



**České vysoké učení technické v Praze**

**Fakulta elektrotechnická**

**Katedra elektrických pohonů a trakce**

**PLC řízení pohonu jedné osy**

**PLC Control of Single Axis Drive**

Diplomová práce

Studijní program: Elektrotechnika, energetika a management

Studijní obor: Elektrické stroje, přístroje a pohony

Vedoucí práce: Ing. Jan Bauer, Ph.D.

**Bc. Tomáš Hošek**

**Praha 2019**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hošek** Jméno: **Tomáš** Osobní číslo: **420172**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra elektrických pohonů a trakce**  
Studijní program: **Elektrotechnika, energetika a management**  
Studijní obor: **Elektrické stroje, přístroje a pohony**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**PLC řízení pohonu jedné osy**

Název diplomové práce anglicky:

**PLC Control of Single Axis Drive**

Pokyny pro vypracování:

- 1) Seznamte se s kitem SIMATIC S7-1200 a měničem V90
- 2) Seznamte se s konfiguračním prostředím TIA Portal
- 3) Popište konfiguraci PLC a V90 v prostředí TIA Portal
- 4) Vytvořte demo aplikaci použití PLC s měničem a pohonem

Seznam doporučené literatury:

- [1] dokumentace k TIA Portal - dostupné online
- [2] dokumentace k PLC SIMATIC S7-1200 - dostupné online
- [3] dokumentace k V90 - dostupné online

Jméno a pracoviště vedoucí(ho) diplomové práce:


**Ing. Jan Bauer, Ph.D., katedra elektrických pohonů a trakce FEL**


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


Datum zadání diplomové práce: **08.02.2019**

Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **20.09.2020**

  
Ing. Jan Bauer, Ph.D.  
podpis vedoucí(ho) práce


  
podpis vedoucí(ho) ústavu/katedry

  
prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

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

16.5.2019  
Datum převzetí zadání

  
Podpis studenta

## Prohlášení

„Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.“

V Praze dne .....

.....  
Podpis autora



## Poděkování

Děkuji především vedoucímu mé diplomové práce panu Ing. Janu Bauerovi, Ph.D. za námět diplomové práce, za obstarání potřebných komponentů, přidělení pracoviště s potřebným zázemím, rady a cenné připomínky, též za ochotnou pomoc při hledání nedostatků a jejich odstraňování.

Děkuji i všem ostatním, kteří mě při tvorbě závěrečné práce podporovali.

# Abstrakt

Tato diplomová práce je zaměřena na konkrétní použití PLC modulu pro řízení synchronního třífázového motoru s permanentními magnety napájeného z výkonového měniče s možností kontrolovat zautomatizovaný proces z dotykového panelu, kde lze kromě monitorování děje nastavovat vybrané parametry. Vytvořená sestava obsahující PLC modul, výkonový měnič, synchronní motor, ovládací panel a podpůrné moduly pro napájení a vzájemnou komunikaci mezi komponenty a počítačem umožní realizovat demonstrativní úlohu použitelnou v průmyslu. Práce se zaměří na tvorbu potřebného kódu ve vývojovém prostředí určeném pro použité komponenty firmy SIEMENS, propojení uvedených modulů průmyslovou komunikační sběrnici s protokolem Profinet, vytvoření jednoduchého procesu a popisu jednotlivých částí.

## Klíčová slova

PLC modul, dotykový panel HMI, výkonový měnič, synchronní motor, vývojové prostředí TIA Portal Step 7

# Abstract

This diploma thesis focuses on the application of PLC module for controlling the synchronous motor with permanent magnets powered by the driver with a possibility of checking the automated process from a touch panel, where it is possible to monitor the process and set some parameters. The complex created from the PLC module, driver, synchronous motor, touch panel and supporting components for powering and communicating between the components and the computer will enable the demonstrative process to be realized in industry. The project will focus on the creation of the necessary program code in the development environment intended for selected components of the SIEMENS company, interconnection of the mentioned modules by the industrial communication bus with the protocol Profinet, creation of simple process and description of individual parts.

## Key words

PLC module, HMI touch panel, driver, servo motor, TIA Portal Step 7 development environment

# Obsah

1	Úvod .....	1
2	Historie a předchůdci PLC.....	3
2.1	Reléové systémy .....	3
2.2	1. generace počítačů .....	3
2.3	2. generace počítačů .....	3
2.4	3. generace počítačů .....	4
2.5	4. generace počítačů .....	4
2.6	Vývoj PLC .....	4
3	Programovatelný logický automat .....	5
3.1	Reléové schéma.....	6
3.2	Funkční bloky.....	6
3.3	Strukturovaný text.....	7
4	Požadavky na PLC .....	8
4.1	Provozní prostředí .....	8
4.2	Zajištěnost provozu .....	8
5	Komunikace .....	9
5.1	Paralelní sběrnice .....	9
5.2	Sériová sběrnice .....	9
5.2.1	Principy přenosu po sériové lince.....	9
5.3	Komunikace v průmyslu .....	11
5.3.1	Profibus .....	11
5.3.2	Profinet.....	11
5.3.3	Ethernet.....	12
6	Způsoby vizualizace dat a procesů .....	14
7	Popis jednotlivých částí sestavy .....	16
7.1	Napájecí zdroj SIMATIC PM1207.....	16
7.2	Přepínací modul SIMATIC NET CSM 1277 .....	17
7.3	PLC modul SIMATIC SZ-1200 CPU 1215C.....	17
7.3.1	Typ 1215C DC/DC/DC .....	18
7.4	Výkonový měnič SINAMICS V90 .....	20
7.5	Motor SIMOTICS S-1FL6 .....	22
7.5.1	Enkodér .....	25
7.6	KTP700 Basic, 6AV2 123-2GB03-0AX0, V-K5AS8061 .....	26
8	Vývojové prostředí .....	28



8.1	V-ASSISTANT.....	28
8.2	TIA PORTAL.....	28
8.2.1	Popis vybraných bloků .....	31
8.2.2	Ovládání přes HMI.....	34
9	Popis výsledného projektu .....	36
10	Závěr.....	46
	Seznam obrázků .....	47
	Seznam tabulek.....	48
	Seznam použitých zkratk.....	49
	Seznam použité literatury .....	50
	Příloha .....	- 1 -



# 1 Úvod

Automatizace výrobních procesů má v posledních desetiletích narůstající trend. Snahou mnoha podniků je zajistit spolehlivější a rychlejší způsob výroby v místech, kde se často výrobní operace opakují, nebo upravují při přechodu na podobnou zakázku. Některé výrobní procesy využívají jednoduché, spíše poloautomatické systémy, které pomáhají obsluze zařízení snížit počet vykonávaných úkonů, a tudíž i počet potenciálních chyb způsobených nepozorností obsluhujícího pracovníka. Příkladem může být pásový dopravník, který dokáže po uvedení do chodu obsluhou zařízení bez dalšího zásahu převést náklad na přesně určené místo a zastavit, přičemž rozjetí a zastavení dopravního pásu může být pozvolné s regulací otáček pohonu, bez škubnutí při rozjezdu a rychlého zastavení. Vyšší úrovní autonomnosti zařízení by bylo vyhodnocení přítomnosti nákladu a automatické uvedení do chodu bez zásahu člověka. Výrobní proces však lze zautomatizovat zcela, kdy na systém pouze dohlíží kvalifikovaná, vyškolená osoba a zasáhne do systému pouze v případě poruchy. Snahou vývojářů automatizace je co nejvíce snížit počet zásahů člověka do systému, nebo zásahy do výrobního procesu zcela vyloučit. Takový systém musí splňovat požadavky na bezpečný chod procesu i v případě neočekávaných stavů. Smyslem automatizace je vytvořit proces, který bude zcela nezávislý na člověku. Zamezí se tak chyb lidského faktoru. Pro dlouhodobý správný chod se však zařízení neobejde bez zásahu kvalifikovaných osob, které mají na starost pravidelnou údržbu a kontrolu všech aktivních i pasivních prvků zajišťujících automatický provoz. Nutná je též výměna opotřebovaných částí, kterou lze provádět preventivně jednak na základě údajů ze strany výrobce jednotlivých komponent a jednak ze statistických údajů získaných z dosavadního provozu.

Automatizace se však netýká pouze výrobních procesů, používá se též v dopravě nebo pro zajišťování služeb. Příkladem automatického systému pro přepravu osob, či zboží je výtah, který byl zároveň jedním z prvních automatických procesů již v době reléových systémů – obsluha pouze zmáčkla tlačítko pro příslušné patro a výtah se bez dalších úkonů rozjel a dosáhl příslušného místa. Zlepšování a zajišťování bezproblémového chodu automatických celků vedlo až k vytvoření dalšího autonomního dopravního prostředku – samořiditelného metra. S provozem metra bez řidiče, kdy na bezproblémový provoz souprav pouze dohlíží dispečer se setkáme již v mnoha zemích světa. V současnosti jsou též známá automaticky řízená silniční vozidla. V Austrálii se lze od roku 2017 setkat i s autonomně řízeným vlakem těžařské společnosti *Rio Tinto Iron Ore*. [1]

Pro zajištění bezproblémového, a především bezpečného chodu autonomního zařízení se používá řada aktivních i pasivních prvků, především senzorů a spínačů. Některé senzorické prvky reagují pouze na překážku, která neumožňuje dokončení prováděné operace a tu ihned ukončí, jiné prvky slouží například k detekci objektu (tlakové čidlo, nebo světelná závora u pásového dopravníku), ke snímání teploty, tlaku kapaliny, nebo k určení stavu obsahu zásobníku kapaliny. Mechanické koncové spínače slouží k zastavení činnosti, došlo-li k otevření ochranných krytů, nebo pokud mechanismus dosáhl krajní polohy.

Naprostá většina automatických procesů se neobejde bez pohonů, výjimkou je snad pouze řízení osvětlovacích nebo akustických systémů. Pohony mohou být různých druhů z hlediska druhu spotřebované energie. Setkáme se tak s řadou pneumatických pohonů, jejichž pohonným médiem je stlačený plyn (nejčastěji vzduch), nebo kapalina u pohonů hydraulických.

Většina pohonů využívá k uskutečnění pohybu elektrickou energii měnící se na mechanickou pomocí elektrických motorů. V průmyslu a dopravě se používají motory různé konstrukce, či principu. Vždy záleží na dané aplikaci, potřebném výkonu a pracovním rozsahu.

Pohony mohou vykonávat rotační, nebo lineární pohyb. Tam, kde je to třeba, musí být pohon regulován v závislosti na rychlosti pohybujících se částí a velmi často musí být možnost určit okamžitou polohu předmětu, nebo pohyblivé části mechanismu. Proto se v takových případech setkáme hlavně se synchronními motory s permanentními magnety nebo s krokovými motory. Z důvodu nutnosti dlouhodobého zajištění bezproblémového chodu se do takovýchto aplikací nehodí stejnosměrné motory s komutátorem, ani asynchronní motory nebo synchronní motory s budícím vinutím. Vyloučením motorů s rotačními prvky přenášejícími elektrický výkon se zabrání jiskření a rychlejšímu opotřebení součástí. Asynchronní motor nelze použít z důvodu nepřesného určení polohy mechanismu. Rozhodujícím faktorem pro výběr mezi střídavým synchronním motorem a bezkartáčovým stejnosměrným motorem může být výsledná cena pohonu obsahující daný motor, výkonový měnič a řídicí jednotku, případně výkon pohonu, nebo přesnost určení polohy, či dosažitelné otáčky. Na trhu je k dispozici mnoho typů motorů vybavených již zabudovaným senzorem pro měření otáček a polohy rotoru. K určení polohy rotoru lze užít senzor na principu optické odrazivosti povrchu, nebo senzor obsahující Hallovu sondu<sup>1</sup> využívající k určení polohy magnetické pole.

---

<sup>1</sup> Hallova sonda je polovodičový senzor řízený magnetickým polem

## 2 Historie a předchůdci PLC

Snaha o automatizaci různých procesů ve výrobě je známá už od začátku 20. století, kdy bylo hlavním požadavkem co nejvíce zjednodušit soubor pracovních operací. K alespoň částečné automatizaci sloužily zprvu jednodušší systémy, které se s postupem času zdokonalovaly až do dnes běžně užívaných systémů, na jejichž vývoj se zaměřím v této kapitole.

### 2.1 Reléové systémy

Po vynalezení relé v 19. století a jeho velkém využití na začátku 20. století se začala uplatňovat reléová logika, která umožňovala řídit technologické procesy. V jednodušších systémech postačovala k řízení soustava relé, bez potřeby zásahu nadřazeného systému – obsluhy. Složitější procesy vyžadovaly určité modifikace řízení během jednoho cyklu opírající se o statistické údaje, nebo vyžadovaly zpětnou vazbu. Proto byly podle potřeby pozměňovány určité parametry systému z místnosti řízení procesu. Reléové systémy se podle potřeby doplňovaly dalšími elektromechanickými prvky. Uplatnění našly časovače a vačkové spínání<sup>2</sup>. Stále však nebylo možné zautomatizovat veškeré procesy. Taktéž úprava stávajících systémů na pozměněný proces byla velmi složitá z důvodu nutnosti elektrického přepojení prvků. [2]

### 2.2 1. generace počítačů

Vynález elektronky umožnil po druhé světové válce vznik počítačů první generace nazývaných též elektronkové počítače podle nezastupitelných součástek. Elektronky bylo možné programovat, nebo nastavit podle algoritmu, který byl pevně dán. Známé byly například elektronkové počítače *Ural* pocházející ze sovětského svazu. [2], [3]

### 2.3 2. generace počítačů

U těchto počítačů je již elektronka nahrazena novým elektronickým prvkem – tranzistorem. Tranzistory umožnily snížit velikost výpočetního zařízení, redukovat spotřebu elektrické energie a zároveň zvýšit jeho spolehlivost. První tranzistorový počítač vznikl roku 1953 na univerzitě v Manchesteru. [2], [4]

---

<sup>2</sup> vačkové spínání: mechanismus tvořící postupné spínání obvodů pomocí tvaru a výřezů vaček, vačka je kroužek s proměnlivým poloměrem po obvodu umístěný na hřídeli [5]

## 2.4 3. generace počítačů

V období let 1965-1980 se již uplatňovaly počítače obsahující integrované obvody<sup>3</sup>. Tyto počítače se již ve větším měřítku objevovaly v kosmonautice a v aplikacích pro armádní účely (řízení balistických mezikontinentálních střel). Ukládání dat již neprobíhalo na magnetické bubny, či děrné štítky, ale na diskety. Prvním osobním počítačem se v roce 1975 stal *Altair 8800*, jenž měl paměť 1 kB. K programování docházelo pomocí přepínačů, nebyl totiž vybaven klávesnicí. Později byl vytvořen první osobní počítač řady Apple. [2], [6]

## 2.5 4. generace počítačů

Dosazením mikroprocesorů došlo ke zmenšení rozměrů a zvětšení výkonnosti oproti předchozí generaci. Miniaturizace systému umožnila jeho rozšíření i do domácností, již se nejednalo o pouhý výpočetní stroj užívaný v průmyslu, armádě nebo kosmonautice. Mikroprocesor je miniaturní procesor, pro který se užívá zkratka CPU z anglického *Central Processing Unit*. Úkolem procesoru je vykonávání programu na základě čtení informací (strojových instrukcí) z paměti počítače. Strojový kód je vlastním jazykem každého procesoru. V průmyslu se lze hojně setkat s mikrokontroléry – jednočipovými počítači. Používají se jako vestavné systémy pro jednoúčelové aplikace. V současnosti se s nimi lze setkat téměř všude (tiskárny, fotoaparáty aj.). [2]

## 2.6 Vývoj PLC

První skutečný PLC automat byl vyvinut roku 1968 ve firmě *Bedford Associates*, která byla později přejmenovaná na *Modicon*, Richardem E. Moerlym. Automat byl v roce 1969 označen názvem *PLC "084"*. První programovatelné PLC představila firma *Allen-Bradley* roku 1970 pod názvem *PLC PDQ II*. Tato společnost následně představila řadu dalších automatů. Roku 1973 představila firma *Siemens* model *PLC S3*, který již využíval binární logiku. PLC si nechala patentovat firma *Allen-Bradley* roku 1974, kdy realizovala první paralelní zpracování a vizualizaci na obrazovce. Do přelomu tisíciletí představilo svůj výrobek PLC několik dalších firem, například *GE Fanuc Automation*, *AEG Schneider Electric* (pův. *Modicon*), *Omron*. [2]

---

<sup>3</sup> integrované obvody jsou obvody tvořené více komponenty umístěnými na jednom čipu (společné základové destičce)

### 3 Programovatelný logický automat

PLC lze popsat jako počítač upravený pro potřeby řízení do průmyslového prostředí. Integrovaná mikroprocesorová jednotka má vlastní operační systém sloužící k řešení automatických úkonů. Řešení úkonů probíhá v reálném čase a klade se požadavek na dosažení minimální doby odezvy. Automat musí být vybaven vstupy a výstupy, aby mohl reagovat na stavy řízeného procesu, popř. okolí a pomocí výstupů mohl řídit proces. Vstupní periferie mohou být různých typů. Přiváděný signál na vstupy je spojitý v případě, že je na vstup připojen analogový senzor předávající do automatu informaci o teplotě, tlaku, nebo jiné veličině vyjádřitelné ve spojitém intervalu měření. Nespojitý signál je přiveden na vstupy automatu především z koncových spínačů, bezpečnostních tlačítek a podobných prvků generujících dva stavy: zapnuto/vypnuto. Tento typ vstupů bývá zastoupen z velké části (zdroj [7] udává 80 % - 100 %). [7], [8]

Výstupy mohou nabývat stavů zapnuto/vypnuto například stykač motoru, nebo jsou výstupní periferie analogové a hodnota výstupní veličiny nabývá hodnot z konkrétního intervalu (například řízení otáček pohonu). [8]

PLC automat může obsahovat i jiné periferie sloužící pro komunikaci s obsluhou, pro vzdálený přístup k automatu nebo pro odesílání dat do externích zobrazovacích jednotek. Vnitřní vyhodnocovací obvody nazývané řídicí logika se starají o správnou reakci výstupů na vstupní parametry řízeného objektu. Způsob reagování řídicí logiky na změny vstupních stavů definuje programátor vytvořením programového algoritmu, který následně uloží do paměti automatu. Důležité je cyklické opakování úkonů PLC tak, aby se reagovalo na vstupy v určitých časech, protože stavy vstupů se s časem mění. Interval mezi dvěma cykly specifikuje rychlost odezvy systému, citlivost a dosažitelnou úroveň řízení. Čím kratší je interval, tím přesnější je odezva systému, ovšem společně s tím rostou i požadavky na rychlost vyhodnocení vstupních stavů a vytvoření nových výstupních stavů, tedy na rychlost mikroprocesoru. Omezením může být i různá délka signálových drah od senzorů. [7], [8]

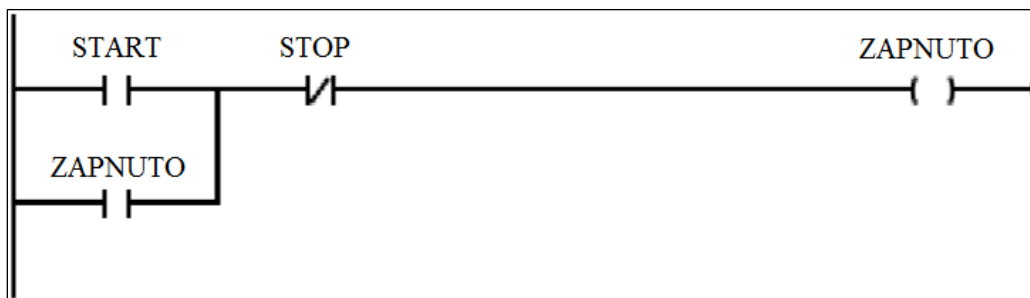
Nastavení PLC automatu, vytvoření souboru instrukcí probíhá ve vývojovém prostředí, které využívá jednu z možných metod. Mezi nejznámější programovací metody patří: [9]

- **Reléové schéma**
- Sekvenční programování
- Mnemokódy
- **Funkční bloky**
- **Strukturovaný text**

Výběr nejvhodnější metody záleží na mnoha parametrech, ať už se jedná o složitost automatizovaného procesu, nebo softwarového vybavení pracoviště. Programování PLC používá oproti programování pomocí vyšších programovacích jazyků, např. jazyk C, zjednodušené formy zápisu v instrukční sadě. Programové jazyky pro PLC se tedy více podobají Assembleru, což je nižší programovací jazyk. Tučně zvýrazněné metody si nyní popíšeme. [7]

### 3.1 Reléové schéma

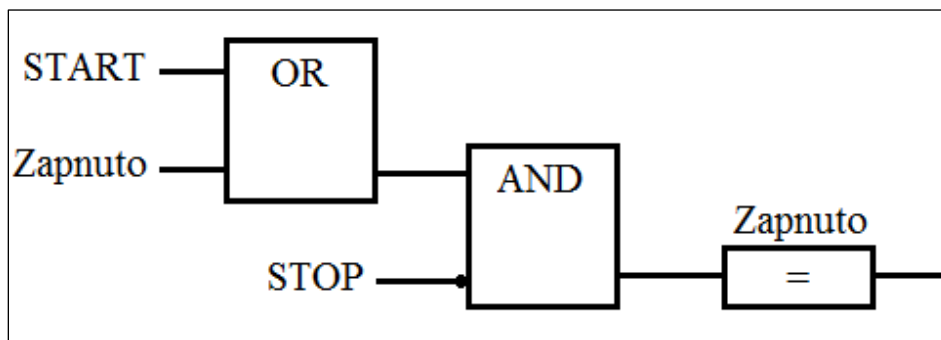
Reléové schéma (angl. *Ladder Diagrams, Ladder Logic, LAD*) je schéma, které se dříve používalo pro logické systémy řízení v průmyslových aplikacích, s použitím relé. Schéma tohoto typu vypadá jako žebřík (proto název ladder = žebřík). Dvě vertikální čáry reprezentují napájení, pro stejnosměrný systém tedy póly + a -. Vodorovné čáry, propojující oba póly obsahují ovládací prvky a spotřebiče. Ovládacími prvky mohou být mechanická tlačítka nebo kontakty relé systémů (výkonové i signální). Cívky stykačů nebo žárovky signalizací představují spotřebiče. Takto lze realizovat návrh ovládání jednodušších systémů, protože je schéma celkově přehledné a snadno si na něm lze představit daný stav systému. V případě složitějšího řízeného obvodu se však schéma stává nepřehledným, nebo těžší realizovatelným. Obr. 1 znázorňuje užití této metody pro popsání jednoduchého obvodu. [10]



