



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Univerzální laboratorní PLC úloha se zaměřením na řízení pohybu

## Bakalářská práce

*Studijní program:*

*Autor práce:*

*Vedoucí práce:*

B0714A270001 Mechatronika

**Nazarij Petryšak**

Ing. Martin Diblík, Ph.D.

Ústav mechatroniky a technické informatiky





## Zadání bakalářské práce

# Univerzální laboratorní PLC úloha se zaměřením na řízení pohybu

*Jméno a příjmení:* **Nazarij Petryšak**  
*Osobní číslo:* M19000138  
*Studijní program:* B0714A270001 Mechatronika  
*Zadávací katedra:* Ústav mechatroniky a technické informatiky  
*Akademický rok:* 2021/2022

### Zásady pro vypracování:

1. Prostudujte koncepci laboratorní úlohy s PLC technikou BR-Automation, seznamte se s mechanickým uspořádáním, elektroinstalací a jednotlivými automatizačními prvky.
2. Na základě zjištěných informací realizujte dvě identické úlohy. Sestavte mechanický rám, osadte automatizační komponenty, realizujte jejich elektrické propojení.
3. Ve vývojovém prostředí BR Automation Studio navrhnete a realizujete program pro PLC automat. Software by měl demonstrovat základní funkcionality použitých automatizačních prvků.
4. V programu se zaměřte na názorné ukázky použití servopohonů pro základní a pokročilé způsoby řízení.

*Rozsah grafických prací:*  
*Rozsah pracovní zprávy:*  
*Forma zpracování práce:*  
*Jazyk práce:*

dle potřeby dokumentace  
30–40 stran  
tištěná/elektronická  
Čeština



### **Seznam odborné literatury:**

- [1] JOHN, Karl-Heinz a Michael TIEGELKAMP. IEC 61131-3: programming industrial automation systems: concepts and programming languages, requirements for programming systems, decision-making aids. Second edition. Berlin : New York: Springer, 2010. ISBN 978-3-642-12014-5
- [2] BR-AUTOMATION. Training materials.
- [3] MARTINÁSKOVÁ, Marie, Ladislav ŠMEJKAL, ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE a STROJNÍ FAKULTA. Řízení programovatelnými automaty III: softwarové vybavení. Praha: Vydavatelství ČVUT, 2003. ISBN 978-80-01-02804-9.

*Vedoucí práce:*

Ing. Martin Diblík, Ph.D.  
Ústav mechatroniky a technické informatiky

*Datum zadání práce:*

12. října 2021

*Předpokládaný termín odevzdání:*

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

L.S.

doc. Ing. Josef Černožorský, Ph.D.  
vedoucí ústavu

V Liberci dne 12. října 2021

## Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

16. května 2022

Nazarij Petryšak



## **Poděkování**

Nejdříve bych rád poděkoval vedoucímu Ing. Martinu Diblíkovi Ph.D za odborné vedení mé bakalářské práce, velkou míru trpělivosti a ochoty, rychlosti, lidský přístup a také za cenné a velmi podnětné rady při zpracovávání.

Také bych rád poděkoval své rodině a všem přátelům, kteří mě při vytváření této práce podpořili, a bez jejichž pomoci by nebylo možné práci dokončit.

## **Abstrakt**

Bakalářská práce se zabývá sestavením dvěma totožnými zařízeními s použitím techniky od firmy B&R Industrial Automation. Cílem bylo osadit komponenty do mechanického rámu a vše elektricky propojit. Tato práce demonstruje řízení pohybu dvou identických synchronních servomotorů, které se ovládají dotykovým panelem. Pro programování pohybu byl využit software Automation Studio, ve kterém jsou používány knihovny ACP10 pro řízení dvou os pomocí funkčních bloků standardizované celosvětově známou organizací PLCopen, která podporuje všechny programovací jazyky IEC 61131-1. Řízení synchronního pohybu využívá metody elektronické vačky a elektronické převodovky. Výsledkem práce bylo sjednocení dvou metod pro synchronní řízení. Pro každou osu byl vytvořen program v jazyce ST, který se zabývá řízením os nezávisle na sobě, a také obsahuje pohyb dvou os pomocí synchronní vazby typu Gear a Cam. Pro možnost ovládání os bylo vytvořeno uživatelské rozhraní pro HMI. Uživatel má možnost pomocí tlačítek řídit osy a volit si, jaká z os bude řízená (SLAVE). Díky vstupním blokům lze nastavit parametry (rychlost, zrychlení atd.) pro každou osu zvlášť a přidáním různých kontrolních statusů lze informovat uživatele o stavu motorů a řídicí jednotky.

## **Klíčová slova:**

PLCopen, synchronizovaný pohyb, Automation studio, HMI, B&R, servomotory

## **Abstract**

This bachelor thesis deals with assembling two identical devices using components from B&R Industrial Automation company. The goal was to put the components into a mechanical frame and connect everything electrically. This work demonstrates control of the movement of two identical synchronous servo motors, which are controlled by a touch display. Software Automation Studio has been used for the programming movement in which ACP10 libraries are used to control two axes by means of functional block standardized by the globally known PLCopen organization, which supports all the programming languages IEC 61131-1. The control of synchronous movement uses electronic cam and electronic transmission methods. The result of the work was to unify two methods for synchronous management. For each axis, a ST program was created that deals with the control of axes independently of each other, and also includes the movement of two axes using synchronous binding of the Gear and Cam type. The user interface for HMI was created for the option to control the axes. The user has the option to use the buttons to control the axes and choose what of the axes will be controlled (SLAVE). With the input blocks, the parameters (speed, acceleration, etc.) can be set for each axis separately and the different control status can be used to inform the user about the state of the engines and the control unit.

## **Key words:**

PLCopen, synchronized movement, Automation studio, HMI, B&R, servomotors

# Obsah

<b>1</b>	<b>Úvod</b> .....	<b>11</b>
<b>2</b>	<b>Řízení pohybu</b> .....	<b>12</b>
2.1	Stavový diagram.....	12
2.1.1	Popis stavového diagramu .....	13
<b>3</b>	<b>PLCopen</b> .....	<b>14</b>
3.1	Norma IEC 61131-3 .....	14
3.1.1	Programovací jazyky .....	14
<b>4</b>	<b>Hardwarové uspořádání</b> .....	<b>17</b>
4.1	Komunikační rozhraní.....	20
4.2	PLC .....	20
4.3	Řídící jednotka servomotorů ACOPOS .....	21
4.4	Vstupní a výstupní moduly .....	22
4.5	Napájecí zdroj .....	23
4.6	Synchronní servomotor .....	24
4.7	Panel HMI .....	25
<b>5</b>	<b>Řešení problematiky</b> .....	<b>26</b>
5.1	Realizace programu.....	26
5.2	Popis vizualizace .....	27
5.2.1	Stránka Safety .....	27
5.2.2	Stránka pro řízení osy .....	28
5.2.3	Stránka pro pokročilý způsob řízení .....	30
5.2.4	Stránka alarmu .....	32
5.2.5	Ostatní stránky .....	33
5.2.6	Popis nezmíněných funkcí .....	34
<b>6</b>	<b>Závěr</b> .....	<b>36</b>

## Seznam obrázků

Obrázek 1: Stavový diagram [10] .....	13
Obrázek 2: Jazyk příčkového diagramu (LD) [8] .....	15
Obrázek 3: Jazyk seznamu instrukcí (IL) (převzato a upraveno z [7]) .....	15
Obrázek 4: Jazyk strukturovaného textu (ST) .....	16
Obrázek 5: Jazyk vývojového diagramu (SFC) [6] .....	17
Obrázek 6: Jazyk funkčního blokového schématu (FBD) [5] .....	17
Obrázek 7: Sestava z přední strany .....	18
Obrázek 8: Sestava ze shora .....	19
Obrázek 9: PLC X20CP0484 [13] .....	20
Obrázek 10: ACOPOS P3 [15] .....	21
Obrázek 11: Zapojené PLC s moduly X20 .....	22
Obrázek 12: Napájecí zdroj PS1050 .....	24
Obrázek 13: 8LVA synchronní motor .....	25
Obrázek 14: HMI panel 6PPT30.0573-20W [28] .....	25
Obrázek 15: Stránka Safety .....	28
Obrázek 16: Stránka pro řízení osy .....	30
Obrázek 17: Stránka pro víceosé řízení .....	32
Obrázek 18: Stránka alarmu .....	33

## Seznam tabulek

Tabulka 1: Popisek k obrázkům 7 a 8 .....	19
Tabulka 2: Technické parametry PLC X20CP0484 .....	20
Tabulka 3: Technické parametry ACOPOS P3 .....	21
Tabulka 4: Popis modulů X20 z obrázku 11 .....	22
Tabulka 5: Technické parametry napájecího zdroje .....	23
Tabulka 6: Technické parametry motoru .....	24
Tabulka 7: Technické parametry HMI .....	25

## Seznam zkratek

PLC	programovatelný logický automat
B&R	Bernecker & Rainer
CAN	řídící síť
I/O	vstup/výstup
HMI	Human Machine Interface
FB	funkční blok
EPL	Ethernet Powerlink
FRAM	Ferroelectric Random Access Memory
VAC	střídavé elektrické napětí
IP65	stupeň krytí

# 1 Úvod

Práce si klade za cíl ukázat použití standardu PLCopen pro řízení pohybu s využitím funkčních bloků, které jsou definované normou IEC 61131-3. Tato norma sjednocuje syntaxi jazyků pro programování řídicích jednotek (PLC). Součástí normy je i deklarace proměnných a datových typů, funkcí a funkčních bloků. Novým uživatelům a programátorům se vyplatí naučit se přímo programovat podle dané normy. Uvítají to především programátoři ve firmách, které jsou nuceny sestavovat své zakázky pokaždé s jinou značkou PLC. Stejná syntaxe programovacích jazyků jim může ušetřit čas a spoustu programátorských chyb.

