

**KATEDRA ELEKTRICKÝCH
POHONŮ A TRAKCE**

**ČESKÉ VYSOKÉ UČENÍ
TECHNICKÉ V PRAZE**



**FAKULTA ELEKTROTECHNICKÁ
ŘÍZENÍ POHONU DVOU OS
POMOCÍ SIMATIC S-1500**

DIPLOMOVÁ PRÁCE

KVĚTEN 2023

**BC. ADAM
STOČEK**



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Stoček** Jméno: **Adam** Osobní číslo: **474520**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektrických pohonů a trakce**
Studijní program: **Elektrotechnika, energetika a management**
Specializace: **Elektroenergetika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Řízení pohonu dvou os pomocí Simatic S-1500

Název diplomové práce anglicky:

Control of two axis drive with Simatic S-1500

Pokyny pro vypracování:

- 1) Seznamte se s problematikou návrhu řízení technologických celků pomocí PLC
- 2) Popište konfiguraci komponent potřebnou pro řízení pohonu dvou os pomocí PLC S1500 v prostředí TIA portál
- 3) Popište možnosti zadávání souřadnic do pohonu pomocí webového rozhraní OPC
- 4) V HMI navrhnete vizualizaci pro ovládání pohonu

Seznam doporučené literatury:

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

Jméno a pracoviště vedoucí(ho) diplomové práce:

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

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

Datum zadání diplomové práce: **01.12.2022** Termín odevzdání diplomové práce: **26.05.2023**

Platnost zadání diplomové práce: **16.02.2025**

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

podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

PODĚKOVÁNÍ

Tímto bych chtěl velmi poděkovat svému vedoucímu diplomové práce Ing. Janu Bauerovi, Ph.D., za cenné rady a čas strávený při realizaci této práce. Dále bych chtěl poděkovat za poskytnutí pracoviště, kde bylo možné ověřit funkčnost celé práce.

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 25. května 2023

.....

ABSTRAKT

Tato diplomová práce se zabývá řízením dvou os pomocí PLC SIMATIC 1511T-1 PN, které pracují jako plotter. V teoretické části je popsána problematika týkající se obecných informací o programování PLC a obsažených komponent. V praktické části je popsána konfigurace jednotlivých komponent ve vývojovém prostředí TIA Portal a následně princip vytvořené aplikace pro ovládání zmíněných os.

Klíčová slova: Siemens, PLC, HMI, SIMATIC, plotter, OPC UA

ABSTRACT

This diploma thesis deals with the control of two axes using PLC SIMATIC 1511T-1 PN, which work as a plotter. In the theoretical part, issues related to general information about PLC programming and included components are described. The practical part describes the configuration of individual components in the TIA Portal development environment and then the principle of the created application for controlling the mentioned axes.

Keywords: Siemens, PLC, HMI, SIMATIC, plotter, OPC UA

OBSAH

ÚVOD	1
KAPITOLA 1: TEORETICKÁ ČÁST	2
1.1 POPIS APLIKACE	2
1.2 PLC	2
1.2.1 Složení.....	3
1.2.2 Rozdělení.....	3
1.2.2.1 Kompaktní PLC.....	3
1.2.2.2 Modulární PLC.....	4
1.2.3 Princip činnosti.....	4
1.2.4 Norma IEC/EN 61131	5
1.2.4.1 Programovací jazyky.....	6
1.2.4.2 Structured Text (ST)	6
1.2.4.3 Instruction List (IL).....	6
1.2.4.4 Ladder Diagram (LD).....	6
1.2.4.5 Function Block Diagram (FBD).....	6
1.2.5 Centrální řízení	7
1.2.6 Decentrální řízení	7
1.3 FREKVENČNÍ MĚNIČ	7
1.4 ABSOLUTNÍ ENKODÉR	7
1.5 TIA PORTAL	8
1.6 KOMUNIKACE	9
1.6.1 RT.....	9
1.6.2 IRT.....	9
1.6.3 Ethernet.....	10
1.6.4 PROFIBUS.....	10
1.6.5 PROFINET	11
1.6.6 Telegramy	11
1.6.7 Telegram 105.....	12
1.7 OPC UA	12
KAPITOLA 2: POPIS SESTAVY (HARDWARE)	14
2.1 CPU SIMATIC 1511T-1 PN	14
2.1.1 SIMATIC S7-1500 DI/DO module.....	15
2.1.2 SINAMICS S210 PN	15
2.1.3 HMI KTP400 Basic PN.....	16
2.1.4 SIMOTICS S-1FK2.....	17
KAPITOLA 3: REALIZACE PROJEKTU V PROSTŘEDÍ TIA PORTAL	18
3.1 ZALOŽENÍ PROJEKTU	18
3.2 KONFIGURACE HARDWARU	18
3.2.1 CPU + DI/DQ modul.....	18
3.2.2 Frekvenční měniče	21
3.2.3 Motory	22
3.2.4 HMI.....	23
3.3 TECHNOLOGICKÉ OBJEKTY	24
3.3.1 Osa x	24
3.3.2 Osa y	26

3.3.3	Plotter	26
3.4	SOFTWARE	28
3.4.1	Library Kinematics Control	28
3.4.2	PLC tagy	29
3.4.3	Main	29
3.4.3.1	FB Zapnout_plotter	29
3.4.3.2	FB LkinCtrl_MC_GroupPower	32
3.4.3.3	FB LkinCtrl_MC_GroupReset	33
3.4.3.4	FB LkinCtrl_MC_GroupHome	35
3.4.3.5	Hlášení errorů	37
3.4.3.6	FB OPC-UA	38
3.4.4	Drawing	39
3.4.4.1	Rozsah zadávaných souřadnic	39
3.4.4.2	Ovládání lineárního pohybu	40
3.4.4.3	Aktivace pera	42
3.4.4.4	STOP	43
3.4.4.5	Přerušeni	43
3.4.4.6	Pokračování	44
3.4.4.7	JOG	45
3.5	HMI	45
3.5.1	HMI tagy	46
3.5.2	Obrazovka „Aktivace HMI“	46
3.5.3	Obrazovka „Ovladani“	47
3.6	OPC UA	50
	ZÁVĚR	52
	LITERATURA	53
	PŘÍLOHA A: SEZNAM ZKRATEK	56

SEZNAM OBRÁZKŮ

Obr. 1-1 Popis aplikace	2
Obr. 1-2 Kompaktní PLC LOGO! Siemens [3]	3
Obr. 1-3 Blokový diagram modulárního PLC [7]....	4
Obr. 1-4 Základní cyklus PLC [4]	5
Obr. 1-5 TIA Portal.....	8
Obr. 1-6 Komunikace RT [13]	9
Obr. 1-7 Komunikace IRT [13]	10
Obr. 1-8 Model komunikační sítě PROFIBUS [16]	11
Obr. 1-9 OPC UA server [19].....	13
Obr. 2-1 SIMATIC CPU 1511T-1 PN [21].....	14
Obr. 2-2 DI/DO přídatný modul [22].....	15
Obr. 2-3 SINAMICS S210 PN [24]	16
Obr. 2-4 HMI KTP400 Basic PN.....	16
Obr. 2-5 SIMOTICS S-1FK2 motor [26]	17
Obr. 2-6 Štítek motoru SIMOTICS S-1FK2.....	17
Obr. 3-1 Založení projektu v TIA Portal	18
Obr. 3-2 Konfigurace DI/DQ modulu.....	19
Obr. 3-3 Konfigurace CPU s DI/DQ modulem	19
Obr. 3-4 Nastavení IP adresy pro CPU	20
Obr. 3-5 Real Time settings CPU.....	20
Obr. 3-6 Konfigurace frekvenčního měniče.....	21
Obr. 3-7 Nastavení IP adresy pro frekvenční měnič	21
Obr. 3-8 Nastavení synchronizace frekvenčního měniče	22
Obr. 3-9 Přiřazení motoru k frekvenčnímu měniči	22
Obr. 3-10 Konfigurace HMI panelu.....	23
Obr. 3-11 Nastavení IP adresy pro HMI panel.....	23
Obr. 3-12 HW konfigurace.....	24
Obr. 3-13 Princip TO_PositioningAxis [27].....	25
Obr. 3-14 Přiřazení TO_PositioningAxis k frekvenčnímu měniči.....	25
Obr. 3-15 Princip TO_Kinematics [28].....	26
Obr. 3-16 Konfigurace TO_Kinematics.....	27
Obr. 3-17 Nastavení os pro TO_Kinematics.....	28
Obr. 3-18 Ovládání kinematiky [29]	29
Obr. 3-19 FB Zapnout_plotter	30

SEZNAM TABULEK

Tab. 2-1 Popis komponentů v sestavě	14
Tab. 3-1 Režimy pro nastavení výchozí polohy ...	36

Obr. 3-20 FB Zapnout_plotter – aktivace osy x	30
Obr. 3-21 FB Zapnout_plotter – pohyb osy na koncový spínač.....	31
Obr. 3-22 FB Zapnout_plotter – zastavení osy x..	31
Obr. 3-23 FB Zapnout_plotter – zapínání a vypínání bloku	32
Obr. 3-24 Funkční blok LKinCtrl_MC_GroupPower	32
Obr. 3-25 Funkční blok LKinCtrl_MC_GroupReset	34
Obr. 3-26 Funkční blok LKinCtrl_MC_GroupHome	35
Obr. 3-27 Posunutí výchozí pozice	37
Obr. 3-28 Logika pro vyhlášení erroru technologických objektů	38
Obr. 3-29 FB OPC-UA.....	39
Obr. 3-30 Logika pro rozsah zadávaných souřadnic	40
Obr. 3-31 Ovládání lineárního pohybu	41
Obr. 3-32 Aktivace pera.....	42
Obr. 3-33 Zastavení plotteru	43
Obr. 3-34 Přerušování pohybu.....	44
Obr. 3-35 Pokračování v pohybu.....	44
Obr. 3-36 Jog.....	45
Obr. 3-37 Funkce obrazovky „Aktivace HMI“	46
Obr. 3-38 Obrazovka “Ovládání”	47
Obr. 3-39 Nastavení objektu pro zapnutí plotteru	47
Obr. 3-40 Nastavení objektu pro aktivaci os.....	48
Obr. 3-41 Nastavení objektu pro výchozí pozici .	48
Obr. 3-42 Nastavení objektu pro souřadnice.....	48
Obr. 3-43 Nastavení objektu pro start	49
Obr. 3-44 Nastavení objektu pro aktivaci pera....	49
Obr. 3-45 Nastavení komunikace pro OPC klienta	50
Obr. 3-46 Výstupy z funkčního bloku pro OPC UA server.....	51
Obr. 3-47 Zapisování hodnot na výstupy přes OPC UA server	51

ÚVOD

Tato diplomová práce se zaměřuje na řízení plotteru pomocí PLC (*Programmable Logic Controller*) a k němu připojeným frekvenčním měničům, jež řídí servopohony. Ovládání plotteru může být realizováno dvěma způsoby. První možností je ovládání uživatelem přes HMI (*Human Machine Interface*) panel. Druhým způsobem je vzdálené ovládání přes OPC UA server. Všechny komponenty, které jsou v aplikaci pro řízení plotteru obsaženy, mají stejného výrobce, kterým je firma Siemens.

V teoretické části jsem se pro porozumění celé práce nejprve zaměřil na popis celého pracoviště, kde je pro jeho ilustraci popsán a vysvětlený obrázek s popisem všech komponent. Následuje povídání o PLC včetně složení, rozdělení, principu činnosti a také možnosti komunikace mezi ním a dalšími prvky v konfiguraci. Dále na normu, podle které se logické automaty programují. Tato norma obsahuje mezinárodní standardy pro jejich programování a dostupné programovací jazyky. Následuje popsání frekvenčních měničů a absolutního enkodéru, který je součástí použitých servopohonů. Celá aplikace je realizována ve vývojovém prostředí TIA Portal, jehož popis je taktéž obsažen v teoretické části.

V další části se věnuji popisu veškerého obsaženého hardwaru včetně použité verze firmwaru. Do něj patří hlavní řídicí jednotka, přídatný vstupně/výstupní modul, frekvenční měniče, servopohony a HMI panel.

V poslední části je popsána tvorba aplikace pro ovládání plotteru v prostředí TIA Portal. Nejprve je vysvětlena prvotní konfigurace všech zmíněných hardwarových prvků a potřebných technologických objektů. Poté je vysvětlen princip samotného kódu, který řídí celý plotter včetně realizace HMI panelu a OPC UA serveru.

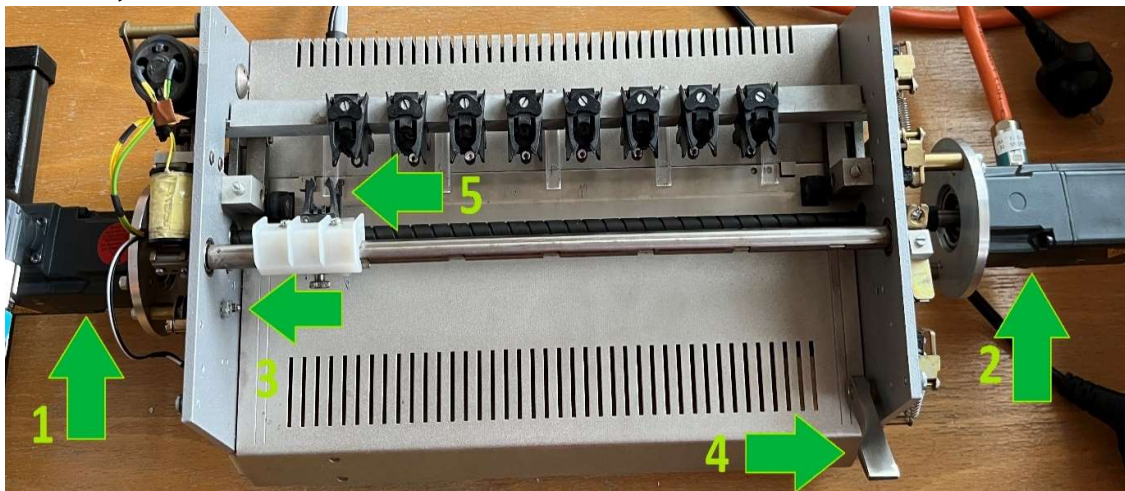
V Praze dne 22.5.2023

Bc. Adam Stoček

KAPITOLA 1: TEORETICKÁ ČÁST

1.1 Popis aplikace

Celá aplikace se skládá z konstrukce, která obsahuje podložku pro papír, který je k ní připevněn mechanickou pákou, dva servopohony ovládající osy, jeden koncový spínač na levé straně, zásobník pro kreslicí pero a příslušný hardware. Rozměr kreslení pro tutu aplikaci je formát papíru A4, tedy rozměry 210x297 mm. Vzhledem ke konstrukčním vlastnostem nelze najet s kreslicím perem až do krajů, proto jsou reálné rozměry omezené na kreslicí plochu o velikosti 190x267 mm. Dva servomotory slouží pro pohyb ve směru os x a y. První zmíněný servomotor obstarává pohyb po ose x, která představuje pohyb pera doleva-doprava. Druhý servomotor řídí pohyb ve směru osy y a posouvá tak papír ve směru nahoru-dolů. Jak již bylo zmíněno, aplikace obsahuje pouze jeden koncový spínač a to na levé straně. Proto je nutné omezit pracovní plochu softwarově přes limity pro osu x směrem doprava a osu y. Příslušný hardware, který je použit pro ovládání celé aplikace, bude popsán později v Kapitola 2:. Celá zmíněná konstrukce je znázorněna na následujícím obrázku.



Obr. 1-1 Popis aplikace

- 1 – Servopohon pro ovládání ve směru osy x
- 2 – Servopohon pro ovládání ve směru osy y
- 3 – Koncový spínač
- 4 – Zakládací páka pro papír
- 5 – Zásobník pro kreslicí pero

1.2 PLC

Programmable Logic Controller – zkráceně PLC, je programovatelný logický automat určený k řízení průmyslových procesů a strojů. Je nezbytnou součástí mnoha automatizovaných výrobních a průmyslových systémů. PLC byly poprvé představeny v 60. letech minulého století jako alternativa k elektromechanickým reléovým systémům. Od té doby se vyvinuly ve vysoce pokročilé systémy, které dokážou řídit složité průmyslové procesy. Hlavní funkcí PLC je přijímat vstupní signály z různých senzorů, zpracovávat informace a poté generovat výstupní signály pro řízení strojů a zařízení.

Vstupy a výstupy mohou být digitální nebo analogové a PLC může také provádět různé aritmetické a logické operace. PLC lze programovat pomocí specializovaného softwaru a programovacích jazyků, které budou popsány později. Tyto programovací jazyky umožňují vytvářet složité řídicí sekvence a algoritmy pro optimalizaci výkonu strojů a procesů, které řídí. Jednou z významných výhod PLC je jejich schopnost pracovat v náročných průmyslových prostředích. Jsou navrženy tak, aby odolávaly extrémním teplotám, prachu, vibracím, a v těchto podmínkách mohli spolehlivě pracovat. PLC způsobily revoluci v průmyslové automatizaci a umožnily zvýšit produktivitu či bezpečnost. Díky své schopnosti řídit a monitorovat více procesů současně se staly nezbytnou součástí moderních průmyslových systémů. [1][2]

1.2.1 Složení

Základem každého PLC systému jsou dva hlavní komponenty: centrální procesorová jednotka (CPU) a digitální/analogové vstupní a výstupní moduly (I/O moduly). Hlavní úkolem CPU je řízení celého programu. I/O moduly pracují jako rozhraní mezi výstupním zařízením (např. strojem) a mikroprocesorem. [3]

1.2.2 Rozdělení

PLC se z hlediska konstrukce dělí na dvě skupiny: kompaktní a modulární

1.2.2.1 Kompaktní PLC

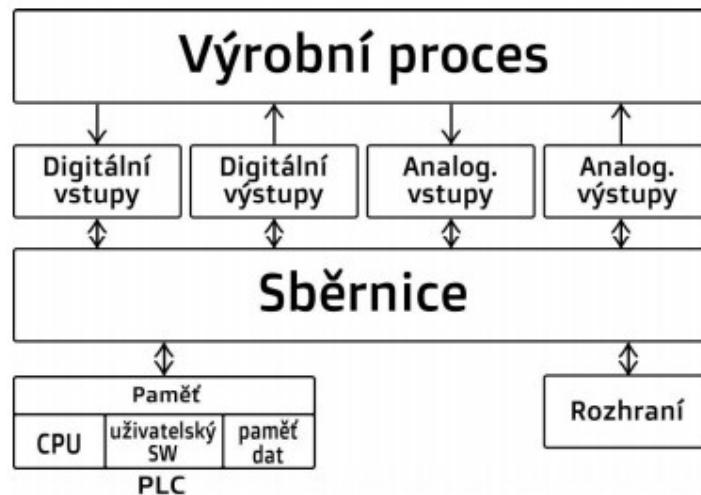
U kompaktních systémů jsou všechny komponenty obsaženy v jednom zařízení. Mezi ně patří CPU, digitální a analogové I/O moduly, základní podpora komunikace a někdy také napájecí zdroj. Tento typ PLC se používá pro řízení jednodušších aplikací vzhledem k pořizovacím nákladům. Jako příklad kompaktního PLC si můžeme uvést kompaktní PLC LOGO! od značky Siemens. [3]



Obr. 1-2 Kompaktní PLC LOGO! Siemens [3]

1.2.2.2 Modulární PLC

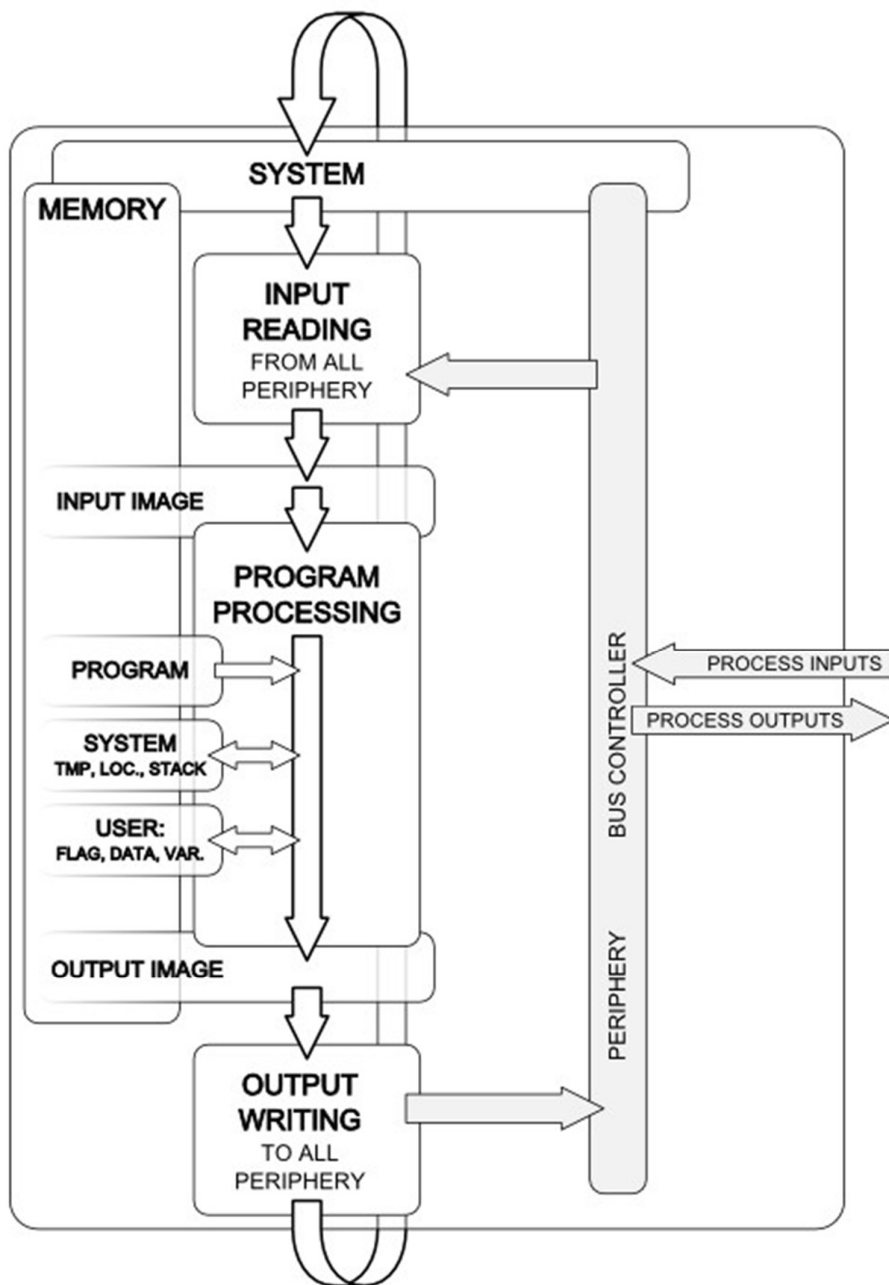
V případě modulárních PLC je systém rozdělen do modulů. Jejich výhodou je možnost přidávání dalších modulů (I/O moduly, komunikační moduly, atd...) podle velikosti aplikace s ohledem na limity daného systému. Používá se v náročnějších aplikacích, kde je například zapotřebí více vstupů či výstupů nebo přidání dalších přídatných modulů. [3]



Obr. 1-3 Blokový diagram modulárního PLC [7]

1.2.3 Princip činnosti

Základním principem PLC je práce v cyklech. V první fázi operačního cyklu jsou vzorkovány vstupní signály. Získané informace ze vstupních jednotek se ukládají do obrazů vstupů (paměti). Následuje provádění programů. To je realizováno prostřednictvím instancí. Automaty jsou schopné provádět více takových instancí najednou a tato vlastnost se nazývá multitasking. Po zpracování programu se data pro obsluhu akčních členů zapisují do obrazů výstupů (paměti) podle pořadí instrukcí. Odtud se obrazy výstupů po ukončení programu posílají do výstupních periférií. [2][4]



Obr. 1-4 Základní cyklus PLC [4]

1.2.4 Norma IEC/EN 61131

Tato norma je soubor mezinárodních standardů pro programovatelné logické automaty (PLC), které definují standardizované programovací prostředí a rozhraní. Standard je udržován Mezinárodní elektrotechnickou komisí (IEC) a je široce používán v průmyslových automatizačních a řídicích systémech. IEC/EN 61131 definuje pět programovacích jazyků, které lze použít k programování PLC: ladder diagram (LD), diagram funkčních bloků (FBD), strukturovaný text (ST), seznam instrukcí (IL) a sekvenční funkční diagram (SFC). Každý jazyk má svou vlastní syntaxi a používá se pro různé typy aplikací. [5][6]

Dělí se na několik částí. V České republice byly přijaty následující části této mezinárodní normy:

- ČSN EN 61 131-1 Programovatelné řídicí jednotky – Část 1: Všeobecné informace
- ČSN EN 61 131-2 Programovatelné řídicí jednotky – Část 2: Požadavky na zařízení a zkoušky
- ČSN EN 61 131-3 Programovatelné řídicí jednotky – Část 3: Programovací jazyky
- ČSN EN 61 131-4 Programovatelné řídicí jednotky – Část 4: Podpora uživatelů
- ČSN EN 61 131-5 Programovatelné řídicí jednotky – Část 5: Komunikace
- ČSN EN 61 131-7 Programovatelné řídicí jednotky – Část 7: Programování fuzzy zařízení [5]

1.2.4.1 Programovací jazyky

Díky části normy IEC 61 131-3 se používá standardizované programování, které zajistí sjednocení datových a programových struktur a způsobu programování nezávisle na výrobcu a typu PLC. Příkazy mohou být psány v několika typech programovacích jazyků, které se vybírají dle typu aplikace. Dělíme je na textové a grafické.

