

**KATEDRA ELEKTRICKÝCH
POHONŮ A TRAKCE**

**ČESKÉ VYSOKÉ UČENÍ
TECHNICKÉ V PRAZE**



**FAKULTA ELEKTROTECHNICKÁ
ŘÍZENÍ POHONU JEDNÉ OSY POMOCÍ
PLC SIEMENS**

DIPLOMOVÁ PRÁCE

LEDEN 2022

**EVGENII
LITVINOV**



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Litvinov** Jméno: **Evgenii** Osobní číslo: **503567**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektrických pohonů a trakce**
Studijní program: **Elektrotechnika, energetika a management**
Specializace: **Elektrické pohony**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Řízení pohonu jedné osy pomocí PLC Siemens

Název diplomové práce anglicky:

Control and vizualization of one axis drive

Pokyny pro vypracování:

- 1) Nastavte komunikaci mezi PLC a měničem se servomotorem
- 2) V PLC vytvořte program pro ovládání pohonu jedné osy (rychlostní a polohová smyčka + rampy rozjezdu a brzdy)
- 3) Vytvořte demonstrační úlohu ovládání výtahu, včetně vizualizace na HMI panelu
- 4) Implementujte funkci paměti zvolených pater

Seznam doporučené literatury:

dodá vedoucí

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: **02.02.2022** Termín odevzdání diplomové práce: **10.01.2023**

Platnost zadání diplomové práce: **30.09.2023**

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

Rád bych poděkoval vedoucímu práce Ing. Janu Bauerovi, PhD., za podporu a odborný výklad při zpracování této práce. Rád bych také poděkoval Katedře pohonů a trakce Fakulty elektrotechnické ČVUT za poskytnutí modelu výtahu.

PROHLÁŠENÍ

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

V Praze dne 8. ledna 2023

.....

ABSTRAKT

Tato práce se zabývá automatizací a programováním PLC. Cílem této práce je vytvořit model pětipodlažního výtahu řízeného PLC Siemens SIMATIC S7-1200 spolu s vizualizací ovládacího panelu výtahu pomocí panelu Siemens KTP700 Basic HMI. V práci je nastíněna teorie programovatelných logických automatů, jejich konstrukce a základní nástroje pro jejich programování, úvod do vývojového prostředí TIA Portal a popis základních algoritmů a funkcí v TIA Portal.

Klíčová slova: ČVUT FEL, katedra elektrických pohonů a trakce, závěrečná práce, automatizace, PLC, Siemens, HMI, řízení, výtah, TIA Portal, servopohon.

ABSTRACT

This work deals with automation and PLC programming. The aim of this work is to create a model of a five-story elevator controlled by Siemens SIMATIC S7-1200 PLC along with visualization of the elevator control panel using Siemens KTP700 Basic HMI panel. The thesis outlines the theory of programmable logic automata, their design and basic programming tools, an introduction to the TIA Portal development environment and a description of the basic algorithms and functions in TIA Portal.

Keywords: CTU FEL, Department of Electric Drives and Traction, final thesis, automation, PLC, Siemens, HMI, control, elevator, TIA Portal, actuator.

ÚVOD.....	12
HISTORIE PROGRAMOVATELNÝCH LOGICKÝCH AUTOMATŮ.....	13
KAPITOLA 1: POPIS FUNKCÍ PLC	14
1.1 ARCHITEKTURA PLC.....	14
1.2 POPIS FUNKCÍ CPU.....	15
1.3 PAMĚŤ PLC	16
1.4 KOMUNIKACE	16
1.5 HMI PANEL	16
1.6 NAPÁJECÍ ZDROJ.....	16
1.7 VSTUPNÍ SIGNÁLY	17
1.8 VÝSTUPNÍ SIGNÁLY	17
1.9 I/O BLOK	17
KAPITOLA 2: PROGRAMOVACÍ JAZYKY A ZPŮSOBY PROGRAMOVÁNÍ	18
2.1 PROGRAMOVACÍ ZAŘÍZENÍ	18
2.2 PROGRAMOVACÍ JAZYKY.....	18
2.3 SFC – SEKVENČNÍ FUNKČNÍ DIAGRAM	18
2.4 LD – LINIOVÉ SCHÉMA	19
2.5 FBD – FUNKČNÍ BLOKOVÉ SCHÉMA	19
2.6 IL – SEZNAM INSTRUKCÍ	20
2.7 ST – STRUKTUROVANÝ TEXT.....	20
KAPITOLA 3: POPIS SOUČÁSTÍ SYSTÉMU	21
3.1 HARDWAROVÁ SOUČÁST	21
3.2 SOFTWAREOVÁ SOUČÁST.....	24
KAPITOLA 4: NAVRHOVANÁ KONCEPCE ŘÍZENÍ VÝTAHU.....	25
4.1 POPIS SOUSTAVY.....	25
4.2 KONCEPCE ŘÍZENÍ VÝTAHU.....	25
KAPITOLA 5: ZAČÁTEK PRÁCE V TIA PORTAL	27
5.1 VYTVOŘENÍ NOVÉHO PROJEKTU.....	27
5.2 PŘIDÁNÍ PLC DO PROJEKTU	27
5.3 PRACOVNÍ OKNO TIA PORTAL	28
5.4 NASTAVENÍ PLC	29
5.5 PROGRAMOVACÍ OKNO.....	29
5.6 VYTVÁŘENÍ TAGŮ PLC.....	29
KAPITOLA 6: ŘÍZENÍ MODELU VÝTAHU.....	31
6.1 PŘIDÁNÍ SERVOPOHONU DO PROJEKTU TIA PORTAL	31
6.2 KONFIGURACE SERVOPOHONU.....	31
6.3 PŘIDÁNÍ A KONFIGURACE TECHNOLOGICKÉHO OBJEKTU	32
6.4 MOTION CONTROL.....	34
KAPITOLA 7: NÁVRH HMI PANELU.....	37
7.1 PŘIDÁNÍ HMI PANELU DO PROJEKTU TIA PORTAL	37
7.2 KONFIGURACE HMI PANELU	37
7.3 VYTVÁŘENÍ A PROGRAMOVÁNÍ OVLÁDACÍCH OBRAZOVEK.....	37
KAPITOLA 8: PROGRAMOVÁNÍ MODELU VÝTAHU	41
8.1 SNÍMAČ POLOHY VÝTAHU	41
8.2 ŘEŠENÍ PAMĚTI.....	41

8.3	URČENÍ SMĚRU JÍZDY VÝTAHU	42
8.4	TLAČÍTKO STOP	43
8.5	ZMĚNA RYCHLOSTI Z DISPEČINKU	43
8.6	ZMĚNA POLOHY VÝTAHU.....	43
8.7	LED INDIKACE	44
8.8	RESETOVÁNÍ SNÍMAČE POLOHY VÝTAHU	44
	KAPITOLA 9: SPUŠTĚNÍ CELÉ APLIKACE.....	46
9.1	STARTOVACÍ PROGRAM	46
9.2	NORMÁLNÍ REŽIM PROVOZU	47
9.3	ZÁZNAM SIGNÁLU Z TLAČÍTKA DO PAMĚTI	49
	KAPITOLA 10: REALIZACE PROJEKTU ŘÍZENÉHO VÝTAHU	50
10.1	NAHRANÍ PROJEKTU DO PLC	50
10.2	ONLINE MONITORING	50
	KAPITOLA 11: NÁVRH OVLÁDACÍHO ROZHRANÍ VÝTAHU.....	52
11.1	SERVISNÍ ROZHRANÍ.....	52
11.2	UŽIVATELSKÉ ROZHRANÍ	52
	ZÁVĚR.....	54
	LITERATURA.....	55
	PŘÍLOHA A : TECHNICKÉ ÚDAJE ZAŘÍZENÍ SESTAVY.....	57
A.1	SIMATIC S7-1200 CPU 1215C.....	57
A.2	SIEMENS SIMATIC PM 1207.....	61
A.3	SIEMENS SIMATIC NET CSM 1277	62
A.4	SIEMENS SIMATIC HMI KTP700 BASIC	65
A.5	SIEMENS SINAMICS V90 PN	67
A.6	SIEMENS SIMOTICS S-1FL6	68

SEZNAM OBRÁZKŮ

Obrázek 1.1 Struktura PLC.....	14
Obrázek 1.2 Monolitická architektura PLC [14] ..	15
Obrázek 1.3 Modulární architektura PLC [14]	15
Obrázek 1.4 HMI panel [17].....	16
Obrázek 2.1 Programovací zařízení [11].....	18
Obrázek 2.2 Sekvenční funkční diagram (SFC) [12]	19
Obrázek 2.3 liniové schéma (LD)	19
Obrázek 2.4 Funkční blokové schéma (FBD).....	20
Obrázek 2.5 Seznam instrukcí (IL) [13]	20
Obrázek 2.6 Strukturovaný text (ST) [13]	20
Obrázek 3.1 PLC SIMATIC S7-1200 CPU 1215C [14]	21
Obrázek 3.2 Napájecí zdroj SIMATIC PM 1207 [15]	22
Obrázek 3.3 komunikační modul SIMATIC NET CSM 1277 [16]	22
Obrázek 3.4 HMI panel SIMATIC HMI KTP700 Basic [17]	23
Obrázek 3.5 Měnič SINAMICS V90 PN [18].....	23
Obrázek 3.6 Měnič SINAMICS V90 PN [18].....	24
Obrázek 4.1 model výtahu	25
Obrázek 4.2 technologické schéma modelu výtahu	26
Obrázek 5.1 založení nového projektu	27
Obrázek 5.2 Přidání PLC do projektu	28
Obrázek 5.3 Pracovní okno Tia Portal	28
Obrázek 5.4 Nastavení IP adresy a masky PLC	29
Obrázek 5.5 Programovací okno Tia Portal.....	29
Obrázek 5.6 Vytváření tagů PLC.....	30
Obrázek 5.7 Vstupy a výstupy PLC.....	30
Obrázek 6.1 Přidání servopohonu do projektu ...	31
Obrázek 6.2 Přidání telegramu 3 do projektu.....	32
Obrázek 6.3 Přidání IP adresy servopohonu	32
Obrázek 6.4 Přidání TO do projektu.....	32
Obrázek 6.5 Přidání pohonu a enkodéru do TO ..	33
Obrázek 6.6 Nastavení dynamiky motoru.....	34
Obrázek 6.7 Dynamické parametry motoru a Active Homing	34
Obrázek 6.8 Funkce MC_Home.....	35
Obrázek 6.9 Funkce MC_Reset.....	35

Obrázek 6.10 Funkce MC_Home	36
Obrázek 6.11 Funkce MC_MoveAbsolute	36
Obrázek 7.1 Přidání HMI panelu	37
Obrázek 7.2 Nastavení IP adresy	37
Obrázek 7.3 Přidání PLC tagů	38
Obrázek 7.4 Vytváření tlačítka	39
Obrázek 7.5 Vytváření přepínače	39
Obrázek 7.6 Vytváření viditelnosti	40
Obrázek 7.7 Nastavení barev	40
Obrázek 8.1 Programování snímačů polohy	41
Obrázek 8.2 Logika zmáčknutí tlačítka	41
Obrázek 8.3 Aktivace příkazů z paměti	42
Obrázek 8.4 Resetování příkazů	42
Obrázek 8.5 Resetování šipky směru jízdy	43
Obrázek 8.6 Nastavení rychlosti z panelu	43
Obrázek 8.7 Změna polohy výtahu	44
Obrázek 8.8 Zpracování polohy pro LED indikace	44
Obrázek 8.9 Vynulování snímače pro opětovné stisknutí tlačítka	45
Obrázek 9.1 Algoritmus startovacího programu.	46
Obrázek 9.2 Algoritmus normálního provozního režimu.	48
Obrázek 9.3 Algoritmus pro zápis do paměťové proměnné.	49
Obrázek 10.1 Nahrání projektu do PLC	50
Obrázek 10.2 Online monitoring v Tia portal	51
Obrázek 11.1 Servisní rozhraní	52
Obrázek 11.2 Ovládací panel výtahu	53

ZKRATKY A SPECIÁLNÍ VÝRAZY

- PLC – programovatelný logický automat
- I/O blok – blok se vstupy a výstupy
- CPU – centrální procesorová jednotka
- RAM – paměť s přímým přístupem
- ON/OFF – Zapnuto/vypnuto
- TO – technologický objekt
- DC – stejnosměrný proud (direct current)
- AC – střídavý proud (alternating current)
- ROM – paměť pouze pro čtení

ÚVOD

Programovatelný logický automat S7-1200 je schopen řešit logické úlohy, úlohy automatické regulace a řízení pohybu i matematické zpracování informací. PLC S7-1200 má široký rozsah funkcí a relativně nízkou cenu. Díky tomu je vhodný pro všechna odvětví průmyslu a systémy automatizace budov. Díky modulární struktuře a relativně malým rozměrům v kombinaci s vysokým výpočetním výkonem je PLC S7-1200 vhodný pro širokou škálu automatizačních úloh. Tyto úlohy mohou sahát od výměny jednoduchých reléových obvodů až po složitější úlohy, jako je vytváření distribuovaných automatizačních struktur s využitím intenzivního síťového propojení. PLC S7-1200 lze použít v oblastech, kde bylo v minulosti použití ovladačů považováno za ekonomické nevýhodné a k řešení automatizačních problémů se používala specializovaná elektronická zařízení.

Cílem tohoto projektu je vybudování řídicího systému pro pohon jedné osy pomocí PLC Siemens Simatic S7-1200, který bude plnit funkce výtahu. Tento článek pojednává o bodech souvisejících s definicí toho, co je PLC, způsoby programování řídicí jednotky Simatic S7-1200 pomocí programovacího jazyku FBD, protokoly pro komunikaci master/slave ASCII, USS, PROFINET a MODBUS TRU a technologii připojení IO-link pro komunikační senzory a pohon.

Výsledkem této práce bude vývoj software a metodická podpora pro studium principů programování a provozu PLC Simatic S7-1200 pro úlohy.

V Praze dne 28.12.2022

EVGENII LITVINOV

HISTORIE PROGRAMOVATELNÝCH LOGICKÝCH AUTOMATŮ

Před nástupem PLC bylo ovládání strojů možné pouze pomocí relé, které fungovalo pomocí cívky, jež pomocí magnetické síly přepínala spínač do dvou poloh: zapnuto nebo vypnuto. Řízení velkého počtu procesních jednotek bylo realizováno pomocí velkého počtu reléových spínačů, které byly umístěny ve speciálních skříních, které tvořily „reléové skříně“. Tyto reléové spínače byly zase ovládány pomocí řídicích relé. Při řešení optimalizačních úloh pro velká průmyslová zařízení je nutné řídit velký počet mechanismů současně, na dálku a s minimálním zásahem člověka.

"Programovatelný automat podrobně popsal první den roku 1968" Richard Morley [4]. Požadavky na programovatelný logický automat byly následující:

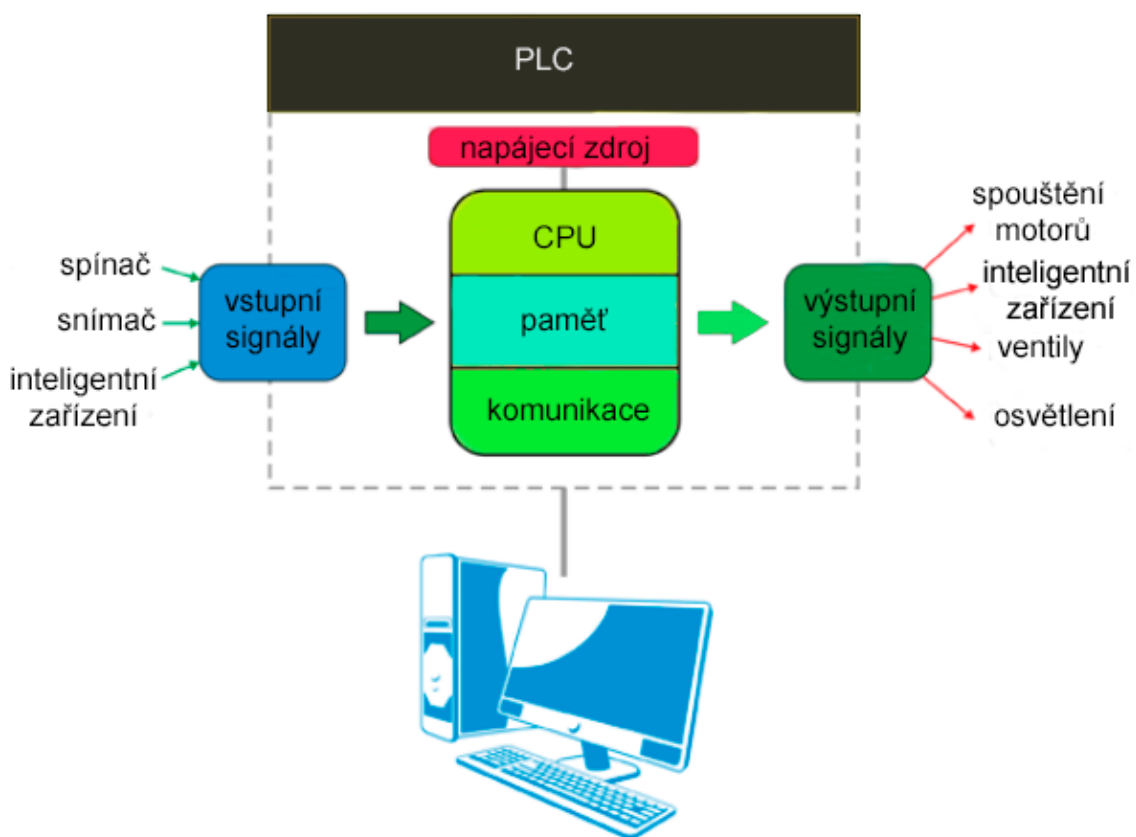
- Polovodičový systém, který byl stejně flexibilní jako počítač, ale cena musí být konkurenceschopná s relé,
- Snadné programování v souladu s přijatou reléovou logikou,
- Dostatečná ochrana pro provoz v nestandardních podmínkách, jako jsou: nečistoty, vlhkost, elektromagnetismus a vibrace,
- Modulární konstrukce jističů pro snadnou výměnu jednotlivých komponent a možnost rozšíření celého systému [4].

„Morley byl zakladatelem a prezidentem společnosti *Bedford Associates*, kde v roce 1968 vytvořil první programovatelný logický automat Modicon“ [4]. PLC se dnes používají v každém výrobním závodě a oproti reléovým spínačům mají několik výhod: jsou flexibilnější, mají menší rozměry, jsou spolehlivější, mohou komunikovat s dalšími zařízeními a být ovládány přes internet, mohou rychleji odhalovat chyby, vyžadují také menší náklady na změnu svých funkcí a struktury.

KAPITOLA 1: POPIS FUNKCÍ PLC

Zkrácený anglický název PLC pochází z *Programmable Logic Controller*, česky Programovatelný logický automat. PLC je soubor vstupů a výstupů pro řešení konkrétních úloh ve výrobě na základě dané logiky. Dva identické PLC mohou ve výrobě vykonávat zcela odlišné úlohy. Úkolem každého PLC je řídit mechanismus nebo skupinu mechanismů na základě předem naprogramované logiky. Program je uložen v paměti PLC a PLC řídí stroj podle zapsaného programu. Původně se PLC používal ke zpracování sekvenčních logických procesů, což vysvětluje jeho název "logický".

PLC se skládá z následujících hlavních částí: vstupní část, centrální část a výstupní část. Vstupní část se skládá ze sady analogových a digitálních vstupů, které váš regulátor obsahuje. Centrální část se skládá z centrálního procesoru, paměti a komunikačního systému. Centrální část je určena ke zpracování vstupních signálů regulátoru a vytváření výstupních signálů podle naprogramované logiky. Výstupní část PLC se skládá ze sady analogových a diskretních výstupů pro připojení řídicí jednotky k řízeným objektům.



Obrázek 1.1 Struktura PLC

1.1 Architektura PLC

PLC lze podle architektury rozdělit na dva hlavní typy: uzavřené nebo tzv. proprietární či otevřené. Otevřený systém umožňuje integraci dalších prvků s otevřeným systémem nebo s jeho složkami [5], [1]. U uzavřeného systému jsou změny součástí nebo prvků stanoveny výrobcem.

Typy monolitické nebo modulární architektury určují prostorovou polohu systémových komponent. V monolitické architektuře PLC jsou všechny komponenty pevně umístěny v jedné skříni. V modulární architektuře PLC mohou být všechny nebo některé části od sebe strukturálně odděleny.



Obrázek 1.2 Monolitická architektura PLC [14] (upraveno).



Obrázek 1.3 Modulární architektura PLC [14] (upraveno).

1.2 Popis funkcí CPU

CPU nebo centrální procesorová jednotka obsahuje paměťové zařízení, hodiny reálného času a *WatchDog* [1]. Obrovská rozmanitost úloh, které jsou PLC přidělovány, a silná závislost ceny na výkonu řídicí jednotky jsou důvodem velké rozmanitosti mikroprocesorů, které se v nich používají, od jednoduchých a levných mikroprocesorů od společnosti *Atmel* a *Microchip* až po nejvýkonnější mikroprocesory řady *Intel Pentium*, včetně dvoujádrových a čtyřjádrových [5].

PLC používají 2 a 4jádrové procesory. 8, 16, 32 jader se používá ve výkonnějších PLC ve velkých průmyslových odvětvích. Kromě počtu jader procesoru mají PLC i další důležité vlastnosti procesoru, jako například:

- Architektura, z nichž nejběžnější je architektura RISC (*Reduced Instruction Set Computing* - Počítače s redukovanou instrukční sadou),
- Hodinová frekvence. Hodinová frekvence popisuje výkon mikroprocesorů. Výkon procesorů se stejnou architekturou závisí na hodinové frekvenci,
- Maximální a normální provozní teplota,
- Spotřeba energie [5].

Výkon procesorů v PLC je však o několik generací pozadu za procesory v osobních počítačích, což je v [5] přičítáno relativně malému trhu s procesory pro PLC. Na tuto zaostalost procesorů se však dá dívat i jinak. V průběhu let byly vyvinuty vlastní obvody procesorů, což zvýšilo jejich

spolehlivost a odolnost. Spolehlivost je u PLC nejdůležitějším parametrem, dokonce důležitějším než výkon.

1.3 Paměť PLC

Kapacita paměti určuje, kolik značek nebo proměnných může PLC uložit a zpracovat. Paměť PLC je rozdělena do několika úrovní hierarchie v závislosti na četnosti používání údajů v nich uložených a výkonnost. ROM – *Read Only Memory* (paměť pouze pro čtení) je typ počítačové paměti, která obsahuje nevolatilní trvalá data. RAM (*Random Access Memory*) - paměť s náhodným přístupem, která se obvykle používá k ukládání často se měnících dat, jako jsou proměnné nebo značky. PLC má také sadu registrů, které jsou nejrychlejšími paměťovými prvky, protože je využívá aritmeticko-logická jednotka (ALU) k provádění elementárních instrukcí procesoru [5].

1.4 Komunikace

Úkolem komunikace je přijímat signály ze vstupu a vysílat signály na výstup a zároveň udržovat komunikaci s programovatelným zařízením. Tento projekt využívá protokol PROFINET. Protokol PROFINET se skládá z protokolu PROFIBUS založeného na protokolu ETHERNET nebo PN/IE (PROFINET/*INDUSTRIAL ETHERNET*). Komunikace může být podporována i s dalšími zařízeními, jako jsou panely HMI a měniče, které jsou v projektu rovněž použity.

1.5 HMI panel

Zařízení HMI (*human-machine interface* — rozhraní člověk-stroj) jsou určena pro interakci mezi obsluhou a pracovním procesem. Dotykový panel a klávesnice jsou hlavními prvky pro sledování a řízení moderních výrobních procesů. Komunikaci s ostatními prvky distribuovaného systému zajišťují moderní zařízení HMI prostřednictvím digitální sítě.

HMI je programovatelné zařízení, stejně jako moduly I/O a PLC. Ke komunikaci s PLC používá komunikační protokoly, jako je PROFIBUS DP nebo PROFINET nebo jiné [5]. Panel HMI v projektu slouží k řízení procesu a vizualizaci procesních stavů.



Obrázek 1.4 HMI panel [17] (upraveno).

1.6 Napájecí zdroj

PLC musí být stejně jako každé jiné zařízení připojeno ke zdroji napájení, v České republice je to 230 V AC. Úkolem PLC je také transformovat toto napětí pro digitální a analogové vstupy a výstupy. Všechny digitální vstupy v běžném provedení regulátorů jsou obvykle navrženy tak, aby přijímaly standardní signály při napětí 24 V DC. Typická hodnota proudu pro diskretní vstup při vstupním

napětí 24 V je 10 mA. SIMATIC S7-1200 má jmenovité napětí pro digitální vstupy 24 V DC, pulzní napětí pro digitální vstupy je 35 V DC. Napětí pro digitální výstupy je 5 V DC až 30 V DC [8].

1.7 Vstupní signály

Diskrétní vstupy PLC slouží k příjmu signálů z diskretních senzorů, jako jsou snímače pohybu, detektory kouře, snímače hladiny vody a další. Hodnota napětí těchto snímačů odpovídá logické jedničce na programovací úrovni. Nízká úroveň napětí odpovídá logické nule [1], [2].

Analogové signály mohou mít na rozdíl od diskretních signálů složitý tvar a závislost, např. úroveň proudu nebo napětí jsou analogové signály. Analogový vstup regulátoru je obvykle analogově-digitální převodník, který převádí hodnotu analogového signálu na digitální signál [5]. Výsledkem je diskretní proměnná o určité kapacitě číslic. Protože existuje mnoho různých fyzikálních jevů, které jsou příčinou vzniku velkého množství analogových senzorů, musí pro každý z nich existovat odpovídající vstupní zařízení. Aby se snížil počet použitých vstupních modulů, používají se zařízení, která signály normalizují. Tato zařízení převádějí měřenou fyzikální veličinu na standardní elektrický signál. Převedené analogové signály se dělí na dva typy: proudové signály 4-20 mA nebo napěťové signály 0-10 V. Analogový signál je zpracováván vnitřní logikou PLC, uživatel systému pracuje s připravenými hodnotami analogových veličin v paměti RAM [1], [5].

Speciální vstupy se používají v případech, kdy analogové a digitální vstupy nejsou schopny uspokojit potřeby systému. Speciální vstupy se používají v případech, kdy je zpracování signálu programově náročné, např. vyžaduje mnoho času nebo signál vyžaduje specifické parametry napájení. PLC jsou často vybaveny speciálními výstupy pro detekci hran a počítání impulzů a vstupy pro příjem signálů PWM.

