



Univerzita Tomáše Bati ve Zlíně

Fakulta technologická

Institut informačních technologií

**IMPLEMENTACE NOVÝCH ALGORITMŮ DO
ŘÍDÍCÍHO SOFTWARE PLC PRO ŘÍZENÍ
TECHNOLOGICKÉHO PROCESU**

Diplomová práce

2005

Repka Lukáš

Zde vložit oficiální zadání DP.

ABSTRAKT

V průmyslové praxi nastává v současné době tendence nahrazovat reléové obvody, používané pro řízení technologický procesů, programovatelnými automaty. Díky jejich nízké pořizovací ceně a vlastnostem, které dokáží splnit složité požadavky se stávají běžnou součástí novějších technologických zařízení.

Tyto programovatelné automaty však obsahují ve svých knihovnách jen základní instrukce pro regulaci soustav.

Tato práce se zabývá rozšířením stávající knihovny pro programovatelný automat SAIA PCD2 určený pro monitorovací a řídicí aplikace o několik programů umožňující regulaci. V rámci práce byl vytvořen dvoupolohový regulátor a PSD regulátor.

Programem Matlab byly regulační pochody těchto regulátorů simulovány a porovnány s výsledky reálné soustavy, která byla řízena naprogramovaným PLC. Vizualizace procesu probíhala pomocí programu Control Web 2000.

SUMMARY

Programmable logic controllers are commonly used in industry for controlling technological processes. However these programmable logic controllers contain in their instruction library only basic instructions for control systems.

This thesis focuses on implementing a few more macro instructions into the instruction library of PLC SAIA PCD2 that will enable control. As part of this thesis, a two-position controller and PSD controller have been devised.

Control performances of these controllers were simulated in the Matlab programme and compared with a real system, that was controlled by a preprogrammed PLC. Visual presentation of the process was performed through the Control Web 2000.

Chtěl bych poděkovat vedoucímu práce Ing. Tomáši Sysalovi, Ph.D. za odborné vedení a pomoc v průběhu řešení této práce.

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího diplomové práce a ředitele ústavu. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že jsem na celé diplomové práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně dne 13.6.2005

.....

podpis diplomanta

OBSAH

ABSTRAKT	3
ÚVOD	6
1. CÍL DIPLOMOVÉ PRÁCE	7
2. TEORETICKÁ ČÁST	8
2.1. Programovatelný automat	8
2.1.1. Základní charakteristika programovatelných automatů	8
2.1.2. Konstrukční a elektrické provedení PLC	9
2.1.3. Programové vybavení.....	11
2.1.4. Přehled programovatelných automatů.....	12
2.2. Rodina automatů SAIA PCD	13
2.3. Programovatelný automat SAIA PCD2.....	13
2.3.1. Architektura PCD2	13
2.3.2. Dodávané typy vstupně výstupních modulů.....	14
2.3.3. Typy datových prostředků využitelných v programech	15
2.3.4. Struktura uložení programu	16
2.4. Vývojové prostředí PG4	17
2.4.1. Instruction list	19
2.4.2. Function plan	19
2.4.3. Graftec	21
2.5. Control WEB 2000	21
2.5.1. Vytvoření podprogramů	22
2.5.2. Virtuální přístroje	24
2.5.3. Konstanty, proměnné a kanály	25
2.5.4. Připojení k PLC Saia	25
2.6. Regulátory	25
2.6.1. Regulátory v knihovnách PLC	25
2.6.2. Dvoupolohový regulátor	27
2.6.3. PSD regulátor	27
3. EXPERIMENTÁLNÍ ČÁST	29
3.1. Laboratorní tepelná soustava.....	29
3.1.1. Popis laboratorní soustavy.....	29
3.1.2. Identifikace laboratorní soustavy.....	29
3.2. Obecný popis vytvořených programů regulátorů.....	31
3.3. Vizualizace řízeného procesu	31
3.4. Dvoupolohová regulace.....	32
3.5. PSD regulace	33
4. ZÁVĚR	34
SEZNAM POUŽITÉ LITERATURY	36
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	37
SEZNAM OBRÁZKŮ	38
SEZNAM TABULEK	39
SEZNAM PŘÍLOH	40
PŘÍLOHY	41
Příloha A – obrazová dokumentace.....	41
Příloha B – parametrický soubor.....	44
Příloha C – mapovací soubor	45
Příloha D – Výsledky řízení dvoupolohového regulátoru.....	47
Příloha E – Výsledky řízení PSD regulátoru	48

ÚVOD

V dnešní době se automatizace stává standardem rozsáhlých výrobních linek a náročných technologických procesů. Kvalitní a inteligentní řízení je dnes dostupné i pro obyčejné stroje, pomocné mechanismy a technologická zařízení ve všech oborech.

Nejrozšířenější řídicí systémy v průmyslové praxi – reléové obvody, či analogové regulátory, jsou v současné době nahrazovány moderním číslicovým řízením. Při použití běžných PID regulátorů je ke spolehlivé regulaci třeba znát vlastnosti regulované soustavy. Při řešení takovéto úlohy bychom se mohli dostat do problémů zvláště tehdy, nemá-li regulovaná soustava v čase konstantní parametry. Tento problém do jisté míry odpadá při použití moderních metod řízení, které využívají adaptivní regulátory. Toto řešení však sebou přináší složitý algoritmus a je tedy náročné na technické požadavky systému, který má toto řízení zastávat. Použití klasického PC je v průmyslovém prostředí prakticky nemožné. Řešením jsou v současné době průmyslové PC anebo programovatelné automaty.

Jejich nízká cena a vlastnosti splňující náročné požadavky je začleňují do technologických procesů jako základní řídicí prvek. Pro řešení mé diplomové práce bude použit programovatelný automat firmy SAIA PCD2.

Úkolem této diplomové práce bude implementovat do tohoto programovatelného automatu nové typy regulátorů, které nejsou standardně obsaženy v jeho knihovnách. Toto bude řešeno pomocí vývojového softwaru PG4 dodávaného firmou SAIA. Následné grafické průběhy regulačních pochodů budou vykresleny pomocí vizualizačního softwaru Control Web 2000.

1. CÍL DIPLOMOVÉ PRÁCE

Cílem diplomové práce je, jak je patrné ze zadání, implementovat různé typy regulátorů do knihovny podprogramů automatu SAIA PCD2. Toto bude řešeno pomocí vývojového softwaru dodávaného firmou SAIA PG4. Algoritmy regulátorů jsou uvedeny v kapitole 2.5. Konečná vizualizace se základními řídicími prvky se bude realizovat programem Control Web 2000.

2. TEORETICKÁ ČÁST

2.1 Programovatelný automat

2.1.1 Základní charakteristika programovatelných automatů

Programovatelný automat je uživatelský programovatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů nebo strojů. Nejčastěji se označuje zkratkou PLC (Programmable Logic Controller).

Zpočátku byly navrženy pro řešení úloh logického řízení, hlavně jako náhrada za pevnou reléovou logiku. S postupem času a s rozvojem výpočetní techniky začaly pronikat i do úloh monitorování a měření a nahrazují dokonce i samotné regulátory.

Programovatelný automat můžeme rozdělit na tyto části:

- centrální procesorová jednotka
- systémové paměti
- uživatelské paměti
- moduly vstupních a výstupních jednotek pro připojení řízeného systému
- moduly komunikačních jednotek
- speciální moduly
- záložní paměťový modul

Na rozdíl od reléových obvodů, jejichž činnost je pevně dána zapojením, jsou programovatelné automaty nesrovnatelně variabilnější, neboť jejich činnost udává program uložený v paměti a je možné jej poměrně snadno změnit.

Program v PLC je vykonáván v cyklech, kdy na začátku jsou přečteny vstupy z paměti a na konci jsou výstupní hodnoty u paměťového místa vloženy na výstupní bránu. Z toho je patrné, že program neumí zachytit změnu vstupu uprostřed cyklu a stejně tak změna na výstupu se projeví až po proběhnutí programové smyčky. Na jedné straně takto může být v omezené míře eliminováno rušení, na straně druhé nezachycení krátkého impulsu nebo jeho náběžné resp. sestupné hrany..

Jedním z hledisek, podle kterých můžeme PLC rozdělit, je jejich konstrukční řešení:

Mikro PLC nabízejí pevnou sestavu vstupů a výstupů. Svým kompaktním provedením, malými rozměry a nízkou cenou se řadí do kategorie spotřebního materiálu. Jejich funkční a programátorský komfort je obvykle redukován na nezbytné minimum, komunikační možnosti mnohdy chybějí.

Kompaktní PLC nabízejí určitou i když omezenou variabilitu. Uživatel může k základnímu modulu připojit jeden nebo několik přídatných modulů z omezeného sortimentu s pevnou kombinací vstupů a výstupů, např. modul s 8 binárními vstupy a 8 binárními výstupy, modul rychlých čítačů, analogový vstupní a výstupní modul, modul regulátoru, apod.

Modulární PLC dává větší volnost ve volbě konfigurace, umožňuje zasouvat libovolné moduly. Jeden systém může tvořit několik rámců, rozšiřovací moduly lze připojit i na vzdálenost stovek metrů.

2.1.2 Konstrukční a elektrické provedení PLC

Pouzdro, základní modul, napájení

Kompaktní PLC jsou umístěny buď v jednom pouzdře nebo dovolují v malé míře rozšíření. Většinou se montují přímo na DIN lištu do rozvaděče.

V základním modulu modulárních PLC (rám, nosná deska, lišta) bývá umístěna centrální procesorová jednotka (CPU), napájecí jednotka, systémová a napájecí sběrnice a několik volných pozic pro zásuvné periferní jednotky. Pokud počet volných pozic v základním modulu nepostačuje, bývají k dispozici rozšiřující moduly. Rozšiřující moduly se připojují pomocí tzv. expanzní jednotky.

Centrální procesorová jednotka je jádrem celého PLC, určuje jeho výkonnost, je jednoprocessorová nebo víceprocesorová. U víceprocesorových systémů bývají použity matematické koprocessory nebo vstupně výstupní procesory, někdy též komunikační procesory, pro urychlení práce používá rychlý bitový procesor. Důležitým charakteristickým parametrem CPU je operační rychlost posuzována podle tzv. doby cyklu, což je doba zpracování 1000 (1k) logických instrukcí. Pohybuje se řádově od desítek až k desetinám milisekund.

Paměťový prostor se může dělit na paměť uživatelskou, systémovou a paměť dat. Do uživatelské paměti se ukládá uživatelský program. Tato paměť bývá typu EPROM nebo

EEPROM s kapacitou řádově od desítek kB až po jednotky MB u modulárních PLC, u kompaktních PLC spíše v desítkách kB. V systémové paměti je umístěn systémový program. Tato paměť bývá rovněž typu EEPROM. V samostatné jednotce může být umístěna přídatná uživatelská paměť. V paměti dat typu RAM jsou umístěny uživateli dostupné registry, čítače, časovače a většinou i vyrovnávací registry pro obrazy vstupů a výstupů. Počet těchto registrů výrazně ovlivňuje možnosti PLC. Adresovatelný prostor vymezený pro vstupy/výstupy omezuje počet připojitelných periferních jednotek.

Důležitým parametrem jsou rozsahy čítačů a časovačů, jeden nebo i více sériových komunikačních kanálů. Většina automatů má dostupné i hodiny reálného času, případně i s kalendářem.

Binární vstupní a výstupní jednotky

Binární vstupní a výstupní jednotky slouží k připojování prvků pro tvorbu vstupů a výstupů s dvouhodnotovým charakterem vstupního a výstupního signálu např. tlačítka, přepínače, koncové spínače, signalizace a podobně.

Analogové vstupní jednotky

Analogové vstupní jednotky zprostředkují kontakt programovatelného automatu se spojitým prostředím. Lze k nim připojit např. snímače teploty, vlhkosti, tlaku, síly, hladiny, rychlosti, ale i inteligentní přístroje s analogovými výstupy. Důležitou součástí je A/D převodník s rozsahem 8 nebo 12 bitů. Sortiment bývá široký, např. jednotky pro určité typy čidel – pro termočlánky. U specializovaných jednotek je poněkud potlačena univerzálnost, zato jsou optimálně přizpůsobeny svému určení a poskytují tak levnější a kvalitnější řešení. Analogové moduly s galvanickým oddělením dovolují zvýšit odolnost systému proti rušení, v některých situacích jsou principiálně nenahraditelné.

Analogové výstupní jednotky

Analogové výstupní jednotky slouží pro ovládání různých akčních členů či zařízení se spojitým charakterem vstupního signálu, jako např. spojitě servopohony, frekvenční měniče, ale třeba i ručkové měřicí přístroje apod. Nezbytnou součástí je D/A převodník, většinou má rozsah 8 nebo 12 bitů. Analogové výstupy jsou buď napětíové nebo proudové. Proudové

výstupy mohou být aktivní nebo pasivní. U aktivních nemusí mít připojený akční člen svůj zdroj proudu a je napájen přímo z analogového proudového výstupu automatu. Připojený akční člen má v tomto případě omezený maximální odpor.

Dále se v PLC můžeme setkat s čítačovými, polohovacími, komunikačními, počítačovými či speciálními jednotkami zastávající různé funkce pro řešení konkrétních úloh.

2.1.3 Programové vybavení

Při své činnosti používá programovatelný automat dva typy programového vybavení:

- systémový program (operační systém)
- uživatelský program, v němž je zapsán algoritmus řešení dané úlohy

Systémový program (operační systém) programovatelných automatů je řešen pro každý automat poněkud odlišně a je závislý na technickém vybavení automatu. Je schopen zpracovávat instrukce příslušné danému typu automatu. Podle typu automatu musí být schopen zajistit v nejjednodušším případě režim jedno-úlohového zpracovávání uživatelského programu bez přerušení, u složitějších systémů více-úlohové zpracovávání (multitasking), u nejvyspělejších systémů s několika procesory pak více-úlohové zpracovávání na všech dostupných procesorech (multitasking i multiprocessing). Aby mohly uživatelské programy pracovat v těchto režimech, musí se při jejich programování dodržovat určitá pravidla.

