

**ČESKÉ VYSOKÉ UČENÍ  
TECHNICKÉ V PRAZE  
FAKULTA ELEKTROTECHNICKÁ  
KATEDRA ELEKTROENERGETIKY**



**DIPLOMOVÁ PRÁCE**

**OŽIVENÍ PRACOVIŠTĚ PRO  
VZDÁLENÉ MĚŘENÍ  
CHARAKTERISTIK POHONU**

**JAKUB RYPAN**

**VEDOUČÍ PRÁCE: DOC. ING. JAN BAUER PH.D.**

**2024**



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	<b>Rypan</b>	Jméno: <b>Jakub</b>	Osobní číslo: <b>483831</b>
Fakulta/ústav:	<b>Fakulta elektrotechnická</b>		
Zadávací katedra/ústav:	<b>Katedra elektrických pohonů a trakce</b>		
Studijní program:	<b>Elektrotechnika, energetika a management</b>		
Specializace:	<b>Elektrické pohony</b>		

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Oživení pracoviště pro vzdálené měření charakteristik pohonu**

Název diplomové práce anglicky:

**Testbed for a Remote Measurement of Drive Characteristics**

Pokyny pro vypracování:

- 1) z dostupných komponentů navrhnete a zprovozníte měřicí pracoviště pro měření charakteristik pohonu a ladění regulace
- 2) připojte měnič k PLC, navrhnete řízení a doplňte vhodnou vizualizaci
- 3) na HMI implementujte obrazovky pro vizualizaci a záznam regulačních dějů a změnu parametrů regulátorů
- 4) zprovozníte vzdálenou správu tohoto pracoviště

Seznam doporučené literatury:

- [1] Weidauer J., Messer R. Electrical Drives, Publics Erlangen, 2014
- [2] SCE Training Curriculum. Siemens AG, 2016
- [3] Durry B. The Control Techniques Drives and Controls Handbook 2nd ed., IeT, 2009

Jméno a pracoviště vedoucí(ho) diplomové práce:

**doc. 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: **25.04.2023** Termín odevzdání diplomové práce: \_\_\_\_\_

Platnost zadání diplomové práce: **16.02.2025**

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

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



## PODĚKOVÁNÍ

---

Tímto bych rád poděkoval svému vedoucímu práce Doc. Ing. Jan Bauer, Ph.D. za cenné rady, doporučení, a hlavně za trpělivost a ochotu pomoci při psaní této práce. Dále bych rád poděkoval oddělení technické podpory firmy Siemens za možnost konzultovat realizační část této práce a za to, že mi vždy ochotně pomohli. Nakonec bych rád poděkoval své rodině a přátelům, kteří mě motivovali po celou dobu studia, díky nim jsem vždy viděl světlo na konci tunelu.

## 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 8. ledna 2024

.....

## ABSTRAKT

---

Tato diplomová práce se zabývá zprovozněním vzdáleného stanoviště k měření přechodových charakteristik pohonu. Stanoviště obsahuje dva mechanicky spřažené asynchronní motory, kdy jeden zastává úlohu zátěže, měnič S120 napájící oba motory, PLC k řízení celého procesu a HMI panel k vizualizaci a ovládání stanoviště. Práce začíná teoretickým úvodem, který obsahuje základní informace o použitých zařízeních. Následuje popis zvoleného postupu při realizaci stanoviště. Výsledkem práce je plně funkční stanoviště umožňující měřit přechodové charakteristiky požadovaného charakteru. Práce je zakončena vzorovými naměřenými průběhy.

## KLÍČOVÁ SLOVA:

---

PLC, Frekvenční měniče, vzdálená správa, S120, HMI, přechodové charakteristiky.

## ABSTRACT

---

This thesis deals with the operation of a remote station for measuring the step response characteristics of the drive. The station contains two mechanically coupled asynchronous motors, one of which acts as a load, an S120 inverter supplying both motors, a PLC to control the whole process and an HMI panel to visualize and control the station. The thesis starts with a theoretical introduction which contains basic information about the devices used. This is followed by a description of the chosen procedure for the implementation of the station. The result of the work is a fully functional station allowing to measure step response characteristics of the required nature. The work is concluded with representative measured relations.

## KEYWORDS:

---

PLC, variable frequency drives, remote control, S120, HMI, step responses.

# OBSAH

---

<b>ÚVOD</b> .....	<b>1</b>
<b>KAPITOLA 1: ÚVOD DO AUTOMATIZACE</b> .....	<b>2</b>
<b>1.1 PROGRAMOVATELNÝ LOGICKÝ AUTOMAT</b> .....	<b>2</b>
1.1.1 Centrální procesní jednotka .....	3
1.1.2 Paměť.....	4
1.1.3 Vstupně-výstupní rozhraní .....	4
1.1.3.1 Digitální (binární) vstupy .....	5
1.1.3.2 Digitální (binární) výstupy .....	7
1.1.3.3 Analogové vstupy .....	9
1.1.3.4 Analogové výstupy.....	11
1.1.4 Komunikační rozhraní .....	12
1.1.4.1 Ethernet.....	13
1.1.4.2 Profinet.....	15
<b>1.2 SIEMENS PLC</b> .....	<b>20</b>
1.2.1 Portfolio systémů SIMATIC.....	21
<b>1.3 HMI PANELY</b> .....	<b>22</b>
<b>1.4 FREKVENČNÍ MĚNIČE</b> .....	<b>22</b>
1.4.1 Měniče Siemens .....	23
1.4.1.1 Základní funkce měniče .....	24
1.4.1.2 Pokročilé technologické funkce.....	25
1.4.1.3 Bezpečnostní funkce .....	25
1.4.1.4 Komunikační funkce .....	26
<b>1.5 REGULACE</b> .....	<b>27</b>
1.5.1 Přejížděvací charakteristika.....	28
1.5.2 Regulátory.....	30
<b>1.6 TYPY ŘÍZENÍ POHONŮ</b> .....	<b>31</b>
<b>KAPITOLA 2: MOŽNOSTI VZDÁLENÉ SPRÁVY</b> .....	<b>32</b>
<b>KAPITOLA 3: POPIS ÚLOHY</b> .....	<b>34</b>
<b>3.1 POUŽITÝ HARDWARE</b> .....	<b>36</b>
3.1.1 Frekvenční měnič.....	36
3.1.1.1 Řídicí jednotka.....	37
3.1.1.2 Smart line module .....	37
3.1.1.3 Motorový modul .....	37
3.1.1.4 Enkodér .....	38

3.1.2	Motory.....	38
3.1.2.1	Teplotní čidlo k ochraně motoru.....	39
3.1.3	PLC.....	39
3.1.4	HMI.....	39
<b>3.2</b>	<b>SCHÉMA ZAPOJENÍ.....</b>	<b>39</b>
<b>3.3</b>	<b>TIA PORTÁL.....</b>	<b>45</b>
3.3.1.1	Žebříčkové diagramy LD.....	46
3.3.1.2	Diagram funkčních bloků FBD.....	47
3.3.1.3	Strukturovaný řídicí jazyk SCL.....	48
<b>KAPITOLA 4: TVORBA PROJEKTU V TIA PORTÁLE.....</b>		<b>49</b>
<b>4.1</b>	<b>HW KONFIGURACE MĚNIČE .....</b>	<b>49</b>
<b>4.2</b>	<b>HW KONFIGURACE PLC .....</b>	<b>51</b>
<b>4.3</b>	<b>NASTAVENÍ KOMUNIKACE .....</b>	<b>52</b>
<b>4.4</b>	<b>TECHNOLOGICKÉ OBJEKTY .....</b>	<b>53</b>
<b>4.5</b>	<b>BEZPEČNOSTNÍ PROGRAM.....</b>	<b>58</b>
<b>4.6</b>	<b>POPIS POUŽITÝCH FUNKČNÍCH BLOKŮ.....</b>	<b>60</b>
4.6.1	FB Motor a Load.....	60
4.6.2	Sina_para_s.....	61
4.6.3	Tvorba zátěžné charakteristiky.....	64
4.6.4	FB Program_control.....	64
<b>4.7</b>	<b>VIZUALIZACE.....</b>	<b>67</b>
<b>4.8</b>	<b>SMARTCLIENT .....</b>	<b>68</b>
<b>KAPITOLA 5: LOGIKA ŘÍDICÍHO PROGRAMU.....</b>		<b>71</b>
<b>5.1</b>	<b>AUTOMATICKÁ SEKVENCE MĚŘENÍ PŘECHODOVÉ CHARAKTERISTIKY .....</b>	<b>71</b>
<b>5.2</b>	<b>MANUÁLNÍ OVLÁDÁNÍ MOTORŮ V REŽIMU MOTOR-ZÁTĚŽ.....</b>	<b>75</b>
<b>5.3</b>	<b>MANUÁLNÍ OVLÁDÁNÍ MOTORŮ SAMOSTATNĚ.....</b>	<b>77</b>
<b>5.4</b>	<b>POPIS PROGRAMU K UKLÁDÁNÍ NAMĚŘENÝCH HODNOT .....</b>	<b>79</b>
<b>KAPITOLA 6: NAMĚŘENÉ PRŮBĚHY.....</b>		<b>82</b>
<b>ZÁVĚR.....</b>		<b>88</b>
<b>LITERATURA.....</b>		<b>89</b>



## SEZNAM ZKRATEK

---

AC	Alternating current ( střídavý proud)
AD	Analog-digital (analogově digitální)
AI	Analog input (analogový vstup)
AO	Analog output (analogový výstup)
AWG	American wire gauge (americká míra kabelů)
BICO	Binector-connector
CC	Conformance class
CPU	Central processing unit (centrální řídicí jednotka)
CRC	Cyclic redundancy check (cyklický redundanční součet)
DA	Digital-analog (digitálně-analogový)
DC	Direct current (stejnoseměrný proud)
DCC	Drive control chart („tabulka k řízení pohonů“)
DI	Digital input (digitální vstup)
DO	Digital output (digitální výstup)
DSL	Digital subscriber line (digitální účastnická přípojka)
EEPROM	Electrically erasable programmable read only memory
EMC	Electromagnetic compatibility (elektromagnetická kompatibilita)
EPOS	Easy basic positioner (jednoduchý základní polohovač)
FB	Function block (Funkční blok)
FBD	Function block diagram (diagram funkčních bloků)
FC	Function (Funkce)
FFB	Free function blocks (volné funkční bloky)
HMI	Human machine interface (rozhraní člověk-stroj)
HTL	High threshold logic
IEC	International electrotechnical commission (mezinárodní elektrotechnická komise)
IEEE	Institute of electrical and electronics engineers (institut pro elektrotechnické a elektronické inženýrství)
IGBT	Insulator gate bipolar transistor (bipolární tranzistor s izolovaným hradlem)
IO	Input output (vstupně výstupní)
IRT	Isochronous real time (izochronní)
ISO	International organization for standardization (mezinárodní organizace pro normalizaci)
ISO/OSI	International organization for standardization/Open systems interconnection (referenční model ISO/OSI)
LAN	Local area network (místní síť)

LD	Ladder diagram (žebříčový diagram)
LTE	Long term evolution
MAN	Metropolitan area network (metropolitní síť)
MMC	Micro memory card (mikro paměťová karta)
MOV	Metal-oxid varistor (metal-oxidový varistor)
NC	Normally closed (rozpínací)
NO	Normally open (spínací)
NRT	Non-time critical (časově nekritické)
OB	Organization block (organizační blok)
PI	Profibus & Profinet International
PLC	Programmable logic controller (programovatelný logický automat)
RAM	Random access memory (paměť s náhodným přístupem)
ROM	Read only memory (paměť určená pro čtení) (elektricky vymazatelná programovatelná paměť určená ke čtení)
RT	Real time (v reálném čase)
SBC	Safe brake control (bezpečné řízení brzd)
SCL	Structured control language (strukturovaný řídicí jazyk)
SD	Safe direction (bezpečný směr)
SLM	Smart line module
SLP	Safety limited position (bezpečné omezení polohy)
SLS	Safety speed limited (bezpečně limitovaná rychlost)
SNMP	Simple network management protocol
SOS	Safe operating stop (bezpečné zastavení a udržení polohy)
SP	Safety position (bezpečná pozice)
SS1	Safe stop 1 (bezpečné zastavení 1)
SS2	Safe stop 2 (bezpečné zastavení 2)
SSM	Safety speed monitor (bezpečně monitorovaná rychlost)
STL	Statement list (seznam instrukcí)
STO	Safe torque off (bezpečné vypnutí momentu)
TEC	Technology extensions (technologické rozšíření)
TCP	Transmission control protocol (protokol k řízení přenosu)
UDP	User defined protocol (uživatelé definovaný protokol)
UMTS	Universal mobile telecommunication system (Univerzální mobilní telekomunikační systém)
VPN	Virtual private network (virtuální privátní síť)

## SEZNAM OBRÁZKŮ

---

Obr. 1-1: Vnitřní struktura typického PLC. [3].....	3
Obr. 1-2: Schéma střídavého digitálního vstupu. [2].....	5
Obr. 1-3: Schéma zapojení stejnosměrného digitálního vstupu. [2] .....	6
Obr. 1-4 Rozdíl mezi sinking a sourcing periferiemi u vstupních modulů. Za a) vlevo je vstupní module typu <i>sourcing</i> , za b) je vstupní modul typu <i>sinking</i> . [2].....	6
Obr. 1-5 Schéma zapojení digitálního výstupu s triakem, pro střídavé obvody. [2] .....	7
Obr. 1-6 Schéma zapojení digitálního výstupu s tranzistorem, pro stejnosměrné obvody. [2] .....	8
Obr. 1-7 Rozlišení výstupních modulů typu sinking a typu sourcing. Za a) vlevo je výstupní module typu <i>sourcing</i> , za b) vpravo je výstupní module typu <i>sinking</i> . [2].....	8
Obr. 1-8 Odporový dělič s přepínáním rozsahů pro měření napětí (vlevo) a odporové bočníky s přepínáním rozsahu pro měření proudu. [9].....	9
Obr. 1-9 Principiální blokové schéma AD převodníku pro 8 kanálů analogových vstupů. [9] .....	11
Obr. 1-10 Principiální blokové schéma modulu DA převodníku pro 8 kanálů analogových výstupů. [10] .....	12
Obr. 1-11 Struktura vrstev referenčního ISO/OSI modelu. [11] .....	13
Obr. 1-12 Rozdělení aplikací podle časové náročnosti a délky komunikačního cyklu. [12].....	16
Obr. 1-13 Struktura Profinet komunikace z pohledu referenčního modelu. [13] .....	17
Obr. -1-14 Rozdělení IRT komunikace na rezervovaný červený interval a na otevřený zelený interval. [14] .....	18
Obr. 1-15 Struktura konformačních tříd. [14].....	18
Obr. 1-16 Možné topologie Profinetu. [12].....	19
Obr. 1-17 Portfolio řídicích systémů SIMATIC. [18] .....	21
Obr. 1-18 Nepřímé měniče frekvence s napěťovým meziobvodem. [22] .....	23
Obr. 1-19 Základní rozdělení měničů SINAMICS na řadu V, G a S. [23] .....	24
Obr. 1-20 Bezpečnostní funkce STO (vlevo) a SS1 (vpravo). [23].....	26
Obr. 1-21 Mezinárodní standardy týkající se bezpečnosti a komunikačních sběrnic v průmyslu. [24] ....	27
Obr. 1-22 Blokové schéma obecného regulačního obvodu. [5] .....	28
Obr. 1-23 Přejímová charakteristika. [5] .....	29
Obr. 1-24 Přejímová charakteristika. [25].....	29
Obr. 1-25 Vnitřní struktura regulátoru. [26] .....	30
Obr. 2-1 Přehled možností vzdáleného připojení s pomocí SINEMA RC. [28] .....	32
Obr. 2-2 Schéma použití Sm@rtServeru. [29] .....	33
Obr. 3-1 Základní silové schéma stanoviště. ....	34
Obr. 3-2 Průběhy zátěžného momentu.....	35
Obr. 3-3 Štítek motoru. ....	38
Obr. 3-4 Schéma zapojení realizovaného stanoviště.....	40
Obr. 3-5 Realizované stanoviště: PLC, měnič, HMI panel a počítač s TIA portálem. ....	42
Obr. 3-6 Mechanicky spřažené motory. ....	42
Obr. 3-7 Detailní fotografie HMI panelu.....	43
Obr. 3-8 Detailní fotografie stojanu. ....	44

Obr. 3-9 Vývojové prostředí TIA portál.....	45
Obr. 3-10 Ukázka žebříčkového diagramu. ....	47
Obr. 3-11 Ukázka diagramu funkčních vstupů.....	47
Obr. 3-12 Ukázka programovací jazyku SCL. ....	48
Obr. 4-1 Struktura použitých komunikačních telegramů. ....	52
Obr. 4-2 Motion control bloky v katalogové části TIA portálu.....	54
Obr. 4-3 Blok MC_power (pro osu reprezentující zátěž).. ....	55
Obr. 4-4 Blok MC_Reset. ....	55
Obr. 4-5 Blok MC_MoveVelocity použit k ovládní rychlostní osy zastávající funkci motoru. ....	56
Obr. 4-6 Blok MC_Stop. ....	57
Obr. 4-7 Blok MC_TorqueRange.....	57
Obr. 4-8 Adresy PROFIsafe telegramů.....	58
Obr. 4-9 Main safety – příklad bezpečnostního programu pro motor. ....	59
Obr. 4-10 Blok Motor, který obsahuje MC bloky k ovládní osy motoru. ....	60
Obr. 4-11 Blok Load, který obsahuje MC bloky k ovládní osy zátěže. ....	61
Obr. 4-12 Funkční blok Sina_para_s.....	62
Obr. 4-13 Logika funkčního bloku, který ovládá asynchronní komunikaci pomocí Sina_para_s. ....	63
Obr. 4-14 FB Asynch_comm, který ovládá FB Sina_para_s a tedy i acyklickou komunikaci mezi měničem a PLC.....	63
Obr. 4-15 Blok Load_char_selection, který vypočte požadovanou zátěžovou charakteristiku. ....	64
Obr. 4-16 První polovina bloku Program_control. ....	65
Obr. 4-17 Druhá polovina bloku Program_control.....	66
Obr. 4-18 Použitá struktura obrazovek v HMI panelu.....	67
Obr. 4-19 Hlavní obrazovka HMI panelu. ....	68
Obr. 4-20 Povolení Smartserveru v HMI panelu v TIA portál projektu. ....	69
Obr. 4-21 Nastavení "Remote connection" v HMI panelu část 1.....	69
Obr. 4-22 Nastavení "Remote connection" v HMI panelu část 2.....	70
Obr. 4-23 Sm@rtclient aplikace před připojením k panelu. ....	70
Obr. 4-24 Sm@rtclient aplikace po připojení k panelu.....	70
Obr. 5-1 Logika automatické sekvence měření přechodových charakteristik. ....	72
Obr. 5-2 Obrazovka k nastavení automatického měření přechodové charakteristiky. ....	73
Obr. 5-3 Obrazovka k monitorování automatického měření a následnému uložení naměřených.....	74
Obr. 5-4 Obrazovka s diagnostikou hodnot.....	74
Obr. 5-5 Logika manuálního řízení motorů, kdy jeden reprezentuje zátěž a druhý motor. ....	76
Obr. 5-6 Obrazovka k manuálnímu ovládní měření. ....	77
Obr. 5-7 Logika manuálního řízení motorů samostatně.....	78
Obr. 5-8 Obrazovka k manuálnímu ovládní motorů zvlášť. ....	79
Obr. 6-1 Změřený průběh kmitavě stabilní přechodové charakteristiky při $k_p=2$ , $T_1=10$ a $M_z$ s konstantní charakteristikou, 10 % jmenovitého momentu motoru. ....	82
Obr. 6-2 Změřený průběh kmitavě stabilní přechodové charakteristiky při $k_p=2$ , $T_1=10$ a $M_z$ s konstantní charakteristikou, 90 % jmenovitého momentu motoru. ....	83

Obr. 6-3 Změřený průběh aperiodicky stabilní přechodové charakteristiky při $k_p=10$ , $T_1=50$ a $M_z$ s konstantní charakteristikou, 10 % jmenovitého momentu motoru. ....	84
Obr. 6-4 Změřený průběh aperiodicky stabilní přechodové charakteristiky při $k_p=10$ , $T_1=50$ a $M_z$ s konstantní charakteristikou, 90 % jmenovitého momentu motoru. ....	84
Obr. 6-5 Změřený průběh aperiodicky stabilní přechodové charakteristiky při $k_p=5$ , $T_1=750$ a $M_z$ s konstantní charakteristikou, 10 % jmenovitého momentu motoru. ....	85
Obr. 6-6 Změřený průběh aperiodicky stabilní přechodové charakteristiky s dlouhou dobou náběhu při $k_p=5$ , $T_1=750$ a $M_z$ s konstantní charakteristikou, 90 % jmenovitého momentu motoru. ....	86
Obr. 6-7 Změřený průběh kmitavé přechodové charakteristiky na hranici stability při $k_p=50$ , $T_1=500$ , $M_z$ s konstantní charakteristikou, 10 % jmenovitého momentu motoru a požadavku na rychlost otáček 800 otáček/sekundu. ....	86

## SEZNAM TABULEK

---

Tab. 3-1 Seznam použitého hardwaru.....	36
Tab. 3-2 Barevné označení kabelu připojeného k enkodéru.....	38
Tab. 3-3 Použité licence v rámci vývojového prostředí TIA portál. ....	45
Tab. 6-1 Tabulka popisující, jak dobře byl naladěn otáčkový regulátor. ....	87







## ÚVOD

Zadáním této práce je vytvořit stanoviště, které umožní vzdáleně měřit charakteristiky pohonu, konkrétně přechodové charakteristiky otáčkového regulátoru asynchronního motoru. Stanoviště je tvořeno dvěma motory, programovatelným logickým automatem, HMI panelem a měničem, který napájí oba motory. Motory jsou mechanicky spřaženy, přičemž jeden motor zastává úlohu zátěže, generuje tedy zátěžný moment požadovaného charakteru. Druhý motor slouží jako pohon, kterému je laděn otáčkový regulátor. Zda je otáčkový regulátor správně naladěn lze poznat právě pomocí přechodových charakteristik, které jsou měřeny na tomto stanovišti. Pro jednoduché a přehledné řízení laboratorní úlohy je vytvořeno několik řídicích obrazovek na HMI panelu, které umožní uživateli řídit a monitorovat celý proces měření a výsledné hodnoty uložit. Stanoviště bude používáno studenty elektrotechnické fakulty ČVUT při laboratorních cvičeních. Stanoviště je sestrojeno vhodně zvoleným hardwarem firmy Siemens.

## KAPITOLA 1: ÚVOD DO AUTOMATIZACE

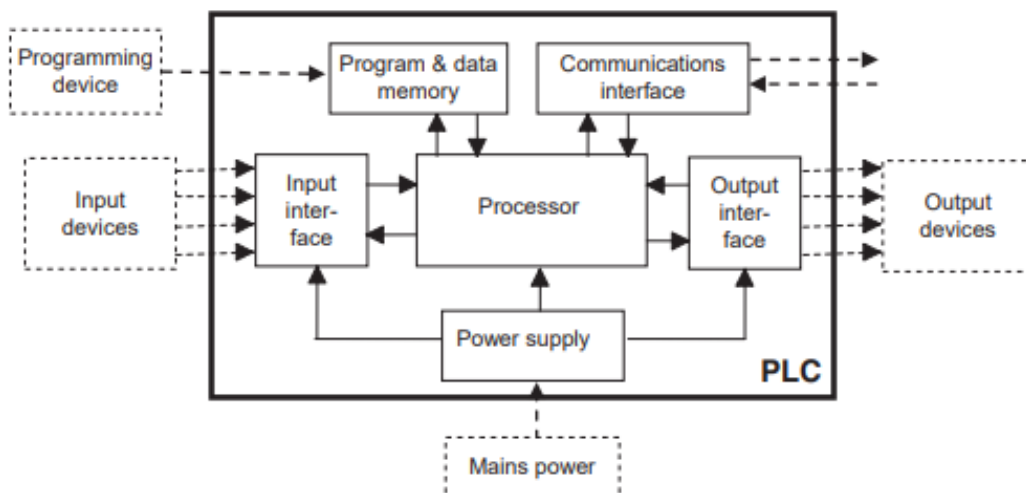
Automatizace prošla v průběhu let značnými změnami, než nabyla dnešní podoby. Počátky automatizace přišly díky vynálezu relé (zařízení sloužící ke spínání signálu). Relé bylo vynalezeno v roce 1835, avšak k automatizaci se relé ve velké míře začaly používat až na začátku 20. století. Tomuto období se říká tzv. reléová automatizace, kdy k automatizaci technologií a procesů, které byly definovány konečným počtem stavů, stačilo použít pouze relé. V případě potřeby složitějšího řízení probíhalo řízení procesu z centrální místnosti, kde po analýze aktuálního stavu operátorem byly předány pokyny k úpravě polohy akčních členů k dosažení požadovaného stavu. [1]

Elektromechanické relé (ovládací relé, pneumatické časové relé atd.) dobře sloužily po mnoho generací, často za nepříznivých podmínek. Stále se zvyšující sofistikovanost a složitost moderních výrobních zařízení však vyžadovala rychlejší a spolehlivější řídicí funkce, než elektromechanická relé mohla nabídnout. Relé musí být zadrátována napevno, aby mohla vykonávat požadovanou funkci, a když se změní požadavky systému, musí se změnit celkové zapojení relé. V extrémních případech, například v automobilovém průmyslu, bylo nutné vyměnit celé ovládací panely, protože nebylo ekonomicky přijatelné přepojovat staré panely při každé změně. Právě požadavky automobilového průmyslu a dalších vysoce specializovaných vysokorychlostních výrobních procesů vytvořily dostatečně velkou poptávku po menších, rychleji působících a spolehlivějších řídicích zařízeních. Elektrotechnický průmysl na to reagoval modulárně konstruovanými polovodičovými elektronickými zařízeními nazývanými PLC (tj. *Programmable Logic Controller*, česky programovatelný logický automat). [2]

### 1.1 Programovatelný logický automat

Programovatelný logický automat (dále jen PLC) je průmyslový počítač, přizpůsobený ztíženým podmínkám průmyslového prostředí jako jsou například široké rozsahy okolních teplot, vibrací, vlhkostí a elektrického šumu. PLC je speciální formou řídicí jednotky založené na mikroprocesoru, která využívá programovatelnou paměť k ukládání instrukcí a k realizaci například logických, sekvenčních, časových, početních a aritmetických funkcí za účelem řízení technologických a výrobních procesů. [3]

PLC se skládá z centrální procesní jednotky (dále jako *CPU – Central Processing Unit*), paměti, napájecí jednotky, vstupně-výstupního rozhraní a komunikačního rozhraní. Struktura PLC je vidět na Obr. 1-1.



Obr. 1-1: Vnitřní struktura typického PLC. [3]

### 1.1.1 Centrální procesní jednotka

Centrální procesní jednotka bývá často zaměňována s procesorem nebo mikroprocesorem. CPU však obsahuje všechny potřebné elementy k vytvoření inteligence systému, zatímco procesor je pouze prvkem CPU. CPU je hlavní část PLC, jedná se o tzv „mozek“ PLC, který umožňuje PLC vykonávat veškeré potřebné operace. Mezi tyto operace patří například:

1. Aktualizování hodnot vstupů a výstupů. Tedy čtení statusu vstupních terminálů a zapínání nebo vypínání výstupních terminálů.
2. Provedení logických nebo aritmetických operací. CPU se stará o všechny matematické a logické operace potřebné k fungování PLC.
3. Komunikace a výměna dat s pamětí.
4. „Skenování“ uživatelského programu. Uživatelský program je set instrukcí naprogramovaných PLC programátorem. Skenovací funkce programu umožňuje funkčnost PLC tak, jak PLC vyžaduje.
5. Komunikace s programovacím zařízením, typicky počítačem, na kterém je nainstalováno programovací prostředí pro PLC dané firmy.

O správnou funkčnost centrální procesní jednotky se stará operační systém. Operační systém je skupina řídicích programů, které jsou výrobcem nahrány a uloženy v permanentní paměti PLC. Operační systém se stará o všechny funkce, sekvence a operace CPU, které nejsou spojeny s vykonáváním uživatelského programu k řízení procesu. Jedná se například o zahájení restartu PLC, aktualizaci hodnot vstupů a výstupů, provoz uživatelského programu, správa paměti, detekce chyb a jejich volání a vytváření spojení s programovacími zařízeními. [4] [5] [6] [7]

### 1.1.2 Paměť

Paměť umožňuje ukládat informace, programy a data v PLC. Paměť lze dělit několika způsoby. Nejčastěji se paměť dělí tzv. stálou a nestálou paměť (z anglického *volatile* a *non-volatile*). Toto dělení souvisí s tím, jak stabilní jsou uložená data. Stálá paměť je stabilní a uchovává data v paměti i při ztrátě napájení, zatímco nestálá paměť při ztrátě napájení o uložená data přijde.

V PLC se nejčastěji vyskytují dva typy paměti: *Read Only Memory* – ROM (paměť určená pouze k čtení) a *Random Access Memory* – RAM (paměť s náhodným přístupem). ROM je typ paměti, do které nelze data zapisovat a uložená data a programy nelze měnit, pouze číst. Je to stálá paměť. V této části paměti je uložený například operační systém (OS) PLC. RAM je typ paměti, ze které lze data jak číst, tak je do ní zapisovat. Je to paměť nestálá. Používá se například pro uživatelské data a aktuální hodnoty proměnných. Zároveň je zde uložený uživatelský program. Jelikož se však jedná o nestálou paměť, tak by při ztrátě napájení došlo ke ztrátě uživatelského programu. Proto je potřeba buď mít v PLC záložní baterie, nebo vytvářet zálohu uživatelského programu. Tato záloha se nejčastěji provádí do dalšího typu paměti, tzv. EEPROM – *Electrically Erasable Programmable Read Only Memory* (elektricky vymazatelná programovatelná paměť určená ke čtení). Do této paměti lze jak zapisovat, tak z ní číst a je to paměť stálá. Rozdíl oproti RAM je, že mazání a zapisování do této paměti není tak rychlé.

V rámci této diplomové práce je použit hardware od firmy Siemens, proto by bylo vhodné přiblížit, jak je uspořádaná paměť u PLC firmy Siemens. Siemens dělí paměti PLC na 3 části:

- **Load memory:** stálá paměť, která slouží k uložení uživatelského programu a hardwarové konfigurace. Při stažení projektu do PLC se uživatelský program uloží právě sem. K rozšíření velikosti této paměti lze použít Siemens paměťové karty (MMC – *micro memory card*).
- **Work memory:** nestálá paměť, která slouží k ukládání některých elementů uživatelského programu při běhu uživatelského programu. Jde o paměť typu RAM.
- **Retentive memory:** stálá paměť, která slouží k uložení limitovaného množství dat, které se zachovají i v případě výpadku napájení. [4] [8]

### 1.1.3 Vstupně-výstupní rozhraní

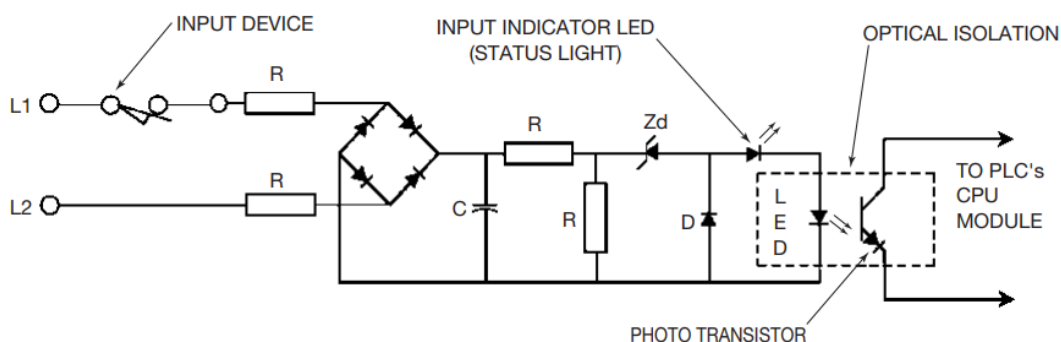
K tomu, aby PLC mohlo ovládat dané procesy je potřeba výstupních zařízení. Stav výstupních zařízení je určen na základně uživatelem naprogramovaném programu. Tento program bere v potaz informace o stavu vstupních zařízení a následně podle logiky programu vyhodnotí požadovaný stav výstupních zařízení. Cílem výstupního rozhraní je následně přeměnit nízké napětí vyskytující se v procesoru na požadovanou hodnotu napětí na výstupním zařízení. Jak vstupní, tak výstupní signály lze dělit na analogové signály a digitální signály. Digitálním

signálem je myšlen diskrétní signál, který nabývá pouze dvou hodnot a který o daném prvku vypovídá, zda je zapnutý či vypnutý. Proto bude tento typ signálu označován jako binární signál. Analogový signál je signál, který obsahuje hodnotu v určitém rozmezí, přičemž tato hodnota se může spojitě měnit v čase. PLC tedy pracuje s binárními vstupy, binárními výstupy, analogovými vstupy a analogovými výstupy.

### 1.1.3.1 Digitální (binární) vstupy

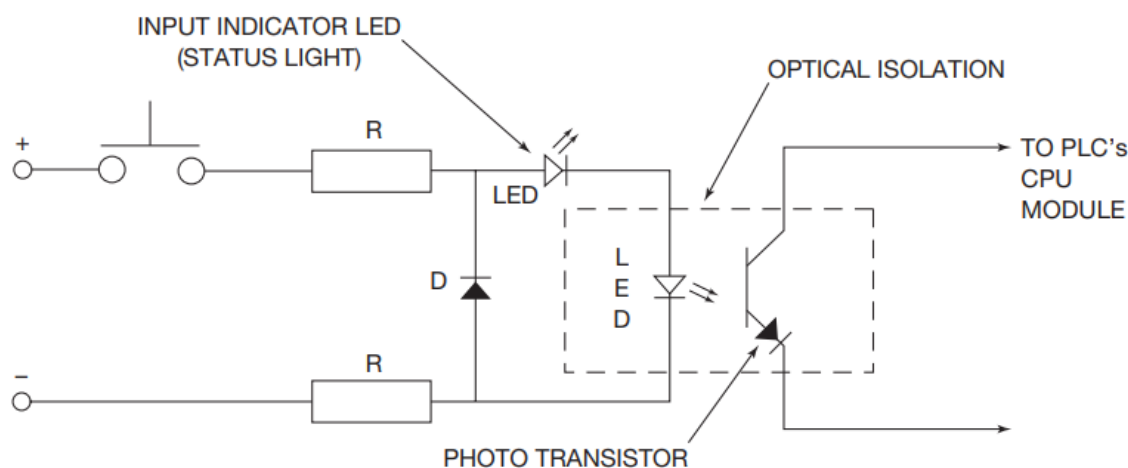
Výše zmíněné digitální vstupy slouží k tomu, abychom rozpoznali, zda je nějaké zařízení zapnuté nebo vypnuté. Informace o stavu digitálních vstupů se přenáší binárním signálem. K digitálním vstupům patří například tlačítka, přepínače, koncové spínače či jakékoliv jiné snímače s dvouhodnotovým charakterem signálu (např. dvouhodnotové snímače tlaku, teploty či hladiny). Digitální vstupy mohou používat stejnosměrné napětí či napětí střídavé. Typické hodnoty pro střídavé napětí jsou 230 V a 120 V a pro stejnosměrné napětí 12 V a 24 V.

Na Obr. 1-2 je zobrazeno schéma zapojení střídavého digitálního vstupu. Střídavé napětí procházející vstupním zařízením (pokud je v sepnutém stavu) je pomocí můstku usměrněno na stejnosměrné. Také se zde vyskytuje filtrovací obvod, který chrání zařízení před elektrickým rušením. Tento obvod je tvořen odporem a kondenzátorem. Pokud by zde tento obvod nebyl, mohlo by docházet ke krátkým stejnosměrným pulzům způsobeným elektrickým rušením, které by procesor mohl interpretovat tak, že zařízení je sepnuté, když zařízení sepnuté není a naopak. Kvůli tomuto filtru je však nutné, aby napětí ze vstupního zařízení bylo v předem definovaném rozmezí a aby se vyskytovalo v obvodu po určitou dobu, protože filtr zpozdí konečný signál o 10-25 ms. Filtrovaný signál jde do procesoru, je oddělen tzv. optočlenem. Optočlen je zařízení, které zajišťuje oddělení vstupního elektrického obvodu a vnitřního elektrického obvodu, který předává informaci o vstupním signálu do procesoru. Toto oddělení je nutné, aby bylo dosaženo spolehlivosti a aby byl ochráněn procesor PLC. Procesory typicky pracují na napěťové hladině 5 V DC, a proto by hrozilo, že bez elektrického oddělení vstupního obvodu by se mohlo zařízení porouchat při nesprávném zapojení nebo vlivem napěťových špiček.



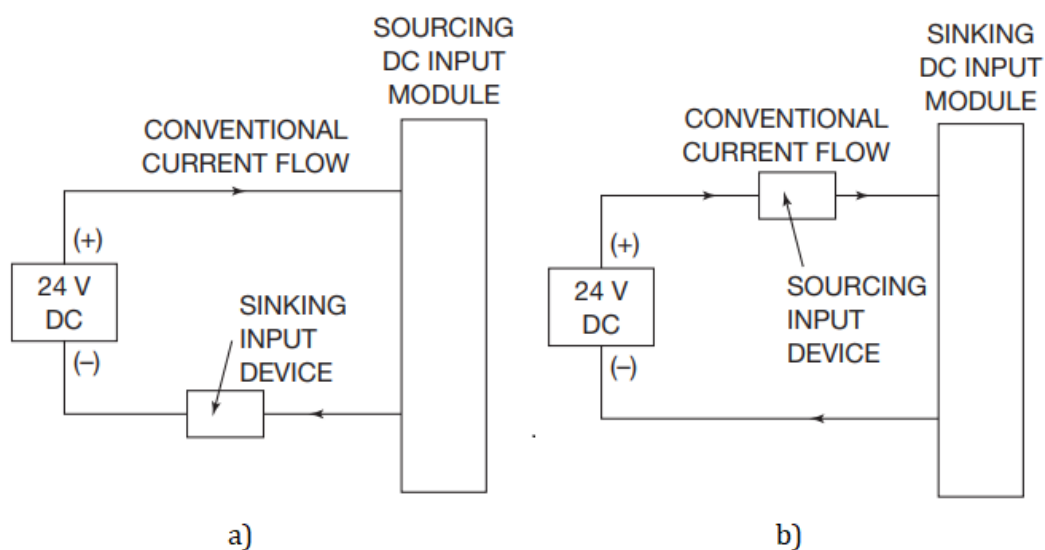
Obr. 1-2: Schéma střídavého digitálního vstupu. [2]

Na Obr. 1-3 je zobrazeno schéma zapojení stejnosměrného digitálního vstupu. Jelikož zde spínané zařízení připojené k stejnosměrnému napětí, není zapotřebí usměrňovacího můstku. Princip činnosti stejnosměrného digitálního vstupu je však stejný, jako v dříve popsaném střídavém digitálním vstupu.



Obr. 1-3: Schéma zapojení stejnosměrného digitálního vstupu. [2]

U digitálních vstupů je potřeba rozlišovat, zda jsou připojené periferie typu *sink* (příjemce) nebo typu *source* (zdroj). Pokud je periferie připojena mezi vstupní modul a kladnou svorku zdroje napětí, pak prochází proud z této periferie do vstupního modulu a periferii označujeme jako *sourcing* (zdrojovou), protože je zdrojem proudu pro vstupní modul. Vstupní modul přijímá proud z periferie, proto by v tomto případě byl označen jako *sinking* (přijímací). Pokud je periferie připojena mezi zápornou svorku zdroje napětí a vstupní modul, pak prochází proud ze vstupního modulu do této periferie a proto ji označujeme jako *sinking*. Vstupní modul dodává proud do vstupní periferie, proto bude označován jako *sourcing*. Toto je zobrazeno na Obr. 1-4.



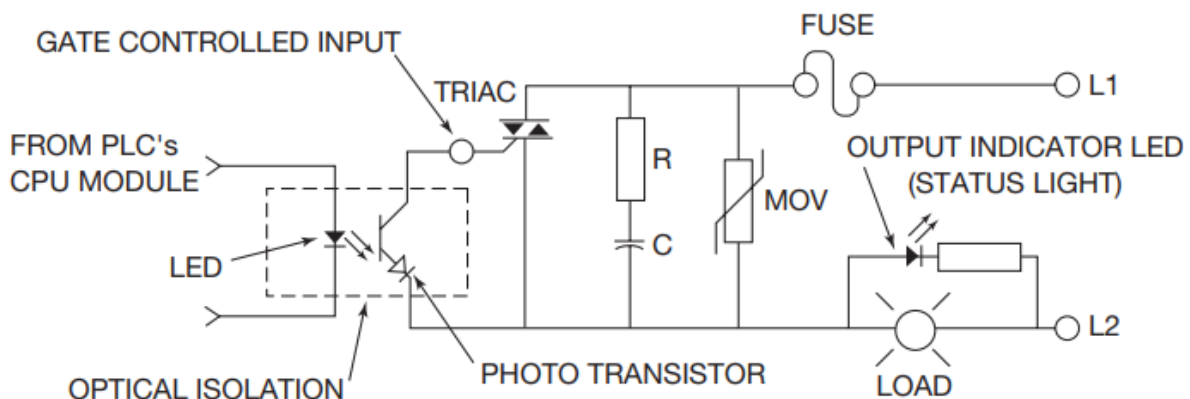
Obr. 1-4 Rozdíl mezi sinking a sourcing periferiemi u vstupních modulů. Za a) vlevo je vstupní module typu *sourcing*, za b) je vstupní modul typu *sinking*. [2]

### 1.1.3.2 Digitální (binární) výstupy

Pomocí digitálních výstupů ovládá PLC řízený proces. Pomocí digitálních výstupů se například budí cívky relé a stykačů, spínání větráků, alarmů, světel atd. Tyto zařízení požadují různé typy a hodnoty výstupního napětí. Cílem výstupního modulu je tedy přeměnit napětí 5 V DC, které je v procesoru na požadované výstupní napětí. Typické hodnoty výstupního napětí bývají 24, 48, 120 a 240 V AC i DC.

Existují 3 typy digitálních výstupů:

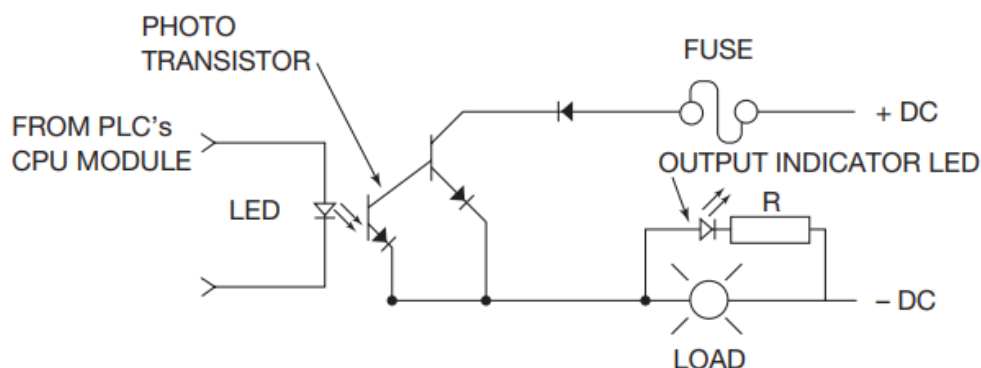
- **Reléový výstup:** u tohoto typu výstupu je použit výstupní signál z procesoru PLC k tomu, aby sepnul relé. S pomocí relé, které je zabudované přímo v PLC, je procesor schopný spínat a rozpínat výstupní proudy až řádu jednotek ampér. Relé zároveň izoluje procesor od výstupního obvodu. Reléové výstupy jsou vhodné jak pro spínání střídavých, tak stejnosměrných obvodů. Zároveň jsou schopné odolávat vysokým napěťovým špičkám a proudovým rázům. Jejich nevýhodou je pomalejší spínání, než např. u tranzistorových výstupů.
- **Tranzistorový výstup:** k sepnutí výstupního obvodu používá tranzistor. Výhodou tranzistorů je velmi rychlé sepnutí. Pomocí tranzistoru však nelze spínat střídavé obvody a tranzistor lze zničit připojením příliš vysokého napětí opačné polaroty nebo příliš velkým proudem. Proto je použita vestavěná elektronická ochrana nebo je použita pojistka. K izolování výstupního obvodu od procesoru je použit optočlen.
- **Výstup s triakem:** výstup s triakem se používá, pokud potřebujeme pomocí digitálního výstupu spínat střídavý obvod. K izolaci se také používá optočlen. Triaky se používají k spínání pouze střídavých obvodů a jsou jednoduše zničeny při příliš vysokých proudech. Proto se k ochraně triaků vždy v obvodu vyskytují pojistky.