Obr. 1: Reléové schéma. START představuje spínací kontakt tlačítka, STOP představuje rozpínací kontakt jiného tlačítka, symbol ( ) představuje cívku relé a spínací kontakt ZAPNUTO je kontaktem této cívky. Obvod zůstane sepnutý po krátkém stisknutí tlačítka START, díky paralelnímu kontaktu cívky a k rozepnutí dojde po krátkém stisknutí tlačítka STOP [11]

### 3.2 Funkční bloky

Metodu funkčních bloků lze užít v případech, kdy potřebujeme opakovaně programovat nejrůznější úkony, které lze vytvořit jednotlivě v rámci funkčního bloku a jednotlivé bloky pak stačí spojovat, či doplňovat specifickými úkony. Metoda je vhodná pro začátečníky a hodí se spíše pro jednodušší aplikace. Výhodou je také knihovna základních bloků v programovacím prostředí. Příklad popsany v podkapitole o reléových schématech přepsaný do podoby funkčních bloků přináší obr. 2. [9]



Obr. 2: Příklad užití funkčních bloků. Výstupem bloku OR je jedna, je-li alespoň jeden vstup v úrovni "1", výstup bloku AND je jedna, jsou-li oba vstupy "1" (negace vstupu STOP naznačena tečkou před vstupem do bloku). Třetí blok nemění stav vstupu, co je na vstupu, bude také na výstupu [11]



### 3.3 Strukturovaný text

Metoda strukturovaného textu je složitější, hodí se více pro pokročilé uživatele. Její výhodou je možnost vytvoření celého kódu na několika řádcích. Strukturu je nutné dodržovat v celém programu pro zvýšení přehlednosti kódu a eliminaci možných chyb. Doporučuje se opětovně užívat části kódů, které se v programu opakují a pouze je modifikovat pro konkrétní obdobné procesy. Díky tomu je celý kód přehlednější a lépe se nalézají případné nedostatky. Významným krokem je vytváření komentářů hned za daným řádkem instrukce, které zrychlí orientaci v programu při jeho pozdější editaci. Komentáře též poslouží jako návod uživatelům, které kód netvořili, ale potřebují s ním pracovat. Vhodné pojmenování proměnných a funkčních bloků opět přináší zlepšení přehlednosti kódu. Během tvoření samotného kódu je více než žádoucí občas kontrolovat správnou syntaxi a funkčnost programu. V mnoha vývojových prostředích slouží k ověření správné činnosti funkce simulace programu a jeho krokování. Příklad zápisu kódu pomocí strukturovaného textu naleznete v kapitole o vývojovém prostředí. [9]

## 4 Požadavky na PLC

Na PLC automaty jsou kladeny nároky odpovídající prostředí, ve kterém se nachází. PLC automat nesmí rušit okolní přístroje a sám musí být odolný proti rušení zvenčí, musí tedy splňovat požadavky z hlediska EMC<sup>4</sup>. Zajištěnost správné funkce automatu také závisí na dané aplikaci.

### 4.1 Provozní prostředí

Provozní prostředí zahrnuje vlivy okolní teploty, relativní vlhkosti vzduchu, prašnosti okolí, chemické agresivity a dalších vnějších aspektů ovlivňujících funkci zařízení. Každé zařízení musí být navrženo tak, aby fungovalo správně v předepsaných provozních podmínkách, které určuje výrobce. Provozní prostředí působí na zařízení degradačně po celou dobu užívání zařízení, a proto se klade požadavek na jeho co největší odolnost a dostatečnou životnost. PLC se často nacházejí v prostředích s velkým teplotním rozdílem, kdy vysoká teplota urychluje degradaci celého, nebo exponovaných částí automatu, nízká teplota zase způsobuje kondenzaci vodních par. Nebezpečným faktorem bývá také prašnost prostředí. Prach se může usazovat na zařízení a brání odvodu ztrátového výkonu a narůstající teplota opět degraduje zařízení. Rizikové je též působení nebezpečných tekutin nacházejících se v okolí zařízení. Zařízení neprosívají ani vibrace, vznikající v okolí PLC automatu, či v samotném procesu. Popsaným negativním účinkům prostředí, je nutno zabráňovat a předcházet.

### 4.2 Zajištěnost provozu

V provozu musí být PLC automat neustále napájen a nesmí docházet k výpadkům napájení. Některé operace nesmí být přerušeny, nebo předběžně ukončeny, proto se používá záložních zdrojů a filtrů. Automat nesmí být ovlivňován nejen výpadky napájecí sítě, ale ani nekvalitou elektrické energie (výkyvy napětí, frekvence, vyšší harmonické). Z hlediska správné funkce a bezpečnosti provozu se musí předcházet negativním vlivům atmosférických výbojů, zabráňovat vzniku elektrostatického výboje, vstupu bludných proudů do zařízení. K tomu slouží nejrůznější přístroje (přepětové ochrany, proudové chrániče) a způsoby elektrického zapojení (pospojování, uzemnění). V žádném případě nesmí dojít ke ztrátě programu a zařízení musí být schopno bezpečného najetí procesu i z nedefinovaných nebo nepředvídatelných stavů. Program musí být uložen v paměti automatu a uchováván i ve vypnutém stavu zařízení. Mnoho aplikací vyžaduje, aby si PLC zapamatovalo, v jakém stavu daný proces skončil například při nadřazeném zásahu a po uvedení do opětovného chodu pokračoval ve vykonávání instrukcí od místa, kde program skončil. [7]

---

<sup>4</sup> EMC: označení pro elektromagnetickou kompatibilitu (zařízení nesmí rušit okolní zařízení a samo nesmí být rušeno okolím)

## 5 Komunikace

Jednotlivá zařízení, která spolu pracují a vytváří celek musí mezi sebou navzájem komunikovat. V průmyslových aplikacích je komunikace nejčastěji zajištěna po sběrnici, která může propojovat více zařízení najednou. Aby komunikace probíhala správně, musí v každém okamžiku vysílat data na sběrnici pouze jedno zařízení. V případě, že dojde k odeslání dat z více zařízení najednou, nelze je pak na straně příjemce zpracovávat. Celek tedy musí obsahovat zařízení, které bude řídit přenos. Toto zařízení je označeno *Master* (pán) a ostatní zařízení označujeme *Slave* (otrok). Master tedy nastavuje komunikaci a vybírá podřízená zařízení ke komunikaci v době, kdy mají vyslat informaci. Podle toho, zda potřebujeme přenášet informace po jednotlivých bitech anebo po více bitech najednou, rozlišujeme přenosové linky na sériové a paralelní. [12]

### 5.1 Paralelní sběrnice

Sběrnice umožňuje přenášení dat současně – paralelně. Každý přenášený bit potřebuje vlastní vodič. Šířka sběrnice je počet bitů, tedy i vodičů a bývá to mocnina čísla 2 (např. 8, 16, ale i více). Velmi často je vodič paralelní sběrnice vytvořen přímo na desce s plošnými spoji a uplatnění nalézá především uvnitř počítačů k propojení jednotlivých komponent. [12]

### 5.2 Sériová sběrnice

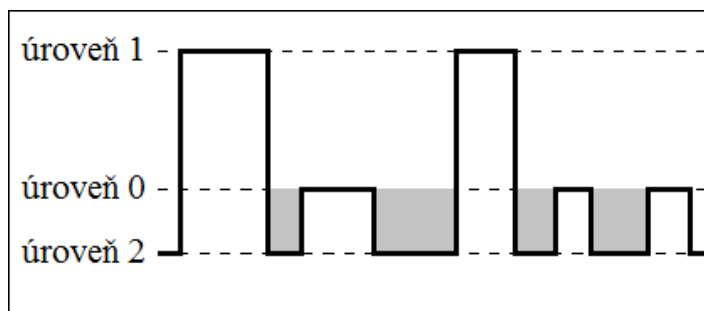
Sběrnice o šířce jeden bit se používá v různých průmyslových aplikacích, k propojení externích částí počítače a v dalších aplikacích, kde není třeba vysílat a přijímat více bitů najednou. Mechanismus přenášení dat po bitech je stejný pro paralelní i sériový přenos, a protože ve své práci využívám sériové komunikace, zaměřím se více na způsoby přenosu po jednom bitu.

#### 5.2.1 Principy přenosu po sériové lince

Přenášená informace se musí nejprve rozdělit na jednotlivé bity, které nabývají dvou diskrétních stavů: logická 0 a logická 1. Při přenášení logické 1 je k vodiči připojeno stejnosměrné napětí o specifikované velikosti a při přenášení informace logická 0 je napětí nižší, nebo nulové (0 V). Datový bit však může být tvořen také přechodem z jednoho stavu do stavu druhého a tento přechod nazýváme hranou. Pro správné vyhodnocení informace na straně příjemce musíme zajistit, aby doba, po kterou vysílač vysílá, byla shodná s dobou příjmu přijímače, jedná se o tzv. bitový interval. Dodržet synchronizaci není vždy jednoduché, neboť každé zařízení používá své hodiny (generátor hodinového signálu) a se zvyšující se přenosovou rychlostí se zvyšuje i riziko předcházení, či opoždování čtení dat. Ke koordinaci zápisu a čtení dat existují dva odlišné způsoby (viz dále). [12], [13]

### 5.2.1.1 Arytmický (asynchronní) přenos

Hlavním a rozhodujícím požadavkem je délka časového úseku, kdy dokáží hodiny vysílače a přijímače jít téměř synchronně (časový rozdíl na konci tohoto úseku nezpůsobí špatné čtení dat). Poté stačí, aby vysílač sdělil příjemci zprávy začátek a konec bitového intervalu. Z tohoto požadavku vyplývá výhoda vysílat různě dlouhé bitové intervaly. Přenosová rychlost asynchronního přenosu je tedy proměnná. Nevýhodou popisovaného přenosu je nutnost třetího stavu (viz obr. 3, úroveň 2) vysílaného signálu, kdy dva stavy představují přenášenou informaci a třetí stav slouží k oznámení začátku a konce bitového intervalu. Určitým odstraněním třístavového signálu je vysílání zpráv v daných stejně dlouhých intervalech – posloupnosti bitů, kterou předchází značka umožňující seřízení hodin příjemce s hodinami odesílatele. Délka posloupnosti bitů musí být opět jen ta dlouhá, aby příjemce správně vyhodnocoval přijímanou zprávu ještě před příchodem následující synchronizační značky. Posloupnost bitů ohraničená dvěma značkami se často nazývá znak a celý přenos je tedy znakově-orientovaný. Značku sloužící k nastavení hodin na straně příjemce můžeme označit jako *start bit* a značku za posloupností přenášených bitů *stop bit*. [13]



Obr. 3: Třístavový signál, šedě: oddělovače bitových intervalů

### 5.2.1.2 Synchronní přenos

Tento proces opět vhodným způsobem odstraňuje rozcházení obou hodinových signálů. Pro efektivní přenášení různě dlouhých posloupností bitů se používá způsobu, kdy je trvale zajišťována synchronizace hodin na straně příjemce. U přenosů na krátké vzdálenosti lze posílat samostatným vodičem chod hodin vysílače a na straně příjemce dle tohoto kanálu seřizovat vnitřní hodinový signál. U přenosů, kde není možnost použití dodatečného vodiče, se musí přenášet informace od hodin spolu s přenášenými daty. Tato varianta vyžaduje dvě změny signálu pro každý bitový interval, což má negativní dopad na šířku přenosového pásma, které se tímto sníží na polovinu. V případě přenášení dat změnou úrovně signálu, poslouží tato hrana i k seřízení hodin příjemce. Problém nastane v okamžiku, kdy se přenáší posloupnost pouze nul, nebo pouze jedniček, kdy v signálu není žádná hrana. Aby během tohoto časového úseku nedošlo k rozhození hodinových signálů, vkládá se do posloupnosti bit opačné úrovně (vytvoří se tak hrana), který zajistí včasné seřízení hodin a po tomto úkonu se zahodí, neovlivňuje tedy data. [13]

## 5.3 Komunikace v průmyslu

Se zesložitováním průmyslových aplikací s narůstajícími požadavky na rychlost, přesnost a objem dat se postupně vyvinuly metody přenosu dat vhodné pro různé stupně automatizace. Mezi obvykle užívané patří sítě *Modbus*, *Can bus*, *Profibus*, *Profinet*, *Ethernet*. Z důvodu rozsáhlosti informací o jednotlivých metodách se zaměřím pouze na některé z uvedených. S posledními dvěma sítěmi jsem se setkal přímo při své práci, ale zmíním se i o síti *Profibus*, která se dodnes využívá a ze které síť *Profinet* částečně vychází. [14]

### 5.3.1 Profibus

Zkratka vznikla z anglických slov *Process Field Bus*, jde o komunikační sběrnici, která byla vyvinuta v osmdesátých letech 20. stol. Vznikla z popudu mnoha velkých firem s určením do průmyslových odvětvích a sloužila jako sériová linka, kdy docházelo k výměně dat postupně po jednotlivých bitech. Z této koncepce následně vznikaly pozdější verze, například *Profibus-PA* určený k řízení spojitých procesů nebo *Profibus-DP* využívaný pro řízení strojů. Přenosová rychlost druhé jmenované verze silně závisela na délce propojení dvou komponent. Pro kratší vzdálenosti byla přenosová rychlost až 12 Mb/s. Pro vzdálenosti kolem 1 000 metrů se přenosová rychlost snížila o tři řády a pro delší vzdálenosti napomáhaly k přenosu dat tzv. opakovače. K jednomu systému lze připojit až 127 zařízení, z nichž každé má přidělenou adresu. Propojovací kabely jsou navrženy tak, aby šlo do systému zapojovat jednotlivá zařízení paralelně. U nejvzdálenějšího (posledního) zařízení se posuvným přepínačem zabudovaným do konektoru kabelu vřadil do sběrnice odpor indukující konec této sběrnice. Sériový konektor je typu *DB-9*, kabel je často rozlišitelný svou barvou – bývá fialový. [14], [15], [16]

### 5.3.2 Profinet

Tento průmyslový komunikační protokol je velmi odlišný od předešlého typu a na první pohled je odlišitelný jiným typem konektoru i kabelu, který bývá nejčastěji zelený s vysokým stupněm stínění potřebným při použití v drsných provozních podmínkách. *Profinet* je založen na průmyslovém *Ethernetu* (typ konektoru je stejný – *RJ-45*) s odstraněním jeho nevýhod. Komunikační rychlost je při délce kabelu do 100 m až 100 Mb/s, s dobou odezvy kratší než 1 ms. Díky tomu je oproti profibusu, kde je doba odezvy až 10 s, výhodnější v časově náročných aplikacích. Další výhodou je členění jednotlivých zařízení do systému. Propojení nemusí být pouze z jednoho zařízení do druhého, ale lze užít i rozvětveného zapojení s použitím standardního ethernetového rozbočovače. Oproti použití profibusu lze zapojit do jednoho systému více zařízení, která jsou snadno identifikovatelná třemi typy adres: *IP adresa*, *MAC adresa* a *název zařízení*. Všechna ethernetová zařízení musí mít IP i MAC adresu a profinet zařízení mají svůj název zařízení, která nastavuje sám uživatel. Důležité adresy v průmyslových aplikacích jsou IP adresa a název zařízení. Počet zařízení připojitelných do systému je omezen počtem možných IP adres (, které jsou pro tento účel vyčleněné). Systém profinet je jednodušší na údržbu a opravu. [15], [16]

### 5.3.3 Ethernet

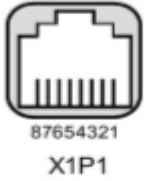
Jelikož profinet vychází z koncepce lokální sítě, je vhodné tuto síť přiblížit. Ethernet lze nazvat komunikačním standardem vyvinutým v 80. letech min. století pro lokální počítačové sítě a podobná zařízení. Tato lokální síť užívaná v budovách, popř. domácnostech nese označení místní síť a zkratku LAN (z angl. Local Area Network). Významem sítě je uchování a přenos dat mezi jednotlivými připojenými zařízeními. Realizace propojení zařízení má mnoho podob. Nejstarším způsobem je propojení zařízení koaxiálním kabelem, které bylo nahrazeno kroucenou dvoulinkou s měděnými vodiči. Propojení zařízení lze vytvořit i s pomocí optických kabelů, které využívají k přenosu informací paprsek monochromatického záření. Kabelová propojení (ať už elektricky vodivá nebo světlo vodivá) byla v roce 1983 standardizována a nesou označení *IEEE 802.3*. Ve zkratce označení je obsažen název institutu, který standardizaci provedl (IEEE – Institute of Electrical and Electronic Engineers). Existují i bezdrátová propojení známá pod označením *IEEE 802.11*. Standard Ethernet *IEEE 802.3* zahrnuje fyzickou vrstvu a vrstvu datového spojení. Obě vrstvy jsou obsaženy v modelu propojení otevřených systémů (OSI). Fyzickou vrstvu představují zařízení a jejich vzájemná propojení: koaxiálními kabely, kroucenou dvojlinkou nebo optickými vlákny. Kroucená dvojlinka se v praxi vyskytuje nejčastěji, především z důvodu snadnějšího připojení k zařízením a nižší výrobní cenou oproti optickým kabelům. I z hlediska rychlosti přenosu dat, zvláště u silně stíněných kabelů (s označením *CAT 6a* a *CAT 7a* a přenosovou rychlostí až 10 Gb/s) jsou kroucené dvojlinky perspektivní. Stínění kabelů a kvalita provedení stínění mají velký vliv na přenosovou rychlost, protože přenos dat kroucenými dvojlinkami je náchylný k šumu přicházejícího z okolního prostředí i samotného zařízení. Optická vlákna tuto nevýhodu nemají, ovšem hlavními nevýhodami je cena a nutnost užití speciálních konektorů. Umožňují však přenos dat vyššími rychlostmi na delší vzdálenosti než při použití kroucených vodičů. Při použití jednoho páru kroucené dvojlinky vznikne tzv. *Half-Duplex*, který umožňuje přenášet data pouze v jednom směru. Pro přenášení dat oběma směry současně je nutno použít dvou párů těchto vodičů (označovaných též jako *twisted pair*), čímž vznikne *Full-Duplex*. [14], [17]

Druhá vrstva OSI modelu zahrnuje datové spojení, které lze rozdělit do dvou podskupin. LLC (z anglického Logical Link Control) slouží k vytvoření signálové trasy pro přenos dat mezi konkrétními zařízeními. Druhá podskupina označená zkratkou MAC (*Media Access Control*) spravuje adresy, které jsou přiděleny ke kartám síťového rozhraní (NIC) jednotlivých zařízení a umožňují tato zařízení identifikovat. Standardní ethernet používá k přenosu dat algoritmus nazvaný CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) sloužící ke zvýšení úspěšnosti přenosu dat snížením datových kolizí (vyslání dat z obou konců přenosu současně). V současnosti se ethernet hojně využívá z důvodu dostatečné přenosové rychlosti, snadnosti instalace, příznivé ceny, rozsahu užití a také z důvodu podpory mnoha síťových protokolů. [17], [18]

Poznámka k funkci autocrossover: MDI/MDI-X (MDI – *Medium Dependent Interface* neboli středně závislé rozhraní, X – *autocrossover*) umožňuje použít přímé kabely v celém rozsahu. Některá zařízení totiž používají zásuvky RJ-45 se standardní ethernetovou konfigurací pinů označenou MDI dle obr. 4 v tab. 1, jiná zařízení užívají zásuvky MDI-X, kde je odlišné uspořádání pinů (viz obr. 5 vpravo). Autonegace je funkce zařízení detekující typ konfigurace a v případě, že je konfigurace pinů opačná, než jaká je požadovaná (místo přímého kabelu je použit křížený, nebo naopak), dojde k automatickému přepnutí pinů uvnitř zařízení (výměna přenosových a přijímacích linek). [11], [19]

Tab. 1: Uspořádání a zapojení pinů zásuvek RJ-45 dvou typů

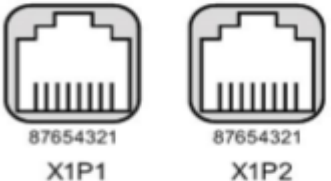
standardní řazení pinů MDI		
číslo pinu	název signálu	popis
1	TD+	vysílání dat
2	TD-	vysílání dat
3	RD+	příjem dat
4	GND	uzemnění
5	GND	uzemnění
6	RD-	příjem dat
7	GND	uzemnění
8	GND	uzemnění



Obr. 4: zásuvka typu RJ-45 MDI [11]

standardní řazení pinů MDI-X		
číslo pinu	název signálu	popis
1	RD+	příjem dat
2	RD-	příjem dat
3	TD+	vysílání dat
4	GND	uzemnění
5	GND	uzemnění
6	TD-	vysílání dat
7	GND	uzemnění
8	GND	uzemnění



Obr. 5: zásuvka typu RJ-45 MDI-X [11]

