přestavníku výhybek.

V původním zapojení byla reléová karta řízena pomocí I²C sběrnice pomocí I²C expandéru PCF8574AP. Při použití modulů LocoIO bylo nutné schéma této jednotky upravit. Hlavní změna spočívá v odstranění expandéru a vyvedení jeho vstupů na výstupní konektor, který je připojen k LocoIO modulu.



Obrázek 4.11: Blokové schéma modulu reléové karty.

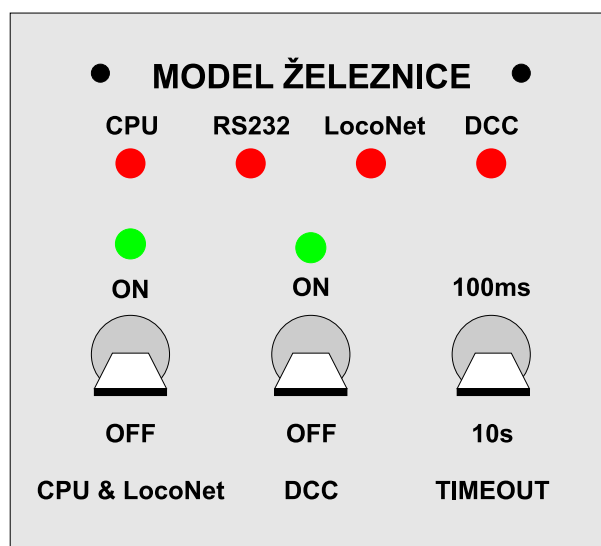
4.8 Ovládací panel

Ovládací panel slouží k jednoduché indikaci stavu modelu železnice, nastavení komunikačního timeoutu na lince RS232 a zapnutí napájení jednotlivých součástí modelu železnice.

Na panelu jsou umístěny čtyři červené LED diody, jejichž význam je uveden v následující tabulce.

LED	Barva	Stav	Význam
CPU	červená	svítí/nesvítí	Řídicí jednotka nefunguje
CPU	červená	bliká	Řídicí jednotka je inicializována
RS232	červená	svítí	Probíhá komunikace po R232
LocoNet	červená	svítí	Probíhá komunikace po LocoNet
DCC	červená	svítí	Kolejiště je napájeno DCC signálem
DCC	červená	nesvítí	Kolejiště není napájeno DCC signálem
CPU&Loconet	zelená	svítí	Připojené napájení pro kolejiště
DCC	zelená	svítí	Připojené napájení pro DCC zesilovač

Tabulka 4.2: Význam LED diod ovládacího panelu



Obrázek 4.12: Ovládací panel modelu kolejiště.

Kapitola 5

Software

5.1 Nastavení mikroprocesoru

V této kapitole jsou stručně popsány nastavení periférií mikroprocesoru a jejich základní funkční principy. Vzhledem ke složitosti jednotlivých periférií jsou vynechány popisy základních principů funkčnosti obecných periférií v mikroprocesorech, protože je předpokládána jejich základní znalost. Kompletní informace o konfiguraci jednotlivých periférií jsou uvedeny v katalogovém listu [13] uloženém na CD, které je součástí této bakalářské práce. Pro snazší porozumění jsou v této kapitole uvedeny i části konfiguračních kódů, které jsou použity v programu mikroprocesoru.

5.1.1 Přerušovací systém mikroprocesoru

Mikroprocesor LPC2106 obsahuje vysoce výkonný řadič přerušení, který umožňuje vlastní adresaci a nastavení priority přerušení nezávisle na periférii, která přerušení vyvolala. Řadič přerušení obsahuje 32 programovatelných přerušovacích vstupů. Tyto vstupy je možné roztrždit do tří oblastí: Fast Interrupt reQuest (FIQ), vectored Interrupt ReQuest (vectored IRQ) a non-vectored Interrupt ReQuest (non-vectored IRQ).

Skupina „Fast Interrupt reQuest“ (dále jen FIQ) má nejvyšší prioritu a je určena pro časově kritické události, protože zaručuje přerušení v co nejkratším čase. FIQ přerušovací vektor je jediný a je určen pouze pro obsluhu jediné události. V hlavní řídicí jednotce se tento přerušovací vektor používá pro generování PWM signálu pro DCC signál, který dle NMRA normy musí být velmi přesný. Při konfiguraci tohoto přerušovacího vektoru je nezbytné splnit následující požadavky: umístění od adresy 0x40000000h výše, název

obslužné funkce `FIQ_Handler` a název funkce musí být doplněn direktivou překladače `_fiq`.

Příklad funkce pro nastavení FIQ přerušení PWM jednotky:

```
void init_PWM(void)          // Nastavení preruseni PWM jednotky - FIQ preruseni
{
    ...
    // Nastavení PWM jednotky
    ...

    PWMCR = 1;                // Povolí přerušení od PWM jednotky
    VICIntSelect = 1<<8;      // Nastaví přerušení řadiče přerušení
    VICIntEnable = 1<<8;      // Povolí přerušení od PWM jednotky
}
```

Příklad funkce pro obsluhu FIQ přerušení PWM jednotky:

```
void FIQ_Handler (void) __fiq      // obsluha preruseni PWM jednotky - FIQ preruseni
{
    ...
    // Obslužný kód
    ...
    PWMIR = 1;                // Vymaže příznak přerušení
}
```

Vectored Interrupt ReQuest (dále jen IRQ) je 16ti vstupý vektorovaný přerušovací řadič, který umožňuje téměř libovolnou prioritu 16 vektorů přerušení. Toto přerušení je pomalejší než FIQ právě z důvodu nutnosti výběru a řízení priorit přerušení. Tohoto přerušení je v řídicí jednotce využito pro komunikaci se sběrnici LocoNet a pro odměřování 100ms časových intervalů. Ačkoliv je možné tyto přerušení za běhu programu měnit, je nutné k obslužným funkcím přidat direktivu překladače `_irq`. Po vyvolání přerušení je nutné smazat bit v registru `VICVectAddr`, který odpovídá vyvolanému přerušení.

Příklad funkce pro nastavení IRQ přerušení Timeru0:

```
void init_timer (void) {
    ...
    TOMCR = 3;                // Interrupt and Reset on MR0
    ...

    VICVectAddr0 = (unsigned long)tc0;    // Uloží adresu obsluhy přerušení
    VICVectCntl0 = 0x20 | 4;             // Přiřadí typ přerušení
    VICIntEnable = 0x00000010;           // Povolí přerušení
}
```

Příklad funkce pro obsluhu FIQ přerušení PWM jednotky:

```
void tc0 (void) __irq {
    ...
    TOIR      = 1;          // Potvrzení přerušení Timeru
    VICVectAddr = 0;        // Potvrzení přerušení Timeru v řadiči přerušení
}
```

Non-vectored Interrupt ReQest slouží převážně k obsluze méně důležitých přerušení a velmi často se používá pro SW generované přerušení. Toto přerušení se používá hlavně v operačních systémech reálného času. V řídicí jednotce toto přerušení není použito.

5.1.2 Výstupní porty

Mikroprocesor LPC2106 obsahuje jeden 32 bitový vstupně-výstupní port. Ke každému z těchto pinů je přiřazeno více funkcí. Aby bylo možné mezi funkcemi přepínat je mikroprocesor vybaven dvěma 32bitovými registry PINSEL0 a PINSEL1 pomocí kterých se vybírá konkrétní funkce daného pinu. Vždy dva bity registrů PINSEL0 a PINSEL1 ovládají jeden vstupně výstupní pin portu. Registr PINSEL0 nastavuje funkci pinů P0.0 až P0.14 a registr PINSEL1 nastavuje funkci pinů P0.16 až P0.31.

Příklad nastavení pinu P0.21 na PWM jednotku - 5. kanál:

```
PINSEL1 |= 1<<10 ;          //Nastaví pin P0.21 na PWM5
PINSEL1 &= ~(1<<11);
```

Pro práci s piny nastavenými jako GPIO slouží registry IOPIN, IOSET, IODIR a IOCLR. Registr IOPIN obsahuje aktuální stav pinů nezávisle na jejich vnitřní konfiguraci. IODIR určuje, zda-li je pin vstupní nebo výstupní. Registry IOSET a IOCLR jsou určeny pro nastavení výstupní úrovně pinu v režimu výstupní pin, v jiném režimu je stav těchto registrů ignorován. Výstupní hodnota na výstupním pinu je dána posledním nastavením příslušného pinu na log „1“ v registru IOSET nebo IOCLR. Bit v IOSET nastavuje výstupní pin do log „1“ a příslušný bit v IOCLR nastavuje pin do log. „0“.

Příklad obsluhy pinu P0.16:

```
IODIRO = (1 << 16);          // Nastavení pinu P0.16 jako výstupní
IOCLRO = (1 << 16);          // Nastavení pinu P0.16 na log 0
IOSET0 = (1 << 16);          // Nastavení pinu P0.16 na log 1
```

5.1.3 PWM jednotka

Mikroprocesor je vybaven jednou PWM jednotkou se šesti PWM kanály. Tyto PWM kanály mohou být konfigurovány v režimech "single edge" nebo "double edge", přičemž ke každému kanálu je připojený jeden výstupní pin. Každý z těchto výstupních pinů může být konfigurován ve čtyřech režimech: přímý, inverzní, otáčí polaritu a neaktivní. PWM jednotka obsahuje celkem 11 registrů, jejichž funkce je popsána v tabulce 5.1.