Obr. 1-5 Schéma zapojení digitálního výstupu s triakem, pro střídavé obvody. [2]

Na Obr. 1-5 je zobrazeno schéma zapojení digitálního výstupu určeného k spínání střídavých obvodů. Ve chvíli, kdy procesor na základě uživatelského programu rozhodne, že je potřeba

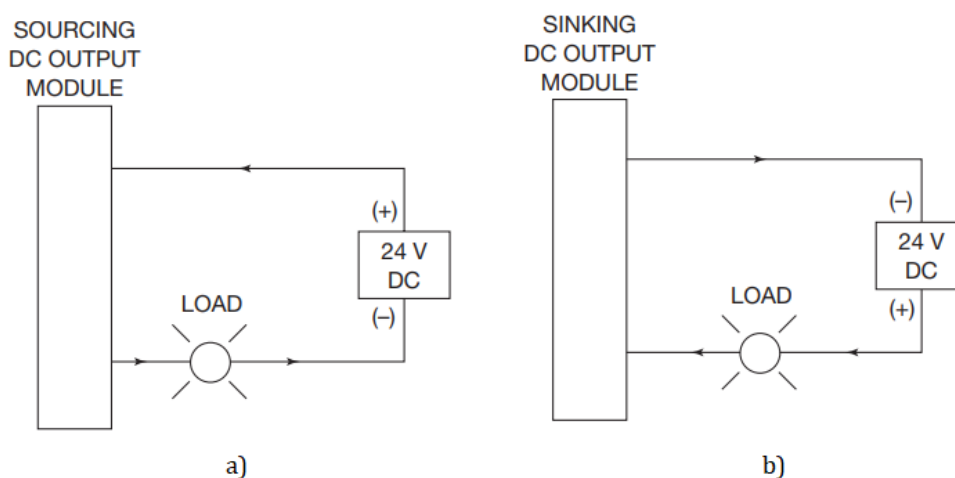
sepnout výstup, dojde k sepnutí vnitřního obvodu a světlo z LED diody uvnitř optočlenu způsobí, že začne vést foto-tranzistor uvnitř optočlenu. Díky tomu začne vést dostatečný proud na řídicí elektrodu triaku a triak sepne. RC člen a metal-oxidový varistor (MOV) jsou v obvodu proto, aby ochránily triak před napěťovými špičkami a elektrickým rušením.



Obr. 1-6 Schéma zapojení digitálního výstupu s tranzistorem, pro stejnosměrné obvody. [2]

Na Obr. 1-6 je zobrazeno schéma zapojení digitálního výstupu určeného k spínání stejnosměrných obvodů. Zapojení je podobné předchozímu zapojení, na rozdíl od triaku zde však spíná výstupní obvod tranzistor. Stejně jako triak, je tranzistor sepnut pomocí fototranzistoru z optočlenu. Pokud je na tranzistor přivedené napětí a fotodiodou optočlenu prochází proud, dojde k sepnutí fototranzistoru optočlenu a začne proudit proud na řídicí elektrodu tranzistoru v digitálním výstupu, čímž dojde k sepnutí výstupu. Stejně jako triaky, tranzistory jsou náchylné na příliš velké napětí a na rázové proudy. Tranzistory jsou chráněny diodou na rozdíl od triaku.

Stejně jako u vstupních modulů, tak i u výstupních modulů je potřeba rozlišovat, zda je modul typu *sourcing* nebo typu *sinking*. Princip je stejný, jako u vstupních modulů. Oba typy digitálních výstupů jsou zobrazeny na Obr. 1-7.



Obr. 1-7 Rozlišení výstupních modulů typu sinking a typu sourcing. Za a) vlevo je výstupní module typu *sourcing*, za b) vpravo je výstupní module typu *sinking*. [2]



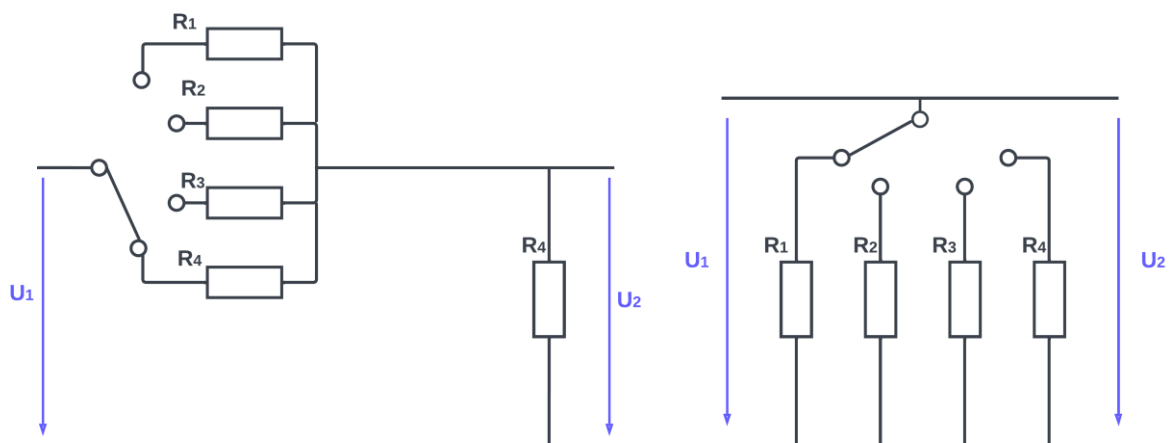
### 1.1.3.3 Analogové vstupy

Analogové vstupy zpracovávají analogové signály, tedy vnější spojitě signály elektrických veličin proudu a napětí. Na rozdíl od digitálních periférií se jedná o složitější řešení zpracování signálu. Analogové signály reprezentují informaci o hodnotě, kterou nelze pomocí binárních signálů vyjádřit. Protože CPU PLC pracuje pouze s diskrétními hodnotami, je potřeba tyto analogové signály převést na diskrétní číselné vyjádření, které bude reprezentovat hodnotu těchto spojitých signálů. Konverzi analogového signálu na signál digitální zajišťuje analogově-digitální převodník (AD převodník).

Analogové signály se nejčastěji využívají k měření fyzikálních veličin, jako je například teplota, tlak, průtok, poloha nebo vlhkost. Tyto veličiny jsou převedeny na veličiny elektrické (proud a napětí) pomocí snímacích členů (čidel). Odvozená veličina, kterou lze také měřit je odpor.

Analogové signály mají vždy nějaký rozsah, pomocí kterého se určí hodnota měřené veličiny. Analogové signály mohou být unipolární, tedy nabývat pouze kladných hodnot, nebo bipolární, tedy nabývat jak kladných, tak záporných hodnot. Často používané rozsahy jsou například 0-10 V, -10-10 V a 0-5 V DC pro napětí a 4-20 mA, 0-20 mA a -20-20 mA pro proud.

Čip vstupního AD převodníku má fyzicky jeden univerzální vstup pro všechny veličiny a obvykle má rozsah  $\pm 1$  V nebo 0-1 V. Typ, rozsah a způsob měření se určuje vstupním analogovým obvodem, realizovaným odporovým děličem. Ten obsahuje kombinaci odporů tak, aby bylo možné předpokládaný rozsah analogového signálu upravit na hodnotu, kterou je AD převodník schopný zpracovat. Měření proudu je také realizováno měřením napětí na zatěžovacím odporu vstupního obvodu. Odporové děliče k měření napětí a proudové bočníky k měření proudu jsou zobrazeny na Obr. 1-8.



Obr. 1-8 Odporový dělič s přepínáním rozsahů pro měření napětí (vlevo) a odporové bočníky s přepínáním rozsahu pro měření proudu. [9]

Převod analogového signálu na digitální probíhá pomocí dvou kroků. Prvním krokem je vzorkování v čase. Používá se vzorkování s konstantní periodou. Při tomto vzorkování ale dochází ke ztrátě informace, protože nevíme, co se dělo s analogovým signálem mezi vzorky. Je potřeba také zmínit tzv. Nyquistův-Shannonův vzorkovací teorém, který říká, jak velkou je potřeba zvolit periodu vzhledem k maximální frekvenci obsažené v signálu, aby ze vzorků bylo možné signál zpět rekonstruovat. Tento vzorkovací teorém je popsán vztahem:

$$f_{vz} > 2 \cdot f_{max} [s^{-1}] \quad (1-1)$$

přičemž  $f_{vz}$  je vzorkovací frekvence a  $f_{max}$  je maximální frekvence obsažená v signálu. Vzorkovací periodu dostaneme pomocí vztahu:

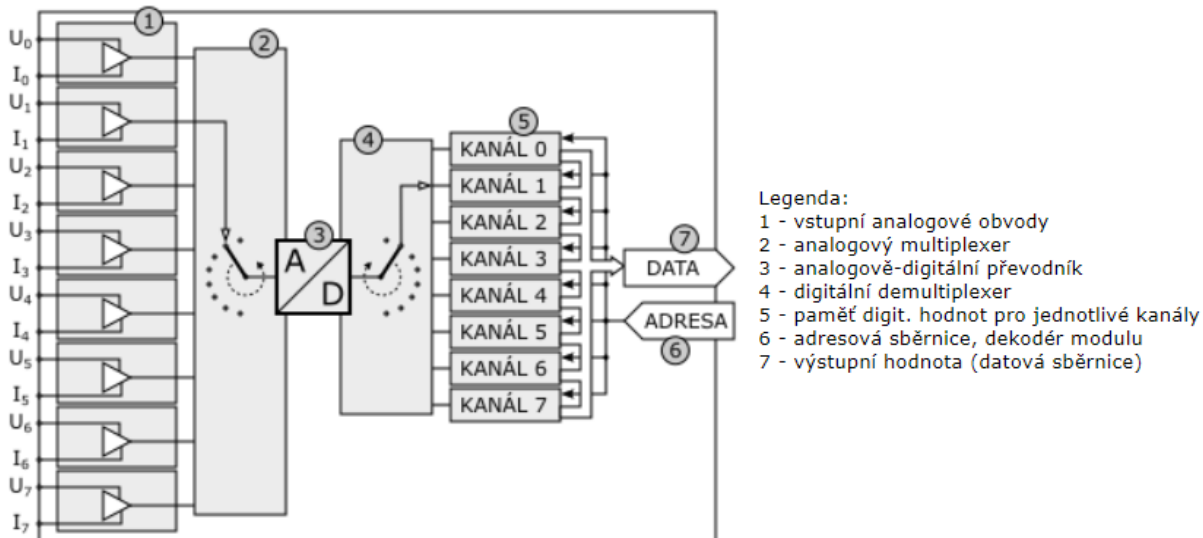
$$\frac{1}{T_{vz}} = f_{vz} [s] \quad (1-2)$$

kde  $T_{vz}$  je vzorkovací perioda. Vztah (1-1) říká, že pokud chceme být schopni zrekonstruovat ze vzorků signál, musí být vzorkovací frekvence více než dvakrát vyšší než maximální obsažená frekvence.

Vzorkování analogového signálu je složitější a časově náročnější než vzorkování digitálního signálu a je tedy obtížné se přiblížit době cyklu PLC tak, aby byla vždy načtena aktuální hodnota analogového signálu. Tento problém lze řešit například tak, že CPU pošle přímý požadavek na periférii ke zjištění analogové hodnoty. Periferie provede převod a pošle aktuální hodnotu do CPU, avšak po dobu převodu je nutné zastavit program v CPU. Proto existuje ještě druhý přístup k řešení problému, kterým je cyklické čtení hodnoty analogového signálu. V tomto případě není zatěžována doba cyklu PLC, avšak není zaručeno, že přečtená hodnota analogového signálu je aktuální.

Druhým krokem převodu analogového signálu na digitální je kvantování v amplitudě. Kvantování je proces, kdy jsou naměřené hodnoty vzorků zaokrouhleny na předem definované kvantizační hladiny. Počet těchto kvantizačních hladin (a tedy i přesnost převodníku) je dán počtem bitů převodu. Analogový signál bývá v typicky reprezentován *integerem*, což je typicky 16 bitů. To znamená, že je možné využít  $2^{16}$  kvantizačních hladin. V praxi se však často z důvodů nižší ceny analogového modulu nepoužívá všech 16 bitů, ale jen 12, 10 nebo 8, což dále ovlivní přesnost AD převodníku. Přesnost je závislá i na amplitudě, pokud budeme mít vstupní analogový signál v rozmezí 0-5 V, bude přesnost 2x vyšší, než když bude vstupní rozsah 0-10 V.

Moduly analogových vstupů typicky neobsahují pouze jeden analogový vstup, ale více analogových vstupů, např. 2,4 nebo 8. Z rozměrových a finančních důvodů bývá v analogovém modulu pouze jeden AD převodník. Výhodou je, že pak stačí před AD převodník zapojit analogový *multiplexer* (levnější cena) a můžeme pomocí jednoho AD převodníku zpracovat signál několika analogových vstupů. Nevýhodou je, že najednou lze převádět pouze jeden signál. Proto jsou občas AD převodníky v modulu dva. Na Obr. 1-9 je zobrazeno blokové schéma AD převodníku pro 8 kanálů.



Obr. 1-9 Principiální blokové schéma AD převodníku pro 8 kanálů analogových vstupů. [9]

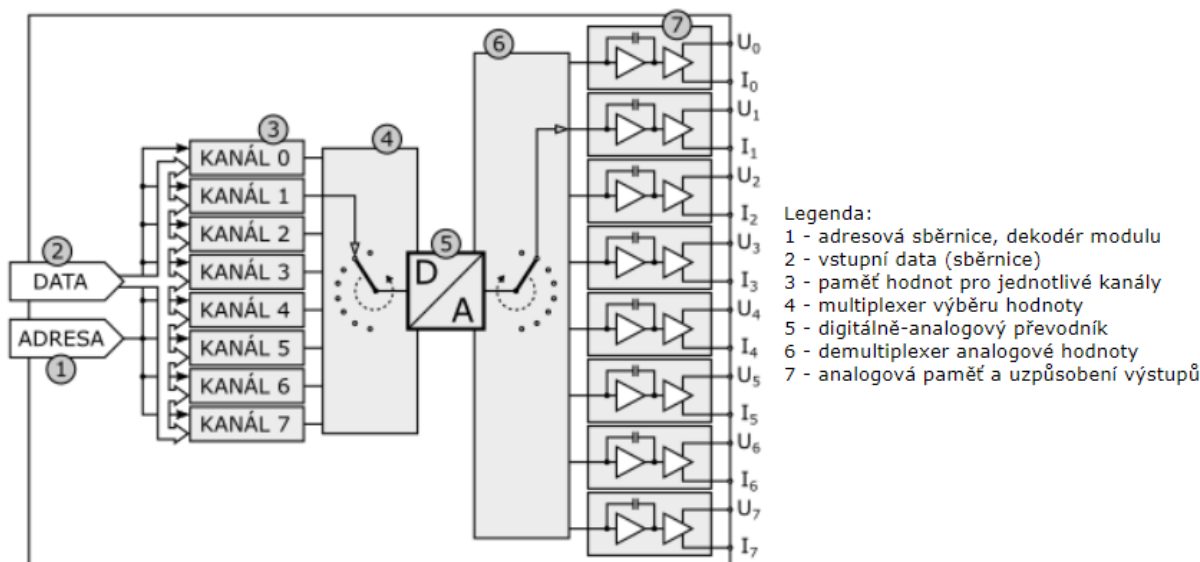
Analogové signály jsou přiváděny do vstupních obvodů (1) jednotlivých kanálů, kde jsou přizpůsobeny ke zpracování v převodníku. Analogový *multiplexer* (2) postupně připojuje jednotlivé kanály k analogově-digitálnímu převodníku, který analogovou hodnotu převede na digitální. Následující *demultiplexer* (4) přiřadí přivedenou digitální hodnotu do mezipaměti kanálu (5), pro který se provádí převod. PLC potom přistupuje k hodnotám jednotlivých kanálů tak, že nastaví adresu modulu a adresu kanálu a na datové sběrnici obdrží číselnou hodnotu konkrétního analogového signálu.

#### 1.1.3.4 Analogové výstupy

Analogové výstupy jsou používány k řízení technologických procesů s použitím vhodných akčních (výkonových) prvků. Jejich hodnota může reprezentovat například požadované otáčky motoru pro frekvenční měnič, výkon regulovaného hořáku, nastavení proporcionálního ventilu atp. Analogový vstup tedy převádí interní číslicovou hodnotu z PLC programu na spojitý elektrický analogový signál charakteru napětí nebo proud. Převod signálu z digitálního na analogový zajišťuje digitálně-analogový převodník (DA převodník). Typické rozsahy analogových výstupů jsou stejné jako u analogových vstupů, tedy 0–10 V, -10-10 V a 0-5 V DC pro napětí a 4-20 mA, 0-20 mA a -20-20 V DC pro proud. Čip výstupního analogového převodníku má fyzicky jeden napěťový a jeden proudový výstup. Podle konfigurace modulu je výstup DA převodníku upraven pro napěťový nebo proudový výstup. V obou případech je však proudově posílen. Stejně jako u analogových vstupů obsahuje z rozměrových a finančních důvodů analogový výstupní modul pouze jeden DA převodník, výstup tohoto DA převodníku je přiřazen požadovanému výstupnímu analogovému signálu pomocí demultiplexoru.

Jsou dva přístupy zapisování hodnoty požadovaného výstupu do analogové periferie. Buď se zapisuje pouze tehdy, pokud dojde ke změně požadované výstupní hodnoty, nebo cyklicky

v každém PLC cyklu. Ani jeden z přístupů však není schopný zaručit, že v aktuálním cyklu PLC bude výstupní analogová hodnota aktualizována. Výstupní analogový modul provádí vlastní cyklus převodu číselné hodnoty na analogovou. Jednotlivé změny požadovaných hodnot jsou ukládány do mezipaměti jednotlivých kanálů a zpracovány vnitřním cyklem modulu až na ně přijde řada. Na Obr. 1-10 je zobrazeno blokové schéma DA převodníku obsaženého v analogově výstupním modulu. CPU (podle programu) nastaví na datové sběrnici (2) hodnotu, která se má převést na analogovou a požadovanou adresu (1) analogového výstupu. Adresa se skládá z adresy modulu a z adresy kanálu. Hodnota je řídicím signálem zapsána do požadovaného modulu a požadovaného kanálu modulu. Vnitřní logika modulu potom cyklicky postupně vybírá přes *multiplexer* (4) digitální hodnoty jednotlivých kanálů a ty jsou převedeny na signál analogový v DA převodníku (5). Analogová hodnota je *demultiplexerem* (6) zapsána do odpovídající analogové paměti (7), podle toho, pro který kanál se převod provádí. Takto se postupně zpracují hodnoty pro kanál 0-7 a cyklus se opakuje.



Obr. 1-10 Principiální blokové schéma modulu DA převodníku pro 8 kanálů analogových výstupů. [10]

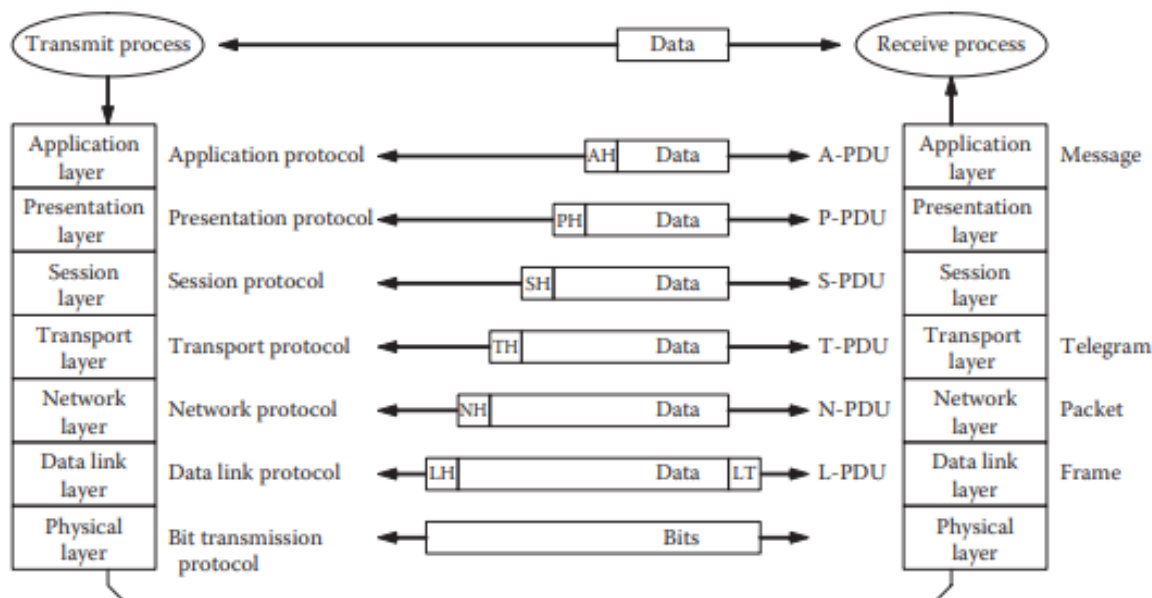
#### 1.1.4 Komunikační rozhraní

K tomu, aby bylo možné automatizovat požadovaný výrobní proces je potřeba nejen řídicích zařízení – typicky PLC, ale také mnoha dalších zařízení – od spínacích prvků, senzorů, ventilů, HMI panelů, motorů, měničů atd. Aby spolu všechna tyto zařízení mohla spolupracovat, je potřeba zařídit vzájemnou komunikaci a předávání potřebných dat. Komunikace mezi zařízeními probíhá po průmyslových sběrnících. Průmyslové sběrnice vznikly za účelem zjednodušení komunikace jednotlivých zařízení s řídicím zařízením a také proto, že je potřeba splnit požadavky extrémních průmyslových podmínek. Komunikace probíhá za použití průmyslových komunikačních protokolů, které splňují požadavky výrobních procesů. Průmyslový protokol je sada pravidel

určujících parametry přenosu dat mezi zařízeními. Průmyslových protokolů je mnoho, mezi nejznámější patří například Profibus, AS-Interface, CAN, HART, Profinet, Interbus a další. Protože v této práci je použit průmyslový protokol Profinet, budu se podrobněji věnovat pouze Profinetu a Ethernetu, což je protokol, ze kterého Profinet vychází.

#### 1.1.4.1 Ethernet

Ethernet je název souhrnu technologií pro počítačové sítě (LAN, MAN), které používají kabely s kroucenou dvojlínkou, optické kabely a koaxiální kabely pro komunikaci přenosovými rychlostmi v rozmezí 1Mbit/s až 400 Gbit/s. První verze Ethernetu byla vyvinuta v letech 1972-1975 firmou Xerox s počáteční přenosovou rychlostí 3 Mbit/s. V roce 1982 došlo k vytvoření IEEE standardu 802.3, který popisoval specifikace fyzické a linkové vrstvy tzv. referenčního modelu ISO/OSI, což je obecný rámec pro návrh protokolů distribuovaných systémů, do nějž jsou zasazována jednotlivá konkrétní řešení.



Obr. 1-11 Struktura vrstev referenčního ISO/OSI modelu. [11]

Referenční ISO/OSI model je definován standardem ISO 7498-1, první vydání je z roku 1984. Referenční model definuje sedmivrstvý protokolový zásobník. Jednotlivé vrstvy poskytují daný typ služeb a interakce mezi vrstvami je definována. Uspořádání vrstev je hierarchické, vrstva N poskytuje služby vrstvě N+1 a využívá služeb vrstvy N-1. Mezi odpovídajícími vrstvami různých uzlů tak vznikají virtuální kanály toku informací. Na Obr. 1-11 je zobrazeno všech 7 vrstev a jejich hierarchická struktura. Zároveň je zde zachycen princip přenosu dat podle referenčního modelu. Nejprve se vysílací zařízení rozhodne, jaká data chce poslat. Následně jsou data postupně zapouzdřována, jak jdou z nejvyšší aplikační vrstvy až na fyzickou vrstvu, po fyzické vrstvě, tedy přenosovém mediu, dojde k fyzickému přenosu dat až do zařízení se správnou fyzickou adresou,

kde jsou data postupně předávána vyšším vrstvám, až se dostanou do poslední aplikační vrstvy, kde dojde ke zpracování dat přijímacím zařízením.

První vrstva se nazývá fyzická a jak název napovídá, tato vrstva popisuje všechny mechanické, fyzické, optické, elektrické a logické vlastnosti komunikačního systému. Patří sem například definice typu konektoru, impedance, přenosové frekvence, přípustné délky vedení, typy vedení, typ kódování jednotlivých bitů, souběžný přenos energie a dat a podobně. Typické standardy fyzické vrstvy jsou například RS-232, RS-485, IEEE 802.3, IEEE 802.11 a další. Fyzická vrstva se zabývá výhradně přenosem bitů.

Druhá vrstva se nazývá spojová (anglicky *data link layer*) a jejím úkolem je zaručit přenos dat mezi dvěma uzly sítě. To zahrnuje nejprve vytvoření datových rámců (anglicky *frames*), což jsou celé bloky dat. Druhým úkolem datové linkové vrstvy je kódování a kontrola rámce (např. pomocí cyklického redundančního součtu (*Cyclic redundancy check* – CRC), aby bylo možné rámeček detekovat chyby v přenesených datech a případně i chyby opravit. Do druhé vrstvy patří také kontrola časových limitů, popř. ověření, zda byly z druhé strany přijaty odpovídající odpovědi a potvrzení.

Třetí vrstva se nazývá síťová (anglicky *network layer*) a stará se o směrování v síti. Poskytuje spojení mezi systémy, které spolu přímo nesousedí a zajišťuje doručení bloků dat, nazývaných pakety (anglicky *packets*), až ke konečnému adresátovi. Na této vrstvě pracují směrovače (routery) a nejčastěji používaným protokolem je protokol IP.

Čtvrtá vrstva se nazývá transportní (anglicky *transport layer*) a zajišťuje přenos dat mezi koncovými uzly. Jejím účelem je poskytnout takovou kvalitu přenosu, jakou požadují vyšší vrstvy. Vrstva nabízí spojově (TCP) a nespojově orientované (UDP) protokoly. TCP protokol zajišťuje, že při přenosu nedojde ke ztrátě žádných paketů, což některé aplikace vyžadují. UDP protokol zajišťuje přenos dat bez záruky toho, že nedojde ke ztrátě paketu. Tento protokol se používá u aplikací, u kterých není žádané zdržení způsobené čekáním na to, až se bezchybně přenesou všechny pakety. Jak protokol UDP, tak TCP jsou funkční pouze pokud na 3. síťové vrstvě je použit protokol IP.

Pátá vrstva se nazývá relační (anglicky *session layer*) a zajišťuje sestavení, řízení a zrušení relací pro spojovanou komunikaci. Dále také zajišťuje synchronizaci, šifrování anebo kompresi dat.

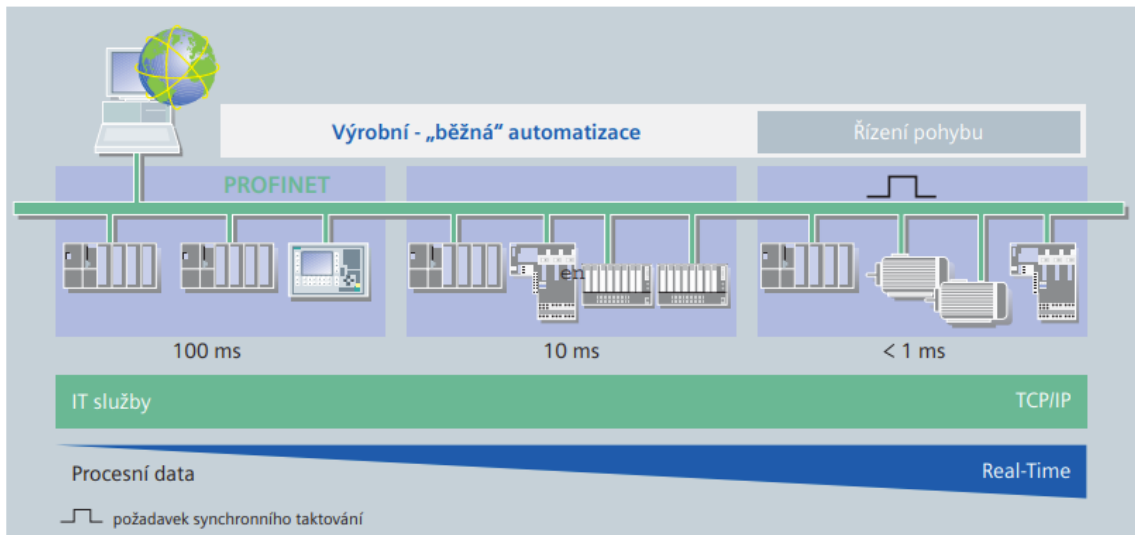
Šestá vrstva se nazývá prezentační (anglicky *presentation layer*) a stará se o převedení dat z relační vrstvy do takového stavu, kterému bude aplikační vrstva rozumět. Tato vrstva je potřeba, protože stejná posloupnost bitů může mít pro příjemce jiný význam než pro odesílatele. To například kvůli kódování znaků (ASCII, EBCDIC, ...), formátu čísel (malý a velký endian), formátu polí, struktur anebo třeba kvůli ukazatelům.

Sedmá vrstva se nazývá aplikační (anglicky *application layer*) a jedná se o nejvyšší vrstvu referenčního modelu, se kterou pracuje koncový uživatel. Původní myšlenkou bylo, že tato vrstva bude obsahovat aplikace. Aplikací je však příliš mnoho a musely by být všechny standardizovány, což je časově nereálné a nepraktické. Proto se přešlo k tomu, že aplikační vrstva bude obsahovat pouze „jádro“ aplikací, které má smysl standardizovat.

Ethernet realizuje pouze fyzickou a spojovou vrstvu referenčního modelu, což znamená, že je možné provozovat více komunikačních protokolů síťových a vyšších vrstev. Standardní Ethernet byl navržen do kancelářského prostředí a není optimalizovaný pro náročné průmyslové prostředí. Lze tedy zavést pojem průmyslový Ethernet, což je vlastně použití Ethernetu v průmyslovém prostředí v kombinaci s protokoly, které zajišťují determinismus a umožňují řízení procesů v reálném čase (*real-time control*). Zároveň jsou první dvě vrstvy referenčního modelu přizpůsobené náročným podmínkám, časté je například použití stíněných kabelů a speciálních konektorů. Protokoly, které lze řadit mezi průmyslový ethernet jsou například EtherCat, EtherNet/IP, Powerlink anebo třeba Profinet.

#### 1.1.4.2 Profinet

Profinet je univerzální komunikační síť na bázi Ethernetu (IEEE 802.3), která zpřístupňuje zavedený standard Ethernet z kancelářského světa všem oblastem automatizační techniky a zároveň navazuje na zkušenosti získané během desetiletí v prostředí průmyslových sběrnic. Profinet byl vyvinut firmou PROFIBUS & PROFINET International (PI) a je standardizován normami IEC 61158 a IEC 61784. Struktura Profinetu podporuje modulární koncept, který uživatelům umožňuje vybrat si požadovanou funkcionalitu. Funkce se liší především typem výměny dat v závislosti na tom, jaké požadavky na rychlost přenosu dat konkrétní aplikace vyžadují. Díky možnosti měnit dobu komunikačního cyklu a rozdělení funkcí do tzv. konformačních tříd (CC - *Conformance class*) lze Profinet přizpůsobit dané aplikaci. Profinet je tedy vhodný pro všechny aplikace od standardních procesních úkonů až po náročné pohonařské aplikace. To je zobrazeno na Obr. 1-12, kde je vidět, že komunikační cyklus Profinetu se může lišit v závislosti na aplikaci. U běžných procesů se pohybuje komunikační cyklus v řádech stovek ms až jednotek ms. U náročnějších procesů, jako je například řízení pohybů mohou dosahovat dokonce vyšší desítky  $\mu$ s.



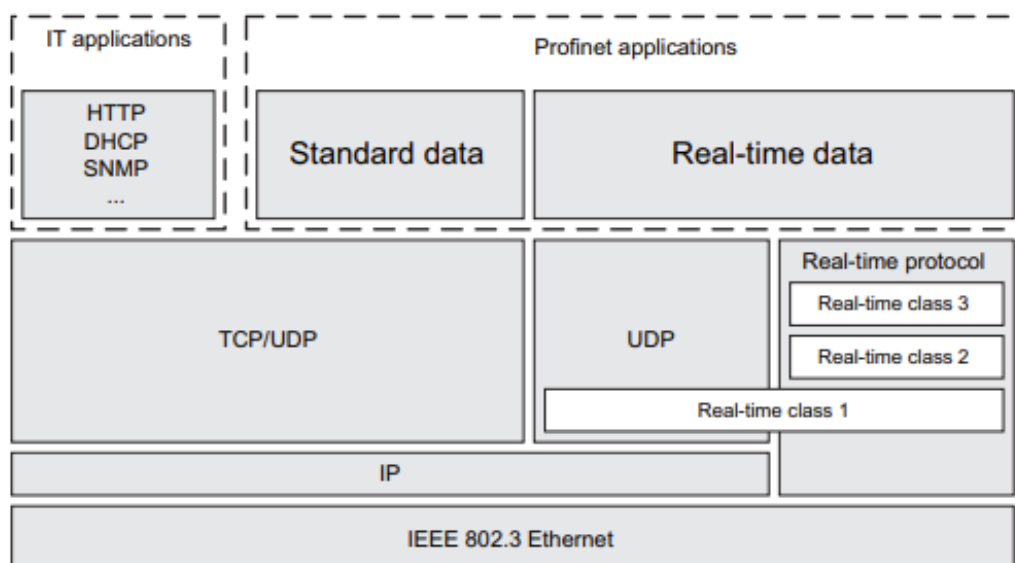
Obr. 1-12 Rozdělení aplikací podle časové náročnosti a délky komunikačního cyklu. [12]

Profinet rozděluje procesy podle časové náročnosti následovně:

- **Časově nekritické (*non-time critical* – NRT):** Jedná se o datové přenosy přes TCP/IP a UDP/IP, které nejsou časově kritické, tzn. většinou komunikačního charakteru, např. parametrizace a konfigurace. Časová odezva se pohybuje v řádech 100 ms. [12]
- **Procesy řízené v reálném čase (*real time* – RT):** Jedná se o časově kritická procesní data, např. cyklická uživatelská data nebo událostmi řízená přerušení. Pro splnění požadavků automatizačních řešení na odezvu v reálném čase disponuje Profinet optimalizovaným RT komunikačním kanálem. Toto řešení výrazně zkracuje časový cyklus a tím zvyšuje výkonnost aktualizace procesních dat. Časová odezva se pohybuje v rozsahu 1-10 ms. [12]
- **Izochronně řízené procesy (*Isochronous real time* – IRT):** Izochronně řízené procesy v reálném čase neboli procesy se synchronním taktováním v reálném čase, jsou procesy specifických aplikací, které vyžadují rychlou odezvu a přesné taktování, např. řízení pohybu. Režim IRT je schopen dosáhnout časového cyklu sběrnice pod 1 ms s časovou nejistotou menší než 1  $\mu$ s, čímž je jasně docíleno deterministického chování systému. Těchto vlastností se dosahuje rozdělením komunikačního cyklu na rezervovanou a otevřenou část. [12]

Všechny tři typy přenosu dat mohou existovat paralelně vedle sebe díky vhodně zvolené architektuře Profinetu. To lze vidět na Obr. 1-13, a také lze vidět, že Profinet se vyskytuje pouze na 1., 2., a 7. vrstvě referenčního modelu, pokud se jedná o RT nebo IRT kanál. Pro zbylou komunikaci jsou také použity 3. a 4. vrstvy díky použitým TCP/IP a UDP/IP protokolům.



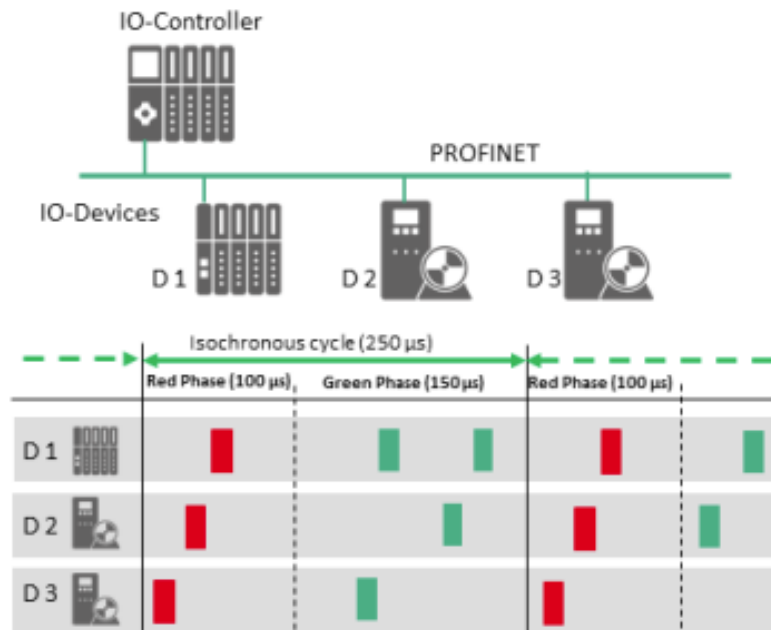


Obr. 1-13 Struktura Profinet komunikace z pohledu referenčního modelu. [13]

Pokud se podíváme konkrétněji jak funguje Profinet IRT, pak vidíme, že komunikační cyklus je rozdělen na rezervovaný (červený) izochronní kanál a na otevřený (zelený) kanál. Během červené fáze cyklu jsou tedy posílány pouze cyklická izochronní data jako synchronizované pakety, což zaručuje determinismus. RT a NRT data mohou být přenášeny pouze během zelené fáze. To je zobrazeno na Obr. -1-14. Aby se koncept izochronní komunikace přenesl do automatizace, kde se využívá mnoho IO zařízení, jsou definovány dva signály, které zajišťují časování rezervované červené fáze:

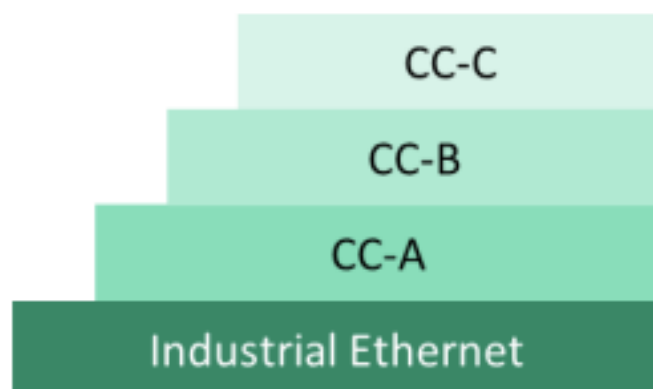
- **T<sub>I</sub>**: V tomto okamžiku musí všichni účastníci izochronní operace uložit své vstupy.
- **T<sub>O</sub>**: V tomto okamžiku musí všichni účastníci izochronní operace zapsat své výstupy.

Protože je veškeré časování definováno na základě časování komunikace, mohou všichni účastníci zachytávat a vypisovat své IO synchronně napříč segmentem IRT. Tímto způsobem je možná vysoce přesná výměna dat s nízkým časovou nejistotou (*jitter*) mezi CPU a provozním zařízením. Vnitřní časování řídicí smyčky řídicí jednotky je tímto způsobem synchronizováno se všemi zařízeními v segmentu. Výsledkem je, že uzavřenou regulační smyčku lze nastavit během jediného cyklu (např. 250  $\mu$ s). [14]



Obr. -1-14 Rozdělení IRT komunikace na rezervovaný červený interval a na otevřený zelený interval. [14]

Profinet nabízí značnou řadu funkcí, které můžeme rozdělit podle již dříve zmiňovaných konformačních tříd (CC). CC jsou tři a jejich struktura je hierarchická, tedy každý vyšší třída poskytuje všechny předchozí funkce služby nižší a rozšiřuje je o další funkce. Struktura CC je zobrazena na Obr. 1-15. Nejnižší třída CC-A nabízí základní funkce pro Profinet s RT komunikací jako je například cyklická výměna IO dat, acyklické čtení a zapisování dat (parametrizace, diagnostika) a také obsahuje model alarmů, který signalizuje chyby zařízení a sítě. CC-A zároveň poskytuje veškeré IT služby bez jakýchkoliv omezení. CC-B rozšiřuje diagnostiku a zároveň zavádí detekci topologie. Topologii může zjistit jakékoliv Profinet zařízení pomocí protokolu SNMP (*simple network management protocol*). CC-C přidává všechny potřebné synchronizační funkce a umožňuje IRT komunikaci. [14] [11]



Obr. 1-15 Struktura konformačních tříd. [14]

V Profinet síti dělíme zařízení, která se zde mohou vyskytovat na 3 typy:

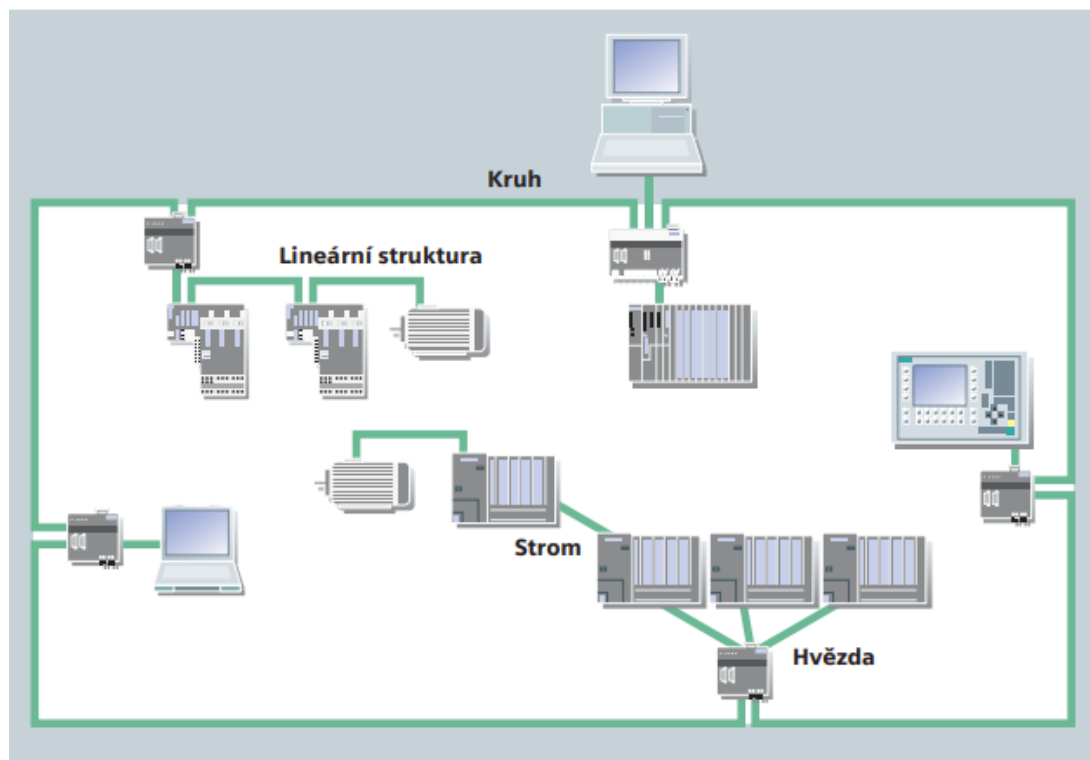
- **IO controller:** Jedná se o řídicí jednotku vstupů a výstupů, ve které je prováděn program automatu, typicky tedy PLC.
- **IO device:** Jedná se o řízenou jednotku IO, tedy o distribuované jednotky v technologickém provozu, přidělené k danému IO controlleru.
- **IO supervisor:** Jedná se o dohlížející jednotku IO, tedy programovací jednotka nebo PC s funkcemi k uvádění do provozu nebo diagnostickými funkcemi.

Co se topologie sítě týče, Profinet poskytuje 4 možné topologie, které lze podle potřeby kombinovat. Topologie lze provozovat jak na metalických (měděných) tak na optických kabelech.

