

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**Elektrotechnická fakulta**

**Katedra elektrických pohonů a trakce**



**Řízení plotru pomocí PLC SIMATIC**

**Plotter Control by SIMATIC**

Diplomová práce

Študijný program: Elektrotechnika, energetika a management

Študijný obor: Aplikovaná elektrotechnika

Špecializácia: Elektrické pohony

Vedúci práce: Ing. Ján Bauer, Ph.D.

**Bc. Tomáš Gajdoš**

**Praha 2021**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	<b>Gajdoš</b>	Jméno: <b>Tomáš</b>	Osobní číslo: <b>466224</b>
Fakulta/ústav:	<b>Fakulta elektrotechnická</b>		
Zadávací katedra/ústav:	<b>Katedra elektrických pohonů a trakce</b>		
Studijní program:	<b>Elektrotechnika, energetika a management</b>		
Specializace:	<b>Elektrické pohony</b>		

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Řízení plotru pomocí PLC SIMATIC**

Název diplomové práce anglicky:

**A Plotter Control by SIMATIC**

Pokyny pro vypracování:

- 1) Seznamte se s programovatelným logickým automatem SIMATIC S1500
- 2) Navrhněte strukturu logického řízení pro ploter
- 3) Implementujte navržené řízení na PLC a plotru
- 4) Doplňte navržené řízení o vhodnou vizualizaci na HMI panelu

Seznam doporučené literatury:

[1] Weidauer J., Messer R. Electrical Drives, Publics Erlangen, 2014  
[2] SCE Training Curriculum. Siemens AG, 2016  
[3] Durry B. The Control Techniques Drives and Controls Handbook 2nd ed., IeT, 2009  
[4] Pavelka J., Kobrle P. Elektrické pohony a jejich řízení. 3. přepracované vydání. Praha: České vysoké učení technické v Praze, 2016. ISBN 978-80-01-06007-0.

Jméno a pracoviště vedoucí(ho) diplomové práce:

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

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

Datum zadání diplomové práce: 24.01.2021      Termín odevzdání diplomové práce: 04.01.2022

Platnost zadání diplomové práce: 30.09.2022

\_\_\_\_\_

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

Student 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

### **Prehlásenie**

„Prehlasujem, že som predloženú prácu vypracoval samostatne a že som uviedol všetky použité informačné zdroje v súlade s Metodickým pokynom o dodržiavaní etických princípov pri príprave vysokoškolských záverečných prác.“

V Prahe dňa:

Podpis:

### **Pod'akovanie**

Ďakujem vedúcemu mojej diplomovej práce Ing. Jan Bauer Ph.D. za účinnú pedagogickú, metodickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej diplomovej práce. Taktiež ďakujem za technické konzultácie poskytnuté odborníkom z firmy Siemens, Ing. Martinom Kozákom.

V Prahe dňa ..... Podpis autora .....

## **Abstrakt**

Diplomová práca sa zaoberá oživením plotru pomocou systému SIMATICS. V prvej časti práce sa venujeme všeobecným vysvetlením pracovného prostredia a pohonu, kde opisujeme prostredie TIA portálu, jeho možnosti a popis aplikácie. V ďalšej časti aplikácie vysvetľujeme celý proces inicializácie a implementácie jednotlivých prvkov s ktorými pracujeme. Okrem samotnej programovej časti opisujeme aj vytvorenie HMI prostredia a popis ovládania programu. Posledná časť obsahuje výsledky simulácie a výsledky reálnej aplikácie.

## **Kľúčové slová**

Ploter, TIA portál, servomotory, SIMATICS, Kinematics, SINAMICS

## **Abstract**

Diploma thesis is focused on setting up a plotter using SIMATICs. In the first part of the thesis, we give a general explanation of the working environment and the drive, where we describe the TIA portal environment, its capabilities and a description of the application. In the next part of the application we explain the whole process of initialization and implementation of the different elements we work with. In addition to the actual program part, we also describe the creation of the HMI environment and a description of the program control. The last section contains the simulation results and the results of the real application.

## **Key words**

Plotter, TIA portal, servomotors, SIMATICs, Kinematics, SINAMICS

# Obsah

Úvod .....	12
<b>1. Teoretická časť .....</b>	<b>13</b>
<b>1.1 Popis pohonu.....</b>	<b>13</b>
<b>1.2 Technologické objekty.....</b>	<b>14</b>
<b>1.2.1 Kinematics .....</b>	<b>14</b>
<b>1.2.2 Základne funkcie .....</b>	<b>17</b>
<b>1.3 Centrálné riadenie .....</b>	<b>18</b>
<b>1.4 Komunikácia.....</b>	<b>19</b>
<b>1.5 Popis funkcií v programe.....</b>	<b>21</b>
<b>2. Praktická časť.....</b>	<b>22</b>
<b>2.1 Popis hardveru.....</b>	<b>22</b>
<b>2.1.1 PLC SIMATICS S7-1500 .....</b>	<b>22</b>
<b>2.1.2 SINAMICS S210.....</b>	<b>23</b>
<b>2.1.3 Servomotor SIMOTICS S-1FK2.....</b>	<b>24</b>
<b>2.1.4 SIMATICS HMI panel .....</b>	<b>25</b>
<b>2.2 TIA portál.....</b>	<b>26</b>
<b>2.2.1 Popis a konfigurácia HW.....</b>	<b>27</b>
<b>2.2.1.1 Konfigurácia PLC.....</b>	<b>29</b>
<b>2.2.1.2 Konfigurácia meničov.....</b>	<b>29</b>
<b>2.2.2 PLC programovanie .....</b>	<b>30</b>
<b>2.2.2.1 Použité knižnice .....</b>	<b>31</b>
<b>2.2.2.2 Technologické objekty .....</b>	<b>31</b>
<b>2.2.2.3 Bloky hlavného programu .....</b>	<b>33</b>
<b>2.2.2.4 Tags – PLC a HMI.....</b>	<b>36</b>
<b>2.2.3 HMI.....</b>	<b>37</b>
<b>2.2.3.1 Obrazovky HMI.....</b>	<b>37</b>
<b>2.2.3.2 Ovládanie .....</b>	<b>42</b>
<b>2.3 TIA portál.....</b>	<b>45</b>
<b>2.3.1.1 Simulácia.....</b>	<b>45</b>
<b>2.3.1.2 Reálna aplikácia.....</b>	<b>46</b>
Záver .....	55
Zoznam použitej literatúry .....	56

## **Zoznam obrázkov**

- Obr. 1.1.1 Zjednodušený popis aplikácie
- Obr. 1.2.1 Spôsob komunikácie medzi technologickými objektami a PLC
  - Obr. 1.2.1.1 Kinematics možnosti aplikácií
  - Obr. 1.2.1.2 Súradnicové systémy Kinematics
  - Obr. 1.2.1.3 Zóny pohybu v Kinematics
  - Obr. 1.2.1.4 Nastavenia zón v Kinematics
  - Obr. 1.2.1.5 3D zobrazenie Kinematics
  - Obr. 1.2.1.6 Spracovávanie príkazov v Kinematics
  - Obr. 1.2.2.1 Základné funkcie Kinematics
- Obr. 1.3.1 Rozdiel medzi centrálnym a decentrálnym riadením pohonu
- Obr. 1.4.1 Komunikácia medzi prvkami aplikácie
- Obr. 1.5.1 Typy PathData
- Obr. 1.5.2 Štruktúra procesu programu
- Obr. 2.1.1.1 PLC S1511-T
- Obr. 2.1.2.1 Frekvenčný menič S210
- Obr. 2.1.3.1 Servomotor S-1FK2
- Obr. 2.1.4.1 HMI panel KTP400
- Obr. 2.1.4.2 HMI panel TP700 Comfort
- Obr. 2.2.1 TIA portál - Pracovné prostredie
  - Obr. 2.2.1.1. TIA portál - Výber motora pre menič
  - Obr. 2.2.1.2 TIA portál - Topologické okno
    - Obr. 2.2.2.1.1 TIA portál - Importované funkcie
    - Obr. 2.2.2.2.1 TIA portál - TO a konfigurácia Kinematics
    - Obr. 2.2.2.2.2 TIA portál - Aktívny a pasívny Homing
    - Obr. 2.2.2.2.3 TIA portál - Zapnuté DSC
    - Obr. 2.2.2.3.1 TIA portál - Hlavná sekvencia programu



Obr. 2.2.2.3.2 TIA portál - Dátový blok Controls

Obr. 2.2.2.3.3 TIA portál - Organizačný blok Draw

Obr. 2.2.2.3.4 TIA portál - Funkčný blok Draw 1/3

Obr. 2.2.2.3.5 TIA portál - Funkčný blok Draw 2/3

Obr. 2.2.2.3.6 TIA portál - Funkčný blok Draw 3/3

Obr. 2.2.2.3.7 TIA portál - Dátový blok Draw

Obr. 2.2.2.3.8 TIA portál - Dátový blok PathData

Obr. 2.2.2.4.1 TIA portál - PLC tagy technologických objektov

Obr. 2.2.2.4.2 TIA portál - HMI tagy

Obr. 2.2.3.1.1 TIA portál - HMI Wizard 1/2

Obr. 2.2.3.1.2 TIA portál - HMI Wizard 2/2

Obr. 2.2.3.1.3 TIA portál - Hlavné okno HMI prostredia

Obr. 2.2.3.1.4 TIA portál - Diagnostické okno HMI prostredia

Obr. 2.2.3.1.5 TIA portál - Drag&Drop Editora z importovanej knižnice

Obr. 2.2.3.1.6 TIA portál - Prepojenie editora s dátovým blokom PLC

Obr. 2.2.3.1.7 TIA portál - Okno editora

Obr. 2.2.3.1.8 TIA portál - Príkazy editora

Obr. 2.2.3.2.1 TIA portál - Program vykonaný

Obr. 2.2.3.2.2 TIA portál - Editor

Obr. 2.2.3.2.3 TIA portál - Editor – špecifikácia príkazu

Obr. 2.2.3.2.4 TIA portál - Editor – Blending nastavenia

Obr. 2.2.3.2.5 TIA portál - Editor – Nastavenie flagov

Obr. 2.3.1.1.1 TIA portál - Ovládacia lišta

Obr. 2.3.1.1.2 TIA portál - Watch table

Obr. 2.3.1.2.1 TIA portál - Konfigurácia alternatívnej aplikácie

Obr. 2.3.1.2.2 TIA portál - Pripojenie

Obr. 2.3.1.2.3 TIA portál - Pripojenie k vybranému zariadeniu

Obr. 2.3.1.2.4 TIA portál - Úspešné pripojenie

- Obr. 2.3.1.2.5 TIA portál - HMI nahrávanie
- Obr. 2.3.1.2.6 TIA portál - Basic Draw 1/2
- Obr. 2.3.1.2.7 TIA portál - Basic Draw 2/2
- Obr. 2.3.1.2.8 TIA portál - Dátový blok Controls\_Basic
- Obr. 2.3.1.2.9 TIA portál - Úvodná obrazovka
- Obr. 2.3.1.2.10 TIA portál - Hlavná obrazovka
- Obr. 2.3.1.2.11 TIA portál - HMI tagy
- Obr. 2.3.1.2.12 TIA portál - Commissioning
- Obr. 2.3.1.2.13 TIA portál - Reálna aplikácia
- Obr. 2.3.1.2.14 TIA portál - Vykreslený obraz
- Obr. 2.3.1.2.15 TIA portál - Súradnice vykresľovania

## Zoznam tabuliek

Tab. 1.4.1 Telegram 105

Tab. 1.4.2 Hodnoty kontrolných a statusových slov

Tab. 2.1.1 Použitý hardware

Tab. 2.1.3.1 Štítkové hodnoty motora

Tab. 2.2.1 Použitý software

Tab. 2.2.2.1 IEC normy a SIEMENS ekvivalenty

Tab. 2.3.1.2.1 Verzie HW alternatívneho programu

## Zoznam skratiek a symbolov

PLC	Programovateľný logický kontrolér
FM	Frekvenčný menič
HMI	Rozhranie človek stroj
TIA	Kompletne integrovaná automatizácia
TO	Technologický objekt
HW	Hardware
SW	Software
FW	Firmware
RT	Reálny čas
IRT	Izochrónny reálny čas
NRT	Nie reálny čas
Nm	Jednotka momentu - Newton metre
N	Jednotka sily - Newton
mm	Jednotka vzdialenosti - milimeter

# Úvod

V dnešnej dobe je snaha o čo najvyššiu mieru automatizácie v priemysle, ale aj v iných oblastiach dennodenného života. Často skloňovaná Automatizácia 4.0 sa dostáva čoraz viac do popredia a ovplyvňuje aj smerovanie celého priemyslu. Základným elementom každého automatizovaného procesu je programovateľná logická jednotka, resp. kontrolér (**P**rogrammable **L**ogic **C**ontroller). Taktiež neodmysliteľnou súčasťou systému je frekvenčný menič. Okrem spoločnosti SIEMENS sa automatizácii venujú mnohé ďalšie spoločnosti ako ABB, Rockwell, Schneider Electric a ďalší.

Frekvenčné meniče používané v súčasnej dobe disponujú veľkou flexibilitou v riadení pohonov vďaka výkonným priemyselným počítačom, ktoré obsahujú, najmä v porovnaní s jednoduchšími spôsobmi riadenia pohonov. Nevýhodou týchto meničov je nákladnejšia investícia, ale zákazník získa pohon s vysokou presnosťou, dynamikou a možnosť synchronizácie s inými pohonmi v systéme súčasne. Ďalšou výhodou FM od popredných spoločností je vysoko štandardizované programovacie prostredie a množstvo dodatočných knižníc, ktoré po importovaní do programu predstavujú výrazné zníženie potrebného času na implementáciu a spustenie reálnej aplikácie. Okrem toho zákazník kúpou získa aj technickú podporu výrobcu a servis výrobkov. Súčasťou programovacích prostredí častokrát bývajú aj možnosti simulácie, ktoré hrajú dôležitú rolu v konfiguračnej a integračnej časti projektu, aby sa všetky prípadné chyby a nežiadúce správanie odstránilo pred implementáciou na reálnu aplikáciu.

Moja práca je zameraná na opísanie riadenia plotra pomocou PLC a meničov, ktoré poháňajú servomotory z portfólia SIEMENS. Konkrétne riadenie pohonu je zabezpečené pomocou meničov SINAMICS, ktoré sú implementované do celku spolu s PLC, ktoré zabezpečuje riadiacu časť, a v neposlednom rade interaktívny ovládací panel (**H**uman **M**achine **I**nterface), ktorý slúži k ovládaniu danej aplikácie operátorom.

Môj projekt je založený na jednej zo štandardizovaných aplikácií od spoločnosti SIEMENS. Okrem tejto aplikácie SIEMENS ponúka ešte množstvo ďalších štandardizovaných aplikácií, ktoré zákazníkom uľahčia implementáciu a integráciu do ich projektu alebo výrobného procesu. Celý projekt je možné aj plne simulovať, nakoľko v čase písania tejto práce som bol na výmennom zahraničnom pobyte a nemal som prístup k fyzickému hardwaru.

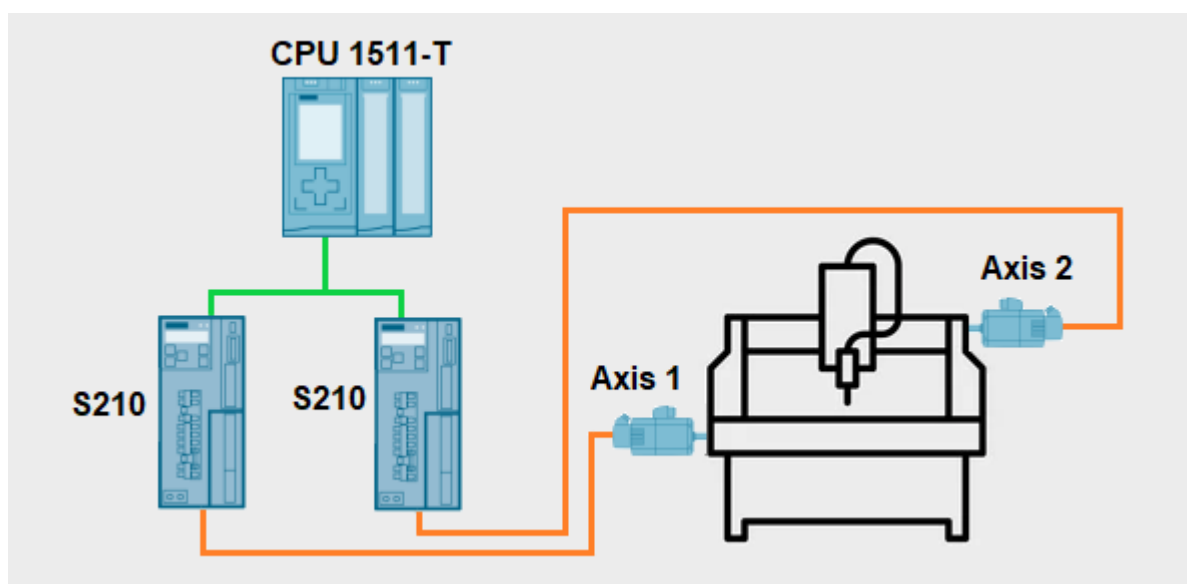
Finálnym zámerom práce je vytvoriť funkčnú aplikáciu, ktorú bude možné využiť na prípadne edukačné účely na pôde univerzity alebo použiť ako základ pre ďalšie aplikácie.

# 1. Teoretická časť

V teoretickej časti práce sa venujeme najmä popisu prostredia v ktorom budeme programovať – TIA portál a následne vysvetlíme a opíšeme používaný HW a SW a jednotlivé prepojenia medzi nimi. Naša aplikácia opisuje ploter s PLC, HMI a servopohonmi, ktoré sú riadené cez pohony SINAMICS od spoločnosti SIEMENS. Princíp našej aplikácie je možné využiť aj na iné aplikácie, keďže sa jedná o pohyb v dvoch osách a je možné meniť nástroj, čo ponúka široké spektrum aplikácií v závislosti na napr. zvolenom nástroji (laser, trysky a pod.), takže možnosti finálnych aplikácií sú rezacie alebo orezávacie plotre, CNC plotre, rôzne kresliace aplikácie a mnoho ďalších. V teoretickej časti tiež opíšeme ako jednotlivé súčasti medzi sebou komunikujú a aké rozhranie na to používajú. Venujeme sa v nej aj popisu jednotlivých technologických objektov, ich konfigurácii a možnostiam v rámci programovacieho prostredia. Taktiež aj vysvetlíme ako program pracuje so vstupnými a výstupnými dátami.

## 1.1 Popis pohonu

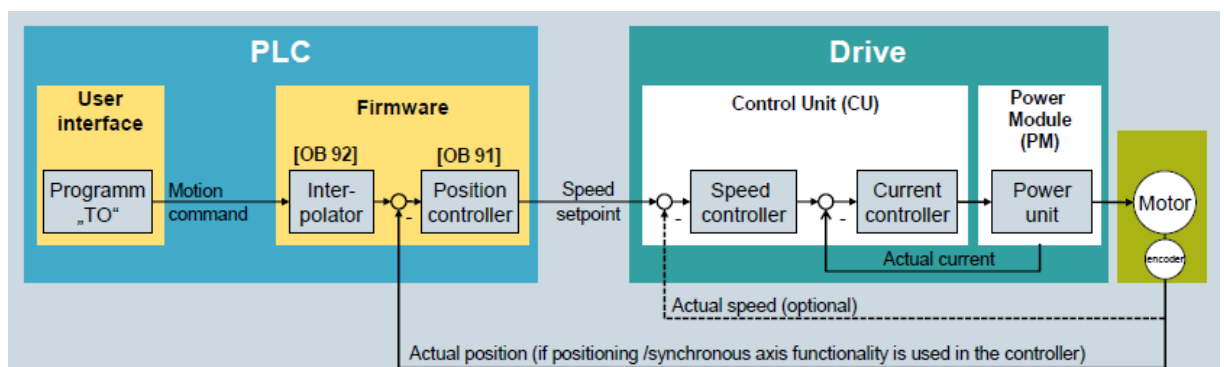
V našej aplikácii využijeme dva servomotory, kde jeden z nich zabezpečuje pohyb v ose X a druhý v ose Y. Oba pohyby sú lineárne. V programe predstavuje každá osa jeden technologický objekt (TO) – polohovaciu osu, ktorý reprezentuje jeden z motorov v programe. Polohovacia osa okrem štandardnej funkcie zrýchlenia a brzdenia riadi aj polohu. Medzi pohonom a program následne dochádza ku komunikácii cez komunikačné telegramy, ktoré budú opísané v ďalšej časti práce. Zjednodušený popis aplikácie môžeme vidieť na Obr. 1.1.1.



Obr. 1.1.1 Zjednodušený popis aplikácie

## 1.2 Technologické objekty

Technologické objekty (TO) sú programové objekty v PLC, ktoré reprezentujú reálne mechanické komponenty pripojené k meniču pohonu a zároveň umožňujú použitie technologických funkcií, ktoré dané TO ponúkajú (MC-xx, PLCopen, atď.). Okrem základnej konfigurácie ponúkajú technologické objekty aj možnosti diagnostiky. V TIA portáli nájdeme základné TO, medzi ktoré patrí rýchlostná osa, polohovacia osa, synchronizačná osa. Tie následne vytvárajú spojenie s hardwarovou súčasťou aplikácie. Komunikujú spolu cez telegramy (viac v kapitole 1.4) a vymieňajú si aktuálne hodnoty a inštrukcie. Na ďalšom obrázku je opísaný spôsob komunikácie medzi TO, pohonom a motorom.











Obr. 1.2.1 Spôsob komunikácie medzi technologickými objektami a PLC [5]

Okrem spomenutých technologických objektov je pre nás ešte jeden dôležitý objekt, ktorý použijeme v našej aplikácii – TO Kinematics. Technologický objekt Kinematics má navyše možnosť sledovania pohybu v 3D a tiež aj záznamu. Samotné PLC ešte ponúka technologické objekty pre rôzne vačky, sledovače vačiek, externé enkodéry a pod. V našom prípade využijeme len TO Kinematics a dve polohovacie osi.