Název registru	Popis	Přístup
PWMIR	PWM registr přerušení. Čtením tohoto registru se identifikuje vyvolané přerušení a zápisem se nuluje událost přerušení.	zápis/čtení
PWMTCR	Registr řízení časovače PWM jednotky.	zápis/čtení
PWMTCC	Registr obsahuje aktuální hodnotu časovače PWM jednotky a je inkrementován každých PWMPR + 1 tikem pclk.	zápis/čtení
PWMPR	Registr určuje vrchol registru PWMPCC.	zápis/čtení
PWMMCR	Tento registr zapíná jednotlivé PWM kanály a určuje způsob generování PWM.	zápis/čtení
PWMMR0	PWM match register 0 určuje délku periody všech PWM kanálů. Při jeho naplnění resetuje čítač PWMTCC a nastaví výstupní piny PWM na požadovanou hodnotu. Zároveň může být použit pro režim "double edge" prvního kanálu PWM jednotky.	zápis/čtení
PWMMR1-6	PWM match registry 1 - 6 obsahují dobu první části periody pulzu v režimu "single-edge". V režimu "double-edge" jsou registry speciálně propojeny.	zápis/čtení
PWMPCCR	PWMPCCR povoluje výstupy PWM kanálů a udává jejich funkci.	zápis/čtení
PWMLER	Nastavováním patřičných bitů v PWMLER se aktualizují hodnoty PWM jednotek na hodnoty zapsané v PWMMR0-6.	zápis/čtení

Tabulka 5.1: Význam registrů PWM jednotky

Příklad použití PWM jednotky jako generátor PWM signálu s FIQ přerušením:

```
void init_PWM(void){
    PNSSEL1 = 1 << 10;    // Nastavi pin P0.21 na PWM5
    PWMMR0 = Pulz10;      // doba periody 116us
    PWMMR5 = Pulz11;      // doba log 1 = 58us
    PWMMCR = 1;           // povoli interupt na Match0
    VICIntSelect = 1 << 8; // Vybere jako zdroj přerušení PWM5
    VICIntEnable = 1 << 8; // Povolí PWM přerušení
    PWMPCR = 1 << 13;      // povolení PWM5
    PWMTCR = (1 << 0) | (1 << 3); // spustení PWM
}
```

5.1.4 Sériový kanál

Mikroprocesor LPC2106 obsahuje dvě UART jednotky. Jednotka UART0 je zjednodušená a obsahuje pouze piny RXD a TXD, oproti tomuto jednotka UART1 obsahuje kompletní rozhraní RS232. Obě tyto UART jednotky obsahují stejné registrové sady, proto bude dále popisována registrová sada jednotky UART0. Funkce jednotlivých registrů je popsána v tabulce 5.2 a rozložení nastavení registrů je zobrazeno na obrázku obr. 5.1.

Hlavní výhody jednotek UART jsou:

- integrovaný generátor přenosové rychlosti
- dvojitá 16 bytová FIFO paměť s možností generování přerušení od čtyř úrovní obsazenosti FIFO paměti
- široká možnost konfigurace datového slova
- kompatibilita se standardem 550 industry standard

Název registru	Popis	Přístup
U0RBR	U0RBR je součástí přijímacího zásobníku FIFO a obsahuje poslední přijaté datové slovo.	čtení
U0THR	U0THR je vrchol vysílacího zásobníku FIFO. Pro přístup do tohoto registru musí být bit DLAB v U0LCR nastaven na log „0“.	zápis
U0IER	Zapsáním do tohoto registru se vybírají zdroje přerušení jednotky UART0.	zápis/čtení
U0IIR	Registr U0IIR obsahuje identifikátor vyvolaného přerušení.	zápis/čtení
U0FCR	Registr U0FCR řídí FIFO jednotky.	zápis/čtení
U0LCR	Nastavováním registru U0LCR se vybírají parametry datového slova.	zápis/čtení
U0LSR	Registr U0LSR obsahuje aktuální stav UART jednotky.	čtení
U0DLL	Registr U0DLL obsahuje dolní byte děličky generátoru přenosové rychlosti.	zápis/čtení
U0DLH	Registr U0DLH obsahuje horní byte děličky generátoru přenosové rychlosti.	zápis/čtení
U0SCR	Tento registr nemá žádné využití.	zápis/čtení

Tabulka 5.2: Význam registrů UART0 jednotky

Address Offset	Name	Description	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	Access	Reset Value*
0xE000C000 DLAB = 0	U0RBR	Receiver Buffer Register	MSB READ DATA LSB								RO	un-defined
0xE000C000 DLAB = 0	U0THR	Transmit Holding Register	MSB WRITE DATA LSB								WO	NA
0xE000C004 DLAB = 0	U0IER	Interrupt Enable Register	0	0	0	0	0	Enable Rx Line Status Interrupt	Enable THRE Interrupt	Enable Rx Data Available Interrupt	R/W	0
0xE000C008	U0IIR	Interrupt ID Register	FIFOs Enabled		0	0	IIR3	IIR2	IIR1	IIR0	RO	0x01
0xE000C008	U0FCR	FIFO Control Register	Rx Trigger		Reserved		-	Tx FIFO Reset	Rx FIFO Reset	FIFO Enable	WO	0
0xE000C00C	U0LCR	Line Control Register	DLAB	Set Break	Stick Parity	Even Parity Select	Parity Enable	Number of Stop Bits	Word Length Select		R/W	0
0xE000C014	U0LSR	Line Status Register	Rx FIFO Error	TEMT	THRE	BI	FE	PE	OE	DR	RO	0x60
0xE000C01C	U0SCR	Scratch Pad Register	MSB LSB								R/W	0
0xE000C000 DLAB = 1	U0DLL	Divisor Latch LSB	MSB LSB								R/W	0x01
0xE000C004 DLAB = 1	U0DLM	Divisor Latch MSB	MSB LSB								R/W	0

Obrázek 5.1: Význam bitů v registrech UART.

Vzhledem k širokým možnostem jednotek UART je popis zjednodušen na základní konfiguraci této jednotky. UART jednotka obsahuje celkem 10 registrů. Pro nastavování komunikačních parametrů přenášeného datového slova slouží registr U0LCR, ve kterém je možné nastavit délku datového slova, počet stop-bitů, generátor parity a také obsahuje bit DLAB, který je důležitý pro konfiguraci generátoru přenosové rychlosti. Pro řízení generátoru rychlosti jsou určeny registry U0DLL a U0DLM, které dohromady obsahují 16ti bitovou hodnotu děličky generátoru rychlosti. Pro modifikaci těchto registrů je nutné, aby byl bit DLAB v registru U0LCR nastaven na log „1“. Samostatné vyslání datového slova se provádí zapsáním byte do registru U0THR, tento registr je připojen na FIFO jednotku tvořící vysílací buffer a je tedy možné do něj zapsat najednou až 16 datových

slov. Příchozí datové slovo je ukládáno do registru U0RBR, tento registr je také připojen na FIFO jednotku tvořící přijímací buffer a může obsahovat až 16 datových slov.

Příklad inicializace UART0 jednotky:

```
void init_uart0(void) { // komunikace s PC
    PINSEL0 |= ( 1<< 0) | ( 1<< 2); // nastavení výstupních pinů
    UOLCR =(1<<0)|(1<<1)|(1<<7); // nastavení datového slova na 8 bitů a
    // jeden stop bit bez parity, nastavení DLAB na log "1"
    UODLL = 0x0E; // nastavení předěličky na 57600Bd
    UODLM = 0x00;
    UOLCR = (1<<0)|(1<<1); // vynulování DLAB
}
```

Příklad přijmutí datového slova:

```
unsigned char uart0GetCh(void){
    while (!(UOLSR & 1)); // čekání na příchozí byte
    return U0RBR; // předání příchozího byte
}
```

Příklad vyslání datového slova:

```
void uart0SendCh(char ch){
    while (!(UOLSR & 0x20)); // testování obsazenosti FIFO paměti
    U0THR=ch; // vyslání byte
}
```

5.1.5 Čítače a časovače

Mikroprocesor obsahuje jednotky TIMER0 a TIMER1, tyto dvě jednotky je možné nakonfigurovat jako čítače, časovače a PWM jednotky. Pro model železnice je použita jednotka TIMER0 v režimu časovače, který je konfigurován pro generování přerušení každých 100ms. Vzhledem k širokým možnostem jednotky TIMER0 je popis zkrácen pouze na konfiguraci použitou v řídicí jednotce.

Při konfiguraci této jednotky na časovač generující přerušení každých 100ms je nutné nastavit registry: T0TCR, T0MCR, T0MR0. Zápisem do registru T0TCR se zapíná a vypíná funkce timeru. Registrem T0MCR se vybírá akce, která se uskuteční při dosažení vrcholu zásobníku timeru (např. nulování timeru, generování přerušení). Registr T0MR0 obsahuje maximální hodnotu do které čítač čítá.

