

Příprava laboratoře pro výuku předmětu Programovatelné automaty

Preparation of the Laboratory for
Programmable Controllers Education

Tomáš Čermák

Bakalářská práce
2012

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš ČERMÁK**

Osobní číslo: **A09011**

Studijní program: **B 3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Téma práce: **Příprava laboratoře pro výuku předmětu
Programovatelné automaty**

Zásady pro vypracování:

1. Popište produktovou řadu programovatelných automatů (PLC) značky ABB, včetně vývojového prostředí.
2. Vytvořte jednoduchý model simulující chování reálného zařízení.
3. Připojte k tomuto modelu PLC značky ABB a realizujte jeho řízení.
4. Vytvořte jednoduchý manuál s návodem tvorby řídicího software pro automaty výše uvedené značky.
5. Navrhněte a realizujte vzorové úlohy pro výuku programování automatů ABB v rámci předmětu Programovatelné automaty.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace, 1.díl. 1. Vyd. Praha: BEN – technická literatura, 1999. ISBN 80-860-5658-9.
2. ŠMEJKAL, Ladislav. PLC a automatizace, 2.díl. 1. Vyd. Praha: BEN – technická literatura, 2005. ISBN 80-7300-087-3.
3. MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.
4. MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty II. Vyd. 1. Praha: ČVUT, Strojní fakulta, 2000. ISBN 80-010-2096-7.
5. VOJÁČEK, Antonín. TEST – PLC ABB PM564 ? 1.díl ? popis hardwaru. Automatizace.hw.cz. [online]. Březen 31, 2011 [cit. 2012-01-26]. Dostupné z: <http://automatizace.hw.cz/test-plc-abb-pm564-1dil-popis-hardwaru>
6. VOJÁČEK, Antonín. TEST – PLC ABB PM564 ? 2.díl ? programování. Automatizace.hw.cz. [online]. Duben 2, 2011 [cit. 2012-01-26]. Dostupné z: <http://automatizace.hw.cz/test-plc-abb-pm564-2dil-programovani>
7. Firemní literatura k produktům ABB.

Vedoucí bakalářské práce:

Ing. Tomáš Sysala, Ph.D.

Ústav automatizace a řídicí techniky

Konzultant:

Ing. Jiří Ivánek

Datum zadání bakalářské práce:

24. února 2012

Termín odevzdání bakalářské práce:

8. června 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Tato práce se zabývá vytvořením nového pracoviště a materiálů pro výuku programování programovatelných automatů (PLC), konkrétně automatů firmy ABB - modelu PM564-T-ETH. Je zde popsán hardware tohoto automatu, způsob programování a vývojové prostředí CoDeSys. Dále tato práce obsahuje návod pro studenty, jenž kompletně popisuje zapojení a nastavení komunikace, nastavení konfigurace a vývoj programu v prostředí CoDeSys. Pro praktickou ukázkou využití PLC byl navrhnout a sestaven model simulující činnost přečerpávací elektrárny. Poslední část bakalářské práce je zaměřena na vytvoření vzorových úloh, ve kterých toto PLC s přídatným modulem digitálních vstupů a výstupů ABB DX561 řídí modely EDUtec.

Klíčová slova:

Programovatelný automat, ABB, manuál, výukové modely EDUtec, přečerpávací elektrárna, vzorové úlohy, CoDeSys

ABSTRACT

This thesis deals with the formation of a new workplace as well as a manual intended for the teaching of programming programmable logic controllers (PLC) from the company ABB – specifically for the model PM564-T-ETH. The thesis contains a description of this controller's hardware, the way of programming and the development environment CoDeSys. Furthermore, there is a manual for students, which describes connection and setting of communication, setting of configuration and development of program in CoDeSys. As a practical example of the PLC usage, a model imitating the function of a pumped storage hydro plant was designed and built. The last part of the thesis is aimed at exemplary tasks, in which this controller along with the additional module with digital inputs and outputs ABB DX561 controls models EDUtec.

Keywords:

Programmable controller, ABB, manual, educational models EDUtec, pumped storage hydro plant, exemplary tasks, CoDeSys

Poděkování

Chtěl bych poděkovat panu Ing. Tomáši Sysalovi Ph.D. za jeho odborné vedení mé bakalářské práce, ochotu a trpělivost. Dále bych chtěl poděkovat panu Ing. Ivánkovi za technickou podporu. V neposlední řadě děkuji své rodině a přítelkyni.


Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně


.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PROGRAMOVATELNÝ LOGICKÝ AUTOMAT	11
1.1 PROGRAMOVATELNÝ LOGICKÝ AUTOMAT	11
1.2 ROZDĚLENÍ PLC.....	12
1.3 PROGRAMOVÁNÍ PLC.....	13
1.3.1 Norma IEC 61131	13
1.3.2 Norma IEC 61131-3	14
1.3.2.1 Popis normy IEC 61131-3.....	14
2 ABB	16
2.1 NABÍDKA PLC OD ABB	16
2.1.1 AC500	16
2.1.2 AC500-eCo	16
2.1.3 Srovnání řad AC500 a AC500-eCo.....	17
3 PLC ABB PM564-T-ETH	18
3.1 HARDWAROVÝ POPIS	18
3.2 PROGRAMOVÁNÍ AUTOMATU	19
3.2.1 ABB Configurator	20
3.2.2 CoDeSys.....	21
II PRAKTICKÁ ČÁST	22
4 NÁVOD NA TVORBU ŘÍDICÍHO SOFTWARE	23
4.1 PŘIPOJENÍ A NASTAVENÍ KOMUNIKACE PRO ÚČELY PROGRAMOVÁNÍ.....	23
4.2 ZALOŽENÍ PROJEKTU A MOŽNÁ NASTAVENÍ V ABB CONFIGURATOR.....	25
4.2.1 Popis projektu v postranní liště Devices	26
4.3 NASTAVENÍ A VÝVOJ PROGRAMU V CODESYS	28
4.3.1 Popis programu CoDeSys	28
4.3.1.1 Nastavení komunikace	28
4.3.1.2 PLC Browser.....	29
4.3.1.3 Tvorba programu	29
4.3.1.4 Tvorba simulace.....	30
5 MODEL PŘEČERPÁVACÍ ELEKTRÁRNY	32
5.1 PŘEČERPÁVACÍ ELEKTRÁRNA	32
5.2 POPIS MODELU.....	33
5.2.1 Elektrická zapojení použitá v modelu	34
5.2.1.1 Zapojení doporučená výrobcem.....	34
5.2.1.2 Zapojení digitálních vstupů a výstupů	35
5.2.1.3 Zapojení analogových výstupů	36
5.2.1.4 Zapojení pro obvod proudu vody.....	37
5.2.2 Model přečerpávací elektrárny	38

5.3	PROGRAMOVÁNÍ MODELU	39
5.3.1	Požadavky na řízení modelu	39
5.3.2	Programové řešení.....	39
5.3.3	Vizualizace	40
6	VZOROVÉ ÚLOHY	41
6.1	PROGRAM Č. 1 – LED DIODY A TLAČÍTKA	41
6.1.1	Potřebné vybavení	41
6.1.2	Zadání.....	41
6.1.3	Řešení.....	41
6.2	PROGRAM Č. 2 – KŘIŽOVATKA S DVOJÍM REŽIMEM.....	42
6.2.1	Potřebné vybavení	42
6.2.2	Zadání.....	42
6.2.3	Řešení.....	43
6.3	PROGRAM Č. 3 - OVLÁDÁNÍ PRAČKY	44
6.3.1	Potřebné vybavení	44
6.3.2	Zadání.....	44
6.3.3	Řešení.....	44
	ZÁVĚR	46
	CONCLUSION	47
	SEZNAM POUŽITÉ LITERATURY.....	49
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	51
	SEZNAM OBRÁZKŮ	52
	SEZNAM TABULEK.....	53
	SEZNAM PŘÍLOH.....	54

ÚVOD

Na začátku první poloviny 70. let vznikl díky rozmachu mikroelektronických technologií efektivní prostředek průmyslové automatizace – programovatelný automat (PLC). Dnes programovatelné automaty vyrábí mnoho firem. Na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně se k výuce používají hlavně programovatelné automaty značky Teco a Saia. [22, str. 7]

Cílem této bakalářské práce je představit a popsat programovatelný automat firmy ABB PM564-T-ETH po stránce hardwaru i softwaru a vytvořit jednoduchý návod pro studenty, kteří s automaty firmy ABB nemají žádné zkušenosti. Návod je pro přehlednost rozdělen do tří částí. První část popisuje zapojení automatu a nastavení komunikace, druhá se zabývá nastavením konfigurace a poslední část je zaměřená na vývoj programu.

Pro ukázkou funkcí tohoto automatu byl vytvořen model přečerpávací elektrárny. Na tomto modelu byla popsána zapojení vstupů a výstupů PLC. Také byl vytvořen program pro řízení modelu a vizualizace, která zobrazuje aktuální stav procesů v elektrárně (množství vody v nádržích, výrobu elektrické energie, přečerpávání vody).

Závěrečná část bakalářské práce obsahuje zadání vzorových úloh pro výuku programování automatů ABB a jejich řešení. Úlohy jsou vytvořené pro řízení modelů ze sady EDUtec. Studenti si díky nim mají možnost vyzkoušet různé typy programování a ovládání.

I. TEORETICKÁ ČÁST

1 PROGRAMOVATELNÝ LOGICKÝ AUTOMAT

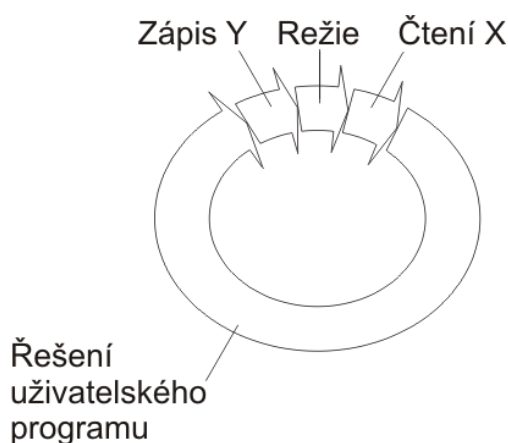
1.1 Programovatelný logický automat

Programovatelný logický automat (Programmable Logic Control - PLC) je elektronický řídicí systém, jehož funkce je určena programem, který je do něj uložen. Původní úlohou programovatelných automatů byla náhrada za reléovou logiku – u které při požadavku na změnu funkce bylo potřeba manuálního zásahu do elektrických obvodů. PLC umožnily bez nutnosti zásahu do fyzické části velmi efektivně funkce obvodu změnit, stačilo jen do jeho paměti nahrát nový program. [12], [21, str. 182]

Dnešní PLC umí mnohem více než jen nahrazovat reléovou logiku - je to z důvodu růstu náročnosti aplikací, pro které byly PLC vytvořeny. Dnes PLC umí kromě své prapůvodní úlohy i komunikovat s nadřizenými systémy, řídit a vyhodnocovat analogové signály a například na nich může být spuštěn webový server.

Technické vybavení programovatelných automatů je navrženo tak, aby obstály v drsném průmyslovém prostředí a mohly správně vykonávat svou funkci. [16, str. 39]

Programovatelný logický automat nejčastěji pracuje v cyklu – tento cyklus obsahuje režii, čtení ze vstupů, řešení uživatelského programu a zápis na výstup. [16, str. 47]



Obrázek 1 - Cyklus PLC [16, str. 46]

1.2 Rozdělení PLC

PLC můžeme rozdělit dle velikosti, výkonnosti a podle vstupně/výstupních kritérií.

1. Dle velikosti:

a) Mikro PLC

Tyto modely jsou nejmenší a nejlevnější PLC systémy, které nabízejí uživateli pevnou sestavu většinou jen binárních vstupů a výstupů. Tyto systémy většinou nelze dále rozšiřovat. Často chybí možnost komunikace. Použití tohoto typu PLC může být například jako logické řízení jednoduchých strojů a mechanismů, která se před nástupem PLC řešila reléovou logikou. Pro toto použití v dnešní době hovoří příznivá cena, velká spolehlivost a efektivita tvorby a změny programu PLC. [16, str. 40]

b) Kompaktní PLC

Tento typ PLC je v dnešní době nejrozšířenější. Vyznačuje se určitou leč omezenou variabilitou ve volbě přídavných modulů, které lze k základnímu modelu připojit. [16, str. 40]

c) Modulární PLC

Základem tohoto PLC je pevné procesorové jádro s napáječem umístěným v rámu, nosné desce nebo na liště. Rozšíření je realizováno přes sběrnici a to buď vzdálených, nebo místních periferních jednotek. Díky této variabilitě lze poskládat systém, který bude přesně vyhovovat dané aplikaci. [13, str. 5]

2. Dle výkonu:

Výkon je tvořen jednotkou CPU, která je osazena mikroprocesorem, u něhož můžeme porovnávat vlastnosti. První vlastností je délka zpracovaného slova, které je vyjádřeno v bitech. Nejčastěji se dnes setkáváme se 16 bitovými, 32 bitovými a výjimečně se 64 bitovými mikroprocesory v PLC. Dalším důležitým parametrem je rychlost mikroprocesoru a počet jader. Dalšími parametry ovlivňujícími výkon jsou velikost a rychlost vnitřních pamětí cpu. [22, str. 12]

3. Dle vstupně/výstupních periférií:

Některé PLC obsahují již v základu vstupně/výstupní periférie. Jiné obsahují pouze sběrnici, ke které se dají vstupně/výstupní jednotky připojit. Mezi základní jednotky řadíme digitální vstupy, digitální výstupy, analogové vstupy, analogové výstupy a jejich kombinace. Dále například čítačové, polohovací, komunikační a speciální jednotky (například pro řešení fuzzy regulace). [13, str. 25-28]

1.3 Programování PLC

Základní pojetí programování PLC lze shrnout do dvou přístupů – grafické a textové. Mnoho výrobců si vytvořilo či upravilo používané jazyky programování PLC, které se mnohdy velice podobají, ale převážně nejsou přenositelné. Tento problém se snaží odstranit norma IEC 61131-3 sjednocující programovací jazyky, která je součástí normy IEC 61131. [10], [11], [15]

1.3.1 Norma IEC 61131

Norma IEC 61131 je souborem požadavků týkajících se moderních řídicích systémů. Tato norma je mezinárodně uznávaná. Vychází z deseti starších norem a skládá se ze sedmi částí, jež jsou vypsány v následující tabulce a týkají se technického i programového vybavení řídicích systémů. [10], [11], [15]

Tabulka 1 - Rozdělení normy IEC 61131 [10]

Označení normy:	Název:
IEC 61131-1	Všeobecné informace
IEC 61131-2	Požadavky na zařízení a zkoušky
IEC 61131-3	Programovací jazyky
IEC 61131-4	Podpora uživatelů
IEC 61131-5	Komunikace
IEC 61131-7	Programování fuzzy logiky
IEC 61131-8	Směrnice pro aplikace a implementace programovacích jazyků pro programovatelné automaty

1.3.2 Norma IEC 61131-3

Tato norma je součástí normy IEC 61131 a zabývá se programovacími jazyky. Dnes se o tuto normu stará organizace PLCopen. Ke vzniku této normy vedla potřeba zjednodušit práci programátorům a především snížit náklady na školení, které vznikaly při přechodu z různých PLC systémů. [10], [11], [15]

1.3.2.1 Popis normy IEC 61131-3

Norma popisuje konfiguraci, která určuje nejvyšší úroveň softwarového řízení. V konfiguraci je možné definovat zdroje, které jsou tvořeny procesorovými jednotkami nebo speciálními procesory. Na zdroji může běžet několik programů. Každý program je spojen s úlohou, která odpovídá za provedení souboru programů či funkčních bloků, které jsou zahrnuty v POU.