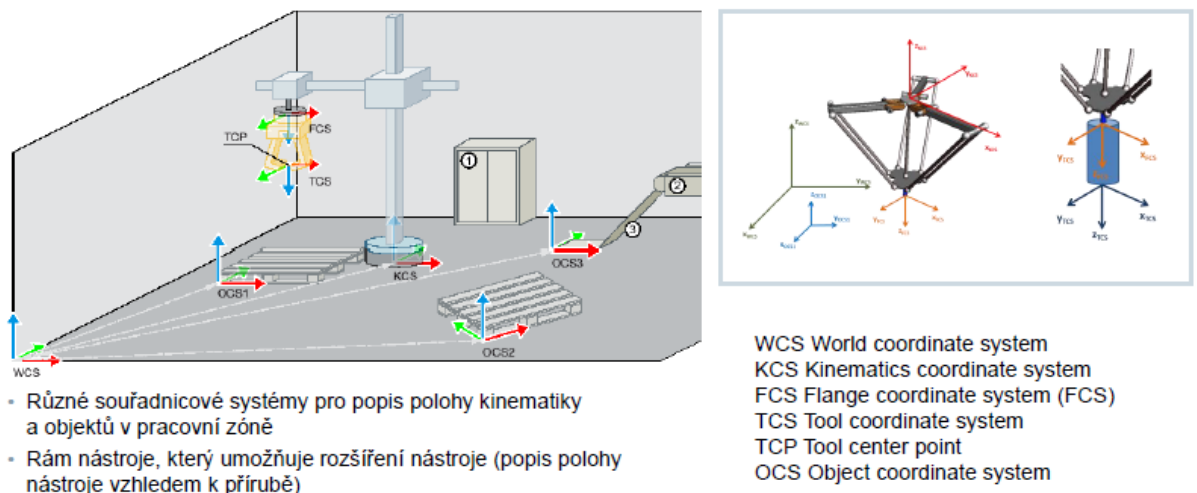
### 1.2.1 Kinematics

V aplikácii využívame okrem polohovacích osí aj technologický objekt Kinematics. Tento technologický objekt je v rámci základného výberu od SIEMENSu určený práve na aplikácie pohybu v dvojrozmernom alebo trojrozmernom smere pri použití rôznych náradí, hlavic alebo smerov v závislosti na aplikácií. Je to preddefinované od výrobcu s cieľom maximálnej univerzálnosti a jednoduchosti už pri konkrétnych aplikáciách. Okrem pohybu 2D alebo 3D v kartézskych súradniciach si je možné zvoliť aj aplikáciu ovládacej ruky, cylindrickeho robota alebo si zdefinovať vlastný systém. Príklady je možné vidieť na Obr. 1.2.1.1.

<b>Cartesian portal</b>  2D (x, z) 2D with orientation (A) 3D (x, y, z) 3D with orientation (A)	<b>Articulated arm</b>  2D (x, z) 2D with orientation (A) 3D (x, y, z) 3D with orientation (A)	<b>Tripod</b>  3D (x, y, z) 3D with orientation (A)
<b>Roller picker</b>  2D vertical (x, z) 2D vertical with orientation (A) 3D vertical (x, y, z) 3D vertical with orientation (A) 3D horizontal with orientation (A)	<b>Cylindrical robot</b>  3D (x, y, z) 3D with orientation (A)	<b>Customized kinematics</b> Free transformation interface  2D (x, z) 2D with orientation (A) 3D (x, y, z) 3D with orientation (A)
<b>Delta picker</b>  2D (x, z) 2D with orientation (A) 3D (x, y, z) 3D with orientation (A)	<b>Scara</b>  3D with orientation (A)	

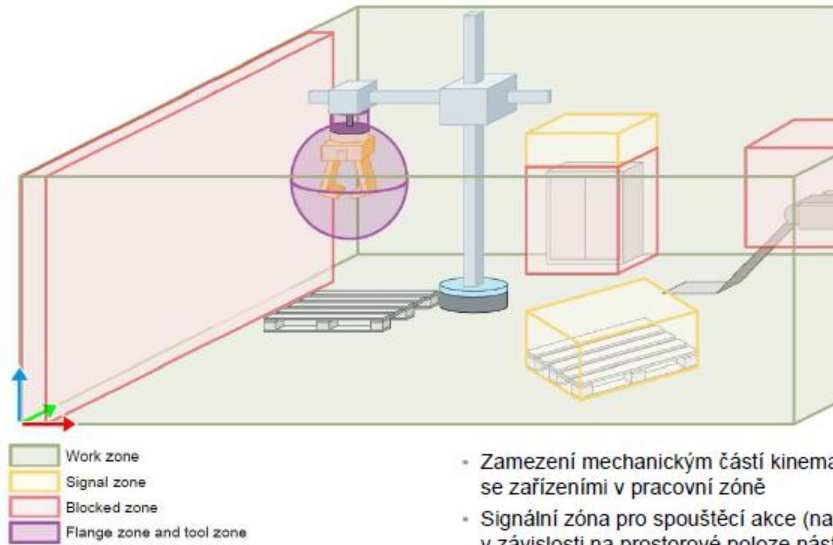
Obr. 1.2.1.1 Kinematics možnosti aplikácií [5]

Po výbere aplikácie Kinematics si ju následne spojíme s technologickými objektami – najčastejšie polohovacie osi. Okrem jednotlivých osí, ktoré v hardvari predstavujú jednotlivé pohony, si ešte prepojíme zvyšné súčasti aplikácie, ako sú koncové a začiatkové HW alebo SW spínače, aby sme pracovali v správnych limitoch. Pri Kinematics si môžeme zvoliť súradnicový systém a zóny, pomocou ktorých popíšeme pohyb v danom priestore a tiež aj pohyb nástroja. Viac môžeme vidieť na Obr. 1.2.1.1.



Obr. 1.2.1.2 Súradnicové systémy Kinematics [5]

Kinematics ponúka relatívne jednoduché SW vymedzenie pracovnej zóny už počas úvodnej konfigurácie. Taktiež celú zónu je možné rozdeliť na jednotlivé pracovné oblasti, zakázané zóny alebo zóny, kde operuje nástroj. Vymedzenie pracovných oblastí môže významne zvýšiť bezpečnosť prevádzky daného stroja, resp. celého systému. Príkladné nastavenie vidíme na Obr. 1.2.1.3.



- Zamezení mechanickým částí kinematicky (příruba, nástroj), před kolizí se zařízeními v pracovní zóně
- Signální zóna pro spouštěcí akce (např. Otevření / zavření uchopovače) v závislosti na prostorové poloze nástroje a / nebo příruby

Obr. 1.2.1.3 Zóny pohybu v Kinematics [5]

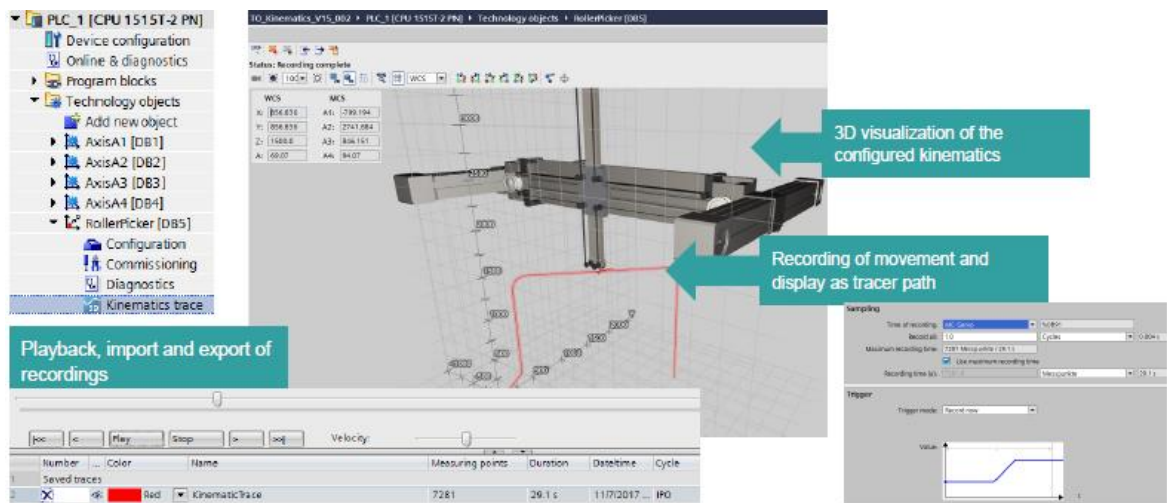
Na ďalšom obrázku vidíme príklad parametrického okna pre nastavenie pracovných, signálnych, blokových a operačných zón stroja a nástroja. Taktiež jednotlivé zóny môžu byť preddefinované, ale do momentu kým ich neaktivujeme v riadku „Status“, tak ostávajú neaktívne.

	Name in function view	Name in DB	Configuration status	Start value in project	Minimum value	Maximum value	Comment
All parameters	[Workspace zone 1] Type	..Type	Active	Work zone			Type of workspace zone
Configuration	[Workspace zone 1] Status	..Type	Active				
Basic parameters	[Workspace zone 1] Form	..Type		Cuboid			Zone geometry
Interconnections	[Workspace zone 1] Length x	..Parameter[1]		210.0	mm -1.0e+12	mm 1.0e+12	mm Zone geometry parameter
Geometry	[Workspace zone 1] Length y	..Parameter[2]		500.0	mm -1.0e+12	mm 1.0e+12	mm Zone geometry parameter
Extended param...	[Workspace zone 1] Length z	..Parameter[3]		297.0	mm -1.0e+12	mm 1.0e+12	mm Zone geometry parameter
Dynamics	[Workspace zone 1] CS	..ReferenceSystem		VCS			Reference coordinate sys...
Kinematics coord...	[Workspace zone 1] x	..x		0.0	mm -1.0e+12	mm 1.0e+12	mm x-coordinate
Object coordin...	[Workspace zone 1] y	..y		0.0	mm -1.0e+12	mm 1.0e+12	mm y-coordinate
Tools	[Workspace zone 1] z	..z		0.0	mm -1.0e+12	mm 1.0e+12	mm z-coordinate
Zones	[Workspace zone 1] A	..Ja		0.0	* -180.0	* 179.999	* A-coordinate
Job sequence	[Workspace zone 1] B	..Jb		0.0	* -180.0	* 179.999	* B-coordinate
Diagnosics	[Workspace zone 1] C	..Jc		0.0	* -180.0	* 179.999	* C-coordinate
Status and error bits	[Workspace zone 2] Type	..Type		Work zone			Type of workspace zone
Motion status	[Workspace zone 2] Status	..Type		Invalid			
Zones and tools	[Workspace zone 2] Form	..Type		Cuboid			Zone geometry
	[Workspace zone 2] Length x	..Parameter[1]		500.0	mm -1.0e+12	mm 1.0e+12	mm Zone geometry parameter
	[Workspace zone 2] Length y	..Parameter[2]		500.0	mm -1.0e+12	mm 1.0e+12	mm Zone geometry parameter
	[Workspace zone 2] Length z	..Parameter[3]		500.0	mm -1.0e+12	mm 1.0e+12	mm Zone geometry parameter
	[Workspace zone 2] CS	..ReferenceSystem		VCS			Reference coordinate sys...
	[Workspace zone 2] x	..x		0.0	mm -1.0e+12	mm 1.0e+12	mm x-coordinate
	[Workspace zone 2] y	..y		0.0	mm -1.0e+12	mm 1.0e+12	mm y-coordinate
	[Workspace zone 2] z	..z		0.0	mm -1.0e+12	mm 1.0e+12	mm z-coordinate
	[Workspace zone 2] A	..Ja		0.0	* -180.0	* 179.999	* A-coordinate
	[Workspace zone 2] B	..Jb		0.0	* -180.0	* 179.999	* B-coordinate
	[Workspace zone 2] C	..Jc		0.0	* -180.0	* 179.999	* C-coordinate
	[Workspace zone 3] Type	..Type		Work zone			Type of workspace zone

Obr. 1.2.1.4 Nastavenia zón v Kinematics

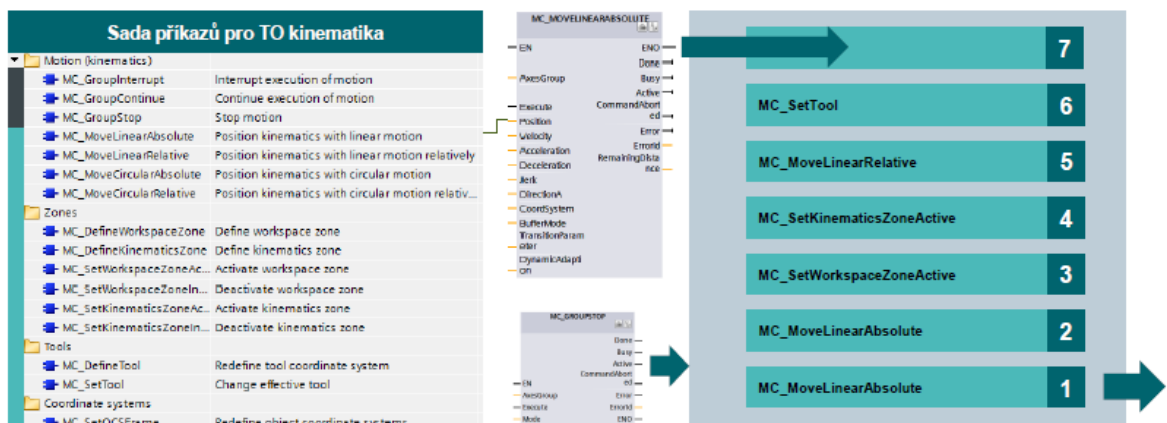
Ďalšia z výhod Kinematics je možnosť záznamu pohybu a 3D vizualizáciu. Túto možnosť nájdeme v možnostiach nastavenia TO\_Kinematics. Takže v momente, keď budeme simulovať pohyb, tak si ho vieme zaznamenať a následne analyzovať, či daný pohyb spĺňa zadanú dráhu, má dostatočnú plynulosť v ohybe a podľa potreby si upraviť parametre pohybu. Táto vizualizácia je navyše v 3D, takže je to prehľadné zo všetkých strán. Príklad môžeme vidieť na Obr. 1.2.1.5.





Obr. 1.2.1.5 3D zobrazenie Kinematics [5]

Samotné programovanie v Kinematics s funkčnými blokmi je vo forme príkazov, ktoré sa zadávajú do fronty, kde je možné do príkazového riadku zadať až 10 príkazov. Dynamické plánovanie sa uskutočňuje v celej sekvencii príkazov. Základný popis vidíme na ďalšej snímke.



Obr. 1.2.1.6 Spracovávanie príkazov v Kinematics [5]

## 1.2.2 Základne funkcie

TO Kinematics ponúka základne preddefinované funkcie pre jednoduchšie programovanie daných aplikácií. Na Obr. 1.2.2.1 vidíme základné z nich, ktoré primárne slúžia na vykonanie, prerušenie, zastavenie pohybu alebo na lineárny a rotačný pohyb absolútny alebo lineárny a rotačný pohyb relatívny. Taktiež sú tu funkcie na definovanie pracovných a kinematických zón, ale aj definovanie používaných nástrojov.

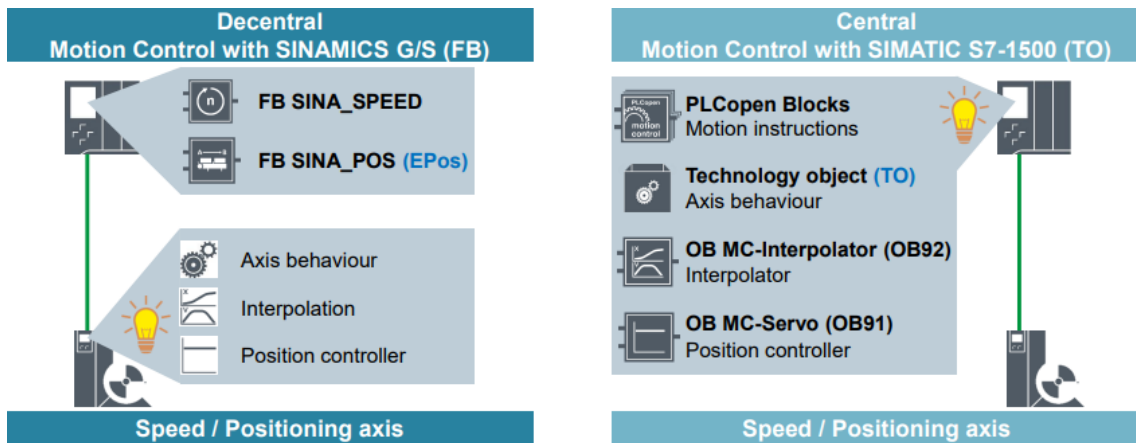
▼ Motion (kinematics)	
MC_GroupInterrupt	Interrupt execution of motion
MC_GroupContinue	Continue execution of motion
MC_GroupStop	Stop motion
MC_MoveLinearAbsolute	Position kinematics with linear motion
MC_MoveLinearRelative	Position kinematics with linear motion relatively
MC_MoveCircularAbsolute	Position kinematics with circular motion
MC_MoveCircularRelative	Position kinematics with circular motion relativ...
Zones	
MC_DefineWorkspaceZone	Define workspace zone
MC_DefineKinematicsZone	Define kinematics zone
MC_SetWorkspaceZoneAc...	Activate workspace zone
MC_SetWorkspaceZoneIn...	Deactivate workspace zone
MC_SetKinematicsZoneAc...	Activate kinematics zone
MC_SetKinematicsZoneIn...	Deactivate kinematics zone
Tools	
MC_DefineTool	Redefine tool coordinate system
MC_SetTool	Change effective tool
Coordinate systems	
MC_SetOCSFrame	Redefine object coordinate systems

Obr. 1.2.2.1 Základné funkcie Kinematics [5]

Okrem týchto funkcií sme importovali do projektu Kinematics knižnicu – Library Kinematics Control (skratka LKinCtrl), ktorá umožňuje ešte jednoduchšie programovanie a kontrolu nad pohybom pomocou príkazov v editore. V editore si operátor zdefinuje celý pohyb, ktorý je následne vykonaný. Knižnica okrem toho ponúka automatický a „single step“ krok. Okrem toho tu nájdeme aj diagnostiku, „flags“ pre aktuátory závislé na aktuálnom statuse inštrukcií pohybu.

### 1.3 Centrálné riadenie

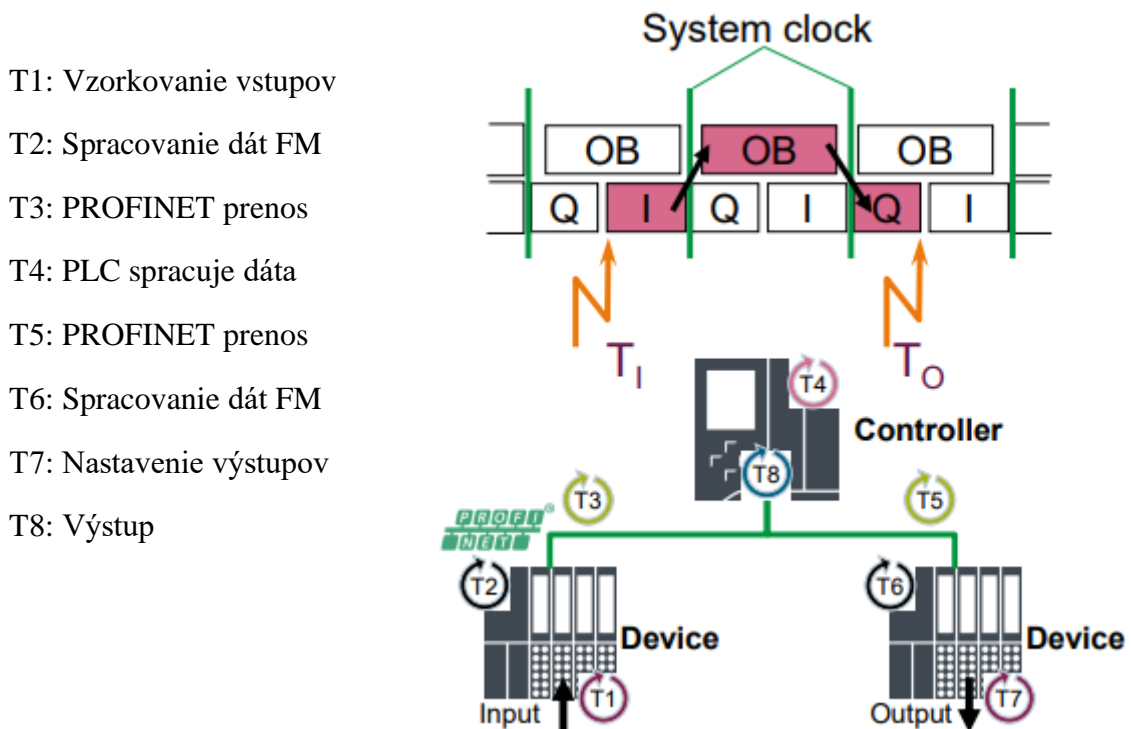
Automatizácia má v princípe dve možnosti riadenia – centrálna alebo decentrálna. V našej aplikácii použijeme centrálna riadenie. Základným rozdielom medzi týmito možnosťami je to, že riadenie ako také vykonáva zvyčajne PLC vzdialené od aplikácie a FM už len dostáva priamo pokyny v mieste aplikácie. V našom prípade PLC 1511-T, ktorý riadi frekvenčný menič S210, vykonáva všetky výpočetné operácie v organizačných blokoch MC-Interpolator, MC-Servo, PLCopen blokoch a posiela informácie na FM S210, ktorý už len vykonáva jednotlivé pokyny od PLC a riadi jednotlivé servomotory.



Obr. 1.3.1 Rozdiel medzi centrálnym a decentrálnym riadením pohonu [5]

## 1.4 Komunikácia

Medzi PLC a FM prebieha komunikácia po PROFINET sieti. Výmena dát funguje pomocou cyklických prenosov a môžu byť 2 možnosti – RT (Real time) a IRT (Isochronous Real time). V našej aplikácii využívame IRT, pretože má vysokú stabilitu pre časovo kritické aplikácie – v našom prípade riadenie pohybu. Inými slovami hlavnou výhodou IRT je rýchla doba cyklu – vysielané a prijímané dáta v správnom poradí a časovom slede a deterministické správanie siete. Okrem IRT komunikácie máme v aplikácii aj NRT (non Real time) – acyklický proces výmeny dát, ktorý je využívaný v komunikácii medzi PLC a HMI, nakoľko sa nejedná o časovo kritickú výmenu dát.



Obr. 1.4.1 Komunikácia medzi prvkami aplikácie [10]

Samotná výmena dát medzi PLC a pohonmi je založená na telegramoch. V našej aplikácii je používaný Telegram 105. Prvý riadok značí prijímané informácie a druhý posiadané.

PZD01	PZD02	PZD03	PZD04	PZD05	PZD06	PZD07	PZD08	PZD09	PZD10
STW1	NSET_B		STW2	MOMRED	G1_STW	XERR		KPC	
ZSW1	NACT_B		ZSW2	MSGW	G1_ZSW	e.g.: G1_XIST1		e.g.: G1_XIST2	

Tab 1.4.1 Telegram 105 [3]

Telegramy sú usporiadané dáta do blokov. Zo samotných telegramov je dôležité, že odosiadaný a prijímaný telegram sa u centrálného riadenia vždy berie z hľadiska meniča pohonu – preto napr. riadiace slovo STW1 je z pohľadu PLC odosiadaný signál a pre menič je to opačne. Telegram 105 je rozšírenie Telegramu 3 a je vhodný pre IRT komunikáciu. Význam jednotlivých častí telegramu:

- STW1, STW2 – riadiace slovo pre základné riadiace informácie
- ZSW1, ZSW2 – stavové slovo, stav polohy
- NSET\_B, NACT\_B – požadovaná a skutočná rýchlosť, 16bit premenná
- G1\_STW, G1\_ZSW – riadiace a stavové slovo enkódera 1, požadovaný režim pozícií, indikácia aktivovanej funkcie, homingu a pod.
- G1\_XIST1, G1\_XISTI2 – skutočná pozícia 1 a 2 enkódera 1
- MOMRED – hodnota pre redukciu momentu
- XERR – rozdiel skutočnej a požadovanej polohy
- KPC – proporčná hodnota regulátora polohy
- MELDW – komunikačné slovo

Každá časť telegramu je premenná o 16 bitoch typu Word. Je vyjadrená v HEX (hexadecimálnom) formáte. Na ďalšom obrázku môžeme vidieť tabuľku kontrolného slova 1 a stavového slova 1.

