

**UNIVERZITA PARDUBICE**  
Fakulta elektrotechniky a informatiky

**REGULOVANÉ POHONY ŘÍZENÉ POMOCÍ KNIHOVNÍCH FUNKCÍ  
OPENPLC PRO S7-1500**

Daniel Prchal

Bakalářská práce  
2015

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2014/2015

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Daniel Prchal**  
Osobní číslo: **I12280**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Řízení procesů**  
Název tématu: **Regulované pohony řízené pomocí knihovních funkcí OpenPLC pro S7-1500**  
Zadávací katedra: **Katedra řízení procesů**

### Zásady pro vypracování:

#### Cíl:

Cílem práce je prostudovat a popsat knihovní funkce dostupné pro S7-1500, které jsou určeny pro řízení regulovaných elektrických pohonů. Bude vytvořen program v PLC a HMI (operační panel SIEMENS), který umožní základní operace s regulovaným elektrickým pohonem SIEMENS (jog, polohování, referencování, diagnostika). Vytvořený program bude sloužit jako vzor pro další rutinní práci.

#### Obsah teoretické části:

1. Nastudovat a popsat prostředí TIA portal V13
2. Nastudovat a popsat HW platformu S7-1500
3. Nastudovat a popsat knihovnu pro práci s regulovanými elektrickými pohony

#### Obsah implementační části:

1. Vytvořit program PLC, HMI a nastavit pohon tak aby z HMI bylo možné ovládat pohon (zapnutí/ vypnutí, jog, referencování, polohování - relativní, absolutní, diagnostika)
2. Zhodnotit pro jaké typy úloh je toto řešení s knihovnou OpenPLC vhodné. Prostředky: Testovací PLC, pohon i HMI a SW pro vývoj bude zapůjčen f. COMPAS na dobu vývoje SW.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**ŠMEJKAL, Ladislav. PLC a automatizace 2. 1. vyd. Praha: BEN - technická literatura, 2005, 208 s. ISBN 80-730-0087-3.**

**SIEMENS. SIMATIC S7-1500: S7-1500 Motion Control [online]. Siemens, 2013 [cit. 2014-10-30]. ISBN A5E03982862-02. Dostupné z:**

**[https://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started\\_simatic-s7-1500/documents/EN/smc\\_en.pdf](https://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/documents/EN/smc_en.pdf)**

**SIEMENS. SIMATIC S7-1500: Getting Started [online]. Siemens, 2014 [cit. 2014-10-30]. ISBN A5E03981761-AC. Dostupné z:**

**[http://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started\\_simatic-s7-1500/documents/EN/software\\_complete\\_en.pdf](http://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/documents/EN/software_complete_en.pdf)**

Vedoucí bakalářské práce:

**Ing. Daniel Honc, Ph.D.**

Katedra řízení procesů

Datum zadání bakalářské práce:

**2. prosince 2014**

Termín odevzdání bakalářské práce:

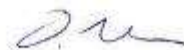
**11. května 2015**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Daniel Honc, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2015

---

## **Prohlášení**

Prohlašuji:

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 11. 5.2015

Daniel Prchal

### **Poděkování**

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Danielu Honcovi, Ph.D. za vedení práce, cenné rady a připomínky ke zpracování práce a pomoc s realizací hardwarové části modelu. Dále Ing. Zbyňku Bezchlebovi a firmě COMPAS za pomoc s realizací bakalářské práce. V neposlední řadě bych chtěl také poděkovat mé rodině a přátelům za podporu během studia.

V Pardubicích dne 11. 5.2015

Daniel Prchal

## **ANOTACE**

Práce je věnována problematice regulace pohonu, pomocí programovatelných průmyslových - automatů rodiny S7-1500. Byl proveden základní popis programového prostředí TIA Portal V13, popis programovatelných automatů S7-1500 a instrukce pro regulaci. V praktické části je vytvořen univerzální program pro regulaci pohonu.

## **KLÍČOVÁ SLOVA**

Regulovaný pohon, programovatelný průmyslový automat, TIA Portal, S7-1500, OpenPLC

## **TITLE**

OpenPLC Library Functions Motion Control for S7-1500

## **ANNOTATION**

Work is devoted to the issue of control of propulsion, using a programmable industrial -family S7-1500 machines. He was made a basic description of the programming environment of the TIA Portal V13, a description of the S7 programmable logic controllers-1500 and the instructions for the control. In the practical part of the universal program is created for the control of the drive.

## **KEYWORDS**

Regulated drive, programmable industrial machine, TIA Portal, S7-1500, OpenPLC

## Obsah

Seznam zkratk	9
Seznam ilustrací	10
Seznam tabulek	12
ÚVOD	14
1 TEORETICKÁ ČÁST	15
1.1 ZÁKLADNÍ POJMY	15
1.1.1 Programovatelný automat	15
1.1.2 Frekvenční měnič	16
1.2 TIA Portal V13	18
1.2.1 Uživatelské rozhraní	20
1.2.2 Nápověda	26
1.2.3 Knihovní pohled	28
1.3 SIMATIC S7-1500	29
1.3.1 Rodiny CPU S7-1500	31
1.3.2 Moduly k SIMATIC S7-1500	31
1.4 KNIHOVNA PRO PRÁCI S REGULOVANÝMI POHONY	34
1.4.1 Knihovna instrukcí	34
1.4.2 Obecné instrukční parametry	35
1.4.3 Dynamické parametry pohybu	37
1.4.4 Povolovací instrukce MC_Power	39
1.4.5 Základní instrukce	41
1.4.6 Pohybové instrukce	43
1.4.7 Vkládání instrukcí do projektu	46
2 PRAKTICKÁ ČÁST	48
2.1 VYTVOŘENÍ A NASTAVATENÍ PROJEKTU V TIA PORTAL	48
2.2 HARDWARE	48
2.3 PROGRAM PRO PLC	50
2.3.1 Programová struktura	50
2.3.2 Konfigurace technologického objektu	51
2.3.3 Hlavní program MAIN	53
2.3.4 Vstupní blok 1. IN	54
2.3.5 Blok chyb 2. ERROR	55

2.3.6	Podmínkový blok 3. IF.....	56
2.3.7	Blok manuálního řízení 4. MAN.....	57
2.3.8	Blok automatické regulace 5 . AUT.....	58
2.3.9	Výstupní blok 6. OUT.....	59
2.3.10	Data programu.....	59
2.3.11	Data vstupně/výstupní.....	59
2.4	VIZUALIZACE.....	60
2.4.1	Šablony.....	60
2.4.2	START obrazovka.....	61
2.4.3	Parametrické a diagnostické obrazovky.....	62
2.4.4	Systémové obrazovky.....	63
2.4.5	Obrazovky chyb.....	63
3	ZHODNOCENÍ.....	64
4	ZÁVĚR.....	66
	LITERATURA .....	67



## Seznam zkratek

CPU	Central Processing Unit
PLC	Programmable Logic Controller
LAD	Ladder Logic
FBD	Function Block Diagram
STL	Statement list
SCL	Structured Control Language
TO	Technology object
IP	Internet protocol
PLC	Programmable Logic Controller
PAC	Programmable Automation Controller
PA	Programovatelný automat
SPS	Speicher Programmierbare Steuerung

## Seznam ilustrací

Obr. 1.1 – Blokové schéma modulárního PLC .....	16
Obr. 1.2 – Měnič s mezi stupněm.....	17
Obr. 1.3 – Charakteristika motoru s kmitočtově řízeným měničem.....	17
Obr. 1.4 – Produkty SIEMENS.....	19
Obr. 1.5 – Projektový pohled.....	21
Obr. 1.6 – Otázkové karty.....	23
Obr. 1.7 – Portálový pohled .....	25
Obr. 1.8 – Knihovní pohled .....	26
Obr. 1.9 – Nápopěda s manuály pomocí prohlížeče.....	27
Obr. 1.10 – Informační systém programu TIA Portal .....	28
Obr. 1.11 – Nastavení programu TIA Portal .....	29
Obr. 1.12 – PLC SIMATIC S7-1500 .....	30
Obr. 1.13 – Možnost hardwarové konfigurace .....	31
Obr. 1.14 – Knihovna instrukcí.....	34
Obr. 1.15 – Nastavení parametrů přímo.....	34
Obr. 1.16 – Nastavení parametrů nepřímo.....	35
Obr. 1.17 – MC_Power v LAD .....	39
Obr. 1.18 – MC_Power v SCL .....	39
Obr. 1.19 – MC_Home v LAD .....	41
Obr. 1.20 – MC_Home v SCL .....	41
Obr. 1.21 – MC_Halt v LAD.....	43
Obr. 1.22 – MC_Halt v SCL .....	43
Obr. 1.23 – MC_Reset v LAD .....	43
Obr. 1.24 – MC_Reset v SCL .....	43
Obr. 1.25 – MC_MoveJog v LAD.....	44
Obr. 1.26 – MC_MoveJog v SCL .....	44
Obr. 1.27 – MC_MoveVelocity v LAD .....	45
Obr. 1.28 – MC_MoveVelocity v SCL .....	45
Obr. 1.29 – MC_MoveAbsolute v LAD .....	45
Obr. 1.30 – MC_MoveAbsolute v SCL .....	45
Obr. 1.31 – MC_MoveRelative v LAD .....	46
Obr. 1.32 – MC_MoveRelative v SCL .....	46

Obr. 2.1 – Hardwarová konfigurace.....	50
Obr. 2.2 – Programové bloky.....	50
Obr. 2.3 – Přidání nového objektu.....	52
Obr. 2.4 – Nastavení technologického objektu.....	53
Obr. 2.5 – Network 1 v cyklickém bloku OB1.....	54
Obr. 2.6 – Network 1 ve funkci 1. IN.....	54
Obr. 2.7 – Network 1 ve funkci 7. IN.....	55
Obr. 2.8 – Část network 1 ve funkci 2. ERROR.....	56
Obr. 2.9 – Část network 2 ve funkci 2. ERROR.....	56
Obr. 2.10 – Ukázka z IF funkce.....	57
Obr. 2.11 – Network 1 ve funkci 4. MAN.....	57
Obr. 2.12 – Network 3 ve funkci 4. MAN.....	58
Obr. 2.13 – Network 1 ve funkci 5. AUT.....	58
Obr. 2.14 – Datový blok DB1.....	59
Obr. 2.15 – Šablona Template_1.....	61
Obr. 2.16 – Parametrická obrazovka.....	62
Obr. 2.17 – Diagnostická obrazovka funkce MC_Power.....	62
Obr. 2.18 – Systémová obrazovka Project information.....	63
Obr. 2.19 – Obrazovka chyb.....	63

## **Seznam tabulek**

Tab. 1.1 – Požadavky na hardware .....	16
Tab. 1.2 – Tabulka chyb pohybových instrukcí.....	35

## ÚVOD

Úkolem této práce je vytvoření programu, který pomůže vysvětlit problematiku regulace pomocí programovatelných automatů S7-1500 a bude sloužit jako vzor pro další práci. K regulaci je využit hardware a software od společnosti SIEMENS. Softwarem SIEMENS TIA PORTAL V13 a hardwarem SIMATIC S7-1500, které vydává a vyrábí společnost SIMENS. Hlavním úkolem je vytvořit univerzální program pro regulaci pohonu. Vytvořený program je vizualizován pro panely SIMENS 7“ Comfort. Panely Comfort mají výhodu dotykového ovládání a integrované funkce pro řízení programů.

První kapitola se zabývá teoretickým rozbohem dané problematiky. Tedy základními pojmy práce jako jsou frekvenční měnič a programovatelný automat. Druhá podkapitola popisuje inženýrské programovací prostředí TIA PORTAL a základní práci v tomto programu. Zaměření této kapitoly je věnováno spíše na programové uživatelské rozhraní a na základní prostředky práce v programu jako je nastavení programu, programovou nápovědu a základní pohyb v rozhraní. Třetí podkapitola popisuje řídicí hardware S7-1500 včetně popisu jeho využití a základních zapojení a možnosti modulace tohoto zařízení. V této kapitole jsou popsány základní možnosti připojení modulů CPU S7-1500, základní typy CPU a vlastnosti této hardwarové platformy. Čtvrtá část teoretické problematiky popisuje knihovnu pro práci s regulovanými pohony a využití těchto instrukcí pro regulaci a řízení pohonu. Obsahuje popis základního vkládání instrukcí do programu, parametry instrukcí a popis funkce jednotlivých instrukcí.

Praktická část práce se zabývá realizací programu. Práce popisuje tvorbu programu pro univerzální aplikaci řízení pohonu, tvorbu a funkce vizualizačních obrazovek a základní interakce mezi nimi. K tvorbě tohoto programu je využit výše zmíněný software TIA Portal V13. Výsledný kód je určen pro nejnovější hardwarovou platformu S7-1500.

Poslední kapitola popisuje program jako celek včetně jeho funkčnosti. Následná fyzická realizace je provedena pomocí hardwaru vypůjčeného firmou COMPAS.

Tomuto tématu se věnuji hlavně z důvodu mého osobního zájmu o problematiku průmyslové automatizace, včetně využití regulovaných pohonů pro řízení pohybu. Očekávaným výsledkem této práce je univerzální program, který je možno dále modifikovat a využít případně pro další práce nebo jako ukázkový program.

# 1 TEORETICKÁ ČÁST

## 1.1 ZÁKLADNÍ POJMY

### 2.2.1 Programovatelný automat

Programovatelný automat je nejčastěji označován zkratkami, PLC, SPS nebo PA. Nejmodernější označení používá PAC, které v překladu znamená programovatelný automatizační kontrolér. Každá z těchto zkratek je používána v závislosti na jazyku autora. Programovatelný automat je řídicí programovatelný systém orientovaný na zpracování bitových a bytových operací. A logické řízení pomocí povelů. Automaty jsou většinou využívány jako modulární systémy, kdy se připojují jednotlivé módy dle požadované aplikace. Vývoj těchto průmyslových zařízení započal již v roce 1969 prvním programovatelným automatem od firmy Allen Bradley, která vyrobila první programovatelný automat. Tento automat sloužil k nahrazování relé, časových relé a mechanických počítadel (Tkotz, 2006, s. 406-420; Zezulka et al., 2003, s. 7-16).

Jejich rychlý vývoj byl příčinou, že se v 80. letech staly nejvýznamnějším řídicím prostředkem výrobních linek. V této době se používaly jako HW jádro bitové procesory. Velkou výhodou těchto zařízení je vysoká spolehlivost, jednodušší rozdělení programové struktury na jednodušší celky a nízké náklady na kabeláž. Tyto parametry se ukázaly natolik výhodné, že dnes již automaty zcela vytlačily dříve používaná relé pro řízení průmyslových linek. Ale k hlavním přednostem patří zcela jistě usnadnění práce, které plyne z faktu, že již není nutné vytvářet řídicí hardware. Stačí pouze vybrat zvolené PLC a k němu potřebné moduly. Následně vytvořit řídicí program. Ovšem tyto průmyslové automaty mají i celou řadu nevýhod. K nevýhodám patří nižší programátorský komfort, než u počítačů, nezbytná strukturální architektura u větších celků a horší výkon než u průmyslových počítačů. Hlavní rozdíl oproti průmyslovým počítačům je ten, že program automaty vykonávají cyklicky, jsou-li nastaveny v základním režimu. Programovatelný automat cyklicky vykonává program tak, že přečte aktuální stav vstupů, zpracuje hodnoty a reaguje výstupem na předešlé kroky (Zezulka et al., 2003, s. 7-16).

Dnes je několik základních typů PLC. Typy jsou rozlišovány dle způsobu rozšiřování o moduly. Prvním typem je modulární PLC. Modulární moduly mají jednu společnou sběrnici, na kterou se instalují jednotlivé moduly zařízení. Druhým typem je kompaktní PLC. Kompaktní provedení má zařízení v jednom pevném pouzdře a rozšiřování lze provádět pouze pomocí systémových konektorů. Poslední provedení je kazetové PLC. Toto provedení pracuje na principu rozšiřovacích karet, které lze připojovat na dané PLC.



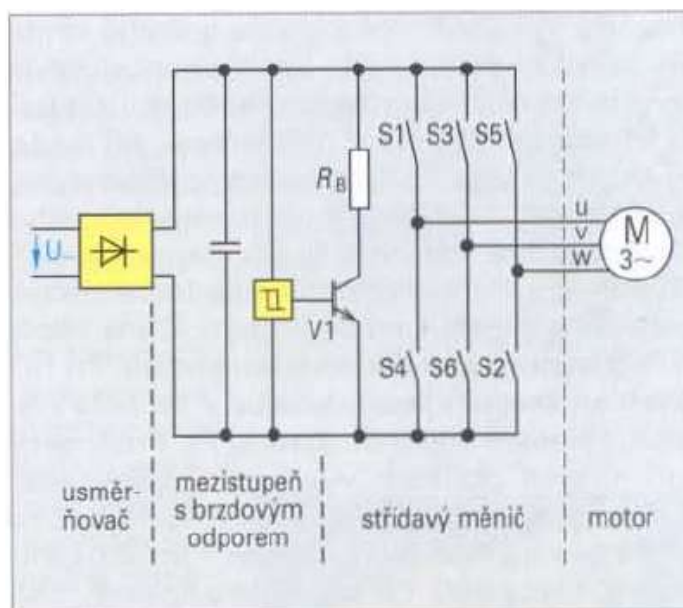
Obr. 1.1 – Blokové schéma modulárního PLC (Zezulka et al., 2003, s. 7-16)

K hlavním výrobcům průmyslových automatů dnes patří výrobci jako Siemens, Festo, ABB, Teco, Allen Bradley a mnoho dalších. U většiny těchto výrobců se způsoby programování sjednocují dle mezinárodního standartu IEC 1131-3. Na obr. 1.1 je zobrazeno blokové schéma standartního modulárního PLC. Z tohoto schématu je zajímavé, že je velmi podobné blokovým schématům mikropočítačů. Například mikropočítač AVR ATmega32 pracuje velmi podobně jako průmyslový automat, ale tento mikropočítač nezvládá úlohy typické pro průmyslové automaty.

Základem každého průmyslového automatu je CPU. CPU je základní modul, který provádí veškeré výpočetní operace. Tento základní modul je dále vybaven komunikačním rozhraním, pomocí něhož lze komunikovat s ostatními jednotkami nebo slouží pouze pro programování modulu. Většina firem zabývajících se průmyslovými automaty, vyrábí rodinu podobných automatů, které se liší výkonem, pamětí a uživatelským rozhraním. CPU pracuje hlavně s 16 nebo 32 bitovým slovem. Dále obsahuje bitové registry označované jako znaky a operační paměť pro ukládání dat a práci s programem. Další moduly se připojují k CPU dle cílené aplikace (Tkotz, 2006, s. 406-420; Zezulka et al., 2003, s. 7-16).

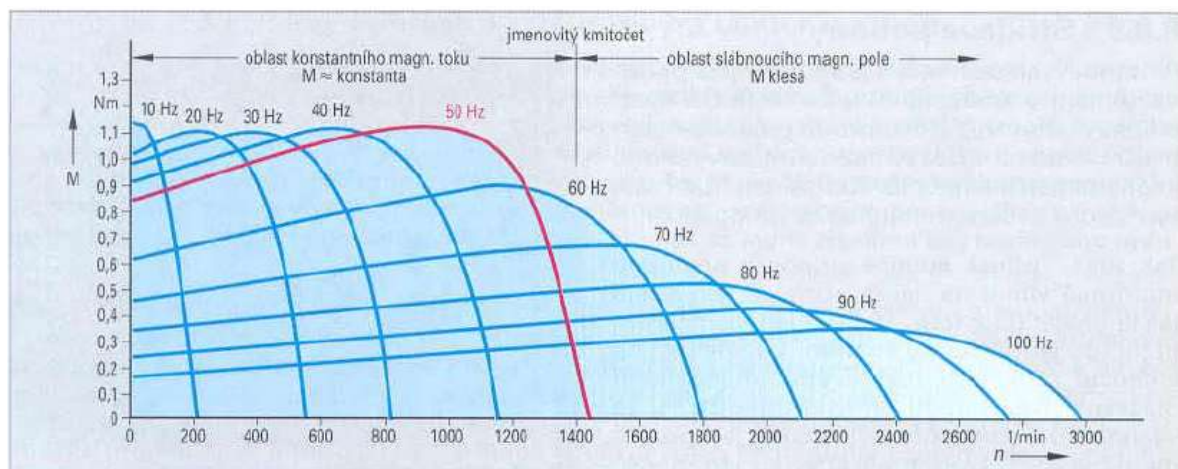
### 1.1.2 Frekvenční měnič

Využívají se většinou se střídavými pohony jako regulátory otáček. Mění současně vstupní napětí a kmitočet na jiné hodnoty kmitočtu a napětí na výstupu. Toto výstupní napětí je následně použito na asynchronním motoru. Jak vyplývá z definice asynchronních motorů, tak se změnou frekvence se mění otáčky asynchronního motoru. Existují dva základní typy frekvenčních měničů. Prvním typem je přímý frekvenční měnič. Ten připojuje postupně motorové vinutí na napětí různých fází a dělí tím kmitočet. Tento typ má horší měnitelnost výstupního kmitočtu (Tkotz, 2006, s. 211-213).



Obr. 1.2 – Měnič s mezi stupněm (Tkotz, 2006, s. 212)

Druhý typ měniče je frekvenční měnič kmitočtu s mezistupněm. Zde se využívá ke změně kmitočtu převod na stejnosměrné napětí, které následně pulzní střídač mění na pulzní napětí o požadované frekvenci. S touto změnou pulzního napětí ale výstupní napětí může dosáhnout maximálně úrovně stejnosměrného napětí, které je získáno mezistupněm. U motorů do 30 kW výkonu je při brždění přebytečná energie přeměněna na teplo v brzdných odporech. Na obr. 1.2 jsou brzdné odpory označené  $R_B$ . Měniče s mezistupněm umožňují funkci až do dvojnásobku síťové frekvence, která je na vstupu frekvenčního měniče. Se změnou kmitočtu se přímo úměrně mění odpor pohonu  $X_L$ , který lze zjednodušit pomocí RL obvodu. To má samozřejmě vliv na celkový proud protékajícím obvodem. Z toho plyne, že se změnou frekvence se mění i proud protékající obvodem (Tkotz, 2006, s. 211-213).



