

**ZÁPADOČESKÁ UNIVERZITA V PLZNI**  
**FAKULTA STROJNÍ**

**Studijní program:** Průmyslové inženýrství a management  
N0715A270012

**Studijní specializace:** Bez specializace

**DIPLOMOVÁ PRÁCE**

**Návrh zlepšení systému výměny forem s využitím  
průmyslového robota**

**Autor:** Bc. Jan Kasík

**Vedoucí práce:** Ing. Tomáš Broum, PhD.

Akademický rok 2023/2024

## ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta strojní

Akademický rok: 2023/2024

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Jan KASÍK**  
Osobní číslo: **S22N0040P**  
Studijní program: **N0715A270012 Průmyslové inženýrství a management**  
Téma práce: **Návrh zlepšení systému výměny forem s využitím průmyslového robota**  
Zadávací katedra: **Katedra průmyslového inženýrství a managementu**

### Zásady pro vypracování

1. Automatizace
2. Robotizace
3. Analýza současného stavu
4. Návrh řešení
5. Závěr

Rozsah diplomové práce: **50 – 70 stran**  
Rozsah grafických prací: **–**  
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. NOVOTNÝ, František. *Úvod do automatizace a robotizace ve strojírenství*. Liberec: Technická univerzita v Liberci, 2020. ISBN 978-80-7494-545-8.
2. CHLEBNÝ, Jan a kol. *Automatizace a automatizační technika: Prostředky automatizační techniky*. Brno: Computer Press, 2014. ISBN 978-80-251-3747-5.
3. KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. Brno: Vysoké učení technické v Brně, 2016. ISBN 978-80-214-4828-5.

Vedoucí diplomové práce: **Ing. Tomáš Broum, Ph.D.**  
Katedra průmyslového inženýrství a managementu  
Konzultant diplomové práce: **Ing. Aleš Kamen**  
Konplan s.r.o.  
Datum zadání diplomové práce: **16. října 2023**  
Termín odevzdání diplomové práce: **24. května 2024**

L.S.

---

**Doc. Ing. Vladimír Duchek, Ph.D.**  
děkan

---

**Doc. Ing. Michal Šimon, Ph.D.**  
vedoucí katedry

## **Prohlášení o autorství**

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

V Plzni dne: .....

.....

podpis autora

## **Poděkování**

Tímto bych chtěl poděkovat vedoucímu bakalářské práce, Ing. Tomáši Broumovi, Ph. D., za odborné vedení, věnovaný čas, cenné rady, podněty a připomínky při zpracování mé práce. Dále bych také rád poděkoval svému konzultantovi Ing. Aleši Kamenovi za zprostředkování tématu diplomové práce v rámci společnosti Krones AG.

## ANOTAČNÍ LIST DIPLOMOVÉ PRÁCE

<b>AUTOR</b>	<b>Příjmení</b> Kasík	<b>Jméno</b> Jan	
<b>STUDIJNÍ PROGRAM</b>	N0715A270012 Průmyslové inženýrství a management		
<b>VEDOUCÍ PRÁCE</b>	<b>Příjmení (včetně titulů)</b> Ing. Broum, Ph.D.	<b>Jméno</b> Tomáš	
<b>PRACOVISŤE</b>	ZČU – FST – KPV		
<b>DRUH PRÁCE</b>	<b>DIPLOMOVÁ</b>	<del><b>BAKALÁŘSKÁ</b></del>	<b>Nehodící se škrtněte</b>
<b>NÁZEV PRÁCE</b>	Návrh zlepšení systému výměny forem s využitím průmyslového robota		

<b>FAKULTA</b>	strojní	<b>KATEDRA</b>	KPV	<b>ROK ODEVZD.</b>	2024
----------------	---------	----------------	-----	--------------------	------

### POČET STRAN (A4 a ekvivalentů A4)

<b>CELKEM</b>	77	<b>TEXTOVÁ ČÁST</b>	66	<b>GRAFICKÁ ČÁST</b>	10
---------------	----	---------------------	----	----------------------	----

<b>STRUČNÝ POPIS (MAX 10 ŘÁDEK)</b>  <b>ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY</b>	Diplomová práce se zabývá návrhem zlepšení systému výměny forem s využitím průmyslového robota. V teoretické části se práce věnuje pojmům automatizace a robotizace. Důraz je kladem především na PLC a jejich programování. Praktická část práce je zaměřena na popis navrženého zlepšení systému výměny forem od analýzy současného stavu procesu po popis všech implementovaných změn. V závěru se pak práce věnuje přínosům implementovaných změn.
<b>KLÍČOVÁ SLOVA</b>  <b>ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE</b>	Automatizace, robotizace, PLC, HMI, učení, robot

## SUMMARY OF DIPLOMA SHEET

<b>AUTHOR</b>	<b>Surname</b> Kasík	<b>Name</b> Jan
<b>STUDY PROGRAMME</b>	N0715A270012 Industrial engineering and management	
<b>SUPERVISOR</b>	<b>Surname (Inclusive of Degrees)</b> Ing. Broum, Ph.D.	<b>Name</b> Tomáš
<b>INSTITUTION</b>	ZČU – FST – KPV	
<b>TYPE OF WORK</b>	<b>DIPLOMA</b>	<b>BACHELOR</b> Delete when not applicable
<b>TITLE OF THE WORK</b>	Proposal to improve the mould change system using an industrial robot	

<b>FACULTY</b>	Mechanical Engineering	<b>DEPARTMENT</b>	Industrial Engineering and Management	<b>SUBMITTED IN</b>	2024
----------------	------------------------	-------------------	---------------------------------------	---------------------	------

### NUMBER OF PAGES (A4 and eq. A4)

<b>TOTALLY</b>	77	<b>TEXT PART</b>	66	<b>GRAPHICAL PART</b>	10
----------------	----	------------------	----	-----------------------	----

<b>BRIEF DESCRIPTION</b> <b>TOPIC, GOAL, RESULTS AND CONTRIBUTIONS</b>	The thesis deals with the proposal of an improved mould exchange system using an industrial robot. The theoretical section of the thesis deals with the concepts of automation and robotization. The focus is mainly on PLCs and their programming. The practical section of the thesis focuses on the description of the proposed improvement of the mould exchange system from the analysis of the current state of the process to the description of all implemented changes. The thesis concludes with a discussion of the benefits of the implemented changes.
<b>KEY WORDS</b>	Automation, robotics, PLC, HMI, learning, robot

## Obsah

Seznam obrázků.....	10
Seznam tabulek.....	13
Přehled použitých zkratk a symbolů.....	14
Úvod.....	15
1 Automatizace.....	16
1.1 Programovatelné automaty.....	17
1.2 Programování PLC.....	18
1.2.1 Ladder Diagram.....	19
1.2.2 Function Block Diagram.....	19
1.2.3 Sequential Function Chart.....	19
1.2.4 Instruction List.....	20
1.2.5 Structured Text.....	20
1.3 Logické kombinační funkce.....	21
1.4 Základní struktura PLC programu v TIA Portal.....	25
1.5 Human Machine Interface.....	26
2 Robotizace.....	29
2.1 Průmyslové roboty.....	29
2.2 Kolaborativní roboty.....	31
2.3 Přehled výrobců průmyslových robotů.....	32
2.4 Programování robotů.....	35
3 Analýza současného stavu.....	37
3.1 Charakteristika společnosti Krones AG.....	37
3.2 Popis pracoviště.....	38
3.3 Učení robotu.....	42
3.4 Cíl a popis programu.....	43
4 Programování průmyslového robota.....	46
4.1 Setup funkce.....	46
4.2 Funkce PNS0001.....	47
5 PLC programování.....	50
5.1 Deklarace datových typů.....	50
5.2 Inicializace proměnné „ActiveID“.....	52
5.3 Network „Selection possible“.....	52
5.4 Network „Request“.....	53
5.5 Network „Active“.....	54



5.6	Komunikace s HMI .....	59
6	Programování HMI .....	65
6.1	Návrh screenu .....	65
6.2	Screen Master .....	66
6.3	Screen Detail .....	67
7	Přínosy navrženého řešení.....	72
	Závěr .....	74
	Seznam použité literatury .....	75

