



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

**Fakulta elektrotechnická**

**Katedra elektrotechnologie**

## **Řízení modelu technologického pracoviště**

**Control of technological workspace model**

Bakalářská práce

Studijní program: Elektrotechnika, energetika a management

Studijní obor: Aplikovaná elektrotechnika

Vedoucí práce: Ing. Karel Künzel, CSc.

**Jan Tluchoř**

---

**Praha 2016**

České vysoké učení technické v Praze  
Fakulta elektrotechnická

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jan Tlučoř**

Studijní program: Elektrotechnika, energetika a management

Obor: Aplikovaná elektrotechnika

Název tématu: **Řízení modelu technologického pracoviště**

### Pokyny pro vypracování:

1. Seznamte se s modelem v laboratoři katedry.
2. Seznamte se s programováním řídicího PLC Simatic.
3. Navrhněte vhodné školní úlohy pro logické řízení uvedeného modelu.
4. Vypracujte vzorová řešení navržených úloh.
5. Ověřte řešení na modelu řízeném logickým automatem.
6. Zpracujte dokumentaci řešených úloh.

### Seznam odborné literatury:

- [1] Perner, J.: Řízení a vizualizace technologického procesu, Diplomová práce ČVUT v Praze Fakulta elektrotechnická, Praha 2005
- [2] Javnický, P.: Revitalizace modelu technologického procesu, Bakalářská práce ČVUT v Praze Fakulta elektrotechnická, Praha 2014
- [3] Blažek J. Kurz programování PLC SIEMENS SIMATIC S7-300, FOXCON, dostupné na <http://www.foxon.cz/cs/blogs/65-kurz-programovani-plc-siemens-simatic-s7-300-dil-1.html>
- [4] Šmejkal, L.: Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 - seriál článků v časopise Automa, 2011/8 až 2014/3, 2011 až 2014, ISSN 1210-9592

Vedoucí: Ing. Karel Künzel, CSc.

Platnost zadání: do konce letního semestru 2016/2017

L.S.

Ing. Karel Dušek, Ph.D.

prof. Ing. Pavel Ripka, CSc.

vedoucí katedry

děkan

V Praze dne 30. 3. 2016

## **Prohlášení**

Čestně prohlašuji, že na bakalářské práci jsem pracoval samostatně na základě vlastních teoretických a praktických poznatků, konzultací a studia odborné literatury. Kompletní přehled je uvedený v seznamu použité literatury. Nemám žádný důvod proti použití této práce ve smyslu § 60 Zákona č. 121/2000 Sb., o autorském právě, o právech souvisejících s autorskými právy a o změně některých zákonů.

V Praze dne 27. 5. 2016

.....

Jan Tlučoř

## **Poděkování**

Rád bych touto cestou poděkoval vedoucímu své práce panu Ing. Karlu Künzelovi, CSc. za vlídný přístup a ochotu. Dále bych chtěl poděkovat zaměstnancům společnosti Continental Automotive Czech Republic, s.r.o. za vypůjčení komponentů a instalaci potřebného softwaru. Také děkuji Tomáši Mizerovi za cenné rady.

## **Abstrakt**

Tato bakalářská práce se zabývá tematikou programovatelných automatů. Začátek práce popisuje základní části modelu kolejisti. Věnuje se teorii programování PLC S7 - 300 a popisu základních i pokročilejších prvků logického řízení. Práce ukazuje různé možnosti řešení ovládání modelu vlaku, který jezdí na kolejisti. Výsledkem jsou okomentované fungující programy. Na konci je krátce zmíněna vizualizace SCADA systémem WinCC Flexible.

## **Klíčová slova**

PLC, S7 - 300, SIMATIC, SCADA, řízení

## **Abstract**

Main topics of my bachelor thesis are programmable controllers. Introduction of the thesis describes essential parts of the railyard model. The main part is focused on the theory of programming PLC S7 - 300 and describes the basic and advanced elements of logical controlling. There are few different options of controlling the train model running in railyard. Results involve commented funtional programs and information about visualization SCADA system called WinCC Flexible.

## **Key words**

PLC, S7 – 300, SIMATIC, SCADA, control

# Seznam zkratek

PLC	Programmable Logic Controller
PC	Personal Computer
HW	Hardware
CPU	Central Processing Unit
IP	Internet Protocol
IR	Infrared
LED	Light Emitting Diode
SCADA	Supervisory Control And Data Acquisition
HMI	Human Machine Interface
MPI	Multi Point Interface
FBD	Function Block Diagram
LAD	Ladder Diagram
STL	Statement List
OB	Organization Block
FB	Function Block
FC	Function
DB	Data Block

# Obsah

1. Úvod .....	1
1.1 Vysvětlení pojmů .....	2
2. Popis pracoviště .....	3
2.1 Hardware .....	3
2.2 Software .....	6
3. Popis PLC v laboratoři .....	8
3.1 Hlavní znaky .....	8
3.2 CPU a zdrojový modul .....	8
3.3 Vstupy a výstupy .....	9
3.4 Adresace u PLC v laboratoři .....	10
4. Programování .....	11
4.1 Programovací jazyky .....	12
4.1.1 LAD (Ladder diagram) .....	13
4.1.2 STL (Statement List) .....	14
4.1.3 FBD (Function Block Diagram) .....	15
4.2 Logické řízení .....	16
4.2.1 AND .....	16
4.2.2 OR .....	17
4.2.3 NOT .....	18
4.3 Další logické členy .....	19
4.3.1 Bitová logika .....	19
4.3.2 Časovač .....	19
4.3.3 Čítač .....	20
4.3.4 Komparátor .....	20



4.4	Zkušenosti z vývoje .....	21
4.5	Simulátor .....	22
4.6	Základní ovládání vlaku .....	23
4.7	Testování modulů.....	25
5.	Návrh vzorových úloh .....	26
	Úloha č. 1 .....	26
	Úloha č. 2 .....	26
	Úloha č. 3 .....	27
6.	Zdokumentované a popsání úlohy.....	28
	Úloha č. 1 .....	28
	Úloha č. 2 .....	29
	Úloha č. 3 .....	30
7.	Ověření funkčnosti.....	32
	7.1 Příprava před spuštěním .....	32
	7.2 Ověření funkčnosti.....	32
8.	WinCC Flexible .....	33
	8.1 Chybové hlášení při instalaci.....	33
	8.2 Vytváření vizualizace .....	33
	8.3 Vizualizace v učebně .....	35
9.	Závěrečné zhodnocení .....	36
	Literatura .....	37
	Seznam obrázků.....	38
	Seznam tabulek .....	40
	Přílohy .....	41

# 1. Úvod

Míra stále vyšší automatizace mě přiměla zabývat se problematikou programování PLC. Tato práce by měla posloužit jako návod a z části učební literatura pro další studenty, kteří se budou chtít programování automatů zabývat. Konkrétně se práce bude zabírat řízením železničního modelu v halové laboratoři T2:E1-3c.

Cílem této závěrečné práce je vytvoření vhodných školních úloh, které budou moci posloužit studentům jako učební pomůcka. Naučí je pracovat se základními logickými vazbami a operacemi, které mohou následně ověřit na reálném modelu, kterým je kolejiště se dvěma modely vlaků.

PLC, které je u modelu použito, je od společnosti Siemens. Označení je S7 - 300. Tento typ je na trhu už od roku 1997, řadí se tedy mezi starší modely. Je kompatibilní se systémy Windows 95 a novějšími.

V závěru práce jsou vhodně okomentované a fungující úlohy. Studenti by se v nich měli lehko zorientovat.

## 1.1 Vysvětlení pojmů

PLC - Programmable Logic Controller. Programovatelný logický automat. Více o PLC v kapitole 3.

SCADA - Supervisory Control And Data Acquisition. Dispečerské řízení a sběr dat. Jedná se o software, který monitoruje jiná technická zařízení a může je ovládat. Jedním ze SCADA softwarů je Control Web 5 nebo WinCC Flexible.

CPU - Central Processing Unit. Centrální procesorová jednotka. Je to základní elektronická část, která umí v počítači vykonávat strojové instrukce, kterými je počítač řízen.

IR - Infrared. Infračervené záření, kterým se předává informace mezi IR přijímačem a IR vysílačem. Záření není viditelné.

LED - Light Emitting Diod. Dioda emitující světlo.

HW - Hardware. Hmatatelná část.

HMI - Human Machine Interface. Rozhraní mezi strojem a člověkem.

MPI - Multi Point Interface. Způsob komunikace mezi PLC a počítačem.

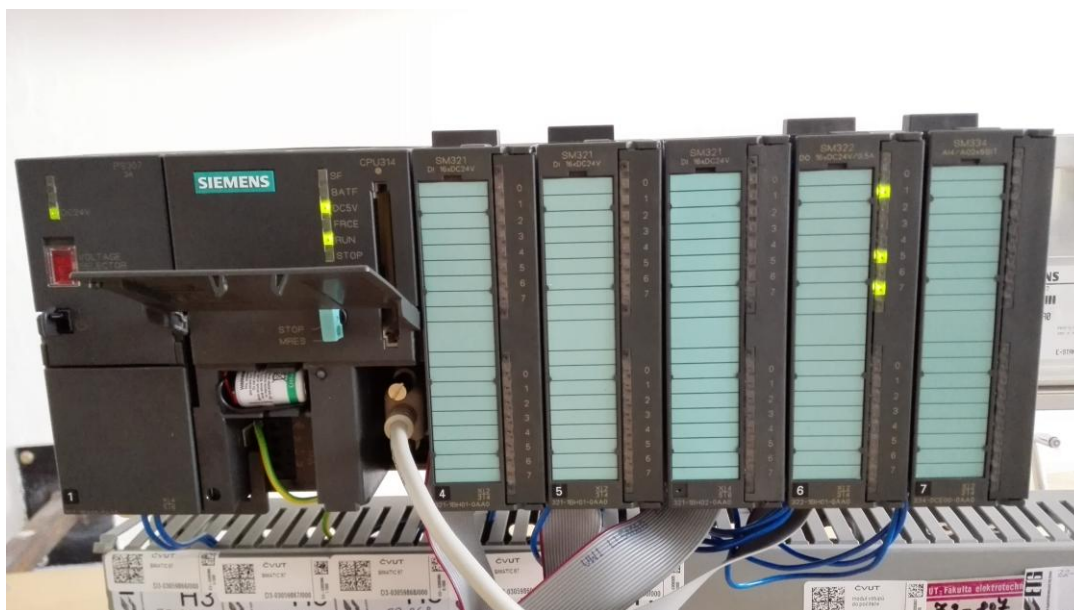
## 2. Popis pracoviště

Pracoviště jsem rozdělil na hardwarovou část a softwarovou část. Jsou zde zmíněny nejdůležitější komponenty, které jsou nezbytné pro řízení technologického procesu.