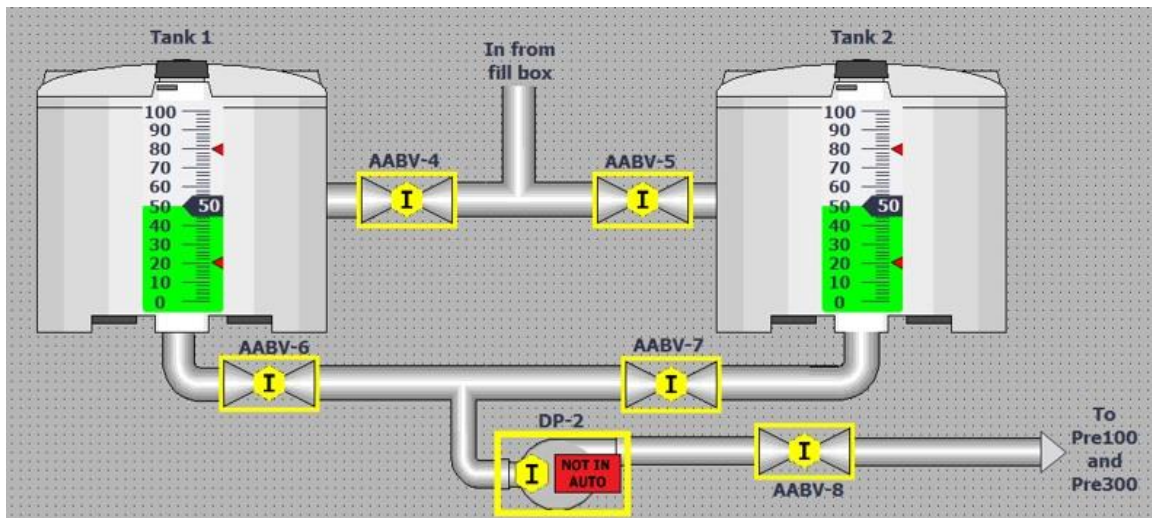
## 6 Způsoby vizualizace dat a procesů

Důležitou částí tvorby procesů je vedle způsobu jejich řízení také způsob zobrazování dat a probíhajících dějů. Pro tento účel byl vytvořen systém SCADA (*Supervisory Control And Data Acquisition*), který umožňuje dohlížet na konkrétní proces, řídit jej a také sbírat data z procesu. Tento systém nezastupuje samotný způsob řízení procesu, pouze jej doplňuje o možnost dohledu a do určité míry i dovoluje do procesu zasahovat. Software, vytvořený na bázi SCADA systémů je provozován nad hardware na vyšší úrovni a zajišťuje propojení a získávání dat z jednotlivých technologických komponent, jakými jsou PLC moduly, I/O (vstupně/výstupní) moduly, senzory. Komunikace s okolím probíhá přes průmyslové linky, např. ethernet využívající standardizované komunikační protokoly například Modbus. Výhodou SCADA systémů je schopnost zpracovávat velká množství dat odvíjející se od složitosti řízeného procesu. Vstupních proměnných může být až několik set tisíc, které vytvářejí v závislosti na rychlosti vzorkování obrovské množství dat, jež nebývá zvykem ukládat do textových souborů, ale posílat je na databázové servery. Díky tomu lze přistupovat k získaným datům i ze vzdálených pracovišť pomocí internetu. SCADA systémy se používají v energetice a v mnoha průmyslových odvětvích, ale jsou vhodné i pro technologie budov (řízení klimatizace, odvětrávání, vytápění). [20]

Vizualizace procesů a stavů se odehrává na zařízení nazvaném HMI (*Human Machine Interface*), což je název pro komunikaci mezi obsluhou a zařízením (HMI = rozhraní člověk – stroj). Nejčastěji se jedná o dotykový panel, na kterém je zobrazen monitorovaný proces. Dle složitosti procesu a počtu jednotlivých komponent podílejících se na správné funkci celého děje je proces zobrazen buďto pomocí jednoduššího schématu, kdy nejsou vyobrazeny méně podstatné komponenty, nebo jedním z následujících způsobů. Složitý proces lze zobrazit na plochu zobrazovacího panelu bez detailních částí s možností rozkliknout tyto části pro podrobné zobrazení komponent a jejich funkcí, nebo stavů. Další možností je celý proces rozdělit do několika částí, které spolu souvisí, ale pro zobrazení a zásahu do části procesu není nezbytné současně sledovat celý proces. Každá část tak bude zobrazena na samostatné obrazovce a přepínání mezi těmito obrazovkami lze realizovat přidáním a naprogramováním tlačítka na dotykové ploše, popř. na tuto akci nastavit jedno z fyzických tlačítek umístěných mimo dotykovou plochu na přední straně zobrazovacího panelu, jsou-li zde tato tlačítka k dispozici. [21]



Obr. 6 uvádí příklad použití HMI panelu pro zobrazení děje, při kterém se mimo jiné hlídá stav hladiny dvou přes ventily vzájemně propojitelných nádob. [22]



Obr. 6: Příklad užití SCADA systému

## 7 Popis jednotlivých částí sestavy

Tento projekt obsahuje sestavu, která má demonstrovat typické užití v automatických procesech. PLC jednotka s vlastním procesorem (CPU) obsahuje v paměti program, dle kterého se vykonávají instrukce. Modul PLC je propojen s dalšími jednotkami pomocí komunikační linky, přes kterou se přenáší zakódované informace v obou směrech. Driver (výkonový měnič, budič) dle instrukcí z PLC modulu řídí synchronní motor s permanentními magnety na rotoru, jenž má na hřídeli připevněný enkodér sloužící jako zpětná vazba pro regulaci pohonu.

Pro vizuální zobrazování aktuálních dějů v pohonu a pro snadnou obsluhu obsahuje sestava dotykový panel s displejem. Komunikace s výkonovým měničem a PLC modulem je realizována opět po komunikační ethernetové lince (viz kapitola Ethernet).

### 7.1 Napájecí zdroj SIMATIC PM1207

Napájecí zdroj je určen k montáži na DIN<sup>5</sup> lištu. Tento modul automaticky rozpoznává napájecí soustavu se jmenovitým napětím 120 V nebo 230 V a frekvencemi 50 Hz a 60 Hz. Na výstupu poskytuje stejnosměrné stabilizované napětí o hodnotě 24 V s tolerancí  $\pm 3\%$  a dokáže dodávat proud do zátěže až 2,5 A. Pro každý pól stejnosměrného napětí jsou vyvedeny z modulu dvě svorky, lze tedy napájet dvě zařízení bez nutnosti použít propojovací svorku. Vybrané technické údaje jsou vypsány v tab. 2. [23]

Tab. 2: Vybrané technické údaje modulu PM1207

Napájecí zdroj SIMATIC PM1207, 6EP1332-1SH71	
rozsah vstupního napětí	85 V ÷ 132 V nebo 176 V ÷ 264 V
rozsah frekvence napájecího napětí	47 Hz ÷ 63 Hz
vstupní proud	1,2 A při 120 V; 0,67 A při 230 V
jmenovité výstupní stejnosměrné napětí	24 V
rozsah výstupního proudu	0 A ÷ 2,5 A
zabudovaná pojistka	T 3,15 A/ 250 V
stupeň krytí	IP 20
třída ochrany	třída I

Zeleně svítící LED kontrolka indikuje přítomnost jmenovité hodnoty výstupního napětí. Na vstupní části obvodu je integrovaná pojistka T 3,15 A, kterou však nelze vyměnit bez rozebrání modulu. Výstupní strana je izolovaná od vstupní, splňuje požadavky na obvod SELV<sup>6</sup> dle normy EN 60950-1 a EN 50178. Výstupní část tedy zajišťuje ochranu před úrazem elektrickým proudem velikostí napětí, které je pro daný účel bezpečné a izolací od ostatních elektrických obvodů. [23], [24]

<sup>5</sup> DIN je kovová nosná lišta normalizovaných rozměrů a tvaru používaná v elektrotechnice, která slouží k fixaci elektrických přístrojů [25]

<sup>6</sup> SELV = Safety Extra Low voltage – bezpečné malé napětí

## 7.2 Přepínací modul SIMATIC NET CSM 1277

Tento modul slouží k propojení více zařízení, jež spolu komunikují pomocí ethernetu. Modul je napájen ze zdroje stejnosměrného napětí 24 V a odběr proudu při jmenovitém napětí činí 70 mA. O přítomnosti napájecího napětí správných parametrů informuje zeleně svítící LED dioda označená *DIAG*. Modul může být uzemněn přes nosnou DIN lištu, ale není to nezbytné pro správnou činnost zařízení. K propojení zařízení slouží čtyři zásuvky typu RJ-45. Zásuvky jsou rozdělené do dvou skupin, které jsou navzájem izolovány do napětí 1,5 kV. Skupina 1 zahrnuje zásuvky označené P1 a P2 a skupina 2 zbylé zásuvky P3 a P4. Výrobce doporučuje užít kroucený dvojlínkový kabel kategorie 5, nebo lepší (CAT 5+, CAT 6). Pro průmyslové kabely *Ethernet FC TP standard* s konektory *IE FC RJ-45 Plug 180* je povolená délka jednoho propojovacího kabelu až 100 metrů. Výhodou tohoto modulu je možnost použití jak křížených, tak přímých kabelů (funkce autocrossover MDI/MDI-X), tím se instalace pro uživatele značně zjednodušuje. Ke každému portu je přiřazena 1 zelená LED dioda, která indikuje spojení (svítí), nebo informuje o přenosu dat (bliká). Modul dokáže udržet v paměti až 2048 MAC adres připojitelných zařízení. Tab. 3 obsahuje vybrané technické údaje potřebné k užití modulu. [19], [26]

Tab. 3: Vybrané technické údaje modulu CSM 1277

Přepínací modul SIMATIC NET CSM 1277, 6GK7277-1AA10-0AA0	
rozsah stejnosměrného napájecího napětí	19,2 V ÷ 28,8 V
typ napájení	SELV – bezpečné malé napětí
stupeň krytí	IP 20
typ komunikačního konektoru	RJ-45

## 7.3 PLC modul SIMATIC SZ-1200 CPU 1215C

Modul je určen pro nejrůznější aplikace. Lze jej použít samostatně, nebo jako v mém případě pro komunikaci s ostatními zařízeními (výkonový měnič + motor, dotykový panel). Existuje několik variant modulu lišících se od sebe druhem napájení a typy výstupů. Pro snadnou identifikaci daného zařízení slouží písemný kód za typovým označením. Zvolený modul pro tento diplomový projekt nese označení *1215C DC/DC/DC* (a pro snadnou identifikaci produktové číslo *6ES7215-1AG40-0XB0*). První zkratka za typem modulu informuje uživatele o napájení stejnosměrným napětím 24 V, druhá zkratka o stejnosměrných 24V vstupech a třetí zkratka o taktě stejnosměrných 24V výstupech. Můžeme se setkat i s typem *AC/DC/Relay*, kdy je napájení zajištěno prostřednictvím síťového třívodičového kabelu z distribuční střídavé jednofázové sítě 120 V ÷ 240 V. Výstupy nejsou realizovány přes tranzistory, jako v předchozím případě, ale pomocí zabudovaných relé. První varianta umožňuje rychlejší spínání výstupů, má vyšší životnost, ovšem neumožňuje galvanické oddělení zátěže od modulu a k napájení zátěže lze použít pouze stejnosměrného zdroje 24 V. Kromě uvedených dvou typů nabízí výrobce typ *DC/DC/Relay*. Každý typ má své specifické vlastnosti vzájemně se lišící od ostatních typů, proto se zaměřím jen na zvolený modul, který je spolu s napájecím a přepínacím modulem zachycen na obr. 8. [11]

### 7.3.1 Typ 1215C DC/DC/DC

Vybraný typ vychází z předešlých řad 1211C, 1212C a 1214C. V technické dokumentaci (viz [11]) jej popisuje kapitola A.7, ve které výrobce přehledně rozepsal veškerou specifikaci produktu.

Modul obsahuje 14 digitálních vstupů, 2 analogové vstupy, 2 analogové výstupy a 10 digitálních výstupů. Vnitřní paměť procesoru je typu *Flash* velikosti 4 MB s možností rozšíření pomocí SD karty, na kterou je v horní části modulu určen slot (otvor). Ke komunikaci jsou zřízeny 2 konektory typu Ethernet MDI-X s uspořádáním pinů dle obrázku v tab. 1. Přenosová rychlost dat silně závisí na délce použitého kabelu a kategorie stínění. Pro kabel CAT5e udává výrobce přenosovou rychlost až 100 Mb/s. [11]

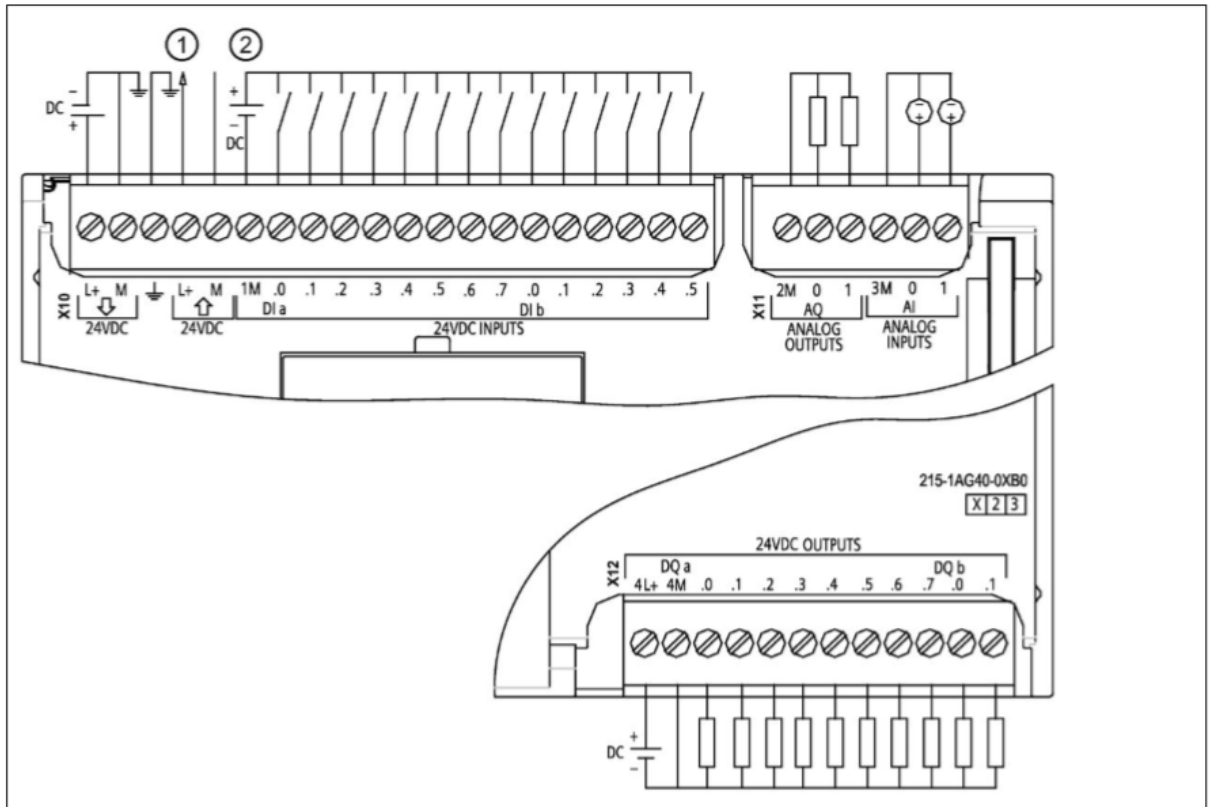
Další technické údaje týkající se napájení, napěťových rozsahů digitálních vstupů napěťového typu, proudových rozsahů digitálních výstupů proudového typu, rozlišení analogových vstupů, výstupů a jejich napěťové rozsahy zobrazuje tab. 4. [11]

Tab. 4: Vybrané technické údaje

rozsah stejnosměrného napájecího napětí	20,4 V ÷ 28,8 V
maximální odběr proudu při užití doplňků	1500 mA při 24 V
digitální vstup, logická 1	> 15 V při 2,5 mA
digitální vstup, logická 0	< 5 V při 1 mA
napěťový rozsah digitálních výstupů	20,4 V ÷ 28,8 V
maximální zátěž digitálních výstupů	5 W
napěťový rozsah analogových vstupů	0 V ÷ 10 V
rozlišení analogových vstupů	10 bitů
rozsah proudu analogového výstupu	0 mA ÷ 20 mA
rozlišení analogových výstupů	10 bitů

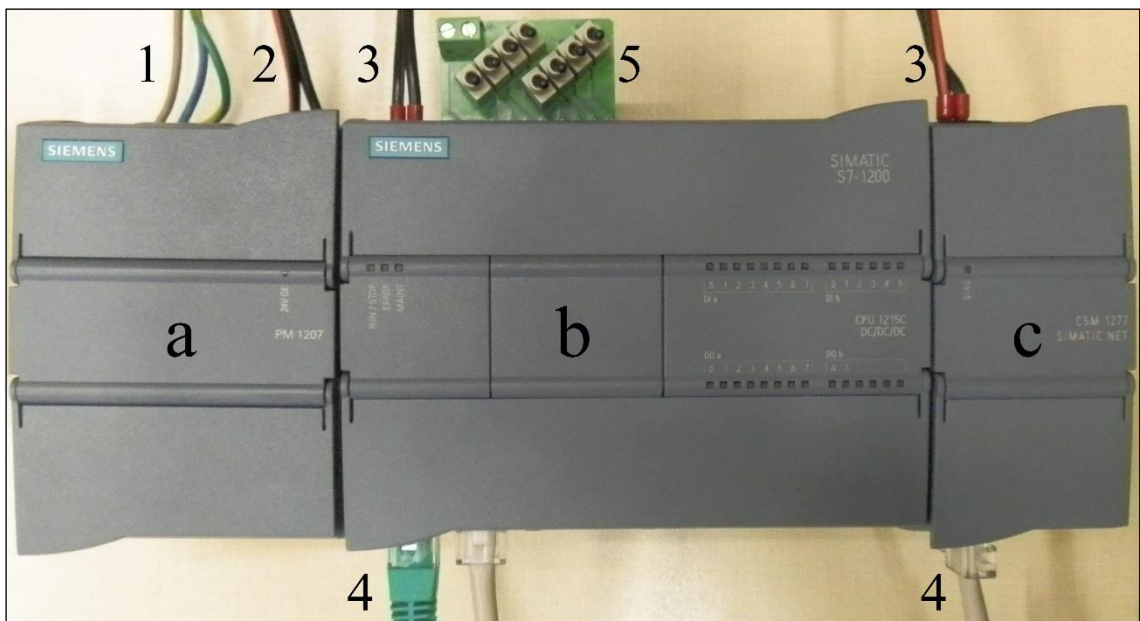
Digitální výstupy jsou realizovány zabudovanými tranzistory MOSFET, což umožňuje rychlé spínání výstupů. Výrobce udává zpoždění při zapínání maximálně 5  $\mu$ s pro výstupy označené Qa.4 až Qb.1 a 1  $\mu$ s pro výstupy označené Qa.0 až Qa.3 (první skupina). Vypínací časy jsou delší, pro první skupinu dosahují doby 3  $\mu$ s a u druhé skupiny až 20  $\mu$ s. Přesto jsou doby spínání o 3 řády kratší, než při užití mechanického spínání (relé). [11]

Schéma zapojení vstupů, výstupů a napájení modulu zachycuje obr. 7.



Obr. 7: Schéma zapojení PLC modulu 1215C DC/DC/DC [11]

- ① není-li použit výstup napětí 24 V, výrobce doporučuje propojit svorku M k uzemnění, aby nedocházelo k ovlivnění vstupů šumem z okolí.
- ② je-li užito vnější napájení digitálních vstupů, ke svorce 1M se připojí záporný pól externího napájecího zdroje (uvedeno na obrázku). V případě užití napájení výstupů přímo z PLC modulu, propojí se svorky M↑ a 1M a svorka L+↑ přes spínač k danému digitálnímu vstupu.