Možné topologie jsou:

- **Hvězda:** Jedná se o topologii s centrálním přepínačem, který má spojení ke všem koncovým zařízením v síti.
- **Strom:** Strom je vlastně struktura vícenásobných hvězd (hvězda ve hvězdě) a umožňuje kombinovat kabeláž měděných a optických kabelů.
- **Lineární struktura:** Je vhodná pro rozlehlé systémy, jako například dopravníkové pásy.
- **Kruh:** Kruhová struktura vznikne spojením konců lineární struktury. Používá se, pokud je u použitých zařízení kladen důraz na vysokou spolehlivost, jelikož je odolná vůči přerušení spojení (pouze na jednom místě) anebo výpadku jedné komponenty.

Jednotlivé topologie lze vidět na Obr. 1-16.



Obr. 1-16 Možné topologie Profinetu. [12]

Profinet lze provozovat buď po měděných metalických kabelech, optických kabelech anebo bezdrátově. Měděné kabely jsou vždy o průřezu AWG 22 a jejich maximální povolená délka mezi dvěma zařízeními je maximálně 100 m. Optické kabely se používají ve variantě 1 mm při použití polymerových optických vláken. Jejich výhodou oproti měděným kabelům je například odolnost proti elektromagnetickému rušení (EMC). Co se konektorů týče, oproti kancelářským ethernetovým konektorům jsou v průmyslu použity speciální konektory, které jsou odolnější vůči okolnímu prostředí. Podle použité aplikace lze pro měděný kabel zvolit dva typy konektoru. Prvním typem je konektor RJ-45, který vychází ze standardu IEEE 802.3 pro průmyslový Ethernet, a to ve dvou provedeních z hlediska krytí – IP 20 a IP 65/IP 67. Druhým typem je M8 nebo M12 konektor, který je šroubového typu. Pro optické kabely jsou nejčastěji používány SCRJ konektory, a to jak pro vnitřní prostředí s krytím IP 20, tak venkovní prostředí s krytím IP 65 nebo IP 67.

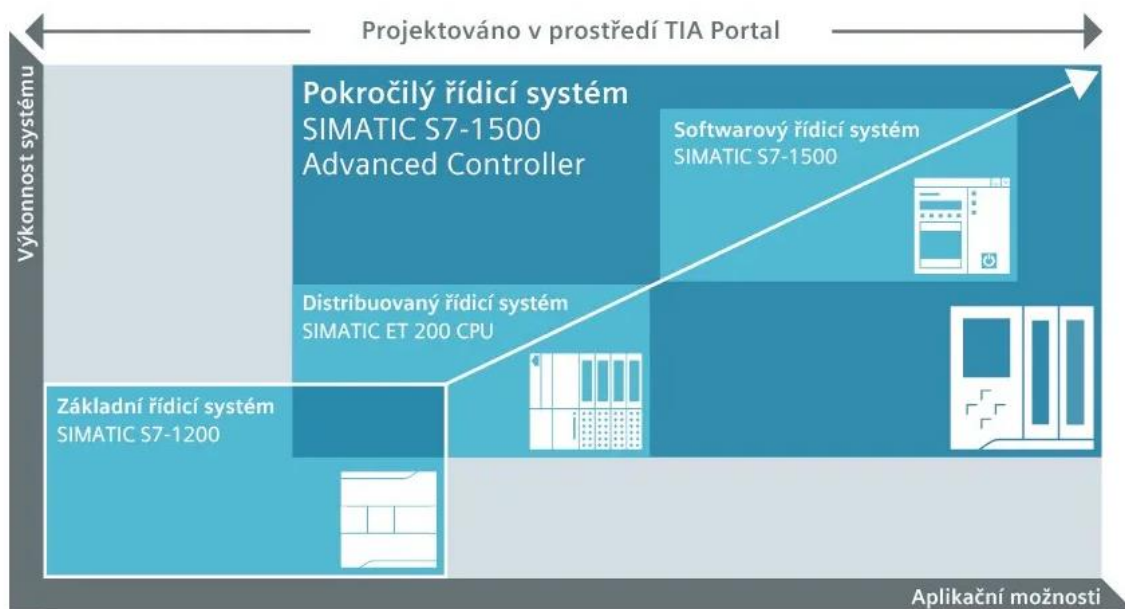
Profinet lze provozovat také pomocí bezdrátového přenosu. Bezdrátový přenos představuje výhody nejen kvůli úspoře nákladů na kabeláž, ale také umožňuje připojení tam, kde by to jinak nebylo možné ať už z mechanických, bezpečnostních nebo environmentálních omezení. Příkladem mohou být například senzory s obtížnou dostupností, autonomní pohyblivé systémy bez řidiče apod. Bezdrátová verze Profinetu musí být schopná fungovat s různými radiovými technologiemi pro různé aplikace se specifickými parametry jako např. rychlost přenosu, dosah nebo počet uzlů. Profinet je schopný přenášet bezdrátové připojení pomocí standardu IEEE 802.11 Wi-Fi nebo prostřednictvím Bluetooth podle standardu IEEE 802.15.1. [15]

## 1.2 Siemens PLC

Řídicí systémy firmy Siemens se nazývají SIMATIC. Tato produktová řada je na trhu již od roku 1958, kdy byl představen první řídicí systém SIMATIC, pojmenovaný SIMATIC G. G proto, že se jedná o řídicí obvod s germaniovými tranzistory. V roce 1964 následoval SIMATIC N, což byl řídicí a spínací obvod, který využíval křemík. V roce 1973 přišlo první PLC, pojmenované SIMATIC S3. Následovala pátá generace systému SIMATIC, SIMATIC S5. Nejnovější generací je systém SIMATIC S7, která je na trhu od roku 1994. Zpočátku obsahovala 3 typy produktů – S7-200, S7-300 a S7-400. V roce 2009 se rozšířilo portfolio o mikrosystém S7-1200, který měl především řešit určité technické komplikace u řídicích jednotek S7-300. Přesto se však za nástupce systému S7-300 nepovažuje S7-1200, ale až S7-1500, který se poprvé objevil na trhu v roce 2013. Jedná se o řídicí systém, který obsahuje mnoho pokročilých funkcí a vyniká velikou výkoností systému. Zároveň přináší nová řešení vizualizace a ovládání pohybu. [16], [17]

## 1.2.1 Portfolio systémů SIMATIC

Siemens nabízí široké spektrum PLC v závislosti na požadavcích. Portfolio SIMATIC PLC obsahuje řídicí systémy základní, pokročilé, distribuované a softwarové. Toto je zobrazeno na Obr. 1-17.



Obr. 1-17 Portfolio řídicích systémů SIMATIC. [18]

Základní řídicí systémy, kam patří S7-1200 jsou vhodné pro automatizační úlohy malého až středního rozsahu. Lze je pořídit ve standardním anebo bezpečnostním provedení. V případě potřeby lze tyto systémy rozšířit vhodnými rozšiřujícími moduly. [19]

Pokročilé řídicí systémy, kam patří S7-1500 a starší produkty S7-300 a S7-400, jsou určeny pro náročné či rozsáhlé aplikace. Slouží k řízení celých výrobních procesů, ale také k řízení aplikací, které vyžadují vysokou výkonnost, flexibilitu a komunikační schopnosti. Sofistikované a náročné aplikace k řízení polohy jsou implementovány s pomocí technologického CPU. Tyto systémy mají extrémně krátkou odezvu a je možné s nimi dosáhnout vysoké rychlosti zpracování signálů. [19]

Distribuované řídicí systémy jsou použité pro stroje s distribuovanou architekturou a pro sériové výrobní stroje s omezenými prostorovými možnostmi. Kombinují výhody SIMATIC S7-1500 a design u SIMATIC ET 200SP. Pokud je potřeba použít řídicí systém ve venkovním prostředí, lze použít ET 200pro s ochranou třídou IP65/IP67. [19]

Softwarové řídicí systémy se používají všude, kde se vyžaduje maximální přesnost, rychlost a počítačová automatizace. Tato řešení založená na platformě PC fungují nezávisle na operačním systému. [19]

Kromě možnosti rozšíření nabízí každá řídicí jednotka i integrované systémové funkce, jako je např. účinné projektování, vysoký výkon, inovativní design, spolehlivá diagnostika a funkce „Safety Integrated“, „Technology Integrated“ a „Security Integrated“. Tím je zajištěna vysoká flexibilita návrhu nebo možnost uzpůsobit si automatizační řešení dle potřeby bez nutnosti neustále získávat další poznatky a zkušenosti. [20]

### 1.3 HMI panely

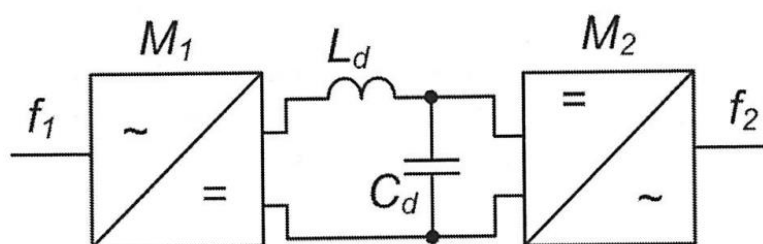
Human machine interface – rozhraní člověk-stroj (HMI) jsou zařízení, která zajišťují interakci mezi lidskou obsluhou a strojem nebo systémem. Dříve byla budována pomocí elektrických a elektromechanických prvků, často v podobě značně komplikovaných a těžkopádných synoptických panelů s tlačítky, spínači, signalizačními LED diodami atd. Tuto technologii však nahradily sofistikovanější zařízení, které používají virtuální rozhraní na CRT nebo LCD obrazovkách v podobě grafických panelů. [21]

HMI panely se začaly používat především proto, že poskytují uživateli názorné a jednoduché rozhraní, pomocí kterého lze řídit daný proces, a zároveň ho monitorovat. HMI panely pracují dohromady s PLC, které řídí proces. HMI posílá PLC informace o požadavcích uživatele, PLC na to následně reaguje podle toho, jak bylo naprogramováno. PLC zároveň posílá HMI informace, které jsou potřeba k monitorování a grafickému znázornění výrobního procesu. Typicky jsou HMI panely spojené s PLC nějakou formou průmyslového ethernetu, v této diplomové práci jsou spojeny pomocí komunikace Profinet, která byla popsána v kapitole 1.1.5.2.

### 1.4 Frekvenční měniče

Frekvenční měniče slouží k přeměně střídavé energie jednoho kmitočtu na střídavou energii jiného kmitočtu. Používají se k rozběhu pohonů asynchronních i synchronních strojů a následnému řízení jejich otáček. Dovolují realizovat pohony s širokým kmitočtovým pásmem s rozmezím v řádech jednotek otáček za sekundu až stovky tisíc otáček za minutu. Dále je lze použít například k indukčnímu ohřevu nebo k spojování nesynchronizovaných střídavých sítí. Měníče frekvence lze dělit na měniče frekvence přímé a nepřímé. V praxi se v automatizaci k řízení pohonů přímé měniče příliš nepoužívají. Frekvenční měnič použitý v této práci je nepřímý měnič frekvence, proto budou popsány nepřímé měniče frekvence.

Nepřímé měniče frekvence jsou složeny z usměrňovače, stejnosměrného obvodu a střídače. Podle stejnosměrného meziobvodu je dělíme na frekvenční měniče s napěťovým meziobvodem a frekvenční měniče s proudovým meziobvodem.



Obr. 1-18 Nepřímé měniče frekvence s napětovým meziobvodem. [22]

Na Obr. 1-18 lze vidět schéma frekvenčního měniče s napětovým meziobvodem. Usměrňovač bývá většinou diodový nebo tyristorový, avšak v takovém případě odebírá ze sítě nesinusové proudy. Stejnoseměrný meziobvod obsahuje kondenzátor s velkou kapacitou na vstupu střídače, což zajišťuje malou impedanci zdroje pro střídač a díky tomu lze úbytky na vstupní impedanci zanedbat a vstupní napětí považovat za čistě stejnosměrné. Tlumivka mezi usměrňovačem a střídačem impedančně odděluje usměrňovač od napětí na kondenzátoru. [22]

Pokud se nejedná o dlouhodobou změnu toku energie, jako například u zabrzdění střídavého motoru, používá se tzv. ztrátové řešení. Do stejnosměrného meziobvodu je zařazen jednoduchý měnič napětí s odporovou zátěží (tento odpor je často označován jako brzdný odpor) a při stoupnutí napětí ve stejnosměrném meziobvodu se přebytečná energie spálí v odporové zátěži. [22]

Protože v této diplomové práci je použit frekvenční měnič firmy Siemens, jsou zde tyto měniče stručně popsány v následující podkapitole.

### 1.4.1 Měniče Siemens

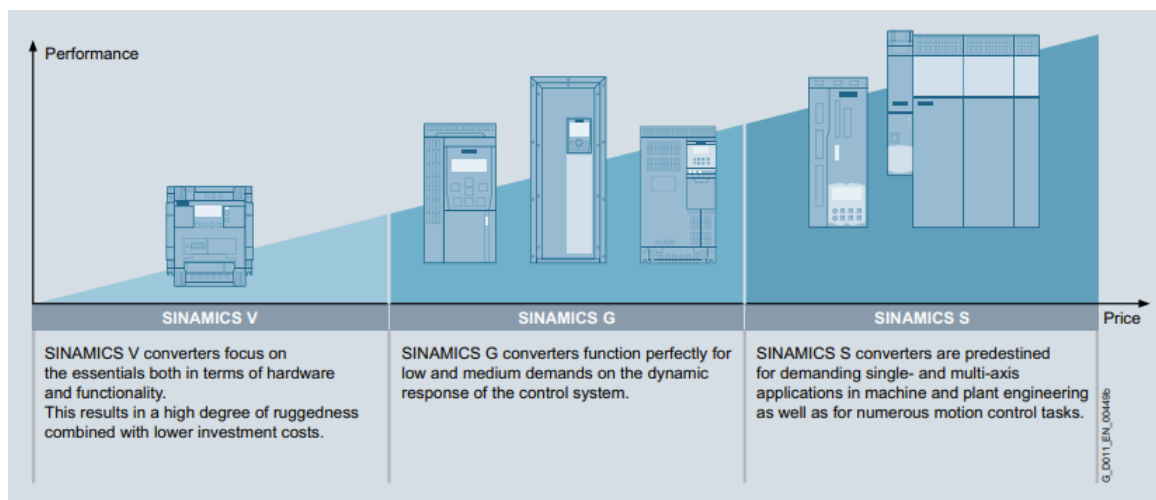
Firma Siemens je jedním z předních dodavatelů frekvenčních měničů na světě. Vedle hlavní produkční řady frekvenčních měničů SINAMICS dodávají také tzv. *motion control systems* - systémů řízení pohybu motorů SIMOTION, což jsou systémy, které spojují funkčnost frekvenčního měniče a PLC do jednoho zařízení.

Produktová řada měničů SINAMICS nabízí řešení pro všechny možné pohonařské aplikace:

- Od jednoduchých aplikací jako jsou čerpadla a ventilátory v procesním průmyslu
- Přes náročné aplikace jednoosých pohonů jako jsou například odstředivky, lisy, extrudéry, výtahy nebo dopravníky
- Až po víceosé servopohony s vysokými požadavky na dynamiku, např. pro obráběcí stroje, balicí stroje nebo tiskařské či papírenské stroje.

V závislosti na aplikaci lze využít konkrétní měnič řady SINAMICS. Základní rozdělení podle požadavků nalezneme na Obr. 1-19. Vidíme, že základní rozdělení je na řadu SINAMICS V,

SINAMICS G a SINAMICS S. Do řady V patří základní jednoduché měniče, které se soustředí na vysokou odolnost a nízkou cenu. Měniče řady G splňují lehké a středně náročné požadavky na dynamiku řídicího systému. Měniče řady S slouží k náročným jedno i víceosým aplikacím. V porovnání s měniči řady V a G se to však promítne na ceně. [23]



Obr. 1-19 Základní rozdělení měničů SINAMICS na řadu V, G a S. [23]

Co se funkcionality měničů SINAMICS týče, hlavní část je implementovaná v softwaru, který je znám také jako firmware. Dostupné firmwarové funkce měničů SINAMICS jsou rozděleny do skupin. K hlavním 8 skupinám patří:

- Základní funkce měniče
- Standardní technologické funkce
- Pokročilé technologické funkce
- Komunikační funkce
- Bezpečnostní funkce měniče
- Funkce energetické účinnosti
- „Běžné inženýrství“ (*common engineering*)
- Aplikace a know-how firem

#### 1.4.1.1 Základní funkce měniče

Základní funkce měniče se dají dále dělit následovně:

- **Řízení:** Jádrem měniče je metoda použitá k řízení měniče. Siemens používá 3 základní metody řízení měniče: U/f řízení (řízení bez zpětné vazby), Vektorové řízení (řízení se zpětnou vazbou) a řízení typu servo (také zpětnovazební řízení). Řízení se dá také dělit na řízení proudu, rychlostní řízení, momentové řízení, polohové řízení a řízení technologického procesu (např. tlaku, průtoku, teploty atp.)



- **Dostupnost:** se týká toho, jak často samostatné zařízení omezí celý výrobní proces. Proto je důležité, aby měnič vstoupil do chybového stavu pouze v případě, že je to nezbytně nutné kvůli ochraně samotného měniče. K možnostem zvýšené dostupnosti patří např. automatický restart, letmý restart, redundance.
- **Požadovaná hodnota:** Kanál požadované hodnoty zajišťuje propojení požadované hodnoty a řízení měniče (tedy i motoru).
- **Limitní a monitorovací funkce:** Zajišťuje omezení vstupních a výstupních proměnných podle potřeby. Monitorovací funkce jsou používány k detekci podmínek, které mohou být nebezpečné pro připojené zařízení.
- **Diagnostika:** Zajišťuje všechny funkce, které pomáhají k nalezení možných příčin problémů. Obsahuje například zásobník chyb a alarmů, diagnostický zásobník, seznam chybějících signálů, status komunikačních telegramů a další.
- **Ochrana:** Týká se jak ochrany hardwaru, tak softwaru. [23]

#### 1.4.1.2 Pokročilé technologické funkce

Pokročilé technologické funkce jsou dostupné pouze pro měniče S120. Patří sem:

- **SINAMICS DCC (drive control chart):** Tabulka řídicích proměnných měniče umožňuje vytvářet vlastní komplexní funkce. Používá se především k řešení aritmetických a řídicích úloh nebo funkcí spojených se složitými aplikacemi.
- **SINAMICS TEC (technology extensions):** Technologické rozšíření jsou konfigurovatelná rozšíření firmwaru, která jsou vytvořena speciálně pro použití v uživatelem přizpůsobených aplikacích se zvláštními požadavky. Tuto funkci lze nainstalovat jako doplněk do standardního rozsahu firmwarových funkcí. Příkladem může být například tlumení vibrací VIBX. [23]

#### 1.4.1.3 Bezpečnostní funkce

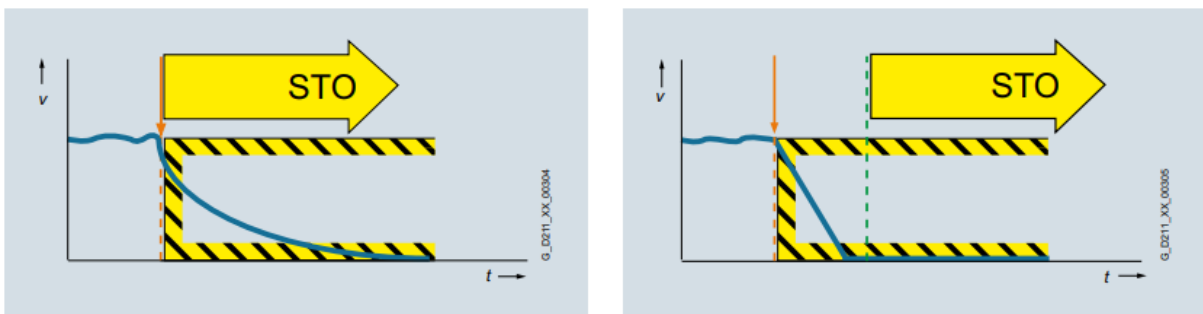
Výrobci strojů a výrobní závody musí zařídit, že jejich stroje nebo zařízení nemohou způsobit nebezpečí vlivem poruchy zařízení. Proto obsahují měniče firmy Siemens bezpečnostní funkce. Tyto bezpečnostní funkce vychází z mnoha standardů, například z EN ISO 12100, IEC 61508, EN 62061 nebo EN ISO 13849-1.

Bezpečnostní funkce lze rozdělit do 4 kategorií:

- **Funkce k bezpečnému zastavení pohonu:**
  - *Safe torque off (STO)* – bezpečné vypnutí momentu
  - *Safe stop 1 (SS1)* – bezpečnostní zastavení 1
  - *Safe stop 2 (SS2)* – bezpečnostní zastavení 2
  - *Safe operating stop (SOS)* – bezpečnostní zastavení a udržení aktuální polohy

- **Funkce k bezpečnému řízení brzdy:**
  - *Safe brake control (SBC)* – bezpečné řízení brzdy
  - *Safe brake test (SBT)* – bezpečnostní test brzd
- **Funkce k bezpečnému monitoringu pohybu pohonu:**
  - *Safety-limited speed (SLS)* – bezpečně limitovaná rychlost
  - *Safe speed monitor (SSM)* – bezpečné monitorování rychlosti
  - *Safe direction (SDI)* – bezpečný směr
- **Funkce k bezpečnému monitorování polohy pohonu:**
  - *Safety-limited position (SLP)* – bezpečné omezení polohy
  - *Safe position (SP)* – bezpečná pozice

V této diplomové práci je použita bezpečnostní funkce SS1. Tato funkce zajistí, že se motor bezpečně a rychle podle předem definované rampy zastaví a zajistí, že měnič není schopný po zastavení dodávat motoru jakýkoliv moment po dobu spuštění této funkce (toho docílí pomocí funkce STO). Funkce STO zajišťuje, že měnič není schopný ani při nečekaném restartování dodávat moment, který by roztočil ovládaný motor. Docílí toho zablokováním pulzů do tranzistorů střídače. Na je vidět průběh funkcí STO a SS1. [23]

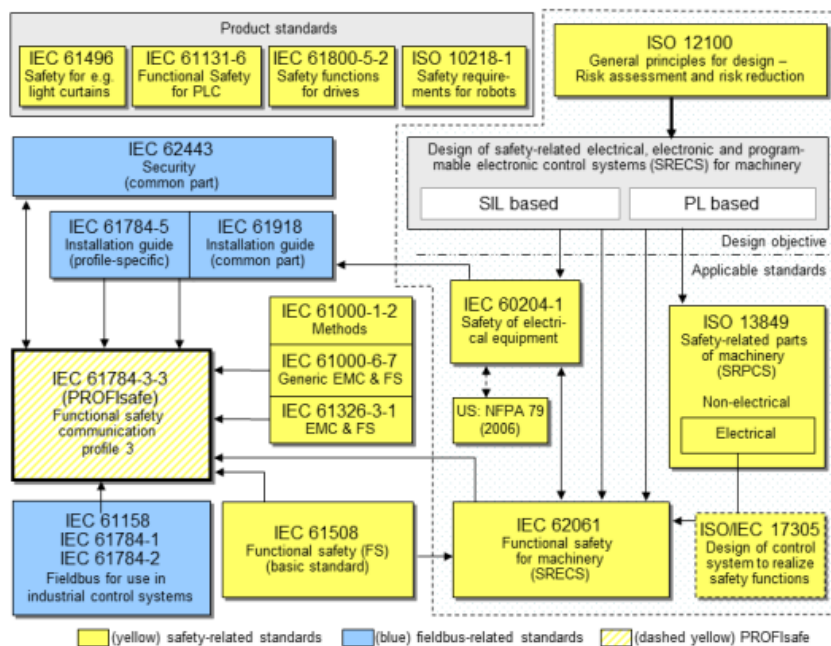


Obr. 1-20 Bezpečnostní funkce STO (vlevo) a SS1 (vpravo). [23]

#### 1.4.1.4 Komunikační funkce

Ke komunikačním funkcím měničů SINAMICS patří především tzv. PROFIDRIVE a PROFISAFE.

Profidrive je aplikační profil Profinetu, definuje chování měniče a způsob přístupu k interním datům měniče, pracuje na 7. vrstvě ISO OSI modelu. Podrobně popisuje přenos informací: cyklický, acyklický přenos, synchronizaci hodinového cyklu (u IRT komunikace), rozhraní snímače, škálování hodnot, definici standardních telegramů, přístup k parametrům měniče a další. Také obsahuje stavový automat (pro sekvenční řízení). Profidrive podporuje centrální i distribuovanou koncepci řízení. [15] [23]



Obr. 1-21 Mezinárodní standardy týkající se bezpečnosti a komunikačních sběrnic v průmyslu. [24]

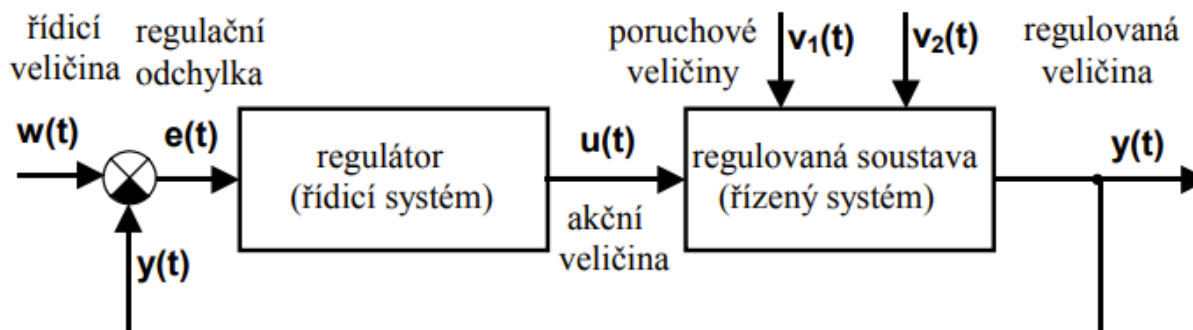
Profisafe je aplikační profil Profinetu, který je realizován na 7. vrstvě ISO/OSI modelu. PROFISAFE snižuje pravděpodobnost vzniku tzv. zbytkové chyby (*residual error*) čímž zvyšuje spolehlivost přenosu dat mezi bezpečnostním kontrolerem F-Host a bezpečnostním zařízením F-device až na úroveň požadovanou relevantními standardy, jako např. IEC 61508 nebo IEC 62280-1. Mezinárodní standardy, ze kterých vychází Profisafe jsou zobrazeny na Obr. 1-21 [24]

## 1.5 Regulace

Regulace je řízení se zpětnou vazbou. Úkolem regulace je nastavení požadované hodnoty dané veličiny (tlak, teplota, otáčky, proud, ...) a následně tuto hodnotu udržovat. Regulace probíhá v systému označovaném jako regulační obvod. Regulační obvod typicky obsahuje dvě hlavní části – regulátor a regulovanou soustavu. Regulovaná soustava neboli řízený systém je regulována regulátorem. Veličina, která je regulátorem udržována na požadované hodnotě, se nazývá regulovaná veličina a označuje se písmenem  $y$ . Regulovaná veličina může být téměř jakákoliv fyzikální veličina, jako např. teplota, tlak, rychlost, pH, průtok, elektrické napětí, elektrický proud, výška hladiny atd. Řídicí veličina, značená písmenem  $w$ , je veličina, pomocí které nastavujeme hodnotu, které má dosáhnout regulovaná veličina. V regulačním obvodu je potřeba neustále měřit rozdíl regulované veličiny od řídicí veličiny. Tento rozdíl se nazývá regulační odchylka, značí se písmenem  $e$  a je definován vztahem (1-3). Jakmile má regulační odchylka nenulovou hodnotu, tak regulátor provádí akční zásah. Tento zásah se uskuteční výstupní veličinou z regulátoru, která se nazývá akční veličina a značí se  $u$ . [5]

$$e = w - y \quad (1-3)$$

Příčinou potřeby regulace jsou poruchy. Poruchové veličiny se značí  $v_1, v_2, \dots$ . Poruchové veličiny působí nežádaným způsobem na regulovanou soustavu, a to nepředvídatelným způsobem. Základní blokové schéma regulačního obvodu s popsány veličinami je zobrazeno na Obr. 1-22.



Obr. 1-22 Blokové schéma obecného regulačního obvodu. [5]

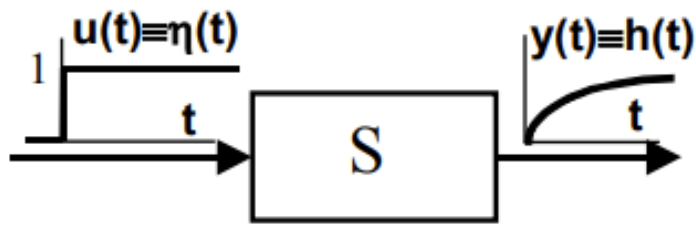
Z hlediska regulace je důležité znát vlastnosti regulačního členu (regulačního systému). Vlastnosti regulačního členu dělíme na dynamické a statické. Statické vlastnosti jsou vlastnosti regulačního členu v ustáleném stavu. Dynamické vlastnosti jsou vlastnosti, kdy se hodnoty vstupních a výstupních veličin mění, jedná se o vlastnosti systému během přechodového děje. A protože u regulace nám jde především o zkoumání chování přechodných dějů, tak je především důležité zabývat se dynamickými vlastnostmi. Dynamické vlastnosti regulačních členů a systémů lze popsat buď vnějším popisem, nebo popisem vnitřním.

Vnější popis zkoumá vlastnosti systému pouze z hlediska vstupních a výstupních veličin, co se děje uvnitř systému nijak nezkoumá. Vnějšího popisu systému můžeme dosáhnout různě. Používané způsoby vnějšího popisu jsou např. diferenciální rovnice systému, přenos, impulsní charakteristika, přechodová charakteristika, frekvenční přenos a frekvenční charakteristika.

Vnitřní popis systému pracuje navíc se stavem systému. Důvodem je, že systémy často nereagují pouze na vstupní proměnné, ale také podle toho, jaký byl počáteční stav systému. Vnitřní popis systému je vyjádřen stavovými rovnicemi. [5]

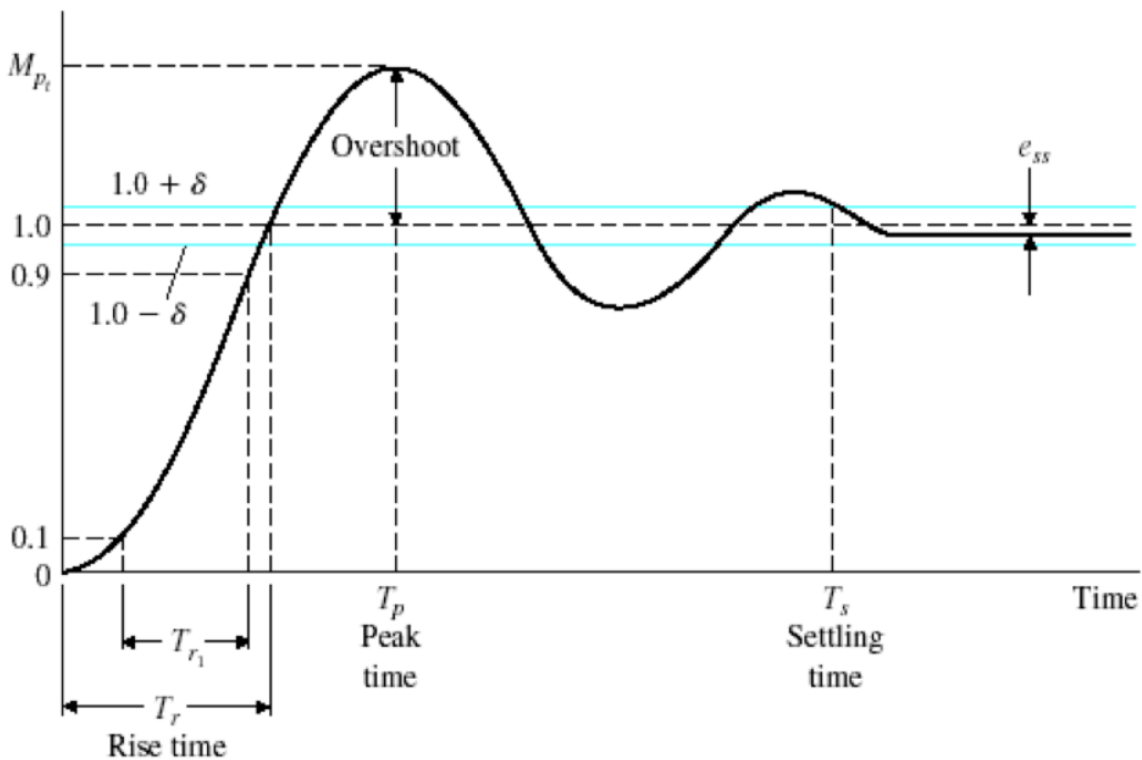
### 1.5.1 Přechodová charakteristika

Přechodová charakteristika je jedním ze způsobů, jak popsat regulační člen nebo systém. Přechodová charakteristika je grafické znázornění přechodové funkce, což je odezva systému na jednotkový skok. Přechodovou charakteristiku značíme  $h(t)$ . jednotkový skok je funkce, která má do času  $t = 0$  nulovou hodnotu a následně v čase 0 „skočí“ na hodnotu 1, značíme ho  $n(t)$ . [5]



Obr. 1-23 Přebodová charakteristika. [5]

Přebodové charakteristiky lze snadno získat experimentálně. Používají se proto často k identifikaci systémů, kde se nedaří identifikovat systém jiným způsobem. U přebodové charakteristiky lze sledovat několik vlastností, které vypovídají o vlastnostech systému. Mezi ně patří doba ustálení, překmit nebo doba náběhu. Na Obr. 1-24 je zobrazen přebodová charakteristika s popisem těchto typických vlastností. Doba ustálení, na obrázku značena jako  $T_s$ , je doba, za kterou se odezva ustálí v definovaném rozmezí od konečné hodnoty. Překmit vyjadřuje, jaké maximální hodnoty dosáhla odezva, bývá vyjádřen procenty vůči ustálené hodnotě. Doba náběhu je čas, během kterého se hodnota zvýší z 10 % konečné hodnoty na 90 % konečné hodnoty.

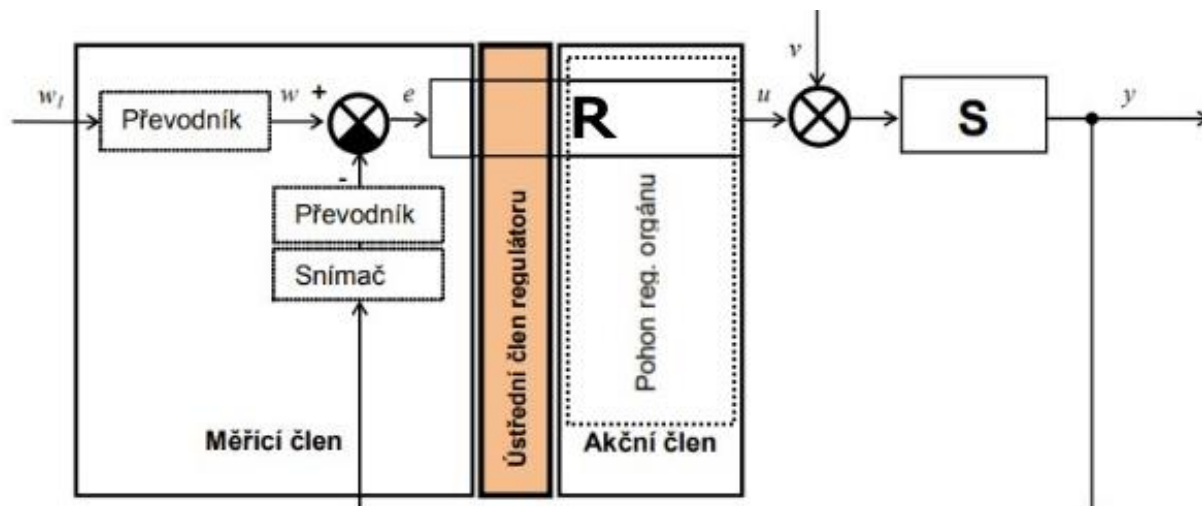


Obr. 1-24 Přebodová charakteristika. [25]

## 1.5.2 Regulátory

Jak bylo již zmíněno, regulátor je zařízení, které se prostřednictvím akční veličiny snaží působit na regulovanou soustavu tak, aby byla regulační odchylka nulová. Regulátory lze dělit několika způsoby. Prvním dělením je dělení na regulátory přímé a nepřímé. Přímé regulátory jsou regulátory, které ke své funkci nepotřebují přídavný zdroj energie. Dnes častěji používané nepřímé regulátory naopak přídavný zdroj energie potřebují. Nepřímé regulátory dělíme na mechanické, pneumatické, hydraulické a elektrické. Dále dělíme regulátory na spojitě a nespojitě podle toho, zda se výstupní signál mění spojitě nebo skokově. Také lze dělit regulátory na lineární a nelineární podle toho, zda realizují lineární nebo nelineární přenosové funkce. [5], [26]

Vnitřní struktura regulátoru je vidět na Obr. 1-25. Každý regulátor je tvořen měřicím členem, ústředním členem a akčním členem. Měřicí člen slouží k změření regulované veličiny  $y$ . Ústřední člen regulátoru slouží k realizaci výpočetních funkcí regulátoru. Pomocí vhodných časových funkcí mění odchylku veličinu tak, aby se vytvořila požadovaná regulační závislost. Akční člen slouží k realizaci vypočtené požadované změny.



Obr. 1-25 Vnitřní struktura regulátoru. [26]

Ústřední člen regulátoru bývá často konstruován jako samostatný prvek, jehož signál je nanejvýše lineárně zesilován, proto se často název regulátor přenáší přímo na ústřední člen. V nadcházejícím textu bude tedy regulátorem myšlen ústřední člen.

Existují 3 základní typy regulátorů:

- **Proporciální - P:** Zesiluje regulační odchylku. Vysoké proporciální zesílení zaručí velkou změnu na výstupu při změně odchylky. Příliš velká proporciální složka však může způsobit nestabilitu.

- **Integrační – I:** Akční veličina je časovým integrálem regulační odchylky. Způsobuje nulovou ustálenou regulační odchylku. Při příliš vysoké integrační složce však hrozí vznik oscilace.
- **Derivační – D:** Akční veličina je úměrná derivaci regulační odchylky v čase. Velikost výstupního signálu je úměrná velikosti změny regulační odchylky v čase. Díky tomu výstupní regulovaná veličina dříve dosáhne ustáleného stavu. Nevýhodou je citlivost na poruchový signál.

V praxi nepoužíváme samostatné regulátory, ale regulátory složené z výše popsaných složek. Nejčastěji používanými regulátory jsou regulátor PI a PID. [26]

## 1.6 Typy řízení pohonů

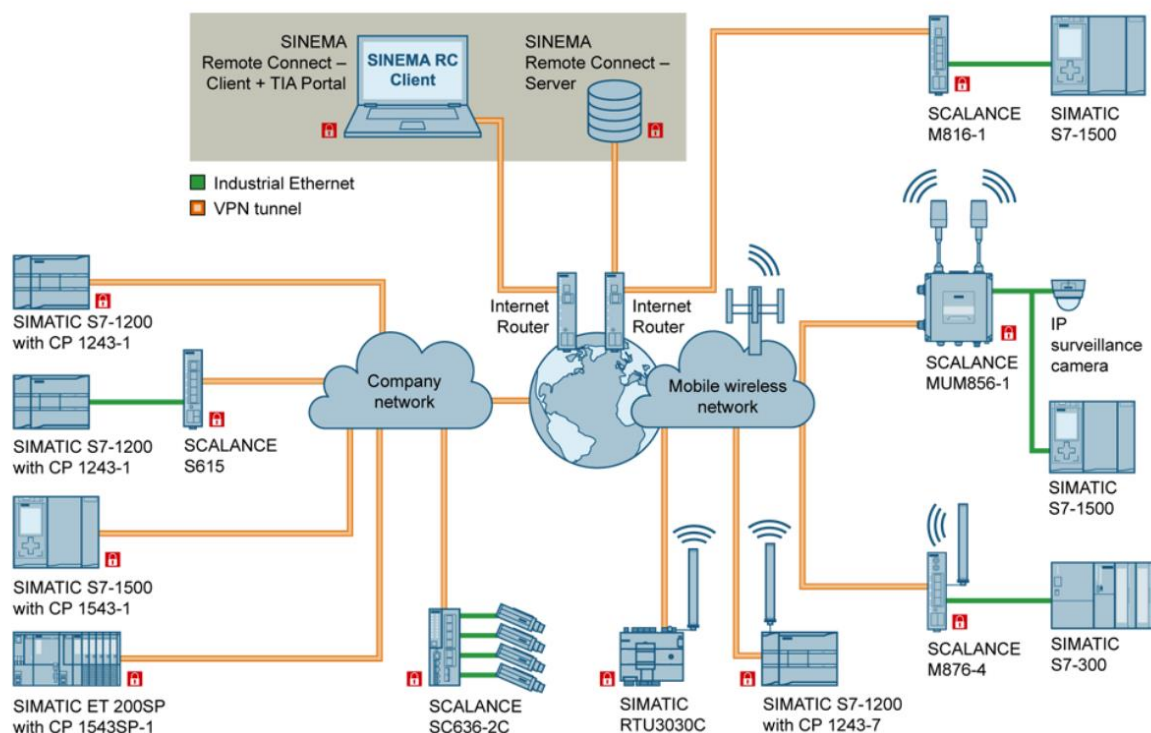
Rozlišujeme dva typy řízení pohonů – centrální a decentrální.

Centrální řízení pohonů znamená, že řízení přebírá PLC a pohon je pouze aktuátor. Výhodou je, že systém má již zabudovanou diagnostiku a alarmy generuje automaticky. Nevýhodou je, že výpočty potřebné k řízení pohonů zatěžují CPU, což znamená, že je možné řídit takto jen omezený počet pohonů. U firmy Siemens je centrální řízení pohonů provozováno s pomocí tzv. technologických objektů v PLC. [27]

Decentrální řízení pohonů znamená, že struktura řízení (chování osy, koordinace osy, interpolace, polohový regulátor) je přímo v pohonu a PLC pouze komunikuje požadované hodnoty, pokyny ke spuštění atd. Výhodou je, že CPU PLC je takto méně zatěžováno, lze tedy takto realizovat vysoký počet pohonů. U tohoto typu řízení je však nutné vytvořit příjem zpráv do a z měniče. K tomu jsou u firmy Siemens používány harmonizované standardní funkční bloky jako například Sina\_speed, Sina\_pos nebo Sina\_para. [27]

## KAPITOLA 2: MOŽNOSTI VZDÁLENÉ SPRÁVY

Firma Siemens ke vzdálené správě poskytuje platformu SINEMA Remote Connect (Sinema RC). SINEMA RC je platforma centrální správy pro výkonný a zabezpečený přístup ke strojům a zařízením kdekoli na světě. Zajišťuje správu VPN tunelů mezi serverem, servisními techniky a instalovanými zařízeními. Servisní technik se nepřipojuje k zařízení přímo, ale prostřednictvím SINEMA RC serveru. Server jednotlivé uživatele a stroje identifikuje pomocí certifikátů a propojí pouze oprávněné uživatele. Stroje lze k platformě Sinema RC připojit pomocí mobilních telefonních sítí (UMTS/LTE), přes pevné linky (DSL) nebo skrz stávající privátní sítě (LAN). K tomuto účelu nabízí společnost Siemens široké portfolio průmyslových routerů řady Scalance. Zařízení lze velmi jednoduše připojit díky autokonfiguračnímu rozhraní. V laboratoři by bylo vhodné k připojení k SINEMA RC serveru použít mobilní VPN router Scalance M876-4, který realizuje připojení přes mobilní síť (2G, 3G, 4G). Ke správné funkci routeru je tedy potřebná SIM karta s aktivovanou službou mobilních dat. Z Obr. 2-1 lze vidět, že možností připojení se k SINEMA RC serveru je mnoho v závislosti na použitém hardwaru.