### 2.1 Hardware

#### S7 - 300

Jedná se o modulární systém, který umožňuje jednoduchou implementaci distribuované struktury. Vzhledem k uživatelsky nenáročnému zacházení ho lze použít jako vhodný ekonomický prostředek k řešení širokého spektra výkonově nenáročných úloh, jako jsou stroje pro speciální účely, textilní stroje, balící stroje, řízené systémy, obráběcí stroje a jiné speciální aplikace [1].



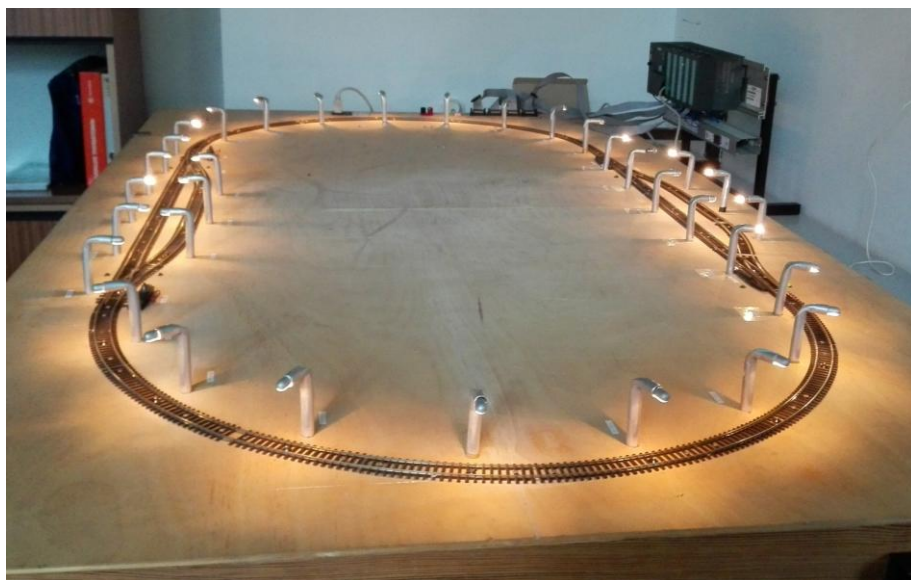
Obrázek 1: Čelní pohled na PLC S7 - 300

#### Stolní počítač

Bude zde k dispozici stolní počítač se systémem XP 2000, což splňuje minimální požadavky pro instalaci STEP 7. Paměť na disku je vyžadována v rozmezí 400 MB až 800 MB. Rozhraní je realizováno přes kartu CP5611. Tento slot je upevněn v zadní části počítače (jedná se o bílý konektor s 9 zdířkami).

## Kolejiště

Kolejiště je oválného tvaru umístěné na dřevěném podstavci o rozměrech 10x120x200 cm. Z druhé strany je tabule, která bude v době výuky překllopena do vertikální polohy a upevněna ke stěně sponami. Kolejiště má dvě stanice, které jsou uprostřed delších stran naproti sobě. Ke každé zastávce směřují vždy dvě výhybky. Tyto výhybky jsou napojeny jako výstupy k PLC. Dají se ovládat tlačítkem a přepínačem, které jsou u kolejiště. Tato tlačítka jsou připojena k PLC jako vstupy. Dále je tu umístěno 34 fototranzistorů, které jsou upevněny pod sloupky po obvodu celé trati. Ve sloupech jsou umístěny žárovky, které osvětlují kolejiště [1][2].



**Obrázek 2: Model kolejiště**

### Model vlaku

Jedná se o klasický model dopravního vlaku. Řízení vlaku je řešeno komunikací mezi IR vysílačem a IR přijímačem. IR vysílač je tvořen čtyřmi infračervenými diodami L53F3BT [9]. IR přijímač je tvořen dvěma fotodiodami SFH5110-36 [10], které jsou k sobě zapojeny paralelně. Jedna by nestačila, pokud by vlak zastavil vždy přesně pod daným osvětlovacím sloupkem a nebylo by ho možné dále ovládat [1][2].



**Obrázek 3: Vlak č. 1**



**Obrázek 4: Vlak č. 2**

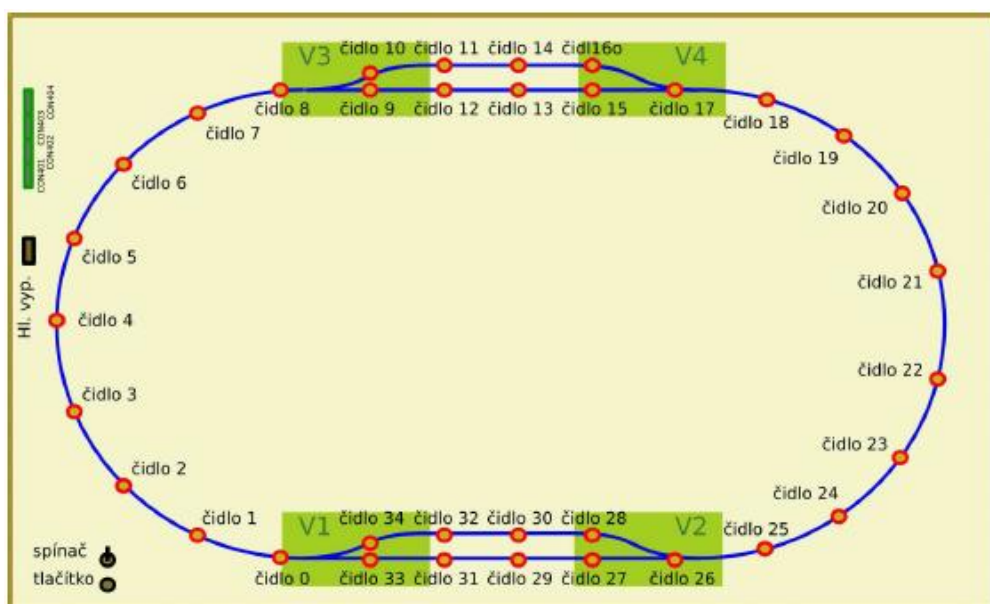
## Napájení

Napájení kolejiště je zde řešeno ze sítě, tedy 230 V. Ovšem toto napětí se musí transformovat na menší dle požadavků částí modelu. Napájení pro výhybky je doporučeno střídavé 16 V. Je tedy zvolený originální transformátor pro napájení výhybek PICO, 230 V / 16 V 1,2 A. Je doporučený výrobcem. Zdroj napětí pro napájení osvětlovací soustavy je 230 V / 12 V. Dále je zde zdroj napětí 5 V, který napájí integrované elektronické obvody, obvody snímání polohy vlaku [1].

## Senzorový systém

Senzorový systém je tvořen 34 fototranzistory. Fototranzistory jsou na obrázku 5 znázorněny jako čidla. Jsou umístěna po délce celé dráhy v pravidelných vzdálenostech. Průjezd vlaku zakrývá jednotlivá čidla, která jsou vyvedena jako vstupy do PLC karty.

Pro představu je zde nutné vidět rozmístění čidel:

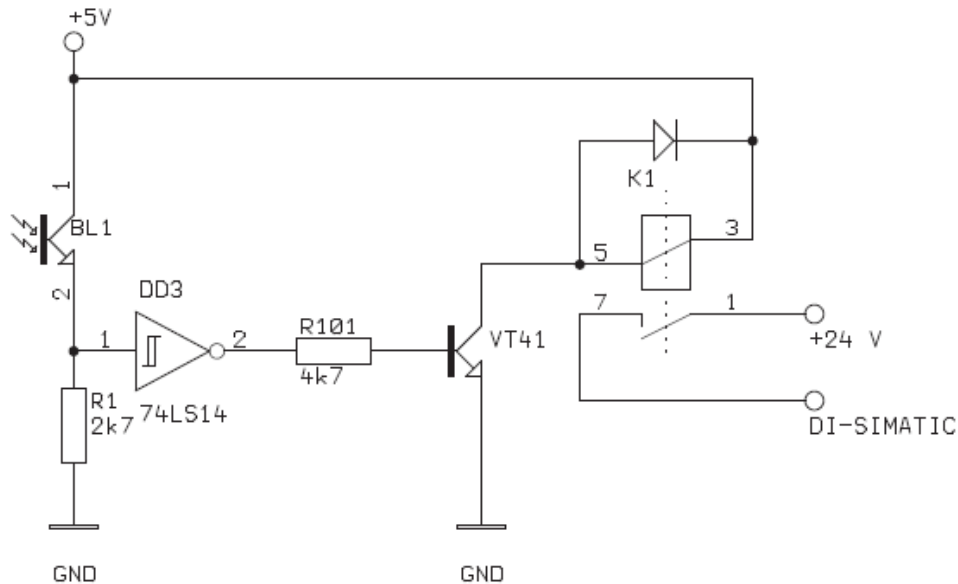


Obrázek 5: Rozmístění senzorů [1]

Nemůže nastat situace, kdy by pozorovatel nevěděl o stávající poloze vlaku. Vzdálenost mezi čidly je 17,5 cm a vlak má velikost menší. Vždy tedy zakryje právě jeden fototranzistor (čidlo). Obvod těchto čidel netvoří pouze fototranzistor BL1. Samotný fototranzistor BL1 je částí obvodu, který zprostředkovává signál do PLC. Celý obvod čidla se skládá ještě ze Schmittova klopného obvodu v invertorovém provedení DD3. Bipolární tranzistor VT41, který je sepnut přes R101, což je rezistor. U bipolárního tranzistoru VT 41

je u kolektoru umístěno relé K1, které spíná napětí na SIMATICu, které se nám zobrazí jako rozsvícená LED žárovka na kartě vstupů [1].