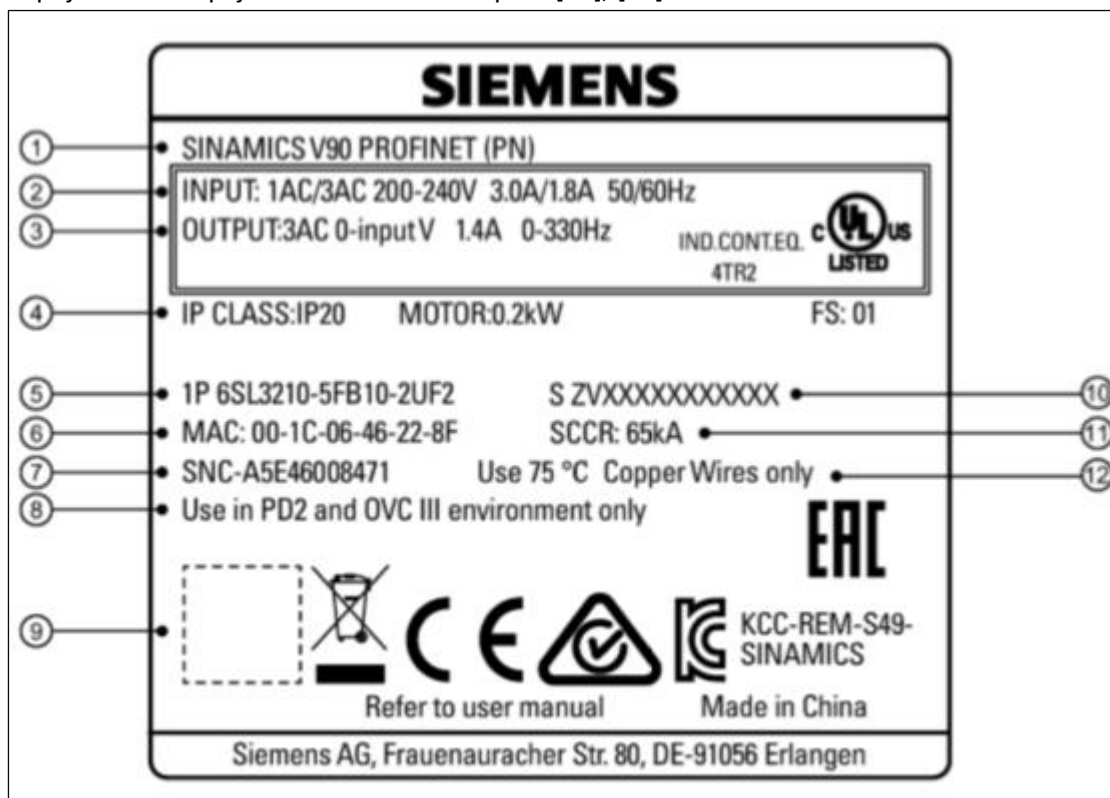


Obr. 8: Sestava a) napájecího modulu, b) PLC modulu, c) přepínacího modulu. 1) napájení 230 V AC, 2) stejnosměrný výstup 24 V, 3) napájení 24 V DC, 4) ethernetové připojení, 5) přípravek s 8 digitálními vstupy (*input simulator*)

## 7.4 Výkonový měnič SINAMICS V90

Tento měnič slouží k řízení třífázového synchronního motoru se zabudovaným enkodérem (senzorem pro snímání polohy rotoru). Modul má dvojí napájení: silové a ovládací. Silové napájení slouží k napájení motoru, a ovládací napájení je potřebné pro chod měniče. Napájení silové části lze zajistit prostřednictvím třífázové sítě 200 V, nebo z jednofázové sítě 230 V, kdy se zapojí dvě libovolné svorky. Dle obrázkové dokumentace uvedené v manuálu výrobce jsem zapojil fázový vodič na svorku L1 a nulový vodič na svorku L3. Ochranný vodič PE se připojí k uzemňovací svorce v dolní části měniče. Výrobce též doporučuje užít EMC filtru před vstupem silového napájení do měniče (není užito). Ovládací napájení je zajištěno ze stejnosměrného 24V zdroje SIMATIC PM1207. Propojení s PLC modulem je zajištěno opět přes PROFINET kabel. [27]

Konektor pro ovládací napětí obsahuje navíc 3 piny označené *STO1*, *STO+* a *STO2*, jež jsou z výroby propojené propojkou. Propojku lze odstranit a mezi piny *STO1* a *STO+*, popř. *STO2* a *STO+* připojit spínací kontakt relé bezpečnostního/vypínacího externího tlačítka pro bezpečné odpojení motoru při poruše. Toto blokování zamezuje opětovnému spuštění motoru. Rozepnutí obvodu lze provést jakýmkoliv rozpínacím tlačítkem, bude-li doba přerušení obvodu delší jak 0,5 s. Pro potřeby této práce jsem funkci použil, mezi svorky *STO1* a *STO+* jsem připojil vypínací tlačítko aktivované stiskem. Motor je tak možné v případě nebezpečí bezpečně odpojit. Po aktivaci této funkce a odstranění příčiny nebezpečného stavu je nezbytné měnič resetovat odpojením od napájecího i ovládacího napětí. [27], [28]



Obr. 9: Výrobní štítek měniče

Na obr. 9 je zobrazen výrobní štítek měniče, ze kterého lze vyčíst důležité technické údaje: [27]

- ① Název měniče
- ② Údaje o napájení měniče (počet fází, velikost napětí, proudu a frekvence)
- ③ Údaje výstupních veličin (počet fází, rozsah napětí, maximální výstupní proud na fázi a rozsah výstupní frekvence napětí)
- ④ Informace o výkonu měniče, stupni krytí
- ⑤ Výrobní číslo produktu
- ⑥ MAC adresa
- ⑦ Číslo dílu
- ⑧ Informace o stupni znečištění a velikosti přepětí (pracovní podmínky)
- ⑨ QR kód pro načtení technických údajů čtecím zařízením obsluhy
- ⑩ Sériové číslo výrobku
- ⑪ Velikost jmenovitého zkratového proudu
- ⑫ informace o použití měniče jen pro motory s měděným vinutím

Výrobního číslo produktu 6SL3210-5FB10-1UF1 obsahuje vybrané technické údaje: [27] písmeno B informuje o možnosti napájení z jednofázové, nebo třífázové sítě 200 V ÷ 240 V, následující trojice čísel, v tomto případě 101 odkazuje na technické údaje motoru (maximální výkon motoru -  $P_{\max} = 0,1$  KW, maximální jmenovité napětí -  $U_{\max} = 200$  V), písmeno F nese informaci o zvoleném způsobu komunikace měniče s nadřazeným systémem, která je zprostředkována ethernetovým připojením

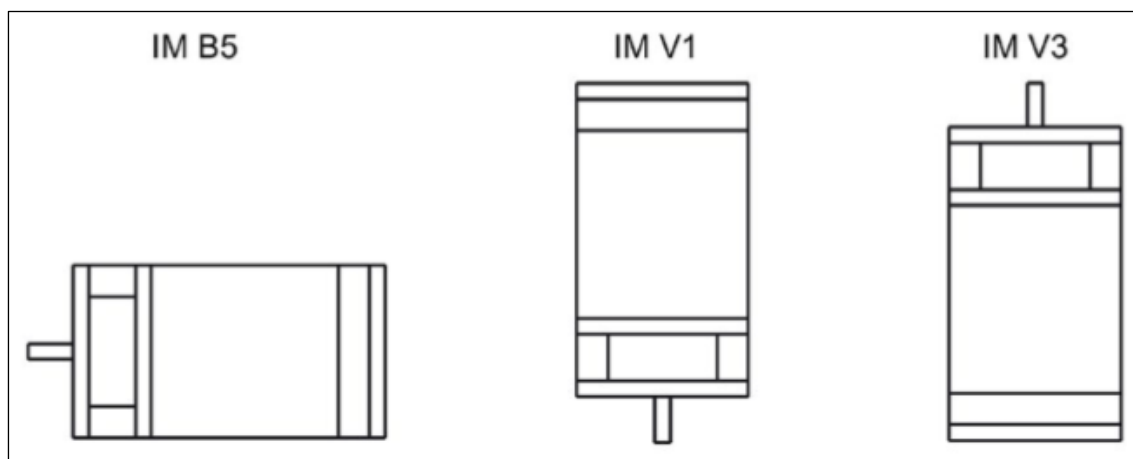
Výrobce nabízí měniče s možností napájení z jedné nebo třífázové soustavy pro motory do výkonu 2 kW a dále měniče napájené z třífázové soustavy (s rozsahem jmenovitého napětí 380 VAC ÷ 480 VAC) pro motory do příkonu 7 kW. [27]



Obr. 10: Výkonový měnič V90; 1) napájení 230 V AC, 2) napájení 24 V DC, 3) napájení motoru, 4) vstup signálu z enkodéru, 5) ethernetová linka, 6) USB výstup pro komunikaci programu V-Assistant s měničem a nahrávání dat do vnitřní paměti měniče. Pozn. Konektor 2) obsahuje též piny pro aktivaci funkce STO

## 7.5 Motor SIMOTICS S-1FL6

Tento třífázový synchronní motor s permanentními magnety na rotoru (viz obr. 13) je v provedení s nízkou setrvačností. Napájení je zprostředkováno přes měnič SINAMIC V90. K propojení s měničem slouží silový čtyř žilový kabel oranžové barvy, který se k měniči připojí přes příslušný konektor na svorky U, V a W. Ochranný vodič se připojí na zemnicí svorku na měniči a též se připojí stínicí oplet kabelu na svorku uzemnění. K motoru je připevněn enkodér (viz další podkapitola) sloužící k určení polohy rotoru. Propojení enkodéru a řídicí jednotky je zprostředkováno samostatným deseti žilovým kabelem, který se na měniči připojí do konektoru označeného X9. Kabel zelené barvy je taktéž stíněn, přičemž výrobce uvádí procentuální velikost krytí vyšší jak 60 %. Celý motor je v provedení krytí proti vniknutí cizích částic a vody IP65 s hřídelovým těsněním založeném na olejové bázi, ovšem konektory na koncích kabelů splňují krytí jen IP20. Tyto konektory jsou odlišné od konektorů na výkonovém měniči. K propojení motoru a měniče je nutné použít propojovací kabel stejné konstrukce a takové délky, aby bylo možné měnič instalovat do míst jemu určených (platí to pro napájecí kabel i kabel vycházející z enkodéru). Motor má vlastní chlazení a je určen do provozu s maximální relativní vlhkostí 90 % se zamezením kondenzace (platí pro teplotu okolí 30 °C). Třída izolace B umožňuje oteplení o 80 °C (pro rotorové vinutí, resp. o 85 °C pro statorové vinutí) při teplotě okolí 40 °C, a navíc musí zahrnovat rezervu 10°C. Výsledná teplota se tedy může vyšplhat až na 130°C. Hodnota je platná pro vnitřní povrch izolace a při překročení této hodnoty dochází k rychlejší tepelné degradaci izolace a ta ztrácí své izolační vlastnosti. V praxi to znamená nutnost snímání teploty uvnitř stroje anebo vhodným přepočtem alespoň povrchovou teplotu stroje pro konkrétní pracovní podmínky a teplotu monitorovat, aby nedocházelo k jejímu překročení. Motor může být upevněn v polohách IM B5, IM V1 a IM V3, jedná se tedy o přírubový motor. Možné způsoby montáže přibližuje obr. 11. Následující tab. 5 přináší přehled technických údajů platných pro zvolený typ motoru. [27], [28], [29]



Obr. 11: Možné způsoby upevnění přírubového motoru [27]



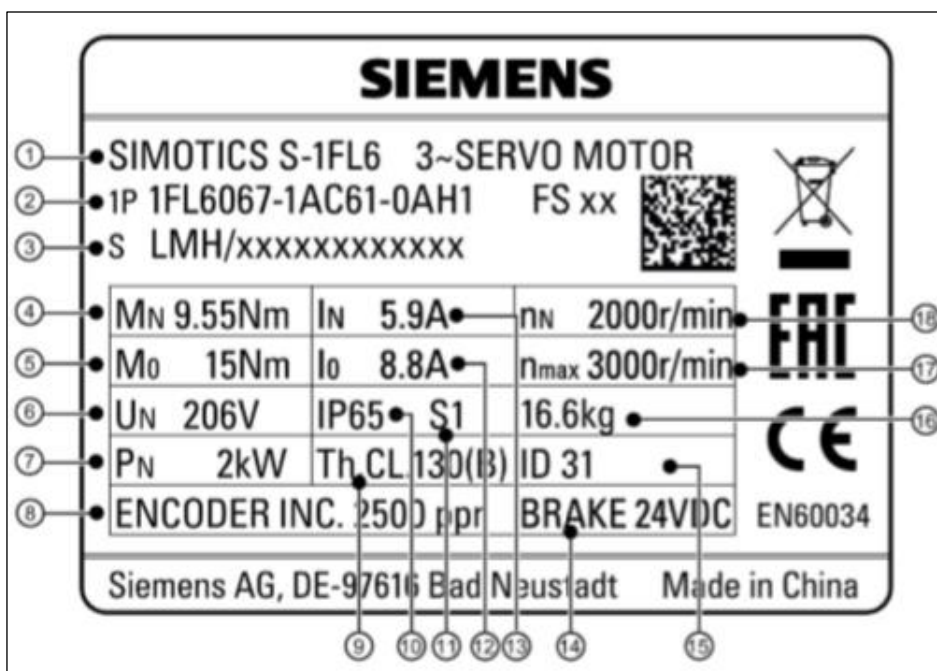
Tab. 5: Technické údaje motoru

SIMOTICS S-1FL6, 1FL6024-2AF21-1AG1	
jmenovitý výkon	100 W
jmenovitý moment	0,32 Nm
maximální moment	0,96 Nm
jmenovité otáčky	3 000 ot·min <sup>-1</sup>
maximální otáčky	5 000 ot·min <sup>-1</sup>
jmenovitá frekvence	200 Hz
jmenovité napětí	73 V
jmenovitý proud	1,2 A
maximální proud	3,6 A
moment setrvačnosti	5,2 · 10 <sup>-6</sup> kg·m <sup>2</sup>
maximální doporučená setrvačnost zátěže	156 · 10 <sup>-6</sup> kg·m <sup>2</sup> *
rozsah provozní teploty	0 °C ÷ 40 °C **

\* 30násobek momentu setrvačnosti motoru

\*\* bez snížení výkonu motoru

Typický výrobní štítek motoru od firmy Siemens na obr. 12 přehledně zobrazuje rozvržení technických údajů o motoru. Vyobrazený štítek je pouze demonstrativního rázu, technické parametry použitého motoru jsou vypsány v tab. 5



Obr. 12: Příklad štítku motoru firmy Siemens [27]

- |   |                                 |
|---|---------------------------------|
| ① typ motoru                            | ② výrobní číslo                 |
| ③ sériové číslo                         | ④ jmenovitý točivý moment       |
| ⑤ moment při zabrzděném rotoru          | ⑥ jmenovité napětí              |
| ⑦ jmenovitý výkon                       | ⑧ typ enkodéru a jeho rozlišení |
| ⑨ tepelná třída izolace motoru          | ⑩ stupeň krytí                  |
| ⑪ druh zatížení (S1 – trvalý chod) [29] | ⑫ proud při zabrzděném rotoru   |
| ⑬ jmenovitý proud                       | ⑭ přidržovací brzda             |
| ⑮ ID motoru                             | ⑯ hmotnost                      |
| ⑰ maximální otáčky                      | ⑱ jmenovité otáčky              |

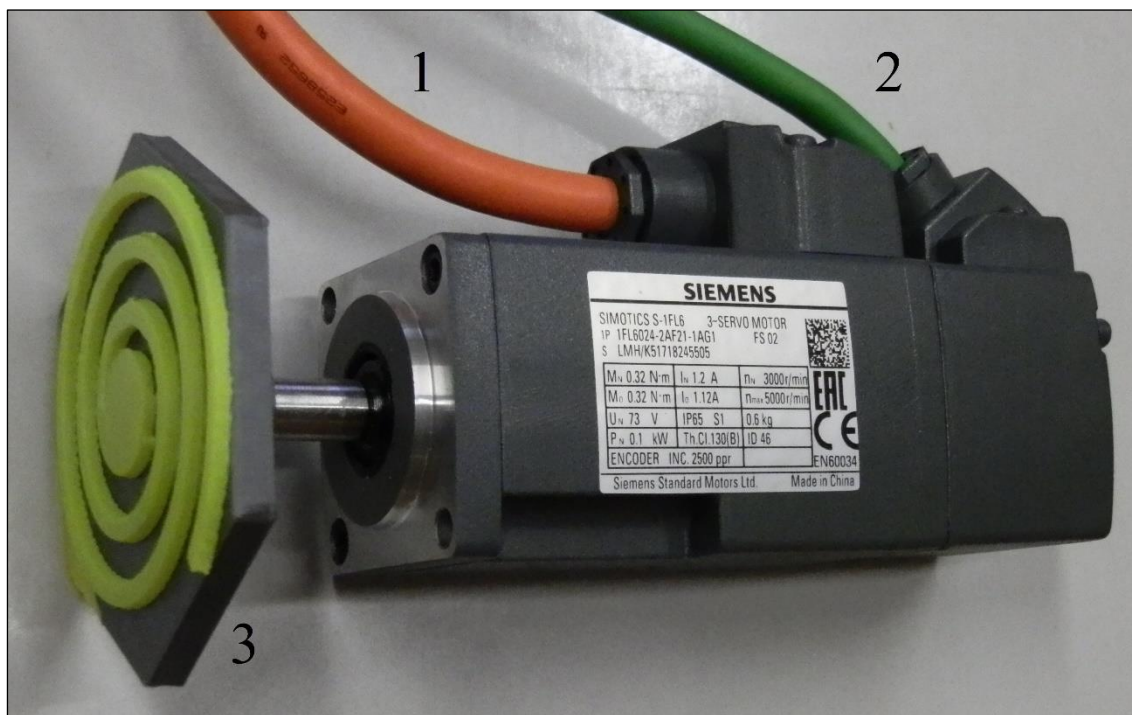
Obdobně jako u měniče obsahuje výrobní číslo 1FL6024-2AF21-1AG1 informace o motoru.

Tab. 6: Kódové značení motoru

pořadová číslice (x)	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
kódové označení	1	F	L	6	0	2	4	2	A	F	2	1	1	A	G	1

Pořadová číslice x nese informaci o:

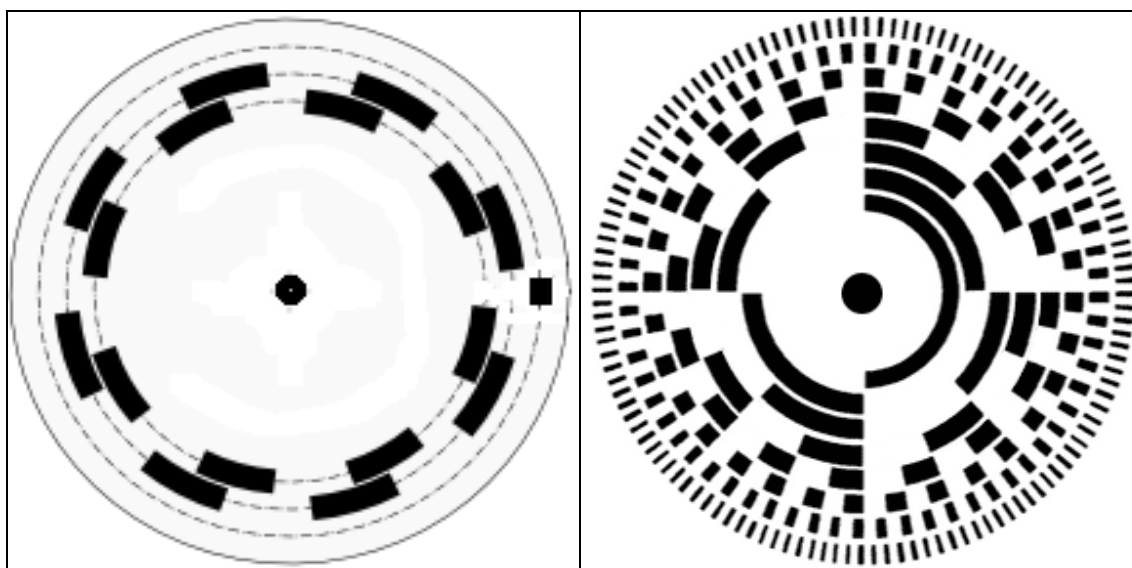
- 5. + 6.: výšce hřídele: 20 mm,
- 7.: jmenovitém točivém momentu: 0,32 Nm,
- 8.: setrvačnosti motoru: nízká setrvačnost,
- 10.: jmenovitých otáčkách: 3 000 ot·min<sup>-1</sup>,
- 11.: napájecím napětí: 200 VAC
- 13.: typu připojení motoru: kabelový vývod,
- 14.: typu použitého enkodéru: inkrementální enkodér TTL s rozlišením 2 500 ppr (2 500 pulzů na otáčku) [30],
- 15.: hřídeli: rovná hřídel bez brzdy,
- 16.: stupni krytí: IP65, hřídelové olejové těsnění



Obr. 13: Synchronní motor s enkodérem; 1) napájecí kabel z měniče, 2) výstupní kabel z enkodéru, 3) hřídelový terčik pro zviditelnění dějů

## 7.5.1 Enkodér

Enkodér je zařízení, které určuje polohu rotační části zařízení, nejčastěji motoru, a proto bývá přímo jeho součástí. Sestává z pohyblivé části umístěné na rotační části zařízení (například na hřídeli motoru nebo převodovky) a z pevné části. Enkodéry lze rozdělit na mechanické, kdy k určení otáčení slouží vodivé plošky na pohyblivém kotoučku a sběrače umístěné na pevné části a na optické, kdy k detekci pohybu slouží světelná závora a štěrbiny v kotouči, nebo se střídá černá a bílá barva kotouče a vyhodnocovací zařízení sestává z infračervené LED diody a fototranzistoru v témže pouzdře. Optický enkodér je užit i u výše popisovaného motoru a je označen zkratkou HSC (High Speed Control). Dále lze rozdělit enkodéry na absolutní a inkrementální. Inkrementální čidlo potřebuje ke své správné funkci výchozí (nulovou) polohu, do které musí zařízení nejprve najet a pak teprve vyhodnocování polohy bude správné. K tomuto účelu slouží resetovací impuls, který enkodér vygeneruje jednou za otáčku. Absolutní čidlo umožňuje určit správnou polohu i bez referenčního bodu. Rozlišení absolutního čidla závisí na počtu jednotlivých stop. Některé enkodéry dokáží určovat nejen polohu, ale také rychlost a směr otáčení. Pro určení směru je nutné enkodér vybavit druhou světelnou závorou a systémem štěrbin (popř. vodivých plošek a sběračů u mechanických čidel), přičemž impulzy od jedné světelné závory předstihují impulzy druhé sady štěrbin. Oproti absolutnímu čidlu jsou všechny impulzy stejně dlouhé. Na obr. 14 vlevo je zachycen kotouč inkrementálního čidla včetně resetovacího znaku a vpravo je zachycen kotouč absolutního čidla s 8 stopami, tedy s rozlišením 8 bitů ( $2^8 = 256$  úrovní na otáčku). [11], [31], [32]



Obr. 14: Kotouč inkrementálního čidla (vlevo) a absolutního čidla (vpravo) [31], [32]

## 7.6 KTP700 Basic, 6AV2 123-2GB03-0AX0, V-K5AS8061

Dotykový ovládací panel (viz obr. 15) je určený k zástavbě do ovládacího pultu ve vnitřních prostorech. Kromě dotykového barevného LCD TFT<sup>7</sup> displeje je na panelu 8 tlačítek pro snadnější ovládání a přístup k jednotlivým operacím. 7palcový displej má rozlišení 800 x 480 pixelů a je schopen zobrazit 65 536 barev (16 bitů). Pro paměť dat je vyhrazeno 256 MB a pro paměť programu 512 MB. Napájení panelu činí 24 V stejnosměrných (povolený rozsah 19,2 V ÷ 28,8 V) a při nedodržení správného zapojení může dojít ke zničení zařízení. Maximální spotřeba panelu činí 5,5 W. K uzemnění slouží uzemňovací svorka a pro správnou činnost je nutné panel uzemnit. Výrobce doporučuje k uzemnění užít měděný vodič o průřezu 4 mm<sup>2</sup> na který je zemnicí svorka dimenzována. Ke komunikaci panelu s ostatními zařízeními slouží konektor RJ-45. Panel je vybaven též USB portem. USB 2.0 slouží k připojení externích zařízení, například klávesnice nebo myši, přičemž tato externí zařízení by se měla připojovat pouze při uvádění zařízení do provozu nebo při servisu. Připojování externích zařízení napájených napětím 230 V výrobce z hlediska bezpečnosti nedoporučuje. Dotykový panel je citlivý k elektrostatickým výbojům a k přepětí, tudíž je nutné k němu přistupovat jako k ESD<sup>8</sup> zařízení, zvláště při manipulaci a montáži. Provozní teplota závisí na typu uložení panelu včetně na sklonu a orientaci displeje. Maximum provozní teploty se pohybuje v rozmezí od 35 °C do 50 °C. Zařízení správně pracuje v rozsahu relativní vlhkosti 10 % až 90 %, je-li zamezena kondenzace vodních par. Se vzrůstající relativní vlhkosti (od cca 30 %) klesá provozní teplota až na hodnotu 30 °C při relativní vlhkosti 90 %. [21]

Je-li panel správně namontován, z horní strany splňuje krytí IP 65 a ze spodní strany IP 20. Nastavení panelu lze provést dvěma způsoby: a) buďto přímo pomocí dotykového displeje, nebo b) z PC přes ethernetový kabel. Druhá varianta vyžaduje nainstalování příslušného softwaru do PC a je uživatelsky náročnější. Ovládací panel taktéž lze připojit přes ethernetový kabel přímo do PLC modulu. Panel pracuje ve třech pracovních režimech. "Offline" režim neumožňuje výměnu dat s připojeným zařízením, v "Online" režimu spolu zařízení komunikují a v režimu "Transfer" lze ovládací panel nastavovat z nadřazeného systému. [21]

---

<sup>7</sup> TFT – Thin Film Transistor, užití tenkovrstvých tranzistorů v LCD displejích (každá buňka obsahuje jeden spínací tranzistor). Úkolem tranzistoru je připojit nebo odpojit danou buňku od obvodu. LCD – Liquid Crystal Display neboli zobrazovací jednotka z kapalných krystalů. [33]

<sup>8</sup> ESD (Electrostatic Sensitive Devices) jsou zařízení a součástky náchylné k elektrostatickému výboji [34]



Obr. 15: Dotykový panel; 1) stejnosměrné napájení 24 V, 2) komunikační výstup, 3) řada s 8 tlačítky, kterým lze programově přiřadit libovolnou funkci, 4) dotyková plocha s úhlopříčkou 7 palců (17,8 cm). Pozn.: v blízkosti napájecího konektoru se nachází uzemňovací svorka (vodič připojen později, není na snímku)

## 8 Vývojové prostředí

Vývojové prostředí sestává z několika programů, ve kterých se nastavují parametry zařízení, definují adresy zařízení pro vzájemnou komunikaci, nahrávají parametry do pamětí, případně simulují naprogramované úkony, kdy není třeba připojit skutečná zařízení. Jednotlivé vývojové programy jsou navrženy tak, aby vzájemně spolupracovaly a mohly být spuštěné současně. Následující podkapitoly obsahují popis funkcí a informace o daných programech. Kromě těchto níže popisovaných programů existují i další, které umožňují vytvářet složitější a obsáhlejší projekty. Příkladem může být vývojové prostředí *WinCC flexible Advanced* sloužící k vizualizaci složitých procesů, které obsahuje knihovny s velkým množstvím ikon představujících nejrůznější objekty a dovoluje vývojáři tvořit složité děje, podrobně sledovat jednotlivé stavy a tyto stavy měnit. [35]

Jelikož jsem nevytvářel natolik složitý proces, abych musel tento program použít, nebudu se o něm dále zmiňovat.

