

Inovace modelu železnice a jeho řízení pomocí programovatelného automatu

Michal Kalandřík

Bakalářská práce
2019



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2018/2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal Kalandřík**
Osobní číslo: **A16199**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Inteligentní systémy s roboty**
Forma studia: **prezenční**

Téma práce: **Inovace modelu železnice a jeho řízení pomocí programovatelného automatu**

Téma anglicky: **The Innovation of a Railroad Model and its Control via a Programmable Logic Controller**

Zásady pro vypracování:

1. Vypracujte literární rešerši na téma týkající se způsobu řízení, komunikace a možností programovatelných automatů.
2. Navrhňte inovaci modelu železnice a to jak po stránce samotného modelu, tak i po stránce ovládacích a řídicích zařízení. Proveďte realizaci tohoto návrhu.
3. Pro upravený model vytvořte ilustrativní příklady, včetně vizualizace daného procesu ve vybraném SCADA/HMI systému. Ověřte také možnost vzdáleného řízení a monitorování modelu.
4. Vytvořte samostatnou dokumentaci obsahující popis jednotlivých částí upraveného modelu a jeho propojení s okolím.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. PLC a automatizace 1: základní pojmy, úvod do programování. Praha: BEN – technická literatura, 2002. ISBN 80-86056-58-9.
2. ŠMEJKAL, Ladislav. PLC a automatizace: Sekvenční logické systémy a základy fuzzy logiky. Praha: BEN – technická literatura, 2005. ISBN 80-730-0087-3.
3. DCC Communications Standard [online]. 2004 [cit. 2018-11-26]. Dostupné z: <https://www.nmra.org/sites/default/files/s-92-2004-07.pdf>
4. VINCENEC, Jiří. Řízení modelu vlakového kolejiště pomocí programovatelného automatu [online]. Zlín, 2011 [cit. 2018-11-27]. Dostupné z: <http://hdl.handle.net/10563/16810>. Bakalářská práce. Univerzita Tomáše Bati. Vedoucí práce Pavel Navrátil.
5. MORAVSKÉ PŘÍSTROJE. Co je Control Web? [online]. 2010 [cit. 2018-11-27]. Dostupné z: <https://www.mii.cz/art?id=380&lang=405>

Vedoucí bakalářské práce:

Ing. Pavel Navrátil, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

21. prosince 2018

Termín odevzdání bakalářské práce:

15. května 2019

Ve Zlíně dne 21. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Jméno, příjmení: Michal Kalandřík

Název bakalářské práce: Inovace moelu železnice a jeho řízení pomocí programovatelného automatu

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 – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- 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, dne

Michal Kalandřík v.r.
podpis diplomanta

ABSTRAKT

Cílem této bakalářské práce je inovace modelu kolejiště pro výuku předmětu programovatelných automatů. Po analýze současného stavu, bylo rozhodnuto vytvořit nový model. Nový model nebude řízen analogově, jak tomu bylo doposud, ale bude použit digitální způsob řízení pomocí standardu DCC. Pro řízení kolejiště bude použitý programovatelný automat (PLC) od firmy Teco. Na nižší úrovni bude použit mikrokontrolér ATmega 328, který bude komunikovat s PLC automaty přes sériovou linku.

Klíčová slova:

PLC, model, DCC standard, Atmega 328P, SCADA

ABSTRACT

Abstrakt ve světovém jazyce

The aim of this bachelor thesis is the innovation of a railroad model used for teaching purposes for PLC programming course. It was decided to create a new model after its current state was analyzed. The new model would not be controlled using analog methods as it was before, but the new digital way of controlling using DCC standard would be used. PLC of Teco company will be used to control the railroad model. Microcontroller ATmega 328 would be used for low-level communications with the PLCs through serial port.

Keywords:

PLC, model, DCC standar, Atmega 328P, SCADA

Děkuji mému vedoucímu Ing. Pavlu Navrátillovi, Ph.D. za návrhy a připomínky a pomoc při řešení této práce.

Dále bych chtěl poděkovat Ing. Petru Dostálkovi, Ph.D. za pomoc s výrobou desky plošných spojů.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 PROGRAMOVATELNÉ AUTOMATY	10
1.1 ROZDĚLENÍ PLC PODLE PROVEDENÍ.....	10
1.1.1 Mikro PLC	10
1.1.2 Kompaktní PLC	10
1.1.3 Modulární PLC.....	11
1.2 STRUKTURA PLC	12
1.3 PROGRAM PLC.....	13
1.4 PROGRAMOVACÍ JAZYKY PLC.....	14
1.5 KOMUNIKACE S OKOLÍM	16
1.5.1 Sériová komunikace RS-232C	16
1.5.2 Sériová komunikace RS-422 a RS-485.....	16
1.5.3 ETHERNET	17
2 MODEL KOLEJIŠTĚ	21
2.1 SOUČASNÝ STAV	21
2.2 NOVÝ MODEL	22
3 DIGITÁLNÍ ŘÍZENÍ MODELOVÉ ŽELEZNICE	23
3.1 DCC.....	23
4 ŘÍDICÍ ELEKTRONIKA	25
4.1 ŘÍDICÍ ELEKTRONIKA.....	25
4.2 MIKROKONTROLÉR	25
4.2.1 RISC.....	25
4.2.2 CISC.....	25
5 SCADA/HMI	26
5.1 CONTROL WEB.....	26
II PRAKTICKÁ ČÁST	27
6 KONSTRUKCE	28
7 HARDWARE	29
7.1 PLC TECOMAT	29
7.2 ATMEGA328	31
8 ŘÍDICÍ ELEKTRONIKA	32
8.1 SNÍMÁNÍ OBSAZENÍ ÚSEKŮ	32
9 SOFTWARE	33

9.1	PROGRAM PRO ATMEGA 328.....	33
9.2	PROGRAM PRO PLC.....	40
ZÁVĚR		43
SEZNAM POUŽITÉ LITERATURY.....		44
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		46
SEZNAM OBRÁZKŮ		48
SEZNAM PŘÍLOH.....		50

ÚVOD

Předmětem této bakalářské práce je vytvoření nového modelu vlakového kolejiště. Ten by měl sloužit jako učební pomůcka při výuce programovatelných automatů (dále jen PLC). Schopnost pracovat s PLC je pro studenty automatizace a řídicí techniky důležitá, jelikož při robotizaci průmyslu je mnohdy nezbytné využívat různé dopravníky, otočné stoly a jiné stroje, které jsou řízeny právě pomocí PLC.

Původním cílem této bakalářské práce byla inovace modelu, který zkonstruoval Ing. Vladimír Hnilica v rámci své bakalářské a diplomové práce. Současná podoba tohoto modelu je pak výsledkem řady úprav, které byly prováděny studenty fakulty v rámci jejich kvalifikačních prací. Po bližším prozkoumání současného stavu modelu, byl cíl práce změněn a bylo rozhodnuto vytvořit zcela nový model, tj. model, který bude využívat moderních technologií. Jako příklad je možno uvést využití lokomotiv vybavených digitálním dekodérem.

Struktura práce zachovává klasické členění na části teoretickou a praktickou.

Teoretická část práce se zabývá jednotlivými druhy PLC automatů a jejich strukturou. Dále pak pojednává o možnostech jejich programování a komunikace. Následuje kapitola o současném stavu modelu kolejiště a způsobu jeho řízení. V závěru teoretické části jsou pak předloženy informace o řídicí elektronice a možnostech vizualizace.

Začátek praktické části je věnován návrhu konstrukce modelu. Dále je uveden popis PLC automatu CP-10x5 a mikrokontroléru ATMega 328P, které jsou použity pro řízení modelu kolejiště. V následující kapitole je pak vysvětlen návrh řídicí desky pro výhybky a taktéž způsob snímání pohybu vlaku. Na závěr jsou uvedeny informace o programu pro mikrokontrolér ATMega 328P, který se stará o modulaci signálu a taktéž informace o programu pro PLC automat, jenž řídí kolejiště.

I. TEORETICKÁ ČÁST

1 PROGRAMOVATELNÉ AUTOMATY

Programovatelné automaty (PLC-Program Logic Controller) můžeme dělit podle více kritérií. Z hlediska provedení může druh PLC rozdělit na mikro PLC, kompaktní PLC a modułární.

1.1 Rozdělení PLC podle provedení

PLC automaty lze rozdělit podle jejich konstrukce do tří skupin.

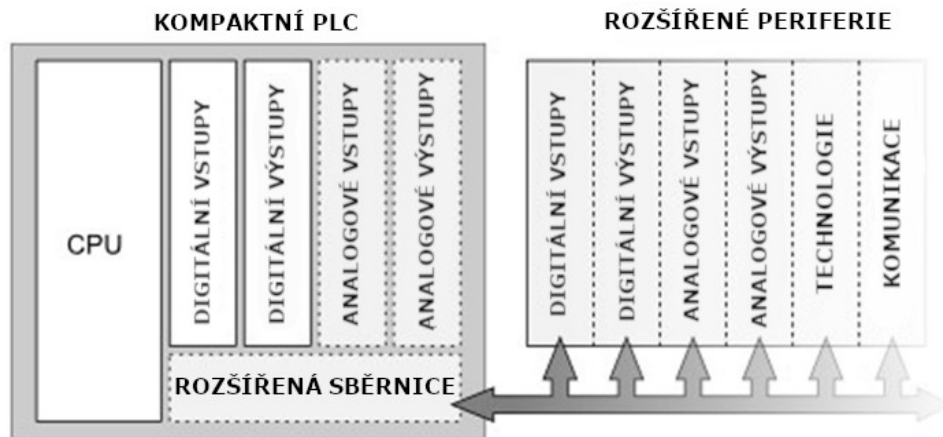
1.1.1 Mikro PLC

Mikro PLC se vyznačuje svými malými rozměry a nízkou cenou. Má pevně daný počet vstupů a výstupů, které není možné dále rozšiřovat. Často zde chybí možnost komunikace s okolím. Programové možnosti jsou redukovány na nejjednodušší minimum. Díky svým malým rozměrům a nízkou cenou se řadí mikro PLC do kategorie spotřební elektroniky. Tyto automaty jsou použity pro jednodušší úlohy, které se tradičně řešili reléovou logikou [1].

1.1.2 Kompaktní PLC

Kompaktní PLC má integrovaný CPU (Central Processing Unit), digitální a analogové vstupy a výstupy. Jsou v jednom modulu. Některé z digitálních vstupů mohou být použity pro HW přerušování, měření frekvence a vstupu rychlých čítačů. Digitální výstupy mohou být realizovány pomocí reléových kontaktů nebo pomocí tranzistorů. U tranzistorových výstupů bývá možnost řídit i PWM (Pulse Width Modulation) složku [2].

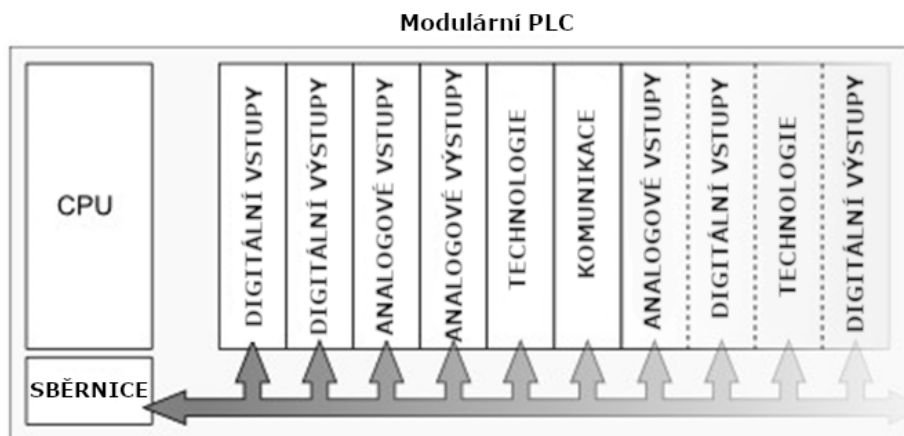
Výhodou kompaktního PLC je jejich rychlost přístupu k perifériím a jejich pořizovací cena. Celková doba cyklu se může pohybovat řádově v jednotkách milisekund. Kompaktní PLC jsou určeny pro malé řídicí systémy, převážně jako centrální systém. Mají menší paměť pro program i data a u některých není možnost rozšíření dalšími I/O perifériemi. Ale i tato možnost rozšíření je omezená na jednotky až desítky modulů [2].



Obrázek 1.1 Schéma kompaktního PLC [2]

1.1.3 Modulární PLC

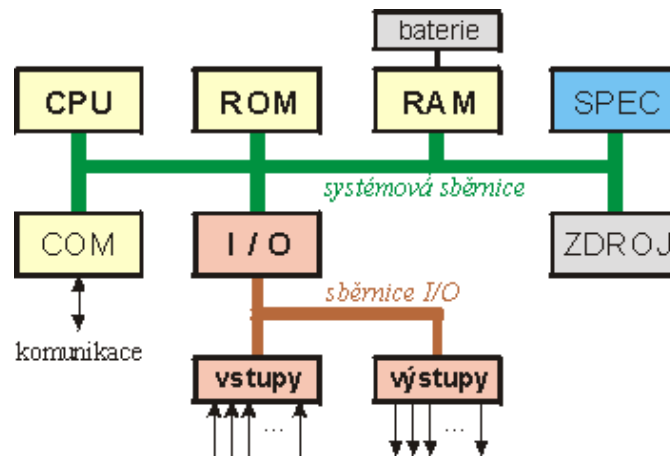
Modulární PLC se skládá z jednotlivých modulů, z čehož jeden modul musí být CPU. Jsou určeny pro větší systémy, proto jsou vybaveny větší pamětí pro program a data. Pomocí jednotlivých modulů (Ethernet, RS485, rychlé čítače, ...) tak lze sestavit PLC automat přesně odpovídající daným požadavkům. Přídavné moduly jsou spolu propojeny komunikační sběrnici. Díky této modularitě lze PLC automat snadno upgradovat [2].