Pro lepší pochopení přikládám schéma zapojení:



Obrázek 6: Schéma senzového systému [1]

## 2.2 Software

### STEP 7

Tento software pochází od firmy Siemens. Je kompatibilní pouze s S7-300 / 400. Verze 5.3 je nainstalována ve stolním počítači. Nejnovější verze je 5.5. Tuto verzi mám nainstalovanou ve svém notebooku, abych nebyl odkázán pouze na stolní počítač v laboratoři. Verze 5.3 a novější jsou už kompatibilní se systémem XP 2000. Obsahuje v sobě SIMATIC Manager.

### PRODAVE 7

Jedná se o program, který je z balíčku STEP 7. Tento nástroj má za úkol vytvoření kanálu mezi počítačem a SIMATICem S7 přes rozhraní MPI [2].

## **Win CC Flexible**

SCADA systém, který pochází opět ze společnosti Siemens. Slouží k vizualizaci procesu nebo třeba k „vytažení“ funkčních tlačítek na panel. V průmyslu má velké využití, kdy se používá k usnadnění přístupu pro operátorské panely.



## 3. Popis PLC v laboratoři

V seznamu hardwaru jsme se dozvěděli, že v laboratoři se nachází typ S7- 300. Zde si informace o tomto systému rozvedeme.

### 3.1 Hlavní znaky

- Malý modulární řídicí systém pro střední výkonový rozsah
- Je možné rozšíření až na 32 modulů
- Zadní sběrnice je integrovaná
- Propojení mezi PC a PLC je za pomoci MPI

### 3.2 CPU a zdrojový modul

PLC systém se odlišuje od obyčejných počítačů nejenom tím, že zpracovává program cyklicky, ale i tím, že periferie PLC systému jsou přímo přizpůsobeny na připojení technologických procesů. U CPU modulu je přepínač, kterým se volí jednotlivé režimy. Vedle je řada LED, které indikují aktuální stavy PLC.

#### **Přepínač režimu**

M RES = Funkce mazání paměti

STOP = Režim stop

RUN = Program se vykonává (možnost pouze čtení z PC)

RUN-P = Program se vykonává (možnost čtení i zápisu z PC)

#### **Indikátor stavu**

SF = Skupinová chyba, může se jednat o chybu na modulu (možno diagnostikovat) nebo o vnitřní chybu CPU

BATF = Chyba baterie (baterie chybí nebo je vybitá)

DC5V = Indikátor vnitřního napětí 5 VDC

FRCE = Indikuje, když je min. jeden vstup / výstup vynucen

RUN = Svití, když program běží (bliká při restartu CPU)

STOP = Svití v režimu stop

### **Paměťová karta**

Do paměťové karty se ukládá obsah programu. Uložení proběhne i v případě výpadku napájení.

CPU v laboratoři má pouze integrovanou paměť. Slot pro kartu je u PLC S7- 300 od roku 2002.

### **Baterie**

Lithiová baterie.

### **MPI spojení**

Slouží k propojení mezi PC a PLC.

### **DP rozhraní**

Rozhraní pro přímé spojení mezi vstupními a výstupními moduly k CPU.

## **3.3 Vstupy a výstupy**

Vzhledem k tomu, že se jedná o modulární systém, jsou zde jednotlivé vstupní či výstupní karty popsány jako moduly:

- Modul zdroje
- Modul komunikačního procesoru
  - CPU 314 (pouze MPI komunikace)
- Signální moduly
  - Zde jsou 3 moduly digitálních vstupů (24 VDC, 120/230 VAC), jeden modul digitálních výstupů (24 VDC, relé) a také jeden vstupně výstupní analogový modul
- Funkční moduly, Interface moduly, Dummy moduly
  - Nejsou zde žádné

Informace o procesu dostává PLC od signálních čidel, které jsou spojené se vstupy PLC. Tato čidla mohou být například senzory polohy nebo i tlačítka a různé spínače.

Údaj určitého vstupu nebo výstupu v programu se označuje jako zadání adresy. Vstupy a výstupy PLC jsou většinou sdružené v 8 početních prvcích nebo bitech, které tvoří jeden byte. Každá tato množina má svoji adresu. Na jedné kartě je těchto bytů více (většinou 4), proto je nutné, aby každá adresa byla originální. Je předdefinovaná od výrobce.

### 3.4 Adresace u PLC v laboratoři



Obrázek 7: Popis konkrétního adresování vstupů / výstupů v laboratoři

Bitsy jsou očíslovány od bitu 0 až po bit 7. První bit shora bude mít adresu I0.0, kde I označuje, že se jedná o vstup, 0 je byte adresa a druhá 0 je bit adresa. S výstupy je to podobné. Jedná se tentokrát o byte 4 a bit adresa je 0. Celá adresa je tedy Q4.0.

Přehled vstupních i výstupních adres naleznete v HW konfiguraci (v následující kapitole se o ní dozvíte). Vyberte přes menu *View > Address Overview*.

## 4. Programování

Programování bude probíhat v prostředí SIMATIC manageru. Začneme tedy vytvořením nového projektu. Můžeme zvolit *Project Wizard*. Začneme HW konfigurací, která je nezbytná. Nová stanice se vloží do aktuálního projektu výběrem přes *Insert > Station > SIMATIC 300*. Zde si nejprve vložíme tzv. Rack, do kterého pak vkládáme jednotlivé karty či moduly. Důležité je dodržet přesné značení. Každý modul je originální, proto je velmi důležité dodržet i to poměrně dlouhé specifické značení.

### HW konfigurace v učebně:

Pozice 1: Zdroj 24 VDC / 5A (PS 307 2A)

Pozice 2: Procesorová karta (CPU 314)

Pozice 3: Rezervována (vyčleněno pro Interface modul)

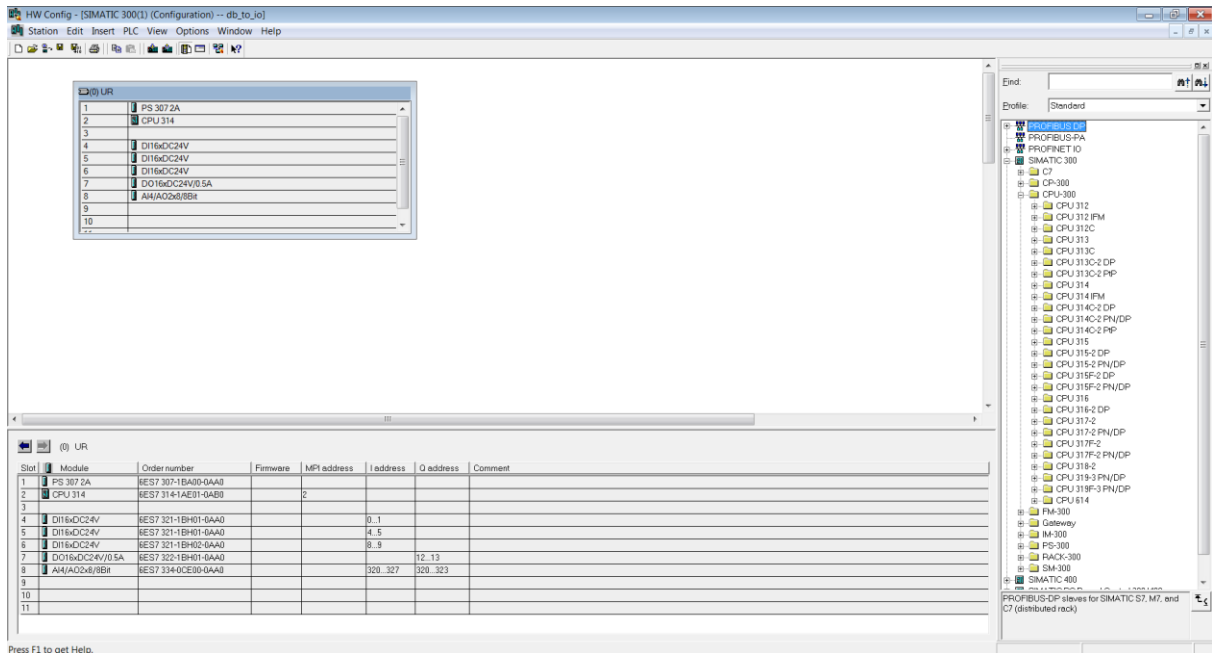
Pozice 4: Karta s 16 digitálními vstupy (DI16xDC24V)

Pozice 5: Karta s 16 digitálními vstupy (DI16xDC24V)

Pozice 6: Karta s 16 digitálními vstupy (DI16xDC24V)

Pozice 7: Karta s 16 digitálními výstupy (DO16xDC24V/O.5A)

Pozice 8: Analogové vstupy a výstupy (AI4/AO2x8/8Bit)



**Obrázek 8: Náhled nastavování HW konfigurace**

Nyní si vytvoříme program, kam vložíme jednotlivé bloky, ve kterých budeme psát kód.

#### Můžeme zvolit:

OB (organizační blok) - blok, který má hlavní prioritu a stále se opakuje.

FC (funkce) - funkce, obsahující dílčí část programu.

FB (funkční blok) - funkční blok má obdobné vlastnosti jako funkce. Navíc má vlastní paměť ve formě instance datového bloku.

DB (datový blok) – blok pro ukládání datových proměnných.

