

VŠB - Technická univerzita Ostrava
Fakulta strojní
Katedra automatizační techniky a řízení

Využití PLC pro řízení laboratorního modelu s možností vizualizace

The Use of PLC for Laboratory Model Control with Visualization Options

Autor práce: Bc. Radim Hýl
Vedoucí práce: Ing. Jaromír Škuta, PhD.

Ostrava 2013

Zadání diplomové práce

Student: **Bc. Radim Hýl**
Studijní program: N2301 Strojní inženýrství
Studijní obor: 3902T004 Automatické řízení a inženýrská informatika
Téma: **Využití PLC pro řízení laboratorního modelu s možností vizualizace**
The Use of PLC for Laboratory Model Control with Visualization Options

Zásady pro vypracování:

1. Charakterizujte programové prostředí Control Web 6 (základní stavební prvky systému, modularita systému, časování).
2. Popište programovatelné automaty v laboratořích katedry včetně prostředí pro jejich programování. Zaměřte se na možnosti komunikace mezi nimi (LAN, ILAN, ...).
3. Navrhněte úlohu využívající vybraná PLC a umožňující přenos technologických dat do SCADA/MMI systému.
4. Navrženou úlohu realizujte a vytvořte podrobnou technickou dokumentaci k této úloze.
5. Zhodnoťte dosažené výsledky a navrhněte směr dalšího řešení.

Seznam doporučené odborné literatury:

BALÁTĚ, J. *Automatické řízení*. Praha: Nakladatelství BEN, 2003, 654 s. ISBN 80-7300-020-2.
BÍLÝ, R., CAGAŠ, P. & AJ. *Control Web 2000*. Průvodce systémem pro tvorbu a nasazení aplikací reálného času. 1. vydání. Praha: Computer Press, 1999. 382 s. ISBN 80-7226-258-0.
JANEČEK, J. 1993. *Distribuované systémy*. Praha : Vydavatelství ČVUT, 1993.
NACHTIGAL CH. L. *Instrumentation and Control - Fundamentals and Applications*. New York : John Wiley & Sons, Inc. 1993.
VAEPROSYS - *AUTOMATIZACE - PLC AC500* - Dokumentace ke stažení [online]. VAE ProSys s.r.o. 2010. [cit. 11.10.2012]. Dostupné z: <<http://vaeprosys.cz/list.asp?p=133&k=32&mk=9&ms=32>>
VLACH, J. *Řízení a vizualizace technologických procesů*. PRAHA: BEN, 1999, 160 S. ISBN 80-86056-66-X.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jaromír Škuta, Ph.D.**

Datum zadání: 14.12.2012

Datum odevzdání: 20.05.2013



prof. Ing. Jiří Tůma, CSc.
vedoucí katedry



doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem celou bakalářskou práci včetně příloh vypracoval samostatně pod vedením vedoucího bakalářské práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě

.....

Radim Hýl

Prohlašuji, že

- byl jsem seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a §60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo výdělečně ke své vnitřní potřebě bakalářskou práci užít (§35 odst. 3).
- souhlasím s tím, že jeden výtisk bakalářské práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu §12 odst. 4 autorského zákona.
- bylo sjednáno, že užití své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě

.....

Bc. Radim Hýl

Bc. Hýl Radim

Jičínská 788,

742 58 Příbor

ANOTACE DIPLOMOVÉ PRÁCE

HÝL, R. *Využití PLC pro řízení laboratorního modelu s možností vizualizace : diplomová práce.* Ostrava: VŠB - Technická univerzita Ostrava, Fakulta Strojní, Katedra automatizační techniky a řízení, 2013, Vedoucí práce: Škuta, J.

Tato práce je směřována do oblasti využití výpočetní techniky v měření a regulaci, zvláště pak do oblasti řízení laboratorních modelů pomocí programovatelných logických automatů využívaných na katedře automatizační techniky a řízení. Je rozdělena do tří dílčí částí. První z nich popisuje používaný vizualizační software Control Web 6.1. Druhá pak zahrnuje popis hardwaru programovatelných PLC jednotek a popis prostředí programu CoDeSys, který slouží pro jejich programování. Ke konci práce jsou navrženy a také realizovány dvě laboratorní úlohy řízené programovatelnými automaty. Druhá laboratorní úloha představuje také mnohorozměrnou soustavu a je na ni realizováno prediktivní řízení s interním modelem.

ANNOTATION OF DIPLOMA THESIS

HÝL, R. *The The Use of PLC for Laboratory Model Control with Visualization Options: Diploma Thesis.* Ostrava: VŠB - Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2013, Thesis head: Škuta, J.

Diploma thesis is directed to the use of computer technology in measurement and control, especially to the communication of PLC systems with laboratory models used at the department of control systems and instrumentation. It is divided into three parts. The first describes visualization software Control Web 6.1. The second involves of hardware equipment of PLC and software CoDeSys for programming applications . At the end of the work are designed changes in two laboratory stands and realized control by programmable logic controllers. Second laboratory model presents MIMO system and there is also realized Model Predictive Control.

OBSAH

ANOTACE DIPLOMOVÉ PRÁCE.....	5
ANNOTATION OF DIPLOMA THESIS.....	5
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	8
1 ÚVOD.....	10
2 VIZUALIZAČNÍ A ŘÍDICÍ SOFTWARE CONTROL WEB 6.1.....	11
2.1 Vývojové prostředí.....	11
2.2 Grafický editor.....	12
2.3 Modularita aplikací	15
2.4 Možnosti komunikace softwaru s okolím	15
3 PRODUKTOVÁ ŘADA PLC SYSTÉMŮ AC500 OD FIRMY ABB	17
3.1 Procesorová jednotka PM 571	18
3.2 Modul binárních vstupů/výstupů DC523.....	19
3.3 Modul analogových vstupů/výstupů AX521	20
4 VÝVOJOVÉ PROSTŘEDÍ PRO PLC - CODESYS.....	21
4.1 Tvorba aplikace	21
4.2 Grafické prostředí.....	22
5 KOMUNIKACE S PLC.....	27
6 LABORATORNÍ ÚLOHA ŘÍZENÍ ELEKTROMOTORU	28
6.1 Popis jednotlivých prvků laboratorního modelu	28
6.2 Provedené úpravy v laboratorní úloze.....	30
6.2.1 Vizualizace úlohy s elektromotorem	31
6.2.2 Vytvoření unifikačního modulu.....	32
7 LABORATORNÍ ÚLOHA ŘÍZENÍ TEPLOVZDUŠNÉHO MODELU	37
7.1 Popis jednotlivých prvků laboratorního modelu	38
7.2 Provedené úpravy v laboratorní úloze.....	40
7.2.1 Vizualizace úlohy s teplovzdušným modelem.....	40
7.2.2 Propojení unifikačního modulu s PLC.....	41
8 APLIKACE PREDIKTIVNÍHO ŘÍZENÍ	42
8.1 Matematický popis aplikace.....	42
8.2 Realizace algoritmu pro PLC Bachmann MPC200	46
8.3 Vytvoření vizualizace v programu SolutionCenter.....	47
8.4 Aplikace algoritmů v MATLABu.....	50
9 VÝSLEDNÉ PROPOJENÍ ÚLOH.....	54

10 ZÁVĚR.....	55
SEZNAM POUŽITÉ LITERATURY.....	57
SEZNAM PŘÍLOH.....	59

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ASCII	American Standart Code for Information Interchange - zkratka amerického standardního kódu pro výměnu informací
ATR	katedra aplikované informatiky a řízení
Cannon	typ konektoru
COM	Component Object Model - softwarová komponenta
CPU	Central Processing Unit - centrální procesorová jednotka
CTRL	označení konektoru
DDE	Dynamic Data Exchange – dynamická výměna dat – protokol MS Windows pro výměnu dat mezi aplikacemi
DMF	mapovací soubor systému Control Web 6
EPROM	Erasable Programable Read Only Memory- Semipermanentní typ paměti
FBD	Function Block Diagram – funkční blokové schéma
GND	Ground, označení vodiče pro společnou zem
Gs	přenos regulované soustavy
HMI	Human Machine Interface – rozhraní člověk-stroj
I	proud [A]
IEC	International Electrotechnical Commission -mezinárodní norma věnující se PLC, jejich funkci, provedení hardwaru a komunikací i způsobům jejich programování
IL	Instruction List – instrukční list
I/O	Innput/Output - vstupně/výstupní
IP	Internet Protocol – standardní síťový protokol
LD	Ladder Diagram – kontaktní schéma
LED	Light-Emitting Diode – dioda emitující světlo
NTC	negistor
OPC	OLE for Process Control – rozhraní pro komunikaci mezi rozdílnými aplikacemi, zařízeními atd.
P	výkon [W]
PAR	parametrický soubor systému Control Web 6
PC	Personal Computer – osobní počítač
PID	proporcionálně integračně derivační analogový regulátor
PLC	Programmable Logic Controller – programovatelný logický automat
PTC	pozistor
RAM	Random Access Memory - paměť s libovolným přístupem

SCADA	Supervisory Control and Data Acquisition – supervizní řízení a sběr dat
SD	Secure Digital - paměťová karta
SFC	Sequential Function Chart – sekvenční funkční diagram
ST	Structured Text – strukturovaný text
TCP/IP	Transmission Internet Protocol - sada síťových protokolů
TVM	Teplovzdušný model
u	akční veličina
USB	Universal Serial BUS - univerzální sériová sběrnice
y	regulovaná, výstupní veličina

1 ÚVOD

Lidé jsou vynalézaví, a proto se snaží ulehčit, případně zautomatizovat, neustále se opakující výpočetní operace. Z tohoto důvodu se počátkem 40. let 20. století začínají objevovat první „počítací stroje“. Následující léta byla ve znamení vývoje velkých sálových počítačů. S přibývajícím časem se počítačovým konstruktérům dostaly do rukou polovodičové součástky, které umožnily zkonstruovat počítač takové velikosti, aby se dal snadno umístit na stůl a tím zpřístupnil práci běžným uživatelům. Poslední léta přinesla obrovský pokrok na poli výpočetní techniky. Došlo nejen k několikanásobnému růstu výkonů dnešních počítačů, ale i k jejich rozšíření snad do všech oborů lidské činnosti. Již na konci 70. let dvacátého století se objevily první pokusy využít osobní počítač pro měření a ovládání fyzikálních procesů. V oboru řízení, měření a regulace, což jsou tradiční inženýrské disciplíny, dnes tato technika nachází široké uplatnění, a to nejen z důvodů klesajících cen, ale hlavně pro její snadnou přizpůsobitelnost při řešení konkrétních technických problémů, jednoduchou rozšiřitelnost, snadné ovládání a možnost poskytnout koncovému uživateli kvalitní datové výstupy s možností jejich archivace a následného zpracování. V současné době jsou v průmyslu nejrozšířenějšími výpočetními jednotkami programovatelné logické automaty (PLC). Nejčastější nasazení PLC systémů je v oblasti řízení průmyslových procesů. Jejich výhodou je především přímá práce s fyzikálními procesy a monitorování dějů v reálném čase. Na současných školních pracovištích proto nechybí fyzikální modely reálných procesů řízené PLC systémy, kterými se zabývá tato práce. Pokroky v této oblasti jsou však bezpodmínečně vázány na tvorbu odpovídajícího programového vybavení. Moderní uživatel dnes klade důraz nejen na jednoduchost ovládání, ale rovněž na kvalitní a graficky dokonalé rozhraní. K tomuto účelu nám slouží řídicí nebo vizualizační systémy, což jsou programy, které prostřednictvím konkrétního zařízení (portu, speciální karty, PLC) mohou měřit, kontrolovat nebo řídit určitý technologický proces. Jedná se například o řízení stroje nebo výrobní linky pomocí PC. Jeden jediný počítač dnes díky velkému výkonu a možnostem softwaru dokáže řídit i poměrně komplikovaný výrobní proces.

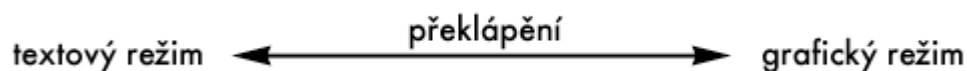
2 VIZUALIZAČNÍ A ŘÍDICÍ SOFTWARE

CONTROL WEB 6.1

Tento software vyvinula společnost Moravské přístroje. Jedná se o víceúčelový virtuální prostředek spadající do kategorie programů SCADA/HMI a je použitelný takřka ve všech průmyslových odvětvích. Můžeme vytvářet nejrůznější virtuální ovládací prvky, měřicí prvky a zobrazovací prvky, které jsou schopny v reálném čase komunikovat s fyzickými zařízeními. Tento software nám nabízí adaptivní prostředí, které je možno snadno měnit podle potřeb úlohy. Vizualizační program poskytuje možnosti jak řízení, tak zobrazování procesu. [BÍLÝ, 1999]

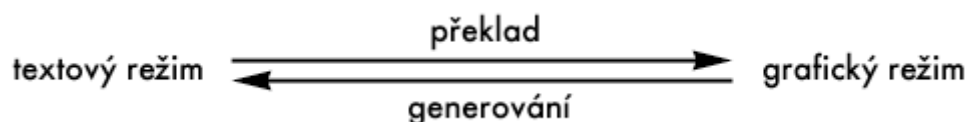
2.1 Vývojové prostředí

Tvorba aplikace je většinou podmíněna jasným stanovením požadavků na funkčnost, vlastnosti a účel této aplikace. Po stanovení těchto požadavků, se však často vyskytne otázka, jakým způsobem tuto aplikaci vytvořit. Systém využívá při vývoji aplikací tzv. *Dvojcestné programování*. Můžeme ho vytvářet ve dvou různých tvarech, mezi kterými lze libovolně překlápět (obr. 1). V podstatě jde o dvě vývojová prostředí s různými programovacími příkazy, ovšem změna v jednom se hned projeví i v prostředí druhém. Záleží jen na programátorovi, které prostředí si zvolí. [Manuál Control Web 6,2013]



Obr. 1 Překlápění v Control Web 6.1

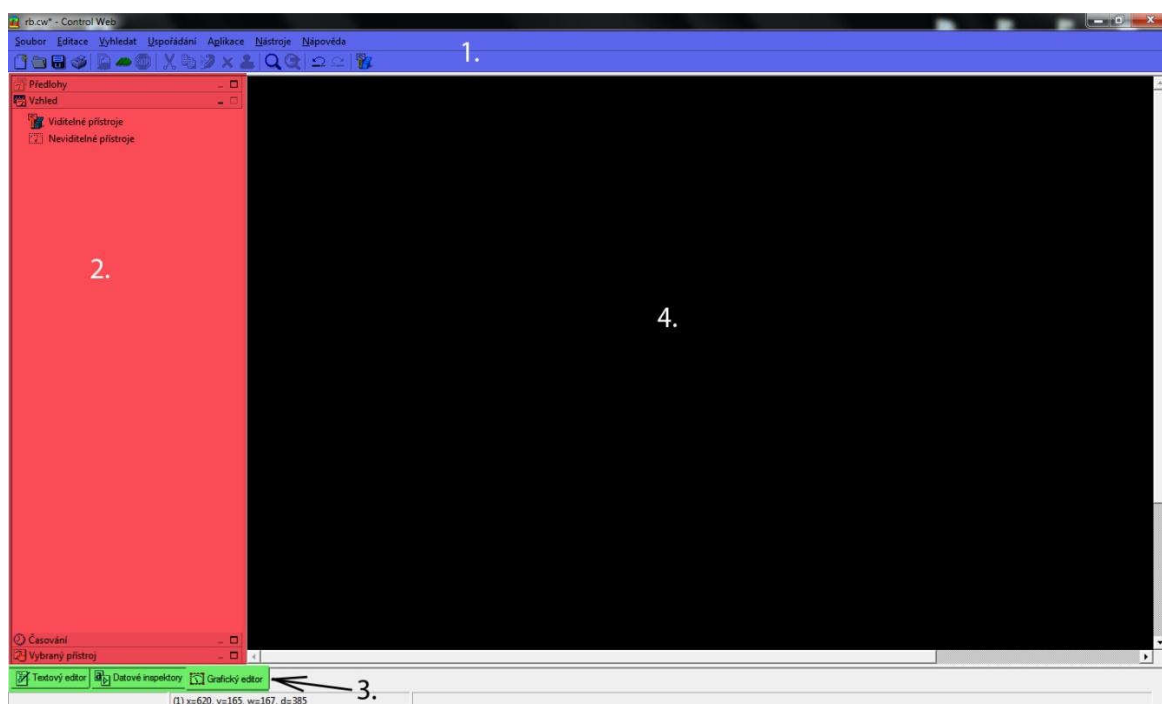
Zdálo by se, že textový a grafický editor jsou si rovny, jsou však zde pevně dány priority. Počítač používá pouze textovou formu programu, proto je vždy program ukládán do této formy a grafickou formu překládá (obr. 2). Z grafické formy do textové je pak program generován pomocí předdefinovaných algoritmů zabudovaných přímo v programu. Aplikace vytvořená v grafickém editoru je na první pohled snadno zapamatovatelná i pro laika v tomto oboru. Tato možnost překlápění je výhodná zejména pro budoucí změnu stávajícího programu. [Manuál Control Web 6, 2013]



Obr. 2 Překlad a generování v Control Web 6.1

2.2 Grafický editor

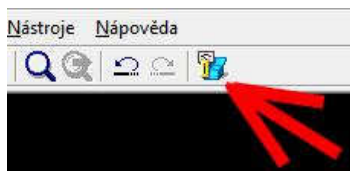
Grafický editor se skládá z několika logicky rozdělených částí. Místo, kde deklarujeme proměnné, přidáváme a nastavujeme ovladače pro komunikaci s vnějšími zařízeními, nastavujeme práva a přihlašování uživatelů obsluhujících aplikaci, vytváříme globální časování a mnoho dalšího se nazývá *Datové inspektory*. Prostor pro vizuální návrh a řízení běhu aplikace pokrývá hlavní část vývojového prostředí, což je vlastní *Grafický editor*. Prostor pro psaní aplikace v textové formě představuje záložka *Textový editor*. [Manuál Control Web 6, 2013]. Hlavní okno grafického editoru je zobrazeno na obrázku 3.



1. Panely nástrojů.
2. Strom možností.
3. Přepínání mezi proměnnými a grafickým i textovým editorem.
4. Hlavní pracovní prostor.

