

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Laboratorní úloha "Řízení programovatelným
logickým automatem (PLC)"

Petr Suchý

Bakalářská práce

2009

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra informačních technologií
Akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr SUCHÝ**

Studijní program: **B2646 Informační technologie**

Studijní obor: **Informační technologie**

Název tématu: **Laboratorní úloha "Řízení programovatelným logickým
automatem (PLC)"**

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

Cíl: Cílem je vytvořit laboratorní úlohu "Řízení programovatelným logickým automatem (PLC)". Obsah teoretické části: Teorie logického řízení. Programovatelné logické automaty a inteligentní relé - hardware a programování. Obsah implementační části: Instalace softwaru pro programování automatu, simulaci a vizualizaci úlohy logického řízení. Vytvoření jednoduché logické úlohy jak po hardwarové tak i po softwarové stránce. Rozsah: 30 stránek.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

SLC 500 Instruction Set Reference Manual : SLC 500, 1747, small logic controller. Rockwell Automation, 2008-11-20. Dostupný z WWW: <http://literature.rockwellautomation.com/idc/groups/literature/documents/rm,rm001_-en-p.pdf>. RSView32 UG book. Rockwell Automation, 2007-05-29. Dostupný z WWW: <http://literature.rockwellautomation.com/idc/groups/literature/documents/um-um001_-en-e.pdf>. RSView Machine Edition. Rockwell Automation, 2002-02-27. Dostupný z WWW: <www.software.rockwell.com/download/training/detinfo/rst-lnx-tgde.pdf>. ŠMEJKAL, Ladislav, MARTINÁSKOVÁ, Marie. PLC a automatizace 1. [s.l.] : Ben, 1999. 223 s. ISBN 8086056589.

Vedoucí bakalářské práce:

Ing. Daniel Honc, Ph.D.

Katedra řízení procesů

Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**



doc. Ing. Simeon Karamazov, Dr.

děkan



Ing. Lukáš Čegan
vedoucí katedry

V Pardubicích dne 31. března 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 14. 5. 2009

Petr Suchý

Poděkování

Rád bych touto cestou poděkoval vedoucímu mé bakalářské práce Ing. Danielovi Honcovi za cenné rady a připomínky, za jeho čas, trpělivost a ochotu při řešení problémů vzniklých při zpracování této práce.

Anotace

Práce se zabývá programovatelnými logickými automaty a teorií logického řízení. Je zaměřená především na automaty Allen-Bradley, jejich hardware a programování. V rámci práce byly řešeny úlohy řízení modelu automatické pračky.

Klíčová slova

PLC, programovatelné logické automaty, automatizace, řízení

Title

Laboratory work "Control by programmable logic controller (PLC)"

Annotation

The work deals with programmable logic controllers and theory of logic control. It is aimed mostly to Allen-Bradley controllers, its hardware and programming. Within the scope of the work several tasks with washing machine control were solved.

Keywords

PLC, programmable logic controllers, automatization, control

Obsah

ÚVOD	9
1 PROGRAMOVATELNÉ LOGICKÉ AUTOMATY (PLC)	10
1.1 Výhody programovatelných logických automatů.....	11
1.2 Konfigurace PLC.....	11
1.2.1 Mikro PLC.....	12
1.2.2 Kompaktní PLC.....	12
1.2.3 Modulární PLC.....	13
1.2.4 Soft PLC.....	13
1.3 Hardware.....	14
1.4 Programování.....	16
1.4.1 Programovací jazyky.....	16
1.4.2 Vykonávání programu.....	18
1.4.3 Způsob vykonávání programu PLC.....	18
2 TEORIE LOGICKÉHO ŘÍZENÍ	20
2.1 Logické funkce.....	20
2.2 Booleova algebra.....	23
2.3 Pravdivostní tabulka.....	25
2.4 Karnaughova mapa.....	26
2.5 Minimalizování logických funkcí.....	27
2.6 Kombinační logické obvody.....	30
2.6.1 Liniové schéma.....	31
2.6.2 Blokovaná schémata.....	32
3 POPIS LABORATORNÍHO MODELU	35
3.1 Allen Bradley SLC 500.....	35
3.1.1 Použité moduly.....	35

3.1.2	RS Linx.....	37
3.1.3	RS Logix.....	37
3.1.4	Programování automatu v RS Logix.....	38
3.2	Model pračky	40
3.2.1	Zapojení	41
4	ÚLOHY ODZKOUŠENÉ NA MODELU.....	44
4.1	Úloha 1.....	44
4.2	Úloha 2.....	46
4.3	Úloha 3.....	47
4.4	Úloha 4.....	47
4.5	Úloha 5.....	49
4.6	Úloha 6.....	50
4.7	Úloha 7.....	50
4.8	Úloha 8.....	51
4.9	Úloha 9.....	51
4.10	Úloha 10.....	53
	ZÁVĚR	54
	LITERATURA:	55
	SEZNAM OBRÁZKŮ A TABULEK.....	56
	Seznam obrázků:.....	56
	Seznam tabulek:	56

Úvod

PLC je zkratkou pro programovatelné logické automaty. Je to malý programovatelný počítač, který slouží k automatizaci elektromechanických procesů.

Téma programování PLC mě oslovilo již na střední škole, kdy jsem měl příležitost naprogramovat menší automat výrobní linky. Programování PLC je v jisté části také podobné programování mikropočítačů, které mě velmi bavilo.

Cílem této bakalářské práce je seznámení s problematikou programovatelných logických automatů a jejich programování, uvedení do teorie logického řízení a vytvoření laboratorní úlohy, vhodné pro výuku programování logických automatů.

Práce je rozdělena do čtyř kapitol, na teoretickou a praktickou část. První kapitola se zabývá programovatelnými logickými automaty, jejich výhodami, rozdělením, hardwarem a programováním. Ve druhé kapitole je popsána teorie logického řízení. Obě tyto kapitoly jsou teoretickou částí práce. Třetí a čtvrtá kapitola patří do části praktické. Kapitola třetí obsahuje informace o hardwarové stránce laboratorní úlohy a je úvodem do programování automatů Allen-Bradley SLC 500. V závěrečné kapitole je popsáno zadání jednotlivých úloh a nastíněno jejich řešení.

Nezbytným vybavením pro realizaci praktické části je automat SLC 500, model automatické pračky a počítač s nainstalovaným operačním systémem Windows 98.

Výsledkem je funkční program pro PLC automat obsluhující model automatické pračky a simulující její chování.

1 Programovatelné logické automaty (PLC)

PLC – Programovatelný logický automat z anglického Programmable Logic Controller, je malý programovatelný počítač, který se používá v automatizaci elektromechanických procesů. Tyto automatizované procesy jsou například využívány k řízení výrobní linky, ovládání některých robotů nebo i k řízení vytápění některých objektů. Využití těchto programovatelných automatů je téměř neomezené. Na rozdíl od počítačů je PLC navrženo pro hromadné využití vstupních a výstupních periférií. Programy pro ovládání tohoto automatu jsou uloženy ve vlastní paměti daného zařízení. PLC je příkladem real time systému, kdy výsledky na výstupu musí odpovídat podmínkám vzniklým na vstupu do tohoto systému v určitém časově ohraničeném intervalu.

„Původně byly PLC určeny pro řízení strojů, jako náhrada za pevnou reléovou logiku. Tomu odpovídal i programovací jazyk kontaktních (reléových) schémat. Jazyky prvních PLC disponovaly několika příkazy (typicky 8 nebo 16) které byly ekvivalentní spínacímu a rozpínacímu kontaktu, paralelnímu a sériovému řazení, cívce, obvodům paměti, čítače a časovače. Dnes je pro každý programovatelný automat k dispozici několik typů jazyků: kromě jazyka kontaktních schémat to bývá jazyk logických schémat, jazyk mnemokódů nebo jiný textový jazyk, nově i jazyk sekvenčního programování. Dnešní programovací jazyky jsou podstatně bohatší – bohužel se poněkud vzdálily mentalitě konstruktérů a projektantů, takže vznikla samostatná profese ‚programátor PLC‘. Sjednocení programovacích jazyků a vývojových systémů pro PLC je cílem nové mezinárodní normy IEC 1131-3.“¹

Velkou předností programovatelných automatů je jejich univerzálnost. Dnes už umí kromě logických úloh řešit také velmi obtížné úlohy (např. vazby mezi regulací různých veličin – teploty a vlhkosti), nahrazují různé

¹ ŠMEJKAL, L. a MARTINÁSKOVÁ, M. 2007. PLC a automatizace, s. 21-22.

specializované a řídicí systémy (např. regulátory, komunikační adaptéry, analyzátoři...), lze jím optimalizovat technologický proces a přizpůsobovat ho měnícím se podmínkám, minimalizovat spotřebu, náklady a ztráty. Některé PLC mají zabudovanou i fuzzy logiku, čímž se rozšiřuje oblast jejich využití například do diagnostiky a zabezpečovací techniky.²

1.1 Výhody programovatelných logických automatů

Rychlá realizace – je velkou předností programovatelných logických automatů. Uživatel nemusí vyvíjet technické vybavení. Stačí navrhnout a objednat vhodnou sestavu modulů automatu, vytvořit projekt, napsat a odladit program a uvést do chodu.

Spolehlivost, odolnost – PLC musí být navrženo tak, aby vydrželo i v těžkých průmyslových podmínkách. Je odolné proti působení různých vnějších vlivů, rušení i poruchám a bývá vybaveno diagnostickými funkcemi pro kontrolování činnosti systému, které napomáhají k lokalizování a odstranění vzniklé závady.³

1.2 Konfigurace PLC

Skutečnou sestavu si volí sám uživatel podle řešené úlohy. Někdy mohou některé moduly zcela chybět a jiné se několikrát opakovat. PLC může být sestaveno jako čistě binární systém s použitím pouze binárních (dvouhodnotových) vstupů a výstupů, nebo jako čistě analogový systém.

Mohou existovat i sestavy pouze vstupní, kdy automat slouží například pouze pro předzpracování dat, které předává dál pro zpracování.

² ŠVARC, I. 2005. Automatizace, s. 30.

³ ŠMEJKAL, L. a MARTINÁSKOVÁ, M. 2007. PLC a automatizace, s. 39.

Podobně můžou existovat i sestavy čistě výstupní. Například jako ovládání svítících segmentových zobrazovačů, nebo jako ovladač souboru pohonů, či jiných akčních členů.⁴

Podle řešené úlohy se vybírá potřebná cílová konfigurace PLC. Na některé úlohy stačí „odlehčené“ automaty, na jiné úlohy je naopak zapotřebí použití plnohodnotných variabilních systémů. Podle rozsáhlosti a variabilnosti systémů se tyto automaty dělí do několika kategorií - mikro PLC, kompaktní PLC, modulární PLC a soft PLC.

1.2.1 Mikro PLC

Nejmenší a nejlevnější PLC systémy, které nabízejí uživateli pevnou sestavu vstupů a výstupů (obvykle jen binárních), například 6 binárních vstupů a 6 binárních výstupů. Uživatel se může tak rozhodnout pro jeden typ systému a dále ho již nelze rozšiřovat. Svým provedením, malými rozměry a nízkou cenou se mikro PLC zařazuje do skupiny „spotřebního materiálu“. Možnosti programování a funkční možnosti jsou obvykle u mikro PLC redukovány na minimum a komunikační možnosti mnohdy chybí.

Využití těchto automatů je v realizaci logické výbavy jednoduchých strojů a mechanismů, která byla dříve řešena pomocí různých ovládacích prvků, relé, stykačů atd. Dnes vyjde jednodušeji a levněji použití mikro PLC namísto pevné reléové logiky.⁵

1.2.2 Kompaktní PLC

Kompaktní systém obsahuje základní modul, který je možný rozšířit o několik dalších přídatných modulů. O binární (reléové nebo tranzistorové) a analogové vstupní/výstupní moduly, moduly pro základní podporu

⁴ ŠMEJKAL, L. a MARTINÁSKOVÁ, M. 2007. PLC a automatizace, s. 40.

⁵ ŠMEJKAL, L. a MARTINÁSKOVÁ, M. 2007. PLC a automatizace, s. 40.

komunikace, v některých případech i zdrojový modul. Rozšiřitelnost kompaktních PLC systémů je omezena.⁶

1.2.3 Modulární PLC

V modulárním systému jsou jednotlivé komponenty celku rozděleny do modulů. Celý systém PLC se potom skládá z modulů: zdroje, CPU, vstupů/výstupů, funkčních modulů. Modulární systém je možno dále rozšiřovat (s ohledem na limity výstavby systému) a to v nepoměrně větším rozsahu než u kompaktních systémů.⁷

Moduly mohou být binární, analogové, polohovací, čítače, fuzzy, počítačové a speciální.