Já se pokouším demonstrovat řízení pohybu s dvěma různými osami s využitím funkčních bloků v knihovně ACP10 ve vývojovém prostředí B&R Automation Studio. Nejprve řídím osy jednotlivě, které pracují nezávisle na sobě a poté mezi nimi vytvořím vazbu, kde si ve vizualizačním rozhraní zvolím, která z dvou os bude hlavní, a která vedlejší pomocí elektrického převodového spojení. Kromě přepínání servomotorů si také mohou zvolit například převodový poměr, rychlost otáčení, nebo i o kolik jednotek se daný motor obrátí. Pro synchronizaci pohybu využívám vazby Gear a Cam, které obsahují funkční bloky pro rozchození motorů. Pro řízení jsem navrhl vizualizaci v programovacím softwaru od B&R, kterou jsem nahrál do dotykového ovládacího panelu.

Začátek práce se věnuje normou pro řízení pohybu s popisem stavového diagramu. Další kapitola je věnována použitému hardwaru, který je využíván pro řízení os. Pro každé zařízení je uveden jednoduchý popis funkce v dané sestavě a základní parametry. V poslední části se zabývám řešením dané problematiky. Vysvětluji postup, kterým řídím dva motory a popisuji dané vizualizační rozhraní. Vizualizace se skládá z několika tlačítek pro pohyb os a nastavování vstupních parametrů. Celkem byly do předem nahreného softwaru od B&R přidány tři stránky. Dvě stránky jsou věnovány řízení jednotlivých os a v poslední stránce lze volit, která z os bude SLAVE a vazbu mezi nimi.

Zařízení byla sestavena dle požadavků mého vedoucího bakalářské práce, která budou využívána budoucími studenty oboru mechatronika. Ti si vyzkouší programování PLC, a kromě řízení pohybu servomotory si mohou vyzkoušet rozhraní vizualizačního prostředí a také funkcionalitu ochranných komponentů.

## 2 Řízení pohybu

Software v dnešní době hraje stále rostoucí roli v průmyslové automatizaci. Řízení motorů patří do důležité oblasti průmyslové automatizace, která dovoluje řídicí výstupy převést na pohyb. V průmyslu se často vyskytuje potřeba přesného řízení servomotorů a krokových motorů prostřednictvím průmyslových počítačů. [1].

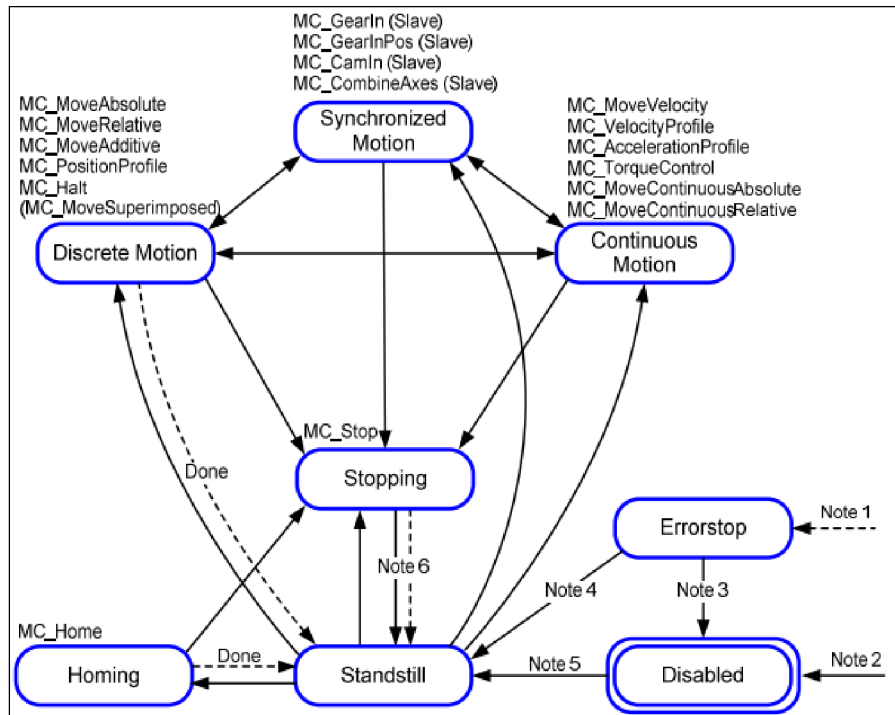
Následující knihovny funkčních bloků použité pro řízení motoru jsou vytvořené pro ovládání os, které jsou konzistentní s předefinovaným standardem IEC 61131-3. Rozhodlo se, že by nebylo ideální vložit všechny jednotlivé úkony jedné osy do jednoho funkčního bloku. Výsledkem bylo vytvoření sady příkazů ve formě funkčních bloků, které budou řídit námi zvolenou osu, což nám dá možnost přizpůsobení se, snadného použití a znovuzvolení.

### 2.1 Stavový diagram

Tento diagram definuje chování osy, když je využíváno více funkčních bloků pro řízení pohybu současně. To nám dá možnost lépe využívat a vytvářet komplexnější řízení a zvládnutí různých neregulérností. Nesmíme zapomenout, že pohybové instrukce jsou vždy sekvenční [10].

Podle stavového diagramu můžeme zjistit jaký je současný stav osy s porovnáním vstupních a výstupních proměnných v rámci PLC programu. Diagram se soustředí na popsání jednoosého řízení. Na víceosé řízení se musíme dívat z pohledu řízení více os najednou. Například řídicí osa je ve stavu plynulého pohybu, řízená osa je v synchronizovaném stavu a připojením této osy k řídicí nemá žádný vliv na řídicí osu [10].





Obrázek 1: Stavový diagram [10]

### 2.1.1 Popis stavového diagramu

Situace 1 (Note 1): Pokud nastane chyba na ose

Situace 2 (Note 2): Pokud `MC_Power.Enable` je ve stavu `FALSE` a není žádná chyba na ose.

Situace 3 (Note 3): Pokud `MC_Reset` a `MC_Power.Status` je ve stavu `FALSE`

Situace 4 (Note 4): Pokud `MC_Reset` a `MC_Power.Status` je ve stavu `TRUE` a `MC_Power.Enable` je ve stavu `TRUE`

Situace 5 (Note 5): Pokud `MC_Power.Enable` je ve stavu `TRUE` a `MC_Power.Status` je ve stavu `TRUE`

Situace 6 (Note 6): Pokud `MC_Stop.Done` je ve stavu `TRUE` a `MC_Stop.Execute` je ve stavu `FALSE`

(převzato a upraveno z [10])

## **3 PLCopen**

Softwarové náklady se zvyšují a pro řízení těchto nákladů je třeba zvýšit efektivitu v průběhu vývoje aplikací a zároveň zvýšit kvalitu softwaru a snížit náklady na údržbu a aktualizaci. PLCopen je nezávislá celosvětově známá organizace, která poskytuje efektivní automatizaci založenou na požadavcích uživatelů. Byla založena v roce 1992 a má sídlo v Nizozemsku [2].

Firma se postarala o vznik mezinárodně uznávané normy IEC 61131-3 pro řízení os s použitím definovaných funkčních bloků. Uživatel má k dispozici stejnou řadu příkazů a stylu programování při použití PLC od různých dodavatelů, kteří se zavázali používat danou normu [2].

Pro ovládání pohybu byly využívány široké škály hardwaru. V minulosti to vyžadovalo vytvoření jedinečného softwaru pro každou aplikaci i když funkce byla stejná. PLCopen poskytuje možnost využívání standardizovaných knihoven, které mohou být opakovaně použity pro více zařízení. Tímto se zmenší náklady na údržbu, vývoj a podporu a zároveň eliminuje případné zmatečné situace. Navíc se sníží náklady na školení a vývoj procesů, protože software je možné opakovaně používat. Vytvoříme menší závislost na programování hardwaru. Standardizace se také zaměřuje na použití v různých architekturách, včetně distribuovaných, integrovaných a síťových systémů. Jako takový je otevřený stávajícím a budoucím technologiím, protože není navržený pro jednu určitou aplikaci, ale může sloužit v různých průmyslových oblastech [4].