Každý operační systém poskytuje pro použití v uživatelských programech určitý soubor tzv. systémových služeb (např. čas systému, datum, různé typy časovačů, interní kód chyby apod.).

Uživatelské programy lze vytvářet buď pomocí samostatných přístrojů, tzv. programátorů, nebo pomocí PC. Pro většinu PLC je k dispozici speciální programové vybavení pro tvorbu uživatelských programů na PC. V něm je možno konfigurovat celý systém, provádět vlastní zápis a editaci programu, editovat alokační tabulky proměnných. Bývá nabízena syntaktická kontrola programu po překladu do jazyka strojových instrukcí. Vyspělejší systémy poskytují možnost odlaďování programu bez přímé účasti PLC v simulačním režimu.

K programování nabízejí PLC systémy specializované jazyky, původně navržené pro snadnou, názornou a účinnou realizaci logických funkcí. Jazyky systémů různých výrobců

jsou podobné, nikoliv však stejné. Mezinárodní norma IEC 1131 se snaží jazyky a zvyklosti různých výrobců PLC co nejvíce sblížit. Ustálili se tři třídy jazyků pro programování PLC.

- Jazyk mnemokódů je obdobou assembleru u počítačů a je také strojově orientován, tzn. že každé instrukci PLC systému odpovídá stejně pojmenovaný příkaz jazyka. Dovolují nejlépe přizpůsobit úlohu možnostem PLC a vytěžit maximum z jeho instrukčního souboru.
- Jazyk kontaktních reléových schémat je grafický program ve kterém se základní logické operace zobrazují ve formě obvyklé pro kreslení schémat při práci s reléovými a kontaktními prvky. Jazyk kontaktních schémat je výhodný při programování nejjednodušších logických operací a v případech, kdy s ním pracuje personál, který nezná (nechce znát) tradiční počítačové programování. Pokud v programu převažují složitější funkce (třeba aritmetické instrukce nebo logické s vektorovými operandy, skoky a volání), pak je kontaktní schéma již násilné a postrádá svoji názornost.
- Jazyk logických schémat je také grafický. Základní logické operace popisuje obdélníkovými značkami Vychází vstříc uživatelům zvyklým na kreslení logických schémat pro zařízení s integrovanými obvody.

Některé systémy pracují ještě s dalšími jazyky, například s jazykem C, s textovým jazykem, vycházejícím z Pascalu. Pascalský zápis programu je úsporný a názorný, zejména pro mladší programátory, kteří v něm na škole programovali.

2.1.4. Přehled programovatelných automatů

Na našem trhu je možno se nejčastěji setkat s programovatelnými automaty těchto nejvýznamnějších světových výrobců (řazeno abecedně): ABB, Allen-Bradley, AEG, B+R, Eberle, Klöckner Moeller, Festo, GE, H+B, Idec, Matsushita, Mitsubishi, Omron, Saia, Siemens, Telemecanique. Mezi české výrobce lze například uvést Amit, Teco a.s. Kolín. V detailech se jednotlivé třídy systémů a jejich představitelé liší, způsoby použití a aplikační možnosti jsou však srovnatelné. Výrobce SAIA je předmětem mého zájmu v této diplomové práci.

2.2. Rodina automatů SAIA PCD

Modelová řada automatů SAIA PCD (Process Control Device) má společnou základní architekturu, stejnou instrukční sadu, shodný přístup k prostředkům a perifériím. Program vytvořený pro jednu verzi PCD poběží bez úprav i na všech ostatních (samozřejmě s přihlédnutím k dostupnosti prostředků, např. počtu procesorů). Jednotlivé modely se liší velikostí a stupněm modularity:

- PCD1 je kompaktní přístroj se čtyřmi pozicemi pro instalaci plochých vstupně-výstupních modulů v jedné rovině se základní deskou. V základním provedení je osazen jedním procesorem, 17 kB paměti rozšiřitelné až na 140 kB, může obsahovat 1 až 2 sériové kanály. Je v úspornou verzi jinak velmi podobného modelu PCD2.
- PCD2 obdobně jako PCD1 je plochý, kompaktní, s možností osazení osmi modulů, čtyři na horním i spodním okraji základní desky. Paměť pro jediný procesor je 32 – 536 kB velká, má 1 – 4 sériová rozhraní.
- PCD4 Již v pravém smyslu modulární, tvořen bloky montovanými na DIN lištu. 1 nebo 2 procesory, paměť 64 – 428 kB, 1 – 4 sériové porty.
- PCD6 je tvořen moduly ve „vanách“ vkládaných do 19“ skříní. 1 až 7 procesorů má k dispozici od 256 kB do 1 MB paměti, 4 až 28 sériových kanálů.

Všechny modely shodně používají procesor Motorola 68340, instrukční sada obsahuje přes 120 instrukcí včetně obsluhy komunikací a aritmetiky s plovoucí desetinnou čárkou, což je významným kritériem při realizaci číslicových regulátorů.

2.3. Programovatelný automat SAIA PCD2

2.3.1. Architektura PCD2

SAIA PCD2 je programovatelný automat kompaktní konstrukce. Základní jednotka obsahuje hlavní desku se všemi aktivními prvky, za ní je sběrníková deska s konektory pro připojení vstupně-výstupních modulů – čtyři na horním okraji a čtyři na dolním. Celek je uzavřen v ploché skřínce, jež umožňuje přístup jen ke konektoru PGU (Programming Unit) pro připojení programovacího zařízení a svorkovnicím jednotlivých modulů na stranách

přístroje. Světlovody přivádějí od LED diod na plošných spojích na povrch optickou indikaci stavu základní jednotky a jednotlivých vstupů a výstupů. Viz obr. [14](#) v příloze A.

Po sejmutí krytu se zpřístupní svorkovnice pro napájení přístroje 24V₊, komunikační rozhraní a kontakty relé watchdogu. Nyní lze na kontakty sběrnice desky zasunout I/O moduly. Na hlavní desce jsou také dvě pozice k instalaci rozšiřujících piggybacků s komunikačními a zobrazovacími obvody a patice pro rozšíření paměti (RAM i EPROM). Při vypnutí napájení jsou paměti zálohovány knoflíkovým článkem. Viz obr. [15](#) v příloze A.

Datová komunikace mezi hlavní deskou a přídatnými moduly probíhá pomocí adresovaných kanálů. Každý I/O modul má vyhrazeno 16 adres, podle pozice, na níž je instalován. Adresace začíná v levém horním rohu sběrnice desky od bazové adresy 0 a pokračuje ve směru hodinových ručiček. Pro každý modul je bazová adresa zvyšována vždy o 16. Toto pravidlo platí vždy, nezáleží na tom, zda modul využívá 16, 8, 6 nebo jen 4 adresy. V případě potřeby připojení více než osmi modulů lze použít rozšiřující jednotku, která obsahuje sběrnici s dalšími osmi pozicemi s bazovými adresami 128 – 240. Viz obr. [16](#) v příloze A.

Pozn.: Protože obvodu watchdog na hlavní desce je interně přiřazena adresa 255, nesmí být na adrese 240 instalován modul, který používá všech 16 adres.

2.3.2 Dodávané typy vstupně-výstupních modulů

- *komunikační rozhraní* (RS 422, RS 485, RS 232, proudová smyčka 20mA) (jako piggyback)
- *Hodiny reálného času + displej* (jako piggyback)
- *Dvouhodnotové vstupní a výstupní moduly* – 4 až 8 vstupů nebo výstupů, 24V₊ nebo 250 V_~, proud až 2A. Galvanicky neoddělené i optočleny oddělené vstupy, výstupy s tranzistory MOSFET nebo relé.
- *Analogové vstupní a výstupní moduly* s 4 až 8 kanály s rozlišením A/D, D/A převodníků 8 až 12 bitů pro signály 0 – 10V, -10 – 0V, -10 – 10V, 0 – 20mA, -20 – 0mA, -20 – 20mA, 4 – 20mA a pro přímé připojení teploměrů Pt, Ni 100, 1000
- *Čítací a polohovací moduly*
- *Zobrazovací moduly a terminály*

2.3.3. Typy datových prostředků využitelných v programech

Data a hodnoty, které program používá jsou striktně rozděleny do datových typů. Do paměti nelze přistupovat libovolně, ale vždy jen k prvku určitého typu a pořadového čísla (adresy) v rámci typu. Tabulka 1 obsahuje přehled použitelných typů dat.

Tab. 1 Přehled datových typů prostředků programovatelného automatu SAIA

Kód typu	Typ	Popis	Rozsah adres
I	Vstup (Input)	Hodnota fyzického dvouhodnotového vstupu I/O modulu (adresa podle umístění modulu)	0..8191
O	Výstup (Output)	Slouží k nastavení nebo přečtení stavu fyzického dvouhodnotového výstupu I/O modulu (adresa podle umístění modulu). I a O sdílejí stejné adresy.	0..8191
F	Flag	Binární proměnná	0..8191
T	Časovač (Timer)	Proměnná k nastavení nebo čtení stavu časovače	0..450
C	Čítač (Counter)	Proměnná k nastavení nebo čtení stavu čítače. T a C sdílejí adresy, je možno si zvolit, co bude čítač a co časovač.	0..1599
R	Registr (Register)	Obecný číselný registr (32 bitů). Uložená hodnota může být celé nebo reálné číslo	0..4095
K	Konstanta (K constant)	Podle přítomnosti desetinné tečky se rozlišují celá a reálná čísla	
X	Text	Slouží k ukládání řetězců do paměti	0..7999
DB	Datový blok (Data Block)	Slouží k ukládání bloků dat do paměti. Výhodné pro větší množství dat, používají se speciální instrukce. X a DB sdílejí stejné adresy, je nutno zvolit, zda půjde o řetězec nebo data.	0..7999

Uvedený rozsah adres platí vždy jen v rámci typu (kromě výjimek sdílení adres). Např. Flag F15 je uložen jinde než registr R15. V programech lze samozřejmě využívat možnosti přiřazení symbolických označení (např. direktivou EQU v Instruction List).

2.3.4. Struktura uložení programu

Podobně jako k datům a proměnným lze přistupovat pouze systematicky pomocí adres v rámci datového typu, program se musí zapisovat formou bloků různých typů podle určení. Tabulka 2 obsahuje přehled použitelných bloků.

Tab. 2 Přehled typů programových bloků

Kód typu	Typ bloku	Popis	Rozsah číslování
COB	Cyklický organizační blok (Cyclic Organization Block)	Části programu, umístěné v těchto blocích, jsou cyklicky prováděny (postupně se vykoná COB0, COB1 až po COB15 a znova od COB0)	0..15
XOB	Blok výjimky (Exception Organization Block)	K jednorázovému spuštění bloků tohoto typu dochází při výjimečných událostech přiřazených každému číslu XOB. Např. XOB0 se provede při výpadku napájení, XOB13 při výskytu chyby, XOB16 při studeném startu PCD	0..31
PB	Programový blok (Program Block)	Blok podprogramu (procedure). Volá se z jiného bloku a po provedení se vrací zpět.	0..299
FB	Funkční blok (Function Block)	Funkce – použití stejné jako u PB, při volání lze navíc předávat parametry.	0..999
SB	Sekvenční blok (Sequential Block)	Blok pro programování sekvenčních algoritmů (Graftec), obsahuje IST, ST a TR	0..31
IST	Úvodní krok (Initial Step)	Provádí se na začátku sekvence definované SB	0..1999
ST	Krok (Step)	Blok obsahující krok – dílčí část sekvence	0..1999
TR	Přechod (Transition)	Blok obsahující podmínky, na jejichž základě se rozhoduje o přechodu na další krok	0..1999

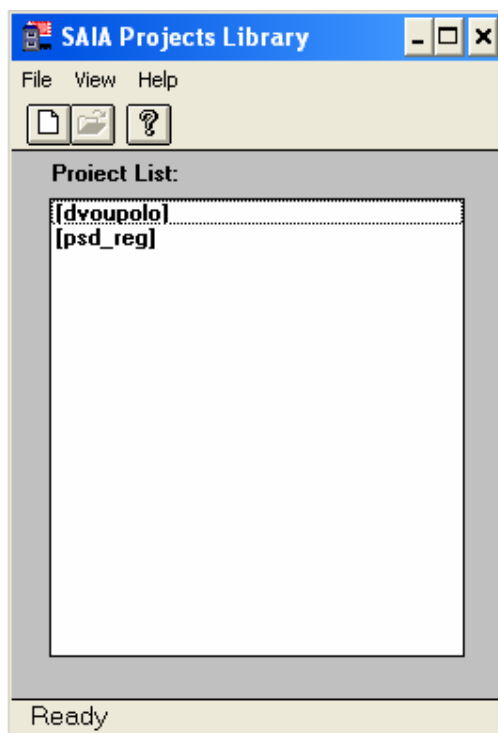
V programu se nesmějí vyskytovat bloky stejného čísla v rámci typu.

2.4 Vývojové prostředí PG4

K tvorbě programů pro rodinu programovatelných logických automatů SAIA PCDx výrobce dodává vývojové prostředí PG4 běžící na PC s operačním systémem MS Windows 3.x, 95, NT, 2000, XP.... (Předešlá verze PG3 byla určena pro MD-DOS). Jedná se o ucelený balík nástrojů pro tvorbu jednotlivých rutin v různých programovacích jazycích, jejich kompilování, správu knihovny takto vzniklých objektů, linkování programů až správu celých projektů. Pomocí PG4 lze navíc on-line připojením k PCD měnit konfiguraci logického automatu, nahrávat do jeho operační paměti vytvořené programy, tyto spouštět a za chodu monitorovat na obrazovce PC či přímo krokovat až na úrovni nejzákladnějších funkčních bloků.