POU (Program Organisation Unit) jsou základním stavebním blokem programovacích jazyků této normy. POU jsou nejmenší nezávislé softwarové jednotky uživatelského programu. Rozlišujeme tři typy POU vzestupně podle funkčnosti – funkci, funkční blok, program. POU je jednotka, která může být kompilovaná nezávisle na ostatních programovacích částech. Nicméně kompilátor potřebuje informace o volajícím rozhraní dalších POU, které jsou volány z dané POU. Nakompilované POU mohou být spolu spojeny později, aby vytvořily kompletní program. Tato nezávislost POU zjednodušuje rozsáhlou modularizaci automatizačních úloh stejně jako znovu použití již zavedených a testovaných softwarových jednotek.

Funkce může mít parametry, ale neobsahuje žádnou statickou paměť. Princip je stejný jako u klopných obvodů, kde výstupní hodnota je pouze dána vstupy a vnitřní funkcí obvodu (případně jen funkcí obvodu). Funkční blok naproti tomu obsahuje statickou paměť, která ovlivňuje výstupní hodnotu – je zde patrná podobnost se sekvenčními obvody, kde výstupní hodnota je dána vstupy, vnitřním stavem a obvodovou funkcí. Program - tento typ POU představuje hlavní program. Všechny proměnné z celého programu přiřazeny k fyzickým adresám (například PLC vstupům a výstupům) musí být deklarovány v této POU nebo nad ní (ve zdroji nebo v konfiguraci). Ve všech ostatních ohledech se program chová jako funkční blok.

Další důležitou součástí jsou datové typy, které nám určují základní typy proměnných, které můžeme používat jako lokální nebo globální proměnné, případně je sdružovat do polí nebo s nimi vytvářet struktury.

Každá POU je rozdělena na část deklarační a kódovou. Do deklarační části se deklarují proměnné daným datovým typem a dalšími vlastnostmi, kterými jsou nastavení inicializační hodnoty, zálohování baterií nebo přiřazení k fyzickým adresám. V kódové části se nachází kód, který je PLC vykonáván, ten můžeme dle typu reprezentace rozdělit na jazyky textové a grafické.

Programovatelné automaty jsou v základu tvořeny centrální jednotkou CPU (Central Processor Unit). Tato jednotka obsahuje sadu instrukcí, které využívá například programovací jazyk IL (Instructions List). Tento jazyk se interpretuje textově. Programátorovi umožňuje psát kód velmi efektivně, ale jeho nevýhodami bývají rozsah a nepřehlednost u velkých programů. Tyto nevýhody jsou odstraněny jazykem Strukturovaného textu (Structured Text), který se velmi podobá vyšším programovacím jazykům, typově Pascalu nebo C. Nevýhodou tohoto jazyka je především menší efektivnost kódu, která vychází z podstaty vyššího jazyka. Typickým zástupcem grafických programovacích jazyků je jazyk kontaktních schémat LD (Ladder Diagram). Tento jazyk vychází z elektromechanických reléových schémat. V programování se používá grafické znázornění reléové logiky a tento jazyk je primárně určen pro zpracování booleovských signálů. Dalším grafickým jazykem je jazyk logických schémat FBD (Function Block Diagram). Tento jazyk vychází vstříc uživatelům, kteří jsou zvyklí na kreslení logických schémat pro zařízení s integrovanými obvody. Jazyk sekvenčního programování SFC (Sequential Function Chart) patří také do grafických jazyků. Je založen na symbolice přechodových grafů a určité třídy Petriho sítí. [10], [11], [15]

2 ABB

Práce se zabývá programovatelnými automaty od firmy ABB, proto je zde stručná charakteristika této firmy. ABB je přední světová společnost poskytující technologii pro energetiku a automatizaci. Vznikla sloučením firem ASEA a BBC v roce 1988. Tyto firmy měly dlouhou tradici sahající až do konce devatenáctého století. Firma nyní zaměstnává 124 tisíc zaměstnanců a její pobočky můžeme nalézt ve více než 100 zemích světa. Působí v těchto odvětvích – výrobky pro energetiku, systém pro energetiku, automatizace výroby a pohony, výrobky nízkého napětí, procesní automatizace. [2], [3]

2.1 Nabídka PLC od ABB

Společnost ABB od roku 2006 nabízí sérii programovatelných logických automatů AC500. Tato série dosáhla velkého průmyslového uznání za vysoký výkon, kvalitu a spolehlivost. Automaty si našly uplatnění v těchto odvětvích a aplikacích – ve výrobě strojů, řízení budov, výrobě energie, potravinářském průmyslu a v lodním průmyslu. Dnes můžeme sérii AC500 rozdělit na dvě plně kompatibilní části – na AC500 a AC500-eCo. [1]

2.1.1 AC500

Tyto modely PLC jsou nejvyšší řadou malých PLC, které ABB nabízí. Vyznačují se velkým výběrem výkonnostních stupňů a rozšiřitelností. Dále byla představena verze AC500-XC, která je navržena, aby odolávala extrémním podmínkám. [1]

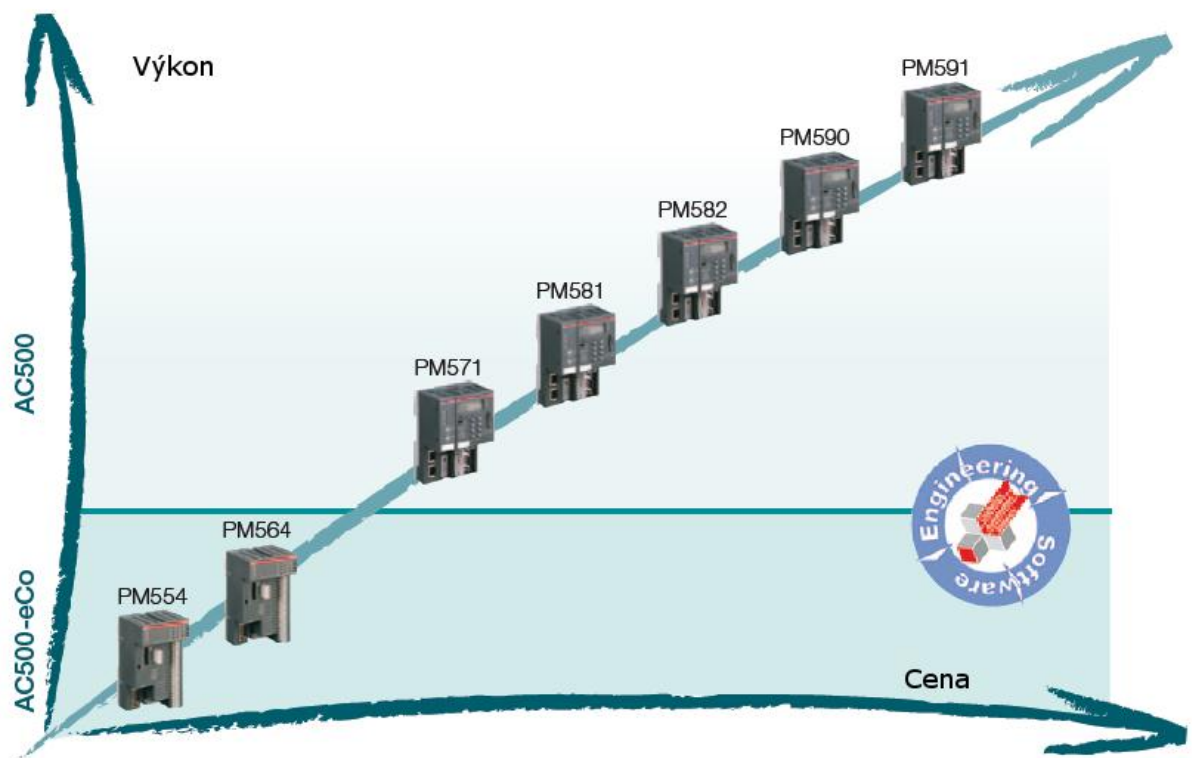
2.1.2 AC500-eCo

Tato řada PLC nabízí velmi výhodný poměr cena/výkon a je levnější variantou AC500. Už v základu nabízí centrální jednotku, která je doplněna o binární a analogové vstupy a výstupy. [1]

2.1.3 Srovnání řad AC500 a AC500-eCo

Tabulka 2 - Srovnání parametrů řad AC500 a AC500-eCo [1]

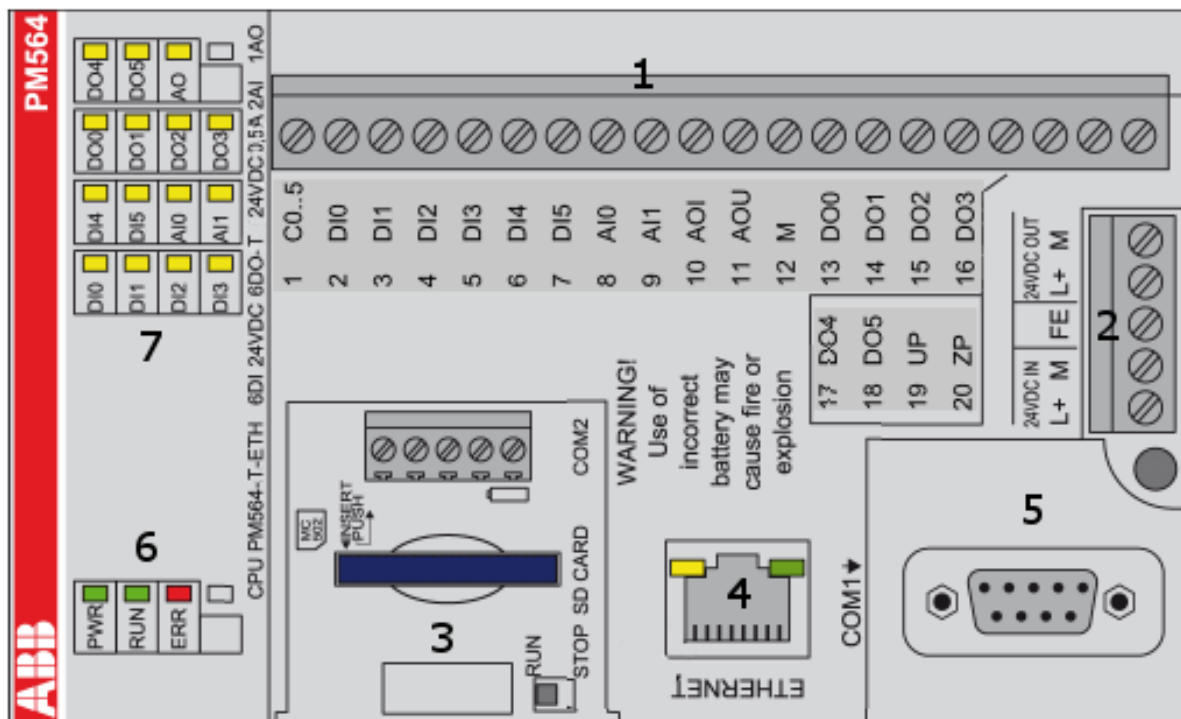
Srovnávaný parametr	AC500	AC500-eCo
Verze pro extrémní použití	Ano	Ne
Napájecí zdroj	24 V DC	24 V DC nebo 100-240 V AC
Paměť programu	128kB a více	pouze 128kB
Integrovaná paměť pro data	nad 128kB	pouze 14kB
Otočka cyklu pro 1 instrukci bit, word, floating point	0,06/0,09/0,7 μ s a méně	0,08/0,1/1,2 μ s
Vlastní I/O centrálních jednotek	žádné	D-I/O:6/6, A-I/O:2/1
Maximální počet připojitelných I/O	D-I/O:320/240,A-I/O:160/160	D-I/O:320/240,A-I/O:160/160
Multi tasking	ano	ne
Rozhraní	více a větší variabilita	základní, malá variabilita
Ethernetové funkce	navíc – IEC60870-5-104,SNTP	TCP/IP,DHCP,FTP,int. Web server



Obrázek 2 - Přehled PLC od firmy ABB [1]

3 PLC ABB PM564-T-ETH

Tento typ PLC patří do produktové řady AC500-eCo od firmy ABB.



Obrázek 3 - PLC ABB PM564-T-ETH [4]

3.1 Hardwarový popis

Pouzdro je tvořeno pevným šedým plastem odolným proti poškrábání a na dolní straně s připravenou zdírkou na posazení na DIN lištu. Z pouzdra vychází svorkovnice – malá (2) pro napájecí napětí a velká (1), která obsahuje vstupy a výstupy PLC. Dále se zde nachází odnímatelný kryt (3), pod nějž se dá vložit adaptér s RTC (Real-time Clock), COM2 nebo adaptér, který obsahuje jak RTC tak COM2. Vedle tohoto adaptéru je možné zde umístit slot pro paměťovou SD kartu. Vpravo od tohoto krytu se nachází ethernetové rozhraní (4) tvořené konektorem RJ-45 a vedle něho 9 pinový Sub-D jack sériového rozhraní COM1(5). Úplně vlevo se nachází informační panel (6) s LED diodami, který obsahuje indikaci napájení PLC (PWR), chodu PLC (RUN) a chyby PLC (ERR). Dále se zde nachází pole LED diod (7), které signalizují stav vstupně výstupních obvodů PLC.

PLC má v sobě ukrytý 32 bitový procesor PowerQUICC PowerPC 100 MHz od firmy Freescale. Na rozdíl od PLC z řady AC500, toto PLC z řady AC500-eCo nepodporuje multitasking. Dále obsahuje 128 kB pevnou Flash paměť programu a 14kB paměti SDRAM pro data, z toho 2 kB jsou zálohovány baterií.

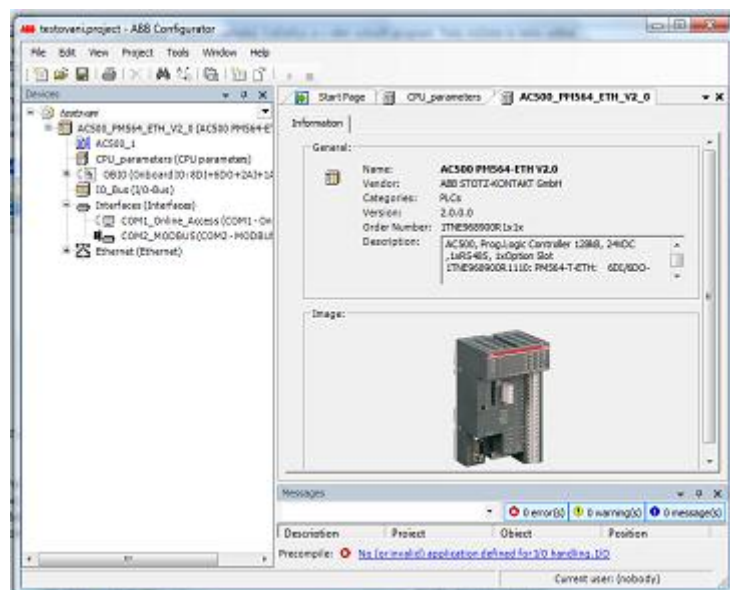
Výbava zahrnuje digitální a analogové vstupy a výstupy. U digitálních vstupů a výstupů je použita 24 V logika. Digitálních vstupů je 6 základních, tento počet se dá rozšířit o dva digitální vstupy na místo analogových vstupů. Dále zde máme 6 digitálních tranzistorových výstupů, které musíme pro chod napájet 24 V DC přivedeného na svorky UP a ZP. PLC má analogové dva vstupy a jeden výstup. U analogového výstupu si můžeme zvolit mezi napěťovým 0-10 V a proudovým módem 0-20 mA nebo 4-20 mA. Analogový vstup i výstup mají rozlišení 12 bitů. K PLC je možné připojit až 7 dalších vstupních nebo výstupních modulů.