1.2.4 Soft PLC

Jedná se o PLC integrovaná do průmyslových počítačů (IPC). Používají se buď PLC koprocesory a speciální Real-Time OS (operační systémy reálného času), nebo samostatné počítačové moduly, sériově komunikující s počítačem. Centrální počítač realizuje funkce PLC a funkce vývojového systému.

Výhodou těchto automatů je možnost soustředění funkce a vývojové prostředí spolu s operátorským rozhraním, s vizualizací a archivací technologických dat, s optimalizačními, expertními a diagnostickými funkcemi.

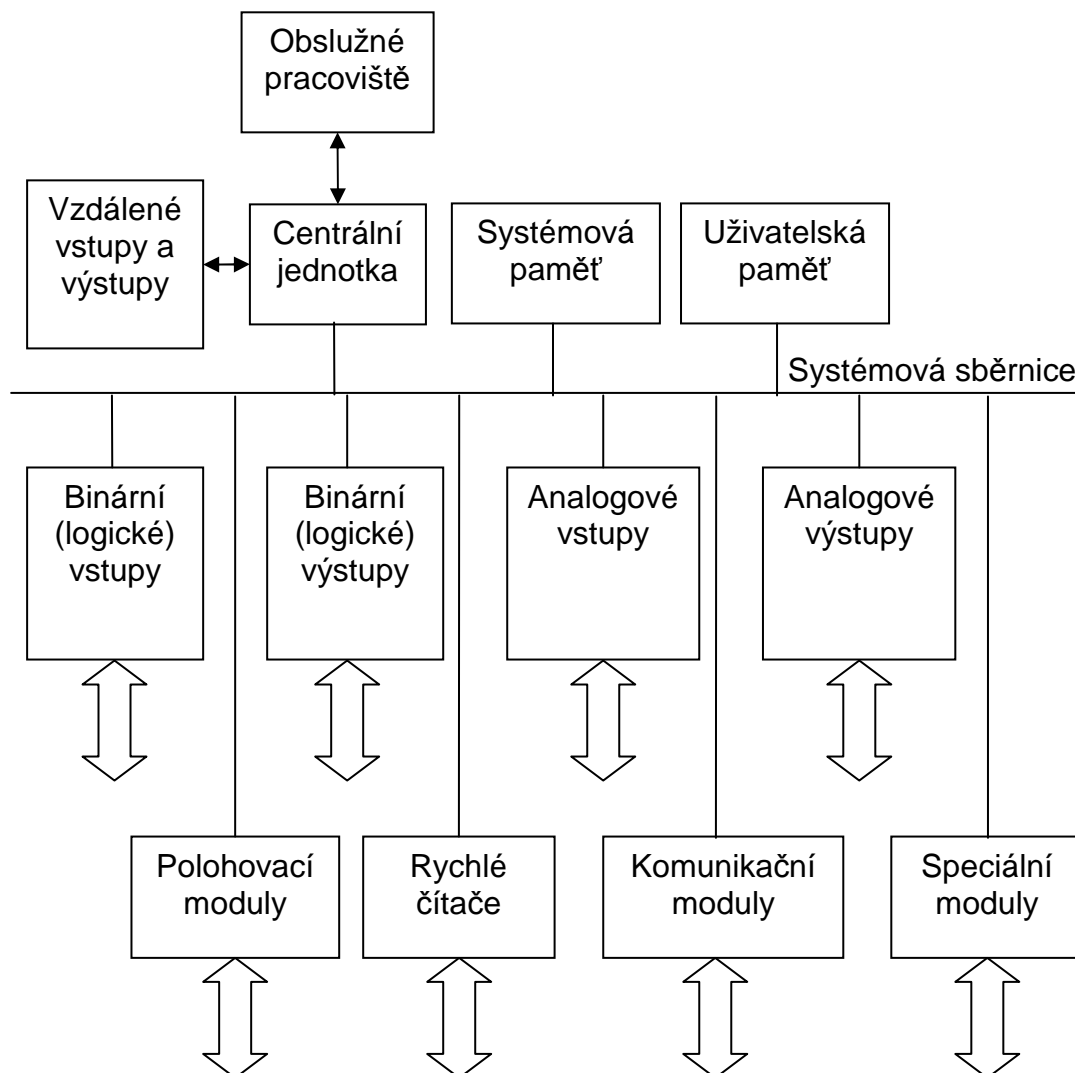
Nevýhodou tohoto řešení je vysoká cena, centralizace funkcí do jednoho počítače a z toho plynoucí zranitelnost. Další nevýhodou je také neexistující spolehlivý a levný Real-Time operační systém.⁸

⁶ WIKIPEDIA. 2006. *Programovatelný logický automat* [online].

⁷ WIKIPEDIA. 2006. *Programovatelný logický automat* [online].

⁸ VONDRA, Z. 2006. *Základy programování PLC*, s. 4-5 [online].

1.3 Hardware



Obrázek 1 - Vnitřní struktury PLC

Schéma vnitřní struktury programovatelných logických automatů je znázorněno výše. Modulární automaty mají variabilní počty vstupních a výstupních modulů a dalších zařízení. Jednotlivé funkční bloky jsou propojeny pomocí jedné nebo dvou sběrnic. Běžně osazované modulové jednotky v PLC jsou: centrální procesorová jednotka, systémová paměť, uživatelská paměť a další moduly pro analogové a binární vstupy či výstupy. Konečnou sestavu volí uživatel tak, aby programovatelný automat co nejlépe vyhovoval řešeným úlohám. Někdy může být PLC sestaven tak, aby měl jen čistě binární rozhraní a nebo čistě analogové rozhraní (některé kompaktní,

nebo mikro PLC zařízení). Protože byly programovatelné automaty původně navrženy pro realizaci logických úloh, nechybí v žádném PLC instrukce pro základní logické operace (logický součet, součin, negace, instrukce pro realizaci paměťových funkcí a klopných obvodů, pro zápis výsledku nebo mezivýsledku na adresované místo) ani pro aritmetiku či operace s čísly.⁹

Binární vstupy – jsou galvanicky oddělené, připojují se sem tlačítka, přepínače, koncové a jiné snímače s dvouhodnotovým charakterem signálu (rozsvíceno/zhasnuto, teplota vysoká/teplota nízká, ...).

Binární výstupy – jsou určeny pro buzení cívek relé, stykačů, elektromagnetických spojek, pneumatických a hydraulických převodníků, k ovládání signálů a k řízení dalších možných zařízení.

Analogové vstupy – lze je připojit na snímače teploty, tlaku, vlhkosti, hladiny a většinu dalších přístrojů a zařízení s analogovými výstupy.

Analogové výstupy – můžeme jimi ovládat servopohony, frekvenční měniče a jiné spojitě ovládané akční členy.

Komunikační moduly – slouží ke komunikaci mezi PLC zařízeními, nebo komunikaci s jinými řídicími systémy.

Rychlé čítače a polohovací moduly – pomocí těchto modulů lze vyhodnocovat rychlé sledy impulsů. Nacházejí uplatnění při řešení úloh geometrického charakteru. PLC tak může realizovat úlohy stejné jako systémy CNC.

Speciální moduly – moduly pro zálohování obsahu, rozšíření kapacity paměti pro uživatelský program a pro datové soubory, časové obvody pro informace o denním a kalendářním čase, svorkovnicové moduly, moduly rozbočovačů a komunikačních kabelů a další.

⁹ ŠVARC, I. 2005. Automatizace, s. 30.

Centrální Procesorová jednotka – tato jednotka vlastně ovládá celý programovatelný automat. Zajišťuje základní komunikační funkce s vlastními i vzdálenými moduly, s nadřazeným systémem a s programovacím přístrojem. Obsahuje mikroprocesor a řadič, zaměřený na rychlé provádění instrukcí. Procesorová jednotka také realizuje soubor instrukcí a základních systémových služeb.

Paměť – paměťové prostory jsou rozděleny na dvě části. První, systémová paměť, kde jsou uživatelské registry, čítače a časovače, komunikační, časové a jiné systémové proměnné. Druhá, uživatelská paměť, slouží pro uložení uživatelského (PLC) programu který se během vykonávání programu se nemění.¹⁰

1.4 Programování

K naprogramování logických automatů slouží programovací přístroje. Dříve byly řešeny jako specializované přístroje v kufříkách. Dnes se již k této činnosti používají počítače s programovým vybavením pro vývoj a ladění uživatelských programů PLC. Příruční programovací přístroje se dnes používají většinou už jen pro malé systémy.

1.4.1 Programovací jazyky

K naprogramování PLC je možné využít několik specializovaných jazyků, některé jsou intuitivní, jiné složitější. Jazyky systémů různých výrobců bývají podobné, ale ne stejné. Přímé přenesení programu na jiný systém je možný jen na podobných systémech stejného výrobce, mezi programovatelnými logickými automaty různých výrobců není možné přímo přenést program.

¹⁰ ŠMEJKAL, L. a MARTINÁSKOVÁ, M. 2007. PLC a automatizace, s. 42-44.

Mezinárodní norma IEC 61131-3 sjednocuje programovací jazyky pro PLC. Definuje čtyři typy jazyků: jazyk mnemokódů, jazyk kontaktních schémat, jazyk logických schémat a jazyk strukturovaného textu.

Jazyk mnemokódů – („Instruction List“, „IL“) je obdobou assembleru u počítačů a je také strojově orientován. Každé jedné instrukci PLC odpovídá jeden příkaz jazyka. Jazyky mnemokódů také dovolují používání skoků a používání návěstí pro volání, symbolická jména pro číselné hodnoty, pojmenování vstupů, výstupů, proměnných a jiných objektů programu, zadávání hodnot v různých číselných soustavách, inicializaci proměnných, a automatické přidělování paměti.

Jazyk kontaktních schémat – („Ladder Diagram“, „LD“) je programovací jazyk s grafickým uživatelským rozhraním. Cílový program se sestavuje z jednotlivých logických operací ve formě obvyklé pro kreslení schémat při práci s reléovými a kontaktními prvky. Symboly kontaktů a cívek jsou zde zjednodušeny, aby mohly být vytvářeny semigraficky (Spínací kontakty jako dvojice svislých čárek, rozpínací kontakt má navíc uprostřed lomítko, cívky jsou značeny jako dvojice závorek. Funkční bloky jako čítače a časovače jsou kresleny jako obdélníkové značky. Instrukce bez analogie v kontaktní symbolice se označují jako dvojice závorek nebo obdélníková značka s vepsaným mnemokódem instrukce.).

Tento jazyk je výhodný při programování jednodušších logických operací. Je rychlý a vhodný pro nalezení a odstranění závady.

Jazyk logických schémat – („Function Block Diagram“, „FBD“) je dalším grafickým programovacím jazykem je podobný LD (jazyku kontaktních schémat). Základní logické operace popisuje obdélníkovými značkami. Ucelené funkční bloky (čítače, časovače, registry, ...) mají také své značky. Tento programovací jazyk je dobrý pro uživatele zvyklé na kreslení logických schémat pro zařízení s integrovanými obvody.

Jazyk strukturovaného textu – („Structured Text“, „ST“) je podobný vyšším programovacím jazykům pro počítače (Pascal, Java, C). Umožňuje

zapisování algoritmů. Tento jazyk je dobrý pro řešení složitých logických operací, aritmetických instrukcí, skoků a volání, atd.

Nadstavbou nad těmito jazyky je grafický jazyk pro sekvenční programování (SFC, GRAFCET). K popisu struktury používá značky stavů, přechodů a větvení. Chování v jednotlivých stavech a definování podmínek přechodů jde popsat prostředky jakéhokoliv z výše pospaných programovacích jazyků, nebo dalším vnořeným sekvenčním grafem.¹¹

1.4.2 Vykonávání programu

Uživatelský program zpracovává a vykonává centrální jednotka. Ta poskytuje programovatelnému logickému automatu „inteligenci“. Realizuje soubor instrukcí a systémových služeb, zajišťuje základní komunikační funkce s moduly, nadřazeným systémem a programovacím přístrojem.

V souboru instrukcí a v systémových službách jsou uloženy základní operace, pomocí kterých je díky centrální jednotce realizován a zpracováván program.