Obrázek 1.2 Schéma modulárního PLC [2]

1.2 Struktura PLC

PLC se dá rozdělit do tří bloků. Blok pro zpracování informací, blok obsahující vstupy/výstupy a blok s pamětí. Všechny bloky jsou spolu spojeny sběrnici.



Obrázek 1.3 Bloková struktura PLC [3]

CPU čte z operační paměti hodnoty vstupů a výstupů a provádí s nimi logické nebo numerické operace podle programu. Výsledek operace uloží zpátky do operační paměti. Všechny tyto úkony se dějí mezi registry. Pro činnost CPU je důležitý obzvláště registr PC (Program Counter), do kterého se ukládá adresa následující instrukce, kterou bude CPU provádět v dalším kroku. Druhý důležitý registr je *Akumulátor* do, kterého je načtena hodnota registru se kterým se má provádět logické operace. Druhá hodnota je v pracovním registru. Výsledná operace se zapíše zpátky do akumulátoru.

Paměť ROM (Read Only Memory) je paměť, která není závislá na napájení. Tato paměť obsahuje programy potřebné pro činnost automatu. Tyto programy jsou dané výrobcem a uživatel je nemůže měnit.

Paměť RAM (Random Acces Memory) je dočasná paměť, která je závislá na napájení. Je určena pro čtení i zápis. Je zde uložen také program podle něhož CPU pracuje.

Digitální I/O mohou nabývat pouze dvou hodnot a to buď log.0 nebo log.1. Musí být galvanicky odděleny od centrální jednotky pomocí optočlenů. Vstupy musí mít ochranu proti přepětí a výstupy musí mít ochranu proti zkratu. Spínání výstupů je zajištěno pomocí relé, tranzistorů nebo triaků.

Analogové I/O jsou realizovány A/D (D/A) převodníky. A/D (Analog-Digital) převodníky slouží pro převod analogového signálu (v čase spojitého) na digitální signál. Obvykle se

využívá 8 nebo 12 bitový převodník. Přesnost převodu je daná rozlišením převodníku (obvykle 8 nebo 12 bit). D/A (Digital-Analog) výstup převádí číslo na spojitý výstup napětí. Přesnost je opět dána rozlišením převodníku (8 nebo 12 bit).

Komunikační jednotka zajišťuje komunikaci se vzdálenými moduly vstupů a výstupů. Dále může předávat informace nadřazeným a podřazeným řídicím systémům. Některé moduly umožňují i bezdrátový přenos informací přes modem, radiomodem nebo přes Wi-Fi. Komunikace však může probíhat i pomocí RS232, RS422 nebo RS485.

[3]Chyba! Nenalezen zdroj odkazů.

1.3 Program PLC

Program PLC je tvořen posloupností příkazů, které se cyklicky vykonávají ve smyčce. Při programování se však smyčky netvoří, to zajistí systémový program. Vytvoření smyčky v programu může naopak způsobit chybu a následný pád programu. Cyklické vykonávání programu tzv. otočka cyklu se dá rozdělit na několik částí. Jako první se provedou režijní operace, naplánování aktivace procesů pro další cyklus... Dále se aktualizují hodnoty vstupů. Následně se tyto hodnoty uloží do registru vstupů. Tím se vytvoří tzv. obraz vstupů. Následně se začne provádět uživatelský program. V programu se pracuje s obrazy vstupů a výstupů. Proto, když dojde ke změně vstupu během provádění programu, budou tyto změny ignorovány. Tak se zajistí synchronizace vstupních a výstupních dat během programu. Stejně tak je pozastavena komunikace po sériové lince nebo i časové údaje. Nakonec po provedení uživatelského programu se zapíše obraz výstupů na piny PLC. Celý cyklus se vykonává ve smyčce [1].



Obrázek 1.4 Otočka cyklu programu [1]

1.4 Programovací jazyky PLC

Podle normy IEC 61131-3 je možnost programovat PLC automat pěti programovacími jazyky. Tyto jazyky můžeme rozdělit na dvě skupiny a to na textové a grafické.

Jazyk IL (Instruction List) je textový programovací jazyk. Program se skládá z instrukcí napsaných na jednotlivých řádcích. Tento jazyk připomíná programování v assembleru.

```
LD VstupA           //načtení VstupuA
AND VstupB          //VstupA AND VstupB
ST Vystup           //Uložení výsledku do výstupu
```

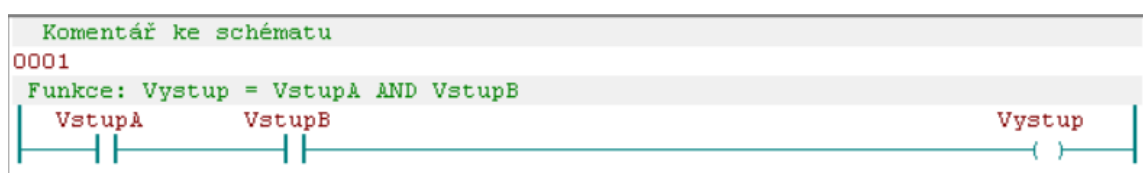
Obrázek 1.5 Příklad programového kódu v jazyce IL

Jazyk ST (Structured Text) je výkonnějším programovacím jazykem než IL. Základ má v programovacím jazyku Pascal a C. Příkazy jsou od sebe odděleny, středníkem a je možné psát více příkazů na jeden řádek. Tento jazyk je vhodný pro definici komplexnějších funkčních bloků a následně je využit v jiných programovacích jazycích.

```
Vystup:= VstupA AND VstupB; //Funkce VstupA AND VstupB
```

Obrázek 1.6 Příklad programového kódu v jazyce ST

LD (Ladder Diagram) je nejjednodušší grafický jazyk. Jedná se o grafické znázornění reléové logiky. Na levé straně jsou vstupy a na pravé straně jsou výstupy. Tento programovací jazyk je určen pro jednodušší úlohy.



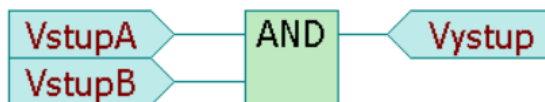
Obrázek 1.7 Příklad programového kódu v grafickém jazyku LD

FBD (Function Block Diagram) je druhý grafický programovací jazyk, který se vytváří pomocí funkčních bloků. Program se provádí po řádcích. Tento druh programování se nehodí pro spojitě a matematické operace.



Obrázek 1.8 Příklad programového kódu v grafickém jazyku FBD

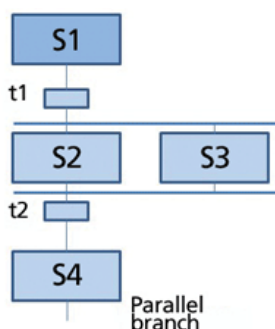
CFC (Continuous Flow Chart) je editor na jehož plochu se nejprve položí funkční bloky a po té se připojí vstupy a výstupy, případně se bloky mohou propojit mezi sebou. Lze si zde i vytvořit vlastní funkční blok. Na rozdíl od FBD lze zde tvořit i zpětnou vazbu. Tento způsob programování je při malém počtu bloků velmi čitelný a přehledný. Velkou výhodou je znázorňování aktivních cest [4].



Obrázek 1.9 Příklad programového kódu v grafickém jazyku CFC

SFC (Sequential Function Chart) je také grafický jazyk, který se skládá ze dvou základních částí, tj. z kroku (Step) a přechodu (Transition). V každém kroku je blok akcí, které jsou aktivní do té doby, než jsou splněny podmínky přechodu. Po splnění této podmínky program přejde k následujícímu kroku, který aktivuje a blok minulého kroku se přechodem deaktivuje.

[5]Chyba! Nenalezen zdroj odkazů.



Obrázek 1.10 Příklad sekvenčního kódu [6]

1.5 Komunikace s okolím

Pro komunikaci PLC automatu s okolím existuje hned několik možností. Následující kapitola je věnována několika způsobům komunikace a komunikačním protokolům.

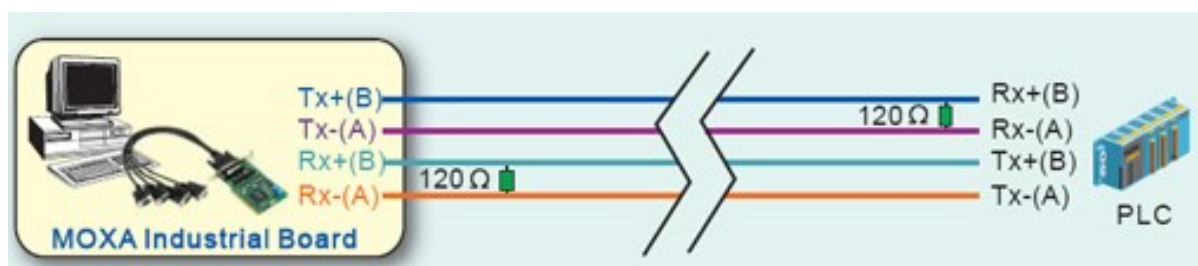
1.5.1 Sériová komunikace RS-232C

Je realizována pomocí tří vodičů. Jedním vodičem se propojí výstup první stanice se vstupem druhé stanice. Obdobně se postupuje i u druhého vodiče. Třetí vodič je společná zem. Na takto propojených stanicích je možné provozovat plně duplexní komunikaci (fullduplex). To znamená, že jedna stanice může přijímat i vysílat data zároveň. Tato konfigurace je v praxi nejtýpější. Komunikace probíhá změnou napětí na jednotlivých vodičích vůči zemi. Z těchto důvodů je tato komunikace náchylná na rušení. Proto je maximální délka přenosu do 15m. Při této délce se obvykle používá přenosová rychlost 9600B/s. Pro krátké vzdálenosti lze dosáhnout rychlosti 115200B/s

1.5.2 Sériová komunikace RS-422 a RS-485

Na rozdíl od RS-232 je zde k přenosu použita kroucená dvojlinka (Twisted pair-TP). Vysílání bitů se děje pomocí změn potenciálů mezi těmito vodiči. Díky tomu je tato komunikace odolnější vůči rušení. Proto ji lze provozovat na větší vzdálenost a to až do 1,2 km při rychlosti 2.5MB/s. Celá linka musí být ukončena rezistory s odporem 120Ω .

RS-422 je určena pro point-to-point komunikaci a je dost často používaná v průmyslovém prostředí. V rámci jedné sítě lze provozovat jedno zařízení v režimu master a až deset ve slave [7].



Obrázek 1.11 Topologie zapojení pro RS422 fullduplex [8].

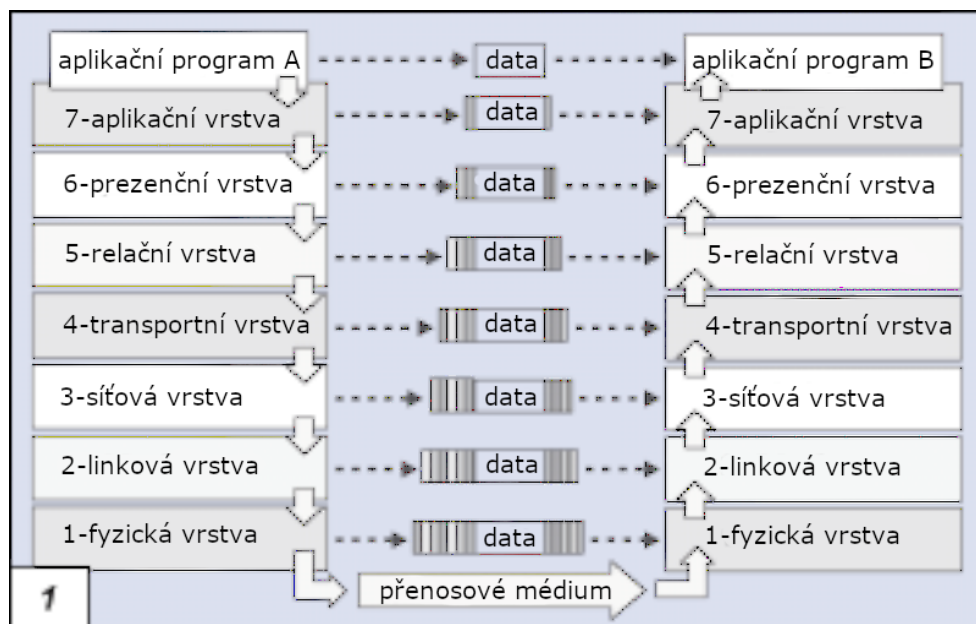
RS-485 využívá vícebodovou komunikaci. To umožňuje provozovat více přijímačů a vysílačů na jedné lince. U většiny se používá Master/Slave architektura. Kde každá slave stanice má svou unikátní adresu. Jednotka odpovídá pouze na pakety jí určené.



Obrázek 1.12 Propojení dvou zařízení na sběrnici RS-485[8].

1.5.3 ETHERNET

Pro přenos na větší vzdálenosti lze využít síť Ethernet. Tento typ komunikace je v dnešní době velmi rozšířený. Je definovaný standardem IEEE802.3 a definuje první dvě vrstvy modelu ISO/OSI. Řízení přístupu se děje pomocí CSMA/CD (Carrier Sense Multiple Access / Collision Detection). Zařízení připravené vysílat, naslouchá, zda je médium volné. Jinak čeká na jeho uvolnění (CSMA). Po zahájení vysílání stanice pořád naslouchá, zda nezačalo vysílat i jiné zařízení. Pokud ano nastává kolize (CD). Stanice přestane vysílat a vyšle JAM signál. Dále nastaví náhodnou hodnotu, za kterou se vysílání pokusí zopakovat. Nevýhodou tohoto systému je, že nelze zaručit přístup k přenosovému médium v definovaném okamžiku. Proto ji nelze použít pro měření, ovládání a řízení v reálném čase. Se zvětšující se rychlostí přenosu (1000 Mbit/s) se však tato situace zlepšuje [9].