Obr. 1.3 – Charakteristika motoru s kmitočtově řízeným měničem (Tkotz, 2006, s. 213)



## 1.2 TIA PORTAL V13

TIA PORTAL je anglická zkratka automatizačního softwaru, která v překladu znamená portál kompletně integrované automatizace. Tento pojem zahrnuje jedno programovací prostředí pomocí kterého je možné provádět programování PLC, vizualizaci aplikací, hardwarové konfigurace zařízení, přímo ovládat zařízení a diagnostikovat zařízení. Program kombinuje funkce několika starších programů od firmy SIEMENS do jediného univerzálního prostředí určeného pro všechny programové úkony a hardwarové aplikace v průmyslové automatizaci.

Optimální požadavky na funkci programu jsou uvedeny v tab. 1.1. Starší programy, které kombinuje jsou STEP 7, WinCC, STARTER a další. STEP 5.x je velmi oblíbený nástroj pro tvorbu programů a konfigurace hardwaru SIMENS, případně hardwaru od jiných výrobců. WinCC slouží pro vizualizaci aplikací pro různé platformy zařízení. Tyto platformy začínají od HMI panelů a končí u vizualizací přímo na počítači. STARTER se využívá pro konfiguraci frekvenčních měničů SINAMICS od společnosti SIMENS. TIA Portal V13 oproti starším programům nezahrnuje starší verze hardwaru, protože je primárně určen pro nové projekty. Z tohoto důvodů může nastat problém při práci se starším hardwarem.

Tab. 1.1 – Požadavky na hardware (Siemens, 2014a)

Požadavky	Doporučené
Operační systém	64 bitový
Procesor	Core i5-3320M; 3,3 GHz
RAM	8 GB

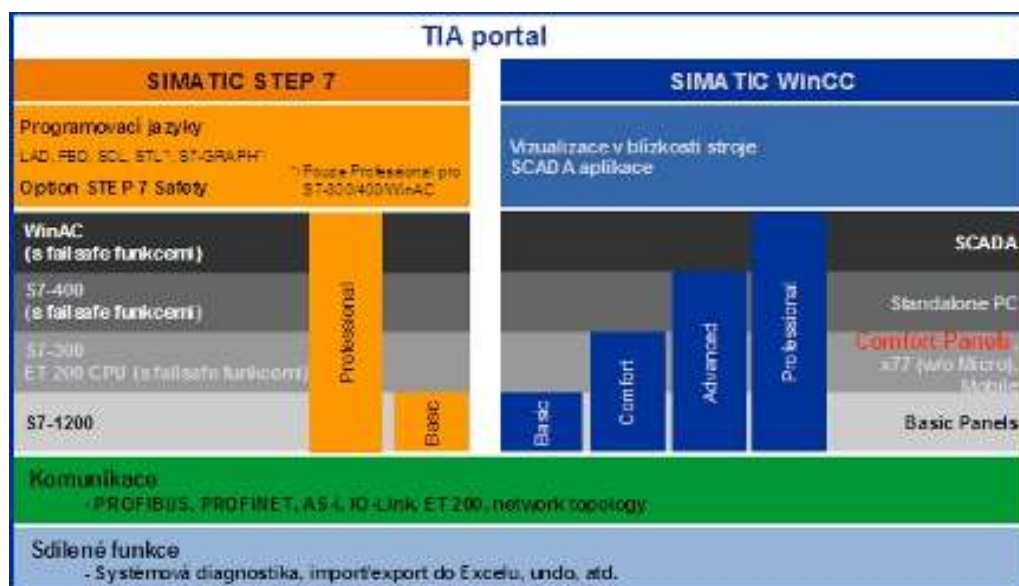
K nejdůležitějším činnostem patří vytváření programů pro PLC pomocí tohoto programu a k tomu lze využít programovací jazyky jako je LAD, FBD, STL, SCL a S7-GRAPH. Vytvořené programy lze nahrát pomocí tohoto programu do zařízení nebo již vytvořené programy z těchto zařízení stahovat. Tyto programy mohou zpracovávat nebo předávat data webovému serveru, který podporují některé typy PLC. Hlavně novější typy zařízení řady S7-1500 a S7-1200. Webový server na PLC pracuje na principu vytvoření webové stránky pomocí html, css a php jazyků v libovolném programu pro tvorbu webových stránek. K těmto validním webovým stránkám jsou následně přidány v komentářích prvky pro komunikaci s programem v PLC. Hotové webové stránky jsou pomocí nástrojů programu TIA Portal V13 transformovány do datových bloků v programové struktuře projektu TIA Portal V13. Vytvořené programy mohou využívat data z jiných zařízení připojené do libovolné síťové struktury, kde existuje mnoho různých možností konfigurací zařízení.

Další velmi užitečnou vlastností je diagnostika zařízení. Sledování aktuálního průběhu programu či aktuálních programových dat. Sledování a diagnostika jsou možné, pokud je počítač s programem TIA Portal propojen přes rozhraní PROFINET, PROFIBUS nebo Wi-Fi se sledovanými zařízeními.

Vizualizaci pomocí tohoto softwaru pro strojově orientované operátorské řízení a monitorování dat na SIMATIC Panely, SIMATIC průmyslových PC, standardních PC s programem WinCC Runtime Advanced nebo SCADA Systémů. Do této vizualizace lze zařadit i výše zmíněnou vizualizaci pomocí webových stránek. Zde je velmi výhodné použít atributy ulehčující práci a činnost uživatelů. K těmto ulehčujícím atributům primárně patří recepturní systém, alarmy, varování a správa uživatelů. Při vizualizaci lze využívat data uložená ve vizualizovaném systému nebo data z PLC. Výhodou dat z PLC je, že jsou zpětně dostupná i při výpadku napájení či poruchy na zařízení. Oproti tomu data pouze ve vizualizačním zařízení jsou při výpadku napájení či poruše ztracena.

Další velmi užitečnou vlastností tohoto programu je možnost konfigurace PLC panelů a síťových zařízení pro síť PROFIBUS, PROFINET, AS-i, IO-Link, ET 200. Díky tomu lze zařízení propojit pomocí jedné sítě a tím zjednodušit údržbu a správu několika zařízení z jednoho místa.

Tato nová verze programu umožňuje sdílenou práci na projektu pomocí funkce *Delta compile* (kompilace změn v programu) nebo *PLC Proxy* (rozdělení projektu do několika částí pro práci několika programátorů- zejména pro HMI panely).



Obr. 1.4 – Produkty SIEMENS (Siemens, 2015)

TIA PORTAL se při instalaci skládá z programových částí, které určují podporovaný hardware a programové funkce. Základní programové části jsou STEP 7 a WinCC. STEP7

je určen pro tvorbu programů pro HMI panely. Tento STEP 7 má dvě základní verze, které jsou Basic a Professional. První verze STEP 7 Basic podporuje práci s rodinou PLC S7-1200. Druhá verze STEP 7 Professional podporuje práci s rodinami kontrolérů S7-1200, S7-1500, S7-300/400 a WinAC. WinCC slouží pro základní vizualizaci aplikací. WinCC má čtyři základní verze, které jsou Basic, Comfort, Advanced a Professional. WinCC Basic je určen pouze pro práci s Basic Panely. WinCC Comfort je určen pro konfiguraci všech typů HMI panelů. WinCC Advanced pro konfiguraci všech typů panelů a počítačů s WinCC Runtime Advanced vizualizačním softwarem. WinCC Professional oproti předchozí verzi Advanced podporuje navíc i SCADA systémy WinCC Runtime Professional. K některým z těchto verzí existují další verze, jenž rozvíjejí další prvky programu TIA PORTAL V13, jako jsou třeba PID Professional a mnoho dalších (Siemens, 2014a).

Pro práci se staršími verzemi programu pro PLC nebo staršími vizualizacemi zde existuje nástroj migrace. Tento nástroj převede starší verzi programu na verzi programu pro TIA PORTAL V13. Při převodu mohou nastat problémy s hardwarem zařízení, který již program TIA Portal V13 nemusí podporovat. V případě těchto problémů je nezbytné buď přidat starší zařízení, jsou-li dostupná, do konfigurace programu TIA Portal nebo historické zařízení nahradit novým zařízením. Zde ovšem vzniká problém s nutností upravit program pro nové zařízení.

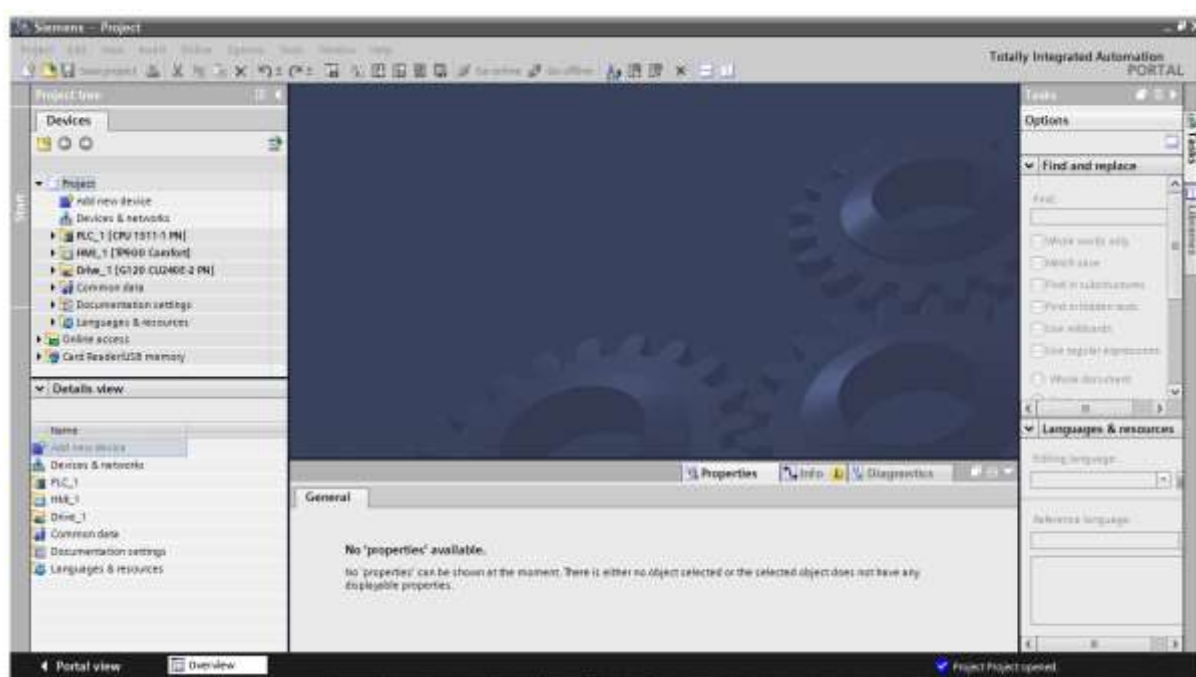
### **1.2.1 Uživatelské rozhraní**

Uživatelské rozhraní programu obsahuje oproti starším verzím programu nové grafické uživatelské rozhraní, které již zahrnuje většinu moderních grafických prvků nebo moderních animací. Toto nové grafické rozhraní vychází ze starších verzí programu WinCC, který má velmi podobné uživatelské rozhraní.

Uživatelské rozhraní programu TIA PORTAL se skládá ze tří základních pohledů, ve kterých uživatel pracuje: portálový, knihovní a projektový. Nejdůležitějším z nich je projektový pohled, který je využíván nejčastěji a je v něm vykonávána většina práce v programu. Pro změnu zobrazovaného pohledu při startu programu lze zvolit v nastavení programu TIA Portal požadovaný pohled. Jednotlivé rozložení komponent v pohledu lze měnit a přizpůsobovat požadavkům uživatele programu. Mezi jednotlivými pohledy lze snadno přepínat pomocí tlačítek umístěných napravo dole a označených *Project view* nebo *Portal view*. Další možnost změny pohledu lze pomocí záložky View na menu lišty a zde vybrat funkci *Go*

to *portal view*. Pohled se také může měnit automaticky. Například při výběru některého úkolu v portálovém pohledu (Siemens, 2014a).

Projektový pohled je strukturovaný pohled všech komponent v projektu. Tento pohled se skládá z několika základních elementů. Těmito základními elementy jsou titulní lišta, menu lišta, nástrojová lišta, projektový strom, referenční projekt, detailní pohled, pracovní oblast, děliče, okno inspektoru, tlačítko pro změnu na projektový pohled, lišta editoru, stavová lišta a karta otázek. Rozložení, velikost a umístění jednotlivých prvků se může dle libosti uživatele měnit pomocí myši tak, že se drží kraj elementu a tahem lze libovolně měnit velikost jednotlivých elementů. Změnit pozici elementu jde pomocí stisku na listu daného okna a to táhnout na požadované místo (Siemens, 2014a).



Obr. 1.5 – Projektový pohled

Jméno projektu je zobrazeno v titulní liště programu vedle loga a názvu společnosti SIMENS.

Lišta menu ležící pod titulní lištou obsahuje všechny příkazy, které jsou možné pro práci v programu. Tato lišta obsahuje záložky *Project*, *Edit*, *View*, *Insert*, *Online*, *Tools*, *Window* a *Help*. V záložce *Project* je možno vytvářet, otevírat, ukončovat, mazat, ukládat nebo převádět projekt ze starších verzí programu na aktuální používanou verzi TIA Portal V13. Dále tato záložka obsahuje možnosti pro archivaci nebo rearchivaci projektu, což je vhodné pro vytváření či obnovování programových záloh. Na závěr tato záložka obsahuje položky pro ukončení programu, práci s hardwarovými kartami počítače a tisk. Záložka *Edit* obsahuje funkce pro krok vpřed a vzad, vložení, vyjmutí a vymazání označeného objektu, výběr všech položek v aktuálně

otevřeném editoru. K dalším vlastnostem patří možnost kompilace programu, čehož se využívá před nahráním programu do PLC nebo pro odhalení syntaktických chyb v programu. Administrátorské funkce v této záložce jsou skok na a nahrazení textu v dokumentu. Záložka *View* se využívá pro změnu pohledu na portálový pohled nebo výběr zobrazovaných oken nebo textu v projektovém okně. Záložka *Insert* lze využít pro funkce ke vkládání nových zařízení do projektu nebo v případě LAD editoru vložit nový network do editoru. Záložka *Online* obsahuje funkce pro přepínání mezi online a off-line módu programu. Při online módu je nutné mít připojené zařízení k počítači a poté je možné sledovat online stav zařízení, aktuální stav programu a dat. Online mód je standardní mód programu, ve kterém běží program v normálním režimu. Dále v této záložce je funkce pro simulaci PLC programu na počítači, k tomuto je nezbytné mít nainstalovaný programový simulátor S7-PLSIM. Tato záložka poskytuje funkce pro nahrání programu do PLC nebo stažení programu z PLC. Další funkce umožňují práci s HMI panely, spuštění, případně zastavení CPU zařízení a další funkce pro simulaci a online úpravu programu. Záložka *Options* obsahuje funkce pro nastavení programu TIA Portal a projektu, informace o nainstalovaných součástech programu TIA Portal V13. Důležitou funkcí v záložce *Options*, je vkládání GSD souboru, využívané jako hardwarová konfigurace zařízení od jiných výrobců, kdy tyto zařízení nejsou ve stávajícím programovém katalogu programu TIA Portal V13 a proto je nutné je vložit do programu, aby s tímto zařízením se mohlo dále pracovat. GSD soubory by měl vydávat výrobce daného hardwaru, proto v případě neznámého hardwaru je nutné navštívit stránky výrobce daného hardwaru a zde by měli být k dostání na přiloženém paměťovém médiu. K posledním funkcím v této záložce patří zobrazení referenčních textů a práce s bloky a globálními knihovny. Záložka *Tools* obsahuje křížové reference, import a export projektového textu, funkce pro práci s projektovým textem a jazyky projektu. Záložka *Window* obsahuje funkce pro práci s oknem projektového pohledu. Zde je možné minimalizovat okno, zavřít, vkládat dvě okna horizontálně nebo vertikálně. Toto nově nastavené rozložení lze uložit nebo nahrát na původní programové nastavení. Záložka *Help* obsahuje funkce pro zobrazení informací o programové nápovědě a nainstalovaných produktech.

Pod lištou menu se nachází nástrojová lišta a v ní umístěné nejčastěji používané funkce. Tyto základní funkce prakticky stačí pro základní práci v programu. A proto není potřeba příliš hledat funkce v menu lišta.

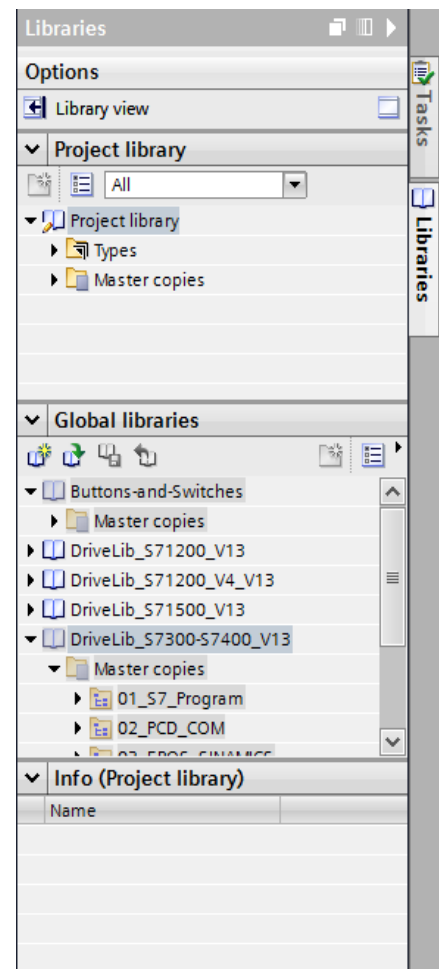
Vlevo pod menu lišta se nachází projektový strom. Tento strom lze srolovat a opětovně obnovit stisknutím na tlačítko *Start*, nacházející se zcela vlevo. Dle mých zkušeností je projektový strom nejzákladnější element celého pohledu, protože obsahuje výčet všech

komponent projektu a projektových dat. Projektový strom je rozdělen na dvě části. První část se nazývá *Devices* a druhá část *Details view*. *Devices* slouží jako stromový seznam prvků projektu. Aktuální otevřený projekt, *Online acces* a *Card Reader/USB memory* jsou kořenové složky v *Devices*. Každé hardwarové zařízení v projektu je zde reprezentováno vlastní složkou v hlavní složce projektu. Tato hlavní složka projektu je pojmenovaná dle aktuálně otevřeného projektu. V dané složce hardwarového zařízení jsou dostupné všechny akce pro dané hardwarové zařízení. Složka *Online acces* obsahuje dostupné funkce pro práci s online přístupem k zařízení. Poslední je kořenová složka *Card Reader/USB memory*. Jsou zde umístěny funkce z důvodu možnosti práce s přenosnými paměťovými zařízení využívanými v jednotlivých hardwarových zařízeních. Oproti tomu část *Details view* obsahuje seznam hardwarových zařízení v projektu (Siemens, 2014a).

Napravo od *Project tree* se nachází *Work area*. Tato oblast slouží pro jednotlivé editory programu a tabulky. Obsah tohoto editoru se mění dle aktuálně zvoleného obsahu. Lze mít zde otevřeno několik pracovních oblastí v jedné *Work area*. Libovolně lze i maximalizovat a minimalizovat *Work area*, pomocí funkcí v liště *Work area*. Pod titulkem *Work area* se nachází lišta označovaná *Editor bar*. Ta obsahuje základní funkce dostupné pro daný editor.

Stavová lišta se nachází na spodní části okna programu. Tato lišta obsahuje tlačítko pro změnu pohledu, seznam otevřených editorů a zobrazení aktuálního postupu. Seznam otevřených editorů může obsahovat jednotlivé editory samostatně nebo v případě otevření více podobných editorů současně. Tyto jsou seskupeny ve skupinách.

Karta otázek se nachází úplně napravo okna programu. Obsah této karty se liší v závislosti na aktuálně otevřeném editoru, protože v případě hardwarové konfigurace obsahuje otázkové karty pro položky *Hardware catalogue*, *Online tools* a *Tasks*. V případě editoru obsahuje *Libraries* a *Tasks*. *Taks* je otázková



Obr. 1.6 – Otázkové karty

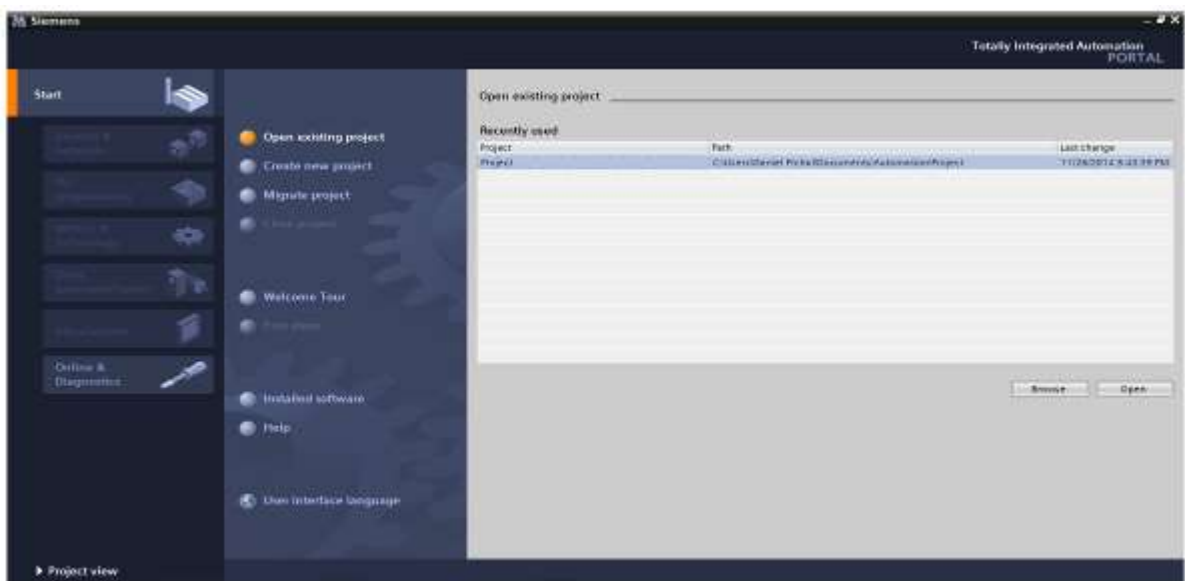
karta určena pro usnadnění hledání slov nebo částí v programu. *Hardware cataloge* je obdoba *Libraries*, ale je určen přímo na práci s hardwarovými prvky, vlastně představuje knihovnu hardwaru. *Libraries* je otázková karta pro práci s knihovnami. Tyto knihovny jsou velmi užitečný nástroj usnadňující práci v programu. Už s vytvořením projektu se automaticky vytvoří projektová knihovna, která obsahuje všechny prvky použité v programu. Do této knihovny lze dále přidávat prvky, které chce programátor dále použít v projektu. Tyto prvky lze uložit jako *Main copies* a *Types*. *Main copies* mohou být všechny projektové prvky a *Types* je přímo určeno na datové typy. Ovšem kromě projektových knihoven mohou být knihovny globální. Tyto globální knihovny lze využít ve více projektech. Na obr. 1.6 lze vidět systémové knihovny přímo od výrobce programu. Systémové knihovny jsou obsaženy v *Global libraries*. Tyto globální knihovny lze dále rozlišit na systémové knihovny od výrobce softwaru, firemní knihovny společné pro společnost vlastníci software a uživatelské knihovny, kde uživatel programu má vytvořeno několik vlastních knihoven. Poskytuje funkce jako jsou: vybrání objektu z knihovny, hledání a nahrazení objektu, vložení předdefinovaného objektu do pracovní oblasti. Tato karta otázek lze srolovat a otevřít klikem na tlačítko srolovat v liště karty (Siemens, 2014a).

