

**UNIVERZITA PARDUBICE**  
Fakulta elektrotechniky a informatiky

**ŘÍZENÍ NĚKTERÝCH VELIČIN MODELU BYTOVÉ JEDNOTKY  
POMOCÍ PLC**

Bc. Tomáš Bizek

Diplomová práce  
2017

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

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš Bizek**  
Osobní číslo: **I15175**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Řízení procesů**  
Název tématu: **Řízení některých veličin modelu bytové jednotky pomocí PLC**  
Zadávající katedra: **Katedra řízení procesů**

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

#### Postup:

Cílem práce je vytvoření modelu bytové jednotky pro simulaci vývoje veličin relevantních pro jeho obyvatele (teplota, osvětlení, ...) a následné řízení těchto veličin pomocí vybraného PLC. Součástí řešení bude uživatelský panel pro příjemné ovládání řídicího software.

#### Teoretická část:

Stručná rešerše principů modelování technologických procesů, popis problematiky PLC, možnosti programování PLC, popis podobných řešení.

#### Praktická část:

Vytvoření komplexního softwarového simulátoru bytové jednotky z pohledu sledování veličin relevantních pro obyvatele, zprovoznění komunikace PC - PLC, vytvoření programu do PLC realizujícího kompletní řídicí algoritmus pro smart ovládání bytové jednotky a pro automatické řízení vybraných veličin. Tvorba software do ovládacího panelu aplikace. Zprovoznění a vyhodnocení úlohy.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**KWASNIEWSKI, J. Programmable Logic Controllers. Cracow: ROMA-POL, 2002. ISBN 83-86320-45-1.**

**MIKLEŠ, J., FIKAR, M. Modelovanie, identifikácia a riadenie procesov I. Bratislava : Vydavateľstvo STU, 1999. 168 s. ISBN 80-227-1289-2**

Vedoucí diplomové práce:

**Ing. Petr Doležel, Ph.D.**

Katedra řízení procesů

Datum zadání diplomové práce: **24. října 2016**

Termín odevzdání diplomové práce: **17. května 2017**



Ing. Zdeněk Němec, Ph.D.  
děkan



L.S.



Ing. Daniel Honc, Ph.D.  
vedoucí katedry

V Pardubicích dne 15. listopadu 2016

## **Prohlášení**

Prohlašuji:

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

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

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

V Pardubicích dne 23. 05. 2017

Bc. Tomáš Bizek

## **Poděkování**

Tímto bych chtěl poděkovat svému vedoucímu práce Ing. Petru Doleželovi, Ph.D. za veškerou pomoc a ochotu při tvorbě této práce.

V Pardubicích dne 23. 05. 2017

Bc. Tomáš Bizek

## **ANOTACE**

*Diplomová práce se věnuje vytvoření modelu bytové jednotky, který bude obsahovat určité veličiny. K těmto veličinám patří ovládání osvětlení, barva světla, intenzita a místnost, která bude osvětlena. Další dostupnou veličinou je ovládání pozice zámku u dveří a změna pinu pro tyto dveře. V pořadí třetí možností je volba pozice rolet na oknech a poslední veličinu, kterou lze v modelu bytové jednotky řídit je teplota vzduchu. K demonstraci je využito PLC a HMI od firmy Siemens.*

## **KLÍČOVÁ SLOVA**

*model bytové jednotky, ovládání, PLC, HMI, simulátor*

## **TITLE**

*CONTROL OF APARTMENT WITH PLC*

## **ANNOTATION**

*The work is oriented to create the model of apartment which contains some quantities to control. These quantities include switching the lights on and off, choosing of color and intensity. Another available quantity is locking the door and changing of PIN for this door. The third possibility is control of a position for the sunblind and the last quantity to control is the temperature of air in the apartment. PLC and HMI from the company Siemens are used for controlling of the apartment*

## **KEYWORDS**

*Model of apartment, Control, PLC. HMI, Simulation*

## OBSAH

Seznam zkratk a značek .....	9
Seznam symbolů proměnných veličin a funkcí.....	10
Seznam ilustrací .....	11
Seznam tabulek .....	13
ÚVOD .....	14
1 PROGRAMOVATELNÉ AUTOMATY .....	15
1.1 Historie.....	15
1.2 Struktura HW .....	15
1.3 SW obsah PLC .....	16
1.4 Programovací jazyk .....	17
2 SIEMENS SIMATIC .....	21
2.1 Siemens Simatic S7-1500 .....	21
2.2 Siemens Simatic HMI TP700 Comfort.....	22
3 MODEL BYTOVÉ JEDNOTKY .....	24
3.1 Třída lighting .....	24
3.2 Třída lockDoor.....	25
3.3 Třída sunBlind .....	27
3.4 Třída heater .....	28
4 KOMUNIKACE .....	31
4.1 OPC Server .....	31
4.1.1 Nastavení OPC serveru .....	32
4.2 Skript RozhraniMatlabPLC .....	35
5 VÝVOJOVÉ PROSTŘEDÍ PLC A HMI .....	37
5.1 Uživatelské rozhraní na panelu HMI .....	37
5.1.1 Hlavní obrazovka .....	39
5.1.2 Obrazovka ovládání světel.....	40
5.1.3 Obrazovka ovládání zámku a změny pinu .....	41
5.1.4 Obrazovka ovládání rolet.....	43
5.1.5 Obrazovka ovládání topení .....	44
5.2 Programy pro PLC .....	45
5.2.1 Komunikace .....	46
5.2.2 Ovládání světel .....	48
5.2.3 Ovládání zámku a změny pinu.....	48

5.2.4	Ovládání rolet .....	52
5.2.5	Ovládání topení.....	53
6	SIMULAČNÍ SCÉNÁŘ .....	57
7	ZÁVĚR.....	61
	LITERATURA .....	62
	PŘÍLOHY .....	63

## SEZNAM ZKRATEK A ZNAČEK

CPU	centrální procesorová jednotka
FBD	Function Block Diagram
LD	Ladder Diagram
Matlab	Matrix Laboratory
HMI	Human machine interface
PLC	Programmable Logic Controller
PID	proporcionálně integračně derivační regulátor
ST	Structured Text
TIA Portal	Totally Integrated Automation Portal

## SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

<i>power</i>	stav osvětlení (Matlab)
<i>intensity</i>	hodnota intenzity (Matlab)
<i>numintensity</i>	hodnota intenzity v RGB spektru (Matlab)
<i>colour</i>	výběr barvy (Matlab)
<i>open</i>	stav otevření zámku
<i>close</i>	stav zamčení zámku
<i>rgcolour</i>	hodnota podsvícení zámku
<i>up</i>	stav pohybu rolety vzhůru
<i>down</i>	stav pohyby rolety dolů
$m_R$	hmotnost vzduchu v místnosti, kg
$c_{air}$	tepelná kapacita vzduchu, J/kg·°C
$Q_{gain}$	termální energie topení, J
$Q_{loss}$	termální energie ztrát do okolí, J
$T_R$	teplota místnosti, °C
$T_H$	teplota vzduchu z topení, °C
$T_{Out}$	venkovní teplota, °C
$R$	tepelný odpor, J/s·m·°C
$\lambda$	tepelná vodivost, J/s·m·°C
$S_Z$	plocha zdí, m <sup>2</sup>
$S_O$	plocha oken, m <sup>2</sup>
$S_D$	plocha dveří, m <sup>2</sup>
$D_D$	hloubka dveří, m
$D_O$	hloubka oken, m
$D_Z$	hloubka zdí, m
<i>heatAct</i>	aktuální teplota vzduchu z topení (TIA Portal)
<i>doorOpen</i>	stav otevřeno pro dveře (TIA Portal)
<i>doorClose</i>	stav zavřeno pro dveře (TIA Portal)
<i>doorMainNotification</i>	výběr hodnoty pro hlavní oznámení stavu pinu (TIA Portal)
<i>doorTestPin</i>	hodnota testovacího pinu (TIA Portal)
<i>windowMethod</i>	výběr metody pohybu rolet (TIA Portal)

## SEZNAM ILUSTRACÍ

Obr. 1.1 – Blokové schéma standardního modulárního PLC (Zezulka 2004).....	16
Obr. 1.2 – Cyklický režim PLC (Zezulka 2004).....	18
Obr. 1.3 – LD jazyk v prostředí TIA Portal V14.....	18
Obr. 1.4 – FBD jazyk v prostředí TIA Portal V14 .....	19
Obr. 1.5 – ST jazyk v prostředí TIA Portal V14.....	20
Obr. 2.1 – Siemens Simatic S7-1500 CPU 1515-2 PN (Siemens AG 12/2014).....	21
Obr. 2.2 – Simatic HMI varianta s klávesnicí a s dotykový displejem (Siemens AG 04/2012) .....	22
Obr. 2.3 – Uživatelské prostředí panelu TP700 (Siemens AG 04/2012).....	23
Obr. 3.1 – Grafický model bytové jednotky .....	24
Obr. 3.2 – Skript třídy lighting .....	25
Obr. 3.3 – Skript třídy lockDoor.....	26
Obr. 3.4 – Skript třídy sunBlind .....	27
Obr. 3.5 – Funkce simulate pro třídu heater .....	30
Obr. 3.6 – Funkce diferenciální rovnice .....	30
Obr. 4.1 – Rozhraní KEPServerEX V5 .....	32
Obr. 4.2 – Nastavení zařízení.....	33
Obr. 4.3 – Struktura v OPC.....	33
Obr. 4.4 – Nastavení vlastností tagu lightPowerAll .....	34
Obr. 4.5 – Definování OPC serveru v prostředí MATLAB.....	35
Obr. 4.6 – Přiřazení tagům jednotlivé proměnné z OPC server .....	35
Obr. 4.7 – Setter a getter pro nastavení/získání hodnoty z OPC serveru.....	36
Obr. 5.1 – Spojení mezi PLC a HMI .....	38
Obr. 5.2 – Vizualizace hlavní obrazovky .....	39
Obr. 5.3 – Vizualizace obrazovky pro ovládání světel.....	40
Obr. 5.4 – Vizualizace pro ovládání zámku.....	42
Obr. 5.5 – Vizualizace pro změnu pinu .....	43
Obr. 5.6 – Vizualizace obrazovky pro ovládání rolet .....	44
Obr. 5.7 – Vizualizace obrazovky pro ovládání topení .....	45
Obr. 5.8 – Komunikační blok TCON .....	46
Obr. 5.9 – Konfigurace bloku TCON .....	46
Obr. 5.10 – Seznam definovaných tagů v prostředí TIA Portal .....	47

Obr. 5.11 – Program pro osvětlení.....	48
Obr. 5.12 – Program odemknutí zámku.....	49
Obr. 5.13 – Nastavení hodnot pro výběr oznámení v případě špatného pinu nebo odemknutí	50
Obr. 5.14 – Program změny pinu.....	51
Obr. 5.15 – Definování doby zobrazení hlášení o úspěšné změně pinu .....	52
Obr. 5.16 – Program pro ovládání pohybu rolet .....	52
Obr. 5.17 – Blok PID regulátoru.....	53
Obr. 5.18 – Funkční blok pro rampu .....	54
Obr. 5.19 – Spuštění programu chlazení.....	55
Obr. 5.20 – Řízení množství vzduchu topení.....	56
Obr. 6.1 – Regulační průběh teploty pro venkovní teplotu 6 °C .....	57
Obr. 6.2 – Regulační pochod pro venkovní teplotu 30 °C.....	58
Obr. 6.3 – Regulační pochod pro venkovní teplotu -10 °C .....	59
Obr. 6.4 – Parametry PID regulátoru .....	60

## **SEZNAM TABULEK**

Tab. 3.1 – Rozměry a konstanty bytové jednotky .....	29
--	----

# ÚVOD

První část této práce se zabývá problematikou PLC. Popisuje stručně jejich historii, charakteristické vlastnosti, způsoby využití, výhody a nevýhody. Popis zahrnuje i možnosti programování PLC. Po tomto obecném úvodu je představen model S7-1500 od firmy Siemens.

Jako další zařízení, které představuje hlavní část této práce, je HMI. Zde jsou opět popsány základní obecné charakteristiky a využití. Demonstrace je pak provedena na modelu od firmy Siemens, Simatic TP700 Comfort.

Tento hardware je pak následně použit k hlavnímu cíli této práce. Jedná se o řízení různých veličin v modelu bytové jednotky, který je vytvořen v prostředí MATLAB. V příslušné kapitole jsou popsány jednotlivé části, ze kterých se model vytvořený v MATLABu skládá.

Samostatnou kapitolu zabírá i komunikace, která je řešena právě mezi modelem a řídicím systémem (PLC, HMI) Pro tento účel je využit KEPServerEX V5 od společnosti Kepware Technologies, který zajistí správu nad OPC serverem.

Mezi prvky určené k řízení v bytové jednotce patří světla, jejich barva a intenzita, poloha rolet na oknech, správa dveří a řízení teploty v celé jednotce. Pro tyto veličiny jsou vytvořeny jednotlivé obrazovky na ovládacím dotykovém panelu, které obsahují potřebné prvky k řízení daných veličin.

Předposlední kapitola této práce je pak věnována jednotlivým programům, které obsluhují ovládání zvolených vlastností bytové jednotky. Podrobně je popsáno jejich provedení ve vývojovém prostředí, které je určeno pro zvolené PLC a HMI, jímž je TIA Portal V14.

Následně jsou zvolené tři simulační scénáře, které testují chování vytápění bytové jednotky při různých venkovních teplotách. Ty jsou zvoleny tak, aby reprezentovaly přibližnou hodnotu teploty v jednotlivých ročních obdobích.

V závěru je následovně shrnut názor autora na práci ve vývojovém prostředí TIA Portal V14 od firmy Siemens, společně s vyhodnocením výsledků simulačních scénářů.

# 1 PROGRAMOVATELNÉ AUTOMATY

## 1.1 HISTORIE

Významnější nástup PLC (Programmable Logic Controllers) započal v první polovině 80. let, kdy se začali využívat pro řízení technologických procesů, strojů a výrobních linek. V té době byly konkurencí pro PLC řídicí počítače a minipočítače. Ty měli výhodu v podobě lepšího programátorského komfortu, nicméně PLC nabízela vyšší spolehlivost, nižší náklady na kabeláž, snazší údržbu, přehlednost, lepší ladění programů, a především modulární stavbu, kdy bylo možné výběrem HW optimalizovat konečnou cenu zařízení.

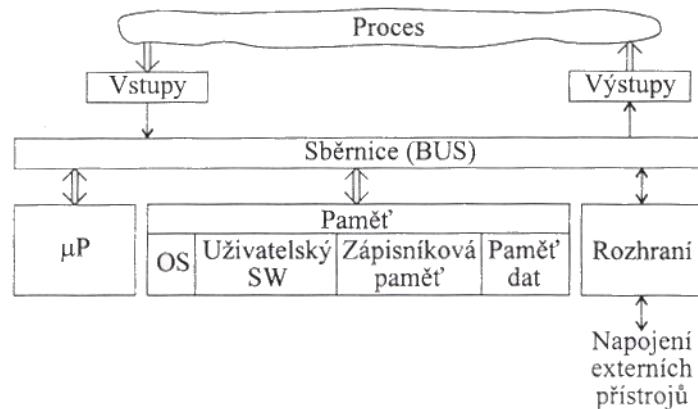
Nahrazena byla i tzv. malá automatizace, která zahrnovala bezkontaktní logiku, relé logiku a průmyslové regulátory. Hlavním požadavkem na tento systém byl pak způsob jeho programování. Programovací jazyk měl být blízký logickým a reléovým schémátům, booleovským rovnicím. Tím došlo k nahrazení předchozím způsobu prezentování bezkontaktní a reléové logiky, která řešila příslušné rovnice fyzickým propojením. Tedy pokud měla být zavedena změna struktury, bylo nutné změnit i fyzické zapojení členů. Naproti tomu v programovatelném automatu stačí danou úpravu rovnice změnit v programu automatu.

V případě nahrazení řídicího počítače nebylo nahrazení pomocí PLC tak rychlé. Jakmile byla zvýšená spolehlivost jednotlivých komponent automatu, chybělo k plnému nahrazení řídicích počítačů a minipočítačů vylepšit programátorský komfort. To bylo splněno s příchodem SCADA systémů. Závěrem tedy jenom rychlé shrnutí výhod a nevýhod programovatelných automatů. Mezi výhody patří rychlá přeprogramovatelnost, flexibilita, modularita, jednoduchý programovací jazyk, spolehlivý a jednoduchý operační systém reálného času. K nevýhodám se dá zahrnout již zmínění nižší programátorský komfort v porovnání s IPC, kde je i příznivější cena přístroje s ekvivalentním výkonem (Zezulka 2004).

## 1.2 STRUKTURA HW

Jako HW jádro využívá PLC slovně orientovaný mikroprocesor. Pro řízení logické úrovně je zapotřebí, aby v základu bylo vybaveno nutnými funkcemi, a to časovou funkcí a funkční čítání impulsů. Tyto funkce zajistí příslušné časovače a čítače. Pro komunikaci mezi vstupy/výstupy a vnitřním HW slouží 16 nebo 32 bitová sběrnice.

Stavba paměti programu a paměti dat byla v případě prvních PLC rozdělena. Dnešní PLC vlastní společnou jednu operační paměť, která má vyhrazené prostory pro data ze vstupů a výstupů, vnitřních proměnných a další paměťový prostor pro vlastní program. Nachází se zde i funkční bloky, systémové funkce a uživatelské funkce. Schéma standardního modulárního PLC je zobrazeno na obr.1.1.



Obr. 1.1 – Blokové schéma standardního modulárního PLC (Zezulka 2004)

### 1.3 SW OBSAH PLC

Strukturovaný uživatelský program je jednou z charakteristických vlastností programovaného automatu. Jeho hlavním obsahem jsou uživatelské bloky a systémové bloky. Následné řídicí programy zahrnují základní moduly, kterými jsou organizační bloky (OB), funkční bloky (FB), funkce (FC) a datové bloky (DB). Jedná se o moduly z uživatelské kategorie.

Následně budou popsány základní vlastnosti těchto bloků. Organizační blok vytváří rozhraní mezi operačním systémem CPU a vytvořeným uživatelským programem. Tento blok se volá v každém cyklu, který vyvolává operační systém. Provedení uživatelského programu právě obstarává tento blok, který při startu cyklu spouští daný program. Výbava PLC nenabízí pouze jeden typ tohoto bloku, ale uživatel má na výběr z více variant. Mezi ně patří Time-of-Day Interrupt OB, který může být spuštěn v určitý den a čas nebo periodicky v závislosti na zvolených časových intervalech. Cycle Interrupt OB má vlastnost, která umožňuje spouštět části programu po stejně velkých časových intervalech. Další základní bloky jsou například,

Time-Delay Interrupt, Status Interrupt. Hardware Interrupt. Speciálním případem organizačního bloku je pak OB100, který je volán po kompletním restartu PLC.