Příklad konfigurace Timeru jako generátor 100ms intervalů s nastavením přerušení:

```
void init_timer (void) {
    TOMR0 = 149999; // 10mSec = 100.000-1 tiků
    TOMCR = 3;      // Nastaví generování přerušení a nulování timeru
    TOTCR = 1;      // Povolí Timer0

    time = timeval; // Uložení počáteční hodnoty časovače

    VICVectAddr0 = (unsigned long)tc0; // Uloží adresu obsluhy přerušení
    VICVectCntl0 = 0x20 | 4;          // Přiřadí typ přerušení
    VICIntEnable = 0x00000010;        // Povolí přerušení}
```

5.2 Popis programu

Tato část kapitoly je věnována popisu řídicího programu v mikroprocesoru LPC2106. Program je rozdělen do tří částí: inicializace mikroprocesoru, hlavní program a obsluha přerušení. Inicializace mikroprocesoru nastavuje parametry komunikačních portů, PWM jednotky a nastaví přerušovací systém mikroprocesoru. Hlavní program je tvořen cyklem, který začíná čekáním na příchozí zprávu z linky RS232. Při přijetí příchozího paketu, je paket nejdříve zkontrolován a poté je postupně vykonáván. Jakmile jsou všechny příkazy v paketu zpracovány a vykonány, přejde hlavní řídicí program opět do stavu čekání na příchozí paket. Obsluha přerušení PWM jednotky obstarává generování DCC signálu a obsluha přerušení jednotky UART1 zajišťuje sběr dat ze sítě LocoNet.

5.2.1 Popis inicializace mikroprocesoru

Inicializace periférií mikroprocesoru je umístěna ve funkci `init_perip`, která je volána na začátku hlavního programu. Z této funkce jsou volány jednotlivé inicializační funkce periférií.

Rozhraní jednotky UART0 pro komunikaci s nadřazeným systémem je konfigurováno jako asynchronní sériové rozhraní s přenosovou rychlostí 57600Bd. Formát datového slova je tvořen start bitem, osmi datovými bity a jedním stop bitem, datové slovo není opatřeno paritním bitem.

Jednotka UART1 je určena pro řízení sběrnice LocoNet a její komunikační parametry jsou dány normou rozhraní LocoNet. Přenosová rychlost linky je stanovena na 16457Bd

a datové slovo je tvořeno jedním start bitem, osmi datovými bity a je ukončeno jedním stop bitem. Zároveň je jednotce UART1 přiřazen jeden vektor přerušení vectored IRQ s nejvyšší prioritou a je zaregistrována obsluha tohoto přerušení.

Časovač TIMER0 generuje každých 100ms přerušení, které využívá jeden IRQ kanál s druhou nejvyšší prioritou.

PWM jednotka je nastavena pro generování "single edge" PWM signálu na pátém kanálu. Výstupní pin je konfigurován na přímý režim. Protože je generování PWM signálu časově kritické, je této jednotce přiděleno FIQ přerušení, které zajistí obsluhu události v co nejkratším čase.

5.2.2 Popis hlavního programu

Hlavní program je rozdělen do tří částí: čekání na příchozí paket, zpracování příchozího paketu a jeho vykonání.

První část programu zajišťuje přijetí paketu a odměřování doby komunikačního timeoutu. Pokud není přijatý platný paket v době do uplynutí timeoutu, je kolejiště uvedeno do výchozího stavu pomocí funkce setDefault.

Pokud byl přijat platný datový paket, je přeposlán funkci sendAck, která jej rekonstruuje a ověří platnosti příkazů v paketu. Pokud je v paketu některý příkaz neplatný jsou všechny následující příkazy označeny za neplatné. Po skončení této funkce je nadřazené jednotce posláno potvrzení o doručení paketu s počtem platných příkazů. Výstup funkce je dále předán funkci CallDrScan, kde je postupně zpracován. Všechny operace jsou vykonávány nad datovou strukturou, která reprezentuje aktuální stav kolejiště.

Po zpracování všech příkazů v paketu je zavolána funkce locoUpdate, která zajistí aktualizaci vybraných LocoIO jednotek. Aktualizace parametrů lokomotiv je zcela automatická. Funkce LocoUpdate vybere LocoIO jednotky, které je nutné aktualizovat a vyšle jim paket pomocí funkce LocoSend. Tato funkce nejdříve zjistí, zda-li v obsluze přerušení jednotky UART1 již není přijímán paket, případně vyčká na jeho dokončení. Jakmile již není v obsluze přerušení přijímán paket, je přerušení jednotky UART1 zakázané a řízení sběrnice převezme hlavní řídicí program. Poté je na sběrnici vyslán paket pro jednotku LocoIO a čeká se na její odpověď. Pokud odpověď nepřijde do stanoveného timeoutu je požadavek na nastavení LocoIO jednotky opakován. Po dvacátém pokuse je signalizována chyba přenosu dat. Po dokončení aktualizací LocoIO jednotek přejde hlavní program opět do stavu čekání na příchozí paket.

5.2.3 Obsluha přerušení PWM jednotky

Vzhledem k tomu, že jednotlivé bity DCC signálu jsou složeny z pulzů s přesně definovanou délkou, je ke generování těchto pulzů je použita PWM jednotka. Obsluha a řízení PWM jednotky je rozdělena do dvou spojených částí.

První část zajišťuje výběr následujícího bitu v DCC paketu a jeho převedení na délku pulzu, kterou následně vygeneruje PWM jednotka. V následujícím přerušení je opět vybrán další bit DCC paketu k převedení na pulz.

Druhá část se provede pouze po vyslání celého DCC paketu. Jejím úkolem je ze seznamu lokomotiv vybrat další následující lokomotivu, její rychlost a připravit datovou strukturu na vyslání nového DCC paketu. Protože není pevně dán počet lokomotiv na kolejišti, jsou data posílána pouze zaregistrovaným lokomotivám u nadřazeného systému.

5.2.4 Obsluha přerušení UART1 jednotky

V obsluze přerušení jednotky UART1 je zajištěno přijímání paketů ze sběrnice LocoNet. Protože jsou pakety tvořeny třemi bajty, což neodpovídá žádné úrovni přerušení FIFO jednotky, jsou bajty paketu přijímány postupně. Zároveň je hlídáno poškození paketu jiným vysílačem a v takovémto případě je přijímaný paket zahozen. Po přijetí platného paketu je paket dekodován a jeho výsledek je uložen do datového obrazu kolejiště.

Kapitola 6

Komunikační protokol

6.1 Popis protokolu

Výměna dat mezi nadřazenou aplikací a řídicí jednotkou se provádí pomocí paketů. Každý paket může začít libovolným příkazem, ale musí být vždy ukončen znakem 0xFF. Pokud nebude tento znak vyslán do 10 sekund od začátku vysílání paketu nadřazenou aplikací nebo od posledního poslaného znaku 0xFF řídicí jednotkou, řídicí jednotka smaže všechna příchozí data a zinicilizuje kolejiště. Stav čekání na paket a komunikace po RS232 nemá vliv na příjem dat ze sběrnice Loconet a řízení modelu železnice. Paket může obsahovat téměř libovolný počet příkazů, jen nesmí jeho délka přesáhnout 100 bajtů. Jednotlivé příkazy jsou zabezpečeny dvojím způsobem:

- Logický XOR všech hodnot příkazu, logicky vynásobený hodnotou 0x7F
- Na pozici MSB smí být log. „1“ pouze u prvního byte příkazu. Ve všech ostatních bytech příkazu musí být na pozici MSB log. „0“.

Řídicí jednotka umí, kromě běžné komunikace s nadřazenou aplikací, oznamovat chyby, případné výpadky nebo nepřítomnost LocoIO jednotek.

6.2 Reset jednotky

Řídicí jednotku je možné restartovat buďto odpojením a znovu připojením napájecího napětí nebo pomocí signálu DTR (pouze krátký pulz do log. „1“, cca >1ms). Při restartu

probíhá inicializace kolejiště. Dokud není inicializace úspěšně provedena, řídicí jednotka cyklicky posílá chybovou hlášku (FE,02,F6,FF). Po úspěšné inicializaci je odeslána inicializační sekvence (FE,01,7F,FF).

Po úspěšném restartu je doporučeno uvést kolejiště do definovaného stavu pomocí systémového příkazu (0xF7,0x00,0x55,0x22,0xFF).

6.3 Průběh komunikace

V dalším textu bude nadřazená aplikace označována jako master, řídicí jednotka modelu jako slave. Jakmile je úspěšně provedena inicializace modelu, je slave schopen přijímat od mastera povely. Na každý přijatý paket od masteru slave odpovídá potvrzovacím pake-tem ve tvaru: 0xFA, počet přijatých platných příkazů, kontrolní součet. Pokud je v paketu odeslaného z masteru neplatný příkaz, je tento příkaz a všechny následující příkazy slavem označeny za neplatné. Po odeslání potvrzení začne slave zpracovávat příkazy. Zapisovací příkazy nemusí být vždy vykonány v pořadí v jakém byly přijaty. Čtecí příkazy jsou vykonány postupně tak jak byly přijaty. Po vykonání všech příkazů, které přijatý paket obsahoval, zašle slave zprávu ve tvaru: 0xFC, 0x55, 0x29, 0xFF. Dokud slave neodpoví zprávou o dokončení příkazů, nepřijímá žádná další data.

Příklad komunikace:

Nadřazený počítač:

Řídicí jednotka:

Inicializace

Pulz na DTR

FE,01,7F,FF/FE,02,F6,FF (ok/chyba)

0xF7, 0x00, 0x55, 0x22,0xFF - uvedení kolejiště do definované pozice

0xFA,0x01,0x7B, - potvrzení příjmu 1 příkazu

0xFC,0x55,0x29,0xFF - příkazy dokončeny

...

