

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

ŘÍZENÍ HYDRAULICKO-PNEUMATICKÉ SOUSTAVY POMOCÍ PLC
MODICON
Jiří Vinduška

Bakalářská práce

2017

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří Vinduška**
Osobní číslo: **I14063**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Řízení procesů**
Název tématu: **Řízení hydraulicko-pneumatické soustavy pomocí PLC Modicon**
Zadávající katedra: **Katedra řízení procesů**

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

Cíl: Cílem je vytvořit aplikaci pro PLC Modicon fy Schneider Electric, pomocí které bude možné řídit hydraulicko-pneumatickou soustavu.

Obsah teoretické části: Teorie logického řízení. Programovatelné logické automaty a inteligentní relé - hardware a programování. HMI interface.

Obsah implementační části: Konfigurace PLC, vytvoření programu pro PLC i HMI panel, který umožní v manuálním režimu ovládní soustavy a v režimu automat bude řídit hladiny v dolních nádržích pomocí dvou PID regulátorů.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

L. Šmejkal, M. Matrinásková, **PLC a automatizace 1, Základní pojmy, úvod do programování. BEN - technická literatura, Praha 1999**

L. Šmejkal, **PLC a automatizace 2, Sekvenční logické systémy a základy fuzzy logiky. BEN - technická literatura, Praha 2005**

M. Martinásková, **Programovací jazyky pro PLC. Automatizace, ročník 47, číslo 6, strana 380, 2004**

Vedoucí bakalářské práce:

Ing. Daniel Honc, Ph.D.

Katedra řízení procesů

Datum zadání bakalářské práce: **23. listopadu 2016**

Termín odevzdání bakalářské práce: **12. května 2017**



Ing. Zdeněk Němec, Ph.D.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 15. prosince 2016

Prohlášení

Prohlašuji:

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

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

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 12. 05. 2017

Jiří Vinduška

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Danielu Honcovi, Ph.D. za vedení práce, ochotu, rady a připomínky ke zpracování. Dále bych rád poděkoval rodině, přítelkyni a přátelům za podporu po celou dobu studia.

V Pardubicích dne 12. 05. 2017

Jiří Vinduška

ANOTACE

Bakalářská práce je zaměřena na řízení laboratorní soustavy pomocí programovatelného automatu. V práci jsou obecně popsány programovatelné automaty a práce s nimi. Práce se konkrétně věnuje PLC Modicon M241, jeho nastavení a programování pomocí software SoMachine. V rámci bakalářské práce byla vytvořena aplikace pro PLC a HMI panel Magelis, ze kterého se nastavují parametry pro řízení soustavy.

KLÍČOVÁ SLOVA

Programovatelný logický automat, Schneider Electric, SoMachine, uživatelské rozhraní, hydraulicko-pneumatická soustava

TITLE

CONTROL OF HYDRAULIC-PNEUMATIC SYSTEM WITH PLC MODICON.

ANNOTATION

The aim of the bachelor's thesis is control of the laboratory system. Programmable logic controllers and how to work with them is described in thesis. The thesis is dedicated specifically to Modicon M241, setup and programming the PLC with software SoMachine. As a part of the thesis an application for human machine interface Magelis from which the parameters for control the system can be set was created.

KEYWORDS

Programmable logic controller, Schneider Electric, SoMachine, human machine interface, hydraulic-pneumatic system

OBSAH

	Seznam zkratk a značek.....	9
	Seznam symbolů proměnných veličin a funkcí.....	10
	Seznam ilustrací.....	11
	Seznam tabulek.....	13
	ÚVOD.....	14
1	TEORETICKÁ ČÁST.....	15
1.1	Programovatelné automaty.....	15
1.1.1	Úvodní seznámení.....	15
1.1.2	Proč se PLC využívá.....	17
1.1.3	Rozdělení PLC.....	17
1.1.4	Programování PLC.....	19
1.1.5	Programovací jazyky.....	19
1.1.6	Vykonávání programu.....	21
1.2	Regulace systému.....	23
1.3	Regulátor.....	24
1.3.1	Spojité regulátor.....	26
1.3.2	Význam složek PID regulátoru.....	27
1.3.3	Metody nastavení PID regulátoru.....	27
1.3.4	Číslicový regulátor.....	30
2	PRAKTICKÁ ČÁST.....	32
2.1	PLC Modicon m241 TM241CEC24T.....	32
2.1.1	Analogový modul TM3.....	34
2.1.2	Ethernetový modul TM4.....	35
2.1.3	Uživatelský panel HMIS5T.....	35
2.2	Laboratorní zařízení.....	37
2.3	Hydraulicko-pneumatická soustava.....	38
2.4	SoMachine.....	39
2.4.1	Vijeo Designer.....	49
2.5	Vlastní řešení úlohy.....	51
2.5.1	Návrh programu.....	52
2.5.2	Návrh displeje.....	55
2.5.3	Identifikace parametrů z přechodové charakteristiky.....	57

2.5.4	Nastavení PID regulátoru pomocí metody T-suma	58
3	ZÁVĚR.....	62
	LITERATURA	64
	PŘÍLOHY	66

SEZNAM ZKRATEK A ZNAČEK

CNC	Computer numerical control
CPU	Central processing unit
HMI	Human machine interface
IEC	International electrotechnical commission
PLC	Programmable logic controller
POU	Program organization unit
RAM	Random access memory
ROM	Read only memory
TFT	Thin-Film Transistor technology
USB	Universal serial bus

SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

d	poruchová veličina
e	regulační odchylka
r_0	zesílení od P-složky
T_d	časová konstanta D-složky
T_i	časová konstanta I-složky
u	akční veličina
w	požadovaná hodnota
y	výstupní veličina

SEZNAM ILUSTRACÍ

Obr. 1.1 – Blokové schéma PLC	16
Obr. 1.2 – Znázornění modulárního PLC	18
Obr. 1.3 – Znázornění kompaktního PLC.....	18
Obr. 1.4 – Cyklus vykonávání programu.....	22
Obr. 1.5 – Otevřené řízení	23
Obr. 1.6 – Zpětnovazební řízení	24
Obr. 1.7 – Zjednodušené schéma zpětnovazebního řízení.....	24
Obr. 1.8 – Statická charakteristika spínacího prvku	26
Obr. 1.9 – Přejímová charakteristika systému.....	28
Obr. 1.10 – Grafická metoda nalezení T-suma	29
Obr. 1.11 – Zjednodušené schéma zapojení číslicového regulátoru v regulačním systému	30
Obr. 2.1 – Rozmístění komponentů na PLC	33
Obr. 2.2 Zapojení vstupů a výstupů na modulu	34
Obr. 2.3 – Rozmístění komponentů na modulu	35
Obr. 2.4 – Displej se zadním modulem	36
Obr. 2.5 – Konstrukce PLC	37
Obr. 2.6 – Konstrukce hydraulicko-pneumatické soustavy	38
Obr. 2.7 – Schéma hydraulicko-pneumatické soustavy.....	39
Obr. 2.8 – Základní obrazovka	40
Obr. 2.9 – Postup vytváření projektu	40
Obr. 2.10 – Pracovní prostor.....	41
Obr. 2.11 – Přidání a) programovatelného automatu b) uživatelského panelu HMI.....	42
Obr. 2.12 – Pracovní okno Logic Builderu.....	43
Obr. 2.13 – Přidání modulu	43
Obr. 2.14 – Pojmenování vstupů a výstupů	44
Obr. 2.15– Přidání POU.....	44
Obr. 2.16 – Průběh vytváření POU.....	44
Obr. 2.17 – Kompilace.....	45
Obr. 2.18 – Přidání POU do MAST.....	46
Obr. 2.19 – Menu Online	46
Obr. 2.20 – POU simulace	47
Obr. 2.21 – Menu Debug	47

Obr. 2.22 – Okno programu po zapsání hodnot.....	47
Obr. 2.23 – Adresy.....	48
Obr. 2.24 – Vybrání aktivního PLC.....	48
Obr. 2.25 – Sdílení proměnných.....	49
Obr. 2.26 – Pracovní plocha Vijeo	49
Obr. 2.27 – Uživatelské rozhraní.....	50
Obr. 2.28 – Nastavení a) spínače b) lampy.....	50
Obr. 2.29 – Průběh simulace a) spuštění b) stisk spínače c) kontrola proměnných	51
Obr. 2.30 – Vytvoření Logfile	53
Obr. 2.31 – LogFile přístupný z FTP.....	53
Obr. 2.32 – LogFile přístupný ze SoMachine.....	53
Obr. 2.33 – Přidání webové vizualizace	54
Obr. 2.34 – Přidání další webové stránky.....	54
Obr. 2.35 – Webové rozhraní a) manuální režim b) automatický režim	54
Obr. 2.36 – Obrazovka manuálního režimu.....	55
Obr. 2.37 – Obrazovka automatického režimu.....	55
Obr. 2.38 – Obrazovka nastavení PID	56
Obr. 2.39 – Obrazovka regulačního pochodu	56
Obr. 2.40 – Odezva regulačního systému na skokovou změnu vstupu a) levé nádrže b) pravé nádrže.....	57
Obr. 2.41 – Simulační model vykreslený spolu s reálným průběhem a) levá nádrž b) pravá nádrž	58
Obr. 2.42 – Vypočítání T-suma a) levá nádrž b) pravá nádrž	59
Obr. 2.43 – Regulační pochod modelů a) levá nádrž b) pravá nádrž	60
Obr. 2.44 – Regulační pochod soustavy a) levá nádrž b) pravá nádrž	61
Obr. 2.45 – Regulační pochod vykreslený s regulačním pochodem modelu a) levá nádrž b) pravá nádrž.....	61

SEZNAM TABULEK

Tab. 1.1 – Nastavení regulátoru.....	28
Tab. 1.2– Nastavení regulátoru.....	30
Tab. 2.1 – Legenda pro PLC TM241CEC24T.....	33
Tab. 2.2 – Legenda pro modul TM4ES4	35
Tab. 2.3 – Legenda k displeji.....	36
Tab. 2.4 – Legenda konstrukce	37
Tab. 2.5 – Legenda k Logic Builderu	43
Tab. 2.6 – Legenda	50
Tab. 2.7 – Vypočítané hodnoty P, PI, PID regulátorů pomocí T-suma.....	60

ÚVOD

V práci se zabývám řízením hydraulicko-pneumatické soustavy pomocí programovatelného automatu. Teoretickou část práce mám rozdělenou na dvě větší kapitoly. První kapitolou jsou programovatelné automaty, které jsou zde detailně popsány. Je zde uvedeno, jaký mají hardware, proč se používají, jaké programovací jazyky se při programování PLC používají atd. Ve druhé kapitole se zabývám regulací, řízením a regulátory obecně. Je zde popsán regulační obvod, základy spojitého řízení a některé metody nastavení PID regulátoru. V praktické části práce jsem se konkrétně věnoval programovatelnému automatu od firmy Schneider Electric, jeho modulům a řízení hydraulicko-pneumatické soustavy. Prakticky jsem vyzkoušel metodu T-suma pro nastavení PID regulátoru.

Cílem bylo vytvořit aplikaci pro PLC, uživatelské prostředí pro grafický panel HMI Magelis, pomocí které bude možné ovládat a řídit hydraulicko-pneumatickou soustavu. Uživatel bude moci v manuálním režimu ovládat výkon čerpadel, nebo se přepne do automatického režimu, kde nastaví parametry PID regulátoru a zadá žádanou hodnotu a programovatelný automat bude automaticky soustavu regulovat.

Programovatelný automat je od firmy Schneider Electric. Model PLC Modicon M241 TM241CEC24T s analogovým modulem TM3AM6, dále jsem využil grafický panel Magelis HMI S5T. Úlohu jsem realizoval na laboratorním modelu hydraulicko-pneumatické soustavy.

1 TEORETICKÁ ČÁST

1.1 Programovatelné automaty

PLC se využívají skoro ve všech odvětví průmyslu. Řídí funkci výrobních linek v továrnách, CNC strojů a soustruhů. Příchodem automatizace do rodinných domů, našli své další uplatnění v inteligentních domech, kde řídí celý dům, uživatel může tedy pomocí operátorského panelu, nebo z tabletu řídit celý svůj dům, od světel, přes klimatizaci až po garážová vrata (Šmejkal, 1999).

1.1.1 Úvodní seznámení

Programovatelný automat, častěji označovaný jako PLC, je hardwarové zařízení, které je přizpůsobené pro řízení průmyslových a technologických procesů. Toto řízení probíhá nespojitě, většinou pomocí nějakého programu, který je nahrán ve vnitřní paměti automatu (Chlebný, 2009).

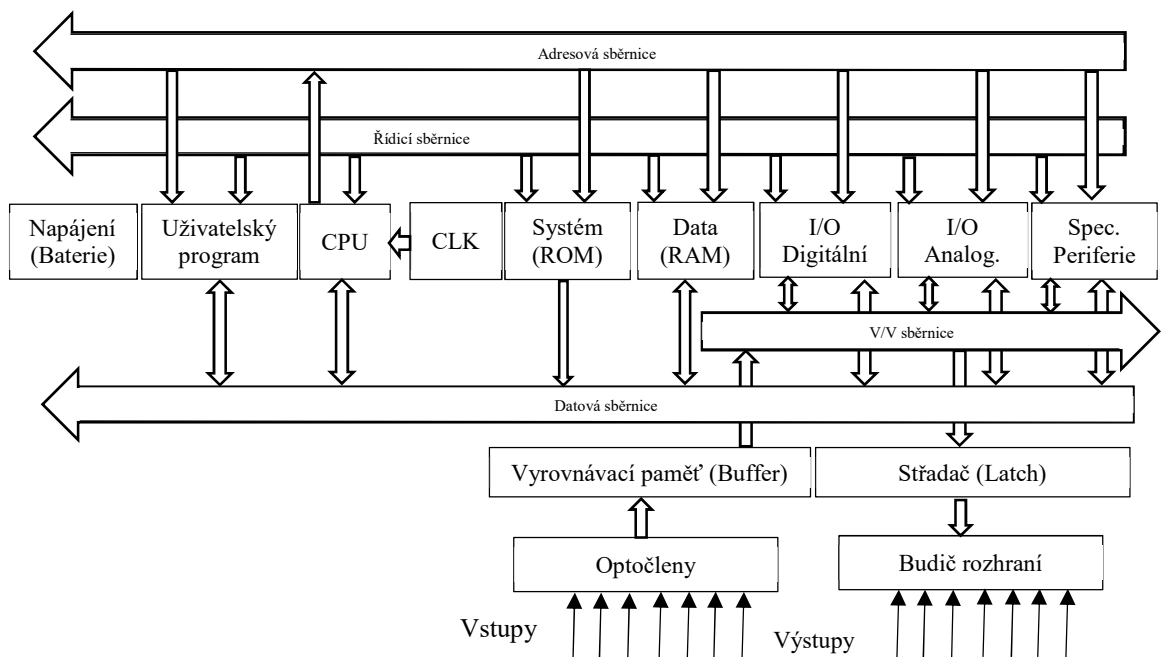
Přesná definice: PLC je číslicově pracující elektronický systém, který byl navržen pro použití v průmyslovém prostředí, využívající programovatelnou paměť pro interní ukládání uživatelsky orientovaných instrukcí pro provádění specifických funkcí, jako jsou logických, sekvenčních, časovacích, čítacích nebo aritmetických, za účelem řízení strojů či procesů, a to prostřednictvím digitálních nebo analogových vstupů a výstupů. Jak programové řídicí jednotka, tak její periferní zařízení jsou konturovány pro snadné začlenění do systémů průmyslového řízení (ČSN EN 61131-1, 2004).

Z uvedené definice plyne, že postup pro řízení systému pomocí PLC se změní, když uživatel změní program v jeho paměti. Na vstup PLC je přiveden spojitý nebo logický signál, který znázorňuje určitý stav, dále jsou vstupy galvanicky odděleny od zbytku hardware PLC. Výstupem jsou signály, které jsou určeny k ovládní akčních členů (Chlebný, 2009).

Hlavní částí PLC je CPU (procesor), neboli hlavní výpočetní jednotka PLC. Dále je součástí programovací rozhraní, rozhraní pro sériovou komunikaci (RS485) a normalizované rozhraní pro komunikaci typu Profibus, CAN, Ethernet atd.

PLC obsahuje čtyři hlavní sběrnice řídicí, adresovou, datovou a vstupně/výstupní sběrnici. Řídicí sběrnici procesor ovládá ostatní periférie PLC. Řídí příjem, vysílání, časové signály a synchronizaci bloků. Adresová sběrnice přenáší aktuální adresu pro data. Datová sběrnice přenáší signály ze vstupů do procesoru, poté se zpracují podle programu a vyšlou se zpět na výstupy. Vstupně výstupní sběrnice obstarává komunikaci s okolním světem.

Dále jsou zde tři bloky s pamětmi. Blok Systém (ROM) je uložen firmware od výrobce PLC, uživatel může firmware aktualizovat, ale jinak k tomuto bloku nemá přístup. Paměť data (RAM), do které uživatel nahrává program, je rozdělena na data a program. Třetí blok s pamětí není na blokovém schématu, ale většinou se na PLC nachází vstupní port pro SD kartu. Pokud je port správně nastaven PLC z něj může načítat program, nebo na něj ukládat data. Vše v PLC je řízeno operačním systémem reálného času. Tento popis je znázorněn na obr. 1.1.



Obr. 1.1 – Blokové schéma PLC

K nejznámějším výrobcům PLC patří Schneider Electric, Siemens, Allen Bradley a český zástupce firma Teco.

1.1.2 Proč se PLC využívá

Rychlá realizace

Pokud chceme s PLC něco řídit, stačí nakoupit správné moduly PLC pro danou aplikaci v systému. Poté musíme napsat program, a zpřístupnit ho přes uživatelské rozhraní operátorům systému a máme hotovo. Nemusíme nic vyvíjet a sestavovat např. z relé nebo logických obvodů, protože PLC nám tyto funkce nahradí (Šmejkal, 1999).