1.8 Výstupní signály

Diskrétní výstupy stejně jako i diskretní vstup představují z hlediska programování 1 bit informace. Nejběžnějším typem diskretních výstupů jsou reléové kontakty. Reléový kontakt je jednoduchý a běžný, ale má nevýhody relé – omezenou životnost, nízkou dobu odezvy, zničení kontaktů při provozu s indukční zátěží [3]. Alternativním a nejběžnějším řešením digitálního vstupu je elektronický prvek – obvod s otevřeným kolektorem [7], [5].

Analogové výstupy regulátoru jsou vytvořeny v digitální podobě a k jejich převodu na analogovou hodnotu se používá digitálně-analogový převodník nebo D/A převodník. Pro analogové výstupy jsou nejběžnější standardní rozsahy stejnosměrného napětí -10 až +10 V a 0 až 10 V a pro proudové vstupy 0 až 20 mA a 4 až 20 mA [5].

Speciální výstupy a speciální vstupy se používají ke generování frekvenčních nebo pulzně šířkových signálů a řídicích signálů pro konkrétní zařízení.

1.9 I/O blok

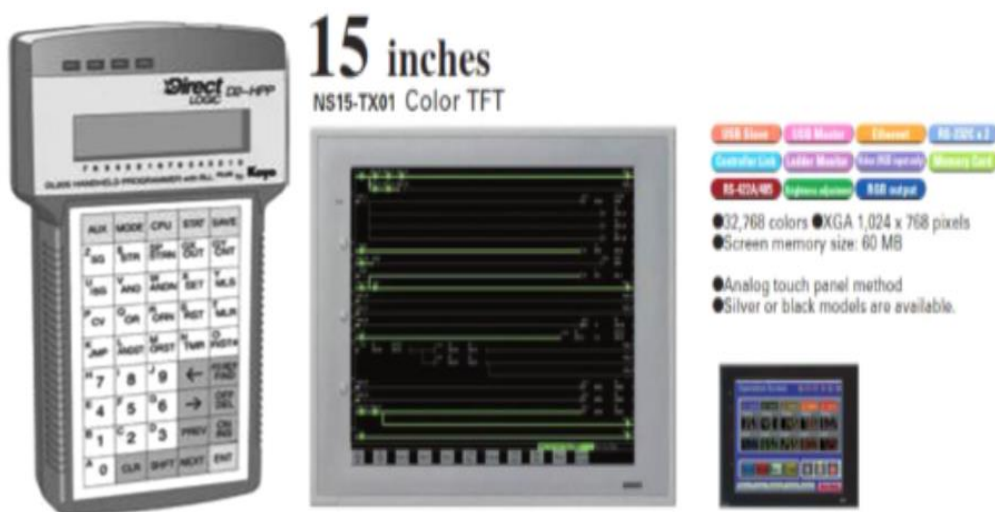
I/O moduly (vstupní/výstupní moduly) slouží jako konektory mezi procesorem a I/O zařízeními. Signál ze vstupu je zpracován tak, aby jej mohl zpracovat procesor. Po zpracování signálu procesorem jsou zpracované signály odeslány podle programu na výstupy PLC pro řízení pracovního postupu.

KAPITOLA 2: PROGRAMOVACÍ JAZYKY A ZPŮSOBY PROGRAMOVÁNÍ

Při práci s PLC je hlavním úkolem projektanta řídicího systému vytvořit program. K tomu je zapotřebí zařízení, pomocí kterého lze vytvořit řídicí logiku, software pro vytvoření programu a znalost programovacího jazyka. Průmyslové PLC se programují v jazycích podle normy IEC61131-3. Existuje celkem 5 základních programovacích jazyků; někteří výrobci přidávají vlastní programovací jazyky. Příkladem je jazyk CFC (*Continuous Function Chart*).

2.1 Programovací zařízení

Abyste mohli vytvořit program PLC, musíte mít zařízení, které je schopno program vytvořit. K vytvoření programu lze použít nejen počítač, ale jakékoli zařízení, které umožňuje vytvořit program a načíst jej do paměti PLC. Příkladem může být programovací zařízení na obrázku 1.4 nebo programovací zařízení s klávesnicí a displejem. Ve většině případů je program vytvořen na vašem počítači. Výrobci PLC mají téměř vždy vlastní software pro tvorbu programu [5].



Obrázek 2.1 Programovací zařízení [11] (upraveno).

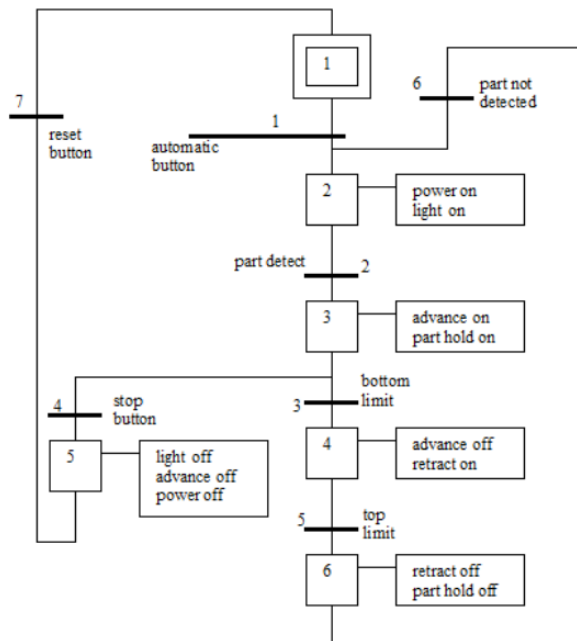
2.2 Programovací jazyky

„Zpočátku existovaly PLC, které bylo možné programovat ve strojově orientovaných jazycích – assemblerech“ [5. Str 50]. I malé změny vyžadovaly zásah speciálně vyškoleného programátora. V současné době se PLC programují převážně v jazycích nízké úrovně (FBD, LD) a střední úrovně (Pascal, C++, SCL) V současné době norma IEC61131-3 podporuje 5 programovacích jazyků [12]:

- SFC – *Sequential Function Chart* nebo sekvenční funkční diagram,
- LD – *Ladder diagram* nebo jazyk liniových diagramů,
- FBD – *Function Block Diagram* nebo jazyk funkčního blokového schématu,
- IL – *Instruction List* nebo jazyk seznamu instrukcí,
- ST – *Structured Text* nebo jazyk strukturovaného textu.

2.3 SFC – sekvenční funkční diagram

Jazyk SFC (*Sequential Function Chart*) popisuje sekvence chování řídicích programů. Sekvence se skládají z kroků a přechodů. Program se skládá ze stavů systému a souvisejících bloků akcí. K přechodu do dalšího kroku dojde, pokud jsou splněny předem naprogramované podmínky [12]. Hlavní výhodou tohoto jazyka je, že program je rozdělen do sekcí a každá sekce může být napsána v libovolném programovacím jazyce, který je v softwaru k dispozici.



Obrázek 2.2 Sekvenční funkční diagram (SFC) [12] (upraveno).

2.4 LD – liniové schéma

Liniové schéma (LD) nazývané také jazyk kontaktních diagramů je grafický programovací jazyk, který je založen na prezentaci reléové logiky. Při vytváření logiky programu se používají grafické komponenty, které vytvářejí grafické znázornění podobné elektrickým reléovým obvodům.

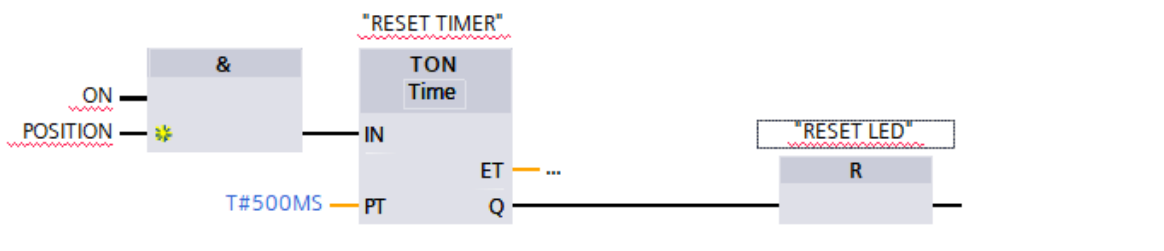
Hlavní myšlenkou autorů jazyka LD bylo usnadnit přechod odborníků z reléové logiky na PLC [5]. „Jazyk liniových schémat je výhodné použít při zpracování jednoduchých logických operací, zejména pro jeho přehlednost, čitelnost a rychlou diagnostiku, ladící prostředky“ [13, s.16].



Obrázek 2.3 liniové schéma (LD)

2.5 FBD – funkční blokové schéma

Funkční blokové schéma (FBD) je grafický programovací jazyk využívající reléovou logiku. Podle normy IEC61131-3 musí mít funkční bloky čtvercový tvar. Vstupní signály jsou připojeny k levé straně obdélníků a výstupní signály k pravé straně. Každý blok představuje mezi program a slouží k přenosu dat mezi bloky. FBD používá bloky pro komunikaci, časování, operace s logickými funkcemi a speciální bloky [12]. Ve FBD je možné vytvářet nové bloky ve vyšších programovacích jazycích, jako je ST. Výhodou FBD je graficky přehledná reprezentace programu, na rozdíl od textových programovacích jazyků, jako jsou IL a ST.



Obrázek 2.4 Funkční blokové schéma (FBD)

2.6 IL – seznam instrukcí

Instruction List (IL) je textový jazyk, který je velmi podobný assembleru. Je vhodný pro uživatele s malými zkušenostmi s programováním, kteří však znají výrobní mechanismy. IL je textový seznam sekvenčních instrukcí. Každá instrukce je zapsána na samostatném řádku a může obsahovat i komentář. Instrukce může obsahovat čtyři pole oddělená mezerami nebo tabulátory [5,12].

1	OPN	"DB1"	%DB1	0	DB1	
2	L	%DBW2	%DBW2	0	16#4D2	DB1
3	L	%DBW4	%DBW4	0	16#3333	DB1
4	+I			0	14341	OS=0,OV=0,A0=0,A1=1
5	T	"Tag_15"	%MW4	0	14341	
6						
7	L	3.0	3.0	0	3.0	
8	T	"Tag_24"	%MD10	0	3.0	
9	L	5.0	5.0	0	5.0	

Obrázek 2.5 Seznam instrukcí (IL) [13] (upraveno).

2.7 ST – strukturovaný text

Strukturovaný text (ST, *Structured Text*) je vysokoúrovňový jazyk, který má společné rysy s jazyky Pascal a Basic. ST používá podobnou syntaxi a logiku pro sestavení programu, ale jsou zde určité rozdíly. „ST obsahuje mnoho konstrukcí pro přiřazování hodnot proměnným, pro volání funkcí a funkčních bloků, pro zápis podmíněných přechodových výrazů, pro výběr operátorů, pro konstrukci iteračních procesů“ [5, s.89]. Jazyk ST se používá pro programování složitých průmyslových procesů, kde nestačí použití reléové logiky a je vyžadován popis chování složitých funkcí a matematických výpočtů [5], [12], [13].

```

1
2 IF "CROSSING" THEN
3     "DB".CROSSING_MEMORY := 1;
4 END_IF;
5
6 IF "DB".CROSSING_MEMORY AND NOT "DB"."CROSSING_ACTIVE" AND NOT "DB".CROSSING_BYPASS THEN
7     "DB"."CROSSING_ACTIVE" := 1;
8 END_IF;
9
10
11 "MAIN_TIMER".TON(IN:= NOT "DB".MAIN_TIME_DONE OR "DB"."CROSSING_ACTIVE",
12     PT:=T#24S,
13     Q=>"DB".MAIN_TIME_DONE,
14     ET=>"DB".ELAPSED TIME);

```

Obrázek 2.6 Strukturovaný text (ST) [13] (upraveno).

KAPITOLA 3: POPIS SOUČÁSTÍ SYSTÉMU

V této kapitole jsou popsány všechny hardwarové a softwarové části systému. Hardwarová část systému se skládá z následujících komponent:

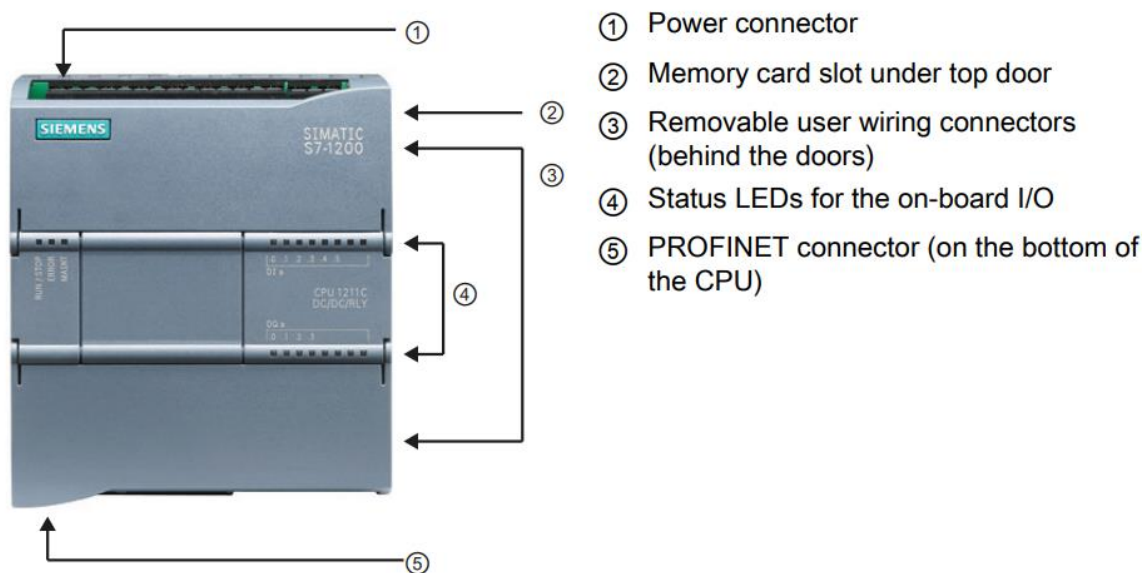
- PLC – SIMATIC S7-1200 CPU 1215C,
- HMI panel – Siemens SIMATIC HMI KTP700 Basic,
- Blok napájení – Siemens SIMATIC PM 1207,
- 4kanálový *switch*– Siemens SIMATIC NET CSM 1277,
- Měnič – Siemens SINAMICS V90 PN,
- Servomotor – Siemens SIMOTICS S-1FL6.

Ze softwarové části popíšeme samotné programovací prostředí TIA PORTAL a další užitečné programy jako S7 – PLCSIM a PRONETA, které pomohou při vytváření řídicí logiky a nastavení komunikace.

3.1 Hardwarová součást

3.1.1 SIMATIC S7-1200 CPU 1215C

Řada PLC S7 - 1200 1215C je modulární typ PLC s možností připojení komunikačních a signalizačních modulů. Spolu s PLC se používá čtyř kanálový komunikační modul SIMATIC NET CSM 1277 a napájecí modul SIMATIC PM 1207. PLC S7 1200 se dodává v kompaktním krytu se stupněm krytí IP 20. Řídicí jednotka je vybavena 14 digitálními vstupy a 10 digitálními výstupy a má 2 analogové vstupy a výstupy. CPU 1215C má vysoký výpočetní výkon, doba provádění logických operací nepřesahuje 0,1 μ s. Pracovní paměť RAM je 125 kB. PLC je vybaveno dvěma ethernetovými vstupy, které umožňují komunikaci s ostatními moduly prostřednictvím protokolu PROFINET. Přidáním modulů do systému je možné podporovat i další protokoly, například sériové rozhraní RS 232 nebo RS 422/ RS 485 založené na ASCII, GSM, LTE, USS a MODBUS RTU [15]. Technické údaje o instalaci viz příloha A.1.



Obrázek 3.1 PLC SIMATIC S7-1200 CPU 1215C [14] (upraveno).

3.1.2 Siemens SIMATIC PM 1207

SIMATIC PM 1207 je napájecí jednotka pro standardní průmyslové aplikace s rozsahem provozních teplot 0 až +60 °C. Napájecí jednotka musí být instalována v krajní levé nebo krajní pravé poloze vzhledem k řídicím modulům [15]. Napájecí jednotka SIMATIC PM 1207 je vybavena dvěma vstupy 120/230 V AC s automatickým nastavením úrovně vstupního napětí a jedním výstupem 24 V/2,5 A DC. Technické údaje o instalaci viz příloha A.2.



Obrázek 3.2 Napájecí zdroj SIMATIC PM 1207 [15] (upraveno).

3.1.3 Siemens SIMATIC NET CSM 1277

Komunikační modul Siemens SIMATIC NET CSM 1277 nebo neřízený čtyř kanálový přepínač nebo přepínací modul je určen k připojení SIMATIC S7-1200 a až 3 dalších uzlů k průmyslovému Ethernetu s rychlostí 10/100 Mbit/s. Komunikační modul má 4 porty RJ45 a externí napájení 24 V DC. Každý port je vybaven LED diodou pro indikaci připojení portu [16]. Technické údaje o instalaci naleznete v příloze A.3.



Obrázek 3.3 komunikační modul SIMATIC NET CSM 1277 [16] (upraveno).

3.1.4 Siemens SIMATIC HMI KTP700 Basic

Panel HMI (*human-machine interface* — rozhraní člověk-stroj) SIMATIC HMI KTP700 Basic je barevný dotykový panel pro realizaci interakce mezi uživatelem a strojem, který se používá pro vizualizaci dat a ovládání systému [5]. Panel má 8 programovatelných tlačítek, má krytí IP 65 a je napájen ze zdroje 24 V DC [17]. Podporuje rozhraní PROFINET a USB - Host. Konfigurace a programování panelu HMI je možné prostřednictvím WINCC BASIC V13 a STEP7 BASIC V13. Technické údaje o instalaci naleznete v příloze A.4.



Obrázek 3.4 HMI panel SIMATIC HMI KTP700 Basic [\[17\]](#) (upraveno).

3.1.5 Siemens SINAMICS V90 PN

SINAMICS V90 PN má dvě možnosti připojení v závislosti na aplikaci. Pro vysoce dynamické aplikace je měnič připojen ke zdroji 3 AC 380/480 V, pro nízko dynamické aplikace je měnič připojen ke zdroji 1 AC/3 AC 200/240 V. Měnič podporuje krokové rozhraní (USS / Modbus RTU) nebo integrované rozhraní PROFINET pro připojení k systému prostřednictvím profilu PROFIdrive [\[18\]](#). Technické údaje o instalaci naleznete v příloze [A.5](#).



Obrázek 3.5 Měnič SINAMICS V90 PN [\[18\]](#) (upraveno).

3.1.6 Siemens SIMOTICS S-1FL6

SIMOTICS S-1FL6 je servomotor Siemens s buzením permanentními magnety. Motor je přirozeně chlazený a teplo je odváděno krytem motoru. Motor je vybaven inkrementálním snímačem TTL 2500 imp/ot. SIMOTICS S-1FL6 má několik modelů v závislosti na aplikaci. Model motoru 024-2AF, který jsme použili v projektu má stupeň krytí IP 65, jmenovitý výkon 0,1 kW, jmenovitý točivý moment 0,32 Nm, jmenovité otáčky 3000 ot/min, jmenovitý proud 1,2 A, maximální proud 3,6 A. SIMOTICS S-1FL6 je navržen pro práci s měničem Siemens SINAMICS V90 PN [\[18\]](#). Technické údaje jednotky naleznete v příloze [A.6](#).



Obrázek 3.6 Měnič SINAMICS V90 PN [18] (upraveno).

3.2 Softwarová součást

3.2.1 TIA PORTAL

Tia Portal (*Totally Integrated Automation Portal*) je programovací prostředí vytvořené společností Siemens. Verze 11 tohoto softwaru umožňuje pracovat se všemi automatizačními komponentami Siemens. Programovací prostředí Tia Portal nabízí možnost vývoje projektů pro řídicí jednotky a distribuované I/O, konfiguraci HMI a systémů SCADA, ladění softwarových řídicích algoritmů a uvádění pohonů do provozu. To vše je spojeno ve společné softwarové struktuře a má jednotné uživatelské rozhraní. Tia Portal nabízí programování PLC v jazycích jako LAD, FBD, STL, SCL, GRAPH. Pro PLC SIMATIC S7-1200, které je v projektu použito, je možné programovat pouze v LAD, FBD a SCL [19].

3.2.2 PRONETA

PRONETA je pomůcka pro analýzu a konfiguraci sítí PROFINET. Pomocí programu PRONETA můžete rychle definovat seznam zařízení připojených k síti a přiřadit jednotlivým částem systému síťové názvy [21]. Software také umožňuje porovnávat různé topologie sítě, např. topologii požadovanou zákazníkem a skutečnou topologii. PRONETA umožňuje rychlou a pohodlnou kontrolu konfigurace systému od samého počátku před zapnutím procesoru [21].

3.2.3 S7 - PLCSIM

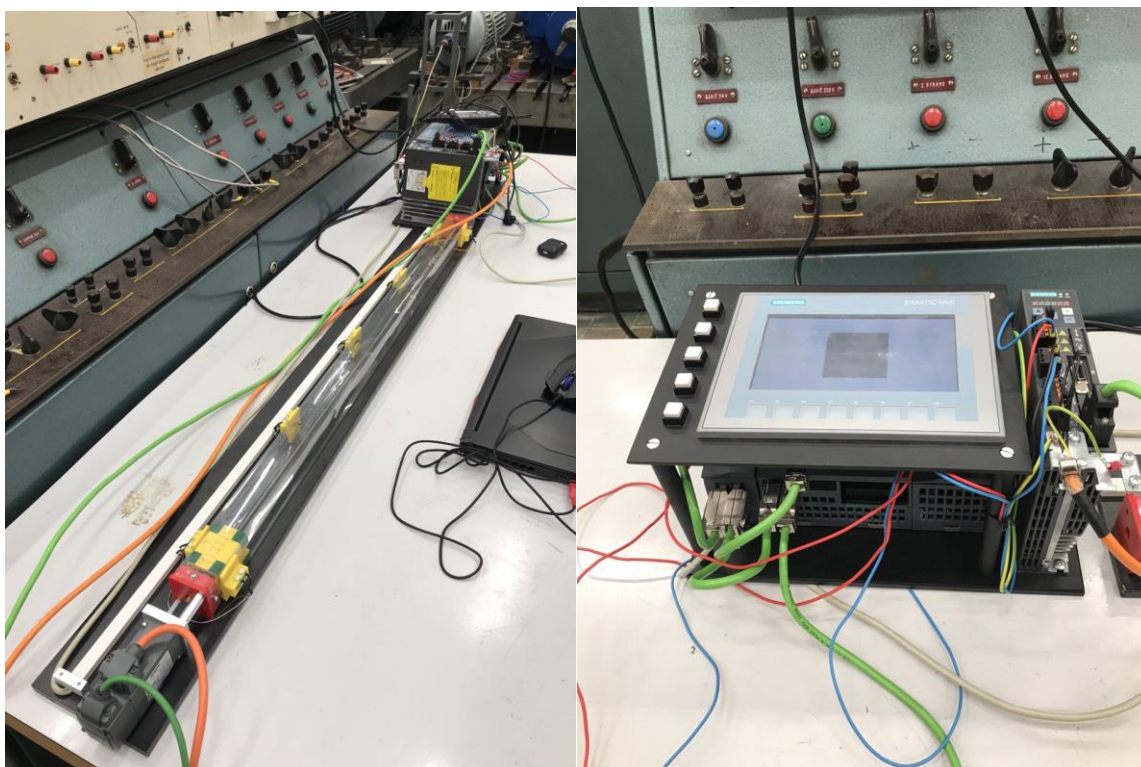
S7 - PLCSIM je pomocný program pro simulaci automatizačního procesu. Je vynikající pomůckou při školení v prostředí Tia portal, protože pomáhá simulovat všechny naprogramované procesory bez nutnosti nahrávat program do paměti PLC. S7 PLCSIM umožňuje také vzdálenou práci na programování a simulaci. Pomocí S7 - PLCSIM je možné simulovat pracovní postup umělým vytvářením vstupních dat, ať už jde o prosté přiřazení hodnoty proměnné pomocí tabulek SIM, nebo o práci se sekvencí definovaných dat pomocí funkce *Sequence*. Software rovněž umožňuje simulovat panel HMI [22].

KAPITOLA 4: NAVRHOVANÁ KONCEPCE ŘÍZENÍ VÝTAHU

4.1 Popis soustavy

Pro demonstraci principů ovládání čtyřpodlažního výtahu byl vytvořen systém, jehož součásti byly popsány v kapitole 3. Systém se skládá z PLC SIMATIC S7-1200 CPU 1215C, napájecího zdroje PLC SIMATIC PM 1207 a čtyř kanálového přepínače SIMATIC NET CSM 1277. Servomotor SIMOTICS S-1FL6 a měnič SINAMICS V90 PN, které společně tvoří elektrický pohon. Pro další ovládání výtahu lze použít panel SIMATIC HMI KTP700 Basic. Model obsahuje také osu s ochranným krytem, podél které se výtah pohybuje. Spínací kontakty na této ose jsou označeny žlutě a označují příslušná patra od 0 do 4, jak je znázorněno na levém obrázku 4.1. Na obrázku 4.1 vpravo vidíte systém řízení výtahu.

Model výtahu využívá principu šroubovitého pohybu, což znamená, že pohyb lze rozložit na dvě části: podél osy a kolem osy [24]. Rychlost pohybu podél osy bude přímo úměrná pohybu kolem její osy, tj. rychlosti otáčení našeho servomotoru.



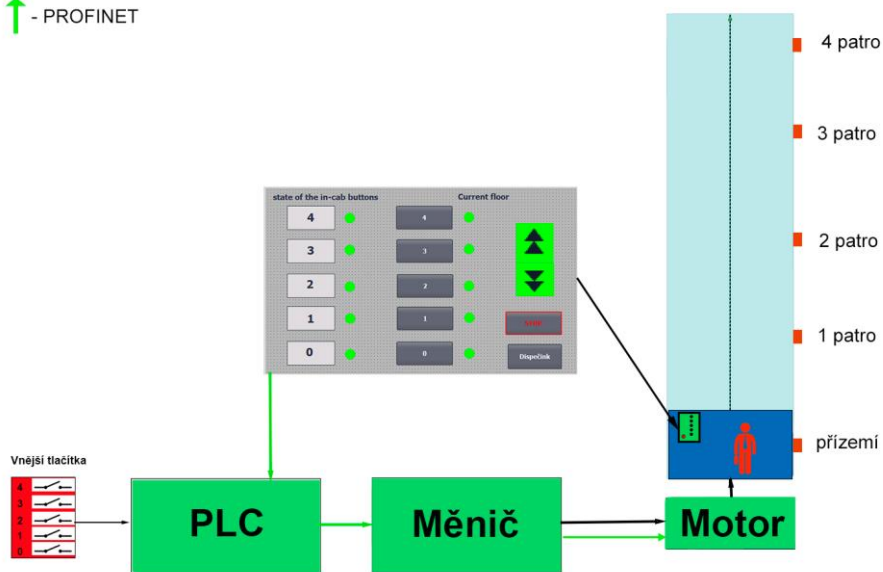
Obrázek 4.1 model výtahu

4.2 Koncepte řízení výtahu

Řídicí systém na pravém obrázku 4.1 má 5 fyzických tlačítek, která jsou připojena k disktrétním vstupům PLC %I1.0 - %I1.4. Tato tlačítka nemají blokování kontaktů, to znamená, že logickou 1 na vstupu uvidíme jenom ve chvíli, když tlačítko bude zmáčknuto. Tato tlačítka simulují přivolání výtahu do pater 0-4 zvenčí, např. přivolání výtahu uživatelem z jiných pater.