Funkční bloky (FB) jsou funkce nebo určité sekvence logických funkcí v bloku, které mají přiřazenou paměť. Ta slouží k ukládání hodnot proměnných. Velmi podobné funkčním blokům jsou funkce (FC). Rozdílem je pouze to, že funkce nepotřebuje instanci datové bloku, jelikož časově omezené proměnné ukládají svoji hodnotu v zásobníkové paměti. To platí v případě, že je funkce stále vykonávána, jakmile dojde k ukončení funkce, proměnné se odstraní.

Datové bloky (DB) neobsahují žádné logické instrukce, v porovnání s předchozími bloky. Jejich obsahem jsou uživatelská data, se kterými pracuje uživatelský program. Nachází se zde i jednotlivé instance datových bloků, zmíněné v předchozích blocích, které jsou právě přiřazená k daným funkčním blokům. Tímto jsou popsány všechny uživatelské bloky.

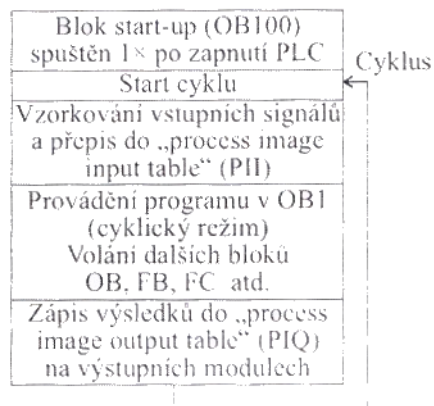
Dále byly zmíněny i systémové bloky. Tuto skupiny zastupují systémové funkce, systémové funkční bloky a systémové datové bloky. Jejich vlastností je zajistit základní chod PLC, kdy se může jednat o nastavování parametrů pro moduly, podpory datové komunikace nebo kopírování funkcí. V případě datových bloků se jedná o uložení informací, které se týkají nastavení systémů a hodnot parametrů modulů (Zezulka 2004).

## 1.4 PROGRAMOVACÍ JAZYK

Jak vyplývá z kapitoly 1.3, vykonávání v programů v PLC se provádí v tzv. programové smyčce. Tedy pracuje v určitých cyklech, ve kterých je daný program stále dokola vykonáván. Tato vlastnost umožňuje programátorovi neřešit návrat programu na start a jeho opětovný běh po vykonání. Délku tohoto cyklu pak nastavuje právě programátor, dojde-li k její překročení, je tato událost vyhodnocena jako chyba. Příklad typického cyklického vykonání programu je uveden na obr. 1.2.

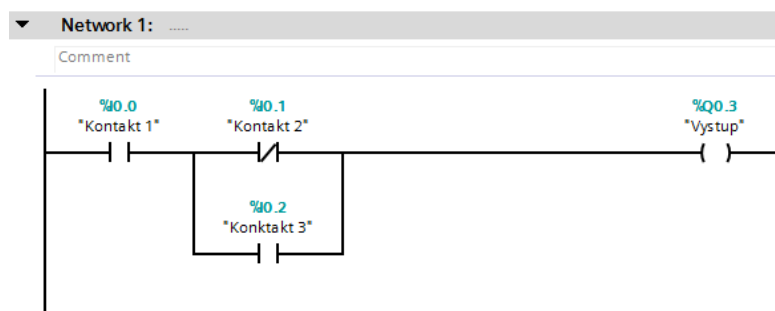
Základním požadavkem na programovací jazyk bylo, aby umožnil projektantům přejít z fyzického provedení logické úrovně řízení na řešení booleovských rovnic programem pro programovatelný automat. K dispozici jsou tedy dva typy programovacích jazyků, a to grafické a algebraické. Skupina grafických jazyků obsahuje například jazyk reléových schémat (ang. Ladder Diagram (LD)).

Jednotlivé logické operace jsou vyobrazeny v obvyklé formě, která je použita pro kreslení schémat při práci s reléovými a kontaktními prvky. Velkou výhodou tohoto jazyka je jasně definovaná posloupnost programu, která zahrnuje i přehlednost.



Obr. 1.2 – Cyklický režim PLC (Zezulka 2004)

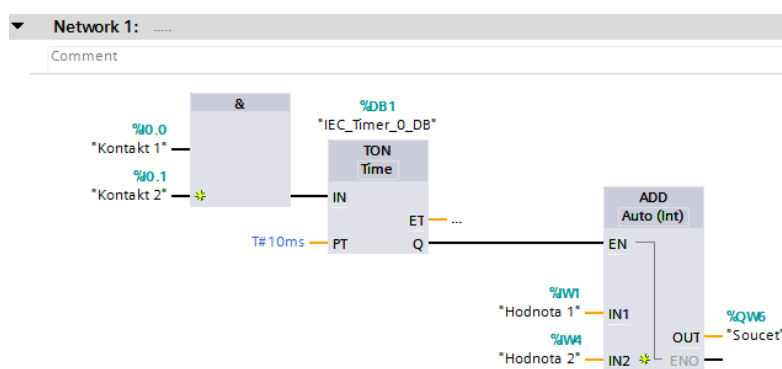
Tyto vlastnosti se velice dobře využijí při programování jednodušších a menších logických úloh. Výhodou je i zobrazení tzv. „vodivé cesty“, kdy tuto schopnost lze využít při servisní kontrole a zjistit například vadný spínač nebo tlačítko. Při používání v složitějších úlohách může dojít k nepřehlednosti programu a jeho nárůstu, který se týká velikosti. V tomto případě je doporučeno naplno využívat komentářů a jednotlivé větve popisovat. Názorná ukázka LD jazyka je na obr. 1.3, kdy se jedná o prostředí TIA Portal V14.



Obr. 1.3 – LD jazyk v prostředí TIA Portal V14

K dalším grafickým programovacím jazykům pro PLC patří jazyk logických schémat (ang. Function Block Diagram (FBD)). Svým způsobem se jedná o vyšší variantu LD jazyka, kdy využíváme místo sériově-paralelního zapojení kontaktů relé přímo klasické značky hradel AND, OR, XOR, NAND, NOR. Bloky jsou například reprezentovány i čítače, časovače, posuvné registry, paměťové registry, které mají své specifické značky.

Tento jazyk je vhodný pro uživatele, kteří jsou zvyklí na kreslení logických schémat pro zařízení s integrovanými obvody. Ukázka FBD jazyka z prostředí TIA Porta V14 je na obr. 1.4.



Obr. 1.4 – FBD jazyk v prostředí TIA Portal V14

Předchozí uvedené jazyky patří k těm nejvýznamnější a k nezákladnějším grafickým jazykům. V případě algebraických jazyků je nezákladnějším zástupce jazyk seznamu instrukcí (ang. Instruction List (IL)), který se velice podobá assembleru, tedy jazyku pro programování mikrokontrolérů. Takový program je pak složen ze sekvencí daných instrukcí, kdy každá instrukce začíná na novém řádku. Mezi základní instrukce patří například MOV, která se využívá pro přesun dat mezi registry, ADD sloužící pro součet dvou hodnot nebo SUB pro odečet hodnot. Programátor má při použití tohoto způsobu programování velký vliv nad strukturou programu a umožňuje účinnou optimalizaci programu. Nevýhodou se může jevit velké množství psaní a velké množství příkazů. To může způsobovat nevýhodu při psaní složitějších aplikací.

Klasičtí programátoři mikroprocesorů pak určitě využijí možnost programování ve strukturovaném textu (ang. Structured Text (ST)). Tento programovací režim je tvořen posloupností symbolických instrukcí. Jedná se o vyšší jazyk, který je podobný například jazyku C nebo Pascalu. Využití toho zápisu je vhodné pro práci s daty, řetězci a databázemi, zahrnující náročné zpracování analogových signálů.

Pro správnou funkčnost je zapotřebí znát příkazy a přesnou syntaxi zápisu. Pokud aplikace vyžaduje zpracování velkých množství logických signálů, je tento způsob méně vhodný v porovnání například s grafickými jazyky. Náhled na strukturu ST je reprezentován na obr. 1.5.

Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Input							
2	InitFB	Bool	false	Non-ret...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	SetPoint	Real	0.0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

```

1 #rv1:= RD_SYS_T(#ActualTime);
2
3 IF #InitFB = true THEN
4     #LastTime := #ActualTime;
5     #AuxOut := #SetPoint;
6     #Out := #AuxOut;
7 END_IF;
8
9 #DeltaTime_T:= T_DIFF(IN1:=#ActualTime, IN2:=#LastTime);
10 #DeltaTime_DI:= TIME_TO_DINT(#DeltaTime_T);
11 #DeltaTime:= DINT_TO_REAL(#DeltaTime_DI);
12 #DeltaTime := #DeltaTime / 1000.0;
13 #LastTime := #ActualTime;
14
15 IF #SetPoint < #AuxOut THEN
16     #Out := #AuxOut;
17     #auxR := #AuxOut - (#Rate * #DeltaTime);
18     #Out := #AuxOut;
19     #AuxOut := MAX(IN1 := #auxR, IN2 := #SetPoint);
20     #Out := #AuxOut;
21     RETURN;
22 ELSE
23 IF #SetPoint > #AuxOut THEN
24     #Out := #AuxOut;
25     #auxR := (#Rate * #DeltaTime) + #AuxOut;
26     #Out := #AuxOut;
27     #AuxOut := MIN(IN1 := #auxR, IN2 := #SetPoint);
28     #Out := #AuxOut;
29 END_IF;
30 END_IF;
31 RETURN;

```

Obr. 1.5 – ST jazyk v prostředí TIA Portal V14

Pro programování byl začátkem 90. let zaveden standard pro neutrální programování PLC s názvem IEC 1311. Nejedná se o závaznou normu, nýbrž pouze o doporučení od International Electrotechnical Commission. Tento standard se skládá z pěti částí, první část zahrnuje obecnou část, požadavky na technické vybavení a testy je popsání v druhé části. Programovací jazyky pak definuje část číslo tři. Pro popis návodu pro uživatele je vyhrazena část čtyři. Poslední částí je pak komunikace. Hlavní úlohou tohoto standardu a jeho částí je zajistit, aby programové vybavení PLC mělo tyto specifické vlastnosti. Nezávislost programu na technickém vybavení, tedy aby bylo možné použít jakýkoliv HW. Zaručit možnost ladění SW již ve fázi návrhu a umožnit jednotný programátorský přístup. Poslední vlastností je pak strukturovanost a modularita. Tyto vlastnosti jsou obsaženy především v části tři popsaného standardu IEC 1311-3 (Zezulka 2004).

## 2 SIEMENS SIMATIC

Pro potřeby této práce byla využita technika od společnosti Siemens z její řady Simatic. Důvodem tohoto výběru byla dlouholetá tradice a kvalita výrobků od této firmy. Velkou výhodou je i široká komunita, kterou lze nalézt na stránkách výrobce pod označením support.

Konkrétně v této práci je využito PLC řady S7-1500 a dotykový panel HMI TP700 Comfort. Tyto výrobky patří mezi nejvyšší řadu, kterou firma Siemens nabízí. Jejich základní popis je uvede v následujících kapitolách.

### 2.1 SIEMENS SIMATIC S7-1500

Řada S7-1500 patří mezi nejvyšší řadu, kterou firma Siemens nabízí na poli PLC. V případě této práce byl použit model s CPU 1515-2 PN, tedy výkonnostně střed této třídy. Modelová řada začíná CPU 1511-1 PN a končí modelem CPU 1518F-4 PN/DP. Odlišnost jednotlivých typů spočívá především v procesním čase pro bitovou operaci, kde je hodna u nejslabšího modelu 60 ns a u nejsilnějšího 1 ns.

Mezi další rozdíly patří velikost pracovní paměti, počet PROFINET rozhraní a PROFIBUS rozhraní. V porovnání například s modelovou řadou S7-1200, která má pouze jeden Ethernet port, je přítomnost více těchto portů vítaným zlepšením. Umožňuje jednodušší propojení například s HMI panelem a PC, kde u řady S7-1200 bylo zapotřebí řešit problém, pokud bylo nutné mít zapojené zároveň HMI panel a PC. Zde bylo řešením použít externí switch nebo router.



Obr. 2.1 – Siemens Simatic  
S7-1500 CPU 1515-2 PN  
(Siemens AG 12/2014)

Zvolený typ CPU 1515-2 PN, zobrazen na obr.2.1, nabízí dvě PROFINET rozhraní, kdy první má dva Ethernet porty a druhé rozhraní disponuje jedním Ethernet portem. Hlavní změnou oproti řadě S7-1200 je přítomnost malého předního displeje a šesti operačních tlačítek. Díky tomu lze nastavit některé parametry PLC bez nutnosti připojení k PC. Jedná se o nastavení IP adres, času a data, kontrola verze firmware, výsledky diagnostik a chyb, které jsou uvedeny v bufferu.

Výhoda, kterou získává starší řada S7-1200, je přítomnost přímo digitálních vstupů/výstupů a analogových vstupů/výstupů na daném automatu. Tímto novější řada S7-1500 nedisponuje a je zapotřebí použít rozšiřovací I/O moduly. Ovšem při použití ve velkém průmyslu jsou tyto moduly samozřejmostí, jelikož i počet těchto vstupů a výstupů, které jsou přítomné na S7-1200, je velmi nedostačující (Siemens AG 12/2014).

## 2.2 SIEMENS SIMATIC HMI TP700 COMFORT

HMI (Human Machine Interface) je LCD dotykový displej, který slouží pro komunikaci s PLC, pro nastavování parametrů ze strany obsluhy, zobrazování hodnot zvolených veličin apod. Často se nachází ve dvou provedení, nezáleží na výrobci, varianty jsou velice podobné. První varianta je ovládána pomocí klávesnice, která je umístěna přímo na panelu. Může se jednat například pouze o pět tzv. funkčních kláves jejichž vlastnosti určuje programátor. Nebo u vyšších modelů lze najít plnohodnotnou numerickou klávesnici, společně s navigačními

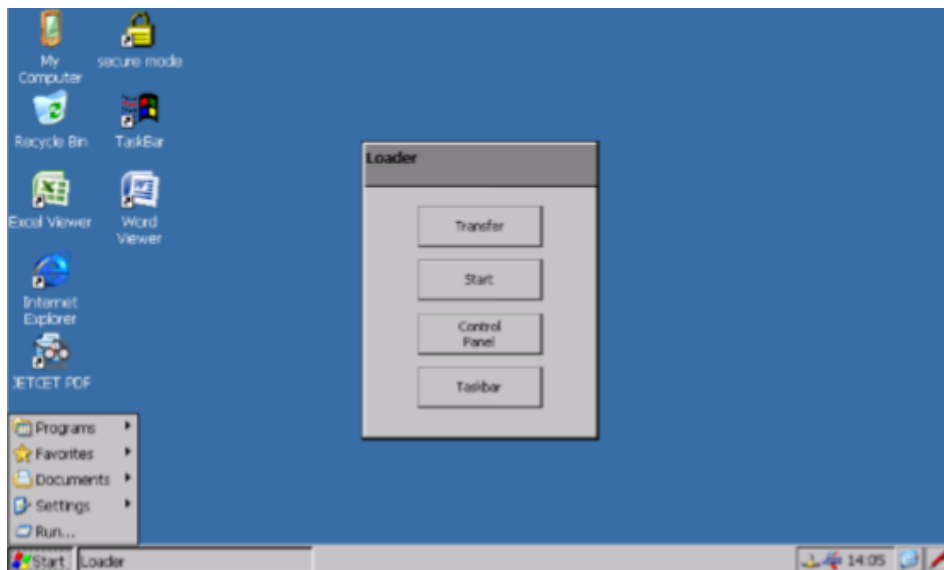


Obr. 2.2 – Simatic HMI varianta s klávesnicí a s dotykový displejem (Siemens AG 04/2012)

klávesami, doplněné o funkční klávesy. Velikost displejů je velice rozmanitá a záleží na výrobci, v případě Siemens jsou tyto panely nabízeny ve velikostech od 4“ až po 15“.

Druhá varianta je nabízená s dotykovou obrazovkou, tím pádem odpadá potřeba fyzické klávesnice a lze přístroj provést v kompaktnějším provedení. Z pohledu programátora tato možnost nabízí určitý stupeň volnosti, který zahrnuje volbu počtu ovládacích prvků, které budou přítomné na obrazovce. Nicméně pro některé aplikace v průmyslu může být přítomnost fyzické klávesnice potřebná. Co se velikosti týká pro dotykové panely, jejich obrazovky se pohybují od 7“ až do velikosti 22“, to je už velikost klasického stolního monitoru. Vyskytuje se pak i kombinace obou variant, kdy panel vlastní jak dotykový displej, tak například funkční klávesy. V této práci je použito HMI řady TP700 Comfort. Jedná se tedy o variantu s dotykovým displejem. Označení 700 reprezentuje rozměr displeje, který činí 7“.

Důležitým hardware vybavením jsou jednotlivá komunikační rozhraní. Na tomto modelu lze využít dva Ethernet porty, dvě USB zdířky typu A, jedno USB typu mini B, komunikační modul RS422/485. Nachází se zde i dva 3,5 mm jack porty pro audio line IN/OUT a zdířka pro SD karty. Tím se dostáváme k software vybavení zařízení, kterým je Windows CE Desktop (Siemens AG 04/2012).



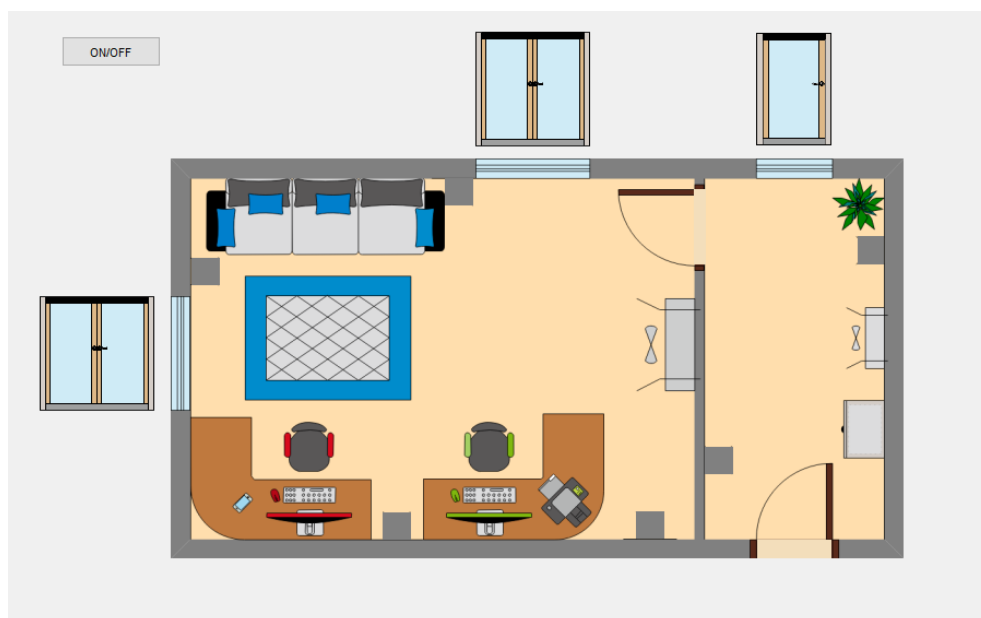
Obr. 2.3 – Uživatelské prostředí panelu TP700 (Siemens AG 04/2012)

Prostředí toho operačního systému je ukázáno na obr. 2.3. Díky tomuto software se zde nachází předinstalované programy, kterými jsou například Excel Viewer, Word Viewer, PDF Viewer a Media Player. Tyto programy lze využít k čtení manuálů, příruček a různých zpráv, například o sběru dat z výroby, nebo záznamů výrobního cyklu pomocí videa, přímo na panelu.

