

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA
STROJNÍ**



**DIPLOMOVÁ
PRÁCE**

2020

**BC. MARTIN
CAHYNA**

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Cahyna** Jméno: **Martin** Osobní číslo: **419591**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Strojní inženýrství**
Studijní obor: **Přístrojová a řídicí technika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Jednotné uživatelské rozhraní pro laboratoř automatizace

Název diplomové práce anglicky:

Unified HMI for the automatic control laboratory tasks

Pokyny pro vypracování:

1. Navrhněte a realizujte SCADA jednotné vizualizační prostředí pro ovládání laboratorních úloh předmětu Automatické řízení v laboratoři automatizace.
2. Sestavte program pro PLC v jazyce dle normy IEC/EN 61131-3 umožňující kooperaci s vizualizačním SCADA systémem.
3. Požadované funkce programu jsou:
 - a) Řízení v uzavřeném regulačním obvodu.
 - b) Manuální řízení v otevřeném obvodu.
 - c) Generování automatického průběhu různých typů vstupního signálu na základě zadaných parametrů.
4. Otestujte kompletní řešení na reálném laboratorním modelu.

Seznam doporučené literatury:

- [1] Šrámek, Filip. Vzdálené řízení PLC SCADA systémem pomocí protokolu MODBUS. Bakalářská práce. Praha: ČVUT v Praze, 2018.
- [2] IEC 61131-3:2013 Programmable controllers - Part 3: Programming languages.
- [3] Riedl, Jan, Bc. Zařízení pro konkurenční ovládání laboratorního modelu. Diplomová práce. Praha: ČVUT v Praze, 2019.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Pavel Trnka, Ph.D., U12110.3

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **31.10.2019**

Termín odevzdání diplomové práce: **17.01.2020**

Platnost zadání diplomové práce: _____

Ing. Pavel Trnka, Ph.D.
podpis vedoucí(ho) práce

_____ podpis vedoucí(ho) ústavu/katedry

prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

_____ Datum převzetí zadání

_____ Podpis studenta



Poděkování

Rád bych poděkoval Ing. Pavlu Trnkovi, Ph.D. za vedení diplomové práce, věcné připomínky a volnost při testování v laboratoři. Dále mé poděkování patří Ing. Cyrilu Oswaldovi, Ph.D. za pomoc s nastavením serveru mySCADA. V neposlední řadě chci poděkovat mé rodině, přátelům a spolužákům, že mi byli nablízku během celé doby mého studia. Ze spolužáků jmenovitě hlavně Jan Riedl, Nikita Mazurenko a Jaroslav Petráš.



Anotace

Tato diplomová práce se zabývá vytvořením unifikovaného prostředí pro ovládání laboratorních úloh. Pro realizaci je použito PLC Foxtrot a SCADA server od firmy mySCADA. Cílem je program na řízení v uzavřeném regulačním obvodu, otevřeném regulačním obvodu a generování automatického průběhu signálu na základě zadaných parametrů.

Klíčová slova

SCADA, Generátor signálu, MODBUS

Abstract

This diploma thesis deals with creation of unified environment for laboratory tasks control. PLC Foxtrot and SCADA server from mySCADA are used for realization. The goal is to control a closed loop control system, open control circuit, and generate an automatic waveform based on the specified parameters.

Keywords

SCADA, Signal generator, MODBUS



1. Obsah

1.	Obsah	4
2.	Úvod.....	6
3.	PLC.....	7
3.1.	PLC Foxtrot CP-1013 [1]	7
3.2.	Siemens S7-1200 [2].....	8
3.3.	PLC Foxtrot CP-1015 [3]	9
3.4.	Výběr finálního PLC	9
4.	SCADA.....	10
4.1.	Reliance [4]	10
4.2.	mySCADA [5]	15
4.3.	Zvolené řešení	17
5.	Komunikace.....	18
5.1.	MODBUS [7]	18
5.2.	Programování PLC [8]	19
5.3.	Nahrávání projektu mySCADA.....	19
6.	Program pro PLC	20
6.1.	Testovací PLC	20
6.2.	Programovací jazyk	20
6.3.	Komunikace MODBUS	20
6.4.	Nastavení chanCode	22
6.5.	Čtení a zápis proměnných	23
6.6.	Nastavení analogových vstupů a výstupů	24
6.7.	PID regulace	25
6.8.	Generátor signálu.....	26
6.9.	Manuální vstup.....	28
6.10.	Výhybka pro LabJack	28
6.11.	Logika přepínání	29
6.12.	Přehled tabulek MODBUS	29
7.	mySCADA Projekt	30
7.1.	Vytvoření projektu	30



7.2. Test spojení.....	30
7.3. Pohled LabJack.....	31
7.4. Pohled URO	31
7.5. Pohled Diagnostický.....	34
7.6. Pohled volby režimu.....	35
7.7. Úvodní obrazovka	35
7.8. Možnosti spojení serveru s PLC.....	36
8. Závěr.....	37
9. Bibliografie	38
Přílohy	40



2. Úvod

Cílem této práce je vytvořit funkční prostředí pro studenty předmětu Automatické řízení. V prostředí by mělo být nastavení regulátoru pro uzavřený regulační obvod, manuální řízení v otevřené smyčce a generátor vstupních signálů.

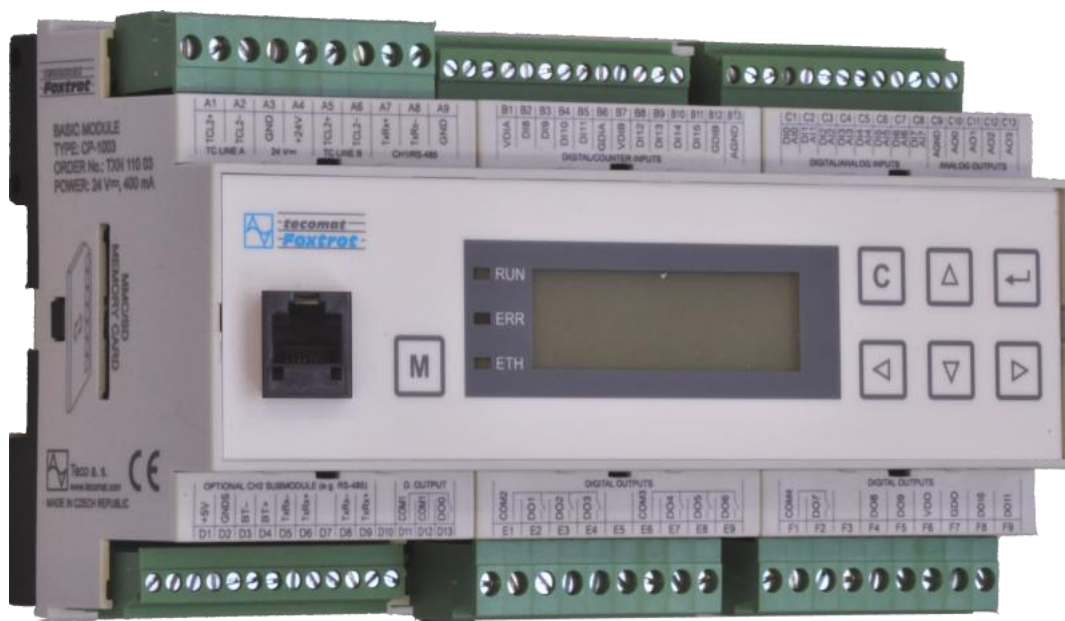
3. PLC

Základem každé laboratorní úlohy v laboratoři 111 je PLC. Je to malý průmyslový počítač používaný pro automatizaci procesů. Charakteristická vlastnost těchto počítačů je, že program se vykonává v cyklech. Počítač má pro komunikaci s vnějším světem (periferiemi) vstupní a výstupní moduly. Ty mohou být digitální nebo analogové. Zpravidla je PLC napájené stejnosměrným napětím 24V.

Tak jako je tomu i v jiných odvětvích, tak i zde je konkurenční prostředí. V laboratoři 111 je momentálně instalováno na laboratorních úlohách 8 PLC různých typů od 3 rozdílných firem. I když existují normalizované jazyky, jejich programovací prostředí se velmi liší. Studenti předmětu Automatické řízení se však o těchto programech neučí. Náplní předmětu Automatické řízení je teorie a praxe regulátorů, frekvenčních charakteristik atd. Z toho důvodu bylo rozhodnuto pro modernizaci stávajícího osazení laboratorních úloh novými PLC a v rámci této úpravy instalovat i jednotné prostředí pro budoucí přidání dalších úloh a jednoduchou obsluhu pro studenty mimo obor Ústavu přístrojové a řídicí techniky.

Pro modernizaci byli vybráni dva kandidáti od dvou různých firem, a to Siemens S7-1200 a Tecomat Foxtrot CP-1013.

3.1. PLC Foxtrot CP-1013 [1]



Obrázek 1 - Tecomat Foxtrot CP-1013 [1]

PLC Foxtrot CP-1013 je vybavené LCD displejem 4x20, šesti klávesami, 384KB paměti pro program a 192KB paměti pro proměnné. Zároveň je vybaveno slotem pro SD kartu. Pro komunikaci je zde vestavěný Ethernet port 100 Mbit/s a sériová linka RS-485. Dále je zde velké množství I/O výstupů:

- 4 vstupy x 2 skupiny binárních / čítačových vstupů DI8 – DI11, DI12 – DI15
- 8 vstupů analogových / binárních DI0/AI0 – DI7/AI7
- 4 binární tranzistorové výstupy DO8 – DO11
- 4 analogové výstupy AO0 – AO3
- 8 binárních reléových výstupů DO0 – DO7

Z uvedených vstupů/výstupů jsou nejpodstatnější tranzistorové výstupy, jelikož je možné pomocí nich přímo řídit DC motory nebo motory krokové s limitem do 100 kHz.

3.2. Siemens S7-1200 [2]



Obrázek 2 - Siemens SIMATIC S7-1200 [2]

Toto PLC má velké množství konfigurací pod stejným názvem, proto volím pro porovnání variantu nejvíce podobnou PLC Foxtrot CP-1013. Její produktové číslo je 6ES7217-1AG40-0XB0. Tato konfigurace má:

- 150 KB paměti pro program
- 10 digitálních vstupů DI
- 4 digitální vstupy pro sériovou linku RS422/RS485
- 6 digitálních výstupů DQ
- 4 digitální výstupy pro sériovou linku RS422/RS485
- 2 analogové vstupy 0 – 10V
- 2 analogové výstupy 0 – 20mA
- 2 porty PROFINET

3.3. PLC Foxtrot CP-1015 [3]



Obrázek 3 - Tecomat Foxtrot CP-1015 [3]

Toto PLC je zde uvedené, jelikož na něm probíhaly následné experimenty a testy komunikace se SCADA serverem. PLC má podobnou výbavu jako CP-1013. Liší se hlavně v tranzistorových výstupech, které toto PLC nemá.

PLC je vybavené šesti Analogovými vstupy, které lze přepnout do Digitálního režimu (24V). Dále 6 Digitálními výstupy a 2 Analogovými výstupy. Datová komunikace probíhá pomocí Ethernetu, a to konkrétně Ethernet 100/10. Další možností pro komunikaci je RS-232.

3.4. Výběr finálního PLC

Ač je finální výběr podřízen výběrovému řízení na Ústavu přístrojové a řídicí techniky, z mého pohledu by bylo dobré vybrat PLC Tecomat Foxtrot CP-1013. Jeho modul je sice dražší, ale není nutné pořizovat další dodatečné karty. PLC Siemens SYMATIC S7-1200 je velmi modulární. Výbava základního modulu je značně malá, takže je nutné dokoupit potřebné karty pro například analogové vstupy. To by mohlo vézt opět k rozdílnosti PLC na různých laboratorních úlohách.

4. SCADA

Supervisory Control And Data Acquisition, tak zní slova skrytá pod zkratkou SCADA. Přeloženo do českého jazyka to znamená „dispečerské řízení a sběr dat“.

Je to software pro monitorování a řízení průmyslových a technických zařízení a procesů. Aplikace tohoto softwaru je především ve strojní nebo procesní výrobě. Dále pak například v technických procesech v distribučních sítích (vody, plynu, elektřiny apod.). V poslední době proniká tento software a jeho technologie i do obyčejných domácností jako takzvaný „Smart Home“.

Principem SCADA je spojení mezi řídicím PLC výroby a hlavním serverem. Předávají si mezi sebou data pomocí spojení, které může být vedené například kroucenou dvojlínkou (Ethernet) nebo sériovou linkou (RS485).

Pro můj úkol bylo nutné vybrat výrobce tohoto software. Vybral jsem 2 hlavní kandidáty, kterými jsou mySCADA a Reliance.

Důvod výběru těchto kandidátů je jejich stabilita na trhu, velké zastoupení u firem v průmyslu a navázaná spolupráce s Fakultou Strojní ČVUT.