### 8.1 V-ASSISTANT

Program je potřebný pro nastavení parametrů motoru a výkonového měniče (budiče). Pro správnou komunikaci s měničem probíhající přes USB rozhraní je nutno napájet ovládací i silové obvody měniče (přítomnost napětí 230 VAC, 24 VDC). Program rozpozná zařízení a po připojení načte údaje o měniči a připojeném motoru. Základní parametry obou zařízení jsou v programu viditelné společně s typovým označením, které musí být shodné s typovým označením motoru i měniče (program dokáže správně identifikovat připojený HW, přesto umožňuje změnit jak měnič, tak motor). V tomto programu se nastaví IP adresa měniče automaticky, lze ji však změnit dle potřeby. Též se automaticky přiřadí MAC adresa a název zařízení. Název zařízení a IP adresa jsou velmi důležitými informacemi pro práci s HW v programu TIA Portal. Dalším důležitým krokem je ladění zpětnovazebního řízení motoru přes měnič, který přijímá informace z enkodéru motoru. Toto ladění proběhne po povolení od uživatele zcela automaticky a výsledkem jsou parametry regulátoru, které se následně nahrají do paměti měniče. Pro ověření správné funkce měniče slouží test motoru, kdy se zadají požadované otáčky i směr otáčení hřídele a v reálném čase jsou zobrazovány informace o otáčkách, či proudu fáze. Dalším krokem je spuštění vývojového prostředí pro nastavení komunikace s ostatními zařízeními (PLC, ovládací panel) a vytvoření programu pro procesor automatu.

### 8.2 TIA PORTAL

Hlavní část vývojového prostředí, slouží k vytvoření projektu, do kterého se nahrají reálné komponenty HW. Ty lze přidávat z knihovny obsahující veškeré možné komponenty určené pro automatizaci, které společnost nabízí. Nejprve je nutné přidat PLC automat, protože pro něj je určen výsledný program (nahrává se do paměti v PLC modulu). Komunikace vývojového prostředí s automatem a dalšími prvky probíhá přes ethernetové propojení. Díky existujícímu připojení lze navázat spojení s automatem přímo, bez přidávání konkrétního zařízení z knihovny a přidat jej do okna obsahujícího všechna používaná zařízení tímto způsobem. Zároveň tak dojde k přiřazení jména, IP a MAC adresy automatu.

Obsahuje-li sestava další prvky, přidají se do projektu výběrem z knihovny. Opět musí souhlasit typová čísla reálného objektu a objektu v projektu, jelikož existuje řada příbuzných produktů lišících se právě produktovým číslem a pozměněnými funkcemi zařízení. V podokně propojení zařízení se jednotlivé komponenty propojí buďto přiřazením IP adresy k danému portu, nebo propojením portů pomocí nabízené možnosti zobrazené na těle zařízení po jeho přidání z knihovny. Pro vložení měniče a motoru je nutné nejprve nahrát parametry do skutečného měniče v programu V-Assistant, protože zde se do projektu přidává měnič a motor jako jeden celek bez možnosti zápisu parametrů do paměti měniče. Po přidání měniče s motorem do projektu se musí přepsat IP adresa a jméno tak, aby bylo zcela shodné s údaji v programu V-Assistant, které byly nahrány do paměti měniče. Řízení motoru se realizuje přes technologické objekty. V technologickém objektu vytvořeném v projektu představujícím motor (respektive jeho hřídel, proto název *Axis* = osa) se přiřadí měnič ovládající motor, dále způsob zapojení enkodéru a vytvoření uzavřené smyčky pro zpětnovazební řízení.

Pro řízení digitálních vstupů a výstupů PLC modulu slouží v programu TIA Portal tři standardní programovací jazyky. Jednotlivé bloky tvořené vývojovým prostředím mohou používat jakýkoliv z těchto 3 nabízených jazyků podle aktuální potřeby, nebo podle specifikace daného bloku. Pro každý blok může být výhodnější použít jiný jazyk ať už z hlediska přehlednosti nebo různého stupně složitosti. Program vytvořený z bloků využívajících různé programovací jazyky nijak neovlivňuje správnou funkci výsledného programu díky kompatibilitě jednotlivých funkčních jazyků. [11]

Prvním z nich je LAD (*ladder logic*) popisovaný v kapitole 3.1, který pomocí spínacích a rozpínacích kontaktů a cívek tvoří uzavřené ovládací obvody. V těchto schématech lze tvořit paralelní větve pro překlenutí vybraných kontaktů při dodržení určitých pravidel. Například překlenutím spínacího tlačítka pomocným spínacím kontaktem příslušného relé lze držet obvod sepnutý i po opětovném rozepnutí spínacího tlačítka a obvod rozpojit pomocí rozpínacího tlačítka umístěným před, nebo za uzlem paralelní větve. Při tvoření žebříkového schématu je důležité dodržovat pravidlo, že energie smí téct pouze jedním směrem zleva doprava a nikdy se nesmí vracet. Též se nesmí užívat paralelní větve bez jediného kontaktu pro překlenutí jiného prvku, kterým by byla tímto znemožněna funkce spínat, či rozpínat příslušnou linii. [11]

Druhým programovacím jazykem je diagram funkčních bloků se zkratkou FBD, který je taktéž grafický. Funkční bloky prezentují logické operace používané v Booleově algebře. Nespornou výhodou vývojového prostředí TIA Portal je počet instrukcí, který není v síti FBD nijak omezen. Bloky AND (logický součin) a OR (logický součet) zastávají stejnou funkci jako paralelní větve a linie s rozpínacími a spínacími kontakty používanými u předešlého jazyka. [11]

Posledním z podporovaných programovacích jazyků je SCL (*Structured Control Language*), který vychází ze strukturovaného textu podobný Pascalu. Instrukce jazyka používají klasické programovací operátory, jakými jsou například matematické funkce (sčítání, odčítání, násobení, dělení). SCL převzal z Pascalu standardní operace řízení: if-then-else, case, repeat-until, return a goto. Mnoho instrukcí v jazyce SCL odpovídá instrukcím jazyků LAD a FBD, příkladem jsou čítače a časovače. [11]

V tomto vývojovém prostředí se nejčastěji užívá zápisu pomocí bloků. Bloky jsou nabízeny ve čtyřech skupinách: *organization block* (organizační bloky), *function block* (funkční bloky), *function* (funkce) a *data block* (bloky dat). Organizační bloky reagují na události v CPU, mohou přerušit běh uživatelského programu. V programu se nachází hlavní organizační blok označený *OB 1*, který provádí cyklické opakování uživatelského programu. Další organizační bloky slouží k provádění specifických operací, např. zpracování přerušení, nebo vykonávání části programového kódu v konkrétních časových intervalech. Podprogramy, které se vykonávají při zavolání z jiné části kódu jsou funkčními bloky. [11]

Při zavolání se do konkrétního funkčního bloku předají parametry a současně se ukládají data do datového bloku, přičemž jeden funkční blok může ukládat data do několika datových bloků a tím řídit více dějů (každý děj má vlastní datový blok). Blok funkce je také podprogram, stejně jako u funkčního bloku je volán jiným kódovým blokem, ale nemá přiřazený datový blok. Data se tedy ukládají buďto do globálního datového bloku, nebo zapisují do paměti procesoru na konkrétní adresu. Způsob vytváření kódu závisí na konkrétním uživateli, na složitosti daného procesu a dalších parametrech. Struktura může být lineární, nebo modulární. Lineární program obsahuje organizační blok, uvnitř kterého se vykonají postupně všechny instrukce bez volání podprogramů. Modulární struktura programu obsahuje jednotlivé kódové bloky pro zpracování specifických operací odpovídajících technologickým funkcím procesu, které jsou volány z hlavního programu. Modulární forma zápisu se uplatní zejména v procesech, kde se vykonává řada podobných úkonů lišících se některými parametry (rychlost, vzdálenost, ...). Vytvoří se tedy jeden obecný podprogram, který je pro jednotlivé procesy modifikován dle konkrétních požadavků. Obecné bloky lze navíc ukládat do knihovny a používat je v jiných projektech. Strukturovaným formám zápisu se přiklání také z důvodu vyšší přehlednosti, rychlejší obměně kódu (aktualizaci) a možnosti testovat funkčnost jednotlivých bloků zvlášť. [11]

Projekt vytvořený ve vývojovém prostředí TIA PORTAL STEP 7 smí obsahovat bloky ze všech tří programových jazyků, bloky lze přidávat a definovat pomocí SCL editoru. Složitější instrukce lze vybrat z nabídky, která je součástí editoru v podobě stromu instrukcí. K napsání instrukcí lze použít i textový editor a poté jej do prostředí vložit. Vzorec pro výpočet hodnoty (výraz) se skládá z operandů, kterými jsou například konstanty, výrazy a z operátorů. Operátory mají přidělený různý stupeň priority – vykonávají se předepsaným způsobem. Nejprve se vyhodnotí operátory s nejvyšší prioritou, naposled operátory s nejnižší prioritou. Nejvyšší prioritu má přiřazení (:=), nejnižší prioritu má z matematických operací mocnění. Pro změnu sledu vyhodnocování operátorů je nutné závorkovat. U operátorů se stejnou prioritou probíhá jejich zpracování zleva doprava. Výsledek výrazu lze použít jako podmínku, přiřazení hodnoty, volání kódového bloku atd. Při užití aritmetických operátorů není nutné používat stejné číselné datové typy, přičemž datový typ výsledku je dán datovým typem nejvýznamnějšího operandu. Řídící příkaz realizuje větvení programu, opakování části programu, skok na jiné části kódu nebo podmíněné provedení výrazu. Pro zpřehlednění kódu se doporučuje příkaz rozdělit do několika řádků. Oddělovače, jako jsou mezery a konce řádků se při kontrole syntaxe ignorují. Ukončení řídicího příkazu zajišťuje příkaz END. [11]

Po vytvoření programu určeným pro zápis do paměti PLC dojde k jeho přeložení a zkontrolování syntaxe. Veškeré nedostatky se po přeložení vypíší do podokna v podobě varování a chyb. Varování nevylučují nahrání programu do paměti, ovšem je nutné zvážit jejich případný dopad na běh programu. Neobsahuje-li vytvořený kód žádnou chybu, lze jej nahrát do paměti PLC modulu. Prostředí TIA Portal umožňuje sledovat v reálném čase děje uvnitř modulu a měnit je programově.

Pokud byl program tvořen pomocí LAD, lze na něm sledovat, které části linií jsou aktivní (jsou pod napětím +24 V – linie, nebo spínač je zbarven zeleně) a které jsou bez napětí (zbarveno černě). Stav tlačítek lze programově měnit (modifikovat): sepnuto/rozepnuto. V případě, že příslušná linie uzavírá elektrický obvod, je relé dané linie napájené a jeho kontakty aktivní (spínací kontakty sepnuté, rozpínací kontakty rozepnuté). Obdobně lze testovat správnou funkci programu přímo na modulu při použití fyzických spínačů umístěných na přídatném modulu připojitelných k digitálním vstupům PLC modulu. Stav digitálních vstupů a výstupů zobrazují zelené LED na těle automatu. Digitální výstupy jsou v programu nahrazeny symboly cívek relé.



V případě spuštění projektu a sledování stavů v online režimu poskytují bloky čítačů, časovačů, zpožďovacích relé údaj o počtu pulzů (čítač), nebo časový údaj. Lze tedy sledovat, kdy dojde k sepnutí/rozepnutí příslušné části obvodu.

### 8.2.1 Popis vybraných bloků

Ve vývojovém prostředí lze vytvořit opět žebříkové schéma (LAD) v okně *Main*, do kterého se přidají z knihovny bloky (funkční bloky) pro ovládání osy motoru. Bloky jsou shodné pro programovací jazyk LAD a FBD, ale lze je realizovat i zápisem kódu v jazyce SCL. Manuál [11] uvádí příklady odlišné realizace pro nejvíce používané funkce. Každý blok určený pro ovládání konkrétního motoru (technologického objektu<sup>9</sup>– osy) vyžaduje přiřazení dané osy, povolení bloku, aktivaci bloku a další parametry charakteristické pro daný blok. Bloky *MC\_Power* a *MC\_Reset* slouží pro povolení, nebo zakázání osy a k vynulování stavu osy (resetu). Bloky obsahují nejen parametry, které je nutno definovat, ale i parametry přednastavené. Tyto parametry se týkají například bezpečnostních funkcí umožňujících nouzové zastavení osy, které může být okamžité, nebo postupné snižováním otáček na nulu. Některé parametry jsou pouze informačního charakteru, nelze je nastavovat. Funkcí bloku *MC\_Reset* je opětovná aktivace osy po jejím zablokování chybovým stavem. Pro opětovné povolení osy je nutné problém odstranit. Blok *MC\_Home* nastavuje absolutní polohu osy. Blokem *MC\_Halt* se zastavuje osa motoru, přičemž klidová poloha není definována (nezohledňuje se poloha osy vůči výchozí pozici). V bloku *MC\_MoveRelative* se nastavuje povolení, přiřazení konkrétní osy, vykonání (spuštění) pohybu, rychlost, která je přednastavená na 10,0 ot/min a vzdálenost (v milimetrech). Pro určení vzdálenosti ze zadaného čísla je nutné znát přepočít, který je nastavitelný v okně příslušné osy a prezentuje pohyb zátěže na otáčku motoru, vývojářem programu nastaveno na 10 mm/otáčka. Přepočít závisí na použitém druhu mechanismu, jenž převádí rotační pohyb na posuvný, je-li potřebný, například pro pásový dopravník. Při jmenovitých otáčkách použitého motoru 3 000 ot/min bude rychlost pohybu přepravovaného objektu 500 mm/s. [11]

---

<sup>9</sup> Technologický objekt je sám o sobě jedním nebo několika organizačními bloky, do kterých nejde zapisovat a jsou chráněné (nelze tedy zjistit ani programový kód), kdežto ovládání jednotlivých procesů tohoto technologického objektu se realizuje pomocí funkčních bloků.

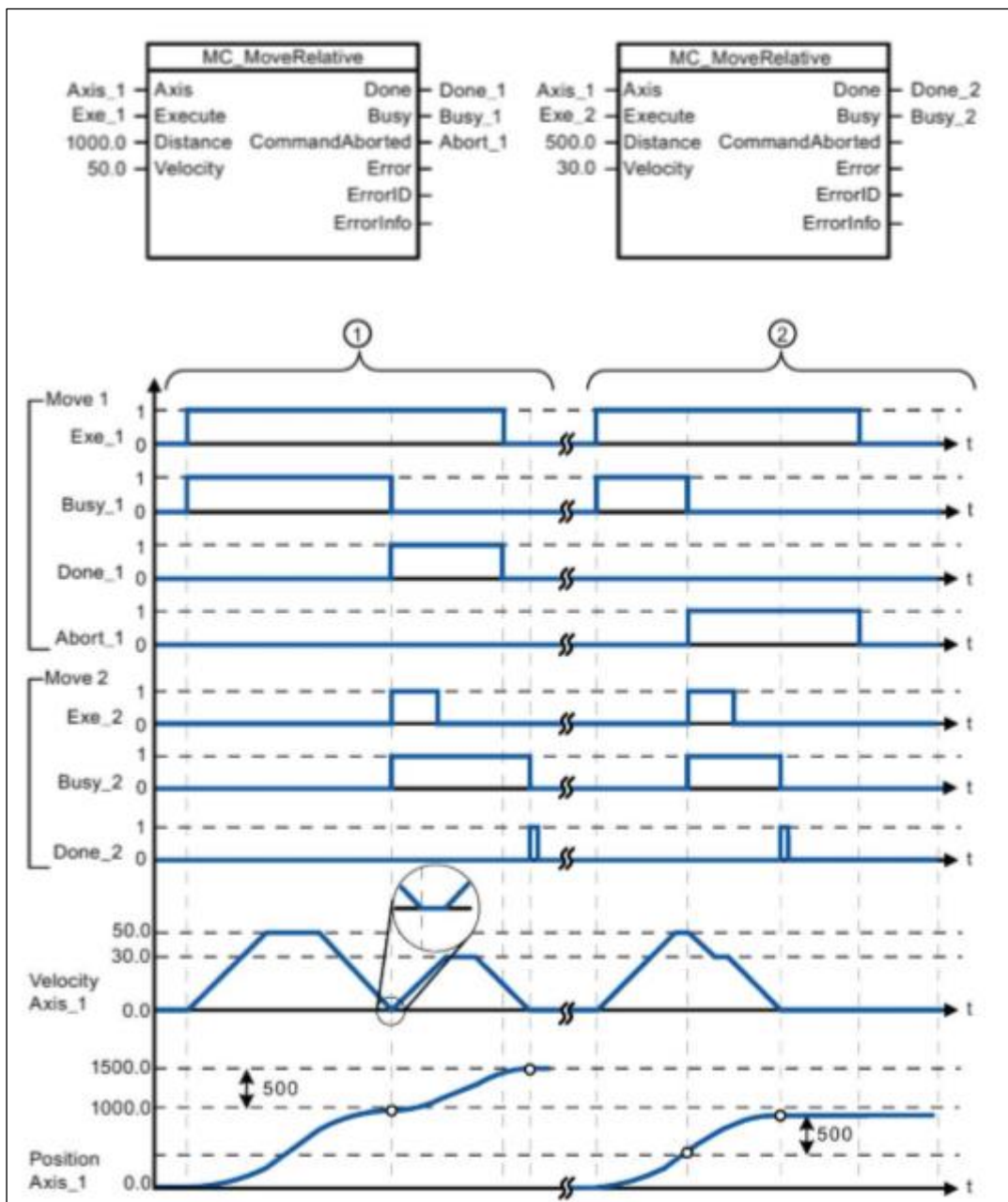
Obr. 16 na protější straně popisuje chování osy při použití bloku *MC\_MoveRelative*. Nejprve bylo v konfiguračním okně nastaveno zrychlení a zpomalení na 10,0. Situace ① vystihuje posunutí o vzdálenost 1 000,0 a po signalizaci ukončeného děje pomocí výstupního parametru *Done* dojde k dalšímu posunutí o 500,0. Mezi oběma ději je krátká časová prodleva z důvodu doby odezvy. První časová osa vystihuje dobu, po kterou je povolen pohyb osy. Na druhé časové ose je zobrazen časový interval, ve kterém se provádí první úkon (posun 1000,0). Konec intervalu nastane v okamžiku dosažení příslušné vzdálenosti. [11]

Třetí časová osa vystihuje interval, během kterého je signalizováno ukončení úkonu. Časová osa *Exe\_2* se již týká druhého děje (posun o 500,0). Vyobrazený interval představuje dobu povolení příkazu. Interval na časové ose *Busy\_2* odpovídá době vykonávání úkonu a po dosažení cíle je stav signalizován příkazem *Done\_2*. Na předposlední časové ose (*Velocity Axis\_1*) je zobrazen průběh rychlosti, která nejprve narůstá s konstantním zrychlením, následuje úsek s konstantní rychlostí a poté zpomalování na nulu, které dosáhne v okamžiku dosažení vzdálenosti 1000,0. Po krátké prodlevě je situace zopakována pro následující děj s obdobnými parametry. Poslední časová osa na obr. 16 vystihuje změnu pozice v čase. [11]

Při vykonávání děje ② došlo k přerušení jiným úkonem. Přerušení je realizováno zakázáním děje 1 (*Move\_1*) pomocí příkazu *Abort\_1*, což dokumentuje čtvrtá časová osa. Ve stejném okamžiku, kdy nastane přerušení, dojde ke spuštění vykonávání druhého děje (*Move\_2*). Doby vykonávání tohoto děje zachycuje osa nazvaná *Busy\_2*. Po vykonání druhého úkonu již nedojde k navrácení k prvnímu ději, osa tedy nedosáhne stejné vzdálenosti jako v prvním případě. Kromě těchto výše popsanych parametrů obsahuje blok informační výstupy chyb, které mohou nastat. V bloku na obr. 16 jsou označeny názvy *Error*, *ErrorID* a *ErrorInfo*. Parametr *Error* pouze signalizuje, že během vykonávání úkonu došlo k chybě, jejíž příčinu lze zjistit z parametrů *ErrorID* a *ErrorInfo*. [11]

Mezi další základní bloky pro zajištění pohybu patří *MC\_MoveVelocity* (posun osy přednastavenou rychlostí), *MC\_MoveJog* (pohyb osy v režimu jog – pohyb pro účel testování a uvedení do provozu), *MC\_CommandTable* umožňuje nastavit posloupnost více různých pohybů, *MC\_ChangeDynamic* pro nastavování různé dynamiky pohybu atd. [11]

Vykonávání úkonů lze testovat na konkrétním pohonu programově v online režimu, kdy se v okně *watch table* vytvoří tabulka obsahující názvy konkrétních bloků a dle jejich specifikací nastavuje daný parametr (povolení/zakázání osy, velikost rychlosti, pozice aj.). Po nahrání programu do PLC lze hodnoty a stavy měnit a v reálném čase sledovat odezvu přímo na hřídeli motoru. [11]



Obr. 16: Blok `MC_MoveRelative`, průběhy vstupních a výstupních parametrů

## 8.2.2 Ovládání přes HMI

Projekt vytvořený za pomoci prostředí TIA Portal lze ovládat z dotykového panelu HMI. Stejně jako ostatní zařízení musí být i ovládací panel připojen k projektu a propojen se zařízeními, se kterými má komunikovat. Komunikace probíhá opět přes ethernetové připojení. Toto propojení musí být v projektu definováno ihned po vložení HMI do projektu (viz dále). Ovládací panel se připojí do projektu přes nabídku *Add new device* obsahující knihovnu *Controllers*, která obsahuje PLC řady *SIMATIC* a knihovnu *HMI*. V knihovně *HMI* se vybere konkrétní ovládací panel dle typového čísla. Následuje nadefinování komunikačního rozhraní a propojení s PLC. Poté se v okně představujícím dotykovou plochu ovládacího panelu nakreslí objekty související s automatizovaným procesem.

