

**KATEDRA ELEKTRICKÝCH
POHONŮ A TRAKCE**

**ČESKÉ VYSOKÉ UČENÍ
TECHNICKÉ V PRAZE**



**FAKULTA ELEKTROTECHNICKÁ
ŘÍZENÍ XY PLOTERU POMOCÍ
PLC SIMATIC**

DIPLOMOVÁ PRÁCE

BŘEZEN 2023

**JIŘÍ
MAREK**

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Marek** Jméno: **Jiří** Osobní číslo: **483530**
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 XY ploteru pomocí PLC Simatic

Název diplomové práce anglicky:

XY Plotter Control by PLC Simatic

Pokyny pro vypracování:

- 1) Seznamte se s problematikou návrhu řízení technologických celků pomocí PLC
- 2) Popište konfiguraci komponent potřebnou pro řízení XY plotru pomocí PLC S1500 v prostředí TIA portál
- 3) Popište komunikační protokol použitý pro řízení plotru
- 4) V HMI navrhnete vizualizaci pro ovládání plotru

Seznam doporučené literatury:

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

Jméno a pracoviště vedoucí(ho) diplomové práce:

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

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

Datum zadání diplomové práce: **25.04.2023**

Termín odevzdání diplomové práce: **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, 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 doc. Ing. Janu Bauerovi, Ph.D. za jeho podporu a profesionální pohled během vypracování této práce. Děkuji také katedře pohonů a trakce Fakulty elektrotechnické ČVUT za zapůjčení přípravku ploteru již připraveného pro využití řízení pomocí PLC. V poslední řadě chci také velice poděkovat technické podpoře českého zastoupení společnosti Siemens za ochotu zapůjčit kriticky důležité zařízení, bez nichž by nebylo možné realizovat kompletní řízení přípravku.

PROHLÁŠENÍ

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

.....
V Praze dne 24. května 2023

ABSTRAKT

Tato práce se věnuje oblasti automatizace a PLC programování. Předmětem práce je návrh řízení XY ploteru včetně samočinného vykreslování výkresů řízeného pomocí PLC zařízení Siemens SIMATIC S7-1500 společně s návrhem obsluhy přípravku pomocí HMI panelu Siemens KTP7400 Basic a KP900 Comfort. Práce obsahuje teorii programovatelných logických automatů, jejich historii, sestavu, principy a základní nástroje k jejich programování, seznámení s vývojovým prostředím TIA Portal, úvod do problematiky CAD/CAM/CNC technologie návrhu a výroby a popis při návrhu zpracování souborů, překladu příkazového souboru, ovládání řízeného pohybu dvouosého pohonu a realizace obsluhy v programu TIA Portal.

Klíčová slova: ČVUT FEL, katedra elektrických pohonů a trakce, diplomová práce, automatizace, PLC, Siemens, řízení, XY ploter, TIA Portal, CAM, CNC, CAD

ABSTRACT

This document is dedicated to automation a PLC programming field. Subject of the thesis is designing a control of XY ploter with self-manufacturing drawing by Siemens PLC device SIMATIC S7-1500 along with designing a operation HMI panels using Siemens KTP400 Basic and KP900 Comfort due to advanced HMI faceplates. Thesis consist of programmable logic controller theory with its history, structure, principle and fundamental programming tools, CAD/CAM/CNC machine technology introduction and getting to know a software platform TIA Portal along File transfer tool and, at last, a description when designing a inner control logic and operation panel using TIA Portal.

Keywords: CTU FEE, Department of Electric Drives and Traction, diploma thesis, automation, PLC, Siemens, control, XY ploter, TIA Portal, CAM, CNC, CAD

OBSAH

ÚVOD	13
HISTORIE PRŮMYSLOVÝCH AUTOMATŮ	14
KAPITOLA 1: POPIS PLC.....	15
1.1 ÚVOD	15
1.2 PLC	15
1.3 SESTAVA PLC	16
1.4 CPU.....	17
1.5 PAMĚŤ	17
1.5.1 Paměť programu.....	18
1.5.2 Paměť dat.....	18
1.6 BLOK NAPÁJENÍ.....	18
1.7 I/O BLOK.....	18
1.8 KOMUNIKAČNÍ ROZHRANÍ	18
1.8.1 HMI panel.....	18
KAPITOLA 2: POČÍTAČOVÉ ŘÍZENÍ VÝROBY CAD/CAM/CNC.....	20
2.1 TECHNOLOGIE CAM.....	20
2.1.1 G-code programovací jazyk.....	20
2.2 CAD.....	21
2.3 CNC STROJ.....	21
KAPITOLA 3: PROGRAMOVÁNÍ PLC.....	22
3.1 PROGRAMOVACÍ ZAŘÍZENÍ	22
3.2 PLC PROGRAMOVACÍ JAZYKY V PROSTŘEDÍ TIA PORTAL.....	22
3.2.1 Liniové schéma (LAD).....	23
3.2.2 Blokové schéma (FBD)	24
3.2.3 Strukturovaný text (SCL).....	24
KAPITOLA 4: SESTAVA ZAŘÍZENÍ ÚLOHY.....	26
4.1 HARDWARE.....	26
4.1.1 SIMATIC S7-1500 CPU 1515TF-2 PN	26
4.1.2 Siemens SIMATIC HMI KTP400 Basic	26
4.1.3 Siemens SIMATIC S7-1500 digital input/output module	27
4.1.4 Siemens SITOP PSU100S power supply module	28
4.1.5 Siemens SIMATIC HMI KP900 Comfort.....	28
4.1.6 Siemens SINAMICS S210.....	29
4.1.7 Siemens SIMOTICS S-1FK2 HD.....	30
4.2 SOFTWARE	31
4.2.1 Siemens TIA Portal.....	31
4.2.2 SIMATIC File Transfer tool.....	31
KAPITOLA 5: POPIS PŘÍPRAVKU	32

KAPITOLA 6: ÚVOD VE VÝVOJOVÉM PROSTŘEDÍM TIA PORTAL.....	35
6.1 ZALOŽENÍ PROJEKTU	35
6.2 PROJECT VIEW	36
6.3 NASTAVENÍ KOMUNIKACE ZAŘÍZENÍ S PROJEKTEM.....	37
6.4 NALADĚNÍ REGULACE DVOUOSÉHO SERVOPOHONU	40
KAPITOLA 7: POPIS POUŽITÝCH NÁSTROJŮ PŘI VÝVOJI	42
7.1 TECHNOLOGICKÝ OBJEKT (TO)	42
7.1.1 TO Positioning Axis.....	43
7.1.2 Motion Control.....	43
7.2 TO KINEMATICS.....	44
7.3 EXTERNÍ KNIHOVNY KINEMATIKY.....	45
7.3.1 Kinematics Control.....	45
7.3.2 Kinematics Language	45
KAPITOLA 8: NÁVRH ŘÍZENÍ DVOUOSÉHO POHONU PLOTERU	47
8.1 ANALÝZA POŽADAVKŮ NA NÁVRH ŘÍZENÍ PŘÍPRAVKU	47
8.2 KONFIGURACE TO PERO A PAPÍR	48
8.3 KONFIGURACE TO PLOTTER.....	53
8.4 IMPORT EXTERNÍCH KNIHOVEN DO PROJEKTU.....	55
8.5 IMPLEMENTACE MOTION CONTROL ŘÍZENÍ OS.....	56
8.5.1 Řízení osy.....	57
8.5.2 Řízení manuálního pohybu osy pera a papíru	59
8.6 REALIZACE MANUÁLNÍHO REFEROVÁNÍ OS.....	61
8.6.1 Referování osy papíru.....	62
8.6.2 Nastavení papíru při vypnutí.....	64
8.6.3 Referování osy pera	65
KAPITOLA 9: NÁVRH OBSLUHY PLOTERU	69
9.1 TECHNOLOGICKÉ SCHÉMA VÝMĚNY PERA	69
9.2 NÁVRH LOGICKÉHO ŘÍZENÍ VÝMĚNY PERA	71
9.3 REALIZACE OVLÁDÁNÍ PŘÍPRAVKU PŘI VÝMĚNĚ PERA.....	74
KAPITOLA 10: REALIZACE AUTOMATICKÉHO VYKRESLOVÁNÍ	76
10.1 ANALÝZA VYUŽITÍ KNIHOVNY LKINLANG	76
10.2 PŘÍPRAVNÉ KROKY PRO ŘÍZENÍ AUTOMATICKÉHO VYKRESLOVÁNÍ	76
10.3 IMPLEMENTACE LKINLANG FUNKCÍ DO PROJEKTU.....	78
10.3.1 Návrh propojení HMI aplikací a PLC programu	79
10.3.2 Implementace automatického vykreslování do řízení TO	80
10.4 KONFIGURACE FILE TRANSFER TOOL.....	83
10.5 LOGIKA PRO KRESLENÍ PERA.....	87
10.6 LADĚNÍ PŘEKladU G-CODE SOUBORU	91
KAPITOLA 11: VIZUALIZACE ŘÍZENÍ XY PLOTERU.....	94
11.1 KONFIGURACE HMI PANELU KTP400 BASIC A KP900 COMFORT	94
11.2 POUŽITÉ PRINCIPY PROGRAMOVÁNÍ HMI PANELŮ	96
11.2.1 Properties	96
11.2.2 Animations	96
11.2.3 Events	97
11.3 NÁVRH VIZUALIZACE PANELU KTP400 BASIC	98

11.3.1	Základní obsluha.....	98
11.3.2	Manuální řízení.....	99
11.3.3	Výměna pera.....	100
11.4	NÁVRH VIZUALIZACE PANELU KP900 COMFORT.....	100
11.4.1	Implementace LKinLang HMI aplikací.....	101
11.4.2	Implementace LKinCtrl_Editor aplikace.....	102
11.4.3	Ovládání překladu výkresu.....	103
11.4.4	Řízení pohybu pohonu MovePath.....	103
KAPITOLA 12: POPIS NAVRŽENÉHO ŘÍZENÍ XY PLOTERU		105
12.1	ZÁKLADNÍ OVLÁDÁNÍ	105
12.1.1	Zapnutí.....	105
12.1.2	Vypnutí.....	106
12.1.3	Reset řízení pera	106
12.2	MANUÁLNÍ ŘÍZENÍ.....	107
12.3	VÝMĚNA PERA.....	108
12.3.1	Ovládání na panelu KP900 Comfort.....	108
12.3.2	Ovládání na panelu KTP400 Basic.....	109
12.4	IMPORT VÝKRESU DO PLC ZAŘÍZENÍ.....	110
12.5	VÝBĚR SOUBORU	111
12.6	PŘEKLAD G-CODE SOUBORŮ	113
12.7	SPUŠTĚNÍ AUTOMATICKÉHO VYKRESLOVÁNÍ.....	114
12.8	EDITORY K ÚPRAVĚ PŘÍKAZŮ.....	115
ZÁVĚR		117
LITERATURA.....		119
PŘÍLOHA A: TECHNICKÉ ÚDAJE ZAŘÍZENÍ SESTAVY		122
A.1	SIMATIC S7-1500 CPU 1515TF-2 PN.....	122
A.2	SIEMENS SIMATIC HMI KTP400 BASIC.....	133
A.3	SIEMENS SIMATIC S7-1500 DIGITAL INPUT/OUTPUT MODULE.....	141
A.4	SIEMENS SIMATIC HMI KP900 COMFORT	142
A.5	SIEMENS SINAMICS S210 PN	144

SEZNAM OBRÁZKŮ

Obr. 1.1 Struktura PLC (upraveno)	15
Obr. 1.2 Modulární typ PLC.....	16
Obr. 1.3 Jednoduchý typ PLC.....	16
Obr. 1.4 Struktura CPU.....	17
Obr. 1.5 Struktura paměti (upraveno)	18
Obr. 2.1 Instrukce G-code (upraveno)	20
Obr. 3.1 Speciální programovací zařízení pro PLC	22
Obr. 3.2 Liniové schéma	23
Obr. 3.3 Blokové schéma.....	24
Obr. 3.4 Strukturovaný text.....	25
Obr. 4.1 PLC zařízení Siemens SIMATIC S7-1500 CPU 1515TF-2 PN	26
Obr. 4.2 Popis HMI panelu Siemens SIMATIC HMI KTP400 Basic (upraveno).....	27
Obr. 4.3 DI/DQ blok Siemens SIMATIC S7-1500 DI/DQ module	28
Obr. 4.4 Napájecí blok Siemens SITOP PSU100S power supply module.....	28
Obr. 4.5 HMI panel Siemens SIMATIC HMI KP900 Comfort.....	29
Obr. 4.6 Měnič Siemens SINAMICS S210 PN	29
Obr. 4.7 Sestava řízeného servopohonu Siemens SINAMICS S210 a Siemens SIMOTICS S – 1FK2 (upraveno)	30
Obr. 5.1 Kompletní sestava zařízení.....	33
Obr. 5.2 XY ploter.....	33
Obr. 5.3 Detail přípravku XY ploteru	34
Obr. 6.1 Vytvoření projektu TIA Portal	35
Obr. 6.2 Konfigurace zařízení TIA Portal.....	36
Obr. 6.3 Struktura Project view TIA Portal	37
Obr. 6.4 Konfigurace komunikace zařízení	38
Obr. 6.5 Přiřazení IP adresy zařízení v sestavě.....	39
Obr. 6.6 Přiřazení PROFINET jména zařízení v sestavě.....	39
Obr. 6.7 Provedení příkazu <i>Go online</i>	40
Obr. 6.8 Funkce One Button Tuning.....	40
Obr. 7.1 Nabídka TO v prostředí TIA Portal	42
Obr. 7.2 TO Polohovací osy (upraveno)	43
Obr. 7.3 Motion Control (upraveno).....	44
Obr. 7.4 Princip TO Kinematiky (upraveno)	44
Obr. 7.5 Použití knihovny <i>Kinematics Control</i>	45
Obr. 7.6 Použití knihovny <i>Kinematics Language</i>	46
Obr. 8.1 Požadavky návrhu.....	47
Obr. 8.2 Založení TO Polohovací osy	48
Obr. 8.3 Konfigurace TO – <i>Basic parametres</i>	49

Obr. 8.4 Konfigurace TO – <i>Hardware interface</i>	50
Obr. 8.5 Pohon osy pohybu papíru - <i>Commsioning funkce</i>	51
Obr. 8.6 Konfigurace TO – <i>Extended parametres a Homming</i>	52
Obr. 8.7 Založení TO Plotter	53
Obr. 8.8 Konfigurace TO Plotter – <i>Basic parametres</i>	54
Obr. 8.9 Konfigurace TO Plotter – <i>Interconnexions</i>	54
Obr. 8.10 Konfigurace TO Plotter – <i>Geometry a Jog sequence</i>	55
Obr. 8.11 Import externí knihovny do TIA Portal prostředí.....	56
Obr. 8.12 Aktualizace projektové knihovny	56
Obr. 8.13 Vložení FB bloku.....	57
Obr. 8.14 LAD schéma FB bloku Papir, chod osy	58
Obr. 8.15 LAD schéma FB bloku Pero, chod osy	59
Obr. 8.16 LAD schéma FB bloku Papir, manuální řízení pohybu ...	60
Obr. 8.17 LAD schéma FB bloku Pero, manuální řízení pohybu	61
Obr. 8.18 Vývojový diagram manuálního pohybu referování, TO Papir	62
Obr. 8.19 LAD schéma FB bloku Papir, manuální pohyb referování, část 1.....	63
Obr. 8.20 LAD schéma FB bloku Papir, manuální pohyb referování, část 2.....	64
Obr. 8.21 LAD schéma FB bloku Papir, nastavení osy při vypnutí.....	65
Obr. 8.22 Vývojový diagram manuálního pohybu referování, TO Pero	66
Obr. 8.23 LAD schéma FB bloku Pero, manuální pohyb referování	67
Obr. 8.24 Odečtení vzdálenosti relativního pohybu referování	68
Obr. 9.1 Technologické schéma výměny pera	69
Obr. 9.2 Diagram výměny pera	70
Obr. 9.3 LAD schéma FB bloku Pero, výměna pera, část 1	72
Obr. 9.4 LAD schéma FB bloku Pero, výměna pera, část 2	73
Obr. 9.5 LAD schéma FB bloku Pero, výměna pera, část 3	74
Obr. 9.6 LAD schéma FB bloku Pero, řízení zásobníku.....	75
Obr. 10.1 Vytvoření <i>Global DB</i>	77
Obr. 10.2 Konfigurace DB Controls	78
Obr. 10.3 Konfigurace DB GlobalData	78
Obr. 10.4 Založení OB CNCMain a HmiMain.....	79
Obr. 10.5 Návrh organizačního bloku OB HmiMain	80
Obr. 10.6 Návrh organizačního bloku OB CNCMain, část 1	81
Obr. 10.7 Návrh organizačního bloku OB CNCMain, část 2	82
Obr. 10.8 Návrh organizačního bloku OB CNCMain, část 3	83
Obr. 10.9 Konfigurace funkce <i>Web server</i>	84
Obr. 10.10 Konfigurace přístupových profilů funkce <i>Web server</i>	85
Obr. 10.11 Přehrání konfigurace do PLC zařízení	86
Obr. 10.12 Připojení PC aplikace k Web serveru	87

Obr. 10.13 Definice PLC tagu PEN.....	87
Obr. 10.14 Založení PENinst data bloku	89
Obr. 10.15 Generování instrukcí pro PLC tag PEN	90
Obr. 10.16 Řízení přiložení pera.....	90
Obr. 10.17 Struktura FB LKinCtrl_MC_MovePath	91
Obr. 10.18 Nastavení funkce preProcessorConfig bloku LKinLang_Parser	92
Obr. 10.19 Úprava příkazového souboru typu G-code	93
Obr. 11.1 Vložení HMI panelu do projektu	94
Obr. 11.2 Nastavení komunikace HMI zařízení.....	95
Obr. 11.3 Přiřazení IP adresy HMI panelu.....	95
Obr. 11.4 Provedení vizualizace, sekce <i>Properties</i>	96
Obr. 11.5 Provedení vizualizace, sekce <i>Animations</i>	97
Obr. 11.6 Provedení vizualizace, sekce <i>Events</i>	97
Obr. 11.7 Základní obsluha přípravku	98
Obr. 11.8 Řešení viditelnosti úvodního okna.....	99
Obr. 11.9 Použití funkce <i>Animations</i>	99
Obr. 11.10 Výměna pera, panel KTP400 Basic.....	100
Obr. 11.11 Rozcestník, panel KP900 Comfort	101
Obr. 11.12 Vložení <i>faceplate</i> HMI aplikace	101
Obr. 11.13 Propojení HMI aplikace s FB blokem programu.....	102
Obr. 11.14 Definice HMI tag propojení.....	102
Obr. 12.1 Výběr formátu papíru	105
Obr. 12.2 Ovládací panel přípravku	106
Obr. 12.3 Rozcestník, panel KP900 Comfort	107
Obr. 12.4 Manuální řízení pera	108
Obr. 12.5 Výměna pera, panel KP900 Comfort	109
Obr. 12.6 Nezpřístupněná výměna pera, panel KP900 Comfort	109
Obr. 12.7 Výměna pera, panel KTP400 Basic	110
Obr. 12.8 Rozhraní aplikace File Transfer tool	111
Obr. 12.9 Výběr souboru.....	112
Obr. 12.10 Rozcestník, zapnutý Enable Manager	112
Obr. 12.11 Řízení překladu souborů G-code	114
Obr. 12.12 Řízení automatického vykreslování.....	115
Obr. 12.13 Rozhraní aplikace LKinCtrl Editor	115
Obr. 12.14 Rozhraní aplikace LKinLang Editor	116

ZKRATKY A SPECIÁLNÍ VÝRAZY

- DC – stejnosměrný proud (*direct current*)
- AC – střídavý proud (*alternating current*)
- PLC – programovatelný logický automat
- I/O blok – blok se vstupy a výstupy
- CPU – centrální procesorová jednotka
- RAM – paměť s přímým přístupem
- EPROM – přepisovatelná paměť, jejíž obsah není závislý na napájení
- ON/OFF – Zapnuto/vypnuto
- Drag & drop – princip vložení objektu do programu pomocí přetažení objektu do žádaného okna spolu se zmáčknutým levým tlačítkem myši
- TO – technologický objekt
- TO Speed Axis – česky TO Rychlostní osy, technologický objekt využívající relativního pohybu,
- TO Positioning Axis, česky TO Polohovací osy, technologický objekt využívající absolutního pohybu
- TO Synchronous Axis – česky TO Synchronní osy, technologický objekt kopírující pohyb přidružené TO Polohovací osy
- TO Kinematics – česky TO Kinematiky, nadřazený technologický objekt víceosého pohonu

ÚVOD

Použití programovatelných logických automatů stejně jako dalších mikrokontrolerů je dnes vskutku široké. Stále větší podíl automatizace na výrobních a technologických procesech přináší stále nové výzvy v této oblasti. S většími požadavky na řízení přichází také nové možnosti uplatnění PLC zařízení. Logické automaty tak najdeme ve stále větším množství odvětví.

Řízení elektrických pohonů mezi nimi bezpochyby najdeme. Výhoda PLC zařízení spočívá v modularitě a rozsahu použití, které však zároveň poskytuje dostatečnou míru individuální realizace v rámci PLC programování.

Návrh řízení XY ploteru má za cíl ukázat implementaci složitější aplikace na víceosém pohonu. Synchronizace pohybu víceosých pohonů vyžaduje specifický návrh řízení, společně s nadřazenou aplikací automatického vykreslování práce ukazuje, kam sahají možnosti PLC zařízení při snaze o maximální využití dostupných programovacích nástrojů.

V Praze dne 28.4.2023

Jiří Marek

HISTORIE PRŮMYSLOVÝCH AUTOMATŮ

Programovatelný logický automat stojí za zrodem automatizace, jak ji dnes známe. Éře PLC automatů předcházely reléové skříně. Jednalo se v podstatě o fyzické obvody, které masivně používaly spínací prvky, především pak relé, z toho vznikl název pro reléové skříně. Logika, tedy soubor instrukcí, byl vytvořen fyzickým zapojením velkého počtu součástek. Hlavním úkolem těchto obvodů bylo řízení spínání at' už elektrických nebo hydraulických strojů. Problém však nastal v případě složitější aplikace. Jak vyplývá z principu, složitější logická operace obsahovala velký počet fyzických součástek, což představovalo nevýhodu, neboť takové reléové skříně již zabíraly velkou část prostoru řízeného procesu. Největší problém však představovalo pevné propojení součástek. Tzv. pevná logika prakticky znemožňovala jakékoli rozsáhlejší úpravy již provedené logické aplikace. Nejenom, že možnost úpravy byla téměř nemožná, ale zároveň realizace nebyla modulární a každá aplikace vyžadovala specifickou realizaci již od počátku [1], [2].

S cílem nahradit reléové skříně a vyřešit jejich neduhy přišla na konci 60. let divize Hydramatic společnosti General Motors, jež vypsala soutěž s požadavkem na nový typ zařízení, které by reléové skříně nahradilo. Soutěž tak dala základ a stanovila první předpoklady, na kterých vznikl první programovatelný logický automat, dnes známý pod označením 084. Se zařízením přišel Richard Morley v roce 1968. Zasadil se tím o novou éru automatického řízení ve jménu programovatelných logických automatů. Programovatelné logické automaty oproti svému předchůdci v podobě reléových skříní vyřešily jejich nedostatky a přinesly do praxe velice pokrokové atributy [3], [4].

KAPITOLA 1: POPIS PLC

1.1 Úvod

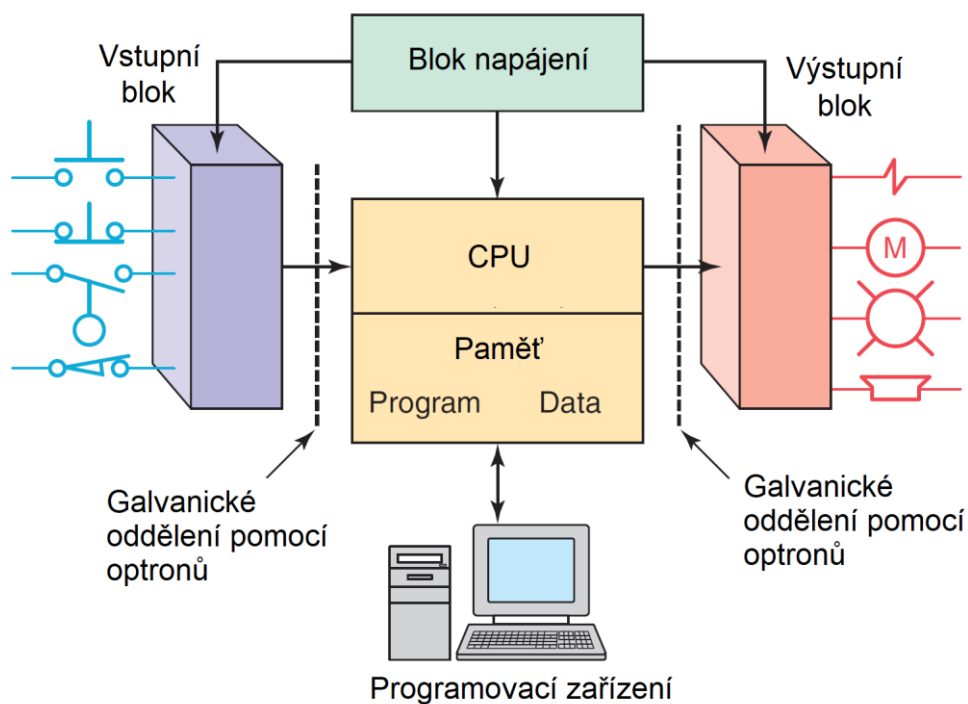
Tato kapitola se věnuje seznámení s oblastí automatizace a PLC zařízeními jako takovými. Je kladen důraz na přiblížení konceptu programovatelných logických automatů. Součástí kapitoly je popis PLC a jeho struktury společně se základním popisem funkce jednotlivých bloků obecného PLC zařízení. Snahou je, aby čtenář získal povědomí o úloze PLC zařízení.

Teoretický popis PLC zařízení byl již podrobně zpracován v mé bakalářské práci Model výtahu řízený PLC SIMATIC [5] a tato kapitola z něj z velké části vychází.

1.2 PLC

K pochopení pojmu PLC nám napoví již jeho název. PLC je pouze zkratka pro programovatelný logický automat v anglickém znění již v korespondenci se zažitou zkratkou *Programmable Logic Controller*. Zařízení je možné si představit jako programovatelný mikrokontrolér určený pro automatizaci, resp. samočinné řízení průmyslových, tedy především výrobních a technologických procesů. Zařízení v jádru vykonává cyklicky vytvořený program, který je uložen v paměti procesoru a vykonává dle reakce prostředí instrukce programu. Klasické operace zahrnují čtení, časování, spínání výstupních signálů apod. Na rozdíl od vestavěných systémů, které charakterizujeme jako jednoduché mikroprocesory zabudované přímo v aplikačním zařízení, jsou PLC zařízení často samostatné bloky připojené k aplikaci externě. To vystihuje možnost modularity a širokého pole využití. Při srovnání s klasickým PC naopak PLC zařízení vykonává program cyklicky a slouží k přímému řízení procesu. Periferie PLC jsou běžně připraveny jak pro logické, tak i výkonové ovládání aplikace [2], [6].

PLC se skládá ze základních stavebních bloků, jakými jsou zejména CPU, paměť, I/O blok, zdroj a komunikační rozhraní [6]. Podle [7] je nedílnou součástí také programovací zařízení. Programovací zařízení zajišťuje vytvoření programu pro PLC v dané úloze. Program je následně do zařízení nahrán a uložen v paměti. Programovací zařízení není běžně součástí hardwaru zařízení, ale figuruje jako aplikace ve vyšším programovacím jazyku.



Obr. 1.1 Struktura PLC (upraveno) [2]

1.3 Sestava PLC

PLC zařízení je koncipováno jako řídicí člen průmyslových aplikací. Jedná se především a o výrobní a technologické postupy. Klasickým případem může být řízení výrobní linky kupříkladu v pekárně. Je tedy zřejmé, že zařízení musí odolat i nepříznivým podmínkám včetně ochrany při havarijních stavech. To znamená, že zařízení musí zůstat provozuschopné i při velkém teplotním rozsahu, musí být dostatečně odolné vůči prachu, značné vlhkosti či vibracím. Stejný princip je uplatněn také na vliv prostředí z hlediska napájení přístroje. Předpokládá se větší riziko výskytu nebezpečných stavů jako je kupříkladu přepětí. Zařízení tak disponuje náležitými ochranami, které jsou mnohem robustnější, než najdeme u běžných spotřebních zařízení [2], [7].

V případě náročnějších aplikací se již nesetkáme s jednoduchou skříňí PLC zařízení, dle [7] tzv. *single box* provedení, ale již s modulárním typem provedení, tzv. *modular/rack type*. U jednoduššího provedení jsou všechny základní bloky součástí jedné přístrojové skříňě. Pokud aplikace například vyžaduje řízení více výstupních signálů, je třeba zvolit modulární typ provedení a k PLC zařízení nainstalovat samostatný I/O blok. Samostatné bloky jsou navzájem propojeny a zpravidla bývají nainstalovány na společném roštu (proto anglický název *rack*). Výhody této varianty spočívají především v přizpůsobení pro žádanou úlohu a možnosti modularity pro případ rozšíření nebo změny aplikace [2].



Obr. 1.2 Modulární typ PLC [7]



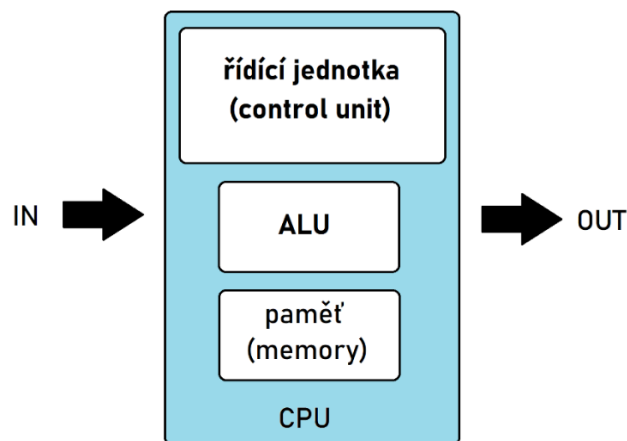
Obr. 1.3 Jednoduchý typ PLC [8]

1.4 CPU

CPU neboli centrální procesorová jednotka (častěji používaný název procesor) čte hodnoty vstupních signálů, realizuje soubor příkazů řídicí logiky a následně vysílá výstupní signály do akčních členů ovládané úlohy. CPU se obecně skládá z registrů sloužících jako krátkodobá paměť, řídicí jednotky nebo také řadiče a aritmeticko-logické jednotky (ALU). Zpravidla CPU tvoří jeden, nebo více mikroprocesorů [4].

Práce CPU, tedy vykonávání souboru instrukcí programu, je řízena hodinami s běžnou frekvencí mezi 1 MHz až 8 MHz. Hodinová frekvence určuje rychlost, s jakou PLC může vykonávat zadaný program, časuje tok programu a zajišťuje synchronizaci všech bloků PLC [8, s. 39,40]. Hodiny jsou součástí tzv. *control unit*, neboli řídicí jednotky.

Aritmeticko-logická jednotka, zkráceně nazývaná ALU, zpracovává data a vykonává základní aritmetické operace (sčítání, odčítání) a logické operace AND, OR, NOT a XOR [9], [7].



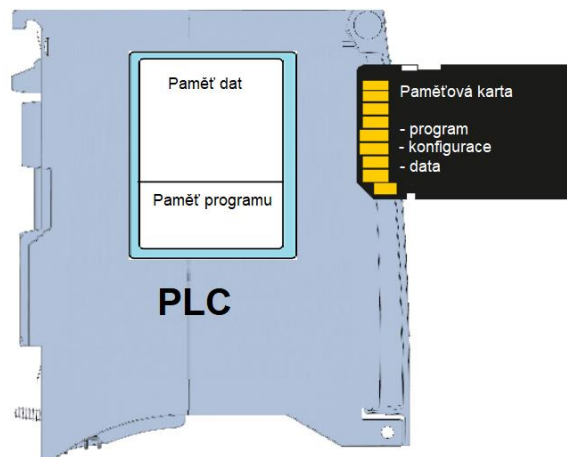
Obr. 1.4 Struktura CPU

1.5 Paměť

Paměť je důležitou součástí každého PLC zařízení. Slouží jak k uložení souboru instrukcí pro CPU a krátkodobému uchování dat ze vstupních kanálů, tak i pro uložení dat a nastavení pro žádanou aplikaci. Paměť samotného PLC automatu většinou rozdělujeme na dvě části: paměť programu a paměť dat [7].

Běžně však najdeme u PLC zařízení také slot pro paměťovou kartu. Jedná se o tzv. *load-memory*, ve volném překladu ji můžeme označit jako nahrávací paměť. Ta představuje externí datové úložiště, které není součástí zařízení, ale většina aplikací se bez ní neobejde. Paměťová karta slouží k uložení všech datových souborů aplikace. Pouze kritické části programu jsou dále přehrány do paměti zařízení a rozděleny mezi paměť dat a programu. Mezi kritické datové soubory patří soubor instrukcí pro CPU, resp. samotný program, hardwarová a softwarová konfigurace sestavy nebo datové soubory, ke kterým je potřeba přistupovat rychle během cyklického vykonávání programu [10].

Paměťová karta funguje z pohledu aplikace jako permanentní úložiště, a proto jsou všechna data nejdříve uložena zde. V případě ztráty napájení jsou důležité části paměti opět nahrány do CPU, resp. do paměti zařízení [10].



Obr. 1.5 Struktura paměti (upraveno) [10]

1.5.1 Paměť programu

Paměť sloužící k uchování programu musí zajistit permanentní uložení dat, vyžadujeme tedy, aby byl její obsah uchován i při ztrátě napájení. Paměť programu je běžně vázána na použitý procesor v zařízení a je dále neměnitelná [2], [6].

1.5.2 Paměť dat

Paměť dat je využívána k uložení dat ze vstupních kanálů, hodnot čítačů nebo časovačů. Data a hodnoty jsou tvořeny během běhu programu a zpracovány v CPU pro další výkon programu. Zde už předpokládáme ztrátu dat bez napájení. Výhodou nestálého uložení dat je možnost použití takového typu paměti, která disponuje rychlejším přístupem k datům, zpravidla paměť typu RAM [6], [7].

1.6 Blok napájení

Blok napájení zajišťuje napájení zařízení, resp. jeho částí požadovanou napětovou hladinou. Vzhledem k výkonové povaze vstupně-výstupního I/O bloku disponuje PLC zařízení několika napětovými hladinami, které je třeba během provozu napájet. Zpravidla zde máme vnitřní procesorovou CPU jednotku napájenou klasicky 5 V DC. Ovšem v případě I/O bloku zde běžně máme více napětových hladin. Konkrétní parametry se liší použitým I/O blokem. Typicky se setkáme s 24 V DC napájením I/O bloku [2], [6], [7].

1.7 I/O blok

Input/Output (zkráceně I/O) blok zajišťuje přenos informací mezi CPU a vstupními a výstupními kanály. Obecně zpracovává informace ze vstupních signálů. Informace jsou dále vyhodnoceny CPU, na základě provedení programu zařízení posílá řídicí signály pomocí výstupních kanálů akčním členům aplikace. Výstupní signály jsou prostřednictvím stejného bloku transformovány tak, aby jimi mohly být řízeny požadované akční členy [7].

Signály, vstupní i výstupní, mohou být dvou typů, analogové, nebo diskrétní [8, s. 39].

1.8 Komunikační rozhraní

Komunikační rozhraní PLC má za úkol posílat a přijímat data z připojených zařízení k PLC [7]. Důležitým zařízením, které k PLC připojujeme, je programovací zařízení (anglicky *programming device*). Kromě programovacího zařízení dnešní PLC běžně umožňují připojit další zařízení sloužící pro rozšíření jejich funkce stejně jako známe z osobních počítačů. Příkladem může být třeba ovládací panel HMI, tiskárna nebo rozhraní SCADA [6].

1.8.1 HMI panel

HMI panel, *human machine interface*, představuje komunikační rozhraní, které zajišťuje dohled a nadřazené řízení žádané aplikace. Panel zprostředkovává důležité informace obsluze zařízení a umožňuje jí aplikaci ovládat do takové míry, jak je HMI panel naprogramován [11].

Většina dnešních HMI panelů je, podobně jako PLC zařízení, zcela programovatelná a schopná téměř kompletně řídit konkrétní aplikaci včetně diagnostiky a kontroly řídicích prvků. Programování HMI panelů je nedílnou součástí každého projektu obsahující toto zařízení a zpravidla ji nazýváme vizualizací. Jedná se jak o naprogramování řízení, kontrolu či diagnostiku úlohy, tak také vizualizaci aktuálních parametrů úlohy.

KAPITOLA 2: POČÍTAČOVÉ ŘÍZENÍ VÝROBY CAD/CAM/CNC

2.1 Technologie CAM

Computer Aided Manufacturing, ve zkratce CAM, je možné volně přeložit jako počítačem řízená výroba. Nejde přitom však o oblast automatizace a snahu nahradit jednoduché opakující se úkoly výrobních či technologických procesů. CAM můžeme popsat jako spolupráci počítačového softwaru a výrobního zařízení. Počítačový program má za úkol řízení výrobního zařízení s cílem vyprodukovat požadovaný výrobek.

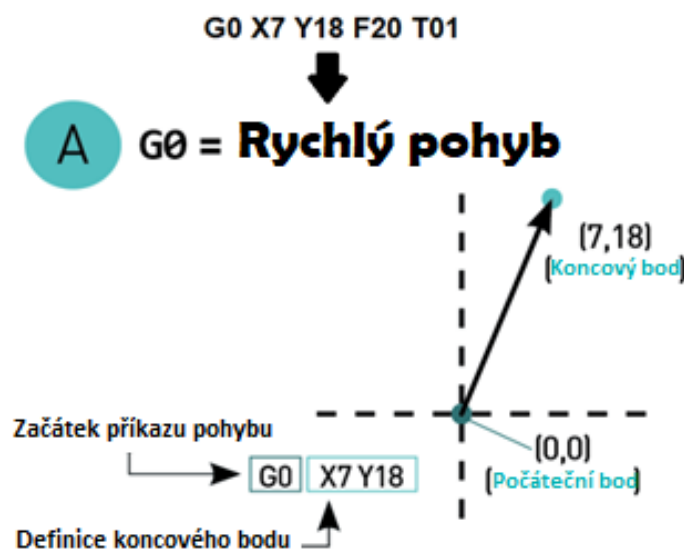
K pochopení pojmu CAM použijme běžnou oblast, kde je jeho využití opravdu obrovské, a to CNC obrábění. Cílem procesu je vyrobit určitý výrobek. Namísto řízení stroje operátorem dle přiloženého výkresu výrobku, je CNC stroj s využitím technologie CAM řízen automaticky na základě počítačového návrhu výrobku. Počítačový návrh výrobku je realizován na základě tzv. CAD systému (Computer Aided Design), který je neodmyslitelně spjat s technologií CAM, potažmo CNC stroji. Výrobní proces začíná s návrhem produktu systémem CAD a pokračuje jeho vyhotovením s využitím technologie CAM, která dokáže soubor s počítačovým návrhem zpracovat do takové formy, aby ho CNC zařízení mohlo vyrobit. CAM software transformuje návrh z pohledu geometrického, resp. prostorového uspořádání, připraví souřadnicový systém a vygeneruje soubor příkazů, které reprezentují příkazy pro samotné výrobní zařízení, tedy především dráhy jeho pohybu, ale také akceleraci nebo příkazy k odebrání materiálu, pokud se jedná například o obráběcí stroj. Vždy záleží na konkrétní výrobní aplikaci [12], [13].

2.1.1 G-code programovací jazyk

Výstupem CAM softwaru je soubor instrukcí, který obsahuje všechny potřebné informace o výrobním procesu v požadovaném strojním jazyce, tzv. G-code soubor. Jedná se o nejrozšířenější CNC programovací jazyk používaný v počítačem řízené výrobě. Běžně se s ním setkáme u 3D tisku, obráběcích strojů či obecně u velkého množství výrobních zařízení řízené počítačem.

Abychom princip CAM technologie lépe pochopili, je dobré se podívat na G-code podrobněji. Programovací jazyk G-code je zcela odlišný od běžných vyšších programovacích jazyků známé z procedurálního programování. Je to dáno tím, že G-code je velice jednoduchým jazykem. Jeho programování má předem stanovenou strukturu a obsahuje pouze omezené množství příkazů. Zároveň nemá za úkol řešit složitější procesy, nýbrž pouze řídí a nastavuje pohyb a provoz výrobního zařízení [13].

Na následujícím obrázku je zobrazen příklad G-code programu.



Obr. 2.1 Instrukce G-code (upraveno) [14]

První slovo je nejdůležitější částí příkazu. Nutno uvést, že jeden příkaz odpovídá jednomu řádku. G01 popisuje typ pohybu a říká, co má zařízení udělat. Následují souřadnice koncového bodu pohybového příkazu. Další části řádku už souvisí s aktuálním nastavením zařízení během výkonu příkazu. Část začínající na písmeno F souvisí s nastavením rychlosti pohybu. Dále je možné nastavit příslušenství, kterým zařízení disponuje a zároveň ovlivňuje výrobní proces. Další příkazy mohou být zcela individuální, resp. přizpůsobené jak pro konkrétní zařízení, tak i pro konkrétní aplikaci. Primární částí však zůstávají příkazy pro dráhu pohybu dané souřadnicemi. Na závěr dodejme, že program je vykonáván postupně v pořadí, které odpovídá daným řádkům [14].

2.2 CAD

Hojně používaná zkratka CAD stojí za názvem technologie Computer Aided design, tedy ve volném překladu počítačem řízený návrh. Český překlad slova design však může být lehce zavádějící vzhledem k tomu, že v českém jazyce neexistuje vhodné slovo reprezentující dané označení do stejné míry jako v jazyce anglickém. V případě technologie CAD slovo design hraje roli v modelování, tvoření nebo, chcete-li, návrhu určitého modelu. Důležitý prvek charakterizující technologii CAD je skutečnost, že celý proces probíhá s využitím výpočetních zařízení, resp. počítačových aplikací. V případě návrhu obecně trojrozměrných modelů reálných produktů či výrobků nám tak počítače obrovsky usnadňují práci a umožňují snadnou realizaci požadovaných modelů. Dnešní aplikace CAD technologie navíc umožňují vývoj a testování přímo s využitím počítačového modelu bez nutnosti vyrábět reálné prototypy produktů [12], [13].

Zastavme se u dalšího slova, a to model. Opět je nasnadě vysvětlit, co za slovem model stojí v technologii CAD. Za předpokladu, že chceme například pro výrobní proces navrhnout určitou součástku, potřebujeme její návrh, tedy dostatečně přesný souhrn informací. Tomu v technologii CAD říkáme model. Během návrhu nepracujeme s reálnou součástkou, ale její počítačovou reprezentací, modelem.

2.3 CNC stroj

Poslední částí procesu je samotný CNC stroj. Označení CNC stroj vychází z anglické zkratky Computer Numerical Control. To můžeme přeložit jako stroj řízený počítačem. Jedná se v zásadě o automatické řízení stroje pomocí mikroelektroniky. Pro nás je důležité, že CNC zařízení je připraveno k zpracování G-code souboru a na základě jeho příkazů provést požadovaný proces [12].

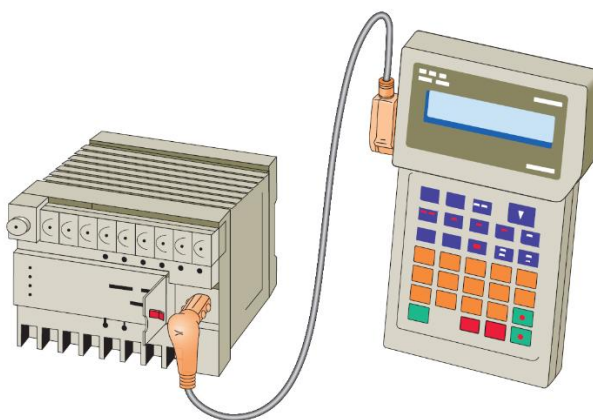
KAPITOLA 3: PROGRAMOVÁNÍ PLC

Funkce programovatelného logického automatu, zkráceně PLC, spočívá v cyklickém vykonávání souhrnného souboru instrukcí, který na základě vstupních parametrů získaných ze vstupních signálů nebo komunikačních rozhraní, ovládá akční členy aplikace. Soubor instrukcí je nutné však pro každou úlohou s ohledem na požadovanou funkci vytvořit. K realizaci souboru instrukcí, často zaměňujeme za program, slouží programovací zařízení. Programovací zařízení kromě vytvoření programu ovšem zajišťuje dále přeložení a nahrání spustitelného programu do zařízení. Při návrhu programu pro PLC se zpravidla uplatňují specifické programovací jazyky. V této kapitole bude věnován prostor programovacím jazykům užívaným ve vývojovém prostředí TIA Portal od spol. Siemens [2], [7].

3.1 Programovací zařízení

Programovací zařízení slouží k tvorbě uživatelského programu dané aplikace a zároveň zajišťuje kompletní jeho obsluhu. Pod tím si představme kompilaci a kontrolu syntaxe či programových chyb včetně diagnostiky programu. Kompilátor tedy zkontroluje, zda je program spustitelný, zda neobsahuje kritické chyby, které by vedly k pádu programu či zda zařízení disponuje dostatečným místem pro nahrání souboru instrukcí do paměti zařízení. Po kontrole je program v rámci kompilace převeden do spustitelného kódu včetně konfigurace zařízení a provedení skutečného adresování programu.

Typickým dnešním příkladem programovacího zařízení je spustitelná aplikace na osobním počítači v běžně používaném operačním systému. Pokud se však jedná o jednodušší aplikaci, resp. jednodušší PLC zařízení (typicky jednoduché provedení), může být k zařízení přiložen kupříkladu programovací ovladač, který však má omezené programovací schopnosti [2], [3]. Příklad můžeme vidět na obr. 3.1.



Obr. 3.1 Speciální programovací ovladač pro PLC [2]

Programovací zařízení jsou typicky určena pro konkrétní typy PLC zařízení, nebo alespoň konkrétní výrobce. Kompatibilita mezi jednotlivými programovacími zařízeními není zaručena a jedná se spíše o výjimku. Je proto nutné si ověřit kompatibilitu, nebo nejlépe disponovat doporučeným programovacím zařízením pro vybraný typ programovatelného automatu [2].

3.2 PLC programovací jazyky v prostředí TIA Portal

Programy pro PLC musí být stejně jako u ostatních mikroprocesorových zařízení nahrány do zařízení ve formě spustitelného strojového kódu, neboť mikroprocesor pracuje s binárním kódem. Vytvořit program ve strojovém kódu je však velice náročné, a proto pro zjednodušení používáme programovací jazyky, které mohou být jak nižší (jazyk symbolických adres/žebříčkové schéma), tak i vyšší (C, Pascal, SCL apod.). Programovací jazyky umožňují používat předem připravené funkce tak, aby bylo možné snadněji vytvořit žádaný program [2], [7].

Zde budou přiblíženy celkem tři programovací jazyky použité pro návrh logického řízení XY ploteru v rámci projektu vytvořeného ve vývojovém prostředí TIA Portal od spol. Siemens. Cílem je snadnější porozumění následného popisu realizace návrhu řízení.

Konkrétně bude věnován prostor následujícím programovacím jazykům:

- Liniové schéma (LAD)
- Blokové schéma (FBD)
- Strukturovaný text (SCL)

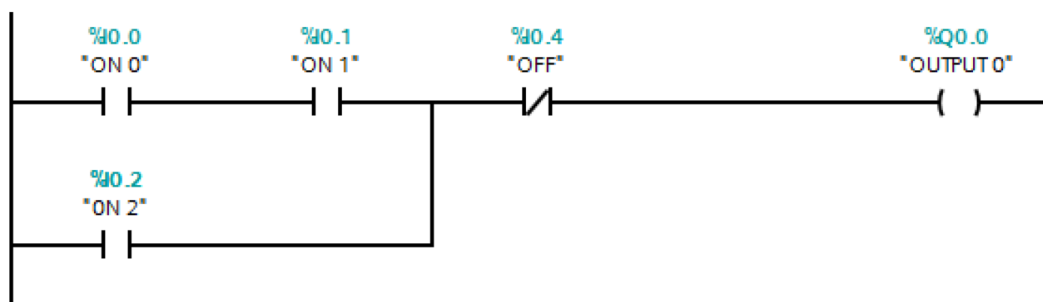
Je nutné podotknout, že všechny tyto programovací jazyky vycházejí z normy IEC 61131, zabývající se mikrokontrolery. Norma IEC 61131 stanovuje celkem pět základních programovacích jazyků. Společnost Siemens však v rámci programování svých PLC zařízení ve vývojovém prostředí TIA Portal používá lehce odlišné programovací jazyky. Tyto programovací jazyky však stále vycházejí ze základního členění dle normy IEC 61131. Většina výrobců PLC zařízení má vlastní vývojová prostředí pro své produkty. Tato vývojová prostředí mnohdy umožňují kombinovat více způsobů programování dohromady, což zajistí nejelegantnější realizaci programu na dané úrovni.

3.2.1 Liniové schéma (LAD)

Liniové schéma je zřejmě nejrozšířenějším programovým jazykem pro PLC. Liniové, často známé pod názvem žebříčkové schéma, z anglického znění *ladder diagram*, vychází z předchůdce logických automatů, neboť konstruuje reléovou logiku. Reprezentace programu je založena na obvodových schématech [8, s. 54].

Výhoda této metody programování spočívá především v její jednoduchosti. Nejenom, že přebírá stejné postupy, které se uplatňovaly dříve při návrhu reléových skříní, ale zároveň nevyžaduje pokročilejší znalosti programování (znalost vyšších programovacích jazyků) [7].

V případě použití liniového schématu ve vývojovém prostředí TIA Portal můžeme využít pro návrh také funkční bloky předem připravených funkcí pokrývající návrh řízení pomocí PLC. Tato skutečnost umožňuje využít žebříčkové schéma i pro složitější aplikace, které vyžadují náročnější příkazy typu čítání, přiřazení nebo počítání. Jedná se zpravidla o nejrozšířenější metodu návrhu.



Obr. 3.2 Liniové schéma

3.2.1.1 Struktura LAD

Žebříčkové schéma je vytvořeno na jedné nebo více horizontálních linkách, které jsou z levé strany připojeny k vertikální lince představující napájení. Napájení reprezentuje binární signál *ON/OFF*, který je spínací logikou přiveden k pravé části horizontální linky, kde se nachází grafické znázornění relé. Relé na konci linky představuje aktivaci požadovaného akčního členu. V projektu je akční člen běžně reprezentován PLC tagem. Výstup je v případě kladného vyhodnocení spínací logiky sepnut. Hodnota výstupního PLC tagu je potom změněn na hodnotu *ON*.

Horizontální linky čteme zleva doprava, proto na levé straně najdeme kombinaci spínačů představující podmínky a na pravé straně relé, které v případě vyhodnocení podmínky spínají akční člen procesu (například stykač motoru) [6].

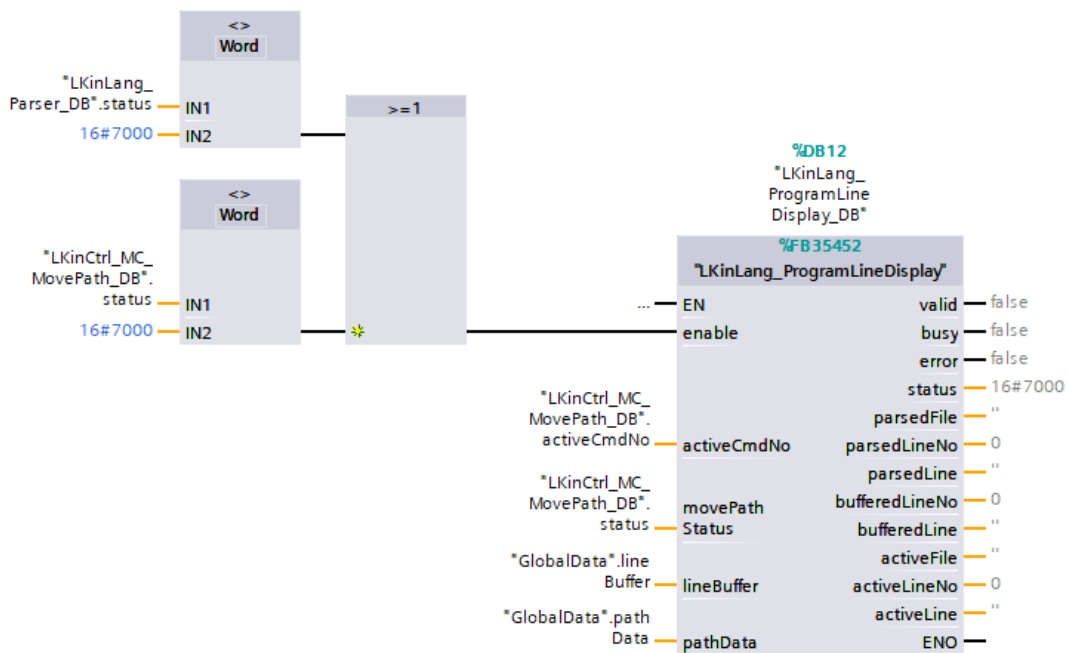
Z popisu vyplývá, že liniové schéma ve své čisté formě umožňuje realizaci pouze binárních signálů. Pokud je vyžadována složitější logika, přichází na řadu implementace funkčních bloků

složitějších příkazů, které svou logiku realizují na základě přivedení tzv. *napájení bloku* (ve vývojovém prostředí TIA Portal označeno jako EN).

3.2.2 Blokové schéma (FBD)

Blokové schéma, v anglickém originálu *function block diagram* (FBD), řadíme stejně jako liniové schéma mezi grafické programovací jazyky. Při realizaci uživatelského programu využíváme funkčních bloků, u kterých můžeme jejich funkci sami naprogramovat, nebo předem předpřipravených bloků. Klasickým příkladem základního bloku jsou logické funkce AND, OR a XOR [7].

Blokové schéma znázorňuje tok signálů dle logického vyhodnocení jednotlivých bloků. Výhodou blokového schématu je především možnost snadné realizace složitějších funkčních bloků, k jejichž realizaci můžeme využít podle potřeb jiných programovacích jazyků. Předdefinované bloky jsou ve vývojovém prostředí TIA Portal zpracovány ve strukturovaném textu SCL a jejich změna je chráněna [2].



Obr. 3.3 Blokové schéma

3.2.3 Strukturovaný text (SCL)

Metoda strukturovaného textu vychází z vyšších programovacích jazyků jako například C, C++ nebo Pascal. V případě vývojového prostředí TIA Portal spol. Siemens přišla s vlastní formou strukturovaného textu, který nazývá zkratkou SCL. Jedná se o strukturovaný text podobný vyššímu programovacímu jazyku Python. Používá klasické příkazy známé z vyššího programování jako například IF-ELSE, SWITCH nebo FOR cyklus. Největší odlišností v rámci syntaxe je adresování PLC proměnný, tzv. PLC tagů. Zde prostředí využívá svoje postupy a je třeba na to dbát zřetel při realizaci programu [4], [6].

Využití strukturovaného textu SCL vyžaduje stejně jako každý jiný strukturovaný text znalost vyšších programovacích jazyků, ze který částečně SCL vychází. V tomto případě postačí znalost programovacích jazyků C či Python. Nutnost znalosti vyššího programování představuje jistou nevýhodu této metody. Na druhou stranu umožňuje přehlednější a jednodušší realizace složitějších, především rozhodovacích struktur programu [4], [6].


```
1 IF #MovePathLang = TRUE THEN
2     IF "PENinst"."THIS"["LKinCtrl_MC_MovePath_DB".activeCmdNo] >= 3 THEN
3         "PEN" := FALSE;
4     END_IF;
5     IF "PENinst"."THIS"["LKinCtrl_MC_MovePath_DB".activeCmdNo] = 1 THEN
6         "PEN" := TRUE;
7     END_IF;
8 END_IF;
```

Obr. 3.4 Strukturovaný text

KAPITOLA 4: SESTAVA ZAŘÍZENÍ ÚLOHY

4.1 Hardware

4.1.1 SIMATIC S7-1500 CPU 1515TF-2 PN

SIMATIC S7-1500 je vyšší produktová řada PLC zařízení vyráběné společností Siemens. Jedná se o pokročilé řídicí průmyslové zařízení vhodné pro středně složitě až náročné technologické procesy. Vhodnou kombinací přidružených modulárních bloků je možné pokrýt širokou oblast použití jako například specifická strojírenská výroba (navíječky apod.), procesy potravinářské výroby nebo i laboratorní nástroje a pomůcky [15]. Konkrétní typ 1515TF-2 PN je vyšším modelem řady S7-1500. To znamená větší výpočetní výkon CPU jednotky s operační pamětí 750 kB pro program a až 3 MB pro data. Jedná se o modulární typ PLC s vestavěným ovládacím panelem a displejem k zajištění základních funkcí a diagnostiky zařízení. Zařízení vyžaduje stejnosměrné napájení 24 V DC. Paměťová karta SIMATIC Memory Card je vyžadována k chodu zařízení. Velikost paměťové karty může být až 32 GB. Zařízení disponuje dvěma zabudovanými rozhraními, prvním je PROFINET IRT s dvěma porty (X1 a X2) a druhým rozhraním Ethernet. K zařízení je možné připojit až 248 vstupně-výstupních signálů s vhodným modulárním I/O blokem [16].

Technické údaje zařízení jsou obsahem přílohy A.1.

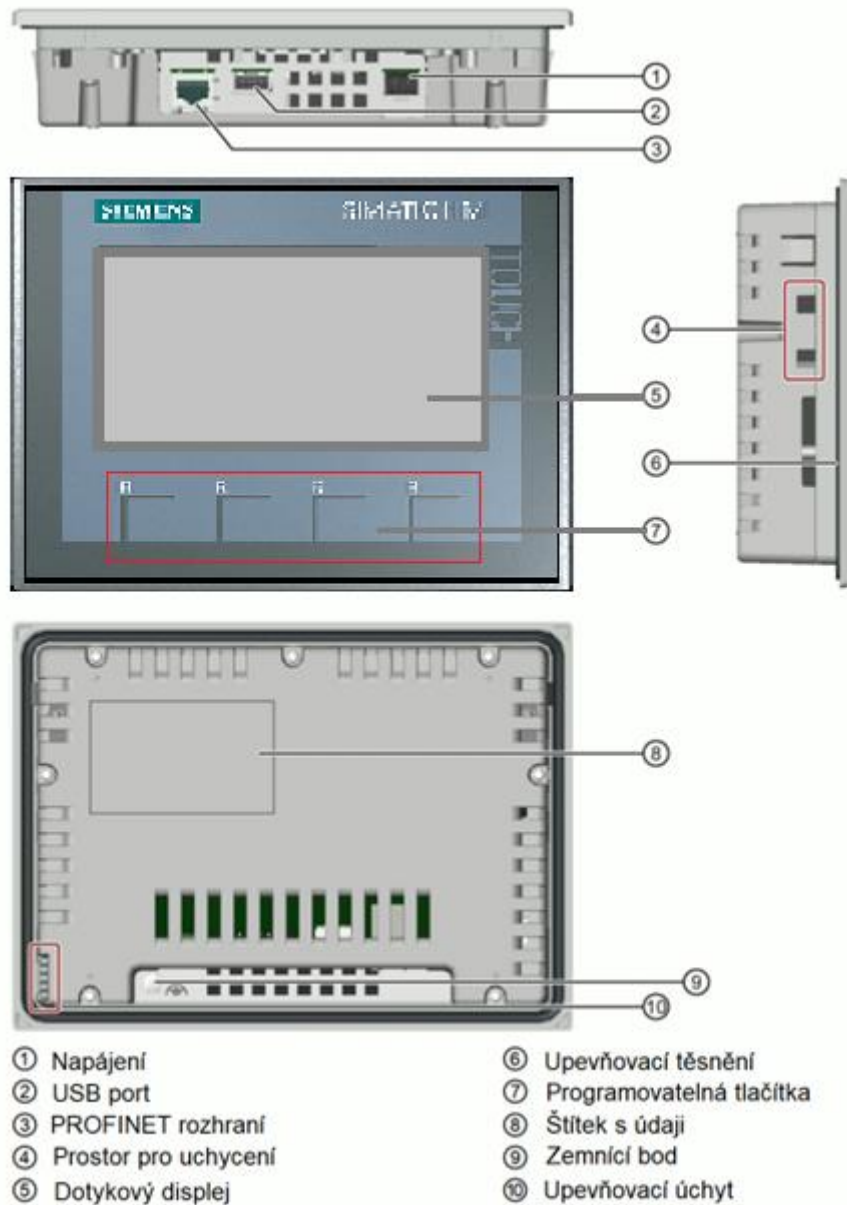


Obr. 4.1 PLC zařízení Siemens SIMATIC S7-1500 CPU 1515TF-2 PN [17]

4.1.2 Siemens SIMATIC HMI KTP400 Basic

HMI KTP400 panel je zařízení s 4" barevným TFT dotykovým displejem a čtyřmi programovatelnými tlačítky pod displejem. Zařízení disponuje komunikačním rozhraním PROFINET. Na zařízení najdeme ethernetový a USB vstup. Jedná se o základní model určený pro malé a střední automatizace ve spojení s SIMATIC S7-1500 [18]. Konfigurace je možná s využitím přidružených programů WinCC Basic V13 nebo STEP 7 Basic V13. V případě naší úlohy použijeme vývojové prostředí TIA Portal, která tyto programy sdružuje v jedno univerzální vývojové prostředí schopné realizace kompletního projektu [18], [19].