V následující podkapitole si ujasníme, v jakém jazyku budeme programovat. Prostředí nabízí 3 možnosti. STL (Statement List), LAD (Ladder Diagram) nebo FBD (Function Block Diagram).

## 4.1 Programovací jazyky

V prostředí manageru lze využívat tři programovací jazyky v souladu s normou ČSN EN 61 131 – 3 [4]. Vývojové prostředí nabízí tři z nich. Následující popis porovnává jejich vlastnosti.

Program v určitém jazyku lze různě přepnout na jiný za pomoci *View* v horním panelu. Mějte však na paměti, že konverze do jazyku STL nevyústí vždy v to nejlepší řešení. Při opačné konverzi z jazyka STL se může stát, že některé části nejsou přeloženy (nedochází ke ztrátě dat).

### **4.1.1 LAD (Ladder diagram)**

Programovací jazyk LAD, jinak také liniové či reléové schéma. Zápis programu vychází z dob reléové logiky, kdy místo procesorů či mikrokontrolérů nebo také integrovaných logických obvodů typu NAND, NOR či XOR, se využívala soustava vzájemně propojených relátek. S jejich pomocí lze snadno realizovat logické funkce AND, OR či NAND. A právě pomocí reléových schémat se snadno tyto logické soustavy „zapisovaly“. Svoji strukturou je pak prakticky ideální pro rychlé a přehledné zpracování velkého množství logických (spínaných vstupů a výstupů) a jejich čítání a časování [4][5].

Hlavní výhodou tohoto zápisu je však jasně definovaná posloupnost programu a tedy i velká přehlednost, která však někdy bývá na úkor kompaktnosti. U větších a složitějších programů či při masivnějším nasazení funkcí typu přesunu dat (např. funkce MOV) pak může délka zápisu dost rychle narůstat, až se zápis začne stávat nepřehledným. Vhodné je tak častější členění do podprogramů a masivní nasazení komentářů. Tento zápis bude vyhovovat lidem, kteří často pracují s elektrickými schématy [4][5].

**Výhody:**

- + Jasně definovaná posloupnost zápisu, kterou nelze porušit
- + Přehlednost zápisu (zvláště u menších programů)
- + Velmi rychlé programování logických operací s funkcemi čítání a časování
- + Ideální pro zpracování velkého počtu logických signálů (vstupů a výstupů)

**Nevýhody:**

- S rostoucí složitostí programu rychle narůstá jeho délka
- Méně pochopitelný pro „klasické programátory“

Tento programovací jazyk je vhodný pro realizaci logických řídicích sekcí a sekvencí programu a zpracování vstupních a výstupních signálů.

## 4.1.2 STL (Statement List)

Seznam instrukcí je textový zápis programu pro PLC. Program je tvořen v mnemotechnických instrukcích CPU, velmi blízkých instrukční sadě mikroprocesoru [4][7].

Seznam instrukcí je základním způsobem pro realizaci programu. Jedná se o nejnižší úroveň způsobu programování PLC a mimo některých systémů (např. programovatelní relé) je dostupný u všech PLC. Pomocí seznamu instrukcí je možno použít veškeré instrukce, které jsou v CPU dostupné. U grafických zobrazení může být tato možnost omezena [4][7].

**Výhody:**

- + Přesná definice chování programu
- + Paměťově úsporný program

**Nevýhody:**

- Nutnost znát nebo se alespoň dobře orientovat v příkazech a registrech
- Nepřehlednost programu

Tato forma zápisu je vhodná pro psaní krátkých vysoce optimalizovaných částí programu.

**4.1.3 FBD (Function Block Diagram)**

V FBD se posloupnost programu vyjadřuje soustavou za sebou propojených bloků, které realizují různé funkce, a však zde již není volná programovací plocha, ale vše je organizováno do řádků (linií) podobně jako v LAD režimu. Prakticky se dá říct, že FBD režim v prostředí SIMATIC manageru je v podstatě LAD režim, kde se u logických operací místo sériově-paralelního zapojování symbolů kontaktů relé využívá klasických značek hradel AND, OR apod. Osobně bych řekl, že tento zápis je vhodný pro toho, kdo potřebuje vlastnosti liniového schéma, tedy přehlednost a pevně dané členění programu, ale nevyhovuje mu „reléový“ zápis logických operací [4][5].

**Výhody:**

- + Definované grafické členění programu do řádků
- + Logické operace v podobě hradel
- + Přehledný zápis programu
- + Ideální pro zpracování velkého počtu logických signálů (vstupů / výstupů)

**Nevýhody:**

- Méně vhodný pro složitější zpracování analogových signálů
- Nevhodný pro programování složitých algoritmů

Schéma funkčních bloků je vhodné pro delší kódy, kde by byl jazyk liniových schémat značně nepřehledný.



## 4.2 Logické řízení

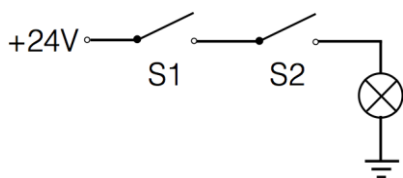
Už víme, jaký jazyk a proč použít. Když opomineme to, že neznáme syntaxi, která při programování PLC není tolik zásadní jako u jiných programovacích jazyků (C, Java), tak je důležité, abychom měli přehled v logických vazbách. S jednoduchou logikou můžeme realizovat i logiku poměrně složitých technologických procesů. Uvedeme si a vysvětlíme základní prvky pro logické řízení.

Ještě zmíním, že software SIMATIC je velmi nápomocný. Na postranním panelu je k výběru logických operací nepřeberné množství. Od jednoduché bitové logiky po komparátory a konvertory. Když chceme zjistit, jakou funkci má daný prvek, klikneme na jeho ikonu a stiskneme klávesu F1. Zobrazí se nám podrobné okno s kompletní nápovědou a malou ukázkou programu, kde je zahrnutý logický prvek.

Nyní vysvětlím několik základních logických členů - AND, OR a negaci NOT:

### 4.2.1 AND

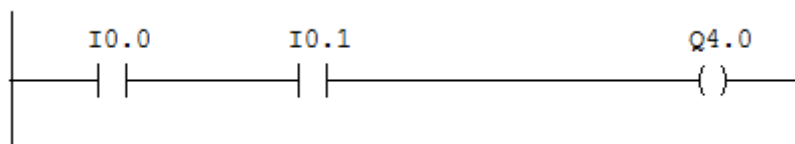
Na obrázku 9 je velmi dobře pochopitelná logická funkce AND. Žárovka bude svítit pouze v jednom případě a to, když budou sepnuty oba přepínače. Na obrázku 10 si můžeme představit, že vstupy I0.0 a I0.1 jsou tlačítka. Když je obě stiskneme, stane se cesta k výstupu vodivá a na výstup Q4.0 přivedeme napětí, které nám rozsvítí například žárovku, pokud na tento výstup bude připojena.



Obrázek 9: Logika AND ukázána na žárovce [2]

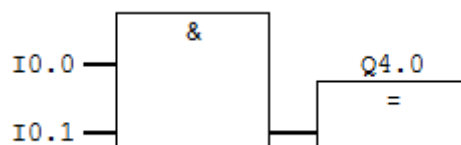
I 0.0	I 0.1	Q 4.0
0	0	0
0	1	0
1	0	0
1	1	1

Tabulka 1: Pravdivostní tabulka logiky AND



Obrázek 10: AND v jazyku LAD

A     I     0.0  
 A     I     0.1  
 =     Q     4.0

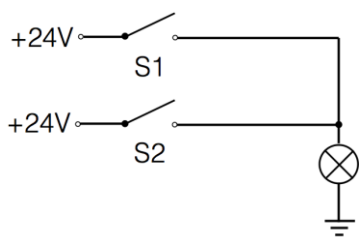


Obrázek 11: AND v jazyku STL

Obrázek 12: AND v jazyku FBD

## 4.2.2 OR

Zde je opět názorně vidět, že bude pouze jedna možnost, kdy se nám žárovka nerozsvítí. To se stane, jestliže oba přepínače budou rozepnuti. Žárovka bude svítit, když bude sepnutý S1 nebo S2. Bude svítit, i když budou sepnuty oba kontakty.



Obrázek 13: Logika OR ukázána na žárovce [2]

I 0.0	I 0.1	Q 4.0
0	0	0
0	1	1
1	0	1
1	1	1

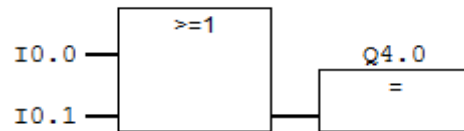
Tabulka 2: Pravdivostní tabulka logiky OR



**Obrázek 14: OR v jazyku LAD**

O	I	0.0
O	I	0.1
=	Q	4.0

**Obrázek 15: OR v jazyku STL**



**Obrázek 16: OR v jazyku FBD**

### 4.2.3 NOT

Negaci využíváme, když chceme znegovat hodnotu. Tedy z logické jedničky vytvořit logickou nulu nebo naopak. Negace má opravdu hojné využití v programování PLC.

## 4.3 Další logické členy

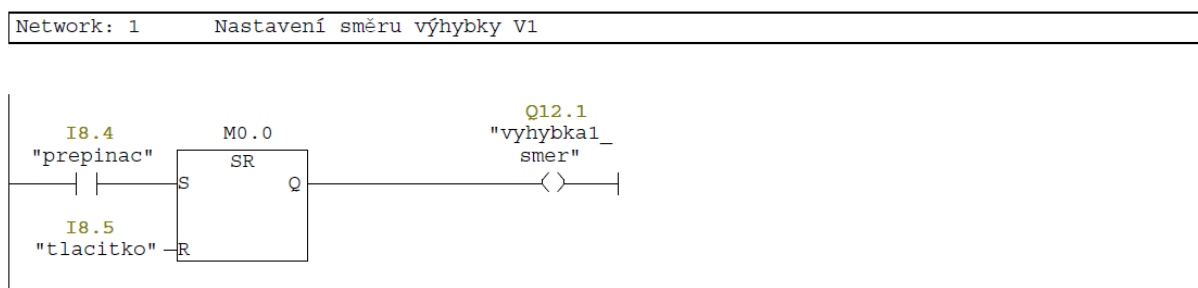
V této podkapitole zmiňuji další často používané logické členy. Jedná se o členy z bitové logiky, časovačů, čítačů a komparátorů.