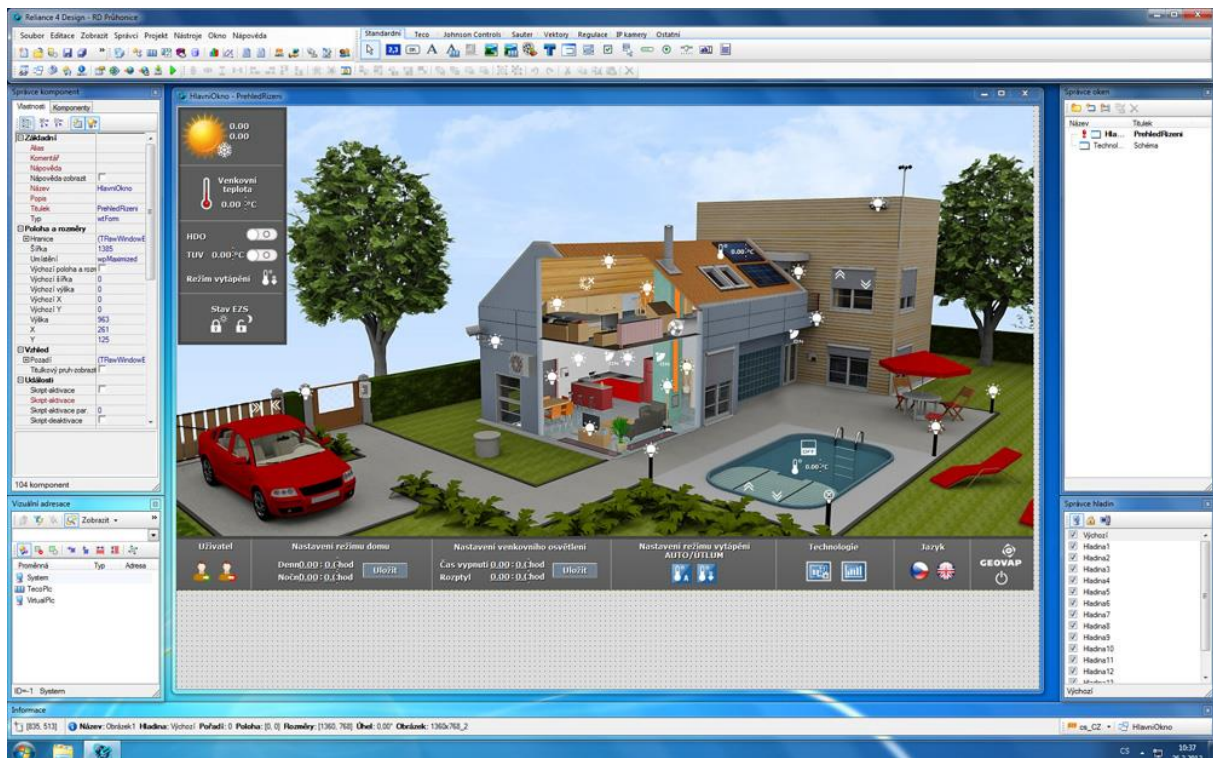
Zpravidla se SCADA software skládá ze tří částí, a to designéru, klienta a runtime modulu.

4.1. Reliance [4]

Prvním z kandidátů je Reliance 4 od firmy GEOVAP s.r.o. Je to software, který je na trhu již od roku 1997, kdy byla určena pro operační systém Windows NT 4.0.

Reliance Design

Reliance 4 Design je vývojové prostředí pro tvorbu a úpravu vizualizačního projektu. Existuje ve dvou verzích, a to Desktop a Enterprise. Verze desktop je určena pro tvorbu lokálních aplikací pro jeden počítač. Verze Enterprise umožňuje tvorbu síťových aplikací pro vizualizace v internetové prohlídce, na chytrém telefonu či tabletu.

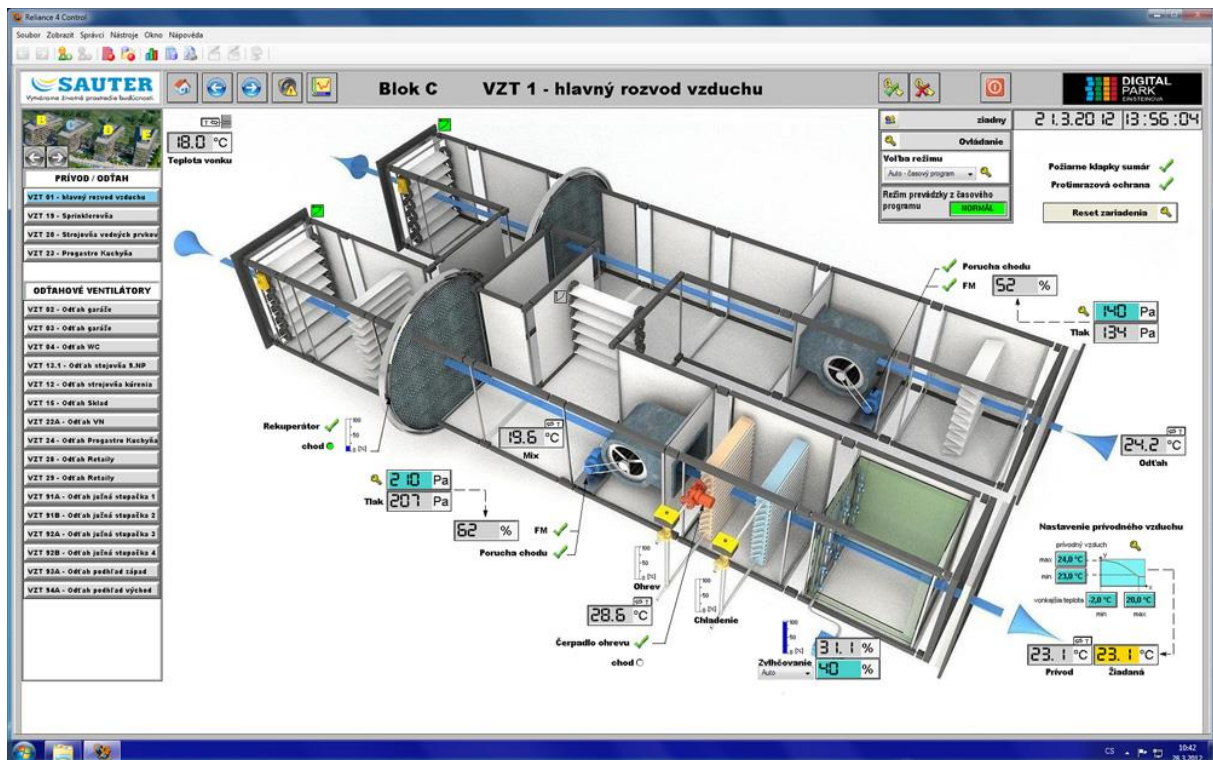


Obrázek 4 - Reliance 4 Design, Rodinný dům Průhonice

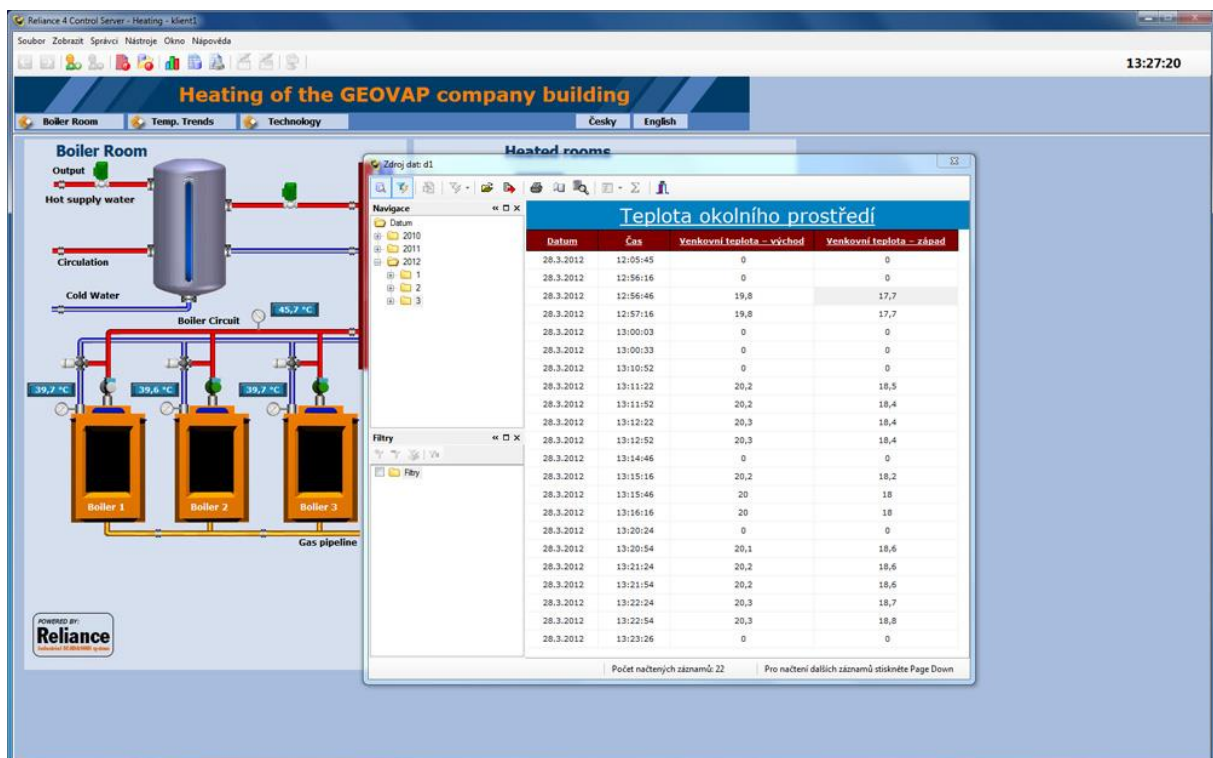
Runtime moduly

Runtime modulů je u softwaru Reliance velké množství, které se liší v licencích, doplňkových serverech (například datový server) a pokročilých funkcích.

- Reliance 4 Server**
 Je určen pouze jako webový server, samotná aplikace nedokáže zobrazit vizualizační okna. Běží jako Windows služba (Windows Service)
- Reliance 4 View**
 Je určen pro nahlížecí pracoviště, např. manažeři nebo vedoucí pracovníci. Umožňuje sledovat trendy a aktuální stav technologického procesu. Neumožňuje však do něj zasahovat a měnit parametry.
- Reliance 4 Control**
 Je to vylepšený modul Reliance 4 View, který kromě zobrazení dokáže i technologický proces ovládat.
- Reliance 4 Control Server**
 Má všechny schopnosti předchozích uvedených modulů. Je určen pro výkonné počítače, jelikož dokáže zároveň obsluhovat dispečerské požadavky, klientské runtime moduly a tenké klienty.



Obrázek 5 - Reliance 4 Control, Rozvod vzduchu Digital Park



Obrázek 6 - Reliance 4 Control Server, Vyhřívání budovy GEOVAPu

Tencí klienti

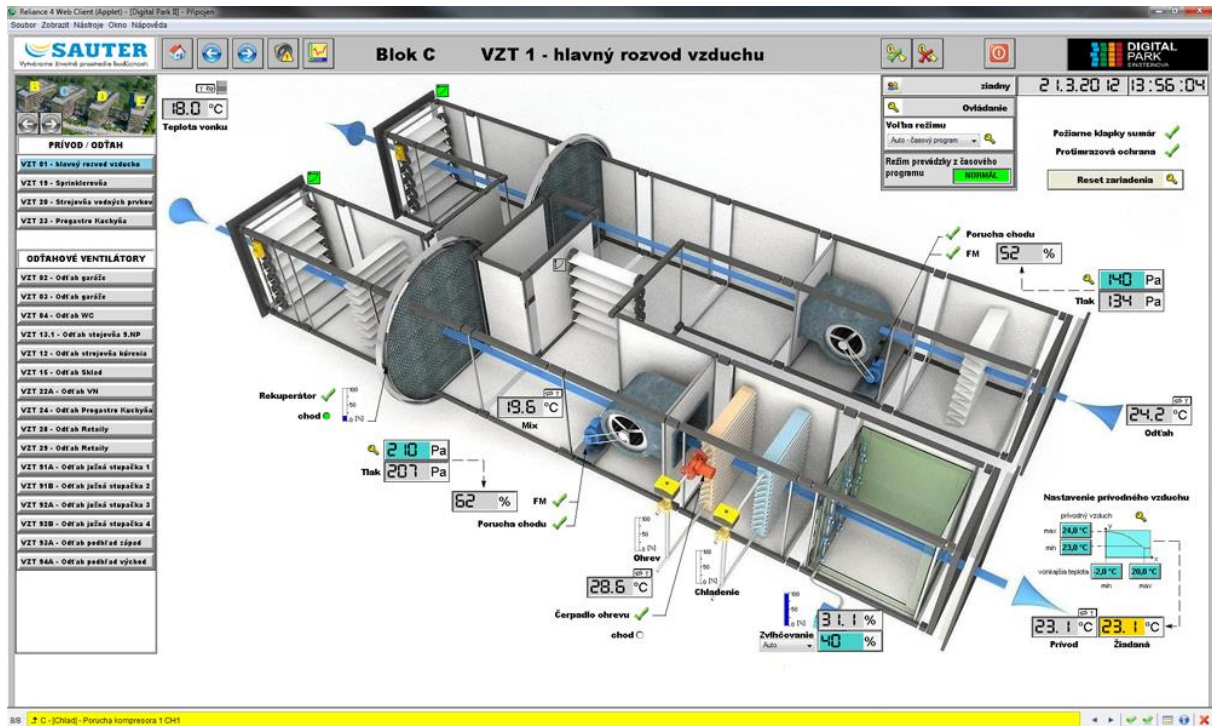
- Reliance 4 Smart Client

Tento klient je nová verze tenkého klienta, který je postavený na HTML5 a je optimalizován primárně pro dotykové použití, například na chytrých telefonech nebo tabletech. Má pouze část funkcí oproti plnohodnotnému klientskému programu jako je třeba Control. I tak ale dokáže zobrazení vizualizačních obrazovek s aktuálními daty, ovládání vizualizovaného technologického procesu, zobrazení a potvrzení aktuálních alarmů a zobrazení historických dat ve formě grafů.



