

# **Podpora výuky předmětu Mikropočítače a PLC zaměřené na bezpečnost s využitím PLC Siemens**

Petr Bardovcy

---

Bakalářská práce  
2019



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2018/2019

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

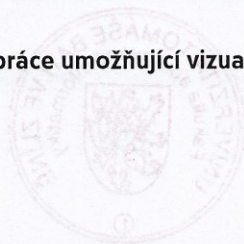
Jméno a příjmení: **Petr Bardovcy**  
Osobní číslo: **A16192**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Bezpečnostní technologie, systémy a management**  
Forma studia: **kombinovaná**

Téma práce: **Podpora výuky předmětu Mikropočítače a PLC zaměřené na bezpečnost s využitím PLC Siemens**

Téma anglicky: **Support of the Microcomputer and PLC Subjects Focused on Safety with the Use of Siemens PLC**

Zásady pro vypracování:

1. Vypracujte literární rešerši na téma "Programovatelné automaty v prostředí průmyslu komerční bezpečnosti."
2. Popište programovatelný automat Siemens Simatic S7-1200 a vývojové prostředí TiaPortál z uživatelského hlediska se zaměřením na studenty.
3. Vypracujte sadu úloh, které demonstrují možnosti a způsoby programování PLC Simatic S7-1200.
4. Navrhněte a realizujte sadu úloh zaměřených na úlohy z oblasti komerční bezpečnosti.
5. Vytvořte ukázkové práce umožňující vizualizaci pomocí HMI panelu.



Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace, 1.díl. 1. Vyd. Praha: BEN – technická literatura, 1999. ISBN 80-860-5658-9.
2. ŠMEJKAL, Ladislav. PLC a automatizace, 2.díl. 1. Vyd. Praha: BEN – technická literatura, 2005. ISBN 80-7300-087-3.
3. MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.
4. SIMATIC Programming with STEP 7 V5.5: Programming manual. Industry Online Support International [online]. Norimberk: Siemens AG, 2010 [cit. 2016-01-29]. Dostupné z:  
<https://support.industry.siemens.com/cs/document/45531107/simatic-programming-with-step-7-v5-5?dti=0&pnid=14341&lc=en-WW>
5. SIMATIC NET S7-300/400 – Industrial Ethernet / PROFINET Configuring and commissioning S7 CPs for Industrial Ethernet [online]. Norimberk: Siemens AG, 2013 [cit. 2016-01-29]. Dostupné z:  
[https://support.industry.siemens.com/cs/attachments/66993266/MN\\_s7-cps-ie\\_76\\_en-US.pdf](https://support.industry.siemens.com/cs/attachments/66993266/MN_s7-cps-ie_76_en-US.pdf)
6. BERGER, Von Hans. Automatisieren mit STEP 7 in AWL: Speicherprogrammierbare Steuerungen SIMATIC S7-300/400. 2., wesentlich überarb. und erw. Aufl. Erlangen: Publicis-MCD-Verl, 1998. ISBN 38-957-8089-8.

Vedoucí bakalářské práce:

**Ing. Tomáš Sysala, Ph.D.**

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

**14. prosince 2018**

Termín odevzdání bakalářské práce:

**15. května 2019**

Ve Zlíně dne 14. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



Ing. Jan Valouch, Ph.D.  
*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Petr Bardovec, v.r.  
podpis diplomanta

## **ABSTRAKT**

Cílem bakalářské práce je seznámit studenty předmětu Mikropočítače a PLC s programovatelnými automaty PLC od společnosti Siemens. V úvodní části práce je zmíněno, co to vlastně PLC je, k čemu je dobré a jak funguje. Následně je popsána možnost a problematika propojení se zabezpečovacími systémy. V další části je práce zaměřená na popis konkrétního PLC SIMATIC S7-1200 a vývojové prostředí TIA Portal, ve kterém je prováděna konfigurace a programování programovatelných automatů od společnosti Siemens, tak aby student dokázal učinit první kroky pro práci s PLC S7-1200. Praktická část je zaměřena na vytvoření vzorových úloh pro programování PLC a HMI panelů v TIA Portalu a jejich řešení.

Klíčová slova: PLC, TIA Portal, SIMATIC S7-1200, HMI, HMI KTP700 Basic, komerční bezpečnost

## **ABSTRACT**

The focus of my the thesis is to familiarize students of Microcomputers and PLC with the programmable automatic PLC from Siemens. The first part explains what PLC actually is, how does it work and what benefits it has. The following chapter deals with the possibility of connecting PLC with security systems. The next part focuses on describing the SIMATIC S7-1200 PLC and the development environment TIA Portal, where the configuration and programming of the programmable automatic PLCs from Siemens is realized, so any student could make their first steps towards effective work with PLC S7-1200. The practical part of the thesis is focused on creating template tasks for PLC and HMI panels programming in TIA Portal and their solution.

Keywords: PLC, TIA Portal, SIMATIC S7-1200, HMI, HMI KTP700 Basic, comercial security

V první řadě bych chtěl velice poděkovat Ing. Tomáši Sysalovi, Ph.D. za umožnění vypracovávat bakalářskou práci na dané téma pod jeho vedením. Moc mu děkuji také za veškeré poskytnuté rady při vytváření práce a za výbornou komunikaci. Dále bych chtěl poděkovat především své přítelkyni a rodině za velkou trpělivost a podporu, kterou jsem při vyváření práce potřeboval. Posledním komu bych chtěl poděkovat, tak je mému zaměstnavateli za jeho vstřícnost při pracovním vytížení a přátelům, kteří mě dokázali vždy podpořit.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD.....</b>	<b>8</b>
<b>I TEORETICKÁ ČÁST.....</b>	<b>9</b>
<b>1 OBECNĚ O PLC.....</b>	<b>10</b>
1.1 HISTORIE.....	10
1.2 CHARAKTERISTIKA PLC.....	11
1.3 STRUKTURA PLC.....	12
1.3.1 CPU.....	12
1.3.2 Digitální vstupní a výstupní jednotky.....	13
1.3.3 Analogové vstupní a výstupní jednotky.....	13
1.3.4 Komunikační jednotky.....	13
1.3.5 Ostatní jednotky.....	14
1.4 DĚLENÍ PLC.....	14
1.4.1 Kompaktní PLC.....	14
1.4.2 Modulární PLC.....	15
1.4.3 Mikro PLC.....	16
1.5 ZPRACOVÁNÍ PROGRAMU V PLC.....	17
1.6 PROGRAMOVACÍ JAZYKY PLC.....	18
1.6.1 Jazyk mnemokódů.....	19
1.6.2 Jazyky kontaktních schémat.....	19
1.6.3 Jazyk logických schémat.....	20
1.6.4 Jazyk strukturovaného textu.....	21
1.7 VYUŽITÍ PLC SE ZABEZPEČOVACÍMI SYSTÉMY.....	21
1.7.1 Systém KNX.....	22
1.7.2 Připojení KNX zařízení k PLC SIMATIC.....	22
1.7.3 Problematika připojení zabezpečovacích systémů.....	23
<b>2 PLC SIEMENS.....</b>	<b>24</b>
2.1 SIMATIC S7 - 1200.....	24
2.1.1 Základní popis.....	24
2.1.2 Pracovní módy S7 - 1200.....	25
2.1.3 Popis SIMATIC S7 - 1214C DC/DC/DC.....	26
2.1.4 WEB server.....	27
2.2 TIA PORTAL.....	27
2.2.1 Step 7 Basic.....	27
2.2.2 Programová struktura.....	28
2.2.2.1 OB bloky.....	29
2.2.2.2 FC bloky.....	29
2.2.2.3 FB bloky.....	29
2.2.2.4 DB bloky.....	30
2.2.3 Datové typy.....	30
2.2.4 WinCC Basic.....	32
2.3 VYTVOŘENÍ PROJEKTU V TIA PORTÁLU.....	32
2.3.1 Vytvoření projektu.....	35

2.3.2	Konfigurace HW .....	35
2.3.3	Vložení programového bloku.....	37
2.3.4	Kompilace projektu .....	40
2.3.5	Nahrání projektu do PLC S7 - 1200.....	41
2.3.6	Online testování projektu .....	43
<b>II</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>45</b>
<b>3</b>	<b>PŘÍKLADY PRO PROCVIČENÍ PROGRAMOVÁNÍ PLC SIMATIC S7 - 1200.....</b>	<b>46</b>
3.1	LOGICKÉ OPERACE V RŮZNÝCH PROGRAMOVACÍCH JAZYCÍCH .....	46
3.1.1	Zadání.....	46
3.1.2	Řešení.....	47
3.1.3	Komentář k řešení .....	48
3.2	POUŽITÍ ČASOVAČŮ A ČÍTAČŮ .....	49
3.2.1	Zadání.....	49
3.2.2	Řešení.....	50
3.2.3	Komentář k řešení .....	50
3.3	BLOKOVÉ PARAMETRY U FC A FB BLOKŮ.....	51
3.3.1	Zadání.....	51
3.3.2	Řešení.....	53
3.3.3	Komentář k řešení .....	54
3.4	POSLÁNÍ E-MAILU V PŘÍPADĚ VYHLÁŠENÍ POPLACHU .....	56
3.4.1	Zadání.....	56
3.4.2	Řešení.....	57
3.4.3	Komentář k řešení .....	57
3.5	PROPOJENÍ PLC S HMI A VYTVOŘENÍ VIZUALIZACE .....	58
3.5.1	Zadání.....	58
3.5.2	Řešení.....	61
3.5.3	Komentář k řešení .....	62
	<b>ZÁVĚR .....</b>	<b>66</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>68</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>71</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>73</b>
	<b>SEZNAM TABULEK.....</b>	<b>75</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>76</b>



## ÚVOD

V dnešní době je velkým trendem automatizovat všechny možné lidské činnosti, od náhrady ovládání složitějších zařízení po implementaci složitějších systémů. Za tímto trendem stojí především rozvoj elektroniky a vidina usnadnění různých činností člověka. Z těchto důvodů čím dál tím více do našich životů vstupují automatizované procesy a zařízení. S automatickým řízením se setkáváme denně. Ať už se jedná například o světelnou křižovatku, automobil, různé chytré spotřebiče, zabezpečení domu, nebo automatizovanou výrobu. Navíc se čím dál více zvyšují nároky na bezpečnost, kdy člověk vnáší do procesu velké bezpečnostní riziko.

Jedním ze zařízení užívaných pro automatické řízení je programovatelný automat (PLC, z anglického Programmable Logic Controller). Často se můžeme v německé literatuře setkat také s označením SPS (z německého Speicherprogrammierbare Steuerung). V češtině je někdy používaná zkratka PA (programovatelný automat). Tato zařízení mají nespornou výhodu oproti ostatním v odolnosti a spolehlivosti.

Tato práce se zaměřuje na PLC Simatic od firmy Siemens, a.s., které jsou na trhu velmi rozšířené jak v oblasti průmyslové automatizace, tak v oblasti průmyslu komerční bezpečnosti. Zároveň má pomoci studentům, kteří se poprvé dostanou do kontaktu se zařízením PLC S7 - 1214C DC/DC/DC, k tomu aby zjistili jak PLC funguje a jakým způsobem lze takové zařízení programovat a případně využít v oblasti komerční bezpečnosti.

Praktická část bakalářské práce se zaměřuje na vytvoření úloh, položených do oblasti zabezpečení, které budou složité studentům nejen k vyzkoušení programování PLC S7 - 1214C, ale také k zjištění možností tohoto PLC a vývojového prostředí TIA Portal. Navíc student bude seznámen s možností vizualizace procesu pomocí HMI panelu KTP700 Basic, taktéž od firmy Siemens, a.s., a s možností jeho propojení s PLC.

## I. TEORETICKÁ ČÁST

# 1 OBECNĚ O PLC

## 1.1 Historie

Programovatelný automat je jedním nejvíce používaných automatizačních prostředků. Již více než 50 let se vyskytuje v oblasti automatizace. Jeho vznik byl iniciován kvůli náhradě, do té doby používané, reléové logiky. Ta totiž přinášela jednu velkou nevýhodu. V případě potřeby změny logiky bylo velice složité přepojovat relé dle aktuální potřeby zařízení. Navíc zde byl velký prostor pro vznik potencionálních chyb v zapojení a tím pádem nefunkčnost celého zařízení, které bylo právě reléovou logikou řízeno. Další nesporná nevýhoda byla, že často tyto relé dokázali zabrat celou místnost. Všechny tyto nevýhody byly příchodem PLC odstraněny [1].

Jako první se myšlenkou náhrady reléových řídicích systémů začal zabývat americký výrobce automobilů General Motors. Výrobce automobilů chtěl vytvořit počítačový systém, který by dokázal pružněji reagovat a přizpůsobovat se potřebám výroby. General Motors tedy vyhlásila v roce 1968 soutěž na dodávku počítačového systému, který by dokázal tyto potřeby zvládnout. Ten pak chtěl aplikovat ve svých výrobních závodech. Celkem čtyři společnosti se přihlásilo do soutěže: Information Instruments, Inc (po roce přejmenována na Allen-Bradley, nyní Rockwell Automation), Digital Equipment Corp. (DEC), Century Detroit a Bedford Associates (později Modicon). Všechny společnosti navrhly řídicí systémy, které se částečně podobali minipočítačům, které se v tomto období vyráběli. Vítězem soutěže se stala společnost Bedford Associates a v roce 1969 bylo vyrobeno první PLC [6].

*První PLC neslo označení 084 jako 84. projekt společnosti Bedford Associates, která se následně začala zabývat vývojem, výrobou a servisem pro tento produkt, který byl uveden na trh pod názvem MODICON (zkratka ze slov MO-dular DI-gital CON-troller) jako modulární systém pro řízení procesů. Číslice 84 na konci označení se dále používaly jako synonymum pro PLC MODICON až do označení 984. Jeden z prvních PLC 084, který byl v provozu 20 let, je vystaven v sídle Modiconu, North Andover, Massachusetts. Jedním z vývojářů PLC byl Richard E. Morley a Dick Morley, který je považován za "otce" PLC [7].*

Návrh od společnosti Bedford Associates disponoval jednou obrovskou výhodou. Tím byl právě programovací jazyk zařízení Modicon, který se velmi podobal právě reléovým liniovým schémátům. Díky této vlastnosti se předpokládal hlavně plynulejší přechod a jednodušší zaškolení odborných pracovníků na tento systém.

System Modicon byl navíc jako jediný z návrhů opatřen odolným pouzdem přizpůsobeným do náročného průmyslového prostředí, což přispělo k vítězství mezi navrženými produkty. Ostatní dvě řešení totiž toto nenabízela [6],[7].

*Další PLC pocházela z produkce společnosti Allen Bradley PLC-5 (Rockwell) která určila směr vývoje a koncept byl tak zdařilý, že vyhovoval pro malé stroje i celé výrobní linky a principiálně byl kompatibilní od starších k novějším systémům [7].*

S rozvíjející se technologií mikroprocesorů a integrovaných obvodů se v roce 1973 objevuje na trhu první PLC od společnosti Siemens, které neslo název SIMATIC S3. Jednalo se o první kontrolér od společnosti Siemens, který nebyl programován přes přímé zapojení, nýbrž byl programován softwarově [5].

## 1.2 Charakteristika PLC

*Programovatelný automat je uživatelský programovatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů [1].*

PLC často bývá specializované pro úlohy, které nesou především logický charakter. Dnešní PLC však dokáží zastat roli také již i v regulační technice [1]. Lze je tedy požit pro široké spektrum aplikací. Mezi tyto aplikace může mimo řízení průmyslových a technologických procesů patřit také možnost řízení dnes moderních chytrých domácnosti, provádět procesní sběr dat, regulovat teploty v budovách nebo spolu se zabezpečovací systémy reagovat na poplašné zprávy.

Možnost pomocí PLC relativně rychlého sestavení systému poskytuje velkou výhodu. Není totiž potřeba vyvíjet celý systém, který by dokázal technicky zastřešit celou aplikaci. Stačí zanalyzovat, co vše je potřeba pro danou aplikaci a dle toho se rozhodnout, který typ PLC by byl ten nejvhodnější. Navíc u takzvaných modulárních PLC je výhodou, že v případě například rozšiřování nebo změny systému není potřeba kompletně vyměnit celé PLC. Stačí jen vybrat vhodný modul pro rozšíření.

PLC dokáže zastat velkou škálu aplikací. Ať už se jedná pouze o vstupní sestavu, kdy PLC může zajišťovat vyhodnocení binární a analogové senzory a předání informací některému z nadřazených systému. PLC může ale zastávat také roli pouze výstupního systému. *Například jako ovladač svítících nebo padáčkových segmentových zobrazovačů, jako ovladač souboru pohonů nebo souboru elektrických spotřebičů a jiných akčních členů [1].*

PLC se používá ale i pro aplikace, kdy není potřeba použít fyzické vstupy a výstupy. Může sloužit jako striktně komunikační jednotka a využívat se jako komunikační most. To se používá v případě, kdy je potřeba zajistit komunikaci více systému, které nemají mezi sebou možnost komunikovat napřímo. PLC pak zprostředkovává inteligentní převodník komunikačních rozhraní a adaptér mezi protokoly různých průmyslových sběrnic.

*Technické vybavení programovatelných automatů je navrženo tak, že jsou extrémně spolehlivé i v drsných průmyslových podmínkách, jsou odolné proti rušení i poruchám, vyznačují se robustností a spolehlivostí. PLC bývají vybaveny i vnitřními diagnostickými funkcemi, které průběžně kontrolují činnost systému a včas zjistí případnou závadu, lokalizují ji, bezpečně ji ošetří a usnadní její odstranění [1].*

### 1.3 Struktura PLC

Struktura programovatelného automatu se může lišit dle požadované aplikace. Jsou ale základní součásti, bez které se žádné PLC v jakékoliv aplikaci neobejde. Jednou z hlavních součástí je CPU (central procesor unit) neboli procesor, který zpracovává uživatelský program PLC. Nedílnou součástí jsou paměti pro ukládání systémových a uživatelských dat. Většina PLC v jakémkoliv provedení obsahuje také digitální i analogové vstupy a výstupy. Součástí bývá také komunikační procesor. To především z důvodu „odlehčení práce“ CPU. Komunikační procesy bývají totiž často velmi náročné na výpočetní výkon, což by mohlo mít negativní vliv na řídicí procesy. Dalšími součástmi mohou být třeba také rychle čítače.

#### 1.3.1 CPU

CPU (Central Processor Unit) neboli centrální procesorová jednotka je nejzákladnější část celého programovatelných automatů, protože jim poskytuje veškerou „chytrou“ a udává výkon celého PLC. Z pohledu PLC totiž CPU značí mikro počítač, který se stará nejen o zpracování uživatelského programu ale obsluhuje i uživatelskou a systémovou paměť. Zároveň však ale obstarává komunikační a časové funkce PLC. Paměťový prostor, který je v PLC součástí CPU jednotky se z pravidla dělí na uživatelskou a systémovou paměť. Uživatelská paměť slouží k ukládání uživatelského programu. Nejčastěji bývá typu EPROM nebo EEPROM a má kapacitu řádově desítky KB a jednotky MB. Systémová paměť je určena k uložení systémového programu neboli firmwaru, který řídí činnost celého PLC. Uživateli dostupné registry, zápisníkové registry, čítače, časovače a většinou i vyrovnávací registry

pro obrazy vstupů a výstupů jsou zpravidla uloženy v paměti RAM. Pro zálohování nebo rozšiřování uživatelské paměti se používají paměťové karty FLASH [1],[8].

### 1.3.2 Digitální vstupní a výstupní jednotky

Digitální neboli binární vstupy a výstupy pracují se zařízeními a akčními členy, které komunikují pomocí dvouhodnotového signálu na jejich vstupu nebo výstupu. Jedná se o zařízení, jako jsou tlačítka, přepínače, senzory, koncové spínače, LED diody, cívky elektropneumatických rozvaděčů a další.

Digitální způsob komunikace přináší velkou výhodu a to je odolnost proti rušení signálu. Navíc digitální vstupní/výstupní jednotky slouží jako ochrana elektroniky uvnitř PLC, protože jsou galvanicky odděleny pomocí optočlenů. Digitální vstupní jednotka filtruje vstupní signály od nežádoucích rušivých impulsů. U výstupů je zase signál zesilován na potřebnou úroveň signálu. Jak digitální vstupy, tak i výstupy bývají opatřeny LED diodou, která slouží k diagnostikování a nalezení problému v případě poruchy [1],[8],[9].

### 1.3.3 Analogové vstupní a výstupní jednotky

Analogové vstupní a výstupní jednotky slouží ke komunikaci se spojitým prostředím. Vykonnávají tak převedení spojitého elektrického signálu na diskrétní digitální signál a naopak. K tomuto účelu obsahují analogově – digitální (A/D převodník) a digitálně analogové převodníky (D/A převodníky). Jelikož analogové jednotky pracují s elektrickým signálem, tak veličiny, které jsou zpracovávány, jsou buď napětí – U, nebo proud - I. Znamená to, že snímač, který snímá spojitý signál jako je například teplota, musí nejdříve převést tuto spojitou veličinu na elektrický signál [1].

K analogovým vstupům pak lze připojit snímače tlaku, teploty, osvětlení, vzdálenosti, hladiny, síly, nebo rychlosti. Analogové výstupy se využívají k řízení technologických procesů s použitím vhodných akčních (výkonových) prvků jako je řízení rychlosti motorů, regulační smyčku, řízení průtoku kapalin nebo nastavení svítivosti [8],[10],[11].

### 1.3.4 Komunikační jednotky

Důležitou funkcí PLC je dokázat komunikovat se vzdálenými moduly, operátorskými panely (HMI, Human Machine Interface – rozhraní mezi člověkem a strojem), různými systémy,

podsystemy a vzdálenými periferiemi. Pro tyto účely je PLC vybaveno komunikační jednotkou, která tuto komunikaci zprostředkovává. Typy rozhraní, které komunikačních jednotky obsahují, jsou například RS 232, RS 485, Ethernet nebo USB [1],[8].

### 1.3.5 Ostatní jednotky

Ze zpracování programu programovatelným automatem plyne jedna nevýhoda, díky které není PLC schopno detekovat velmi krátké impulzy. Z tohoto důvodu se pro možnost rychlého čítání připojuje jednotka pro rychlé čítání. V případě připojení polohovacího modulu, pak PLC dokáže plnit úlohy, u kterých je potřeba přesného polohového měření pomocí absolutních a inkrementálních snímačů, kdy je potřeba zpracovávat rychlý sled impulzů [1].

## 1.4 Dělení PLC

### 1.4.1 Kompaktní PLC

V kompaktním provedení PLC je vše integrováno v jednom uzavřeném pouzdře. To znamená, že vše podstatné jako je CPU, digitální vstupy a výstupy, často i analogové vstupy a výstupy a i komunikační rozhraní (RS 232, RS 485, Ethernet) bývají součástí tohoto jednoho uzavřeného pouzdra. Výhoda tohoto provedení je nižší cena. Navíc jelikož je vše integrováno v jednom provedení, tak signály nemusí procházet přes řadič sběrnice, čímž se zrychlí přístup k periferiím. Nevýhodou tohoto typu PLC bývá menší paměť. To především protože jsou kompaktní PLC určené především pro menší aplikace, kde se nepředpokládá velké rozšiřování. Některé typy totiž ani nemají možnost dalšího rozšiřování (přidání vstupů nebo výstupů, technologických modulů). Někteří výrobci však tuto možnost umožňují. Ne ale v takovém rozsahu jako je to možné u modulárních PLC [12].