K simulaci ovládání výtahu uvnitř kabiny bude použit panel HMI. Uživatel výtahu si bude moci vybrat jedno z pěti podlaží. Na panelu HMI bude uživatel také moci pomocí LED diod zjistit, ve kterém patře se výtah nachází. Směr šipek nahoru nebo dolů upozorní uživatele na směr, kterým se výtah pohybuje. Stav fyzických tlačítek bude také přidán na panel HMI, kde bude uveden jejich aktuální stav. Tlačítko STOP na HMI panelu přeruší veškerou paměť výtahu a zastaví uživatele v posledním zvoleném patře. Panel HMI bude mít také funkci přepnutí na servisní rozhraní výtahu, což pomůže s jeho případnou údržbou v případě poruchy nebo při uvádění výtahu do provozu. V sekci servisního rozhraní bude zobrazení aktuální polohy výtahu a jeho aktuální rychlost. bude také možné změnit polohu výtahu nebo změnit rychlost výtahu nastavením její hodnoty z panelu. Na obrázku 4.2 vidíte vývojový diagram modelu výtahu, zelené šipky označují připojení PROFINET.

↑ - PROFINET



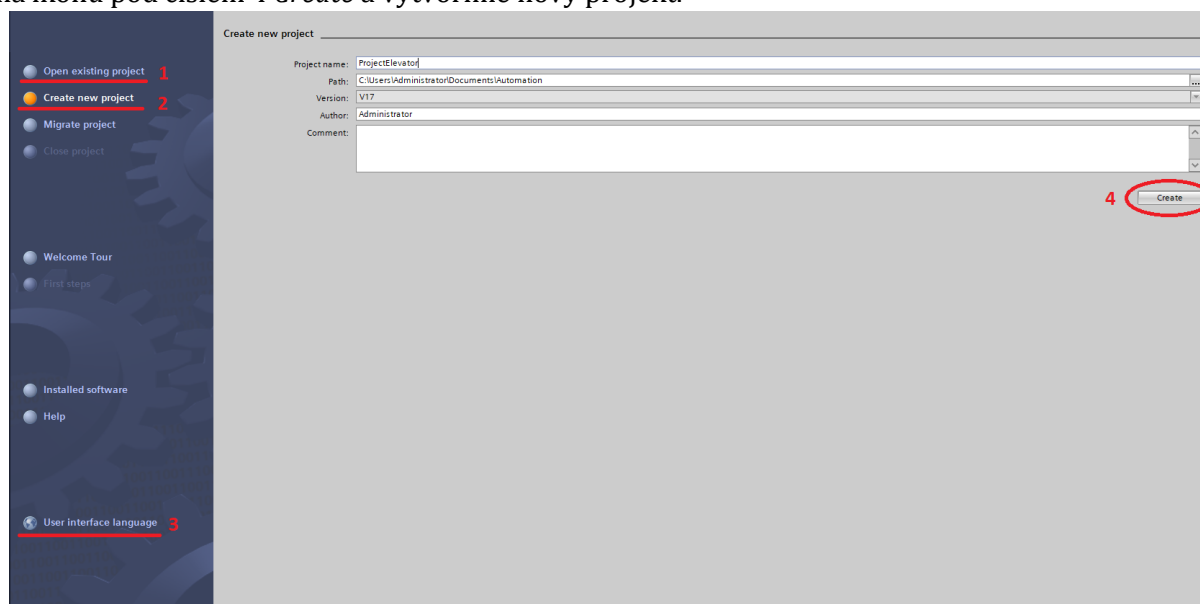
Obrázek 4.2 technologické schéma modelu výtahu

KAPITOLA 5: ZAČÁTEK PRÁCE V TIA PORTAL

Tato kapitola popisuje, jak začít pracovat ve vývojovém prostředí Tia Portal. Bude vysvětlen postup při vytváření nového projektu a popsáno rozhraní programu. Vysvětlíme hlavní okna a způsob zahájení programování v Tia Portal.

5.1 Vytvoření nového projektu

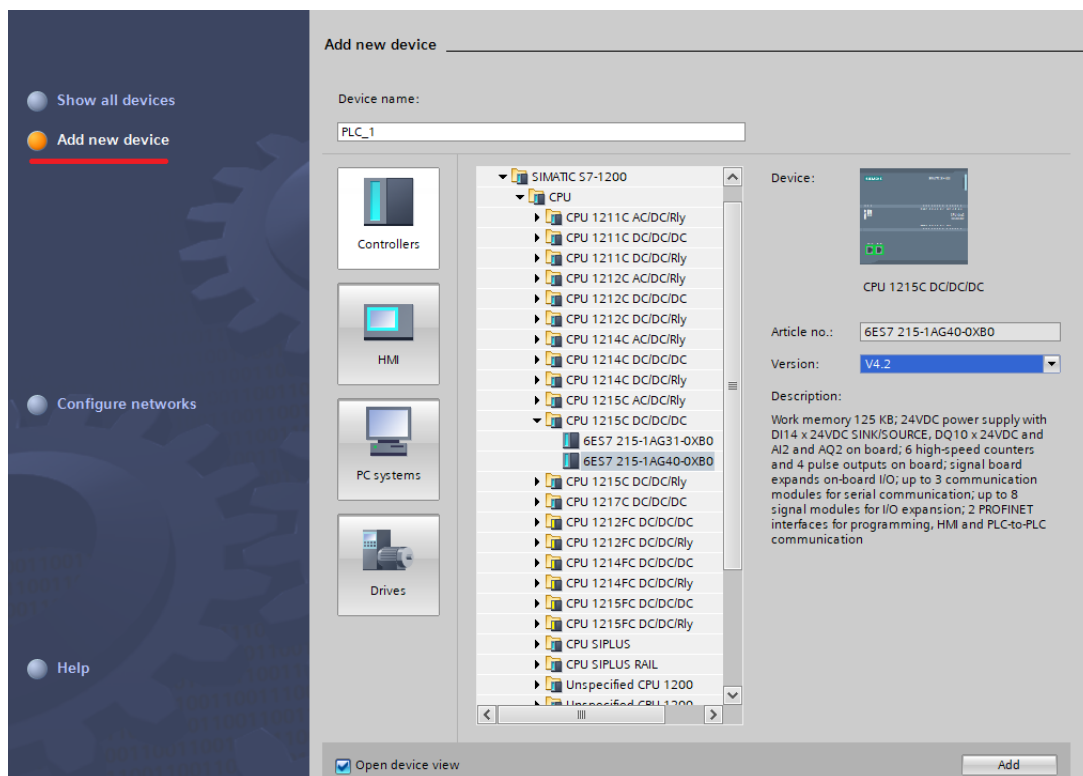
Prvním krokem při vytváření projektu je, stejně jako v každém jiném programovacím prostředí, vytvoření názvu projektu, výběr složky, do které bude náš projekt uložen, a zapsání jména autora. Po otevření programu máte dvě možnosti: otevřít existující projekt nebo vytvořit nový projekt. Na obrázku 5.1 jsou tyto možnosti označeny čísly 1 a 2. Pod číslem 3 se nachází funkce, ve které můžete změnit jazyk (výchozí jazyk je angličtina). Poté do řádku *Project name* musíme napsat název našeho projektu, do řádku *Path* musíme zvolit složku, do které bude náš budoucí projekt uložen, a klikneme na ikonu pod číslem 4 *Create* a vytvoříme nový projekt.



Obrázek 5.1 založení nového projektu

5.2 Přidání PLC do projektu

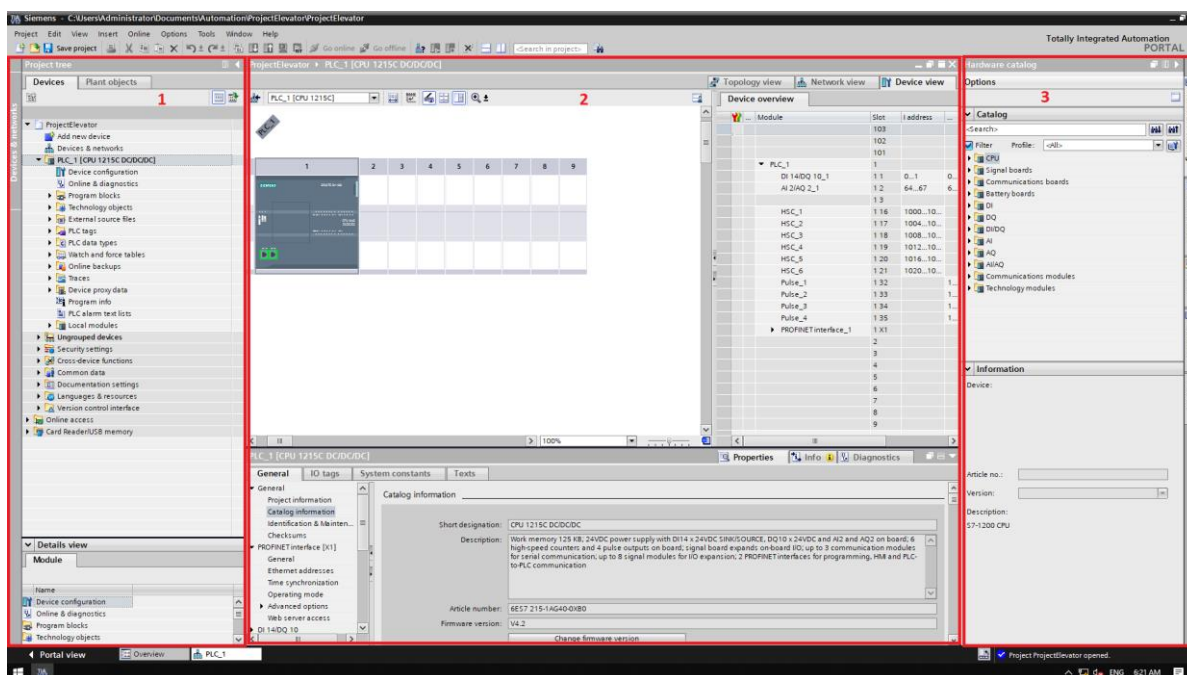
Po vytvoření projektu nás Tia Portal po kliknutí na sekci *Add new device* vyzve k výběru typu PLC, který budeme programovat. Číslo modelu PLC a jeho procesoru je uvedeno na víku samotného PLC, kde je také uvedeno číslo procesoru. Tady musíte otevřít *SIMATIC S7 - 1200 -> CPU -> CPU1215C DC/DC/DC -> 6ES7 215-1AG40-0X80* a vyberte verzi V4.2 a stiskněte tlačítko *Add*.



Obrázek 5.2 Přidání PLC do projektu

5.3 Pracovní okno Tia Portal

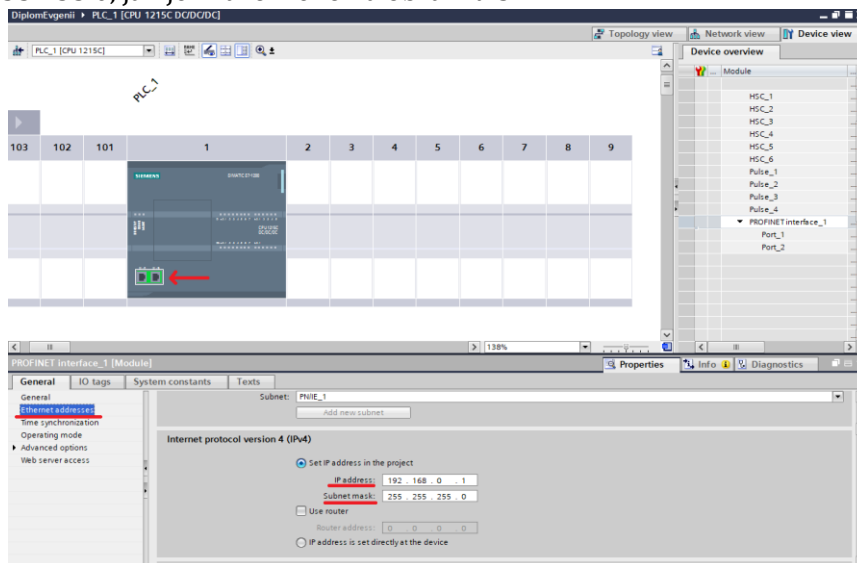
Po přidání PLC do našeho projektu se před námi otevře pracovní okno Tia Portal. Na obrázku 5.3 pod číslem 1 vidíte *Project tree*. Ve *Project tree* si můžete prohlédnout konfiguraci systému, přidat nové prvky a moduly, například panel HMI nebo I/O moduly. Pomocí bloku *Program blocks* můžete vytvořit nový programový blok. Pod číslem 2 se nachází pracovní okno, ve kterém můžete vidět zobrazení vašeho PLC. Kliknutím na ikonu *Properties* můžete zobrazit všechny vlastnosti PLC. Pod číslem 3 je uveden seznam hardwaru, který můžete do projektu přidat.



Obrázek 5.3 Pracovní okno Tia Portal

5.4 Nastavení PLC

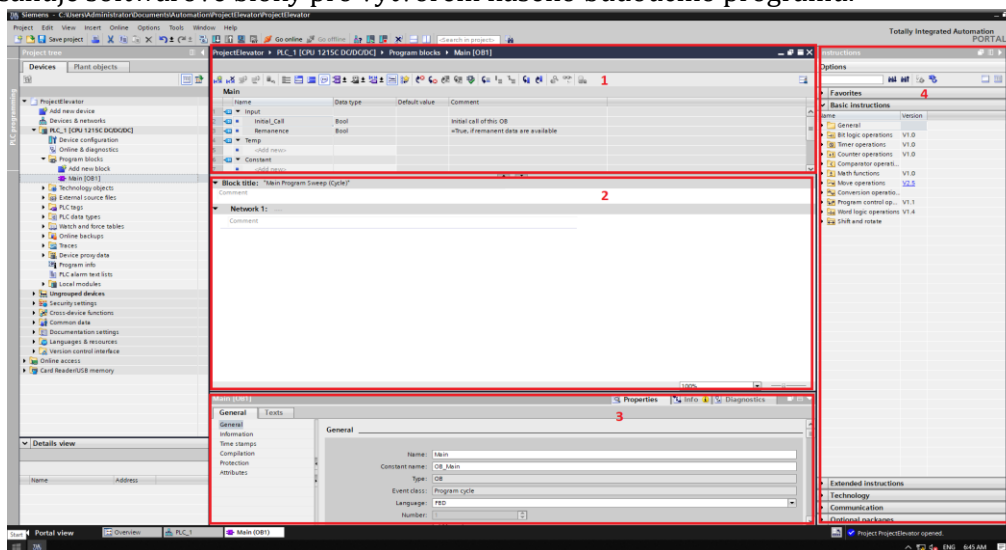
Aby zařízení mohla spolupracovat, musí být mezi nimi nastavena komunikace. V této kapitole se budeme věnovat konfiguraci PLC. Nejprve klikněte na ikonu vstupu *Ethernet* na obrázku PLC a otevřete kartu *Properties* -> *Ethernet addresses*. Poté musíme zapsat IP adresu 192.168.0.1 a masku podsítě 255.255.255.0, jak je znázorněno na obrázku 5.4.



Obrázek 5.4 Nastavení IP adresy a masky PLC

5.5 Programovací okno

Pro začátek programování PLC v Tia Portal, vyberte v okně stromu projektů ikonu *Program blocks* a klikněte na ikonu *Main* (Obrázek. 5.5). Tím se otevře sekce okna programování, kde můžete začít programovat. Nejprve je třeba zvolit programovací jazyk, pro hlavní sekce si můžete vybrat mezi LAD a FBD. Pokud chcete používat programovací jazyk ST je třeba vytvořit nový blok pomocí kliknutí na ikonu *Add new block*. V okně 1 můžete deklarovat proměnné a konstanty, v horní části okna je panel nástrojů, kde můžete spustit monitorování a přiřadit novou pracovní síť. Pod číslem 2 je samotné programovací okno, ve kterém uživatel může vytvořit program. V okně 3 můžete zobrazit vlastnosti bloků programu, výsledky kompilace programu a spustit diagnostiku. Okno 4 obsahuje knihovnu, která obsahuje softwarové bloky pro vytvoření našeho budoucího programu.

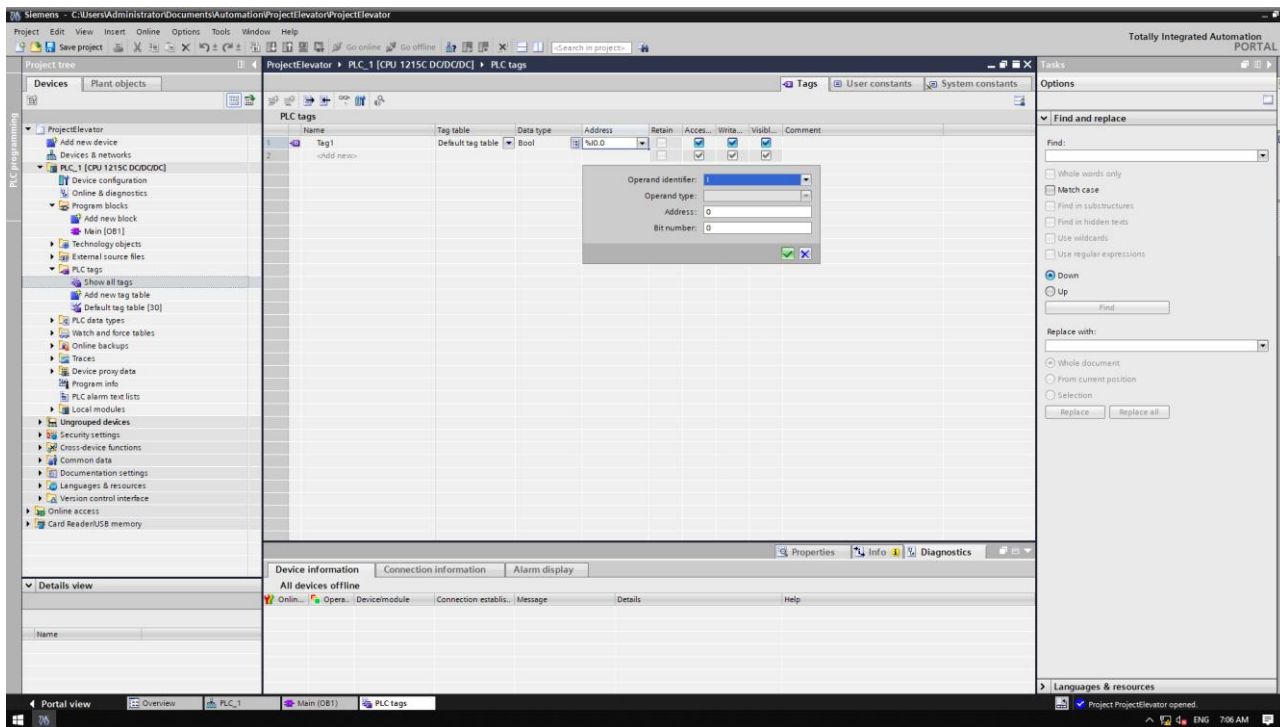


Obrázek 5.5 Programovací okno Tia Portal

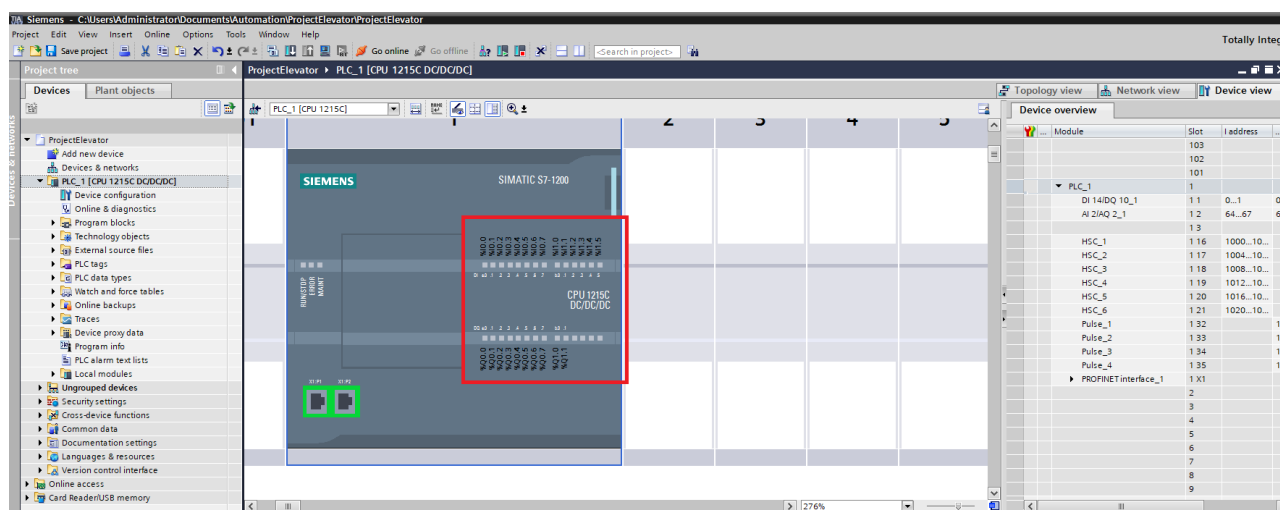
5.6 Vytváření tagů PLC

Kromě deklarace proměnných a konstant budeme při programování potřebovat vytvářet tagy. Otevřením záložky PLC *tags* ve stromu projektu se otevře sekce, kde můžeme přidávat nové tagy

(Obr. 5.6). Tagy nám pomáhají vázat hodnotu proměnné na fyzický vstup nebo výstup PLC. Pro vytvoření tagu je třeba zapsat název tagu do okna *Name*, vybrat datový typ tagu v okně *Data type* a vybrat adresu tagu pod oknem *Address*. V okně *Adresa* vyberte operand v okně *Operand identifier*, číslo bajtu v okně *Adresa* a číslo bitu v okně *číslo bitu*. Existují pouze tři operandy, kde I – *Input*, znamená fyzický vstup PLC, Q – *output*, znamená fyzický výstup PLC. Poslední operand M – *Memory* slouží k zápisu značky do paměti PLC. Zápis tagů do paměti umožňuje vytvářet tagy, které nebudou vázány na fyzické vstupy nebo výstupy PLC. Výsledkem je, že tag má následující podobu: %I1.0, kde I znamená vstup PLC, 1 je číslo bajtu a 0 je číslo bitu. Pro zjištění fyzických adres vstupů a výstupů stačí přiblížit zobrazení PLC pomocí klávesové zkratky Ctrl + kolečko myši (obrázek 5.7).



Obrázek 5.6 Vytváření tagů PLC



Obrázek 5.7 Vstupy a výstupy PLC

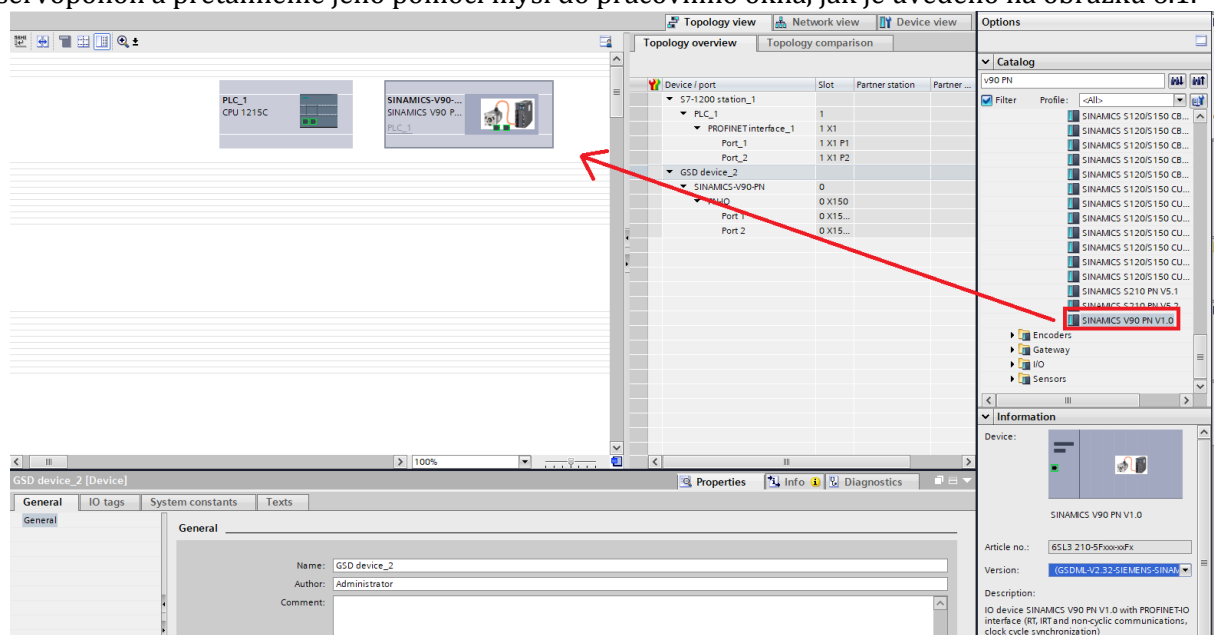
KAPITOLA 6: ŘÍZENÍ MODELU VÝTAHU

V této kapitule rozebereme konfiguraci servopohonu a přidáme servopohon do našeho projektu Tia Portal, vytvoříme technologický objekt a nastavíme jeho konfiguraci. Rozebereme si základní bloky při práci s knihovnou *Motion Control*, jako jsou *MC_Power*, *MC_Home*, *MC_Reset*, *MC_MoveAbsolute*.

6.1 Přidání servopohonu do projektu Tia Portal

Měnič SINAMICS V90 PN a motor SIMOTICS S-1FL6 společně tvoří servopohon, který rovněž patří společnosti Siemens. To znamená, že ho můžeme konfigurovat také v Tia Portal.

Nejprve je třeba přidat servopohon do našeho projektu a nakonfigurovat komunikaci mezi PLC a pohonem. Samotný servopohon lze konfigurovat v aplikaci V-ASSISTANT nebo v samotné aplikaci Tia Portal. Abychom mohli přidat servopohon do našeho projektu, musíme otevřít okno projektu *device&networks* a kliknout na ikonu *Topology view*. Pak můžete pomocí vyhledávacího okna najít náš servopohon zadáním názvu "V90 PN" nebo ho můžete najít ve složce: *Other field devices -> PROFINET IO -> Drivers -> SIEMENS AG -> SINAMICS -> Head module*. V otevřené složce najdeme náš servopohon a přetáhneme jeho pomocí myši do pracovního okna, jak je uvedeno na obrázku 6.1.