Obrázek 7 - Reliance 4 Smart Client, Rodinný dům Průhonice

- Reliance 4 Web Client
Stejný klient jako Smart Client, pouze starší, postavený na platformě Java. Stále je však používanou částí Reliance SCADA, například na počítačích.



Obrázek 8 - Reliance 4 Web Client, Rozvod vzduchu Digital Park

Komunikační drivery

Reliance SCADA obsahuje velké množství driverů, které je možné zakoupit. Pro naše účely je podstatné, aby byla SCADA schopná spojení s PLC Siemens nebo PLC Tecomat. Reliance má pro tyto účely 4 možnosti spojení:

- Komunikační driver Teco – umožňuje využít vestavěnou základní komunikaci navrženou výrobcem, který ji používá například i ve svém programovacím prostředí Mosaic (PLC Tecomat)
- Komunikační driver Siemens – opět umožňuje využít vestavěnou základní komunikaci navrženou výrobcem (PLC Siemens)
- Komunikační driver MODBUS – umožňuje vytvořit datové propojení Runtime modulů Reliance se stanicemi (PLC) komunikujícími protokoly Modbus RTU nebo Modbus TCP (PLC Siemens i PLC Tecomat)
- Komunikační driver OPC UA Client – standardizované rozhraní na nových PLC. Bohužel v tomto případě se firma TECO vyjádřila, že zatím neplánují instalaci modulu OPC UA do svého firmware pro PLC Foxtrot. Proto tedy pouze pro PLC Siemens

4.2. mySCADA [5]

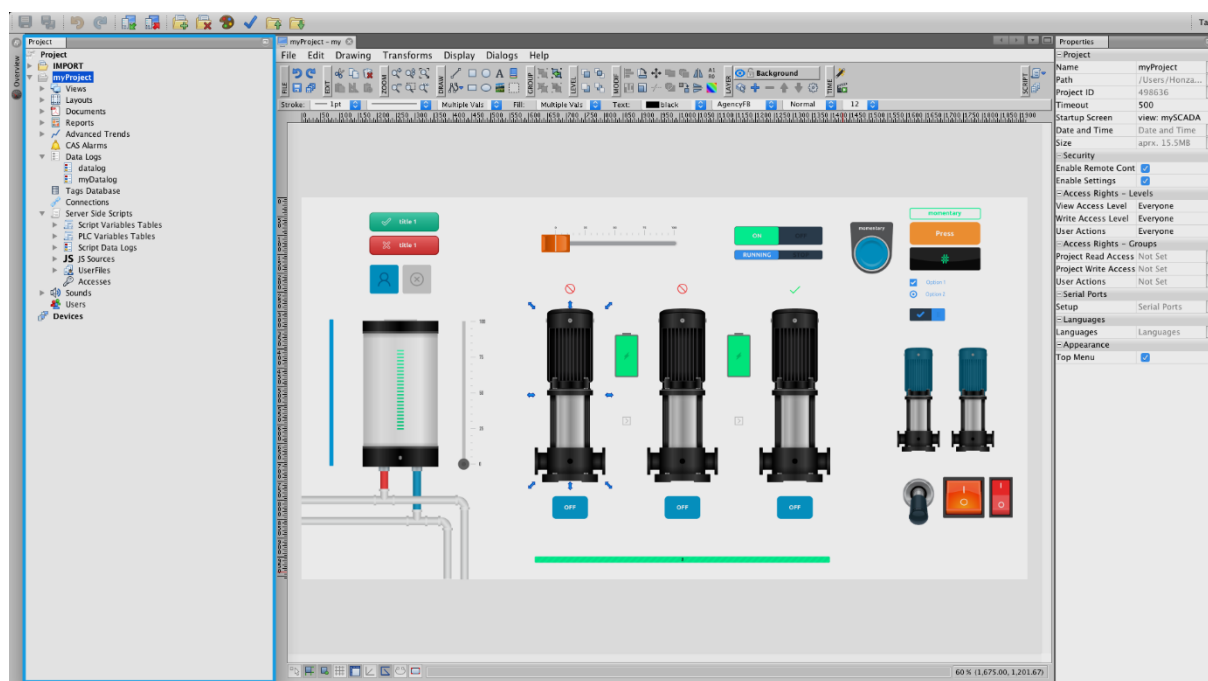
Druhým z kandidátů na SCADA systém je mySCADA od firmy mySCADA Technologies s.r.o. Tento software je spíše nováčkem na trhu, jelikož byla firma založena teprve na konci roku 2009. [6] I tak si ale za krátkou dobu své existence dokázala vybojovat určité místo na trhu (například u firem ČEZ nebo ZOOT) a navázat spolupráci s ČVUT.

Software mySCADA se skládá jen ze dvou částí: myDESIGNER a myPRO.

Má ale na rozdíl od Reliance na prodej i vlastní podporovaný hardware: myPANEL a myBOX.

myDESIGNER

myDESIGNER je prostředí pro tvorbu projektu. Momentálně jsou aktivní 2 verze tohoto programu, a to konkrétně verze 7 a verze 8. Ty se od sebe liší hlavně licenčními podmínkami. Verze 7 byla kompletně zdarma a je možné ji libovolně používat jak pro domácí, tak i firemní projekty. Verze 8 už existuje ve verzi classic a Enterprise. Verze Enterprise je placená a umožňuje přístup k pokročilým funkcím, které byli ve verzi 7 zdarma. Jde například o skriptování, různé módy diagramů, časové osy nebo pokročilá správa uživatelů.



Obrázek 9 - myDESIGNER 8

myPRO

myPRO je hlavní serverová aplikace. Její účel je zprostředkovávat spojení mezi PLC a serverem. Zároveň odpovídá na požadavky klientů. Do myPRO aplikace se nahrává projekt vytvořený pomocí aplikace myDESIGNER.

Klient

Pro prohlížení/ovládání mySCADA zde není žádná speciální aplikace, stačí webový prohlížeč. Obecně pro vývoj doporučuje pan Ondřej Salák z mySCADA prohlížeč Google Chrome, jelikož má schopnost spustit konzoli a vidět tak diagnostické informace. Funkčnost je však zajištěna ve všech moderních prohlížečích.

myBOX a myPANEL

Hardware vyvinutý firmou mySCADA pro server myPRO. MyBOX je určen pro montáž na DIN lištu přímo do ovládacího rozvaděče daného stroje nebo výrobního celku. MyPANEL je podobný jako myBOX, ale má na sobě i obrazovku, díky čemuž je vizualizace zobrazena přímo na zařízení.



Obrázek 10 - myBOX od mySCADA



Obrázek 11 - myPANEL od mySCADA

Komunikace

mySCADA neobsahuje tak velké množství komunikačních driverů, jako Reliance. Cílí spíše na velké hráče na trhu s PLC. Pro můj účel jsou nejvhodnější:

- Opět komunikační driver pro Siemens používající vestavěnou komunikaci (PLC Siemens)
- Dále OPC UA (PLC Siemens)
- A MODBUS (PLC Siemens a PLC Tecomat)

4.3. Zvolené řešení

Ze SCADA systémů nakonec volím software mySCADA. Důvodem je dostupnost plné licence v rámci spolupráce ČVUT a mySCADA Technologies. Po vizuální stránce je mySCADA též lepší, svěžejší.

5. Komunikace

Nyní, když máme vybrané PLC (PLC Tecomat) a vybraný i systém SCADA (mySCADA), je nutné zvolit komunikační protokol. mySCADA má pro komunikaci s PLC Tecomat jediné řešení, a tím je Modbus TCP.

5.1. MODBUS [7]

MODBUS je komunikační protokol v úrovni aplikační vrstvy ISO/OSI. Byl vytvořen v roce 1979 firmou MODICON. V současné době podporuje celou řadu komunikačních médií. V mém případě použijeme síť Ethernet s protokolem TCP/IP.

Tabulka 1 - Schéma MODBUS TCP v ISO/OSI

ISO/OSI	MODBUS protokol
Aplikační vrstva	Aplikační vrstva MODBUS
Prezentační vrstva	
Relační vrstva	MODBUS na TCP
Transportní vrstva	TCP
Síťová vrstva	IP
Linková vrstva	Ethernet II/802.3
Fyzická vrstva	Ethernet

Datový modul komunikace MODBUS je založen na sadě tabulek:

Tabulka 2 - Tabulky MODBUS

Tabulka	Název	Typ položky	Přístup	Adresa (MODICON)
Diskrétní vstupy	inputs	1-bit	Pouze čtení	10000÷19999
Cívky	coils	1-bit	Čtení/zápis	0÷9999
Vstupní registry	inputRegs	16-bit	Pouze čtení	30000÷39999
Uchovávací registry	holdingRegs	16-bit	Čtení/zápis	40000÷49999

Pohled na tabulky, co se týče přístupu, je ze serveru SCADA. PLC může zapisovat do všech tabulek.

Standartní port určený pro posílání dat na TCP je 502.

5.2. Programování PLC [8]

Pro komunikaci a programování PLC jsem použil program Mosaic. Mnou použitá verze je 2019.2.SP1. Tento program používá pro komunikaci s PLC UDP. Konkrétně na portu 61682. Výhodou UDP oproti TCP je rychlost, jelikož nedochází ke kontrole dat. Dále komunikuje na jiném portu, díky čemuž jsem schopný sledovat pomocí programu Mosaic stav MODBUS komunikace uvnitř PLC.

5.3. Nahrávání projektu mySCADA

Pro testování mé diplomové práce jsem měl k dispozici 2 servery myPRO. Jeden je školní, hostovaný na webových stránkách aspicc. Licenci na něj poskytla firma mySCADA Technologies v rámci spolupráce. Druhý server jsem měl nainstalovaný na svém PC, avšak bez licence. Omezení mezi licencovanou verzí a verzí bez licence je, že free běží pouze 2 hodiny. Poté je nutné počítač restartovat. Pro testovací účely to však stačí. Projekt se nahrává pomocí programu myDESIGNER (viz. kapitola 4.2) jako ZIP. Server jej následně rozbalí a spustí.

6. Program pro PLC

6.1. Testovací PLC

Testování probíhalo na PLC Foxtrot CP-1015 (více v kapitole 3.3). Toto PLC bylo pro testování zvoleno z více důvodů, z nichž hlavním byl ten, že obsahuje plnou licenci, která aktivuje kompilace složitých programů v programu Mosaic.

6.2. Programovací jazyk

Pro programování jsem zvolil dva programy podle normy IEC 61131-3. LD (Ladder diagram) pro hlavní program a ST (Structured text) pro definování proměnných a vlastní funkční bloky.