### 3 MODEL BYTOVÉ JEDNOTKY

Cílem této práce je řízení veličin v bytové jednotce. Ta byla vytvořena v prostředí MATLAB ve verzi R2015a, za pomoci objektově orientovaného programování. Základní jednotkou tohoto programování je objekt, který nese vlastnosti, metody a možnost identifikace průběhu určité události. Byly vytvořené třídy určitých vlastností bytové jednotky, které jsou vzorem či šablonou pro jednotlivé objekty.

Patří mezi ně třída heater, lighting, lockDoor, sunBlind a poslední třída RozhraniMatlabPLC, která se zabývá komunikací a je blíže popsána v kapitole 4. Jednotlivé třídy jsou popsány v následujících podkapitolách. Návrh vzhledu bytové jednotky je reprezentován na obr. 3.1.



Obr. 3.1 – Grafický model bytové jednotky

#### 3.1 TŘÍDA LIGHTING

Mezi atributy této třídy patří stav napájení (*power*), výběr intenzity osvětlení (*intensity*), v případě intenzity je tu ještě jeden atribut, který nastavuje hodnotu RGB dané barvy (*numintensity*) a pro výběr již zmíněné barvy je zde atribut *colour*. Počet barev je omezen pouze na čtyři, červenou, zelenou, modrou a žlutou. V třídě se nachází settery a gettery pro jednotlivé atributy, ty umožní nastavování jejich hodnot.

Ve funkci doLight() se do pomocných proměnných uloží hodnoty *power*, *intensity* a *colour*. Podle *colour* se ve switch vybere příslušná barva a hodnota *intensity* se využije pro nastavení spektra RGB, která se pomocí setteru pak nastaví do atributu *numintensity*. Tato hodnota mění vlastnosti, barvu, zvoleného objektu, který v modelu prezentuje svítidlo. Příslušný skript třídy je zobrazen na obr. 3.2.

```

47 | %Funkce tridy
48 | methods
49 |     function doLight(this) %Zapnutí světla, barva, intenzita
50 |         inte = double(this.getIntensity());
51 |         powe = this.getPower();
52 |         colo = this.getColour();
53 |         if powe == 0
54 |             this.setNumintensity(1/255*[128,128,128]);
55 |         else
56 |             switch colo
57 |                 case 0 %zluta
58 |                     pom = double(1/255*([255,255,inte]));
59 |                     this.setNumintensity(pom);
60 |                 case 1 %cervena
61 |                     pom = double(1/255*([255,inte,inte]));
62 |                     this.setNumintensity(pom);
63 |                 case 2 %modra
64 |                     pom2 = double(128+(inte/2));
65 |                     pom = double(1/255*([inte,pom2,255]));
66 |                     this.setNumintensity(pom);
67 |                 case 3 %zelena
68 |                     pom = double(1/255*([inte,255,inte]));
69 |                     this.setNumintensity(pom);
70 |             end
71 |         end
72 |     end
73 | end

```

Obr. 3.2 – Skript třídy lighting

## 3.2 TŘÍDA LOCKDOOR

Třída reprezentuje zamykání a odemykání dveří. Při vytváření byla myšlena jako pouze elektrický zámek, který lze ovládat, a který dokáže předat informaci o svém stavu, tedy jestli je v poloze zamčeno nebo otevřeno. Veškerá důležitá logika jako je změna pinu, jeho uchování, kontrola pinu, jsou ovládané v PLC. Proto je tato třída poměrně jednoduchá a obsahuje pouze tři atributy, těmi jsou *open*, *close* a *RGcolour*.

První dva zmíněné atributy mohou nabývat pouze dvou hodnot, 0 a 1, jsou tedy booleovského typu. Třetí atribut nese hodnotu pro barevné spektrum v RGB formátu, to signalizuje stav zámku, červená barva pro zamknuto, zelená pro odemčeno. Za definicí hodnot atributů, která je v konstruktoru, následují jednotlivé gettery a settery. Hlavní funkcí třídy je pak `statusDoor`, která čte hodnoty atributů `open`, `close` a podle nich nastavuje hodnotu `RGcolour`, která reprezentuje stav zámku. Část skriptu třídy `lockDoor` je uvedena na obr. 3.3.

```

8  | □ | methods
9  |   | % Konstruktor
10 | □ | function this = lockDoor()
11 | - |     this.open = 0;                                %signal pro odemknuti
12 | - |     this.close = 1;                               %signal pro zamknuti
13 | - |     this.RGcolour = 1/255*[255,0,0];             %zakladni barva objektu
14 |   |
15 | - |     end
16 |   |
17 |   | % Settery a Gettery
18 | □ | function [RGcolor] = getRGColor(this)
19 | - |     RGcolor = this.RGcolour;
20 | - |     end
21 | □ | function setOpen(this,open)
22 | - |     this.open = open;
23 | - |     end
24 | □ | function [open] = getOpen(this)
25 | - |     open = this.open;
26 | - |     end
27 | □ | function setClose(this,close)
28 | - |     this.close = close;
29 | - |     end
30 | □ | function [close] = getClose(this)
31 | - |     close = this.close;
32 | - |     end
33 |   |
34 |   | % Hlavni funkce tridy
35 | □ | function statusDoor(this)
36 | - |     if (this.open == 1)
37 | - |         this.RGcolour = 1/255*[0,255,0];
38 | - |     end
39 | - |     if (this.close == 1)
40 | - |         this.RGcolour = 1/255*[255,0,0];
41 | - |     end
42 | - |     end
43 |   |
44 | - | end

```

Obr. 3.3 – Skript třídy `lockDoor`

### 3.3 TRÍDA SUNBLIND

Další možnost řízení, kterou by mohla nabízet bytová jednotka, je ovládání polohy rolet, v modelu reprezentovány třídou sunBlind. Pomocí pěti atributů se určuje pozice a pohyb rolety. Pro pohyb směrem nahoru je určen atribut *up*, pro pohyb dolů je zde atribut *down*. Ty získávají od uživatele jeho volbu pro pohyb, jejich hodnota nabývá, stejně jako v případě třídy lockDoor a jejich dvou atributů, 0 nebo 1.

```
53 %Funkce tridy
54 methods
55     function PositionUP(this)           %Pohyb nahoru
56         if (this.up == 1);
57             this.setResetUp(0);
58             basic = this.getPos();      %Zisk souradnic objektu
59             if basic(4) > 6
60                 basic(4) = basic(4)-1;
61                 basic(2) = basic(2)+1;
62                 this.setPos(basic);
63             else
64                 this.setUp(0);
65                 this.setResetUp(1);
66             end
67         else
68             this.setResetUp(1);
69         end
70
71
72     end
73     function PositionDOWN(this)        %Pohyb dolu
74         if (this.down == 1);
75             this.setResetDown(0);
76             basic = this.getPos();      %zisk souradnic objektu
77             if basic(4) < 105
78                 basic(4) = basic(4)+1;
79                 basic(2) = basic(2)-1;
80                 this.setPos(basic);
81             else
82                 this.setDown(0)
83                 this.setResetDown(1)
84             end
85         else
86             this.setResetDown(1);
87         end
88     end
89 end
```

Obr. 3.4 – Skript třídy sunBlind

K získání polohy žaluzie a k následnému výpočtu pro její pohyb je použit atribut *pos*. Ten pomocí setteru, který je pro něj definován, získává souřadnice pozice objektu, kterým je v modelu reprezentována vybraná žaluzie. Souřadnice jsou v 4 rozměrném vektoru, kdy první dva rozměry definují pozici objektu v souřadném systému, tedy hodnotu x a y souřadnice. Zbylé dva určují rozměr objektu, šířku (ang. width) a výšku (ang. height).

Pro výpočet vlastního pohybu žaluzie, který probíhá ve dvou funkcích třídy pro každý pohyb, jsou zapotřebí dvě hodnoty z vektoru souřadnic. Jedná se o y souřadnici a hodnotu výšky daného objektu. K objasnění výpočtu je důležité uvést, že hodnoty jednotlivých souřadnic jsou uvedeny v pixelech. Pokud je zvolen pohyb nahoru, volá se funkce PositionUP, nastaví se nulová hodnota resetovacího signálu, který zde slouží pro ukončení pohybu, a začne se provádět samostatný výpočet.

V podmínce cyklu if, je nastavena minimálně hodnota výšky objektu, tedy čtvrtého parametru. Následně se provede odečet jednoho pixelu od hodnoty výšky objektu, která je uložena v pomocné proměnné, a zároveň se přičte jeden pixel k hodnotě y souřadnice. Tím nedochází k posouvání objektu po plátně. Analogický postup je i v případě, kdy je volána funkce PositionDOWN, rozdílem je hodnota podmínky a výpočet přičítá k výšce objektu jeden pixel, zatím co od souřadnice y odčítá. Části skriptu, který obsahuje funkce a definice atributů, je uveden na obr. 3.4.

### 3.4 TRŽDA HEATER

Základem této třídy je výpočet teploty vzduchu uvnitř v místnosti, za pomoci diferenciální rovnice (3.1). Zde bylo zapotřebí definovat potřebné vlastnosti bytové jednotky, mezi které patří její rozměr, tepelnou vodivost materiálů, ze kterých jsou vyrobeny zdi, dveře a okna

$$\frac{dT_R}{dt} = \frac{1}{m_R \cdot c_{air}} \cdot \left( \frac{dQ_{gain}}{dt} - \frac{dQ_{loss}}{dt} \right), \quad (3.1)$$

kde  $Q_{gain}$  – termální energie topení

$Q_{loss}$  – termální energii ztracena do okolí.

Výpočet těchto dvou energií je definován rovnicemi (3.2) a (3.3) (MathWorks 2017)

$$\frac{dQ_{gain}}{dt} = \frac{dm_{Ha}}{dt} \cdot c_{air} \cdot (T_H - T_R), \quad (3.2)$$

$$\frac{dQ_{loss}}{dt} = \frac{(T_R - T_{Out})}{R}. \quad (3.3)$$

Rozměry celé místnosti a jednotlivé konstanty jsou uvedeny v tab. 3.1 Při tvorbě této třídy bylo využito z části modelu, který je prezentován na stránkách [MathWorks](#). V následující tabulce jsou uvedeny všechny konstanty, které byly zapotřebí k výpočtu. Jako je například tepelná vodivost, která posloužila k výpočtu jednotlivých tepelných odporů materiálů. Ty pak pomocí vzorce pro paralelní zapojení odporů určili hodnoty celkového tepelného odporu pro celou bytovou jednotku.

Tab. 3.1 – Rozměry a konstanty bytové jednotky

Rozměry bytové jednotky	Plocha zdí	$S_Z = 171,36 \text{ m}^2$
	Plocha oken	$S_O = 3,84 \text{ m}^2$
	Plocha dveří	$S_D = 3,20 \text{ m}^2$
	Síla dveří a oken	$D_D = D_O = 0,01 \text{ m}$
	Síla zdí	$D_Z = 0,20 \text{ m}$
	Sklo stavební	$\lambda = 0,760 \text{ J/s}\cdot\text{m}\cdot^\circ\text{C}$
Tepelná vodivost materiálů	Dřevo	$\lambda = 0,075 \text{ J/s}\cdot\text{m}\cdot^\circ\text{C}$
	Cihelná hmota	$\lambda = 0,510 \text{ J/s}\cdot\text{m}\cdot^\circ\text{C}$
Tepelná kapacita vzduchu		$c_{\text{air}} = 1005.4 \text{ J/kg}\cdot^\circ\text{C}$
Hmotnost vzduchu v místnosti		$m_R = 95 \text{ kg}$
Celkový tepelný odpor		$R = 1,322 \cdot 10^{-3} \text{ J/s}\cdot\text{m}\cdot^\circ\text{C}$
	Hodnota v simulaci	$R = 2,204 \cdot 10^{-5} \text{ J/min}\cdot\text{m}\cdot^\circ\text{C}$

Z důvodu snížení času trvání simulace při získávání dat, která jsou pak vyhodnocena v kapitole 7, byla zvolena, označena v tabulce tab. 3.1, hodnota celkového tepelného odporu, která udává hodnotu odporu v časové jednotce jedné minuty.

Nyní k programové části skriptu, který je zobrazen na obr. 3.5. Jednotlivé atributy mají přiřazené hodnoty daných konstant a funkcí simulate je volána simulace výpočet teploty vzduchu pomocí diferenciální rovnice (3.1)

```
function simulate(this, input, time)
    dr = @(t, TR) this.difRov(t, TR, input);
    [time,TR] = ode45(dr, [0 time], this.tr(end));
    this.t = [this.t(1:end-1), (this.t(end)+time)'];
    this.tr = [this.tr(1:end-1), TR'];
    this.T_heatCelk = [this.T_heatCelk(1:end-1) ones(size(time))' * input(1)];
    this.M_heatCelk = [this.M_heatCelk(1:end-1) ones(size(time))' * input(2)];
end
```

Obr. 3.5 – Funkce simulate pro třídu heater

Samotná funkce pro diferenciální rovnici je volána anonymně ve funkci simulate. Funkce diferenciální rovnice je pak v třídě heater vedena jako privátní metoda. Tedy, že tuto funkci může volat pouze samotná třída, nikoliv uživatel.

```
methods (Access = private)
    function [dTR] = difRov(this, t, TR, input)
        T_heat = input(1);
        M_heat = input(2);
        Q_gain = M_heat*this.c_air*(T_heat - TR);
        Q_loss = (TR - this.T_outside)/this.R;
        dTR = (Q_gain-Q_loss)/(this.m_room*this.c_air);
    end
end
```

Obr. 3.6 – Funkce diferenciální rovnice

## 4 KOMUNIKACE

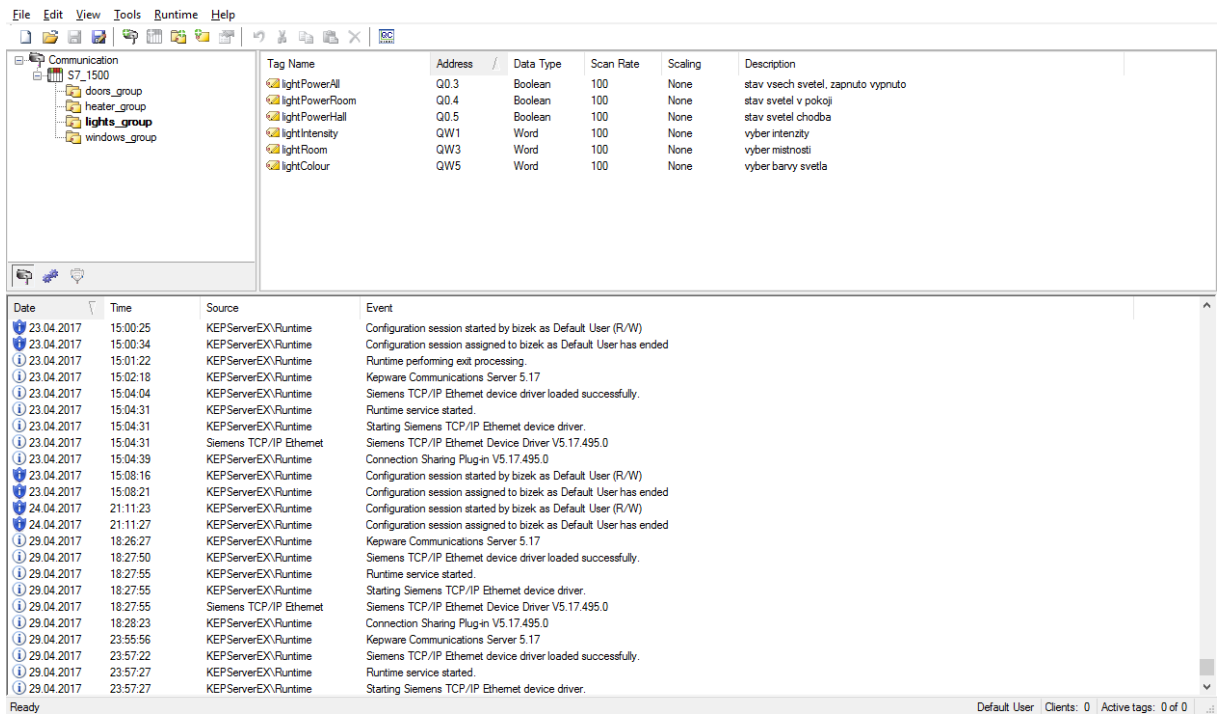
Mezi další cíl této kapitoly je popsat způsob komunikace, který je použit mezi vybraným PLC a modelem vytvořeným v MATLABu. Komunikace mezi HMI a PLC zde není uvedena, jelikož se jedná o poměrně triviální věc. Technologie, která umožňuje v této práci potřebnou komunikaci, se nazývá OPC server. Společně i s potřebným skriptem, který obstarává předávání dat mezi OPC a modelem, jsou uvedeny v následujících podkapitolách.

### 4.1 OPC SERVER

Jedná se o technologii, která je hojně využívána v automatizačním průmyslu a představuje pro něj, a jemu podobným odvětvím, komunikační standard. Mezi hlavní výhody tohoto standardu je jeho univerzálnost, kdy není zapotřebí mít hardware a software od jednoho specifického dodavatele. Díky tomu nabízí uživateli mnohem otevřenější a kompaktnější řešení.

Základní princip OPC serveru je následovný. Uživatel, označován jako OPC Client, má na výběr několik OPC serverů, které mohou být spravovány různými poskytovateli. Na jednotlivých serverech se nachází jednotlivé kanály, pod kterými jsou zahrnuta daná zařízení. Každé definované zařízení pak obsahuje určité skupiny, ty seskupují jednotlivé položky. Díky této struktuře je uživateli umožněno efektivní uspořádání a následná efektivní správa. Položky pak reprezentují tzv. tagy, to jsou proměnné, o jejichž hodnoty má uživatel zájem, a které sleduje (OPC Foundation 2017).

V případě této práce byl použit software od firmy Kepware Technologies, na jejichž stránkách je možné získat, po registraci, demo verzi OPC serveru zvanou KEPServerEX V5. I když se jedná o demo verzi, veškeré funkce a vlastnosti aplikace nejsou omezeny, hlavní rozdílem od plné verze je dvouhodinový limit provozu v propojeném režimu. Po uplynutí limitu je zapotřebí restartovat aplikaci, pak je znovu k dispozici limit o stejné hodnotě. Rozhraní aplikace je zobrazeno na obr. 4.1.