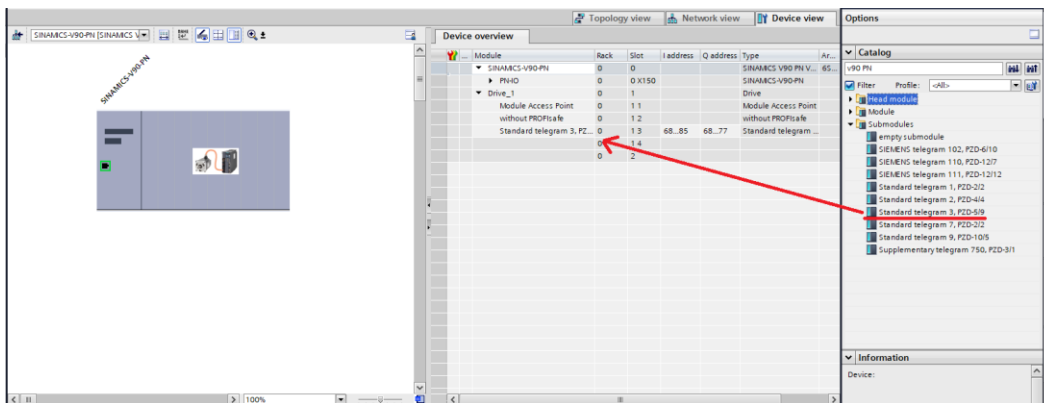


Obrázek 6.1 Přidání servopohonu do projektu

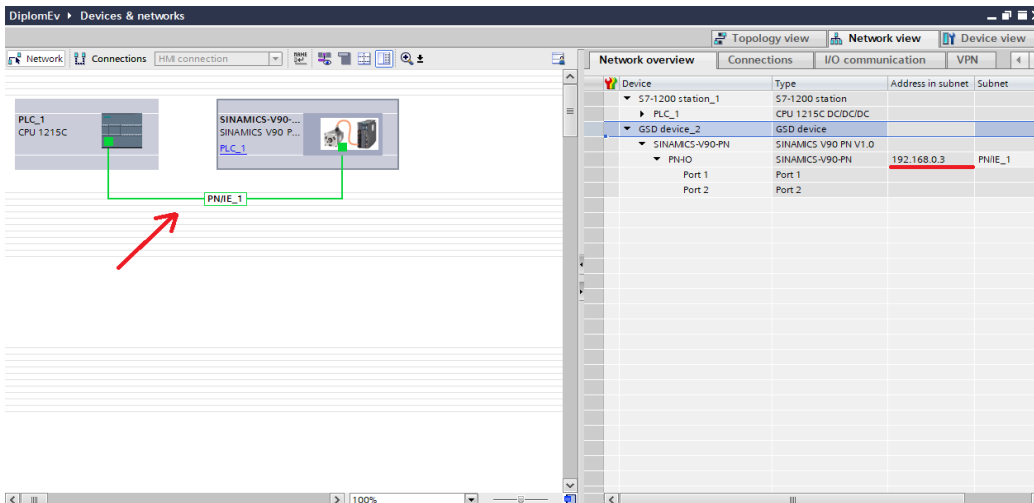
6.2 Konfigurace servopohonu

Aby si PLC a servopohon mohly vzájemně posílat zprávy, je nutné správně nakonfigurovat komunikaci s pohonem a přidat komunikační protokol. Pro nastavení správného komunikačního protokolu klikněte v pracovním okně na náš pohon a pak klikněte na ikonu *Device view*. V pravém okně vyberte položku *Submodules -> Standard Telegram 3* a přetáhněte ho do pravého okna, jak je znázorněno na obrázku 6.2.

Poté přejděte do sekce *Network view*, v tomto okně můžeme přidat IP adresu našeho servopohonu na 192.168.0.3. Teď musíme vybrat ethernetový vstup PLC a pomocí myši jej připojit k ethernetovým vstupům našeho servopohonu, jak je znázorněno na obrázku 6.3.



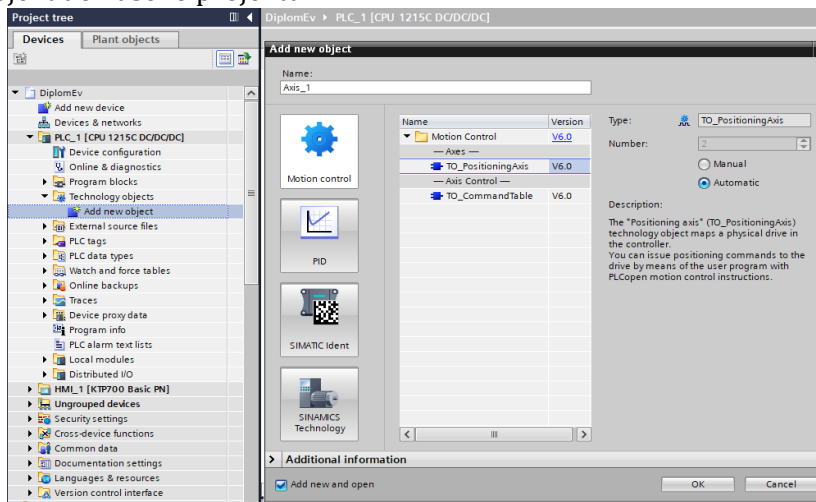
Obrázek 6.2 Přidání telegramu 3 do projektu



Obrázek 6.3 Přidání IP adresy servopohonu

6.3 Přidání a konfigurace technologického objektu

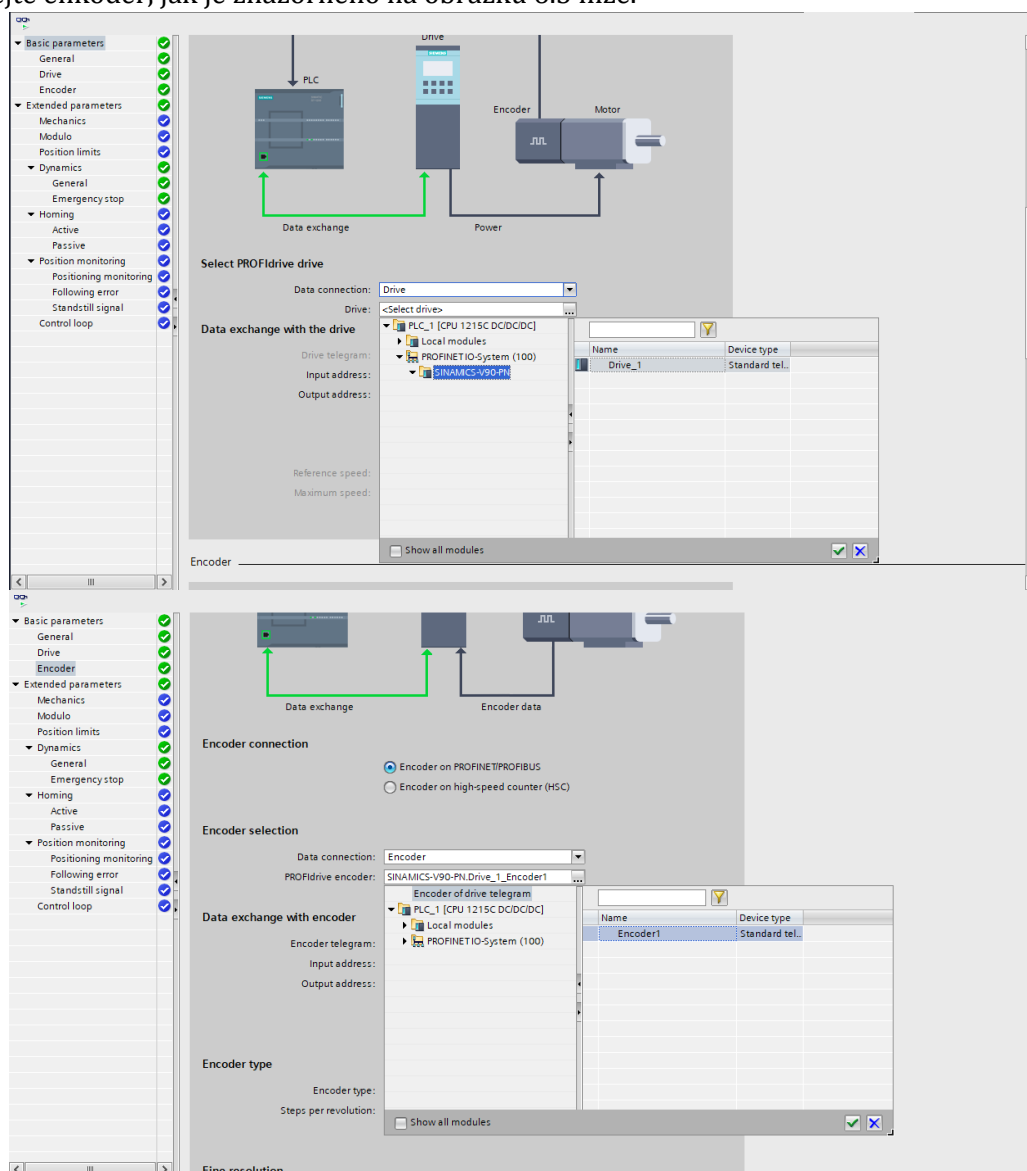
Po vytvoření komunikace mezi pohonem a PLC musíme přidat technologický objekt (*TO – Technology object*). Za tímto účelem zmáčkneme ikonu PLC ve stromu projektu a klikneme na položku *Technology objects -> Add new object*. V otevřeném okně vybereme ikonu *Motion control* a poté vybereme položku *TO_PositioningAxis*. Ted musíme napsat název projektu a kliknout na tlačítko OK, čímž jsme přidali technologický objekt do našeho projektu.



Obrázek 6.4 Přidání TO do projektu

Po přidání technologického objektu je nutné jej správně nakonfigurovat. Za tímto účelem otevřete ve stromu projektu náš technologický objekt a klikněte na tlačítko *Konfigurace* [23]. V okně *Basic parameters* vyberte připojení *PROFIdrive*. V okně *Drive* musíte vybrat náš měnič s protokolem

telegram 3, jak je znázorněno na horním obrázku 6.5. Poté v sekce *Encoder* vyberte komunikační protokol pro enkodér v části *Encoder connection* vyberte možnost *Encoder on PROFINET/PROFIBUS* a přidejte enkodér, jak je znázorněno na obrázku 6.5 níže.

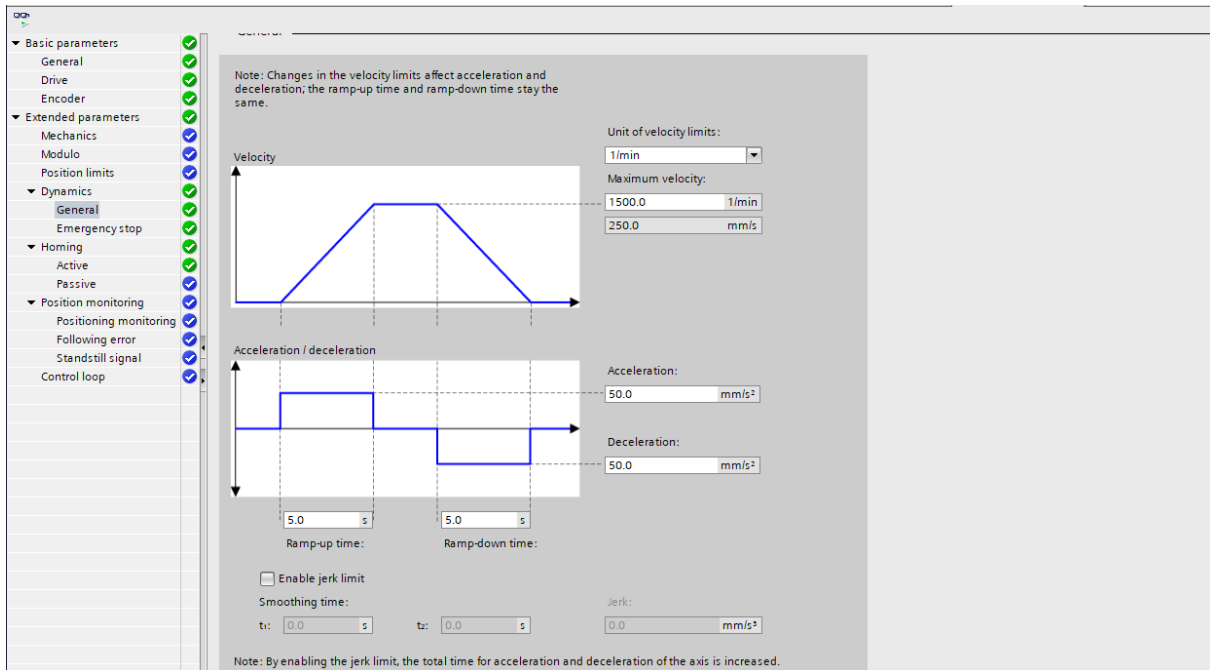


Obrázek 6.5 Přidání pohonu a enkodéru do TO

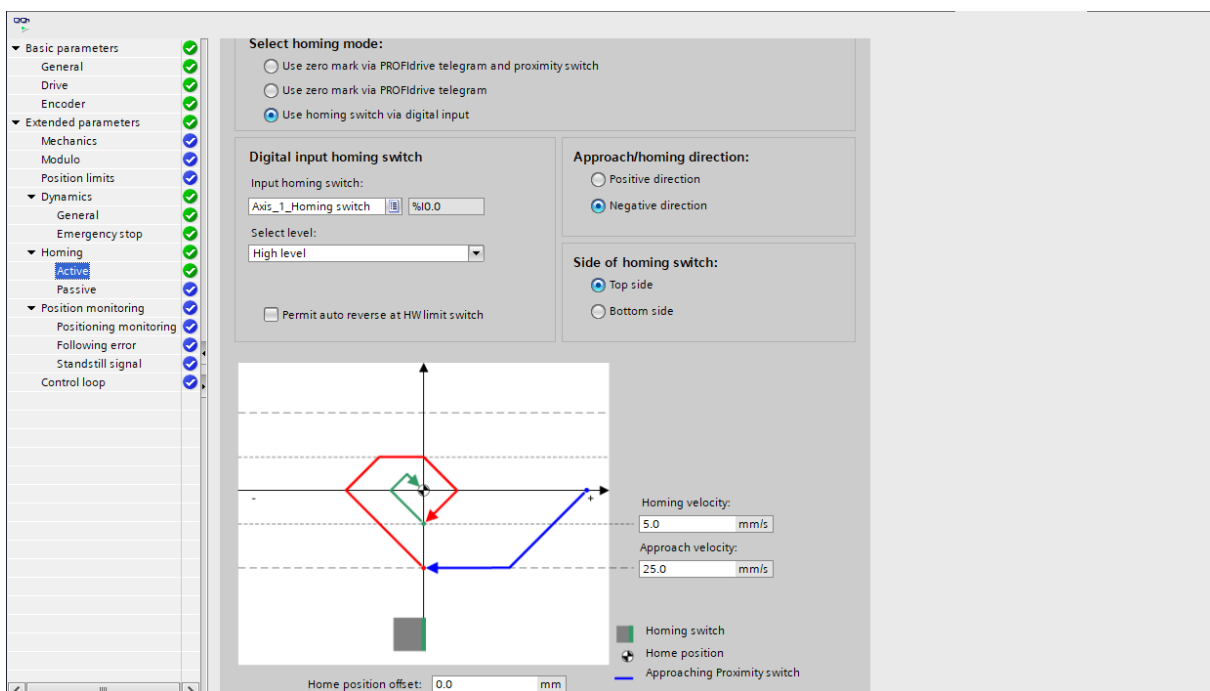
Další položkou nastavení je nastavení dynamiky motoru a nastavení parametru *Active Homing*. Dynamické nastavení motoru vidíte v horní části obrázku 6.6, zde můžete v sekci *Maximum velocity* zvolit maximální rychlost motoru, pro tento projekt bylo zvoleno 1500 1/min. V položkách *Acceleration* a *Deceleration* můžete zvolit možné zrychlení a zpomalení, pro tento projekt bylo v obou případech zvoleno 50 mm/s². V části *Ramp – up time* a *Ramp – down time* můžete nastavit časy rozjezdu a zpomalení při změně nastavené hodnoty otáček.

Dále je třeba nakonfigurovat mechanické vlastnosti v sekce *Mechanics*. V této sekce se nastavuje rychlost zatížení osy, tj. náš zdvih na 1 otáčku motoru. V této sekce vybereme 10 mm, což znamená, že při jednom otočení motoru se náš výtah posune o 10 milimetrů.

V sekci *Homing* -> *Active* vyberte možnost *Use homing switch via digital input*, která nám umožní provádět kalibrace osy pomocí příslušného tagu. K tomu je třeba vytvořit tag s datovým typem *Bool* a pojmenovat ho v sekce *PLC tags*, poté můžeme tento tag přidat v sekce *Input homing switch*. Poté musíme vybrat směry pohybu a stranu naváděcího spínače, vybereme záporný směr pohybu a horní stranu osy. Pro tento projekt byla zvolena rychlost kalibrace 5 mm/s



Obrázek 6.6 Nastavení dynamiky motoru



Obrázek 6.7 Dynamické parametry motoru a Active Homing

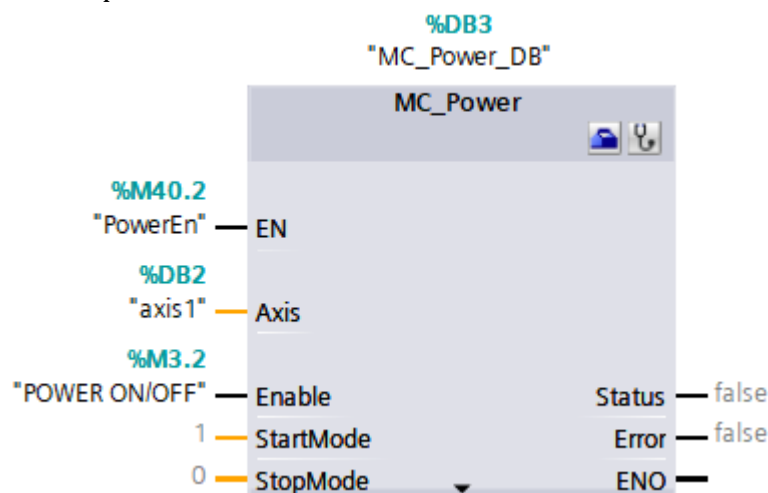
Pro další použití bloku absolutního polohování osy musíme zjistit souřadnice bodů pro každé podlaží. Za tímto účelem musíme přejít do sekce *Commissioning* našeho technologického objektu. Pro spuštění uvedení do provozu stisknete tlačítko *Activate* -> *Enable*, poté pomocí dvou tlačítek *Forward* a *Backward* můžeme ovládat výtah ve obou směrech osy a získat souřadnice pro každé podlaží [23]. Tyto souřadnice pak můžeme použít v blocích *MC_MoveAbsolute*.

6.4 Motion control

Po nastavení technologického objektu můžeme přejít k programování servopohonu. Tia portal má k dispozici knihovnu *Motion control* pro ovládání technologického objektu. Knihovna pro řízení pohybu používá následující funkční bloky jako *MC_Power*, *MC_Home*, *MC_Reset*, *MC_MoveAbsolute*.

6.4.1 MC_Power

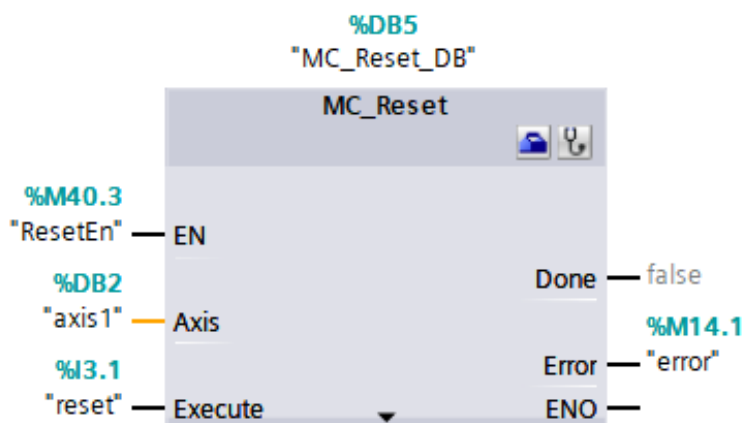
Abychom mohli začít používat funkce *Motion control*, musíme zapnout měnič. Funkce *MC_Power* nám umožňuje zapnout měnič. Nejprve přidejte technologický objekt do sekce Axis bloku *MC_Power*. Pro zapnutí měniče musíme vytvořit dva tagy, například *PowerEn* a *POWER ON/PFF*. Aktivace těchto dvou tagů o hodnotu 1 zapne náš měnič.



Obrázek 6.8 Funkce MC_Home

6.4.2 MC_Reset

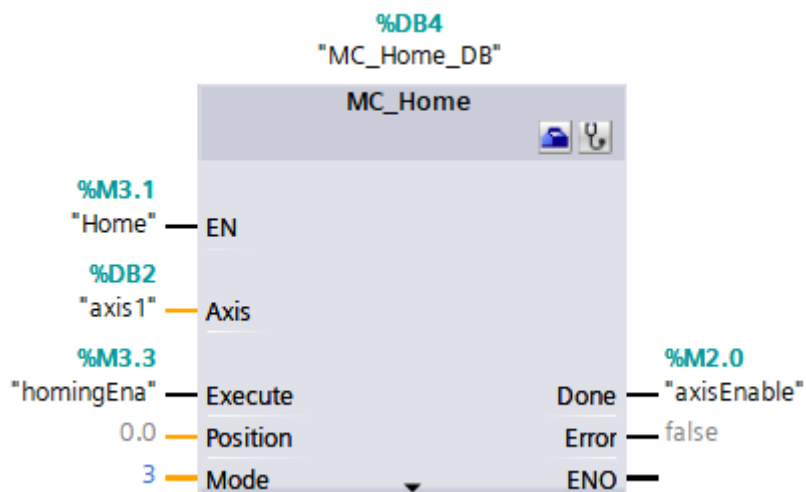
Během spouštění nebo provozu měniče může dojít k chybě v jednom z bloků knihovny *Motion Control*. Pokud dojde k chybě, měnič přestane pracovat a následně přestane pracovat i celý servopohon. Po vymazání chyby mohou jednotky knihovny *Motion control* zůstat v chybovém stavu a nebudou moci pokračovat v provozu. Blok *MC_Reset* pomáhá odmazat chybu v ostatních blocích knihovny. Pro použití tohoto bloku potřebujeme 2 tagy, aby blok mohl fungovat, a také musíme přidat technologický objekt do sekce *Axis*. Poslední tag v sekce *Error* nám pomůže sledovat chybový stav ostatních bloků v knihovně *Motion control*.



Obrázek 6.9 Funkce MC_Reset

6.4.3 MC_Home

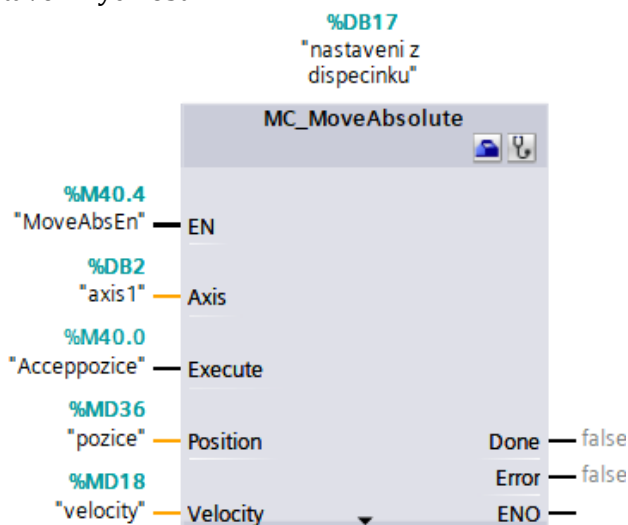
Abychom mohli použít *MC_MoveAbsolute* z kapituly 6.4.4, musíme použít funkce *MC_Home*. *MC_MoveAbsolute* používá absolutní polohování objektu na osy z daného bodu. K nastavení tohoto bodu se používá blok *MC_Home*. Výchozí bod osy je nastaven na 0,0, tj. začátek naší osy, který odpovídá podlaze 0. Stejně jako u ostatních bloků v knihovně *Motion control* budeme muset vytvořit dva tagy, abychom blok aktivovali, a přidat technologický objekt. Do sekce *Done* přidáme také tag *AxisEnable*, pro indikace toho, že zpracování funkce proběhlo úspěšně.



Obrázek 6.10 Funkce MC_Home

6.4.4 MC_MoveAbsolute

Blok *MC_MoveAbsolute* slouží k absolutnímu polohování naší osy vzhledem k počátečnímu bodu, který jsme deklarovali pomocí funkce *MC_Home*. Pět bloků absolutního polohování slouží k přesunu do pater 0-4. Pomocí funkce *Commissioning* jsme zjistili vzdálenost od bodu 0,0, který využívá funkce *MC_home*. Abychom tento blok knihovny aktivovali, musíme v sekce *Axis* přidat technologický objekt. Budeme také potřebovat dva tagy, které tento blok aktivují a povolí. Do sekce *Position* musíme přidat konstantu nebo tag, který запиše souřadnici polohy, do které se má výtah přesunout. Do sekce *Velocity* musíme zapsat rychlost, se kterou musí servopohon příkaz vykonat, nebo přidat tag, který bude odpovědný za nastavení rychlosti.