Normální komunikace

0xF2,0x01,0x00,0x3B,0x48, - semafor 1 a 2 nastavit na červenou
 0xF0,0x00,0x00,0x40,0x30, - lokomotiva 1 vpřed rychlosti 59
 0xFF
 0xFA,0x02,0x78, - potvrzení příjmu 2 příkazu
 0xFC,0x55,0x29,0xFF - příkazy dokončeny
 ...

Chybová komunikace

0xF2,0x01,0x00,0x3B,0x48, - semafor 1 a 2 nastavit na zelenou
 0xF0,0x00,0x00,0x40,0x30, - lokomotiva 1 vpřed rychlosti 59
 0xFF
 0xFA,0x02,0x78, - potvrzení příjmu 2 příkazu
 0xFE,0x03,0x7D - kolejiště není napájeno
 0xFC,0x55,0x29,0xFF - příkazy dokončeny
 ...

6.4 Příkazy a příklady

Základními stavebními prvky komunikace mezi nadřazenou aplikací a řídicí jednotkou kolejiště jsou následující parametry:

- Loconr - udává číslo ovládané lokomotivy. Číslování začíná od 0, tj. 0 je první lokomotiva.
- Speed - určuje rychlost a směr lokomotivy. Směr jízdy je dán nejvyšším bitem hodnoty. Rychlost je možné nastavovat v rozmezí 0, 2 ... 127. Hodnota 1 je vyhrazena pro nouzové zastavení lokomotivy.
- Extend - obsahuje chybějící MSB hodnotu z následujících bajtu. Avšak maximálně

4 následujících bajtů, poté je generován nový extend byte - toto se uplatňuje pouze u odpovědi SLAVE na příkaz 0xF6.

- Chck - obsahem parametru je logický součin hodnoty 0x7F a logického XORu všech datových bajtů v příkazu.
- Val - zapisovaná hodnota do paměti.
- Seg - určuje segment, do kterého se zapisuje hodnota val, segmenty se liší podle použitého příkazu viz. tabulky níže.
- Addr - fyzická adresa lokodekodéru lokomotivy.
- Poz - pozice lokomotivy v paměti 0 - 7.

K tomu, aby bylo možné přenést celé 8mi bitové číslo při zvoleném zabezpečení dat, je nutné příkazy doplnit parametrem extend, který obsahuje vynechané hodnoty MSB z příkazů. Následující příklad demonstruje vytvoření hodnoty extend bajtu. Rozsvícení červeného světla na semaforech č. 3 a 4 se provede zápisem datové hodnotě val=0xBB na řídicí pozice těchto semaforů. Pro zápis do řídicí jednotky použijeme 8mi bitový zápis - příkaz 0xF2. Protože v příkazu může být na MSB pozici log. „1“ pouze na začátku příkazu (v našem případě 0xF2), nesmí další hodnoty v příkazu obsahovat na MSB log. „1“. Aby bylo možno přenést celé 8mi bitové slovo, byl příkaz doplněn o byte extend, do kterého se vkládá chybějící bit z následujících slov.

Příklad příkazu:

0xF2 - příkaz pro 8bitový zápis do paměti

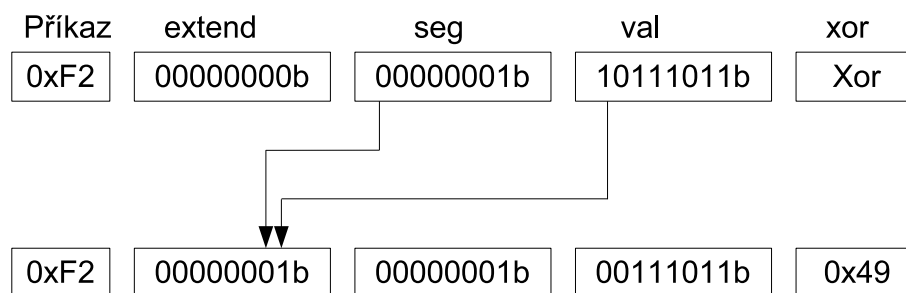
0x01 - extend byte, na jeho LSB pozici je MSB hodnota zapisovaného bajtu

0x00 - pozice semaforů v paměti

0x3B - vysílaná hodnota bez MSB pozice

0x48 - kontrolní XOR všech datových bitů, bez MSB pozice

0xFF - ukončovací znak



Obrázek 6.1: Tvorba extend byte.

Poznámka: U některých příkazů je byte extend redundantní, ale je do příkazu přidán pro dodržení stejné délky příkazu. Hodnota 0x55 u některých příkazů v popisu je důležitá pro vytvoření kontrolního součtu. Tato hodnota může být libovolná s ohledem na podmínku MSB = 0. Nicméně je doporučeno vysílat hodnotu 0x55.

Příkazy

Řízení lokomotiv

0xF0, extend, loconr, speed, chck

- příkaz na změnu rychlosti.

0xF1, extend, poz, addr, chck

- příkaz pro zadání adresy nové lokomotivy (pokud není předdefinována), max. 8 lokomotiv celkem

Příklady:

0xF0,0x00,0x00,0x50,0x20,0xFF

- nastaví lokomotivu č.1 na rychlost 80, směr dozadu

0xF1,0x00,0x01,0x33,0x43,0xFF

- nastaví adresu lokomotivy č.2 na adresu 0x33

Zápis do paměti výstupu

0xF2, extend, seg, val, chck - bajtový zápis do paměti. Datová paměť je rozdělena do 8bit segmentu.

0xF3, extend, seg, val, chck - segmentový zápis do 4bitově rozdělené paměti. Datová paměť je rozdělena do 4bit segmentu. Vhodné pro semaforey.

0xF4, extend, seg, val, chck - bitový zápis do paměti. Datová paměť je rozdělena bitově. Vhodné pro ovládání výhybek.

Příklady:

0xF2,0x01,0x01,0x3B,0x49,0xFF

- na semaforu 3 a 4 se rozsvítí zelená barva

0xF3,0x00,0x02,0x0F,0x7E,0xFF

- zhasne semafor 3.

0xF4,0x00,0x30,0x00,0x44,0xFF - nastaví výhybku č.1 na přímý směr.

Čtení z paměti vstupu

0xF5, extend, seg, chck - 8 bitové čtení z paměti dat.

Procesor vrátí F4, extend,seg,val, chck

0xF6, extend, 0x55, chck - vrátí obsah cele paměti vstupů (proudové snímače, senzory výhybek). Procesor vrátí F5,extend1,val1, val2, val3,val4,extend2, ..., chck

Příklady:

0xF5,0x00,0x01,0x74,0xFF - přečte hodnotu 9-16 proudových snímačů
 0xFA,0x01,0x7B, - potvrzení příjmu 1 příkazu
 0xF5,0x00,0x01,0x74, - na úseku 16 je lokomotiva
 0xFC,0x55,0x29,0xFF - příkazy dokončeny

0xF6,0x00,0x55,0x23,0xFF - Přečte celý obsah paměti vstupů
 0xFA,0x01,0x7B, - potvrzení příjmu 1 příkazu
 0xF6, - začátek odpovědi na příkaz F6
 0x00,0x00,0x01,0x00,0x00 - na úseku 16 je lokomotiva
 0x01,0x00,0x00,0x00,0x7F - úseky 33-53 nejsou obsazené
 0x02,0x6A,0x20,0x2A,0x2A - výhybka č.1 je přestavená, ostatní ne
 0x00,0x00,0x00,0x00,0x00 - neobsazené vstupy
 0x00,0x00,0x00,0x00,0x00 - neobsazené vstupy
 0x00,0x00,0x00,0x00,0x00 - neobsazené vstupy
 0x00,0x00,0x00,0x00,0x00 - neobsazené vstupy
 0x00,0x00,0x00,0x00,0x00 - neobsazené vstupy
 0x75, - kontrolní XOR příkazu F6
 0xFC,0x55,0x29,0xFF - příkazy dokončeny

Čtení z paměti výstupu

0xF7, extend, 0x55, chck - vrátí obsah celé paměti výstupů (semaforů a výhybek)

Příklady:

0xF7,0x00,0x55,0x22,0xFF - přečte celý obsah paměti výstupů.
 FA,0x01,0x7B, - potvrzení příjmu 1 příkazu

0xF7,	- začátek odpovědi na příkaz F7
0x0F,0x5D,0x3B,0x5D,0x5D	- semaforey 3 + 4 svítí zeleně, sem. 1,2,5,6,7,8 červeně
0x0F,0x5D,0x5D,0x7F,0x7F	- semaforey 9-12 svítí červeně, výhybky 0-15 nepřestaveny
0x0F,0x7F,0x7F,0x7F,0x7F	- nulová hodnota, neobsazeno
0x0F,0x7F,0x7F,0x7F,0x7F	- nulová hodnota, neobsazeno
0x0F,0x7F,0x7F,0x7F,0x7F	- nulová hodnota, neobsazeno
0x0F,0x7F,0x7F,0x7F,0x7F	- nulová hodnota, neobsazeno
0x0F,0x7F,0x7F,0x7F,0x7F	- nulová hodnota, neobsazeno
0x0F,0x7F,0x7F,0x7F,0x7F	- nulová hodnota, neobsazeno
0x10,	- kontrolní XOR příkazu F7
0xFC,0x55,0x29,0xFF	- příkazy dokončeny

Systémové příkazy

0xF8,0x00,0x55,chck

- uvede kolejiště do definovaného stavu. Na všech semaforech se rozsvítí červená barva, lokomotivy stojí, všechny výhybky jsou nastaveny na vedlejší kolej.