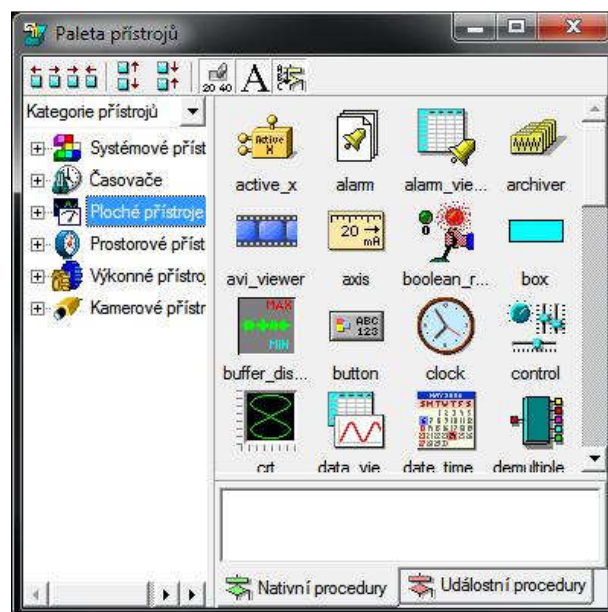
Obr. 3 Grafické prostředí Control Web 6.1

V levé dolní části najdeme přepínání mezi textovým a grafickým editorem (3.). Při volbě grafického editoru, máme na levé straně strom (2.) s jednotlivými virtuálními přístroji, s jejich časováním, předvolbami atd. Pokud pracujeme v textovém editoru, máme celé okno prázdné a vepisujeme do něj jednotlivé algoritmy a příkazy. V horní části hlavního okna máme panely nástrojů (1.) se standardními příkazy týkající se aplikace, ať už jejího ukládání, načítání, nebo chodu, ale také ikonu pro zobrazení okna "Paleta přístrojů" (obr. 4). [Manuál Control Web 6, 2013]



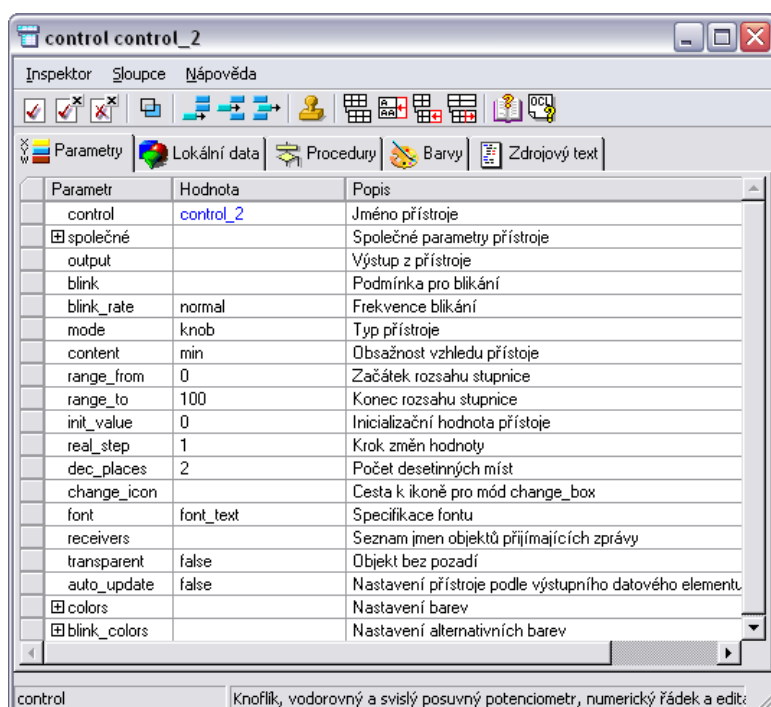
Obr. 4 Ikona pro spuštění Palety přístrojů

Paleta přístrojů (obr. 5) obsahuje všechny dostupné virtuální přístroje a grafické prvky, které je do grafického editoru možné přímo přetahovat myší a jsou rozčleněné podle své funkce, druhu časování a použití. Přístroje umísťujeme pro přehlednost do panelů, které připomínají okna v klasických Windows, a tímto je i sdružujeme v další celky, kterým je možno definovat vlastnosti hromadě a dokonce je i hromadně ovládat za běhu programu. [Manuál Control Web 6, 2013]



Obr. 5 Paleta přístrojů

Po umístění přístroje nebo ovládacího prvku na pracovní plochu můžeme s ním dále pracovat, nastavovat velikost, umístění, barvu, vlastnosti chování a mnoho dalšího. Toto vše nastavujeme v "Inspektoru přístroje" (obr. 6), který je další velmi důležitou částí grafického prostředí. Nejenom že slouží k nastavování vlastností jednotlivých přístrojů, ale také k přidávání procedur a hlavně k navázání přístroje na proměnnou. Dále zde najdeme záložku vnitřních proměnných, v nichž jsou deklarovány pomocné proměnné, které využívá pouze tento přístroj, a zdrojového textu, kde najdeme zdrojový text pouze tohoto přístroje. [BÍLÝ, 1999]



Obr.6 Inspektor přístroje

V programu Control Web 6.1 máme k dispozici tyto datové typy:

Tab. 1 Datové typy v Control Web 6.1

Typ	Popis
BOOLEAN	Jedná se o dvouhodnotovou proměnnou (Ano-Ne, Pravda-Nepravda atd.) zabírající v paměti pouze jeden bit. Používá se pro logické funkce.
STRING	Proměnná, do které lze ukládat textové řetězce.
REAL	Můžeme přiřadit libovolné desetinné, nebo celé číslo v rozsahu od $\pm 2.3 \times 10^{-308}$ do $\pm 1.7 \times 10^{+308}$. Hodnota tohoto typu obsadí 4 bajty paměti a jedná se o číslo s plovoucí desetinnou čárkou.
SHORTCARD	Pro nezáporná čísla od 0 do 255, která zabírají v paměti 1 bajt.
CARDINAL	Podobná proměnná jako Shortcard, pouze s větším rozsahem od 0 do 65535. Samozřejmě zabírá více paměti (2 bajty).
LONGCARD	Rozšířená proměnná typu Cardinal v rozsahu od 0 do 4294967295 a v paměti si alokuje 4 bajty.
SHORTINT	Má za základ typ Integer, jen je jeho rozsah omezen na -32768 až 32767 a zabírá pouze 16 bitů.
INTEGER	Celočíselná proměnná, do níž jsme schopni zapsat číslo v rozsahu od -32768 do 32767 a v paměti nám bude zabírat 2 bajty.
LONGINT	Opět celočíselná proměnná, rozšířená na rozsah od -2147483648 do 2147483647 a místem v paměti velikosti 4 bajty.
SHORTREAL	Pro reálná čísla v rozsahu od $\pm 1.2 \times 10^{-38}$ do $\pm 3.4 \times 10^{+38}$. V paměti si alokuje 4 bajty.
DATA	Jedná se o obecný datový typ, jehož vnitřní struktura se může měnit. Tato změna je definována v ovladačích, nebo přímo v přístroji.

2.3 Modularita aplikací

System Control Web umožňuje provozovat modulární, distribuované a synchronizované aplikace.

- *Modulární aplikace* je aplikace skládající se z více modulů. System Control Web modul považuje za jeden aplikační soubor, tudíž nejjednodušší aplikace je jednomodulární. Většina aplikací v systému Control Web je jednomodulární. [Manuál Control Web 6, 2013]
- *Distribuovaná aplikace* se skládá z více autonomních uzlů (počítačů) a aplikací, vzájemně propojených, komunikujících a jevících se (zvnějšku) jako jednotný integrovaný systém. Distribuovaná aplikace v systému Control Web vždy běží v módu klient-server. [Manual Control Wev 6, 2013]
- *Synchronizované aplikace* se používají především pro zvýšení spolehlivosti rozsáhlých řídicích nebo vizualizačních systémů. Stejně (nebo alespoň z části stejné) aplikace běží na dvou nebo více počítačích. Část dat těchto aplikací je synchronizovaná, to znamená, že Control Web zajistí, aby byly na všech počítačích hodnoty těchto dat v jeden okamžik stejné. Všechny aplikace udržují spojení a informace o dostupnosti ostatních aplikací na okolních počítačích. V jeden okamžik je vždy jedna aplikace aktivní. Pokud tato aplikace selže (například z důvodu výpadku napájení) vyberou ostatní aplikace mezi sebou jinou, která se stane aktivní. Tím je zajištěno zálohování a pokud jedna selže, druhá bude vždy zajišťovat řízení nebo vizualizaci technologie. [Manual Control Wev 6, 2013]

2.4 Možnosti komunikace softwaru s okolím

Pro komunikaci s okolím využíváme ovladačů a dalších komponent. Díky těmto ovladačům můžeme i simulovat různé přístroje a jejich chování v praxi. Máme na výběr s velkého množství ovladačů, jež lze rozdělit na dvě hlavní kategorie a to na "Virtuální" (simulační) a "Fyzické". Virtuální ovladače se neváží na žádné skutečné kanály, ale používají se za účelem ladění aplikace. Umějí simulovat různé vstupní signály, nebo posílají signály podle definovaných parametrů. Pro nasazení aplikace v praxi se používají pouze ovladače fyzické. Mezi tyto patří zejména tyto ovladače: ASCII, TCP/IP, ADVBUF, OPC, DDE, ADVPCL atd. Ovladač OPC je popsán blíže, jelikož je použit při realizaci laboratorních úloh. [Manuál Control Web 6, 2013]

OPC Ovladač je standardizované komunikační rozhraní, neziskový standard, pomocí kterého můžeme spojit například průmyslovou automatizaci přímo s informačním systémem. Norma OPC používá ke komunikaci technologii COM, která umožňuje posílání jak jednotlivých hodnot, tak i celé pole hodnot. To jestli poslaná data jsou hodnoty, nebo pole záleží pouze na OPC serveru, jaká data zpřístupní [ŠTĚPÁN, 2011].

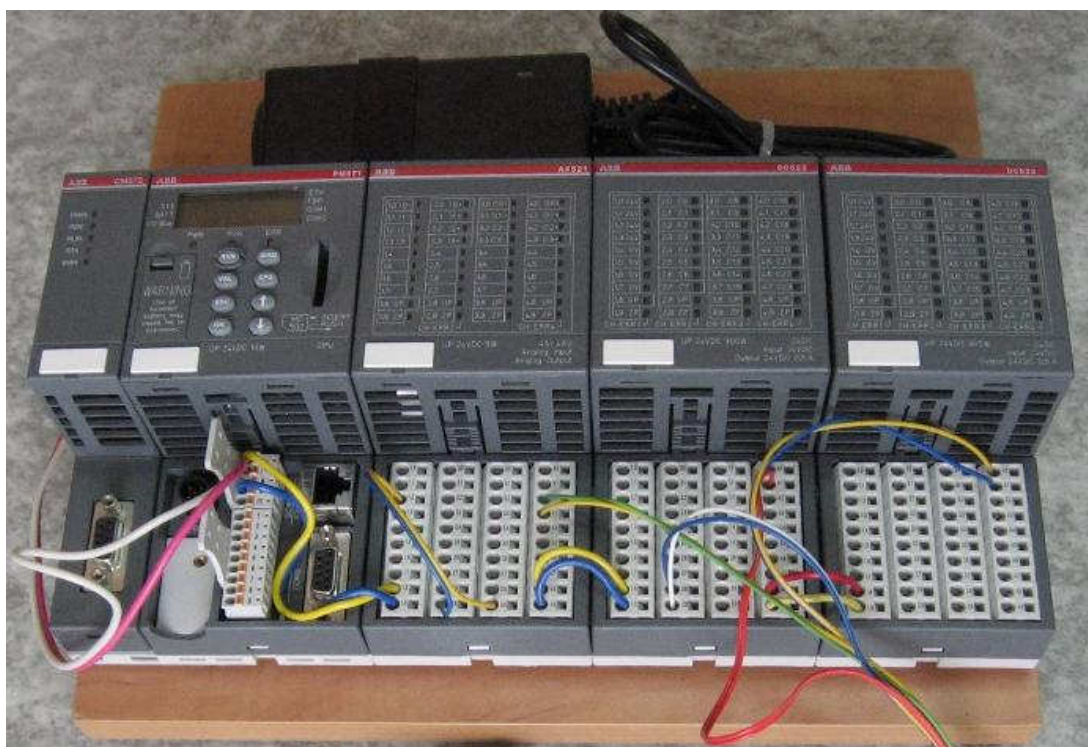
Přenos dat může probíhat čtyřmi různými způsoby:

- Synchronní komunikace vždy čekající na přenos dat z/do zařízení.
- Synchronní komunikace pracující s vyrovnávací pamětí serveru (cache).
- Asynchronní komunikace (vždy komunikuje se zařízením).
- Periodická komunikace serveru se zařízením a zpětné volání klienta při změně dat.

V případě druhého a čtvrtého způsobu komunikace je zapotřebí, aby server byl schopen sám vyvolávat komunikaci s perifériemi a dále informace ukládat do vyrovnávací paměti, nebo je předávat klientovi zpětným voláním. Díky používané technologii COM je každý klient schopen sám rozpoznat, s jakou verzí OPC serveru pracuje a může tímto požadovat odpovídající rozhraní dané verze. U některých klientů by mohl být problém s komunikací jednotlivých verzí, a proto každá nová v sobě integruje všechny předchozí, a tím zajišťuje kompatibilitu. [Manuál Control Web 6, 2013]

3 PRODUKTOVÁ ŘADA PLC SYSTÉMŮ AC500 OD FIRMY ABB

Programovatelné automaty (Programmable Logic Controlrs PLC) jsou v současné době jedním z nejvýznamnějších řídicích prostředků pro řízení technologickým procesů, výrobních linek a strojů. Mezi jejich významné přednosti patří robustnost, jednoduchost a spolehlivost provozu i v těžkých provozech. Modulární PLC AC500 od firmy ABB (obr. 7) je koncipováno jako stavebnice s univerzálními patičkami a zásuvnými moduly, které je možno připojit bez jakékoliv kabeláže. Základem celého systému je řídicí procesor, na který dále připevňujeme jednotlivé potřebné moduly, jako jsou komunikační procesory, I/O moduly, apod. [Modulární programovatelný automat AC500, 2007]



Obr. 7 Použité PLC řady AC500

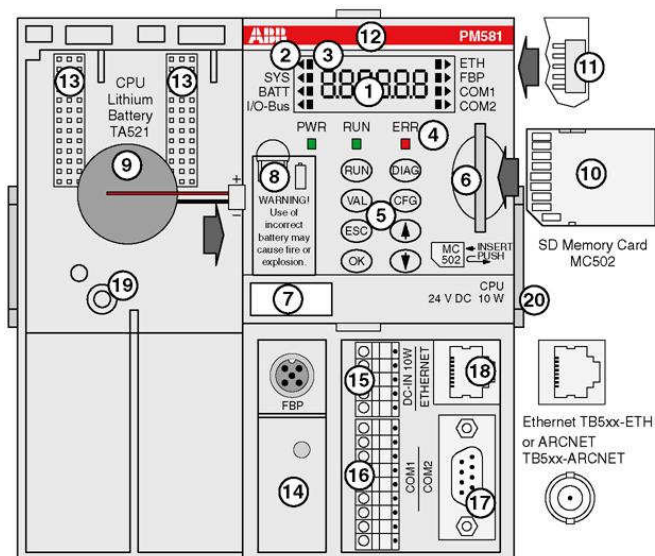
3.1 Procesorová jednotka PM 571

Procesorová jednotka se připevňuje jako všechny moduly na tzv. "Terminal Base", což je sběrnice, na které komunikuje s ostatními členy. Napravo od této jednotky je možno připojit až 7 digitálních nebo analogových modulů. Již základní jednotka obsahuje Ethernet a COM rozhraní. Dále také display pro zobrazení základních informací jako jsou: indikace připojení k síti, indikace připojení I/O modulů, stavu záložní baterie, chod programu, chybové hlášení a další. Důležitou součástí tohoto modulu je paměť pro ukládání údajů z procesu. Hlavní paměť je typu RAM (4MB) s možností zálohy baterií, další je typu Flash EPROM, nebo externě připojitelná běžná Secure Digital (SD) volitelná karta. Na čelním panelu CPU jednotky najdeme také trojici led diod, zobrazující spuštění programu, připojení napájení a chybový stav. Pro ovládání modulu zde máme osmici tlačítek, jejichž funkce je popsána v tabulce 2. [Modulární programovatelný automat AC500, 2007]

Tab. 2 Tlačítka panelu CPU jednotky

TLAČÍTKO	FUNKCE
RUN	Přepíná CPU do režimu Start a Stop
VAL	Slouží k procházení chybových kódů
ESC	Pro odchod z menu bez uložení
OK	Pro vstup do menu a následný odchod z menu s uložení
DIAG	Diagnostikuj a vyhodnotí chybové hlášení
CFG	Nastavuje síťovou adresu
↑	Zvyšuje hodnotu nebo nastavuje hodnotu na vyšší
↓	Snižuje hodnotu nebo nastavuje hodnotu na nižší

Tlačítka slouží pouze pro ovládání základních funkcí, nebo základních nastavení. Všechny ostatní funkce a nastavení se provádí pomocí síťového připojení, kdy je ale potřeba tuto síť dostatečně zabezpečit proti neoprávněnému zásahu. Samotný modul totiž postrádá jakoukoliv funkci pro identifikaci připojeného uživatele, a tudíž nedokáže rozlišovat práva. [Modulární programovatelný automat AC500, 2007]



- | | |
|--|---|
| 1. Sedmi segmentový display s podsvícením | 11. Konektor pro připojení I/O modulů |
| 2. Trojúhelníkový zobrazovač aktivních položek | 12. Slot pro CPU (CPU je integrováno) |
| 3. Čtvercový zobrazovač stavu | 13. Slot pro propojky (maximálně 4) |
| 4. Led indikující status | 14. Rozhraní pro připojení FieldBus |
| 5. Tlačítka | 15. Konektor pro připojení 24V |
| 6. Slot pro SD paměťovou kartu | 16. Sériový konektor COM1 |
| 7. Štítek | 17. Sériový konektor COM2 |
| 8. Místo pro lithiovou baterii | 18. Síťové rozhraní (Ethernet, nebo ARCNET) |
| 9. Lithiová baterie | 19. Otvor pro montáž na zeď |
| 10. SD paměťová karta | 20. DIN kolejnice |

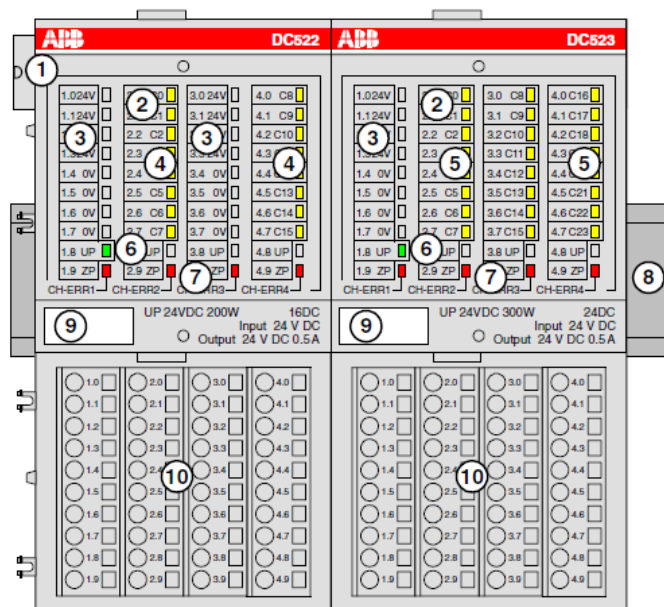
Obr. 8 Modul PM571 [Modulární programovatelný automat AC500,2007]

3.2 Modul binárních vstupů/výstupů DC523

Obsahuje celkem 40 konektorů pro připojení různých zařízení. Z toho jich je 24 konfigurovatelných. Dále jsou rozděleny podle vlastností:

- pro připojení senzorů, s proudem maximálně 0,5A
- s nulovým výstupním napětím
- 24 binárních vstupů/výstupů.
- pro připojení GND
- pro připojení 24V

[Modulární programovatelný automat AC500, 2007]



1. Komunikační I/O rozhraní
2. Indikátory k jednotlivým I/O rozhraním
3. Indikátor připojení 24V /0,5A
4. 16 žlutých indikátorů pro vstupně/výstupní konektory C0 až C15
5. 24 žlutých indikátorů pro vstupně/výstupní konektory C0 až C23
6. 4 červené indikátory, indukující procesní napětí
7. 4 červené indikátory zobrazující chyby
8. DIN kolejnice
9. Popisek
10. I/O terminál pro připojení 40 vstupně výstupních konektorů