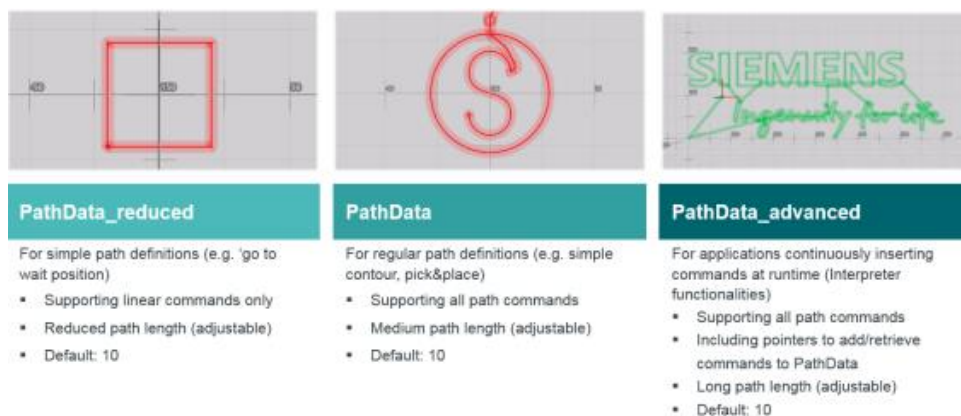
Control word 1 (STW1)		Status word 1 (ZSW1)	
Bit	Meaning	Bit	Meaning
00	ON / OFF1	00	Ready for switching on
01	OFF2	01	Ready for operation
02	OFF3	02	Operation enabled
03	Enable operation	03	Fault active
04	Enable ramp-function generator	04	No coasting down active
05	Continue ramp-function generator	05	No fast stop active
06	Enable speed setpoint	06	Switching on inhibited active
07	Acknowledge fault	07	Alarm active
08	Reserved	08	Speed setp - act val deviation in tolerance t_off
09	Reserved	09	Control request
10	Control by PLC	10	Comparison value reached/exceeded
11	Reserved	11	Alarm class bit 0
12	Open holding brake	12	Alarm class bit 1
13	Reserved	13	Reserved
14	Torque / speed control	14	Closed-loop torque control active
15	Reserved	15	Reserved

Tab. 1.4.2 Hodnoty kontrolných a statusových slov [3]

Zvyšné hodnoty pre jednotlivé časti telegramu nájdete v dokumentácii pre menič pohonu.

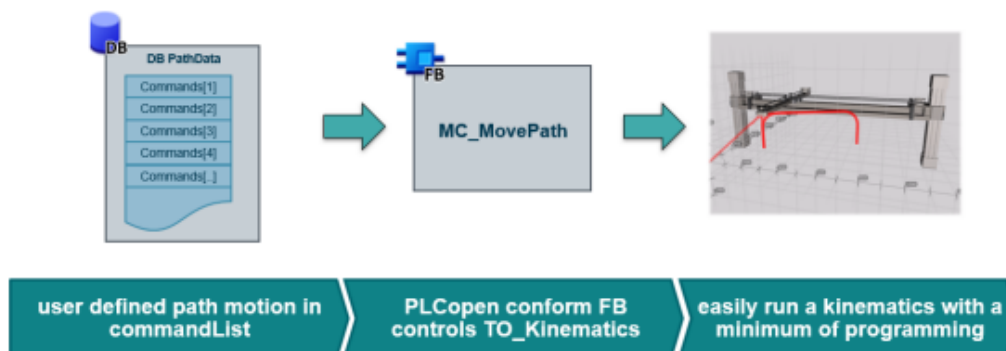
## 1.5 Popis funkcií v programe

V programe sú použité 3 technologické objekty (TO). Máme tu dve polohovacie osi a TO Kinematics. Technologický objekt Kinematics využíva obe osi s cieľom vykonania žiadaného pohybu na základe vstupu priamo z PLC alebo od operátora cez HMI interface. Dané polohovacie osi fungujú ako 2 nezávislé objekty a Kinematics je prvok, ktorý ich spája a vďaka ktorému je vykonaný požadovaný pohyb. Následne tu používame funkcie MC\_MovePath, ktorý zabezpečuje celkovú funkcionálnosť našej aplikácie. Táto funkcia kontroluje aj TO Kinematics a dráhu, ktorú chceme vykresliť. Okrem tohto potrebujeme aj v hlavnej sekcii programu funkciu na aktiváciu osí – MC\_GroupPower a tiež nastaviť ich možný reset – MC\_GroupReset a „Home“ – MC\_GroupHome pred každým úkonom, aby sme sa uistili, že program vychádza zo správnych hodnôt. V podsekcii na kreslenie ďalej používame okrem opísaných funkcií aj LKinCtrl\_Editor, ktorý slúži na komunikáciu medzi editorom v HMI a PLC a tiež potrebujeme funkciu na aktiváciu nástroja – v našom prípade pera. Okrem týchto funkcií tu máme množstvo použitých podblokov z preddefinovanej knižnice. Ďalej si priblížime PathData štruktúru. Táto štruktúra reprezentuje príkazový riadok, ktorý definuje celý pohyb. Na ďalšom obrázku môžeme pozorovať rozdiel v jednotlivých typoch PathData.



Obr. 1.5.1 Typy PathData [3]

Vďaka importovanej knižnici máme k dispozícii editor, v ktorom si nastavíme požadovaný pohyb. Následne zjednodušená štruktúra programu vyzerá ako na Obr. 1.5.2



Obr. 1.5.2 Štruktúra procesu programu [3]

## 2. Praktická časť

V tejto časti predstavíme použitý hardware. Hlavným riadiacim prvkom je PLC S7-1511T. Nevyhnutnou súčasťou aplikácie sú aj dva frekvenčné meniče S210, ktoré riadia SIMOTICS servomotory. Okrem toho používame aj HMI, konkrétne sa jedná o model TP700 Comfort. Všetky súčasti projekty sú prepojené na PROFINET sieti, kde dochádza k výmene informácií.

### 2.1 Popis hardveru

V nasledujúcej tabuľke môžeme nájsť základný popis používaného hardwaru z ktorých je náš systém zložený. Okrem základného názvu je tu možné nájsť aj verzie firmwaru a objednávkové čísla.

Názov	Verzia FW	MLFB
SIMATIC S7-1511T-1 PN	V2.8	6ES7511-1TK01-0AB0
SIMATIC S7 DI/DO module	V1.0	6ES7523-1BL00-0AA0
SIMATICS S7, pamäťová karta		6ES7954-8LE03-0AA0
SIMATICS S7-1500, DIN lišta 160 mm		6ES7590-1AB60-0AA0
SINAMICS S210 PN, 0.1kW	V5.2.3	6SL3210-5HB10-1UF0
SINAMICS KT400 Basic PN	V15.1.0.0	6AV2123-2DB03-0AX0
SINAMICS TP700 Comfort	V16.0.0.0	6AV2 124-0GC01-0AX0
2x SIMOTICS S-1FK2 HD Servomotor		1FK2102-0AG00-1MA0

Tab. 2.1.1 Použité hardware

#### 2.1.1 PLC SIMATICS S7-1500

V rámci SIEMENS portfólia rada S7-1500 patrí medzi vyššie, resp. najvyššie rady PLC. Okrem vysokého výpočetného výkonu ponúka aj väčšie možnosti prispôsobenia aplikácie skrz ďalšie prídavné moduly. Väčšina základných modelov ma 1.35“ displej s cieľom základnej diagnostiky. Okrem toho je možné skrz PLC sledovať takmer všetky premenné uložené v PLC. Dané PLC S7-1511T-1 PN je radené medzi základné PLC ,ktoré je vhodné pre aplikácie so strednými nárokmi na výpočet. Systémové označenie „T“ značí technologické využitie. V danom PLC sú integrované bloky ako motion control a tiež má k dispozícii operačnú pamäť o veľkosti 225 kB a tiež 1MB pamäte určené pre dáta prostredníctvom pamäťovej karty SD.

K PLC tiež máme pripojený ďalší DI/DO modul na DIN lište, nakoľko samotné PLC neobsahuje žiadne extra vstupy/výstupy. Daný modul je pripojený hneď vedľa a má k dispozícii 16 vstupov a výstupov.

Okrem digitálnych vstupov a výstupov je nutné PLC doplniť ešte pamäťovou kartou, nakoľko základná pamäť nespĺňa požiadavky stability. Pri spustení PLC si



proces načíta dáta z danej pamäťovej karty a predstavuje podstatne stabilnejší prvok v systéme ako pamäť PLC. Ďalšia výhoda pamäťovej karty je to, že danú kartu je možné prenášať medzi viacerými PLC zariadeniami.



Obr. 2.1.1.1 PLC S1511-T [14]

## 2.1.2 SINAMICS S210

Jedná sa o nástupcu frekvenčného meniča S110. Daný model ešte rozširuje jeho možnosti. Aplikácie tohto meniča sú primárne také, ktoré nevyžadujú dlhodobý plynulý pohyb v stredných a vysokých požiadavkách na riadenie rýchlosti, polohy, momentu a koordinácie vyššieho množstva os. Pre daný model frekvenčného meniča sú typické aplikácie ako lisy, valcovacie stolice alebo navíjačky. Daný FM obsahuje aj kompletnú bezpečnostnú sadu funkcií podľa normy DIN EN 31800-5-2, a český ekvivalent ČSN EN 61800-5-2 (351720).

Výstupné napätie daného frekvenčného meniča je 200 až 240 V 1AC pri maximálnom prúde 1.4A s výstupnou frekvenciou 0 až 550 Hz. Výkonový modul má 0.1 kW s krytím IP20, ktoré tvorí krytie voči zvisle dopadajúcim kvapkám vody. Riadiaca jednotka ponúka päť digitálnych vstupov, z ktorých dva môžu byť použité ako bezpečnostné funkcie.



Obr. 2.1.2.1 Frekvenčný menič S210 [16]

Daný menič ponúka aj Safety funkcie. Má k dispozícii jeden fail-safe (F-DI) a dva rýchle Safety digitálne vstupy. K dispozícii je 10 bezpečnostných funkcií – 3 sú súčasťou

základných funkcií, ktoré slúžia na bezpečné zastavenie a bezpečné brzdenie. Ďalšie funkcie nájdeme v rozšírenej ponuke funkcií a prístup k nim je limitovaný Safety licenciou. Všetky tieto funkcie vyžadujú firmware meničov V5.1 SP1+. Jedná sa napr. o funkcie limitácie rýchlosti, bezpečného rozbehu, limitácie zrýchlenia a bezpečného brzdenia. Nakoľko v našom programe Safety funkcie nepoužívame, tak bližšie informácie si v prípade potreby dohľadáme v manuáli pre menič. Dôvodom, prečo nemáme žiadne bezpečnostné prvky v našej aplikácii je hneď niekoľko. Výkony pohonov, resp. motorov sú malé a hodnoty mechanického namáhania sú obdobne nízke, takže ochrana ako taká v tomto prípade nedáva zmysel. Ďalší dôvod je, že naše PLC tieto bezpečnostné funkcie ani nemá. Každé PLC, ktoré má centrálny Safety funkcie má označenie F na konci popisu.

### 2.1.3 Servomotor SIMOTICS S-1FK2

K pohonom S210 je vhodné použiť motory z rady 1FK2. Daná rada ponúka množstvo výkonových stupňov od 0.05 kW až po 7 kW. Dané motory boli vyvíjané od začiatku spolu s pohonom S210. Dané motory sú typovo synchronne motory s permanentnými magnetmi (PMSM) a enkodérom. Nami používané motory majú nasledovné charakteristiky:

Menovitý výkon	0.05	kW
Menovité otáčky	3000	ot/min
Menovitý moment	0.14	Nm

Tab. 2.1.3.1 Štítkové hodnoty motora

Ďalej dané motory majú enkóder AM22DQC. Jedná sa o absolútny enkóder 22bit + 12bit multiturn. Výstupom daného motora je absolútny uhol natočenia rotora v určenom rozlíšení. Tiež je možné pomocou neho počítať až 4 096 otáčok pomocou dodatočných 12 bitov.



Obr. 2.1.3.1 Servomotor S-1FK2 [17]



## 2.1.4 SIMATICS HMI panel

K našej aplikácii máme fyzicky dostupný KTP400 Basic. Jedná sa o veľmi základný panel, ktorý ponúka komplexne všestranné riešenia pre mnohé aplikácie. Daný panel je ponúkaný v 4“ až 12“ rozlíšení a jedná sa stále o TFT displej so 64 000 farbami a USB vstupom pre pripojenie klávesnice, myši alebo iných zariadení. Dané HMI ponúka max. 800 využiteľných tagov v jednom komunikačnom rozhraní. Vďaka USB je tiež možné ukladať sledované dáta.

Náš konkrétny panel má 4“ displej (pomer strán 16:9). Pri rozbore označenie KTP400 zistíme, že označenie KTP znamená „Key Touch Panel“, čo predstavuje dotykový displej. Typovo sa jedná o kapacitný displej a teda reaguje na tepelnú citlivosť a nie na odpor. Tento panel má navyše k dispozícii ešte 4 hardverové tlačidlá (F1-F4), ktorými je možné uľahčiť operáciu s displejom.



Obr. 2.1.4.1 HMI KTP400 panel [18]

Nakoľko daný panel nepodporuje funkcie ktoré potrebujeme, v praktickej časti aplikácie pracujeme s virtuálnym HMI panelom, ktorý bude dostupný na počítači. Jedná sa o panel TP700 Comfort. Jedná sa o 7“ Comfort panel s rozlíšením displeju 800x480 pixelov so 16 mil. farieb.



Obr. 2.1.4.2 HMI TP700 Comfort panel [20]

## 2.2 TIA portál

TIA alebo Totally Integrated Automation portál je program od SIEMENSu s cieľom vytvoriť jedno univerzálne programové prostredie, v ktorom je možné takmer všetky výrobky od SIEMENSu programovať spoločne a zjednodušiť celú inicializáciu a prepojenie medzi sebou. Okrem toho TIA ponúka aj možnosť diagnostiky, simulácií, Safety funkcií a mnoho ďalšieho. TIA portál sa skladá z troch hlavných častí:

- WinCC – slúži na návrh a programovanie HMI
- StartDrive – slúži na konfiguráciu meničov
- STEP 7 – slúži na programovanie PLC a konfiguráciu snímačov a monitoringu, sieťových prvkov a pod.

V našej aplikácii pracujeme s verziou 16, nakoľko sa jedná o poslednú dostupnú verziu, ktorú máme k dispozícii. V ďalšej tabuľke uvedieme prehľad aktuálnych verzií programu a jeho súčastí, s ktorými pracujeme.

Názov	Verzia	Názov	Verzia
Automation License Manager	V6.0 + SP8	SINAMICS Startdrive S210	V16, U5
TIA portal	V16, U3	STEP 7 Professional	V16, U3
TIA portal Control Interface	V16, U2	STEP 7 Safety	V16
TIA portal Openess	V16, U3	WinCC Advanced	V16, U3
SINAMICS Startdrive	V16, U5		

Tab. 2.2.1 Použitý software

Ako sme už vyššie spomenuli, TIA je plne integrovaný program na takmer všetky súčasti od SIEMENSu. To nám poskytuje výhodu, že v jednom prostredí konfiguruje a programujeme PLC, menič s motorom a aj HMI panel. TIA portál ponúka mnoho rôznych pohľadov, prehľadov, typov programovania a pod.

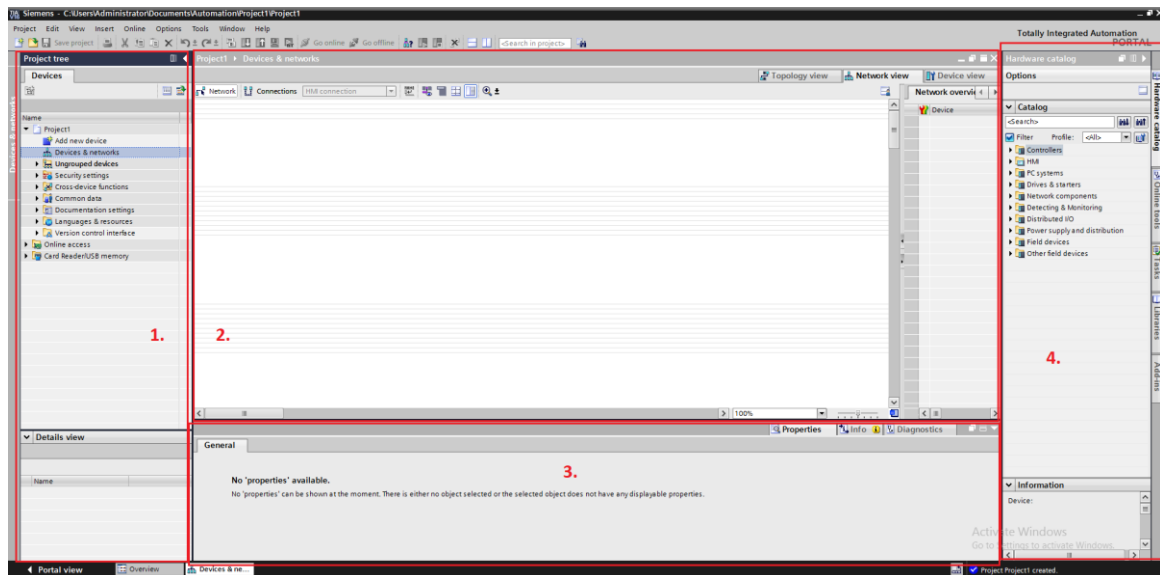
Pri spustení TIA portálu V16.0 sa nám hneď naskytne možnosť vytvoriť nový projekt, ktorý hneď využijeme a založíme projekt. Následne sa ocitneme v Portal View. Alternatívou je možnosť sa prepnúť do Project View, ktorá je podrobnejšia. V našom prípade nám lepšie vyhovuje podrobnejší pohľad, takže sa prepneme do tohto pohľadu. V danom pohľade sa nám naskytnú 4 hlavné okná.

Prvé okno predstavuje štruktúru programu a jednotlivých súčastí projektu – PLC, HMI, meniče, hlavný programový blok, bloky funkcií a dátové bloky, diagnostika, sledovacie tabuľky a pod. vo forme zoznamu, kde po rozkliknutí sa nám zobrazí obsah.

V druhom okne sa nám zobrazí aktivovaná časť projektu (napr. otvorené programové bloky, konfigurácie meničov, konfigurácia HMI a pod.). V tomto okne dochádza ku hlavnej činnosti programátora a môžeme ho nazvať ako hlavnú pracovnú plochu.

V treťom okne máme zvyčajne vlastnosti a všeobecné informácie aktivovaného objektu. Tu môžeme meniť vlastnosti objektu, ktorý je aktívny v okne 2, resp. na hlavnej ploche.

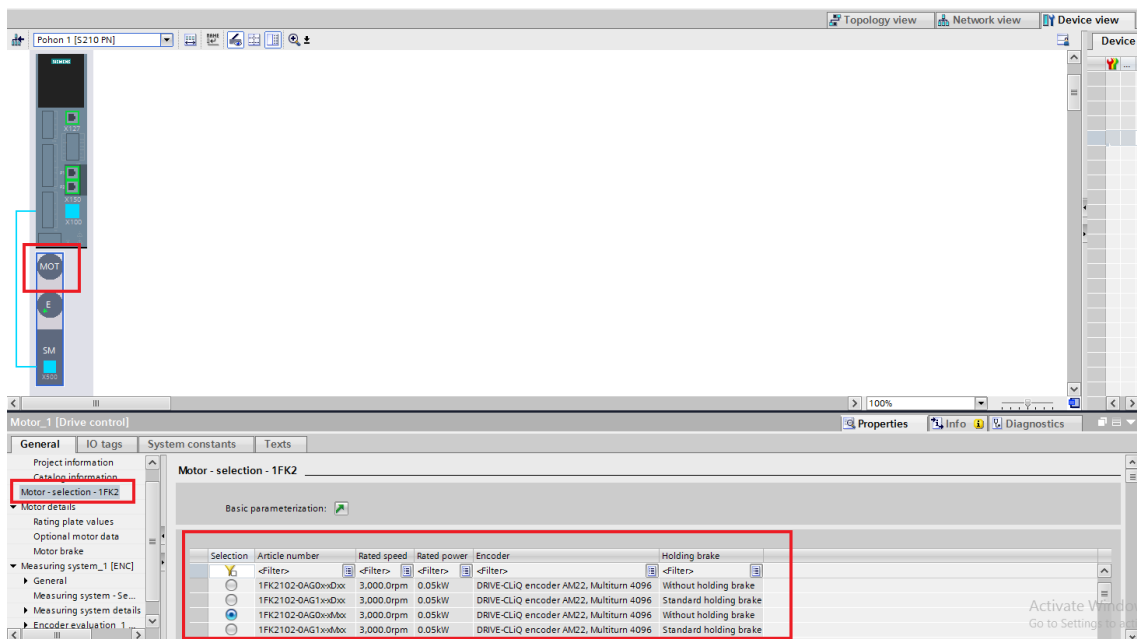
A posledné, štvrté okno slúži ako katalógové okno s hardwarom, knižnicami, inštrukciami a funkciami do programu (logické, matematické, časovače a pod.). Dané prvky vieme jednoduchých potiahnutím (Drag&Drop) požadovaného prvku do prvého alebo druhého okna použiť v našom programe.