Spolehlivost, odolnost, diagnostika

Programovatelný automat je navržen tak aby byl velmi spolehlivý i v průmyslových podmínkách, je odolný proti rušení i poruchám, vyznačuje se dlouhou střední dobou mezi poruchami (Mean Time Between Failures). PLC jsou vybaveny vnitřními diagnostickými funkcemi, které kontrolují činnosti vnitřních periférií. Při závadě ji dokáží včas najít a ošetřit, nebo uloží zprávu pro programátora, aby věděl, co opravit.

Nekončící změny v zadání

PLC bylo vyvinuto tak aby se dalo přizpůsobit požadavkům jak programátora, tak i zadavateli projektů. Program nahraný v PLC lze rozšířit, vyměnit nebo aktualizovat podle aktuální potřeby v systému. Ladění chyb je mnohem jednodušší než v případě systému s pevnou logikou. Pokud v průběhu času budeme chtít systém rozšířit, stačí nakoupit rozšiřující moduly pro PLC. Rozšíření v případě pevné logiky je prakticky nemožné, nebo těžko realizovatelné.

Samozřejmě ani programování PLC není bezproblémové. Nákladnost a pracnost změn v programu je ovšem výrazně nižší než v systémech s relé logikou.

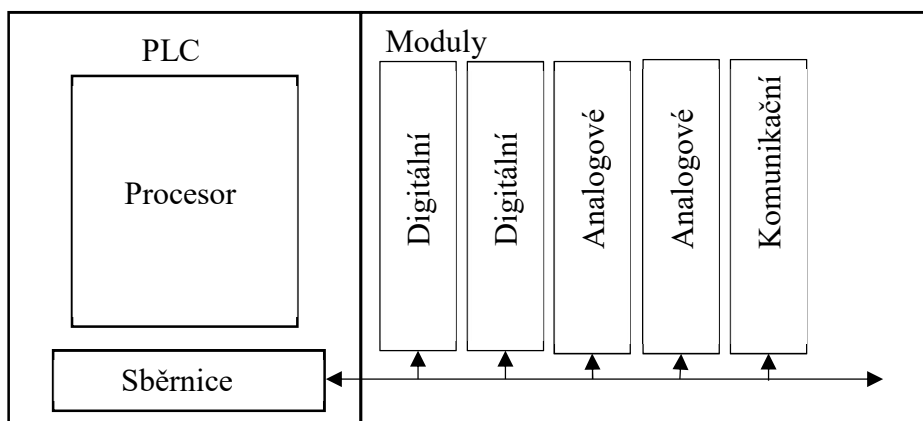
1.1.3 Rozdělení PLC

Programovatelné automaty lze, z hlediska konstrukčního provedení a uživatelské koncepce, rozdělit do dvou kategorií – modulární a kompaktní.

Modulární PLC

Modulární PLC (obr. 1.2) obsahuje dva hlavní prvky, modul procesoru a modul napájení. Ostatní prvky jako jsou moduly s digitálními, nebo analogovými vstupy/výstupy, modul pro připojení k sběrnici atp. se nasouvají na nosný rám (DIN lištu) a připojí se k modulu procesoru. Moduly jsou vybaveny vnitřní paralelní sběrnici pro signálové a elektrické připojení

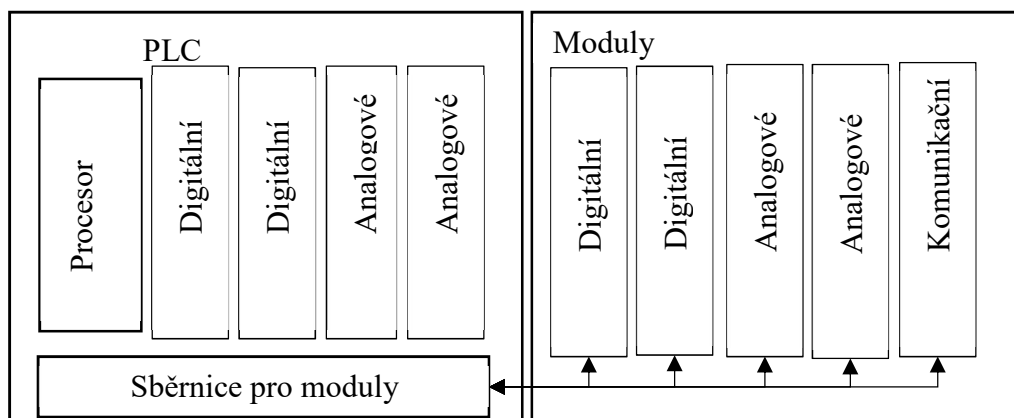
k procesoru. Mechanické upevnění modulů je v provedení plug and play, klade tím minimální nároky na obsluhu. Umístění modulů v rámu vzhledem k procesoru určuje adresu tohoto modulu. Pokud budeme potřebovat soustavu modulů rozšířit, můžeme kdykoliv připojit další modul, nebo celou lištu s moduly, to nám dává výhodu v rozšiřitelnosti oproti kompaktnímu PLC (Chlebný, 2009).



Obr. 1.2 – Znáznornění modulárního PLC

Kompaktní PLC

Kompaktní PLC (obr. 1.3) se většinou používají v menších systémech, kde je pevně daná konfigurace vstupů a výstupů (např. pračka, klimatizace budov, jednoúčelové stroje atd.). Tyto PLC mají tedy pevnou konfiguraci, které může obsahovat například 12 digitálních vstupů/výstupů, 4 analogové vstupy výstupy, čítače/časovače, připojení ke sběrnici a připojení k PC. Moderní kompaktní PLC lze rozšířit o další moduly stejně jako tomu je u modulárního PLC.



Obr. 1.3 – Znáznornění kompaktního PLC

Striktní rozdělení PLC na čistě modulární a čistě kompaktní v moderní době odpadá. Oba dva druhy PLC se programují totožně.

1.1.4 Programování PLC

Nejdůležitější součástí každého PCL je centrální jednotka neboli procesor (CPU), která poskytuje programovatelnému automatu inteligenci, řídí celou činnost PLC. Centrální jednotka provádí uživatelský program pomocí souboru instrukcí. Zajišťuje komunikaci s vestavěnými moduly ale i se vzdálenými členy systému (Šmejkal, 1999).

Paměť PLC je rozdělena na dvě části, podle Harvardské architektury, program a data. Paměť s programem se během vykonávání programu většinou nemění. Druhá část obsahuje data, se kterými program pracuje, tj. uživatelské registry, čítače časovače, obrazy vstupů a výstupů, komunikační a jiné proměnné. Tato paměť se v některých literaturách nazývá operační, protože jsou nad ni prováděny operace, dynamicky se mění během programu.

Původní programovatelné automaty byly určeny k náhradě logických (reléových) systému, proto se kladl požadavek, aby programovací jazyky pro PLC vycházeli z této logiky. Proto v žádném PLC nechybí instrukce pro základní logické operace, paměťové funkce, čítače a časovače, posuvné registry, aritmetické operace atd. Moderní PLC obsahuje instrukce pro popis spojitých i nespojitých regulátorů. Protože bylo mnoho výrobců různých PLC musela se stanovit norma pro programování PLC IEC 1131, která definuje podmínky a požadavky na programové vybavení PLC. Mezi hlavní požadavky patří (Chlebný, 2009):

- Nezávislost programu na technickém vybavení PLC. Jednou vyvinutý a odladěný program lze použít na jiný typ PLC a bude fungovat stejně.
- Jednotný programátorský přístup. Všechny PLC by se měli dát programovat stejným způsobem, stejnými jazyky.
- Možnost ladění programu již ve fázi návrhu. Možnost simulace programu, ještě před nasazením, aby se předešlo poruchám.
- Strukturovanost a modularita. Programovací jazyk by měl být přehledný a v jakékoliv fázi vývoje by měla být možnost úpravy.

1.1.5 Programovací jazyky

PLC lze programovat až šesti základními jazyky, které jsou popsány v normě IEC 61131-3. Základní rozdělení je na textové jazyky, mezi ně patří instruction list (IL) a structured text (ST). Další skupinou jsou grafické jazyky, mezi ně patří ladder diagram (LD), function

block diagram (FBD), continuous function chart (CFC) a sequential function chart (SFC). Tyto jazyky lze mezi sebou libovolně kombinovat, můžeme část programu napsat v ST, poté obsluhovat vstupy výstupy pomocí FBD a jednoduché funkce ve kterých požadujeme rychlost napsat v IL. Všechny PLC nemusí podporovat všechny uvedené jazyky, ale je to pravidlem (Vojáček, 2011).

IL – instruction list (list instrukcí, posloupnost instrukcí)

IL je nejzákladnější možný popis programu, velmi podobný jazyku assembler. Program se skládá ze základních instrukcí, na jednom řádku je pouze jedna instrukce. Tento způsob programování PLC se vyznačuje dokonalým přehledem programátora nad ovládaným hardwarem. Pomocí tohoto způsobu lze optimalizovat velikost ale i rychlost programu. Jeho nevýhodou je nepřehlednost při psaní složitých aplikací a nutnost detailní znalosti hardware. Výhodou ovšem je, že přesně nadefinujeme činnost programu, program bude velmi rychlý a bude velmi úsporný na paměť.

ST – structured text (strukturovaný text)

ST je vyšší programovací jazyk podobný Pascalu nebo jazyku C. Jeden příkaz vykoná posloupnost instrukcí. Syntaxí se velmi podobá Pascalu. Programování pomocí strukturovaného textu je tedy vhodné pro práci se složitějšími algoritmy, funkcemi a daty. Je vhodný pro práci s řetězci textu. Po ovládnutí syntaxe ST lze v něm napsat všechny možné aplikace PLC. Bohužel některé konstrukce jsou v ST nepřehledné, a proto je výhodnější je vykonat v jiném jazyce. Tento jazyk je vhodný zejména pro programátory, kteří už mají nějakou zkušenost s programováním z mikroprocesorů.

LD – ladder diagram (reléové schéma)

První grafický programovací jazyk, který vychází z reléové logiky, na vše používá spínací relé a rozpínací relé. Je vhodný pro zpracování velkého množství logických vstupů a výstupů. Jazyk obsahuje i složitější funkce jako jsou časovače a čítače. Program se „píše“ v liniích (řádků) každá linie vykonává určitou činnost. Výhodou je přehlednost zápisu a rychlost zpracování. Nevýhodou je ovšem velmi složitá práce s textem, při narůstající délce programu klesá přehlednost LD.

FBD – function block diagram (schéma funkčních bloků)

Program v programovacím jazyce FBD se zapisuje pomocí soustavy propojených bloků. Tyto bloky mohou realizovat různé funkce (logická hradla, časovače/čítače, uživatelské funkce). I zde je program uspořádán do řádků a tyto řádků se vykonávají za sebou. Výhodou je, jak tomu bylo u LD, přehlednost programu, protože je rozčleněn na řádky. Také můžeme složitou funkci zamaskovat do jednoho bloku. Stejně jako LD je nevhodný pro práci s textem a program se stává nepřehledný při složitějších algoritmech.

CFC – continuous function chart (volně propojovatelné bloky)

Jazyk CFC je velmi podobný FBD, také se zde propojují bloky s různými funkcemi, ale zde není žádné omezení řádků. Při vytvoření programu se vytvoří volná plocha, na kterou vkládáme bloky a ty poté volně propojujeme mezi sebou. Tento jazyk má všechny výhody FBD je vhodný pro zpracování logických i analogových vstupně výstupních operací. Při simulaci můžeme sledovat, kudy prochází signál. Má ovšem i společné nevýhody s FBD, při větší složitosti programu se stává pro programátora nepřehledný a je třeba udržovat dobré rozvržení bloků na ploše. Tento jazyk není součástí normy IEC 61131-3, ale většina výrobců PLC ho používá.

SFC – sequential function chart (vývojový diagram)

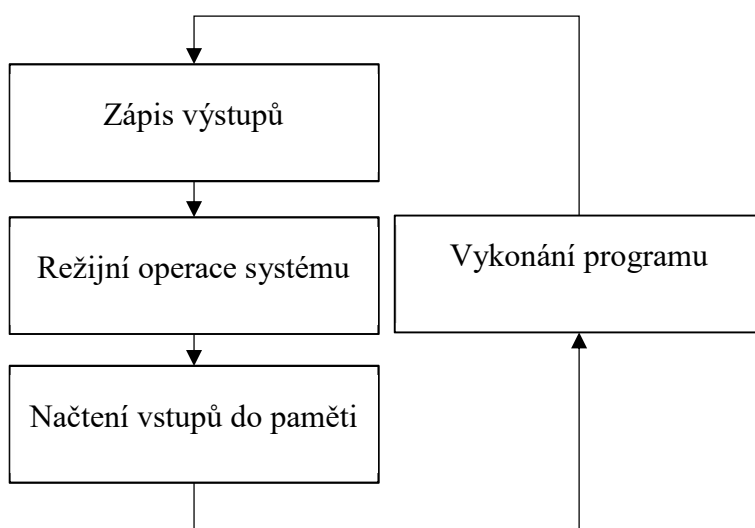
Jak už český překlad napovídá, tento programovací jazyk je velmi podobný vývojovému digramu. Umožňuje definovat chování programu pomocí vývojového schéma, větvení na základě podmínek. Tento režim je ideální jako hlavní program, ze kterého se na základě rozhodnutí z podmínky spustí podprogram napsaný v jiném jazyce. Hlavní výhoda tohoto zápisu je velmi dobrá přehlednost i při rozsáhlém projektu, můžeme ošetřit všechny stavy v programu. Bohužel tento zápis je nevhodný pro vstupně výstupní zpracovávání signálů.

1.1.6 Vykonávání programu

Programátor vytváří programovou organizační jednotku – POU, která obsahuje zápis programu pomocí libovolného programovacího jazyka PLC. Po zkompileování POU se vytvoří program, posloupnost instrukcí srozumitelných pro PLC. Program se v programovém automatu většinou vykonává cyklicky. Na rozdíl od programovacích jazyků pro PC, zde se programátor nemusí starat o to, aby se na konci programu vykonávání vrátilo na začátek, vše řeší systém uvnitř PLC. Programátor nastavuje jen pořadí vykonávání POU, může se jich vykonávat víc po

sobě, a taky nastaví hlídací časovač, po přetečení tohoto časovače se restartuje PLC, neboť nastala chyba v programu, protože se časovač nestihl obnovit (Šmejkal, 1999).

V jednom cyklu programu se vykonávají čtyři úkony, zobrazené na obr. 1.4. První část úkonů je aktualizace vstupů a výstupů. Nejprve se z paměti načtou obrazy výstupů, které se následně zapíší na určené periferní zařízení. Následně proběhnou režijní operace, tj. aktualizace systémových a časových proměnných (zde se obnoví hlídací časovač), ošetření komunikace mezi moduly, naplánování dalšího cyklu programu atd. V závěru této činnosti proběhne sejmutí aktuálních hodnot vstupů a uložení jejich obrazů do paměti, aby mohl proběhnout uživatelský program.



Obr. 1.4 – Cyklus vykonávání programu

Program PLC nepracuje s aktuálními hodnotami vstupů ale pouze s jejich obrazy, to jsou jejich poslední záznamy v paměti. To je nezbytné pro správné vykonávání programu. Pokud by program pracoval s aktuálními hodnotami, mohlo by docházet k tzv. hazardním stavům. Hazardní stav je takový, kdy se na výstupu objeví hodnota, která tam nesmí být, například mezi výsledek z výpočtu.

Při vykonávání programu je PLC v režimu RUN, v tomto režimu probíhá načítání vstupů, v procesoru PLC se vykonávají instrukce podle programu a zapisují se výsledky na výstupy. Dalším režimem PLC je HALT, v tomto režimu není vykonávána žádná činnost ani komunikace s okolím, do tohoto stavu se PLC může dostat při chybě programu, programátor chybu ošetří a přepne PLC do režimu RUN. (Teco, 2014)

PLC mají také dva druhy restartu, teplý restart, při kterém se ponechávají aktuální hodnoty vstupů v paměti a studený restart při kterém se všechny hodnoty vynulují.

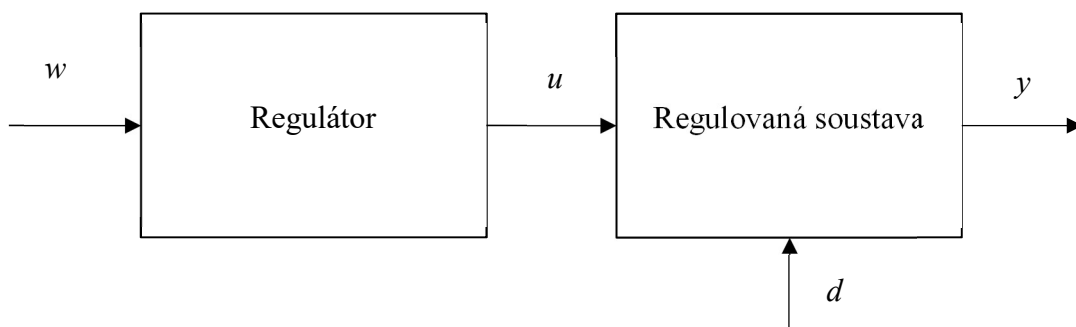
1.2 Regulace systému

Regulace systému, je činnost udržování požadovaných hodnot v systému. Toto udržování konstantních hodnot nám zajistí regulátor, který svou akční veličinou reguluje systém. Výsledkem působení regulátoru na soustavu má být požadovaný průběh regulačního pochodu. Regulace může být dvojího typu:

- Dopředná regulace,
- Zpětnovazební regulace.

Dopředná regulace

Dopředná regulace, nebo taky otevřené řízení, můžeme použít tam, kde jsme plně seznámeni s ovládanou soustavou. Vnější vlivy jsou konstantní, nebo se mění podle předem známého průběhu. I bez akčního zásahu od regulátoru jsme si jistí s výsledkem reakce od soustavy, tu zjistíme výpočtem anebo ze zkušeností víme, jak se soustava chová při určitých akčních zásazích (Cvejn, 2015).