6.3. Komunikace MODBUS

První, co bylo nutné zprovoznit byla komunikace MODBUS. Pro komunikaci jsem použil knihovnu „ModbusRTU“ verze v3.6. Z této knihovny jsem použil funkční blok „fbModbusTCPSlave“. [9]

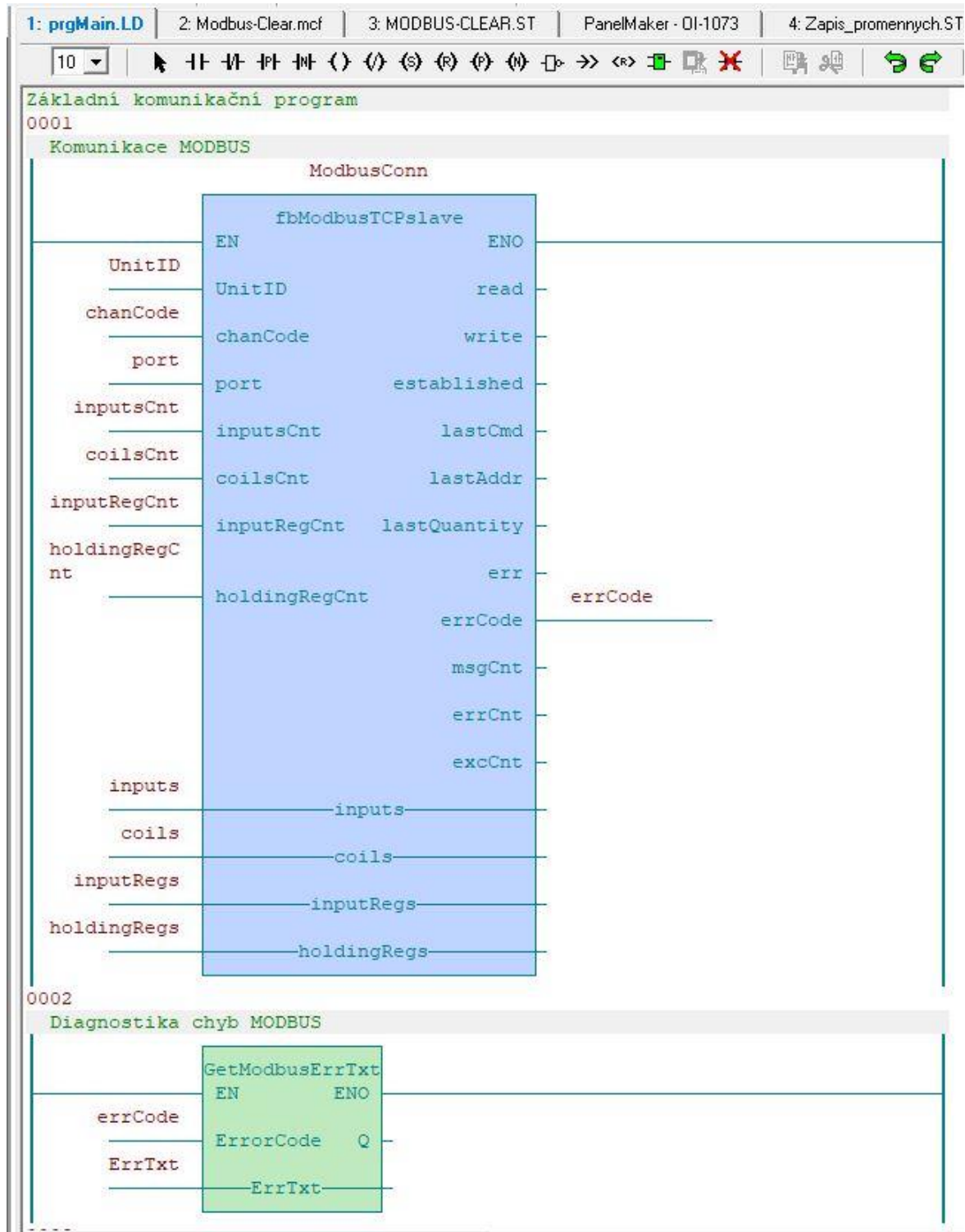
Funkční blok jsem pojmenoval ModbusConn a vložil ho do prvního obvodu v hlavním programu v Mosaic. Pro vstupy jsem vytvořil proměnné, které jsou pojmenované stejně jako názvy vstupů. Zároveň jsem si zadefinoval i výstupní proměnnou errCode, kvůli diagnostice.

Pro proměnnou errCode jsem přidal ještě jeden blok z knihovny ModbusRTU, a to GetModbusErrTxt, který překládá číselný kód chyby do textu.

Tabulka 3 - Hodnoty proměnných ModbusConn [9]

Proměnná	Hodnota	Typ	Poznámka
UnitID	1	USINT	Číslo PLC
chanCode	ETH1_uni0	UINT	Viz kapitola 6.4
port	502	UINT	Číslo portu, viz. kapitola 5.1
inputsCnt	64	UINT	Velikost pole inputs
coilsCnt	64	UINT	Velikost pole coils
inputRegCnt	64	UINT	Velikost pole inputReg
holdingRegCnt	64	UINT	Velikost pole holdingReg
inputs	ARRAY[0..inputsCnt]	BOOL	Viz Tabulka 2
coils	ARRAY[0..coilsCnt]	BOOL	
inputRegs	ARRAY[0..inputRegCnt]	UINT	
holdingRegs	ARRAY[0..holdingRegCnt]	UINT	
errCode		USINT	Chybový kód
ErrTxt		STRIN	Chybová hláška z kódu

Tento blok provádí příjem a vysílání 4 tabulek definovaných v tabulce 2. V tabulce (na bloku) je také proměnná chanCode. Tato proměnná určuje, jaký komunikační kanál má blok používat pro MODBUS spojení. Hodnota se mění, pokud by mělo PLC například 2 ethernetové porty nebo se používal k přenosu sériový port. O nastavení této proměnné je pojednáno v kapitole 6.4.



Obrázek 12 - Výstřižek obrazovky nastavení MODBUS bloku

6.4. Nastavení chanCode

The screenshot shows the ModbusClear software interface. The main window displays a table of communication channels. The selected channel is 'ETH'. A dialog box titled 'Nastavení univerzálního režimu kanálu' is open, showing settings for 'UNIO' mode. The dialog box has several sections:

- 5** (tab):
 - Prjmací zóna: 259
 - Delka zóny: 32768
 - Adresa zóny: ETH_UNIO_IN
 - Výlaci zóna: ETH_UNIO_OUT
- Typ protokolu:
 - TCP master
 - UDP
- Validační IP adresa: 0.0.0.0
- Validační port: 61000
- Míra šířky pásma: 61000

The main window table shows the following data:

Struktura kanálu	nám / pozice	Režim kanálu	Adresa pro komunikaci	Komunikační rychlost	Prostředí odpovědi	Délka zóny	Adresa zóny	Průměr s páteří
CP-1015	0 / 0	PC	0	38 400	10	off		on
CH1		OFF						
CH2		OFF						
CH3		OFF						
CH4		OFF						
EE1/4		PC, MOD	147.032.164.018					
ETH		PLC						
ETH		uni						
ETH		BAC						

Obrázek 13 - Nastavení komunikačního kanálu

Na Obrázek 13 vidíme postup nastavení komunikačního kanálu pro MODBUS:

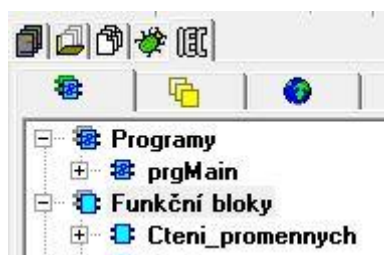
1. Nejprve klikneme na Manažera projektu
2. Poté konfigurace HW
Pokud nám nesedí typ PLC s reálným, klikneme na tlačítko „Načíst z PLC“
3. Poté klikneme na ikonku složky u CPU
4. Zobrazí se nám okno se všemi komunikačními kanály. Zvolíme kanál ETH1, v něm máme 4 podsložky. Vybereme složku „uni“ a klikneme opět na ikonku složky. [10]
5. Otevře se nám okno, ve kterém je nutné vyplnit:
 - a. Délku přijímací zóny na 259 [9]
 - b. Délku vysílací zóny na 259 [9]
 - c. Nastavit typ protokolu na TCP Slave [9]

V tomto okně je sice uveden i port, ale ten bude přepsán nastavením bloku fbModbusTCPSlave z předchozí kapitoly.

Parametr, který se vyplňuje do chanCode, v mém případě tedy ETH1__uni0, nalezneme v řádku přijímací nebo vysílací zóna. Například v řádku přijímací zóna je uvedeno ETH1__uni0__IN. Označení __IN udává, že jde o přijímací zónu. Jelikož ale chceme mít obousměrnou komunikaci, odebereme „__IN“ z názvu a získáme název celého kanálu.

6.5. Čtení a zápis proměnných

Dále pokračujeme tvorbou programu. Nyní jsem si vytvořil dva své vlastní funkční bloky následujícím postupem. V projektu v levém panelu klikneme na IEC ikonu a pod ní na ikonku „Programové organizační jednotky“. Zde klikneme na „Funkční bloky“ pravým tlačítkem myši a vybereme možnost „Přidat programovou organizační jednotku“.



Obrázek 14 - Přidání funkčního bloku

Po zobrazení menu zvolíme název a jazyk, ve kterém chceme daný funkční blok psát. Já si zvolil názvy „Cteni__promennych“ a „Zapis__promennych“ v jazyce ST. Tyto dva funkční bloky budou překládat hodnoty uložené na určitých adresách v registrech na proměnné používané v ostatních blocích.

Ukázka kódu:

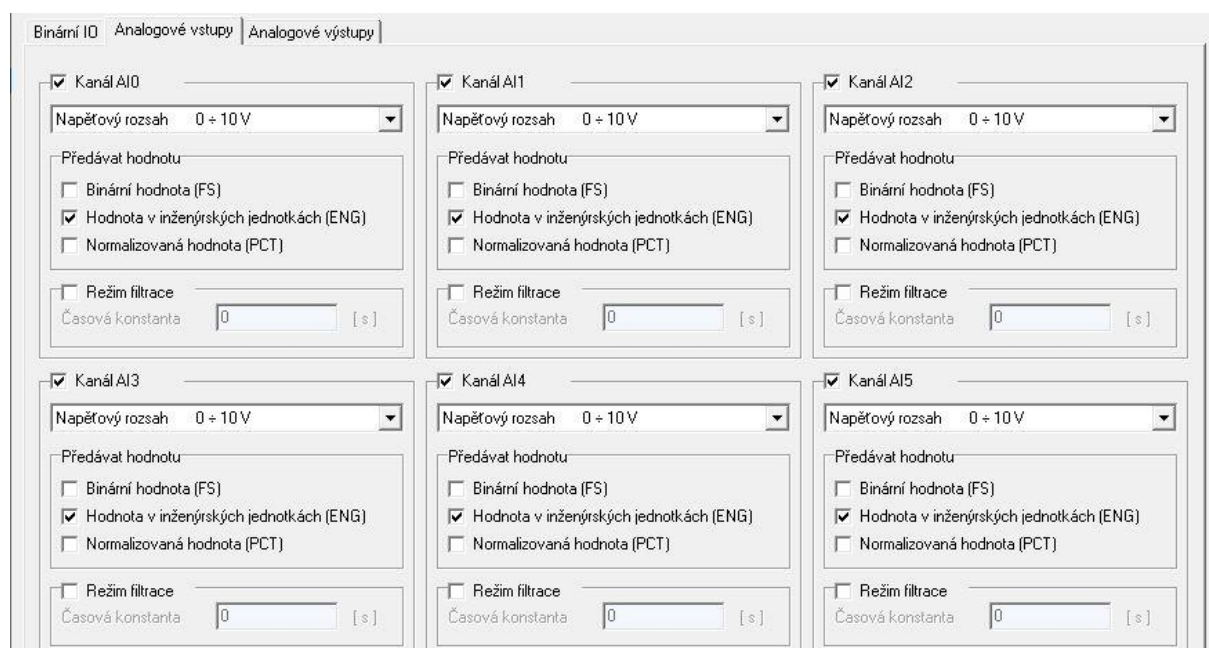
```
FUNCTION_BLOCK Cteni_promennych
AO0 := UINT_TO_REAL(holdingRegs[0])/100;
AO1 := UINT_TO_REAL(holdingRegs[1])/100;
END_FUNCTION_BLOCK
```

V kódu na řádku 2 vidíme příkaz, který mění hodnotu přijatou v tabulce holdingRegs na prvním místě [0] z UINT na REAL, dále ji vydělí 100 a zapíše do AO0, což je analogový výstup.

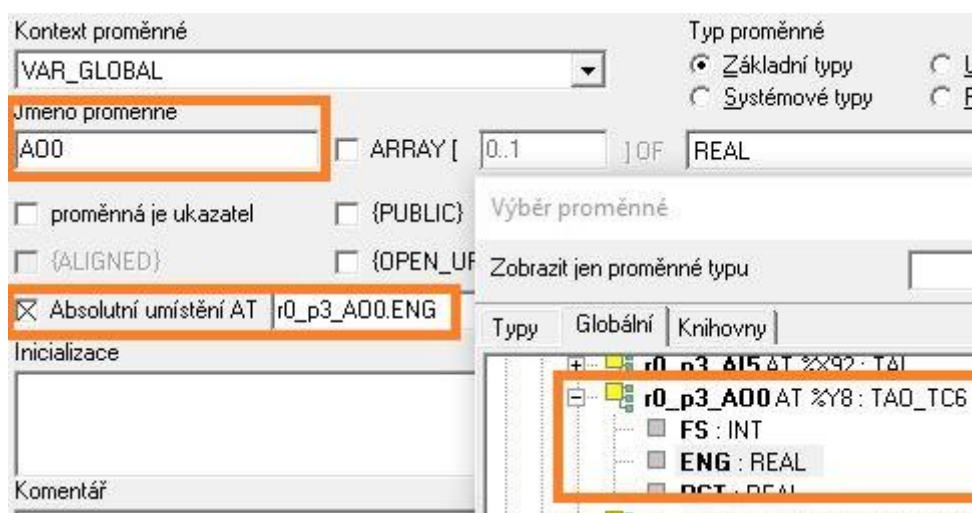
6.6. Nastavení analogových vstupů a výstupů

Aktivování analogových vstupů a výstupů se provádí podobně jako nastavení chanCode z kapitoly 6.4. Na obrázku 13 vidíme u 3 bodu 4 moduly dostupné v PLC. CPU, panel, CIB a I/O. Klikneme na složku u ikony I/O. Otevře se nám okno s dostupnými kanály a zvolíme si, které chceme aktivovat. (Obrázek 15) Ještě je nutné zadefinovat jejich přímou adresu jako naše AO0.