Nejčastěji se na obrazovce vykreslují různé ovládací prvky (tlačítka, přepínače) a zobrazovací prvky informující obsluhu o vykonávaných dějích. Lze však vkládat i další objekty informující o stavu média, počtu cyklů, aktuálním čase atd. Následuje definování událostí pro jednotlivé objekty a přiřazení k objektům nakreslených v okně bloků *Main*. Tlačítkům na dotykové ploše panelu se vybere událost, kterou jsou aktivována, popř. deaktivována. Každé tlačítko může reagovat na kliknutí, zmáčknutí, uvolnění. Následuje nastavení děje souvisejícího s daným tlačítkem a přiřazení ke konkrétnímu objektu (kontaktnímu cívkou, nebo tlačítku) v *Mainu* přes adresu tohoto objektu. Má-li tlačítko umístěné na ovládacím panelu reagovat jako mechanické tlačítko, přiřadí se mu děj *SetBit* nebo *ResetBit*. Tyto funkce nastavují digitální výstupy nabývající hodnot *logická 1*, nebo *logická 0*. Každé tlačítko má svůj název zobrazený na těle tlačítka a pro snadnější identifikaci je vhodné tlačítka mnemotechnicky pojmenovat (a použít anglické názvosloví). U objektů pro vizualizaci odehrávaného děje z nabídky *Basic objects* se místo události nastaví děj (animace) poté, co se daný objekt přiřadí k objektu z *Mainu* například cívkou. Pro cívkou bude animace spočívat ve změně barvy objektu pro stavy cívkou pod napětím a cívkou bez napětí. V podokně nazvaném HMI tags jsou zobrazeny veškeré prvky použité v projektu pro zařízení HMI. Kromě názvu obsahuje tabulka též datový typ, typ propojení, jméno PLC automatu a akviziční cyklus související se získáním výsledku procesu primárně nastavený na 1 s (lze jej nastavit v rozmezí 100 ms až 1 hodina). Po vytvoření výše popsáných objektů se projekt nahrává zvlášť do PLC modulu a do HMI panelu. Proces nahrávání do ovládacího panelu lze sledovat přímo na jeho zobrazovací ploše. Před samotným spuštěním procesu říditelného z ovládacího panelu lze nastavit časové údaje (datum a čas), jas displeje, aktivaci akustického tónu při dotyku na ovládací objekt a další.

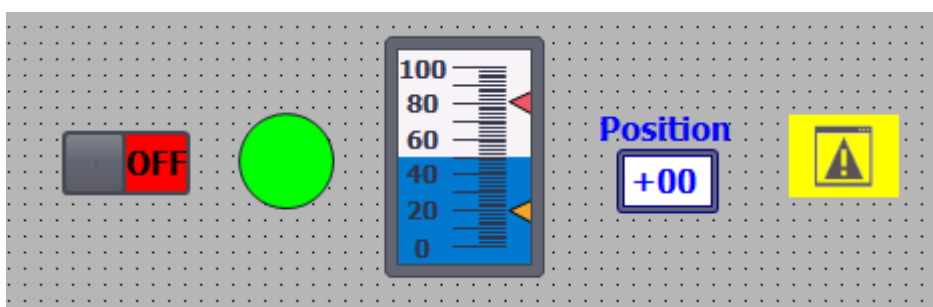
Vývojové prostředí TIA Portal nabízí funkci *RT Simulator* pro simulaci ovládání procesu skrze ovládacího panelu přímo z počítače. Tuto funkci lze využít v případě, že vytvořený projekt nahráváme pouze do imaginárního zařízení (v nabídce *Download to device* → *Software*).

Správnou funkci ovládání procesu z ovládacího panelu lze sledovat z konkrétních objektů na obrazovce, popř. z LED umístěných na PLC modulu spjatých s každým digitálním vstupem (tlačítkem vytvořeným v *Mainu*) a digitálním výstupem (simulujícím cívkou relé/stykače) byly-li tyto prvky v projektu použity.

Objekty zobrazitelné na dotykovém panelu však nemusí být nijak spjaty s reálnými objekty, může se jednat pouze o podpůrné ovládací prvky k nimž lze přiřadit mnoho různých funkcí z nabídky možností pro jednotlivé ovládací a zobrazovací prvky, které lze vyobrazit v editoru na dotykovou plochu (*Screen*). Dotykových ploch je možné vytvořit více, nevede-li se celý ovládací a zobrazovací soubor objektů na plochu dotykového displeje panelu. Přepínání mezi jednotlivými *screeny* lze realizovat objektem tlačítka s přiřazenou funkcí *ActivateScreen* a zadáním názvu té obrazovky, na kterou má tlačítko po jeho stisknutí přesměrovat.

Tlačítkům nacházejícím se pod dotykovou plochou panelu lze nadefinovat libovolnou funkci, například přepínání obrazovek, změnu jasu, volbu jazyka textu atd.

Zobrazovací prvky můžeme rozdělit do několika skupin, podle významu a způsobu vyobrazování informací. Dvoustavové zobrazovací prvky nabývají pouze dvou úrovní výstupů: zapnuto/vypnuto, přičemž konkrétní stav určuje nejčastěji barva objektu. Kromě různé barvy objektu pro jednotlivé úrovně může být stav prezentován blikáním objektu, či textu spjatým s objektem v jeho blízkosti. Druhé skupiny obsahují prvky se stupnicí, na které se zobrazuje údaj vyjádřitelný procentně od 0 do 100 pomocí barvy stupnice (například zobrazení stavu hladiny kapaliny v nádrži). Poslední skupina obsahuje prvky, které zobrazují číselný údaj. Patří sem čítače, časovače, prvky nahrazující měřicí přístroj, například digitální voltmetry, tlakoměr aj. Typy vybraných zobrazovacích prvků a jejich podobu pro konkrétní využití přináší obr. 17.



Obr. 17: Typy zobrazovacích prvků

Na obr. 17 se zleva nachází ikona přepínače, na které je dynamický nápis *OFF* pro stav vypnuto a *ON* pro stav zapnuto, navíc lze nastavit barvu pozadí pro daný stav, například červenou, jako je zobrazeno. Zelené kolečko představuje signalizační kontrolku, stupnice stav naplnění nádrže pro rozsah 0 – 100. Taktéž prvek *I/O Field* lze použít pro zjištění stavu určitého parametru (tlak, teplota atd.). Ikona vykřičníku na žlutém pozadí může symbolizovat například upozornění na nezakryté rotující části v pohybu.

## 9 Popis výsledného projektu

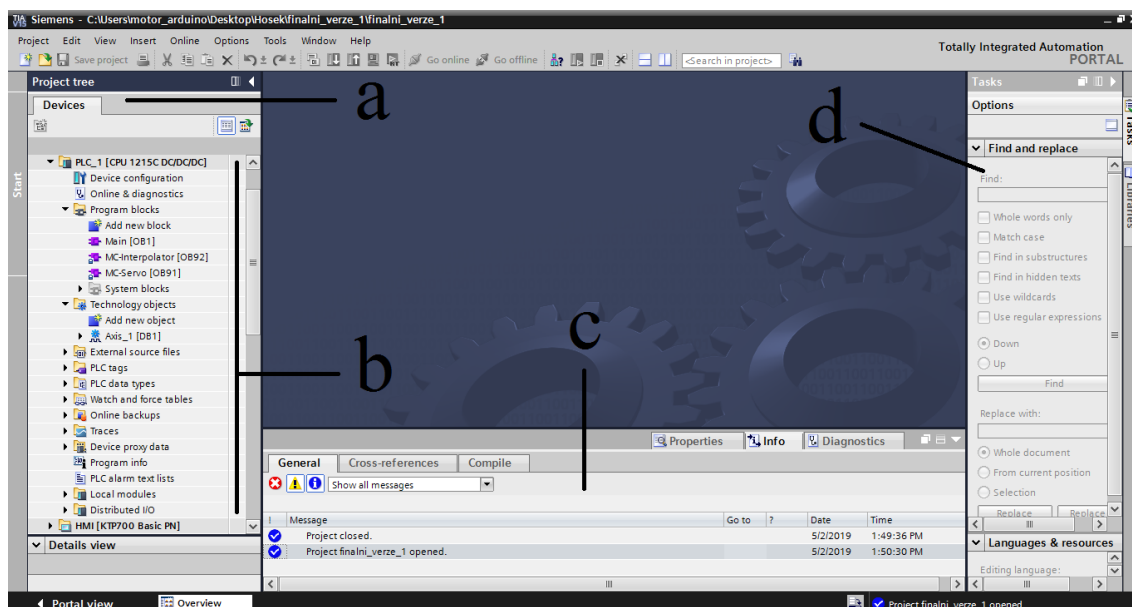
### Nastavení pohonu:

Výsledný projekt obsahuje všechny výše popsané komponenty. Nejprve byl pomocí programu V-Assistent vytvořen soubor parametrů týkajících se zvoleného motoru, především vybraných technických a elektrických skutečností (maximální proud, napětí, otáčky, jmenovitý proud, napětí, otáčky) a také parametry regulátoru pro zpětnovazební řízení motoru. Tyto parametry lze získat pomocí automatického ladění (Commission -> optimize drive -> one button auto tuning), kdy program automaticky otestuje daný pohon a výsledkem je tabulka vytvořených parametrů, které se následně nahrají do paměti měniče. Automaticky vytvořené parametry, zvláště ty, které se týkají regulace pohonu nejsou nikterak přesné. Pro tento projekt jsem se ovšem rozhodl takto získané parametry použít, neboť se jedná o demonstrativní úlohu, která má za cíl vytvořit funkční proces zapsaný v paměti PLC modulu a monitorovaný přes ovládací panel, kdy ne zcela přesně dodržované hodnoty otáček nehrají příliš velkou roli.

### Vytvoření projektu:

Po nahrání parametrů do paměti měniče přes USB komunikační rozhraní bylo možné přistoupit k tvoření samotného projektu v programu TIA Portal V15. Prvním krokem po spuštění programu je vytvoření nového projektu, kdy se zadává název projektu a místo uložení projektu. Též lze zapsat autora projektu a poznámku se stručným popisem tvořeného projektu (jaká zařízení obsahuje, k čemu bude výsledný program sloužit atd.) Následují dva odlišné způsoby vložení konkrétního zařízení do projektu. Buďto se pokračuje na úvodní obrazovce programu kliknutím na ikonu *First step* a vybráním z nabídky *Devices & networks* se přidá nové zařízení. Po vybrání konkrétního zařízení se postupuje dále v tvorbě PLC programu a dalších částech. Já jsem se rozhodl pro druhý způsob práce v tomto vývojovém prostředí, kdy se po vytvoření nového projektu přepne pomocí ikony *Project view* do nového okna programu, které obsahuje mnoho dalších podoken, které souvisí s konkrétním vykonávaným krokem během tvoření celého projektu. K jednotlivým podoknům a nabídkám lze přistupovat z hlavního podokna v levé části obrazovky obsahujícího veškerá použitá zařízení, bloky, nastavení a další nezbytné části programu. Obr. 18 vystihuje nabídku pro zařízení PLC použité v mém projektu. Pro vložení modulu PLC, které je v popisovaném projektu centrálním zařízením propojující ostatní zařízení, se po kliknutí na ikonu *Add new device* objeví nabídka knihoven několika typů zařízení, přičemž nyní nás zajímá zařízení z knihovny nazvané *Controllers*. Zde se nachází několik řad modulů SIMATICS. Použitý modul je z řady S7-1200, kde se vybere konkrétní typ, kterých je také celá řada. Pro zamezení zaměnitelnosti podobných typů je každý modul vybaven typovým označením, které je unikátní pro každý typ modulu a toto označení musí v projektu souhlasit s číslem na modulu. Následně se v projektu objeví okno *Devices & networks* obsahující tři karty. V kartě *Device view* se nachází konkrétní zařízení (pokud je v projektu více zařízení, vybere se z nabídky to, které má být zobrazeno). Další dvě karty slouží pro specifikaci komunikačního propojení zařízení. Nyní se v kartě *Network view* nachází pouze modul PLC. Současně se v pravé části obrazovky objeví katalog obsahující knihovny všech dostupných zařízení z nabídky výrobce. Výkonový měnič použitý v projektu lze přidat z nabídky v záložce *Drivers & starters*, pokud tato záložka konkrétní měnič obsahuje, nebo lze měnič V 90 PN přidat ze záložky *ostatní zařízení* následujícím způsobem: *Other field devices -> Profinet IO -> Drivers -> Siemens AG -> Sinamics*. Následně se objeví v kartě *Network view* a uživatel jej propojí s PLC modulem (*assign to new IO controller*).

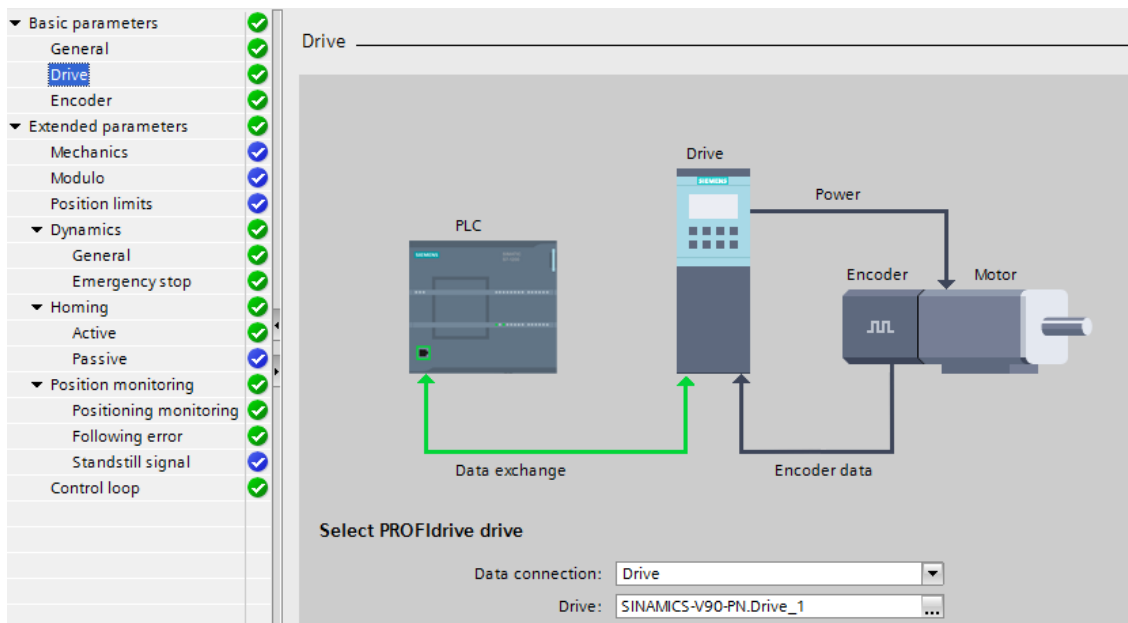
Třetí karta *Topology view* zobrazuje skutečné komunikační propojení (pokud má zařízení více komunikačních portů, musí být fyzické propojení totožné s propojením vytvořeným v projektu – proto je každý port vybaven vlastním označením viditelným jak v projektu, tak na modulu). Tato karta je však nepovinná a pro správnou funkci komunikace zde není třeba nic nastavovat. Nezbytné je však v první popisované kartě přiřadit měniči komunikační telegram, který se nachází opět v pravé části obrazovky v katalogu obsahujícím několik telegramů. Vhodné a postačující je vybrat telegram nazvaný *Standard telegram 3, PZD 5/9*, který vložíme na nový řádek tabulky měniče umístěné v pravé části karty *Device view*. Nyní se v levém hlavním podokně vybere z nabídky měniče podokno *Device configuration*, v němž je nutné pro správnou funkci komunikace měniče s PLC modulem nastavit shodnou IP adresu a název s údaji vloženými do paměti měniče v programu V-Assistant. Dále se v záložce *Modul parameters* vybere z nabídky pro aktivaci diagnostiky *Standard alarms*.



Obr. 18: Hlavní obrazovka programu TIA Portal; a) hlavní podokno, b) rozvětvená nabídka pro modul PLC, c) podokno s informacemi o stavu programu (stav přeložení kódu, nahrání programu do PLC, chyby, varování aj.), d) místo pro podokno s knihovnami a nastaveními (nyní neaktivované)

### Vytvoření a nastavení technologického objektu – osy:

Nyní se do projektu vloží motor řízený měničem přes nabídku pro PLC modul nazvanou *technologické objekty (Technology objects)*. Objekt je obecně nazván *TO\_PositioningAxis*, neboť v této části projektu nás více než motor samotný zajímá jeho hřídel (osa), která přenáší rotační pohyb na mechanickou část zařízení vykonávající pohyb (např. ozubené soukolí, pásový dopravník) a motor je řízen dle informací o rychlosti otáčení hřídele a poloze natočení hřídele vůči nulové pozici definované resetovacím bitem v inkrementálním čidle. Tento objekt lze jakkoliv pojmenovat, přednastavené jméno je *Axis\_1*. Poté se v nastavení osy specifikuje propojení motoru, enkodéru, měniče a PLC modulu. Také se vybere z nabídky druh měniče, který je v tomto případě PROFIdrive (komunikace s enkodérem motoru přes profinet) a konkrétní zařízení měniče (v našem případě do projektu vložený V\_90). Dále se nastaví propojení enkodéru a jeho umístění (*encoder on PROFINET/PROFIBUS – použito, encoder on driver*). Ostatní nastavení jsou přednastavena a většinou není třeba je měnit, nebo se nastaví až podle konkrétní situace (např. nastavení rychlosti zpomalení při nouzovém zastavení pohonu, vypnutí sledování chyb, nastavení limitů pozice osy). Obr. 19 zachycuje nastavení propojení měniče s motorem, enkodérem a PLC jednotkou.



Obr. 19: Výřez části okna pro nastavení technologického objektu; v levé části nabídka prvků, které lze nastavit a stav aktuálního nastavení (zelený symbol v pravé části příslušného řádku – splněno, modrý symbol – volitelné, splněno, červený symbol (na obrázku není zchycen) – chybějící nebo nekompletní nastavení)

#### Použité bloky, nastavení a zapojení:

Po nastavení technologického objektu je pohon tvořený motorem a měničem připraven k použití. Jelikož je k řízení pohonu určena PLC jednotka, vytvoří se pro toto zařízení programový kód. Nejsnadnější cestou je použití předdefinovaných bloků z knihovny. V levém podokně se pomocí záložky *Program blocks* otevře podokno *Main*, ve kterém se pomocí jazyka LAD dá vytvořit schéma obsahující bloky napsané jazykem FBD. Informace o příslušných blocích jsou dohledatelné v manuálu výrobce nebo v nápovědě spustitelné ve vývojovém prostředí klávesou F1. Datové bloky určené pro řízení pohonu jsou dostupné v podokně v pravé části obrazovky v záložce *Technology* pod nabídkou *Motion Control*. Každý blok se pro zpřehlednění celkového schéma vkládá do samostatného oddílu nazvaného *Network x* ( $x$  = pořadová číslice). Pokud však chceme v jednom programovém bloku *Main* vytvořit více na sobě nezávislých procesů, může být celý proces vytvořen v jediném oddílu, nebo rozložen do více oddílů podle druhů bloků. Při překladu se všechny části programu spojí do jednoho celku.