Obr. 9 Modul DC523 [Modulární programovatelný automat AC500,2007]

3.3 Modul analogových vstupů/výstupů AX521

Modul obsahuje 4 analogové vstupy a 4 analogové výstupy konfigurovatelné jako:

- 0 - 10V
- -10 - 10V
- 0 - 20mA
- 4 - 20mA



Obr. 10 Modul AX521

4 VÝVOJOVÉ PROSTŘEDÍ PRO PLC - CODESYS

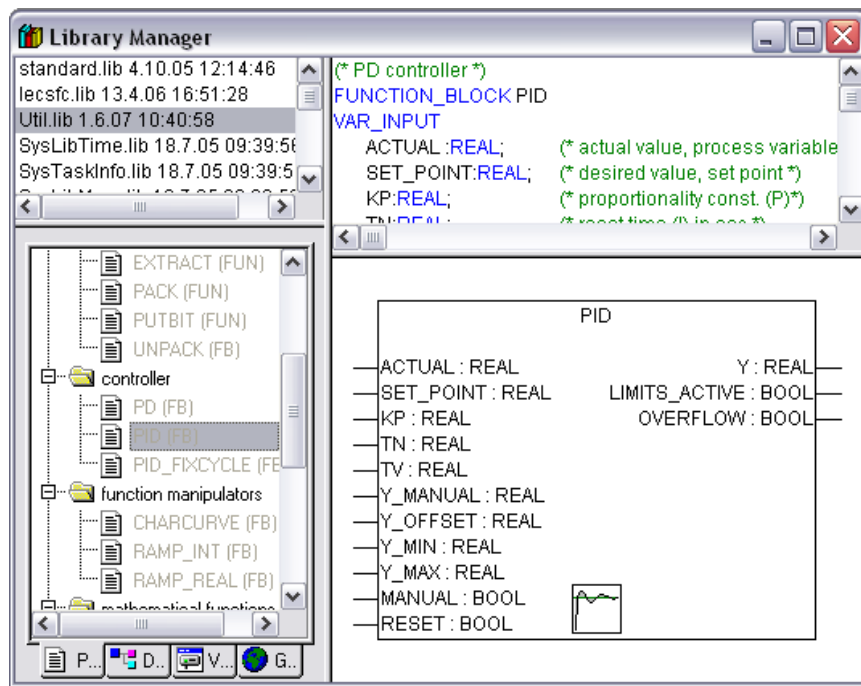
Vývojové prostředí CoDeSys vyvinulo mezinárodní sdružení CoDeSys Automation Alliance a představuje komplexní řešení pro programování PLC firmy ABB. Ovšem disponuje programovacím jazykem IAE, který je definován mezinárodní normou IEC 61131. Tuto normu programovacího jazyka dnes již používá drtivá většina výrobců programovatelných automatů (např.: Siemens, Mitsubishi, Unitronics, Teco atd.). Lze tedy bez ohledu na použitý hardware vytvářet ovládací programy a datové struktury. [ŠTĚPÁN, 2011]

4.1 Tvorba aplikace

Při tvorbě aplikací v tomto softwaru je k dispozici 6 programovacích jazyků, které je možné mezi sebou kombinovat a vytvářet tak uživatelem definované funkční bloky a makra, které podstatně usnadňují další práci. Tyto jazyky jsou:

- IL – instrukční list (Instruction List), syntaxe podobná Assembleru.
- LD – příčkový diagram (Ladder Diagram), v současné době nejoblíbenější jazyk užívaný programátory.
- FBD – funkční diagram (Function Block Diagram), LD rozšířený o volání funkcí.
- SFC – sekvenční funkční diagram (Sequential Function Chart), jazyk vhodný pro sekvenční aplikace, kdy vstup do další části programu je podmíněn vykonáním části předchozí a splněním zadaných podmínek.
- ST – strukturovaný text (Structured Text), vyšší programovací jazyk vhodný pro pokročilé matematické aplikace a složité algoritmy, jazyk je velmi podobný např. prostředí Pascal.
- CFC – spojitý funkční diagram (Continuous Function Chart).

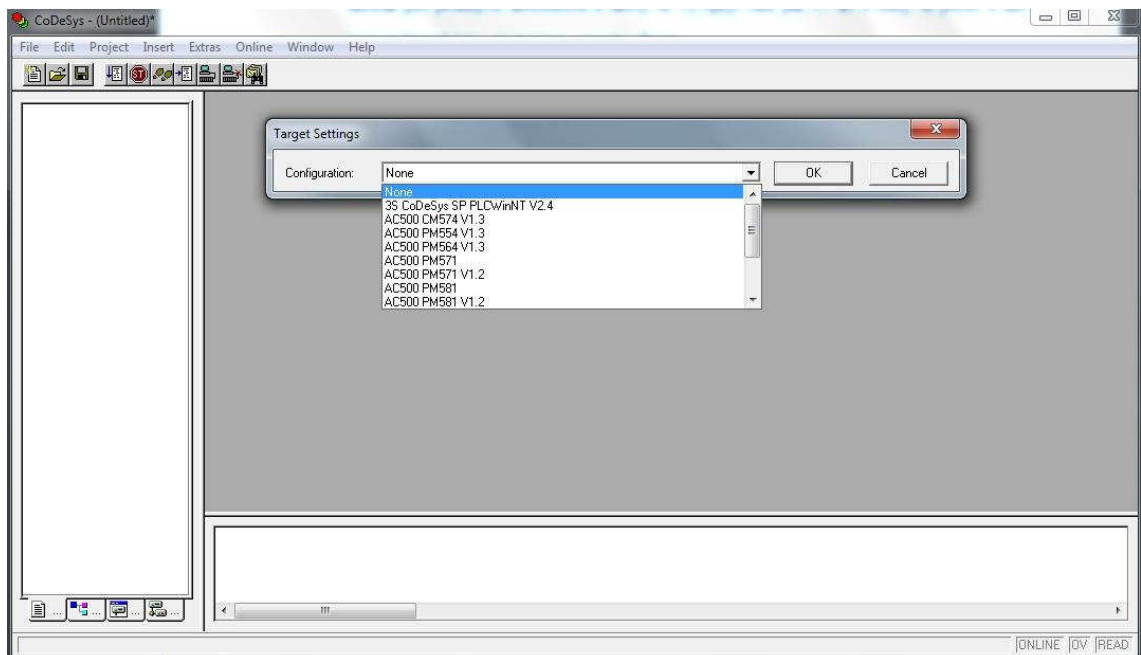
Velkou oporu a ulehčení práce představuje knihovna bloků (obr. 11). Jedná se o rozsáhlou knihovnu předdefinovaných funkčních bloků, která už v základní verzi čítá velké množství prvků, počínaje nejjednoduššími funkcemi AND až po složité funkční bloky rychlých čítačů, PID regulátorů a podobně.



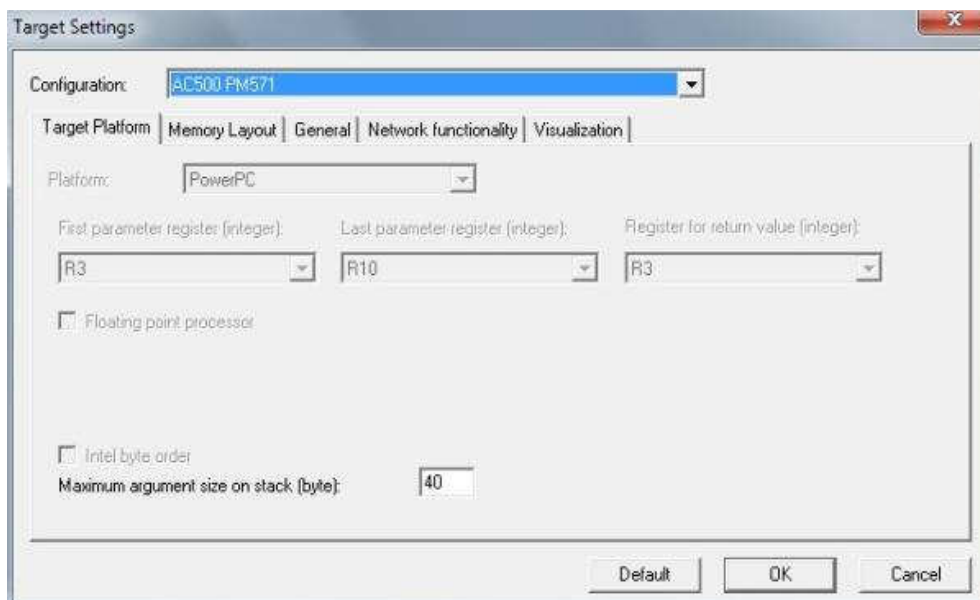
Obr. 11 Knihovna funkčních bloků

4.2 Grafické prostředí

Základ uživatelského prostředí je systém šablon tzv. *Targets* (obr. 12). Tento systém nám umožňuje výběr konkrétního hardwarového prostředku, který můžeme snadno konfigurovat a parametrizovat (obr. 13). Poté nám software automaticky načte potřebné knihovny k tomuto hardwaru a nastaví optimální hodnoty pro komunikaci a chod aplikace.

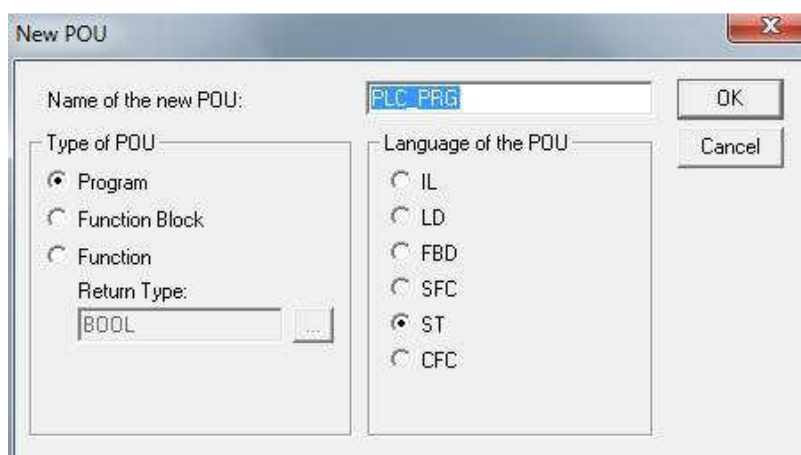


Obr. 12 Vložení šablony



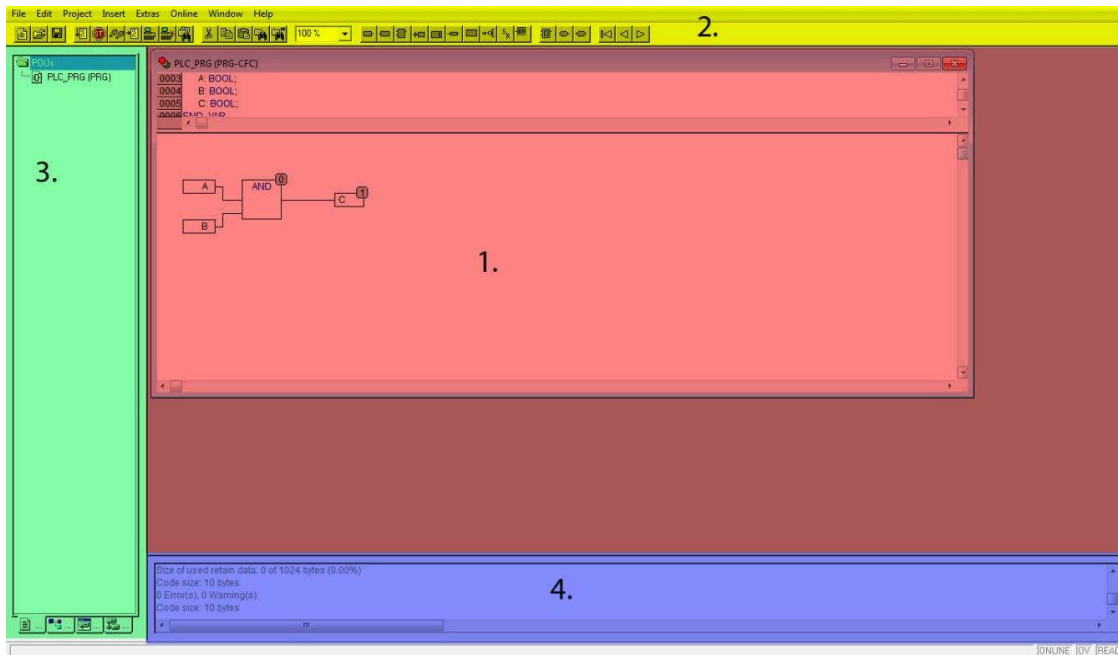
Obr. 13 Ukázka konfigurace parametrů hardwaru

Jakmile si nastavíme všechny potřebné parametry hardwaru, tak nás systém vyzve k volbě programovacího jazyka (obr. 14). Máme na výběr z již zmíněných šesti jazyků a dále můžeme vybírat s možností vytvoření programu, funkce, nebo jen funkčního bloku.



Obr. 14 Volba programovacího jazyka

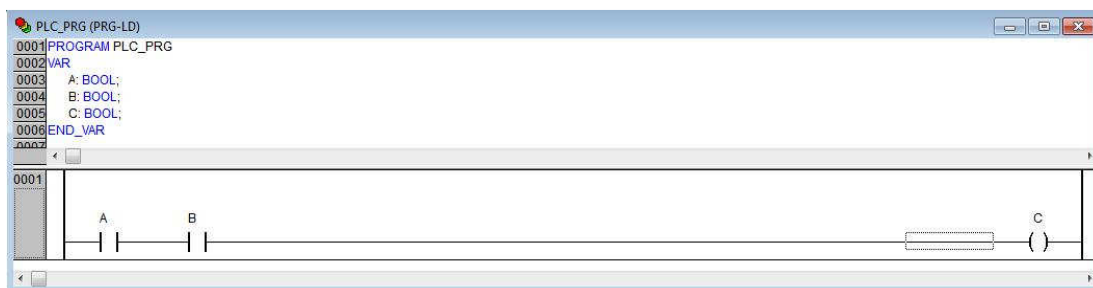
Nyní se dostáváme do hlavního grafického rozhraní, kde máme pracovní plochu rozdělenou do čtyř hlavních zón, jak je vidět na obrázku 15.



1. Hlavní pracovní prostor.
2. Panel nástrojů.
3. Objektový organizátor.
4. Okno zpráv.

Obr. 15 Pracovní plocha CoDeSys

Hlavní pracovní prostor slouží k vytváření samotného programu v jednotlivých programovacích jazycích a je pro většinu jazyků rozdělen na dvě části. V první části zadáváme název programu, funkce, nebo funkčního bloku a současně deklarujeme používané proměnné. V druhé části vytváříme samotný program v daném programovacím jazyku. Ukázka je na obrázku 16. [CoDeSys Service Tool CST, 2010]



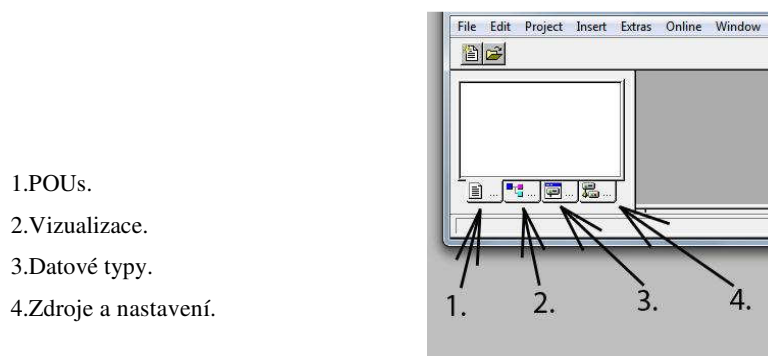
Obr. 16 Příklad jednoduchého programu v jazyku LD

Panel nástrojů obsahuje všechny příkazy ovládající chod programu, ale i příkazy pro vytváření programu, ovládání aplikace, ukládání, načítání a mnoho dalších (obr. 17).



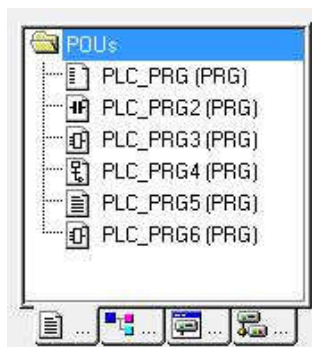
Obr. 17 Panel nástrojů

Objektový organizátor slouží pro správu celého projektu a jeho jednotlivých programů v různých, nebo i stejných jazycích.



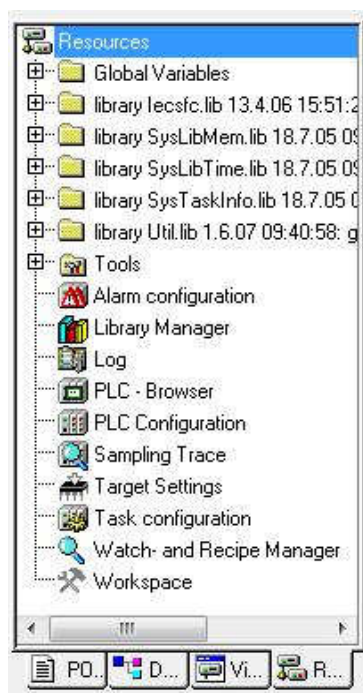
Obr. 18 Organizátor projektu

První záložka objektového organizátoru se nazývá "POUs" (1.), zde máme ve stromu seřazeny všechny programy projektu (obr. 19), které zde můžeme přejmenovávat, měnit pořadí a jejich typy.