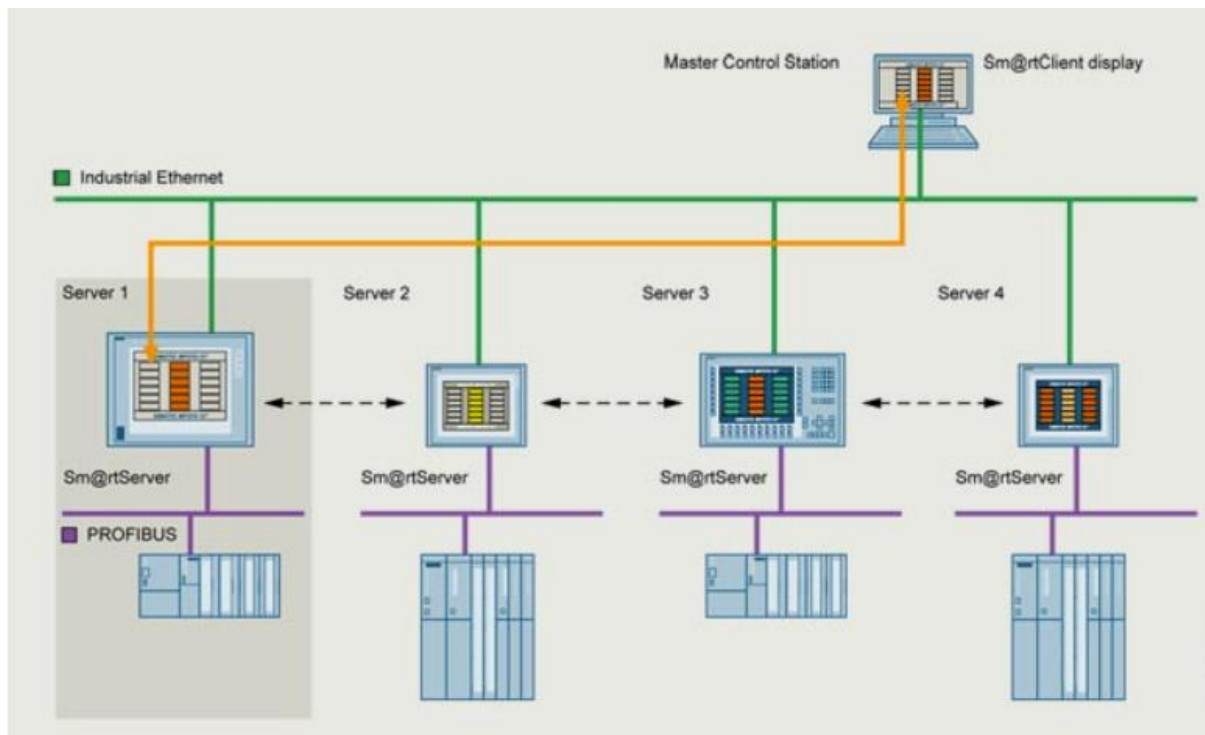


Obr. 2-1 Přehled možností vzdáleného připojení s pomocí SINEMA RC. [28]

Druhou možností, která lze použít, je využití vzdáleného přístupu k panelu pomocí WinCC Sm@rtServer. Sm@rtServer umožňuje ovládat a monitorovat HMI panel. Panel, který je Sm@rtServerem, poskytuje své uživatelské rozhraní Sm@rtClientům. Sm@rtClient může být různé zařízení – počítač, notebook, tablet, jiný HMI panel nebo chytrý telefon. Pokud jsme



připojení s klientem do stejné sítě, jako HMI panel, lze se vzdáleně připojit k HMI panelu a ovládat stanoviště stejně, jako kdybychom fyzicky ovládali HMI panel.



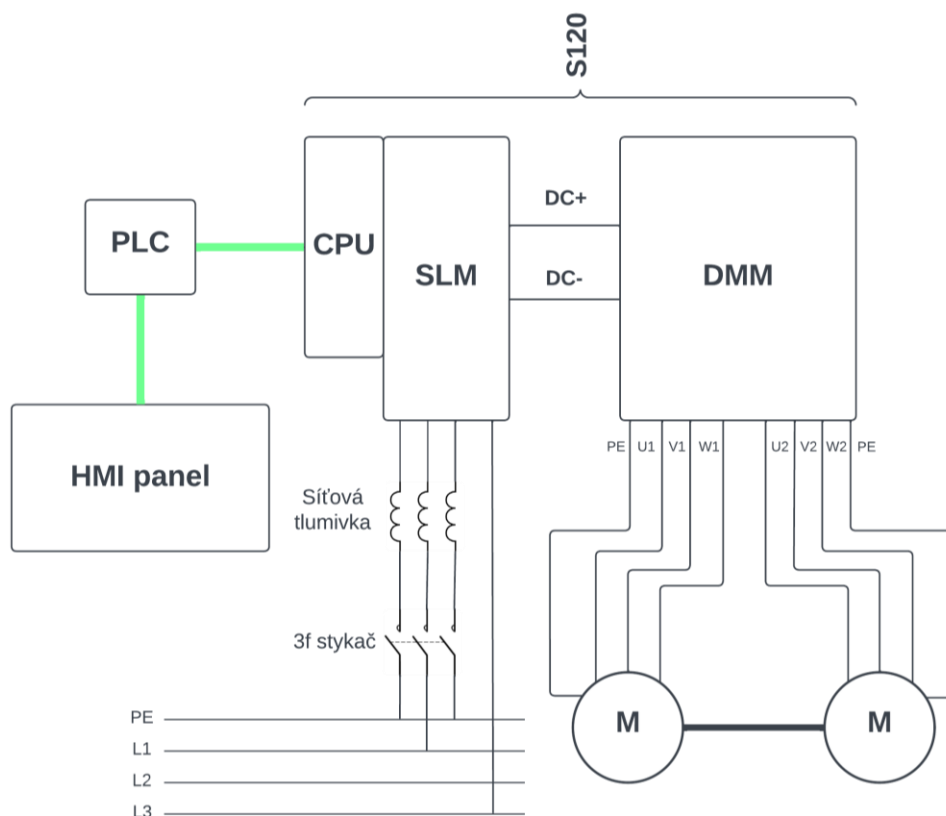
Obr. 2-2 Schéma použití Sm@rtServeru. [29]

Třetí možností vzdálené správy je OPC UA – *OPC unified architecture*. OPC UA je průmyslovým komunikačním standardem, který je založen na TCP/IP a HTTP standardech. Umožňuje přenášení procesních dat, alarmů a historických dat. Specifikace OPC UA je založená na předávání dat mezi OPC UA klientem a OPC UA serverem. OPC UA protokol je specifikován jako služebně orientovaná architektura SOA - *Service Oriented Architecture*, což znamená, že jsou definované služby, na které se klient může dotazovat a server na každý dotaz reaguje příslušnou odpovědí. Služby, které server poskytuje, vytváří abstraktní komunikační model. Po navázání fyzického spojení se pomocí služeb vytváří a udržuje zabezpečený kanál *SecuredChannel* a relace *Session*. Zabezpečený kanál je nutné mít aktivní pro veškerou komunikaci, relaci je nutné mít vytvořenou pro dotazování klienta na služby serveru. Obě tyto části komunikace jsou často dohromady označovány jako komunikační zásobník - *Stack*. Popis adresního prostoru se provádí pomocí uzlů. OPC UA umožňuje tedy přímo přistupovat k datům PLC a případně měnit požadované proměnné PLC. [30]

V této práci je zvoleno druhé řešení, tedy *screen mirroring* HMI panelu s pomocí Sm@rtserveru. Tato možnost byla zvolena, protože aplikace již obsahuje HMI panel, kterého tak lze lehce využít a na rozdíl od OPC UA přístupu uživatel vidí obrazovky HMI panelu při vzdáleném přístupu.

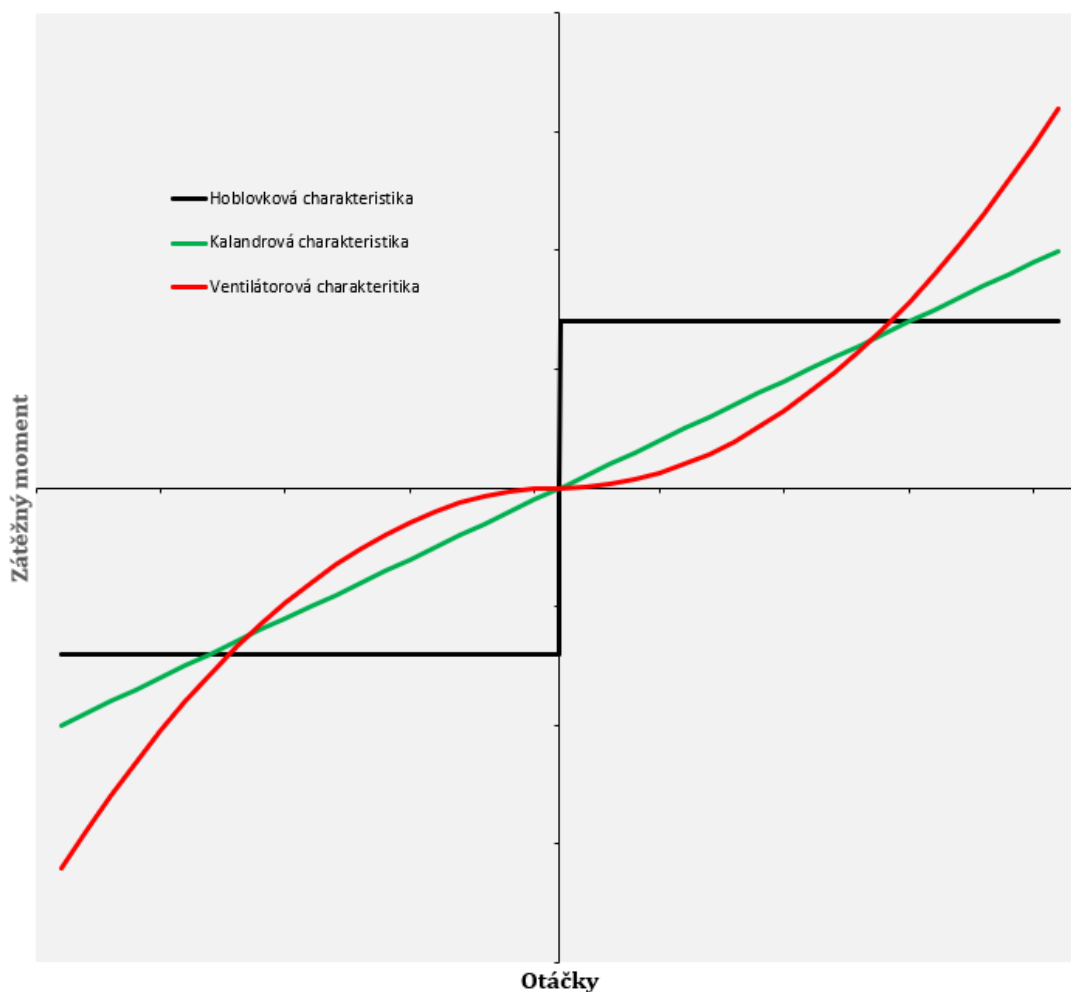
## KAPITOLA 3: POPIS ÚLOHY

Cílem této diplomové práce bylo sestavit a zprovoznit pracoviště k měření charakteristik pohonu, které bude následně využíváno studenty elektrotechnické fakulty ČVUT na laboratorních cvičeních. Konkrétně je stanoviště určeno k měření přechodových charakteristik otáčkového regulátoru asynchronního motoru, který je zatěžován různým způsobem. Základní schéma pro lepší představu stanoviště je na Obr. 3-1. Stanoviště se skládá z dvou mechanicky spřažených motorů, frekvenčního měniče S120, PLC a HMI panelu. Ze schématu lze vidět, že oba motory jsou poháněné z měniče S120, konkrétně z dvojmotorového modulu DMM, což jsou vlastně dva nezávislé střídače, které na výstupu poskytují napětí potřebné velikosti a frekvence. Oba střídače mají stejný SS meziobvod, přičemž usměrněné napětí ze sítě je na meziobvod přivedeno Smart line modulem SLM, což je diodový usměrňovač, který obsahuje tranzistorovým můstkem, který umožňuje dodávání přebytečné energie do sítě a především čtyř-kvadrantový chod měniče. Pulzy usměrňovače v regeneračním chodu a pulzy obou střídačů řídí CPU měniče, stejně tak jako komunikaci s dalšími připojenými zařízeními, jako je například PLC. PLC obsahuje řídicí program, který ovládá celé stanoviště. HMI panel je zde jako operátorské rozhraní pro studenty. S jeho pomocí jsou studenti schopni celé stanoviště řídit, sledovat průběh měření a uložit naměřené hodnoty. PLC, HMI a CPU měniče jsou spolu spojeny pomocí Profinetu, který je na schématu značen zeleně.



Obr. 3-1 Základní silové schéma stanoviště.

Motory jsou mechanicky spřažené, protože zátěžný moment je realizován pomocí druhého asynchronního motoru, zatímco první motor zastává úlohu motoru, kterému je laděn otáčkový regulátor a u kterého je měřena přechodová charakteristika. Zátěžný moment může být různého charakteru, zadáním úlohy však bylo, aby druhý motor byl schopný vytvářet zátěžný moment tří základních charakterů. Tyto tři základní charakteristiky zátěžného momentu se nazývají hoblovková charakteristika, kalandrová charakteristika a ventilátorová charakteristika. Hoblovková charakteristika je charakteristika s konstantním zátěžným momentem, typická je pro obráběcí stroje (hoblovky, odtud hoblovková), pojezdy jeřábů, šroubové kompresory a další. Kalandrová charakteristika je charakteristika, kdy zátěžný moment stoupá lineárně s rostoucími otáčkami. Je typická pro mechanismy s převládajícím viskózním třením, jako jsou kalandry a většina textilních strojů. Ventilátorová charakteristika má průběh v závislosti na otáčkách kvadratický a je typická pro čerpadla, ventilátory, kompresory atp. Průběhy jsou zobrazeny na Obr. 3-2.



Obr. 3-2 Průběhy zátěžného momentu.

### 3.1 Použitý hardware

Kompletní seznam použitého hardwaru je v Tab. 3-1.

Popis	Číslo výrobku	Množství
<b>Pohony</b>		
Řídicí jednotka CU 320-2 PN	6SL3040-1MA01-0AA0	1
Flash card	6SL3054-0FC30-1BA0	1
Kabel drive-cliq 1.0 m	6FX2002-1DC00-1AB0	1
BOP-základní panel obsluhy	6SL3055-0AA00-4BA0	1
SLM - smart line modul	6SL3130-6AE21-0AB1	1
DMM - double motor modul	6SL3120-2TE21-0AD0	1
Síťová tlumivka	6SL3000-0CE21-0AA0	1
Napájecí konektor	6SL3162-2MA00-0AC0	2
SMC30 - sensor modul	6SL3055-0AA00-5CA2	2
Kabel drive-cliq 1.2 m	6SL3060-4AW00-0AA0	2
Kabel motion connect 500	6FX5002-2CA12-1CA0	2
Asynchronní motor SIMOTICS GP	1LE1004-1AB52-2AB4-ZG11	2
Napájecí kabel MC500 50.0 m	6FX5008-1BB11-1FA0	1
<b>HMI</b>		
MTP1500 komfortní panel	6AV2128-3QB06-4AB1	1
<b>PLC</b>		
FDI modul bezpečnostních vstupů	6ES7136-6BA00-0CA0	1
SIMATIC ET 200SP síťový adaptér	6ES7193-6AR00-0AA0	1
SIMATIC ET 200 EP base unit	6ES510-6BP00-0DA0	1
Řídicí jednotka 1510SP F-1 PN	6ES7510-1SJ01-0AB0	1
Paměťová karta	6ES7954-8LE03-0AA0	1
Kabely a konektory pro profinet	6GK1901-1BB10-2AE0	1
<b>Ostatní</b>		
Síťový přepínač SCALANCE XC208	6GK5208-0BA00-2AC2	1
Zdroj napájení SITOP PSU200M	6EP1333-3BA10	1

Tab. 3-1 Seznam použitého hardwaru.

#### 3.1.1 Frekvenční měnič

Frekvenční měnič SINAMICS S120 je vhodným kandidátem pro toto stanoviště. Měnič SINAMICS S120 byl vybrán, protože je vhodný díky svému modulárnímu designu k řízení více os najednou. Také je určený pro náročné aplikace s požadavkem na vysokou dynamiku. Existují 3 metody řízení tohoto měniče: U/f řízení (řízení bez zpětné vazby), Vektorové řízení (řízení se zpětnou vazbou) a řízení typu servo (také zpětnovazební řízení, vhodné především k polohování). Také jsou tři typy řízení os: Rychlostní řízení, momentové řízení a polohové řízení. V tomto případě je měnič provozován ve vektorovém řízení s enkodérem, což je vhodné pro rychlostní osy. Frekvenční měnič S120 se skládá z řídicí jednotky, výkonové jednotky a motorových modulů. V závislosti na použité výkonové jednotce je nebo není měnič schopný rekuperace.

### 3.1.1.1 Řídicí jednotka

Řídicí jednotka provádí výpočty k řízení výkonové části frekvenčního měniče a také je zodpovědná za komunikaci s ostatními zařízeními, především s nadřazeným řídicím systémem, zde PLC. Řídicí jednotka je schopna ovládat od 3 po 12 os, v závislosti na tom, zda je použité řízení U/f, vektorové řízení nebo řízení typu servo, jaká je požadovaná dynamika, a zda obsahuje měnič tzv. *performance license extension* – tedy licenci, která umožňuje řídicí jednotce ovládat více os najednou.

Obvykle má řídicí jednotka také digitální vstupy a výstupy. V této diplomové práci je použita řídicí jednotka CU320-2 PN. Tato řídicí jednotka obsahuje 2 porty k Profinet komunikaci, 12 digitálních vstupů, 8 vstupů typu DI/DO (lze použít podle potřeby) a 4 porty k drive-cliq kabelům. K tomu, aby řídicí jednotka fungovala je také potřeba Paměťová karta typu Flash, na které je nahrán firmware měniče. Řídicí jednotka také typicky obsahuje základní operátorský panel, který zobrazuje například chybové a alarmové hlášky.

### 3.1.1.2 Smart line module

SLM – *smart line module* je výkonová jednotka, která má za úkol usměrnit napětí ze sítě. Usměrněné napětí je následně pomocí stejnosměrného meziobvodu připojeno k motorovému modulu. SLM obsahuje diodový můstek v napájecím směru a IGBT můstek ve zpětném směru, který umožňuje rekuperaci energie zpátky do sítě a čtyřkvadrantový provoz. Použitý SLM je dimenzován na výkon 10 kW, vstupní 3f napětí 380-480 V o frekvenci 50 nebo 60 Hz, výstupní napětí SS meziobvodu 600 V, proud 17 A a obsahuje integrovaný větrák k vlastnímu chlazení.

Mezi sítí a SLM je ještě zapojena síťová tlumivka, která má za úkol snížit odběr vyšších harmonických ze sítě, a tedy snížit celkový negativní dopad na síť způsobený frekvenčním měničem. Síťová tlumivka je přímo doporučena výrobcem, doporučená verze tlumivky má hodnotu jmenovitého proudu 28 A.

### 3.1.1.3 Motorový modul

Motorový modul napájí připojený motor, jedná se vlastně o střídač. Firma Siemens vyrábí motorové moduly jednomotorové a dvoumotorové. Zde je použit dvoumotorový modul, jelikož stanoviště obsahuje dva asynchronní motory. Výhodou modulárního designu měniče S120 je, že ze stejnosměrného meziobvodu může být napájeno víc motorových modulů. V tomto stanovišti je použit motorový modul, který je dimenzovaný na 600 V stejnosměrný meziobvod, a 3f výstupní napětí 400 V a 9 A. Také obsahuje vnitřní větrák, který slouží k vlastnímu chlazení.

### 3.1.1.4 Enkodér

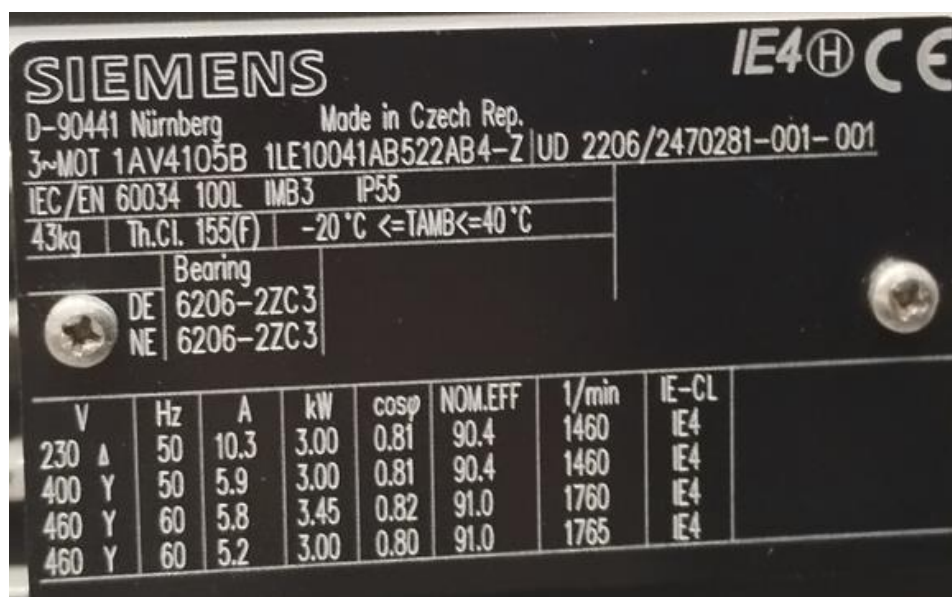
Oba motory obsahují enkodér. Díky enkodéru je možné provozovat frekvenční měnič ve vektorovém režimu s enkodérem, který je přesnější, než když je použit vektorové řízení bez enkodéru, kdy se aktuální rychlost motoru dopočítává z modelu ASM. Použité enkodéry jsou inkrementální enkodéry Sendix 5020, typu HTL se signálovými stopami A, B a R s rozlišením 1024 pulsů za otáčku. Tyto enkodéry také vyžadují napájení, které je poskytnuto senzorovým modulem SMC30, který zároveň zpracovává signály enkodéru a posílá je řídicí jednotce pomocí drive-cliq kabelů. Signály jsou přenášeny pomocí signálového prefabrikovaného kabelu motion-connect 500, který je k enkodéru připojen pomocí konektoru M12. Barevné značení kabelů jednotlivých signálů je v Tab. 3-2.

Pin	Značka signálu	Popis	Barva kabelu
10	M-encoder	Napájení enkodéru - zem	bílo-žlutá
12	P-encoder	Napájení enkodéru - kladný pól	bílo-černá
5	A	Inkrementální signál A	žlutá
6	A*	Inverzní inkrementální signál A	zelená
1	B*	Inverzní inkrementální signál B	hnědá
8	B*	Inkrementální signál B	černá
3	R*	Inverzní inkrementální signál R	fialová
4	R	Inkrementální signál R	modrá

Tab. 3-2 Barevné označení kabelu připojeného k enkodéru.

### 3.1.2 Motory

V této práci jsou použity dva stejné asynchronní motory. Použitý motor je z řady motorů Simotics GP (*general purpose*) motors firmy Siemens. Jmenovité hodnoty motorů jsou zobrazeny na štítku motoru, který je na Obr. 3-3.



Obr. 3-3 Štítek motoru.

### 3.1.2.1 Teplotní čidlo k ochraně motoru

Oba motory obsahují 3 PTC termistory v sérii, jejichž vývody lze nalézt ve svorkovnici. S jejich pomocí je možné v případě přehřátí motoru vypnout měnič. PTC termistory nejsou schopné nám dát přesnou hodnotu teploty motoru, ta je počítána z matematického modelu uvnitř měniče, ale po překročení dané teploty dají měniči signál, že došlo k přehřátí a měnič zahlásí chybu a vypne provoz.

### 3.1.3 PLC

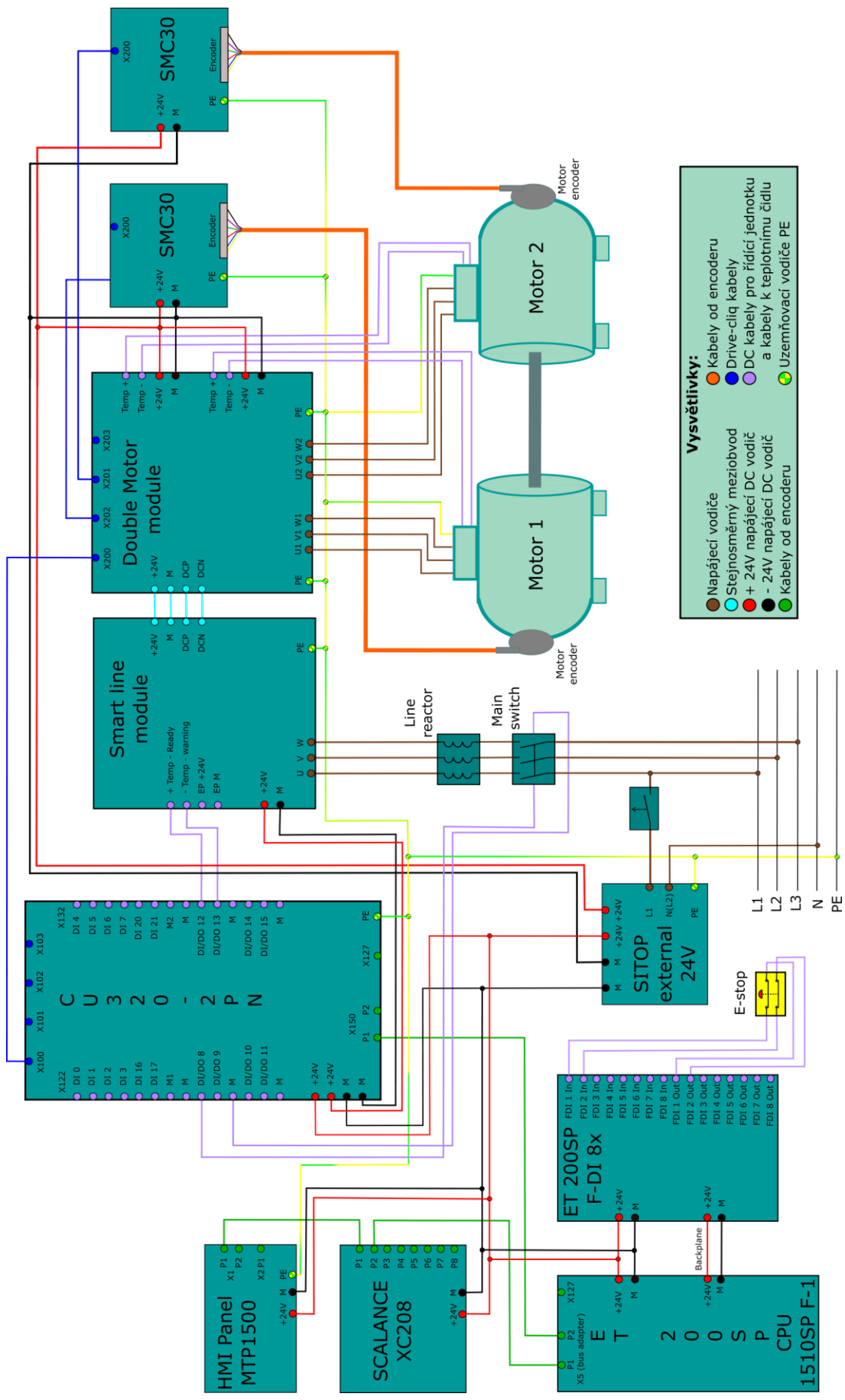
Použité PLC je z řady ET 200SP, jedná se o PLC, které je schopné bezpečnostních funkcí. ET 200 SP je PLC, které má CPU stejného typu jako PLC SIMATIC S7-1500. Použité PLC má CPU 1510SP F-1 PN. PLC má *work memory* rozdělenou na 150KB paměť určenou pro program a 750KB paměť určenou pro data PLC. K tomu obsahuje 128 KB retentivní paměti. PLC je také rozšířeno o paměťovou kartu o velikosti 12 MB, což je paměť typu *load memory* (flash eeprom). Síťový port je rozšířen o síťový adaptér, který rozšiřuje Profinetový port X1 P1 na dva porty, které podporují IRT komunikaci. Použitá firmwarová verze je V2.9. K PLC je připojena jedna karta s bezpečnostními digitálními vstupy, která zpracovává signály z bezpečnostního červeného tlačítka, které slouží k okamžitému a bezpečnému zastavení motorů v případě potřeby.

### 3.1.4 HMI

Použitý HMI panel patří do rodiny *unified comfort* panelů, což jsou nejnovější panely firmy Siemens. Použitý panel MTP1500 má 15.6palcovou obrazovku, 2 porty podporující Profinet, 4 USB porty a 2 porty pro SD kartu. Panel je možné konfigurovat pomocí TIA portálu, je však potřeba mít správnou verzi WinCC, která tyto panely podporuje. Unified panely podporují například psaní skriptů v programovacím jazyce JavaScript, což bylo pro toto stanoviště důležité, protože díky tomu bylo možné uložit požadovaný soubor s naměřenými hodnotami na USB flash disk.

## 3.2 Schéma zapojení

Na Obr. 3-4 nalezneme kompletní schéma zapojení realizovaného stanoviště včetně legendy. Celé stanoviště je napájeno zdrojem napájení (24 V DC) SITOP PSU200M. Jistič zapojený mezi síťové napětí a zdroj slouží nejen k ochraně zdroje napájení, ale také k jeho zapnutí a vypnutí. Po jeho zapnutí je přivedeno napětí ke všem komponentám vyjma motorů, jak lze vidět ze schématu. Co se silové části týče, vidíme, že měnič S120, konkrétně SLM, je připojen k síti, avšak je mu předřazena síťová tlumivka, která snižuje vyšší harmonické složky odebíraného proudu, a trojfázový stykač. Tento stykač je ovládán řídicí jednotkou, konkrétně pomocí DI/DO 8, které je nastaveno jako digitální výstup.



Obr. 3-4 Schéma zapojení realizovaného stanoviště.



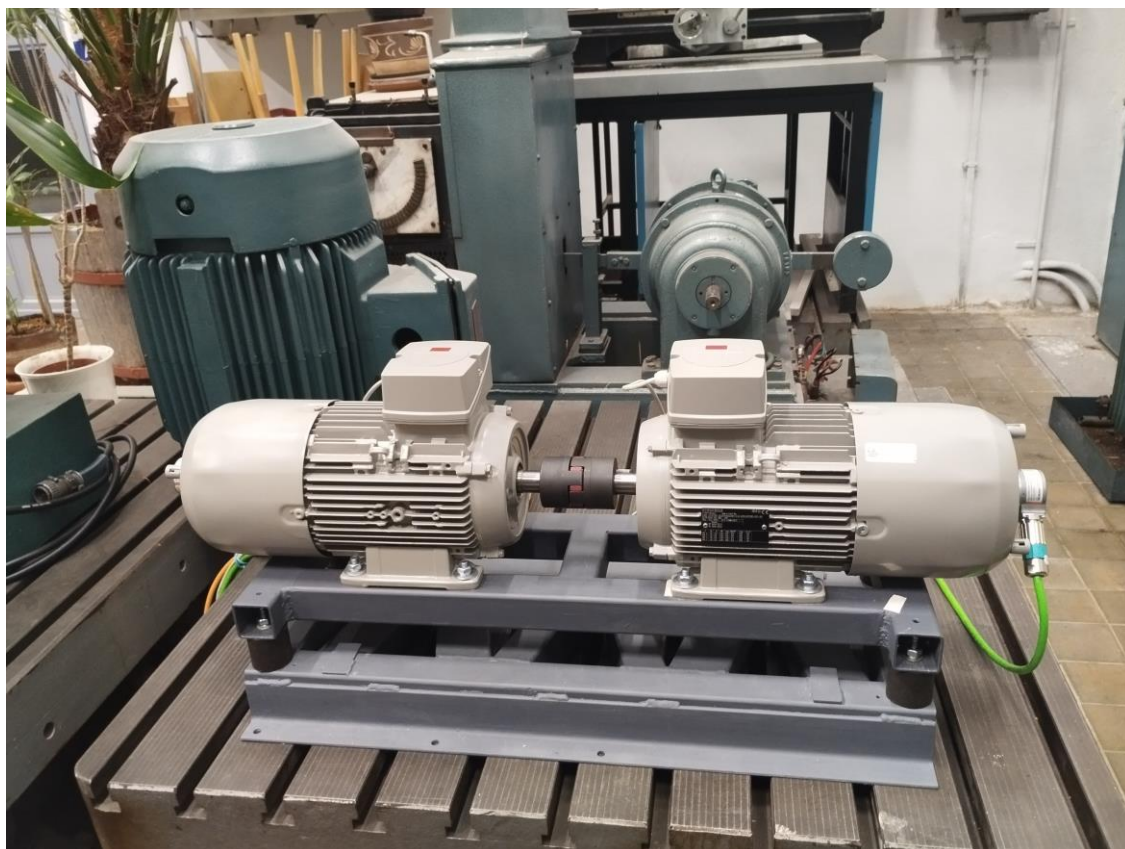
Usměrněné napětí pokračuje přes stejnosměrný meziobvod do dvojmotorového modulu, který následně podle pokynů řídicí jednotky vytváří potřebné střídavé napětí na svorkách U1, V1, W1 a U2, V2, W3, které jsou připojeny pomocí silových kabelů k oběma asynchronním motorům. Je důležité, aby všechny zařízení byly uzemněna. Co se týče měniče, lze vidět, že řídicí jednotka je spojena pomocí tzv. drive-cliq kabelů s dvojmotorovým modulem a ten je následně spojen s oběma sensorovými moduly SMC30. Drive-cliq je systémové rozhraní, které umožňuje komunikaci mezi měničem a prvky souvisejícími s měničem, konkrétně enkodéry. Protože použité asynchronní motory nemají integrované enkodéry s přímým rozhraním drive-cliq, jsou signály z enkodérů posílány přes signální kabely do sensorového modulu, který následně tyto signály zpracovává a umožňuje převod na drive-cliqové rozhraní, pomocí kterého jsou informace o signálech z enkodéru přeneseny až do řídicí jednotky měniče. PTC termistory uvnitř motorů jsou zapojeny s příslušným motorovým modulem DMM, aby byl měnič schopen vyhodnotit, že došlo k přehřátí motoru. Také je potřeba zapojit signál „Temp – Ready“ pomocí DI do řídicí jednotky, jelikož tento signál posílá řídicí jednotce informaci o tom, zda je SLM bez chyby a připravená k provozu.

Co se týče zapojení pro komunikaci Profinet, je potřeba k sobě zapojit HMI panel, řídicí jednotku měniče a PLC, přičemž je potřeba ještě jeden volný port, aby bylo možné připojit počítač s programovacím prostředím a konfigurovat jednotlivá zařízení. V tomto provedení není potřeba Použitý síťový switch Scalance XC208, ale v případě rozšíření úlohy by potřeba byl a jelikož byl objednan fakultou, je ve stanovišti použit. Je potřeba si však uvědomit, že tento switch má konformační třídu CC-B, tedy nepodporuje IRT komunikaci, kterou zde chceme použít mezi měničem a PLC. Proto není zapojená topologie typu hvězda, ale typu strom, kdy ze switche vede spojení k HMI a PLC a následně z PLC vede kabel do řídicí jednotky měniče, přičemž zde je použita IRT komunikace. Posledním, avšak důležitým prvkem je bezpečností červené tlačítko, které je zapojeno do karty bezpečnostních vstupů. Je zapojeno dvěma kanály, což umožňuje bezpečně vyhodnotit, zda je tlačítko ve stavu zapnuto, vypnuto anebo zda došlo k poruše (například přerušením vedení).

Na Obr. 3-5 je fotografie realizovaného stanoviště. Vlevo je stojan, který obsahuje: PLC a kartu pro bezpečnostní vstupy (vlevo nahoře), síťový switch (vpravo od PLC), oba sensorové moduly (vpravo od switche), měnič S120 (vpravo), zdroj napájecího napětí 24 V SITOP (vlevo dole), 1fázový jistič (vpravo od SITOPu), 3fázový stykač (vpravo od 1f jističe) a síťovou tlumivku (pod měničem S120).



*Obr. 3-5 Realizované stanoviště: PLC, měnič, HMI panel a počítač s TIA portálem.*

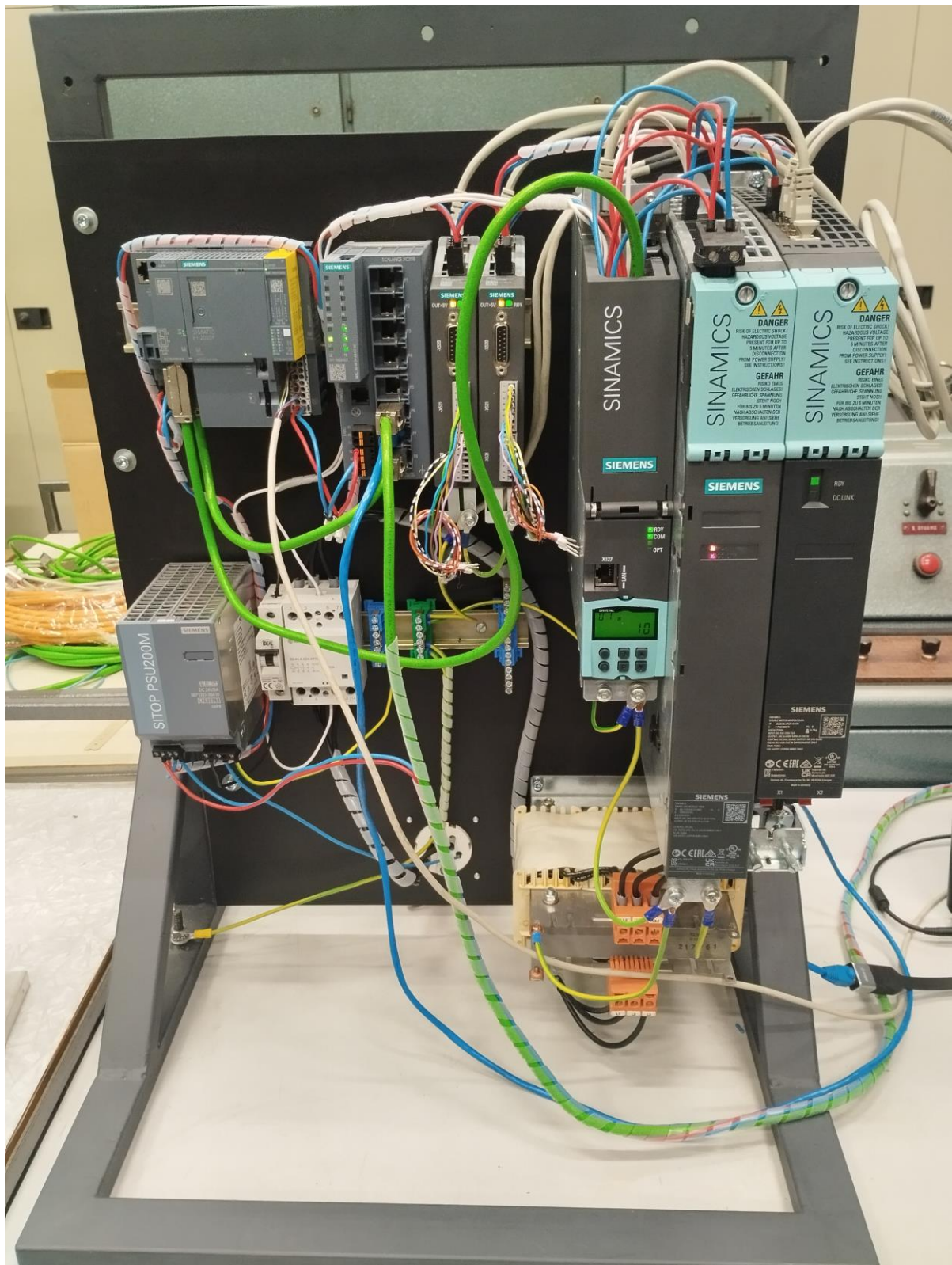


*Obr. 3-6 Mechanicky spřažené motory.*



*Obr. 3-7 Detailní fotografie HMI panelu.*

Napravo od stojanu je notebook s TIA portálem, pomocí kterého byly konfigurovány všechna zařízení a vpravo od notebooku je HMI panel, s jehož pomocí je celé stanoviště ovládáno. Motory, jejichž hřídele jsou mechanicky spřaženy jsou na Obr. 3-6. Detail stojanu je na Obr. 3-8, detail HMI je na Obr. 3-7.



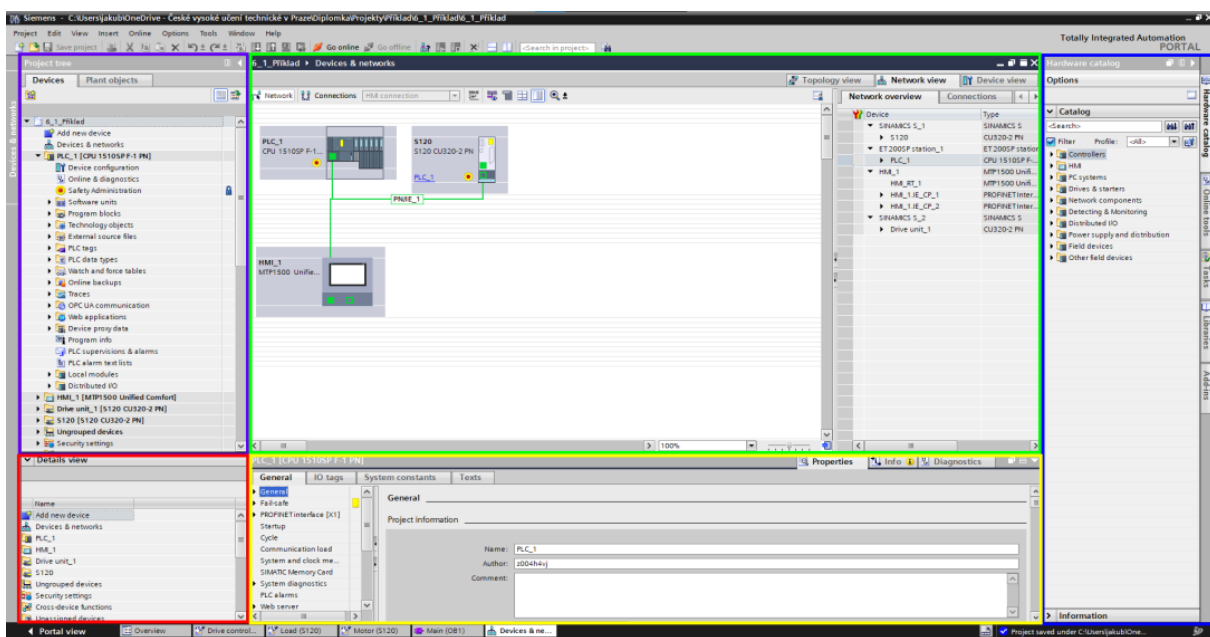
Obr. 3-8 Detailní fotografie stojanu.

### 3.3 TIA portál

Ke konfiguraci, projektování a programování hardwaru firmy Siemens se používá inženýrský softwarový systém TIA portál. V něm jsou ve společném softwarovém prostředí s jednotným ovládáním integrovány veškeré potřebné nástroje. Například k projektování a konfigurování PLC je použit STEP 7, k HMI je použito WinCC a k elektrickým pohonům SINAMICS Startdrive. Konkrétní seznam použitého softwaru, který umožňuje pracovat s použitým hardwarem pro tuto diplomovou práci je v Tab. 3-3.

Zařazení	Produkt
SIMATIC HMI	WinCC Unified PC (100k) ES
SIMATIC HMI	Wincc Unified Comfort ES
SINAMICS	Startdrive Advanced
SIMATIC STEP 7	STEP 7 Professional
SIMATIC STEP 7	STEP 7 SAFETY

Tab. 3-3 Použité licence v rámci vývojového prostředí TIA portál.



Obr. 3-9 Vývojové prostředí TIA portál.

Na Obr. 3-9 je zobrazeno vývojové prostředí TIA portál. To lze rozdělit do několika sekcí, pro názornost jsou tyto sekce na obrázku barevně označeny. Zelenou barvou je ohraničeno hlavní okno editoru, modrou je označen katalog a knihovny, žlutou je označeno okno vlastností a diagnostiky, fialová označuje projektový strom a červená je detailní zobrazení právě rozkliknutého objektu v projektovém stromu. Při vytváření řídicího programu PLC existuje několik typů bloků kódu, které můžeme použít.