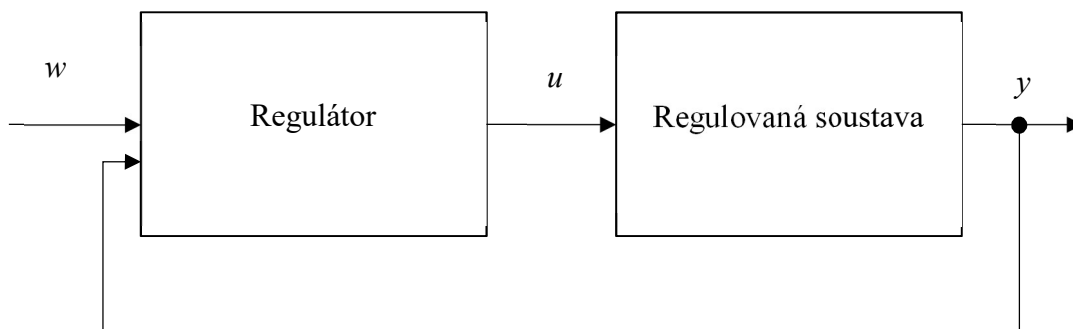


Obr. 1.5 – Otevřené řízení

Na obr. 1.5 vidíme schéma otevřeného řízení. Regulátor neví, jak soustava na jeho zásah zareagovala.

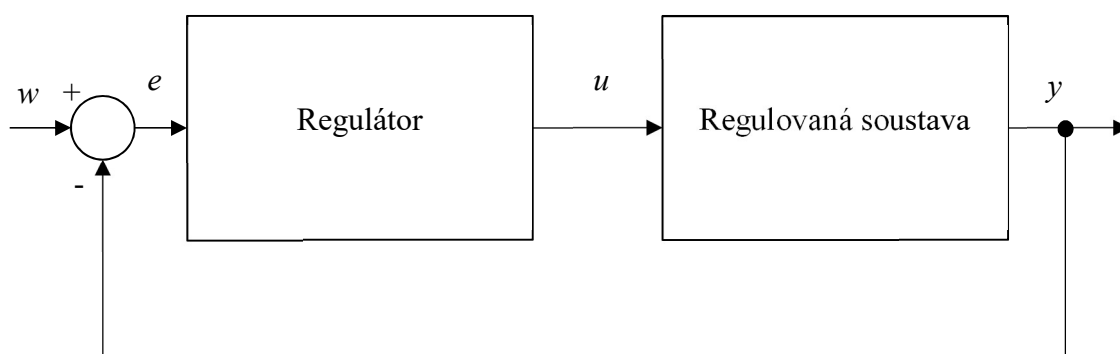
Zpětnovazební regulace

Při zpětnovazebním řízení je výstupní veličina soustavy přivedena zpět jako další vstup regulátoru. Regulátor pro své další rozhodování využívá i aktuální informaci o průběhu procesu v soustavě. Touto metodou lze dosáhnout přesné regulace i bez předchozí znalosti soustavy, nebo průběhu poruchové veličiny.



Obr. 1.6 – Zpětnovazební řízení

Na obr. 1.6 je zobrazeno zpětnovazební řízení, regulátor má druhý vstup, na který se přivádí výstup ze soustavy. V praxi se ovšem více používá následující schéma uvedené na obr. 1.7.



Obr. 1.7 – Zjednodušené schéma zpětnovazebního řízení

Na zjednodušeném schématu zpětnovazebního řízení má regulátor pouze jeden vstup a na ten se přivádí rozdíl požadované hodnoty a výstupní veličiny e .

1.3 Regulátor

Regulátor je tedy zařízení, které nám zajišťuje regulaci systému. Regulátor může řídit systém, buď se stejnou veličinou, s jakou pracuje soustava, takový regulátor se nazývá bez převodu veličiny. Tento typ regulátorů se v praxi využívá výjimečně. Častější typ regulátorů je s převodem veličin. Nejčastěji se dnes můžeme setkat s elektronickými regulátory, které převádí výstupní veličinu systému na elektrický signál. V praxi se ale využívají i jiné typy regulátorů, především hydraulické, které pracují s nějakou kapalinou a pneumatické, které využívají vzduch.

Dále regulátory můžeme dělit na **analogové** a **číslicové**.

Analogový regulátor pracuje se spojitým, analogovým, signálem. Požaduje převod veličiny na straně výstupu ze soustavy i na straně vstupu do soustavy. Na straně výstupu ze soustavy se nachází snímač, který čte aktuální hodnotu výstupu a převádí ji na elektrický signál. Na straně vstupu do soustavy se nachází akční člen, který zase naopak převádí elektrický signál na akční veličinu, se kterou systém pracuje. Při přenosu elektrického signálu ze snímače do regulátoru musíme zajistit, aby nedocházelo k rušení signálu, zkreslení signálu znehodnocuje výslednou regulaci.

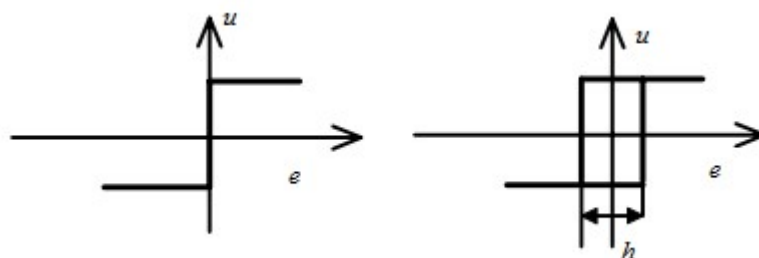
Na rozdíl od analogového regulátoru, **číslicový regulátor**, nebo také diskrétní regulátor, pracuje s nespojitým signálem, tzv. vzorky. Součástí číslicového regulátoru musí být vzorkovač, ten je realizován pomocí analogově/digitálního převodníku, kdy periodicky snímá hodnotu výstupu soustavy a převádí ji na digitální slovo. Tato perioda mezi dvěma vzorky se nazývá perioda vzorkování. Převod z analogového signálu na diskrétní probíhá s omezenou přesností. Digitální signál může nabývat pouze určitý počet hodnot na rozdíl od analogového, který může mít nekonečně mnoho hodnot, při převodu z analogového na digitální signál tedy musíme „zaokrouhlovat“ toto se nazývá kvantizační chyba. Akční zásah je digitálně/analogovým převodníkem převeden zpět na analogový signál a přiveden na vstup soustavy. Digitální regulátory jsou výhodné svojí programovatelností (např. můžeme si vytvořit program regulátoru pro PLC nebo mikropočítač, který bude zpracovávat digitální data a bude nad nimi provádět různé operace, které by byli složitě proveditelné v analogové podobě), bohužel pro velmi rychlé procesy jsou nepoužitelné, protože vzorkování a převod na digitální podobu trvá nějakou dobu.

Regulátory můžeme rozdělit taky podle toho, jak zasahují do systému, a to buď **nespojité** nebo **spojité**.

Nespojitý regulátor

Nespojitý regulátor, může být dvupolohový nebo tří polohový, používá se v jednoduchých systémech, např. ohřev teploty. Prakticky se pro dvupolohový regulátor využívá relé.

Dvupolohový regulátor spíná na základě regulační odchylky, jelikož by docházelo k opotřebení relé při častém spínání, zavádí se hystereze. Na obr. 1.8 (Cvejn, 2015) je zobrazena statická charakteristika relé. Vlevo na obrázku je průběh bez hystereze, vpravo s hysterezí.



Obr. 1.8 – Statická charakteristika spínacího prvku

1.3.1 Spojitý regulátor

Spojité regulátor, jak už z názvů vyplývá, pracuje spojitě. Nejčastější typ spojitého regulátoru je PID regulátor. Akční zásah regulátoru vyplývá z regulační odchylky, kdy používá přímou úměrnost, integraci nebo derivaci. PID regulátor je jednoduchý konstrukcí, dá se použít na většinu systémů a dá se snadno realizovat, např. pomocí operačních zesilovačů.

$$u(t) = r_0 e(t) + \frac{r_0}{T_i} \int_{\tau=0}^t e(\tau) d\tau + r_0 T_d \frac{de(t)}{dt}, \quad (1.1)$$

kde r_0 – zesílení od P-složky,

T_i – časová konstanta I-složky,

T_d – časová konstanta D-složky.

Rovnice (1.1) odpovídá následující přenos

$$R = \frac{U}{E} = r_0 + \frac{r_{-1}}{s} + r_1 s \quad (1.2)$$

Z uvedené rovnice vyplývají následující použitelné varianty PID regulátoru:

P-regulátor ... $r_{-1} = r_1 = 0$

PI-regulátor ... $r_1 = 0$

PD-regulátor ... $r_{-1} = 0$

I-regulátor ... $r_0 = r_{-1} = 0$

1.3.2 Význam složek PID regulátoru

P-složka

P-složka regulátoru je zesílení regulační odchylky, tedy zesílení zpětné vazby. Čím větší hodnota P složky, tím je rychlejší regulační pochod. Při vysokých hodnotách ale dochází k rozkmitání soustavy. Kmitání může být netlumené, to může být žádané, při nastavování PID regulátoru pomocí metody Zieglera-Nicholse. Ostatní druhy kmitání jsou nežádoucí, protože se soustava stává nestabilní nebo se prodlužuje doba ustálení.

Samotná P-složka nezaručí nulovou regulační odchylku, a proto musíme použít P regulátor spolu s dalším typem regulátoru.

I-složka

Přidáním I-složky dosáhneme nulové regulační odchylky. Přidáme integraci regulační odchylky. I-regulátor přidává do soustavy nulový pól, zvýší se řád soustavy. U pomalého systému nastavujeme malou integrační konstantu a naopak. Získáme přesnost za cenu prodlouženého regulačního děje. Urychlení regulačního pochodu docílíme přidáním poslední složky.

D-složka

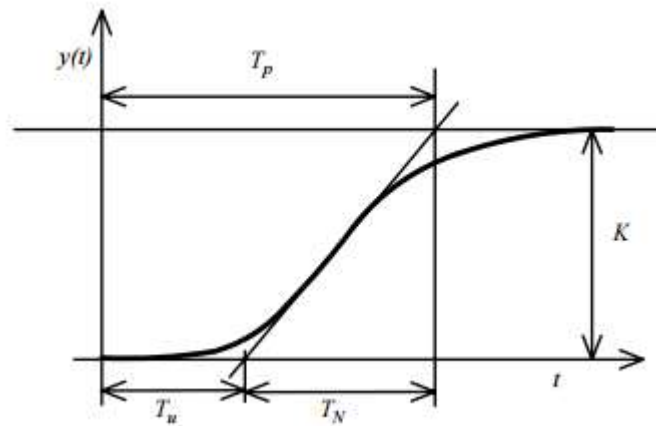
D-složka urychlí regulační pochod soustavy. Přidáme derivaci regulační odchylky. D-regulátor nelze prakticky sestavit, protože jeho amplituda by musela být nekonečná v nekonečně malém čase. Při praktické realizaci dochází ke zkreslení. Tento typ regulátoru reaguje pouze na změnu regulačního pochodu, při ustálené hodnotě nereaguje. Jeho největší nevýhodou je, že zesiluje šum.

1.3.3 Metody nastavení PID regulátoru

V této kapitole popíšu dvě metody optimálního nastavení PID regulátoru.

Nastavení PID regulátoru pomocí metody určení parametrů z přechodové charakteristiky od autorů Zieglera a Nicholse

Tuto metodu můžeme použít, pokud známe přechodovou charakteristiku systému. Průběh soustavy je zobrazen na obr. 1.9 (Cvejn, 2015).



Obr. 1.9 – Přejchodová charakteristika systému

Poté můžeme určit dobu průtahu T_u a dobu náběhu T_N . Definujeme normalizované dopravní zpoždění jako poměr těchto dvou hodnot, $\Theta = \frac{T_u}{T_N}$. Pro nastavení příslušných regulátorů využijeme tab. 1.1.

Tab. 1.1 – Nastavení regulátoru

Regulátor	r_0	T_i	T_d
P	$1/(K\Theta)$	-	-
PI	$0,9/(K\Theta)$	$3T_u$	-
PID	$1,2/(K\Theta)$	$2T_u$	$0,5T_u$

Tato metoda je vhodná pouze pro nekmitavé systémy a hodnoty normalizovaného zpoždění v rozsahu 0,1 až 0,5. Metoda nastavení regulátoru z přechodové charakteristiky vychází z další metody Ziegler-Nicholse a to z metody kritického zesílení, obě metody patří do dnes k nejpoužívanějším.

Metoda T-suma

Další metoda, ve které musíme znát přechodovou charakteristiku soustavy, je Kuhnova metoda T-suma. Metoda nastavení je velmi spolehlivá a má dobré výsledky v případě, že přechodová charakteristika má esovitý průběh. (Kuhn, 1995)

Pokud přenosová funkce systému je ve tvaru (3).

$$G(s) = K_s \frac{(1 + T_{D1}s)(1 + T_{D2}s)\dots(1 + T_{Dm}s)}{(1 + T_1s)(1 + T_2s)\dots(1 + T_ns)} e^{-sT_t}, \quad (1.3)$$

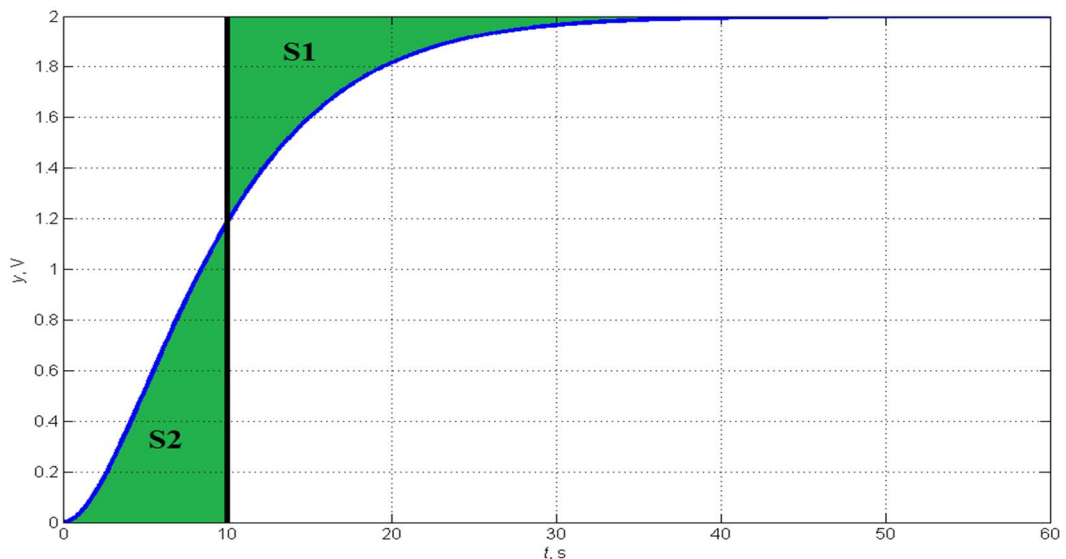
kde K_s – zesílení systému,

$T_{D1}, T_{D2}, T_{Dm}, T_1, T_2, T_n$ – časové konstanty.

Pak lze odvodit T_Σ

$$T_\Sigma = T_1 + T_2 + \dots + T_n - T_{D1} - T_{D2} - \dots - T_{Dm} + T_t. \quad (1.4)$$

Protože toto určení konstanty T_Σ je možné, pouze pokud známe všechny časové konstanty, je možné použít praktičtější metodu hledání konstanty T_Σ . Pokud máme přechodovou charakteristiku, pak v ní posouváme svislou čáru zleva doprava tak dlouho, dokud nenajdeme místo, ve kterém se oblasti S1, S2 rovnají, viz obr. 1.10. Tento postup lze i matematicky odvodit.



Obr. 1.10 – Grafická metoda nalezení T-suma

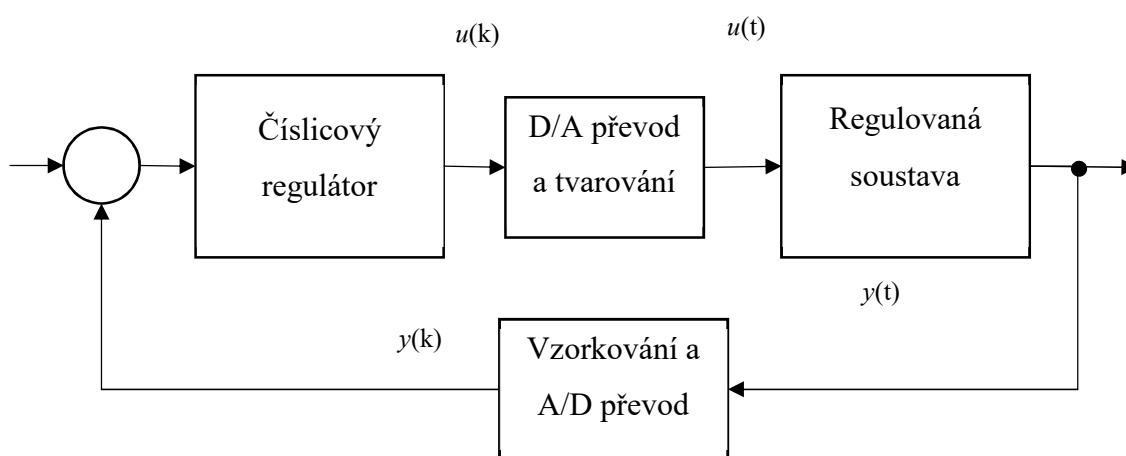
Po nalezení konstanty T_Σ , a známého zesílení systému Z , můžeme určit parametry PID regulátoru z tab. 1.2.

Tab. 1.2– Nastavení regulátoru

Regulátor	r_0	T_i	T_d
P	$1/Z$	-	-
PI normální	$0,5/Z$	$0,5 T_\Sigma$	-
PI rychlý	$1/Z$	$0,7 T_\Sigma$	-
PID normální	$1/Z$	$0,66 T_\Sigma$	$0,17 T_\Sigma$
PID rychlý	$2/Z$	$0,8 T_\Sigma$	$0,19 T_\Sigma$

1.3.4 Číslicový regulátor

Při řízení soustavy pomocí mikročipu nebo počítače. Musíme použít tvar regulátoru, který pracuje s po sobě jdoucími vzorky. Na vstupu do regulátoru je vzorkovač, který v pravidelných intervalech snímá výstupní veličinu, tato hodnota se převede v A/D převodníku na digitální podobu (posloupnost bitů). V mikropočítači se zpracuje a vypočítá se příslušný akční zásah, který se pomocí D/A převodníku převede zpátky na analogovou hodnotu. Analogová hodnota projde přes tvarovač na vstup soustavy. Tvarování signálu znamená, že ze vzorků se vytvoří zpátky spojitý signál. Zjednodušené schéma číslicového regulačního systému na obr. 1.11(Cvejn, 2015).