1.2.4.2 Structured Text (ST)

Jedná se o výkonný vyšší programovací jazyk, který má kořeny v jazycích Pascal a C. Použití je vhodné pro programování složitějších výpočetních algoritmů, práci s daty, řetězci a databázemi, ale také pro definování FB. Využívá se i pro zpracování analogových signálů. Naopak pro zpracování většího množství digitálních signálů není ST úplně vhodný z důvodu nepřehlednosti. Syntaxe jazyka je definována výrazy a příkazy. Příkazy jsou psány do řádku, kde jich může být více za sebou a jsou odděleny středníkem. [6]

1.2.4.3 Instruction List (IL)

Kód psaný v IL je složen z posloupností základních operací, které představují základní instrukce. Každá instrukce začíná na novém řádku. Slouží většinou pro psaní menších podprogramů, které jsou volány v hlavním programu. Tyto menší programy nejsou náročné na výkon řídicího systému. Nevýhodou může být nepřehlednost a špatná orientace v programu. [6]

1.2.4.4 Ladder Diagram (LD)

Způsob zápisu programu pomocí grafického jazyka LD vychází z reléové logiky. Jedná se o jeden z nejpoužívanějších programovacích jazyků PLC. Hlavní části jsou 2 příčky na stranách, které představují nejčastěji 0 a +24 V. Příkazy se skenují zleva doprava a program probíhá od shora dolů. Nejčastější využití je pro aplikace, kde se nachází velké množství logických signálů a operací s nimi. [6]

1.2.4.5 Function Block Diagram (FBD)

Tento grafický jazyk umožňuje programování pomocí bloků, kterými je možné realizovat různé funkce. Jednotlivé bloky jsou propojeny čarami. Vstupní proměnné se připojují k blokům z levé strany. Typ každé proměnné musí být totožný s typem připojeného vstupu.

Výstupní proměnné se připojují z pravé strany a musí odpovídat příslušnému datovému typu. Výstupem může být vnitřní nebo výstupní proměnná. Zápis je velice přehledný a vhodný pro realizaci řídicích sekcí. [6]

1.2.5 Centrální řízení

V automatizaci máme dvě možnosti řízení procesu. Centrální a decentrální. V případě centrálního řízení jde o to, že řízení vykonává samotné PLC a frekvenční měnič dostává instrukce přímo v místě aplikace a ovládá jednotlivé motory. Tento typ řízení je použit právě v této aplikaci. [9]

1.2.6 Decentrální řízení

Druhou možností řízení je decentrální. Decentralizovaný model si lze představit jako komunikační síť, přičemž každá část komunikuje přímo s ostatními částmi, které vyžadují informace. Namísto jednoho PLC, které řídí každou část stroje nebo procesu, je po celém stroji rozmístěno několik menších PLC, z nichž každé autonomně řídí, předzpracovává nebo shromažďuje data z menší části stroje. [8]

1.3 Frekvenční měnič

Důležitou součástí elektrického pohonu je frekvenční měnič. Jeho použití nám umožní přeměnu střídavé energie s počátečním kmitočtem na stejnou energii jiné frekvence. Využití nachází v řízení synchronních i asynchronních pohonů, kde je možné pohony plynule rozjíždět bez proudových nárazů a následně řídit v určitém pásmu otáček. Toto pásmo může začínat již na velmi nízkých otáčkách, jako jsou jednotky ot/min), až po velmi vysoké rychlosti v řádech desítek tisíc ot/min. Jejich využití není jen v oblasti elektrických pohonů, ale také například pro indukční ohřev. Kromě samotného řízení frekvence se v některých aplikacích požaduje současně řídit výstupní napětí nebo měnit počet výstupních fází. Dělí se na dvě základní skupiny – přímé a nepřímé. [10]

V případě přímého frekvenčního měniče se jedná o změnu vstupního kmitočtu na výstupní kmitočet přímo. To znamená, že neobsahuje stejnosměrný meziobvod. Naopak nepřímý měnič frekvence je složen z usměrňovače a střídače. Usměrňovač mění vstupní střídavou hodnotu na stejnosměrnou a střídač poté tuto hodnotu na střídavou požadované frekvence. [10]

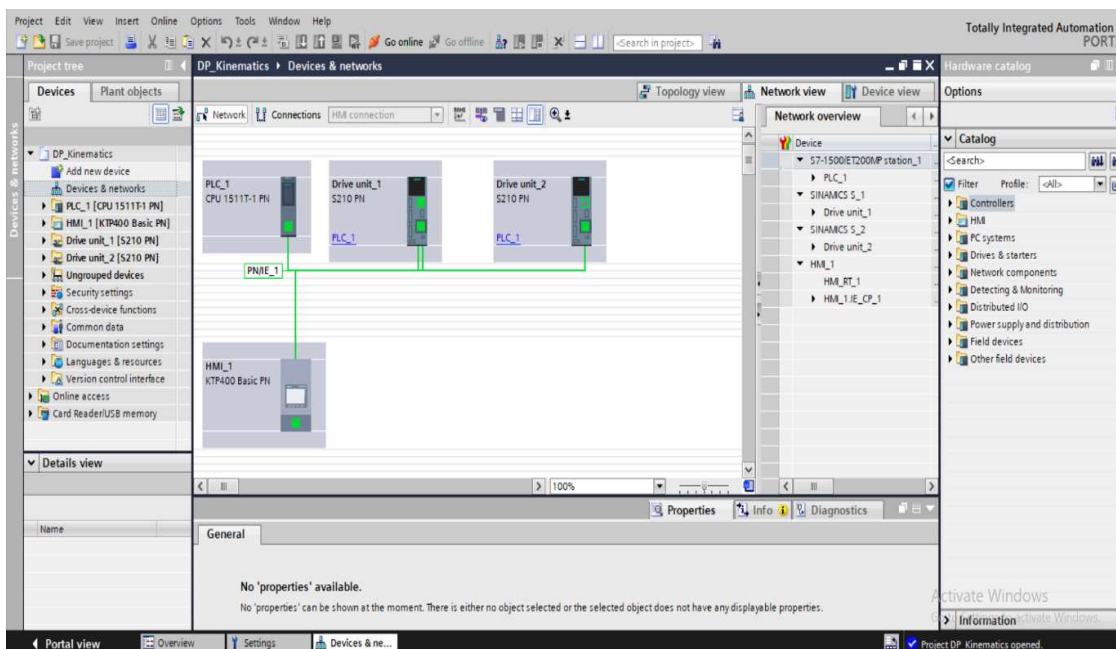
1.4 Absolutní enkodér

V automatizačních aplikacích, kde se využívají pohony, je potřeba zajistit přesné měření polohy a otáček hřídele, aby bylo možné zajistit jejich přesné řízení. Existuje mnoho metod jak tento pohyb vyhodnotit, jednou z nich je použití tzv. rotačních enkodérů. Endodéry představují elektromechanické převodníky, které převádí rotační pohyb na sekvence elektrických digitálních impulzů. Podle principu snímání se dělí na optické a magnetické. Podle poskytované informace na inkrementální (přírůstkové) a absolutní. V této aplikaci je použit absolutní enkodér, a proto se budeme věnovat pouze tomuto typu. [11]

Absolutní rotační enkodér je snímač, který pracuje s počtem otáček a zároveň umožňuje dát informaci o tom, kde přesně se v rámci otáčky (360°) v danou chvíli nachází. Při výpadku napájení tak umí navázat na informaci o své poloze a pokračovat ve snímání. Absolutní enkodéry se dělí na jednotáčkové a víceotáčkové, které poskytují unikátní informaci o poloze i v rámci více otáček. [11]

1.5 TIA Portal

TIA Portal (*Totally Integrated Automation Portal*) je softwarová platforma vyvinutá společností Siemens pro programování a konfiguraci automatizačních produktů, včetně programovatelných logických automatů (PLC), rozhraní člověk-stroj (HMI), pohonů a průmyslových komunikačních sítí. Je to výkonný nástroj, který poskytuje bezproblémové a integrované inženýrské prostředí pro všechny automatizační úlohy, od plánování a uvádění do provozu až po provoz a údržbu. Platforma TIA Portal poskytuje řadu nástrojů pro realizaci, simulaci a testování automatizačních aplikací. Nabízí uživatelsky přívětivé rozhraní, které zjednodušuje proces automatizace a zkracuje dobu projektování a uvádění do provozu. Softwarová platforma je navržena tak, aby integrovala všechny aspekty automatizační techniky do jediného prostředí, včetně programování PLC, návrhu HMI a konfigurace sítě. To umožňuje inženýrům vyvíjet a testovat své aplikace v jediném prostředí, což šetří čas a snižuje chyby. Jednou z klíčových výhod TIA Portal je jeho škálovatelnost. Platformu lze použít jak pro malé automatizační projekty, tak i pro velké a složité aplikace. Podporuje také různé programovací jazyky, což poskytuje flexibilitu pro programování a konfiguraci produktů automatizace. TIA Portal také podporuje různé komunikační protokoly, včetně PROFINET, PROFIBUS a AS-Interface, což umožňuje snadnou integraci s jinými průmyslovými protokoly a systémy. Platforma také poskytuje nástroje pro diagnostiku a údržbu, které umožňují snadnou identifikaci a opravu chyb v automatizačních systémech. [12]



Obr. 1-5 TIA Portal

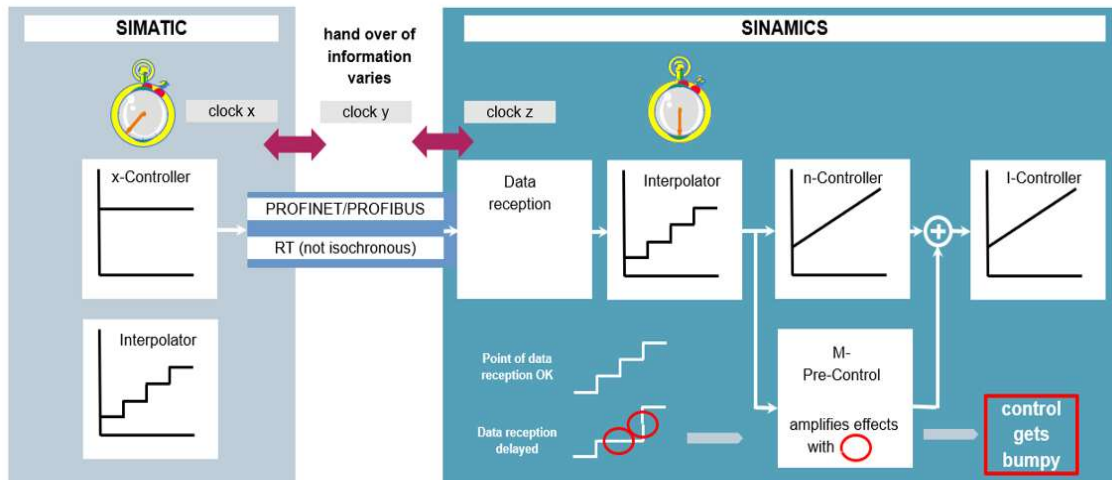
1.6 Komunikace

Komunikace mezi PLC a dalšími prvky, které si můžeme nakonfigurovat v HW konfiguraci, je zprostředkována pomocí PROFINET sítě. Výměna dat funguje pomocí cyklických přenosů.

Existují dvě možnosti těchto přenosů: RT (*Real time*), IRT (*Isochronous Real time*) a NRT (*non Real time*). V případě této práce využíváme přenos IRT. Je to z důvodu vysoké stability pro časově náročné aplikace, mezi které se řadí právě pohybové řízení (motion control). Potřebujeme tedy velmi rychlou dobu cyklu. Cyklem se rozumí vysílání a přijímání dat ve správném pořadí. Přenos pomocí NRT je acyklický typ výměny dat, který se využívá například mezi PLC a HMI.

1.6.1 RT

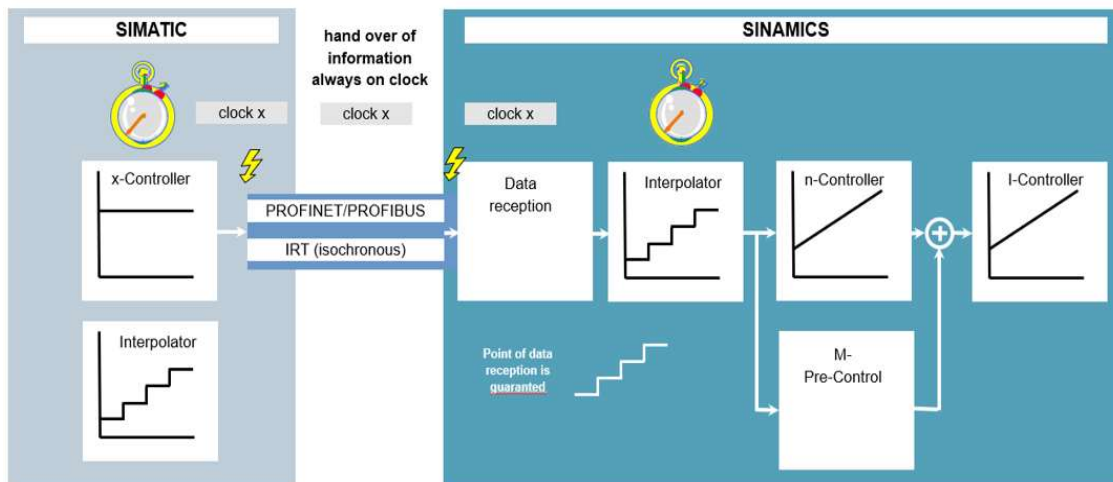
Procesní data a alarmy jsou vždy přenášena v reálném čase. Ten je v systému PROFINET založen na definicích IEEE a IEC, které umožňují pouze omezený čas pro provádění služeb v reálném čase v rámci cyklu sběrnice. Real time komunikace představuje základ pro výměnu dat v systému PROFINET. Data jsou v tomto případě zpracována s vyšší prioritou než data TCP/IP. Doba cyklu sběrnice je v tomto případě menší než 10 ms. [13]



Obr. 1-6 Komunikace RT [13]

1.6.2 IRT

Jak již bylo zmíněno, IRT přenosu je vyžadován například pro aplikace, kde se vyskytují procesy řízení polohy (synchronizace zařízení). Rozdíl oproti RT komunikaci je vysoký stupeň determinismu (začátek síťového cyklu je udržován s vysokou přesností). Cykly výměny dat jsou obvykle v rozmezí několika stovek mikrosekund až jednotky milisekund. [13]



Obr. 1-7 Komunikace IRT [13]

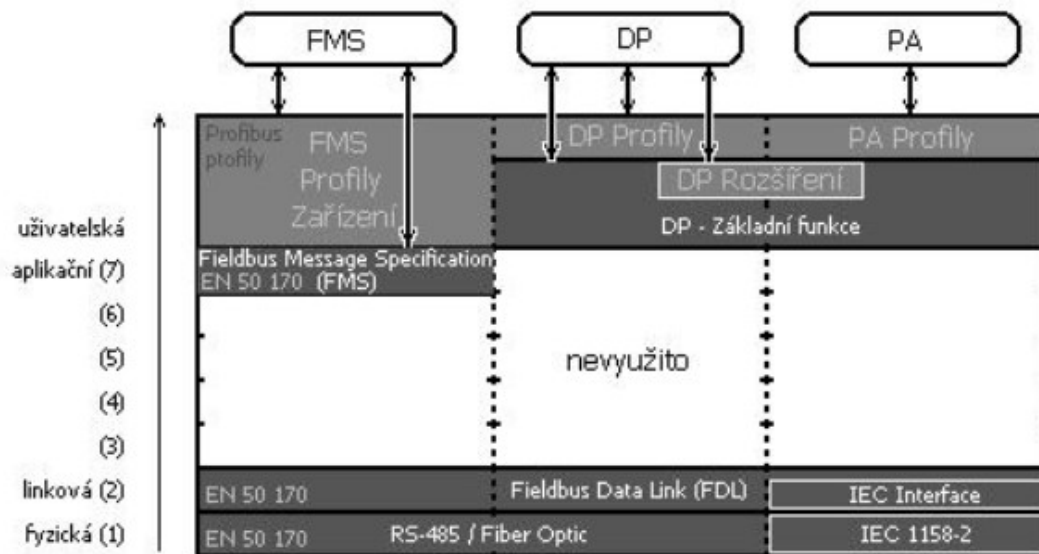
1.6.3 Ethernet

Ethernet představuje souhrn technologií, které se zabývají přenosem dat především v LAN sítích. Většina jeho verzí podléhá standardizaci institutu IEEE. Ethernet je realizací první (fyzické) a druhé (spojové) vrstvy referenčního modelu ISO/OSI. V rámci Ethernetu se jako přenosové médium nejčastěji používá kroucená dvojlinka či optický kabel, v prvotních verzích se pracovalo i s kabely koaxiálními. Nezbytnou součástí je konektor RJ-45. Existuje několik typů Ethernetu, v závislosti na druhu kabeláže a výše přenosové rychlosti. [14]

1.6.4 PROFIBUS

PROFIBUS je komunikační sběrnice, která se začala používat v roce 1987 při automatizaci výrobních linek. Provozuje se jako otevřený komunikační protokol. V dnešní době je tato sběrnice standardizována 2 normami – IEC 61158 a IEC 61784. PROFIBUS-DP a PROFIBUS-PA patří mezi nejrozšířenější variantu tohoto komunikačního protokolu. Třetí variantou je PROFIBUS-FMS, která však není tak rozšířená. Připojování k tomuto protokolu se realizuje stíněnými kabely s konektory RS 485. [15]

PROFIBUS-DP se používá pro řízení analogových a diskrétních signálů. Komunikuje přes různé fyzické vrstvy, jako je RS-485, bezdrátové připojení nebo optická vlákna. Nejvíce používanou fyzickou vrstvou je RS-485, protože dokáže přenášet data vysokou rychlostí na vzdálenost až 1200 m. PROFIBUS-PA se používá především pro automatizaci procesů a doporučuje se používat ve vnitřních prostorech. Poslední typ PROFIBUS-FMS nabízí komunikační standard pro komunikaci v heterogenním prostředí a s velkou množinou služeb pro práci s daty, programy a alarmy. Komunikačním médiem je podobně jako u varianty DP buď kroucená dvojlinka (standard RS-485) nebo optické vlákno, ale rychlost je nižší. [15]



Obr. 1-8 Model komunikační sítě PROFIBUS [16]

1.6.5 PROFINET

PROFINET je nezávislý komunikační standard pro průmyslovou automatizaci. Je to nejpokročilejší řešení propojení celých automatizačních procesů. Tento standard vychází z Ethernetové sítě s protokoly TCP/IP a IP, ale PROFINET je přímo určen pro průmyslové využití buď jako komunikace mezi stroji nebo jako komunikace mezi PLC a jeho periferiemi. PROFINET umožňuje vybudování rozsáhlejších sítí a oproti PROFIBUSu má také větší přenosovou rychlost, která dosahuje hodnoty 100 Mbit/s. Přenosová rychlost v síti PROFIBUS je omezena hodnotou 12 Mbit/s. Další velkou výhodou je, že po PROFINETU může běžet také klasická ethernetová komunikace. [17]

1.6.6 Telegramy

Výměna dat mezi PLC a dalšími zařízeními v dané síti je založena na telegramech. Telegramy obsahují informace, o adrese odesílatele a příjemce a jsou uspořádány do bloku. Telegramy Siemens se používají v různých průmyslových komunikačních protokolech, včetně PROFIBUS, PROFINET a AS-Interface. Tyto protokoly definují formát a strukturu telegramů a také pravidla pro výměnu dat mezi zařízeními. Například v PROFIBUS síti se telegramy používají k výměně dat mezi hlavním zařízením, jako je programovatelný logický kontrolér (PLC), a podřízenými zařízeními, jako jsou senzory nebo akční členy. Hlavní zařízení odešle telegram požadavku do podřízeného zařízení a podřízené zařízení odpoví telegramem obsahujícím požadovaná data.

1.6.7 Telegram 105

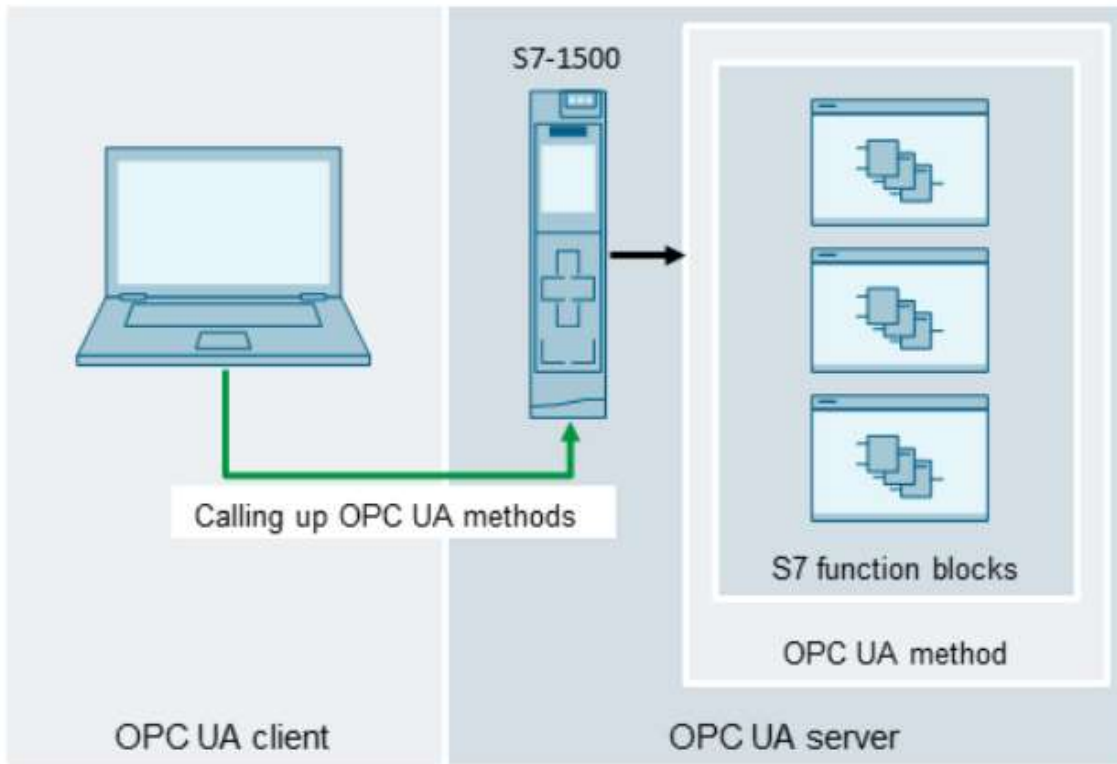
V této aplikaci pro výměnu dat používáme telegram 105. Tento typ telegramu je komunikační protokol, který je využíván v oblasti průmyslové automatizace. Ke komunikaci používá standardizovanou strukturu zpráv. Jeho klíčovou vlastností je přenos dat v reálném čase, což je pro tuto aplikaci velmi důležité z hlediska nároků na rychlost výměny dat. Samotný telegram 105 je rozšířením Telegramu 3. U centrálního řízení je důležité, že odesílaný a přijímaný signál si představíme jako signál ze strany měniče.