### **3.1 Norma IEC 61131-3**

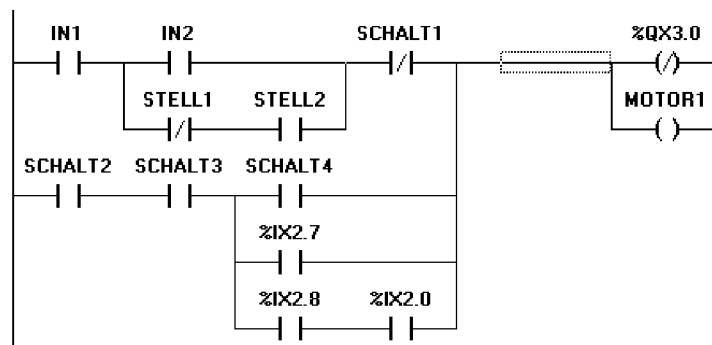
Programování PLC podle standardizované normy IEC 61131-3 ještě nezaručuje systematickост navrhovaného programu a jeho bezchybnost. Standard popisuje jak dané funkce nebo funkční bloky fungují a jaké mají vstupní a výstupní hodnoty. Více o funkčních blocích, které využívám v této práci popisují v kapitole 5.2. Norma sjednocuje syntaxi pro programování řídicích jednotek PLC. Uživatelé a programátoři PLC mohou používat programovací jazyk, které používají světový výrobci a bez větších potíží jim porozumějí [3].

#### **3.1.1 Programovací jazyky**

Podle standardu IEC 61131-3 je k dispozici pět typů programovacích jazyků:

- LD – Ladder Diagram (liniové nebo reléové schéma)
- IL – Instruction List (posloupnost instrukcí)
- ST – Structured Text (vyšší programovací jazyk)
- SFC – Sequential Function Chart (vývojový diagram)
- FBD – Function Block Diagram (schéma funkčních bloků)

**Ladder Diagram** je grafický programovací jazyk, kde pomocí propojení několika relé lze vytvořit logické funkce typu OR, AND, NAND, XOR. Výhodný je především pro svoji přehlednost a jasně danou posloupnost programu. Bohužel s rostoucí složitostí programu se zvětšuje jeho délka a také je méně vhodný pro práci s daty, jakou jsou řetězce, a aritmetické operace [5].



Obrázek 2: Jazyk příčkového diagramu (LD) [8]

**Instruction List** patří mezi základní možnost pro popis programu, který se podobá assembleru. Zpracování programu je rychlé a paměťově nenáročné, ale psaní programu je většinou zdlouhavé a lehce nepřehledné, a proto se nehodí pro návrh složitých aplikací [5].

LD	bVar	
ST	fbTimer1.IN	
JMP	mark1	
CAL	fbTimer1 (	
	PT:=tIn1,	
	ET:=>tOut1)	
LD	fbTimer1.Q	
ST	fbTimer2.IN	
mark1:		
LD	nVar	
ADD	230	

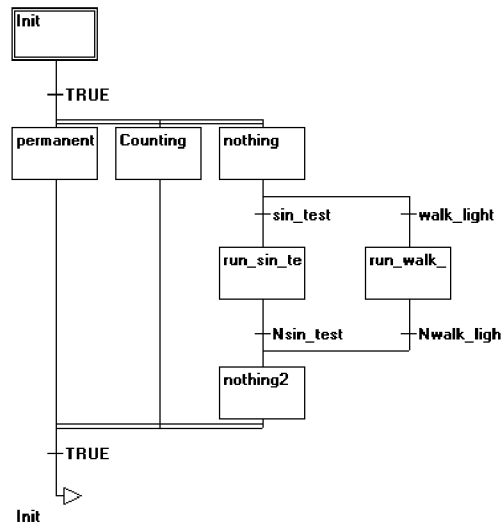
Obrázek 3: Jazyk seznamu instrukcí (IL) (převzato a upraveno z [7])

**Structured Text**, který je využíván v této bakalářské práci, je pro klasické programátory logických automatů ideální. Je to výkonný vyšší programovací jazyk, který zahrnuje důležité prvky moderního programovacího jazyka, jako je větvení (IF nebo CASE) i iterační smyčky (WHILE nebo FOR). Je vhodný nástroj pro vytváření funkčních bloků, které mohou být vnořeny v jakémkoliv jiném programovacím jazyku. Výhodný je pro snadné programování složitých operací, práci s textovými řetězci, a se značnými bloky dat a databázemi. Je potřeba znát přesnou syntaxi zápisu a příkazy a je méně přehledný ohledně zpracování velkého počtu logických vstupů a výstupů [5].

```
IF (slave2 = TRUE AND start_slave = TRUE) THEN
  CamControl.Command.StartSlave := TRUE;
  CamControl.Command.DisengageSlave := FALSE;
  disengage_slave := FALSE;
  start_slave := FALSE;
ELSIF (slave2 = TRUE AND disengage_slave = TRUE) THEN
  CamControl.Command.DisengageSlave := TRUE;
  CamControl.Command.StartSlave := FALSE;
  start_slave := FALSE;
  disengage_slave := FALSE;
END_IF
```

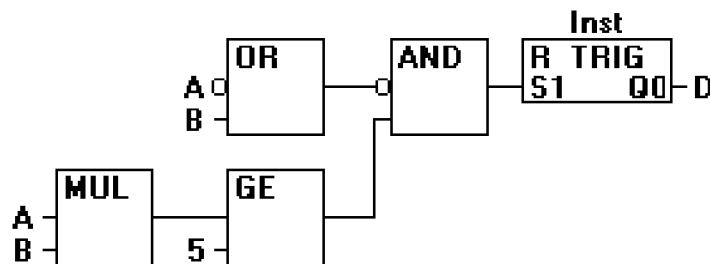
Obrázek 4: Jazyk strukturovaného textu (ST)

**Sequential Function Chart** je zápis, který je graficky orientovaný pro lehčí definování programu, konkrétně jeho chod a odezva na různé rozhodovací případy. Definiuje program na základě vývojového schéma, tedy větvení na základě vyhovění rozhodovacích podmínek. Přehledně definuje a ošetřuje různé stavy, a také je vhodný pro snadnou práci s ASCII řetězci. Není výhodný pro programování složitých algoritmů, práci s analogovými signály nebo velkým počtem logických signálů [5].



Obrázek 5: Jazyk vývojového diagramu (SFC) [6]

**Function Block Diagram** vyjadřuje chování programu ve funkčních blocích, které jsou za sebou propojené a spouští různé funkce, ale není možné volně programovat na programovací ploše, jelikož je vše spořádané do řádků podobně jako v LD jazyku. Využívá obvyklých značek hradel OR, AND a podobně. Zápis programu je přehlednější a je ideální pro práci s velkým počtem logických signálů. Na druhou stranu je méně vhodný pro náročnější zpracování analogových signálů a pro vytváření složitých algoritmů [5].

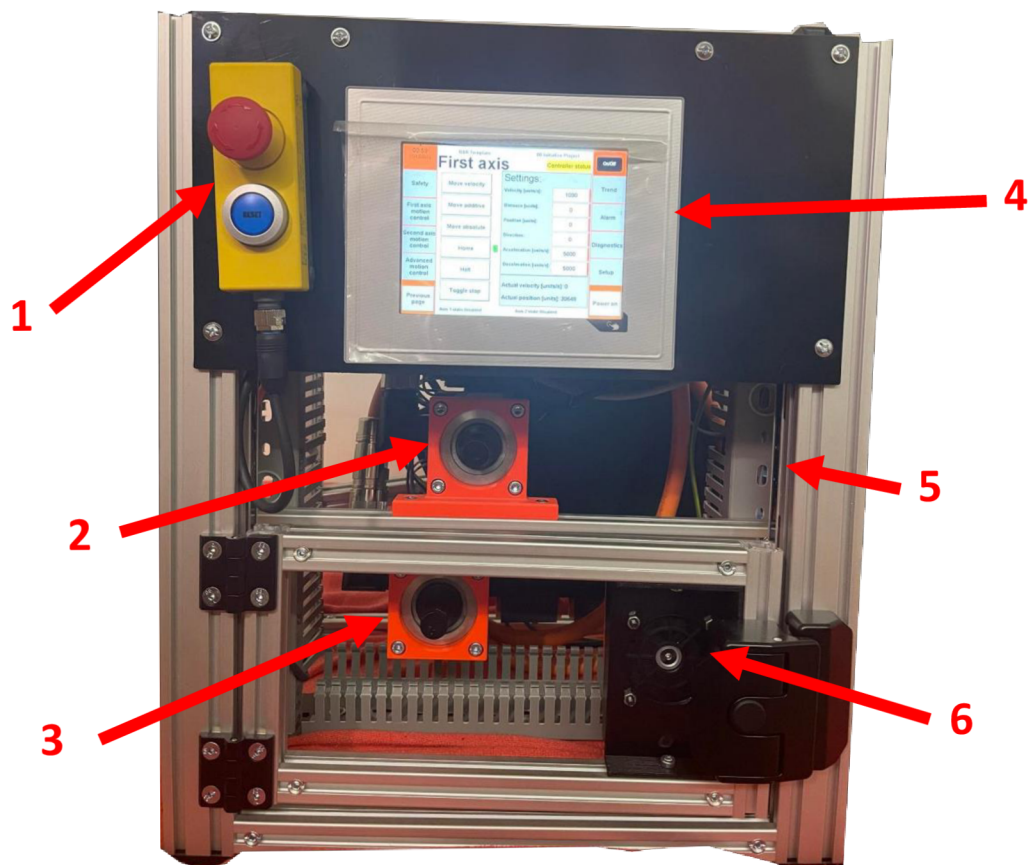


Obrázek 6: Jazyk funkčního blokového schématu (FBD) [5]