Výpočetní výkon PLC je vysoký a závisí na použité centrální jednotce a na použitém souboru instrukcí a systémových služeb. Doba na provedení jednoduchých instrukcí se pohybuje v řádu mikrosekund, u složitějších operací je tato doba několikanásobně delší.¹²

1.4.3 Způsob vykonávání programu PLC

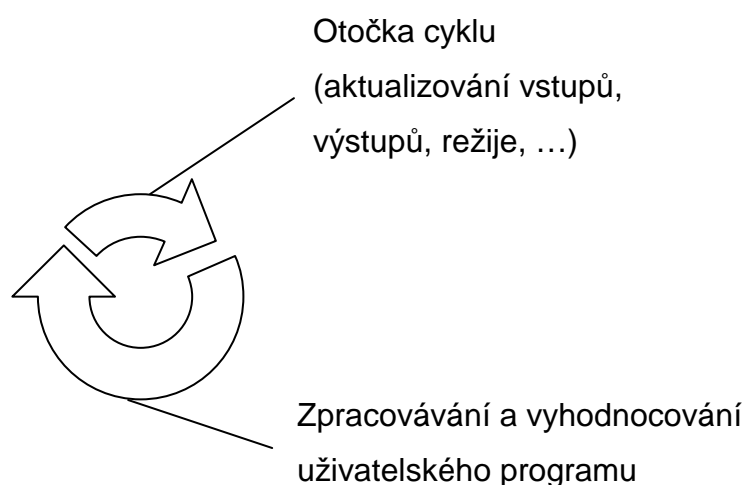
Program je posloupnost instrukcí a příkazů jazyka. U PLC je program vykonáván v cyklu (programové smyčce). Uživatel se nemusí starat o vracení na začátek programu. Každé dlouhodobé setrvání programu ve smyčce je fatální chybou a systém jej hlásí jako překročení doby cyklu.

¹¹ ŠMEJKAL, L. a MARTINÁSKOVÁ, M. 2007. PLC a automatizace, s. 49-51.

¹² ŠMEJKAL, L. a MARTINÁSKOVÁ, M. 2007. PLC a automatizace, s. 45.

Vždy po vykonání poslední instrukce uživatelského programu je řízení předáno systémovému programu, který aktualizuje hodnoty vstupů, aktualizuje hodnoty výstupů, aktualizuje časové údaje pro časovače a systémové registry, ošetří komunikaci a provede další režijní úkony. Toto je nazýváno otočkou cyklu. Po otočce cyklu začíná řízení opět první instrukcí uživatelského programu.

PLC nepracuje s aktuálními hodnotami vstupů a výstupů, ale s jejich obrazy uloženými v registrech. Aktualizace těchto hodnot je prováděna v otočce cyklu, tím je omezena možnost chyb způsobených nevhodnou změnou hodnot a zajištěna synchronizace vstupních a výstupních dat.¹³



Obrázek 2 - Cyklus programu PLC

¹³ ŠMEJKAL, L. a MARTINÁSKOVÁ, M. 2007. PLC a automatizace, s. 46-57.

2 Teorie logického řízení

Pojem řízení si můžeme vysvětlit jako působení dostupnými prvky na nějaký předmět (osobu, zvíře nebo jakoukoliv věc) takovým způsobem, abychom dosáhli požadovaného výsledku. V automatizaci je podle Ivana Švarce tento pojem definován jako „**cílené působení na řízený objekt tak, aby se dosáhlo určitého předepsaného cíle**“.¹⁴

Teorie řízení se v praxi zabývá zkoumáním a popisováním všech vlastností řízení a chováním řízených objektů. V automatizaci se tato teorie zabývá zkoumáním řízení systémů a jejich podsystémů.

Teorie logického řízení se tedy zabývá řízením systémů pomocí logických obvodů. Tyto obvody pracující s dvouhodnotovými stavy lze je popsat pomocí logických prvků a logických veličin. Na rozdíl od analogových obvodů, kde se pracuje s kontinuálními (spojitými) informacemi, v logických obvodech se pracuje s digitálními (nespojitými) informacemi, které jsou nejčastěji prezentovány jako logická 0 (stav vypnuto) a logická 1 (stav zapnuto). V technice si tyto stavy můžeme představit například jako: nesvítí, zavřeno, motor se netočí (logická 0), nebo jako svítí, otevřeno, motor se točí (logická 1).

2.1 Logické funkce

Funkce využívané v logickém řízení se nazývají logickými funkcemi. Proměnné i samotné funkce nabývají dvou hodnot, jsou to výše zmíněné hodnoty 1 a 0.

Logickou funkci lze zadat různými způsoby, slovně – kdy text popisuje souvislost vstupních a výstupních (logických) veličin, pravdivostní tabulkou – kam se zapisují všechny možné kombinace hodnot vstupních k hodnotám

¹⁴ ŠVARC, I. 2005. Automatizace, s. 8.

výstupním, Karnaughovou mapou a nebo algebraickým výrazem obsahujícím operátory základních logických funkcí.

Matematický zápis logické funkce, kde výsledek funkce y je závislý na proměnných x_1 a x_2 vypadá například takto:

$$y = f(x_1, x_2) \quad (1)$$

Nejjednoduššími příklady jsou logické funkce s jednou proměnnou, které si vysvětlíme na pravdivostní tabulce. Pro jednu proměnnou x existují čtyři různé funkce y . Falsum, kde výsledkem funkce y je pro jakoukoliv hodnotu proměnné x hodnota 0. Verum, která je opakem funkce falsum a výsledkem této funkce je tedy vždy hodnota 1. Třetí funkcí je aserce. Zde hodnota funkce y nabývá stejných hodnot jako proměnná x . A poslední, v praxi jedinou využívanou funkcí z této čtveřice, je negace. Při negaci je výsledkem funkce opačná hodnota proměnné. Tedy pokud proměnná x nabývá hodnoty 0, potom výsledkem y bude 1 a naopak. Jestliže bude proměnná x rovna 1, tak funkce y bude 0.

Tabulka 1 - Logické funkce jedné proměnné

X	y	x	y	x	y	x	Y
0	0	0	1	0	0	0	1
1	0	1	1	1	1	1	0
falsum		verum		aserce		negace	

Logické funkce dvou proměnných jsou o něco málo složitější než logické funkce jedné proměnné. Je jich celkem 16, ale v praxi se nejvíce používají čtyři, které jsou popsány níže v tabulce – konjunkce (logický součin), disjunkce (logický součet), negovaná konjunkce (negovaný logický součin – NAND) a negovaná disjunkce (negovaný logický součet – NOR).

Tabulka 2 - Logické funkce dvou proměnných

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

AND

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

OR

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

NAND

x_1	x_2	Y
0	0	1
0	1	0
1	0	0
1	1	0

NOR

Logický součin (AND, konjunkce) je charakterizována tím, že funkční hodnota y nabývá jedničky pouze tehdy, když obě proměnné x_1, x_2 (obecně všechny proměnné) jsou jedničky. Logický operátor této funkce je „ \cdot “, někdy se používá znak „ \wedge “.

Logický součet (OR, disjunkce) je charakterizována tím, že funkční hodnota y nabývá jedničky tehdy, když alespoň jedna z proměnných x_1, x_2 (obecně ze všech proměnných) je jednička. Logický operátor této funkce je „ $+$ “, často se také používá znak „ \vee “.

Negace logického součtu (NOR, negace disjunkce – někdy též Piercova funkce) je charakterizována tím, že funkční hodnota y je jednička, když žádná z proměnných x_1, x_2 (obecně když žádná z proměnných) není jednička.

Negace logického součinu (NAND, negace konjunkce – někdy též Shefferova funkce) je charakterizována tím, že funkční hodnota y nabývá jedničky tehdy, když proměnné x_1, x_2 (obecně všechny proměnné) nejsou současné jedničky.¹⁵

Všechny logické funkce lze vyjádřit Booleovskými funkcemi (disjunkce, konjunkce, negace), nebo pomocí funkce NAND, či pomocí funkce NOR.

¹⁵ ŠVARC, I. 2005. Automatizace, s. 13.

2.2 Booleova algebra

„Booleova algebra pracuje s dvouhodnotovými logickými veličinami a používá tři základních logických funkcí: negace, konjunkce a disjunkce...“¹⁶

Pravidla Booleovy algebry se používají pro minimalizaci (zmenšení a zjednodušení) logických funkcí. Po této minimalizaci je například možné efektivně sestavit logický obvod tak, aby jsme k tomuto sestavení využili co nejméně součástí. K minimalizaci logických funkcí slouží tato pravidla (zákony):

- zákon vyloučení třetího

$$x \vee \bar{x} = 1 \quad (2)$$

- zákon logického rozporu

$$x \wedge \bar{x} = 0 \quad (3)$$

- zákon dvojité negace

$$\overline{\overline{x}} = x \quad (4)$$

- zákon opakování

$$x \vee x = x \quad x \wedge x = x \quad (5)$$

- komutativní zákon

$$x_1 \vee x_2 = x_2 \vee x_1 \quad x_1 \wedge x_2 = x_2 \wedge x_1 \quad (6)$$

- asociativní zákon

$$x_1 \vee (x_2 \vee x_3) = x_1 \vee x_2 \vee x_3 \quad x_1 \wedge (x_2 \wedge x_3) = x_1 \wedge x_2 \wedge x_3 \quad (7)$$

¹⁶ ŠVARC, I., LACKO, B., NĚMEC, Z. c1995. Automatizace, s. 11.

- distributivní zákon

$$\begin{aligned}x_1 \wedge (x_2 \vee x_3) &= x_1 \wedge x_2 \vee x_1 \wedge x_3 \\x_1 \vee x_2 \wedge x_3 &= (x_1 \vee x_2) \wedge (x_1 \vee x_3)\end{aligned}\quad (8)$$

- absorpční zákon

$$\begin{aligned}x_1 \vee x_1 \wedge x_2 &= x_1 & x_1 \wedge (x_1 \vee x_2) &= x_1 \\x_1 \vee \bar{x}_1 \wedge x_2 &= x_1 \vee x_2 & x_1 \wedge (\bar{x}_1 \vee x_2) &= x_1 \wedge x_2\end{aligned}\quad (9)$$

- zákon neutrálnosti

$$0 \vee x = x \quad 1 \wedge x = x \quad (10)$$

- zákon agresivity

$$1 \vee x = 1 \quad 0 \wedge x = 0 \quad (11)$$

- De Morganovy zákony

$$\overline{x_1 \vee x_2} = \bar{x}_1 \wedge \bar{x}_2 \quad \overline{x_1 \wedge x_2} = \bar{x}_1 \vee \bar{x}_2 \quad (12)$$

De Morganovy zákony jsou důležité, protože se uplatňují v úpravě algebraických výrazů a také v dalším převádění Booleovy algebry na NAND a NOR algebru.

Pro minimalizaci se používají pravidla Booleovy algebry a výrazy se upravují tak dlouho, dokud nedostaneme nejkratší výraz. Toto je však pracná metoda, obzvláště pokud se upravují složité výrazy. Navíc pomocí této metody nelze zjistit, zda jsme dostali nejmenší možný tvar výrazu. Z toho důvodu se minimalizace neprovádí tímto způsobem. Místo upravování algebraických výrazů se používají Karnaughovy mapy.¹⁷

¹⁷ ŠVARC, I. 2005. Automatizace, s. 15-16.

2.3 Pravdivostní tabulka

Základní formou popisu logické funkce je popis pravdivostní tabulkou. Do tabulky se zapisují všechny možné kombinace hodnot pro které je funkce definována.¹⁸ Přiřazují se zde kombinacím vstupních proměnných hodnoty logické funkce. Počet kombinací (řádků tabulky) příslušných k počtu vstupních proměnných je vždy:

$$N = 2^k \quad (13)$$

kde k je počet vstupních proměnných a N je výsledný počet kombinací.

Například pro tři vstupní proměnné (a , b , c) existuje osm kombinací hodnot pro výstupní proměnnou y .

Tabulka 3 - Pravdivostní tabulka

Stav	Vstup a	Vstup b	Vstup c	Výstup y
1	0	0	0	0
2	0	0	1	1
3	0	1	0	1
4	0	1	1	0
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

První sloupec tabulky (Stav), udává pořadí a počet jednotlivých kombinací. Hodnoty logické funkce doplňuje návrhář logického obvodu nebo automatu a řídí se jeho požadovanou funkcí. Další tři sloupce (Vstup a , Vstup b , Vstup c) tvoří všechny možné kombinace funkčních hodnot, které pro dané proměnné mohou nastat. A v posledním sloupci (Výstup y) je zapsáno pro

¹⁸ ŠVARC, I. 2005. Automatizace, s. 17.

kteře kombinace je vřsledek pravdivř, tedy nabřvř hodnot logické 1 a kde je nepravdivř, tedy nabřvř hodnot logické 0.