Příklady:

0xF8,0x00,0x55,0x2D,0xFF - uvede kolejiště do definovaného stavu

Zprávy řídicí jednotky

0xFA,počet přijatých příkazů, chck - potvrzení přijetí paketu s předáním počtu přijatých příkazů

0xFC,0x55,0x29,0xFF - potvrzení o vykonání příkazů

0xFE,0x01,0x7F,0xFF - proběhl úspěšně start řídicí jednotky

0xFE,0x02,0x7C,0xFF - kolejiště není napájeno, tento stav je opakován do doby zapnutí napájení elektroniky kolejiště. Platné při zapnutí/restartování řídicí jednotky.

0xFE,0x03,0x7D - kolejiště není napájeno v průběhu komunikace

0xFE,0x04,0x7A - chyba zápisu do některé jednotky

LocoIO/piny	0 - 7	8 - 15
0x10 Semaforey 1,2,3,4	0	1
0x11 Semaforey 5,6,7,8	2	3
0x12 Semaforey 9,10,11,12	4	5
0x13 Spínače výhybek 0-15	6	7
0x14 Nevyužitě	8	9
0x15 Nevyužitě	10	11
0x16 Nevyužitě	12	13
0x17 Nevyužitě	14	15
0x18 Nevyužitě	16	17
0x19 Nevyužitě	18	19
0x1A Nevyužitě	20	21
0x1B Nevyužitě	22	23
0x1C Nevyužitě	24	25
0x1D Nevyužitě	26	27
0x1E Nevyužitě	28	29
0x1F Nevyužitě	30	31

Tabulka 6.1: Hodnoty seg pro 8mi bitový mód - zápis

LocoIO/piny	0 - 3	4 - 7	8 -11	12 - 15
0x10 Semaforey 1,2,3,4	0	1	2	3
0x11 Semaforey 5,6,7,8	4	5	6	7
0x12 Semaforey 9,10,11,12	8	9	10	11
0x13 Spínače výhybek 0-15	12	13	14	15
0x14 Nevyužité	16	17	18	19
0x15 Nevyužité	20	21	22	23
0x16 Nevyužité	24	25	26	27
0x17 Nevyužité	28	29	30	31
0x18 Nevyužité	32	33	34	35
0x19 Nevyužité	36	37	38	39
0x1A Nevyužité	40	41	42	43
0x1B Nevyužité	44	45	46	47
0x1C Nevyužité	48	49	50	51
0x1D Nevyužité	52	53	54	55
0x1E Nevyužité	56	57	58	59
0x1F Nevyužité	60	61	62	63

Tabulka 6.2: Hodnoty seg pro 4 bitový mód - zápis

LocoIO/piny	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0x10 Semaforey 1,2,3,4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0x11 Semaforey 5,6,7,8	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0x12 Semaforey 9,10,11,12	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0x13 Spínače výhybek 0-15	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
0x14 Nevyužité	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
0x15 Nevyužité	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
0x16 Nevyužité	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
0x17 Nevyužité	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
0x18 Nevyužité	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
0x19 Nevyužité	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
0x1A Nevyužité	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
0x1B Nevyužité	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
0x1C Nevyužité	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
0x1D Nevyužité	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
0x1E Nevyužité	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
0x1F Nevyužité	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Tabulka 6.3: Hodnoty seg pro 1 bitový mód - zápis

LocoIO/piny	0 - 7	8 - 15
0x20 Segmenty kolejiště 1-16	0	1
0x21 Segmenty kolejiště 17-32 5,6,7,8	2	3
0x22 Segmenty kolejiště 33-48	4	5
0x23 Segmenty kolejiště 49-53	6	7
0x24 Snímače výhybek 0-7	8	9
0x25 Snímače výhybek 8-15	10	11
0x26 Nevyužité	12	13
0x27 Nevyužité	14	15
0x28 Nevyužité	16	17
0x29 Nevyužité	18	19
0x2A Nevyužité	20	21
0x2B Nevyužité	22	23
0x2C Nevyužité	24	25
0x2D Nevyužité	26	27
0x2E Nevyužité	28	29
0x2F Nevyužité	30	31

Tabulka 6.4: Hodnoty seg pro 8mi bitový mód - čtení

LocoIO/piny	0 - 3	4 - 7	8 -11	12 - 15
0x20 Segmenty kolejiště 1-16	0	1	2	3
0x21 Segmenty kolejiště 17-32	4	5	6	7
0x22 Segmenty kolejiště 33-48	8	9	10	11
0x23 Segmenty kolejiště 49-53	12	13	14	15
0x24 Snímače výhybek 0-7	16	17	18	19
0x25 Snímače výhybek 8-15	20	21	22	23
0x26 Nevyužité	24	25	26	27
0x27 Nevyužité	28	29	30	31
0x28 Nevyužité	32	33	34	35
0x29 Nevyužité	36	37	38	39
0x2A Nevyužité	40	41	42	43
0x2B Nevyužité	44	45	46	47
0x2C Nevyužité	48	49	50	51
0x2D Nevyužité	52	53	54	55
0x2E Nevyužité	56	57	58	59
0x2F Nevyužité	60	61	62	63

Tabulka 6.5: Hodnoty seg pro 4 bitový mód - čtení

Poznámka:

- Jeden semafor zabírá v paměti 4 bity, aby LED svítla je nutné na požadovanou pozici zapsat log. „0“, led jsou řazeny: horní oranžová, červená, zelená, dolní oranžová.
- Spínač výhybek je jednobitová hodnota. Při zapsání log. „0“ odpovídá přímý směr na výhybce.
- Snímač výhybek je dvou bitový. Přímému směru odpovídá hodnota 01, zatáčky 10.

Kapitola 7

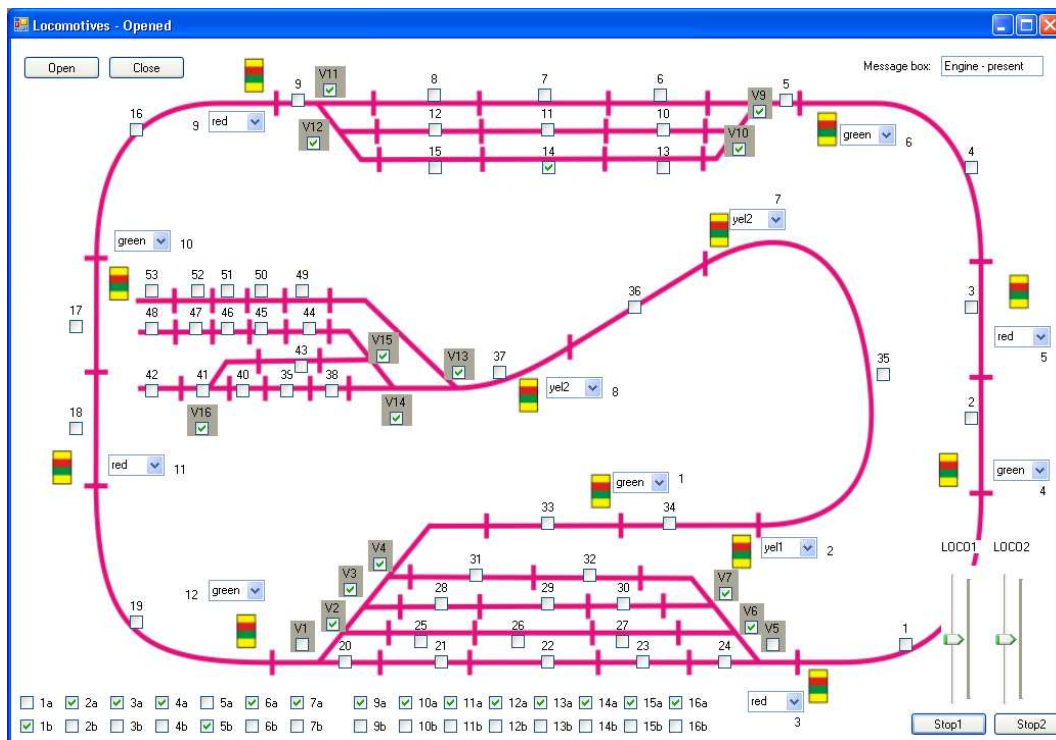
Řídicí aplikace - Locomotives

Pro ověření funkčnosti modelu železnice byla vytvořena aplikace Locomotives v jazyce C# .NET. Pomocí této aplikace je možné monitorovat stav kolejiště a ovládat jeho periferie. Aplikace je napsána v anglickém jazyce.

Navázání komunikace mezi aplikací a modelem železnice se provede pomocí tlačítka Open. Za předpokladu, že je vše v pořádku, změní se popis aplikace na Locomotives-opened a v políčku Message box se objeví nápis Engine - present. V opačném případě jsou uživateli zobrazeny chybové hlášky. V případě, že inicializace kolejiště proběhne v pořádku a na kolejišti není umístěna žádná lokomotiva, je do pole Message box opakovaně tisknuta zpráva "Engine lost" se zvyšujícím se číslem, které znamená počet zaznamenaných ztrát kontaktu s lokomotivami. Toto číslo se zvyšuje s každým přečteným stavem kolejiště z řídicí jednotky. Pro ukončení komunikace s modelem železnice slouží tlačítko "Close". Kliknutím na tlačítko "Close" se provede bezpečné ukončení komunikace s řídicí jednotkou modelu železnice.