Knihovna projektů

Po spuštění vývojového prostředí SAIA PG4 se objeví okno knihovny projektů (SAIA Projects Library). Toto je nejvyšší úroveň, zde se k další práci vybírá jeden ze samostatných již vytvořených projektů nebo se zakládá nový.



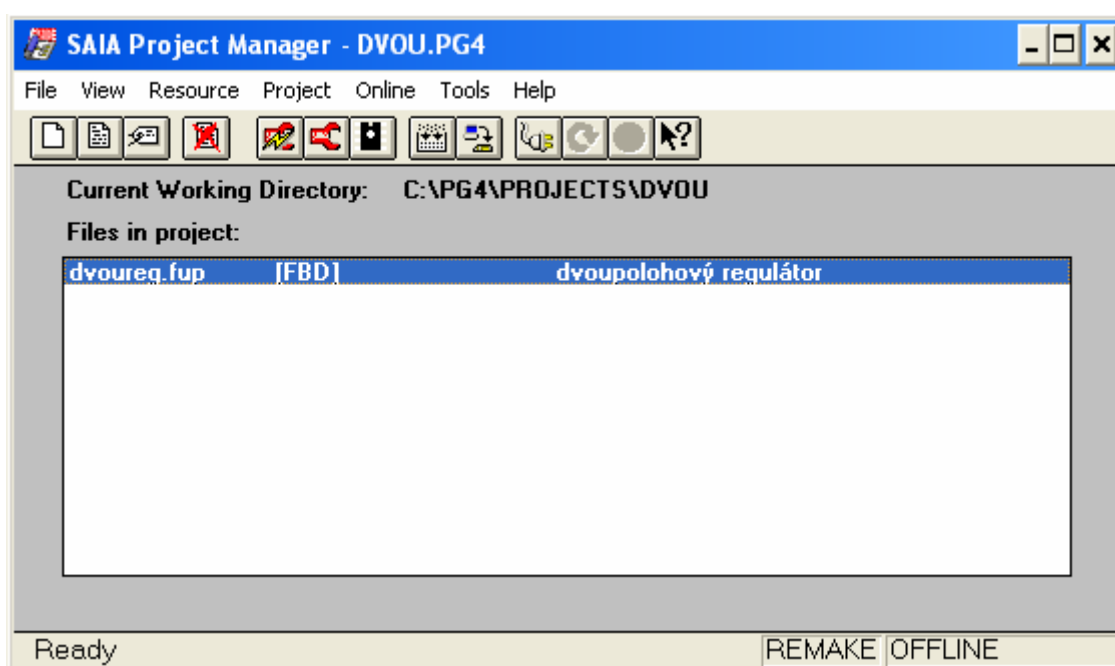
Obr. 1 Okno knihovny projektů

Po zvolení jednoho z projektů s ním lze začít pracovat ve Správci projektu (SAIA Project Manager)

Správce projektu

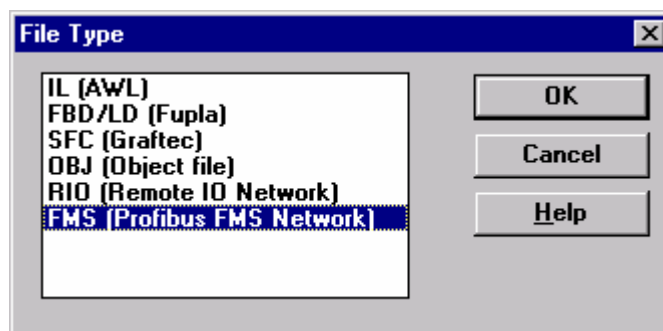
Okno správce projektu obsahuje seznam programových modulů v tomto projektu obsažených, z nichž se sestavuje výsledný program. V případě potřeby lze samozřejmě určitý modul ze sestavování vyjmout, aniž by se musel zcela odstranit z projektu.

Správce je jádrem PG4. Odtud lze linkovat program, připojit se on-line k PCD, nahrát do něj program a následně jej spustit, krokovat. K dispozici jsou zde také nástroje ke konfigurování paměťových zdrojů PCD – rozvržení prostoru RAM pro dynamické proměnné, čítače, časovače, uživatelské texty a datové bloky. Online Configurator umožňuje přímo nastavovat systémové prostředky logického automatu - vnitřní obvod reálného času, parametry komunikačních rozhraní, případně zablokovat další přístup k paměti PCD heslem (pro zabránění neoprávněného kopírování či modifikace software).



Obr. 2 Okno správce projektu

Při vytváření nového objektu se volí programovací jazyk, v němž bude vytvořen, a tím i editor, který se pro něj bude používat:



Obr. 3 Výběr typu programového objektu

- **IL (AWL)** - Instruction List (assembler), editor AWL (textový editor)
- **FBD/LD** - Function Block Diagram/Ladder Diagram (schéma funkčních bloků), editor FUPLA (Function Plan)
- **SFC** - Grafcet – grafický jazyk pro programování sekvencí bloků

Dalšími možnostmi jsou:

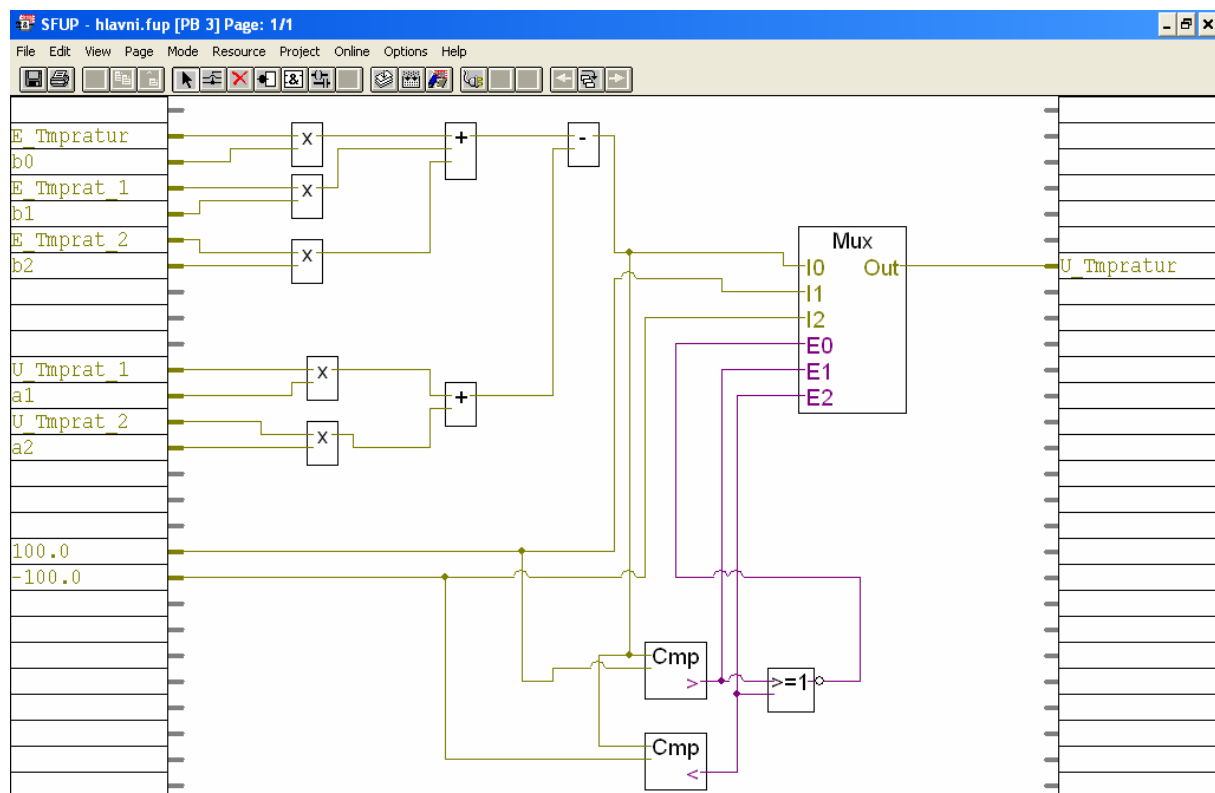
- **OBJ** - předem zkompilovaný objekt k zahrnutí do výsledného programu
- **RIO** - Remote IO Network
- **FMS** - Profibus FMS Network – objekty připojené ze vzdáleného zdroje po síti

2.4.1 Instruction List (IL)

Instruction List (výpis instrukcí) je forma zápisu programu přímo v základních instrukcích. K zápisu slouží obyčejný textový editor, řádky se nečíslují, komentáře se uvozují středníkem.

2.4.2 Function Plan – FUPLA (FBD)

FUPLA (Function Plan) je grafický jazyk založený na makrech s výhodou dynamického přiřazování datových prostředků.



Obr. 4 Okno editoru FUPLA

Do „sběrnice“ na levém okraji strany se vepisují vstupní data bloku – konstanty, registry, flagy, vstupy, výstupy, čítače, časovače apod. a to buď absolutně, tj. kódem datového typu a číslem, nebo symbolicky. Podobně pravá „sběrnice“ obsahuje prostředky přijímající výstupní hodnoty.

Program je sestaven z funkcí, reprezentovaných obdélníky s popisem, vstupy vlevo a výstupy vpravo (FBox). Předávání dat mezi nimi definují spojnice. Funkce se provádějí postupně z levého horního rohu dolů, při zachování logické návaznosti. Jsou definovány v knihovnách, dodávaných s vývojovým prostředím, v nichž je uložen jim odpovídající kód ve formě výpisu instrukcí a symbolickým vyjádřením vstupujících a vystupujících dat. Při překládání programu se použije tento kód jako makro (překopíruje se) a symboly se nahradí konkrétními čísly datových prostředků. Opět se čerpá z oblastí rezervovaných pro dynamické přiřazování.

Každý soubor v jazyce FUPLA může obsahovat více bloků. Každý blok může mít rozsah více stran.

Jazyk FUPLA je přehledný, programování v něm je jednoduché, ale některé nedostatky přinášejí omezení:

- nelze použít skoky v programu, podmíněné ani nepodmíněné, což zcela znemožňuje realizaci cyklu
- ze stejného důvodu lze podmíněně jen volat podprogramy, i jednoduchý program se tak může stát velmi nepřehledným
- při volání funkčních bloků (FB) jim nelze předat parametry, což je degraduje na procedury (PB)

Jako vyšší jazyk přirozeně FUPLA oproti Instruction List vytváří redundantní kód, to je však samozřejmá daň za používání knihoven funkcí k univerzálnímu použití.

2.4.3 Graftec (SFC)

Graftec je grafický jazyk pro zápis sekvenčních algoritmů. Vývojový diagram určuje postup provádění dílčích kroků a podmíněných přechodů mezi nimi. Procedury kroků (ST) i vyhodnocování přechodů (TR) jsou napsány buď v jazyce IL nebo FUPLA.

[13]

2.5 Control web 2000

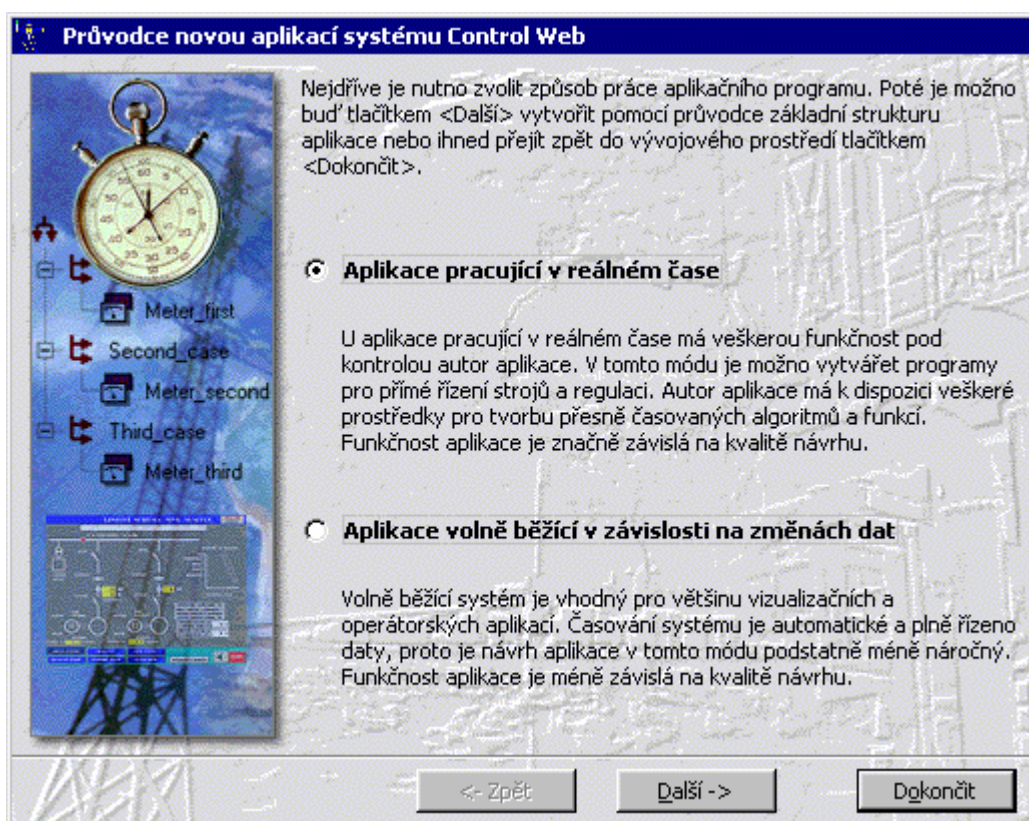
Control Web je otevřený komponentový průmyslový řídicí a informační systém reálného času pro operační systémy Windows. Je to univerzální nástroj pro vývoj a nasazování vizualizačních a řídicích aplikací, aplikací sběru, ukládání a vyhodnocování dat, aplikací rozhraní člověk-stroj. Unikátní objektově-orientovaná komponentová architektura zajišťuje aplikacím systému Control Web 2000 nejširší rozsah nasazení od prostých časově nenáročných vizualizací až po řídicí aplikace reálného času.