*Obr. 1 Kompaktní PLC Simatic S7 - 1200 od společnosti Siemens [22]*

#### 1.4.2 Modulární PLC

Modulární provedení PLC umožňuje z jednotlivých modulů, které výrobce nabízí, složit PLC dle potřeb dané aplikace. Vždy ale musí, jeden z modulů být právě CPU. K němu se přes sběrnici pak připojují další periferie (moduly). V praxi pak většinou bývají tyto moduly poskládány do jedné řady a umístěny v rámu. Pokud množství potřebných modulů je větší než je možné umístit na rám, tak se dá použít další rozšiřující rám. Nejčastěji pak bývají připojovány moduly pro digitální vstupy a výstupy, analogové vstupy a výstupy, komunikační moduly, modul pro rychlé čítače nebo různé technologické moduly. Často jeden z modulů bývá také zdroj napájení. Modulární systémy se používají pro aplikace středního a velkého rozsahu a tam kde se předpokládá v budoucnu potřeba rozšíření systémů, které PLC řídí. Není totiž pak třeba kupovat jiné PLC. Což by například bylo pravděpodobné u kompaktního systému. Stačí pouze dokoupit modul, který je zapotřebí. Modul nainstalovat a upravit stávající projekt pro PLC. U modulárního systému se předpokládá velké množství připojených modulů. Tím pádem se dá očekávat potřeba zpracování většího množství signálů. Proto modulární PLC obvykle jsou vybaveny větší pamětí jak pro program, tak i pro data. Nevýhodou tohoto systému porovnáním s modulárními systémy bývá několikanásobně větší cena. Stejně jako kompaktní PLC i modulární PLC jsou vybaveny rozhraním pro programování [12].





*Obr. 2 Modulární PLC Simatic S7 - 1500 od společnosti Siemens [23]*

### 1.4.3 Mikro PLC

Jedná se o rozměrově malá PLC, které jsou z pravidla velice levné. Mají stejně jako kompaktní PLC pevně danou konfiguraci vstupů a výstupů. Nejčastěji bývají osazeny pouze digitálními vstupy a výstupy. Nové mikroPLC však také již obsahují v základním provedení analogové vstupy a výstupy. Uživatel před integrací však musí důkladně zvážit, které mikro PLC je vhodné pro jeho aplikaci, protože možnost dalšího rozšíření je velmi nízká. Nejčastěji jsou tedy používány pro malé aplikace, jako jsou jednoduché stroje nebo mechanismy, kdy slouží jako „chytré relé,“ které dokáže nahradit jednoduchou reléovou logiku. Dochází pak k redukci nákladů, které by byly potřeba při použití reléové logiky. Integrátor v tu chvíli ušetří za materiál, jako jsou relé, stykače nebo časové relé. Programové možnosti jsou sice

z pohledu programátora obvykle zredukovány na minimum, avšak pro potřeby malých aplikací jsou dostačující [1].



Obr. 3 Mikro PLC SIPLUS LOGO 24CE 8 od společnosti Siemens [24]

Všechna provedení PLC jsou většinou určena pro montáž na lištu DIN do rozvaděčů.

## 1.5 Zpracování programu v PLC

Program v PLC je zpracováván cyklicky, což znamená, že program je stále a opakovaně prováděn bez toho aniž by to musel sám programátor programově ošetřit.

První úkon, který PLC provede na začátku každého cyklu, je provedení systémových operací v CPU, které nelze uživatelsky ovlivnit. Tím se provede vnitřní kontrola a diagnostika systému. Následně dochází k načtení stavů na vstupech PLC a jejich uložení do paměti. Takzvané uložení obrazu vstupů. PLC pak při zpracování programu pracuje právě s tímto obrazem vstupů nikoli s fyzickými vstupy. Tím je dosaženo jednoznačnosti vstupního signálu během zpracování programu a omezení chyb, které by mohli vzniknout při nevhodném souběhu měnících se hodnot. Vyhodnocování více logických podmínek v rámci jednoho

cyklu programu PLC s odlišným stavem stejného signálu by pak vedlo k nejednoznačným stavům a výsledkům.

Ve chvíli kdy je dokončeno načtení stavů vstupních signálů, tak dochází ke zpracovávání uživatelského programu. Program bývá umístěn v hlavním organizačním bloku PLC a je zpracováván shora dolů. Je důležité si uvědomit, že programové bloky, které nejsou umístěny nebo volány z hlavního organizačního bloku, nejsou prováděny. V průběhu zpracovávání programu v PLC na základě vytvořeného algoritmu jsou generovány výstupní signály. Stejně jako vstupní signály, tak i u výstupní signály pracují s obrazem hodnot a to s výstupním obrazem hodnot. Změny na vstupech generované programem, se tedy zapisují do výstupního obrazu hodnot. Po provedení poslední instrukce PLC změní stav výstupů dle aktuálního obrazu vstupu, čím se stejně jako u vstupních signálů zajistí jednoznačnost stavů výstupních signálů. V dalším kroku provede PLC aktualizaci časových údajů pro časovače a systémové registry, ošetří komunikaci a provede ještě řadu režijních úkonů[1],[2],[3].

PLC, která na rozdíl od automatů Siemens nebo Rockwell nepoužívají tzv.obraz procesu jsou například automaty SAIA PCD. V automatech SAIA PLC se vždy čte a zapisuje okamžitě z a do vstupů a výstupů. Je to velmi výhodné, když nějaké části programu (a jejich vstupy a výstupy) mají být zpracovány častěji, než zbytek programu [25].

## 1.6 Programovací jazyky PLC

Pro programování PLC je normou IEC 61131-3 stanoveno několik programovacích jazyků. Programátor pak má při programování možnost volit programovací jazyk, který je nejvhodnější pro tu danou aplikaci. Norma IEC 61131-3 má za úkol definici a sjednocení těchto jazyků pro programovatelné automaty. Nezáleží při tom od jakého výrobce PLC pochází. Představa možnosti přenositelnosti programů mezi jednotlivými PLC od různých výrobců je však takřka nemožná. Umožňuje to ale rychlejší orientaci v programu, pokud se programátor dostane k PLC od jiného výrobce.

Každý programovací jazyk je vhodný pro jiný druh aplikace. Norma sjednocuje způsoby programování pomocí těchto jazyků pro výrobce PLC. K programování nabízejí PLC systémy specializované jazyky, původně navržené pro snadnou, názornou a účinnou realizaci logických funkcí. Jazyky systémů různých výrobců jsou podobné, nikoliv však stejné. Přenositelnost programů mezi PLC různých výrobců není možná, daří se to obvykle jen mezi systémy téhož výrobce [1].

### 1.6.1 Jazyk mnemokódů

(„Instructions List“, „IL“, v německé terminologii „Anweisungslist“, „AWL“) je obdobou assembleru u počítačů, a je také strojově orientován. To znamená, že každé instrukci PLC systému odpovídá stejně pojmenovaný příkaz jazyka. Jazyky mnemokódů poskytují i obvyklý „assemblerský komfort“, tj. aparát symbolického označení návěští pro cíle skoků a volání, symbolická jména pro číselné hodnoty, pro pojmenování vstupních, výstupních a vnitřních proměnných a jiných objektů programu (datových bloků a tabulek, struktur a jejich prvků), pro automatické přidělování paměti pro uživatelské registry a pro jiné datové objekty, pro jejich inicializaci (zadání počátečního obsahu), pro zadávání číselných hodnot v různých číselných soustavách [1].

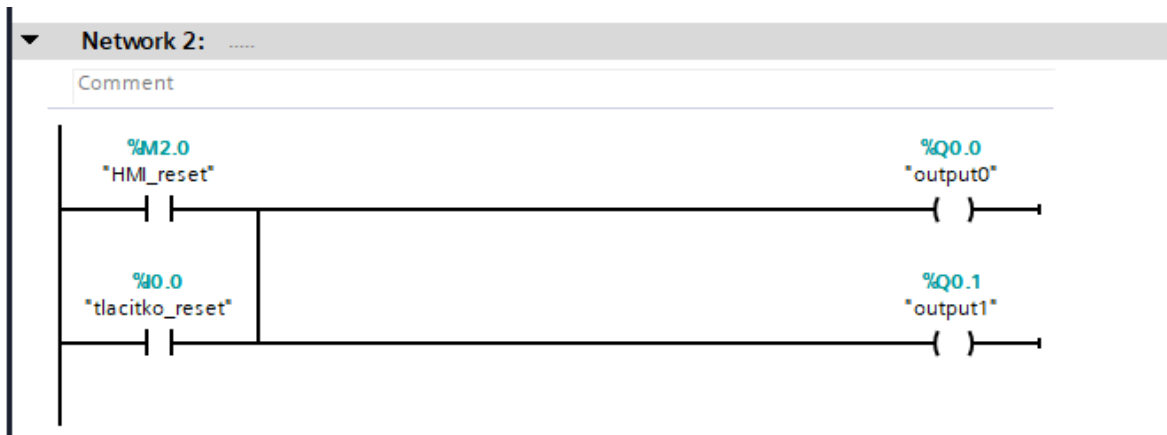
Výhodou programování pomocí mnemokódů spočívá v přesné definici chování programů. Navíc obvykle program vytvořený pomocí mnemokódů je paměťově nenáročný a rychlejší na zpracování. Jasnou nevýhodou je na první pohled nepřehlednost programu a mnohdy velice zdlouhavé psaní programu. Jazyk je navíc náročný na znalost programových registrů.

V prostředí Siemens se můžeme setkat s označením STL. Programovací jazyk STL není ale u nové řady PLC S7 - 1200 v TIA Portálu dostupný a nelze ho u toho typu PLC použít. Lze jej ale použít u řady S7 - 300/400 a u nejnovější řady S7 -1500.

### 1.6.2 Jazyky kontaktních schémat

(„Ladder Diagram“, „LD“, německy „Kontaktplan“, „KOP“) je grafický. Program se základními logickými operacemi zobrazuje schéma ve formě obvyklé pro kreslení schémat při práci s reléovými a kontaktními prvky. Pouze symboly pro kontakty a cívky jsou zjednodušeny, aby mohly být vytvářeny semigraficky: spínací kontakty jako dvojice svislých čárek, rozpínací kontakt je navíc „přetržen“ lomítkem, cívky jsou označovány dvojicí závorek. Funkční bloky (např. čítače, časovače) jsou kresleny jako obdélníkové značky. Instrukce, které nemají svou analogii v kontaktní symbolice (a těch bývá většina) se obvykle zobrazují jako dvojice závorek nebo obdélníková značka s vepsaným mnemokódem instrukce. Jazyk kontaktních schémat je výhodný při programování nejjednodušších logických a v případech, kdy s ním pracuje personál, který nezná (a nechce znát) tradiční počítačové programování. Je nezastupitelný při požadavku rychlého servisu, obzvláště pokud ladící prostředky dovolují zvýraznit na schématu „vodivou cestu“. Pak je nalezení závady na stroji (třeba vadného

spínače) otázkou několika minut. Pokud v programu převažují složitější instrukce (třeba aritmetické instrukce nebo logické s vektorovými operandy, skoky a volání), pak je kontaktní schéma již násilně a postrádá svou názornost [1].

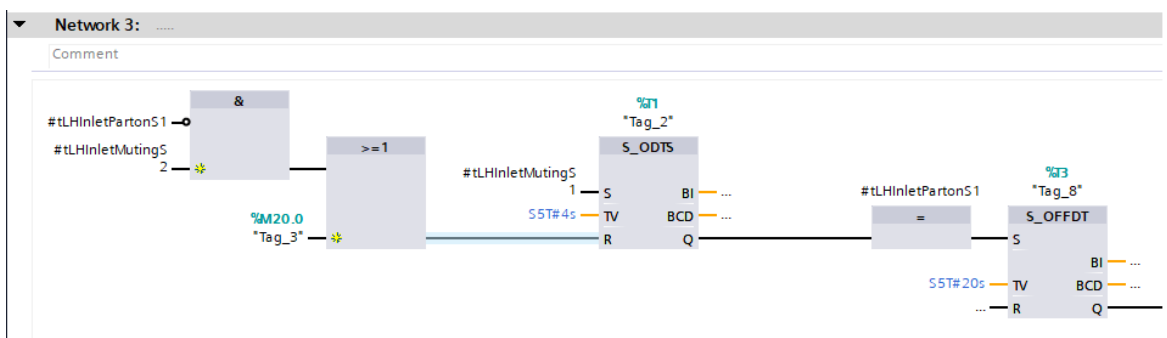


Obr. 4 Příklad programovacího jazyku LAD používaného v TIA Portalu

Nevýhodou je tedy nevhodnost použití při aritmetických operacích a při práci s daty. U programovatelných automatů od společnosti Siemens je tento jazyk označován jako LAD – ladder diagram.

### 1.6.3 Jazyk logických schémat

Jazyk funkčních bloků, „Function Block Diagram“ německy „FUP“ je opět grafický. Základní logické operace popisuje obdélníkovými značkami. Výška značky je přizpůsobena počtu vstupů. Své značky mají i ucelené funkční bloky, např. čítače, časovače, posuvné registry, paměťové členy, ale i aritmetické a paralelní logické instrukce. Vychází vstříc uživatelům zvyklým na kreslení logických schémat pro zařízení s integrovanými obvody. Obdobný, ale obecnější, jazyk se využívá při popisu a programování systémů, zpracovávajících analogové proměnné, při programování regulačních a měřicích úloh [1].



Obr. 5 Příklad programovacího jazyka FBD v TIA Portalu

Z pohledu programátora je vhodný pro sestavení připravených programových bloků dle potřeb. Program může připomínat zapojování logických obvodů pomocí hradel. Nevýhodou stejně jako u jazyku kontaktních schémat je problémová práce s aritmetickými operacemi a daty. Možnost použít „FBD“ je u všech PLC řady S7 od společnosti Siemens.

#### 1.6.4 Jazyk strukturovaného textu

(„Structured Text“ „ST“) je obdobou vyšších programovacích jazyků pro PC nebo mikrořadiče (např. Pascal nebo C). Umožňuje úsporný a názorný zápis algoritmů [1].

```
CASE #Robots[#tLC].Status.Index OF
  1: // Wait for Job
    IF #Robots[#tLC].Status.LoadOP XOR #Robots[#tLC].Status.PickOP XOR #Robots[#tLC].Status.TQOP
      XOR #Robots[#tLC].Status.SafePosOP AND NOT #Robots[#tLC].Status.TPEnabled THEN
        #Robots[#tLC].Status.Index := 2;
      END_IF;
```

Obr. 6 Příklad použití programovacího jazyka SCL v TIA Portalu

Výhoda spočívá v možnosti použití pro náročné aritmetické operace, přesuny dat a práci s textovými řetězci. Nutnost znát příkazy, přesnou syntaxi a menší přehlednost při zpracovávání velkého počtu logických operací tento programovací jazyk ale znevýhodňují. V prostředí Siemens se používá označení SCL a lze jej použít u všech automatů nové generace S7. To znamená jak S7 – 1200, tak i S7 – 1500.

### 1.7 Využití PLC se zabezpečovacími systémy

V současné době se čím dál tím víc začíná skloňovat pojem „inteligentní budova.“ Toto označení se přiřazuje objektu, kde je většina vnitřních systémů integrovaná do jednoho systému a řízena některým s nadřazených systémů. Celý objekt je pak dokonale zautomatizován a všechny systémy mezi sebou dokážou komunikovat a předávat si informace. Jedním z nadřazených systémů, který dokáže řídit celou inteligentní budovu, může být právě PLC.

Majitelé inteligentních budov mohou díky integraci systémů ušetřit nejen provozní náklady budovy, ale i náklady spojené s počáteční instalací. Integrací se také většinou docílí eliminace nadbytečných řídicích prvků. Řízení a komfort se u toho typu objektu daleko zjednoduší a zkomfortní. Představa automatického rozsvěcování a zhasínání světel, regulovatelnosti teploty dle zadané hodnoty v mobilním telefonu, nebo automatické zatažení žaluzií po příchodu do místnosti je pro každého uživatele příjemná [14].

Mezi vnitřní systémy inteligentní budovy lze zahrnout vzduchotechniku, vytápění, osvětlení, stínící techniku, zásuvkový okruh ale také zabezpečovací systémy jako jsou PZTS, CCTV a EPS. PLC pak přes komunikační sběrnici se všemi systémy dokáže komunikovat a řídit je. V případě zabezpečovacích systémů však pouze přijímá signály a na základě nich dokáže v případě vyhlášení požáru nebo zjištění nedovoleného vniknutí provést určitá opatření [14].

Pro vzájemnou komunikaci jednotlivých zařízení a senzorů instalovaných v inteligentních budovách je nezbytné použít sběrnici, která by dokázala použít shodný způsob přenosu a byla nejen dostatečně robustní, všestranná, ale i zabezpečená. Jednou ze sběrnic, které se používají pro tyto účely je sběrnice KNX .

### 1.7.1 Systém KNX

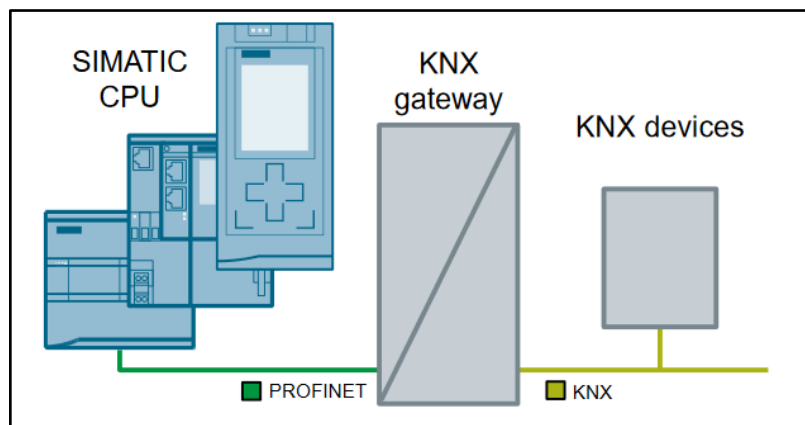
KNX je evropský instalační systém sloužící k síťovému připojení jednotlivých prvků používaných v inteligentních objektech a k jejich vzájemné komunikaci. Pro přenos dat je možné využít různá komunikační média:

- kroucený pár (KNX TP) – komunikace po krouceném páru datového kabelu (sběrnice)
- silová síť KNX (KNX PL) - používá existující 230 V silovou síť
- radiofrekvenční přenos (KNX RF) – komunikace rádiovým signálem
- IP protokol (KNX IP) – komunikace přes Ethernet

Výhoda KNX spočívá v jeho univerzálnosti a možnosti propojit zařízení od různých výrobců. O standartizaci systému KNX se stará asociace Konnex. Asociace sdružuje vedoucí firmy v oblasti automatizované elektroinstalace v Evropě [16].

### 1.7.2 Připojení KNX zařízení k PLC SIMATIC

Existují programovatelné automaty, které mají již integrované rozhraní pro komunikaci KNX. To však neplatí pro programovatelné automaty řady SIMATIC od společnosti Siemens. Jediná možnost jak připojit prvky KNX k automatům od společnosti Siemens je, že se použije takzvaný gateway, který zprostředkuje komunikaci dvou odlišných komunikačních sítí založených na rozdílných protokolech – KNX TP a Profinet, což je průmyslová komunikační sběrnice vyvinuta společností Siemens [13].



Obr. 7 Komunikace mezi PLC Siemens a zařízením KNX [13]

V manuálu, který nabízí Siemens na svých webových stránkách odkazuje na gateway Triple-X od společnosti MBS GmbH. Pomocí něj lze pak zrealizovat výše popsanou komunikaci [13].

### 1.7.3 Problematika připojení zabezpečovacích systémů

Právě systémy jako je KNX nebo jiné instalační systémy pro inteligentní objekty umožňují relativně jednoduché propojení prvků a jejich integraci. V případě připojení zabezpečovacích systémů, však musí být dodržena legislativa s touto problematikou spojená. Důležitým faktem je, že většina společností, která obdobné systémy zřizuje a instaluje, opomíjí skutečnost, že prvky a celé zabezpečovací systémy připojené například právě KNX sběrnici musí být nainstalovány pouze pracovníky a společnostmi, které mají příslušnou koncesi pro instalace zabezpečovacích systémů a certifikát pro daný zabezpečovací systém [14].

*Jelikož v okamžiku, kdy je bezpečnostní systém připojen prostřednictvím řídicí sběrnice domu (bude tedy provedena integrace), stává se i tato sběrnice podle platného znění normy součástí zabezpečovacího systému, a musí být tedy instalována odpovídající osobou (firmou) s odpovídajícím bezpečnostním ověřením. Ještě vážnější je však problém z pohledu třídy bezpečnosti. Podle normy platí, že bezpečnostní systém má tu třídu bezpečnosti, kterou má prvek systému s nejnižší bezpečnostní certifikací. A po připojení jakéhokoliv prvku bez bezpečnostní certifikace k bezpečnostnímu systému ztrácí celý systém bezpečnostní certifikaci. To je problém především u objektů s předepsanou bezpečností. Pro pojišťovnu je to důvod krátit plnění v případě škodné události [14].*



## 2 PLC SIEMENS

Jedním z nejznámějších výrobců programovatelných automatů současnosti je společnost Siemens a to nejen v České republice, ale i ve světě. K těm základním programovatelným automatům, které Siemens v současné době nabízí a podporuje jejich rozvoj, jsou programovatelné automaty skupiny SIMATIC (S7 - 1200, S7 - 1500). Díky těmto programovatelným automatům, které nabízí v různých provedení, lze komplexně pokrýt většinu automatizačních úloh. Softwarový produkt, který Siemens vyvinul pro možnosti programování PLC, se nazývá Step 7. Spolu s programovacím nástrojem WinCC pro panely HMI, jsou integrovány do inženýrského prostředí TIA Portal (Totally Integrated Automation Portal). V oblasti výrobního průmyslu se často vyskytují PLC předešlé generace, které nesou označení S7 - 300 a S7 - 400. To především protože spousta aplikací byla původně na těchto programovatelných automatech vystavena před lety a běží dodnes. Migrace na novější systém většinou navíc nebývá otázkou krátké doby a vyžaduje prostoj na výrobní lince kvůli instalaci, což se například managementu výrobní společnosti nemusí zamlouvat.

V této bakalářské práci se ale budu věnovat především řídicímu systému S7 – 1200, který bude dostupný v laboratořích Univerzity Tomáše Bati spolu se softwary Step 7 Basic a WinCC Basic integrovanými do TIA portálu. Systém S7 - 1200 může být navíc dobrou volbou pro řízení procesů v inteligentních budovách zmiňovaných v předešlé teoretické části, této bakalářské práce.

### 2.1 SIMATIC S7 - 1200

#### 2.1.1 Základní popis

Programovatelný automat SIMATIC S7 - 1200 je kompaktní systém, který do jisté míry ale umožňuje modularitu a tím pádem možnost rozšíření stávajícího automatu o další moduly. Určen je především pro automatizační úlohy malého a středního rozsahu. Cenově patří do skupiny dostupnějších programovatelných automatů současnosti. Všechny typy, které Siemens nabízí, jsou vybaveny komunikačním rozhraním Ethernet pro moderní průmyslovou komunikaci – Profinet. Zároveň díky několika integrovaným funkcím, jako je například možnost rychlého čítání u některých vstupů, je možné tento programovatelný automat použít pro opravdu široké spektrum úloh. Výhodou je i přítomnost analogových vstupů u všech dostupných typů PLC S7 - 1200 [16].

Dostupné je pět základních jednotek, které nesou označení CPU 1211C, CPU 1212C a CPU 1214C, CPU 1215C a CPU 1217C. Odlišnosti mezi nimi najít především v počtu digitálních vstupů a výstupů, přítomnosti analogových výstupů, počtem rychlých čítačů, velikostí nahrávací a pracovní paměti nebo počtem modulů, které je možné připojit. U všech variant je možné zvolit si napájecí napětí (24VDC nebo 230 VAC) a typ výstupů (tranzistorové nebo releové). Jedinou výjimku tvoří varianta CPU 1217C, jenž je dostupná pouze s napájením 24VDC a tranzistorovými výstupy [16].