## Seznam obrázků

Obrázek 1: Wattův odstředivý regulátor [30].	16
Obrázek 2: Modulární PLC od společnosti Allen Bradley [12]	18
Obrázek 3: Kompaktní PLC od společnosti Allen Bradley [13]	18
Obrázek 4: Příklad programu v jazyce LD	19
Obrázek 5: Příklad programu v jazyce FBD	19
Obrázek 6: Příklad programu v jazyce SFC	20
Obrázek 7: Podmínka pro přechod do dalšího stavu programu v jazyce SFC	20
Obrázek 8: Operace „AND“ v programovacím jazyku IL	20
Obrázek 9: Operace „AND“ v programovacím jazyku ST	21
Obrázek 10: Logická kombinační funkce NOT	21
Obrázek 11: Logická kombinační funkce AND	22
Obrázek 12: Logická kombinační funkce OR	22
Obrázek 13: Logická kombinační funkce NAND	23
Obrázek 14: Logická kombinační funkce NOR	23
Obrázek 15: Logická kombinační funkce XOR	23
Obrázek 16: Logická kombinační funkce XNOR	24
Obrázek 17: Funkce MOVE	24
Obrázek 18: Funkce TRIG	25
Obrázek 19: Struktura programu v TIA Portal	26
Obrázek 20: Příklad HMI od společnosti Siemens [36]	27
Obrázek 21: Klávesnice pro textový vstup	27
Obrázek 22: Klávesnice pro číselný vstup	28
Obrázek 23: Sériový manipulátor [16]	30
Obrázek 24: Paralelní manipulátor [17]	30
Obrázek 25: Spolupráce operátora s kobotem značky Universal Robots [32]	32
Obrázek 26: Robot od společnosti FANUC [19]	32
Obrázek 27: Robot od společnosti ABB [21]	33
Obrázek 28: Robot od společnosti KUKA [23]	33
Obrázek 29: Robot od společnosti Yaskawa [26]	34
Obrázek 30: Robot od společnosti Stäubli [27]	34
Obrázek 31: Robot od společnosti Universal Robots [29]	34
Obrázek 32: Teach-pendant od společnosti FANUC [31]	35
Obrázek 33: Mapa poboček společnosti Kronos AG [34]	37
Obrázek 34: Stroj Contiform od společnosti Kronos AG [35]	38

Obrázek 35: Plastové preforma.....	38
Obrázek 36: Forma na vyfouknutí .....	39
Obrázek 37: Obal formy připevněný ke stroji .....	39
Obrázek 38: Robotický manipulátor Fanuc M-710iC/50S nesoucí formu .....	40
Obrázek 39: Snímek pracoviště .....	40
Obrázek 40: Schéma pracoviště.....	41
Obrázek 41: Human Machine Interface na pracovišti .....	41
Obrázek 42: Dotyk špičky cylindru robota se špičkou cylindru daného bodu .....	42
Obrázek 43: Zásobník forem .....	43
Obrázek 44: Vývojový diagram práce .....	44
Obrázek 45: Model pracoviště v SW Roboguide .....	46
Obrázek 46: Rozšíření teach funkce .....	46
Obrázek 47: Funkce PNS0001 – výběr funkce.....	47
Obrázek 48: Funkce PNS0001 – teach funkce .....	48
Obrázek 49: Funkce PNS0001 – funkce pro přepravní polohu .....	48
Obrázek 50: Vstupně výstupní rozhraní .....	49
Obrázek 51: UDT pro jednotlivé setup funkce .....	51
Obrázek 52: Rozšíření stávajícího UDT robota.....	52
Obrázek 53: Hodnoty ID pro jednotlivé funkce .....	52
Obrázek 54: Network pro iniciaci ID proměnné .....	52
Obrázek 55: Network „Selection possible“ pro teach funkci „Magazine P1“ .....	53
Obrázek 56: Network „Selection possible“ pro teach funkci „Blowing station“ .....	53
Obrázek 57: Network „Request“ .....	54
Obrázek 58: Network Active – první část .....	55
Obrázek 59: Označení jednotlivých příkazů.....	55
Obrázek 60: Network Active – druhá část.....	56
Obrázek 61: Network Active – třetí část .....	58
Obrázek 62: UDT pro vstupní signály .....	59
Obrázek 63: UDT pro výstupní signály .....	60
Obrázek 64: Mapování vstupních signálů .....	60
Obrázek 65: Mapování výstupních signálů .....	60
Obrázek 66: Network pro zpracování signálu – první část.....	61
Obrázek 67: Network pro zpracování signálu – druhá část .....	62
Obrázek 68: Network pro zpracování signálu – třetí část.....	62
Obrázek 69: Network pro zpracování signálu pro funkci „Delivery position“ .....	63

Obrázek 70: Přiřazení hodnoty proměnné ID .....	63
Obrázek 71: Layout navrženého screenu .....	65
Obrázek 72: Vytvořený screen.....	66
Obrázek 73: Definování uživatelských textů.....	66
Obrázek 74: Struktura „TabControl“ .....	67
Obrázek 75: Přiřazení administrátorských práv.....	67
Obrázek 76: Návodný obrázek pro polohování robota [37] .....	68
Obrázek 77: Implementace návodných obrázků.....	68
Obrázek 78: Tlačítka pro ovládání robota .....	68
Obrázek 79: Propojení Master a Detail screenu .....	68
Obrázek 80: Definování tlačítek pro ovládání robota.....	69
Obrázek 81: Pomocný text pro bod „Blowing station“ .....	69
Obrázek 82: Tlačítko pro polohování robota do přepravní pozice .....	69
Obrázek 83: Detail screen – bez zvolené funkce .....	70
Obrázek 84: Detail screen – zvolena funkce.....	70
Obrázek 85: Pomocný obrázek pro provedení akce [37].....	71
Obrázek 86: Fotografie celého HMI .....	71

## Seznam tabulek

Tabulka 1: Nabývající hodnoty logické kombinační funkce NOT .....	22
Tabulka 2: Nabývající hodnoty logické kombinační funkce AND .....	22
Tabulka 3: Nabývající hodnoty logické kombinační funkce OR .....	22
Tabulka 4: Nabývající hodnoty logické kombinační funkce NAND .....	23
Tabulka 5: Nabývající hodnoty logické kombinační funkce NOR.....	23
Tabulka 6: Nabývající hodnoty logické kombinační funkce XOR.....	24
Tabulka 7: Nabývající hodnoty logické kombinační funkce XNOR.....	24
Tabulka 8: Překlad uživatelských textů .....	66

## **Přehled použitých zkratk a symbolů**

AG	Aktiengesellschaft
CPU	Central processing unit
FB	Function Block
FBD	Function Block Diagram
FC	Function
FD	Data Block
HMI	Human Machine Interface
I/O	Input/Output
IB	Organization Block
IEC	International Electrotechnical Commission
ID	Identifier
IL	Instruction List
ISO	International Organization for Standardization
LD	Ladder Diagram
PET	Polyethylentereftalát
PLC	Programmable Logic Controller
SFC	Sequential Function Chart
SPS	Speicherprogrammierbare Steuerung
ST	Structured Text
UDT	User Defined Types
P(1-3)	Punkt(1-3)

## Úvod

Hlavními tématy této diplomové jsou automatizace a robotizace, jež jsou klíčovými trendy, které transformují současný průmysl a společnost jako celek. Stále sofistikovanější technologie umožňují nahrazování lidské práce stroji a roboty, což přináší efektivitu, přesnost a rychlost do různých odvětví průmyslu. Tato revoluce však také přináší výzvy, zejména v oblasti zaměstnanosti a nutnosti přeškolení pracovníků. Klíčové bude nalézt rovnováhu mezi výhodami automatizace a zachováním lidské účasti na pracovním trhu, aby byla tato transformace prospěšná pro celou společnost. Teoretická část práce bude rozdělena do dvou základních kapitol, konkrétně Automatizace a Robotizace.

První kapitola je věnována tématu automatizace. Nejdříve budou popsány průmyslové revoluce historie, které jsou nepochybně spojené s pojmem automatizace. Velká část této kapitoly bude věnována programovatelným automatům. Bude představen princip jejich fungování a také základní rozdělení. Poté se kapitola bude věnovat programování těchto automatů se zaměřením na normované programovací jazyky a datové typy. Následovat bude přehled základních logických kombinačních funkcí. Na závěr této kapitoly bude představen pojem HMI.

V druhé kapitole se bude práce zabývat tématem robotizace. Nejprve bude vysvětlen pojem robotizace jako takový. Poté se kapitola zaměří na průmyslové roboty a jejich možná dělení. Samostatná podkapitola bude věnována kolaborativním robotům. Poté bude uveden přehled největších výrobců průmyslových robotů společně s příklady. Na závěr kapitoly bude zmíněno programování robotů.

Praktická část je opět rozdělena do čtyř kapitol. První z těchto kapitol – Analýza současného stavu, má za cíl analyzovat pracoviště, na kterém byla zpracována diplomová práce. Z počátku bude představena společnost Krones AG, ve které byla práce provedena. Následovat bude popis pracoviště, včetně jednotlivých částí řídicího systému. Na závěr této kapitoly bude popsáno učení robota a vytyčen cíl, kterého by tato práce měla dosáhnout.