Control Web 2000 pracuje jako spousta jiných SCADA/HMI systémů používaných v průmyslu. K dispozici jsou všechny komponenty nutné k tvorbě vizualizačních aplikací - zobrazovací a ovládací prvky, alarmy a archivy, historické trendy apod. Navíc ale poskytuje skutečnou programovatelnost a otevřenou, komponentovou architekturu. Množina virtuálních přístrojů není pevně dána a zabudována v systému. Každý přístroj je dynamicky linkovaná knihovna detekovaná při startu systému. Není problém množinu virtuálních přístrojů libovolně rozšiřovat. Control Web 2000 umožňuje práci v reálném čase. Nespoléhá se na tzv. databázi reálného času, která je doplňována "maximální možnou rychlostí" (což v praxi může znamenat i intervaly několika desítek sekund mezi komunikacemi s automaty připojenými

přes DDE). Každý vstupně/výstupní kanál je čten přesně v době, kdy jej nějaký virtuální přístroj (nebo skupina virtuálních přístrojů) požaduje. Real-time časování je přesně monitorováno a řízeno.

2.5.1 Vytváření programů

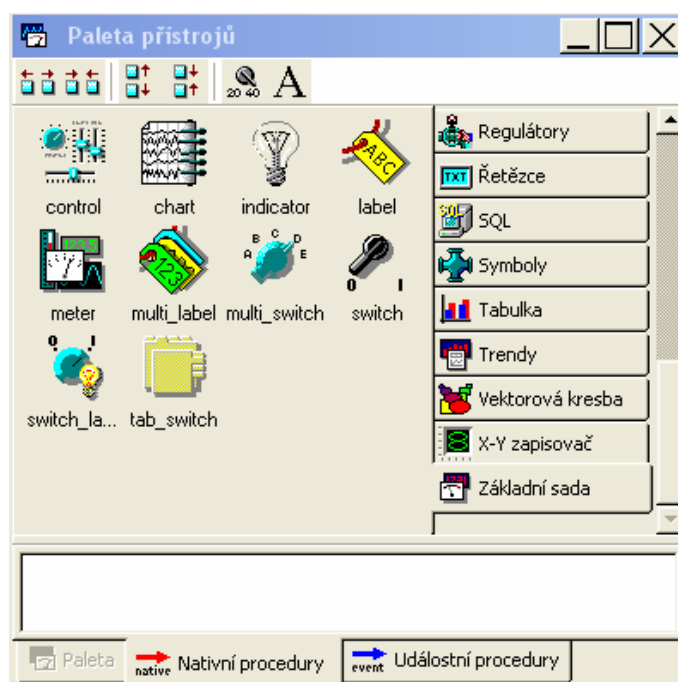
Ve vývojovém prostředí systému Control Web není autor aplikace nikdy ponechán svému osudu nad prázdnou pracovní plochou. Již od samého počátku tvorby aplikace je zde k dispozici účinná rada a pomoc v podobě Průvodce novou aplikací. Tento průvodce pracuje shodně jak v grafickém tak i v textovém módu vývojového prostředí



Obr 5 – Průvodce novou aplikací

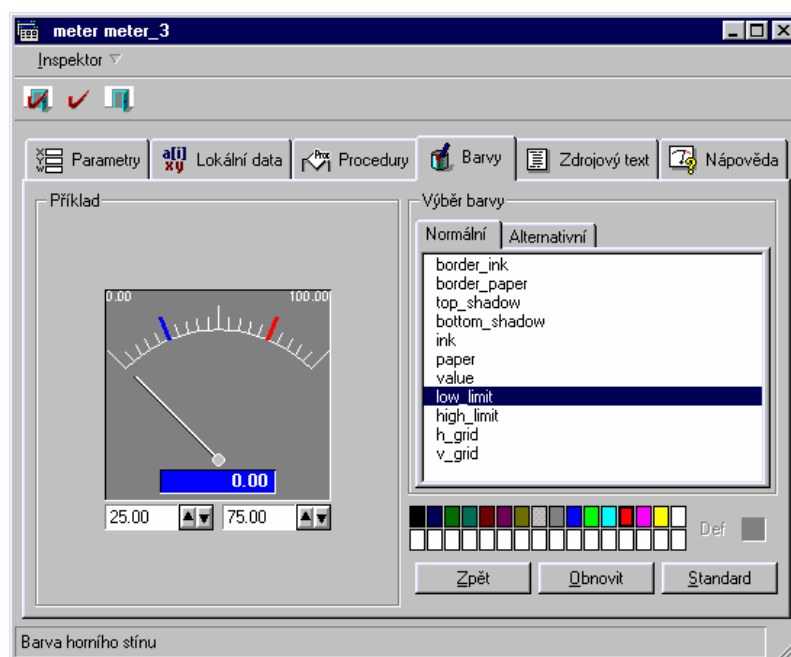
2.5.2 Virtuální přístroje

Aplikační program v prostředí systému Control Web je sestaven z jednotlivých virtuálních přístrojů. Každý virtuální přístroj je zcela samostatnou programovou komponentou.



Obr. 6 Paleta přístrojů

Každý přístroj lze editovat pomocí tzv. Inspektoru přístroje. Je to vlastně dialogové okno s několika záložkovými plochami.



Obr 7 Inspektor přístroje

2.5.3 Konstanty, proměnné a kanály

Řada virtuálních přístrojů potřebuje při své činnosti odněkud přečíst data a výsledek jejich zpracování zase někam uložit. Data systém Control Web uchovává v tzv. datových elementech. Datové elementy mohou být buď globální nebo lokální. Globální jsou společné pro celý aplikační program nebo programový modul. Lokální jsou vlastní jednomu virtuálnímu přístroji.

Z hlediska vlastností a způsobu použití rozlišujeme tři základní druhy datových elementů:

Konstanty – slouží pro uchování neměnných dat. Konstanty trvale obsahují svou inicializační hodnotu, kterou za běhu programu již nelze měnit. Konstanty mohou být globální i lokální.

Proměnné – slouží pro uchování dat, která se mění během činnosti aplikačního programu. Do proměnné lze kdykoliv zapsat novou hodnotu. Proměnné mohou být globální i lokální.

Kanály – slouží pro přenos dat mezi aplikačním programem a vstupně/výstupními zařízeními. Každý kanál je spojen s patřičným ovladačem, který automaticky zajišťuje přenos dat z daného vstupně/výstupního zařízení, kdykoliv je obsah kanálu čten, a přenos dat do daného vstupně/výstupního zařízení, kdykoliv jsou do kanálu uložena nová data. S kanály lze tedy pracovat podobně jako s proměnnými - pro data v kanálech uložená systém Control Web automaticky zajišťuje propojení se vstupně/výstupními zařízeními. Podle směru přenosu dat rozlišujeme tři druhy kanálů:

Vstupní kanály – hodnoty vstupních kanálů jsou čteny ze vstupně/výstupních zařízení. Tyto kanály je možno pouze aplikačním programem číst a nelze do nich data zapisovat.

Výstupní kanály – hodnoty výstupních kanálů je možno aplikačním programem číst i zapisovat. Při čtení dat z těchto kanálů ale nejsou hodnoty získávány ze vstupně/výstupních zařízení. Je přečtena vždy naposledy zapsaná hodnota. Při zápisu dat jsou samozřejmě hodnoty přenášeny přes ovladače do vstupně/výstupních zařízení.

Obousměrné kanály – hodnoty těchto kanálů jsou při čtení získávány a při zápisu přenášeny přes ovladače do vstupně/výstupních zařízení.

Kanály mohou být vzhledem ke své vazbě na ovladače pouze globální. Ovladače vstupně/výstupních zařízení jsou totiž také v rámci aplikace nebo modulu globálními objekty. Ovladače také nemusí být nutně napojeny na nějaké vstupně/výstupní zařízení. Data

poskytovaná některými ovladači jsou např. výsledky složitých matematických modelů a simulací.

2.5.4 Připojení k PLC Saia

Aby bylo možné zajistit komunikaci programu Contol Web 2000 s programovatelným automatem SAIA, je nutné nainstalovat odpovídající ovladač pro PLC SAIA a poté nastavit konfigurační soubory ovladače. Aby bylo možné ovladač používat, je nutné vždy před spuštěním aplikace Control Web 2000 spustit také program *SCODRV32.exe*. Ovladač pracuje s tzv. mapovacím a parametrickým souborem. Mapovací soubor obsahuje informace o kanálech, které je možné používat, včetně standardně přidělených čísel. Tento soubor je charakteristický pro každý ovladač. Zasahovat lze do parametrického souboru. Ten obsahuje informace o použitém protokolu, dále o portu, ke kterému je zařízení připojeno a v neposlední řadě také uživatelem definovaných kanálech.

2.6. Regulátory

2.6.1 Regulátory v knihovnách PLC

V současné době lze na našem trhu nalézt spoustu výrobců PLC (viz kap. 0). Jejich knihovny obsahují většinou základní PID regulátor, popř. jeho PSD modifikaci. Podrobněji jsem se zaměřil na výrobce PLC Omron, Saia, Teco a.s. a Amit.

Teco

Knihovna PLC od tohoto výrobce obsahuje instrukci „PID“, pomocí níž lze řídit soustavy, u kterých je doba přechodového děje alespoň o řád větší než perioda vzorkování.

PID instrukce zajišťuje ve volitelných násobcích 10 ms výpočet hodnoty akčního zásahu podle algoritmu PSD. Ovládání algoritmu je zajištěno pomocí struktury proměnných, která je definována na registrech PLC. PID pracuje v zásadě podle diskrétní verze této rovnice

$$u(t) = K \left[e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right] \quad (1)$$

Řídicí algoritmus zajišťuje následující funkce:

- Beznárazové přepínání ručního i automatického režimu, které je prováděno na základě odhadu stavu řízené soustavy. Parametry regulátoru je možné měnit i v automatickém režimu.
- Přepínání druhého pásma proporcionality podle znaménka odchylky (určeno pro rychlejší potlačení překmitu regulované soustavy).
- Nastavení rozsahu zóny působení integrační a derivační složky v % rozsahu měřené veličiny.
- Nastavení pásma necitlivosti regulátoru, tj. povolení nových změn zásahů až od dané úrovně odchylky v % rozsahu měřené veličiny.
- Přírůstkové řízení polohových ventilů i bez nutnosti odměřování jejich polohy. Je-li poloha ventilu měřená, je prováděna korekce akčního zásahu podle skutečného polohy ventilu.
- Realizaci poměrové regulace, filtrace nebo lineární interpolace žádané hodnoty.
- Zadávání rozsahů akčního zásahu. Pro akční zásah je možné zadávat i omezení jeho přírůstku.

Amit

Výrobci tohoto PLC zařadili do knihoven instrukcí instrukci „PID“. Tato instrukce reguluje podle rovnice (1). Lze taktéž nastavit základní parametry regulátoru.

Omron

Tento výrobce poskytuje ve svých knihovnách instrukci „PID“, umožňující řízení podle diskrétní verze algoritmu PID (1). Umožňuje nastavení těchto parametrů: zesílení, integrační a derivační časová konstanta, perioda vzorkování, zda se jedná o dopředné nebo zpětné řízení, maximální a minimální akční zásah.

Tento výrobce ještě navíc poskytuje instrukce „BAND“, „LMT“ a „ZONE“, pomocí níž se víceméně dá realizovat dvupolohová regulace.

Saia

Programovatelné automaty od tohoto výrobce nabízejí knihovně základních instrukcí FUPLA instrukci „PID Fbox“. Tato instrukce je obdobou instrukce „PID“ od Teca. Poskytuje uživateli však pouze základní parametry pro nastavení regulátoru, jako periodu vzorkování, zesílení, integrační a derivační konstantu, pásmo necitlivosti, rozlišení akčního zásahu. Řídící algoritmus je stejný jako u Teca.

2.6.2 Dvoupolohový regulátor

Je to nespojitý regulátor, kdy akční zásah nabývá pouze hodnot 0 a 1, nebo-li zapnuto a vypnuto. Tohoto způsobu regulace se používá se všude tam, kde nejsou kladeny vysoké nároky na přesnost, často se s ním setkáme např. u domácích spotřebičů. Jeho předností je, že je jednoduchý a levný.

Regulátor bude pracovat podle rovnice:

$$\begin{aligned} u(t) &= +u_{\max}, \text{ pokud } w - y > 0 \\ u(t) &= 0, \text{ pokud } w - y < 0 \end{aligned} \quad (2)$$

Abychom zamezili hazardním stavům, regulujeme soustavu s určitou periodou vzorkování. Jako další parametr instrukce se zavádí tzv. pásmo necitlivosti, které nám určuje, v jak velkém intervalu kolem hodnoty w nebude akční člen reagovat.