Pro spuštění (aktivaci pohonu) je nejdůležitější blok *MC\_Power* jehož jediným úkolem je povolení dané osy. V levé části bloku se nacházejí vstupy označené *EN*, *Axis*, *Enable*, *StartMode* a *StopMode*. Nutné je nastavit první tři vstupy. Vstup *EN* je přímo propojen se svislou čarou v levé části podokna představující napájecí vodič. Druhý vstup slouží k propojení bloku s konkrétním technologickým objektem, v našem případě s motorem označeným *Axis\_1*. Mezi napájecí vodič a třetí vstup je třeba vložit spínací kontakt, který může představovat tlačítko na dotykovém panelu, mechanické tlačítko na ovládacím pultu, nebo spínací kontakt cívky aktivované povely z programu tak, jak jsem to realizoval ve své práci. V pravé části bloku se nacházejí tři výstupy: *ENO*, *Status* a *Error*. První výstup uzavírá elektrický okruh (vývod je ukončen vodorovnou čarou, ke které se za blokem již nic nepřipojuje). Druhý a třetí výstup slouží k vysílání informací o stavu technologického objektu.

Druhým blokem vloženým do projektu je *MC\_Reset*, který umožňuje resetovat pohon v případě vyskytnutí chyby během spuštěného procesu. Také k tomuto bloku se přiřadí konkrétní pohon z technologických objektů vložených do projektu, v našem případě *Axis\_1*.



Reset pohonu se realizuje spojením levé svislé čáry se vstupem *Execute* přes příslušné tlačítko, které jsem nazval *RESET*.

Další bloky jsou již volitelné a slouží k vytvoření povelů pro pohon. Z nabídky jsem vybral několik bloků umožňujících otáčení hřídele motoru na obě strany, nastavení rychlosti a vzdálenosti, které má fiktivní objekt dosáhnout (v nastavení pohonu je nastaven převod dosažení vzdálenosti 10 mm na 1 otáčku hřídele). Blok *MC\_MoveJog* má kromě vstupů *EN* a *Axis* také vstupy pro zadání rychlosti a směru točení hřídele nazvané *Velocity*, *JogForward* a *JogBackward*. *Velocity* je reálná proměnná, do které lze přímo zapsat hodnotu otáček (ot/min), nebo tento vstup propojit s proměnnou vytvořenou v programu a její hodnotu měnit dle potřeby za chodu děje například z ovládacího dotykového panelu. Výstupy má tento blok tři, kromě výstupů, které jsou shodné s výstupy pro veškeré bloky použitelnými pro technologický objekt *Axis* (výstupy *ENO* a *Error*) je zde i výstup nabývající stavů *TRUE* nebo *FALSE* podle toho, zda fyzický objekt dosáhl požadované rychlosti, či nikoliv. Výstup je tedy proměnná typu *boolean*.

Blok *MC\_MoveVelocity* dovoluje uživateli definovat rychlost, kterou se má hřídel otáčet. Nepovinně nastavitelným vstupem je vstup nazvaný *Current*, který je typu *boolean* a slouží k řízení otáček sledováním proudu. Pokud je tento vstup nastaven na hodnotu *False*, rychlost otáčení osy se řídí z hodnot zadaných do parametrů *Velocity* (rychlost) a *Direction* (směr otáčení). Při vstupu nastaveném na hodnotu *True* se hodnoty rychlost a směr neuvažují a osa se řídí udržováním odpovídající hodnoty proudu fáze. V režimu *True*, je aktivován i výstup *InVelocity*, který je též typu *boolean* a při dosažení nastavené rychlosti otáčení vrátí hodnotu *True*. Vstup *Current* jsem nepoužil, rychlost otáčení osy nastavuji přímo z ovládacího panelu zadáváním požadované hodnoty rychlosti otáčení (v ot/min) do vstupu *Velocity*, který je typu *real* (lze do něj nastavovat reálná čísla, ale z dotykového panelu jsem povolil zadávat pouze celá čísla kladná pro jeden směr otáčení a záporná pro druhý směr otáčení). Tento vstup lze nastavit i na hodnotu 0 ot/min. Vstup *Direction* může nabývat hodnot 0, 1 a 2, přičemž hodnota 1 odpovídá jednomu směru otáčení (bez vlivu znaménka hodnoty vstupu *Velocity*), hodnota 2 obdobně pro druhý směr otáčení a hodnota 0, která je v bloku přednastavena a byla použita, ponechává směr otáčení zadaný znaménkem u hodnoty vstupu *Velocity*. Vstupy *EN*, *Axis* jsou nastaveny totožně, jako u předešlého bloku osy. Vstup povolující vykonávání otáčení osy nazvaný *Execute* je spínán kontaktem příslušného ovládacího relé. K aktivaci tohoto bloku postačuje mžikové spojení uvedeného vstupu s levou svislou linií, lze tedy místo spínacího kontaktu použít i tlačítko, ale k ukončení děje je nutné nastavit vstup *Velocity* na hodnotu 0, nebo zakázat (odpojením vstupu *Enable*) blok *MC\_Power*.

Poslední blokem použitým v mém projektu pro ovládání osy motoru je blok *MC\_MoveRelative*, do kterého vstupují parametry nabývající hodnot 0 a 1 (vstupy označené *EN* a *Execute*), vstup *Axis* přiřazující konkrétní osu a dva vstupy, které jsou proměnné typu *real* nazvané *Distance* a *Velocity*. Do těchto proměnných lze zapsat přímo rychlost v ot/min a vzdálenost, které má pohybujiící se část pohonu dosáhnout. Druhým způsobem je stejně jako u předešlých bloků možnost vytvoření globální reálné proměnné a její připojení ke konkrétnímu vstupu a následné zapsání hodnoty do proměnné z programu, nebo ovládacího zařízení. Názvy vytvořených proměnných, názvy kontaktů a cívek jsou automaticky zapsány do tabulky v záložce *PLC tags*. Příklad části takovéto tabulky je zachycen na obr. 20. Z tabulky je patrné, že kromě názvu proměnné (*tagu*) se zde nachází též její adresa, datový typ a další vlastnosti. Názvy proměnných je vhodné tvořit na základě mnemotechnických názvů pro snadnější orientaci při přiřazování a zadávání hodnot do proměnných, opět se doporučuje užít anglického názvosloví pro univerzálnost a srozumitelnost pro všechny potenciální uživatele. Názvy proměnných lze v tabulce přejmenovávat, nastavovat jejich adresu, nebo je úplně smazat.

Při mazání proměnných by si měl uživatel být vědom toho, že danou proměnnou v projektu skutečně nepotřebuje a mazat jen ty proměnné, které sám vytvořil. Projekt kromě vlastnoručně vytvořených proměnných obsahuje i takové, které jsou v projektu vytvořené automaticky (například na obr. 20 zachycené *tagy* označené č. 1 a 2) a po jejich odstranění je projekt nespustitelný. Následné vrácení takových proměnných do projektu je značně komplikované. Totéž platí i pro *tagy* umístěné v tabulce pro zařízení HMI (obr. 21), kdy navíc hrozí, že uživatel záměrně odstraní *tag* z tabulky *PLC tags*, ale zapomene totožný *tag* smazat i z tabulky *HMI tags*, je-li zde také použit a při kompilaci (resp. při kontrole syntaxe) nedojde k povolení nahrát projekt do paměti zařízení HMI. Vhodné je tedy během vytváření projektu pravidelně kontrolovat syntaxi a ukládat jednotlivé etapy, aby bylo v případě problému snadné chybu vyhledat a odstranit.

	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Axis_1_Drive_IN	Default tag table	*PD_TEL3_IN*	%I68.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Axis_1_Drive_OUT	Default tag table	*PD_TEL3_OUT*	%Q68.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	STOP	Default tag table	Bool	%M0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	START	Default tag table	Bool	%M0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	main_coil	Default tag table	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	STOP_PLC	Default tag table	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	START_PLC	Default tag table	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Obr. 20: Výřez podokna s PLC tagy

	Name	Tag table	Data type	Connection	PLC name	PLC tag	Address	Access mode	Acquisition ...
	START	Default tag table	Bool	HMI_Connection_1	PLC_1	START		<symbolic access>	100 ms
	start_button	Default tag table	Bool	HMI_Connection_1	PLC_1	start_button		<symbolic access>	100 ms
	STOP	Default tag table	Bool	HMI_Connection_1	PLC_1	STOP		<symbolic access>	100 ms
	stop_button	Default tag table	Bool	HMI_Connection_1	PLC_1	stop_button		<symbolic access>	100 ms
	support_button	Default tag table	Bool	HMI_Connection_1	PLC_1	support_button		<symbolic access>	100 ms
	Tag_ScreenNumber	Default tag table	UInt	<Internal tag>		<Undefined>			1 s
	VELOCITY	Default tag table	Bool	HMI_Connection_1	PLC_1	VELOCITY		<symbolic access>	100 ms

Obr. 21: Výřez podokna s HMI tagy

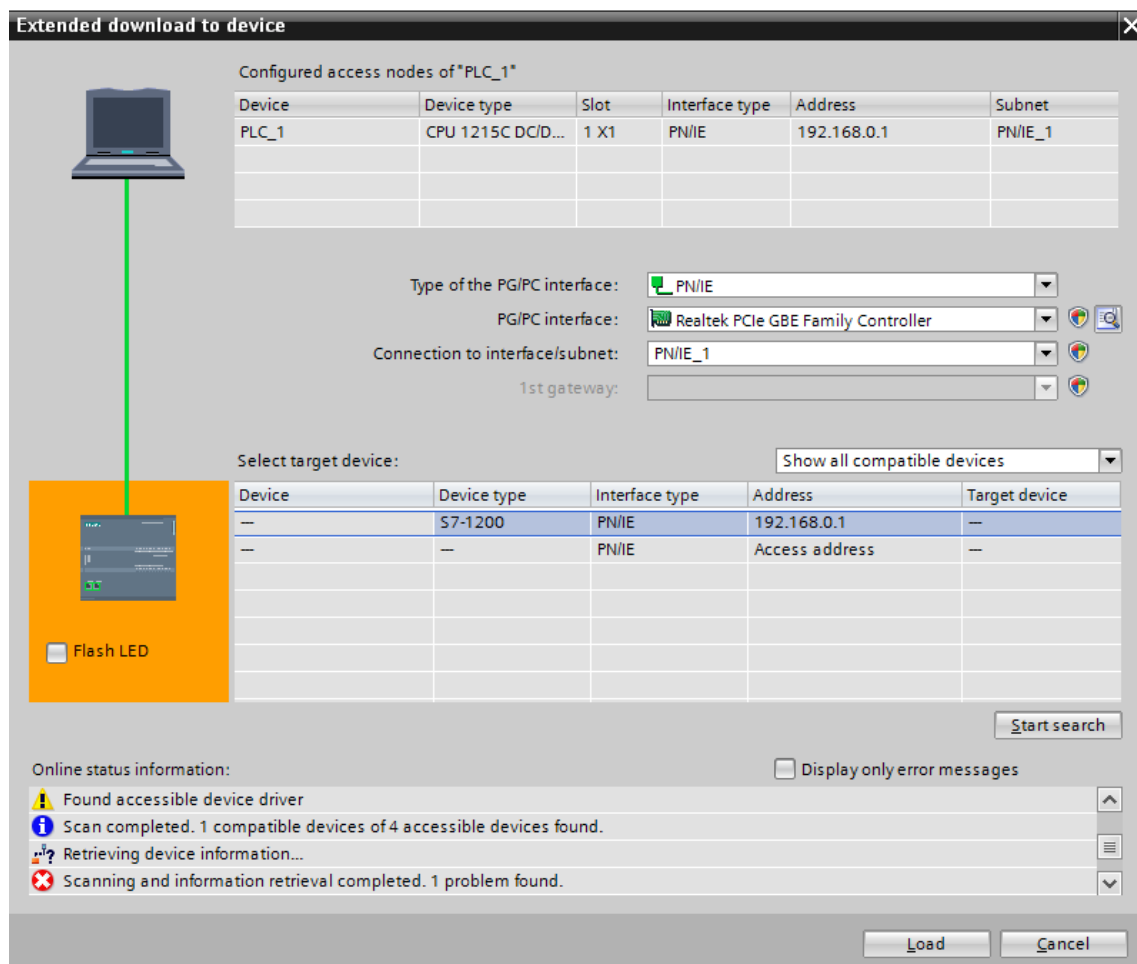
Blok *MC\_ReadParam* dovoluje uživateli sledovat hodnoty proměnných spjatých s řízenou osou. Vstupní proměnná bloku nazvaná *Parameter* je volitelného datového typu, uživatel musí vybrat dovolený datový typ z nabídky na těle bloku. Podle zvoleného datového typu lze k tomuto vstupu z nabídky přiřadit parametr osy, který je třeba monitorovat. Stav dvoustavové proměnné informující o stavu pohonu lze zobrazit na ploše dotykového panelu změnou barvy přiřazeného objektu. Aktuální hodnoty parametrů typu *real* lze na panelu zobrazovat pomocí objektu *I/O field*, na kterém mohou být zobrazována celá čísla v rozmezí od -999 999 do +999 999, nebo reálná čísla. Tento objekt umožňuje zvolení rozsahu a charakteru čísel, které mají být zobrazovány (čísla pouze celá/reálná, pouze kladná/se znaménkem). Má-li být objekt použit pro zobrazování hodnot z bloku motoru, nastaví se jeho vstupně-výstupní parametr na výstup. Pro přečtení hodnoty zvoleného parametru musí být výstup *Valid* (typu *boolean*) bloku *MC\_ReadParam* nastaven na hodnotu *True*. Vstupně-výstupní parametr *Value* je shodného datového typu jako *Parameter* a ukazuje na místo v paměti, kde je sledovaná hodnota zapsána. Informativní výstupy označené *ErrorID* a *ErrorInfo* jsou proměnné typu *word* a vracejí továrně přednastavené hodnoty informující uživatele o případných chybách typu *nesprávná délka parametru*, nebo *chybná hodnota* dohledatelných v tabulce manuálu. Pro činnost bloku je nezbytné použít a nastavit vstupy označené: *EN*, *Enable*, *Parameter*, *Value*. Tento blok jsem použil k zobrazení aktuální hodnoty otáček a pozice pomyslného objektu, jenž se posune o 10 mm za jednu otáčku osy.

Některé výše popsané bloky jsem ve svém projektu použil nejen k demonstraci jejich funkce, k čemuž slouží obrazovka root screen na dotykovém panelu, ale také k vytvoření automatického cyklu, který uživatel pouze spustí a děj se cyklicky vykonává bez zásahu obsluhy, která pouze sleduje aktuální hodnoty rychlosti otáčení a pozici pohybujícího se objektu. Obsluze je však dovoleno tento cyklus kdykoliv přerušit a opětovně spustit. K pozastavení děje dojde také při rozepnutí některého z pomyslných koncových spínačů (v projektu realizováno fyzickým přepínačem na PLC modulu) představujících nepovolený stav, či zásah člověka. Tento proces lze přirovnat k dopravnímu pásu s přídržným mechanismem (jinak by se pás po vyložení zboží nemusel vracet do výchozí polohy), který se pohybuje definovanou rychlostí a převáží objekt (zboží) z jednoho definovaného místa na druhé. Na každém koncovém místě se dopravní pás zastaví na přednastavenou dobu (v projektu realizováno blokem *TON* – časové relé se zpožděním přitahu) pro potřeby nakládky/vykládky zboží. Spínací logika a bloky vytvořené v hlavním programovém bloku (*Main*) jsou přidány do přílohy na konec této diplomové práce.

### Kontrola syntaxe, nahrání projektu do paměti PLC modulu:

Po vytvoření celého zápisu v okně *Main* následuje kontrola a přeložení projektu do kódu, který se následně nahraje do paměti PLC modulu. Před nahráním projektu do modulu je nezbytné nastavit komunikační cestu mezi modulem a počítačem, které zprostředkovává ethernetové připojení. Po kliknutí na ikonu *Download to PLC* nacházející se na liště v horní části obrazovky se objeví okno s možnostmi připojení, kde je při prvním připojování zařízení nezbytné nastavit typ komunikačního rozhraní a adaptér počítače tak, jak to zachycuje obr. 22. Po nastavení správné komunikační cesty se vyhledají všechna dostupná zařízení, v tomto případě pouze PLC modul, protože HMI panel je zařízení, do kterého daný typ kódu určený pro PLC nahrát nelze. Po nalezení PLC modulu se objeví řádek s informacemi o tomto zařízení (název, typ zařízení, rozhraní, IP a MAC adresa) a po kliknutí na tento řádek lze program do zařízení nahrát.

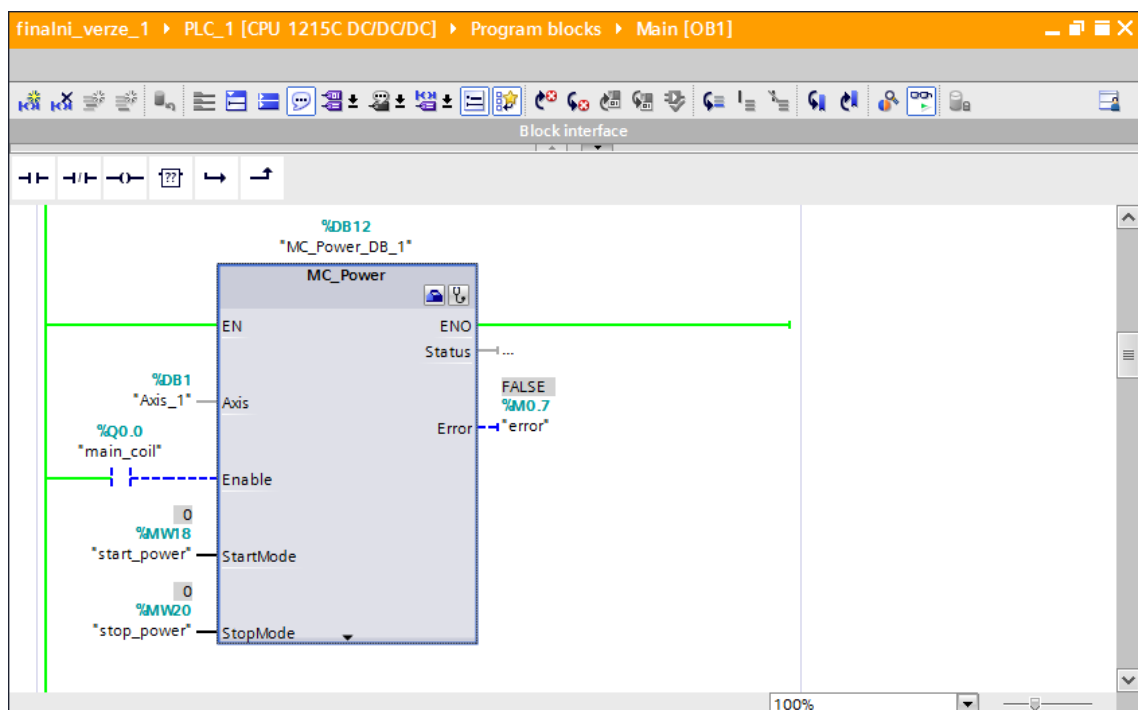
Pozn.: podokno nazvané *Online status information* informuje uživatele o probíhajících akcích. Na obr. 22 je v tomto podokně patrná informace s nalezeným nedostatkem, který je odstraněn po kliknutí na ikonu *Load*, kdy se v programu objeví samostatné okno vyzývající uživatele k přiřazení shodné IP adresy zařízení do ovladače rozhraní. Poté je programový kód nahrán do zařízení a lze jej spustit.



Obr. 22: Detekce a připojení zařízení

### Online sledování dějů:

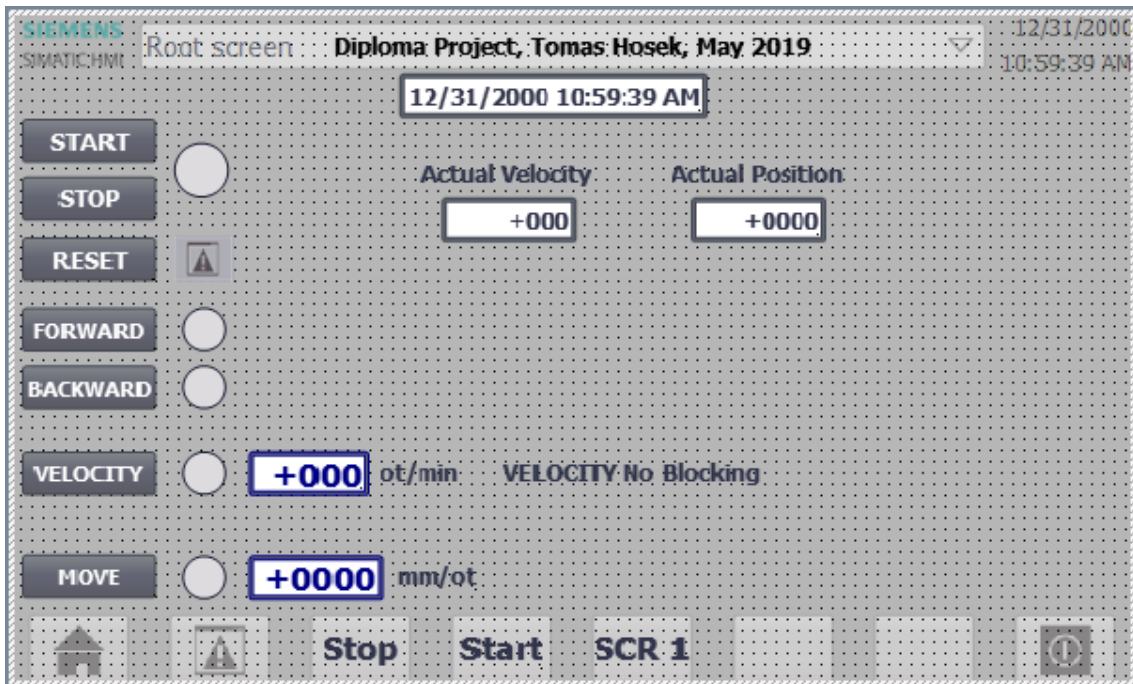
Nespornou výhodou vývojového prostředí TIA Portal je mód online sledování dějů probíhajících v zařízení, kdy zde můžeme sledovat děje a vyhodnocovat jejich funkci. Po přepnutí do online módu (ikona *Go online*) se horní část okna *Main* žlutě podbarví. Po kliknutí na ikonu *Monitoring on/off* se linie představující elektrické spoje zbarví do zelena v případě, že jsou propojené s napájením. Tímto lze sledovat, které části jsou pod napětím a ověřovat tak správnou funkci spínačů, cívek a dalších prvků (viz obr. 23). Především u bloků pro pohon lze sledovat, zdali je příslušný blok povolen (celá vodorovná čára vstupující do bloku vstupem *EN* a vystupující výstupem *ENO* je zbarvena zeleně) a jestli je blok aktivován (vstup *Execute* též musí být podbarven zeleně). Pozn.: Mezi levou svislou linií (pomyslný napájecí vodič) a vstupem bloku *EN* by neměl být umístěn jakýkoliv spínací, nebo rozpínací prvek. Pro aktivaci jednotlivých bloků je primárně určen vstup *Enable*, popř. *Execute*. Při nedodržení tohoto pravidla může dojít k nesprávné činnosti realizovaného zapojení.