1.7 OPC UA

OPC Unified Architecture (OPC UA) je průmyslovým M2M (machine-to-machine) komunikačním standardem, který patří mezi relativně nové a je podporován širokou škálou výrobců PLC na svých průmyslových zařízeních, mezi které patří mimo jiné právě Siemens. Na rozdíl od původní specifikace OPC, která je založena na technologii COM/DCOM od firmy Microsoft jde o technologii založenou na obecně používaných komunikačních standardech jako jsou TCP/IP, HTTP a SOAP. OPC UA komunikaci je možné zabudovat i do vlastních PLC automatů a jiných zařízení. Na rozdíl od OPC Classic, které odděleně definuje přístup k procesním datům (OPC DA), alarmům (OPC AE) a historickým datům (OPC HDA), nové OPC UA nedefinuje tyto konkrétní přístupy, ale pouze formát předávaných zpráv. To znamená, že jeden standard OPC UA umožňuje přenášení procesních dat, alarmů a historických dat. Specifikace OPC UA je založená na předávání dat mezi OPC UA klientem a OPC UA serverem. [18]

OPC UA protokol je specifikován jako služebně orientovaná architektura SOA (Service Oriented Architecture), což znamená, že jsou definované služby, na které se klient může dotazovat a server na každý dotaz reaguje příslušnou odpovědí. Služby, které server poskytuje, vytváří abstraktní komunikační model.

Po navázání fyzického spojení se pomocí služeb vytváří a udržuje zabezpečený kanál (SecuredChannel) a relace (Session). Zabezpečený kanál je nutné mít aktivní pro veškerou komunikaci, relaci je nutné mít vytvořenou pro dotazování klienta na služby serveru. Pro to, aby byla komunikace funkční, není nutné implementovat veškeré dostupné funkce, ale stačí splnit jen nutné minimum pro provoz a další funkce je možné přidávat podle potřeby. [18]



Obr. 1-9 OPC UA server [19]

KAPITOLA 2: POPIS SESTAVY (HARDWARE)

V této kapitole je popsán veškerý hardware, který je obsažen v sestavě pro řízení plotteru. Patří do ní následující komponenty:

Název komponentu	FW	Označení
CPU SIMATIC 1511T-1 PN	2.8	6ES7511-1TK01-0AB0
SIMATIC S71500 DI/DO module	1.0	6ES7523-1BL00-0AA0
SIMATIC HMI, KTP400 Basic	17.0.0.0	6AV2 123-2DB03-0AX0
2x SINAMICS S210 PN, 0,1 kW	5.2.3	6SL3210-5HB10-1UF0
2x SIMOTICS S-1FK2 Servo motor	1.0	1FK2102-0AG00-1MA0

Tab. 2-1 Popis komponentů v sestavě

2.1 CPU SIMATIC 1511T-1 PN

SIMATIC S7-1500 je modulární systém, který je určený pro středně obtížné, až obtížné úlohy, které vyžadují maximální výkonnost, spolehlivou komunikaci, flexibilitu a technologické funkce, jako například řízení pohonů. Jak již bylo zmíněno, jedná se o modulární systém a tím pádem umožňuje připojení rozšiřujících modulů, díky kterým je možné ho optimalizovat podle náročnosti a typu aplikace. Nahrazuje SIMATIC S7-300, S7-400 a S7-1200. Mimo funkčnost těchto systémů je rozšířen o nové možnosti, např.: integrace motion control (ovládání pohonů), zvýšení výkonu, integrovaný displej (pro základní ovládání jednotky, zobrazení dostupných zařízení a diagnostiky). V našem případě používáme PLC s označením S7-1511T-1 PN. Označení „T“ je pro technologické využití. [20]. Verze FW v této aplikaci je 2.8.2.



Obr. 2-1 SIMATIC CPU 1511T-1 PN [21]

2.1.1 SIMATIC S7-1500 DI/DO module

Přesné označení modulu je: DI 16x24VDC / DQ 16x24VDC/0.5A BA. Tento modul obsahuje 16 digitálních vstupů se jmenovitou vstupní hodnotou napětí 24 VDC. Kromě digitálních vstupů obsahuje také 16 digitálních výstupů se jmenovitou výstupní hodnotou napětí 24 VDC a jmenovitou výstupní hodnotou proudu 0,5 A. Verze FW tohoto modulu je 1.0. Na obrázku níže můžeme vidět samotný modul. [20]



Obr. 2-2 DI/DO přídatný modul [22]

2.1.2 SINAMICS S210 PN

SINAMICS S210 PN je frekvenční měnič, který se používá pro řízení různých typů motorů v aplikacích, které vyžadují přesné řízení pohybu. Typicky se používá v aplikacích jako lisy nebo navíječky. [23]

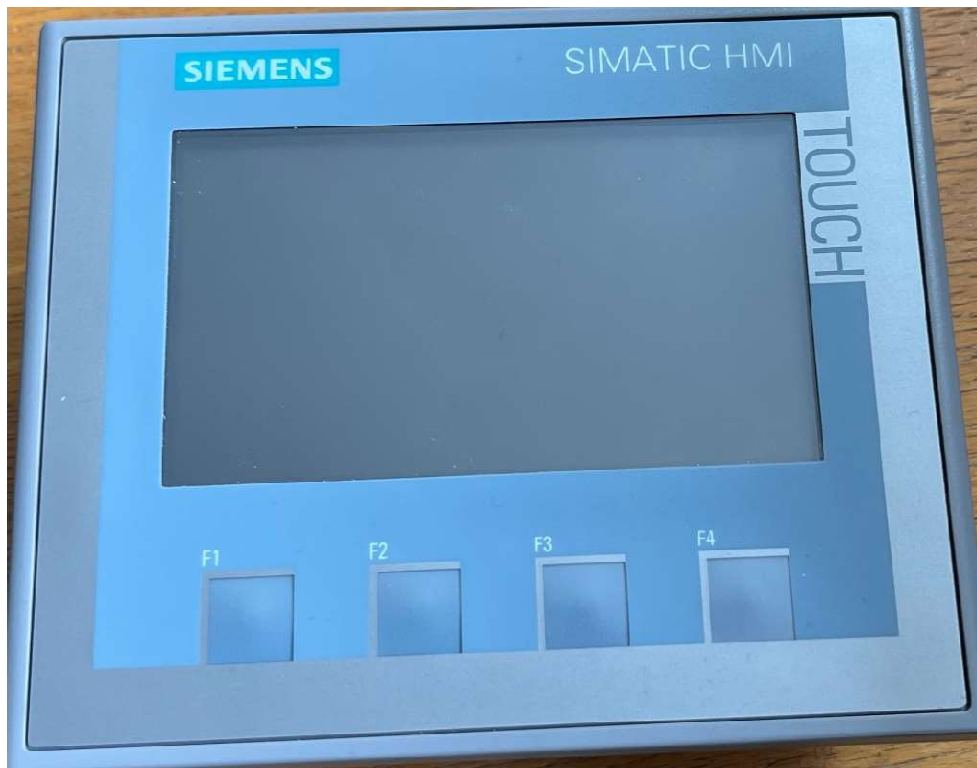
Tento typ měniče má výstupní napětí v rozmezí 200 až 240 V 1AC. Vzhledem k použitým motorům je pro tuto aplikaci zvolen měnič s výkonem 0,1 kW. Obsahuje dva rychlostní digitální vstupy pro měřicí sondy, jeden elektricky izolovaný failsafe digitální vstup pro STO (Safe Torque Off) a SS1 (Safe Stop 1) a jeden digitální vstup pro měření teploty. SINAMICS S210 PN disponuje krytím IP20. [23] Použitá verze FW je 5.2.3.



Obr. 2-3 SINAMICS S210 PN [24]

2.1.3 HMI KTP400 Basic PN

KTP400 Basic je základní panel s displejem, který umožňuje uživateli řízení a monitorování průmyslových aplikací. Samotný displej má rozlišením 480x272 pixelů a jeho velikost je 4,3". Panel obsahuje 4 funkční tlačítka, která mají označení F1-F4. Dále obsahuje jeden USB a PROFINET konektor. KTP400 Basic umožňuje vytvoření až 800 HMI tagů. [25]. Použitá verze FW je 17.0.0.0.



Obr. 2-4 HMI KTP400 Basic PN

2.1.4 SIMOTICS S-1FK2

Pro daný typ frekvenčního měniče S210 je vhodné použít motor řady 1FK2. Jedná se o synchronní motor s permanentními magnety (PNSM) s absolutním enkodérem (AM22DQC). Pro tuto aplikaci jsou použity dva motory s výkonem 0,05 kW. Jeden motor slouží pro pohyb ve směru svislé osy a druhý pro horizontální osu. [26]



Obr. 2-5 SIMOTICS S-1FK2 motor [26]

Na obrázku níže můžeme vidět jmenovité hodnoty, které jsou obsaženy na štítku motoru.

SIEMENS		
SIMOTICS 1P 1FK2102-0AG00-1MA0		
3 ~ Mot. S IR L7618 5186 01 003		
M_0 0,16 Nm	I_0 0,75 A	n_{max} 8000 /min
M_N 0,16 Nm	I_N 0,75 A	n_N 3000 /min
U_{IN} 43,0 V		Th.CI.130 (B)
m: 0,4 kg	IP 64	IC410
Encoder AM22DQC G02		
RN 000		
Siemens AG, DE-97616 Bad Neustadt Made in Romania		

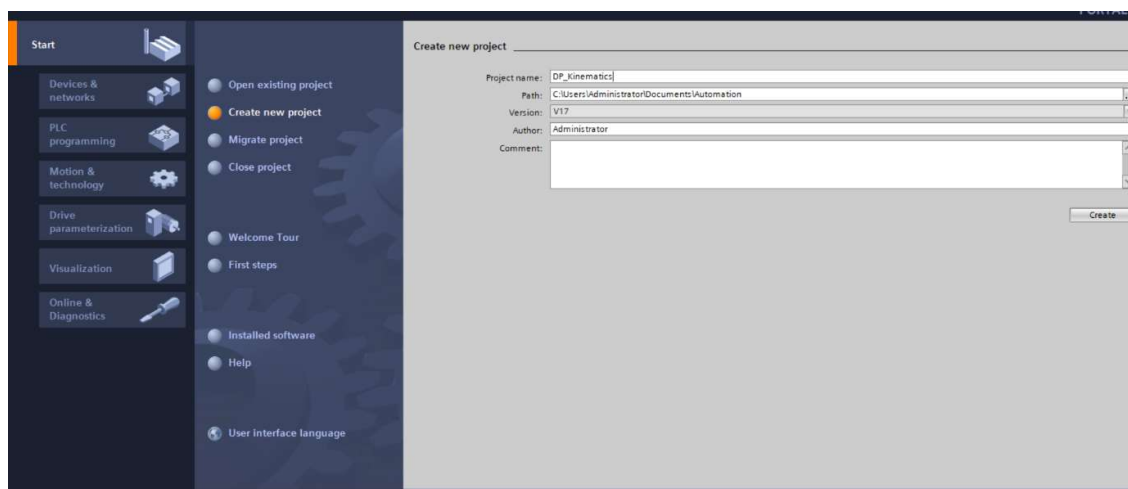
Obr. 2-6 Štítek motoru SIMOTICS S-1FK2

KAPITOLA 3: REALIZACE PROJEKTU V PROSTŘEDÍ TIA PORTAL

V této kapitole bude popsána realizace projektu v TIA portalu od samotného založení projektu přes konfiguraci hardwaru, měničů, motorů, technologických objektů a následně také fungování celé aplikace.

3.1 Založení projektu

Jako první krok celé realizace aplikace je založení projektu. Klikneme na možnost „Create new project“ a zvolíme si název projektu. Dále vybereme úložiště, na které chceme projekt uložit. Můžeme si zvolit jméno autora, případně i vložit komentář k danému projektu. Verze se vyplní automaticky podle verze TIA portalu.



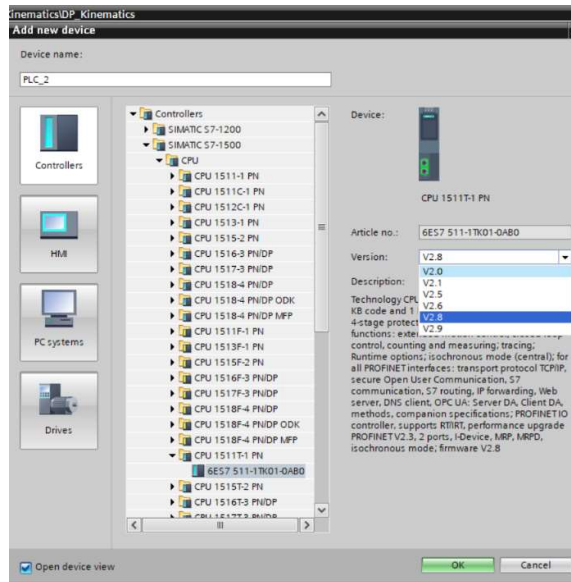
Obr. 3-1 Založení projektu v TIA Portal

3.2 Konfigurace hardwaru

Dalším krokem je HW konfigurace. Pro přidání nového zařízení do konfigurace slouží tlačítko „Add new device“, které se nachází na levé straně v okně project tree. Zobrazí se nám okno, kde si vybereme požadované zařízení, které chceme do konfigurace přidat.

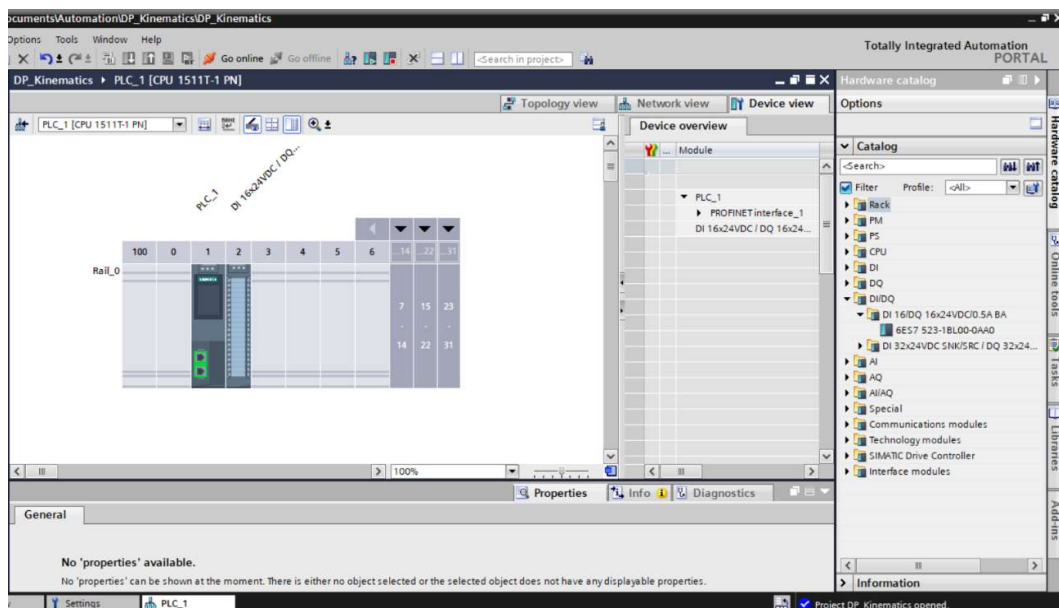
3.2.1 CPU + DI/DQ modul

Vybereme si typ daného CPU, které chceme přidat. V našem případě je to SIMATIC S7-1500 -> CPU 1511T-1 PN -> 6ES7 511-1TK01-0AB0. Zvolíme si název zařízení a verzi FW. Pro naši aplikaci jsem zvolil FW 2.8.



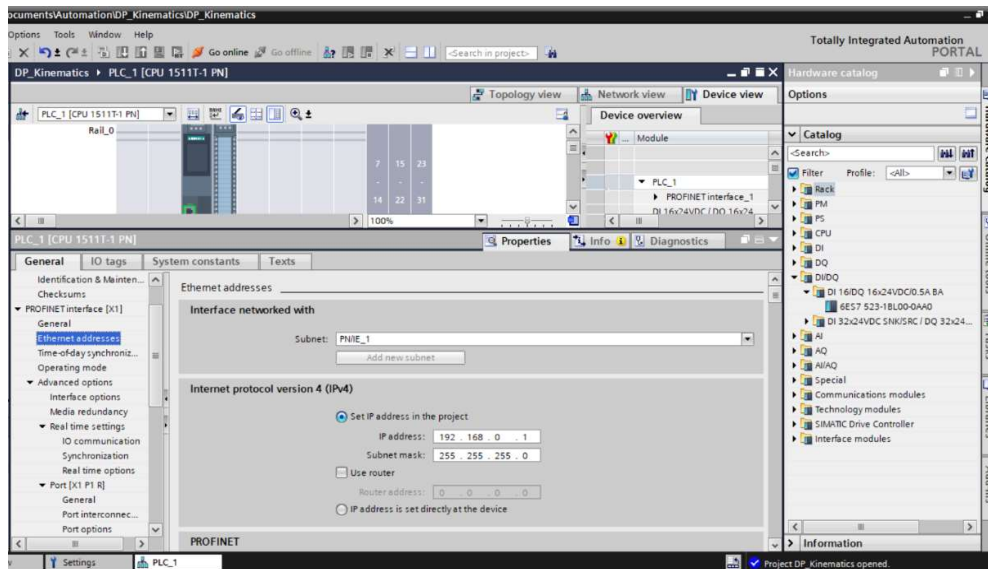
Obr. 3-2 Konfigurace DI/DQ modulu

V tuto chvíli máme zařízení přidané do konfigurace. Potřebujeme k němu ještě nadefinovat příslušný DI/DQ modul. Přejdeme do záložky „Device view“ a z pravé strany obrazovky, kde se nachází „Hardware catalog“, si vybereme příslušný modul a pomocí funkce „Drag&drop“ ho přetáhneme na pozici číslo 2 vedle CPU.



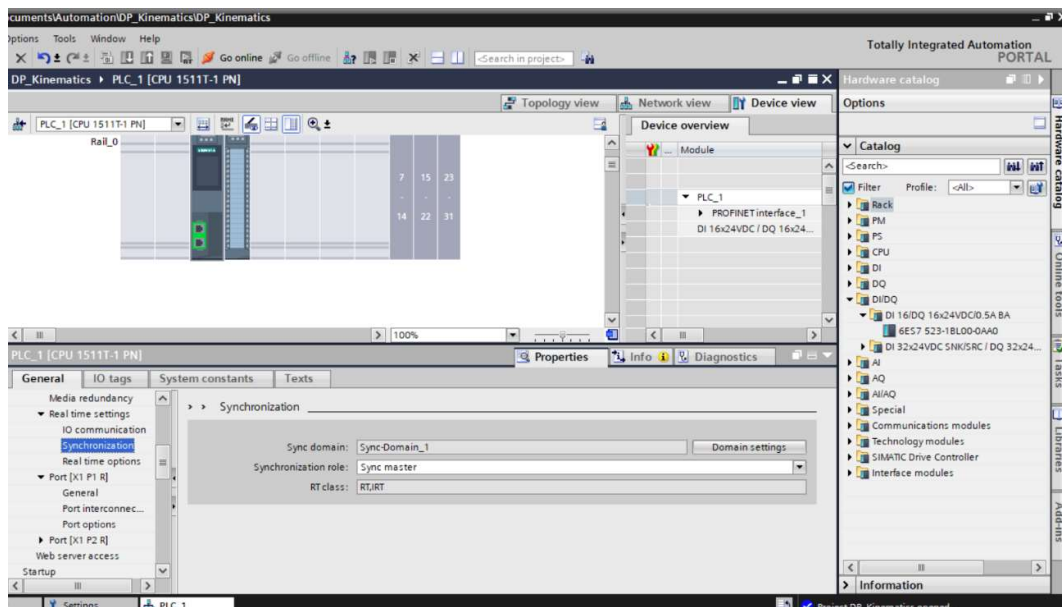
Obr. 3-3 Konfigurace CPU s DI/DQ modulem

Nyní máme přidáný CPU společně s DI/DQ modulem. Přejdeme tedy ke konfiguraci CPU. Rozklikneme si dané CPU a zaměříme se na záložku „General“. Jako první je zapotřebí nadefinovat IP adresu, která se nachází v rolovací záložce „PROFINET interface [X1]“ -> „Ethernet addresses“. IP adresa v našem projektu je: 192.168.0.1. Zároveň ve stejné záložce zvolíme PN/IE_1 jako subnet.



Obr. 3-4 Nastavení IP adresy pro CPU

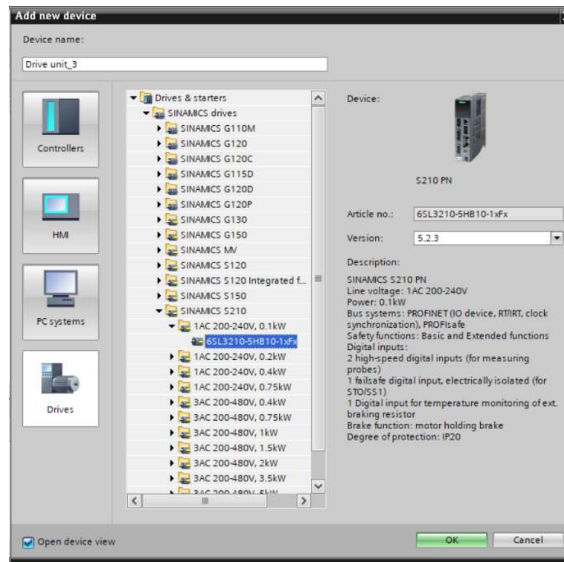
Další záložka, která nás bude zajímat, je „Real time settings“ a v ní položka „Synchronization“. Zde se nastaví „Synchronization role“ jako „Sync master“.



Obr. 3-5 Real Time settings CPU

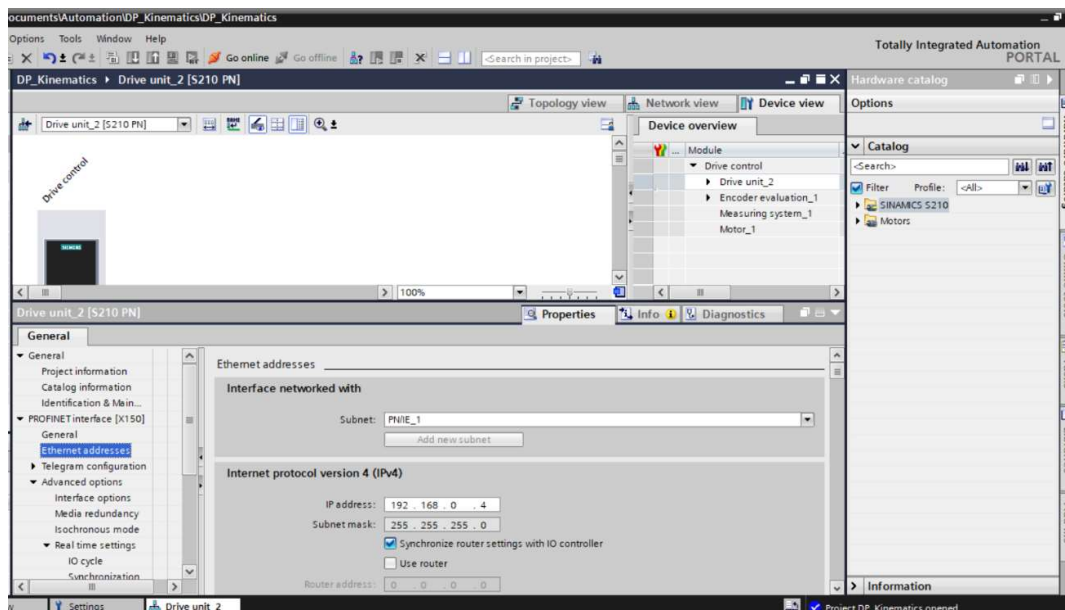
3.2.2 Frekvenční měniče

Jako další si přidáme do konfigurace frekvenční měniče. Stejně jako v předchozím případě si zvolíme „Add new device“, ale tentokrát se přesuneme do záložky „Drives“. Zde si najdeme náš typ měniče, zvolíme název a FW 5.2.3.



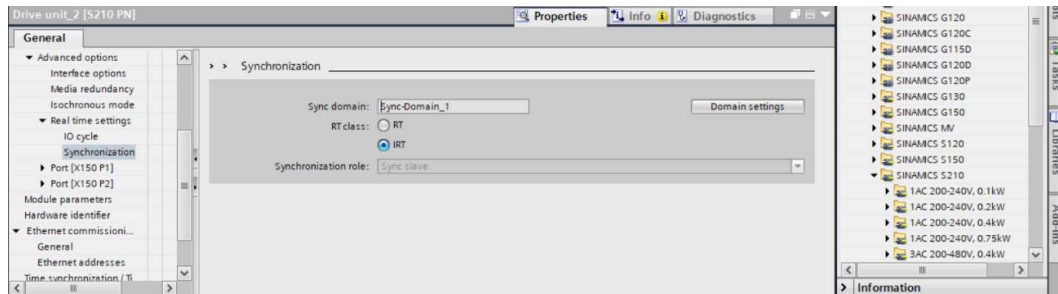
Obr. 3-6 Konfigurace frekvenčního měniče

Po přidání obou měničů přejdeme opět k definování IP adres. Opět si v „Device view“ najdeme záložku „PROFINET interface [150]“ -> „Ethernet addresses“. V našem případě je IP adresa měničů 192.168.0.3 respektive 192.168.0.4. Ve stejné záložce také zvolíme subnet jako PN/IE_1.