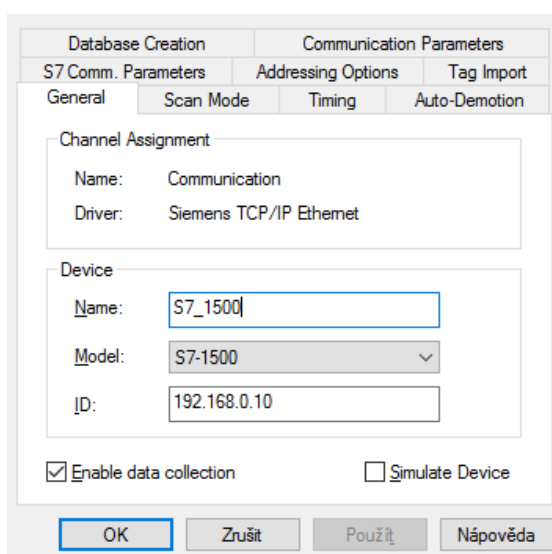
Obr. 4.1 – Rozhraní KEPServerEX V5

#### 4.1.1 Nastavení OPC serveru

Jak již bylo uvedeno, v této práci je využita verze programu KEPServerEX s označením V5. V nynější době je už přítomná novější verze s označením V6.1, nicméně autor práce zůstal u starší verze, jelikož novější demo verzi provázely technické problémy na hardware, který měl autor k dispozici. Rozdíly mezi jednotlivými verzemi nejsou pro použití v této práci vůbec znatelné, jediná změna je pouze modernější grafika.

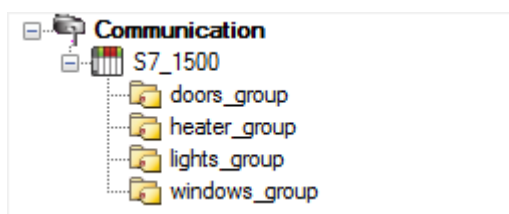
Prvním krokem nastavení je založení hlavního kanálu. Průvodce provede uživatele jednotlivými body nastavení. Nejdůležitějším bodem je výběr ovladače zařízení, kde v tomto případě je použit Siemens TCP/IP Ethernet. K dispozici jsou i další možnosti nastavení, například výběr síťového adaptéru, nastavení optimalizační metody pro zápis hodnot tagů. Lze s těmito parametry manipulovat, v případě práce zůstali v předdefinovaném stavu.

Do vytvořeného kanálu lze přidávat neomezený počet jednotlivých zařízení. Stejně jako u vytváření kanálu, je zde přítomný průvodce, který prochází jednotlivé kroky nastavení. Mezi hlavní parametry patří model použitého PLC, jelikož byl zvolen komunikační protokol od firmy Siemens, nabízí se uživateli jednotlivé modely řady S7. Po pojmenování a vybrání modelu zařízení musí uživatel nastavit IP adresu, která se nachází v kolonce ID. Specifické nastavení je zobrazeno na obr. 4.2. Pro další nastavitelné parametry zařízení byly použity předdefinované hodnoty.



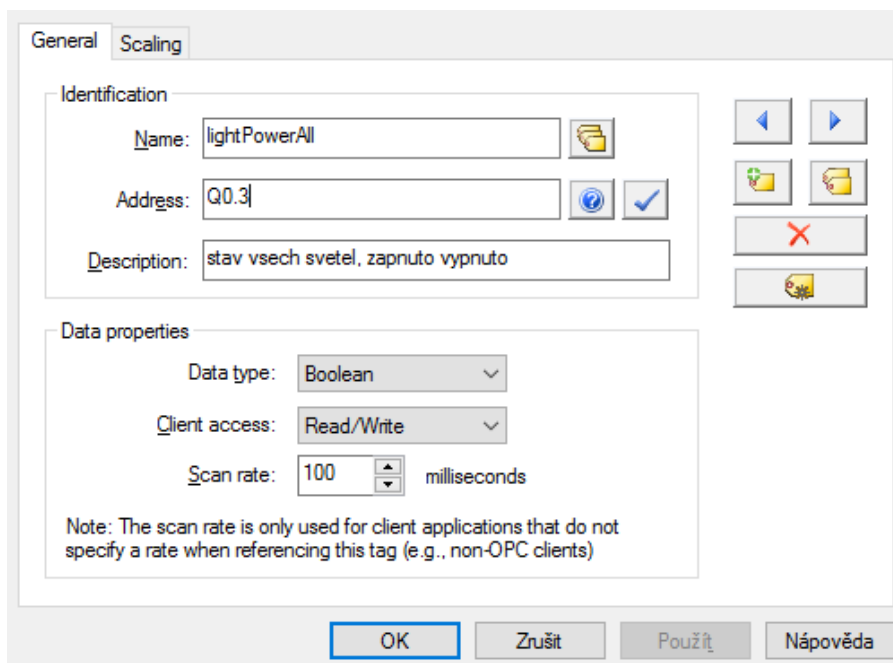
Obr. 4.2 – Nastavení zařízení

V tomto kroku je tedy definován komunikační kanál a použité PLC. Nyní může uživatel vytvářet jednotlivé skupiny, počet není omezen, a do nich umístit libovolný počet tagů. Skupiny lze rozdělit podle jednotlivých objektů, kterých se týkají, nebo podle ovládaných veličin. Rozdělení v této práci je ukázáno na obr. 4.3, kde se na vrcholu stromové struktury nachází komunikační kanál, pojmenován Communication, pod ním definované PLC S7-1500, které obsahuje jednotlivé veličiny pro ovládaní, rozříděné do skupin dle ovládaných objektů v bytové jednotce.



Obr. 4.3 – Struktura v OPC

Pro demonstraci je vybrána skupina tagů, který patří do skupiny lights\_group. Ta zajišťuje nastavení atributů pro ovládání světel v místnostech. Mezi vlastnosti tagu patří, jeho název, adresa, ta musí být shodná s adresou, která je použita v PLC, datový typ tagu a přístup klienta. Lze zkontrolovat i validitu dané adresy a v případě chyby je zde přítomná nápověda, která uživateli nabídne příklady adres pro určitý datový typ. Těch je na výběr celá řada, v této práci jsou především použity datové typy boolean, word, float. Ukázka nastavení vlastností tagu ve skupině lights\_group je prezentována na obr. 4.4 (Kepware 2016).



Obr. 4.4 – Nastavení vlastností tagu lightPowerAll

## 4.2 SKRIPT ROZHHRANIMATLABPLC

Po korektním nastavení OPC serveru v programu KEPServerEx, který vytvořil komunikační most mezi PLC a OPC, bylo zapotřebí provést podobný komunikační most mezi modelem v MATLABu a OPC. Stejně jako v případě tvoření modelu, byla vytvořena, pomocí objektově orientovaného programování, třída s názvem RozhraniMatlabPLC.

```
36 methods
37 function this = RozhraniMatlabPLC()
38 %Nainstalovani toolboxu OPC a spusteni OPC Serveru
39 if mpcchecktoolboxinstalled('opc')
40     opcreset
41     clear mpcopcPlatnStep;
42     clear mpcopcMPCstep;
43     try
44         this.opcServer = opcda('192.168.0.12','Kepware.KEPServerEX.V5');
45         connect(this.opcServer);
46     catch ME
47         disp('The Kepware.KEPServerEX.V5 must be running on the local machine. ');
48         return
49     end
50 end
```

Obr. 4.5 – Defínování OPC serveru v prostředí MATLAB

Počet atributů třídy odpovídá počtu použitých tagů v OPC severu. První funkcí metody je založení serveru, kdy je zkontrolováno, zda je nainstalován potřebný toolbox. V případě nepřítomnosti se provede dodatečná instalace. Následuje napojení na server, dle zadané IP adresy.

Dalším krokem je přiřazení každému atributu třídy jednotlivý tag, který se nachází na serveru. Je vhodné atribut a tag pojmenovat stejně. Důležitá je správná cesta k tagu, která musí obsahovat potřebný kanál, definované zařízení, skupinu v zařízení a název tagu, tak, jak je prezentováno na obr. 4.6

```
%Nadefinovani skupin a tagu pro dany OPC Server
%osvetleni
lights_group = addgroup(this.opcServer);
this.opcData.lightPowerAll = additem(lights_group, 'Communication.S7_1500.lights_group.lightPowerAll');
this.opcData.lightPowerRoom = additem(lights_group, 'Communication.S7_1500.lights_group.lightPowerRoom');
this.opcData.lightPowerHall = additem(lights_group, 'Communication.S7_1500.lights_group.lightPowerHall');
this.opcData.lightColour = additem(lights_group, 'Communication.S7_1500.lights_group.lightColour');
this.opcData.lightIntensity = additem(lights_group, 'Communication.S7_1500.lights_group.lightIntensity');
this.opcData.lightRoom = additem(lights_group, 'Communication.S7_1500.lights_group.lightRoom');
```

Obr. 4.6 – Přiřazení tagům jednotlivé proměnné z OPC server

Za touto definicí následují gettery a settery pro proměnné. Není nutné pro každou proměnnou definovat setter a getter. Pokud se jedná o výstupní hodnotu, z pohledu PLC, stačí pouze getter. V případě vstupní hodnoty, opět z pohledu PLC, je postačující setter. Na následujícím obr. 4.7, lze vidět strukturu obou zmíněných případů specificky pro ovládání pohybu rolet. Analogický postup se uplatní u všech následujících proměnných.

```
149 | | %getter chodba pro pohyb dolu
150 | | function windowSetHallDown = getWindowSetHallDown(this)
151 | |     pom = read(this.opcData.windowSetHallDown);
152 | |     this.windowSetHallDown = pom.Value;
153 | |     windowSetHallDown = this.windowSetHallDown;
154 | | end
155 | | %resetovací signaly zaluzii
156 | | %setter chodba
157 | | function setWindowResetHallUp(this, windowResetHallUp)
158 | |     this.windowResetHallUp = windowResetHallUp;
159 | |     write(this.opcData.windowResetHallUp, this.windowResetHallUp);
160 | | end
```

Obr. 4.7 – Setter a getter pro nastavení/získání hodnoty z OPC serveru

## **5 VÝVOJOVÉ PROSTŘEDÍ PLC A HMI**

Pro produkty firmy Siemens je použit software s názvem TIA Portal (Totally Integrated Automation Portal), který zahrnuje jednotné prostředí pro vytváření programů pro PLC a aplikací pro HMI. Jednotlivé komponenty lze získat samostatně, kdy PLC využívá STEP7 a HMI program WinCC. Obě tyto komponenty jsou dostupné v TIA Portal. Specificky v této práci je využíván TIA Portal ve verzi 14.

Nabízí dvě pracovní rozhraní, těmi jsou Portal view a Project view. V prvním zmiňovaném se nachází základní funkce, prostředky pro založení nového projektu, správa sítě, či identifikování připojeného zařízení a jeho online diagnostiku. Z prostřední Portal view se dá plynule přejít do Project view, které již přímo nabízí nástroje pro tvorbu, správu jednotlivých programů, aplikací, ať pro samostatné PLC nebo HMI.

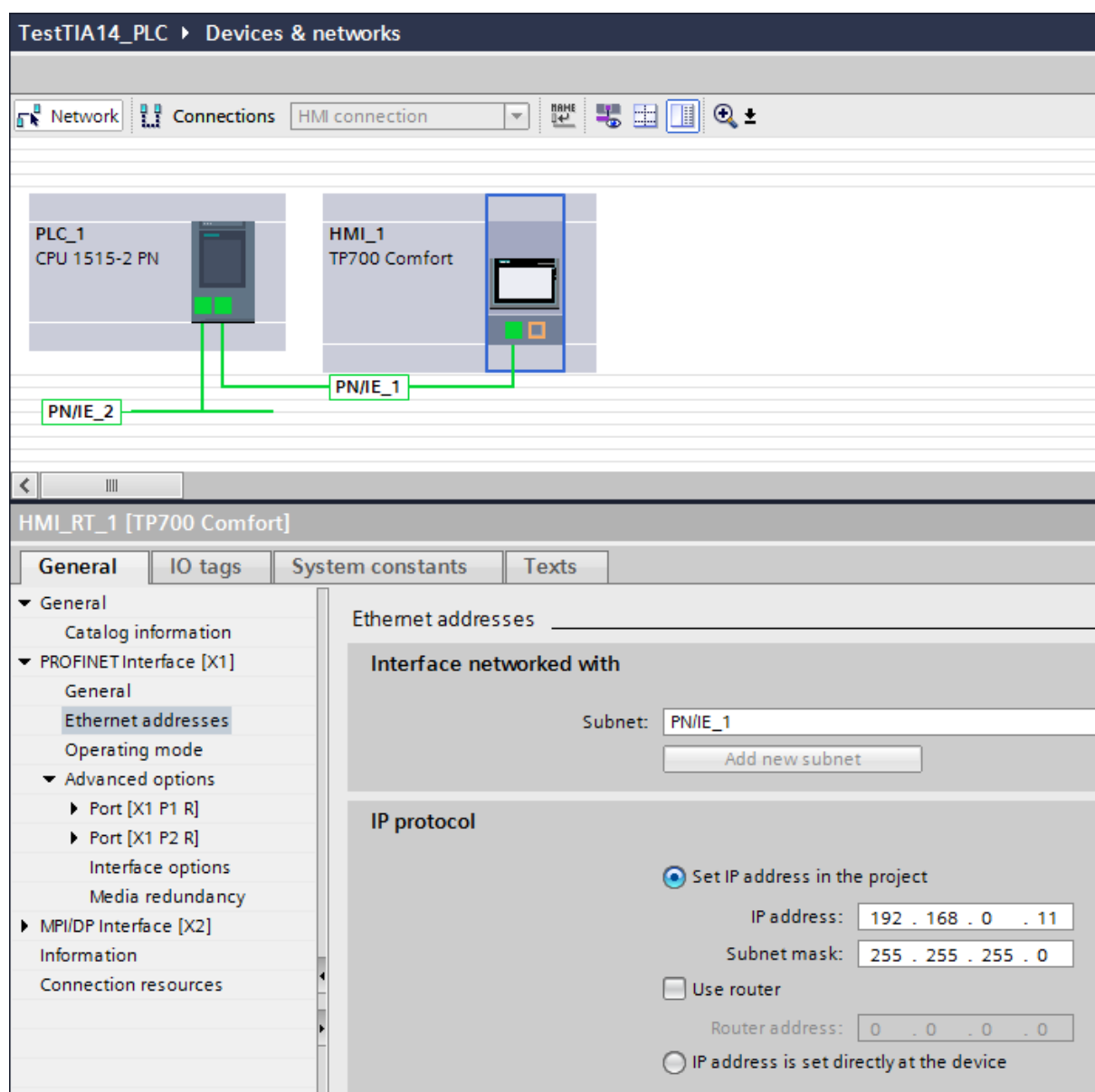
### **5.1 UŽIVATELSKÉ ROZHŘANÍ NA PANELU HMI**

Při vytváření aplikace pro HMI panel následují tyto kroky. Prvním z nich je přidání zařízení do projektu a jeho následná konfigurace. Přes Add new device, v rozhraní Portal view, je k dispozici seznam podporovaných HMI zařízení, programátor vybere příslušný panel a jeho verzi firmware. Pokud TIA Portal disponuje novější verzí firmware pro vybrané zařízení, nabídne programátorovi příslušnou upgrade, který se provede při nahrání první verze uživatelského programu.

V dalším kroku se zobrazí základní průvodce nastavením panelu, ten nabízí možnosti propojení s PLC, které v této chvíli ještě není definováno, nicméně možnost dodatečného spojení PLC a HMI je možná. Zbylé nabízené body pak již nastavují základní vzhled obrazovky. Patří mezi ně výběr základní šablony (ang. Screen layout), zobrazování alarmů, výběr počtu obrazovek, například, startovní obrazovka, ovládací obrazovky, systémová obrazovka a uživatelské obrazovky. Programátor má v tomto průvodci možnost výběru předem definovaných tlačítek, které se nachází v příslušné oblasti. Těmi jsou, tlačítko pro návrat na startovní obrazovku (Start screen button), změna jazyka (Language button), vypnutí pracovního režimu a přechod do prostřední Windows (Exit button).

Nicméně, tento průvodce se využívá pouze pro začáteční konfiguraci, veškeré kroky, které jsou v něm dostupné, lze následně provést i v rozhraní Project view. Nastavení IP adresy panelu se provádí v Device configuration, kde v záložce General zvolí kategorii PROFINET Interface. Doplňující propojení s příslušným PLC se nachází v Devices & Networks, které je provedeno zvolením portu jednoho zařízení a následným výběrem portem druhého zařízení. Propojení těchto zařízení, jak je prezentováno na následujícím obr. 5.1.

V následujících podkapitolách jsou popsány všechny použité obrazovky a jejich princip ovládání s návazností na PLC. Při vytváření vizualizace pro určitou technologii je vždy doporučeno dodržovat jednotný a uspořádaný styl jednotlivých obrazovek, z důvodu přehlednosti a jednotnosti z pohledu obsluhy.

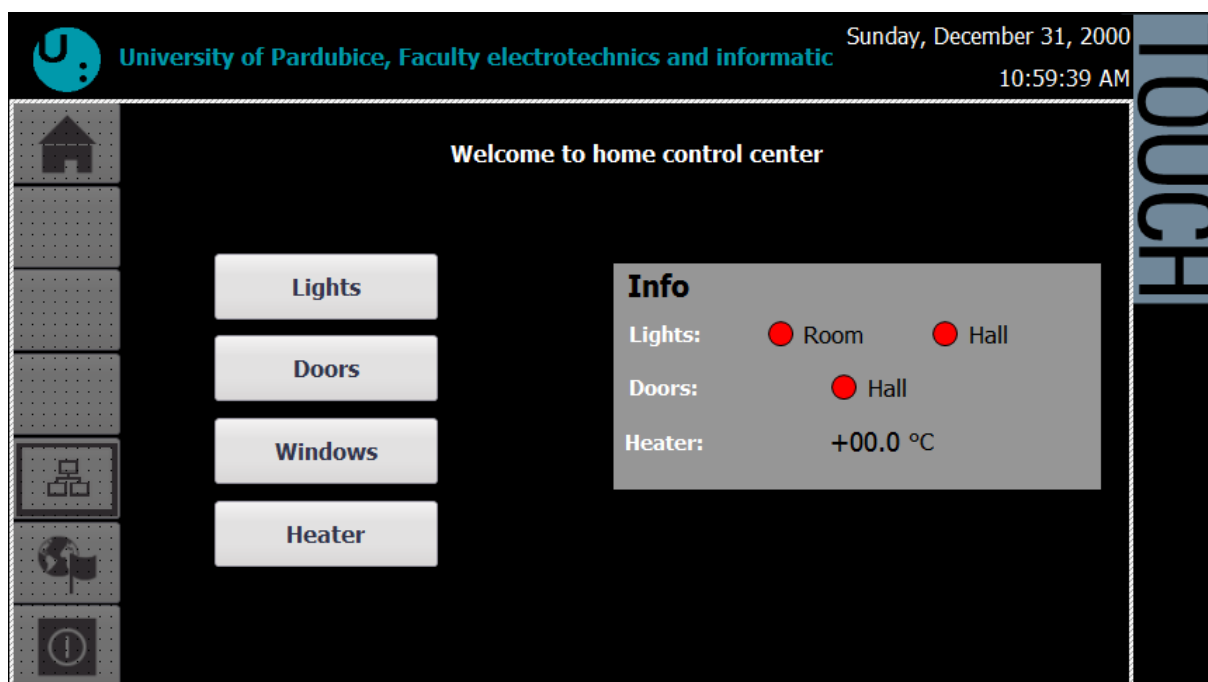


Obr. 5.1 – Spojení mezi PLC a HMI

### 5.1.1 Hlavní obrazovka

V programu označená jako Root screen, se tato obrazovka zobrazí při spuštění panelu. Na levé straně se nachází boční panel, ten obsahuje předefinovaná tlačítka, zmíněných v předchozí kapitole, jedná se o Start screen, Project information, Language, Exit. V hlavní části obrazovka nabídne uživateli čtyři hlavní tlačítka, která mají nastavenou událost přenesení na požadovanou stránku, při jejich zmáčknutí.