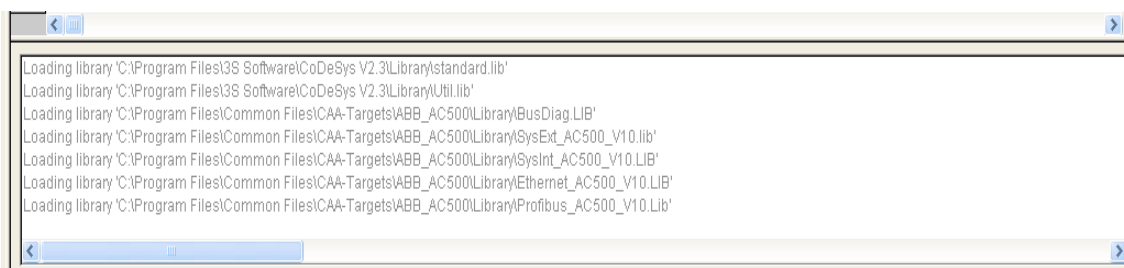
Obr. 19 Objektový manažer

V záložce "Vizualizace" (2.) pracujeme s vizualizací programovaného procesu. Můžeme vytvářet jednoduché objekty, tlačítka, indikátory, alarmy, tabulky pro trendy atd. V záložce "Datové typy" (3.) spravujeme uživatelsky definované datové typy, které si před svým spuštěním program alokuje v paměti PLC, na němž je spuštěn. Můžeme použít datové typy pro deklaraci čísla, textu, nebo řetězce. Uživatel může použít standardní typy, jako jsou ARRAY, BOOL, BYTE, DATE, DINT, DT, DWOD, INT, REAL, SINT, STRING, TIME, TOD, UDINT, UINT, USINT, WORD nebo si může nadefinovat typy vlastní. Záložka "Zdroje a nastavení" (4.) obsahuje několik velice důležitých položek (obr. 20). Záložku "Tools", která umožňuje nastavit prvotní komunikaci PLC v síti. Dále "PLC Configuration" pro nastavení všech připojených modulů v systému, "Sampling trace" nám nahrazuje funkci osciloskopu pro zobrazování vstupních a výstupních hodnot za běhu programu a v neposlední řadě také "Task configuration" kde nastavujeme, nebo editujeme časování jednotlivých úloh programu. [CoDeSys Service Tool CST, 2010]



Obr. 20 Zdroje a nastavení

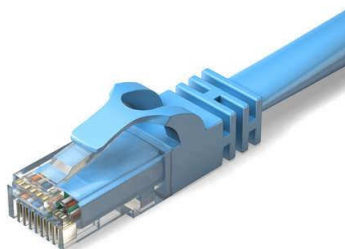
V Oknu zpráv se zobrazuje přehled provedených akcí, nebo nás program varuje o chybě v programu a současně nám nabízí možné řešení. [CoDeSys Service Tool CST, 2010]



Obr. 21 Okno zpráv

5 KOMUNIKACE S PLC

S PLC můžeme komunikovat několika různými způsoby. Ovšem pro nás bude nejvýhodnější použít stávající řešení komunikace. PLC je již připojeno do sítě a má přidělenou svou IP adresu. Přes tuto adresu je pak možné přistupovat k PLC, a tak do něj nahrávat náš program, nebo kontrolovat chod programu. Po připojení nám PLC bude posílat informace o stavu a my přímo uvidíme, který vstup je sepnutý a který rozepnutý. Tak stejně nám bude zobrazovat i stavy jednotlivých výstupů. Tato komunikace probíhá prostřednictvím zásuvky a kabelu RJ-45 (obr. 22). Konektor RJ-45 má přesně definované pořadí jednotlivých vodičů a jednotlivé vodiče mají svou funkci, podle použité přenosové rychlosti. Kabel je připojen do přepínače (switche), který spojuje jednotlivé komponenty v síti a dovoluje nám komunikovat prostřednictvím IP adres. [Vlach, 2000]



Obr. 22 Konecovka RJ-45 [www.cablestogo.com]

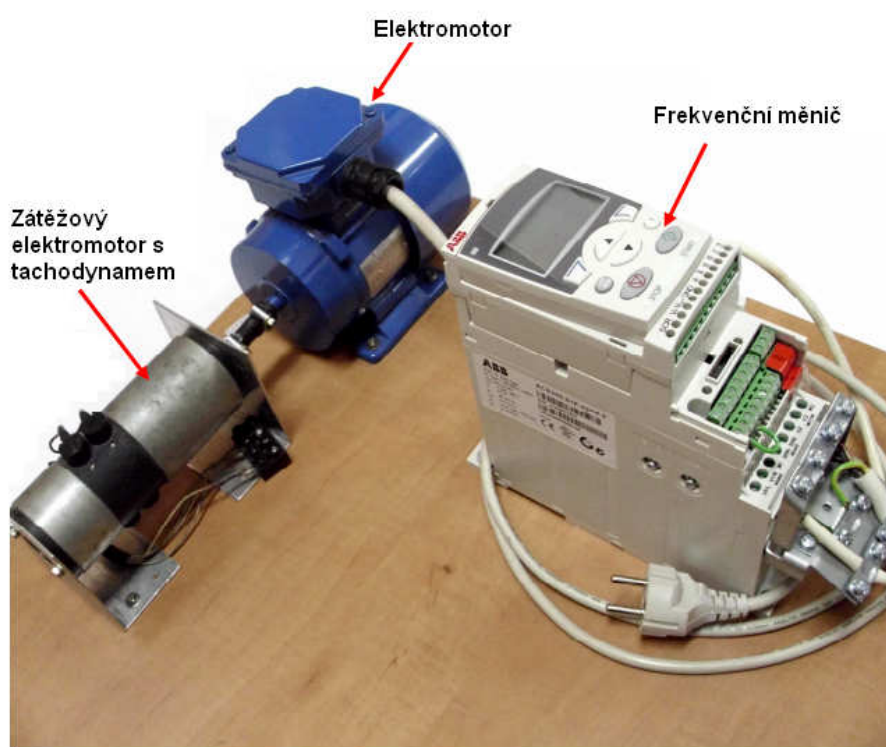
IP protokol v komunikaci

IP adresa slouží k rozlišení počítačů v síti. V dnešní době se používají IP adresy s 32 bitovým adresováním zapsané dekadicky po jednotlivých oktetech [Vlach, 2000].

Například: 192.168.1.1

6 LABORATORNÍ ÚLOHA ŘÍZENÍ ELEKTROMOTORU

První laboratorní model, na kterém byly prováděny úpravy, se skládá z frekvenčního měniče, střídavého indukčního elektromotoru a zátěžového elektromotoru s tachodynamem (obr. 23). Frekvenční měnič je připojen ke zdroji napájení, dále je k němu připojen samotný elektromotor. Frekvenční měnič slouží jako akční veličina pro ovládání elektromotoru, na který se z PC posílá žádaná hodnota otáček. Tachodynamo slouží na snímání otáček zátěžového motoru. Ten je pevně spojen s hřídelí ovládaného elektromotoru, tudíž snímané otáčky odpovídají také jeho otáčkám. Zátěžový elektromotor v chodu plní taktéž funkci brzdy. Jednotlivé prvky jsou umístěny na dřevěné desce čtvercového tvaru.



Obr. 23 Základní prvky laboratorního modelu

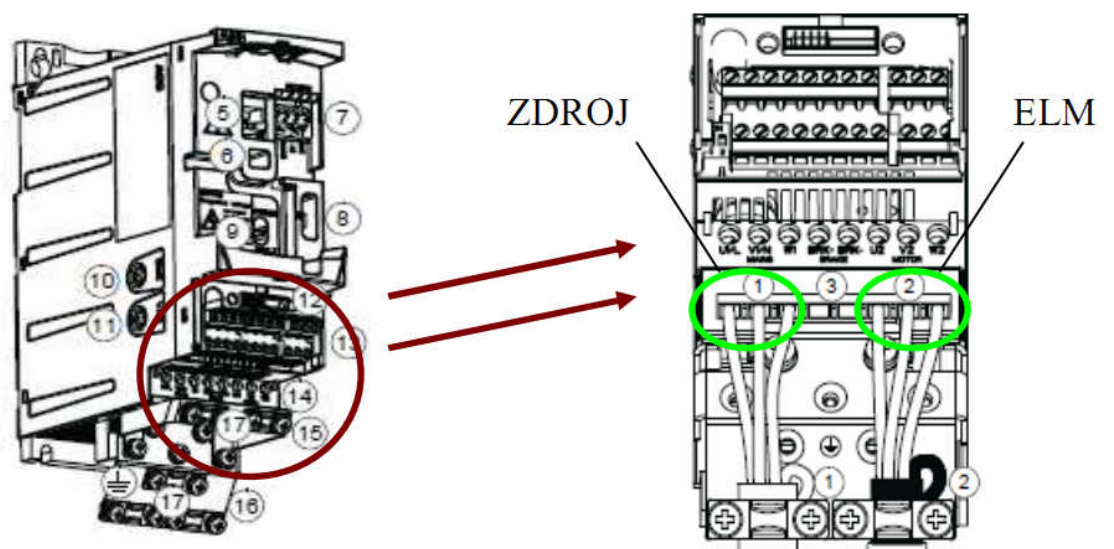
6.1 Popis jednotlivých prvků laboratorního modelu

Frekvenční měnič ACS 355

Tento měnič se používá pro řízení střídavých indukčních motorů a synchronních motorů s permanentním magnetem (obr. 24). Usměrňovač převádí jednofázové střídavé napětí na stejnosměrné napětí. Sada kondenzátorů v meziobvodu stabilizuje stejnosměrné napětí. Invertor převádí stejnosměrné napětí zpět na střídavé napětí pro střídavý motor. Brzdový chopper připojuje externí brzdový rezistor k stejnosměrnému meziobvodu, když napětí obvodu překročí maximální limit. Připojení měniče je znázorněno na obr. 25. [KADLEC, 2011]



Obr. 24 Pohled na nastavovací rozhraní frekvenčního měniče



Obr. 25 Schéma zapojení zdroje a elektromotoru do frekvenčního měniče

Střídavý indukční elektromotor (trojfázový)

Střídavý indukční elektromotor 220/380V 50Hz (obr. 26).

Technické údaje:

$P = 0,09\text{kW}$

$I = 0,5/0,29\text{A}$

ot. = 2820 r.p.m.

$\cos \varphi = 0,79$



Obr. 26 Střídavý indukční elektromotor

Zátěžový elektromotor s tachodynamem

Tachodynamo je napojeno na elektromotor viz obr. 27 a používá se pro generování napětí v závislosti na jeho rychlosti otáčení. Toto napětí slouží jako výstupní hodnota pro snímání otáček elektromotoru. Výstupní hodnota napětí se lineárně mění s otáčkami rotoru tachodynamu. Zátěžový elektromotor svým opačným směrem otáčení působí na ovládaný elektromotor jako proměnlivá zátěž.



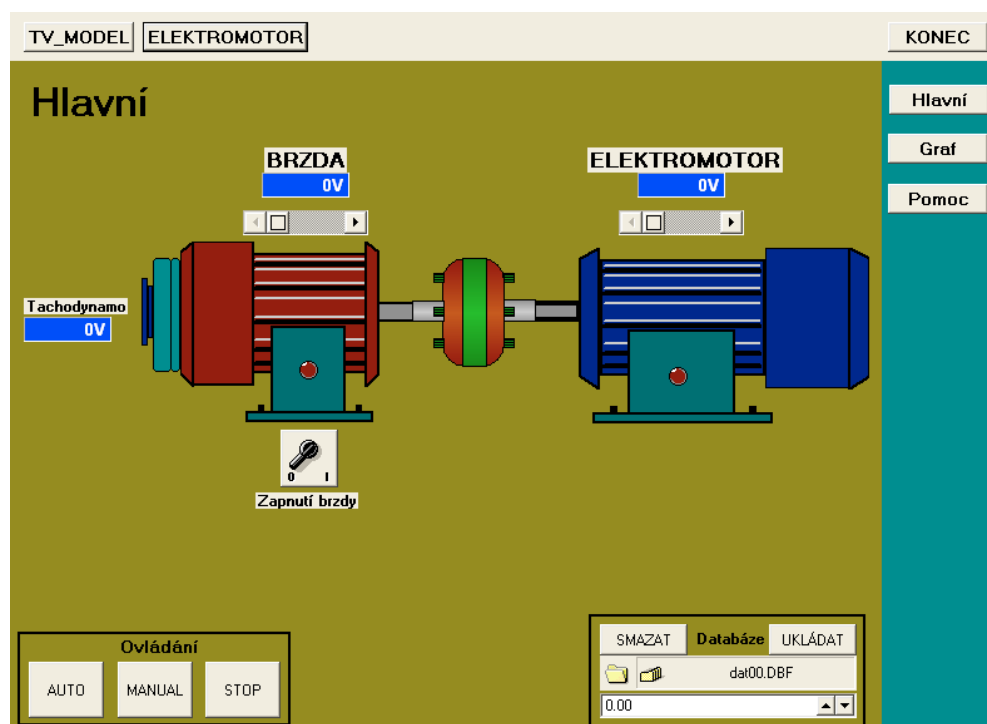
Obr. 27 Zátěžový elektromotor s tachodynamem

6.2 Provedené úpravy v laboratorní úloze

Byla provedena úprava stávající úlohy jak z programové, tak i z hardwarové stránky. K laboratorní úloze jsem připojil výše popsané PLC od firmy ABB, které slouží pro ovládání laboratorní úlohy a zaznamenávání napětí ze snímače otáček. Bylo vytvořeno vizualizační prostředí pro ovládání úlohy a také obvod pro unifikaci signálu s regulovatelným zdrojem napětí, který byl vyveden do konektoru Cannon 9. Pro přenos dat mezi systémem Control Web 6 a PLC byl použit protokol OPC.

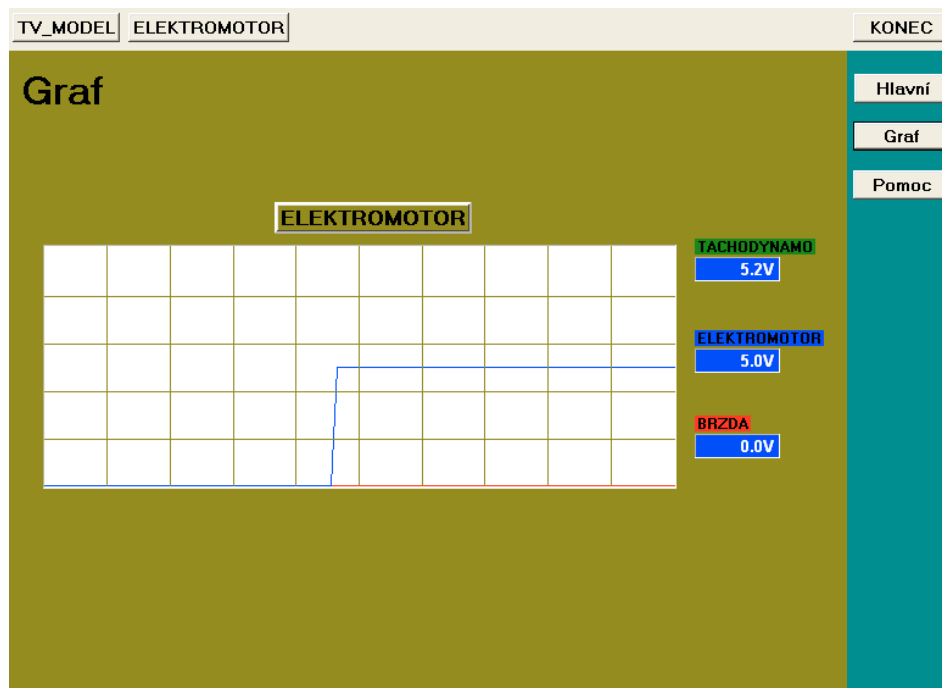
6.2.1 Vizualizace úlohy s elektromotorem

Na hlavní panel béžové barvy jsem umístil jednotlivé ovládací prvky a zobrazovací zařízení (obr. 28). V dolní části jsou umístěny tlačítka pro volbu manuálního nebo automatického režimu realizovaného PID regulací. Parametry PID regulátoru je možno nastavit po kliknutí na grafické znázornění elektromotoru. Dále je zde umístěn switch pro zapnutí a vypnutí brzdy. V levé části je umístěn zobrazovač napětí ze snímače otáček a nahoře ovladače pro ovládání analogových výstupů PLC, respektive pro ovládání výstupního napětí frekvenčního měniče a napětí na zátěžovém elektromotoru. Uprostřed je vyobrazen samotný laboratorní model. OPC Server je nainstalován pro program CoDeSys, pomocí něhož probíhá komunikace mezi systémem Control Web a PLC, potažmo programem CoDeSys. Jako každý jiný ovladač v systému Control Web i OPC klient potřebuje dva konfigurační soubory — parametrický soubor (.PAR) a mapovací soubor (.DMF — Driver Map File). Pro jejich vytvoření jsem použil nástroj Control Webu s grafickým rozhraním umožňující pohodlnou tvorbu obou souborů.



Obr. 28 Hlavní obrazovka programu

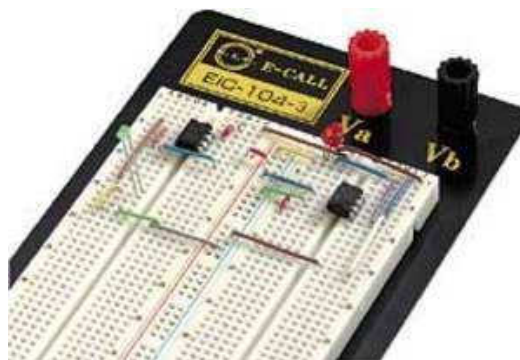
Jak je patrné v pravém dolním rohu, vytvořená vizualizace umožňuje měření a nastavované údaje ukládat do souboru nebo zobrazit tyto údaje v grafu viz obr. 29.



Obr. 29 Obrazovka s grafem

6.2.2 Vytvoření unifikačního modulu

Pro komunikaci laboratorního modelu s okolím jsem vytvořil unifikační modul se vstupy a výstupy vyvedenými do konektoru Cannon 9. Unifikační modul jsem nejprve realizoval v kontaktním poli (obr. 30), otestoval a následně vytvořil schéma a návrh unifikační desky v programu Eagle, které jsem umístil do příloh A a B. Následně jsem zhotovil samotnou desku s obvodem, vyvrtal do ní dírky pro součástky a napájel je. Vytvořenou desku jsem umístil do plastového krytu a připevnil na dřevěnou podstavu laboratorního modelu.



Obr. 30 Kontaktní pole

Napěťový dělič pro unifikaci signálu z tachodynamu

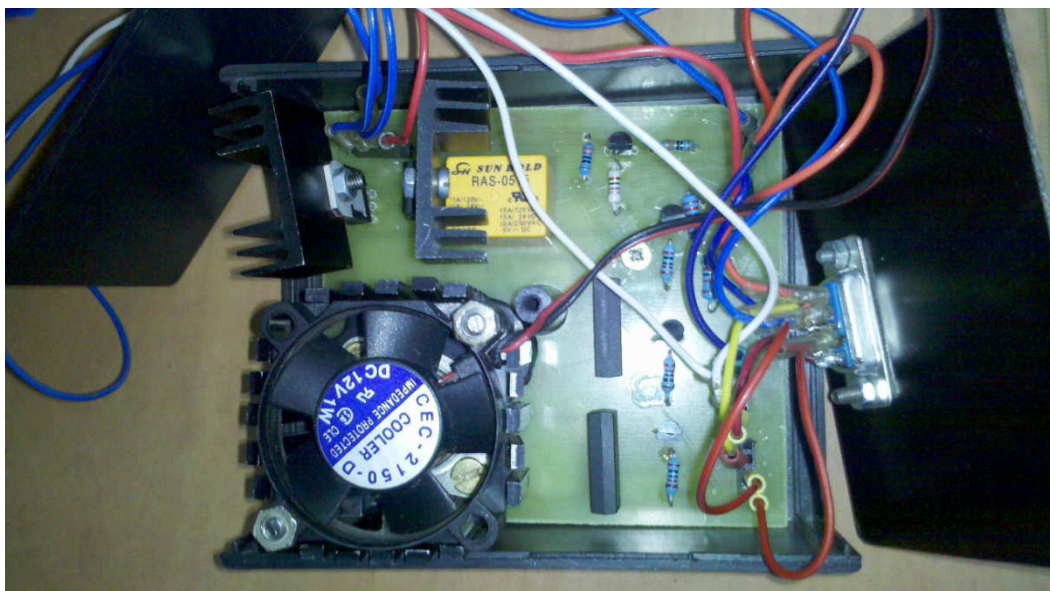
Po seznámení s laboratorní úlohou jsem zjistil, že výstupní napětí z tachodynamu je v rozmezí -20 až 20V, ale vstupní rozsah analogového vstupu PLC je jen -10 až 10 V, tudíž prvním úkolem bylo navrhnout napěťový dělič, který jsem vytvořil ze dvou odporů o velikosti 10kΩ.