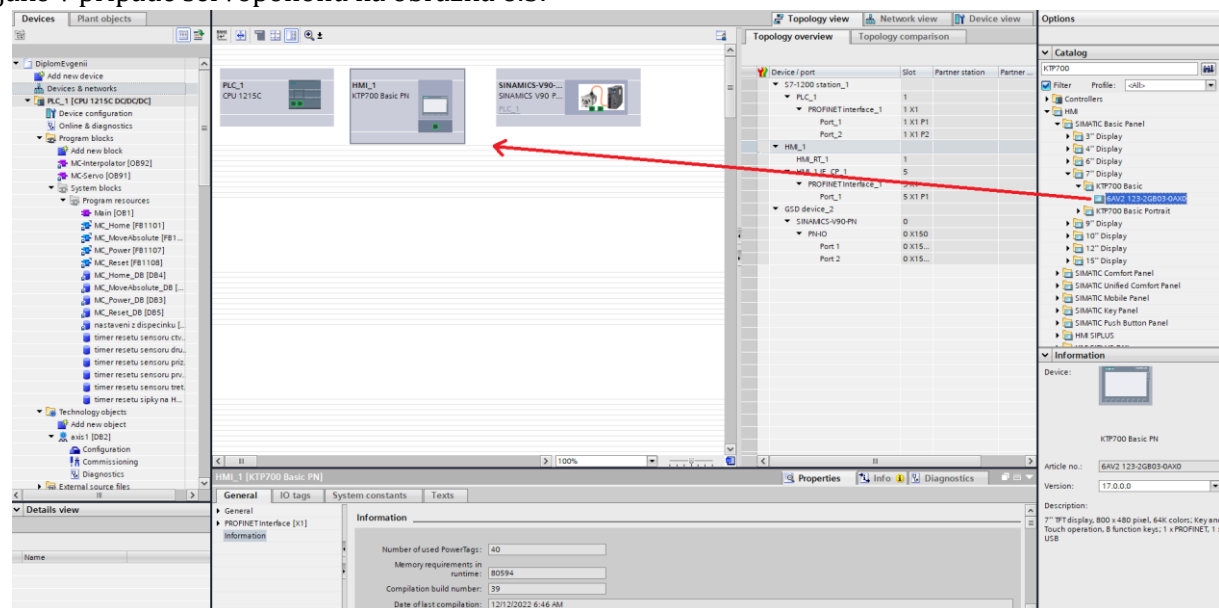
Obrázek 6.11 Funkce MC_MoveAbsolute

KAPITOLA 7: NÁVRH HMI PANELU

Tato kapitola popisuje nastavení komunikace mezi panelem HMI KTP700 Basic a PLC. Bude vysvětlen způsob vytváření vizualizace na panelu, například vytváření tlačítek pro vyvolání výtahu, přenos PLC tagů do HMI tagů, vytváření spínačů, vytváření indikací stavu tlačítek a polohy výtahu.

7.1 Přidání HMI panelu do projektu Tia Portal

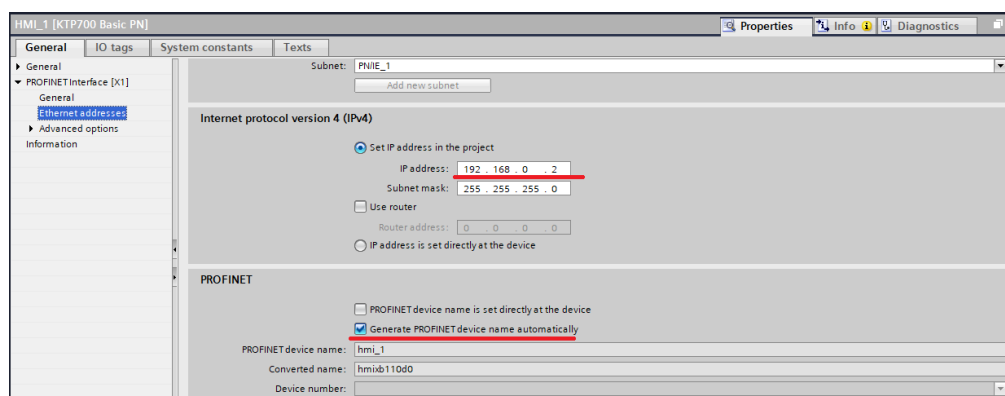
Stejně jako jsme do projektu přidali servopohon, musíme nyní přidat panel HMI. Chcete-li do projektu přidat panel HMI, přejděte do sekce *Devices&networks* našeho projektu, otevřete pravou stranu *Topology view* a do vyhledávacího řádku zadejte KPT700 Basic nebo klikněte na *HMI -> SIMATIC Basic Panel -> 7" displej -> KTP700 Basic*. Po nalezení panelu je třeba jej přesunout do pracovního okna, kde se již nachází PLC a měnič. Poté přejděte do sekce *Network view* a připojte panel HMI k PLC stejně jako v případě servopohonu na obrázku 6.3.



Obrázek 7.1 Přidání HMI panelu

7.2 Konfigurace HMI panelu

Přidaný HMI panel je třeba správně nakonfigurovat, k tomu klikněte na náš panel v okně Tia portal, klikněte na spodní části obrazovky ikonu *Properties*. Pak je třeba v sekce *General* kliknout na sekce *PROFINET Interface [X1] -> Ethernet addresses* a zadejte IP adresu našeho panelu jako 192.168.0.2. Poté klikněte na *Generate PROFINET device name automatically*. Nastavení panelu je vidět na obrázku 7.2.



Obrázek 7.2 Nastavení IP adresy

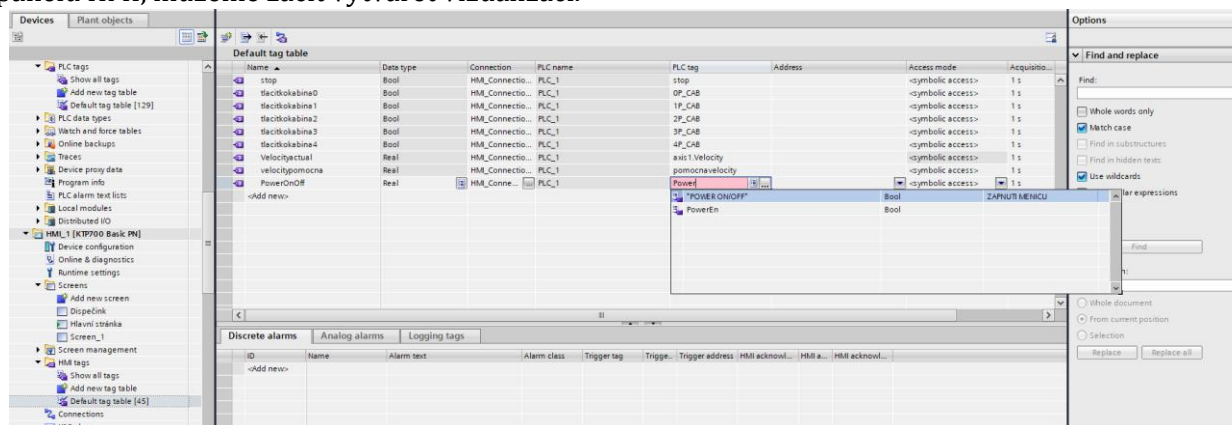
7.3 Vytváření a programování ovládacích obrazovek

Vytváření vizualizací v Tia portal se provádí pomocí ovládacích obrazovek. Můžete si vytvořit několik ovládacích obrazovek pro různé účely, a pak mezi nimi přepínat. Tento projekt používá dvě ovládací

obrazovky. Chcete-li vytvořit ovládací obrazovku, klikněte na *HMI KTP700 Basic -> Screens -> Add new screen* ve stromu projektu, poté se otevře prázdná ovládací obrazovka. Nyní je třeba z knihovny portálu Tia přidat bloky potřebné k vytvoření ovládací obrazovky. Za tímto účelem přetáhněte tyto objekty z pravého okna knihovny na naši ovládací obrazovku

7.3.1 Použití PLC tagů

Abychom mohli začít programovat ovládací obrazovku HMI panelu, musíme přidat tagy, které bude systém ovlivňovat během procesu ovládání. Protože program vytváříme v PLC, chceme ovlivňovat PLC tagy. Za tímto účelem musíme panelu HMI sdělit, které tagy PLC chceme ovlivnit. Za tímto účelem otevřete sekce *HMI tags -> Default tag table* a otevře se před námi tabulka tagů. V této tabulce musíme do řádku *Name* napsat název tagu a do řádku *PLC tag* napsat tag, ke kterému chceme tag panelu HMI přiřadit, jak je znázorněno na obrázku 7.3. Po přidání všech tagů, které chceme ovlivnit pomocí panelu HMI, můžeme začít vytvářet vizualizaci.



Obrázek 7.3 Přidání PLC tagů

7.3.2 Button

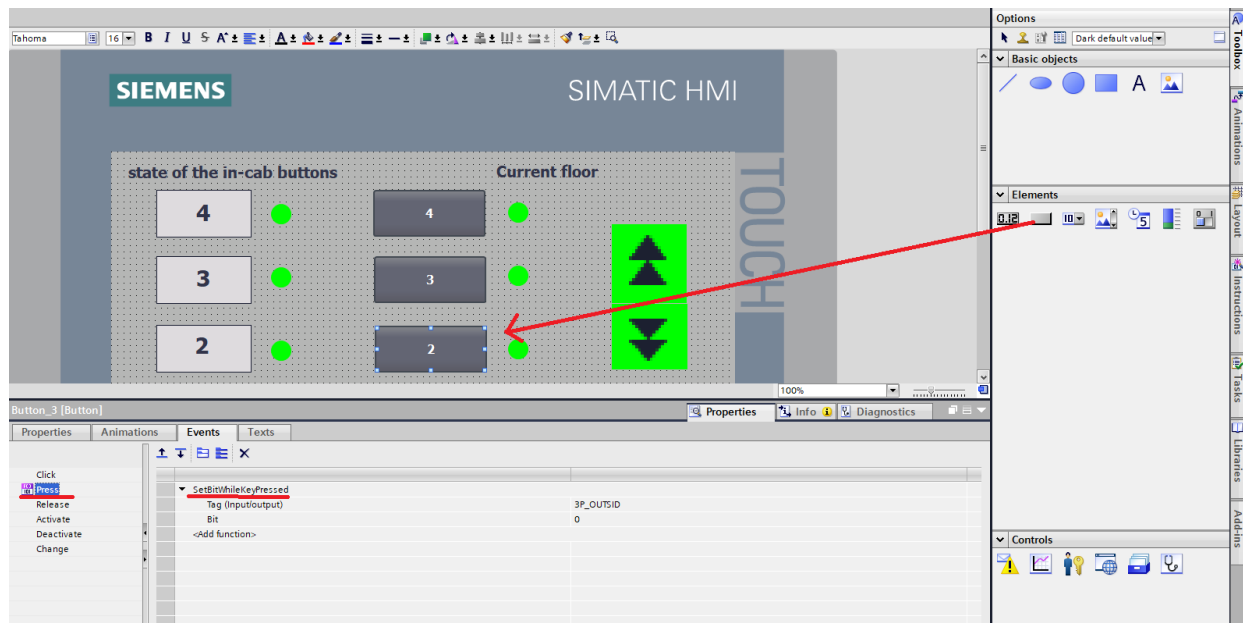
Chcete-li v okně vizualizace vytvořit tlačítko, najdete ve sloupci položek knihovny Tia portal položku s názvem *Button*. Poté by se tlačítko mělo přenést do okna vizualizace.

Poté byste měli přidat značku, na kterou chcete použít efekt pomocí tlačítka. Za tímto účelem klikněte na tlačítko a otevřete *Properties -> Events*. V pravém okně se zobrazí 6 možných efektů tlačítka:

- *Click* – vytvoří puls na tlačítku,
- *Press* – aktivace tagu při náběžné hraně,
- *Release* – aktivace tahu při sestupné hraně,
- *Activate* – zapnutí přístupu k objektu na obrazovce,
- *Deactivate* – vypnutí přístupu k objektu na obrazovce,
- *Change* – změna hodnoty tagu při impulsní akci.

Přejdeme do sekce *Press*, klikneme na otevřeném okně *Add function* a přidáme příkaz *SetBitWhilekeyPressed*. Tento příkaz znamená, že dokud budeme tlačítko držet, bude mít tag připojený k tomuto tlačítku logickou jedničku, jakmile tlačítko pustíme, bude mít tag hodnotu 0. Poté musíme v okně tag (Vstup/výstup) přidat tag, kterému budeme akci posílat. Podobným způsobem můžeme ze všech tlačítek vyvolat výtah do příslušných pater.

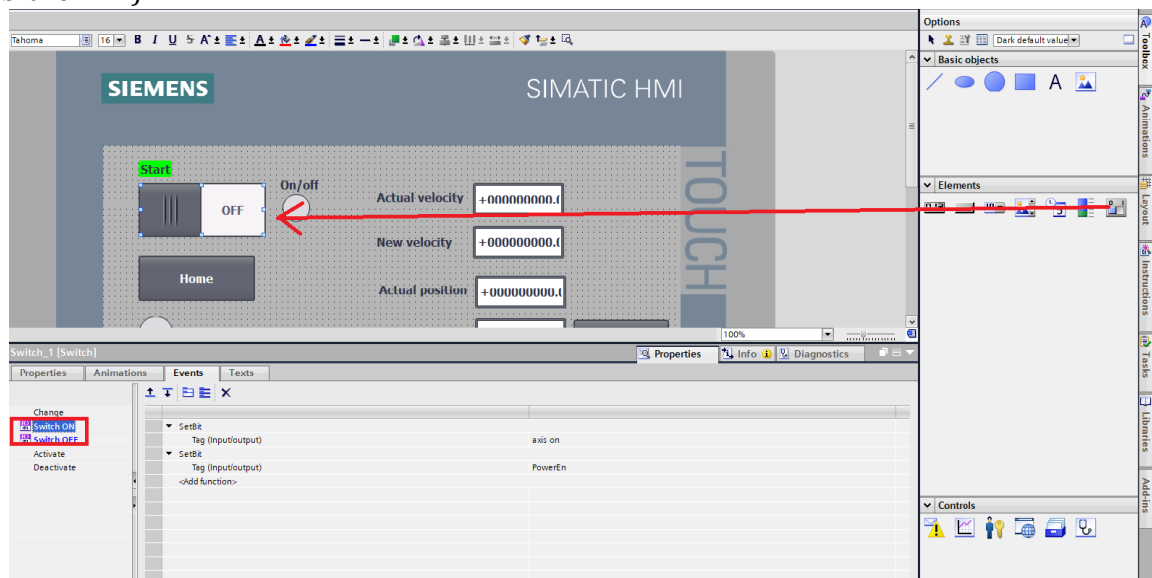
Protože v projektu budou dvě ovládací obrazovky, musíme vytvořit tlačítko, které bude mezi těmito obrazovkami přepínat. Za tímto účelem musíme vytvořit nové tlačítko, přejít do sekce *Events* a přidat funkci *ActivateScreen* v sekci *Click* a poté do této funkce přidat naši druhou obrazovku. Na druhé obrazovce musíme vytvořit stejné tlačítko, které nás přenesse zpět na obrazovku 1.



Obrázek 7.4 Vytváření tlačítka

7.3.3 Switch

Chcete-li vytvořit přepínač, přidejte do panelu vizualizace přepínač z knihovny Tia Portal. Po přidání přepínače na něj klikněte a vyberte *Properties* -> *Events*. V tomto okně vidíme akce *Switch ON* a *Switch OFF*, které provedou určité příkazy, když je spínač v poloze ON a OFF. V tomto projektu je spínač použit pouze jednou, a to k zapnutí a vypnutí měniče. Pro zapnutí měniče vybereme funkci *SetBits* v sekci *SwitchOn* a přidáme dva tagy, které povolí funkční blok *MC_Power*. Chcete-li měnič vypnout, vyberte funkci *ResetBits* v sekci *SwitchOff* a přidejte stejné dva tagy, které nyní měnič vypínají (obrázek 7.4).



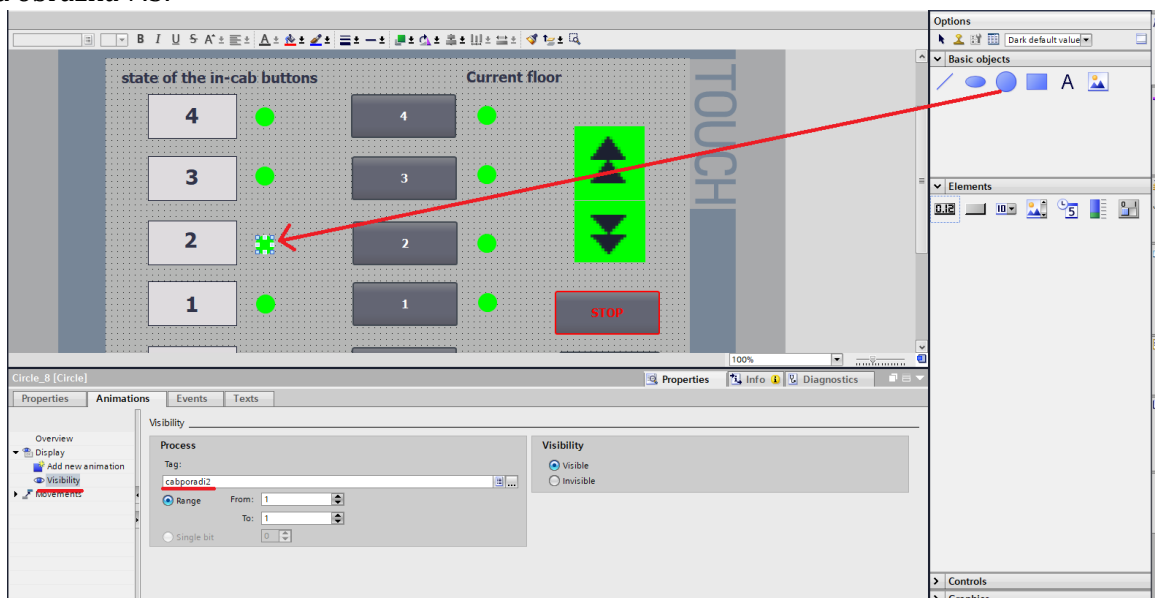
Obrázek 7.5 Vytváření přepínače

7.3.4 Zobrazení objektů

Kromě tlačítek hrají ve vizualizaci důležitou roli animace, které vytvářejí intuitivnější rozhraní. Tia Portal má dvě animační funkce – viditelnost objektu a změnu barvy objektu.

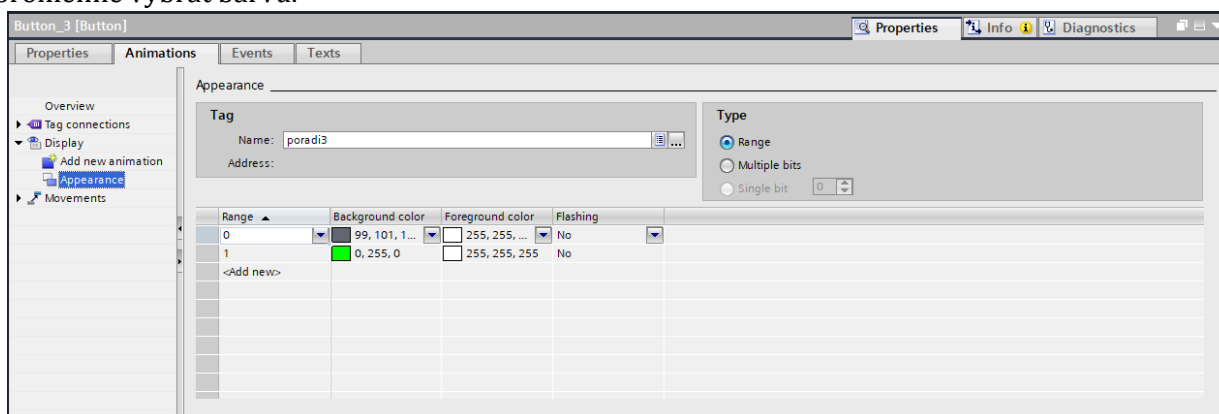
Viditelnost je v projektu použita k zobrazení směru zdvihu a LED diody, které slouží k indikaci polohy výtahu na ose. Pro přidání viditelnosti objektu je nutné tento objekt vytvořit. Za tímto účelem přesuňte požadovaný objekt do okna vizualizace a roztáhněte náš objekt na požadovanou velikost. Poté přejděte do *Properties* -> *Animations* -> *Display* -> *Add new animation* -> *Visibility*. Poté je třeba vybrat požadovaný tag a v sekci *Visibility* vybrat funkce *Visible*, pak je třeba

nastavit hodnotu tagu ve sloupci *Range*, při které se objekt zobrazí na obrazovce. Příklad je uveden na obrázku 7.5.



Obrázek 7.6 Vytváření viditelnosti

Druhá animace se aktivuje ve stejném okně. Vybereme požadovaný objekt, klikneme na sekce *Display* -> *Add new animation* a klikneme na možnost *Appearances* a otevře se okno animace. Poté je třeba přidat tag, v závislosti na jejíž hodnotě se změní barva, a v pravé části okna vybrat možnost *Range*. Poté je třeba ve spodní tabulce vybrat hodnotu proměnné a v závislosti na hodnotě proměnné vybrat barvu.



Obrázek 7.7 Nastavení barev

7.3.5 I/O field

Funkční blok *I/O field* slouží k zadávání a zobrazování analogových signálů. Chcete-li vytvořit pole I/O, přesuňte blok *I/O field* do okna vizualizace podobným způsobem. Poté klikněte na funkční blok a přejděte do sekce *Properties* -> *General* a vyberte tag, jejíž hodnotu chceme sledovat, nebo ji zadejte z panelu v okně *Proces*.

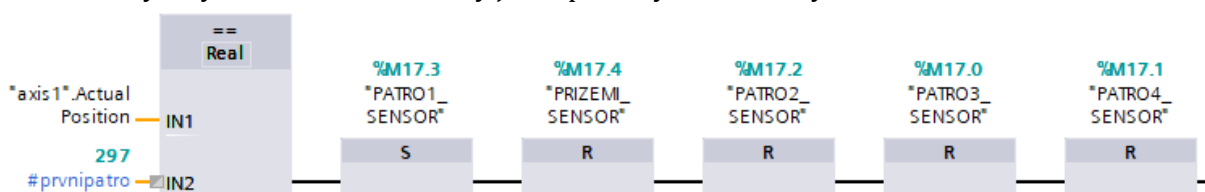
KAPITOLA 8: PROGRAMOVÁNÍ MODELU VÝTAHU

V této kapitule bude popsáno vytvoření programu pro ovládání výtahu pomocí reléové logiky v programu FBD. Budou diskutována softwarová řešení, jako je vytvoření paměťového bloku pro ukládání stisknutých tlačítek na panelu výtahu, programování senzorů pro určení aktuální polohy výtahu, určení směru výtahu, programování tlačítka STOP, vytvoření indikace LED, změna rychlosti a nastavení nové polohy z panelu HMI.

8.1 Snímač polohy výtahu

Abychom mohli nastavit příkazy v bloku *MC_MoveAbsolute* a zjistit, že výtah dorazil do správného patra, musíme vytvořit proměnnou, která nám po aktivaci sdělí aktuální polohu výtahu na ose.

Příklad realizace programu snímače polohy výtahu je na obrázku 8.1. Je nutné vytvořit konstantu, která bude mít hodnotu souřadnice polohy určitého patra na ose. Poté se použije funkce *Compare -> Equal*, která porovná aktuální polohu výtahu na ose a naši konstantu. Tím aktivujeme tag, který si zapamatuje, že se výtah nachází v tomto patře. Současně je nutné resetovat hodnoty tagů pro ostatní senzory, aby nedošlo k situaci, kdy jsou spuštěny dva senzory současně.



Obrázek 8.1 Programování snímačů polohy

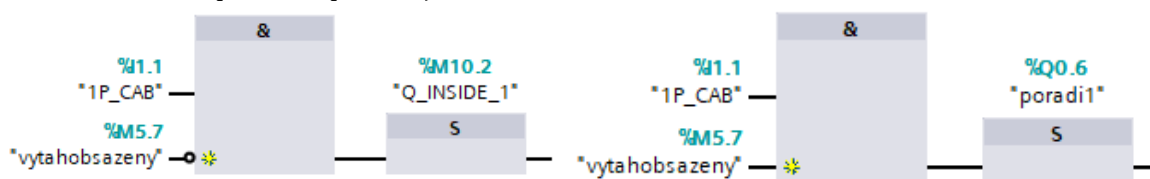
8.2 Řešení paměti

8.2.1 Zpracování signálu z tlačítka

Při stisknutí tlačítka volání do určitého patra z kabiny výtahu nebo zvenčí výtahu mohou být zpracovány dva různé algoritmy.

Obrázek 8.1 vlevo ukazuje situaci, kdy je stisknuto tlačítko výtahu a výtah neprovádí žádné příkazy. Tím se sepne blokováný kontakt, který vyše příkaz bloku *MC_MoveAbsolute*.

Obrázek 8.1 vpravo ukazuje situaci, kdy je výtah v provozu a nemůže právě teď provést příkaz k přesunu do patra. V tomto případě se příkaz k návštěvě určitého patra zapíše do paměťové proměnné, která si příkaz zapamatuje.

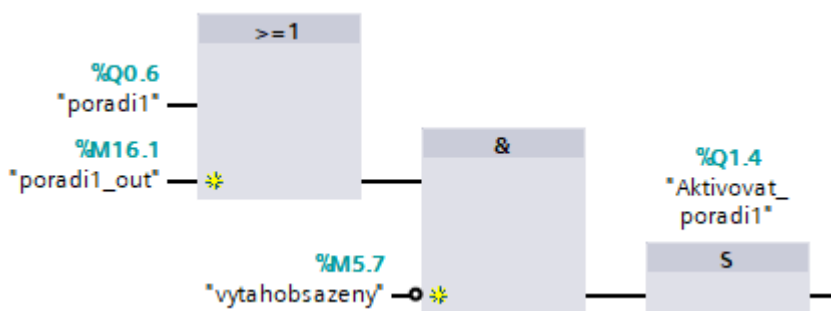


Obrázek 8.2 Logika zmáčknutí tlačítka

8.2.2 Zpracování signálu z paměti

Když nastane situace zobrazená na obrázku 8.2 vpravo, zapíše se do paměťové proměnné další patro. Poté potřebujeme příkaz, který tento příkaz dostane z paměti v okamžiku, kdy výtah bude volný.

Situace je znázorněna na obrázku 8.3. Pokud je paměťová proměnná pro určité patro nastavena na logickou 1 a výtah je volný, nastaví se aktivační bit příkazu pro proměnnou paměti z fronty. Příkaz k aktivaci paměťové proměnné vyše impuls do bloku *MC_MoveAbsolute* našeho měniče a výtah se podle příkazu pohne. V tomto projektu se aktivace paměťové proměnné provádí od nejnižšího patra k nejvyššímu, tj. pokud je ve frontě například 0 a 2 patra, výtah pojedje do 0.

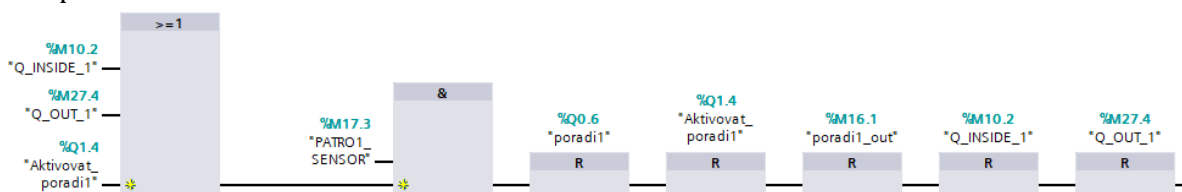


Obrázek 8.3 Aktivace příkazů z paměti

8.2.3 Resetování příkazů

Při programování kontaktů s blokováním zůstanou tyto kontakty i po provedení příkazů blokovány a budou mít hodnotu 1. Tyto tagy je třeba po provedení příkazu vynulovat, aby jednotka *MC_MoveAbsolute* mohla provést další příkaz z fronty.