Nahrávací paměť (load memory) sloužící k uložení programu, dat a konfigurace je energeticky nezávislá, což znamená, že při odpojení napájení nedochází ke smazání programu. Je možné ji rozšířit pomocí paměťové karty, kterou Siemens nabízí. V CPU je také obsažená pracovní paměť (work memory). Pracovní paměť používá automat při vykonávání programu pro uložení konkrétních částí programu. Je energeticky závislá a dochází k její smazání při odpojení napájení. Posledním typem paměti v CPU je obnovitelná paměť (retentive memory). Používá se k uložení dat z pracovní paměti, které se mají zachovat při výpadku napájení [16].

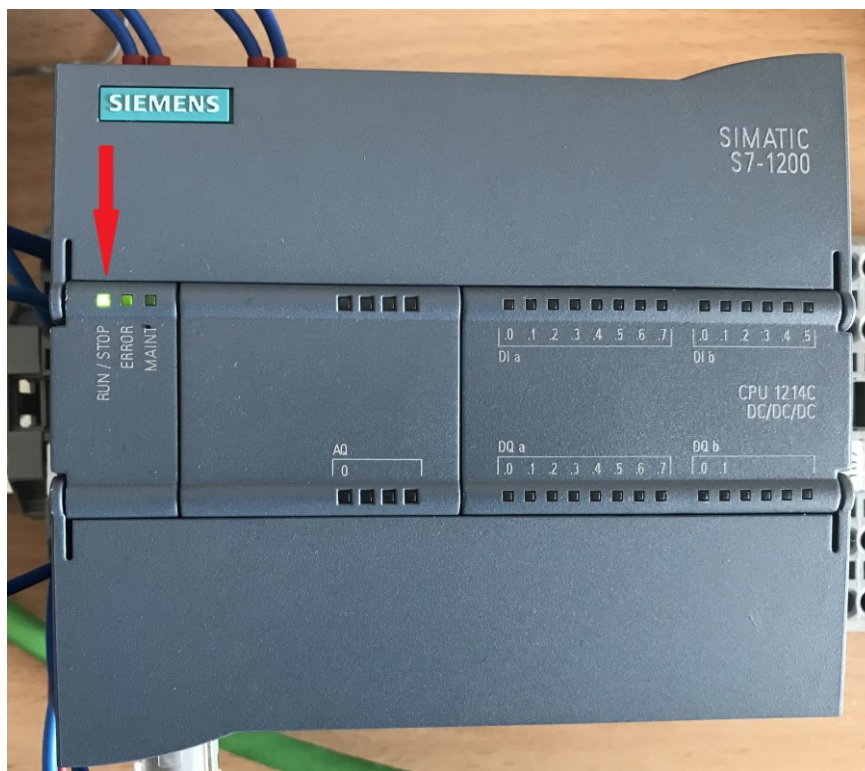
### 2.1.2 Pracovní módy S7 - 1200

Automat obsahuje celkem tři pracovní módy:

- V módu „**STOP**“ není prováděn program nahraný v PLC a uživatel může nahrávat projekt z programovacího zařízení.
- V módu „**STARTUP**“ provádí automat start systému.
- Mód „**RUN**“ slouží k cyklickému provádění uživatelského programu.

Indikační LED diody na přední straně automatu umožňují zjistit aktuální status. Status LED dioda „**RUN / STOP**“ určuje pracovní mód, ve kterém se PLC nachází:

- žlutá značí STOP mód,
- zelená značí RUN mód,
- blikání značí STARTUP mód [18].



Obr. 8 Signalizace pracovního módu PLC S7 - 1214C DC/DC/DC

Oproti jiným programovatelným automatům od společnosti Siemens, tento nemá fyzický přepínač pro přepnutí módů. Přepínání módu je možné ale z inženýrského prostředí TIA Portal. LED dioda „ERROR“ signalizuje chybu automatu. Dioda s označením „MAINT“ značí potřeby údržby [18].

### 2.1.3 Popis SIMATIC S7 - 1214C DC/DC/DC

V laboratořích UTB se student setká s programovatelným automatem, který nese označení S7 - 1214C DC/DC/DC. Už z názvu je patrné, že přívodní napětí automatu je 24VDC a disponuje pouze tranzistorovými výstupy. Obsahuje celkem 14 digitálních vstupů, 10 digitálních výstupů a 2 analogové vstupy. Pokud by byli potřeba analogové vstupy, tak by bylo nutné pořídit rozšiřující modul s analogovými výstupy. Automat je možné rozšířit o maximálně osm technologických modulů a tři komunikační moduly. Load memory pro uživatelský program je možné rozšířit paměťovou kartou SIMATIC Memory Card. Obrazy vstupů (PII, Proces Image Input) a výstupů (PIQ, Proces Image Output) pro uložení hodnot na fyzických vstupech a výstupech mají shodně velikost 1024 bajtů. Maximální počet bloků použitelných v automatu je stanoven na 1024 bloků. Jedná se o bloky nesoucí názvy OB, FB,

FC a DB. Popisu jednotlivých bloků bude věnována kapitola s názvem „Programová struktura.“ Pro komunikační možnosti automat obsahuje Ethernetový port. Jako u všech automatů řady S7 - 1200 tak zmiňovaného konkrétního typu je možné použít pouze programovací jazyky LAD, FBD a SCL. Řada S7 - 1200 totiž nepodporuje STL a GRAPH. Tyto programovací jazyky jsou dostupné například u řady S7 - 1500 nebo u automatů předešlé generace S7 - 300 a S7 - 400 [16].

#### 2.1.4 WEB server

Všechny automaty řady S7 -1200 mají integrovaný webový server umožňující přístup do automatu z počítače či chytrého telefonu pomocí webových stránek integrovaných přímo v PLC. Uživatel tak může jednoduše zjistit informace o typu používaného PLC, typu používaných modulů, komunikaci, diagnostickém bufferu, hodnotě proměnných a uložených souborech. Přístup pomocí webového serveru může být zaheslován, tak aby k němu měli oprávnění pouze důvěryhodné osoby. Povolení používání webového serveru je nutné aktivovat v konfiguraci PLC v TIA Portalu. PLC S7 - 1200 dovoluje také podporu uživatelsky vytvořených webových stránek. Z pohledu inteligentních domácností je to zajímavá vlastnost, protože to umožňuje velké možnosti řízení budovy například přímo z mobilního telefonu [16],[21].

## 2.2 TIA Portal

TIA Portal (Totally Integrated Automation Portal) je softwarové prostředí vyvinuté společností Siemens za účelem integrace všech softwarových balíčků, potřebných ke zprovoznění některé z automatizačních úloh, do jednoho společného softwaru. Umožnila tím uživatelům spravovat části projektu (konfigurace a parametrizace hardwaru, aplikace pro PLC, projekty pro HMI nebo síťové konfigurace komponent a komunikačních prvků, diagnostikování poruch) v jednom vývojovém prostředí a uchovat je jako jeden vytvořený projekt (soubor) pro celou automatizační úlohu [17].

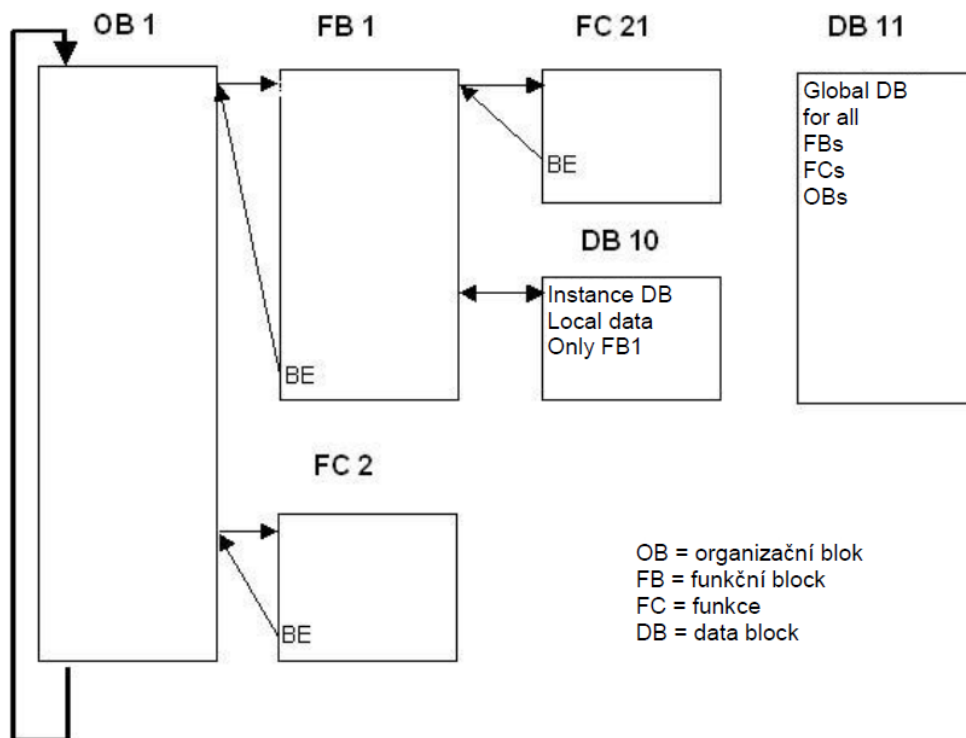
### 2.2.1 Step 7 Basic

V případě této bakalářské práce se setkáme s integrací softwarového balíčku Step 7 Basic, který slouží k vytváření aplikací pro programovatelný automat řady S7 - 1200. Programovací jazyky podporované touto verzí jsou LAD, FBD a SCL. Za zmínku stojí potřeba vyšší verze

Step 7 a to Step 7 Professional v případě, kdy chceme vytvářet aplikace pro ostatní programovatelné automaty řady SIMATIC (S7 - 300, S7 - 400, S7 - 1500). Ta dovoluje používat i programovací jazyk STL a GRAPH [17].

### 2.2.2 Programová struktura

Struktura programu programovatelného automatu S7 - 1200 se skládá z takzvaných bloků (OB, FB, FC, DB). Program nebo programové bloky vytvořené uživatelem musí být umístěny v bloku OB1. Tento blok je totiž automaticky volán v každém cyklu PLC. Z těchto důvodů blok OB1 je generován automaticky při vytváření projektu. Bloky nebo operace, které nejsou v tomto bloku volány, nejsou vůbec prováděny a to i v případě, kdy jsou do automatu nahrány. V rozsáhlých úlohách se program dělí do menších jednodušších bloků (FB, FC). Pro lepší představu můžeme říct, že tvoří takzvané podprogramy. Přináší to výhody, jako jsou přehlednost a srozumitelnost programu. Mohou být volány z jakéhokoliv jiného bloku. Samotný blok nebo blok, ze kterého je blok volán musí být umístěn v OB1 jinak není prováděn. Po jejich skončení se vrací program opět do bloku za místo, ze kterého byl volán [16], [18].



Obr. 9 Schématické zobrazení volání programových bloků z OB1 [16]

### 2.2.2.1 OB bloky

Organizační bloky (OB, Organization block) slouží jako rozhraní mezi operačním systémem PLC S7 - 1200 a uživatelským programem. Sám operační systém je pak volá a řídí následující procesy:

- cyklické zpracování programu v OB1,
- zpracování alarmů,
- zpracování chyb,
- přerušování uživatelským programem.

Organizační bloky není možné volat navzájem a ani z funkčních bloků (FB) a funkcí (FC). Každé OB má svojí prioritu. OB s větší prioritou je vždy operačním systémem voláno nejdříve. V případě OB1 je zvolená priorita nejnižší. To znamená, že v případě volání jakéhokoliv OB s vyšší prioritou může přerušit provádění uživatelského programu a provést patřičné procesy [16],[17],[19].

### 2.2.2.2 FC bloky

Funkce (FC, Function) umožňuje uživateli rozdělit program do menších přehlednějších a lépe zpravujících se celků. Pro lepší představu vezmeme znovu v úvahu inteligentní budovu, kde máme několik prvků, které je potřeba řídit a komunikovat s nimi. Právě pro každý prvek může být vytvořena jedna funkce. Příkladem může být funkce, která bude pouze sloužit pro řízení komunikace se zabezpečovacími systémy. Tu pak můžeme kdykoliv volat z OB1 k získání aktuálního stavu zabezpečení. Funkce nemají přidělenou paměť. Je možné použít dočasné proměnné. Ty jsou ale po dokončení funkce smazány. V případě nutnosti uložení dat z funkce musíme využít globální proměnné (DB, M). Funkce se dají volat několikrát a to z různých míst [16],[17],[19].

### 2.2.2.3 FB bloky

Funkční bloky (FB, Function block) prokazují stejné vlastnosti jako funkce. Oproti FC bloku lze při použití FB bloku uložit hodnotu používaných proměnných do instančních datových bloků. Hodnoty jsou pak k dispozici i po ukončení a opětovném volání bloku. Takové proměnné se pak nazývají statické. Blok je někdy označován jako „blok s pamětí.“ U FB bloků se uživatel setkává s pojmem instance funkčního bloku. Znamená to, že při volání FB bloků z jiné části programu vytvoříme takzvanou instanci funkčního bloku, která má svoji určenou

paměť (konkrétní datový blok). Při dalším volání stejného FB opět vytvoříme instanci. Data této instance jsou ale nezávislá na datech předešlé instance funkčního bloku, protože má svůj vlastní datový blok. To znamená, že každá instance funkčního bloku má vyhrazenou určenou část paměti. Uživatel může ale i zvolit možnost využití paměti pro více instancí v jednom datovém bloku. Tento způsob je označován jako multi-instance. Využití multi-instance lze pouze při programování ve funkčním bloku. Datové bloky pro FB bloky jsou automaticky vytvářeny při vkládání FB bloku do programu. Uživatel může pak zvolit, jestli chce použít datový blok instanční nebo multi-istanční [16],[17],[19].

#### 2.2.2.4 DB bloky

Datové bloky (DB, Data block) slouží pouze k ukládání dat uživatele. Nikoli ke vkládání programových instrukcí. To znamená, že uživatel zde pouze ukládá hodnoty proměnných, které používá v programu. PLC S7 - 1200 rozlišuje konkrétně dva typy datových bloků.

Jedním z nich jsou globální datové bloky, které mohou být použity ve všech ostatních blocích. Z jakéhokoliv jiného bloku (OB, FB, FC), lze tedy číst i zapisovat do globálního DB bloku. Maximální velikost globálního DB bloku je závislá na zvoleném typu CPU. Datovou strukturu v globálním DB bloku si uživatel definuje, jak potřebuje.

Druhým typem DB bloků je instanční datový blok, který je používán pro FB bloky. Instanční datový blok je určený především příslušnému funkčnímu bloku. Dá se k němu přistupovat ale i z ostatních programových bloků. Strukturu určují proměnné deklarované v FB bloku. Ukládají se v něm statická data a parametry FB bloku. Dočasné proměnné použité v FB bloku nejsou v DB bloku ukládány. Stejně jako u globálního datového bloku, tak i u instančního datového bloku je maximální velikost závislá na typu CPU [16],[17],[19].

#### 2.2.3 Datové typy

PLC S7 - 1200 dokáže pracovat s dostatečným počtem datových typů proměnných. Zvládne pracovat například s:

- bitově orientovanými datovými typy (Bool, Byte, Word),
- celočíselnými datovými typy (Int, DInt),
- desetinnými datovými typy (Real),
- časovými datovými typy (Time, Date, Time\_of\_Day),
- datovými typy jako jsou Char a String.

Při práci s jakýmkoliv datovými typy je vhodné nahlédnout do nápovědy, kterou TIA Portal nabízí a nebo použít manuál pro S7 - 1200. Zde najdeme vše potřebné k datovým typům i s příklady použití. Níže jsou uvedeny příklady tabulek zmíněných datových typů z manuálu pro S7 - 1200.

*Tab. 1 Bitově zaměřené datové typy - část první [16]*

Data type	Bit size	Number type	Number range	Constant examples	Address examples
Bool	1	Boolean	FALSE or TRUE	TRUE, 1,	I1.0 Q0.1 M50.7 DB1.DBX2.3 Tag_name
		Binary	0 or 1	0, 2#0	
		Octal	8#0 or 8#1	8#1	
		Hexadecimal	16#0 or 16#1	16#1	
Byte	8	Binary	2#0 to 2#11111111	2#00001111	IB2 MB10 DB1.DBB4 Tag_name
		Unsigned integer	0 to 255	15	
		Octal	8#0 to 8#377	8#17	
		Hexadecimal	B#16#0 to B#16#FF	B#16#F, 16#F	
Word	16	Binary	2#0 to 2#1111111111111111	2#1111000011110000	MW10 DB1.DBW2 Tag_name
		Unsigned integer	0 to 65535	61680	
		Octal	8#0 to 8#177777	8#170360	
		Hexadecimal	W#16#0 to W#16#FFFF, 16#0 to 16#FFFF	W#16#F0F0, 16#F0F0	
DWord	32	Binary	2#0 to 2#11111111111111111111111111111111	2#111100001111111100 001111	MD10 DB1.DBD8 Tag_name
		Unsigned integer	0 to 4294967295	15793935	

*Tab. 2 Bitově zaměřené datové typy - část druhá [16]*

Data type	Bit size	Number type	Number range	Constant examples	Address examples
		Octal	8#0 to 8#3777777777	8#74177417	
		Hexadecimal	DW#16#0000_0000 to DW#16#FFFF_FFFF, 16#0000_0000 to 16#FFFF_FFFF	DW#16#F0FF0F, 16#F0FF0F	

*Tab. 3 Celočíselné datové typy [16]*

Data type	Bit size	Number Range	Constant examples	Address examples
USInt	8	0 to 255	78, 2#01001110	MB0, DB1.DBB4, Tag_name
SInt	8	-128 to 127	+50, 16#50	
UInt	16	0 to 65,535	65295, 0	MW2, DB1.DBW2, Tag_name
Int	16	-32,768 to 32,767	30000, +30000	
UDInt	32	0 to 4,294,967,295	4042322160	MD6, DB1.DBD8, Tag_name
DInt	32	-2,147,483,648 to 2,147,483,647	-2131754992	



Tab. 4 Datové typy pro desetinné operace [16]

Data type	Bit size	Number range	Constant Examples	Address examples
Real	32	-3.402823e+38 to -1.175 495e-38, ±0, +1.175 495e-38 to +3.402823e+38	123.456, -3.4, 1.0e-5	MD100, DB1.DBD8, Tag_name
LReal	64	-1.7976931348623158e+308 to -2.2250738585072014e-308, ±0, +2.2250738585072014e-308 to +1.7976931348623158e+308	12345.123456789e40, 1.2E+40	DB_name.var_name Rules: <ul style="list-style-type: none"> <li>No direct addressing support</li> <li>Can be assigned in an OB, FB, or FC block interface table</li> </ul>

Tab. 5 Časové datové typy [16]

Data type	Size	Range	Constant Entry Examples
Time	32 bits	T#-24d_20h_31m_23s_648ms to T#24d_20h_31m_23s_647ms Stored as: -2,147,483,648 ms to +2,147,483,647 ms	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms 10d20h30m20s630ms
Date	16 bits	D#1990-1-1 to D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31
Time_of_Day	32 bits	TOD#0:0:0.0 to TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (Date and Time Long)	12 bytes	Min.: DTL#1970-01-01-00:00:00.0 Max.: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

Tab. 6 Datové typy Char a String [16]

Data type	Size	Range	Constant Entry Examples
Char	8 bits	ASCII character codes: 16#00 to 16#FF	'A', 't', '@'
String	n+ 2 bytes	n = (0 to 254 character bytes)	'ABC'

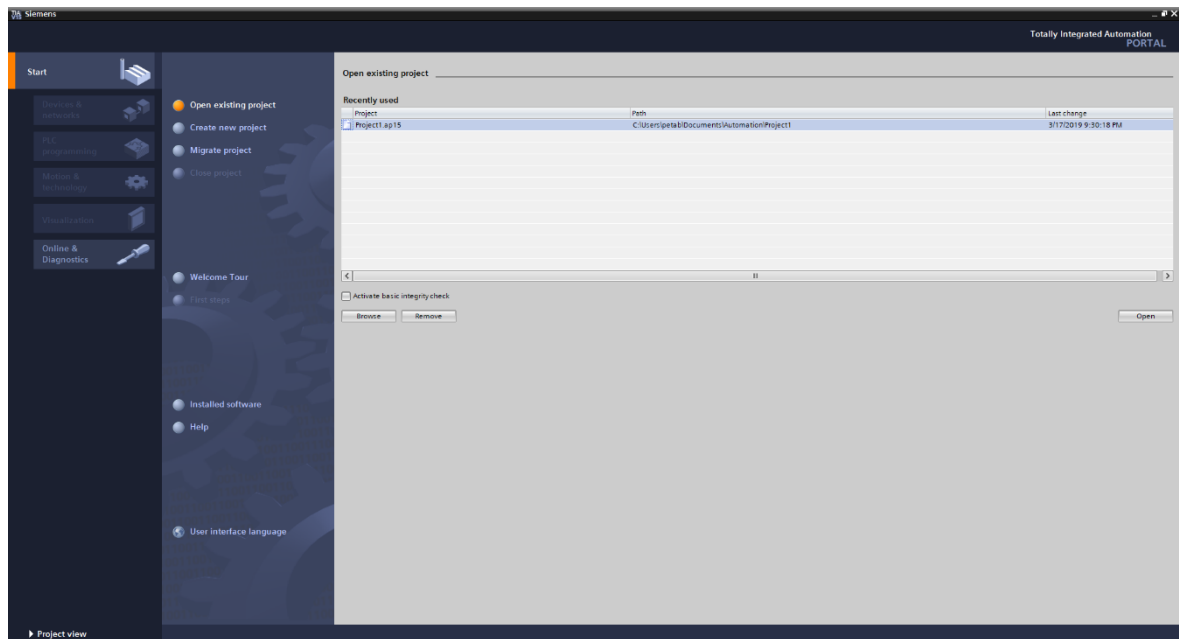
### 2.2.4 WinCC Basic

Součástí Step 7 Basic je balíček WinCC Basic pro projektování panelů HMI od společnosti Siemens a to pouze pro panely řady Basic. Pro projektování panelů vyšší řady Comfort je potřeba mít verzi WinCC Comfort [17].

Student si bude moci vyzkoušet projektování HMI panelu KTP700 Basic od společnosti Siemens a to pomocí WinCC Basic v úloze, která je navržena v praktické části této bakalářské práce.

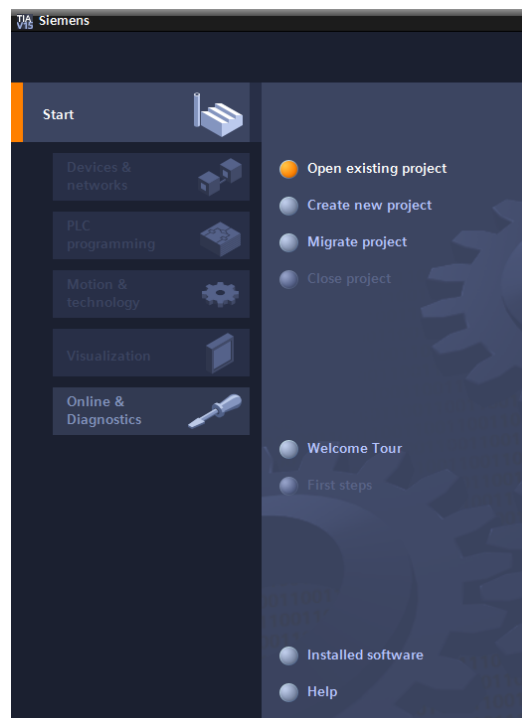
## 2.3 Vytvoření projektu v TIA Portálu

TIA Portal po spuštění zobrazí takzvané Portálové zobrazení – Portal View.