Obrázek kolejiště zobrazuje jednotlivé úseky kolejiště. Na těchto úsecích se nacházejí číslovaná zaškrťovací pole s průhledným pozadím, které reprezentují přítomnost lokomotivy na úseku odpovídajícím číslu pole. Zaškrtnuté pole znamená přítomnost lokomotivy na daném úseku. Tyto pole není možné editovat. Dále jsou na jednotlivých úsecích umístěna šedá zaškrťovací pole s číslem, které znázorňují požadovaný stav přestavení výhybky na úseku, kde jsou umístěny. Zaškrtnuté pole znamená přehození výhybky na vedlejší kolej. Tyto pole je možné editovat. Semaforey jsou v aplikaci znázorněny pomocí obrázku semaforů, vedle kterých se nalézají pole pro výběr barvy semaforu. Ve spodní části ovládací aplikace jsou umístěna zaškrťovací políčka zobrazující stav výhybek. Každá výhybka je znázorněna dvojicí zaškrťvajících polí, které mají stejné číslo jako výhybka. V pravém dolním rohu aplikace jsou umístěné dva posuvníky a tlačítka Stop1 a Stop2,

které slouží k ovládání rychlosti a směru dvou lokomotiv. Nulová rychlost lokomotiv odpovídá střední pozici posuvnému, případně ji lze nastavit tlačítkem Stop1 nebo Stop2.



Obrázek 7.1: Aplikace Locomotives

Kapitola 8

Závěr

Řešení bakalářské práce bylo rozděleno do tří na sebe navazujících částí. V první části byly identifikovány problémy s řízením modelu železnice. Dále byl proveden návrh a částečná realizace nového řídicího systému. Ve druhé části proběhla samostatná náročná přestavba modelu. V poslední části byl vytvořen program pro řídicí jednotku a zároveň aplikace Locomotives.

Jak již bylo zmíněno, v první části byly identifikovány problémy stávající řídicí jednotky a celého modelu železnice. V závislosti na těchto problémech bylo po konzultaci s Ing. Radkem Šindelářem rozhodnuto o téměř kompletní přestavbě hardware modelu železnice. Hlavní řídicí jednotka byla vyměněna a pro model železnice byl využit sběrníkový systém LocoNet firmy DigiTrax [1] a sběrnice DCC [7], které jsou využity v moderních elektronicky řízených modelech. Díky tomuto je možné celý model rychle rozšiřovat. V současné době je hlavní řídicí jednotka schopna ovládat celkem 8 lokomotiv a 32 modulů LocoIO (16 vstupních modulů a 16 výstupních modulů), není ale problém rozšířit počet modulů LocoIO nebo počet lokomotiv. Moduly LocoIO byly upraveny a převzaty od belgického železničního modeláře Hanse Deloofa [3].

Při přestavbě modelu železnice byla z modelu kvůli špatné orientaci v kabeláži vyňata téměř celá elektroinstalace. Dále byly do modelu zabudovány nové komponenty modelu pro síť LocoNet a byly upraveny ponechané komponenty pro propojení s moduly LocoIO. Na závěr přestavby hardware modelu kolejiště byla do modelu dána nová elektroinstalace. Samostatná přestavba kolejiště trvala přibližně 14 dní prací a v modelu je přibližně 200m kabelů zajišťujících řízení modelu kolejiště a komunikaci sítě LocoNet.

Program v řídicí jednotce zajišťuje komunikaci s nadřazeným systémem, zpracování a vykonání jeho příkazů. Program dále provádí generování DCC signálu pro lokomotivy a obsluhu sběrnice LocoNet. Řídicí jednotka komunikuje s nadřazeným systémem

pomocí nově definovaného komunikačního protokolu a v případě, že od nadřazeného systému nedostane do definované doby pokyny pro řízení modelu, řídicí jednotka uvede model železnice do inicializačního stavu. Při generování DCC signálu je kladen důraz na přesnost definovaných pulzů. V průběhu vytváření kódu obsluhy sítě LocoNet byl zjištěn nedostatečný popis této sběrnice firmou Digitrax což následně vedlo k problémům a neodhalení chyby při úpravě jednotek LocoIO. Díky této skutečnosti jsou v modulech LocoIO prohozeny některé piny. V příloze tohoto dokumentu je již opravené zapojení LocoIO a na přiloženém CD se nacházejí obě verze zapojení.

Přestavba a vývoj nové řídicí jednotky modelu byla sponzorována projektem CEPOT (Centrum podpory talentovaných studentů), kde byl projekt speciálně odměněn za mimořádné pracovní nasazení. Přestavený model železnice je již rok využíván ve výuce předmětu „Návrhy automatizovaných zařízení“ a nově i v předmětu „Řídicí systémy“. Za tuto dobu byl celý model důkladně otestován.

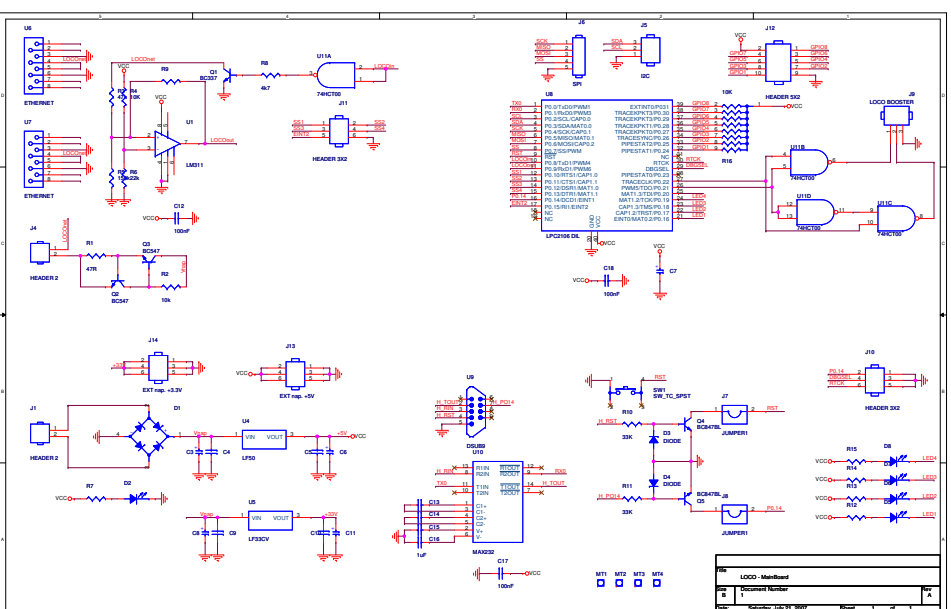
Literatura

- [1] **Digitrax**, Společnost zabývající se železničnými modely, *[online]* .
<http://www.digitrax.com/>.
- [2] **LocoNet**, Norma LocoNet, říjen 1997 *[online]* .
<http://www.digitrax.com/ftp/LOCOPE1.PDF>.
- [3] **Hans DEloof, HDL**, Železniční modelář *[online]* .
<http://users.pandora.be/deloof/>.
- [4] **National Model Railroad association**, *[online]* .
<http://www.nmra.org/>.
- [5] **National Model Railroad association, DCC Standard S-9.1, Poslední re-vize červenec 2004** *[online]* .
http://www.nmra.org/standards/DCC/standards_rps/S-91-2004-07.pdf.
- [6] **National Model Railroad association, DCC Standard S-9.2, Poslední re-vize červenec 2004** *[online]* .
http://www.nmra.org/standards/DCC/standards_rps/S-92-2004-07.pdf.
- [7] **National Model Railroad association, DCC - Introduction** *[online]* .
<http://www.nmra.org/beginner/dccbasic.html>.
- [8] **Acorn**, Výrobce jádra ARM7 *[online]* .
<http://www.arm.com/>.
- [9] **www.mcu.cz**, Server a e-shop zabývající se problematikou programování mikroprocesorů *[online]* .
<http://www.mcu.cz/>.
- [10] **uVisionV3**, Vývojové prostředí firmy Keil *[online]* .
<http://www.keil.com/arm/>.

- [11] **GM electrinic** , Katalog součástek pro elektrotechniku, 2006 [*Online*] .
<http://www.gme.cz/>.
- [12] **Ing. Dušan Havlík**, **Model železnice**, Diplomová práce Ing. Dušana Havlíka -
Model železnice, 2004 [*PDF*]
- [13] **LPC2106**, Katalogový list mikroprocesoru LPC2106 od firmy Philips, září 2003
[*PDF*] .
- [14] *Trevor Martin BSc. CEng. MIEE* **The Insider's Guide To The Philips
ARM7® based Microcontrollers** , Literatura pro programování mikroprocesorů
řady LPC21xx, únor 2006 [*PDF*] .