## 4 Hardwarové uspořádání

V bakalářské práci bylo použito mnoho elektronických zařízení. V této části bych se zaměřil na funkci zařízení a popis základních parametrů. Jak již bylo zmíněno, moje práce se zaměřuje na demonstraci řízení pohybu dvou motorů. Pro mé zadání jsem využil tlačítka **CENTRAL STOP** a **RESET**, dotykový panel, servozesilovač **ACOPOS P3**,

napěťový zdroj PLC s I/O digitálními moduly a dva synchronní servomotory. Zbývající prvky jako bezpečnostní závory, blokovací zařízení s jištěním a bezdotykové blokovací zařízení od firmy SICK jsou uplatňovány pro demonstraci bezpečnostních funkcí. Vše je osazeno do pevného hliníkového mechanického rámu pro lepší seskupení všech komponentů a snadnou manipulaci.



Obrázek 7: Sestava z přední strany



Obrázek 8: Sestava ze shora

Tabulka 1: Popisek k obrázkům 7 a 8

1	Tlačítka Nouzové zastavení a RESET
2	Synchronní servomotor pro řízení druhé osy
3	Synchronní servomotor pro řízení první osy
4	Dotykový panel HMI
5	Bezpečnostní světelné závory
6	Blokovací zařízení s jištěním
7	PLC s I/O digitálními moduly
8	ACOPOS P3
9	Zdroj napětí

## 4.1 Komunikační rozhraní

Abychom mohli zajistit rychlou a spolehlivou komunikaci mezi zařízeními, je třeba použít technologie, které splňují všechny dané požadavky. Díky nabízenému sortimentu od společnosti B&R jsou tyto aspekty vhodně vyřešeny.

Komunikace mezi procesorovou jednotkou X20 a zesilovačem servomotorů ACOPOS je zajištěna technologií standardu Ethernetu POWERLINK, protože klasický standard Ethernetu nedokáže zaručit přenos dat v reálném čase [12]. Propojení dotykového ovládacího panelu je řešené klasickým standardem Ethernetu.

## 4.2 PLC

Byla použita jednotka modelu X20CP0484 (více viz [16]), která patří mezi hlavní části celého systému. Jak již bylo zmíněno, ovládá ACOPOS, který dále komunikuje a řídí servomotory a také zajišťuje spojení mezi ovládacím panelem. Jako samotný neobsahuje žádné vstupní a výstupní moduly.

Tabulka 2: Technické parametry PLC X20CP0484

Procesor	ARM Cortex-A9 667 MHz
Paměť	256 MB DDR3 SDRAM
Remanentní proměnné	64 kB FRAM
Spotřeba energie	2,95 W
Komunikační rozhraní	1x Ethernet, 1x POWERLINK, 2x USB, 1x X2X



Obrázek 9: PLC X20CP0484 [13]



### 4.3 Řídicí jednotka servomotorů ACOPOS

Pro řízení dvou synchronních servomotorů byl použit model jednotky ACOPOS P3 8EI4X5MWDS0.0200-1 (více viz [14]). Zařízení je využíváno pro dvouosé řízení servopohonů. Bez PLC není možné řídit motory samostatně, protože nedisponuje žádnými ovládacími parametry, ale pouze monitoruje a řídí průběh otáčení motoru danými algoritmy. O napájení motoru se stará výkonová část, která obsahuje napěťový frekvenční měnič aplikující pulsní šířkovou modulaci.

Tabulka 3: Technické parametry ACOPOS P3

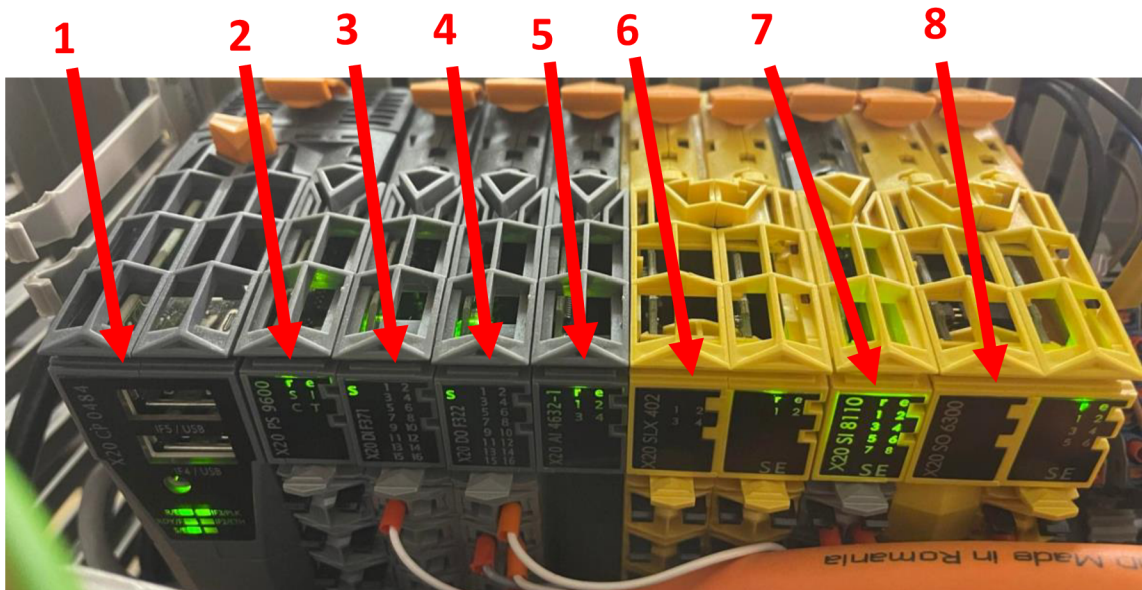
Vstupní napětí	1x 110-230 VAC, 3x 200-230 VAC
Frekvence	50-60 Hz
Síťové rozhraní	TN-S, TN-C-S s uzemněním
Počet připojených motorů	2
Komunikační rozhraní	2x RJ45 POWERLINK, 100 Mbit/s
Stacionární proud	4,5 A



Obrázek 10: ACOPOS P3 [15]

## 4.4 Vstupní a výstupní moduly

Pro lepší kompaktnost byly pořízeny moduly systému X20, které se připojily k PLC, aby bylo možné komunikovat s použitými zařízeními, které jsou připojeny digitálními kanály. Vše je společně s procesorovou jednotkou instalováno na nosnou DIN lištu. Připojení je rozděleno do tří částí a každý z modulů je napájen sběrnici X2X link. V sestavě jsou využívány i bezpečnostní karty, které zajišťují spolehlivé vykonávání procesů.



Obrázek 11: Zapojené PLC s moduly X20

Tabulka 4: Popis modulů X20 z obrázku 11

1	X20CP0484	Programovatelný logický automat.
2	X20PS9600	Napěťový modul používaný společně s PLC. Dodává elektrickou energii do ostatních modulů díky rozhraní X2X (více viz [19]).
3	X20DIF371	Modul s 16 digitálními vstupy (více viz [20]).
4	X20DOF322	Modul s 16 digitálními výstupy (více viz [21]).

5	X20AI4632-1	Modul se nevyužívá v této práci. Disponuje 4 analogovými vstupy a 16bitovým digitálním převodním rozlišením. (více viz [22]).
6	X20SLX402	Modul s 4 bezpečnostními digitální vstupy, 2 digitálními výstupy a 4 impulsními výstupy (více viz [23]).
7	X20SI8110	Modul s 8 bezpečnostními digitálními vstupy a 4 impulsními výstupy (více viz [24]).
8	X20SO6300	Modul s 6 bezpečnostními výstupy, který mají za úkol zabránit automatickému restartu, pokud nastane chyba (více viz [25]).

## 4.5 Napájecí zdroj

Zařízení PS1050 (více viz [18]) transformuje jednofázové síťové napětí 230 VAC do stejnosměrného napětí o 24 V. Napájí všechna elektrická zařízení obsažená v sestavené jednotce.