[12]

2.6.3 PSD regulátor

Spojité regulátory PID se v praxi natolik osvědčily, že realizace jejich funkcí byla převedena i do číslicových algoritmů a běžně se při počítačovém řízení využívá. Při převodu funkcí byly spojitě operace (integrace, derivace) nahrazeny numerickými formulami. Spojitý polohový PID regulátor je běžně popisován ve tvaru

$$u(t) = u(0) + r_0 \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{d\tau} \right] \quad (3)$$

nebo ve tvaru

$$u(t) = u(0) + r_0 e(t) + r_{-1} \int_0^t e(\tau) d\tau + r_1 \frac{de(t)}{dt} \quad (4)$$

přičemž $e(t) = w(t) - y(t)$ a převod mezi (3) a (4) je

$$T_I = \frac{r_0}{r_{-1}}, T_D = \frac{r_1}{r_0} \quad (5)$$

kde $u(t)$ je akční veličina, $y(t)$ regulovaná veličina, $e(t)$ regulační odchylka a $w(t)$ žádaná hodnota regulované veličiny. Pro $t=k.T$

$$\begin{aligned} u(kT) = u_k = u_0 + r_0 e_k + \frac{r_0}{T_I} \int_0^{kT} e(\tau) d\tau + r_0 T_D \frac{de(t)}{dt} = \\ = u_0 + r_0 e_k + \frac{r_0}{T_I} I_k + r_0 T_D D_k \end{aligned} \quad (6)$$

Použijeme přírůstkový algoritmus

$$\begin{aligned} \Delta u(k) = u(k) - u(k-1) &= r_0(e_k - e_{k-1}) + \frac{r_0}{T_I}(I_k - I_{k-1}) + r_0 T_D(D_k - D_{k-1}) \\ \Delta u(k) &= r_0(e_k - e_{k-1}) + \frac{r_0 T}{T_I} e_k + r_0 T_D \frac{e_k - 2e_{k-1} + e_{k-2}}{T} = (r_0 + \frac{r_0 T}{T_I})e_k - (r_0 + 2\frac{r_0 T_D}{T})e_{k-1} + \frac{r_0 T_D}{T}e_{k-2} = \\ &= q_0 e_k + q_1 e_{k-1} + q_2 e_{k-2} \end{aligned} \quad (7)$$

V následující tabulce je uveden výpočet konstant PSD regulátoru pro levou a pravou obdélníkovou metodu:

	q_0	q_1	q_2
LOBD	$r_0 + \frac{r_0 T}{T_I} + r_0 \frac{T_D}{T}$	$-r_0 - 2\frac{r_0 T_D}{T}$	$\frac{r_0 T_D}{T}$
POBD	$r_0 + r_0 \frac{T_D}{T}$	$-r_0 + \frac{r_0 T}{T_I} - 2\frac{r_0 T_D}{T}$	$\frac{r_0 T_D}{T}$

Tab. 3 – Hodnoty konstant obdélníkových metod

Pro výpočet parametrů PID regulátoru z přenosu soustavy použijeme následující vztahy:

$$\begin{aligned} c_i &= e^{-\frac{T}{T_i}} \\ T_I &= \frac{c_1 + c_2 - 2c_1 c_2}{1 - c_1 - c_2 + c_1 c_2} T, T_D = \frac{c_1 c_2 T}{c_1 + c_2 - 2c_1 c_2}, T_w = \text{round}\left(\frac{T}{0,286}\right) \\ c_w &= e^{-\frac{T}{T_w}}, k_r = \frac{(1 - c_w) T_I}{k_1 T} \\ q_0 &= k_r \left(1 + \frac{T}{T_I} + \frac{T_D}{T}\right), q_1 = -k_r \left(1 + \frac{2T_D}{T}\right), q_2 = k_r \left(\frac{T_D}{T}\right) \end{aligned} \quad (8-16)$$

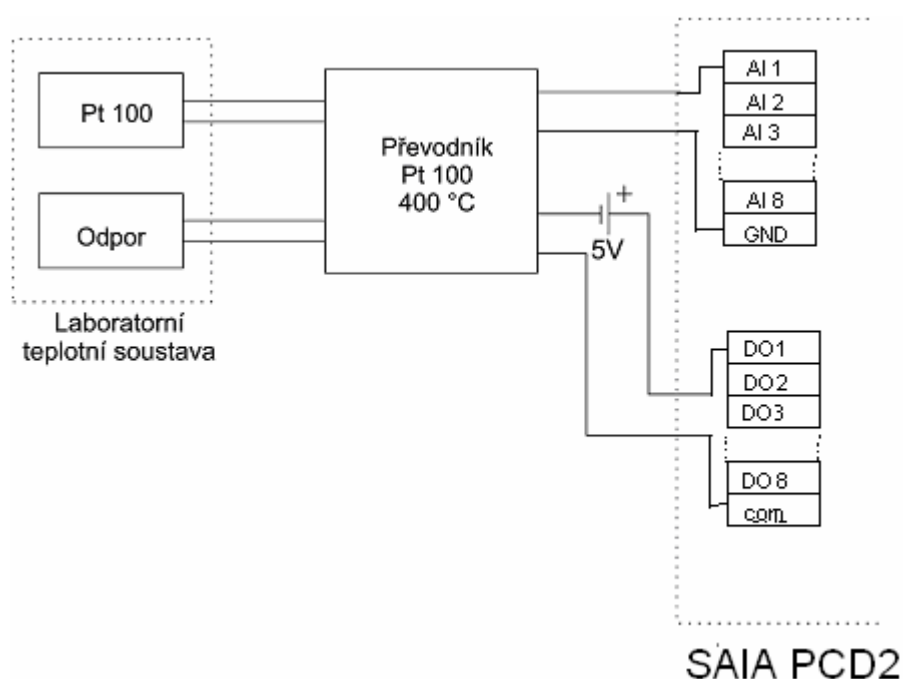
3. EXPERIMENTÁLNÍ ČÁST

Funkčnost nově vytvářených algoritmů byla ověřována na tepelné soustavě.

3.1. Laboratorní tepelná soustava

3.1.1. Popis laboratorní soustavy

Měřená veličina vstupuje do PLC jako analogový signál (Vstup AI 1). Akční zásah je realizován pouze dvěma stavy 0 – netopí a 1 – topí (Výstup DO 1).



Obr. 8 Schéma zapojení laboratorní soustavy do PLC

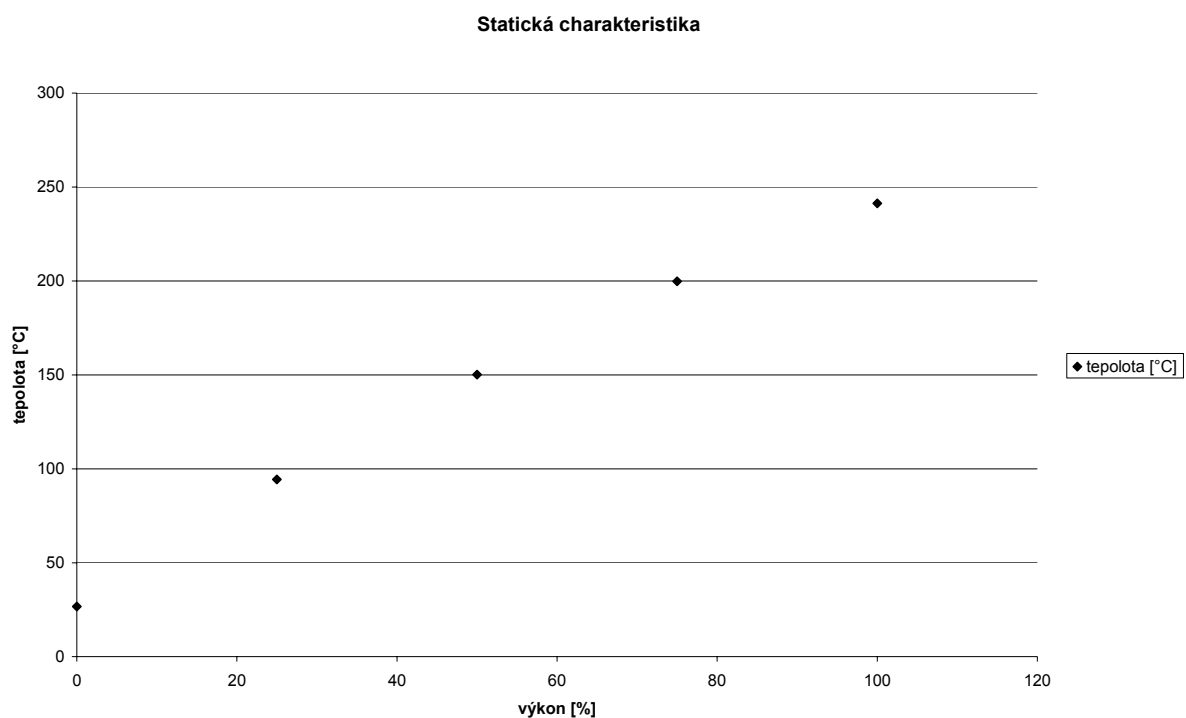
3.1.2. Identifikace laboratorní soustavy

Byla změřena maximální (ustálená) hodnota teploty při daných výkonech:

Výkon [%]	Teplota [°C]
0	26.66
25	94.34
50	150.15
75	199.81
100	241.26

Tab 4 – Tabulka hodnot statické charakteristiky

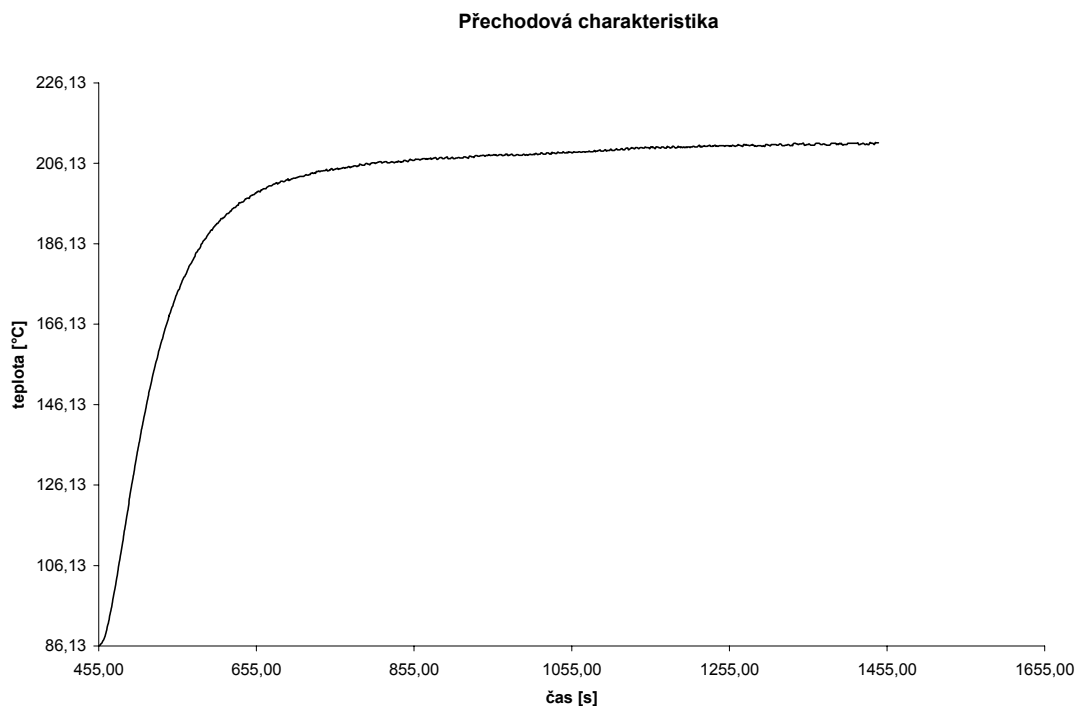
Z daných hodnot se určila statická charakteristika:



Obr. 9 Statická charakteristika

Rovnice této závislosti je: $y = 2,1395x + 35,488$

Lineární oblast statické charakteristiky je asi mezi 20 – 80 procenty výkonu, proto byla přechodová charakteristika změřena při změně akční veličiny z 20% na 80%.



Obr. 10 Přechodová charakteristika

Numerickou metodou aproximace přechodové charakteristiky (pomocí funkce „Řešitel“ v programu Microsoft Excel) byly získány tyto hodnoty:

$k = 2,082517$

$T1[s] = 79,04082$

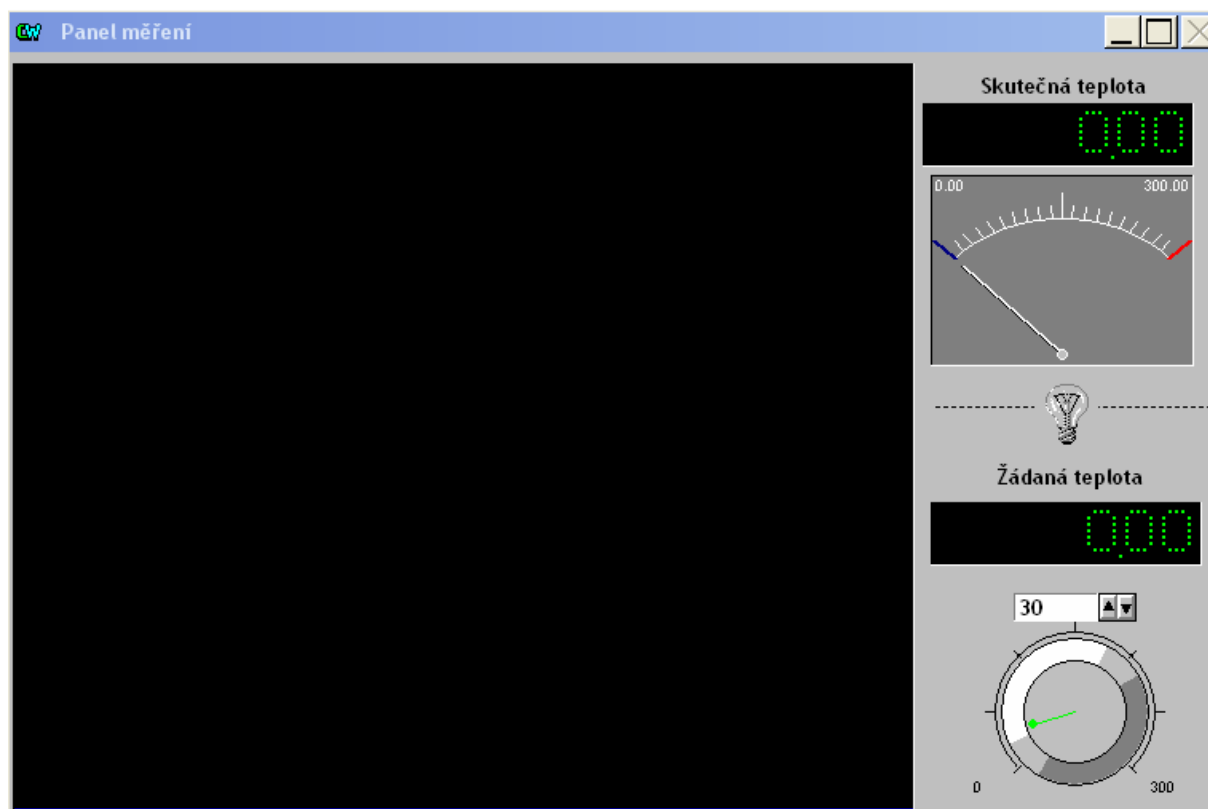
$T2[s] = 18.32918$

3.2 Obecný popis vytvořených programů regulátorů

Jednotlivé programy regulátorů jsou realizovány jako tzv. „functions plans (fu-pla)“, které podporuje vývojový software SAIA PG4. Tyto programy pro řízení regulovaných soustav je možno velmi lehce začlenit do jakéhokoliv jiného uživatelského programu a může je používat programátor, jenž je seznámen s parametry těchto instrukcí nutnými pro spolehlivé řízení.