Otevřeme si menu systémové proměnné (v záložce IEC ikona planety) a přidáme globální proměnnou pravým tlačítkem myši. V zobrazeném dialogu zvolíme název (tedy např. AO0) a zaškrtneme „Absolutní umístění AT“. Poté klikneme na „...“ vpravo od tohoto pole a najdeme příslušnou adresu. Analogové vstupy a výstupy se nachází v globálních systémových proměnných. V mém případě je to adresa „r0_p3_AO0“. Je však důležité tuto proměnnou rozbalit a zvolit ENG hodnotu. Viz. Obrázek 16.



Obrázek 15 - Analogové vstupní kanály



Obrázek 16 - Definice proměnné A00

6.7. PID regulace

Jedním z bodů zadání je regulace v uzavřeném regulačním obvodu. Tu budu realizovat pomocí PID bloku „SimplePID“ z knihovny „ModelLib“ [11].

Postup je jednoduchý a velmi podobný jako u bloku Modbus. Do hlavního programu v Mosaic vložíme mezi naše bloky „Cteni_promennych“ a „Zapis_promennych“ nový obvod a do něj umístíme blok „fbSimplePID“.

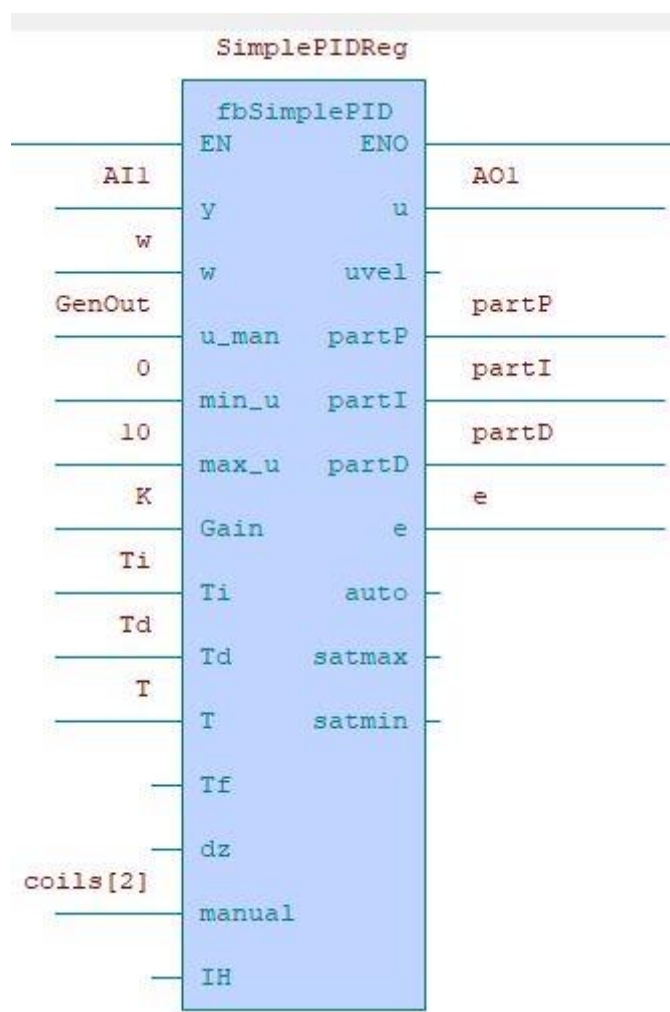
U tohoto bloku jsem si opět vytvořil proměnné, které jsem nazval tak, jak je vidět na obrázku 17.

Proměnná GenOut a coils[2] souvisí s kapitolou 6.8. Parametry „w, K, Ti, Td a T“ jsem zapsal do mého bloku čtení proměnných, jelikož jsou to proměnné, které chceme měnit mySCADOU.

Zápis:

```
w := UINT_TO_REAL(holdingRegs[8])/100;  
K := UINT_TO_REAL(holdingRegs[9])/100;  
Ti := UINT_TO_REAL(holdingRegs[10])/100;  
Td := UINT_TO_REAL(holdingRegs[11])/100;  
T := UINT_TO_REAL(holdingRegs[12])/100;
```

Důvod, proč jsou tyto proměnné dělené stem je ten, že protokol MODBUS nedokáže přenášet proměnné typu FLOAT (v PLC Tecomat se jmenují REAL). Proto v mySCADA odesíláme hodnotu násobenou stem a v PLC jí vydělíme, čímž získáme 2 desetinná místa.



Obrázek 17 - PID Regulátor

6.8. Generátor signálu

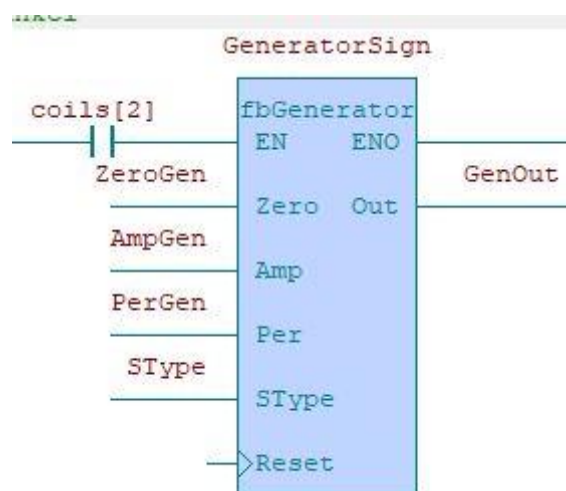
Pro automatické generování různých typů signálů jsem použil blok „Generator“ z knihovny „ModelLib“. [12]

Pro přidání bloku generátoru do programu je důležité, aby byl umístěn před PID regulátorem, jelikož využívá jeho manuálního vstupu pro zápis akční veličiny.

Parametr Zero posouvá spodní limit generovaného signálu, tj. když zadám 4, bude mít obdélníkový signál ve stavu „vypnuto“ hodnotu 4.

Parametr Amp udává rozdíl od hodnoty Zero, tj. když zadám 3, můj předchozí obdélníkový signál se bude pohybovat mezi 4 a 7.

Parametr Per udává periodu signálu v sekundách, tj. když zadám hodnotu 2, první sekundu bude hodnota 4, druhou sekundu hodnota 7.



Obrázek 18 - Generátor signálu

Velikým problémem byla u tohoto bloku proměnná `SType`. Podle její hodnoty se určuje, jaký signál bude vystupovat ven. Může nabývat hodnot: `gst_Sin`, `gst_Square`, `gst_Saw`, `gst_DblSaw`. Zprvu jsem se obával, že jde o `STRING` proměnnou, ale když jsem se podíval do nápovědy, zjistil jsem, že jde o speciální proměnnou typu „`TGeneratorSignalType`“. Nicméně se mi experimentálně podařilo zjistit, že tato proměnná nemusí být `STRING` s konkrétním textem, ale stačí označení 0 – 3, viz tabulka 4.

Tabulka 4 - Hodnoty pro nastavení signálu

Signál	String	Zadávaná hodnota REAL
Sinus	<code>gst_Sin</code>	0
Obdélník	<code>gst_Square</code>	1
Rampa	<code>gst_Saw</code>	2
Pila	<code>gst_DblSaw</code>	3

Bylo však nutné přetypovat `UINT`, který odesílá `MODBUS` na `TGeneratorSignalType`, který je na vstupu bloku. Toto jsem vyřešil zadefinováním proměnné `SType` stejně jako u `AO0` (kapitola 6.6).

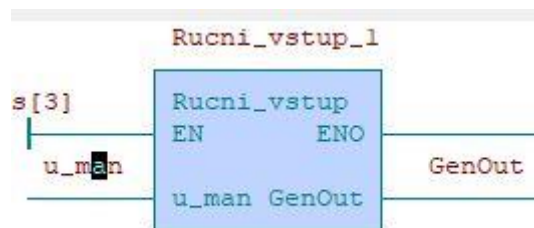
Na obrázku 19 jsou vidět vyplněná pole. Důležitý je typ proměnné a „absolutní umístění `AT`“, zde nastavené na `holdingRegs[3]`.

Kontext proměnné		Typ proměnné	
VAR GLOBAL		<input type="radio"/> Základní typy	<input checked="" type="radio"/> Uživatelské typy
Jméno proměnné		<input type="radio"/> Systémové typy	<input type="radio"/> Funkční bloky
SType	<input type="checkbox"/> ARRAY [0..1] OF	TGeneratorSignalType	
<input type="checkbox"/> proměnná je ukazatel	<input type="checkbox"/> {PUBLIC}	<input type="checkbox"/> {PUBLIC_IFOX}	
<input type="checkbox"/> {ALIGNED}	<input type="checkbox"/> {OPEN_UP}	<input type="checkbox"/> {TRACK_ADR} <input type="checkbox"/> {PLCNET} <input type="checkbox"/> {PUBLIC_API}	
<input checked="" type="checkbox"/> Absolutní umístění AT	holdingRegs[3]		

Obrázek 19 - Definice proměnné SType

6.9. Manuální vstup

Pro manuální vstup jsem si vyrobil jednoduchý funkční blok stejným způsobem jako v kapitole 6.5. Jako vstupní proměnná je zde `u_man` typu REAL. Výstupní proměnná je `GenOut`, která je stejná jako z generátoru.



Obrázek 20 - Blok ručního vstupu

6.10. Výhybka pro LabJack

Kolega Ing. Jan Riedl pracoval ve své diplomové práci na zařízení pro konkurenční ovládání laboratorního modelu. Zjednodušeně jde o signálovou výhybku ovládanou digitálním vstupem. Toto zařízení bude přiloženo ke každé laboratorní úloze, na které dochází i k měření v rámci předmětu identifikace dynamických systémů. Ve SCADA projektu pak musí být přepínač, který odepne zápisy veškerých hodnot na výstupy, vypne PID i generátor a PLC bude fungovat pouze v režimu čtení. Tento hlavní vypínač jsem umístil do tabulky coils na první pozici, tj. [0]. Je nutné po hardwarovém zapojení daný digitální vstup propojit s touto proměnnou, tedy nastavit DI AT coils[0].

6.11. Logika přepínání

Přepínání je řešené pomocí základních spínačů v jazyce LD (-| | -), případně jejich negací (-|/| -). Konkrétně coils[0] je negovaný a vypíná všechny bloky kromě MODBUS a „Zapis_promennych“. Coils[1] zapíná PID regulátor. Tomu je nadřazen Generátor průběhů, který je na coils[2]. To sepne manuální vstup do PID regulátoru. Nad Generátorem průběhů je manuální vstup, který je ovládán pomocí coils[3].

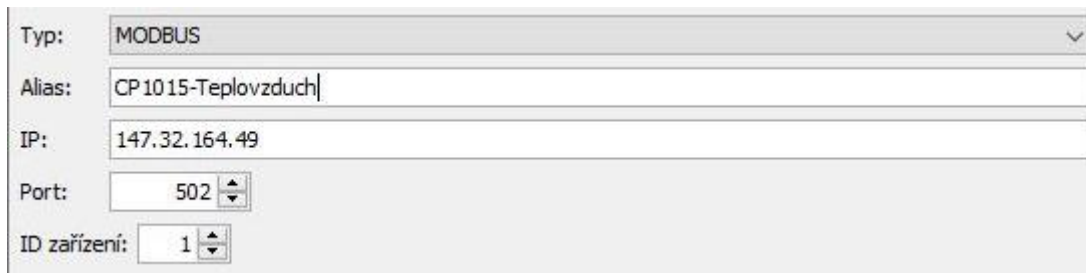
6.12. Přehled tabulek MODBUS

Pro přehlednost jsem vytvořil Excel tabulku se čtyřmi listy, aby bylo snadné se orientovat v použitých proměnných a nedošlo ke kolizi. Nachází se jako digitální příloha.

7. mySCADA Projekt

7.1. Vytvoření projektu

Vytvoříme si základní prázdný projekt. Jako první do něj přidáme spojení s PLC.



Typ:	MODBUS
Alias:	CP1015-Teplovzduch
IP:	147.32.164.49
Port:	502
ID zařízení:	1

Obrázek 21 - Nastavení spojení ve SCADA

Zvolíme typ spojení MODBUS, do *Alias*: vyplníme jméno PLC. V mém případě je to teplovzdušný model. Dále zadáme IP adresu, port 502 (viz. kapitola 5.1) a ID zařízení takové, jaké jsme zvolili v kapitole 6.3 (Tabulka 3, proměnná UnitID).

7.2. Test spojení

Následně jsem otestoval spojení vytvořením pohledu a umístěním přepínače ovládajícího coils[0]. První pokus nefungoval. Po krátké diagnostice pomocí aplikace telnet jsem zjistil, že mám blokované spojení na portu 502. Po udělení výjimky v mém firewallu začal přepínač fungovat. Mohl jsem tedy začít vytvářet pohledy.

7.3. Pohled LabJack

LabJack

Vypnutí ovládání pomocí SCADA a přepnutí na vstup z LabJack.