Funkční hodnoty nemusř břt definovřny pro vřechny kombinace vstupnřch proměnnřch. Z definice mohou břt odebrřny kombinace, kteře nemohou nastat.¹⁹

2.4 Karnaughova mapa

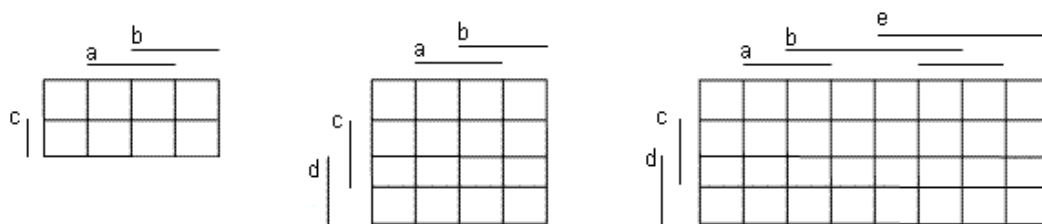
Karnaughova mapa je vlastnř takř tabulka (K-mapa), jejřz pole jsou řuřelne uspořřdřna tak, aby bylo mořnř jednoduře optimalizovat zřpis logické funkce. Kombinaci hodnot proměnnřch urřuje poloha pole a přřsluřnř hodnota logické funkce je vepsřna na plochu tohoto pole.²⁰

Tato tabulka mř tolik polřcek, kolik je kombinaci proměnnřch vyřetřovaně funkce. Funkci s n proměnnřmi vyjadřujeme mapou s n^2 polřcky. Kařdř polřcko odpovřdř jednř z mořnřch kombinaci a zapisujeme do nřj odpovřdřajřcř funkční hodnotu. V Karnaughovř mapř se sousednř polřcka liřř hodnotou jedině proměnně a přřsluřnř proměnně rovně jednř, se oznařujř vedle mapy svislou nebo vodorovnou řarou ke kteře se přřpisuje jměno přřsluřnř proměnně. V řřdcřch nebo sloupcřch, kteře nejsou takto znařeny je přřsluřnř proměnnř rovna nule (Dř se takř řřci, ře v mřstech, kteřa nejsou oznařenř řarou, je hodnota daně proměnně negovanř.). Zapsřnř funkce do mapy spořřvř v přřpsřnř funkčních hodnot do přřsluřnřch polřcek. Vychřzet mřžeme buď z pravdivostnř tabulky, nebo z algebraickěho vřrazu v řplnř normřlnř disjunktivnř formě.²¹

¹⁹ TŮMA, Jiřř. c2007. Logické řřzenř, s. 2-3 [online].

²⁰ TŮMA, Jiřř. c2007. Logické řřzenř, s. 2-3 [online].

²¹ řVARC, I. 2005. Automatizace, s. 18-19.



Obrázek 3 - Karnaughova mapa

2.5 Minimalizování logických funkcí

Pro sestavení logického obvodu, který je popsán logickou funkcí, je nutné, z ekonomického a technického hlediska, nejprve zjednodušit (minimalizovat) logickou funkci, jenž daný logický obvod popisuje. Po minimalizaci takovéto funkce, bude pro sestavení logického obvodu zapotřebí minimální počet logických prvků. Tím se daný logický obvod zjednoduší a také zlevní. V některých případech se také tento obvod stane spolehlivějším.

První metodou minimalizace je algebraická minimalizace, kde se logická funkce zjednodušuje pomocí aplikace různých pravidel Booleovy algebry až do nejmenšího tvaru. Tato metoda má ale tři nevýhody. Jednou z těchto nevýhod je pracnost, u složitějších příkladů také časová náročnost a složitost. Další nevýhodou je fakt, že nevíme zda-li je náš výsledek opravdu nejmenším možným výrazem, který jde z dané funkce dostat. Poslední nevýhodou algebraické minimalizace je také větší možnost chybování ve výpočtu.

Druhou metodou minimalizace je použití Karnaughovy mapy. Tato metoda je velmi výhodná, protože je rychlá, spolehlivá a navíc (pokud pracujeme správně) vždy víme, že jsme dosáhli nejmenšího možného výrazu logické funkce.

Minimalizace pomocí Karnaughovy mapy spočívá v nalezení minimálního algebraického tvaru z mapy. Postupem při hledání takovéto funkce je hledání sousedních políček s jedničkou tak, aby tvořili co největší

možné skupiny, které jde ohraničit čtvercem či obdélníkem. V Karnaughově mapě se považují za sousední políčka také políčka, která jsou na opačných koncích mapy. Hledají se dvojice, čtveřice, osmice, šestnáctice atd.

„Základní pravidla pro minimalizaci logických funkcí Karnaughovými mapami – jak provést seskupení jedniček v mapě do izolovaných jedniček, dvojic, čtveřic, ...

- **Všechny jedničky v mapě musí být zakroužkovány, žádnou nesmíme vynechat**
- **Každá jednička se může při kroužkování vzít několikrát, může být současně součástí dvojice, čtveřice, ... (to umožňuje zákon opakování $x + x + x + \dots = x$)**
- **Přednost mají ... osmice před čtveřicemi, čtveřice před dvojicemi a dvojice před izolovanými jedničkami**
- **V rámci pravidla podle kterého žádnou jedničku nesmíme vynechat, se snažíme o co nejmenší počet smyček²²**

Ukázka postupu při minimalizaci pomocí Karnaughovy mapy:

Z následujícího algebraického tvaru vytvoříme Karnaughovu mapu. Pro jednoduchost a přehlednost je v tomto tvaru znaménko „^“ vynecháno a proměnné jsou spojeny a znaménko „v“ nahrazeno znaménkem „+“.

$$y = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + ab\bar{c}\bar{d} + ab\bar{c}d + abcd + \bar{a}\bar{b}cd \quad (14)$$

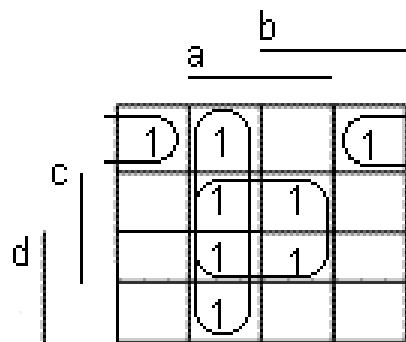
²² ŠVARC, I. 2005. Automatizace, s. 21.

Tento algebraický tvar by se dal také popsat pravdivostní tabulkou:

Tabulka 4 - Algebraická funkce vyjádřená pomocí pravdivostní tabulky

stav	y	d	c	b	a	stav	y	d	c	b	a
1	1	0	0	0	0	9	0	1	0	0	0
2	1	0	0	0	1	10	1	1	0	0	1
3	1	0	0	1	0	11	0	1	0	1	0
4	0	0	0	1	1	12	0	1	0	1	1
5	0	0	1	0	0	13	0	1	1	0	0
6	1	0	1	0	1	14	1	1	1	0	1
7	0	0	1	1	0	15	0	1	1	1	0
8	1	0	1	1	1	16	1	1	1	1	1

Z algebraického výrazu, nebo pravdivostní tabulky dále vytvoříme Karnaughovu mapu, kterou dále využijeme při minimalizaci a opět z ní získáme algebraický výraz, který již bude v minimálním tvaru.



Obrázek 4 - Minimalizace pomocí Karnaughovy mapy

Při spojení dvou jedniček, jedna umístěná v prvním políčku prvního řádku a druhá umístěná ve čtvrtém posledním prvního řádku (první nalezená dvojice), vznikne nám výraz:

$$y = \bar{a}\bar{c}\bar{d} \quad (15)$$

Dále můžeme podle obrázku spojit čtyři prostřední jedničky, které nám dají při převedení do algebraického tvaru následující rovnici:

$$y = ac \quad (16)$$

Poslední zbývají dvě jedničky. Jedna nahoře (první řádek, druhý sloupec) a jedna dole (poslední řádek, druhý sloupec). Pokud tyto jedničky spojíme s již vyjádřenými jedničkami uprostřed, dostaneme menší tvar. Takto vypadá výsledný výraz vertikálního spojení druhého sloupce.

$$y = a\bar{b} \quad (17)$$

Pokud nyní spojíme všechny jednotlivé výrazy, které jsme našli, vznikne nám minimální logická funkce, která má stejný výsledek jako logická funkce, která byla uvedena na začátku. Správnost výsledku si můžeme ověřit na pravdivostní tabulce.

$$y = \bar{a}\bar{c}\bar{d} + a\bar{b} + ac \quad (18)$$

2.6 Kombinační logické obvody

Kombinační logické funkce můžeme realizovat pomocí následujících technických prvků a technických prostředků:

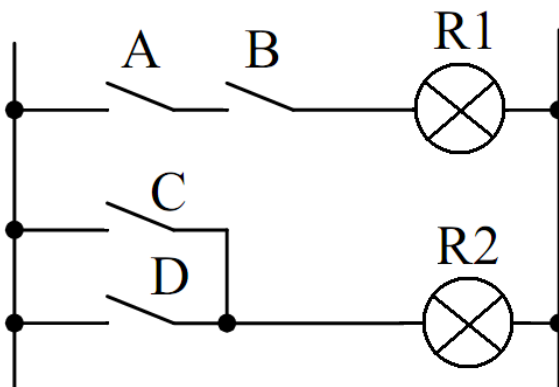
- mechanická relé,
- integrované obvody,
- programovatelné logické automaty.

Každý z těchto prvků má specifický způsob kreslení schématu. U mechanických relé se používá liniových schémat, u integrovaných obvodů se využívá blokových schémat a u PLC (programovatelných logických automatů) je způsob kreslení schémat popsán dokumentací softwaru k danému automatu.

2.6.1 Liniové schéma

„Toto schéma znázorňuje zapojení elektrických kontaktů, které umožňují průchod elektrického proudu mezi dvěma póly zdroje obvykle kreslenými jako svislé vodiče. Mezi těmito vodiči prochází proud přes různě zapojené kontakty do zátěže. Na pravé straně je kladné napájecí napětí a levá strana je obvykle uzemněna. Zátěží může být cívka elektromagnetického relé nebo jiný spotřebič, například žárovka nebo houkačka. Kontakty relé dimenzované na potřebné proudové zatížení mohou být napojeny i na elektrický motor nebo soustavu osvětlení.“²³

Základní liniová schémata:



Obrázek 5 - Liniové schéma

$$\begin{aligned} R_1 &= A \wedge B \\ R_2 &= C \vee D \end{aligned} \tag{19}$$

V obrázku je znázorněno, že pokud bude stisknut spínač *a* i *b* zároveň, dojde k rozsvícení žárovky označené *R1*. Druhá žárovka *R2* se dá rozsvítit buď stisknutím tlačítka *c*, nebo stisknutím tlačítka *d*, či stisknutím obou tlačítek zároveň.

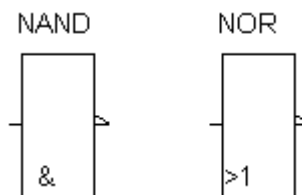
²³ TŮMA, Jiří. c2007. Logické řízení, s. 8 [online].

2.6.2 Bloková schémata

„Tento typ schématu odpovídá zapojení elektronických součástek plnících logické funkce. Tyto součástky se nazývají hradla. Jsou především typu NAND a NOR. Signál logické jedničky v tzv. pozitivní logice TTL (Tranzistor-Tranzistor-Logic) je kladné elektrické napětí +5V, naproti tomu logické nule odpovídá 0V. K označení úrovně napětí pro logickou jedničku se používá H (High) a L (Low) pro logickou nulu. Samozřejmě, že je přípustná tolerance velikosti napětí.“²⁴

Výhodou NAND a NOR logiky je, že k sestavení celého obvodu popsaného logickou funkcí stačí buď jen prvky typu NAND (negovaný logický součin) a nebo prvky typu NOR (negovaný logický součet).

Na následujícím obrázku je prvek typu NAND (vlevo) a prvek typu NOR (vpravo):



Obrázek 6 - Prvky NAND a NOR

Proto, abychom sestavili obvod, musíme nejprve minimalizovat celou logickou funkci – nejlépe pomocí Karnaughovy mapy. Vezmeme například již logickou funkci, kterou jsme minimalizovali v kapitole 3.5.:

$$y = \overline{a}c\overline{d} + a\overline{b} + ac \quad (20)$$

Poté funkci musíme upravit pomocí De Morganových zákonů buď tak, abychom výsledek měli v logice NAND:

$$y = \overline{\overline{\overline{a}c\overline{d} + a\overline{b} + ac}} = \overline{\overline{a}c\overline{d} \wedge a\overline{b} \wedge ac} \quad (21)$$

²⁴ TŮMA, Jiří. c2007. Logické řízení, s. 9 [online].