Pro komunikaci má toto PLC vestavěné sériové rozhraní pro RS485 pojmenované COM1, které je možné použít jako programovací port, pro komunikační protokol MODBUS, sériovou ASCII komunikaci a CS31. Přidáním adaptéru COM2 je možné rozšířit PLC o druhé takové rozhraní. Další možná komunikace může být pomocí rozhraní Ethernetu, díky kterému se dá PLC také programovat. [4], [19]

3.2 Programování automatu

Automat se programuje pomocí softwaru Control Builder PS501, který obsahuje dvě části. Jedna část je ABB Configurator a druhou tvoří vývojové prostředí CoDeSys.

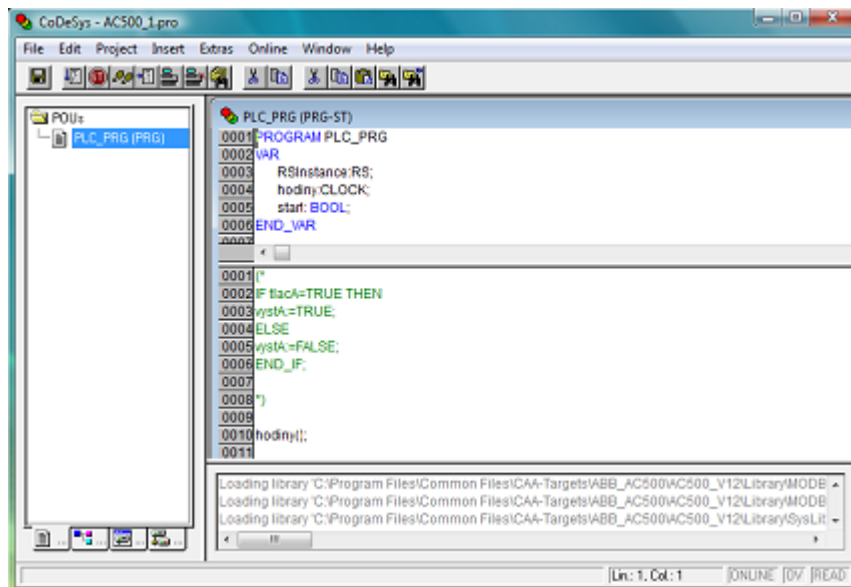
3.2.1 ABB Configurator



Obrázek 4 - ABB Configurator

V této aplikaci se tvoří projekty. Po nastavení všech parametrů PLC uživatel vstoupí do vývojového prostředí CoDeSys. V ABB Configurator se při zakládání projektu vybírá typ PLC a dále zde jsou volby nastavení CPU parametrů, integrovaných vstupně/výstupních jednotek, přídavných vstupně/výstupních jednotek, rozhraní COM1 a COM2 (je-li připojen) a Ethernetu. [4]

3.2.2 CoDeSys



Obrázek 5 - CoDeSys

CoDeSys (Controlled Development System) byl vyvinut firmou 3S-Smart Software Solutions GmbH. Základem tohoto prostředí je textový editor a debugger. Velkou výhodou je, že program nabízí programování a odzkoušení programu v simulovaném módu bez nutnosti fyzické účasti PLC. Dále se zde dají velmi jednoduše tvořit grafické vizualizace. V tomto vývojovém prostředí se programuje dle normy IEC 61131-3. Tato norma obsahuje 5 programovacích jazyků – Instruction List (IL), Structured Text (ST), Sequential Function Chart (SFC), Function Block Diagram (FBD), Ladder Diagram (LD) a navíc v CoDeSys můžeme použít Continuous Function Chart (CFC). Tyto jazyky se dají navzájem kombinovat za pomoci standardů normy IEC 61131-3. Tyto standardy jsou Struktura (datové typy), POU (Program Organisation Unit) a Globální proměnné. [16, str. 49-51]

II. PRAKTICKÁ ČÁST

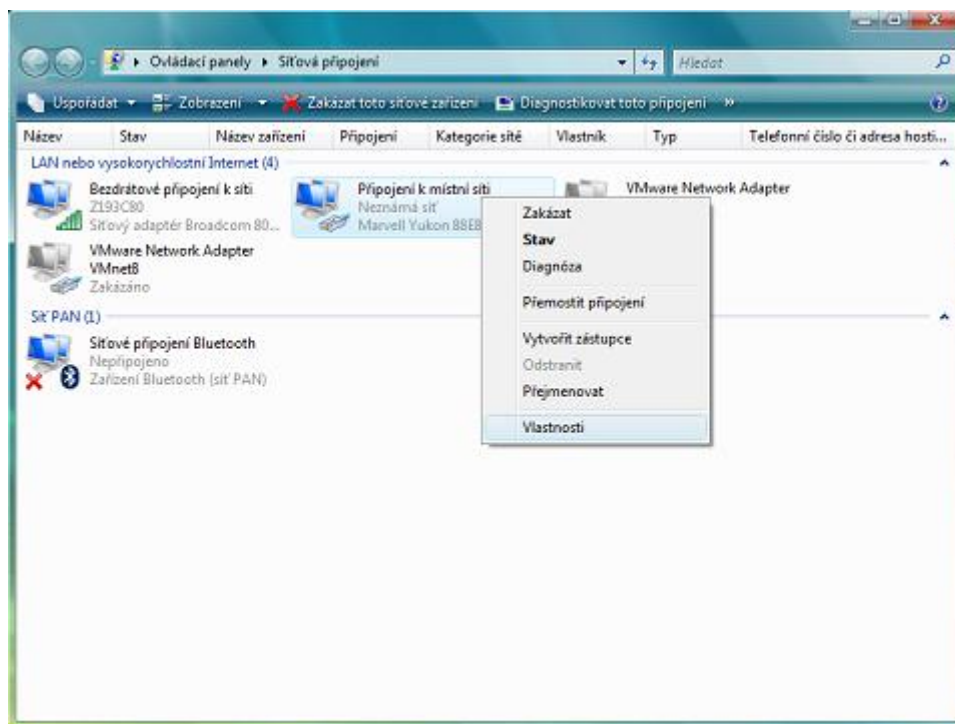
4 NÁVOD NA TVORBU ŘÍDICÍHO SOFTWARE

4.1 Připojení a nastavení komunikace pro účely programování

Automat lze programovat pomocí převodníku RS485 na USB nebo pomocí Ethernetu.

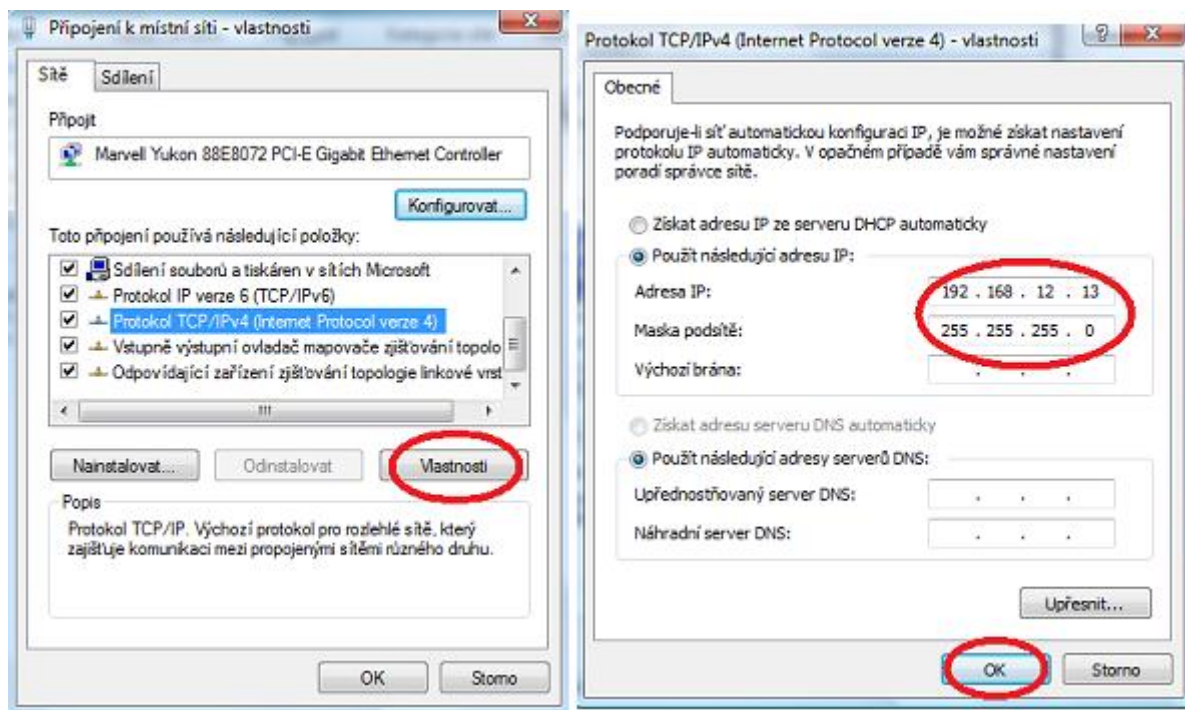
V tomto návodu bude popsán postup programování pomocí Ethernetu.

1. Zkontrolujte, zda PLC je napájeno 24 V DC a toto napětí je přivedeno na svorky L+ a M v sekci 24 VDC IN (na L+ kladnou svorku napájení a na M zápornou). Zapnuté PLC značí zelená kontrolka s popisem PWR na PLC.
2. Připojte ethernetový patch kabel do PLC a do počítače nebo routeru. (Pokud se připojujete ke staršímu počítači, u kterého síťová karta neumí automatické křížení, musíte použít křížený patch kabel.)
3. Pokud jste připojili PLC k routeru, pokračujte bodem 4. Pokud jste se připojili přímo k počítači, postupujte dle těchto pokynů:
 - a) Nabídka Start -> Ovládací panely -> Centrum sítí a sdílení -> Spravovat síťová připojení. (Tento postup platí pro Windows Vista)
(Pro Windows 7 platí tento postup: Nabídka Start -> Ovládací panely -> Síť a Internet -> Centrum síťových připojení a sdílení -> Změnit nastavení adaptéru.)
 - b) Levým tlačítkem klikněte na Připojení k místní síti a vyberte položku vlastnosti.



Obrázek 6 - Popis nastavení komunikace mezi PLC a PC

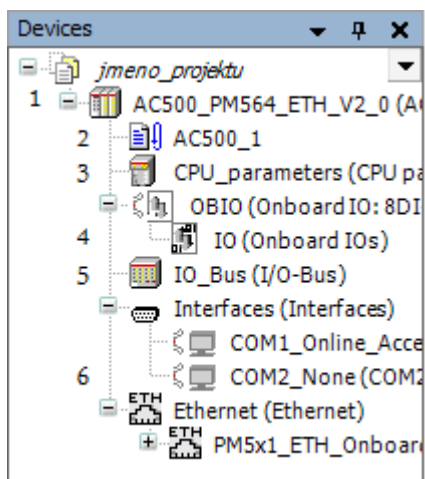
- c) Ze seznamu zvolte Protokol TCP/IPv4 (Internet Protocol verze 4) a klikněte na tlačítko vlastnosti.
- d) V nově otevřeném okně zvolte možnost - Použít následující adresu IP a do textových polí doplňte IP adresu (např. 192.168.12.13) a masku podsítě (např. 255.255.255.0). Potvrďte obě otevřená okna.



Obrázek 7 - Okna nastavení z bodu c a d

4. Otevřete program ABB Configurator. Zde zvolte záložku Tools -> IP config tool a klikněte na tlačítko Scan. Zde se Vám objeví všechna PLC, která jsou k dispozici. (V tomto okně je možné nastavit IP adresu PLC – po kliknutí na PLC ze seznamu se objeví formulář s kolonkou pro IP adresu, tu přepíšete, zvolíte Send a poté znovu načtete seznam PLC tlačítkem Scan.)

4. V levé části obrazovky se projekt přidá do sloupce Devices – zde můžete projekt rozkliknout a poklikáním na jeho části, které se budou otevírat ve formě oken, v nich můžete provést úpravu a nastavení parametrů konfigurace PLC.



Obrázek 10 - Postranní lišta Devices s očíslovanými částmi pro usnadnění popisu

4.2.1 Popis projektu v postranní liště Devices

Po kliknutí na „AC500_PM564_ETH_V2_0“ (1) se Vám zobrazí informace o PLC. Kliknutím na „AC500_1“ (2) se přesunete do vývojového prostředí CoDeSys. V části „CPU_parameters“ (3) můžete nastavovat konfigurační parametry CPU. V části „IO (Onboard IOs)“ (4) se nacházejí v nově otevřeném okně dvě záložky. V první záložce „8DI+6DO+2AI+1A0 Configuration“ můžete nastavit volbu vstupů a výstupů (například nastavení režimu analogového vstupu). V druhé záložce „8DI+6DO+2AI+1A0 I/O Mapping“ můžete nastavit symbolická jména vstupů a výstupů, která Vám při programování zjednoduší a zpřehlední práci. Nastavíte je poklikáním na příslušný řádek a napsáním jména, které chcete používat.

Variable	Mapping	Channel	Address	Type	Unit	Description
Digital + analog inputs						
		Inputs 0...	%IB4000	BYTE		Digital input...
vstup1		Input 0	%IX40...	BOOL		Input 0
vstup2		Input 1	%IX40...	BOOL		Input 1
		Input 2	%IX40...	BOOL		Input 2
		Input 3	%IX40...	BOOL		Input 3
		Input 4	%IX40...	BOOL		Input 4
		Input 5	%IX40...	BOOL		Input 5
		Input AI0	%IX40...	BOOL		Digital input...
		Input AI1	%IX40...	BOOL		Digital input...
		Interrupt	%IB4001	BYTE		Interrupt
		Analog i...	%IW20...	INT		Analog inp...
		Analog i...	%IW20...	INT		Analog inp...
PWM						
Fast counter						
Digital + analog outputs						
		Outputs...	%QB40...	BYTE		Digital outp...
vystup1		Output 0	%QX40...	BOOL		Output 0
		Output 1	%QX40...	BOOL		Output 1
		Output 2	%QX40...	BOOL		Output 2
		Output 3	%QX40...	BOOL		Output 3
		Output 4	%QX40...	BOOL		Output 4

Obrázek 11 - Mapování vstupů a výstupů

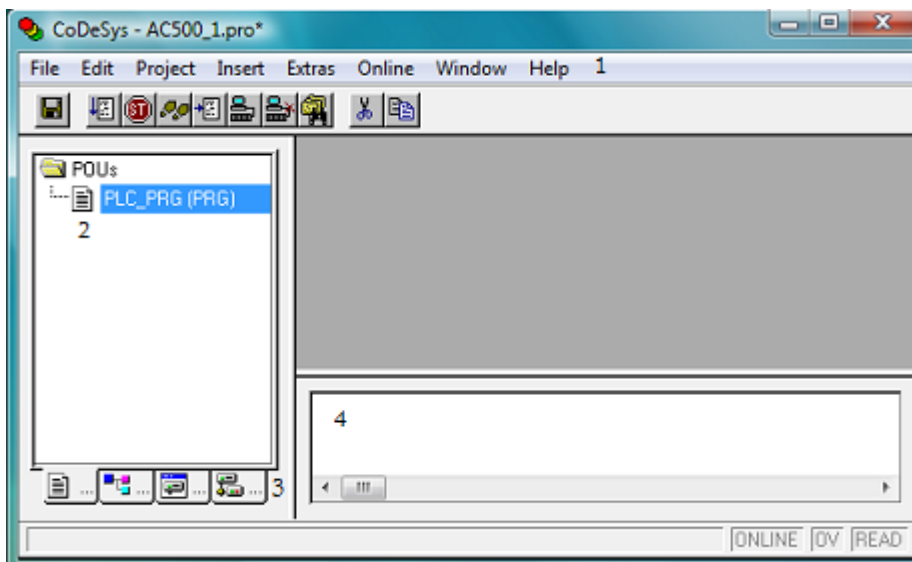
Po kliknutí levým tlačítkem na „IO_Bus(I/O-Bus)“ (5) a výběru položky Add device můžete k PLC přidat přídavné moduly. Po kliknutí na přidávaný modul vstupů nebo výstupů, můžete tyto vstupy a výstupy namapovat. Mapují se po otevření okna, které se objeví poklikáním na přidávaný model v záložce Mapping stejným způsobem jako v části IO (Onboard IOs).