Reléové obvody pro diskrétní vstupy frekvenčního měniče

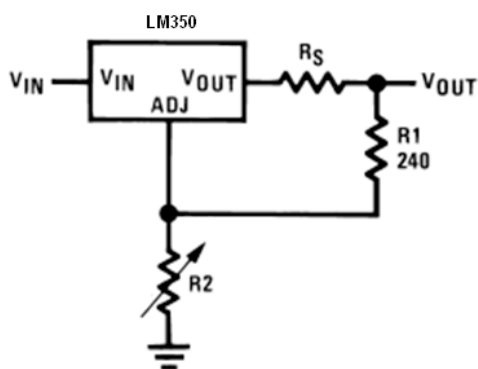
Pro ovládání frekvenčního měniče jsem navrhl obvody s reléovými prvky. Jako zdroj proudu mi posloužilo výstupní pomocné napětí měniče. Jako ovládací napětí použitých reléových součástek jsem použil referenční napětí měniče, které jsem pomocí stabilizátoru L78S05 stabilizoval na 5V. Do obvodu jsem umístil tranzistor BC547A jako realizaci spínače a na jeho bázi jsem připojil diskrétní výstup PLC jednotky. Takto jsem realizoval zapnutí a vypnutí měniče i změnu směru otáčení elektromotoru.

Realizace zdroje napětí pro zátěžový elektromotor

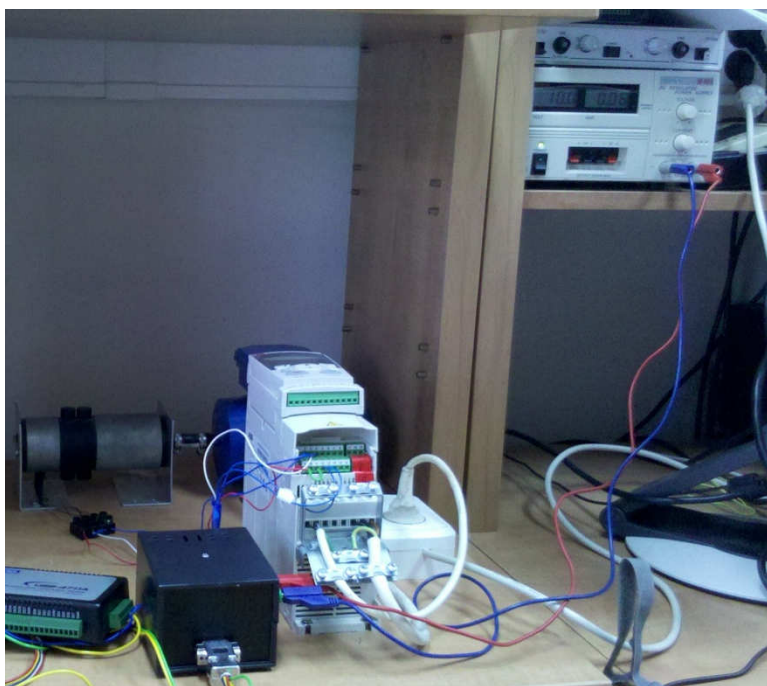
Jako zdroj napětí jsem zvolil regulovatelný zdroj stejnosměrného napětí Manson NP-9615. Po prvotním testování jsem dospěl k názoru, že pro brzdění ovládaného elektromotoru nejlépe vyhovuje napětí zdroje 10V. Abych mohl analogově ovládat velikost zátěže, musel jsem navrhnout obvod s regulovatelným stabilizátorem LM350K. Výstupní napětí stabilizátoru se obvykle nastavuje proměnným odporem (obr. 31, 32). Abych mohl výstupní napětí na stabilizátoru nastavovat analogově z vizualizace, nahradil jsem jej tranzistorem, na jehož bázi jsem přivedl napětí z analogového výstupu PLC. Obvod byl ještě doplněn rezistory pro nastavení pracovního bodu tranzistoru. Pro snadné vypínání a zapínání brzdy jsem do obvodu umístil relé se spínacím tranzistorem, podobně jako výše, avšak díky velkým proudům protékajících relé jsem musel zvolit typ s maximálním proudem 15A ve výkonové větvi. Po důkladném testování funkčnosti jsem dospěl k názoru, že je potřeba chladit stabilizátor LM350K. Připevnil jsem na něj proto chladič s větráčkem. Propojení mezi zdrojem Manson a unifikačním modulem je realizováno pomocí propojovacích kabelů, kdy v krabici s unifikační deskou jsou umístěny zdířky pro připojení banánků od propojovacích kabelů (obr. 33).



Obr. 31 Horní pohled na desku s unifikační mi obvody



Obr. 32 Obvyklé zapojení stabilizátoru LM350K



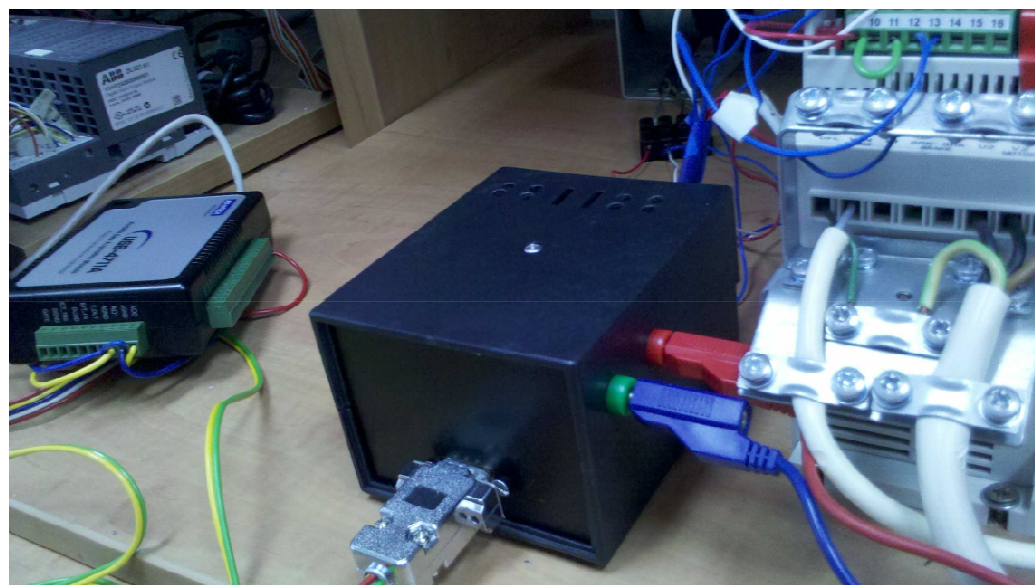
Obr. 33 Propojení modelu s regulovatelným zdrojem stejnosměrného napětí Manson NP-9615

Propojení unifikačního modulu s PLC

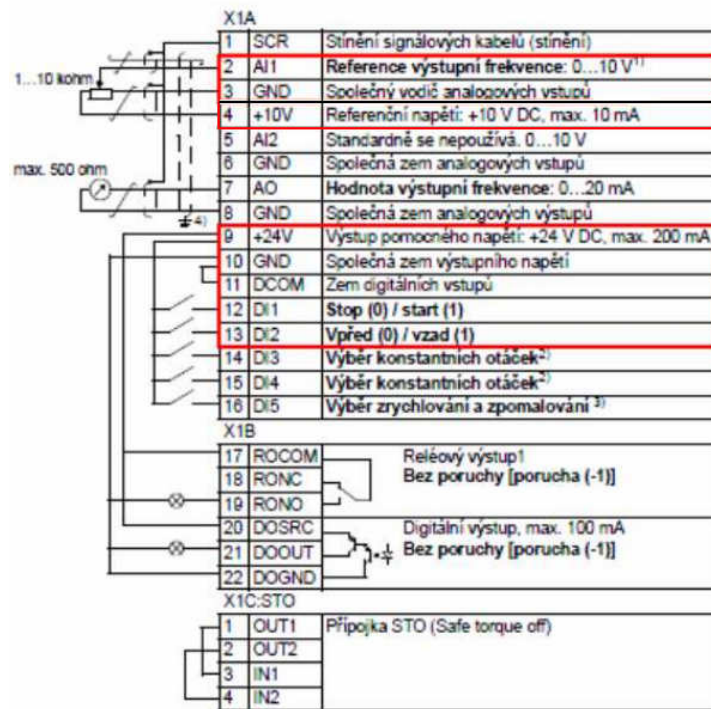
Do přední strany krabičky s vyrobenou unifikační deskou (obr. 34) jsem vyvrtal otvor a umístil do něj konektor Cannon 9 (samce), ke kterému jsem připájel vstupy a výstupy unifikační desky. Do PLC jsem připevnil propojovací kabely a druhé konce kabelů jsem připájel opět ke konektoru Cannon 9 (samice). Zapojení pinů je popsáno v tabulce tab. 5 a zapojení vstupů a výstupů měniče je znázorněno na obr. 35.

Tab. 3 Zapojení jednotlivých pinů konektoru Cannon 9

Popis	Kontakt
Vstupy do PLC:	Cannon 9
ANALOG – AI0 – Napětí z tachodynamy	7
Výstupy z PLC:	
ANALOG – AO0 – Ovládací napětí pro měnič	2
ANALOG – AO1 – Ovládací napětí pro zátěž	1
DISKRÉTNÍ – DO0 – START/STOP měniče	5
DISKRÉTNÍ – DO1 – Směr otáčení elektromotoru	4
DISKRÉTNÍ – DO2 – Zapnutí zátěže (brzdy)	6
ZEM	9



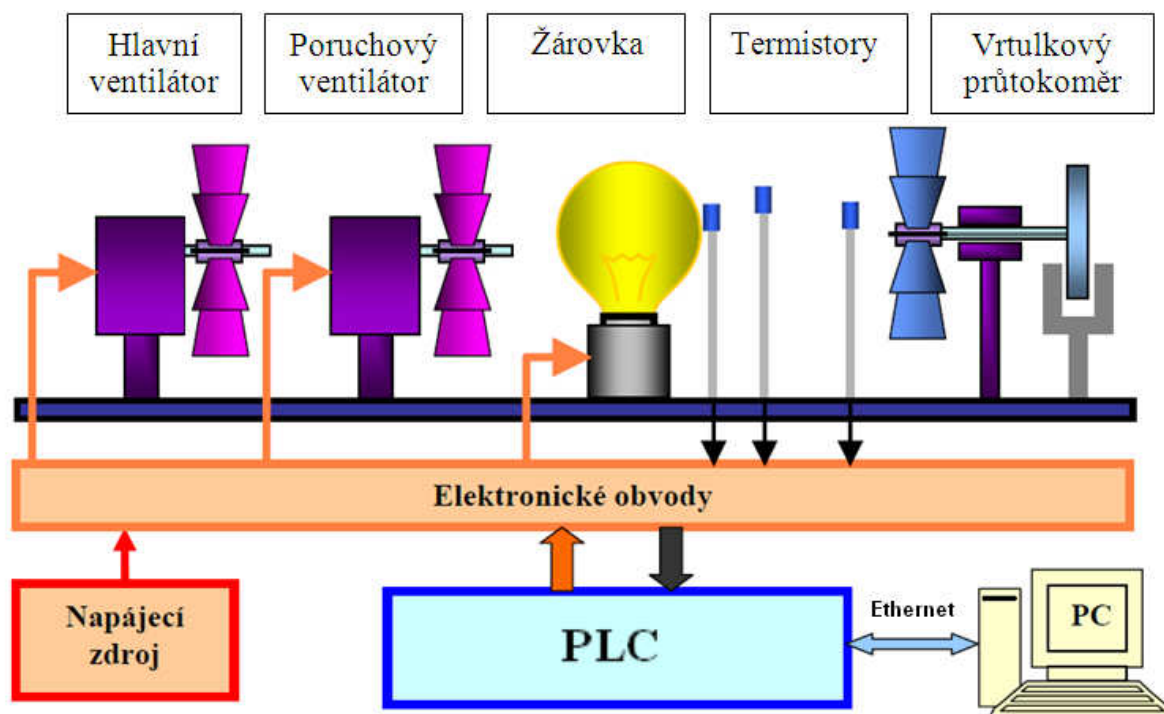
Obr. 34 Unifikační modul



Obr. 35 Zapojení vstupů a výstupů frekvenčního měniče

7 LABORATORNÍ ÚLOHA ŘÍZENÍ TEPELOVZDUŠNÉHO MODELU

Jako druhou úlohu jsem si zvolil řízení teplovzdušného modelu. Teplovzdušné modely (TVM) realizované na katedře ATR představují zjednodušený fyzikální model klimatizační jednotky. Je tvořen žárovkou napájenou z ovladatelného zdroje napětí. Žárovka je tepelným i světelným zdrojem a je umístěná v krytém tunelu, kterým je nasáván či vytlačován vzduch pomocí hlavního a případně poruchového (vedlejšího) ventilátoru (ten je rovněž napájen pomocí říditelného zdroje napětí). Popisovaný model umožňuje realizovat jak jednorozměrové, tak mnohorozměrové regulační úlohy. Model teplovzdušného obvodu můžeme ovládat dvěma akčními členy tj. zdroj tepla – žárovka a hlavní ventilátor. Obsluha má možnost volby výstupní (měřené) veličiny, kterou může být teplota měřená snímači umístěnými v různých vzdálenostech od zdroje tepla – žárovka, nebo průtok vzduchu v tunelu měřený vrtulkovým průtokoměrem. V tunelu je umístěno několik termistorových snímačů teploty, jeden snímač průtočného množství vzduchu, což je již výše zmíněný vrtulkový průtokoměr. [SMUTNÝ, 2005] Stejně jako v předešlé úloze používám pro ovládání a řízení modelu PLC od firmy ABB. Schéma úlohy můžete vidět na obrázku 36.



Obr. 36 Schéma úlohy teplovzdušného modelu

7.1 Popis jednotlivých prvků laboratorního modelu

Unifikační modul

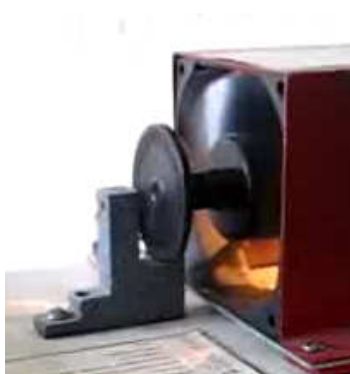
Jde o část modelu, kde jsou umístěny vyhodnocovací a zesilovací obvody (obr. 37). Jeho základ tvoří základní deska. Hlavním účelem základní desky je propojit jednotlivé součástky do fungujícího celku a poskytnout jim elektrické napájení. Do slotů základní desky propojených společnou sběrnici se svisle zasouvají jednotlivé bloky pro unifikaci a zesílení, což umožňuje modifikovatelnost celého systému v závislosti na požadavcích reálné úlohy. Jsou to menší desky s elektronikou k jednotlivým částem teplovzdušného modelu, což jsou deska termistorů, deska pro ovládání žárovky, deska pro ventilátory a deska pro vrtulkový průtokoměr. Pomocí stabilizátorů je odděleno napájení výkonové části modelu od napájení části zpracovávající signály.



Obr. 37 Horní pohled na odkrytovaný přístrojový modul

Vrtulkový průtokoměr

Požitý vrtulkový průtokoměr je tvořen vrtulkou, kterou uvádí do otáčivého pohybu kinetická energie proudícího plynu (vzduchu), snímačem otáček a modulem pro zpracování pulsů ze světelné závory. Rychlost otáčení vrtulky je úměrná střední rychlosti proudu, procházejícího příčným průřezem tunelu. Tyto průtokoměry pracují spolehlivě až do určité velikosti průtoku. Základem je monostabilní klopný obvod 74123 nebo MAF115 (obr. 38). [SMUTNÝ 2005]



Obr. 38 Pohled na vrtulkový průtokoměr v činnosti

Termistory

Termistor je také elektrotechnická součástka, jejíž elektrický odpor je závislý na teplotě. Rozlišujeme dva druhy termistorů - NTC a PTC termistor. NTC je termistor s negativním teplotním koeficientem, což znamená, že s rostoucí teplotou součástky odpor klesá. Jejich velikost umožňuje téměř bodové měření teploty a spolu s vysokou citlivostí splňují tyto součástky základní nároky na miniaturizaci techniky. Jejich většímu rozšíření brání jejich časová nestabilita a za nevýhodu lze považovat značnou nelineární závislost jejich odporu na teplotě.



Obr. 39 Příklad termistoru [www.elektronikshop.sk]

Ventilátor

Jako hlavní a i poruchový ventilátor je použit model ventilátoru jako je v klasickém PC.



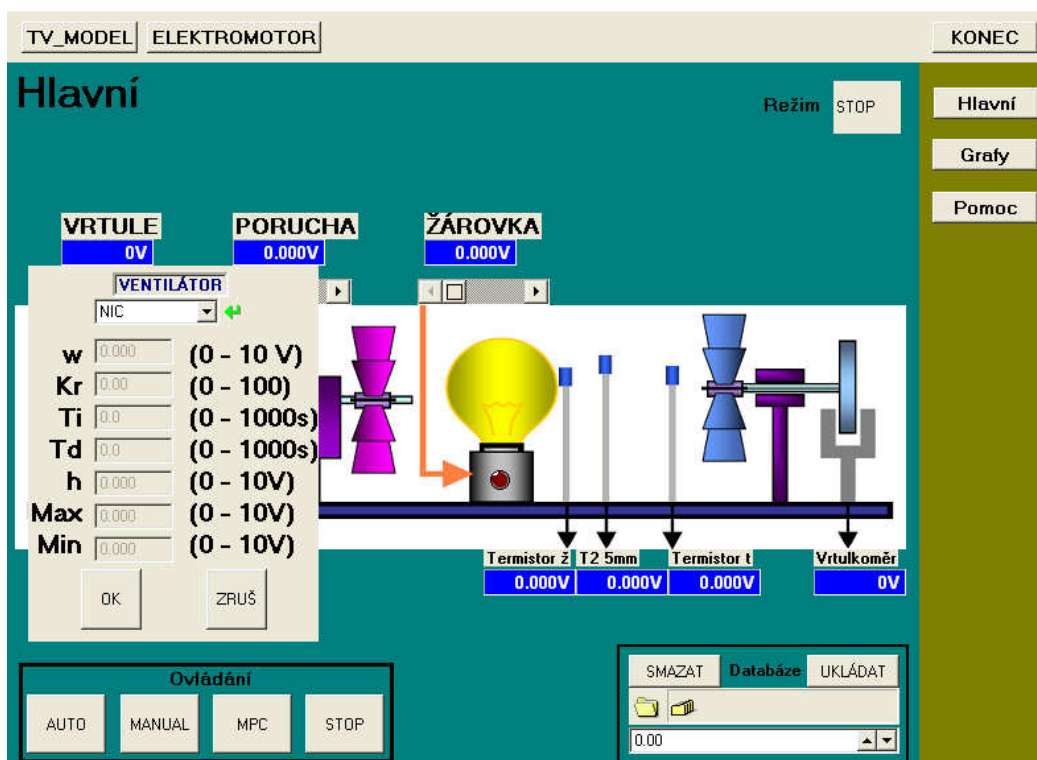
Obr. 40 Příklad ventilátoru [www.alza.cz]

7.2 Provedené úpravy v laboratorní úloze

K laboratorní úloze jsem připojil výše popsané PLC od firmy ABB, které slouží pro ovládání laboratorní úlohy a zaznamenávání napětí ze snímače otáček. Bylo vytvořeno vizualizační prostředí pro ovládání úlohy. Pro přenos dat mezi systémem Control Web 6 a PLC byl použit protokol OPC.