Obr. 2.2.1 TIA portál - Pracovné prostredie

## 2.2.1 Popis a konfigurácia HW

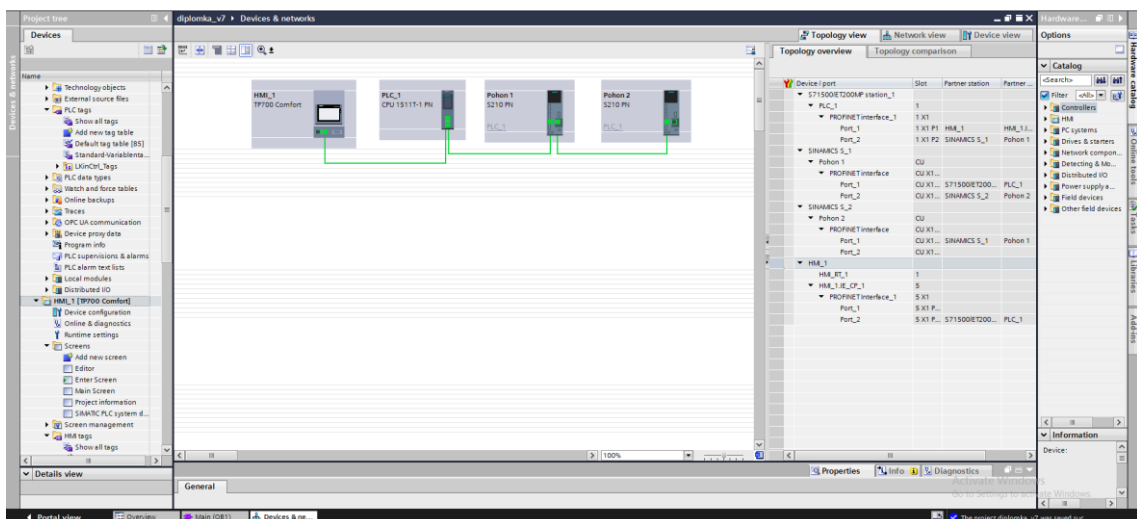
Ďalší krok po vytvorení projektu je pridať doň používaný hardware, ktorý plánujeme použiť v našom projekte. V prvom okne zvolíme „Add new device“ a vyberieme si PLC, ktoré máme k dispozícii. Ak nevieme presné označenie, môžeme použiť „Unspecified CPU 1500“ a po pripojení cez Ethernet sa nám sám identifikuje. Po úspešnom vložení môžeme pridať z katalógu panel so 16 digitálnymi vstupmi a výstupmi DI/DO, ktorý osadíme na lištu hneď vedľa PLC (opäť vieme použiť princíp Drag&Drop). Po úspešnom vložení ďalej pridáme frekvenčné meniče – dvakrát S210. Následne v hlavnom pracovnom okne si vyberieme „Device view“ – nájdeme to v pravom hornom rohu pracovnej plochy. Ďalším krokom bude výber motora. Pravým klikom a zvolením „Properties“ na ikonu „MOT“, ktorú nájdeme na nami vloženom pohone v pracovnom okne. Následne zo zoznamu vo vlastnostiach vyhladáme „Motor – selection“, kde si vyberieme motor, ktorý máme k dispozícii. Týmto spôsobom vložíme 2 rovnaké pohony. Na ďalšom Obr. 2.2.1.1 môžeme vidieť pohľad meniča a výber motora.



Obr. 2.2.1.1. TIA portál - Výber motora pre menič

Posledným vloženým zariadením bude HMI. Rovnakým spôsobom ako sme vložili meniče alebo PLC, vložíme aj HMI. Konkrétne máme SIMATIC Comfort panel so 7“ –TP 700 Comfort. Následne v hlavnom okne v „Topology view“ spojíme PLC s HMI displejom a jednotlivými pohonnými jednotkami a upravíme cez vlastnosti IP adresy. PLC necháme na adrese 192.168.0.1 a jednotlivé pohony nastavíme na 192.168.0.3 a 192.168.0.4 a HMI nastavíme na 192.168.0.2. Takto by sme už mali mať jednotlivé súčasti prepojené cez Ethernetovú sieť PROFINET.

Po vložení všetkých súčastí by to malo vyzerať ako na obrázku.



Obr. 2.2.1.2 TIA portál - Topologický pohľad

V karte „Network view“ je dobré ešte skontrolovať, že všetky súčasti sú prepojené medzi sebou a nehlási to chybu s IP adresou a pod.

### 2.2.1.1 Konfigurácia PLC

V tejto časti sa budeme venovať konfigurácii programovateľného logického kontroléra. Prídavný DI/DO modul sme už pridali, teraz ešte bude potrebné nastaviť komunikačný kanál. Ten nastavíme cez vlastnosti PLC v podsekcii „Profinet Interface [X1] – Advanced options – Real time settings - IO communication“ a nastavíme to na 4ms. V „Real time options“ sa dopočíta a zobrazí čas jedného taktu spolu s odhadnutým vytážením. Vypočítané zaťaženie je 0,554 %, takže priestor na rozšírenie aplikácie určite existuje a je naozaj veľký. Ďalším krokom bude nastaviť „Synchronization“. Nachádza sa to v tej istej podsekcii a nastavíme to na „Sync-Domain\_1“. Jedná sa o komunikáciu PROFINET s taktom 4ms a ešte nastavíme rolu PLC na „Master“. Pri konfigurácii PLC ešte aktivujeme v sekcii „System and clock memory“ systémové aj časové bity. Preddefinované adresy nemeníme.

### 2.2.1.2 Konfigurácia meničov

Podobným spôsobom ako sme nakonfigurovali PLC, tak potrebujeme nastaviť aj oba meniče. Ako bolo spomenuté vyššie, k meniču je nutné zvoliť motor. Ten nastavíme pravým klikom na ikonu motora v pohone, ku ktorému sa dostaneme v „Device view“. V našom prípade sa jedná o motor 1FK2. V tabuľke vyššie sme špecifikovali už katalógové číslo. Jedná sa o model 1FK2102-0AG0x-xMxx. Jedná sa o synchronne servomotory s permanentnými magnetmi o výkone 0.05 kW, menovitých otáčkach 3000 ot/min a menovitej prúde 0.75A.

Ďalej je potrebné nastaviť komunikáciu. Podobne ako v PLC, otvoríme si vlastnosti pohonu a zvolíme „Synchronization“. Tu sa uistíme, že máme nastavené v RT class – IRT. Taktiež sa v „Isochronous mode“ uistíme, že máme zapnutý izochrónny mód. Posledná vec, ktorú si môžeme skontrolovať sú telegramy. Ako sme ich opísali v podkapitole 1.4, daný pohon a PLC komunikuje pomocou Telegramu 105, ktorý je cyklicky odosielaný a prijímaný. V prípade potreby je možné dokonfigurovať aj Safety funkcie, ktoré sú potrebné najmä pre priemyselné aplikácie, resp. na všetky, kde vzniká nejaké nebezpečenstvo pri bežnej prevádzke alebo v prípade chybového stavu. Safety zahŕňa viaceré aspekty ako sú preddefinované funkcie pre zastavenie, ale aj prípadná chybovosť a rýchlosť digitálnych vstupov. Cieľom týchto všetkých Safety opatrení je minimalizovať riziko nehody a zaistenie čo najbezpečnejšieho správania stroja, aby nedošlo k zraneniu na osobách, ale aj majetku.

## 2.2.2 PLC programovanie

TIA portál ponúka viaceré programovacie možnosti od jednoduchších po komplikovanejšie. Najjednoduchší z nich je STL - je najprimitívnejší z ponúkaných možností a podobá sa jazyku symbolických adres. Ďalší jazyk je SCL. Je mierne sofistikovanejší a základ vychádza z Pascalu. Je to podstatne štruktúrovanejšie programovanie a ponúka viacero prispôbení pre využitie s PLC – volanie blokov, čítača a pod. Tretí spôsob programovania je FBD. Jedná sa o grafický programovací jazyk, ktorý popisuje vstupy a výstupy, ktoré zapája do väčších celkov. Ďalšia možnosť je LAD. Je to grafický rebríkový diagram, ktorý opisuje kontaktné schémy. Posledný z možností a zároveň najmodernejší je S7-Graph. Je to znova grafický programovací jazyk, ktorý je vhodný napr. na popis sekvenčných cyklov. V tabuľke nižšie môžeme vidieť ekvivalenty od SIEMENSu k aktuálnym technickým normám.

Norma IEC 61131-3	SIEMENS ekvivalent (nie náhrada)
Instruction list (IL)	STL
Structured text (ST)	SCL (S7-SCL)
Function Block Diagram (FBD)	FBD
Ladder diagram (LAD)	LAD
Sequential Function Chart	S7-Graph

Tab. 2.2.2.1 IEC normy a SIEMENS ekvivalenty

SIEMENS ekvivalenty pre normované jazyky spĺňajú európsku normu (následne to prebrala aj ČR). Pomocou tejto normy je zjednotená syntax aj pravidlá pre kombinácie symbolov s cieľom zjednotenia v priemysle a aj pri zvyšných aplikáciách. V norme sú ďalej definované spoločné prvky – deklarácie dátových typov, funkčných blokov a funkcií.

V našej aplikácii používame najmä programovanie v jazyku FBD. Každopádne všetky importované funkcie sú naprogramované v SCL a pomocou FBD jazyka ich spájame do funkčného celku.

Celkový program v TIA portáli je rozdelený na 4 základne typy:

- OB – organizačné bloky volané operačným systémom/programom
- FC – funkcie bez pamäte
- FB – funkčné bloky so stavovou pamäťou s prídruženým dátovým blokom
- DB – dátové bloky, držia dáta organizované v štruktúrach

Každý z týchto elementov má unikátne označenie v danom projekte, ktoré je síce možné meniť, ale dva objekty nemôžu mať rovnaké označenie. Organizačné bloky tvoria rozhranie medzi programom a PLC a povolané sú podľa situácie (udalosti) napr.:

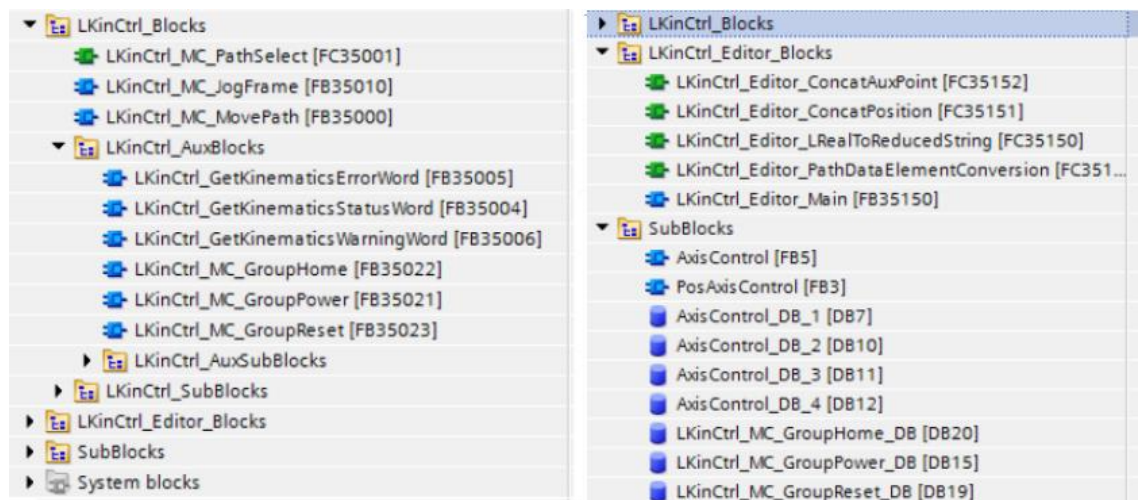
- Main block – OB1 – cyklický program
- Startup – OBxxx – povolané v momente keď sa PLC reštartuje
- Cyclic / Time interrupt – OBxx – volané v pravidelných časových intervaloch

- Programming / IO access error – volané pri chybe

Okrem týchto základných OB máme v našom programe aj špeciálny na riadenie servopohonu – MC-Servo (OB91).

### 2.2.2.1 Použité knižnice

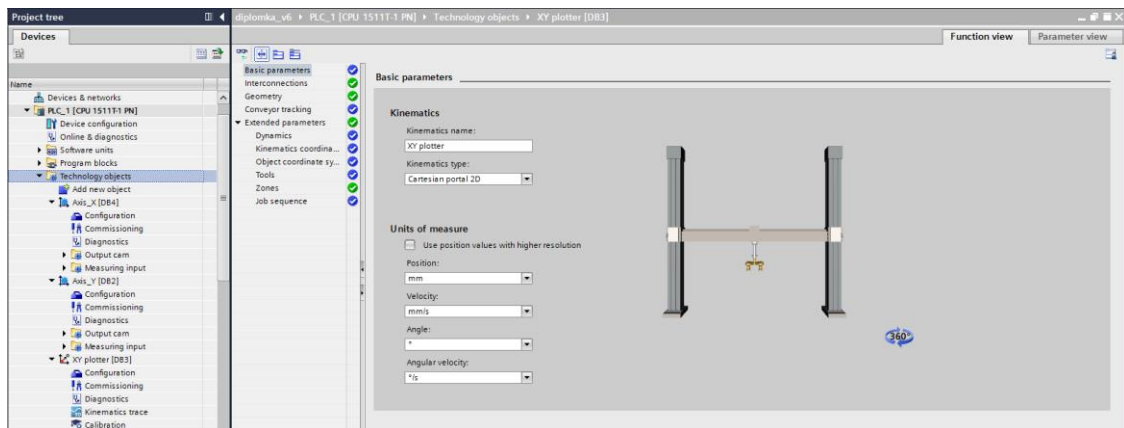
Naša aplikácia je založená na knižniciach Kinematics a následne ešte LKinCtrl - je nadstavba na základný Kinematics, ktorý ešte ponúka navyše editor, v ktorom je možné zadeinovať dráhu požadovaného pohybu („PathData“). Na nasledujúcich obrázkoch môžeme vidieť všetky importované bloky. Samotný program obsahuje okrem týchto blokov ešte hlavný blok (OB), novovytvorené bloky na kreslenie (OB, FB, 2x DB). Okrem týchto tu máme ešte 3 uzamknuté OB, keďže sa jedna o „know-how“ firmy SIEMENS. Sú to bloky MC-Interpolator, MC-LookAHead a MC-Servo (ako sme už vyššie spomenuli, tento blok slúži na riadenie servomotora). Tieto bloky sa automaticky importujú na základe použitých TO a zariadení, ktoré si do projektu vložíme.



Obr. 2.2.2.1.1 TIA portál - Importované funkcie

### 2.2.2.2 Technologické objekty

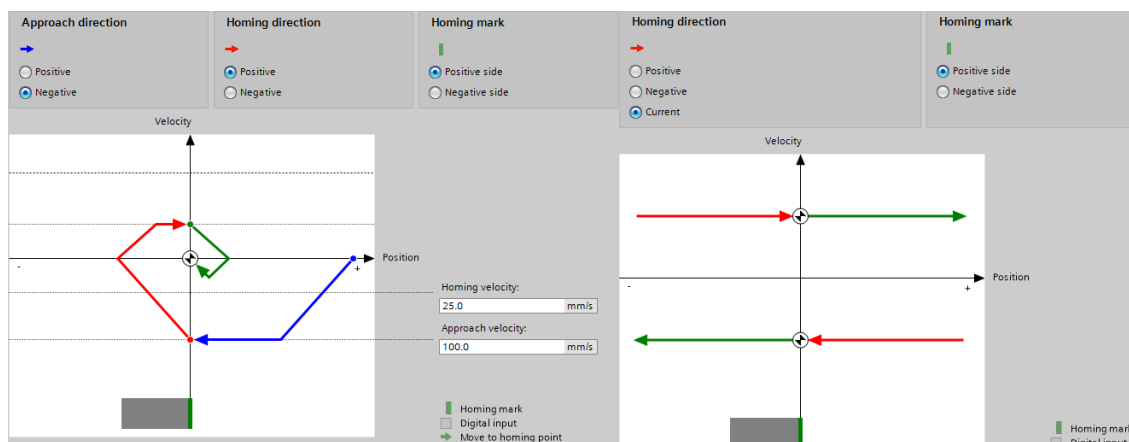
V našej aplikácii máme použité 3 technologické objekty – dve polohovacie osi a Kinematics. Teraz sa pozrieme na ich základne nastavenie. Začneme s nastavením TO Kinematics. Po tom ako sme si ho vložili do projektu, tak po rozkliknutí začneme s nastavením základných parametrov. Na ďalšom obrázku vidíme použité technologické objekty v projekte, kde jedna polohovacia osa sa nazýva Axis X a druhá Axis Y, ktoré zabezpečujú pohyb v jednotlivých osách podľa názvu. Prvok, ktorý ich následne ovláda je Kinematics. Na obrázku vidíme konfiguračné okno TO Kinematics.



Obr. 2.2.2.2.1 TIA portál - TO a konfigurácia Kinematics

Ďalej v „Interconnections“ si vyberieme jednotlivé osi (vyberáme z polohovacích TO). Tretiu osu a orientáciu nechávame nezadefinované. V ďalších častiach konfigurácie zadefinujeme rozmery na základe nášho HW a jednotlivé dynamické vlastnosti necháme na predvolených hodnotách. Pri implementácii na reálny HW ich modifikujeme podľa reálnych hodnôt. V rámci konfigurácie si vieme zadefinovať aj pracovné zóny a nástroje.

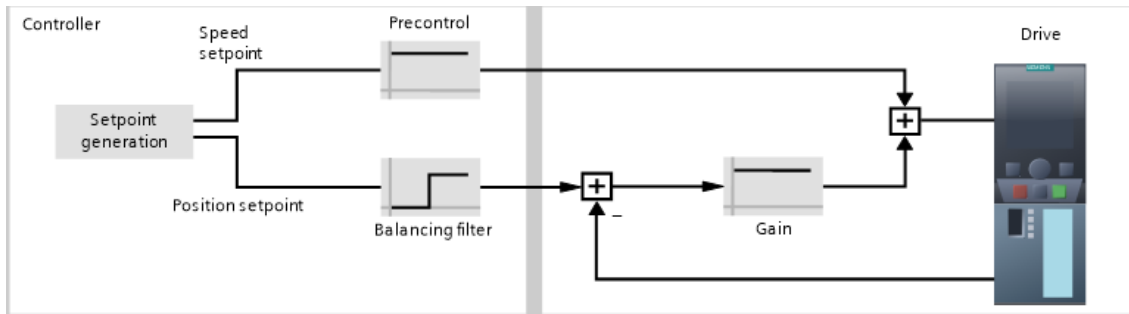
Konfigurácie polohovacích os sú takmer totožné pre obe, jediný rozdiel je, že smer pohybu v osi X má lineárny charakter a smer v osi Y má rotačný charakter. Keďže plánujeme simulácie na našej aplikácii, je nutné hneď na konci prvej strany zaškrtnúť „Activate simulation“. Všetky jednotky máme nastavené v mm, Nm alebo N. V „Hardware interface“ vyberieme požadovaný pohon. Typ pohonu aj komunikácie sa nastaví automaticky. Ďalej zaškrtneme enkóder 1. V ďalšej časti si vyberieme v „Drive telegram“ štandardný telegram 105. Ten istý zvolíme aj pre komunikáciu s enkóderom. Následne nastavíme momentový limit, aby sa meral na strane motora. Ďalej nastavíme „Homing“ – môžeme to vidieť na ďalšom obrázku.



Obr. 2.2.2.2.2 TIA portál - Aktívny a pasívny Homing

A posledná vec, ktorú aktivujeme, je DSC. Túto možnosť nájdeme v konfigurácii pohonu pod „Control loop“ a zaškrtneme „Position control in the drive (DSC enabled)“. Následne zvolená možnosť vyzerá nasledovne.



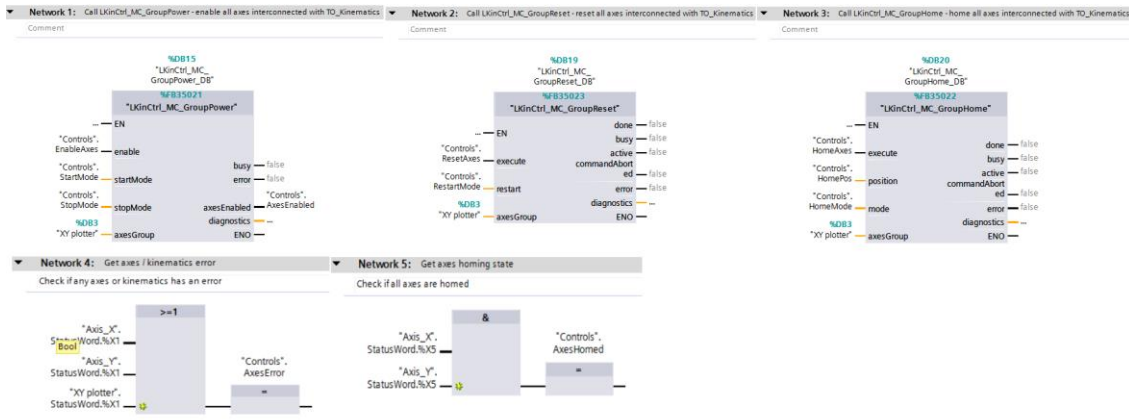


Obr. 2.2.2.2.3 TIA portál - Zapnuté DSC

DSC (Dynamic Servo Control) umožňuje preniesť časť polohovej regulácie do pohonu. V prípade, že DSC je vypnuté, tak hranica regulácie je hneď za meničom pohonu. Týmto sme nastavili všetky použité TO v projekte. Poslednú vec, ktorú nastavíme sú HW limity, na základe reálneho HW a pomocou tagov ich zdefinujeme.

### 2.2.2.3 Bloky hlavného programu

Blok hlavnej sekvencie Main je rozdelený do piatich podsietí, kde využívame 3 FB z importovanej knižnice *LKinCtrl – GroupPower, GroupReset* a *GroupHome*. Následne ešte v Main bloku máme informáciu o errore a home pozícii na základe statusových slov jednotlivých os.



Obr. 2.2.2.3.1 TIA portál - Hlavná sekvencia programu

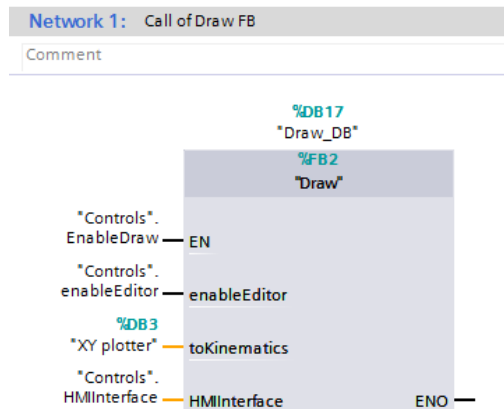
Funkcia *LKinCtrl\_MC\_GroupPower* uvoľní alebo blokuje všetky osi, ktoré sú prepojené k skupine osí - *axesGroup*. Ekvivalentné k tomu je spustenie *startMode* a *stopMode* pre danú skupinu os. Ďalšia z funkcií je *Mc\_GroupReset*, ktorá slúži k reštartovaniu všetkých os pripojených k danej skupine os. Posledná preddefinovaná funkcia je *MC\_GroupHome*. Táto funkcia slúži na nastavenie všetkých os v skupine na *home* pozíciu. Posledné dve funkcie slúžia k signalizácii chyby a k signalizácii *home* pozície daných osí.

Ďalej tu máme dátový blok „Controls“, ktorý sme vytvorili s cieľom riadiť jednotlivé pokyny programu a ktoré ďalej využívame v HMI cez tagy alebo v iných častiach programu.