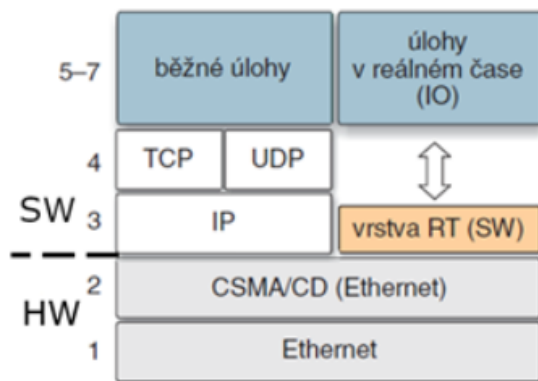


Obrázek 1.13 Referenční model ISO/OSI [9].

Mezinárodní standardizační organizace (International Electrotechnical Commission – IEC) připravila modifikaci standardu sériové sběrnice IEC 61158 World fieldbus a to v oblasti průmyslového ethernetu. Díky velkému počtu variant průmyslového ethernetu a jeho rozšíření musela komise standardizovat téměř všechny varianty.

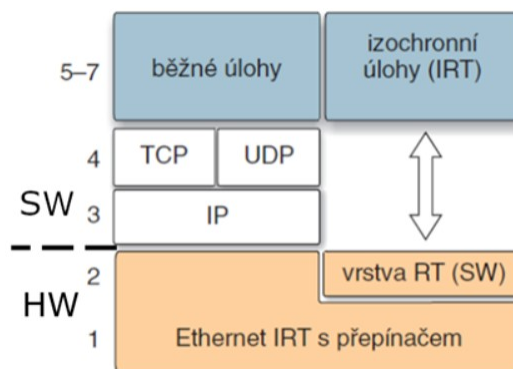
Profinet

Profinet byl vyvinut organizací PNO (Profibus Nutzerorganisation) ve spolupráci s firmou Siemens. Komunikační model Profinet Verse 2 označován též jako Profinet IO, pracuje tak, že zprávy nepožadující přenos v reálném čase (non real-time) jsou přenášena TCP/UDP/IP. Druhý kanál obsahuje programové překlenutí (SW by-pass) vrstev 3 a 4 komunikačního zásobníku. Tím lze dosáhnout lepších vlastností reálného času. Dalším vylepšením je redukce délky přenášených bloků a jsou zde zavedeny také priority.



Obrázek 1.14 Komunikační model Profinet IO [10]

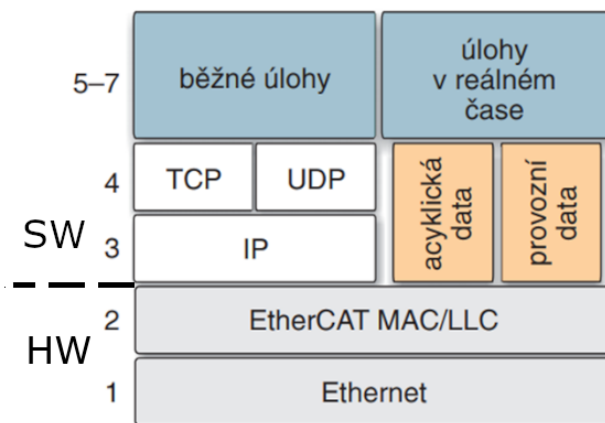
Profinet IRT (Isochronous Real Time) je využíván tam kde jsou kladeny požadavky na dodržení odezvy a synchronizaci. Pro přenos je využit speciálního hardwaru, který překlene vrstvu TCP/IP Ethernetu (HW by-pass). Pro přenos zpráv nepožadující obsluhu v reálném čase a pro přístup na internet se realizuje paralelní cestou TCP/UDP/IP.



Obrázek 1.15 Komunikační model Profinet IRT [10]

EtherCAT

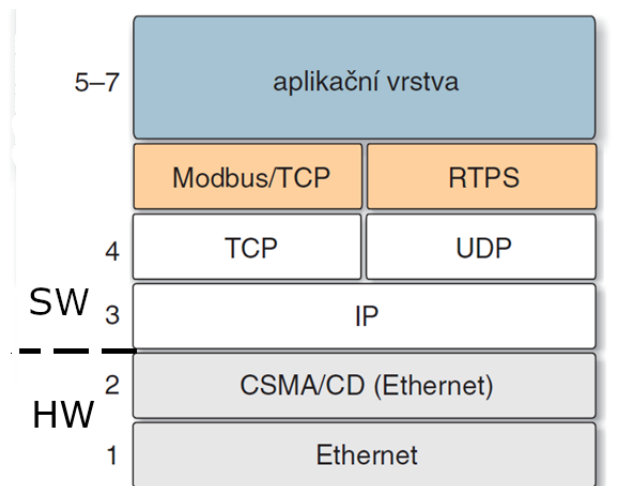
Standard EtherCAT (Ethernet for Control Automation Technology) byl vyvinut firmou Beckhoff pro možnosti řízení v reálném čase. Na rozdíl od standardního ukládání rámce Ethernetu a následném zpracování, je zde rámec zpracován během jeho příjmu. Každé zařízení obsahuje 2 porty a tak může rámec procházet skrz zařízení. A tak může zařízení, pro které byla zpráva určena, číst data, zatímco rámec může být poslán dalšímu zařízení. Obdobně pracuje i ukládání dat do rámce. To zajistí zdržení rámce v každém segmentu jen několik nanosekund. Z toho plyne i vyšší přenosová rychlost dat oproti jiným systémům na bázi Ethernetu. Poslední segment v síti posílá rámec zpět prvnímu. Pro zajištění přenosu rámce, který má danou dobu odezvy (tzv. real-time frames) je využita vyšší priorita. Rámce s nižší prioritou (např. konfigurační, diagnostická) jsou přenášena, pokud je k dispozici dostatečně dlouhá časová přestávka.



Obrázek 1.16 komunikační model EtherCAT [10]

Modbus-RTPS

Modbus-RTPS (Real Time Publish-Subscribe) je americký protokol vycházející z protokolu Modbus/TCP. Jeho cílem je zlepšit vlastnosti reálného času u protokolu Modbus/TCP. Výhodou tohoto protokolu je to, že vychází z nejrozšířenějšího komunikačního protokolu v oboru automatizace a to protokolu Modbus. Modbus byl vyvinut firmou Modicon (později AEG) v roce 1989. Jeho specifikace je volně dostupná a od jeho vzniku nedošlo k žádné změně. Díky tomu mají systémy Modbus/TCP a Modbus-RTPS dobré předpoklady pro úspěch na trhu.



Obrázek 1.17 Komunikační model Modbus [10]

Protokol Modbus zabalí datagram a protokoly TCP/IP do jednoho rámce. Po té tento rámeček odešle standardní přístupovou metodou, která je přenášena fyzickou vrstvou. Standardní rámeček Modbus obsahuje adresu řízené stanice (slave), kód operace definovanou v protokolu Modbus, přenášená data a kontrolní součet. Kód operace říká řízené stanici co má dělat s daty. Modbus/TCP neobsahuje kontrolní součet, protože kontrola se provádí ve vrstvách 1-4. V protokolu Ethernet/TCP, dle normy IEC/PAS, které definuje protokol s vlastnostmi reálného času, existuje rozšíření umožňující komunikaci způsobem publish-subscribe přes standardní cestu UDP/IP. Toto rozšíření je základem protokolu Modbus-RTPS.

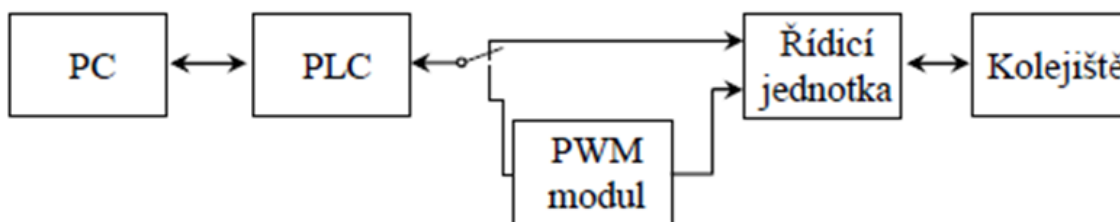
[10]

2 MODEL KOLEJIŠTĚ

Původní verzi modelu vlakového kolejiště zkonstruoval Vladimír Hnilica v rámci své bakalářské práce, již v roce 1999. Model kolejiště se využívá při výuce předmětu Programovatelné automaty. Pomocí PLC automatu je možno řídit pohyb vlaků po kolejišti. V rámci bakalářských prací byl původní model již několikrát upravován.

2.1 Současný stav

Současný model kolejiště je rozdělen na deset úseků, na kterých se pomocí PWM řídí pohyb lokomotiv. Model železnice se rozkládá na ploše 2m^2 , na které jsou umístěny dvě nádraží. Celá trať obsahuje 6 jednoduchých výhybek a jednu křížovou výhybku. Celý model je řízený pomocí externí převodníkové skříně [11].



Obrázek 2.1 Blokové schéma modelu vlakového kolejiště [11].

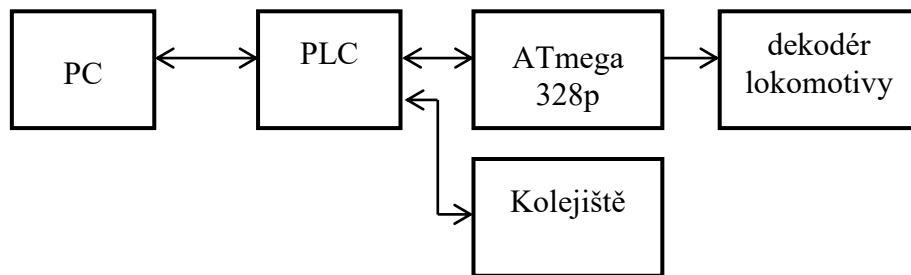
Po prohlédnutí stavu kolejiště proběhla testovací jízda, která odhalila problémovou část modelu kolejiště. V úseku před tunelem je zatáčka vytvořená pomocí flexi koleje. Protože tato zatáčka má malý poloměr, lokomotiva ve vyšší rychlosti vykolejí. Rozložení kolejí je dáno modelem a nelze jej moc měnit, jednotlivé úseky jsou poskládány z nestandardních částí kolejí.

Další problematickou částí bylo samotné napojení modelu kolejiště na řídicí jednotku, které je realizováno pomocí plochého kabelu. Zde hrozí nebezpečí poškození či vytržení kabelu.

Souhrnně, tedy z výše uvedených důvodů bylo rozhodnuto vytvořit model nový.

2.2 Nový model

Nově navržený model bude složen ze standartních kolejí, mající typické rozměry. Tento zvolený postup by měl zajistit větší stabilitu vlaku při průjezdu zatáčkou a dále také zjednodušení případné výměny kolejnic při opravě kolejiště. Lokomotivy budou vybaveny digitálním dekodérem, pro digitální provoz kolejiště. Díky digitalizaci provozu nebude třeba kolejiště dělit na jednotlivé úseky a taky bude možno ovládat více lokomotiv na jedné koleji. Pro digitální řízení modelu bude použit standart DCC (Digital Command Control). Model bude vybaven snímači pro snímání pohybu vlaku po kolejišti. Dále bude celá řídicí část již součástí modelu. Vstupní rozhraní modelu bude obsahovat ethernetovou přípojku pro programování PLC automatů, USB konektor pro programování ATmega 328 a vstupní konektor pro připojení napájecího adaptéru. Celý model bude usazen na podstavě s kolečky pro snazší manipulaci.



Obrázek 2.2 Blokové schéma zapojení modelu kolejiště

3 DIGITÁLNÍ ŘÍZENÍ MODELOVÉ ŽELEZNICE

Rozeznáváme dva druhy digitálního řízení modelového kolejiště. První způsob je řízení pomocí PWM modulace. Při tomto typu řízení je nutné mít kolejiště rozděleno na jednotlivé úseky a pomocí PWM modulace se řídí energie přiváděná do lokomotivy. Při tomto typu řízení není možné řídit dvě lokomotivy na jednom úseku.

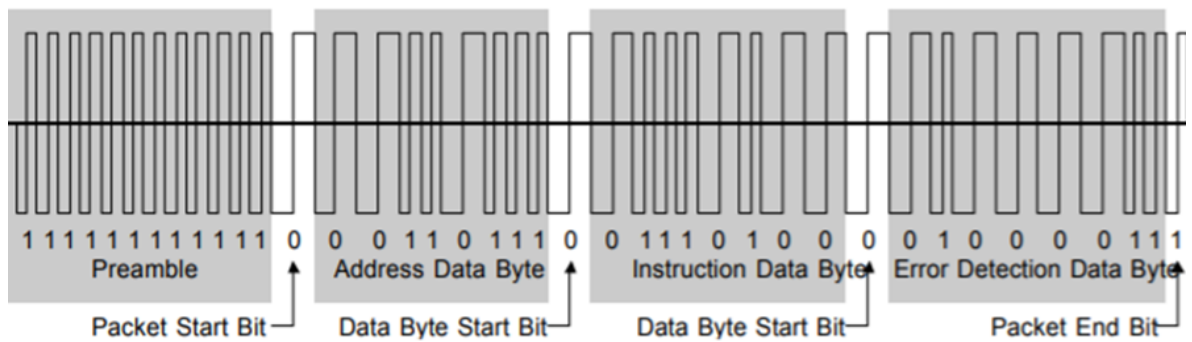
Při použití standardu DCC je kolejiště spojeno v jeden celek. Pomocí změny polarit napájení kolejí se vysílají zprávy k dekodérům. Díky těmto dekodérům lze řídit více lokomotiv na jednom úseku kolejiště.

3.1 DCC

DCC standard byl definován společností NMRA. Díky standardizaci lze na jednom kolejišti provozovat komponenty od různých výrobců [12]. Jedná se o digitální řízení modelového kolejiště. Při tomto způsobu řízení nemusí být kolejiště složeno z elektricky oddělených úseků. Řízení probíhá pomocí jednosměrného zasílání paketů. Tyto pakety jsou vysílány přes koleje k jednotlivým dekodérům přes napájecí koleje. Výhodou tohoto způsobu je to, že lze řídit více lokomotiv na jednom úseku kolejiště. Nevýhodou toho řízení je nutnost ke každému komponentu mít vlastní dekodér.