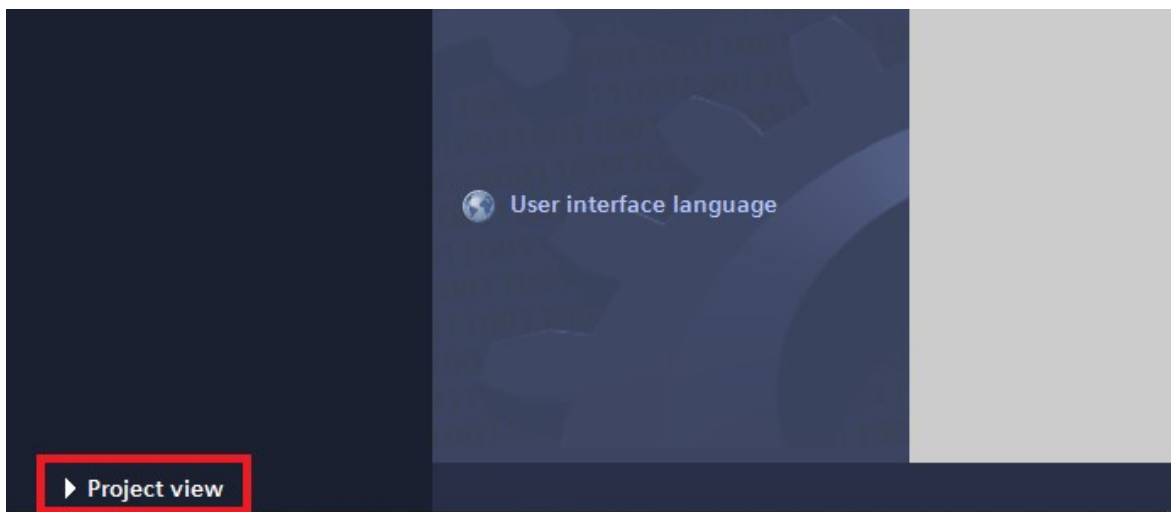
Obr. 10 Portálové zobrazení – Portal View v TIA Portalu

Zde je možné učinit první kroky pro vytvoření projektu, nebo pokud již projekt máme vytvořený, tak je zde možnost požadovaný projekt otevřít. Uživatel v Portálovém zobrazení také může učinit první kroky potřebné k migraci projektu, který byl vytvořený v předešlých verzích TIA Portalu.



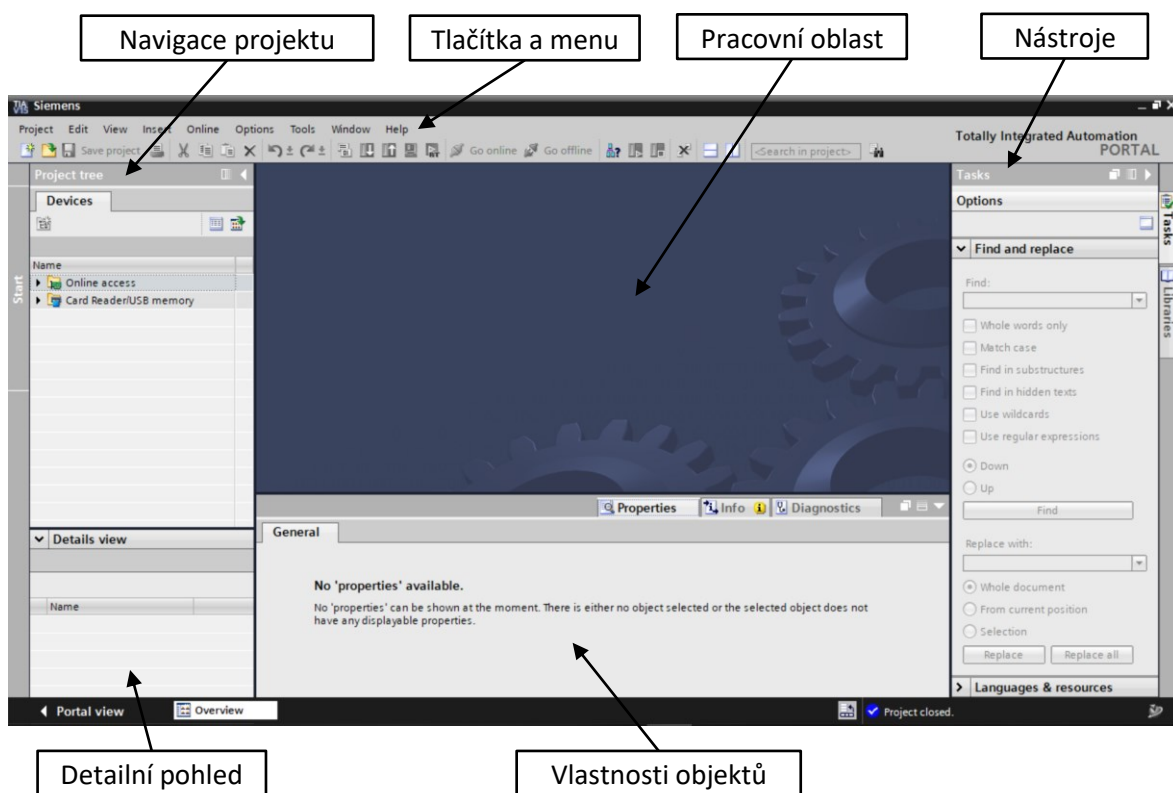
Obr. 11 Detail na možnosti v Portálovém zobrazení

Druhou možností úvodního zobrazení je Projektové zobrazení – Project View. Na to lze přepnout ikonou zobrazenou vlevo dole v Portálovém zobrazení.



Obr. 12 Tlačítko na přepnutí do Project view

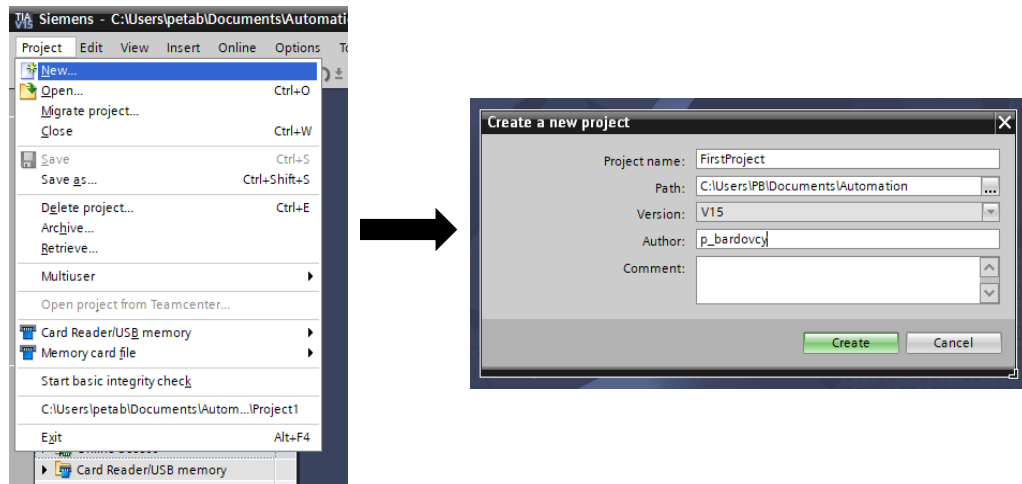
V Projektovém zobrazení se dají provést stejné kroky jako v Portálovém zobrazení. Takže i zde uživatel dokáže provést vytvoření, otevření nebo migraci projektu. V tomto zobrazení se navíc většinou uživatel pohybuje při správě projektu, a to především z důvodu větší přehlednosti.



Obr. 13 Projektové zobrazení – Project view

### 2.3.1 Vytvoření projektu

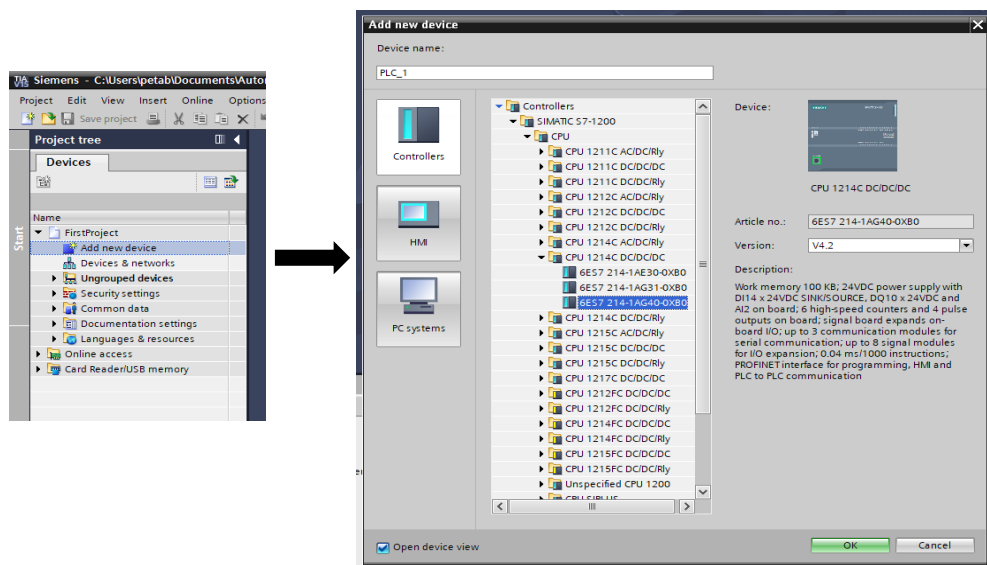
Pro vytvoření nového projektu zvolíme v menu „Project“ možnost „New...“ a po zobrazení okna „Create a new project“ vyplníme název projektu, cestu pro uložení, případně autora a komentář k projektu. Pak klikneme na tlačítko „Create.“



Obr. 14 Vytvoření projektu

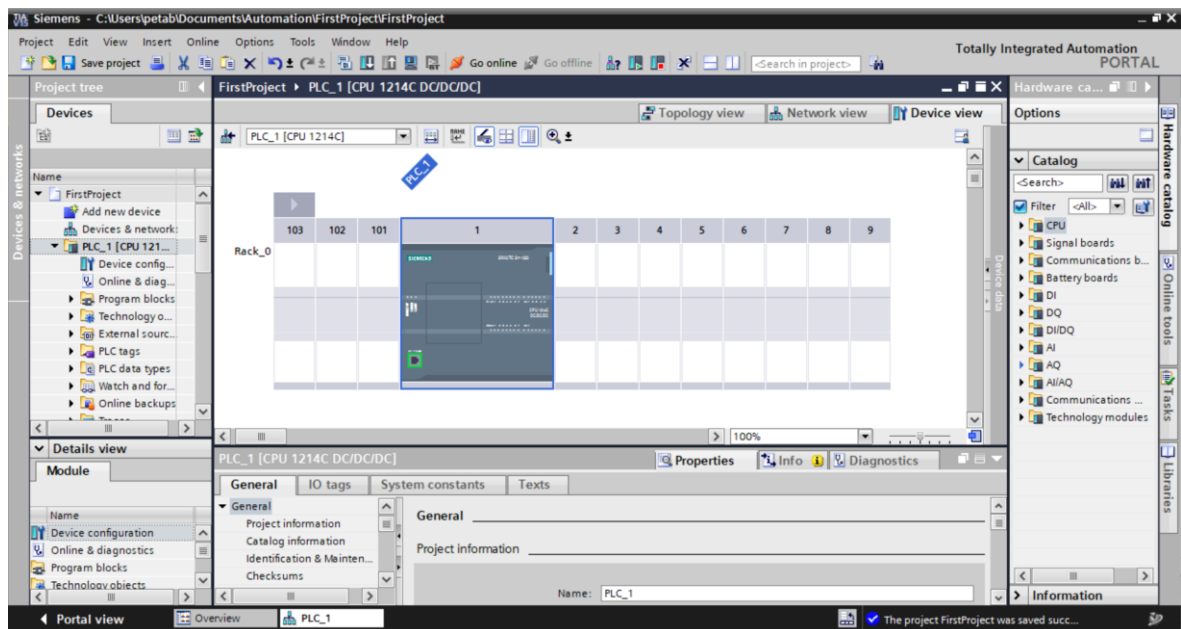
### 2.3.2 Konfigurace HW

Nyní je potřeba přidat do vytvořeného projektu hardware, který chceme použít. V tomto případě se bude jednat o PLC S7 - 1200 a to konkrétně PLC s CPU 1214C DC/DC/DC. V „Project tree“ (v navigaci projektu) zvolíme „Add new device“ a z katalogu vybereme správný typ PLC. Označení se dá nalézt přímo na PLC S7 - 1214C.



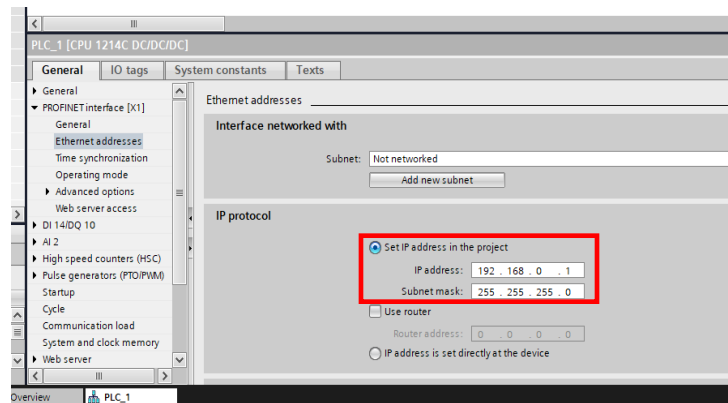
Obr. 15 Přidání PLC S7 - 1200 s CPU 1214C DC/DC/DC z katalogového listu

Po označení konkrétního typu PLC, může uživatel vidět na pravé straně okna s katalogovým listem popis daného PLC. V tomto případě zde jsou vlastnosti PLC S7 - 1200 s CPU 1214C popisované v předešlých bodech této bakalářské práce. Pro pokračování zvolíme naše konkrétní PLC a potvrdíme „OK.“ Zobrazí se nám okno konfigurace hardwaru.



Obr. 16 Okno konfigurace hardwaru

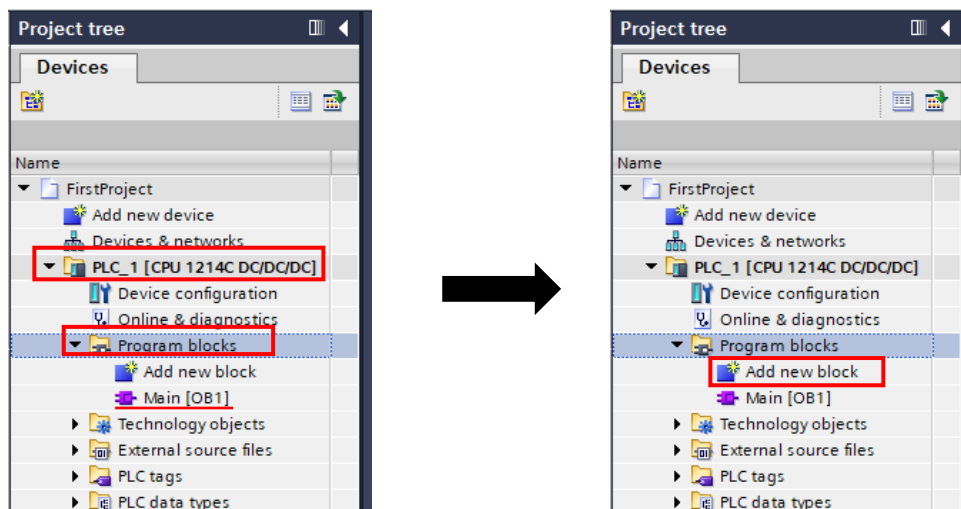
Jedním z prvních kroků, abychom se k PLC dokázali připojit, by mělo být nastavení IP adresy v nastavení konfigurace. IP adresa se nastavuje ve vlastnostech (Properties) PLC v záložce „General“ v kategorii „PROFINET interface [X1].“ Zde je ještě potřeba zvolit podkategorii „Ethernet addresses.“ Adresu volíme dle rozsahu a dostupnosti adres v dané síti, kde PLC ale i používaný počítač s TIA Portalem bude připojené. Pokud adresu v PLC nastavujeme poprvé, tak je nutné být k PLC připojené napřímo síťovým kabelem z počítače a nastavit IP adresu počítače tak, aby nekorespondovala s adresou 192.168.0.1, což bývá právě defaultní adresa PLC, a zároveň aby byla ve stejném rozsahu adres. V případě, kdy je adresa jiná než defaultní, tak lze tuto adresu zjistit nebo změnit pomocí diagnostických funkcí v TIA Portálu.



Obr. 17 Nastavení IP adresy

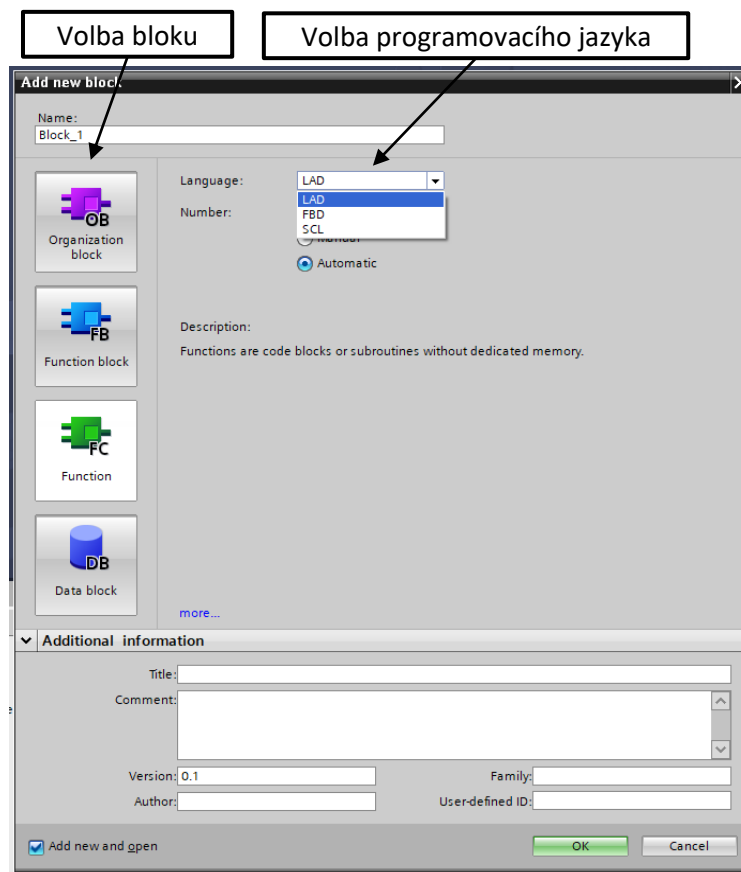
### 2.3.3 Vložení programového bloku

Vkládání programového bloku se provádí v „Project tree.“ Nejdříve je potřeba rozkliknout záložku s názvem PLC, pak záložku „Program block“ Zde je vidět, že TIA Portal automaticky vygeneroval již blok OB1, který slouží k cyklickému provádění projektu. Pro přidání dvakrát klikneme na „Add new block.“



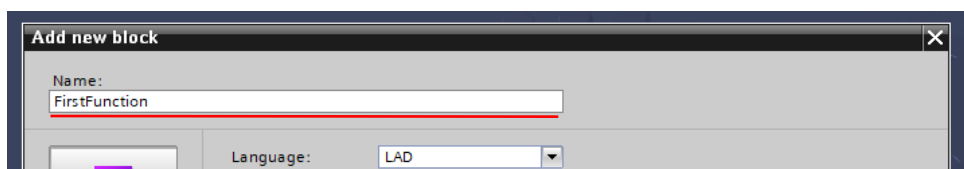
Obr. 18 Přidání programového bloku

Otevře se nabídka, kde je potřeba zvolit jaký blok chceme přidat (OB, FB, FC, DB). Jelikož OB1 je již vytvořený, tak pro ukázkou zvolíme například blok FC, který následně bude muset být umístěn v OB1, aby byl programem volán a prováděn. Kromě volby bloku, je zde možné zvolit, jaký programovací jazyk chceme pro blok použít. Pro ukázkou zvolíme programovací jazyk LAD (jazyk kontaktních schémat).



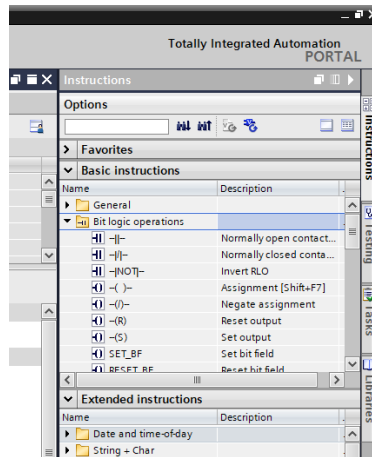
Obr. 19 Volba bloku a programovacího jazyka

Pro úplnost blok pojmenujeme například „FirstFunction“ a vše potvrdíme tlačítkem „OK.“



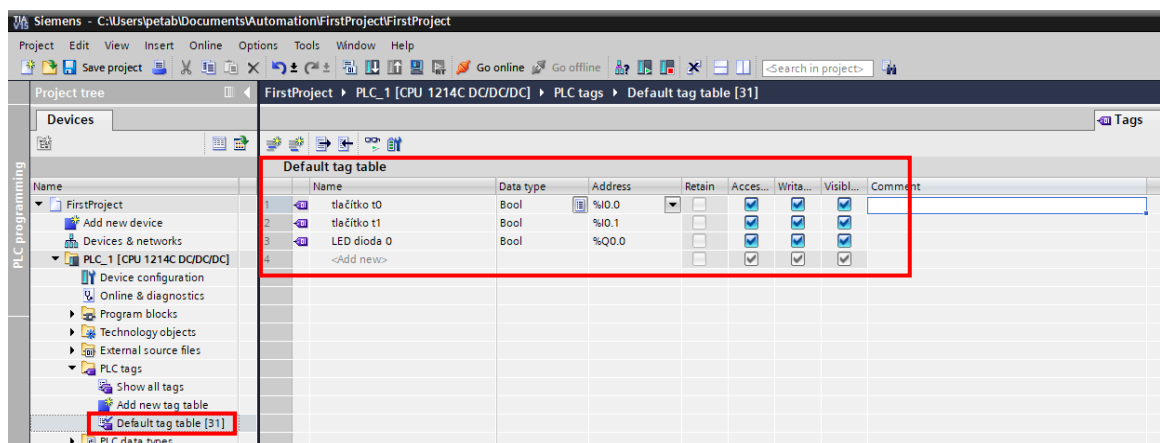
Obr. 20 Pojmenování bloku

Pokud se blok neotevřel po přidání bloku, tak můžeme blok otevřít dvojklikem na jeho ikonu v „Project Block,“ kde je umístěn. Po otevření bloku můžeme začít vytvářet program. K tomu lze používat programové instrukce, které jsou umístěny v okně „Instructions.“ Najdeme zde instrukce od těch nezákladnější až po ty nejsložitější. K předvedení jak vytvářet program použijeme bitové logické operace (Bit logic operation).