Obr. 3-7 Nastavení IP adresy pro frekvenční měnič

U konfigurace měničů je znovu důležité nastavit komunikaci. Přejdeme do záložky „Advanced options“ -> „Isochronous mode“. Zde zaškrtneme možnost „Isochronous mode“. Dále v záložce „Real time settings“ -> „Synchronization“ je zapotřebí nastavit RT class jako IRT. Ve stejné záložce je nutné u obou měničů zvolit stejnou doménu („Sync domain“), jako je u CPU, aby spolu zařízení mohla komunikovat.

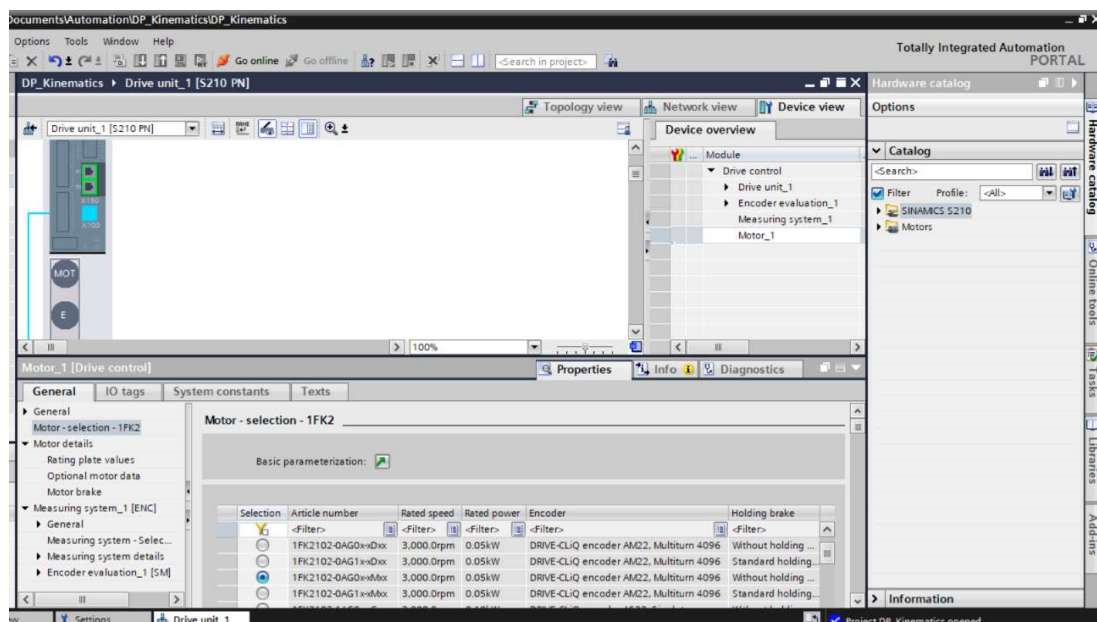


Obr. 3-8 Nastavení synchronizace frekvenčního měniče

3.2.3 Motory

Frekvenční měniče jsou již nakonfigurovány, nyní je potřeba ke každému definovat motor. Jak již bylo zmíněno, vybereme motor s označením 1FK2102-0AG00-1MA0. V záložce „Device view“ u měniče vybereme tento typ v „General“ -> „Motor-selection“ a přiřadíme. Analogicky u druhého měniče.

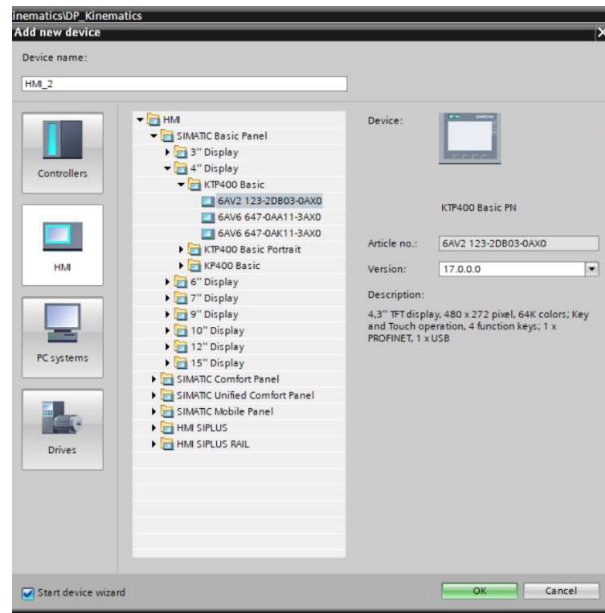
Jako další krok je nutné naladit regulační blok obou os. TIA Portal umožňuje automatické naladění konstant regulátorů pomocí funkce *One Button Tuning*. Tato funkce musí být provedena pro oba pohony. Pro její provedení slouží záložka *commissioning*, kde se zmíněná funkce nachází. Automatické ladění se poté provede automaticky spuštěním tlačítka *Activate* a poté *Start*. Analogicky i pro druhý pohon.



Obr. 3-9 Přiřazení motoru k frekvenčnímu měniči

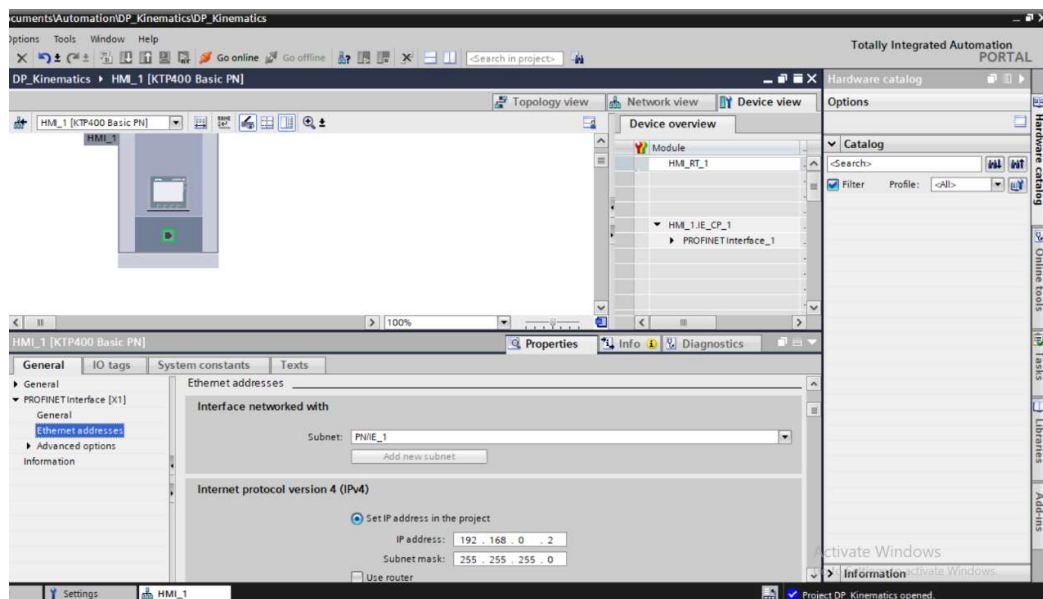
3.2.4 HMI

Jako poslední zařízení zbývá nakonfigurovat HMI panel. Opět přejdeme do „Add new device“ a přesuneme se do záložky „HMI“. Zde vybereme požadovaný typ, pojmenujeme a zvolíme FW. V našem případě je zvolen FW 17.0.0.0.



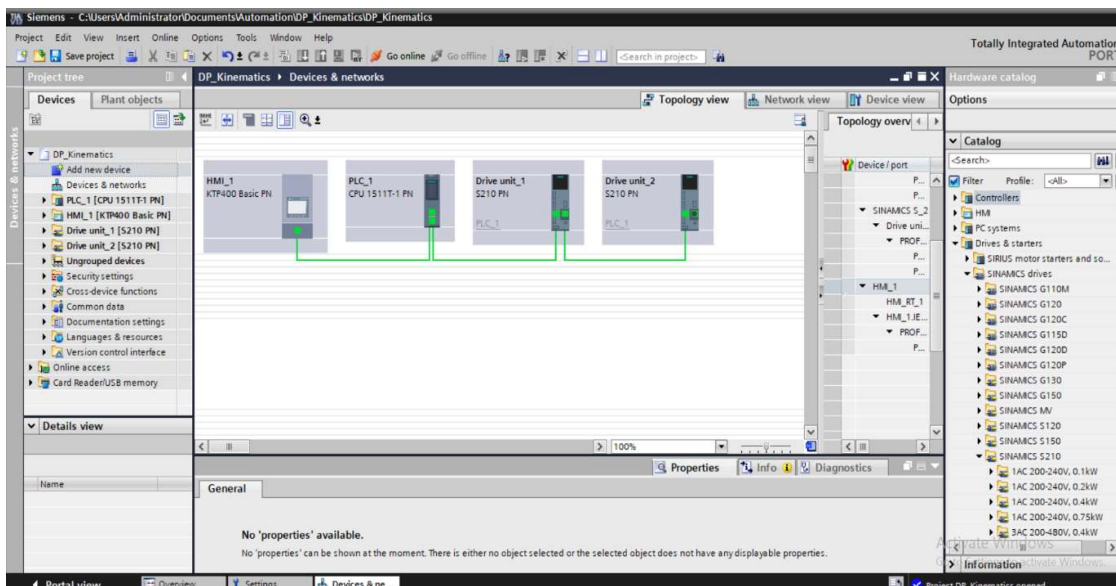
Obr. 3-10 Konfigurace HMI panelu

Stejně jako u předchozích zařízení musíme definovat IP adresu a název zařízení. Ta je pro HMI 192.168.0.2.



Obr. 3-11 Nastavení IP adresy pro HMI panel

Když máme nakonfigurována všechna potřebná zařízení, přejdeme do záložky „Topology view“. Zde je potřeba zařízení pospojovat tak, jak jsou ve skutečnosti propojena spolu. Každé zařízení kromě HMI panelu obsahuje dva konektory. Podle skutečného zapojení si zařízení propojíme přes jednotlivé porty. To je nutné udělat pro isochronní mód, jelikož CPU posílá konkrétní data na konkrétního partnera. Výsledná konfigurace vypadá následovně:



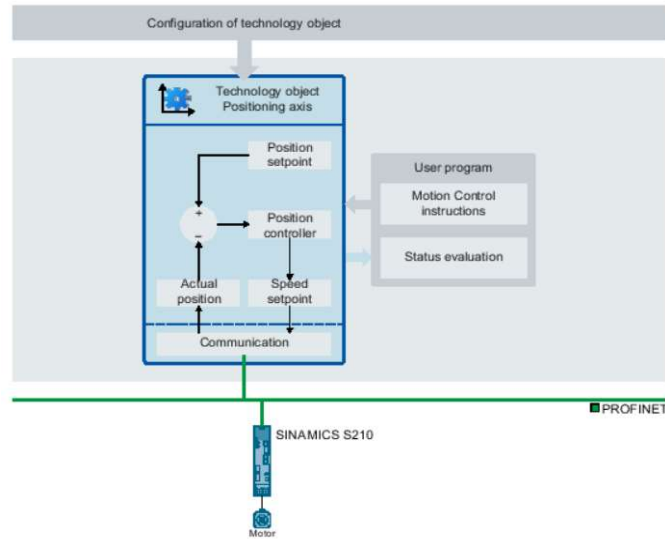
Obr. 3-12 HW konfigurace

3.3 Technologické objekty

V aplikaci jsou použité tři technologické objekty. Dva z nich tvoří polohové osy („TO_PositioningAxis“) a třetím objektem je „TO_Kinematics“.

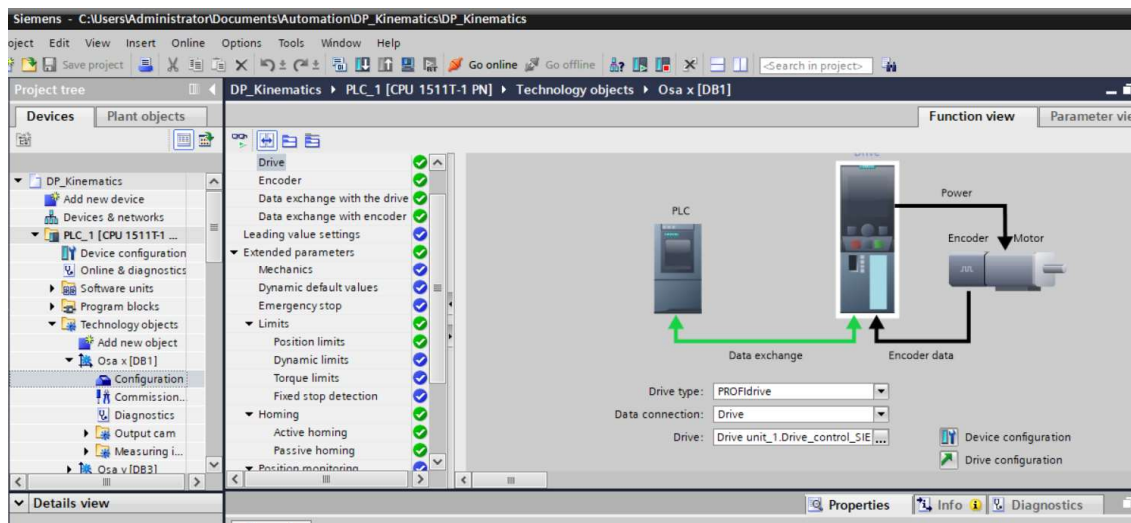
3.3.1 Osa x

Prvním technologickým objektem je polohovací osa „TO_PositioningAxis“, která představuje osu x – tedy pohyb pera. Technologický objekt polohové osy vypočítává požadované hodnoty polohy a odesílá odpovídající požadované hodnoty rychlosti do měniče. V polohově řízeném režimu probíhají všechny pohyby polohovací osy jako polohově řízené pohyby. Pro absolutní polohování musí být technologickému objektu polohovací osy známa fyzická poloha.



Obr. 3-13 Princip TO_PositioningAxis [27]

Pro vložení technologického objektu přejdeme na záložku „Technology objects“ a přidáme nový. Zde vybereme polohovou osu, pojmenujeme a přidáme. Nyní přichází na řadu její konfigurace. Jako první je potřeba daný technologický objekt přiřadit k příslušnému meniči.



Obr. 3-14 Přiřazení TO_PositioningAxis k frekvenčnímu měniči

Dále se aktivuje enkodér a přiřadí se opět k danému měniči, ke kterému je technologický objekt spjat. Pro výměnu dat mezi objektem a drivem respektive enkodérem použijeme již zmiňovaný Telegram 105.

Pro technologické objekty je k dispozici velké množství různých parametrů, které se dají nastavit. Patří mezi ně například dynamické limity pro nouzové zastavení, limity pro rychlost, zrychlení, brzdění, atd... Dále je možnost nastavení pozičních limitů jak hardwarových, kde se využívá například digitálních vstupů z koncových čidel, tak softwarových. Pro polohové osy lze nastavit také aktivní, či pasivní homing.

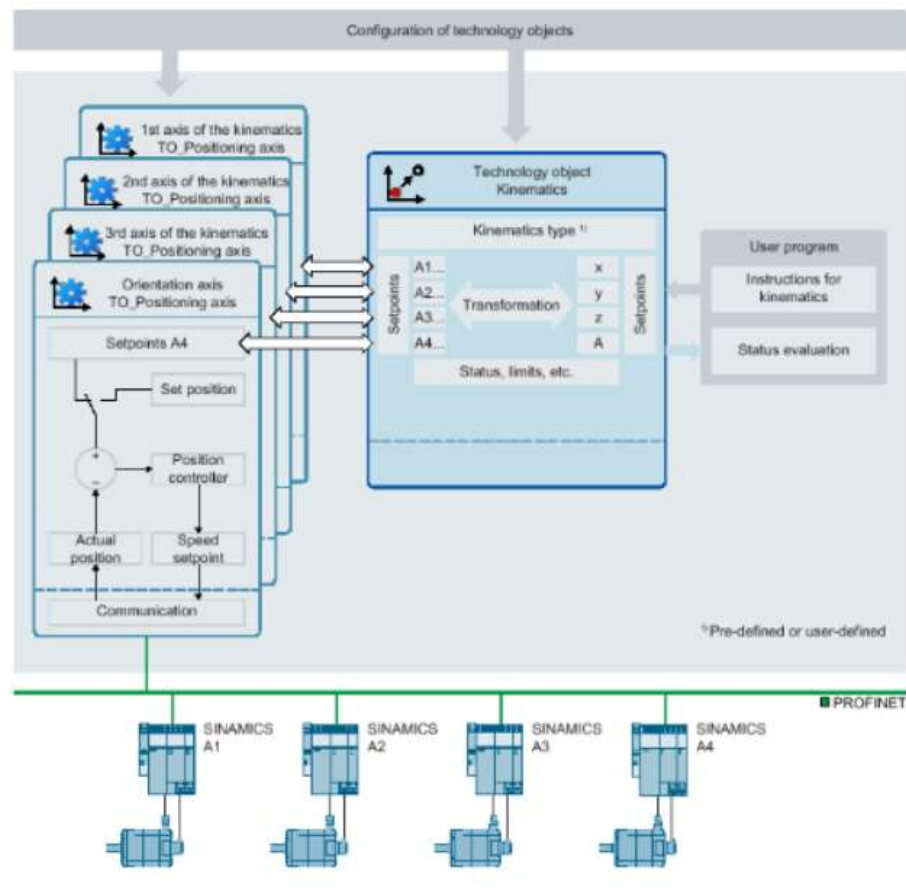
TIA Portal disponuje funkcí commissioning, díky které můžeme jezdit s danou osou doleva či doprava libovolnou rychlostí, kterou lze nastavit přes setpoint „Velocity setpoint“.

3.3.2 Osa y

Dalším technologickým objektem je opět polohovací osa „TO_PositioningAxis“, která však nyní představuje osu y – tedy pohyb papíru nahoru a dolů. Přidání objektu a následná konfigurace probíhá analogicky jako u osy x. Tedy přiřazení k příslušnému drivu, nastavení enkodéru, telegramu, atd...

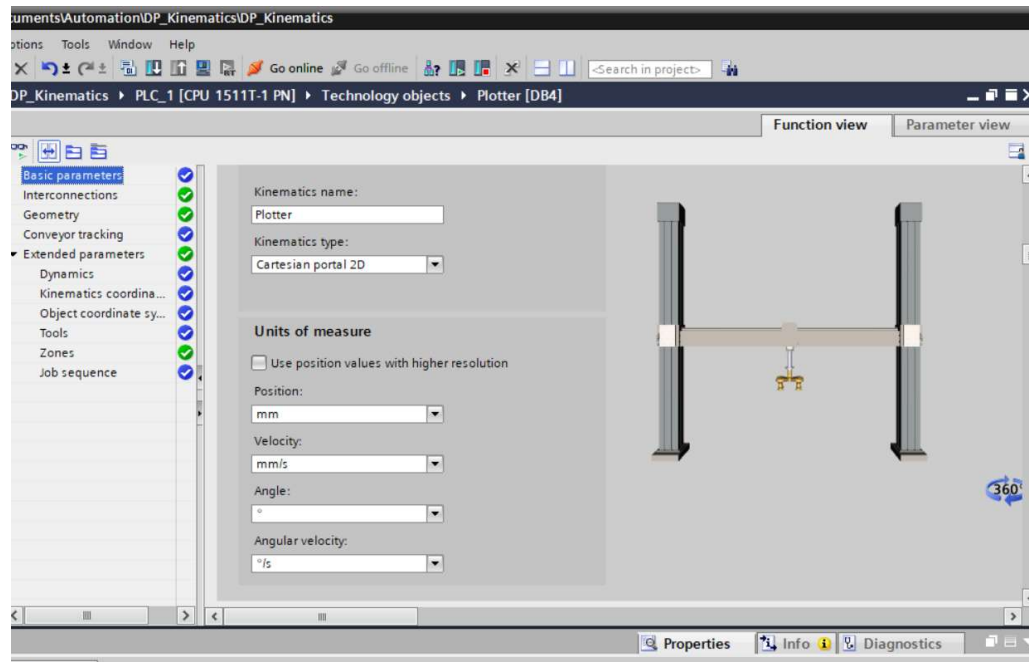
3.3.3 Plotter

Posledním technologickým objektem je „TO_Kinematics“, který je pojmenován jako Plotter. Vypočítává požadované hodnoty pohybu pro jednotlivé osy kinematiky z aktuálních hodnot os pomocí kinematické transformace. Objekt kinematické technologie odesílá požadované hodnoty pohybu specifické pro osu do propojených polohovacích os. Jednotlivé osy pro daný objekt mohou být polohové nebo synchronní. Umožňuje propojení až čtyř os.



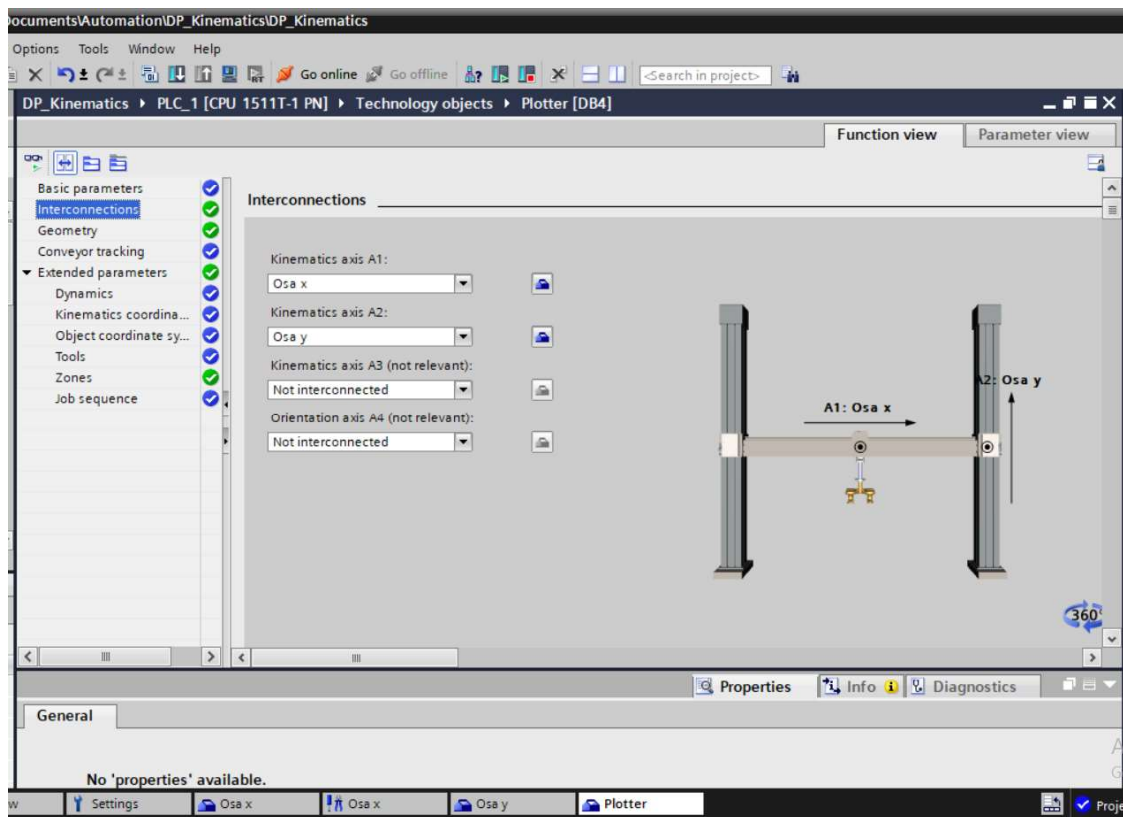
Obr. 3-15 Princip TO_Kinematics [28]

Pro vložení tohoto objektu opět přejdeme na záložku „Technology objects“, vybereme „TO_Kinematics“, pojmenujeme a vložíme. Po vložení je potřeba technologický objekt nakonfigurovat. Jako první si zadefinujeme typ kinematismu. Pro naši aplikaci je vhodné zvolit „Cartesian portal 2D“, jelikož ovládáme 2 osy (x a y). Dále je možné nastavit jednotky pro pozici, rychlost, úhel a úhlovou rychlost.



Obr. 3-16 Konfigurace TO_Kinematics

V záložce „Interconnections“ zadefinujeme propojení os ve správném směru. Tedy kinematická osa A1 odpovídá ose x a kinematická osa A2 odpovídá ose y.