V úloze slouží k základnímu ovládání ploteru, což zahrnuje zapínání, vypínání a manuální pohyb zařízení včetně možnosti měnit barvu či přítomnost pera. Technické údaje HMI panelu jsou přiloženy v příloze A.2.



Obr. 4.2 Popis HMI panelu Siemens SIMATIC HMI KTP400 Basic (upraveno) [20]

4.1.3 Siemens SIMATIC S7-1500 digital input/output module

Jedná se o modulární blok digitální vstupně-výstupní DI/DO blok umožňující připojit až 16 vstupních signálů a 16 výstupních signálů. Napěťová hladina modulu je 24 V DC. Zařízení umožňuje ochranné odpojení signálů SIL2 dle normy IEC 62061:2021 a *Category2 / PL c* dle normy ISO 13849-1:2015. Proudové rozložení je v případě vstupu 4 A na jeden vstup a v případě výstupů 4 A na dva výstupy [21].

DI/DQ blok slouží k připojení vstupně-výstupních signálů přípravku. Jedná se o čtení koncových spínačů a spínání jak pohonů pera, tak i zásobníku. Technické údaje zdroje jsou obsahem přílohy A.3.



Obr. 4.3 DI/DQ blok Siemens SIMATIC S7-1500 DI/DQ module [21]

4.1.4 Siemens SITOP PSU100S power supply module

Siemens SITOP PSU100S je stabilizovaný zdroj napájení. Úkolem je transformace jednofázového napájení 230/120 V AC na výstupní stejnosměrné napájení 24 V DC pro sestavu zařízení. Volba napěťové hladiny probíhá automaticky. Ze vstupní strany je doporučen jistič od hodnoty proudu 6 A s charakteristikou C. Na výstupní straně je možné připojit dvě spínatelné jednotky [22].



Obr. 4.4 Napájecí blok Siemens SITOP PSU100S power supply module [22]

4.1.5 Siemens SIMATIC HMI KP900 Comfort

Siemens SIMATIC HMI KP900 Comfort je středně velký ovládací HMI panel. Jak vyplývá z názvu, jedná se o zařízení z vyšší produktové řady Comfort HMI panelů spol. Siemens. To znamená, že zařízení je možné využívat pro složitější aplikace vyžadující speciální okna k řízení programových elementů. Zařízení disponuje 9" barevným TFT, který ovšem na rozdíl od předchozího HMI panelu není dotykový. Kolem displeje jsou tedy umístěna důležitá ovládací tlačítka společně se standardními programovatelnými tlačítky a klávesnicí. K plnohodnotnému ovládnutí vizualizace je vhodné však připojit k zařízení například počítačovou myš. Zařízení využívá operačního systému Windows CE 6.0 a podporuje PROFINET rozhraní. Jeho konfigurovatelná paměť činí 12 MB. Na zařízení najdeme celkem dva Ethernetové porty a dva porty USB [23].

Technické údaje jsou obsaženy v příloze A.4.



Obr. 4.5 HMI panel Siemens SIMATIC HMI KP900 Comfort [24]

4.1.6 Siemens SINAMICS S210

Siemens SINAMICS S210 je měnič střední řady od spol. Siemens. Slouží k řízení a napájení střídavých servomotorů 1FK2 [19]. Komunikace s řídicím PLC zařízením probíhá pomocí komunikačního rozhraní PROFINET. Měnič SINAMICS S210 je navrhnut speciálně pro použití se servomotorem SIMOTICS S-1FK2. Z tohoto důvodu zde najdeme speciální OCC MOTION-CONNECT kabel mezi motorem a měničem. Toto konkrétní spojení zajišťuje jak komunikaci a řízení motoru, tak i napájení [25].

SINAMICS S210 měnič umožňuje IRT komunikaci s nadřazeným PLC zařízením. Měničem můžeme tedy kompletně řídit zvolenou osu, resp. motor. K naladění servopohonu je k dispozici také funkce One Button Tuning s automatickým naladěním regulační struktury pohonu [26].

Technické údaje jsou obsaženy v příloze A.5.

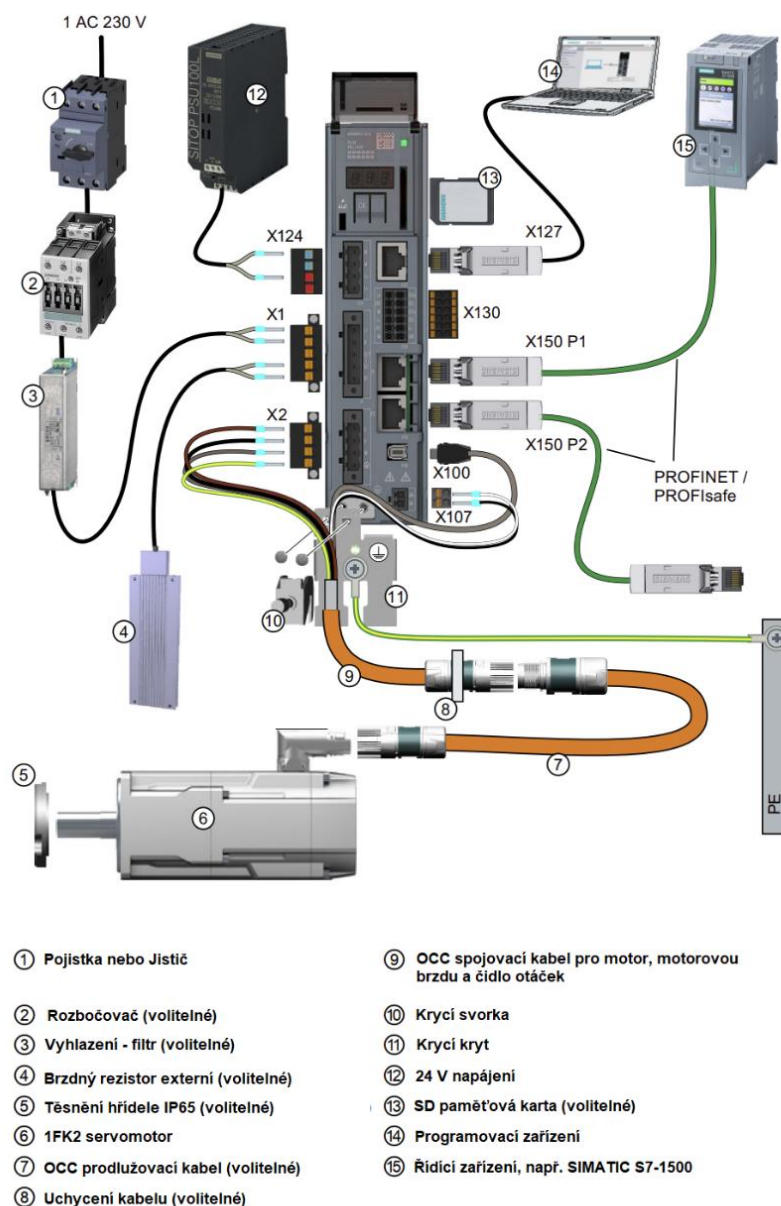


Obr. 4.6 Měnič Siemens SINAMICS S210 PN [25]

4.1.7 Siemens SIMOTICS S-1FK2 HD

Jedná se o základní servomotor od spol. Siemens na principu synchronního motoru s buzením permanentními magnety. Servomotor dosahuje jmenovitým výkonem 0,1 kW, jmenovitými otáčkami 3000 ot/min a jmenovitým momentem 0,16 Nm. Součástí zařízení je absolutní enkodér AM22DQC, 22-bitový + 12-bitový víceotáčkový. Vzhledem k tomu, že provedení motoru splňuje krytí IP64, je zařízení zcela prachotěsné a chráněno proti stříkající vodě ze všech úhlů [27]. Může být tak použit v náročnějších průmyslových aplikacích. Maximální otáčky činí 8000 ot/min, maximální proud pak činí 3,1 A a maximální moment potom 0,56 Nm. Servomotor je navržen pro použití převážně s měničem SINAMICS S210 a dohromady tvoří servopohon určený pro středně náročné aplikace [28].

Na obr. 4.7 je zobrazen popis sestavy servopohonu v čele s měničem SINAMCS S210 a servomotorem SIMOTICS S-1FK2. Sestava je řízena nadřazeným PLC zařízením a napájena z běžné sítě 230 V AC.



Obr. 4.7 Sestava řízeného servopohonu Siemens SINAMICS S210 a Siemens SIMOTICS S – 1FK2 (upraveno) [28]

4.2 Software

4.2.1 Siemens TIA Portal

Totally Integrated Automation Portal, zkráceně TIA Portal, je software od společnosti Siemens určený pro návrh, vývoj a dohled nad projekty určené pro automatizaci, resp. aplikacemi těchto projektů. TIA Portal vznikl jako zastřešující vývojové prostředí s cílem nahradit všechny dosavadní aplikace do jedné. Myšlenka aplikace TIA Portal je založena na integraci několika již dostupných softwarových nástrojů v čele s STEP7, WinCC, SINAMICS Startdrive a SIMOTION SCOUT TIA do jednoho vývojového prostředí s přehledným, a především jednotným ovládáním. Ke kompletnímu návrhu a realizaci projektu včetně údržby a diagnostiky již implementované aplikace postačí uživateli pouze prostředí TIA Portal. Prostředí TIA Portal umožňuje pracovat jak na technické části pohonů a návrhu logického řízení projektu, tak také umožňuje kompletní dohled nad technologickými objekty nebo vizualizací aplikace skrze HMI panely. Důležitou část představuje tzv. *Commisioning* a *Online* přístup k projektu. [29], [30].

4.2.2 SIMATIC File Transfer tool

SIMATIC File Transfer je doplňková softwarová aplikace běžící na osobním počítači s operačním systémem Windows, resp. programovacím zařízením se systémem Windows, které je připojeno k PLC zařízením prostřednictvím lokální LAN sítě. File Transfer tool je součástí externí knihovny Kinematics Language pro TIA Portal programování. Program umožňuje přenos datových souborů z osobního počítače na paměťovou kartu připojeného SIMATIC PLC. File Transfer tool tak zajišťuje správu a přenos souborů mezi připojeným PLC zařízením SIMATIC a osobním počítačem. S vhodnou konfigurací tak nahrazuje složitější přístup skrze WEB server službu, kterou SIMATIC PLC také umožňuje [31].

Přenos probíhá na základě stejného principu jako prostřednictvím WEB server služby, a to pomocí internetového protokolu http včetně možnosti zabezpečení přenosu s využitím protokolu HTTPS [31].

KAPITOLA 5: POPIS PŘÍPRAVKU

Předmětem práce je návrh logického řízení XY ploteru společně s návrhem implementace automatického vykreslování CAD modelů pomocí technologie CAM včetně zpracování CAM technologie do logiky Siemens SIMATIC PLC projektů. Cílem úlohy je funkční aplikace na předpřipraveném přípravku XY ploteru, který je zobrazen na obr. 5.1.

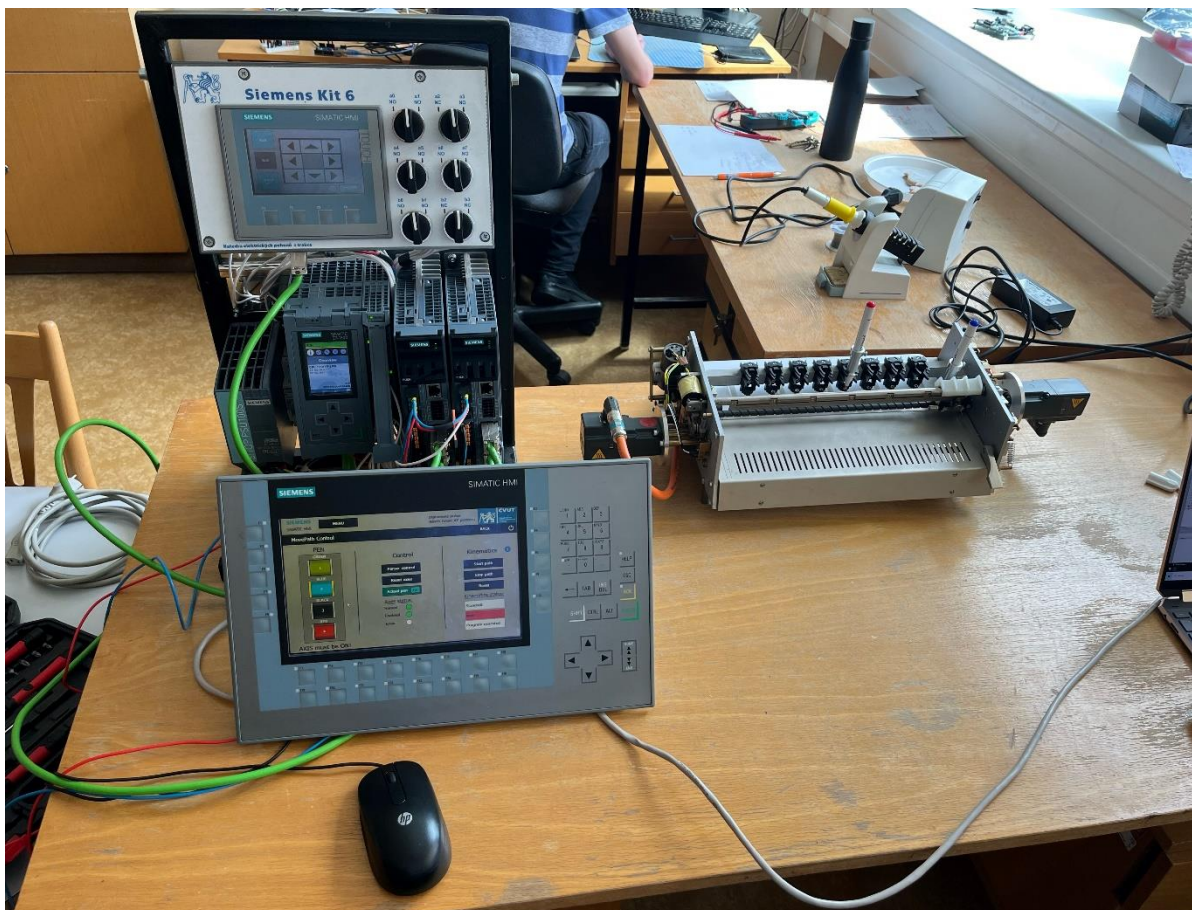
K přípravku byla připojena sestava zařízení popsaná v kapitole 4 tak, aby byl maximálně využit potenciál původního zařízení. Sestava zahrnuje zabudované dva servomotory SIMOTICS 1-FK2 řízené dvěma měniči SINAMICS S210 PN, měniče jsou dále nadřazeně obsluhovány pomocí PLC zařízení SIMATIC S7 1515TD-2 PN. PLC sestavu tvoří také celkem dva HMI panely KTP400 Basic a KP900 Comfort. Větší z panelů, KP900 Comfort, bude mít za úkol obsluhu automatického vykreslování pomocí technologie CAM a menší panel základní funkce přípravku.

Přípravek slouží k tvorbě požadovaných výkresů na vloženém papíru. Logické řízení ploteru předpokládá využití pouze formátu papíru A4 nebo A3, s možností jeho volby. Hlavní pohyb pera po papíru je tvořen dvouosým pohonem. Vertikální osa při pohledu zepředu (pohled totožný s obr. 5.2) zajišťuje pohyb pera, horizontální osa zajišťuje pohyb papíru. K zajištění pohybu obou os je přípravek osazen dvěma motory SIMOTICS 1-FK2, jejich popis je uveden v kapitole 4. V případě vertikální osy pera je využito principu šroubového pohybu pro pohyb uchycení pera na hřídeli. Zároveň je vertikální osa pera natáčena za účelem přiložení a zvednutí pera z papíru. O tento pohyb se stará spínací elektromagnet využívající 24 V napájecí zdroj. Kromě elektromagnetu najdeme na přípravku sadu cívek a kondenzátorů společně s rezistory, které mají zajistit dostatečnou dynamiku přiložení pera. K pohybu papíru se využívá rotujících válečků. Na každé straně je pod vloženým papírem umístěn jeden váleček. Válečky jsou uloženy na společné hřídeli. Pomocí rotace válečků je papír uveden do posuvného pohybu a společně s vertikální osou tak umožňuje pohyb pera po celém povrchu papíru. Rychlost posunu jak papíru, tak i pera činí 10 mm/ot.

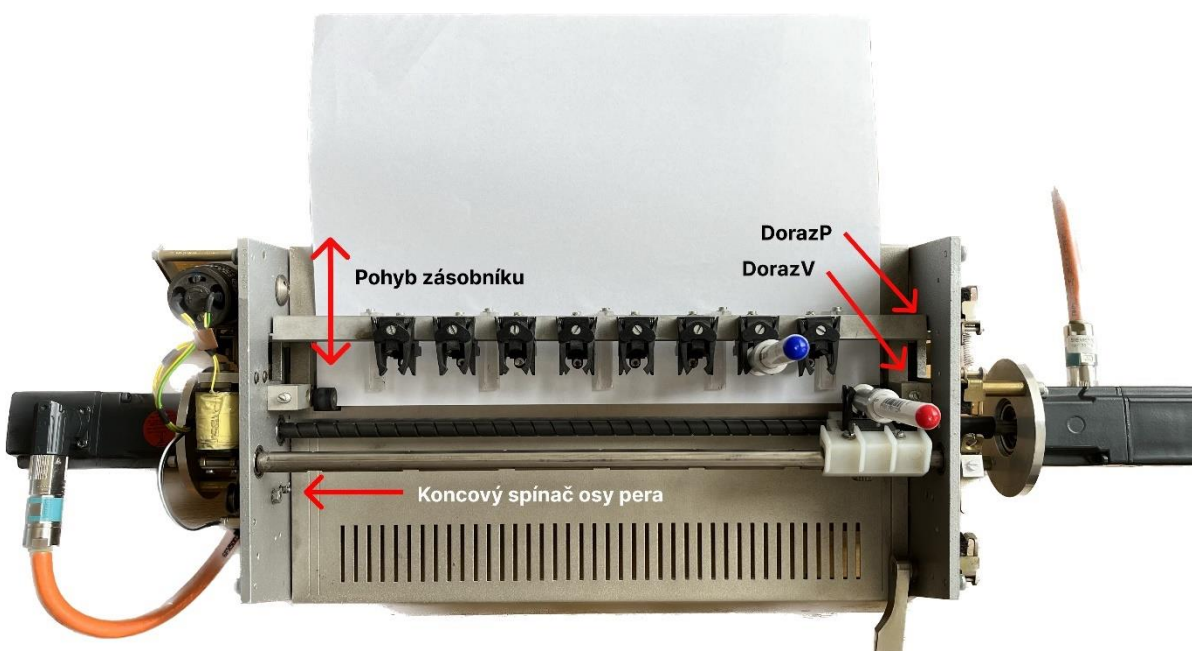
Výměna je možná pouze při dodržení správného postupu. Polohu zásobníku dostatečně přesně signalizují dva koncové spínače DorazV a DorazP. Oba spínače jsou sepnuty, pokud se zásobník nachází v mezipoloze. Za mezipoložu je považován okamžik mezi oběma krajními polohami, tedy když je zásobník uveden do pohybu. Rozepnutý spínač DorazV tak signalizuje polohu zásobníku v parkovací pozici a spínač DorazP potom informuje o poloze zásobníku v pozici pro výměnu pera. Pohyb zásobníku ovládáme spínacím relé MotorZOUT. Před uvedením zásobníku do pohybu je však nutné nejdříve nastavit směr pohybu zásobníku. K tomu slouží signál ZDir. Hodnotu *TRUE* (logická jednička) signálu ZDir nastavíme při pohybu zásobníku z parkovací polohy a hodnotu *FALSE* (logická nula) nastavíme při pohybu opačném. Směr pohybu je možné měnit pouze když je motor vypnutý, tzn. MotorZ má hodnotu *FALSE*. Jinak dochází k opalování přepínacích kontaktů ve spínacím relé motoru. Dále je nutné zajistit bezprostřední vypnutí motoru, jakmile se mechanicky zastaví na dorazu, abychom předcházeli chodu nakrátko a zbytečnému opotřebením motoru.

Zásobník řídíme pomocí logických vstupně výstupních signálů připojených k DI/DQ bloku, které je součástí sestavy PLC.

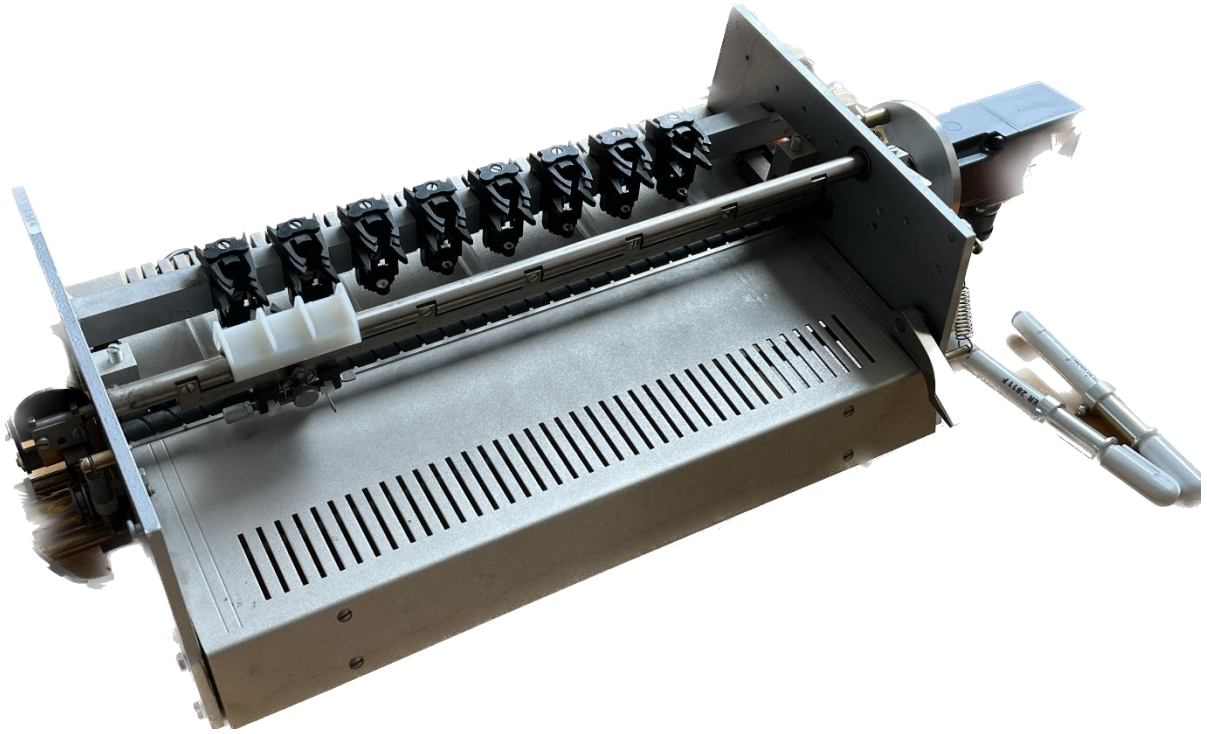
Poslední částí přípravku jsou signalizace vložení papíru do přípravku, dva dorazy pro určení polohy zásobníku během výměny pera a koncového spínače osy pohybu pera na levé straně. Tyto signalizace jsou dále v návrhu kritické pro tzv. *homming*, neboli zareferování os pro vytvoření požadovaného souřadného systému využívaného při automatickém vykreslování. Jedná se zejména o koncový spínač vertikální osy pera. Dorazy informující o poloze zásobníku per jsou pak důležitou částí sekvence příkazu pro úspěšnou výměnu pera ze zásobníku.



Obr. 5.1 Kompletní sestava zařízení



Obr. 5.2 XY ploter



Obr. 5.3 Detail přípravku XY ploteru

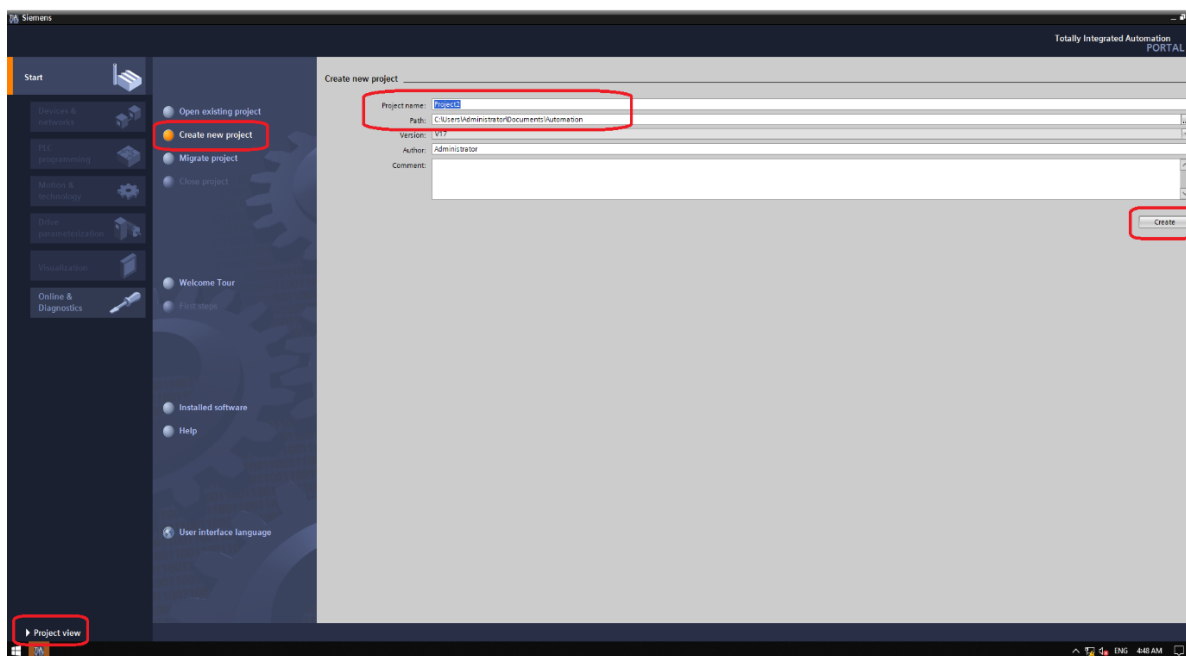
KAPITOLA 6: ÚVOD VE VÝVOJOVÉM PROSTŘEDÍ TIA PORTAL

V této kapitole je přiblížen začátek jakéhokoli projektu ve vývojovém prostředí TIA Portal. Budou zde nastíněny první kroky, které jsou třeba při vytváření projektu, včetně krátkého popisu struktury pracovního okna prostředí. Podrobnější popis je možné najít v mé bakalářské práci [5], kde je základům ve vývojovém prostředí TIA Portal věnován větší prostor a z níž následující odstavce vycházejí.

Vývojové prostředí TIA Portal nemá českou lokalizaci. Z tohoto důvodu se v textu objeví původní anglické výrazy, vždycky zpozorovatelné kurzívou.

6.1 Založení projektu

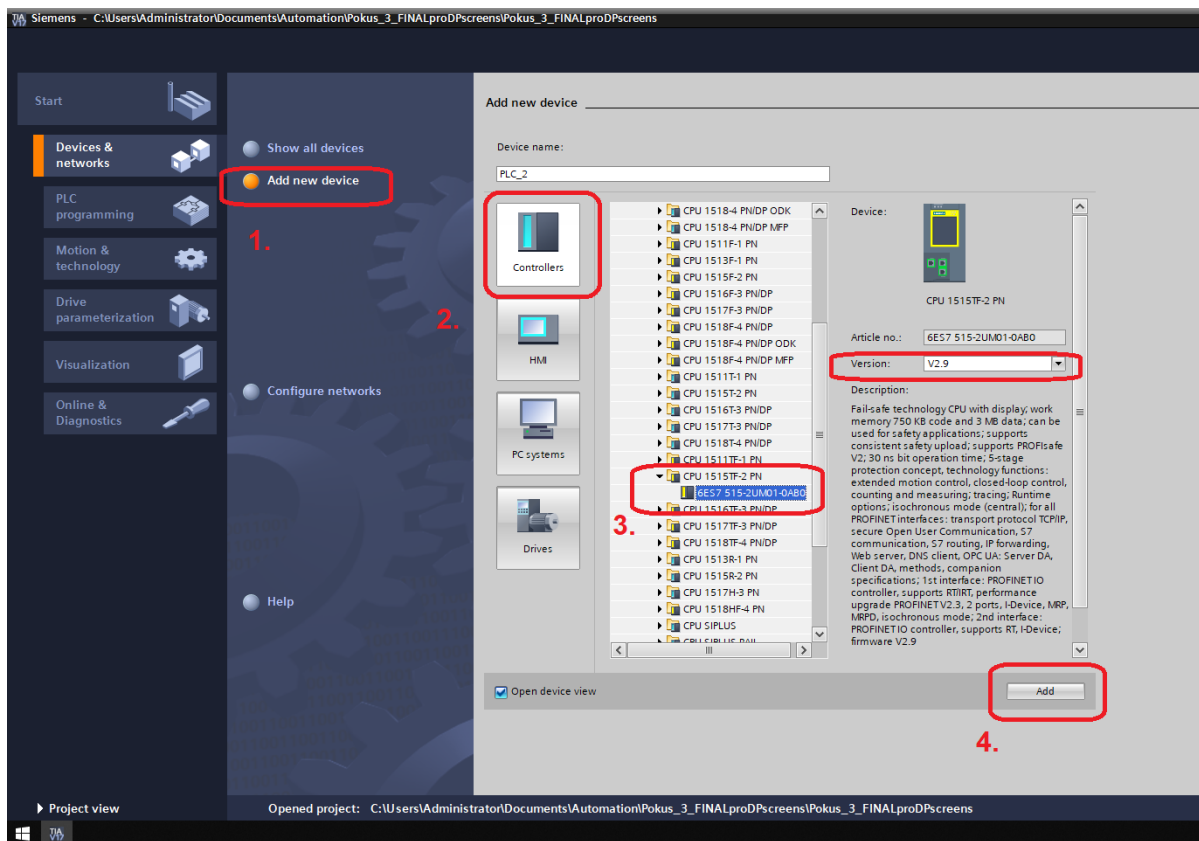
Veškerá práce ve vývojovém prostředí TIA Portal začíná v úvodním okně, tzv. *Portal view* okně. Úvodní okno slouží kromě jiného hlavně ke správě již vytvořených projektů a založení nového. Pro založení nového projektu je nutné kliknout na *Create new project*, kliknutí nás přeneso do nového okna, zde projekt pojmenujeme, zkontrolujeme požadovanou adresu projektu a vytvoříme projekt kliknutím na *Create*.



Obr. 6.1 Vytvoření projektu TIA Portal

S již založeným projektem se nyní nacházíme v záložce *First steps*. Tato záložka znázorňuje nástrojové bloky, mezi které patří konfigurace sestavy zařízení, návrh PLC programu, konfigurace technologických objektů, parametrizace pohonů a naprogramování vizualizace. Celkem se jedná o pět důležitých oblastí každého projektu. Bloky mají za úkol komplexně postihnout všechny části projektu od konfigurace zařízení po jeho vizualizaci. Oblasti jsou graficky znázorněny v hlavní pravé části *Portal view* okna. Bloky jsou interaktivní, stačí tedy kliknout na zamýšlený krok.

Jak naznačuje grafika záložky *First steps*, prvním krokem projektu je konfigurace použitých zařízení. Klikneme tedy na ikonu *Configure a device*, která nás přesune do bloku *Devices&Network*. Nyní provedeme konfiguraci kompletní sestavy zařízení z kapitoly 4. Aby konfigurace zařízení byla úspěšná, musíme vždy vybrat správné zařízení včetně správné verze firmwaru. Takto vložíme do projektu PLC zařízení, dva servopohony a dva HMI panely.

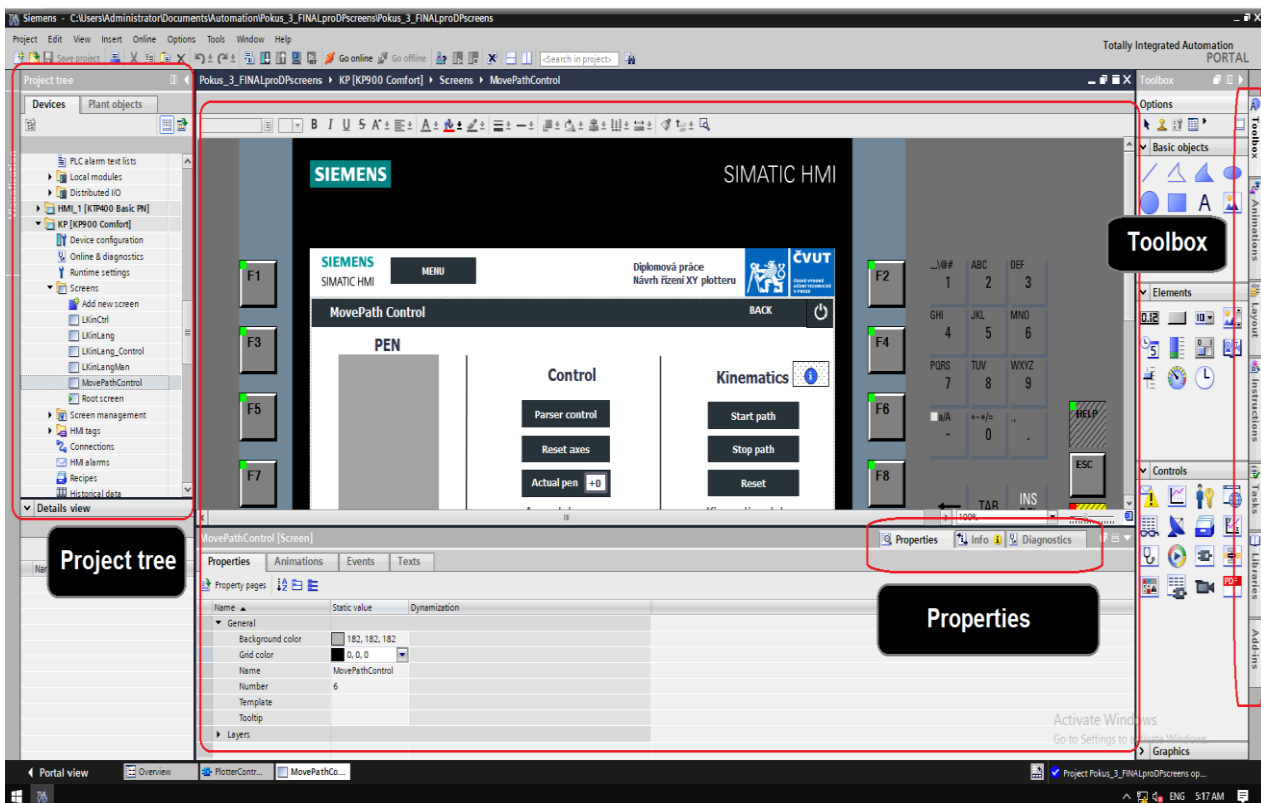


Obr. 6.2 Konfigurace zařízení TIA Portal

6.2 Project view

Další postup provedeme již v tzv. *Project view* okně. *Project view* okno představuje hlavní prostředí obsahující veškeré dostupné nástroje a možnosti nastavení. Přepnutí lze provést kliknutím na ikonu *Project view* v levém spodním rohu okna programu TIA Portal viz obr. 6.1.

Pracovní okno *Project view* je sestaveno z několika funkčních oken. V pravé části najdeme tzv. *Project tree* sloužící k orientaci a uspořádání projektu. V prostřední části se nachází hlavní pracovní okno právě maximalizované aktivity. Pracovní okno je rozděleno na dvě části. Spodní část ukazuje vlastnosti otevřené aktivity společně se záložkami informací a diagnostických hlášek. Tyto záložky využijeme především při online přístupu k zařízením. Již otevřená pracovní okna můžeme otevřít zpět na dolní liště. Pravá část funguje jako nabídka konkrétních nástrojů a instrukcí, které můžeme při daném návrhu použít.

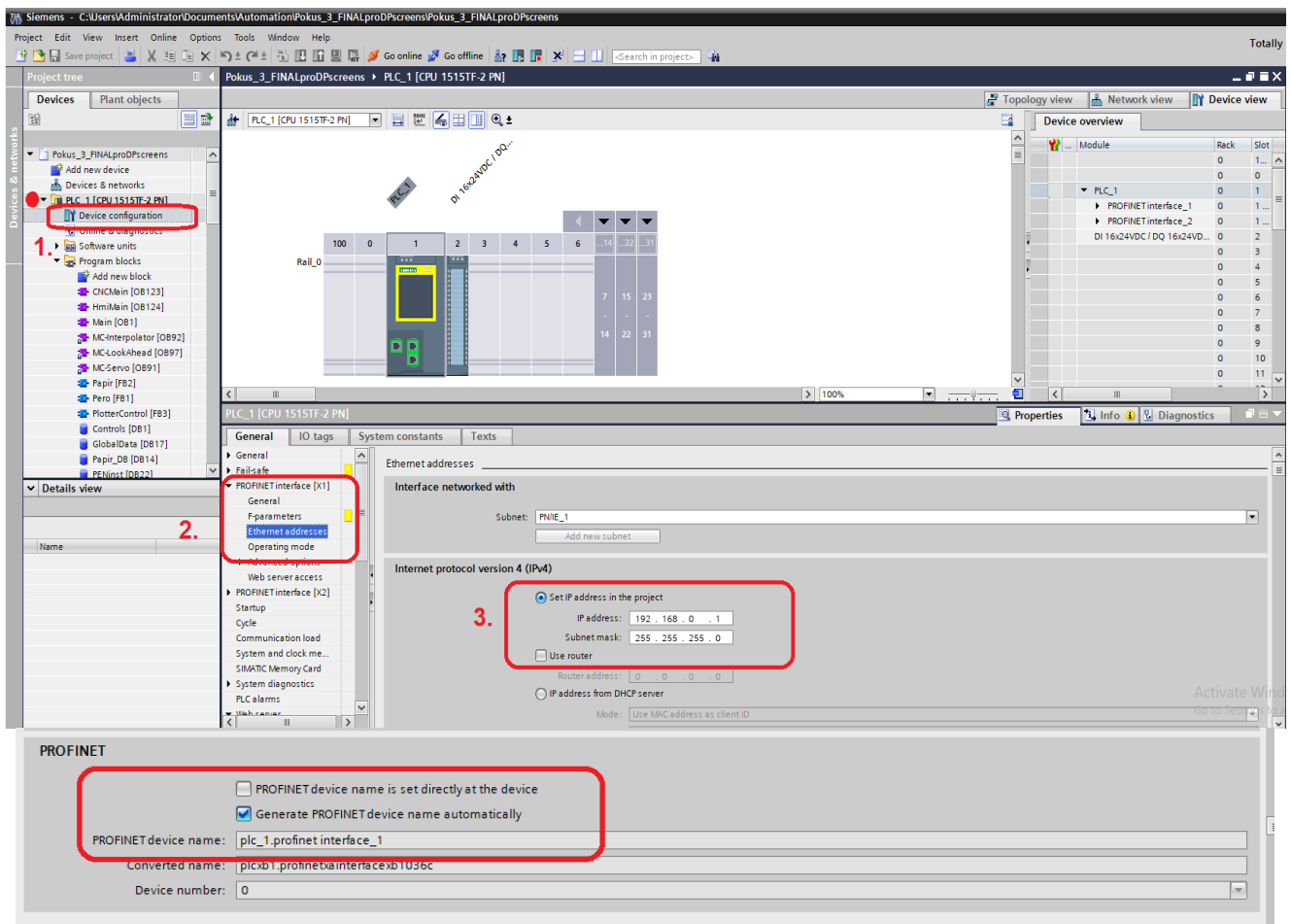


Obr. 6.3 Struktura Project view TIA Portal

6.3 Nastavení komunikace zařízení s projektem

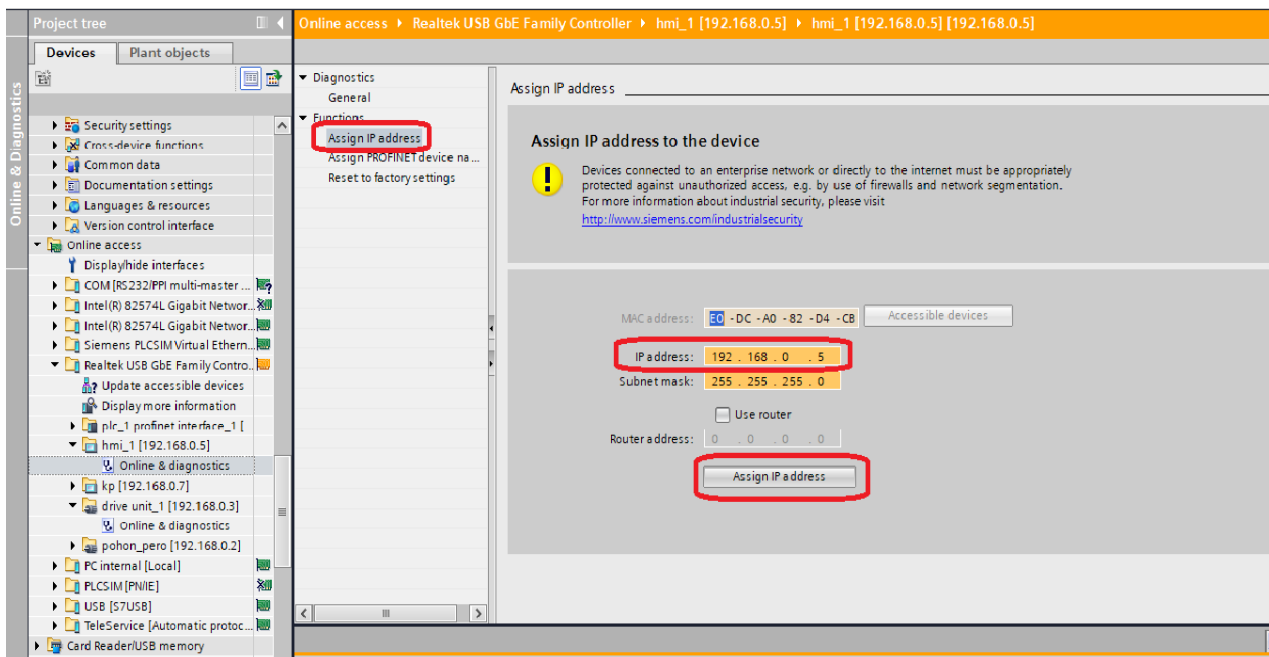
Nastavení konfigurace zařízení ve vývojovém prostředí TIA Portal není úplně bez nastavení ethernetového rozhraní, resp. konfigurace komunikace mezi zařízeními v prostředí TIA Portal. Následující kroky je možné rozdělit do dvou částí, konfigurace projektového zařízení a nastavení skutečného zařízení, které je k počítači s aplikací TIA Portal připojeno.

V případě první části otevřeme v levém okně *Project tree* dané zařízení a u jeho záložky pomocí kliknutí nabídku otevřeme a vybereme *Device configuration*. V hlavním okně se musíme nacházet v záložce *Properties*, přejdeme do nastavení *PROFINET interface [X1]* -> *Ethernet addresses* a zvolíme si IP adresu zařízení. IP adresa může zůstat přednastavená, běžně ponecháváme PLC zařízení nejnižší index (192.168.0.1). Následně zkontrolujeme vygenerované jméno zařízení v rámci PROFINET komunikace a přejdeme k druhé části.



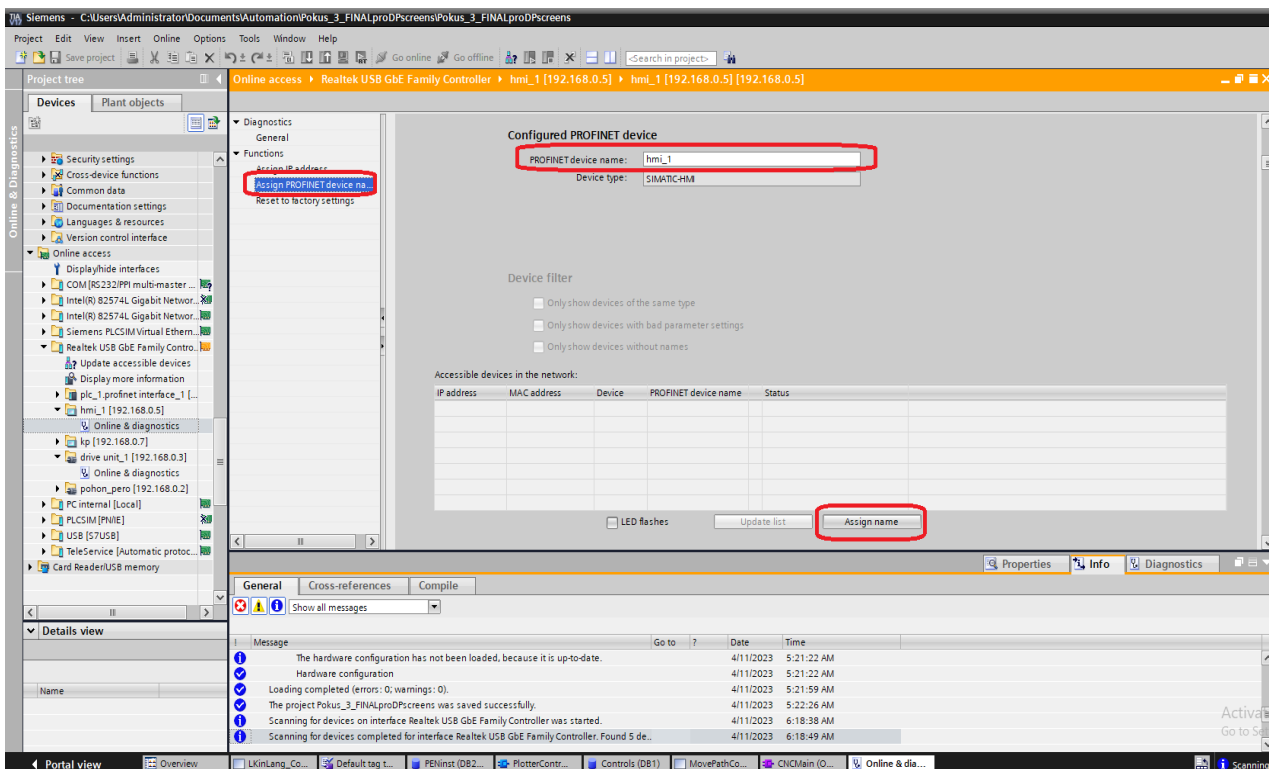
Obr. 6.4 Konfigurace komunikace zařízení

Druhá část spočívá v replikaci nastavení pro skutečné zařízení v sestavě. V prvním kroku je potřeba zařízení najít a spojit se s ním v rámci online přístupu. V levém okně *Project tree* přejdeme na záložku *Online access*, vybereme správné rozhraní, přes které jsme připojeni a obnovíme seznam dostupných zařízení. Po zobrazení připojených zařízení vybereme požadované zařízení. V tomto případě se jedná o PLC zařízení, ale v případě problémů je třeba provést konfiguraci komunikace u všech zařízení v projektu.



Obr. 6.5 Přiřazení IP adresy zařízení v sestavě

K zařízení se připojíme skrze blok *Online & diagnostics*. Jakmile je online přístup úspěšně proveden, v hlavním okně přejdeme do záložky *Functions* a klikneme na *Assign IP address*. Zde je velice důležité nastavit v případě rozkolu stejnou IP adresu jako máme u zařízení v projektu a tuto adresu pomocí příkazu *Assign IP address* zařízení přiřadit. Následně přejdeme do nabídky *Assign PROFINET device name* (nabídkové okno v levé části hlavního okna) a ujistíme se, že jména PROFINET komunikace opět odpovídají zařízení v projektu. Pokud ne, v projektu jej u zařízení změníme na zde uvedené jméno. Předjdeme tak budoucím problémům během komunikace se zařízeními.

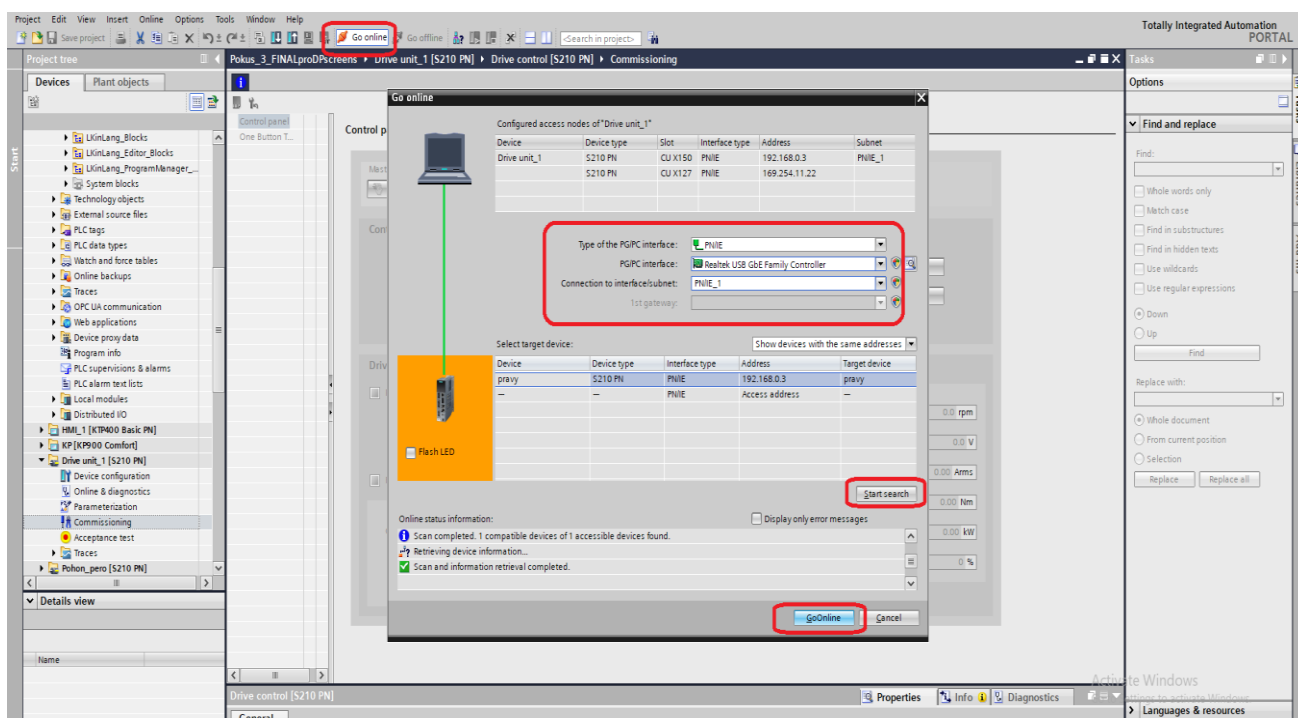


Obr. 6.6 Přiřazení PROFINET jména zařízení v sestavě

6.4 Naladění regulace dvousóého servopohonu

Před implementací řídicích technologických objektů je vhodné naladit regulační blok obou os pohonu. Přestože zde máme synchronní motory s permanentními magnety, které dosahují širokého regulačního rozsahu a vysoké dynamiky řízení, ideální odezvy se nám dostane při naladění regulace pro konkrétní aplikaci. Zde se jedná o pohyb pera v přípravku.

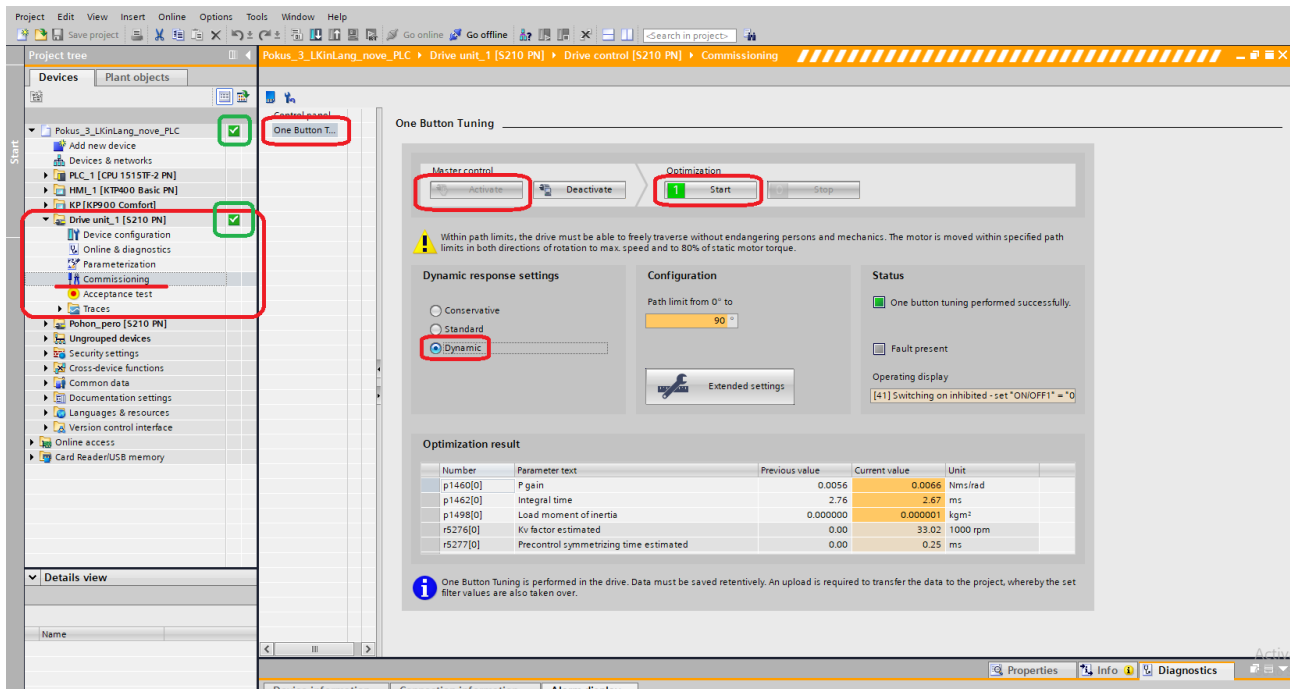
Vývojové prostředí TIA Portal pro pohony od spol. Siemens umožňuje automatické naladění pomocí funkce *One Button Tuning*. Jak je již patrné, program sám dle požadované odezvy naladí regulační schéma, resp. regulátory pro daný servopohon. Naladění musíme v projektu udělat pro oba pohony, tedy jak pohon pera, tak i pohon papíru.



Obr. 6.7 Provedení příkazu *Go online*

Před naladěním je třeba přejít do *Online access* režimu. K tomu stiskneme tlačítko *Go online* v horní liště zkratk, zvýrazněno na obr. 6.7. Pokud se k sestavě připojujeme poprvé, objeví se v prostředí nové okno *Go online*, zde stačí vybrat odpovídající zařízení a potvrdit volbu. Vyhledávání se konfiguruje stejně jako v případě nahrávání programu do zařízení, více na obr. 6.7.

Automatické naladění provedeme v záložce daného pohonu v sekci *Commissioning*. V nabídce vybereme druhou možnost *One Button Tuning*. Postup je zobrazen na obr. 6.8. Posloupnost kroků je číselně znázorněna. V případě naší aplikace je vhodné vybrat dynamickou odezvu ve výběru *Dynamic response settings*. Zeleně jsou označeny záložky obou pohonů v projektu, které musíme při návrhu naladit.



Obr. 6.8 Funkce One Button Tuning

Funkce automatického ladění *One Button Tuning* se provede automaticky spuštěním tlačítka *Activate* a následně *Start*. Postup je nutné opakovat pro druhý pohon.

KAPITOLA 7: POPIS POUŽITÝCH NÁSTROJŮ PŘI VÝVOJI

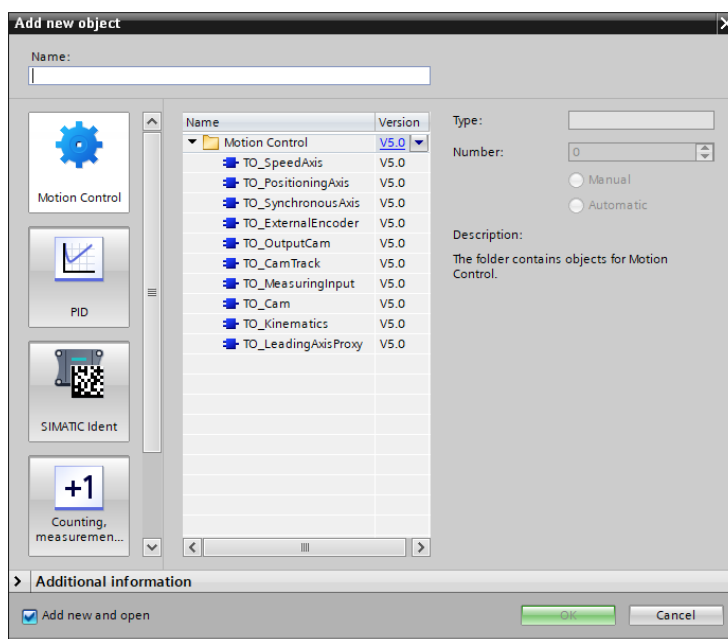
V rámci návrhu a realizace vnitřní logiky práce XY ploteru jsou často využívány vývojové nástroje specifické pouze pro PLC návrh ve vývojovém prostředí TIA Portal. Jedná se především o určité funkční bloky, které jsou využívány na úrovni technologických objektů, nebo speciální aplikace HMI panelů. Vývojové prostředí spol. Siemens disponuje řadou vlastních nástrojů, které mají široké pole působnosti. Do jisté míry jsou však tyto nástroje vytvořeny pro určité aplikace, zákazníci často požadované, s cílem usnadnit jejich návrh. Pro zachování funkčnosti není možné tyto nástroje individualizovat, což přináší výzvu při implementaci nástrojů do návrhu, který je odlišný od původního konceptu. Je proto zásadní tyto nástroje co možná nejlépe pochopit.