Portálový pohled je strukturován jako otázkově orientovaný pohled pro nástroje. Pomocí tohoto pohledu lze snadno zvolit požadovanou otázku a s ní dále pracovat, většinou v projektovém pohledu. Případně s ní lze pracovat v podoknu portálového pohledu, které se liší dle vybrané otázky (Siemens, 2014a).

Obsahuje základní otázky. Počet těchto otázek se liší dle nainstalovaných komponent do systému. Například základní otázky v mé nainstalované verzi jsou *Start*, *Devices & Networking*, *PLC programming*, *Motion & Technology*, *Drive parametrization*, *Vizualization* a *Online & Diagnostic*. Otázka *Start* je otázka vyskytující se u všech verzí. Tato otázka obstarává práci s projektem, základní práci s projektem a programem. Otázka *PLC Programming* obsluhuje základní správu programových bloků včetně přidávání, odebírání a další manipulaci s těmito programovými bloky. Oproti tomu *Vizualization* obstarává základní správu vizualizovaných obrazovek pro HMI panely. Většinou po vybrané otázce je tento pohled přepnut na projektový pohled.

Tento pohled se skládá z několika základních elementů. Těmito základními elementy jsou *Portals for different tasks*, *Actions for the selected portal*, *Selection panel for the selected action*, *Change to the project view* a *Display of the project that is currently open*. Titulní lišta programu obsahuje jméno aktuálně otevřeného projektu, je-li otevřen. Nalevo pod lištou se nachází oblast *Portals for different tasks*, poskytující základní funkce pro individuální

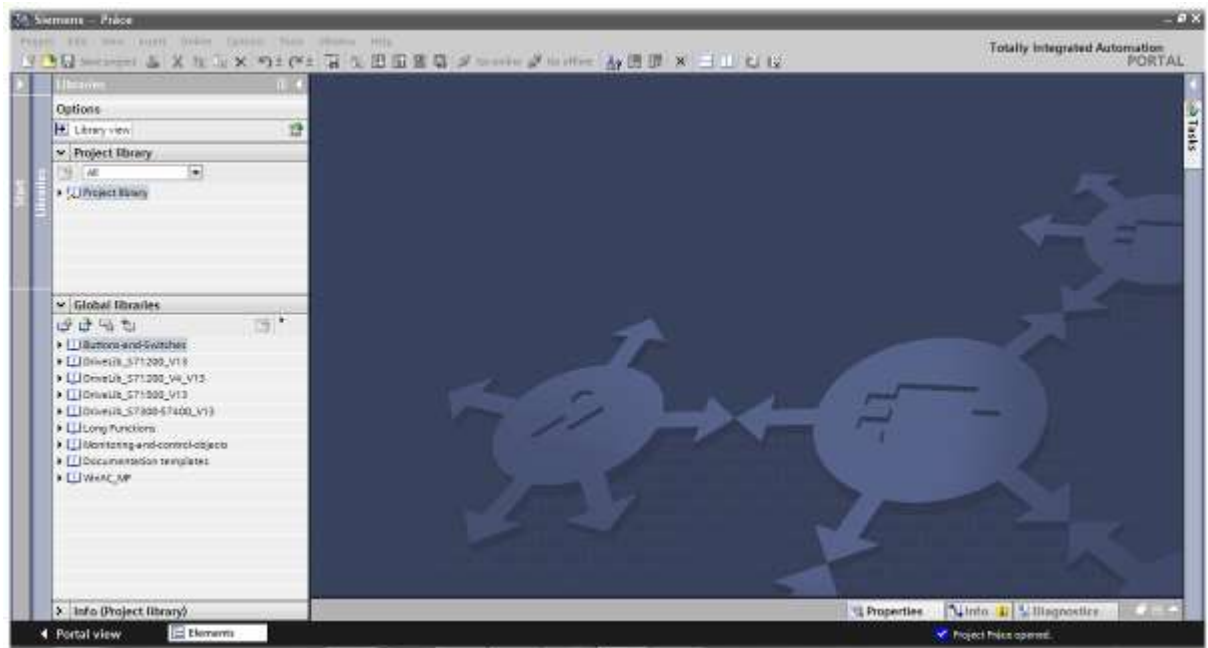
otázkové oblasti. K základním otázkám patří *Devices & Networks* pro správu hardwaru a sítí, *PLC programming* pro programování programu pro PLC, *Visualization* pro vizualizaci HMI a *Online & Diagnostics* pro diagnostiku zařízení. Další otázky, které má uživatel programu k dispozici, jsou odvislé na nainstalovaném softwaru. Vedle oblasti s otázkami je oblast *Actions for the selected portal*, obsahující akce pro vybraný portál, kde se zobrazí dostupné funkce pro vybranou otázku. Tento panel je dostupný pro všechny otázky. Napravo je vybraný panel *Selection panel for the selected action*, pro akci jehož obsah se mění ve vztahu k vybrané sekci. Na spodní liště programu nalevo je odkaz pro změnu na projektový pohled. Na spodní liště napravo se nachází informace o aktuálně otevřeném projektu (Siemens, 2014a).



Obr. 1.7 – Portálový pohled

Knihovní pohled obsahuje přehled prvků v knihovně projektu a otevřenou globální knihovnu. Na tento pohled lze přepnout s použitím otázky *Librarie* v otázkové kartě a zde zvolením tlačítka *Library view*. Tento knihovní pohled je vlastně v rámci projektového pohledu. Skládá se ze zjednodušené otázkové karty *Libraries* a pracovní oblasti, která je na obr. 1.8 zobrazena tmavě černou barvou. Tento knihovní pohled je primárně určen pro práci a správu knihoven (Siemens, 2014a).

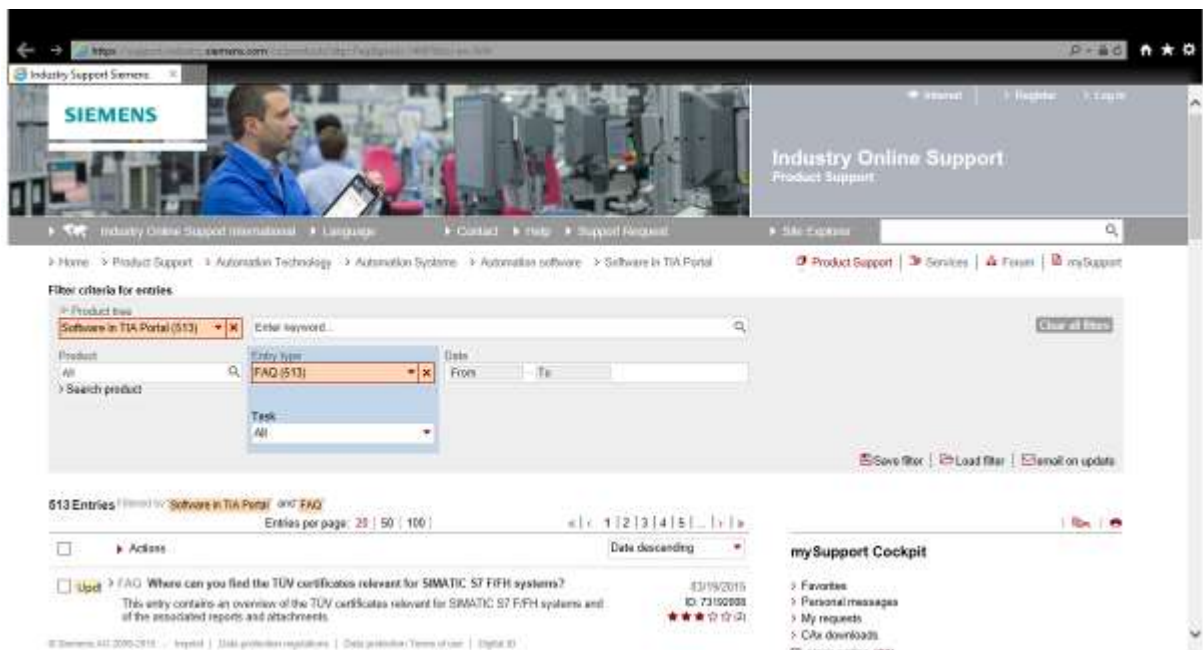




Obr. 1.8 – Knihovní pohled

## 1.2.2 Nápověda

Program TIA Portal V13 poskytuje několik možností pro programovou nápovědu. K hlavním prostředkům nápovědy bych doporučil před začátkem práce v programu manuál k tomuto programu. Tento manuál je dostupný v celé řadě světových jazyků, tedy angličtině, němčině, španělštině francouzštině, italštině a čínštině. Tento manuál je možno stáhnout přímo z webu výrobce tohoto softwaru, tedy firmě SIEMENS. Pro tyto stránky manuálu jsou k dispozici stránky výrobce, kterých lze dosáhnout pomocí webového prohlížeče a zde již jen najít požadovaný dokument nebo pomocí programu, kdy zvolíme v menu liště záložku *Help* a zde vybereme funkci *Product support*. Po této volbě by se mělo otevřít okno vašeho prohlížeče se stránkou obr. 1.9, která obsahuje všechny manuály, prezentace a další informace k produktům od firmy SIEMENS. Zde je možno pomocí formulářů již hledat vámi hledané dokumenty k dané problematice či produktu.

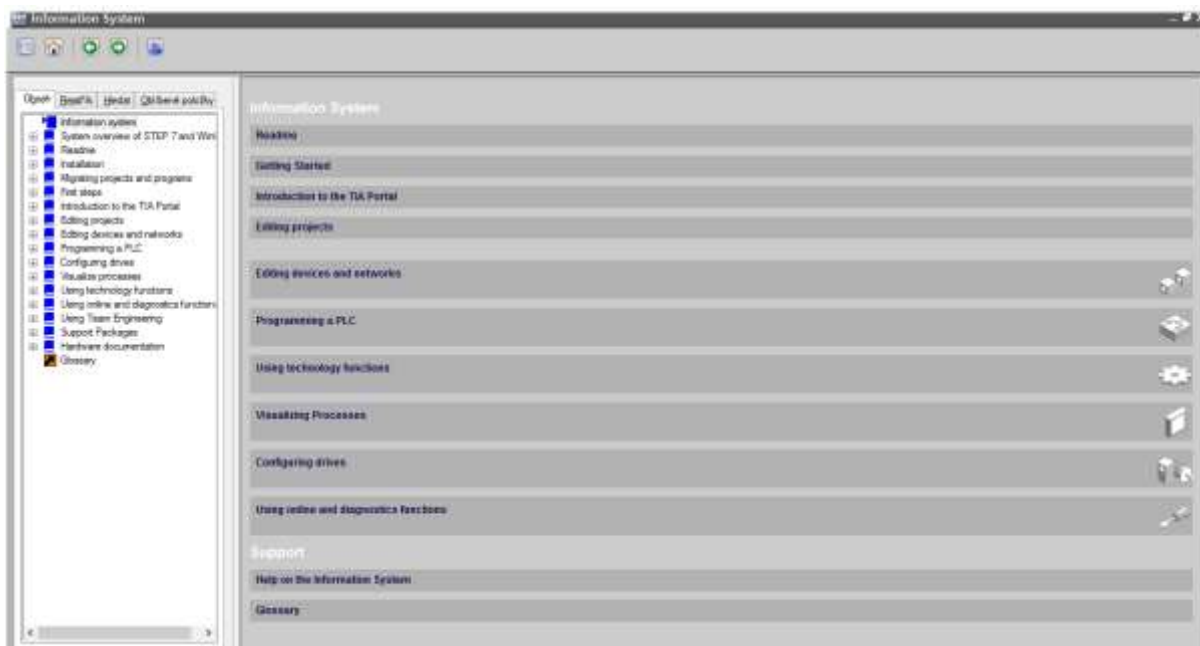


Obr. 1.9 – Nápověda s manuály pomocí prohlížeče

Druhou možností je využít informační systém programu TIA Portal. Tento informační systém jsou manuálové stránky ve velmi přehledné podobě, formulované jako pro mnohé uživatele s velmi dobrou, známou nápovědou v operačních systémech Windows. Tento informační systém navíc obsahuje celou řadu příkladů, konceptů a instrukcí. K tomuto informačnímu systému se lze dostat pomocí v menu lišty záložkou Help a zde vybereme funkci *Show help*, stiskem klávesy F1 nebo pomocí ikony *Online Help for TIA Portal* se otevře informační systém. Poté se otevře okno zobrazené na obr. 1.10. V tomto okně lze již snadno prohlížet a hledat požadované informace. Tato nápověda může být i obsahově citlivá, kdy se dle vybraného obsahu otevře požadovaná stránka s nápovědou vzhledem k vybrané položce. Například, pokud budu mít označenou některou instrukci v editoru programu TIA Portal V13 a stisku F1, tak se otevře informační systém, který již zobrazuje informace o dané instrukci.

Další nápovědu poskytuje program formou vyskakovací nápovědy pro správné vstupy v dialogovém boxu nebo formou nástrojového typu pro informace na elementech. Vyskakovací nápověda pro správné vstupy se zobrazí v případě vstupního pole, do kterého programátor zadává určitou hodnotu této proměnné. Zobrazí se vyskakovací zbarvená nápověda informující programátora o možnostech vstupu do tohoto programu. V případě špatně zadaného vstupu má tato nápověda červenou barvu a tím varuje programátora před špatně zadanou hodnotou této proměnné. Nástrojový typ se zobrazí v programu při delším podržení myši kurzoru nad určitým elementem, kaskádový nástrojový typ k danému elementu. Při rozkliknutí daného typu se

zobrazí další informace o daném elementu a případně odkazy do nápovědy v informačním systému.



Obr. 1.10 – Informační systém programu TIA Portal

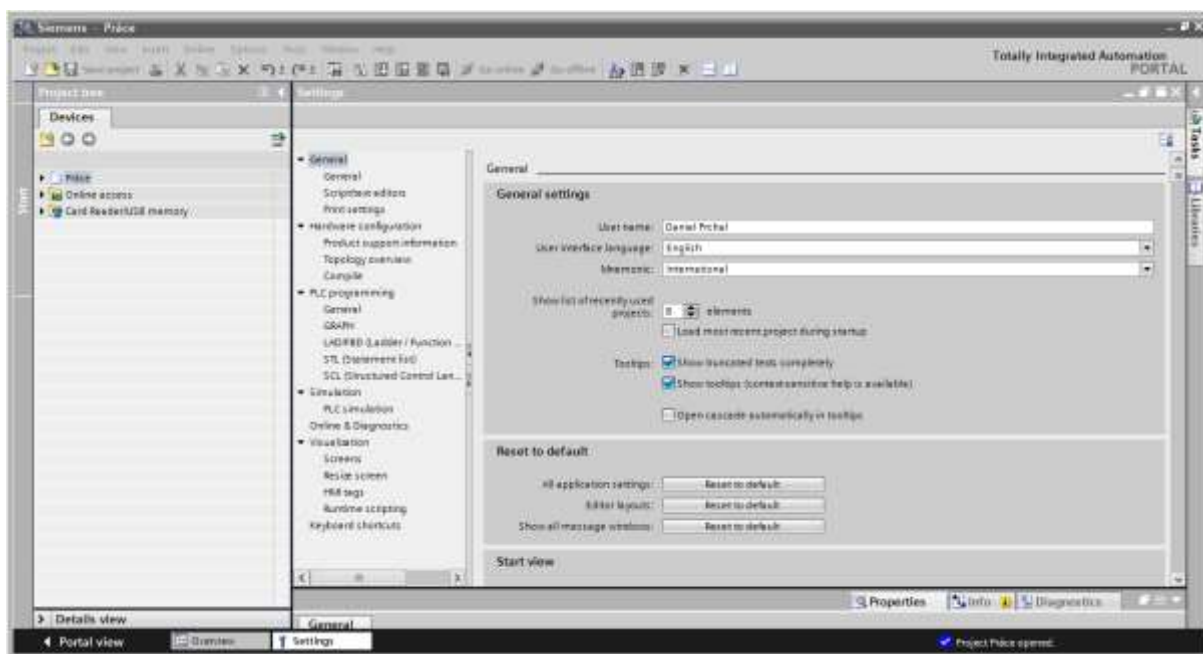
### 1.2.3 Nastavení

Před začátkem práce s programem je nutné správně nastavit program TIA PORTAL V13. Toto nastavení lze provést v projektovém pohledu v liště menu výběrem příkazu *Setting* v menu *Options*. Zde je možné nastavit program, skriptovací a textové editory a provést nastavení pro tisk.

V programovém nastavení je možné nastavit jméno autora projektu a rozhraní programového rozhraní. Dále je zde možné nastavit mnemoniku, která určuje použité mnemonické znaky v programu, jako jsou německé a mezinárodní znaky. Například při nastavení mnemoniky na mezinárodní, bude použita pro označení výstupu zkratka O1.1 a naproti tomu při německé mneomice bude výstup označen A1.1. Dále je zde možné nastavit různá zobrazení posledních otevřených projektů v programu, uložení pro výměnu dat, paměťová nastavení, nastavení startovacího pohledu a pohled pro projektové zobrazení. V této sekci lze i resetovat programové nastavení celého programu nebo jeho částí do původního nastavení definovaného výrobcem softwaru. Základní jazykové nastavení programu lze nastavit při instalaci programu nebo v nastavení. Je zde na výběr z několika základních světových jazyků, jako jsou angličtina, němčina, španělština a další světové jazyky. K těmto jazykům bohužel nepatří podpora českého jazyka a proto já osobně používám anglickou jazykovou verzi.

V globálním nastavení skriptů a textového editoru lze nastavit velikost používaného písma v programu, barvu písma pro různé typy textových elementů, tabulky, automatické doplňování a detailní nastavení pro STL a SCL editory.

Poslední možnosti nastavení jsou pro tisk dat z programu, hardwarové konfigurace, PLC programů, technologii pro řízení pohybů a HMI obrazovky.



Obr. 1.11 – Nastavení programu TIA Portal

### 1.3 SIMATIC S7-1500

SIMATIC S7-1500 je nástupce PLC řady SIMATIC S7-300 a S7-400. Oproti těmto předchozím zařízením se vyznačuje vyšším výkonem, integrovanými funkcemi pro řízení pohybu, PROFINET IO IRT, integrovaný display usnadňující diagnostiku a operace se zařízením v provozu bez nutnosti připojit PC a inovovaný STEP 7 jazyk. K nejnovějším podporovaným funkcím patří webový server, pomocí něhož je na S7-1500 vytvořen webový server a tento server je následně dostupný na síti (Siemens, 2014b).



Obr. 1.12 – S7-1500 (Siemens, 2014b)

Integrované funkce jsou řízení pohybu, PID regulátor a regulátor teploty. Stupeň ochrany tohoto zařízení splňuje IP20, z toho plyne, že je vhodný k instalaci do instalačních skříní. Pro tuto instalaci se využívá montáž na běžnou lištu. K jednomu CPU lze připojit až 32 modulů a dále tento CPU může být propojen s mnoha dalšími prvky, mezi kterými mohou být jak distribuované vstupně výstupní systémy, frekvenční měnič, vážící systémy a mnoho dalších. Místo prvního modulu na pozici 0 je předem rezervováno pro lokální napěťový zdroj PM nebo pro systémový napájecí zdroj PS. Další místo označované jako 1 je vždy vyhrazeno pro CPU. Další moduly je již lze umísťovat dle potřeby a není potřeba je umísťovat na konkrétní místo. Je nutno dodat, že fyzické umístění jednotlivých modulů by mělo odpovídat hardwarové konfiguraci zařízení vytvořené v programu z důvodu možných problémů s komunikací. Zabezpečení tohoto typu PLC je také na velmi vysoké úrovni. Hlavním ochranným prvkem je ochrana proti neoprávněnému přístupu. Při této ochraně lze přístup k PLC chránit heslem. Tedy nikdo, kdo nezná heslo, nemůže se zařízením pracovat. Další ochrana je ochrana proti kopírování programu. Nahraný program je pevně svázan s jednou paměťovou kartou. Paměťová karta se vkládá do zařízení a bez této karty program nelze spustit. Velká škála možností kombinace s dalšími moduly, které se připojují pomocí U konektoru, umožňuje toto PLC využít k řízení celé škály řídicích aplikací v průmyslu. Tyto moduly jsou řazeny do několika kategorií, které jsou, CPU, digitální a analogové I/O moduly, komunikační moduly, technologické moduly a systémy napájecí podpory (Siemens, 2014b).