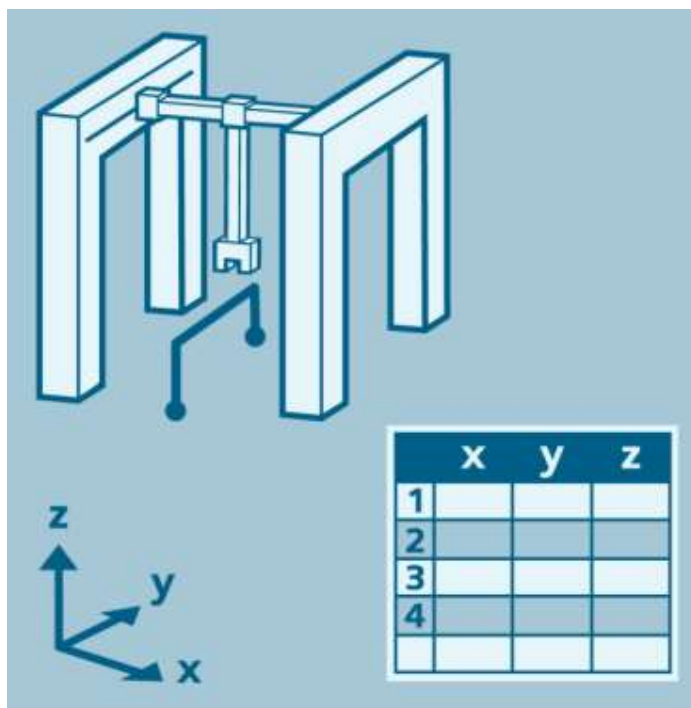
Obr. 3-17 Nastavení os pro TO_Kinematics

3.4 Software

V této podkapitole se zaměříme na samotou funkcionalitu aplikace, která je realizována v prostředí TIA Portal. Jak již bylo zmíněno, k programování v tomto vývojovém prostředí, máme na výběr ze široké škály programovacích jazyků. V této aplikaci se převážně vyskytují dva programovací jazyky – Ladder Diagram (LD) a Function Block Diagram (FBD). Právě tyto jazyky jsem zvolil z toho důvodu, jelikož se k programování aplikací podobného charakteru hodí nejvíce. Zaměříme se na popis funkčních bloků, funkcí, PLC tagů a také na HMI panel a jeho vlastnosti. Pro některé dílčí funkce byla použita knihovna „Kinematics Control“. Celá aplikace se skládá z hlavního organizačního bloku „Main“, kde jsou vykonávány hlavní procesy celého programu. Obsahuje ještě další organizační blok „Drawing“, který slouží pro vykonávání logiky kreslení. Dále se zde nachází několik funkčních bloků a jim příslušných datových bloků, PLC tagů a HMI tagů. Všechny tyto komponenty budou popsány právě v této kapitole.

3.4.1 Library Kinematics Control

Library Kinematics Control, zkráceně „*LkinCtrl*“, je knihovna, která poskytuje funkce pro technologický objekt „TO Kinematics“ a tím zajišťuje programování a řízení pohybů os, které jsou na daný objekt implementovány. Popíšeme si zde všechny její funkce a funkční bloky, které jsou v aplikaci použity.



Obr. 3-18 Ovládání kinematiky [29]

3.4.2 PLC tagy

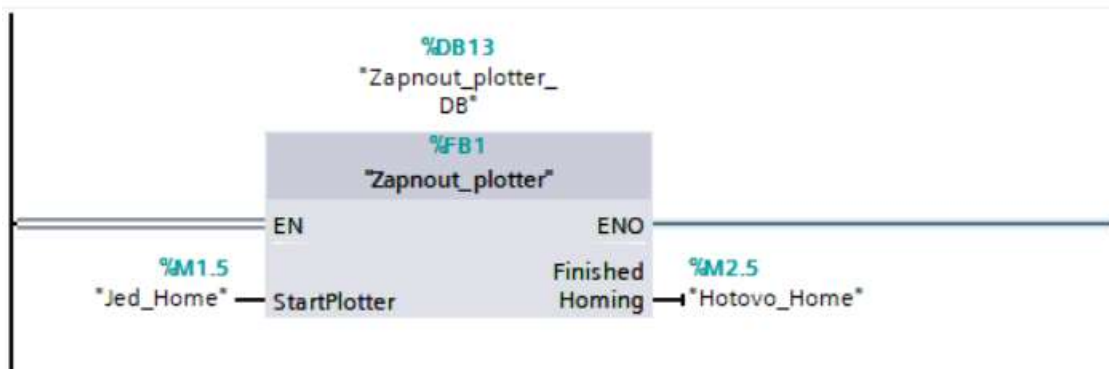
PLC tagy v TIA portálu slouží pro práci se signály a informacemi v celém programu. Při definici PLC tagu je nutné definovat jeho datový typ. Takto nadefinovaný tag může sloužit jako fyzický vstup nebo výstup, konstanta a nebo pouze jako paměťový signál. V případě fyzického vstupu ze vstupně/výstupního modulu je tvar %I.x.x, kde číslice za označením vstupu slouží pro adresu. Stejně tak je tomu u výstupů, kde je tvar %Q.x.x. Paměťové signály se označují %M.x.x. Po definici tagů je můžeme použít v jednotlivých částech programu. Rozdíl mezi tagy a daty z datového bloku je ten, že tagy vytvořené v tabulce patří ke globálním hardwarovým prostředkům a standardně neuchovávají paměť. Globální datové bloky jsou softwarově přiřazené, mají více funkcí a uchovávají paměť.

3.4.3 Main

Hlavní organizační blok obsahuje celkem deset sítí („Networks“), kterými se rozumí rozdělení daného bloku pro přehlednost. Jednotlivé sítě mohou mít svůj název a popis, aby pro uživatele co byl program co nejvíce přehledný. V jednotlivých sítích se nachází následující bloky:

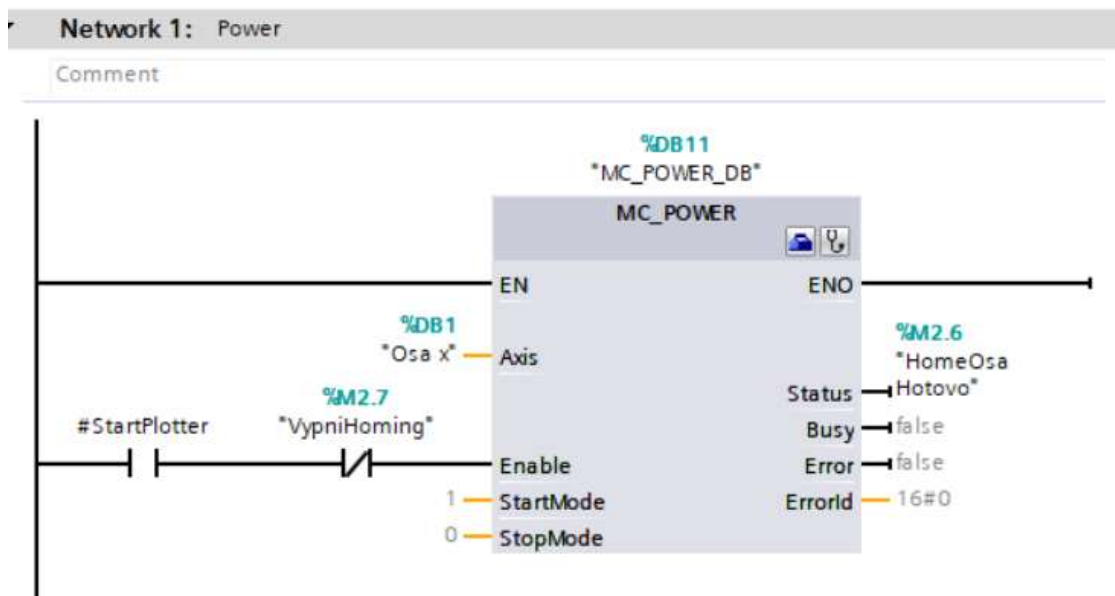
3.4.3.1 FB Zapnout_plotter

Prvním funkčním blokem, který se nachází v organizačním bloku „Main“, je vytvořený blok „Zapnout_plotter“. Obsahuje jeden vstup a výstup. Na vstup „StartPlotter“ je připojený přepínač kterým se plotter spouští a vypíná. Výstup „FinishedHoming“ se aktivuje ve chvíli, kdy osa x sepne koncový spínač.



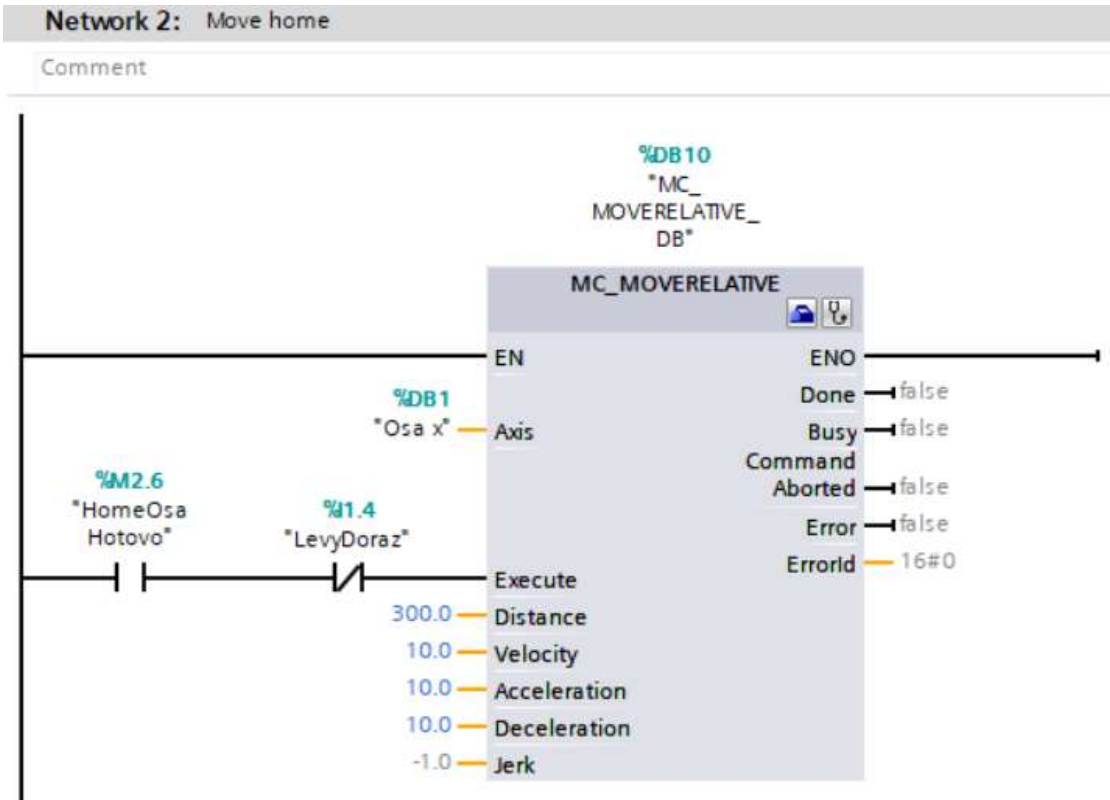
Obr. 3-19 FB Zapnout_plotter

Při stisknutí tlačítka, kterým se celý plotter zapíná, dojde k aktivování osy x pomocí instrukce „MC_POWER“. To, že je osa x aktivní, signalizuje výstup „Status“, jehož hodnota se přeplojí do stavu „TRUE“.



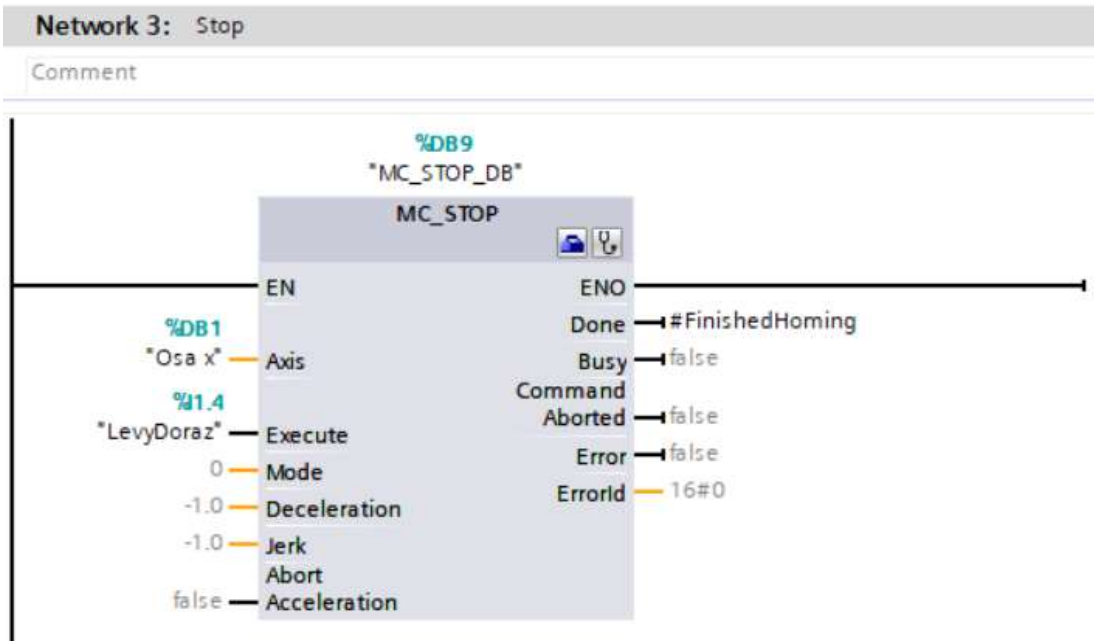
Obr. 3-20 FB Zapnout_plotter – aktivace osy x

Tento výstup následně provede další instrukci „MC_MOVERELATIVE“, která pošle osu x směrem doleva na koncový spínač.



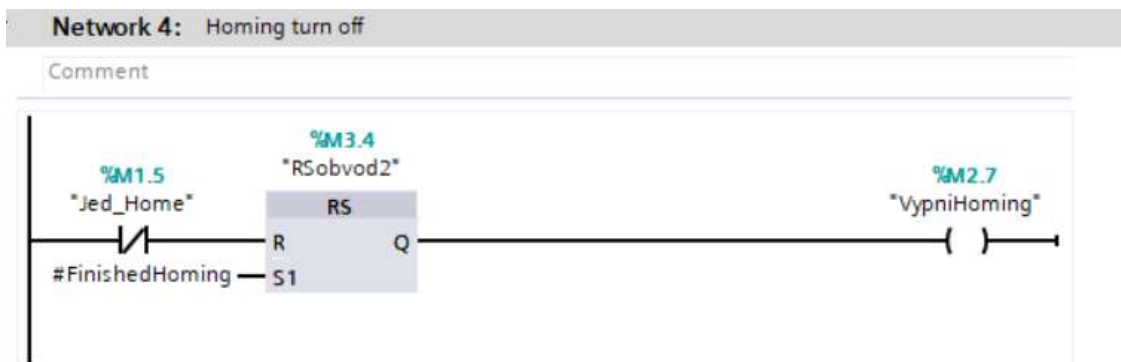
Obr. 3-21 FB Zapnout_plotter – pohyb osy na koncový spínač

V tuto chvíli se osa x pohybuje směrem doleva na koncový spínač do té doby, dokud na něho nenarazí. Sepnutím fyzického vstupu pro tento spínač „Levy_doraz“ dojde k vyvolání instrukce „MC_STOP“, která osu x deaktivuje. Vypnutí této osy je signalizováno výstupem „Done“, na který je připojen výstup celého funkčního bloku „Zapnout_plotter“.



Obr. 3-22 FB Zapnout_plotter – zastavení osy x

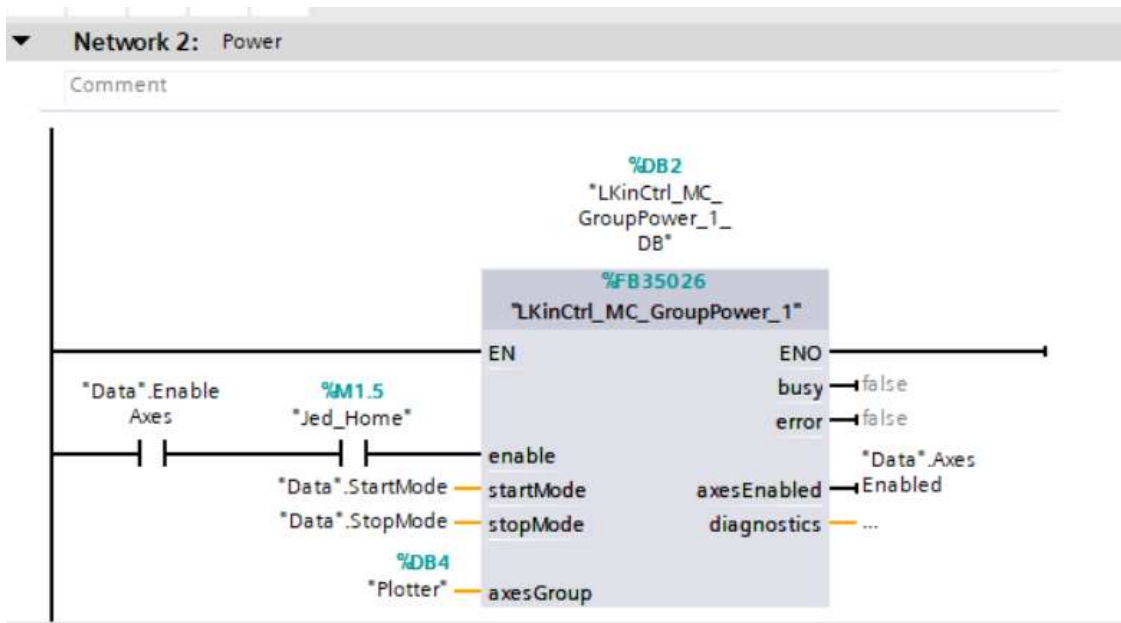
Aby mohlo docházet k zapínání a vypínání plotteru cyklicky, má tento funkční blok ještě jednu síť, která obsahuje RS klopný obvod. Ten zajišťuje právě přepínání mezi těmito dvěma stavy.



Obr. 3-23 FB Zapnout_plotter – zapínání a vypínání bloku

3.4.3.2 FB LkinCtrl_MC_GroupPower

Jako další se zde nachází funkční blok, který slouží pro zapnutí či vypnutí všech propojených os v dané skupině („axesGroup“). V této aplikaci je pro nás skupinou os náš technologický objekt Plotter, ve kterém jsou definovány právě naše dvě osy – osa x, y. Tento blok disponuje následujícími vstupy a výstupy:



Obr. 3-24 Funkční blok LKinCtrl_MC_GroupPower

Enable: Tento vstup slouží pro aktivaci daného bloku a je datového typu „BOOL“. K němu je připojen kontakt, kterým se aktivuje technologický objekt skrz tlačítko na HMI panelu nebo přes OPC UA server. Sériově je ještě připojen kontakt „Jed_Home“.

StartMode: Další vstup slouží pro zvolení režimu osy. Datovým typem je pole, které má rozměr podle počtu os, které jsou nadefinovány k danému technologickému objektu – Plotter. Možné režimy jsou dva. 0 – vypnuté poziční řízení. 1 – zapnuté poziční řízení. Hodnotu tohoto vstupu jsem nastavil jako 1, tedy zapnutí pozičního řízení.

StopMode: Předposledním vstupem, který daný blok obsahuje, je blok StopMode. Jeho konfigurací nastavujeme režim zastavení. Jako v předchozím případě se opět jedná o pole s rozměrem dle počtu os. Tentokrát jsou režimy tři. 0 – nouzové zastavení, 1 – okamžité zastavení a 2 – zastavení s maximální dynamikou. V této aplikaci je používán režim pro okamžité zastavení.

AxesGroup: Na poslední vstup se, jak název napovídá, připojuje daný technologický objekt, který obsahuje skupinu os, jež jsou na něj napojeny. V této aplikaci je název tohoto technologického objektu „Plotter“. Jeho konfiguraci můžeme nalézt v předchozí kapitole.

Busy: První výstup slouží pro signalizaci toho, že daný blok je právě vykonáván, což je znázorněno jako stav „TRUE“. Datový typ v tomto případě je „BOOL“. Pro účely této aplikace jsem daný výstup nenašel žádné uplatnění, proto zůstává nenakonfigurován.

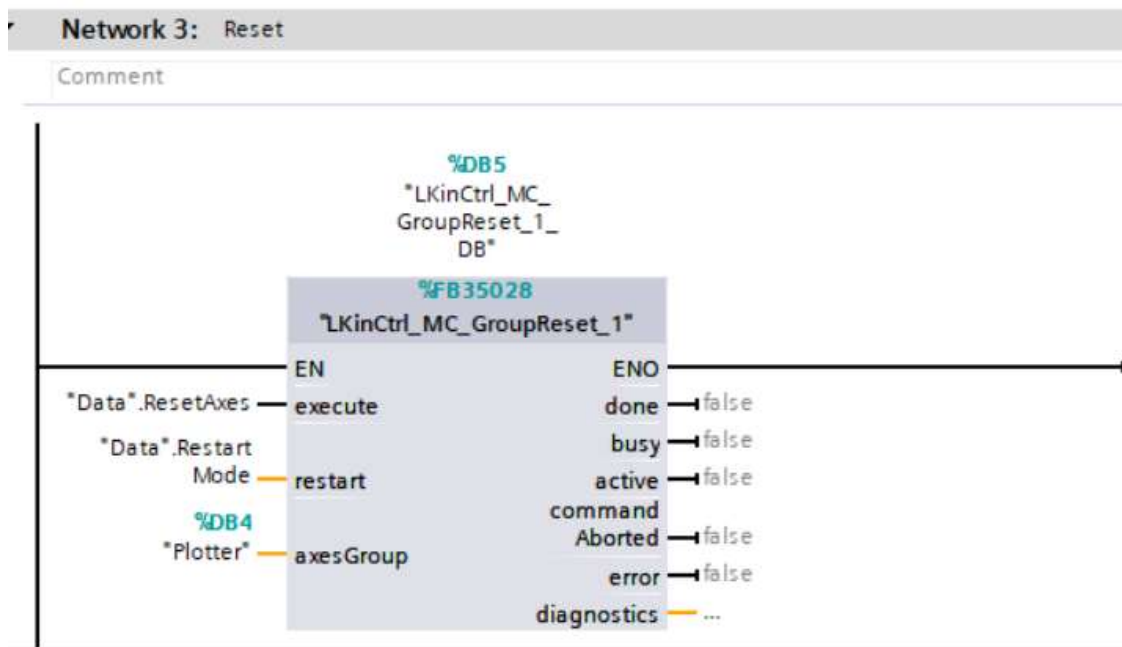
Error: Dalším výstupem stejného datového typu je Error. Ten slouží k tomu, aby uživateli dal informaci o tom, že došlo k chybě během provádění tohoto funkčního bloku. Opět jsem tento výstup nechal nenakonfigurován, jelikož hlášení errorů je vyřešeno jinak a bude popsáno později.

AxesEnabled: Poslední výstup je opět datovým typem „BOOL“. Slouží pro indikaci stavu úspěšné aktivace připojených os, které jsou tak připraveny pro další použití.

Diagnostics: Tento výstup je spíše informativního rázu a v případě aktivace výstupu Error, napíše chybovou hlášku, která lze poté vyhledat v seznamu, kde se nachází všechny možné kódy pro error („ERROR_ID“). Vyskytuje se ve většině funkčních bloků, proto již u dalších nebude popisován, jelikož funkce zůstává stejná, mění se pouze kód.

3.4.3.3 FB LkinCtrl_MC_GroupReset

Dalším funkčním blokem je blok, který slouží pro resetování nakonfigurované skupiny os. Používá se v případě, kdy se objeví nějaký error na osách nebo pokud by bylo potřeba nastavit novou výchozí pozici.



Obr. 3-25 Funkční blok LKinCtrl_MC_GroupReset

Execute: První vstup funkčního bloku slouží pro jeho aktivování náběžnou hranou. Jeho datovým typem je „BOOL“ a je k němu připojen kontakt, kterým se právě tento funkční blok aktivuje ať už přes tlačítko na HMI panelu nebo přes OPC UA server.

Restart: Dalším vstupem se volí režim restart. Datovým typem v tomto případě je pole datového typu *BOOL* s rozměrem [1..5]. Stav „1“ slouží pro režim s restartem a „0“ pro režim bez restartu.

AxesGroup: Poslední vstup pro připojení technologického objektu funguje stejně jako u předchozího funkčního bloku.

Done: První výstup tohoto funkčního bloku se používá pro signalizaci toho, že byl reset vykonán. V této aplikaci není nakonfigurován, protože jeho funkce zde dle mého názoru není nutná. Může se využít v případech rozsáhlejších aplikací pro monitorování stavu linky.