Tabulka 5: Technické parametry napájecího zdroje

Vstupní napětí	100-240 VAC
Vstupní proud	Přibližně 1,65 A (120 VAC), 0,9 A (230 VAC)
Výstupní napětí	24 VDC
Výstupní proud	5 A
Výstupní výkon	120 W
Účinnost	>89 %
Spolehlivost	500 000 h



Obrázek 12: Napájecí zdroj PS1050

## 4.6 Synchronní servomotor

Pohyb osy je vykonáván synchronním servomotorem 8LVA23.B1030D000-0, napájení motoru je zajištěno hybridním kabelem (více viz [26]). Otáčky jsou řízeny frekvenčním měničem obsaženém v jednotce ACOPOS. Pro účely této práce jsou používány dva stejné motory, aby bylo možné demonstrovat nezávislý pohyb dvou os, a také jejich synchronizované ovládání pomocí funkčních bloků definovaných normou IEC 61131-1.

Tabulka 6: Technické parametry motoru

Rychlost otáčení	3000 ot./min.
Počet pól párů	4
Moment	1,3 Nm
Výkon	408 W
Záběrový proud	5,8 A
Záběrový moment	1,35 Nm
Brzdový moment	2,2 Nm



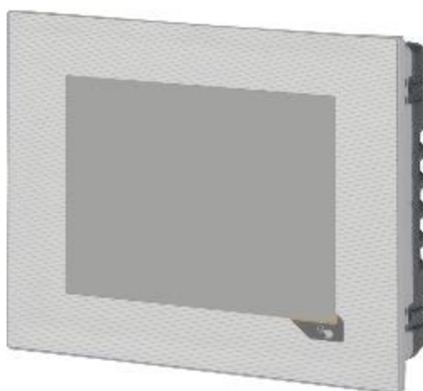
Obrázek 13: 8LVA synchronní motor

## 4.7 Panel HMI

Pro vizualizaci je použitý operátorský grafický dotykový panel 6PPT30.0573-20W. Přední část poskytuje ochranu IP65, čímž je zařízení dostatečně kryté vůči nežádoucím vlivům průmyslového prostředí (více viz [27]).

Tabulka 7: Technické parametry HMI

Procesor	ARM Cortex-A8, 600 MHz
Paměť	512 MB
Velikost obrazu	5,7", 640 x 480 VGA
Rozhraní	2x Ethernet, 2x USB 2.0
Nahrany software	Servisní stránka, VNC klient a webový prohlížeč



Obrázek 14: HMI panel 6PPT30.0573-20W [28]

## 5 Řešení problematiky

Úkolem bylo demonstrovat funkcionalitu řízení dvou os. Pro tento účel byly použity a upraveny knihovny ACP10 MC v programovacím softwaru Automation Studio od B&R, které poskytují mnoho funkčních bloků starající se o pohyb. V počítačovém softwaru se vybral ukázkový program pro synchronní dvouosé řízení, kde jedna z os má vazbu SLAVE (řízená) a druhá MASTER (řídící).

Nejprve jsem vyzkoušel řízení jedné osy v předepsaném programu od B&R. Jednalo se sice o synchronní řízení vačkového mechanismu, ale pro začátek nebylo využíváno. Po úspěšném řízení jedné osy jsem vytvořil stejný program, který byl upraven pro řízení druhé osy. Následně jsem si úspěšně vyzkoušel ovládání konstantního pohyb dvou motorů nezávisle na sobě.

Pro pokročilý způsob řízení byl použit synchronizovaný pohyb pomocí elektronických vaček a elektronického převodu. Tyto dva způsoby synchronního pohybu jsem programově sloučil. Uživatel si zvolí, jakou osu použije jako SLAVE a jakým typem vazby bude řízena. Pro tuto možnost jsem vytvořil vizuální prostředí, které je zobrazeno v HMI dotykovém panelu.

### 5.1 Realizace programu

Pro synchronizované řízení používám dvě metody, jedna z metod je elektronický převod a druhá elektronická vačkou. Mým cílem bylo zjednodušení vykonávání programu. Jelikož jsem využíval předem vytvořený program, který mi byl nabízen v Automation Studiu, na první pohled bylo vidět, že každá z metod řízení používá spoustu stejných funkčních bloků.

Obě metody jsem zkombinoval, a to tak, že jsem funkční bloky pro řízení elektrickým převodem použil v programu pro řízení elektronickou vačkou. Samozřejmě bylo třeba program upravit tak, aby funkční bloky pro řízení elektrickým převodem mohly získávat informace o parametrech kterými jsou například rychlost, zrychlení nebo i zastavení. To samé bylo potřeba nastavit i pro druhou osu. Danou realizaci programu budu konkrétněji popisovat v dalších kapitolách, kde se zabývám jednotlivými vizualizačními stránkami v další kapitole.

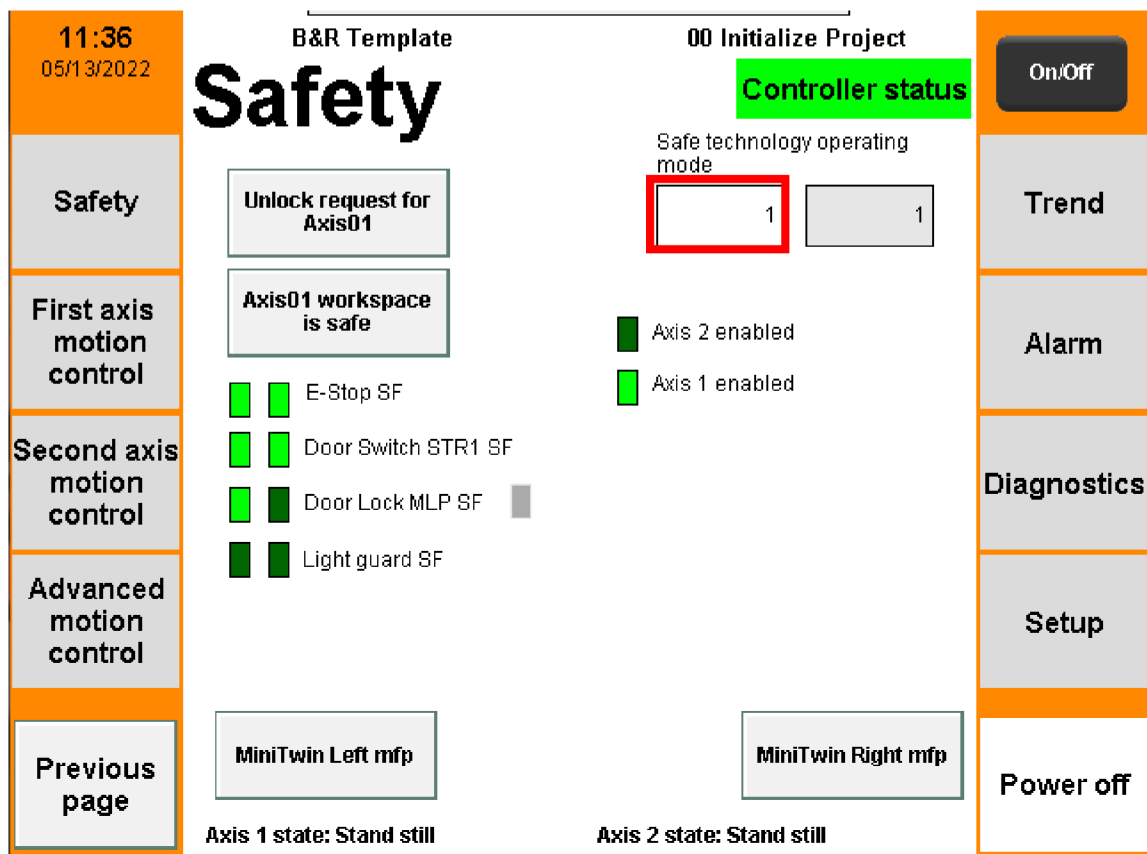
## 5.2 Popis vizualizace

Dlouho jsem se zamýšlel, jaká by byla nejlepší forma popisu mého řešení. Nakonec jsem zvolil metodu, kde budu popisovat jednotlivé stránky v HMI. Pro mé řešení jsem použil šablonu vytvořenou výrobcem a pouze jsem ji upravil, aby bylo možné demonstrovat moje řešení práce. Každá stránka obsahuje různé funkce pro řízení pohybu a jejich parametrizaci společně s indikačními kontrolkami.

V této kapitole se zaměřuji na deskripci správného postupu pro ovládání os a jejich nastavování. Zmíním také funkcionalitu jednotlivých indikačních kontrollek a také řešení alarmových hlášení. Pro přechod mezi jednotlivými stránkami jsou přidána tlačítka. Abych nemusel používat k ovládání neustále HMI panel na sestaveném zařízení, tak jsem využíval jsem software VNC Viewer verze 5.3.0

### 5.2.1 Stránka Safety

Tato část nepatří k mému řešení, ale je potřebná k povolení řízení motorů. Stránka obsahuje několik indikátorů, které poskytují informaci o SAFETY komponentech. Pro moji práci je důležité zvolení módu a jeho následné odsouhlasení pomocí modrého tlačítka RESET. Dále je potřeba zkontrolovat tlačítko TOTAL STOP, zda není v sepnutém stavu, jelikož slouží pro zastavení přísunu elektrické energie do jednotky ACOPOS.



