

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



BAKALÁŘSKÁ PRÁCE

Vizualizace a řízení programovatelnými
automaty ControlLogix

Praha, 2012

Autor: Pavel Šimek

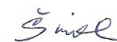
Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce panu Ing. Jindřichu Fukovi za jeho cenné rady a poskytnutí zázemí v laboratoři. Dále bych rád poděkoval své přítelkyni, rodině a všem, kteří mě během studia podporovali.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 24. 5. 2012



podpis

Abstrakt

Tato bakalářská práce se zabývá vizualizací a řízením programovatelnými automaty. Pro tyto účely byly využity moderní programovatelné automaty ControlLogix od firmy Rockwell Automation, které se nacházejí v laboratoři Allen-Bradley (K23) na katedře Řídicí techniky.

Jelikož hlavním cílem je podpora výuky, vzniklo v rámci mé práce několik zajímavých příkladů řízení a vizualizací, jejichž úkolem je naznačit studentům možnosti programovatelných automatů ControlLogix. Pro názorné seznámení s programovatelnými automaty bylo využito laboratorního modelu podavače, pro pokročilejší řízení pak modelu vodáren. V rámci mé práce vzniklo také několik studijních návodů popisujících práci se softwarovými nástroji firmy Rockwell Automation.

Abstract

This work deal with visualisation and control using programmable logic controllers. For these purposes the modern programmable controllers ControlLogix from Rockwell Automation were used. These controllers are located in the Allen-Bradley Laboratory at the Department of Control Engineering.

Since the main objective of this work is to support teaching, several interesting examples of visualisation and control were created. The aim of this examples is to show basic possibilities of ControlLogix programmable controllers. For basic control the position device laboratory model was used, for advanced examples coupled tanks were employed. Also several study guides describing work with Rockwell Automation software were created.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Pavel Šimek**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Vizualizace a řízení programovatelnými automaty ControlLogix**

Pokyny pro vypracování:


1. Seznamte se s výukovými modely podavačů a vodáren v laboratoři KN:E-23, které je možné řídit programovatelným automatem ControlLogix, a s vizualizačním softwarem RSVIEW32.
2. Navrhněte a vypracujte příklady řízení a vizualizace pro podporu výuky s využitím výše uvedených systémů.
3. Příklady musí obsahovat základní typy úlohy logického řízení (čítače, časovače, zásobníky, přerušení apod.) a navazující spojitě řízení včetně PID instrukce s příslušnou vizualizací.
4. Kromě vlastního zadání a řešení úloh je potřeba připravit vhodné kostry programů pro zrychlení vlastního řešení studenty, které zároveň usnadní řešení složitějších úloh.

Seznam odborné literatury:

- [1] firemní literatura www.ab.com
- [2] Martinásková Marie, Šmejkal Ladislav: PLC a automatizace I a II, Ben Praha, 2005

Vedoucí: Ing. Jindřich Fuka

Platnost zadání: do konce zimního semestru 2012/2013


Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 15. 5. 2012

Obsah

Seznam obrázků	ix
Seznam tabulek	x
1 Úvod	1
2 Programovatelné automaty	2
2.1 Programovatelné automaty obecně	2
2.1.1 Historie	2
2.1.2 Vlastnosti	3
2.1.3 Možnosti programování	4
2.2 ControlLogix	4
2.2.1 Rodina Logix	5
2.2.2 Vlastnosti	6
2.2.2.1 Plánování úloh	8
2.2.3 Komunikace	9
2.2.4 Softwarové nástroje	10
2.2.4.1 RSLinx Classic	10
2.2.4.2 RSLogix 5000	10
2.2.4.3 RSVIEW32	11
3 ControlLogix programování	13
3.1 Žebříčkový (Příčkový) diagram (LAD)	13
3.1.1 Ukázka programu	14
3.2 Funkční bloky (FBD)	15
3.2.1 Základní prvky	15
3.2.2 Synchronizace	15
3.2.3 Pořadí vykonávání	15

3.2.4	Vytváření smyček	16
3.2.5	Ukázka programu	17
3.3	Strukturovaný text (ST)	18
3.3.1	Programové konstrukce	18
3.3.2	Ukázka programu	19
3.4	Sekvenční diagram (SFC)	20
3.4.1	Základní prvky	20
3.4.2	Ukázka programu	21
4	Výukové modely	22
4.1	Podavač	22
4.1.1	Popis modelu	23
4.1.2	Vnitřní zapojení	23
4.1.2.1	Napájení modelu	24
4.1.2.2	Princip měření polohy	24
4.1.2.3	Princip ovládání motoru	25
4.1.3	Připojení k PLC	26
4.2	Vodárna	27
4.2.1	Princip	28
4.2.2	Akční členy a senzory	28
4.2.2.1	Čerpadlo	28
4.2.2.2	Snímače výšky hladiny	28
4.2.2.3	Digitální ventil	29
4.2.2.4	Proporcionální ventil	29
4.2.3	Řídící elektronika	29
4.2.4	Připojení k PLC	29
5	Řízení a vizualizace	31
5.1	Podavače	32
5.1.1	Úloha č. 1 - seznámení	32
5.1.2	Úloha č. 2 - úlohy a průběhy (tasks and trends)	32
5.1.3	Úloha č. 3 - čítače	33
5.1.4	Úloha č. 4 - časovače	33
5.1.5	Úloha č. 5 - přerušení	33
5.1.6	Úloha č. 6 - zásobníky	34

5.1.7	Úloha č. 7 - základní vizualizace	34
5.2	Vodárny	35
5.2.1	Úloha č. 1 - seznámení s modelem vodáren	35
5.2.2	Úloha č. 2 - PID regulace	35
5.2.3	Úloha č. 3 - vodárna - vizualizace 1	35
5.2.4	Úloha č. 4 - vodárna - vizualizace 2	36
6	Podpora výuky	37
7	Závěr	39
	Literatura	40
A	Obsah přiloženého CD	I
B	Seznam zkratk	II
C	Použitý software	III
D	Podavač - celkové schéma zapojení	IV
E	Konfigurace automatů v laboratoři Allen-Bradley	VI

Seznam obrázků

2.1	Fotografie programovatelného automatu ControlLogix	4
2.2	Blokové schéma procesoru ControlLogix	6
2.3	ControlLogix cyklus	6
2.4	Klasický cyklus PLC	7
2.5	Ukázka plánování úloh (zelené vykonávány)	8
2.6	Ukázka zobrazení konfigurace automatu v prostředí RSLinx Classic . . .	10
2.7	Vývojové prostředí RSLogix 5000	11
2.8	Ukázka vizualizace vytvořené v prostředí RSVIEW32	12
3.1	Ukázka programu vytvořeného v žebříčkovém diagramu	14
3.2	Ukázka vytváření zpětnovazební smyčky	16
3.3	Ilustrativní ukázka programu vytvořeného pomocí funkčních bloků	17
3.4	Lišta s nabídkou jednotlivých prvků (RSLogix 5000)	21
3.5	Ukázka programu vytvořeného pomocí sekvečního diagramu	21
4.1	Fotografie modelu podavače	22
4.2	Napájení modelu	24
4.3	Měření polohy	25
4.4	Fotografie clonícího kotoučku - převzato z [2]	25
4.5	Blokové schéma ovládání motoru	26
4.6	Fotografie modelu vodárny	27
4.7	Princip funkce modelu - obrázek převzatý z [1]	28
5.1	Fotografie automatu s adresou 147.32.87.137	32
6.1	Ukázka části návodu popisujícího vytvoření projektu v prostředí RSLogix 5000	38
D.1	Celkové schéma ovládání motoru	V

Seznam tabulek

3.1	Pravdivostní tabulka start/stop programu	17
3.2	Aritmetické a logické operátory	18
3.3	Řídící konstrukce	19
3.4	Komentáře	19
4.1	Seznam tlačítek modelu podavače	23
4.2	Připojení vodárny V3 k automatu 147.32.87.137	30
E.1	Konfigurace automatu 147.32.87.132 (šasi 1756-A10/A)	VI
E.2	Konfigurace automatu 147.32.87.133 (šasi 1756-A10/A)	VII
E.3	Konfigurace automatu 147.32.87.134 (šasi 1756-A10/A)	VII
E.4	Konfigurace automatu 147.32.87.135 (šasi 1756-A13/A)	VIII
E.5	Konfigurace automatu 147.32.87.136 (šasi 1756-A10/A)	VIII
E.6	Konfigurace automatu 147.32.87.137 (šasi 1756-A13/A)	IX

Kapitola 1

Úvod

Hlavním účelem mé bakalářské práce je podpora výuky v laboratoři Allen-Bradley a podpora výuky řízení programovatelnými automaty obecně. Již v dřívějších letech vzniklo na katedře Řídicí techniky několik prací, které se věnovaly řízení programovatelnými automaty. Má práce z některých částečně vychází a dává si za úkol vypracovat další zajímavé příklady řízení a vizualizací včetně studijních návodů a materiálů, které umožní studentům snáze proniknout do dané problematiky.

Práce se dělí do několika částí. První část (Kap. 2) je věnována obecnějšímu popisu programovatelných automatů s důrazem na moderní programovatelné automaty ControlLogix od firmy Rockwell Automation, které jsou využívány v laboratoři K23 na katedře Řídicí techniky. Následující část (Kap. 3) popisuje základní vlastnosti a možnosti jednotlivých programovacích technik, pomocí kterých lze automaty ControlLogix naprogramovat. V další části (Kap. 4) je pojednáno o dvou výukových modelech nacházejících se v laboratoři K23. Jde o model podavače, který umožňuje názorné seznámení s programovatelnými automaty a o model vodárny, který je určen pro pokročilejší řízení. Zde jsou uvedeny základní popisy těchto systémů, které jsou nezbytné pro další práci. Následující část (Kap. 5) je již věnována samotnému řízení a vizualizacím, přičemž pro oba modely je vypracováno několik vzorových úloh, a to od jednodušších až po komplexnější. Tyto úlohy mají za cíl naznačit studentům možnosti programovatelných automatů ControlLogix. Poslední část (Kap. 6) je věnována tvorbě výukových materiálů. Tyto materiály popisují především práci se softwarovými nástroji, které budou studenti při práci potřebovat.

Hlavním výstupem této práce je několik příkladů řízení a vizualizací obou dříve zmíněných modelů a základní návody pro práci s programovatelnými automaty ControlLogix a příslušnými softwarovými nástroji. Veškeré příklady a výukové materiály budou dostupné na počítačích přímo v laboratoři, aby byly snadno přístupné pro studenty.

Kapitola 2

Programovatelné automaty

V dnešním moderním světě, ve kterém hraje automatizovaná výroba a technika obecně stále větší roli, mají programovatelné automaty široké využití. Jsou to vůbec nejčastěji používané počítače pro řízení. Tato kapitola se dělí na dvě podkapitoly. První z nich se zabývá programovatelnými automaty obecně, druhá se zaměřuje na programovatelné automaty ControlLogix.

2.1 Programovatelné automaty obecně

2.1.1 Historie

Hlavní iniciativou pro vznik programovatelných automatů byla poptávka automobilového průmyslu (konec 60. let v USA), pro který začalo být postupem času neúnosné neustálé předělávání (při změně modelu) napevno zadrátované logiky, které bylo spojeno s velkým časovým a pracovním vytížením. Dle [23] bylo dokonce při změně ve výrobě obvykle výhodnější vybudovat celou logiku od začátku.

Z tohoto důvodu vydalo GM Hydramatic (divize pro výrobu automatické převodovky v General Motors) požadavek na zařízení, které by umožňovalo snížení nákladů a zkrácení časové prodlevy při změnách ve výrobě. Vítězný návrh byl od firmy Bedford Associate, kde tým konstruktérů pod vedením Dicka Morleyho vytvořil první programovatelný automat, který pojmenovali Modicon 084 (z anglického Modular Digital Controller a 84, jelikož se jednalo o 84. projekt). Dick Morley je proto často považován za jakéhosi otce PLC.

2.1.2 Vlastnosti

Programovatelný automat je počítač určený pro řízení. Původně nesl označení PC (Programmable Controller). Po rozšíření osobních počítačů, logicky též označovaných zkratkou PC (Personal Computer), vzniklo pro programovatelné automaty označení PLC (Programmable Logic Controller). S dalším vývojem se již programovatelné automaty neomezovaly pouze na logické řízení, a proto pro ně vzniklo nové označení PAC (Programmable Automation Controller). Zkratka PLC je tudíž pro označení moderních systému nepřesná, její používání je však natolik zažitě, že ve většině případů se stále s tímto termínem setkáváme.

Programovatelné automaty mají obrovské využití v průmyslu (řízení výrobních linek, řízení strojů, řízení technologických procesů apod.). Na rozdíl od běžného počítače třídy PC jsou přizpůsobeny pro práci v průmyslovém prostředí, z čehož plynou některé jejich vlastnosti:

- odolnost vůči elektromagnetickému rušení,
- konstrukce pro práci v širokém teplotním rozsahu,
- odolnost vůči mechanickým vibracím atd..

Programovatelné automaty můžeme rozdělit do dvou základních skupin:

1. Kompaktní

- jeden samostatný modul, který obsahuje procesor, vstupy a výstupy, základní podporu komunikace a v některých případech zdroj,
- výhodou je nižší cena – orientačně od 2000 Kč do 10 000 Kč.

2. Modulární

- jsou systémy, ve kterých jsou jednotlivé komponenty rozděleny do modulů,
- celý systém PLC se potom skládá z jednotlivých modulů,
- obsahují obvykle zdroj, modul procesoru, moduly vstupů a výstupů,
- výhodou tohoto systému je jeho univerzálnost a možnost rozšiřování.

Moderní PLC disponují rozsáhlou programovou a datovou pamětí, umožňují vykonávat matematické operace, zpracovávat analogové signály a mnoho dalších funkcí. Jelikož jsou programovatelné automaty nasazeny v situacích, kde by jejich selhání mohlo znamenat obrovské škody (nejen finanční), jsou na jejich spolehlivost kladeny vysoké nároky.