Všechny zařízení jsou určeny pro umístění v rozvodných skříních pro montáž na montážní lišty pomocí PE montážních prvků. Montážní lišty jsou vyráběny ve standardizovaných délkách od 160 mm do 830 mm. Na montážní lištu lze připojit všechny typy modulů. Pro propojení jednotlivých modulů na montážní liště je určen U konektor. Hlavní

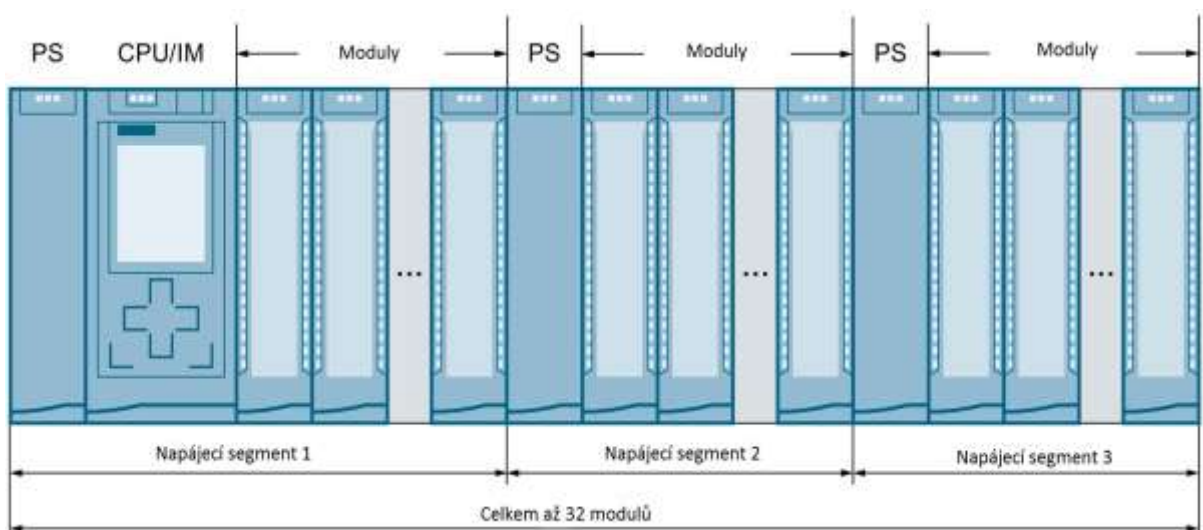
výhodou těchto konektorů je, že propojení probíhá pod moduly a není nutné tedy moduly propojovat pomocí dalších vodičů (Siemens, 2014b).

### 1.3.1 Rodiny CPU S7-1500

Z důvodu, že CPU rodiny S7-1500 jsou určeny pro univerzální aplikace, tak zařízení má několik různých variant CPU S7-1500. Varianty se v zásadě liší, jednak dle výkonu CPU pracovní paměti a také dle počtu PROFINET/PROFIBUS rozhraní pro malé, střední, velké a rychlostní aplikace. Dále CPU lze dělit dle určení pro standardní nebo bezpečnostní aplikace. Pokud je CPU odolné vůči selhání a tedy využitelné pro bezpečné aplikace, je CPU označeno za číslem za pomoci písmene F. Například CPU 1511-1 PN je určeno pro standardní aplikace a CPU 1511-1F PN je určeno pro bezpečnostní aplikace (Siemens, 2014b).

CPU 1511-1 je určeno pro malé a střední aplikace, z důvodu jeho vybavenosti pouze jedním rozhraním PROFINET. Toto nabízené CPU je nejpomalejší. Čas na jednu bitovou operaci je 60 ns. Pro střední a velké aplikace je určeno CPU 1513-1 PN s časem 40 ns pro vykonání jedné bitové operace. Komunikační úkony s velkými aplikacemi je určeno CPU 1516-3 PN/DP s jedním PROFIBUS a dvěma PROFINET rozhraními. Pro rychlé aplikace a komunikační otázky je přímo určeno rozhraní CPU 1517-3 PN/DP s časem pro jednu operaci 2ns. Toto CPU má dvě PROFINET a jedno PROFIBUS rozhraní. Nejvýkonnější typ CPU je CPU 1518-4 PN/DP s třemi PROFINET a jedním PROFIBUS rozhraním. Tento nejvýkonnější typ má čas 1 ns na jednu operaci (Siemens 2014b).

### 1.3.2 Moduly k SIMATIC S7-1500



Obr. 1.13 – Možnosti hardwarové konfigurace (Siemens 2014b)

CPU řady S7-1500 z důvodu univerzálnosti má celou řadu modulů. Tyto moduly jsou analogové, digitální, dále pak moduly rozhraní a technologické moduly. Tyto moduly lze konfigurovat pomocí programu STEP7 nebo pomocí programu s GSD soubory.

Digitální moduly slouží pro vstupní nebo výstupní digitální signály. Tyto digitální moduly jsou vyráběny ve variantách pro stejnosměrné napětí 24V nebo střídavé napětí 230V. Maximální výstupní proud digitálních výstupních modulů se liší dle použité verze modulu na 0,5 A a 2 A. V jednom modulu je 16 nebo 32 portů, které jsou izolovány ve skupinách po čtyřech. Digitální moduly jsou vybaveny v horní části modulu dvojicí diod pro signalizaci událostí. Tyto diody jsou pojmenovány RUN a ERROR. Nesvíti-li ani jedna z diod RUN a ERROR, není zařízení napájeno nebo je nízké napětí na sběrnici. Při startu modulu bliká dioda RUN a dioda ERROR je zhasnutá. Hardwarová porucha na digitálním modulu je signalizována blikáním obou dvou signalizačních diod. Pokud zařízení v pořádku běží, tak svítí pouze dioda RUN. Pro každý digitální vstup je zde dioda označená CHx. Tato dioda pro každý vstupní port je zde pouze z důvodu signalizace stavu. Tedy, je-li stav daného kanálu 1, tak dioda svítí a při stavu 0 je dioda zhasnutá (Siemens, 2014b).

Analogové moduly jsou určeny pro práci s analogovými signály. Dle typu modulu mohou na jednotlivých kanálech měřit nebo pracovat s napětím, proudem, odporem, termočlánky, termometrem, diagnostikou nebo přerušeními se dvěma limity pro kladné a záporné hodnoty vstupního signálu. Toto nastavení lze provést pro každý kanál. Rozlišení modulů je 16 bitů včetně rozlišení znaménka. Moduly jsou vybaveny 8 nebo 4 vstupními kanály. Kanál je schopen pracovat s napětím 1 až 5 V, při proudu 0 až 20 mA, 0 až 10 V při proudu 4 až 20 mA a  $\pm 10$  V při proudu  $\pm 20$  mA. Stejně jako digitální moduly je kanál vybaven diodami RUN, ERROR, které fungují stejně jako v případě digitálních modulů. Oproti digitálním modulům jsou analogové vybaveny diodou PWM. Je-li tato dioda zhasnutá, tak je úroveň napájení analogového modulu menší než přípustná hodnota napětí. Svítí-li dioda PWM, tak napájení modulu je v pořádku. Stejně jako digitální moduly má analogový modul diodu pro každý kanál. Pokud dioda nesvíti, tak kanál je odpojen. Zelené rozsvícení diody značí, že kanál je nakonfigurován a pracuje v pořádku. Červené rozsvícení kanálu značí chybu nebo špatnou konfiguraci kanálu (Siemens, 2014b).

K vytvoření distribuovaného vstupně výstupního systému lze využít moduly rozhraní. Tyto moduly rozhraní jsou značeny ET200 a pomocí nich lze vytvořit distribuované vstupně výstupní moduly. To znamená, že CPU bere moduly umístěné za ET200 za jeho vlastní moduly, které jsou reálně umístěny zcela mimo 32 možných modulů CPU. CPU s ET200 a komunikuje pouze pomocí síťového rozhraní. Tímto síťovým rozhraním je PROFINET nebo PROFIBUS.

Výkon těchto ET200 je 14 W, a jsou tedy bez nutnosti připojit se na další zdroj, schopni podporovat až 12 vstupních nebo výstupních modulů. ET200 jsou malá CPU a lze je využít i samostatně pro řízení velmi malých a jednoduchých zařízení (Siemens, 2014b).

Pro rozšíření možností komunikace s ostatními zařízeními lze připojit k zařízení komunikační moduly. Lze využít komunikační moduly přes rozhraní RS422/485, RS232, PROFINET, PROFIBUS nebo průmyslový internet (Siemens, 2014b).

Napájecí moduly slouží k výkonovému rozšíření, možnosti připojení dalších modulů, protože každý modul odebírá určitý výkon a CPU jsou výkonově omezena dle verze. Je nutné při překročení maximálního možného výkonu dodávaného CPU připojit napájecí výkonový modul a tím celkově zvýšit celkový výkon, se kterým lze na systému pracovat. S výšemi výkonu lze přidávat další a další zařízení. V manuálu každého modulu je uvedena jeho výkonová náročnost. Celková náročnost se sčítá. Napájecí moduly jsou vyráběny dle výkonu a napájecího napětí. Napájecí napětí je stejnosměrné 24 V, 48 V nebo 60 V. V případě střídavého napájení se využívá 120 V nebo 230 V. Výkon těchto modulů je 25 W a 60 W (Siemens, 2014b).

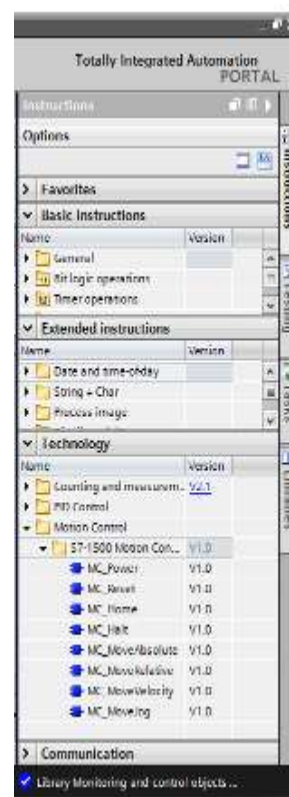
Pro napájení zařízení, jako jsou senzory a motory, lze využít zatěžovací proudové napájecí moduly. Tyto moduly jsou napájeny střídavým napětím 120 V nebo 230 V o běžné síťové frekvenci. Výstupní napětí je vždy stejnosměrné, o hodnotě 24 V. Jediné, v čem se zásadně druhy těchto modulů liší, je výkon a s ním i proud. Vyrábí se ve verzích s výkonem 70 W, při tomto výkonu je maximální výstupní proud 3 A, a 190 W. Modul o výkonu 190 W je schopen poskytnout maximální výstupní proud 8 A (Siemens 2014b).



## 1.4 KNIHOVNA INSTRUKCÍ PRO PRÁCI S REGULOVANÝMI POHONY

### 1.4.1 Knihovna instrukcí

Knihovna pro práci s regulovanými pohony se skládá ze tří skupin bloků. První skupinu tvoří bloky starající se o základní funkce. Sem patří bloky: MC\_Reset, MC\_Home a MC\_Halt. Druhou skupinu tvoří povolovací funkce a je to pouze jeden blok MC\_Power. Tato funkce se stará o samotné povolení funkce technologického objektu. Třetí skupinu tvoří polohovací funkce a ty jsou MC\_MoveJog, MC\_MoveRelative, MC\_MoveAbsolute a MC\_MoveVelocity. Tyto funkce se starají o provádění samotného pohybu. Užití jednotlivých instrukcí pro řízení se liší dle použitého typu technologického objektu. Tyto instrukce se při řízení technologického objektu vzájemně ovlivňují a každá instrukce se chová v závislosti na vstupních parametrech a ovlivňuje výstupní parametry instrukce.



Obr. 1.14 – Knihovna instrukcí

Je nutno nastavit jednotlivé parametry pro instrukci. Jednotlivé parametry instrukcí lze nastavit přímo jako parametr instrukce pro řízení pohybu nebo nepřímo za pomoci přiřazení hodnoty proměnné do datové struktury instrukce.

Nastavení parametrů instrukce přímo je velmi jednoduché a stačí pouze dvojklik na hodnotu parametru instrukce, který je uveden vždy vedle určité instrukce. Poté se otevře možnost upravit parametr této instrukce. Zde lze uvést buď konstantní hodnotu parametru, absolutní adresu proměnné nebo přiřadit proměnnou. Tyto možné hodnoty parametru lze zapsat do výběrového pole. Přímé nastavení parametrů zobrazuje obrázek obr. 4.2.

Další možností je v požadované části programu uvedení hodnoty pro datový blok dané instrukce, protože tento instrukční datový blok je vždy spojen s instrukcí. Příklad je zobrazen na obr.



Obr. 1.15 – Nastavení parametru přímo

4.3, kde hodnota parametru *Enable* pro instrukci MC\_POWER je vždy povolena, díky přímému nastavení parametru na hodnotu 1.



Obr. 1.16 – Nastavení parametru nepřímou

Veličina těchto parametrů je většinou bezrozměrná, kromě parametrů, které nastavují rychlost, zrychlení, zpomalení, zrychlení a pozice. Tyto parametry mají rozměr a veličina těchto parametrů závisí na nastavení technologického objektu. V technologickém objektu lze zvolit několik veličin.

#### 1.4.2 Obecné instrukční parametry

Obecné instrukční parametry jsou normální parametry instrukcí, ale vyskytují se u téměř všech instrukcí. Většinou se jedná o chybové parametry, informativní a ovládací parametry. K obecným instrukčním parametrům patří *Error*, *ErrorID*, *Axis*, *Busi*, *Execute*, *Done* a *CommandAborted*.

Hlavním z těchto parametrů v každé instrukci je vstupně výstupní parametr *Axis*. Parametr *Axis* definuje, který technologický objekt bude instrukce ovládat. Tento parametr je nutné uvést u každé instrukce, protože není-li tento technologický objekt uveden, nepůjde program ani zkompileovat. Hlavní problém s kompilací spočívá na syntaktické chybě v daném místě umístění instrukce v programu. Tuto syntaktickou chybu bude program hlásit z důvodu, že instrukce by nevěděla, co má ovládat. Pokud není uveden technologický objekt je zobrazeno místo jména osy v parametru, červeným textem hodnota „<??>“. Po přiřazení se místo červeného textu zobrazí jméno ovládaného technologického objektu (Siemens, 2014c).

Chybový výstupní parametr *Error* slouží pro informování o výskytu chyby při vykonávání určité instrukce. Výchozí hodnota parametru je 0. Tento parametr je typu BOOL a může nabývat tedy hodnot 1 nebo 0. Signalizuje-li tento parametru 1, značí tak výskyt chyby při vykonávání dané instrukce. Tento výstupní parametr se využívá u všech instrukcí. Pro určení přesné chyby tento parametr neposkytuje dostatečné informace z důvodu jeho typu BOOL parametru (Siemens, 2014c).

K získání více informací o charakteru vzniklé chyby je zde umístěn i výstupní parametr *ErrorID* sloužící k informování o výskytu chyby při vykonávání instrukce a blíže oproti

parametru *Error* specifikuje danou chybu. Parametr je typu WORD a signalizuje přesně číslo vzniklé chyby technologického objektu. Toto číslo je v hexadecimálním tvaru. Hexadecimální tvar čísla lze poznat pomocí předpony 16# před číslem. Například číslo 16#111 označuje hexadecimální číslo 111. Tabulka možných chyb je uvedena v tab. 1.2.

Tab. 1.2 – Tabulka chyb pohybových instrukcí (Siemens, 2014c)

Error ID	Popis chyby	Ošetření chyby
16#0000	Žádná chyba	-
16#8001	Chyba TO při zpracování instrukce	Nutnost v datovém bloku <i>TO</i> najít znak chyby v <i>ErrorDetail.Number</i>
16#8002	Neplatná hodnota Axis	Zkontrolovat hodnotu parametru <i>Axis</i>
16#8003	Neplatná rychlost	Nastavit správnou hodnotu v parametru <i>Velocity</i>
16#8004	Neplatné zrychlení	Nastavit správnou hodnotu v parametru <i>Acceleration</i>
16#8005	Neplatné zpomalení	Nastavit správnou hodnotu v parametru <i>Deceleration</i>
16#8006	Neplatné trhnutí	Nastavit správnou hodnotu v parametru <i>Jerk</i>
16#8007	Neplatný směr nebo jsou aktivní oba parametry <i>JogForward</i> a <i>JogBackward</i>	Nastavit správnou hodnotu v parametru <i>Direction</i> nebo resetovat <i>Jog</i> parametry
16#8008	Neplatná vzdálenost	Nastavit správnou hodnotu v parametru <i>Distance</i>
16#8009	Neplatná pozice	Nastavit správnou hodnotu v parametru <i>Position</i>
16#800A	Neplatné nastavení operačního modu	Nastavit správnou hodnotu v parametru <i>Mode</i>
16#800B	Neplatné nastavení parametru zastavení	Nastavit správnou hodnotu v parametru <i>StopMode</i>
16#800C	Jeden TO ovládán více MC_Power	Pro jeden TO může být pouze jedna MC_Power
16#800D	Úkon není povolen v aktuálním módu	Počkat na dokončení reset TO
16#800E	Restart není možný při aktivované ose	Před Restartem deaktivujte technologický objekt
16#800F	TO je zakázán a činnost nemůže vykonat	Povolit TO
16#8010	Neplatné nastavení naváděcího módu pro inkrementační kodér	Módy 6 a 7 nejsou možné pro inkrementační kodér, pouze módy 0, 1, 2, 3, 4, 5, 8
16#8011	Neplatné nastavení naváděcího módu pro absolutní kodér	Pasivní a aktivní módy 2, 3, 4, 5 nejsou možné pro inkrementační kodér
16#8012	Příkaz nelze vykonat z důvodu práce v řídicím panelu osy	Navrátit řízení programu z řídicího panelu na uživatelský program
16#8013	Online propojení mezi CPU a TIA Portal je přerušeno	Zkontrolujete propojení
16#8014	Není dostupná vnitřní paměť	Je dosaženo maximálního počtu řízení pohybu
16#8015	Chyba v konfiguraci TO	Zkontrolujete konfiguraci TO
16#8016	Aktuální hodnoty nejsou dostupné	Zkontrolovat stav aktuální hodnoty
16#8FFF	Nespecifikovaná chyba	Kontaktujte technickou podporu

Parametr *Execute* je obdoba parametru *Enable*, ale slouží k začátku vykonávání úkonu dané instrukce. Tedy nepovoluje činnost, místo toho pouze spustí instrukci. Tato vstupní proměnná je typu BOOL a její výchozí hodnota je 0. Výchozí hodnota tohoto parametru je 0. Po nastavení hodnoty parametru *Execute* na 1, se začne vykonávat činnost dané instrukce. Výstupní parametr *Done* je typu BOOL. *Done* slouží k signalizaci, zda je činnost dané instrukce dokončena. Tedy, jestliže signalizuje tento parametr 1, tak je činnost dané instrukce dokončena. Výstupní parametr *CommandAborted* je typu BOOL a je využíván pro signalizaci vykonávání úlohy, při které bylo zakázáno vykonávání jiné úlohy. Při stavu 0 je vše v pořádku a úkoly instrukce jsou vykonávány běžně, ale při změně stavu na hodnotu 1 je zakázána činnost úlohy. Tento zákaz může být například vykonán z důvodu možné chyby při navádění. Tedy nelze současně posunovat osu a navádět osu technologického objektu. Parametr *Busy* slouží pro informování o začátku vykonávání činnosti instrukce. Jedná se o výstupní parametr instrukce. Tento parametr je typu BOOL a jeho výchozí nastavení je nula. Signalizuje-li tento parametr hodnotu jedna, tak bylo započato vykonávání činnosti instrukce (Siemens, 2014c).

### 1.4.3 Dynamické parametry pohybu

Všechny tyto parametry *Velocity*, *Acceleration*, *Deceleration* a *Jerk* se využívají k definování vlastností fyzického pohybu technologického objektu, kdy se po nastavení těchto parametrů započne vykonávat pohyb s ohledem na nastavené parametry. Všechny parametry jsou typu LREAL. Tím vzniká možnost nastavit velké množství nastavení i s využitím čísel s desetinnou čárkou. V proměnných typu LREAL lze využívat hodnoty v rozmezí zaokrouhlených čísel  $-1,798 \cdot 10^{308}$  až do  $1,798 \cdot 10^{308}$ , což činí dostatečný prostor pro jakékoliv vstupní hodnoty (Siemens, 2014c).

Vstupní parametr *Velocity* slouží k definování rychlosti, kterou má daná instrukce použít při pohybu osy technologického objektu. Výchozí hodnota tohoto parametru je nastavena na hodnotu 100,0. V tomto parametru je povolena jakákoliv hodnota z rozsahu pro proměnné typu LREAL (Siemens, 2014c).

Vstupní parametr *Acceleration* představuje zrychlení, kterým bude zrychlována osa, dokud nedosáhne požadované rychlosti. Hodnota tohoto parametru nesmí být nastavena na hodnotu 0,0, protože tato hodnota je neplatná. Při této hodnotě bude signalizovat chybu výstupní parametr *Error* a *ErrorID*. Číslo této chyby je 16#8004, tedy chyba při špatně nastavené rychlosti. Pokud programátor nastaví hodnotu tohoto parametru na menší než 0, tak místo hodnoty zrychlení definované parametru *Acceleration* je využita hodnota zrychlení

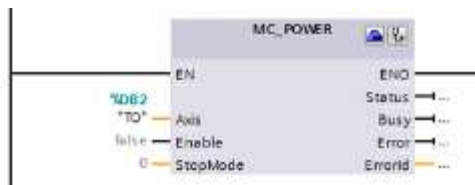
definovaná v technologickém objektu v umístění *Technology object > Configuration > Extended parameters > Dynamic defaults* nebo v datovém bloku technologického objektu na adrese *<TO>.DynamicDefaults.Acceleration* (Siemens, 2014c).