Obrázek 15: Stránka Safety

Stisknutím na červeně vyznačenou oblast na obrázku 15 se zobrazí číselná klávesnice. Je potřeba zvolit číslo 1, kterým se nastaví mód pro vykonávání bezpečnostního programu, který povolí řízení motorů.

## 5.2.2 Stránka pro řízení osy

Na této stránce je možné řídit osu za použitím funkčních bloků z knihovny PLCopen. Je zde možné nastavit i několik vstupních parametrů. Budu se zde zabývat funkcí jednotlivých tlačítek a jakými funkčními bloky se na této stránce zabývám. Pro ovládání os se používá stejná stránka, akorát jsou použity odlišné reference pro každý motor.

### Tlačítko Move velocity

Tímto tlačítkem volám funkční blok `MC_MoveVelocity`, kterým se dostane osa do nepřetržitého pohybu. Je potřeba mít nastavený vstupní parametr `Velocity`, který nám udává, jak rychle se motor bude otáčet. V inicializační části programu je tato hodnota nastavena na 1000 jednotek za sekundu. Jedna otáčka motoru se rovná 1000 jednotek.



### **Tlačítko Move additive**

K tomuto tlačítku se váže nastavení hodnoty `Distance`. Tuto hodnotu je potřeba mít nastavenou, jelikož nám udává, o kolik jednotek se osa bude pohybovat. Tímto způsobem volíme funkční blok `MC_MoveAdditive`, který nám zaručuje pohyb na ose po určité vzdálenosti.

### **Tlačítko Move absolute**

Při stisknutí se motor bude točit do nastavených jednotek. Pro tuto funkci je použit funkční blok `MC_MoveAbsolute`. K tomu, aby bylo možné nastavit do jakých jednotek se bude osa otáčet využijeme vstupní parametr `Position`.

### **Tlačítko Home**

Po stisknutí se motor vrátí do nulté pozice. Je možné ji v parametrech změnit na požadovanou hodnotu. Tlačítko spustí funkční blok `MC_Home`. Pokud je tento funkční blok aplikován ve stejný moment, ve který je osa v pohybu, tak se vyskytne chyba, která udává, že tento typ pohybu není povolen. Proces je nutný k aktivování pohybu. Pokud je řídicí systém ACOPOS ve stavu, kdy ho není možné spustit a přejde do stavu, ve kterém to možné je, tak je potřeba provést „homing“. Chyba nastane, pokud se volají funkční bloky pro pohyb osy, když není vykonán „homing“. K procesu byla vytvořena kontrolka, která se nachází vpravo od daného tlačítka. Pomocí ní lze zjistit, zda na ose byl proveden „homing“. Pokud ano, tak kontrolka bude svítit světle zelenou barvou, naopak bude svítit tmavou zelenou barvou. Aby bylo možné provést tuto aplikaci, tak je zapotřebí aby byla jednotka ACOPOS spuštěna.

### **Tlačítko Halt**

Tlačítkem lze přerušit jakýkoliv pohyb na ose. K tomu se využívá funkční blok `MC_Halt`. Osa přejde do stavu `DiscreteMotion`, dokud není rychlost osy na nule, poté přejde do stavu `StandStill`. Po vykonání procesu je možné znovu osu otáčet. Pokud jsme v synchronním pohybu dvou os a na řízenou osu SLAVE zavoláme funkci `MC_Halt`, tak se zastaví pouze řízený motor, který přejde do stavu `StandStill`. Řídicí motor bude stále běžet a bude ve stavu `ContinuousMotion`. Jestliže zastavíme MASTER osu, tak se zastaví i SLAVE osa, protože pohyb řízené osy závisí na pohybu osy řídicí.

## Tlačítko Toggle Stop

Po stisknutí tlačítka se spustí funkční blok MC\_Stop, který zastaví osu a přivede ji do stavu Stopping. Pokud je osa v tomto stavu, tak není možné provést žádný jiný pohyb. Osa je ve stavu Stopping stále i poté, co přejde do nulové rychlosti. Tlačítko po stisknutí zůstane v sepnutém stavu. Je potřeba tlačítka znovu stisknout a tím přivést do počáteční polohy. Tím se vykonávání funkčního bloku zastaví a je možné osou znova otáčet.

The screenshot displays the 'First axis' control interface. At the top, it shows 'B&R Template' and '00 Initialize Project'. The main title is 'First axis'. Below the title, there are several control buttons: 'Move velocity', 'Move additive', 'Move absolute', 'Home', 'Halt', and 'Toggle stop'. A green indicator light is visible next to the 'Home' button. The 'Settings' panel includes the following parameters and values:

Parameter	Value
Velocity [units/s]:	1000
Distance [units]:	0
Position [units]:	0
Direction:	0
Acceleration [units/s <sup>2</sup> ]:	5000
Deceleration [units/s <sup>2</sup> ]:	5000

Below the settings, the actual values are displayed: 'Actual velocity [units/s]: 0' and 'Actual position [units]: 8264'. The status of the axes is 'Axis 1 state: Disabled' and 'Axis 2 state: Disabled'. The interface also features a vertical menu on the left with buttons for 'Safety', 'First axis motion control', 'Second axis motion control', 'Advanced motion control', and 'Previous page'. On the right, there is a vertical menu with buttons for 'On/Off', 'Trend', 'Alarm', 'Diagnostics', 'Setup', and 'Power on'. The top left corner shows the time '13:48' and date '05/13/2022'.

Obrázek 16: Stránka pro řízení osy

### 5.2.3 Stránka pro pokročilý způsob řízení

Na této stránce řídím osy, které jsou v synchronizovaném stavu. To znamená, že si volím, která z os bude řízená a která řídící. Pro vykonávání jsem použil funkční bloky pro víceosé řízení, kterými se dále budu zabývat.

Pro správné řízení je nutné vybrat, která osa bude SLAVE. Proto jsou použity dva programy. V programu je poté použitý jeden motor pro řízení první osy jako SLAVE a

v druhém programu je motor pro řízení druhé osy nastaven také na SLAVE. Tlačítka `Axis 1` a `Axis 2` si volím, která z os bude v synchronním stavu.

Dále je potřeba zvolit typ synchronního pohonu. Na výběr je zde ovládání elektronickou vačkou nebo elektronickým převodem.

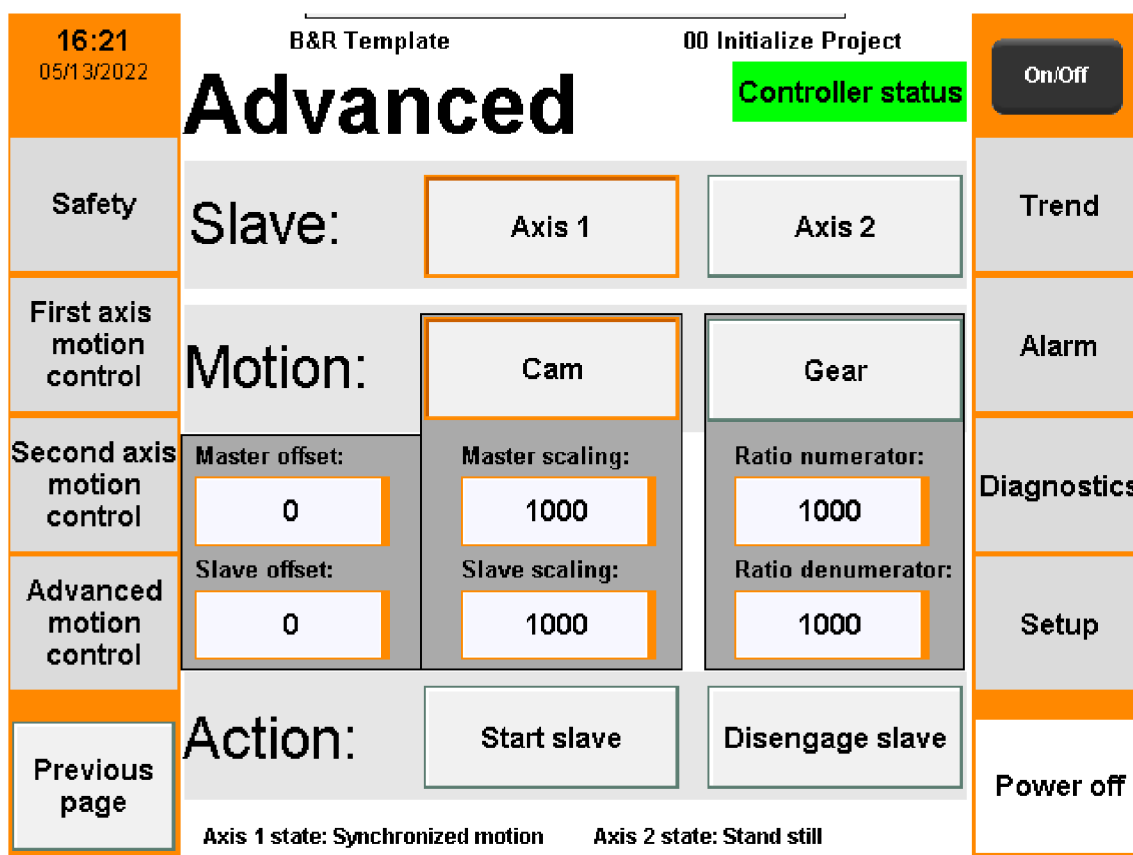
Pro pohyb elektronickou vačkou jsou použity funkční bloky `MC_CamIn` a `MC_CamOut`. Pokud je vykonána jedna otáčka řízeného motoru, tak se změní směr otáčení a takhle je motor řízen neustále závisle na řídicí ose. Po jedné otáčce MASTER osy se vykonají dvě otáčky u SLAVE osy (jedna jedním směrem a druhá směrem opačným). Pro dodatečné nastavení pohybu lze využít vstupní okna pod tlačítkem `Cam`.