Obr. 21 Instrukce pro vytváření programu

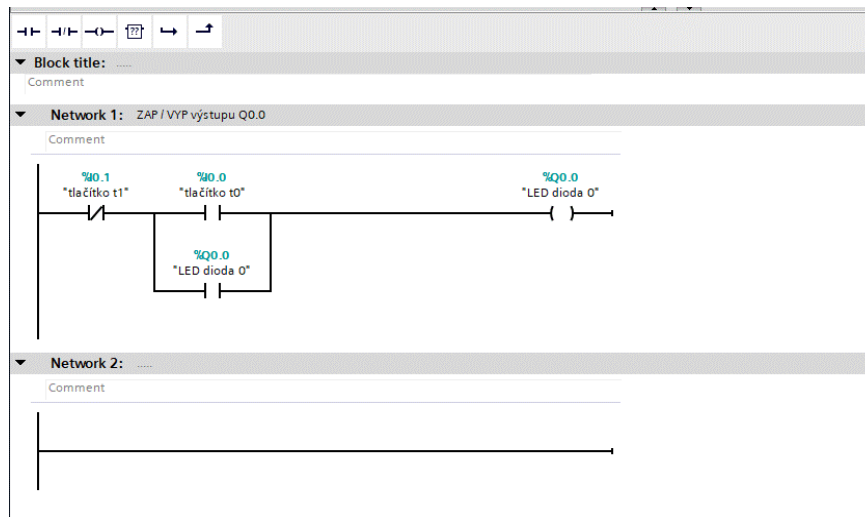
Pomocí bitových operací vytvoříme program, který pomocí dvou spínačů (tlačítko t0 a tlačítko t1) připojených na vstupy I0.0 a I0.1 bude zapínat LED diodu umístěnou u výstupu Q0.0 pro signalizaci aktivace fyzického výstupu. Tlačítko t0 bude tento vstup aktivovat a tlačítko t1 deaktivovat. V moderním pojetí programování PLC se více setkáme se symbolickým pojmenováním místo přímé adresace vstupů a výstupů. To znamená, že například vstupům a výstupům přiřadíme názvy, které pak použijeme v programu. V TIA Portalu jsou tyto názvy označovány jako „PLC tag.“ Toto přiřazení se provádí v záložce „PLC tags,“ která se nachází v „Project tree.“ Pro přiřazení můžeme použít „Default tag table,“ kde k absolutním adresám vstupů „%I0.0,“ „%I0.1“ a výstupu „%Q0.0“ přiřadíme symbolické názvy „tlačítko t0“, „tlačítko t1“ a pro výstup „LED dioda 0.“



Obr. 22 Přiřazení symbolických adres

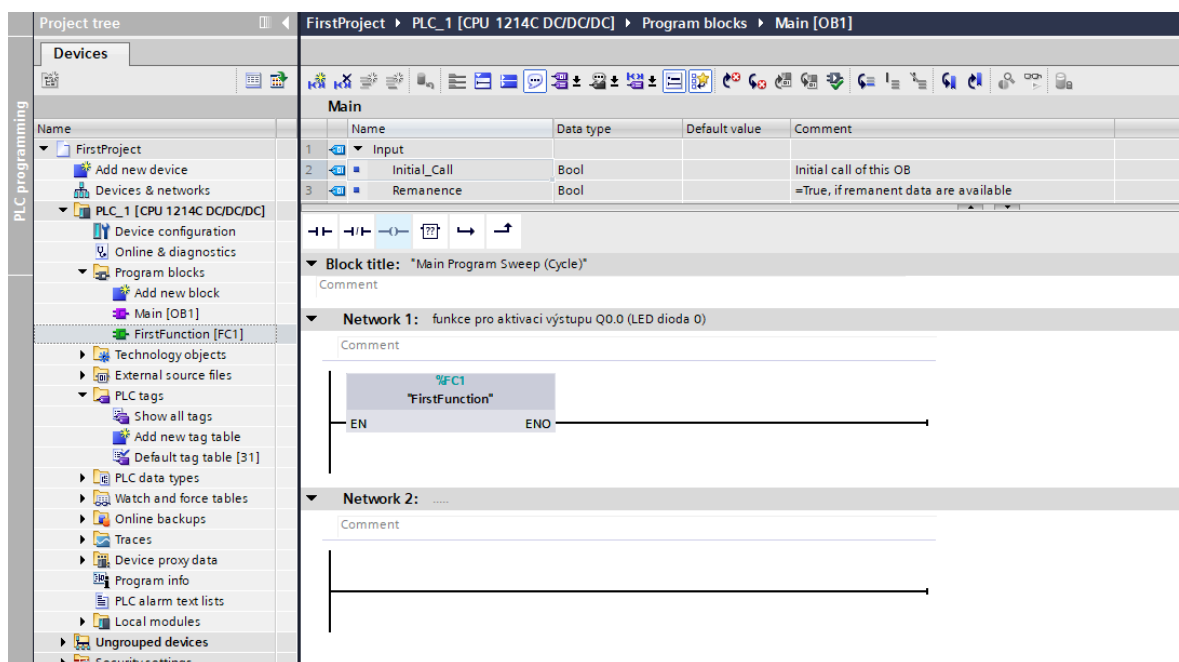
Nyní vytvoříme program pomocí logických bitových operací. Program se skládá z Normally Open prvku (NO), Normally Closed (NC) prvku a z prvku pro aktivaci výstupu - „cívky.“





Obr. 23 Příklad programu v LAD

V dalším kroku je potřeba funkci umístit do OB1 pro volání funkce. To znamená, že otevřeme blok OB1 a například pomocí funkce „drag and drop“ přetáhneme funkci ze záložky „Project block“ do Networku 1 v OB1.

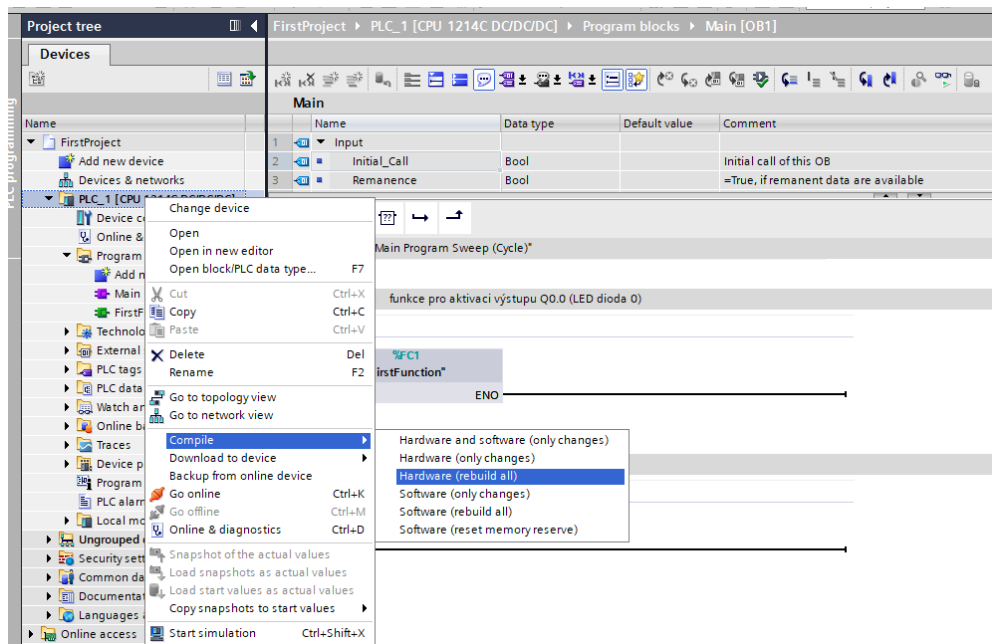


Obr. 24 Volání funkce z OB1

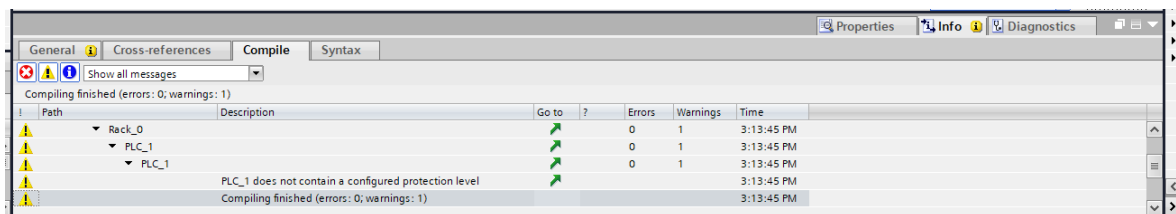
### 2.3.4 Kompilace projektu

Kompilace projektu provádíme, abychom zjistili, jestli jsme neprovedli systémovou chybu v konfiguraci nebo při vytváření programu. Proto musíme provést kompilaci hardwaru i softwaru. Oboje se provádí v menu, které se zobrazí při kliknutí pravým tlačítkem na ikonu PLC

v „Project tree.“ Zde zvolíme možnost „Compile“ a nejdříve vybereme „Hardware (rebuild all)“ a pak „Software (rebuild all).“ Pokud budeme posléze provádět kompilace úprav, tak volíme možnost „Hardware (only changes)“ nebo „Software (only changes).“ Volíme podle toho, jestli jsme prováděli změny v konfiguraci nebo v programu. O úspěšné nebo neúspěšné kompilaci budeme informováni v okně „Info“ v záložce „Compile.“ V případě, že zapomeneme kompilaci provést, tak bude provedena před nahráním projektu do PLC.



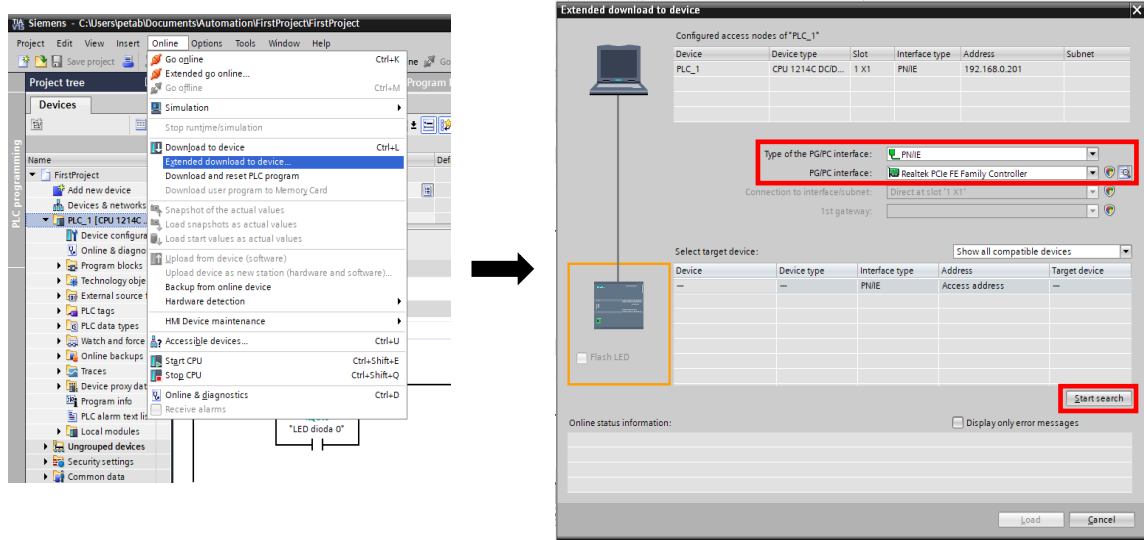
Obr. 25 Kompilace hardwaru PLC



Obr. 26 Informace o úspěšném zkompilování hardwarové konfigurace

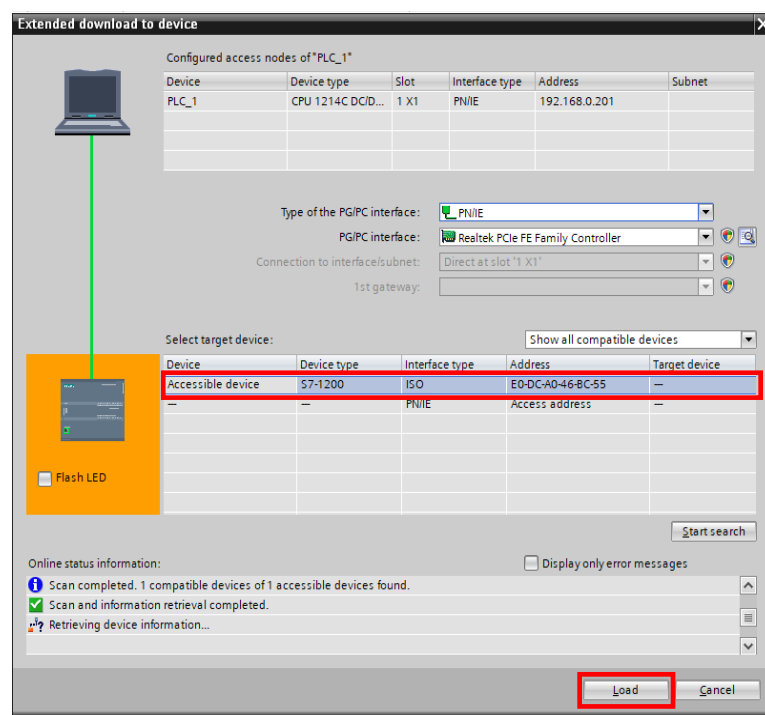
### 2.3.5 Nahrání projektu do PLC S7 - 1200

Prvním krokem pro nahrání projektu do PLC je označit si PLC v „Project tree“ a v horní liště zvolit v záložce „Online“ volbu „Extended download to device.“ Zobrazí se nabídka, kde je možné zvolit, jak se chceme k PLC připojit.



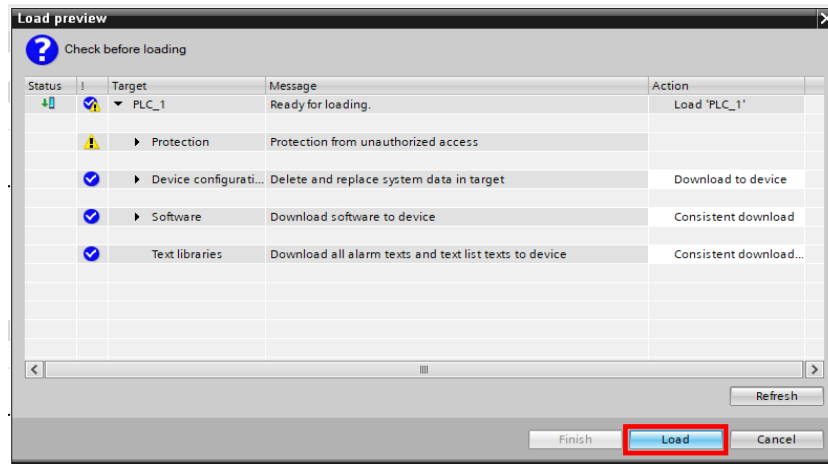
Obr. 27 Nastavení komunikace pro nahrání projektu do PLC

V nastavení komunikace zvolíme typ komunikace a rozhraní počítače, s kterým se budeme připojovat. Pokud se k PLC připojujeme síťovým kabelem napřímo z počítače, tak zvolíme jako typ komunikace (Type of the PG/PC interface) volbu „PN/IE,“ která označuje připojení pomocí Profinetu nebo Ethernetu. Jako rozhraní počítače zvolíme síťovou kartu počítače. Ve chvíli, kdy je nastavena komunikace, tak klikneme na tlačítko „Start search“ a TIA Portal vyhledá dostupná zařízení dle zvolené komunikace. Až dokončí hledání, tak zvolíme zařízení, do kterého chceme projekt nahrát a klikneme na „Load.“



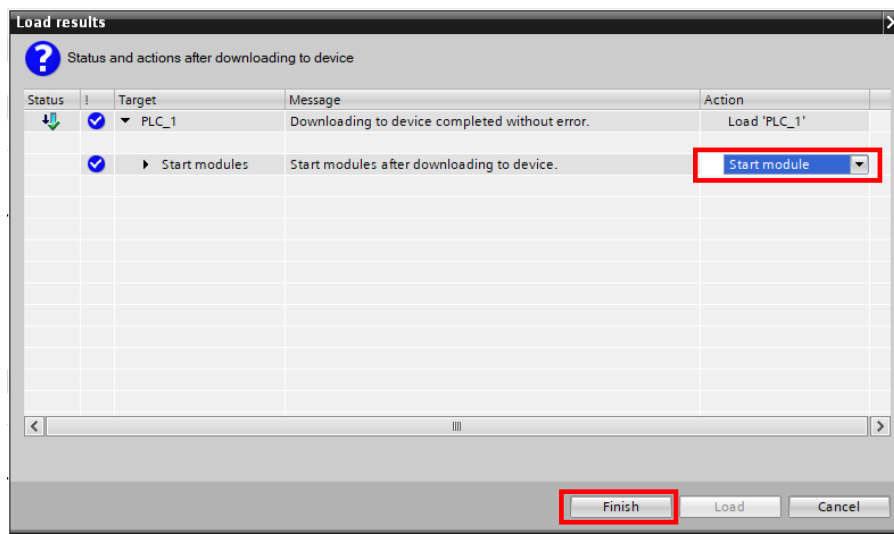
Obr. 28 Nahrání projektu do zvoleného zařízení

V dalším okně opět klikneme na „Load.“ Potvrdíme tím, že chceme nahrát konfiguraci a program do PLC.



Obr. 29 Potvrzení nahrání konfigurace a programu do PLC

Po zobrazení další nabídky zvolíme ve sloupci „Action“ možnost „Start module“ a potvrdíme stisknutím „Finish.“ Tím docílíme přepnutí PLC do režimu „RUN,“ kdy zpracovává uživatelský program.

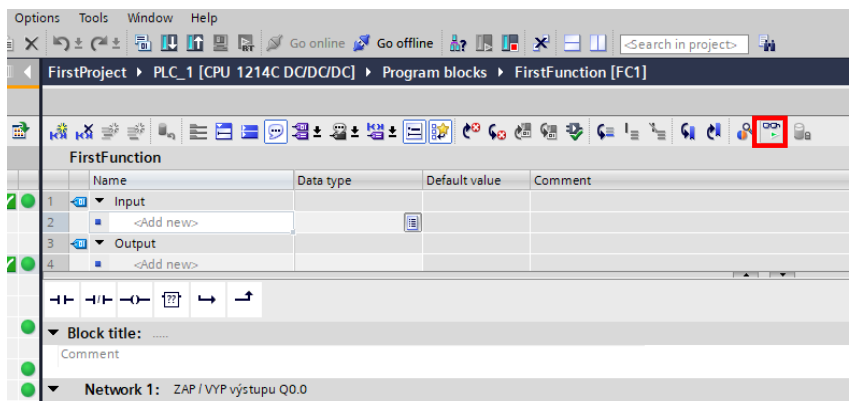


Obr. 30 Přepnutí režimu PLC do režimu „RUN“ po nahrání projektu

### 2.3.6 Online testování projektu

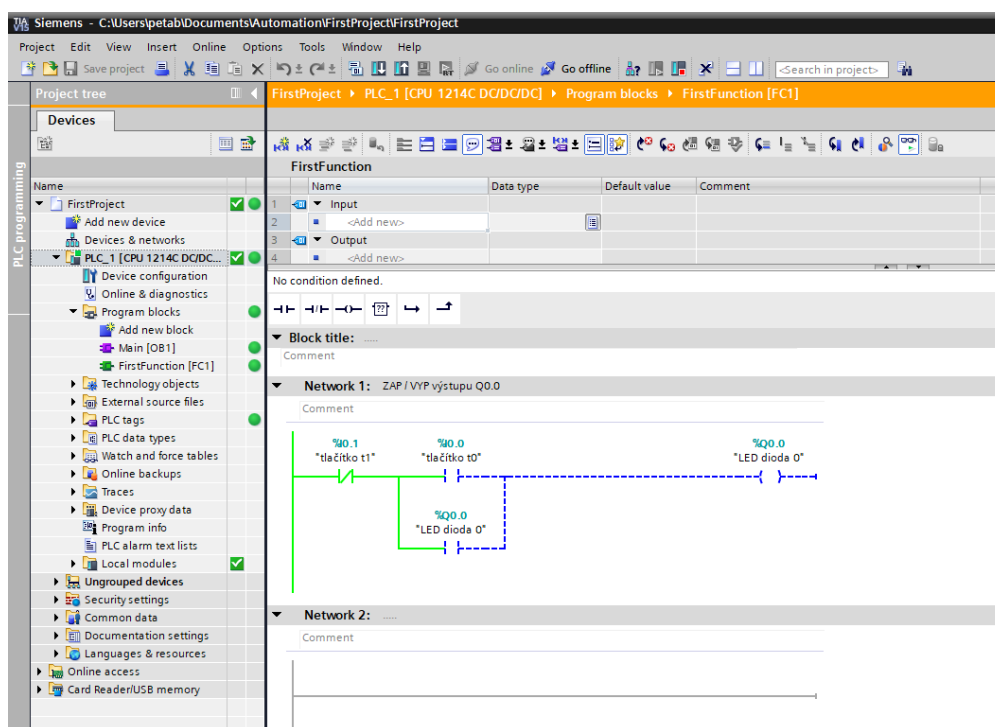
K testování a ověření funkčnosti programu se využívá online režim. Uživatel pak může vidět, jak je uživatelský program zpracováván a zároveň ověřit zda se offline verze projektu (konfigurace a program) shodují s projektem nahráním v PLC. K přepnutí do online režimu je potřeba označit PLC v „Project tree“ a kliknout na ikonu „Go online.“ Poté je možné

detektovat, které části projektu se shodují s tím, co je reálně nahané v PLC a ověřit funkčnost vytvořeného programu. Online monitorování uživatelského programu lze zobrazit po kliknutí na ikonu „Monitoring on/off“ v nástrojové liště daného bloku.



Obr. 31 Ikona „Monitoring on/off“ pro přepnutí do monitorovacího režimu

Uživatel pak může vidět v jakém stavu se program v PLC reálně nachází a vidět například aktivní a neaktivní části programu.



Obr. 32 Ukázka monitorovacího režimu FC bloku

## **II. PRAKTICKÁ ČÁST**

### 3 PŘÍKLADY PRO PROCVIČENÍ PROGRAMOVÁNÍ PLC SIMATIC S7 - 1200

Praktická část bakalářské práce je zaměřena na vytvoření vzorových příkladů pro studenty, kteří si v předmětu Mikropočítače a PLC budou tak moct vyzkoušet pracovat s programovatelným automatem řady S7 - 1200 od společnosti Siemens. Příklady mají demonstrovat možnosti a způsoby programování PLC Simatic S7 - 1200 v prostředí TIA Portalu zaměřených na komerční bezpečnost.

#### 3.1 Logické operace v různých programovacích jazycích

Následující úloha má za úkol demonstrovat používání logických operací (AND a OR) v různých programovacích jazycích (LAD, FBD, SCL) a zároveň možnost si „osahat“ práci v TIA Portalu.

##### 3.1.1 Zadání

Vytvořte úlohu dle následujícího zadání:

- a) Vytvořte projekt v TIA Portalu podle postupu uvedeného v teoretické části bakalářské práce a nakonfigurujte PLC S7 - 1214C DC/DC/DC. Nastavte IP adresu PLC a vyzkoušejte propojení PLC s PC (nahrajte konfiguraci do PLC). Připravený nakonfigurovaný projekt si uložte jako výchozí projekt pro další úlohy.
- b) Následně uvažujte o domě, který má 3 patra. V každém patře jsou 3 okna a jedny dveře (celkem čtyři detektory na jedno patro), kdy jejich detektory (pro zjištění stavu otevřeno/zavřeno) jsou v klidovém stavu stále sepnuty (kontakt NC). V tomto případě jsou okna nebo dveře zavřeny, a pokud dveře nebo okno někdo otevře, tak je detektor ve stavu rozepnut (vstup pro detektor je aktivní). Pokud po zastřežení někdo otevře okno nebo dveře, tak se spustí alarm v celém domě a svítí kontrolka (výstup) pro konkrétní patro, kde nastal poplach. Navrhněte pro tento případ program a to ve třech programovacích jazycích (LAD, FBD, SCL). To znamená, vytvořte tři funkce (FC), kdy jedna bude naprogramována pomocí LAD, druhá funkce pomocí FBD a třetí pomocí SCL.
- c) Doplněte možnost kvitování vyhlášeného poplachu ve všech třech FC blocích.
- d) Volitelně doplněte možnost kvitování během stavu nezastřeženo pro FC blok naprogramovaný pomocí SCL.