2.1.3 Možnosti programování

První programovatelné automaty byly určeny jako náhrada klasických reléových logických systémů. Pro tyto PLC byl proto navržen způsob programování, který měl blízko ke klasické reléové logice (kterou technici běžně používali). Tento programovací jazyk dostal jméno žebříčkový diagram (ladder diagram, LAD). Programování automatů pomocí žebříčkových diagramů mělo obrovskou výhodu v tom, že pracovníci, kteří používali klasickou reléovou logiku, si na nový systém velmi rychle zvykli a tím se minimalizovaly náklady na potřebná školení.

Moderní programovatelné automaty přidávají další možnosti programovacích technik a umožňují programování pomocí žebříčkových diagramů (také nazýván jako liniové nebo kontaktní schéma), programovacích jazyků podobných C (případně Basicu), stavové logiky a funkčních bloků.

2.2 ControlLogix

Programovatelné automaty ControlLogix jsou moderní výkonné systémy patřící do rodiny řídicích systémů Logix od firmy Rockwell Automation. V Evropě byly představeny na veletrhu v Hamburku v roce 1999. V následující kapitole je popsáno několik důležitých vlastností těchto automatů včetně stručného představení zástupců rodiny Logix.



Obrázek 2.1: Fotografie programovatelného automatu ControlLogix

2.2.1 Rodina Logix

- **1756 ControlLogix**

Jedná se o nejvýkonější řídicí systém založený na procesoru Logix. Procesor umožňuje vykonávat až 32 paralelních úloh. Uživatelská paměť se pohybuje v rozmezí od 750 KB do 32 MB. Jako vstupní a výstupní moduly používá 1756 ControlLogix I/O. Pro komunikaci je zabudován sériový port (u novějších je nahrazen USB portem).

- **1759 CompactLogix**

Je řídicí systém určený pro lokální řízení menších aplikací, které nekladou velké požadavky na komunikaci. V závislosti na konkrétním typu umožňuje procesor vykonávat 4, 6 nebo 8 úloh. Velikost uživatelské paměti nabývá hodnot od 512 KB do 1,5 MB (opět v závislosti na konkrétním typu). Systém využívá moduly lokálních vstupů a výstupů Compact I/O. Pro účely komunikace je každý typ osazen dvěma komunikačními porty. Jeden z nich je vždy sériový a druhý dle konkrétního typu sériový, ControlNet nebo Ethernet/IP.

- **1789 SoftLogix**

Jedná se o softwarovou realizaci programovatelného automatu pracujícího pod operačním systémem windows. Umožňuje vykonávat až 32 úloh. Uživatelská paměť je 2 MB (1789-L10) nebo 64 MB (1789-L30 a 1789-L61).

- **1794 FlexLogix**

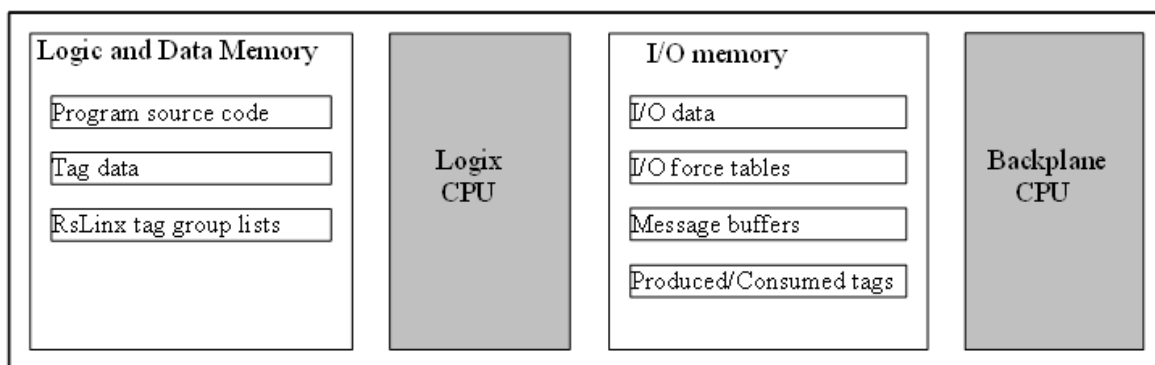
Je řídicí systém určený pro řízení malých až středně velkých aplikací využívajících moduly vzdálených vstupů a výstupů Flex I/O (až 8 přímo připojených modulů). Procesor umožňuje vykonávání až 8 úloh. Pro komunikaci je určen zabudovaný sériový port RS232. Tento typ řídicího systému se již nevyrábí.

- **DriveLogix**

Řídicí systémy DriveLogix vznikly začleněním procesoru Logix do frekvenčního měniče PowerFlex 700 (řízení rychlosti, případně točivého momentu 3 fázových indukčních motorů). Procesor umožňuje vykonávání až 8 nezávislých úloh. Uživatelská paměť nabývá hodnot od 256 KB do 1,5 MB (v závislosti na typu). Pro komunikaci je každý typ vybaven sériovým portem a jedním slotem pro komunikační kartu. Jako vstupně výstupní moduly využívá Flex I/O, nebo Compact I/O (opět dle typu).

2.2.2 Vlastnosti

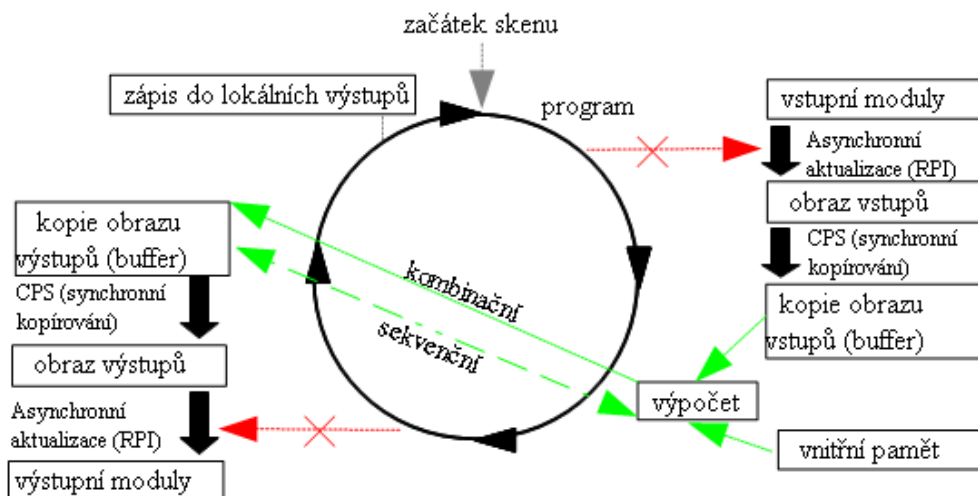
Blokové schéma (viz. Obr. 2.2) ukazuje strukturu procesorového modulu. Z obrázku je patrné, že paměť je rozdělena do dvou oddělených sekcí a modul obsahuje dva procesory. Ten „hlavní“ (Logix CPU) vykonává program, „vedlejší“ (Backplane CPU) obstarává distribuci dat v šasi.



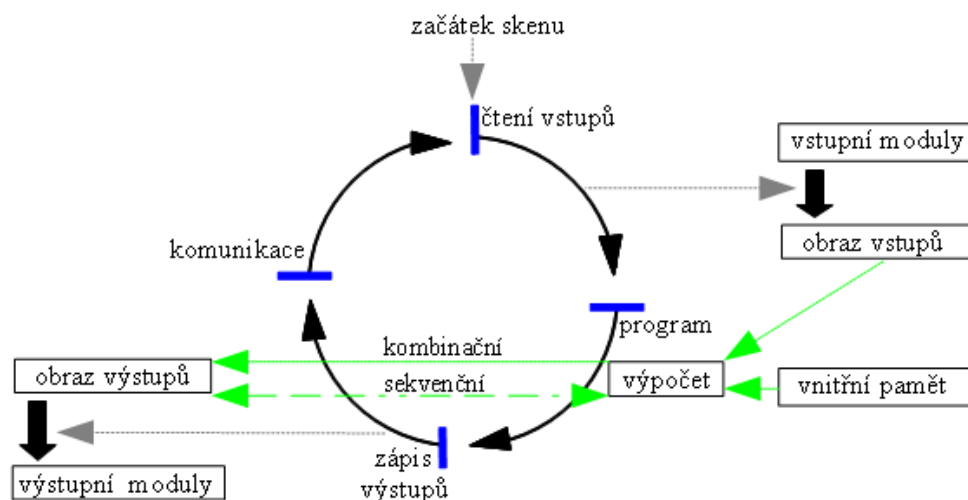
Obrázek 2.2: Blokové schéma procesoru ControlLogix

Pozn.: U novějších procesorů 1756-L7x obsahuje hlavní procesor dvě jádra.

Jelikož vedlejší procesor pracuje nezávisle na hlavním, posílá a přijímá vstupně výstupní informace asynchronně vůči vykonávání programu, z čehož plynou některá specifika při práci s těmito automaty. Obsluhujeme-li proto například některý ze vstupů z více míst programu, je nutná synchronizace. Nebude-li synchronizace provedena, je možné, že se z různých míst programu během jednoho skenu přečte rozdílná hodnota vstupu.



Obrázek 2.3: ControlLogix cyklus



Obrázek 2.4: Klasický cyklus PLC

Obrázky 2.3 a 2.4 ukazují cyklus programovatelného automatu ControlLogix a cyklus „klasického“ PLC.

Jedná se o modulární systémy, ve kterých jsou jednotlivé moduly dle požadavků dané aplikace umísťovány do šasi (součástí šasi je zdroj), které mívá 4, 7, 10, 13 nebo 17 pozic. Základní instrukční soubor je v podstatě shodný s instrukčním souborem staršího typu PLC-5 (je pouze doplněn o několik specifických instrukcí). ControlLogix přidává možnosti řízení pohybů a procesů (motion and process control).

Procesor je schopen pracovat až se 128 000 logickými nebo 4 000 analogovými I/O kanály. Dále může zpracovávat až 32 nezávislých úloh, které mohou být následujícího typu:

1. kontinuální úloha (continuous task)

- je automaticky vytvořena při tvorbě nového projektu (ale není nutné ji v projektu mít) a má nejnižší prioritu (16),
- není-li CPU alokovan jinou operací (například periodickou úlohou), je využíván právě k vykonávání kontinuální úlohy,
- důležité je, že přechod z bezpečných stavů výstupů je proveden až po vykonání prvního skenu kontinuální úlohy (je-li v projektu definována).

2. periodická úloha (periodic task)

- vykonává určitou funkci v předem specifikovaném tempu,

- nastane-li její čas, přeruší tato periodická úloha jakoukoliv úlohu s nižší prioritou, vykoná jeden svůj sken a vrátí řízení zpět úloze, kterou přerušila,
- perioda (T) může být v rozsahu 1 ms až 2000 s (defaultní hodnota 10 ms),
- přiřazuje se jí priorita v rozsahu 1 = nejvyšší priorita až 15 = nejnižší priorita.

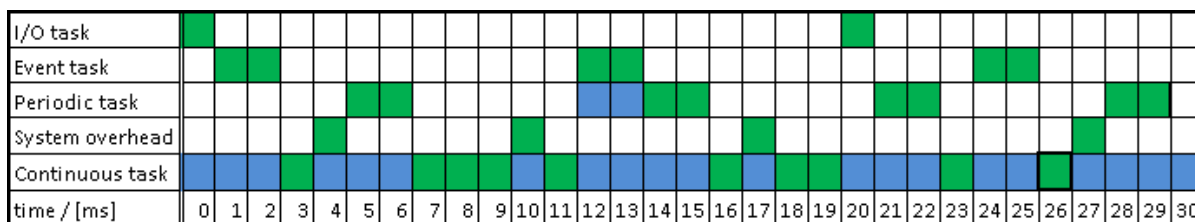
3. úloha řízená událostí (event task)

- provádí se pouze tehdy, objeví-li se nějaká specifická událost (tzv. trigger), kterou může být například změna dat od vstupního modulu,
- je to jakási analogie přerušení v mikroprocesorové technice.

2.2.2.1 Plánování úloh

Jelikož automaty ControlLogix umožňují vykonávání více úloh (tasks), je nutné vykonávání těchto úloh plánovat. Pro názornost je plánování ukázáno na konkrétním příkladu, ve kterém jsou nadefinovány následující úlohy s danou dobou vykonávání:

- I/O task - periodická úloha zpracovávající vstupně/výstupní data. Její priorita je pevně daná a je rovna šesti. Perioda je rovna minimálnímu RPI - zde 20 ms. Tato úloha není zobrazena v seznamu úloh v projektovém průzkumníku, ale je třeba ji brát v potaz. Doba vykonávání 1 ms.
- Event task - priorita 9, doba vykonávání 2 ms.
- Periodic task - priorita 10, perioda 5 ms, doba vykonávání 2 ms.
- System overhead - úloha, která obstarává neplánovanou komunikaci (například MSG), doba vykonávání 1 ms.
- Continuous task - má nejnižší možnou (a pevně danou) prioritu, doba vykonávání 9 ms.



Obrázek 2.5: Ukázka plánování úloh (zelené vykonávány)

Obr. 2.5 ukazuje průběh vykonávání jednotlivých úloh. Zeleně vyznačené jsou úlohy, které se v daném okamžiku vykonávají. Modrá barva značí, že daná úloha čeká, až jí bude přidělen procesorový čas.

Je patrné, že úlohy s vyšší prioritou nedovolují běh úloh s nižší. Proto například v čase 12 ms a 13 ms musí periodická úloha (Periodic task) čekat, až se vykoná úloha řízená událostí (Event task). V čase 26 ms dojde k restartování kontinuální úlohy (naznačeno tmavým rámečkem). Pokud se mají současně vykonávat úlohy se stejnou prioritou, prokládají se automaticky po 1 ms.