Dále se zde nachází jednoduchý informační panel. Jeho podstatou je prezentovat aktuální stav vybraných veličin uživateli v přehledné formě. Jedná se například o stav světel v místnostech, kdy zapnutý stav je označen zelenou barvou kruhového indikátoru, červená barva pak signalizuje stav vypnutý. Analogický princip je použit i v případě zámku na dveřích v chodbě, kde stav zamčeno je prezentován opět červenou barvou a stav odemčeno zelenou. Pro přehled aktuální teploty v místnosti, kterou má na starost topení, je její hodnota zobrazena ve výstupním poli. To je propojeno s tagem *heatAct*, který tuto hodnotu uchovává. Vizualizace hlavní obrazovky je na obr. 5.2.

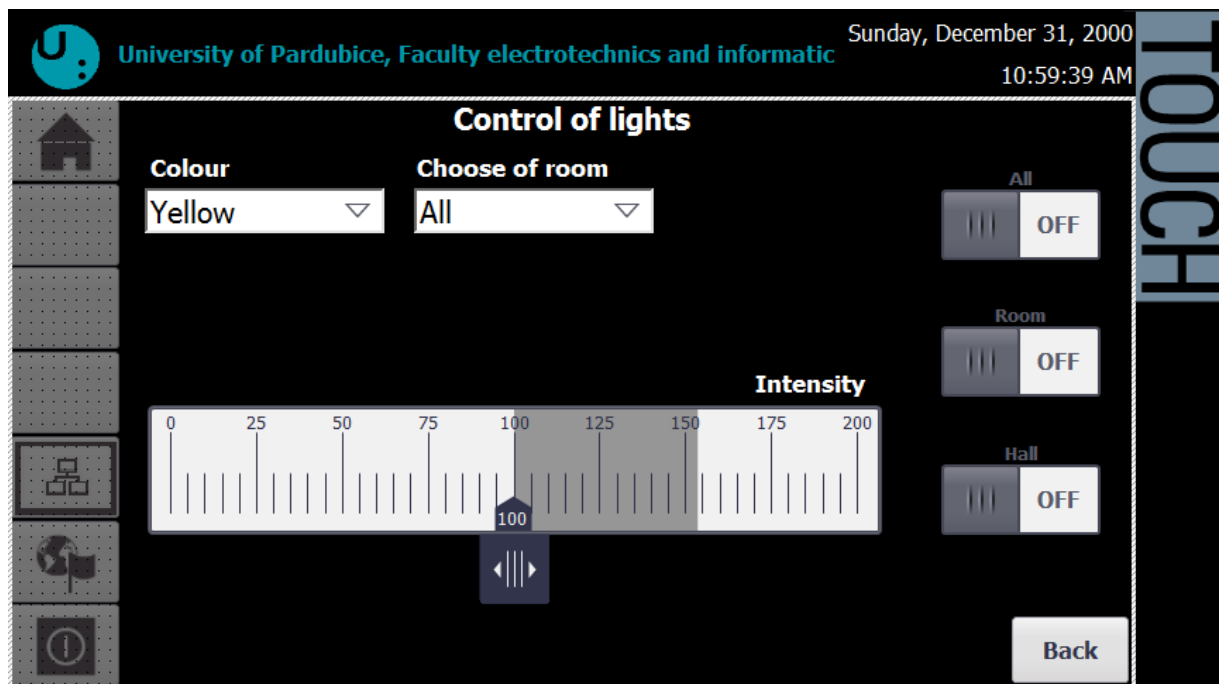


Obr. 5.2 – Vizualizace hlavní obrazovky

### 5.1.2 Obrazovka ovládání světel

První možností uživatele, kterou nabízí hlavní obrazovka je přesunutí na ovládací obrazovku světel. Zde jsou přítomny šest prvků, dvě symbolická vstupně výstupní pole (ang. Symbolic I/O field), posuvník (ang. Slider) a tři přepínací tlačítka (ang. Switch). Obě pole mají analogickou funkčnost. Obsahují textový seznam, kdy každé položce v seznamu je přiřazena určitá hodnota. Ta pak vystupuje z pole pod označením process value, a je navázaná na příslušný tag. Tyto pole slouží k výběru barvy a místnosti, která má být osvětlena.

Pomocí posuvníku je nastavována intenzita osvětlení ve vybrané místnosti, kdy nulová hodnota intenzity je definována, jako nejnižší dosažitelná intenzita při zapnutém osvětlení. Maximální hodnota posuvníku je nastavena na 200, důvodem této hodnoty je následovná použitelnost v modelu, kde je barva nastavována pomocí RGB spektra.



Obr. 5.3 – Vizualizace obrazovky pro ovládání světel

Při použití maximální hodnoty 100 pro intenzitu, která pak mohla reprezentovat procentuální vyjádření intenzity, nebyla změna barvy v modelu tolik znatelná. Proto autor zvolil hodnotu dvojnásobnou. Tato hodnota je opět provázána s příslušným tagem.

Posledními nepopsanými prvky, které se nachází na obrazovce jsou přepínací tlačítka. Každá místnost má svoje vlastní přepínací tlačítko sloužící k zapnutí osvětlení. Nicméně je k dispozici i třetí tlačítko, které zapne všechna dostupná světla. Vnitřní události tohoto tlačítka, označeno nadpisem All, jsou následovné. Pokud je přepnuto do stavu ON, nastaví hodnotu logické jedničky tagům, které jsou přiřazeny procesním hodnotám zbylých dvou tlačítek a tím je přepne do stavu ON. Zároveň nastaví hodnotu pole pro výběr místnosti na variantu All. Nyní uživatel nastavuje barvu a intenzitu pro obě místnosti. Pokud by ovšem chtěl obě místnosti mít osvětlené rozdílně, stačí změnit v poli pro výběr místnosti variantu. Při přepnutí tlačítka All do stavu OFF se varianta místnosti opět přepne na všechny místnosti, a tím zajistí vypnutí osvětlení v obou místnostech.

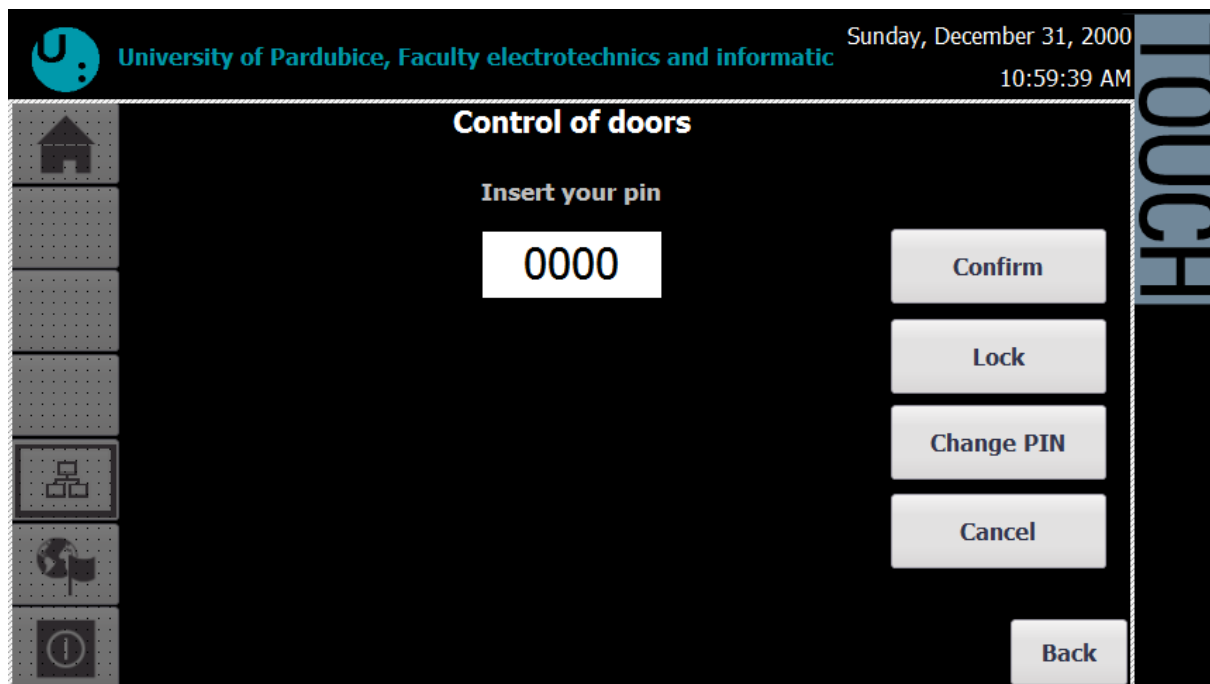
Zbývá dvě tlačítka obsahují jednodušší ovládání, jelikož každé obsluhuje jednotlivou místnost. Informace, zda je tlačítko v poloze ON nebo OFF, je opět přenášena příslušným tagem do programu v PLC. Zde, pokud je splněna podmínka, že tlačítko All má nulovou hodnotu, se nastaví příslušná hodnota tagu, který nese informaci o volbě místnosti. Tato informace je předána do panelu, kde se pole s volbou místnosti přepne na příslušnou variantu dle hodnoty. Tím není uživatel nucen vybírat ručně variantu v poli, pokud chce osvětlit pouze jednu místnost. S touto informací se nadále pracuje i v modelu bytové jednotky. Obrazovku pro ovládání světel se všemi prvky lze vidět na obr. 5.3.

### **5.1.3 Obrazovka ovládání zámku a změny pinu**

Pro ověření správnosti pinu a následnému odemčení dveří byla vytvořena primární obrazovka, na kterou se dostanu uživatel z hlavní obrazovky. Změnu pinu je dostupná na další obrazovce. V případě kontroly pinu obrazovka obsahuje jedno vstupní pole, do kterého uživatel zadává čtyřmístný pin. Po zadání následuje jeho uložení do datového bloku door\_data, který se nachází v PLC a je definován právě pro potřeby ovládání zámku a změnu pinu. Společně v tomto bloku je definován i správný, základní, pin, se kterým je porovnáván zadaný.

Na pravé straně se nachází čtyři ovládací tlačítka, kde první z nich, označeno Confirm, po stisknutí vyvolá funkci v PLC pro kontrolu uloženého a zadaného pinu. Pokud jsou tyto hodnoty shodné, pozadí tlačítka změní barvu na zelenou. Společně s ním se uživateli zobrazí hlášení, které se nachází pod zadávacím polem, o správnosti pinu a následném odemčení. V případě opačné situace, kdy se piny neshodují, se opět zobrazí hlášení, které informuje o nesprávném pinu a zamčení dveří. Pokud nastane tato situace, tlačítko Lock je podsvíceno červenou barvou. Tohoto efektu je dosaženo pomocí vlastnosti objektu tlačítka.

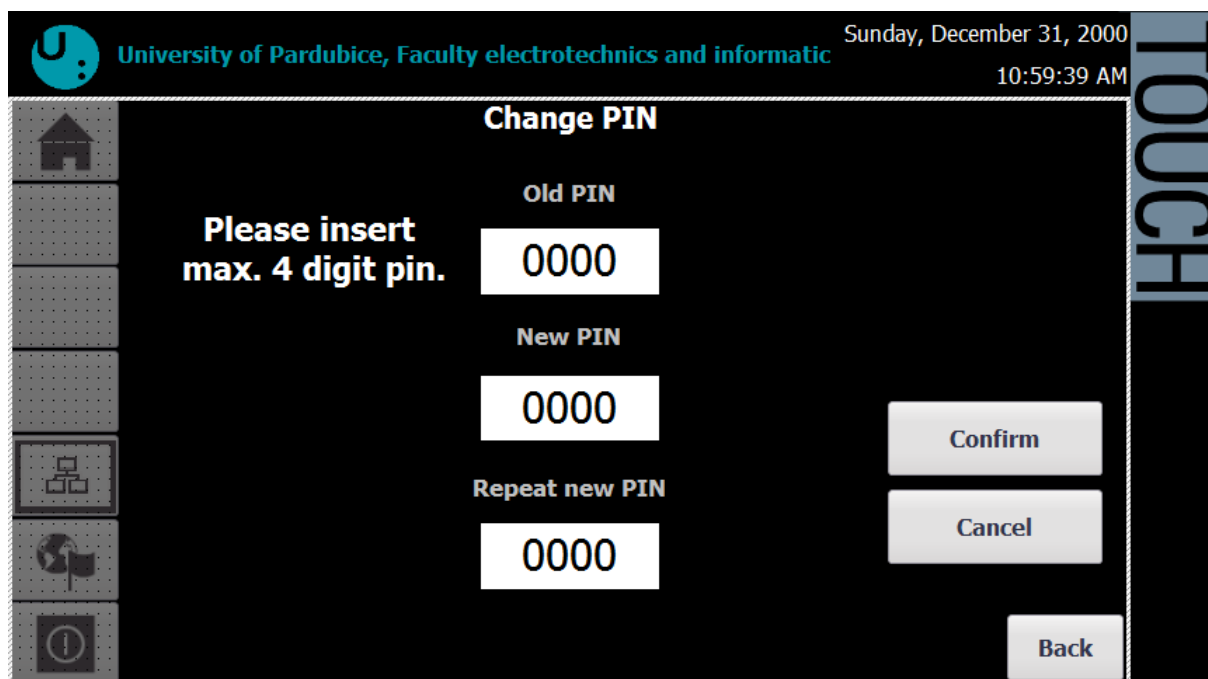
Kdy v záložce Animations, kategorii Display, programátor přidá novou animaci s názvem Appearance, kde je možnost zvolit příslušnou barvu podle hodnoty zvoleného tagu. Podobný princip je použit i v případě zobrazování příslušného hlášení o korektnosti pinu. Barva pozadí pole je definovaná stejně a zobrazovaný text je opět nastavován podle hodnoty tagu, která vybere předdefinovaný text z textového seznamu.



Obr. 5.4 – Vizualizace pro ovládání zámku

Poslední tlačítko s označením Cancel vynuluje hodnotu zadaného pinu a vypne oznámení o nesprávném pinu. Předposlední tlačítko Change PIN přepne na další ovládací obrazovku, která obsahuje prvky a funkce fungující analogicky jako v předchozím případě.

Rozdílem je přítomnost tří vstupních polí, do kterých se zadává starý pin, nový pin a opakovaný nový pin. Po stisknutí potvrzovací tlačítka se opět provede kontrola starého pinu a taky zda zadaný pin souhlasí s opakovaným pinem. I v tomto případě dostává uživatel zpětnou vazbu v podobě hlášení, které se objeví na pravé straně nad tlačítky, zda veškeré zadané atributy jsou správné nebo se vyskytla chyba v případě starého pinu či nesouhlasí nový pin z potvrzovacím.

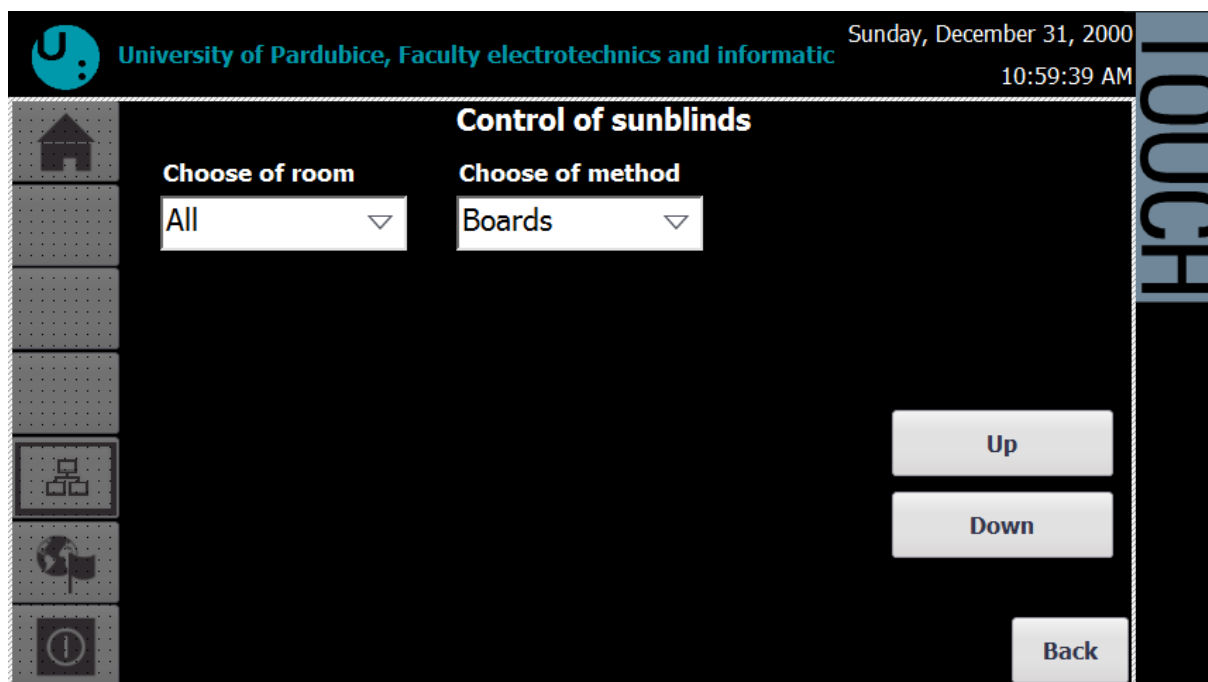


Obr. 5.5 – Vizualizace pro změnu pinu

#### 5.1.4 Obrazovka ovládání rolet

Princip ovládací obrazovky pro rolety je velice podobný s obrazovkou, která obsluhuje osvětlení místností. Opět se zde nachází dvě symbolická vstupně výstupní pole, první nabízí uživateli výběr místnosti, ve které chce manipulovat s roletami. Druhé pole obsahuje dvě metody pohybu rolet. Jestliže uživatel zvolí první variantu nesoucí označení Boards a následně pomocí dvou tlačítek, nacházejících se na pravé straně, zvolí pohyb nahoru či dolů, rolet začnou vykonávat pohyb, který potrvá do té doby, než roleta bude plně stažena či vytažena.