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
Static									
EnableDraw	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			control variable to enable drawing
EnableAxes	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			control variable to enable axes with MC_POWER
AxesEnabled	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			status variable to see all axes enabled
ResetAxes	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			control variable to reset axes with MC_RESET
AxesError	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			status variable to see error on one axis
HomeAxes	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			control variable to home axes with MC_HOME
AxesHommed	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			status variable to see all axes hommed
HomePos	Array[1..4] of LReal								control variable storing homing positions for MC_GroupHome
HomeMode	Array[1..4] of Int								control variable storing homing modes for MC_GroupHome
RestartMode	Array[1..5] of Bool								control variable storing restart modes for MC_GroupReset
StartMode	Array[1..4] of DInt								control variable storing start modes for MC_GroupPower
StopMode	Array[1..4] of Int								control variable storing stop modes for MC_GroupPower
WritePathData	Bool	false							control variable to write PathData for pick and place scenario
AutoAcknowledge	Bool	TRUE							control variable to auto acknowledge wait flags in pick and place s.
StartPath	Bool	false							control variable to start a scenario
P_TRIG_flag_0	Bool	false							storage variable for positive trigger flag 0
P_TRIG_flag_1	Bool	false							storage variable for positive trigger flag 1
enableEditor	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			control variable to enable editor FB
HMIInterface	*LKinCtrl_Editor_typeHmiInt...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		interface variable to HMI for editor
ActivatePEN	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		activating pen

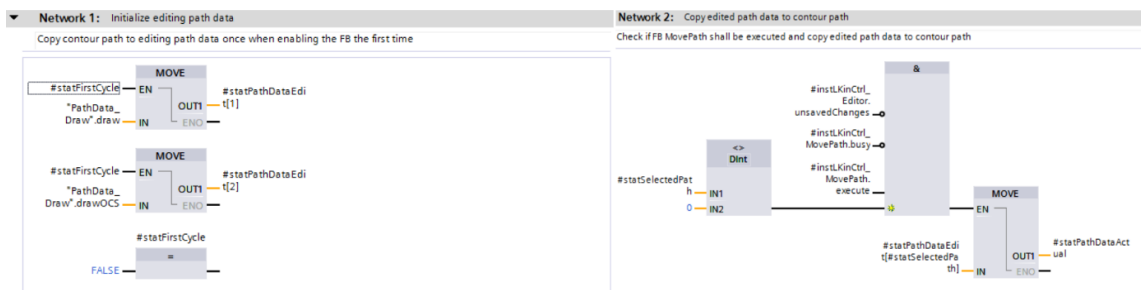
Obr. 2.2.2.3.2 TIA portál - Dátový blok Controls

Ďalšia zložka je zložená z organizačného bloku, funkcie a dvoch dátových blokov. Jedná sa o funkciu „Draw“, ktorá je založená na jednej zo základných aplikácií od SIEMENSu. Jednotlivé bloky môžeme vidieť na nasledujúcich obrázkoch. Vstupujú do nej najmä premenné z dátového bloku a samotný technologický objekt.

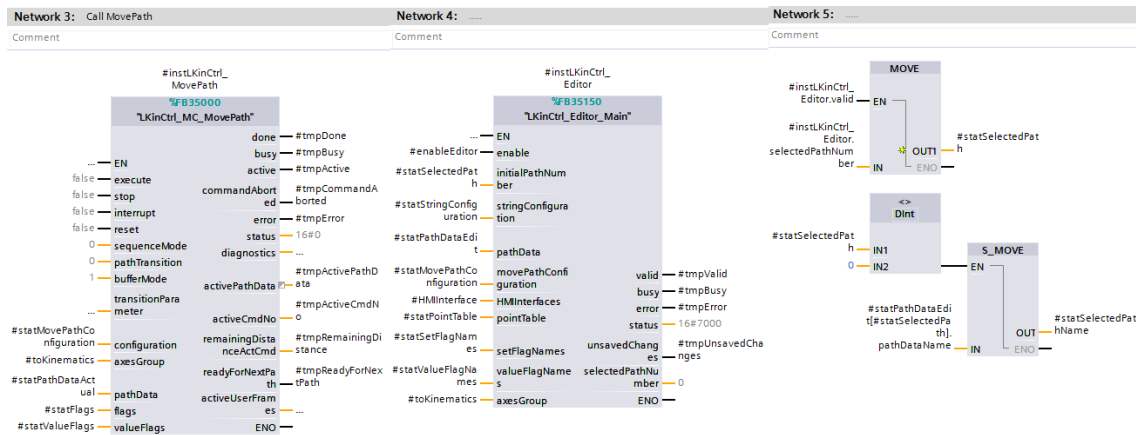


Obr. 2.2.2.3.3 TIA portál - Organizačný blok Draw

V ďalšej časti nájdeme FB Draw a jednotlivé siete, do ktorých sú rozdelené kvôli prehľadnosti. Tieto funkcie už boli naprogramované spoločnosťou SIEMENS a následne naimportované do projektu a prikladám ich z dôvodu úplnosti celého kódu.

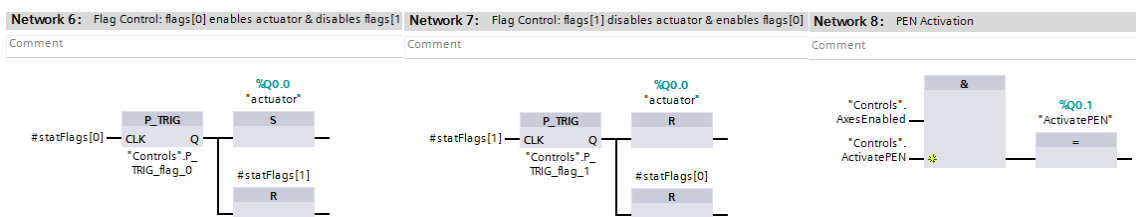


Obr. 2.2.2.3.4 TIA portál - Funkčný blok Draw 1/3



Obr. 2.2.2.3.5 TIA portál - Funkčný blok Draw 2/3

V druhej časti vidíme *LKinCtrl\_MC\_MovePath*, ktorý už názvom indikuje, že zabezpečuje pohyb nástroja. Táto funkcia organizuje a zabezpečuje celkovú funkčnosť aplikácie podľa požiadaviek operátora a tiež umožňuje aktiváciu relé alebo iných zariadení v závislosti na aktuálnej pozícii v rámci programu pomocou flagov. Druhá funkcia – *LkinCtrl\_Editor\_Main* slúži priamo na prepojenie s HMI a výmenu dát.



Obr. 2.2.2.3.6 TIA portál - Funkčný blok Draw 3/3

Na Obr. 2.2.2.3.6 môžeme vidieť aktiváciu a deaktiváciu digitálneho výstupu na základe aktivovaného alebo deaktivovaného flagu, ktorý vychádza z predošlých funkcií. Ďalší prvok je dátový blok Draw, ktorý je spätý s FB Draw, nakoľko ten nemá žiadnu pamäť. Na obrázku vidíme všetky zadefinované premenné s celou charakteristikou.

Name	Data type	Start value	Retain	Accessible f.	Write...	Visible in ...	Setpoint	Supervis...	Comment
1	Input								
2	enableEditor	Bool	false						TRUE: Enable functionality of FB
3	Output								
4	InOut								
5	toKinematics	TO_Kinematics							reference to TO_Kinematics
6	HMIInterface	"LKinCtrl_Editor_typeHMIInterface"							reference to HMI Interface
7	Static								
8	statFirstCycle	Bool	TRUE						static to store the first program cycle
9	statSelectedPath	Dint	1						static to store selected path
10	statSelectedPathName	String["LKINCTRL_LENGTH_OF_PATHDATA_NAME"]							
11	instLkinCtrl_MC_MovePath	"LKinCtrl_MC_MovePath"							Core function block of LKinCtrl library
12	instLkinCtrl_Editor	"LKinCtrl_Editor_Main"							This function block offers editing a path
13	statFlags	Array[0..LKINCTRL_NO_OF_LAST_SETFLAG] of Bool							static storing flag information
14	statValueFlags	Array[0..LKINCTRL_NO_OF_LAST_VALUEFLAG] of LReal							static storing value flag information
15	statMovePathConfigur...	"LKinCtrl_typeConfiguration"							static storing MovePath configuration
16	statPointTable	Array[0..10] of "LKinMctrl_typePointTable"							static storing the PointTable
17	statStringConfiguration	"LKinCtrl_Editor_typeStringConfiguration"							static storing the String configuration
18	statPathDataEdit	Array[1..2] of "LKinCtrl_typePathData"							static storing the pathData for editing
19	statPathDataActual	"LKinCtrl_typePathData"							
20	statSetFlagNames	Array[0..LKINCTRL_NO_OF_LAST_SETFLAG] of WString["LKINCTRL_LENGTH_OF_FLAG_NAME"]							static to store set flag names
21	statValueFlagNames	Array[0..LKINCTRL_NO_OF_LAST_VALUEFLAG] of WString["LKINCTRL_LENGTH_OF_FLAG_NAME"]							static to store value flag names

Obr. 2.2.2.3.7 TIA portál - Dátový blok Draw

Posledná súčasť je PathData Draw, ktorý slúži na definovanie pohybu.

PathData_Draw (snapshot created: 8/30/2021 10:41:20 AM)										
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Comment
1	Static									
2	draw	"LKinCtrl_typePathData"								
3	drawOCS	"LKinCtrl_typePathData"								

Obr. 2.2.2.3.8 TIA portál - Dátový blok PathData

Všetky zvyšné funkcie a dátové bloky z knižnice *LKinCtrl* máme importované pre úplnú funkčnosť aplikácie. Funkčnosť a prepojenie medzi HMI a PLC je zabezpečené pomocou *HMIInterface* v „Controls“ s dátovým typom *LKinCtrl\_Editor\_typeHMIInterface*. Jedná sa o dátový typ zadaný v importovanej knižnici. Následne to používame ako vstup do funkcie Draw.

## 2.2.2.4 Tags – PLC a HMI

Tagy v TIA programe fungujú na prácu so signálmi a informáciami a predstavujú symbolické konštanty, ktoré sa používajú skrz celý program. Po priradení tagov na digitálne vstupy a výstupy (DI/DO) dokážeme tieto signály použiť v jednotlivých programoch pomocou premenných definovaných v dátových blokoch. Na Obr. 2.2.2.4.1 môžeme vidieť PLC tagy, kde okrem názvu a dátového typu ešte máme adresu. Jedná sa o fyzické adresy vstupov a výstupov. Konkrétne na obrázku vidíme komunikáciu a jednotlivé signály cez Telegram 105, ktorý sme opísali v teoretickej časti práce. Pod adresou sú jednotlivé HW vstupy a výstupy.

Standard-Variablenabelle										
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervis...	Comment	
1	actuator	Bool	%Q0.0							
2	Axis_X_Actor_Interface_Addres...	"PD_TEL105_IN"	%Q256.0							
3	Axis_X_Actor_Interface_Addres...	"PD_TEL105_OUT"	%Q256.0							
4	Axis_X_Sensor1_Interface_Add...	"PD_TEL105_IN"	%Q256.0							
5	Axis_X_Sensor1_Interface_Add...	"PD_TEL105_OUT"	%Q256.0							
6	Axis_Y_Actor_Interface_Addres...	"PD_TEL105_IN"	%Q276.0							
7	Axis_Y_Actor_Interface_Addres...	"PD_TEL105_OUT"	%Q276.0							
8	Axis_Y_Sensor1_Interface_Add...	"PD_TEL105_IN"	%Q276.0							
9	Axis_Y_Sensor1_Interface_Add...	"PD_TEL105_OUT"	%Q276.0							
10	ActivatePEN	Bool	%Q0.1							
11	<Add new>									

Obr. 2.2.2.4.1 TIA portál - PLC tagy technologických objektov

Na ďalšom obrázku vidíme nami vytvorené HMI tagy, ktoré priamo používame v HMI interface. Dátové typy sú totožné s PLC tagmi, pretože tak sú zadané v dátových typoch a zachovávajú si tento typ skrz celý program. Ďalej potrebujeme zvoliť „Connection“. Tu máme možnosť vybrať PLC, s ktorým chceme spárovať dané HMI. Po rozkliknutí sa nám otvoria všetky dostupné PLC v danom projekte. To následne vidíme aj v „PLC name“, s ktorým PLC to máme spárované a v ďalšom stĺpci máme „PLC tag“ a vidíme aj párovanie s tagmi z PLC.

Default tag table									
Name	Data type	Connection	PLC name	PLC tag	Address	Access mode	Acquisition cycle	Logged	Source comment
Controls_ActivePen	Bool	HMI_Connection_2	PLC_1	Controls.ActivatePEN		<symbolic access>	1 s		activating pen
Controls_AxesEnabled	Bool	HMI_Connection_2	PLC_1	Controls.AxesEnabled		<symbolic access>	1 s		status variable to see all axes enabled
Controls_AxesError	Bool	HMI_Connection_2	PLC_1	Controls.AxesError		<symbolic access>	1 s		status variable to see error on one axis
Controls_AxesHomed	Bool	HMI_Connection_2	PLC_1	Controls.AxesHomed		<symbolic access>	1 s		status variable to see all axes homed
Controls_EnableAxes	Bool	HMI_Connection_2	PLC_1	Controls.EnableAxes		<symbolic access>	1 s		control variable to enable axes with MC_POWER
Controls_EnableDraw	Bool	HMI_Connection_2	PLC_1	Controls.EnableDraw		<symbolic access>	1 s		control variable to enable drawing
Controls_enableEditor	Bool	HMI_Connection_2	PLC_1	Controls.enableEditor		<symbolic access>	1 s		control variable to enable editor FB
Controls_HMIInterface	LKinCtrl_Editor_type...	HMI_Connection_2	PLC_1	Controls.HMIInterface		<symbolic access>	1 s		interface variable to HMI for editor
Controls_HomeAxes	Bool	HMI_Connection_2	PLC_1	Controls.HomeAxes		<symbolic access>	1 s		control variable to home axes with MC_HOME
Controls_ResetAxes	Bool	HMI_Connection_2	PLC_1	Controls.ResetAxes		<symbolic access>	1 s		control variable to reset axes with MC_RESET
Draw_DB_enableEditor	Bool	HMI_Connection_2	PLC_1	Draw_DB.enableEditor		<symbolic access>	1 s		TRUE: Enable functionality of FB
Draw_DB_instLKinCtrl_MovePat.	Bool	HMI_Connection_2	PLC_1	Draw_DB.instLKinCtrl.M...		<symbolic access>	1 s		FB in control of axesGroup / kinematic in motion
Draw_DB_instLKinCtrl_MovePat.	Bool	HMI_Connection_2	PLC_1	Draw_DB.instLKinCtrl.M...		<symbolic access>	1 s		rising edge starts action once
Draw_DB_instLKinCtrl_MovePat.	Bool	HMI_Connection_2	PLC_1	Draw_DB.instLKinCtrl.M...		<symbolic access>	1 s		reset FB / acknowledge errors
Draw_DB_instLKinCtrl_MovePat.	Word	HMI_Connection_2	PLC_1	Draw_DB.instLKinCtrl.M...		<symbolic access>	1 s		status of the FB
Draw_DB_instLKinCtrl_MovePat.	Bool	HMI_Connection_2	PLC_1	Draw_DB.instLKinCtrl.M...		<symbolic access>	1 s		abort path motion
Draw_DB_statSelectedPath	Dint	HMI_Connection_2	PLC_1	Draw_DB.statSelectedPath		<symbolic access>	1 s		static to store selected path
Draw_DB_statSelectedPath_Na.	String	HMI_Connection_2	PLC_1	Draw_DB.statSelectedPat.		<symbolic access>	1 s		
SubNavigationState	Uint	<internal tag>		<Undefined>			1 s		
Tag_ScreenNumber	Uint	<internal tag>		<Undefined>			1 s		

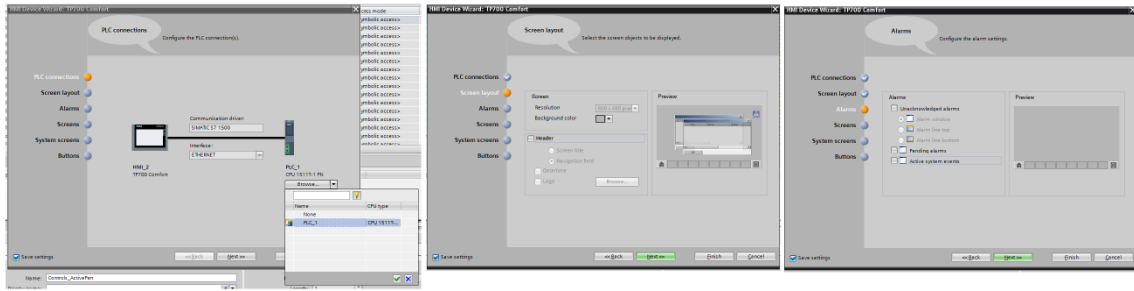
Obr. 2.2.2.4.2 TIA portál - HMI tagy

## 2.2.3 HMI

HMI, s ktorým pracujeme, nám poskytuje možnosť importovať editor z vlozenej knižnice, čo v prípade našej aplikácie je kľúčové, keďže potrebujeme daný editor na definovanie požadovaného pohybu. To znamená, že aj po pripojení k fyzickému PLC budeme mať virtuálny HMI panel na počítači, kde operátor cez panel povolí spustenie programu *Draw*, pohyby jednotlivých osí, zadefinuje pohyb, aktivuje nástroj a spustí pohyb. V prípade potreby ho môže prerušiť a reštartovať proces. Zároveň má k dispozícii základné kontrolky, ktoré signalizujú aktuálny stav, prípadný error alebo dokončenie procesu.

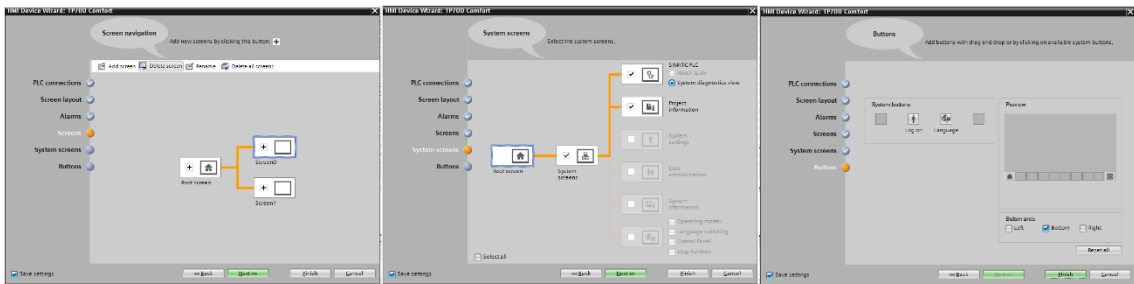
### 2.2.3.1 Obrazovky HMI

HMI vložíme do projektu cez „Add new device“. Po tom, ako vyberieme nami zvolený panel, nám vyskočí HMI Wizard. Je to efektívny nástroj na primárnu konfiguráciu HMI panelu, ktorý zabezpečí prepojenie s PLC a následne vytvorí základné obrazovky podľa našich požiadaviek. Na nasledujúcom obrázku vidíme úvodne 3 okná, v ktorom konfigurujeme HMI. V prvom okne si vyberieme PLC, na ktoré chceme dané PLC pripojiť. Keďže pracujeme len s jedným PLC, zvolíme „PLC\_1“ a cez Ethernetovú sieť nám to prepojí s HMI. V druhom okne máme možnosť zvoliť predvolenú farbu pozadia a môžeme si vybrať, či máme záujem o hlavičku, do ktorej môžeme vložiť logo, prepínač obrazoviek a čas s dátumom. Keďže to nepotrebujeme, tak to necháme neoznačené. V treťom okne nám Wizard ponúkne možnosti alarmov. Prvá možnosť sú alarmy, s ktorými ešte nie sme oboznámení. Tu máme možnosť vybrať, v ktorej časti okna sa nám budú zobrazovať – môžeme ich nechať vo vyskakovacom okne alebo si môžeme vybrať, aby sa nám zobrazovali v hornej časti obrazovky, prípadne v spodnej vo forme aktívneho riadku. Ďalej tu máme možnosť zaznačiť vyskakovacie okno na stále pretrvávajúce alarmy a aktívne operácie systému.



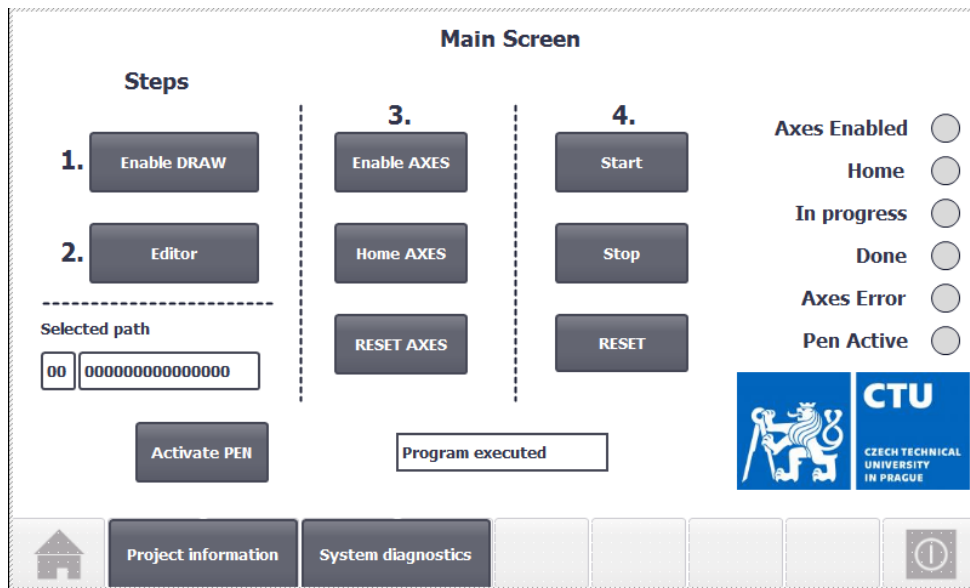
Obr. 2.2.3.1.1 TIA portál - HMI Wizard 1/2

V ďalšom okne nám Wizard ponúkne výber, o aké predvolené obrazovky máme záujem – tzv. systémové okná. V našej aplikácii sme zvolili len diagnostické okno, aby sme mali možnosť aktívne nahliadať na prípadné chyby a zvolili sme okno s informáciami o projekte. Okrem toho je tu možnosť okna so systémovými nastaveniami, systémovými informáciami alebo administráciou užívateľov. V našom prípade sa nejedná o aplikáciu v širšom spektre, kde by k operácii pristupovali viacerí užívatelia, tak to nepotrebujeme. Každopádne administrácia užívateľov je dôležité okno, kde v prípade komplexnejšej aplikácie je určite vhodné rozdeliť práva minimálne pre programátorov programu a operátorov. Posledné okno ponúka základné tlačidlá, ktoré sa zobrazujú na každej obrazovke a máme možnosť výberu, kam tieto tlačidlá chceme umiestniť – na ľavú, spodnú alebo pravú hranu obrazovky alebo na akúkoľvek z týchto kombinácií. Čo sa týka tlačidiel, jedná sa o štyri základné tlačidlá – domovské tlačidlo, vypnutie HMI prostredia, zmena užívateľa alebo zmena jazyka. Na Obr. 2.2.3.1.2 vidíme, ako to vyzerá.



Obr. 2.2.3.1.2 TIA portál - HMI Wizard 2/2

Program začína úvodnou obrazovkou, v ktorej operátor potvrdí vstup a oboznámi sa s autorom, popisom a cieľom práce. Následne sa dostane na hlavnú operačnú obrazovku, kde prebieha celý operačný proces. Tú môžeme vidieť na nasledujúcom Obr. 2.2.3.1.1.