Užitečným nástrojem pro monitorování úloh je Logix5000 Task Monitor. Tento softwarový nástroj přehledně zobrazuje seznam všech vykonávaných úloh spolu s důležitými informacemi o průběhu vykonávání.

Bližší informace o tomto nástroji lze nalézt v [20].

2.2.3 Komunikace

Komunikace mezi lokálními moduly probíhá po vnitřní sběrnici **ControlBus** (otevřená). Ta používá nový model - producent a konzument (spotřebitel), který je efektivnější než klasické Master (obvykle Procesor) - Slave (ostatní moduly) dotazování. Výhodou tohoto řešení je podstatné zrychlení komunikace na sběrnici. Data vystavená na sběrnici jedním producentem může přijímat několik konzumentů zároveň. Moduly mají ke sběrnici naprosto rovnocenný přístup (nerozlišuje se, zda jde o procesor, komunikační modul, nebo jiné).

Pro komunikaci je možné využít klasický sériový port RS232 (u novějších typů nahrazen USB), Ethernet/IP, ControlNet nebo DeviceNet.

ControlNet je deterministická opakovatelná průmyslová síť, která využívá architekturu producent/spotřebitel. Přenosová rychlost je 5Mbit/s, přičemž jako přenosové médium lze použít koaxiální kabel (RG-12, 75Ω), jehož výhodou je snadná dostupnost. Umožňuje vytvářet síť s topologií sběrnice, hvězdy nebo stromu. Pro připojení k PC je použita PCI karta 1784-PCIC, komunikační modul nese označení 1756-CNB (bližší informace v [17]). Pro návrh sítě ControlNet slouží softwarový nástroj RSNetWorx for ControlNet (bližší informace v [18]).

Jednou z hlavních předností sítě ControlNet je to, že umožňuje vysílat časově kritická vstupně/výstupní data deterministicky a opakovatelně. Ostatním datům je přiřazena nižší priorita a nemohou tak narušit přenos časově kritických dat.

Ethernet/IP (IP = industrial protocol = průmyslový protokol) vychází z celosvětově

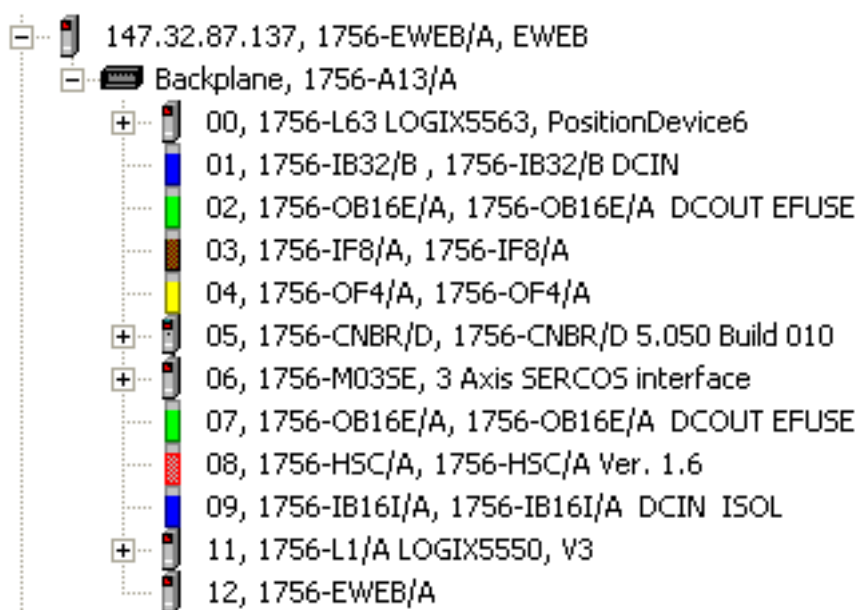
známé sítě Ethernet s protokolem TCP/IP. Výhodou tohoto řešení je možnost připojení k počítači běžnou síťovou kartou.

Pozn.: Průvodce výběrem nejvhodnější sítě a jejich porovnání naleznete v [19].

2.2.4 Softwarové nástroje

2.2.4.1 RSLinx Classic

Je softwarový nástroj firmy Rockwell Automation sloužící jako mezičlánek mezi programovatelným automatem a aplikačním softwarem (RSLogix 5000, RSView32 apod.). Tento program má v sobě integrovány ovladače pro průmyslové sítě, ethernet a sériový port. Program obsahuje rozhraní pro přístup k datům přes protokoly DDE nebo OPC.

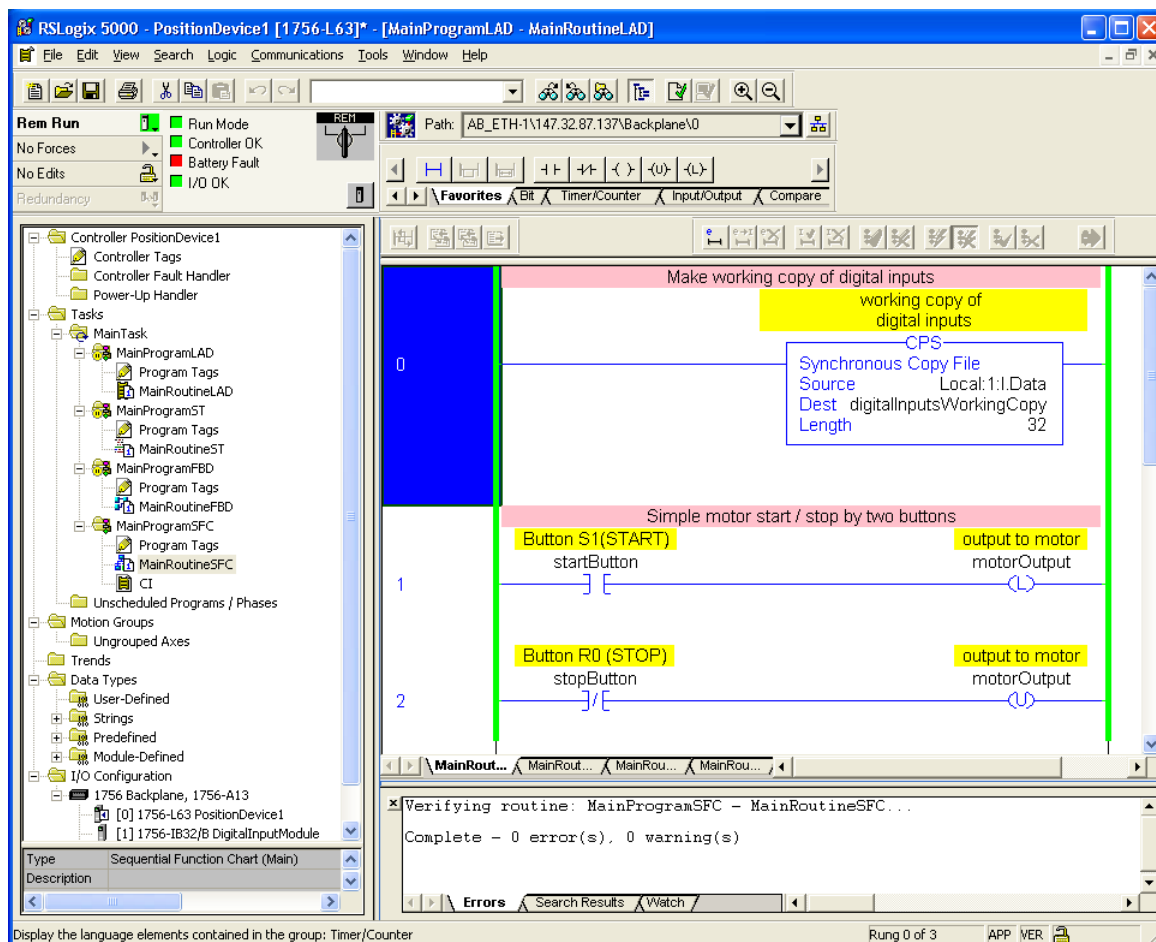


Obrázek 2.6: Ukázka zobrazení konfigurace automatu v prostředí RSLinx Classic

2.2.4.2 RSLogix 5000

Je softwarový nástroj sloužící k vytváření projektů (včetně konfigurace modulů) a následnému programování. Toto vývojové prostředí umožňuje uživateli programování ve čtyřech programovacích jazycích (viz Kap. 3). V rámci jednoho projektu je možno tyto jazyky kombinovat.

Mezi některé užitečné funkce patří například možnost verifikace (ať už celého kontroléru či konkrétního programu), zobrazování průběhů, přehledné vybírání modulů z databáze, či automatické doplňování nápovědy při zadávání instrukcí.



Obrázek 2.7: Vývojové prostředí RSLogix 5000

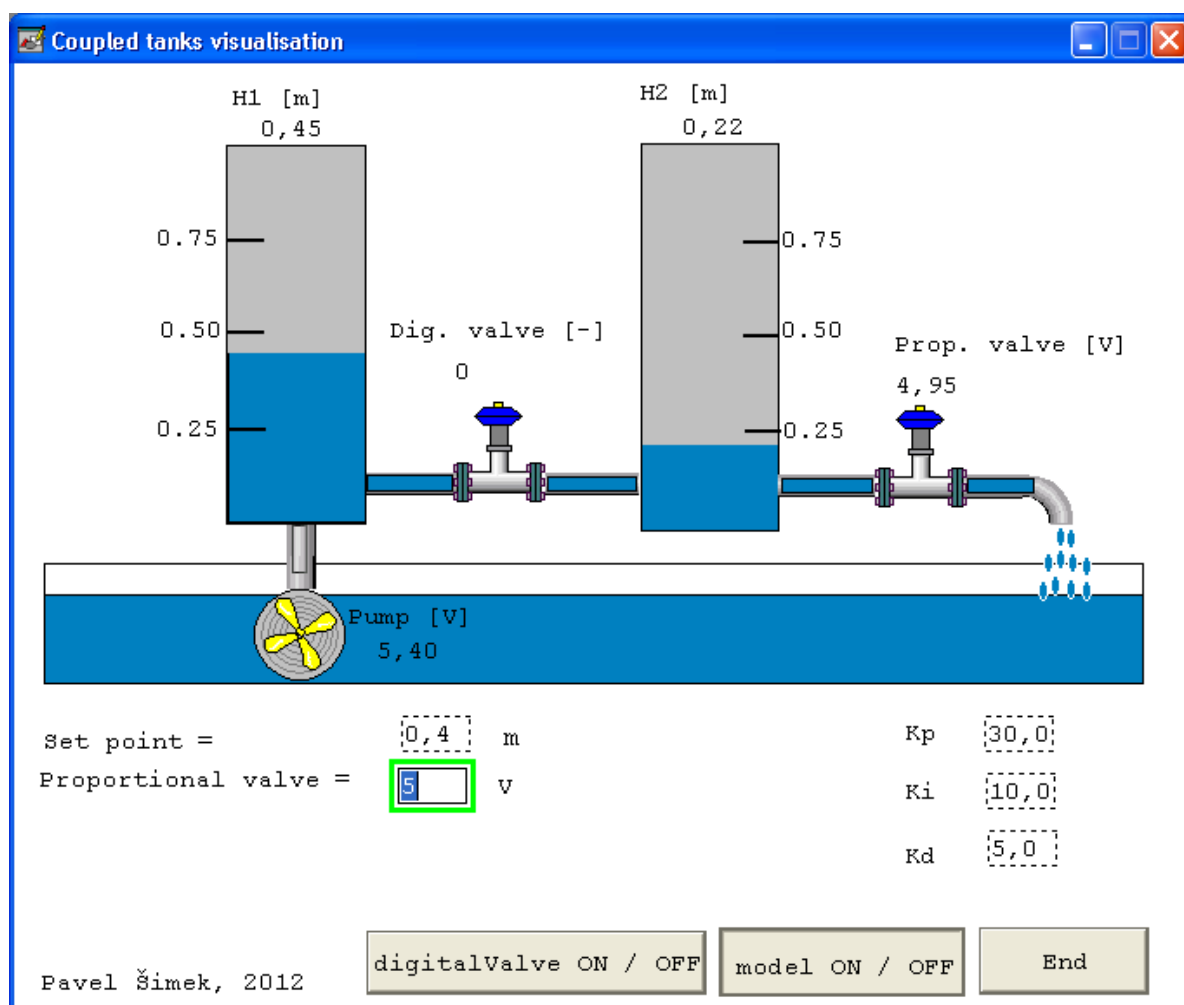
2.2.4.3 RSVIEW32

RSView32 je softwarový nástroj určený pro 32 bitovou platformu MS Windows umožňující tvorbu vizualizací. Vizualizace slouží jako meziklánek mezi technologickým procesem a operátorem. Jejím hlavním úkolem je nejen informovat operátora o stavu procesu, ale také mu umožnit proces ovlivnit, čímž je umožněno zefektivnění výroby. Tento nástroj poskytuje široké možnosti pro efektivní tvorbu vizualizace. Pro komunikaci mezi vizualizací a programovatelným automatem lze použít přímé spojení Direct Driver Communication, druhou možností je použití DDE (Dynamic Data Exchange) či OPC (Ole for Process Control).

K tvorbě uživatelského rozhraní slouží jak základní prvky, jako například tlačítka (Buttons), popisky (Labels), vstupní textová pole (Input text fields), tak i složitější prvky, které jsou součástí bohatě vybavených knihoven - například motory, čerpadla, zásobníky a mnohé další.

Mezi další zajímavé možnosti tohoto prostředí stojí za zmínku například:

- možnost vytvářet logovací soubory,
- možnost zobrazovat průběhy.



Obrázek 2.8: Ukázka vizualizace vytvořené v prostředí RSVIEW32

Kapitola 3

ControlLogix programování

Programovatelné automaty ControlLogix umožňují programování pomocí čtyř programovacích jazyků. Tato kapitola je věnovaná základnímu popisu těchto jazyků.