Nebo vzorec upravíme tak, aby nám výsledek vyšel v logice NOR:

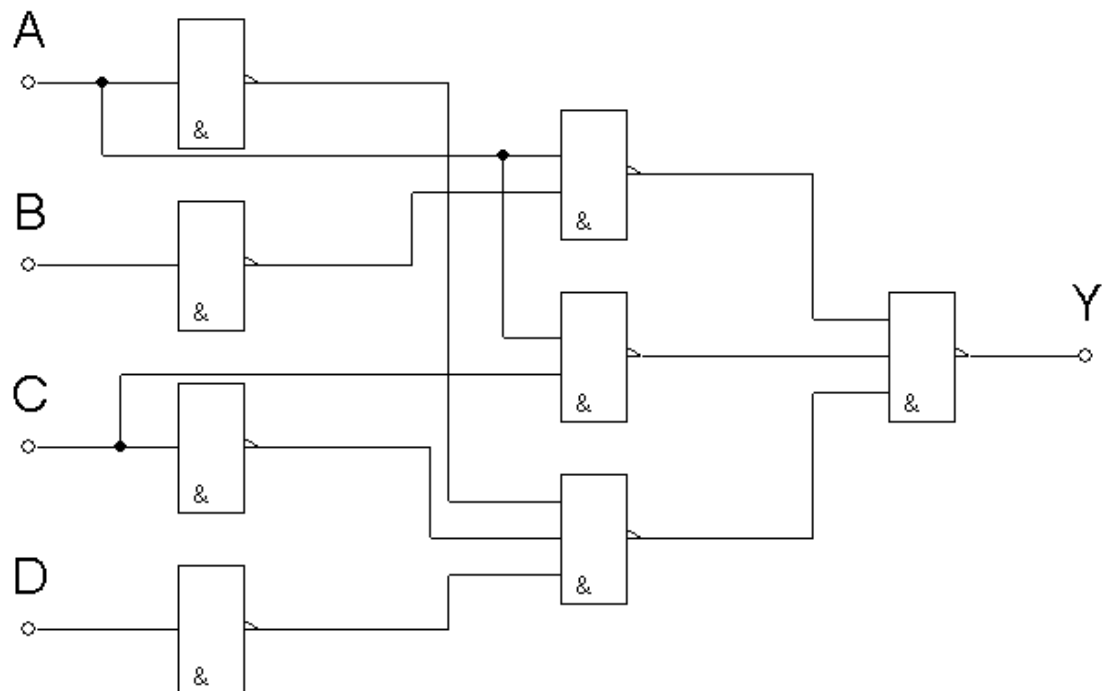
$$y = \overline{\overline{\overline{a\bar{c}\bar{d}} + \overline{a\bar{b}} + \overline{ac}}} = \overline{a + c + d + \bar{a} + b + \bar{a} + \bar{c}} \quad (22)$$

V logice NOR nemůžeme jednoduše sečíst všechny tři negované celky, proto je nutné celý výraz ještě dvakrát znegovat:

$$\bar{y} = \overline{\overline{\overline{a + c + d + \bar{a} + b + \bar{a} + \bar{c}}}} \Rightarrow y = \overline{\overline{\overline{\overline{a + c + d + \bar{a} + b + \bar{a} + \bar{c}}}}} \quad (23)$$

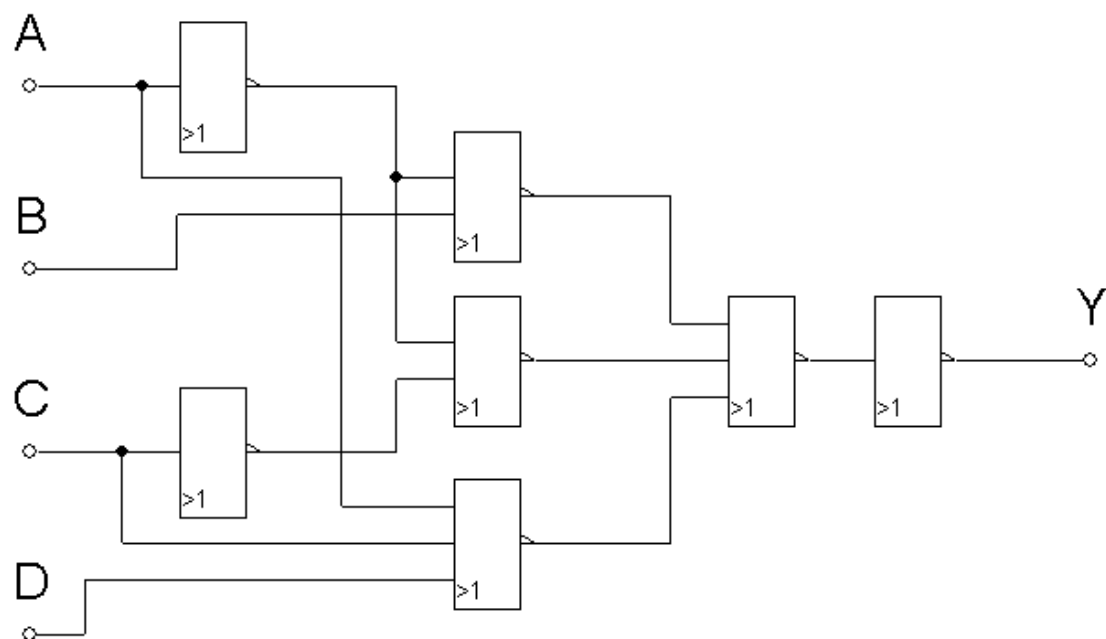
Nyní můžeme nakreslit daná bloková schémata.

Takto bude vypadat schéma vytvořené pomocí funkce NAND:



Obrázek 7 - Schéma pomocí NAND

Takto bude vypadat schéma vytvořené pomocí funkce NOR:



Obrázek 8 - Schéma pomocí NOR

3 Popis laboratorního modelu

K vytvoření laboratorní úlohy automatické pračky bylo použito následující vybavení: PLC Allen Bradley SLC 500 s komunikačním rozhraním 1747-PIC, osobní počítač, model automatické pračky od firmy Kohout a šest spínacích tlačítek.

3.1 Allen Bradley SLC 500

Allen Bradley SLC 500 je malý, výkonný, modulární a spolehlivý řídicí systém. Tento programovatelný logický automat je možné vybavit velkým množstvím modulů procesorů, napájecích zdrojů a nespočtem dalších vstupních nebo výstupních modulů. Pomocí vysoké variability je možné si sestavit Allen Bradley SLC 500 přesně na míru řešeného problému.

U série SLC 500 může být použito pět typů procesorů. Tyto jsou značeny jako SLC 5/01, SLC 5/02 až SLC 5/05. Zvolený typ procesoru určuje instrukční sady, počet vstupů a výstupů, velikost paměti a komunikační porty pomocí kterých je možné komunikovat s jinými systémy. Všechny tyto procesory mají port DH-485, který umožňuje naprogramování a základní komunikaci s PLC.

Jednotlivé moduly se zasouvají do speciálního šasi, které je označováno jako platforma 1746 nebo 1747. Do této platformy se také zasouvá modul procesoru, který se dává na první levou pozici. Šasi 1746 je možné osadit (podle typu velikosti šasi) čtyřmi, sedmi, deseti nebo třinácti zásuvnými moduly. Šasi je dále možné propojit a tím vytvořit automat do maximálního možného osazeného počtu 30 modulů.

3.1.1 Použité moduly

Použitý PLC Allen Bradley SLC 500 pro laboratorní úlohu se skládal z těchto částí: platforma 1746-A7, zdroj, 5/02 procesor 1747-L524, vstup

1746-IV16, vstup 1746-IB16, výstup 1746-OV16, výstup 1746-OW16, čítač 1746-HSCE, vstup/výstup 1746-NI04I.

Platforma 1746-A7 – je základem pro SLC 500, má 7 slotů pro zásuvné moduly.

5/02 procesor 1747-L524 – je procesorem a hlavní řídicí jednotkou pomocí níž můžeme nakonfigurovat až 4096 vstupů a 4096 výstupů. Má 4K instrukční paměti a 19 přídavných instrukcí zahrnujících posílání zpráv pro peer to peer komunikaci. Doba provádění jednoho tisíce instrukcí je přibližně 4,8 ms a doba skenování vstupů a výstupů je přibližně 1,6 ms (Záleží na typech vstupů a výstupů a typech instrukcí). Pro komunikaci je použit port DH485.

1746-IV16 – je digitální vstup s 16 vstupy fungujícími jako zdroj s maximálním zpožděním 8 ms na sepnutí/rozepnutí. Rozsah napětí na vstupu je nastavitelný od 10 do 30 V stejnosměrného napětí.

1746-IB16 – je digitální vstup s 16 vstupy fungujícími jako zátěž s maximálním zpožděním 8 ms na sepnutí/rozepnutí. Rozsah napětí na vstupu je nastavitelný od 10 do 30 V stejnosměrného napětí.

1746-OV16 – je digitální výstup s 16 výstupy fungujícími jako zátěž. Rozsah napětí na výstupu je nastavitelný od 10 do 50 V stejnosměrného napětí.

1746-OW16 – je digitální reléový výstup s 16 výstupy fungujícími jako zdroj. Rozsah napětí na výstupu je nastavitelný od 5 do 265 V střídavého nebo od 5 do 125 V stejnosměrného napětí.

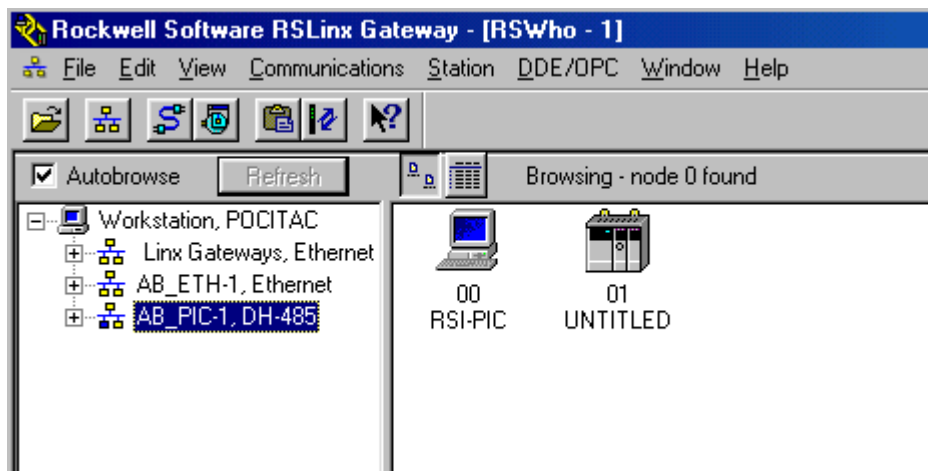
1746-HSCE – je speciální modul rychlého čítače. Používá se pro ovládání aplikací, kde je zapotřebí rychlého ovládání. Má 16-bitový čítač, který umí zachytit vstupní puls do frekvence 50 kHz.

1746-NI04I – je analogový modul se dvěma vstupy (± 10 V stejnosměrného napětí) a dvěma výstupy s šířkou pásma 10 Hz. Rychlost odezvy na vstupu je 60 ms a na výstupu 2,5 ms.

3.1.2 RS Linx

RS Linx je aplikace, která zajišťuje komunikaci mezi počítačem a PLC. Zajišťuje také sdílení a předávání dat pro aplikace, které využívají protokol DDE nebo OPC. Pro tuto laboratorní úlohu posloužil program RS Linx jako zprostředkovatel komunikace mezi programem RS Logix a PLC.

V této aplikaci nastavíme způsob komunikace s PLC. V našem případě je to komunikace přes rozhraní DH-485 pomocí konvertoru 1747-PIC. Po nastavení a potvrzení této komunikace se přidá název komunikačního rozhraní do levého okna (AB_PIC-1, DH-485) a v pravém okně se zobrazí nalezené zařízení PLC (01 UNTITLED).