Při volbě druhé metody, která nese označení Free, je uživateli umožněno volit libovolnou polohu rolety. Držením tlačítka Up nebo Down provádí příslušný pohyb. Jestliže je dané tlačítko puštěno, roleta se zastaví přesně v tu samou chvíli. Na následujícím obrázku je prezentována vizualizace ovládací obrazovky pro rolety.



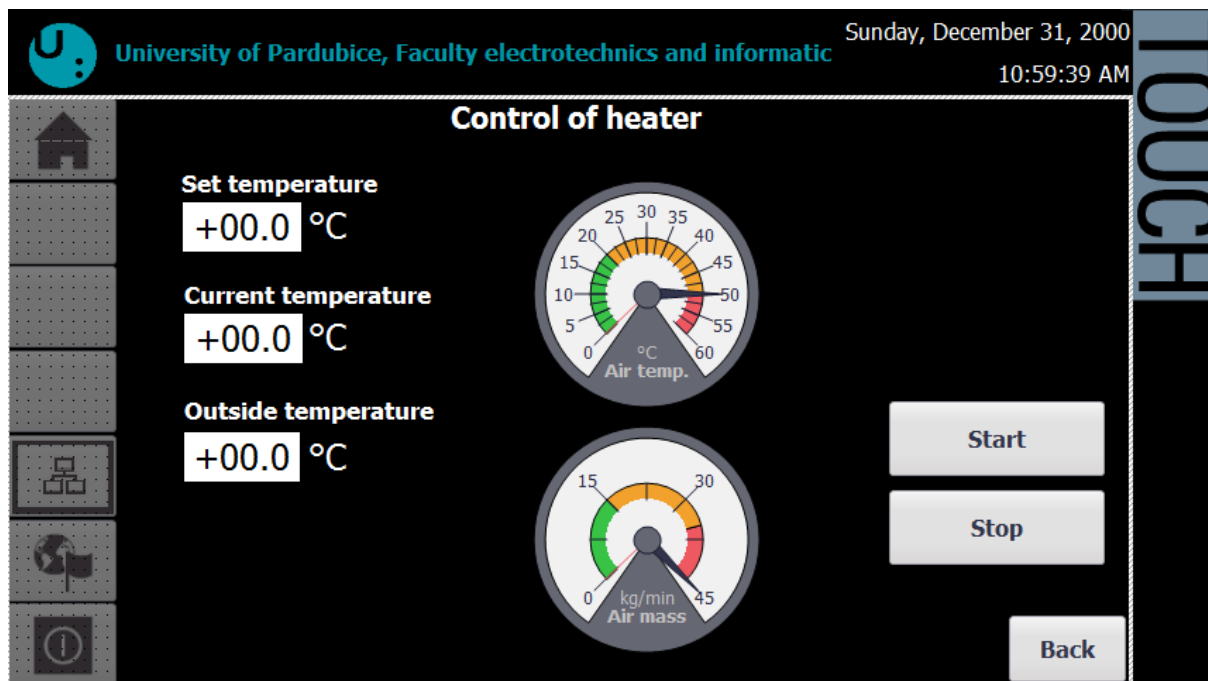
Obr. 5.6 – Vizualizace obrazovky pro ovládání rolet

### 5.1.5 Obrazovka ovládání topení

K nastavování požadované teploty slouží první vstupní pole, které se nachází v horní levé části obrazovky, označeno Set temperature. Zbylá dvě pole zobrazují aktuální teplotu v místnosti a venkovní teplotu. Tyto dvě hodnoty jsou získávány z modelu a přes OPC server posílány do PLC. V případě fyzického modelu by se tyto hodnoty získávali pomocí snímačů teploty.

Uprostřed obrazovky se nachází dva nové prvky, které nebyly použity v předchozích případech. Jedná se o dvě měřidla. Vrchní měřidlo zobrazuje údaj o teplotě vzduchu, který vychází z topení, druhé měřidlo informuje uživatele o množství vzduchu vycházejícího z topení. V případě horního měřidla je jeho stupnice definována podle minimální a maximální hodnoty teploty, kterou je schopno těleso produkovat.

Množství vzduchu, které projde topení může nabývat pouze tří hodnot, pokud se nepočítá nulová hodnota při vypnutém režimu. Tomu je přizpůsobená stupnice druhého měřidla. Pomocí dvojice tlačítek Start a Stop je topení spuštěno nebo vypnuto.



Obr. 5.7 – Vizualizace obrazovky pro ovládání topení

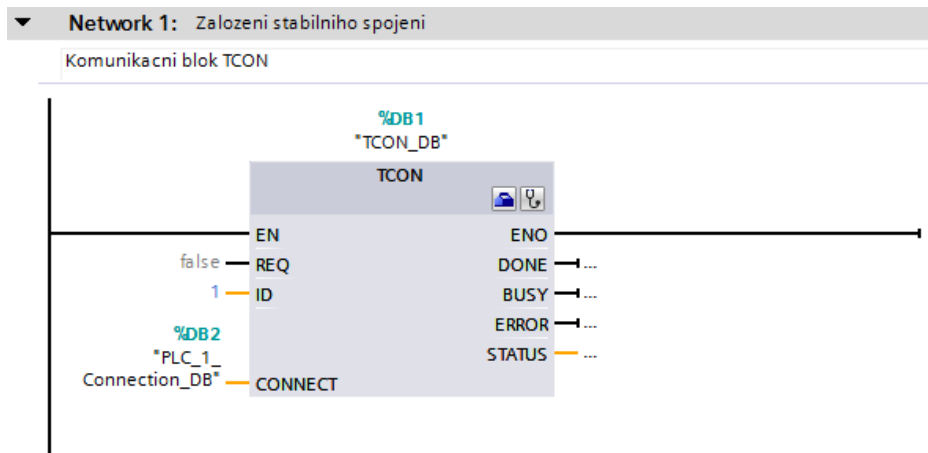
## 5.2 PROGRAMY PRO PLC

Stejně jako v případě HMI panelu je zapotřebí PLC prvně přidat do projektu, definovat jeho připojení a vlastnosti. Přidání zařízení do projektu je naprosto stejné jako v předchozím případě. K orientaci při výběru typu CPU u PLC může posloužit výrobní číslo výrobku (ang. Article no.). Důvod může být následovný, kdy v případě této práce je vybráno PLC, které obsahuje CPU 1515-2 PN, ovšem v nabídce jsou dvě varianty. Podle výrobního čísla automatu se zjistí, která varianta je správná.

Po přidání zařízení je zapotřebí nadefinovat IP adresy pro jednotlivé porty. Důležitou položkou nastavení je povolení možnosti komunikaci s OPC serverem. Toto nastavení se nachází v záložce General, kategorii Protection & Security, kde se permanentně povolí v Connection mechanism přístup vzdáleného komunikačního partnera, kterým může být například zvolený OPC server. Pokud by tato možnost nebyla povolena, PLC by ignorovalo veškeré požadavky, které by přicházeli od OPC serveru. Samotná komunikace mezi PLC a OPC je popsána v následující kapitole.

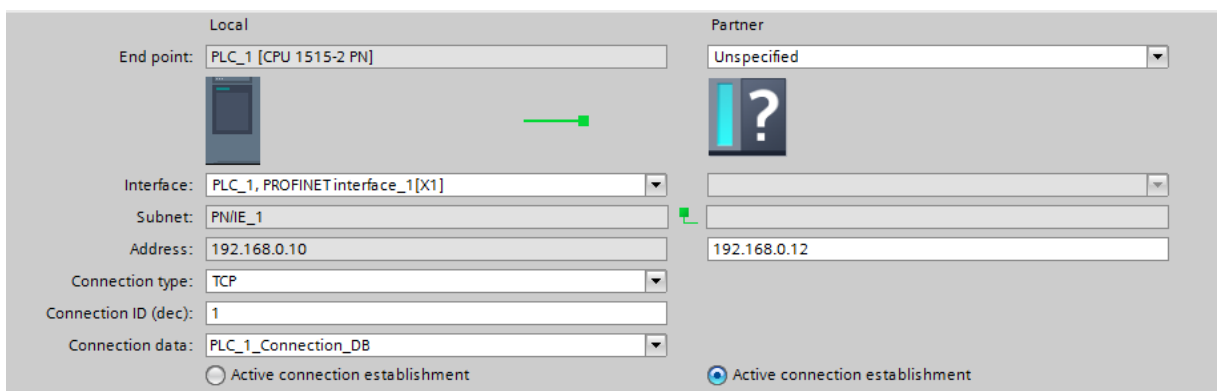
## 5.2.1 Komunikace

První program, který je zapotřebí v PLC zprovoznit, bude mít za úkol obsluhovat komunikaci s OPC serverem. Nový programový blok bude mít strukturu organizovaného bloku a zvolený programovací jazyk LAD. Do takhle vytvořeného bloku pak programátor vkládá jednotlivé funkce a logické operace. Funkční blok, který vytvoří stabilní komunikační spojení, nese název TCON a je zobrazen na obr. 5.8.



Obr. 5.8 – Komunikační blok TCON

Blok stačí přidat do sítě a následně nakonfigurovat. Společně s ním se vytvoří i potřebný datový blok. Po otevření záložky Configuration a vybrání kategorie Connection parameter má programátor k dispozici nastavení lokálního stroje a komunikačního partnera, který má být k němu připojen. Pro komunikace s OPC server je vybrán typ partnera Unspecified. Následuje volba komunikačního rozhraní, nastavení IP adresy partnera a volba typu připojení. Důležité je nastavit na straně partnera aktivní připojení (ang. Active connection establishment).



Obr. 5.9 – Konfigurace bloku TCON

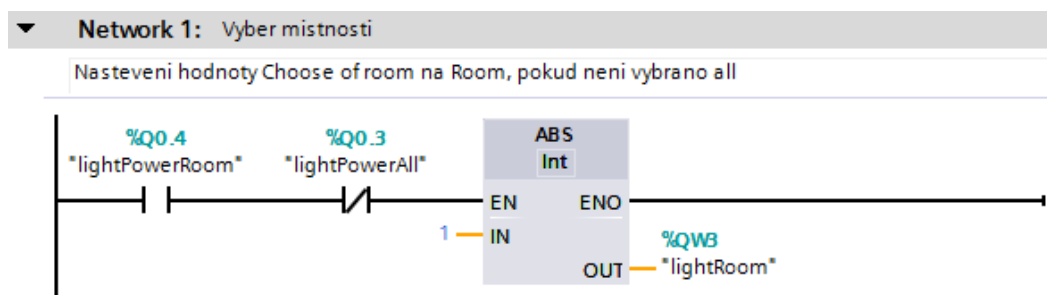
Po zkompilování a nahrání programu do PLC je vytvořené stabilní a funkční propojení mezi PLC a OPC serverem. K úspěšné výměně hodnot je zapotřebí ještě definovat použité tagy. Adresy se musí shodovat a pro lepší přehlednost je doporučeno, aby název tagu byl stejný jako tagu na OPC serveru. Na obr. 5.10 jsou zobrazeny všechny použité tagy v této práci. V následujících kapitolách jsou popsány jednotlivé programy na ovládání jednotlivých částí bytové jednotky.

TestTIA14_PLC ▶ PLC_1 [CPU 1515-2 PN] ▶ PLC tags ▶ Tag table_1 [23]									
Tag table_1									
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervision	Comment
1	lightIntensity	Int	%QW1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		nastaveni intezity
2	lightRoom	Int	%QW3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		vyber mistnosti pro svetla
3	lightColour	Int	%QW5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		vyber barvy
4	lightPowerRoom	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		switch pro pokoj
5	lightPowerHall	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		switch pro chodbu
6	lightPowerAll	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		switch pro celou bytovou jednotku
7	windowSetAllUp	Bool	%Q7.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set pohybu nahoru pro vsechny zaluzie
8	windowSetAllDown	Bool	%Q7.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set pohybu dolu pro vsechny zaluzie
9	windowSetRoomUp	Bool	%Q7.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set pohybu nahoru pro zaluzie v pokoji
10	windowSetRoomDown	Bool	%Q7.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set pohybu dolu pro zaluzie v pokoji
11	windowSetHallUp	Bool	%Q7.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set pohybu nahoru pro zaluzie v chodbe
12	windowSetHallDown	Bool	%Q7.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set pohybu dolu pro zaluzie v chodbe
13	windowResetHallUp	Bool	%I10.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set resetovani pohybu nahoru pro chodbu
14	windowResetHallDown	Bool	%I10.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set resetovani pohybu dolu pro chodbu
15	windowResetRoomUp	Bool	%I10.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set resetovani pohybu nahour pro pokoj
16	windowResetRoomDown	Bool	%I10.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set resetovani pohybu dolu pro pokoj
17	doorOpen	Bool	%Q14.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		signal pro otevreni dveri
18	doorClose	Bool	%Q14.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		signal pro zavreni dveri
19	heaterSet	Real	%ID20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		set zadane teploty
20	heaterAct	Real	%ID24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		hodnota aktualni teploty
21	heaterOut	Real	%ID28	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		hodnota venkovni teploty
22	heaterTemp	Real	%QD22	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		hodnota teploty z topeni
23	heaterAir	Real	%QD26	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		hodnota mnozství vzduchu z topeni

Obr. 5.10 – Seznam definovaných tagů v prostředí TIA Portal

### 5.2.2 Ovládání světel

Program pro ovládání světel, který obsahuje PLC je poměrně jednoduchý. Jelikož hodnota pro nastavení intenzity je přímo předávána z panelu do modelu. Podobně je na tom i předávání informace o zapnutí a vypnutí příslušných světel, volba barvy či místnosti, kdy tyto vlastnosti jsou nastavovány v třídě modelu. Program v PLC pouze nastavuje hodnotu pole pro výběr místnosti, kdy je přepnuto tlačítko pro jednu místnost a není sepnuto tlačítko pro všechny místnosti. Pokud tak nastane, hodnota v poli je nastavena na příslušnou místnost. Ukázka programu je na obr. 5.11

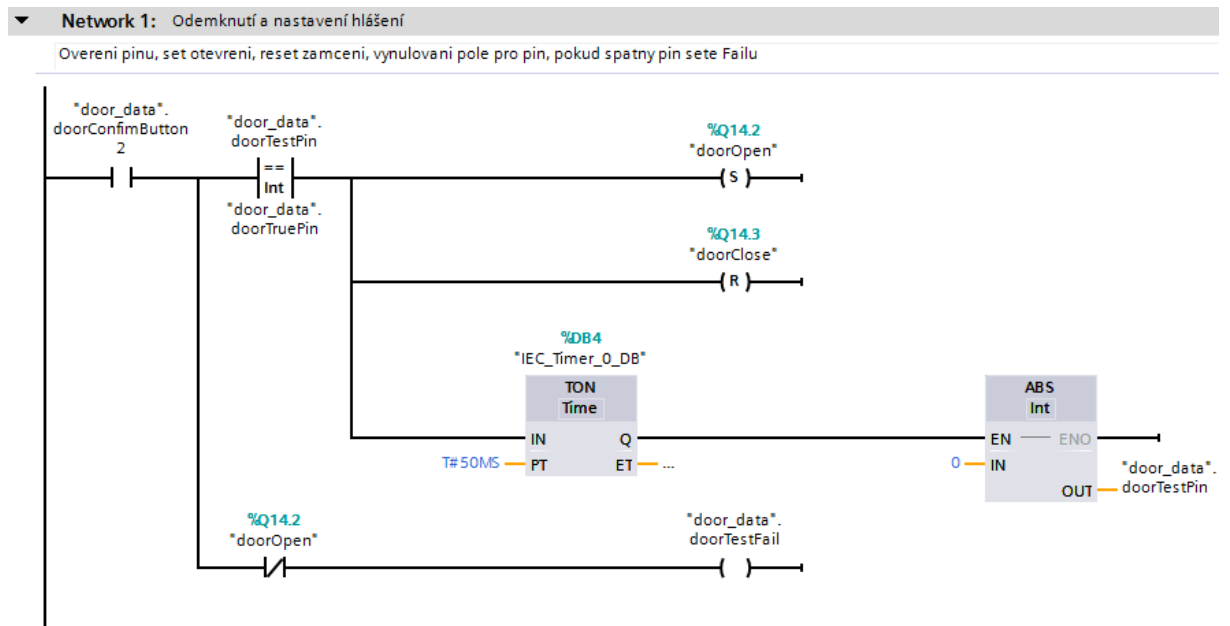


Obr. 5.11 – Program pro osvětlení

### 5.2.3 Ovládání zámku a změny pinu

V případě ovládání světel se jednalo opravdu o jednoduchý program, kde stačil pouze základní organizační blok, ve kterém byl vytvořen krátký program. Pro ovládání zámku a změnu pinu bylo zapotřebí použít i datový blok společně s organizačním. Datový blok obsahuje například hodnoty, které určují stav tlačítek použitých uživatelem, nebo hodnotu základního a později hodnoty testovacího, nového a potvrzovacího pinu. Pro uložení hodnot tlačítek je tento způsob úspornější, než používání jednotlivých tagů mezi PLC/HMI, které zbytečně zabírají adresový prostor.

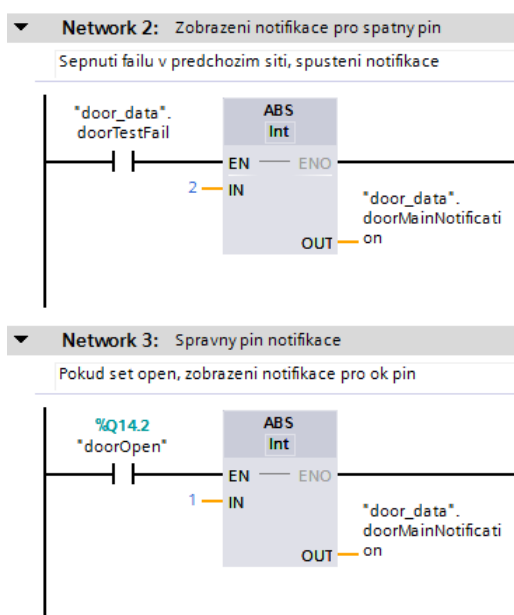
Spuštění první části programu proběhne při stlačení tlačítka Confirm, kdy se v následujícím kroku porovnají hodnoty uloženého pinu a testovacího pinu. Pokud se hodnoty shodují, dojde k nastavení logické jedničky pro výstupní tag *doorOpen*, který odemkne zámek. Zároveň je vyresetován zamykací tag *doorClose* a spuštěn generátor zpoždění TON. Tento generátor po uplynutí dané doby pošle spouštěcí signál do bloku ABS, který uloží absolutní hodnotu zadanou na vstupu do příslušného tagu definovaného na výstupu.