Projekt implementuje resetování všech proměnných spojených s určitým patrem, když výtah do tohoto patra přijede. To je realizováno stejně jako ve skutečném výtahu, protože když se nacházíte v určitém patře, nemůžete znovu stisknout tlačítko stejného patra, protože se okamžitě resetuje, příklad realizace je uveden na obrázku 8.3. Logika této části programu je následující: pokud se výtah nachází na patře a je aktivována jakákoli paměťová funkce stisknutím tlačítka, ať už jde o tlačítko z PLC nebo tlačítko z panelu HMI, všechny paměťové funkce se vynulují, protože výtah se již nachází na tomto patře.



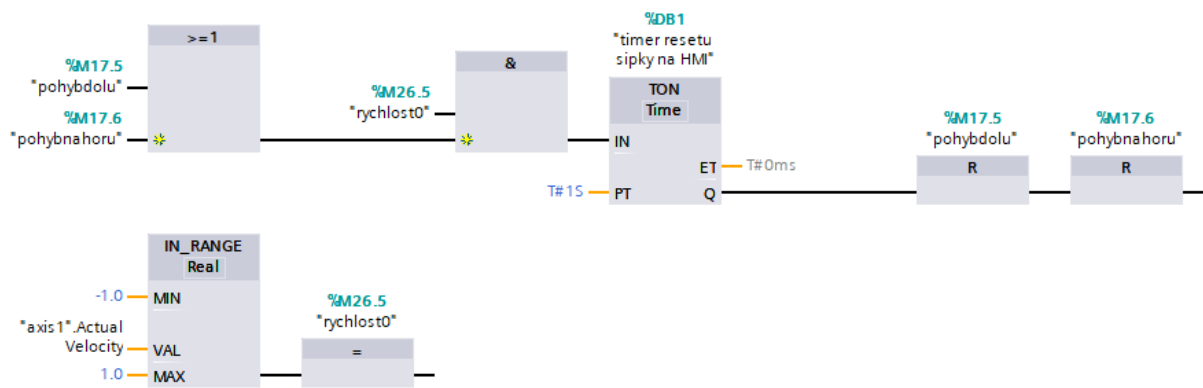
Obrázek 8.4 Resetování příkazů

8.3 Určení směru jízdy výtahu

V kapitole 7 jsme vytvořili rozhraní šipky pohybu výtahu. Technologický objekt, který jsme přidali do projektu nemá parametr směru pohybu objektu na ose, proto musíme vytvořit logiku, která nám určí směr pohybu výtahu.

Ke stanovení pohybu byla použita metoda hrubé síly [25]. Podle této metody je možné na základě identifikace všech možných variant pohybu z určitých podlaží určit směr výtahu.

Přidali jsme také funkci pro resetování šipky, pokud výtah nikam nejede a stojí na místě (Obrázek 8.5). V tomto algoritmu byl použit časovač, který odpočítává 1 sekundu od okamžiku, kdy je detekována nulová rychlost. Blok *IN_Range*, který pomáhá identifikovat rychlost výtahu s odchylkou -1 až 1. Po nastavení nulové rychlosti se spustí jednosekundové odpočítávání a proměnné odpovíděné za zobrazení šipek směru zdvihu se vynulují.



Obrázek 8.5 Resetování šipky směru jízdy

8.4 Tlačítko STOP

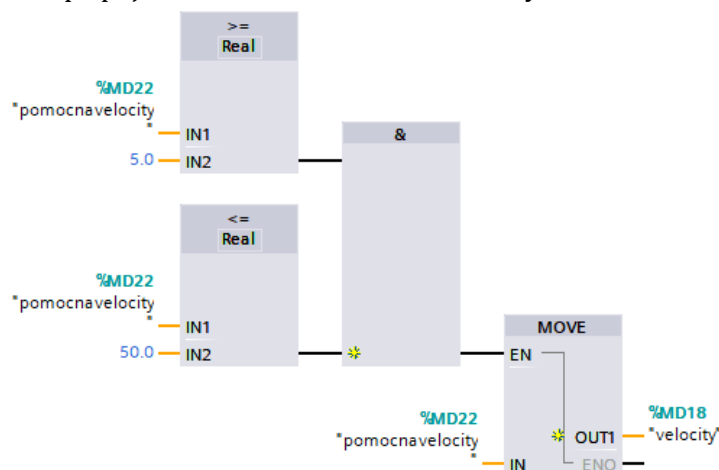
Tlačítko STOP slouží k nouzovému zastavení výtahu z panelu. Pro jeho realizaci můžeme použít blok *MC_Halt*, který zastaví kabinu výtahu. Toto řešení pro nás není vhodné vzhledem k tomu, že všechny paměťové proměnné v programu jsme nastavili na logickou jedničku, dokud výtah nedojede do určitého patra. To znamená, že blok *MC_Halt* nebude fungovat, protože hned bude přerušen blokem *MC_MoveAbsolute*. Také blok *MC_Halt* by zastavil výtah ihned po obdržení příkazu STOP, což by znamenalo možné zastavení výtahu mezi patry a vznik nebezpečné situace.

Vzhledem k výše uvedeným skutečnostem bylo tlačítko STOP implementováno tak, aby resetovalo všechny paměťové tagy a umožnilo výtahu ukončit poslední příkaz, který právě prováděl.

8.5 Změna rychlosti z dispečinku

V kapitole 7 jsme přidali bloky *I/O field* pro sledování aktuální rychlosti výtahu a možnost zadat nový parametr rychlosti.

Abychom mohli implementovat zadání rychlosti, musíme vytvořit dvě proměnné. První proměnná bude mít hodnotu, kterou zadáme do servisního rozhraní HMI panelu. Při tom rychlost bude mít omezení, jak je znázorněno na obrázku 8.6. Jediné omezení rychlosti zdvihu pro nás upravuje ČSN 73 0802 a ČSN 73 0804 [9]. Podle těchto norem musí být minimální rychlost výtahu taková, aby se kabina mohla přemístit z dolního do horního patra za 2,5 minuty. Maximální rychlost výtahu byla stanovena na základě běžné rychlosti výtahu v budovách, která je 0,6 m/s. Druhá proměnná bude pomocí bloku *MOVE* předávat hodnotu zadané rychlosti proměnné *velocity*. Proměnná *velocity* bude připojena ke všem blokům z knihovny *Motion control*.



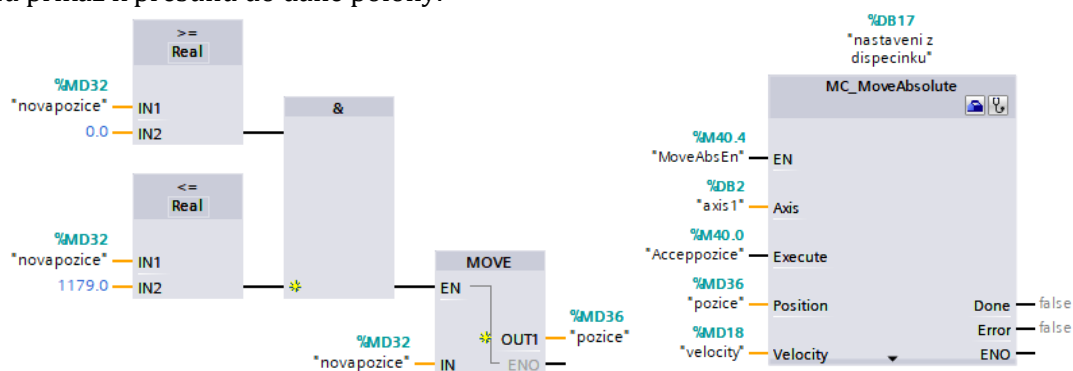
Obrázek 8.6 Nastavení rychlosti z panelu

8.6 Změna polohy výtahu

Aby bylo možné nastavit polohu výtahu ze servisního rozhraní, byl použit algoritmus na obrázku 8.7. Skutečnou polohu výtahu lze sledovat v okně *Actual position* na obrázku 7.8. Novou polohu výtahu

zaznamenáte pomocí servisního rozhraní v okně *New position*. Za tímto účelem klikněte na ikonu *New position* a zadejte hodnotu pozice od 0 do 1179.

Stejným způsobem, jakým jsme zapsali novou rychlost pomocí bloku *MOVE*, se nová pozice zapíše do proměnné pozice. Příslušná poloha se pak zapíše do bloku *MC_MoveAbsolute*, který odešle výtahu příkaz k přesunu do dané polohy.

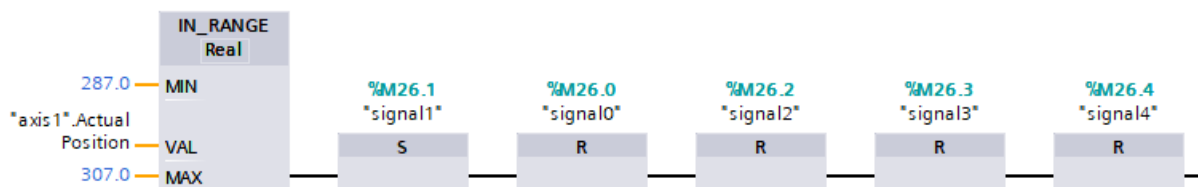


Obrázek 8.7 Změna polohy výtahu

8.7 LED indikace

Pomocí LED indikace uživatel v kabině výtahu vidí, ve kterém patře se výtah právě nachází. Zelené LED diody jsou zobrazeny na obrázku 7.7 v kapitole 7.

Indikátory LED v rozhraní HMI je třeba odpovídajícím způsobem naprogramovat. Bohužel TLL enkodér, který využívá servopohon v našem projektu, není vhodný pro přesné operace [18]. Z tohoto důvodu nejsou snímače polohy pro tento problém vhodné, protože jejich tagy fixují polohu pouze tehdy, když se výtah zcela zastaví. K vyřešení tohoto problému byl použit funkční blok *IN_RANGE*, který detekuje polohu výtahu s chybou 10 mm a pak aktivuje odpovídající tag. Stejně jako u snímačů polohy v kapitole 8.1 musíme vynulovat všechny ostatní značky LED, aby se nespustilo několik LED diod najednou.

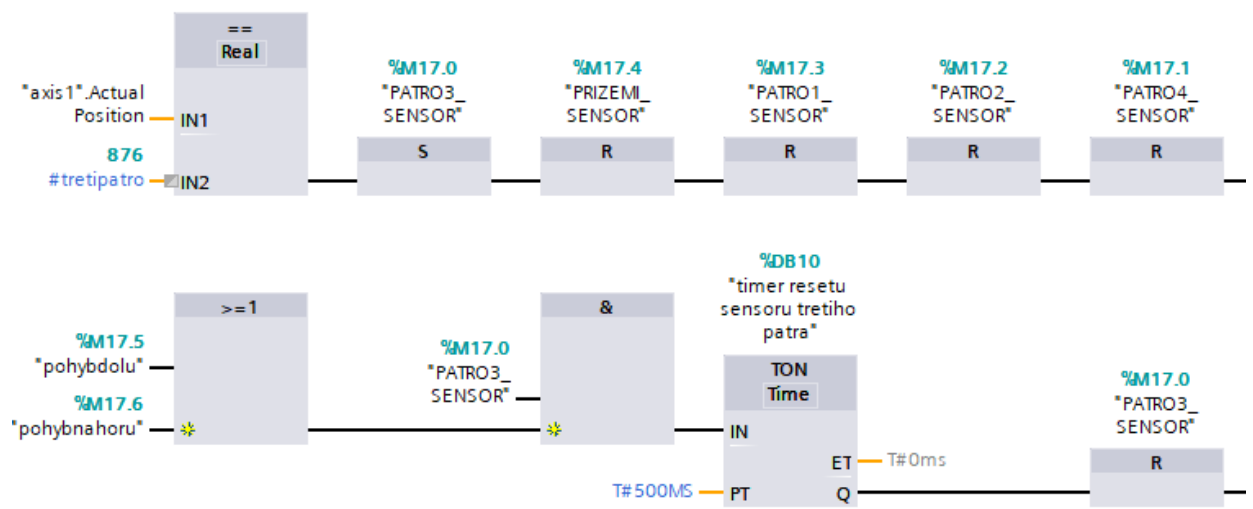


Obrázek 8.8 Zpracování polohy pro LED indikace

8.8 Resetování snímače polohy výtahu

Když je snímač polohy výtahu aktivován do určité polohy, aktivuje se odpovídající tag. Tento tag bude mít hodnotu logické 1, dokud výtah nedojede do jiného patra a všechna ostatní možná patra výtahu se vynulují. Jinými slovy, může se stát, že se uživatel nebude moci vrátit do předchozího patra, dokud bude tento senzor aktivní. K tomu je třeba vynulovat příslušný snímač pro detekci podlahy.

V tomto projektu je nulování tohoto snímače realizováno pomocí časovače, který odpočítává 1 sekundu od okamžiku, kdy se výtah pohne, a deaktivuje příslušnou značku. Uživatel výtahu tak bude moci 1 sekundu po zahájení pohybu výtahu stisknout tlačítko podlaží, ze které přijel. Příklad realizace je vidět na obrázku 8.9.



Obrázek 8.9 Vynulování snímače pro opětovné stisknutí tlačítka

KAPITOLA 9: SPUŠTĚNÍ CELÉ APLIKACE

Po vysvětlení, jak fungují některé podprogramy, bychom rádi vysvětlili, jak funguje celý program. V této kapitole budou vysvětleny 3 algoritmy pro spuštění výtahu a zahájení jeho provozu.

9.1 Startovací program

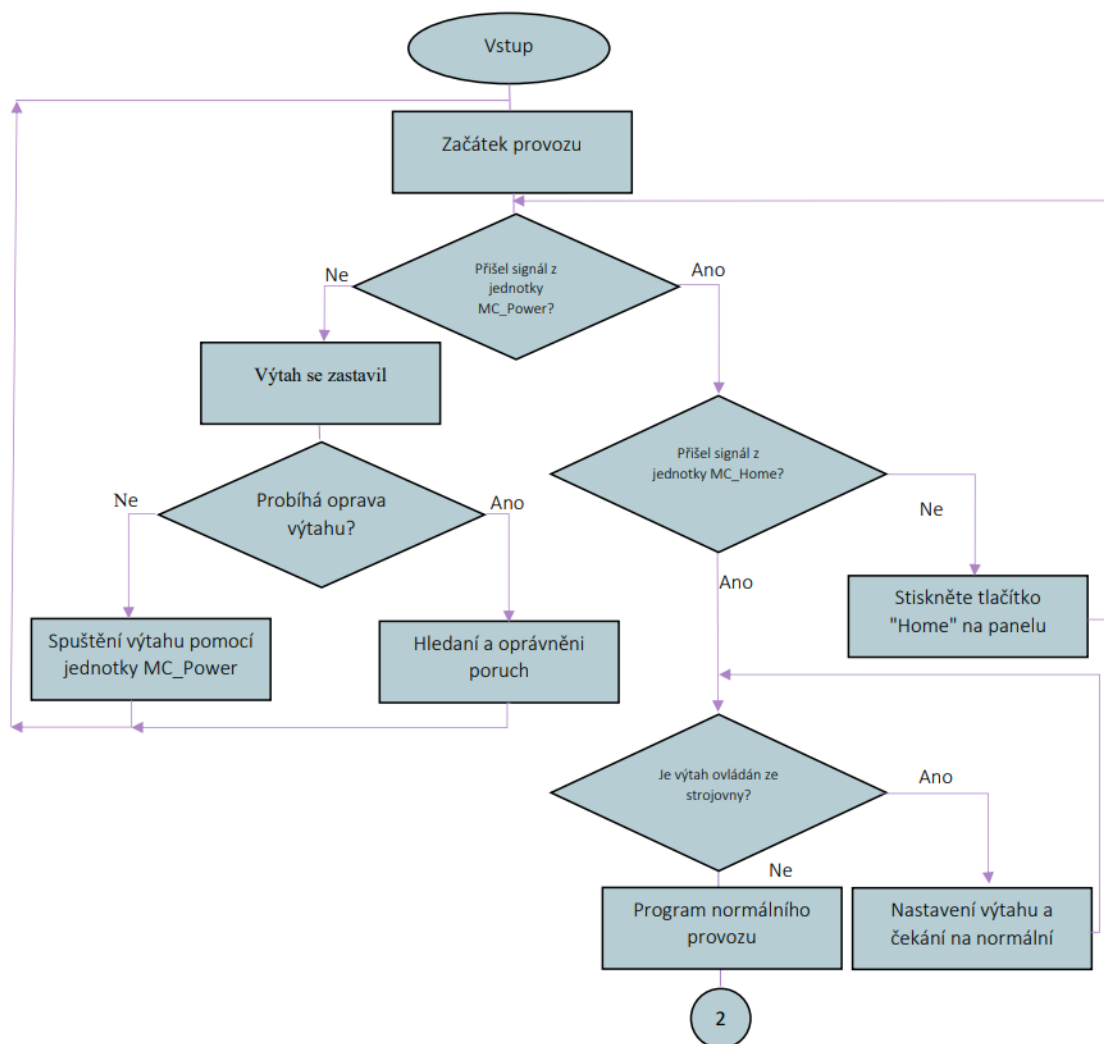
První algoritmus pomáhá určit, zda je výtah připraven k normálnímu provozu. Tato část programu je kompletně řízena ze strojovny, tj. z panelu HMI, pomocí kterého můžeme výtah spustit k dalšímu provozu.

Nejprve musíme zajistit, aby byl výtah spuštěn použitím logické jednotky na blok *MC_Power*. Pokud není žádný signál, výtah se dále nerozjede, dokud nebude vyslán signál do bloku *MC_Power*, který zapne měnič. Pokud je výtah vypnutý z důvodu poruchy, počkáme, dokud nebude opraven a zapnut. Pokud výtah nevyžaduje opravu, dalším krokem je spuštění výtahu stisknutím tlačítka *Switch* na panelu HMI.

Dále se musíme ujistit, že do jednotky *MC_Home* dorazil signál. Pokud není k dispozici žádný signál, musíme stisknout tlačítko *Home* na HMI panelu. Pokud se objeví signál na jednotce *MC_Home*, přejdeme k dalšímu kroku.

V dalším kroku zkontrolujeme, zda je výtah ovládán ze strojovny pomocí panelu HMI. Pokud je výtah ovládán z panelu HMI za účelem nastavení nové rychlosti nebo polohy, jak je popsáno v odstavcích 8.5 a 8.6 kapitoly 8, program počká, dokud nebude nastavení ze strojovny dokončeno.

Po všech těchto krocích můžeme přejít do sekce normálního provozu (podprogram 2). Algoritmus tohoto podprogramu vidíte na obrázku 9.1.



Obrázek 9.1 Algoritmus startovacího programu.

9.2 Normální režim provozu

Po absolvování prvního programového cyklu z kroku 9.1 můžeme přejít k dalšímu kroku, a to k normálnímu provoznímu režimu.

Normální provozní režim začíná čekáním na první příkaz z panelu HMI nebo tlačítek připojených k PLC.

Po příchodu prvního příkazu se zjistí aktuální poloha výtahu. Pokud je výtah ve správném patře, vyšle se signál do příslušné jednotky *MC_MoveAbsolute*. Protože výtah je již ve správném patře, ale jednotka *MC_MoveAbsolute* neudělá nic jiného, než že aktivuje příslušnou proměnnou odpovědnou za indikaci patra (sekce 8.1 kapitoly 8). Program zaregistruje, že příkaz již byl proveden, a vynuluje všechny příkazy tlačítek, která mají zablokované kontakty, a vynuluje všechny hodnoty paměťových proměnných pro dané patro.

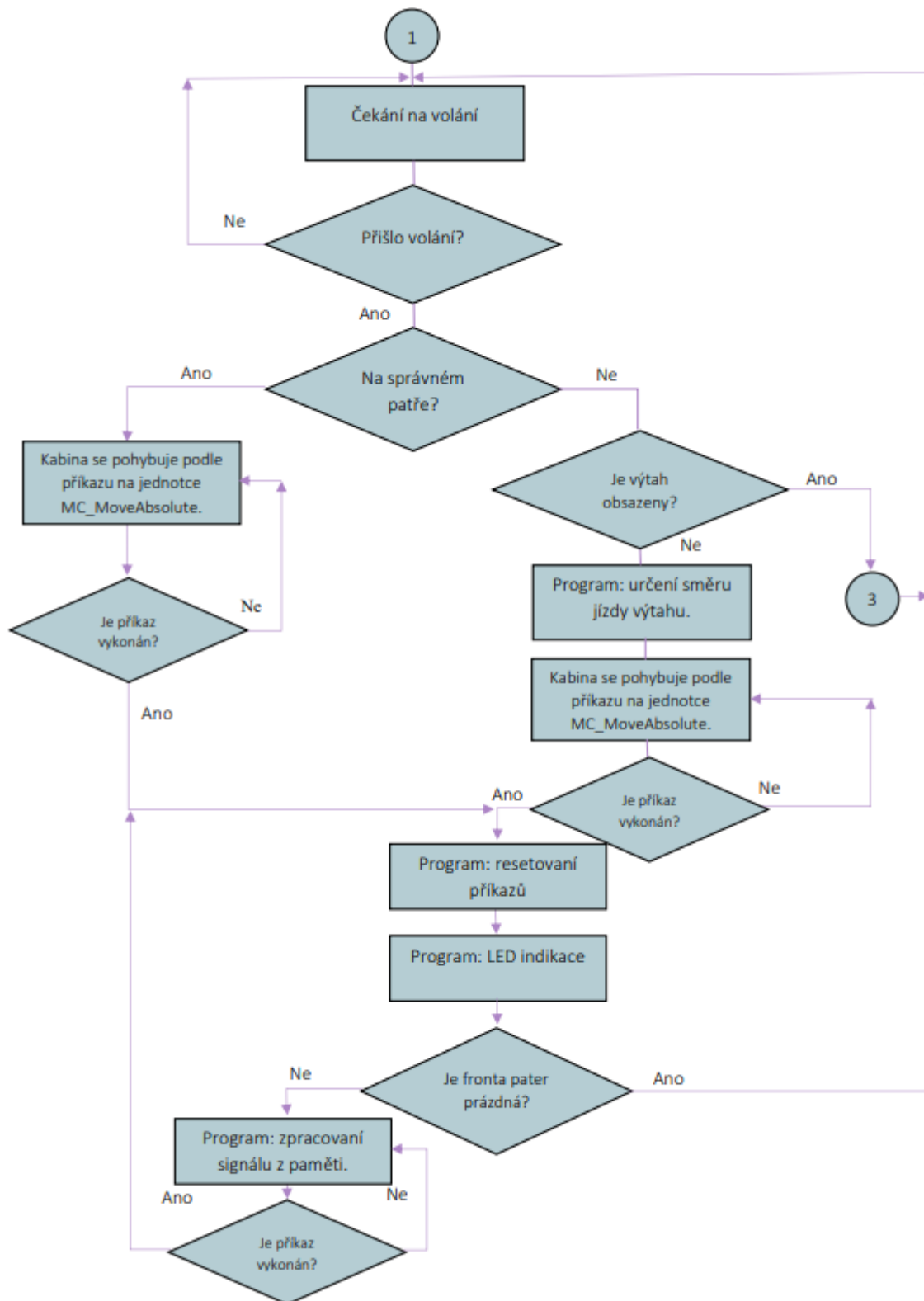
Pokud se výtah nenachází v patře, ze kterého byl příkaz vydán, aktivuje se blok pro kontrolu, zda je výtah obsazen. Vlastní kontrola, zda je výtah obsazen, byla popsána v sekci 8.2.1 kapitoly 8. Pokud je výtah obsazen, přistoupíme k zápisu signálu z tlačítka do paměťové proměnné, tento krok je popsán v sekce 9.3. Pokud je výtah volný, je třeba nejprve aktivovat směrovou šipku kabiny výtahu. Program pro vytvoření této šipky byl popsán v části 8.3, kapitola 8.

Kabina se pak začne pohybovat podle příkazu, který jsme aktivovali stisknutím tlačítka. V sekci *Execute* bloku *MC_MoveAbsolute* musí být přidán odpovídající příkaz pro blokování kontaktů, aby bylo možné zahájit pohyb. Program blokování kontaktů byl popsán v sekci 8.2 kapitoly 8. Výtah se pak začne pohybovat podle příkazu zaslání bloku *MC_MoveAbsolute*.

Po provedení příkazu musíme resetovat příslušný bit, který je zodpovědný za uzamčení kontaktů našeho tlačítka na softwarové úrovni. Program pro resetování tohoto bitu má stejnou logiku, jaká byla popsána dříve. Program byl podrobněji popsán v sekci 8.2.3 kapitoly 8. V tomto okamžiku se také provede příkaz LED indikace polohy výtahu, který je popsán v sekci 8.7 kapitoly 8.

Po dokončení příkazu začne program zpracovávat příkazy z paměti. Pokud je paměť prázdná a během provádění příkazu výtahu nebylo stisknuto žádné tlačítko, přejde výtah do fáze čekání na příkaz, jak ukazuje algoritmus na obrázku 9.2. Pokud není paměť prázdná, začne tato část algoritmu zpracovávat další příkaz až po dokončení předchozího příkazu. Pokud je v paměťové proměnné bit (zápis do paměti – kapitola 9.3) a výtah dokončil předchozí příkaz, aktivuje se příslušná proměnná, která nyní nabývá hodnoty logické 1. Tato proměnná aktivace paměti je také připojena k sekci *Execute* příslušného bloku *MC_MoveAbsolute*. Proměnná aktivace paměti vyšle odpovídající signál bloku *MC_MoveAbsolute* a výtah pojedou podle příkazu ve frontě. Příkaz ke zpracování paměti je uspořádán tak, že nižší patra mají prioritu. Je to proto, že bloky aktivace proměnné paměti mají v programu své vlastní pořadí a program nejdříve provede bloky, které jsou v programu první. Po provedení příkazu z fronty se vynuluje příslušná paměťová proměnná a proměnné aktivace paměti. Taky se aktivuje příslušná LED na panelu HMI. Podle tohoto algoritmu se aktivují paměťové proměnné pro signály z tlačítek na panelu HMI a signály z tlačítek připojených k PLC.

Ve výsledku 3 proměnné jsou připojeny k sekci *Execute* bloku *MC_MoveAbsolute* a vzájemně propojeny blokem *OR*. První proměnná se aktivuje po stisknutí tlačítka, když je výtah volný a připravený k provedení příkazu. Druhou proměnnou je aktivace paměťové proměnné, která obsahuje informace o stisknutí tlačítka z panelu HMI. Třetí proměnnou je aktivace paměťové proměnné, která obsahuje informace o stisknutí tlačítek připojených k PLC.