7.1 Technologický objekt (TO)

Technologický objekt představuje softwarovou reprezentaci hardwarové komponenty, kterou chceme řídit či ovlivňovat. V našem případě se jedná o řízení obou os elektrického pohonu. Uplatnění technologického objektu najdeme nejenom u řízení pohonů, ale také u řízení dalších technologických zařízení či procesů. Technologickým objektem můžeme řídit teplotu ohřívání výrobku v peci nebo dohlížet na pásovou výrobu. V této práci se však zaměříme pouze na oblast řízení elektrických pohonů. Jinde technologický objekt využít při návrhu nebude [32].

Technologický objekt nahrazuje skutečné zařízení v projektu. Umožňuje tím zjednodušení manipulace a nakládání s danou osou pohonu. Všechny náležitosti řízení například rychlostní osy jsou v technologickém objektu zabaleny do uceleného objektu s přehledným nastavením a konfigurací. To znamená, že všechny náležitosti optimalizace, konfigurace a řízení osy obstarává v projektu nakonfigurovaný technologický objekt. Výhoda tohoto uspořádání spočívá především ve snadné implementaci složitějších aplikací. V případě návrhu řízení dvouosého pohonu pohybu pera nám tak stačí správná konfigurace tří technologických objektů a návrh řízení s již připravenými funkčními bloky *Motion Control* [32], [19].

Existuje několik základních typů technologických objektů pro řízení pohonů. Dostupnost jednotlivých typů *TO* je závislá na typu použitého PLC zařízení v projektu. Na obr. 7.1 je zobrazena nabídka *TO Motion Control* pro námi použité zařízení. V základních aplikacích se setkáme nejčastěji s *TO Speed Axis*, *TO Positioning Axis*, *TO Synchronous Axis* nebo námi využívaného *TO Kinematics*.

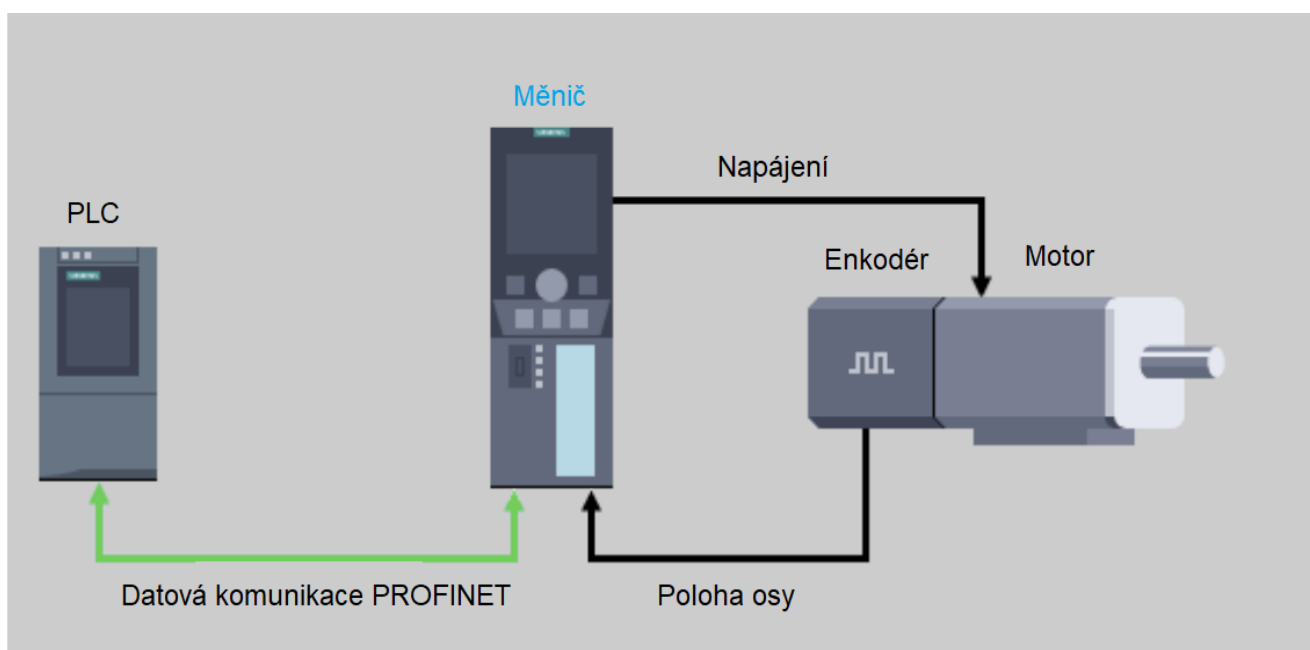


Obr. 7.1 Nabídka TO v prostředí TIA Portal

7.1.1 TO Positioning Axis

Pokud chceme využít absolutní polohování při pohybu osy pohonu, je nutné zvolit *TO Positioning Axis*. Pokud mluvíme o jedné řízené ose, připadají nám v úvahu pouze tři *TO*, a to *TO Speed Axis*, *TO Positioning Axis* nebo *TO Synchronous Axis*. *TO Speed Axis* je objektem úplně nezákladnějším. Pohyb osy je postaven na řízení rychlosti. Rychlostní osa umožňuje tedy pouze pohyb relativní, kde probíhá posun o zadanou relativní vzdálenost [10].

V případě požadavku na absolutní polohování osy, tedy pohyb do zadané polohy, využijeme polohovací osy. *TO Positioning Axis* již umožňuje absolutní pohyb. S absolutním pohybem však souvisí tzv. *homming*, neboli provedení referování polohy pohonu. K tomu se využívá připravený funkční blok funkce *MC_Home* ze standardní knihovny *Motion Control* vývojového prostředí TIA Portal. *Homming* zajišťuje určení referenční polohy enkodéru. Servomotory SIMOTICS 1-FK2 disponují absolutním enkodérem, a proto je možný pouze tzv. pasivní *homming*, tedy manuální pohyb osy do požadované referenční polohy.



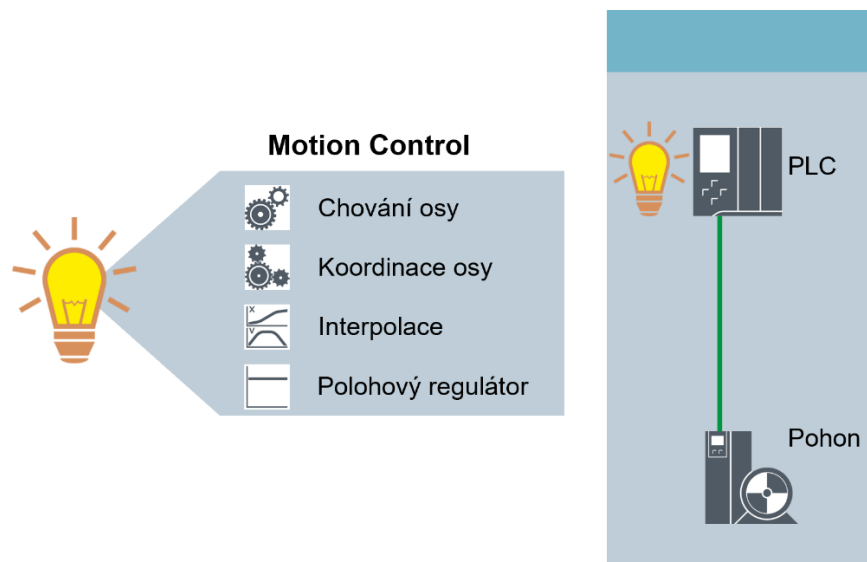
Obr. 7.2 TO Polohovací osy (upraveno) [32]

Provedení správného referování je klíčový faktor pro automatické vykreslování, které je řízeno pomyslným souřadnicovým systémem, a proto se tomuto kroku budeme věnovat v další kapitole návrhu vnitřní logiky.

7.1.2 Motion Control

Motion Control, zkráceně MC, je sada funkčních bloků připravená k využití technologických objektů řízení pohonů. Sada je standardně dostupná ve vývojovém prostředí TIA Portal v sekci *Instructions* v záložce *Technology*.

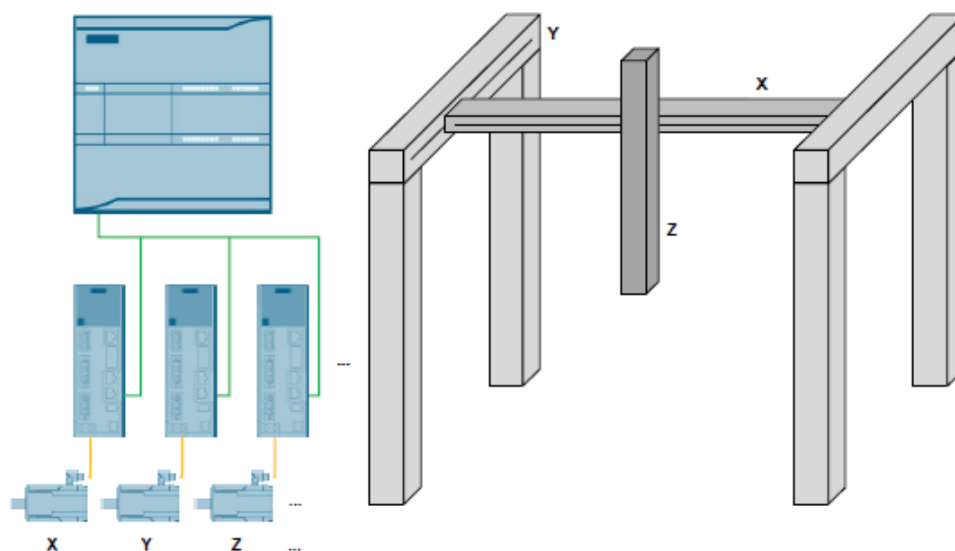
Mezi základními bloky Motion Control figurují funkce, které pokrývají základní příkazy k řízení osy pohonu. Bloky mají na starosti chování osy, koordinaci osy, její interpolaci a polohový regulátor. To znamená, že mimo jiné řídí akceleraci, decelaci, moment servomotoru, rychlost a s tím i následně polohu osy [33].



Obr. 7.3 Motion Control (upraveno) [19]

7.2 TO Kinematics

Kinematický technologický objekt představuje nadřazený objekt sdružující již existující technologické objekty polohovacích os. Jeho úkolem je výpočet parametrů a požadovaných hodnot nutných pro pohyb víceosého pohonu. Technologický objekt uvažuje vzájemnou dynamiku jednotlivých os v rámci pohybu a řídí jejich spolupráci na pohybu. Jedním z dalších úkolů je transformace absolutního polohování v rámci každé osy. Jednoduše řečeno, technologický objekt kinematiky má za úkol zajistit hladký pohyb technologie v případě, že využíváme víceosého pohonu.



Obr. 7.4 Princip TO Kinematiky (upraveno) [34]

Víceosý pohon je případ řízení XY ploteru. Myšlenkou řízení přípravku je vytvoření pomyslného souřadnicového systému na daném povrchu papíru a pohyb pera v obou osách, X a Y. V případě pohybu pera zahrnující pohyb po obou osách, je nutné vzájemně zkoordinovat servopohony os X a Y. Pokud chceme po technologickém objektu přímý pohyb do absolutní polohy, musí se všechny parametry pohonů po celou dobu pohybu řídit s ohledem na druhou osu. Jedině tímto způsobem dosáhneme synchronizované pohybu. Bez vzájemné koordinace bychom

museli požadovaný pohyb složit z po sobě jdoucích pohybů každé osy zvlášť, což je pro aplikaci vykreslování zcela nežádoucí.

7.3 Externí knihovny kinematiky

Víceosý pohon vyžaduje speciální přístup využívající místo *TO Speed Axis* či *TO Positioning Axis* technologický objekt Kinematiky (*TO Kinematics*) sdružující osy do jediného objektu. Pro využití plného potenciálu technologického objektu Kinematiky je třeba naimportovat externí knihovny poskytující rozšiřující funkce pro daný technologický objekt do vývojového prostředí TIA Portal.

Použití externích knihoven má určitá specifika, která jsou vysvětlena dále v textu.

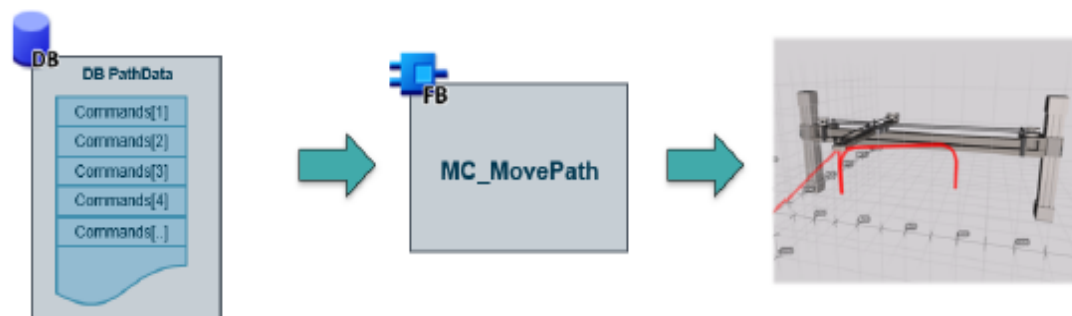
7.3.1 Kinematics Control

Externí knihovna *Kinematics Control* rozšiřuje funkčnost *TO Kinematics*. Funkční bloky knihovny umožňují pohyb po předdefinované trajektorii s využitím speciálního datového bloku *pathData*. Datový blok obsahuje speciální formátování, se kterým pracuje blok *LKinCtrl_MC_MovePath*, který obstarává dílčí kroky pohybu technologického objektu [35].

Hlavní přínos knihovny *Kinematics Control* spočívá v částečné transformaci PLC zařízení s nakonfigurovaným *TO Kinematika* v CNC stroj schopný pohybu po zadané trajektorii v nastaveném souřadnicovém systému. Soubor instrukcí pro pohyb je možné editovat pomocí HMI panelu s běžící HMI aplikací *LKinCtrl_Editor*.

Knihovna ovšem neumožňuje zpracování běžně používaného G-code souboru pro potřeby automatického vykreslování. Návrh ve vývojovém prostředí TIA Portal bohužel respektuje jenom vlastní strukturu typu *pathData*. Soubor instrukcí popisující trajektorii pohybu je tedy nutné pro vykonání zpracovat a transformovat ve správně naformátovaný data blok.

Oficiální detailní dokumentace týkající se knihovny je volně k nahlédnutí zde [35].



Obr. 7.5 Použití knihovny *Kinematics Control*

7.3.2 Kinematics Language

Externí knihovna *Kinematics Language* dále rozšiřuje funkcionalitu *TO Kinematika* nad rámec předešlé knihovny *Kinematics Control*. Umožňuje zpracování textových příkazů v programovacím jazyku G-code typickém pro technologii CAM. Kromě toho je možné si definovat vlastní příkazový programovací jazyk. Při správné konfiguraci funkční blok *LKinLang_Parser* dále dokáže transformovat uživatelsky definovaný jazyk instrukcí v datový blok připravený pro blok *LKinCtrl_MovePath* starající se o pohyb objektu pro zadané trajektorie [31].

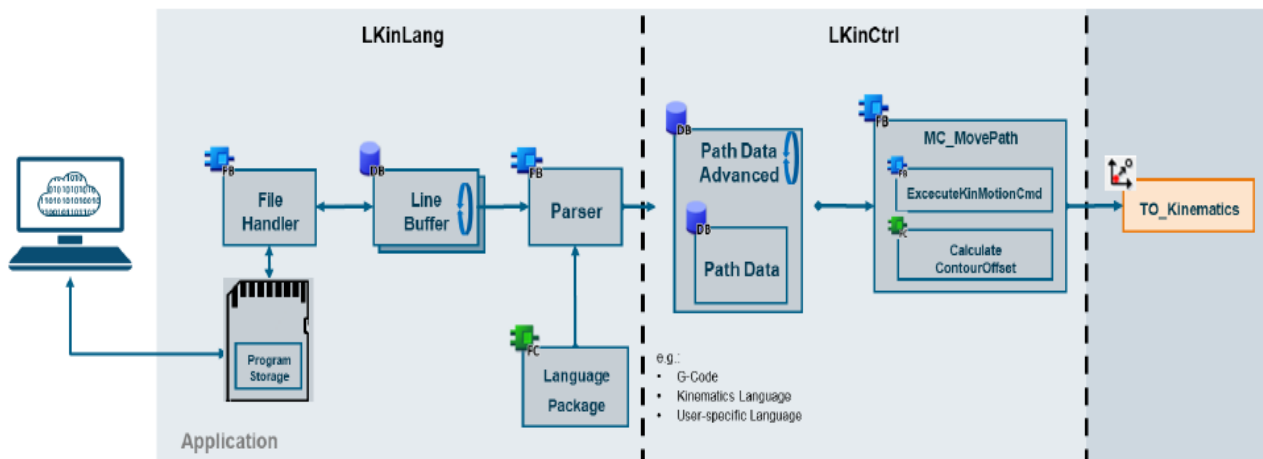
Všechny tyto kroky dokáže automatizační sestava zařízení podniknout bez nutnosti trvale připojeného projektu ve vývojovém prostředí TIA Portal a zásahu vývojáře.

Na překladu příkazových souborů typu G-code se podílí speciální HMI aplikace *LKinLang_ProgramManager*. Její využití velice připomíná počítačový adresář s možností správy souborů na paměťové kartě PLC zařízení. Kromě toho však tato aplikace dokáže vybraný soubor přemístit z externího úložiště na paměťovou kartu PLC zařízení a z paměťové karty soubor nahrát do zásobníku v rámci operační paměti zařízení. Obrovskou výhodou aplikace je možnost provádět

všechny tyto operace během chodu PLC zařízení. Není tak narušen chod programu a zařízení není nutné zastavit, resp. uvést do stavu *STOP*, během přehrávání dat [31].

V poslední řadě balíček nástrojů obsahuje společně s projektovými bloky také speciální počítačovou aplikaci File Transfer tool umožňující snadnou výměnu datových souborů mezi počítačem se spuštěnou aplikací a PLC zařízením. Komunikace využívá funkce *Web server* bez nutnosti manuálního připojování. To znamená, že stačí, aby obě zařízení, PC i PLC zařízení, byla připojena ke společné lokální ethernetové síti. Více o aplikaci najdete v podkapitole 4.2.2.

Oficiální dokumentace je volně k nahlédnutí zde [31].



Obr. 7.6 Použití knihovny *Kinematics Language* [31]

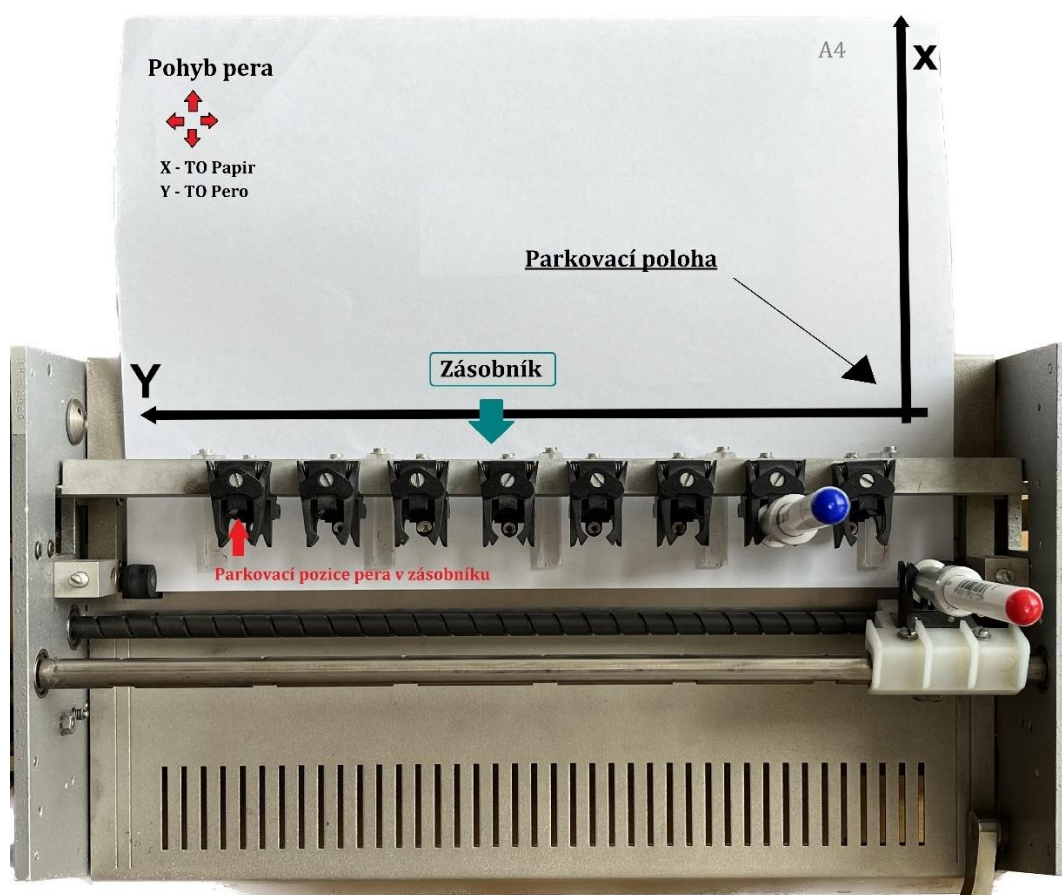
KAPITOLA 8: NÁVRH ŘÍZENÍ DVOUOSÉHO POHONU PLOTERU

Kapitola 8 pojednává o dílčích krocích návrhu řízení dvouosého pohonu. Budou stanoveny požadavky a popsán postup návrhu řízení pohonu. Postup zahrnuje vytvoření a konfiguraci technologických objektů pro sestavu servopohonu a přípravku, implementaci logického řízení obou os pohonu s využitím funkčních bloků technologického objektu Polohovací osy a realizaci souboru příkazů k tzv. *hommingu* obou os. Zavedení synchronizace je nutné pro pozdější využití technologického objektu Kinematiky a souřadnicového pohybu pera po papíru.

8.1 Analýza požadavků na návrh řízení přípravku

V případě, že předpokládáme již připravený projekt s nakonfigurovanou funkční sestavou zařízení, provedeme analýzu požadavků na řízení přípravku, pro které následně navrheme řešení.

V rámci obsluhy přípravku je nutné vytvořit základní ovládání přípravku v podobě zapnutí a vypnutí. Je patrné, že provedení úspěšného zapnutí přípravku musí být podmíněno výběrem formátu a založením papíru do zařízení. Pero musí být možné manuálně ovládat nezávisle na funkci automatického vykreslování výkresů. K ovládání pohybu pera musíme doplnit ovládání přiložení pera, resp. řízení, kdy pero píše a kdy se pohybuje nad papírem. Dále je nutné zpracovat logiku výměny pera ze zásobníku. To znamená realizovat logické řízení přidruženého pohonu zásobníku dle přiloženého postupu a k tomu vytvořit vnitřní logiku, která na základě kontroly aktuálně používaného pera vrátí pero do zásobníku a automaticky vybere ze zásobníku nově požadované.



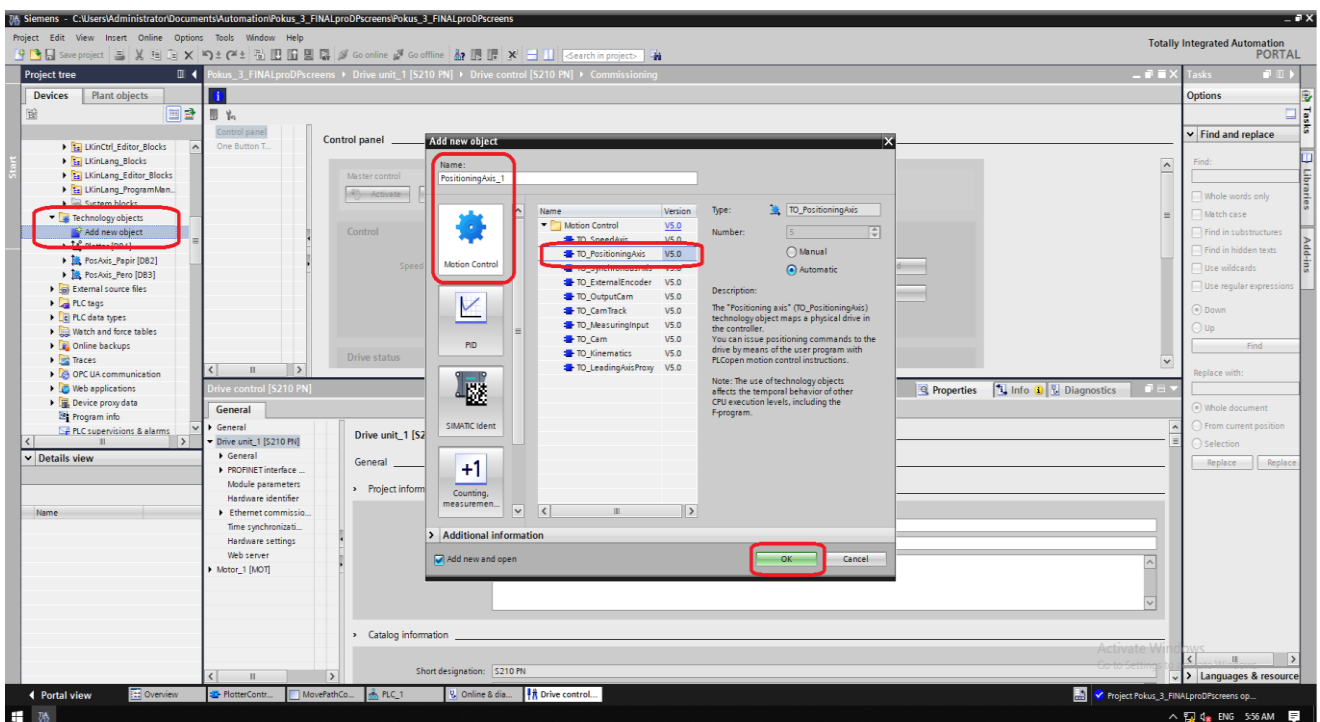
Obr. 8.1 Požadavky návrhu

8.2 Konfigurace TO Pero a Papír

Požadavky na návrh ovládání přípravku budou stanoveny v podkapitole 9.1. Z nich vyplývá, že je nutné zkonstruovat strukturu, přes níž bude možné manuálně řídit pohyb pera. K řízení osy pohonu máme k dispozici ve vývojovém prostředí TIA Portal soubor funkčních bloků *Motion Control*. Jejich význam a použití bylo vysvětleno v kapitole 7.

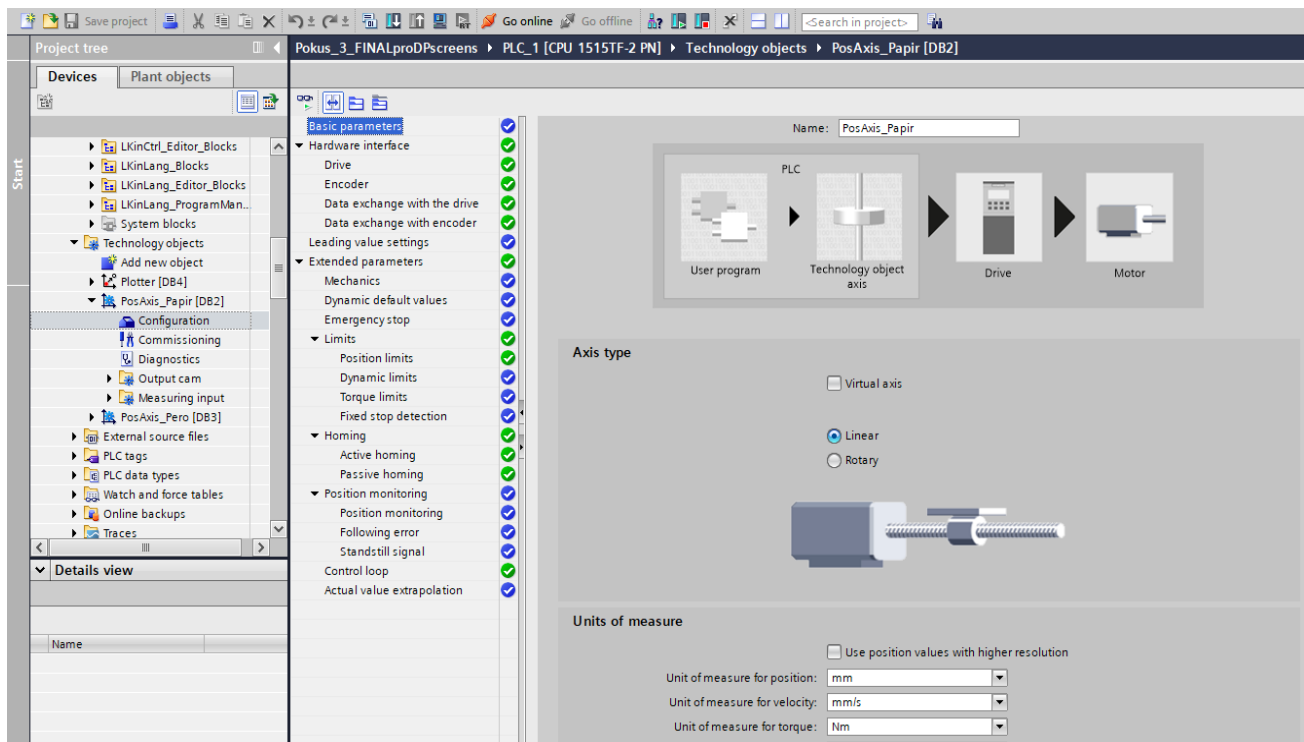
K použití bloků *Motion Control* musíme nejdříve vytvořit a nakonfigurovat odpovídající technologické objekty jednotlivých os. Technologické objekty pro nás budou reprezentovat skutečné osy pohybu přípravku. Od pohybu vyžadujeme absolutní polohování pro účely souřadnicového řízení pohybu, založíme tedy dva *TO Positioning Axis* pro osu pohybu pera a pro kolmou osu pohybu papíru.

Správu technologických objektů najdeme v projektu u příslušného řídicího PLC zařízení. V záložce *Technology objects* zvolíme *Add new object* a po zobrazení nabídky se ujistíme, že jsme v nabídce *Motion Control* a zvolíme *TO_PositioningAxis*. Jedná se o druhou volbu odshora. Volbu potvrdíme dole tlačítkem *Ok*.



Obr. 8.2 Založení TO Polohovací osy

Potvrzení nás automaticky přeneso do konfiguračního okna technologického objektu. V projektu konfiguraci najdeme v záložce daného *TO* pod názvem *Configuration*. Zde je zobrazeno přehledně veškeré nastavení, které TIA Portal pro odpovídající technologický objekt poskytuje.



Obr. 8.3 Konfigurace TO – Basic parametres

Konfiguraci naší osy začneme v sekci *Basic parametres* nastavením jména. Z projektu vidíme, že osa byla pojmenována *PosAxis_Papir*. Pokračujeme typem osy, zde ponecháme nastavení *Linear*, což koresponduje s přípravkem.

Pokračujeme do další sekce *Hardware interface*. Zde je třeba spojit softwarovou část *TO* s její hardwarovou reprezentací. Nastavení je vcelku snadné, neboť vývojové prostředí TIA Portal sdružuje sestavu měniče a motoru v jedno pod názvem *Drive_unit*. Zároveň servomotory SIMOTICS 1-FK2 disponují zabudovaným absolutním enkodérem. Veškeré nastavení je pouze záležitostí správného výběru ze zařízení projektu. Vzhledem k tomu, že v projektu máme identické servopohony, je snadné jednotky *Drive_unit* bez řádného popisu při vkládání do projektu zaměnit. Je proto dobré se před výběrem k měniči připojit skrze *Online access* a provést *Commissioning*, abychom se ujistili o správném výběru měniče pro osu papíru či pera. V případě pohonu a enkodéru vybereme správný pohon. Pro komunikaci s osou pohonu a enkodérem ponecháme *telegram 105*, standardní pro měnič SINAMICS S210. Výběr pro potřeby našeho projektu je k dispozici na obr. 8.4 níže.

The image displays three panels of the SIMATIC Manager configuration interface, all enclosed in a red border. The leftmost panel is a navigation tree with the following items and status icons (green checkmarks for active/checked, blue checkmarks for inactive/unchecked):

- Basic parameters: [checked]
- Hardware interface: [checked]
 - Drive: [checked]
 - Encoder: [checked]
 - Data exchange with the drive: [checked]
 - Data exchange with encoder: [checked]
- Leading value settings: [checked]
- Extended parameters: [checked]
 - Mechanics: [checked]
 - Dynamic default values: [checked]
 - Emergency stop: [checked]
 - Limits: [checked]
 - Position limits: [checked]
 - Dynamic limits: [checked]
 - Torque limits: [checked]
 - Fixed stop detection: [checked]
 - Homing: [checked]
 - Active homing: [checked]
 - Passive homing: [checked]
 - Position monitoring: [checked]
 - Position monitoring: [checked]
 - Following error: [checked]
 - Standstill signal: [checked]
 - Control loop: [checked]
 - Actual value extrapolation: [checked]

The top panel, titled "Drive", shows a schematic diagram of a PLC connected to a Drive, which is connected to an Encoder and a Motor. The Drive is labeled "Drive" and the Motor is labeled "Motor". The Encoder is labeled "Encoder". The diagram shows "Data exchange" between the PLC and Drive, and "Encoder data" between the Drive and Encoder. Below the diagram are configuration fields:

- Drive type: PROFIdrive
- Data connection: Drive
- Drive: Drive unit_1.Drive_control_SIE ...

The middle panel, titled "Encoder 1", shows configuration for the first encoder:

- Use encoder:
- Data connection: Encoder
- Encoder: Drive unit_1.Drive_control_SIE ...
- Encoder type: Cyclic absolute

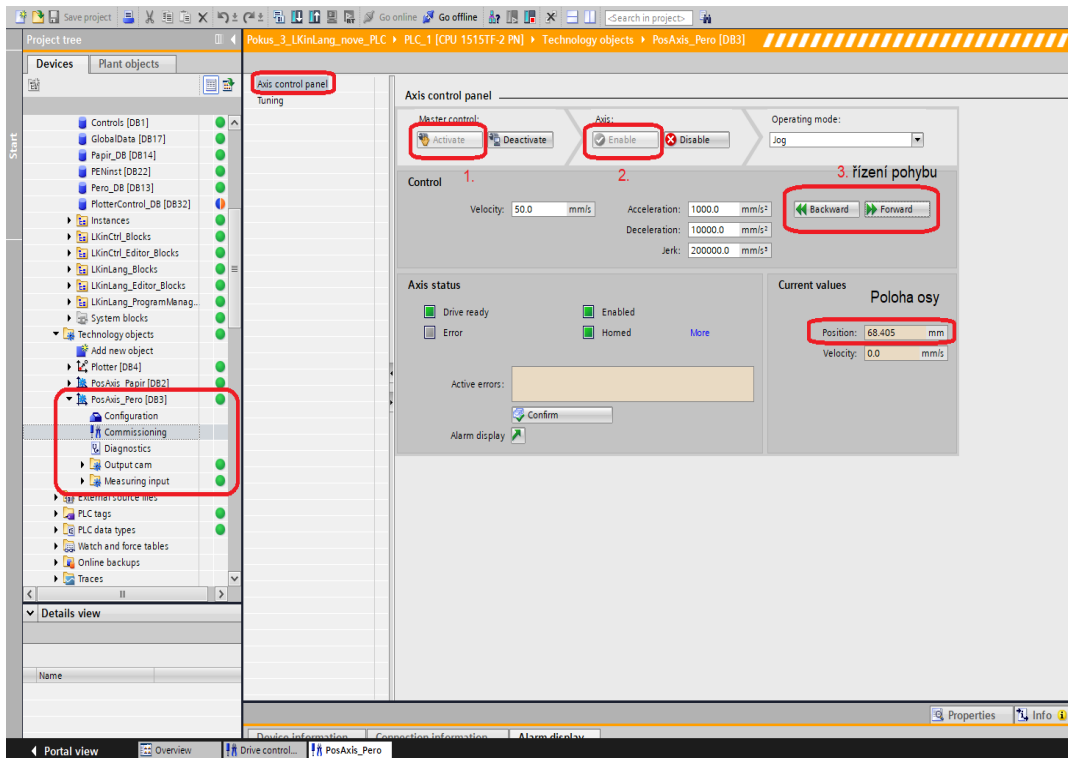
The bottom panel, titled "Encoder 2", shows configuration for the second encoder:

- Use encoder:
- Data connection: Encoder
- Encoder: <Select encoder>

The bottom panel also shows "Drive data" configuration:

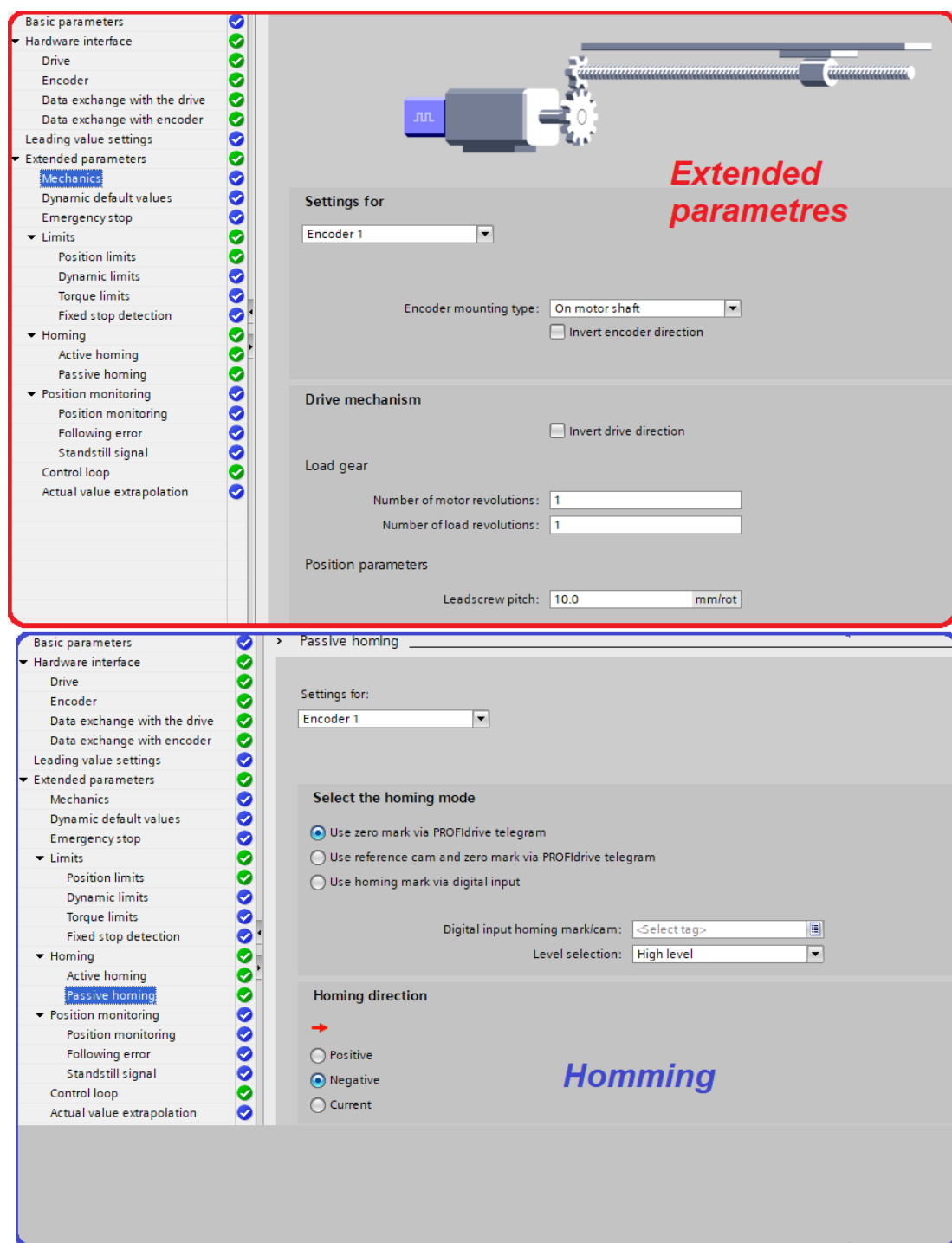
- Drive telegram: Standard telegram 105
- Automatically apply drive values during configuration (offline):
- Automatically apply drive values at runtime (online):
- Reference speed: 8000.0 1/min
- Maximum speed: 8000.0 1/min
- Reference torque: 1.23605 Nm

Obr. 8.4 Konfigurace TO – Hardware interface



Obr. 8.5 Pohon osy pohybu papíru - Commssioning funkce

Dále nastavíme záložku *Extended parameters*. V sekci *Mechanics* vybereme typ uložení enkodéru na hřídeli, tedy *On motor shaft* a dole vyplníme *Leadscrewpitch* hodnotou 10.0 mm/rot. Jedná se o referenční hodnotu převodu rotačního pohybu hřídele motoru na translační pohyb osy přípravku. Bez referenční hodnoty bychom nemohli správně číst pohyb pera po papíru. Hodnota je součástí příložených informací k přípravku. Další sekce necháme v základním nastavení.



Obr. 8.6 Konfigurace TO – *Extended parameters a Homming*

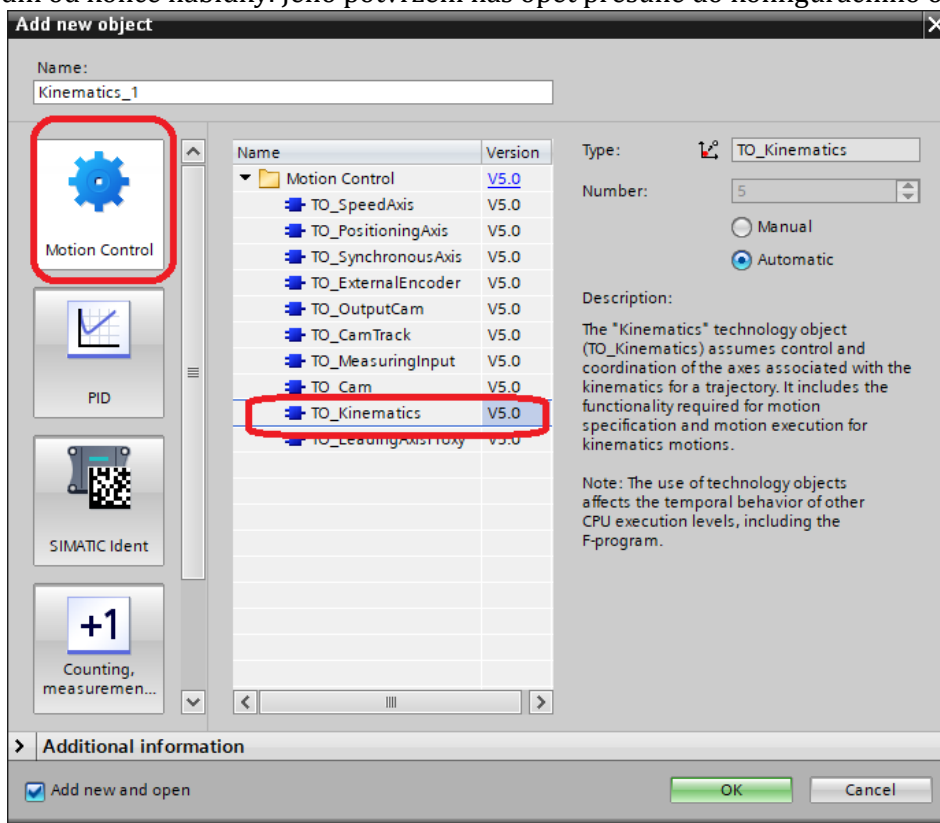
Pokračujeme záložkou *Homming*. Zde se v obou sekcích ujistíme, že tzv. *Homming direction*, tedy směr referování, je záporný. Záporný směr pohybu referování vyplývá z určení referenční polohy, resp. určení počátku souřadné systému XY pro pohyb pera. Při založení papíru do ploteru je proveden řízený pohyb referování záporným směrem po obou pomyslných osách X a Y, které korespondují s osami pohonu.

Stejnou konfiguraci s rozdílem výběru druhého pohonu provedeme i pro druhý TO reprezentující osu pohybu pera. Z obr. 8.5 vidíme, že druhý TO byl pojmenován PosAxis_Pero.

8.3 Konfigurace TO Plotter

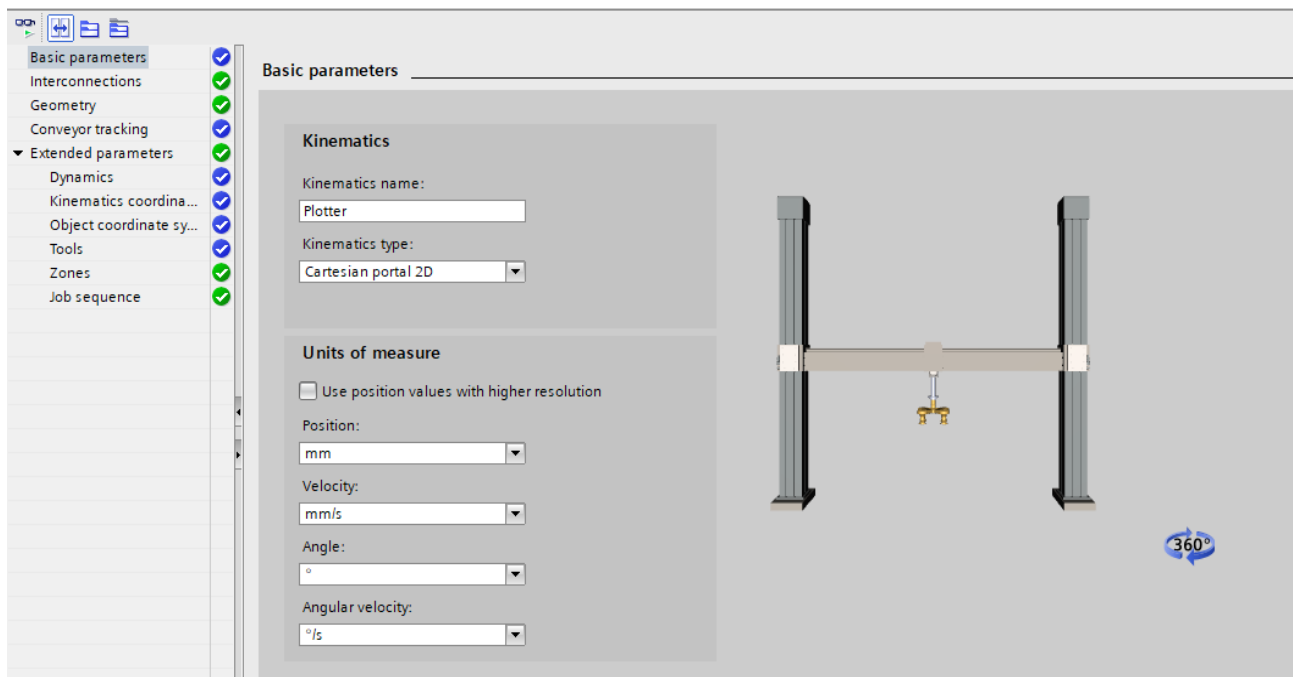
Pohyb pera v přípravku je zprostředkován z pohledu projektu dvěma technologickými objekty pera a papíru. Pro koordinaci obou os založíme technologický objekt Kinematiky, který sdružuje technologické objekty obou os a zajišťuje jejich vzájemnou spolupráci a koordinaci při složitějším požadavku logického řízení. Technologický objekt ploteru využijeme v návrhu souřadnicového pohybu pera.

V projektu najdeme technologický objekt Kinematiky na stejném místě jako předchozí objekty Polohovací osy v záložce *Technology objects*. Postup vložení technologického objektu do projektu je totožný s předchozím postupem vložení *TO Positioning Axis*, a proto se jenom krátce zmíníme o výběru správného *TO* z nabídky. V nabídce technologických objektů opět vybereme sekci *Motion Control* pro řízení pohonů a tentokrát vybereme *To_Kinematics*, předposlední od konce nabídky. Jeho potvrzení nás opět přesune do konfiguračního okna *TO*.



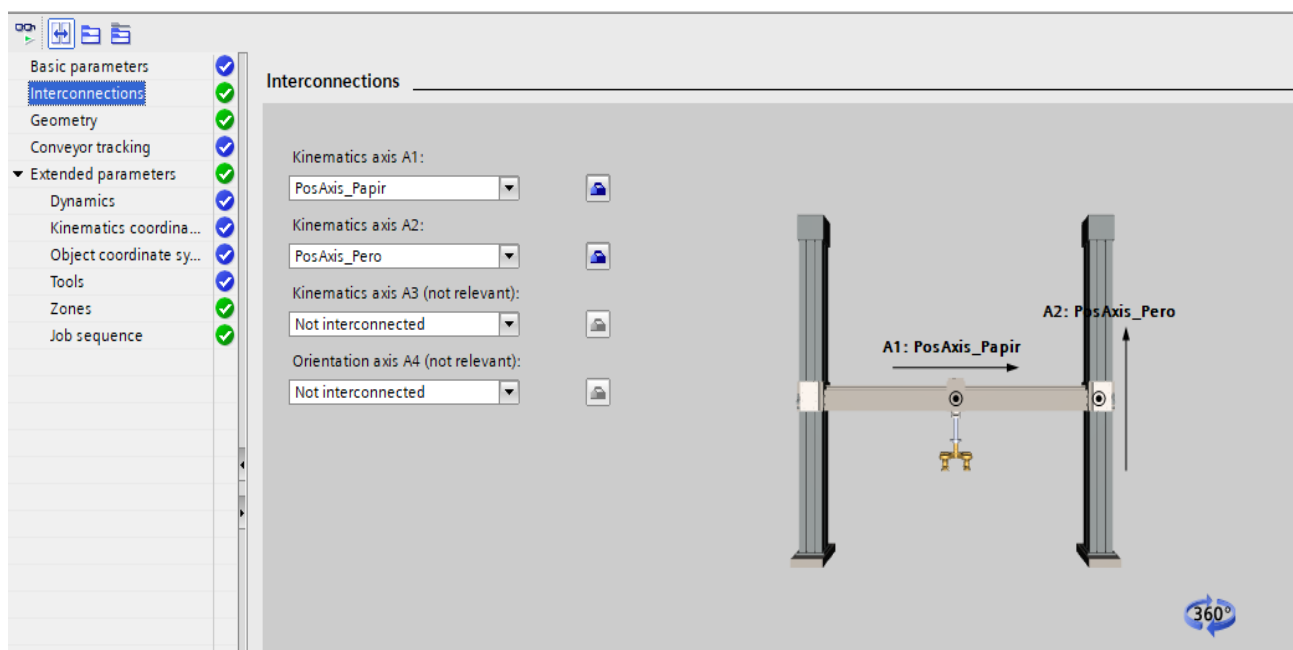
Obr. 8.7 Založení TO Plotter

V rámci konfigurace začneme v sekci *Basic parameters*. Jak je patrné z obr. 8.8, *TO* byl pojmenován Plotter a pro naše účely nejlépe poslouží výběr *Cartesian portal 2D* z nabídky *Kinematics type*. Volbou vznikne komplikace s pojmenováním os, jelikož prostředí TIA Portal v tomto případě pracuje s XZ rovinou namísto běžně zavedené XY roviny pro účely 2D ploch. Více však o této záležitosti v pozdější fázi návrhu.



Obr. 8.8 Konfigurace TO Plotter – *Basic parameters*

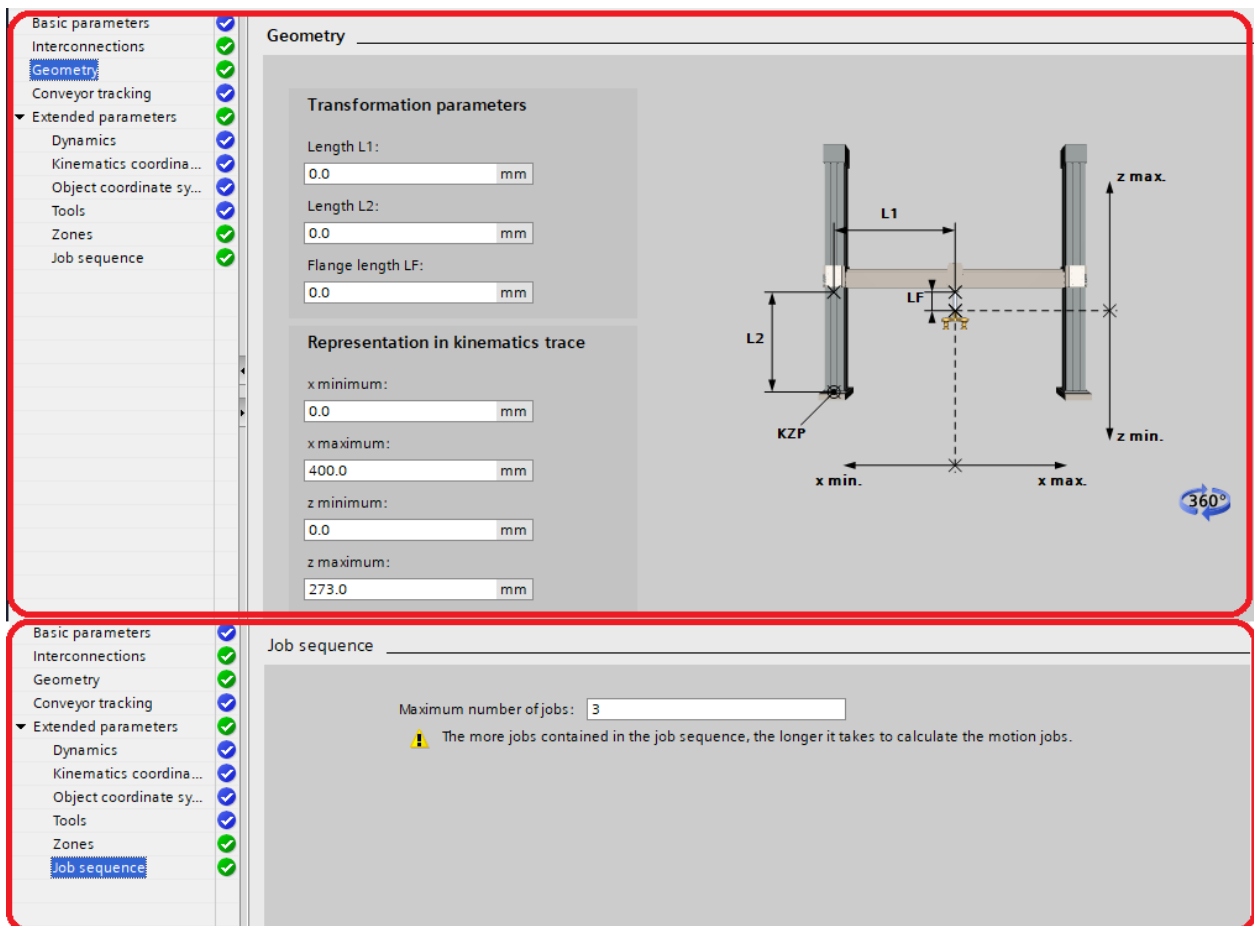
Pokračujeme sekci *Interconnections*. Zde je nutné správně zvolit propojení technologických objektů z hlediska postavení, neboť s propojením souvisí také provedení referování a nastavení kladného směru dané osy. Po úspěšných pokusech byl vybrán tento výběr propojení, k dispozici na obr. 8.9.



Obr. 8.9 Konfigurace TO Plotter – *Interconnections*

V sekci *Geometry* doplníme limitní hranice pohybu, která je dána rozměry A3 formátu. Doplnění je opět k dispozici na obr. 8.10.

Poslední částí konfigurace je nastavení maximálně tří vykonávaných pohybových příkazů najednou v sekci *Job sequence*.



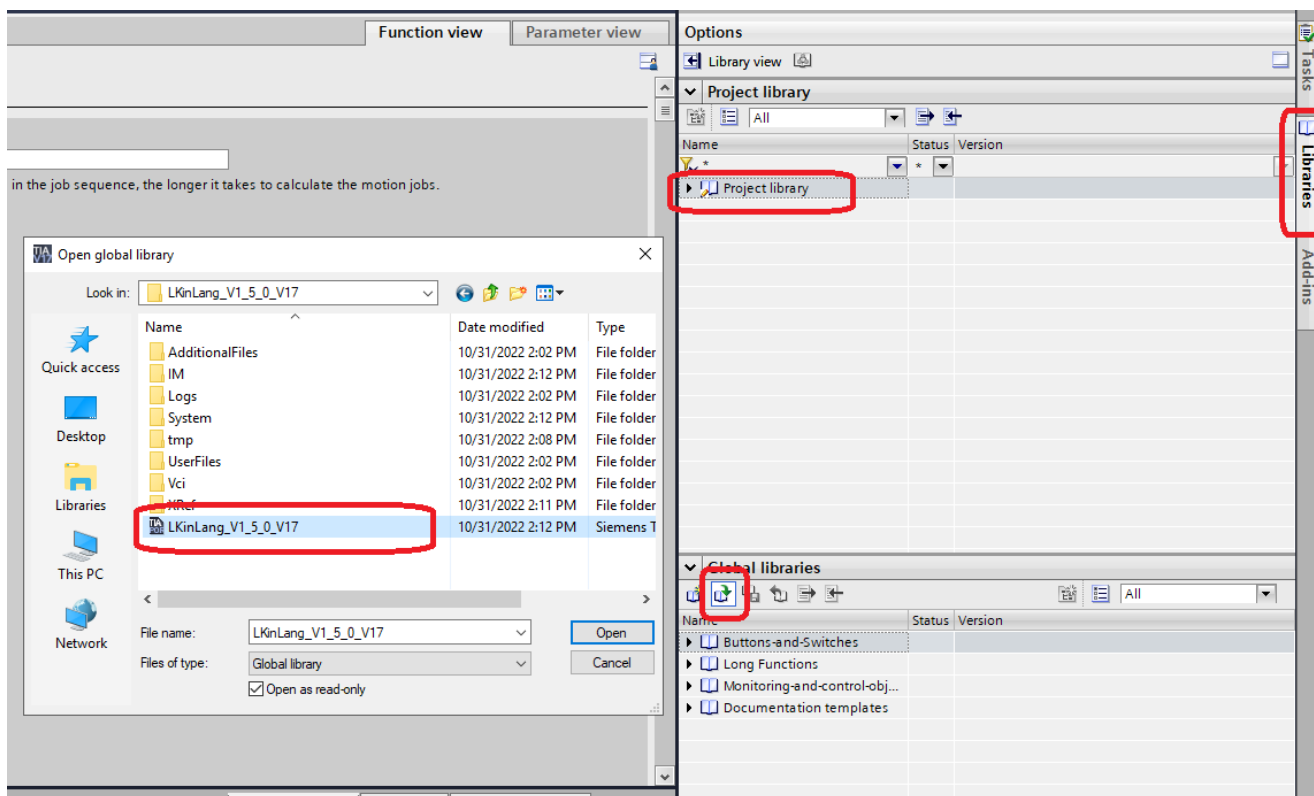
Obr. 8.10 Konfigurace TO Plotter – *Geometry a Job sequence*

Tímto máme zcela hotovou hardwarovou reprezentaci přípravku v projektu a můžeme začít návrhem logického řízení a obsluhy ploteru.