Obr. 1.11 – Zjednodušené schéma zapojení číslicového regulátoru v regulačním systému

Číslicový regulátor má rovnici, která pracuje se vzorky, s hodnotou aktuální a minulou. V rovnici spojitého PID regulátoru, (5), nahradíme integraci sumací vzorků, derivaci diferencí po sobě jdoucích vzorků a čas t za diskretní krok, dostaneme následující rovnici.

$$u(k) = r_0 \left(e(k) + \frac{T}{T_i} \sum_{i=1}^k e(i) + T_d \frac{e(k) - e(k-1)}{T} \right), \quad (1.5)$$

kde T – perioda vzorkování.

U tohoto tvaru není jednoduchá aplikace anti-windup při omezení akční veličiny. Tuto nevýhodu můžeme odstranit úpravou na přírůstkový tvar PSD (proporcionálně sumačně diferenčního) regulátoru.

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (1.6)$$

Následuje vyjádření jednotlivých konstant.

$$q_0 = r_0 \left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right), \quad (1.7)$$

$$q_1 = -r_0 \left(1 + \frac{2T_d}{T}\right), \quad (1.8)$$

$$q_2 = r_0 \frac{T_d}{T}. \quad (1.9)$$

Regulátor pak pracuje s přírůstkem akční veličiny.

2 PRAKTICKÁ ČÁST

V práci jsem použil zařízení od firmy Schneider Electric. Konkrétně se jedná o PLC Modicon M241 TM241CEC24T, analogový modul TM3AM6, ethernetový modul TM4ES4 a uživatelský panel Magelis HMI S5T. Vše je upevněno v rámu a PLC s moduly zasunuto na DIN lištu. Projekt jsem vytvořil v prostředí SoMachine 4.1.

2.1 PLC Modicon M241 TM241CEC24T

Programovatelný automat M241 lze naprogramovat pomocí software SoMachine, který podporuje všechny programovací jazyky, které se nachází v normě IEC 61131-3. PLC lze tedy naprogramovat pomocí IL, ST, FBD, SFC, LD dále podporuje jazyk CFC, který není součástí normy. (Schneider Electric, 2014)

PLC lze napájet 24 V stejnosměrného napětí. Napětí je v mém případě dodáváno ze zdroje umístěném v rámu.

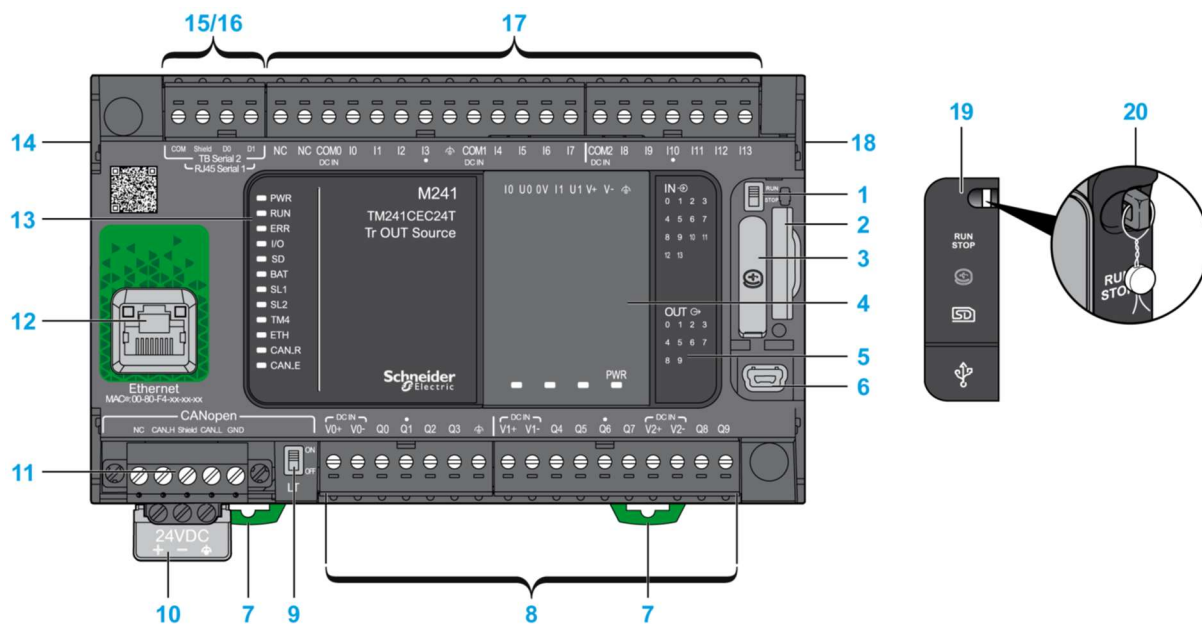
Dále obsahuje systém reálného času, který je napájen z baterie, aby nebyl závislí na napětí ze zdroje. Na PLC se také nachází hardwarový přepínač Run/Stop, kterým lze zastavit veškerou činnost PLC. Jedná se o systém ochrany, při naléhavé potřebě zastavit proces.

Programovatelný automat obsahuje dva typy pamětí. První typ paměti je RAM o velikosti 64 MB, z této velikosti je 8 MB vyhrazeno pro vykonávání aplikace. Druhý typ paměti je Flash o velikosti 128 MB, která se používá pro ukládání uživatelských dat a programu. V případě výpadku napájení jsou data v paměti Flash zachována. Na PLC se také nachází slot na SD kartu.

PLC TM241CEC24T obsahuje:

- 6 normálních digitálních vstupů/výstupů,
- 8 rychlých digitálních vstupů (čítačů), 4 rychlé digitální výstupy,
- 2 porty pro sériovou komunikaci, USB port a Ethernetový port.

Normální vstupy mohou pracovat s maximální frekvencí 5 kHz, rychlé vstupy se používají pro rychlé čítání nad 5 kHz. PLC je zobrazeno na obr. 2.1.



Obr. 2.1 – Rozmístění komponentů na PLC

Tab. 2.1 – Legenda pro PLC TM241CEC24T

číslo	Popis
1	Run/Stop přepínač
2	Slot na SD kartu
3	Slot na baterku k RTC
4	Slot na cartridge
5	Indikační LED diody
6	USB mini-B programovací port/port pro připojení k SoMachine
7	Úchyt na DIN lištu
8	Vestavěné výstupy
9	Přepínač pro CAN sběrnici
10	24V napájení
11	Port pro CAN sběrnici
12	Ethernetový port
13	Status LED diody
14	Sběrnice pro připojení TM4 modulů
15	Sériová linka pro RS 232 nebo RS 485
16	Sériová linka pro RS 486
17	Vestavěné vstupy
18	Sběrnicové připojení pro TM3/TM2 moduly
19	Ochranný kryt
20	Úchyt pro zámek

2.1.1 Analogový modul TM3

Analogový modul TM3AM6 obsahuje,

- 4 analogové vstupy,
- 2 analogové výstupy.

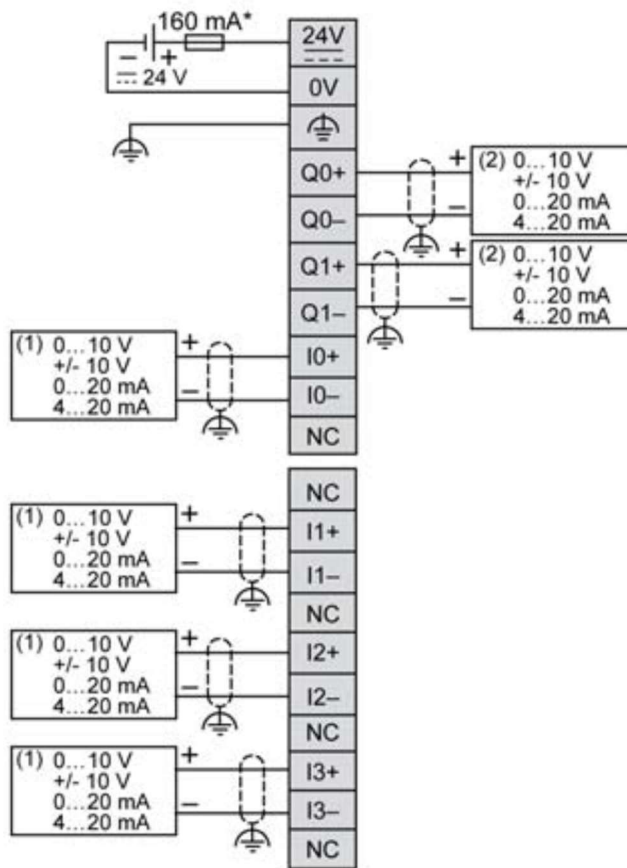
Zapojení je znázorněné na obr. 2.2.

V nastavení modulu v software SoMachine lze nastavit typ a rozsah vstupního, nebo výstupního signálu a to:

- proudový v rozsahu 0 až 20 mA, nebo 4 až 20 mA,
- napěťový v rozsahu 0 až 10 V, nebo -10 až +10 V.

Modul byl připnut na DIN lištu a do správné sběrnice PLC. Dále jsem musel modul připojit na externí 24 V zdroj napětí.

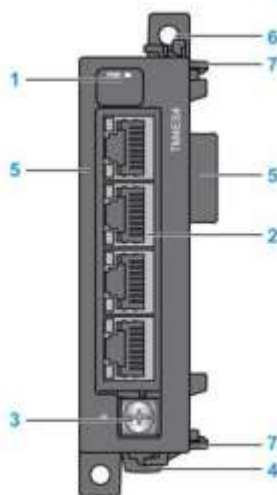
Vstupní signál se vzorkuje s periodou 1 ms, nebo 10 ms, lze upravit v parametrech modulu. Použitý analogově/digitální převodník je sigma delta. Rozlišení převodníku je 12bitů, popřípadě 11bitů a 1bit pro znaménko. (Schneider Electric, 2014)



Obr. 2.2 Zapojení vstupů a výstupů na modulu

2.1.2 Ethernetový modul TM4

Pro komunikaci mezi PLC a PC byl využit ethernetový modul TM4ES4 (obr. 2.3). Modul je připojen k PLC na sběrnici pro moduly TM4. Ethernetový modul pracuje v režimu standalone, to znamená, že funguje jako klasický ethernetový switch, z PLC se pouze napájí. Pro propojení mezi PLC, PC a uživatelským panelem HMI byli použity 3 síťové patch kabely. (Schneider Electric, 2014)



Obr. 2.3 – Rozmístění komponentů na modulu

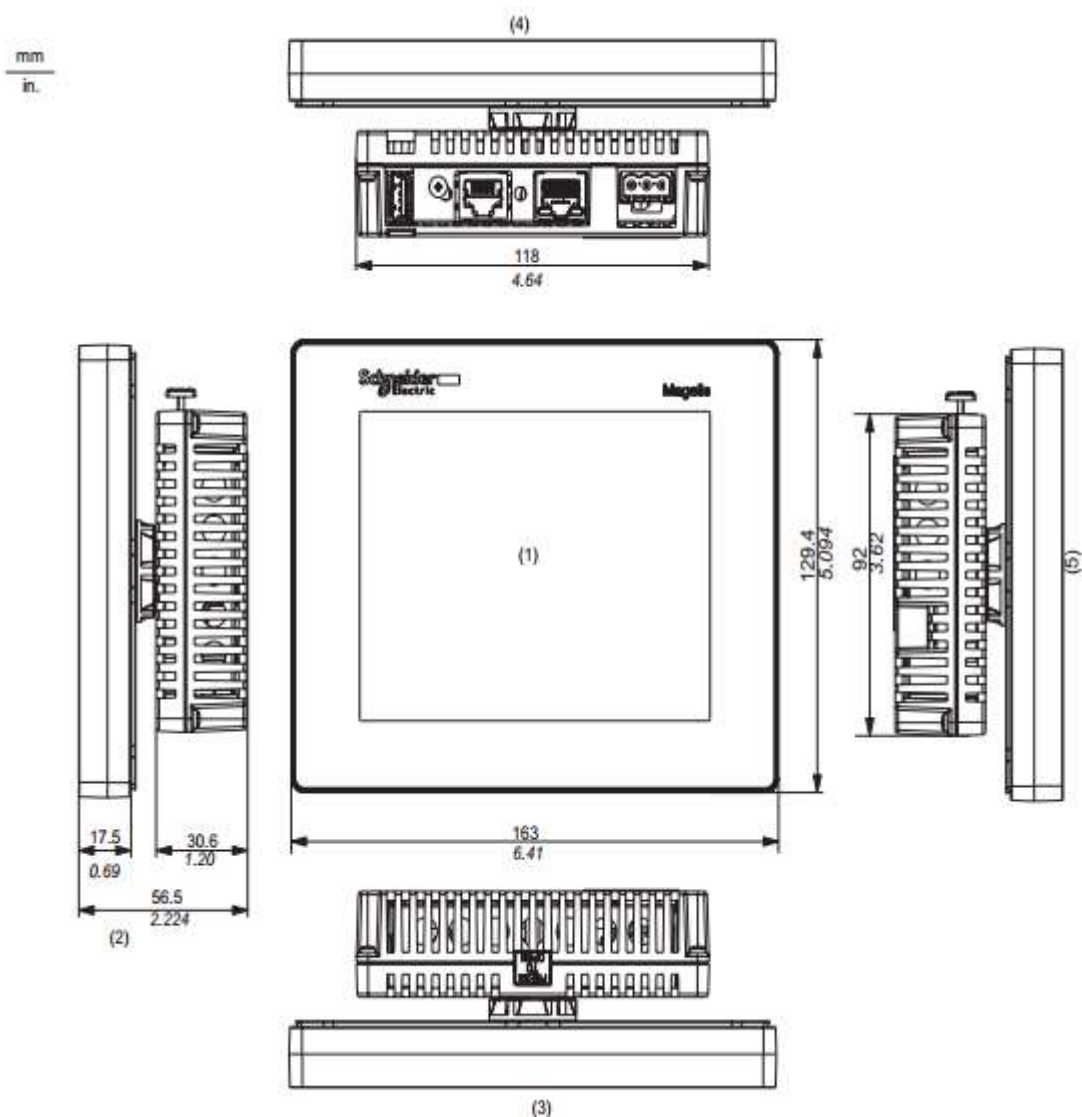
Tab. 2.2 – Legenda pro modul TM4ES4

číslo	Popis
1	LED dioda zobrazující stav napětí
2	4 Ethenetové RJ45 zásuvky
3	Šroub pro připojení země
4	Úchyt pro DIN lištu
5	Konektor pro připojení dalšího TM4 modulu
6	Zámek pro připojený modul
7	Úchyt pro připojený modul

2.1.3 Uživatelský panel HMIS5T

HMIS5T (obr. 2.4) je zadní, ovládací modul pro dotykové uživatelské panely HMI STU 635, který je 3,5" a panel HMI STU 855, který je 5,7". Oba dva panely jsou vybaveny zobrazovací technologií TFT LCD, 65 K barev a podsvícený LED, dotyková vrstva je vyrobena z odporové sítě. K vytvoření uživatelského panelu je potřeba speciální software Vijeo Designer, který je součástí SoMachine. Displej musí být napájen 24 V stejnosměrným napětím. Zadní

modul obsahuje konektory na připojení USB, Ethernet a Sériové linky. (Schneider Electric, 2014)



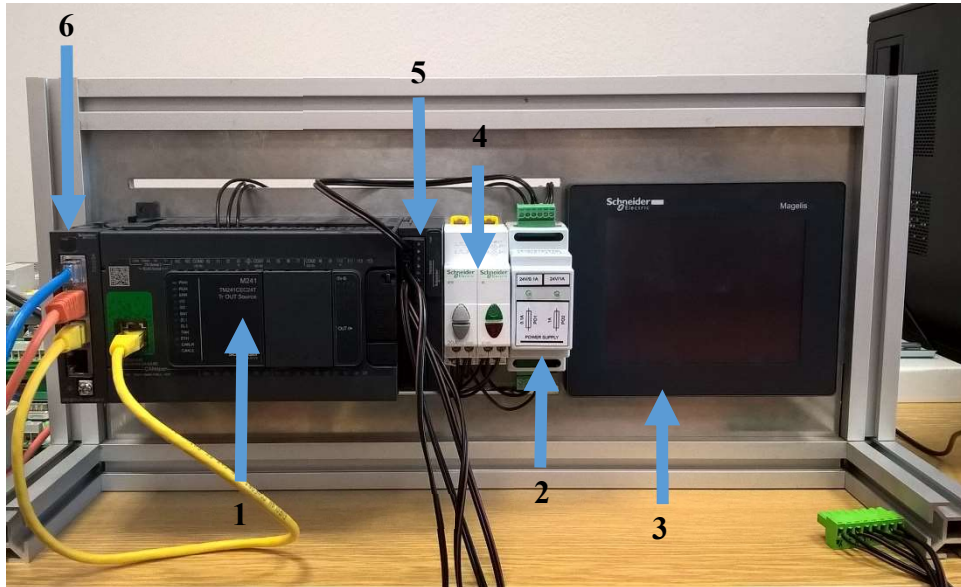
Obr. 2.4 – Displej se zadním modulem

Tab. 2.3 – Legenda k displeji

číslo	Popis
1	Přední panel
2	Pohled zprava
3	Pohled shora
4	Pohled ze spodu
5	Pohled zleva

2.2 Laboratorní zařízení

Konstrukce z kovového rámu (obr. 2.5), na kterém je upevněna DIN lišta. Na liště je připevněno PLC spolu s rozšiřujícími moduly, 24 V napájením a modulem s tlačítky pro názornou ukázkou funkčnosti PLC. V konstrukci je vyříznutá díra pro uživatelský panel.