### 4.3.1 Bitová logika

#### SR

Klopný obvod má nastavovací vstup SET a resetovací vstup RESET. Všechny vstupy i výstupy musí být datového typu BOOLEAN. Adresa obvodu se nastaví na volné místo paměťové proměnné např. M0.0.

Na příkladu je ukázáno ovládání směru výhybky za pomoci tlačítka a přepínače. Po přepnutí přepínače do horní polohy (v tu chvíli vstupu I8.4 odpovídá log. 1). Na Q12.1 a se dostává log. 1 a výhybka je připravena se odklonit od zastávky. Jakmile stiskneme tlačítko a je jedno, jestli na vstupu S je log. 1. Na výstupu se objeví log. 0.



Obrázek 17: SR klopný obvod

### 4.3.2 Časovač

#### S\_PEXT

Časovač S\_PEXT funguje následovně. Jakmile přivedeme log. 1 na vstup S, začne se odpočítávat čas na vstupu TV. Po tento čas bude na výstupu Q stále log. 1. Odpočítávání běží, i když je na vstupu S přivedena log. 0. Pokud na vstup přivedeme v průběhu odpočítávání opět log. 1, časomíra se znovu spustí. Vstup reset funguje jako u předešlého obvodu.

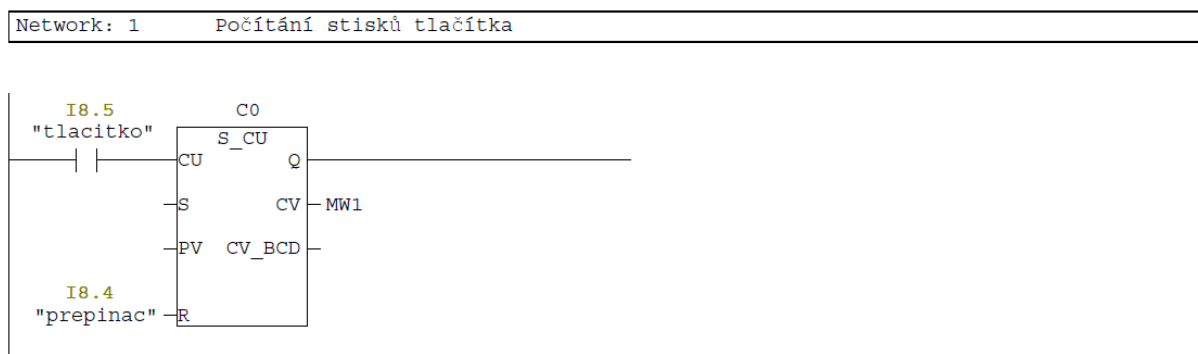
## S\_PULSE

Tento časovač funguje tak, že na vstup označený S přivedeme log. 1. Spustí se odpočítávání nastavené hodnoty v TV a na výstupu Q se objeví log. 1. Na výstupu je log. 1, dokud nepřivedeme na vstup log. 0 anebo neuplyne nastavený čas. Na výstup časovače dokážeme přivést log. 0 i funkcí reset. Jak na vstup funkce reset R přivedeme log. 1, časovač se vypne. Když přivedeme log. 1 na vstup S, opět se zapne odpočítávání.

### 4.3.3 Čítač

#### S\_CU

Jedná se o inkrementační čítač. V tomto zapojení logický člen počítá, kolikrát stiskneme tlačítko. Průběžně hodnotu ukládá na výstup CV do proměnné MW1. Přepnutím přepínače do horní polohy (log. 1) dojde k vynulování aktuální hodnoty v MW1.

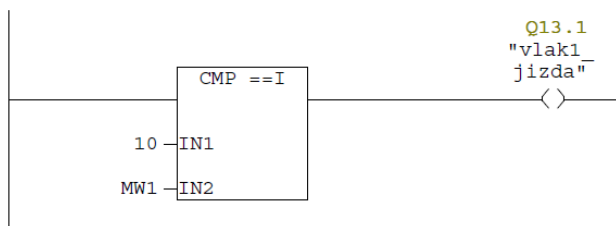


Obrázek 18: S\_CU čítač

### 4.3.4 Komparátor

#### EQ\_I

Zde se hodnota načtená z čítače výše porovnává s konstantou 10. Tento rovnostní komparátor vyšle na výstup log. 1 pouze, když se IN1 a IN2 rovnají. Z obrázku 18 a obrázku 19 tedy vyplývá, že pokud stiskneme 10x tlačítko, vlak pojedí. Vlak se zastaví, pokud přepneme přepínač do horní polohy (log. 1) nebo pokud opět stiskneme tlačítko.



Obrázek 19: EQ\_I komparátor

## 4.4 Zkušenosti z vývoje

Z mých zkušeností, které jsem získal po dobu, kdy jsem se zabýval logickým automatem, vyplývá několik doporučení:

### Merkerý

Nezapomeňte označovat logické členy paměťovými proměnnými. Podívejte se, jaké jsou ještě nepoužívané ve vnitřní paměti PLC. Paměťové proměnné (jinak také merkerý) se značí M0.0, pro časovače T0 a pro čítače C0. Každá proměnná by měla mít své specifické přiřazené číslo.

### Tabulka symbolů

Je užitečné si na začátku programování vytvořit tzv. Symbol table, kde si pojmenujete adresy vstupů a výstupů, popřípadě si je i vhodně okomentujete. Je to přehlednější nežli pracovat se samotnými adresami.

### Vhodné komentování

Nemusím dlouze zmiňovat důležitost komentářů. Zde se komentují dílčí Networky. Je tedy vhodné alespoň krátce popsat funkci daného Networku.

## Archivace

Archivace se spustí v hlavním menu *File > Archive*. Vytvoříte zálohu, kterou můžete zároveň zkomprimovat. Přes výběr *File > Retrieve* zálohu znovu otevřete.

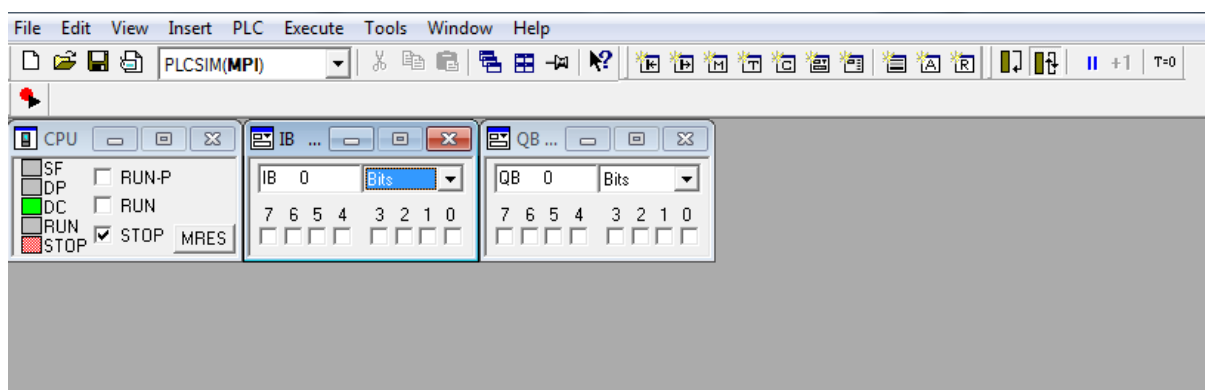
## Online režim

Naleznete ho v horní nabídce v podobě ikony s brýlemi. Online režim umožňuje pozorovat přímo, kde se v obvodu nachází log. 1 a kde ne. Tím můžeme pozorovat logické nesrovnalosti v kódu a včas je změnit. Online režim je totiž možný i za použití simulátoru. O simulátoru se dozvíte v další podkapitole.

## 4.5 Simulátor

Simulátor je velmi užitečná část programu pro ty, kteří nemají kontrolér u sebe. Chtějí si však funkčnost ověřit v pohodlí domova. Simulátor je obsažen v balíčku Step 7, proto s nainstalováním SIMATIC manageru je výhodné odsouhlasit i instalaci simulátoru.

Práce s ním je poměrně jednoduchá. Po otevření simulátoru se přidá automaticky políčko s CPU. Na výběr je pouze jedna PLC stanice, která je univerzální. To znamená, že se do ní nahraje jakákoliv HW konfigurace. Dále přidáme potřebné karty, které chceme simulovat. Dostupné jsou vstupy, výstupy až po čítače a časovače. Zkratka téměř jako reálná stanice.



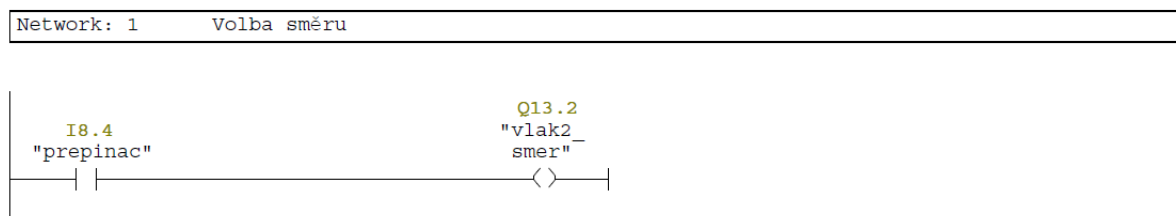
**Obrázek 20:** Pohled na prostředí simulátoru (vlevo CPU, uprostřed karta vstupů, vpravo karta výstupů)

Poté, co máte založený projekt a nastavenou HW konfiguraci, můžete projekt spustit a jako u skutečné stanice, prohlížet v online režimu. Rozdíl je, že pokud je něco špatně, projeví se to jako chyba v programu.