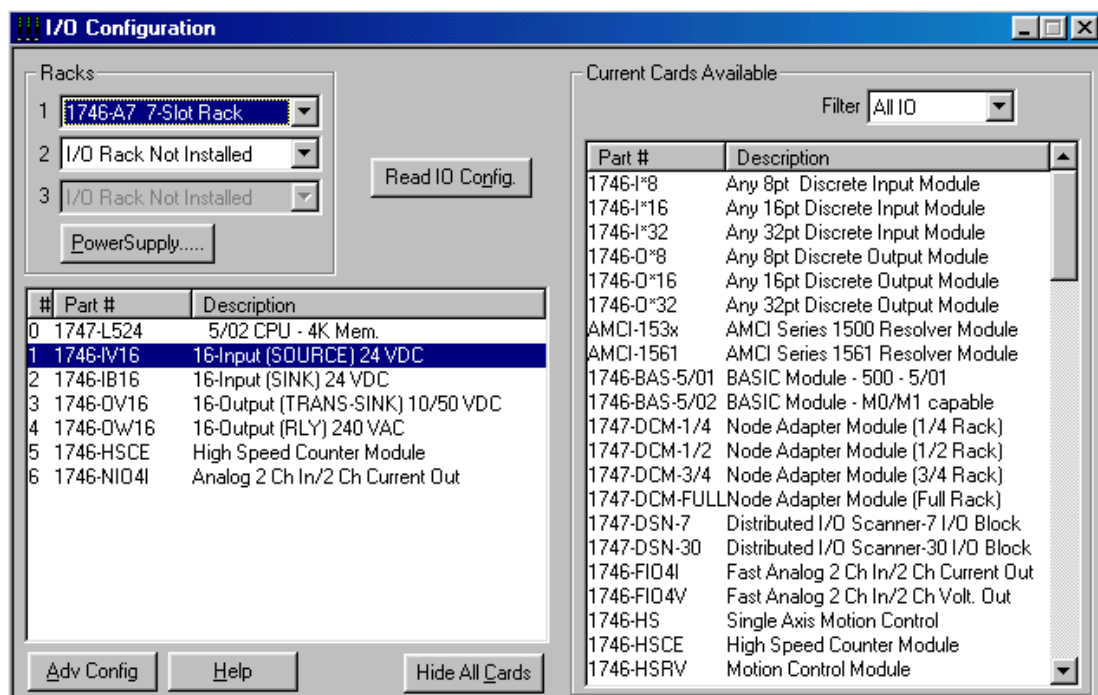


Obrázek 9 - Okno programu RS Linx

3.1.3 RS Logix

Tato aplikace od firmy Rockwell Software umožňuje programování procesorů řady SLC 500 nebo MikroLogix. Výhody této aplikace jsou hlavně v jednoduchém programování pomocí jazyku kontaktních schémat (LD), v rychlém vyhledávání a následném opravování chyb, v editaci pomocí DRAG and DROP a v možnosti sledovat chování PLC online.

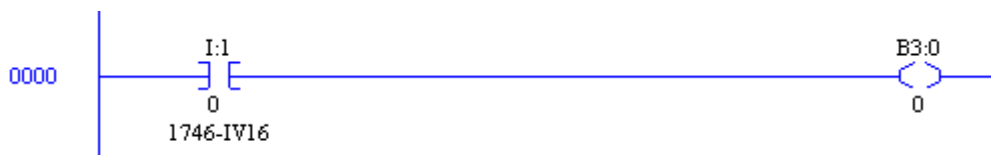
Pro komunikaci s PLC je nutné nejprve v programu RS Linx nastavit komunikaci s PLC přes 1747-PIC. Poté vytvořit nový soubor a nakonfigurovat všechny použité moduly včetně procesoru a použité platformy. Všechny tyto moduly vybíráme ze seznamu dostupných modulů (Current Cards Available) podle jejich označení. Jakmile jsou nastaveny moduly, můžeme začít s programováním.



Obrázek 10 - Konfigurace modulů v RS Logix

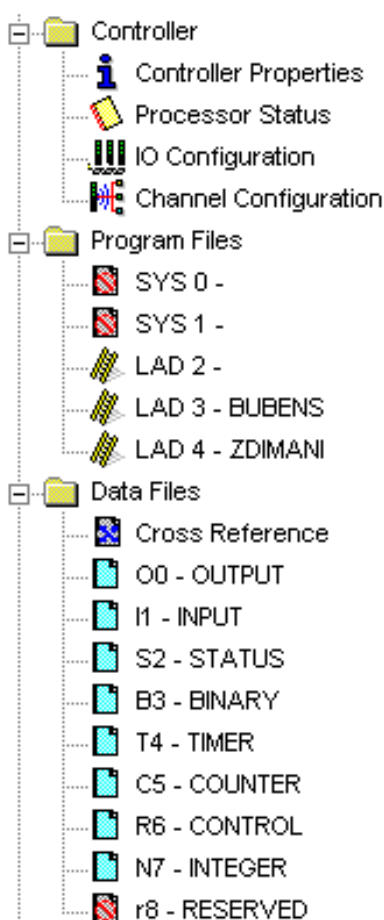
3.1.4 Programování automatu v RS Logix

V programu RS Logix se programuje pomocí kontaktních schémat. Celý program se skládá z jednotlivých příček (rung) a každá příčka obsahuje vstupní a výstupní instrukce. Vstupní instrukce jsou umístěny vlevo a výstupní instrukce vpravo. Pokud je splněna podmínka nalevo (v programu RS Logix je pravdivá hodnota označena zeleně), provede se výstupní instrukce. Prvky spojeny do série jsou vyhodnocovány jako logický součin, prvky spojeny paralelně jsou vyhodnocovány jako logický součet.



Obrázek 11 - Přička s jedním vstupem a jedním výstupem

Automat ke své funkci využívá tři typy souborů, ke kterým lze přistupovat v levém okně programu RS Logix:



Obrázek 12 – Levé okno RS Logix s ukázkou souborů

- **Procesorový soubor** – obsahuje programové i datové soubory. Obsahuje všechny instrukce, data a konfiguraci. Tento soubor je přenositelný mezi jednotlivými procesory.
- **Programové soubory** – obsahují řídicí informace, hlavní kontaktní schéma (LD) a podprogramy. Skládá se ze

systemového programu (soubor 0, který obsahuje informace o typu procesoru a jednotlivých modulech), rezervovaného programu (soubor 1), hlavního žebříčkového programu (soubor 2, který obsahuje kontaktní schéma a určuje chování programu) a případně taky z podprogramů (soubor 3-255).

- **Datové soubory** – obsahují data, která se vztahují k instrukcím v programových souborech. Projekt může mít až 256 datových souborů. Prvních devět datových souborů je přednastaveno. Jsou jimi výstupy (O), vstupy (I), stavové bity (S), bitový soubor (B), časovač (T), čítač (C), řídicí soubor (R) a celočíselný soubor (N). Ostatní si může nastavit uživatel. Při nastavování si uživatel může vybrat z následujících datových typů: bitový, celočíselný, čítač, časovač a řídicí. Tyto soubory, v závislosti na zvoleném datovém typu se ještě dále dělí na podstruktury, slova a bity.

3.2 Model pračky

Pro naprogramování a odzkoušení zadaných úloh byl použit elektronický model automatické pračky od firmy Kohout.

Pračka se napájí pomocí stejnosměrného napětí 24 V, které se přivádí na první dva vodiče 20 žilového kabelu. První vodič je označen červenou barvou. Šest signálů funguje jako vstupy, pomocí kterých se model pračky ovládá (napouštění vody, otáčení bubny, ...) a sedm signálů funguje jako výstupy z modelu (teplota a hladina), díky nimž máme přehled o stavu pračky. Některé vodiče nemají (podle konkrétního zařízení) žádnou funkci.

Vstupy do pračky fungují jako zátěž – je na ně nutné přivést napětí a výstupy z pračky fungují jako zdroj – dodávají napětí. Celá pračka funguje na úrovni 24 V logiky a stavy pračky jsou zobrazovány pomocí LED.

Po přivedení napájecího napětí do pračky (zapnutí), se nastaví počáteční hodnoty – na bubnu se rozsvítí dvě kontrolní LED, sloužící k signalizaci otáčení bubnem a ostatní LED jsou zhasnuté. Nulová teplota a prázdná pračka.

Model navíc generuje dva druhy chyb. Opravitelná a neopravitelná. Opravitelná chyba vznikne, pokud současně přivedeme signál pro otáčení bubnem vpravo a otáčení bubnem vlevo. Chyba se dá opravit odebráním jednoho ze signálů pro otáčení.

Neopravitelná chyba vzniká buď při přetečení pračky, nebo při teplotě vody větší než 90° C. Pokud neopravitelná chyba nastane, je označena rozsvěcením kontrolky „ERR“ a z tohoto stavu se lze dostat pouze pomocí stisknutí tlačítka „RES“ (Tlačítko je ukryto pod obrázkem ruky se zdviženým palcem, stačí model pračky v tomto místě zmáčknout.).



Obrázek 13 - Model automatické pračky,
zdroj: http://www.edumat.cz/foto/pracka_mala.gif

3.2.1 Zapojení

Model pračky byl připojen pomocí 20 žilového kabelu ke vstupům a výstupům PLC. První červený vodič (zem) a druhý vodič (+24 V) byly připojeny do zdroje PLC.

Vodiče číslo 3, 4, 5 a 9, 10, 11, sloužící k ovládání modelu pračky, byly zapojeny do digitálního reléového výstupního modulu 1746-OW16. Byl sem také zapojen vodič číslo 8, který sloužil jako přívod napětí (+ 24 V stejnosměrné) a díky kterému se při sepnutí reléového výstupu toto napětí přivedlo na připojený vodič daného výstupu.

Vodiče 15-20, které sloužily ke zjištění stavu, ve kterém se pračka nachází, byly připojeny k digitálním vstupům, fungujícím jako zátěž, modulu 1746-IB16. Do tohoto modulu byl zapojen také vodič č. 7, na nějž byla přivedena zem, proti které se jednotlivé vstupy spínaly.

Vodiče 12-14 a vodič č. 6 nemají využití a proto zůstaly nezapojeny.

Tabulka 5 - Seznam adres zapojených vstupů a výstupů

	proměnná	č. vodiče	adresa
Vstupy do PLC	TLACITKO_1	-	I:1/0
	TLACITKO_2	-	I:1/1
	TLACITKO_3	-	I:1/2
	TLACITKO_4	-	I:1/3
	TLACITKO_5	-	I:1/4
	TLACITKO_6	-	I:1/5
	HLADINA_50	15	I:2/0
	HLADINA_100	16	I:2/1
	TEPLOTA_30	20	I:2/2
	TEPLOTA_40	19	I:2/3
	TEPLOTA_60	18	I:2/4
	TEPLOTA_90	17	I:2/5
Výstupy z PLC	OVODA	10	O:4/0
	OBUBEN_P	3	O:4/1
	OBUBEN_L	4	O:4/2
	OOTACKY	5	O:4/3
	OTOPENI	9	O:4/5
	OCERPADLO	11	O:4/6

Pro ovládání modelu pračky bylo zapotřebí připojení tlačítek. Šestice tlačítek byla připojena do digitálních vstupů modulu 1746-IV16. Do tohoto modulu bylo také připojeno napájecí napětí ze zdroje (+ 24 V), které bylo při sepnutí tlačítka spojeno se zemí na zdroji.

K naprogramování automatu, který ovládal chování pračky bylo také zapotřebí propojit počítač s automatem. To bylo provedeno pomocí propojovacího konvertoru 1747-PIC, který se na jedné straně připojuje k počítači přes RS-232 a na druhé straně k automatu pomocí RJ-45.

4 Úlohy odzkoušené na modelu

Na modelu pračky bylo naprogramováno a odzkoušeno deset úloh. Na těchto úlohách bylo ukázáno použití různých instrukcí, které se obvykle používají při programování PLC automatů.

Všechny úlohy jsou také vyřešeny a umístěny jako programové soubory v cd příloze k této práci.

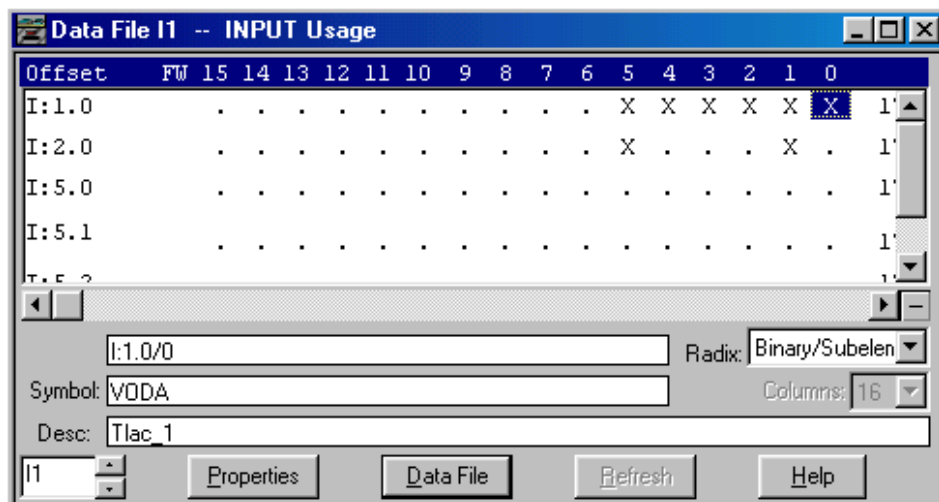
4.1 Úloha 1

Zadání:

Proveďte test adresace výstupních a vstupních signálů modelu pračky. Vytvořte soubor s definicemi symbolických jmen vstupních a výstupních proměnných PLC.