Dělí se na:

- **Organizační bloky (OB):** Jsou bloky, které tvoří rozhraní mezi operačním systémem a uživatelem. Tyto bloky jsou volané operačním systémem a umožňují například: Cyklicky provádět uživatelský program pomocí hlavního bloku PLC OB1, který je pojmenován Main, v závislosti na přerušení provést uživatelem naprogramovaný kód, provést specifické instrukce při zapnutí PLC nebo reagovat na chybné stavy.
- **Funkce (FC):** Jsou bloky, které obsahují kód, který nepotřebuje paměť k ukládání dat nebo proměnných, a proto nemá žádný datový blok. Funkce jsou volány jiným blokem a v závislosti na aktuální hodnotě vstupních proměnných a vnitřních instrukcí funkce přiřadí hodnotu výstupním signálům. Funkce se používají, protože umožňují zvýšit přehlednost programu, a také proto, že daná funkce lze použít mnohokrát jednoduchým zkopírováním a uživatel tak nemusí programovat stejný kód vícekrát.
- **Funkční bloky (FB):** Jsou bloky, které obsahují kód, na rozdíl od funkcí však má každý funkční blok vyhrazený datový blok, kam může ukládat potřebná data či parametry ke své funkci. Například k vytvoření náběžné hrany je potřeba znát předchozí stav proměnné a k tomu lze využít datový blok funkčního bloku.
- **Datové bloky (DB):** Datové bloky lze vytvořit samostatně, nemusí být vždy spojené s funkčním blokem. To se hodí pro definování a správu proměnných, což je přehlednější než vytvářet proměnné přes PLC tagy.

Zároveň existuje 5 typů programovacích jazyků, s jejichž pomocí lze programovat uživatelský program v Siemens PLC s CPU řady S7-1500:

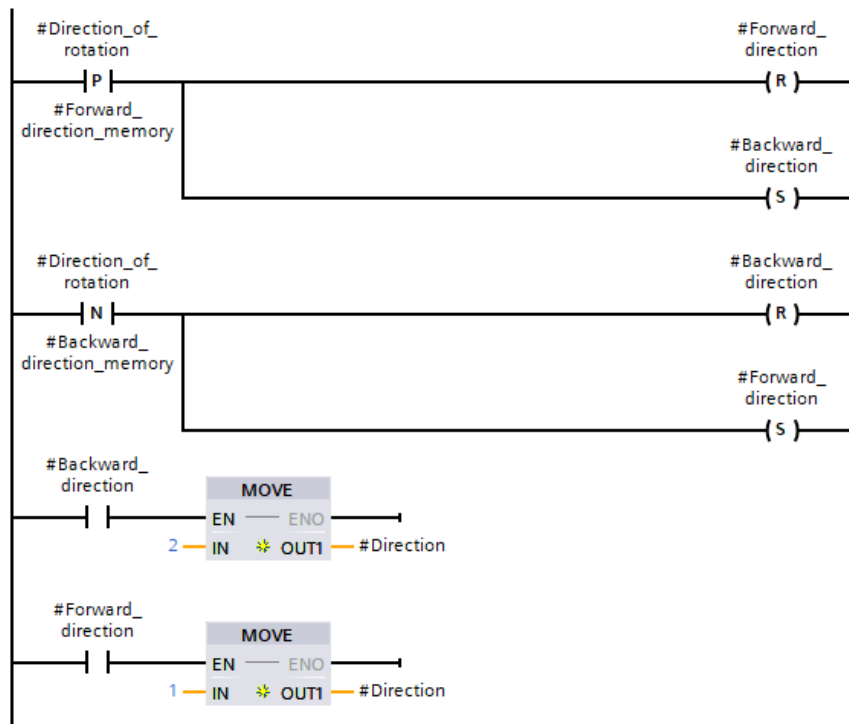
- Žebříčkový diagram (ladder diagram LD)
- Diagram funkčních bloků (function block diagram FBD)
- Seznam instrukcí (statement list STL)
- Strukturovaný řídicí jazyk (structured control language SCL)
- Graf (graph)

V diplomové práci jsou použity LD, SCL a FB, proto jsou krátce popsány.

### 3.3.1.1 Žebříčkové diagramy LD

Žebříčkové schéma je programovací jazyk, který byl vytvořen z reléových schémat, která se hojně používaly v období reléové automatizace předtím, než se začaly používat PLC. Výhodou žebříčkového diagramu je jeho přehlednost. Tyto diagramy se skládají ze dvou sběrnic – levé kladné a pravé záporné, mezi které uživatel zapojuje kontakty, cívky, výstupní bloky, bloky s vstupem EN a ENO výstupem (časovače, funkční bloky, ...) a další. Kontakty jsou typicky typu NC *normaly closed* – rozpínací nebo NO *normaly open* – spínací, nebo kontakty které detekují

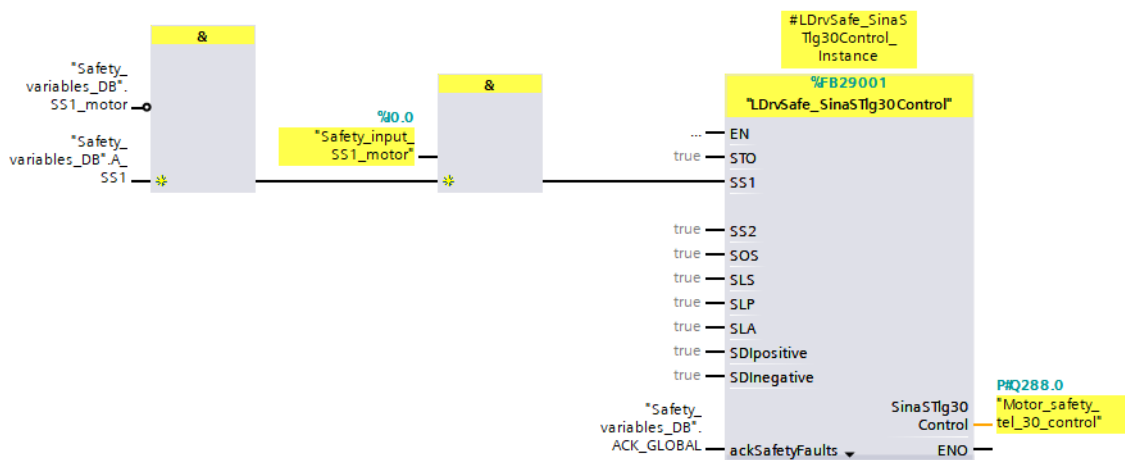
náběžnou nebo sestupnou hranu. Na Obr. 3-10 je zobrazena část programu, která je vytvořena pomocí žebříčkového diagramu. Jedná se o kód, který na základě vstupní hodnoty Direction\_of\_rotation přesune do proměnné Direction hodnotu 1 nebo 2.



Obr. 3-10 Ukázka žebříčkového diagramu.

### 3.3.1.2 Diagram funkčních bloků FBD

Diagram funkčních bloků je programovací jazyk, který využívá základní booleovské logiky. Typické jsou pro něj základní funkce jako je AND, OR, XOR, NOT atp., ale stejně jako u žebříčkového schématu lze libovolně použít funkčních bloků, časovačů a dalších funkcí. Příklad programu, který je programován pomocí FBD je na Obr. 3-11. Jedná se o logiku spínání bezpečnostní funkce SS1.



Obr. 3-11 Ukázka diagramu funkčních vstupů.

### 3.3.1.3 Strukturovaný řídicí jazyk SCL

Strukturovaný řídicí jazyk je programovací jazyk, který je založený na programovacím jazyku Pascal. Jedná se o procedurální programovací jazyk. Tento jazyk je vhodný k použití v případě, že je potřeba vytvořit složité logické operace, které by pomocí LD nebo FBD byly příliš komplikované. Ukázka programovacího jazyku SCL je na Obr. 3-12.

```
15 IF (#M_Requested_I_memory <> #M_Requested_I) THEN
16     #Requested_I := #M_Requested_I;
17     #Selection_P_or_I := TRUE;
18 END_IF;
19
20 IF (#M_Requested_P_memory <> #M_Requested_P) THEN
21     #Requested_P := #M_Requested_P;
22     #Selection_P_or_I := FALSE;
23 END_IF;
24
25 #M_Requested_I_memory := #M_Requested_I;
26 #M_Requested_P_memory := #M_Requested_P;
27 #Status := 'Manual';
28 #Turn_off_measurement_after_5sec := FALSE;
29 #Turn_off_measurement_due_to_time := FALSE;
30 #Measure_remaining_values := FALSE;
```

Obr. 3-12 Ukázka programovací jazyku SCL.



## KAPITOLA 4: TVORBA PROJEKTU V TIA PORTÁLE

V předchozí kapitole byly popsány všechny hardwarové záležitosti a bylo vysvětleno co je to program TIA portál. V této kapitole je popsáno, jakým způsobem je potřeba jednotlivá zařízení nakonfigurovat a jak naprogramovat PLC pomocí TIA portálu, aby sestava byla plně funkční.

Nejprve je potřeba založit nový projekt. Následně je potřeba přidat všechna zařízení, která jsou ve stanovišti použita, tedy PLC, měnič a HMI panel. To lze realizovat pomocí tlačítka přidání nového zařízení, které je v projektovém stromě. Všechna přidaná zařízení lze najít v projektovém stromě anebo v sekci „*Devices & networks*“. Je potřeba se ujistit, že bylo přidáno správné zařízení, ale také že firmwarová verze zařízení se shoduje s firmwarovou verzí, která je fyzicky nahrána v zařízení. V této práci jsou použity následující firmwarové verze: V2.9 pro PLC, V5.2.3 pro řídicí jednotku měniče a V17.0.0.0 pro HMI panel. V případě, že je třeba zkontrolovat, jaký firmware je nahráný v zařízení, pak je možné se připojit k zařízení pomocí tzv. *online access*. *Online access* umožňuje zobrazit všechna dostupná zařízení na jednotlivých komunikačních rozhraních.

Poté, co jsou všechna zařízení přidána do projektu, je vhodné je propojit požadovaným komunikačním rozhraním, v tomto případě Profinetem, a nastavit Profinet jméno zařízení a IP adresu. Zvolené IP adresy musí být ve stejné podsíti. Jméno a IP adresa v projektu musí být shodné s jménem a IP adresou fyzického zařízení. Pokud je třeba nejprve změnit jméno a IP adresu u fyzického zařízení, toho lze dosáhnout také pomocí *online access*.

### 4.1 HW konfigurace měniče

Pomocí tlačítka *add new device* byla přidána pouze řídicí jednotku měniče S120. Dále je však potřeba přidat topologii zapojení drive-cliq. To se provede tak, že se rozklikne řídicí jednotka a je přidán DDM z HW katalogu vpravo. Po přidání DDM ho je potřeba rozkliknout a ve vlastnostech dále nastavit typ řízení na vektorový (to je vhodné pro rychlostní osy) a následně vybrat správné výrobní číslo DDM. Poje je potřeba přidat z HW katalogu synchronní motory a ve vlastnostech nastavit jejich štítkové hodnoty. Nakonec je potřeba zvolit správný systém měřící rychlost, v tomto případě HTL/TTL enkodér. Po vložení se zvolí správná konfiguraci 1024 pulzů za otáčku, signálové stopy A, B a R. Po tomto zvolení je automaticky TIA portálem přidán kompatibilní sensorový modul SMC30 a připojen k DDM. Tím je drive-cliq topologie nastavena, protože použitý SLM nemá drive-cliq rozhraní a s řídicí jednotkou komunikuje napřímo pomocí DI/DO.

Následně je potřeba správně parametrizovat měnič – řídicí jednotku a oba motorové moduly, které reprezentují osu pohonu. U řídicí jednotky jsou potřeba pouze 2 změny oproti původnímu nastavení. Je potřeba nastavit parametr p728.8 tak, aby vstupně výstupní modul DI/DO 8 byl

nastaven na výstup. Jeho pomocí se bude ovládat trojfázový stykač, který je připojen mezi síťové napětí a tlumivku před SLM.

Dalším krokem je nastavení obou os měniče. Osa měniče je definovaná typem řízení, žádanou hodnotou (*setpoint*), limitními hodnotami, rampou – tedy jak rychle reaguje motor na změnu požadavku, bezpečnostními funkcemi a technologickými funkcemi. První nastavovanou osou bude osa, která se má chovat jako rychlostní osa a má reprezentovat motor, u kterého budou studenti ladit otáčkový regulátor. Je potřeba nastavit typ řízení na rychlostní s enkodérem. Nastavit limitní hodnoty motoru, jako například rychlost, proud, a dobu náběhu a doba poklesu. Následně je potřeba nastavit referenční hodnoty. Poté je třeba nastavit, že signál *infeed enable*, který podává informaci a provozuschopnosti SLM, je připojen k řídicí jednotce na vstup DI/DO 12. U generátoru rampy je potřeba nastavit parametr p1122 na logickou hodnotu 1, což znamená, že rampový generátor nebude brán v úvahu. Toto je nastaveno proto, že rampa bude generována pomocí technologického objektu v PLC

V tuto chvíli je vhodné uložit nastavené parametry a nahrát je do měniče. Po nahrání je vhodné provést měření, která slouží k optimalizaci některých parametrů. Nejprve se provádí měření k identifikaci parametrů motoru připojeného k měniči, u tohoto měření se motor neotáčí. Měření lze provést v režimu online v sekci „*rotate and optimaze*“. Následně je vhodné provést také měření, které slouží k optimalizaci parametrů rychlostního regulátoru. U tohoto měření se motor již bude otáčet, proto je potřeba se před měřením ujistit, že v blízkosti motoru nikdo není. Po provedení těchto měření je potřeba uložit nově nastavené parametry z paměti RAM do paměti ROM, aby se parametry uchovaly i po restartování měniče, což se dělá pomocí tlačítka „*Retentively save the data of the complete device*.“ Následně je potřeba nastavit bezpečnostní funkce měniče. Bezpečnostní funkce měniče je potřeba měnit v online režimu kdy je uživatel připojen k měniči. Bezpečnostní parametry se mění, když je měnič ve specifickém stavu (konkrétně parametr p10 je nastaven na 95 = *safety integration commissioning*). Do tohoto stavu se lze dostat např. stisknutím tlačítka „*start editing*“ v TIA portále v části *safety intagrated*. Pro tuto aplikaci nám stačí bezpečné zastavení motoru a udržení motoru v zastaveném stavu. K tomu stačí funkce STO a SS1, které jsou popsány v kapitole 1.4.1.3. Tyto funkce jsou součástí základních funkcí, není tedy třeba nastavovat rozšiřující funkce. Řízení bezpečnostních funkcí je nastaveno pomocí komunikačního protokolu PROFIsafe, který je integrován v komunikaci Profinet. V dalším kroku je nastaveno, za jak dlouho bude požadováno testovací zastavení, heslo pro bezpečnostní funkce a zmáčkneme tlačítko k ukončení editace bezpečnostních funkcí. Heslo pro bezpečnostní funkce je „0“. Následně jsou uloženy parametry pomocí tlačítka „*Retentively save the data of the complete device*“ a měnič je restartován. V tuto chvíli je osa nastavena, avšak „offline“ a „online“ parametry se liší, proto je potřeba stáhnout parametry z měniče do „offline“ projektu a projekt uložit.

U druhé osy se bude principiálně postupovat stejně. Je však potřeba si uvědomit, že zde budou určité rozdíly, což je způsobené tím, že oba motory mají spřaženou osu a tento motor se má chovat jako zátěž se zátěžným momentem daným podle požadované závislosti (závislosti na otáčkách). Toho lze dosáhnout dvěma způsoby. První možnost je, že je u této osy nastaveno momentové řízení a moment je žádán podle charakteristiky, která bude spočtena v PLC. Druhou možností je nastavit osu jako rychlostní. Pak je však třeba si uvědomit, že rychlostní regulátory obou motorů by se snažily zasahovat do řízení rychlosti společné osy, což by způsobovalo problémy. Proto u této možnosti je potřeba nastavit požadovanou rychlost nulovou po celou dobu provozu a zároveň nastavit momentové omezení. To způsobí, že regulátor se bude neustále snažit dosáhnout nulové rychlosti, jakmile budou nenulové otáčky, a bude vytvářet zátěžný moment, který bude celou dobu velikosti momentového omezení měniče. Pokud bude toto momentové omezení aktivně měněno podle požadované charakteristiky vypočítané v PLC, pak je dosaženo požadovaného chování. Zvoleno bylo právě toto řešení.

## 4.2 HW konfigurace PLC

Podobně jako měnič, i PLC je potřeba správně nastavit. V tuto chvíli je v projektu pouze řídicí jednotka CU 1500SP F-1 PN, je však potřeba přidat z HW katalogu modul s bezpečnostními digitálními vstupy a server modul, který uzavírá vnitřní napájecí sběrnici PLC. Následně je potřeba nastavit bezpečnostní modul. V sekci vlastností modulu jsou nastaveny požadované parametry, například co se má stát, pokud dojde k chybě. Nejdůležitější je nastavit, jak má modul vyhodnocovat bezpečnostní kanály. Zde je připojené jedno bezpečnostní tlačítko, které má dva kanály, které jsou současně ve vypnutém nebo zapnutém stavu v závislosti na tom, zda je tlačítko zmáčknuté či nikoliv. To znamená, že pro kanály, ke kterým jsou připojeny kabely od tohoto tlačítka je nutné nastavit tzv. 1oo2 ekvivalentní evaluaci, tedy to, že pokud se tyto dva signály neshodují déle, než je stanovená doba, hlásí modul chybu.

Co se týče řídicí jednotky, zde lze nastavit například zda mají být použité bezpečnostní funkce či nikoliv, délka komunikačního cyklu, možné zatížení komunikace, alarmy, web server, zabezpečení nahraného programu anebo lze také povolit využití systémových bitů a časovačů. K zabezpečení programu bylo pro PLC zvoleno heslo „Siemens123!“. Díky tomu není možné editovat program, dokud není zadáno správné heslo. Toto heslo slouží následně i jako heslo pro bezpečnostní funkce v PLC.

## 4.3 Nastavení komunikace

Komunikace mezi měničem a PLC probíhá pomocí PROFIdrive a PROFIsafe, které jsou součástí komunikace Profinet. Přenos dat probíhá pomocí tzv. telegramů. Nastavení telegramů nalezneme v nastavení řídicí jednotky měniče. Lze použít buď některý ze standardizovaných telegramů, nebo v případě potřeby využít tzv. free telegram, kde si uživatel může nastavit libovolná data, která chce přenést mezi měničem a PLC. V této aplikaci řídicí jednotka komunikuje s PLC pomocí telegramu 390, který v sobě obsahuje informaci o aktuálních hodnotách DI/DO rozhraní, případně požadavek hodnoty daného DI/DO, které je nastavené v měniči jako digitální výstup. To je případ např. pro DI/DO 8, pomocí kterého lze takto z PLC programu spínat stykač, který připojuje měnič k napájecí síti. Pro obě osy je použit standardní telegram 2, což je telegram, který je určen k přenosu požadavku rychlosti s 32bitovým rozlišením. Dále je pro obě osy použit rozšiřující Siemens telegram 750. Tento telegram přenáší informaci o aktuální hodnotě momentu a umožňuje zadávat měniči požadavek na momentové limity. Toto je potřeba především pro osu, která má zastávat funkci proměnné zátěže. Dále je použit pro obě osy bezpečnostní telegram PROFIsafe 30. Díky tomuto telegramu může PLC využívat bezpečnostní funkce měniče. Hodnota bezpečnostních vstupů, tedy vstupů, které vyhodnocují stav bezpečnostního tlačítka, je přenesena do bezpečnostního vstupního modulu v PLC, následně vyhodnocena PLC a bezpečně předána měniči pomocí PROFIsafe telegramu. Aby bylo možné zobrazit na HMI panelu některé další informace, které nejsou posílány ve standardních telegramech, je v této aplikaci použit také free telegram, jehož pomocí jsou posílány například informace o aktuální hodnotě napětí stejnosměrného meziobvodu, aktuální velikosti proudu obou motorových modulů, nebo třeba teplota obou motorů.

The screenshot shows the 'Telegram configuration' window in the Siemens S120 software. The interface is divided into three main sections: Drive control-Telegramms (Item 1), Motor-Telegramms (Item 2), and Load-Telegramms (Item 3). Each section lists various telegram types and their parameters.

Name	Item	Link	Telegram	Length	Extension	Type	Partner	Partner data a...
<b>Drive control-Telegramms</b>								
Send (Actual value)	1	→	SIEMENS telegram 390	2	words: 0	words: →	CD	PLC_1 I 256...259
Receive (Setpoint)	1	←	SIEMENS telegram 390	2	words: 0	words: ←	CD	PLC_1 Q 256...259
<b>Motor-Telegramms</b>								
Send Safety Integrated tel.	2	→	PROFIsafe telegram 30	6	bytes: —	bytes: →	F-CD	PLC_1 I 288...293
Receive Safety Integrated ...	2	←	PROFIsafe telegram 30	6	bytes: —	bytes: ←	F-CD	PLC_1 Q 288...293
Send (Actual value)	2	→	Standard telegram 2	4	words: 0	words: →	CD	PLC_1 I 274...281
Receive (Setpoint)	2	←	Standard telegram 2	4	words: 0	words: ←	CD	PLC_1 Q 274...281
Send Extension (Actual va...	2	→	Free telegram	6	words: —	words: →	CD	PLC_1 I 300...311
Send Torque (Actual value)	2	→	SIEMENS telegram 750	1	words: —	words: →	CD	PLC_1 I 282...283
Receive Torque (Setpoint)	2	←	SIEMENS telegram 750	3	words: —	words: ←	CD	PLC_1 Q 282...287
<b>Load-Telegramms</b>								
Send Safety Integrated tel.	3	→	PROFIsafe telegram 30	6	bytes: —	bytes: →	F-CD	PLC_1 I 294...299
Receive Safety Integrated ...	3	←	PROFIsafe telegram 30	6	bytes: —	bytes: ←	F-CD	PLC_1 Q 294...299
Send (Actual value)	3	→	Standard telegram 2	4	words: 0	words: →	CD	PLC_1 I 260...267
Receive (Setpoint)	3	←	Standard telegram 2	4	words: 0	words: ←	CD	PLC_1 Q 260...267
Send Extension (Actual va...	3	→	Free telegram	4	words: —	words: →	CD	PLC_1 I 320...327
Send Torque (Actual value)	3	→	SIEMENS telegram 750	1	words: —	words: →	CD	PLC_1 I 268...269

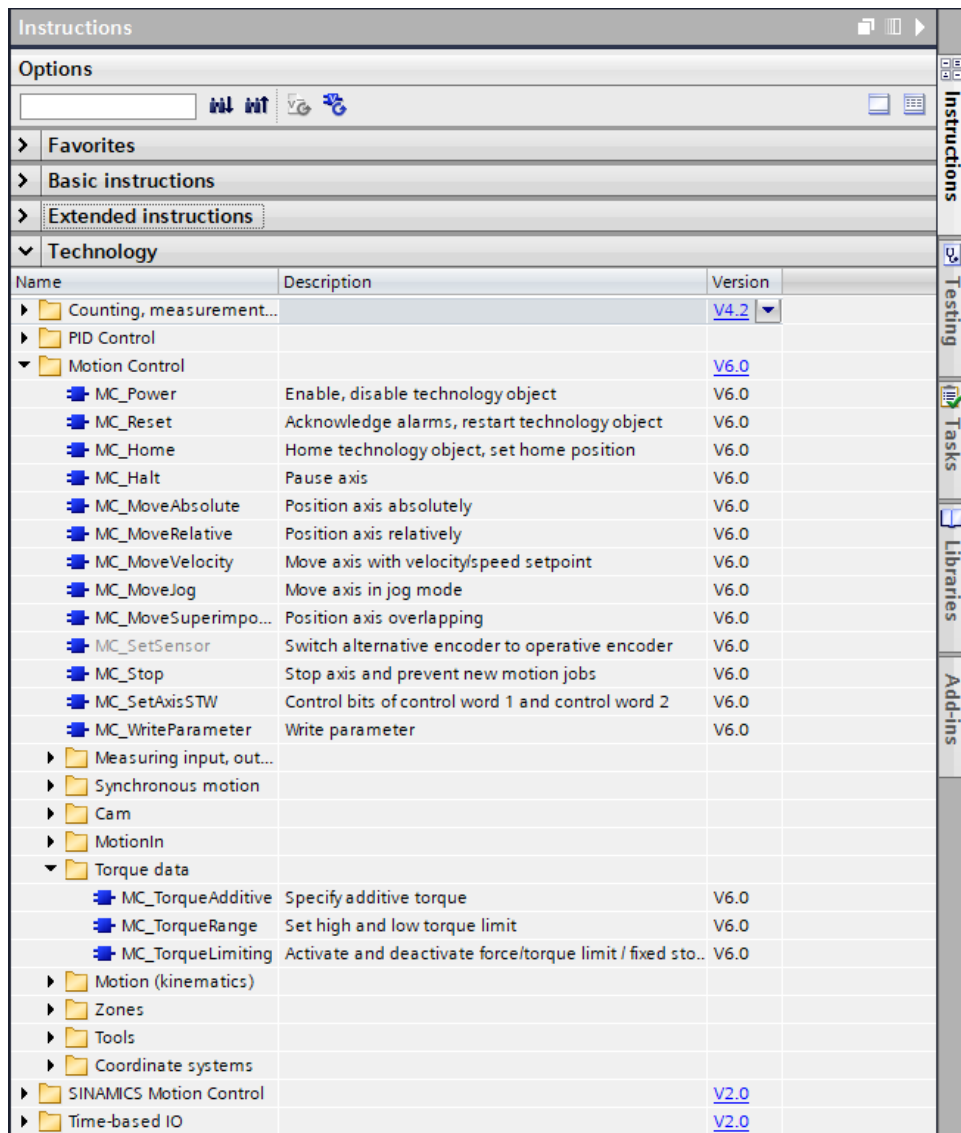
Obr. 4-1 Struktura použitých komunikačních telegramů.

Následně je potřeba nastavit délku cyklu komunikace a režim komunikace IRT. Délka cyklu je nastavena ve vlastnostech PLC v záložce Profinet rozhraní X1, zde je nastaven cyklus 4 ms. Také je potřeba zkontrolovat, že je nastavena synchronizace a že PLC má roli „*sync master*“. Pak je potřeba správně zadat topologii zapojení, bez té není možné provozovat IRT komunikaci. Topologii vytvoříme v záložce „*topology view*“ v části „*Devices & networks*“. Následně je nastavena IRT komunikaci v pokročilých možnostech měniče.

#### 4.4 Technologické objekty

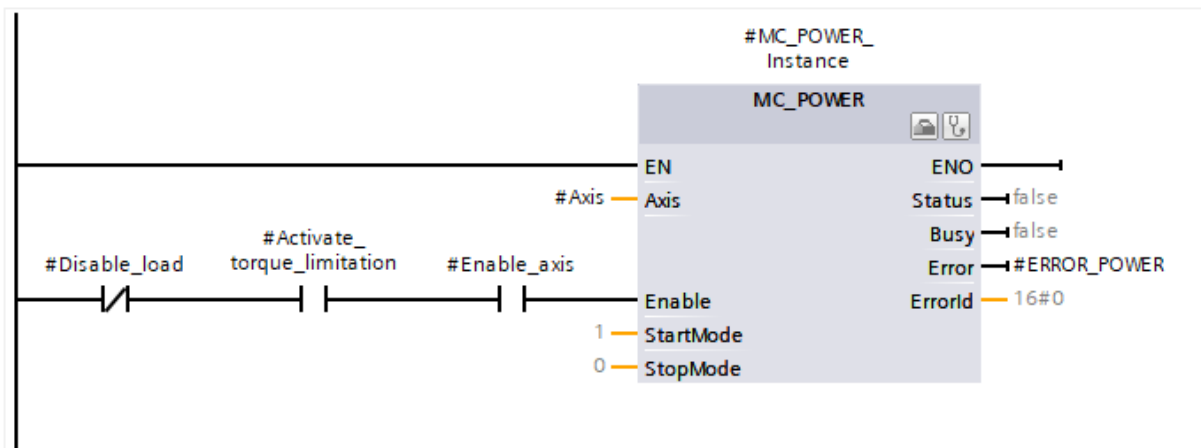
K tomu, aby bylo možné začít programovat řídicí program PLC je potřeba vytvořit technologické objekty. Technologické objekty (TO) jsou softwarové objekty uvnitř PLC, které reprezentují mechanické komponenty (typicky motory připojené k měniči, které se chovají jako rychlostní, polohové nebo synchronní osy) a umožňují jednoduchou konfiguraci a diagnostiku. Jsou součástí centrálního řízení, které bylo popsáno v kapitole 1.6. Technologický objekt lze vytvořit v PLC v projektovém stromu v záložce technologické objekty. Protože v tomto projektu je potřeba ovládat oba motory rychlostně, byly vytvořeny dva TO typu rychlostní osa. Po vytvoření TO je potřeba ho nakonfigurovat. Konfigurace obsahuje zvolení jednotek pro měření veličin, zvolení komunikace mezi měničem a PLC (zde PROFIdrive), dynamické vlastnosti rychlostní osy a dynamické limity. Protože již byla v měniči vytvořena struktura komunikačních telegramů mezi měničem a PLC, stačí v konfiguraci zvolit standardní telegram 2 a doplňující telegram s informacemi o momentu telegram 750 u obou os. Následně je potřeba zvolit stejné referenční hodnoty rychlosti a momentu, které byly nastaveny v měniči, aby PLC správně interpretovalo přijímaná data. Také je potřeba nastavit dynamické vlastnosti os, tedy dynamické chování výchozí, limitní a při požadavku na nouzové zastavení. Vzhledem k tomu, že cílem stanoviště je sledovat přechodové charakteristiky, tedy reakce na skok žádané hodnoty, je potřeba zvolit tyto limity velmi dynamicky. V ideálním případě by požadavek nebyl tvořen rampou, ale skokovou žádostí, avšak to není možné, proto to obcházíme velmi krátkou dobou náběhu a poklesu. Výsledná rampa je poté velmi podobná skokové změně.

Toto je však pouze konfigurace technologických bloků. Pokud máme takto konfigurované osy uvnitř technologických bloků ovládat, je potřeba využít dalších bloků integrovaných v PLC. Tyto bloky se nazývají MC, což je zkratka *motion control*, tedy ovládání pohybu. Tyto bloky lze najít v katalogové části *technology*, jak lze vidět na Obr. 4-2. Detailně jsou popsány pouze použité bloky. K ovládání osy reprezentující motor používám bloky MC\_Power, MC\_Reset, MC\_MoveVelocity a MC\_Stop. K ovládání osy reprezentující zátěž jsou použity bloky MC\_Power, MC\_Reset a MC\_TorqueRange. Tyto bloky jsou, co se struktury týče funkční bloky.



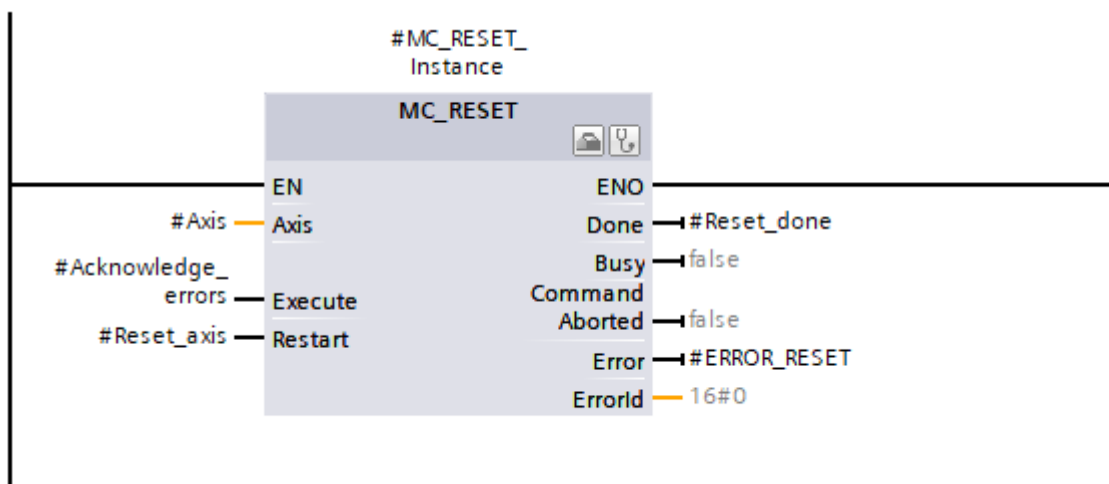
Obr. 4-2 Motion control bloky v katalogové části TIA portálu

MC\_Power je blok, který povoluje či zakazuje ovládaný technologický objekt a podle potřeby řízenou osu zapíná nebo vypíná. Lze jím ovládat rychlostní osu, polohovou osu, synchronní osu a externí enkodér. Na Obr. 4-3 je vidět struktura bloku. Je potřeba blok přiřadit správné ose pomocí vstupu *axis*. Vstup „Enable“ rozhoduje, zda je ovládaný technologický objekt zapnutý či vypnutý. Start mode rozhoduje, zda se jedná o polohovou osu či nikoliv (v našem případě nikoliv). „Stop mode“ rozhoduje, po jaké rampě dojde k zastavení ovládaného technologického objektu v případě, že vyšleme příkaz k vypnutí objektu a objekt není v klidu. „Stop mode“ 0 znamená, že se má zastavit po stejně jako při nouzovém zastavení. Výstup „Error“ informuje o tom, zda došlo k chybě během používání tohoto bloku a pokud ano, detailnější informace o chybě nalezneme podle čísla ve výstupu „ErrorId“.



Obr. 4-3 Blok MC\_power (pro osu reprezentující zátěž)..

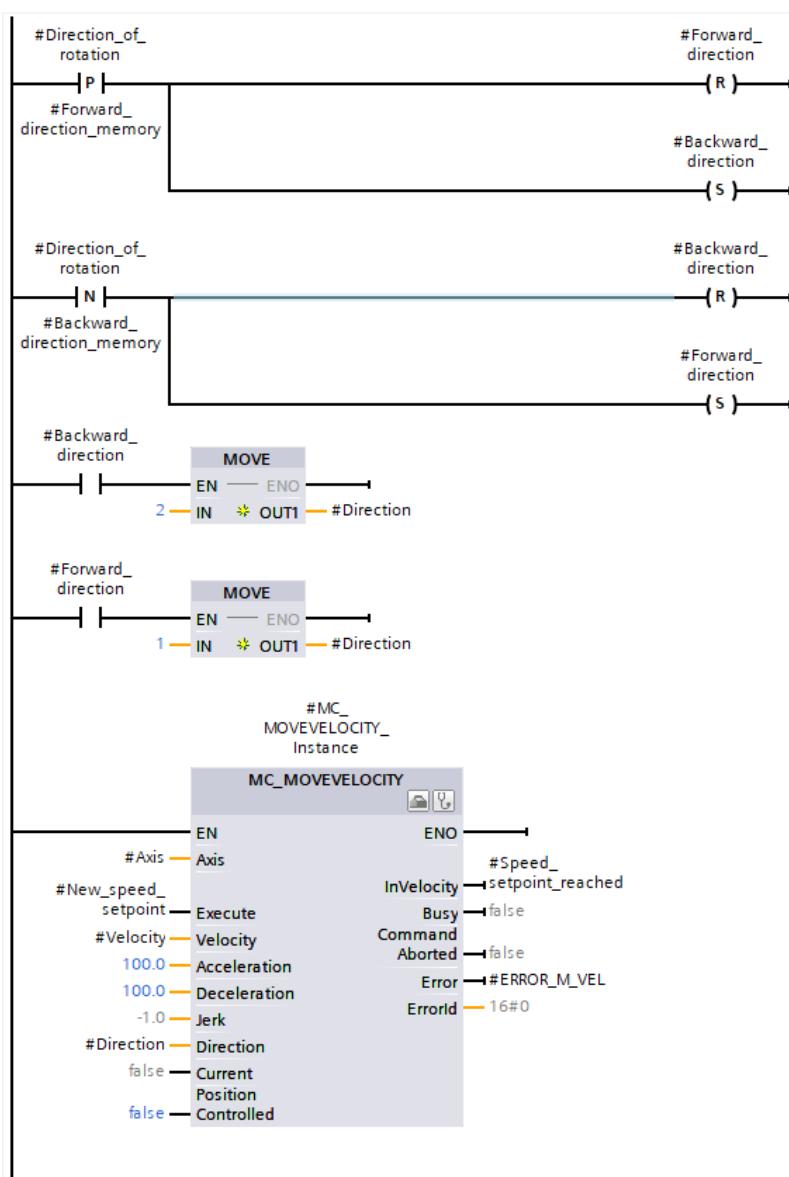
MC\_Reset je určen k potvrzení všech chyb (chyb, které lze potvrdit) týkajících se daného technologického objektu. Tento blok lze použít pro všechny technologické objekty. Nejprve je potřeba blok přiřadit správné ose pomocí vstupu „Axis“. Vstup „Execute“ provede potvrzení všech chyb, přičemž potvrzení se provede na náběžnou hranu. Vstup „Restart“ provede re-inicializaci technologického objektu a potvrdí všechny aktuální alarmy technologického objektu. Co se výstupů týče, výstup „Done“ informuje o tom, zda požadovaná akce byla provedena. Výstup „Busy“ informuje o tom, že operace stále probíhá.



Obr. 4-4 Blok MC\_Reset.

MC\_MoveVelocity umožňuje roztočit danou osu na konstantní otáčky a dle potřeby požadavek otáček měnit. Tento blok lze použít pro rychlostní, polohovou a synchronní osu. K tomu, aby bylo možné osu otáčet požadovanou rychlostí je nejprve nutné povolit technologický objekt pomocí bloku MC\_Power a zároveň musí být technologický objekt správně konfigurován. Náběžnou hranou vstupu „Execute“ dojde k aktivaci požadavku na roztočení osy na požadovanou rychlost zadanou vstupem „VelocityL“. Vstupy „Acceleration“, „Deceleration“ a „Jerk“ umožňují nastavit požadované dynamické vlastnosti. V případě, že hodnota těchto vstupů je záporná, jsou

použity hodnoty z konfigurační části technologického bloku popsané výše. Vstup „Direction“ nastavuje směr otáčení. Hodnota 0 znamená, že znaménko hodnoty vstupu „Velocity“ určuje směr otáčení. Hodnota 1 požaduje kladný směr otáčení, hodnota 2 záporný směr otáčení. Parametr „Current“ umožňuje ignorovat hodnoty vstupů „Direction“ a „Velocity“ a ponechat požadavek rychlosti stejný jako v době, kdy byl tímto vstupem vydán požadavek k udržování aktuální rychlosti (logická hodnota 1 vysílá požadavek na udržování aktuální rychlosti). Vstup „PositionControlled“ informuje o tom, zda jde o aplikaci s polohovou osou. Jediný výstup, který ještě nebyl popsán a není principiálně stejný dříve popsáným MC blokům je „CommandAborted“, což je výstup, který informuje o tom, že došlo k přerušení funkce bloku MC\_MoveVelocity kvůli zavolání jiného příkazu s vyšší prioritou.

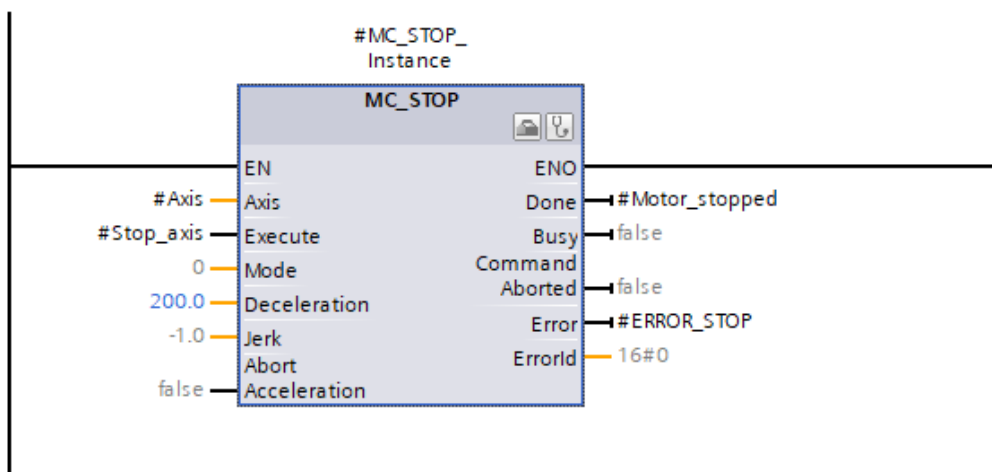


Obr. 4-5 Blok MC\_MoveVelocity použit k ovládání rychlostní osy zastávající funkci motoru.



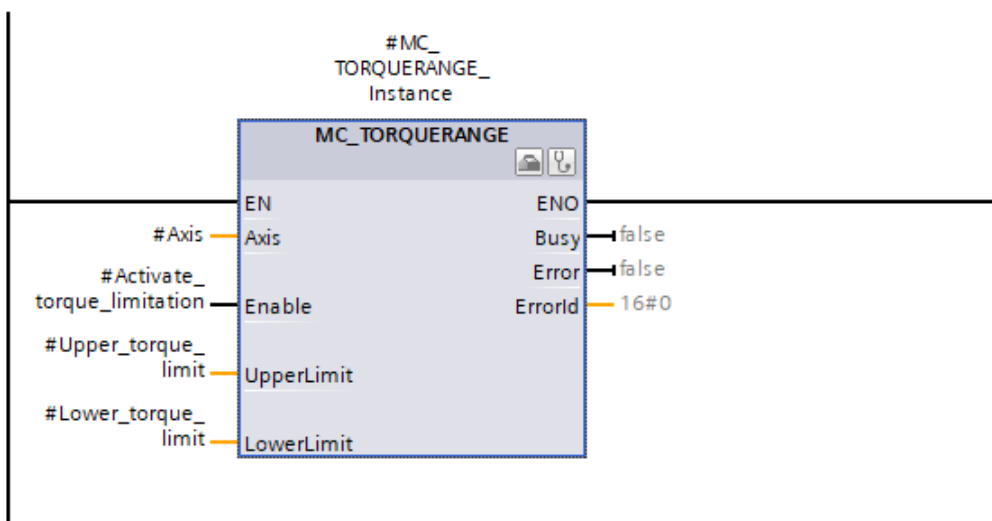
Blok MC\_MoveVelocity u zátěže není použit, protože u zátěže je potřeba, aby měla požadavek na otáčky po celou dobu nulový, což je však výchozí hodnota při zapnutí měniče pomocí bloku MC\_power.

Blok MC\_Stop vysílá požadavek na zastavení osy a po dobu, co je tento požadavek aktivní, nelze uvést osu znovu do pohybu. Tento požadavek se spouští náběžnou hranou vstupu „Execute“ a po dobu hodnoty logické 1 je vydán požadavek na zastavení osy. Vstup „Mode“ určuje dynamické chování požadovaného zastavení. Hodnota 0 nastaví stejnou dynamiku jako při nouzovém stopu, hodnota 2 nastaví dynamiku podle maximálních limitů v konfiguraci technologického objektu a hodnota 3 nastaví dynamiku podle aktuálních hodnot vstupů „Deceleration“ a „Jerk“. Tento blok je použit pouze pro osu, která zastává charakter motoru, osa s charakterem zátěže má požadavek rychlosti po celou dobu nulový, proto tento blok není použit.



Obr. 4-6 Blok MC\_Stop.

Blok MC\_TorqueRange umožňuje pomocí telegramu 750 posílat měniči dolní a horní limit momentu, který je specifikován hodnotami vstupů „UpperLimit“ a „LowerLimit“. Hodnoty těchto vstupů jsou použity pouze v případě, že má vstup „Execute“ logickou hodnotu 1.



Obr. 4-7 Blok MC\_TorqueRange.