3.1 Žebříčkový (Příčkový) diagram (LAD)

Je zřejmě nejoblíbenějším a nejrozšířenějším jazykem pro programování programovatelných automatů. Jeho obrovskou předností je podobnost s klasickým reléovým schématem, které je pro techniky známé. Jeho další výhodou spočívá v přehlednosti a snadné čitelnosti programů.

Program vytvořený v tomto programovacím jazyku je organizován do jakéhosi žebříku (proto název Ladder diagram). Do jednotlivých příček (rungs) se vkládají instrukce, které jsou následujících typů:

- **Vstupní instrukce** (angl. někdy nazývány contacts) - jsou instrukce, které kontrolují stav určitého bitu v paměti. Typickým příkladem je instrukce XIC (Examine If Closed), která zkoumá, zda je daný bit nastaven.
- **Výstupní instrukce** (angl. někdy nazývány coils) - jsou instrukce, které nastavují stav určitého bitu v paměti. Typickým příkladem je instrukce OTE (Output Energize), která nastaví nebo vynuluje daný bit v závislosti na vstupní podmínce (rung condition).

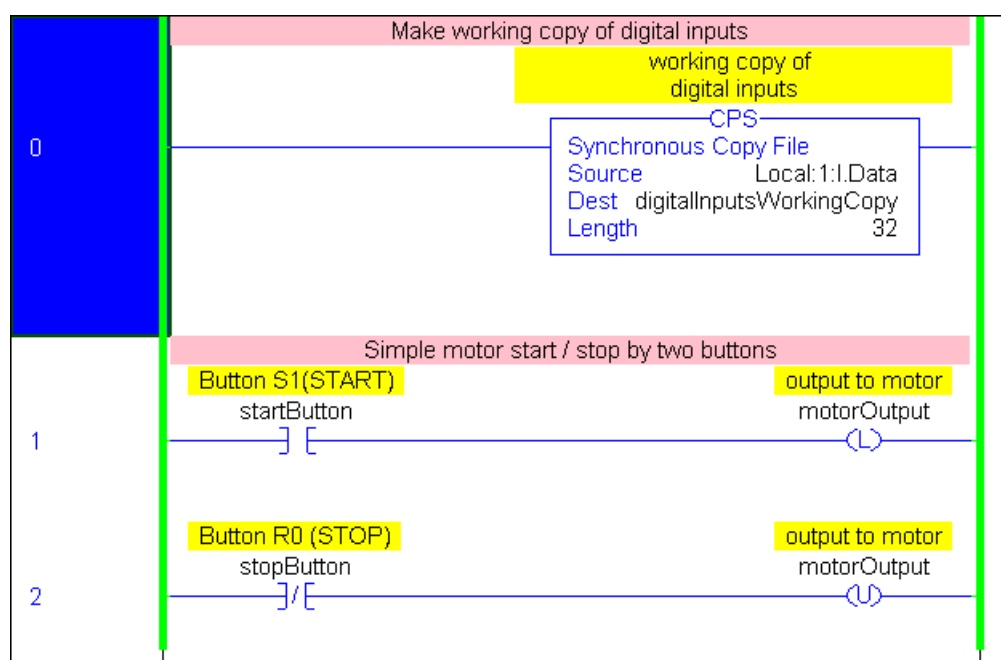
Výstupní instrukce se dělí na nepřídržné (např. OTE) a přídržné (např. OTL, OTU). Nepřídržná instrukce nastavuje daný bit do požadované hodnoty na dobu jednoho

skenu. Po použití přídržné instrukce naopak daný bit zůstává v nastavené hodnotě, dokud není použitím opačné přídržné instrukce jeho stav změněn. Konkrétní použití přídržných instrukcí viz příklad níže (Obr. 3.1).

- **Ostatní instrukce** - jsou například instrukce čítačů, časovačů, přesunů, matematické funkce a podobně. Tyto instrukce byly do instrukčních sad implementovány proto, aby vývojářům usnadnily práci (v původních programovatelných automatech byly pouze základní kontakty a cívky).

3.1.1 Ukázka programu

Na následujícím obrázku je program vytvořený v žebříčkovém diagramu. Tento program obstarává spuštění a zastavení motoru pomocí dvou tlačítek.



Obrázek 3.1: Ukázka programu vytvořeného v žebříčkovém diagramu

V prostředí RSLogix 5000 je možno instrukce vkládat dvěma způsoby. Pomocí jejich textových zkratk - textový zápis (XIO, OTE, CTU, TON apod.), nebo je možné využít lištu s nabídkou instrukcí a tyto instrukce přetahovat (Drag and Drop) na potřebné místo v žebříku.

3.2 Funkční bloky (FBD)

Jedná se o vizuální programovací jazyk. Program se vytváří pomocí vhodného propojování funkčních bloků. Jednotlivé části programu jsou rozdělovány do listů (angl. sheets). Hodí se především pro procesní řízení.

3.2.1 Základní prvky

Vstupní reference IREF (Input REference) - poskytuje hodnotu určitého vstupu nebo tagu. Chcete-li například číst hodnotu vstupního tlačítka, použijte IREF.

Výstupní reference OREF (Output REference) - zapisuje hodnotu určitého výstupu nebo tagu. Chcete-li ovládat stav určitého výstupu, použijte OREF.

Funkční blok (Function block) - vykonává operace se vstupy a podle výsledku nastavuje výstupy. Jako konkrétní příklad můžeme uvést blok AND, který provádí logický součin vstupů a podle výsledku této operace nastaví nebo vynuluje svůj výstup. Většina funkčních bloků umožňuje konfigurovat parametry (například měnit počty vstupů).

Propojovací konektory OCON (Output wire CONnector) a **ICON** (Input wire CONnector) - slouží k distribuci dat mezi funkčními bloky. Je-li například v listu3 potřeba využít výsledek operace, který je poskytován výstupem funkčního bloku nacházejícího se v listu1, použije se OCON a ICON.

3.2.2 Synchronizace

Výhodou programování pomocí funkčních bloků je to, že je automaticky zajištěna synchronizace. Synchronizace probíhá tak, že na začátku vykonávání rutiny jsou načteny hodnoty ze vstupních referencí IREF a tyto hodnoty jsou použity během výkonu daného listu.

3.2.3 Pořadí vykonávání

O pořadí vykonávání je rozhodováno automaticky. Nejsou-li funkční bloky propojeny, pak na pořadí jejich vykonávání nezáleží.

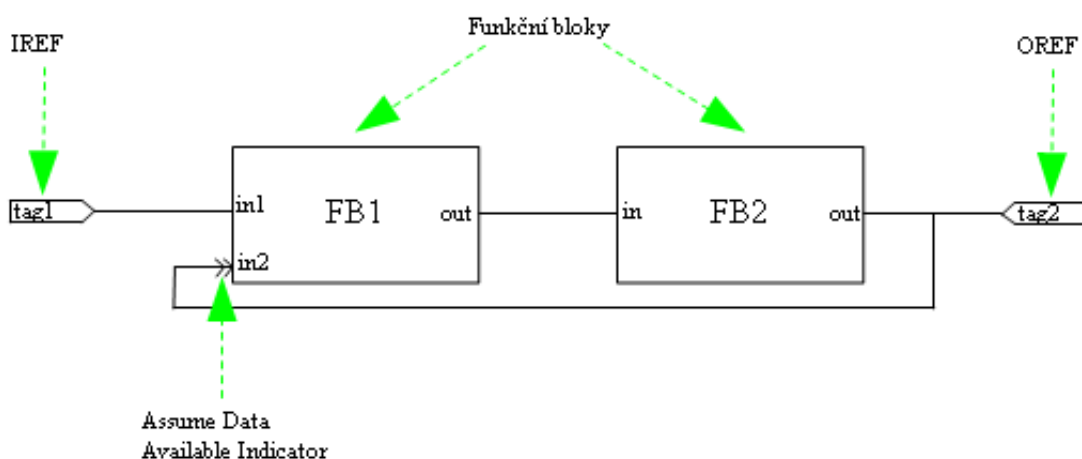
Jsou-li funkční bloky propojeny sekvenčně, pak se vykonávají v tom pořadí, v jakém jsou propojeny (postupuje se od vstupu směrem k výstupu). Je to dáno tím, že každý

blok pro své vykonání potřebuje mít na vstupu platná data. A jsou-li tato data zároveň výstupem předchozího bloku, pak se logicky musí předchozí blok vykonat dříve.

3.2.4 Vytváření smyček

Při vytváření zpětnovazebních smyček (a propojení více funkčních bloků) již RSLogix 5000 nemůže určit pořadí vykonávání těchto bloků. V tomto případě musí uživatel rozhodnout o průběhu vykonávání pomocí přiřazení značky indikující předpokládanou dostupnost dat (Assume Data Available Indicator).

Ukázka vytváření zpětnovazební smyčky je uvedena na Obr. 3.2. Zde jsem značku indikující předpokládanou dostupnost dat umístil na vstup in2 funkčního bloku FB1. Tím jsem určil, že při vykonávání FB1 bude na jeho vstupu použit výstup out FB2, který byl vyprodukovaný při předešlém vykonávání rutiny.



Obrázek 3.2: Ukázka vytváření zpětnovazební smyčky

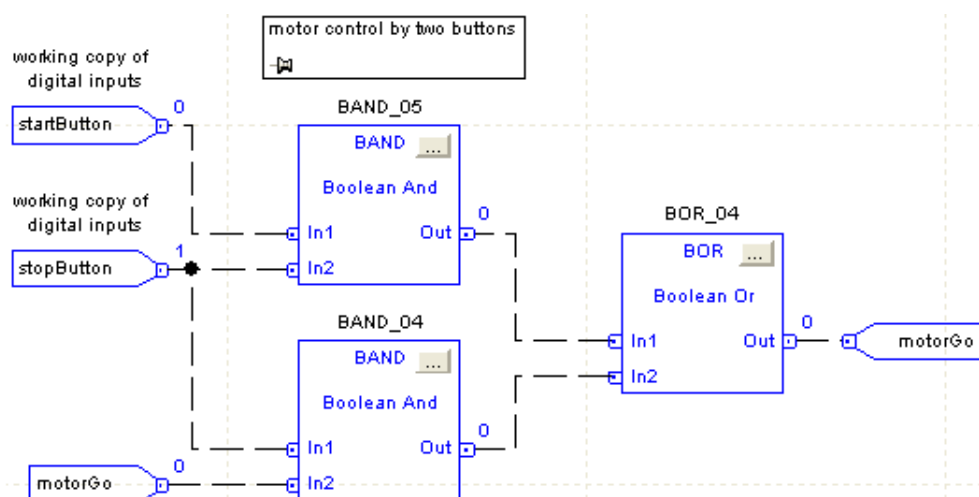
3.2.5 Ukázka programu

Následuje ukázka programu, který obstarává rozběh a zastavení motoru v závislosti na stisku tlačítek. Funkčnost je implementována pomocí logické funkce, která je reprezentována následující pravdivostní tabulkou.

tlačítko START	tlačítko STOP	stav motoru	výstup motor
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Tabulka 3.1: Pravdivostní tabulka start/stop programu

Nyní provedeme minimalizaci Karnaughovy mapy a výslednou funkci $y = AB + BC$ zadrátujeme pomocí funkčních bloků:



Obrázek 3.3: Ilustrativní ukázka programu vytvořeného pomocí funkčních bloků

Pozn.: Čárkované propojovací vodiče značí, že se po nich přenáší datový typ BOOL. U jiných datových typů (SINT, INT, DINT, REAL) je vodič vykreslen plnou čarou.

3.3 Strukturovaný text (ST)

Je textový programovací jazyk, jehož syntaxe je podobná Pascalu, Basicu a jazyku C. Tento programovací jazyk obsahuje velké množství řídicích konstrukcí (podmínky, cykly apod.), umožňuje vykonávání matematických operací (goniometrické funkce, absolutní hodnota, atd.) a mnohé další. Program v tomto jazyku se skládá z výrazů, přiřazení, instrukcí, řídicích konstrukcí a komentářů. Při zápisu je každá instrukce zakončena středníkem, jazyk není citlivý na malá/velká písmena (Case Insensitive). Jeho využití je například při řízení pohybů (složité vztahy, podmínky apod.).

3.3.1 Programové konstrukce

Následuje výčet základních programových konstrukcí, se kterými se student při tvorbě zajisté setká:

Přiřazení se používá tehdy, chceme-li určitému tagu přiřadit nějakou hodnotu. Existuje přiřazení dvojího typu:

- přídržné (alternativa OTL, OTU) tag := výraz;
- nepřídržné (alternativa OTE) tag [:=] výraz;

Aritmetické a logické operátory

zápis	popis	zápis	popis
+	aritmetický součet	<	menší než
-	aritmetický rozdíl	>	větší než
*	aritmetický součin	<=	menší nebo roven
/	aritmetický podíl	>=	větší nebo roven
MOD	zbytek po celočíselném dělení	AND	logický součin
**	x na y	OR	logický součet
=	rovnost	XOR	exkluzivní disjunkce
<>	nerovnost	NOT	logická negace

Tabulka 3.2: Aritmetické a logické operátory

Řídicí konstrukce jsou části programu, které rozhodují o tom, jak se bude program vykonávat - větví jeho běh, vytvářejí cykly, nebo jinak ovlivňují běh programu.

zápis	popis
IF THEN	podmíněný příkaz
CASE OF	větvení programu
FOR DO	cyklus
WHILE DO	cyklus
REPEAT UNTIL	cyklus

Tabulka 3.3: Řídící konstrukce

Komentáře - existuje několik typů komentářů, které jsou shrnuty v následující tabulce:

zápis	popis
//	jednořádkový komentář
(* *)	víceřádkový komentář
/* */	víceřádkový komentář

Tabulka 3.4: Komentáře

3.3.2 Ukázka programu

Následuje výpis programu, který obstarává rozběh a zastavení motoru v závislosti na stisku tlačítek.

```

/* This routine is responsible for motor control. To run motor push button Start (S1),
to stop motor push button Stop (R0) */

//make working copy of digital inputs
CPS(Local:1:I.Data, digitalInputsWorkingCopy, 32);

//if startButton(S1) is pushed then run the motor
IF startButton THEN
    motorGo := 1;
END_IF;

//if stopButton(R0) is pushed then stop the motor
IF NOT stopButton THEN
    motorGo := 0;
END_IF;

```

3.4 Sekvenční diagram (SFC)

Je vizuální programovací jazyk podobný vývojovému diagramu. Program vytvořený v tomto jazyku se skládá z kroků (steps) a přechodů (transitions). V pozadí každého kroku je struktura typu SFC_STEP a každému kroku je přiřazena nějaká akce. Průchod programem probíhá postupným vykonáváním jednotlivých kroků, přičemž přechod na další krok proběhne, je-li splněna podmínka přechodu.