Vstupní parametr *Deceleration* oproti parametru *Acceleration* zpomaluje rychlost otáčení pohonu. *Deceleration* je vlastně zpomalení, tedy pokud je potřeba snížit rychlost otáček nebo zastavit pohon. Právě k této akci se využívá vstupní parametr *Deceleration*. Obdobně jako v případě *Acceleration*, tak v případě parametru *Deceleration* nelze nastavit na hodnotu 0,0. Pokud přesto programátor tento parametr nastaví na 0, bude program hlásit chybu na výstupních parametrech *Error* a *ErrorID*. Číslo této chyby je 16#8005. Chyba 16#8005 znamená špatně nastavené zpomalení osy technologického objektu. Při nastavení hodnoty parametru na číselnou hodnotu menší jak 0,0. Program místo hodnoty obsažené v parametru *Deceleration* využije hodnotu nastavenou v technologickém objektu. Hodnotu technologického objektu lze nastavit na programovém umístění *Technology object > Configuration > Extended parameters > Dynamic Defaults* nebo přímo v datovém bloku technologického objektu. V datovém bloku technologického objektu tuto hodnotu nalezneme na umístění *<TO>.DynamicDefaults.Deceleration* (Siemens, 2014c).

Posledním parametrem upravujícím vlastnosti pohybu je *Jerk*. *Jerk* znamená při překladi do češtiny trhnutí. Z fyzikálního hlediska se jedná o vektor, kdy lze vypočítat pomocí první derivace zrychlení v čase. Výsledkem je opět vektor. Tento vektor představuje stupeň změny zrychlení. Jednotkou tohoto parametru je  $\text{ms}^{-3}$ . Parametr *Jerk* má hlavní vliv na rychlostní profil. Výchozí hodnota tohoto parametru je -1,0. Toto výchozí nastavení způsobí nastavení konstantně zrychlujícího se rychlostního profilu. Konstantně zrychlující rychlostní profil je nastaven, dokud hodnota tohoto parametru je menší jak 0,0. Při tomto profilu je rychlost konstantně zrychlována dle parametru *Acceleration* a hodnota parametru *Jerk* je používána pro činnost pohonu. Dalším možným rychlostním profilem je lichoběžníkový rychlostní profil. Lichoběžníkový průběh má nejprve lineární rychlostní profil do dosažení limitní rychlosti a při zastavování je opět rychlost lineárně snižována. Tento profil odpovídá hodnotě parametru odpovídající 0,0. Při hodnotě parametru vyšší jak 0,0 není využívána hodnota tohoto parametru, ale pouze hodnota parametru *Jerk* definovaná v technologickém objektu *Technology object > Configuration > Extended parameters > Dynamic Defaults* nebo přímo v datovém bloku technologického objektu. V datovém bloku technologického objektu tuto hodnotu nalezneme na umístění *<TO>.DynamicDefaults.Jerk* (Siemens, 2014c).

#### 1.4.4 Povolovací instrukce MC\_Power

Instrukce MC\_Power patří k základním instrukcím. Tato instrukce zajišťuje povolovací funkci pro technologický objekt, proto jeho činnost nemůže být zakázána jinou z instrukcí pro řízení pohybu.



Obr. 1.17 – MC\_Power v LAD

```
1 MC_POWER_Data(Axis:=multi_fb_in_,
2 Enable:=false,
3 StopMode:=0,
4 Status=>_bool_out_,
5 Busy=>_bool_out_,
6 Error=>_bool_out_,
7 ErrorId=>_word_out_);
```

Obr. 1.18 – MC\_Power v SCL

Tato instrukce zakazuje nebo povoluje činnost technologického objektu pomocí vstupního parametru *Enable*. Stav technologického objektu signalizuje výstupní parametr *Status*. Vykonávání instrukce MC\_Power je signalizováno pomocí výstupního parametru *Busy*. Pokud je zakázán technologického objekt parametrem *Enable*. Zastavení osy je řízeno pomocí vstupního parametru *StopMode*. Vzniklé chyby při vykonávání instrukce jsou signalizovány na parametrech *Error* a *ErrorID*. Využití této instrukce je u všech typů technologických objektů. V případě této instrukce může být pro jeden technologický objekt použita pouze jedna instrukce MC\_Power. V případě použití několika instrukcí MC\_Power pro jeden technologický objekt budou instrukce signalizovat chybu 16#800C, která je uvedena v tab. 1.2. Pokud byl technologický objekt zakázán z důvodu alarmu, tak odstraněním příčiny alarmu a potvrzením tohoto alarmu bude tento technologický objekt znovu povolen. Během tohoto procesu musí být držen parametr *Enable* ve stavu pravda, protože v případě přerušení stavu parametru *Enable* by byl proces instrukce MC\_Power ukončen a instrukce by začala svoji činnost od začátku (Siemens, 2014c).

Tato instrukce má celkem sedm parametrů. Vstupně výstupní parametr je pouze *Axis*. Vstupní parametry jsou *Enable* a *StopMode*. Výstupními parametry této instrukce jsou *Status*, *Busy*, *Error* a *ErrorID*.

K povolení činnosti technologického objektu se využívá vstupní parametr *Enable*. Jedná se o vstupní parametr pro instrukci MC\_Power a u jiných instrukcí se nevyužívá. Tento parametr je typu BOOL, tedy může nabývat hodnot 1 nebo 0, přičemž 0 je současně výchozí hodnota tohoto parametru. Při nulové hodnotě tohoto parametru není technologický objekt povolen. Tedy všechny ostatní instrukce pro řízení pohybu jsou odpojeny od technologického objektu a neovlivňují jeho vlastnosti, ale ani nejsou aktualizovány vlastnosti technologického objektu z fyzického stavu osy. Stav 1 značí, že činnost technologického objektu je povolena

a lze využívat další instrukce pro regulaci regulovaného pohonu. V tomto stavu jsou i současně aktualizovány hodnoty technologického objektu z fyzického stavu osy.

Parametr *StopMode* slouží pro nastavení módu, který je použit po zakázání technologického objektu k zastavení technologického objektu. Jedná se o vstupní parametr pro instrukci *MC\_Power* a u jiných instrukcí není využit. Tento parametr se využívá pouze u instrukce *MC\_Power*. Parametr je typu INT a může nabývat hodnoty -32768 až 32767. V případě tohoto parametru se využívají pouze hodnoty 0 až 2. Hodnota tohoto parametru se neaplikuje v případě, že technologickým objektem je externí kodér. Parametr *StopMode* má nulovou hodnotu při výchozím stavu. Hodnota tohoto stavu značí nastavení módu nouzové zastavení. Mód nouzové zastavení značí zakázání technologického objektu, brždění osy bez jakéhokoliv omezení zpomalení. Toto zpomalení je definováno v konfiguraci technologickém objektu na umístění *Technology object > Configuration > Extended parameters > Emergency stop ramp*. Další možnost nastavení tohoto zpomalení lze provést přímo na adrese datového bloku technologického objektu *<TO>.DynamicDefaults.EmergencyDeceleration*. Kromě módu nouzového zastavení existují módy pro okamžité zastavení a zastavení s maximální dynamickou hodnotou. Okamžité nastavení se nastavuje při hodnotě 1 parametru *StopMode*. Po zakázání technologického objektu je nastaven parametr na 0 a osa pohonu je bržděna dle konfigurace pohonu. Pohon začne brzdit dle parametrů nastavených na frekvenčním měniči do doby, dokud nedosáhne rychlost pohybu osy nulových hodnot. Poslední zastavovací mód zastavení s maximální dynamickou hodnotou se nastavuje hodnotou 2 parametru. Po zakázání technologického objektu je osa bržděna s použitím maximálního zpomalení konfigurovaného v technologickém objektu na umístění *Technology object > configuragiton>Expended parameters > Dynamic limits*. Zde je konfigurováno jak maximální trhnutí, tak i maximální zpomalení datovém bloku technologického objektu *<TO>.DynamicLimits.MaxDecelaration* a *<TO>.DynamicLimits.MaxJerk* (Siemens, 2014c).

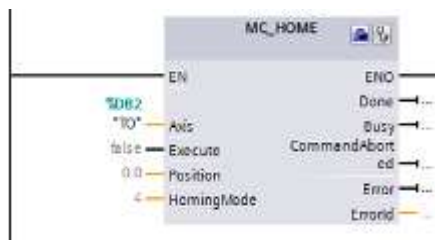
Parametr *Status* slouží pro informování o stavu povolení technologického objektu. Tento parametr je typu BOOL, může nabývat hodnot pouze pravda nebo nepravda a jeho výchozí nastavení je na hodnotě nepravda. Jedná se o výstupní parametr k instrukci *MC\_Power*. Pokud instrukce nastaví hodnotu tohoto parametru na jedničku, signalizuje, že technologický objekt je povolen. To znamená příjem úkonů pro poziční a rychlostní osy řízení pohybu. Rychlostní a poziční řízení je dostupné a aktuální hodnoty technologického objektu odpovídají hodnotám fyzického stavu osy pohonu. Je-li instrukcí hodnota tohoto parametru nastavena na hodnotu 0, je instrukcí signalizováno zakázání technologického objektu. Při tomto stavu poziční a rychlostní osa neplní žádné úkony pro řízení pohybu a rychlostní nebo poziční řízení není

aktivní. Ani aktuální hodnoty technologického objektu neodpovídají hodnotám fyzického stavu osy pohonu (Siemens, 2014c).

### 1.4.5 Základní instrukce

Tabulky Základní instrukce MC\_Reset, MC\_Home a MC\_Halt se starají o základní funkce technologického objektu, mezi tyto základní funkce patří navádění, reset a zastavení pohybu. Ostatní funkce pro řízení pohybu k těmto funkcím nepatří.

Instrukce MC\_Home patří k základním instrukcím. Tato instrukce slouží k vytvoření vztahu mezi pozicí technologického objektu a mechanickou pozicí osy fyzického tělesa. Tedy hodnotě pozice v technologickém objektu je přiřazena k fyzickému umístění naváděcí značky v čase. Využití této instrukce je u všech typů technologického objektu kromě rychlostní osy, protože v případě rychlostní osy záleží pouze na rychlosti osy a počáteční poloha osy je nepodstatná.



Obr. 1.19 – MC\_Home v LAD

```
12 MC_HOME_DB" (Axis:=multi_fb_in_,
13 Execute:=false,
14 Position:=-0.0,
15 HomingMode:=4,
16 Done=>_bool_out_,
17 Busy=>_bool_out_,
18 CommandAborted=>_bool_out_,
19 Error=>_bool_out_,
20 ErrorId=>_word_out_);
```

Obr. 1.20 – MC\_Home v SCL

Celkem lze při práci s touto instrukcí využít devět parametrů. Instrukce má jeden vstupně výstupní parametr Axis, tři vstupní parametry a pět výstupních parametrů. Vstupní parametry instrukce MC\_Home jsou: *Execute*, *Position* a *Mode*. Výstupní parametry jsou: *Done*, *Busy*, *CommandAborted*, *Error* a *ErrorID*. Vstupní parametr *Position* slouží k nastavení hodnoty, která je použita při vybraném naváděcím módu instrukce MC\_Home. Tento parametr je typu LREAL a jeho výchozí hodnota je 0,0. Vstupní parametr *Mode* definuje naváděcí mód, který technologický objekt provede po aktivaci instrukce MC\_Home. Tento vstupní parametr je typu INT, ale prakticky se využívá z celkových možných -32 768 až 32 767 kombinací pouze 10 číselných hodnot. Tyto hodnoty jsou v rozmezí 0 až 10 a výchozí nastavení tohoto vstupního parametru je 4. Je zde možnost volby z přímého, pasivního, aktivního, absolutního navádění kodéru a zakázání navádění. Přímé naváděcí módy využívají aktuální pozici technologického objektu, nastavenou na hodnotu parametru Position. K přímým naváděcím módům patří přímé absolutní navádění a přímé relativní navádění. Naváděcí mód přímého absolutního navádění



lze nastavit hodnotou parametru 0 a přímé relativní navádění lze nastavit při hodnotě parametru 1. Při absolutním navádění je aktuální pozice technologického objektu nastavena na hodnotu parametru *Position*. Oproti tomuto se relativní přímé navádění posune pozici technologického objektu o hodnotu parametru *Position* a až poté se provede nastavení aktuální pozice na hodnotu parametru *Position*. Pasivní naváděcí módy pracují na principu detekce naváděcí značky. Po detekci naváděcí značky provedou dle nastaveného módu navádění. Pasivní módy lze nastavit pomocí hodnot 2, 3 nebo 8, kdy se tyto jednotlivé módy liší minimálně svými vlastnostmi. Při hodnotě 2 je po detekci naváděcí značky aktuální pozice nastavena na hodnotu v parametru *Position*. Oproti tomuto má pasivní mód 3 pouze změnu v tom, že nehledí na hodnotu parametru *Position*, ale nastaví pouze hodnotu definovanou v technologickém objektu. Pasivní, naváděcí mód 8 pracuje stejně jako mód při hodnotě 2. Pouze při resetu instrukce není potřeba znovu provádět navádění, protože naváděcí status se nemění. Hlavní změnou oproti pasivním naváděcím módům je u aktivních to, že provedou pohyb osy dle nastaveného módu. Při nastaveném módu na parametr 4 se započne vykonávat mód aktivní navádění. Osa se posune dle nastavené konfigurace technologického objektu a po dokončení tohoto pohybu je nastavena aktuální pozice na hodnotu parametru *Position*. Obdobně pracuje mód absolutního navádění při parametru 5, kromě toho, že parametr *Position* nemá vliv na aktuální pozice a ta je nastavena dle hodnoty parametru v technologickém objektu. Navádění kodéru se využívá, pokud je k určování pozice objektu využíván externí kodér. Tyto módy jsou absolutní a relativní absolutního kodéru, kdy je při relativním absolutním kodéru při hodnotě parametru 6, nastavena aktuální pozice na hodnotu parametru *Position* po dokončení pohybu o vzdálenost definovanou v parametru *Position*. Absolutní mód kodéru pracuje obdobně, pouze není vykonáván pohyb a je přímo nastavena hodnota parametru *Position*. Tento mód lze nastavit pomocí hodnoty parametru 7. K zakázání pasivního navádění slouží poslední mód ukončení pasivního navádění. Tento mód lze nastavit pomocí hodnoty 9 na parametru *Mode* a tím jsou všechny běžící úlohy pro pasivní navádění zakázány. Tedy, technologický objekt není naveden a běží-li před změnou módu pasivní mód, tak je tento mód ukončen a výstupní parametr *CommandAborted* začne signalizovat 1. Dokončení přerušení probíhajícího navádění se signalizuje výstupním parametrem *Done* (Siemens, 2014c).

Instrukce *MC\_Halt* slouží k přerušení pohybu osy a uvedení jejího stavu do klidu. Brzdění je prováděno dle definovaných parametrů instrukce.



Obr. 1.21 – MC\_Halt v LAD

```

1 □ "MC_HALT_DB_1" (Axis:= multi_fb_in_,
2     Execute:=false,
3     Deceleration:=-1.0,
4     Jerk:=-1.0,
5     Done=>_bool_out_,
6     Busy=>_bool_out_,
7     CommandAborted=>_bool_out_,
8     Error=>_bool_out_,
9     ErrorId=>_word_out_);

```

Obr. 1.22 – MC\_Halt v SCL

Využití této instrukce je možné pouze u poziční a rychlostní osy technologického objektu. Činnost MC\_Halt může přerušit následující činnosti instrukcí při navádění s módy 4 a 5, jinou zakazující instrukcí nebo v případě činnosti pohybových instrukcí (Siemens, 2014c).

Celkem lze při práci s touto instrukcí využít devět parametrů. Instrukce má jeden vstupně výstupní parametr *Axis*, tři vstupní parametry a pět výstupních parametrů. Vstupní parametry instrukce MC\_Halt jsou *Execute*, *Deceleration*, *Jerk* a *Mode*. Výstupní parametry jsou *Done*, *Busy*, *CommandAborted*, *Error* a *ErrorID*.



Obr. 1.23 – MC\_Reset v LAD

```

14 □ "MC_RESET_DB_1" (Axis:= multi_fb_in_,
15     Execute:=false,
16     Restart:=false,
17     Done=>_bool_out_,
18     Busy=>_bool_out_,
19     CommandAborted=>_bool_out_,
20     Error=>_bool_out_,
21     ErrorId=>_word_out_);
22

```

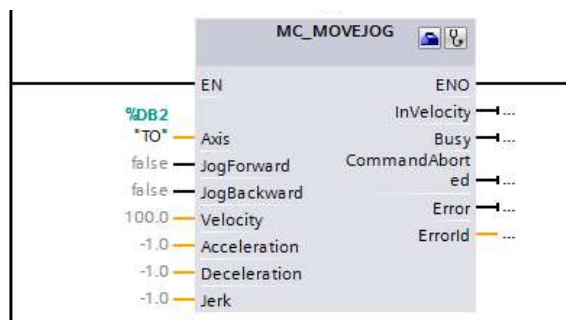
Obr. 1.24 – MC\_Reset v SCL

Instrukce MC\_Reset slouží k potvrzení všech varování a chyb v technologickém datovém bloku. Při vstupním parametru *Execute* jsou potvrzena všechna varování a navíc, pokud má i parametr *Reset* hodnotu 1, je provedena nová inicializace technologického objektu. Využití této instrukce je u všech typů technologického objektu. Celkem lze při práci s touto instrukcí využít devět parametrů. Instrukce má jeden vstupně výstupní parametr *Axis*, dva vstupní parametry a pět výstupních parametrů. Vstupní parametry instrukce MC\_Reset jsou *Execute*, *Restart*. Výstupní parametry jsou *Done*, *Busy*, *CommandAborted*, *Error* a *ErrorID* (Siemens, 2014c).

#### 1.4.6 Pohybové instrukce

Polohové instrukce slouží pro provádění pohybových funkcí, jako je pohyb určitou rychlostí, pohyb na pozici nebo pohyb pokud je parametr v hodnotě 1.

Většina polohových instrukcí využívá speciální vstupní parametry pro nastavení směru pohybu. Nastavit směr pohybu lze pomocí parametru *Direction*. Tento parametr je typu INT s výchozí hodnotou 0. Při výchozí hodnotě tohoto parametru je směr pohybu určován z nastavené rychlosti v parametru *Velocity*. Pokud hodnota parametru rovná 1, je nastaven kladný směr pohybu osy. Záporný směr pohyb znamená nastavení hodnoty parametru na hodnotu 2.



Obr. 1.25 – MC\_MoveJog v LAD

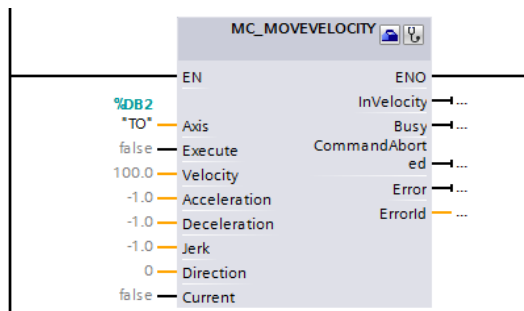
```

25
26 MC_MOVEJOG_DB (Axis:="TO",
27   JogForward:=false,
28   JogBackward:=false,
29   Velocity:=100.0,
30   Acceleration:=-1.0,
31   Deceleration:=-1.0,
32   Jerk:=-1.0,
33   InVelocity=>_bool_out_,
34   Busy=>_bool_out_,
35   CommandAborted=>_bool_out_,
36   Error=>_bool_out_,
37   ErrorId=>_word_out_);

```

Obr. 1.26 – MC\_MoveJog v SCL

Instrukce MC\_Jog slouží k pohybu osy technologického objektu. Tento pohyb oproti ostatním instrukcím je vykonáván pouze po dobu, dokud jsou parametry *JogForward* nebo *JogBackward* ve stavu 1 a pokud není signalizována chyba, ale stejně jako u ostatních instrukcí je daný pohyb vykonán definovanou rychlostí, zrychlením, zpomalením a trhnutím, definovaných v parametrech instrukce. Využití této instrukce je u všech typů technologického objektu kromě externího kodéru. Tato instrukce má jeden vstupně výstupní parametr *Axis*, pět vstupních parametrů a pět výstupních parametrů. Vstupními parametry jsou *JogForward*, *JogBackward*, *Velocity*, *Acceleration*, *Deceleration* a *Jerk*. Výstupními parametry jsou *InVelocity*, *Busy*, *CommandAborted*, *Error* a *ErrorID*. Vstupní parametry *JogForward* a *JogBackward* jsou typu BOOL a slouží pro informování instrukce, kdy má vykonávat pohyb v určitém pohybovém směru osy. Tento směr je definován, jak už z názvu vyplývá, dle zvoleného parametru. Tedy vstupní parametr *JogForward* slouží k spouštění pohybu v kladném směru pohybu a parametr *JogBackward* spouští pohyb v záporném směru osy. Při stavu 1 vstupního parametru se započne vykonávat pohyb a po vynulování parametru je tento pohyb dle nastavených parametrů ukončen. Výstupní parametr *InVelocity* poskytuje signalizaci informace o dosažení nastavené rychlosti pohybu osy a tato rychlost bude následně udržována. Toto signalizuje stavem 1. Výchozí hodnota tohoto parametru je 0 (Siemens, 2014c).



Obr. 1.27 – MC\_MoveVelocity v LAD

```

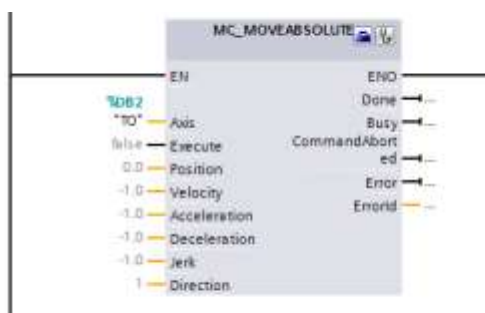
24 MC_MOVEVELOCITY_DB_1 (Axis:=multi_fb_in_,
25   Execute:=false,
26   Velocity:=100.0,
27   Acceleration:=-1.0,
28   Deceleration:=-1.0,
29   Jerk:=-1.0,
30   Direction:=0,
31   Current:=false,
32   InVelocity=>_bool_out_,
33   Busy=>_bool_out_,
34   CommandAborted=>_bool_out_,
35   Error=>_bool_out_,
36   ErrorId=>_word_out_);

```

Obr. 1.28 – MC\_MoveVelocity v SCL

Instrukce MC\_MoveVelocity umožňuje pohyb osy technologického objektu s konstantní rychlostí, dle definovaných dynamických parametrů trhnutí, zrychlení a zpomalení. Využití této instrukce je u všech typů technologického objektu, kromě externího kodéru. Instrukce MC\_MoveVelocity může přerušit nebo může být sama přerušena některými činnostmi. K těmto činnostem patří navádění v módu 4 a 5, zastavení osy a ostatní pohybové instrukce. Pokud nastavená nulová rychlost pohybu zastaví osa pohyb dle definovaného zpomalení a výstupní parametr *InVelocity* signalizuje 1, tedy dosažení maximální rychlosti pohybu (Siemens, 2014c).