Zbylé kapitoly slouží jako dokumentace postupu vypracování práce. První část se bude zaměřovat na programování robota, přičemž budou popsány provedená rozšíření jednotlivých funkcí. Další část se bude věnovat samotnému programování PLC, kdy bude popsána jak logika programu, tak i vytvořené datové typy. To vše v souvislosti s programem robota a grafickým rozhraním. Poslední kapitola je věnována programování HMI, kde budou detailně popsány postupy pro vytvoření uživatelského rozhraní, které umožní operátorům snadné monitorování a ovládání celého systému. Na závěr budou zhodnoceny přínosy této práce.

Z důvodu ochrany dat společnosti Krones AG bude tato práce dokumentovat pouze změny, které byly provedeny autorem.

## 1 Automatizace

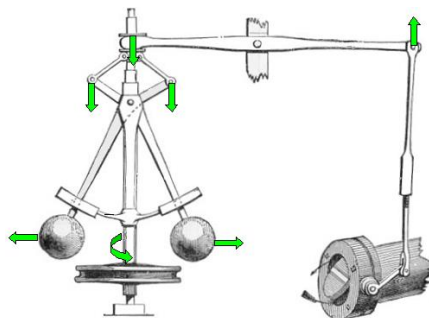
Automatizace je pojem užívaný pro zdokonalování procesů navazující na instrumentaci a mechanizaci. Tyto procesy lze rozdělit do několika úrovní [4]:

1. Instrumentace – proces je opatřen ručními nástroji
2. Mechanizace – fyzická práce člověka je nahrazena strojem
3. Automatizace – stroj je schopný kromě fyzické práce i proces řídit

Automatizace patří k velmi často skloňovaným pojmům nejen v posledních letech. Automatizaci lze chápat jako nástroj sloužící ke zdokonalení procesů, kdy stroj přebírá od člověka kromě samotné fyzické práce i schopnost daný proces řídit. Ve výrobě je cílem dosáhnout vyšší produktivity či lepší jakosti. Mimo to lze díky automatizaci zajistit vyšší bezpečnost na pracovišti a ulehčit výrobním dělníkům od fyzicky namáhavé práce [4].

Důvody, proč podniky přicházejí s podnětem automatizace, jsou různé. Jednak se může jednat o případy, kdy je automatizace vynucená. To jsou situace, kdy daná práce je pro lidského pracovníka zdraví či dokonce životu nebezpečná, například při práci ve velmi vysokých nebo naopak velmi nízkých teplotách. Dále se jedná o případy, kdy člověk není schopen dosáhnout požadované rychlosti nebo přesnosti. Kromě vynucené automatizace se může jednat o automatizaci pramenící z ekonomických důvodů. Může jít například o snížení nákladů na procesy a tím z dlouhodobého hlediska zvýšit obrat za cenu počáteční investice (nákup zařízení). S tím je také spojen nárůst objemu výroby, protože jednak mohou stroje být rychlejší než člověk a jednak pracovat nepřetržitě po celou dobu bez nutnosti přestávky. Dalšími důvody automatizace může být například renomé firmy, jelikož plně automatizovaná výroba bývá pro populaci atraktivní, či prostě zjednodušení dané práce [4].

Automatizace je nepochybně spojena s tzv. průmyslovými revolucemi. Do dnešní doby se uvádějí čtyři průmyslové revoluce. První z nich započala v 18. století v Anglii a je charakteristická termínem industrializace, tedy přechodem od ruční práce v manufakturách k výrobě ve větších továrnách za použití strojů. Díky tomu bylo možné dosáhnout většího objemu výroby. V této době byly stroje poháněny vodní parou. Z tohoto období také pochází jeden z prvních regulátorů, které jsou neodmyslitelnou součástí automatizace. Jednalo se o Wattův odstředivý regulátor pro regulaci otáček parního stroje (viz. Obrázek č. 1 níže). Tento regulátor byl mechanický a fungoval na principu odporové síly tak, že čím vyšší byly otáčky stroje, tím více se zvyšovala odstředivá síla, která rozevírala závaží od středu mechanismu. Toto rozevření bylo přeneseno na vertikální pohyb, kterým pak byl ovládán přívod vodní páry do stroje [5].



Obrázek 1: Wattův odstředivý regulátor [30].



Počátek druhé průmyslové revoluce je datován na konec 19. století a je spojen s elektrifikací a vznikem montážních linek. Elektřina nahradila doposud využívané parní stroje. První montážní linka byla sestavena roku 1870 v jatečním závodě v americkém městě Cincinnati [5].

Počátky automatizace lze spojovat s počátkem třetí průmyslové revoluce, který je datován do roku 1969, kdy byl vynalezen a použit první programovatelný logický automat neboli PLC. Klasické počítače tehdejší doby představovaly velké a často velmi náchylné výpočetní zařízení. Kvůli vysokým požadavkům na provozní teplotu, čistotu a mnoho dalších, bylo nutné ukládat počítače ve speciálních, k tomuto účelu určených místnostech. PLC byly o mnoho menší programovatelné počítače, které byly schopné vydržet v prostorách výroby. PLC jsou dodnes velmi často využívané, jak bude podrobněji rozebráno v následující kapitole [5].

Koncept poslední, a tedy čtvrté průmyslové revoluce neboli Průmysl 4.0 byl poprvé popsán v roce 2011 a probíhá až do dnešních dní. Průmysl 4.0 se zaměřuje na automatizaci, digitalizaci a robotizaci. Populární pojmy, tzv. „buzzwords“, jsou například „Internet věcí“ (IoT), kdy je snaha propojit různá zařízení přes internet a umožnit jim výměnu dat. Dále třeba „Big Data“, kdy v současné době je dispozici tolik dat, že je třeba neustále rozvíjet způsoby, jak tato data zpracovávat a používat. Nepochybně je také možné zmínit rozšířenou a virtuální realitu či digitální dvojče. Samostatnou kategorií by mohla být umělá inteligence, která postupně proniká i do každodenního života [5].

Z důvodu zaměření praktické části následné diplomové práce bude celá tato kapitola zaměřena zejména na programovatelné logické automaty (které hrají zásadní roli v oblasti automatizace), jejich základní funkcionalitu a programování, včetně představení základních logických kombinačních funkcí. Dále pak bude představen pojem HMI.

## 1.1 Programovatelné automaty

Programovatelné automaty, anglicky Programmable Logic Controller (PLC), německy Speicherprogrammierbare Steuerung (SPS) jsou do dnešní doby jedny z nejpoužívanějších prostředků automatického řízení. Jedná se o elektronická zařízení navržena pro řízení a monitorování různých průmyslových procesů a zařízení v reálném čase. Jejich hlavní funkcí je provádět logické operace a na základě vstupních signálů generovat odpovídající výstupy, které řídí různá zařízení a mechanismy. PLC byly revolučním řešením v oblasti automatického řízení, jelikož jejich programovatelnost umožňovala rychlé změny v procesu bez fyzické přestavby zařízení. Tím se zvýšila efektivita a flexibilita průmyslových aplikací a umožnilo se rychlé reagování na změněné podmínky. [4]

Dle [3] a [4] je možné PLC rozdělit na dva základní druhy:

1. Modulární PLC – tyto PLC jsou specifické tím, že je možné je konfigurovat pomocí tzv. modulů. Základem je samotný procesor (CPU) a napájecí zdroj. Další „funkce“ lze dodat připojením přídatných modulů. Může se jednat o modul umožňující I/O (Input/Output), tedy vstupně výstupní operace. Mezi další, velmi často používané moduly patří moduly pro komunikaci přes síť. Moduly nemusí pouze „přidávat“ funkcionalitu, ale mohou také zlepšit současné vlastnosti PLC. Příkladem může být modul pro zvětšení adresovatelné paměti a další. Z konstrukčního hlediska je modulární PLC řešeno pomocí konstrukčního rámu, na který lze připojit jednotlivé moduly. Příklad modulárního PLC od společnosti Allen Bradley je možné vidět na Obrázku č. 2 níže.



Obrázek 2: Modulární PLC od společnosti Allen Bradley [12]

2. Kompaktní PLC – na rozdíl od modulárních PLC jsou kompaktní PLC pevně sestavené s danými integrovanými komponenty. Oproti předchozímu typu jsou tyto PLC menší za cenu nižšího výkonu. Slouží tedy primárně pro řízení menších a jednodušších systémů. Avšak cenově jsou dostupnější než modulární PLC. Příklad kompaktního PLC od společnosti Allen Bradley je možné vidět na Obrázku č. 3 níže.