Výhodou programování v tomto jazyku je to, že se vždy vykonává pouze jeden krok. To umožňuje vývojářům rozdělit program na více menších částí (alternativa k segmentovému programování). Program se tak stane efektivnějším a přehlednějším, než při použití jednoho dlouhého žebříku, ve kterém by byla v dané chvíli aktivní jen malá část příček. Další výhodou je možnost automatického „uklizení“ (Automatic Reset) během posledního skenu (Last Scan). Při využití této možnosti jsou vynulovány všechny výstupy, resetovány časovače atd.. Jinou možností je využít k „uklizení“ vlastní logiku (s využitím bitu SFC_STEP.LS).

3.4.1 Základní prvky

Krok (Step) je základním stavebním prvkem při programování v tomto jazyku. Každému kroku je přiřazena příslušná akce (ta může být dvojího typu - viz níže), která se začne vykonávat v okamžiku, kdy program dospěje k danému kroku.

Přechod (Transition) slouží k přechodu mezi kroky. Každému přechodu je přiřazena podmínka. Je-li tato podmínka vyhodnocena jako pravdivá, dojde k přechodu na další krok.

Akce (Action) slouží k vykonání nějaké činnosti (samotné kroky a přechody nevykonávají žádný užitečný kód). V rámci této akce je možné přímo vykonávat kód napsaný v ST, nebo je možné volat podprogramy instrukcí JSR.

Booleovská akce (Boolean action) je obdoba dříve zmíněné akce. Rozdíl je v tom, že boolean action umožňuje pouze nastavení hodnoty daného tagu (který se poté monitoruje z jiné části programu).

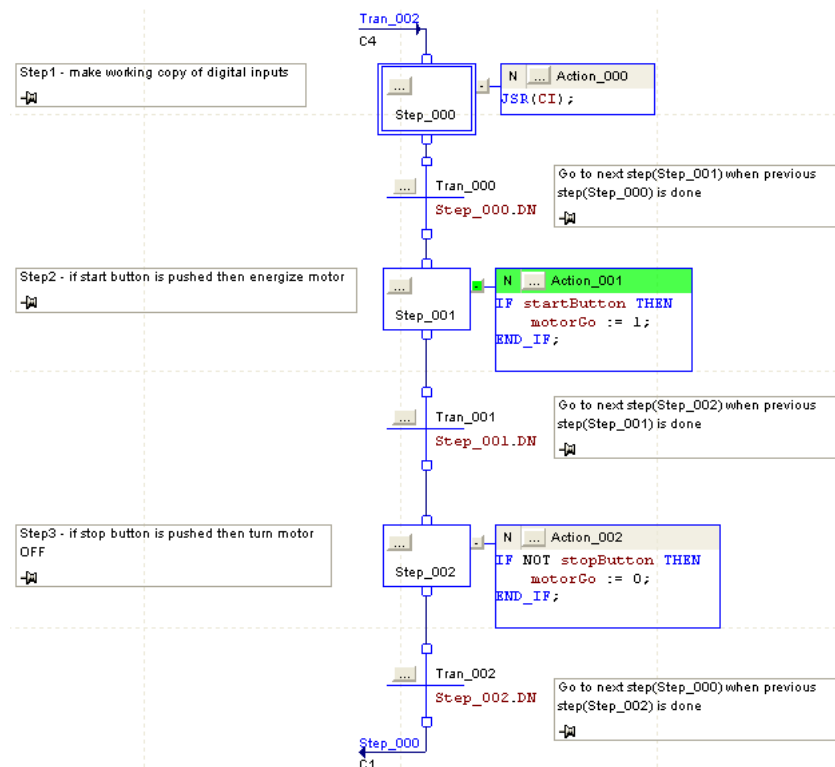


Obrázek 3.4: Lišta s nabídkou jednotlivých prvků (RSLogix 5000)

Obrázek 3.4 ukazuje nabídku všech dostupných prvků. Zleva: Step+transition (klasický krok následovaný přechodem), Step (krok), Transition (přechod), Action (akce), Boolean action (booleovská akce), Selection Branch Diverge (slouží k výběru jedné z paralelních větví na základě podmínky), Simultaneous Branch Diverge (paralelní vykonávání více větví), Converge SFC Elements (slouží ke spojení výše rozvětvené části programu), Extend Branch (rozšiřuje paralelní větve o další větev), Stop (umožňuje zastavit vykonávání programu), Soubroutine/Return (slouží k předávání parametrů), Text Box (slouží zejména k okomentování programu).

3.4.2 Ukázka programu

Na následujícím obrázku je program vytvořený v sekvečním diagramu. Tento program obstarává spuštění a zastavení motoru pomocí dvou tlačítek.



Obrázek 3.5: Ukázka programu vytvořeného pomocí sekvečního diagramu

Kapitola 4

Výukové modely

Tato kapitola slouží jako seznámení s modely podavače a vodáren. Účelem je poskytnutí základních informací o těchto modelech, které budou užitečné při další práci.

4.1 Podavač



Obrázek 4.1: Fotografie modelu podavače

Výukové modely (v lab. K23 se nacházejí 4 tyto modely) podavačů, přesněji řečeno jejich nejnovější verze, vznikly v rámci diplomové práce Jiřího Kose. Tyto modely umožňují studentům se zájmem o programovatelné automaty snáze proniknout do jejich problematiky. Na těchto modelech je možno provádět jak ty nejjednodušší úlohy vhodné pro první seznámení s PLC - například rozsvěcení LED pomocí tlačítka apod., tak i pokročilejší využívající snímání polohy, řízení servomotorku či dokonce vizualizaci.

V této podkapitole jsou popsány základní vlastnosti těchto modelů, princip jejich vnitřního zapojení a popis připojení k PLC.

4.1.1 Popis modelu

Jak je patrné z Obr. 4.1 model podavače obsahuje několik tlačítek (spínací, rozpínací, přepínací), dále několik indikačních LED diod, servomotorek a sedmisegmentový displej. Tento model také umožňuje diskrétní měření polohy motorku, která je indikována příslušnými LED diodami.

název	typ
S1 (Start)	spínací tlačítko
R0 (Stop)	rozpínací tlačítko
T2	spínací tlačítko
T3	spínací tlačítko
R4	rozpínací tlačítko
S5	přepínač
S6	přepínač
S7	přepínač
S8	přepínač
A/D	přepínač
MAN	spínací tlačítko

Tabulka 4.1: Seznam tlačítek modelu podavače

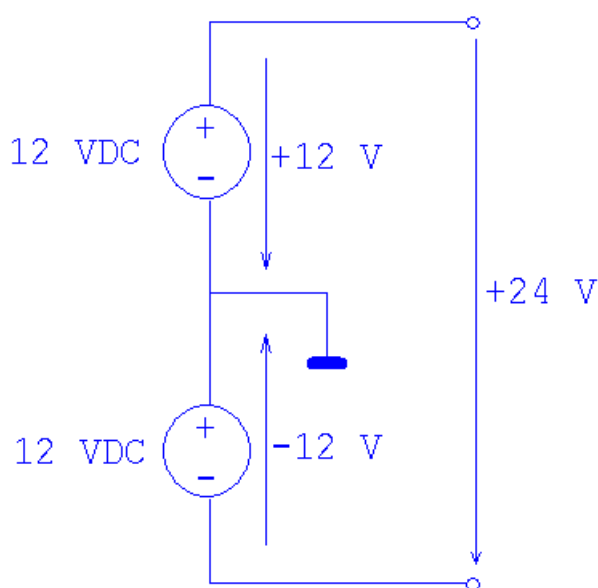
4.1.2 Vnitřní zapojení

Vnitřní zapojení modelu je navrženo tak, aby vyhovovalo požadavkům na připojení k PLC (24 V logika).

4.1.2.1 Napájení modelu

Napájení modelu bylo zvoleno tak, aby vyhovovalo požadavkům, které jsou na model kladeny. Vzhledem k tomu, že automaty ControlLogix (a programovatelné automaty obecně) využívají 24 V logiku, je nutné mít těchto 24 V k dispozici. Další požadavek na zdroj plyne z toho, že chceme-li motorem točit oběma směry, potřebujeme obě polarity napájecího napětí - tedy jak +5 V, tak -5 V.

Logickým řešením je použití dvou stejnosměrných 12 V zdrojů zapojených do série, viz Obr. 4.2.

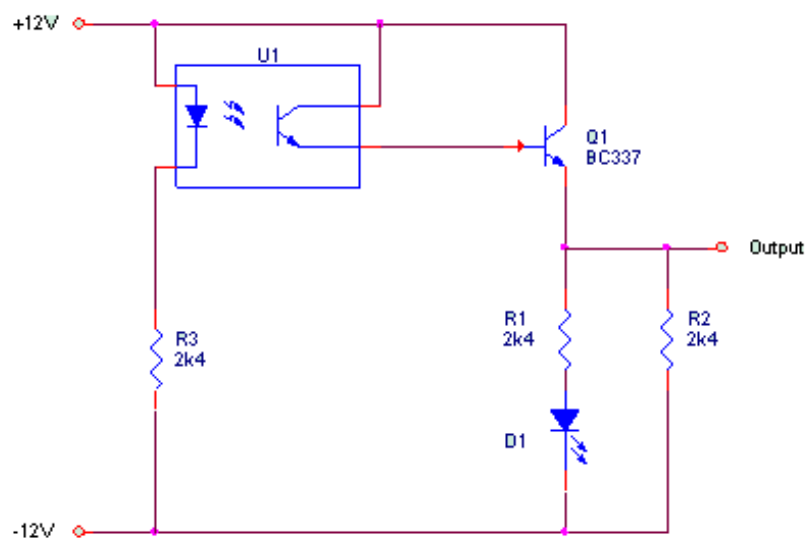


Obrázek 4.2: Napájení modelu

4.1.2.2 Princip měření polohy

Měření polohy (Obr. 4.3) je založeno na optickém principu využívajícím jako zdroj světla LED diodu a jako snímač světla fototranzistor. Světelný tok z LED diody na fototranzistor je přerušován pomocí rotačního kotoučku (Obr. 4.4), ve kterém je vyříznuta štěrbin. Je-li rotační kotouček natočen tak, že se štěrbin nachází v místě senzoru, dojde k osvětlení fototranzistoru, což způsobí jeho otevření. To má za následek otevření tranzistoru Q1 (na jeho bázi je nyní téměř +12 V). Následkem toho je rozsvícena indikační LED dioda a na výstupu se nyní objeví 24 V (napětí mezi +12 V a -12 V), které sepne příslušný vstup PLC.

Pozn.: Každý model obsahuje 6 snímačů a proto umožňuje měřit pozici na 6 místech.



Obrázek 4.3: Měření polohy



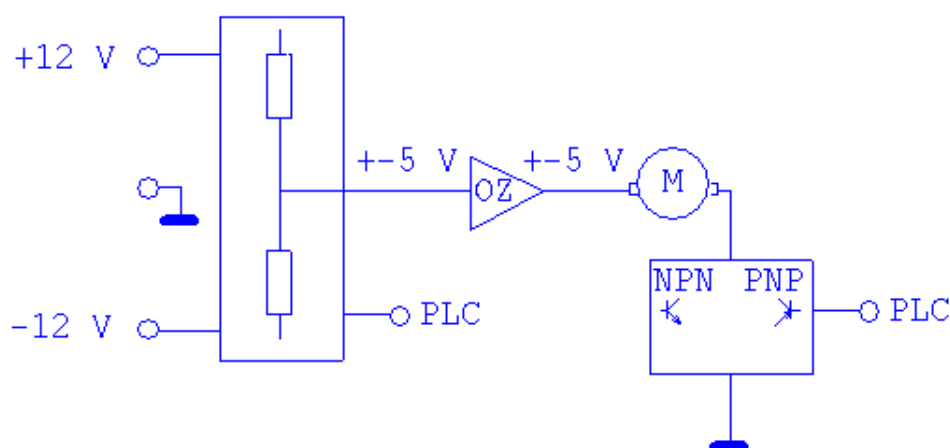
Obrázek 4.4: Fotografie clonicího kotoučku - převzato z [2]

4.1.2.3 Princip ovládání motoru

Navržené zapojení je uspořádáno tak, aby splňovalo zadané požadavky - aby umožňovalo manuální řízení pomocí tlačítka, řízení digitálními signály z PLC a umožňovalo pohyb oběma směry. Celý model je napájen napětím $\pm 12\text{ V}$ a jelikož napájecí napětí motorku je 5 V , je nutné mezi motor a zdroj vřadit obvod, který upraví 12 V na požadovaných 5 V .

V zapojení je použit princip napěťového děliče, který vytváří napětí $+5\text{ V}$ nebo -5 V v závislosti na sepnutí tranzistoru. Výstup děliče je impedančně oddělen pomocí

napětového sledovače, na jehož výstup je připojen motor. Samotné spínání motorku je poté řešeno pomocí polovičního tranzistorového můstku, ve kterém jedna větev umožňuje průchod proudu jedním směrem a druhá větev druhým. Blokové schéma je uvedeno na Obr. 4.5, celkové schéma poté v příloze [D].