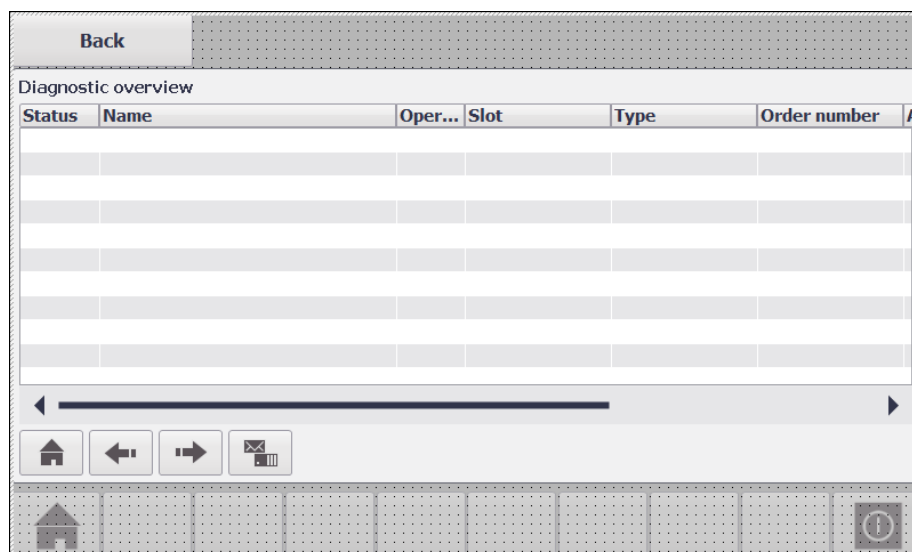


Obr. 2.2.3.1.3 TIA portál - Hlavné okno HMI prostredia

Na tejto obrazovke vidíme hlavnú operačnú časť. Skladá sa z niekoľkých krokov:

1. Povolenie programu kreslenia
2. Vstup do editoru – v tejto časti operátor zadefinuje požadovaný pohyb
3. Povolenie osí, nastavenie osí do *home* pozície a prípadný reset
4. Spustenie programu, prípadné zastavenie a reset

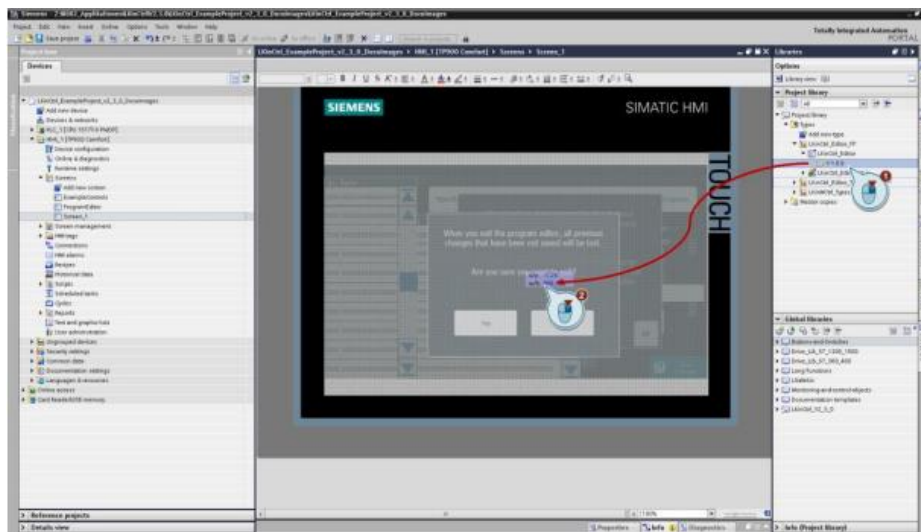
Okrem týchto možností tu máme ešte aktiváciu nástroja – v našom prípade kresliaci nástroj, sledovanie stavu aplikácie (program pripravený/v činnosti/dokončený), zvolený pohyb a tiež máme možnosť sledovať list hlásení z diagnostiky po kliknutí na tlačidlo v spodnej lište – „System diagnostics“. LED indikátory nám hlásia základné aktivity, prípadne stav procesu. Na ďalšom obrázku môžeme nazrieť do diagnostického okna.



Obr. 2.2.3.1.4 TIA portál - Diagnostické okno HMI prostredia

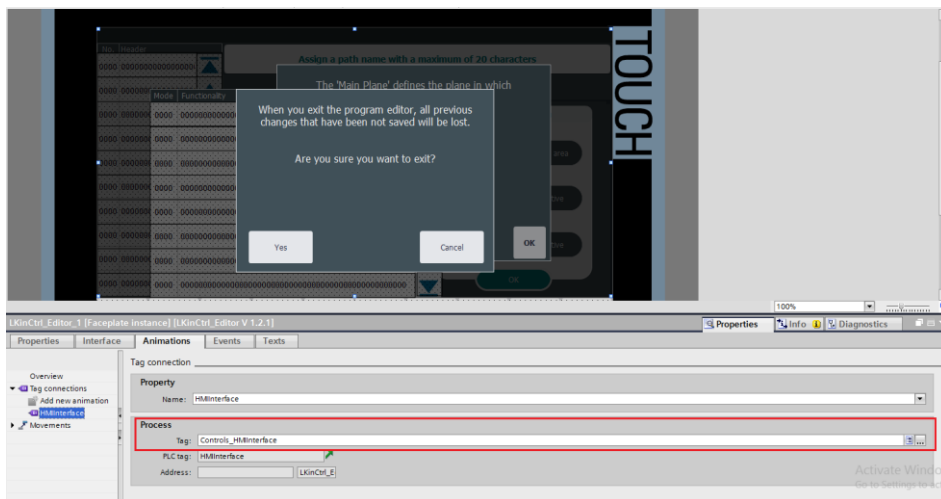


V tomto okne nájdeme zoznam hlásení programu s prípadnými poruchami alebo chybami v procese. Ďalšie okno zobrazuje editor, v ktorom definujeme pohyb. Daný editor je už preddefinovaný od SIEMENSu v rámci importovaných knižníc. Na Obr. 2.2.3.1.5 vidíme ako importujeme daný Editor, resp. faceplate na jednu z obrazoviek, ktorú si vytvoríme. Jedná sa o jednoduchý Drag&Drop. Tento editor ale nie je možné použiť na všetky typy HMI. Práve kvôli tomuto dôvodu sme museli zvoliť virtuálne HMI vyššej triedy, pretože to, ktoré fyzicky máme k dispozícii – KTP400 Basic nepodporuje túto možnosť, keďže sa jedná len o základný panel s veľmi základnými zobrazeniami. Daný editor je vytvorený pre 7 palcové displeje a preto sme vybrali HMI TP700 Comfort, ktorý spĺňa veľkosť aj podporuje danú knižnicu.



Obr. 2.2.3.1.5 TIA portál - Drag&Drop editora z importovanej knižnice [3]

Na to, aby bol editor funkčný, potrebujeme ho prepojiť s funkciami a dátovými blokmi, ktoré sme importovali do PLC. V rámci PLC sme vytvorili v dátovom bloku „Controls“ premennú „HMIInterface“ s preddefinovaným dátovým typom, ktorý sprostredkuje informácie pre editor v HMI. Na Obr. 2.2.3.1.7 vidíme, ako vytvoríme prepojenie. Na hlavnej pracovnej ploche aktivujeme vložený editor a vo vlastnostiach ho napojíme.

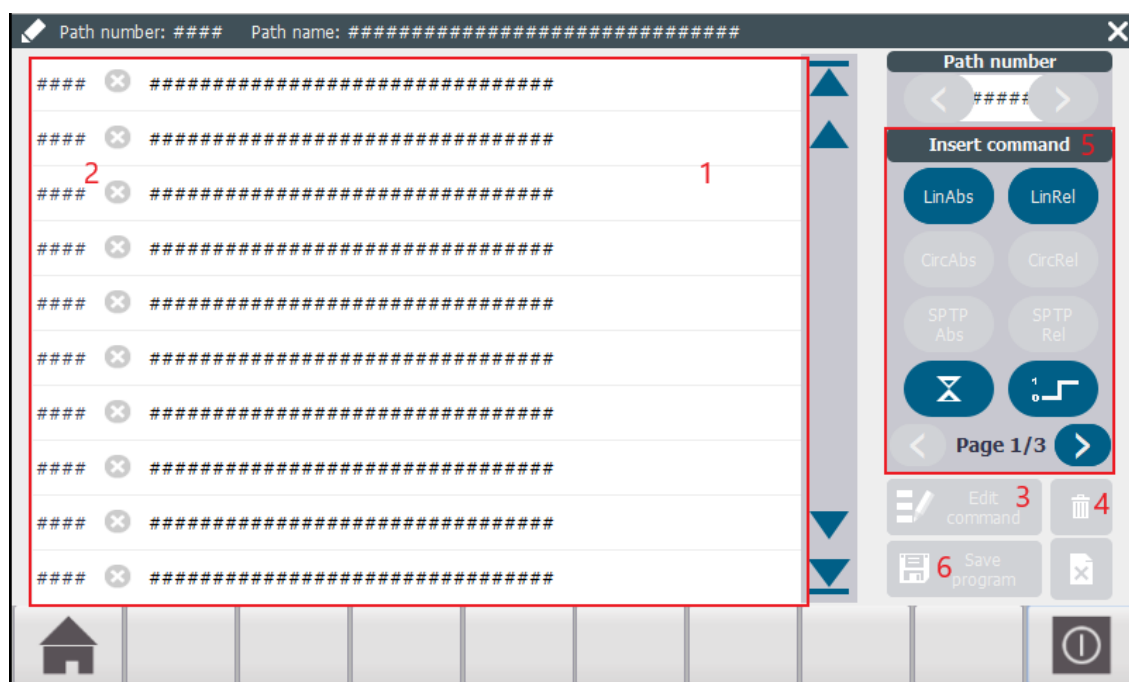


Obr. 2.2.3.1.6 TIA portál - Prepojenie editora s dátovým blokom PLC [3]



Ďalší obrázok ponúka pohľad na editor. Popis jednotlivých bodov v editačnom prostredí:

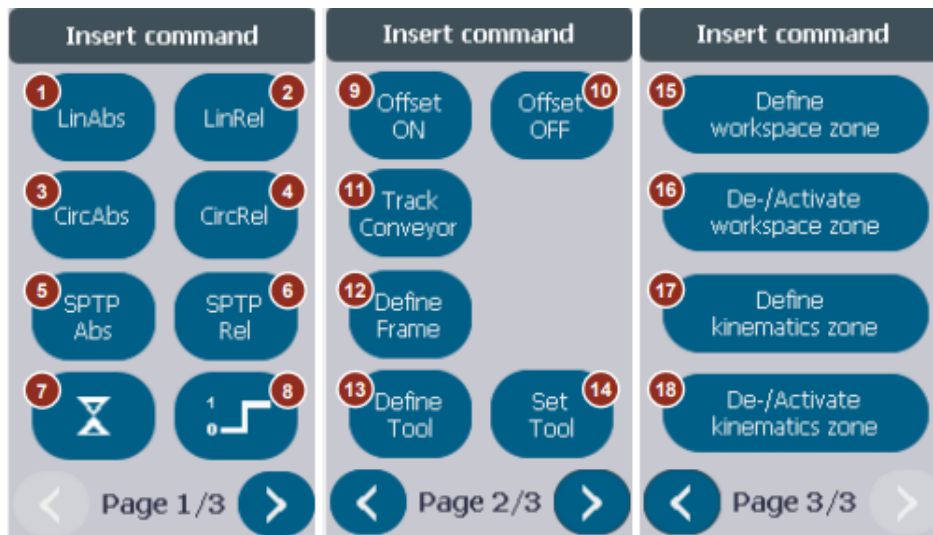
1. Zoznam príkazov
2. Aktivácie a deaktivácie daného príkazu
3. Zmena príkazov
4. Vymazanie príkazu
5. Pridanie nového príkazu
6. Uloženie inštrukcií programu



Obr. 2.2.3.1.7 TIA portál - Okno editora

V rámci bodu 5 – „Insert command“, nájdeme jednotlivé položky v poradí, ktoré definujú lineárny pohyb absolútny, lineárny pohyb relatívny, okružný pohyb absolútny, okružný pohyb lineárny a pod. Možnosti pokynov pokračujú aj na ďalších dvoch stranách, kde napr. nájdeme definovanie pracovných zón. Znak presýpacích hodín znamená „Wait time“ a znak jednotkového signálu znamená „FlagOnly command“. Tento príkaz je aktívny len keď si ho aktivujeme v PathData DB a slúži k aktivácii flagu, ktorý spôsobí akciu, ktorú zadefinujeme.

Po otvorení editora sa nám zobrazí okno ako na Obr. 2.2.3.1.7. Následne ,ak si otvoríme celú ponuku príkazov, tak tam máme možnosti ako na nasledujúcom obrázku.



Obr. 2.2.3.1.8 TIA portál - Príkazy editora [3]

Jednotlivé možnosti podľa čísel:

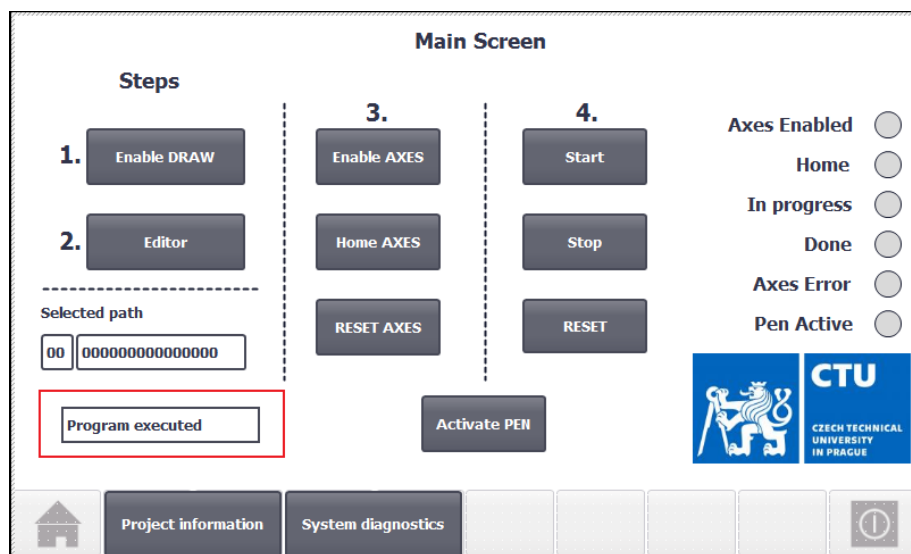
1. Lineárny absolútny pohyb
2. Lineárny relatívny pohyb
3. Okružný absolútny pohyb
4. Okružný relatívny pohyb
5. Priamy absolútny pohyb
6. Priamy relatívny pohyb
7. Čakací čas – umožní počkať za istým príkazom definovaný čas
8. FlagOnly príkazy – v istých momentoch pohybu sa vykoná istý príkaz
9. Povolenie kontúrového offsetu a rádiusovej kompenzácie
10. Vypnutie kontúrového offsetu a rádiusovej kompenzácie
11. Sledovanie dopravného pásu
12. Definovanie OCS alebo užívateľského rámca
13. Definovanie rámca nástroja
14. Nastavenie aktívneho nástroja
15. Definovanie pracovnej zóny
16. De-/Aktivácia pracovnej zóny
17. Definovanie kinematickej zóny
18. De-/Aktivácia kinematickej zóny

## 2.2.3.2 Ovládanie

Celý proces ovládania danej aplikácie spočíva v inicializácii TIA portálu a nahrání programu do PLC. Po úspešnom spustení a pripojení je PLC v online stave. Následne spustíme aj HMI simuláciu, v ktorom budeme ovládať program. K tomu ešte potrebujeme spustiť do online stavu aj oba frekvenčné meniče. Po spustení nám v HMI rozhraní

vyskočí úvodné okno. Po odkliknutí vstupného tlačidla sme presmerovaní na hlavnú plochu. Postup je nasledovný:

1. Povolíme „DRAW“ (v prípade aktivovanej funkcie sa zobrazí na zeleno a zmení názov na „Disable DRAW“)
2. Otvoríme editor a zadefinujeme si požadovaný pohyb
3. Povolíme osi a nastavíme home pozíciu osí
4. V prípade, že chceme hneď kresliť, tak aktivujeme PEN
5. Spustíme program (v tom momente program začne kresliť)
  - a. V prípade potreby zastavíme a zresetujeme príkaz
6. Sledujeme stav v okne v ľavej spodnej časti obrazovky
7. V prípade dokončenia nám ukáže okno „Program executed“



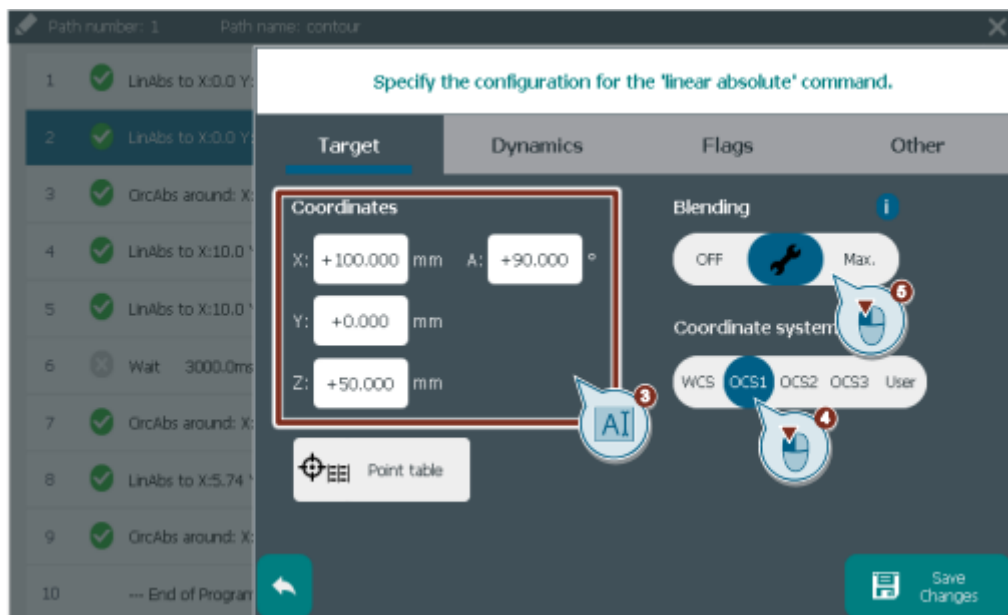
Obr. 2.2.3.2.1 TIA portál - Program vykonaný

Po otvorení editoru sa nám zobrazí okno ako na Obr. 2.2.3.2.2. V tomto okne si zvolíme napr. druhý príkaz a následne vyberieme lineárny absolútny pohyb.



Obr. 2.2.3.2.2 TIA portál - Okno editora [3]

V tomto momente nám vyskočí okno so súradnicami, typom koordinačného systému a tzv. „Blending“. Nastavenie „Blending“ umožňuje zmiešavanie jednotlivých pohybov.



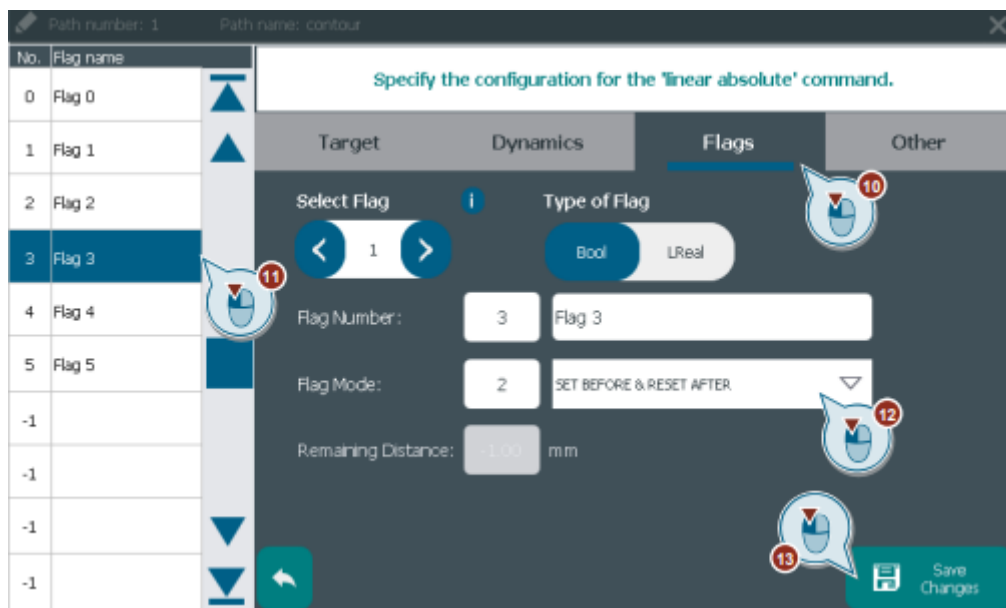
Obr.2.2.3.2.3 TIA portál - Editor - Špecifikácia príkazu [3]

V nastaveniach blendingu nastavíme východziu vzdialenosť na 20 mm a zvolíme rýchly mód. Nastavenia blendingu vyzerajú ako na ďalšom obrázku.



Obr. 2.2.3.2.4 TIA portál - Editor - Blending nastavenia [3]

Ďalej v lište si vyberieme kartu „Dynamics“. V tomto okne v rámci dynamickej adaptácie si zvolíme predvolenú možnosť – aktívnu dynamickú adaptáciu bez segmentácie požadovanej cesty. V okne flagov si nastavíme jednotlivé typy flagov (ak máme záujem) a po zvolení dané možnosti uložíme. Nastavenie flagov vyzerá ako na Obr. 2.2.3.2.5.



Obr. 2.2.3.2.5 TIA portál - Editor - Nastavenie flagov [3]

V poslednom kroku už len v základnom zobrazení editora uložíme program.

## 2.3 TIA portál

V čase písania tejto práce som bol na zahraničnom výmennom pobyte, a tak celá práca bola pripravovaná pomocou simulácií, ktorú ponúka TIA portál vo svojej základnej výbave prídavných programov. Taktiež je tu možnosť simulácie HMI rozhrania, kde vieme vyskúšať celú funkčnosť programu vrátame HMI panelu. Pri implementácii na reálnu aplikáciu to budeme musieť ešte doladiť reálnym podmienkam a to opíšeme v kapitole 2.3.1.2.