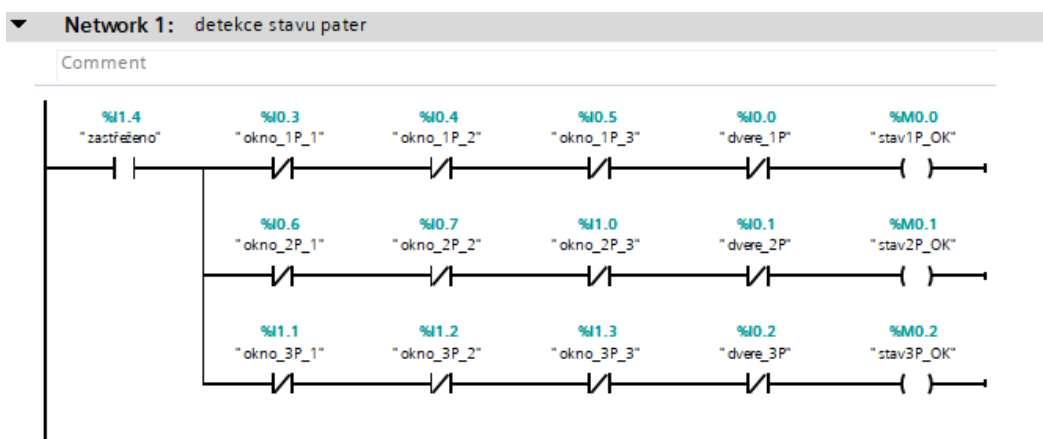
Pozn.: Nejdříve si nadefinujte symbolické adresy všech potřebných vstupních a výstupních signálů spolu s případnými paměťovými bity v „Default tag table.“ Nezapomeňte volat funkci v OB1 a nevolejte funkce zároveň, aby se navzájem neovlivňovali. Projekt nahrajte do PLC a odzkoušejte funkčnost programu dle zadání.

Default tag table							
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	dvere_1P	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	dvere_2P	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	dvere_3P	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	okno_1P_1	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	okno_1P_2	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	okno_1P_3	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	okno_2P_1	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	okno_2P_2	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	okno_2P_3	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	okno_3P_1	Bool	%I1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	okno_3P_2	Bool	%I1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	okno_3P_3	Bool	%I1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	zastřeženo	Bool	%I1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	kvitace poplachu	Bool	%I1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	alarm_1P	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	alarm_2P	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	alarm_3P	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	vyhlaseeni_poplachu	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	stav1P_OK	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	stav2P_OK	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	stav3P_OK	Bool	%M0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Obr. 33 Příklad nadeřinování symbolických adres

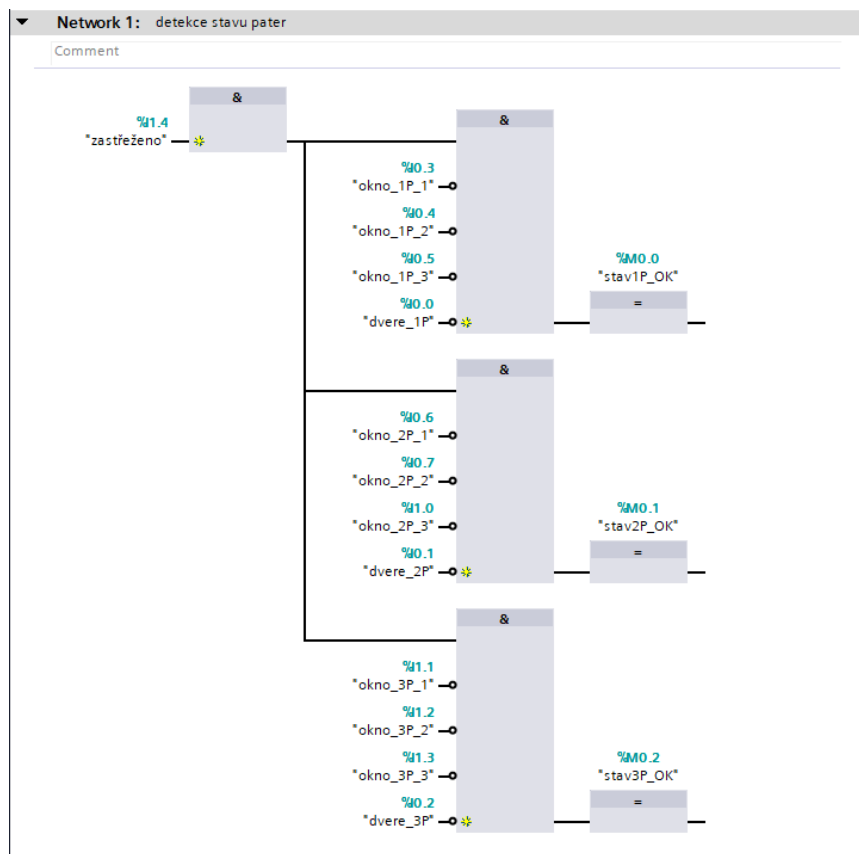
### 3.1.2 Řešení

Viz. příloha této bakalářské práce.



Obr. 34 Ukázka řešení detekce stavu jednotlivých pater v LAD





Obr. 35 Ukázka řešení detekce stavu jednotlivých pater v FBD

```

3
4 //kontrola stavu 1. patra
5 IF NOT "okno_1P_1" AND NOT "okno_1P_2" AND NOT "okno_1P_3" AND NOT "dvere_1P" THEN
6   "stav1P_OK" := TRUE;
7 ELSE
8   "stav1P_OK" := FALSE;
9 END_IF;
10
11 //kontrola stavu 2. patra
12 IF NOT "okno_2P_1" AND NOT "okno_2P_2" AND NOT "okno_2P_3" AND NOT "dvere_2P" THEN
13   "stav2P_OK" := TRUE;
14 ELSE
15   "stav2P_OK" := FALSE;
16 END_IF;
17
18 //kontrola stavu 3. patra
19 IF NOT "okno_3P_1" AND NOT "okno_3P_2" AND NOT "okno_3P_3" AND NOT "dvere_3P" THEN
20   "stav3P_OK" := TRUE;
21 ELSE
22   "stav3P_OK" := FALSE;
23 END_IF;
24
25 ..

```

Obr. 36 Ukázka řešení detekce stavu jednotlivých pater v SCL

### 3.1.3 Komentář k řešení

Základem k řešení úlohy je si nadefinovat symbolické adresy příslušných vstupních a výstupních signálů a potřebných paměťových bitů pro stav jednotlivých pater pro následné vyhodnocení. V FC bloku naprogramovaného pomocí LAD jsou použity pouze tři druhy

bitově zaměřených operací a to kontakty NO (normály open, spínací kontakt) a NC (normaly closed, rozpínací kontakt) a „Assignment“ prvek (cívka) pro aktivaci bitu. Program využívá sériové (AND) a paralelní (OR) seřazení prvků. Pomocí kombinací těchto prvků je sestaven celý program. Popis jednotlivých Networků (sítí) je uveden v popisu programu.

V bloku naprogramovaného pomocí FBD, byly použity bloky AND a OR pro sestavení podmínek. Tam kde bylo potřeba použít negaci signálu při vstupu do bloku, tak byla použita funkce „invert RLO,“ která se nachází v nástrojové liště pro vytváření programu. Pro aktivaci konkrétního bitu je použit blok „Assigment.“

V bloku programovaného pomocí SCL je použita funkce „IF,“ pomocí které jsou kontrolovány podmínky potřebné pro spuštění alarmu. Syntaxi funkce můžeme zjistit z nápovědy TIA Portalu. To platí i o tom, jak používat veškeré funkce, programové prvky a bloky.

V bloku programovaném pomocí SCL v „ELSE“ části funkce „IF“ je přidána možnost kvitace v případě, kdy je dům ve stavu nezastřeženo.

## 3.2 Použití časovačů a čítačů

V úloze s čítači a časovači si student vyzkouší jak s těmito funkcemi zacházet a jak je lze použít.

### 3.2.1 Zadání

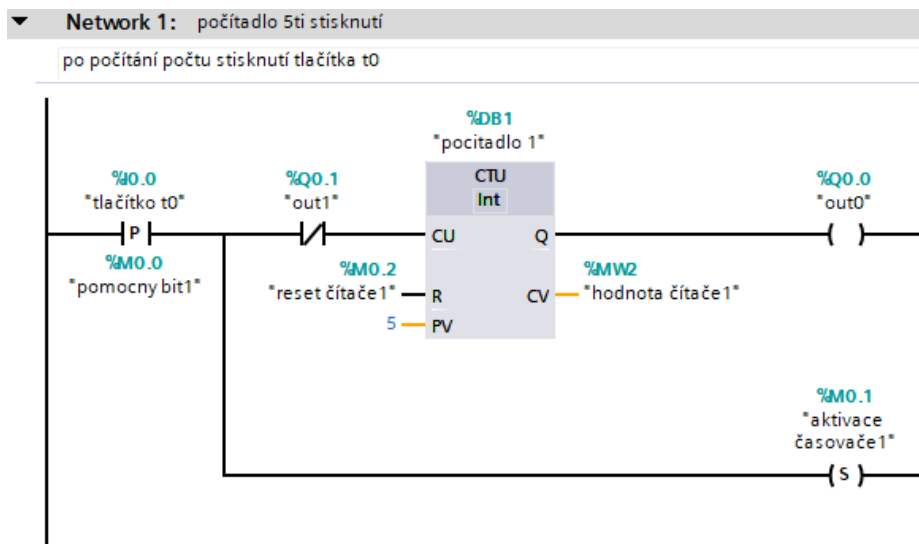
Vytvořte úlohu dle následujícího zadání:

- a) Realizujte program v TIA Portalu za pomoci čítačů a časovačů, kdy po přesně pěti stisknutích tlačítka  $t0$  (připojeného ke vstupu I0.0) aktivujete výstup  $out0$  (Q0.0). Na aktivaci má uživatel pouze 5 sekund od prvního stisknutí. Výstup  $out1$  (Q0.1) bude signalizovat překročení času (5 sekund). Pro opětovný pokus bude potřeba stisknout tlačítko  $t1$  (reset), který deaktivuje výstupy  $out0$  a  $out1$  a zresetuje čítače a časovače. Úlohu realizujte pomocí čítačů a časovačů v TIA Portalu.
- b) Pokud uživatel stihne během pěti sekund stisknout přesně pětkrát tlačítko  $t0$ , tak na šesté stisknutí v časovém limitu bude vše uvedeno do počátečních podmínek (vše zhasnuto a zresetovány čítače a časovače). Program vytvořte ve dvou programovacích jazycích a to v LAD a FBD. To znamená vytvořit dvě funkce (FC), kdy jedna bude naprogramovaná pomocí LAD a druhá funkce pomocí FBD.

Pozn.: Nezapomeňte volat funkci v OB1 a nevolejte funkce zároveň, aby se navzájem neovlivňovali. Projekt nahrajte do PLC a odzkoušejte funkčnost programu dle zadání.

### 3.2.2 Řešení

Viz. příloha této bakalářské práce.



Obr. 37 Ukázka použití čítače CTU v řešení úlohy

### 3.2.3 Komentář k řešení

Prvním krokem je definování symbolických adres vstupů, výstupů a potřebných paměťových bitů. Navíc je potřeba ukládat hodnotu čítače. Hodnota se ukládá v proměnné (Tagu) „*hodnota čítače1*“ pro FC blok programovaný pomocí LAD. Pro FC blok programovaný pomocí FBD se hodnota ukládá do proměnné „*hodnota čítače2*.“

V úloze je použit programový prvek pro detekci náběžné hrany „*tlačítka t0*.“ Tento prvek vždy potřebuje mít paměť a je potřeba tedy mu přiřadit příslušný bit. V ukázce řešení je to „*pomocný bit1*“ pro FC blok programovaný v LAD a „*pomocný bit2*“ pro FC blok programovaný v FBD. V obou blocích se pak využívá možnosti přiřazení logické nuly (Reset output) nebo logické jedničky (Set output) příslušnému bitu. Při použití těchto logických prvků je v bitu zachována příslušná hodnota a to i ve chvíli, kdy příslušná větev (Network) není aktivní.

Počet stisknutí čítá programový blok CTU, který čítá směrem nahoru (přičítá ke stávající hodnotě). Čítač CTU musí mít přiřazený příslušný datový blok (DB). Svůj výstup aktivuje ve chvíli, kdy počet stisknutí je roven pěti. Hodnota, při které se aktivuje výstup čítače, se

nastavuje jako vstupní parametr bloku CTU. Název vstupního parametru je „PV.“ Vstup „R“ je pro reset hodnoty čítače. Výstup „CV“ se používá pro zápis aktuální hodnoty čítače. Jak přesně čítač funguje, je možné si prohlédnout v nápovědě TIA Portalu.

Pro čítání času je použit časovač TON, který aktivuje svůj výstup po dosažení nastaveného času. Zároveň však musí být stále aktivní vstup „IN.“ Čas se nastavuje jako vstupní parametr „PT.“ A musí být zapsán ve formátu „T#5s“ Jak přesně časovač funguje, je možné si prohlédnout v nápovědě TIA Portalu.

Reset je proveden dvěma způsoby. Jedním způsobem je tlačítko „reset tl.“ Druhým je, pokud hodnota čítače je rovna šesti nebo větší. K tomu slouží funkce „Less or equal“ („je menší nebo rovno“). Lze případně použít i funkci „Greater or equal“ („je větší nebo rovno“). Hodnoty by v tu chvíli museli být porovnávány v opačném pořadí.

Při práci s časovači a čítači je vhodné se nejprve podívat na nápovědu konkrétního čítače a časovače. V nápovědě lze totiž zjistit jak daný čítač nebo časovač pracuje. Stejně tvrzení se dá říct o všech nových funkcích, prvcích a blocích, se kterými se programátor setká.

Obě vytvořené funkce jsou prakticky totožné. Rozdíly lze najít pouze v grafickém vyobrazení. K tomu nasvědčuje i možnost změny programovacího jazyku ve vlastnostech bloku a to i pro hotový blok. Automaticky tak dojde k převedení bloku buď z LAD do FBD a nebo opačně. Pro SCL toto možné není.

### 3.3 Blokové parametry u FC a FB bloků

Následující úloha slouží především k procvičení užívání parametrů bloků, které se používají pro FC a FB bloky a uvědomění si rozdílů mezi těmito bloky. Zároveň je v úloze zahrnuta možnost využití funkce pro práci s proměnou typu STRING a s časovými datovými typy.

#### 3.3.1 Zadání

Vytvořte program v programovacím jazyce LAD nebo FBD dle následujícího zadání:

- a) Vytvořte jeden FC blok, který dokáže řídit jednu zabezpečenou místnost. Při vytváření funkce využijte její parametry (vstupní, výstupní, vstupně/výstupní, dočasné atd.) tak, aby se ta samá funkce dala volat minimálně dvakrát v OB1. Jednou bude volána pro místnost 1 a podruhé pro místnost 2. Uvažujte o místnosti, kde je jedno okno a jedny dveře. Jak okno, tak i dveře jsou vybaveny magnetickým detektorem.

Navíc v místnosti je detektor pro detekci tříštění skla. Také se zde nachází PIR detektor. Všechny detektory v klidovém stavu jsou sepnuty (NC kontakt). V místnosti je vyhlášen alarm, pokud alespoň jeden z detektorů je rozepnut (vstup do PLC je aktivní) ve chvíli, kdy je místnost zastřežena. Pro kvitování alarmu použijte jeden z volných vstupů (tlačítek), který bude kvitovat alarm v obou místnostech. V místnosti je světlo, které se rozsvěcí, pokud není zastřeženo a otevřou se dveře. Zhasíná, ve chvíli kdy jsou dveře zavřeny a místnost je zastřežená. Rolety, které jsou umístěné v místnosti na oknech, vyjíždí nahoru, pokud někdo deaktivuje režim zastřeženo a sjíždí dolů, pokud někdo zastřeží. Jsou ovládány dvěma výstupy. Jeden výstup musí být aktivní, pokud chceme, aby roleta vyjela nahoru a druhý výstup aby sjela dolů.

- b) Vytvořte FB blok, kde bude možnost ukládání posledního času, kdy byly otevřeny dveře v místnosti 1. Blok FB zvolte z důvodu potřeby uložení času. Použijte časové funkce, které jsou v záložce „Date and time-of-day.“ Vyzkoušejte v monitorovacím režimu, jak se program bude chovat, pokud uložíte hodnotu času jako statickou proměnnou (Static) nebo jako dočasnou proměnnou (Temp) v parametrech bloku. Využijte nápovědu k tomu, aby jste zjistili jaké bloky jsou potřeba a jak je použít.
- c) Vytvořte FB blok pro místnost 1, který vypíše dle stavu místnosti hlášení o stavu do datového typu STRING (použijte funkci „S\_MOVE“).
  - V první proměnné typu STRING bude vždy uloženo stav zastřežení (zastřeženo/nezastřeženo).
  - Ve druhé proměnné typu STRING bude vypisovat stav světel (svítí/nesvítí).
  - V třetí proměnné typu STRING bude ve stavu zastřeženo vypisovat, který detektor vyhlásil poplach. Pokud se bude jednat o všechny detektory, tak vypíše ten, co reagoval jako poslední.

Pozn.: Nejdříve si nadefinujte symbolické adresy všech potřebných vstupních a výstupních signálů pro „místnost 1“ a pro „místnost 2“. Příklad definice vstupů a výstupů místností je vidět na obrázcích níže. Nezapomeňte volat FC a FB bloky v OB1.

uloha\_zabezpecena\_mistnost > PLC\_1 [CPU 1214C DC/DC/DC] > PLC tags > detektory a zařzení - místnost 1 [8]

detektory a zařzení - místnost 1

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	PIR_detektor_m1	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	detektor_třítění skla_m1	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	okno - mag_detektor_m1	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	dveře - mag_detektor_m1	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	světlo_m1	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	rolety_nahoru_m1	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	rolety_dolu_m1	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	aktivace_alarmu_m1	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Obr. 38 Symbolické adresy detektorů a zařzení pro místnost 1

uloha\_zabezpecena\_mistnost > PLC\_1 [CPU 1214C DC/DC/DC] > PLC tags > detektory a zařzení místnosti 2 [8]

detektory a zařzení místnosti 2

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	PIR_detektor_m2	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	detektor_třítění skla_m2	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	okno - mag_detektor_m2	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	dveře - mag_detektor_m2	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	světlo_m2	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	rolety_nahoru_m2	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	rolety_dolu_m2	Bool	%Q0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	aktivace_alarmu_m2	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

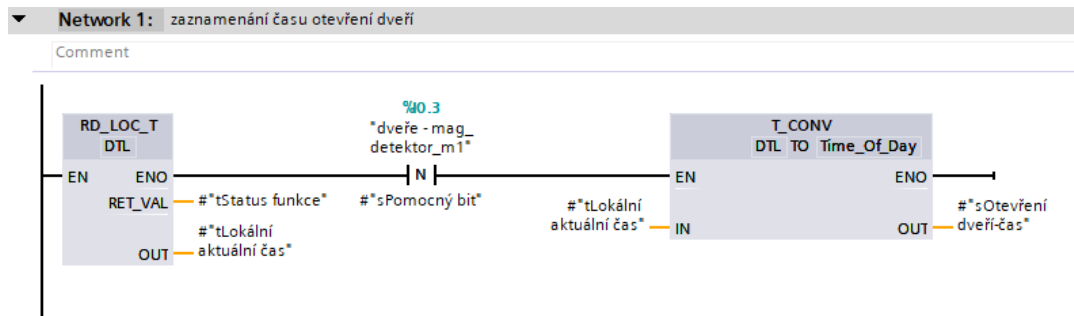
Obr. 39 Symbolické adresy detektorů a zařzení pro místnost 2

### 3.3.2 Řešení

Viz. příloha této bakalářské práce.

	Name	Data type	Default value	Comment
1	Input			
2	iPIR detektor	Bool		
3	iDetektor třítění skla	Bool		
4	iMag detektor okno	Bool		
5	iMag detektor dveře	Bool		
6	Output			
7	oRolety nahoru	Bool		
8	oRolety dolu	Bool		
9	InOut			
10	ioZvuková signalizace	Bool		
11	ioAktivace světla	Bool		
12	ioDetekce hrany paměť	Bool		
13	Temp			
14	tMístnost OK	Bool		
15	Constant			
16	Return			
17	místnost	Void		

Obr. 40 Ukázka řešení definice parametrů FC bloku „místnost“



Obr. 41 Ukázka řešení ukládání času otevření dveří

### 3.3.3 Komentář k řešení

U FC bloku „*místnost*“ je potřeba si nastavit nejdříve parametry bloku (vstupní, výstupní, vstupně-výstupní, dočasné atd.), které víme, že budeme potřebovat a můžou být různé pro „*místnost 1*“ a „*místnost 2*“ (například vstupy, výstupy). První písmena u názvů parametrů bloků jsou zvoleny podle toho, o jaký typ parametru se jedná. To především z důvodu větší přehlednosti při vytváření programu. FC blok „*místnost*“ je programován pomocí grafického programovacího jazyku LAD.

První network funkce „*místnost*“ pouze detekuje, jestli všechny detektory jsou v klidovém stavu, pokud je zastřeženo. Jelikož signál „*zastřeženo*“ je společný pro obě místnosti, tak je ve funkci přímo definován výstup pro signál *zastřeženo*. Pro testování úlohy by signál „*zastřeženo*“ mohl být použit jak paměťový bit, kterému lze v monitorovacím režimu měnit hodnotu. Protože funkci chceme volat pro dvě místnosti, tak je potřeba nastavit vstupní parametry pro signály detektorů. To proto, abychom jsme mohli pro různá volání funkce přiřadit různé vstupy. Stav detektorů se zapisuje do dočasného parametru „*tMístnost OK*.“ To znamená, že s touto hodnotou pracujeme pouze po dobu provádění funkce a o tuto hodnotu přijdeme po dokončení funkce. V dalším cyklu se však průběh opakuje. To znamená, opět se kontroluje stav detektorů a ten je opět zapsán do parametru „*tMístnost OK*“, který je posléze ve funkci znovu používán.

Druhý network funkce „*místnost*“ slouží pro aktivaci zvukové signalizace a její kvitace. Jelikož tímto signálem aktivujeme zvukovou signalizaci a zároveň potřebujeme vědět její aktuální stav, tak je pro signál „*ioZvuková signalizace*“ zvolen vstupně-výstupní parametr (InOut). Což znamená, že lze do tohoto parametru nejen zapisovat, ale i jej číst.

Třetí network funkce „*místnost*“ zapíná a vypíná světlo dle požadavku. Protože chceme, aby světlo svítilo ve chvíli, kdy otevřeme dveře a nezhaslo opět při zavření dveří, tak je zde prvek

detekce náběžné hrany. Tento prvek potřebuje paměť, která se nemaže po dokončení bloku. Není možné tedy použít dočasný parametr a je nutné použít vstupně-výstupní parametr, kdy při volání bloku je k tomuto parametru přiřazena globální proměnná (paměťový bit). V případě, že by byl program vytvářen v FB bloku, tak by se dalo využít statického parametru, kdy hodnotu neztrácíme po dokončení bloku.

Ve čtvrtém networku je pouze aktivace pohybu rolet dle stavu zastřežení. V případě aplikace této funkce na reálné místnosti by bylo potřeba v tomto networku přiřadit koncový spínač poloh pro vypnutí aktivace pohybu.

Vytvořená funkce je dvakrát volán v OB1. Pokaždé jsou však FC bloku přiřazeny jiné proměnné. V případě, že by vznikla potřeba zabezpečení další místnosti, která by měla stejné požadavky, tak by bylo nutné pouze vložit funkci „*místnost*“ do OB1 a přiřadit příslušné proměnné pro novou místnost k parametrům bloku.

U FB bloku „*čas otevření dveří*“, který slouží k získání času, kdy byly otevřeny dveře v místnosti 1, jsou potřeba dvě funkce. Jedna z nich je „RD\_LOC\_T“, která slouží k uložení hodnoty místního času a data z PLC. Výstup funkce musí být uložen do proměnné typu DTL. V tomto případě se jedná o dočasný parametr „*tLokální aktuální čas*“. Prvkem pro detekci sestupné hrany (zavření dveří), se aktivuje blok „T\_CONV“, který převede hodnotu parametru „*tLokální aktuální čas*“ do statického parametru „*sOtevření dveří-čas*“. Datový typ tohoto parametru je TOD (Time-of-day, čas dne). Tím je získán aktuální čas dne, kdy byly dveře otevřeny, což lze vidět v monitorovacím režimu. Pokud by parametr „*sOtevření dveří-čas*“ byl zvolen jako dočasný, tak by jeho hodnota byla ztracena po dokončení bloku. Tím pádem by jeho hodnota nebyla ani vidět v monitorovacím režimu. Aby byl blok prováděn, tak je umístěn v OB1.