Instrukce MC\_MoveVelocity má jeden vstupně výstupní parametr *Axis*, sedm vstupních parametrů a pět výstupních parametrů. Vstupními parametry instrukce jsou *Execute*, *Velocity*, *Acceleration*, *Deceleration*, *Direction* a *Current*. Výstupní parametry jsou *InVelocity*, *Busy*, *CommandAborted*, *Error* a *ErrorID*. K udržování aktuální rychlosti využívají pohybové instrukce vstupní parametr *Current*. Tento parametr je typu BOOL s výchozí hodnotou nula. Výchozí hodnota znamená, že aktuální rychlost není limitována hodnotou rychlosti při spuštění parametru *Current*. Tedy parametry jako rychlost a směr regulují dynamické vlastnosti řízení pohybu. Při nastavení parametru do polohy 1 se přestanou využívat vstupní parametry definující rychlost a směr pohybu a místo nich bude udržována rychlost a směr z chvíle spuštění instrukce (Siemens, 2014c).



Obr. 1.29 – MC\_MoveAbsolute v LAD

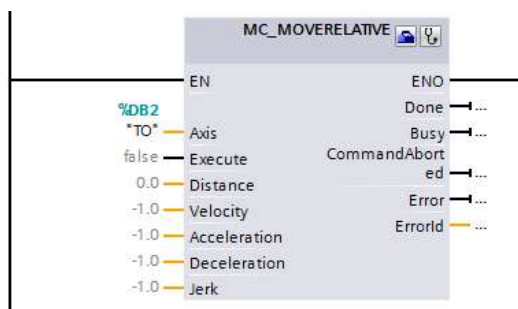
```

39 MC_MOVEABSOLUTE_DB_1 (Axis:=multi_fb_in_,
40   Execute:=false,
41   Position:=0.0,
42   Velocity:=-1.0,
43   Acceleration:=-1.0,
44   Deceleration:=-1.0,
45   Jerk:=-1.0,
46   Direction:=1,
47   Done=>_bool_out_,
48   Busy=>_bool_out_,
49   CommandAborted=>_bool_out_,
50   Error=>_bool_out_,
51   ErrorId=>_word_out_);

```

Obr. 1.30 – MC\_MoveAbsolute v SCL

Oproti instrukci MC\_MoveVelocity, pracující pouze s rychlostí, instrukce MC\_MoveRelative a MC\_MoveAbsolute umožňují dosažení přesné polohy na ose. Pro tyto polohové instrukce je nutné mít vytvořené navedení osy technologického objektu na nulovou pozici. Následně, dle nastavené polohy instrukce, vykonávají pohyb na danou pozici dle parametrů pro pohyb. Tyto parametry jsou rychlost, zrychlení, zpomalení a trhnutí. Tyto parametry se využívají pouze u poziční osy. Obě instrukce mohou přerušit nebo být přerušeny naváděcí instrukcí v módu 4 a 5, při zastavení pohybu nebo ostatní pohybové úlohy. Tyto instrukce pracují prakticky stejně. Liší se pouze v tom, že instrukce MC\_MoveRelative slouží k relativnímu pozicování a instrukce MC\_MoveAbsolute slouží k absolutnímu pozicování. Relativní pozicování znamená, že osa se pohybuje ve vzdálenosti Distance od pozice, kdy byla osa umístěna před aktivaci instrukce MC\_MoveRelative. Oproti tomu absolutní pozicování znamená pohyb osy vzhledem k nulovému bodu vytvořenému při navádění osy pomocí instrukce MC\_Home.



Obr. 1.31 – MC\_MoveRelative v LAD

```

55 MC_MOVERELATIVE_DB_1 (Axis:=_multi_fb_in_,
56   Execute:=false,
57   Distance:=0.0,
58   Velocity:=-1.0,
59   Acceleration:=-1.0,
60   Deceleration:=-1.0,
61   Jerk:=-1.0,
62   Done=>_bool_out_,
63   Busy=>_bool_out_,
64   CommandAborted=>_bool_out_,
65   Error=>_bool_out_,
66   ErrorId=>_word_out_);

```

Obr. 1.32 – MC\_MoveRelative v SCL

Instrukce MC\_MoveRelative a MC\_MoveAbsolute obsahuje každá jeden vstupně výstupní parametr *Axis*, šest vstupních parametrů a pět výstupních parametrů. Vstupními parametry instrukce jsou *Execute*, *Distance*, *Velocity*, *Acceleration* a *Deceleration*. Výstupní parametry jsou *InVelocity*, *Busy*, *CommandAborted*, *Error* a *ErrorID*.

### 1.2.7 Vkládání instrukcí do projektu

Všechny instrukce pro práci s regulovanými pohony jsou dostupné v otázce *Instructions* a složce *Technology*. Podsložka *Technology* obsahuje různé typy instrukcí určené pro práci s různými prvky od měření teploty po PID regulátory. Ve složce *Motion Control* jsou obsaženy všechny instrukce pro práci s regulovanými pohony.

Možností vložení instrukce do programu jsou dvě. Buď dvakrát kliknout na zvolenou instrukci v seznamu instrukcí, ale je nutné označit místo vložení instrukce v programu nebo přetáhnutím instrukce ze seznamu instrukcí do požadovaného místa umístění instrukce. Tuto instrukci je nutné při vkládání do projektu propojit s datovým blokem. Po dokončení

vložení instrukce program zobrazí podokno nazvané *Call option*, kde lze nastavit název datového bloku propojeného s instrukcí, sloužící jako uložisko dat této instrukce. Nově vytvořený datový blok se uloží v programové struktuře projektu ve zvláštní složce *Program resources*, jež je podsložkou ve složce *Systém blocks*. Je-li podokno *Call options* přerušeno, tak je instrukce vložena bez datového bloku. Instrukce je zobrazena stejně, ale nad ní místo názvu propojeného datového bloku je zobrazeno „<??>“. Tento znak znamená nepropojenost instrukce s datovým blokem a instrukce by pracovala špatně. V tomto případě je nutné dodatečně propojit instrukci s již existujícím datovým blokem. Propojení se provede pomocí dvojkliku nad instrukcí a zvolením ze seznamu již existujícího datového bloku. Po dokončení je nad instrukcí místo neznámého bloku zobrazeno jméno propojeného bloku.

## 2 PRAKTICKÁ ČÁST

Program má celkově sloužit jako modelový univerzální program, proto je navržen tak, aby byl univerzální, snadno modelovatelný pro případné další použití tohoto programu a snadný na porozumění. Snadné ovládání tohoto programu je zajištěno pomocí vizualizace provedené pomocí HMI Comfort panelu 7<sup>6</sup>. K tomuto ovládání jsou využity znaky v HMI panelu, které jsou přímo propojené se znaky v PLC.

### 2.1 VYTVOŘENÍ A NASTAVENÍ PROJEKTU V TIA PORTAL

Závislost Před začátkem práce je nutné nastavit program a projekt. Já jsem se rozhodl pro nastavení jazyka programu TIA Portal V13 na angličtinu z důvodů mé znalosti tohoto jazyka. Dále jsem zvolil mezinárodní mnemoniku, což vychází také ze znalosti mnemonických zkratk v tomto jazyce, kdy tyto zkratky vycházejí z anglického jazyka.

Dále je nezbytné vytvořit nový projekt pro práci. Vytvořit nový projekt lze z portálového pohledu, s využitím otázky Start a zde funkce *Create new project*, nebo v projektovém pohledu, umístěném v záložce *Project* a zde funkce *New*. Po aktivaci jedné z funkcí se otevře podokno *Create a new project*. V tomto podokně lze zadat název nového projektu, umístění pro uložení projektu, jméno autora a další komentář týkající se nově vytvořeného projektu. Po potvrzení se začne vytvářet nový projekt a lze vytvářet program, kdy je nutné nejprve vytvořit hardwarovou konfiguraci. Následná práce je vedena hlavně v projektovém pohledu. V případě další práce v portálovém pohledu je postup činnosti obdobný.

### 2.2 HARDWARE

V obecné hardwarové koncepci jsem se rozhodl pro zvolení propojovacího rozhraní PROFINET. Toto síťové rozhraní jsem zvolil z důvodů snadného propojení se zařízením, a jelikož můj počítač není vybaven kartou pro práci se síťovým rozhraním PROFIBUS. Případné vypůjčení této karty se z důvodu možné konfigurace s připojením k mému počítači nevyplatí. Pro práci se síťovým rozhraním PROFINET stačí běžná síťová karta, kterou je vybaven v dnešní době každý počítač. Toto rozhraní je vlastně průmyslový ETHERNET a tím i práce pro běžného uživatele je s ním snadnější. Prakticky stačí k dané zařízení připojit pomocí ETHERNET kabelu k počítači. Následně v TIA PORTAL se dané zařízení zobrazí v *Project view* ve *Project tree* a zde ve složce *Online acces*. Zde je zobrazeno se jménem

a v závorce s jeho statickou IP adresou. Občas nastane problém s dynamickou IP adresou a je nutné nastavit na počítači statickou IP adresu. Tato IP adresa, pokud počítač není připojen k jinému zařízení nebo síti, může být zcela libovolná a jediné omezení nastává, je-li počítač připojen k dalším zařízením. Další problém může přinést brána firewall nebo antivirová ochrana. Tento problém vzniká v závislosti na používaném bezpečnostním softwaru a lze odstranit vypnutím ochranného bezpečnostního prvku nebo povolením aplikace v bezpečnostním softwaru.

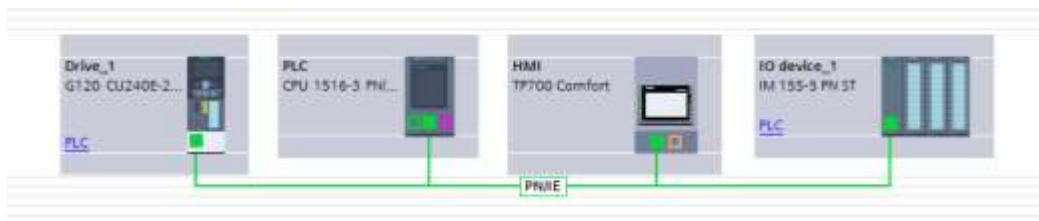
Pro vykonávání programu je určeno PLC SIMATIC S7-1500. Toto PLC je pro tento konkrétní projekt vhodné z důvodů velké škály modulů a tím vznikající univerzální škály použití ve všech druzích zařízeních. Toto PLC má odpovídající výkon i možnosti komunikace. Další výhodou je, že toto zařízení je vybaveno PROFINET portem, které lze propojovat i klasickým ETHERNET kabelem a odpadá problém s případnými propojeními. K tomuto zařízení je připojen HMI panel COMFORT 7“. COMFORT panel je propojen pomocí rozhraní PROFINET. PLC je propojeno s frekvenčním měničem SINAMICS pomocí PROFINET.

Topologie zapojení hardwaru je vytvořena v programu TIA PORTAL V13. Zde výběrem podsložky *Hardware & Networks* v *Project tree* se zobrazí graficky aktuální hardwarová konfigurace. Vložení zařízení lze dosáhnout vložení zvoleného zařízení z otázkové karty *Hardware catalog*. Zde zvolením vhodného zařízení, kdy identifikační číslo daného zařízení musí odpovídat zvolenému zařízení v *Hardware catalog*. Přidat zařízení z *Hardware cataloge* lze přetažením na požadované místo v *Device & networks* nebo dvojklikem na zvolené zařízení. Po přidání zařízení se dané zařízení zobrazí v *Devices & networks* jako položka a současně je zobrazeno v *Project tree* ve složce projektu. V případě projektu se vloží z hardwaru kontrolér z rodiny S7-1500 s identifikačním číslem 6ES7 516-3AN00-0AB0, HMI panel s identifikačním číslem 6AV2 124-0GC01-0AX0 a nakonec frekvenční měnič SINAMICS s identifikačním číslem 6SL 3210-1SB11-0AA0. Pro snímání polohy je na modelu umístěn ET200 s číslem 6ES7 550-1AA00-0AB0.

Vložená zařízení je nutné síťově spojit dle zvoleného síťového rozhraní, proto jednotlivá zařízení se spojí pomocí síťového rozhraní PROFINET. Na zařízení je zobrazen PROFINET port zeleně a PROFIBUS oranžově. Spojení se provede pouze pomocí tažením z jednoho portu na další port rozhraní. Pro komunikaci PLC a HMI panelu je nutné ještě provést *HMI connection*. *HMI connection* se provede pomocí vybrání funkce *Connection* v listě editoru *Devices & networks* a je nutné vybrat funkci *HMI connection* ve výběrovém poli vedle funkce *Connections*. Spojení se provede pomocí tažení spojovacího porty PROFINET HMI a PLC. Poslední akcí pro dokončení funkce je nutné přiřazení frekvenčního měniče PLC. Přiřazení



může být provedeno za pomoci zvolením funkce *Not assigned*, která je umístěna na frekvenčním měniči v *Device & networks*.



Obr. 2.1 – Hardwarová konfigurace

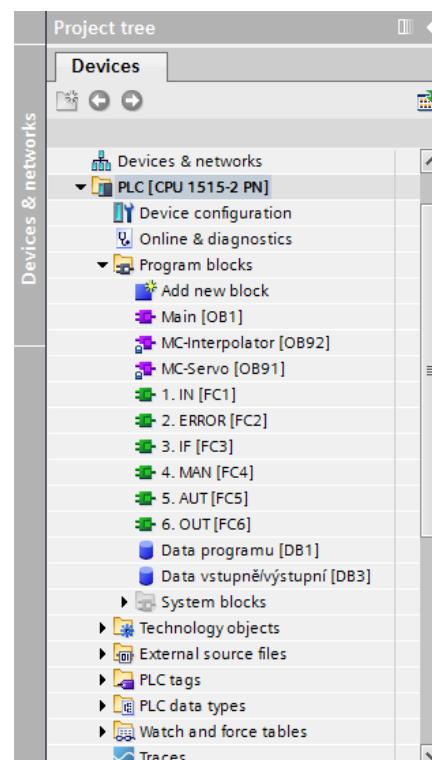
## 2.3 PROGRAM PRO PLC

Závislost Program pro PLC je napsán v jazyce LAD z důvodů názornosti a jednoduchého pochopení i pro osobu neznalou programovacích jazyků. Jazyk LAD je grafický programovací jazyk. Syntaxe jazyka LAD odpovídá obvodovému diagramu, umožňuje jednoduché sledování toku informací a programu. Tento jazyk je velmi výhodný při online sledování průběhu dat v programu. Také syntaxe tohoto jazyka odpovídá tlačítkům a blokům. Všechny programové bloky jsou umístěny v PLC, které má vykonávat daný program. Prakticky je lze nalézt v Projektovém stromě v určitém programovatelném průmyslovém automatu v podsložce *Program blocks*.

### 2.3.1 Programová struktura

Program je založen na cyklickém vykonávání hlavního bloku OB1, který má zkrácené označení Main. Tento programový blok je vykonáván cyklicky a z tohoto bloku jsou volány další programové bloky, tvořící celkový program. Z tohoto cyklicky vykonávaného programu jsou volány jednotlivé funkce, které jsou zmíněny výše.

Z důvodu obecného použití jednotlivých funkčních bloků jsou k tvorbě programu využity funkce, které oproti funkčním blokům mají výhodu jejich obecného použití, protože neudržují globální data v datovém bloku. Funkční programová struktura je zobrazena na obr. 2.2. Program je z důvodu univerzálnosti rozdělen do šesti



Obr. 2.2 – Programové bloky

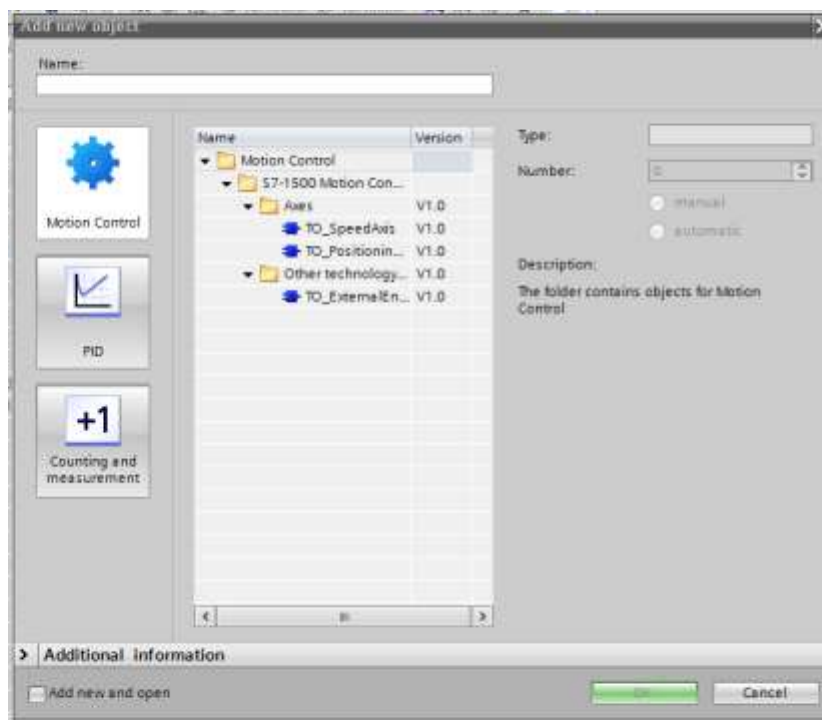
základních funkcí. Všechny tyto funkce jsou podmíněny pomocí podmínkového datového bloku jehož obsah je ovládán podmínkovým blokem IF. Data z funkcí jsou obsazena v datovém bloku pojmenovaném Data programu a Data vstupně/výstupní. Tyto funkce jsou IN, ERROR, IF, MAN, AUT a OUT.

Každý řádek kódu je okomentován, i když to odporuje zásadě čistého kódu, používané v programování některých programovacích jazyků. Komentáře jsou výborné pro jiného programátora nebo vrací-li se programátor k programu po delší době. Porozumění programu lze usnadnit použitím komentáře. Při použití LAD jazyka v TIA PORTAL je pod každým networkem umístěn prostor pro komentáře programátora. Programátor by měl každý network okomentovat tak, aby komentář vysvětlil funkci daného networku.

### **2.3.2 Konfigurace technologického objektu**

Před nahráním programu do PLC je nezbytné správně nakonfigurovat technologický objekt z důvodu, že programová data technologického objektu jsou nahrána do příslušného PLC. PLC řídí dle těchto nakonfigurovaných dat řízené zařízení.

Technologický objekt lze vložit v projektové složce *Project tree*. Zde v daném PLC řídicí a vykonávající program pro frekvenční měnič SINAMICS je podsložka *Technology objects*. Podsložka obsahuje všechny využívané technologické objekty a současně obsahuje funkci pro přidání technologického objektu. Aktivací této přidávací funkce pro technologické objekty, v programu označené *Add new object*, se zobrazí podokno *Add new object*. V tomto okně lze vybrat požadovaný technologický objekt. K dispozici jsou technologické objekty pro řízení pohybu, PID regulátor, čítací a měřící reulátor. Zvolením technologického objektu pro řízení pohybu je označován *Motion Control*. Následně výběrem požadovaného technologického objektu ze seznamu dostupných technologických objektů se zobrazí popis daného objektu. Tento přidávaný objekt je dobré pojmenovat dle požadovaného jména v položce *Name*, okna *Add new object*. Po potvrzení tlačítkem OK se dokončí přidávání technologického objektu.



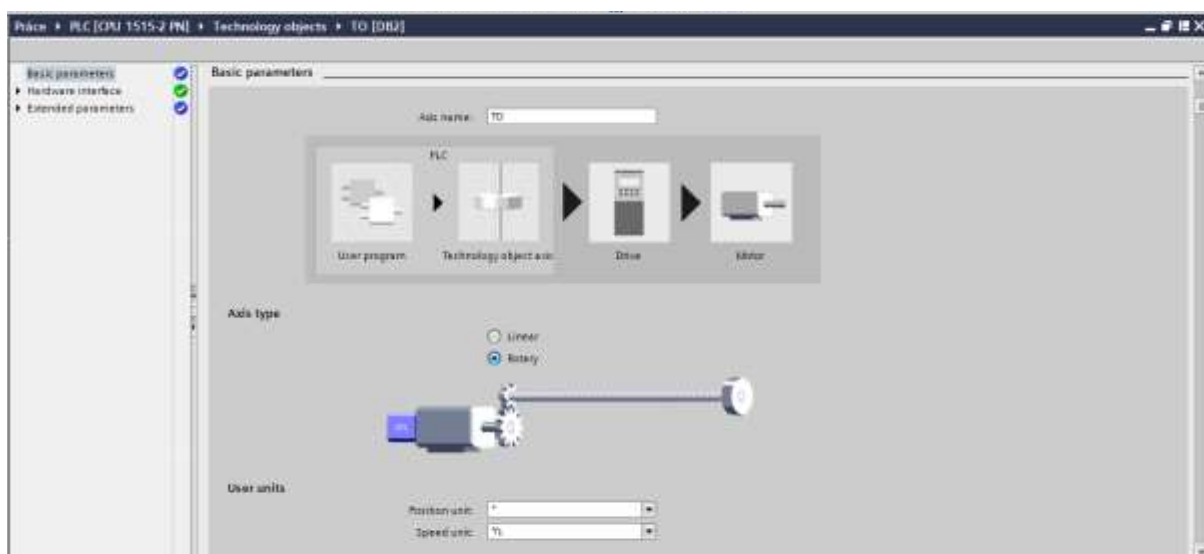
Obr. 2.3 – Přidání nového objektu

Následně je nezbytné technologický objekt nastavit. Nastavení lze spustit z podsložky *Technology objects*, ve které jsou zobrazeny všechny technologické objekty. Otevření složky požadovaného technologického objektu se zobrazí dostupné funkce pro tento typ objektu. V případě instrukcí pro řízení pohybu tyto funkce jsou *Configuration*, *Commissioning* a *Diagnostics*. *Diagnostics* slouží k online diagnostice daného zařízení. *Commissioning* je funkce, ze které lze přímo ovládat pohon bez nutnosti využívat program. Poslední funkce *Configuration* slouží ke konfiguraci technologického objektu a tedy výběrem této funkce se začne nastavovat objekt. Konfigurační editor je zobrazen na obr. 2.4. Zde jsou vidět jednotlivé funkce pro nastavení technologického objektu.