Pakety se dají rozdělit do tří skupin. První skupinou je paket IDLE, který neobsahuje žádné data. Vysílá se v době, kdy nejsou jiné zprávy k odeslání a slouží tak pouze pro napájení dekodérů na kolejišti. Druhou skupinu tvoří pakety pro řízení lokomotiv. Po vyslání této zprávy je přečtena všemi dekodéry, které jsou připojeny na daném úseku. Zpráva však bude vykonána pouze dekodérem pro, který byla určena. Na stejném principu pracuje i třetí skupina paketů pomoci, které můžeme ovládat periferie (výhybky, návěstidla, osvětlení ...) [13]

Základní DCC paket se skládá ze 4 částí. První je hlavička paketu tzv. Preambule. Skládá se z minimálně 12 a maximálně 14 po sobě jdoucích log.1. Mezi každou část se vkládá oddělovací bit tj. log.0. Další byte je 8 bitová adresa dekodéru, kterému je zpráva určena. Pokud je adresa hodnota 0 tak se jedná o broadcast zprávu. Další byte obsahuje data pro dekodér. A poslední byte je kontrolní součet vytvořený pomocí funkce xor mezi adresovým a datovým bytem. Ukončovací bit je paketu je pak log.1 [13]



Obrázek 3.1: DCC paket [13]

Vysílání jednotlivých bitů se děje pomocí obdélníkového signálu. Pro log.1 je definován čas jedné půlperiody v rozmezí 52 až 64 us při poměru 1:1. Logická 0 je definována obdobně s tím rozdílem, že poměr se může měnit. Ovšem pokud je nejkratší trvání jedné půlperiody delší než 90us a celkový čas periody nesmí přesáhnout 12 000 us [13].

4 ŘÍDICÍ ELEKTRONIKA

4.1 Řídicí elektronika

PLC automaty jsou příliš pomalé pro generování příkazů pro DCC standart. Proto je třeba vložit mezi PLC automat a model železnice mikrokontrolér, který dokáže zajistit generování signálu ve správný čas. Pro tuto úlohu byl zvolen mikrokontrolér ATmega 328. Mikrokontrolér bude komunikovat s nadřazeným PLC po sériové lince RS232. Z důvodu malého výkonu mikrokontroléru byl přidán H můstek, který bude spínat silovou část napájení.

4.2 Mikrokontrolér

Mikrokontrolér je integrovaný obvod obsahující celý mikropočítač. V mikrokontroléru se nachází CPU (Central Process Unit), paměti ROM (Read Only Memory) a RAM (Random Access Memory) a jiné podpůrné obvody. Mezi ně patří obvody pro digitální vstupy a výstupy a obvod pro analogové vstupy a výstupy. Dále se zde mohou nacházet obvody čítačů a časovačů a obvody pro komunikaci.

Díky malým rozměrům a velké spolehlivosti se mikrokontroléry používají například v embeddit systémech, automobilovém průmyslu, hračkách a jiné.

4.2.1 RISC

RISC (Reduced Instruction Set Computer) označuje procesor s redukovanou instrukční sadou. Tyto procesory mají malý počet instrukcí (typicky 32 instrukcí). Velká většina instrukcí se provádí během jednoho strojového cyklu. Díky redukci instrukcí lze zmenšit plochu mikrokontrolér. Pro složitější operace je však nutno použít větší počet těchto instrukcí. Proto RISC procesory mohou mít větší počet registrů. Mezi neznámější zástupce této řady patří např. ARM, Atmel AVR nebo PIC[14].

4.2.2 CISC

CISC (Complex Instruction Set Computer) je kompletní sada instrukcí pro mikrokontrolér. Tyto instrukce mají proměnlivou délku a dobu vykonávání. Použitím komplexnějších instrukcí klesá nutnost mít mnoho registrů. Například použitím jedné instrukce pro násobení v CISC architektuře se musí v RISC použít sčítání a následný bitový posun dvou registrů [14].

5 SCADA/HMI

SCADA (Supervisory Control And Data Acquisition) je vizualizační systém, který zajišťuje dispečerský dohled, monitoring a sběr dat. Obvykle se jedná o software, který monitoruje z centrálního místa různé technické procesy jako například aktuální spotřebu elektřiny nebo vody či plynu. Díky možnosti zobrazení v internetovém prohlížeči lze daný systém kontrolovat odkudkoliv. Pomocí různých alarmů může být dispečer upozorněn na problém v řídicím procesu. Pokud se dispečerovy nepodaří tento problém vyřešit, může proces zastavit nebo vyslat na místo obsluhu. V důsledku tak může jeden dispečer řídit velmi složité systémy. Tento systém umožňuje ukládání data z několika set až stovky tisíc vstupů a výstupů. Vše se ukládá na SQL databázové servery[15].

HMI (Human Machine Interface) je software s GUI (Grafical User Interface) prostředím, zobrazující informace o řízeném procesu. Je zde možnost zadávat operátorské povely, zobrazovat data z databází, zobrazování alarmů a jiné [15].

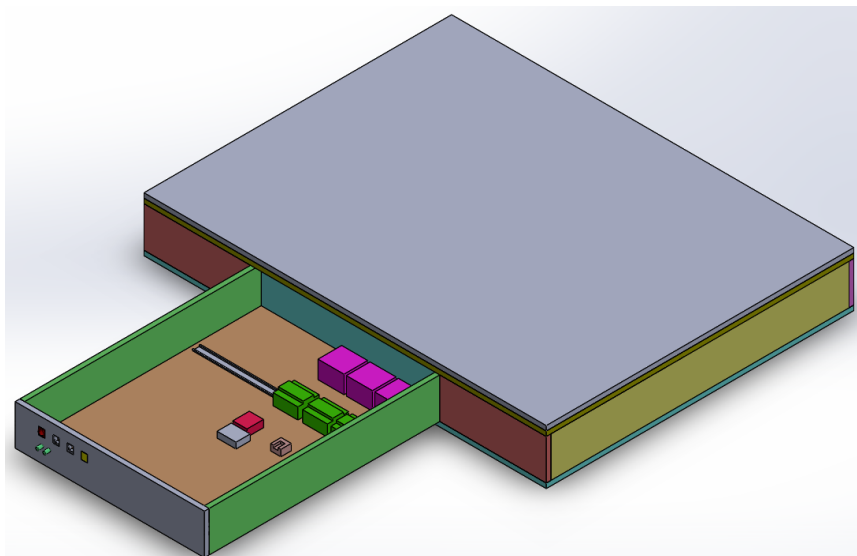
5.1 Control Web

Control Web je software pro vytváření a provoz aplikací pro průmysl. Dokáže nejen sbírat data a vizualizovat je ale taky dokáže modelovat a simulovat výrobní procesy, vytvářet náročné vizualizace nebo může sloužit jako datový server. Velká většina systémů je již napojena na počítačovou síť. Často jsou tyto systémy složeny z různých částí, které spolu musí komunikovat. Control Web se snaží zajistit komunikaci mezi těmito systémy a tak vše spojit v jeden celek[16].

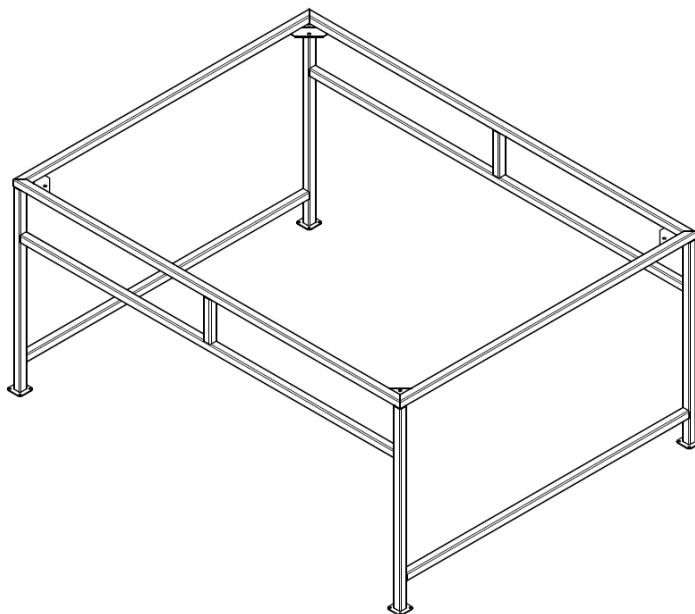
II. PRAKTICKÁ ČÁST

6 KONSTRUKCE

Jeden z požadavků na konstrukci byl takový, aby všechna řídicí elektronika byla umístěna pod modelem kolejistě. Zároveň měl být k této elektronice snadný přístup pro případnou úpravu zapojení. Model má mít možnost volby, kterým PLC automatem se má řídit. Proto zde bylo potřeba vyřešit přepínání PLC automatů. Měla by však zde být i rezerva místa pro případné rozšiřování modelu o další řídicí prvky. Celý model má být usazený na kovové konstrukci s kolečky, která usnadní případnou manipulaci s modelem.



Obrázek 6.1 Dřevěná konstrukce



Obrázek 6.2 Kovová část konstrukce

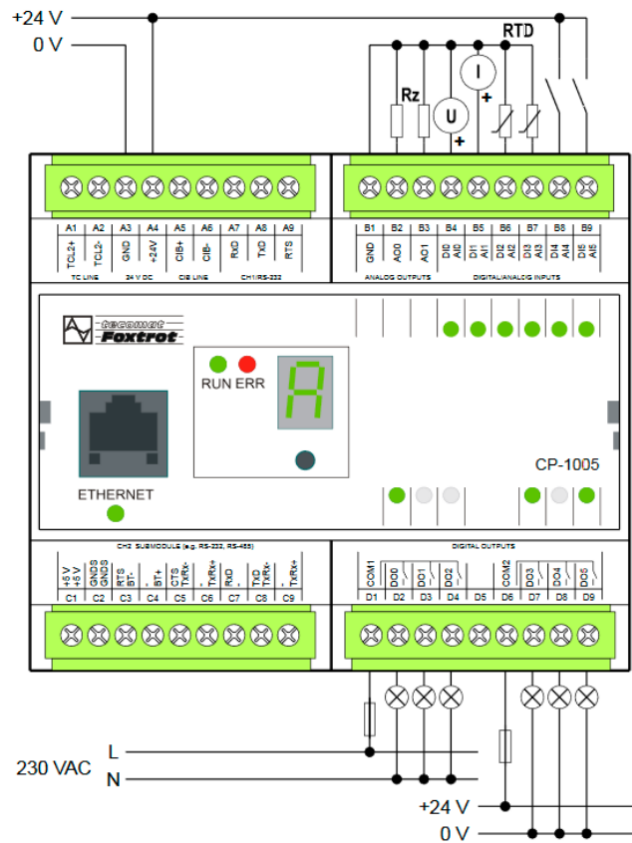
7 HARDWARE

Pro provoz PLC automatů je nutné připojení na 24V. Proto je použitý adaptér, který je připojen do PLC automatů. Protože lokomotivy jsou stavěny na napětí 16V bude napětí zdroje omezeno měničem napětí. Měnič má možnost nastavení proudového omezení. V tomto případě byl nastaven na 2A. Tato hodnota je maximální dlouhodobě spínaný proud H můstku, který se stará o změnu polarity napájení v kolejích. Takže při zkratu na kolejišti by nemělo dojít k poškození řídicí elektroniky.

7.1 PLC Tecomat

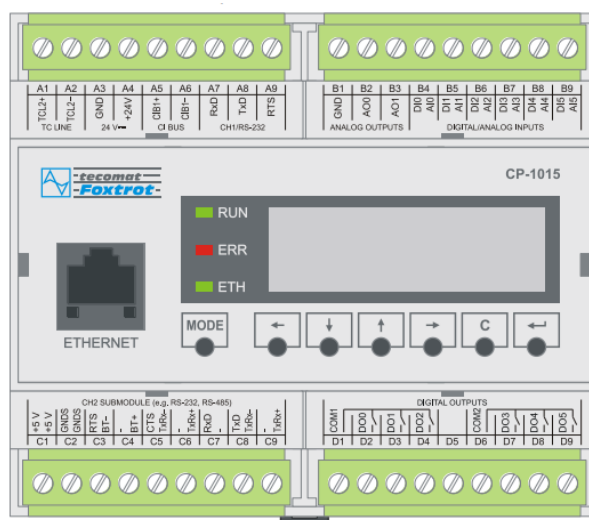
Pro řízení modelu železnice byl vybrán PLC automat od firmy Teco řady CP-1015. Programovatelné automaty od firmy Teco s typovým označením CP-10x5 jsou zástupci kategorie malých kompaktních PLC automatů. Jsou určeny pro nepřetržitý provoz v průmyslovém odvětví automatizace. První asynchronní kanál je pevně osazen sériovým rozhraním RS-232. Druhé rozhraní může být osazeno podle požadavků zákazníka (RS-232, RS-485, RS-422) Jsou podporovány různé průmyslové protokoly a sběrnice např.(MODBUS, PROFIBUS DP, CAN atd). Automaty jsou vybaveny 32 bit RISC procesorem a uživatelskou pamětí 192 kB. Oba moduly mají ethernetovou přípojku 100Mb. Dále automat obsahuje 6 vstupů s možností volby, zda mají být vstupy binární či analogové. Při použití analogových vstupů lze měřit napětí, odpor a proud. Rozlišení těchto vstupů je 12 bit s dobou převodu 80 μ s. Maximální měřicí rozsah napětí je 0-10V. Spínání je zajištěno 6 releovými výstupy s maximálním zatížením 230 V a 3A. Při zapojování je třeba dbát na správnost zapojení, protože výstupy nemají ochranu proti zkratu. PLC automaty jsou napájeny 24V DC a jsou vybaveny vratnou pojistkou [17].

Modul CP-1005 obsahuje 7 segmentový display a tlačítko MODE. Po stisknutí tohoto tlačítka začne na displeji rotovat IP adresa PLC automatu [17].