Obr. 2.5 – Konstrukce PLC

Tab. 2.4 – Legenda konstrukce

číslo	Popis
1	Programovatelný automat
2	Napájení
3	Uživatelské rozhraní
4	Tlačítka
5	Modul TM3
6	Modul TM4

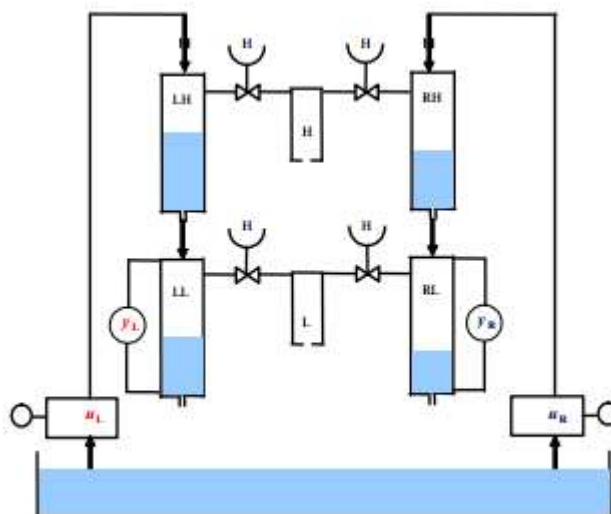
Konstrukce hydraulicko-pneumatické soustavy je zobrazena na obr. 2.6.



Obr. 2.6 – Konstrukce hydraulicko-pneumatické soustavy

2.3 Hydraulicko-pneumatická soustava

Laboratorní model hydraulicko-pneumatické soustavy byl vytvořen pracovníky katedry řízení procesů a výpočetní techniky Fakulty chemicko-technologické Univerzity Pardubice. Model obsahuje kombinaci hydraulických a pneumatických prvků. Skládá se ze čtyř vodních nádrží a dvou vzdušníků, viz obr. 2.7 (Macháček, 2005).



Obr. 2.7 – Schéma hydraulicko-pneumatické soustavy

Hlavní částí modelu jsou čtyři vodní nádrže s tím, že pravé nádrže soustavy mají menší průměr. Voda je pumpována ze zásobníku dvěma čerpadly do horních nádrží. Z horních nádrží vytéká voda přes otvor do dolních nádrží a z nich zpátky do zásobníku. Průtok vody je ovládán vstupními signály čerpadel, který je v rozsahu 0 až 5 V. Výška hladiny je měřena diferenčním tlakovým senzorem s výstupním signálem 0 až 10 V. Senzor měří rozdíl tlaků nad hladinou a pod hladinou. Vzduchové prostory nad hladinou vody jsou propojeny pneumatickými obvody, vzdušníky. Otvory ve vzdušnicích slouží k vyrovnání tlaku mezi nádržemi a atmosférou. Výsledkem je dvou rozměrová soustava s vnitřními interakcemi. Pomocí čtyř ventilů lze nastavit míru vzájemného ovlivňování mezi nádržemi (Honc, 2012).

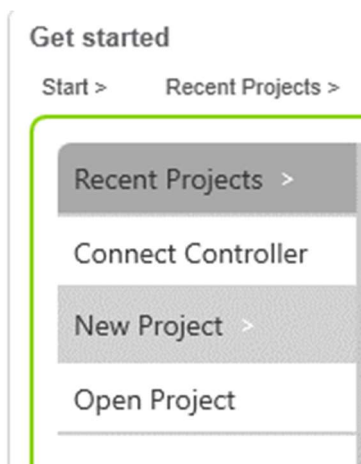
2.4 SoMachine

SoMachine se profesionální software pro programování, nastavování a řízení programovatelných automatů od firmy Schneider Electric. Program se skládá ze dvou částí. První částí je Logic Builder, software, ve kterém můžeme programovat samotné PLC. Druhou částí je Vijeo-Designer, zde se vytvářejí uživatelské prostředí pro panely HMI. (Schneider Electric, 2014)

Instalace probíhá přes speciální software SoMachine Configuration Manager. Přes tento program je umožněno nainstalovat celý balík SoMachine, nebo jen některé jeho funkce, a také SoMachine aktualizovat a spravovat licence. V práci byl použit SoMachine verze 4.1 SP2 a

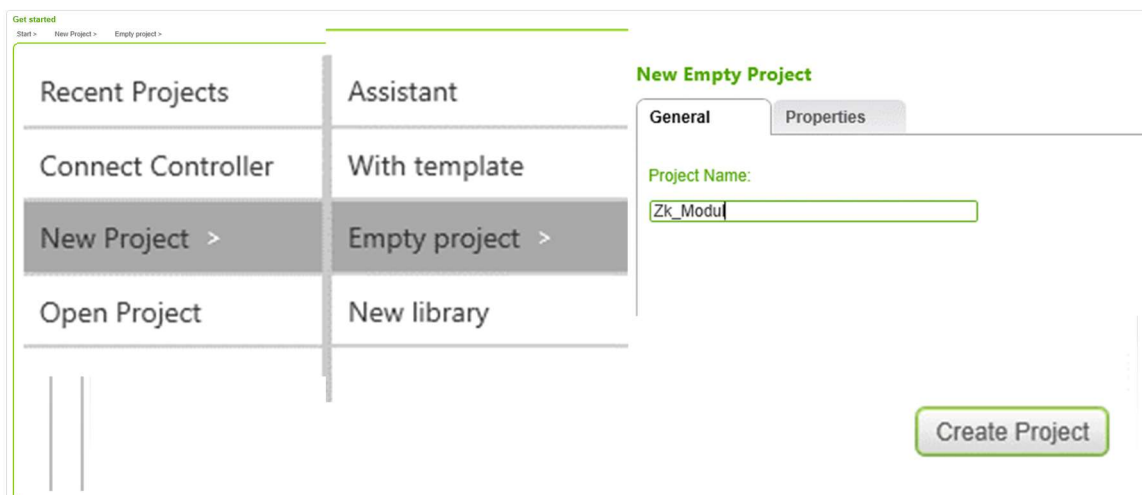
Vijeo-Designer SP3.1. Uvedený software je k dostání v příloženém odkazu¹, program lze vyzkoušet zdarma po dobu 21 dní.

Vytvoření projektu



Obr. 2.8 – Základní obrazovka

Po spuštění SoMachine uvidíme obr. 2.8, základní obrazovku programu, ze které se lze rychle přepnout na poslední zobrazovaný projekt, připojit se k PLC nebo HMI, vytvořit projekt a otevřít existující projekt. Následující text popisuje základy při vytváření nového projektu.

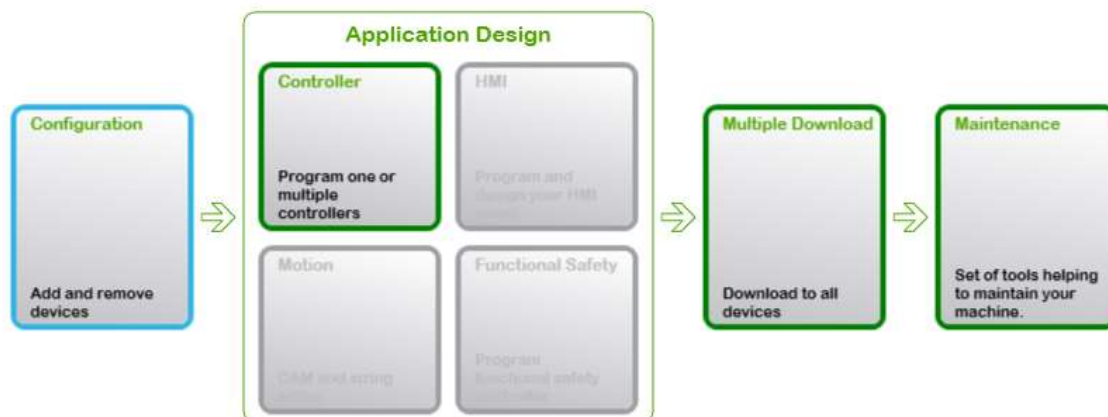


Obr. 2.9 – Postup vytváření projektu

Podle obr. 2.9 po kliknutí na nový projekt byl vybrán prázdný projekt, který lze pojmenovat dle uživatelského uvážení, dále je zde možnost upravit vlastnosti projektu (verzování,

¹ <http://www2.schneider-electric.cz/software/software-ke-stazeni/somachine/>

jméno autora atd.), po vyplnění těchto částí, uživatel může potvrdit vytvoření projektu. Další obrazovkou je pracovní prostor projektu, obr. 2.10.

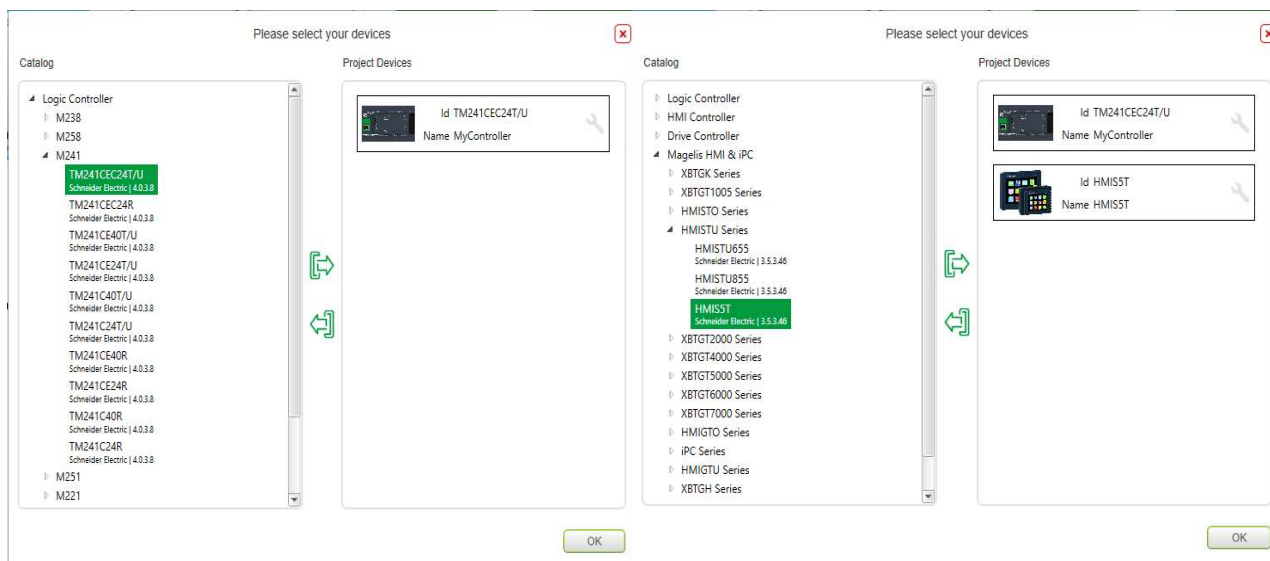


Obr. 2.10 – Pracovní prostor

Zde je několik důležitých částí:

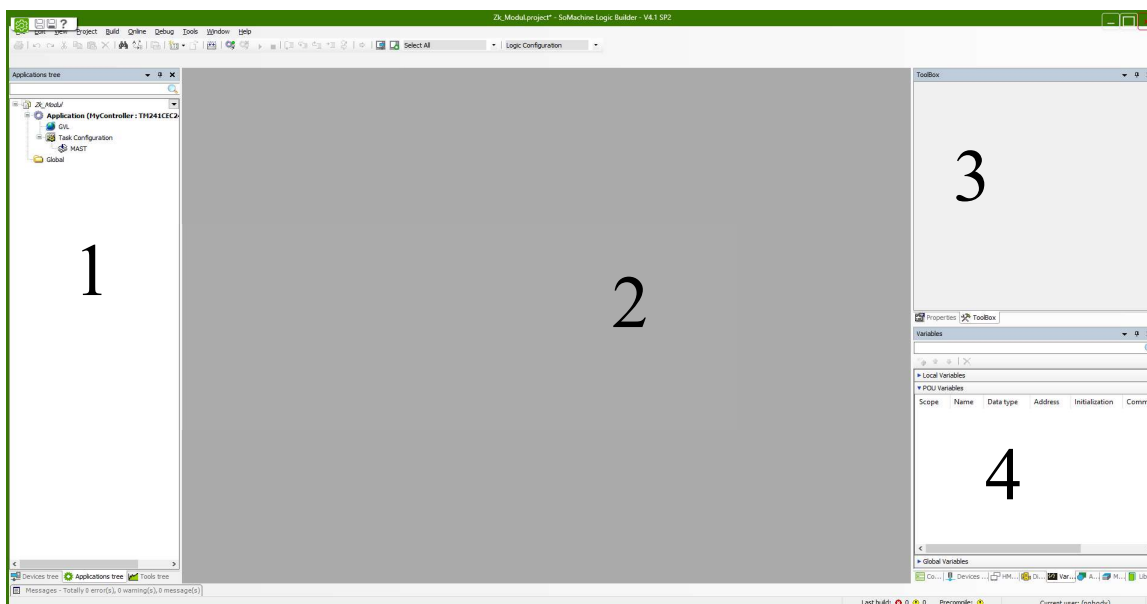
- Configuration – výběr zařízení.
 - Zde lze k projektu přidat určité zařízení. Jeho výběr by měl být první krok po vytvoření. Po přidání se nahrají všechny důležité knihovny a software bude umět pracovat se vstupy a výstupy.
- Application Design – vytvoření aplikace.
 - Controller – vytvoření programu v prostředí Logic Builder pro PLC.
 - HMI – vytváření uživatelského prostředí v software Vijeo pro panely HMI.
 - Motion a Functional Safety jsou specializované nástroje pro určité typy modulů.
- Multiple Download – nahrání programu do PLC či HMI.
- Maintenance – nástroje pro správu PLC. (aktualizace firmware atd.)

Do projektu se vždy musí přidat PLC a moduly se kterými bude uživatel pracovat, to lze provést v části Configuration a přidat ty, které se budou využívat, podle obr. 2.11, samotný programovatelný automat a uživatelský panel HMI.



Obr. 2.11 – Přidání a) programovatelného automatu b) uživatelského panelu HMI

V menu Configuration a podnabídce Logic Controller bylo vybráno PLC, které bude použito M241 typ TM241CEC24T/U. Také byl vybrán panel, který bude použit v případě této práce to je Magelis HMI řády HMISTU typ HMIS5T. Po potvrzení zadaných modulů lze přejít na programování aplikace pro PLC v menu Controller, obr. 2.12.



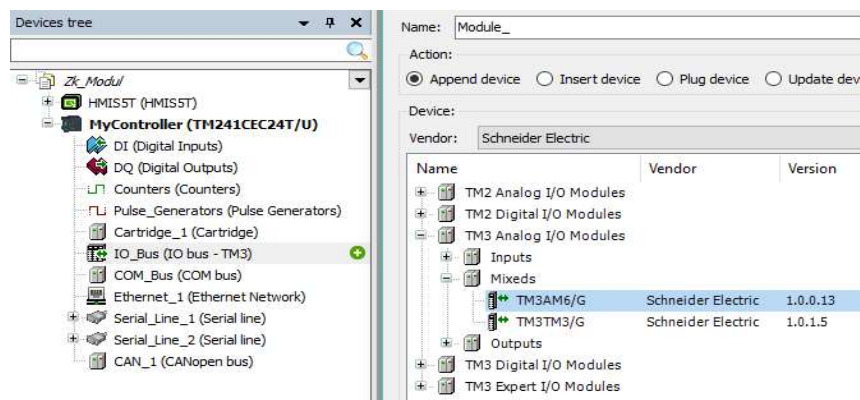
Obr. 2.12 – Pracovní okno Logic Builderu

Tab. 2.5 – Legenda k Logic Builderu

Číslo	Popis
1	Záložky pro správu PLC a programu
2	Pracovní prostor
3	ToolBox, ze kterého do pracovního prostoru přesunujeme funkce
4	Záložky pro správu programu a sdílení proměnných

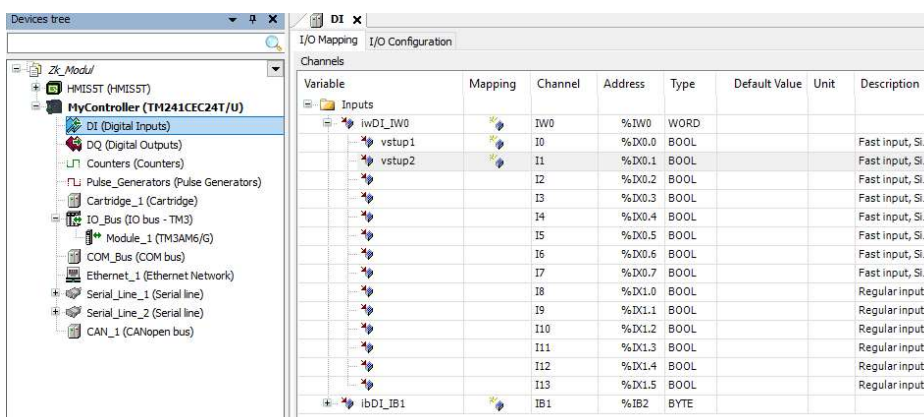
Vytvoření programu

Před vytvořením programu v záložce Device tree (obr 2.12 číslo 1) je nutné do projektu přidat všechny přídatné moduly. V případě této práce to je analogový vstupně výstupní modul TM3, obr. 2.13.



Obr. 2.13 – Přidání modulu

Dále lze pojmenovat vstupy a výstupy pro lepší orientaci při psaní programu, obr. 2.14. Tento krok lze přeskočit, protože k pojmenování dojde v pozdější části.

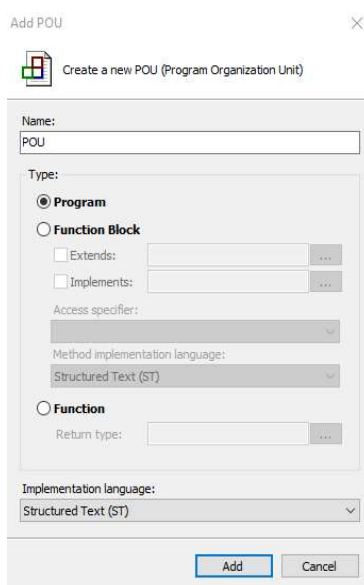


Obr. 2.14 – Pojmenování vstupů a výstupů

A pokračovat vytvořením samotného programu v záložce Application Tree, obr. 2.15.