## 4.6 Základní ovládání vlaku

### Ovládání směru a řízení pohybu vlaku

Směr vlaku je umístěný v PLC na adrese výstupu Q13.2 a samotný pohyb vlaku ovládá výstup na adrese Q13.3. Přesná funkce Networku níže je: Přepneme přepínač do horní polohy (objeví se na vstupu log. 1) a vlak jede po směru hodinových ručiček. Přepneme přepínač do spodní polohy (objeví se na vstupu log. 0) a vlak vyjede proti směru hodinových ručiček. Stejná logika platí i u vlaku č. 1.



Obrázek 21: Ovládání směru vlaku

### Ovládání výhybky

Ovládání výhybek můžeme realizovat za pomoci tlačítka a přepínače. Ovládání výhybky se skládá ze dvou částí. Ze směru výhybky a impulzu, který výhybku zapne (přesune na požadovaný směr). Směr výhybky V1 je možné ovládat výstupem na adrese Q12.1. a impulz je potřeba přivést na výstup s adresou Q12.0. Pokud bude na výstupu Q12.1 log. 0, výhybka bude nastavena na směr k zastávce. Pokud bude na zmiňovaném výstupu log. 1, výhybka bude nastavena na směr mimo zastávku.

Pro spuštění výhybky je potřeba krátký impulz. Musíme mít stále na paměti, že impulz, který přepne výhybku, musí být kratší než 250 ms. Tento čas se nesmí překročit, aby nedošlo k poničení výhybky. Tomuto intervalu odpovídá krátké stlačení tlačítka [2].



Network: 1      Nastavení směru výhybky V1



Network: 2      Impulz, který přepne výhybku

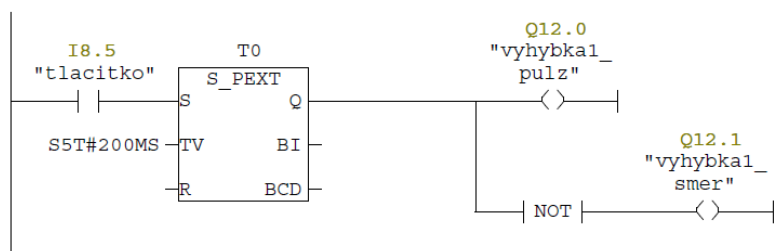


**Obrázek 22: Ovládání výhybky V1**

### Ovládání výhybky za pomoci časovače

Ovládání výhybky za pomoci časovače je určitě čistší řešení. Časovače jsme si probrali už dříve, takže zde funkci vysvětlím krátce: Po stisku tlačítka se výhybka V1 nastaví na směr k zastávce a zároveň vznikne dostatečně dlouhý pulz, který výhybku V1 přepne. Po vypršení časomíry bude směr výhybky připraven na odklonění od zastávky.

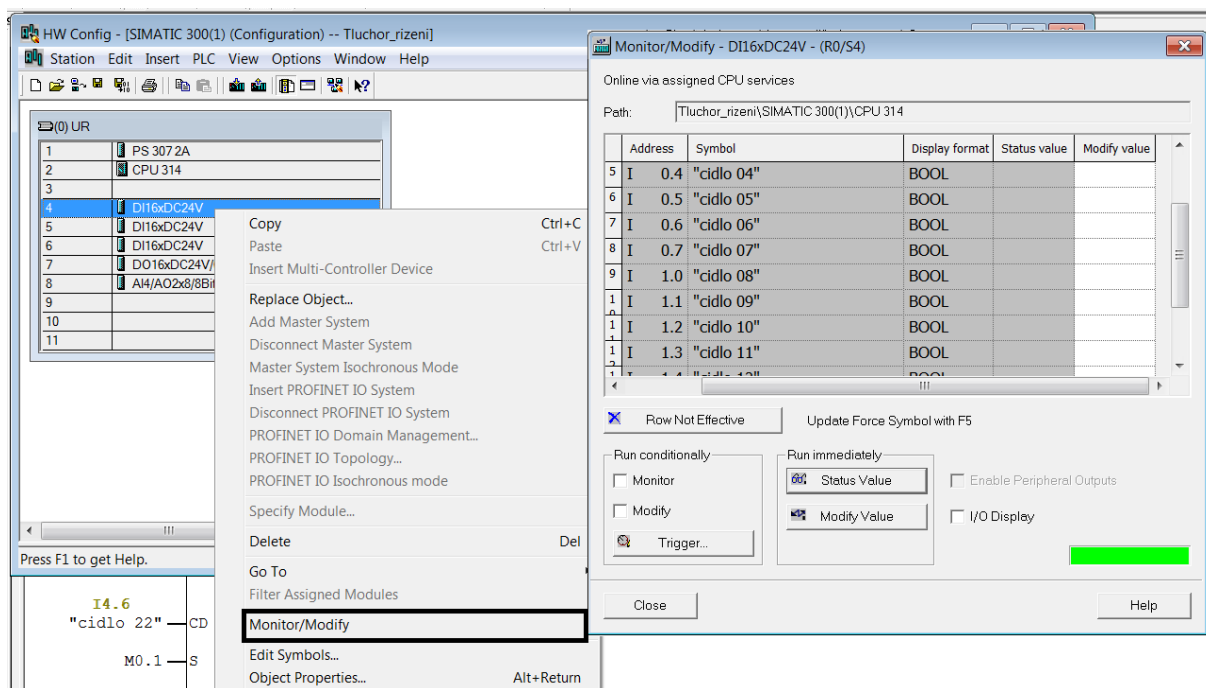
Network: 1      Nastavení směru a sepnutí výhybky V1



**Obrázek 23: Ovládání výhybky V1 za pomoci časovače**

## 4.7 Testování modulů

Při prvotním spojení mezi PLC a počítačem jsem za pomoci nástroje HW Config přistoupil přímo k symbolice proměnných u modulů v laboratoři. Díky tomuto nástroji můžete přímo testovat konfigurované moduly. Signály vstupních modulů mohou být „kontrolovány“ a signály výstupních modulů mohou být „ovládány“, jak plyne z anglického popisu.



Obrázek 24: Ukázka testování vstupního modulu

Klikneme pravým tlačítkem myši na modul v HW konfiguraci, který chceme testovat. Vybereme *Monitor/Modify*. Zobrazí se nám tabulka, kde vidíme adresy, symboliky a formáty vstupů. Testování probíhá stiskem kolonky *Modify Value* a *Status Value*. Pokud je vše správně, objeví se nám hodnota TRUE jak ve sloupci *Modify Value*, tak i ve sloupci *Status Value*. Zde není vidět žádná hodnota, protože jsem v tu chvíli nebyl připojen k PLC.

## 5. Návrh vzorových úloh

V následující kapitole popíši návrh úloh od zárodku až po jeho realizaci a odzkoušení. Vymyslel jsem 3 úlohy o různé složitosti. Kompletní kódy naleznete na přiloženém CD.

Častěji budu používat vlakový model číslo 2, protože je kratší a méně náchylnější na vykolejení, jak jsem zjistil při úvodních zkouškách. V úlohách počítám s mírnou setrvačností dojezdu vlaků (cca 2 zastávky). Dále je nutné počítat s vyšší rychlostí vlaku č. 2 oproti vlaku č. 1, který má defaultně nastavenou nižší rychlost (při vyšší rychlosti by docházelo k častému vykolejení). Vlaky mají určité zpoždění při rozjezdu (vlak č. 1 má toto zpoždění o něco menší). Dle mého názoru, je důvodem zpoždění vliv dekódování na přijímací straně. Pan Perner v rámci jeho diplomové práce vytvořil svůj vlastní proprietární protokol [1].

### Úloha č. 1

#### **Zadání:**

Přiložte vlak č. 2 na libovolnou část kolejiště. Vlak pojedje po směru hodinových ručiček. Pohyb vlaku bude realizován držením tlačítka. Výhybky budou odkloněné od zastávek. Jakmile se tlačítko přestane držet, model vlaku se zastaví.

### Úloha č. 2

#### **Zadání:**

Přiložte model vlaku na libovolnou část kolejiště. Vlak se přepnutím přepínače do spodní polohy rozjede proti směru hodinových ručiček a bude projíždět bližší zastávku po krátkém stisku tlačítka (zastávka mezi výhybkami V1 a V2). Jakmile vlak projede třikrát libovolnou zatáčkou, zastaví. Stiskem tlačítka se proces zopakuje.

## Úloha č. 3

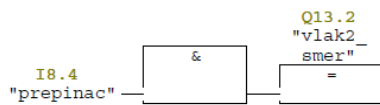
### **Zadání:**

Zde budou disponovat dva modely vlaků. Vlaky se rozjedou pouze, když bude jeden stát na čidle 3 a druhý na čidle 20 (vlak č. 2 postavte na čidlo 3 a vlak č. 1 na čidlo 20). Po splnění této podmínky se vyčká 10 s a oba vlaky se rozjedou stejným směrem (proti směru hodinových ručiček). Vlak č. 2, který pojede první k výhybce V1, zajede na zastávku. Vyčká určitý čas a vyjede ze zastávky. Jakmile vjede do zastávky vlak č. 1, vlak č. 2 zastaví. Vlak č. 1 zastaví stiskem tlačítka.