### 2.3.1.1 Simulácia

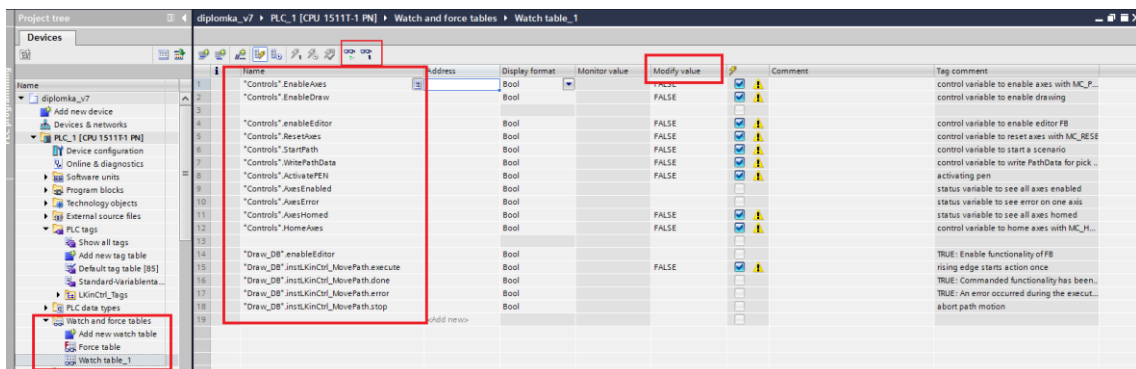
Predtým než budeme chcieť spustiť simuláciu, je potrebné skompilovať program PLC (č. 1 na obrázku) a aj HMI. Ak nemáme žiadne zásadné chyby a skompilujeme to, tak môžeme spustiť simuláciu. Možnosť simulácie PLC nájdeme v hornej lište TIA portálu pod sekciou „Online“ (č. 2 na obrázku), kde následne zvolíme „Simulation – Start“. Po odkliknutí tejto možnosti nám vybehne PLCSIM, ktorý reprezentuje simulované PLC. Spolu s tým sa nám otvorí okno s pripojením k PLC, kde po zvolení správneho „Interface/Subnet“ necháme vyhľadať zariadenia pripojené na danú

komunikačnú sieť. Zvolíme PLC a odklikneme „Load“. Následne zvolíme možnosť spustenia modulu a dokončíme to potvrdením „Finish“. V tomto momente sme nahrali náš projekt do PLC. Teraz nám stačí len zvoliť možnosť „Go online“ (č. 4 na obrázku) a PLC s programom máme plne spustený. Ďalej v lište nájdeme aj spustenie simulácie HMI. To vidíme na obrázku pod číslom 5.



Obr. 2.3.1.1.1 - TIA portál - Ovládacia lišta

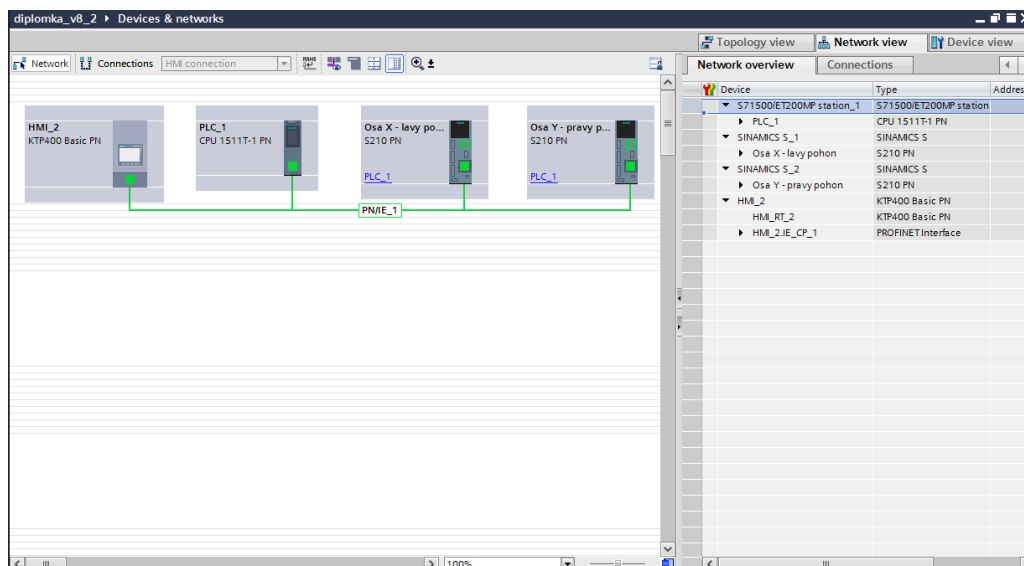
Po spustení simulácie PLC a pripojení online vieme meniť jednotlivé parametre pomocou „Watch“ tabuľky. Tú si môžeme vytvoriť v bočnom menu. Následne si v hlavnom okne nahrajeme parametre, ktoré chceme meniť a sledovať. Ďalej v stĺpci „Modify value“ vieme meniť hodnoty a sledovať ostatné parametre, ako sa menia. Táto možnosť nám poskytne sledovať správanie aplikácie a funkcií a na základe toho vieme ladiť prípadné chyby alebo nežiadúce správanie programu. Na to, aby sme mohli využiť možnosť monitoringu musíme byť online a mať aktivované logo zvýraznené nad sledovanými premennými na obrázku. V tom momente to máme spustené a môžeme ladiť.



Obr. 2.3.1.1.2 TIA portál - Watch table

## 2.3.1.2 Reálna aplikácia

Nakoľko reálna aplikácia obsahuje len 12 MB pamäťovú kartu, náš program nebolo možné stiahnuť, pretože celkový program vyžaduje pamäť cez 13 MB. Z toho dôvodu sme boli nútení vytvoriť alternatívny program na demonštráciu funkčnosti aplikácie. Preto sme zvolili len jednoduchý program, kde je možné vykonať lineárny alebo okružný pohyb s nastavovaním súradníc, oblúku, aktivácie pera a pod. Taktiež na túto základnú aplikáciu sme mohli využiť hardwarovo prítomný HMI panel KTP400 Basic, ktorý dostačuje a nie je preto potrebný virtuálny HMI panel vyššej edície.



Obr. 2.3.1.2.1 TIA portál - Konfigurácia alternatívnej aplikácie

Na Obr. 2.3.1.2.1 vidíme našu aktuálnu konfiguráciu. Jedná sa o PLC, ako z pôvodnej aplikácie, dva frekvenčné meniče S210 a HMI panel KTP400 Basic. IP adresy sme zmenili podľa adries, ktoré boli zadefinované v reálnom HW. Ak by sme chceli meniť IP adresy zariadení, slúži na to program PRONETA od spoločnosti SIEMENS. V našom prípade sa jedná o sieť 192.168.112.x, a konkrétne adresy sú nasledovné:

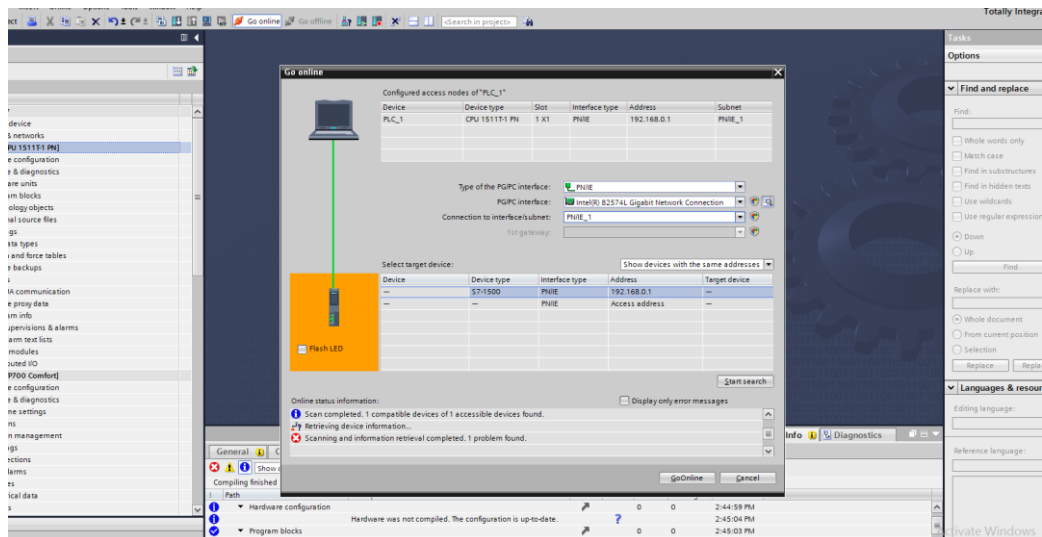
- PLC – 192.168.112.1
- HMI – 192.168.112.2
- S210 – os X – 192.168.112.3
- S210 – os Y - 192.168.112.4

Ďalším dôležitým aspektom, ktorý bolo nutné správne nastaviť v TIA portáli, boli jednotlivé verzie firmwaru jednotlivých častí aplikácie. Na túto časť je nutné dať obzvlášť pozor, pretože ak je rozdiel vo verziách medzi programom a reálnym HW, tak aplikácia a komunikácia medzi jednotlivými aspektami nefunguje. Jednotlivé verzie vidíme v nasledujúcej tabuľke.

Názov	Verzia	Verzia FM
CPU 1511-T PN	15.1.0.0.	2.8
Modul DI 16x24VDC / DQ 16x24VDC/0.5A BA		1.0
HMI KTP400 Basic PN	15.1.0.0.	
FM S210 PN		5.2.3

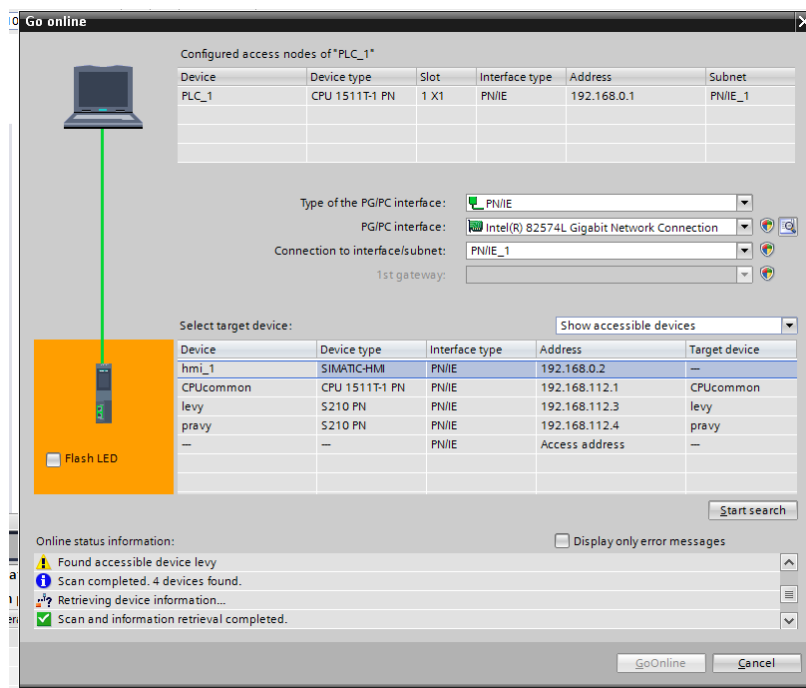
Tab. 2.3.1.2.1 Verzie HW alternatívneho programu

Akonáhle máme zvolené správne verzie aj v našom projekte, môžeme prejsť k pripojeniu. Pripojenie inicializujeme v hornej lište – *GoOnline* a ak máme nastavené rovnaké IP adresy, tak rovno môžeme zvoliť pri výbere cieľových zariadeniach – Zariadenia s rovnakými adresami.



Obr. 2.3.1.2.2 TIA portál - Pripojenie

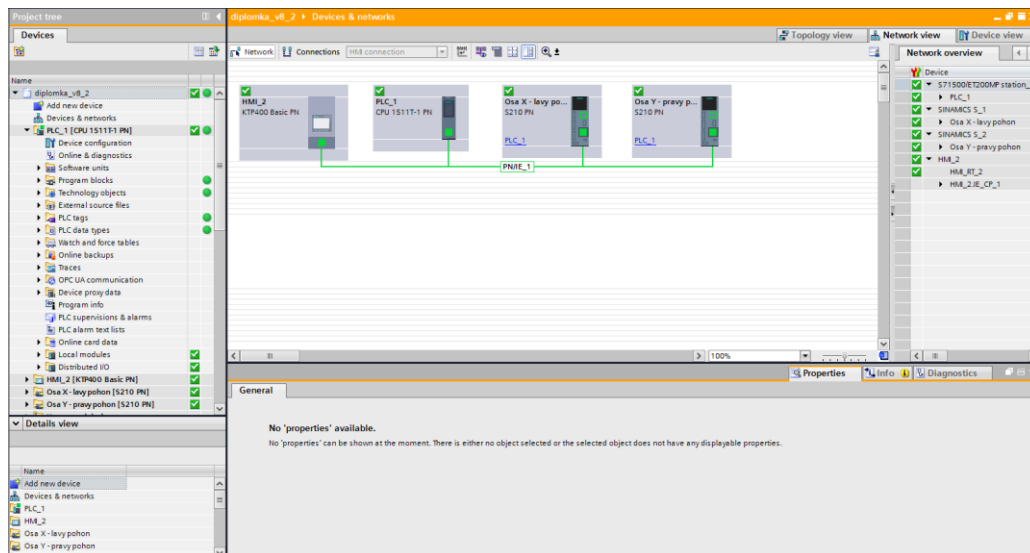
Ak by sa IP adresy líšili, nemusíme ich vyslovene meniť v projekte, stačí zvoliť pri výbere cieľových zariadení všetky dostupné zariadenia a vyjdú nám všetky dostupné zariadenia. To máme aj na Obr. 2.3.1.2.3. Potom zvolíme zariadenie, ku ktorému sa chceme pripojiť a vyskočí nám okno, ktoré ponúkne prideliť rovnakú IP adresu v rámci projektu.



Obr. 2.3.1.2.3 TIA portál - Pripojenie k vybranému zariadeniu

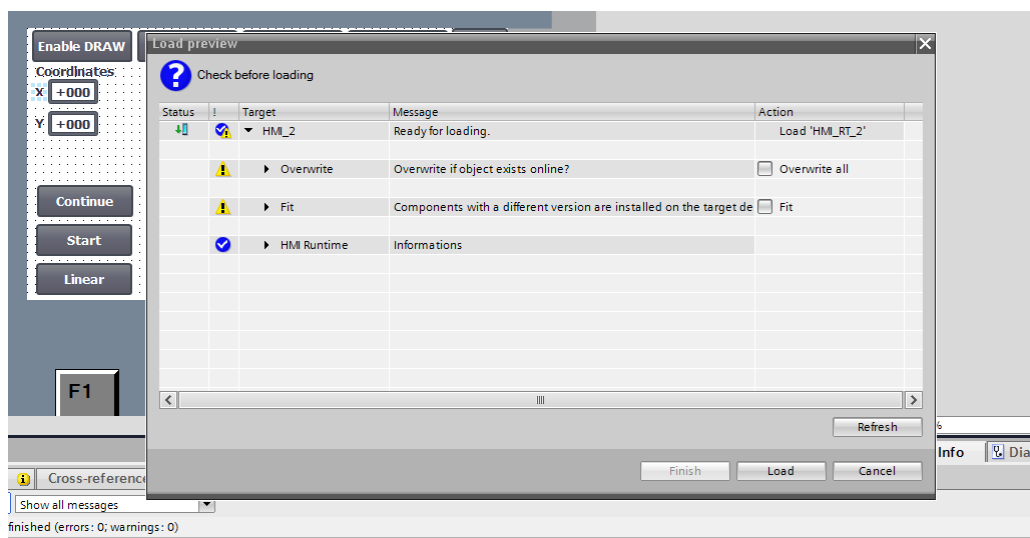
Následne sa len pripojíme a stiahneme program do PLC zariadenia. Rovnakým spôsobom pripojíme aj zvyšné súčasti, až kým všetky súčasti nebudú úspešne pripojené. Po úspešnom spojení a nastavení rovnakých verzií by sme mali byť pripojení bez akýchkoľvek chýb a hlásení, ako to môžeme vidieť na nasledujúcom obrázku.





Obr. 2.3.1.2.4 TIA portál - Úspešné pripojenie

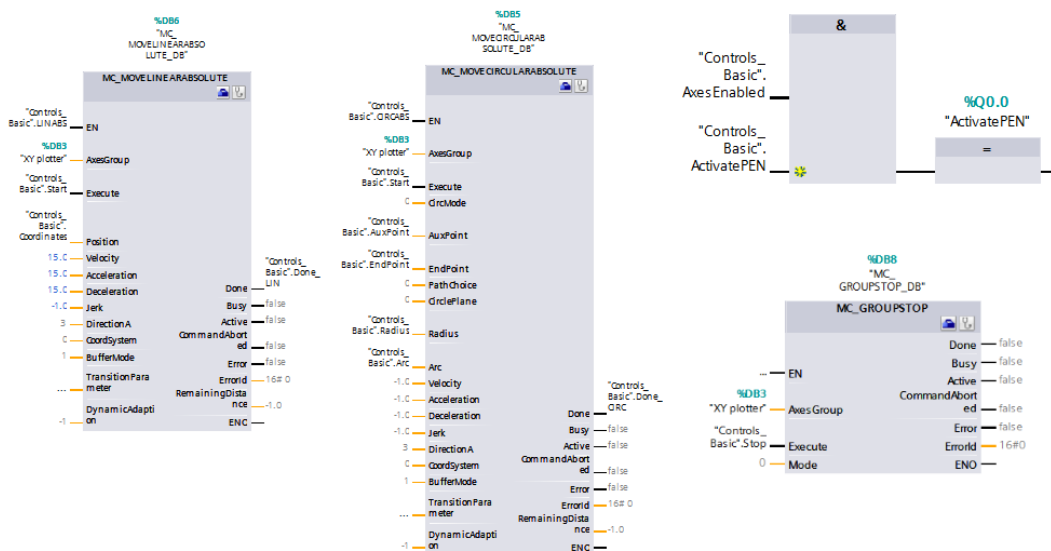
Nakoľko verzia HMI je 15.1.0.0. a celý program sme vytvárali v novšej verzii TIA 16.0.0.0, potrebujeme nastaviť aj HMI verziu v programe na 15.1.0.0., inak by nebolo možné úspešne pripojiť HMI. Následne pri nahrávaní programu do HMI, sa nám naskytne okno ako na ďalšom obrázku. Tu je vhodné nezaškrtnávať nič a nechať to rovno nahráť.



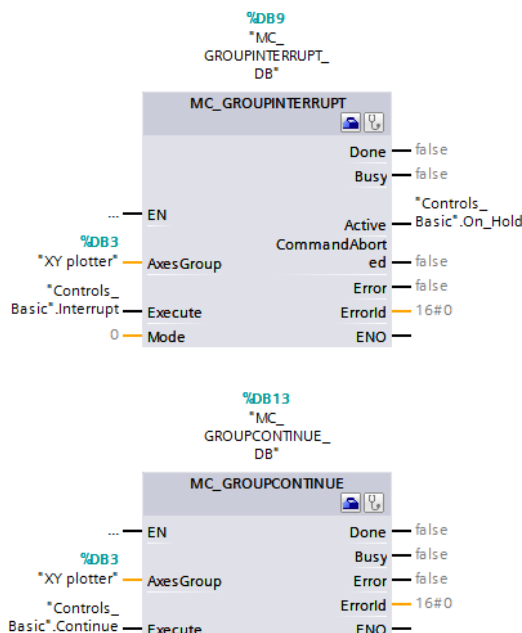
Obr. 2.3.1.2.5 TIA portál - HMI nahrávanie

V programovej časti sme zachovali hlavný OB Main, v ktorom sa osi spúšťajú (MC\_GroupPower), resetujú (MC\_GroupReset) a dávajú do Home pozície (MC\_GroupHome). Vďaka týmto funkciám vieme inicializovať a robiť základné operácie s celou skupinou osí cez Kinematics. Taktiež sme zachovali funkcie na signalizáciu chyby a Home pozície.

Ďalej sme vytvorili OB Basic Draw. Ako aj názov napovedá, jedná sa o jednoduché kreslenie, čisto na demonštráciu základných pohybov po osi. Funkciu môžeme vidieť na ďalších obrázkoch.



Obr. 2.3.1.2.6 TIA portál - Basic Draw 1/2



Obr. 2.3.1.2.7 TIA portál - Basic Draw 2/2

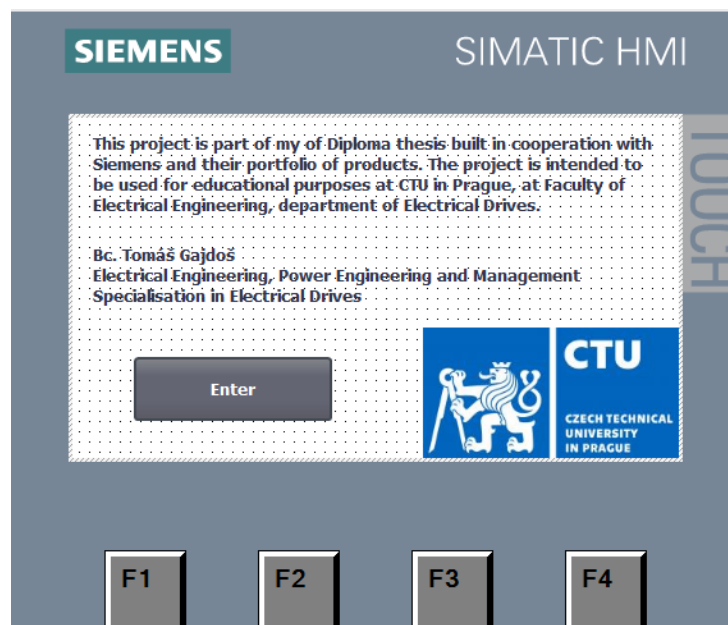
Na Obr. 2.3.1.2.6 a Obr. 2.3.1.2.8 vidíme funkcie v rámci Basic Draw. Jedná sa o preddefinované funkcie, ktoré sme využili z knižnice Kinematics na základne pohyby. Jedná sa o funkcie lineárny absolútny pohyb a okružný absolútny pohyb, kde sme len pripojili TO\_Kinematics „XY plotter“ a následne premenné, ktoré aktivujú dané funkcie. Ďalej sme k tomu pripojili premenné, ktoré odštartujú danú funkciu s danými koordinátami, ktoré zadefinujeme. V prípade okružného pohybu tu máme ďalšie hodnoty ako Aux point, End point a Arc, ktoré potrebujeme zadefinovať. Taktiež tu máme funkciu „Activate PEN“, ktoré slúži k priklopeniu nástroja, v našom prípade pera. Na záver sme zadefinovali funkciu STOP, Interrupt (prerušenie) a Continue (pokračuj). Tieto funkcie sme vytvorili preto, ak by sme chceli proces zastaviť, prerušiť a následne v ňom ďalej pokračovať.

V pôvodnej aplikácii sme mali vytvorený dátový blok „Controls“. Pre potreby tejto aplikácie sme vytvorili nový „Controls\_Basic“, výhradne pre účely tejto aplikácie. V základe sa tieto dátové bloky podobajú, len sme aktuálny dátový blok obohatili o ďalšie potrebné premenné, ktoré potrebujeme. Daný dátový blok vidíme na ďalšom obrázku.