Obrázek 3: Kompaktní PLC od společnosti Allen Bradley [13]

Z hlediska zpracování programu lze říci, že každé PLC funguje v cyklech, tzv. scanech. V každé iteraci tohoto cyklu probíhají čtyři základní činnosti [3]:

1. Ze vstupního zařízení jsou čteny aktuální vstupní hodnoty, které jsou následně uloženy do paměti
2. Vyhodnocení programů, kdy vstupní signály projdou naprogramovanou logikou, a dle této logiky dojde k nastavení výstupních hodnot
3. Po vyhodnocení všech částí programu a uložení výstupních proměnných do paměti, jsou tyto hodnoty předány do výstupních modulů, které poté aktivují akční členy napojené na dané PLC
4. Posledním krokem je příprava pro další scan. Například se vynuluje registr doby současného scanu

Zcela zásadním krokem při implementaci PLC je jejich správné naprogramování, o kterém se lze dočíst více v následující podkapitole.

## 1.2 Programování PLC

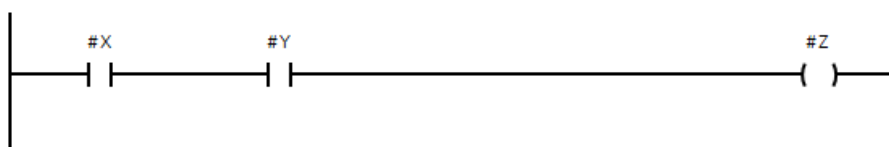
Způsob programování PLC vychází z jejich historie. Původní logicky řídicí systémy byly pevně naprogramovány, což znamenalo, že pro změnu programu bylo třeba řídicí systém manuálně přepojit. Tento proces zajišťovali pracovníci nijak odborně znalí počítačového programování. Při programování PLC, jakožto nástupců těchto zařízení, bylo tak snahou

programovací jazyky co nejvíce připodobnit původnímu způsobu programování. Přechod na používání PLC byl tak plynulejší [4].

Jeden z nejstarších a dodnes nepoužívanějších jazyků programovacích „Ladder diagram“ (podrobnější popis viz níže) připomíná reléové schéma, které se může „virtuálně“ přepojit a tím změnit program. Z počátku se programovací jazyky přizpůsobovaly jednotlivým zařízením od různých výrobců, čímž vzniklo nepřehledné množství těchto jazyků. Proto vznikla iniciativa pro zavedení normy, která by zajistila standardizaci. Dodnes platnou je norma IEC/EN 61131-3 (dále jen „norma“) spravována organizací PLC Open. Dle normy je platných 5 programovacích jazyků, které jsou jak grafické (tři z těchto) tak textové (zbylé dva) [3,4]. Dále bude následovat stručný přehled těchto pěti programovacích jazyků, kdy u každého bude uveden příklad naprogramování kombinační funkce „AND“ (tato i jiné základní kombinační funkce budou popsány v následující podkapitole). Všechny příklady byly vytvořeny v softwaru TIA Portal od společnosti Siemens. Při zpracování v softwarech od jiných výrobců je možné nalézt obdobné grafické zpracování s drobnými odlišnostmi, avšak funkcionality zůstává stejná.

### 1.2.1 Ladder Diagram

Ladder Diagram (LD, česky „jazyk příčkového diagramu“) – jedná se, jak už bylo zmíněno výše, o jazyk, jenž má připomínat schéma zapojení relé v elektrotechnice. Základním schématem jsou vždy dvě svislé čáry na levé a pravé straně PLC obvodu, které reprezentují sběrnice. Program se pak tvoří zleva doprava, kdy se přidávají jednotlivé kontakty reprezentující například signály ze senzorů či hodnoty pro akční členy (motor, ventil, apod.) Příklad programu v jazyce LD je vidět na Obrázku č. 4 níže. Tento jazyk je jeden z nejvíce používaných jazyků díky své intuitivnosti. V rámci LD jazyka lze mnohdy vidět i bloky, a tedy je možné si toto splést s jazykem FBD (popsán níže). Jedná se však pouze o volání (tzv. „block call“), kdy spuštěním bloku je zavolán jiný příčkový diagram, což umožňuje tvorbu komplexnějších programů při zajištění adekvátní přehlednosti [3, 4].



Obrázek 4: Příklad programu v jazyce LD

### 1.2.2 Function Block Diagram

Function Block Diagram (FBD, česky „jazyk funkčního blokového schématu“) – program napsaný v tomto grafickém jazyce je podobný programu v jazyce LD, ovšem s tím rozdílem, že logické operace se vykonávají v blocích odpovídajících funkci. Tyto bloky jsou navzájem propojené, tedy výstup z jednoho bloku může být vstupem druhého bloku, čímž je možné vytvořit požadovanou logiku. Příklad programu v jazyce FBD je vidět na Obrázku č. 5 níže [3, 4].

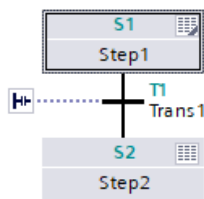


Obrázek 5: Příklad programu v jazyce FBD

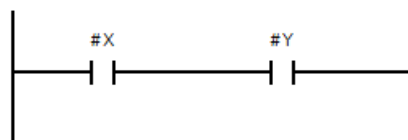
### 1.2.3 Sequential Function Chart

Sequential Function Chart (SFC, česky „jazyk sekvenčního blokového schématu“) – jak již název napovídá, je tento poslední grafický jazyk podobný jazyku předchozímu. Využití však

nalézá zejména v cyklických operacích, kdy se neustále opakují stejné stavy, reprezentovány jednotlivými bloky. Po splnění konkrétní podmínky přejde systém do dalšího stavu, kde setrvá do splnění další podmínky, a tak dále. Příkladem by mohlo být řízení semaforu na křižovatce, kdy každý stav reprezentuje nastavení jednotlivých barev na semaforech. Podmínkou pro přechod do dalšího stavu by bylo uplynutí daného času či signál ze senzoru, kterým je semafor osazen (pro detekci počtu aut v koloně před křižovatkou). Z důvodu soudržnosti textu bude představen program odpovídající logické operaci „AND“ jako u ostatních jazyků, ač v jazyce sekvenčního schématu nemá praktický smysl. Na Obrázku č. 6 je vidět celé schéma tvořené dvěma stavy (inicializační stav 1 a stav druhý, kdy dochází nastavení hodnoty bitu „Z“ na jedna), do kterého systém přejde po splnění podmínky uvedené na Obrázku č. 7 [3, 4].



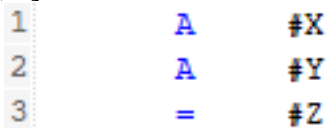
Obrázek 6: Příklad programu v jazyce SFC



Obrázek 7: Podmínka pro přechod do dalšího stavu programu v jazyce SFC

#### 1.2.4 Instruction List

Instruction List (IL, česky „jazyk seznamu instrukcí“) – jde o první ze dvou textových jazyků. Programy se skládají z předem definovaných základních instrukcí, tzv. mnemokódů. Svoji strukturou připomíná nižší klasické programovací jazyky, jako například Assembler. Pro každý řádek kódu je jedna instrukce. Na Obrázku č. 8 je vidět sestavení logické operace „AND“, kdy pro tuto operaci existuje instrukce „A“. Zde je možná dobré upozornit na způsob vyhodnocení operace. Program se vyhodnocuje po řádcích, tedy ve chvíli, kdy je čítač instrukcí na řádku jedna, bude hodnota rovna jedné v případě, že hodnota „X“ bude rovna jedné. Na řádku druhém se již však vyhodnocuje jak řádek dva, tak i řádek jedna (instrukce „A“). Tedy zde pro zajištění výsledku rovnému jedné je třeba, aby byly hodnoty „X“ i „Y“ rovné jedné, tedy jako u klasické operace „AND“ [3, 4].