3.3 Vizualizace řízeného procesu

Pro názornou vizualizaci průběhů jednotlivých veličin jsem zvolil vizualizační software ControlWeb 2000. Příklad programu pro regulaci tepelné soustavy je uveden na Obr. 11.



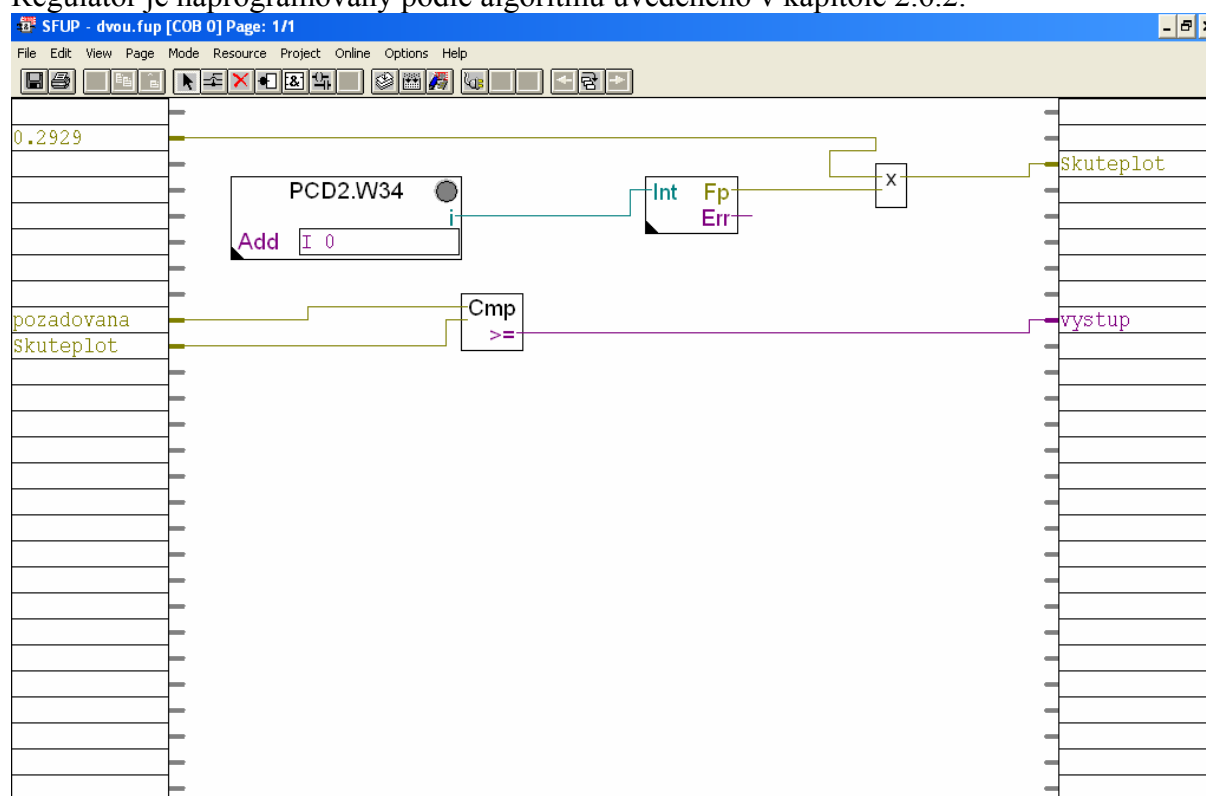
Obr. 11 Okno řídicího programu

Program Control Web 2000 čte jednotlivá data z PLC pomocí tzv. parametrického souboru, ve kterém musí programátor přiřadit kanálům adresy v PLC, ze kterých jsou načítána popř. zapisována data. V mapovacím souboru programátor přiřazuje jednotlivým

množinám kanálů různé datové typy. Příklad parametrického souboru a mapovacího souboru je uveden v Příloze [B](#) a [C](#).

3.4 Dvoupolohová regulace

Regulátor je naprogramovaný podle algoritmu uvedeného v kapitole 2.6.2.

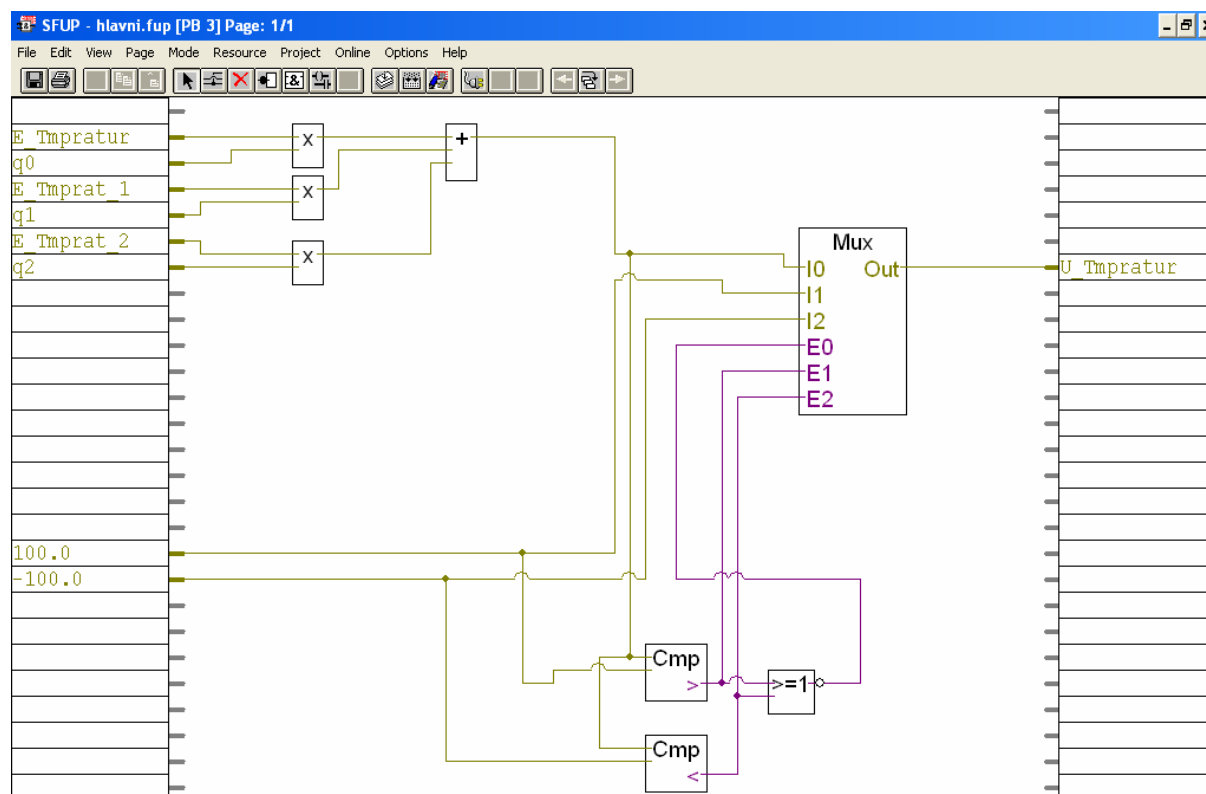


Obr. 12 Zapojení dvoupolohového regulátoru

Na obr.10 je znázorněna dvoupolohová regulace. Analogový signál o hodnotě výstupní veličiny regulované soustavy (v našem případě teploty) vstupuje do analogového modulu PCD2.W34 na kanálu Input 0 a je uschován jako proměnná *Teplota*. Ta je dále převedena z proměnné integer na proměnnou floating point a pronásobena konstantou 0,2929 (10-ti bitový převodník při rozsahu teplot 0-300 °C → konstanta je pak $300/2^{10}=0,2929$) a uložena coby proměnná *Cislo*. Ta je dále porovnávána s žádanou hodnotou uloženou v proměnné *Zadana*. Podle toho zda je žádaná hodnota (*Zadana*) větší nebo menší než okamžitá teplota (*Teplota*) zapisuje se do proměnné *Vystup* binární hodnota 1 nebo 0. Proměnná *Vystup* je spojena s kanálem Output 16 a udává zda má být proveden akční zásah, v našem případě zda má regulovaná soustava „topit“ či „netopit“.

Grafické znázornění průběhu regulačního pochodu a simulace v programu matlab simulink je uvedeno v příloze.

3.5. PSD regulace



Obr. 13 – Zapojení PSD regulátoru

Na obrázku 13 je znázorněna PSD regulace. Regulační odchylky e_k , e_{k-1} a e_{k-2} , spočítané v jiném bloku programu, jsou násobeny s konstantami regulátoru q_0 , q_1 a q_2 a jsou vzájemně sečteny. Výsledek vstupuje na vstup IO v bloku MUX. Další vstupy bloku MUX jsou 100 a -100 a pomocí jednoduchého algoritmu je zařízeno aby akční zásah na výstupu z bloku MUX ($U_{\text{Tempratur}}$) nebyl nikdy větší či menší než 100% eventuelně -100%. Akční zásah ($U_{\text{Tempratur}}$) je v dalších blocích programu přepočítán pomocí algoritmu pulzní šířkové modulační.

Grafické znázornění průběhu regulačního pochodu a simulace v programu matlab simulink je uvedeno v příloze.

4. ZÁVĚR

Předmětem této diplomové práce byla implementace nových algoritmů do řídicích procesů programovatelných automatů.

Úvodní teoretická část popisuje výhody i nevýhody použití programovatelných automatů obecně v procesech řízení v porovnání s ostatními možnostmi řešení. Pro PLC hovoří především jejich konstrukce, odolná proti rušivým vlivům, nepostradatelná v drsných průmyslových podmínkách, a modularita vstupně/výstupních obvodů, umožňující sestavení přístroje, jehož velikost, vlastnosti a cena odpovídají rozsahu a charakteru aplikace.

Dále byl ve středu zájmu konkrétní model programovatelného automatu – SAIA PCD2. Seznámení se s jeho architekturou a škálou nabízených rozšiřujících obvodů bylo základem k návrhu hardwarové konfigurace. Efektivní využití všech nabízených vlastností a funkcí bylo podmíněno nastudováním všech dostupných informací o způsobech a technologiích programování tohoto PLC. Jako nejvhodnější prostředky k vývoji software pro řízení procesů se jeví zápis kódu formou základních instrukcí a grafický jazyk FUPLA (Function Plan). Výhodou prvního způsobu je přímý zápis algoritmu na nejzákladnější programovací úrovni, bez redundantních kroků. Vyžaduje však hlubší znalost instrukcí procesoru. Naopak grafický nástroj FUPLA vyniká přehledností a možností intuitivního přístupu, v případě složitějších programů si ale vynucuje spoustu kompromisů a omezení, jenž vyplývají jak z filozofie jazyka, tak z nedostatků v aktuálních verzích knihoven funkcí.

Při začleňování programovatelného automatu do stávajícího systému jsem se snažil o co nejmenší zásahy do jeho struktury, ve smyslu jak elektrického schématu, tak datových rozhraní, což se mi podařilo. PLC je připojen konektorovými spoji k sériovému portu monitorovacího počítače a k rozváděcí desce výkonových modulů. V případě potřeby je snadné jej v těchto bodech zaměnit za původní PC. Na monitorovacím počítači nebyly nutné žádné programové změny, datové rozhraní je zcela shodné s původním.

Původním záměrem této diplomové práce bylo naprogramovat některé typy regulátorů pro programovatelný automat SAIA PCD2. Nejprve bylo nutné provést identifikaci regulované soustavy. Zadána byla laboratorní tepelná soustava. Soustavu se podařilo úspěšně identifikovat pomocí Strejcovy metody. Údaje získané při identifikaci byly použity pro syntézu vytvářených regulátorů. Tyto regulátory byly vytvořeny ve vývojovém prostředí SAIA PG4. Následná vizualizace byla realizována pomocí integrovaného vývojového prostředí Control Web 2000.

Vzájemná komunikace mezi aplikacemi SAIA PG4 a Control Web 2000 byla zajištěna pomocí ovladače SCODRV32 a datových kanálů. Nastavení kanálů lze konfigurovat prostřednictvím parametrického a mapovacího souboru (Saia.par resp. Saia.dmf). Propojení programovatelného automatu a PC bylo tímto úspěšně dokončeno.