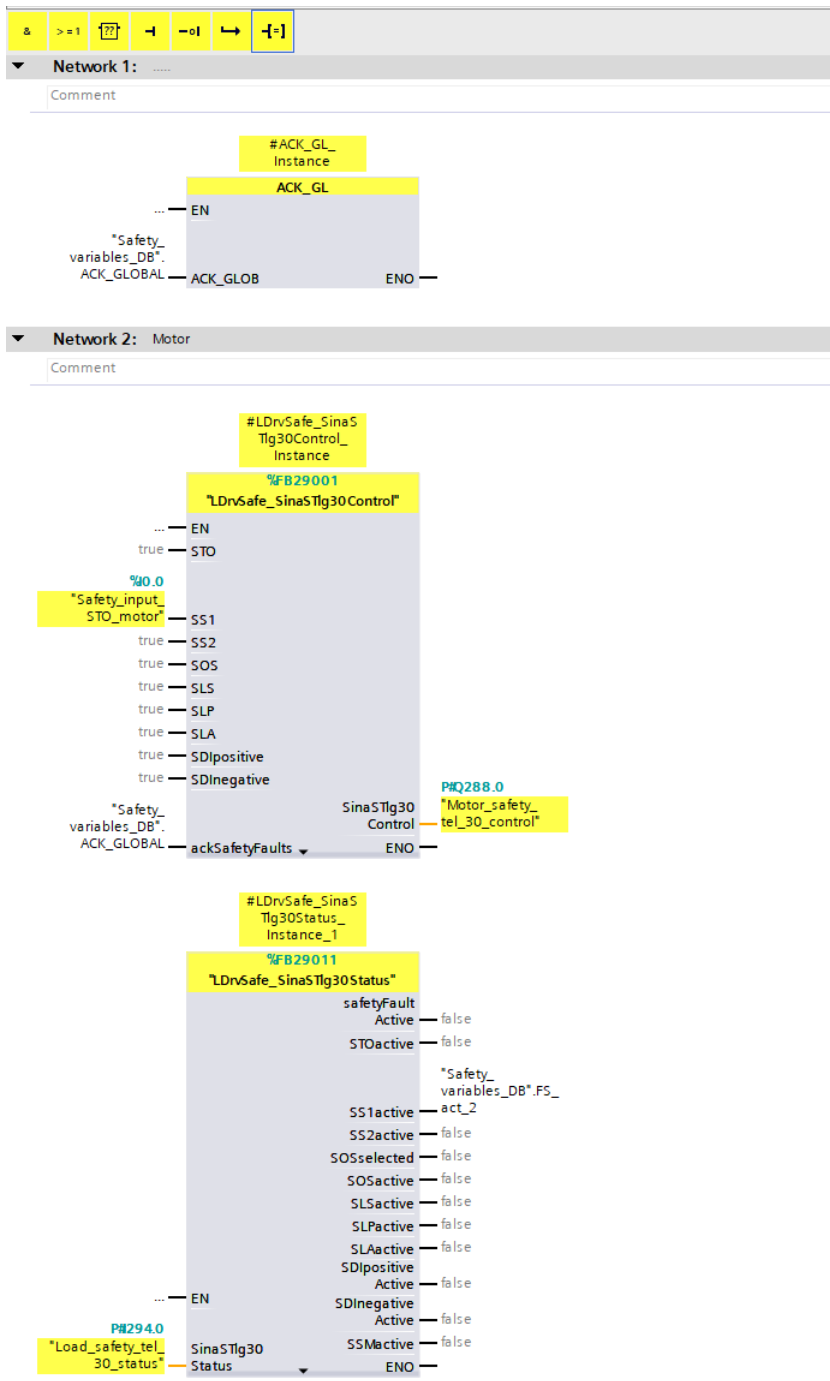
## 4.5 Bezpečnostní program

K tomu, aby správně fungovaly bezpečnostní funkce je potřeba implementovat bezpečnostní program. Firma Siemens poskytuje standardizované bloky, s jejichž pomocí lze bezpečnostní program lehce vytvořit. Tyto bloky jsou součástí knihovny LDrvSafe, která je dostupná ke stažení na stránkách Siemensu. Knihovnu lze vložit do TIA portálu v pravé části katalogů a knihoven, pokud jsou rozkliknuty „Libraries“, „Global libraries“ a je kliknuto na tlačítko „open global library“. Protože použitý měnič je z řady SINAMICS S a použitý bezpečnostní telegram je PROFIsafe 30, je potřeba importovat do PLC datových typů datové typy LDrvSafe\_SinaSTlg30Status a LDrvSafe\_SinaSTlg30Control a také přidat stejnojmenné funkční bloky. Tyto funkční bloky jsou použity k tomu, aby bylo možné využít bezpečnostních funkcí měniče. Tyto funkční bloky jsou vloženy do funkčního bloku Main\_Safety\_RTG1, což je blok, ve kterém musí být vytvořen bezpečnostní program. Bezpečnostní program nelze vytvořit uvnitř hlavního bloku OB1(main). Po vložení je potřeba těmto funkčním blokům přiřadit správný bezpečnostní telegram PROFIsafe 30. K tomu je potřeba vytvořit si PLC tagy, které budou mít datový typ LDrvSafe\_SinaSTlg30Status a LDrvSafe\_SinaSTlg30Control. Zároveň musí být přiřazeny k správným adresám. Adresy, na kterých nalezneme telegram PROFIsafe 30 najdeme v nastavení telegramů měniče.

Name	Item	Link	Telegram	Length	Extension	Type	Partner	Partner data area
▼ Drive control-Telegrams: 1								
Send (Actual value)			SIEMENS telegram 390	2	words	→	CD	PLC_1 I 256..259
Receive (Setpoint)			SIEMENS telegram 390	2	words	←	CD	PLC_1 Q 256..259
<Add telegram>								
▼ Motor-Telegrams: 2								
Send Safety Integrated tel.			PROFIsafe telegram 30	6	bytes	→	F-CD	PLC_1 I 288..293
Receive Safety Integrated ..			PROFIsafe telegram 30	6	bytes	←	F-CD	PLC_1 Q 288..293
Send (Actual value)			Standard telegram 2	4	words	→	CD	PLC_1 I 274..281
Receive (Setpoint)			Standard telegram 2	4	words	←	CD	PLC_1 Q 274..281
Send Extension (Actual va..			Free telegram	6	words	→	CD	PLC_1 I 300..311
Send Torque (Actual value)			SIEMENS telegram 750	1	words	→	CD	PLC_1 I 282..283
Receive Torque (Setpoint)			SIEMENS telegram 750	3	words	←	CD	PLC_1 Q 282..287
<Add telegram>								
▼ Load-Telegrams: 3								
Send Safety Integrated tel.			PROFIsafe telegram 30	6	bytes	→	F-CD	PLC_1 I 294..299
Receive Safety Integrated ..			PROFIsafe telegram 30	6	bytes	←	F-CD	PLC_1 Q 294..299
Send (Actual value)			Standard telegram 2	4	words	→	CD	PLC_1 I 260..267
Receive (Setpoint)			Standard telegram 2	4	words	←	CD	PLC_1 Q 260..267
Send Extension (Actual va..			Free telegram	4	words	→	CD	PLC_1 I 320..327
Send Torque (Actual value)			SIEMENS telegram 750	1	words	→	CD	PLC_1 I 268..269
Receive Torque (Setpoint)			SIEMENS telegram 750	3	words	←	CD	PLC_1 Q 268..273

Obr. 4-8 Adresy PROFIsafe telegramů

Vytvořené PLC tagy jsou přiřazeny k funkčním blokům. Následně u řídicího bloku lze přidat proměnné podle potřeby, které budou ovládat bezpečnostní funkce. V tomto projektu je použita pouze funkce SS1. Příklad provedení bezpečnostního programu pro jednu osu je na Obr. 4-9.



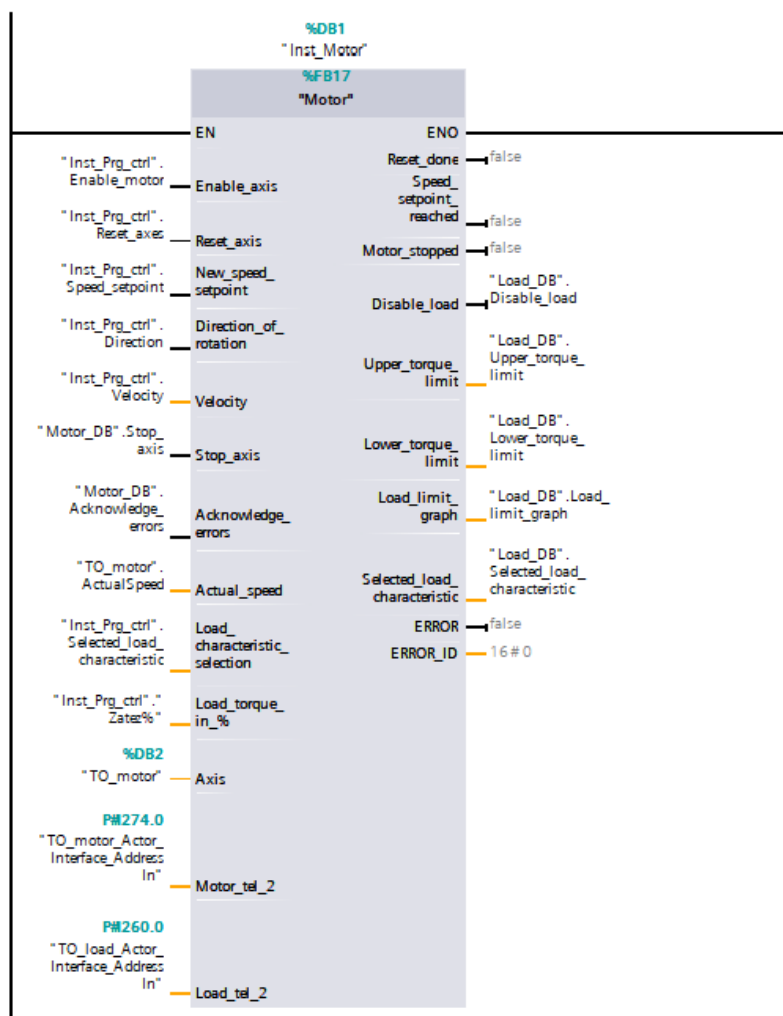
Obr. 4-9 Main safety – příklad bezpečnostního programu pro motor.

## 4.6 Popis použitých funkčních bloků

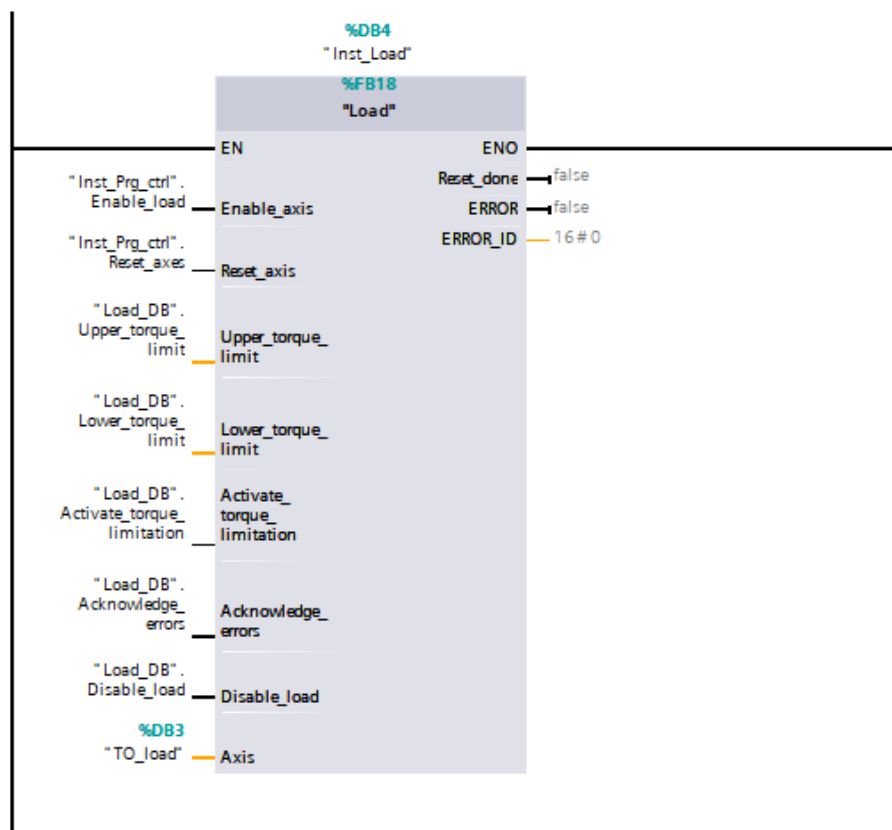
V této kapitole budou popsány funkční bloky vytvořené k řízení PLC. Funkční bloky budou vysvětleny pouze principiálně, jelikož popis jednotlivých bloků dopodrobna by byl zbytečně zdouhavý a k pochopení realizace programu není potřeba. Logika řídicího programu, který následně využívá FB popsané v této kapitole bude popsána v kapitole Kapitola 5:

### 4.6.1 FB Motor a Load

Jak bylo popsáno v kapitole 4.4, k ovládání os technologických objektů jsou potřeba MC bloky. Pro lepší přehlednost jsou v hlavním programu PLC vytvořeny dva funkční bloky, pojmenované *Motor* a *Load*, kdy každý z těchto funkčních bloků má uvnitř dříve popsané MC bloky, potřebné k ovládání osy, kterou blok reprezentuje. Bloky *Motor* a *Load* lze vidět na Obr. 4-10 a Obr. 4-11. Aby byla zajištěna správná funkce obou os, je uvnitř bloků zavedena následující logika. Nejprve se vždy zapíná osa reprezentující zátěž. Toto je možné pouze, pokud je zapnuté momentové omezení s pomocí bloku MC\_TorqueRange a pokud není osa zakázána (lze vidět na Obr. 4-3 u vstupu „Enable“).



Obr. 4-10 Blok Motor, který obsahuje MC bloky k ovládání osy motoru.

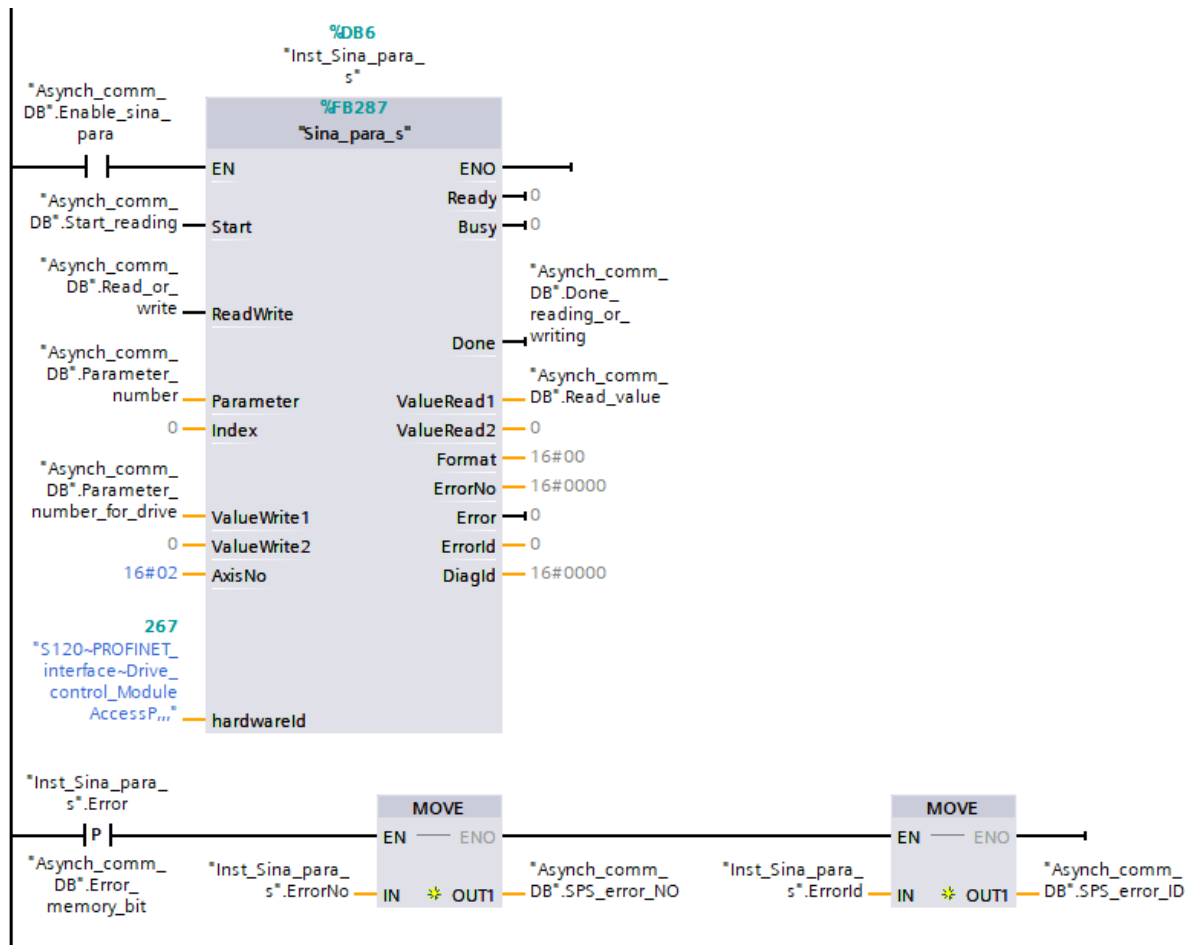


Obr. 4-11 Blok Load, který obsahuje MC bloky k ovládání osy zátěže.

Zapnutí osy reprezentující motor je podmíněné tím, že osa reprezentující zátěž je již zapnutá a že není v chybném stavu. Tato logika je uvnitř bloku Motor implementována ve funkčním bloku Safety\_motor\_control, který navíc v případě, že během měření dojde k vypnutí motoru (uživatel, nebo vlivem chyby) zakáže osu zátěže a tím ji vypne.

#### 4.6.2 Sina\_para\_s

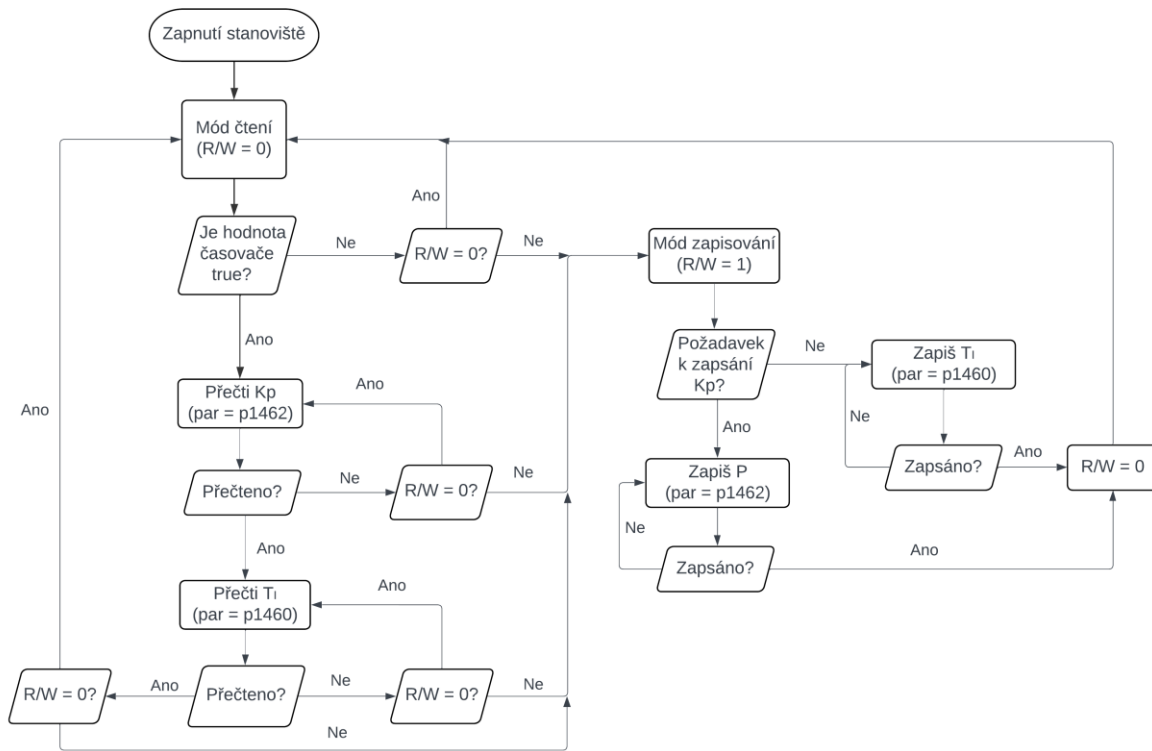
Dále je potřeba popsat funkční blok Sina\_para\_s, který se v programu vyskytuje. Tento blok umožňuje acyklické čtení nebo zapisování parametrů měniče. S pomocí tohoto bloku budou moci následně studenti měnit konstanty otáčkového regulátoru. Tento blok lze importovat z katalogové části, konkrétně v sekci *optional packages*, SINAMICS. Co se týče ovládání funkčního bloku Sina\_para\_s, náběžná hrana vstupu Start započne čtení nebo zapisování. Vstup „ReadWrite“ rozhoduje o tom, zda požadujeme zapsání do měniče, nebo přečtení aktuální hodnoty parametru v měniči. Vstup „Parameter“ udává číslo parametru, ke kterému chceme přistoupit. Vstup „Index“ je potřeba, pokud se jedná o parametr s více indexy. Vstup „ValueWrite1“ slouží k předání hodnoty, kterou chceme do měniče zapsat (datový typ *Real*). Co se výstupů týče, výstup „Done“ říká, zda zapsání či čtení již proběhlo, „ValueRead1“ v sobě nese hodnotu, která byla přečtena z parametru měniče (datový typ *Real*) a v případě chyb při komunikaci se dozvíme o chybách více v „ErrorNo“ a „ErrorId“, přičemž výstup „Error“ bude po dobu chyby mít logickou hodnotu 1.



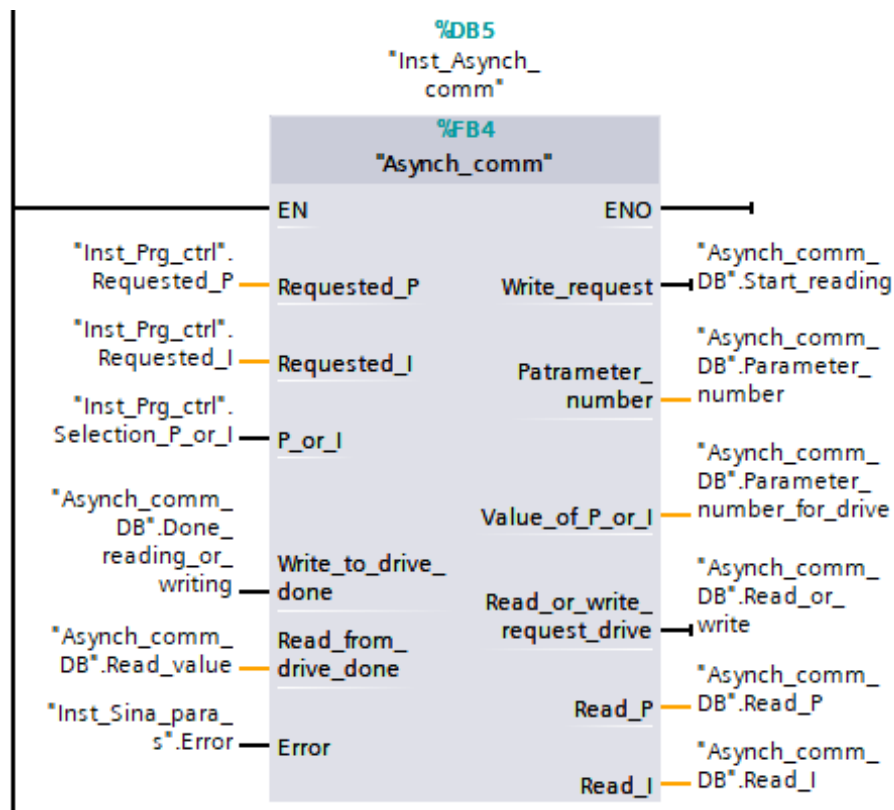
Obr. 4-12 Funkční blok `Sina_para_s`

K ovládání bloku `Sina_para_s` je vytvořen FB `Asynch_comm`, který vyhodnocuje vstupy z HMI a následně posílá požadavky bloku `Sina_para_s` tak, aby probíhala asynchronní komunikace dle očekávání uživatele. Struktura řídicího programu bloku FB `Asynch_comm` je zobrazena na vývojovém diagramu na Obr. 4-13, FB `Asynch_comm` na Obr. 4-14. Vidíme, že struktura je rozdělena na mód, kdy je cyklicky požadováno přečtení obou parametrů otáčkového regulátoru a na mód zapisování, kdy uživatel požaduje zapsání jednoho z parametrů otáčkového regulátoru. K cyklickému požadavku čtení parametrů je využit vnitřní časovač PLC nazvaný „`Clock0.5Hz`“, který má střídavě dvě sekundy logickou hodnotu 1 a 2 sekundy hodnotu 0. Tato struktura je použita proto, že parametry p1460 a p1462, tedy parametry otáčkového regulátoru nelze přidat do free telegramu. Cyklickou žádostí o přečtení parametru předcházíme situaci, kdy by acyklická komunikace selhala a uživatel neměl informaci o aktuálních hodnotách parametrů regulátoru. Existují jiné způsoby, jak toto řešit, ale zde byl použit tento postup. Požadavek na zapsání (tedy změna proměnné R/W na hodnotu 1) je vyslán, kdykoliv uživatel změní jeden z parametrů otáčkového regulátoru skrz HMI panel. V případě, že blok pro acyklickou komunikaci selže a dostane se do chybového stavu, má uživatel skrz HMI panel restartovat komunikaci pomocí tlačítka. Tlačítkem restart se zakáže a následně povolí blok `Sina_para_s`. Je potřeba restart

opakovat, dokud nedojde k vyřešení chyby, protože blok Sina\_para\_s na rozdíl od jiných bloků nemá vstup, s jehož pomocí lze potvrdit chybu.



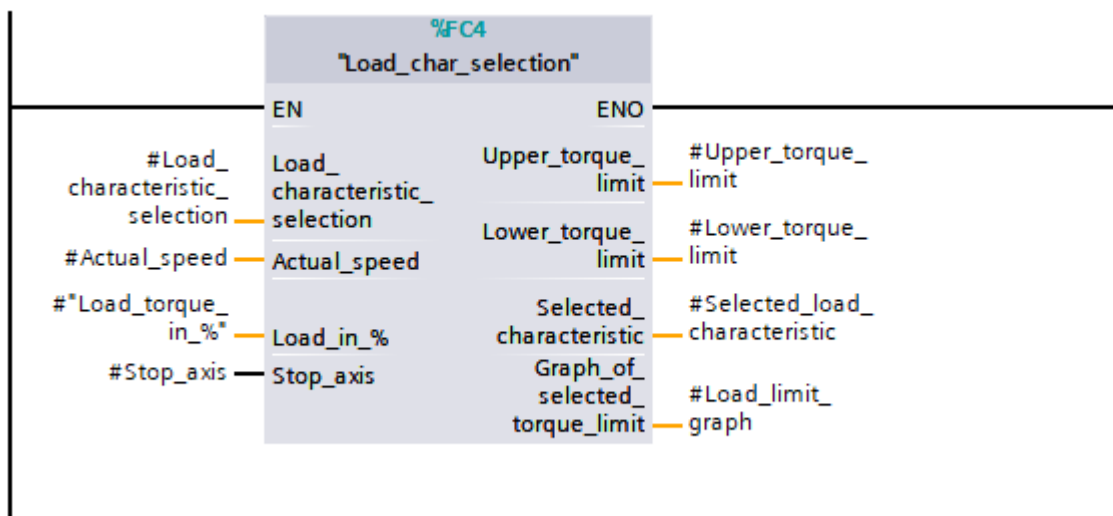
Obr. 4-13 Logika funkčního bloku, který ovládá asynchronní komunikaci pomocí Sina\_para\_s.



Obr. 4-14 FB Asynch\_comm, který ovládá FB Sina\_para\_s a tedy i acyklickou komunikaci mezi měničem a PLC.

### 4.6.3 Tvorba zátěžné charakteristiky

K tomu, aby mohli studenti volit zátěžný moment je vytvořena funkce, která výslednou charakteristiku vytvoří a požadovanou hodnotu momentu podle aktuálně zvolené charakteristiky předá dál bloku MC\_TorqueRange obsaženého uvnitř FB *Load*, který určuje velikost zátěžného momentu. Vstupními proměnnými této funkce jsou aktuální otáčky, typ požadované charakteristiky (jeden ze tří výše zmíněných) a hodnota 0 až 100 %, který říká, jak vysoká má být maximální hodnota zátěžného momentu v porovnání s jmenovitým momentem motoru (počítáno při jmenovité rychlosti motoru 1460 otáček za sekundu).

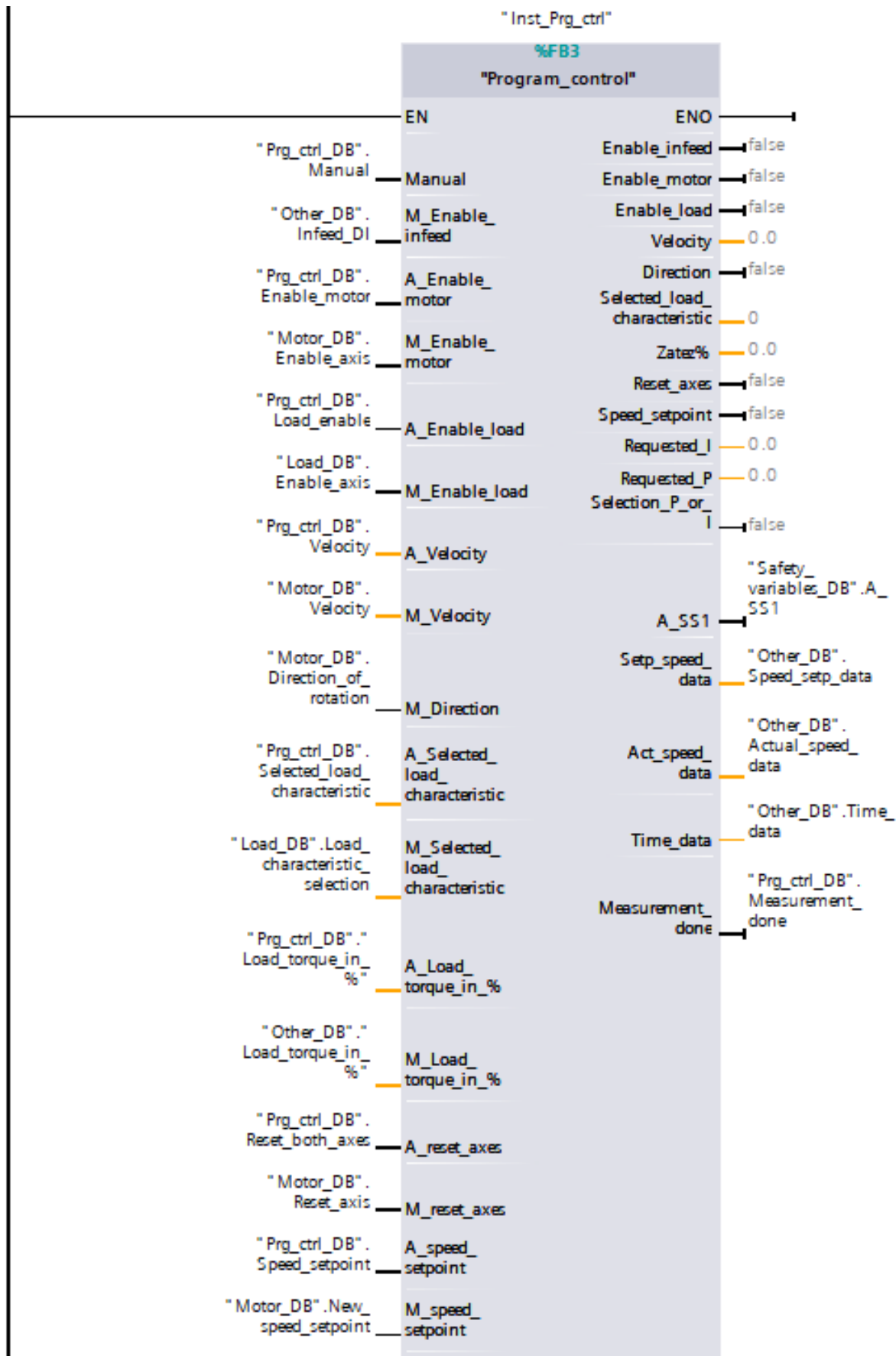


Obr. 4-15 Blok *Load\_char\_selection*, který vypočte požadovanou zátěžovou charakteristiku.

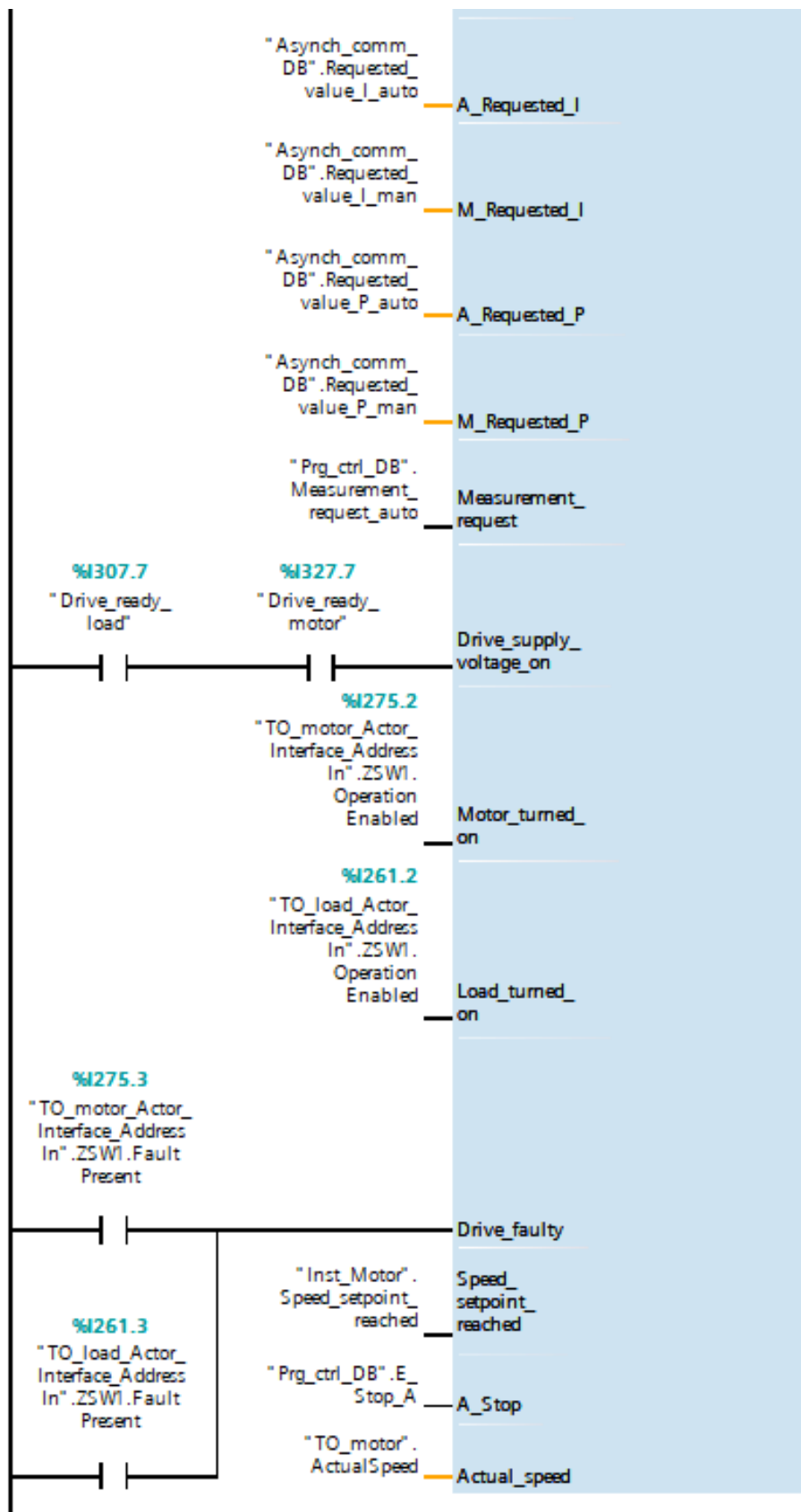
### 4.6.4 FB Program\_control

FB, který zpracovává vstupy z HMI panelu a zároveň na základě řídicí logiky rozhoduje, jak mají být ovládány dřívější popsané bloky FB Motor, FB Load a FB Asynch\_comm, je pojmenován FB Program\_control. Tento blok na základě zvolené obrazovky, a tedy i typu řídicí logiky, řídí celé stanoviště. Logika řízení, které je obsažené uvnitř tohoto bloku je popsána v kapitole Kapitola 5:. Tento blok má mnoho vstupních proměnných a výstupních proměnných, což způsobuje, že se nevejde na jednu stránku A4, proto je rozpuřen napůl na obrázcích Obr. 4-16 a Obr. 4-17.





Obr. 4-16 První polovina bloku Program\_control.



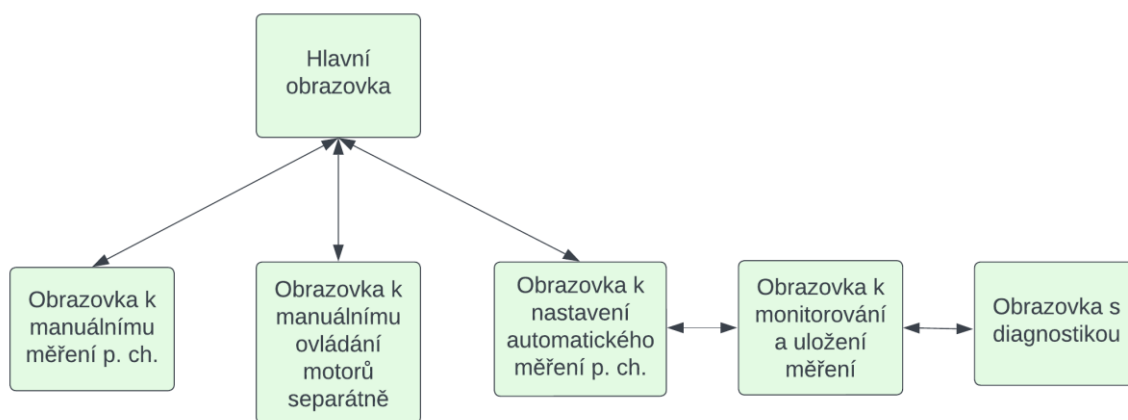
Obr. 4-17 Druhá polovina bloku Program\_control.

## 4.7 Vizualizace

Ve chvíli, kdy je hotový řídicí program, konfigurace měniče a os, tak zbývá vytvořit operátorské rozhraní pomocí HMI panelu. Stejně jako měnič a PLC, nejprve je potřeba pomocí tlačítka „add new device“ vložit příslušnou verzi HMI panelu, v našem případě MTP1500. Při vložení do projektu se objeví dialogové okno, které umožňuje provést základní konfiguraci panelu. V rámci tohoto okna lze nastavit například spojení s konkrétním PLC již existujícím v projektu, základní strukturu obrazovek HMI, vzhled obrazovky, alarmy, systémové obrazovky a tlačítka. Následně je potřeba nastavit panelu správnou IP adresu, podle potřeby lze panelu fyzicky změnit IP adresu, pokud jsme k němu připojeni. Adresa v projektu pak musí být stejná jako fyzická IP adresa panelu.

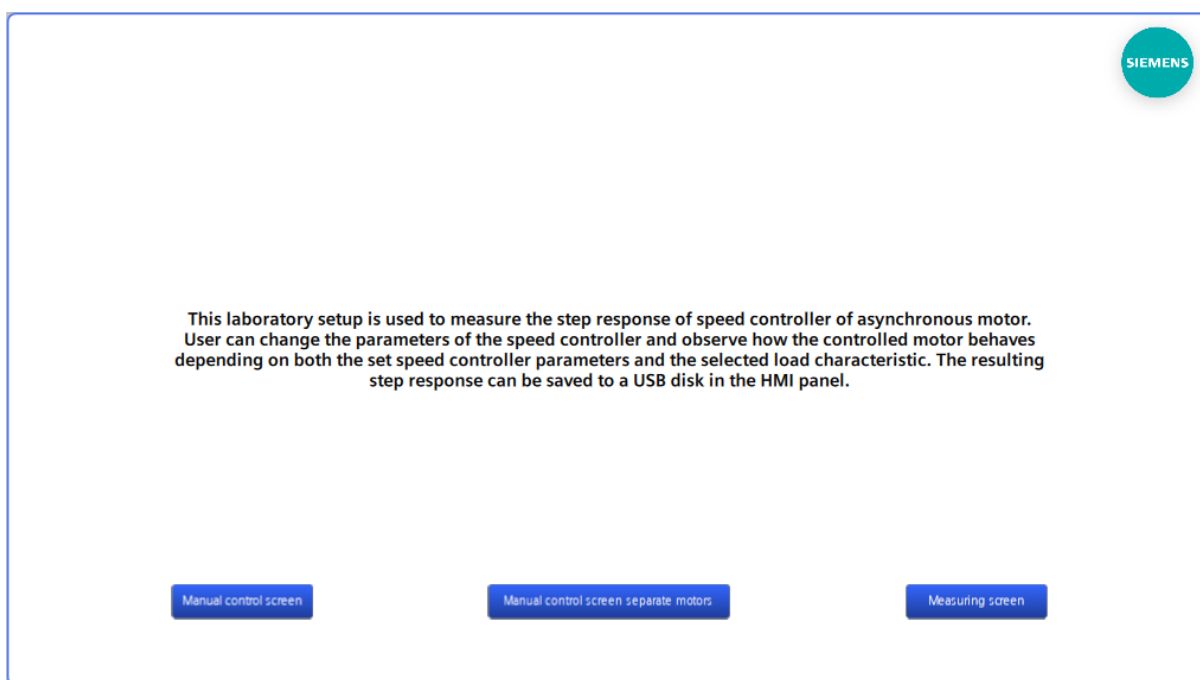
Při vytváření vizualizace lze v sekci obrazovky přidat libovolný počet obrazovek. Mezi obrazovkami se pak dá přecházet pomocí tlačítek, které uživatel umístí na obrazovku. K objektům, které lze umístit na obrazovku patří například základní geometrické obrazce, text, tlačítka, přepínače, posuvníky, měřicí ukazatele a další. Vlastnosti každého objektu lze libovolně měnit v jejich vlastnostech. K tomu, aby bylo možné připojit funkci tlačítek, posuvníků, ukazatelů a dalších objektů. K proměnným z PLC je potřeba vytvořit spojení s těmito proměnnými. Toho lze nejjednodušeji dosáhnout tak, že v projektovém stromě najdeme DB, který obsahuje požadované proměnné a ten přetáhneme do části HMI tagy. V sekci HMI tagy jsou všechny proměnné PLC, ke kterým má HMI panel přístup. Tyto PLC proměnné jsou konvertovány na HMI tagy, s kterými může pak uživatel nadále v projektu pracovat a přiřazovat je k jednotlivým objektům. Zároveň lze nastavit například jak často dochází k čtení HMI tagu. V tomto projektu jsem nastavil všechny HMI tagy tak, že je jejich hodnota čtena cyklicky a to každých 100 ms, což je nejkratší možná doba.

Na Obr. 4-18 je zobrazena struktura vytvořených obrazovek k této diplomové práci.



Obr. 4-18 Použitá struktura obrazovek v HMI panelu.

Hlavní obrazovka je obrazovka, která se uživateli zobrazí po zapnutí stanoviště. Hlavní obrazovka obsahuje stručný popis, k čemu je určeno laboratorní stanoviště a 3 tlačítka, která umožňují přejít k ovládání stanoviště jedním ze tří dříve zmíněných způsobů.