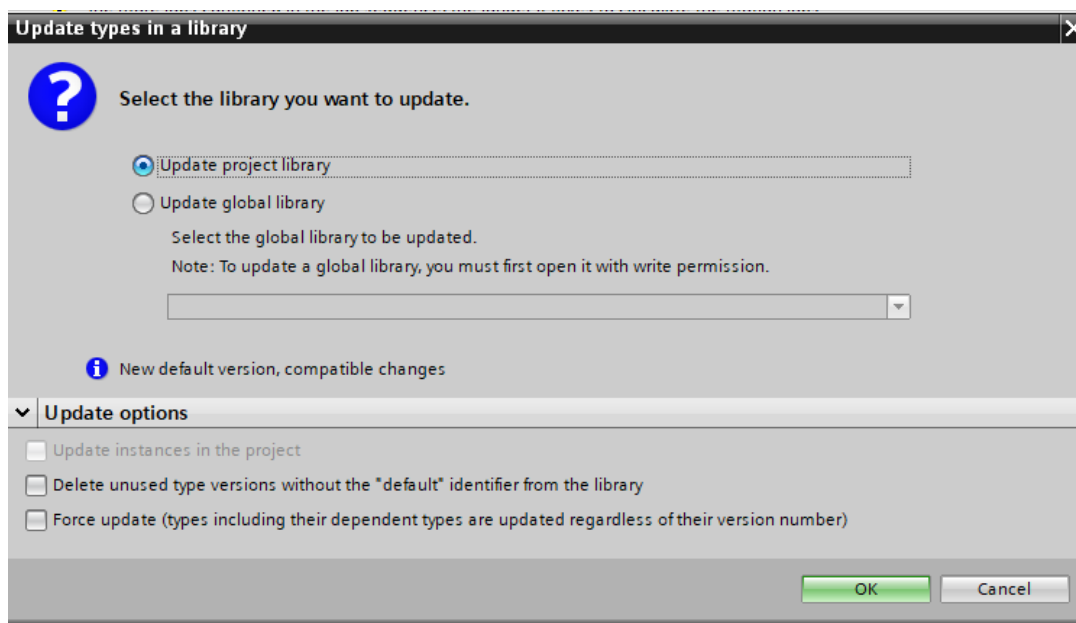
8.4 Import externích knihoven do projektu

Na pravé straně hlavního okna je potřeba otevřít záložku *Libraries*. Následně v dolní části *Global libraries* klikneme na zvýrazněnou ikonu *Open global library*. V adresáři vybereme požadovanou knihovnu a nahrajeme ji do prostředí TIA Portal. Po nahrání je knihovna vidět v nabídce *Global Libraries*. Nabídku knihovny rozklikneme a všechny soubory v sekci *Master copies* přesuneme pomocí *Drag & drop* do projektového PLC zařízení. Pro správné využití funkčních bloků knihovny musí být v projektové knihovně nahrána část externí knihovny. Import provedeme pravým kliknutím na externí knihovnu a dále vybereme *Update -> library*. Objeví se potvrzovací okno, v němž zkontrolujeme, že je vybrána aktualizace projektové knihovny místo globální. Při kontrole musí být vybrána položka *Update project library*. Část knihovny se nahraje do projektové knihovny. Jedná se o tzv. *faceplate*, tedy HMI aplikace, ovládající určité funkční bloky externí knihovny. Tuto část knihovny využijeme při vizualizaci HMI panelu.

Podrobný návod je k dispozici zde [35].



Obr. 8.11 Import externí knihovny do TIA Portal prostředí



Obr. 8.12 Aktualizace projektové knihovny

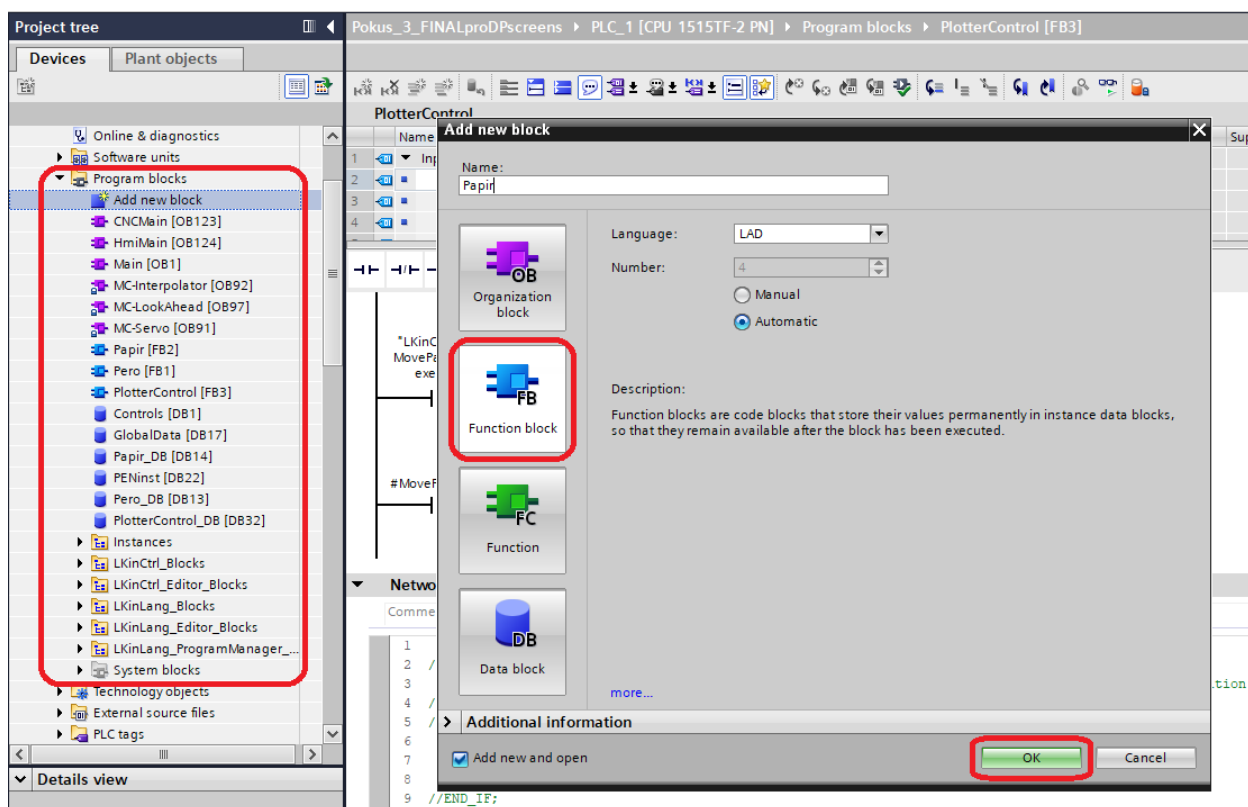
8.5 Implementace Motion Control řízení os

Především pro účely provedení správného referování a možnosti manuálního ovládání je vhodné vytvořit logické řízení každé osy pohonu zvlášť. Využijeme k tomu nabídku funkčních bloků *Motion Control*. Nástroje *Motion Control* najdeme standardně v záložce *Instructions* v záložce *Technology*.

8.5.1 Řízení osy

Vzhledem k tomu, že řízení technologického objektu osy představuje velkou a především složitou část PLC programu, vytvoříme funkční blok týkající se dané osy. Funkční blok bude obsahovat veškeré náležitosti řízení dané osy. Funkční blok následně umístíme do hlavní funkce Main (OB1), kde bude cyklicky volán. Vytvořená struktura umožní snadnější organizaci lokálních proměnných bez zbytečného zabírání místa v operační paměti. Přestože používáme vyšší řadu PLC S7-1515TF-2, operace s textovými soubory typu G-code znamenají opravdu maximální využití operační paměti řídicího zařízení.

V záložce *Program blocks* PLC zařízení zvolíme možnost *Add new block* a vložíme do projektu nový *Function block* s názvem *Papir*, více na obr. 8.13. Stejně založíme funkční blok pro osu *Pera* s názvem *Pero*.



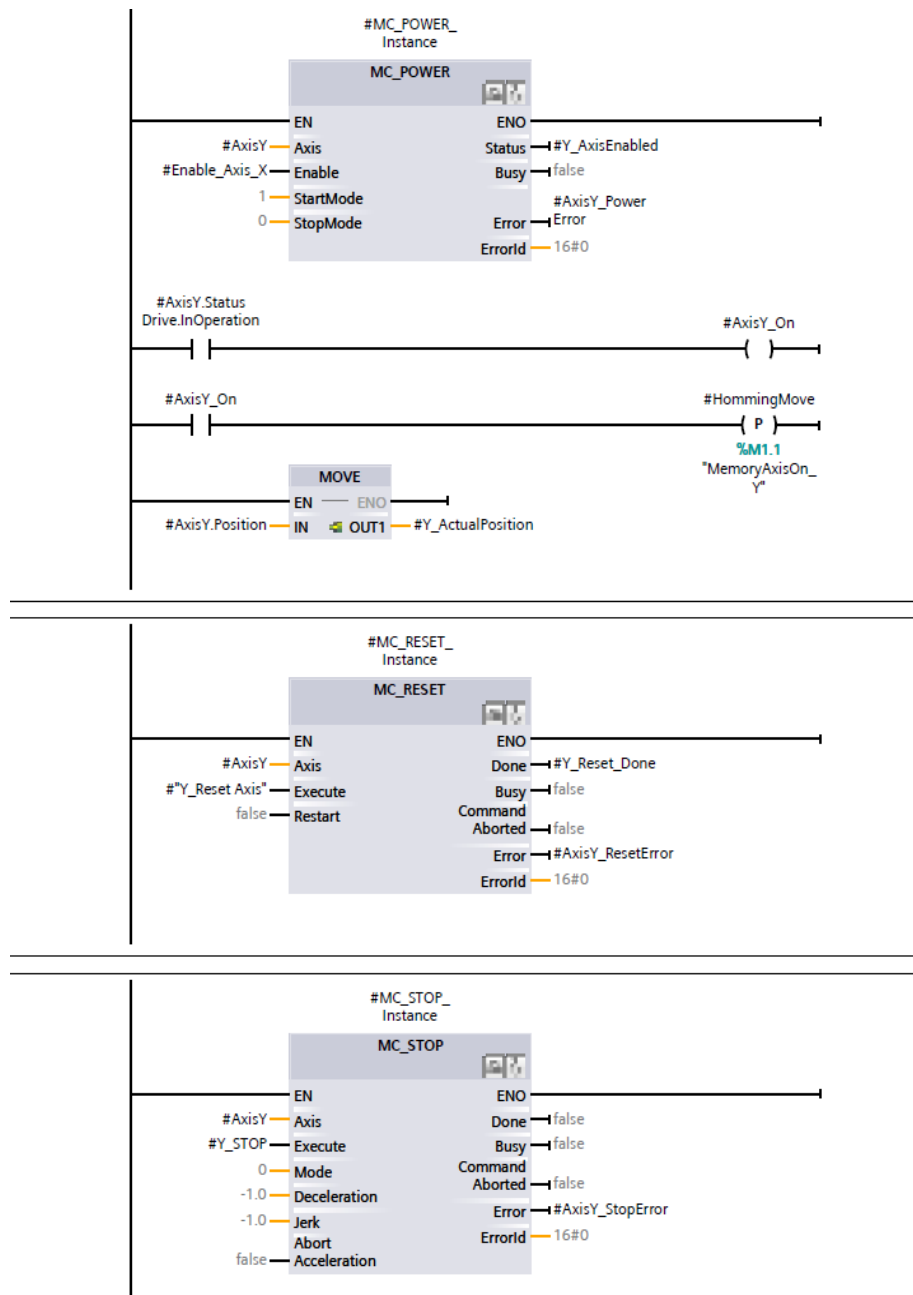
Obr. 8.13 Vložení FB bloku

K řízení chodu osy využijeme bloky *MC_POWER*, *MC_RESET* a *MC_STOP*. Pro jejich implementaci stačí požadované funkce přesunout pomocí *Drag & drop* do projektu. Otevřeme tedy příslušný funkční blok pera či papíru a v hlavním okně přidáme do programu výše zmíněné bloky z nabídky na pravé straně okna. Kvůli organizaci každý blok vložíme do své vlastní sekce, v prostředí nazvané *Network*. Sekci můžeme pro přehlednost pojmenovat dle funkce, název se však nikam nepromítne.

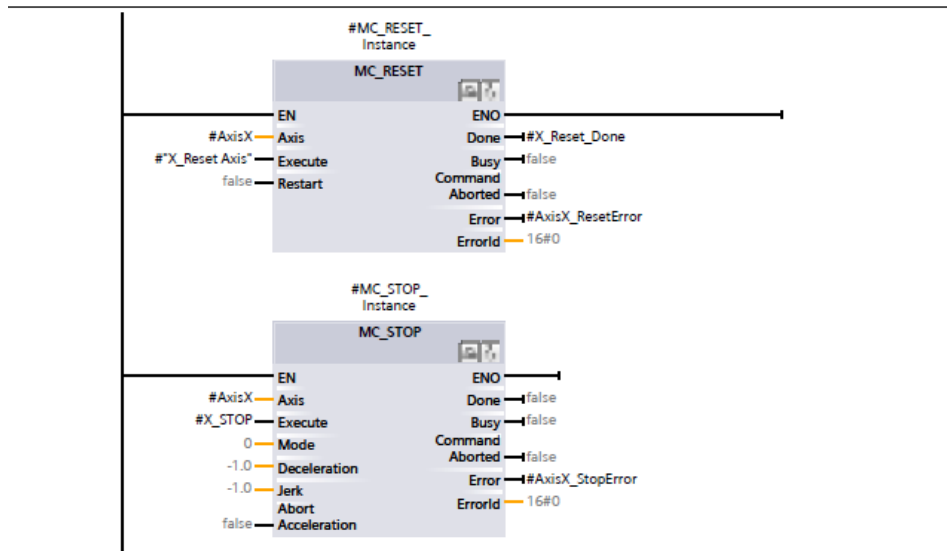
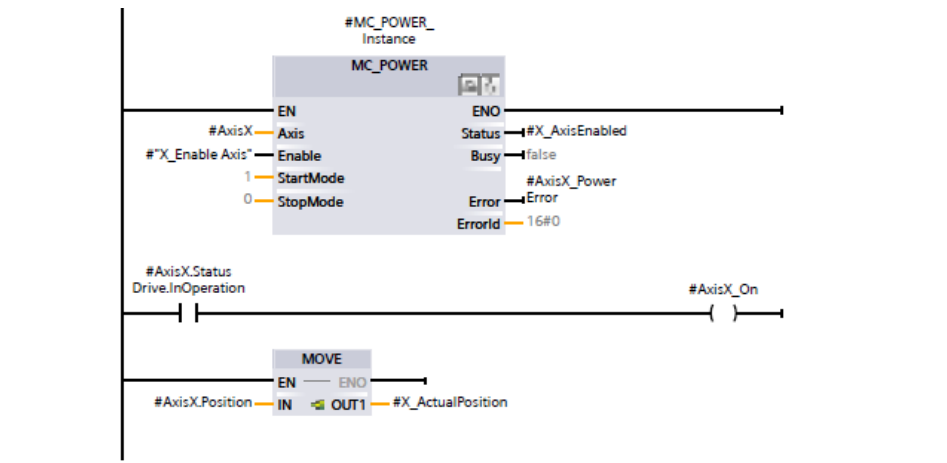
Použití funkčních bloků v programu má určitá specifika. Záleží, jestli je organizační blok programovaný v jazyku FBD či LAD. Programovací jazyky jde libovolně přepínat, vývojové prostředí dosavadní postup umí automaticky transformovat. Funkční blok musí být připojen k napájení u *EN* značky. Abychom mohli řídit a číst funkční blok, musíme k funkčnímu bloku definovat patřičné proměnné.

V našem případě byla k funkci *MC_POWER* přidána logika starající se o signalizaci zapnuté osy skrze čtení statusových bitů řízeného *TO*. Sekce napájení v rámci řízení technologického objektu papíru již obsahuje část logiky pro referování osy. Jedná se o sepnutí pohybu pro

referování v závislosti na signalizaci zapnuté osy. Kompletní program chod obou os je k dispozici na následujících obrázcích 8.14 a 8.15.



Obr. 8.14 LAD schéma FB bloku Papier, chod osy



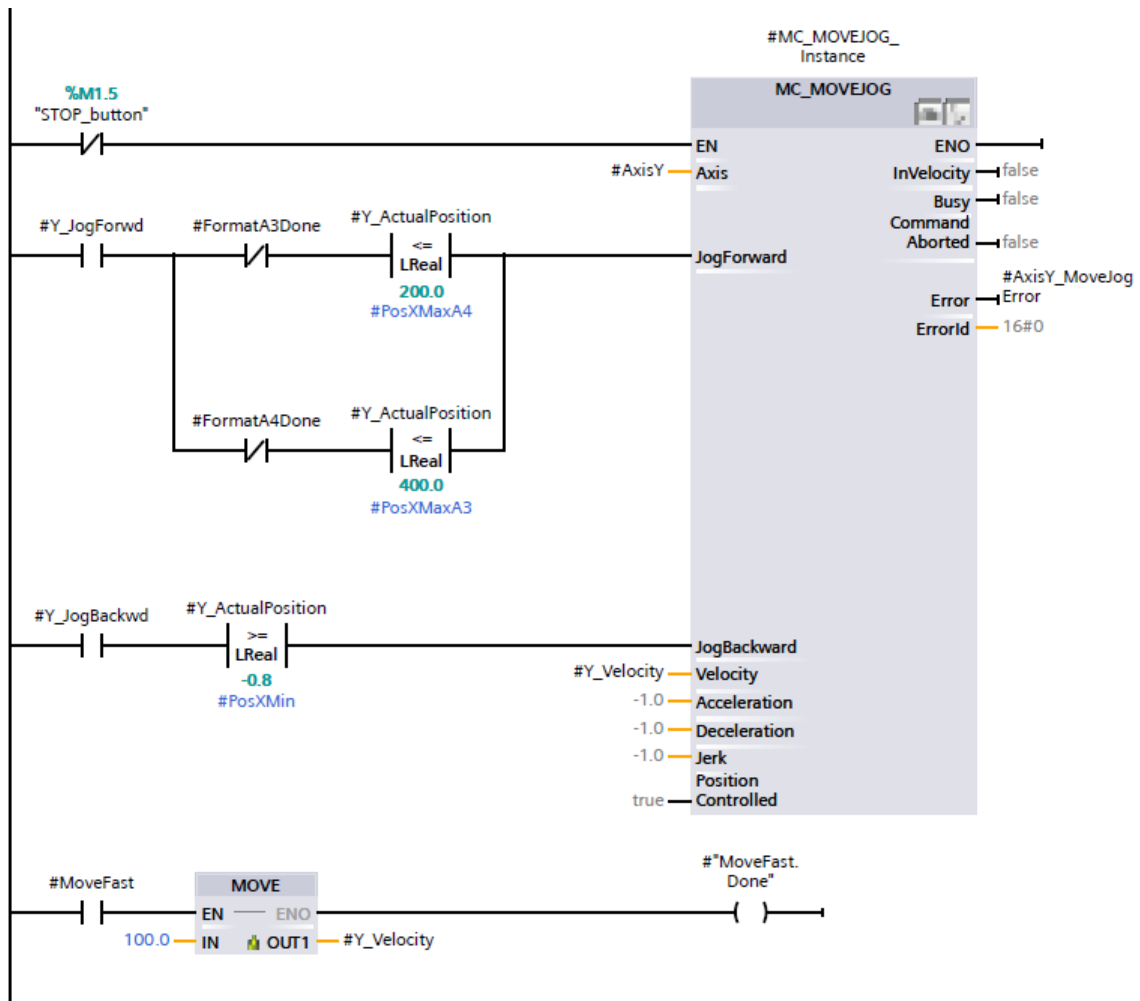
Obr. 8.15 LAD schéma FB bloku Pero, chod osy

8.5.2 Řízení manuálního pohybu osy pera a papíru

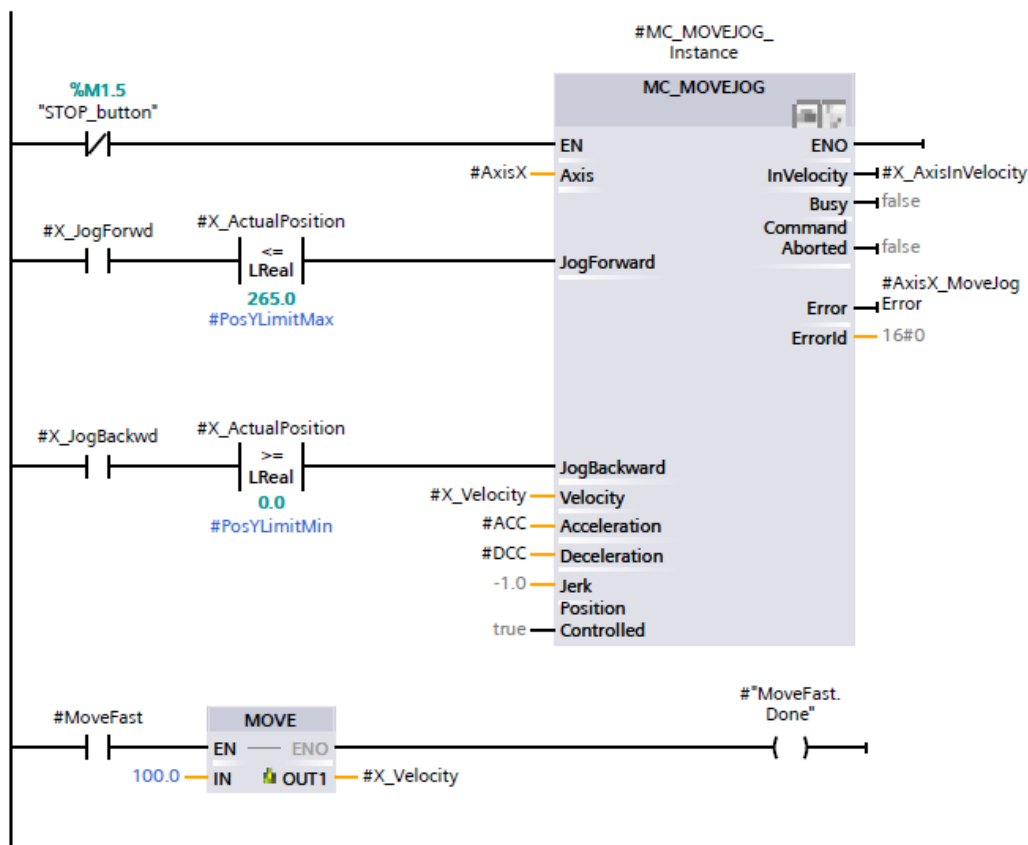
Manuální řízení obou os je možné realizovat řízením os odděleně. Nepotřebujeme k tomu tak využívat pokročilejší kinematické knihovny *Kinematics*. Stačí použít základní bloky *Motion Control* z vývojového prostředí TIA Portal. Úskalí řízení technologických objektů zvláště leží v pohybu, na kterém se podílí obě osy.

Jak dále uvidíme, ovládací HMI panely spol. Siemens nedisponují *multi-touch* technologií, z čehož vyplývá, že pro každý směr pohybu bude muset existovat jedno řídicí tlačítko. Při odděleném řízení stačí vytvořit příkazy, u kterých se na pohybu podílí pouze jedna osa (nahoru-dolů, doprava-doleva) nebo obě osy rovnoměrně, tzn. pohyb do směru natočeného o 45 stupňů od vertikální, či horizontální osy souřadného systému.

Pro manuální řízení pohybu pera je nejvhodnější tzv. *jog* pohyb, který vykonává funkční blok *MC_MOVEJOG*. Funkční blok pomocí *Drag & drop* přesuneme jak do bloku Papír, tak i Pero, opět do vlastní sekce. Realizaci pohybu osy papíru vidíme na obr. 8.16 a pera na obr. 8.17.



Obr. 8.16 LAD schéma FB bloku Papir, manuální řízení pohyb



Obr. 8.17 LAD schéma FB bloku Pero, manuálně řízení pohyb

Pohyb osy je vykonáván spínáním vstupů *JogForward* a *JogBackward*, tedy pohyb vpřed a pohyb dozadu. Orientace je dána konfigurací *TO* společně s nastavením enkodéru v servomotoru. Pro spínání pohybu vytvoříme lokální logické proměnné *Y_JogForwd* a *Y_JogBackwd*. Lokální proměnné můžeme vytvářet v horní části hlavního okna uprostřed. Logické proměnné pohybu pomocí spínacího kontaktu spínají pohyb jedné osy dozadu či dopředu. Nesmíme však zapomenout, že pohyb papíru není umožněn, pokud nebyl do přípravku vložen papír. To zajišťuje rozpínací kontakt signálu *FormatAxDone*. Máme dva možné formáty, a proto zde máme dvě paralelní větve. Poslední část řízení tvoří komparační blok, který funguje jako koncový spínač okraje papíru. Opět závisí, jaký formát papíru byl vložen do přípravku. Na rozdíl od rozpínacího kontaktu komparátor v závislosti na poloze osy neumožní pohyb za hranici papíru, ale umožní pohyb zpět v případě, že se pero nachází na hranici. Rozpínací kontakt by další pohyb již neumožnil. Nastavené referenční hodnoty komparátoru odpovídají formátům A3 a A4 s ohledem na experimentální odzkoušení funkčnosti návrhu.

8.6 Realizace manuálního referování os

Provedení tzv. *homing* funkce je základní podmínkou pro absolutní polohování pohybu technologického objektu kinematiky pro náš přípravek. Vzhledem k záměru souřadnicového pohybu s využitím technologie CAM musí být referování provedeno ve správném směru a s použitím referenční hodnoty přesně v počátku souřadnicového systému.

Servomotory SIMOTICS 1-FK2 jsou v našem případě vybaveny absolutním enkodérem. Tento typ bohužel neumožňuje tzv. *Active homing*, tedy automatický pohyb referování až do nastavené polohy reference. Výhoda aktivního referování spočívá v tom, že nezáleží, v jaké poloze se vypnutá osa nachází. Pro naše účely musíme však vytvořit vlastní řízený pohyb referování. Zde přichází ale komplikace, neboť přípravek nedisponuje koncovým spínačem na obou koncích. K dispozici máme pouze jeden koncový spínač v rámci osy pohybu pera, a to na opačném konci, než potřebujeme z hlediska umístění počátku souřadnicového systému. Nutnost zachovat referenční

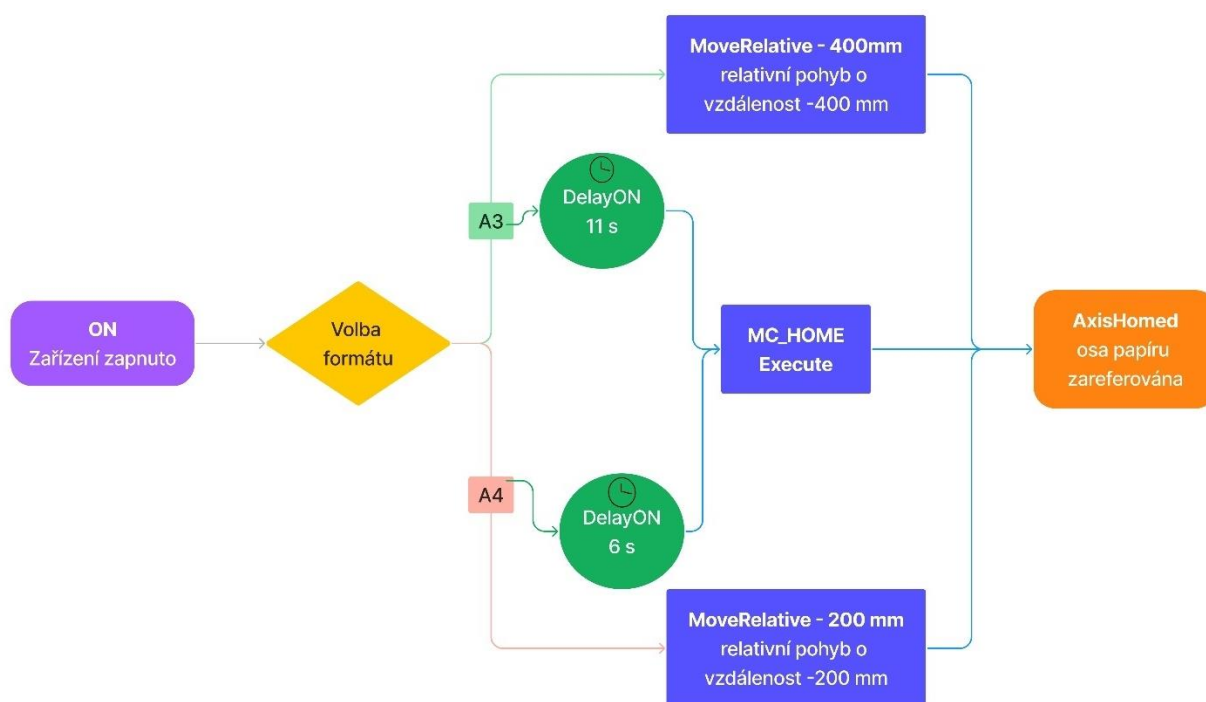
polohu v levém dolním rohu vloženého papíru souvisí s orientací kartézského souřadnicového systému. Pro účely vykreslování totiž potřebujeme vytvořit pravotočivou bázi našeho systému při zachování nastaveného směru čítání enkodéru obou os.

Funkci *homming* má na starosti blok z nabídky *Motion Control MC_HOME*. Blok opět jednoduše vložíme do organizačního bloku Papír a Pero. Pro naše účely tzv. pasivního referování je nutné nastavit režim bloku *MC_HOME* na hodnotu 7, tzv. *Mode* parametr. Jedná se o režim, při kterém určíme aktuální polohu osy jako referenční při vybavení bloku *MC_HOME*. Aktuální poloha se tak v daný okamžik stane počátkem osy.

Návrhem vlastního pohybu referování osy papíru a pera se věnují následující podkapitoly Referování osy papíru a Referování osy pera.

8.6.1 Referování osy papíru

V případě osy řídící pohyb papíru je referování o poznání jednodušší. Přestože na přípravku pro tuto osu nenajdeme žádné koncové spínače, je možné provést referování vcelku snadno. Synchronizační pohyb provedeme jedním příkazem relativního posunu. Využijeme skutečnosti, že papír se před spuštěním zařízení vkládá do přípravku tak, že je poloha osy jeho pohybu přesně na opačném konci. Stačí tedy po uvedení osy do provozu provést relativní posun papíru o délku jeho jedné hrany. V případě formátu A4 se jedná o kratší hranu dlouhou 200 mm, u formátu A3 se jedná o delší z hran o délce 400 mm.



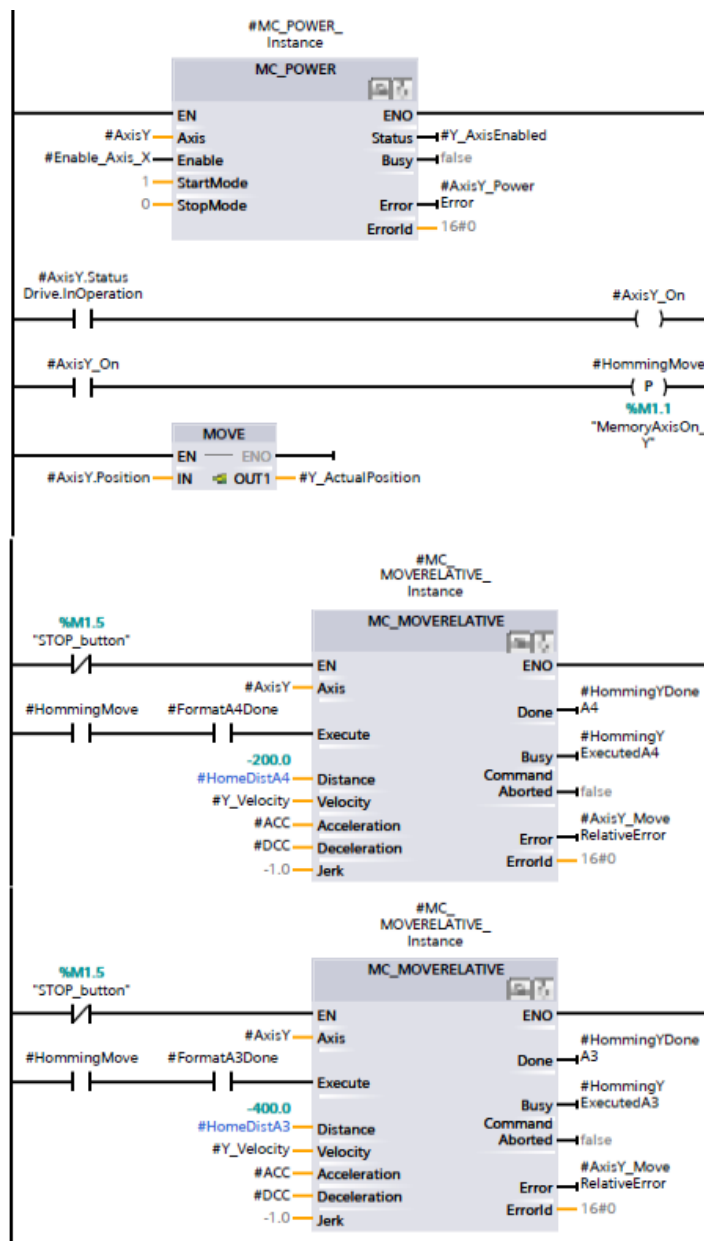
Obr. 8.18 Vývojový diagram manuálního pohybu referování, TO Papír

K realizaci využijeme blok relativního pohybu *MC_MOVERELATIVE* z nabídky *Motion Control* bloků. Blok vložíme do organizačního bloku, opět doporučuji do vlastní sekce. Provedení pohybu podmíníme spínacím kontaktem zapnuté osy a zvoleným formátem papíru.

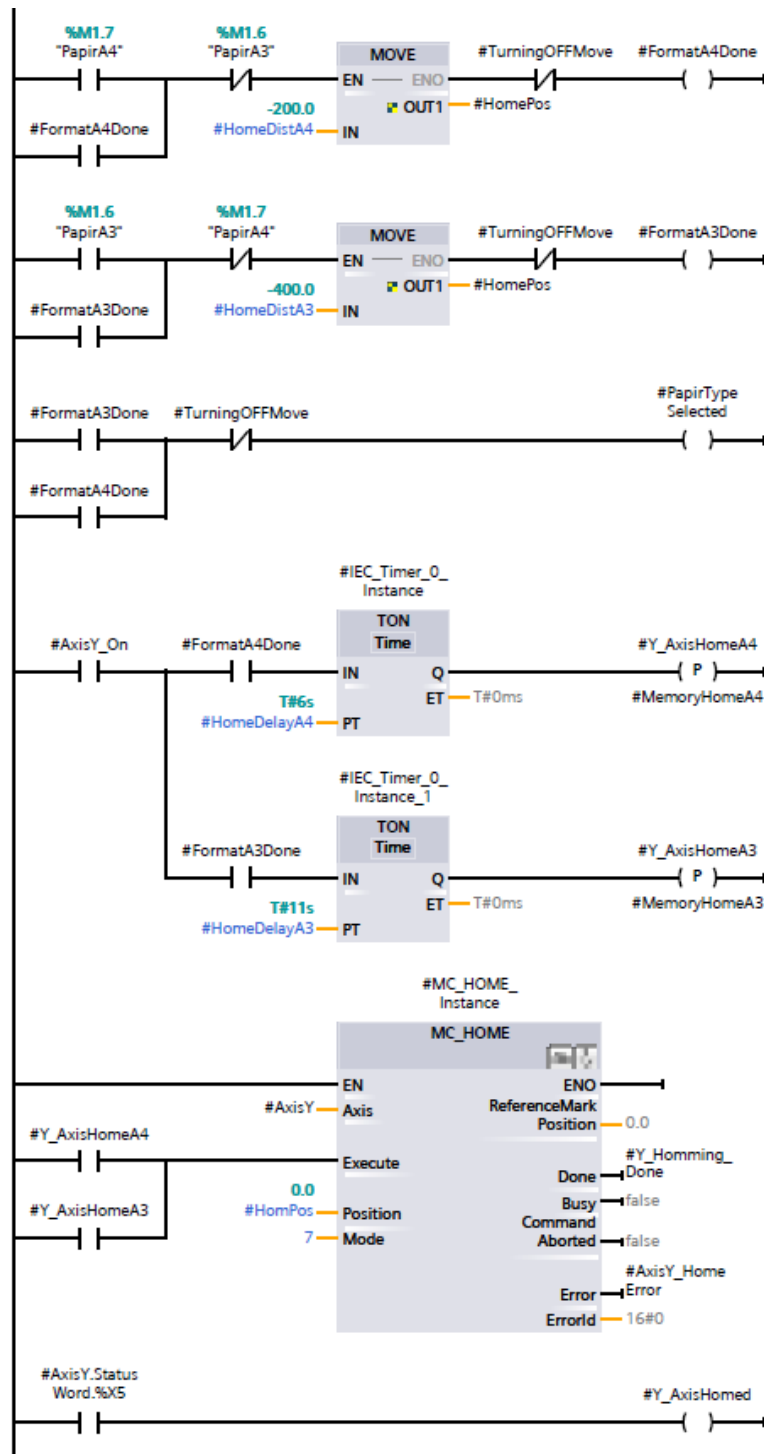
Logiku volby formátu papíru je třeba vytvořit souběžně s pohybem reference. Volba formátu je zprostředkována skrze vnitřní PLC tagy ovládané z HMI panelu. Volba jednoho formátu automaticky vylučuje volbu formátu druhého pomocí rozpínacího kontaktu křížem. Dále je signalizace zvoleného formátu vypnuta pomocí rozpínacího kontaktu *TurningOFFMove*, který zahajuje pohyb papíru do parkovací polohy při vypínání zařízení. V parkovací poloze je papír pak

možné vysunout z přípravku. Celá logika je k dispozici na obr. 8.19 a 8.20. O nastavení papíru do parkovací polohy při vypnutí bude zmínka v následující podkapitole.

V návrhu bylo implementováno řešení s dvěma použitými bloky MC_MOVERELATIVE se staticky nastavenou hodnotou vzdálenosti, jejich spuštění je podmíněno zvoleným formátem papíru. Logika je však připravená pro použití pouze jednoho bloku pohybu. K tomu stačí použít na vstupu *Distance* lokální proměnnou *HomePos*. Proměnná *HomePos* je již součástí programu a mění hodnotu v závislosti na vybraném formátu. Toto řešení nepatrně ušetří místo v operační paměti.



Obr. 8.19 LAD schéma FB bloku Papir, manuálně pohyb referování, část 1



Obr. 8.20 LAD schéma FB bloku Papir, manuálně pohyb referování, část 2

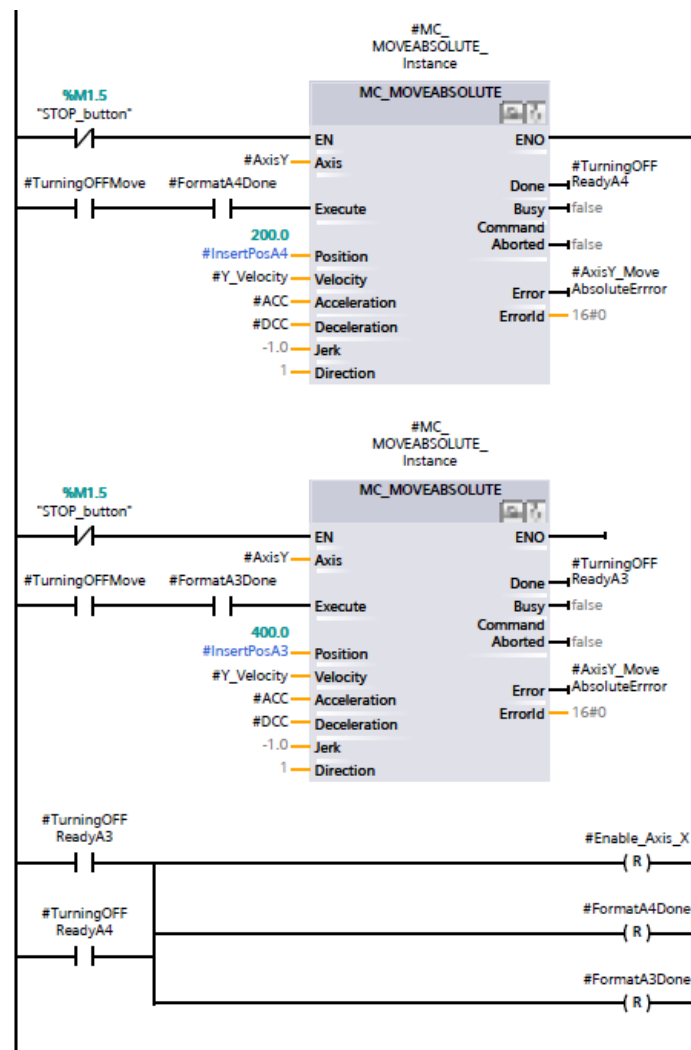
8.6.2 Nastavení papíru při vypnutí

V případě opakovaného používání ploteru je nutné při vypnutí přípravku vrátit papír do tzv. parkovací polohy, kdy je možné papír vysunout ze zařízení.

Při vypínacím požadavku daný logickou proměnnou *TurningOFFMove* je spuštěn řízený pohyb do parkovací polohy papíru. Z hlediska zareferované osy papíru se jedná o pohyb papíru do koncové polohy. Záleží tedy na formátu vloženého papíru. K pohybu použijeme blok *Motion Control MC_MOVEABSOLUTE* s pevně danou cílovou polohou, pro formát A4 se jedná o hodnotu 200 mm a pro formát A3 se jedná o polohu 400 mm. Jakmile je pohyb dokončen,

signalizace dokončeného pohybu je impulzem pro vypnutí osy pomocí rozepnutí logického signálu *Enable_Axis_X* a vyresetování proměnných starajících se o výběr formátu papíru.

Kompletní návrh řešení je k dispozici na obr. 8.21.

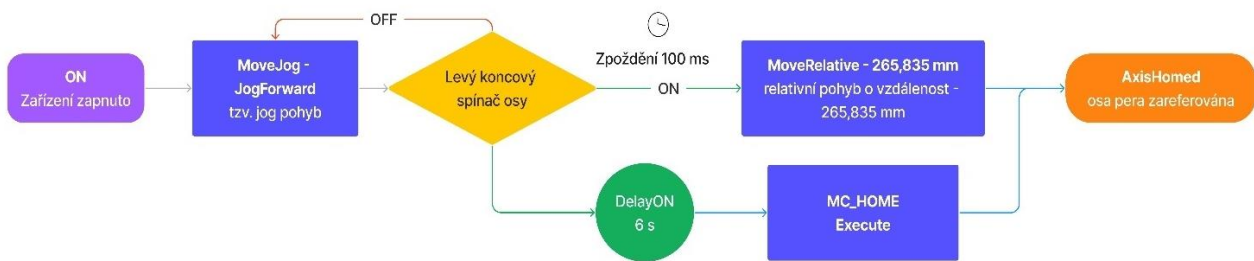


Obr. 8.21 LAD schéma FB bloku Papír, nastavení osy při vypnutí

Z obrázku 8.21 vidíme, že opět máme definované dva stejné bloky a jejich spuštění je podmíněno volbou formátu. Zároveň je na pohyb navázáno vypnutí osy.

8.6.3 Referování osy pera

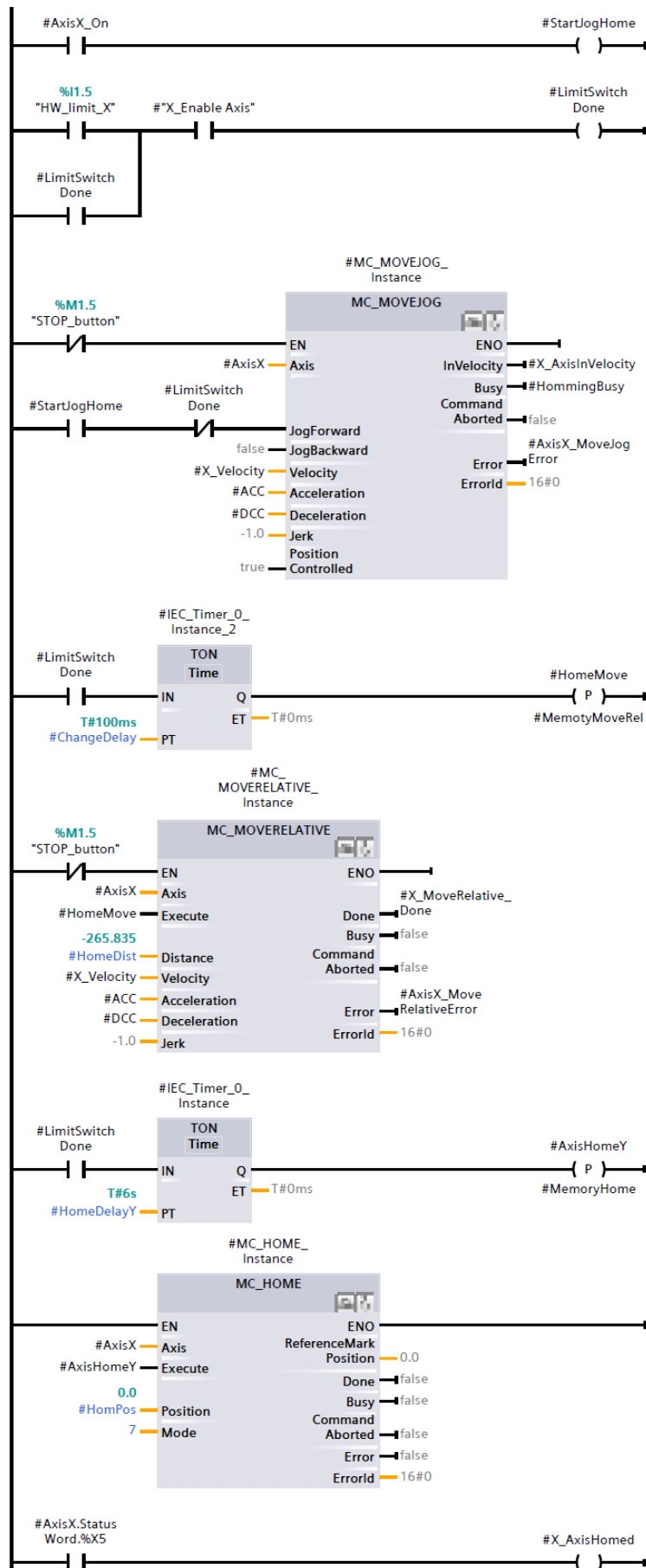
Referování osy pera využívá koncového spínače přípravku na opačném konci osy pera než je počátek pomyslného souřadného systému. Abychom mohli provést synchronizaci nezávisle na předchozí poloze pera, použijeme koncového spínače jako tzv. předreferenčního bodu. Při sepnutí osy nastavíme pohyb osy kladným směrem, tzv. *jog* pohyb, dokud nedorazí k limitnímu kontaktu. Jakmile spínač signalizuje konec osy, víme, že pero se nachází na opačném konci osy a provedeme relativní pohyb do pomyslného počátku osy. Po dokončení pohybu polohu označíme za referenční a spustíme blok *MC_HOME*.



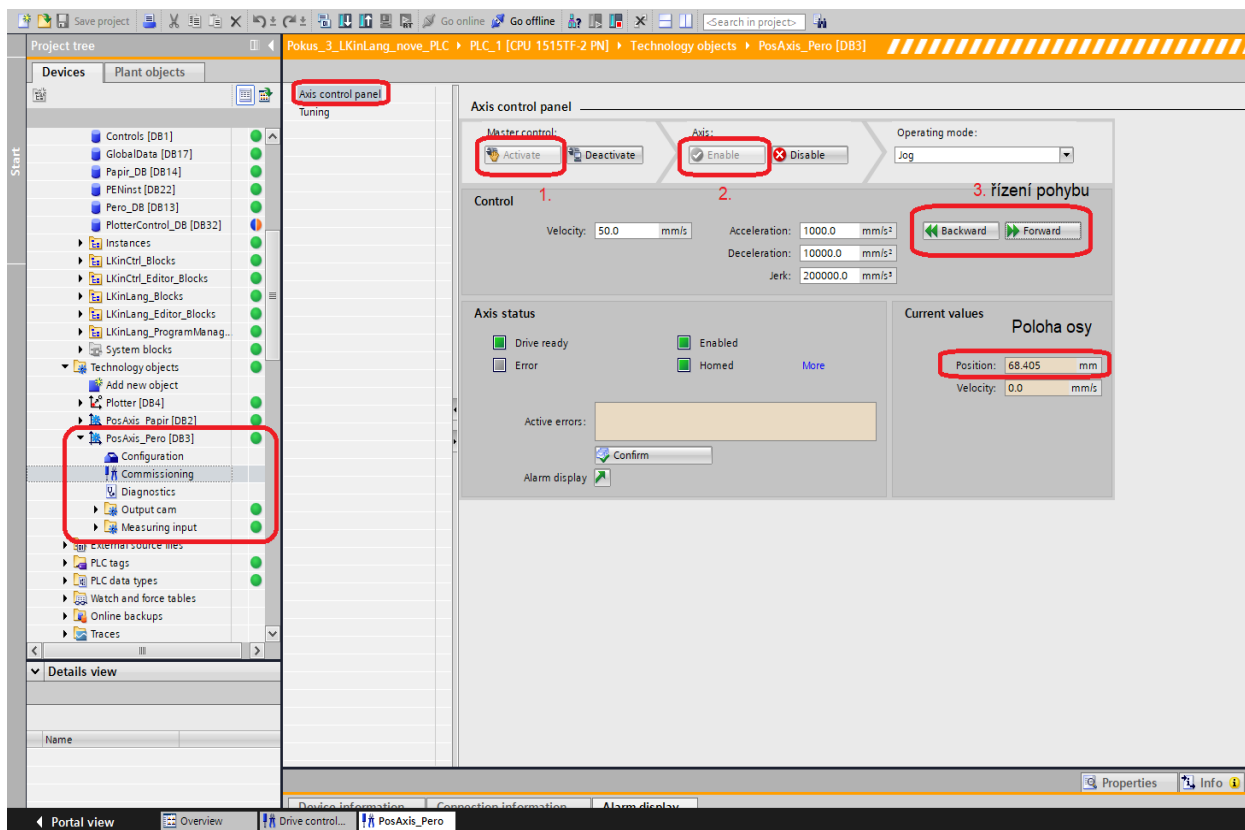
Obr. 8.22 Vývojový diagram manuálního pohybu referování, TO Pero

Dosud však neznáme vzdálenost relativního pohybu od koncového spínače do počátku. V ideálním případě by se jednalo o délku hrany papíru. Vzhledem však k provedení přípravku je vzdálenost lehce odlišná. Z tohoto důvodu je potřeba provést skrze *Online access* režim funkci *Commissioning* technologického objektu pera a vzdálenost ručně změřit. V tomto případě najdeme záložku odpovídajícího technologického objektu, v našem projektu *PosAxis_Pero*, a vybereme z nabídky aplikaci *Commissioning*. Přesuneme se do nového hlavního okna, zde zaklikneme *Activate -> Enable* a vybereme *Operating mode: Jog*. Poté stačí manuálně odečíst pozici u koncového spínače a polohu v počátku, jejich rozdíl bude pevně zvolená vzdálenost relativního pohybu referování. Názornější postup je zobrazen na obr. 8.24.

Návrh vlastního referování osy pera je kompletně k dispozici na obr. 8.23.



Obr. 8.23 LAD schéma FB bloku Pero, manuálně pohyb referování



Obr. 8.24 Odečtení vzdálenosti relativního pohybu referování

KAPITOLA 9: NÁVRH OBSLUHY PLOTERU

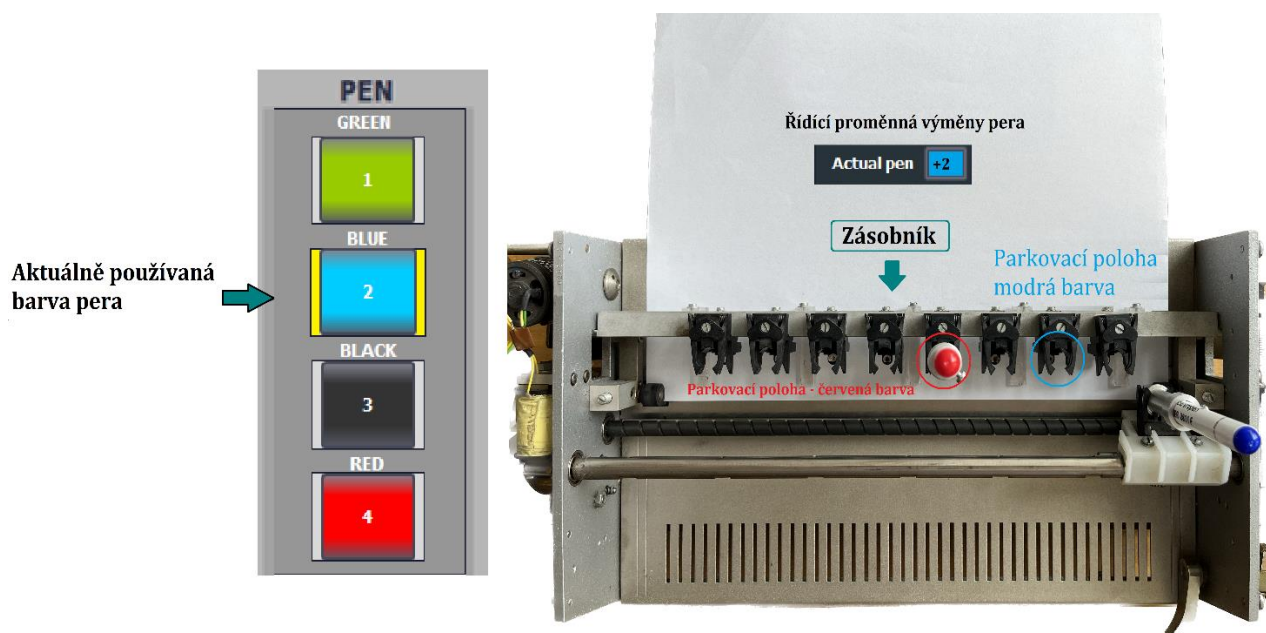
Tato kapitola má za cíl popsat realizaci obsluhy přípravku. Z kapitoly 5 Popis přípravku víme, že na ploteru najdeme zásobník připojený na stejnosměrný motor. Spínáním relé motoru při dodržení správného sledu logiky včetně správné polohy osy pera dosáhneme samočinné výměny pera mezi zásobníkem a uchycením pera pro kreslení. Je potřeba však navrhnout program, který se bude starat o organizaci zásobníku a dohlížet nad pohybem zásobníku a osy pera při výměně.

9.1 Technologické schéma výměny pera

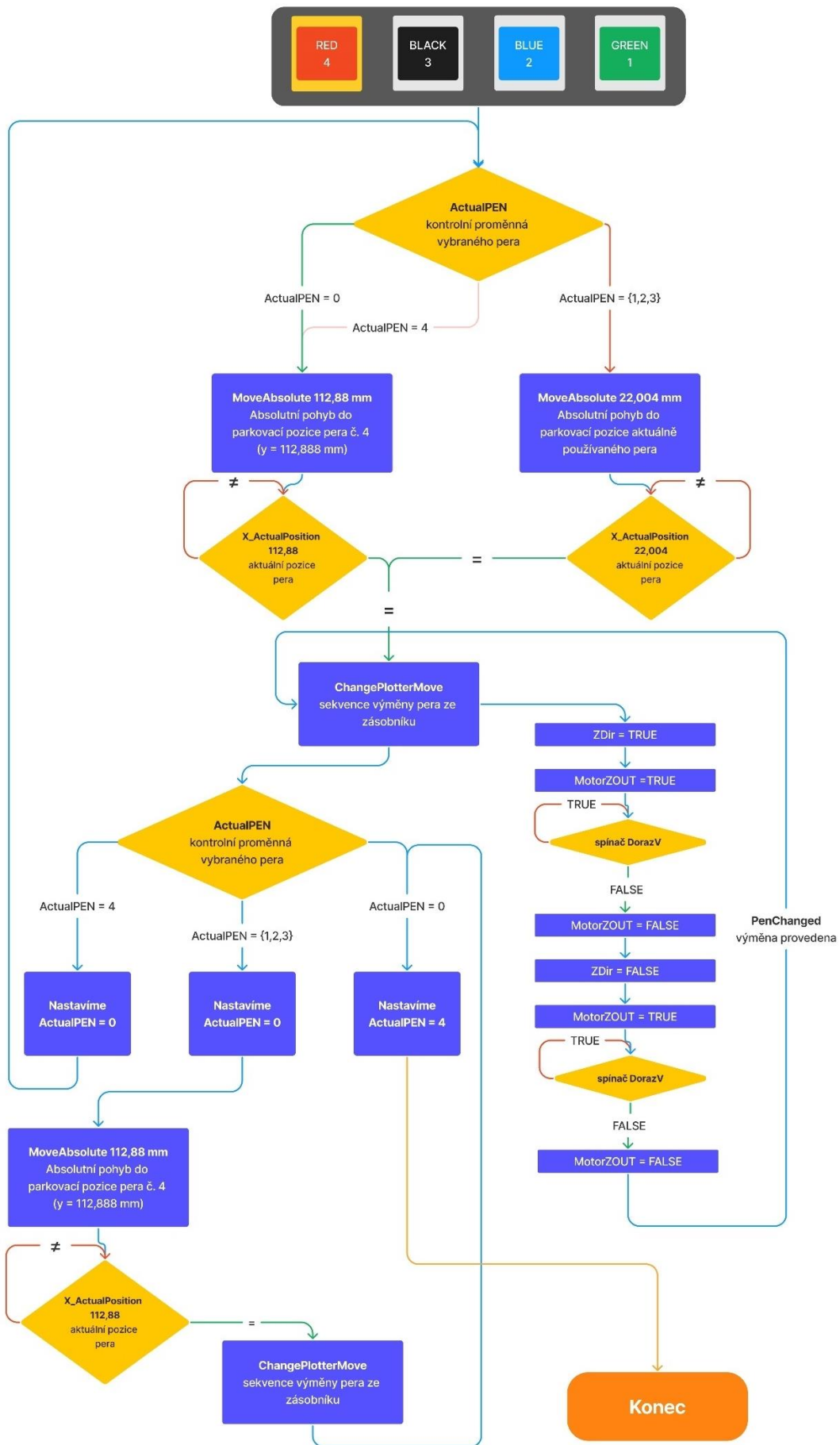
Abychom mohli vytvořit liniové schéma logického řízení v projektu, vytvoříme nejdříve technologické schéma dílčí úlohy projektu. Koncepční technologické schéma je naznačeno na obr. 9.1. Logický program umístíme v rámci organizace projektu do funkčního bloku řídicího osu pera.

Ovládání zásobníku omezíme na pouhá čtyři tlačítka reprezentující volbu pera s požadovanou barvou. Signalizace naší volby bude hlásit kontrastní změna zbarvení tlačítka pro danou barvu pera společně s číselným ukazatelem aktuálně zvoleného pera. Pokud žádné pero nebude v uchycení osy, nebude svítit žádné z tlačítek. Pro číselnou reprezentaci prázdného výběru zvolme číslo 0.

V situaci prvního výběru pera bude po stisknutí ovládacího tlačítka požadované pero ze zásobníku vybráno. To znamená přesun kreslicího uchycení přesně na osu pozice požadované barvy v zásobníku a provedení vyměňovací sekvence ploteru. Jakmile pero úspěšně vybereme, uloží se do paměti právě zvolené pero, resp. jeho charakteristická barva. V případě požadavku na změnu barvy se nejdříve provede automaticky vrácení aktuálně zvolené barvy do zásobníku a až teprve poté se vybere ze zásobníku momentálně požadované pero, resp. barva. Pokud budeme chtít pero dané barvy vrátit do zásobníku bez požadavku na jinou barvu, stiskneme znovu tlačítko dané barvy a pero se automaticky vrátí do zásobníku bez dalšího výběru.