LabJack zapnutý

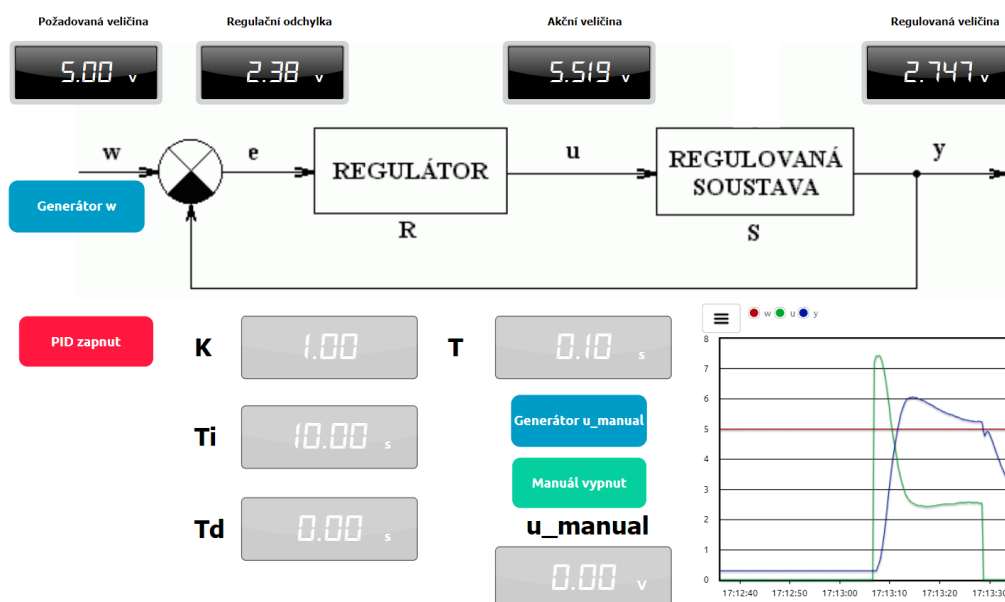


LabJack vypnutý

Obrázek 22 - Pohled LabJack

Po menší úpravě jsem z testovacího pohledu vytvořil pohled pro přepínání mezi ovládaním z PLC a z LabJacku.

7.4. Pohled URO



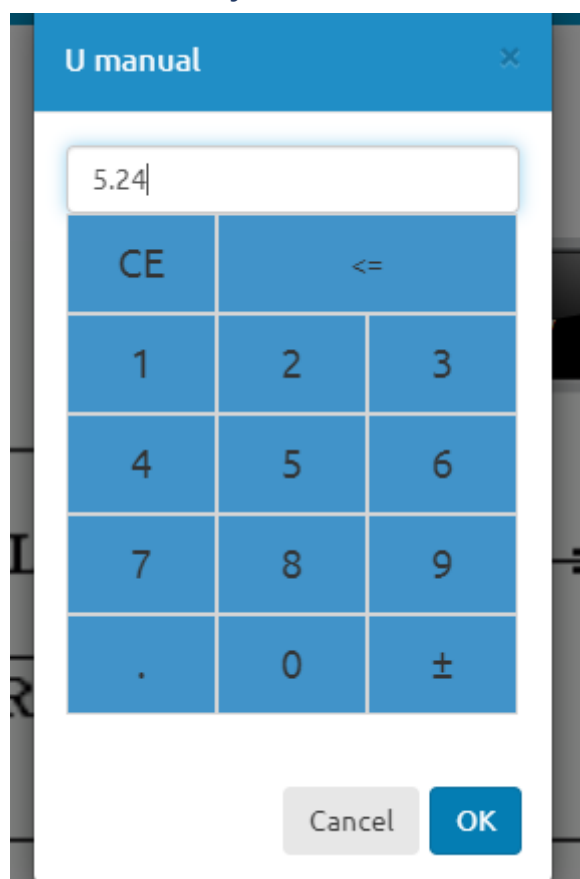
Obrázek 23 - Pohled URO

Na obrázku 23 je vidět pohled v mySCADA s běžícím PID regulátorem. V tomto pohledu jsou aktivní pole na zadávání hodnot.

Seznam polí:

1. w – požadovaná veličina [V]
Po kliknutí je možné nastavit hodnotu požadované veličiny
2. K – zesílení regulátoru
Po kliknutí na hodnotu je možné nastavovat velikost zesílení
3. T_i – integrační časová konstanta [s]
Zadává se hodnota integrační konstanty
4. T_d – derivační časová konstanta [s]
Zadává se hodnota derivační časové konstanty
5. T – perioda vzorkování [s]
Možnost změnit periodu vzorkování, výchozí hodnota 0,1s
6. u_manual – manuální hodnota akční veličiny
Možnost použít ruční vstup akční veličiny

Zadávací panel číselné hodnoty



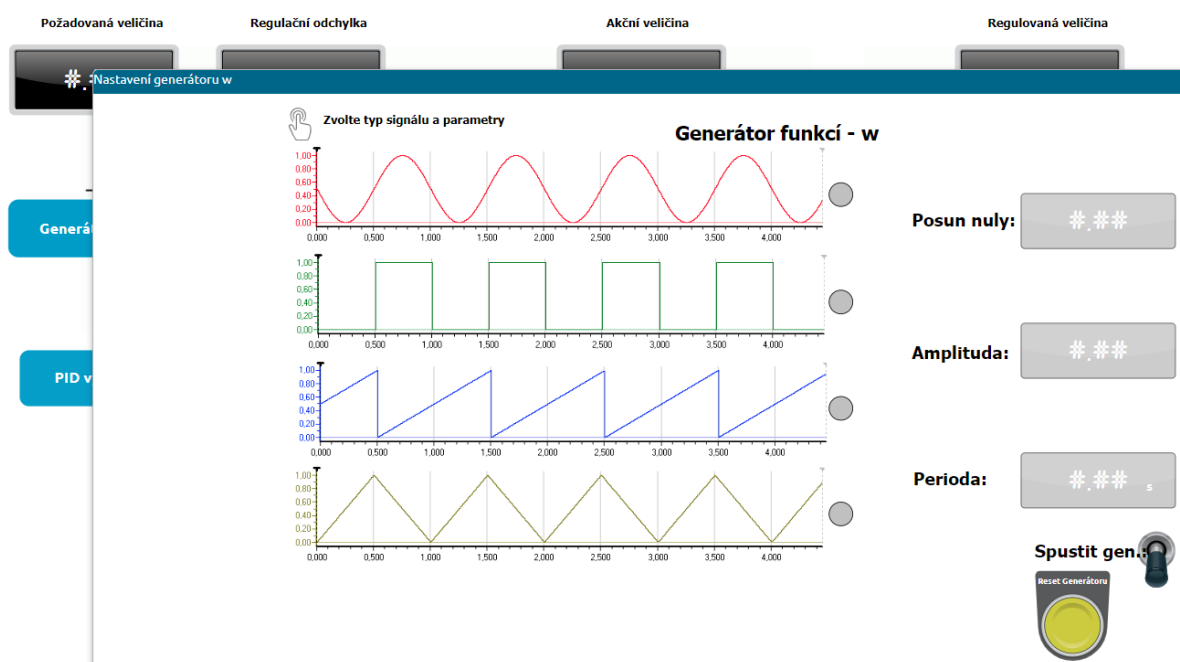
Obrázek 24 - Zadávací pole hodnoty

Toto dialogové okno je optimalizované i pro dotykové ovládání, což má výhodu, pokud by na pracovištích byly dotykové stanice. Ušetřilo by se tím místo nutné pro klávesnici a myš.

Na tomto pohledu jsou také 2 aktivní tlačítka: PID a Manuál. Tato tlačítka ovládají a zároveň indikují stav. Pokud je tlačítko zelené, je možné zasahovat do laboratorní úlohy. Pokud je však červené, laboratorní úloha je aktivní a dochází k měření.

Je zde i graf zobrazující průběh w , u a y .

Jsou zde i tlačítka s nápisy generátor. Ta otevřou aktivní okna s nastavením příslušného generátoru (viz. obrázek 25).

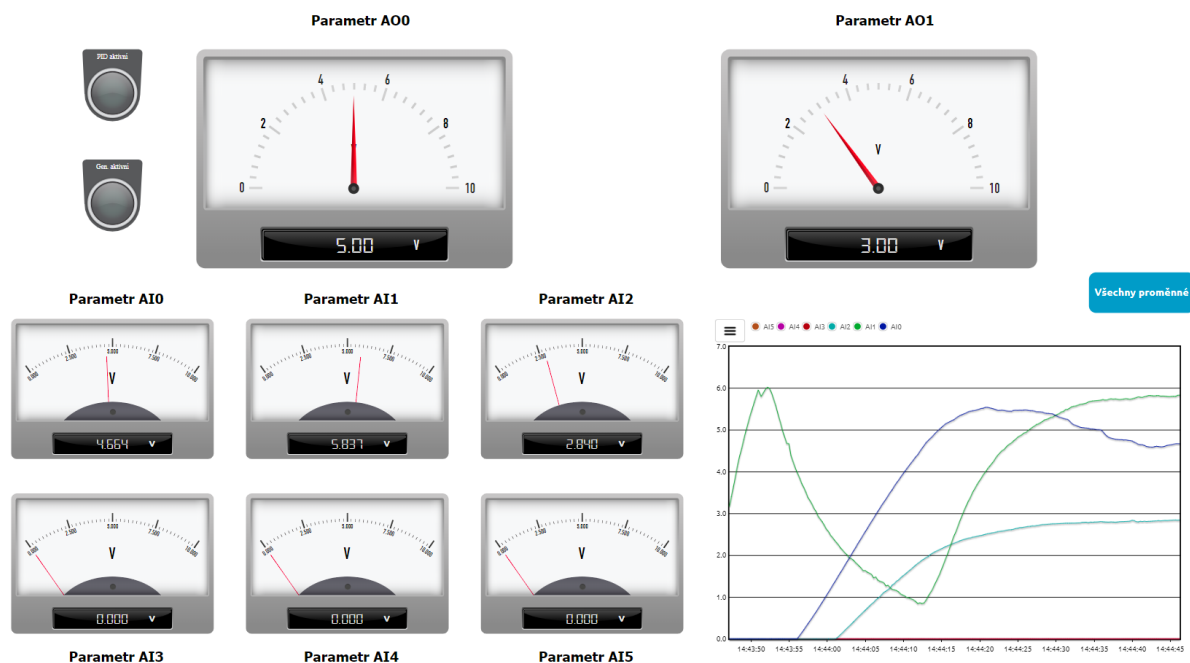


Obrázek 25 - Pohled na aktivní okno generátoru

Na obrázku 25 je vidět pohled na Generátor funkcí. Opět jsou zde zadávací pole: Posun nuly, Amplituda a Perioda. Tyto proměnné jsou popsány v kapitole 6.8.

I zde jsou také ovládací prvky. Je zde páčkový přepínač, který zapíná a vypíná generátor. Je zde také tlačítko na reset generátoru, aby začal „od nuly“. Průběh je možné volit kliknutím na příslušný obrázek. Zelená kontrolka vpravo od grafu indikuje, který je právě aktivní. Pátý graf je živý pohled na výstupní veličinu u .

7.5. Pohled Diagnostický



Obrázek 26 - Úplný manuální režim

Tento pohled je určen jen pro zkušební účely. V tomto pohledu můžeme nastavovat přímo napětí na analogové výstupy. Kliknutím na velké voltmetry se zobrazí zadávací dialog stejný jako na obrázku 24.

Dále zde máme malé voltmetry, které sledují napětí na jednotlivých analogových vstupech. Ty se zakreslují do grafu vpravo od nich.

Nad grafem je tlačítko „Všechny proměnné“. Toto tlačítko zobrazí v grafu všechny analogové vstupy naráz. V případě kliknutí na konkrétní voltmetr se graf přepne a zobrazí jen tuto jednu určitou proměnnou.

7.6. Pohled volby režimu



Obrázek 27 - Pohled volby režimu práce s úlohou

Toto jsou odkazy, které otevírají jednotlivé pohledy z minulých kapitol. Bodově následující:

- PLC nebo PC – kapitola 7.3
- URO – kapitola 7.4
- Přímé řízení – kapitola 7.5

7.7. Úvodní obrazovka



Obrázek 28 - Úvodní obrazovka

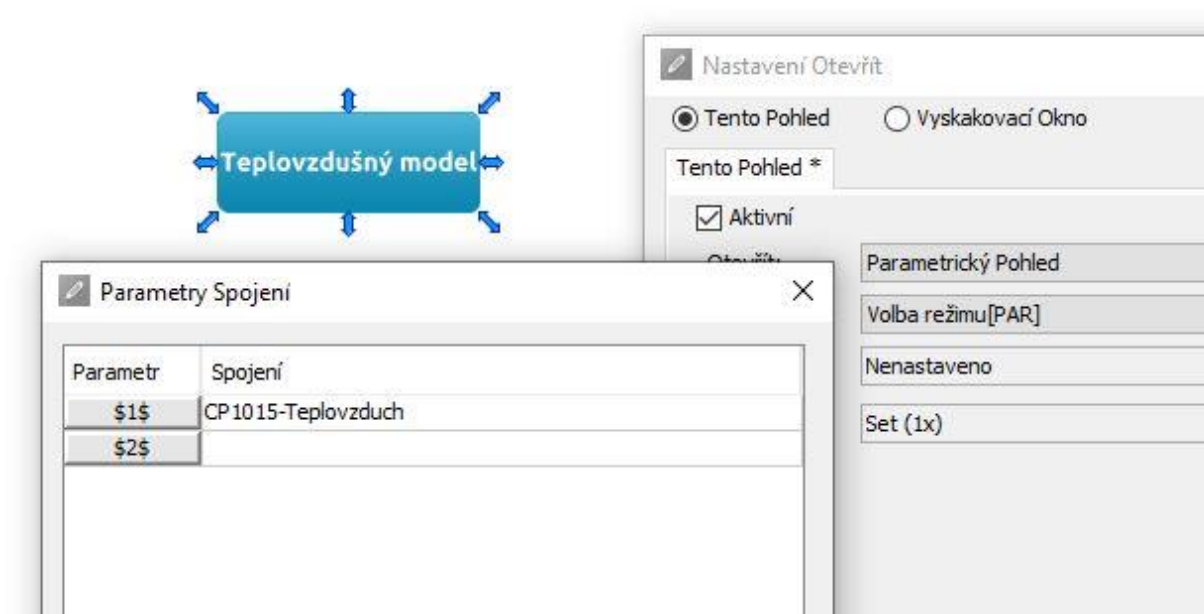
Tato obrazovka je určena ke zvolení laboratorní úlohy, se kterou bude student pracovat.