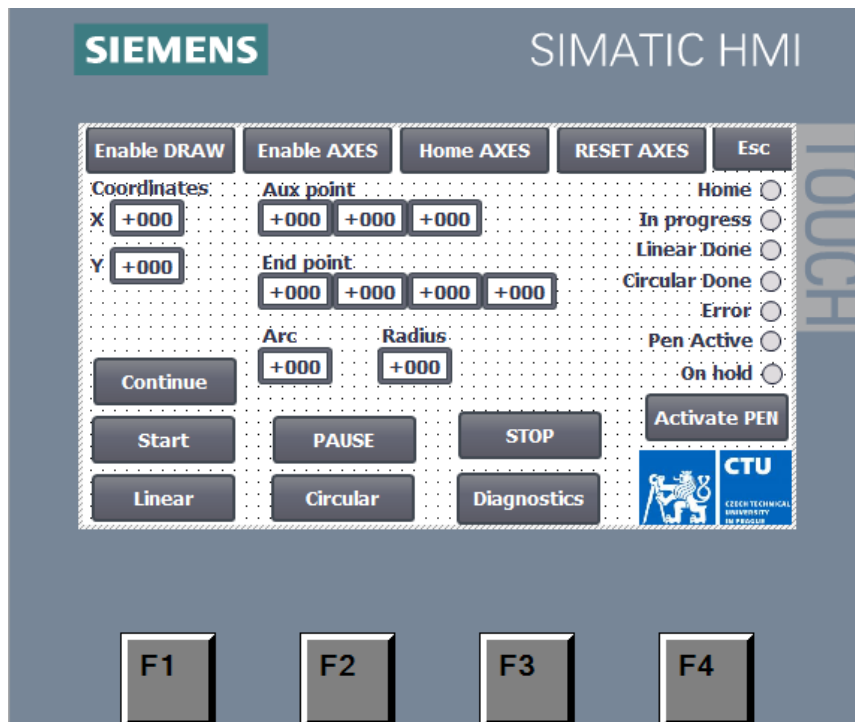
EnableDraw	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	control variable to enable drawing
EnableAxes	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	control variable to enable axes with MC_POWER
AxesEnabled	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	status variable to see all axes enabled
ResetAxes	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	control variable to reset axes with MC_RESET
AxesError	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	status variable to see error on one axis
HomeAxes	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	control variable to home axes with MC_HOME
AxesHommed	Bool	false		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	status variable to see all axes hommed
HomePos	Array[1..4] of LReal			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable storing homing positions for ...
HomeMode	Array[1..4] of Int			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable storing homing modes for M...
RestartMode	Array[1..5] of Bool			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable storing restart modes for MC...
StartMode	Array[1..4] of DInt			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable storing start modes for MC_G...
StopMode	Array[1..4] of Int			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable storing stop modes for MC_G...
WritePathData	Bool	false		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to write PathData for pick and...
AutoAcknowledge	Bool	TRUE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to auto acknowledge wait fla...
StartPath	Bool	false		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	control variable to start a scenario
P_TRIG_flag_0	Bool	false		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	storage variable for positive trigger flag 0
P_TRIG_flag_1	Bool	false		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	storage variable for positive trigger flag 1
enableEditor	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	control variable to enable editor FB
HMIInterface	*LKinCtrl_Editor_by...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	interface variable to HMI for editor
ActivatePEN	Bool	false		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	activating pen
Coordinates	Array[1..4] of LReal			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Radius	LReal	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Arc	LReal	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
AuxPoint	Array[1..3] of LReal			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
EndPoint	Array[1..4] of LReal			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Start	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
LINABS	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
CIRCABS	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Done_CIRC	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Done_LIN	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Active_command	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Stop	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Interrupt	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Continue	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
On_Hold	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Obr. 2.3.1.2.8 TIA portál - Dátový blok Controls\_Basic

Posledným krokom bolo vytvorenie nového HMI rozhrania. Vytvorili sme dve základné obrazovky, kde prvá je úvodná, v ktorej približujeme účel aplikácie a autora a druhá, hlavná, ktorá slúži k ovládaniu programu. Okrem toho sme využili aj preddefinované okno diagnostiky, v ktorej je možné zobrazit' chyby. Preddefinované okno sme vložili pri inicializácii HMI cez Wizarda, ktorý sme už opísali v predošlých kapitolách.



Obr. 2.3.1.2.9 TIA portál - Úvodná obrazovka



Obr. 2.3.1.2.10 TIA portál - Hlavná obrazovka

V hlavnej obrazovke máme okrem funkcií tlačidiel ovládania aj signálové kontrolky, ktoré indikujú v akom štádiu sa proces nachádza, či už je príkaz vykonaný, či nastal nejaký error a podobne. Taktiež napr. bez aktivácie „Circular“ sa nám nezobrazia možnosti „Aux point“, „End point“, „Arc“ alebo „Radius“. Podobne funguje aj tlačidlo „Continue“, ktoré sa zobrazí až v momente aktivácie „Pause“. Obdobne funguje aj zobrazenie „Coordinates“ v prípade lineárneho pohybu. Je to nastavené z dôvodu prehľadnosti pre operátora.

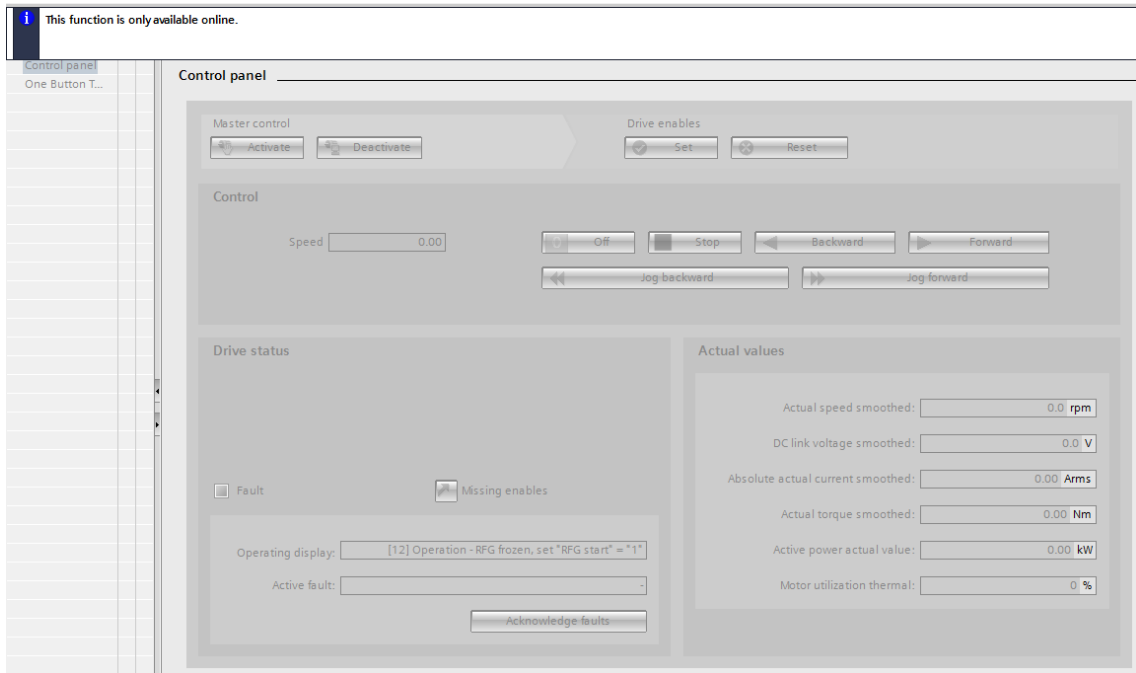
Na to, aby nám HMI mohlo správne fungovať, potrebujeme vytvoriť ešte HMI tagy, ktorými popárujeme jednotlivé tlačidlá s funkciami. Vytvorené HMI tagy máme na ďalšej snímke.

Symbol	Data Type	Connection	PLC	Symbol	Data Type	Symbol
Control_Active_command	Bool	HMI_Conne...	PLC_1	Controls_BasicActive_command	<symbol...>	1 s
Controls_ActivePen	Bool	HMI_Connectio...	PLC_1	Controls_BasicActivatePEN	<symbol...>	1 s
Controls_ActivePen	Bool	HMI_Connectio...	PLC_1	Controls_BasicActivatePEN	<symbol...>	activating pen
Controls_Arc	LReal	HMI_Connectio...	PLC_1	Controls_BasicArc	<symbol...>	1 s
Controls_Aux_Point_1	LReal	HMI_Connectio...	PLC_1	Controls_BasicAuxPoint[1]	<symbol...>	1 s
Controls_Aux_Point_2	LReal	HMI_Connectio...	PLC_1	Controls_BasicAuxPoint[2]	<symbol...>	1 s
Controls_Aux_Point_3	LReal	HMI_Connectio...	PLC_1	Controls_BasicAuxPoint[3]	<symbol...>	1 s
Controls_AxesEnabled	Bool	HMI_Connectio...	PLC_1	Controls_BasicAxesEnabled	<symbol...>	1 s
Controls_AxesError	Bool	HMI_Connectio...	PLC_1	Controls_BasicAxesError	<symbol...>	1 s
Controls_AxesHomed	Bool	HMI_Connectio...	PLC_1	Controls_BasicAxesHomed	<symbol...>	1 s
Controls_CIRCABS	Bool	HMI_Connectio...	PLC_1	Controls_BasicCIRCABS	<symbol...>	1 s
Controls_Continue	Bool	HMI_Connectio...	PLC_1	Controls_BasicContinue	<symbol...>	1 s
Controls_Coordinate_1	LReal	HMI_Connectio...	PLC_1	Controls_BasicCoordinates[1]	<symbol...>	1 s
Controls_Coordinate_2	LReal	HMI_Connectio...	PLC_1	Controls_BasicCoordinates[2]	<symbol...>	1 s
Controls_Coordinate_3	LReal	HMI_Connectio...	PLC_1	Controls_BasicCoordinates[3]	<symbol...>	1 s
Controls_Coordinate_4	LReal	HMI_Connectio...	PLC_1	Controls_BasicCoordinates[4]	<symbol...>	1 s
Controls_Done_CIRC	Bool	HMI_Connectio...	PLC_1	Controls_BasicDone_CIRC	<symbol...>	1 s
Controls_Done_LIN	Bool	HMI_Connectio...	PLC_1	Controls_BasicDone_LIN	<symbol...>	1 s
Controls_EnableAxes	Bool	HMI_Connectio...	PLC_1	Controls_BasicEnableAxes	<symbol...>	1 s
Controls_EnableDraw	Bool	HMI_Connectio...	PLC_1	Controls_BasicEnableDraw	<symbol...>	1 s
Controls_EnableDraw	Bool	HMI_Connectio...	PLC_1	Controls_BasicEnableDraw	<symbol...>	control variable to enable axes with MC_POWER
Controls_EnableEditor	Bool	HMI_Connectio...	PLC_1	Controls_BasicEnableEditor	<symbol...>	1 s
Controls_EnableEditor	Bool	HMI_Connectio...	PLC_1	Controls_BasicEnableEditor	<symbol...>	control variable to enable editor FB
Controls_End_Point_1	LReal	HMI_Connectio...	PLC_1	Controls_BasicEndPoint[1]	<symbol...>	1 s
Controls_End_Point_2	LReal	HMI_Connectio...	PLC_1	Controls_BasicEndPoint[2]	<symbol...>	1 s
Controls_End_Point_3	LReal	HMI_Connectio...	PLC_1	Controls_BasicEndPoint[3]	<symbol...>	1 s
Controls_End_Point_4	LReal	HMI_Connectio...	PLC_1	Controls_BasicEndPoint[4]	<symbol...>	1 s
Controls_HomeAxes	Bool	HMI_Connectio...	PLC_1	Controls_BasicHomeAxes	<symbol...>	1 s
Controls_HomeAxes	Bool	HMI_Connectio...	PLC_1	Controls_BasicHomeAxes	<symbol...>	control variable to home axes with MC_HOME
Controls_Interrupt	Bool	HMI_Connectio...	PLC_1	Controls_BasicInterrupt	<symbol...>	1 s
Controls_LINABS	Bool	HMI_Connectio...	PLC_1	Controls_BasicLINABS	<symbol...>	1 s
Controls_OnHold	Bool	HMI_Connectio...	PLC_1	Controls_BasicOn_Hold	<symbol...>	1 s
Controls_Radius	LReal	HMI_Connectio...	PLC_1	Controls_BasicRadius	<symbol...>	1 s
Controls_ResetAxes	Bool	HMI_Connectio...	PLC_1	Controls_BasicResetAxes	<symbol...>	1 s
Controls_ResetAxes	Bool	HMI_Connectio...	PLC_1	Controls_BasicResetAxes	<symbol...>	control variable to reset axes with MC_RESET
Controls_Start	Bool	HMI_Connectio...	PLC_1	Controls_BasicStart	<symbol...>	1 s
Controls_STOP	Bool	HMI_Connectio...	PLC_1	Controls_BasicStop	<symbol...>	1 s
SubNavigationState	UInt	<internal tag>		<Undefined>	<symbol...>	1 s

Obr. 2.3.1.2.11 TIA portál - HMI tagy

Taktiež oproti pôvodnému testovaniu aplikácie cez simulácie nesmieme zabudnúť pri prechode na reálny HW vypnúť v prvom okne konfigurácie technologických objektoch simuláciu, inak dané TO nebudú priamo prepojené na fyzické FM S210.

Užitočnou funkciou pri doladovaní aplikácie je Commissioning v rámci frekvenčných meničov. Týmto spôsobom je možné doladiť HW nastavenia jednotlivých pohonov a vyskúšať funkčnosť. Táto možnosť je dostupná len v online stave.

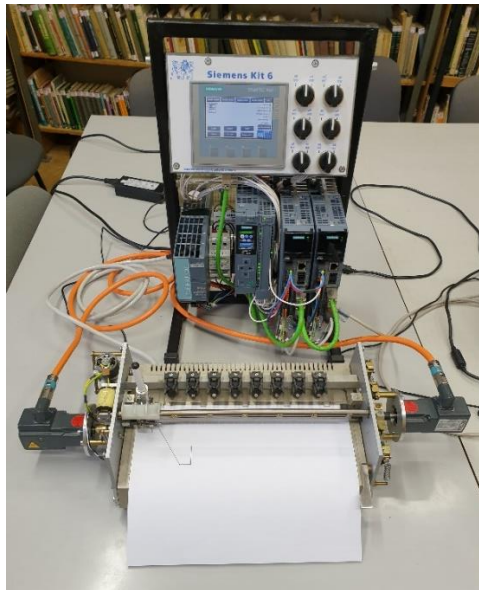


Obr. 2.3.1.2.12 TIA portál - Commissioning

Samotné ovládanie programu spočíva v nahratí programu do fyzického HW. Po úspešnom nahratí ovládame program už len cez HMI panel. Nasledovný postup:

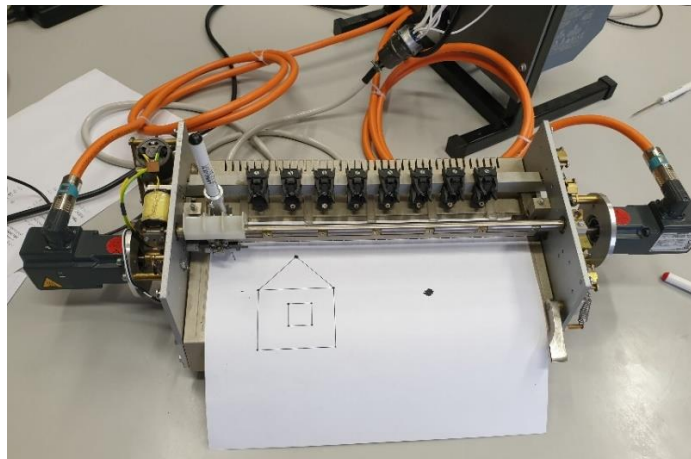
1. Potvrdíme vstup cez úvodnú obrazovku
2. Povolíme Draw
3. Povolíme osi
4. Povolíme lineárny alebo okružný pohyb
5. Zadáme buď súradnice požadovaného pohybu v prípade lineárneho pohybu alebo zadáme zvyšné potrebné údaje pre okružný pohyb
6. Aktivujeme nástroj (v našom prípade pero)
7. Odklikneme štart a program začne vykresľovať
8. Následne po dokončení deaktivujeme pero a zadáme nové súradnice
9. Znova aktivujeme pero a pokračujeme kým nevykreslíme, čo potrebujeme

Na snímke vidíme, ako naša aplikácia vyzerá.



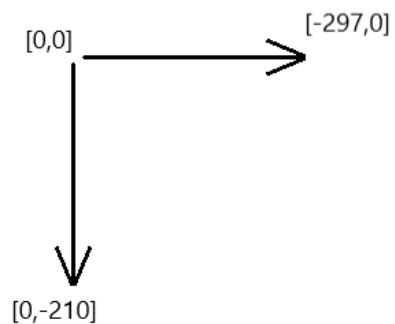
Obr. 2.3.1.2.13 Reálna aplikácia

Na ďalšej snímke máme vyobrazené vykreslenie obrázka.



Obr. 2.3.1.2.14 Vykreslený obraz

V rámci vykresľovania systému súradníc zadávame súradnice v rámci rozmerov papiera s negatívnym znamienkom.



Obr. 2.3.1.2.15 Súradnice vykresľovania

## Záver

Úvodná časť práce bola venovaná popisu samotného projektu a jeho účelu. Ďalej v teoretickej časti rozoberáme možnosti TIA portálu, ako funguje komunikácia medzi jednotlivými súčastami a pomocou čoho si dané súčasti vymieňajú informácie. Taktiež opisujeme technologický objekt Kinematics a jeho možnosti programovania, nastavení a limitov a možnosti ladenia. Okrem toho sme priblížili aj importované knižnice a funkcie v nich. V praktickej časti sme sa už venovali konkrétnym konfiguráciám použitých komponentov a ich verziám, kde sme začali s popisom PLC a programových súčastí a tagov. Ďalej v nej máme popísané nastavenia jednotlivých technologických objektov, frekvenčných meničov s príslušnými motormi a enkódermi. Taktiež popisujeme vytvorenie ovládacieho panela, vytvorenie jednotlivých obrazoviek a ich prepojenie cez HMI tagy na PLC tagy, pomocou ktorých ovládame aplikáciu. Ďalej v práci popisujeme jednotlivé organizačné, funkčné, dátové bloky a samotné funkcie. V rámci HMI popisujeme aj import funkcií vytvorených spoločnosťou SIEMENS.

Nakoľko celá práca bola vytváraná počas môjho zahraničného pobytu a celú funkčnosť sme skúšali len cez simulácie, tak pri implementácii sme narazili na problém s pamäťou, pretože samotné PLC spoločne s rozširujúcou pamäťovou kartou o veľkosti 12 MB neposkytlo dostatočnú pamäť na nahratie nášho programu. Z tohto dôvodu sme vytvorili alternatívnu aplikáciu na demonštráciu funkčnosti, kde sme použili pôvodný hlavný organizačný blok a vytvorili sme k tomu nový organizačný blok, ktorý zabezpečuje ovládanie pohybu spolu s prídavnými funkciami. Táto časť je popísaná v poslednej kapitole práce.

## **Zoznam použitej literatúry**

- [1] Pavelka J., Koblík P. Elektrické pohony a jejich řízení. 3. přepracované vydání. Praha: České vysoké učení technické v Praze, 2016. ISBN 978-80-01-06007-0
- [2] Weidauer J., Messer R. Electrical Drives, Publics Erlangen, 2014
- [3] Library Kinematics Control [online]. Erlangen, Německo: Siemens AG Digital Industries Division, V3.1, 08/2021. Dostupné na: [https://support.industry.siemens.com/cs/attachments/109755891/109755891\\_LKinCtrl\\_Doc\\_v3\\_1\\_3\\_en.pdf](https://support.industry.siemens.com/cs/attachments/109755891/109755891_LKinCtrl_Doc_v3_1_3_en.pdf)
- [4] SCE Training Curriculum: Automation System SIMATIC S7-1500 [online]. Nuremberg, Německo: SIEMENS AG Digital Factory Division, 04/2016. Dostupné z: <https://www.automation.siemens.com/sce-static/learning-training-documents/tiaportal/summary-sce-training-curriculum-s7-1500-en.pdf>
- [5] NOVOTNÝ, Radek. SIMATIC S7-1500 T-CPU: Technologické objekty [online]. Praha: SIEMENS AG, 2019 Dostupné na: <https://www.tianadosah.cz/upload/01-technologicke-objekty-vsismatic-s7-1500-t-cpu.pdf>
- [6] PLACHÝ, Ladislav. SIMATIC HMI: Nástroje pro všechny vizualizační úlohy [online]. SIEMENS AG, 2018. Dostupné na: <https://moodle.fel.cvut.cz/mod/resource/view.php?id=174275>
- [7] MARČAN, Peter. Daily Automation. 06 PROFINET: Isochronous Real-Time (IRT) komunikácia [online]. Dostupné na: <https://www.dailyautomation.sk/irt/>
- [8] MARČAN, Peter. Daily Automation. 02 Komunikačné kanály – PROFINET RT-Real-Time, NRT, IRT [online]. Dostupné na: <https://www.dailyautomation.sk/profinet-real-time-rt-nrt-irt/>
- [9] Totally Integrated Automation Portal. V16.0 Software. SIEMENS AG, Nápoveda v programe.
- [10] WESTLAKE, Robert. PROFINET – RT vs IRT [online]. SIEMENS AG, 2020. Dostupné na: <https://assets.new.siemens.com/siemens/assets/api/uuid:30e0ba94-b1af-4488-98dd-8fe716382cc0/siemens-profinet-rt-vs-irt-webinar-13oct2020.pdf>
- [11] Configuration of technology objects with SIMATIC S7-1500 and SINAMICS S210 in the TIA Portal [online]. SIEMENS AG, 2019. Dostupné na: <https://support.industry.siemens.com/cs/document/109749795/configuring-technology-objects-with-simatic-s7-1500-and-sinamics-s210-in-tia-portal?dti=0&pnid=13204&lc=en-WW>
- [12] SINAMICS S210 Servo Drive System: Catalog D 32 [online]. SIEMENS AG, 2020 [cit. 2021-02-06]. Dostupné na: <https://support.industry.siemens.com/cs/document/109754381/catalog-d-32%3Asinamics-s210-servo-drive-system?dti=0&pnid=13204&lc=en-WW>



- [13] SINAMICS S210 / SIMOTICS S-1FK2: Operating Instructions [online]. SIEMENS AG, 2020. Dostupné na:  
<https://support.industry.siemens.com/cs/document/109781874/sinamics-s210-simotics-s-1fk2?dti=0&pnid=13204&lc=en-WW>
- [14] Industry Mall: 6ES7511-1TK01-0AB0 [online]. SIEMENS AG, 2020. Dostupné na:  
<https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7511-1TK01-0AB0>
- [15] Industry Mall: 6ES7523-1BL00-0AA0 [online]. SIEMENS AG, 2020. Dostupné na:  
<https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7523-1BL00-0AA0>
- [16] Industry Mall: 6SL3210-5HB10-1UF0 [online]. SIEMENS AG, 2020. Dostupné na:  
<https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6SL3210-5HB10-1UF0>
- [17] Industry Mall: 1FK2102-0AG00-1MA0 [online]. SIEMENS AG, 2020. Dostupné na:  
<https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/1FK2102-0AG00-1MA0>
- [18] Industry Mall: 6AV2123-2DB03-0AX0 [online]. SIEMENS AG, 2020. Dostupné na:  
<https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6AV2123-2DB03-0AX0>
- [19] Industry Mall: 6GK5816-1BA00-2AA2 [online]. SIEMENS AG, 2020. Dostupné na:  
<https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6GK5816-1BA00-2AA2>
- [20] Industry Mall: 6AV2124-0GC01-0AX0 [online]. SIEMENS AG, 2020. Dostupné na:  
<https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6AV2124-0GC01-0AX0>