Obr. 5.12 – Program odemknutí zámku

Program ještě obsahuje variantu, kdy se hodnoty uloženého pinu a testovacího neshodují. Dojde k nastavení výstupu, který provede uložení odpovídající hodnoty do proměnné *doorMainNotification*, podle této hodnoty je vybráno oznámení v předefinovaném textovém seznamu. Analogicky je tento princip použit v případě, kdy jsou piny shodné a je povoleno odemknout zámek.

Podobná funkčnost je i v případě, kdy uživatel použije tlačítko Cancel. Aktivuje se program, jehož cílem je nastavit nulovou hodnotu do tagu *doorMainNotification*. Nulová hodnota značí v textovém seznamu prázdný řádek. Stejná hodnota je nastavena i pro tag *doorTestPin* Tímto krokem je zajištěno vynulování zadané hodnoty od uživatele a nahrazení hodnoty v poli na základní. Stejný princip, z hlediska zobrazení hlášení, je použit i při zmáčknutí tlačítka Lock, které vyresetuje signál pro otevření a nastaví signál pro zamknutí. Nyní přichází na řadu změna pinu.

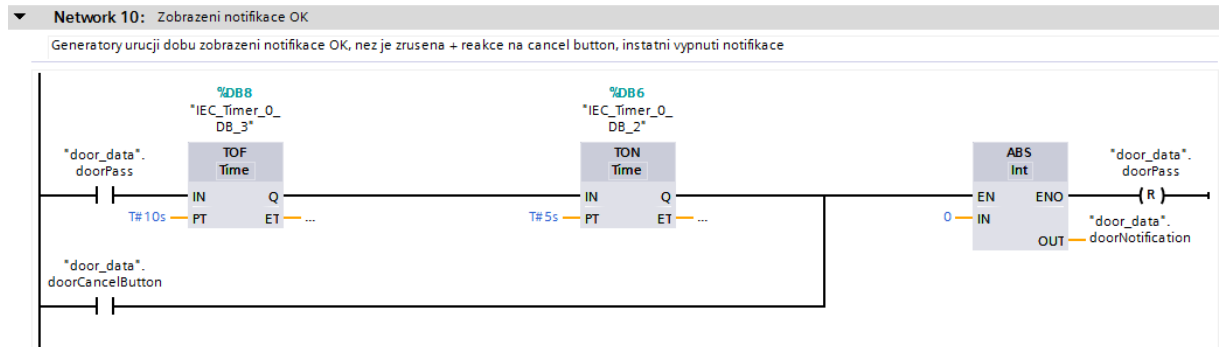


Obr. 5.13 – Nastavení hodnot pro výběr oznámení v případě špatného pinu nebo odemknutí

Struktura programu pro změnu pinu je velice podobná struktuře programu pro kontrolu pinu. Nedochozí zde pouze k porovnání zadaného pinu a pinu uloženého, ale i ke kontrole nového pinu a následně k jeho opětovnému správnému zadání. Proto v porovnání s předchozím programem přibyla ještě jedna větev.



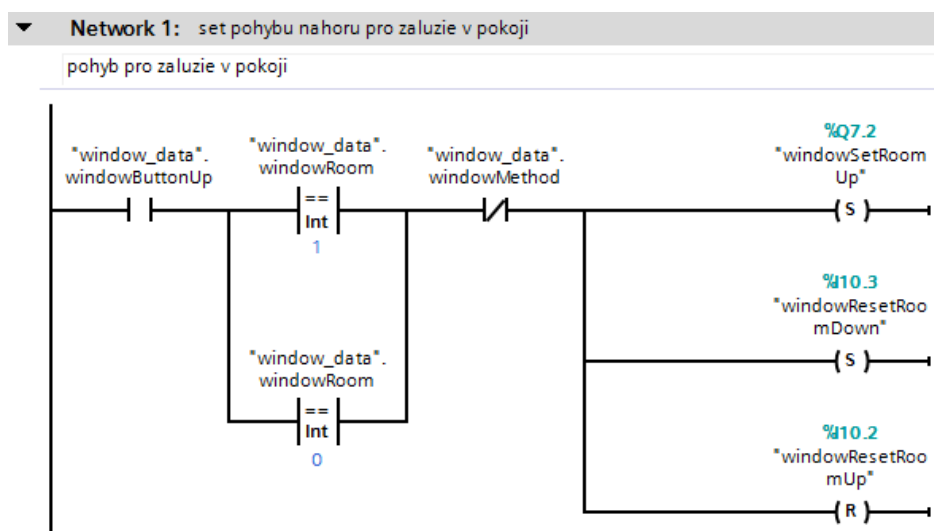
Ten je sepnut po uplynutí doby za podmínky přítomnosti signálu na vstupu. Poslední částí programu je opět vynulování hodnot jednotlivých zobrazovaných polí na panelu po úspěšné změně pinu.



Obr. 5.15 – Definování doby zobrazení hlášení o úspěšné změně pinu

## 5.2.4 Ovládání rolet

Rolety mohou být ovládány dvěma způsoby. Uživatel může zvolit na panelu pohyb v mezích, kdy může být roleta buď plně otevřena nebo zavřena. Druhým způsobem pohybu uživatel volí libovolnou polohu rolety. Programově jsou tyto dva způsoby velice podobné a rozdíl je jenom minimální. Po stlačení tlačítka pro pohyb nahoru nebo dolů, dojde ke kontrole výběru místnosti. Pokud odpovídá hodnota uložená v příslušné proměnné, hodnotě v porovnávacím bloku, je signál aktivní.

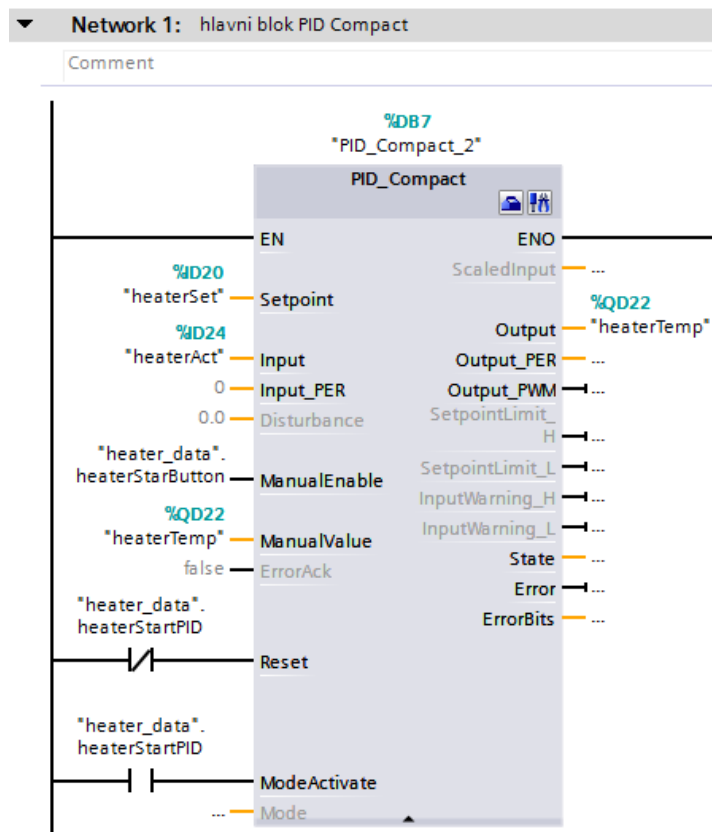


Obr. 5.16 – Program pro ovládání pohybu rolet

Rozepínací kontakt je pak sepnut v případě, že je zvolená metoda pro pohyb mezi mezemi, tedy proměnná *windowMethod* je rovna nule. Jestliže je tato podmínka splněna, nastaví se logická 1 tagu pro pohyb nahoru nebo dolů. Zároveň je nastavena logická 0 pro resetovací signál stejného pohybu a logická 1 pro resetovací signál opačného pohybu. Jedná se tedy o šablonu programu, kde jsou pouze měněny hodnoty pro porovnání výběru místnosti a výstupů rolet. Ukázka části programu pro pohyb v mezích v pokoji je prezentován na obr. 5.16.

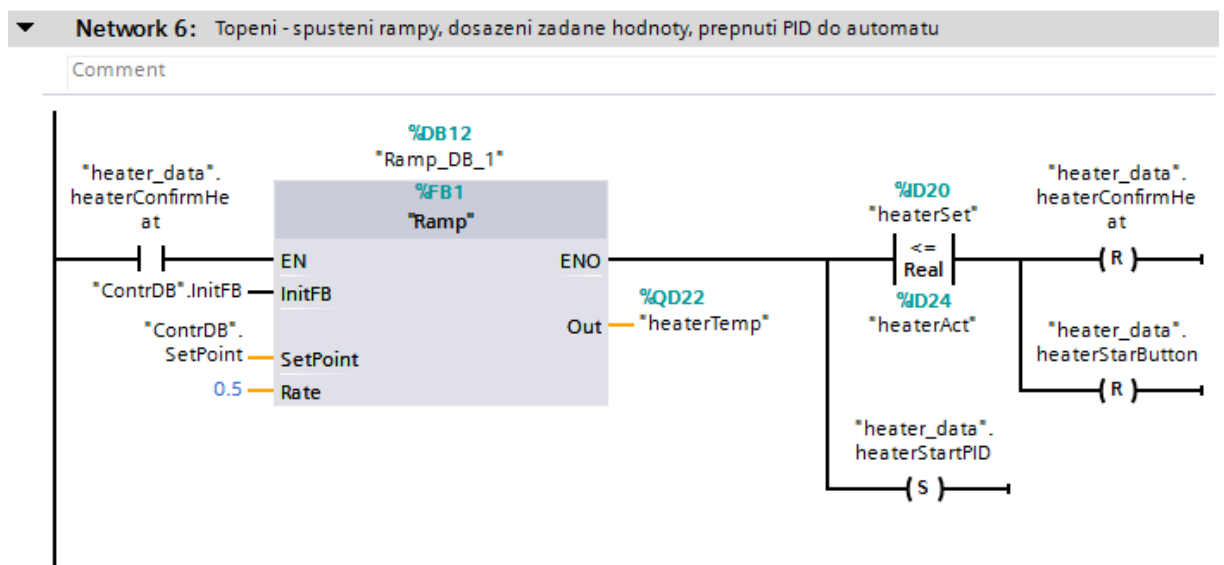
### 5.2.5 Ovládání topení

V případě topení jsou ovládány jeho dvě vlastnosti. Jedná se o teplotu vzduchu vycházející z topení a jeho množství za jednotku času. Pro ovládání teploty jsou přítomny dva prvky. Prvním z nich je funkční blok, který vytváří signál v podobě rampy. Za ním následuje klasický blok pro PID regulátor. Princip činnosti je následovný. Uživatel zadá požadovanou teplotu a stiskne tlačítko Start. Dojde k porovnání hodnot zadané teploty a venkovní teploty. Když nastane případ, kdy venkovní teplota je nižší než zadaná hodnota, spouští se program pro topení. Ten obsahuje dvě fáze.



Obr. 5.17 – Blok PID regulátoru

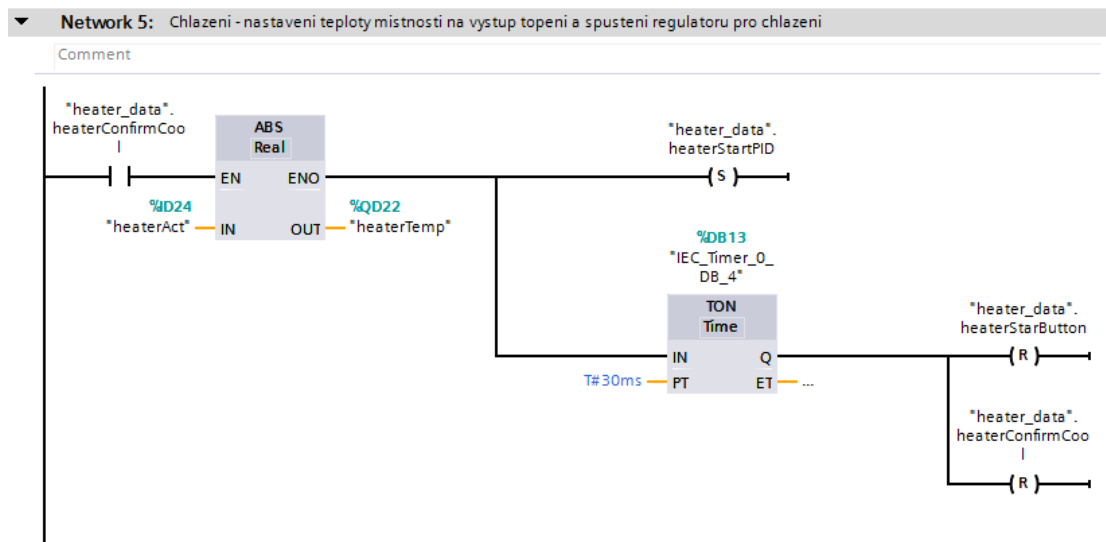
V první fázi je spuštěn funkční blok rampy, prezentován na obr. 5.18., který má inicializovanou počáteční hodnotu dle aktuální teploty v místnosti a konečnou hodnotu je nastavena na maximální teplotu, kterou je schopno topení dodávat. Zároveň je spuštěn regulátor do manuálního režimu. Dojde-li k vyrovnání aktuální teploty na hodnotu požadované teploty, spouštěcí signály pro manuální režim regulátoru a startovací režim topení jsou resetovány. Tím dojde k přepnutí regulátoru do automatického režimu a začne plynulá regulace případné regulační odchylky nebo změny žádané veličiny. Díky akčnímu zásahu ve tvaru rampy, který byl generován na začátku, až do doby dosažení požadované teploty, nemusí sám regulátor generovat velký akční zásah na začátku regulace.



Obr. 5.18 – Funkční blok pro rampu

Pokud při porovnání venkovní teploty a žádané teploty, které je prováděno na začátku programu, dojde k opačnému výsledku, tedy venkovní teplota má větší hodnotu než žádaná hodnota teploty v místnosti, následuje spuštění programu pro chlazení. Počáteční kroky jsou analogické, jako v předchozím případě. Spuštění regulátoru a jeho přepnutí do manuálního režimu. Na jeho výstup je nastavena aktuální hodnota teploty v místnosti. Důvodem tohoto kroku je, že regulátor má nastaven interval teploty, se kterou může pracovat. Pokud by nedošlo k počátečnímu inicializaci výstupní hodnoty, na hodnotu aktuální teploty v místnosti, regulátor by začal regulovat od spodní hranice intervalu teploty, tedy v tomto případě od 12 °C.

Došlo by ke zkrselení dat a nesprávnému průběhu regulace, jelikož topení je umístěno v místnosti. Tedy regulátor musí začít pracovat s počáteční teplotou, která odpovídá té v místnosti. Při nastavení hodnoty je následně spuštěn generátor zpožděného spuštění, který po 30 ms přivede spouštěcí signál na resetovací výstupy manuálního režimu regulátoru. To má za následek přepnutí regulátoru do automatického režimu a začátek regulace. Programové řešení spouštěcí části prezentuje obr. 5.19.

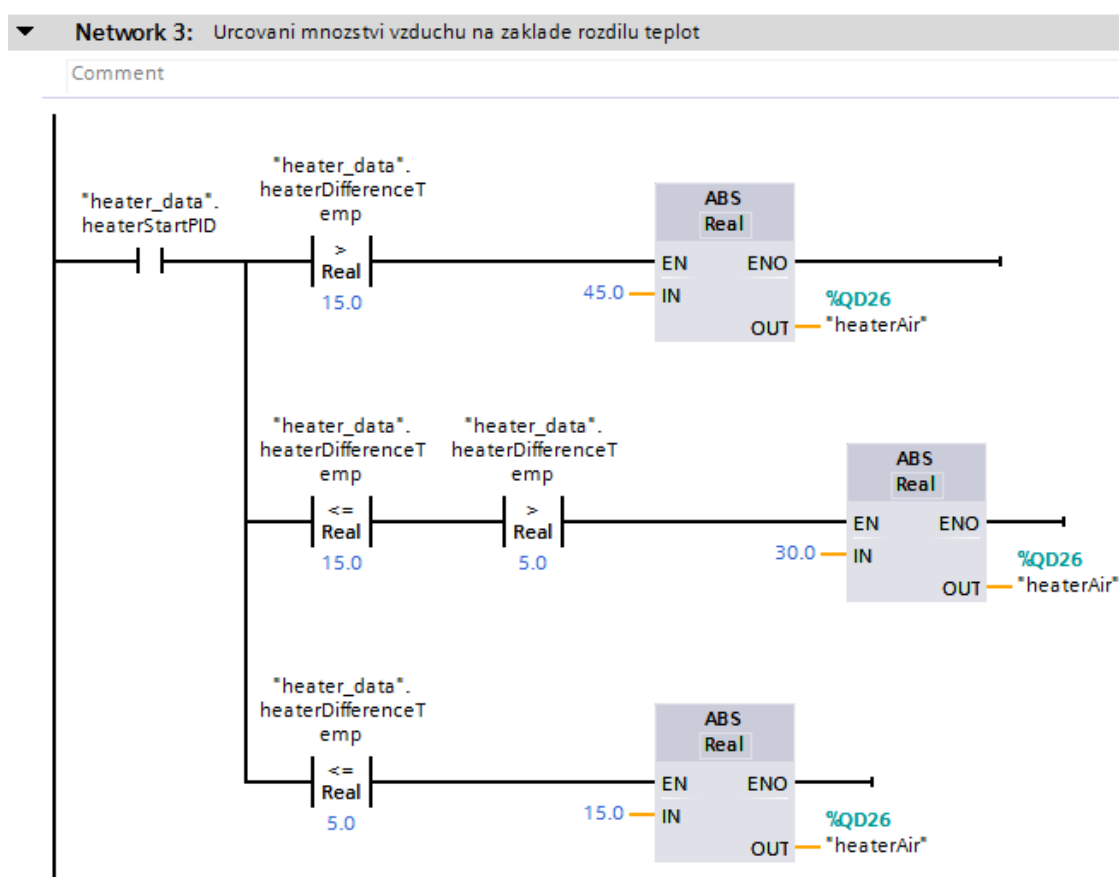


Obr. 5.19 – Spuštění programu chlazení

Poslední částí ovládání obsahuje určení množství vzduchu, které má vycházet z topení. Tato hodnota je nastavována v závislosti na rozdílu žádané a venkovní teploty. Jestliže je tento rozdíl menší nebo roven 5 °C, množství vzduchu pak odpovídá hodnotě 15 kg/min. Analogický postup je pak použit i v následujících dvou případech.