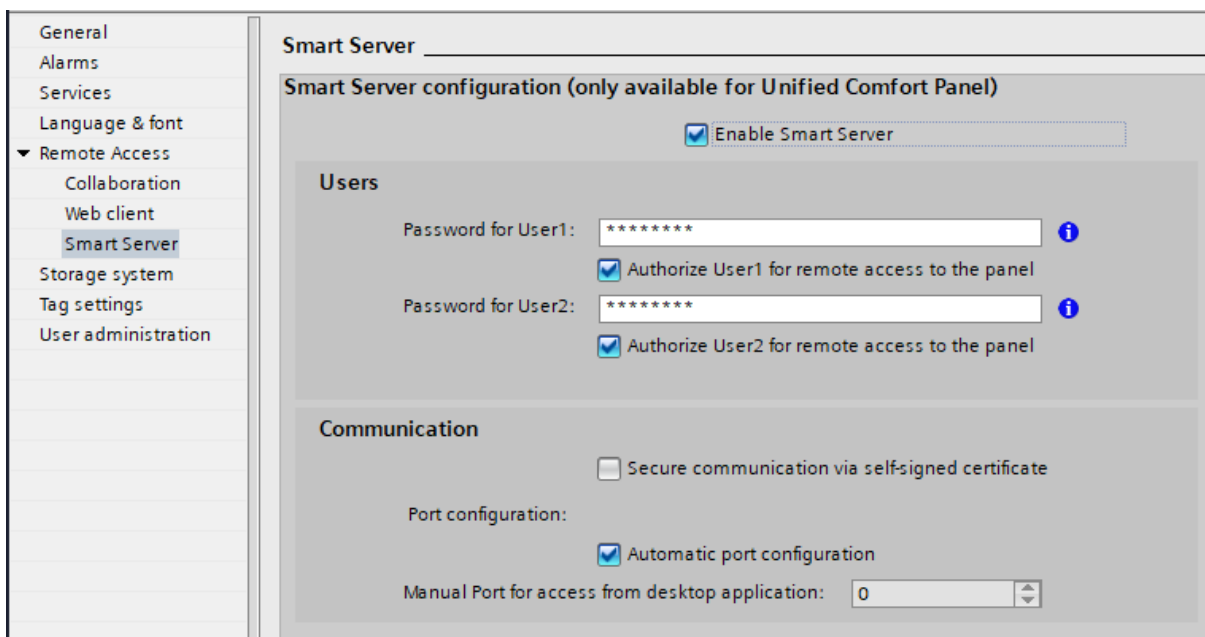
*Obr. 4-19 Hlavní obrazovka HMI panelu.*

Ostatní obrazovky jsou popsány v následující kapitole, protože přímo souvisí se zvolenou logikou řízení motorů.

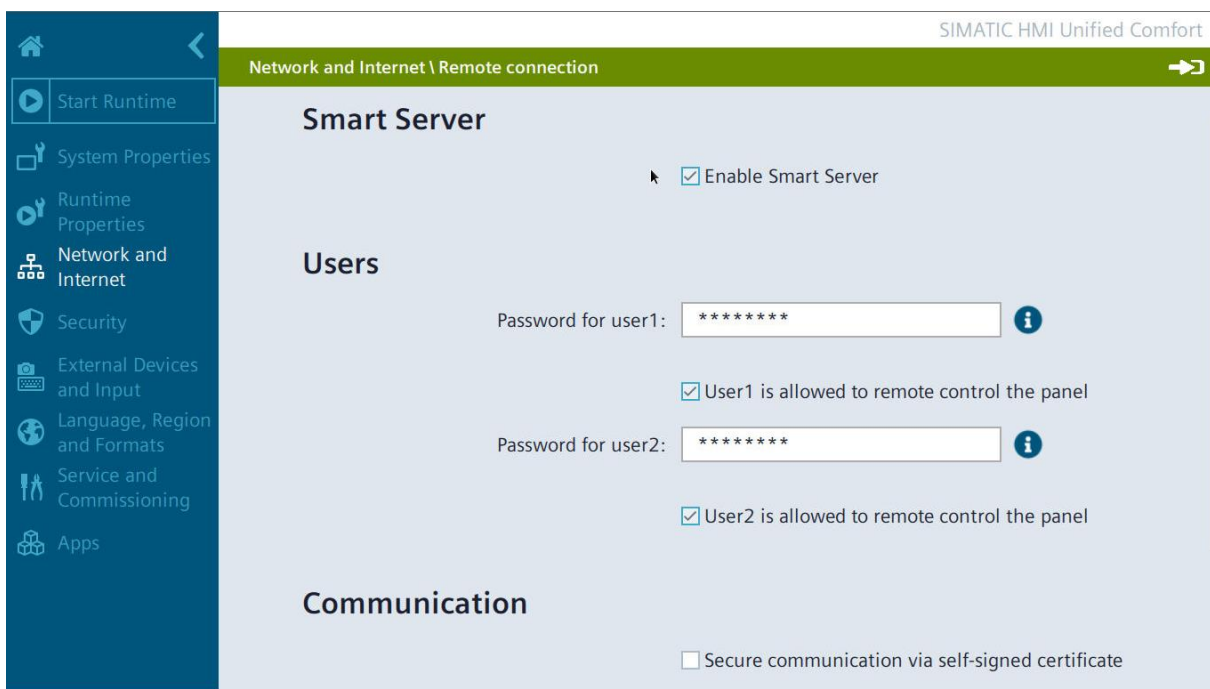
## 4.8 Smartclient

V této kapitole je popsáno, jak správně nakonfigurovat panel, aby bylo možné se k němu vzdáleně připojit s pomocí aplikace Smartclient. Nejprve je potřeba při vytváření projektu v TIA portálu ve vlastnostech panelu povolit Smartserver. Poté jsou nastaveny hesla pro uživatele 1 a 2, a je zaškrtnuto povolení uživatelům ke vzdálenému přístupu k panelu. Heslo pro uživatele 1 bylo zvoleno „Siemen1! a pro uživatele 2 „Siemen2!“. Toto je zobrazeno na Obr. 4-20. Následně je tato konfigurace nahrána do HMI panelu. Je potřeba se ujistit, že je tato konfigurace opravdu úspěšně nahrána v panelu, proto v nastavení panelu v části „remote control“ zkontrolujeme, že je vše správně nastaveno. To je zobrazeno na Obr. 4-21 a Obr. 4-22. Následně je možné se k panelu připojit, pokud jsme ve stejné lokální síti připojeni se zařízením, které umožňuje přístup k panelu jako Smartclient. Takové zařízení může být například počítač nebo notebook, který obsahuje aplikaci „Sm@rtclient“, která umožní nové spojení k Smartserveru. Je potřeba zadat IP adresu HMI panelu, ke kterému si přejeme připojit a zadat heslo pro uživatele 1 nebo heslo pro uživatele 2. V tu chvíli dojde ke připojení k HMI panelu a lze vzdáleně ovládat HMI panel. Obrazovka HMI panelu v aplikaci je duplikátem fyzické obrazovky stanoviště, proto cokoliv zvolíme přes Smartclient aplikaci je zvoleno i fyzicky, osoba fyzicky přítomná u HMI panelu tedy vidí, co se

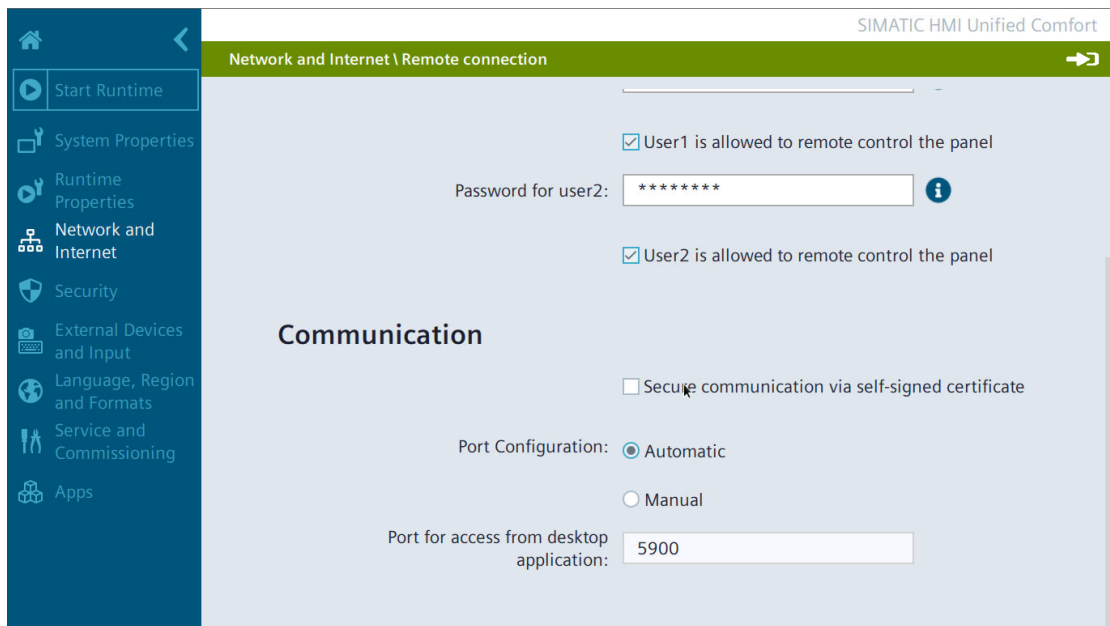
zrovna děje a pokud zmáčkne nějaké tlačítko, tak tato změna se přenese zpět i uživateli připojenému přes „Sm@rtclient“ aplikaci. Na Obr. 4-23 je vidět spuštěná aplikace „Sm@rtclient“ před připojením k panelu a na Obr. 4-24 je vidět spuštěná aplikace po připojení.



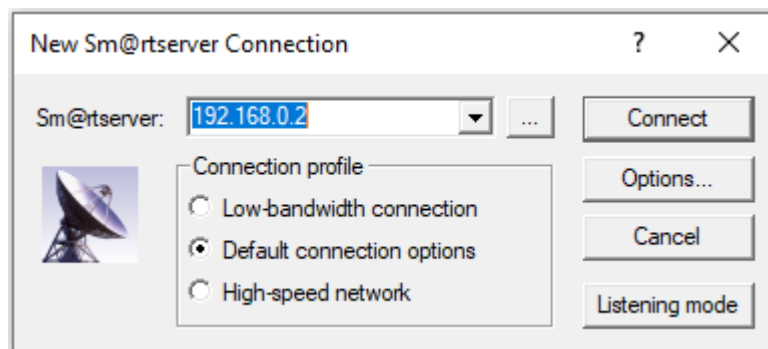
Obr. 4-20 Povolení Smartserveru v HMI panelu v TIA portál projektu.



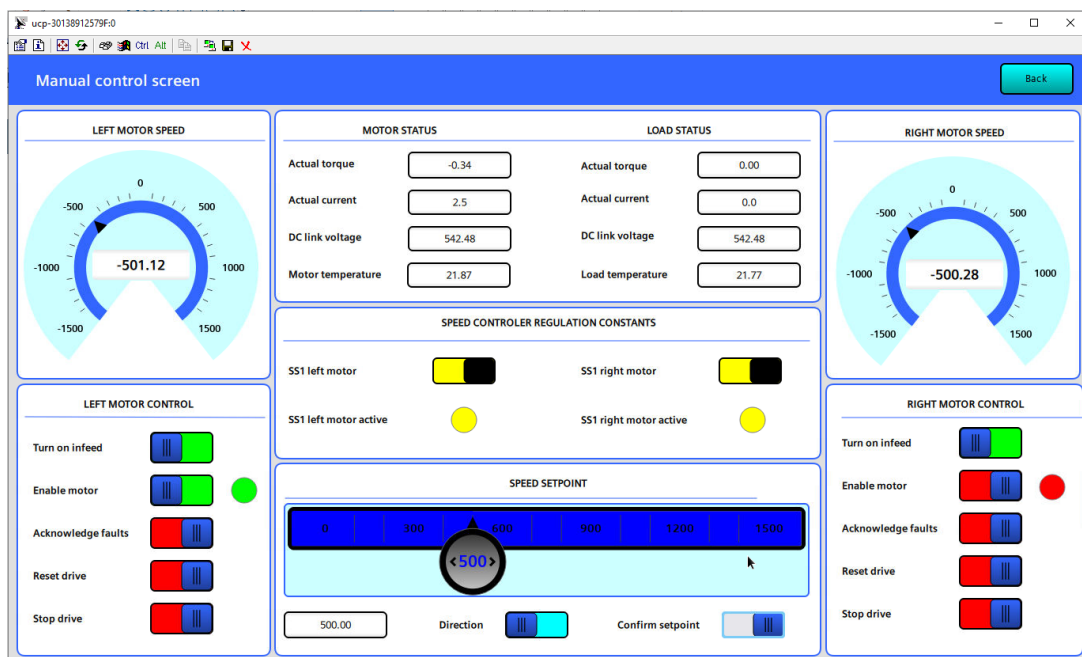
Obr. 4-21 Nastavení "Remote connection" v HMI panelu část 1.



Obr. 4-22 Nastavení "Remote connection" v HMI panelu část 2.



Obr. 4-23 Sm@rtclient aplikace před připojením k panelu.



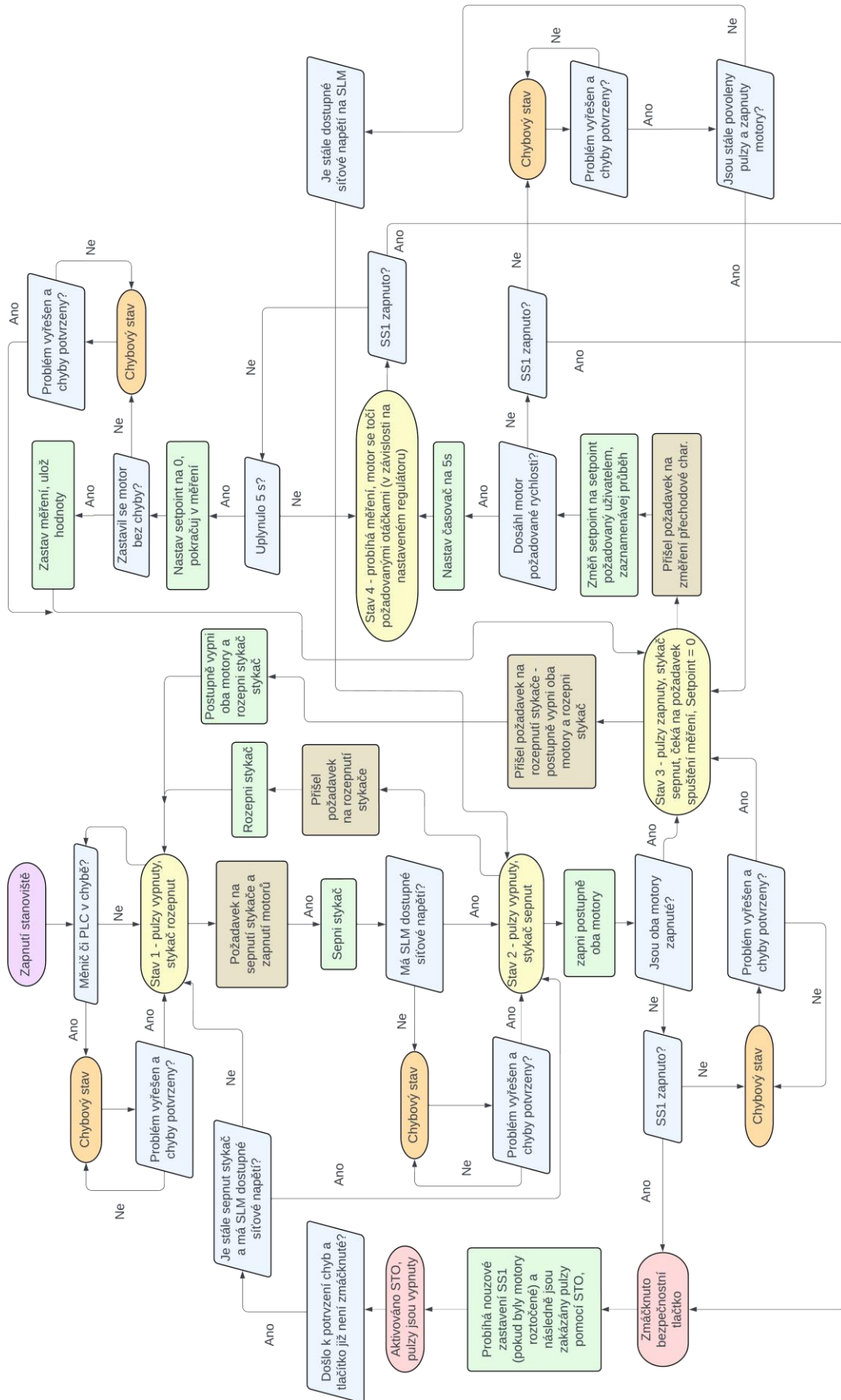
Obr. 4-24 Sm@rtclient aplikace po připojení k panelu.

## KAPITOLA 5: LOGIKA ŘÍDICÍHO PROGRAMU

Řídicí program lze rozdělit na 3 vývojové diagramy. To proto, že stanoviště obsahuje 3 možnosti, jak motory ovládat. První možnost je ovládat vždy jeden motor, mezitím co druhý motor je vypnutý. Druhá možnost je manuálně ovládat motory současně, přičemž jeden motor zastává úlohu zátěže, kterou můžeme nastavit dle zvolené charakteristiky a druhý úlohu motoru, u kterého můžeme měnit jeho otáčkový regulátor. Třetí možnost je stejné konfigurace, ale lze provést automatickou sekvenci k změření přechodové charakteristiky motoru pro zvolenou zátěž, zvolené konstanty otáčkového regulátoru a požadavek roztočení motoru na zvolenou rychlost. Po dokončení automatické sekvence má uživatel možnost uložit naměřenou přechodovou charakteristiku na USB Flash disk připojený k HMI panelu.

### 5.1 Automatická sekvence měření přechodové charakteristiky

Na Obr. 5-1 lze vidět vývojový diagram pro automatickou sekvenci změření přechodové charakteristiky. Při zapnutí stanoviště, pokud nehlásí měnič nebo PLC chybu, pak je měnič ve stavu 1, kdy jsou vypnuté pulzy měniče, osy jsou vypnuté a měnič nemá přístup k síťovému napětí, protože je rozepnutý stykač. Pokud uživatel požádá skrz HMI panel o sepnutí stykače a zapnutí obou motorů (obrazovka k ovládání automatické sekvence je na obr. Obr. 5-2), pak řídicí program automaticky zažádá o sepnutí stykače. Pokud při sepnutí stykače nedojde k chybě, pak měnič je připojen k síťovému napětí a měnič je ve stavu 2, kdy pulzy měniče jsou vypnuté, obě osy jsou vypnuté, ale stykač je sepnut. Následně program automaticky posílá požadavek na zapnutí motoru reprezentujícího zátěž a po úspěšném zapnutí zátěže i druhý motor. Pokud se nepovede oba motory zapnout, znamená to buď, že je stisknuté nouzové tlačítko a pulzy jsou potlačeny funkcí SS1 (technicky vzato STO, protože motory se netočí) anebo že došlo k chybě a chybu je potřeba vyřešit. Po vyřešení chyby a zmáčknutí tlačítka na potvrzení chyb je motor ve stavu 3, kdy jsou oba motory zapnuté, motor má *speed setpoint* (žádanou hodnotu otáček) 0 a program čeká na požadavek od uživatele k změření přechodové charakteristiky. V tomto stavu je vhodné skrz uživatelské rozhraní změnit konstanty otáčkového regulátoru a nastavit požadovanou charakteristiku zátěže pro motor reprezentující zátěž. Následně stačí zadat požadavek rychlosti, na kterou se má v rámci měření přechodové charakteristiky motor rozjet a zadat požadavek na začátek měření. V tu chvíli zadá program požadovaný *setpoint* měniči a motor se začne postupně otáčet, zatímco je zatěžován zátěžným momentem druhého motoru, až do doby, kdy dosáhne žádaných otáček. V tu chvíli program nastaví 5 sekund pro vnitřní časovač a nechá motor otáčet se dále stejnou rychlostí, což je stav 4. Po uplynutí 5 s dá požadavek na nulové otáčky a čeká, až se motor zastaví, čímž je motor zpátky ve stavu 3.



Obr. 5-1 Logika automatické sekvence měření přechodových charakteristik.

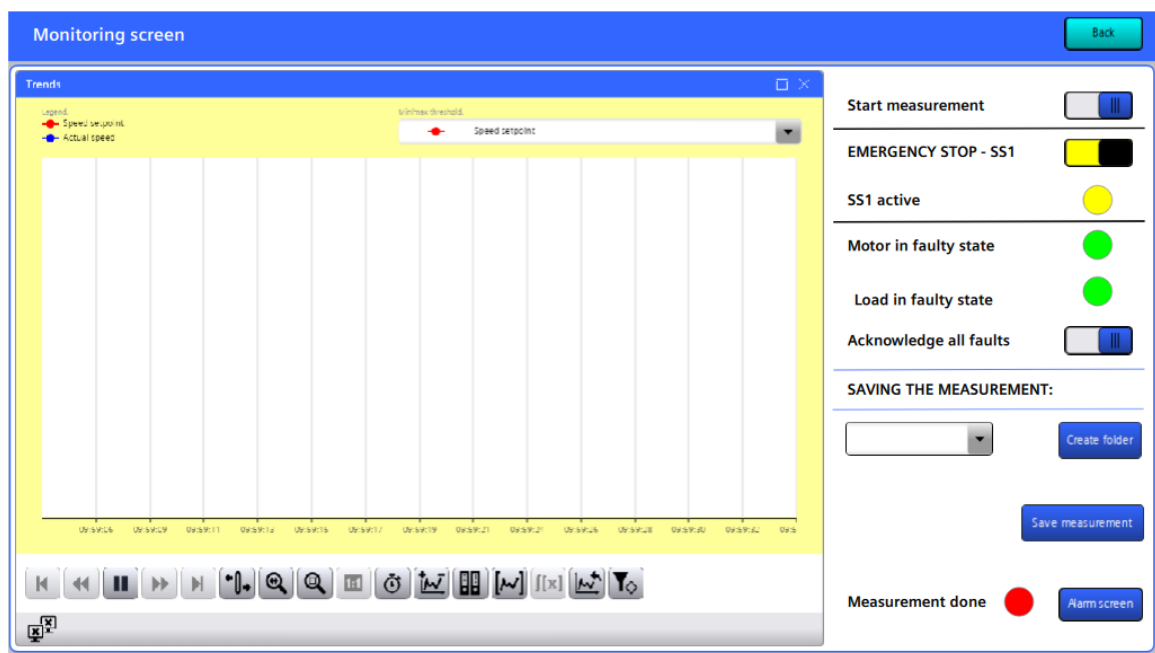


Během celé měřicí sekvence může uživatel zmáčknout nouzové tlačítko (ať už fyzicky, nebo na HMI panelu), které okamžitě zastaví motory a měření. Jakým způsobem je realizováno zaznamenávání přechodové charakteristiky je popsáno později v kapitole 5.4. Do původního stavu 1 ze stavu 3 se dostane uživatel, pokud vydá požadavek k rozepnutí stykače. Tím se spustí automatická sekvence k vypnutí obou motorů a rozepnutí stykače a zároveň se je ukončena automatická sekvence měření přechodové charakteristiky.

Obrazovka, kde uživatel nastaví parametry k automatickému měření přechodové charakteristiky je na Obr. 5-2. Nejprve je potřeba zapnout stykač a napětí, následně změnit parametry otáčkového regulátoru a nastavit požadovanou charakteristiku zátěže. Poslední krok je nastavení požadovaného skoku rychlosti. Nahoře je anglický popis ke správnému nastavení automatické sekvence. Zmáčknutím tlačítka „Measurement screen“ uživatel přejde do následující obrazovky, kde může začít měření.

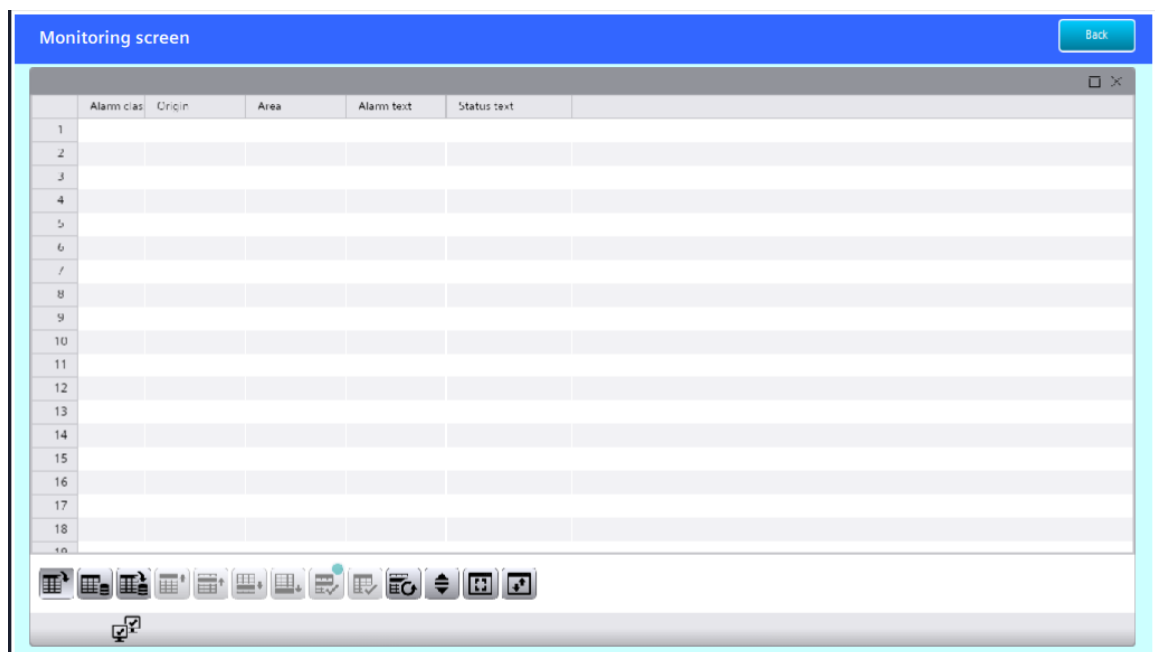
Obr. 5-2 Obrazovka k nastavení automatického měření přechodové charakteristiky.

Obrazovka k monitorování průběhu automatického měření a následnému uložení je na Obr. 5-3. V levé části obrazovky je rozhraní zvané „trend control“, které umožňuje zobrazovat průběh žádaných a aktuálních otáček. V pravé části obrazovky je tlačítko k spuštění měření, tlačítko k nouzovému zastavení, tlačítko potvrzení chyb, tlačítko k vytvoření složky, do které má být uložen naměřený průběh, tlačítko k uložení průběhu a tlačítko, které přepne obrazovku na obrazovku s diagnostickým oknem. Také zde jsou signalizační prvky, které značí, zda je zapnuté bezpečnostní zapnutí, osa motoru či zátěže v chybovém stavu anebo zda již skončilo měření hodnot. Jak probíhá měření hodnot je popsáno v kapitole 5.4.



Obr. 5-3 Obrazovka k monitorování automatického měření a následnému uložení naměřených

Obrazovka s diagnostikou obsahuje rozhraní nazvané „System diagnostics control“, kde jsou zobrazeny veškeré chyby a alarmy HMI panelu. V tomto diagnostickém okně mohou uživatelé sledovat, zda došlo k úspěšnému vytvoření složky a také souboru s naměřenými hodnotami. Tato funkce není automatická, ale je součástí scriptu, který řídí ukládání naměřených hodnot. O tom více v kapitole 5.4.



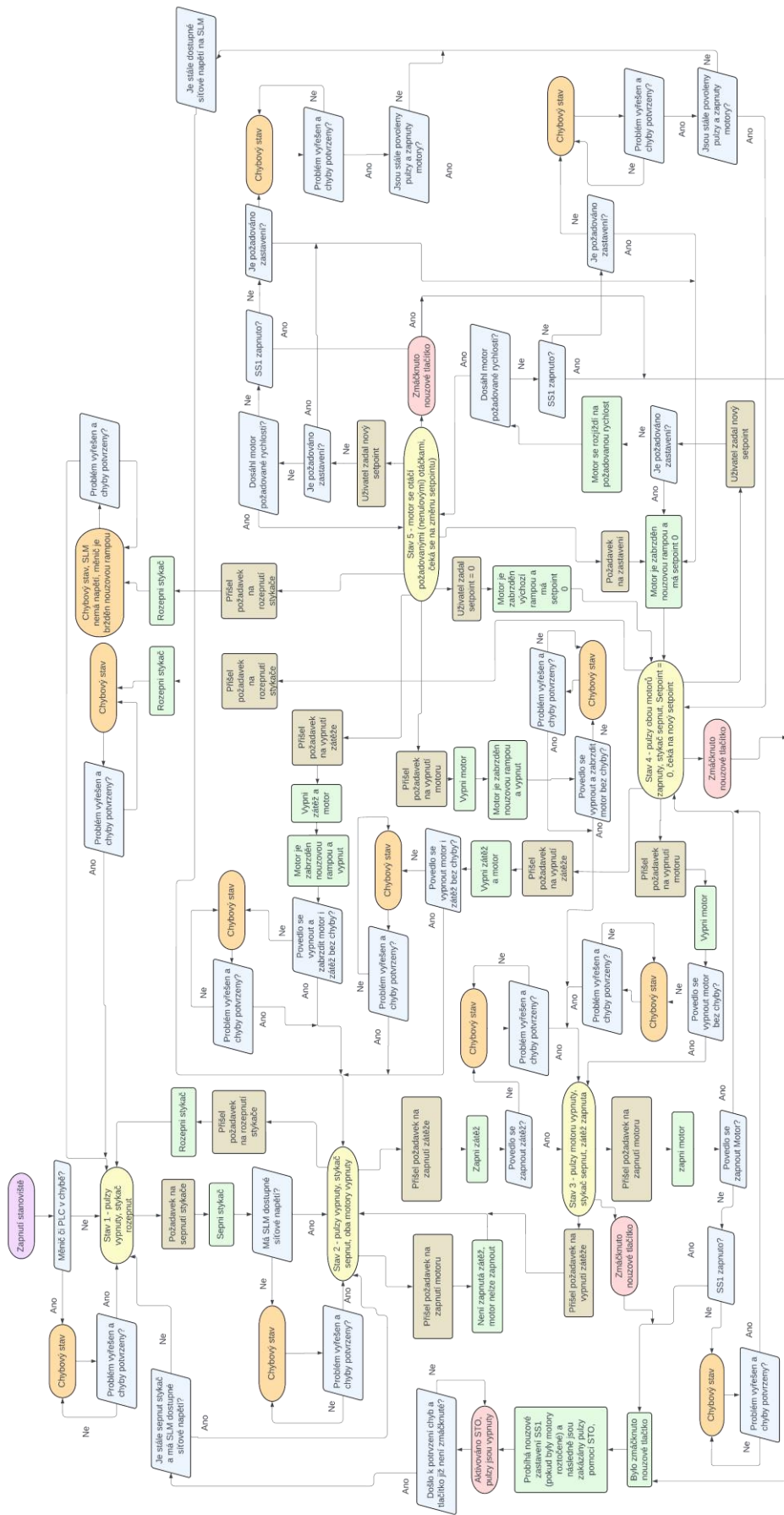
Obr. 5-4 Obrazovka s diagnostikou hodnot.

## 5.2 Manuální ovládání motorů v režimu motor-zátěž

Druhý způsob řízení, kdy jsou řízeny oba motory současně, avšak podle manuálních požadavků uživatele z HMI panelu je zobrazen na obrázku Obr. 5-6. Stejně jako u předchozí logiky řízení obou motorů je postup po zapnutí stanoviště, přes zapnutí stykače po zapnutí obou motorů technicky vzato stejná. Je však rozšířena o bezpečnostní podmínky, jelikož zde jsou motory řízeny manuálně, proto je potřeba zajistit, že uživatel zadává požadavky ve správném pořadí. Příkladem toho je přechod ze stavu 2, kdy je sepnut stykač a SLM je tedy připojeno k síťovému napětí, ale ani jeden motor ještě není sepnut do stavu 4, kdy jsou zapnuty oba motory. U automatické sekvence se postaral vnitřní program o to, že je nejdříve zapnuta zátěž a až poté i druhý motor. Zde je potřeba zakázat v programu zapnutí motoru ve stavu 2. Až když uživatel zapne pomocí tlačítka z HMI panelu motor reprezentující zátěž, tak řídicí program přejde do stavu 3 kdy již lze zadat žádost o zapnutí druhého motoru, který se následně zapne (nedojde-li k poruchovému stavu).

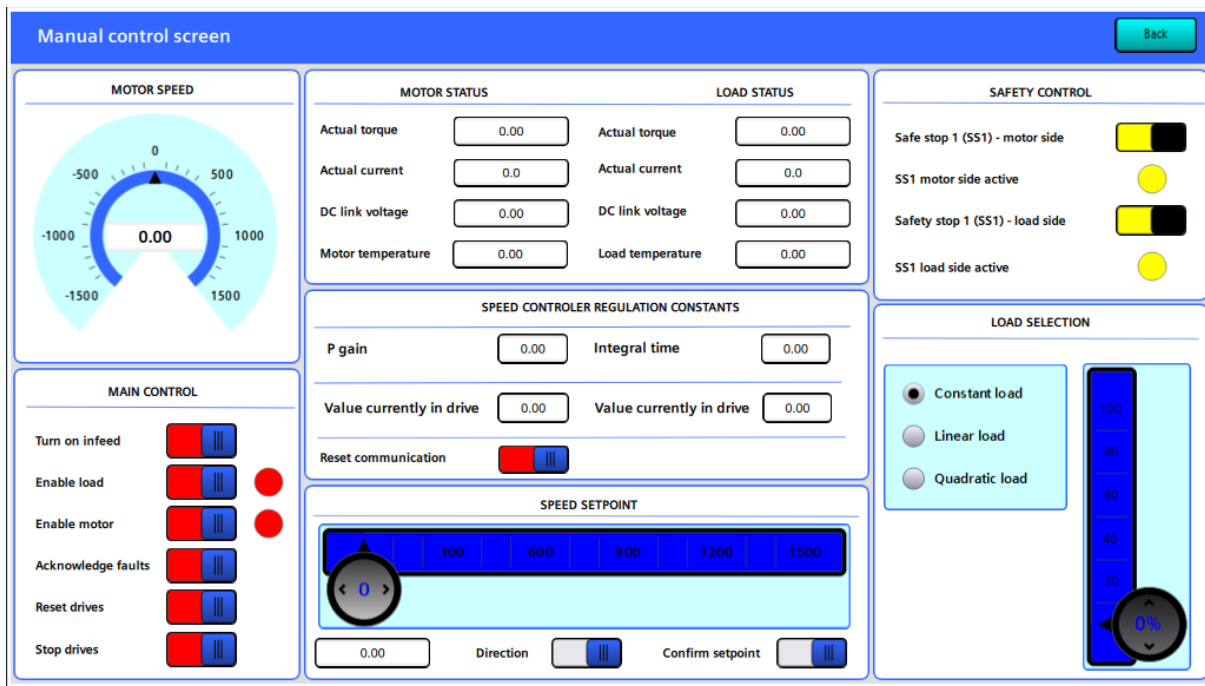
Stejně tak je potřeba očekávat, že pokud jsou zapnuty oba motory (stav 4) a uživatel dá příkaz k rozepnutí stykače, pak stykač bude rozepnut, ale měnič spadne do chyby a motory se vypnou, protože SLM nebude připojeno k napájecímu napětí ze sítě. Ze stavu 4 do stavu 5, kdy se motor otáčí požadovanými nenulovými otáčkami se uživatel dostane tak, že si zvolí požadované otáčky a potvrdí je pomocí tlačítka, tím se pošle z měniče nový *setpoint* měniči a měnič začne reagovat na změnu požadovaných otáček. U manuálního řízení lze měnit parametry otáčkového regulátoru a parametry zátěže kdykoliv, pomocí rozhraní vytvořeného na obrazovce HMI panelu. Stejně tak lze kdykoliv zadat požadavek na zastavení motorů, přičemž jsou dva způsoby, buď pomocí stop tlačítka na HMI panelu, které využívá funkce MC\_Stop a zastaví motor pomocí nadefinované rampy, nebo pomocí zmáčknutí nouzového bezpečnostního tlačítka, které je dostupné buď fyzicky na rámu panelu, nebo softwarově na obrazovce HMI panelu. Pokud jsou měniče zastaveny pomocí nouzového zastavení, je využita bezpečnostní funkce měniče SS1 a pulzy motorů jsou po zastavení vypnuty a motory nelze zapnout, dokud není nouzové tlačítko znovu ve stavu „nezmáčknuto“ a dokud nebyl vyslán požadavek na potvrzení chyb. V případě, že měnič nebo PLC je v chybovém stavu, lze chybu potvrdit tlačítkem na panelu v případě, že se nejedná o chybu, která takto vyřešit nelze (například pokud PLC ztratí spojení s měničem).

K realizaci manuálního ovládání motorů je určena obrazovka zobrazena na Obr. 5-6. V levé dolní části se vyskytují tlačítka k zapnutí stykače a obou motorů, potvrzení chyb, zastavení motorů a restartu MC\_bloků měniče, pokud je to potřeba. V horní prostřední části jsou zobrazeny aktuální hodnoty momentu, proudu a teploty obou motorových modulů a napětí na DC linku. V pravé horní části je možnost aktivovat bezpečnostní zastavení, ta je zde především pro vzdálenou funkci, uživatel v laboratoři může jednoduše zmáčknout fyzické bezpečnostní tlačítko přidělané k rámu HMI panelu.



Obr. 5-5 Logika manuálního řízení motorů, kdy jeden reprezentuje zátěž a druhý motor.

Pravá spodní část umožňuje nastavit požadované parametry zátěže. Uprostřed jsou zobrazeny konstanty otáčkového regulátoru a také jsou zde pole, které je umožňují měnit, včetně tlačítka k restartu bloku pro acyklickou komunikaci, pokud by došlo k chybě komunikace. Zátěž i konstanty regulátoru lze měnit kdykoliv i během toho, co jsou motory roztočeny. V dolní prostřední části se nastavuje požadovaná rychlost otáček a směr otáčení.

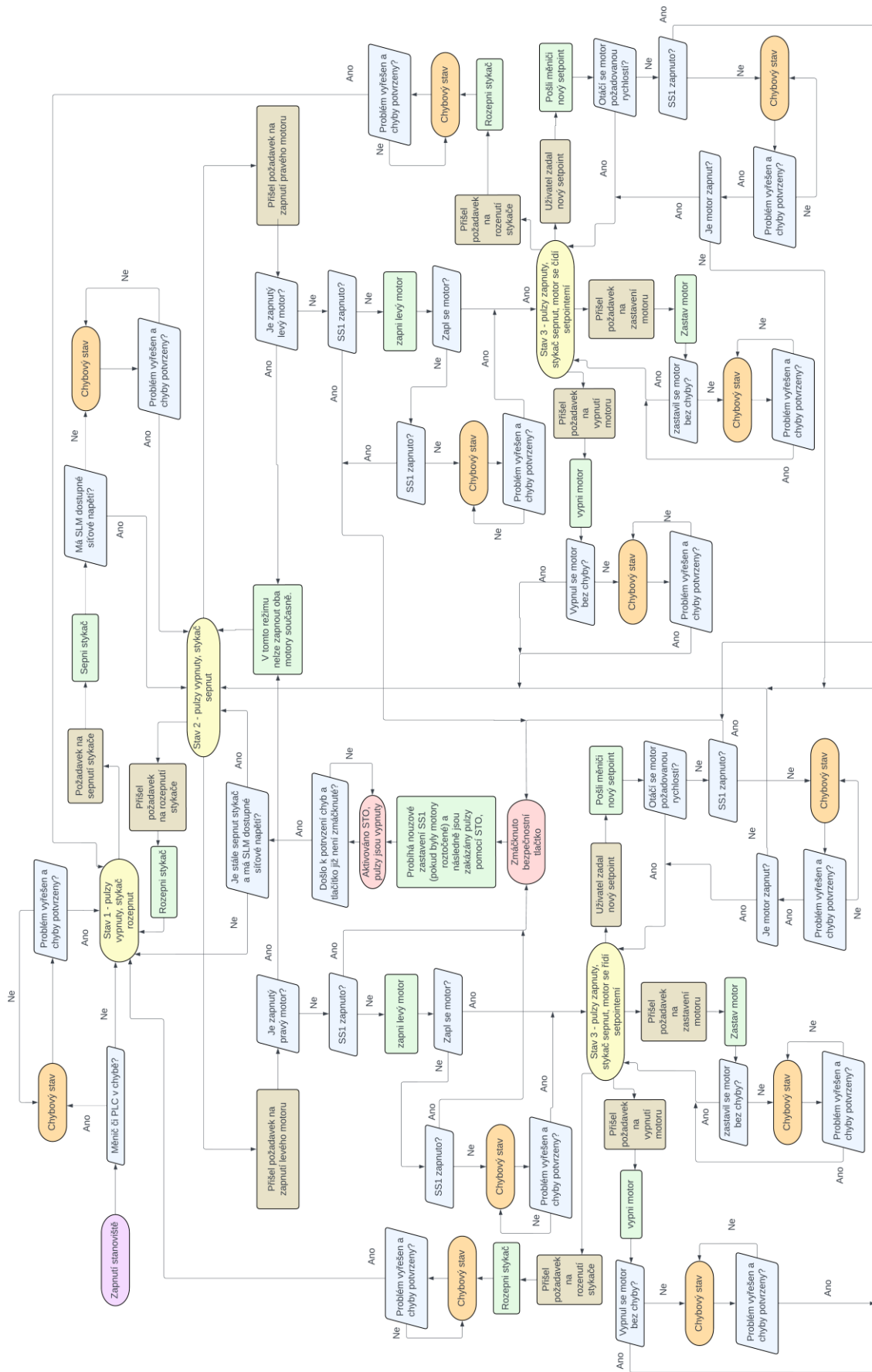


Obr. 5-6 Obrazovka k manuálnímu ovládní měření.

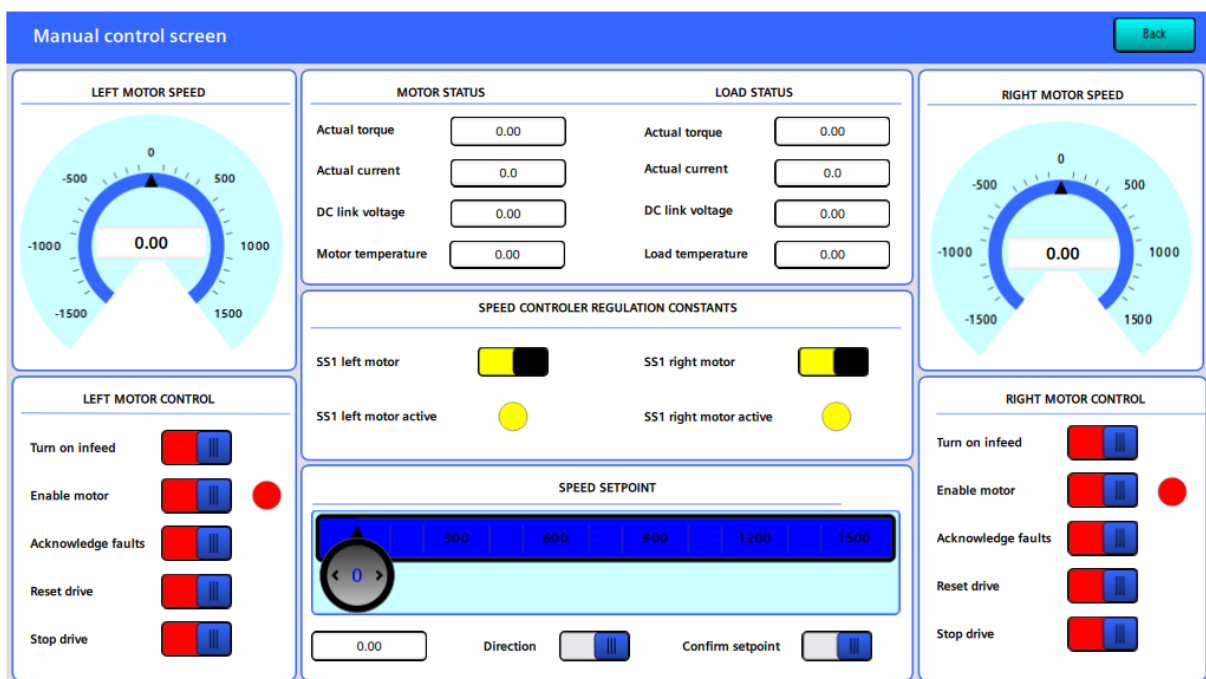
### 5.3 Manuální ovládní motorů samostatně

Poslední způsob řízení je manuální řízení motorů samostatně. Struktura řízení je v principu stejná, jako u předchozího způsobu, avšak s tím rozdílem, že pokud je zapnutý levý motor, nemůže být zapnut motor pravý a naopak. Vývojový diagram popisující tento způsob řízení je na Obr. 5-7.

Obrazovka k manuálnímu ovládní motorů zvlášť je strukturou velmi podobná předchozí obrazovce. Jediný rozdíl je, že každý motor lze ovládat pouze samostatně, a proto má každý motor vlastní tlačítka k zapnutí stykače, motoru, potvrzení chyb, restartu MC\_bloků a zastavení motoru. Obrazovka je zobrazena na Obr. 5-8.