7.8. Možnosti spojení serveru s PLC

Vzhledem k tomu, že toto řešení není plně koncové, hleděl jsem hlavně na univerzálnost v případě rychlého nasazení do výuky. V tomto projektu jsou 2 možnosti, jak přidat další úlohu.

Metoda parametrického řešení

Všechny pohledy, které jsou zde uvedené, jsou parametrické. Veškeré spojení v pohledech je indexované na PLC \$1\$. Pro přidání nové úlohy stačí přidat nové spojení (popsáno v kapitole 7.1). A následně zkopírovat odkazové tlačítko (obrázek 28). V jeho parametrech pak stačí změnit, kam odkazuje spojení na \$1\$. Tato možnost se skrývá v záložce Příkazy, Otevřít.



Obrázek 29 - Změna parametrů spojení

Na obrázku 29 je parametr \$1\$ namířený na spojení PLC teplovzdušného modelu. Místo tohoto PLC nastavíme spojení s novým PLC.

Metoda více serverů

Druhá možnost je uskutečnitelná díky způsobu vedení serverů mySCADA na ČVUT. Jednotlivé servery jsou vedené v Dockeru. To znamená, že stačí zduplikovat celý projekt a změnit jen IP adresu ve spojení.

8. Závěr

Navrhl jsem prostředí SCADA v programu mySCADA ve spojení s PLC firmy Teco, konkrétně CP-1013.

O PLC je pojednáno v kapitole 3. O SCADA systémech v kapitole 4.

Jejich komunikace je řešená pomocí protokolu MODBUS, viz. kapitola 5.1.

Pro PLC jsem navrhl program v jazyce LD a pomocné funkční bloky v jazyce ST. Programy jsou přiloženy v příloze.

Dále je v příloze tabulka MODBUS vstupů a výstupů.

Podařilo se vytvořit funkční Uzavřený regulační obvod, popsán v kapitole 7.4. Tento pohled ve SCADA obsahuje i Otevřený regulační obvod.

Generátor signálu byl vytvořen pro vstup akční veličiny (u) nebo žádané veličiny (w).

Celý obvod jsem otestoval na teplovzdušném modelu a spolehlivě fungoval.

Je zřejmé, že drobné doladění projektu bude nutné při přechodu na PLC 1013 místo testovacího 1015, avšak budou to jen drobné úpravy.

Největší problém při řešení této diplomové práce mi dělalo spojení, jelikož mě často blokoval firewall nebo byl jiný problém se sítí. Dalším problémem bylo ze začátku udržení koncepce rozložení tabulek MODBUS.

Jsem rád, že jsem mohl pracovat na takovémto projektu. Velmi mě to bavilo a těším se na pokračování, kdy tento systém s vedoucím diplomové práce uvedeme do plné praxe na nových PLC a uvidím i zrod poslední části tohoto projektu, tedy přenos naměřených dat do mySCADA.

9. Bibliografie

- [1] CP-1013 - Základní moduly - PLC Tecomat Foxtrot - Katalog produktů - TECO - Automatizace. *Tecomat Produkty* [online]. Kolín: Teco a.s., 2017 [cit. 2019-06-06]. Dostupné z: <https://www.tecomat.cz/Products/cz/plc-tecomat-foxtrot/zakladni-moduly/1191-cp-1013/>
- [2] Siemens S7-1200. *Automatizační systémy Siemens* [online]. Česká republika: Siemens s.r.o., 2019 [cit. 2019-11-27]. Dostupné z: <https://new.siemens.com/cz/cs/products/automation/systems/industrial/plc/s7-1200.html>
- [3] CP-1015 - Základní moduly - PLC Tecomat Foxtrot - Katalog produktů - TECO - Automatizace. *Tecomat Produkty* [online]. Kolín: Teco a.s., 2017 [cit. 2019-06-05]. Dostupné z: <https://www.tecomat.cz/Products/cz/plc-tecomat-foxtrot/zakladni-moduly/131-cp-1015/>
- [4] *Reliance 4 SCADA* [online]. Pardubice: GEOVAP s.r.o., 2020 [cit. 2019-12-26]. Dostupné z: <https://www.reliance-scada.com/cs/products/reliance4-scada-hmi-system>
- [5] *MySCADA* [online]. Praha: mySCADA Technologies s.r.o., 2019 [cit. 2019-12-15]. Dostupné z: <https://www.myscada.org/en/>
- [6] MySCADA Technologies s.r.o. Penize.cz. *PENIZE.CZ* [online]. Praha: PENIZE.CZ, 2009 [cit. 2020-01-04]. Dostupné z: <https://rejstrik.penize.cz/28978650-myscada-technologies-s-r-o>
- [7] MODBUS APPLICATION PROTOCOL SPECIFICATION. In: *Modbus website* [online]. Hopkinton: Modbus Organization, Inc., 2006 [cit. 2020-12-17]. Dostupné z: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf
- [8] Mosaic - pro vývoj PLC programu. *Tecomat Software* [online]. Kolín: Teco a.s., 2019 [cit. 2020-10-18]. Dostupné z: <https://www.tecomat.cz/ke-stazeni/software/mosaic/>
- [9] 3.2 Funkční blok fbModbusTCPslave: Knihovna: ModbusRTUlib. *Tecomat Software: Mosaic* [software]. Kolín: Teco a.s., 2019 [cit. 2020-01-21]. Dostupné z: <https://www.tecomat.cz/ke-stazeni/software/mosaic/>



- [10 ŠRÁMEK, Filip. *Vzdálené řízení PLC SCADA systémem pomocí protokolu MODBUS*. Praha, 2018. Bakalářská práce. ČVUT v Praze.
- [11 1.2.8 fbSimplePID: knihovna: ModelLib. *Tecomat Software: Mosaic* [software]. Kolín: Teco a.s., 2019 [cit. 2020-01-15]. Dostupné z: <https://www.tecomat.cz/ke-stazeni/software/mosaic/>
- [12 1.2.10 fbGenerator: knihovna: ModelLib. *Tecomat Software: Mosaic* [software]. Kolín: Teco a.s., 2019 [cit. 2020-01-15]. Dostupné z: <https://www.tecomat.cz/ke-stazeni/software/mosaic/>



Přílohy

Příloha 1 – Hlavní program do PLC

Příloha 2 – Program bloku Cteni__promennych

Příloha 3 – Program bloku Zapis__promennych

Příloha 4 – Program bloku Generator__enable

Příloha 5 – Návod k bloku fbModbusTCPSlave z Mosaic programu

Digitální přílohy

 Excel tabulka pro MODBUS

Název souboru: MODBUS.xlsx

Obsah: Obsahuje popisy jednotlivých tabulek v MODBUS spojení a jejich propojení s proměnnými v programu PLC