Často můžete potřebovat využít obvodu reálného času, proto musíte v sekci Interface kliknout levým tlačítkem na „COM2“ a zde dát možnost Plug Device. V nově otevřeném okně vyberete „COM2-MODBUS“, potvrdíte a toto okno zavřete pomocí Close.

- Po nastavení všech parametrů, které požadujete, se po kliknutí na „AC500_1“ (2) otevře vývojové prostředí CoDeSys, ve kterém budete PLC programovat.

4.3 Nastavení a vývoj programu v CoDeSys

4.3.1 Popis programu CoDeSys



Obrázek 12 - Popis vývojového prostředí CoDeSys

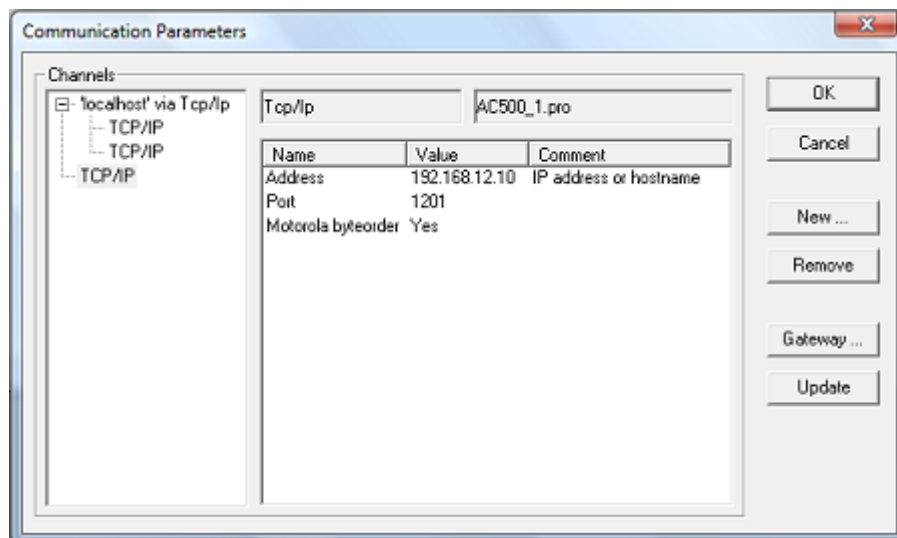
Po zapnutí CoDeSys si můžete všimnout v horní části okna panelu záložek (1). Tyto záložky ukrývají nabídky, které tvoří ovládání celého programu. V levé dolní části okna se nacházejí záložky POU's, Data types, Visualizations a Resources (3). Levá část (2) slouží k prohlížení dolních záložek. V dolní části (4) se objevuje stav překladače a chyby, jež byly odhaleny.

4.3.1.1 Nastavení komunikace

V CoDeSys můžete volit programování ve dvou režimech – v simulovaném a s připojeným PLC. Pro simulovaný režim zvolte v záložce Online volbu Simulation mode.

Pokud chcete pracovat s reálným PLC, postupujte následovně:

1. V záložce Online zvolte Communication Parameters.
2. V nově otevřeném okně klikněte na tlačítko New a vyberte položku Tcp/Ip.
3. U nově přidaného připojení vyplňte kolonku Address dle IP adresy PLC, ke kterému se chcete připojit. Dále nastavte port na 1201 a na Motorola Byteorder zvolte Yes.



Obrázek 13 - Okno, ve kterém se nastavují komunikační parametry

4.3.1.2 PLC Browser

Tento nástroj najdete po kliknutí na dolní záložku Resources. Pomocí něho můžete například zjistit chyby (příkaz diagshow all), které PLC obsahuje a také je vymazat (diagreset). Seznam dalších užitečných funkcí získáte poklikáním na tři tečky u textového pole, do kterého se zadávají příkazy.

4.3.1.3 Tvorba programu

Program tvoříte v okně PLC_PRG, které představuje POU typu program v jazyku Strukturovaného textu. Toto okno je rozdělené na část deklaraci a kódovou. Pokud byste chtěli tvořit program v jiném jazyce, přeložte projekt, levým tlačítkem myši na něj klikněte a vyberte volbu Convert object. V okně si zvolte ze tří jazyků (IL, FBD, LD) a potvrďte.

V programu můžete vytvářet své vlastní POU typu funkce, funkčního bloku a programu ve všech jazycích normy plus jazyku CFC. Globální proměnné jsou definované do samostatného listu, který najdete v dolní záložce Resources ve složce Global Variables. V tomto programu můžete využít už předprogramovaných POU, které najdete v dolní záložce Resources po poklikání na Library Manager. Použití těchto POU z knihovny je jednoduché - stačí je inicializovat stejným způsobem jako Vaše vlastní vytvořená POU.

Po napsání programu program přeložte kliknutím na záložku Project a v ní zvolte Build (klávesa F11). Po přeložení programu spusťte tento program v PLC nebo v simulaci pomocí záložky Online a volby Login. V tento okamžik je program připravený na spuštění a spustíte ho přes záložku Online a volbu Run (klávesa F5).

Pro úpravu programu se musíte odhlásit pomocí záložky Online a volby Logout.

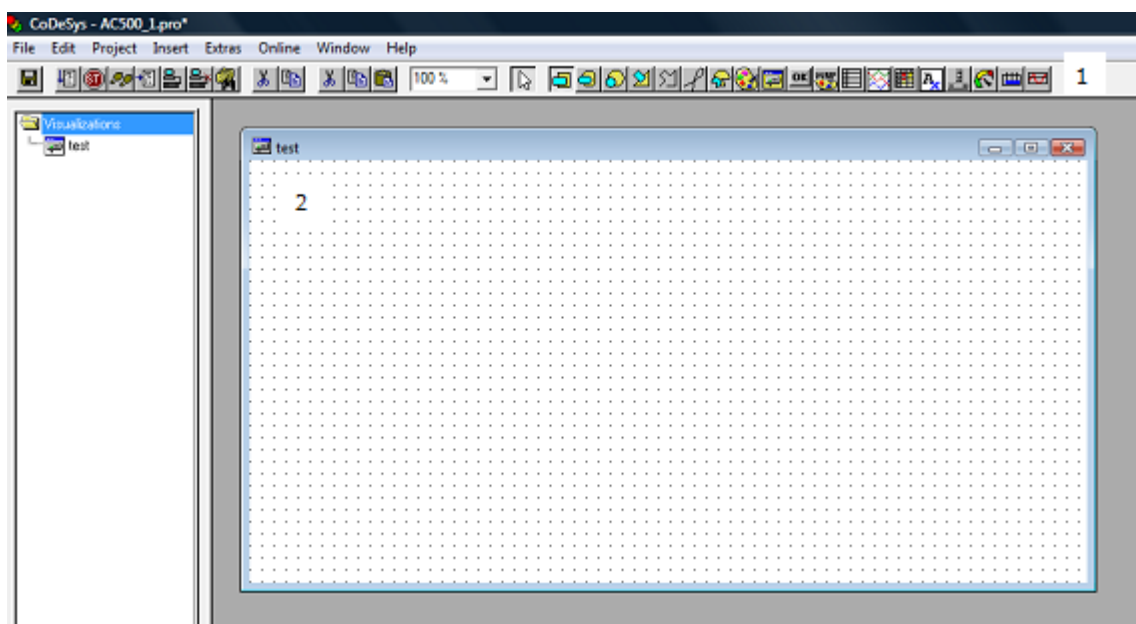
V záložce Online dále naleznete volbu Create boot project, která Vám umožní nahrát Váš program do PLC a vždy po spuštění PLC se tento program automaticky spustí. Pokud budete chtít v programu nastavovat proměnné při jeho chodu, musíte změnit proměnnou poklikáním na příslušné jméno proměnné a zvolit záložku Online a položku Write Values.

4.3.1.4 Tvorba simulace

V CoDeSys můžete vytvářet simulace, které je možné využít k řízení a sledování řízených objektů.

Postup tvorby simulace:

1. V CoDeSys přepněte dolní levou záložku na Visualizations.
2. Do této záložky klikněte levým tlačítkem a zvolte Add object.
3. Otevře se Vám okno, které požaduje jméno vizualizace. Jméno napište a potvrďte.



Obrázek 14 - Prostředí pro tvorbu simulací

4. Otevře se Vám okno (2), do kterého můžete táhnutím umísťovat různé komponenty z nově vzniklé nabídky komponentů v horní části okna (1).
5. Po natáhnutí požadovaných komponentů na ně klikněte a nastavte jejich vlastnosti.

Tvorba základních prvků simulací:

1. V programu máte globální proměnnou typu bool (možné stavy true nebo false) a chcete tuto proměnnou měnit stiskem tlačítka. Postup bude následovný:

- a) Natáhněte si tlačítko na formulář.
- b) Klikněte na něj a otevře se Vám okno s nastavením.
- c) V nastavení v sekci Category klikněte na Input, zde zatrhnete „Toggle variable“ a do textového pole zapište název proměnné.

2. V programu chcete zobrazit stav proměnné změnou barvy kruhu z modré na červenou.

- a) Natáhněte si prvek Ellipse.
- b) Klikněte na něj.
- c) V sekci Category zvolte Colors – zde nastavte modrou barvu v podsekci Color a červenou v podsekci Alarm color.
- d) Přepněte v sekci Category do položky Variable a zde nastavte jméno proměnné v kolonce s názvem Change color.

5 MODEL PŘEČERPÁVACÍ ELEKTRÁRNY

5.1 Přečerpávací elektrárna

O využití energie vody v přírodě se lidé snaží už od starověku. Ve středověku se energie vody požívala v kovárnách na pohon strojních kladiv, mlýnech a třeba na pilách. Koncem osmnáctého století se využití vody stalo efektivnějším díky vynálezu vodní turbíny. Vodní turbíny využívají měrnou energii vody, a to buď kinetickou nebo tlakovou, s velkou účinností. Nejvýznamnějšími druhy turbín jsou Peltonova, Francisova a Kaplanova turbína. Tyto turbíny se v jistých modifikacích využívají dodnes ve vodních elektrárnách. [6], [8]

Přečerpávací elektrárna je speciální typ vodní elektrárny, která pro svůj chod vyžaduje dvojici nádrží. Jedna je situovaná ve vyšší nadmořské výšce tak, aby voda proudící do níže položené nádrže měla potřebný spád pro výrobu elektrické energie. Přečerpávací elektrárny nabývají v poslední době na významu, poněvadž sice přibývá zdrojů elektrické energie, většina z nich však nedokáže pružně reagovat na momentální potřebu energie v elektrické síti. Přečerpávací elektrárna zastává důležitou funkci v rámci elektrické sítě. Přes den, když je nedostatek elektrické energie, elektrárna ji dodává do sítě a v noci, kdy jí už není potřeba, je energie využita k přečerpání vody z dolní nádrže do horní. Přečerpávací elektrárna tedy představuje možnost akumulovat energii, která může být téměř okamžitě použita. [7], [18]

V České Republice fungují tři přečerpávací elektrárny. Největší z nich je elektrárna Dlouhé Stráně, nacházející se u obce Loučná nad Desnou v Jeseníkách. V této elektrárně je použita reverzní Francisova turbína, díky níž elektrárna disponuje výkonem 2 x 325 MW, čímž se stává nejvýkonnější vodní elektrárnou v České Republice. [7]

Principem fungování přečerpávacích elektráren jsem se inspiroval pro výrobu jednoduchého modelu, na kterém chci demonstrovat funkce a vlastnosti programovatelného automatu od firmy ABB typu PM564-T-ETH.

5.2 Popis modelu

Model je tvořen dvěma nádržemi a dvojím potrubím. V jednom potrubí se nachází klapka a turbína. V druhém potrubí je čerpadlo. V každé nádrži jsou dvě čidla signalizující výšku hladiny – jedno pro minimální hladinu a druhé pro maximální hladinu.



Obrázek 15 - Fotografie modelu přečerpávací elektrárny

V modelu jsou akční členy klapka, turbína a čerpadlo nahrazeny LED diodami zelené barvy. Funkci snímačů hladiny plní páčkové spínače. Ovládání je realizováno pomocí trojice páčkových spínačů a u nich je trojice LED diod, které signalizují stav spínačů. Zatížení elektrické sítě se simuluje otočným potenciometrem. Pro efekt tekoucí vody je zde sestava modrých diod, které při otevřené klapce a zapnutém čerpadle simulují proud tekoucí vody.