Posledním blokem volaným v OB1 je blok FB „*hlášení o stavu*“. Zde je především využita funkce „S\_MOVE“. Ta slouží k přesunu textového řetězce do jiné proměnné. V tomto případě do jiného parametru.

V prvním networku FB bloku „*hlášení o stavu*“ je dle stavu zastřeženo zapsána do dočasného parametru „*tHláška zastřeženo*“ hláška „zastřeženo“ nebo „nezastřeženo“. Datový typ parametru je STRING. Parametr může být dočasný, protože v každém cyklu je stav zastřežení kontrolován a zapsán. Stejný princip je použit ve druhém networku pro vypsání hlášky o stavu světla.



Ve třetím networku FB bloku „*hlášení o stavu*“ je vypisován stav detektorů. Respektive pokud některý z detektoru zareaguje, tak prvek pro detekci náběžné hrany aktivuje funkci „S\_MOVE“, která provede zápis příslušné textové hlášky do statického parametru „*sHláška detektor*“. Statický parametr je zvolen proto, aby hodnota byla uložena i po dokončení bloku. Datový typ tohoto parametru je STRING. Pro detekci hrany je opět potřeba paměť. Jelikož je použit tentokrát FB blok, tak je využito statické proměnné. V části, kde se nastavují statické parametry tak pro *sPomocný bit* je jako datový typ zvolen „Array[0..3] of Bool.“ To znamená, že pro *sPomocný bit* jsou deklarovány čtyři proměnné typu Bool, které mají stejný název a jsou číslovány od nuly do tří. Aby byl blok prováděn, tak je umístěn v OB1.

### 3.4 Poslání e-mailu v případě vyhlášení poplachu

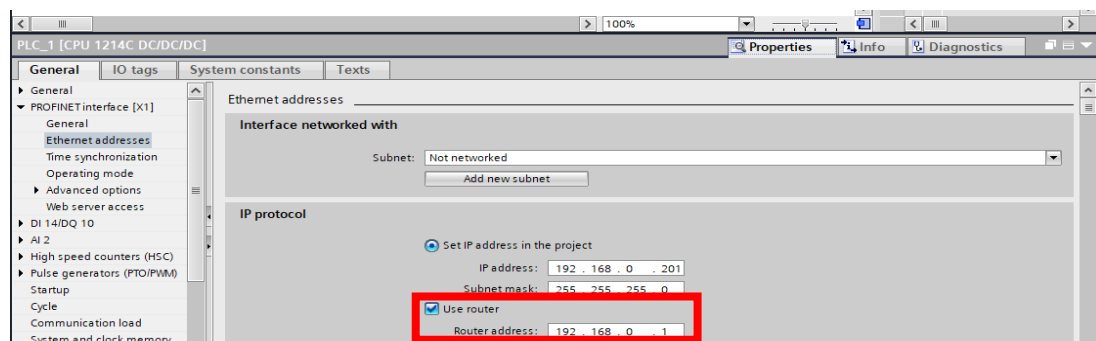
Úloha má za úkol demonstrovat možnost zaslání e-mailu na soukromý e-mail například pokud je vyhlášen poplach v domě.

#### 3.4.1 Zadání

Uvažujte o situaci, kdy chcete dostávat z PLC pomocí protokolu SMTP email o tom, že alespoň jedny ze tří dveří, které máte v domě, byly otevřeny. Detektory dveří jsou připojeny ke vstupům PLC.

Vytvořte funkci, která toto bude provádět a volejte ji z OB1. K řešení použijte funkci „TMAIL\_C“, která slouží ke komunikaci se SMTP serverem a vytvořte DB blok, kde budou vloženy všechny data potřebné k zaslání e-mailu. Data dle potřeb přiřaďte k parametrům funkce „TMAIL\_C.“ Ke vstupnímu parametru „MAIL\_ADDR\_PARAM“ přiřaďte proměnnou z vytvořeného datového bloku. Datový typ této proměnné zvolte „TMail\_V4.“ Nezapomeňte v DB bloku ve sloupci „Start value“ nadefinovat všechny potřebné hodnoty používaných proměnných. Pro potřeby vyzkoušení zaslání emailu bez toho aniž by jste museli zadávat své osobní údaje, tak můžete použít email, který byl vytvořen pro tento účel. Jedná se o e-mail, který byl vytvořen na webových stránkách Seznam.cz. Přihlašovací jméno je „uzivatelutb@seznam.cz“ a heslo je „FAI2019.“ IP adresa SMTP serveru Seznamu je 77.75.78.48. Aby PLC komunikovalo s prostředím Internetu, tak je zapotřebí nastavit v konfiguraci PLC IP adresu Acces pointu (AP, přístupový bod).

Pozn.: Při řešení pracujte s nápovědou v TIA Portalu. Nezapomeňte volat funkci v OB1. Projekt nahrajte do PLC a odzkoušejte funkčnost zasílání e-mailu.



Obr. 42 Nastavení IP adresy Acces pointu

### 3.4.2 Řešení

Viz. příloha této bakalářské práce.

DB_e-mail									
Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment	
1	Static								
2	žádost o odeslání	Bool	false						
3	adresa příjemce	String[240]	'uzivatelutb@seznam.cz'						
4	předmět	String[240]	'email z PLC'						
5	text	String[240]	'Dvere byly otevreny!'						
6	příloha	Byte	16#0						
7	dokončení posílání	Bool	false						
8	probíhá posílání	Bool	false						
9	chyba odeslání	Bool	false						
10	chybové hlášení	Word	16#0						
11	parametry komunikace	TMail_V4							
12	InterfaceId	HW_ANY	64						HW-identifier of IE-interface submodule
13	ID	CONN_OUC	100						connection reference / identifier
14	ConnectionType	Byte	16#32						type of connection: 32= TMail_V4
15	ActiveEstablished	Bool	false						active/passive connection establishment
16	CertIndex	Byte	16#0						index of SSA certificate on CM/CP
17	WatchDogTime	Time	T#5s						watchdog time to monitor SMTP server association
18	MailServerAddress	IP_V4							IP address (IPv4) of mail server
19	ADDR	Array[1..4] of Byte							IPv4 address
20	ADDR[1]	Byte	77						IPv4 address
21	ADDR[2]	Byte	75						IPv4 address
22	ADDR[3]	Byte	78						IPv4 address
23	ADDR[4]	Byte	48						IPv4 address
24	UserName	String[254]	'uzivatelutb@seznam.cz'						name of user
25	PassWord	String[254]	'FAI2019'						user password
26	From	EMAIL_ADDR							source mail address
27	LocalPartPlusAt...	String[64]	'uzivatelutb@'						local part of e-mail address plus "@" sign
28	FullQualifiedD...	String[254]	'seznam.cz'						full qualified domain name part of e-mail address

Obr. 43 Ukázka řešení definice datového bloku pro zaslání e-mailu

### 3.4.3 Komentář k řešení

Ukázkové řešení je provedeno v programovacím jazyce LAD. Ve funkci „zaslání e-mailu“ je v prvním networku detekce otevření dveří, kdy pokud se otevrou alespoň jedny dveře, tak je zažádáno o zaslání e-mailu.

Ve druhém networku je umístěna funkce „T\_MAIL.“ K této funkci jsou přiřazené proměnné z globálního datového bloku „DB\_e-mail.“ V něm jsou uloženy všechna potřebná data ke

komunikaci se SMTP serverem včetně údajů k zaslání e-mailu. Správná definice hodnot tohoto datového bloku je pro úlohu klíčová. V řešení úlohy se e-mail odesílatele a příjemce shoduje.

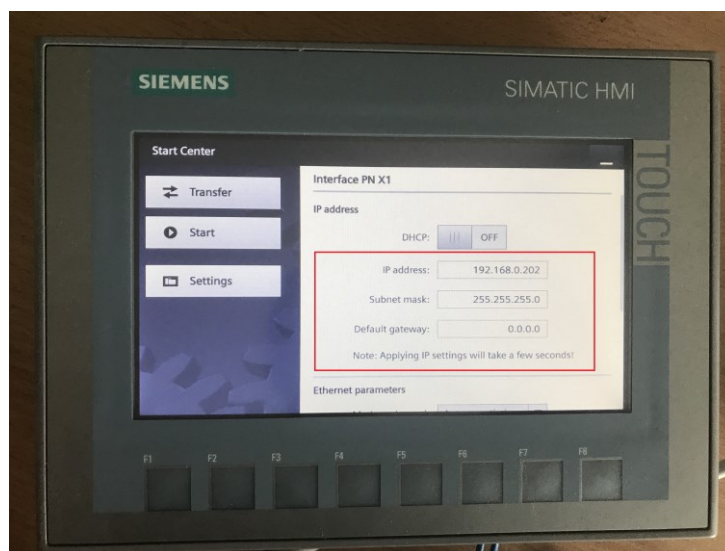
Funkce „zaslání e-mailu“ je poté volána z OB1.

### 3.5 Propojení PLC s HMI a vytvoření vizualizace

HMI přináší možnost vizualizace procesu a možnost do něj zasahovat. Vytváří tak rozhraní mezi člověkem a strojem (HMI, Human Machine Interface). Následující úloha má za úkol demonstrovat jak lze propojit PLC S7 - 1200 a HMI KTP700 Basic od společnosti Siemens. V dalším kroku si pak student bude moct vyzkoušet zobrazení hodnot z PLC a vytvořit jednoduchou vizualizaci zabezpečené místnosti. K tomu můžeme využít projekt s úlohou zaměřenou na blokové parametry u FC a FB bloků.

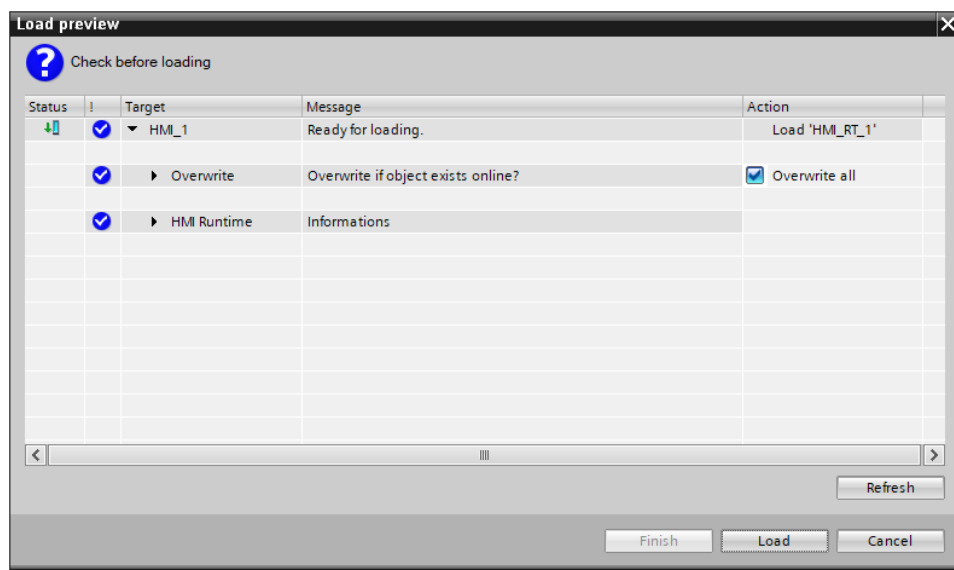
#### 3.5.1 Zadání

Otevřete si projekt pro úlohu Blokové parametry u FC a FB bloků a vložte do konfigurace projektu HMI panel KTP700 Basic, který najdete v katalogu. Propojte PLC a HMI v konfiguraci projektu a nastavte ve vlastnostech HMI panelu jeho IP adresu. Tu samou IP adresu přiřadte HMI panelu v jeho nastavení, které se zobrazí po zapnutí panelu.



Obr. 44 Nastavení IP adresy přímo v panelu HMI  
KTP700 Basic

Po dokončení konfigurace v TIA Portalu se v „Project tree“ zobrazí vložený HMI panel. V jeho záložce „Screen“ vložte novou obrazovku (screen). V pravé části TIA Portalu se zobrazí „Tool box,“ kde lze najít prvky, které je možné využít a zobrazit na panelu (tlačítka, grafické objekty, I/O pole atd.). Vložte na obrazovku dvě tlačítka (prvek Button). Jedno tlačítko bude sloužit k aktivaci zastřežení a druhé k deaktivaci zastřežení. Pokud jste v úloze Blokové parametry u FC a FB bloků použili pro zastřežení jeden ze vstupních signálů, tak pro tuto úlohu adresu proměnné (tagu) „zastřeženo“ změňte na paměťový bit (například „%M0.0“). V záložce „Default tag table“ vložte proměnou, kterou bude používat HMI panel a propojte ji s proměnou v PLC. Vizualizaci tlačítek nahrajte do HMI panelu a vyzkoušejte jejich funkčnost. Projekt do HMI panelu nahrajete obdobně jako projekt do PLC. Prvním krokem pro nahrání projektu do HMI panelu je označit si HMI panel v „Project tree“ a v horní liště zvolit v záložce „Online“ volbu „Extended download to device.“ Pak lze postupovat stejně jako při nahrávání projektu do PLC. Pokud se zobrazí při nahrávání hláška, zda chcete přepsat existující objekt, tak jej potvrďte a projekt nahrajte.



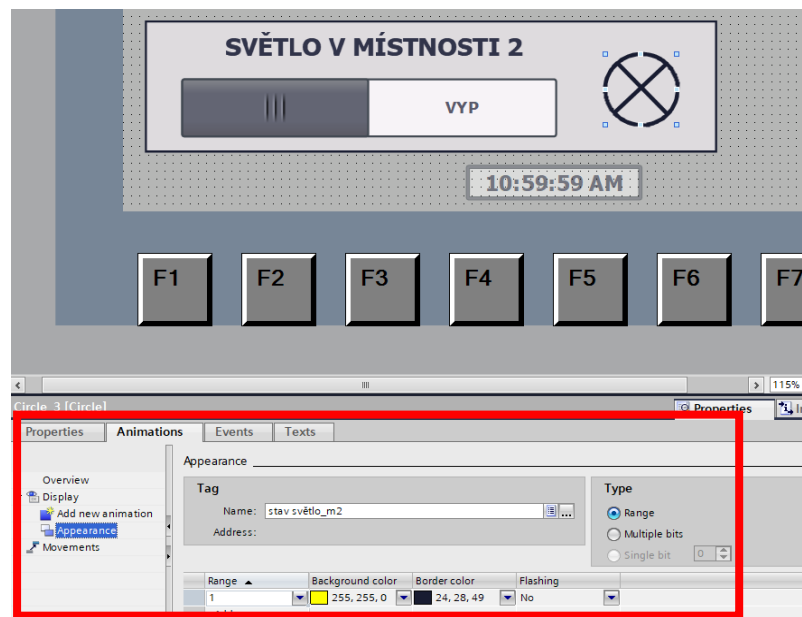
Obr. 45 Hláška pro přepsání projektu

Následně pomocí dostupných grafických prvků vytvořte vizualizaci dle následujícího zadání:

- Vytvořte úvodní obrazovku (MENU), kde bude možnost přepínat na ostatní obrazovky a možnost kvitování vyhlášeného alarmu. V „Default tag table“ upravte adresu kvitace ze vstupního signálu na paměťový bit.
- Vytvořte „Template,“ což je stránka, na které jsou grafické prvky, které se mají zobrazit na všech požadovaných obrazovkách. Umístěte zde tlačítko pro návrat na

- úvodní obrazovku. Volitelně umístěte také zobrazení času. Je potřeba získání času z PLC. Využijte a případně upravte blok, kde se získával čas otevřených dveří.
- c) Vytvořte obrazovku, kde bude možnost měnit stav zastřežení (využijte stránky vytvářené v úvodu úlohy). Stav zastřeženo/nezastřeženo signalizujte zelenou/červenou barvou obou tlačítek.
  - d) Vytvořte obrazovku, kde bude možnost sledovat stav detektorů obou místností. Stav signalizujte některým grafickým prvkem, který bude měnit barvu. Například zelenou barvou bude označen OK stav a červenou barvou bude označen NOK stav detektoru.
  - e) Vytvořte obrazovku, kde bude možnost aktivovat zvukovou signalizaci místnosti 1 i místnosti 2 kvůli otestování funkčnosti. Možnost testu zvukové signalizace by měl být především v režimu zastřeženo.
  - f) Vytvořte obrazovku, kde bude možnost aktivovat světla v obou místnostech, a doplňte signalizaci rozsvícení světla. Manuálně bude možnost aktivovat světla pouze při stavu nezastřeženo. Doplňte možnost zvolení automatické a manuální aktivace. Automatická aktivace bude rozsvěcet světla pomocí funkce z původní úlohy. Manuální aktivace umožní rozsvícení světla z HMI panelu. Dle potřeby upravte FC blok „*místnost*.“
  - g) Volitelně doplňte do „Template“ možnost signalizace, že je v místnosti 1 nebo v místnosti 2 vyhlášen alarm.

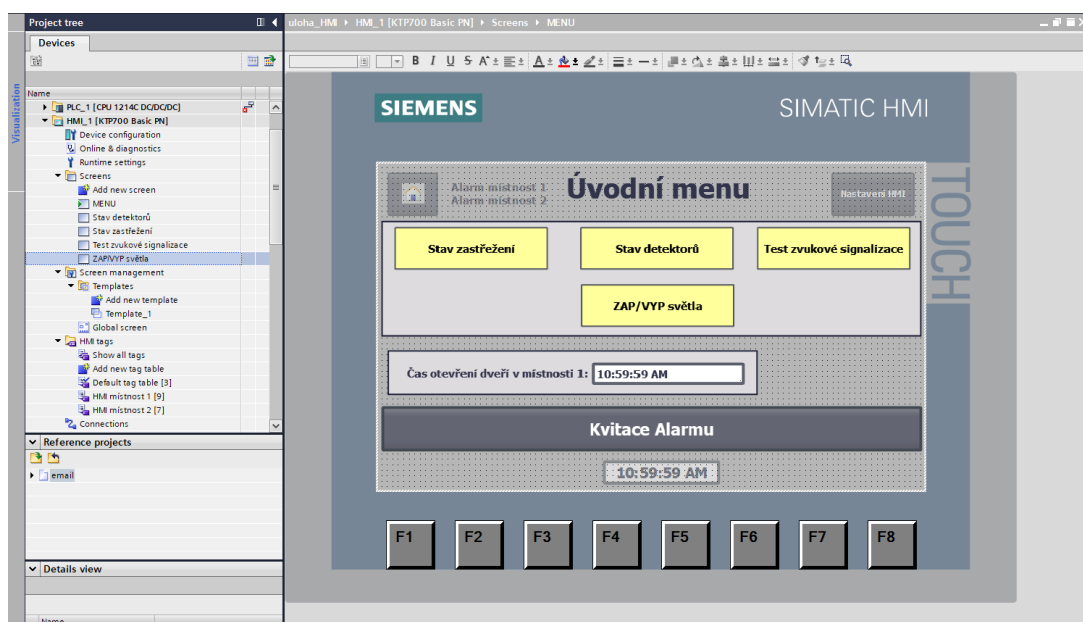
Pozn.: Nadefinujte si proměnné potřebné v HMI a propojte je s proměnnými v PLC. Pro signalizace využijte funkce v animacích grafických objektů („Animations“). Kde je možnost přidání animace. Konkrétně možnost viditelnosti objektu dle hodnoty přiřazené proměnné nebo možnost změny barvy dle hodnoty přiřazené proměnné. Pokud uděláte změnu v programu pro PLC nezapomeňte tuto změnu do PLC nahrát.



Obr. 46 Ukázka možností animací grafických prvků

### 3.5.2 Řešení

Viz. příloha této bakalářské práce.



Obr. 47 Návrh grafické vizualizace úvodní obrazovky

### 3.5.3 Komentář k řešení

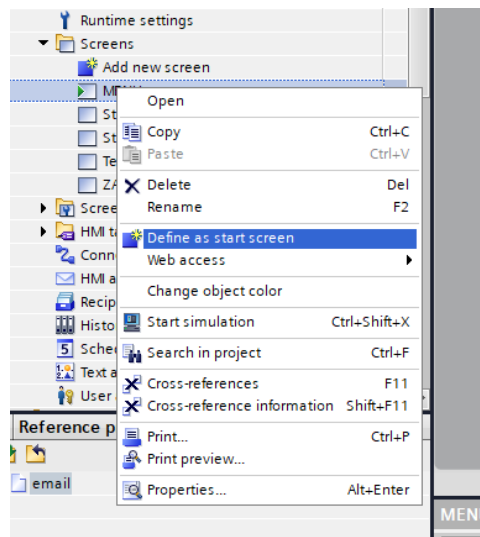
Úloha demonstruje možnost vizualizace stavů a hodnot z PLC. V úloze je důležité v prvním kroku si nadefinovat proměnné (tagy) pro HMI a propojit je s proměnnými používanými v PLC.

Na úvodní obrazovce „MENU“ jsou umístěná čtyři tlačítka, která slouží k zobrazení ostatních obrazovek. Při stisku tlačítka se pomocí funkce „ActiveScreen,“ která se pro tlačítko definuje v záložce „Events,“ zobrazí příslušná obrazovka.



Obr. 48 Aktivace zobrazení obrazovky „Stav zastřežení“

Aby se po zapnutí panelu zobrazila vždy obrazovka „MENU,“ tak při kliknutí pravým tlačítkem myši na obrazovku „MENU“ v „Project tree“ je zvolena možnost „Define as start screen.“

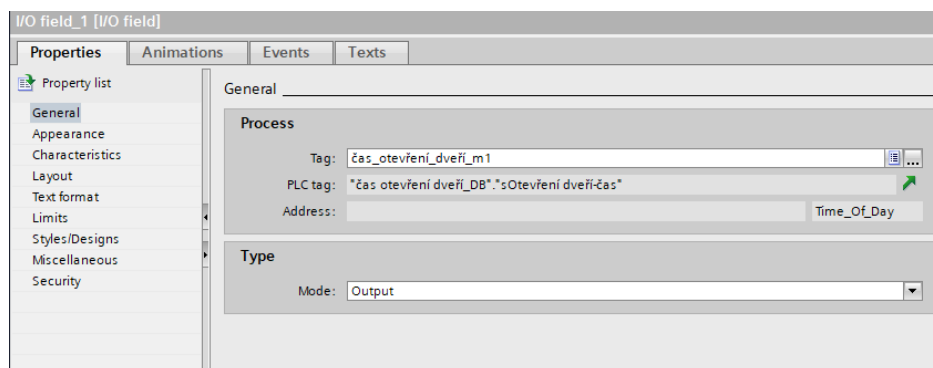


Obr. 49 Volba úvodní obrazovky  
po zapnutí HMI paneulu

Tlačítko „Kvitace“ umístěné na obrazovce „MENU“ využívá v záložce „Events“ při stisknutí funkce „SetBit“ pro aktivaci bitu. A následně při puštění tlačítka se aktivuje funkce

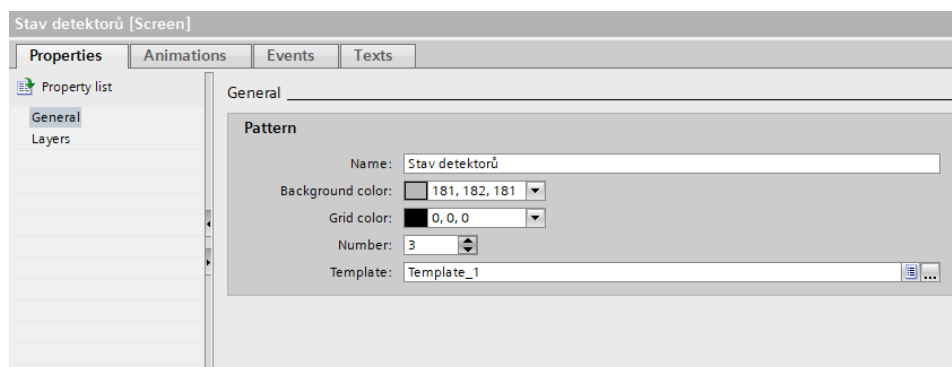
„ResetBit“ pro reset bitu, aby nebyl stále aktivní. Stejný princip je použit pro tlačítka, kterými se testuje zvuková signalizace na obrazovce „Test zvukové signalizace.“ V PLC pro zjednodušení ukázky byl vstup pro kvitování předadresován na paměťový bit „%M0.1.“

Úvodní obrazovka je ještě doplněna o zobrazení času, kdy byly naposled otevřeny dveře v místnosti 1. K tomu slouží prvek „I/O field.“ Jeho mód je v záložce „Properties“ nastaven na „Output,“ což znamená, že lze tento prvek použít pouze pro zobrazení hodnoty. Prvek je možné využít případně i pro zápis hodnoty. To ale v tomto případě není žádoucí.