Obrázek 8: Operace „AND“ v programovacím jazyku IL

#### 1.2.5 Structured Text

Structured Text (ST, česky „jazyk strukturovaného textu“) – jedná se o druhý textový jazyk. Tento jazyk je objektově orientovaný a tedy podobný dnes již klasickým programovacím jazykům jako jsou například C++, Python nebo Java. Jeho výhodou je právě tato podobnost a lze zde aplikovat obecné principy programování. Další výhodou je také snazší sestavování komplexnějších operací na rozdíl od například jazyku LD. Příklad operace „AND“ je na Obrázku č. 9 níže [3, 4].

```

4 IF #X AND #Y THEN
5     #Z := TRUE;
6 END_IF;

```

Obrázek 9: Operace „AND“ v programovacím jazyku ST

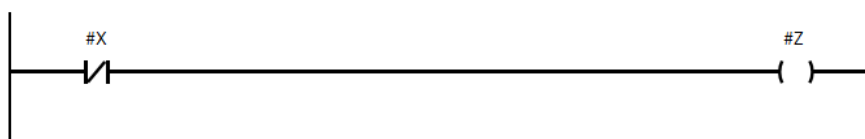
Kromě programovacích jazyků definuje norma mimo jiné i elementární datové typy, které by měl mít každý software od výrobce dodávajícího tuto normu. Nejzákladnějším datovým typem je BOOL, který může nabývat bitových hodnot. Pro řetězce těchto proměnných pak existují BYTE s 8 bity, WORD se 16 bity, DWORD s 32 bity a LWORD se 64 bity. U celočíselných hodnot lze vybrat ze dvou variant, pro kladná i záporná čísla, či pouze čísla nezáporná. S ohledem na předchozí větu tak norma definuje proměnné typu SINT (8 bitů), která obsahuje čísla od -128 do 127 včetně nuly, anebo USINT (opět 8 bitů), která nabývá hodnot od 0 do 255. Pokud by tyto rozsahy nestačily, existují pak typy INT (UINT) s 16 bity a DINT (UDINT) s 32 bity. Dále pak existují čísla s plovoucí řádkou (tzv. „float“), pro která jsou určeny datové typy REAL s 32 bity a LREAL se 64 bity. Dále norma stanovuje typ STRING pro znakové řetězce. Pro časové hodnoty pak dle normy existují typy TIME, DATE, DATE\_OF\_TIME a DATE\_AND\_TIME [4].

V praktické části diplomové práce se bude pracovat s jazykem Ladder diagram. Při programování průchodu signálů se v PLC využívají logické kombinační funkce. Proto jim bude věnována následující podkapitola.

### 1.3 Logické kombinační funkce

Základem každé logiky tvořené v PLC jsou logické kombinační funkce. Ty operují s proměnnými, které mohou nabývat dvou hodnot. Jedná se tedy o proměnnou typu boolean (bool) a hodnoty mohou být pouze nula a jedna, které lze také označit jako „True“ nebo „False“ (pravda nebo nepravda). V praxi toto může reprezentovat například ventil, kdy hodnota nula značí, že je ventil zavřený a hodnota jedna značí ventil otevřený. Při programování složitějších funkcí se využívá sekvence podmínek, které vychází z tzv. výrokové logiky. Proto je vhodné v této kapitole stručně představit základní kombinační logické funkce, pomocí kterých lze sestavit libovolné sekvence podmínek [4]. Pro každou elementární funkci bude sestaven PLC obvod v jazyce Ladder diagram, tedy tzv. network, a také uvedena tabulka nabývajících hodnot. V příkladech budou užity názvy proměnných „X“, „Y“ a „Z“, kdy „X“ a „Y“ jsou vstupy do funkce a „Z“ je výsledek logické operace. Výjimkou je první funkce (NOT), která má pouze jeden vstup a to „X“. Na závěr budou uvedeny ještě funkce používané v praktické části diplomové práce, aby bylo poté zřejmé jejich použití. Opět je nutné podotknout, že logické obvody byly vytvořeny v programu TIA Portal od společnosti Siemens. Nyní bude následovat výčet základních logických kombinačních funkcí a v závěru budou také okomentovány další elementární funkce použité v praktické části diplomové práce.

1. NOT – negace (viz. Obrázek č. 10 a tabulka hodnot č. 1)

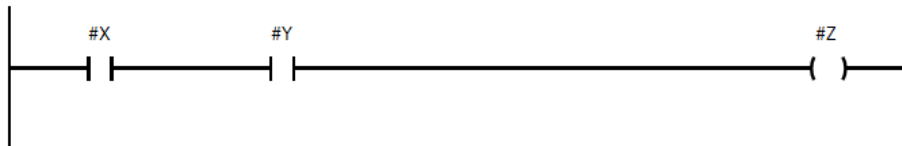


Obrázek 10: Logická kombinační funkce NOT

Tabulka 1: Nabývající hodnoty logické kombinační funkce NOT

X	Z
0	1
1	0

2. AND – logický součin, konjunkce (viz. Obrázek č. 11 a tabulka hodnot č. 2)

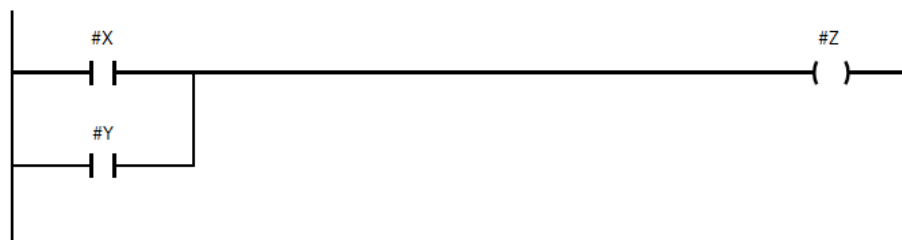


Obrázek 11: Logická kombinační funkce AND

Tabulka 2: Nabývající hodnoty logické kombinační funkce AND

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

3. OR – logický součet, disjunkce (viz. Obrázek č. 12 a tabulka hodnot č. 3)

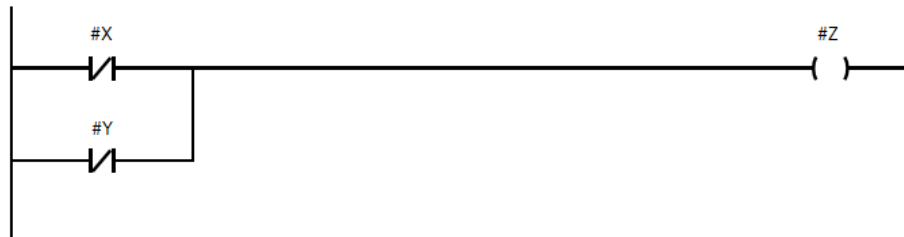


Obrázek 12: Logická kombinační funkce OR

Tabulka 3: Nabývající hodnoty logické kombinační funkce OR

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

4. NAND – negovaný logický součin, Shefferova funkce (viz. Obrázek č. 13 a tabulka hodnot č. 4)

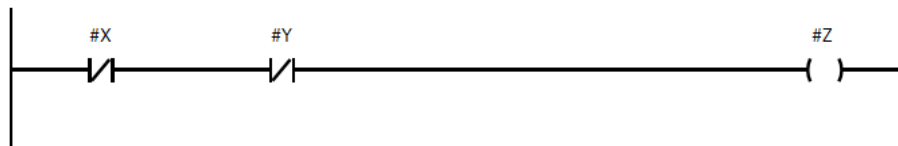


Obrázek 13: Logická kombinační funkce NAND

Tabulka 4: Nabývající hodnoty logické kombinační funkce NAND

X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

5. NOR – negovaný logický součet, Pierceova funkce (viz. Obrázek č. 14 a tabulka hodnot č. 5)

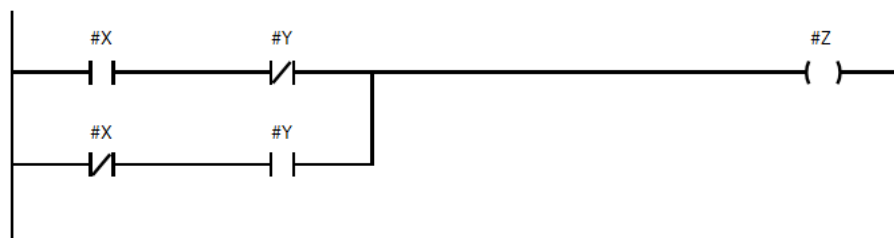


Obrázek 14: Logická kombinační funkce NOR

Tabulka 5: Nabývající hodnoty logické kombinační funkce NOR

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

6. XOR – neshoda (viz. Obrázek č. 15 a tabulka hodnot č. 6)

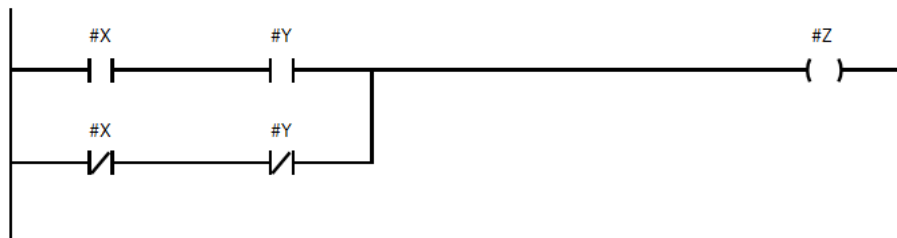


Obrázek 15: Logická kombinační funkce XOR

Tabulka 6: Nabývající hodnoty logické kombinační funkce XOR

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

7. XNOR – shoda, ekvivalence (viz. Obrázek č. 16 a tabulka hodnot č. 7)

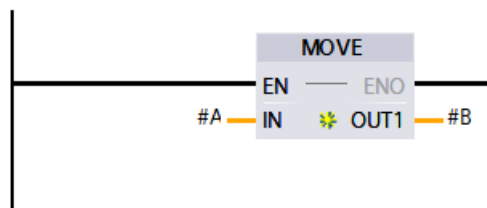


Obrázek 16: Logická kombinační funkce XNOR

Tabulka 7: Nabývající hodnoty logické kombinační funkce XNOR

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

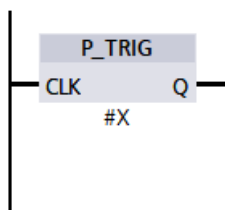
8. MOVE – tato funkce slouží k uložení hodnoty (či proměnné) do jiné proměnné. Na obrázku dojde uložení hodnoty v proměnné A do proměnné B. Je také samozřejmě nutné podotknout, že obě proměnné musí být stejného datového typu a nemohou být typu boolean. Funkci MOVE je možné vidět na Obrázku č. 17 níže.