Obr. 9.1 Technologické schéma výměny pera



Obr. 9.2 Diagram výměny pera

9.2 Návrh logického řízení výměny pera

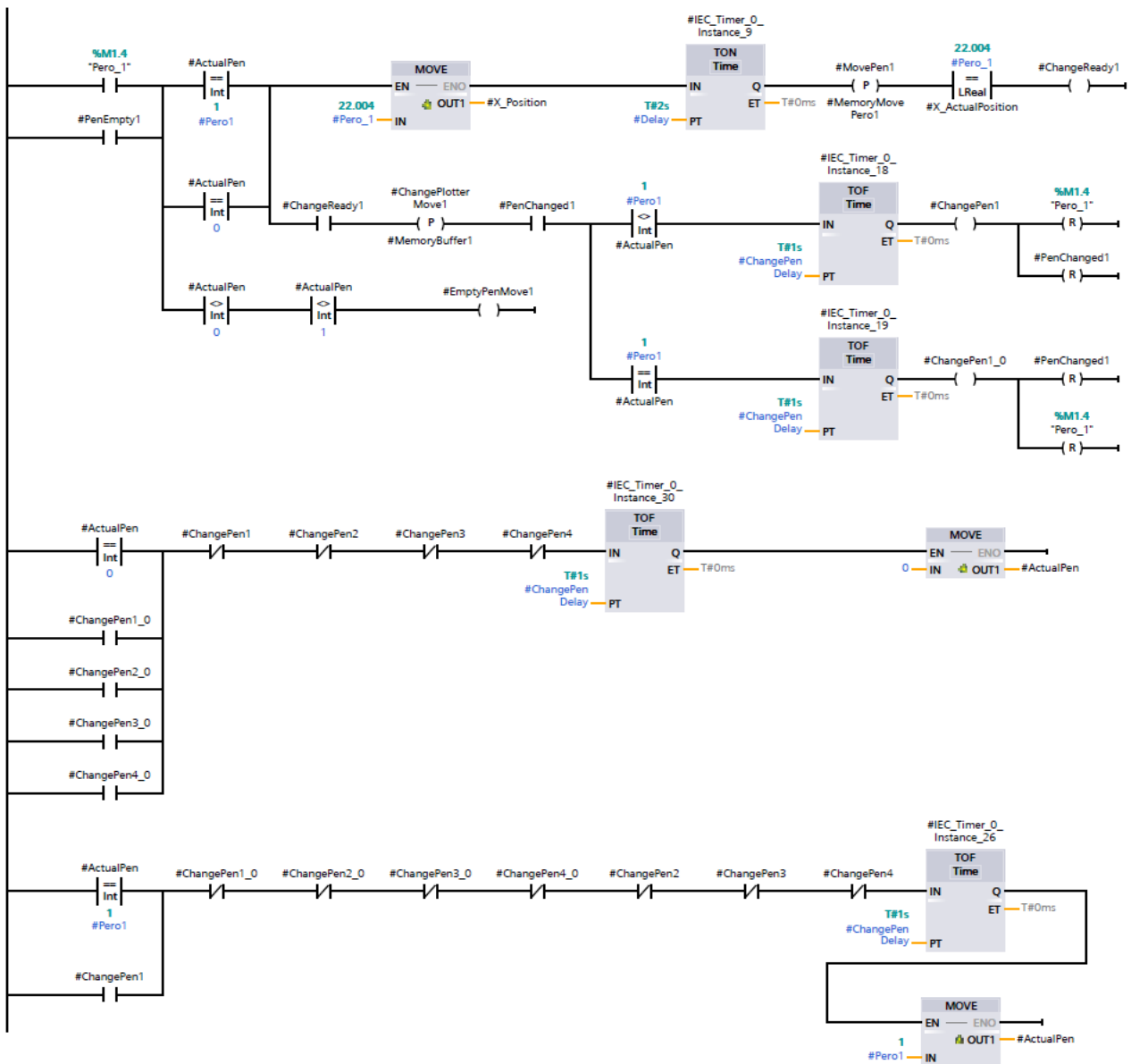
Výměna pera bude řízena vnitřními PLC tagy reprezentující požadavek na použití určité barvy pera. To znamená, že v zásobníku budou připravena celkem čtyři pera o různých barvách. Počet per v zásobníku lze s drobnými úpravami následně škálovat. Maximální počet míst v zásobníku činí 8. Barvám pro potřeby programu přiřadíme číselné hodnoty. Inicializujeme logickou proměnnou monitorující volbu používaného pera *ActualPen*. Předpokládáme, že u technologického objektu osy pera proběhlo referování a k pohybu mezi pozicemi v zásobníku využijeme tedy absolutního pohybu prostřednictvím bloku *MC_MOVEABSOLUTE*.

Každá barva pera bude mít svoji neměnnou pozici v zásobníku. Polohu osy pozice barvy v zásobníku je nutné pro výměnu experimentálně odměřit se zareferovanou osou pera skrze funkci *Commissioning TO PosAxis_Pero*. Postup se neliší od manuálního odečtení polohy referenční pozice při realizaci synchronizačního pohybu osy pera v podkapitole 8.6.3.

Jakmile zjistíme hodnoty parkovací pozice barev v zásobníku, můžeme realizovat vnitřní logiku. Logické řízení se bude skládat ze čtyřech základních větví, vždy pro každé pero určité barvy zvlášť.

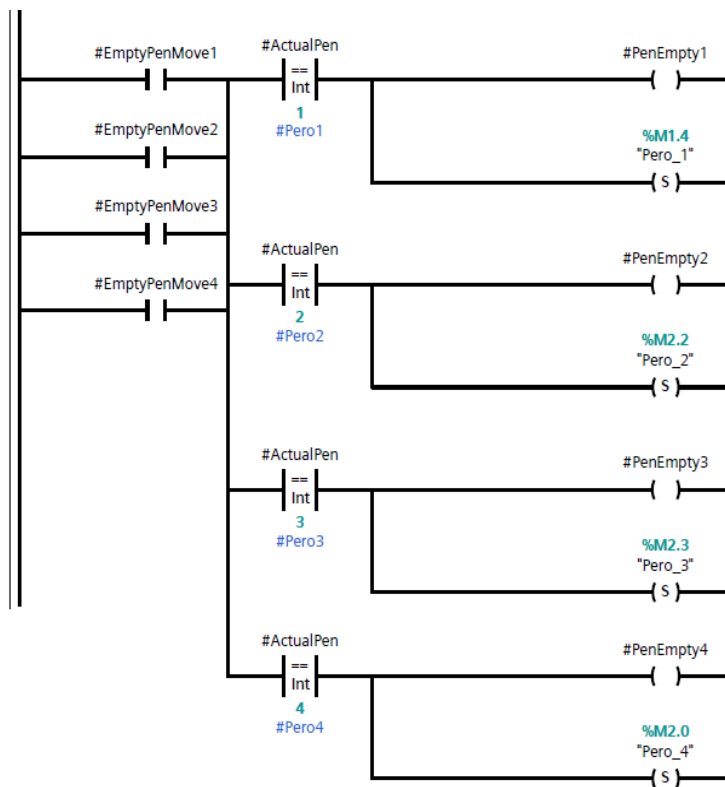
Větev je inicializována spínacím kontaktem vnitřní PLC tagu dané barvy pera, nebo příkazem vyprázdnění dané barvy pera při požadavku výměny aktuální barvy za jinou. Při vygenerování požadavku volby dané barvy pera mohou nastat tři situace. Logiku rozdělíme pomocí komparačního bloku kontrolující proměnnou aktuálně vybrané barvy pera. Situace jsou následující. Dosud jsme žádné pero nezvolili, v tom případě je hodnota proměnné *ActualPen* nulová, první situace. *ActualPen* má stejnou hodnotu jako požadujeme, pero chceme vrátit do zásobníku. To je situace druhá. Poslední situací je požadavek, při kterém požadujeme jinou barvu pera, než je aktuálně vybraná. To je situace číslo 3.

Rozhodovací logika využívá komparační bloky porovnání hodnoty aktuální barvy pera. Kvůli liniovému schématu je realizace poněkud náročnější. V případě třetí situace vyšleme příkaz na vyprázdnění aktuálně zvoleného pera. Opět s využitím komparačního bloku sepne tu větev, která odpovídá aktuálně zvolené barvě. Tím vytvoříme ve větvi aktuálně používaného pera druhou situaci. Pero chceme vrátit do zásobníku. Pošleme s využitím bloku *MOVE* polohu parkovací pozice pera do proměnné *X_Position*, která parametrizuje blok absolutního polohování *MC_MOVEABSOLUTE* a s jistým zpožděním funkční blok pohybu spustíme s náběžnou hranou logického signálu dané větve pomocí proměnné *MovePenX* (*X* odpovídá číslu barvy 1 až 4). Úspěšný pohyb do požadované pozice pro výměnu pera je podmíněn opět komparačním blokem kontrolující aktuální polohu osy pera. Poloha musí odpovídat zjištěné ose parkovací pozice pera v zásobníku, jinak se může poškodit uchycení pera v zásobníku při dalším postupu.



Obr. 9.3 LAD schéma FB bloku Pero, výměna pera, část 1

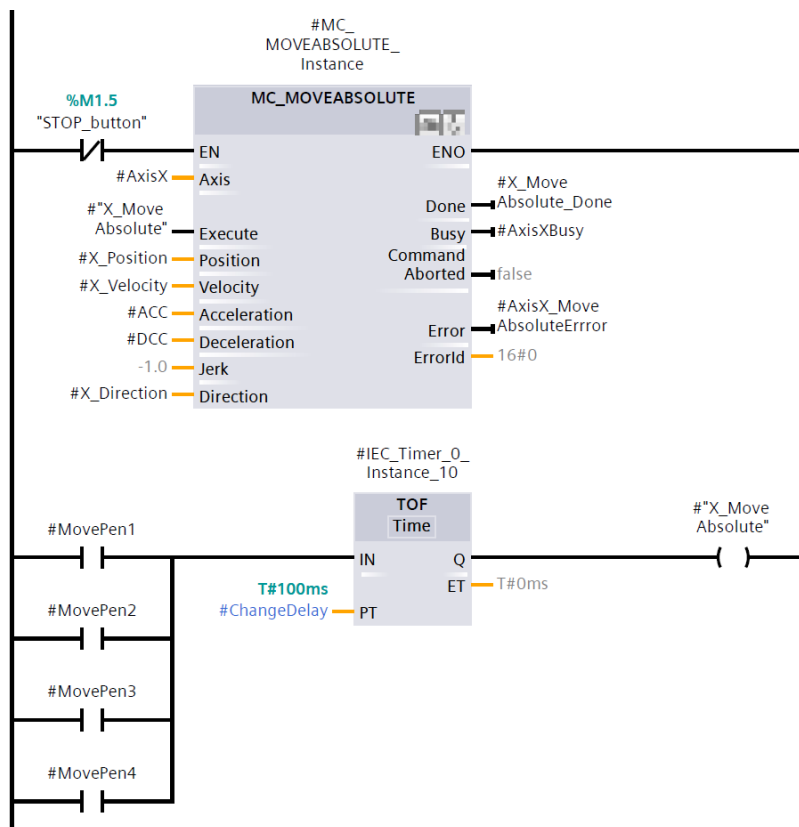
Úspěšné dokončení pohybu je signalizováno proměnnou *ChangeReadyX*. Tato proměnná zároveň podmiňuje spuštění sekvence fyzické výměny pera v zásobníku skrze navázanou proměnnou *ChangePlotterMoveX*. Sekvence se opět spouští s náběžnou hranou logického signálu. Řízení obsluhy přípravku během pohybu zásobníku bude rozebráno v následující podkapitole 9.3.



Obr. 9.4 LAD schéma FB bloku Pero, výměna pera, část 2

Jakmile je fyzická výměna pera provedena, ať už se jedná o výběr či vrácení pera, dokončíme sled příkazů resetem vnitřní PLC tagu dané barvy pera a signalizací provedení výměny pera *ChangePenX* pro vybrání pera ze zásobníku a *ChangePenX_0* pro vrácení pera do zásobníku. Rozhodování opět probíhá na základě komparačního bloku porovnání aktuální hodnoty proměnné *ActualPen*. Buď pero vracíme a proměnná má hodnotu odpovídající dané barvě pera, či opačně.

Poslední část logiky zajišťuje správné určení hodnoty aktuálně používaného pera. Návrh liniového schématu má své úskalí. V tomto bodě potřebujeme nastavit proměnnou *ActualPen* na základě posledního kroku logiky. Nastavení hodnoty provedeme pomocí bloku *MOVE*. Pokud k bloku *MOVE* nebude přiveden signál napájení (připojení *EN*), blok přestane plnit svou funkci a proměnná ztratí nastavenou hodnotu. Problém však nastává v tom, že proměnná nezůstane na poslední známé hodnotě, nýbrž je automaticky nastavena na základní hodnotu danou při definici. Pro správnou funkci proměnné *ActualPen* tedy musíme zajistit logiku, která přivede k bloku signál přesně v moment, kdy danou barvu pera vybereme, a napájecí signál udržuje v logické jedničce po celou dobu, co pero je vybrané až do chvíle, kdy barvu vrátíme do zásobníku. V ten moment musí však hodnotu začít poskytovat *MOVE* blok jiné barvy pera, jinak hrozí ztracení aktuální hodnoty. K tomu byla navržena výše zobrazená logika, a to pro každou barvu. Příklad je k dispozici v dolní polovině obr. 9.3. Řízení signálu pro bloky *MOVE* zajišťují proměnné *ChangePenX* a *ChangePenX_0*. Speciální případ představuje vyhodnocení prázdné volby, které má mírně odlišnou stavbu.



Obr. 9.5 LAD schéma FB bloku Pero, výměna pera, část 3

9.3 Realizace ovládání přípravku při výměně pera

Provedení výměny pera v zásobníku má svoje určitá specifika. Ty jsou více přiblíženy v kapitole 5 Popis přípravku.

Pohyb zásobníku je řízen na základě koncových spínačů *DorazV* a *DorazP*. Směr pohybu zásobníku k ose pera a zpět závisí na nastavení logického signálu *ZDirOUT*. Spínání motoru zásobníku obstarává logická proměnná *MotorZOUT*.

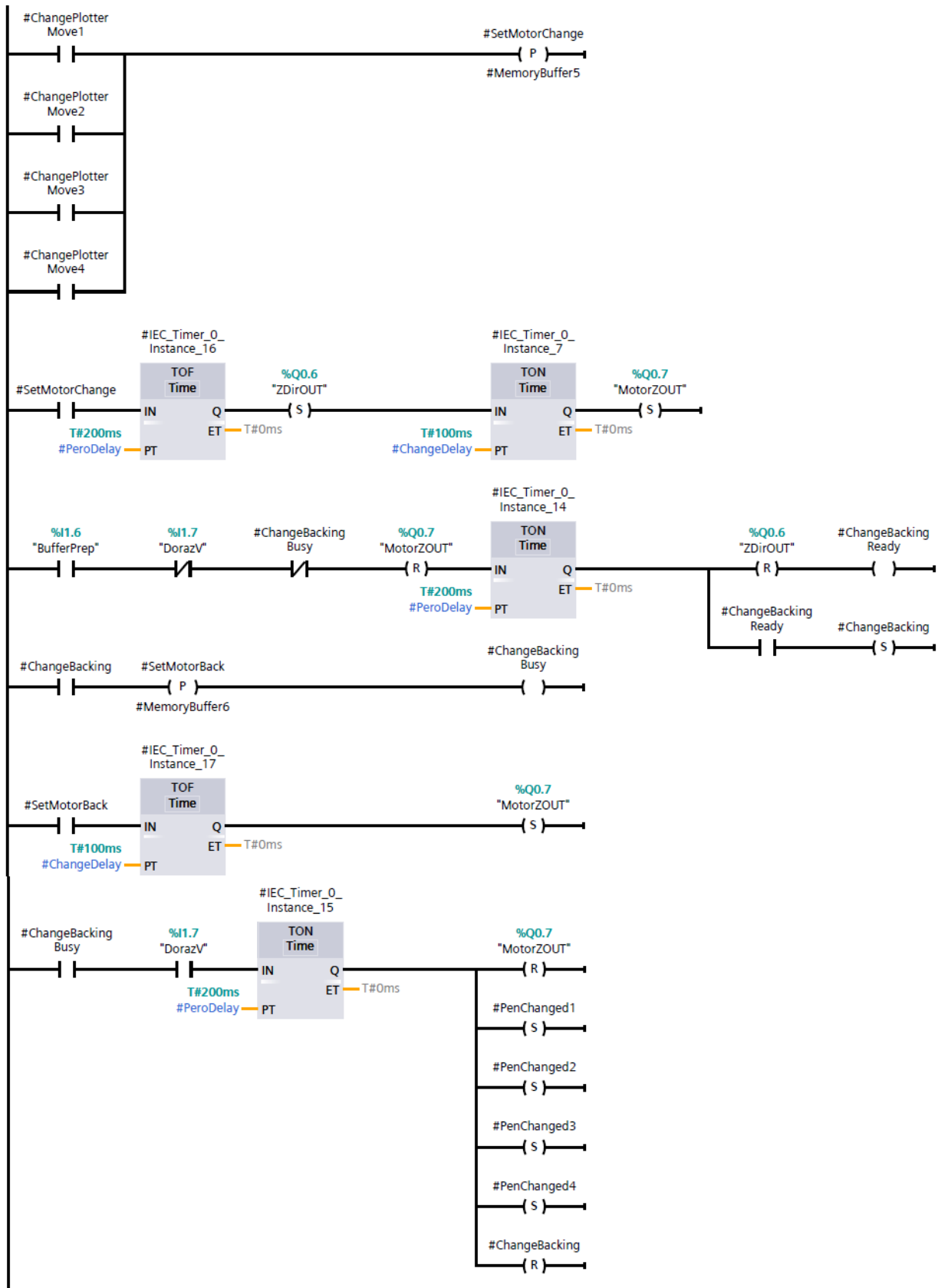
Koncové spínače signalizují polohu zásobníku. Buď se zásobník nachází v parkovací poloze, v tom případě je spínač *DorazP* v logické nule. Mezipoloha je signalizována logickou jedničkou obou koncových spínačů a poloha u osy pera představuje logickou nulu spínače *DorazV*.

Při pohybu zásobníku je nutné nejprve nastavit požadovaný směr pohybu pomocí vstupního signálu přípravku *ZDirOUT*. Logická jednička odpovídá pohybu z parkovací polohy k ose pera, logická nula pohybu opačnému. Změna směru musí proběhnout bez sepnutého motoru, jinak dochází k opalování přepínacích kontaktů v relé.

Motor je spínán pouze při pohybu zásobníku, který trvá maximálně jednotky sekund. Je proto třeba vypnout napájení motoru, jakmile se vybaví koncový spínač *DorazV* či *DorazP*, jinak dochází k chodu nakrátko, při kterém hrozí nevratné poškození motoru.

Navrhnuté logické řízení zásobníku respektuje výše popsané požadavky. Společně s tím je v logice implementováno propojení s řízením výměny pera, které posílá pokyny k obsluze zásobníku. Abychom dosáhli správné výměny pera, je nutné provést správně výše popsanou sekvenci a její ukončení ve správný okamžik signalizovat.

Návrh obsluhy zásobníku je k dispozici na následujícím obrázku 9.6. Sled příkazů koresponduje s předepsaným postupem, který byl přiložen k přípravku. Jisté části logiky jsou navrhnuté s ohledem na zajištění funkčnosti v líniovém schématu LAD PLC programování. Z tohoto důvodu zde například najdeme zpoždovací členy *TON* či *TOF*. *DorazP* je přejmenován v projektu na *BufferPrep*.



Obr. 9.6 LAD schéma FB bloku Pero, řízení zásobníku

KAPITOLA 10: REALIZACE AUTOMATICKÉHO VYKRESLOVÁNÍ

10.1 Analýza využití knihovny LKinLang

Knihovna *Kinematics Language*, zkráceně *LKinLang*, rozšiřuje nabídku funkcí knihovny *Kinematics Control*. Použití určitých nástrojů z knihovny *Kinematics Language* je podmíněno již vloženými bloky z knihovny *Kinematics Control*.

Implementace řízeného pohybu o více příkazech je uskutečnitelná pouze za předpokladu použití specializovaných knihoven *Kinematics*. Ty obsahují specifické funkční bloky přímo přizpůsobené na tzv. *CNC* řízení pohybu technologického objektu kinematiky. Jedná se o situaci, kdy technologie vykonává pohyb po požadované trajektorii s určitými parametry. Trajektorie je dána souřadnicovým způsobem pomocí G-code příkazových souborů. Jak již bylo řečeno, hlavním problémem takových aplikací je synchronizace a spolupráce jednotlivých os ve víceosém pohonu.

Pro takové aplikace jsou připraveny funkční bloky *LKinCtrl_MC_MovePath* z knihovny *Kinematics Control* schopné řízení *TO Kinematics* souřadnicově. Vzhledem však k struktuře PLC projektu, není možné pohyb řídit na základě příkazového souboru typu G-code. Takový soubor je totiž napsán v programovacím jazyku CNC strojů G-code. Přestože se jedná o velice jednoduchý jazyk, funkční blok *LKinCtrl_MC_MovePath* není kompatibilní s příkazovým souborem typu G-code. Pro funkci musíme definovat specificky strukturovaný data blok *pathData*, jehož definice je popsána v technické dokumentaci knihovny *Kinematics Control*. Popis příkazového souboru *pathData* pro řízení najdete zde [35].

Technologie *CAM* starající se o transformaci počítačového výkresu na soubor instrukcí je však neodmyslitelně spjata s programovacím jazykem G-code. Pokud však chceme z PLC sestavy částečně vytvořit *CNC* stroj, musíme se tedy postarat o přeměnu G-code souboru na speciálně strukturovaný data blok *pathData*. Aby nedošlo k omylu, nejedná se o pouhou změnu struktury souřadnic trajektorie. G-code soubor disponuje kromě souřadnic také příkazovými znaky a doplňujícími parametry, řídicí přídavné technologie zařízení. Všechna tato data musí být přeložena a vytvořena s ohledem na princip výkonu bloku *LKinCtrl_MC_MovePath*.

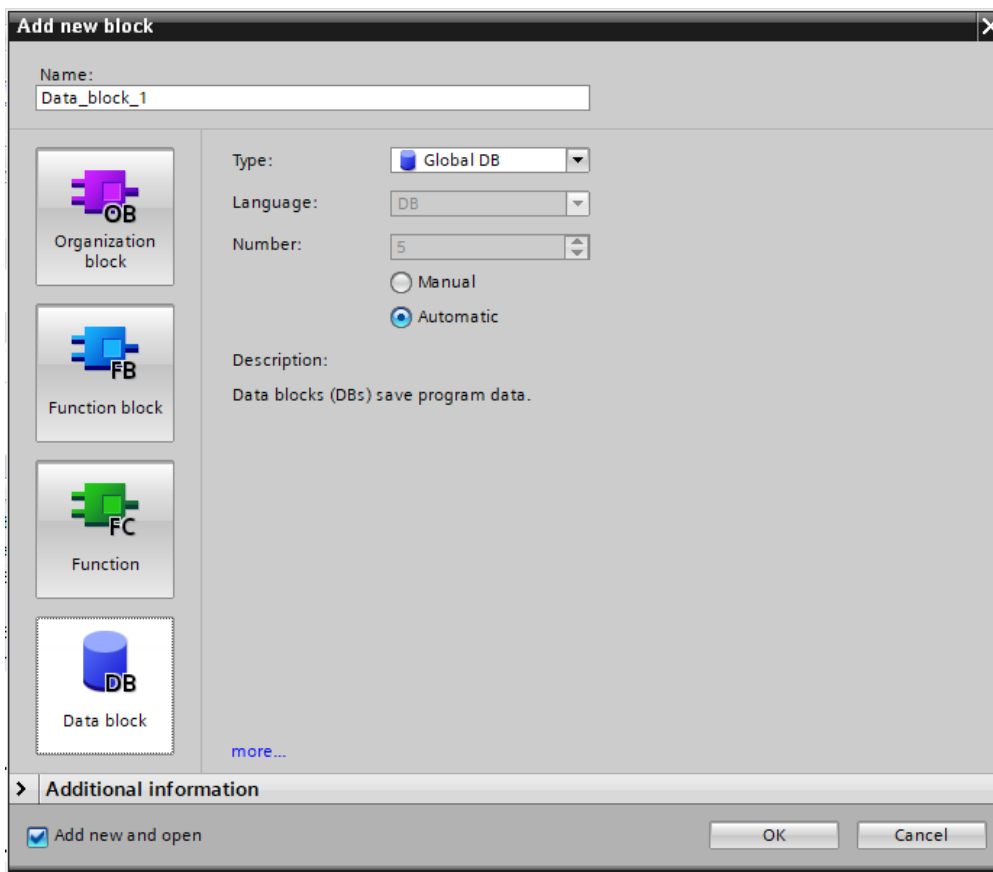
Problém s překladem G-code souboru na požadovaný data blok řeší knihovna *Kinematics Language*. Nejenom, že obsahuje funkce umožňující import a správu příkazových souborů uložených na paměťové kartě, ale disponuje také blokem *LKinLang_Parser*, jenž zajistí stěžejní krok překladu G-code souboru na speciální PLC blok *pathData*. *PathData* jsou totiž jediným formátem, který dokáže spolupracovat s řízením *TO Kinematics LKinCtrl_MC_MovePath*. Blok starající se o organizaci souborů nese název *LkinLang_FileHandler*.

10.2 Přípravné kroky pro řízení automatického vykreslování

Prvním krokem návrhu automatického vykreslování je import obou knihoven *Kinematics*. Postup byl již vysvětlen v kapitole 8.

Implementace funkčních bloků z nabídky knihovny *LKinLang* vyžaduje inicializaci odpovídajících řídicích proměnných. Vzhledem k tomu, že se jedná o vícero bloků napříč projektem, je ideální vytvořit globální datový blok, do kterého můžeme přistupovat ze všech částí programu.

K ovládání tedy vytvoříme globální data blok v záložce *Program blocks* kliknutím klasicky na *Add new block*, v nově zobrazeném okně vybereme v levé nabídce grafiku s názvem *Data block*, v nabídce *Type* vybereme *Global DB* a tlačítkem *Ok* potvrdíme přidání bloku. V projektu se řídicí blok jmenuje *Controls*.



Obr. 10.1 Vytvoření *Global DB*

Druhý blok, který je minimálně příhodné vytvořit, bude fungovat jako určité místo v programu, které zastřešuje konfigurace a datová pole nutné pro implementaci *LKinCtrl_MC_MovePath*, *LKinLang_FileHandler*, *LKinLang_Parser*, *LKinLang_Editor* a *LKinLang_ProgramManager*. Globální datový blok zabaluje všechny konfigurační proměnné výše uvedených bloků. Jedná se o speciální vnitřní PLC tagy či jejich pole. Proměnné mají specifickou strukturu danou vlastní *PLC Data Type* formátem. Všechny proměnné byly vytvořeny na základě technické dokumentace knihovny *Kinematics Language* a odpovídají doporučené definici funkčních bloků. Data blok byl vytvořen stejným postupem jako blok *Controls*, pouze s názvem *GlobalData*. Pokud pomineme konfigurační nastavení, nejdůležitější částí *GlobalData* bloku je tzv. *lineBuffer* a *pathData*. *LineBuffer* funguje jako zásobník pro načtené příkazy z vybraného souboru typu G-code. Soubor nacházející se na paměťové kartě je přečten řádek po řádku do zásobníku *lineBuffer* jako datový typ *String*, tedy pouhý text. Následně blok *LKinLang_Parser* provede překlad každého řádku na příkazový formát *TO Kinematics pathData*. *PathData* je opět proměnná speciálního datového typu, který si můžeme představit jako víceúrovňové pole obsahující všechny informace o trajektorii k vytvoření přeloženého výkresu.

Kompletní definice obou datových bloků je k nahlédnutí na obr. 10.2 a 10.3.

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	Static								
2	EnableAxes	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to enable axes with MC_POWER
3	AxesEnabled	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	status variable to see all axes enabled
4	ResetAxes	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to reset axes with MC_RESET
5	AxesError	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	status variable to see error on one axis
6	HomeAxes	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to home axes with MC_HOME
7	AxesHomed	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	status variable to see all axes homed
8	RestartMode	Array[1..5] of Bool		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable storing restart modes for MC_...
9	StartMode	Array[1..4] of Dint		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable storing start modes for MC_...
10	StopMode	Array[1..4] of Int		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable storing stop modes for MC_...
11	AutoAcknowledge	Bool	TRUE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to auto acknowledge wait fl...
12	StartPath	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to start a MovePath move
13	P_TRIG_flag_0	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	storage variable for positive trigger flag 0
14	P_TRIG_flag_1	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	storage variable for positive trigger flag 1
15	enableEditor	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to enable editor FB
16	HMIInterface	*LKinCtrl_Editor_ty...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	interface variable to HMI for editor
17	loadFile	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to execute FileHandler load file
18	saveFile	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to execute FileHandler save file
19	stopFileHandler	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to stop FileHandler
20	executeParser	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to execute Parser
21	stopParser	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to stop Parser
22	executeMovePath	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to execute MovePath
23	interruptMovePath	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to interrupt MC_MovePath
24	stopMovePath	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to stop MC_MovePath
25	resetMovePath	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to reset MC_MovePath
26	executeProgram	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to execute a program once
27	stopProgram	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to stop the program execution
28	enableHmiEditor	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to enable Editor FB
29	enableHmiProgramM...	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to enable ProgramManager FB

Obr. 10.2 Konfigurace DB Controls

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	Static								
2	fileName	String	'Cola_4.nc'	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Name of file to be parsed
3	lineBuffer	*LKinLang_typeLineBuffer		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	LineBuffer structure
4	info	*LKinLang_typeFileInfo		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	status of current loaded file
5	pointers	*LKinLang_typeLineBufferPointer		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	pathData pointer structure
6	lines	Array[LKinLang_FIRST_INDEX_LINE_BUFFER]...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	line buffer structure
7	pathData	*LKinCtrl_typePathData_advanced		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PathData structure
8	pathDataName	String[LKinCtrl_LENGTH_OF_PATHDATA_NA...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	name of PathData
9	pointers	*LKinCtrl_typeAdvPointers		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	pointers to add / retrieve cmds in PathData
10	commands	Array[1..LKinCtrl_NO_OF_PATHDATA_ELEM...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	command list
11	preProcessorConfig	*LKinLang_typePreProcessorReplacement		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Preprocessor config to search & replace
12	parameterDefaults	*LKinLang_typeGCodeDefaults		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Default values for GCode commands
13	flagModeDefaults	*LKinLang_typeFlagModeDefaults		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Default values for flags
14	interfaceVariables	*LKinLang_typeInterfaceVariables		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Interface variables
15	movePathConfiguration	*LKinCtrl_typeConfiguration		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Configuration for MovePath
16	booleanFlagArray	Array[0..LKinCtrl_NO_OF_LAST_SETFLAG] o...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Boolean flag array connected to MovePath
17	valueFlagArray	Array[0..LKinCtrl_NO_OF_LAST_VALUEFLAG]...		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Value flag array connected to MovePath
18	hmiInterfaceEditor	*LKinLang_typeHmiInterfaceEditor		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Hmi interface for editor faceplate
19	hmiInterfaceProgram...	*LKinLang_typeHmiInterfaceProgramManager		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Hmi interface for program manager faceplate

Obr. 10.3 Konfigurace DB GlobalData

10.3 Implementace LKinLang funkcí do projektu

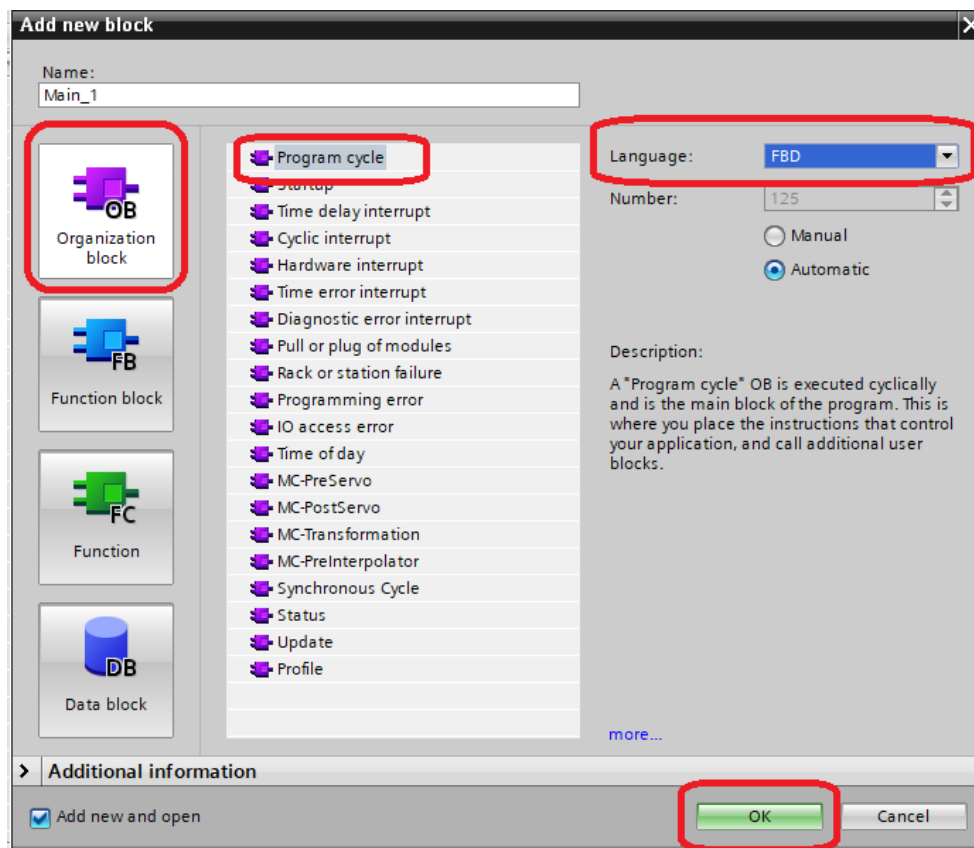
Implementace hlavních bloků knihovny *Kinematics Language* vyžaduje správnou konfiguraci a parametrizaci vložených bloků. Použijeme bloky *LKinLang_FileHandler*, *LKinLang_Parser*, *LKinLang_ProgramLineDisplay* včetně funkčních bloků obstarávajících HMI *faceplate* aplikace *LKinLang_Editor* a *LKinLang_ProgramManager*. Poslední uvedený blok hraje zásadní roli při výběru požadovaného souboru na paměťové kartě. Kromě knihovny *LKinLang* použijeme také funkční blok řízení pohybu *TO Kinematics LKinCtrl_MC_MovePath*.

K inicializaci daných bloků dle přiložené dokumentace, dostupné zde [31], použijeme připravené globální bloky *Controls* a *GlobalData*. Jejich význam a postup vytvoření bude vysvětlen v podkapitole 11.2.

Pro lepší přehlednost zvolíme pro návrh této části programu programovací jazyk FBD.

K této části programu vytvoříme nové organizační bloky programu. Tyto bloky jsou tedy vyhodnoceny v každém cyklu programu v případě, že není zavolán jiný funkční blok hlavního bloku *Main*, jejich vyhodnocení tak probíhá na pozadí programu, což zrychluje chod celého programu.

Vytvoříme tedy dva organizační bloky pro HMI aplikace a pro PLC aplikace. V záložce *Program blocks* klikneme na *Add new block*, v levé nabídce vybereme *Organization block*, v hlavní nabídce vybereme *Program cycle* typ a blok pojmenujeme a potvrdíme výběr tlačítkem *Ok*. Bloky v našem projektu byly pojmenovány *HmiMain* pro bloky zajišťující HMI aplikace a *CNCMain* pro bloky zajišťující proces přehrání, překladu a vykreslování. Názornější postup je k dispozici na obr. 10.4.



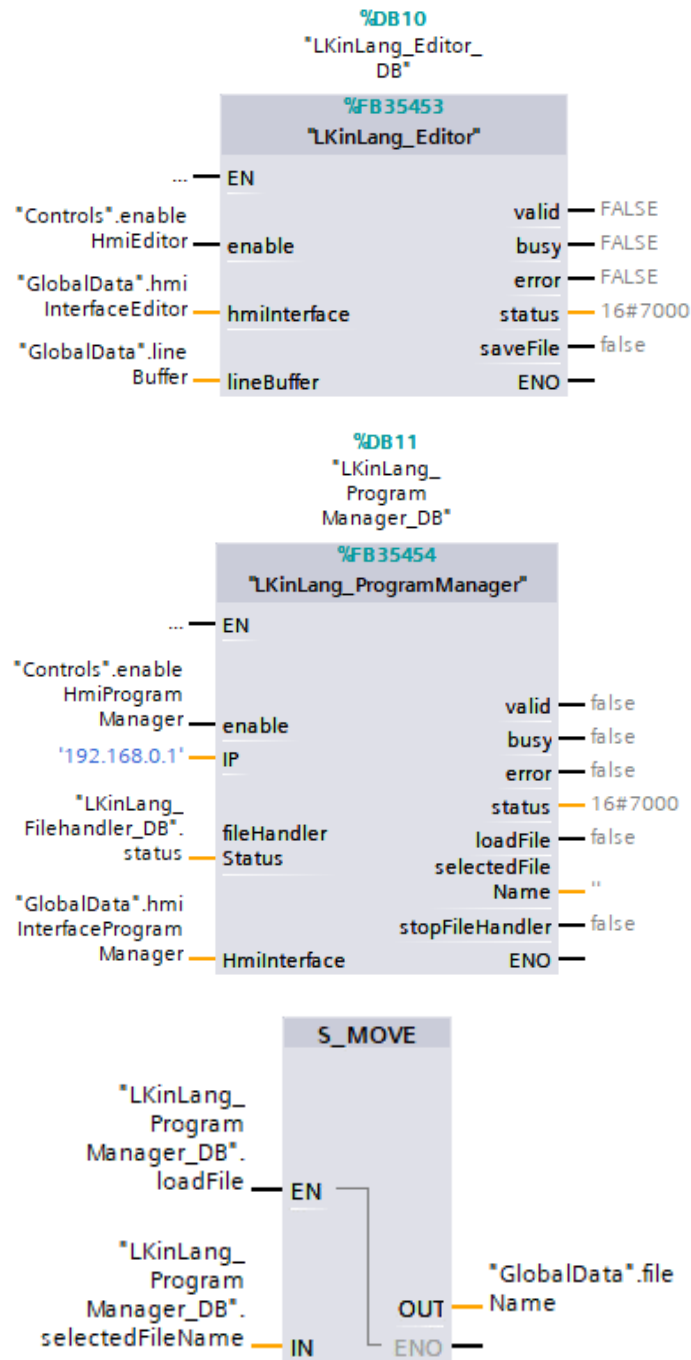
Obr. 10.4 Založení OB CNCMain a HmiMain

10.3.1 Návrh propojení HMI aplikací a PLC programu

První částí implementace bloků knihovny *Kinematics Language* do projektu je zprovoznění bloků *LKinLang_Editor* a *LKinLang_ProgramManager* obstarávající provoz tzv. *faceplate* HMI aplikací.

Bloky vložíme ze záložek *LKinLang_Editor_Blocks* a *LKinLang_ProgramManager_Blocks* do nově vytvořeného bloku *HmiMain*, každý do své sekce (*Network*). Přesun je možné udělat pomocí *Drag & drop* instrukce.

Kromě již zmíněných bloků vložíme do programu také funkci *S_MOVE*, resp. *Move character string*. Blok plní funkci přesunu názvu ve formátu *String* z parametru HMI aplikace *LKinLang_ProgramManager* do proměnné jména výkresu v datovém bloku *GlobalData*. Použití HMI aplikace bude předmětem návrhu vizualizace později. Adresování proměnných datových bloků probíhá přes tečkovou notaci. Přesun názvu pomocí bloku *S_MOVE* je podmíněn příkazem *loadFile* zajišťující přehrání souboru z paměťové karty PLC zařízení do zásobníku *lineBuffer*.



Obr. 10.5 Návrh organizačního bloku OB HmiMain

Jak bude později vysvětleno, aplikace *ProgramManager* umožňuje vzdálený přístup k adresáři paměťové karty prostřednictvím lokální sítě, a proto je při její konfiguraci nutné nastavit správnou IP adresu řídicího PLC zařízení. Zde je IP adresa PLC zařízení nastavena na *192.168.0.1*, jak odpovídá konfiguraci hardwarové sestavy. Konfigurace řídicích signálů obou bloků je k dispozici na obr. 10.5.

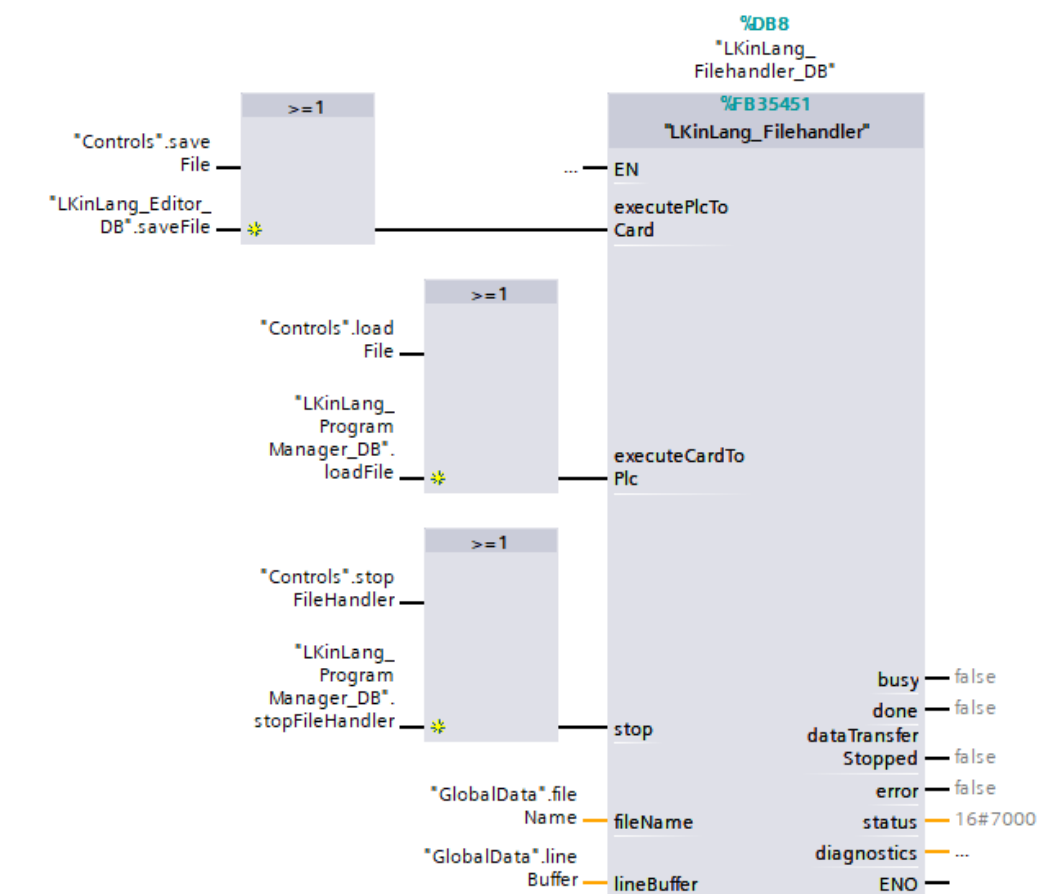
10.3.2 Implementace automatického vykreslování do řízení TO

Druhá část implementace bloků pro automatické vykreslování představuje import již zmíněných bloků *LKinLang_FileHandler*, *LKinLang_Parser*, *LKinLang_ProgramLineDisplay* a

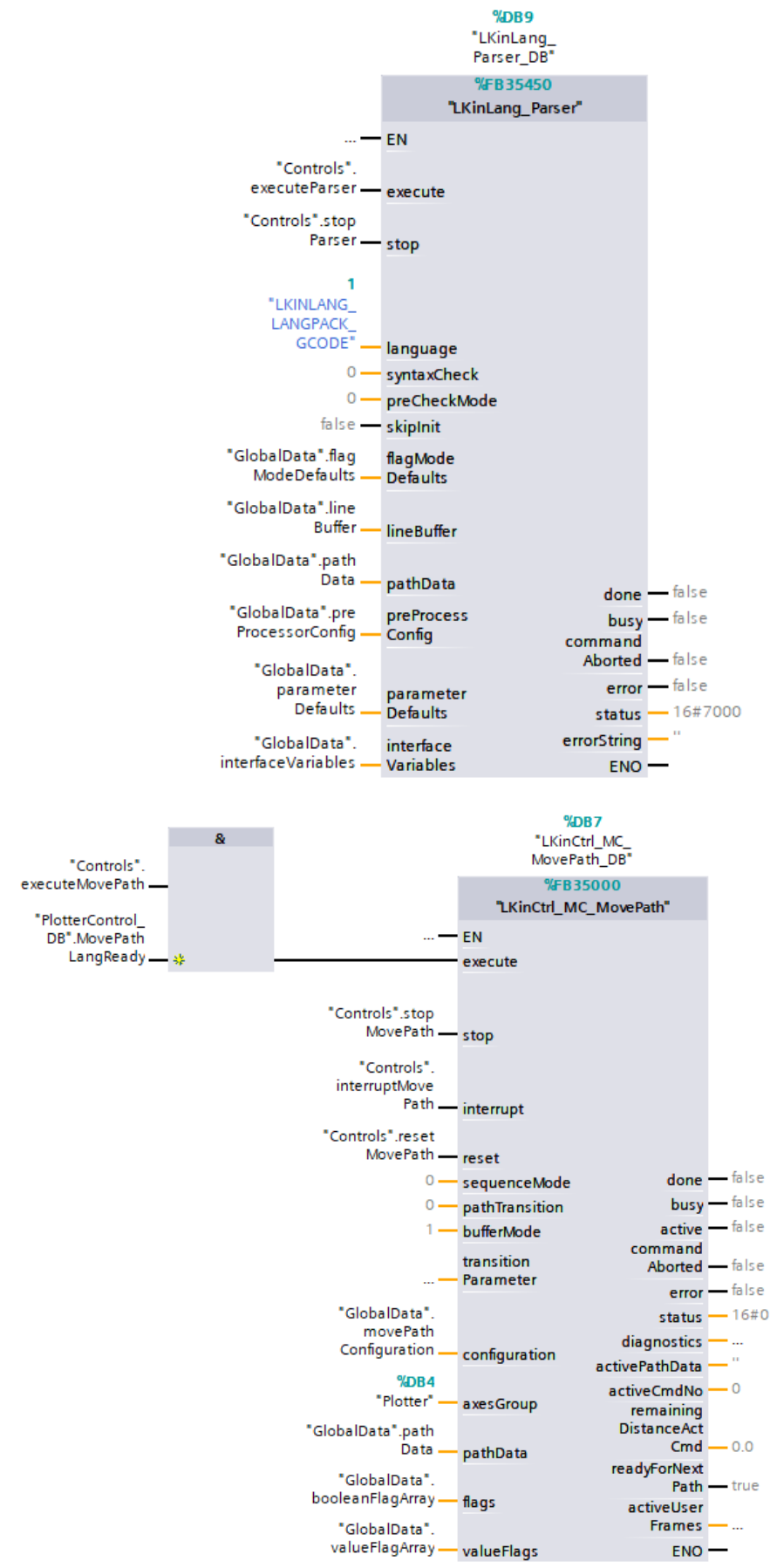
LKinCtrl_MC_MovePath. Bloky opět najdeme v záložce *Program blocks* našeho projektu a pomocí instrukce *Drag & drop* je přetáhneme do námi vytvořeného organizačního bloku CNCMain.

Zde je konfigurace bloků o poznání rozsáhlejší než v případě bloků HMI aplikací. Jedná se však stále o parametrizaci a nastavení všech bloků dle přiložené dokumentace knihoven. Zpozorněme při nastavení bloku *LKinLang_Parser*, kde je nutné zvolit režim 1 pro základní překlad z obecného jazyku G-code. Dále je třeba mít na paměti, že spuštění importu souboru, *LKinLang_FileHandler* -> *executePlcToCard/executeCardToPlc/stop* je možné jak z ovládacího bloku *Controls*, tak i pomocí tzv. *faceplate* HMI aplikace na ovládacím panelu. Výkon bloku *MovePath* zajišťující řízení pohybu vykreslování je podmíněn zpracováním sekvence pro řízení přiložení pera *MovePathLangReady* z bloku *PlotterControl*. Realizaci řízení přiložení pera bude věnován prostor později v podkapitole 11.5.

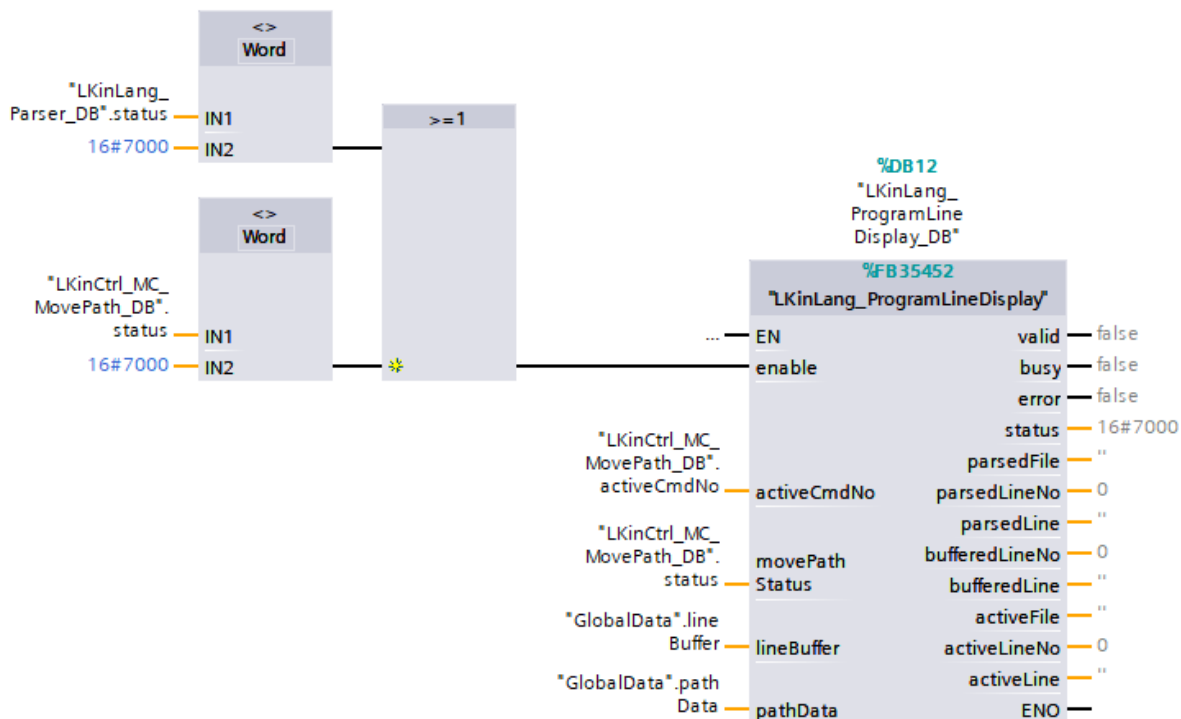
Na kompletní návrh celého organizačního bloku CNC Main je možné nahlédnout na obr. 10.6, 10.7 a 10.8.



Obr. 10.6 Návrh organizačního bloku OB CNCMain, část 1



Obr. 10.7 Návrh organizačního bloku OB CNCMain, část 2

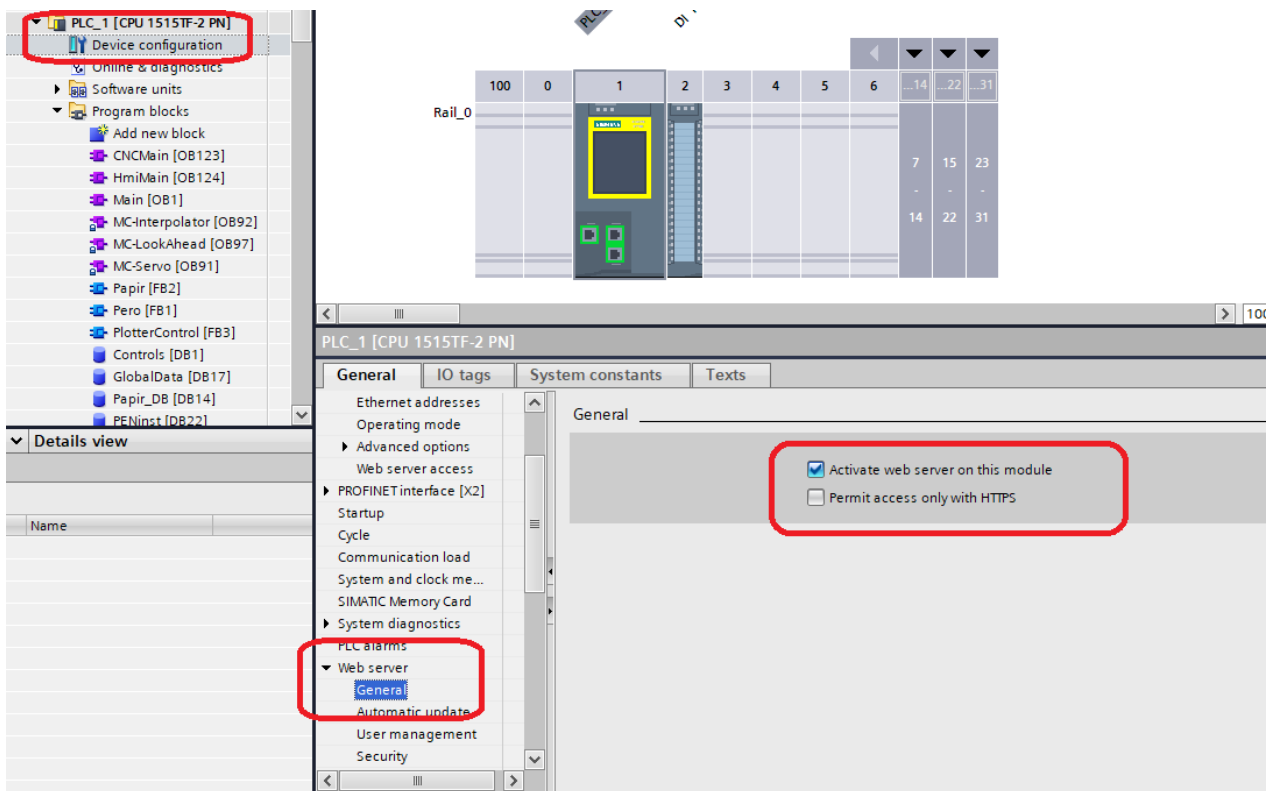


Obr. 10.8 Návrh organizačního bloku OB CNCMain, část 3

10.4 Konfigurace File Transfer tool

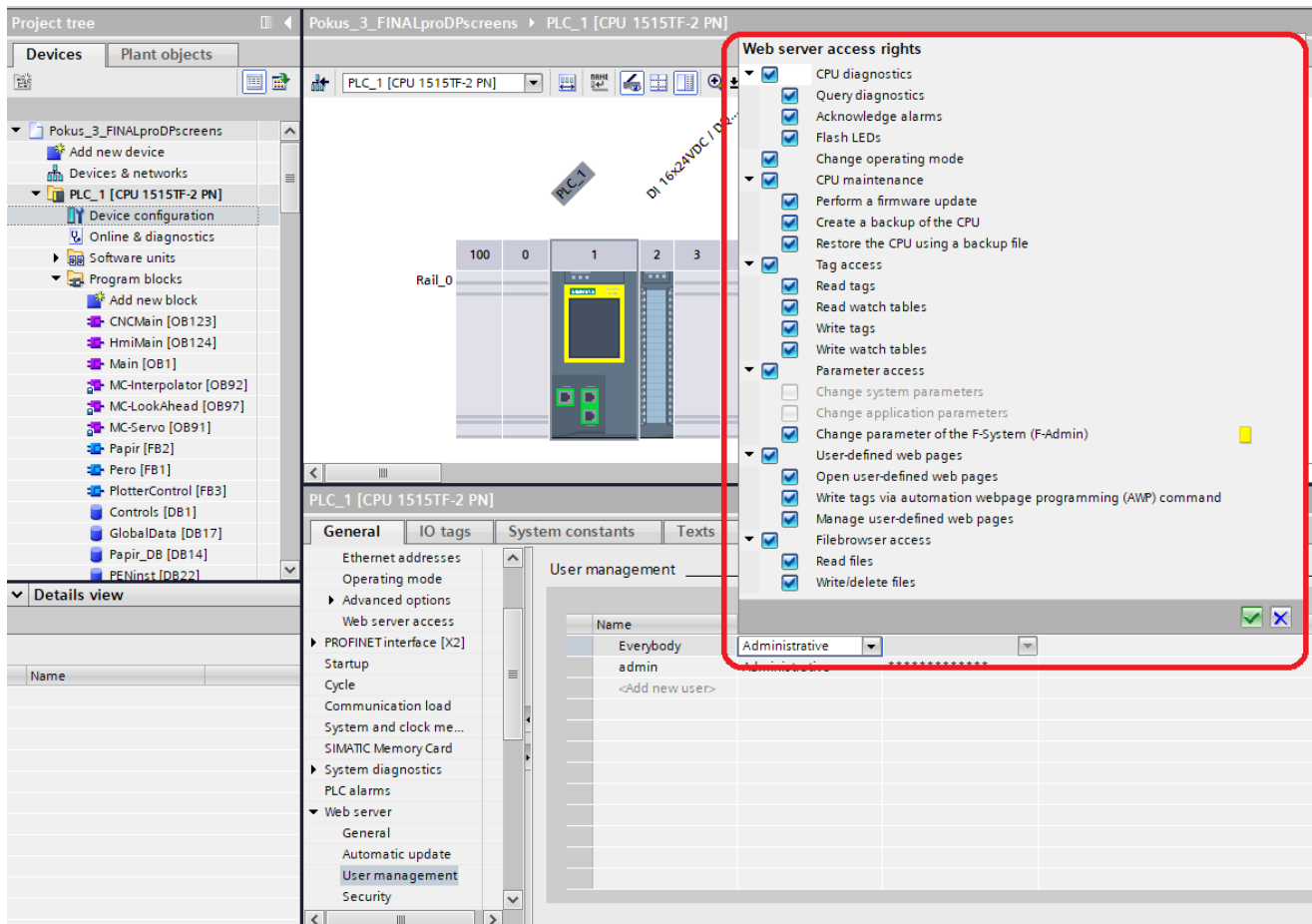
Počítačová aplikace *File Transfer tool* poskytuje vzdálený přístup na paměťovou kartu PLC zařízení připojeného do lokální sítě LAN s využitím služby *Web server*. Abychom datovou cestu mohli navázat, musíme nejdříve nakonfigurovat spojení v konfiguraci PLC zařízení v projektu. Konfigurace kompletně vychází z postupu uvedeného v dokumentaci ke knihovně *Kinematics Language* pouze s menšími rozdíly, které byly odhaleny při vývoji projektu.

Postup začíná otevřením vlastností PLC zařízení v projektu ve vývojovém prostředí TIA Portal. To je možné pravým kliknutím na záložku PLC zařízení a otevřením poslední možnosti z nabídky *Properties*. Následně se objeví požadované okno nastavení. Zde přejdeme do záložky *Web server*, sekce *General* a potvrdíme volbu *Activate web server on this module*, více na obr. 10.9.