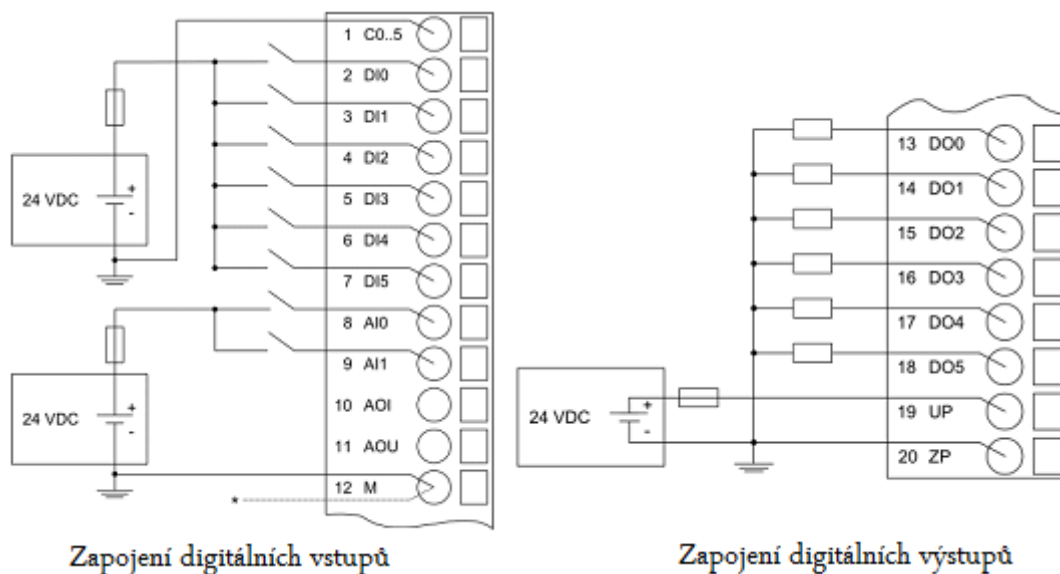
Tabulka 3 - Seznam vstupů a výstupů modelu

Typ:	I/O	V modelu představuje:
digitální	vstup	horní spínač horní nádrže
digitální	vstup	dolní spínač horní nádrže
digitální	vstup	horní spínač dolní nádrže
digitální	vstup	dolní spínač dolní nádrže
digitální	výstup	klapka + tekoucí voda
digitální	výstup	turbína + tekoucí voda
digitální	výstup	čerpadlo
digitální	výstup	signalizace režimu
digitální	výstup	signalizace výroby el. energie
digitální	výstup	signalizace čerpání vody
analogový	vstup	ovládací panel se 3 spínači
analogový	vstup	potenciometr - zatížení sítě

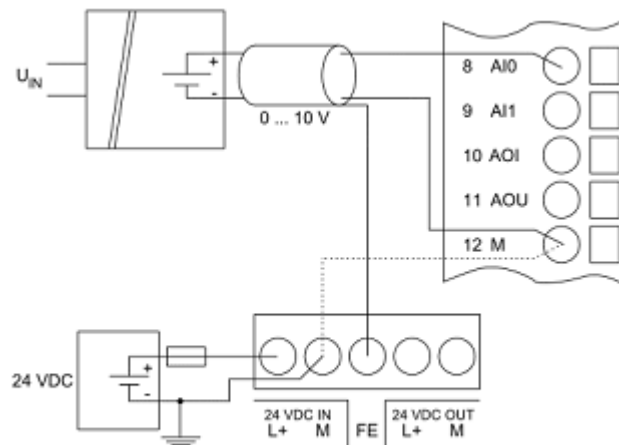
5.2.1 Elektrická zapojení použitá v modelu

5.2.1.1 Zapojení doporučená výrobcem

Programovatelný automat PM564-T-ETH má pro své vstupy a výstupy tato doporučená zapojení, která byla použita při tvorbě modelu:



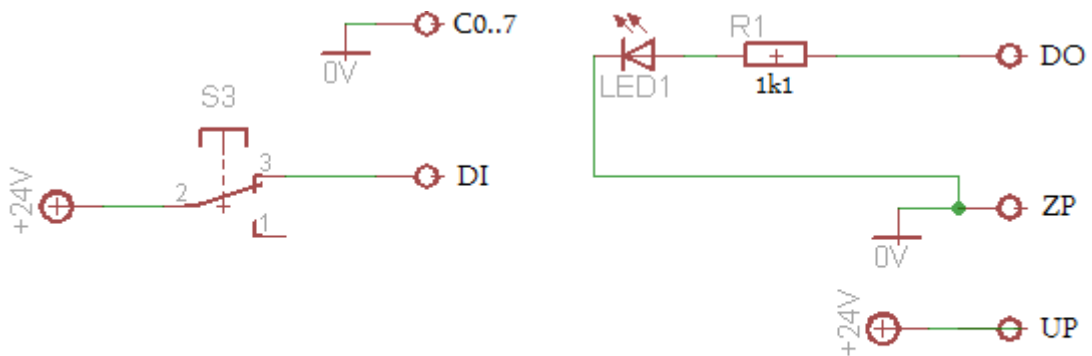
Obrázek 16 - Základní zapojení doporučená výrobcem pro digitální vstupy a výstupy



Zapojení analogového vstupu s aktivní zátěží

Obrázek 17 - Základní zapojení doporučené výrobcem pro analogový vstup s aktivní zátěží

5.2.1.2 Zapojení digitálních vstupů a výstupů



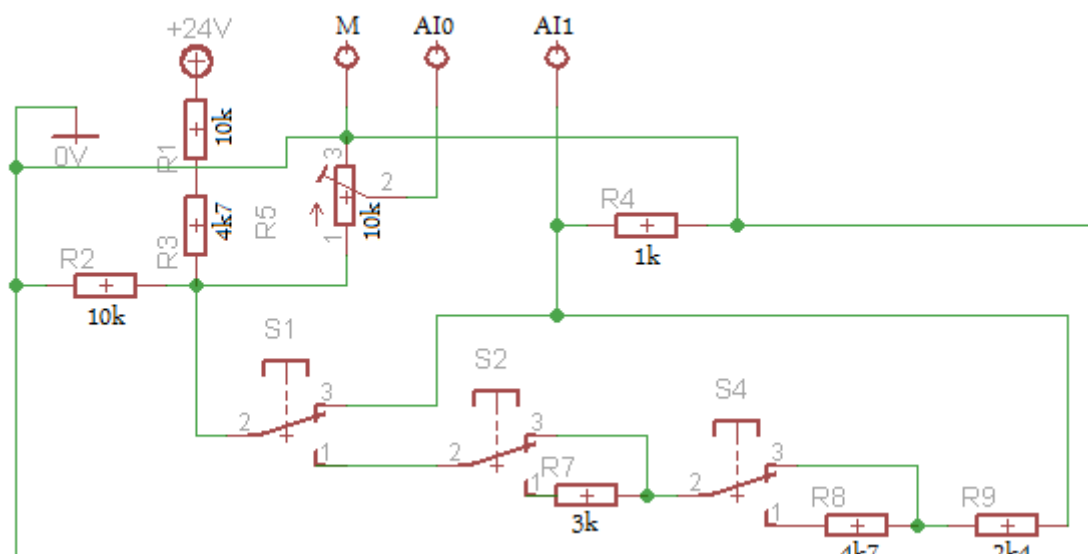
Zapojení digitálních vstupů

Zapojení digitálních výstupů

Obrázek 18 - Zapojení digitálních vstupů a výstupů

Podle schématu na obrázku 18 – zapojení digitálních vstupů, jsou zapojena čidla výšky hladiny, která jsou realizovaná páčkovým spínačem. Dle schématu zapojení digitálních výstupů jsou zapojeny klapka, turbína, čerpadlo a trojice signalizačních kontrolky. Tyto členy jsou tvořeny předřadným odporem a LED diodami zelené barvy.

5.2.1.3 Zapojení analogových výstupů



Obrázek 19 - Zapojení analogových výstupů

Schéma na obrázku 19 se skládá ze dvou obvodů, které jsou díky napětíovému děliči napájeny napětím o velikosti 10 V. První obvod je tvořen otočným potenciometrem, na němž měříme napětí vzhledem k pozici jezdce. Druhý obvod byl sestaven z důvodů, že PLC má k dispozici pouze šest digitálních vstupů a k ovládání modelu je zapotřebí o jeden vstup více. Díky tomuto zapojení je využit pouze jeden analogový vstup PLC. Princip tohoto zapojení spočívá ve změně poměru napětí mezi odpory zapojenými do série a odporem, na kterém je napětí měřeno. Odpory, jež jsou zapojené do série, jsou závislé na poloze spínače – buď může být spínač sepnutý (poloha 1), čímž dojde k přidání odporu do série, anebo rozepnutý (poloha 3), což způsobí přemostění odporu. Tabulka 4 shrnuje možné stavy při kombinaci spínačů, přičemž 0 znamená rozepnutý spínač a 1 značí sepnutý spínač.

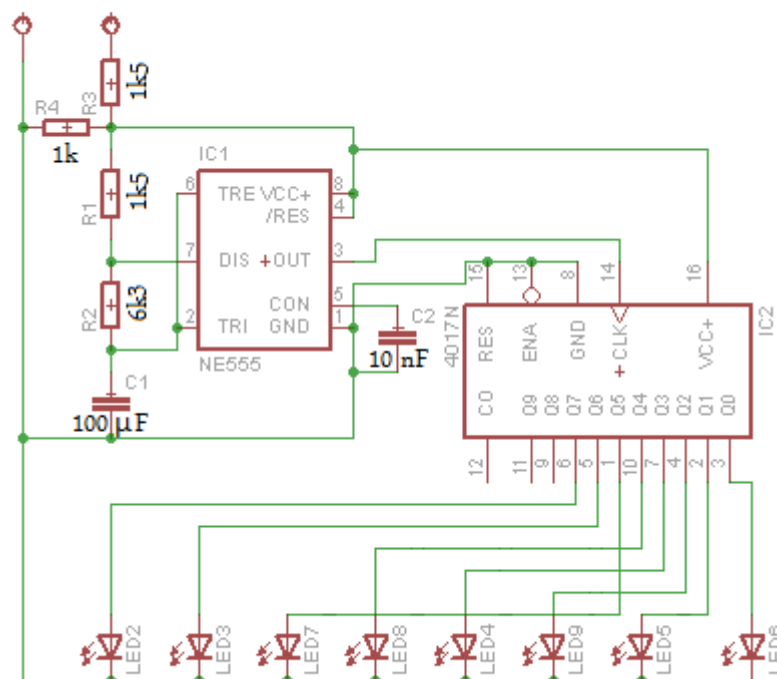
Tabulka 4 - Kombinace možných stavů spínačů v obvodě pro analogové vstupy

spínač 1	spínač 2	spínač 3	stav
0	0	0	A
0	1	0	A
0	0	1	A
0	1	1	A
1	0	0	B
1	1	0	C
1	0	1	D
1	1	1	E

Na výstupu obvodu můžeme tedy naměřit 5 různých stavů (A-E). Využití těchto stavů provedeme programově.

Výstupy obou obvodů jsou tvořeny třemi svorkami - dvěma samostatnými a jednou společnou. Obě dvě samostatné svorky jsou zapojeny každá do jednoho analogového vstupu PLC. Společná svorka se zapojuje do vstupu M. Měřené napětí se vždy nachází mezi samostatnou a společnou svorkou.

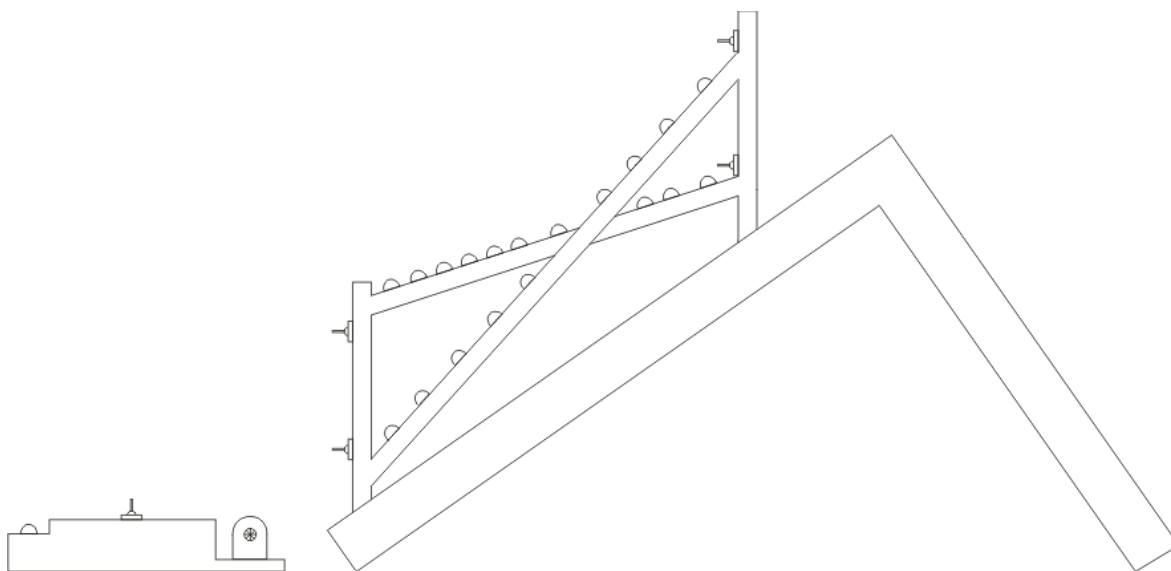
5.2.1.4 Zapojení pro obvod proudu vody



Obrázek 20 - Zapojení pro efekt tekoucí vody

Toto zapojení využívá integrovaného obvodu NE555, jenž je zapojen jako astabilní multivibrátor generující impulzy. Princip tohoto zapojení je, že na výstupu je hodnota logické jedničky a zároveň se přes odpory nabíjí kondenzátor. Po nabití kondenzátoru na $\frac{2}{3}$ napájecího napětí se změní výstup na logickou nulu a kondenzátor se vybíjí. Po vybití na $\frac{1}{3}$ napájecího napětí se tento cyklus znovu opakuje. Frekvence impulzů lze nastavit podle parametrů součástek – odporů R1 a R2 a kondenzátoru C1. [5], [9]

Druhým obvodem použitým v zapojení je integrovaný obvod 4017, který je známý jako Jahnsonův čítač. Když na jeho vstup přivedeme logický signál, na jeho výstupech Q0-Q9 se bude postupně objevovat logická jednička. Tohoto principu využijeme při připojení LED diod, které se budou postupně rozsvěcovat a zhasínat, což bude simulovat tok vody. [9]



Obrázek 21 - Schéma modelu přečerpávací elektrárny

5.2.2 Model přečerpávací elektrárny

Model se skládá ze dvou dřevěných desek, které jsou spojeny tak, aby vytvářely svah. Na svahu jsou umístěny dvě nádrže vyrobené z modelářské překližky. Na stěně každé z nádrží se nachází dvojice spínačů. Nádrže jsou spojeny potrubím, které je také vyrobeno z překližky. K ovládání slouží osazená deska plošných spojů, jež je umístěná v ochranném obalu vyrobeném z překližky a kartonu.

5.3 Programování modelu

5.3.1 Požadavky na řízení modelu

Program umožní řídit elektrárnu dvěma způsoby. Prvním způsobem bude manuální ovládání a druhým automatický chod přečerpávací elektrárny. Mezi těmito způsoby se bude dát přepínat prvním páčkovým spínačem na ovládacím panelu – stav bude signalizován LED diodou u spínače.

V prvním způsobu řízení – manuálním režimu - bude možno pomocí 2 spínačů na ovládacím panelu ovládat vyrábění elektrické energie turbínou a přečerpávat vodu z dolní nádrže do horní. V případě mezních stavů nádrží – výšky hladiny nad horní čidlo nebo pod dolní čidlo u jedné z nádrží se automaticky přeruší výroba el. energie nebo přečerpávání vody. Čerpání a vyrábění el. energie budou signalizovat LED diody u spínačů.

V druhém způsobu řízení – automatickém režimu - bude řízení plně automatické a bude reagovat na stav el. sítě v závislosti na množství vody v nádržích.

Pro manuální i automatický režim bude vytvořena simulace, která bude ukazovat vytíženost elektrické sítě, stav nádrží a stav akčních členů a signalizací.