Obrázek 7.1 PLC Tecomat CP-1005 s příkladem zapojení [17]

Obdobně je tomu tak i u modulu CP-1015. Ten oproti předchozímu obsahuje LCD displej na kterém lze zobrazit 4 x 20 znaků. Dále obsahuje kromě tlačítka MODE i 6 uživatelských tlačítek tlačítek.



Obrázek 7.2 PLC Tecomat CP-1015 [18]

V této práci bylo, pro řízení vytvořeného laboratorního modelu, využito PLC Tecomat model CP-1015.

7.2 ATmega328

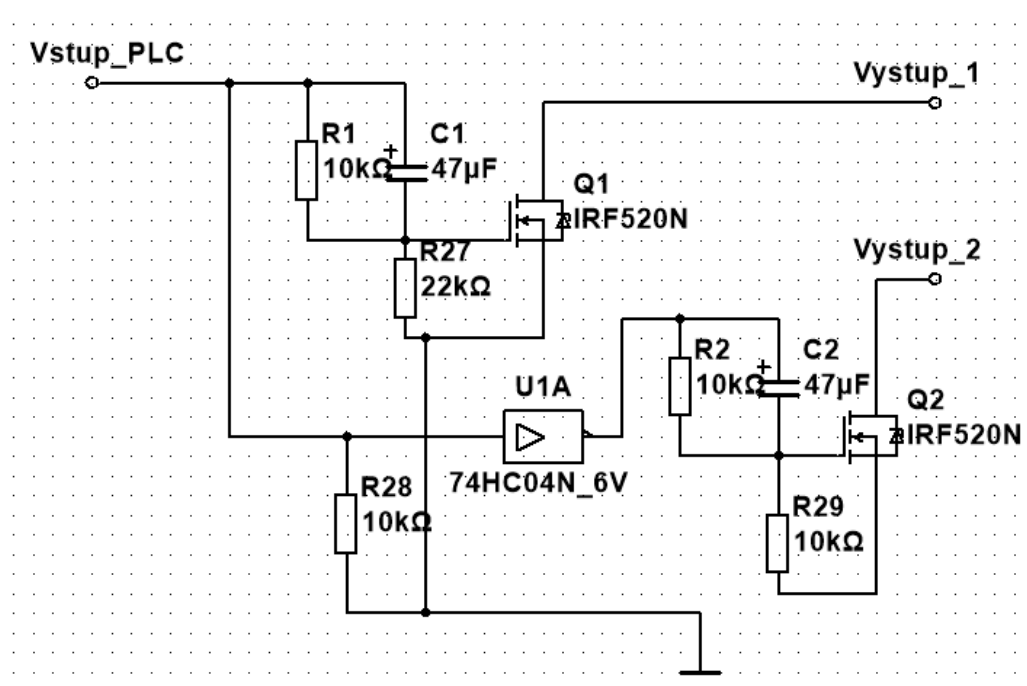
Pro zprostředkování zpráv z PLC do lokomotivy byl vybrán mikrokontrolér ATmega 328. ATmega 328 je 8 bitový RISC mikrokontrolér s 32 kB SPI flash pamětí od firmy Atmel. Tento mikroprocesor pracuje na frekvenci 16MHz, obsahuje 32 pracovních registrů a sériové rozhraní.

Díky redukované sadě instrukcí je mikrokontrolér velmi rychlý. Pomocí zapisování logické jedničky nebo nuly na portB mikrokontroléru se tato hodnota objeví na jeho výstupním pinu. Na tyto piny je připojen H můstek, který spíná silovou část a přivádí tak napětí o správné polaritě do kolejiště. Pro zápis logické hodnoty na pin jsem použil instrukce trvající jeden strojový cyklus.

Mikrokontrolér obsahuje 3 časovače, dva 8 bitové a jeden 16 bitový. Timer 0 je 8 bitový a ovládá časování pro sériovou komunikaci. Timer 1 je 16 bitový a využívá se při příkazech pro ovládání serva a při zapisování analogové hodnoty na piny 9 a 10. Timer 2 je 8 bitový a využívá se při funkci pro generování tónů. Časovače pracují na frekvenci 16 MHz, která je generována krystalickým oscilátorem. Díky obsluze přerušeni generovaným časovačem mikrokontroléru lze tyto změny polarity provádět ve správném čase.

8 ŘÍDICÍ ELEKTRONIKA

Výhybky od firmy Piko jsou vyrobeny tak, že po přepnutí do požadované polohy se elektromagnet odpojí od obvodu. Pokud se výhybka nemůže přestavit, teče do elektromagnetu proud neustále. Proto byl vytvořen návrh obvodu, který po určitém čase výrazně omezí proud procházející elektromagnetem. To by mělo zajistit ochranu elektromagnetu při poruše výhybky. Výstup 1 obvodu je ovládán vstupem z PLC automatu. Výstup 2 je ovládán negací vstupu. Toto zapojení by mělo zabránit stavu kdy na jednu výhybku je kladen požadavek na přepnutí do polohy 1 a do polohy 2 v jednom okamžiku.



Obrázek 8.1 Schéma ovládání výhybky

Návrh desky plošných spojů je určen pro 6 výhybek. Rozměry návrhu jsou uzpůsobeny tak aby jej bylo možné zapouzdřit a připevnit na DIN lištu.

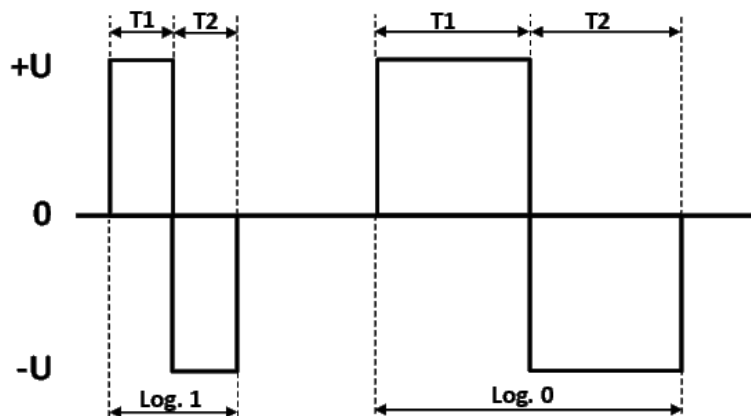
8.1 Snímání obsazení úseků

Pro snímání obsazení úseků byl použit stejný princip, jako to bylo u předchozího modelu a to pomocí jazýčkových relé. Pro správnou funkci musí být lokomotivy vybaveny permanentním magnetem. Pomocí magnetu se pak spíná jazýčkové relé, které vyvolá impulz. Tento impulz je dále zpracován PLC automatem. Na hranici dvou úseků musí být tyto senzory dva, aby bylo možné určit, z kterého směru lokomotiva přijela.

9 SOFTWARE

9.1 Program pro ATmega 328

Vysílání jednotlivých bitů paketu se děje pomocí změny polaroty napájení kolejí. To probíhá tak, že přivedeme kladnou část signálu na jednu kolejnici a druhá kolejnice má hodnotu 0V (GND). Po určitém čase se pak hodnota napětí na kolejnicích prohodí. Pro zápis log.1 se musí časy T_1 a T_2 rovnat. Přičemž doba T_1 je mezi 52 až 64 mikrosekund. Pokud se má vyslat hodnota 0 tak se časy T_1 a T_2 nemusí rovnat. T_1 a T_2 se mohou pohybovat v rozmezí od 90 až do 10 000 mikrosekund, přičemž celková doba nesmí přesáhnout 12 000 mikrosekund.



Obrázek 9.1 Polarita napájení při vysílání jednotlivých úrovní v DCC

Změna polaroty v kolejích je zde zajištěna pomocí mikrokontroléru ovládající H můstek. Přivedením log.1 na první nebo na druhý vstup H můstku můžeme kontrolovat polaritu napájení kolejí.

Čítač je registr, který dokáže zvýšit (snížit) svoji hodnotu o 1 při příchodu signálu. Čítač může čítat signály přicházející z externího nebo vnitřního zdroje. Vnitřním zdrojem pulzů může být oscilátor generující takt pro mikropočítač. ATmega 328 má krystalový oscilátor s frekvencí 16MHz. Čítáním těchto pulzů tak dostaneme časovač s velkou přesností. ATmega 328 disponuje třemi časovači Timer0-Timer2, kde Timer0 a Timer2 jsou 8 bitové a Timer1 je 16 bitový.

Pro generování přerušení ve správném čase byl vybrán k nastavení časovač Timer2. Pro ovládání funkce tohoto čítače slouží registr TCCR2B. Nastavením jeho bitů CS22-CS20 lze vybrat zdroj signálu a jeho děličky (prescaler). Při konfiguraci bitů CS22 a CS20 na

log.0 a bitu CS21 na log.1 dostaneme čítač počítající pulzy z interního zdroje. Na vstupu tohoto čítače je nastavena dělička 8. Tímto nastavením dostaneme čítač, který se inkrementuje o 1 s frekvencí 2MHz. Dále nastavením bitu TOIE2 registru TIMSK2 je povoleno přerušení od přetečení časovače. Posledním krokem je přednastavení hodnoty čítače/časovače tak aby přetekl po požadovaném čase. V tomto případě byl použit čítač o velikosti 8 bitů. Dále byla brána vstupní frekvence 16MHz s děličkou 8. Výsledná frekvence čítání pak tedy byla 2MHz. Hodnota čítače se tedy inkrementuje o 1 každých $0,5 \cdot 10^{-6}$ s. Tedy za dobu 57μs bude mít čítač hodnotu vyšší o 114. Maximální hodnota čítače je 255 a pro přetečení za dobu 57 μs je potřeba nastavit jeho počáteční hodnotu na :

$$cas_kratky = 255 - 114 = 141$$

```
//-----Set up Timer 2-----  
void SetupTimer2(){  
    noInterrupts();  
    TCCR2A = 0;  
    TCCR2B = 0<<CS22 | 1<<CS21 | 0<<CS20;  
    TIMSK2 = 1<<TOIE2;  
    TCNT2=cas_kratky;  
    interrupts();  
}  
  
//-----
```

Obrázek 9.2 Příklad nastavení časovače Timer2

Obsluha přerušeni se vyvolá při příchodu signálu přerušeni. V tomto případě je to obsluha přerušeni při přetečení časovače Timer2. V obsluze přerušeni se rozhoduje jaká část periody DCC signálu se má vykonat, zda T1 nebo T2 (`isr_znak`). Při vysílání T1 se musí výstup 1 nastavit na log.1 a výstup 2 na log.0. Logická 1 na výstupu 1 se docílí pomocí log. součtu registru PORTB a log.1, která je posunuta na pozici PB1. Následně je obdobným postupem nastavena log.0 na pozici PB2.

```
//-----obsluha přerušeni-----
ISR(TIMER2_OVF_vect){
    char latency;
    if(isr_znak){
        PORTB |= (1<<PORTB1);
        PORTB &= ~(1<<PORTB2);
        isr_znak=0;
        latency=TCNT2;
        TCNT2=latency+cas_pred;
    } else {
        PORTB &= ~(1<<PORTB1);
        PORTB |= (1<<PORTB2);
        isr_znak=1;
    }
}
```

Obrázek 9.3 Obsluha přerušeni - příklad zápisu log. hodnot na výstupní porty

Po té se v obsluze řeší, jaké bity DCC kódu se mají vysílat. Jako první se vysílá tzv. preambule. To je 12 jedniček jdoucích po sobě. Tato preambule je pro každou zprávu stejná.

```
switch(stav){
    case preamble:
        flag=1;
        preamble_pocet++;
        if(preamble_pocet==12){
            stav=separator;
            preamble_pocet=0;
        }
        break;
}
```

Obrázek 9.4 Příklad zápisu log. hodnot při vysílání hlavičky

Po preambuli se vysílá oddělovací bit (tj. log.0). Proto se nastaví flag na log.0. Při tom se nastaví maska pro posílání dat (vystupni_index) na hodnotu 128 dec. (tj. 1000 0000 bin.).

```
case separator:  
    flag=0;  
    stav=data_posli;  
    vystup_index=0x80;  
    vystup=zpr[zpr_index].data[byte_index];  
break;
```

Obrázek 9.5 Příklad nastavení proměnných pro zápis separátoru

Pro posílání dat se využívá logického součinu masky (vystupni_index) a bytu obsahující zprávu pro vysílání (vystup). Po vyhodnocení 7. bitu jednoho bytu zprávy, dochází k rotaci log.1 masky do prava. Tímto způsobem se postupně čte byty zprávy. Po jeho přečtení se opět pošle separátor. Takto se pokračuje dokud nejsou všechny byty postupně přečteny. Po jejich přečtení se vysílá ukončovací bit.

```
case data_posli:  
    if (vystup & vystup_index){  
        flag=1;  
    }else{  
        flag=0;  
    }  
    vystup_index=vystup_index>>1;  
    if(vystup_index==0){  
        byte_index++;  
        if(byte_index>=data_pocet){  
            stav=konec;  
            byte_index=0;  
        } else {  
            stav=separator;  
        }  
    }  
break;
```

Obrázek 9.6 Příklad nastavení proměnných pro zápis bitu zprávy

Nakonec se vysílá ukončovací bit (log.1). Po odvísílání se začne číst další řádek zprávy. Po odvysílání všech řádků se vysílá pouze první řádek obsahující zprávu IDLE. Ta má význam nečinného paketu. Vysílá se pro udržení napájení po dobu neaktivity vysílání dalších zpráv.

```

case konec:
    flag=1;
    stav=preamble;
    zpr_index++;
    if(zpr_index>=Max_zprav){
        zpr_index=0;
    }
    if(Max_zprav<=1)
    {
        Max_zprav=1;
    }
    break;

```

Obrázek 9.7 Příklad nastavení proměnných pro ukončení paketu vysílání

Struktura zpráv pro vysílání se skládá ze dvou řádků. První řádek obsahuje zprávu IDLE a do druhého řádku se zapisují požadavky na lokomotivu. Na prvním místě každé zprávy je adresa. Pro zprávu IDLE je adresa 255 dec. Tato adresa má význam jako paket pro nikoho. Na dalším místě zprávy je datový byt, kde se ukládají data o požadovaném směru a rychlosti jízdy lokomotivy. Poslední byt obsahuje kontrolní součet . Ten se vytváří log. xorem mezi adresovým a datovým bytem.

```

struct Zprava zpr[2]={
    {0xFF, 0, 0}
    {loco_adr, 0x40, 0},
};

```

Obrázek 9.8 Příklad nastavení struktury zprávy

Napsat o dalších částech jako příjem zprávy přes seriál a kontrola zprávnosti; plnění struktury daty.