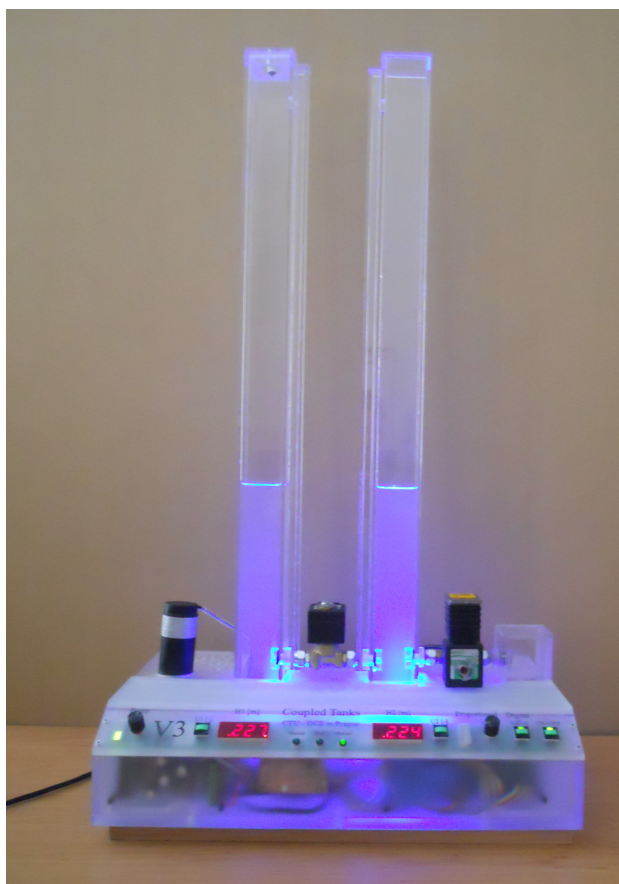


Obrázek 4.5: Blokové schéma ovládání motoru

4.1.3 Připojení k PLC

Model podavače je k automatu s adresou 147.32.87.137 připojen přes lokální vstupní a výstupní moduly. Konkrétně se jedná o vstupní digitální modul 1756-IB32/B a výstupní digitální modul 1756-OB16E/A. Zapojení jednotlivých vstupů a výstupů bylo zvoleno tak, aby odpovídalo popiskům na modelu. Proto je například tlačítko označené R0 připojeno na nulový bit vstupního modulu. Motor se spíná přes nulový bit výstupního modulu, volba směru otáčení se nastavuje prvním bitem.

4.2 Vodárna



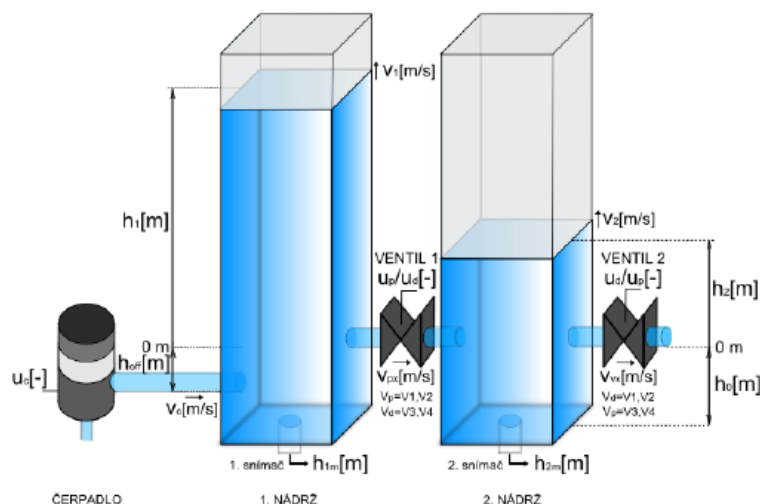
Obrázek 4.6: Fotografie modelu vodárny

Modely vodáren, které vznikly v rámci diplomové práce Jiřího Hanzlíka, jsou fyzikálním systémem modelujícím procesy, se kterými se lze setkat v mnoha odvětvích průmyslu (vodní hospodářství, chemický průmysl atd.). Jako konkrétní příklad si můžeme uvést například přečerpávací elektrárnu Dlouhé Stráně.

Tyto modely umožňují řízení výšky hladiny, což dále zajišťuje například konstantní výtokovou rychlost a tlak. V laboratoři K23 se nacházejí celkem 4 modely vodáren. Každý z nich je možno řídit třemi způsoby se stoupající prioritou: manuálně, pomocí PC a pomocí PLC. V této práci se zaměříme na řízení modelu pomocí programovatelného automatu ControlLogix od firmy Rockwell Automation.

4.2.1 Princip

Princip a konstrukce modelu je patrná z Obr. 4.7. Odstředivé čerpadlo čerpá vodu ze zásobníku do nádrže č. 1, odkud dále může odtékat přes přepouštěcí ventil (V1) do nádrže č. 2. Z nádrže č. 2 může voda odtékat zpět do zásobníku přes druhý vypouštěcí ventil (V2).



Obrázek 4.7: Princip funkce modelu - obrázek převzatý z [1]

4.2.2 Akční členy a senzory

Model je osazen průmyslovými senzory a akčními členy. V této části jsou uvedeny základní informace o použitých ventilech, čerpadlu a snímačích hladiny.

4.2.2.1 Čerpadlo

V modelu je použito čerpadlo Johnson Pump CM10P7-1, které je konstantním zdrojem tlaku. K ustálení hladiny v nádrži dojde v okamžiku, kdy se v místě přívodu kapaliny do soustavy vyrovná hydrostatický tlak s protitlakem čerpadla. Je-li hydrostatický tlak vyšší než tlak dodávaný čerpadlem, vrací se voda přes čerpadlo zpět do zásobníku.

Bližší informace lze nalézt v [25].

4.2.2.2 Snímače výšky hladiny

Hydrostatický senzor LM331 je určen pro kontinuální měření výšky hladiny kapalin. Ve vodárně jsou tyto senzory dva - jeden v nádrži č. 1 a druhý v nádrži č. 2. K měření výšky

vodního sloupce využívá tlakové čidlo DSP401. Senzor je kalibrován v metrech vodního sloupce (0 až 1 m), čemuž odpovídá výstupní napětí (0 až 10 V).

Bližší informace lze nalézt v [26].

4.2.2.3 Digitální ventil

Digitální ventil typu Asco Joucomatic SCG262A264 může být zcela otevřen, nebo zcela zavřen. V závislosti na konfiguraci modelu zajišťuje odtok vody z nádrže č. 2 zpět do zásobníku, nebo přepouštění vody z nádrže č. 1 do nádrže č. 2. V klidovém stavu je otevřený.

Bližší informace lze nalézt v [27].

4.2.2.4 Proporcionální ventil

Proporcionální ventil Asco Joucomatic SCG202A057V zajišťuje v závislosti na konfiguraci modelu odtok vody z nádrže č. 1 do nádrže č. 2, nebo odtok vody z nádrže č. 2 zpět do zásobníku. Průběh otevření ventilu v závislosti na vstupním napětí je nelineární.

Bližší informace lze nalézt v [27].

4.2.3 Řídicí elektronika

Řídicí elektronika, jejímž „mozkem“ je 8 bitový mikrokontrolér Atmega8 umožňuje ovládání modelu třemi způsoby: pomocí PLC, pomocí PC a manuální ovládání (nejvyšší prioritu má PLC, nejnižší manuální ovládání). Jednotka je napájena zdrojem 24 VDC. Tato jednotka má za úkol přijímat řídicí signály (dva analogové a dva digitální), ovládat akční členy (čerpadlo, digitální ventil, proporcionální ventil) a měřit fyzikální veličiny (výška hladiny h_1 a h_2). Výhodou této jednotky je to, že byla navržena jako univerzální a její použití se tudíž neváže na jeden konkrétní laboratorní model.

Bližší informace lze nalézt v [1].

4.2.4 Připojení k PLC

Zde budeme uvažovat model V3, který je připojen k automatu s adresou 147.32.87.137. Pro připojení slouží následující lokální vstupní a výstupní moduly:

vstupní digitální modul **1756-IB16I**,

vstupní analogový modul **1756-IF8**,

výstupní digitální modul **1756-OB16E**,
výstupní analogový modul **1756-OF4**.

1756-IB16I		1756-OB16E	
I2	stav ON/OFF	O12	ON/OFF
I3	stav DV	O10	DV
1756-IF8		1756-OF4	
I4	h1	O2	čerpadlo
I5	h2	O3	PV
I6	stav čerpadla		
I7	stav PV		

Tabulka 4.2: Připojení vodárny V3 k automatu 147.32.87.137

Kapitola 5

Řízení a vizualizace

Tato kapitola je věnována vizualizaci a řízení programovatelnými automaty ControlLogix. Jelikož účelem této práce je podpora výuky, bude zde prezentováno několik příkladů, které by studentům měly usnadnit začátky při práci s PLC. Při řešení zadaných úloh jsem se neomezoval na jeden konkrétní programovací jazyk, ale snažil jsem se naznačit možnosti všech čtyř dříve zmíněných programovacích technik.

Pro každý z dříve představených modelů je vytvořeno několik úloh, které jsou koncipovány od jednodušších ke složitějším. Jelikož každá další úloha částečně navazuje na předchozí, doporučuji tyto úlohy řešit postupně. Součástí každé úlohy je zadání (soubor zadani.pdf) a 2 projekty. První projekt (složka RSLogix5000_project_core) obsahuje pouze základní kostru programu, druhý projekt (složka RSLogix5000_project_complete) obsahuje kompletní řešení zadané úlohy. Poslední součástí je potom soubor (tip.pdf), který obsahuje stručnou nápovědu pro řešení daného problému - typ na to, jaké použít instrukce, kde vyhledat informace a podobně. U úkolů, které vyžadují vizualizaci, jsou navíc dvě další složky. První (RSView_project_core) obsahuje pouze základní kostru vizualizace, druhá (RSView_project_complete) obsahuje kompletní řešení zadané úlohy. Vytvořené projekty a soubory mohou studentovi posloužit ve dvou případech: student si se zadáním neví rady a potřebuje nápovědu, nebo jako kontrola, zda je jím zvolený postup správný. Případně se může podívat na jiné možné řešení téhož problému, které vede ke stejnému výsledku.

Dále bylo vytvořeno několik ukázkových příkladů, které popisují další zajímavé možnosti automatů ControlLogix. Některé ukázky se netýkají přímo dvou dříve zmíněných modelů. Tyto příklady ukazují, jak používat produkované/konzumované tagy, posílání zpráv mezi automaty, vytváření vlastních datových typů apod..

Veškeré úlohy jsou zamýšleny pro automat s adresou 147.32.87.137, avšak po menších

úpravách jsou bez problému použitelné i na ostatních automatech ControlLogix.



Obrázek 5.1: Fotografie automatu s adresou 147.32.87.137

5.1 Podavače

5.1.1 Úloha č. 1 - seznámení

Popis úlohy: Tato úloha si klade za cíl seznámit studenty s programovatelnými automaty ControlLogix. Hlavním smyslem této úlohy je seznámení s vývojovým prostředím RSLogix 5000, dále si studenti vyzkoušejí vyhledávání informací o I/O konfiguraci prostřednictvím RSLinxu a v neposlední řadě si naprogramují jednoduchou úlohu pro řízení motoru.

Zadání: Seznamte se s modely podavačů v laboratoři K23. V prostředí RSLogix 5000 vytvořte projekt pro automat s adresou 147.32.87.137 (procesor L63). Vyhledejte informace o způsobu připojení modelu k automatu a následně přidejte potřebné moduly. Vytvořte kontinuální úlohu, která bude pomocí jednoho tlačítka (S1 - START) spouštět motor a pomocí druhého tlačítka (R0 - STOP) motor zastavovat. Nahrajte program do automatu a následně otestujte.

Pozn.: Řešení této úlohy je navíc dostupné pro automaty 147.32.87.132, 147.32.87.133 a 147.32.87.134.

5.1.2 Úloha č. 2 - úlohy a průběhy (tasks and trends)

Popis úlohy: V této úloze si studenti vyzkoušejí přidávání/odebírání a konfiguraci úloh. Dále se seznámí s možnostmi zobrazování průběhů přímo v prostředí RSLogix 5000.

Zadání: V prostředí RSLogix 5000 vytvořte projekt pro automat s adresou 147.32.87.137 (procesor L63). Naprogramujte úlohu, která bude:

- umožňovat start i stop motoru pomocí jednoho (spínacího) tlačítka T2,
- na jednom výstupu bude blikat LED dioda s periodou 2 s,
- na dalším výstupu bude blikat LED dioda s periodou 100 ms,
- přidejte trend (graf), který bude zobrazovat stav tlačítka, které spíná motor a stav motoru,
- přidejte trend (graf), který bude zobrazovat stavy výstupů, na které jsou připojeny LED diody.

5.1.3 Úloha č. 3 - čítače

Popis úlohy: V této úloze se studenti seznámí s možnostmi, konfigurací a použitím čítačů. Dále si vyzkoušejí práci se sedmisegmentovým displejem.

Zadání: Naprogramujte úlohu, která bude simulovat počítání lahví a balení v průmyslové výrobě. Láhev bude detekována náběžnou hranou na jednom ze vstupů (tlačítko T2). Každé balení obsahuje 6 lahví. Počet lahví a balení průběžně zobrazujte na sedmisegmentovém displeji následujícím způsobem: je-li stisknuté tlačítko T3, bude na displeji zobrazen počet balení, není-li stisknuté tlačítko T3, bude na displeji zobrazen počet lahví.

5.1.4 Úloha č. 4 - časovače

Popis úlohy: V této úloze se studenti seznámí s možnostmi, konfigurací a použitím časovačů.

Zadání: Naprogramujte úlohu, která bude simulovat zpožděné zapnutí a vypnutí míchačky (motor představuje míchačku). Míchačka bude ovládána přepínacím tlačítkem S5.