5.3.2 Programové řešení

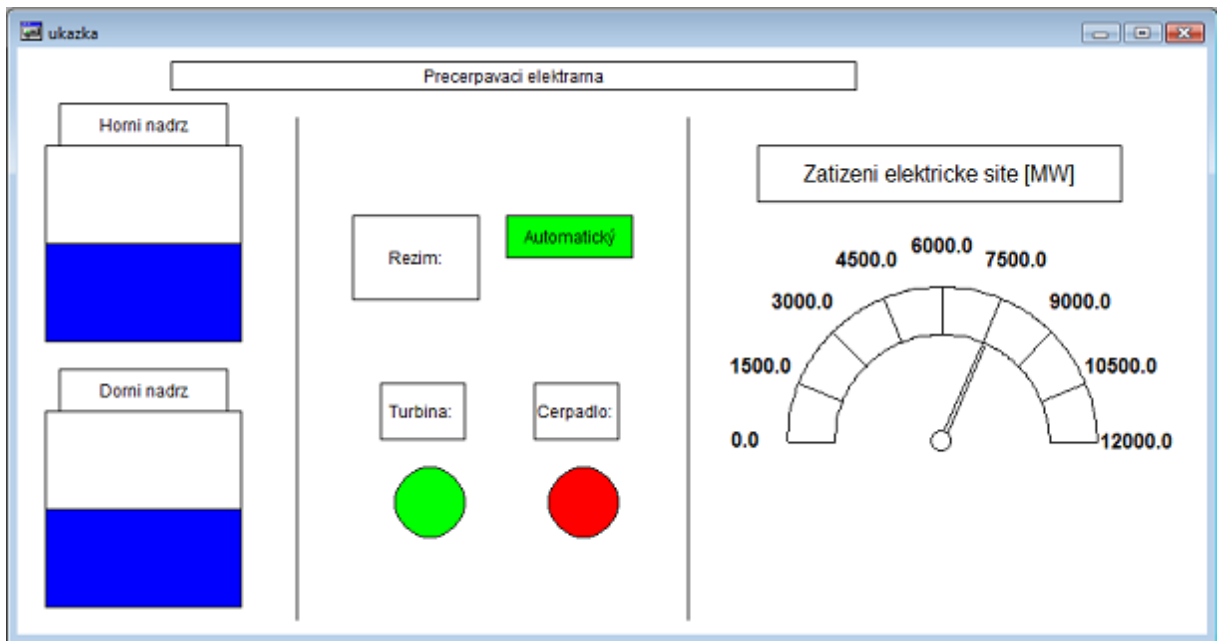
Program pomocí podmínky if rozdělíme na dvě části – první část bude automatický režim a druhá část manuální režim.

Automatický režim vytvoříme tak, že podle stavu elektrické sítě (hodnoty potenciometru) podmínkou rozdělíme tuhle část programu na výrobu elektrické energie a na přečerpávání vody. Parametrem rozhodování bude stav čidel v nádržích.

V manuálním režimu musíme nejdříve zpracovat hodnotu z analogového vstupu. Zpracování probíhá tak, že hodnota analogového vstupu je porovnávána s krajními hodnotami intervalů a po nalezení správného intervalu je nastaven výstupní stav. Možné výstupní stavy jsou vyrábí/nevyrábí el. energii a čerpá/nečerpá vodu z dolní nádrže do horní. Následně jsou podmínkami ošetřeny stavy nádrží podle čidel.

5.3.3 Vizualizace

Vizualizace byla vytvořena pro možnost sledování činnosti přečerpávací elektrárny z monitoru počítače. Obsahuje horní a dolní nádrž, která dle čidel ukazuje množství vody v nádržích. Dále informuje o aktivním režimu a stavu výroby elektrické energie pomocí turbíny a přečerpávání vody pomocí čerpadla. Poslední částí simulace je stupnice, která ukazuje aktuální zatížení sítě.



Obrázek 22 - Simulace přečerpávací elektrárny

Zdrojový kód je v příloze.

6 VZOROVÉ ÚLOHY

6.1 Program č. 1 – LED diody a tlačítka

6.1.1 Potřebné vybavení

- modul tlačítek EDU-mod
- 4 LED diody (např. ze signalizace výstupů PLC)

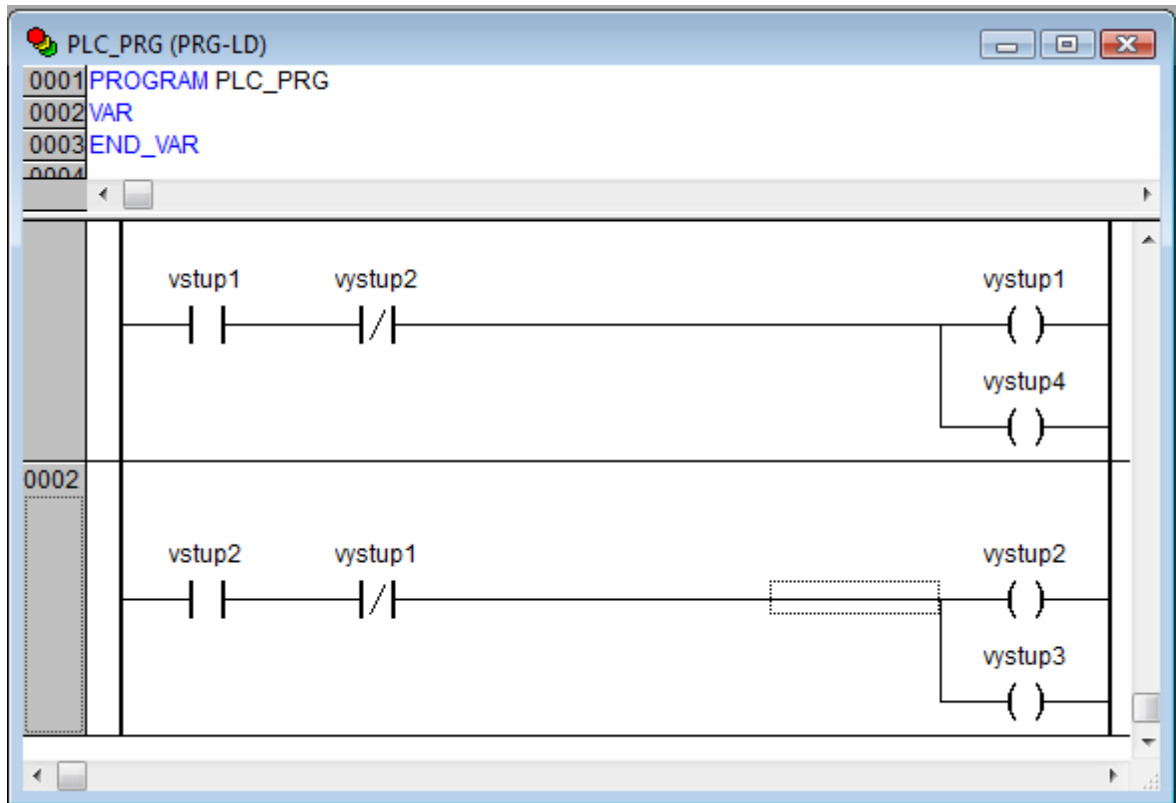
6.1.2 Zadání

Naprogramujte program, který po zapnutí prvního tlačítka rozsvítí 2 krajní LED diody a opět je po vypnutí tlačítka zhasne. LED diody uprostřed se zapínají a vypínají druhým tlačítkem. Když krajní LED diody budou zapnuté, tak nepůjdou zapnout LED diody uprostřed a naopak, když budou zapnuty LED diody uprostřed, tak nepůjdou zapnout LED diody na kraji.

6.1.3 Řešení

Nejdříve spustíme konfiguraci a CoDeSys, kde zjistíme adresy vstupů a výstupů, které chceme využít. V konfiguraci v sekci mapování nastavíme jména pro vstupy – tlačítka a pro výstupy – LED diody. V CoDeSys přeložíme projekt (F11) a změním typ POU – levým tlačítkem myši klikneme na POU v levé části obrazovky a vybereme volbu „Convert object“, zde zvolíme volbu LD a potvrdíme OK.

Program vytvoříme ve dvou oddělených větvích – první větev bude obsahovat spínací kontakt prvního tlačítka a rozpínací kontakt výstupní LED diody. Na konci budou paralelně zapojené dvě výstupní cívky, které představují výstupy – LED diody. Druhá větev bude stejná dle prvků, ale lišit se bude v použitých proměnných.



Obrázek 23 - Schéma programu

6.2 Program č. 2 – Křižovatka s dvojím režimem

6.2.1 Potřebné vybavení

- modul křižovatky EDU-mod
- modul tlačítek EDU-mod

6.2.2 Zadání

Naprogramujte program, který bude řídit trojici semaforů. První semafor (na hlavní silnici) bude mít zelenou po dobu 15 sekund, druhý semafor po dobu 10 sekund a třetí semafor (pro chodce) po dobu 5 sekund. Na semaforech nesmí být nikdy zelená ve stejnou dobu. Od 22 do 6 hodin bude aktivní režim noc – na semaforech bliká jen oranžové světlo. Režim noc v případě potřeby půjde zapnout a vypnout tlačítkem nezávisle na hodině.

6.2.3 Řešení

Nejdříve spustíme konfiguraci a CoDeSys, kde zjistíme adresy vstupů a výstupů, které chceme využít. Poté namapujeme proměnné – jedno tlačítko a osm LED diod a přepneme do CoDeSys. Zde založíme nové POU typu funkčního bloku pro denní režim, v němž vytvoříme instanci časovače TON a pomocnou proměnnou tlacitko1 a začneme psát kód. V kódu zavoláme časovač, u kterého nastavíme rozběh a konečný čas časování. Poté pomocí sekvence if a elsif postupně nastavujeme časová rozmezí a svícení světél semaforu. Nakonec sestavíme podmínku if, která nám vynuluje časovač. Na začátek kódu doplníme dvě podmínky pro nový rozběh časovače tak, aby po každém přepnutí z nočního na denní režim začínal časovač čítat od začátku a tím i sekvence semaforu šla od začátku.

Dále založíme nové POU typu funkčního bloku pro noční režim. Postupujeme prakticky stejně. Jediným rozdílem je, že nedáváme podmínku pro začátek sekvence od začátku (v nočním režimu bliká pouze oranžové světlo).

V hlavním programu vytvoříme instanci funkčního bloku CLOCK, který nám zprovozní hodiny reálného času. Dále uděláme instanci funkčních bloků, které jsem vytvořili a nakonec definujeme proměnnou hodina. V kódové části zavoláme funkční blok hodin reálného času i s nastavením času a data. Nastavíme aktuální hodinu z funkčního bloku do proměnné hodina. Vytvoříme podmínku pro případ zapnutého tlačítka, kterým se aktivuje noční režim. Pokud je tlačítko vypnuté, režim semaforů se řídí dle aktuální hodiny. V této podmínce se volají funkční bloky pro denní a noční režim.

Nakonec vytvoříme globální proměnnou pomocná, pomocí které budeme zajišťovat, aby semaforey v denním režimu fungovali od začátku sekvence (tuto proměnnou budeme nastavovat v hlavním programu v podmínkách a ve funkčním bloku pro denní režim ji budeme využívat).

Zdrojový kód je k nalezení v příloze.

6.3 Program č. 3 - Ovládání pračky

6.3.1 Potřebné vybavení

- modul pračky EDU-mod

6.3.2 Zadání

Naprogramujte program, který bude ovládat model pračky.

Prací program po startu začne napouštět vodu a topit na zadanou teplotu. Po napuštění vody a jejím ohřátí na požadovanou teplotu se začne točit buben 20 sekund doprava a následně 20 sekund doleva. Poté se vypustí voda a pračka se nechá zchladnout. V poslední fázi – odstředění, se zapnou otáčky a buben se bude točit 5 sekund doprava a 5 sekund doleva.

K ovládání vytvořte vizualizaci, ve které si uživatel zvolí teplotu praní s indikátorem zvolené teploty. Dále pak tlačítko, kterým se odstartuje praní a signalizaci, jež upozorní na to, že se pere.

6.3.3 Řešení

Nejdříve spustíme konfiguraci a CoDeSys, kde zjistíme adresy vstupů a výstupů, které chceme využít. V konfiguraci si namapujeme vstupy a výstupy. Poté přepneme do CoDeSys a založíme nové POU typu funkční blok.

Program bude mít následující posloupnost: start -> topení a napouštění vody -> praní -> vypouštění vody, chladnutí, odstředění vody -> konec.

Pro napouštění vody vytvoříme podmínku, abychom nepřesáhli hladinu 100 procent a topili jen na začátku pracího cyklu. Pro topení taktéž vytvoříme podmínku, která bude mít za úkol dosáhnout potřebné teploty a regulovat tuto teplotu, dokud se bude prát.

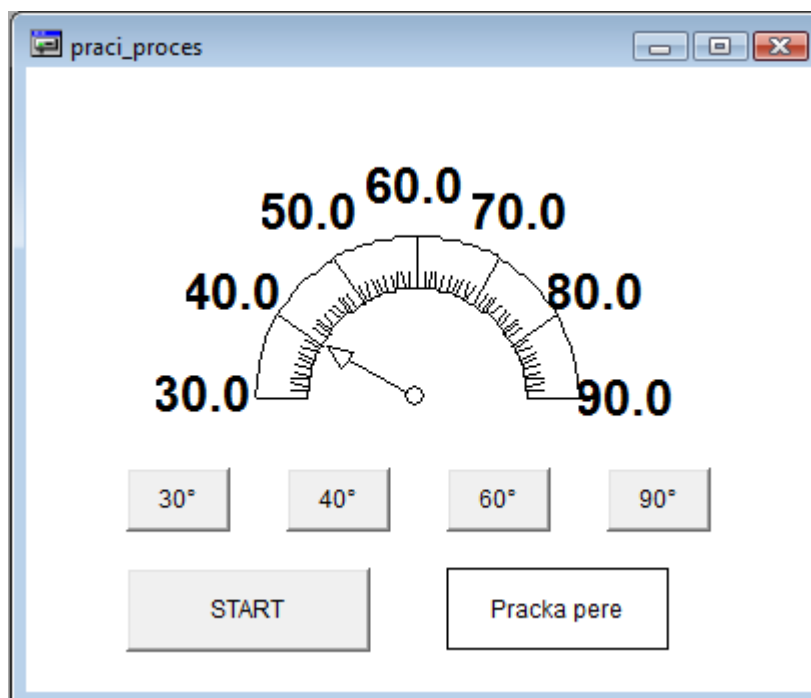
Část praní se skládá z podmínek, které nám s časovačem TON umožňují otáčet buben doprava a doleva v určitých časových intervalech. Po doprání se počká, až se vypustí voda a pračka zchladne – teprve poté může být spuštěna část odstředování. V ní opět využijeme časovače TON a budeme s bubnem točit 5 sekund doprava a 5 sekund doleva.

V hlavní části programu vytvoříme instanci právě vytvořeného funkčního bloku. V kódové části tento funkční blok zavoláme. Dále vytvoříme v listu globálních proměnných proměnné pro tlačítka a prvky, které budou použity v simulaci. V hlavním programu vytvoříme dělicí podmínku pro nastavení teploty praní ze simulace.

Nakonec vytvoříme vizualizaci – do ní natáhneme 5 tlačítek, prvek Meter a prvek Rectangle. Nastavíme tlačítka poklikáním na příslušné tlačítko a otevře se nám nabídka, v níž můžeme měnit jeho parametry – v kategorii Text napíšeme text, který se bude zobrazovat na tlačítku a v kategorii Input zatrhneme Tab variable a napíšeme název proměnné, kterou po stisku tlačítka budeme měnit.

Po kliknutí na prvek Meter zvolíme Variable/Scale a zde do textového pole Scale start uvedeme počáteční hodnotu a do Scale end konečnou hodnotu rozsahu, který se nám bude zobrazovat. V Main Scale určíme velikost dílčích rozsahů mezi jednotlivými hodnotami. Do textového pole Variable napíšeme název proměnné, která bude určovat hodnotu, jež chceme zobrazit.

Klikneme na prvek Rectangle a v sekci Text nastavíme zobrazovaný text. V sekci Variable nastavíme do textového pole Invisible proměnnou, která se bude měnit podle toho, zda budeme prát či nikoliv.