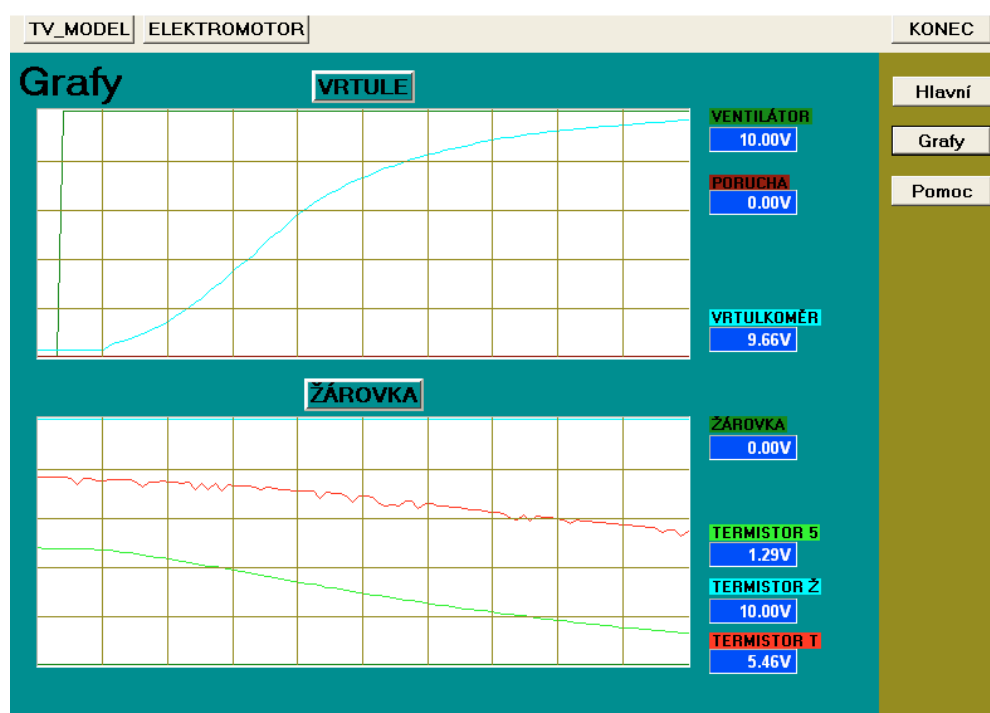
7.2.1 Vizualizace úlohy s teplovzdušným modelem

Na hlavní panel modré barvy jsem umístil jednotlivé ovládací prvky a zobrazovací zařízení. (obr. 41). V dolní části jsou umístěny tlačítka pro volbu manuálního nebo automatického režimu realizovaného PID regulací a také pro možnost připojení prediktivního regulátoru. Nahoře jsou umístěny analogové ovladače pro ovládání analogových výstupů PLC, respektive pro ovládání napětí na žárovce a poruchovém či hlavním ventilátoru. Uprostřed je vyobrazen samotný laboratorní model. Pokud chceme nastavit parametry PID regulátoru nebo dvoupolohové regulace, klikneme na daný prvek, který bude reprezentovat akční zásah a zobrazí se okno pro vložení parametrů. Poruchovou veličinu můžeme nastavit jako trojúhelníkový nebo obdélníkový impuls. Vizualizace a konfigurace si předává hodnoty proměnných přes OPC server.



Obr. 41 Hlavní obrazovka programu

Jak je patrné v pravém dolním rohu, vytvořená vizualizace umožňuje měření a nastavované údaje ukládat do souboru nebo zobrazit tyto údaje v grafu viz obr. 42.



Obr. 42 Obrazovka s grafy úlohy

7.2.2 Propojení unifikačního modulu s PLC

Pro propojení laboratorního modelu s PLC používám jedenácti žilný vodič zakončený konektorem Cannon 25. zapojení jednotlivých pinů je patrné z tabulky 4.

Tab.4 Zapojení jednotlivých kontaktů

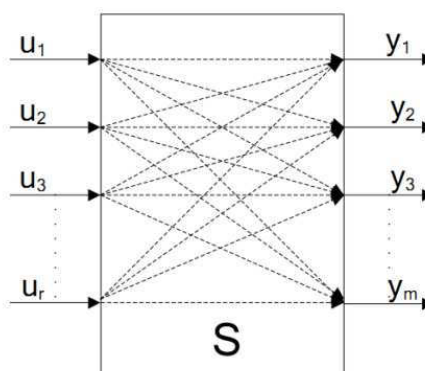
Popis	Kontakt
Vstupy do PLC: signály ze snímačů	Cannon 25
ANALOG – Termistor na baňce (NR 354 20K U)	1
ANALOG – Termistor 5 mm od baňky (NR 354 20K U)	2
ANALOG – Termistor na baňce (KTY 81 210)	3
ANALOG – Vrtulkový průtokoměr	4
Výstupy z PLC: akční veličina	
ANALOG – Žárovka	19
ANALOG – Ventilátor (primární)	20
DISKRÉTNÍ – Ventilátor (sekundární, poruchový)	15-18
ZEM	25

8 APLIKACE PREDIKTIVNÍHO ŘÍZENÍ

Během zimního semestru 2012/2013 jsem absolvoval pracovní stáž ve Vídni ve firmě VOIGT+WIPP Engineers GmbH, kde jsem se zúčastnil projektu vývoje prediktivního algoritmu pro PLC na bázi PC. Mým úkolem bylo vytvořit v programu CoDeSys aplikaci pro PLC ve standardu IEC 1131 určenou pro převod matice přenosů mnohorozměrného systému s maximálním počtem šesti vstupních veličin na stavový popis a také jeho diskretizaci. Druhým úkolem bylo vytvořit vizualizaci ve vizualizačním softwaru SolutionCenter od firmy Bachmann, přes kterou se budou vkládat data pro prediktivní řízení.

8.1 Matematický popis aplikace

Mnohorozměrový regulační obvod je takový obvod, který má více regulovaných veličin y . (obr. 43)



Obr. 43 Závislosti vstupních a výstupních veličin

Vztah mezi jednou vstupní veličinou u a jednou výstupní veličinou y je v našem případě popsán přenosem jako ve vzorci 2, který popisuje proporcionalní systém se setrvačností třetího řádu. Třetí řád systému uvažujeme jako maximální. Všechny dané přenosy můžeme zapsat do matice přenosů G .

$$\mathbf{G}(s) = \begin{bmatrix} G_{11}(s) & \dots & G_{1m}(s) \\ \vdots & \ddots & \vdots \\ G_{r1}(s) & \dots & G_{rm}(s) \end{bmatrix} \quad (1)$$

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_1 s^2 + b_2 s + b_3}{a_0 s^3 + a_1 s^2 + a_2 s + a_3} = \frac{\frac{b_1}{a_0} s^2 + \frac{b_2}{a_0} s + \frac{b_3}{a_0}}{s^3 + \frac{a_1}{a_0} s^2 + \frac{a_2}{a_0} s + \frac{a_3}{a_0}} \quad (2)$$

Vztah mezi jednou vstupní a jednou výstupní veličinou popsany přenosem jako ve vzorci 2 můžeme pomocí metody postupné integrace převést na maticovou diferenciální rovnici neboli stavový popis v kanonickém tvaru pozorování.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\frac{a_1}{a_0} & 1 & 0 \\ \frac{a_2}{a_0} & 0 & 1 \\ -\frac{a_3}{a_0} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{b_1}{a_0} \\ \frac{b_2}{a_0} \\ \frac{b_3}{a_0} \end{bmatrix} u(t) \quad (3)$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 0u(t)$$

Což lze zapsat jako:

$$\dot{\mathbf{x}} = \mathbf{Ax}(t) + \mathbf{Bu}(t) \quad (4)$$

$$y = \mathbf{Cx}(t) + \mathbf{Du}(t)$$

kde

A...je matice systému

B...je matice vstupu

C...je matice výstupu

D...je matice přímé vazby výstupu na vstup

Zde použitý způsob (metoda postupné integrace) má výhodu v tom, že výstupní veličina $y(t)$ je přímo jednou ze stavových veličin a že v rovnici vystupuje jen vstupní veličina $u(t)$, ne její derivace. Použitý systém splňuje silnou podmínku fyzikální realizovatelnosti a výstup závisí na stavových veličinách a rovnice výstupu tudíž neobsahuje vstupní veličiny. Pro tyto systémy je **D** nulová matice (**D**=0).

$$\dot{\mathbf{x}} = \mathbf{Ax}(t) + \mathbf{Bu}(t) \quad (5)$$

$$y = \mathbf{Cx}(t)$$

Dalším úkolem bylo systém diskretizovat na tvar

$$\mathbf{x}[(k+1)T] = \mathbf{Fx}(kT) + \mathbf{Gu}(kT) \quad (6)$$

$$y[kT] = \mathbf{Cx}(kT)$$

kde platí

$$\mathbf{F} = e^{\mathbf{AT}} \quad (7)$$

$$\mathbf{G} = \left(\int_0^T e^{\mathbf{AV}} dV \right) \mathbf{B} \quad (8)$$

Pro rozklad maticové exponenciály a jejího integrálu byl použit rozklad exponenciály na Taylorovu řadu.

$$\mathbf{V} = \int_0^T e^{\mathbf{AV}} dV = \mathbf{IT} + \frac{\mathbf{AT}^2}{2!} + \frac{\mathbf{A}^2\mathbf{T}^3}{3!} + \dots + \frac{\mathbf{A}^i\mathbf{T}^{i+1}}{(i+1)!} \quad (9)$$

Po dosazení platí:

$$\mathbf{F} = \mathbf{I} + \mathbf{AV} \quad (10)$$

$$\mathbf{G} = \mathbf{VB} \quad (11)$$

k tomuto rozkladu jsem v algoritmu použil cyklus WHILE s podmínkou na začátku, která vychází z Eukleidovské normy viz příloha C.

$$\left\| \frac{\mathbf{A}^i \mathbf{T}^{i+1}}{(i+1)!} \right\| = \sqrt{\sum_{k=1}^p \sum_{l=1}^m a_{kl}^2} \quad (12)$$

$$\left\| \frac{\mathbf{A}^i \mathbf{T}^{i+1}}{(i+1)!} \right\| > 0.0001$$

Po diskretizaci stavového modelu mají jeho matice následující tvar.

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \quad (13)$$

$$\mathbf{G} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}$$

$$\mathbf{C} = [1 \quad 0 \quad 0]$$

Dalším požadavkem na mou aplikaci byla také možnost vložení přenosu s dopravním zpožděním T_d .

$$G(s) = \frac{\frac{b_1 s^2 + b_2 s + b_3}{a_0}}{s^3 + \frac{a_1 s^2 + a_2 s + a_3}{a_0}} e^{-T_d s} \quad (14)$$

Toto by se dalo vyřešit například tím, že by se dopravní zpoždění aproximovalo přenosem x . řádu pomocí Padého rozvoje a počítalo by se s takto upravenou soustavou. Dalším řešením bylo přímo v aplikaci, aproximovat zadané dopravní zpoždění, kdy při známé velikosti dopravního zpoždění T_d a známé periodě vzorkování T , byla soustava, jelikož výpočet probíhá v diskrétním tvaru, zpožděna o z - k kroků a k by muselo být celé číslo. Druhá možnost byla realizována a diskrétní stavový popis se změnil na tvar

$$\mathbf{x}[(k+1)T] = \mathbf{F}\mathbf{x}(kT) + \mathbf{G}u[(k-nd)T] \quad (15)$$

$$y[kT] = \mathbf{C}\mathbf{x}(kT)$$

, kde nd je výsledek po dělení dopravního zpoždění a vzorkovací periody.

$$nd = \frac{T_d}{T} \quad (16)$$

Pro názornost uvádím příklad pro $nd=3$, kde je vidět jak aproximace funguje.

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & g_1 & 0 & 0 \\ f_{21} & f_{22} & f_{23} & g_2 & 0 & 0 \\ f_{31} & f_{32} & f_{33} & g_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

$$\mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C} = [1 \ 0 \ 0 \ 0 \ 0 \ 0]$$

Tímto způsobem je pracováno s každým přenosem v matici \mathbf{G} a je převeden na diskretní stavový popis. Vytvořené dílčí matice \mathbf{A} , \mathbf{B} , \mathbf{C} stavových popisů archivují v paměti a následně sestavují výsledný stavový popis systému popsaného přenosovou maticí \mathbf{G} mnohorozměrného systému. Postup sestavení výsledných matic \mathbf{A}_c , \mathbf{B}_c a \mathbf{C}_c je patrný v následujících vzorcích.

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{F}_{11} & & & & & \\ & \ddots & & & & \\ & & \mathbf{F}_{1m} & & & \\ & & \mathbf{F}_{21} & & & \\ & & & \ddots & & \\ & & & & \mathbf{F}_{pm} & \end{bmatrix} \quad (18)$$

$$\mathbf{B}_c = \begin{bmatrix} \mathbf{G}_{11} & & & & \\ & \ddots & & & \\ & & \mathbf{G}_{1m} & & \\ \mathbf{G}_{21} & & & & \\ & & & \ddots & \\ & & & & \mathbf{G}_{pm} \end{bmatrix}$$

$$\mathbf{C}_c = \begin{bmatrix} \mathbf{C}_{11} \dots \mathbf{C}_{1m} & & & & \\ & \mathbf{C}_{21} \dots \mathbf{C}_{2m} & & & \\ & & \ddots & & \\ & & & \mathbf{C}_{p1} \dots \mathbf{C}_{pm} & \end{bmatrix}$$

Celkový diskretní stavový popis systému pak je ve tvaru:

$$\mathbf{x}[(k+1)T] = \mathbf{A}_c \mathbf{x}(kT) + \mathbf{B}_c \mathbf{u}(kT) \quad (19)$$

$$\mathbf{y}[kT] = \mathbf{C}_c \mathbf{x}(kT)$$

8.2 Realizace algoritmu pro PLC Bachmann MPC200

Ve Vídni jsem pracoval s PLC řady MPC200 od firmy Bachmann, ve kterém běží real-time operační systém VxWorks podporující multitasking.

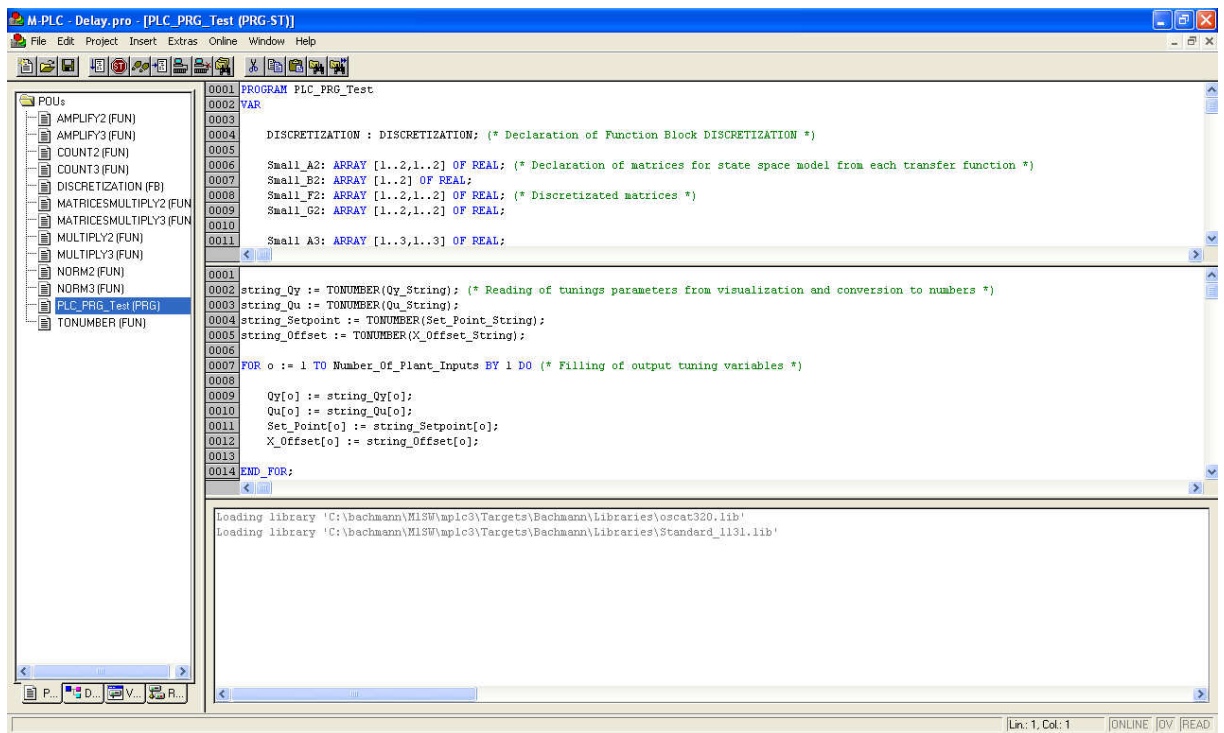


Obr. 44 Procesorová jednotka MPC240

Technické parametry PLC:

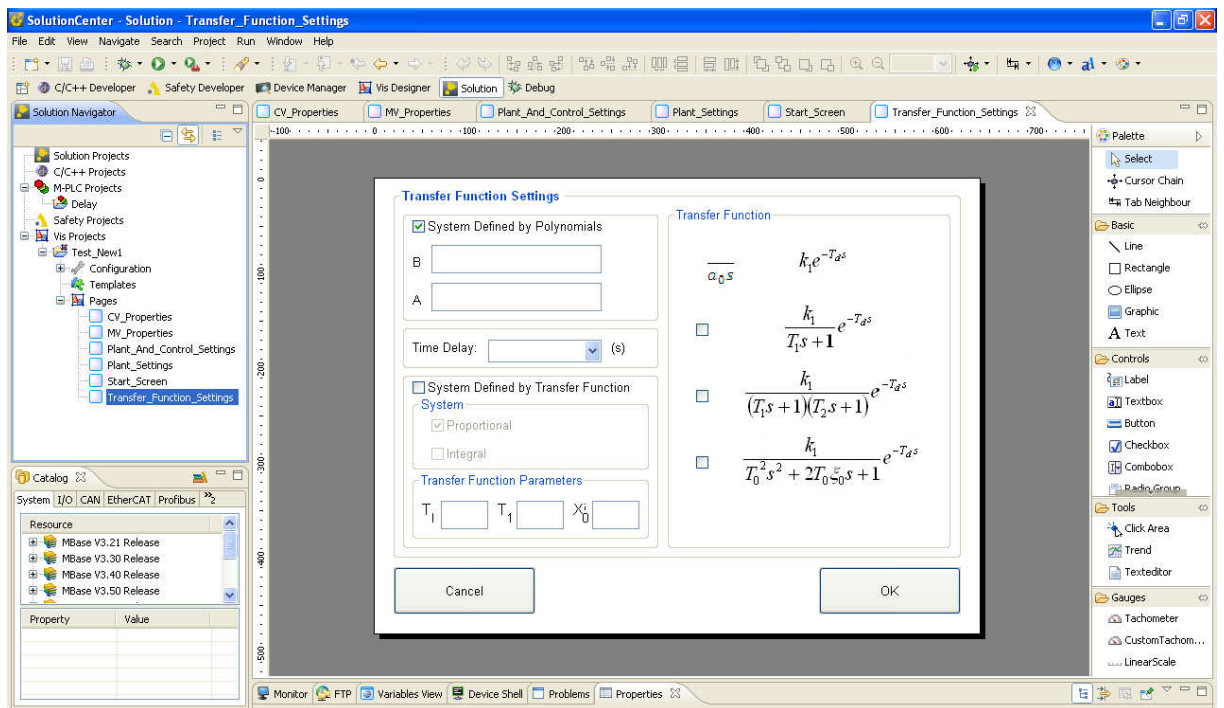
- CPU modul: 700 MHz Pentium class
- SRAM: 512 kB (baterií zálohovaná)
- DRAM: 128 MB,
- FLASH: 8 MB interní
- FRAM: 64 kB
- 1x USB, 1x RS232, 1x RS232/422/485, 2x Ethernet 10 / 100 Mbit / s

Pro napsání aplikace byl použit výše uvedený software CoDeSys, který bohužel nedisponuje knihovnamí pro práci s maticemi, takže prvním krokem bylo napsání funkcí pro sčítání a násobení matic, dále pro výpočet normy matice. Pro načtení koeficientů přenosů z vizualizační obrazovky jsem musel napsat funkci pro převod množiny čísel ve formě textového řetězce na jednotlivé číselné koeficienty. Dále jsem vytvořil nový funkční blok, který realizoval diskretizaci stavového popisu. Celý vytvořený program je zobrazen na obrazovce s programem CoDeSys na obr.45.