Další důležitou částí programu je příjem zpráv přes sériové rozhraní. Při příchodu zprávy program počká do odvysílání všech zpráv. Poté přečte přijaté data na sériové lince. V příchozí zprávě je na prvním místě (byt 0) uložena adresa lokomotivy. Dále pak zpráva obsahuje data (byt 1) o požadované rychlosti a směru jízdy, případně funkce, lokomotivy. Předposlední položkou (byt 2) je kontrolní součet vytvořený logickou funkcí XOR mezi adresovým a datovým bytem. Nakonec se posílá číslo odeslané zprávy.

```
void SoftSerEvent(){  
    if(SoftSer.available()){  
        while (Max_zprav>1 && kolo>1)  
        {  
        }  
        int x;  
        x=SoftSer.available();  
        i=0;  
        int ZprCis=0;
```

Obrázek 9.9 Příklad detekce délky zprávy

Následně se zakáže přerušení a začne se pracovat s přijatou zprávou. Jako první se uloží číslo příchozí zprávy. Poté se kontroluje správnost přijaté zprávy. Kontrola se provádí srovnáním příchozího kontrolního součtu a kontrolního součtu vytvořeného z adresy a dat příchozí zprávy. Pokud se tyto kontrolní součty rovnají, je obsah těchto bytů zapsán do struktury zpráv pro vysílání. Program zajistí i zpětnou vazbu odesláním přijatých bytů zpět do PLC automatu. Po provedení těchto kroků se opět povolí přerušení.

```
noInterrupts();
ZprCis=inByte[3];
if(inByte[2]==(inByte[0]^inByte[1]))
{
zpr[1].data[0]=inByte[0];
zpr[1].data[1]=inByte[1];
zpr[1].data[2]=inByte[2]
SoftSer.write(inByte[0]);
delay(1);
SoftSer.write(inByte[1]);
delay(1);
SoftSer.write(inByte[2]);
delay(1);
SoftSer.write(inByte[3]);
interrupts();
}
```

Obrázek 9.10 Příklad čtení a zpětné zaslání zprávy

Pokud však kontrolní součet nesouhlasí, přijatá zpráva se neuloží do struktury. Jako zpětná vazba pro PLC program odešle zprávu obsahující v prvních třech bitech hodnotu 255 a na posledním místě se posílá číslo zprávy.

```
else
{
noInterrupts();
Serial.println("chyba");
SoftSer.write(255);
delay(1);
SoftSer.write(255);
delay(1);
SoftSer.write(255);
delay(1);
SoftSer.write(inByte[3]);
interrupts();
}
```

Obrázek 9.11 Příklad kódu při detekci chyby zprávy

9.2 Program pro PLC

Řídící program pro PLC Teco je vytvořený pomocí programovacího jazyku ST (Structured Text). PLC čte vstupní hodnoty z jazýčkových kontaktů a podle pořadí jejich spínání pak provádí blokaci úseků. Při vjezdu další lokomotivy do již zablokovaného úseku PLC automat vyšle signál pro centrální zastavení všech lokomotiv. Také se vypíše zpráva s číslem úseku, který tento stav vyvolal.

```
IF((Sestupna_1 and IN_2) or (Sestupna_4 and IN_3))
THEN
Sekce_1:=TRUE;
END_IF;

IF( Sekce_1 and ((Nastupna_2 and not IN_1) or (Nastupna_3 and
not IN_4)))
THEN
Sekce_1:=FALSE;
END_IF;

IF( Sekce_1 and ((Nastupna_1 and not IN_2) or (Nastupna_4 and
not IN_3)))
THEN
CentrStop:=TRUE;
Err_1:=TRUE;
END_IF;
```

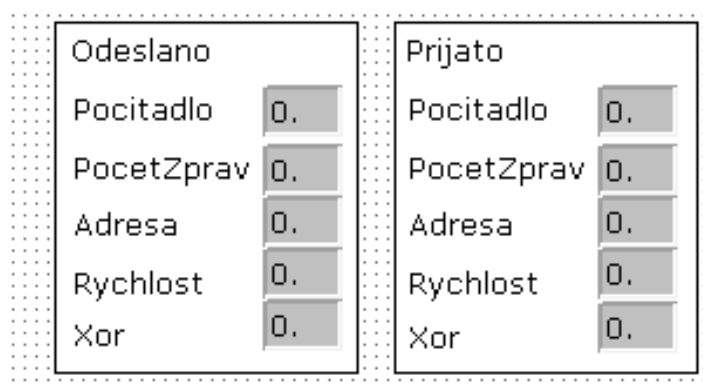
Obrázek 9.12 ukázka kódu řízení 1. úseku.

Dále z PLC automatu lze zasílat zprávy do mikrokontroléru a tím řídit pohyb jednotlivých lokomotiv. Komunikace probíhá zasíláním zpráv přes rozhraní RS232. Pro pohodlnou obsluhu je vytvořena vizualizace ve webovém serveru PLC Tecomat. Na první záložce je vytvořeno rozhraní pro sériovou komunikaci. Zde si obsluha zvolí počet bitů pro odeslání, adresu lokomotivy, její rychlost a směr. Potvrzení, odešle zprávu do mikrokontroléru. Lze zde i vyvolat manuálně centrální zastavení provozu na kolejišti.



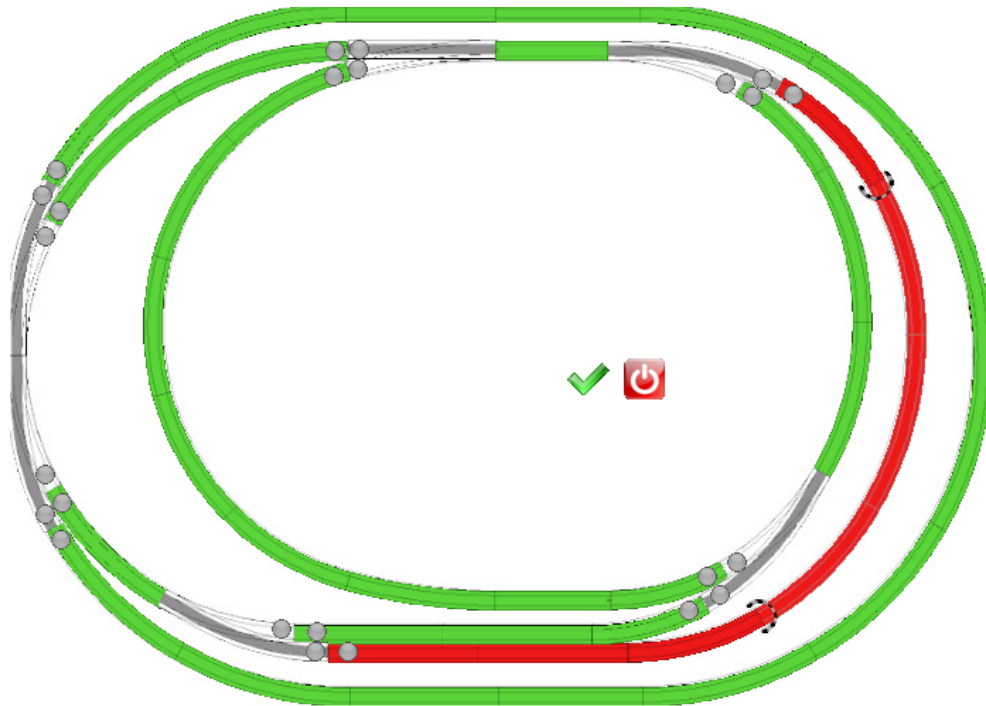
Obrázek 9.13 Vizualizace ovládání v prostředí Mosaic

Na záložce pro ovládání nalezneme i okýnko pro kontrolu komunikace. Kdy v levém sloupci jsou zobrazeny data odeslané přes RS232. V pravém sloupci jsou zobrazeny data přijatá z mikrokontroléru. Tak lze provést kontrolu, zda přenos zprávy proběhl správně.



Obrázek 9.14 Vizualizace komunikace mezi PLC a mikrokontrolérem

Na druhé záložce nalezneme vizualizaci kolejiště. Zde lze sledovat obsazenost úseků kolejiště a aktuální pozici výhybek. Přes webový server lze stavy výhybek měnit. V prostřední části je tlačítko pro nouzové zastavení provozu kolejiště.



Obrázek 9.15 Vizualizace úseků kolejí v programu Mosaic

ZÁVĚR

V úvodu bakalářské práce bylo již zmíněno, že původní cíl, který spočíval v inovaci již vytvořeného modelu vlakového kolejiště, musel být po jeho bližším prozkoumání přehodnocen.

Cílem této práce tedy bylo vytvoření modelu pro výuku programovatelných automatů na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně.

Dílčí práce, které vedly ke konečné podobě projektu, probíhaly následovně.

Nejprve byl vytvořen model konstrukce tak, aby bylo možné umístit řídicí elektroniku do modelu. Dále bylo nezbytné prostudovat způsob řízení digitálních lokomotiv, načež bylo přistoupeno k vytvoření programu pro mikrokontrolér ATmega 328P. Ten přijímá zprávy z PLC a následně je překládá do DCC kódu, jenž je srozumitelný pro dekodéry lokomotiv. Následující krok spočíval ve vytvoření programu pro PLC s vizualizací pomocí webového serveru. Jeho prostřednictvím je možno řídit pohyb lokomotiv a ovládat výhybky.

V rámci práce byl také realizován návrh a konstrukce řídicí elektroniky pro ovládání výhybek. Tato část je vložena mezi PLC a výhybky pro ochranu elektromagnetů přestavníků.

Vytvoření modelu bylo časově velmi náročné a při realizaci se vyskytla řada různých problémů. Jako příklad lze uvést obtíže při pokusu o vlastní výrobu desky plošných spojů, kterou bylo v konečném důsledku nezbytné nechat vyrobit odborníkem.

Veškeré nesnáze byly nakonec překonány a lze proto konstatovat, že všechny dílčí práce rezultovaly ve vytvoření funkčního modelu, čímž byl cíl splněn. Díky tomu si mohou studenti na tomto modelu procvičit programování PLC TECOMAT FOXTROT CP-1015.

SEZNAM POUŽITÉ LITERATURY

- [1] MARTINÁKOVÁ, Marie a Ladislav ŠMEJKAL. *PLC a automatizace. 1. Základní pojmy, úvod do programování*. Praha: BEN-technická literatura, 2009. ISBN 80-86056-58-9.
- [2] PLC Automatizace: *Sestava PLC* [online]. [cit. 2019-03-08] Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-hw-sestava.htm>
- [3] Měřicí a řídicí technika: *5 Logické řízení* [online]. [cit. 2019-04-12] Dostupné z: <http://uprt.vscht.cz/kminekm/mrt/F5/F5k53-PLC.htm>
- [4] TECO: *Informační bulletin pro uživatele systémů firmy Teco* [online]. [cit. 2019-04-14]. Dostupné z <https://www.tecomat.cz/modules/DownloadManager/download.php?alias=tecoinfo-39-cz>
- [5] Střední škola odborné přípravy Kroměříž: *Programovací jazyky pro PLC* [online]. [cit. 2019-04-07] Dostupné z: <https://coptkm.cz/portal/?action=2&doc=3905&docGroup=179&cmd=0&instance=1>
- [6] Motion Control Tips: *What are sequential function charts (SFCs) for PLCs?*: [online]. [cit. 2019-04-18] Dostupné z: <https://www.motioncontroltips.com/sequential-function-charts-sfcs-plcs/>
- [7] VYVOJ.HW.CZ: *RS 485 & 422* [online]. [cit. 2019-04-08]. Dostupné z: <https://vyvoj.hw.cz//teorie-a-praxe/dokumentace/rs-485-422.html>
- [8] ROOT.CZ: *Sběrnice RS-422, RS-423 a RS-485*. [online]. 18.12.2008 [cit. 2019-04-23]. Dostupné z: http://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-2005_04_30417_493/
- [9] ZEŽULKA, František a Ondřej HYNČICA: *Průmyslový Ethernet. AUTOMA* [online]. [cit. 2019-04-23]. Dostupné z: http://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-2005_04_30417_493/
- [10] ZEŽULKA, František a Ondřej HYNČICA: *Průmyslový Ethernet VII: Přehled současných standardů. AUTOMA* [online]. [cit. 2019-04-23]. Dostupné z: http://www.uamt.feec.vutbr.cz/~zezulka/download/KPPA/A02_08s26-VII.pdf

- [11] VINCENEC, Jiří: *Řízení modelu vlakového kolejiště pomocí programovatelného automatu*. Zlín, 2011 [cit.2018-11-27] Dostupné z: <http://hdl.handle.net/10563/16810>. Bakalářská práce. Univerzita Tomáše Bati. Vedoucí práce Pavel Navrátil
- [12] National Model Railroad association: *DCC Working Group* [online]. [cit. 2019-05-03] Dostupné z: <https://www.nmra.org/dcc-working-group>
- [13] National Model Railroad association: *DCC Working Group* [online]. [cit. 2019-05-03] Dostupné z: <https://www.nmra.org/sites/default/files/s-92-2004-07.pdf>
- [14] PROCESORY CISC A RISC: *Studijní materiál pro předmět Architektury počítačů* [online]. [cit. 2019-04-03]. Dostupné z: <http://poli.cs.vsb.cz/edu/arp/download/procrisc.pdf>
- [15] PROMOTIC: *Co je to SCADA* [online]. [cit. 2019-04-09]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsScada.htm>
- [16] MORAVSKÉ PŘÍSTROJE: *Co je Control Web?* [online]. [cit. 2019-04-03]. Dostupné z: <https://www.mii.cz/art?id=380&cat=146&lang=405>
- [17] TECO: *Příručky HW* [online]. [cit.:2019-05-01]. Dostupné z : <https://www.tecomat.cz/modules/DownloadManager/download.php?alias=foxtrot-cz-cp-1005>
- [18] TECO: *Příručky HW Tecomat CP-1005, cp-1015* [online]. [cit.:2019-05-01]. Dostupné z : https://www.tecomat.cz/download/get/txv00435_01_foxtrot_cp_1005/164/