Obrázek 17: Funkce MOVE

9. TRIG – funkce trigger sleduje hodnotu signálu uloženého v proměnné „X“ (na Obrázku č. 18 uvedeném níže). Konkrétně sleduje jeho náběhovou hranu. Signál tedy bude propuštěn pouze v situaci, kdy se hodnota proměnné „X“ změní z nuly na jedna. Ovšem signál bude propuštěn pouze v cyklu, kdy k této změně dojde. To znamená, že pokud v předchozím cyklu již hodnota „X“ byla rovna jedné, signál nebude propuštěn znovu. Funkci TRIG je možné vidět na Obrázku č. 18 níže.





Obrázek 18: Funkce TRIG

Diplomová práce se ve své praktické části bude věnovat PLC programování za využití programu TIA Portal, proto bude v následující podkapitole přiblížena základní struktura programování PLC.

## 1.4 Základní struktura PLC programu v TIA Portal

Ačkoliv je každý PLC program jiný, všechny programy spojuje společná základní struktura, ze které se skládají. Zaprvé je potřeba v softwaru TIA Portal nastavit hardwarovou konfiguraci tak, aby odpovídala skutečnému stavu PLC, které pak bude daný systém řídit. S ohledem na to, že software TIA Portal je vyvíjen společností Siemens, která je i mimo jiné výrobcem programovatelných automatů, lze v softwaru vybrat konkrétní model PLC i jej osadit doplňkovými moduly tak, jako je skutečný automat fyzicky složen. Kromě konfigurování hardwaru jsou nedílnou součástí programování tři základní prvky (Programové bloky, Tagy a Uživatelsky definované typy), kterou budou nyní představeny.

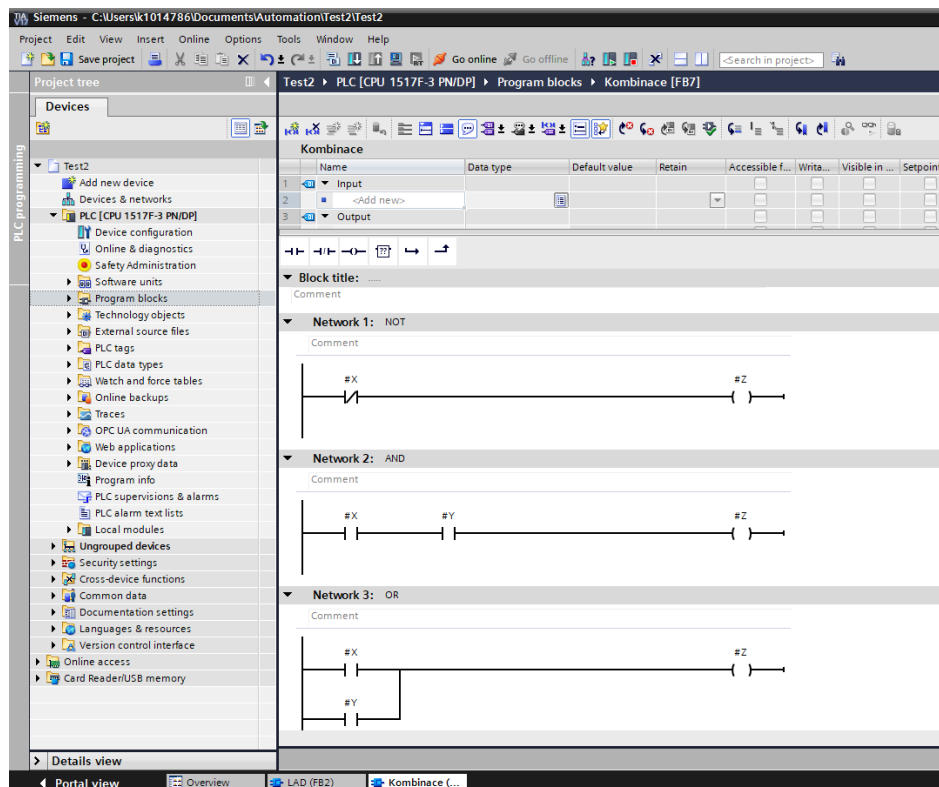
Programové bloky jsou stavební kameny každého PLC programu. Obsahují veškerou logiku a správu dat dané aplikace. V SW TIA Portal operuje programátor se čtyřmi základními druhy bloků [14]:

1. Organization Block (OB, Organizační blok) – organizační bloky jsou speciální programové bloky sloužící pro správu specifických úkolů a událostí. Nejdůležitějším je OB1 neboli Main, který nutně musí obsahovat každý program, jelikož celý program řídí. Po spuštění programu dojde k zavolání bloku OB1, který zajišťuje volání dalších funkcí. „Život“ bloku Main je spojen s druhou fází cyklu PLC (scanu).
2. Function (FC, Funkce) – Funkce je základním blokem pro programování logiky programu. Funkce dokáže přistupovat ke globálním proměnným a zároveň také pracovat s dočasnými (temporary) proměnnými, které „ztrácí“ svoji hodnotu po ukončení scanu.
3. Function Block (FB, Funkční blok) – Funkční bloky slouží ke stejnému účelu jako Funkce, avšak narozdíl od ní může mít přiřazenou svojí lokální paměť (DB), ve které je možné ukládat i tzv. statické proměnné, které s koncem cyklu neztrácejí svoji hodnotu. Je tak možné, a mnohdy i nutné tohoto využít pro nastavení fyzických vstupů a výstupů. Po skončení cyklu by v případě použití Funkce došlo k situaci, kdy fyzický výstup nemá přiřazenou žádnou hodnotu.
4. Data Block (DB, Datový blok) – Datové bloky jsou, jak z názvu vyplívá, určeny pro ukládání dat (vstupů, výstupů, a i interních proměnných). Obsahují jak proměnné, tak i jejich hodnoty.

Dalším základním prvkem PLC programu jsou takzvané TAGs (Tagy). Tagy jsou proměnné, které reprezentují fyzické vstupy (I), výstupy (O) a paměťové adresy (M). Tagy poskytují uživatelsky přívětivý a organizovaný způsob práce s daty, což zlepšuje efektivitu a udržitelnost programování PLC. Programátor pracuje se symbolickými jmény namísto přímých adres paměti. Tagy dokážou pracovat se základními datovými typy (boolean, integer, double).

Posledním základním prvkem jsou tzv. UDT (User Defined Types), neboli uživatelsky definované datové typy, které je nutné využít, pokud je třeba použít složitější datovou strukturu. UDT umožňují programátorovi vytvářet složitější struktury tvořené z proměnných základních datových typů. Tyto struktury dovolují opět jednodušší práci s proměnnými, jelikož lze seskládat k sobě patřící proměnné do jednoho většího celku a poté se lze odkazovat na tyto proměnné pomocí tečkové notace přes tuto strukturu [14].

Příklad struktury PLC programu a ukázka programu TIA Portal je vidět na Obrázku č. 19 níže.



Obrázek 19: Struktura programu v TIA Portal

Pro komunikaci s PLC je důležitou součástí řídicího systému také HMI, které bude představeno v následující podkapitole.