8.3 Vytvoření vizualizace v programu SolutionCenter

Druhým úkolem bylo vytvořit vizualizaci v programu SolutionCenter, přes kterou se budou vkládat koeficienty jednotlivých přenosů a parametrů pro prediktivní řízení. Na obr. 46 je vidět pracovní plocha programu SolutionCenter, který jsem použil pro tvorbu obrazovek. Tento systém je od firmy Bachmann stejně jako použité PLC a patří do kategorie programů SCADA/HMI a je optimalizovaný právě pro řízení, nastavování a ovládání produktů této firmy.



Obr. 46 Pracovní prostředí SolutionCenter

Jednotlivé obrazovky vytvořené pro nastavování prediktivního regulátoru můžete vidět na následujících obrázcích.



Obr. 47 Vstupní obrazovka

Plant and Controller Settings

Execution Interval: (Seconds) Set-point:

Prediction Horizon: (Executions) X offset:

Control Horizon: (Executions) Qy:

Number of Plant Inputs: (-) Qu:

Set Up Plant Read Matrices Inicialization

Cancel Save Matrices OK

Obr. 48 Obrazovka pro nastavení parametrů řízení

Plant Settings

	MV1	MV2	MV3	MV4	MV5	MV6
CV1	G 1,1	G 1,2	G 1,3	G 1,4	G 1,5	G 1,6
CV2	G 2,1	G 2,2	G 2,3	G 2,4	G 2,5	G 2,6
CV3	G 3,1	G 3,2	G 3,3	G 3,4	G 3,5	G 3,6
CV4	G 4,1	G 4,2	G 4,3	G 4,4	G 4,5	G 4,6
CV5	G 5,1	G 5,2	G 5,3	G 5,4	G 5,5	G 5,6
CV6	G 6,1	G 6,2	G 6,3	G 6,4	G 6,5	G 6,6

Reset

OK

Obr. 49 Vstup do obrazovky pro zadávání přenosů

Properties

Name:

MV Range

Maximum: (-)

Minimum: (-)

Cancel OK

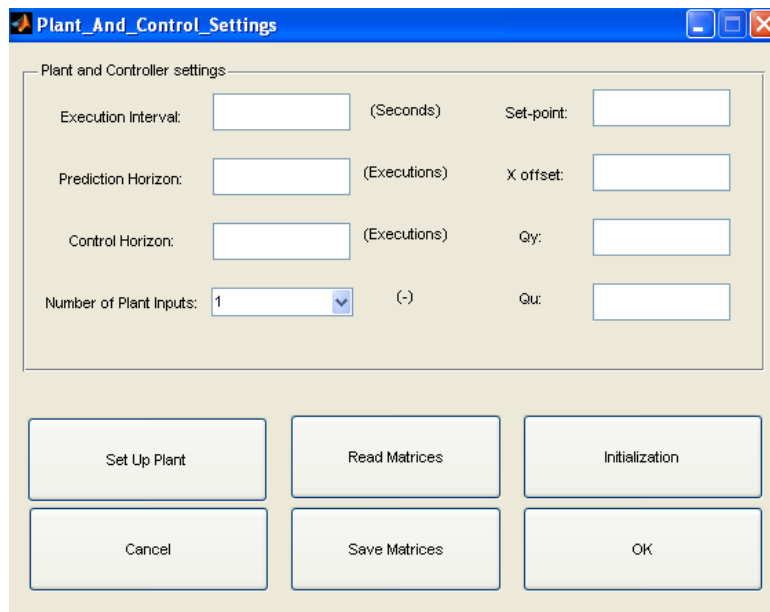
Obr. 50 Obrazovka pro zadávání rozsahu výstupní veličiny

Obr. 51 Obrazovka zadávání přenosů

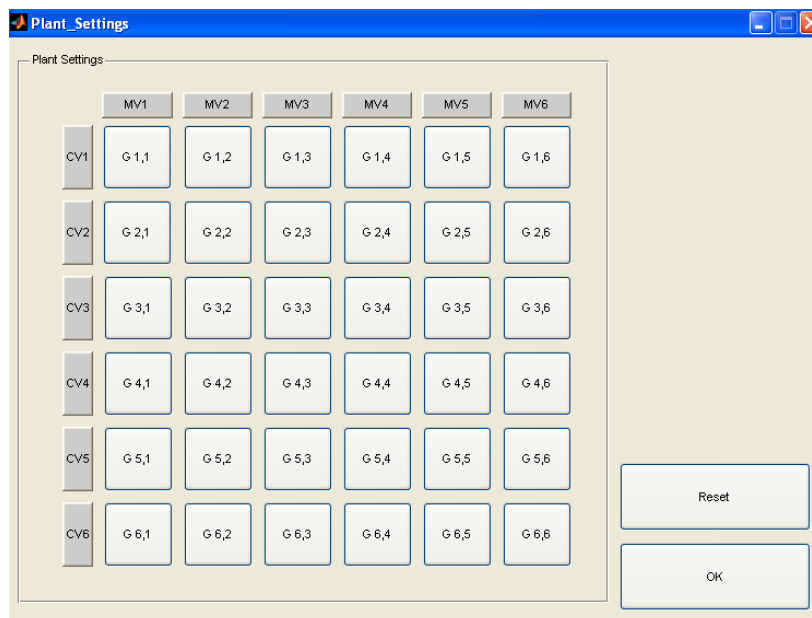
Všechny matematické operace popsané výše jsou zahrnuty v algoritmu vytvořené aplikace, nahrány do PLC a komunikují s vizualizačním prostředím. Výstupní data z mé aplikace používá dále i jiná aplikace vytvořená spolupracovníkem v novém standardu IEC1499. V této aplikaci běží algoritmus prediktivního řízení. Výhodou tohoto nového standardu je, že aplikace nahrané v PLC mohou být napsány i ve starším standardu IEC 1131, běží na jednom PLC a mohou si jednoduše předávat data mezi sebou.

8.4 Aplikace algoritmů v MATLABu

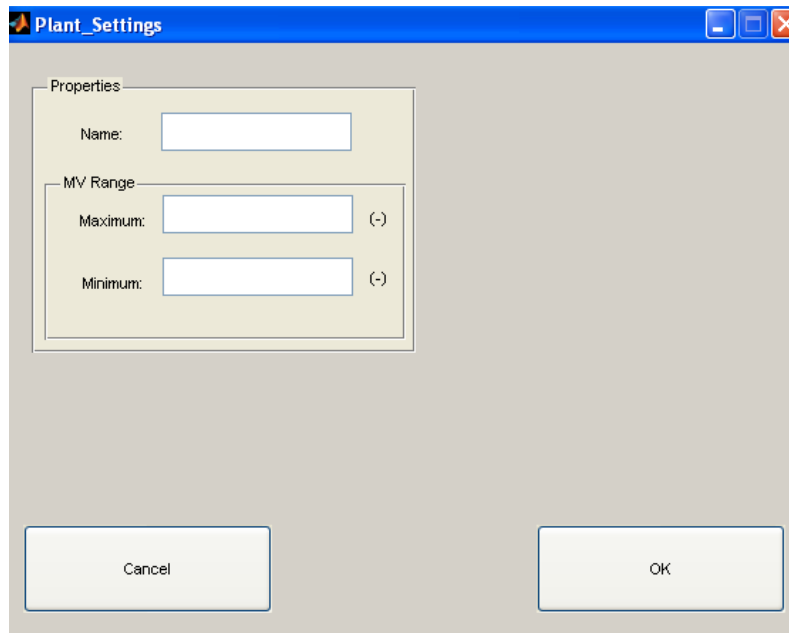
Jelikož PLC od firmy Bachmann a používané softwarové vybavení, na kterém byla tato úloha realizována, se nenachází v laboratořích katedry 352, není možnost dále s vytvořeným programem pracovat. Taktéž nelze použít kód určený pro PLC, jelikož nemám k dispozici používané knihovny. Proto jsem zvolil cestu přepsání mého algoritmu a algoritmu prediktivního řízení pro program Matlab. Dále jsem v tomto programu vytvořil podobné ovládací obrazovky jako dříve používané z programu SolutionCenter viz obr. 52 až 55. Obrazovka pro vkládání přenosů je navíc obohacena o zobrazení základních charakteristik vkládaného přenosu.



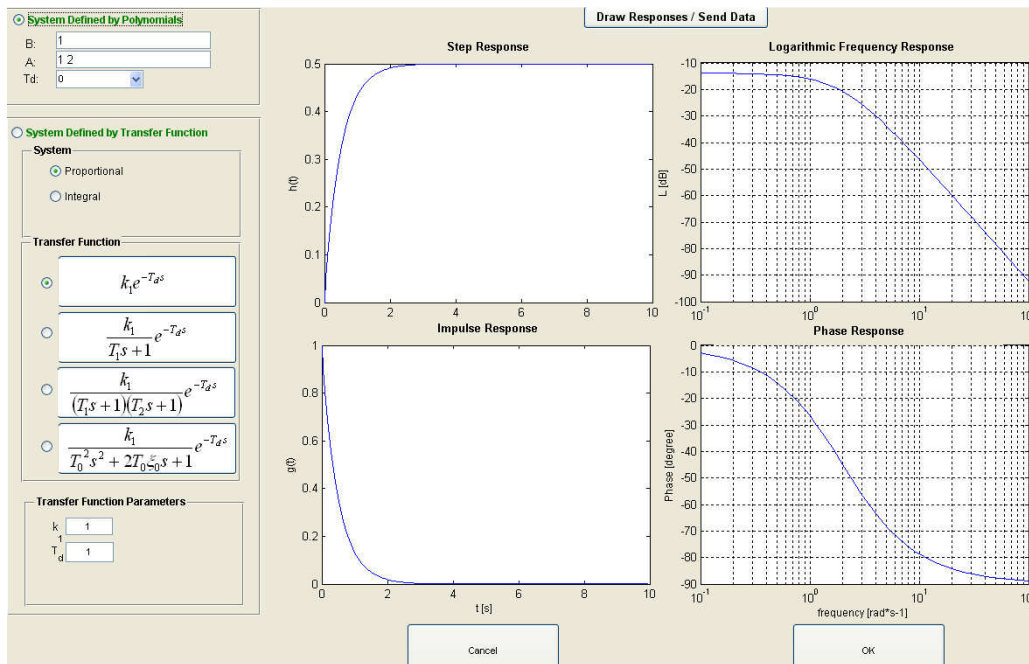
Obr. 52 Obrazovka pro nastavení parametrů řízení



Obr. 53 Vstup do obrazovky pro zadávání přenosů



Obr. 54 Obrazovka pro zadávání rozsahu výstupní veličiny

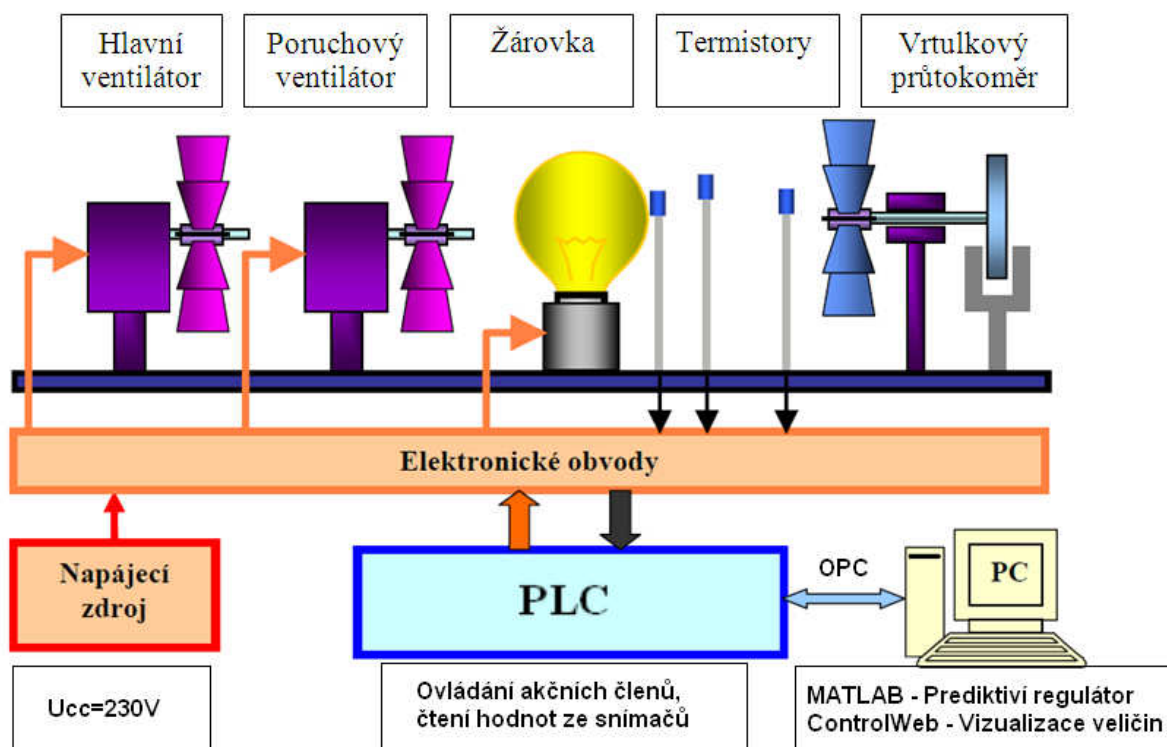


Obr. 55 Obrazovka zadávání přenosů

Tyto obrazovky mi slouží k načítání parametrů pro prediktivní regulátor. Používaný algoritmus prediktivního regulátoru vychází z knihy Model Predictive Control System Design and Implementation Using MATLAB od Liuping Wanga s jedinou modifikací, kdy pro výpočet optimálního řízení se používá místo funkce quadprog z Optimization Toolboxu funkce qpOASES z volně šiřitelné knihovny, která používá tuto formu zápisu funkcionálu:

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^T H + x^T g(w_0) \\ \text{s. t.} \quad & lbA(w_0) \leq Ax \leq ubA(w_0) \\ & lb(w_0) \leq x \leq ub(w_0) \end{aligned} \quad (20)$$

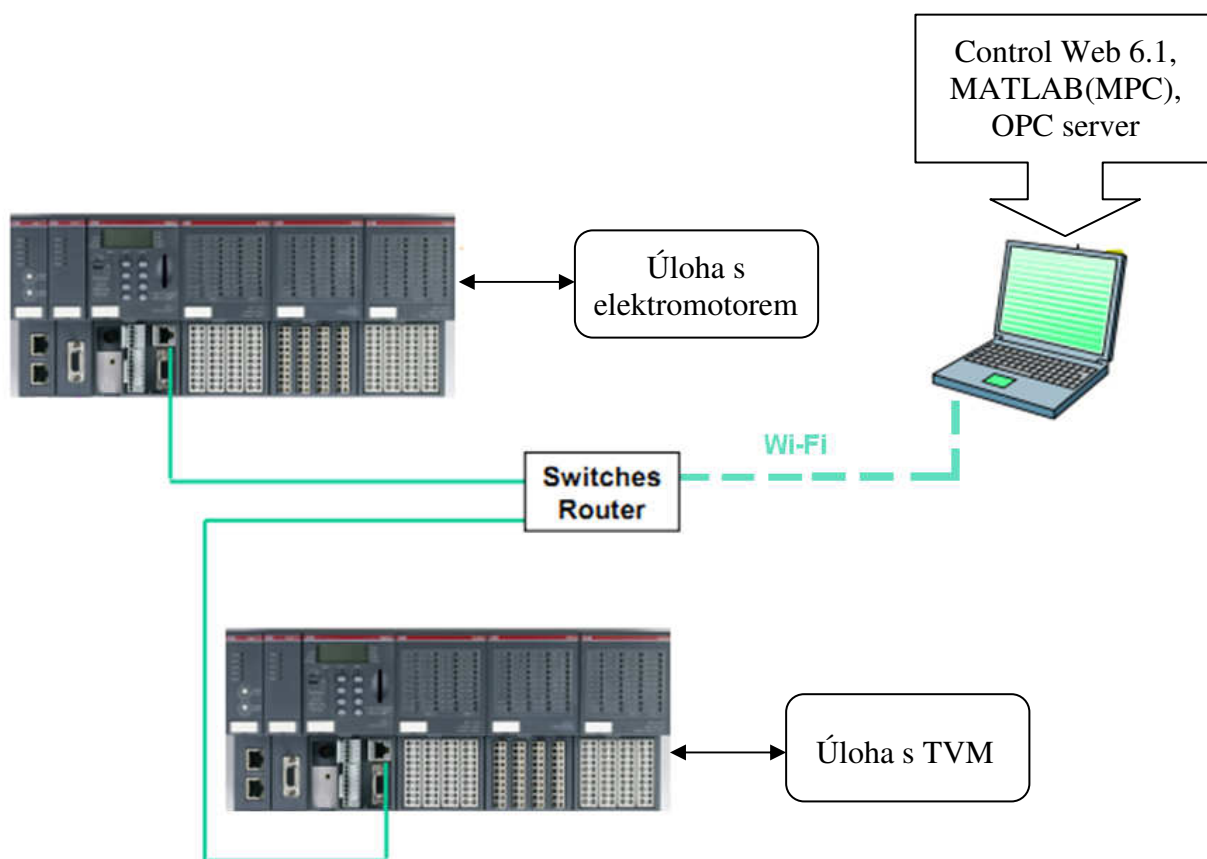
Tento prediktivní regulátor je použit pro řízení mnohorozměrné soustavy, v našem případě teplovzdušného modelu tak, jak ukazuje obr. 56. Vypočtené akční zásahy regulátoru jsou okamžitě přes OPC server poslány do PLC, které nastaví příslušné hodnoty akčním členům, ventilátoru a žárovce. Zpětná vazba pro prediktivní regulátor je realizována opět přes OPC komunikaci, kdy jsou snímány hodnoty průtoku vzduchu a teploty uvnitř laboratorního modelu.