Obrázek 24 - Ukázka vizualizace ovládání pračky

ZÁVĚR

Tato bakalářská práce se zabývala především programovatelným automatem od firmy ABB, konkrétně typem PM564-T-ETH. Jejím cílem bylo vytvořit pracoviště obsahující tento automat pro potřeby výuky předmětu Programovatelné automaty.

Na začátku teoretické části se nachází obecné informace o programovatelných automatech a jejich rozdělení. Následuje popis mezinárodní normy o programování PLC. Dále jsou v teoretické části uvedeny informacemi o firmě ABB s přehledem a srovnáním jejích produktů. V poslední části teorie je automat PM564-T-ETH popsán po hardwarové stránce. Praktická část začíná manuálem, v němž je popsán celkový postup od zapojení PLC, přes nastavení komunikace a konfigurace, až po vývoj programu v CoDeSys. Dále následuje popsání modelu přečerpávací elektrárny a vytvoření programu pro jeho řízení. Praktická část je zakončena zadáním a vyřešením vzorových úloh, které mohou být použity při výuce.

Během prvních pokusů naprogramovat automat se objevil problém. PLC hlásilo sadu hardwarových chyb. Jak se ukázalo, za chyby mohl zastaratý firmware automatu, který znemožňoval jeho správné fungování. Problém byl vyřešen aktualizací firmwaru, kterou provedl pan Ing. Ivánek z firmy Konel s.r.o. Dalším problémem byl nedostatek digitálních vstupů PLC. Pro optimální řízení modelu přečerpávací elektrárny byly potřeba tři digitální vstupy, po zapojení všech základních obvodů však na PLC zbývaly pouze dva digitální vstupy. Proto bylo potřeba změnit původní koncept a navrhnout zapojení, při němž byl využit jen jeden analogový vstup k řízení modelu.

Velkou výhodou vývojového prostředí automatu ABB PM564 je to, že při vytváření programu není nutné mít připojené PLC. Existuje zde totiž simulační mód, na kterém se dají programy testovat. Další užitečnou součástí tohoto vývojového prostředí je tvorba vizualizací, kterými je možno řídit a sledovat daný technologický proces.

Programovatelný automat ABB PM564 se může plně zapojit do výuky, protože díky rozšiřujícímu modulu digitálních vstupů a výstupů je možné s ním řídit i modely ze sady EDUtec.

CONCLUSION

This bachelor thesis mainly dealt with a programmable logic controller from the company ABB, specifically with the type PM564-T-ETH. The aim of the thesis was to create a workplace containing this controller for the teaching of a subject Programmable controllers.

General information about programmable controllers and their division is at the beginning of the theoretical part. Next part is focused on the description of an international standard concerning the programming of PLC. Thereinafter, information about the company ABB with the summary and the comparison of its products is mentioned. In the last part of the theory the controller PM564-T-ETH is described from the hardware point of view. The practical part begins with the manual, in which the complete process from the connection of PLC, through the setting of communication and configuration up to the development of program in CoDeSys is described. After that, a model of the pumped storage hydro plant is depicted along with the creation of the program for its controlling. At the end of the practical part there are assignments and solutions of exemplary tasks, which can be used for teaching purposes.

Two difficult problems had to be solved during the working on the thesis. First problem occurred within first attempts to program the controller. PLC announced the series of hardware errors. Errors were caused by out of date firmware which excluded the right function of the controller. The problem was solved by the updating of the firmware, which was made by Mr. Ing. Ivánek from the company Konel s.r.o. The lack of digital inputs of PLC was the second problem. Three digital inputs were necessary for the optimal control of the model of the pumped storage hydro plant, however, after the connection of all basic circuits PLC had only two remaining digital inputs. Therefore, the original concept had to be changed. A new connection used only one analog input to control the model.

The big advantage of the development environment of the controller ABB PM564 is that a program can be created without the activation of PLC. This is possible because of the simulation mode, in which programs can be tested. Another useful part of this development environment is the creation of visualizations that allow to control and observe the technological process.

Programmable controller ABB PM564 can be fully engaged to the school training because along with the broadening model of digital inputs and outputs the controller can control models from the EDUtec series, too.

SEZNAM POUŽITÉ LITERATURY

[1] ABB. *Automation products: AC500, CP400, CP600, DigiVis 500, Wireless*. [online]. [cit. 2012-04-26]. Dostupné z: [http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/a9d2e5d590304dc7c125799800336877/\\$file/1sbc125003c0204-automation%20products_new-br.pdf](http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/a9d2e5d590304dc7c125799800336877/$file/1sbc125003c0204-automation%20products_new-br.pdf)

[2] ABB. ABB Power and Productivity for a Better World. *Timeline*. [online].

© 2012 ABB [cit. 2012-05-01]. Dostupné z:

<http://www.abb.com/cawp/abbzh252/c2db8bbe34ab0961c1256aed003368b8.aspx>

[3] ABB. ABB v České republice. *Vítejte v ABB*. [online]. © 2012 ABB [cit. 2012-05-01]. Dostupné z:

<http://www.abb.cz/cawp/czabb014/>

[49db58a47c2d18d4c1257598004151b4.aspx?v=C82&leftdb=GLOBAL/CZABB/czabb014.NSF&e=cz&leftmi=cc70b0fdf470bdbcc1256a850029b508](http://www.abb.cz/cawp/czabb014/49db58a47c2d18d4c1257598004151b4.aspx?v=C82&leftdb=GLOBAL/CZABB/czabb014.NSF&e=cz&leftmi=cc70b0fdf470bdbcc1256a850029b508)

[4] ABB. Verze V2.0.0. *Nápověda HTML*. © 2005-2010. Datum vydání 2010-07-12.

[5] Automatizace. *Svělený had: návod*. [online]. [cit. 2012-03-27]. Dostupné z: http://skola.hellebrand.cz/text0910/au/aut_4-01/had1.pdf

[6] BEDNÁŘ, Josef. *Malé vodní elektrárny 2: turbíny*. 1. vyd. Praha: SNTL, 1989.

[7] ČEZ. Skupina ČEZ. *Přečerpávací vodní elektrárna Dlouhé Stráně*. [online]. © 2012 ČEZ, a.s. [cit. 2012-04-21]. Dostupné z:

<http://www.cez.cz/cs/vyroba-elekriny/obnovitelne-zdroje/voda/dlouhe-strane.html>

[8] HÁJEK, Gustav. *Vodní motory: vodní energie a její využití. Vodní díla. Vodní kola a turbíny. Montáž a provoz turbin*. Praha: J. Hokr, 1947.

[9] HÁJEK, Jan. *Blikače s časovačem 555*. 1. vyd. Praha: AA Praha a BEN - technická literatura, 2006. ISBN 80-85230-30-5.

[10] JOHN, Karl and Michael TIEGELKAMP. *IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids*. 2nd ed. Berlin: Springer-Verlag, 2010. ISBN 978-3-642-12014-5.

[11] KOHOUT, Luděk. Učební Texty. *Úvod do programování podle normy IEC 61131-3*. [online]. © 2008 Luděk Kohout. [cit. 2012-04-20]. Dostupné z: http://www.edumat.cz/texty/zaciname_IEC61131.pdf

[12] KOVÁŘ, Josef. *Programování PLC*. [online]. [cit. 2012-5-28]. Dostupné z: http://www.spszl.cz/soubory/plc/programovani_plc.pdf

[13] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. *Řízení programovatelnými automaty*. Vyd. 1. Praha: ČVUT, 1998. ISBN 80-01-01766-4.

[14] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. *Řízení programovatelnými automaty II*. Vyd. 1. Praha: ČVUT, Strojní fakulta, 2000. ISBN 80-010-2096-7.

[15] Mosaic Lite. Verze 2011.3. *Nápověda: programování podle IEC 61 131-3*. © 1990-2011 by mxSOFT. © 2000-2011 by Teco a.s.

[16] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace: 1. díl - Základní pojmy, úvod do programování*. 1. vyd. Praha: BEN - technická literatura, 1999. ISBN 978-80-86056-58-6.

[17] ŠMEJKAL, Ladislav. *PLC a automatizace, 2.díl. 1. Vyd.* Praha: BEN - technická literatura, 2005. ISBN 80-7300-087-3.

[18] Vodní a tepelné elektrárny. *Vodní elektrárny v ČR*. [online]. [cit. 2012-04-21]. Dostupné z: <http://www.vodni-tepelne-elektrarny.cz/vodni-elektrarny-cr.htm>

[19] VOJÁČEK, Antonín. *TEST - PLC ABB PM564 – 1.díl – popis hardwaru*. Automatizace.hw.cz. [online]. Březen 31, 2011 [cit. 2012-01-26]. Dostupné z: <http://automatizace.hw.cz/test-plc-abb-pm564-1dil-popis-hadwaru>

[20] VOJÁČEK, Antonín. *TEST - PLC ABB PM564 – 2.díl – programování*. Automatizace.hw.cz. [online]. Duben 2, 2011 [cit. 2012-01-26]. Dostupné z: <http://automatizace.hw.cz/test-plc-abb-pm564-2dil-programovani>

[21] VORÁČEK, Rudolf. *Automatizace a automatizační technika 2: Automatické řízení*. Vyd. 1. Praha: Computer Press, 2005. ISBN 80-251-0796-5.

[22] ZEŽULKA, František. *Programovatelné automaty*. [online]. Brno: FEKT VUT, 1. 10. 2003, [cit. 2012-5-28]. Dostupné z: http://www.vaeprosys.cz/Dokumentace/Programovatelné_automaty/Programovatelné_automaty-Skripta_FEKT_VUT_Brno.pdf

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CFC Continuous Function Chart (volně propojované bloky)

CPU Central processing unit (centrální procesorová jednotka)

FBD Function Block Diagram (schéma z funkčních bloků)

IL Instructions List (instrukční soubor)

IP Internet Protocol (internetový protokol)

LD Ladder Diagram (kontaktní schémata)

LED Light Emitting Diode (světlo emitující dioda)

PLC Programmable Logic Controller (programovatelný automat)

POU Program Organisation Unit (programově organizační jednotka)

RTC Real-time Clock (hodiny reálného času)

SFC Sequential Function Chart (vývojová schémata)

ST Structured Text (strukturovaný text)

USB Universal Serial Bus (univerzální sériová sběrnice)

SEZNAM OBRÁZKŮ

Obrázek 1 - Cyklus PLC [16, str. 46]	11
Obrázek 2 - Přehled PLC od firmy ABB [1]	17
Obrázek 3 - PLC ABB PM564-T-ETH [4]	18
Obrázek 4 - ABB Configurator	20
Obrázek 5 - CoDeSys	21
Obrázek 6 - Popis nastavení komunikace mezi PLC a PC	23
Obrázek 7 - Okna nastavení z bodu c a d	24
Obrázek 8 - Okno pro hledání PLC, která jsou k dispozici	25
Obrázek 9 - Okno, které musíte vyplnit při zakládání nového projektu	25
Obrázek 10 - Postranní lišta Devices s očíslovanými částmi pro usnadnění popisu	26
Obrázek 11 - Mapování vstupů a výstupů	27
Obrázek 12 - Popis vývojového prostředí CoDeSys	28
Obrázek 13 - Okno, ve kterém se nastavují komunikační parametry	29
Obrázek 14 - Prostředí pro tvorbu simulací	30
Obrázek 15 - Fotografie modelu přečerpávací elektrárny	33
Obrázek 16 - Základní zapojení doporučená výrobcem pro digitální vstupy a výstupy	34
Obrázek 17 - Základní zapojení doporučené výrobcem pro analogový vstup s aktivní zátěží	35
Obrázek 18 - Zapojení digitálních vstupů a výstupů	35
Obrázek 19 - Zapojení analogových výstupů	36
Obrázek 20 - Zapojení pro efekt tekoucí vody	37
Obrázek 21 - Schéma modelu přečerpávací elektrárny	38
Obrázek 22 - Simulace přečerpávající elektrárny	40
Obrázek 23 - Schéma programu	42
Obrázek 24 - Ukázka vizualizace ovládání pračky	45
Obrázek 25 - Rozšiřující modul DX561	57
Obrázek 26 - Zapojení doporučená výrobcem pro vstupy a výstupy	58

SEZNAM TABULEK

Tabulka 1 - Rozdělení normy IEC 61131 [10]	13
Tabulka 2 - Srovnání parametrů řad AC500 a AC500-eCo [1]	17
Tabulka 3 - Seznam vstupů a výstupů modelu	34
Tabulka 4 - Kombinace možných stavů spínačů v obvodě pro analogové vstupy	37

SEZNAM PŘÍLOH

Příloha PI: Zdrojový kód přečerpávací elektrárny.....	56
Příloha PII: Modul ABB DX561.....	58
Příloha PIII: Program č. 2.....	61
Příloha PIV: Program č. 3.....	64

PŘÍLOHA P I: ZDROJOVÝ KÓD PŘEČERPÁVACÍ ELEKTRÁRNY

```
PROGRAM PLC_PRG
VAR

END_VAR

IF ana2>8500 THEN (*analogový vstup z ovládacího panelu*)

(*režim auto*)
IF potenciometr>5000 THEN
vystup3:=TRUE; (*signalizace režimu z ovládacího panelu*)
(*klapka + turbína*)
vystup2:=FALSE; (*čerpadlo*)
vystup5:=FALSE; (*signalizace čerpadla z ovládacího
panelu*)
IF (vstup1=TRUE) AND (vstup2=FALSE) THEN
vystup0:=TRUE; (*klapka*)
vystup1:=TRUE; (*turbína*)
vystup4:=TRUE; (*signalizace turbíny z ovládacího
panelu*)
ELSE
vystup0:=FALSE;
vystup1:=FALSE;
vystup4:=FALSE;
END_IF;

ELSE
(*čerpadlo*)
vystup3:=TRUE;
vystup0:=FALSE;
vystup1:=FALSE;
vystup4:=FALSE;
IF (vstup0=FALSE) AND (vstup3=TRUE) THEN
vystup2:=TRUE;
vystup5:=TRUE;
ELSE
vystup2:=FALSE;
vystup5:=FALSE;
END_IF;

END_IF;

ELSE
(*režim manuál*)
IF ana2>8000 THEN (*je-li spínač režimu sepnutý*)
vystup3:=TRUE;
vystup4:=FALSE;
vystup5:=FALSE;
ELSE (*je-li spínač režimu rozepnutý*)
vystup3:=FALSE;

IF (ana2>6700) AND (ana2<7250) THEN (*je-li hodnota na AI v intervalu*)
vystup4:=FALSE;
vystup5:=FALSE;
END_IF;

IF (ana2>6100) AND (ana2<6500) THEN
vystup4:=FALSE;
vystup5:=TRUE;
END_IF;

IF (ana2>5300) AND (ana2<5700) THEN
vystup4:=TRUE;
vystup5:=FALSE;
END_IF;

IF (ana2>4800) AND (ana2<5300) THEN
vystup4:=TRUE;
vystup5:=TRUE;
```

```
END_IF;
```

```
END_IF;
```

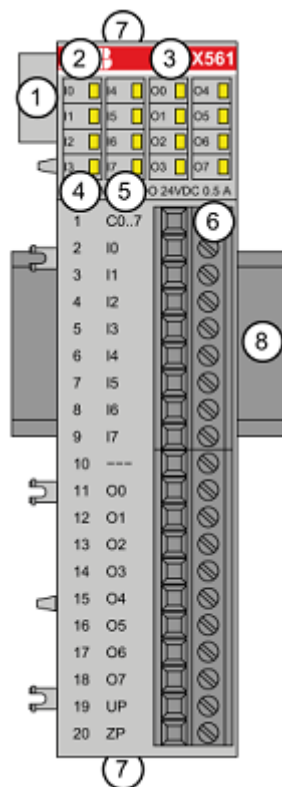
```
(*ohraničující podmínky čidel výšky hladin na elektrárně*)  
IF (vystup4=TRUE) AND (vstup1=TRUE) AND (vstup2=FALSE) THEN  
vystup0:=TRUE;  
vystup1:=TRUE;  
vystup4:=TRUE;  
ELSE  
vystup0:=FALSE;  
vystup1:=FALSE;  
vystup4:=FALSE;  
END_IF;
```

```
IF (vystup5=TRUE) AND (vstup0=FALSE) AND (vstup3=TRUE) THEN  
vystup2:=TRUE;  
vystup5:=TRUE;  
ELSE  
vystup2:=FALSE;  
vystup5:=FALSE;  
END_IF;
```

```
END_IF;
```


PŘÍLOHA P II: MODUL ABB DX561

ABB DX561



Obrázek 25 - Rozšiřující modul DX561

ABB DX561 je rozšiřující modul digitálních vstupů a výstupů pro PLC řady ABB AC500 (PM5XX).