Busy: Dalším výstupem je signalizován stav, kdy je daný funkční blok vykonáván a ještě není dokončen. Opět zde zůstává tento výstup nenakonfigurován.

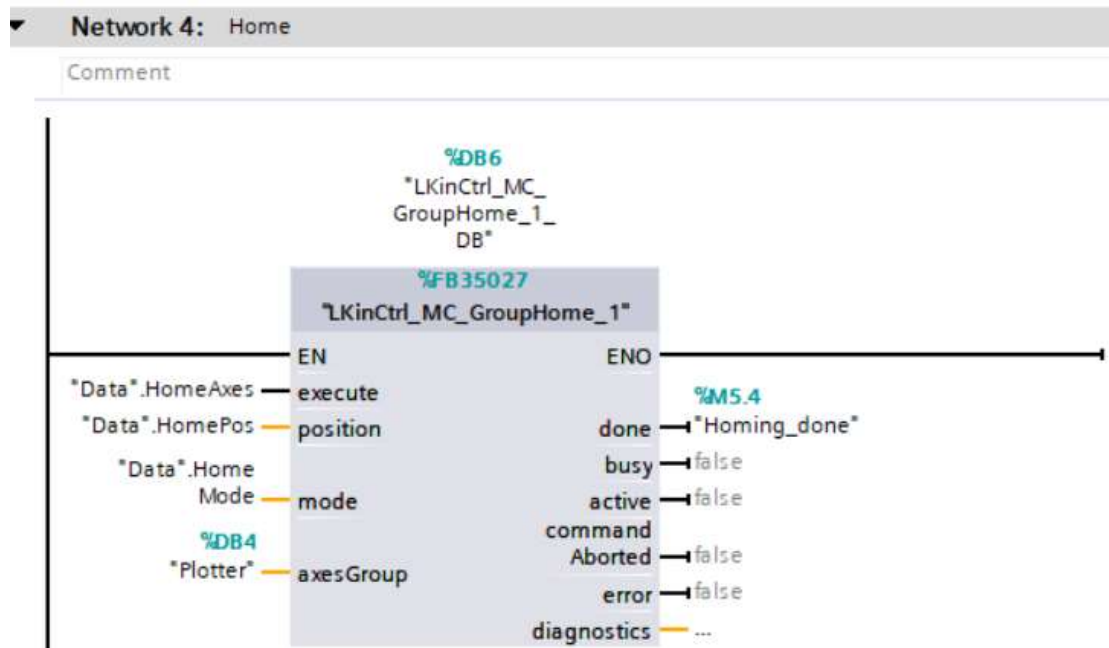
Active: Výstup je funkčností velmi podobný předchozímu. Dává informaci o tom, že daný funkční blok je aktivní.

Command Aborted: Předposledním výstupem je signalizován případ, že daný příkaz je přerušen jiným příkazem.

Error: Poslední výstup signalizuje stav, kdy došlo k nějaké chybě v době, kdy se daný funkční blok prováděl.

3.4.3.4 FB LkinCtrl_MC_GroupHome

Funkční blok, který slouží pro zvolení výchozí pozice technologického objektu ovládající osy.



Obr. 3-26 Funkční blok LkinCtrl_MC_GroupHome

Execute: Tento vstup slouží pro vykonání daného bloku a je datového typu „BOOL“. K němu je připojen kontakt, který se aktivuje náběžnou hranou skrz tlačítko na HMI panelu nebo přes OPC UA server.

Position: Vstup, jehož datový typ je pole o rozměru [1..4]. Do tohoto pole se zadávají souřadnice, které se zapisují do jednotlivých os, při aktivaci vstupu execute. Jednotlivé hodnoty pole se označují jako 1..4, přičemž čísla odpovídají osám v pořadí x,y,z,A. V našem případě používáme pouze dvě osy, takže zbylé dvě souřadnice uvnitř pole zůstanou nevyplněné.

Mode: Další vstup slouží pro nastavení režimu nastavování výchozí pozice pro osy připojené přes technologický objekt „Plotter“. Datovým typem je zde opět pole typu integer o rozměru [1..4]. Každé číselné označení hodnoty v poli opět přísluší odpovídajícím osám jako v předchozím případě. Zmíněných režimů tento blok obsahuje celkem jedenáct. V této aplikaci je použit režim 7, protože se používá pro absolutní enkodéry, kterými dané motory disponují.

Režim	Popis režimu
0 – Přímý absolutní	Do souřadnicového systému se zapíše aktuální pozice osy.
1 – Přímý relativní	Do souřadnicového systému se zapíše aktuální pozice osy posunutá o parametr "Position".
2 – Pasivní bez resetu	
3 - Aktivní	Osa provádí pohyb podle nastavení z technologického objektu -> "Active homing" a hodnota je nastavena podle parametru "Position".
5 – Aktivní (parametr "Position" nemá žádný vliv)	Stejně jako režim 3, ale souřadnice jsou zapsány podle nastavení z technologického objektu.
6 – Absolutní nastavení kodéru s posunutím o parametr "Position"	Aktuální pozice je posunuta o hodnotu parametru "Position". Vypočítaná absolutní hodnota offsetu je uložena v CPU.
7 – Absolutní nastavení kodéru s nastavením na parametr "Position"	Aktuální poloha je nastavena na hodnotu parametru "Position". Vypočítaná absolutní hodnota offsetu je uložena remanentně v CPU.
8 - Pasivní	Když je detekována naváděcí značka, hodnota se nastaví na hodnotu parametru "Position". Tato naváděcí značka se nastaví v technolickém objektu -> Passive homing.
9 – Zrušení pasivního režimu	Pasivní režim je přerušen.
10 – Pasivní (parametr "Position" nemá žádný vliv)	Když je detekována naváděcí značka, hodnota se nastaví na výchozí polohu, která je nastavena v technologickém objektu
11 – Posunutí požadované polohy	Požadovaná poloha technologického objektu je posunuta o hodnotu parametru "Position".

Tab. 3-1 Režimy pro nastavení výchozí polohy

AxesGroup: Stejně jako u předchozích.

Done: První výstup slouží pro signalizaci stavu, že došlo k úspěšnému nastavení výchozí pozice. Jedná se o datový typ „BOOL“. V mém případě jsem na tento výstup zapojil kontakt „Homing_done“, se kterým se poté dále pracuje v programu.

Busy: Stejně jako u předchozích funkčních bloků.

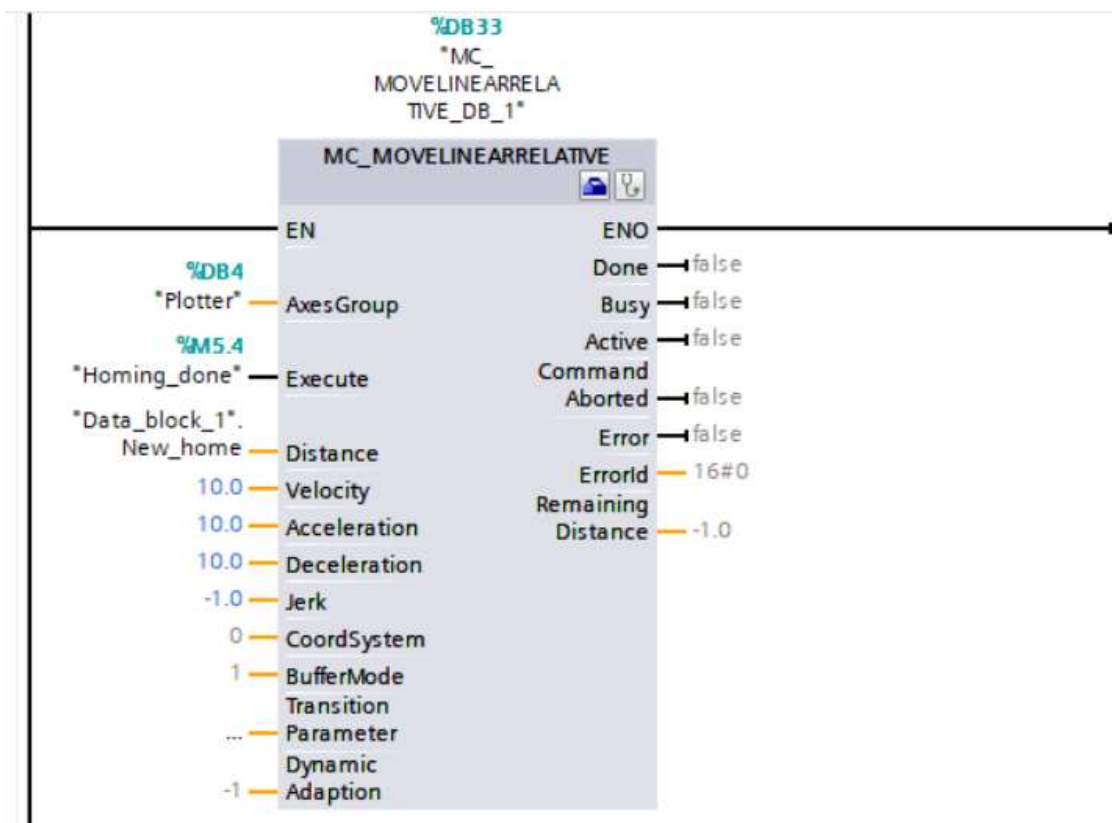
Active: Stejně jako u předchozích funkčních bloků.

Command aborted: Stejně jako u předchozích funkčních bloků.

Error: Stejně jako u předchozích funkčních bloků.

Diagnostics: Stejně jako u předchozího funkčního bloku.

K logice nastavování výchozí pozice os patří ještě jedna síť v organizačním bloku „Main“. V této chvíli by osa x, která ovládá pero, byla přímo na koncovém spínači a ten by tak při zavolání systému na výchozí pozici byl aktivní. Koncový spínač má tu vlastnost, že slouží jako kontakt pro vyvolání nouzového zastavení celého systému, aby nedošlo ke zničení příslušného motoru. Proto je tedy potřeba výchozí pozici posunout směrem vpravo, aby při zavolání os na výchozí (domovskou) pozici nedošlo k aktivování koncového spínače a tím také k zastavení celého systému. Použil jsem tedy blok, který posune osu x záporným směrem (doprava) o 1 mm.



Obr. 3-27 Posunutí výchozí pozice

Funkce tohoto funkčního bloku je jednoduchá. Na vstup „AxesGroup“ je opět připojen náš technologický objekt ovládající obě osy. Na vstup „Execute“ je zapojen kontakt „Homing_done“, jež signalizuje dokončení přechodního funkčního bloku „LkinCtrl_MC_GroupHome“. Ve chvíli kdy je tento vstup aktivován, dojde k posunutí obou os o hodnotu 1 mm v záporném směru, která je zadána v nově vytvořeném poli „New_home“. Přednastavená rychlost pro tento posun je 10 mm/s.

3.4.3.5 Hlášení errorů

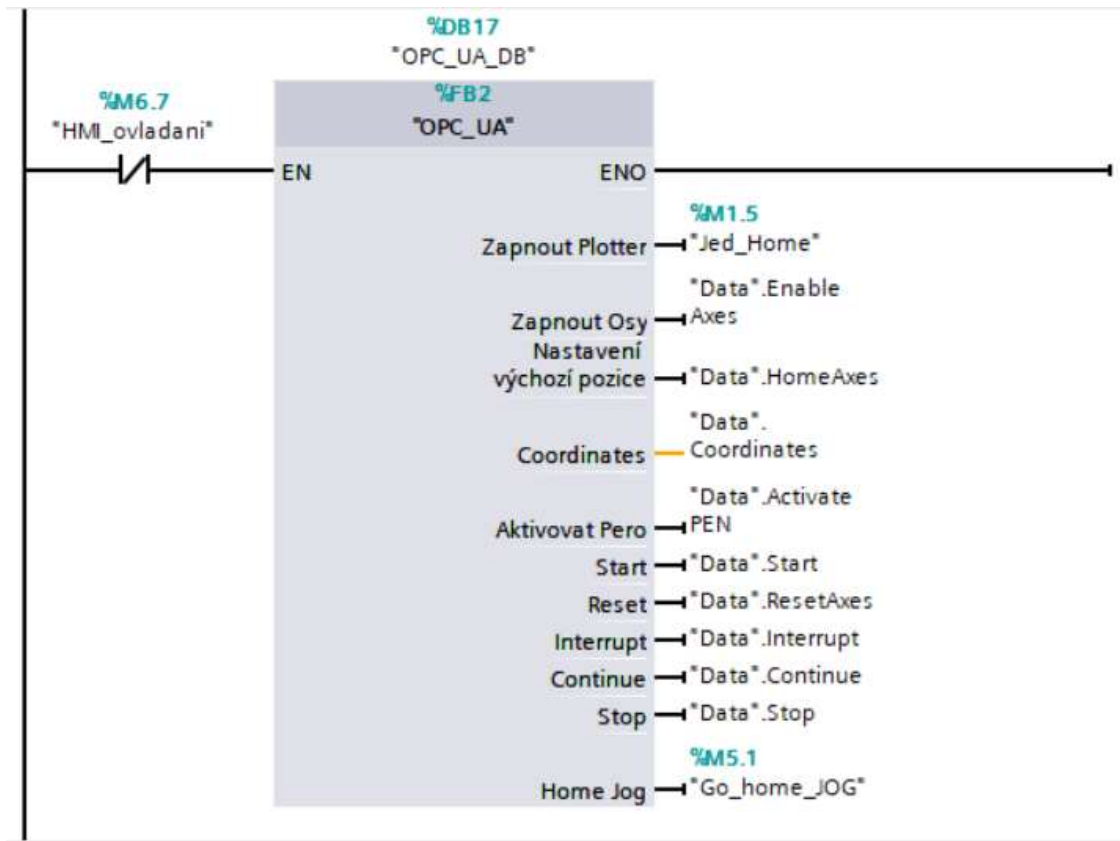
Další dílčí síť, která nese název „Errors“, slouží pro vyhlášení chyby jakéhokoliv technologického objektu. Jedná se o tři paralelně zapojené kontakty, kde každý z nich reprezentuje část z proměnné příslušného technologického objektu. Tato část indikuje výskyt chyby v daném technologickém objektu a v případě aktivace alespoň jednoho z kontaktů dojde k vyhlášení erroru. Tím se na výstup „Axes Error“ zapíše hodnota „TRUE“.



Obr. 3-28 Logika pro vyhlášení erroru technologických objektů

3.4.3.6 FB OPC-UA

Co znamená samotný OPC UA server a k čemu slouží již bylo zmíněno v teoretické části. K tomu, abychom mohli tuto aplikaci přehledně ovládat přes vzdálený OPC server, jsem vytvořil funkční blok, do kterého byly přidány všechny potřebné hodnoty, jež slouží k jejímu ovládání. Tento funkční blok má název „OPC-UA“. Všechny výstupy tohoto funkčního bloku mají stejný název, jako tlačítka na HMI panelu a jsou připojeny ke stejným proměnným. Vstup „EN“ slouží pro samotnou aktivaci daného funkčního bloku. Zapnutí je prováděno pomocí rozpínacího kontaktu „HMI_ovladani“. Pokud je tento kontakt ve stavu „FALSE“, probíhá ovládání aplikace přes OPC UA server. Když na HMI panelu stiskneme tlačítko „Aktivovat HMI“, dojde k deaktivaci tohoto funkčního bloku a aplikace je ovládána uživatelem právě přes HMI panel.



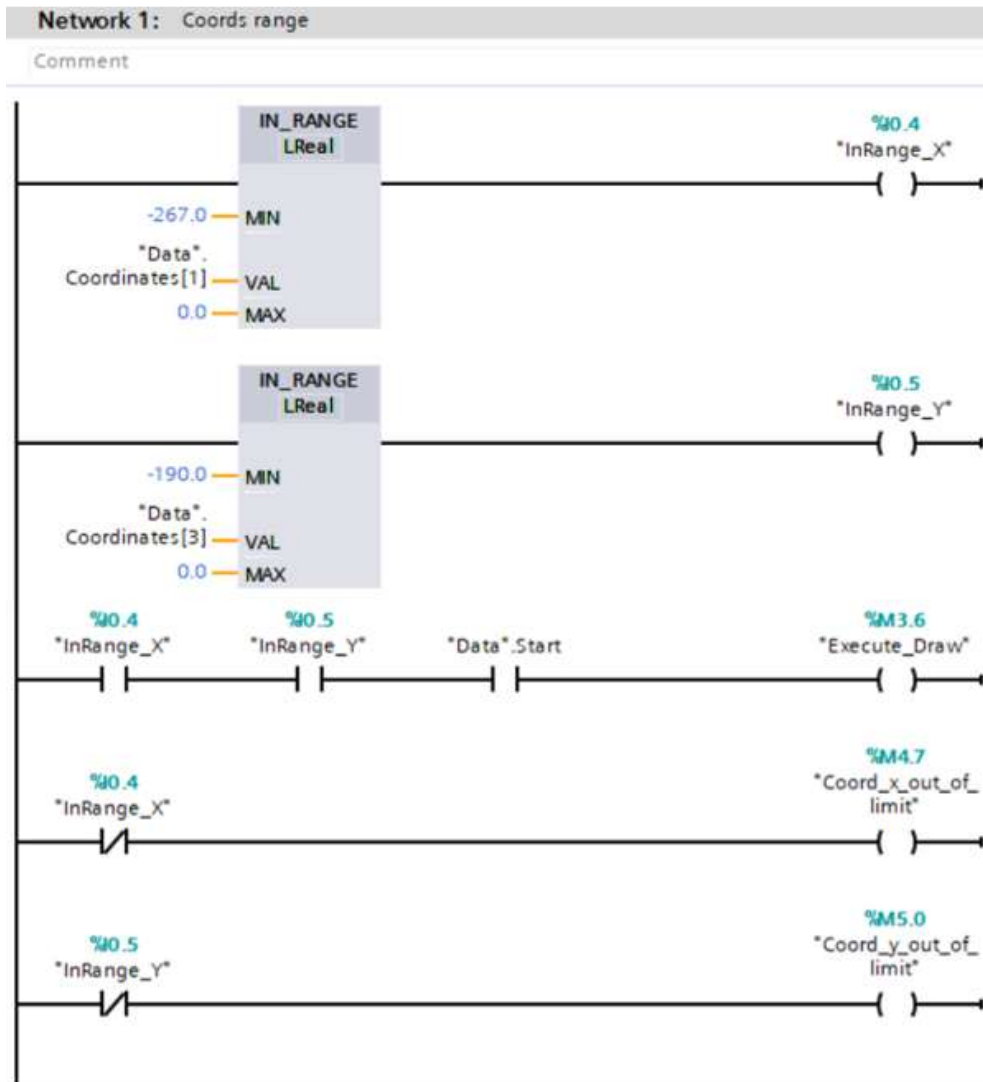
Obr. 3-29 FB OPC-UA

3.4.4 Drawing

Jak již bylo zmíněno v úvodu této kapitoly, program obsahuje dva organizační bloky. Jeden z nich je „Main“ a druhým je organizační blok „Drawing“. V něm je ovládána veškerá logika kreslení plotteru. Obsahuje devět sítí („Networks“), které si v dalších podkapitolách opět podrobně popíšeme.

3.4.4.1 Rozsah zadávaných souřadnic

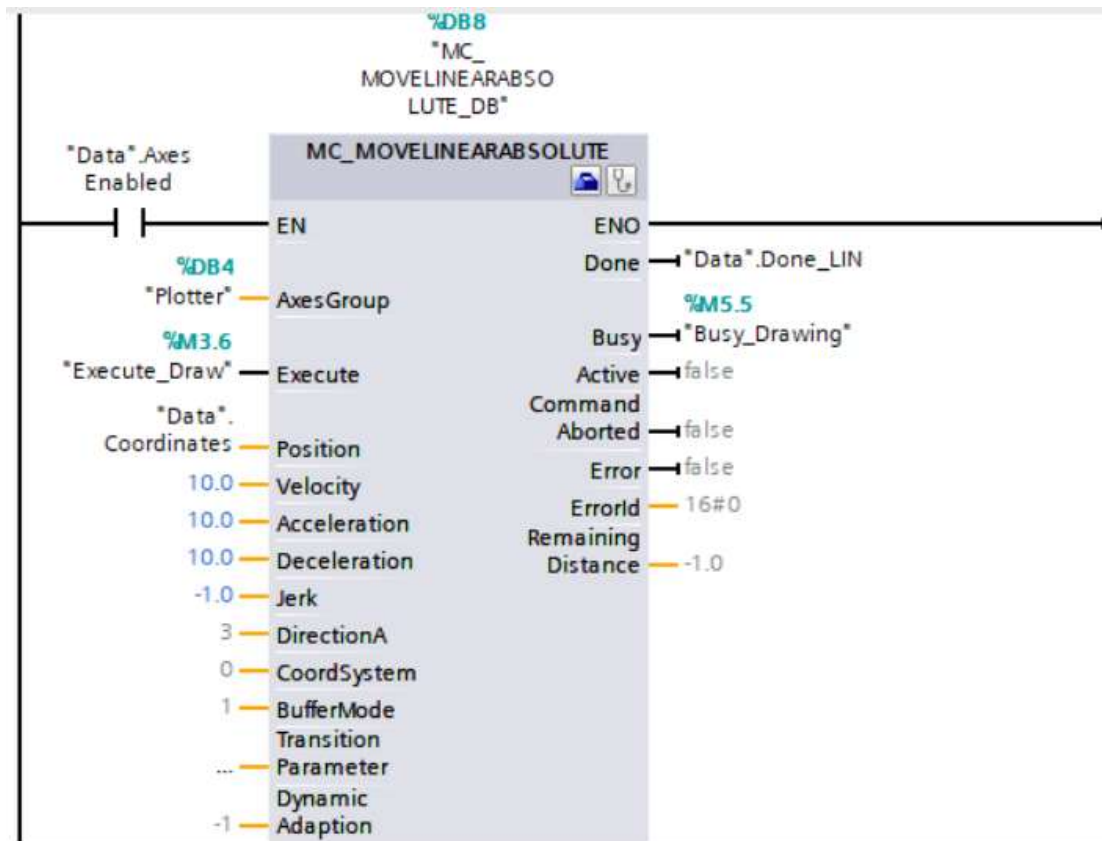
V této části organizačního bloku „Drawing“ je vytvořena logika pro zadávání souřadnic, podle kterých se plotter pohybuje. Jsou k tomu použity dva bloky „IN_Range“, který obsahuje tři vstupy. Dva z nich slouží pro zadání minimální a maximální hodnoty a třetím vstupem je porovnávaná hodnota – v tomto případě hodnota dané souřadnice. Na výstupu každého z obou bloků je kontakt „InRange_X“, respektive „InRange_Y“. Tyto kontakty jsou aktivní právě když je zadávaná souřadnice v daných mezích. Plotter je pro kreslení na papír formátu A4, který má rozměry 210x297 mm. Meze jsou však pro delší stranu papíru 0 – 267 a pro kratší 0-190. Důvod, proč se nedá zadat souřadnice, které odpovídají reálné velikosti papíru A4, je čistě konstrukční. Na obou stranách plotteru pero nemůže dojet až na konec. Pokud jsou obě souřadnice zadány v daném rozsahu, tak už se čeká pouze na pokyn „Start“ a plotter může začít kreslit. V opačném případě je aktivován výstup „Coord_x_out_of_limit“ pokud se jedná o nesplnění rozsahu pro osu x, respektive „Coord_y_out_of_limit“ pokud se jedná o osu y.



Obr. 3-30 Logika pro rozsah zadávaných souřadnic

3.4.4.2 Ovládání lineárního pohybu

Další část se věnuje ovládání lineárního pohybu plotteru. K tomu je použit funkční blok „MC_MOVELINEARABSOLUTE“. Nachází se v záložce instrukcí -> „Motion (kinematics)“ a jedná se o blok vytvořený vývojáři TIA portalu. Obsahuje následující vstupy a výstupy:



Obr. 3-31 Ovládání lineárního pohybu

EN: Stejně jako u jednoho z předchozích bloků, tento vstup využíváme pro samotnou aktivaci bloku. V tomto případě je to kontakt „Axes Enabled“, který je ve stavu „TRUE“ ve chvíli, kdy jsou aktivovány osy.

AxesGroup: Stejně jako u předchozích funkčních bloků.

Execute: Jak již bylo zmíněno, jedná se o vstup pro vykonání daného funkčního bloku. Zde je připojen kontakt „Execute_Draw“. Tento kontakt můžeme vidět na předchozím obrázku. Pokud jsou souřadnice pro obě osy zadány v daném rozsahu a je aktivován kontakt „Start“, je vykonán tento funkční blok.

Position: Pole o rozměru [1..4], kde opět každé číslo náleží příslušné ose. Do tohoto pole jsou zadávány souřadnice, do kterých má plotter dojet. Zadání souřadnic může proběhnout buď přes HMI panel nebo přes OPC UA server.