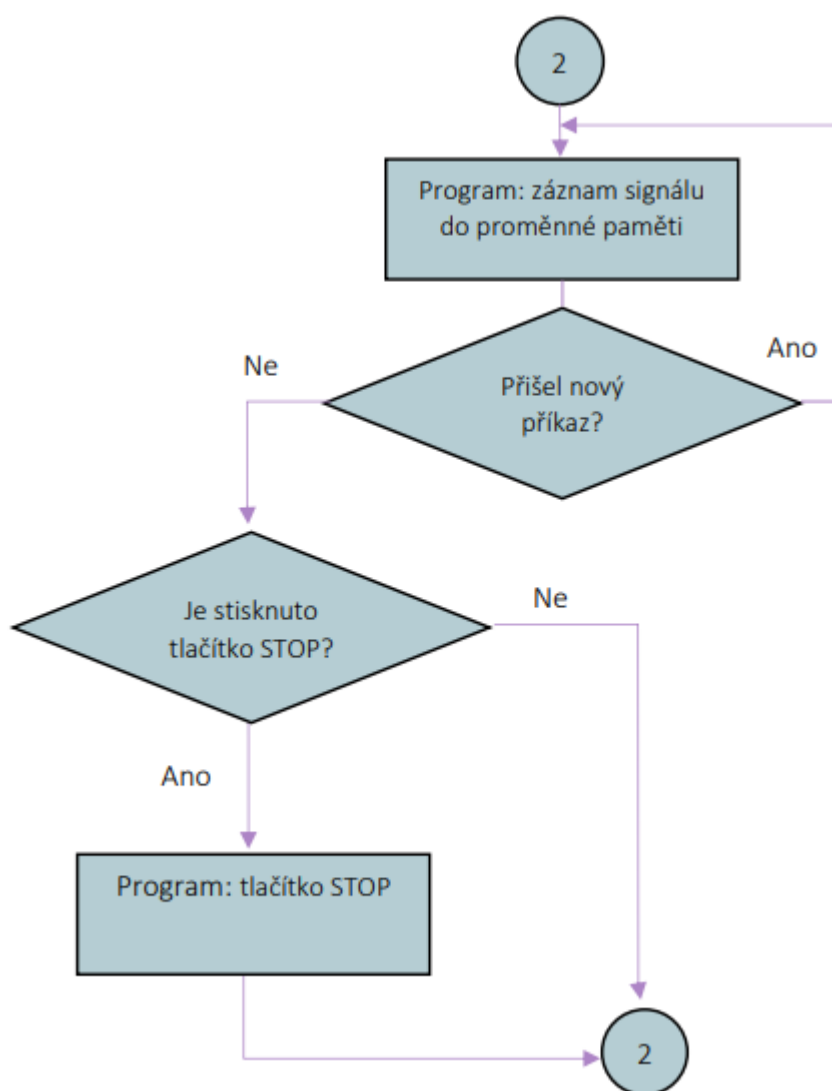
Obrázek 9.2 Algoritmus normálního provozního režimu.

9.3 Záznam signálu z tlačítka do paměti

Tento program zapiše signál z tlačítka do proměnné paměti. Signál je zapsán do proměnné paměti, když výtah nemůže provést příkaz právě teď, protože provádí předchozí příkaz.

Nejprve se provede záznam do proměnné paměti a totéž se provede pro všechny přijaté signály, zatímco výtah vykonává příkaz. K určení obsazenosti výtahu se používá proměnná "vytahobsazenu" (obrázek 8.2), která je zodpovědná za aktuální obsazenost výtahu. Tato proměnná je připojena k sekci Busy bloku *MC_MoveAbsolute* (obrázek 6.11). Pokud alespoň jeden z bloků *MC_MoveAbsolute* provede příkaz, výstup bloku v sekci Busy vypíše stav *True*, který odpovídá logické 1. To způsobí, že výtah pracuje podle algoritmu, který vidíte na obrázku 8.2 vpravo. Tím se aktivuje paměťová proměnná aktivací příslušného bitu.

Dále následuje kontrola stavu tlačítka *STOP*. Pokud je na panelu HMI stisknuto tlačítko *STOP*, aktivuje se příslušný program "tlačítko STOP", jak je popsáno v kapitole 8.4, kapitola 8.



Obrázek 9.3 Algoritmus pro zápis do paměťové proměnné.

KAPITOLA 10: REALIZACE PROJEKTU ŘÍZENÉHO VÝTAHU

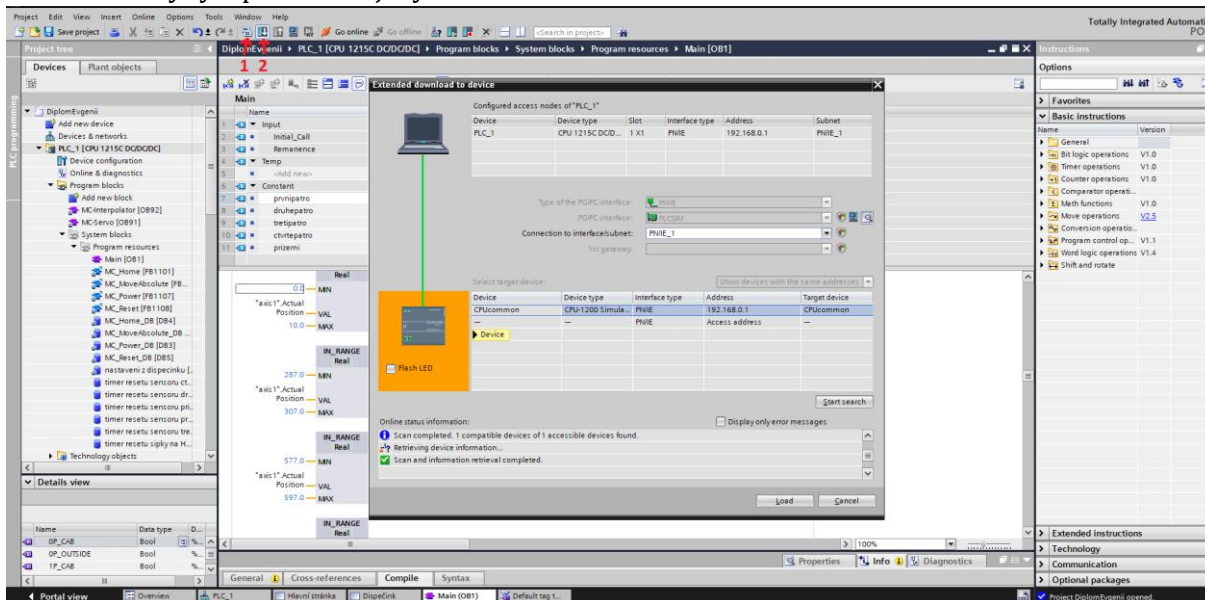
Po přidání všech systémových komponent do Tia portal, správném nastavení komunikace a vytvoření řídicího programu můžeme nahrát náš projekt do paměti PLC.

10.1 Nahrání projektu do PLC

Jakmile je program připraven, můžeme se jej pokusit nahrát do PLC. Za tímto účelem klikněte na ikonu PLC ve stromu projektů a klikněte na ikonu zkompilovat program (obrázek 10.1, číslo 1). Po zkompilování programu se ve spodním okně zobrazí výsledek kompilace. Pokud se program zkompiloval bez chyb, můžeme jej nahrát do PLC.

Pro stažení programu do PLC musíme kliknout na ikonu *Download to device*, označenou číslem 2 na obrázku 10.1. Otevře se okno zobrazené na obrázku 10.1. V tomto případě nahráváme program do simulátoru, takže v řádku rozhraní *PG/PC* nemáme na výběr. Pokud bychom program nahrávali do PLC, musel by být na tomto řádku vybrán náš PLC. V části *Připojení k rozhraní/podsíti* je třeba vybrat rozhraní PROFINET, jak je znázorněno na obrázku 10.1. Kliknutím na tlačítko *Load* v dolní části okna načtete náš projekt do PLC. Pomocí stejného postupu můžeme nahrát vizualizace do našeho HMI panelu.

Pokud se při kompilování programu nebo při jeho načítání do PLC vyskytne chyba, je třeba na ni kliknout. Software automaticky přejde do diagnostického okna. V diagnostickém okně si můžete přečíst text chyby a pokusit se ji vyřešit.



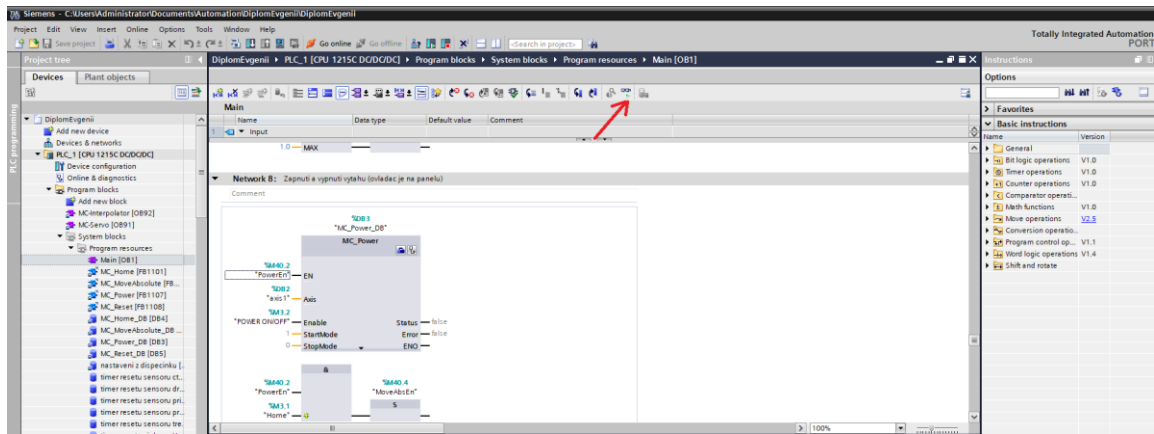
Obrázek 10.1 Nahrání projektu do PLC

10.2 Online monitoring

Po úspěšném načtení programu do paměti PLC musíme zkontrolovat jeho správnost. To lze provést pomocí modelu výtahu, který jsme naprogramovali. Pro kontrolu našeho systému musíme sledovat odezvu modelu na zadané příkazy pomocí tlačítek připojených k PLC a pomocí naprogramované vizualizace.

Fáze testování je velmi důležitá, protože pomáhá odhalit příkazy, které nejsou provedeny správně nebo nejsou provedeny vůbec. K otestování a ověření programu v prostředí Tia portal slouží pomocná funkce *Online monitoring*. Kliknutím na tuto funkci se celé rozhraní programu změní na modré. Modře podbarvené bloky a programové prvky odpovídají logické nule. Bloky a programové prvky podbarvené zeleně odpovídají logické 1.

Vstupní signály do těchto bloků lze zadávat prostřednictvím PLCSIM nebo stisknutím ovládacích tlačítek na panelu PLC nebo HMI.



Obrázek 10.2 Online monitoring v Tia portal

KAPITOLA 11: NÁVRH OVLÁDACÍHO ROZHRAŇÍ VÝTAHU

Tato kapitola popisuje rozhraní HMI panelu a jeho funkce. Počet naprogramovaných obrazovek v ovládacím rozhraní je celkem dvě. Jedna z obrazovek slouží k simulaci ovládní výtahu uživatelem v kabině a představuje uživatelské rozhraní. Druhá obrazovka slouží k vizualizaci servisního rozhraní a simuluje údržbu výtahu, např. v případě poruchy.

11.1 Servisní rozhraní

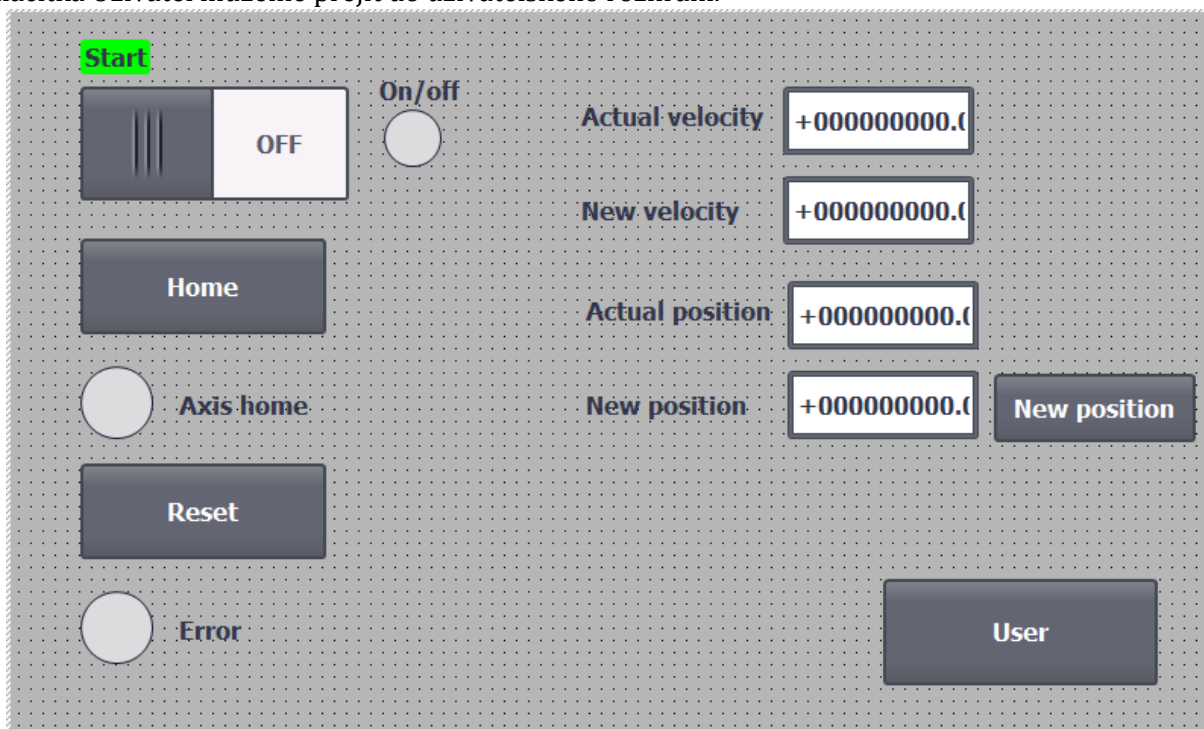
Pomocí servisního rozhraní můžeme servopohon zapnout stisknutím přepínače pod nápisem Start. LED dioda s nápisem On/off se rozsvítí zeleně a signalizuje, že je servopohon zapnutý.

Poté stisknete tlačítko *Home*, abyste výtah uvedli do provozu. Tím se spustí kalibrace výtahu osy a výtah pojedje do výchozího bodu, ze kterého se bude odpočítávat absolutní polohování. Po dokončení této funkce se vedle nápisu *Axis home* rozsvítí zelená LED dioda, která signalizuje, že příkaz byl proveden.

Pokud je zjištěna chyba, rozsvítí se LED dioda na pohonu červeně, blízko popisu *Error*. Pokud je chyba v blocích knihovny *Motion control*, můžeme zkusit stisknout tlačítko *Reset*, které chybu odstraní.

Na pravé straně rozhraní se nachází graf zobrazení rychlosti nazvaný *Actual velocity*. Kliknutím na ikonu *New velocity* se otevře nové okno, do kterého můžeme zadat rychlost výtahu.

Chcete-li nastavit novou polohu, musíte kliknout na ikonu *New position* a zadat novou polohu výtahu na ose v milimetrech. Poté je nutné novou pozici potvrdit tlačítkem *New position*. Pomocí tlačítka *User* můžeme přejít do uživatelského rozhraní.



Obrázek 11.1 Servisní rozhraní

11.2 Uživatelské rozhraní

Na levé straně uživatelského rozhraní vidíme tlačítka 0-4, která v tomto panelu nejsou fyzická, ale pouze animovaná. Tyto animace slouží ke sledování stavu fyzických tlačítek, která jsou připojena k diskretním vstupům PLC. Indikátory vedle nich se rozsvítí, když je příkaz z tlačítka ve frontě na provedení.

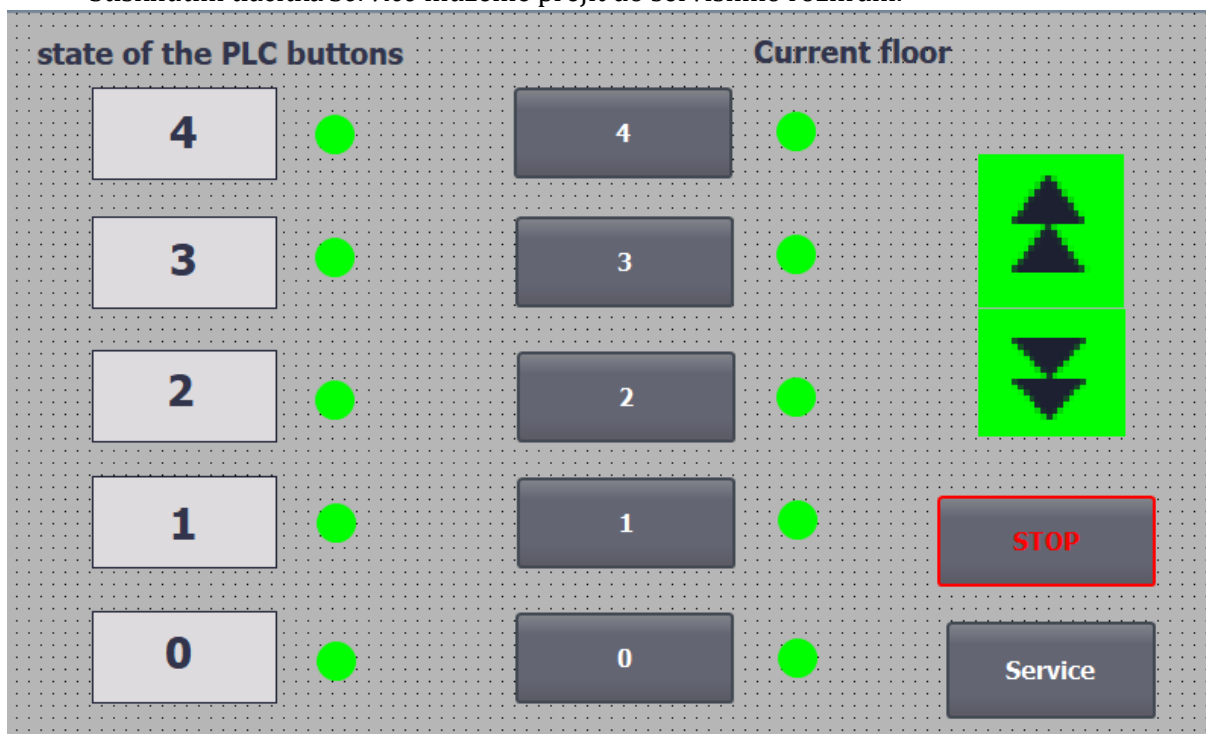
Na pravé straně uživatelského rozhraní vidíte tlačítka 0-4, což jsou již tlačítka, která lze stisknout. Když stisknete jedno z tlačítek, rozsvítí se zeleně, což znamená, že probíhá příkaz k provedení příkazu z tohoto tlačítka. Po provedení příkazu tlačítko opět zežedne.

Vpravo od tlačítek pro volání na podlahu se také nachází indikace. Tento údaj nám ukazuje aktuální polohu výtahu.

Zelené šipky ukazují směr jízdy výtahu. Když se výtah pohybuje, svítí pouze jeden z nich kvůli parametru viditelnosti objektu. Když vozidlo zastaví, obě šipky po vteřině zhasnou.

Tlačítkem stop vynulujeme všechny příkazy výtahu a umožníme mu dokončit pouze poslední příkaz, který právě provádí.

Stisknutím tlačítka *Service* můžeme přejít do servisního rozhraní.



Obrázek 11.2 Ovládací panel výtahu

ZÁVĚR

Cílem této práce bylo vyvinout řídicí systém pro výtah pomocí tlačítek připojených k PLC a vizualizovaného rozhraní na HMI panelu.

V úvodu jsme hovořili o vzniku PLC, jeho historii a praktickém využití. Popsali jsme hlavní součásti systému a programovací jazyky, které se používají k programování PLC. Seznámili jsme se také se základy používání vývojového prostředí Tia portal, konkrétně s vytvořením projektu a nastavením našeho PLC.

V kapitolách 1,2 a 3 je popsána teorie, která umožňuje pochopit, jak PLC funguje a z čeho se skládá. Byly popsány funkce procesoru, paměti a komunikačního rozhraní. Bylo popsáno programovací jazyky používané ve vývojovém prostředí Tia portal. Dále byla popsána funkce a technické vlastnosti hlavních komponent systému, konkrétně PLC – SIMATIC S7-1200 CPU 1215C, HMI panelu SIMATIC HMI KTP700 Basic, Napájení Blok - Siemens SIMATIC PM 1207, čtyř kanálového modulu Siemens SIMATIC NET CSM 1277 a akčního členu - sestávajícího z převodníku SINAMICS V90 PN a servomotoru SIMOTICS S-1FL6.

V kapitolách 4, 5, 6 a 7 byly popsány základy vytvoření projektu v prostředí Tia portal. Byly vysvětleny všechny kroky od vytvoření projektu až po nastavení komunikace PLC se servopohonem a HMI panelem. Pro komunikaci mezi PLC a servopohonem a panelem HMI byl nastaven komunikační protokol *PROFINET*. Byl rozebrán proces vytvoření technologického objektu a další využití bloků knihovny *Motion control*. Byl rozebrán proces tvorby vizualizace pro panel HMI.

V kapitole 8 byly rozebrány hlavní body a softwarová řešení tohoto projektu. Popsali jsme algoritmy pro řízení servopohonu pomocí funkčních bloků z knihovny *Motion control*. Bylo navrženo softwarové řešení funkce paměti podlaží. Toto řešení bylo realizováno pomocí reléové logiky a využívá algoritmus pro doručení kabiny dolů, tj. prioritní patro bude vždy patro 0. Nebylo použito tzv. sortování příkazů podle priority, protože toto řešení by z pohledu reléové logiky bylo složité. Proto je tato cesta možným způsobem rozšíření tohoto projektu.

V kapitolách 9 a 10 byl popsán způsob nahrání projektu do PLC a funkce servisního a uživatelského rozhraní.

Při vytváření projektu se často vyskytoval problém s nastavením komunikace, kdy se komunikace mezi PLC a pohonem nechtěla navázat a zobrazovala se chyba. V tomto případě bylo často nutné zavřít a znovu otevřít Tia portal a ve většině případů to pomohlo. Pokud chyba přetrvává, doporučujeme znovu přidat pohon do projektu a znovu jej nakonfigurovat, protože najít chybu v existující komunikaci může být někdy velmi obtížné, a to i s použitím diagnostické sekce.

LITERATURA

- [1] К. Е. Нестеров, А. М. Зюзев. *Программирование промышленных контроллеров: учеб.-метод, пособие.* Екатеринбург. Изд-во Урал, ун-та, 2019. 96 с. ил. ISBN 978-5-7996-2693-8. [online]. Dostupné z: https://elar.urfu.ru/bitstream/10995/77148/1/978-5-7996-2693-8_2019.pdf
- [2] Петров И. В. *Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования.* Моск. Под ред. проф .В. П. Дьяконова М.: СОЛОН-Пресс 2010. ISBN 5-98003-079-4. [online]. Dostupné z: https://books.google.cz/books?id=muzOBwAAQBAJ&printsec=frontcover&hl=ru&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.
- [3] *History of the PLC*, online knihovna library.automationdirect.com. [online]. Dostupné z: <https://library.automationdirect.com/history-of-the-plc/>
- [4] CHRIS VAVRA. *Dick Morley remembered as 'Father of the PLC'*, online knihovna *Control Engeneering*, 20.10. 2017 [online]. Dostupné z: <https://www.controleng.com/articles/dick-morley-remembered-as-father-of-the-plc/>
- [5] О.В. Шишов. *Программируемые контроллеры и системах промышленной автоматизации : учебник — Москва : ИНФРА-М, 2020. — 365 с. + Д оп. материалы* [online]. — DOI 10.12737/ 17505, ISBN 978-5-16-103331-9. [online]. Dostupné z: <https://library.atu.kz/flgl/48787.pdf>
- [6] Ajith Srikanth, Nehaa Vijay, T. Muthuramalingam. *Implementation of a Four-Floor Programmable Logic Controlled Elevator System.* Online knihovna *ResearchGate*. ISSN: 2321-9653, 06. 06. 2022. [online]. Dostupné z: https://www.researchgate.net/publication/361865098_Simulation_and_Modelling_of_3-Floor_Elevator_System_using_PLC
- [7] Угрюмов Е. П. *Цифровая схемотехника: Учеб. пособие для вузов. Санкт- Петербург. 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2007. — 800 с.: ил.* ISBN 978-5-94157-397-4. [online.] Dostupné z: https://www.bsuir.by/m/12_100229_1_85482.pdf
- [8] SIMATIC PM1207/1AC/DC24B/2.5A. Siemens. Munich. Germany. květen 2021. [online]. Dostupné z: <https://mall.industry.siemens.com/mall/ru/ru/Catalog/Product/6EP1332-1SH71>
- [9] Technické požadavky na evakuační výtahy ČSN 73 0802 čl.9.6.5 c, ČSN 73 0804 čl. 10.6.4.
- [10] А. И. Сергеев, А. М. Черноусова, А. С. Русяев. *Программирование контроллеров систем автоматизации: учебное пособие.* Оренбург. *Оренбургский гос. ун-т.* – Оренбург: ОГУ, 2016. – 125 с. ISBN 978-5-7410-1649-7. [online]. Dostupné z: http://elib.osu.ru/bitstream/123456789/13153/1/32822_20170111.pdf
- [11] Ephrem Ryan Alphonsus a , Mohammad Omar Abdulla. *A review on the applications of programmable logic controllers (PLCs).* červenec 2016. Online knihovna *ResearchGate*. [online]. Dostupné z: https://www.researchgate.net/publication/297889210_A_review_on_the_applications_of_programmable_logic_controllers_PLCs
- [12] *Programovací jazyky pro PLC, 24 listopadu 2019.* Vyukový portál COptel. [online]. Dostupne z: <https://coptel.cz/mod/page/view.php?id=6737>
- [13] KADLEČÍK, Pavel. *Možnosti využití programovatelných automatů Simatic řady S7-1200 od firmy Siemens.* Zlín, 2017. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Ing. Pavel Navrátil, Ph.D. [online]. Dostupné z: https://digilib.k.utb.cz/bitstream/handle/10563/41283/kadle%C4%8D%C3%ADk_2017_dp.pdf?sequence=1&isAllowed=y
- [14] *S7-1200 Programmable controller.* Siemens. Munich. Germany. Duben 2012. [online] . Dostupne z: https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf
- [15] *S7-1200 Programmable controller and PM 1207.* Siemens. Munich. Germany. 2017. [online]. Dostupne z: <https://assets.new.siemens.com/siemens/assets/api/uuid:79ee896b84a9c93e685fa0ea2cce6f7c1419fff1/s7-1200-2017-rus.pdf>
- [16] *Siemens SIMATIC NET CSM 1277.* Siemens. Munich. Germany Leden 2022 manual. [online]. Dostupne z: <https://mall.industry.siemens.com/mall/ru/ua/Catalog/DatasheetDownload?downloadUrl=teddatasheet%2F%3Fformat%3DPDF%26caller%3DMall%26mlfbs%3D6GK7277-1AA10-0AA0%26language%3Dru>
- [17] *SIMATIC HMI KTP700 Basic.* Siemens. Munich. Germany. Červenec 2015 manual. [online]. Dostupne z: https://www.distrelec.cz/Web/Downloads/eng/siemens_6AV21232GB030AX0_eng.pdf

- [18] SINAMICS V90 and SIMOTICS S-1FL6. *Optimized servo drive solution for motion control applications*. Siemens. Munich. Germany. 2016. [online]. Dostupne z: http://www.mcs.com.tr/images/sinamics_v90document_2dcrund_75135.pdf
- [19] TIA Portal STEP 7 Basic V10.5. Siemens. Munich. Germany. 2009.[online]. Dostupne z: https://cache.industry.siemens.com/dl/files/542/40263542/att_829827/v1/GS_STEP7Bas105enUS.pdf
- [20] SIMATIC S7-1200 CPU1215C, DC/DC/DC, 14DI/10DO/2AI/2AO. Siemens. Munich. Germany. [online]. Dostupne z: <https://support.industry.siemens.com/cs/pd/142915?pdtdi=td&dl=en&lc=en-CZ>
- [21] PRONETA Basic 3.4 SP1 *Commissioning and Diagnostics Tool for PROFINET*. Siemens. Munich. Germany. [online]. Dostupné z: <https://support.industry.siemens.com/cs/document/67460624/proneta-basic-3-4-sp1-commissioning-and-diagnostics-tool-for-profinet?dti=0&lc=en-CZ>
- [22] *Testing and simulating HMI projects (TIA Portal) in conjunction with SIMATIC controllers, PLCSIM and PLCSIM Advanced*. Siemens. Munich. Germany. 2019. [online]. Dostupné z: [https://support.industry.siemens.com/cs/document/109748099/testing-and-simulating-hmi-projects-\(tia-portal\)-in-conjunction-with-simatic-controllers-plcsim-and-plcsim-advanced?dti=0&dl=en&lc=ru-RU](https://support.industry.siemens.com/cs/document/109748099/testing-and-simulating-hmi-projects-(tia-portal)-in-conjunction-with-simatic-controllers-plcsim-and-plcsim-advanced?dti=0&dl=en&lc=ru-RU)
- [23] *Encoderless positioning with SIMATIC S7-1200 / 1500*, červen 2019. Siemens. Munich. Germany. [online]. Dostupné z : <https://www.tianadosah.cz/upload/463-1305439148.pdf>
- [24] Doc. RNDr. Zdeněk Hlaváč . ŠROUBOVÝ A PROSTOROVÝ POHYBOTAČNĚ SYMETRICKÉHO TĚLESA. [online]. Dostupné z : <https://docplayer.cz/13333518-Sroubovy-a-prostorovy-pohyb-rotacne-symetrickeho-telesa.html>
- [25] Mgr. Petr Osička, Ph.D . Metoda hrubé síly, backtracking a branch-and-bound. 2014. [online]. Dostupné z: <https://docplayer.cz/12378445-Metoda-hrube-sily-backtracking-a-branch-and-bound.html>

PŘÍLOHA A: TECHNICKÉ ÚDAJE ZAŘÍZENÍ SESTAVY

A.1 SIMATIC S7-1200 CPU 1215C

Tab. Technické údaje zařízení Siemens SIMATIC S7-1200 CPU 1215C [20].