Obr. 10.9 Konfigurace funkce *Web server*

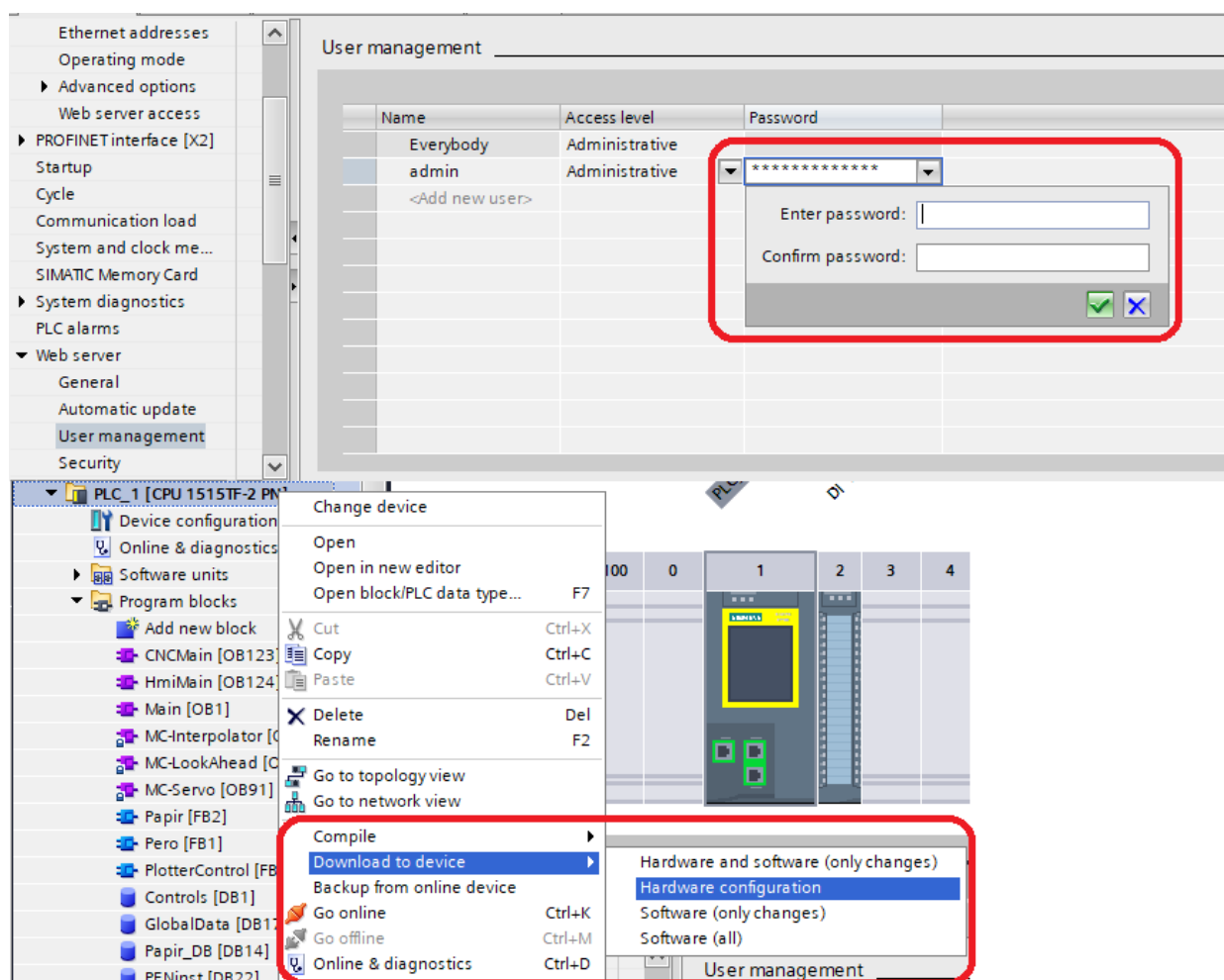
Pokračujeme sekci *User management*. Zde doporučuji automatickému přístupu *Everybody* udělit administrativní práva, a to rozkliknutím sloupce *Access level* a zaškrtnutím všech možností. Bez administrativního přístupu *Everybody* uživatele nebude fungovat přístup HMI aplikace k *Web serveru* PLC zařízení bez dodatečné realizace funkce *User administration* v rámci vizualizace HMI panelu. Dále vytvoříme dalšího uživatele s libovolným jménem, doporučuji však ponechat název *admin*. Zde opět zaškrtneme všechny možnosti a poskytneme tak uživateli administrátorská práva s tím rozdílem, že definujeme heslo pro jeho přihlášení. Nastavení hesla se vygeneruje dvojklikem na buňku hesla daného uživatele.



Obr. 10.10 Konfigurace přístupových profilů funkce *Web server*

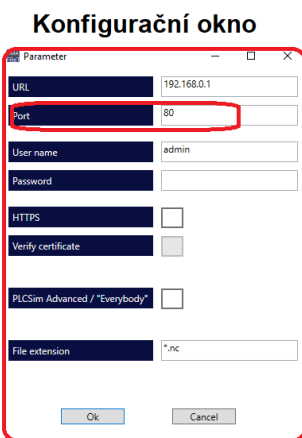
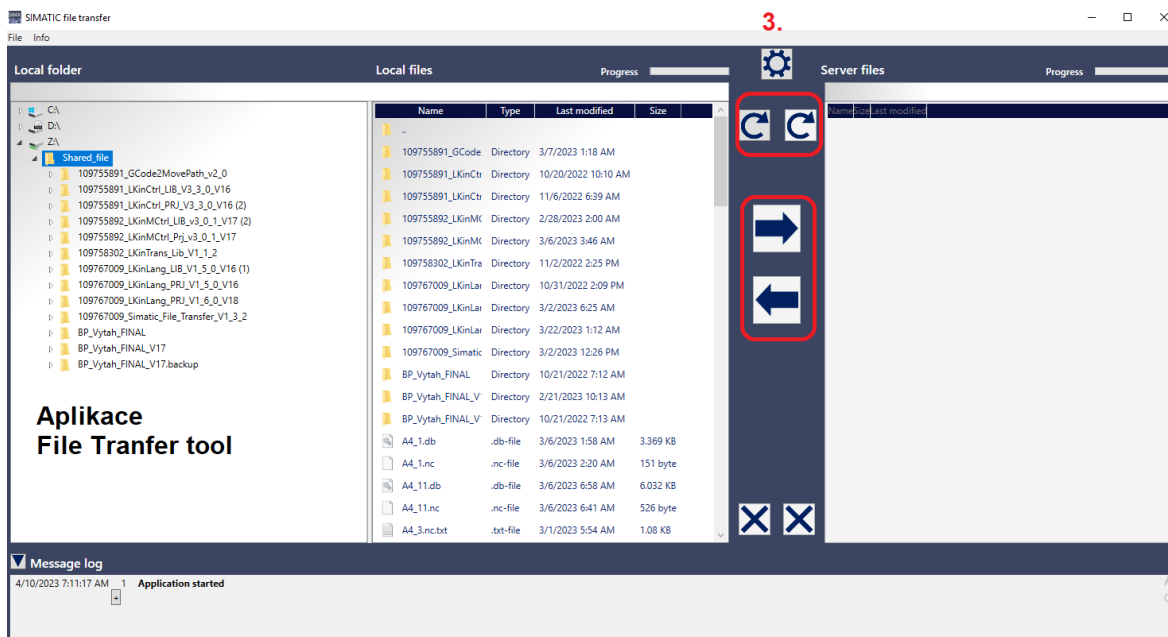
V případě použití šifrovaného přenosu dat pomocí *HTTPS* protokolu, je nutné vložit certifikát týkající se PLC funkce *Web server* do operačního systému počítače se spuštěnou aplikací *File Transfer tool*. V našem projektu využijeme běžný protokol *http*, u kterého však tento krok není potřeba.

Posledním krokem konfigurace v prostředí TIA Portal je přehrání nového nastavení, resp. *Hardware configuration*, do reálného PLC zařízení. To provedeme pomocí pravého kliknutí na záložku projektového PLC zařízení a pod možností *Download to device* vybereme *Hardware configuration*.



Obr. 10.11 Přehrání konfigurace do PLC zařízení

Následuje nastavení aplikace v osobním počítači. Zde je třeba mít PLC zařízení připojeno k osobnímu počítači a spustit aplikaci. Aplikaci je třeba stáhnout z webové stránky technické podpory Siemensu, více zde [31]. Po spuštění aplikace se objeví přihlašovací okno spojení. Zde zkontrolujeme volbu IP adresy PLC zařízení (192.168.0.1), port spojení (80 pro HTTP) a volbu komunikačního protokolu. V případě naší konfigurace musí být volba šifrovaného přenosu dat odškrtnuta. Správná konfigurace je k dispozici na obr. 10.12. Po kontrole naší volbu potvrdíme a aplikace se automaticky připojí k PLC zařízení. V pravé části je posléze zobrazen adresář paměťové karty PLC zařízení a v levé cílový adresář osobního počítače. Přesun souborů je možný použitím směrových tlačítek s grafikou šipek. Adresáře obnovíme pomocí nástrojových tlačítek nad šipkami. Ke konfiguraci se zpětně dostaneme tlačítkem č. 3 nad obnovením.



Obr. 10.12 Připojení PC aplikace k Web serveru

10.5 Logika pro kreslení pera

Všechny programové části automatického vykreslování ve smyslu zpracování potřebné trajektorie a její vykonání byly připraveny a vysvětleny. Zbývá pouze navázat spolupráci přípravku a výkonu funkčního bloku pohybu *LKinCtrl_MC_MovePath*. Jedná se o synchronizované řízení přiložení pera během pohybu v rámci vykreslování. Logické řízení přiložení pera je součástí přípravku a je řízen vstupním logickým signálem ploteru. Pro tento signál si tedy vytvoříme výstupní PLC tag PÉN. Je nutné zvolit správný vstup v DI/DQ bloku PLC zařízení. Definici PLC tagu provedeme v záložce *PLC tags* v projektu PLC zařízení v sekci *Default tag table*. Definice odpovídá definici běžné proměnné.

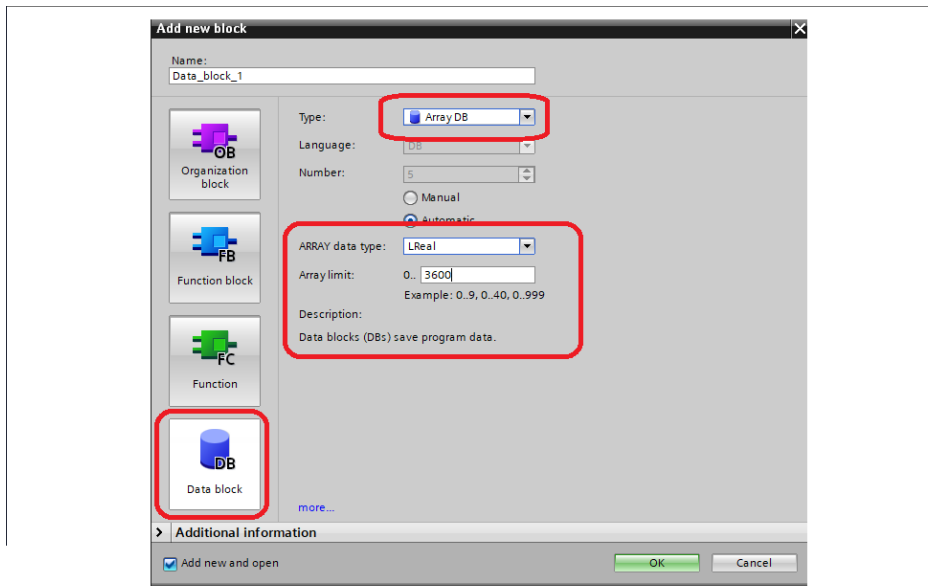
Default tag table								
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervision
1	MemoryAxisOn_Y	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	MoveRelative	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	PositioningAxis_1_Actor_Interf...	*PD_TEL105_IN*	%I256.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	PositioningAxis_1_papir_Actor...	*PD_TEL105_IN*	%I276.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	PÉN	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Obr. 10.13 Definice PLC tagu PÉN

Běžné CAM technologie generující G-code soubor používají v případě 2D modelu třetí rozměr, resp. souřadnici třetí osy pro nástroj, kterým dochází k tvorbě daného modelu. Soubor totiž obsahuje kromě trajektorie popisující tvorbu modelu (zde výkresu) ve dvojrozměrném prostoru, také instrukce pro pohyb mezi tahy pera. To znamená, že je potřeba vytvořit program, který zajistí synchronizované načítání souřadnice třetího rozměru ve zdrojovém bloku *pathData* a dle jeho hodnoty vybaví funkci přiložení pera pomocí spínání logického signálu přípravku.

Řešení logického řízení přiložení pera bylo značnou výzvou, neboť PLC program pracuje s přesně definovanou logikou programu. Kompilátor proto třeba nedovolí přistupovat k poli proměnných pomocí jiných proměnných, které mohou nabývat hodnot mimo rozsah definice daného pole, přestože logická sekvence programu přiřazuje proměnné pouze hodnoty v potřebném rozsahu. S ohledem na tyto komplikace bylo vytvořeno řízení přiložení pera schopné synchronizace s výkonem funkčního bloku *LKinCtrl_MC_MovePath* pro řízení pohonu následujícím způsobem.

Přeložený soubor výkresu typu G-code obsahuje ve struktuře formátu *pathData* informace o každém kroku technologického objektu. Maximální počet instrukcí na jeden soubor *pathData* je při konfiguraci pevně nastaven. Využijeme toho a vytvoříme přidružené pole typu *Array of LReal* o stejné velikosti jako je struktura *pathData*. Záměrně zmiňuji slovo struktura, jelikož reprezentace *pathData* v projektu není typ *Array of*, tedy pole, ale *LKinCtrl_typePathData_advanced*, speciální PLC datový typ. Pole typu *Array of LReal* založíme podobným způsobem jako datové bloky *Controls* a *GlobalData*, pouze s rozdílem výběru typu *Array DB* a volby typu *LReal*. V našem projektu se blok pole jmenuje PENinst. Maximální velikost pole bude činit 3600. Jedná se o nastavený limit počtu instrukcí pro *pathData* strukturu.

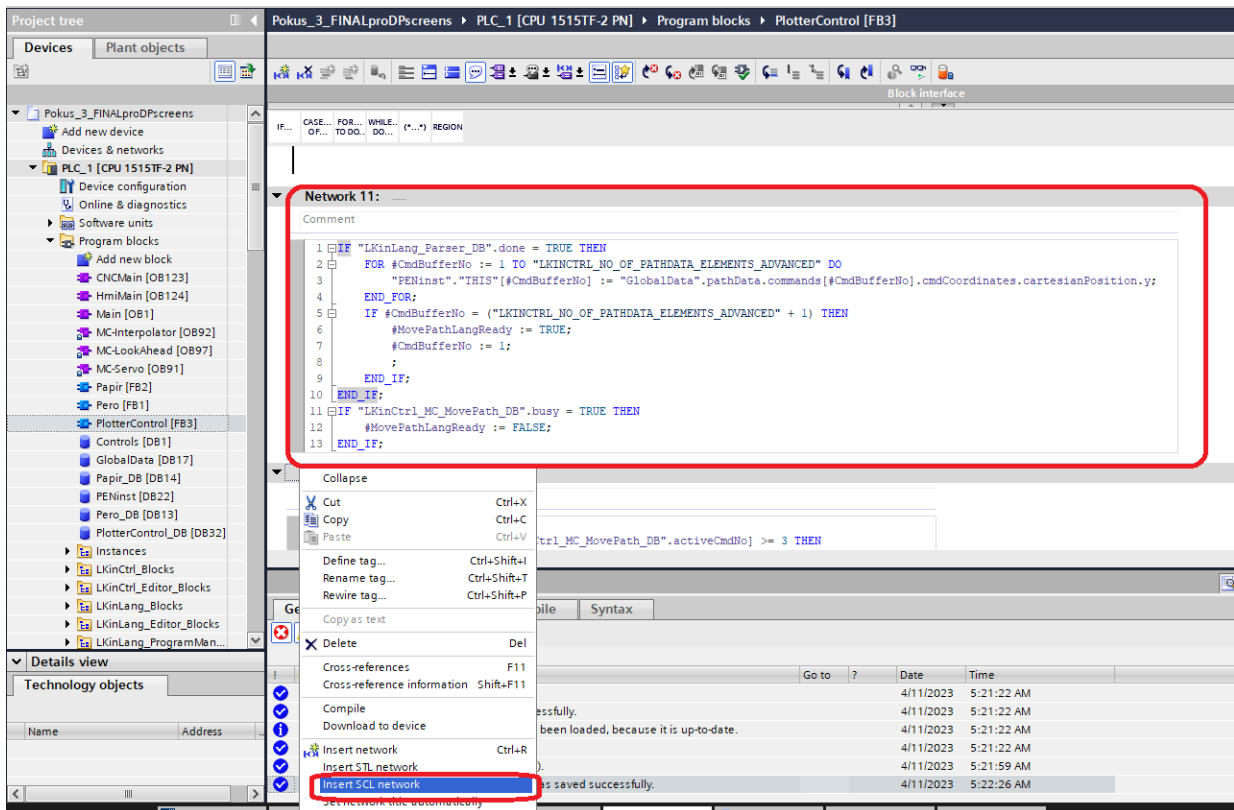


	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	PEIninst	Array[0..4500] ...								
2	PEIninst[0]	LReal	0.0							
3	PEIninst[1]	LReal	0.0							
4	PEIninst[2]	LReal	0.0							
5	PEIninst[3]	LReal	0.0							
6	PEIninst[4]	LReal	0.0							
7	PEIninst[5]	LReal	0.0							
8	PEIninst[6]	LReal	0.0							
9	PEIninst[7]	LReal	0.0							
10	PEIninst[8]	LReal	0.0							
11	PEIninst[9]	LReal	0.0							
12	PEIninst[10]	LReal	0.0							
13	PEIninst[11]	LReal	0.0							
14	PEIninst[12]	LReal	0.0							
15	PEIninst[13]	LReal	0.0							
16	PEIninst[14]	LReal	0.0							
17	PEIninst[15]	LReal	0.0							
18	PEIninst[16]	LReal	0.0							
19	PEIninst[17]	LReal	0.0							
20	PEIninst[18]	LReal	0.0							
21	PEIninst[19]	LReal	0.0							
22	PEIninst[20]	LReal	0.0							
23	PEIninst[21]	LReal	0.0							
24	PEIninst[22]	LReal	0.0							
25	PEIninst[23]	LReal	0.0							
26	PEIninst[24]	LReal	0.0							

Obr. 10.14 Založení PENinst data bloku

Nyní navrheme instrukce pro naplnění pole na základě přeloženého souboru typu *pathData*. Vzhledem k tomu, že budeme procházet velké množství dat a budeme jejich obsah kopírovat do nově vzniklého pole, využijeme pro návrh SCL textové programování.

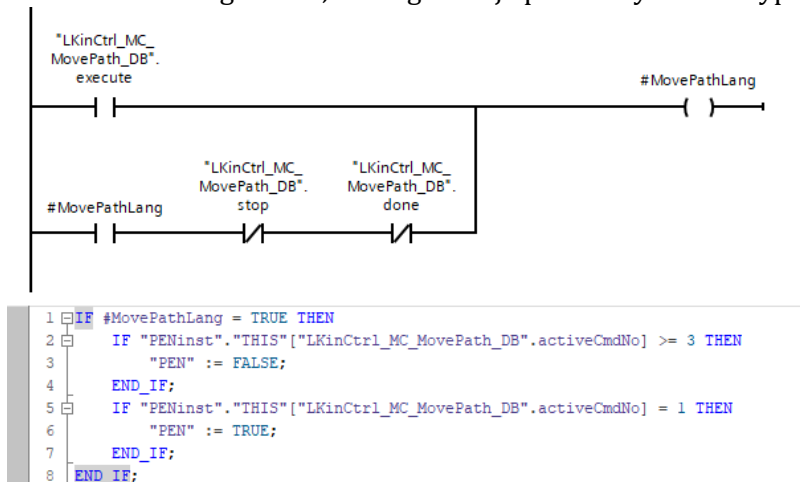
Ve funkčním bloku PlotterControl při pravém kliknutí na sekci *Network* vybereme volbu *Insert SCL network*. Do nově vzniklé sekce implementujeme program zahrnující instrukci FOR cyklus. FOR cyklus bude řídit proměnná, která bude fungovat jako čítač. Pomocí této proměnné vždy projdeme strukturu *pathData* od první po poslední instrukci. Čítání ve FOR cyklu bude nastaveno od jedné do hodnoty maximálního počtu instrukcí. Definici je možné udělat parametricky v závislosti na konfigurační proměnné *LKINCTRL_NO_OF_PATHDATA_ELEMENTS_ADVANCED*.



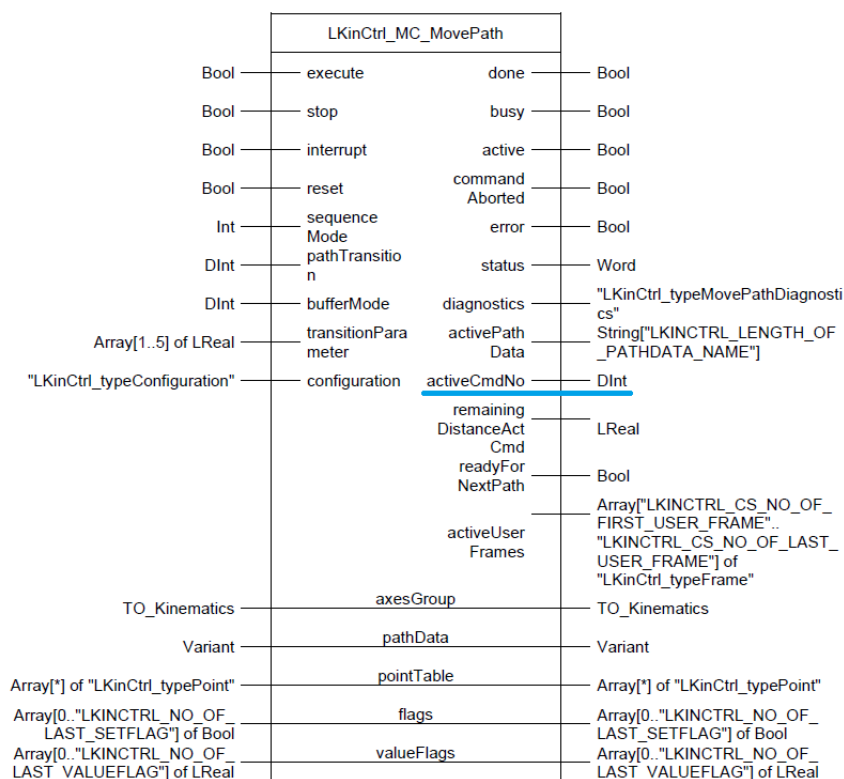
Obr. 10.15 Generování instrukcí pro PLC tag PEN

Uvnitř FOR cyklu budeme pomocí proměnné čítání procházet Y souřadnici a její hodnotu kopírovat do nově vzniklého pole, které taky řídíme čítačem. Na závěr po doběhnutí FOR cyklus nastavíme čítač opět na jedničku a oznámíme, že instrukce pro řízení pera jsou připraveny. To je realizováno pomocí proměnné *MovePathLangReady*. Tato proměnná podmiňuje spuštění bloku *MovePath*.

Zbývá nám návrh synchronizovaného procházení pole během zavolaného bloku *MovePath*. Řešením je použití proměnné *activeCmdNo*. Jedná se o parametr bloku *LKinCtrl_MC_MovePath*, který ukazuje hodnotu aktuálně prováděné instrukce struktury *pathData*. K parametru nejdříve vytvoříme samozřejmě odpovídající proměnnou. Touto proměnnou pak ukazujeme na aktuální pozici v bloku *PENinst* a jeho aktuální hodnotu testujeme pomocí příkazu IF. Hodnota větší než tři odpovídá zvednutému peru a hodnota jedna odpovídá přiloženému kreslicímu peru. Tento postup však záleží na nastavení technologie CAM, která generuje příkazový soubor typu G-code.



Obr. 10.16 Řízení přiložení pera



activeCmdNo	OUT	DInt	number of actual PathData set
-------------	-----	------	-------------------------------

Obr. 10.17 Struktura FB LKinCtrl_MC_MovePath

Poslední částí je ošetření doby, kdy řídíme přiložení pera automaticky pomocí vyhodnocení pole PENinst. Jedná se o to, že logický signál výstupního PLC tagu musí nabývat určité hodnoty, a proto musíme tzv. vypnout část programu, která řídí přiložení pera na základě bloku *MovePath* v době, kdy vykreslování není aktivní. Ošetření přepíná řízení přiložení pera v případě spuštěného bloku *MovePath* a rozpínacím kontaktem řízení předává zpět pomocí signalizace provedeného pohybu či zastavení vykonávání bloku *MovePath*. Realizace je k dispozici na obr. 10.16.

Všimněme si, že téměř celá logika byla vytvořena pomocí SCL textového programování. Je to dáno tím, že řešení vyžadovalo operace jednodušeji realizovatelné ve vyšším programovacím jazyku, než je blokové FBD či liniové LAD schéma.

10.6 Ladění překladu G-code souboru

Praktické testování projektu na aplikační sestavě přineslo zjištění, že překladač G-code souborů *LKinLang_Parser* vyžaduje určitou úpravu stávajících příkazových souborů. Nutno říct, že se jedná o úpravu souborů vytvořených konkrétní CAM technologií programu *Inkscape* v rámci rozšíření *Gcodetools*.

Vzhledem k tomu, že programovací jazyk instrukcí G-code nemá pevně stanovenou terminologii a pravidla pro jeho použití, dochází ke komplikacím při snaze přeložit obecný soubor v rámci PLC zařízení.

Grafický editor *Inkscape* se osvědčil jako velice efektivní nástroj pro tvorbu výkresů v požadovaném formátu papíru A4 nebo A3. Pro správné vykreslení je totiž nutné správně synchronizovat kartézskou soustavu souřadnic výkresu včetně polohy počátku systému a jeho orientace. Běžně dostupné softwarové nástroje umožňující použití technologie CAM bohužel neumožňují personalizaci překladu modelu na příkazový soubor G-code, což vede k dalším úpravám souboru tak, aby soubor by přeložitelný v PLC zařízení.

Jedná se o úpravu systémového příkazu GXX z dvouciferného čísla příkazu na pouze jednociferný příkaz GX. Tento problém jde ovšem vyřešit bez potřeby dodatečné úpravy. Funkční blok *LKinLang_Parser* umožňuje nakonfigurovat tzv. *preProcessorConfig*, kde je možné provést záměnu určitého znaku za jiný. Vzhledem k tomu, že PLC překladač dokáže pracovat pouze s určitými příkazy CNC strojů, stačí nastavit záměnu slova G00 za G0 a slova G01 za G1. Nastavení je možné provést v bloku *GlobalData*, více na obr. 10.18.

GlobalData										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	fileName	String	'Cola_4.nc'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Name of file to be parsed
3	lineBuffer	*LKINLang_typeLin...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		LineBuffer structure
4	pathData	*LKINCtrl_typePath...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		PathData structure
5	preProcessorConfig	*LKINLang_typePre...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Preprocessor config to search & replace
6	replacements	Array[1..*LKINLANG...		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pre-processor replacement structure
7	replacements[1]	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pre-processor replacement structure
8	searchTerm	String[*LKINLANG_...	'G01'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		term to be searched
9	replaceTerm	String[*LKINLANG_...	'G0'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		term to replace search term
10	replacements[2]	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pre-processor replacement structure
11	searchTerm	String[*LKINLANG_...	'G00'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		term to be searched
12	replaceTerm	String[*LKINLANG_...	'G0'	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		term to replace search term
13	replacements[3]	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pre-processor replacement structure
14	replacements[4]	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pre-processor replacement structure
15	replacements[5]	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pre-processor replacement structure
16	replacements[6]	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		pre-processor replacement structure

Obr. 10.18 Nastavení funkce preProcesorConfig bloku LKinLang_Parser

Další nutnou úpravu představují dodatečné komentáře v souboru vygenerované technologií *Gcodetools* programu *Inkscape*. Technologie CAM doplňuje příkazový soubor o komentáře popisující jednotlivé instrukce. Bohužel programovací jazyk G-code nemá pevně stanovaný formát komentáře jako například programovací jazyk C a při snaze přeložit takový soubor dochází k chybě překladu, dokud komentáře nejsou odstraněny. Možnost odstranit komentáře přímo v programu *Inkscape* bohužel nenajdeme. Každý soubor se tak před vykreslením musí upravit ručně v textovém editoru. Nejlépe se osvědčil volně dostupný editor Notepad++.

Další problém překladu představuje konfigurace technologického objektu kinematiky. Při nastavení dvojrozměrného kinematického objektu s kartézskou soustavou souřadnic z blíže nezjištěného důvodu prostředí TIA Portal preferuje osu Z místo běžně používané osy Y. Bohužel vývojové prostředí TIA Portal neumožňuje další nastavení. Je tedy nutné před vložením soubor typu G-code upravit ve smyslu záměny osy Y za osu Z a naopak, neboť pomocí třetí, doplňkové, souřadnice řídíme přikládání pera v rámci kreslení.

Posledním krokem je vložení příkazu M30 na poslední řádek souboru. Příkaz M30 označuje konec souboru. Opět se jedná o příkaz, který není obecně nutné používat. Překladač *LKinLang_Parser* ho ovšem vyžaduje.

Všechny tyto úpravy jsou výsledkem ladění přeložitelnosti souboru G-code vygenerovaného programem *Inkscape* s CAM doplňkem *Gcodetools*. Pokud bychom použili jiný CAM software, či dokonce i jiný CAD program, výsledný soubor může mít zcela odlišnou podobu. Od toho se dále odvíjí potřebné úpravy souboru do přeložitelné podoby v PLC zařízení. Na následujícím obrázku můžeme porovnat soubor libovolně zvoleného výkresu před úpravou a po úpravě. Jedná se o přeložitelný soubor výkresu.

```

1 G00 S1; endstops
2 G00 E0; no extrusion
3 G01 S1; endstops
4 G01 E0; no extrusion
5 G21; millimeters
6 G91 G0 F300.0 Z20.000; pen park !!Zsafe
7 G90; absolute
8 G28 X; home
9 G28 Y; home
10 G28 Z; home
11 G00 F300.0 Z21.000; pen park !!Zpark
12 G00 F2400.0 Y0.000; !!Ybottom
13 G00 F2400.0 X0.000; !!Xleft
14 G00 F2400.0 X142.284 Y93.927; move !!Xleft+142.284 Ybottom+93.927
15 G00 F300.0 Z1.000; pen down !!Zwork
16 G01 F2100.0 X145.774 Y97.417; draw !!Xleft+145.774 Ybottom+97.417
17 G00 F300.0 Z3.000; pen up !!Zup
18 G00 F2400.0 X145.488 Y96.771; move !!Xleft+145.488 Ybottom+96.771
19 G00 F300.0 Z1.000; pen down !!Zwork
20 G01 F2100.0 X137.514 Y104.744; draw !!Xleft+137.514 Ybottom+104.744
21 G00 F300.0 Z3.000; pen up !!Zup
22 G00 F2400.0 X136.694 Y103.517; move !!Xleft+136.694 Ybottom+103.517
23 G00 F300.0 Z1.000; pen down !!Zwork
24 G01 F2100.0 X144.791 Y95.420; draw !!Xleft+144.791 Ybottom+95.420
25 G00 F300.0 Z3.000; pen up !!Zup
26 G00 F2400.0 X143.727 Y94.437; move !!Xleft+143.727 Ybottom+94.437
27 G00 F300.0 Z1.000; pen down !!Zwork
28 G01 F2100.0 X134.807 Y103.357; draw !!Xleft+134.807 Ybottom+103.357
29 G00 F300.0 Z3.000; pen up !!Zup
30 G00 F2400.0 X135.365 Y103.387; move !!Xleft+135.365 Ybottom+103.387
31 G00 F300.0 Z1.000; pen down !!Zwork
32 G01 F2100.0 X128.679 Y96.701; draw !!Xleft+128.679 Ybottom+96.701
33 G00 F300.0 Z3.000; pen up !!Zup
34 G00 F2400.0 X129.712 Y96.168; move !!Xleft+129.712 Ybottom+96.168
35 G00 F300.0 Z1.000; pen down !!Zwork
36 G01 F2100.0 X125.251 Y100.629; draw !!Xleft+125.251 Ybottom+100.629
37 G00 F300.0 Z3.000; pen up !!Zup
38 G00 F2400.0 X124.307 Y100.518; move !!Xleft+124.307 Ybottom+100.518
39 G00 F300.0 Z1.000; pen down !!Zwork
40 G01 F2100.0 X123.275 Y99.486; draw !!Xleft+123.275 Ybottom+99.486
41 G00 F300.0 Z3.000; pen up !!Zup
42 G00 F2400.0 X123.360 Y98.425; move !!Xleft+123.360 Ybottom+98.425
43 G00 F300.0 Z1.000; pen down !!Zwork
44 G01 F2100.0 X126.153 Y95.632; draw !!Xleft+126.153 Ybottom+95.632
45 G00 F300.0 Z3.000; pen up !!Zup
46 G00 F2400.0 X126.219 Y94.241; move !!Xleft+126.219 Ybottom+94.241

```

Výkres před úpravou



```

1 G1 F300.0 Y21.000
2 G1 F2400.0 Z0.000
3 G1 F2400.0 X0.000
4 G1 F2400.0 X65.373 Z116.683
5 G1 F300.0 Y1.000
6 G1 F2100.0 X115.028 Z105.901
7 G1 F2100.0 X119.306 Z104.800
8 G1 F2100.0 X128.099 Z103.217
9 G1 F2100.0 X132.011 Z101.645
10 G1 F2100.0 X131.607 Z99.862
11 G1 F2100.0 X130.387 Z98.791
12 G1 F2100.0 X128.553 Z98.298
13 G1 F2100.0 X126.309 Z98.253
14 G1 F2100.0 X117.275 Z99.897
15 G1 F2100.0 X112.752 Z101.005
16 G1 F2100.0 X107.936 Z101.764
17 G1 F2100.0 X103.464 Z102.957
18 G1 F2100.0 X101.557 Z103.963
19 G1 F2100.0 X99.976 Z105.371
20 G1 F2100.0 X72.988 Z111.114
21 G1 F2100.0 X70.608 Z111.984
22 G1 F2100.0 X68.485 Z113.147
23 G1 F2100.0 X66.710 Z114.686
24 G1 F2100.0 X65.373 Z116.683
25 G1 F2100.0 X66.350 Z116.471
26 G1 F300.0 Y3.000
27 G1 F2400.0 X69.801 Z112.426
28 G1 F300.0 Y1.000
29 G1 F2100.0 X65.591 Z116.636
30 G1 F300.0 Y3.000
31 G1 F2400.0 X65.955 Z115.814
32 G1 F300.0 Y1.000
33 G1 F2100.0 X66.566 Z116.424
34 G1 F2100.0 X68.248 Z116.059
35 G1 F2100.0 X66.799 Z114.610
36 G1 F2100.0 X67.895 Z113.659
37 G1 F2100.0 X69.930 Z115.694
38 G1 F2100.0 X71.612 Z115.328
39 G1 F2100.0 X69.096 Z112.812
40 G1 F2100.0 X70.419 Z112.087
41 G1 F2100.0 X73.294 Z114.963
42 G1 F2100.0 X74.976 Z114.598
43 G1 F2100.0 X71.893 Z111.514
44 G1 F300.0 Y3.000
45 G1 F2400.0 X73.443 Z111.017
46 G1 F300.0 Y1.000

```

Výkres po úpravě

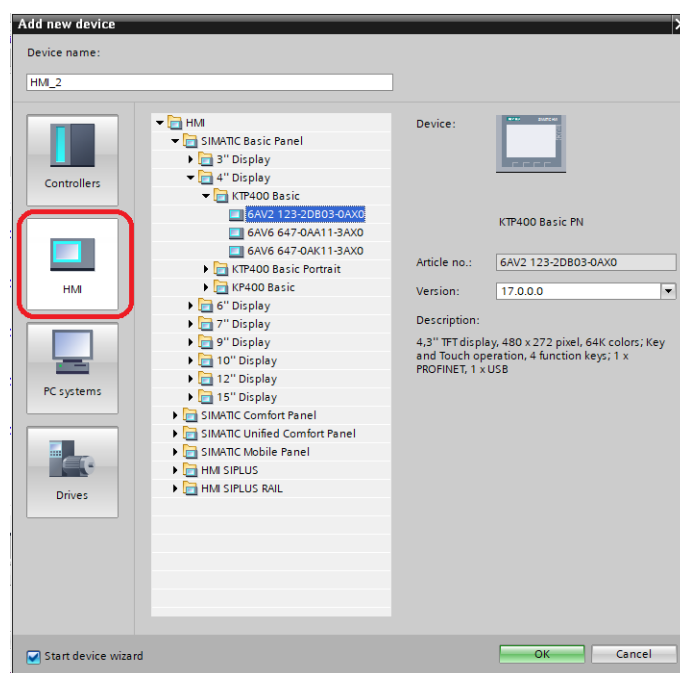
Obr. 10.19 Úprava příkazového souboru typu G-code

KAPITOLA 11: VIZUALIZACE ŘÍZENÍ XY PLOTERU

11.1 Konfigurace HMI panelu KTP400 Basic a KP900 Comfort

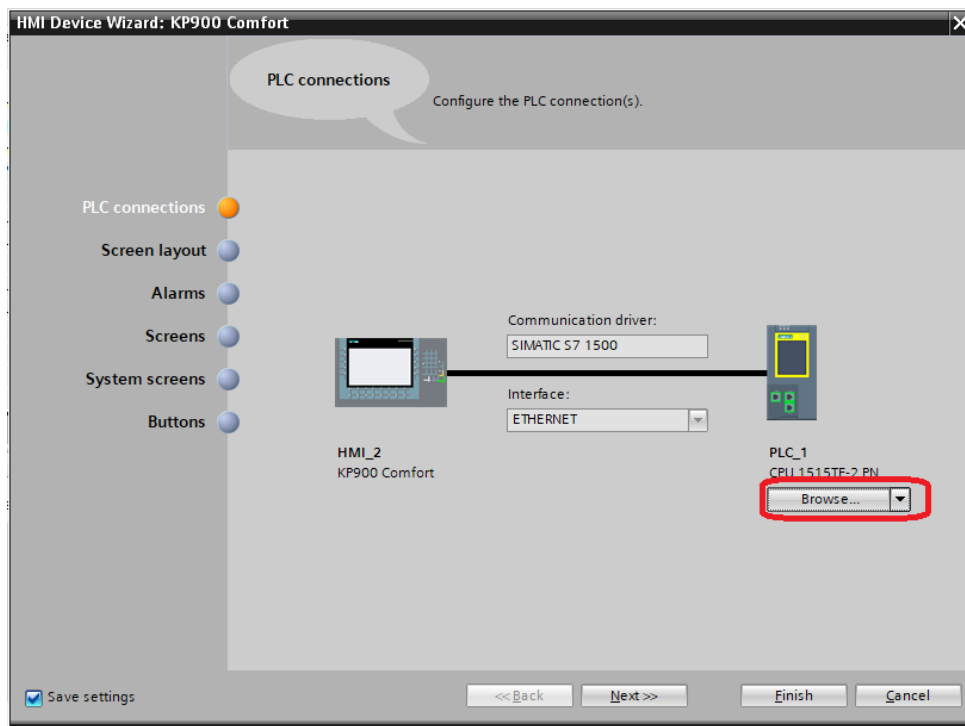
Konfigurace ovládacích HMI panelů ve vývojovém prostředí TIA Portal se v zásadě neliší od konfigurace ostatních, již popsaných, zařízení. Opět je nutné zařízení vložit do projektu a zajistit nastavení komunikačního rozhraní v čele přiřazení odpovídající IP adresy a PROFINET jména.

Pokud již máme jako v našem případě založený projekt, do kterého chceme pouze dodatečně naimportovat zařízení, klikneme na první volbu nabídky sekce *Project tree*, tedy *Add new device*. V nově zobrazeném okně vybereme v levé části HMI zařízení a v seznamu zařízení vyhledáme námi použité zařízení. Pokud jde o naši soustavu, v prvním případě se jedná o SIMATIC Basic Panel, 4" Display, KTP 400 Basic. Konkrétní výběr je k dispozici na obr. 11.1. V případě druhého panelu vybereme SIMATIC Comfort Panel, 9" Display a KP900 Comfort. Výběr potvrdíme dole tlačítkem *Ok*.



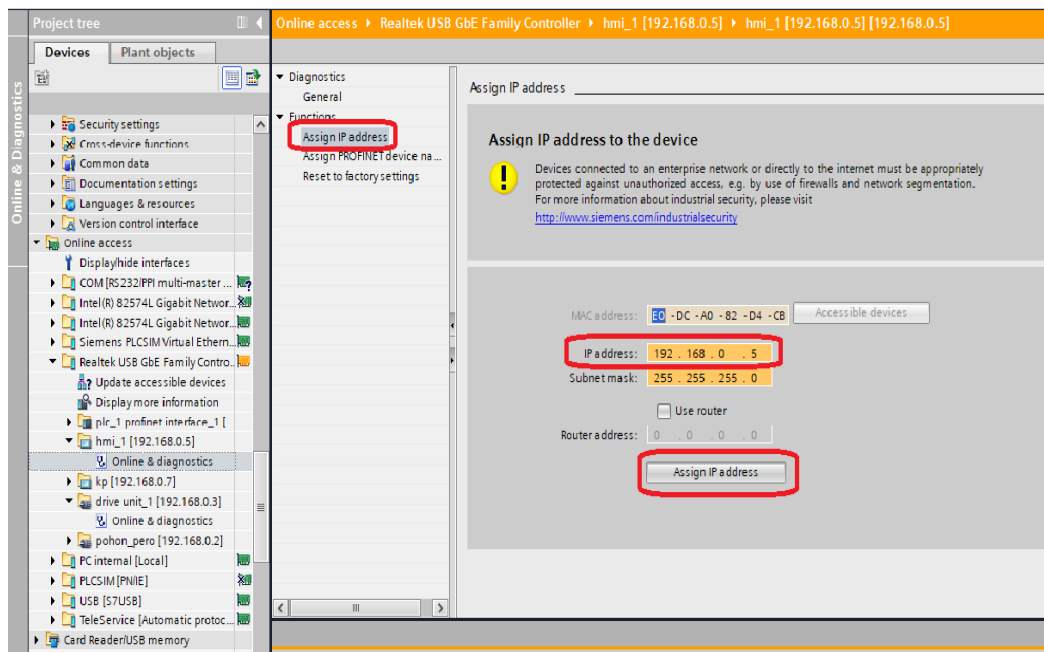
Obr. 11.1 Vložení HMI panelu do projektu

V dalším kroku se v prostředí objeví průvodce zrychlenou konfigurací zařízení. Zde nastavíme pouze sekci *PLC connections*. V seznamu *Select PLC*, zvýrazněno na obr. 11.2, vybereme projektové zařízení PLC a postup ukončíme kliknutím na dolní tlačítko *Finish*. Postup opakujeme pro druhý panel.



Obr. 11.2 Nastavení komunikace HMI zařízení

V dalším kroku je nutné zkontrolovat a popřípadě nastavit IP adresu zařízení v soustavě tak, aby odpovídala přiřazené IP adrese v projektovém zařízení. To provedeme skrze nabídku *Online access*. Zde vybereme námi používaný ovladač, najdeme požadované HMI zařízení a v nabídce zvolíme funkci *Online & diagnostics*. Přiřazení IP adresy je k dispozici v sekci *Functions* -> *Assign IP address*. Postup je stejný jako v případě hardwarové konfigurace ostatních zařízení.



Obr. 11.3 Přiřazení IP adresy HMI panelu

11.2 Použité principy programování HMI panelů

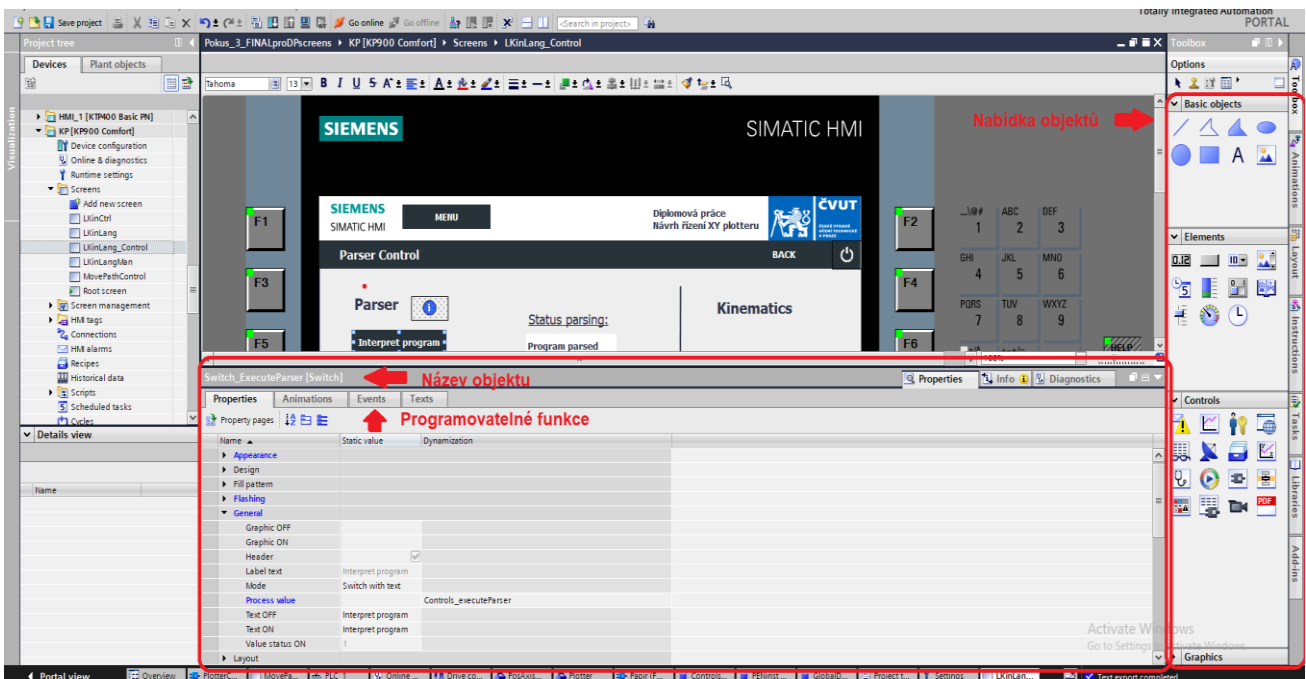
Při návrhu vizualizace na HMI panelech máme jasně stanovené možnosti úpravy grafických bloků z nabídky *Toolbox*. Nabídka se nachází na pravé straně okna vývojového prostředí TIA Portal a obsahuje kromě základních grafických objektů, viz. *Basic objects* na obr. 11.4, také jednoduché elementy, z nichž jmenujme kupříkladu tlačítko, přepínač, *I/O field* nebo ručkový ukazatel. Kromě těchto objektů v nabídce najdeme také poslední sekci *Controls*. Zde jsou pokročilejší HMI aplikace, díky kterým si můžeme například zobrazit diagnostické hlášky, zobrazit trend, resp. průběh určité proměnné v čase, nebo dokonce otevřít soubor ve formátu PDF.

Téměř každý programovací objekt vizualizace umožňuje podobnou formu personalizace. Při jeho označení se v dolní části hlavního okna objeví v záložce *Properties* možnosti jeho nastavení.

11.2.1 Properties

Jako první volbu v levé části dolního okna najdeme nabídku *Properties*. Ta se stará jak o grafické nastavení objektu, tak i programové. Uvedme příklad pro objekt tlačítka *Button*. V sekci *Properties* můžeme nastavit jak barvu pozadí, tak i formát jeho okrajů, efekt odrazu světla, ale také text popisu tlačítka. V případě textu máme sekci *Text format*, kde můžeme nastavit styl, velikost či třeba zvýraznění nebo podtržení textu. Veškeré grafické nastavení objektu je však pouhá část nabídky *Properties*, dále tu narazíme na sekce *General* nebo *Security*. *General* sekce umožňuje obecné propojení tlačítka s HMI tagem, *Security* naopak kupříkladu omezuje použití tlačítka pro různé uživatelské profily.

Příklad naprogramovaného tlačítka z vizualizace projektu je k nahlédnutí na obr. 11.4.



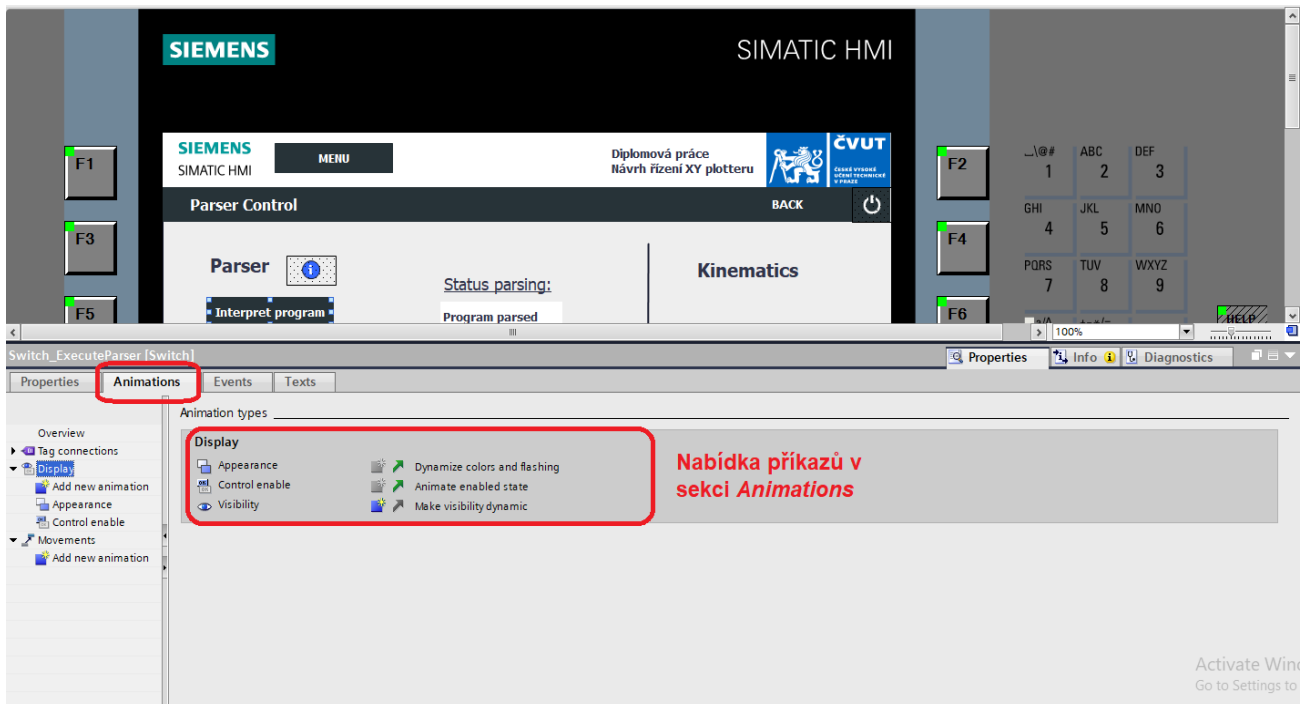
Obr. 11.4 Provedení vizualizace, sekce *Properties*

11.2.2 Animations

Záložka *Animations* slouží k vygenerování odezvy či reakce objektu na určitou signalizační proměnnou projektu. Představme si to tak, že logická proměnná ovlivňuje například viditelnost či zbarvení určitého bloku. Jedná se tak o grafickou odezvu logického stavu uvnitř PLC programu.

Na výběr máme řízení viditelnosti objektu v sekci *Visibility* či změny vzhledu v sekci *Appearance*. Dále můžeme naprogramovat řízený pohyb objektu na obrazovce panelu na základě určité proměnné, a to ve všech čtyřech základních směrech.

V našem projektu jsou použité pouze nastavení viditelnosti a zbarvení objektů.

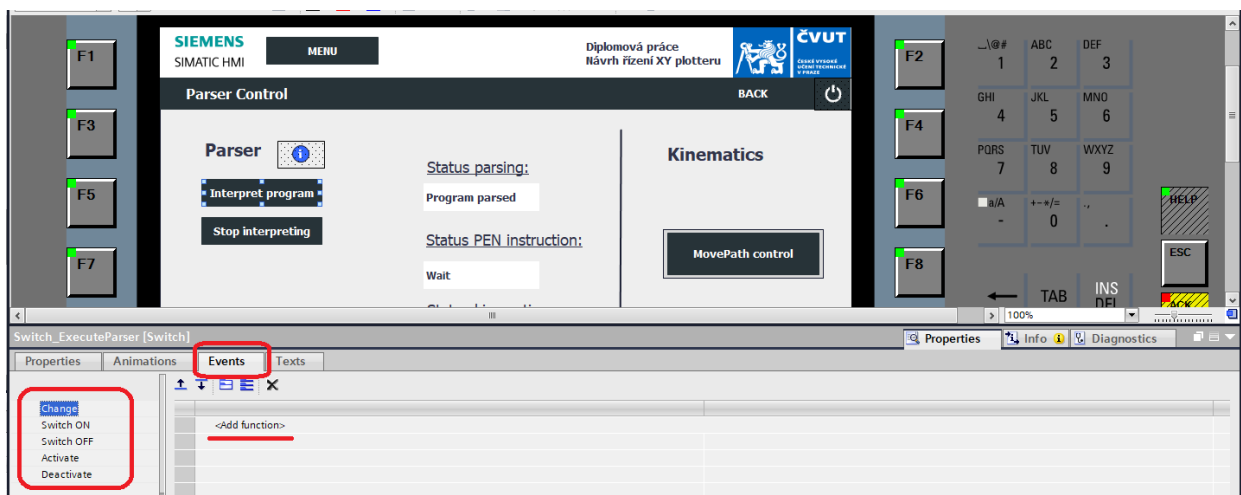


Obr. 11.5 Provedení vizualizace, sekce *Animations*

11.2.3 Events

Záložku *Events* můžeme označit za nejdůležitější sekci nastavení programového objektu panelu. Zde se jednotlivé možnosti návrhu liší dle typu objektu. To znamená, že pro objekt tlačítka můžeme kupříkladu naprogramovat určitý příkaz při stlačení, uvolnění a zmáčknutí tlačítka. Pokud bychom zvolili například přepínač, budeme moci příkaz spojit se sepnutým, vypnutým spínačem, nebo jeho přepnutím. Vždy tedy záleží na typu programovaného objektu.

Při požadavku na vytvoření příkazu při určitém úkonu s objektem (například stlačení tlačítka) máme na výběr z omezených příkazů, které můžeme vytvořit. Vždy se příkaz pojí s určitým HMI tagem.



Obr. 11.6 Provedení vizualizace, sekce *Events*

Organizace HMI tagů probíhá poté v samostatné záložce projektového panelu *HMI tags*. Jedná se o analogii správy PLC tagů v projektovém PLC zařízení.

11.3 Návrh vizualizace panelu KTP400 Basic

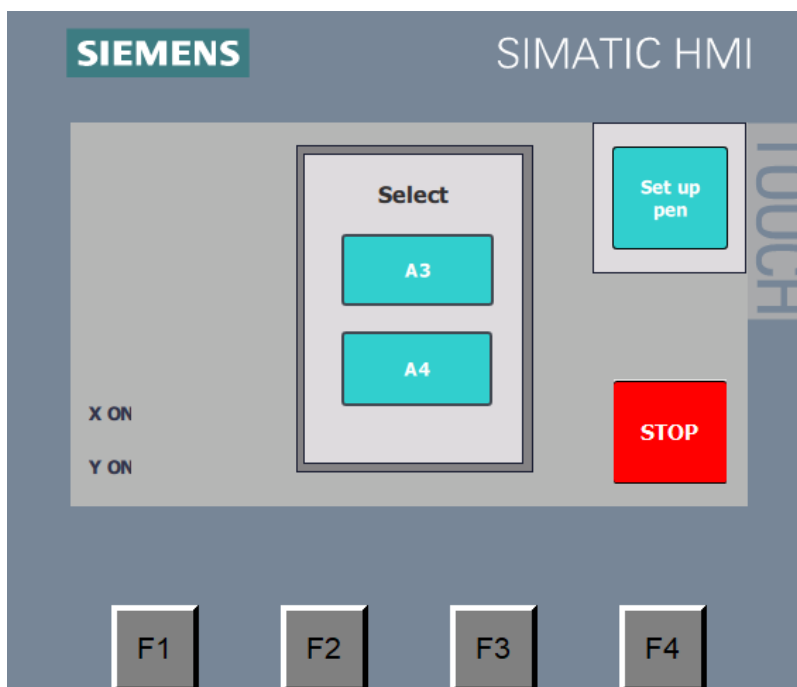
Velkou výhodou menšího z dvou panelů je dotykový displej. Druhý z panelů KP900 Comfort dotykovým displejem nedisponuje, a je tedy pro jeho ovládání nutné připojit externí ovládací zařízení, například počítačovou myš. Vzhledem k tomu je tak vhodné realizovat základní ovládání přípravku na dotykový displej, u kterého je jisté, že jím aplikaci můžeme ovládat bez ohledu na další příslušenství.

Pro KTP400 panel navrhne vizualizaci základní obsluhy přípravku jako jeho zapnutí a vypnutí, volbu formátu, možnost manuálního řízení pera po papíru a v poslední řadě také výměnu pera.

11.3.1 Základní obsluha

Pro základní funkce přípravku vytvoříme novou obrazovku v systému panelu. To provedeme v příslušné záložce daného panelu v oddílu *Screens* kliknutím na tlačítko *Add new screen*. Ovšem v případě obrazovky základní obsluhy vytvoříme vizualizaci již na základní obrazovce tzv. *Root screen*. Jedná se o úvodní obrazovku programu běžícím na panelu. *Root screen* však v tento moment přejmenujeme pomocí příkazu *Rename* v nabídce pravého kliknutí na nové označení *Menu*. Je potřeba se ujistit, že obrazovka je opravdu úvodní, to je signalizováno zeleným trojúhelníkem u ikony dané obrazovky.

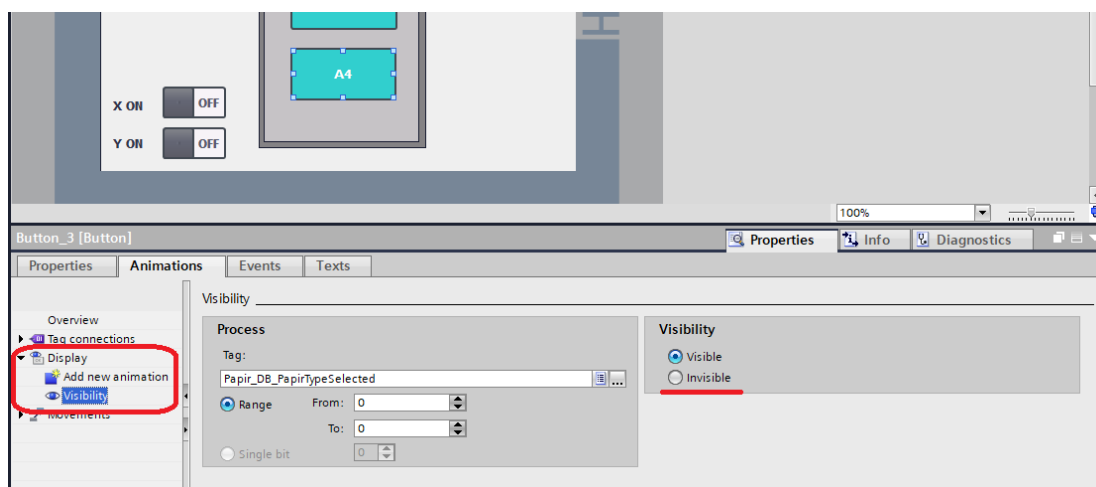
Na obrázku 11.7 je zobrazena vizualizace pro obrazovku základní obsluhy. Uprostřed okna figuruje základní výběr formátu papíru. Zapnutí zařízení je totiž podmíněné založením papíru do přípravku a výběrem jeho formátu. Zapnutím přípravku totiž automaticky spustíme referování os technologických objektů, během kterého dochází již k pohybu papíru a vložení papíru by již nebylo možné.