Velocity: Tento vstup slouží pro zadání rychlosti, jakým se bude plotter pohybovat. Jeho datovým typem je LReal, který umožňuje zadání čísla s desetinnou čárkou. Rychlost je zde zadávána v jednotkách mm/s. Může mít tři stavy. Pokud je zadána hodnota >0.0, blok pracuje s touto hodnotou rychlosti. Hodnota = 0 zakazuje pohyb. V případě hodnoty <0.0 blok pracuje se zadanou hodnotou rychlosti z pevného nastavení technologického objektu.

Acceleration: Opět se jedná o datový typ LReal a slouží pro nastavení zrychlení připojeného technologického objektu. Také nabízí možnost tří stavů, které jsou shodné, jako u předchozího vstupu.

Deceleration: Stejná funkce jako u předchozího vstupu, ale jedná se o nastavení zpomalení. Implicitně po přidání bloku má tento vstup společně s „Velocity“ a „Acceleration“ nastavenou hodnotu -1.0.

DirectionA: Tento vstup slouží pro nastavení směru pohybu ve smyslu kartézské orientace. Datovým typem je DINT a může nabývat tří stavů: 1 – pozitivní směr, 2 – negativní směr, 3 – nejkratší vzdálenost. Implicitně je nastaven režim č. 3, což je i v případě této aplikace.

CoordSystem: Datovým typem je opět DINT a slouží pro nastavení referenčního systému souřadnic. Obsahuje čtyři možné režimy: 1 – systém WCS (World Coord System), 2,3,4 – Objektový souřadnicový systém (OCS 1,2,3). V našem případě používáme režim 1.

Done: Prvním výstupem je signalizován stav, kdy plotter dokončil lineární pohyb, respektive se nachází na požadovaných souřadnicích.

Busy: Stejně jako u předchozích bloků – signalizuje stav, kdy je vykonáváno kreslení. Je k němu připojen kontakt „Busy_Drawing“, který dává uživateli na HMI panelu informaci o tom, že právě probíhá kreslení.

3.4.4.3 Aktivace pera

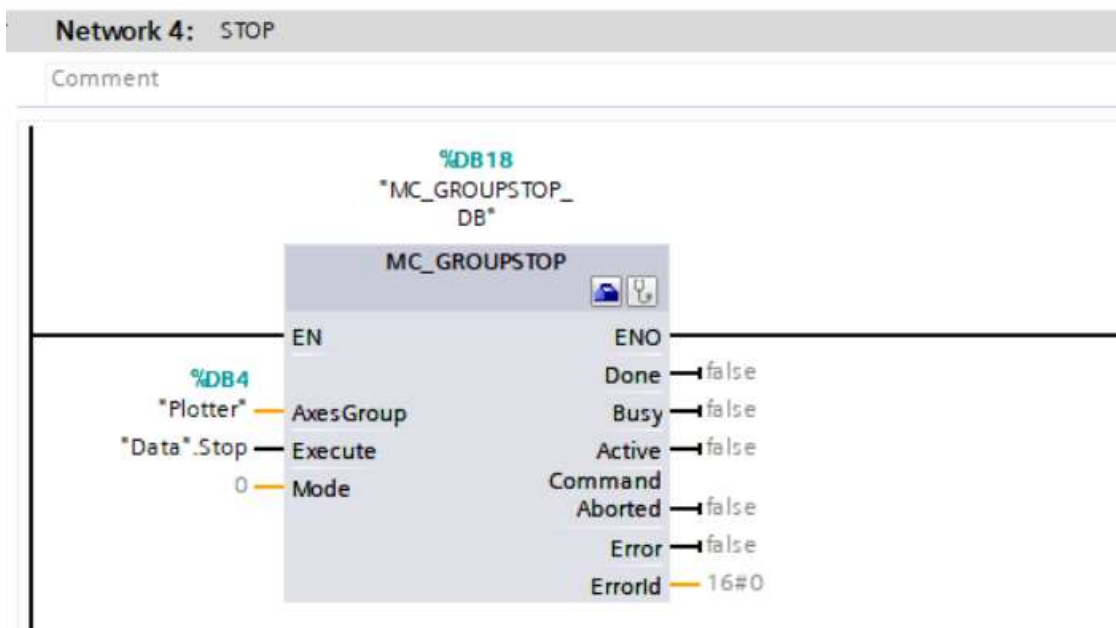
Plotter má na ose x umístěn zásobník, do kterého je možné vložit pero, kterým je pak možné kreslit po podložce – papíru. K tomu slouží fyzický výstup „%Q0.0“, kterým dojde k posunu zásobníku s perem směrem dolů a pero bude v kontaktu s papírem. K tomu slouží tři sériově zapojené kontakty. Jeden kontakt zajišťuje, aby se pero aktivovalo pouze v případě, že jsou aktivovány osy. Dalším je kontakt, který se aktivuje buď na HMI panelu „Aktivovat Pero“ nebo přes OPC UA server. Posledním je kontakt „ZakladaciPaka“. Pokud je ve stavu „TRUE“, na podložce se nachází papír a je upevněn právě pomocnou zakládací pákou, aby nedošlo k jeho nechtěnému pohybu. Slouží tedy k tomu, abychom se vyhnuli situaci, že by pod perem nebyl papír a docházelo by ke kreslení na konstrukci plotteru. Pokud jsou tedy všechny tři kontakty ve stavu „TRUE“, je aktivován výstup „PEN“ a pero je v kontaktu s papírem.



Obr. 3-32 Aktivace pera

3.4.4.4 STOP

Při provozu plotteru může dojít k situacím, kdy je potřeba jej urychleně zastavit. K této funkci se používá instrukce „MC_GROUPSTOP“, která okamžitě zastaví aktivní pohyb obou os připojených k technologickému objektu „Plotter“. Nachází se zde opět vstup pro konfiguraci technologického objektu. Na vstup pro vykonání objektu je připojen kontakt „Stop“, který je napojen na velké červené tlačítko STOP na HMI panelu. Posledním vstupem můžeme konfigurovat režim zastavení.

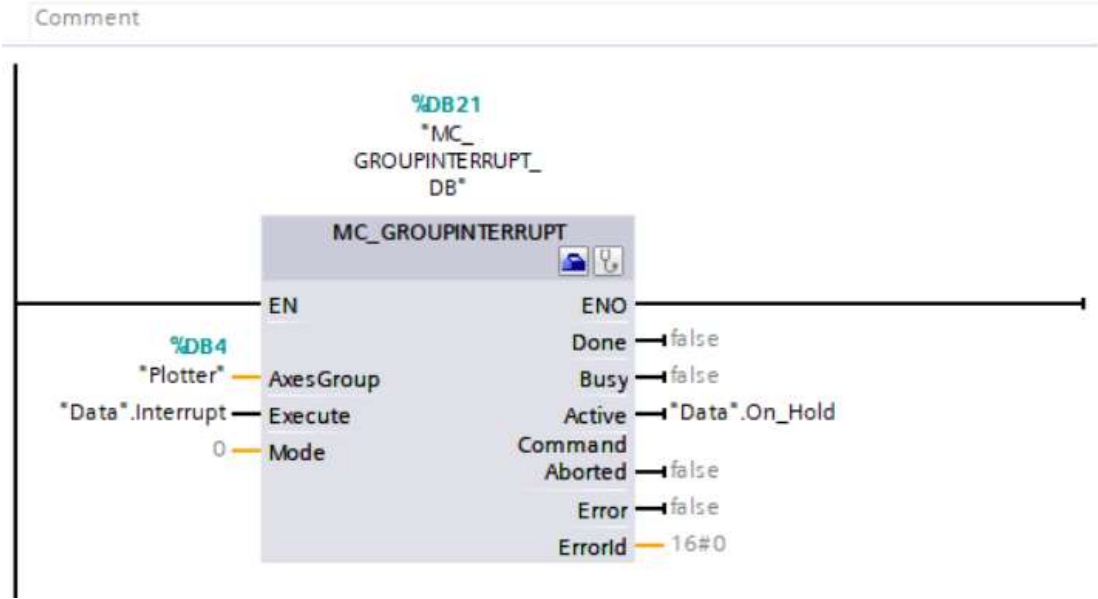


Obr. 3-33 Zastavení plotteru

3.4.4.5 Přerušení

Narozdíl od předchozí instrukce, která plotter kompletně zastaví, existuje také možnost jej pouze přerušit. K tomu slouží instrukce „MC_GROUPINTERRUPT“. Stejně jako u předchozích nakonfigurujeme technologický objekt, pro který se má daná instrukce vykonat. Pokud je instrukce z nějakého důvodu vykonána, nedojde k úplnému zastavení, ale lze v zadaném pohybu pokračovat.

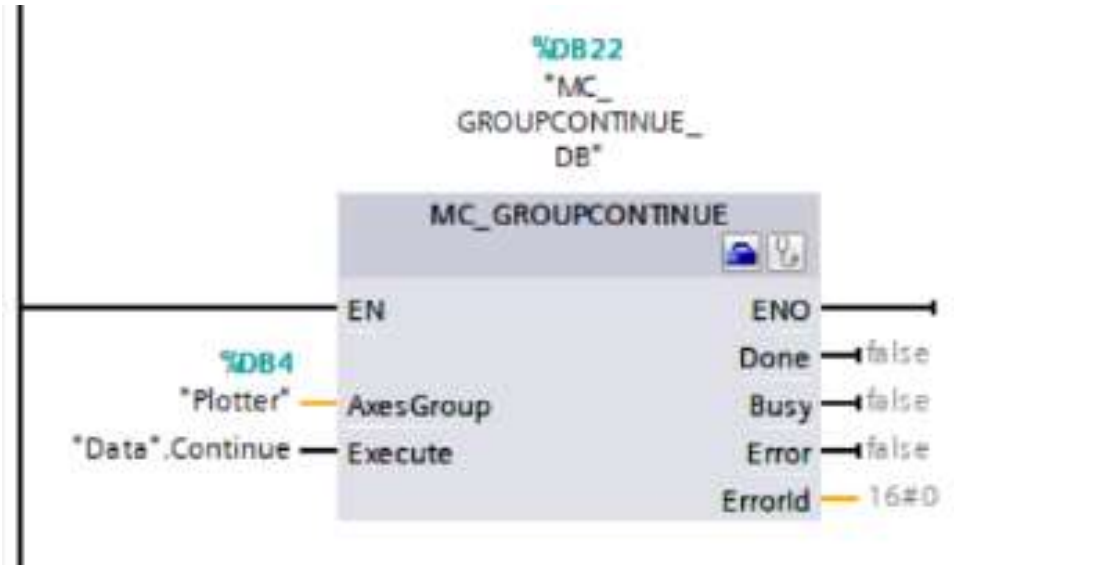
Network 5: Interrupt + Continue



Obr. 3-34 Přerušení pohybu

3.4.4.6 Pokračování

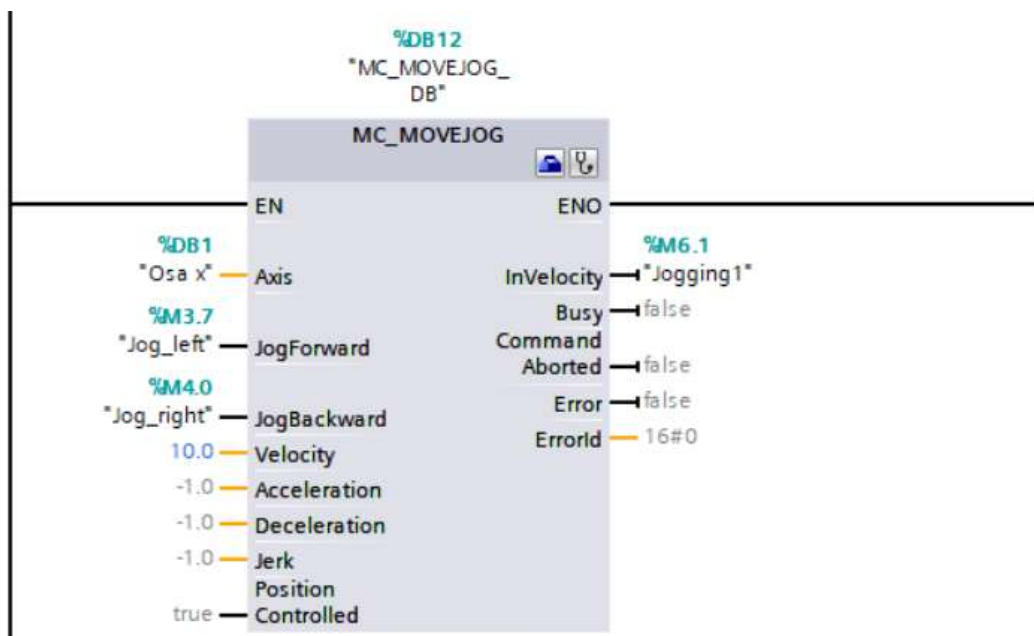
K pokračování v zadaném kinematickém pohybu, který byl dříve přerušen předchozí instrukcí, slouží „MC_GROUPCONTINUE“.



Obr. 3-35 Pokračování v pohybu

3.4.4.7 JOG

Jako poslední v tomto organizačním bloku je vytvořena logika pro „Jogging“. Tímto názvem rozumíme manuální posouvání plotteru v osmi směrech. K tomu slouží instrukce „MC_MOVEJOG“. Zde už se nenastavuje technologický objekt, který obsahuje obě osy, ale úloha je řešena pro každou osu zvlášť. Zmíněných instrukcí „MC_MOVEJOG“ se zde nachází celkem šest a pro ilustraci bude uveden pouze jeden. Nejdříve pro pohyb doleva–doprava, nahoru–dolů a poté pro pohyb do zbylých čtyřech směrů. Na každý výstup „Busy“ jsou připojeny kontakty „Jogging 1-6“. Ty jsou paralelně spojeny a přivedeny na jeden výstup „Jogging“ který nám následně signalizuje, že plotter je v režimu manuálního pohybu.



Obr. 3-36 Jog

Tato ilustrační instrukce obstarává manuální posouvání předvolenou rychlostí 10 mm/s ve směru doleva-doprava, což odpovídá ose x, která je také připojena na vstup „Axis“. Jak vidíme, smysl posouvání je zde takový, že „JogBackward“ je ve smyslu zápornějších souřadnic – tedy doprava a „JogForward“ naopak. Proto jednou instrukcí dokážeme obstarat posouvání ve dvou směrech. Stejně tak je tomu u směru nahoru-dolů, kde je naopak připojena osa y a analogicky vstupy pro její oba směry. Opět vidíme, že jednou instrukcí zajistíme pohyb do dvou stran. U manuálního pohybu do zbylých čtyř směrů musí být pro každý směr jedna instrukce.

3.5 HMI

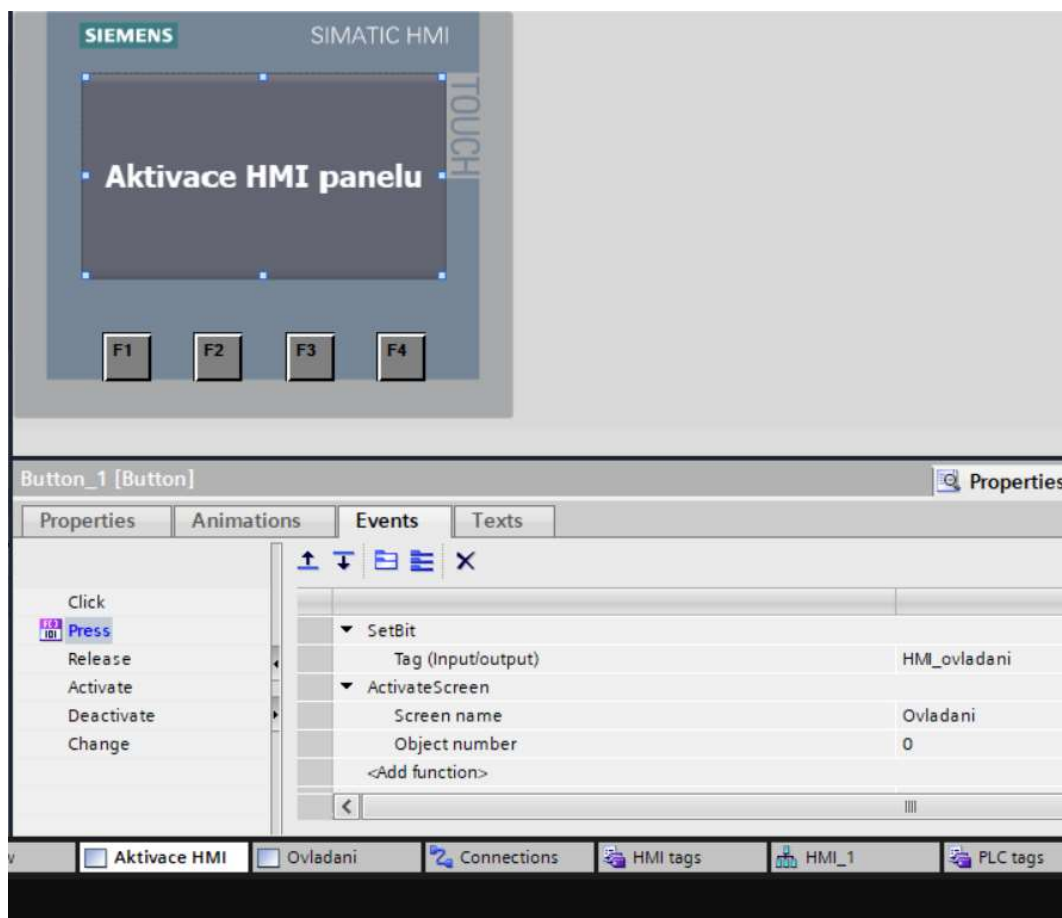
Nejdříve je nutné nadefinovat propojení PLC a HMI panelu, aby spolu obě jednotky mohly komunikovat. To zajistíme v záložce „Connections“. Obsahuje dvě obrazovky. Jako vstupní obrazovka, která se objeví při spuštění, je „Aktivace_HMI“. Zde se nachází pouze jedno tlačítko, kterým se deaktivuje ovládání přes OPC-UA, respektive aktivuje ovládání přes HMI panel. Další funkce tohoto tlačítka je spuštění hlavní obrazovky „Ovladani“, přes který je plotter ovládán.

3.5.1 HMI tagy

Stejně jako při vytváření programu pro PLC, kde se pracuje s PLC tagy, tak u tvorby vizualizace pro HMI panel je nutné používat HMI tagy. Ty mohou být buď interní nebo propojené na PLC tag. Mezi interní patří například čísla obrazovek. Při vytváření externího HMI tagu je nutné jej propojit s odpovídajícím PLC tagem přes název spojení, které jsme si dříve definovali. V našem případě „HMI_Connection_1“. Jejich název může a nemusí být shodný. Datový typ však musí mít shodný, jako jejich propojený PLC tag.

3.5.2 Obrazovka „Aktivace HMI“

Jak již bylo zmíněno, HMI panel obsahuje dvě obrazovky. Ta, která se zobrazí při spuštění, je obrazovka pro aktivaci panelu. Obsahuje jedno tlačítko („Button“), které má dvě funkce. První funkcí je deaktivace funkčního bloku „OPC-UA“ a tím aktivace ovládání systému přes HMI panel. Druhou funkcí je přechod na druhou obrazovku při jeho stisknutí. To je realizováno přes funkci „SetBit“, respektive „ActivateScreen“, jak můžeme vidět na následujícím snímku obrazovky.



Obr. 3-37 Funkce obrazovky „Aktivace HMI“

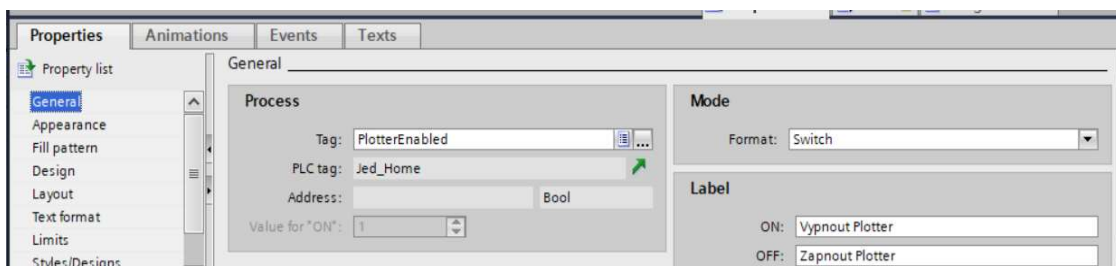
3.5.3 Obrazovka „Ovladani“

Druhá obrazovka sloužící pro ovládání plotteru obsahuje spoustu tlačítek, přepínačů, signalizačních prvků, výstražných hlášek nebo také pole pro zadávání souřadnic. Všechny tyto objekty budou popsány a vysvětlena jejich funkčnost.



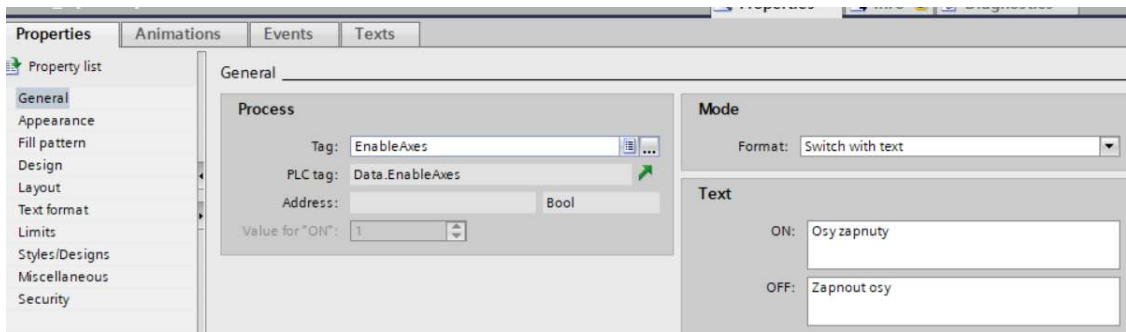
Obr. 3-38 Obrazovka „Ovladani“

Jako první si představíme přepínač pro zapnutí a vypnutí plotteru. Ten je propojením tagů připojen na PLC tag „Jed_Home“, který se vyskytuje ve funkčním bloku „Zapnout_plotter“ viz kapitola 3.4.3.1.



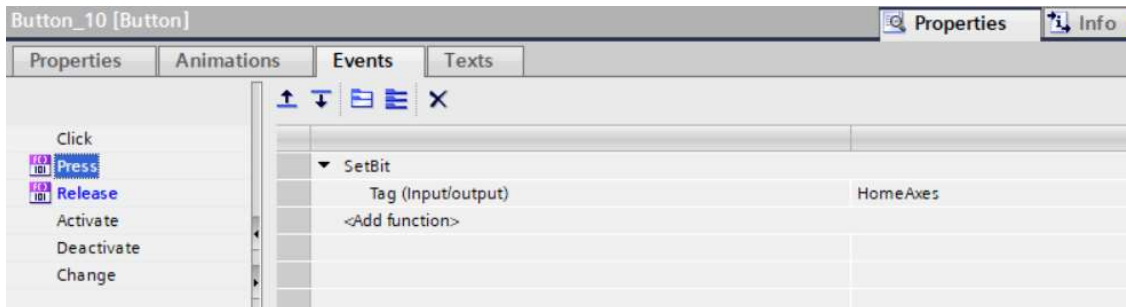
Obr. 3-39 Nastavení objektu pro zapnutí plotteru

Ve chvíli, kdy osa x plotteru dojde na levý koncový spínač, objeví se na obrazovce přepínač pro zapnutí os. Ten je připojen na PLC tag „EnableAxes“ viz kapitola 3.4.3.2. Při zapnutí os plotteru se přepínač na znamení úspěšné aktivace os zbarví do zelena a text se změní na „Osy zapnuty“. Viditelnost tohoto objektu je tedy omezena pouze na situaci, kdy osa x plotteru dojde na levý koncový spínač a plotter je připraven k dalším instrukcím.