Řešení:

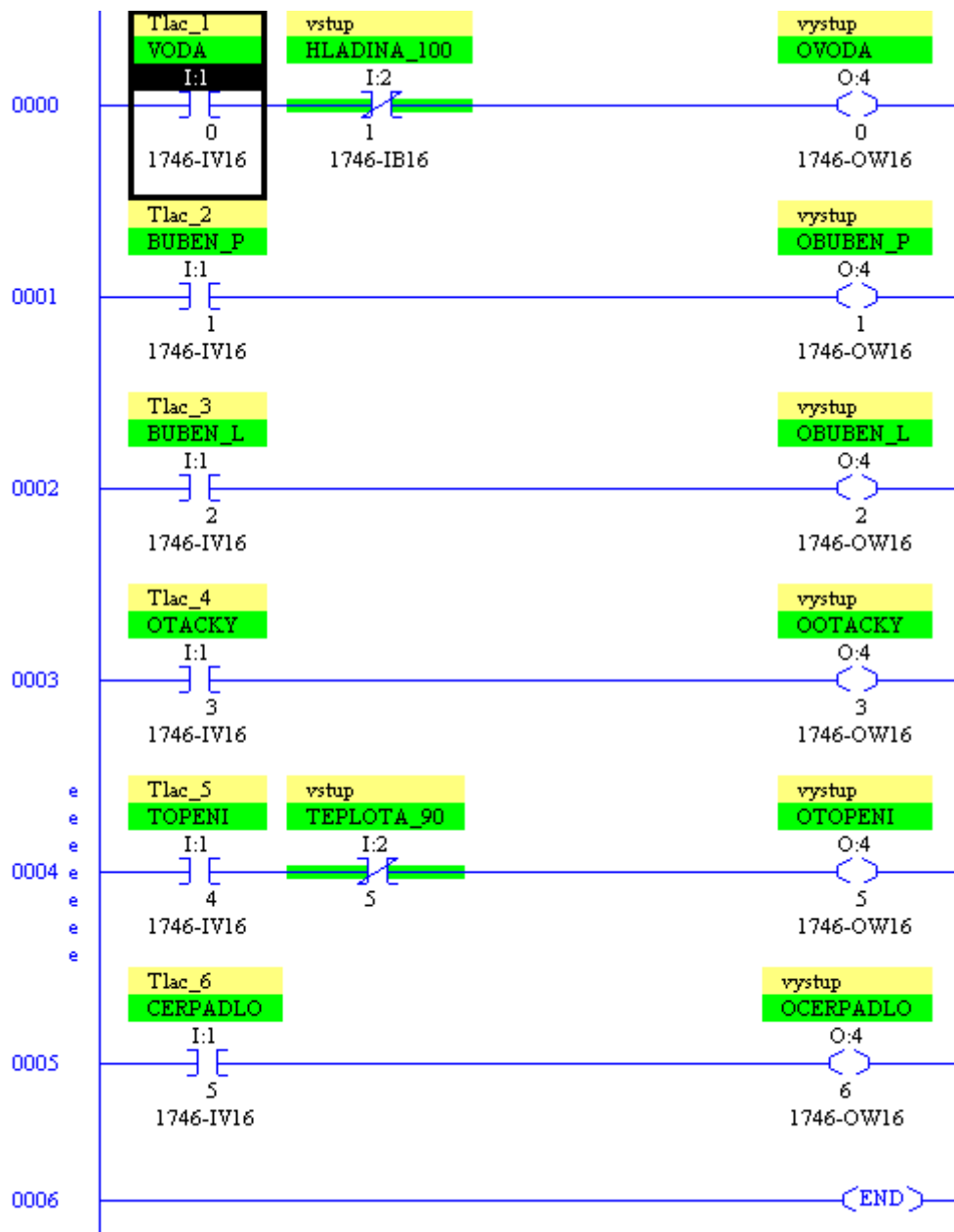
Řešení tohoto příkladu je poměrně jednoduché, ovšem důležité pro pokračování a programování složitějších příkladů. Nejprve je nutné si pojmenovat použité vstupy a výstupy. Pojmenování se provádí v okně datových souborů, které si otevřeme dvojklikem na název daného datového souboru v levé části obrazovky. Klikneme si na bit, který chceme pojmenovat a do pole symbol napíšeme název. Poté se na daný bit můžeme v programu odkazovat pouze jeho názvem a nemusíme vypisovat celou adresu.



Obrázek 14 - Datový soubor I1

Po pojmenování vstupů a výstupů stačí každému stisknutému tlačítku přiřadit určitou akci. Schéma první úlohy je na následujícím obrázku.

Celá úloha byla vyřešena pouze pomocí vstupních instrukcí XIC (pokud je bit na adrese funkce v „1“, je tato příčka spojena, pokud je v „0“, příčka je rozpojena), XIO (opak XIC) a výstupní instrukce OTE (pokud je příčka spojena, bit na adrese funkce se nastaví do „1“, je-li rozpojena, bit se nastaví do „0“). Pro ošetření chybových stavů byly přidány dvě funkce. První funkcí bylo vypnutí přívodu vody při dosažení hladiny vody 100 % a druhou funkcí bylo vypnutí ohřívání vody při dosažení teploty vody 90°C.



Obrázek 15 - Řešení úlohy 1

4.2 Úloha 2

Zadání:

Po stisku tlačítka START roztočte buben pračky doprava a začněte napouštět vodu. Po naplnění nádrže na 50 % buben zastavte a po napuštění celé nádrže vypněte vodu. Oba výstupy bude možné vypnout stiskem tlačítka STOP.

Řešení:

K vyřešení této úlohy stačily stejné instrukce jako v úloze 1. Sestavení samotného diagramu je intuitivní. Budou zde čtyři vstupy (tlačítko start, tlačítko stop, hladina 50 % a hladina 100 %) a dva výstupy (otáčení bubnem vpravo a napouštění vody).

4.3 Úloha 3

Zadání:

Při stisku tlačítka START začnete napouštět vodu a roztočte buben pračky doleva. Po naplnění nádrže na 50 % zapněte topení. Po naplnění celé nádrže ukončete napouštění, po ohřátí vody na 60°C vypněte topení a zastavte buben.

Řešení:

Řešení této úlohy je velmi podobné úloze 2. Pro vyřešení bylo záměrně použito jiných instrukcí výstupu (OTL a OTU). OTL je funkce, která reaguje na náběžnou hranu a dokáže bit nastavit do „1“, při rozpojení příčky však tento bit již nenastavuje do „0“. OTU funkce naopak slouží k resetování nastaveného bitu.

4.4 Úloha 4

Zadání:

Zadání z úlohy 3 vyřešte pomocí sequencerů.

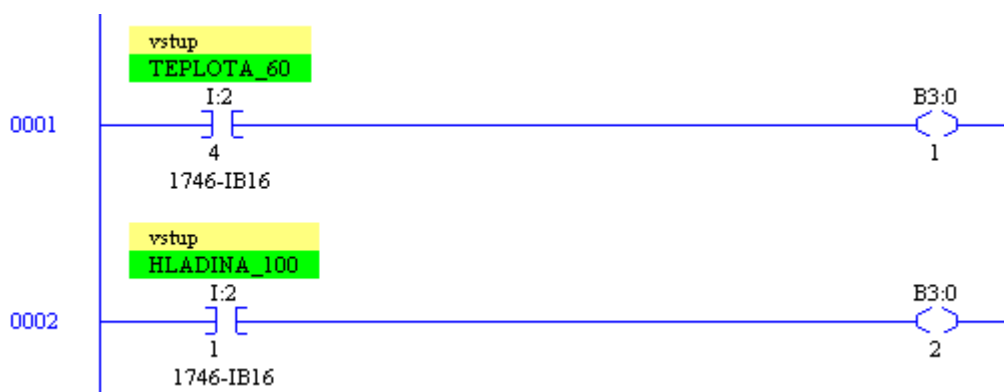
Řešení:

Nejprve je nutné si připravit redukovanou tabulku přechodů, která nám pomůže v následném nastavení, tuto tabulku si vytvoříme například v celočíselném datovém souboru N7, který přepneme do binárního módu. Stav pračky sledujeme na vstupu pomocí binárního datového souboru B3:0

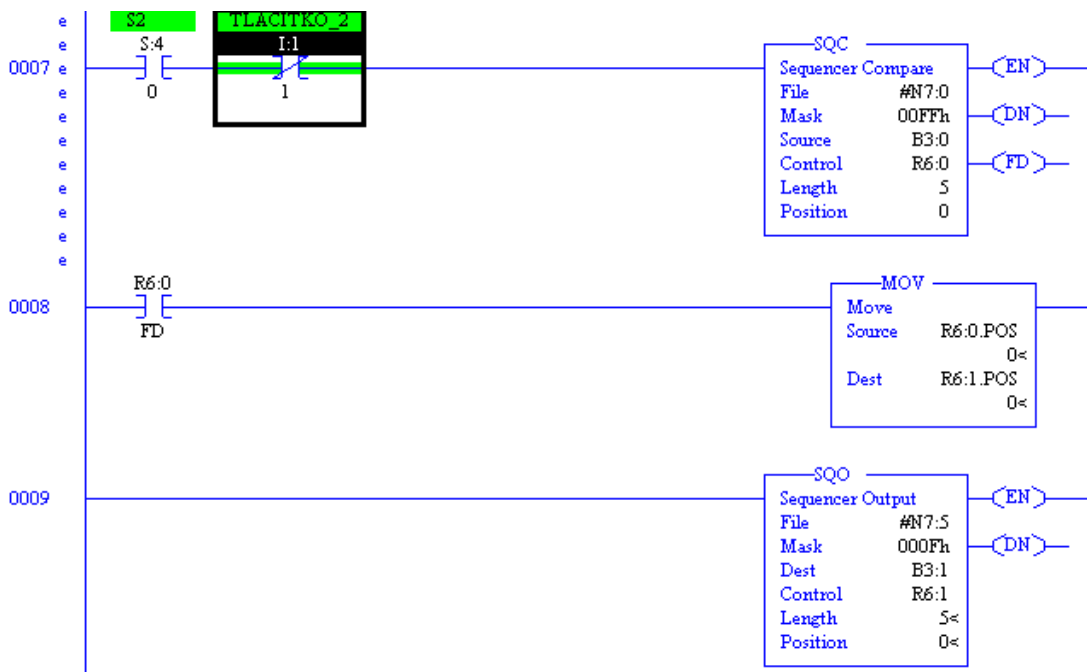
a porovnávané se stavy v tabulce pomocí SQC. SQC se na náběžnou hranu pootočí na další pozici a data zde uložená porovná s daty na pozici source. Pokud je nalezena shoda, nastaví se bit FD. Po poslední pozici následuje opět první pozice. SQC přetáčíme pomocí stavového bitu S:4/0 (hodiny). Pokud najdeme shodu, přetočíme sequencer (SQO). K přetočení nám slouží instrukce MOV. Bity na výstupu si nastavíme stejným způsobem jako na vstupu. Můžeme k tomu použít binární datový soubor obdobným způsobem.

Tabulka 6 - Redukovaná tabulka přechodů

Stav	Stavy				Akce		
	HL_50	HL_100	TEPL_60	START	BUB_L	VODA	TOPENI
0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0
2	1	0	0	0	1	1	1
3	1	1	0	0	1	0	1
4	1	1	1	0	0	0	0



Obrázek 16 – Sledování stavu pračky pomocí binárního datového souboru



Obrázek 17 - Zapojení sequencerů

4.5 Úloha 5

Zadání:

Po stisku tlačítka start roztočte buben pračky na 20 s doleva.

Řešení:

Zde stačí nastavit časovač TON na 20 s a pomocí bitu DN u časovače rozpojit příčku, která spouští otáčení bubnem pračky. Pro správný chod časovače, který čítá časové intervaly pouze pokud jsou předchozí podmínky v příčce splněny, je nutné si stisknutí tlačítka start zapamatovat například pomocí bitu B3:0/0.

4.6 Úloha 6

Zadání:

Po stisku tlačítka START vyvolejte cyklus, který bude střídavě otáčet bubnem pračky na obě strany až do stisku tlačítka STOP.