K roztočení míchačky dojde 3 s po stisknutí tlačítka, k vypnutí poté 5 s po jeho uvolnění. Přidejte trend, který bude zobrazovat stav tlačítka, motoru a aktuální hodnoty čítačů.

5.1.5 Úloha č. 5 - přerušení

Popis úlohy: V této úloze se studenti seznámí s možnostmi, konfigurací a použitím úlohy řízené událostí (event task).

Zadání: Ve vývojovém prostředí RSLogix 5000 vytvořte program, který bude umožňovat řízení motoru pomocí jednoho spínacího tlačítka (T2). Při každém stisknutí tohoto tlačítka

dojde ke změně stavu motoru (byl-li zastavený, pak se rozběhne a naopak). Při řešení této úlohy využijte přerušení.

5.1.6 Úloha č. 6 - zásobníky

Popis úlohy: V této úloze se studenti seznámí s možnostmi použití zásobníků.

Zadání: Seznamte se s možnostmi použití zásobníku typu FIFO. Vytvořte program, který bude tento zásobník využívat pro ukládání požadovaných pozic motoru. Požadovaná pozice (0 až 6) se bude nastavovat pomocí tlačítka R4 a bude průběžně zobrazována na displeji. Při každém stisku tlačítka R4 dojde k inkrementaci hodnoty zobrazené na displeji. Zásobník bude umožňovat zadání až 5 požadovaných pozic a bude plněn při stisknutí tlačítka T2 - při stisku se do zásobníku uloží hodnota z displeje. Při stisknutí tlačítka T3 dojde k vyzvednutí hodnoty ze zásobníku a v závislosti na přečtené hodnotě bude motor natočen do příslušné pozice.

5.1.7 Úloha č. 7 - základní vizualizace

Popis úlohy: Tato úloha si klade za cíl seznámit studenty s komplexním vizualizačním nástrojem RSView32 a s konfigurací spojení mezi automatem a vizualizací.

Zadání: Rozšiřte úlohu č. 1 (start/stop motoru pomocí dvou tlačítek) o jednoduchou vizualizaci vytvořenou v prostředí RSView32. Ke spojení mezi automatem a vizualizací použijte OPC Server. Vizualizace bude obsahovat jedno okno, které bude obsahovat graf zobrazující aktuální stav výstupu (motoru) a 3 tlačítka:

- Motor Start - tlačítko pro spuštění motoru,
- Motor Stop - tlačítko pro zastavení motoru,
- End - tlačítko pro ukončení vizualizace (při ukončení vizualizace dojde k zastavení motoru).

5.2 Vodárny

5.2.1 Úloha č. 1 - seznámení s modelem vodáren

Popis úlohy: Účelem této úlohy je seznámit studenta s modelem vodáren. V rámci této úlohy není vykonáván žádný program, úkolem je pouze najít informace o připojení modelu k automatu, nakonfigurovat potřebné moduly a otestovat funkci jednotlivých vstupů a výstupů.

Zadání: Seznamte se s modely vodáren v laboratoři Allen-Bradley. V prostředí RSLogix 5000 vytvořte projekt pro automat s adresou 147.32.87.137 (procesor L63), ke kterému je připojená vodárna V3. Vyhledejte informace o způsobu připojení modelu k automatu a následně přidejte potřebné moduly. Dbejte především na správné nastavení analogových modulů (rozsahy, alarmy apod.). Nahrajte program do automatu a ověřte funkci jednotlivých vstupů a výstupů (pokuste se zapnout čerpadlo, otevřít digitální ventil, přečíst hodnotu hladiny).

Pozn.: Řešení této úlohy je navíc dostupné pro automat 147.32.87.135 (vodárna V2).

5.2.2 Úloha č. 2 - PID regulace

Popis úlohy: V této úloze je po studentech požadována implementace PID regulátoru, který bude regulovat výšku hladiny v první nádrži (h1).

Zadání: Předchozí cvičení rozšiřte o regulaci výšky hladiny h1. Pro regulaci použijte PID regulátor. Regulace by měla probíhat v periodické úloze, periodu této úlohy zvolte s ohledem na dynamiku modelu. Ověřte chování navrženého regulátoru na poruchu, kterou může představovat otevírání/zavírání digitálního ventilu, nebo změna stavu proporcionálního ventilu (při otevřeném digitálním ventilu).

5.2.3 Úloha č. 3 - vodárna - vizualizace 1

Popis úlohy: V této úloze mají studenti za úkol vytvořit mírně pokročilejší vizualizaci, která bude poskytovat informace o stavu modelu (výšky hladin, stavy ventilů a čerpadla) a umožní uživateli ovládat model (prostřednictvím akčních členů).

Zadání: Úlohu č. 1 doplňte o vizualizaci. Tato vizualizace se bude skládat z jednoho hlavního okna, které bude:

- zobrazovat výšky hladin, napětí na proporcionálním ventilu a čerpadle, stav digitálního ventilu,
- obsahovat tlačítko pro zapnutí/vypnutí modelu,
- obsahovat tlačítko pro otevření/zavření digitálního ventilu,
- umožňovat nastavení hodnoty napětí na čerpadle,
- umožňovat nastavení hodnoty napětí na proporcionálním ventilu,
- obsahovat tlačítko pro ukončení vizualizace (při ukončení vizualizace dojde k vypnutí modelu).

Při tvorbě vizualizace dbejte na přehlednost a logické rozmístění prvků.

5.2.4 Úloha č. 4 - vodárna - vizualizace 2

Popis úlohy: V této úloze mají studenti za úkol vytvořit mírně pokročilejší vizualizaci, která vychází z předchozího cvičení. Tato úloha přidává navíc logování do souboru.

Zadání: Úlohu č. 2 doplňte o vizualizaci. Vizualizace se bude skládat z jednoho hlavního okna, které bude uživateli umožňovat:

- nastavit žádanou výšku hladiny h_1 ,
- nastavit konstanty PID regulátoru,
- vypnout model pomocí tlačítka,
- otevřít/zavřít digitální ventil pomocí tlačítka.

Vizualizace bude také zobrazovat aktuální hodnoty hladin, stav digitálního ventilu a stav čerpadla. Přidejte logování, které bude obsahovat informace o výšce hladiny h_1 , stavu čerpadla, žádané hodnoty a stavu digitálního ventilu.

Kapitola 6

Podpora výuky

Pro co nejjednodušší a nejrychlejší proniknutí do problematiky programovatelných automatů bylo v rámci mé práce vytvořeno několik návodů, s cílem názornou formou seznámit studenta s danou částí problematiky. Jedná se zejména o návody k softwarovým nástrojům od firmy Rockwell Automation, odkazy na nejdůležitější literaturu apod..

Většina těchto návodů má formu screenshotů, které krok po kroku ukazují, jak provést danou úlohu. Veškeré kroky a obrázky jsou bohatě okomentované tak, aby pokud možno nedošlo k nejasnostem. Konkrétním příkladem je návod na vytvoření projektu, přidání modulů a otestování viz Obr. 6.1.

- literature_overview.pdf - dokument obsahuje seznam z mého pohledu nejdůležitější literatury,
- RSLinx_find.info.pdf - dokument popisuje, jak vyhledat informace o konfiguraci automatu (typy modulů, jejich revize atd.),
- RSLinx_OPC_topic_config.pdf - dokument popisuje konfiguraci spojení mezi vizualizací a automatem,
- RSLogix5000_analog_module_configuration.pdf - dokument popisuje konfiguraci analogových modulů,
- RSLogix5000_create_project.pdf - dokument popisuje vytvoření projektu v prostředí RSLogix 5000,
- RSLogix5000_trends.pdf - dokument popisuje množnosti zobrazování průběhů v prostředí RSLogix 5000,

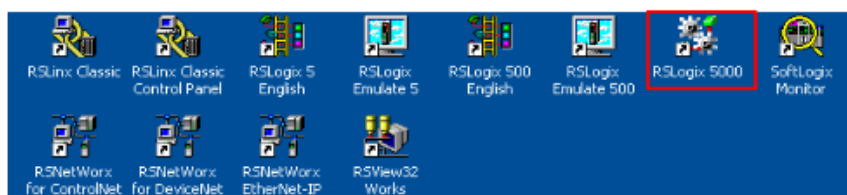
- RSVIEW32_create_project.pdf - dokument popisuje vytvoření projektu v prostředí RSVIEW32,
- RSVIEW32_data_logging.pdf - dokument popisuje konfiguraci logování v prostředí RSVIEW32.

RSLogix 5000 – vytvoření projektu

Tento dokument popisuje, jak vytvořit a nakonfigurovat projekt v prostředí RSLogix 5000. Je zde popsán postup konfigurace a otestování projektu pro práci s modelem podavače, který se nachází v laboratoři Allen-Bradley (K23) na katedře Řídící techniky. Pro ukázkou bude uvažován model připojený k automatu ControlLogix s adresou 147.32.87.133.

1) Spuštění vývojového prostředí RSLogix 5000

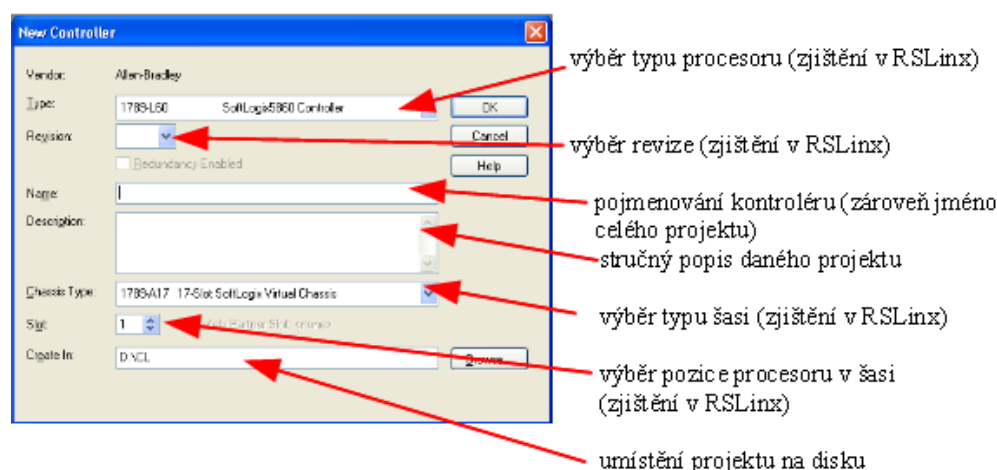
- přes zástupce na ploše
- není-li dostupný zástupce na ploše, tak přes **Start / Programy / Rockwell Software / RSLogix 5000 Enterprise Series / RSLogix 5000**



2) Vytvoření nového projektu

- klasicky položka menu **File / New** → dojde k otevření následujícího dialogu:

POZN.: V tomto kroku již bude třeba zjišťovat některé informace prostřednictvím RSLinx.



Obrázek 6.1: Ukázka části návodu popisujícího vytvoření projektu v prostředí RSLogix 5000

Kapitola 7

Závěr

Tato práce se věnovala vizualizaci a řízení programovatelnými automaty. Celou práci lze rozdělit do čtyř hlavních částí. První část práce se zabývala teoretickým seznámením s programovatelnými automaty, druhá měla za cíl seznámení s vybranými výukovými modely nacházejícími se v laboratoři Allen-Bradley, třetí byla věnována praktickému použití programovatelných automatů ControlLogix a poslední popisovala vzniklé výukové materiály.

V rámci této práce bylo úspěšně vytvořeno velké množství studijních příkladů. Tyto příklady jsou koncipovány jako cvičení, které mají studenti splnit. Ke každému cvičení je dostupné kompletní řešení plus základní šablona. Student má tedy možnost řešit celý problém naprosto samostatně, nebo může využít zmíněnou šablonu.

Dalšími výstupy jsou dokumenty popisující práci se softwarovými nástroji a nejdůležitější literaturu. Většina těchto návodů byla pojata formou popsaných snímků obrazovky, které mají studentům názornou formou naznačit, jak se daná část provádí.

I přes veškerou snahu poskytnout široký pohled na dané téma se vzhledem k rozsahu problematiky programovatelných automatů práce nevěnuje mnoha dalším zajímavým zákoutím tohoto oboru. Pro komplexnější proniknutí do problematiky je samozřejmě od zájemců vyžadováno značné množství dalšího samostudia. I přesto se domnívám, že práce poskytuje slušný základ a studenti tak budou mít na čem stavět.

Literatura

- [1] *J. HANZLÍK*. Distribuovaný řídicí systém s automaty Rockwell Automation. Diplomová práce, ČVUT, Fakulta elektrotechnická, 2004.
- [2] *J. KOS*. Průmyslové modely a měření. Diplomová práce, ČVUT, Fakulta elektrotechnická, 2007.
- [3] *L. HALÍŘ*. Vizualizace a řízení výukových modelů. Bakalářská práce, ČVUT, Fakulta elektrotechnická, 2012.
- [4] *P. CHLAD*. Vizualizace manuálního ovládání modelu manipulátoru. Bakalářská práce, VUT, Fakulta elektrotechniky, 2010.
- [5] *J. NOVÁK*. Model lineárního servomechanismu a jeho řízení. Diplomová práce, ČVUT, Fakulta elektrotechnická, 2002.
- [6] *M. KONOPA*. Řízení výukových modelů typu spojené nádoby. Bakalářská práce, ČVUT, Fakulta elektrotechnická, 2009.
- [7] *J. NEKVINDA*. Příprava projektově orientované výuky řídicích systémů. Diplomová práce, ČVUT, Fakulta elektrotechnická, 2004.
- [8] *ROCKWELL AUTOMATION*. Logix5000 Controllers Design Considerations [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 1756-RM0946-EN-P, 2011.
- [9] *ROCKWELL AUTOMATION*. Logix5000 Controllers General Instruction Reference Manual [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 1756-RM003-EN-P, 2011.
- [10] *ROCKWELL AUTOMATION*. RSView32 User's Guide [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: vw-UM001F-EN-E, 2011.