General information	
Product type designation	CPU 1215C DC/DC/DC
Firmware version	V4.4
Engineering with	
• Programming package	STEP 7 V16 or higher
Supply voltage	
Rated value (DC)	
• 24 V DC	Yes
permissible range, lower limit (DC)	20.4 V
permissible range, upper limit (DC)	28.8 V
Reverse polarity protection	Yes
Load voltage L+	
• Rated value (DC)	24 V
• permissible range, lower limit (DC)	20.4 V
• permissible range, upper limit (DC)	28.8 V
Input current	
Current consumption (rated value)	500 mA
Current consumption, max.	1 500 mA; CPU with all expansion modules
Inrush current, max.	12 A; at 28.8 V DC
I^2t	0.5 A ² ·s
Output current	
for backplane bus (5 V DC), max.	1 600 mA; Max. 5 V DC for SM and CM
Encoder supply	
24 V encoder supply	
• 24 V	L+ minus 4 V DC min.
Power loss	
Power loss, typ.	12 W
Memory	
Work memory	
• integrated	125 kbyte
• expandable	No
Load memory	
• integrated	4 Mbyte
• Plug-in (SIMATIC Memory Card), max.	with SIMATIC memory card

Tab. Technické údaje zařízení Siemens SIMATIC S7-1200 CPU 1215C

Backup	
<ul style="list-style-type: none"> • present 	Yes
<ul style="list-style-type: none"> • maintenance-free 	Yes
<ul style="list-style-type: none"> • without battery 	Yes
CPU processing times	
for bit operations, typ.	0.08 µs; / instruction
for word operations, typ.	1.7 µs; / instruction
for floating point arithmetic, typ.	2.3 µs; / instruction
CPU-blocks	
Number of blocks (total)	DBs, FCs, FBs, counters and timers. The maximum number of addressable blocks ranges from 1 to 65535. There is no restriction, the entire working memory can be used
OB	
<ul style="list-style-type: none"> • Number, max. 	Limited only by RAM for code
Data areas and their retentivity	
Retentive data area (incl. timers, counters, flags), max.	10 kbyte
Flag	
<ul style="list-style-type: none"> • Number, max. 	8 kbyte; Size of bit memory address area
Local data	
<ul style="list-style-type: none"> • per priority class, max. 	16 kbyte
Address area	
Process image	
<ul style="list-style-type: none"> • Inputs, adjustable 	1 kbyte
<ul style="list-style-type: none"> • Outputs, adjustable 	1 kbyte
Hardware configuration	
Number of modules per system, max.	3 comm. modules, 1 signal board, 8 signal modules
Time of day	
Clock	
<ul style="list-style-type: none"> • Hardware clock (real-time) 	Yes
<ul style="list-style-type: none"> • Backup time 	480 h; Typical
Digital inputs	
Number of digital inputs	14; Integrated
<ul style="list-style-type: none"> • of which inputs usable for technological functions 	6; HSC (High Speed Counting)
Source/sink input	Yes
Number of simultaneously controllable inputs	
all mounting positions	
— up to 40 °C, max.	14
Input voltage	
<ul style="list-style-type: none"> • Rated value (DC) 	24 V
<ul style="list-style-type: none"> • for signal "0" 	5 V DC at 1 mA
<ul style="list-style-type: none"> • for signal "1" 	15 V DC at 2.5 mA
Input delay (for rated value of input voltage)	
for standard inputs	

— parameterizable	0.2 ms, 0.4 ms, 0.8 ms, 1.6 ms, 3.2 ms, 6.4 ms and 12.8 ms, selectable in groups of four
— at "0" to "1", min.	0.2 ms
— at "0" to "1", max.	12.8 ms
for interrupt inputs	
— parameterizable	Yes
for technological functions	
— parameterizable	Single phase: 3 @ 100 kHz & 3 @ 30 kHz, differential: 3 @ 80 kHz & 3 @ 30 kHz
Cable length	
• shielded, max.	500 m; 50 m for technological functions
• unshielded, max.	300 m; for technological functions: No
Digital outputs	
Number of digital outputs	10
• of which high-speed outputs	4; 100 kHz Pulse Train Output
Limitation of inductive shutdown voltage to	L+ (-48 V)
Switching capacity of the outputs	
• with resistive load, max.	0.5 A
• on lamp load, max.	5 W
Output voltage	
• for signal "0", max.	0.1 V; with 10 kOhm load
• for signal "1", min.	20 V
Output current	
• for signal "1" rated value	0.5 A
• for signal "0" residual current, max.	0.1 mA
Output delay with resistive load	
• "0" to "1", max.	1 μ s
• "1" to "0", max.	5 μ s
Switching frequency	
• of the pulse outputs, with resistive load, max.	100 kHz
Relay outputs	
• Number of relay outputs	0
Cable length	
• shielded, max.	500 m
• unshielded, max.	150 m
Analog inputs	
Number of analog inputs	2
Input ranges	
• Voltage	Yes
Input ranges (rated values), voltages	
• 0 to +10 V	Yes
— Input resistance (0 to 10 V)	$\geq 100k$ ohms
Cable length	
• shielded, max.	100 m; twisted and shielded

Analog outputs	
Number of analog outputs	2
Output ranges, current	
<ul style="list-style-type: none"> • 0 to 20 mA 	Yes
Analog value generation for the inputs	
Integration and conversion time/resolution per channel	
<ul style="list-style-type: none"> • Resolution with overrange (bit including sign), max. 	10 bit
<ul style="list-style-type: none"> • Integration time, parameterizable 	Yes
<ul style="list-style-type: none"> • Conversion time (per channel) 	625 µs
Analog value generation for the outputs	
Integration and conversion time/resolution per channel	
<ul style="list-style-type: none"> • Resolution with overrange (bit including sign), max. 	10 bit
Encoder	
Connectable encoders	
<ul style="list-style-type: none"> • 2-wire sensor 	Yes
1. Interface	
Interface type	PROFINET
Isolated	Yes
automatic detection of transmission rate	Yes
Autonegotiation	Yes
Autocrossing	Yes
Interface types	
<ul style="list-style-type: none"> • RJ 45 (Ethernet) 	Yes
<ul style="list-style-type: none"> • Number of ports 	2
<ul style="list-style-type: none"> • integrated switch 	Yes
Protocols	
<ul style="list-style-type: none"> • PROFINET IO Controller 	Yes
<ul style="list-style-type: none"> • PROFINET IO Device 	Yes
<ul style="list-style-type: none"> • SIMATIC communication 	Yes
<ul style="list-style-type: none"> • Open IE communication 	Yes; Optionally also encrypted
<ul style="list-style-type: none"> • Web server 	Yes
<ul style="list-style-type: none"> • Media redundancy 	Yes; as MRP client
PROFINET IO Controller	
<ul style="list-style-type: none"> • Transmission rate, max. 	100 Mbit/s
Services	
<ul style="list-style-type: none"> — PG/OP communication 	Yes
<ul style="list-style-type: none"> — Isochronous mode 	No
<ul style="list-style-type: none"> — IRT 	No
<ul style="list-style-type: none"> — PROFIenergy 	No
<ul style="list-style-type: none"> — Prioritized startup 	Yes
<ul style="list-style-type: none"> — Number of IO devices with prioritized startup, max. 	16

— Number of connectable IO Devices, max.	16
— Number of connectable IO Devices for RT, max.	16
— of which in line, max.	16
— Activation/deactivation of IO Devices	Yes
— Number of IO Devices that can be simultaneously activated/deactivated, max.	8
— Updating time	The minimum value of the update time also depends on the communication component set for PROFINET IO, on the number of IO devices and the quantity of configured user data.
PROFINET IO Device	
Services	
— PG/OP communication	Yes
— Isochronous mode	No
— IRT	No
— PROFINergy	Yes
— Shared device	Yes
— Number of IO Controllers with shared device, max.	2

A.2 Siemens SIMATIC PM 1207

Tab. Technické údaje zařízení Siemens SIMATIC PM 1207 [15].

Input	
Input	1-phase AC
• Note	Automatic range selection
supply voltage	
• 1 at AC rated value	120 V
• 2 at AC rated value	230 V
input voltage	
• 1 at AC	85 ... 132 V
• 2 at AC	176 ... 264 V
Wide-range input	No
Overvoltage resistance	2.3 × Vin rated, 1.3 ms
Mains buffering	at Vin = 93/187 V
Mains buffering at Iout rated, min.	20 ms; at Vin = 93/187 V
Rated line frequency 1	50 Hz
Rated line frequency 2	60 Hz
Rated line range	47 ... 63 Hz
input current	
• at rated input voltage 120 V	1.2 A
• at rated input voltage 230 V	0.67 A
Switch-on current limiting (+25 °C), max.	13 A
duration of inrush current limiting at 25 °C	
• maximum	3 ms
I ² t, max.	0.5 A ² ·s
Built-in incoming fuse	T 3,15 A/250 V (not accessible)

Protection in the mains power input (IEC 898)	Recommended miniature circuit breaker: 16 A characteristic B or 10 A characteristic C
Output	
Output	Controlled, isolated DC voltage
Rated voltage Vout DC	24 V
Total tolerance, static ±	3 %
Static mains compensation, approx.	0.1 %
Static load balancing, approx.	0.2 %
Residual ripple peak-peak, max.	150 mV
Spikes peak-peak, max. (bandwidth: 20 MHz)	240 mV
product function output voltage adjustable	No
Output voltage setting	-
Status display	Green LED for 24 V OK
On/off behavior	No overshoot of Vout (soft start)
Startup delay, max.	6 s; 2 s at 230 V, 6 s at 120 V
Voltage rise, typ.	10 ms
Rated current value Iout rated	2.5 A
Current range	0 ... 2.5 A
supplied active power typical	60 W
short-term overload current	
• on short-circuiting during the start-up typical	6 A
• at short-circuit during operation typical	6 A
duration of overloading capability for excess current	
• on short-circuiting during the start-up	100 ms
• at short-circuit during operation	100 ms
Parallel switching for enhanced performance	Yes
Numbers of parallel switchable units for enhanced performance	2
Efficiency	
Efficiency at Vout rated, Iout rated, approx.	83 %
Power loss at Vout rated, Iout rated, approx.	12 W

A.3 Siemens SIMATIC NET CSM 1277

Tab. Technické údaje zařízení Siemens SIMATIC NET CSM 1277 [16].

product type designation	SCALANCE CSM 1277
transfer rate	
transfer rate	10 Mbit/s, 100 Mbit/s

interfaces / for communication / integrated	
number of electrical connections • for network components or terminal equipment	4
number of 100 Mbit/s SC ports • for multimode	0
number of 1000 Mbit/s LC ports • for multimode • for single mode (LD)	0 0
interfaces / other	
number of electrical connections • for power supply	1
type of electrical connection • for power supply	3-pole terminal block
supply voltage, current consumption, power loss	
type of voltage / 1 / of the supply voltage • supply voltage / 1 / rated value • power loss [W] / 1 / rated value • supply voltage / 1 / rated value • consumed current / 1 / maximum • type of electrical connection / 1 / for power supply • product component / 1 / fusing at power supply input	DC 24 V 1.6 W 19.2 ... 28.8 V 0.07 A 3-pole terminal block Yes
ambient conditions	
ambient temperature • during operation • during storage • during transport	0 ... 60 °C -40 ... +70 °C -40 ... +70 °C
relative humidity • at 25 °C / without condensation / during operation / maximum	95 %
protection class IP	IP20
design, dimensions and weights	
design	SIMATIC S7-1200 device design
width	45 mm
height	100 mm
depth	75 mm
net weight	0.15 kg
fastening method • 35 mm top hat DIN rail mounting • wall mounting • S7-300 rail mounting • S7-1500 rail mounting	Yes Yes No No
product functions / management, configuration, engineering	

product function	
<ul style="list-style-type: none"> • multiport mirroring 	No
product function / switch-managed	No
product functions / redundancy	
product function	
<ul style="list-style-type: none"> • Parallel Redundancy Protocol (PRP)operation in the PRP-network 	Yes
<ul style="list-style-type: none"> • Parallel Redundancy Protocol (PRP)Redundant Network Access (RNA) 	No
standards, specifications, approvals	
standard	
<ul style="list-style-type: none"> • for FM 	FM3611: Class 1, Divison 2, Group A, B, C, D / T..., CL.1, Zone 2, GP. IIC, T.. Ta
<ul style="list-style-type: none"> • for safety / from CSA and UL 	UL 508, CSA C22.2 No. 142
<ul style="list-style-type: none"> • for emitted interference 	EN 61000-6-4 (Class A)
<ul style="list-style-type: none"> • for interference immunity 	EN 61000-6-2
MTBF	273 y
standards, specifications, approvals / CE	
certificate of suitability / CE marking	Yes
standards, specifications, approvals / hazardous environments	
standard / for hazardous zone	EN 600079-15:2005, EN 600079-0:2006, II 3 G Ex nA II T4, KEMA 08 ATEX 0003 X
certificate of suitability	
<ul style="list-style-type: none"> • CCC approval hazardous zone 	Yes
standards, specifications, approvals / other	
certificate of suitability	EN 61000-6-2, EN 61000-6-4
<ul style="list-style-type: none"> • C-Tick 	Yes
<ul style="list-style-type: none"> • KC approval 	No

A.4 Siemens SIMATIC HMI KTP700 Basic

Tab. Technické údaje zařízení Siemens SIMATIC HMI KTP700 Basic [17].

General information	
Product type designation	KTP700 Basic color PN
Display	
Design of display	TFT widescreen display, LED backlighting
Screen diagonal	7 in
Display width	154.1 mm
Display height	85.9 mm
Number of colors	65 536
Resolution (pixels)	
• Horizontal image resolution	800 pixel
• Vertical image resolution	480 pixel
Backlighting	
• MTBF backlighting (at 25 °C)	20 000 h
• Backlight dimmable	Yes
Control elements	
Keyboard fonts	
• Function keys	
— Number of function keys	8
• System keys	No
• Numeric/alphabetical input	
— Numeric keyboard	Yes; Onscreen keyboard
— alphanumeric keyboard	Yes; Onscreen keyboard
Touch operation	
• Design as touch screen	Yes
Installation type/mounting	
Mounting in portrait format possible	Yes
Mounting in landscape format possible	Yes
maximum permissible angle of inclination without external ventilation	35°
Supply voltage	
Type of supply voltage	DC
Rated value (DC)	24 V
permissible range, lower limit (DC)	19.2 V
permissible range, upper limit (DC)	28.8 V
Input current	
Current consumption (rated value)	230 mA
Starting current inrush I st	0.2 A ² ·s
Power	
Power consumption, typ.	5.5 W

Processor	
Processor type	ARM
Memory	
Flash	Yes
RAM	Yes
Type of output	
Acoustics	
• Buzzer	Yes
• Speaker	No
Time of day	
Clock	
• Hardware clock (real-time clock)	Yes
• Software clock	Yes
• retentive	Yes; Back-up duration typically 6 weeks
• synchronizable	Yes
Interfaces	
Number of RS 485 interfaces	0
Number of USB interfaces	1; Up to 16 GB
Number of 20 mA interfaces (TTY)	0
Number of RS 232 interfaces	0
Number of RS 422 interfaces	0
Number of parallel interfaces	0
Number of other interfaces	0
Number of SD card slots	0
With software interfaces	No
Industrial Ethernet	
• Number of industrial Ethernet interfaces	1
• Industrial Ethernet status LED	2
Protocols	
PROFINET	Yes
Supports protocol for PROFINET IO	No
IRT	No
MRP	No
PROFIBUS	No
MPI	No
Protocols (Ethernet)	
• TCP/IP	Yes
• DHCP	Yes
• SNMP	Yes
• DCP	Yes
• LLDP	Yes
WEB characteristics	
• HTTP	No
• HTML	No

Further protocols	
• CAN	No
• MODBUS	Yes; Modicon (MODBUS TCP/IP)
• EtherNet/IP	Yes
Interrupts/diagnostics/status information	
Diagnostic messages	
• Diagnostic information readable	No
EMC	
Emission of radio interference acc. to EN 55 011	
• Limit class A, for use in industrial areas	Yes
• Limit class B, for use in residential areas	No
Degree and class of protection	
IP (at the front)	65
Enclosure Type 4 at the front	Yes
Enclosure Type 4x at the front	Yes


A.5 Siemens SINAMICS V90 PN

Tab. Technické údaje zařízení Siemens SINAMICS V90 PN [18].

Line supply		200 ... 240 V 1AC/3AC							
Article No.	Pulse train: 6SL3210-5F PROFINET: 6SL3210-5F	B10-1UA0 B10-1UF0	B10-2UA0 B10-2UF0	B10-4UA1 B10-4UF1	B10-8UA0 B10-8UF0	B11-0UA1 B11-0UF1	B11-5UA0 B11-5UF0	B12-0UA0 B12-0UF0	
Max. motor power (kW)		0.1	0.2	0.4	0.75	1	1.5	2	
Rated output current (A)		1.2	1.4	2.6	4.7	6.3	10.6	11.6	
Max. output current (A)		3.6	4.2	7.8	14.1	18.9	31.8	34.8	
Line supply	Voltage	1/3AC 200 V ... 240 V (-15%/+10%)				3AC 200 V ... 240 V (-15%/+10%)			
	Frequency	50 Hz/60 Hz, (-10%/+10%)							
	Capacity (kVA) (1AC)	0.5	0.7	1.2	2	-	-	-	
	Capacity (kVA) (3AC)	0.5	0.7	1.1	1.9	2.7	4.2	4.6	
Cooling		Natural cooling				Fan cooling			
Frame size		FSA*		FSB	FSC	FSD			
Dimensions WxHxD (mm)		45x170x170*		55x170x170	80x170x195	95x170x195			
Weight approx. (kg)		1.07		1.20	1.94	2.49			
Line supply		380 ... 480 V 3AC							
Article No.	Pulse train: 6SL3210-5F PROFINET****: 6SL3210-5F	E10-4UA0 E10-4UF0	E10-8UA0 E10-8UF0	E11-0UA0 E11-0UF0	E111-5UA0 E111-5UF0	E12-0UA0 E12-0UF0	E13-5UA0 E13-5UF0	E15-0UA0 E15-0UF0	E17-0UA0 E17-0UF0
Max. motor power (kW)		0.4	0.75	1	1.75	2.5	3.5	5	7
Rated output current (A)		1.2	2.1	3	5.3	7.8	11	12.6	13.2
Max. output current (A)		3.6	6.3	9	15.9	23.4	33	37.8	39.6
Line supply	Voltage	3AC 380 V ... 480 V (-15% / +10%)							
	Frequency	50 Hz/60 Hz, (-10% / +10%)							
	Capacity (kVA)	1.7	3	4.3	6.6	11.1	15.7	18	18.9
Cooling		Natural cooling				Fan cooling			
Frame size		FSA*	FSA		FSB			FSC	
Dimensions WxHxD (mm)		60x180x20		80x180x200	100x180x220		140x260x240		
Weight approx. (kg)		1.45		2.09	2.73		5.95		
Control power supply	Voltage**	24 V DC (-15% / +20%)							
	Current***	1.6 A (without holding brake) 3.6 A (with holding brake)							
Line supply system		TN, TT, IT, TT earthed line							
Overload capacity		300% x rated current for 300 ms every 10 s							
Control system		Servo control							
Braking resistor		Integrated							
Ambient temperature	Operation	0 °C to 45 °C: without power derating, 45 °C to 55 °C : with power derating							
	Storage	-40 °C to +70 °C							
Ambient humidity	Operation	<90% (no condensation)							
	Storage	90% (no condensation)							
Pollution class		2							
Vibration	Operation	Shock	Operational area II Peak acceleration: 5 g, 30 ms, 15 g, 11 ms Quantity of shocks: 3 per direction x 6 direction Duration of shock: 1 s						
		Vibration	Operational area II 10 Hz to 58 Hz: 0.075 mm deflection 58 Hz to 200 Hz: 1 g vibration						
	Product packaging	Vibration	2 Hz to 9 Hz: 3.5 mm deflection 9 Hz to 200 Hz: 1 g vibration Quantity of cycles: 10 per axis Sweep speed: 1 octave/min						
Degree of protection		IP20							
Altitude		≤ 1000 m (without power derating); > 1000 m and up to 5000 m (with power derating)							
Standards		CE, KC, EAC, cULus, C-tick							

A.6 Siemens SIMOTICS S-1FL6

Tab. Technické údaje zařízení Siemens SIMOTICS S-1FL6 [18].

Technical data								
Article number 1FL6	022-2AF	024-2AF	032-2AF	034-2AF	042-2AF	044-2AF	052-2AF	054-2AF
Shaft height (SH)	20		30		40		50	
Rated power (kW) ¹⁾	0.05	0.10	0.20	0.40	0.75	1.00	1.50	2.00
Horsepower (HP)	0.07	0.14	0.27	0.54	1.02	1.36	2.04	2.72
Rated torque (Nm)	0.16	0.32	0.64	1.27	2.39	3.18	4.78	6.37
Rated speed (rpm)	3000							
Maximum torque (Nm)	0.48	0.96	1.91	3.82	7.2	9.54	14.3	19.1
Maximum speed (r/min)	5000							
Rated current (A)	1.2	1.2	1.4	2.6	4.7	6.3	10.6	11.6
Maximum current (A)	3.6	3.6	4.2	7.8	14.2	18.9	31.8	34.8
Torque constant (Nm/A)	0.14	0.29	0.48	0.49	0.51	0.51	0.46	0.55
Moment of inertia (10 ⁻⁴ kg·m ²) (with brake)	0.031 (0.038)	0.052 (0.059)	0.214 (0.245)	0.351 (0.381)	0.897 (1.06)	1.15 (1.31)	2.04 (2.24)	2.62 (2.82)
Thermal class	B (130 °C)							
Degree of protection	IP65							
Recommended load to motor inertia ratio	Max. 30x				Max. 20x		Max. 15x	
Encoder types	Incremental encoder TTL 2500 ppr; Absolute encoder single-turn 21-bit (available in the 2nd half of 2016)							
Type of construction	IM B5 (IM V1 and IM V3)							
Weight (kg) (with brake)	0.47 (0.70)	0.63 (0.86)	1.02 (1.48)	1.46 (1.92)	2.8 (3.68)	3.39 (4.20)	5.35 (6.76)	6.56 (8.00)
Operating temperature	0 – 40 °C (without any restrictions)						0 – 30 °C (without any restrictions)	
Operating humidity	90% RH maximum (no condensation at 30 °C)							
Vibration severity grade	Grade A							
Radial runout tolerance	Class N							
Installation altitude	≤ 1000 m (without power derating); > 1000 m and up to 5000 m (with power derating)							
Standards								
Holding brake data ²⁾								
Holding torque (Nm)	0.32 Nm		1.27 Nm		3.18 Nm		6.37 Nm	
Rated voltage (V)	24 V DC ± 10%							
Opening time (ms)	35		75		105		90	
Closing time (ms)	10		10		15		35	
Rated current (A)	0.25		0.3		0.35		0.57	