Obr. 11.7 Základní obsluha přípravku

Podmínka výběru formátu je realizována pomocí grafické viditelnosti okna výběru formátu. Ten totiž překrývá tlačítka ON a OFF, dokud není jeden formát vybrán. Řešení viditelnosti je

k nahlédnutí na obr. 11.8. V tomto případě musí být stejný postup implementovaný pro všechny objekty vizualizace okna, tedy pro samotné okno, dvě tlačítka A4 a A3 a text Select.



Obr. 11.8 Řešení viditelnosti úvodního okna

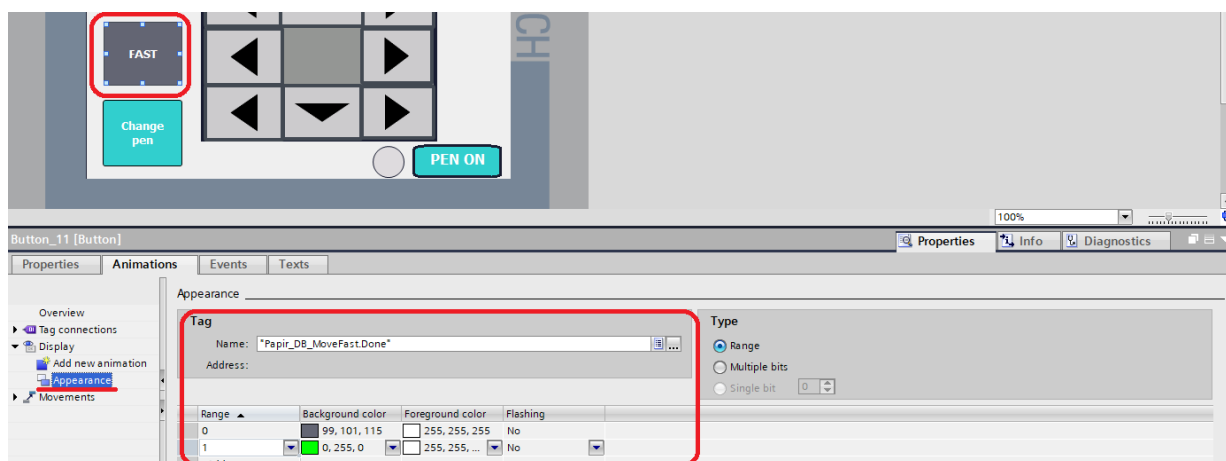
Jak již bylo řečeno, kromě toho na obrazovce najdeme zapínací tlačítko ON, vypínací tlačítko OFF a tlačítko Control. Zapínací tlačítko ON je viditelné pouze v případě vypnutých obou technologických objektů, jak osy papíru, tak i pera. Tlačítko Control slouží k aktivování obrazovky s manuálním řízením osy a tlačítko OFF pro vypnutí zařízení.

V poslední řadě najdeme na obrazovce výběr Set up pen pro případ, že není vybráno žádné pero a tlačítko STOP. Viditelnost výběru Set up pen je podmíněna hodnotou proměnné ActualPen = 0. Tlačítko STOP je bezpečnostní tlačítko odpojící všechny základní bloky pohybu TO.

11.3.2 Manuální řízení

Další obrazovka, na kterou v rámci vizualizace panelu KTP400 narazíme, slouží k manuálnímu řízení pera po papíru. Na obrazovku se dostaneme ze základní obrazovky stisknutím tlačítka Control realizující příkaz *ActivateScreen*.

Viditelnost směrových tlačítek pohybu je podmíněna provedeným referováním obou os. Dále na obrazovce najdeme tlačítko pro navrácení na předchozí obrazovku, tlačítko přechodu na obrazovku výměny pera a tlačítko pro rychlejší pohyb pera. Zde při stisknutí vyvoláme příkaz invertující bit proměnné *MoveFast* u obou os. Zároveň je rychlejší režim signalizován zeleným zbarvením tlačítka skrze *Animations* nastavení objektu tlačítka, zobrazena na obr. 11.9.

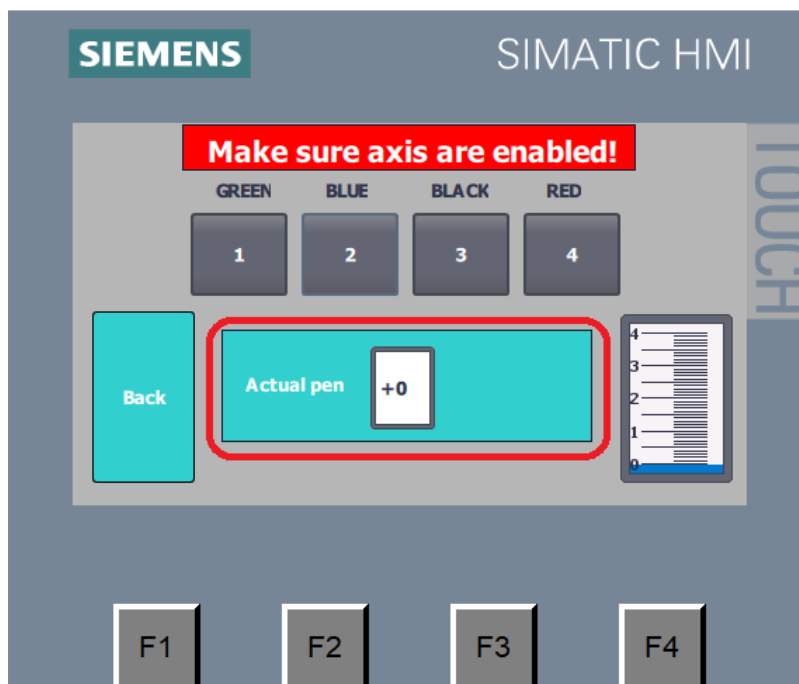


Obr. 11.9 Použití funkce *Animations*

11.3.3 Výměna pera

Poslední používanou obrazovkou je Výměna pera. Jak již z názvu vyplývá, slouží k organizaci a provedení výměny pera. Najdeme zde čtyři tlačítka realizující výběr pera dané barvy. Tlačítková logika funguje na základě návrhu vnitřní logiky výměny pera.

Dále tu najdeme textový objekt nad tlačítky a objekt *I/O field*, zvýrazněný na obr. 11.10, který zobrazuje aktuální číselné hodnoty vybraného pera. Jedná se o zobrazení hodnoty proměnné *ActualPen*. Stejnou funkci zastává také objekt *Bar*. Ten ukazuje také číselnou hodnotu vybraného pera, pouze s tím rozdílem, že hodnota je reprezentována graficky. Jedná se především o demonstraci dalšího z nástrojů vizualizace HMI panelů.



Obr. 11.10 Výměna pera, panel KTP400 Basic

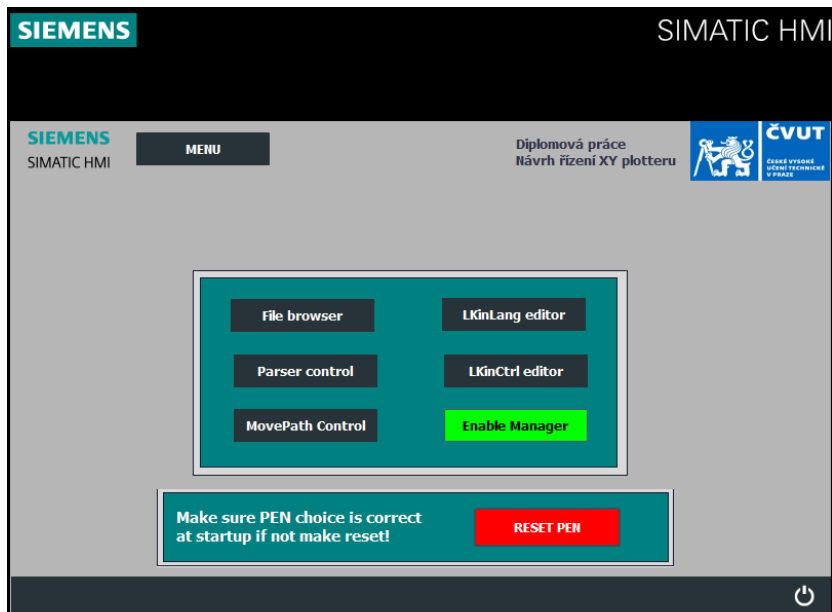
11.4 Návrh vizualizace panelu KP900 Comfort

Část řízení byla již obsloužena vizualizací panelu KTP400 Basic. Pro zbývající části navrhne obsluhu na druhém z panelů, SIMATIC KP900 Comfort.

Hlavním důvodem pro použití odlišného panelu je skutečnost, že se jedná o panel řady Comfort, bez něhož se neobejdeme při implementaci tzv. *faceplate* HMI aplikací z knihovny *Kinematics Control* a *Kinematics Language*.

V rámci návrhu obsluhy pro tento panel tedy implementuje celkem tři HMI aplikace, jednu z knihovny *LKinCtrl* a dvě z knihovny *LKinLang*. Dále vytvoříme obsluhu a dohled nad překladem G-code souboru na *pathData* formát. Další obrazovku bude tvořit ovládání bloku *MovePath* pro pohyb automatického vykreslování společně s možností výměny barvy pera.

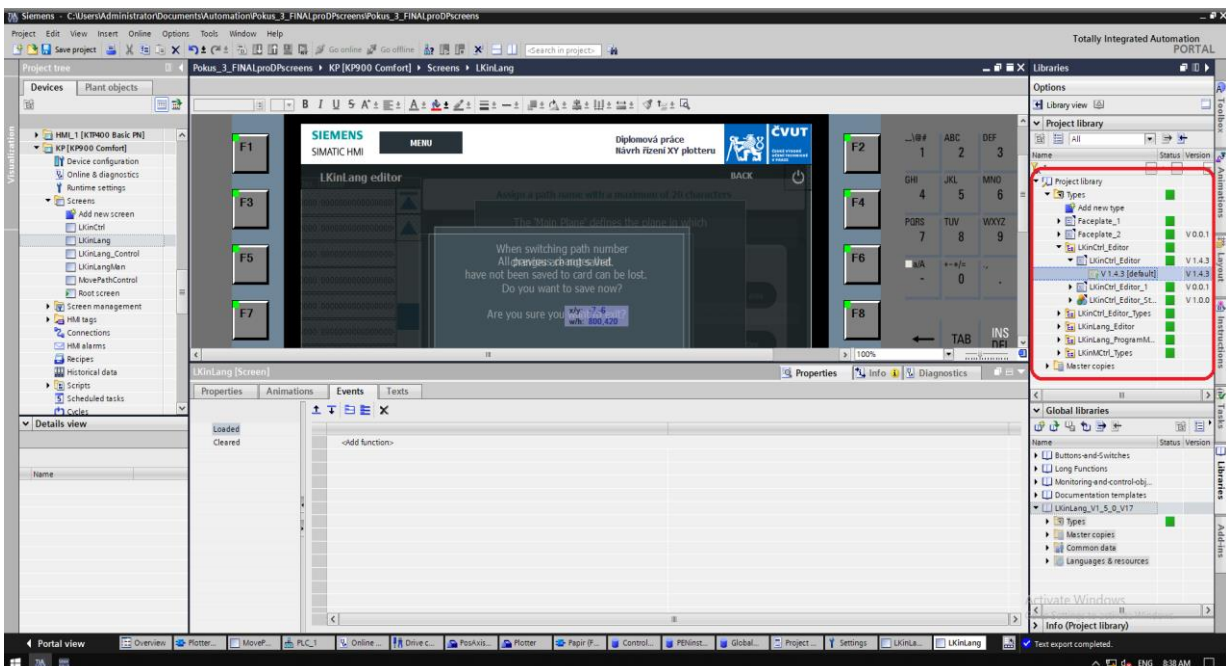
Než provedeme výše popsané kroky, vytvoříme úvodní obrazovku. Základní obrazovka bude sloužit jako rozcestník k dalším obrazovkám podle aktuální potřeby. Do okna vložíme pět přepínačů sloužících k aktivaci určité obrazovky panelu a jeden přepínač spínající organizační HMI aplikaci *LKinLang_ProgramManager*. Dále zde umístíme omezeně viditelný blok, který bude umožňovat vynulovat proměnnou *ActualPen*. Tento krok byl realizován na základě toho, že nemůžeme zajistit 100% funkčnost výměny pera v situaci, kdy někdo vymění pero manuálně bez zapnutého přípravku. V takovém případě by hodnota proměnné neodpovídala skutečnosti a mohlo by dojít k poškození přípravku.



Obr. 11.11 Rozcestník, panel KP900 Comfort

11.4.1 Implementace LKinLang HMI aplikací

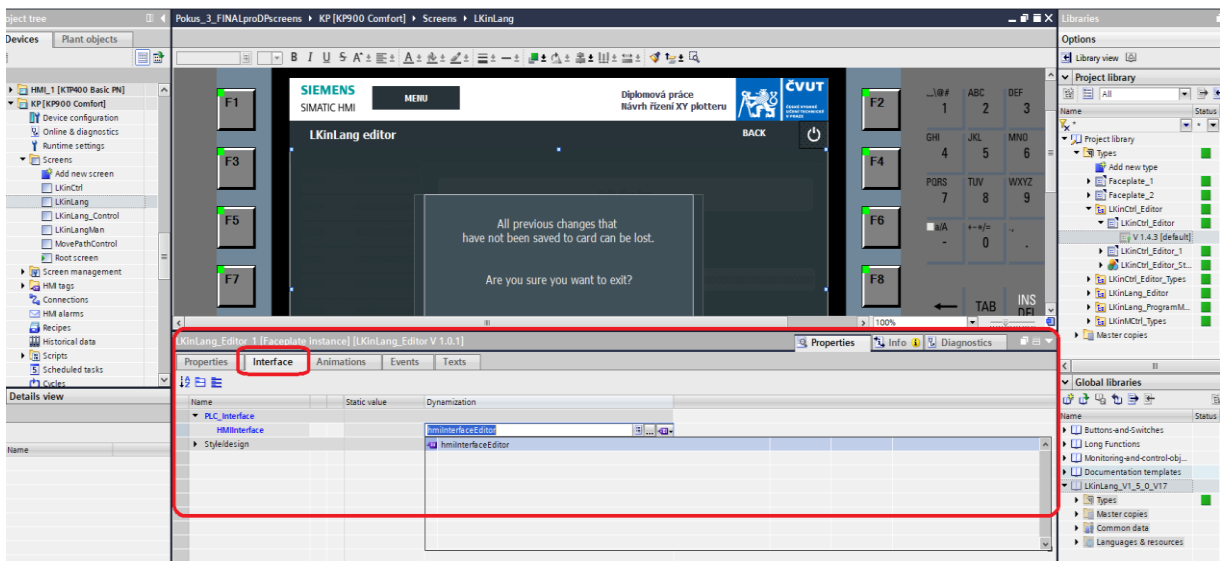
Realizace HMI aplikace z externí knihovny je vcelku jednoduchým krokem. Aplikaci, resp. *faceplate*, jsme již v rámci importu knihovny nahráli do projektové knihovny. Stačí tedy v nabídce na pravé straně stránky, zvýrazněno na obr. 11.12, vybrat nabídku *Libraries* a otevřít v nabídce záložku *Project library*. Následně vybereme záložku *Types* a po rozbalení nabídky přetáhneme položky *LKinLang_Editor* a *LKinLang_ProgramManager* pomocí *Drag & drop* instrukce na vlastní obrazovky.



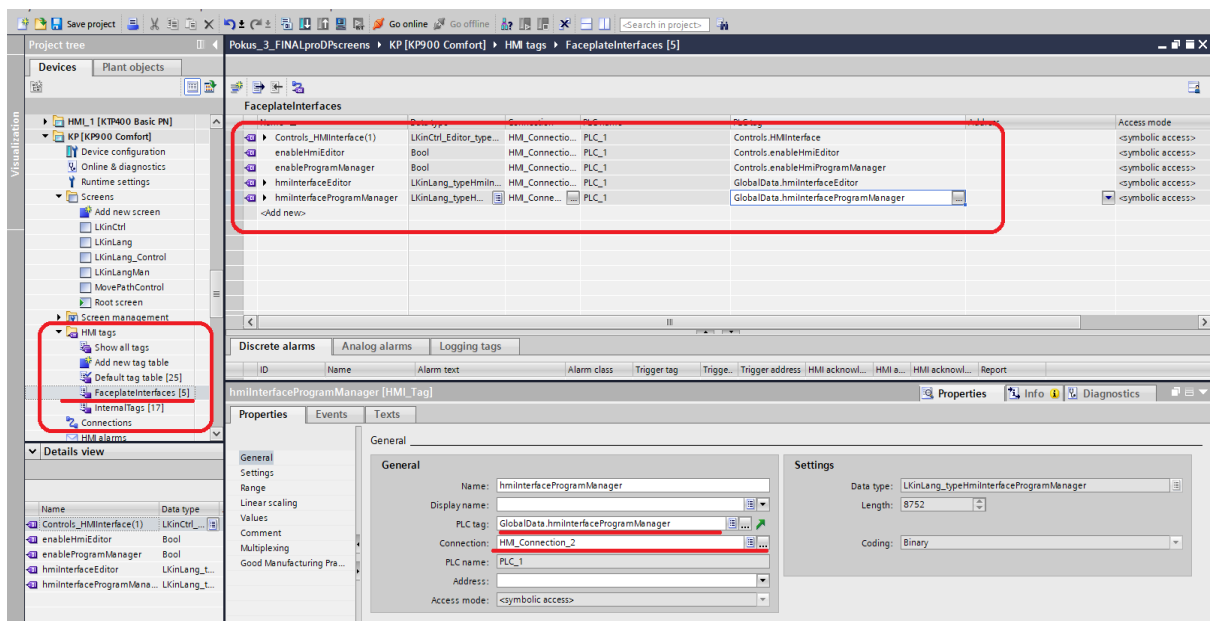
Obr. 11.12 Vložení *faceplate* HMI aplikace

Jakmile je aplikace vložena, stačí na aplikaci kliknout a v dolním okně v záložce *Interface* propojit HMI aplikaci s PLC tagem připraveného pro propojení aplikace a funkčního bloku v PLC návrhu. Proměnnou najdeme inicializovanou v data bloku *GlobalData*. Postup je totožný pro obě

aplikace. Aplikaci v rámci vizualizace propojujeme s HMI tagem, a proto je nutné pro proměnnou z bloku *GlobalData* vytvořit odpovídající HMI tag. Jeho definici včetně realizace propojení najdete na obr. 11.13 a 11.14.



Obr. 11.13 Propojení HMI aplikace s FB blokem programu



Obr. 11.14 Definice HMI tag propojení

Aplikace *LKinLang_Editor* slouží k úpravě aktuálně načteného souboru typu G-code. Můžeme tak tvořit, mazat či upravovat jednotlivé příkazy souboru v připraveném prostředí skrze HMI panel.

Aplikace *LKinLang_ProgramManager* hraje zásadní roli v návrhu, neboť funguje jako adresář a správce souborů na paměťové kartě a externích úložištích. Prostřednictvím této aplikace vybíráme požadované výkresy k překlada.

11.4.2 Implementace LKinCtrl_Editor aplikace

Z knihovny *Kinematics Control* použijeme pouze jednu HMI aplikaci, a to *LKinCtrl_Editor*. Ta umožňuje úpravu a tvorbu *pathData* bloků v již plně integrovaném vývojovém prostředí na HMI panelu bez nutnosti PLC programování uvnitř projektu.

Postup její implementace je totožný s implementací aplikace z knihovny *LKinLang*. Tentokrát se proměnná pro propojení odpovídajícího funkčního bloku v PLC programu nachází v data bloku *Controls*. Konkrétní umístění proměnné záleží na konfiguraci projektu, a je tedy jinak libovolné.

11.4.3 Ovládání překladu výkresu

K obsluze překladu souboru G-code na formát pathData vytvoříme samostatnou obrazovku. K řízení potřebujeme minimálně dvou tlačítek *Interpret program* a *Stop interpreting*. Tyto tlačítka budou ovládat funkční blok *LKinLang_Parser*. Tlačítka doplníme o funkci změny barvy v sekci *Animations* na základě hodnoty vnitřního PLC tagu. Dále využijeme informačních bloků *I/O field* k zobrazení aktuálního stavu procesu překladu. Pomyslný překlad souboru je složen nejenom z překladu příkazového souboru typu G-code, ale také přípravy instrukcí pro ovládání přiložení pera při kreslení. Abychom měli dostatečný přehled o technologii, vytvoříme kontrolní panel složený z informačního bloku překladu, stavu instrukcí pro přiložení pera a stavu technologického objektu kinematiky. Zde je potřebou pouze správně propojit odpovídající nástroj *I/O field* s vybranou status proměnnou. Funkční blok *LKinLang_Parser* stejně jako blok *LKinCtrl_MC_MovePath* disponují požadovými status parametry v rámci své struktury. V případě kontroly generování instrukcí pro přiložení pera využijeme kontrolní proměnnou *MovePathLangReady* vytvořenou při návrhu logického řízení přiložení pera. Propojení objektu *I/O field* s daným PLC tagem provedeme výběrem daného tagu v záložce *Properties* v sekci *General*, buňka *Tag*.

U funkčního bloku *LKinLang_Parser* ale najdeme kromě běžného status parametru také proměnné dohlížející nad překladem v podobě zobrazení aktuálně překládané instrukce a zobrazení diagnostické hlášky v případě chyby překladu. Pro lepší přehled nad operací tedy využijeme těchto statusových parametrů a vložíme další dva informační bloky *I/O field*. Rozdíllem však bude, že blok diagnostické hlášky bude viditelný pouze v případě chyby překladu. To uděláme pomocí nastavení viditelnosti objektu v sekci *Animations* v závislosti na statusové proměnné překladového bloku *Parser*. Řídící PLC tagy objektů jsou v tomto případě *LKinLang_ProgramLineDisplay_DB_parsedLine* pro aktuálně překládanou instrukci a *LKinLang_Parser_DB_errorString* pro chybovou hlášku.

Posledním krokem je vložení tlačítka *MovePath control* pro aktivaci obrazovky řízení pohybu *MovePath*. Po grafické stránce vytvoříme stejné tlačítko jako pro řízení překladu, pouze naprogramujeme funkci *Appearance* v sekci *Animations*, tedy změnu barvy na základě proměnné *MovePathLangReady*. To znamená, že jakmile bude přeložený výkres připraven k vykreslování, tlačítko *MovePath control* změní barvu na zelenou žlutého blikání rámečku.

11.4.4 Řízení pohybu pohonu MovePath

Poslední krok vyhotovení výkresu obstará obrazovka *MovePath Control*. Tato obrazovka bude mít na starosti řízení bloku *LKinCtrl_MC_MovePath*. Vytvoříme zde tedy sekci *Kinematics* tvořenou třemi tlačítky řídící funkční blok *MovePath*. Tlačítka budeme opět stylizovat do stejné podoby jako předchozí. Přestože hovoříme o tlačítkách, v návrhu použijeme přepínače. To znamená, že logickou proměnnou PLC tagu můžeme jednoduše přepínat. Pro sepnutý přepínač nastavíme funkci *Appearance*, opět se jedná změnu barvy na zelenou v případě sepnutí přepínače.

Dále zde, stejně jako v případě obrazovky překladu, implementujeme informační objekty *I/O field* zobrazující informace o technologickém objektu pohonu. Tentokrát však použijeme status proměnné týkající se čistě bloku *MovePath* (*LKinCtrl_MC_MovePath_DB_status* a *LKinCtrl_MC_MovePath_DB_active*) a řízení přiložení pera. Pod informačními bloky navíc přidáme *I/O field* objekt zobrazující právě vykonávanou instrukci přeloženého souboru typu G-code blokem *MovePath*, podobný jako v případě obrazovky ovládání překladu, pouze se jedná o zobrazení PLC tagu *LKinLang_ProgramLineDisplay_DB_activeLine*.

Dále vytvoříme sekci *Control*. Zde umístíme zbývající kontrolní a řídicí prvky zařízení. Jedná se o tlačítko *Reset axes*, které má na starosti řízení PLC tagu *Controls_ResetAxes* a kontrolní objekty

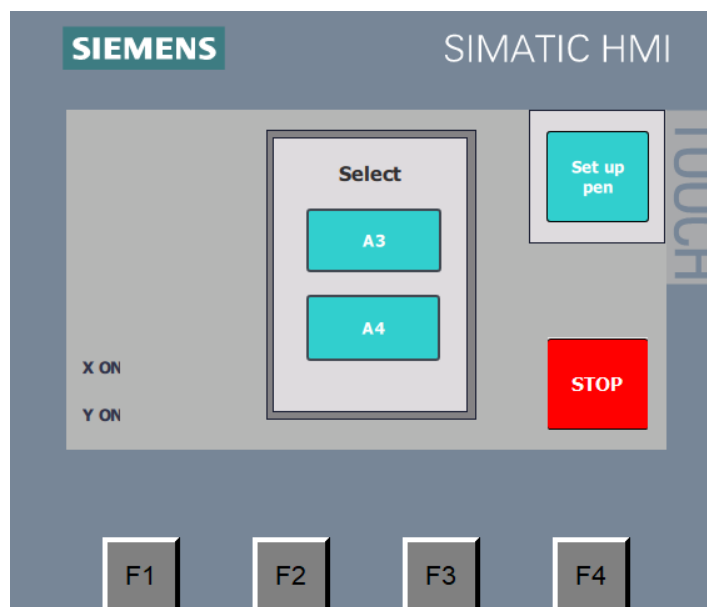
ukazující aktuálně vybranou barvu pera (*Actual pen*) a status technologického objektu kinematiky. V případě informace o statusu technologického objektu využijeme kontrolních proměnných z data bloku *Controls*.

Na závěr vytvoříme na levé straně obrazovky ovládací rozhraní výměny pera. Tlačítka jsou graficky upravena na základě odpovídající barvy pera. Výměna pera je možná pouze po dokončení referování obou os pohonu, to je zajištěno opět pomocí příkazu viditelnosti (nastavení *Visibility* v sekci *Animations*) všech objektů rozhraní. Signalizace právě použité barvy pera je vytvořena pomocí signalizačního rámečku kolem tlačítka každé barvy pera. Rámeček v případě odpovídající hodnoty proměnné *ActualPen* změní svou barvu na žlutou. Zde znovu využijeme funkci *Appearance* ve spojení s proměnnou *ActualPen*.

KAPITOLA 12: POPIS NAVRŽENÉHO ŘÍZENÍ XY PLOTERU

12.1 Základní ovládání

Přípravek XY ploter vyžaduje pro své použití založení papíru pomocí zakládací páky. Přípravek je schopný pojmout pouze standardní formáty papíru A3 nebo A4. Pokud je papír založen, je nutné výběr formátu před zapnutím potvrdit kliknutím na příslušný formát na obrazovce panelu KTP400 Basic, podrobnější pohled na obr. 12.1.

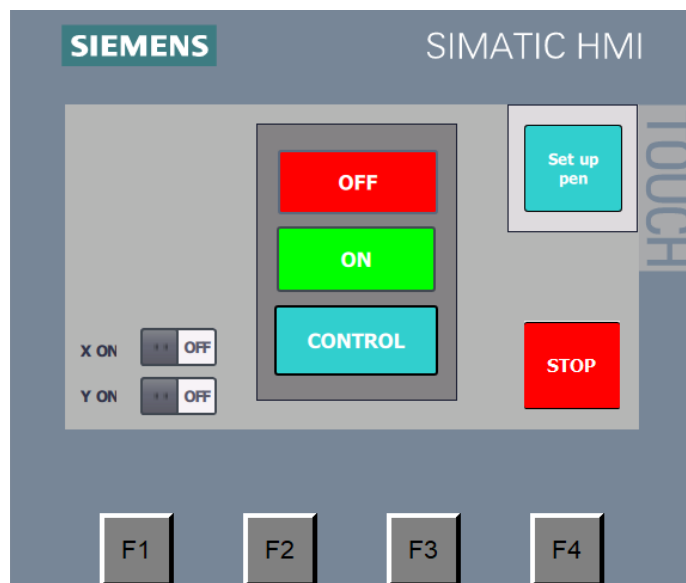


Obr. 12.1 Výběr formátu papíru

Základní ovládání přípravku probíhá na již zmíněném panelu KTP400 Basic. Mimo úvodní obrazovku, kde je možné po potvrzení výběru formátu papíru přípravek zapnout, či naopak vypnout, zde najdeme také manuální ovládání pera a možnost výměny pera.

12.1.1 Zapnutí

Pro zapnutí je nutné založit papír dovnitř přípravku a potvrdit volbu formátu. Po potvrzení se na panelu KTP400 Basic objeví ovládací panel. Na něm stiskneme zelené tlačítko ON, zvýrazněno na obr. 12.2. Zařízení započne úvodní proces referování obou os pohybu, při němž se pero dostane do počátku pomyslného souřadnicového systému. Konečná poloha referování os je zobrazena na obr. 8.1, označena jako parkovací poloha.



Obr. 12.2 Ovládací panel přípravku

12.1.2 Vypnutí

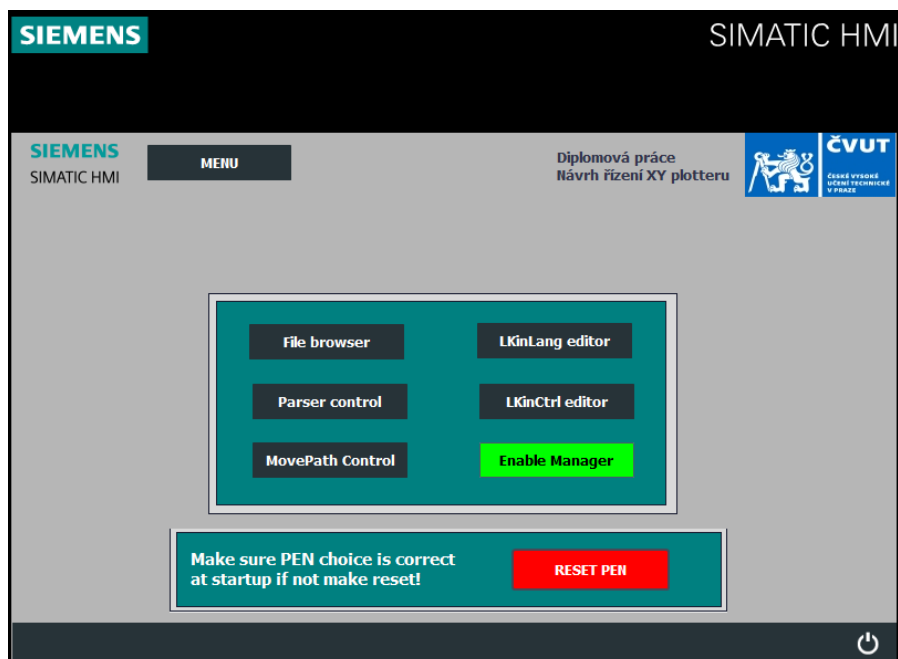
V případě, že chceme přípravek vypnout, je nutné se ujistit, že zařízení momentálně nevykonává jiný požadavek, který by byl příkazem vypnutí narušen, a mohlo by dojít k zastavení chodu přípravku.

Vypnutí přípravku je možné provést pouze na úvodní obrazovce panelu KTP400 Basic. Pokud na panelu není momentálně zobrazena úvodní obrazovka, přepneme se na ni pomocí tlačítka Back. Vypínací tlačítko je červené tlačítko s popisem OFF. Po jeho stisku se vrátí papír do základní polohy, při níž je možné ho z přípravku odebrat. Jakmile je pohyb dokončen, PLC aplikace není aktivní a přípravek je vypnut.

Vypínací tlačítko OFF zobrazena výše na obr. 12.2.

12.1.3 Reset řízení pera

V případě, že s přípravkem bylo během vypnutého stavu manipulováno ve smyslu manuální výměny pera, proměnná kontrolující aktuálně vybranou barvu pera nemusí fungovat správně. Pro tyto případy zde je možnost resetu aktuálně vybraného pera. Tlačítko resetu paměti barvy pera se objeví po zapnutí přípravku na panelu KP900 Comfort v dolní části úvodního rozcestníku a po dobu 20 vteřin umožňuje vymazat proměnnou aktuálně vybraného pera. Pokud tak učiníme, řízení výměny předpokládá, že není vybráno žádné pero a všechny se nacházejí ve svých parkovacích pozicích v zásobníku.



Obr. 12.3 Rozcestník, panel KP900 Comfort

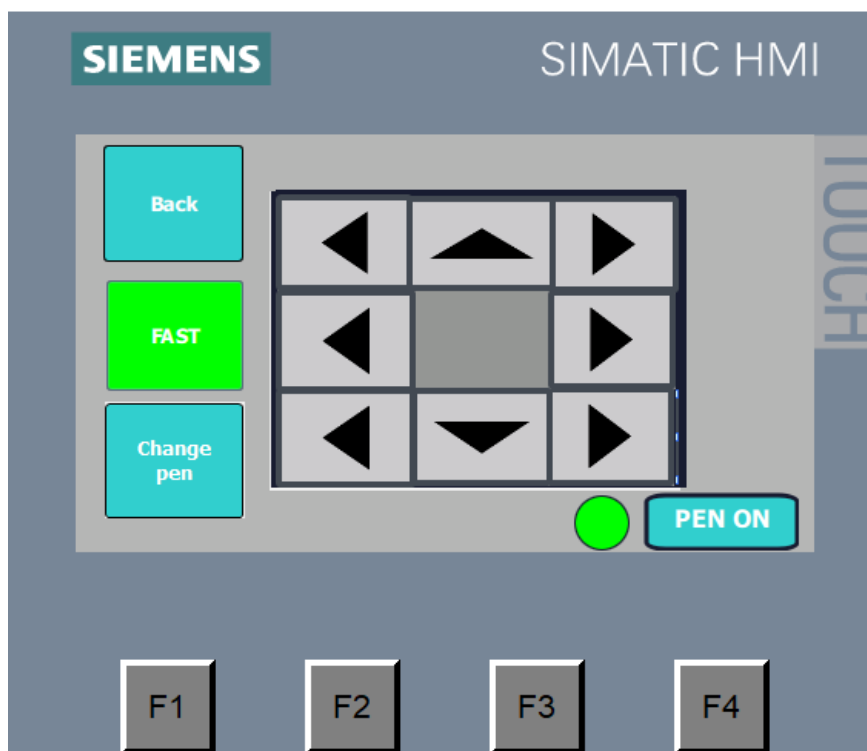
12.2 Manuální řízení

Mimo úvodní obrazovku, kde je možné po potvrzení výběru formátu papíru přípravek zapnout, či naopak vypnout, najdeme na panelu KTP400 Basic také manuální ovládání pera.

K manuálnímu ovládání pera se dostaneme prostřednictvím tlačítka Control z úvodní obrazovky. Pokud je na panelu KTP400 Basic aktuálně zobrazena jiná obrazovka, je nutné se přepnout nejdříve na úvodní obrazovku Menu pomocí příkazu Back a dále stisknout tlačítko Control.

Zde máme k dispozici celkem osm směrových tlačítek, kterými je možné pohybovat perem po papíru. Vlivem absence *multi-touch* technologie je možné současně použít pouze jeden směr. Pokud požadujeme přiložení pera, je nutné stisknout čtvrté programové tlačítko pod displejem. Jedná se o první tlačítko z pravé strany displeje. Tlačítko pro přiložení pera funguje jako přepínač, a proto první stisknutí pero trvale přiloží k papíru. Druhé stisknutí potom pero zvedne. Přiložení pera je signalizováno kontrolkou nad tlačítkem, v případě přiložení svítí zeleně.

Vedle směrových ovládacích tlačítek na obrazovce najdeme také tlačítko FAST, které nastavuje dvojnásobnou rychlost pohybu pera oproti základní rychlosti. V případě zeleného zbarvení tlačítka je zvolena větší rychlost. Dále zde najdeme možnost přechodu na obrazovku výměny pera, a to tlačítkem Change pen, a na závěr zde máme nahoře tlačítko Back, kterým se dostaneme zpět na úvodní obrazovku ovládání.



Obr. 12.4 Manuální řízení pera

12.3 Výměna pera

Změna barvy kreslicího pera je možná z obou panelů. V této podkapitole bude uveden popis výměny pera na obou HMI ovládacích panelech.

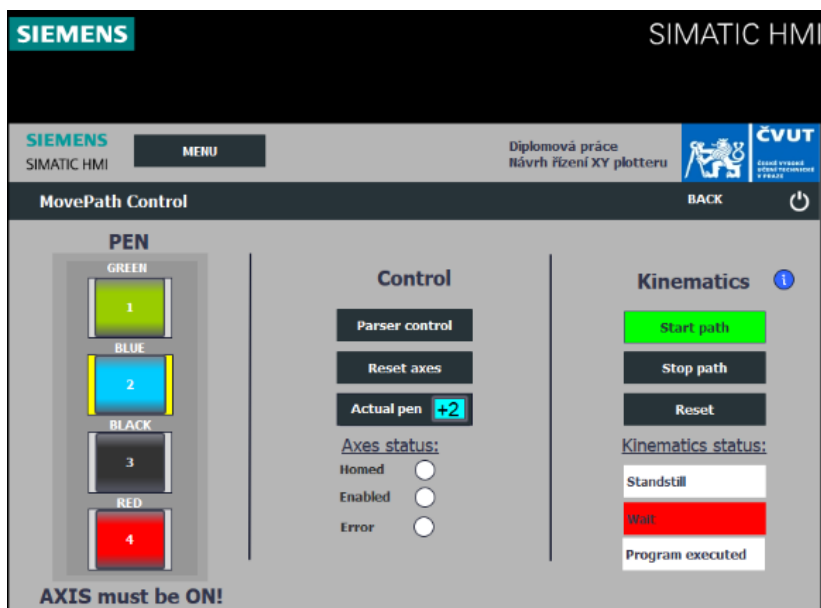
12.3.1 Ovládání na panelu KP900 Comfort

Ovládání výměny pera se v případě panelu KP900 Comfort nachází v sekci MovePath Control. Možnost výměny pera není umožněna, dokud není dokončeno referování os pohonu. V tom případě je ovládání výměny neviditelné.

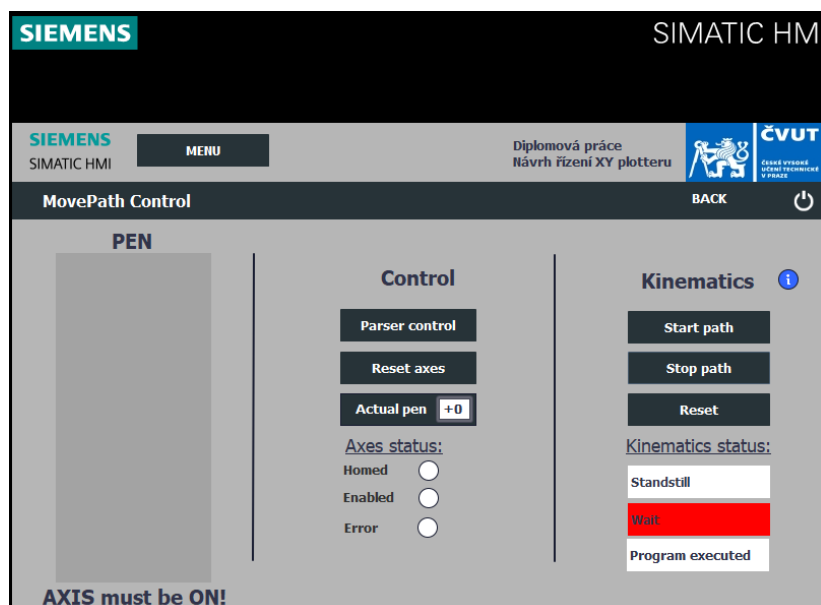
K ovládání slouží čtyři barevně odlišená tlačítka s názvem určité barvy, kterou reprezentují. Rámeček kolem každého tlačítka signalizuje, zda je daná barva právě vybraná. V případě, že ano, má rámeček žlutou barvu.

Ovládání pera je závislé na předchozích krocích. V případě, že se jedná o první výběr pera ze zásobníku, je proveden výběr požadované barvy pera. Jakmile však již určitá barva pera byla vybrána předtím, můžeme požadovat její výměnu. To uděláme pouhým stiskem požadované barvy pera, která je odlišená od právě vybrané. V tomto případě řízení automaticky právě používanou barvu pera vrátí a poté vybere nově požadovanou. V případě, že chceme výběr vyprázdnit, stačí stisknout na právě používanou barvu pera a pero se vrátí do zásobníku bez náhrady.

Na obr. 12.5 vidíme již zpřístupněnou možnost výměny pera, kdežto na obr. 12.6 probíhá zatím referování os a volba není umožněna.



Obr. 12.5 Výměna pera, panel KP900 Comfort



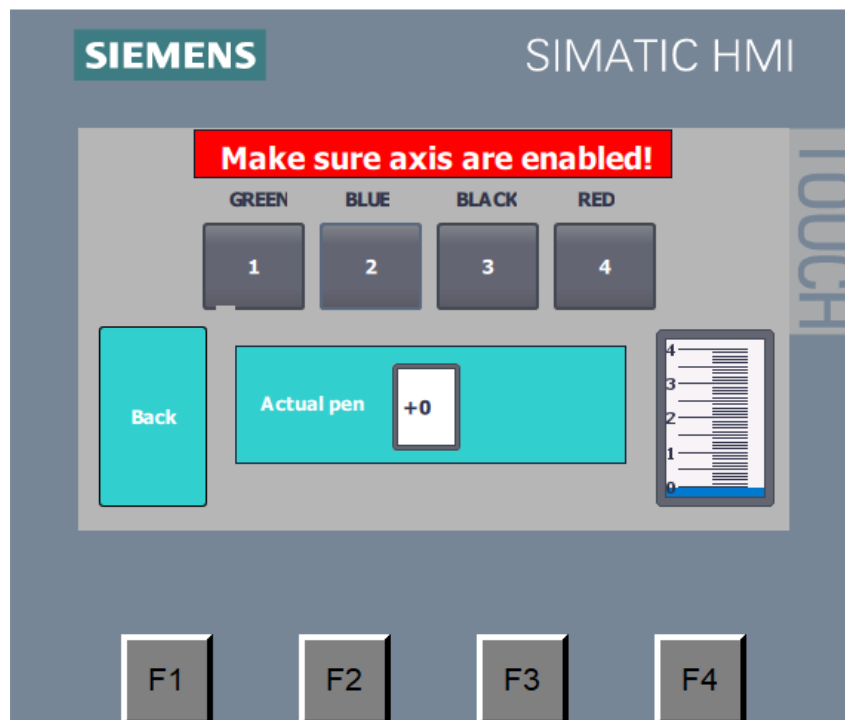
Obr. 12.6 Nezpřístupněná výměna pera, panel KP900 Comfort

12.3.2 Ovládání na panelu KTP400 Basic

Zde ovládání pera najdeme na obrazovce Pen choice. K výměně pera se dostaneme ze základní obrazovky pomocí tlačítka Set up pen pouze v případě, že se jedná o první výběr a máme prozatím tzv. prázdný výběr. V ostatních případech je nutné přejít na obrazovku Control manuálního řízení pera a zde stisknout tlačítko Change pen.

Ovládání výměny pera pomocí čtyř programových tlačítek funguje úplně stejně jako u panelu KP900 Comfort. Popis ovládání byl vysvětlen již v podkapitole 12.3.1.

Číselnou hodnotu aktuálně vybrané barvy pera vidíme na ukazateli pod programovými tlačítky. V případě tzv. prázdného výběru je hodnota ukazatele nula.



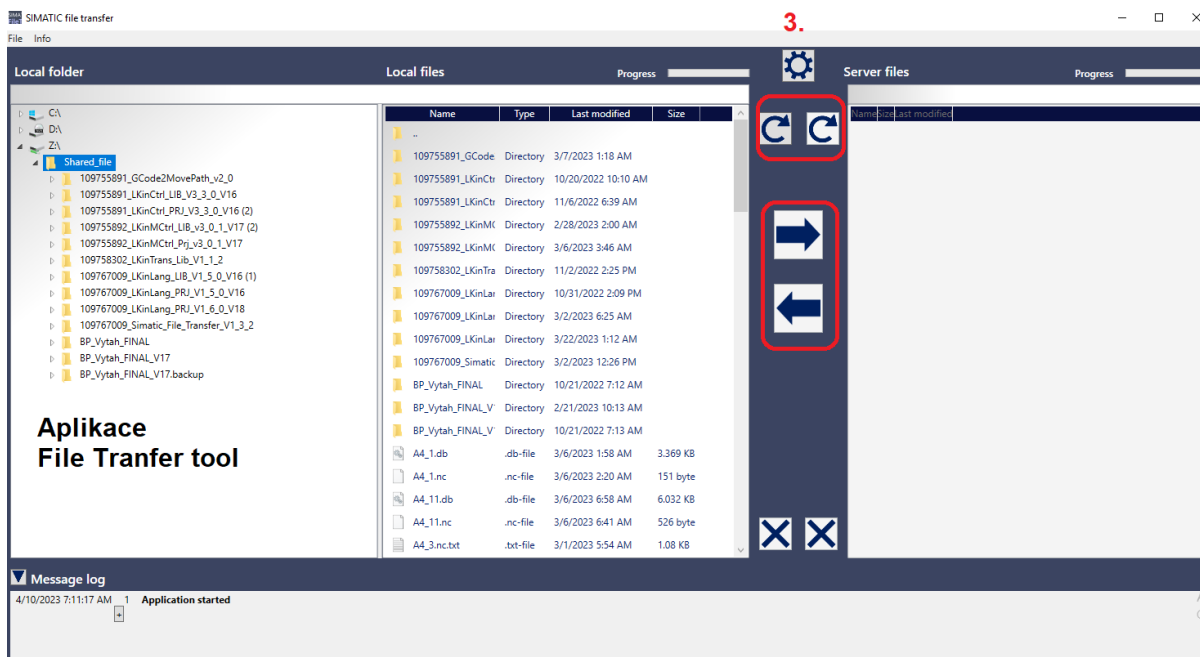
Obr. 12.7 Výměna pera, panel KTP400 Basic

12.4 Import výkresu do PLC zařízení

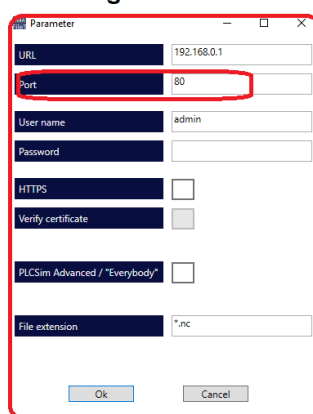
Import příkazových souborů typu G-code vybraných výkresů z osobního počítače na paměťovou kartu PLC zařízení se provádí pomocí doplňkové aplikace *File Transfer tool*. Pro přehrání souborů je nutné mít počítač s aplikací *File Transfer tool* připojený k lokální síti, kam je připojeno také PLC zařízení. Jakmile je spojení provedeno, otevřeme aplikaci a v úvodním okně nastavíme parametry dostupné na obr. 12.8. U přihlašovacího profilu se jedná o stejné údaje, které byly uvedeny při konfiguraci funkce *Web server* na příslušném PLC zařízení.

Po potvrzení nastavení komunikace prostřednictvím tlačítka *Ok*, je ustanovena komunikace mezi adresářem PC a adresářem paměťové karty PLC zařízení. Na pravé straně najdeme adresář paměťové karty a na levé adresář osobního počítače. Přesun mezi jednotlivými adresáři je možný směrovými tlačítky umístěnými mezi adresáři. V případě správy obou adresářů, je možné označit jakýkoli soubor a pravým kliknutím se zobrazí nabídka běžně dostupných úprav.

Pokud bychom náhodou potřebovali obnovit adresáře, použijeme obnovovací tlačítka nad směrovými tlačítky přenosu, zvýrazněno na obr. 12.8.



Konfigurační okno



Obr. 12.8 Rozhraní aplikace File Transfer tool

12.5 Výběh souboru

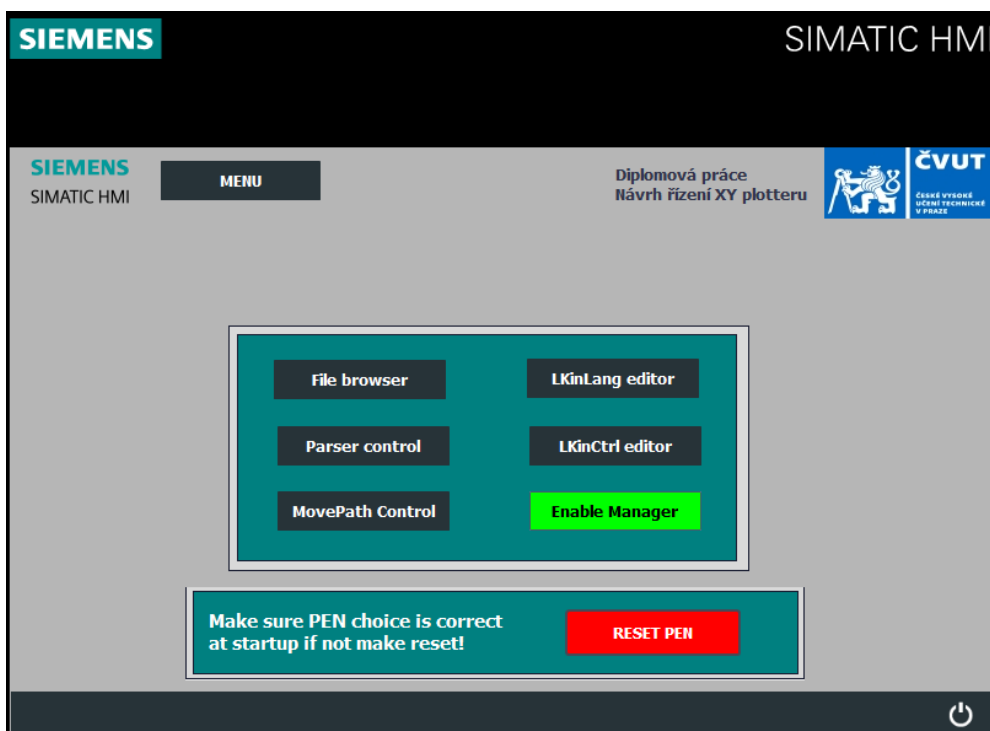
Soubory typu G-code musí být patřičně upraveny dle již zmíněných kritérií, jinak nebude překlad souboru v PLC zařízení úspěšný. Jakmile je soubor připraven k překladu, můžeme soubor nahrát pomocí aplikace *File Transfer tool* přímo na paměťovou kartu PLC zařízení, nebo připojíme k HMI panelu KP900 Comfort externí datové úložiště, na kterém se výkres v podobě souboru G-code nachází.

Pokud soubor není importován přímo na paměťovou kartu PLC zařízení, je nutné ho nejdříve na paměťovou kartu přehrát. Tento krok provedeme na obrazovce *File browser* panelu KP900 Comfort.

Obrazovku *File browser* spustíme kliknutím na tlačítko File browser v základním rozcestníku panelu. Pokud se aktuálně nacházíme na jiné obrazovce, pomocí systémového tlačítka MENU přejdeme zpátky na rozcestník a odtud již popsaným způsobem přejdeme do adresáře *File browser*. Systémové tlačítko MENU najdeme v záhlaví vizualizace v levém horním rohu, zvýrazněno na obr. 12.9. Před spuštěním obrazovky je nutné se ujistit, že funkce Enable Manager byla zapnuta. Pokud ne, tlačítko bliká červeně. Jakmile dojde k zapnutí funkce, tlačítko přejde do zapnutého stavu signalizovaného zelenou barvou. Příkaz Enable Manager spouští HMI aplikaci *LKinLang_ProgramManager*.



Obr. 12.9 Výběr souboru



Obr. 12.10 Rozcestník, zapnutý Enable Manager

Kliknutí na ikonu File browser nás přenese do nového okna. Jedná se o prostředí HMI aplikaci *LKinLang_ProgramManager*. V levé části najdeme nabídku datových úložišť. Jako první nabízená stojí paměťová karta PLC zařízení. Další dvě buňky reprezentují dva externí USB porty HMI panelu umožňující připojit až dvě datová úložiště. Hlavní okno aplikace tvoří adresář zvoleného úložiště. Na pravé straně potom máme nabídku instrukcí pro správu souborů v adresáři.

V případě, že soubor s výkresem máme na externím úložišti, musíme takové úložiště nejdříve připojit k HMI ovládacímu panelu pomocí jednoho ze dvou USB portů. Dokud zařízení není připojeno, je výběr šedý. Jakmile je úložiště přístupné, možnost výběru externího úložiště je zpřístupněna. V tom okamžiku můžeme úložiště v levé nabídce vybrat. Tím zobrazíme adresář zařízení v prostředním okně aplikace. Označíme vybraný soubor a na pravé straně obrazovky přehrajeme soubor na paměťovou kartu pomocí funkce *Transfer*. Po stisknutí tlačítka funkce započne přenos souboru na paměťovou kartu, dokončení operace je signalizováno na informační liště v horní části aplikace těsně pod záhlavím.

Pokud se výkresový soubor nachází na paměťové kartě, ať už přehráním z externího úložiště, či použitím aplikace *File Transfer tool*, přehrajeme daný soubor do PLC operační paměti k překladu.

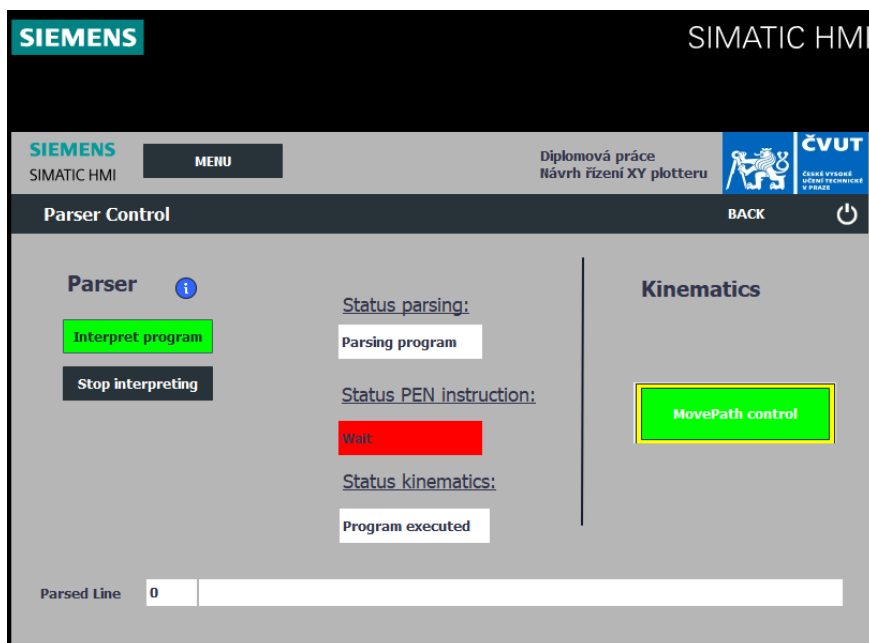
V nabídce datových úložišť přejdeme na adresář paměťové karty, označíme požadovaný soubor výkresu a v pravé nabídce klikneme na funkci *Load*. Soubor se začne nahrávat do operační paměti. Stav operace je možné sledovat na horní informační liště. Jakmile se objeví informace *File loaded*, je soubor připravený k překladu. Nyní můžeme opustit obrazovku *File browser* pomocí systémového tlačítka MENU v záhlaví vizualizace.

12.6 Překlad G-code souborů

Předpokládáme, že vybraný soubor výkresu máme nahraný v operační paměti PLC zařízení. To znamená, že soubor je připraven k překladu na spustitelný formát *pathData*. K tomu je potřeba v rozcestníku zvolit obrazovku Parser control. Pokud není rozcestník aktuálně zobrazen, je potřeba se do obrazovky rozcestníku nejdříve dostat pomocí tlačítka MENU v záhlaví vizualizace.

Po přesunutí na obrazovku Parser control stačí pro překlad stisknout tlačítko *Interpret program*. Soubor typu G-code se začne automaticky překládat na PLC formát *pathData*. Překlad je vhodné kontrolovat v informační liště Parsed Line. Zde můžeme kontrolovat právě překládanou instrukci výkresu. Každý G-code soubor končí příkazem M30 signalizující konec souboru. Tím si uživatel mj. může zkontrolovat, že soubor byl úspěšně přeložen. Pozornost však musí věnovat také informační liště hned nad lištou Parsed Line. Jedná se o informační lištu Error Parser zobrazující chybovou hlášku překladu. Pokud se během překladu vyskytne chyba, překladač sám vypíše diagnostickou hlášku popisující problém při překladu. Dle něho jsme následně schopni původní soubor upravit tak, aby byl přeložitelný. Signalizaci chyby v překladu hlásí informační buňka *Status parsing*. V případě úspěšného překladu hlásí *Status parsing* hlášku *Program parsed*.

Jakmile je program úspěšně přeložen, spustí se příprava instrukcí pro ovládání přiložení pera, tento proces je možné sledovat pomocí buňky *Status PEN instruction*. Jakmile buňka zahlásí informaci *PEN Ready*, výkres je kompletně připraven ke spuštění. V tom okamžiku tlačítko *MovePath control* změní své zbarvení na zelenou barvu. Tlačítkem *MovePath control* je možné se poté dále přesunout do ovládacího rozhraní pohybu pera a vykreslování spustit.



Obr. 12.11 Řízení překladač souborů G-code

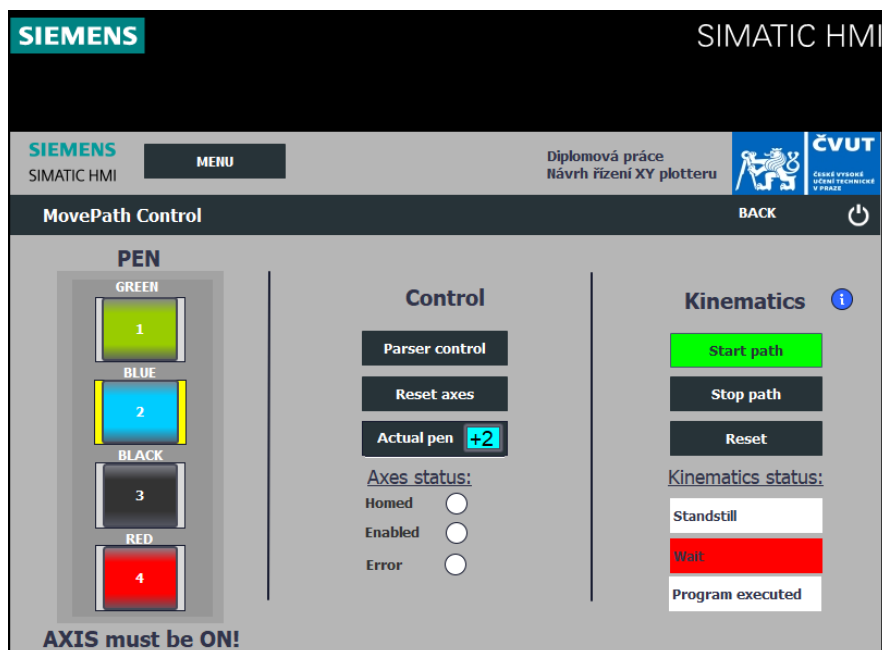
12.7 Spuštění automatického vykreslování

V okamžiku, kdy jsou připraveny instrukce pro ovládání přiložení pera, se na obrazovce Parser control objeví možnost přesunu do ovládacího rozhraní MovePath control. Ta je signalizována zeleným zbarvením tlačítka MovePath control umožňující přímý přestup na obrazovku posledního kroku sekvence, a to spuštění automatického vykreslení.

Pokud se nenacházíme na obrazovce Parser control, opět musíme zpátky na rozcestník a přejít do rozhraní pomocí tlačítka příslušné obrazovky, tedy MovePath control.