- [11] *ROCKWELL AUTOMATION*. Logix5000 Controllers Tasks, Programs and Routines [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 1756-PM005C-EN-P, 2009.
- [12] *ROCKWELL AUTOMATION*. Logix5000 Controllers [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 1756-QR107C-EN-P, 2005.
- [13] *ROCKWELL AUTOMATION*. Logix5000 Controllers Ladder diagram [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 1756-PM008A-EN-P, 2007.
- [14] *ROCKWELL AUTOMATION*. Logix5000 Controllers I/O and Tag Data [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 1756-PM004C-EN-P, 2009.
- [15] *ROCKWELL AUTOMATION*. RSLogix 5000 Getting Started with RSLogix 5000 [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 9399-RLD300GR-AUG00, 2011.
- [16] *ROCKWELL AUTOMATION*. RSView32 Getting Results Guide [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 1vw32-GR0001F-EN-E, 2011.
- [17] *ROCKWELL AUTOMATION*. ControlLogix ControlNet Interface Module [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: 1756-RN627E-EN-P, 2009.
- [18] *ROCKWELL AUTOMATION*. RSNetWorx for ControlNet Getting Results Guide [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: CNET-GR001E-EN-E, 2011.
- [19] *ROCKWELL AUTOMATION*. NetLinx Selection Guide [online]. Dostupné z WWW: <http://www.ab.com/literature>, číslo publikace: NETS-SG001C-EN-P, 2009.
- [20] *ROCKWELL AUTOMATION*. Comparing performance of L7x vs L6x using Logix5000 Task Monitor [online]. Dostupné z WWW: <http://www.rockwellautomation.com/solutions/integratedarchitecture/get/LogixTaskMonitorInstruction.pdf>.

- [21] *M. KUTIL*. Programovatelné automaty Allen-Bradley. Studijní návod, ČVUT, Fakulta elektrotechnická, 2005.
- [22] *M. MARTINÁSKOVÁ, L. ŠMEJKAL*. Řízení programovatelnými automaty. Vysokoškolská skripta, ČVUT, Fakulta strojní.
- [23] *Jon Stenerson*. Fundamentals of Programmable Logic Controllers, Sensors, and Communications. Prentice hall, 1999, ISBN:0-13-746124-0.
- [24] *Gary Dunning*. Introduction to Programmable Logic Controllers 2nd Edition. Delmar, 2002, ISBN:0-7668-1768-7.
- [25] *JOHNSON PUMP*. <http://www.dc-pump.cz/> [online].
- [26] *SMARIS s.r.o.* <http://www.smaris.cz/> [online].
- [27] *ASCO JOUCOMATIC s.r.o.* <http://www.ventil.cz/> [online].

Příloha A

Obsah přiloženého CD

K této práci je přiloženo CD, na kterém jsou uloženy ukázkové úlohy, cvičení pro studenty, manuály, studijní návody a text bakalářské práce.

- BP - text bakalářské práce.
- K23 - složka připravená k umístění na PC v laboratoři.
- MAN - manuály v elektronické podobě.
- NAV - studijní návody.
- STUD - cvičení a ukázkové úlohy.
- README.txt - obsahuje informace o obsahu CD.

Příloha B

Seznam zkratek

- DC (Direct Current) - stejnosměrný proud.
- FIFO (First In First Out) - zásobník.
- LED (Light Emmiting Diode) - světlo vyzařující dioda.
- PAC (Programmable Automation Controller) - programovatelný řídicí automat.
- PC (Personal Computer) - osobní počítač.
- PLC (Programmable Logic Controller) - programovatelný logický automat.
- PID (Proportional Integral Derivate controller) - regulátor s proporcionální, integrační a derivační složkou.
- RPI (Request Packet Interval) - požadovaný interval posílání dat.
- SW (Software) - software.
- USB (Universal Serial Bus) - univerzální sériová sběrnice.

Příloha C

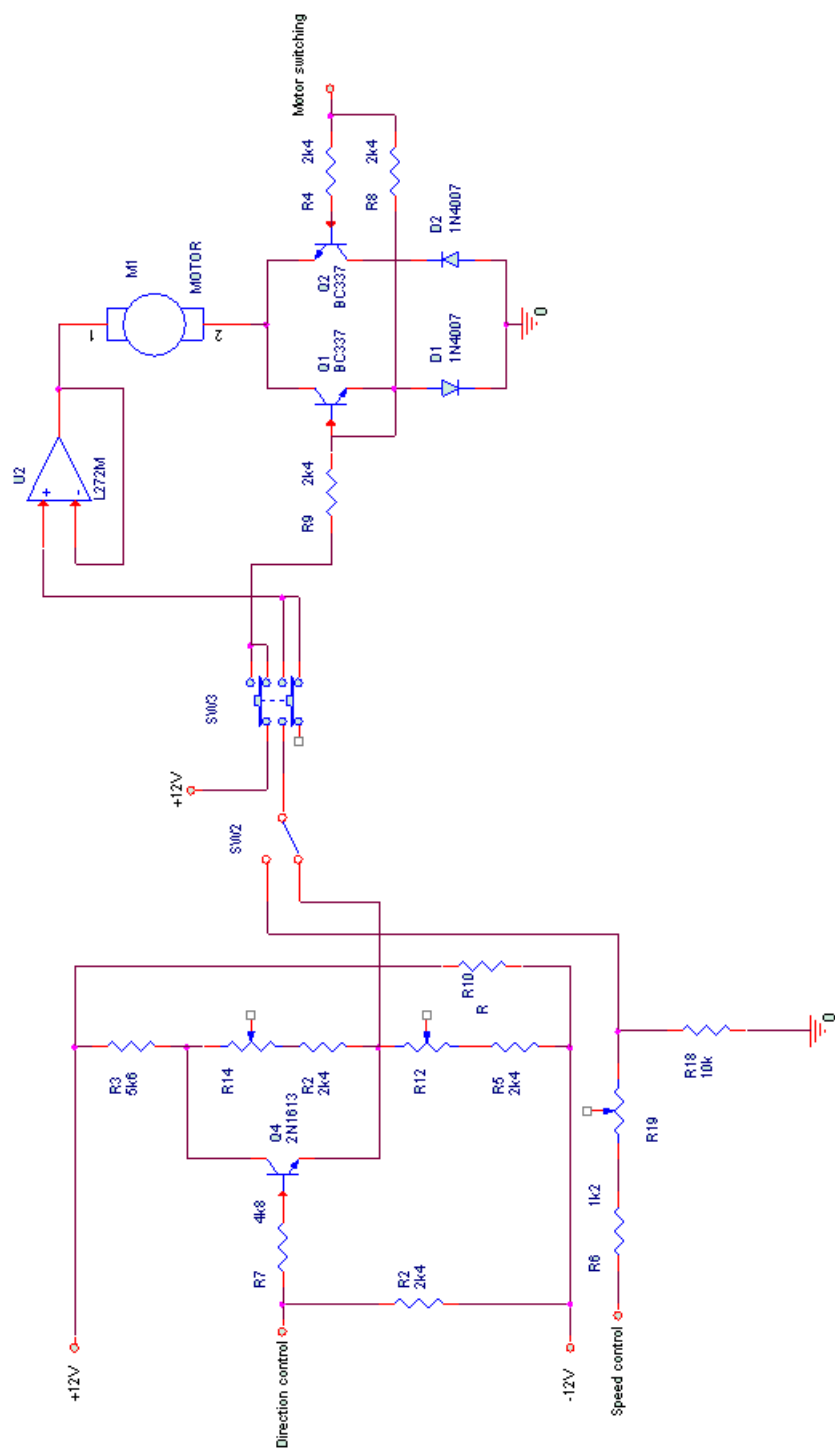
Použitý software

- MikTex. Verze 2.9, Christian Schen, 2011 .
- OpenOffice.org. Verze 3.1.1, Sun Microsystems, Inc., 2009.
- OrCAD Capture CIS Lite. Verze 16.5, Cadence Design Systems, Inc., 2011.
- PhotoFiltre. Verze 6.3.2, Antonio Da Cruz, 2009.
- RSLinx Classic. Verze 2.57.00, Rockwell Automation, Inc., 2010.
- RSNetWorx for ControlNet. Verze 10.01.00, Rockwell Automation, Inc., 2010.
- RSLogix 5000. Verze 15.02.00, Rockwell Software, Inc, 1997 - 2005.
- Logix5000 Task monitor. Verze 2.4.0, Rockwell Automation, 2006 - 2010.
- RSView32. Verze 7.50.00, Rockwell Automation Technologies, Inc., 2008.

Příloha D

Podavač - celkové schéma zapojení

Na následující straně (Obr. D.1) je uvedeno celkové schéma modelu podavače. Oproti součástem popsaných v Kapitole 4 je doplněno o možnost manuálního spínání pomocí tlačítka (SW3) a součást umožňující analogové řízení. Vzhledem k tomu, že v této práci je model využíván pro seznámení s automaty ControlLogix, není zde možnost analogového řízení využita.



Obrázek D.1: Celkové schéma ovládání motoru

Příloha E

Konfigurace automatů v laboratoři Allen-Bradley

V této příloze je uvedena konfigurace jednotlivých automatů ControlLogix v laboratoři Allen-Bradley.

Pozn.: Adresa automatů v síti ControlNet je rovna poslednímu číslu jejich IP adresy (například automat s IP adresou 147.32.87.132 má v síti ControlNet adresu 2).

slot	označení modulu	popis modulu
0	1756-L62	ControlLogix5562 procesor
1	1756-IB16ISOE/A	vstupní stejnosměrný digitální modul
2	1756-OB16IS/A	výstupní stejnosměrný digitální modul
3	1756-OA8/A	výstupní střídavý digitální modul
4	1756-IF4FXOF2F/A	vstupně/výstupní analogový modul
5	1756-CNBR/E	komunikační modul ControlNet
6	1756-ENBT/A	komunikační modul Ethernet
7	neobsazeno	
8	1756-L63	ControlLogix5563 procesor
9	neobsazeno	

Tabulka E.1: Konfigurace automatu 147.32.87.132 (šasi 1756-A10/A)

PŘÍLOHA E. KONFIGURACE AUTOMATŮ V LABORATOŘI ALLEN-BRADLEYVII

slot	označení modulu	popis modulu
0	1756-L62	ControlLogix5562 procesor
1	1756-IB16D/A	vstupní digitální modul
2	1756-OB16D/A	výstupní digitální modul
3	1756-IF6I/A	vstupní analogový modul
4	1756-OF6VI/A	výstupní analogový modul
5	1756-CNB/E	komunikační modul ControlNet
6	1756-ENBT/A	komunikační modul Ethernet
7	neobsazeno	
8	neobsazeno	
9	neobsazeno	
10	neobsazeno	

Tabulka E.2: Konfigurace automatu 147.32.87.133 (šasi 1756-A10/A)

slot	označení modulu	popis modulu
0	1756-L62	ControlLogix5562 procesor
1	1756-IB16D/A	vstupní digitální modul
2	1756-OB16D/A	výstupní digitální modul
3	1756-IF6I/A	vstupní analogový modul
4	1756-OF8/A	výstupní analogový modul
5	1756-CNBR/B	komunikační modul ControlNet
6	1756-ENBT/A	komunikační modul Ethernet
7	1756-L1/A	ControlLogix5550 procesor
8	1756-DNB	komunikační modul DeviceNet
9	1756-DRHIO/D	komunikační modul DH+ a RIO
10	1756-IB32/B	vstupní digitální modul
11	neobsazeno	
12	neobsazeno	

Tabulka E.3: Konfigurace automatu 147.32.87.134 (šasi 1756-A10/A)

PŘÍLOHA E. KONFIGURACE AUTOMATŮ V LABORATOŘI ALLEN-BRADLEYVIII

slot	označení modulu	popis modulu
0	1756-L1/A	ControlLogix5550 procesor
1	1756-IB32/A	vstupní digitální modul
2	1756-OB16E/A	výstupní digitální modul
3	1756-IF8/A	vstupní analogový modul
4	1756-OF6VI/A	výstupní analogový modul
5	1756-MO2AE	pohybový modul
6	1756-OW16I/A	výstupní digitální modul
7	1756-IN16/A	vstupní digitální modul
8	1756-CNB/D	komunikační modul ControlNet
9	1756-DHRIO/B	komunikační modul DH+ a RIO
10	neobsazeno	
11	neobsazeno	
12	1756-EWEB/A	komunikační modul Ethernet

Tabulka E.4: Konfigurace automatu 147.32.87.135 (šasi 1756-A13/A)

slot	označení modulu	popis modulu
0	1756-L62	ControlLogix5562 procesor
1	1756-IB16/A	vstupní digitální modul
2	1756-OB16I/A	výstupní digitální modul
3	1756-IR6I/A	vstupní analogový modul
4	1756-OF4/A	výstupní analogový modul
5	1756-CNB/B	komunikační modul ControlNet
6	neobsazeno	
7	neobsazeno	
8	neobsazeno	
9	ENBT/A	komunikační modul Ethernet
10	neobsazeno	

Tabulka E.5: Konfigurace automatu 147.32.87.136 (šasi 1756-A10/A)

PŘÍLOHA E. KONFIGURACE AUTOMATŮ V LABORATOŘI ALLEN-BRADLEYIX

slot	označení modulu	popis modulu
0	1756-L63	ControlLogix5563 procesor
1	1756-IB32/B	vstupní digitální modul
2	1756-OB16E/A	výstupní digitální modul
3	1756-IF8/A	vstupní analogový modul
4	1756-OF4/A	výstupní analogový modul
5	1756-CNBR/D	komunikační modul ControlNet
6	1756-MO3SE	pohybový modul
7	1756-OB16E/A	výstupní digitální modul
8	1756-HSC/A	modul čítače
9	1756-IB16I/A	vstupní digitální modul
10	neobsazeno	
11	1756-L1/A	ControlLogix5550 procesor
12	1756-EWEB/A	komunikační modul Ethernet

Tabulka E.6: Konfigurace automatu 147.32.87.137 (šasi 1756-A13/A)