Elektrický převod vykonává funkci, při které se osy otáčejí nastaveným převodovým poměrem. Pro tuto funkci jsou vykonávány bloky `MC_GearIn` a `MC_GearOut`. Pro nastavení převodového poměru lze využít vstupní okna, který se nacházejí pod tlačítkem `Gear`.

Aby bylo možné přejít do stavu `SynchronizedMotion` pro zvolenou SLAVE osu, je nutné stisknout tlačítko `Start slave`. Ukončení vazby pro danou osu se vykoná stisknutím tlačítka `Disengage slave`.

Tlačítka je nutné znovu stisknout, pokud chceme provést jiný výběr. Například pokud je zvolena `Axis 1` a je potřeba změnit výběr na `Axis 2`, musí se zrušit výběr předchozí volby znovu stisknutím tlačítka `Axis 1`. To samé i pro volbu synchronizovaného pohybu.

Jak již bylo zmíněno, pro vykonávání tohoto pohybu bylo potřeba vytvořit dva programy. Výběrem některého z tlačítek bylo potřeba předat informaci do obou programů. Abych zajistil, že daný výběr bude vykonán správně, tak bylo potřeba vytvořit pomocné globální proměnné. Tyto proměnné jsou pak vázány na tlačítka pro výběr pokročilého pohybu a následnou kombinací se vykonala část programu. Tímto jsem eliminoval spoustu dalších tlačítek, které by byli potřeba pro řízení a nastavování pro každou osu zvlášť.

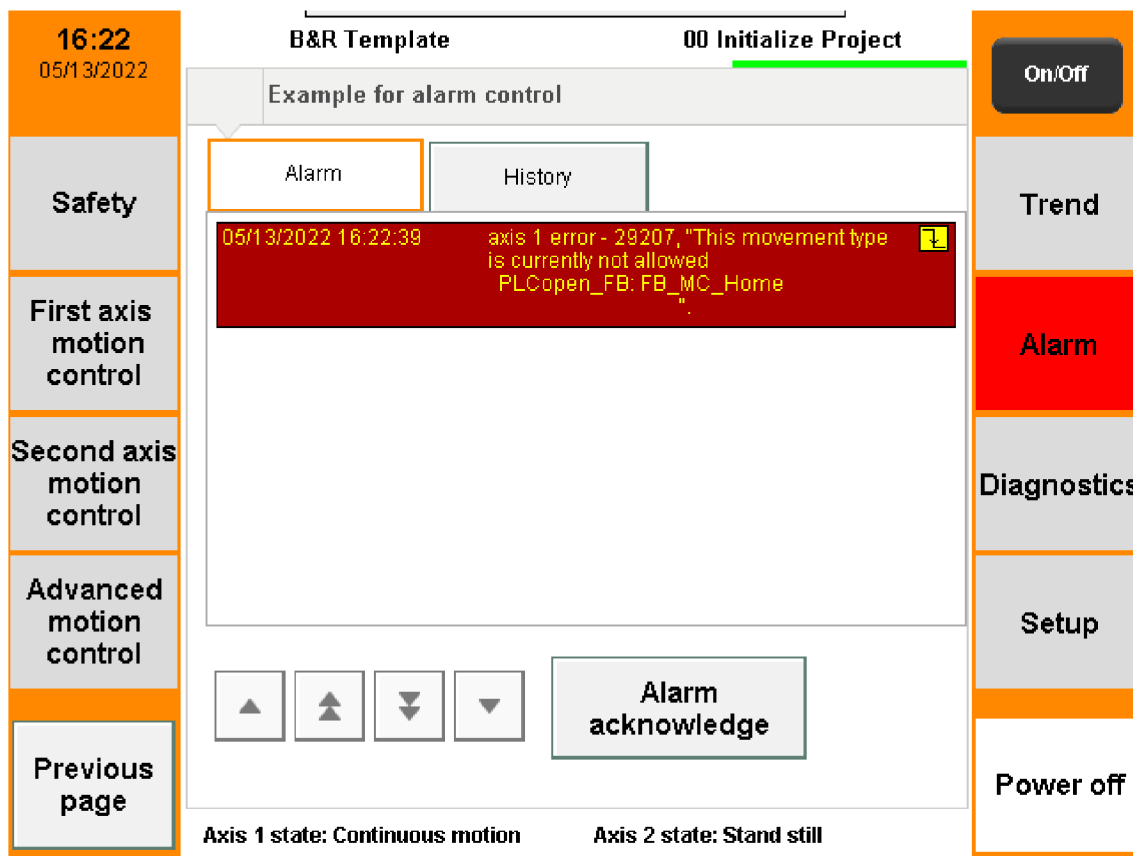


Obrázek 17: Stránka pro víceosé řízení

## 5.2.4 Stránka alarmu

Na této stránce jsou obsaženy alarmy, které vzniknou při procesu řízení. Alarmové okno bylo převzato a upraveno dle potřeby. Tlačítko `Alarm` bylo změněno, tak aby bylo možné indikovat chybu na první pohled nehlédě na to jaká stránka je zrovna zobrazena. Tlačítko se změní na červenou barvu a poté co se odsouhlasí všechny chyby, tak se barva změní zase zpátky.

Potvrzení probíhá tlačítkem `Alarm Acknowledge`. Po stisknutí chyba z alarmového okna zmizí. Pokud vzniklo více chyb, je potřeba všechny odsouhlasit postupně. V historii pak lze vidět všechny chyby, které nastaly od zapnutí systému.



Obrázek 18: Stránka alarmu

## 5.2.5 Ostatní stránky

Stránky, které nebyly zmíněny v předchozích kapitolách nejsou pro řešení této práce využívány. Vzhledem k tomu, že jsou funkční a je možné na ně přejít, tak bych jim věnoval krátký popis o tom, co se na každé z nich nachází.

### Stránka Trend

Na stránce lze získat aktuální informaci o teplotě procesoru programovatelného automatu a teplotě uvnitř pouzdra. Z grafu lze vyčíst, že teplota procesoru se pohybuje kolem 68 °C a teplota v pouzdru kolem 52 °C.

### Stránka Diagnostics

Stisknutím lze přejít do internetového rozhraní, kde lze získat spoustu informací o systému, softwaru, hardwaru nebo pohonech.

Konkrétně zde můžeme zjistit například informace o procesoru. Jeho softwarovou verzi, teplotu nebo procentuální paměťové využití. Nachází se zde i informace o všech připojených PLC karet ale i motorech. Ze stránky Motion zjistíme konkrétní název

připojeného motoru, na jakou adresu je připojen, v jaké je poloze, jeho aktuální rychlost anebo jestli je ve stavu `Enable`.

### **Stránka Setup**

Na stránce lze měnit jazykovou volbu, pokud je vizualizace psána ve více jazycích. Dále je zde možnost změnit jas a kontrast HMI. Nastaví se zde i adresa PLC nebo čas a datum.

## **5.2.6 Popis nezmíněných funkcí**

Zde budou popsány funkce, které nebyly zmíněny v předchozích kapitolách, které se věnovaly popisu vizualizačního rozhraní. Jedná se o tlačítka a informační texty, které se nachází na každé mnou vytvořené stránce.