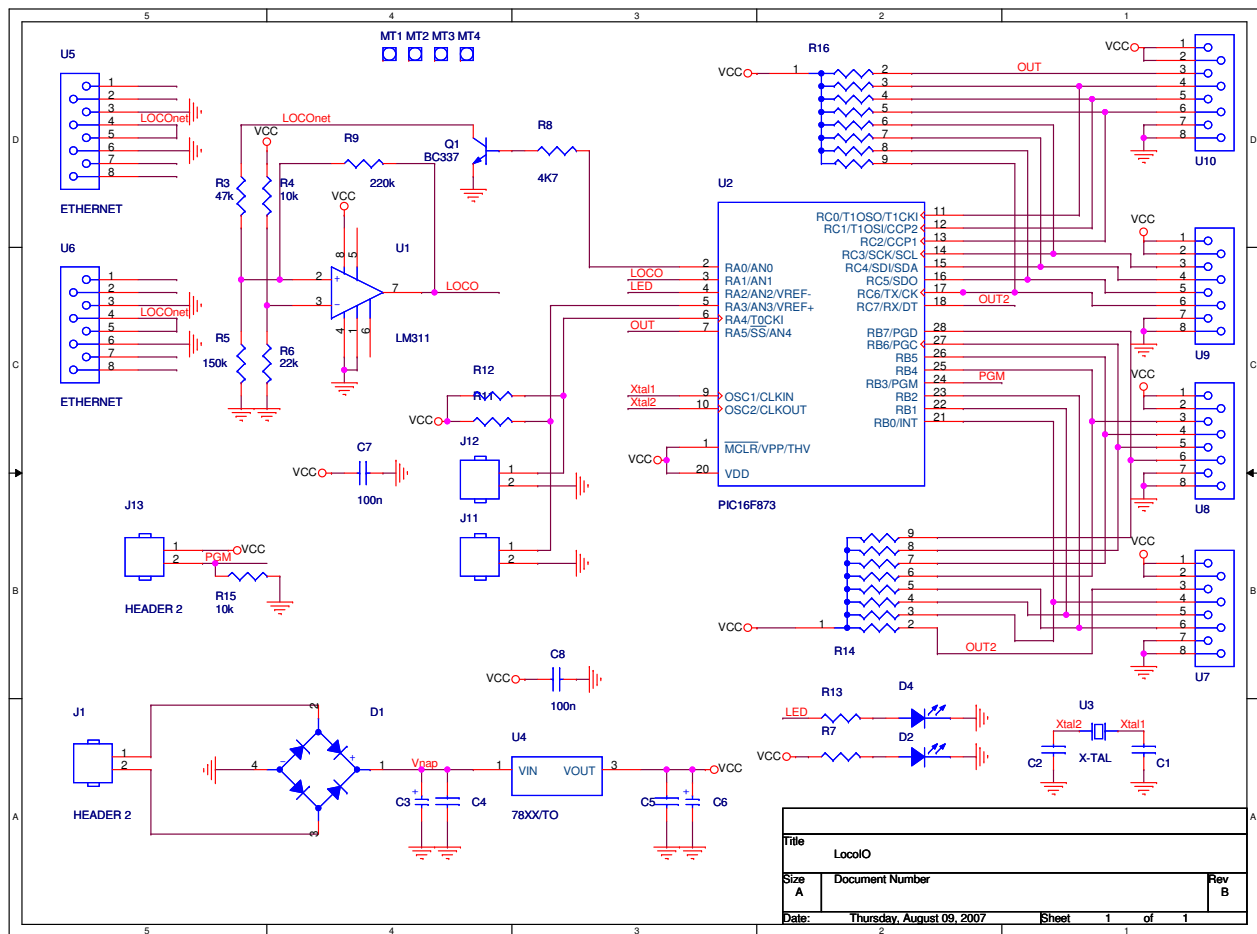
Příloha A

Schématická zapojení

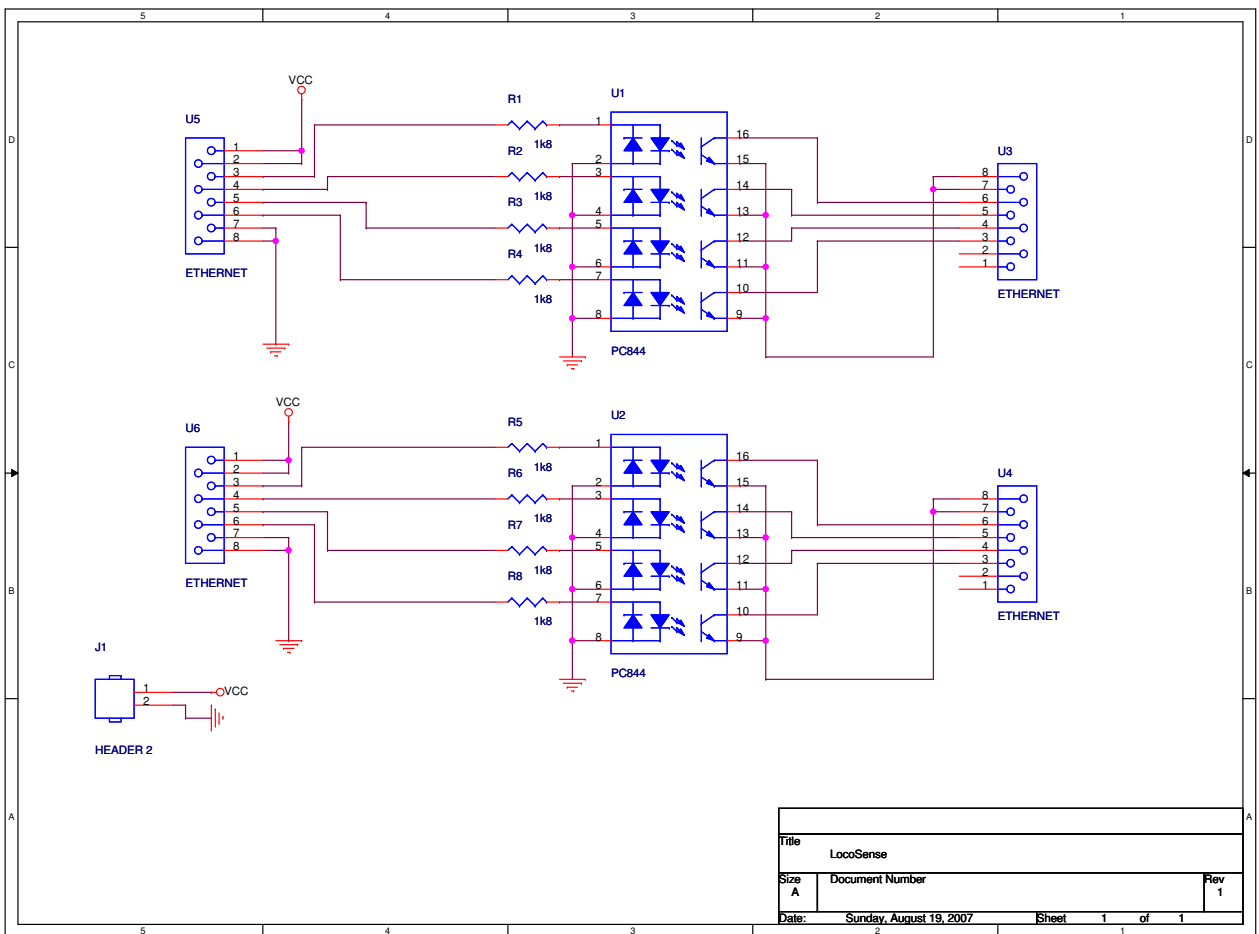


Obrázek A.1: Modul Řídicí jednotky

Obrázek A.2: Modul LocoIO





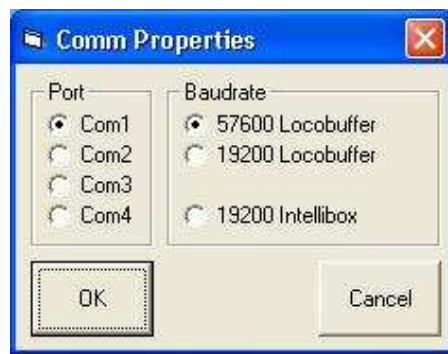


Obrázek A.4: Modul Snímače optické oddělení

Příloha B

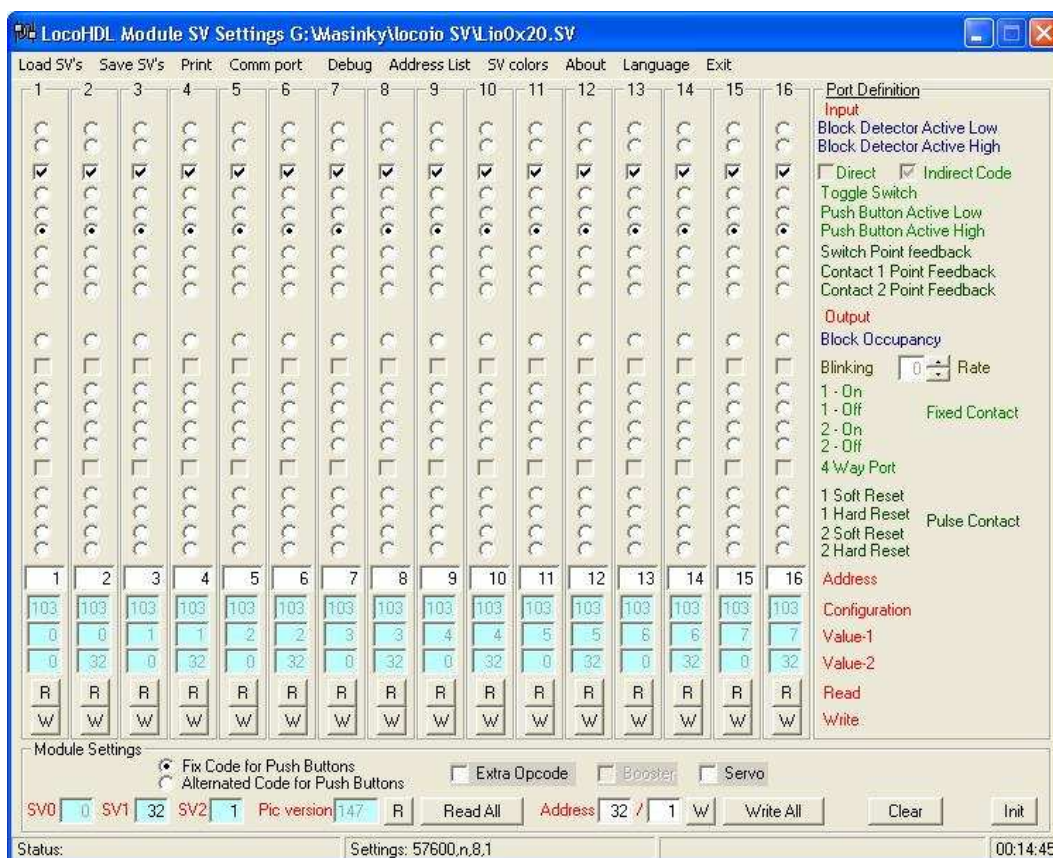
Konfigurace modulu LocoIO

Ke konfiguraci LocoIO modulů je potřebný modul LocoBuffer a SW aplikace locoHDL. Popis této aplikace je velmi dobře zdokumentován na stránkách HDL [4]. Pro to zde bude uveden pouze stručný popis aplikace a naprogramování LocoIO na vstupní bránu. Při prvním spuštění aplikace je nutné v zobrazeném okně (viz. obr. B.1) zvolit sériový port a komunikační rychlost, na kterou je modul LocoBuffer nakonfigurován. V současném provedení je modul nakonfigurován na komunikační rychlost 57600Bd a je připojen na sériový port číslo 1.



Obrázek B.1: Zapojení konektoru pro LocoNet

Po nastavení komunikačních parametrů se zobrazí konfigurační okno jednotky LocoIO (viz. obr. B.2). V tomto okně lze nastavit jednotlivě parametry pro každý vstupně-výstupní pin jednotky.



Obrázek B.2: Zapojení konektoru pro LocoNet

V dolní části obrázku jsou položky SV0, SV1 a SV2 pro výběr LocoIO modulu, který chceme konfigurovat. Položka SV0 udává síť a nelze ji měnit. Položky SV1 a SV2 udávají adresu LocoIO modulu. Položka SV1 udává číslo jednotky v decimálním formátu (čísla napsaná na LocoIO modulech jsou v hexadecimálním formátu), SV2 udává číslo podmodulu - to je u všech modulů nastaveno na 1. Tlačítko R ověří dostupnost tohoto modulu a načte verzi firmware. Při úspěšném načtení hodnot políčka zezelenej a adresa modulu se zkopíruje do políček Address. Tlačítko Read All přečte aktuální konfiguraci LocoIO modulu.

Pokud chceme změnit adresu modulu musíme nejdříve zadat jeho adresu do políček SV1, SV2 a ověřit dostupnost tohoto modulu. Poté je nutné změnit políčka v Address na požadovanou hodnotu a stisknout tlačítko W. Tlačítko Write All slouží k zapsání celé konfigurace do LocoIO modulu.

Pro inicializaci nového modulu slouží tlačítko Init. Pro inicializaci modulu je nutné odpojit od jednotky LocoBuffer celou síť LocoNet kromě jednotky LocoIO, kterou chceme

konfigurovat. Jestliže se tak nestane budou inicializovány všechny jednotky na síti LocoNet. Po této proceduře má jednotka adresy $SV1 = 81$ a $SV2 = 1$.

V horní části okna je výběr konfigurace každého pinu. Každému pinu je přiřazena unikátní adresa v síti LocoNet (kromě speciálních případů), v použitém příkladu v rozsahu 1-16.