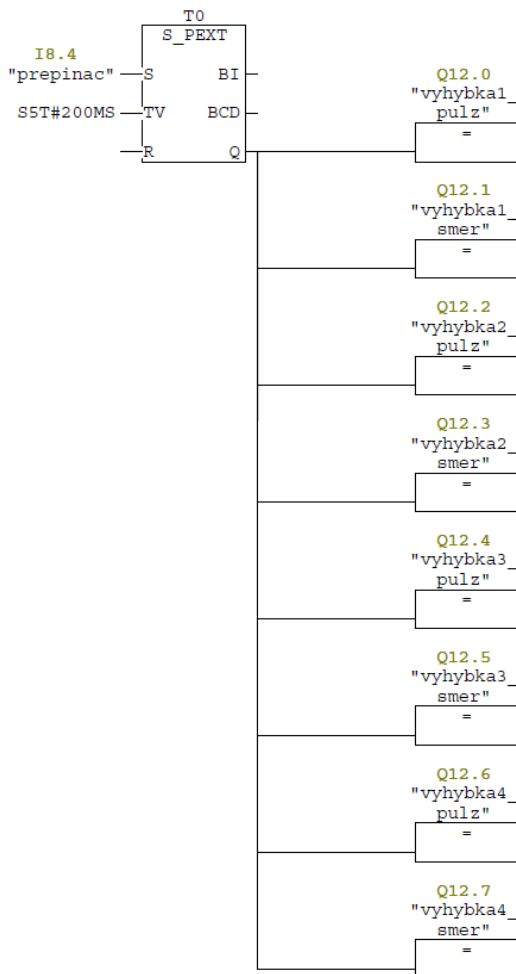
## 6. Zdokumentované a popsané úlohy

### Úloha č. 1

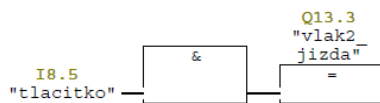
Network: 1 Volba směru



Network: 2 Nastavení výhybek



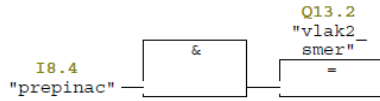
Network: 3 Jízda



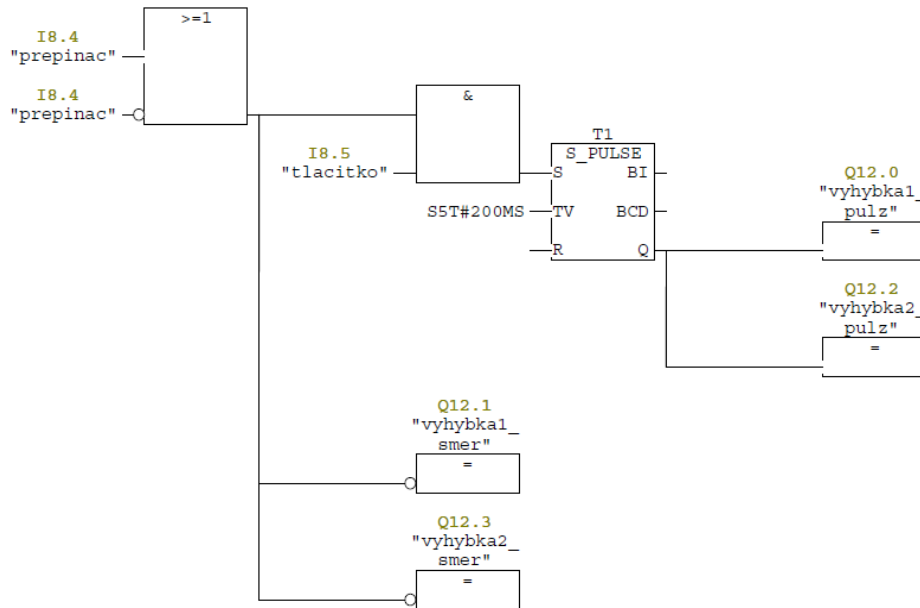
Obrázek 25: Úloha č. 1 v jazyku FBD

## Úloha č. 2

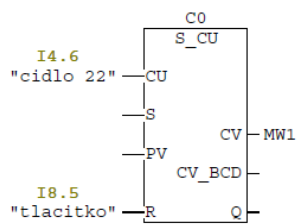
Network: 1      Určení směru



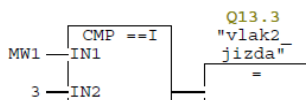
Network: 2      Aktivace výhybek V1 a V2



Network: 3      Počítání projetí zatáčkou



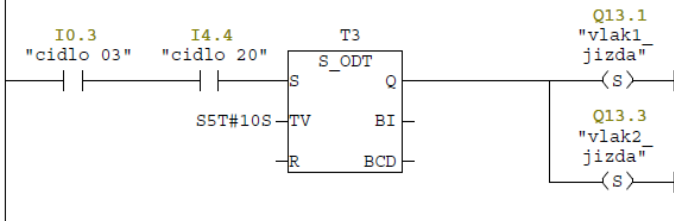
Network: 4      Porovnání počtu projetí zatáčkou



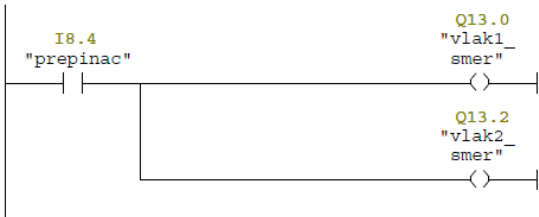
Obrázek 26: Úloha č. 2 v jazyku FBD

# Úloha č. 3

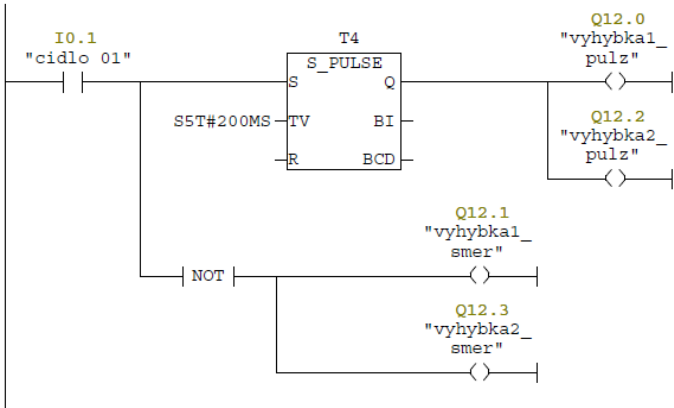
Network: 1 Zpoždění rozjezdu vlaků



Network: 2 Určení směru obou vlaků



Network: 3 Sepnutí výhybky V1 a V2 k zastávce



Network: 4 Zastavení vlaku č. 2 v zastávce



Network: 5      Odjetí vlaku č. 2 ze zastávky



Network: 6      Zastavení vlaku č. 1



**Obrázek 27: Úloha č. 3 v jazyku LAD**



## 7. Ověření funkčnosti

Zde se dostáváme od teorie k praxi, kde ověřím svoje programy na reálném modelu. Úlohy jsou vloženy do jednotlivých funkcí. Vzhledem k tomu, že jediný OB1 se cyklicky opakuje a má nejvyšší prioritu, je nezbytné, aby se vždy jednotlivá funkce vložila do tohoto organizačního bloku.

Model vlaku je nutné položit správnou polaritou na kolejiště (uslyšíte mírné pískání). Doba rozjezdu vlaků se pohybuje mezi 1 až 3 sekundami.

### 7.1 Příprava před spuštěním

1. Otevřete přední kryt CPU a zkontrolujte, zda je zasunuta záložní baterie.
2. Vhodně připojte kabeláž z PLC k modelu kolejiště.
3. Připojte napájení kolejiště i PLC
4. Reset musí být proveden během prvního spuštění (držte spínač na pozici MRES po dobu 3 s)
5. Připevněte IR vysílač na železnou destičku na stropě.
6. Propojte MPI spojením PLC s kartou v počítači.

### 7.2 Ověření funkčnosti

Všechny úlohy jsou zdokumentovány videozáznamem pořízeným na mobilní telefon. Naleznete v příloze 2.

## **8. WinCC Flexible**

Jedná se o SCADA software, pracující na platformě Windows XP / 7, který je vhodný pro všechny náročné aplikace ve všech oblastech průmyslu. Slouží k simulaci nebo spíše vizualizaci procesu na panelu.

### **8.1 Chybové hlášení při instalaci**

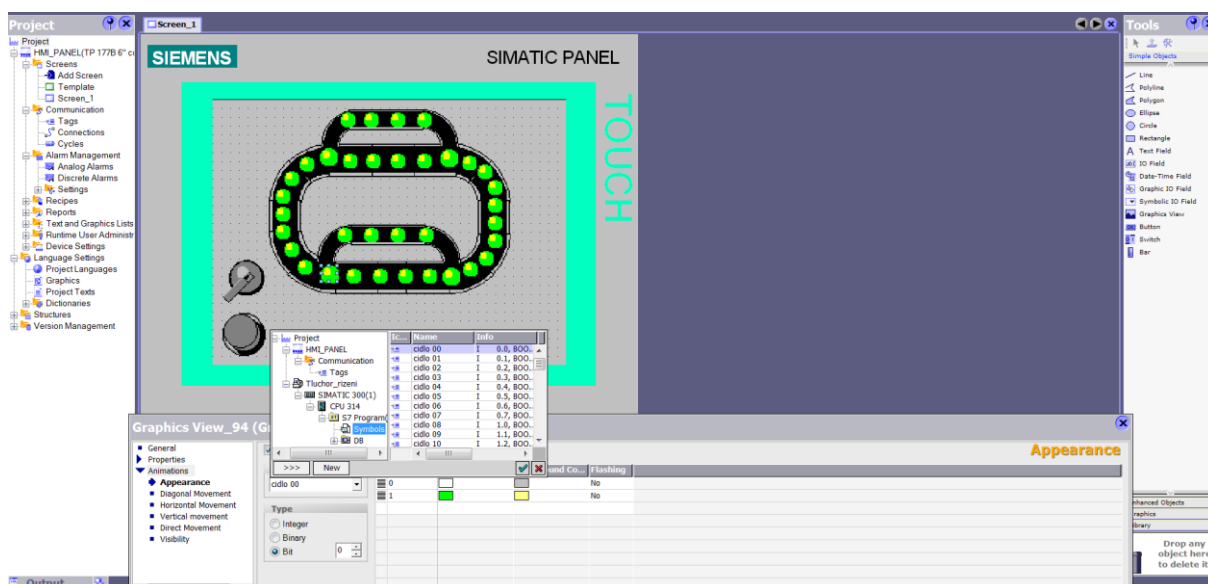
Pokud máte počítač HP, postupujte obezřetně s instalací tohoto software. Při instalaci dochází k dobře známé systémové chybě, zvané slangově též jako „Modrá smrt“. Počítač není schopen se z ní zotavit.