První je Basic parametr, kde lze nastavit jméno technologického objektu v položce *Axis name*. Dále je nezbytné nastavit typ osy pohonu v *Axis type*, tedy, zda se jedná o osu s lineárním pohybem nebo s rotačním pohybem. Tento parametr musí být nastaven dle reálných vlastností osy. Pod typem osy se nastavují jednotky a limity osy pohybu. Jednotky pro pozici jsou stupně nebo radiány. Od pozičních jednotek se odvíjejí jednotky pro rychlost, které jsou stupeň za sekundu nebo stupeň za minutu při nastavení stupně. Pokud se nastaví radián, tak v rychlostních jednotkách bude stupeň nahrazen radiány. Modulo slouží pro nastavení limitů polohové osy. V případě tohoto projektu je nezbytné nastavit rotační osu, jednotky na stupně a stupně za sekundu a deaktivovat modulo.

Druhý je *Hardware interface*. Ve *Drive* je nutné nastavit řízený pohon a rozhraní, pokud by nebylo nastaveno. Řízení pohonu by nefungovalo. Pod *Drive* se nachází *Encode*, ve kterém se nastavuje nastavení snímacího kodéru polohy. Zda bude technologický objekt přijímat data z frekvenčního měniče, z technologického modulu nebo z kodéru. Na konci této položky se nastavuje přenosový rámec pro komunikaci s pohonem. V tomto případě se nastavuje pouze spojení přímo z frekvenčního měniče další položky není třeba nastavovat.

Poslední položkou je *Expanded parameters*. Zde se nastavují hardwarové limity polohy a dynamické limity pohybu. V tomto případě tak není potřeba zásadně měnit parametry.



Obr. 2.4 – Nastavení technologického objektu

### 2.3.3 Hlavní blok MAIN

Blok OB1, pojmenovaný MAIN je blok, který je spouštěn při startu zařízení. Tento blok je vykonáván cyklicky, vždy po načtení vstupů. Využití tohoto bloku je v téměř každém programu. Tento blok patří do skupiny systémových programových bloků, kdy jsou tyto systémové bloky volány pouze při určité příležitosti. Touto příležitostí může být například start zařízení, cyklické vykonávání programu, cyklické přerušení, hardwarové přerušení a mnoho dalších. Pro každou z těchto příležitostí existuje systémový blok OBx. Z těchto důvodů je žádáno, aby po startu programu nebo začátku činnosti zařízení se začal vykonávat program od systémového bloku MAIN, značeného v programu OB1.

Z bloku OB1 jsou postupně volány, v každém networku jednotlivé funkce, které tvoří základní bloky programu, vytvořeného programátorem. Při vykonávání programu jsou postupně v každém networku volány uživatelem vytvořené funkce FCx. Do tohoto bloku lze dále přidávat další funkce, které celkově mohou obohatit již existující program. Tím odpadá nutnost přímo zasahovat do kódu funkcí.



Obr. 2.5 – Network 1 v cyklickém bloku OB1

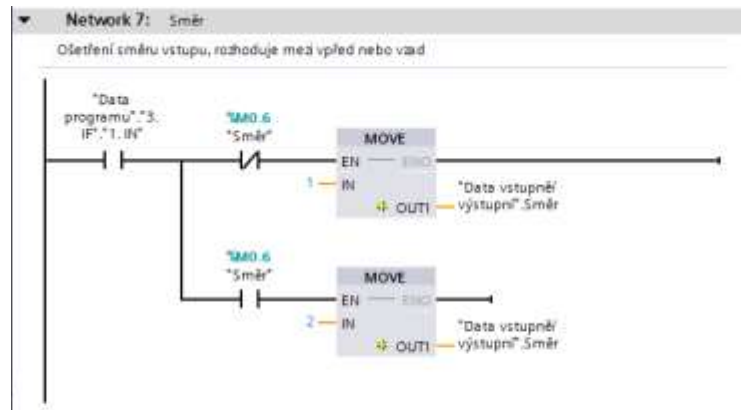
### 2.3.4 Vstupní blok 1. IN

Vstupní blok je zde umístěn z důvodu zpracování vstupních dat a informací. V případě tohoto programu tento blok není příliš využit. Tento blok je umístěn hlavně z důvodu obecnosti kódu. Protože v případě přidávání dalších vstupů by bylo nutné pro tyto vstupy vytvářet zcela nové funkce a mohl by nastat problém se zakomponováním do celé programové struktury. Vstupní obvod obsahuje pouze přenos informace ze znaku do datového bloku DB3. Toto je provedeno hlavně z důvodu, aby bylo možné k programu v budoucnu přidat ovládání z dalšího panelu nebo digitálních vstupů. Dalším účelem vstupního bloku je ošetřit případné vstupy při chybě v případě rozlišení programu.



Obr. 2.6 – Network 1 ve funkci 1. IN

Jediná změna využití nastává v případě vstupu přepínače určujícího směr pohybu. Přepínač, umístěný na HMI panelu, je propojen se znakem typu BOOL, kdy dle stavu znaku je definován směr pohybu. K tomuto jsou využívány dvě funkce MOVE. Spouštění jedné ze dvou funkcí je provedeno spínacím a rozpínacím kontaktem.



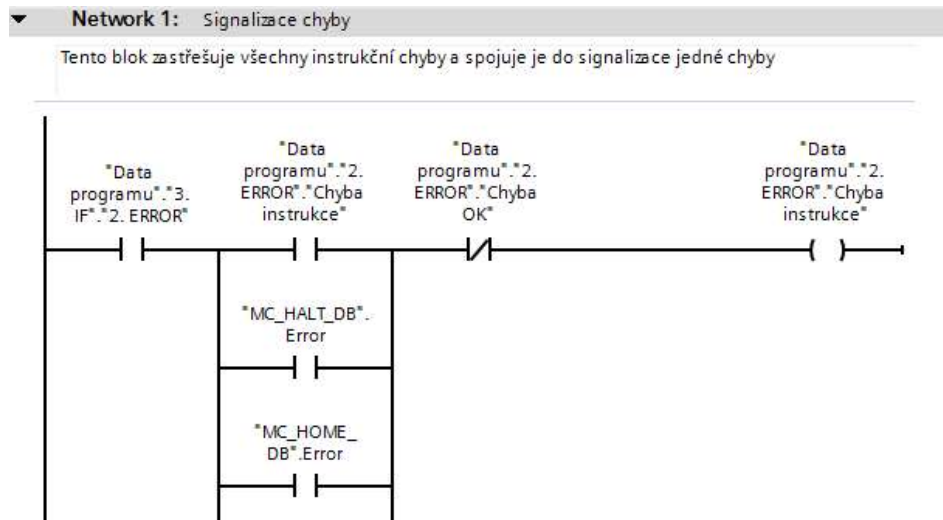
Obr. 2.7 – Network 9 ve funkci 1. IN

### 2.3.5 Blok chyb 2. ERROR

Blok chyb se stará o vyhodnocování a reakci na chyby vzniklé při běhu programu. Hlavním úkolem tohoto bloku je vyhodnotit vstupní, výstupní a stavová data z programu. Následně reagovat odpovídajícím způsobem na vzniklou situaci. Odpovídající reakce může být chyba nebo pouze varování ostatních bloků programu. Další důležitý úkol tohoto bloku je vzniklé chybě nebo varování přiřadit kód a uložit informaci o chybě do proměnné, která bude později využita v HMI panelu pro informování uživatele o vzniklé chybě.

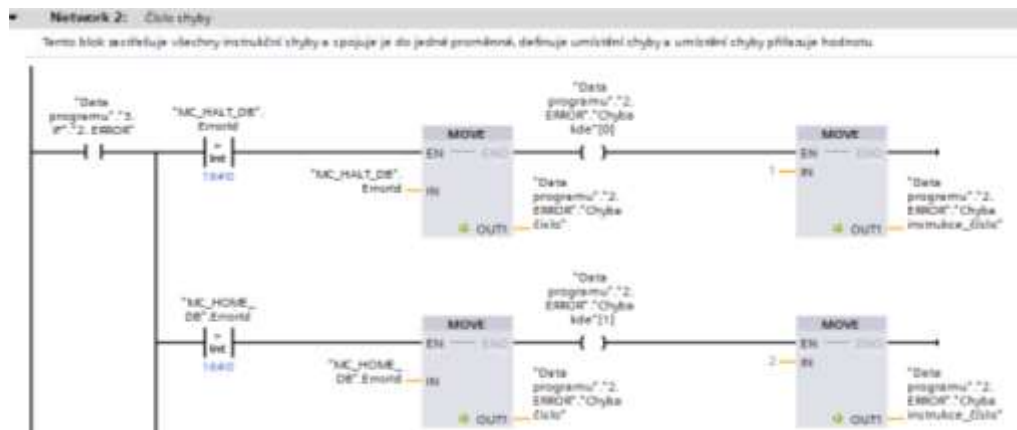
Pro uložení dat tohoto bloku je vytvořena nová datová struktura v datovém bloku DB1, který je globálním datovým blokem. Tato datová struktura souboru obsahuje všechna data bloku chyb. Tuto datovou strukturu je možno nahradit pomocí globálních znaků v programu. K tomuto jsem se nerozhodl z důvodu zvýšení nepřehlednosti v těchto globálních tazích a i z důvodu následného a snadného přenosu programu do jiného programu.

Network 1 funkce 2. ERROR je zobrazen na obr. 2.8 a funguje tak, aby spojoval jednotlivé chyby na jednotlivých instrukcích v jednu globální signalizační proměnnou typu BOOL pro chybu. V tomto networku je využíván samo držící princip, který se využívá též u relé nebo stykačů. Samo držící princip funguje tak, že výstup se drží sám po aktivaci ve stavu 1. Přerušit tento stav výstupu lze pouze přerušovacím kontaktem. Samo držící výstup představuje globální signalizaci chyby na obr. 2.8 jako „Data programu“. „2. ERROR“. „Chyba instrukce“. Přerušovací kontakt představuje potvrzovací vstup a výstupní bit je spuštěn pomocí chyby na některé z pohybových instrukcí.



Obr. 2.8 – Část network 1 ve funkci 2. ERROR

Network 2 funkce 2. ERROR je zobrazen na obr. 2.8 a funguje tak, aby spojoval jednotlivá identifikační čísla chyb na jednotlivých instrukcích v jednu globální identifikační proměnnou typu WORD. Toto je vytvořeno hlavně kvůli HMI panelu, protože panel má signalizovat jednu chybu a ne více chyb současně. Network, pokud je hodnota na dané instrukci větší jak 0, tak hodnotu této chyby přemístí pomocí instrukce MOVE do globální identifikační proměnné. K zrušení této chyby je využíván network 3. A network, pokud se v obvodu nevyskytuje chyba a obsluha chybu potvrdí, aktivuje MOVE instrukci, která místo identifikačního čísla chyby vloží 0.



Obr. 2.9 – Část network 2 v bloku 2. ERROR

### 2.3.6 Podmínkový blok 3. IF

Tabulky Podmínkový blok IF obstarává podmiňování funkce jednotlivých bloků. To je provedeno pomocí spínacího kontaktu na začátku každé sítě v ostatních funkcích programu. Tyto podmiňovací data jsou uložena v globálním datovém bloku, v datovém bloku

DB1 Data programu, která zahrnují podmiňující bity pro každou funkci. Zvolil jsem variantu podmiňujících bitů z důvodu jednoduchosti použití v programu a jednoduchého přidávání dalších podmínkových bitů pro případné další programové bloky, které lze do projektu přidat.

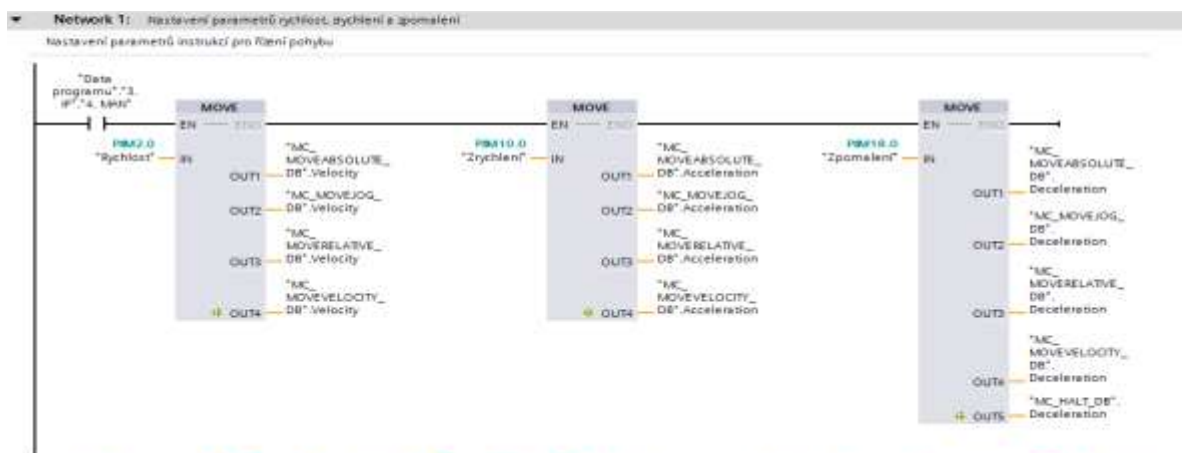
Program v případě tohoto bloku je velmi jednoduchý, protože je zde požadavek, aby byla všem funkcím umožněna vždy činnost. Z tohoto důvodu je pro každý blok vytvořený network. Tento network je přímo vyveden na výstupní bit a tím je tento bit vždy nastaven do 1. Detailně je network 1 podmiňovací funkcí zobrazen na obr. 2.10.



Obr. 2.10 – Ukázka z IF funkce

### 2.3.7 Blok manuálního řízení 4. MAN

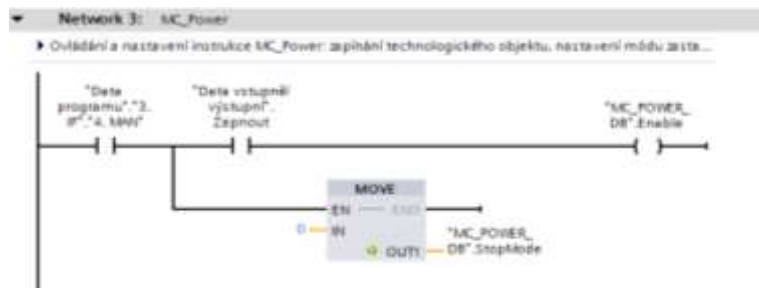
Blok manuálního řízení obstarává funkce manuálního řízení programových funkcí. Na začátku v network 1 a 2 jsou nastaveny dynamické parametry pohybu pro instrukce MC\_MoveAbsolute, MC\_MoveJog, MC\_MoveRelative, MC\_MoveVelocity a MC\_MoveHalt. Tyto nastavované parametry jsou rychlost, zrychlení, zpomalení, trhnutí a pozice. K přiřazení parametrů ze vstupně/výstupního datového bloku DB3 se využívá přenosová datová funkce MOVE. K hlavním výhodám funkce MOVE je možnost jeden vstup přenést na několik výstupů. Vstupním přenášeným parametrem je hodnota nastavená na HMI panelu. Příklad je zobrazen na obr. 2.11.



Obr. 2.11 – Network 1 ve funkci 4. MAN



Následně jsou nastavovány a ovládány jednotlivé parametry pro pohybové instrukce, které jsou umístěny v bloku automatického řízení. Příklad tohoto nastavení je zobrazen na obr. 2.12.



Obr. 2.12 – Network 3 ve funkci 4. MAN

### 2.3.8 Blok automatické regulace 5. AUT

Blok automatické regulace obstarává automaticky vykonávané činnosti a autonomní činnosti běžící automaticky na pozadí programu. Tyto autonomní úkony regulují výsledné výstupní zařízení dle stavu vstupních dat. Tento blok proto obsahuje všechny instrukce pro regulaci pohonu. Tyto instrukce jsou umístěny v network 1 až 9. Pro přehlednost každý network obsahuje jednu instrukci.

Network je na začátku ošetřen otevřeným kontaktem, který povoluje činnost této instrukce. Za kontaktem následuje instrukce, která se stará o povolovací funkce pro technologický objekt v práci označený TO. Tato instrukce je propojena s automaticky vytvořeným datovým blokem. Parametry dané instrukce jsou nastavovány v jiných blocích pomocí přiřazení dat do datového bloku instrukce. Nastavování parametrů je prováděno primárně ve funkci pro manuální ovládání 4. MAN. Toto hlavně zjednodušuje celkový program, protože nebude v programu zbytečně mnoho znaků a bude se pracovat pouze s jedním datovým blokem.

Například obr. 2.13 zobrazuje případ pro první network. Tento network obsahuje spínací kontakt a instrukci MC\_Power. Všechny ostatní networky pracují na stejném principu, pouze vykonávají odlišné funkce.



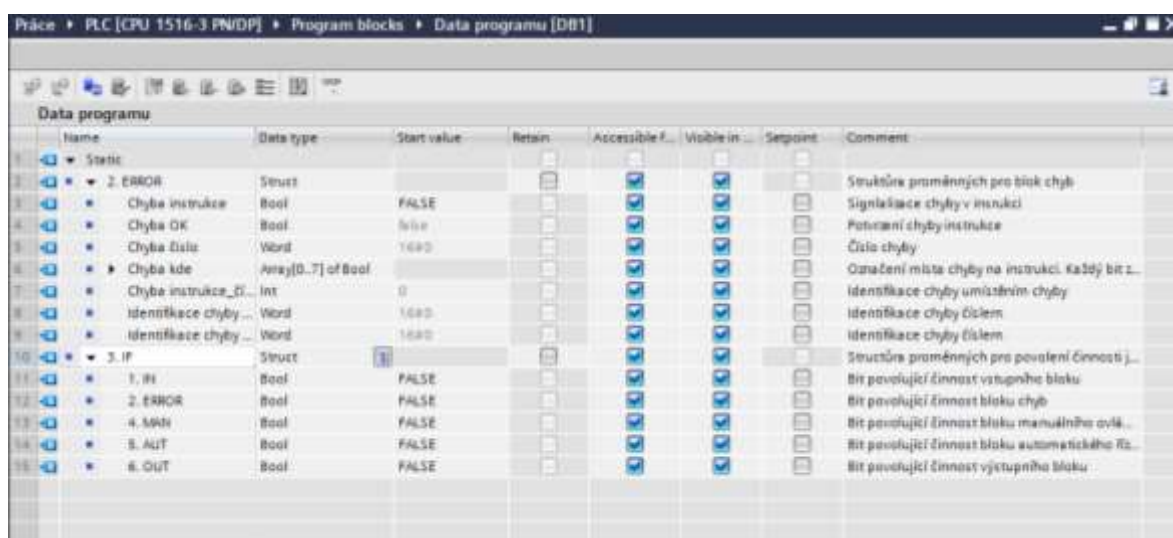
Obr. 2.13 – Network 1 ve funkci 5. AUT

### 2.3.9 Výstupní blok 6. OUT

Výstupní blok se stará o obsluhu výstupů, transformaci výstupních veličin na požadované hodnoty a případně odpojení výstupu při možnosti vzniku chyby, kdy by hrozilo poškození zařízení. V práci není využit z důvodu minimálního využití výstupních veličin.

### 2.3.10 Data programu

Tento datový blok slouží pro ukládání a práci s programovými daty. Pro jeho vnitřní přehlednost obsahuje několik datových struktur, kdy je vždy jedna vytvořena pro jednu funkci. Data ve strukturách jsou uložena hlavně z důvodu následné přehlednosti výsledného kódu, protože pokud programátor hledá určitou proměnnou, nemusí prohledávat všechny znaky nebo položky datového bloku. Pouze rozklikne příslušnou strukturu dané funkce a pracuje s těmito daty. Tento blok je možné nahradit i pomocí tagů, které z důvodu snadné přenositelnosti kódu a přehlednosti jsou použity v omezené míře.



Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Segment	Comment
Static							
2. ERROR	Strukt			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Struktura proměnných pro blok chyb
Chyba instrukce	Bool	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Signalizace chyby v instrukci
Chyba OK	Bool	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Potvrzení chyby instrukce
Chyba číslo	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Číslo chyby
Chyba kde	Array[0..7] of Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Označení místa chyby na instrukci. Každý bit...
Chyba instrukce_0...	Int	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Identifikace chyby umístěním chyby
Identifikace chyby...	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Identifikace chyby číslem
Identifikace chyby...	Word	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Identifikace chyby číslem
3. IF	Strukt			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Struktura proměnných pro povolení činnosti j...
1. IN	Bool	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Bit povolující činnost vstupního bloku
2. ERROR	Bool	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Bit povolující činnost bloku chyb
4. MAN	Bool	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Bit povolující činnost bloku manuálního ovlá...
5. AUT	Bool	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Bit povolující činnost bloku automatického ří...
6. OUT	Bool	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Bit povolující činnost výstupního bloku

Obr. 2.14 – Datový blok DB1

### 2.3.11 Data vstupně/výstupní

Tento datový blok slouží pro ukládání a práci se vstupními a výstupními daty, která jsou v programu použita. Tento datový blok slouží spíše pro obecné použití vstupních signálů z HMI panelu. Pro zjednodušení lze tento datový blok vynechat.