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PLC	Programmable Logic Controller
DCC	Digital Command Control
CPU	Central Processing Unit
HW	Hardware
SW	Software
PWM	Pulse Width Modulation
I/O	Input/Output
PC	Program Counter
ROM	Read Only Memory
RAM	Random Access Memory
A/D	Analog-Digital
D/A	Digital-Analog
IL	Instruction List
ST	Structured Text
LD	Ladder Diagram
FBD	Function Block Diagram
CFC	Continuous Flow Chart
SFC	Sequential Function Chart
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
IEC	International Electrotechnical Commission
PNO	Profibus Nutzerorganisation
TCP/UDP/IP	Transmission Control Protocol/User Datagram Protocol/Internet Protocol
IRT	Isochronous Real Time
EtherCAT	Ethernet for Control Automation Technology

RTPS	Real Time Publish-Subscribe
NMRA	National Model Railroad Association
RISC	Reduced Instruction Set Computer
CISC	Complex Instruction Set Computer
ARM	Advanced RISC Machine
SCADA	Supervisory Control And Data Acquisition
HMI	Human Machine Interface
GUI	Grafical User Interface

SEZNAM OBRÁZKŮ

Obrázek 1.1 Schéma kompaktního PLC [2]	11
Obrázek 1.2 Schéma modulárního PLC [2].....	11
Obrázek 1.3 Blokovaná struktura PLC [3]	12
Obrázek 1.4 Otočka cyklu programu [1]	13
Obrázek 1.5 Příklad programového kódu v jazyce IL	14
Obrázek 1.6 Příklad programového kódu v jazyce ST	14
Obrázek 1.7 Příklad programového kódu v grafickém jazyku LD.....	14
Obrázek 1.8 Příklad programového kódu v grafickém jazyku FBD	15
Obrázek 1.9 Příklad programového kódu v grafickém jazyku CFC.....	15
Obrázek 1.10 Příklad sekvenčního kódu [6].....	15
Obrázek 1.11 Topologie zapojení pro RS422 fullduplex [8].....	16
Obrázek 1.12 Propojení dvou zařízení na sběrnici RS-485[8].	17
Obrázek 1.13 Referenční model ISO/OSI [9].....	17
Obrázek 1.14 Komunikační model Profinet IO [10]	18
Obrázek 1.15 Komunikační model Profinet IRT [10]	18
Obrázek 1.16 komunikační model EtherCAT [10].....	19
Obrázek 1.17 Komunikační model Modbus [10]	20
Obrázek 2.1 Blokované schéma modelu vlakového kolejiště [11].	21
Obrázek 2.2 Blokované schéma zapojení modelu kolejiště	22
Obrázek 3.1: DCC paket [13]	24
Obrázek 6.1 Dřevěná konstrukce	28
Obrázek 6.2 Kovová část konstrukce	28
Obrázek 7.1 PLC Tecomat CP-1005 s příkladem zapojení [17]	30
Obrázek 7.2 PLC Tecomat CP-1015 [18].....	30
Obrázek 8.1 Schéma ovládání výhybky	32
Obrázek 9.1 Polarita napájení při vysílání jednotlivých úrovní v DCC	33
Obrázek 9.2 Příklad nastavení časovače Timer2	34
Obrázek 9.3 Obsluha přerušení - příklad zápisu log. hodnot na výstupní porty.....	35
Obrázek 9.4 Příklad zápisu log. hodnot při vysílání hlavičky	35
Obrázek 9.5 Příklad nastavení proměnných pro zápis separátoru	36
Obrázek 9.6 Příklad nastavení proměnných pro zápis bitu zprávy.....	36
Obrázek 9.7 Příklad nastavení proměnných pro ukončení paketu vysílání.....	37

Obrázek 9.8 Příklad nastavení struktury zprávy	37
Obrázek 9.9 Příklad detekce délky zprávy	38
Obrázek 9.10 Příklad čtení a zpětné zaslání zprávy	39
Obrázek 9.11 Příklad kódu při detekci chyby zprávy.....	39
Obrázek 9.12 ukázka kódu řízení 1. úseku.	40
Obrázek 9.13 Vizualizace ovládání v prostředí Mosaic	41
Obrázek 9.14 Vizualizace komunikace mezi PLC a mikrokontrolérem	41
Obrázek 9.15 Vizualizace úseků kolejiště v programu Mosaic.....	42

SEZNAM PŘÍLOH

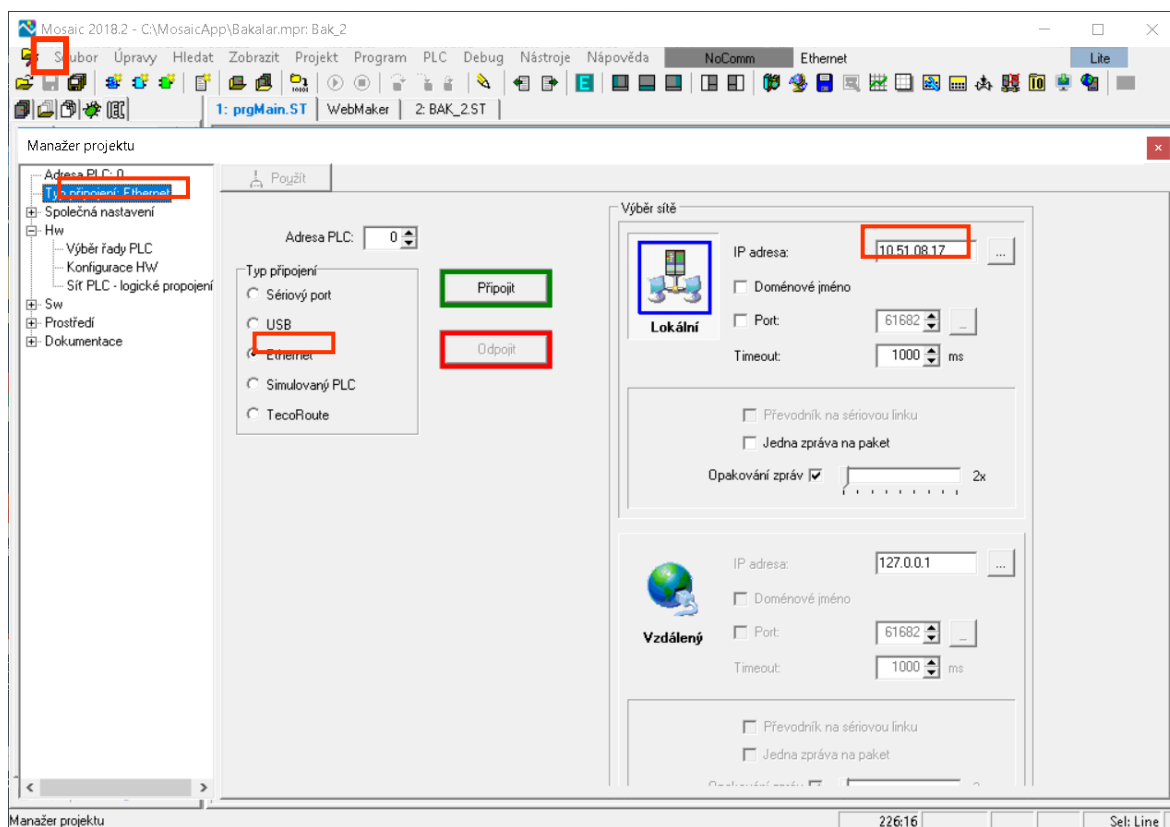
PŘÍLOHA 1 – NASTAVENÍ KOMUNIKACE V PROSTŘEDÍ MOSAIC

PŘÍLOHA 2 – ŘÍDICÍ ELEKTRONIKA VÝHYBEK

PŘÍLOHA P I: NASTAVENÍ KOMUNIKACE V PROSTŘEDÍ MOSAIC

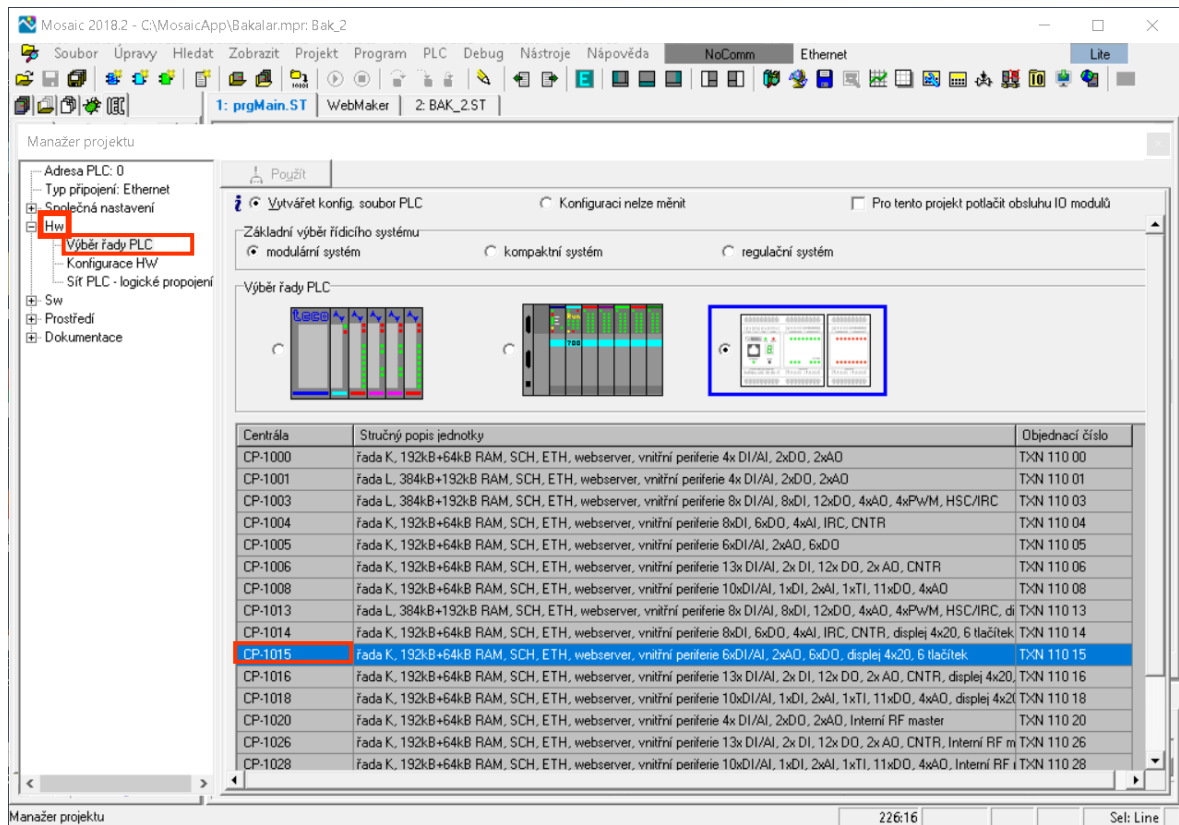
Po založení projektu v prostředí Mosaic je nutné nastavit typ připojení, typ PLC a připojené periferie. Postup je následující:

1. V levém horním rohu je ikona *Manažer Projektu*. Po otevření manažeru je možné se přepnout do záložky *Typ připojení*, kde se zvolí nastavení *Ethernet* a poté se nastaví odpovídající IP adresa.



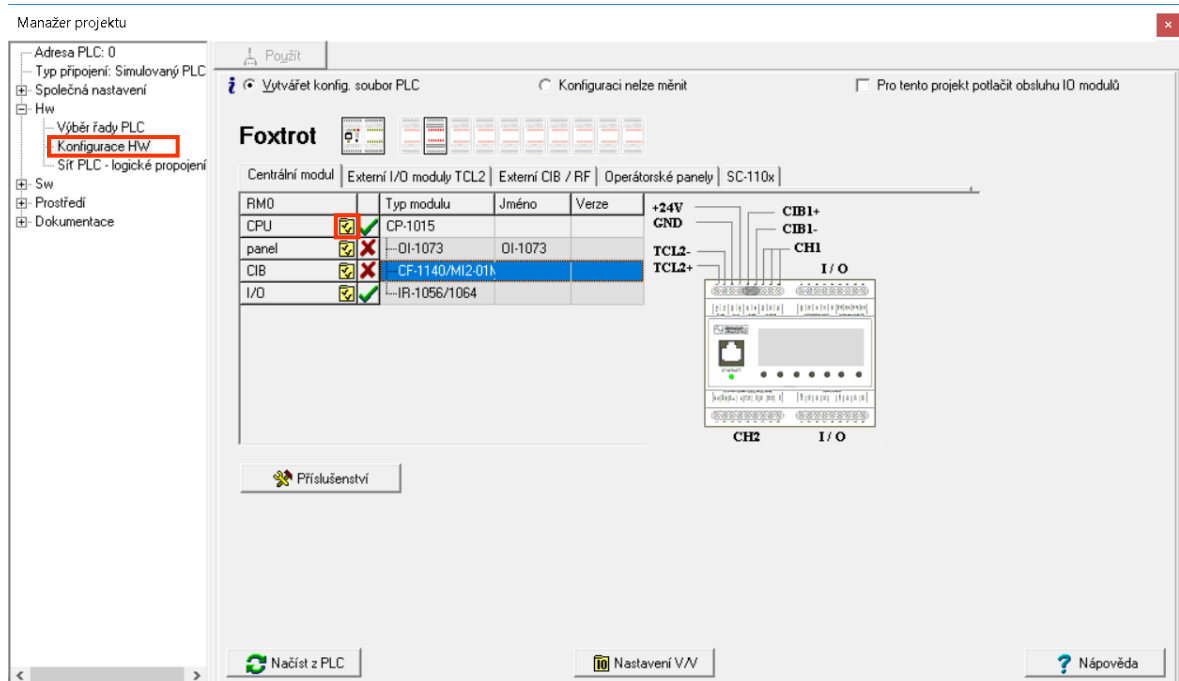
Obrázek I.1 Postup připojení na PLC v prostředí Mosaic

2. Dále se vybere v záložce *HW* položka *Výběr řady PLC* a poté se vybere odpovídající model. V tomto případě je to model CP-1015.