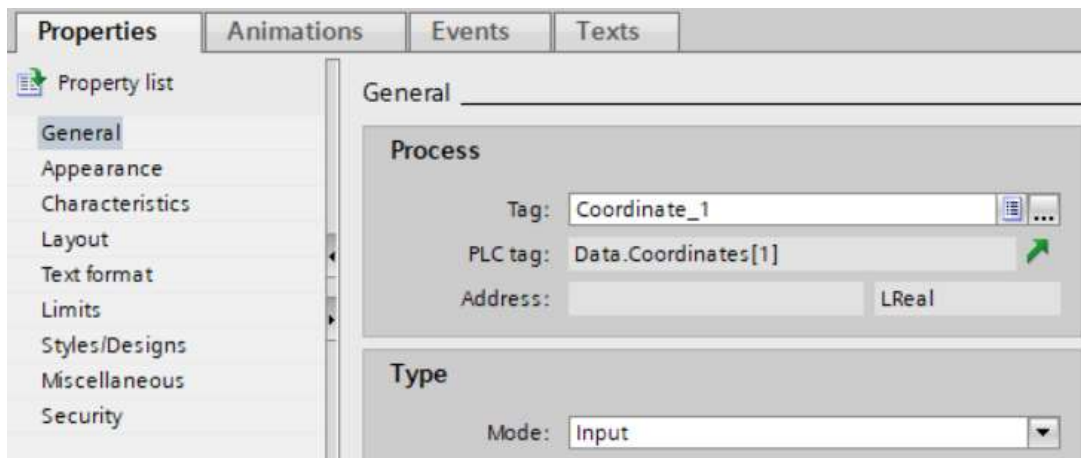
Obr. 3-40 Nastavení objektu pro aktivaci os

Po aktivování os plotteru se na obrazovce objeví tlačítko „Nastavení výchozí polohy“. Narozdíl od předchozích se nejedná o přepínač, ale tlačítko. V tomto případě nelze tlačítko propojit s určitým tagem, jako tomu je u přepínače. Proto je pro toto tlačítko nastavena událost stisknutí a uvolnění „SetBit“, respektive „ResetBit“ na tag „HomeAxes“. Tím se vykoná funkční blok „MC_GROUPHOME“ viz kapitola 3.4.3.4. Jak již bylo zmíněno v této kapitole, po stisknutí tlačítka pro nastavení výchozí pozice následně dojde k posunutí osy x směrem doprava o 1 mm.



Obr. 3-41 Nastavení objektu pro výchozí pozici

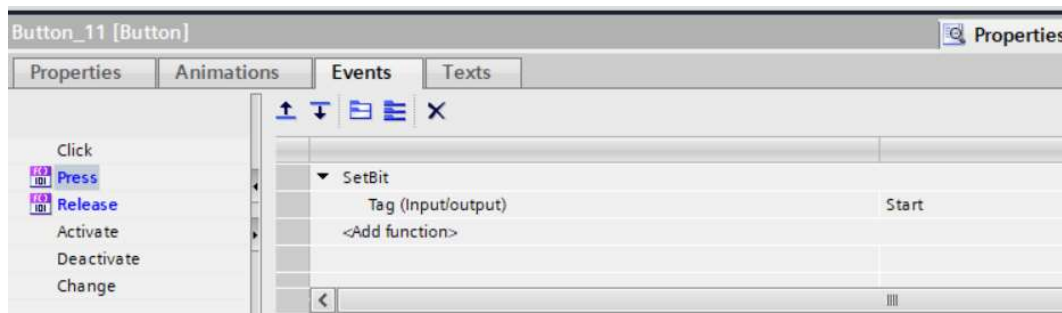
Když je dokončeno nastavení výchozí pozice, může uživatel přejít k zadávání souřadnic, do kterých má systém dojet. To je realizováno pomocí objektů „I/O field“, kdy pro každou osu je zde obsažen jeden. Ten je poté napojen na příslušnou souřadnici, která se nachází v datovém objektu. Tento objekt je na HMI panelu viditelný poté, co jsou aktivovány osy.



Obr. 3-42 Nastavení objektu pro souřadnice

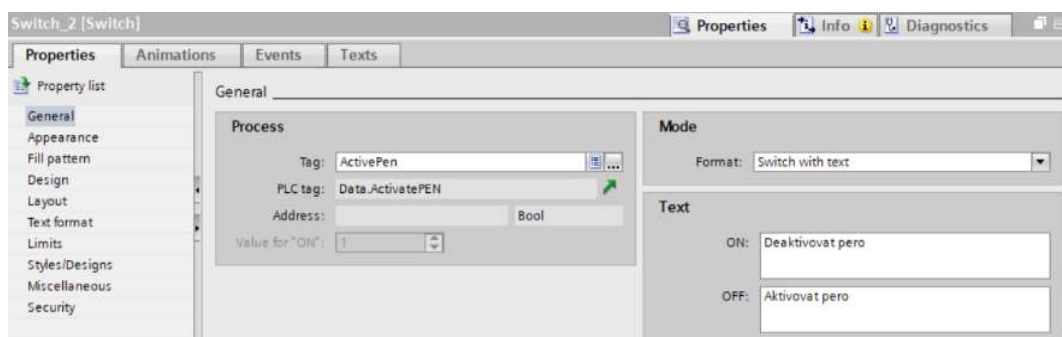
Zadávání souřadnic má pro každou svůj daný rozsah. Pokud je do pole pro souřadnice zadána hodnota mimo tento rozsah, vyskočí chybová hláška, která na tuto skutečnost upozorní uživatele a zároveň zobrazí správný rozsah, který musí být dodržen viz Obr. 3-38.

Po zadání souřadnic ve správném rozsahu můžeme dát systému pokyn pro start kreslení. To je realizováno pomocí tlačítka, které je provázané na vstup „Start“ z datového bloku. Tímto vstupem se aktivuje pomocný výstup „Execute_Draw, který vyvolá funkční blok pro lineární kreslení viz Obr. 3-31. Důvod toho, proč není zmíněný funkční blok volán rovnou přes vstup „Start“ je ten, aby se proces kreslení aktivoval pouze v případě, když je dodržen rozsah zadaných souřadnic a nedošlo tak ke zničení pohonů os.



Obr. 3-43 Nastavení objektu pro start

Před začátkem procesu kreslení lze aktivovat pero, které poté podle pohybu os vykresluje linární obrazce na podložku. Tuto instrukci lze zavolat i při probíhající proces, ovšem obvykle z důvodu deaktivace pera, pokud by z něho například vytékal inkoust nebo byla poničena podložka, na kterou je právě vykreslováno. Pro tuto funkci je využito přepínače, který je provázán na vstup z datového bloku „ActivatePEN“. Tento vstup následně aktivuje fyzický výstup příslušný pro ovládání zásobníku s perem („PEN“). Důvod je abychom zabránili aktivaci pera při chybějící nebo nepřipevněné kreslicí podložce pomocí zakládací páky a nebyla tak pokreslena konstrukce. Tato logika je vysvětlena na Obr. 3-32. Při aktivaci se ještě přepínač podbarví zelenou barvou na znamení toho, že je pero skutečně aktivní a nápis se změní na „Deaktivovat pero“.



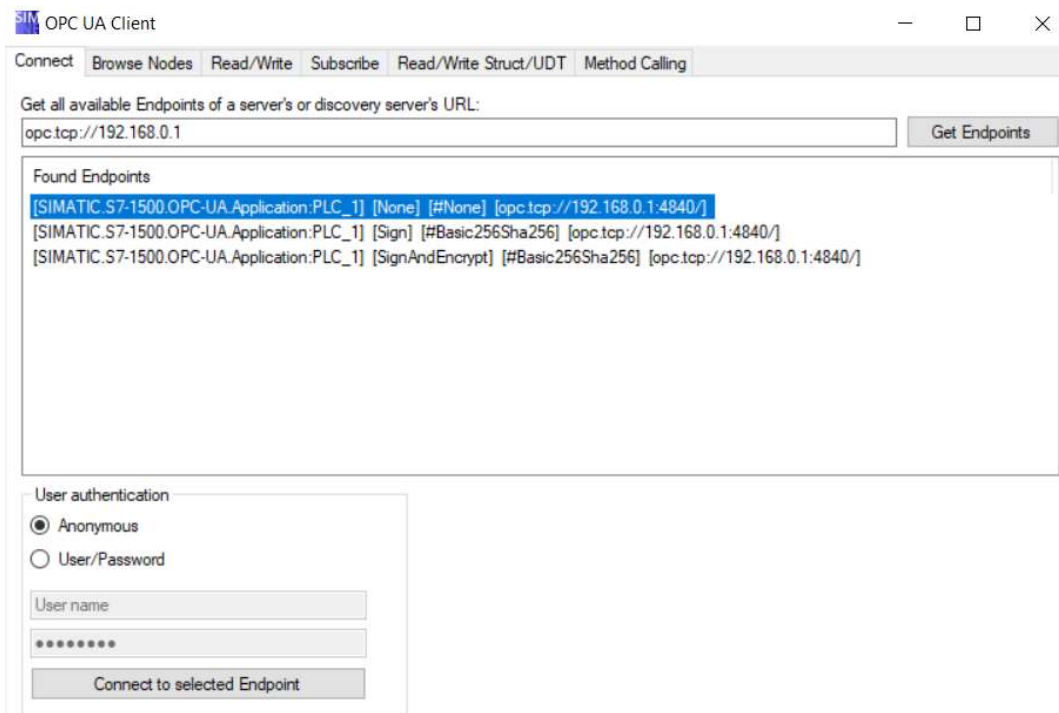
Obr. 3-44 Nastavení objektu pro aktivaci pera

Pokud je zrovna systém v režimu aktivního kreslení, můžeme jej zastavit, přerušit a následně pokračovat v procesu nebo resetovat. K tomu slouží tlačítka „STOP“, „Interrupt“, „Continue“ a „Reset“. Zastavení je pro případy nouze a po vyvolání této instrukce je vždy nutné celý systém resetovat a následně provést znovu nastavení výchozí pozice. Pokud vyvoláme přerušování kreslení, systém se v daném místě zastaví a po pokynu pro pokračování dokončí instrukci, která mu byla zadána. Všechny tyto instrukce jsou řešeny přes objekt tlačítka a jsou přiřazeny ke stejnojmenným vstupům z datového bloku. Tlačítko STOP a Reset je z bezpečnostních důvodů viditelné a použitelné od doby zapnutí systému, ostatní tlačítka jen v době, kdy probíhá kreslení.

Posledními objekty na obrazovce pro ovládání systému jsou šipky a symbol domu. Tyto šipky slouží pro manuální pohyb „Jogging“. Tato instrukce byla vysvětlena v podkapitole 3.4.4.7. Tlačítko, na kterém je vyobrazen dům slouží k tomu, aby se po jeho stisknutí systém vrátil do nastavených výchozích hodnot. Opět jsou tyto instrukce řešeny přes objekt tlačítka, kdy ke každému směru je přiřazen příslušný PLC tag. Manuální pohyb je možné provádět kdykoliv od aktivace os, ale ne při procesu kreslení, kdy z obrazovky zmizí.

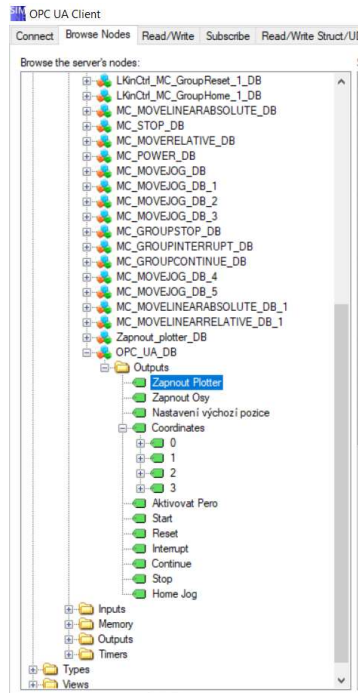
3.6 OPC UA

Nejprve je nutné aktivovat funkci OPC UA serveru v nastavení PLC. Dále je možné nastavit parametry jako minimální interval vzorkování, minimální interval publikování nebo maximální počet řízených veličin. Pro ovládání přes OPC UA server lze definovat také uživatele, kdy každý může mít své přihlašovací údaje a různé práva, co smí ovládat a co ne. Dále je nutné definovat runtime licenci. Pro tuto aplikaci stačí nejmenší, tedy „SIMATIC OPC UA S7-1500 small“. Pro samotné ovládání příslušných veličin jsem použil OPC UA client přímo od firmy Siemens, který je volně stažitelný na jejich webových stránkách.



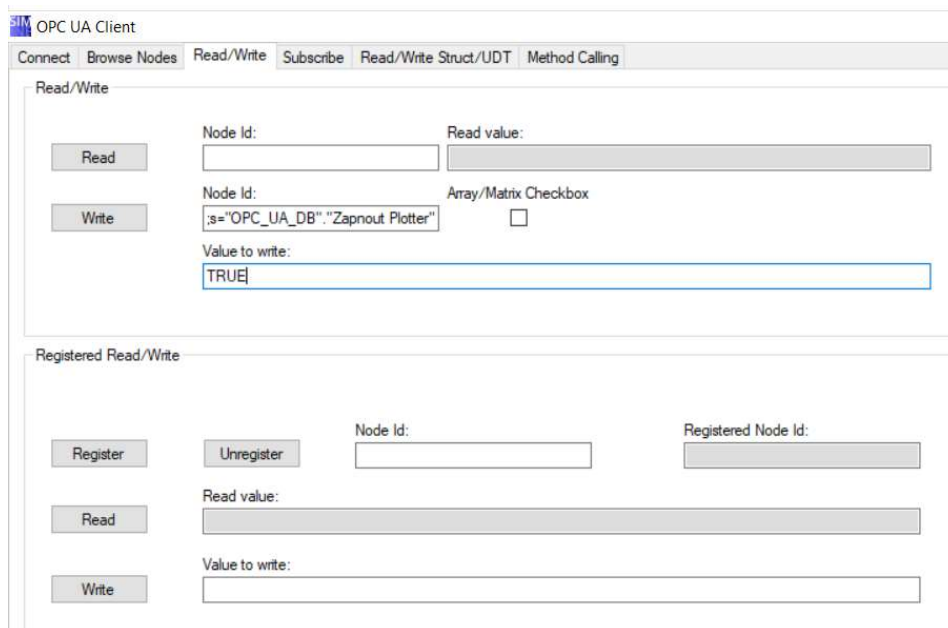
Obr. 3-45 Nastavení komunikace pro OPC klienta

V tomto programu nejdříve musíme zajistit komunikaci mezi PLC a serverem zadáním IP adresy PLC, kterou jsme již dříve nastavili. Poté zvolíme licenci, kterou jsme dříve nastavili a můžeme se připojit k PLC. V tuto chvíli můžeme ovládat výstupy, které jsme si definovali do funkčního bloku pro OPC UA server.



Obr. 3-46 Výstupy z funkčního bloku pro OPC UA server

Pro ovládání jednotlivých výstupu slouží záložka „Read/Write“, kde do nich lze zapisovat příslušné hodnoty. Pro souřadnice je nutné zapisovat hodnotu s jedním desetinným místem, protože se jedná o datový typ *LReal*.



Obr. 3-47 Zapisování hodnot na výstupy přes OPC UA server

ZÁVĚR

Cílem této diplomové práce bylo vytvořit fungující aplikaci pro ovládání plotteru pomocí PLC a dalších komponent od firmy Siemens. To může být ovládáno buď přes vzdálený OPC UA server nebo HMI panel. Plotter je určen pro kreslení na podložku formátu A4, ovšem vzhledem ke konstrukčním limitům bylo nutné tento rozměr oříznout na 190x267 mm.

Nejprve bylo nutné si řádně prostudovat obecné informace o řízení technologických objektů pomocí PLC, příslušných servomotorech, frekvenčních měničích, HMI panelu a typech komunikace. Dále jsem si musel najít informace k OPC UA serveru, protože jsem se s problematikou vzdáleného řízení nikdy nesetkal. Samozřejmostí bylo také seznámení se s pracovištěm, které mi bylo poskytnuto a je popsáno v první podkapitole. Při tomto procesu se menším problémem jevila skutečnost chybějícího koncového spínače na pravé straně plotteru, kterou jsem nakonec vyřešil pomocí softwarových limitů.

Po prostudování teoretické části práce jsem se mohl vrhnout na realizování aplikace ve vývojovém prostředí TIA Portal. Prvním krokem po založení projektu bylo pro daný HW vytvořit konfiguraci, do které patří PLC společně se vstupně/výstupním modulem, dva frekvenční měniče, k nim příslušně servomotory a HMI panel. Pro každý prvek bylo nutné nastavit IP adresu, verzi FW a název zařízení (*Device name*). Poté jsem nastavil typ komunikace, jejíž typ je v tomto případě *IRT*. Nejprve jsem měl problémy s komunikací jednoho z měničů, ale záhy jsem zjistil, že jsem udělal chybu ve zmíněné HW konfiguraci při přidělování jednotlivých portů.

Když byla konfigurace v pořádku a komunikace probíhala mezi všemi prvky v pořádku, mohl jsem se pustit do samotného programování. Nejprve jsem pro každý frekvenční měnič definoval technologický objekt, kterým je poziční osa (*Positioning Axis*). Ovšem pro ovládání obou pohonů zároveň bylo nutné definovat ještě jeden technologický objekt kinematiky (*Kinematics*). Zde jsem nastavil obě ovládané osy a definoval další potřebné informace pro jejich ovládání. Některé funkční bloky pro pohyb plotteru jsou z přídatné knihovny *Library Kinematics Control*. Problém se vyskytl při vytváření logiky nastavování výchozí pozice pomocí předdefinovaného funkčního bloku, a proto jsem byl vytvořil nový pomocný funkční blok, aby bylo možné tuto pozici správně nadefinovat. Dalším krokem bylo vytvoření obrazovek pro ovládání přes HMI panel.

Když byla aplikace plně funkční, vytvořil jsem funkční blok pro ovládání přes vzdálený OPC UA server. To je realizováno pomocí klienta, který je volně dostupný na webových stránkách firmy Siemens.

LITERATURA

- [1] PLC. PLC Automatizace [online]. HaPeSoft [cit. 2023-03-03]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc.htm>
- [2] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. Praha: BEN - technická literatura, 1999. ISBN 80-86056-58-9.
- [3] Elektronická učebnice - ELUC. Elektronická učebnice - ELUC [online]. [cit. 2023-03-03] Dostupné z: <https://eluc.ikap.cz/verejne/lekce/970>
- [4] CYKLUS PLC. PLC Automatizace [online]. HaPeSoft [cit. 2023-03-04]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-cyklus.htm>
- [5] ČSN EN 61131-3. PLC: Programovací jazyky. Praha, 2003
- [6] NORMA PRO ŘÍDÍCÍ SYSTÉMY IEC 61 131 [online]. [cit. 23-03-04] Dostupné z: http://www.edumat.cz/texty/zaciname_IEC61131.pdf
- [7] PLC SESTAVA. PLC Automatizace [online]. HaPeSoft [cit. 2023-03-04]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-hw-sestava.htm>
- [8] Siemens, s.r.o. + ČVUT FEL, Katedra elektrických pohonů a trakce: Decentrální řízení [online]. [cit. 23-03-10] Dostupné z: <https://moodle.fel.cvut.cz/mod/resource/view.php?id=256982>
- [9] Siemens, s.r.o. + ČVUT FEL, Katedra elektrických pohonů a trakce: Centrální řízení [online]. [cit. 23-03-10] Dostupné z: <https://moodle.fel.cvut.cz/mod/resource/view.php?id=256983>
- [10] LETTL, Jiří, Jiří PAVELKA a Jan BAUER. Výkonová elektronika. 4. přepracované vydání. V Praze: České vysoké učení technické, 2019. ISBN 978-80-01-06514-3.
- [11] Rozdíl mezi inkrementálním a absolutním rotačním enkodérem - D-Ex Instruments. Profesionálna technika nielen pre priemysel a Ex prostredie - D-Ex Instruments [online]. Copyright © 2023 [cit. 23-03-12]. Dostupné z: <https://www.dex.sk/clanek/rozdil-mezi-inkrementalnim-a-absolutnim-rotacnim-ekoderem/>
- [12] Totally Integrated Automation Portal - Automation Software - Global. [online]. Copyright © Siemens 1996 [cit. 2023-03-19]. Dostupné z: <https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html>
- [13] Siemens, s.r.o. + ČVUT FEL, Katedra elektrických pohonů a trakce: Komunikace – RT x IRT [online]. [cit. 2023-04-07] Dostupné z: <https://moodle.fel.cvut.cz/mod/resource/view.php?id=256981>

- [14] MARTIN P. CLARK. Data Networks, IP and the Internet: Protocols, Design and Operation, 2003. ISBN 978-0-470-84856-2.
- [15] PROFIBUS PA. Praktická teorie: Základní informace o průmyslové sběrnici PROFIBUS - část I. [online]. Liberec: Foxon [cit. 2023-04-02]. Dostupné z: <https://www.foxon.cz/blog/prakticka-teorie/162-zakladni-informace-oprumyslove-sbernici-profibus-cast-i>
- [16] PROFIBUS. [online]. [cit. 2023-04-07] Dostupné z: <http://www1.fs.cvut.cz/cz/u12110/site/profibus>
- [17] PROFINET. Praktická teorie: A tohle jste o PROFINETU věděli? 2. díl průmyslové komunikace [online]. Liberec: Foxon [cit. 2023-04-02]. Dostupné z: <https://www.foxon.cz/blog/prakticka-teorie/206-a-tohle-jste-o-profinetu-vedeli2-dil-prumyslove-komunikace>
- [18] Průmyslová komunikace OPC UA - 1.díl - popis protokolu | Automatizace.HW.cz. Automatizace.HW.cz | Elektronika v automatizaci [online]. [cit. 2023-04-07] Dostupné z: <https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html>
- [19] OPC UA methods for the SIMATIC S7-1500 OPC UA server [online]. [cit. 2023-04-08] Dostupné z: <https://support.industry.siemens.com/cs/document/109756885/opc-ua-methods-for-the-simatic-s7-1500-opc-ua-server-?dti=0&lc=en-CZ>
- [20] SCE Training Curriculum. Siemens AG, 2016 [online]. [cit. 2023-03-19]. Dostupné z: <https://www.automation.siemens.com/sce-static/learning-training-documents/tia-portal/summary-sce-training-curriculum-s7-1500-en.pdf>
- [21] Product Details - Industry Mall - Siemens WW. [online] [cit. 2023-04-10]. Dostupné z: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7511-1TL03-0AB0>
- [22] Product Details - Industry Mall - Siemens WW. [online]. [cit. 2023-04-10] Dostupné z: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7523-1BL00-0AA0>
- [23] SINAMICS S210 servo drive - SINAMICS Servo Drives - Global. [online]. [cit. 2023-04-11]. Dostupné z: <https://www.siemens.com/global/en/products/drives/sinamics/low-voltage-converters/servo-converter/sinamics-s210.html>
- [24] 6SL3210-5HB10-1UF0 Siemens. WiAutomation | Industrial Automation Spares parts Supplier [online]. [cit. 2023-04-11]. Dostupné z: https://cz.wiautomation.com/siemens/drives-motors-circuits-protection/sinamics/6SL32105HB101UF0?gclid=CjwKCAjwgqejBhBAEiwAuWHioEtPVSQLVnbqHNbOA86VAF3gB7Wgw8HVXiIE5Efncj1rF80aEHNqRoCqEIQAvD_BwE
- [25] 6AV2123-2DB03-0AX0 SIMATIC HMI KTP400 Basic [online]. [cit. 2023-04-11] Dostupné z: <https://support.industry.siemens.com/cs/pd/379924?ptdi=pi&dl=en&lc=en-CZ>

- [26] Detaily produktů - Industry Mall - Siemens Ceska republika. [online]. [cit. 2023-04-12]
Dostupné z: <https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/1FK2102-0AG00-1MA0>
- [27] S7-1500/S7-1500T Axis functions [online]. [cit. 2023-04-15]. Dostupné z:
https://cache.industry.siemens.com/dl/files/462/109766462/att_1004378/v1/s71500_s71500t_axis_function_manual_en-US_en-US.pdf
- [28] S7-1500T Kinematics functions [online]. [cit. 2023-04-17] Dostupné z:
https://support.industry.siemens.com/cs/attachments/109766463/s7-1500t_kinematics_function_manual_en-US_en-US.pdf?download=true
- [29] Library Kinematics Control [online]. Erlangen, Nemecko: Siemens AG Digital Industries Division, V3.3.0. [online]. [cit. 2023-04-20] Dostupné z:
https://support.industry.siemens.com/cs/attachments/109755891/109755891_LKinCtrl_Doc_v3_3_0_en.pdf

PŘÍLOHA A: SEZNAM ZKRATEK

A.1 Seznam zkratek

PLC	Programovatelný logický automat (Programmable Logic Controller)
I/O	Vstupně/Výstupní (Input/Output)
HMI	Rozhraní stroj-člověk (Human Machine Interface)
CPU	Centrální procesorová jednotka (Central processing unit)
FB	Funkční blok (Function block)
LD	Žebříkový diagram (Ladder diagram)
FBD	Diagram funkčních bloků (Function block diagram)
ST	Strukturovaný text (Structured text)
IL	Seznam instrukcí (Instruction list)
RT	Reálný čas (Real time)
IRT	Isochronní reálný čas (Isochronous real time)
NRT	Ne reálný čas (Non real time)
HW	Hardware
SW	Software
FW	Firmware