Při vlastní implementaci zadaných regulátorů se vyskytly komplikace, když použitý operační systém Windows XP pravidelně po krátkém časovém období (řádově minuty) zhavaroval a bylo nutné provést kompletní restart počítače. Byla snaha tento problém vyřešit přeinstalováním programovacího prostředí za novější verzi (PG4 na PG5). Bohužel novější verze tohoto prostředí odmítla komunikovat s použitými zásuvnými moduly. Totéž se opakovalo při použití jiného PC s operačním systémem Windows 2000. Další práce probíhala na starším PC se systémem Windows 3.11 v původním programovacím prostředí PG4, kde se již výše uvedené problémy nevyskytovaly.

U programovatelného automatu SAIA PCD2 používaného při realizaci této diplomové práce se během aplikace regulátorů objevila neodstranitelná závada, kterou se nepodařilo diagnostikovat ani po pokusech nahradit vstupní a výstupní moduly, po proměření vstupních a výstupních signálů, ani po telefonické konzultaci s odborníkem, který denně pracuje s programovatelnými automaty SAIA. Bylo proto nutné tento programovatelný automat nahradit novým.

Plánované algoritmy regulátorů, které měly tvořit významnou část této diplomové práce, se z výše popsaných důvodů a tím pádem i z nedostatku času nepodařilo naprogramovat. Bylo proto nutné se omezit pouze na implementaci a ověření funkčnosti dvou regulátorů a to regulátoru dvoupolohového a PSD. Grafické ověření regulačních pochodů těchto regulátorů je součástí této diplomové práce a je uvedeno v příloze.

V započaté práci je možné dále pokračovat. Zejména oblast adaptivních regulátorů otevírá široké pole možností pro aplikace programovatelných automatů při řízení technologických procesů nejrůznějšího charakteru.

SEZNAM POUŽITÉ LITERATURY

- (1) Martinásková, M., Šmejkal, L.: Řízení programovatelnými automaty, Vydavatelství ČVUT, Praha, 1998
- (2) Bílý, R., Cagaš, P., Cagaš, R., Hladůvka, D., Kolařík, M., Sobotík, J., Zálešák, M., Zgarba, Z.: Control Web 2000, Computer Press, Praha, 1999
- (3) Bobál, V., Böhm, J., Prokop, R., Fessler, J.: Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace, VUTIUM, Brno, 1999
- (4) Balátě, J.: Vybrané statě z automatického řízení, Vydavatelství VUT, Brno, 1996
- (5) Bobál, V., Vašek, V.: Teorie automatického řízení I, Ediční středisko VUT Brno, Brno, 1990
- (6) Vašek, V.: Teorie automatického řízení II, Ediční středisko VUT Brno, Brno, 1990
- (7) Bobál, V.: Teorie automatického řízení III, Ediční středisko VUT Brno, Brno, 1990
- (8) Martinásková, M., Šmejkal, L.: Řízení programovatelnými automaty II, Vydavatelství ČVUT, Praha, 2000
- (9) SAIA-Burgess Electronics: Hardware – Řady PCD1 a PCD2. Vydání 26/737 C5, 1998
- (10) SAIA-Burgess Electronics: PCD Series Reference Guide (anglicky). Vydání 26/733 E6, 1997
- (11) SAIA-Burgess Electronics: Programming tools for MS Windows PG4 (anglicky). Vydání 26/748 E1, 1996
- (12) Dudr, T.: Aplikace nových algoritmů při řízení technologických procesů pomocí programovatelných automatů, Diplomová práce, UTB FT, Zlín 2003
- (13) Horných, J.: Aplikace PLC automatů pro řízení enzymatické hydrolýzy koželužských odpadů, Diplomová práce, VUT FT, Zlín 1999

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

t	čas
u	hodnota akčního zásahu
$u_{\max}$	maximální akční zásah
e	hodnota regulační odchylky
K, r_0	zesílení regulátoru
T_I	integrační časová konstanta
T_D	derivační časová konstanta
k	krok regulačního pochodu
w	žádaná hodnota
y	regulovaná veličina
T_0	perioda vzorkování
I_k	diskrétní náhrada spojitého integrálu
D_k	diskrétní náhrada derivace
q_0, q_1, q_2	konstanty regulátoru

SEZNAM OBRÁZKŮ

Obr. 1 - Okno knihovny projektů	17
Obr. 2 - Okno správce projektu	18
Obr. 3 - Výběr typu programového objektu	19
Obr. 4 - Okno editoru FUPLA	20
Obr. 5 – Průvodce novou aplikací	22
Obr. 6 – Paleta přístrojů	23
Obr. 7 – Inspektor přístroje	23
Obr. 8 – Schéma zapojení laboratorní soustavy do PLC	29
Obr. 9- Statická charakteristika	30
Obr. 10 – Přejchodová charakteristika	30
Obr. 11 – Okno řídicího programu	31
Obr. 12 – Zapojení dvoupolohového regulátoru	32
Obr. 13 – Zapojení PSD regulátoru	33
Obr. 14 - Programovatelný automat SAIA PCD2	41
Obr. 15 - Hlavní deska SAIA PCD2	42
Obr. 16 - Adresace modulů v hlavní a rozšiřovací jednotce	43
Obr. 17 – Výsledky řízení dvoupolohového regulátoru	47
Obr. 18 – Simulace řízení dvoupolohového regulátoru	47
Obr. 19 – Výsledky řízení PSD regulátoru	48
Obr. 20 – Simulace řízení PSD regulátoru	48

SEZNAM TABULEK

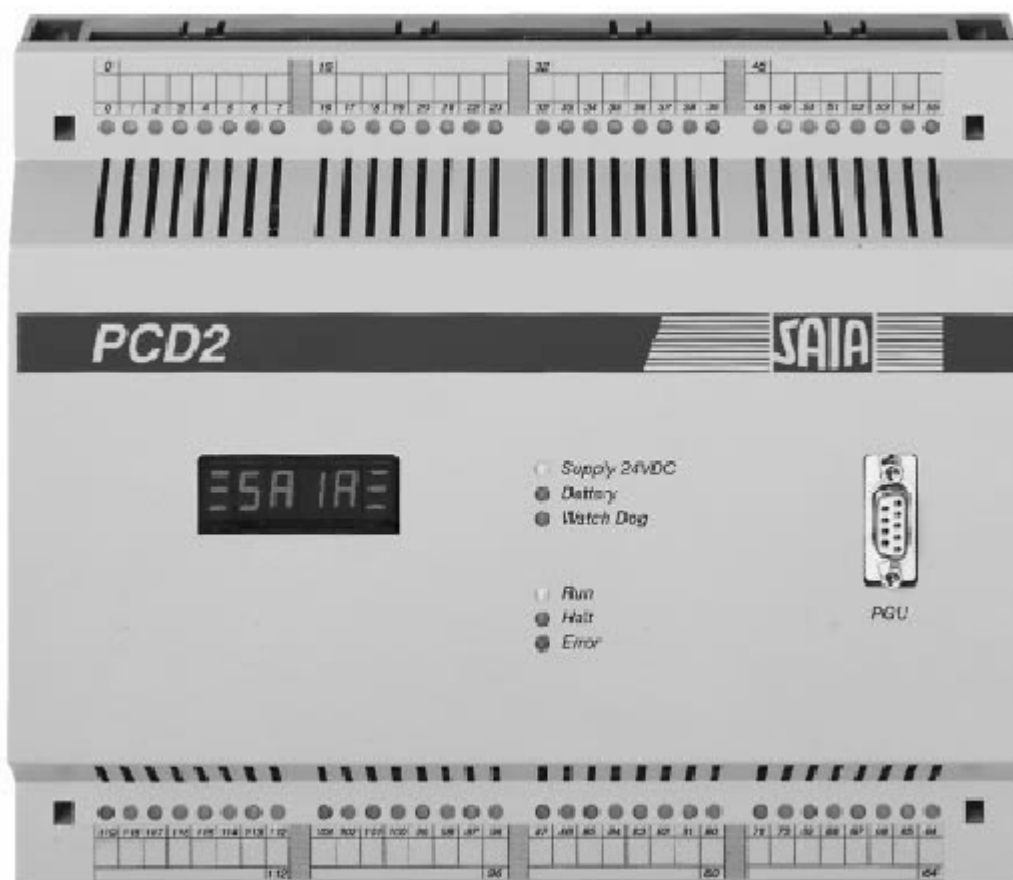
Tab. 1 - Přehled datových typů prostředků programovatelného automatu SAIA	15
Tab. 2 - Přehled typů programových bloků.....	16
Tab. 3 – Hodnoty konstant obdélníkových metod	28
Tab. 4 – Tabulka hodnot statické charakteristiky	29

SEZNAM PŘÍLOH

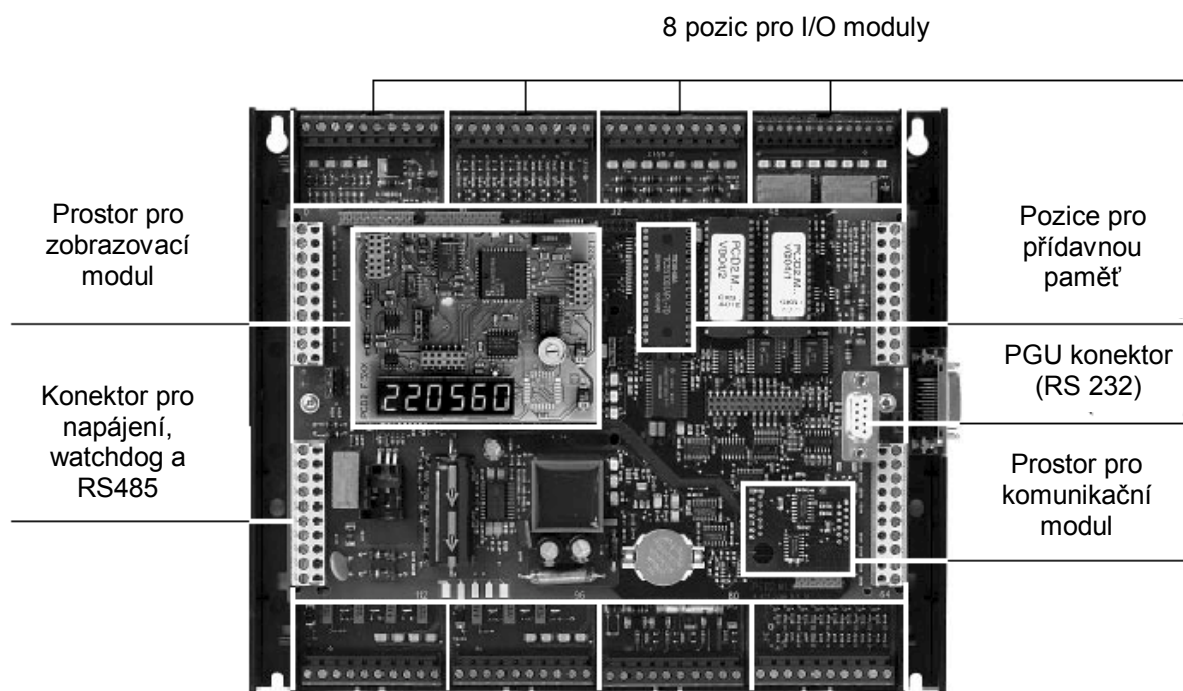
PŘÍLOHA A – OBRAZOVÁ DOKUMENTACE.....	41
PŘÍLOHA B – PARAMETRICKÝ SOUBOR.....	44
PŘÍLOHA C – MAPOVACÍ SOUBOR	45
PŘÍLOHA D – VÝSLEDKY ŘÍZENÍ DVOUPOLOHOVÉHO REGULÁTORU.....	47
PŘÍLOHA E – VÝSLEDKY ŘÍZENÍ PSD REGULÁTORU	48

PŘÍLOHY

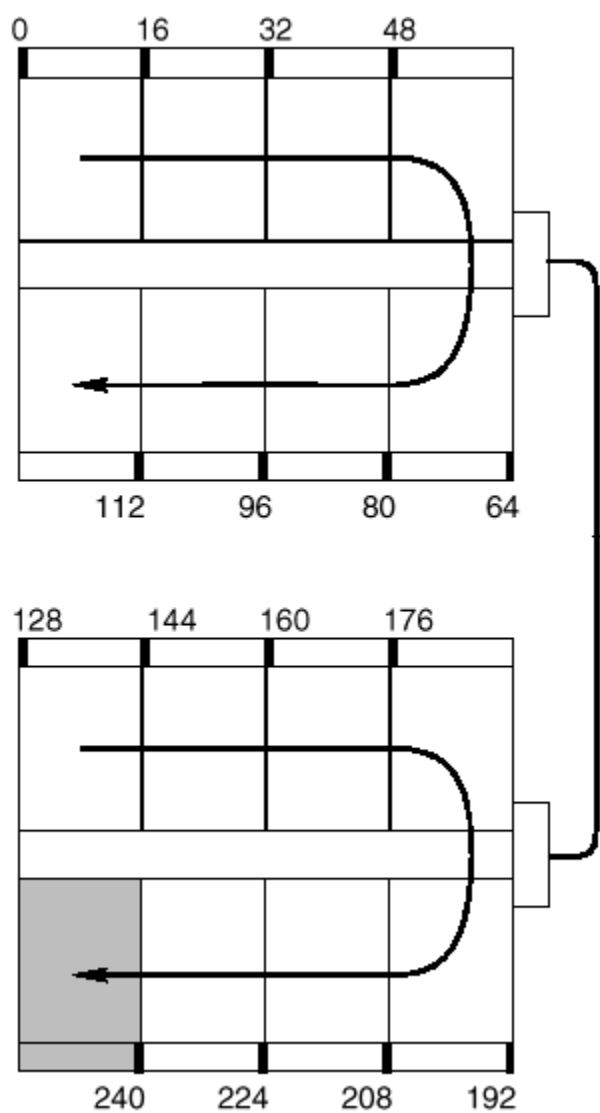
Příloha A – Obrazová dokumentace



Obr. 14 Programovatelný automat SAIA PCD2



Obr. 15 Hlavní deska SAIA PCD2



Obr. 16 Adresace modulů v hlavní a rozšiřovací jednotce

Příloha B – Parametrický soubor

```
[ComPort]
```

```
Com = COM1
```

```
Protocol = p800
```

```
SBusMode = Parity
```

```
BaudRate = 9600
```

```
Timeout = 200
```

```
RTS = enable
```

```
[Channels]
```

```
;-
```

```
-
```

```
;Block = Station, ChannelFrom, ChannelTo, Media, Offset,
```

```
[Subtype], [Bidirect]
```

```
;-
```

```
-
```

```
Block = 01, 100, 100, R, 0, float, bidirect
```

```
;skutecna
```

```
Block = 01, 101, 101, R, 1, float, bidirect
```

```
;zadana
```

```
Block = 01, 102, 102, R, 2, float, bidirect
```

```
;q0
```

```
Block = 01, 103, 103, R, 3, float, bidirect
```

```
;q1
```

```
Block = 01, 104, 104, R, 4, float, bidirect
```

```
;q2
```

```
Block = 01, 105, 105, R, 5, float, bidirect
```

```
Block = 01, 106, 106, R, 6, int, bidirect
```

```
;T0
```