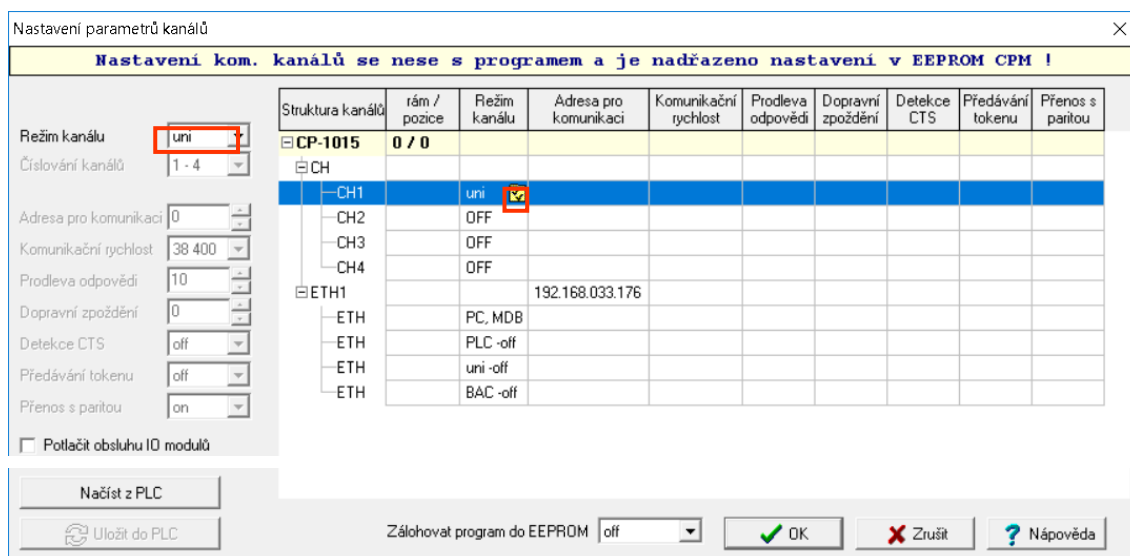
Obrázek I.2 Výběr modelu PLC automatu v prostředí Mosaic

3. V záložce HW se vybere položka *Konfigurace HW*, zde se nastaví položka CPU



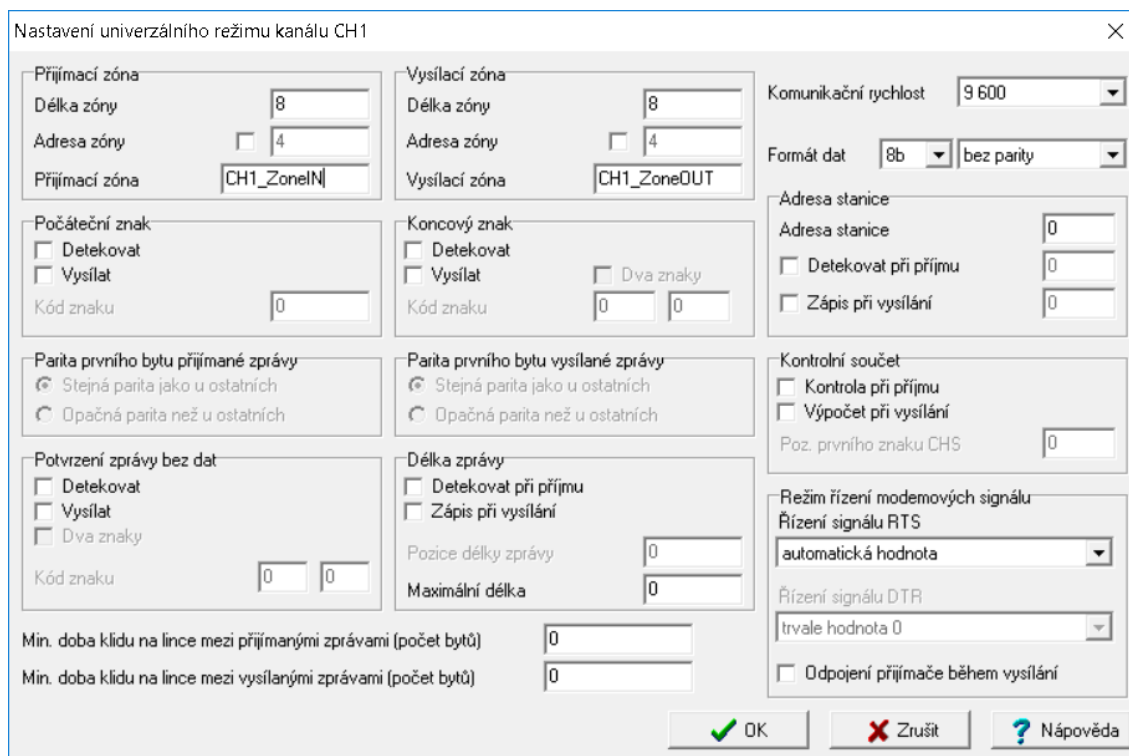
Obrázek I.3 Nastavení CPU PLC v prostředí Mosaic

4. Po otevření okna *Nastavení parametrů kanálů* se zvolí 1. kanál a nastaví se režim na *uni*. Dále se rozbalí možnosti kanálu 1.



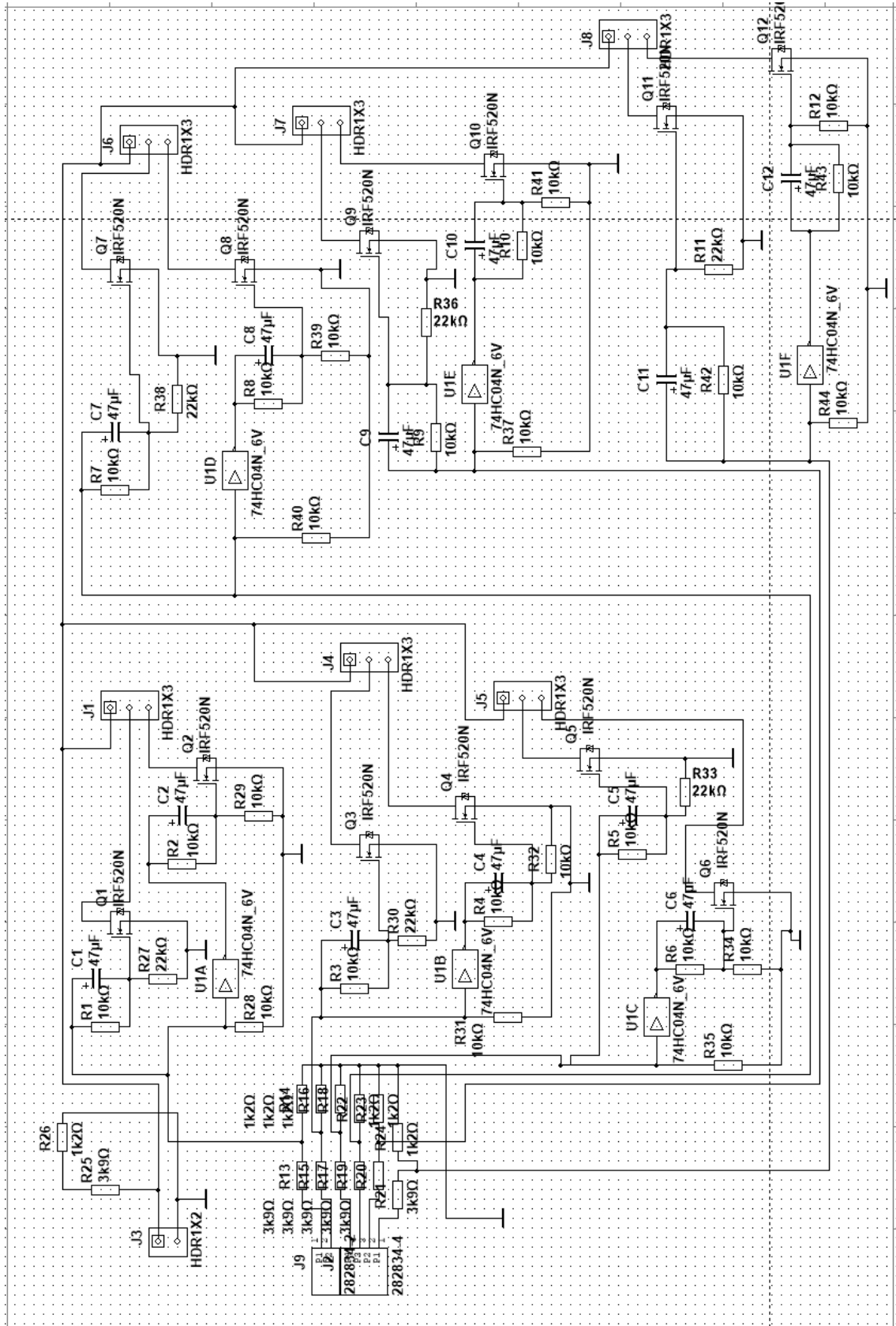
Obrázek I.4 Nastavení typu sériové komunikace v prostředí Mosaic

5. Dále se nastaví délka přijímací a vysílací zóny na 8 bitů. Dále je možné změnit názvy přijímací a vysílací zóny (CH1_ZoneIN, CH1_ZoneOUT). Komunikační rychlost se nastaví na 9600b a formát dat.

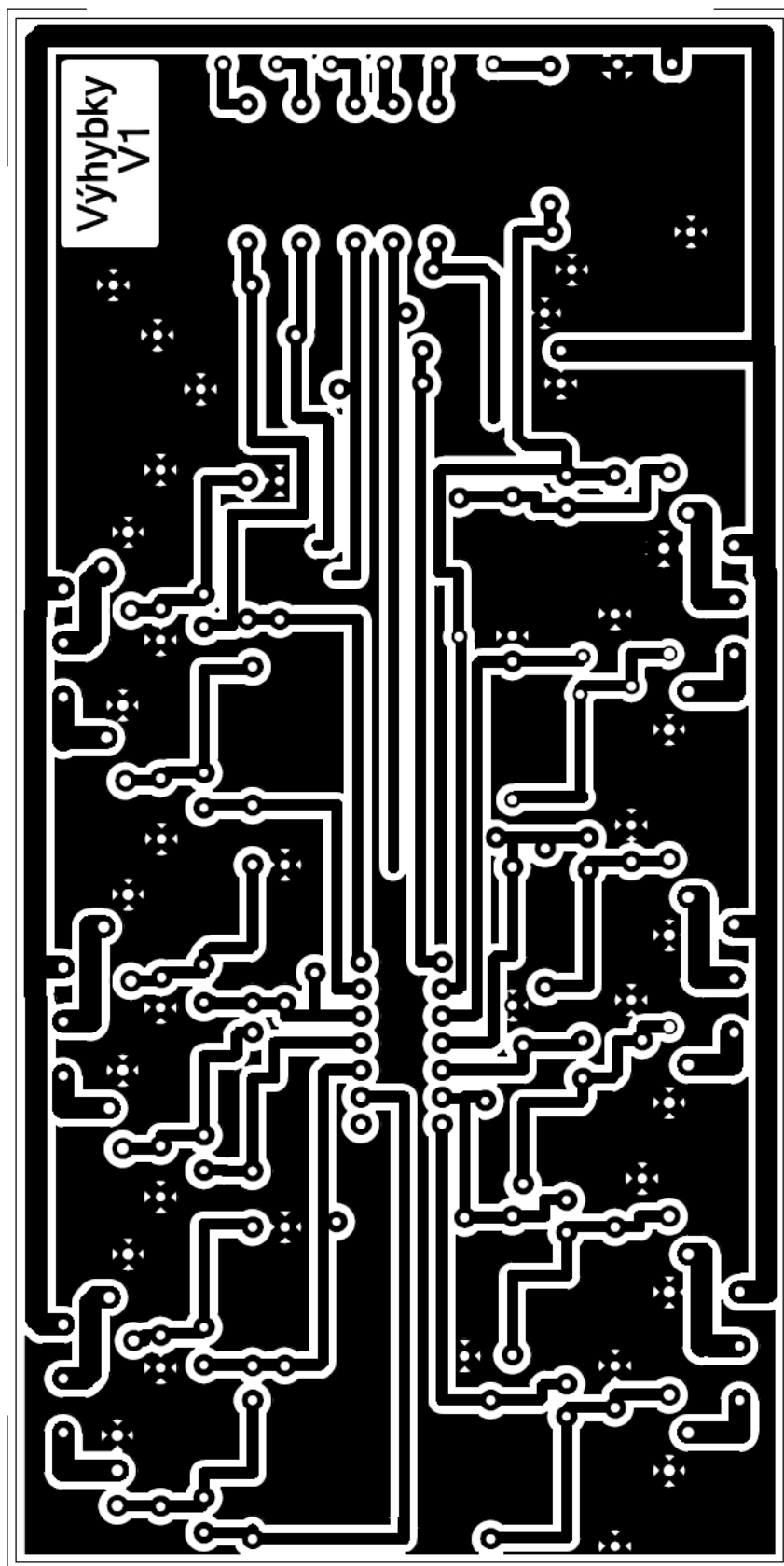


Obrázek I.5 Nastavení parametrů sériové komunikace v prostředí Mosaic

PŘÍLOHA P II: ŘIDICÍ ELEKTRONIKA VÝHYBEK



Obrázek II.1 Schéma řídicí elektroniky výhybek



Obrázek II.2 Návrh DPS řídicí elektroniky výhybek