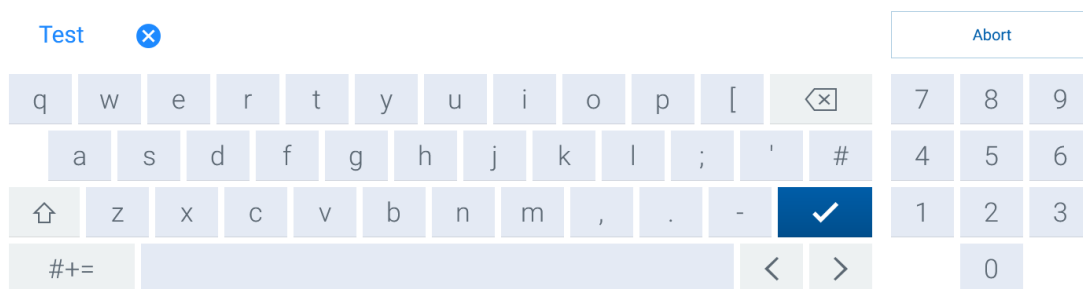
## 1.5 Human Machine Interface

Human Machine Interface (dále HMI), neboli rozhraní mezi člověkem a strojem, je část řídicího systému, který usnadňuje komunikaci mezi strojem a zaměstnancem, který stroj ovládá. Jedná se o grafický prvek řídicího systému, jehož cílem je umožnit operátorovi daný stroj ovládat a monitorovat, aniž by musel mít znalosti programování (ať už samotného výkonného článku řídicího systému – PLC, či samotného grafického rozhraní – HMI). Zjednodušeně lze říci, že se jedná o zařízení připomínající přenosný počítač, či častěji dotykový tablet, se kterým lze interagovat [7, 10]. Příklad provedení HMI společnosti Siemens lze vidět na Obrázku č. 20 níže.

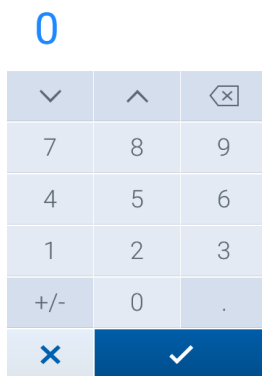


Obrázek 20: Příklad HMI od společnosti Siemens [36]

Samotný displej (screen) je složen z potřebných struktur tak, aby poskytoval potřebnou funkcionalitu. Jednou z klíčových částí jsou ovládací prvky. Pomocí ovládacích prvků může člověk stroj ovládat. Jedním z nejjednodušších prvků jsou tlačítka, která může operátor stisknout a tím provést nějakou akci (například zapnout stroj). Dalšími prvky jsou například tzv. „checkbox“ a „radio button“. Tyto slouží pro výběr, kdy u „checkboxů“ je možné zaškrtnout více možností najednou, zatímco u „radio button“ lze vybrat jen jednu možnost. Program je také možné ovládat pomocí „comboboxu“, který po rozbalení zobrazí dostupné možnosti. Dále je samozřejmě možné vkládat různé vstupy, ať už hodnoty číselné či slovní. K tomu se standardně využívá tzv. on-screen klávesnice, tedy dotyková klávesnice, která se zobrazí při zvolení možnosti pro vložení vstupu. Klávesnice pro textový vstup je vidět na Obrázku č. 21 a klávesnice pro číselný text na Obrázku č. 22 níže.



Obrázek 21: Klávesnice pro textový vstup



Obrázek 22: Klávesnice pro číselný vstup

Jednou z nejčastějších automatizačních technik je využití robotizace, tedy procesu nasazení robotů do dílčích procesů výroby, logistiky apod. Druhou hlavní částí této práce je proto kapitola věnující se pojmu robotizace, následně konkrétním typům robotů a příkladům jejich výrobců.

## 2 Robotizace

Robotika je inovativní a multidisciplinární obor, jenž se zabývá vývojem, konstrukcí, programováním. S pojmem robotika úzce souvisí pojem robotizace, který se zaměřuje spíše na praktické nasazení a využití robotů v procesu automatizace. Oba tyto obory hrají důležitou roli v moderních technologiích a mají široké uplatnění v různých odvětvích lidské činnosti. Robotizace je stále více spojována s pokrokem v oblasti umělé inteligence a strojového učení, což umožňuje robotům se adaptovat na různé situace a plnit složitější úkoly. Roboty lze najít nejen v průmyslu, ale i v medicíně či výzkumu. Jsou navrhovány tak, aby zvýšily efektivitu výroby, minimalizovala se rizika pro lidské pracovníky, nebo dokonce zachraňovaly životy při medicínských zákrocích (příkladem může být chirurgický robot Da Vinci). V současné době se robotika stává stále důležitější součástí také každodenního života. Autonomní vozidla, drony, ba dokonce roboty v domácnostech jsou jen některými příklady, jak robotizace ovlivňuje současnou společnost. Není překvapením, že s narůstajícím využíváním robotů vyvstává řada otázek, včetně etických a právních aspektů [2].

Cílem této práce je představení oblasti průmyslových robotů. Jiné oblasti robotizace jako například využití humanoidních robotů, androidů a dalších nebudou dále rozvedeny.

### 2.1 Průmyslové roboty

Jak již bylo uvedeno v úvodu kapitoly výše, v současné době existuje nepřeberné množství robotů. Tato práce se zaměří na průmyslové roboty, tedy na roboty působící jak ve výrobě, tak logistice nezávisle na odvětví daného průmyslu. Na tyto roboty lze pohlížet jako na manipulační zařízení, jelikož jejich úkolem jsou tzv. pick and place aplikace. Manipulátor má za úkol na jednom místě předmět uchopit a tento předmět na jiné místo v požadovaném čase a přesnosti umístit.

[1] rozděluje tyto roboty na jednoúčelové a univerzální manipulátory. Jednoúčelové manipulátory jsou jak konstrukčně, tak i programově sestaveny tak, aby co nejefektivněji vykonávaly danou, předem určenou činnost. Pro tuto činnost pak zpravidla dosahují větší přesnosti i rychlosti, avšak jsou vázané na konkrétní aplikaci. Zde přicházejí na řadu právě univerzální manipulátory, které nejsou přizpůsobeny žádné konkrétní aplikaci, nýbrž je lze, jak z názvu vyplývá, využít všestranně. Z hlediska programování, respektive ovládání, lze roboty rozdělit na synchronní a programovatelné. Synchronní roboty jsou ovládány teleoperátorem, tedy pracovníkem, který robot na dálku ovládá. To může být vhodné v situacích, kdy je třeba lidského řízení, avšak člověk sám operaci vykonávat nemůže (například z důvodu bezpečnosti). Dle míry autonomie a „schopností“ lze dle [1] rozdělit programovatelné roboty do tří hlavních generací:

1. Průmyslové roboty 1. generace – jedná se o roboty, kdy robot vykonává pevně naprogramované aplikace. Typickou úlohou pro tuto generaci je již výše zmíněná pick and place úloha.
2. Průmyslové roboty 2. generace – tyto roboty jsou již osazeny jak vnějšími senzory například pro strojové vidění (tedy například kamerou), tak vnitřními senzory pro detekci (například tlaku). Díky tomu mohou být využity i pro složitější operace.
3. Průmyslové roboty 3. generace – zde se jedná již o inteligentní roboty, tedy zařízení s využitím umělé inteligence. Cílem těchto robotů není již pouhé nahrazení fyzické práce člověka, ale i řízení, respektive rozhodování. Charakteristickým rysem jsou pak i kognitivní vlastnosti, tedy vnímání vlastního okolí a spolu se strojovým učením adaptace na nečekané události či dynamický provoz.

Kromě těchto tří generací ještě publikace [1] uvádí tzv. „mezigenerace“, které tvoří přechod mezi třemi hlavními generacemi. Sám autor v publikaci [1] zmiňuje, že rozdělení na generace je velmi variabilní a lze takovýchto podobných rozdělení najít mnoho.

Jiným způsobem rozdělení manipulátorů může být na základě jejich konstrukce. Z tohoto hlediska lze definovat dva druhy manipulátorů [2, 1, 8]:

1. Sériové manipulátory – tyto jsou zkonstruovány jako tzv. otevřený kinematický řetězec. Každé rameno kromě základny a koncového efektory navazuje na právě jedno rameno a stejně tak další rameno navazuje na jeho opačný konec. Klíčovou výhodou sériových manipulátorů je jejich dosah, který umožňuje právě jejich konstrukce, která je zároveň i jednodušší. Příklad sériového robota společnosti FANUC je možné vidět na Obrázku č. 23 níže.



Obrázek 23: Sériový manipulátor [16]

2. Paralelní manipulátory – na rozdíl od předchozího typu jsou paralelní manipulátory konstruovány jako uzavřený kinematický řetězec. To znamená, že základna je spojena s koncovým efektem pomocí vícero otevřených kinematických řetězců. Jejich výhodami oproti sériovému typu je nepochybně přesnost, jelikož se chyba z jednotlivých řetězců průměruje, na rozdíl od sériových robotů, kde se chyba přes všechny členy řetězce sčítá. Další výhodou je, že paralelní roboty nemusí být tak „mohutný“, protože hmotnost jak robota, tak neseného břemene, je rozložena do několika ramen. Příklad paralelního robota společnosti FANUC je vidět na Obrázku č. 24 níže.