Obr. 5-7 Logika manuálního řízení motorů samostatně.



Obr. 5-8 Obrazovka k manuálnímu ovládní motorů zvlášť.

## 5.4 Popis programu k ukládání naměřených hodnot

V této kapitole je popsáno, jakým způsobem je realizované ukládání změřených hodnot automatického měření přechodové charakteristiky. Původně bylo plánováno využít rozhraní „trend control“, které je použito na obrazovce k monitorování měření. Toto rozhraní umožňuje nejen monitorovat průběh měření, ale pomocí tlačítka jednoduše uložit daný průběh do souboru s příponou .csv. Později se však ukázalo, že toto řešení není pro tuto aplikaci vhodné. Důvodem je, že rozhraní „trend control“ zobrazuje a následně ukládá pouze průběhy proměnných HMI tagů. Tyto tagy lze však aktualizovat s nejkratší délkou cyklu 100 ms, což je doba příliš dlouhá a výsledné průběhy nebyly dostatečně přesné. K zobrazování právě probíhaného měření bohužel nebylo nalezeno řešení, jak tento problém vyřešit s použitým hardwarem a softwarem, a proto jsou průběhy zobrazovány s frekvencí 100 ms. Existuje však způsob, jak ukládat hodnoty s kratším časovým intervalem, než 100 ms, které bylo úspěšně implementováno.

Uvnitř řídicího FB *program control*, který je zobrazen na Obr. 4-16 a na Obr. 4-17 a realizuje řídicí logiku automatických i manuálních způsobů ovládní, je i program, který ukládá aktuální a požadované hodnoty otáček motoru. Základem programu jsou tři pole o velikosti 250, přičemž 2 pole jsou pole datových typů *Real* a jedno pole je pole datových typů *Time*. Tyto pole slouží jako zásobník hodnot potřebných k vytvoření přechodové charakteristiky, tedy k ukládání času měření a hodnot aktuálních a požadovaných otáček v daném čase. Cílem je uložit každých 40 ms tyto údaje. K tomu je využita funkce `Time_tck()`, jejíž výstupem je aktuální systémový čas CPU. Díky tomu lze odměřit 40 ms od doby, kdy byly naposledy uloženy hodnoty do zásobníku a v následujícím cyklu PLC programu, což je nejpozději za 4 ms dojde k uložení dalších hodnot.

Měření je koncipováno tak, že je neustále ukládáno 10 hodnot do takzvaného „předzásobníku“, aby byly tyto hodnoty dostupné v jakémkoliv čase. To proto, že program není schopný určit, v jakou dobu uživatel zadá požadavek na počátek měření. Jelikož není žádané, aby byl požadovaný skok přechodové charakteristiky na zobrazeném průběhu v čase 0 s, tak byl vytvořen tento „předzásobník“. Ve chvíli, kdy program dostane požadavek na spuštění automatické sekvence k změření přechodové charakteristiky s požadovanými parametry, uloží se hodnoty z „předzásobníku“ do prvních deseti indexů polí zásobníku, díky čemuž není požadovaný skok v čase 0, ale okolo 400. ms od počátku měření. Poté, co motor dosáhne požadované rychlosti, je řídicím programem odpočítáno 5 s (jak již bylo popsáno v kapitole Kapitola 5:) a je nastavena požadovaná rychlost 0 otáček za sekundu a dojde k zastavení motoru. Poté se čeká, až se zaplní všech 250 hodnot zásobníku, což dohromady dá čas 10 s (s drobnou odchylkou) a je vyslána informace, že měření skončilo. Díky tomu jsou nyní v PLC uloženy hodnoty, jejichž pomocí lze zobrazit přechodovou charakteristiku s frekvencí měření 40 ms. Tak je dosaženo vyšší přesnosti měření než při použití rozhraní *trend control*. Hodnota 40 ms byla zvolena jako dostatečně přesná hodnota k změření těchto přechodových dějů. Bylo by možné zvolit i kratší dobu, ale poté by byla zbytečně zatěžováno CPU a přenos výsledných dat by trval delší dobu (především uložení dat z HMI na USB flash disk).

Následně je potřeba tyto data uložená v PLC proměnných předat HMI panelu a umožnit studentům je jednoduchým způsobem uložit do souboru s příponou .csv. Toto je možné díky tomu, že panel je z řady Unified comfort panelů, které umožňují tvorbu skriptů v programovacím jazyce JavaScript. Na obrazovce k monitorování a ukládání panelů jsou tedy vytvořeny dvě tlačítka, která těchto skriptů využívají. První tlačítko slouží k vytvoření složky, do které budou ukládány průběhy. Jméno složky závisí na aktuálním datu a na vybrané laboratorní skupině (je na výběr cvičení od hodin 7:00 ráno do 18:30 večer). Výběr laboratorní skupiny probíhá pomocí symbolického IO pole vlevo od tlačítka „*Create folder*“. Toto je vytvořeno pro přehlednost, aby studenti nemuseli složitě hledat naměřené průběhy v dalších desítkách naměřených průběhů od předchozích kolegů. Druhé tlačítko je určeno k vytvoření csv souboru s naměřenými hodnoty. Skript v tomto tlačítku nejprve připraví jméno výsledného csv souboru. Jméno souboru v sobě obsahuje datum, čas měření a konstanty regulátoru v průběhu měření. Toto je pro usnadnění práce studentům, aby se jim nespletly naměřené průběhy. Následně skript vytvoří csv soubor pomocí funkce `WriteFile()`, který obsahuje 3 sloupce dat. V prvním sloupci je čas měření, v druhém požadovaná hodnota otáček a v třetím aktuální hodnota otáček.

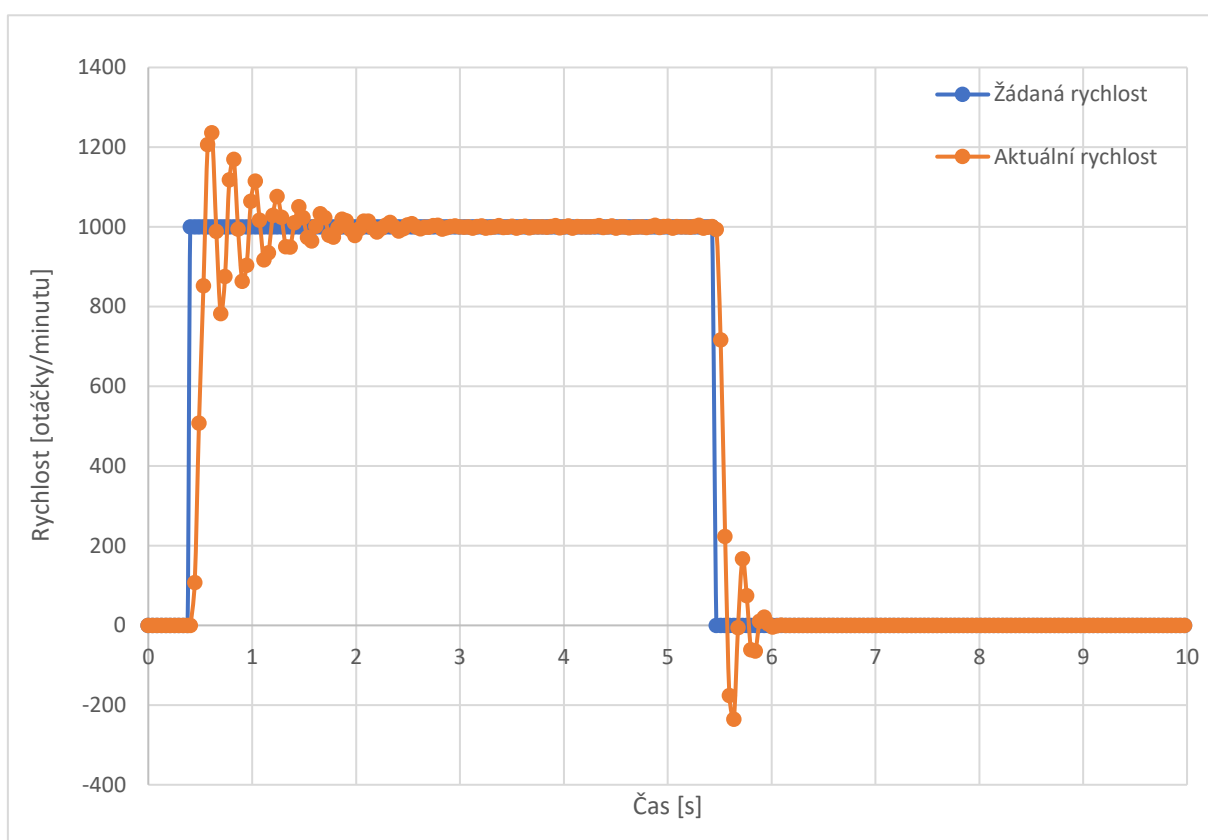
Pokud vše proběhlo bez chyby, pak se v diagnostice objeví dvě zprávy, první o tom, že se povedlo vytvořit složku a druhá o tom, že se povedlo uložit data. Pokud se nepovede vytvořit složku nebo csv soubor, objeví se v diagnostice informace o tom, že se uložení nepovedlo. To může být způsobeno například tím, že studenti nevrátí do správného USB portu USB flash disk a HMI



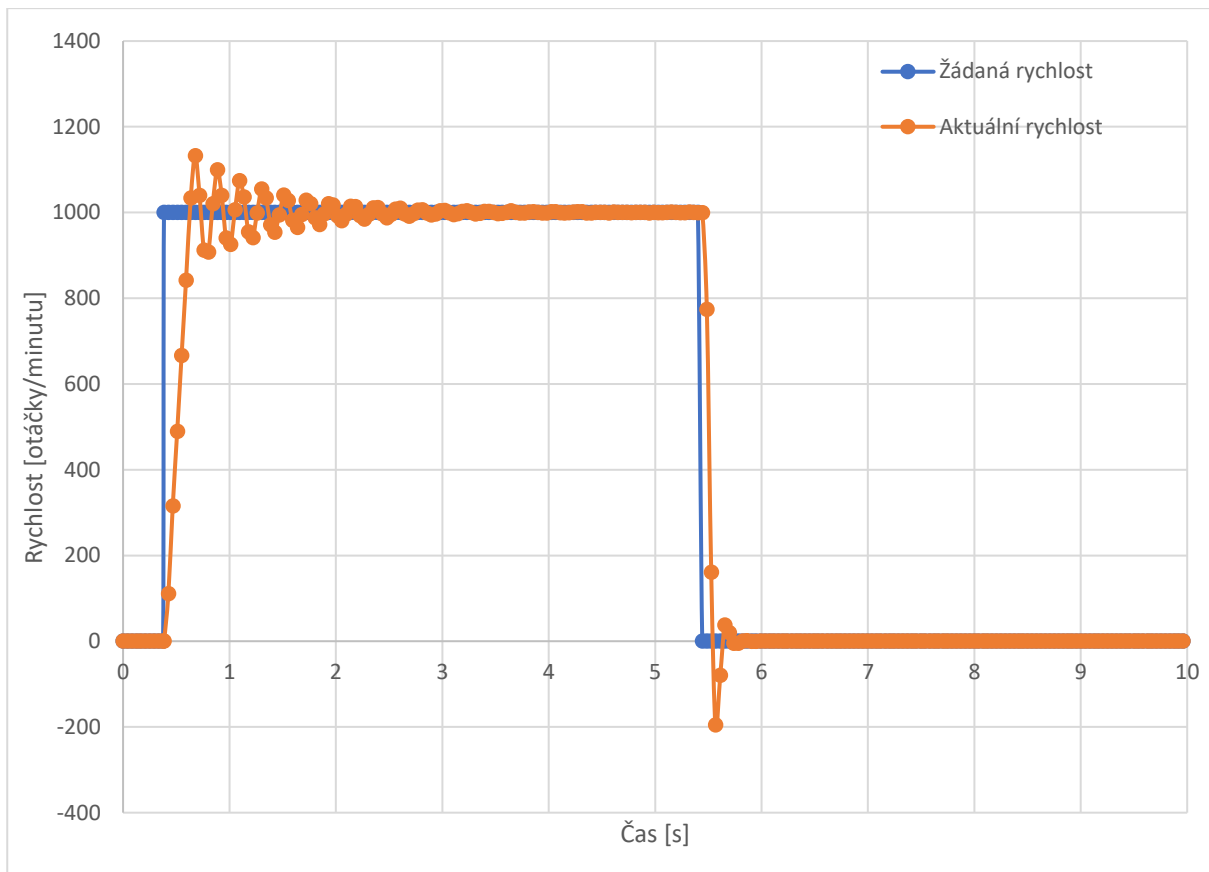
panel tak nemá kam vytvořit složku a soubor. Toto není standardní funkce, tyto diagnostické zprávy jsou součástí skriptů obou tlačítek.

## KAPITOLA 6: NAMĚŘENÉ PRŮBĚHY

V této kapitole jsou zobrazeny vzorové průběhy, které byly naměřeny pomocí automatické sekvence měření. Průběhy Obr. 6-1 a Obr. 6-2 zobrazují přechodové charakteristiky, které jsou sice stabilní, ale kmitavého charakteru. Kmitání je způsobeno špatně naladěným otáčkovým regulátorem. Takto naladěný regulátor není vhodný nejen proto, že překmit je 23 %, ale také proto, že kmitavým přechodovým dějům je obecně snaha se vyhýbat, a protože doba ustálení na požadované hodnotě je příliš dlouhá. Rozdíl mezi prvním a druhým průběhem je zátěžný moment, který způsobí, že kmity jsou tlumené a překmit nedosahuje u druhého průběhu 23 %, ale 13 %. Průběh Obr. 6-2 je zde pro ukázkou, že na charakteristice zátěžného momentu působícího proti momentu poháněného motoru z hlediska ladění regulátorů záleží.

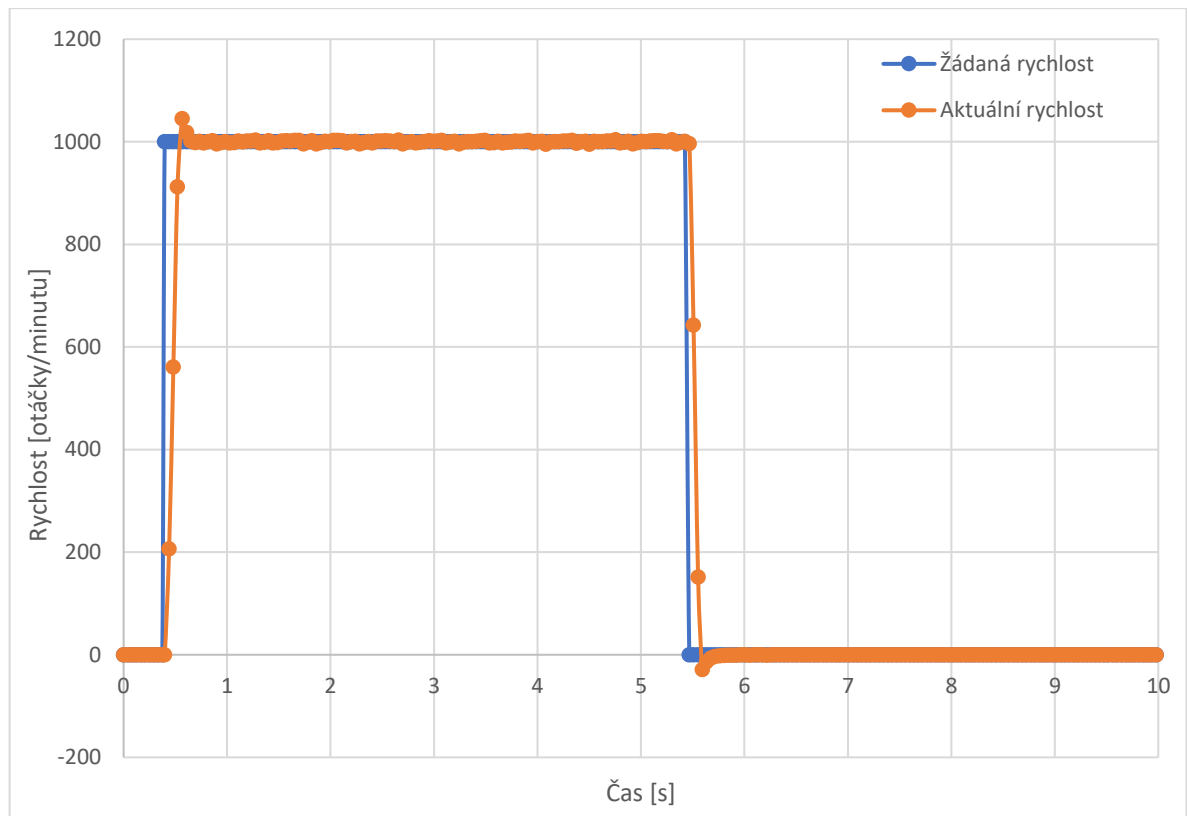


Obr. 6-1 Změřený průběh kmitavě stabilní přechodové charakteristiky při  $k_p=2$ ,  $T_I=10$  a  $M_z$  s konstantní charakteristikou, 10 % jmenovitého momentu motoru.

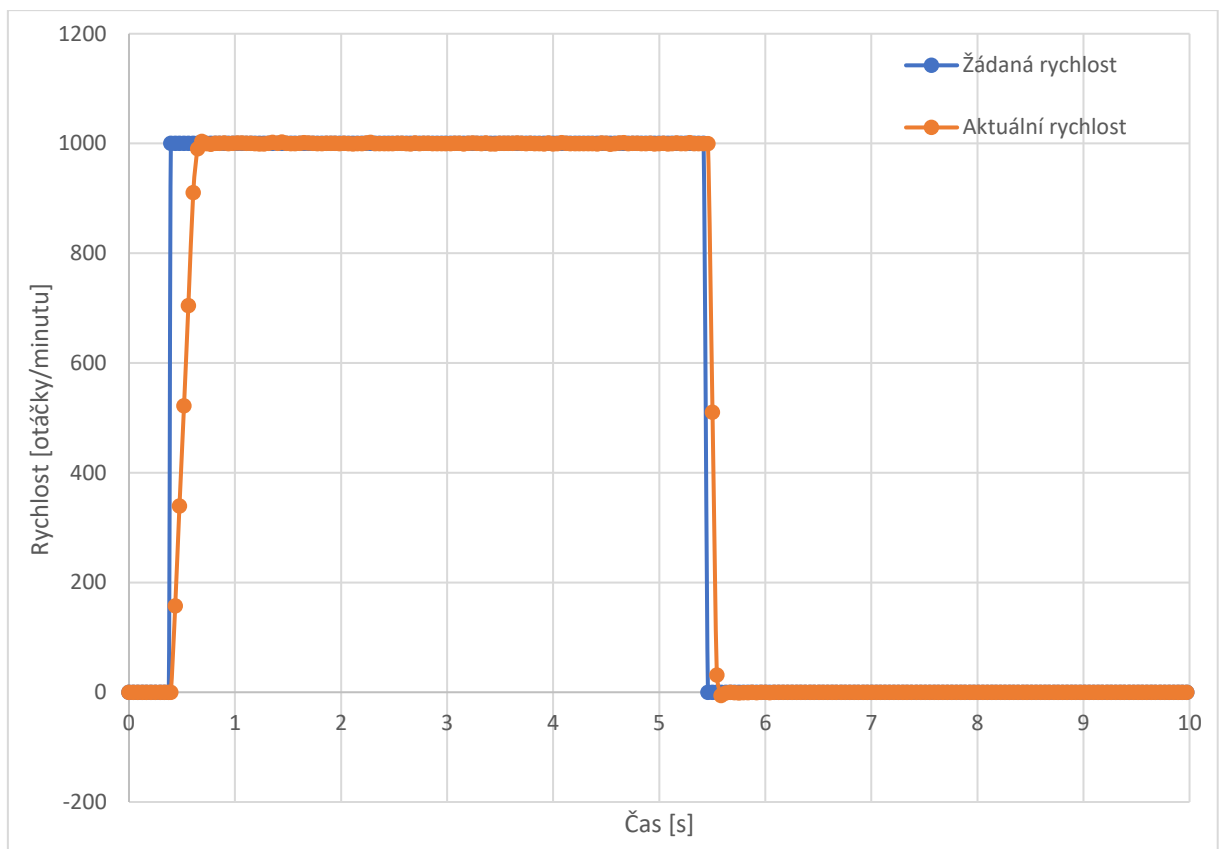


*Obr. 6-2 Změřený průběh kmitavě stabilní přechodové charakteristiky při  $k_p=2$ ,  $T_I=10$  a  $M_z$  s konstantní charakteristikou, 90 % jmenovitého momentu motoru.*

Změřené přechodové charakteristiky na Obr. 6-3 a Obr. 6-4 jsou stabilní a astatické. Toto jsou charakteristiky správně naladěného regulátoru a takto naladěný otáčkový regulátor lze použít v průmyslových aplikacích. U průběhu Obr. 6-3 se sice vyskytuje překmit, avšak v rozumných mezích a po velmi krátkou dobu. V průmyslových aplikacích jsou navíc motory rozjížděny po rampě a ne skokovým požadavkem, v takovém případě by se překmit ani neprojevil.

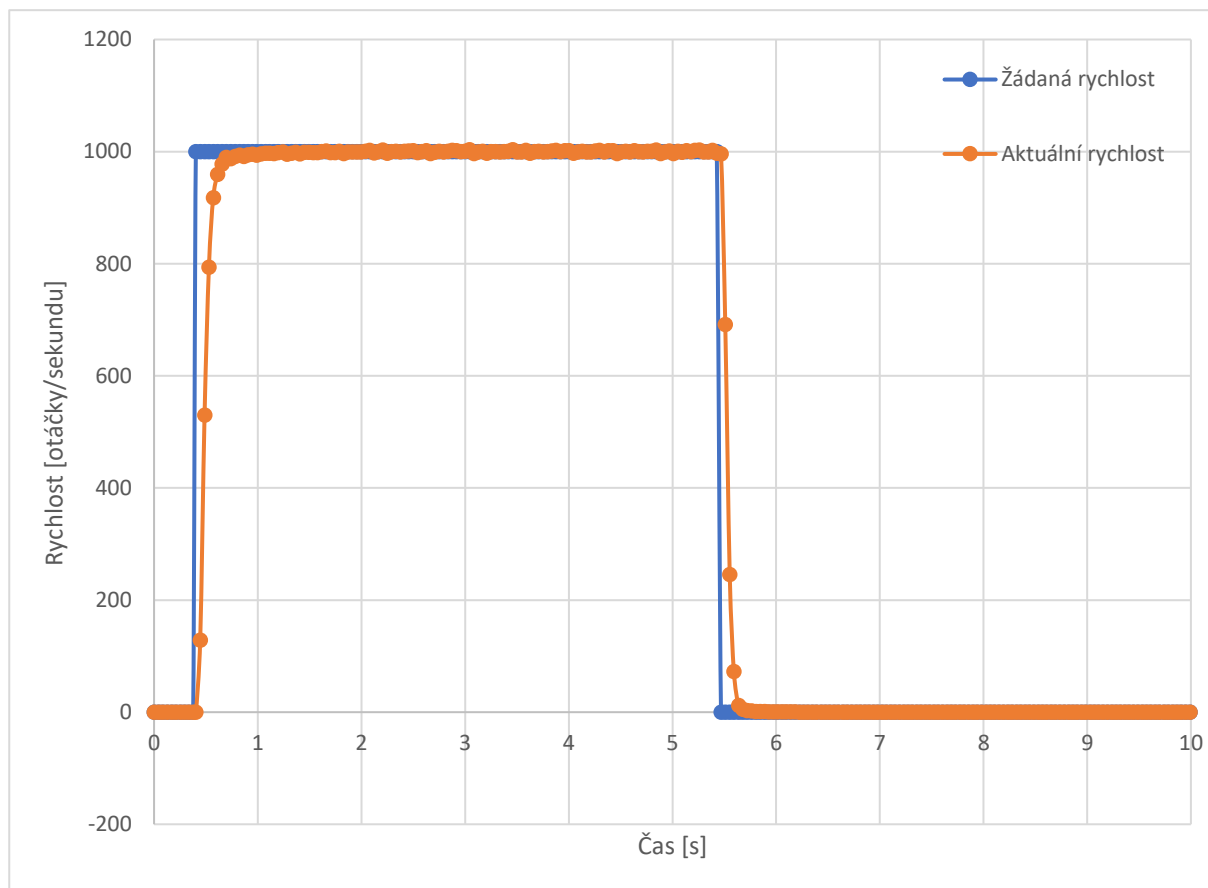


Obr. 6-3 Změřený průběh aperiodicky stabilní přechodové charakteristiky při  $k_p=10$ ,  $T_I=50$  a  $M_z$  s konstantní charakteristikou, 10 % jmenovitého momentu motoru.



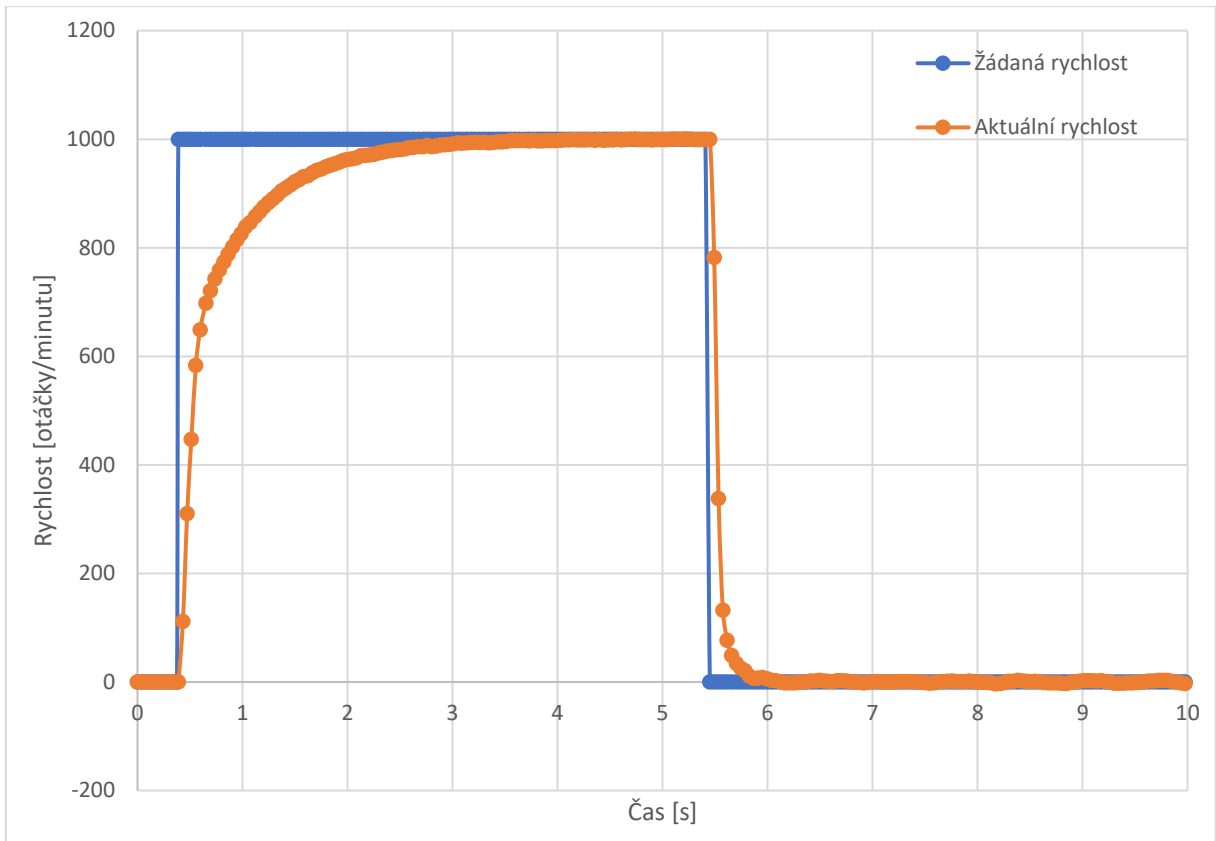
Obr. 6-4 Změřený průběh aperiodicky stabilní přechodové charakteristiky při  $k_p=10$ ,  $T_I=50$  a  $M_z$  s konstantní charakteristikou, 90 % jmenovitého momentu motoru.

Na Obr. 6-5 a Obr. 6-6 jsou změřené přechodové charakteristiky s také aperiodicky stabilní. První průběh je ještě docela použitelný, pokud by u použité aplikace šlo zaručit, že zátěžný moment nebude vyšší než aktuální zátěžný moment u této přechodové charakteristiky. V případě, že však zátěžný moment vzroste, pak vidíme, že doba náběhu je velmi dlouhá, což je nežádoucí. Toto je reprezentováno druhým průběhem.

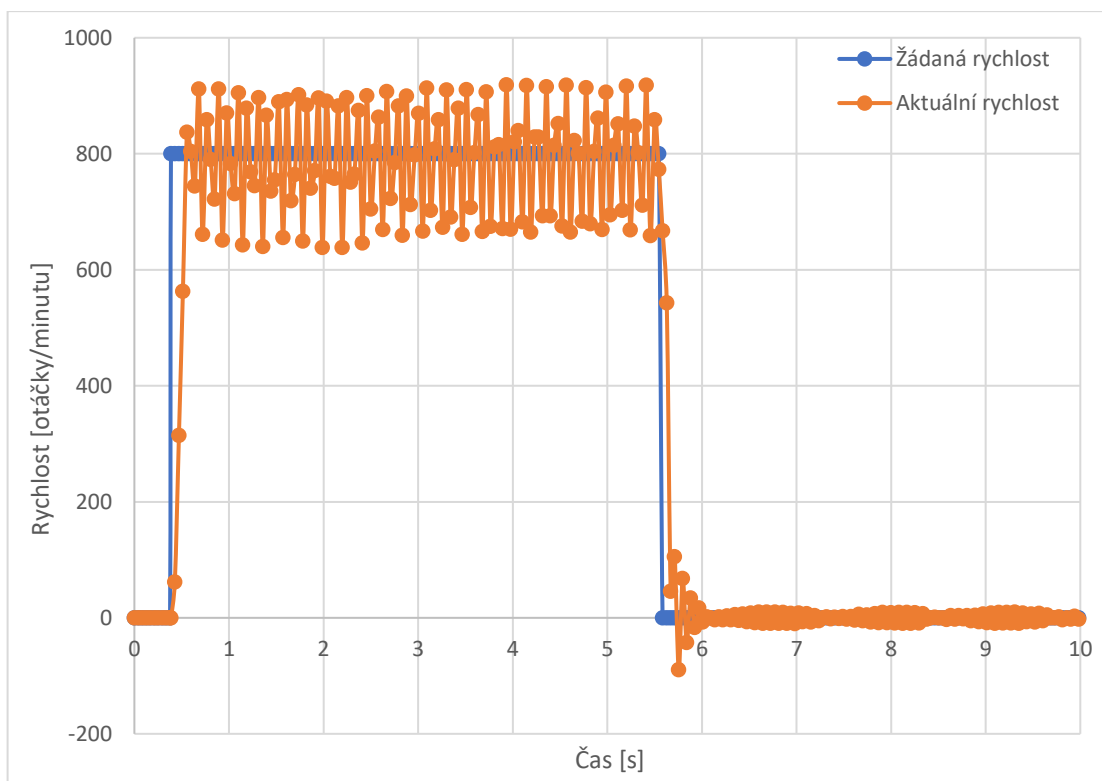


Obr. 6-5 Změřený průběh aperiodicky stabilní přechodové charakteristiky při  $k_p=5$ ,  $T_I=750$  a  $M_z$  s konstantní charakteristikou, 10 % jmenovitého momentu motoru.

Na posledním ukázaném typickém průběhu přechodové charakteristiky na Obr. 6-7 je zobrazen stav, kdy systém pracuje na hranici stability. To je typické tím, že aktuální hodnota otáček neustále kmitá a nikdy se neustálí na požadované hodnotě. Tento stav je nežádoucí.



Obr. 6-6 Změřený průběh aperiodicky stabilní přechodové charakteristiky s dlouhou dobou náběhu při  $k_p=5$ ,  $T_I=750$  a  $M_z$  s konstantní charakteristikou, 90 % jmenovitého momentu motoru.



Obr. 6-7 Změřený průběh kmitavé přechodové charakteristiky na hranici stability při  $k_p=50$ ,  $T_I=500$ ,  $M_z$  s konstantní charakteristikou, 10 % jmenovitého momentu motoru a požadavku na rychlost otáček 800 otáček/sekundu.

Bylo potřeba stanovit rozumné meze konstant otáčkového regulátoru, které může uživatel zadat s pomocí HMI panelu. Proto byly změřeny přechodové charakteristiky pro různé kombinace parametrů. Tabulka Tab. 6-1 popisuje charakter přechodové charakteristiky při daných konstantách otáčkového regulátoru při konstantní zátěži o velikosti 50 % jmenovité hodnoty momentu motoru a požadavku na 800 otáček za minutu. Zelená pole značí to, že přechodová charakteristika je stabilní, doba náběhu je krátká a přesah je v rozumných mezích, otáčkový regulátor je tedy správně naladěn. Žluté pole značí to, že přechodová charakteristika je stabilní, ale má buď dlouhou dobu náběhu, moc vysoký přesah anebo před ustálením dochází ke kmitům. To znamená, že otáčkový regulátor není vhodně naladěn. Červené pole značí to, že je přechodová charakteristika na mezi stability, dochází tedy k netlumeným kmitům a rychlost motoru se nikdy neustálí na požadovaných otáčkách, regulátor je špatně nastaven. Z výsledného měření bylo zvoleno rozmezí parametrů regulátoru  $k_p$  od 1 do 100 a  $T_i$  od 1 do 1000. To proto, že při nižších nebo vyšších hodnotách parametrů byly kmity a spřažených motorů příliš velké a při opakovaném provozování v tomto stavu by docházelo k opotřebení motorů. Zároveň je však žádané, aby rozmezí obsahovalo i nepříznivé chování, aby studenti mohli v praxi vidět, jak se motor chová, pokud jsou konstanty regulátoru nevhodně zvoleny. To je důvod, proč pro horní hodnotu parametru  $k_p$  nebylo zvoleno 25, ale 100.

		kp							
		1	2	5	10	25	50	75	100
Ti	1								
	2								
	5								
	10								
	25								
	50								
	75								
	100								
	200								
	300								
	500								
	750								
999									

Tab. 6-1 Tabulka popisující, jak dobře byl naladěn otáčkový regulátor.

## ZÁVĚR

Cílem této práce bylo vytvořit stanoviště, které umožní vzdáleně měřit charakteristiky pohonu, konkrétně přechodové charakteristiky otáčkového regulátoru asynchronního motoru. Tyto charakteristiky jsou vhodné k určení, zda jsou konstanty otáčkového regulátoru pohonu vhodně nastaveny. Stanoviště bylo zhotoveno pomocí zařízení od firmy Siemens, od motorů, přes měnič, PLC až po HMI panel. S výrobou stojanů, ke kterým je HW přidělán pomohli pracovníci katedry. Se zapojením stanoviště nebyly žádné problémy a výsledkem této práce je plně funkční stanoviště k měření přechodových charakteristik regulátoru otáček asynchronního motoru.

Stanoviště je předně určeno k automatické sekvenci měření přechodových charakteristik, je zde však možnost pohánět motory manuálně, a to jak v režimu motor-zátěž, tak v režimu ovládní motorů samostatně. Při realizaci měření charakteristik došlo nejprve ke komplikacím, jelikož původně implementovaná funkce starající se o záznam a zobrazování dat nebyla schopná dostatečně vysoké frekvence ukládání naměřených hodnot. Tyto problémy se povedlo vyřešit implementováním vlastního programu k zaznamenávání naměřených hodnot a následně vytvořením tlačítka se skriptem, který umožňuje tyto hodnoty uložit. Po domluvě s vedoucím práce byla vzdálená správa realizována pouze pomocí Smartserveru v dosahu lokální sítě. Vzdálená správa pomocí Sinema RC serveru realizována nebyla, protože stanoviště bude provozováno pouze v osobní formě. Vytvoření bezpečnostní zóny nebo monitorovacího systému, který by umožnil bezpečně pracoviště používat vzdáleně je mimo rámec diplomové práce.



## LITERATURA

- [1] Hapesoft s.r.o., "Historie Automatizace." Accessed: Nov. 20, 2023. [Online]. Available: <http://plc-automatizace.cz/knihovna/historie/historie-automatizace.htm>
- [2] T. R. Borden and R. A. Cox, *Technician's Guide to Programmable Controllers 6th Edition*. 2012. [Online]. Available: [www.cengage.com/highered](http://www.cengage.com/highered)
- [3] William Bolton, *Programmable Logic Controllers*, 6th ed. 2015.
- [4] C3 controls, "A guide to programmable logic controllers: Back to basics." Accessed: Nov. 20, 2023. [Online]. Available: <https://www.c3controls.com/white-paper/back-to-plc-basics-guide-programmable-logic-controller/>
- [5] I. Švarc and Z. Automatizace, *VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ Fakulta strojího inženýrství Učební texty pro kombinovanou formu bakalářského studia*. Accessed: Nov. 11, 2023. [Online]. Available: [https://ucebnaaut.wz.cz/wp-content/uploads/2017/09/ZakladyAutomatizace\\_BRNO\\_Svarc.pdf](https://ucebnaaut.wz.cz/wp-content/uploads/2017/09/ZakladyAutomatizace_BRNO_Svarc.pdf)
- [6] Colin D. Simpson, *Programmable logic controllers*. Prentice hall, 1994. [Online]. Available: <https://archive.org/details/programmablelogi0000simp>
- [7] Chong Chen, "Introduction to programable logic controllers part I," 2020. Accessed: Jul. 20, 2023. [Online]. Available: <https://pdhonline.com/courses/e116/PLC-module1.pdf>
- [8] Siemens AG, "Structure and Use of the CPU Memory," 2013. Accessed: Jul. 18, 2023. [Online]. Available: [https://cache.industry.siemens.com/dl/files/101/59193101/att\\_897341/v2/s71500\\_structure\\_and\\_use\\_of\\_the\\_PLC\\_memory\\_function\\_manual\\_en-US\\_en-US.pdf](https://cache.industry.siemens.com/dl/files/101/59193101/att_897341/v2/s71500_structure_and_use_of_the_PLC_memory_function_manual_en-US_en-US.pdf)
- [9] Hapesoft s.r.o., "Analogové vstupy." Accessed: Sep. 01, 2023. [Online]. Available: <http://plc-automatizace.cz/knihovna/periferie/analogove/analogove-vstupy.htm>
- [10] Hapesoft s.r.o., "Analogové výstupy." Accessed: Dec. 10, 2023. [Online]. Available: <http://plc-automatizace.cz/knihovna/periferie/analogove/analogove-vystupy.htm>
- [11] R. Zurawski, *Industrial communication technology handbook*, 2nd ed. 2017.
- [12] Siemens AG, "PROFINET-Standard pro průmyslový Ethernet v automatizaci." Accessed: Dec. 09, 2023. [Online]. Available: <https://docplayer.cz/780224-Profinet-standard-pro-prumyslovy-ethernet-v-automatizaci.html>
- [13] R. Pigan and M. Metter, *Pigan / Metter Automating with PROFINET*, 2nd ed. 2008.
- [14] Profibus & Profinet International, "PROFINET System Description Technology and Application PROFINET System Description." Accessed: Dec. 09, 2023. [Online]. Available: <https://www.profibus.com/index.php?eID=dumpFile&t=f&f=82430&token=7cbb78f5ba6b3e17762ab594f803f1901eb24fdf>

- [15] Profibus & Profinet International, "PROFINET Technology and Application System Description Open Solutions for the World of Automation." Accessed: Dec. 09, 2023. [Online]. Available: [https://readthedocs.web.cern.ch/download/attachments/21178119/Profinet\\_English.pdf?version=1&modificationDate=1246437858000&api=v2](https://readthedocs.web.cern.ch/download/attachments/21178119/Profinet_English.pdf?version=1&modificationDate=1246437858000&api=v2)
- [16] "Siemens SIMATIC slaví 60. narozeniny." Accessed: Dec. 18, 2023. [Online]. Available: <https://www.blaja.cz/plc-automaty/siemens-simatic-slavi-60-narozneniny.html>
- [17] Antonio Armenta, "Siemens SIMATIC PLCs - Hardware history." Accessed: Dec. 18, 2023. [Online]. Available: <https://control.com/technical-articles/siemens-simatic-plcs-hardware-history/>
- [18] Siemens AG, "Řídící systémy SIMATIC - chytré řešení pro vaše automatizační úlohy." Accessed: Dec. 18, 2023. [Online]. Available: <https://www.siemens.com/cz/cs/products/automation/systems/industrial/plc.html>
- [19] Siemens AG, "SIMATIC controllers - the smart choice for your automation tasks." Accessed: Dec. 18, 2023. [Online]. Available: <https://assets.new.siemens.com/siemens/assets/api/uuid:c67e0a36-d9a5-4fbb-8704-a83eae40708e/version-of-dffa-b10100-02-controller8-seiter-1144dpi.pdf>
- [20] Siemens AG, "Řídící systémy SIMATIC ." Accessed: Dec. 18, 2023. [Online]. Available: <https://www.siemens.com/cz/cs/products/automation/systems/industrial/plc.html>
- [21] F. Adamo, F. Attivissimo, G. Cavone, and N. Giaquinto, "SCADA/HMI Systems in Advanced Educational Courses," *IEEE Trans Instrum Meas*, vol. 56, no. 1, pp. 4–10, Jan. 2007, doi: 10.1109/tim.2006.887216.
- [22] Lettl Jiří, Pavelka Jiří, and Bauer Jan, *Výkonová elektroniky*, 4. vydání. Praha: ČVUT, 2019.
- [23] Siemens AG, "Motion Control Drives siemens.com/drives SINAMICS S120 and SIMOTICS," 2017. Accessed: Dec. 18, 2023. [Online]. Available: [https://cache.industry.siemens.com/dl/files/019/109747019/att\\_918077/v1/motion-control-drives-D21-4-complete--English-2017.pdf](https://cache.industry.siemens.com/dl/files/019/109747019/att_918077/v1/motion-control-drives-D21-4-complete--English-2017.pdf)
- [24] Profibus & Profinet International, "PROFIsafe Technology and Application - System Description."
- [25] "Step response." Accessed: Dec. 03, 2023. [Online]. Available: <https://tex.stackexchange.com/questions/312523/step-response-in-tikz>
- [26] R. I. R. Garzinová, Z. Jančíková, O. Zimný, I. R. Garzinová, and I. O. Zimný, *VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA OSTRAVA FAKULTA STROJNÍ ZÁKLADY AUTOMATIZACE TECHNOLOGICKÝCH PROCESŮ V TEORII Ostrava 2013*. Accessed: Dec. 10, 2023. [Online]. Available: [https://projekty.fs.vsb.cz/463/edubase/VY\\_01\\_035/Z%C3%A1klady%20automatizace%](https://projekty.fs.vsb.cz/463/edubase/VY_01_035/Z%C3%A1klady%20automatizace%20v%20teorii)

20technologick%C3%BDch%20proces%C5%AF%20v%20teorii/02%20Text%20pro%20e-learning/Z%C3%A1klady%20automatizace%20technologick%C3%BDch%20proces%C5%AF%20v%20teorii%20%2006.pdf

- [27] SIEMENS s.r.o. and ČVUT FEL katedra elektrických pohonů a trakce, “Řízení elektrických pohonů - Servomotory (prezentace)”, Accessed: Dec. 18, 2023. [Online]. Available: <https://moodle.fel.cvut.cz/mod/resource/view.php?id=216445>
- [28] Siemens AG, “Sinema Remote Connect V2.0.” Accessed: Nov. 28, 2023. [Online]. Available: [https://wydarzenia.siemens-info.com/siemensevents-srv/api/v1/event-file/49/file#:~:text=The%20management%20platform%20for%20remote%20networks%2C%20SINEMA%20Remote%20Connect%20\(SINEMA,the%20machine%20to%20be%20maintained.](https://wydarzenia.siemens-info.com/siemensevents-srv/api/v1/event-file/49/file#:~:text=The%20management%20platform%20for%20remote%20networks%2C%20SINEMA%20Remote%20Connect%20(SINEMA,the%20machine%20to%20be%20maintained.)
- [29] Siemens AG, “Wincc Sm@rtserver.” Accessed: Dec. 14, 2023. [Online]. Available: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10092250>
- [30] AutomatizaceHW, “Průmyslová komunikace OPC UA 1- díl.” Accessed: Dec. 09, 2024. [Online]. Available: <https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html>