## 2.4 VIZUALIZACE

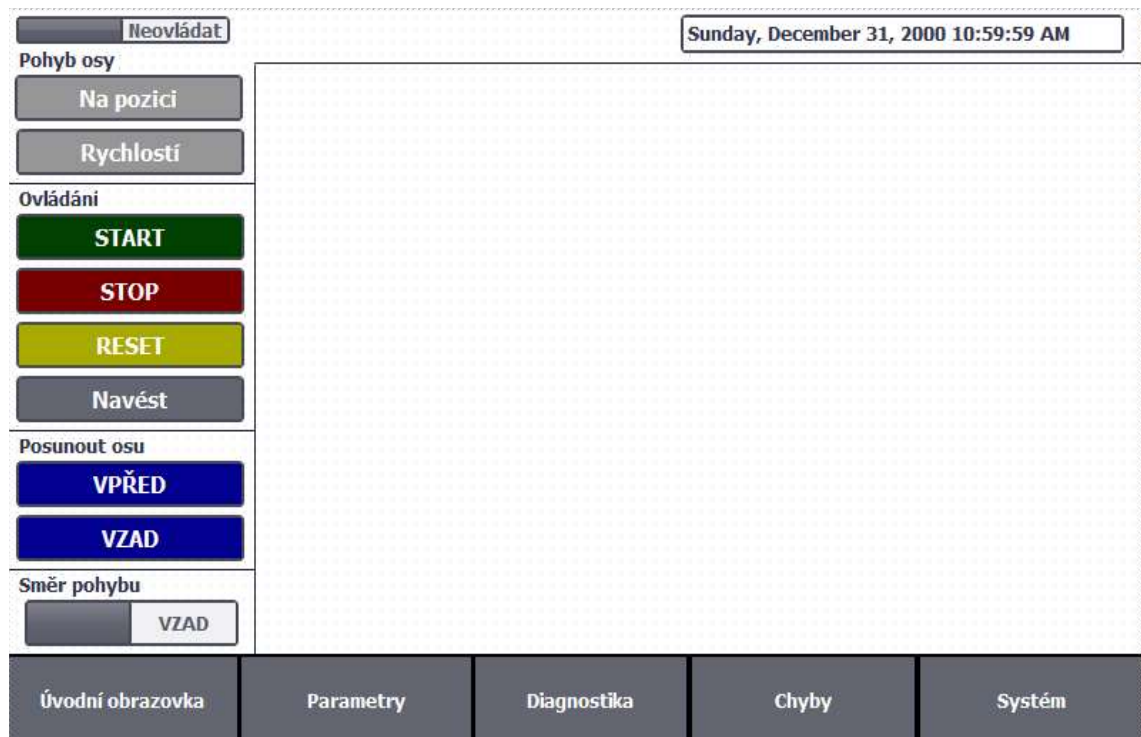
Vizualizace pro vytvořený program obsahuje vizualizované obrazovky pro ovládání pohonu, aktuální informace o činnosti programu, hlášení chyb včetně popisu chyby, aktuální data z pohonu a grafickou signalizaci činnosti programu. Tato vizualizace je primárně určena pro 7“ display SIMENS SINAMIC HMI Comfort, který je vybaven dotykovou obrazovkou. Případná konverze tohoto programu na jiný typ displeje je velmi jednoduchá a stačí pouze malá úprava již vizualizovaného programu a náhrada hardwaru v hardwarové konfiguraci v programu TIA Portal V13. Jediný problém při převodu na jiný hardware může nastat v případě, že nový hardware nebude podporovat dotykové ovládání. V tomto případě je nutná větší úprava vizualizovaného programu, která bude limitována počtem tlačítek nového displeje.

### 2.4.1 Šablony

Nejprve je nezbytné vytvořit si šablonu pro obrazovky. Tato šablona je výhodná z důvodů usnadnění následné vizualizace programu. Toto zjednodušení znamená, že nebudeme muset pro každou obrazovku vytvářet znovu stejné prvky a ty následně nastavit. Program TIA Portal tyto šablony podporuje a je zde možno vytvořit i velké množství těchto šablon. V tomto programu budou stačit dvě univerzální šablony. Jedna pro ovládání a další pro systémové obrazovky.

Šablona `Template_1` obsahuje základní rozložení vizualizovaných stránek. Vizualizovaná stránka obsahuje tlačítka pro ovládání pohonu, rozložení stránek a základní pohyb po obrazovkách. Úplně nad tlačítka je obsažen přepínač, který odpojí tlačítka od ovládání pohonu a ten lze ovládat pouze pomocí diagnostických obrazovek pomocí instrukcí. Při aktivaci ovládání je ovládan dle tlačítek a nastavených parametrů. Tlačítka pro ovládání pohonu na většině obrazovek jsou z důvodu, aby operátor zařízení mohl kdykoliv ovládat pohon bez nutnosti přepínání mezi obrazovkami. U těchto ovládacích tlačítek je využita animace. Animace jsou vytvořeny tak, aby zvýraznily stav daného tlačítka. Například, je-li pohyb zařízení zastaven pomocí tlačítka STOP, je tlačítko zvýrazněno světle červenou barvou. Obdobně pracují další tlačítka. Animace má sloužit hlavně pro informování operátora zařízení o stavu zařízení. Ovládací tlačítka, ovládající znaky v PLC, slouží pro řízení programu. Dole pod ovládacími tlačítky je umístěn přepínač. Tento přepínač přepíná směr pohybu. Rozložení stránek je vytvořeno pouze pomocí statických černých čar, které mají vyznačovat základní rozložení těchto stránek. Aktuální hodiny umístěné vpravo nahoře slouží obsluze pro informaci o čase a aktuálním datu. Dole je umístěna lišta pro pohyb mezi vizualizovanými stránkami

projektu. Přepínání mezi obrazovkami je vytvořeno pouze pomocí *Events*, kdy po stisku tlačítka je využita funkce *ActivateScreen*. Tato funkce aktivuje vybranou obrazovku.



Obr. 2.15 – Šablona Template\_1

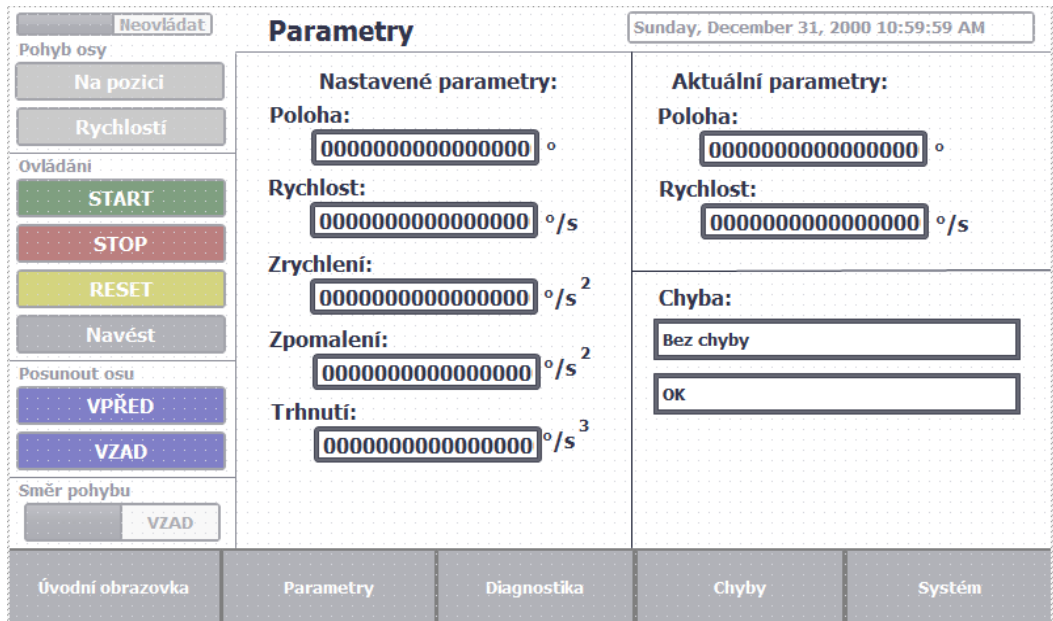
Šablona Template\_2 je vizualizovaná obdobně jako šablona Template\_1. Největší rozdíl mezi těmito šablonami je, že místo ovládacích tlačítek a přepínače směru obsahuje tlačítka pro systémové obrazovky. Tyto systémové obrazovky se využívají z důvodů snadnější práce s PLC.

#### 2.4.2 Start obrazovka

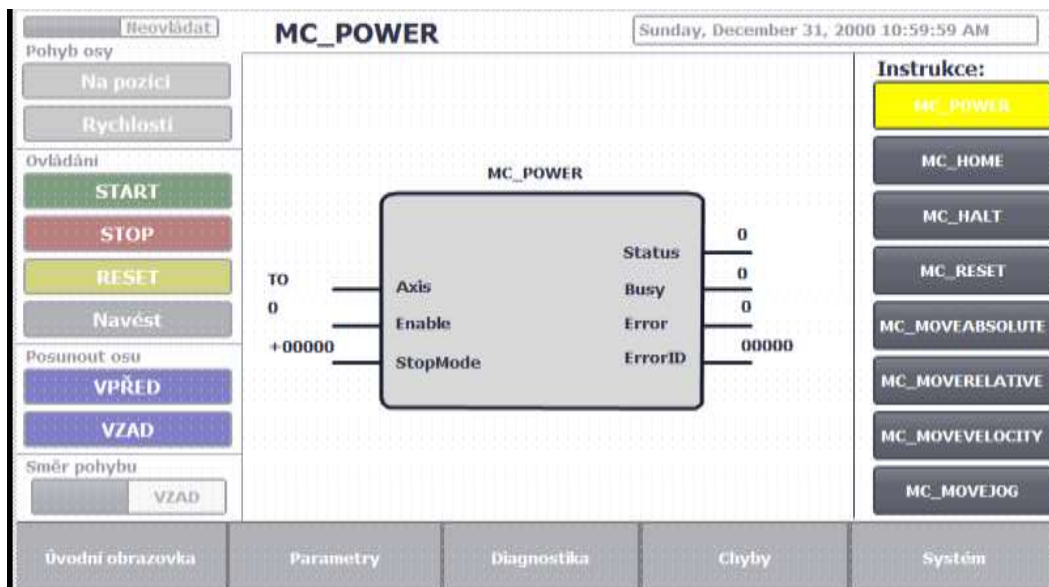
Start obrazovka je obrazovka, která se objeví vždy při spuštění HMI display. Tato obrazovka je navržena tak, aby působila jako uvítání do vizualizované části programu. Z tohoto důvodu bude tato obrazovka zobrazena dokud operátor nestiskne tlačítko Zapnout program. Po stisku tohoto tlačítka je povolena činnost technologického objektu díky propojení znaků HMI panelu a PLC. V tomto okně není využita žádná šablona. Start obrazovka je tvořena pouze statickými textovými poli obsahující text a dynamické tlačítko.

### 2.4.3 Parametrické a diagnostické obrazovky

Tyto obrazovky využívají šablonu Template\_1. V parametrické obrazovce lze zadávat aktuální dynamické parametry pohybu a zobrazují aktuální hodnoty rychlosti a polohy, dále zobrazují aktuální stav zařízení. Diagnostické obrazovky obsahují aktuální informace pro jednotlivé instrukce pro řízení pohybu. Pro každou instrukci existuje jedna diagnostická obrazovka. Pokud uživatel odpojí přepínač umístěný nahoře vpravo, může přímo měnit parametry na instrukcích.



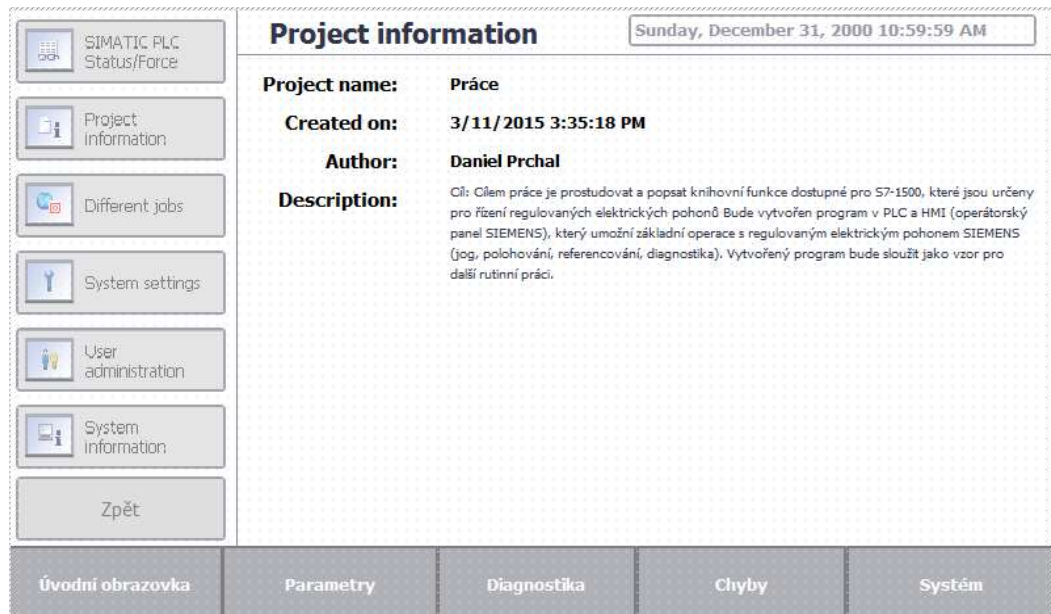
Obr.2.16 – Parametrická obrazovka



Obr. 2.17 – Diagnostická obrazovka funkce MC\_Power

## 2.4.4 Systémové obrazovky

Systémové obrazovky zde slouží pro snadnější správu PLC a celého zařízení. K vytvoření těchto obrazovek byly využity automaticky generované prvky, při vkládání panelu do hardwaru pomocí *Portal view*. Tyto obrazovky využívají šablonu *Template\_2*.



Obr. 2.18 – Systémová obrazovka Project information

## 2.4.5 Obrazovka chyb

Obrazovka chyb slouží pouze pro zobrazení všech vzniklých chyb. Tato obrazovka kromě šablony *Template\_1* obsahuje pouze automatický prvek *Alarm view*. V tomto prvku budou automaticky zobrazovány vzniklé chyby jako seznam chyb.



Obr. 2.19 – Obrazovka chyb

### 3 ZHODNOCENÍ

Program lze ovládat pomocí tlačítek nebo přímo zadáváním parametrů do jednotlivých instrukcí. Zadávat přímo parametry do instrukce lze na diagnostických obrazovkách k této činnosti určených. Pro každou řídicí instrukci je vytvořena jedna obrazovka. Ovládání programu pomocí tlačítek je limitováno nastavenými parametry v bloku manuálního ovládání. Oproti tomuto je zadávání parametrů přímo řízeno pouze zadanými parametry v daných instrukcích. Přepínání mezi jednotlivými ovládacími módy se provádí pomocí přepínače umístěného nahoře vpravo.

Frekvenční měnič je spouštěn automaticky při zapnutí programu na spouštěcí uvítací obrazovce. Poté již lze dále ovládat program pomocí tlačítek umístěných napravo. Pro výběr mezi polohováním a rychlostním pohybem osy se využívají tlačítka na polohu a rychlost. Dle zvoleného pracovního módu následně program bude vykonávat další činnosti, které se spouští tlačítkem start a zastavují tlačítkem stop. Tlačítko reset je zde umístěno pro resetování chyb v programu. Při nastaveném polohování je nezbytné nastavit výchozí pozici, protože by program hlásil chybu při polohování. Při rychlostním pohybu osy není potřeby výchozí pozici nastavovat. K nastavení směru pohybu slouží pod tlačítka umístěny přepínač mezi směry vpřed a vzad. Kdy směr vpřed hýbe osou v kladném směru a směr vzad v záporném směru. Poslední tlačítka vpřed a vzad zde slouží k pohybu osy, dokud jsou aktivována.

Dynamické vlastnosti pohybu se nastavují v editovatelných oknech na tyto vlastnosti určené. Zapisují se na parametrické obrazovky do nastavených parametrů. Aktuální informace o rychlosti a pozici osy je zobrazena napravo od nastavených parametrů.

Pro pohyb mezi obrazovkami je zde vytvořena spodní lišta, ze které lze přejít na kteroukoliv ze základních obrazovek. Základní obrazovky jsou: úvodní obrazovka, parametrická, diagnostická, systémové obrazovky a obrazovka chyb. Další pohyb mezi obrazovkami lze pomocí listy umístěné nalevo. Lišta slouží primárně pro přepínání mezi jednotlivými diagnostickými instrukcemi. Tedy diagnostické obrazovky jsou MC\_POWER, MC\_HALT, MC\_HOME, MC\_MOVEVELOCITY, MC\_MOVEABSOLUTE, MC\_MOVERELATIVE, MC\_RESET a MC\_MOVEJOG.

Systémové obrazovky mají místo tlačítek pro ovládání pohonu funkce pro přepínání mezi jednotlivými systémovými obrazovkami. Pomocí systémových obrazovek lze měnit režim práce PLC, diagnostikovat chyby, zobrazit informace o panelu a projektu nebo měnit nastavení systému.

Praktické využití tohoto programu je primárně pro ilustrační účely využití jednotlivých instrukcí pro řízení pohybu. Části programu lze jistě využít v jiných aplikacích, protože program je psán univerzálně. Při využití programu v jiné aplikaci by byla hlavně nutná změna v bloku manuální řízení. Tato funkce provádí řízení všech pohybových instrukcí. K složitějšímu využití programu je nezbytné přidat správu uživatelů. Dále by se dal využít recepturní systém, který by měl vkládat hodnoty do parametrů.

Obecně lze říci, že pomocí programu TIA Portal a hardwarové platformy S7-1500, lze realizovat celou řadu aplikací pro řízení pohybu osy. Proto s těmito prostředky lze realizovat velmi rozsáhlé projekty, které mohou představovat například výrobní linky v továrnách, jednoduché aplikační přístroje nebo aplikace náročné na bezpečnost a přesnost.



## 4 ZÁVĚR

Cílem práce bylo vytvořit program pro vysvětlení principu řízení pohybu regulovaného elektrického pohonu. Dle popsaných vlastností a funkcí programu TIA Portal byl vytvořen program pro popsanou hardwarovou platformu S7-1500.

Celý program je řízen popsanými pohybovými instrukcemi z knihovny OpenPLC. Lze konstatovat, že tyto funkce ve spojení s novou generací programovatelných automatů řady S7-1500, jsou schopny zcela pokrýt všechny aplikace, kde se využívá řízení pohybu. Výkonově nová řada automatů, dle výrobce, by měla být výkonnější, než starší verze daného zařízení. Toto tvrzení nelze potvrdit z důvodu, že aplikace je příliš malá na porovnání aplikačního výkonu. Obecně považuji rodinu programovatelných automatů S7-1500 za velmi kvalitní, výkonný produkt pro všechny typy průmyslové automatizace.

Programovací komfort v TIA Portal je zcela určitě zvýšen oproti starším programům, jako je STEP 5. K zásadnímu usnadnění přispělo i to, že není potřeba používat několik zvláštních programů, ale vše je vloženo v jednom univerzálním programu, se kterým lze naprogramovat frekvenční měnič, HMI panel i programovatelný automat. Zásadním problémem programu TIA Portal V13 je velká škála přetrvávajících programových chyb. Z důvodu těchto chyb, program občas chybně pracuje. Ovšem oproti starším verzím TIA Portal, došlo ke značnému zlepšení. Ve verzi programu TIA Portal V10 až V11 docházelo k občasným chybám v programu, které vedli až k selhání celého programu. Z tohoto plynoucí ztráta dat je samozřejmě nežádoucí a tedy práce s těmito verzemi programu byla velmi náročná.

Vytvořený program slouží dobře jako ukázka využití daného hardwaru a softwaru firmy SIEMENS. Cíle dané práce byly dle mého názoru plně splněny a vytvořený software může sloužit jako vzor pro další činnosti, kdy aplikace pro regulaci pohonu mohou využívat vytvořené komponenty v programu. Dále velmi oceňuji možnost přepínání mezi automatickým módem, kdy je program řízen dle programu nebo manuálním módem. Při manuálním módu lze jednotlivě ovlivňovat jednotlivé parametry instrukcí pro řízení pohybu. Z tohoto důvodu je tento mód velmi vhodný pro pochopení dané problematiky.

Další prohloubení problematiky by mělo být zaměřeno na hlubší funkce využití programovatelného automatu S7-1500. Toto prohloubení by mohlo obsahovat například vytvoření webového serveru na programovatelném automatu, využití recepturního systému HMI panelu a PID regulátoru na S7-1500.

## LITERATURA

- SIEMENS AG. 2014a. *STEP 7 Professional V13.0: System Manual* [online]. [cit. 2015-04-17]. Dostupné z: <https://support.industry.siemens.com/cs/document/89515142?lc=en-BR>
- SIEMENS AG. 2014b. *SIMATIC: S7-1500/ET200MP* [online]. [cit. 2015-04-17]. Dostupné z: <https://support.industry.siemens.com/cs/document/89515142?lc=en-BR>
- SIEMENS, AG. 2014c. *SIMATIC S7-1500/ET 200MP* [online]. [cit. 2015-04-17]. Dostupné z: <https://support.industry.siemens.com/cs/document/86140384?lc=en-AR>
- SIEMENS. 2015., *Totally Integrated Automation Portal* [online]. [cit. 2015-04-19]. Dostupné z: <http://stest1.etnetera.cz/ad/current/?ctxnh=2416f2e791&ctxp=home>
- TKOTZ, Klaus. 2006. *Příručka pro elektrotechnika. 2.*, dopl. vyd. Praha: Europa-Sobotáles, 623 s. ISBN 80-86706-13-3.
- ZEZULKA, František et al. 2003. *Programovatelné automaty*. Skriptum VUT FEKT UAMT, Brno 2003