Obr. 2.15– Přidání POU



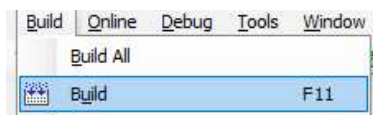
Obr. 2.16 – Průběh vytváření POU

V nabídce podle obr. 2.16 lze zvolit název, typ (Program, Funkční blok nebo Funkce) a programovací jazyk, ve kterém program bude psán. Po potvrzení se objeví hlavní programovací obrazovka, kde je umožněno vytvářet proměnné a psát program. Pro ilustrační účely a vyzkoušení všech funkcí programu SoMachine byl vytvořen následující program a proměnné.

Příklad 1 – Proměnné
<pre>PROGRAM POU VAR vstup1: BOOL; vstup2: BOOL; vystup1: BOOL; vystup2: BOOL; END VAR</pre>
Příklad 2 – Program
<pre>vystup1:= vstup1 OR vstup2; vystup2:= NOT(vystup1);</pre>

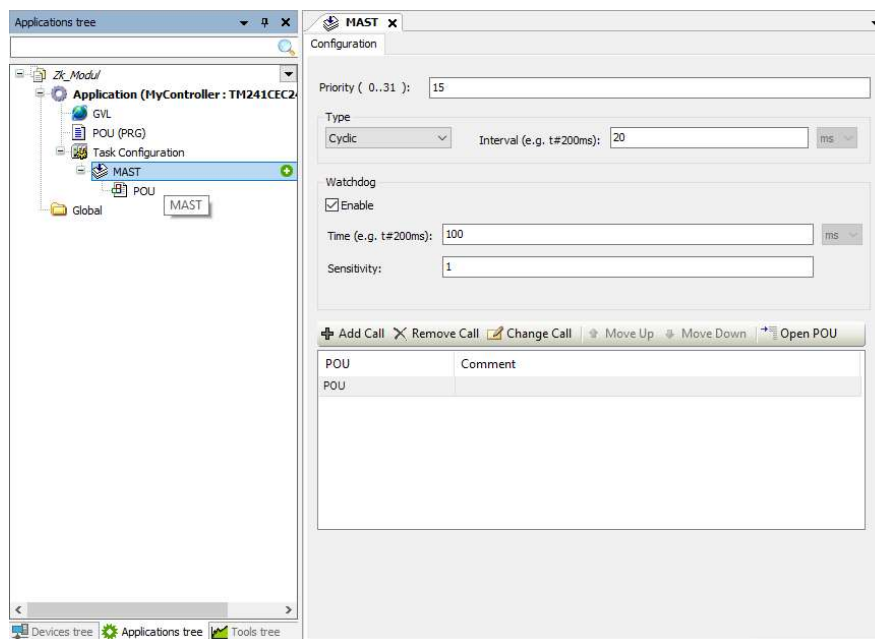
Nově vytvořené proměnné v POU je vhodné propojit se vstupy a výstupy podle obr. 2.14 a to dvojklikem na jméno proměnné a výběrem nově vytvořené v POU.

Po napsání programu a propojení proměnných se vstupy a výstupy následuje kompilace projektu z horní menu Build podle obr. 2.17.



Obr. 2.17 – Kompilace

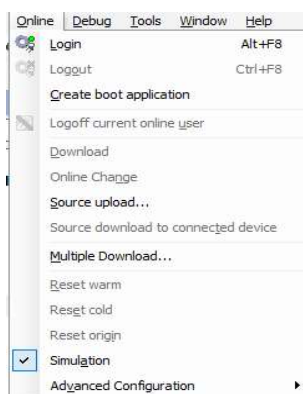
Po zkompilování projektu se musí vložit POU do MAST (Master Task) PLC. To se provádí ze záložky Application Tree dvoj kliknutím na MAST. V tomto okně je umožněno nastavit priority různých POU, nastavit si WatchDog aby restartoval PLC při zacyklení programu a také je zde umožněno pomocí tlačítka Add Call přidávat POU. Pokud je jen jeden program na vykonání, obrazovka by měla vypadat podle obr. 2.18.



Obr. 2.18 – Přidání POU do MAST

Simulace a nahrání programu do PLC

Pokud jsou splněny všechny předchozí kroky, program lze vyzkoušet v simulaci. V některých případech je simulace těžce proveditelná a tento krok se přeskakuje. Z horního menu Online bylo vybráno Simulation, podle obr. 2.19, což se projeví červeným nápisem Simulation v dolní liště programu. Poté ze stejného menu jsem vybral Login, během připojování lze rovnou nahrát program do simulačního prostředí.



Obr. 2.19 – Menu Online

Pokud se program do simulačního prostředí nenahrál při připojení, lze ho tam nahrát pomocí Multiple Download... ze stejného menu. Po spuštění simulace je hlavní obrazovka programu změněna tak, že uživatel může libovolně měnit proměnné, obr. 2.20.

Expression	Type	Value	Prepared value	Address	Comment
vstup1	BOOL	FALSE			
vstup2	BOOL	FALSE			
vystup1	BOOL	FALSE			
vystup2	BOOL	FALSE			

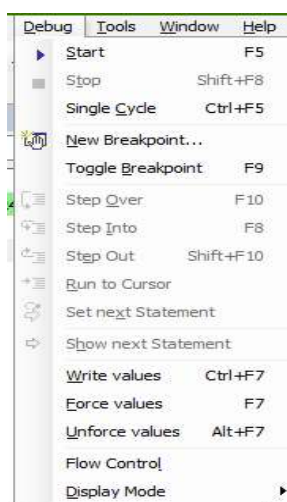

```

1 | vystup1 FALSE := vstup1 FALSE OR vstup2 FALSE;
2 | vystup2 FALSE := NOT (vystup1 FALSE); RETURN;

```

Obr. 2.20 – POU simulace

Zde si uživatel může připravit hodnoty na zapsání při spuštění programu, pomocí okna Prepared value. V uvedeném příkladu byla zapsána hodnota TRUE na vstup1. Pro zapsání hodnoty v simulačním prostředí jsem musel z menu Debug, obr. 2.21, vybrat Start a poté Write values.



Obr. 2.21 – Menu Debug

Okno programu se změní podle obr. 2.22. Tím že na vstup1 byla zapsána hodnota TRUE program se vykonal a na vystup1 se podle programu zapsala hodnota TRUE a vystup2 se změnil z TRUE na FALSE. Program je tedy funkční a je čas přejít k nahrání programu do PLC.

MyController.Application.POU					
Expression	Type	Value	Prepared value	Address	Comment
vstup1	BOOL	TRUE			
vstup2	BOOL	FALSE			
vystup1	BOOL	TRUE			
vystup2	BOOL	FALSE			


```

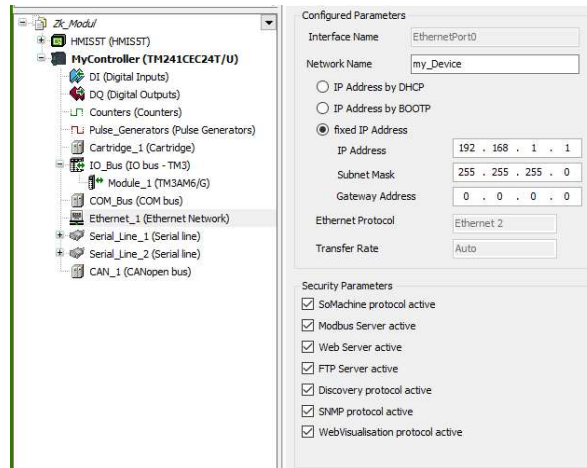
1 | vystup1 TRUE := vstup1 TRUE OR vstup2 FALSE;
2 | vystup2 FALSE := NOT (vystup1 TRUE); RETURN;

```

Obr. 2.22 – Okno programu po zapsání hodnot

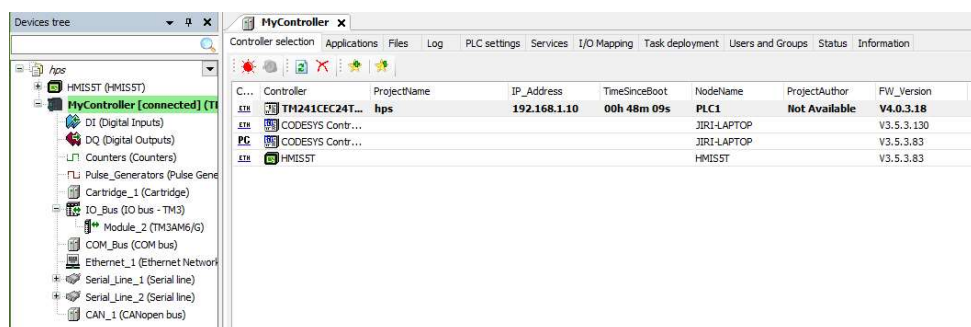
Nahrání programu do PLC

Před nahráním programu do PLC se musí vyplnit adresy PLC v záložce Device tree, podzáložce Ethernet_1, podle obr. 2.23. V této práci komunikuji s PLC pouze pomocí rozhraní Ethernet, proto je důležité, aby PLC, PC i HMI byli ve stejné podsíti. PLC si pomocí Ethernetu vyměňují informace s panelem HMI.



Obr. 2.23 – Adresy

Po vyplnění adres je vše připravené k nahrání programu do PLC. Automat, který je programován se musí vybrat za aktivní v záložce Device tree podle obr. 2.24. Dále lze pokračovat tak jako by byl program nahráván do simulace, obr. 2.19. Před nahráním programu do PLC se objeví potvrzující obrazovka, pokud je uživatel jistý s výběrem PLC stiskne ALT+F a tím potvrdí výběr automatu a provede se nahrání programu.

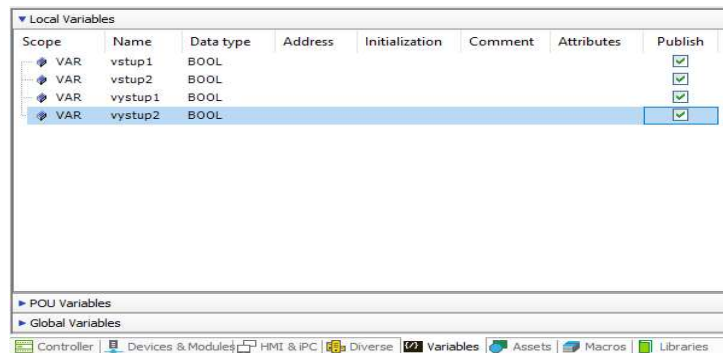


Obr. 2.24 – Vybrání aktivního PLC

2.4.1 Vijeo Designer

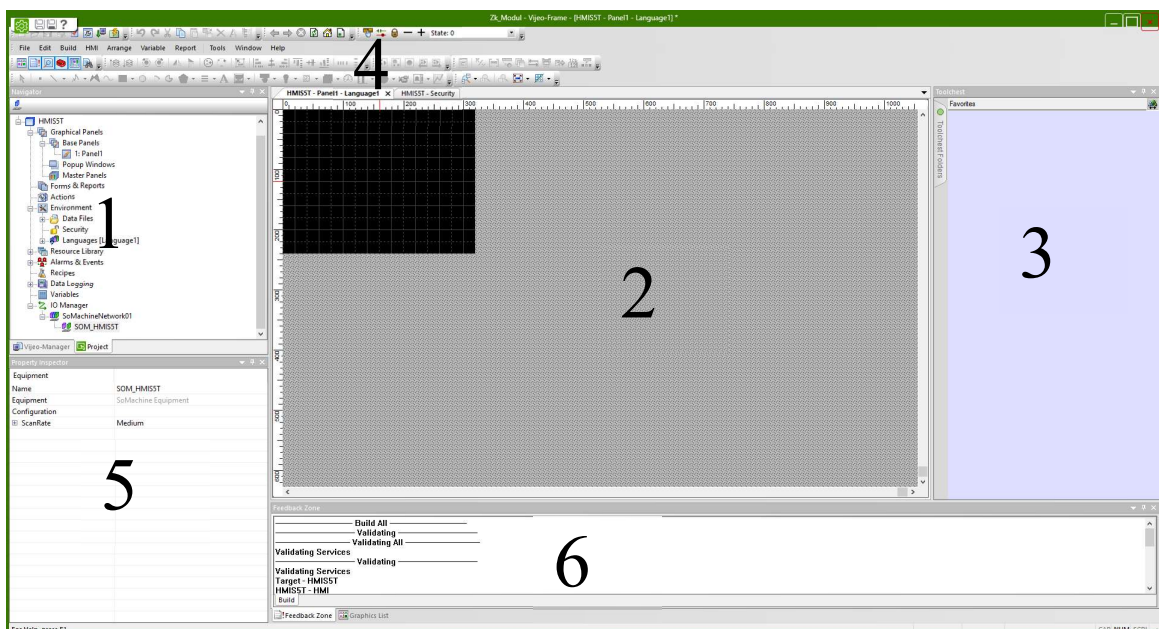
Je grafické prostředí určené pro tvorbu uživatelských rozhraní pro panely HMI. Toto prostředí je součástí SoMachine a je přístupné přes hlavní pracovní prostor z menu Application Design výběrem HMI.

Před vytvářením aplikace pro HMI je důležité, aby mezi PLC a HMI byli sdílené proměnné. Toho bylo docíleno ještě v Logic Builderu. V pravé dolní části v záložce Variables byly vybrány ty proměnné, které budou využity v obou systémech, podle obr. 2.25.



Obr. 2.25 – Sdílení proměnných

Po přepnutí do HMI se objeví následující výchozí obrazovka, obr. 2.26.

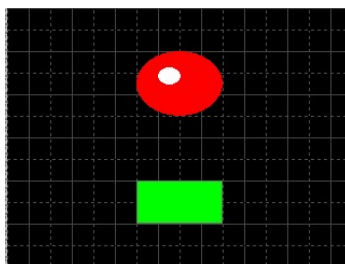


Obr. 2.26 – Pracovní plocha Vijeo

Tab. 2.6 – Legenda

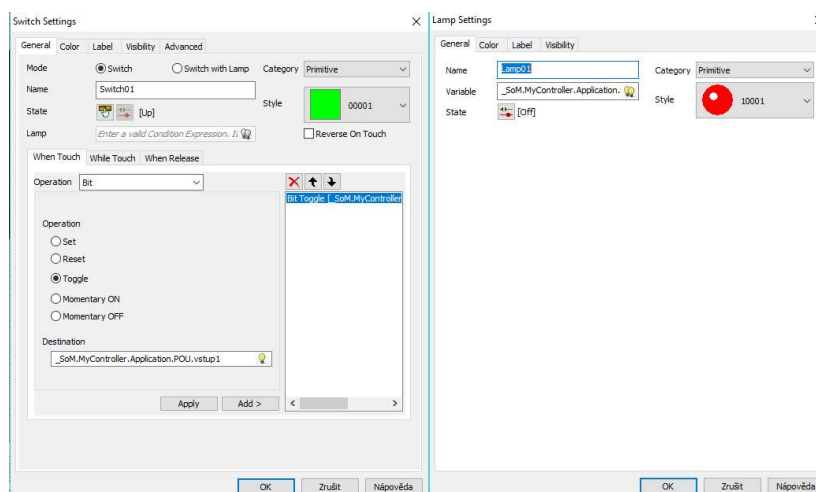
Číslo	Popis
1	Strom projektu
2	Hlavní pracovní plocha
3	Nabídka nejčastěji používaných komponentů
4	Hlavní menu a nástroje pro tvorbu uživatelského prostředí
5	Nastavení vlastností komponenty
6	Zpětná vazba po kompilaci

Před začátkem návrhu displeje je důležité vyplnit adresu HMI v nastavení HMIS5T, které je přístupné před dvojklik na HMIS5T ve stromu projektu. Také je podstatné vyplnit správná jména zařízení v IO manageru, aby se panel spároval s PLC. Pro účely simulace je vytvořeno jednoduché uživatelské rozhraní, obr. 2.27.



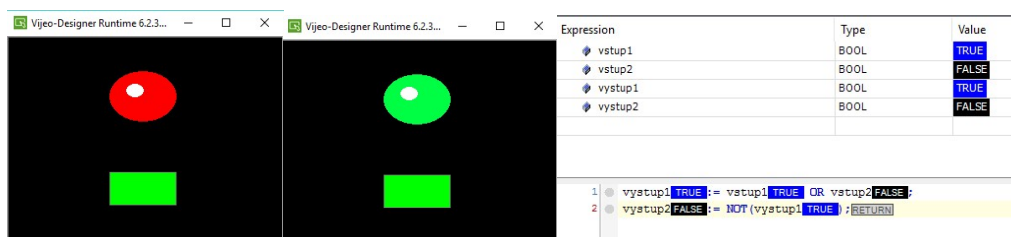
Obr. 2.27 – Uživatelské rozhraní

Rozhraní obsahuje jeden spínač (Switch) a lampu. Komponenty jsou nastavené podle obr. 2.28. Důležité je propojit prvek s proměnnou z POU. Je zde také možnost nastavit různé události, které se mají stát, když se provede stisk nebo změna tlačítka.



Obr. 2.28 – Nastavení a) spínače b) lampy

V menu Build lze program zkompileovat a pokud se nevyskytne žádný error při kompilaci může se ze stejného menu vyzkoušet simulovat program. Pokud v Logic builderu je pořád zapnuta simulace projektu a ve Vijeo vše proběhlo v pořádku, program by měl vypadat následovně, obr. 2.29 při stisku spínače se změní barva lampy a zároveň se přepíše hodnoty proměnných v POU. Nahrání projektu do panelu proběhne spolu s nahráním programu do PLC.



Obr. 2.29 – Průběh simulace a) spuštění b) stisk spínače c) kontrola proměnných

2.5 Vlastní řešení úlohy

Součástí práce bylo vytvoření programu pro PLC, který ovládá čerpadla v soustavě, čte hodnoty ze snímačů hladiny a všechny tyto hodnoty zapisuje do své interní paměti, aby s nimi mohl uživatel pracovat. Dále bylo vytvořeno uživatelské prostředí pro panel HMI, ze kterého uživatel může nastavovat výkon čerpadla v rozsahu 0 až 100 % nebo se může přepnout do automatického režimu, kde nastaví žádanou hodnotu hladiny v centimetrech a zabudovaný programový blok PID regulátoru tuto hodnotu udržuje. Součástí prostředí je také vizualizace průběhu hodnoty vstupů v čase a volitelná obrazovka pro měření napětí na snímačích.