Rozdíl teplot je větší jak 5 °C a menší nebo roven 15 °C, množství vzduchu nastaveno na 30 kg/min. Poslední varianta obsluhuje případ pro rozdíl větší jak 15 °C, kdy dochází k nastavení množství na maximální hodnotu 45 kg/min. Provedení v TIA Portal je ukázáno na obr. 5.20.

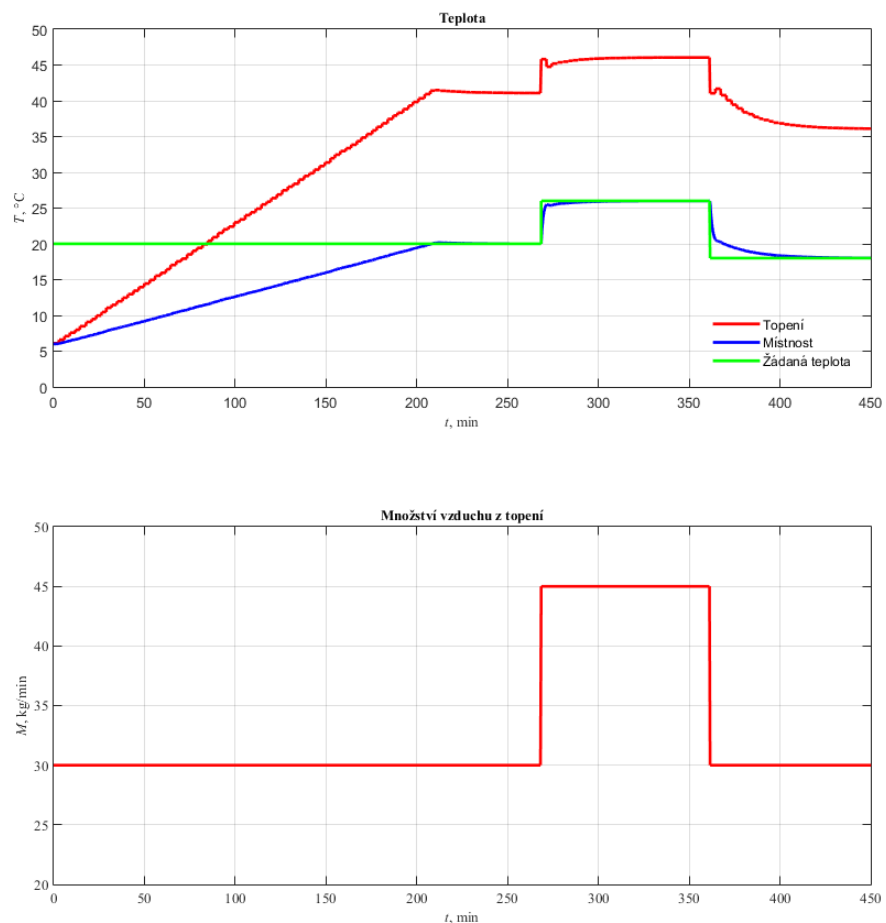
Důvodem volby této možnosti přepínání množství vzduchu bylo udržení stabilní hodnoty jedné z veličin, kterou topení ovlivňuje teplotu v místnosti. Pokud by byla plynule měněna hodnota množství vzduchu a zároveň by byla regulována teplota vzduchu topení, mohlo by dojít k zhoršení regulačního pochodu. Proto byly vybrány určité meze, definované rozdílem venkovní a žádané teploty, ve kterých je množství vzduchu pevně definováno.



Obr. 5.20 – Řízení množství vzduchu topení

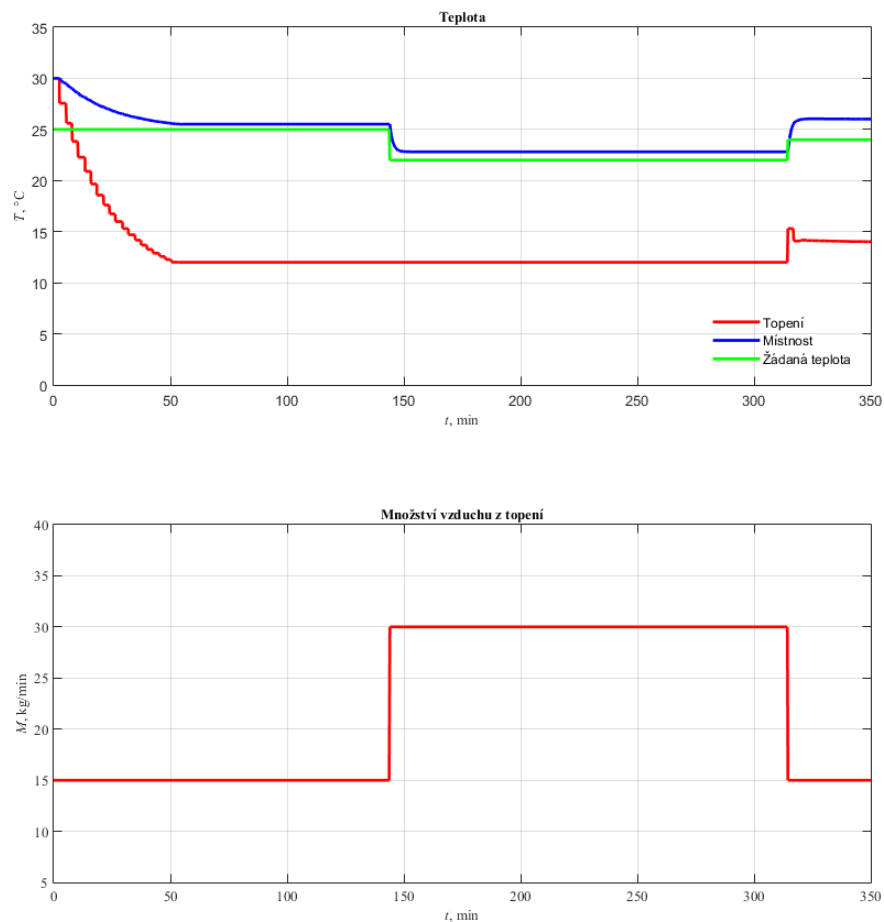
## 6 SIMULAČNÍ SCÉNÁŘ

Pro simulační scénář byla vybrána pouze jedna vlastnost bytové jednotky, a to řízení teploty. Zbylé vlastnosti nemohly nabídnout žádná data ke zpracování, či k porovnání, jelikož jejich nastavení závisí čistě na uživateli. V této kapitole budou simulované tři scénáře, kdy v každém bude jiná venkovní teplota a uživatel provede třikrát nastavení žádané teploty pro každý scénář. Venkovní teploty byly vybrány, tak, aby reprezentovaly přibližně teploty, které mohou nastat v každém ročním období. Teplota pro jaro a podzim je brána jako shodná, to je důvod, proč jsou pouze jenom tři scénáře. První byla naměřena data pro jaro, jejichž průběh je zobrazen na obr. 6.1.



Obr. 6.1 – Regulační průběh teploty pro venkovní teplotu 6 °C

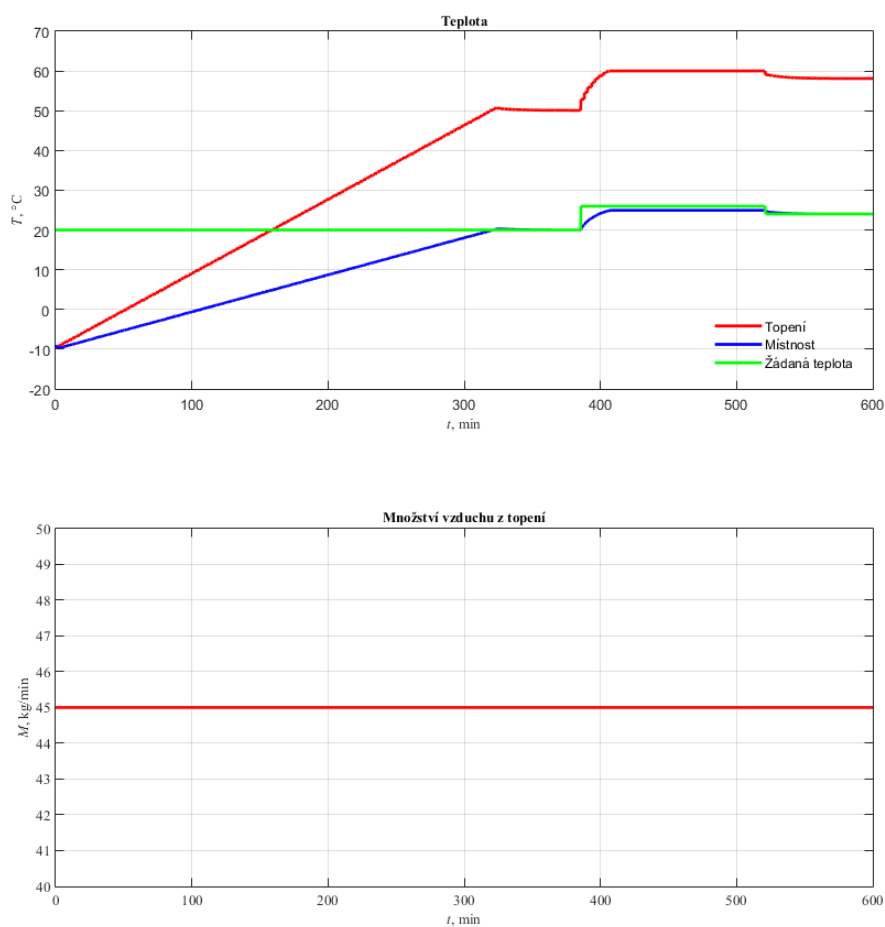
V tomto případě byla zvolena venkovní teplota 6 °C, stejná hodnota byla přidělena i místnosti. Uživatel nastavil první žádanou teplotu na hodnotu 20 °C a spustil topení. Rozdíl venkovní a aktuální vnitřní teploty je 14 °C, to má za následky výběr střední hodnoty množství vzduchu, která je 30 kg/min. Na začátku regulačního pochodu lze vidět funkci rampy, která postupně zvyšuje teplotu topení, než je dosaženo žádané hodnoty. Pak spíná regulátor a kompenzuje mírné překročení žádané hodnoty, které bylo způsobeno rampou.



Obr. 6.2 – Regulační pochod pro venkovní teplotu 30 °C

Po ustálení regulačního pochodu uživatel zadal změnu žádaná teploty na 26 °C. V této situaci už topení obsluhuje pouze regulátor a rampa je vypnutá. Poslední volba žádané teploty je rovna hodnotě 18 °C. Při první změna na vyšší teplotu a následně na nižší lze pozorovat menší špičku v případě akční veličiny teploty. Ta je způsobena tím, že při změně teploty dochází i ke změně množství vzduchu, kvůli rozdílu žádané a venkovní teploty. Regulátor pak vygeneruje zaznamenaný zásah.

Následuje změna venkovní teploty, která bude reprezentovat letní období, a tedy potřebu chladit. Venkovní teplota byla zvolena na hodnotu 30 °C, stejnou hodnotu má i vzduch, který se nachází v místnosti. Byla zadána žádaná hodnota 25 °C a spuštěna regulace. Zde není spuštěná rampa, ale okamžitě sepnut regulátor, který začne snižovat teplotu topného tělesa. Po ustálení regulačního pochodu, byla provedena změna žádané hodnoty na 22 °C. Tato změna zvýší rozdíl mezi venkovní a žádanou teplotou. Dojde ke zvýšení přívodu množství vzduchu. Poslední změnou teplotou bylo testováno chování při zvýšení žádané teploty na 24 °C.



Obr. 6.3 – Regulační pochod pro venkovní teplotu -10 °C

Z průběhu, který je zobrazen na obr. 6.2 lze zjistit, že nebylo ani v jednom ze tří případů volby teploty dosaženo žádané hodnoty. Nejbližší přiblížení bylo v případě první volby. Tato odchylka je způsobena omezením minimální teploty, kterou je schopno topení dodávat, na 12 °C. Ke zlepšení výsledků by mohlo vést snížení této hranice nebo zvětšení množství přiváděného vzduchu

Finální varianta simuluje teplotu venkovního vzduchu v zimních měsících, kdy byla zvolena hodnota  $-10\text{ }^{\circ}\text{C}$ , stejná hodnota platí i pro vnitřní teplotu bytové jednotky. Po spuštění topení je průběh podobný, jako v případě prvního scénáře. Posloupnost žádaných teplot je následující  $20\text{ }^{\circ}\text{C}$ ,  $26\text{ }^{\circ}\text{C}$  a pak snížení na  $24\text{ }^{\circ}\text{C}$ . U volby druhé teploty se objevuje stejný problém, který nastával i v případě letních měsíců, kdy nebylo topení schopno dosáhnout požadované teploty v místnosti. Pro zadanou teplotu  $26\text{ }^{\circ}\text{C}$  topení ohřála vnitřní vzduch na teplotu  $25\text{ }^{\circ}\text{C}$ . Rozdíl  $1\text{ }^{\circ}\text{C}$  není tolik pocitově znát, nicméně v případě, kdyby venkovní teplota klesla pod simulovanou hodnotu, rozdíl by postupně narůstal. Proto poslední zvolená teplota byla  $24\text{ }^{\circ}\text{C}$ , této hodnoty bylo topení schopno dosáhnout. Řešení tohoto problému je stejné, jako v případě simulace letních měsíců.

Na posledním obr. 6.4 jsou zaznamenány parametry PID regulátoru, které byly použity pro tuto simulaci. Jejich získání bylo provedeno pomocí Autotuning, kterým program TIA Portal disponuje. Testovány byly i vlastní parametry, které získal autor. Ty ovšem nezaručovaly tak hladký průběh, jako je tomu v případě parametrů získaných pomocí Autotuning.

Parameter	Value	Unit
Proportional gain	3.806153E-1	
Integral action time	4.649515E-1	s
Derivative action time	9.15702E-2	s
Derivative delay coefficient	0.1	
Proportional action weighting	1.0	
Derivative action weighting	0.0	
Sampling time of PID algorithm	0.099999	s

**Tuning rule**

Parameter	Value
Controller structure	PID

Obr. 6.4 – Parametry PID regulátoru

## 7 ZÁVĚR

Cílem této práce bylo vytvoření modelu bytové jednotky ve vývojovém prostředí MATLAB, která bude obsahovat několik veličin, jež bude možno ovládat uživateli této jednotky. Bylo zvoleno osvětlení, to nabízí možnosti změny barvy v osvětlení v intervalu čtyř barev, následně i intenzitu osvětlení a výběr místnosti, která by měla být osvětlena. Druhá volba veličiny padla na odemykání a zamykání vchodových dveří pomocí čtyřmístného pinu. Jelikož byly ovládány dveře, další logickou volbou byly rolety na oknech a závěrem se přidalo i ovládání teploty v místnosti.

Všechny tyto vlastnosti jsou řízeny pomocí PLC a ovládacího panelu HMI, který vlastní pro každou veličinu jednotlivou obrazovku. Ta umožňuje obyvatelům jednotky volbu hodnot vybraných vlastností. Použité PLC a HMI bylo od firmy Siemens a jednalo se modely S7-1500 CPU1515-2 PN a TP700 Comfort. Vývojové prostředí bylo též použito od společnosti Siemens, a to TIA Portal V14. Při zprovoznění komunikace či samotného chodu velmi dobře posloužil obsáhlý Help programu TIA Portal, následně i support na stránkách Siemens.

Ke komunikaci mezi PLC a modelem bytové jednotky slouží OPC server. Konkrétně byl použit pro tento server software od firmy Kepware Technologies, která používá tuto technologii pod označením KEPServerEX V5 Tato společnost nabízí momentálně i novější verzi klienta, nicméně verze s označením V6 mi způsobovala problémy, které se týkaly mého hardware, tedy notebooku. Odlišnost novější verze a verze použité v diplomové práci je naprosto minimální. Jednalo se především o grafickou modernizaci prostředí. Ke zprovoznění komunikace byl opět použit především help příslušných programů a znalosti, které jsem měl z bakalářské práce.

Programy pro ovládání jsou řešeny pomocí LAD jazyka a funkčních bloků, se kterými je schopno dané PLC pracovat. Pohyb v prostředí TIA Portal byl poměrně intuitivní a přehledný, díky rozdělení do jednotlivých větví. Velmi zajímavé pak bylo řešení ovládání teploty vzduchu v místnosti, kde se nastavovala hodnota jak teploty vzduchu, tak jeho množství, které vycházelo z topení. Pro teplotu byl použit klasický PID regulátor, jeho parametry byly nastaveny pomocí Autotuning, který zajistil nejlepší možné výsledky. A množství vzduchu bylo vybíráno pomocí jednoduchého rozhodovacího programu, který stavěl na rozdílu mezi venkovní teplotou a požadovanou teplotou.

## LITERATURA

- KEPWARE INC.. 2016. *KEPServerEX: KEPServerEX V5*. [online]. [cit. 25. 04. 2017]  
Dostupné z: <https://www.kepware.com/en-us/products/kepserverex/documents/kepserverex-v5-manual.pdf>
- MATHWORKS INC.. ©2017. *Model and Dynamic Systém*. [online]. [cit. 04. 05. 2017]  
Dostupné z: <https://www.mathworks.com/help/simulink/gs/define-system.html>
- OPC FOUNDATION, ©2017. What is OPC?. In: *Opcfoundation.org* [online].  
[cit. 24. 04. 2017]. Dostupné z: <https://opcfoundation.org/about/what-is-opc/>
- SIEMENS AG. 12/2014. *Simatic S7-1500 CPU 1515-2 PN: Manual*. [online]. Nürnberg. [cit. 01. 05. 2017]. Dostupné z:  
[https://cache.industry.siemens.com/dl/files/167/81162167/att\\_92038/v1/s71500\\_cpu\\_1515\\_2\\_pn\\_manual\\_en-US\\_en-US.pdf](https://cache.industry.siemens.com/dl/files/167/81162167/att_92038/v1/s71500_cpu_1515_2_pn_manual_en-US_en-US.pdf)
- SIEMENS AG. 04/2012. *Simatic HMI: HMI devices Comfort Panels*. [online]. Nürnberg. [cit. 01. 05. 2017]. Dostupné z:  
[https://cache.industry.siemens.com/dl/files/233/49313233/att\\_59649/v1/hmi\\_comfort\\_panels\\_operating\\_instructions\\_en-US\\_en-US.pdf](https://cache.industry.siemens.com/dl/files/233/49313233/att_59649/v1/hmi_comfort_panels_operating_instructions_en-US_en-US.pdf)
- ZEZULKA, F. 2004. *Prostředky průmyslové automatizace*. Brno: VUTIUM. 176 s. ISBN 80-214-2610-1.

# **PŘÍLOHY**

**A – CD**

**Příloha k diplomové práci**

Řízení některých veličin bytové jednotky pomocí PLC

Tomáš Bizek

**CD**

## **Obsah**

- 1 Text diplomové práce v PDF
- 2 Programy pro PLC a HMI
- 3 Zdrojový kód modelu a komunikace