Obr. 50 Zvolení módu „I/O field“ pro zobrazení času otevření dveří

Použitý „Template“ je vytvořen v „Project tree“ HMI panelu v podzáložce „Screen management“ a „Tempalates.“ Tlačítko „Home“ slouží k návratu na úvodní obrazovku, a jelikož je umístěn v „Template,“ tak po přiřazení „Template“ k ostatním obrazovkám budou všechny zde vytvořené prvky zobrazeny na všech obrazovkách. Přiřazení „Template“ se provádí ve vlastnostech jednotlivých obrazovek.



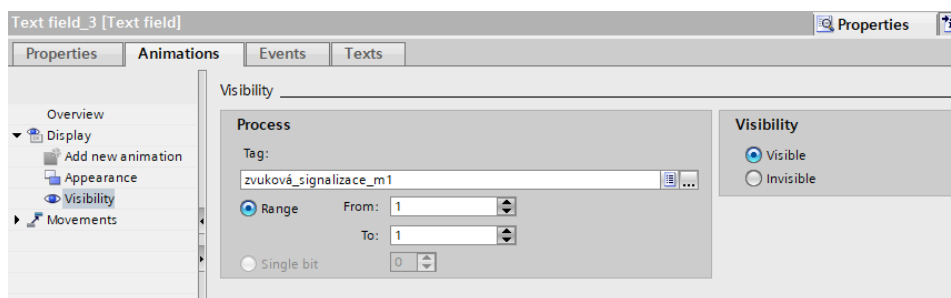
Obr. 51 Přiřazení „Template“ k obrazovce „Stav detektorů“

V pravém okraji se nachází tlačítko „Nastavení HMI,“ které pomocí funkce „StopRuntime“ v „Events“ zobrazí nastavení HMI panelu, což může být někdy žádoucí.



K získání času z PLC je využit FB blok „čas otevření dveří.“ Zde je přidán statický parametr „sČas PLC,“ do které je ukládán čas běžící v PLC. Tato proměnná je poté propojena s proměnou v HMI a přiřazena do „I/O field“ v „Template.“ Čist hodnotu času z PLC by bylo vhodné číst nejspíše v samostatném programovém bloku. Pro zjednodušení bylo ale využito FB bloku „čas otevření dveří.“

Textové pole „Alarm místnosti 1“ a „Alarm místnosti 2“ využívají dvě animační funkce a to „Appearance“ a „Visibility.“ Funkce Appearance mění barvu textu na červenou a to tak, že text bliká červeně a pouze ve chvíli, kdy je vyhlášen poplach v konkrétní místnosti (je aktivován příslušný výstup). Text je zobrazen, pouze pokud je vyhlášen poplach. K tomu slouží právě funkce „Visibility,“ která umožňuje zobrazení nebo skrytí textu podle hodnoty přiřazené proměnné.



Obr. 52 Funkce „Visibility“

Zastřežení se aktivuje a deaktivuje na obrazovce „Stav zastřežení.“ Děje se tak pomocí dvou tlačítek, kde tlačítko „Aktivace zastřežení“ aktivuje příslušný bit a tlačítko „Deaktivace zastřežení“ příslušný bit deaktivuje. Navíc je zde přidán „I/O field“ kvůli vyobrazení hlášky, který detektor byl jako poslední aktivní v místnosti 1.

Na obrazovce „Stav detektorů“ je přehled všech detektorů a jejich stavů. Ty jsou signalizovány grafickým prvkem „Circle,“ který mění barvu pomocí funkce „Appearance,“ podle stavu detektoru.

Poslední obrazovka „ZAP/VYP světla“ slouží ke zvolení módu svícení (AUTO/MANUAL) v obou místnostech zároveň. V módu „AUTO“ se světla rozsvěcí stejně jako v úloze Blokové parametry u FC a FB bloků. V módu „MANUAL“ je možné světla rozsvěcet a zhasínat z HMI panelu, pokud jsou místnosti ve stavu nezastřeženo.

Kvůli volbě módu byly vloženy do funkce „místnost“ dva NO a dva NC prvky ve druhém networku pro umožnění/neumožnění aktivace světel dle požadovaného módu. K těmto prvkům byl nadefinován paměťový bit „HMI\_aktivace\_AUTO\_svícení,“ který je propojen

s proměnou v HMI panelu a ta je přiřazena přepínači pro volbu módu. Pro ovládání světel v módu „MANUAL“ byl přidán navíc jeden NO a jeden NC prvek do druhého networku ve funkci „*místnost*.“ Parametr přiřazený k těmto dvěma prvkům je vstupní. K nim jsou pak přiřazeny při volání bloku paměťové bity a ty jsou propojeny s proměnou v HMI panelu. Proměnná v HMI je svázaná s grafickým prvkem přepínač (Switch), který aktivuje a deaktivuje bit dle polohy přepínače. Jako signalizace rozsvíceného a zhasnutého světla slouží symbol žárovky, který je nakreslen za pomoci grafických prvků v TIA Portalu. Prvek mění barvu na žlutou za předpokladu, že je aktivován výstup aktivující světlo. Opět je k tomuto účelu využita funkce „Appearance.“

## ZÁVĚR

Bakalářská práce je vypracovaná tak, aby studentům přinesla nejdříve základní informace o tom, co to PLC vůbec je a jakým způsobem funguje, o čemž je pojednáváno v úvodu práce. Student v úvodní části tak získá základ pro to, aby byl schopen pochopit problematiku PLC. V závěru první kapitoly práce je zmíněná možnost propojení PLC se zabezpečovacími systémy pomocí sběrnice KNX a také týkající se problémy tohoto propojení a to hlavně legislativní. Ty se týkají především problému určení bezpečnostní třídy po připojení zabezpečovacího systému pomocí řídicí sběrnice. Z těchto důvodů spíše při implementaci PLC jako řídicího členu například chytré domácnosti, kde jsou zabezpečovací systémy jako je PZTS nebo EPS, uživatel musí uvažovat o tom, že spíše bude dostávat pouze informativní data z těchto zabezpečovacích zařízení. PLC jako řídicí člen může ale na získané informace reagovat. Například zasláním e-mailu o stavu zabezpečení budovy. Čímž se zabývá jedna z vytvořených úloh pro studenty.

Druhá kapitola je přímo zaměřená na popis PLC řady S7 - 1200 a vývojového prostředí TIA Portal. Zde je především snaha blíže specifikovat a vysvětlit fungování konkrétního PLC a přiblížit možnosti PLC S7 - 1214C DC/DC/DC a TIA Portalu, které by mohl student při řešení jakékoliv úlohy využít. Při zpracování této kapitoly bylo především využito oficiálních manuálů, které jsou volně k dispozici převážně v anglickém jazyce na webových stránkách společnosti Siemens a.s. Nutno podotknout, že je zde možné najít velké množství manuálů, což umocňuje rozšiřování nejen PLC řady S7 - 1200, ale také ostatních řad, které Siemens nabízí. V závěru druhé kapitoly byl vytvořen popis jak postupovat při prvním kontaktu s PLC S7 - 1200 a TIA Portalem. To znamená sepsání popisu jak vytvořit, zkompilovat a nahrát projekt do PLC pomocí TIA Portalu. Pro lepší prezentaci popisu jsem vložil obrázky k jednotlivým krokům tak, aby co nejlépe demonstrovaly, jak přesně postupovat.

Závěrečná kapitola je především zaměřená na vytvoření úloh pro studenty. Snahou bylo, aby úlohy, co nejlépe demonstrovali možnosti programování PLC S7 - 1200 a zároveň byly zaměřeny na oblast zabezpečovacích systémů. Při vypracovávání úloh je podstatnou věcí se naučit pracovat s nápovědou TIA Portalu. Ke každé úloze je popsáno, co bylo v úloze klíčové pro její vytvoření a jaké nové funkce nebo jiné prvky byly použity, tak aby studenti mohli porovnat výsledek, kterého dosáhli oni při vypracování úlohy. K těm zajímavějším úlohám patří především úloha se zasláním e-mailu nebo úloha s HMI panelem. V té si stu-

dent nejen vyzkouší propojení PLC a HMI panelu, ale také vytvořit malou vizualizační aplikaci pro simulované řízení dvou zabezpečených místností. Vypracované projekty s řešením úloh jsou jako archivované soubory, které lze vygenerovat V TIA Portalu, přílohou této bakalářské práce.

**SEZNAM POUŽITÉ LITERATURY**

- [1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace, 1.díl. 1. Vyd. Praha: BEN – technická literatura, 2009. ISBN 80-860-5658-9
- [2] ŠMEJKAL, Ladislav. PLC a automatizace, 2.díl. 1. Vyd. Praha: BEN – technická literatura, 2005. ISBN 80-7300-087-3.
- [3] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.
- [4] SIMATIC Programming with STEP 7 V5.5: Programming manual. Industry Online Support International [online]. Norimberk: Siemens AG, 2010 [cit. 2016-01-29]. Dostupné z: <https://support.industry.siemens.com/cs/document/45531107/simatic-programming-with-step-7-v5-5?dti=0&pnid=14341&lc=en-WW>
- [5] 60 years of SIMATIC | History Features | Siemens. 301 Moved Permanently [online]. Copyright © Siemens 1996 [cit. 27.03.2019]. Dostupné z: <https://new.siemens.com/global/en/company/about/history/history-features/60-years-of-simatic.html>
- [6] Historické pozadí boje o první PLC - 12/02/2015 - Control Engineering Česko. Hlavní strana - Control Engineering Česko [online]. Copyright © 2007 [cit. 27.03.2019]. Dostupné z: <http://www.controlengcesko.com/hlavni-menu/artykuly/artykul/article/historicke-pozadi-boje-o-prvni-plc/>
- [7] Historie - PLC. PLC AUTOMATIZACE [online]. Dostupné z: <http://plc-automatizace.cz/knihovna/historie/historie-plc.htm>
- [8] Střední průmyslová škola a Vyšší odborná škola Chomutov. Automatizace 1 [elektronická skripta]. Chomutov: Střední průmyslová škola a Vyšší odborná škola Chomutov [cit. 1.03.2019]. Dostupné z: <http://web.spscv.cz/~madaj/skra3.pdf>
- [9] ROBENEK, Jan. Návrh PLC očima vývojáře – 8. část (Digitální vstupy / výstupy). Vývoj.HW.cz [online]. 14. 9. 2010 [cit. 15.03.2019]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/navrh-plc-ocima-vyvojare-8-cast-digitalni-vstupy-vystupy.html>
- [10] KRŠÁK, Jan. Příprava laboratoře pro předmět Programovatelné automaty. Zlín, 2006. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Ústav automatizace a řídicí techniky. Vedoucí bakalářské práce Tomáš Sysala.

- [11] VOJÁČEK, Antonín. Analogový vs. digitální přenos hodnot. Kdy ještě volit analogový výstup?. Automatizace.HW.cz [online]. 7. 4. 2015 [cit. 27.03.2019]. Dostupné z: <https://automatizace.hw.cz/mereni-a-regulace-prumyslove-sbornice-a-komunikace/analogovy-vs-digitalni-prenos-hodnot-kdy-jeste-volit-analogovy-vystup>
- [12] PLC - sestava. PLC AUTOMATIZACE [online]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-hw-sestava.htm>
- [13] Communication with S7 CPU via KNX Gateway. Industry Support Siemens [online]. 2017 [cit. 30. 3. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/document/109739689/communication-with-s7-cpu-via-knx-gateway?dti=0&lc=en-WW>
- [14] VOTRUBA, Zdeněk. Integrace bezpečnostních systémů v budovách. Automa časopis pro automatizační techniku [online]. 2012 [cit. 30. 3. 2019]. Dostupné z: [http://automa.cz/cz/casopis-clanky/integrace-bezpecnostnich-systemu-v-budovach-2012\\_12\\_0\\_10042/](http://automa.cz/cz/casopis-clanky/integrace-bezpecnostnich-systemu-v-budovach-2012_12_0_10042/)
- [15] VOJÁČEK, Antonín. Sběrnice KNX pro řízení budov - 1.část. Automatizace.HW.cz [online]. 10. 6. 2006 [cit. 30. 3. 2019]. Dostupné z: <https://automatizace.hw.cz/clanek/2006061001>
- [16] SIMATIC S7 S7 - 1200 Programmable controller System manual. Industry Support Siemens [online]. 2014. [cit. 1. 4. 2019]. Dostupné z: [https://cache.industry.siemens.com/dl/files/465/36932465/att\\_106119/v1/s71200\\_system\\_manual\\_en-US\\_en-US.pdf](https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf)
- [17] SIMATIC STEP 7 and WinCC Engineering V15 System Manual. Industry Support Siemens [online]. 2012 [cit. 3. 4. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/document/109764516/simatic-step-7-basic-professional-v15-and-simatic-wincc-v15?dti=0&lc=en-AR>
- [18] Začátek programování s SIMATIC S7 - 1200. Industry Support Siemens [online]. 2012 [cit. 4. 4. 2019]. Dostupné z: [http://www1.siemens.cz/ad/current/content/data\\_files/automatizacni\\_systemy/mikrosystemy/simatic\\_s71200/manualy/ucebnice\\_simatic-s71200.zip](http://www1.siemens.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/simatic_s71200/manualy/ucebnice_simatic-s71200.zip)
- [19] Programming Guideline for S7 - 1200/S7 - 1500. Industry Support Siemens [online]. 2014 [cit. 9. 4. 2019]. Dostupné z: [https://support.industry.siemens.com/cs/attachments/90885040/81318674\\_Programming\\_guideline\\_DOC\\_v16\\_en.pdf?download=true](https://support.industry.siemens.com/cs/attachments/90885040/81318674_Programming_guideline_DOC_v16_en.pdf?download=true)
- [20] KNX Základy. KNX.cz [online]. [cit. 28. 3. 2019]. Dostupné z: [https://knxcz.cz/images/clanky/KNX\\_Basics\\_CZ\\_screen2.pdf](https://knxcz.cz/images/clanky/KNX_Basics_CZ_screen2.pdf)

- [21] BLAŽEK, Jaroslav. Webový server v PLC Simatic S7 - 1200. Jaroslav Blažek -Blaja.cz [online]. 10. 1. 2013 [cit. 15. 4. 2019]. Dostupné z: <https://www.blaja.cz/plc-automaty/webovy-server-v-plc-simatic-s7-1200.html>
- [22] CPU 1217C. Industry Support Siemens [online]. [cit. 9. 4. 2019]. Dostupné z: [https://www.automation.siemens.com/bilddb/interfaces/InterfaceImageDB.asmx/GetImageVariant?objectkey=P\\_ST70\\_XX\\_06682&imagevariantid=1&lang=XX&interfaceuserid=MALL](https://www.automation.siemens.com/bilddb/interfaces/InterfaceImageDB.asmx/GetImageVariant?objectkey=P_ST70_XX_06682&imagevariantid=1&lang=XX&interfaceuserid=MALL)
- [23] CPU 1511C-1PN. Industry Support Siemens [online]. [cit. 9. 4. 2019]. Dostupné z: [https://www.automation.siemens.com/bilddb/interfaces/InterfaceImageDB.asmx/GetImageVariant?objectkey=P\\_ST70\\_XX\\_07838&imagevariantid=1&lang=XX&interfaceuserid=MALL](https://www.automation.siemens.com/bilddb/interfaces/InterfaceImageDB.asmx/GetImageVariant?objectkey=P_ST70_XX_07838&imagevariantid=1&lang=XX&interfaceuserid=MALL)
- [24] SIPLUS LOGO 24CE. Industry Support Siemens [online]. [cit. 9. 4. 2019]. Dostupné z: [https://www.automation.siemens.com/bilddb/interfaces/InterfaceImageDB.asmx/GetImageVariant?objectkey=P\\_SP01\\_XX\\_00404&imagevariantid=1&lang=XX&interfaceuserid=MALL](https://www.automation.siemens.com/bilddb/interfaces/InterfaceImageDB.asmx/GetImageVariant?objectkey=P_SP01_XX_00404&imagevariantid=1&lang=XX&interfaceuserid=MALL)
- [25] Firemní materiály, CD firmy EWWH, s.r.o. - distributora PLC SAIA pro Českou republiku, 2018 [cit. 25. 4. 2019].

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

A/D	Analogově - Digitální
AC	Alternating Current
AWL	Anweisungslist
CCTV	Closed Circuit Television
CPU	Centrální procesorová jednotka
CTU	Count Up
D/A	Digitálně - Analogový
DB	Data Block
DC	Direct Current
DEC	Digital Equipment Corp
EEPROM	Electrically Erasable Programmable Read Only Memory
EPROM	Erasable Programmable Read Only Memory
EPS	Elektrická Požární Signalizace
FB	Function Block
FC	Function
FUP	Funktionsplan
HMI	Human Machine Interface
IP	Internet Protocol
KB	Kilobyte
KNX	Konnex
KOP	Kontaktplan
LD (LAD)	Ladder Diagram
LED	Light Emitting Diode
MB	Megabyte



---

MODICON	Modular Digital Controller
NC	Normally Closed
NO	Normally Open
OB	Organization Block
PII	Proces Image Input
PIQ	Proces Image Output
PLC	Programmable Logic Controller.
PZTS	Poplachové Zabezpečovací a Tísňové Systémy
RAM	Random Access Memory
SCL	Structured Control Language
SMTP	Simple Mail Transfer Protocol
ST	Structured Text
STL	Statement List
TIA	Totally Integrated Automation
TON	On Delay Timer
USB	Universal Serial Bus

**SEZNAM OBRÁZKŮ**

Obr. 1 Kompaktní PLC Simatic S7 - 1200 od společnosti Siemens [22].....	15
Obr. 2 Modulární PLC Simatic S7 - 1500 od společnosti Siemens [23].....	16
Obr. 3 Mikro PLC SIPLUS LOGO 24CE 8 od společnosti Siemens [24].....	17
Obr. 4 Příklad programovacího jazyku LAD používaného v TIA Portalu.....	20
Obr. 5 Příklad programovacího jazyka FBD v TIA Portalu.....	20
Obr. 6 Příklad použití programovacího jazyka SCL v TIA Portalu.....	21
Obr. 7 Komunikace mezi PLC Siemens a zařízením KNX [13].....	23
Obr. 8 Signalizace pracovního módu PLC S7 - 1214C DC/DC/DC.....	26
Obr. 9 Schématické zobrazení volání programových bloků z OB1 [16].....	28
Obr. 10 Portálové zobrazení – Portal View v TIA Portalu.....	33
Obr. 11 Detail na možnosti v Portálovém zobrazení.....	33
Obr. 12 Tlačítko na přepnutí do Project view.....	34
Obr. 13 Projektové zobrazení – Project view.....	34
Obr. 14 Vytvoření projektu.....	35
Obr. 15 Přidání PLC S7 - 1200 s CPU 1214C DC/DC/DC z katalogového listu.....	35
Obr. 16 Okno konfigurace hardwaru.....	36
Obr. 17 Nastavení IP adresy.....	37
Obr. 18 Přidání programového bloku.....	37
Obr. 19 Volba bloku a programovacího jazyka.....	38
Obr. 20 Pojmenování bloku.....	38
Obr. 21 Instrukce pro vytváření programu.....	39
Obr. 22 Přiřazení symbolických adres.....	39
Obr. 23 Příklad programu v LAD.....	40
Obr. 24 Volání funkce z OB1.....	40
Obr. 25 Kompilace hardwaru PLC.....	41
Obr. 26 Informace o úspěšném zkompilování hardwarové konfigurace.....	41
Obr. 27 Nastavení komunikace pro nahrání projektu do PLC.....	42
Obr. 28 Nahrání projektu do zvoleného zařízení.....	42
Obr. 29 Potvrzení nahrání konfigurace a programu do PLC.....	43
Obr. 30 Přepnutí režimu PLC do režimu „RUN“ po nahrání projektu.....	43
Obr. 31 Ikona „Monitoring on/off“ pro přepnutí do monitorovacího režimu.....	44
Obr. 32 Ukázka monitorovacího režimu FC bloku.....	44

Obr. 33 Příklad nadefinování symbolických adres .....	47
Obr. 34 Ukázka řešení detekce stavu jednotlivých pater v LAD.....	47
Obr. 35 Ukázka řešení detekce stavu jednotlivých pater v FBD .....	48
Obr. 36 Ukázka řešení detekce stavu jednotlivých pater v SCL .....	48
Obr. 37 Ukázka použití čítače CTU v řešení úlohy .....	50
Obr. 38 Symbolické adresy detektorů a zařízení pro místnost 1 .....	53
Obr. 39 Symbolické adresy detektorů a zařízení pro místnost 2 .....	53
Obr. 40 Ukázka řešení definice parametrů FC bloku „místnost“ .....	53
Obr. 41 Ukázka řešení ukládání času otevření dveří .....	54
Obr. 42 Nastavení IP adresy Acces pointu .....	57
Obr. 43 Ukázka řešení definice datového bloku pro zaslání e-mailu .....	57
Obr. 44 Nastavení IP adresy přímo v panelu HMI KTP700 Basic.....	58
Obr. 45 Hláška pro přepsání projektu .....	59
Obr. 46 Ukázka možností animací grafických prvků .....	61
Obr. 47 Návrh grafické vizualizace úvodní obrazovky .....	61
Obr. 48 Aktivace zobrazení obrazovky „Stav zastřežení“ .....	62
Obr. 49 Volba úvodní obrazovky po zapnutí HMI paneulu .....	62
Obr. 50 Zvolení módu „I/O field“ pro zobrazení času otevření dveří .....	63
Obr. 51 Přiřazení „Template“ k obrazovce „Stav detektorů“ .....	63
Obr. 52 Funkce „Visibility“ .....	64

**SEZNAM TABULEK**

Tab. 1 Bitově zaměřené datové typy - část první [16] .....	31
Tab. 2 Bitově zaměřené datové typy - část druhá [16] .....	31
Tab. 3 Celočíselné datové typy [16] .....	31
Tab. 4 Datové typy pro desetinné operace [16] .....	32
Tab. 5 Časové datové typy [16] .....	32
Tab. 6 Datové typy Char a String [16].....	32

## SEZNAM PŘÍLOH

P I            Projekty vytvořené v TIA Portalu – řešení úloh z praktické části

## **PŘÍLOHA P I: PROJEKTY VYTVOŘENÉ V TIA PORTALU – ŘEŠENÍ ÚLOH Z PRAKTICKÉ ČÁSTI**

Tato příloha obsahuje řešení všech úloh z praktické části této bakalářské části. Soubory jsou uloženy na DVD, které je přiloženo k této bakalářské práci.

1. Archivovaný projekt s řešením úlohy Logické operace v různých programovacích jazycích
2. Archivovaný projekt s řešením úlohy Použití časovačů a čítačů
3. Archivovaný projekt s řešením úlohy Blokované parametry u FC a FB bloků
4. Archivovaný projekt s řešením úlohy Poslání e-mailu v případě vyhlášení poplachu
5. Archivovaný projekt s řešením úlohy Propojení PLC s HMI a vytvoření vizualizace