2.5.1 Návrh programu

Manuální režim

Uživatel může manuálně ovládat čerpadlo hodnotou, kterou zadá na vstup. Vstupní informace je převedena na výstup. Pro uživatelské pohodlí jsou vstupní hodnoty v rozsahu 0 až 100 %, kde 0 % je 0 V a 100 % 5 V. Pro každé čerpadlo lze výkon ovládat zvlášť.

Automatický režim

Ve druhém režimu jsou čerpadla ovládány PID regulátory. O výpočet akční veličiny se stará programový blok PID regulátoru. Tento blok je z knihovny od sdružení oscat.de. Blok přesně vypočítává akční zásah podle zadaných parametrů, integrál se počítá lichoběžníkovou metodou a také je zde naprogramován anti-wind up. Žádanou hodnotu nastavuje uživatel z panelu, zadává se v centimetrech. Protože systém nevyžaduje velmi rychlé reakce, a perioda vzorkování je velmi krátká můžeme brát v úvahu spojitý tvar rovnice PID regulátoru. Rovnice je ve tvaru

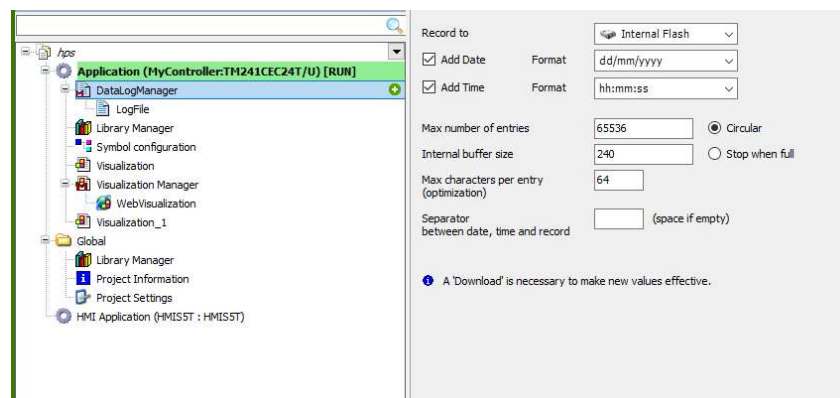
$$u(t) = r_0(e(t) + \frac{1}{T_i} \int_{\tau=0}^t e(\tau) d\tau + T_d \frac{de(t)}{dt}). \quad (2.12)$$

Převod hodnot

Všechny vstupní hodnoty jsou v uživatelsky přívětivých jednotkách, tedy centimetry a procentní rozsah. PLC ovšem pracuje s jednotou milivoly, musel jsem všechny jednotky převádět z uživatelsky přívětivých na mV. Převod centimetrů na milivoly jsem docílil ze znalosti statické charakteristiky snímače hladiny, pro většinu rozsahu je tento vztah lineární. Procentní rozsah bylo jednodušší naprogramovat, když 5 V je 100 %.

Ukládání hodnot

Ukládání hodnot má na starost druhá POU v cyklu MAST. Tato POU obsahuje jeden FBD blok, který má ovládat celou činnost ukládání hodnot. Hodnoty lze uložit jen v případě, když jsou ve speciálním datovém typu WSTRING, proto je zapotřebí všechny veličiny před uložením převést na tento typ. FBD blok má dva vstupy první je pro datový typ WSTRING a druhý je časovač, který jsem pevně nastavil na 200ms. Je tu zapotřebí spolupráce dvou programovacích jazyků ST a FBD. V jazyce ST se sestaví WSTRING s potřebnými údaji pro uložení a FBD provede uložení. Aby bylo ukládání hodnot kompletní, musí se vytvořit LogFile, viz obr. 2.30, v záložce Tools Tree v Logic Builderu. Kliknutím na plus v DataLogManager.

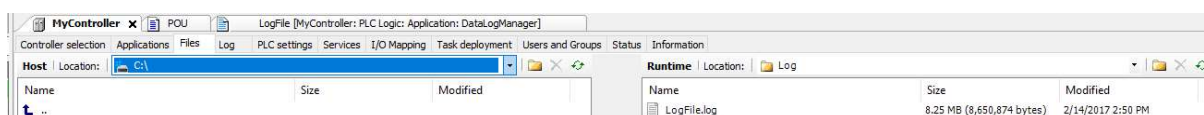


Obr. 2.30 – Vytvoření Logfile

Tento LogFile musí mít stejné pojmenování jako blok ve FBD. Pro stáhnutí logfile lze využít SoMachine, který je připojen k PLC anebo ftp server PLC přístupný z webové adresy ftp://192.168.1.10. Výchozí jméno do PLC je USER a heslo USER. Soubor se nachází ve složce /usr/log, viz obr. 2.31. Nebo ze záložky Device Tree v rozšířeném nastavení PLC je umožněno přesouvat soubory mezi PC a PLC přímo ze software SoMachine, které je ukázáno na obr. 2.32.



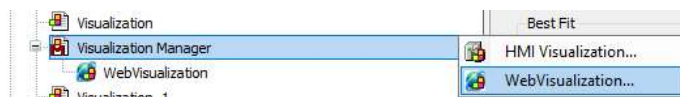
Obr. 2.31 – LogFile přístupný z FTP



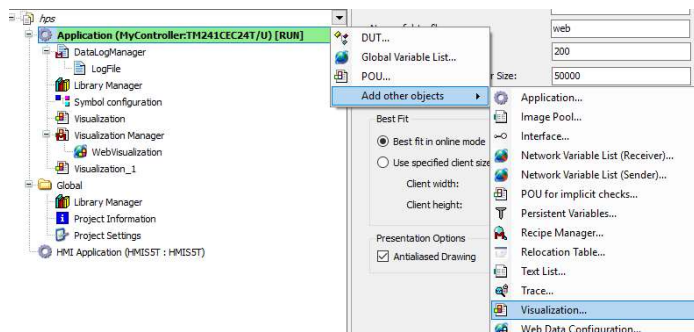
Obr. 2.32 – LogFile přístupný ze SoMachine

Webové rozhraní

V rámci vyzkoušení programových možností PLC jsem vytvořil jednoduché webové rozhraní pro ovládání čerpadel a čtení hodnot. Toto rozhraní se vytvoří na záložce Tools tree podle obr. 2.33 Protože jsem měl dva režimy, potřeboval jsem dvě stránky, druhou jsem přidal podle obr. 2.34, mezi stránkami lze libovolně přepínat. Také lze nastavit výchozí stránku.

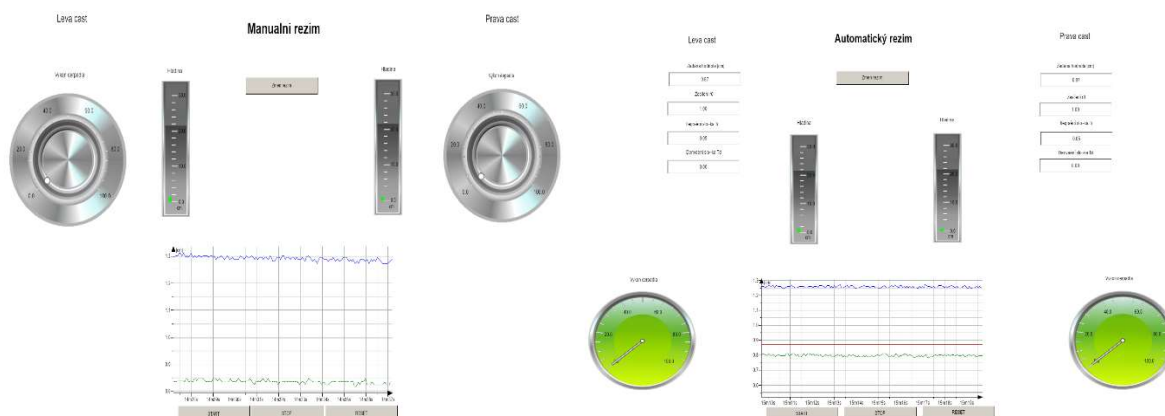


Obr. 2.33 – Přidání webové vizualizace



Obr. 2.34 – Přidání další webové stránky

Rozhraní je zobrazeno na obr. 2.35.



Obr. 2.35 – Webové rozhraní a) manuální režim b) automatický režim

Je uděláno velice jednoduše, jen pro ověření funkcí PLC. Pokud by uživatel nastavoval hodnoty z webových stránek, nemusejí se rovnat nastavení z uživatelského panelu. Proto doporučuji, aby webové rozhraní využívali jen zkušenější uživatelé, nebo se musí využívat jen jedno rozhraní pro nastavení buď panel, nebo stránky. Rozhraní je přístupné z webové adresy, při výchozím nastavením programu, 192.168.1.10: 8080/web.htm. Koncovka web.htm lze změnit v nastavení vizualizace. Na stránce 192.168.1.10 se také nachází základní diagnostika PLC.

2.5.2 Návrh displeje

Displej byl navrhnout tak aby byl co nejjednodušší na použití a ukazoval ty nejdůležitější informace. Po spuštění displeje se načte první obrazovka s manuálním režimem, obr. 2.36.



Obr. 2.36 – Obrazovka manuálního režimu

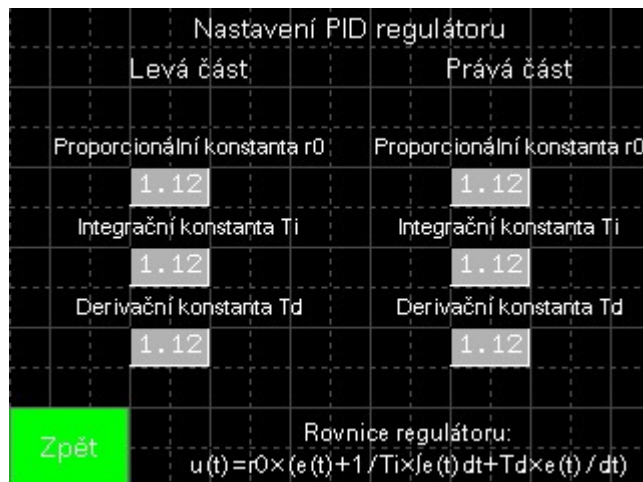
Na této obrazovce může uživatel přímo nastavit výkon požadovaného čerpadla v procentech a čerpadlo ihned spustí svoji činnost. Je zde zobrazena hladina v centimetrech a uživatel se může přepnout na obrazovku, která zobrazuje regulační pochod. Dále se uživatel může přepnout do automatického režimu, obr. 2.37.



Obr. 2.37 – Obrazovka automatického režimu

Kde si uživatel nastaví požadovanou hodnotu a regulátor se automaticky spustí s předdefinovanými hodnoty r_0 a T_i . Když uživatel přepne z manuálního režimu na automatický a hladina byla nenulová, je tato hladina nastavená jako žádaná hodnota. Uživatel si může

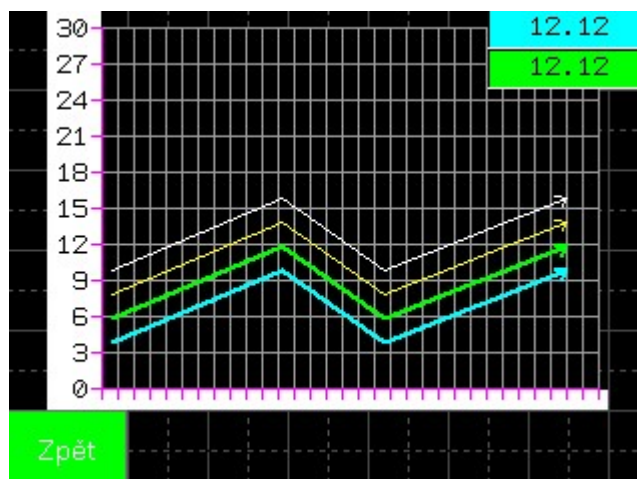
nastavit vlastní hodnoty PID regulátoru na obrazovce Nastav. PID, obr. 2.38. Tato obrazovka obsahuje informaci o regulované veličině a akční veličině, tedy výkonu čerpadla.



Obr. 2.38 – Obrazovka nastavení PID

Na obrazovce nastavení PID si uživatel může nastavit a experimentovat s různými hodnotami r_0 , T_i a T_d . Musí ovšem brát v úvahu uvedený tvar rovnice, který je zde zapsán ve spojitém vyjádření.

Poslední obrazovka je s regulačním pochodem soustavy, obr. 2.39.



Obr. 2.39 – Obrazovka regulačního pochodu

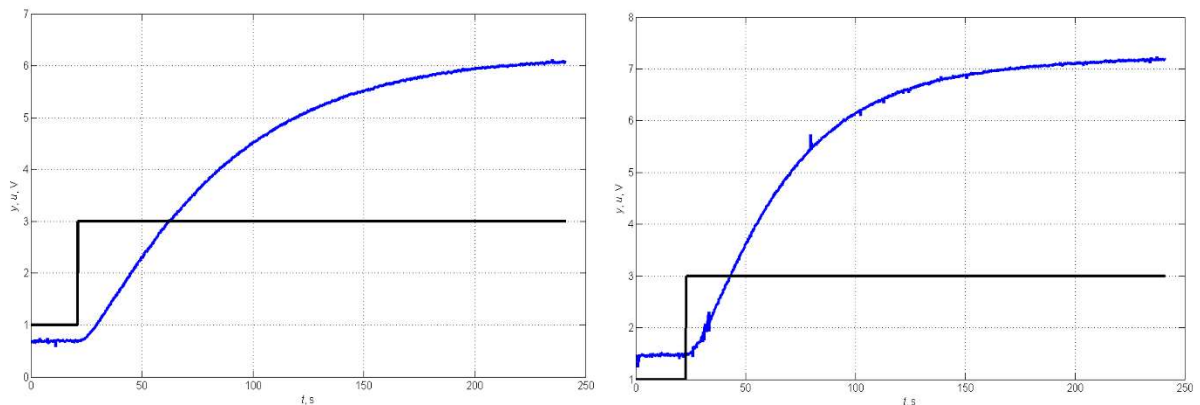
Zde je zobrazen regulační pochod a aktuální hodnoty čidel, převedeny na cm. Na ypsilonové ose jsou zobrazeny žádané hodnoty a regulační veličiny. X-ová je časová.

Každé tlačítko v aplikaci je nastaveno tak aby si neohrozilo pochod při chybě displeje. Je zde několik ochran např. pevné procentní rozsahy, pevný rozsah žádané hodnoty, nemožnost kliknout na hodnoty, ke kterým uživatel nesmí mít přístup. Tlačítka pro přepnutí režimů

obsahují nulovací funkci pro výkon čerpadla a také funkci ochrany režimu, kdy se nemůže nastavovat výkon čerpadla při automatickém režimu a žádaná hodnota při manuálním. Zkušenější uživatelé si mohou v software Vijeo odblokovat obrazovku s údaji uvedenými v jednotkách se kterými pracuje PLC, tedy milivolty.

2.5.3 Identifikace parametrů z přechodové charakteristiky

Pro identifikaci parametrů přenosů soustav jsem obě dvě nádrže od sebe oddělil, uzavřením ventilu u vzdušníků, aby na sebe nepůsobili a provedl jsem měření pomocí PLC. Čerpadla jsem zapnul na 20 % výkonu, což se rovná jednomu voltu, a nechal ustálit. Poté jsem výkon zvedl na 60 %, tedy na 3 V, a znovu nechal ustálit. Výsledek měření je zobrazen na obr. 2.40.



Obr. 2.40 – Odezva regulačního systému na skokovou změnu vstupu a) levé nádrže b) pravé nádrže

Po změření odezev na skokovou změnu vstupu jsem mohl využít nabyté znalosti z předmětu modelování a simulace. Hodnoty jsem načtl do software Matlab, zajistil jsem nulové počáteční podmínky a vytvořil jsem pomocí uvedeného skriptu v příkladu 3 a příkladu 4 model soustavy. Jako výchozí model pro aproximaci jsem použil soustavu druhého řádu se dvěma časovými konstantami.

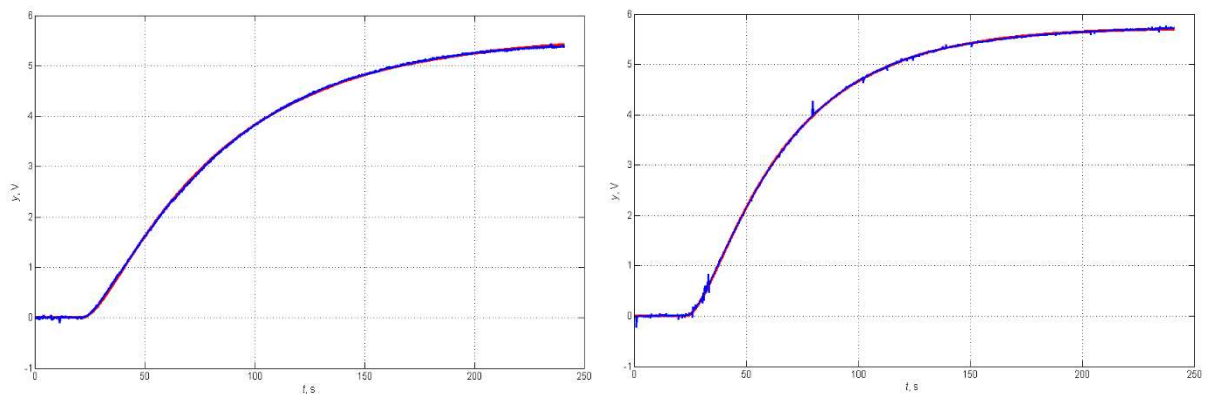
Příklad 3 – Skript pro hledání přenosu z dat

```
X=fminsearch('DruhyRad',[1 1 1],[],t,u1,y1);
Z=X(1);
T1=X(2);
T2=X(3);
S=tf(Z,[T1*T2 T1+T2 1]);
```

Příklad 4 – Funkce s přenosem druhého řádu

```
function K = DruhyRad(X,t,u,y)
Z=X(1);
T1=X(2);
T2=X(3);
S=tf(Z,[T1*T2 T1+T2 1]);
yp=lsim(S,u,t,'zoh');
K=(yp-y)'*(yp-y);
```

Model byl simulován na stejný vstupní signál jako reálná soustava, výsledek simulace je zobrazen na obr. 2.41 naměřené hodnoty velmi dobře kopírují identifikovaný model (červenou křivkou).