V případě konfigurace pinu jako vstupu, je nutné mít zaškrtnuto položku "Indirect code". Zaškrtnutí "Indirect code" zapříčiní vyslání informace o změně stavu na vstup řídicí jednotce a ne jiné LocoIO jednotce. Jestliže je vstup konfigurován v pozitivní logice (označení položky "Push Button Active High"), log.1 odpovídá +5V a log.0 odpovídá 0V. Pokud by byla požadována opačná logika musí se označit položka "Push Button Active Low".

Při konfiguraci pinu jako výstupu, je možné nadefinovat jeho počáteční hodnotu zaškrtnutím některé z následujících možností:

- 1-on - po zapnutí napájení je na výstupu +5V
- 1-off - po zapnutí napájení je na výstupu 0V
- 2-on - po vyslání příznaku Global power OFF je na výstupu +5V
- 2-off - po vyslání příznaku Global power OFF je na výstupu 0V

Vstupům a výstupům je možné přiřadit i další funkce, které ale v modelu nejsou použity. V případě, že výstupy obsahují nenulovou adresu, je možné je řídit přímo pomocí jednoduchého příkazu (v kolejišti této možnosti není využito, protože nemají žádnou zpětnou vazbu). Pokud by se tato adresa shodovala s adresou některého ze vstupů a nebyl by zaškrtnut mód Indirect, bude vstup ovládat přímo výstup s touto adresou bez zásahu řídicí jednotky. Každou z těchto voleb je možné zapsat zvlášť do LocoIO modulu.

Aplikace LocoHDL umožňuje sledování sítě LocoNet pomocí volby Debug v horním menu. Po kliknutí na Debug se otevře nové okno do něhož se provádí výpis paketu sítě LocoNet. Nejsou zde však vypisovány všechny chyby sítě LocoNet, vypisují se zde pouze chyby se špatným kontrolním součtem! Při programování řídicí jednotky jsem zjistil, že některé pakety chybně označí za vadné. Dále je možné jednotlivé nastavení jednotek zálohovat a zpětně nahrávat.

Během práce s tímto SW jsem zaznamenal jeho nestabilitu. Pokud při komunikaci s modulem nastane chyba, aplikace se nesnaží o znovu vyslání paketu a zapisovaný byte přeskočí. Toto je indikováno zčervenáním políčka value 1,2 nebo configuration u příslušného aktuálně zapisovaného pinu. Také při závažnější chybě komunikace ztratí informace o na-

stavení pinů (vynulují se konfigurační hodnoty pinů) a je nutné v nejhorším případě znova inicializovat jednotku a znova ji naprogramovat.

Příloha C

Seznam zkratk

DCC	- Digital Command Control
ARM7	- Acron Risc Machine verze 7
NMRA	- National Model Railroad Association
UART	- Universal Asynchronous Treciever Transmitter
TTL	- Tranzistor tranzistor Logic
CMOS	- Complementary Metal Oxide Semiconductor
pc1k	- Processor Clock

Příloha D

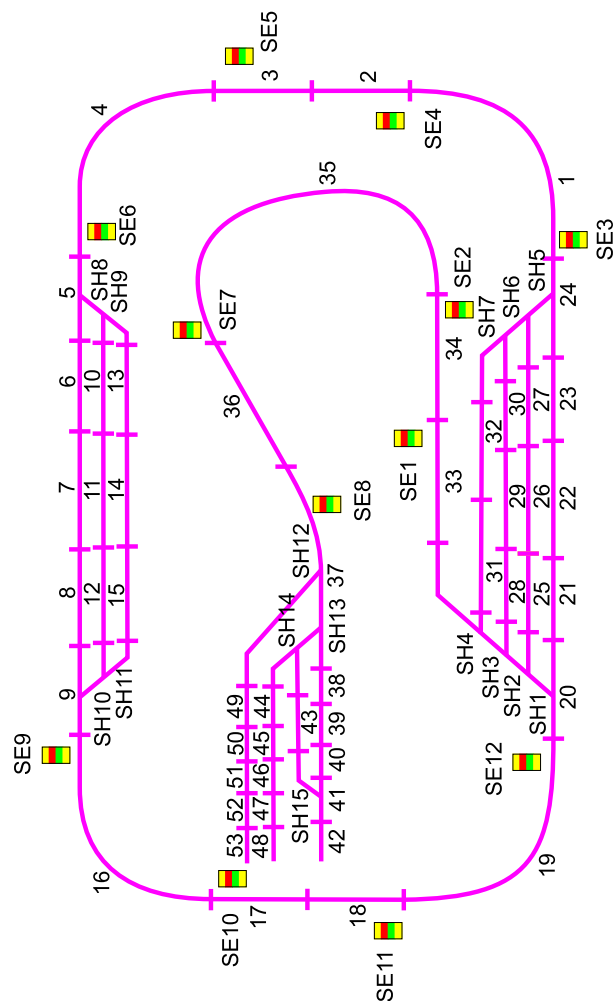
Obsah přiloženého CD

K této práci je přiloženo CD, na kterém jsou uloženy zdrojové kódy aplikací, katalogové listy a schémata k jednotlivým DPS.

- datasheets: katalogové listy mikroprocesorů a popisy standardů
- orcad: schémata jednotlivých DPS
- sources: zdrojové kódy aplikací (Locomotives, LaTeX, CPU)
- text: text této bakalářské práce

Příloha E

Nákres kolejiště



Obrázek E.1: Nákres kolejiště