Obr. 23: Podokno s programovým blokem *Main* v režimu online monitorování

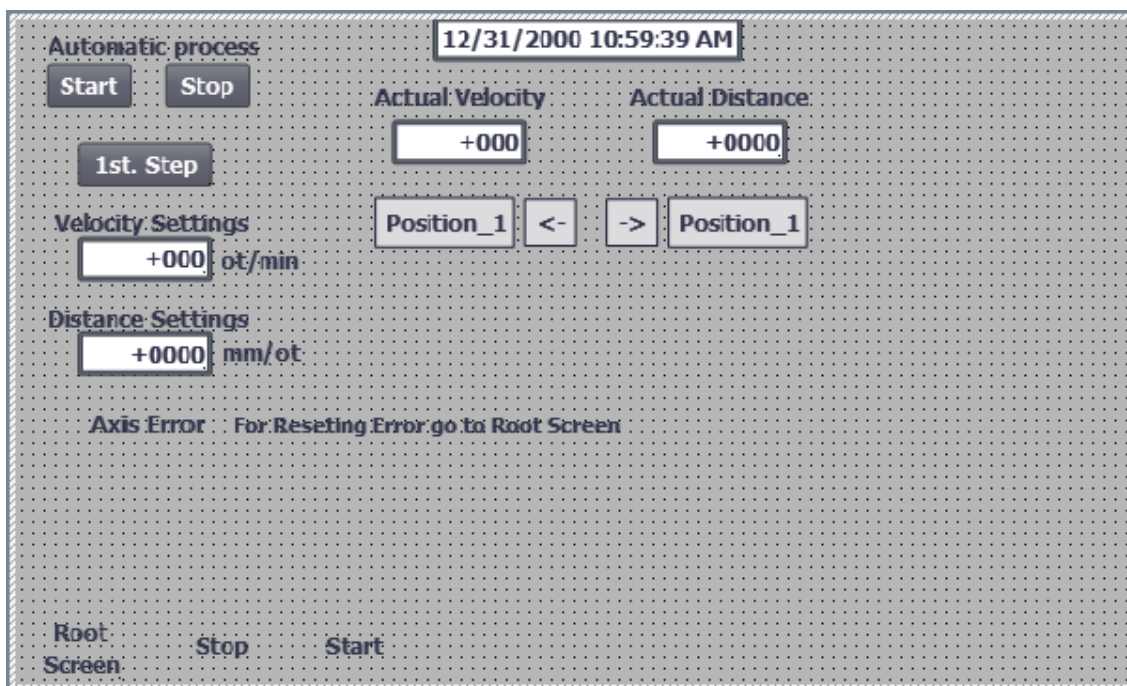
### Dotykový panel, nastavení, ovládání:

V poslední části projektu jsem v prostředí TIA Portal vytvořil vizualizaci dějů a ovládání procesů pro HMI zařízení. Na tomto dotykovém panelu jsem realizoval dvě samostatné obrazovky nazvané *Root screen* a *Screen 1*, první pro demonstraci ovládání a zadávání rychlosti, směru otáčení osy přímo z dotykového panelu, druhou pro sledování automatického procesu. Obr. 24 přináší pohled na úvodní obrazovku nazvanou *Root screen*, obr. 25 na obrazovku *Screen 1*.



Obr. 24: Úvodní obrazovka dotykového panelu

Ikony představující tlačítka nazvaná *START* a *STOP* slouží k povolení osy prostřednictvím hlavní cívky a příslušných bloků. Třetí tlačítko umožňuje restartovat osu v případě, že dojde k reakci ochrany například při prudké změně otáček zadáním nepřiměřené hodnoty, nebo zadáním příkazu na změnu směru otáčení bez zastavení osy. Na vyskytnutý problém upozorňuje symbol vedle tlačítka *RESET*, který se zbarví do červena. Následující čtveřice tlačítek je navzájem blokována a slouží k demonstraci funkcí jednotlivých bloků. Tlačítko *Velocity* pro blok ovládající rychlost otáčení vyžaduje nastavení požadované hodnoty a pro zastavení otáčení musí uživatel zadat nulové otáčky a potvrdit opětovným zmáčknutím tlačítka. Všechna čtyři tlačítka představují mechanické spínače s aretací polohy (aretační poloha je signalizována zeleným zbarvením tlačítka). Blok aktivovaný tlačítkem *Move* vyžaduje nastavení rychlosti a vzdálenosti, které má dosáhnout smyšlený objekt umístěný na posuvném mechanismu. Čísla v rámečcích pod nápisy *Actual Velocity* a *Actual Position* představují aktuální rychlost a aktuální absolutní polohu objektu. Obě hodnoty je nutné brát s rezervou, hodnota rychlosti je spíše informativního rázu, nepřesnosti jsou způsobené použitými parametry regulace. Vybraná tlačítka mimo dotykovou plochu jsou nastavena pro stejný účel, jako první dvě popsané ikony a pro přepnutí do druhé obrazovky.



Obr. 25: Druhá obrazovka dotykového panelu týkající se automatického procesu

Tlačítka *Start* a *Stop* na obr. 25 spouštějí a zastavují automatický proces, jehož stav zobrazují ikony (a jejich zbarvení) pod okny s aktuální rychlostí otáčení hřídele a pozicí objektu. Tlačítko *1st. Step* je podpůrné tlačítko pro prvotní spuštění děje, poté je již neaktivní. Ikony *Position\_1*, *Position\_2* jsou zbarvené modře, je-li objekt v koncové pozici, ikony se šipkami jsou zelené v případě pohybu objektu daným směrem. Prvním třem tlačítkům umístěným pod dotykovou obrazovkou (mimo snímek) na ovládacím panelu je přiřazena funkce pro přepnutí na úvodní obrazovku, zastavení děje a spuštění děje. Popisky pro tato tlačítka jsou viditelná v levém dolním rohu snímku.

V příloze této práce je reléové schéma vytvořené v prostředí TIA Portal. Pozn.: Výše v textu jsem psal, že mezi svislou linií a vstupem *EN* se nekládají žádné spínací, či rozpínací prvky. Přesto se u bloku *MC\_Power* nachází rozpínací kontakt relé a jeho přítomnost je nutná k bezproblémovému chodu dvou nezávislých dějů určených pro tentýž objekt (*Axis\_1*), protože jeden technologický objekt by v běžném užití mohl mít přiřazen pouze jeden blok *MC\_Power*. Pro demonstraci různých naprogramovatelných stavů pro jeden objekt jsem byl nucen použít zmíněný blok dvakrát.

Kromě zmiňovaných bloků jsou zde i bloky matematických funkcí pro násobení a dělení použité pro přepočet otáček, aby na panelu zadaná hodnota odpovídala otáčkám za minutu a pro další podpůrné výpočty (rychlost je zadána najednou pro oba směry, pro druhý směr je nutné parametr násobit číslem -1). Bloky reprezentující časová relé se zpožděním přitahu (*TON*) zase slouží pro setrvání v koncové pozici po definovanou dobu.

## 10 Závěr

Během práce na tomto diplomovém projektu jsem se seznámil s vývojovým prostředím TIA Portal Step 7, programem V-Assistant pro nastavení a zapsání parametrů do paměti měniče a s jednotlivými komponenty sestavy od firmy SIEMENS. Postupně jsem oživoval a do projektu přidával použitá zařízení, kdy jsem nejprve ověřil funkci PLC modulu vytvořením jednoduchého kódu v hlavním programovém bloku, poté přidal měnič s motorem, následně přepínací komunikační modul pro propojení více zařízení, a nakonec ovládací panel HMI.

Byť začátky práce ve vývojovém prostředí a porozumění funkcí jednotlivých částí programu i zařízení nebyly nijak jednoduché, podařilo se mi realizovat úlohu, kdy pohon ovládám z dotykové obrazovky HMI zařízení a současně na stejné ploše sleduji aktuální stav pohonu. Jelikož jsem se nijak podrobněji nezaměřil na zpětnovazební řízení motoru, na získání vhodnějších parametrů regulátoru a dalších vstupů zapisovaných do paměti měniče, které jsem pouze převzal z vytvořených hodnot ladění v režimu automatického nastavení parametrů pohonu, nabízí se možnost v této práci pokračovat a zdokonalovat jednotlivé realizované funkce, nebo přidávat novější. Pro potenciálního následovatele může tato diplomová práce sloužit jako návod na seznámení se s vývojovým prostředím a popsánymi komponenty, kdy nemusí veškeré informace vyhledávat v obsáhlých manuálech jednotlivých modulů a rovnou se seznámit s dalšími funkcemi vývojového prostředí, které jsem ve svém projektu nepoužil.

Sám za sebe si myslím, že se mi podařilo vytvořit demonstrační úlohu s použitím PLC modulu, která dovoluje řídit pohon jedné osy, což byl záměr této práce.



## Seznam obrázků

Obr. 1: Reléové schéma [11] .....	6
Obr. 2: Příklad užití funkčních bloků [11] .....	6
Obr. 3: Třístavový signál, šedě: oddělovače bitových intervalů.....	10
Obr. 4: zásuvka typu RJ-45 MDI [11] .....	13
Obr. 5: zásuvka typu RJ-45 MDI-X [11].....	13
Obr. 6: Příklad užití SCADA systému .....	15
Obr. 7: Schéma zapojení PLC modulu 1215C DC/DC/DC [11] .....	19
Obr. 8: Sestava a) napájecího modulu, b) PLC modulu, c) přepínacího modulu .....	19
Obr. 9: Výrobní štítek měniče .....	20
Obr. 10: Výkonový měnič V90.....	21
Obr. 11: Možné způsoby upevnění přírubového motoru [27].....	22
Obr. 12: Příklad štítku motoru firmy Siemens [27] .....	23
Obr. 13: Synchronní motor s enkodérem .....	24
Obr. 14: Kotouč inkrementálního čidla (vlevo) a absolutního čidla (vpravo) [31], [32].....	25
Obr. 15: Dotykový panel .....	27
Obr. 16: Blok MC_MoveRelative, průběhy vstupních a výstupních parametrů .....	33
Obr. 17: Typy zobrazovacích prvků .....	35
Obr. 18: Hlavní obrazovka programu TIA Portal .....	37
Obr. 19: Výřez části okna pro nastavení technologického objektu .....	38
Obr. 20: Výřez podokna s PLC tagy .....	40
Obr. 21: Výřez podokna s HMI tagy .....	40
Obr. 22: Detekce a připojení zařízení.....	42
Obr. 23: Podokno s programovým blokem Main v režimu online monitorování .....	43
Obr. 24: Úvodní obrazovka dotykového panelu .....	44
Obr. 25: Druhá obrazovka dotykového panelu týkající se automatického procesu .....	45

## Seznam tabulek

Tab. 1: Uspořádání a zapojení pinů zásuvek RJ-45 dvou typů .....	13
Tab. 2: Vybrané technické údaje modulu PM1207 .....	16
Tab. 3: Vybrané technické údaje modulu CSM 1277 .....	17
Tab. 4: Vybrané technické údaje .....	18
Tab. 5: Technické údaje motoru.....	23
Tab. 6: Kódové značení motoru.....	24

## Seznam použitých zkratek

AC	alternating current (střídavý proud)
CPU	central processing unit (centrální procesorová jednotka)
CSMA/CD	carrier sense multiple access with collision detection (označení pro přístupovou metodu u ethernetových sítí)
DC	direct current (stejnoseměrný proud)
EMC	electromagnetic compatibility (elektromagnetická kompatibilita)
ESD	electrostatic sensitive devices (zařízení náchylná k elektrostatickému výboji)
FBD	function block diagram (typ blokového programovacího jazyka)
HSC	high speed control (označení pro typ vysokorychlostního enkodéru)
HMI	human machine interface (rozhraní člověk – stroj)
HW	hardware (fyzická část počítače)
IEEE	Institute of Electrical and Electronic Engineers (Institut pro elektrotechnické a elektronické inženýrství)
IP	internet protocol (protokol pro komunikaci vzájemně propojených zařízení)
LAD	ladder diagrams (název pro reléové schéma)
LAN	local area network (lokální místní síť)
LCD	liquid crystal display (displej z tekutých krystalů)
LLC	logical link control (logické řízení spojení)
MAC	media access control (identifikační číslo zařízení připojeného do sítě)
MDI	medium dependent interface (středně závislé rozhraní)
NIC	network interface controller (označení pro karty síťového rozhraní)
OSI	open systems interconnection (propojení otevřených systémů)
PLC	programmable logic controller (programovatelný logický automat)
SCADA	supervisory control and data acquisition (dispečerské řízení a sběr dat)
SELV	safety extra low voltage (bezpečné malé napětí)
SCL	structured control language (typ strukturovaného programovacího jazyka)
STO	safe torque off (funkce bezpečného odpojení motoru)
TFT	thin film transistor (technologie užití tenkovrstvých tranzistorů)
TTL	transistor transistor logic (tranzistorově tranzistorová logika)
USB	universal serial bus (univerzální sériová sběrnice)

## Seznam použité literatury

- [1] ČERNÝ, Jiří. Australští těžaři využívají plně autonomní vlak, do roku 2018 chtějí automatizovat celou železnici. In: E15.cz [online]. Czech News Center a.s., 2019. [1. 4. 2019]. Dostupné z: <https://www.e15.cz/byznys/prumysl-a-energetika/australsti-tezari-vyuzivaji-plne-autonomni-vlak-do-roku-2018-chteji-automatizovat-celou-zeleznici-1338271>
- [2] PLC Automatizace [online]. HaPeSoft s.r.o. [vid. 18. 1. 2019]. Dostupné z: <http://plc-automatizace.cz/knihovna/historie/historie-automatizace.htm>
- [3] TIŠNOVSKÝ, Pavel. Elektronkové a tranzistorové počítače Ural. In: Root.cz [online]. Root.cz, 2010. [vid. 22. 1. 2019]. Dostupné z: <https://www.root.cz/clanky/elektronkove-a-tranzistorove-pocitace-ural/>
- [4] Historie počítačů [online]. Sokolici.eu. [vid. 22. 1. 2019]. Dostupné z: <http://www.historie.sokolici.eu/30az50leta.html>
- [5] Vačkové spínače VSN [online]. Obzor, 2015. [vid. 20. 1. 2019]. Dostupné z: <https://www.obzor.cz/vyrobky/prumyslova-elektroinstalace/vackove-spinace-vsn?tabContentVSN-nodeId=364&tabContentVSN-rootNodeId=380&do=tabContentVSN-changeNode>
- [6] Historie počítačů [online]. Sokolici.eu. [vid. 22. 1. 2019]. Dostupné z: <http://www.historie.sokolici.eu/60a70leta.html>
- [7] PLC Automatizace [online]. HaPeSoft s.r.o. [vid. 25. 1. 2019]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-pozadavky.htm>
- [8] PLC Automatizace [online]. HaPeSoft s.r.o. [vid. 18. 1. 2019]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc.htm>
- [9] UniPi.technology. 7 tipů pro programování PLC. In: tzbinfo [online]. Topinfo s.r.o., 2018. [vid. 20. 1. 2019]. Dostupné z: <https://elektro.tzb-info.cz/informacni-a-telekomunikacni-technologie/18125-7-tipu-pro-programovani-plc>
- [10] All about Circuits [online]. All about Circuits. [vid. 20. 1. 2019]. Dostupné z: <https://www.allaboutcircuits.com/textbook/digital/chpt-6/ladder-diagrams/>
- [11] Siemens [online]. SIMATIC S7 S7-1200 Programmable controller. 2018. [vid. 6. 3. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/document/109759862/simatic-s7-s7-1200-programmable-controller?dti=0&pnid=13613&lc=en-WW>
- [12] Komunikace po sběrnici [online]. Mendel University in Brno. [vid. 12. 3. 2019]. Dostupné z: [https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=9947](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=9947)
- [13] Peterka, Jiří. Synchronní, asynchronní a arytmičkový přenos. In: eArchiv.cz [online]. Jiří Peterka, 2015. [vid. 22. 3. 2019]. Dostupné z: <http://www.earchiv.cz/a96/a650k150.php3>

- [14] PLC Automatizace [online]. HaPeSoft s.r.o. [vid. 23. 3. 2019]. Dostupné z: <http://plc-automatizace.cz/knihovna/periferie/komunikacni-periferie.htm>
- [15] Murrelektronik CZ, spol. s r. o. In: Automa [online]. Automa, 2016. [vid. 23. 3. 2019]. Dostupné z: [http://automa.cz/cz/casopis-clanky/profinet-versus-profibus-2012\\_05\\_0\\_9618/](http://automa.cz/cz/casopis-clanky/profinet-versus-profibus-2012_05_0_9618/)
- [16] What is the Difference between Profibus and Profinet? In: YouTube [online]. 19. 11. 2018 [vid. 15. 3. 2019]. Dostupné z: <https://www.youtube.com/watch?v=YxF9QgRAx8A&t=1s>. Kanál uživatele RealPars.
- [17] What is Ethernet? In: YouTube [online]. 14. 1. 2019 [vid. 17. 3. 2019]. Dostupné z: <https://www.youtube.com/watch?v=HLziLmaYsOO>. Kanál uživatele RealPars.
- [18] CSMA/CD [online]. IT slovník.cz. [vid. 22. 4. 2019]. Dostupné z: <https://it-slovník.cz/pojem/csma-cd>
- [19] Siemens [online]. S7-1200 Compact Switch Module CSM 1277. 2010. [vid. 2. 3. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/document/36087313/s7-1200-compact-switch-module-csm-1277-?dti=0&lc=en-WW>
- [20] Promotic [online]. Microsys, spol. s r. o. [vid. 10. 4. 2019]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/WhatIsPromotic/WhatIsScada.htm>
- [21] Siemens [online]. SIMATIC HMI HMI devices Basic Panels 2nd Generation. 2019. [vid. 15. 3. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/document/90114350/simatic-hmi-hmi-devices-basic-panels-2nd-generation?dti=0&pnid=14738&lc=en-WW>
- [22] Eden-Screen. In: The Vault [online]. WTW Media LLC and its. Licensors, 2019. [vid. 10. 4. 2019]. Dostupné z: <https://www.totallyintegratedautomation.com/wp-content/uploads/2017/09/Eden-Screen.jpg>
- [23] 6EP1332-1SH71. [online]. Siemens AG, 2019. [vid. 1. 3. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/pd/399209?pdtdi=td&dl=en&lc=en-WW>
- [24] Kříž M. Otázky a odpovědi z elektrotechnické praxe. In: Elektro [online]. FCC Public s. r. o., 2019. [vid. 1. 3. 2019]. Dostupné z: <http://www.odbornecasopisy.cz/elektro/casopis/tema/otazky-a-odpovedi-z-elektrotechnicke-praxe--14178>
- [25] Nosná lišta. [online]. Wikipedie, 2015. [vid. 1. 3. 2019]. Dostupné z: [https://cs.wikipedia.org/wiki/Nosn%C3%A1\\_li%C5%A1ta](https://cs.wikipedia.org/wiki/Nosn%C3%A1_li%C5%A1ta)
- [26] 6GK7277-1AA10-0AA0. [online]. Siemens AG, 2019. [vid. 2. 3. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/pd/393283?pdtdi=td&dl=en&pnid=13613&lc=en-WW>
- [27] Siemens [online]. SINAMICS V90 PROFINET, SIMOTICS S-1FL6 Getting Started. 2018. [vid. 8. 3. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/document/109763151/sinamics-v90-profinet-simotics-s-1fl6-getting-started?dti=0&pnid=13211&lc=en-WW>

- [28] Siemens [online]. SINAMICS V90 PROFINET, SIMOTICS S-1FL6 Operating Instructions. 2018. [vid. 8. 3. 2019]. Dostupné z: <https://support.industry.siemens.com/cs/document/109763150/sinamics-v90-profinet-simotics-s-1fl6-operating-instructions?dti=0&pnid=13204&lc=en-WW>
- [29] KOBRLE, Pavel a Jiří PAVELKA. Elektrické pohony a jejich řízení. 3. přepracované vydání. V Praze: České vysoké učení technické, 2016. ISBN 9788001060070.
- [30] TTL [online]. IT slovník.cz. [vid. 11. 5. 2019]. Dostupné z: <https://it-slovník.cz/pojem/ttl>
- [31] Aspone [online]. Radek Fejt, 2012. [vid. 5. 4. 2019]. Dostupné z: <http://senzory.aspone.cz/ois.htm>
- [32] PLC Automatizace [online]. HaPeSoft s.r.o. [vid. 5. 4. 2019]. Dostupné z: <http://plc-automatizace.cz/knihovna/data/kodovani/IRC-code.htm>
- [33] DOLEČEK, Jaroslav. *Moderní učebnice elektroniky*. Praha: BEN - technická literatura, 2005. ISBN 8073001845.
- [34] MACH, Pavel, Jan URBÁNEK a Vlastimil SKOČIL. *Montáž v elektronice: pouzdření aktivních součástek, plošné spoje*. Praha: Vydavatelství ČVUT, 2001. ISBN 8001023923.
- [35] SIMATIC WinCC flexible [online]. Siemens AG, 2019. [vid. 14. 4. 2019]. Dostupné z: <https://w3.siemens.com/mcms/human-machine-interface/en/visualization-software/wincc-flexible/pages/default.aspx>

# Příloha

Schéma vytvořené ve vývojovém prostředí kombinací programovacích jazyků LAD a FBD