Obr. 2.41 – Simulační model vykreslený spolu s reálným průběhem a) levá nádrž b) pravá nádrž

Výsledné přenosy modelů soustav

$$F(s) = \frac{2,799}{479,6s^2 + 69,18s + 1}, \quad (2.13)$$

$$F(s) = \frac{2,87}{299,8s^2 + 48,56s + 1}. \quad (2.14)$$

2.5.4 Nastavení PID regulátoru pomocí metody T-suma

Po zjištění přenosů soustav a jejich přechodových charakteristik jsem přešel k nastavení PID regulátorů pomocí metody T-suma. Konkrétněji jsem využil druhou zmíněnou metodu v teoretické části, a to grafickou metodu zjištění T-Suma. Nalezením na přechodové

charakteristice takové časové hodnoty, kdy se oblasti S1 a S2 rovnají. Tento bod lze matematicky odvodit jako

$$T_{\Sigma} = T - \frac{\int_0^{5T} y(t)}{Z} \quad (2.15)$$

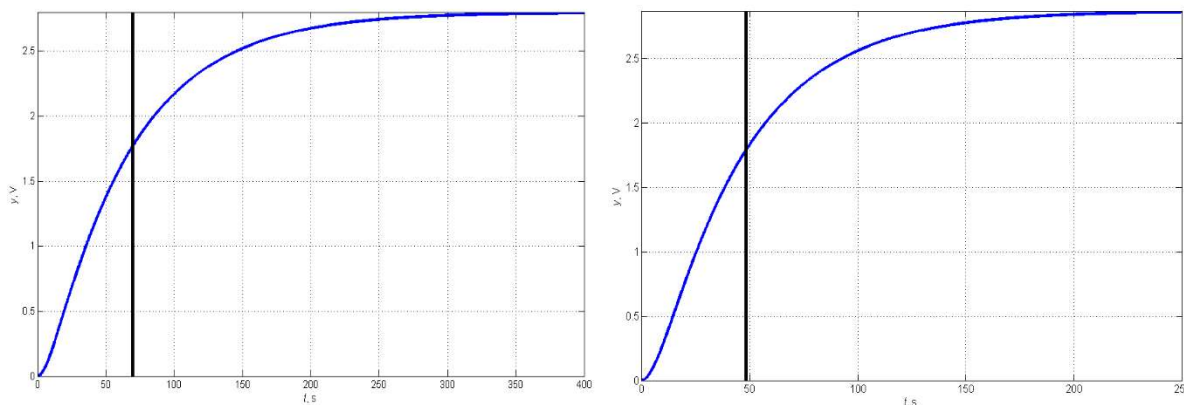
kde T – doba ustálení,
 Z – zesílení soustavy

V software Matlab lze využít následující příklad 5 pro přesné určení T-suma.

Příklad 5 – Skript pro hledání T-Suma
<pre>Z=5; T=20; Ts=0.1; S=tf(Z, [T 1]) [y, t]=step(S, 0:Ts:5*T); Tsuma=5*T-sum(y)*Ts/Z;</pre>

Samozřejmě, že uvedený skript není univerzální a musí se upravit pro určitou soustavu.

Skript jsem upravil a použil na přechodové charakteristiky modelů nádrží. Výsledek je zobrazen na obr. 2.42 (černá svislá čára je znázornění T-sumy).



Obr. 2.42 – Vypočítání T-suma a) levá nádrž b) pravá nádrž

Pro levou nádrž se T-suma a zesílení rovnají

$$T_{\Sigma} = 69,0834, \quad Z = 2,799.$$

A pro pravou

$$T_{\Sigma} = 48,3662, \quad Z = 2,87.$$

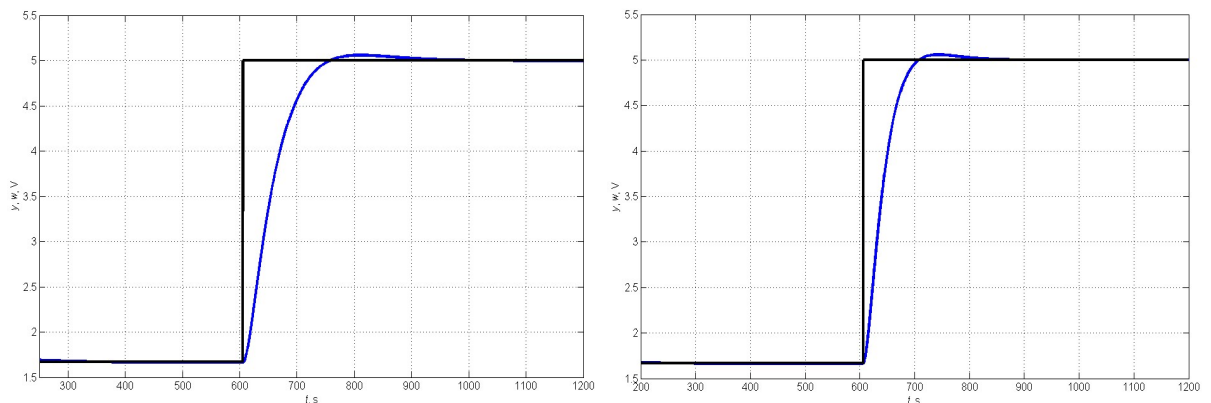
Tyto hodnoty jsem dosadil do tabulky uvedené v teoretické části (tab. 1.2) a vypočítal hodnoty P, PI a PID regulátorů.

Tab. 2.7 – Vypočítané hodnoty P, PI, PID regulátorů pomocí T-suma

Regulátor	Levá nádrž			Pravá nádrž		
	r_0	T_i	T_d	r_0	T_i	T_d
P	0,35727	-	-	0,348432	-	-
PI normální	0,178635	34,5417	-	0,174216	24,1831	-
PI rychlý	0,35727	48,35838	-	0,348432	33,85634	-
PID normální	0,35727	45,59504	11,74418	0,348432	31,92169	8,222254
PID rychlý	0,714541	55,26672	13,12585	0,696864	38,69296	9,189578

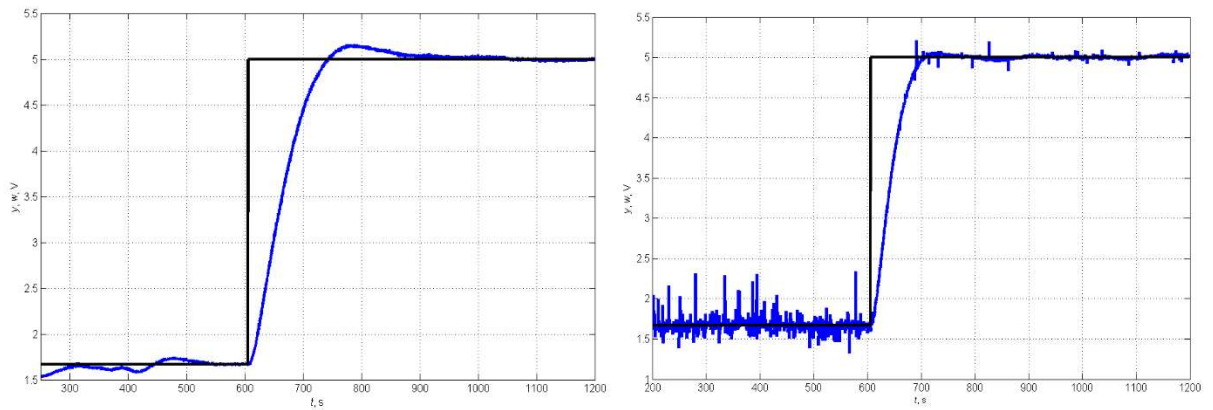
Test nastavení PI regulátoru

Metoda T-suma vykazuje nejlepší výsledky regulačního pochodu při nastavení PI regulátorů, proto jsem si tento typ vybral pro regulaci hydraulicko-pneumatické soustavy. Nejdříve jsem si vytvořil jednoduché schéma uzavřeného regulačního obvodu v software Simulink a otestoval nastavení PI rychlých regulátorů na obou modelech nádrží. Regulační pochod jsem začal s žádanou hodnotou 5 cm (1,7 V) a v čase 606 vteřin jsem nastavil žádanou na 15 cm (5 V). Tyto žádané hodnoty se nachází v pracovní oblasti, která byla identifikována v předchozí kapitole. Regulační pochody modelů nádrží jsou na obr. 2.43.



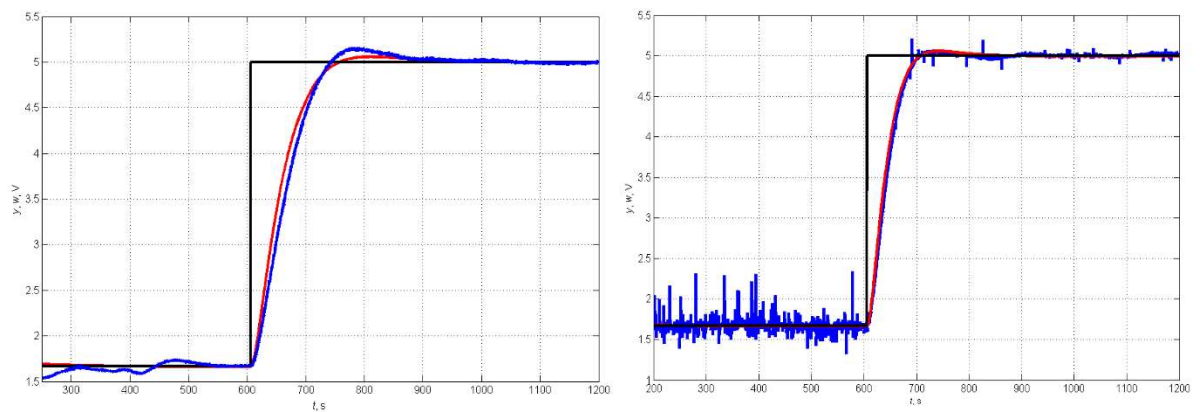
Obr. 2.43 – Regulační pochod modelů a) levá nádrž b) pravá nádrž

Nastavení na modelech funguje obstojně, proto jsem přistoupil k testu na reálném systému hydraulicko-pneumatické soustavy. Hodnoty PI regulátoru jsem nastavil stejně jako pro regulátor modelů a také jsem postupoval stejným způsobem s regulačním pochodem. Hodnoty jsem nastavil přes uživatelský panel. Výsledek měření byl zaznamenán pomocí ukládání hodnot PLC. Tyto hodnoty jsem zobrazil v prostředí Matlab, výsledek měření je na obr. 2.44.



Obr. 2.44 – Regulační pochod soustavy a) levá nádrž b) pravá nádrž

Tento výsledek měření jsem nakonec ještě zobrazil v jednom grafu spolu s regulačním pochodem modelu, abych si ověřil přesnost identifikace a nastavení, obr. 2.45



Obr. 2.45 – Regulační pochod vykreslený s regulačním pochodem modelu a) levá nádrž b) pravá nádrž

3 ZÁVĚR

Cílem práce bylo vytvořit aplikaci pro ovládání hydraulicko-pneumatické soustavy. Teoretická část práce obecně popisuje programovatelné automaty, jejich hardware a software a jak se programují. Také jsou zde shrnuty potřebné základy řízení systémů a základní typy regulátoru a jejich nastavení.

V praktické části se čtenář mohl blíže seznámit s programovatelným automatem Modicon M241 od firmy Schneider Electric. Jsou zde popsány jeho základní moduly, které jsou v aplikaci použity. Čtenář je v praktické části také seznámen s vlastní hydraulicko-pneumatickou soustavou. Dále je zde popsán detailní návod pro vytvoření základní aplikace v prostředí SoMachine.

V další kapitole je vytvořena vlastní aplikace pro řízení hydraulicko-pneumatické soustavy. Byly vytvořeny dva základní režimy ovládání soustavy, a to režim automatický, kde uživatel nastaví parametry PID regulátoru a žádanou hodnotu a režim manuální, kde uživatel ovládá samotná čerpadla. V automatickém režimu byla použita PID knihovna od sdružení oscat.de, která řeší mnoho věcí za programátora. Je zde vysvětlena funkčnost programu a možnosti nastavení a upravení. V neposlední řadě se práce věnuje uživatelskému panelu a jeho obsluze. Prostředí pro display bylo navrženo tak, aby práce s ním byla co nejsnazší. Nakonec praktické části byla soustava identifikována a pomocí software Matlab byl vytvořen její model a z přechodové charakteristiky modelu bylo provedeno nastavení PID regulátoru metodou T-suma. Závěrem je toto nastavení otestováno na soustavě.

Soustava byla řízena PI regulátorem v nastavení pro rychlý regulační pochod, protože metoda T-suma vykazuje nejlepší kvalitu regulace právě s tímto nastavením. Regulační pochod byl proveden v pracovní oblasti, ve které byla soustava identifikována, a proto je výsledek regulace je velmi podobný simulovaným průběhům. Na obr. 2.45 jsou zobrazeny oba regulační pochody v jednom grafu. Modrá křivka výstupu reálné soustavy se velmi přibližuje průběhu červené křivky výstupu modelu. Tímto byla ověřena správnost funkce číslicového PID regulátoru použitého v PLC. Obě dvě nádrže mají dobu ustálení do 5 minut, proto je tato soustava vhodná pro experimenty při výuce.

Práce se nevěnuje konstrukčnímu nedostatku soustavy, a to, že zde není žádná ochrana proti přetečení. Proto bych jako první možné rozšíření aplikace navrhoval přidat ochranný režim, aby ochrana proti přetečení byla řešena alespoň programově. Může být vytvořena další obrazovka pro uživatelský panel, kde si uživatel nastaví vlastní omezující podmínky. Také by mohlo být vylepšeno webové rozhraní, aby byla zajištěna koexistence s uživatelským panelem.

Toto rozhraní je velmi pohodlné, protože uživatel může ovládat soustavu, nebo PLC pomocí tabletu, nebo mobilu.

Tato bakalářská práce může sloužit jako návod pro vytvoření aplikace v software SoMachine od vytvoření projektu až po nahrání programu do PLC. Může také sloužit jako laboratorní úloha pro budoucí studenty, ve které si mohou otestovat různé nastavení PID regulátoru a změřit regulační pochod PID regulátoru při regulaci hladiny hydraulicko-pneumatické soustavy.

LITERATURA

- CVEJN, Jan. *Automatizace I: Kap. 10 – Automatická regulace*. Přednáška. Pardubice. 2015 [cit. 2017-03-01]. Osobní komunikace.
- CVEJN, Jan. *Automatizace I: Kap. 13 – Praktické metody nastavení PID regulátorů*. Přednáška. Pardubice. 2015 [cit. 2017-03-01]. Osobní komunikace.
- ČSN EN 61131-1. Programovatelné řídicí jednotky: Část 1: Všeobecné informace. 2004.
- CHLEBNÝ, Jan. *Automatizace a automatizační technika. 4., aktualiz. vyd.* Brno: Computer Press, 2009. ISBN 978-80-251-2523-6.
- HONC, Daniel a František DUŠEK. Novel multivariable laboratory plant. In: *26th European Conference on Modelling and Simulation*, Koblenz, Germany, May 29 – June 1 2012, ECMS – European Council for Modelling and Simulation, 2012, pp. 468-473, ISBN 978-0-9564944-4-3
- KUHN, Udo. Eine praxisnahe Einstellregel für PID-Regler: Die T-Summen-Regel. *Automatisierungstechnische Praxis*. 1995, (5), 10-16.
- MACHÁČEK, Jiří, Daniel HONC a František DUŠEK. Výukový laboratorní model hydraulicko-pneumatické soustavy. *Automa*. 2005, (8-9), 108-109. ISSN 1210-9592.
- SCHNEIDER ELECTRIC. *Modicon M241 Logic Controller: Hardware Guide* [online]. 2014 [cit. 2017-03-01]. Dostupné z: http://www.eschneider.pl/download/04%20Automatyka%20przemyslowa/PLC%20sterowniki/M241_PLC_Modicon-Hardware_Guide_2014ENG.pdf
- SCHNEIDER ELECTRIC. *Modicon TM3: Analog I/O Modules Hardware Guide* [online]. 2014 [cit. 2017-03-01]. Dostupné z: http://www.electrocentr.com.ua/files/documentation/SE/plc/m221-241-251/ug/Modicon-TM3_Analog_IO_Modules_Hardware_guide.pdf
- SCHNEIDER ELECTRIC. *Getting Started With SoMachine: Self Study Manual* [online]. 2014 [cit. 2017-03-01]. Dostupné z: http://www2.schneider-electric.com/resources/sites/SCHNEIDER_ELECTRIC/content/live/FAQS/288000/FA288633/en_US/Training%20Manual%20EN.pdf
- SCHNEIDER ELECTRIC. *Magelis HMI STU 655/855: User Manual* [online]. 2012 [cit. 2017-03-01]. Dostupné z: <http://www.farnell.com/datasheets/1975768.pdf>

- SCHNEIDER ELECTRIC. *Modicon TM4: Expansion Modules Hardware Guide* [online]. 2014 [cit. 2017-03-01]. Dostupné z: <http://docs-europe.electrocomponents.com/webdocs/12ce/0900766b812ced60.pdf>
- ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. Praha: BEN – technická literatura, 1999. ISBN 80-86056-58-9.
- TECO. *PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT*. [online]. 21. Kolín, 2014 [cit. 2017-03-01]. Dostupné z: http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00410_01_General_Foxtrot.pdf
- VOJÁČEK, Antonín. Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys). In: *Automatizace. HW.cz | Elektronika v automatizaci* [online]. 2011 [cit. 2017-01-18]. Dostupné z: <http://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-61131-codesys>

PŘÍLOHY

A – CD

Příloha k bakalářské práci

Řízení hydraulicko-pneumatické soustavy pomocí PLC Modicon

Jiří Vinduška

CD

Obsah

- 1 Text bakalářské práce ve formátu PDF
- 2 Aplikace pro PLC a uživatelské panel
- 3 Dokumentace