Řešení:

V datovém souboru N se nastaví doba otáčení bubnu doleva, doba otáčení bubnu doprava a doba mezi otáčeními. K vypočítání jednotlivých časů budeme potřebovat ještě v tomto souboru rezervovat dvě pozice pro dva mezivýpočty. Pomocí aritmetických funkcí ADD si vypočítáme do první rezervované pozice dobu otáčení bubnem doleva s dobou mezery a do druhé pozice sečteme dobu otáčení doprava s předchozí spočítanou dobou. Třetí aritmetickou instrukci ADD použijeme k sečtení doby mezery s předchozím výpočtem, tím získáme celkový potřebný čas cyklu, který uložíme do nastavení časovače. Poté použijeme porovnávací funkce a v jednotlivých časech spouštíme buď otáčení doleva, nebo otáčení doprava. V době mezery jsou oba výstupy vypnuté.

4.7 Úloha 7

Zadání:

Po stisku tlačítka START začněte napouštět vodu, po 5 s roztočte buben pračky doleva a po naplnění nádrže na 50 % zapněte topení. Po naplnění nádrže ukončete napouštění a po ohřátí vody na 60°C vypněte topení. Buben pračky nechte otáčet ještě 20 s.

Řešení:

Řešení této úlohy bylo podobné jako řešení úlohy č. 3. Stačí doplnit úlohu 3 o dva časovače, jeden zpozdí roztočení bubnu pračky o 5 s po zapnutí vody a druhý vypne otáčení bubnem po 20 s od vypnutí topení.

4.8 Úloha 8

Zadání:

Zadání z úlohy 7 vyřešte pomocí sequencerů.

Řešení:

Zde můžeme použít řešení z úlohy 4, které rozšíříme o dva časovače a redukovanou stavovou tabulku rozšíříme o další stavy.

Tabulka 7 - Redukovaná stavová tabulka

Stav	Stavy						Akce		
	HL_50	HL_100	TEP_60	TOC	ZAST	START	BUB_L	VODA	TOP
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	1	0
2	0	0	0	1	0	0	1	1	0
3	1	0	0	0	0	0	1	1	1
4	1	1	0	0	0	0	1	0	1
5	1	1	1	0	0	0	1	0	0
6	1	1	1	0	1	0	0	0	0

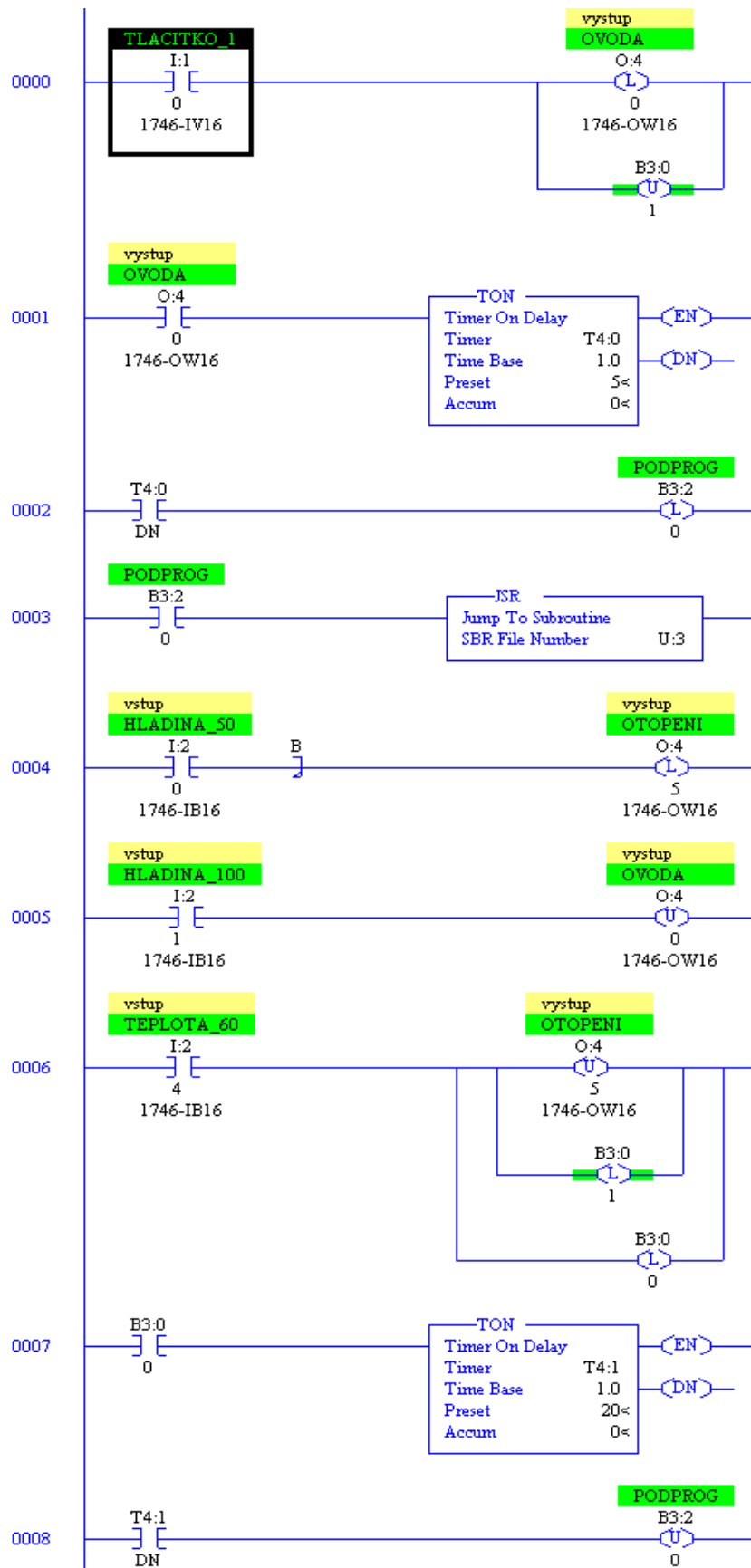
4.9 Úloha 9

Zadání:

Doplňte zadání z úlohy 7 tak, aby se buben pračky střídavě otáčel na obě strany.

Řešení:

Použijeme zadání z úlohy 7 a místo zapnutí otáčení bubnu použijeme skok do podprogramu (JSR). Vytvoříme podprogram (nový programový soubor), který bude podobný úloze 6, kde na začátku bude instrukce SBR.



Obrázek 18 - Hlavní program úlohy 9

4.10 Úloha 10

Zadání:

Po stisku tlačítka START začnete napouštět vodu a po naplnění nádrže na 50 % zapněte topení a po dalších pěti sekundách roztočte buben pračky střídavě na obě strany. Po naplnění nádrže ukončete napouštění a po ohřátí vody na 90°C vypněte topení. Buben pračky nechte otáčet ještě 30 s. Potom vypusťte vodu a prádlo vyždímejte (asi 30 s).

Řešení:

Postup řešení bude podobný jako u předchozího příkladu. Střídavé otáčení bubnu vyřešíme stejným podprogramem jako u předchozího zadání, navíc však přidáme podprogram ždímání, kde se budou v určitém časovém intervalu střídát akce vypouštění vody a rychlé otáčení bubnem pračky na obě strany.

Závěr

Při zpracování své bakalářské práce jsem se zaměřil především na problematiku logického řízení a programovatelných logických automatů a vytvoření úlohy řízení automatické pračky, která by mohla být v budoucnu použita jako úloha při výuce programování logických automatů.

Během její tvorby jsem narazil na problémy s instalací softwarového vybavení počítače. Šlo o produkty firmy Rockwell a jednalo se o programy RS Linx a RS Logix. Jako první bylo problémové nainstalování programu RS Linx pod operačním systémem Windows XP. To bylo vyřešeno nainstalováním systému Windows 98. Další menší problém s registrací programů byl vyřešen doinstalováním disketové mechaniky do počítače. První seznámení s těmito produkty proběhlo rychle, pracovní prostředí je intuitivní a uživatel se v něm rychle zorientuje.

Dalším úkolem, který jsem řešil bylo připojení modelu automatické pračky k PLC automatu. Zde bylo nutné prostudovat manuál s popisem jednotlivých modulů automatu a navrhnout korektní řešení připojení.

Složitějším úkolem bylo naprogramování úloh číslo 4 a 10. V úloze 4 jsem se nově seznamoval se sequencery, u kterých mi chvíli trvalo pochopit způsob, jakým se používají. Úloha 10 mi dělala potíže při spouštění podprogramu v těle jiného podprogramu. Ačkoliv jsem tuto úlohu vyřešil, několikrát odzkoušel a výsledek odpovídal zadání, nejsem si zcela jist správností způsobu naprogramování. Myslím že úloha 10 by šla vyřešit i jiným, lepším způsobem, který mě bohužel nenapadl.

V rámci práce se mi podařilo shrnout poznatky teorie logického řízení a programovatelných logických automatů a úspěšně sestavit a naprogramovat zadané úlohy. Splnil jsem tak zadání a vytvořil podklad použitelný pro laboratorní úlohu programování PLC.

Literatura:

ŠMEJKAL, Ladislav, MARTINÁSKOVÁ, Marie. *PLC a automatizace : 1. díl - Základní pojmy, úvod do programování*. 1. vyd. Praha : BEN, 2007. 224 s. ISBN 978-80-86056-58-6.

ŠVARC, Ivan. *Automatizace : Automatické řízení*. 2. dopl. vyd. Brno : Vysoké učení technické, Fakulta strojního inženýrství, 2005. 262 s. ISBN 80-214-2943-7.

ŠVARC, Ivan, NĚMEC, Zdeněk, LACKO, Branislav. *Automatizace*. Brno : PC-DIR, c1995. 192 s. ISBN 80-214-0704-2.

TŮMA, Jiří. *Logické řízení* [online]. c2007 [cit. 2009-04-01]. Pdf. Dostupný z WWW: <<http://homel.vsb.cz/~tum52/download/LogickeRizeni.pdf>>.

VONDRA, Zdeněk. *Základy programování PLC* [online]. Praha : Střední průmyslová škola elektrotechnická, Ječná, 2006 [cit. 2009-04-12]. Pdf. Dostupný z WWW: <http://web.spsejecna.cz/projekt/PLC_zakl.pdf>.

WIKIPEDIA. *Programovatelný logický automat* [online]. 31.10.2006 , 1.3.2009 [cit. 2009-04-07]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Programovateln%C3%BD_logick%C3%BD_automat>.

Seznam obrázků a tabulek

Seznam obrázků:

OBRÁZEK 1 - VNITŘNÍ STRUKTURY PLC.....	14
OBRÁZEK 2 - CYKLUS PROGRAMU PLC	19
OBRÁZEK 3 - KARNAUGHOVA MAPA.....	27
OBRÁZEK 4 - MINIMALIZACE POMOCÍ KARNAUGHOVY MAPY	29
OBRÁZEK 5 - LINIOVÉ SCHÉMA.....	31
OBRÁZEK 6 - PRKY NAND A NOR	32
OBRÁZEK 7 - SCHÉMA POMOCÍ NAND	33
OBRÁZEK 8 - SCHÉMA POMOCÍ NOR	34
OBRÁZEK 9 - OKNO PROGRAMU RS LINX	37
OBRÁZEK 10 - KONFIGURACE MODULŮ V RS LOGIX	38
OBRÁZEK 11 - PŘÍČKA S JEDNÍM VSTUPEM A JEDNÍM VÝSTUPEM	39
OBRÁZEK 12 – LEVÉ OKNO RS LOGIX S UKÁZKOU SOUBORŮ	39
OBRÁZEK 13 - MODEL AUTOMATICKÉ PRAČKY.....	41
OBRÁZEK 14 - DATOVÝ SOUBOR II.....	45
OBRÁZEK 15 - ŘEŠENÍ ÚLOHY 1	46
OBRÁZEK 16 – SLEDOVÁNÍ STAVU PRAČKY POMOCÍ BINÁRNÍHO DATOVÉHO SOUBORU.....	48
OBRÁZEK 17 - ZAPOJENÍ SEQUENCERŮ	49
OBRÁZEK 18 - HLAVNÍ PROGRAM ÚLOHY 9	52

Seznam tabulek:

TABULKA 1 - LOGICKÉ FUNKCE JEDNÉ PROMĚNNÉ.....	21
TABULKA 2 - LOGICKÉ FUNKCE DVOU PROMĚNNÝCH	22
TABULKA 3 - PRAVDIVOSTNÍ TABULKA	25
TABULKA 4 - ALGEBRAICKÁ FUNKCE VYJÁDŘENÁ POMOCÍ PRAVDIVOSTNÍ TABULKY	29
TABULKA 5 - SEZNAM ADRES ZAPOJENÝCH VSTUPŮ A VÝSTUPŮ.....	42
TABULKA 6 - REDUKOVANÁ TABULKA PŘECHODŮ	48
TABULKA 7 - REDUKOVANÁ STAVOVÁ TABULKA	51