Obr. 56 Aplikace prediktivního regulátoru na teplovzdušný model

9 VÝSLEDNÉ PROPOJENÍ ÚLOH

V každé laboratorní úloze je použito jedno PLC z řady AC500 od firmy ABB, ty jsou dále připojeny přes switch a Wi-Fi router do bezdrátové sítě v učebně H303 viz obr. 57. Každé PLC má svou IP adresu a je v něm spuštěna aplikace určená dané úloze. Obě realizované laboratorní úlohy lze řídit z jednoho počítače, který je připojen do Wi-Fi sítě v učebně a na kterém je spuštěna řídicí aplikace v programu Control Web 6.1. Obě úlohy je možno řídit nezávisle na sobě i obě najednou. Hodnoty proměnných si PLC a Control Web vyměňují pomocí OPC serveru. Pokud při řízení laboratorní úlohy s teplotovzdušným modelem přepneme do režimu MPC a na daném počítači spustíme také MATLAB a m-soubory realizující prediktivní regulátor, tak v úloze s teplotovzdušným modelem je možné realizovat také prediktivní řízení mnohorozměrné soustavy. Nejprve ale musíme znát případně určit matematický popis řízené soustavy a zadat jej přes vstupní obrazovky do tohoto regulátoru. Stejně tak musíme zadat parametry regulace a průběhy žádaných veličin.



Obr. 57 Výsledné propojení úloh

10 ZÁVĚR

Výsledkem diplomové práce je realizace řízení dvou laboratorních úloh pomocí programovatelných logických automatů (PLC). Dále vytvoření jednotné vizualizace v programu Control Web 6.1, z které je možno ovládat obě laboratorní úlohy, sledovat měřené veličiny, archivovat naměřená data a ke každé úloze je napsán i krátký help zobrazitelný ve vizualizaci. Ke konci práce byly naprogramovány aplikace pro realizaci prediktivního řízení za pomoci programu MATLAB a programu CoDeSys, který je prostředím pro programování logických automatů.

V první části diplomové práce je popsán použitý vizualizační a řídicí software Control Web 6.1, jeho pracovní prostředí a jeho komunikační a aplikační možnosti. Dále jsou zde uvedeny použité programovatelné logické automaty řady AC500 od firmy ABB a popsáno prostředí programu CoDeSys používané pro vývoj aplikací pro tyto automaty.

V první realizované úloze byl stávající laboratorní model elektromotoru s frekvenčním měničem rozšířen o zátěžový elektromotor, který vykonává funkci brzdy s regulovatelným brzdícím účinkem nebo také funkci poruchové veličiny, kdy žádanou veličinou jsou otáčky hlavního elektromotoru. Realizace PID regulátoru je provedena v programu v PLC a nastavení jeho parametrů probíhá přes vizualizační obrazovku. Pro komunikaci PLC s laboratorním modelem byly vytvořeny unifikační obvody, které jsem umístil do černé skříně. Pro potřeby napájení zátěžového elektromotoru jsem realizoval regulovatelný napěťový zdroj, jehož základem je stabilizátor napětí LM350, který je také umístěn v unifikačním modulu. Navržené schéma a deska jsou umístěny v příloze.

V druhé použité laboratorní úloze s teplovzdušným modelem byla CTRL jednotka nahrazena programovatelným logickým automatem od firmy ABB. Tato laboratorní úloha má dvě akční veličiny napětí na žárovce a napětí na ventilátoru a dvě výstupní veličiny a to napětí na snímači otáček a napětí na snímači teploty. V této úloze je možno řídit tyto akční členy dvěma možnými způsoby a to buď PID regulací, nebo dvoupolohovou regulací. Parametry regulátorů se zadávají z vizualizačního prostředí a samotné regulátory jsou realizovány v PLC. Můžeme zde také simulovat poruchovou veličinu pomocí poruchového ventilátoru a to buď manuálně, nebo nastavit trojúhelníkový či obdélníkový průběh poruchové veličiny. Dále mi tato úloha posloužila jako mnohorozměrná soustava pro realizaci prediktivního řízení.

Během zimního semestru 2012/2013 jsem absolvoval pracovní stáž ve Vídni ve firmě VOIGT+WIPP Engineers GmbH, kde jsem se zúčastnil projektu vývoje prediktivního

algoritmu pro PLC na bázi PC. Mým úkolem bylo vytvořit v programu CoDeSys aplikaci pro PLC ve standardu IEC 1131 určenou pro převod matice přenosů mnohorozměrného systému s maximálním počtem šesti vstupních a šesti výstupních veličin na stavový popis s následnou diskretizací tohoto stavového popisu. Druhým úkolem bylo vytvořit vizualizaci ve vizualizačním softwaru SolutionCenter od firmy Bachmann, přes kterou se budou vkládat data pro prediktivní řízení. Pro napsání aplikace byl použit výše uvedený software CoDeSys, který bohužel nedisponuje knihovnamy pro práci s maticemi, takže nezbytným krokem bylo napsání funkcí pro sčítání a násobení matic a také pro výpočet normy matice. Pro načtení koeficientů přenosů z vizualizační obrazovky jsem napsal funkci pro převod množiny čísel ve formě textového řetězce na jednotlivé číselné koeficienty přenosů. Dále jsem vytvořil nový funkční blok, který realizoval diskretizaci stavového popisu. Jelikož PLC od firmy Bachmann a používané softwarové vybavení, na kterém byla tato úloha realizována, se nenachází v laboratořích katedry 352, není možnost dále s vytvořeným programem pracovat. Taktéž nelze použít kód určený pro PLC, jelikož nemám k dispozici používané knihovny. Proto jsem zvolil cestu přepsání mého algoritmu a algoritmu prediktivního řízení pro program MATLAB. Dále jsem v tomto programu vytvořil podobné vstupní obrazovky pro zadávání modelu a parametrů prediktivního regulátoru jako v programu SolutionCenter. Algoritmus prediktivního řízení běží v programu MATLAB a vypočtené optimální akční zásahy jsou pomocí PLC nastavovány na akční členy teplovzdušného modelu.

SEZNAM POUŽITÉ LITERATURY

- BALÁTĚ, J. *Automatické řízení*. 1. vydání. Praha: BEN, 2003. 664 s. ISBN 80-7300-020-2.
- BÍLÝ, Radek, et al. *Control Web 2000 : Průvodce systémem pro tvorbu a nasazení aplikací reálného času*. Praha : Computer press, 1999. 382 s, ISBN 80-7226-258-0.
- BUCHTA, Richard. *Modulární programovatelný automat AC500*. Automa. 2007, 03, s.1. Dostupný také z WWW: <www.odbornecasopisy.cz>.
- CoDeSys Service Tool CST* [online]. 20013 [cit. 2012-10-07]. Dostupné z WWW: <<http://www.3s-software.com>>.
- Control Web 6. Manuál*. Moravské přístroje, a.s. [online]. [cit. 20013-15-05]. Dostupný z WWW: <<http://www.mii.cz/>>.
- KADLEC, David. *Návrh a realizace stavových pozorovatelů*. Ostrava: Katedra automatizační techniky a řízení, Fakulta strojní, VŠB – Technická univerzita Ostrava, 2011, 72s. Diplomová práce, Vedoucí: doc. Ing. Wagnerová R., Ph.D.
- KULHÁNEK, Jiří. *Počítače a sítě* [online]. První. Ostrava : Editační středisko VŠB-TUO, 2007 [cit. 2010-05-12]. Dostupné z WWW: <http://vyuka.fs.vsb.cz/file.php/6/Skripta_Pocitace_a_site.pdf>. ISBN 978-80-248-1516-9.
- LEWIS, Robert. *Modelling control systems using IEC 61499*, 1. vyd., Londýn: Institution of Electrical Engineers, ISBN 978-0-85296-796-6.
- WANG, Liuping. *Model Predictive Control System Design and Implementation Using MATLAB*. Londýn: Springer 2009, 371s, ISBN 978-1-84882-330-3.
- NACHTIGAL CH. L. *Instrumentation and Control - Fundamentals and Applications*. New York : John Wiley & Sons, Inc. 1993.
- NAVRÁTIL, Pavel. *S počítačem nejen k maturitě: 1.díl*. PhDr. Dagmar Procházková; Ing. Michal Jiříček, Pavel Navrátil. 5 : Computer Media s.r.o., 2004. 175 s. ISBN 80-86686-19-1.
- SMUTNÝ, Lubomír. *Počítačová podpora zpracování a prezentace experimentálních dat* [online]. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 1999. Dostupný z www: <<http://www.fs.vsb.cz/akce/1999/ASR99/Proceedings/papers/3/3.htm>>

SMUTNÝ, Lubomír, et al. *Teplovzdušný model TVM (klimatizační jednotka)*. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 1999., 2005. 46 s. Závěrečná zpráva projektu GAČR 102/03/0628.

STREJC, V. 1978. *Stavová teorie lineárního diskrétního řízení*. Praha: Academia, 1978

ŠKUTA, Jaromír., SMUTNÝ, Lubomír. *Popis TVM 2005* [online]. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2005. Dostupný z [www:<http://352lab.vsb.cz/ServerFinalVer/PodklVyuka/PopisTVM.htm>](http://352lab.vsb.cz/ServerFinalVer/PodklVyuka/PopisTVM.htm)

ŠTEPÁN, Martin. *Využití PLC pro řízení části výrobní linky*: bakalářská práce. Ostrava: Katedra automatizační techniky a řízení, Fakulta strojní VSB-Technická univerzita Ostrava, 2011, 51s. Vedoucí práce: Škuta, J.

ŠVARC, Ivan, ŠEDA, Miloš, VÍTEČKOVÁ Miluše. *Automatické řízení*, 1. vyd. Brno: AKADEMICKÉ NAKLADATELSTVÍ CERM, 2007, 324s, ISBN 978-80-214-3491-2.

VÍTEČKOVÁ, Miluše., VÍTEČEK, Antonín. 2006. *Základy automatické regulace*. 1. vyd. Ostrava: katedra automatizační techniky a řízení, VŠB – TU Ostrava, 2006. 200s. ISBN 80-248-1068-9

VLACH, Jaroslav. *Počítačová rozhraní : Přenos dat a řídicí systémy*. 2.rozšířené vydání. Praha : BEN, 2000. 176 s. ISBN 80-7300-010-5.

VLACH, Jaroslav. *Řízení a Vizualizace technologických procesů*. Praha : BEN, 1999.160 s. ISBN 80-86056-66-X.

SEZNAM PŘÍLOH

Příloha A Schéma unifikační desky	s. 32
Příloha B Navržená deska s unifikačními obvody	s. 32
Příloha C Ukázka zdrojového kódu	s. 44
Příloha D CD s prací v elektronické podobě	

Příloha A SCHÉMA UNIFIKAČNÍ DESKY

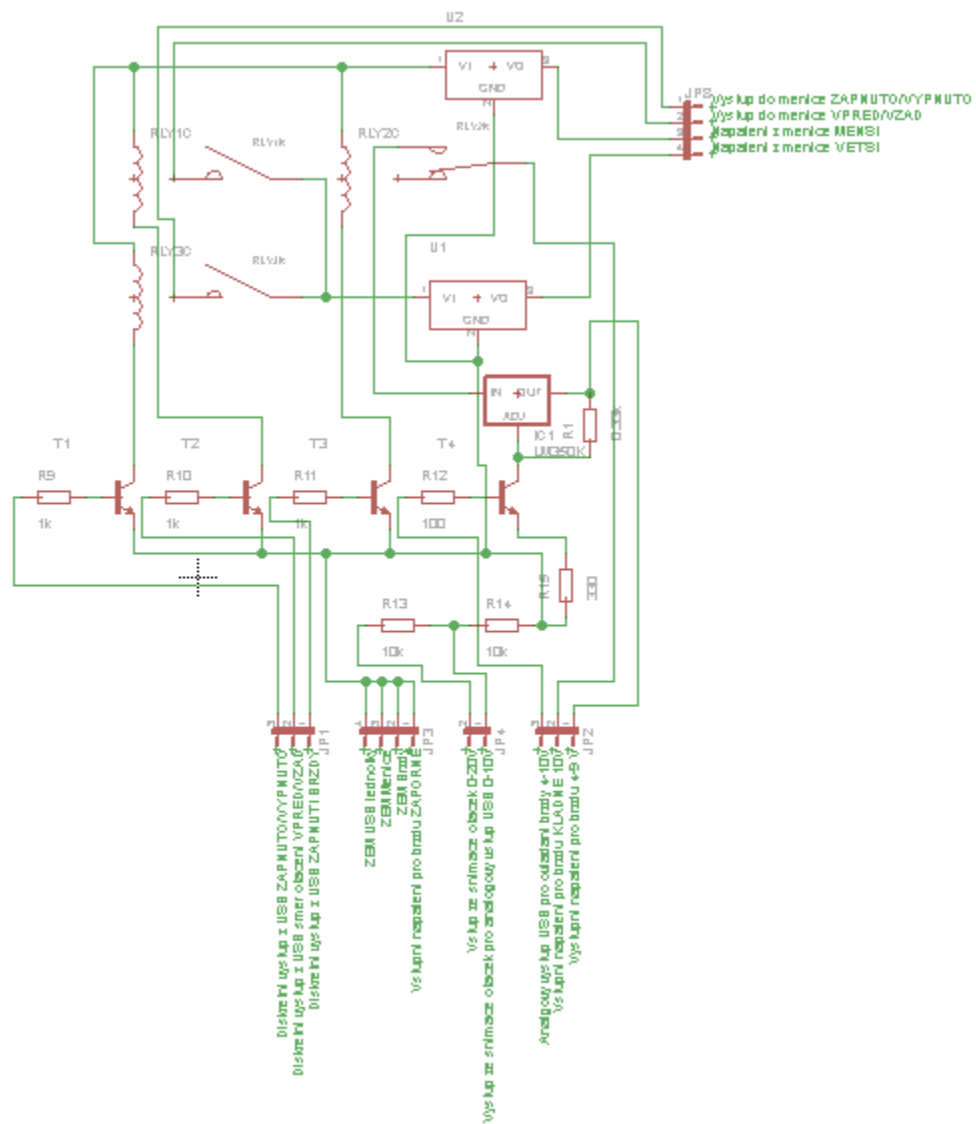
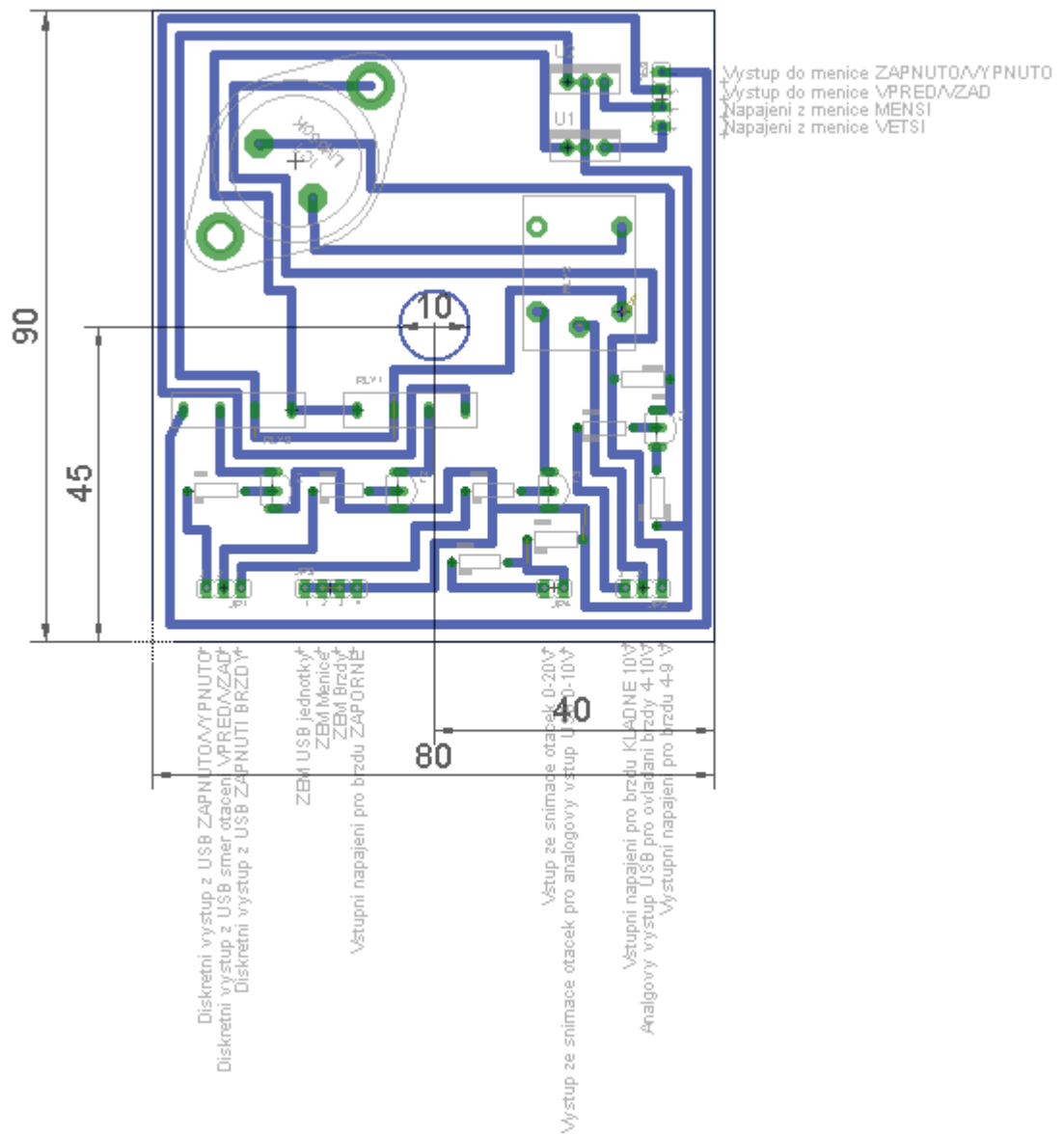


Schéma unifikační desky

Příloha B NAVRŽENÁ DESKA S UNIFIKAČNÍMI OBVODY



Navržená deska s unifikačními obvody

Příloha C UKÁZKA ZDROJOVÉHO KÓDU

(Diskretizace stavového popisu, kde rozměr matice A je 3x3)


```

FOR o := 1 TO 3 BY 1 DO (*Creation of Unit matrix*)
    I3[o, o] := 1;
END_FOR;

x := 1; faktorial := 1; normA := 1;

(*Calculation of first element in Taylor serie*)
Ih3 := MULTIPLY3(I3,h);

(*Accurancy of calculation by matrix norm, variable number of cycles*)
WHILE normA > Err DO

    faktorial := faktorial * (x+1); (*Calculation of factorial*)
    AA3 := MULTIPLY3(AMPLIFY3(A3,x),EXPT(h,x+1)/faktorial);
    (*Calculation of next element*)
    Ih3 := COUNT3(Ih3,AA3); (*Calculation of Taylor approximation*)
    x := x + 1;
    normA := NORM3(AA3); (*Calculation of matrix norm*)

END_WHILE;

(*Transformation matrix is the same like Taylor approximation*)
V3 := Ih3;
(*Calculation of discrete matrix A*)
F3 := COUNT3(I3,MATRICESMULTIPLY3(A3,V3));
(*IN MY FUNCTIONS I CAN WORK JUST WITH SQUARE MATRICES*)

FOR o := 1 TO 3 BY 1 DO (*Transfer matrix B to square matrix B2*)
    B223[o,1]:=B3[o];
END_FOR;

(*Calculation of discrete square matrix B(G2)*)
G3 := MATRICESMULTIPLY3(V3,B223);

```