 Archiv obsahující program PLC

Název souboru: __PG__Diplomka__2020-01-17__23-20-45.piz

Obsah: Soubor určený k importování do programu Mosaic

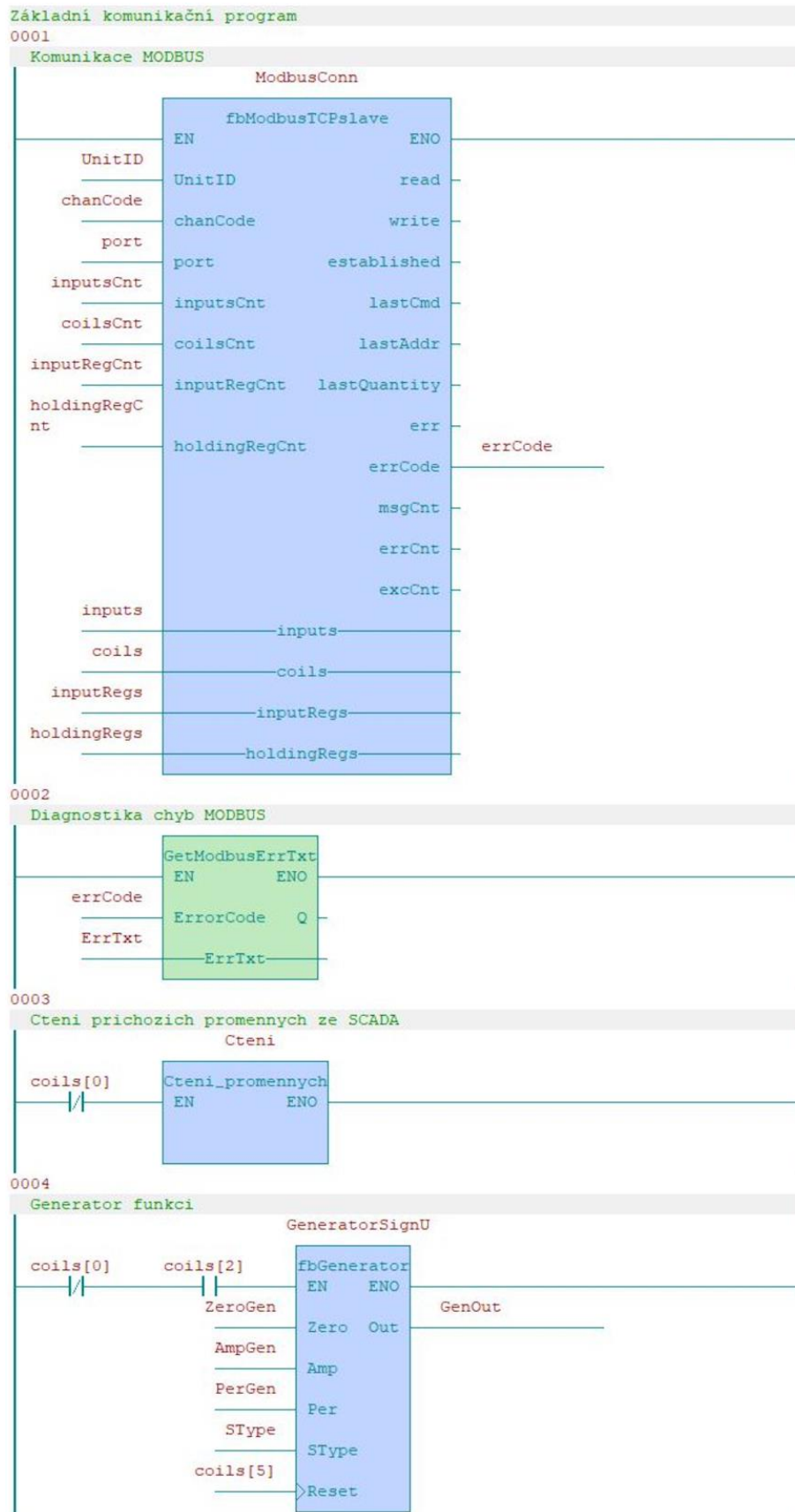
 Archiv obsahující projekt mySCADA

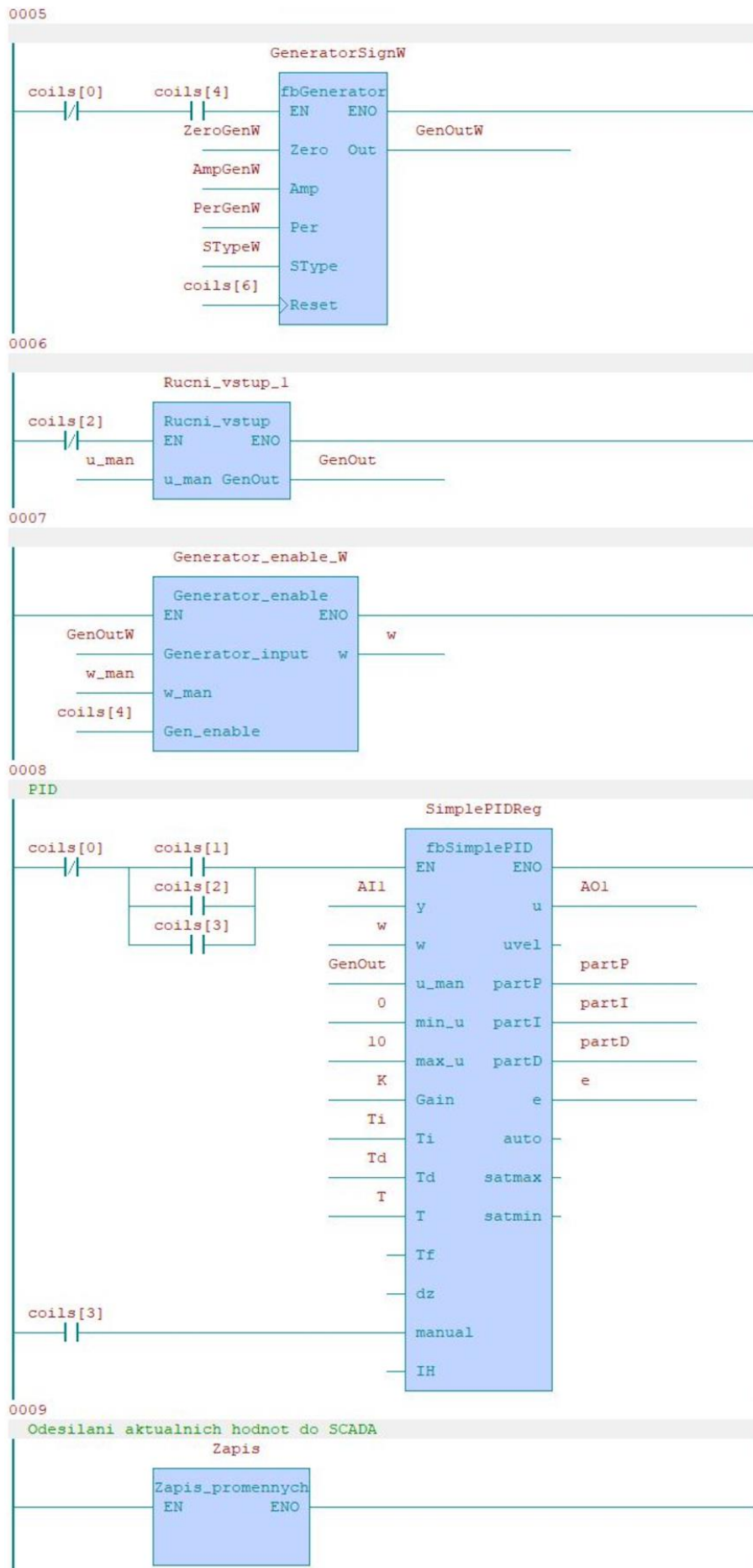
Název souboru: rizeni-laborky.mep

Obsah: Soubor určený k importování do programu myDESIGNER 8



Příloha 1 – Hlavní program do PLC







Příloha 2 – Program bloku Cteni_promennych

```
FUNCTION_BLOCK Cteni_promennych
  VAR_INPUT
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR_IN_OUT
  END_VAR
  VAR
  END_VAR
  VAR_TEMP

  END_VAR

  A00 := UINT_TO_REAL(holdingRegs[0])/100;
  A01 := UINT_TO_REAL(holdingRegs[1])/100;
  ZeroGen := UINT_TO_REAL(holdingRegs[4])/100;
  AmpGen := UINT_TO_REAL(holdingRegs[5])/100;
  PerGen := UINT_TO_REAL(holdingRegs[6])/100;
  w_man := UINT_TO_REAL(holdingRegs[8])/100;
  K := UINT_TO_REAL(holdingRegs[9])/100;
  Ti := UINT_TO_REAL(holdingRegs[10])/100;
  Td := UINT_TO_REAL(holdingRegs[11])/100;
  T := UINT_TO_REAL(holdingRegs[12])/100;
  u_man := UINT_TO_REAL(holdingRegs[14])/100;
  ZeroGenW := UINT_TO_REAL(holdingRegs[17])/100;
  AmpGenW := UINT_TO_REAL(holdingRegs[18])/100;
  PerGenW := UINT_TO_REAL(holdingRegs[19])/100;

  END_FUNCTION_BLOCK
```



Příloha 3 – Program bloku Zapis_promennych

```
FUNCTION_BLOCK Cteni_promennych
  VAR_INPUT
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR_IN_OUT
  END_VAR
  VAR
  END_VAR
  VAR_TEMP

  END_VAR

inputRegs[0] := REAL_TO_UINT(AI0*1000);

inputRegs[1] := REAL_TO_UINT(AI1*1000);
inputRegs[2] := REAL_TO_UINT(AI2*1000);
inputRegs[3] := REAL_TO_UINT(AI3*1000);
inputRegs[4] := REAL_TO_UINT(AI4*1000);
inputRegs[5] := REAL_TO_UINT(AI5*1000);
inputRegs[6] := REAL_TO_UINT(AO0*1000);
inputRegs[7] := REAL_TO_UINT(AO1*1000);
inputRegs[8] := REAL_TO_UINT(e*100);
inputRegs[9] := REAL_TO_UINT(partP*1000);
inputRegs[10] := REAL_TO_UINT(partI*1000);
inputRegs[11] := REAL_TO_UINT(partD*1000);
inputRegs[12] := REAL_TO_UINT(w*100);

END_FUNCTION_BLOCK
```



Příloha 4 – Program bloku Generator_enable

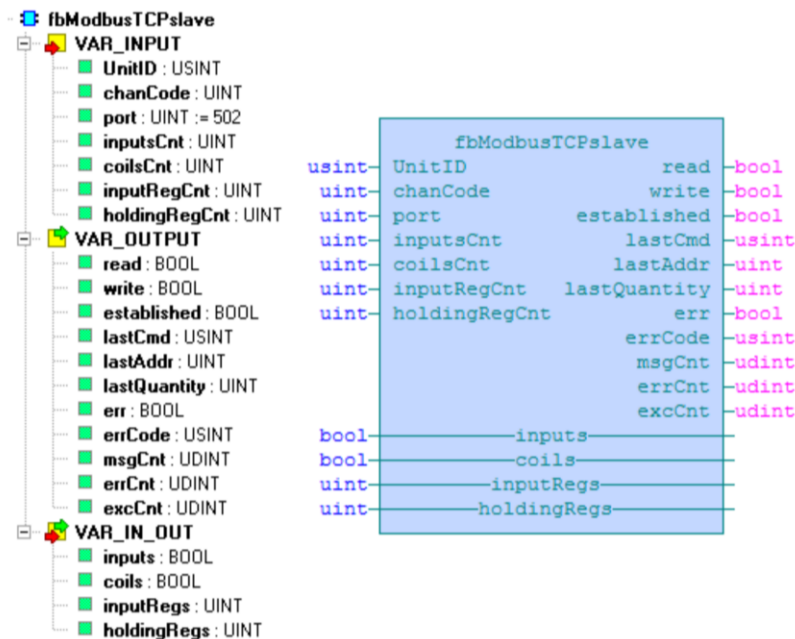
```
FUNCTION_BLOCK Generator_enable
  VAR_INPUT
    Generator_input : REAL;
    w_man : REAL;
    Gen_enable : BOOL;
  END_VAR
  VAR_OUTPUT
    w : REAL;
  END_VAR
  VAR_IN_OUT
  END_VAR
  VAR
  END_VAR
  VAR_TEMP
  END_VAR

  IF Gen_enable THEN w := Generator_input;
  ELSE w := w_man;
  END_IF;
END_FUNCTION_BLOCK
```

Příloha 5 – Návod k bloku fbModbusTCPslave z Mosaic programu

3.2 Funkční blok fbModbusTCPslave

Knihovna: *ModbusRTUlib*



Funkční blok *fbModbusTCPslave* interpretuje příkazy ModbusTCP na TCP spojení specifikovaných vstupem *chanCode* s identifikátorem zařízení daným vstupem *UnitID*. Podporované příkazy viz. Kap. 1.3 Tab. 1.

Datové bloky jsou definovány vstupy *inputs* (diskrétní vstupy), *coils* (cívky), *inputRegs* (vstupní registry) a *holdingRegs* (vnitřní registry). Počty objektů v zónách jsou dány vstupy *inputsCnt*, *coilsCnt*, *inputRegCnt* a *holdingRegCnt*.

Je důležité, aby vstupy *inputs* a *coils* odkazovali na proměnné typu *BOOL* nebo *ARRAY OF BOOL*. Vstupy *inputRegs* a *holdingRegs* mohou odkazovat na jakýkoli typ kromě *BOOL*.

Počty objektů v zónách musí být rovny nebo menší než je velikost proměnných, na které odkazují.

Pokud tyto podmínky nejsou dodrženy může dojít k zápisu do jiné části paměti.

Zóny se mohou vzájemně překrývat. Překrytí zón umožňuje ke stejné paměti přistupovat jak po 16 bitových slovech tak i po bitech.

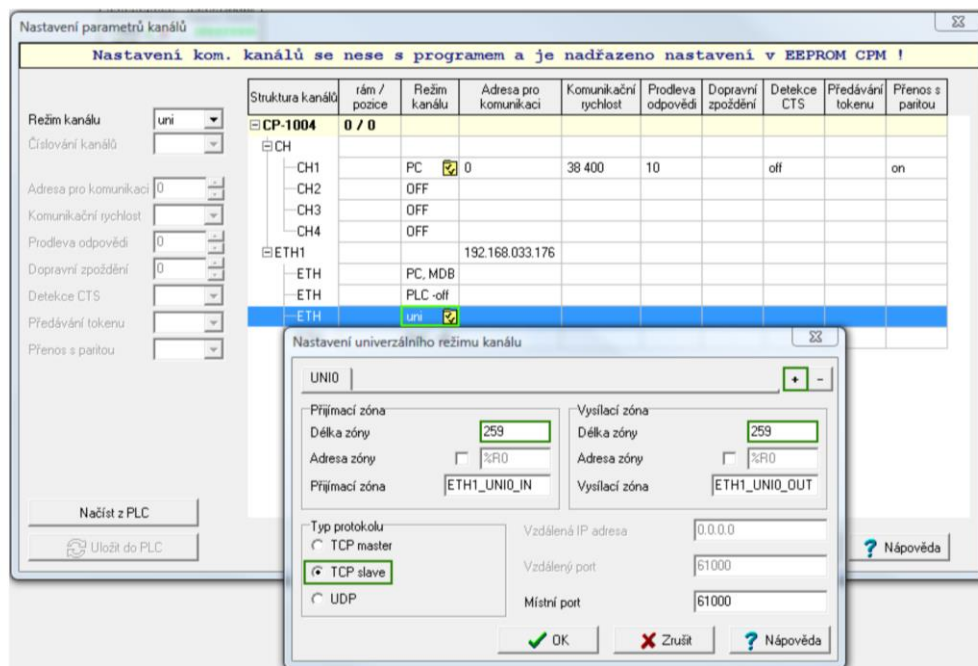
Standardně má PLC Tecomat na ethernetovém rozhraní aktivní dvě spojení TCP a UDP v režimu MDB, které zpracovávají Modbus příkazy. Voláním tohoto bloku se TCP spojení deaktivují, aby nedocházelo ke kolizi výchozího ovladače a funkčního bloku. Pro úplnou deaktivaci režimu MDB (TCP i UDP) lze použít funkci [fcModbusTcpUdpOff](#).



Popis proměnných :

Proměnná	Typ	Význam
VAR_INPUT		
<i>UnitID</i>	USINT	Identifikátor slave zařízení
<i>chanCode</i>	UINT	kód kanálu (CH1_uni, ..., CH10_uni)
<i>port</i>	UINT	Číslo místního portu (502)
<i>inputsCnt</i>	UINT	Počet diskretních vstupů (počet BOOLů)
<i>coilsCnt</i>	UINT	Počet diskretních výstupů (počet BOOLů)
<i>inputRegCnt</i>	UINT	Počet vstupních registrů (počet WORDů)
<i>holdingRegCnt</i>	UINT	Počet registrů (počet WORDů)
VAR_OUTPUT		
<i>read</i>	BOOL	Byla čtena data
<i>write</i>	BOOL	Byla zapsána data
<i>established</i>	BOOL	TCP spojení navázáno
<i>lastCmd</i>	USINT	Poslední přijatý příkaz
<i>lastAddr</i>	UINT	Poslední čtená/zapisovaná adresa
<i>lastQuantity</i>	UINT	Poslední počet čtených/zapisovaných objektů
<i>err</i>	BOOL	Chyba
<i>errCode</i>	USINT	Chybový kód
<i>msgCnt</i>	UDINT	Počet zpracovaných zpráv
<i>errCnt</i>	UDINT	Počet chyb
<i>excCnt</i>	UDINT	Počet výjimek
VAR_IN_OUT		
<i>inputs</i>	BOOL	První diskretní vstup v poli (musí být BOOL)
<i>coils</i>	BOOL	První diskretní výstup v poli (musí být BOOL)
<i>inputRegs</i>	UINT	První vstupní registr v poli
<i>holdingRegs</i>	UINT	První registr v poli

Zvolené spojení v režimu UNI musí být nastaveno na typ protokolu TCP slave. Minimální délka přijímací a vysílací zóny je **259** bytů. Místní port může být nastaven na libovolnou hodnotu, protože bude doplněný dle vstupu *port* funkčního bloku.



Příklad nastavení spojení UNI0 pro blok ModbusTCPslave (další spojení mohou být přidána tlačítkem +)