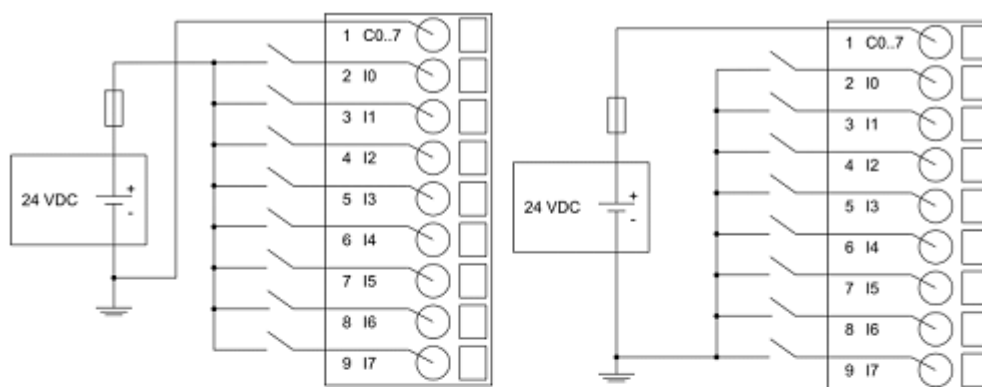
Popis modulu

Modul se skládá ze I/O sběrnice (1), osmi LED diod indikujících stav digitálních vstupů (2), osmi LED diod indikujících stav digitálních výstupů (3), číselného označení svorek (4) a jejich funkcí (5), ovládání svorek (6), díry na šroub k připevnění na zeď (7) a drážky na DIN lištu (8).

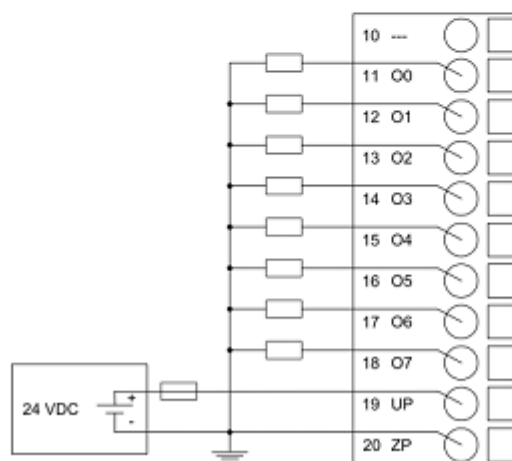
Připojení modulu DX561 a PLC PM564-T-ETH

Přídavný modul se umísťuje na bok PLC, kde je výstup sběrnice. Připevnění modulu k PLC je velmi intuitivní a jednoduché – sběrnice modulu se pouze zapojí do sběrnice PLC a celý modul se tak přichytí k PLC. Stejně jednoduchá je i jeho demontáž. Po přidělení modulu k PLC je potřeba v programu ABB Configurator v projektové části modul přidat (klikneme levým tlačítkem myši na IO_Bus a vybereme Add device. Otevře se nám seznam připojitelných zařízení, z něho vybereme modul DX561 a potvrdíme).

Výrobce udává tato základní zapojení pro digitální vstupy a výstupy:



Dvě možná zapojení pro analogové vstupy



Zapojení pro digitální výstupy

Obrázek 26 - Zapojení doporučená výrobcem pro vstupy a výstupy

Modely EDU-mod a DX561

Modely EDU-mod mají jako vstupy a výstupy vyvedené konektory Cannon. Vstupní konektor obsahuje pouze 4 datové vodiče, které jsou umístěny na pinech 6, 7, 8 a 9. (Za datové vodiče považují vodiče, které nesou vstupní/výstupní informaci do/z PLC.) Pro správnou funkci PLC ABB PM564-T-ETH s modulem DX561 je nutné připojit zápornou svorku zdroje na vstup C0..7.

Výstupní konektory obsahují napájení na 4. pinu a zem na 1. pinu. Datové vodiče jsou opět na pinech 6, 7, 8 a 9. Aby nám výstupy na modulu fungovaly, musíme je napájet - kladné napětí na UP a záporné na ZP.

Speciálním zapojením konektoru Cannon pro modely EDU-mod jsou moduly tlačítek. Konektor obsahuje napájení na 5. pinu a dále 4 datové vodiče na pinech 6, 7, 8 a 9.

Užité modely

Pro program č. 1 byl využit jeden konektor Cannon pro modul tlačítek a LED diody byly použity ze signalizace výstupů PLC. Pro program č. 2 (křižovatku) byl využit konektor Cannon pro tlačítka a dva výstupní konektory pro model křižovatky. Pro program č. 3 byly využity 2 výstupní a 1 vstupní konektor Cannon.

PŘÍLOHA P III: PROGRAM Č. 2

```
PROGRAM PLC_PRG
VAR
    den:rezim_den;
    noc:rezim_noc;
    hodiny:CLOCK;
    hodina: BYTE;
END_VAR

(*Hlavní program *)
(*nastavení reálného času a jeho spuštění*)
hodiny(EN:=TRUE,SET:=TRUE,HOURL_SET:=15,MIN_SET:=29,SEC_SET:=30,YEAR_SET:=2012,MON
_SET:=65,DAY_SET:=2);
hodina:=hodiny.HOURL_SET; (*nastavení hodiny do proměnné hodina*)

IF vstup1=FALSE THEN
    (*je-li tlačítko vypnuté, tak se provádí režim dle hodin - buď denní nebo noční*)
    IF (hodina<22 AND hodina>6) THEN (*zde se definuje rozmezí hodin pro denní
    režim*)
        den(); (*volání funkčního bloku denního režimu*)
        pomocna:=TRUE; (*pomocná proměnná pro vymazání časovače*)
    ELSE
        noc(); (*volání funkčního bloku pro noční režim*)
        pomocna:=FALSE; (*pomocná proměnná pro vymazání časovače*)
    END_IF;
    ELSE
        noc(); (*volání funkčního bloku pro noční režim*)
        pomocna:=FALSE; (*pomocná proměnná pro vymazání časovače*)
    END_IF;

FUNCTION_BLOCK rezim_den
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
    casovac:TON;
    tlacitkol: BOOL;
END_VAR

F tlacitkol=FALSE AND casovac.Q=FALSE THEN (*podmínka pro odstartování
časovače*)
    tlacitkol:=TRUE;
END_IF;

IF pomocna=FALSE THEN (*podmínka pro zapnutí semaforu od
začátku*)
    sem2o:=FALSE;
    tlacitkol:=FALSE;
END_IF;

casovac(IN:=tlacitkol,PT := T#50s);
(*volání časovače*)
IF casovac.ET>=T#0s AND casovac.et<T#3s THEN
(*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
    sem1c:=TRUE;
    sem1o:=FALSE;
    sem1z:=FALSE;
    sem2c:=TRUE;
    sem3c:=TRUE;
ELSIF casovac.et>=T#3s AND casovac.ET<T#5s THEN
(*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
    sem1c:=TRUE;
    sem1o:=TRUE;
    sem1z:=FALSE;
ELSIF casovac.ET>=T#5s AND casovac.ET<T#21s THEN
(*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
    sem1c:=FALSE;
```

```

sem1o:=FALSE;
sem1z:=TRUE;
ELSIF casovac.ET>=T#21s AND casovac.ET<T#23s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem1c:=FALSE;
sem1o:=TRUE;
sem1z:=FALSE;
ELSIF casovac.ET=T#23s THEN
sem1c:=TRUE;
sem1o:=FALSE;
sem1z:=FALSE;
ELSIF casovac.ET>=T#23s AND casovac.ET<T#25s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem2c:=TRUE;
sem2o:=FALSE;
sem2z:=FALSE;
ELSIF casovac.ET>=T#25s AND casovac.ET<T#27s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem2c:=TRUE;
sem2o:=TRUE;
sem2z:=FALSE;
ELSIF casovac.ET>=T#27s AND casovac.ET<T#38s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem2c:=FALSE;
sem2o:=FALSE;
sem2z:=TRUE;
ELSIF casovac.ET>=T#38s AND casovac.ET<T#40s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem2c:=FALSE;
sem2o:=TRUE;
sem2z:=FALSE;
ELSIF casovac.ET=T#40s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem2c:=TRUE;
sem2o:=FALSE;
sem2z:=FALSE;
ELSIF casovac.ET>=T#40s AND casovac.ET<T#43s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem3c:=TRUE;
sem3z:=FALSE;
ELSIF casovac.ET>=T#43s AND casovac.ET<T#48s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem3c:=FALSE;
sem3z:=TRUE;
ELSIF casovac.ET>=T#48s THEN
  (*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem3c:=TRUE;
sem3z:=FALSE;
END_IF;

IF casovac.Q=TRUE THEN
  tlacitko1:=FALSE;
END_IF;

(*podmínka pro nulování časovače*)

FUNCTION_BLOCK rezim_noc
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
  casovac:TON;
  tlacitko2:BOOL;
END_VAR

sem1c:=FALSE;
sem1z:=FALSE;
sem2c:=FALSE;
sem2z:=FALSE;
sem3c:=FALSE;
sem3z:=FALSE;
(*nastavení světla na semaforu*)
(*nastavení světla na semaforu*)
(*nastavení světla na semaforu*)
(*nastavení světla na semaforu*)
(*nastavení světla na semaforu*)
(*nastavení světla na semaforu*)

```

```

IF tlacitko2=FALSE AND casovac.Q=FALSE THEN
(*podmínka pro odstartování časovače*)
tlacitko2:=TRUE;
END_IF;

casovac(IN:=tlacitko2,PT := T#3s);
(*volání časovače*)

IF casovac.ET>=T#0s AND casovac.ET<T#1s THEN
(*nastavení časového rozmezí pro jednotlivá světla na semaforu*)
sem1o:=FALSE; (*nastavení světla na semaforu*)
sem2o:=FALSE; (*nastavení světla na semaforu*)
ELSIF casovac.ET>=T#1s AND casovac.ET<T#2s THEN
sem1o:=TRUE; (*nastavení světla na semaforu*)
sem2o:=TRUE; (*nastavení světla na semaforu*)
ELSIF casovac.ET>=T#2s THEN
sem1o:=FALSE; (*nastavení světla na semaforu*)
sem2o:=FALSE; (*nastavení světla na semaforu*)
END_IF;

IF casovac.Q=TRUE THEN (*podmínka pro nulování časovače*)
tlacitko2:=FALSE;
END_IF;

VAR_GLOBAL
pomocna: BOOL;
END_VAR

```

PŘÍLOHA P IV: PROGRAM Č. 3

```
PROGRAM PLC_PRG
VAR
    prog:praci_program;
END_VAR

(*podmínky pro simulaci*)
IF devadesat=TRUE THEN      (*vstup z modelu pračky 90 stupňů*)
    pozadovana:=90;
ELSIF sedesat=TRUE THEN    (*vstup z modelu pračky 60 stupňů*)
    pozadovana:=60;
ELSIF ctycicet=TRUE THEN   (*vstup z modelu pračky 40 stupňů*)
    pozadovana:=40;
ELSIF tricet=TRUE THEN     (*vstup z modelu pračky 30 stupňů*)
    pozadovana:=30;
END_IF;

IF prom=FALSE AND doprano=TRUE THEN
    nepracuje:=TRUE;
ELSE
    nepracuje:=FALSE;
END_IF;

prog();(*volání funkčního bloku praci_program *)

FUNCTION_BLOCK praci_program
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
    teplota:INT;
    casovac:TON;
END_VAR

(*plnění hladiny na 100 procent*)
IF hladina100=TRUE OR doprano=TRUE THEN (*vstup z modelu pračky výška 100%*)
    voda:=FALSE; (*vstup do modelu-voda*)
    IF doprano=TRUE THEN
        IF hladina50=TRUE THEN
            cerpadlo:=TRUE;
        ELSE
            cerpadlo:=FALSE;
        END_IF;
    END_IF;
    ELSIF hladina100=FALSE THEN
        voda:=TRUE;
    END_IF;

(*nastavení hodnoty teplota podle aktivních výstupu*)
    IF teplota90=TRUE THEN
        teplota:=90;
    ELSIF teplota60=TRUE THEN
        teplota:=60;
    ELSIF teplota40=TRUE THEN
        teplota:=40;
    ELSIF teplota30=TRUE THEN
        teplota:=30;
    ELSE
        teplota:=0;
    END_IF;

(*topení a regulace*)
    IF teplota<pozadovana AND doprano=FALSE THEN
        topeni:=TRUE;
    ELSE
        topeni:=FALSE;
    END_IF;
END_FUNCTION_BLOCK
```

```

END_IF;

IF start=TRUE THEN
casovac(IN:=FALSE);
prom:=FALSE;
start:=FALSE;
doprano:=FALSE;
END_IF;

(*praní*)
IF teplota=pozadovana AND hladinal00=TRUE AND doprano=FALSE THEN
casovac(IN:=TRUE,PT:=T#40s);
IF casovac.ET<T#20s THEN
buben_vpravo:=TRUE; (*vstup do modelu pračky točení bubnu vpravo*)
buben_vlevo:=FALSE; (*vstup do modelu pračky točení bubnu vlevo*)
ELSIF casovac.ET<T#40s THEN
buben_vpravo:=FALSE;
buben_vlevo:=TRUE;
ELSIF casovac.ET=T#40s THEN
buben_vpravo:=FALSE;
buben_vlevo:=FALSE;
prom:=TRUE;
casovac(IN:=FALSE);
doprano:=TRUE;
END_IF;
END_IF;

(*ždímnání*)
IF prom=TRUE AND hladina50=FALSE AND teplota=0 THEN
otacky:=TRUE; (*vstup do modelu pračky-otáčky*)
casovac(IN:=TRUE,PT:=T#10s);
IF casovac.ET<T#5s THEN
buben_vpravo:=TRUE;
buben_vlevo:=FALSE;
ELSIF casovac.ET<T#10s THEN
buben_vpravo:=FALSE;
buben_vlevo:=TRUE;
ELSIF casovac.Q=TRUE THEN
buben_vpravo:=FALSE;
buben_vlevo:=FALSE;
otacky:=FALSE;
prom:=FALSE;
END_IF;
END_IF;

VAR_GLOBAL
    pozadovana: INT:=30;
    start: BOOL:=FALSE;
    prom: BOOL;
    doprano: BOOL:=TRUE;

    devadesat: BOOL;
    sedesat: BOOL;
    ctyricet: BOOL;
    tricet: BOOL;
    nepracuje: BOOL;
END_VAR

```