### **Tlačítko Power**

Tlačítkem se ovládají pohony. Aby nebylo třeba vytvořit tlačítka pro zapnutí každé osy zvlášť, tak bylo navrženo jedno tlačítko, které nastavuje osy ze stavu `Disabled` do stavu `StandStill`. Po stisknutí tlačítka nelze s osami otáčet a je potřeba zavolat funkční blok, kterým se pohon začne otáčet

### **Stav osy**

Pro tento účel byly vytvořeny ve vizualizačním prostředí textové skupiny, které se měnily na základě proměnné `AxisState`. Parametry lze sledovat ve spodní části dotykového panelu. Stav dvou os lze sledovat neustále mnou vytvořených stránkách. V globálních proměnných jsem přidal proměnnou, která měnila hodnoty na základě současného stavu osy. Například pokud se nastavila osa do stavu `StandStill`, tak k proměnné `AxisState` byla přiřazena hodnota 1. K této hodnotě byl přiřazen text „Stand still“, který se zobrazil v textovém poli.

Také byla vytvořena pole pro sledování aktuální rychlosti motoru a pozice, kterou osa urazila. Pokud se osa otáčí opačným směrem, tak je tato hodnota odečítána a může dosahovat i záporných hodnot. Sledování pozice a rychlosti je možné sledovat pro každou osu zvlášť na daných stránkách.

### **Zrychlení a zpomalení**

Dalším parametrem je možnost nastavení zrychlení a zpomalení pro každý motor zvlášť

### **Kontrolka pro sledování stavu řídicí jednotky**

Kontrolka se nachází v horním pravém rohu a indikuje nám stav jednoty ACOPOS. Kontrolka mění barvy podle toho, v jakém stavu se jednotka zrovna nachází. Pokud kontrolka svítí červeně, jednotka není připravená k zapnutí. Pro žlutou barvu platí, že je jednotka připravena ke spuštění. Zelená barva indikuje stav, pro který ACOPOS řídí dané pohony.

## 6 Závěr

V práci byla aplikována metoda řízení dvou pohonů pomocí knihoven definovaných normou PLCopen. Dle předem zadaných požadavků vedoucího práce byla vytvořena dvě totožná zařízení, která by měla být v budoucnu výukovým prostředkem oboru mechatronika. Dané komponenty jsem osadil do mechanického rámu a elektricky propojil dle zadané dokumentace.

Pro řízení synchronizované řízení os sloužily programy pro Cam a Gear, které byly vloženy z knihovny Automation Studio od B&R. Pro zjednodušení přehlednosti byly použity funkce, které bylo nezbytné použít pro řízení synchronizovaných. V návaznosti na tuto fázi byly funkce sloučeny do jednoho funkčního programu. Bylo potřeba pozměnit hodnoty takovým způsobem, aby funkční bloky získávaly vstupní parametry, které byly nutné pro jejich funkcionalitu. Za účelem správného určení osy SLAVE bylo potřeba vytvoření dvou programů, jejichž vstupní parametry byly pozměněny pro přiřazení příslušné reference správné ose

V poslední fázi práce bylo vytvořeno uživatelské rozhraní, které bylo následně nahráno do HMI. Výrobcem předem dané vizualizace jsem doplnil celkem o tři stránky, které se zabývaly demonstrací pohybu dvou os. Každá osa zvlášť sleduje aktuálních parametry a možnosti změn. Dále byla předvedena tlačítka zabývající se jednoduchým řízením osy. Závěrem je prezentována možnost volby SLAVE osy a dané vazby. Tu doplňuje eventualita změny parametrů, které patří ke každé vazbě zvlášť.

Chtěl bych také zmínit návrh o vylepšení této práce. Pro dodatečné řízení elektronickou vačkou je možnost využít funkční blok `MC_CamTableSelect`. Tímto způsobem je možnost vybírání z tabulky vaček, které jsou definované uživatelem, pro použití funkčním blokem `MC_CamIn`. Poté vstupem `Periodic` lze měnit periodický nebo neperiodický vačkový profil.



## Použitá literatura

- [1] *Řízení motorů (Motion Control)* [online]. Antonín Vojáček, 2004 [cit. 2022-04-08]. Dostupné z: <https://automatizace.hw.cz/clanek/2005090801>
- [2] *What is PLCopen* [online]. [cit. 20220408]. Dostupné z: <https://plcopen.org/what-plcopen>
- [3] *Programování PLC podle normy IEC EN 61131-3 – víc než jednotné jazyky* [online]. Luboš Urban, [cit. 2022-04-10]. Dostupné z: [https://automa.cz/cz/casopis-clanky/programovani-plc-podle-normy-iec-en-61131-3-vic-nez-jednotne-jazyky-2005\\_02\\_30310\\_1237/](https://automa.cz/cz/casopis-clanky/programovani-plc-podle-normy-iec-en-61131-3-vic-nez-jednotne-jazyky-2005_02_30310_1237/)
- [4] *Creating Reusable, Hardware Independent Motion Control Applications via IEC 61131-3 and PLCopen Function Blocks* [online]. [cit. 2022-04-08]. Dostupné z: [https://plcopen.org/sites/default/files/downloads/plcopen\\_motion\\_control\\_overview.pdf](https://plcopen.org/sites/default/files/downloads/plcopen_motion_control_overview.pdf)
- [5] *Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys)* [online]. Antonín Vojáček, 2011 [cit. 2022-04-10]. Dostupné z: <https://automatizace.hw.cz//programovaci-rezimy-pro-plc-dle-iec-611313-codesys>
- [6] *Sequential Function Chart (SFC)* [online]. [cit. 2022-04-08]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/TcPlcCtrl\\_Languages%20SFC.htm&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/TcPlcCtrl_Languages%20SFC.htm&id=)
- [7] *FBD/LD/IL* [online]. [cit. 2022-04-08]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_plc\\_intro/135987851.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/135987851.html&id=)
- [8] *Ladderdiagram (LD)* [online]. [cit. 2022-04-08]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplcctrl\\_languages%20ld.htm&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplcctrl_languages%20ld.htm&id=)
- [9] *PLCopenMC V20* [online]. [cit. 2022-04-08]. Dostupné z: [https://plcopen.org/system/files/downloads/plcopen\\_motion\\_control\\_part\\_1\\_version\\_2.0.pdf](https://plcopen.org/system/files/downloads/plcopen_motion_control_part_1_version_2.0.pdf)
- [10] *TC2 Task Force Motion Control: Function blocks for motion control. In: PLCopenMC V20* [online]. PLCopen, c2009-2011, 17.3.2011 [cit. 2022-04-10].

Dostupné z: [plcopen.org/system/files/downloads/plcopen\\_motion\\_control\\_part\\_1\\_version\\_2.0.pdf](http://plcopen.org/system/files/downloads/plcopen_motion_control_part_1_version_2.0.pdf)

[11] *B&R: Produkty*. [online]. [cit. 2022-04-12]. Dostupné z: <http://www.br-automation.com/cs/>

[12] *POWERLINK – Technologie*. [online]. [cit. 2022-04-12]. Dostupné z: <https://www.br-automation.com/cs/produkty/networks-and-fieldbus-modules/additional-information/powerlink-technologie/>

[13] *B&R X20CP0484-1*. [online]. [cit. 2022-04-12]. Dostupné z: <http://img.interempresas.net/FotosArtProductos/P197191.jpg>

[14] *B&R X20CP0484-1*. [online]. [cit. 2022-04-12]. Dostupné z: <http://img.interempresas.net/FotosArtProductos/P197191.jpg>

[15] *8EI4X5MWDS0.XXXX-1*. [online]. [cit. 2022-04-12]. Dostupné z: <https://www.br-automation.com/fileadmin/1447878621203-de-bigicon-1.0.jpg>

[16] *X20(c)CP041x, X20CP0420 and X20CP048x*. [online]. [cit. 2022-04-12]. Dostupné z: [https://download.brautomation.com/BRP44400000000000000665818/X20CP04xx-en\\_V2.14.pdf?px-hash=51255c8c8c6b3ff5c899df84abe6e3dc&px-time=1652338460](https://download.brautomation.com/BRP44400000000000000665818/X20CP04xx-en_V2.14.pdf?px-hash=51255c8c8c6b3ff5c899df84abe6e3dc&px-time=1652338460)

[17] *X20 System*. [online]. [cit. 2022-04-12]. Dostupné z: <https://www.br-automation.com/fileadmin/1331004070910-de-html-1.0.jpg>

[18] *0PS1050.1*. [online]. [cit. 2022-04-13]. Dostupné z: <https://www.br-automation.com/index.php?eID=dumpFile&t=f&f=2%3A%2FBRP44400000000000000666540&token=41dfd8ac6be10b941ab7faf679a511253918724b>

[19] *X20PS9600*. [online]. [cit. 2022-04-12]. Dostupné z: <https://www.br-automation.com/index.php?eID=dumpFile&t=f&f=2%3A%2FBRP44400000000000000665820&token=b609762352a51fe8644fdf92fabf457f2c1e2aa9>

[20] *X20DIF371*. [online]. [cit. 2022-04-12]. Dostupné z: <https://www.br-automation.com/index.php?eID=dumpFile&t=f&f=2%3A%2FBRP44400000000000000680059&token=984f6230ee517d68f8117a7e6842c4485fb795b1>