Příloha C – Mapovací soubor

begin

1 real input

10 real input

11 real input

12 string input

13 real input

14 real input

15 real input

16 real input

17 boolean output

18 boolean output

20 real input

21 real input

22 string input

23 real input

24 real input

25 real input

26 real input

27 boolean output

28 boolean output

100 - 119 real bidirectional

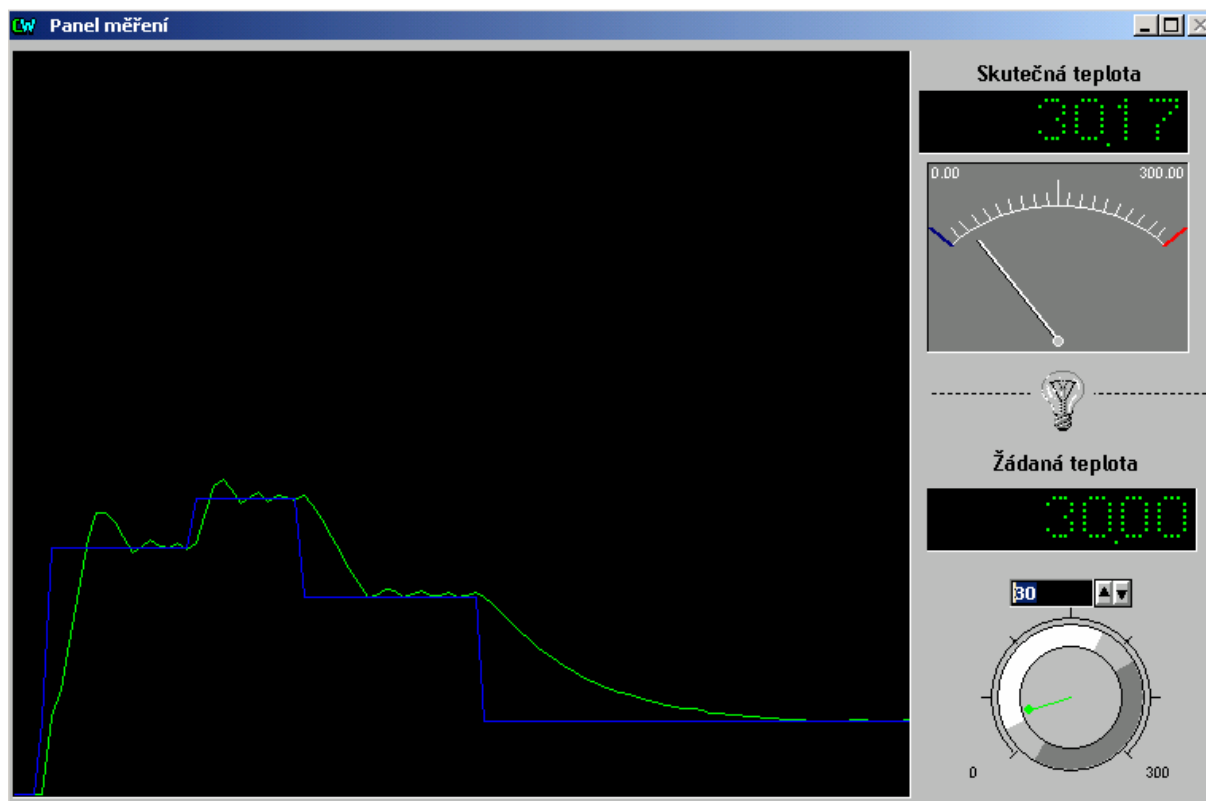
200 - 263 boolean bidirectional

900 - 917 real bidirectional

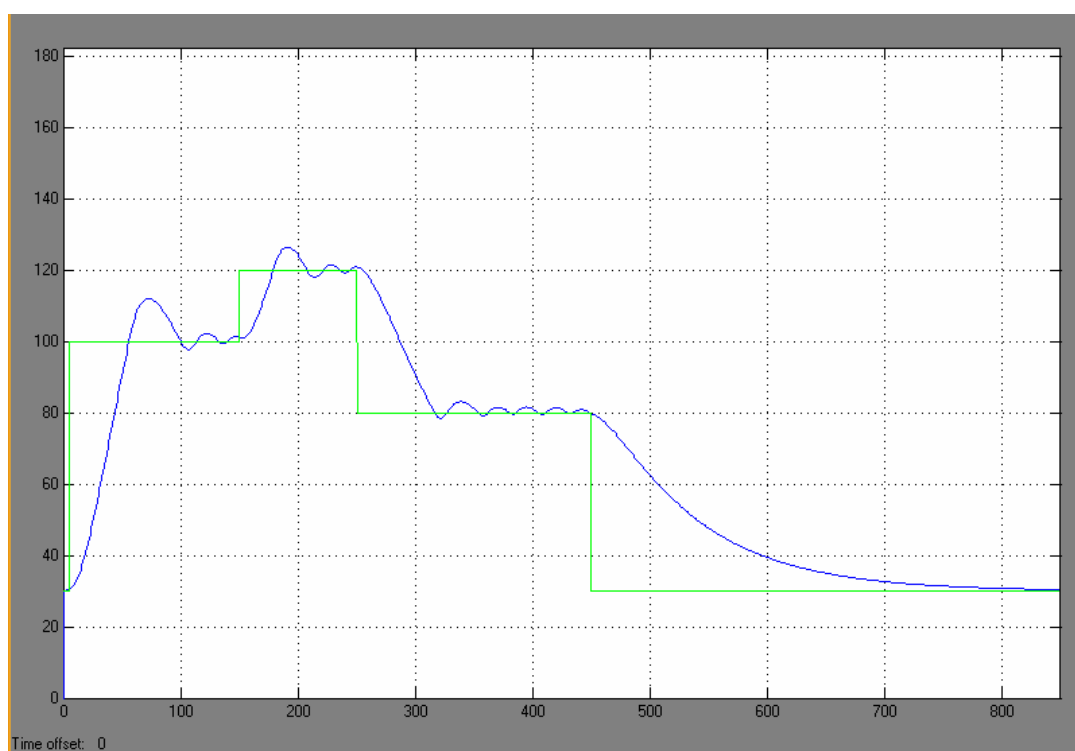
1000 - 1007 boolean bidirectional

1100 - 1107 boolean input

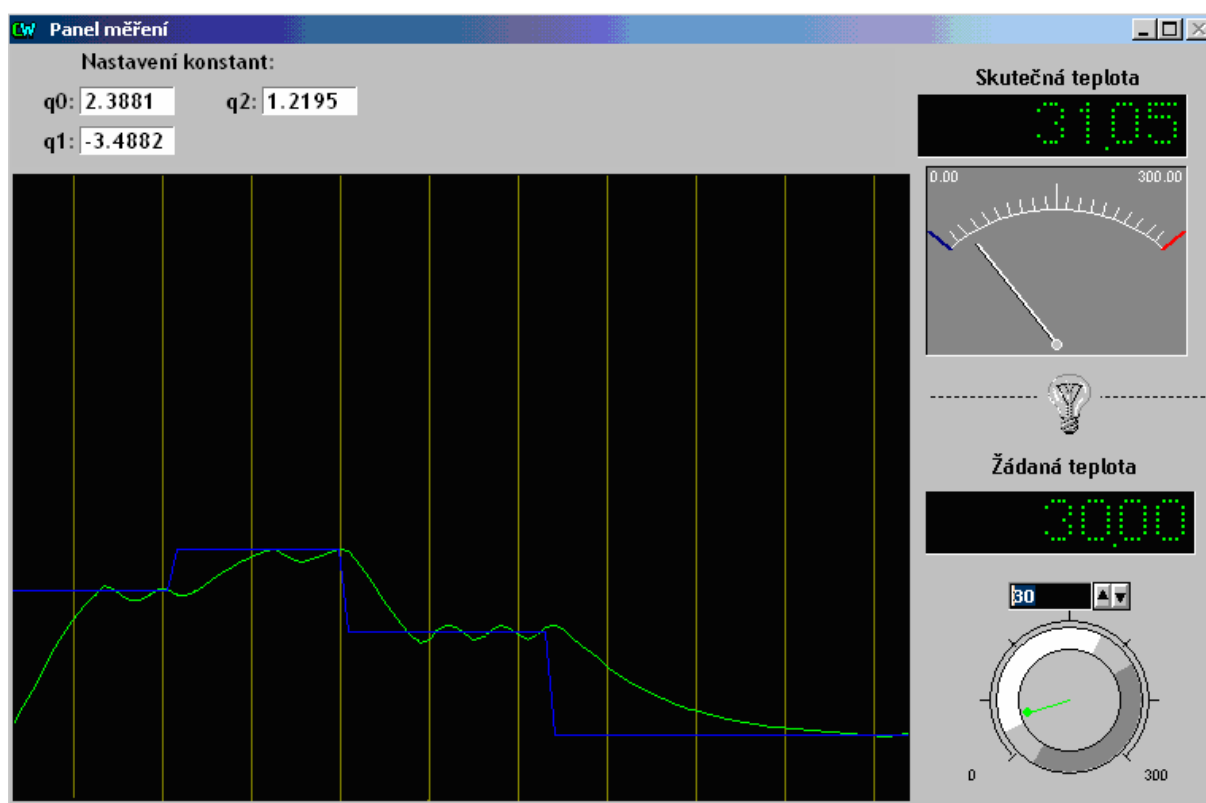
1200 - 1207 boolean input
end.

Příloha D: Výsledky řízení Dvoupolohového regulátoru

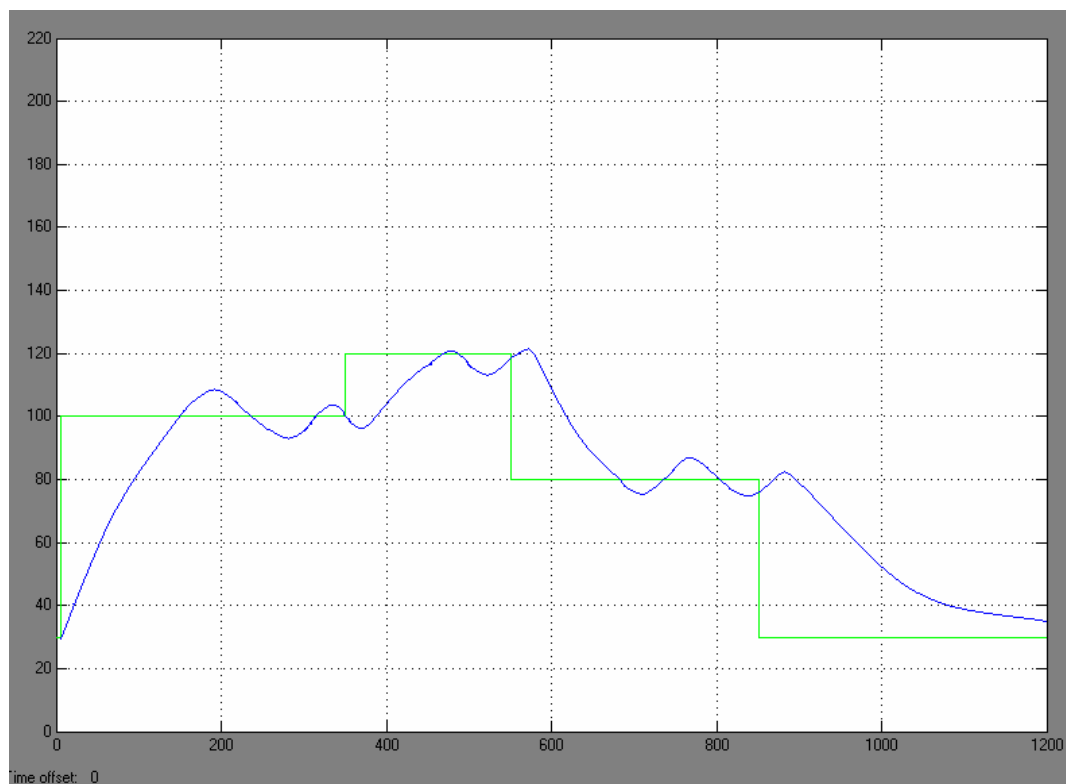
Obr. 17 – Výsledky řízení dvoupolohového regulátoru



Obr 18 – Výsledky simulace dvoupolohového regulátoru v programu Matlab-Simulink

Příloha E: Výsledky řízení PSD regulátoru

Obr. 19 – Výsledky řízení PSD regulátoru



Obr 20 – Výsledky simulace PSD regulátoru v programu Matlab-Simulink