Zde najdeme kromě ovládání řízeného pohybu také výměnu pera, již výše popsanou. Řízený pohyb je možné řídit pomocí tří tlačítek v sekci Kinematics. Jedná se o tlačítka *Start path*, *Stop path* a *Reset*. Tlačítka *Start path* a *Stop path* řídíme výkon bloku *LKinCtrl_MC_movePath* zajišťující pohyb pera na základě instrukcí bloku *pathData*. Tlačítka v návrhu tvoří přepínače, a proto je nutné pro sepnutí dané funkce stisknout tlačítko znovu pro navrácení do původního stavu. Význam přepínače je důležitý především u funkce *Stop path*, při kterém dojde k zastavení pohybu, dokud je však přepínač sepnut, pohyb pohonu je zastaven bez ohledu na další příkazy. V případě, že tlačítko *Stop path* přepneme hned zpátky, může být ihned přerušeno jiným příkazem. Tlačítko *Reset* převede blok pohybu do základního stavu. V případě, že pohyb je zastaven tlačítkem *Stop path*, vykreslování je pouze pozastaveno a při opětovném spuštění pohybu pomocí *Start path* řízení pokračuje ve výkonu zvoleného výkresu. V případě, že ve vykreslování nechceme pokračovat, je nutné pohyb také vyresetovat tlačítkem *Reset*.

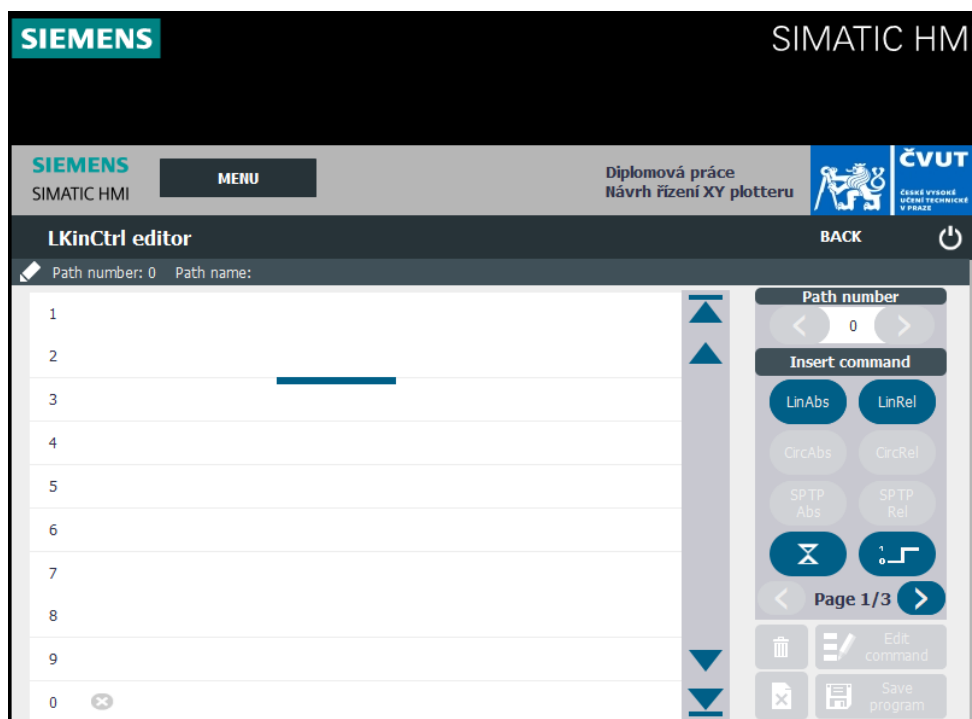
Vykreslování je možné kontrolovat pomocí informačních buněk pod řídicími tlačítky v sekci Kinematics. Najdeme zde informace o stavu osy pohonu, řízení přiložení pera a status bloku *MovePath* obstarávající pohyb.



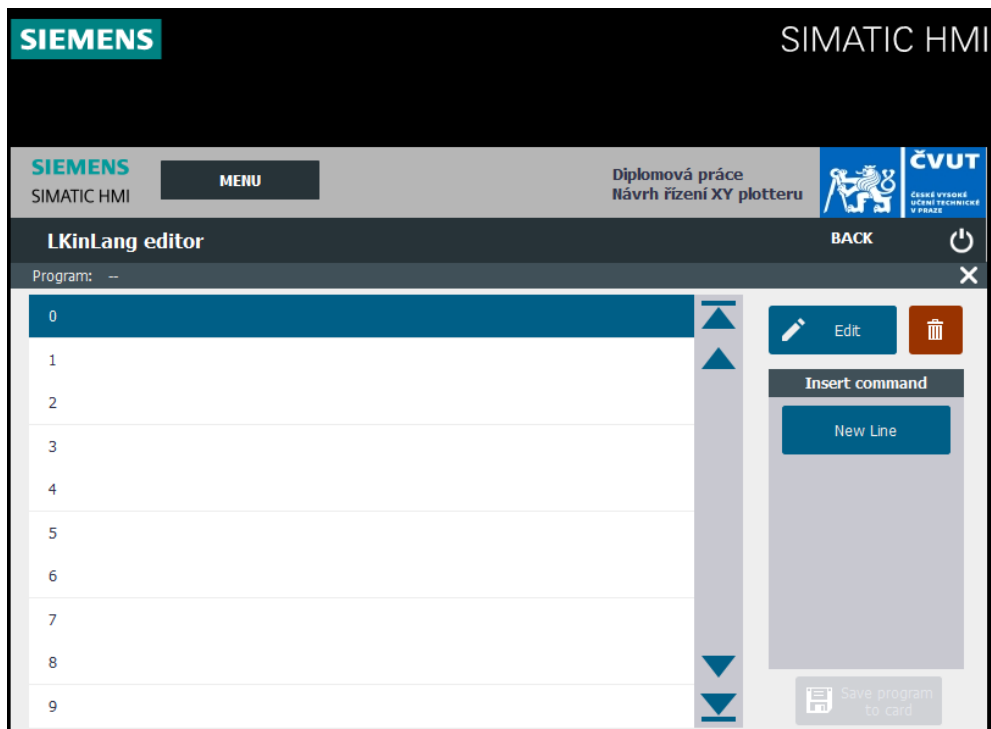
Obr. 12.12 Řízení automatického vykreslování

12.8 Editory k úpravě příkazů

Oba editory knihoven *LKinCtrl* a *LKinLang* slouží k úpravě nahraných výkresových souborů. V případě *LKinCtrl* editoru se jedná o úpravu příkazů formátu *pathData*. V případě *LKinLang* editoru se jedná o úpravu přímo souboru typu G-code. HMI aplikace mají vlastní vestavěné prostředí s možnostmi úpravy a tvorby příkazů pohybu bez dalších nástrojů a nutnosti měnit projekt.



Obr. 12.13 Rozhraní aplikace LKinCtrl Editor



Obr. 12.14 Rozhraní aplikace LKinLang Editor

ZÁVĚR

Předmětem práce byl návrh řízení XY ploteru společně s realizací automatického vykreslování výkresů s využitím technologie CAM. K ovládání přípravku byla použita sestava PLC Simatic zařízení.

Úvodní část práce byla věnována krátkému seznámení s oblastí automatizace se zaměřením na princip a využití PLC zařízení a přiblížení CAD/CAM/CNC technologií počítačového návrhu a výroby. V krátkosti jsem popsal základní strukturu PLC zařízení a nastínil základní principy typické pro PLC programování. Stejně tak byly představeny technologie počítačového návrhu, tedy technologie CAD, dále navazující technologie počítačem asistované výroby CAM a na závěr byl uveden princip počítačem řízených strojů neboli CNC strojů. Programovatelné logické automaty mají často na starosti jednodušší úlohy v rámci procesu. Myšlenka využití PLC zařízení jako náhradu CNC stroje tak vyžaduje hlubší pochopení všech kroků CAD/CAM/CNC technologií.

Další část práce, kap. 4 a kap. 5, popisuje sestavu použitých zařízení a přípravek v podobě XY ploteru. Sestava zařízení se skládá z PLC Siemens S7 - 1500 CPU 1515TF-2 PN, dvou měničů Siemens SINAMICS S210 PN, rozšiřujícího DI/DQ bloku a dvou HMI panelů Siemens KTP400 Basic a Siemens KP900 Comfort. Nedílnou součástí je také představení vývojového prostředí TIA Portal, v němž byl kompletní návrh řízení vytvořen a odladěn a aplikace File Transfer tool, která má na starosti datové spojení externích zařízení a řídicího PLC zařízení. Kapitola 5 pak popisuje přípravek XY ploteru a jeho využití. Zde byly uvedeny specifické faktory návrhu řízení daného přípravku.

Zbývající kapitoly se pak věnují samotnému návrhu jak řízení pohonu přípravku, tak i návrhu obsluhy, řešení ovládání a v neposlední řadě zpracování nadřazené logiky automatického vykreslování včetně konfigurace datového klienta. V kapitole 6 až 11 jsem zdokumentoval všechny důležité kroky, které jsem při vývoji řízení přípravku podnikl. Vzhledem k tomu, že programování PLC Simatic zařízení ve spojení s vývojovým prostředím TIA Portal nabízí využití objektů specifických pro ekosystém spol. Siemens, popsal jsem v kapitole 7 použité nástroje při návrhu, aby bylo možné jejich principu využití lépe porozumět.

V rámci vypracování návrhu logického řízení jsem popsal základní kroky při založení projektu ve vývojovém prostředí TIA Portal. Dále jsem stanovil strukturu požadovaných funkcí řízení ve smyslu, jak by zařízení mělo fungovat. Při analýze požadavků vyplynulo, že je nutné do projektu implementovat externí knihovny *Kinematics Control* a *Kinematics Language* schopné pokročilejšího řízení víceosých pohonů. Následoval popis přípravných kroků pro implementaci nadřazených funkčních bloků z výše uvedených knihoven a jejich konfigurace pro potřeby aplikace.

Dále jsem vytvořil logické řízení obsluhy přípravku. Jedná se především o řízení zásobníku per, které umožňuje automatickou změnu barvy vybraného pera na základě ovládání na HMI panelu. Kromě toho bylo nutné navrhnout několik dílčích řídicích bloků v návaznosti na rozvržení kontrolních spínačů přípravku XY ploteru a použitý typ enkodéru u servomotorů Siemens SINAMCSS210 PN. Absolutní enkodér ve spojení s nedostatkem koncových spínačů představovaly hlavní problém při tzv. *homming* funkci, resp. referování obou os pohonu. Bylo nutné vytvořit manuální sekvenci referování respektující požadavky automatického vykreslování a uživatelskou použitelnost při opakovaném použití.

Realizace automatického vykreslování zahrnuje důmyslné využití funkčních bloků z knihovny *Kinematics Language* společně s vlastním návrhem synchronizovaného řízení přiložení pera v přípravku. V kap. 10 jsem uvedl konkrétní kroky vedoucí k finální podobě funkčního řešení automatického vykreslování na základě datových souborů typu G-code. Návrh umožňuje tvorbu výkresů jak z externího zdroje vloženého do USB slotu HMI panelu KP900 Comfort, tak i skrze spojení s datovým adresářem osobního počítače připojeného k lokální síti.

Následně jsem k logickému řízení navrhl odpovídající vizualizaci. Návrh umožňuje řídit všechny požadované funkce pouze prostřednictvím dvou přiložených HMI panelů bez přítomnosti vývojového prostředí TIA Portal. V kap. 11 byl věnován prostor představení použitých programovacích nástrojů vizualizace ve vývojovém prostředí TIA Portal a přiblížen postup návrhu výsledné podoby ovládání. Během návrhu jsem uplatnil téměř veškeré možnosti vizualizace základních ovládacích prvků, které souvisely s implementovaným řízením XY ploteru.

Poslední část práce patří popisu kompletního ovládání přípravku s implementovaným řízením. V popisu byly uvedeny všechny nezbytné kroky pro zajištění hladkého provozu zařízení. Kvůli časovým možnostem a komplikacím s dostupností jednotlivých zařízení sestavy nebylo možné finální verzi obsluhy, resp. vizualizaci zcela odladit a odzkoušet. Pro zachování bezporuchového provozu bez nutnosti zásahu do chodu zařízení, je proto nutné dodržet správný postup při ovládání přípravku, který je uveden v kap. 12.

Během vypracování návrhu logického řízení se velké množství nápadů ukázalo jako lichých. Dokumentace finální podoby projektu dostatečně nereflktuje množství strávených hodin při ladění funkčního řešení. Návrh programu ve vývojovém prostředí TIA Portal je do jisté míry omezený kvůli zachování bezpečnosti a ochrany samotných zařízení, a proto byla implementace některých funkcí opravdu velkou výzvou. Spojení moderních servomotorů SINAMICS S210 a sestavy spínacích relé a stejnosměrného motoru XY ploteru vyžadovalo vlastní řešení z pohledu synchronizace během automatického vykreslování.

Zadání práce bylo rozšířeno o funkci automatického vykreslování včetně lokálního překladu a zpracování příkazového souboru typu G-code CAM technologie. K tomu bylo implementováno prostředí datového klienta na panelu KP900 Comfort s možností přístupu k adresáři lokálně připojeného externího zařízení. Je tedy možné opakovaně využít přípravek k vytvoření požadovaného výkresu z externě připojeného zařízení.

Navrhnuté logické řízení XY ploteru nyní díky instalaci moderní PLC soustavy zařízení v čele s pokročilým PLC zařízením SIMATIC S7 - 1500 1515TF-2 PN a HMI panelem KP900 Comfort překračuje schopnosti původního zařízení XY ploteru, které v minulosti bylo zřejmě osazeno reléovou logikou umožňující manuální kreslení, a ukazuje programovací možnosti dnešních PLC zařízení.

LITERATURA

- [1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. 1. Praha: BEN - technická literatura, 1999. ISBN 80-860-5658-9.
- [2] PETRUZELLA, Frank D. *Programmable Logic Controllers*. 4. New York: McGraw-Hill, 2011. ISBN 9780073510880.
- [3] ALPHONSUS, Ephrem a Mohammad ABDULLAH. A review on the applications of programmable logic controllers (PLCs). *ScienceDirect* [online]. 2016, (60) [cit. 2020-11-27]. Dostupné z: <https://www-sciencedirect-com.ezproxy.techlib.cz/science/article/pii/S1364032116000551?via%3Dihub>
- [4] KANDRAY, Daniel E. *Programmable Automation Technologies - An Introduction to CNC, Robotics and PLCs*. New York: Industrial Press, 2010. ISBN 978-0-8311-3346-7.
- [5] *Model výtahu řízený PLC Simatic*. Praha, 2021. Dostupné také z: <http://hdl.handle.net/10467/94686>. Bakalářská práce. FEL ČVUT. Vedoucí práce Doc. Ing. Jan Bauer, PhD.
- [6] HANSEN, Dag H. *Programmable Logic Controllers: A Practical Approach to IEC 61131-3 Using CoDeSys*. 1. vyd. Chichester: John Wiley & Sons, Incorporated, 2015. ISBN 9781118949245.
- [7] BOLTON, William. *Programmable logic controllers*. 5th ed. Burlington: Newnes, 2009. ISBN 978-185-6177-511.
- [8] REDDY, Y. Jaganmohan. *Industrial Process Automation Systems - Design and Implementation*. 1. Oxford: Elsevier, 2015. ISBN 978-0-12-800939-0.
- [9] THE EDITORS OF ENCYCLOPAEDIA BRITANNICA. Central processing unit. *Britannica* [online]. Encyclopædia Britannica, 2018 [cit. 2021-01-03]. Dostupné z: <https://www.britannica.com/technology/central-processing-unit>
- [10] *TIA Portal information system*. Siemens. 2023. Dostupné také z: <https://support.industry.siemens.com/cs/document/65601780/tia-portal-an-overview-of-the-most-important-documents-and-links-controller?dti=0&lc=en-CZ>
- [11] HMI (vizualizace a ovládání). In: *PLC automatizace* [online]. Neuvedeno: Neuvedeno, Neuvedeno [cit. 2021-01-28]. Dostupné z: <http://plc-automatizace.cz/knihovna/hmi.htm>
- [12] *Integration of CAD/CAPP/CAM: De Gruyter STEM Ser.* [online]. 1. De Gruyter, Inc., 2018 [cit. 2023-04-12]. ISBN 9783110573091. Dostupné z: <https://ebookcentral.proquest.com/lib/cvut/detail.action?docID=5505277#>
- [13] What is CAM (Computer-Aided Manufacturing)?. In: *Autodesk* [online]. [cit. 2023-04-12]. Dostupné z: <https://www.autodesk.com/products/fusion-360/blog/computer-aided-manufacturing-beginners/>
- [14] Get to Know Your CNC: How to Read G-Code. In: *Makezine* [online]. [cit. 2023-04-12]. Dostupné z: <https://makezine.com/article/digital-fabrication/machining/get-to-know-your-cnc-how-to-read-g-code/>
- [15] Siemens 6ES7515-2UM01-0AB0 6ES75152UM010AB0 CPU pro PLC. In: *Conrad* [online]. [cit. 2023-04-22]. Dostupné z: <https://www.conrad.cz/cs/p/siemens-6es7515-2um01-0ab0-6es75152um010ab0-cpu-pro-plc-2263715.html?refresh=true#productDescription>
- [16] SIMATIC S7-1500T, CPU 1515TF-2 PN. In: *Siemens industry* [online]. [cit. 2023-04-12]. Dostupné z:

- <https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7515-2UM01-0AB0>
- [17] CPU 1515TF-2 PN. In: *Automation Siemens* [online]. [cit. 2023-04-12]. Dostupné z: <https://www.automation.siemens.com/bilddb/search.aspx?multipleObjectTypes=4|70>
- [18] SIMATICS HMI KTP400 Basic. In: *Siemens industry support* [online]. [cit. 2023-04-12]. Dostupné z: <https://support.industry.siemens.com/cs/pd/379924?pdtdi=td&dl=cs&lc=cs-CZ>
- [19] *Přednášky z předmětu BOM14KOP* [online]. 2020 [cit. 2023-04-12]. Dostupné z: <https://moodle.fel.cvut.cz/local/kos/pages/course/info.php?id=7590>. Přednášky FEL ČVUT. Vedoucí práce Jan Bauer, Martin Kozák.
- [20] *SIMATIC HMI devices Basic panels 2nd Generation: Operating instructions* [online]. In: . s. 136 [cit. 2021-02-02]. Dostupné z: https://support.industry.siemens.com/cs/attachments/90114350/hmi_basic_panels_2nd_gen_operating_instructions_enUS_en-US.pdf?download=true
- [21] SIMATIC S7-1500 digital input/output module. In: *Siemens industry mall* [online]. [cit. 2023-04-12]. Dostupné z: <https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7523-1BL00-0AA0>
- [22] *Data sheet SITOP PSU100S* [online]. In: . s. 4 [cit. 2023-04-12]. Dostupné z: <https://www.jork.shop/priloha.php?ak=96701>
- [23] SIMATIC HMI KP900 Comfort. In: *Siemens industry support* [online]. [cit. 2023-04-12]. Dostupné z: <https://support.industry.siemens.com/cs/pd/469655?pdtdi=td&dl=en&lc=en-CR>
- [24] SIMATIC HMI KP900 Comfort. In: *Automation Siemens* [online]. [cit. 2023-04-12]. Dostupné z: https://www.automation.siemens.com/bilddb/interfaces/InterfaceImageDB.aspx/GetImageVariant?objectkey=P_ST80_XX_02165&imagevariantid=16&lang=XX&interfaceuserid=SIOS
- [25] *SINAMICS S210 servo drive system* [online]. 666 [cit. 2023-04-12]. Dostupné z: https://cache.industry.siemens.com/dl/files/824/109771824/att_999097/v1/S210_1FK2_op_instr_092019_en-US.pdf
- [26] Technology overview SINAMICS S210. In: *Siemens* [online]. [cit. 2023-04-12]. Dostupné z: <https://www.siemens.com/global/en/products/drives/sinamics/low-voltage-converters/servo-converter/sinamics-s210.html#Technologyoverview>
- [27] Stupeň krytí. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2023-04-12]. Dostupné z: https://cs.wikipedia.org/wiki/Stupe%C5%88_kryt%C3%AD
- [28] *Údajový list pro SIMOTICS S-1FK2*. Siemens.
- [29] Operátorské panely SIMATIC HMI Basic. In: *SIEMENS* [online]. Neuvedeno: Neuvedeno, 2019 [cit. 2021-01-28]. Dostupné z: <https://new.siemens.com/cz/cs/products/automation/simatic-hmi/panels/basic-panels.html>
- [30] Totally Integrated Automation Portal: Vaše brána do světa automatizace a digitalizace. In: *SIEMENS* [online]. Neuvedeno: Neuvedeno, 2019 [cit. 2021-01-28].

- Dostupné z: <https://new.siemens.com/cz/cs/products/automation/industry-software/automatizacni-software/tia-portal.html>
- [31] *Library Kinematics Language*. 2023. Dostupné také z: <https://support.industry.siemens.com/cs/document/109767009/simatic-s7-1500t-kinematics-language?dti=0&lc=en-CR>
- [32] Technology objects. In: *Siemens industry cache* [online]. [cit. 2023-04-12]. Dostupné z: https://cache.industry.siemens.com/dl/files/134/109743134/att_928448/v2/109743134_S7-1500T_TechnologyObjects_DOC_v10_en.pdf
- [33] *Řízení elektrických pohonů*. 2020. Přednáška. ČVUT FEL.
- [34] TO Kinematics. In: *Siemens industry support* [online]. [cit. 2023-04-12]. Dostupné z: <https://support.industry.siemens.com/cs/document/109792862/simatic-s7-1200-handling-library-for-pick-place-applications?dti=0&lc=en-CR>
- [35] *Library Kinematics Control*. 2023. Dostupné také z: <https://support.industry.siemens.com/cs/document/109755891/simatic-s7-1500t-kinematics-control?dti=0&lc=en-CZ>

PŘÍLOHA A: TECHNICKÉ ÚDAJE ZAŘÍZENÍ SESTAVY

A.1 SIMATIC S7-1500 CPU 1515TF-2 PN

Tab. Technické údaje zařízení Siemens SIMATIC S7-1500 CPU 1515TF-2 PN (upraveno)¹

General information	
Product type designation	CPU 1515TF-2 PN
HW functional status	FS04
Firmware version	V2.9
Product function	
• I&M data	Yes; I&M0 to I&M3
• Isochronous mode	Yes; Distributed and central; with minimum OB 6x cycle of 500 µs (distributed) and 1 ms (central)
Engineering with	
• STEP 7 TIA Portal configurable/integrated from version	V17 (FW V2.9) / V14 (FW V2.1) or higher
Configuration control	
via dataset	Yes
Display	
Screen diagonal [cm]	6.1 cm
Control elements	
Number of keys	6
Mode selector switch	1
Supply voltage	
Rated value (DC)	24 V
permissible range, lower limit (DC)	19.2 V
permissible range, upper limit (DC)	28.8 V
Reverse polarity protection	Yes
Mains buffering	
• Mains/voltage failure stored energy time	5 ms
• Repeat rate, min.	1/s
Input current	
Current consumption (rated value)	0.8 A
Inrush current, max.	2.4 A; Rated value
I^2t	0.02 A ² ·s
Power	
Infeed power to the backplane bus	12 W
Power consumption from the backplane bus (balanced)	6.2 W
Power loss	
Power loss, typ.	6.3 W

¹ 6ES7515-2UM01-0AB0 CPU 1515TF-2 PN. Siemens industry support [online]. 2022 [cit. 2023-04-12]. Dostupné z: <https://support.industry.siemens.com/cs/pd/972780?pdtdi=td&dl=cs&lc=cs-CZ>

Memory	
Number of slots for SIMATIC memory card	1
SIMATIC memory card required	Yes
Work memory	
<ul style="list-style-type: none"> integrated (for program) 	750 kbyte
<ul style="list-style-type: none"> integrated (for data) 	3 Mbyte
Load memory	
<ul style="list-style-type: none"> Plug-in (SIMATIC Memory Card), max. 	32 Gbyte
Backup	
<ul style="list-style-type: none"> maintenance-free 	Yes
CPU processing times	
for bit operations, typ.	30 ns
for word operations, typ.	36 ns
for fixed point arithmetic, typ.	48 ns
for floating point arithmetic, typ.	192 ns
CPU-blocks	
Number of elements (total)	8 000; Blocks (OB, FB, FC, DB) and UDTs
DB	
<ul style="list-style-type: none"> Number range 	1 ... 60 999; subdivided into: number range that can be used by the user: 1 ... 59 999, and number range of DBs created via SFC 86: 60 000 ... 60 999
<ul style="list-style-type: none"> Size, max. 	3 Mbyte; For DBs with absolute addressing, the max. size is 64 KB
FB	
<ul style="list-style-type: none"> Number range 	0 ... 65 535
<ul style="list-style-type: none"> Size, max. 	500 kbyte
FC	
<ul style="list-style-type: none"> Number range 	0 ... 65 535
<ul style="list-style-type: none"> Size, max. 	500 kbyte
OB	
<ul style="list-style-type: none"> Size, max. 	500 kbyte
<ul style="list-style-type: none"> Number of free cycle OBs 	100
<ul style="list-style-type: none"> Number of time alarm OBs 	20
<ul style="list-style-type: none"> Number of delay alarm OBs 	20
<ul style="list-style-type: none"> Number of cyclic interrupt OBs 	20; With minimum OB 3x cycle of 500 μ s
<ul style="list-style-type: none"> Number of process alarm OBs 	50
<ul style="list-style-type: none"> Number of DPV1 alarm OBs 	3
<ul style="list-style-type: none"> Number of isochronous mode OBs 	2
<ul style="list-style-type: none"> Number of technology synchronous alarm OBs 	2
<ul style="list-style-type: none"> Number of startup OBs 	100
<ul style="list-style-type: none"> Number of asynchronous error OBs 	4
<ul style="list-style-type: none"> Number of synchronous error OBs 	2
<ul style="list-style-type: none"> Number of diagnostic alarm OBs 	1

Nesting depth	
• per priority class	24; Up to 8 possible for F-blocks
Counters, timers and their retentivity	
S7 counter	
• Number	2 048
Retentivity	
— adjustable	Yes
IEC counter	
• Number	Any (only limited by the main memory)
Retentivity	
— adjustable	Yes
S7 times	
• Number	2 048
Retentivity	
— adjustable	Yes
IEC timer	
• Number	Any (only limited by the main memory)
Retentivity	
— adjustable	Yes
Data areas and their retentivity	
Retentive data area (incl. timers, counters, flags), max.	512 kbyte; In total; available retentive memory for bit memories, timers, counters, DBs, and technology data (axes): 472 KB
Extended retentive data area (incl. timers, counters, flags), max.	3 Mbyte; When using PS 6 0W 24/48/60 V DC HF
Flag	
• Size, max.	16 kbyte
• Number of clock memories	8; 8 clock memory bit, grouped into one clock memory byte
Data blocks	
• Retentivity adjustable	Yes
• Retentivity preset	No
Local data	
• per priority class, max.	64 kbyte; max. 16 KB per block
Address area	
Number of IO modules	8 192; max. number of modules / submodules
I/O address area	
• Inputs	32 kbyte; All inputs are in the process image
• Outputs	32 kbyte; All outputs are in the process image
per integrated IO subsystem	
— Inputs (volume)	8 kbyte
— Outputs (volume)	8 kbyte
per CM/CP	

— Inputs (volume)	8 kbyte
— Outputs (volume)	8 kbyte
Subprocess images	
• Number of subprocess images, max.	32
Hardware configuration	
Number of distributed IO systems	64; A distributed I/O system is characterized not only by the integration of distributed I/O via PROFINET or PROFIBUS communication modules, but also by the connection of I/O via AS-i master modules or links (e.g. IE/PB-Link)
Number of DP masters	
• Via CM	8; A maximum of 8 CMs/CPs (PROFIBUS, PROFINET, Ethernet) can be inserted in total
Number of IO Controllers	
• integrated	2
• Via CM	8; A maximum of 8 CMs/CPs (PROFIBUS, PROFINET, Ethernet) can be inserted in total
Rack	
• Modules per rack, max.	32; CPU + 31 modules
• Number of lines, max.	1
PtP CM	
• Number of PtP CMs	the number of connectable PtP CMs is only limited by the number of available slots
Time of day	
Clock	
• Type	Hardware clock
• Backup time	6 wk; At 40 °C ambient temperature, typically
• Deviation per day, max.	10 s; Typ.: 2 s
Operating hours counter	
• Number	16
Clock synchronization	
• supported	Yes
• in AS, master	Yes
• in AS, slave	Yes
• on Ethernet via NTP	Yes
Interfaces	
Number of PROFINET interfaces	2
1. Interface	
Interface types	
• RJ 45 (Ethernet)	Yes; X1
• Number of ports	2
• integrated switch	Yes

Protocols	
• IP protocol	Yes; IPv4
• PROFINET IO Controller	Yes
• PROFINET IO Device	Yes
• SIMATIC communication	Yes
• Open IE communication	Yes; Optionally also encrypted
• Web server	Yes
• Media redundancy	Yes
PROFINET IO Controller	
Services	
— PG/OP communication	Yes
— Isochronous mode	Yes
— Direct data exchange	Yes; Requirement: IRT and isochronous mode (MRPD optional)
— IRT	Yes
— PROFIenergy	Yes; per user program
— Prioritized startup	Yes; Max. 32 PROFINET devices
— Number of connectable IO Devices, max.	256; In total, up to 1 000 distributed I/O devices can be connected via AS-i, PROFIBUS or PROFINET
— Of which IO devices with IRT, max.	64
— Number of connectable IO Devices for RT, max.	256
— of which in line, max.	256
— Number of IO Devices that can be simultaneously activated/deactivated, max.	8; in total across all interfaces
— Number of IO Devices per tool, max.	8
— Updating times	The minimum value of the update time also depends on communication share set for PROFINET IO, on the number of IO devices, and on the quantity of configured user data
Update time for IRT	
— for send cycle of 250 µs	250 µs to 4 ms; Note: In the case of IRT with isochronous mode, the minimum update time of 500 µs of the isochronous OB is decisive
— for send cycle of 500 µs	500 µs to 8 ms
— for send cycle of 1 ms	1 ms to 16 ms
— for send cycle of 2 ms	2 ms to 32 ms
— for send cycle of 4 ms	4 ms to 64 ms
— With IRT and parameterization of "odd" send cycles	Update time = set "odd" send clock (any multiple of 125 µs: 375 µs, 625 µs ... 3 875 µs)
Update time for RT	
— for send cycle of 250 µs	250 µs to 128 ms
— for send cycle of 500 µs	500 µs to 256 ms
— for send cycle of 1 ms	1 ms to 512 ms
— for send cycle of 2 ms	2 ms to 512 ms
— for send cycle of 4 ms	4 ms to 512 ms

PROFINET IO Device	
Services	
— PG/OP communication	Yes
— Isochronous mode	No
— IRT	Yes
— PROFlenergy	Yes; per user program
— Shared device	Yes
— Number of IO Controllers with shared device, max.	4
— activation/deactivation of I-devices	Yes; per user program
— Asset management record	Yes; per user program
2. Interface	
Interface types	
• RJ 45 (Ethernet)	Yes; X2
• Number of ports	1
• integrated switch	No
Protocols	
• IP protocol	Yes; IPv4
• PROFINET IO Controller	Yes
• PROFINET IO Device	Yes
• SIMATIC communication	Yes
• Open IE communication	Yes; Optionally also encrypted
• Web server	Yes
• Media redundancy	No
PROFINET IO Controller	
Services	
— PG/OP communication	Yes
— Isochronous mode	No
— Direct data exchange	No
— IRT	No
— PROFlenergy	Yes; per user program
— Prioritized startup	No
— Number of connectable IO Devices, max.	32; In total, up to 1 000 distributed I/O devices can be connected via AS-i, PROFIBUS or PROFINET
— Number of connectable IO Devices for RT, max.	32
— of which in line, max.	32
— Number of IO Devices that can be simultaneously activated/deactivated, max.	8; in total across all interfaces
— Number of IO Devices per tool, max.	8
— Updating times	The minimum value of the update time also depends on communication share set for PROFINET IO, on the number of IO devices, and on the quantity of configured user data
Update time for RT	
— for send cycle of 1 ms	1 ms to 512 ms

PROFINET IO Device	
Services	
— PG/OP communication	Yes
— Isochronous mode	No
— IRT	No
— PROFlenergy	Yes; per user program
— Prioritized startup	No
— Shared device	Yes
— Number of IO Controllers with shared device, max.	4
— activation/deactivation of I-devices	Yes; per user program
— Asset management record	Yes; per user program
Interface types	
RJ 45 (Ethernet)	
• 100 Mbps	Yes
• Autonegotiation	Yes
• Autocrossing	Yes
• Industrial Ethernet status LED	Yes
Protocols	
PROFIsafe	Yes
Number of connections	
• Number of connections, max.	192; via integrated interfaces of the CPU and connected CPs / CMs
• Number of connections reserved for ES/HMI/web	10
• Number of connections via integrated interfaces	108
• Number of S7 routing paths	16
Redundancy mode	
• H-Sync forwarding	Yes
Media redundancy	
— Media redundancy	only via 1st interface (X1)
— MRP	Yes; MRP Automanager according to IEC 62439-2 Edition 2.0, MRP Manager; MRP Client
— MRP interconnection, supported	Yes; as MRP ring node according to IEC 62439-2 Edition 3.0
— MRPD	Yes; Requirement: IRT
— Switchover time on line break, typ.	200 ms; For MRP, bumpless for MRPD
— Number of stations in the ring, max.	50
SIMATIC communication	
• PG/OP communication	Yes; encryption with TLS V1.3 pre-selected
• S7 routing	Yes
• S7 communication, as server	Yes
• S7 communication, as client	Yes
• User data per job, max.	See online help (S7 communication, user data size)
Open IE communication	

<ul style="list-style-type: none"> ● TCP/IP <ul style="list-style-type: none"> — Data length, max. — several passive connections per port, supported ● ISO-on-TCP (RFC1006) <ul style="list-style-type: none"> — Data length, max. ● UDP <ul style="list-style-type: none"> — Data length, max. — UDP multicast ● DHCP ● DNS ● SNMP ● DCP ● LLDP ● Encryption 	<p>Yes</p> <p>64 kbyte</p> <p>Yes</p> <p>Yes</p> <p>64 kbyte</p> <p>Yes</p> <p>2 kbyte; 1 472 bytes for UDP broadcast</p> <p>Yes; Max. 5 multicast circuits</p> <p>Yes</p> <p>Yes</p> <p>Yes</p> <p>Yes</p> <p>Yes</p> <p>Yes; Optional</p>
Web server	
<ul style="list-style-type: none"> ● HTTP ● HTTPS 	<p>Yes; Standard and user pages</p> <p>Yes; Standard and user pages</p>
OPC UA	
<ul style="list-style-type: none"> ● Runtime license required ● OPC UA Client <ul style="list-style-type: none"> — Application authentication — Security policies — User authentication — Number of connections, max. — Number of nodes of the client interfaces, recommended max. — Number of elements for one call of OPC-UA_NodeGetHandleList/OPC-UA_ReadList/OPC-UA_WriteList, max. — Number of elements for one call of OPC-UA_NameSpaceGetIndexList, max. — Number of elements for one call of OPC-UA_MethodGetHandleList, max. — Number of simultaneous calls of the client instructions for session management, per connection, max. — Number of simultaneous calls of the client instructions for data access, per connection, max. — Number of registerable nodes, max. — Number of registerable method calls of OPC-UA_MethodCall, max. — Number of inputs/outputs when calling OPC-UA_MethodCall, max. ● OPC UA Server <ul style="list-style-type: none"> — Application authentication — Security policies — User authentication 	<p>Yes; "Medium" license required</p> <p>Yes</p> <p>Yes</p> <p>Available security policies: None, Basic128Rsa15, Basic256Rsa15, Basic256Sha256</p> <p>"anonymous" or by user name & password</p> <p>10</p> <p>2 000</p> <p>300</p> <p>20</p> <p>100</p> <p>1</p> <p>5</p> <p>5 000</p> <p>100</p> <p>20</p> <p>Yes; Data access (read, write, subscribe), method call, custom address space</p> <p>Yes</p> <p>Available security policies: None, Basic128Rsa15, Basic256Rsa15, Basic256Sha256</p> <p>"anonymous" or by user name & password</p>

— GDS support (certificate management)	Yes
— Number of sessions, max.	48
— Number of accessible variables, max.	100 000
— Number of registerable nodes, max.	20 000
— Number of subscriptions per session, max.	20
— Sampling interval, min.	100 ms
— Publishing interval, min.	200 ms
— Number of server methods, max.	50
— Number of inputs/outputs per server method, max.	20
— Number of monitored items, recommended max.	2 000; for 1 s sampling interval and 1 s send interval
— Number of server interfaces, max.	10 of each "Server interfaces" / "Companion specification" type and 20 of the type "Reference namespace"
— Number of nodes for user-defined server interfaces, max.	5 000
• Alarms and Conditions	Yes
— Number of program alarms	200
— Number of alarms for system diagnostics	100
Further protocols	
• MODBUS	Yes; MODBUS TCP
Isochronous mode	
Equidistance	Yes
S7 message functions	
Number of login stations for message functions, max.	64
Program alarms	Yes
Number of configurable program messages, max.	10 000; Program messages are generated by the "Program_Alarm" block, ProDiag or GRAPH
Number of loadable program messages in RUN, max.	5 000
Number of simultaneously active program alarms	
• Number of program alarms	800
• Number of alarms for system diagnostics	200
• Number of alarms for motion technology objects	160
Test commissioning functions	
Joint commission (Team Engineering)	Yes; Parallel online access possible for up to 8 engineering systems
Status block	Yes; Up to 8 simultaneously (in total across all ES clients)
Single step	No
Number of breakpoints	8
Status/control	
• Status/control variable	Yes; without fail-safe
• Variables	inputs/outputs, bit memories, DBs, peripheral I/Os (without fail-safe), times, counters
• Number of variables, max.	
— of which status variables, max.	200; per job

— of which control variables, max.	200; per job
Forcing	
• Forcing	Yes; without fail-safe
• Forcing, variables	peripheral inputs/outputs (without fail-safe)
• Number of variables, max.	200
Diagnostic buffer	
• present	Yes
• Number of entries, max.	3 200
— of which powerfail-proof	500
Traces	
• Number of configurable Traces	4; Up to 512 KB of data per trace are possible
Interrupts/diagnostics/status information	
Diagnostics indication LED	
• RUN/STOP LED	Yes
• ERROR LED	Yes
• MAINT LED	Yes
• Connection display LINK TX/RX	Yes
Supported technology objects	
Motion Control	Yes; Note: The number of technology objects affects the cycle time of the PLC program; selection guide via the TIA Selection Tool
• Number of available Motion Control resources for technology objects	2 400
• Required Motion Control resources	
— per speed-controlled axis	40
— per positioning axis	80
— per synchronous axis	160
— per external encoder	80
— per output cam	20
— per cam track	160
— per probe	40
• Number of available Extended Motion Control resources for technology objects	120
• Required Extended Motion Control resources	
— per cam (1 000 points and 50 segments)	2
— per cam (10 000 points and 50 segments)	20
— for each set of kinematics	30
— Per leading axis proxy	3
• Positioning axis	
— Number of positioning axes at motion control cycle of 4 ms (typical value)	7
— Number of positioning axes at motion control cycle of 8 ms (typical value)	14
Controller	
• PID_Compact	Yes; Universal PID controller with integrated optimization

<ul style="list-style-type: none"> • PID_3Step • PID-Temp 	<p>Yes; PID controller with integrated optimization for valves</p> <p>Yes; PID controller with integrated optimization for temperature</p>
<p>Counting and measuring</p> <ul style="list-style-type: none"> • High-speed counter 	Yes
Standards, approvals, certificates	
Highest safety class achievable in safety mode	
<ul style="list-style-type: none"> • Performance level according to ISO 13849-1 • SIL acc. to IEC 61508 	<p>PLe</p> <p>SIL 3</p>
Probability of failure (for service life of 20 years and repair time of 100 hours)	
— Low demand mode: PFDavg in accordance with SIL3	< 2.00E-05
— High demand/continuous mode: PFH in accordance with SIL3	< 1.00E-09
Ambient conditions	
Ambient temperature during operation	
<ul style="list-style-type: none"> • horizontal installation, min. • horizontal installation, max. • vertical installation, min. • vertical installation, max. 	<p>0 °C</p> <p>60 °C; Display: 50 °C, at an operating temperature of typically 50 °C, the display is switched off</p> <p>0 °C</p> <p>40 °C; Display: 40 °C, at an operating temperature of typically 40 °C, the display is switched off</p>
Ambient temperature during storage/transportation	
<ul style="list-style-type: none"> • min. • max. 	<p>-40 °C</p> <p>70 °C</p>
Altitude during operation relating to sea level	
<ul style="list-style-type: none"> • Installation altitude above sea level, max. 	5 000 m; Restrictions for installation altitudes > 2 000 m, see manual
configuration / header	
configuration / programming / header	
Programming language	
— LAD	Yes; incl. failsafe
— FBD	Yes; incl. failsafe
— STL	Yes
— SCL	Yes
— GRAPH	Yes
Know-how protection	
<ul style="list-style-type: none"> • User program protection/password protection • Copy protection • Block protection 	<p>Yes</p> <p>Yes</p> <p>Yes</p>
Access protection	
<ul style="list-style-type: none"> • protection of confidential configuration data • Password for display • Protection level: Write protection • Protection level: Read/write protection 	<p>Yes</p> <p>Yes</p> <p>Yes; Specific write protection both for Standard and for Failsafe</p> <p>Yes</p>

<ul style="list-style-type: none"> • Protection level: Write protection for Failsafe 	Yes
<ul style="list-style-type: none"> • Protection level: Complete protection 	Yes
programming / cycle time monitoring / header	
<ul style="list-style-type: none"> • lower limit 	adjustable minimum cycle time
<ul style="list-style-type: none"> • upper limit 	adjustable maximum cycle time
Dimensions	
Width	70 mm
Height	147 mm
Depth	129 mm
Weights	
Weight, approx.	830 g

A.2 Siemens SIMATIC HMI KTP400 Basic

Tab. Technické údaje zařízení Siemens SIMATIC HMI KTP400 Basic (upraveno)²

General information	
Product type designation	KTP400 Basic
Display	
Design of display	TFT widescreen display, LED backlighting
Screen diagonal	4.3 in
Display width	95 mm
Display height	53.9 mm
Number of colors	65 536
Resolution (pixels)	
<ul style="list-style-type: none"> • Horizontal image resolution 	480 pixel
<ul style="list-style-type: none"> • Vertical image resolution 	272 pixel
Backlighting	
<ul style="list-style-type: none"> • MTBF backlighting (at 25 °C) 	20 000 h
<ul style="list-style-type: none"> • Backlight dimmable 	Yes
Control elements	
Keyboard fonts	
<ul style="list-style-type: none"> • Function keys <ul style="list-style-type: none"> — Number of function keys — Number of function keys with LEDs 	4 0
<ul style="list-style-type: none"> • Keys with LED 	No
<ul style="list-style-type: none"> • System keys 	No
<ul style="list-style-type: none"> • Numeric keyboard 	Yes; Onscreen keyboard
<ul style="list-style-type: none"> • alphanumeric keyboard 	Yes; Onscreen keyboard
Touch operation	
<ul style="list-style-type: none"> • Design as touch screen 	Yes; Analog-resistive

² 6AV2123-2DB03-0AX0 SIMATIC HMI KTP400 Basic. Siemens industry support [online]. 2021 [cit. 2023-04-12]. Dostupné z: <https://support.industry.siemens.com/cs/pd/379924?pdtd=td&dl=en&lc=en-CZ>

Installation type/mounting	
Mounting position	vertical
Wall mounting/direct mounting	No
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)	125 mA
Starting current inrush I ² t	0.2 A ² ·s
Power	
Active power input, typ.	3 W
Processor	
Processor type	ARM
Memory	
Flash	Yes
RAM	Yes
Memory available for user data	10 Mbyte
Type of output	
Acoustics	
• Buzzer	Yes
• Speaker	No
Time of day	
Clock	
• Hardware clock (real-time)	Yes
• Software clock	Yes
• retentive	Yes; Back-up duration typically 6 weeks
• synchronizable	Yes
Interfaces	
Number of industrial Ethernet interfaces	1
Number of RS 485 interfaces	0
Number of RS 422 interfaces	0
Number of RS 232 interfaces	0
Number of USB interfaces	1; Up to 16 GB
Number of 20 mA interfaces (TTY)	0
Number of parallel interfaces	0
Number of other interfaces	0
Number of SD card slots	0

With software interfaces	No
Industrial Ethernet	
<ul style="list-style-type: none"> Industrial Ethernet status LED 	2
Protocols	
PROFINET	Yes
Supports protocol for PROFINET IO	No
IRT	No
PROFIBUS	No
EtherNet/IP	Yes
MPI	No
Protocols (Ethernet)	
<ul style="list-style-type: none"> TCP/IP DHCP SNMP DCP LLDP 	Yes Yes Yes Yes Yes
WEB characteristics	
<ul style="list-style-type: none"> HTTP HTML 	No No
Redundancy mode	
Media redundancy	
— MRP	No
Further protocols	
<ul style="list-style-type: none"> CAN MODBUS 	No Yes; Modicon (MODBUS TCP/IP)
Interrupts/diagnostics/status information	
Diagnoses	
<ul style="list-style-type: none"> Diagnostic information readable 	No
EMC	
Emission of radio interference acc. to EN 55 011	
<ul style="list-style-type: none"> Limit class A, for use in industrial areas Limit class B, for use in residential areas 	Yes No
Degree and class of protection	
IP (at the front)	IP65
IP (rear)	IP20
NEMA (front)	
<ul style="list-style-type: none"> Enclosure Type 4 at the front Enclosure Type 4x at the front 	Yes Yes
Standards, approvals, certificates	
CE mark	Yes
cULus	Yes
RCM (formerly C-TICK)	Yes

KC approval	Yes
Use in hazardous areas	
• ATEX Zone 2	No
• ATEX Zone 22	No
• IECEx Zone 2	No
• IECEx Zone 22	No
• cULus Class I Zone 1	No
• cULus Class I Zone 2, Division 2	No
• FM Class I Division 2	No
Marine approval	
• Germanischer Lloyd (GL)	Yes
• American Bureau of Shipping (ABS)	Yes
• Bureau Veritas (BV)	Yes
• Det Norske Veritas (DNV)	Yes
• Lloyds Register of Shipping (LRS)	Yes
• Nippon Kaiji Kyokai (Class NK)	Yes
• Polski Rejestr Statkow (PRS)	No
• Chinese Classification Society (CCS)	No
Ambient conditions	
Suited for indoor use	Yes
Suited for outdoor use	No
Ambient temperature during operation	
Operation (vertical installation)	
— For vertical installation, min.	0 °C
— For vertical installation, max.	50 °C
Operation (max. tilt angle)	
— At maximum tilt angle, min.	0 °C
— At maximum tilt angle, max.	40 °C
Operation (vertical installation, portrait format)	
— For vertical installation, min.	0 °C
— For vertical installation, max.	40 °C
Operation (max. tilt angle, portrait format)	
— At maximum tilt angle, min.	0 °C
— At maximum tilt angle, max.	35 °C
Ambient temperature during storage/transportation	
• min.	-20 °C
• max.	60 °C
Relative humidity	
• Operation, max.	90 %; no condensation
Operating systems	
proprietary	Yes
pre-installed operating system	
• Windows CE	No

configuration / header	
Message indicator	Yes
Alarm system (incl. buffer and acknowledgment)	Yes
Process value display (output)	Yes
Process value default (input) possible	Yes
Recipe management	Yes
Configuration software	
<ul style="list-style-type: none"> ● STEP 7 Basic (TIA Portal) 	Yes; via integrated WinCC Basic (TIA Portal)
<ul style="list-style-type: none"> ● STEP 7 Professional (TIA Portal) 	Yes; via integrated WinCC Basic (TIA Portal)
<ul style="list-style-type: none"> ● WinCC flexible Compact 	No
<ul style="list-style-type: none"> ● WinCC flexible Standard 	No
<ul style="list-style-type: none"> ● WinCC flexible Advanced 	No
<ul style="list-style-type: none"> ● WinCC Basic (TIA Portal) 	Yes
<ul style="list-style-type: none"> ● WinCC Comfort (TIA Portal) 	Yes
<ul style="list-style-type: none"> ● WinCC Advanced (TIA Portal) 	Yes
<ul style="list-style-type: none"> ● WinCC Professional (TIA Portal) 	Yes
Languages	
Online languages	
<ul style="list-style-type: none"> ● Number of online/runtime languages 	10
Project languages	
<ul style="list-style-type: none"> ● Languages per project 	32
Functionality under WinCC (TIA Portal)	
Libraries	Yes
Applications/options	
<ul style="list-style-type: none"> ● Web browser 	Yes
<ul style="list-style-type: none"> ● SIMATIC WinCC Sm@rtServer 	Yes; Available with WinCC (TIA Portal) V14 or higher
Number of Visual Basic Scripts	No
Task planner	Yes
<ul style="list-style-type: none"> ● time-controlled 	No
<ul style="list-style-type: none"> ● task-controlled 	Yes
Help system	Yes
<ul style="list-style-type: none"> ● Number of characters per info text 	500
Message system	
<ul style="list-style-type: none"> ● Number of alarm classes 	32
<ul style="list-style-type: none"> ● Bit messages <ul style="list-style-type: none"> — Number of bit messages 	1 000
<ul style="list-style-type: none"> ● Analog messages <ul style="list-style-type: none"> — Number of analog messages 	25
<ul style="list-style-type: none"> ● S7 alarm number procedure 	No
<ul style="list-style-type: none"> ● System messages HMI 	Yes
<ul style="list-style-type: none"> ● System event, more (SIMATIC S7, SINUMERIK, SIMOTION, ...) 	Yes; System message buffer of the SIMATIC S7-1200 and S7-1500

• Number of characters per message	80
• Number of process values per message	8
• Acknowledgment groups	Yes
• Message indicator	Yes
• Message buffer	
— Number of entries	256
— Circulating buffer	Yes
— retentive	Yes
— maintenance-free	Yes
Recipe management	
• Number of recipes	50
• Data records per recipe	100
• Entries per data record	100
• Size of internal recipe memory	256 kbyte
• Recipe memory expandable	No
Variables	
• Number of variables per device	800
• Number of variables per screen	100
• Limit values	Yes
• Multiplexing	Yes
• Structures	No
• Arrays	Yes
Images	
• Number of configurable images	250
• Permanent window/default	Yes
• Global image	Yes
• Pop-up images	No
• Slide-in images	No
• Image selection by PLC	Yes
• Image number in the PLC	Yes
Image objects	
• Number of objects per image	100
• Text fields	Yes
• I/O fields	Yes
• Graphic I/O fields (graphics list)	Yes
• Symbolic I/O fields (text list)	Yes
• Date/time fields	Yes
• Switches	Yes
• Buttons	Yes
• Graphic display	Yes
• Icons	Yes
• Geometric objects	Yes
Complex image objects	
• Number of complex objects per screen	10

• Alarm view	Yes
• Trend view	Yes
• User view	Yes
• Status/control	No
• Sm@rtClient view	No
• Recipe view	Yes
• f(x) trend view	No
• System diagnostics view	Yes; System message buffer of the SIMATIC S7-1200 and S7-1500
• Media Player	No
• HTML browser	Yes
• PDF display	No
• IP camera display	No
• Bar graphs	Yes
• Sliders	No
• Pointer instruments	No
• Analog/digital clock	No
Lists	
• Number of text lists per project	300
• Number of entries per text list	100
• Number of graphics lists per project	100
• Number of entries per graphics list	100
Archiving	
• Number of archives per device	2; One message and one process value archive
• Number of entries per archive	10 000
• Message archive	Yes
• Process value archive	Yes
• Archiving methods	
— Sequential archive	Yes
— Short-term archive	Yes
• Memory location	
— Memory card	No
— USB memory	Yes
— Ethernet	No
• Data storage format	
— CSV	No
— TXT	Yes
— RDB	No
Security	
• Number of user groups	50
• Number of user rights	32
• Number of users	50
• Password export/import	Yes

• SIMATIC Logon	No
Character sets	
• Keyboard fonts	
— US English	Yes
Transfer (upload/download)	
• MPI/PROFIBUS DP	No
• USB	No
• Ethernet	Yes
• using external storage medium	Yes
Process coupling	
• S7-1200	Yes
• S7-1500	Yes
• S7-200	Yes
• S7-300/400	Yes
• LOGO!	Yes
• WinAC	Yes
• SINUMERIK	Yes; No access to NCK data
• SIMOTION	Yes
• Allen Bradley (EtherNet/IP)	Yes
• Allen Bradley (DF1)	No
• Mitsubishi (MC TCP/IP)	Yes
• Mitsubishi (FX)	No
• OMRON (FINS TCP)	No
• OMRON (LINK/Multilink)	No
• Modicon (Modbus TCP/IP)	Yes
• Modicon (Modbus)	No
Service tools/configuration aids	
• Backup/Restore manually	Yes
• Backup/Restore automatically	No
• Simulation	Yes
• Device switchover	Yes
Peripherals/Options	
Printer	No
SIMATIC HMI MM memory card: Multi Media Card	No
SIMATIC HMI SD memory card: Secure Digital memory card	No
SIMATIC HMI CF memory card Compact Flash Card	No
USB memory	Yes
SIMATIC IPC USB Flashdrive (USB stick)	Yes
SIMATIC HMI USB stick	Yes
Mechanics/material	
Enclosure material (front)	
• Plastic	Yes
• Aluminum	No

- Stainless steel

No

Dimensions

Width of the housing front	141 mm
Height of housing front	116 mm
Mounting cutout, width	123 mm
Mounting cutout, height	99 mm
Overall depth	33 mm

Weights

Weight (without packaging)	360 g
Weight (with packaging)	470 g

A.3 Siemens SIMATIC S7-1500 digital input/output module

Tab. Technické údaje zařízení A.3 Siemens SIMATIC S7-1500 digital input/output module (upraveno)³

Produkt	
Objednáací číslo	6ES7523-1BL00-0AA0
Popis výrobku	SIMATIC S7-1500 digital input/output module, DI16x 24VDC BA, 16 channels in groups of 16, input delay typ. 3.2 ms input type 3 (IEC 61131), DQ16XDC 24V/0.5A BA; 16 channels in groups of 8; 4 A per group; the module supports the safety-oriented shutdown of load groups up to SIL2 according to EN IEC 62061:2021 and Category 2 / PL c according to EN ISO 13849-1:2015. delivery including front connector push-in,
Produktová skupina	<u>SM 523 digital input/output modules</u>
Status výrobku (PLM)	PM300:Výrobek je aktivní
Informace o ceně	
Regionální cenová skupina / Cenová skupina Stammhausu	219
Katalogová cena	<u>Zobrazit ceny</u>
Vaše cena	<u>Zobrazit ceny</u>
Příplatek za barevné kovy a kovy vzácných zemin	není
Kód barevných kovů a kovů vzácných zemin	není
Dodací informace	
Exportní označení	AL : N / ECCN : N
Standardní dodací lhůta bez započtení doby přepravy	70 pracovních dní
Váha netto za cenovou jednotku	0,346 kg
Rozměry balení	15,10 x 20,30 x 3,70

³ 6ES7523-1BL00-0AA0 SIMATIC S7-1500 DI/DQ module. Siemens industry mall [online]. [cit. 2023-04-12]. Dostupné z: <https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7523-1BL00-0AA0>

Jednotky rozměru zásilky	CM
Cenová jednotka	1 Kus
Balné množství	1
Další informace o produktu	
EAN	4047623402312
UPC	887621779047
Číslo celní nomenklatury	85389091
Označení katalogu (LKZ)	ST73
Skupina výrobků (PCK/FAGR)	4501
Klíč obchodního úseku (GCK/GBK)	R151
Země původu:	Německo
Směrnice RoHS k omezení nebezpečných látek	splňuje od: 21.10.2013

A.4 Siemens SIMATIC HMI KP900 Comfort

Tab. Technické údaje zařízení A.4 Siemens SIMATIC HMI KP900 Comfort (upraveno) ⁴	
General information	
Product type designation	KP900 Comfort
Display	
Design of display	TFT
Screen diagonal	9 in
Display width	195 mm
Display height	117 mm
Number of colors	16 777 216
Resolution (pixels)	
• Horizontal image resolution	800 pixel
• Vertical image resolution	480 pixel
Backlighting	
• MTBF backlighting (at 25 °C)	80 000 h
• Backlight dimmable	Yes; 0-100 %
Control elements	
Keyboard fonts	
• Function keys	
— Number of function keys	26
— Number of function keys with LEDs	26
• Keys with LED	Yes
• System keys	Yes

⁴ 6AV2124-1JC01-0AX0 SIMATIC HMI KP900 Comfort. Siemens industry support [online]. [cit. 2023-04-12]. Dostupné z: <https://support.industry.siemens.com/cs/pd/469655?pdti=td&dl=cs&lc=cs-CZ>

• Numeric keyboard	Yes
• alphanumeric keyboard	Yes; similar to mobile phone
Touch operation	
• Design as touch screen	No; without
Expansions for operator control of the process	
• DP direct LEDs (LEDs as S7 output I/O) — F1...Fx	26
• Direct keys (keys as S7 input I/O) — F1...Fx	26
• Direct keys (touch buttons as S7 input I/O)	0
Installation type/mounting	
Mounting position	vertical
Wall mounting/direct mounting	No
Mounting in portrait format possible	No
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)	0.75 A
Starting current inrush I ² t	0.5 A ² ·s
Power	
Active power input, typ.	18 W
Processor	
Processor type	X86
Memory	
Flash	Yes
RAM	Yes
Memory available for user data	12 Mbyte
Type of output	
Info LED	Yes
Power LED	No
Error LED	No
Acoustics	
• Buzzer	No
• Speaker	Yes

A.5 Siemens SINAMICS S210 PN

Tab. Technické údaje zařízení A.5 Siemens SINAMICS S210 PN (upraveno) ⁵	
Produkt	
Objednací číslo	6SL3210-5HB10-1UF0
Popis výrobku	SINAMICS S210 Vstupní napájecí napětí: 200-240V 1AC, -10%/+10% , 1.4A, 45-66Hz, Výstupní napětí: 0-vstupní napětí V, 0.8 A 0-550Hz, Motor: 0.1 kW, Krytí: IP20, velikosti A, 45 x 170 x 170 (ŠxVxH)
Produktová skupina	Ordering Data Overview
Status výrobku (PLM)	PM300:Výrobek je aktivní
Informace o ceně	
Regionální cenová skupina / Cenová skupina Stammhausu	DMM
Katalogová cena	Zobrazit ceny
Vaše cena	Zobrazit ceny
Příplatek za barevné kovy a kovy vzácných zemin	není
Kód barevných kovů a kovů vzácných zemin	není
Dodací informace	
Exportní označení	AL : N / ECCN : 3A999A
Standardní dodací lhůta bez započtení doby přepravy	25 pracovních dní
Váha netto za cenovou jednotku	1,100 kg
Rozměry balení	217,00 x 237,00 x 67,00
Jednotky rozměru zásilky	MM
Cenová jednotka	1 Kus
Balné množství	1
Další informace o produktu	
EAN	4042948675748
UPC	Není dostupný
Číslo celní nomenklatury	85044095
Označení katalogu (LKZ)	D32
Skupina výrobků (PCK/FAGR)	9019
Klíč obchodního úseku (GCK/GBK)	R220
Země původu:	Čína
Směrnice RoHS k omezení nebezpečných látek	splňuje od: 31.05.2017

⁵ 6SL3210-5HB10-1UF0 SINAMICS S210 PN. Siemens industry mall [online]. [cit. 2023-04-12]. Dostupné z: <https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6SL3210-5HB10-1UF0>