Předejít této chybě můžete za pomoci odinstalování driverů klávesnice HP. Podrobnější návod je ve zdrojích [6].

### **8.2 Vytváření vizualizace**

Vizualizace v tomto prostředí se skládá z několika kroků. Nejprve je nutné vybrat stanici, kterou používáte (S7 - 300). Dále je nutné vybrat, jaké máte spojení mezi PC a PLC (MPI). Pokračujete vybráním HMI panelu, na kterém bude proces spuštěn (TP 177B 6“ color PN / DP).

Jako u jiných SCADA softwarů začnete tvorbou grafických objektů, které se nejvíce podobají reálným objektům. Jakmile budete mít grafiku vytvořenou, začnete s párováním vstupů a výstupů.



**Obrázek 28: Náhled projektu v prostředí WinCC**

Přiřazení objektů konkrétní adresy je poměrně snadné, poněvadž lze projekt ve WinCC přidat do SIMATICu. Pak už stačí kliknout na objekt a spárovat ho s adresou, kterou vyberete ze seznamu symbolů, který máte vytvořený v SIMATICu. Zvolíte datový typ a informace, jak bude objekt vypadat v různých pravdivostních stavech.

### 8.3 Vizualizace v učebně

Program ve WinCC se mi podařilo spárovat se SIMATICem za pomoci návodu [8]. Spárovaný program jsem nahrál z PC do panelu ethernetovým kabelem. Musel jsem při tom nastavit specifickou IP adresu. Při spojení HMI panelu s PLC (převodníkem MPI) došlo k chybě ve spojení. Na vině může být zastaralejší verze CPU. Pozitivní zprávou je, že při použití simulátoru mi vizualizace procesu fungovala správně.



Obrázek 29: Náhled vizualizace

## 9. Závěrečné zhodnocení

Cílem práce bylo seznámit se s programováním PLC v laboratoři a tyto znalosti využít na vytvoření vzorových školních úloh. K tomu jsem využil technologický proces – již sestrojený model kolejistič včetně dalších komponentů. Své znalosti o logickém řízení jsem využil k tvorbě jednodušších programů, které jsem ověřoval za pomoci simulátoru.

Průběžně jsem své nově nabyté znalosti vkládal do této práce. Vždy jsem se snažil vyvarovat chybám a zdůraznit ty věci, které pro mě byly v začátcích hůře pochopitelné.

V závěru práce jsem navrhl několik vzorových úloh. Řešení úloh bylo zaměřeno tak, aby čtenář využil co nejvíce informací, které se dozvěděl z této práce. Funkčnost jednotlivých úloh je zdokumentovaná na videích v příloze.

Dále jsem zde navíc vytvořil vizualizaci procesu za pomoci WinCC Flexible. Bohužel se mi nepodařilo navázat spojení mezi PLC a dotykovým panelem. Pouze jsem ověřil funkčnost na simulátoru připojeném k programu.

Doufám, že úlohy poslouží studentům jako pomůcka a neodradí je od další práce s programovatelnými automaty. U modelu kolejistič je určitě mnoho prostoru pro inovace.

# Literatura

- [1] Perner, J.: Řízení a vizualizace technologické procesu, Diplomová práce ČVUT v Praze Fakulta elektrotechnická, Praha 2005.
- [2] Javnický, P.: Revitalizace modelu technologického procesu, Bakalářská práce ČVUT v Praze Fakulta elektrotechnická, Praha 2014.
- [3] Blažek, J.: Kurz programování PLC SIEMENS SIMATIC S7-300, dostupné na <http://www.foxon.cz/cs/blogs/65-kurz-programovani-plc-siemens-simatic-s7-300-dil-1.html>
- [4] Šmejkal, L: Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 - seriál článků v časopise Automa, 2011/8 až 2014/3, 2011 až 2014, ISSN 1210-9592
- [5] Programovací režimy pro PLC. *Automatizace* [online]. [cit. 2016-04-27]. Dostupné z: <http://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-611313-codesys>
- [6] Industry Online Support. *Support Industry Siemens* [online]. [cit. 2016-05-04]. Dostupné z: <https://support.industry.siemens.com/cs/document/58217334/was-ist-bei-der-installation-von-wincc-flexible-2008-sp3-oder-wincc-v11-sp2-auf-einem-hp-notebook-zu-beachten?dti=0&lc=de-WW>
- [7] ZOBRAZENÍ PROGRAMU V PLC. *PLC-automatizace* [online]. [cit. 2016-05-18]. Dostupné z: <http://plc-automatizace.cz/knihovna/program/zobrazeni-programu.html>
- [8] Blažek, J.: Kurz programování WinCC Flexible. *FOXCON* [online],[cit. 2016-05-20]. Dostupné z: <http://www.foxon.cz/blogs/category/19-kurz-wincc-flexible.html>
- [9] Kingbright. Datasheet L53F3BT. [CD]. <153f3bt.pdf>.
- [10] Siemens. Datasheet SFH5110-36. [CD]. <5110.pdf>.

# Seznam obrázků

Obrázek 1: Čelní pohled na PLC S7 - 300 .....	3
Obrázek 2: Model kolejště .....	4
Obrázek 3: Vlak č. 1 .....	4
Obrázek 4: Vlak č. 2 .....	4
Obrázek 5: Rozmístění senzorů [1].....	5
Obrázek 6: Schéma sensorového systému [1].....	6
Obrázek 7: Popis konkrétního adresování vstupů / výstupů v laboratoři.....	10
Obrázek 8: Náhled nastavování HW konfigurace.....	12
Obrázek 9: Logika AND ukázána na žárovce [2] .....	16
Obrázek 10: AND v jazyku LAD .....	17
Obrázek 11: AND v jazyku STL .....	17
Obrázek 12: AND v jazyku FBD.....	17
Obrázek 13: Logika OR ukázána na žárovce [2] .....	17
Obrázek 14: OR v jazyku LAD .....	18
Obrázek 15: OR v jazyku STL.....	18
Obrázek 16: OR v jazyku FBD.....	18
Obrázek 17: SR klopný obvod.....	19
Obrázek 18: S_CU čítač.....	20
Obrázek 19: EQ_I komparátor .....	21
Obrázek 20: Pohled na prostředí simulátoru (vlevo CPU, uprostřed karta vstupů, vpravo karta výstupů).....	22
Obrázek 21: Ovládání směru vlaku.....	23
Obrázek 22: Ovládání výhybky V1 .....	24
Obrázek 23: Ovládání výhybky V1 za pomoci časovače .....	24

Obrázek 24: Ukázka testování vstupního modulu .....	25
Obrázek 25: Úloha č. 1 v jazyku FBD.....	28
Obrázek 26: Úloha č. 2 v jazyku FBD.....	29
Obrázek 27: Úloha č. 3 v jazyku LAD .....	31
Obrázek 28: Náhled projektu v prostředí WinCC.....	34
Obrázek 29: Náhled vizualizace.....	35



## Seznam tabulek

Tabulka 1: Pravdivostní tabulka logiky AND.....	16
Tabulka 2: Pravdivostní tabulka logiky OR .....	17
Tabulka 3: Tabulka vstupů a výstupů.....	41

# Přílohy

## Příloha 1 – Tabulka vstupů a výstupů

Symbol	Adresa	Datový typ
cidlo 00	I 0.0	BOOL
cidlo 01	I 0.1	BOOL
cidlo 02	I 0.2	BOOL
cidlo 03	I 0.3	BOOL
cidlo 04	I 0.4	BOOL
cidlo 05	I 0.5	BOOL
cidlo 06	I 0.6	BOOL
cidlo 07	I 0.7	BOOL
cidlo 08	I 1.0	BOOL
cidlo 09	I 1.1	BOOL
cidlo 10	I 1.2	BOOL
cidlo 11	I 1.3	BOOL
cidlo 12	I 1.4	BOOL
cidlo 13	I 1.5	BOOL
cidlo 14	I 1.6	BOOL
cidlo 15	I 1.7	BOOL
cidlo 16	I 4.0	BOOL
cidlo 17	I 4.1	BOOL
cidlo 18	I 4.2	BOOL
cidlo 19	I 4.3	BOOL
cidlo 20	I 4.4	BOOL
cidlo 21	I 4.5	BOOL
cidlo 22	I 4.6	BOOL
cidlo 23	I 4.7	BOOL
cidlo 24	I 5.0	BOOL
cidlo 25	I 5.1	BOOL
cidlo 26	I 5.2	BOOL
cidlo 27	I 5.3	BOOL
cidlo 28	I 5.4	BOOL
cidlo 29	I 5.5	BOOL
cidlo 30	I 5.6	BOOL
cidlo 31	I 5.7	BOOL
cidlo 32	I 8.0	BOOL
cidlo 33	I 8.1	BOOL
cidlo 34	I 8.2	BOOL
prepinac	I 8.4	BOOL
tlacitko	I 8.5	BOOL
vyhybka1_pulz	Q 12.0	BOOL
vyhybka1_smer	Q 12.1	BOOL
vyhybka2_pulz	Q 12.2	BOOL
vyhybka2_smer	Q 12.3	BOOL
vyhybka3_pulz	Q 12.4	BOOL
vyhybka3_smer	Q 12.5	BOOL
vyhybka4_pulz	Q 12.6	BOOL
vyhybka4_smer	Q 12.7	BOOL
vlak1_smer	Q 13.0	BOOL
vlak1_jizda	Q 13.1	BOOL
vlak2_smer	Q 13.2	BOOL
vlak2_jizda	Q 13.3	BOOL

## Tabulka 3: Tabulka vstupů a výstupů

## Příloha 2 – CD s videozáznamy vzorových úloh a programy