Obrázek 24: Paralelní manipulátor [17]

Z hlediska kinematiky pak ještě existuje zásadní rozdíl v řešení kinematických úloh. V zásadě existují dvě kinematické úlohy, a to přímá a inverzní. U přímé kinematiky je cílem vypočítat polohu a orientaci koncového efektoru v prostoru na základě znalosti polohy kloubů robota. Naopak úkolem inverzní kinematické úlohy je vypočítat polohu jednotlivých kloubů na základě souřadnic koncového efektoru. Rozdílem v řešení těchto úloh je, že u sériových robotů existuje vždy analytické řešení přímé úlohy, zatímco u paralelních manipulátorů existuje analytické řešení inverzní kinematické úlohy [8].

Trendem současné doby v průmyslu v rámci robotizace je kolaborativní robotika, které bude proto věnována následující podkapitola.

## 2.2 Kolaborativní roboty

Předchozí podkapitola byla zaměřena na průmyslové roboty. Ty dle normy ISO 10218 musí být odděleny mechanickými zábranami (například umístěny v ocelové kleci) nebo optickými senzory, které detekují vyjetí robota z jeho povolené oblasti. V tomto ohledu jsou opakem kolaborativní roboty, tzv. koboty (anglicky cobots). Tyto, nejen že nemusí být odděleny od operátora, ale je možné, jak již jejich název napovídá, s operátorem kooperovat. Proto koboty představují nové možnosti automatizace procesů [2, 6, 11].

Klíčovou otázkou při uvažování o kolaborativních robotech je bezpečnost pracovníků, kteří budou s kobotem interagovat. Na bezpečnost je dbán zřetel již ve fázi konstruování. Koboty jsou typické svou lehčí konstrukcí než průmyslové roboty, což umožňuje rychlejší zastavení robota v případě kolize. To ovšem s sebou přináší jednu z klíčových nevýhod, konkrétně nižší nosnost oproti robotům. Zároveň koboty bývají konstruovány s oblými hranami a jsou potaženy měkčím materiálem, aby se snížily možné následky kolize s člověkem. Dalším typickým znakem je snížená maximální rychlost, která je stanovena na 250 milimetrů za sekundu, která umožňuje zastavení robota v minimálním čase. Právě tato maximální rychlost je jedna z největších nevýhod kobotů, protože oproti standartním průmyslovým robotům nejsou tak efektivní a mnohdy kvůli tomu nemohou být nasazeni, protože by nestíhaly takt výroby [2, 6, 11].

Přesto koboty najdou své uplatnění. Typickými aplikacemi jsou například manipulace s materiálem, kdy kobot přenáší materiál operátorovi, který dále s daným materiálem pracuje (například montáž). To z dlouhodobého hlediska může ulehčit práci zaměstnancům a vyvarovat se nemocem z povolání z dlouhodobého a pravidelného zatížení. Další z mnoha aplikací je situace, kdy kobot kooperuje s operátorem při montáži. Kobot s operátorem paralelně vykonávají jednotlivé kroky montáže, čímž se celý proces urychlí. Koboty tedy nalézají mnohá uplatnění a pokud to podmínky umožňují, mohou být užitečným nástrojem pro automatizování procesů se zachováním lidské pracovní síly [2, 6, 11].

Příklad spolupráce operátora s kobotem značky Universal Robots je možné vidět na Obrázku č. 25 níže.



Obrázek 25: Spolupráce operátora s kobotem značky Universal Robots [32]

V případě, že se společnost rozhodne pro robotizaci daného výrobního nebo logistického procesu, je třeba zvolit vhodný typ a výrobce robota. Proto bude následující podkapitola věnována přehledu vybraných výrobců průmyslových robotů.

### 2.3 Přehled výrobců průmyslových robotů

Jak již bylo v této kapitole zmíněno, robotizace je jedním z aktuálních trendů, proto existuje mnoho společností, které průmyslové (a kolaborativní) roboty vyrábějí. Každá společnost se specializuje na určité typy robotů a aplikací. Výběr výrobce závisí tak na konkrétních potřebách a specifikacích konkrétní průmyslové aplikace. Důležité je také mít na paměti inovace a nové technologie, které výrobci neustále přinášejí na trh, a které mohou mít významný vliv na budoucí výrobu a automatizační procesy. Proto budou do následujícího přehledu vybráni více významní výrobci průmyslových robotů:

1. FANUC – Jedná se o japonskou firmu, která se zaměřuje na výrobu automatizační techniky. Mezi hlavní produkty patří mimo průmyslových robotů i CNC obráběcí a vstřikovací stroje. Společnost nabízí širokou paletu průmyslových robotů, od robotů pro pick and place aplikace, tak i svařovací, lakovací či paletizační roboty. Z hlediska kinematiky nabízí FANUC nejen klasické kloubové roboty, ale i DELTA či SCARA roboty. Firma se také zaměřuje na vývoj a výrobu kolaborativních robotů. I v těchto kategoriích lze najít široký výběr, od malých kompaktních robotů po roboty s užitečným zatížením 2 300 kilogramů nebo dosahem přes 4,5 metrů. Na Obrázku č. 26 níže je vidět robot FANUC M-2000iA/2300, který umožňuje maximální zatížení až výše zmíněných 2 300 kg [15].



Obrázek 26: Robot od společnosti FANUC [19]



2. ABB – Švédsko-švýcarská společnost, která patří mezi největší technologické firmy na celém světě. Společnost má široký záběr v oblasti automatizace, digitalizace a robotizace. Mezi produkty patří automatizační technika (PLC), stroje pro metalurgii (slévání), nabíjecí stanice pro elektro automobily a také průmyslové roboty. Záběr průmyslových robotů je opět široký stejně jako u společnosti FANUC. ABB nabízí robotická řešení pro svařování, manipulaci s materiálem, řezání, montáž, kontrolu a mnoho dalších aplikací. Společnost opět nabízí širokou paletu robotů od menších po velké, kdy největší robot má nosnost až 800 kg a dosah 4,2 metrů [20]. Příklad robota od společnosti ABB je vidět na Obrázku č. 27 níže.



Obrázek 27: Robot od společnosti ABB [21]

3. KUKA – Německá společnost (vlastněná čínskými vlastníky) se sídlem v Augsburgu. Kromě průmyslových robotů se také zaměřuje na výrobu strojů a výrobních buněk pro svařování či výrobních linek pro automobilový průmysl. Průmyslové roboty KUKA mohou být využity pro svařování, montáž, mechanické opracování či samozřejmě manipulaci s materiálem. Průmyslové roboty společnosti KUKA mají nosnost až 1 300 kilogramů a dosah až 3,6 metrů [22]. Příklad robota je možné vidět na Obrázku č. 28 níže.



Obrázek 28: Robot od společnosti KUKA [23]

4. Yaskawa – Japonská společnost, která se zabývá výrobou frekvenčních měničů, automatizační techniky (PLC a HMI) a robotů. Průmyslové roboty mohou být využity pro různé aplikace, například svařování, řezání, montáž či balení. Firma také nabízí kolaborativní roboty. Maximální nosnost průmyslových robotů je 800 kilogramů a

dosah přes 3 metry [25]. Příklad robota od společnosti Yaskawa je vidět na Obrázku č. 29 níže.



Obrázek 29: Robot od společnosti Yaskawa [26]

5. Stäubli – Stäubli je švýcarská firma, která se zaměřuje na výrobu elektrických konektorů, upínacích systémů a průmyslových robotů. Firma se zabývá výrobou robotů typu SCARA a kloubových robotů, jejichž maximální nosnost je 170 kg a dosah přes 2,5 metrů [26]. Příklad robota značky Stäubli je vidět na Obrázku č. 30 níže.



Obrázek 30: Robot od společnosti Stäubli [27]

6. Universal Robots – tato dánská společnost se zabývá výrobou kolaborativních robotů. Koboty řady UR jsou charakteristické svými poměrně malými rozměry a jejich hlavní předností je rychlá a snadná implementace, díky intuitivnímu grafickému způsobu programování. Největším kobotem je UR20, jehož nosnost je 20 kilogramů s dosahem 1,75 metrů [28]. Tohoto kobota je možné vidět na Obrázku č. 31 níže.



Obrázek 31: Robot od společnosti Universal Robots [29]

Při automatizování procesu za pomoci robota, je třeba myslet kromě výběru samotného robota na jeho programování. Prakticky každý výrobce má svůj způsob programování respektive programovací jazyk. Detailnějšímu rozebrání problematiky programování průmyslových robotů (pro rozsah této práce postačí vysvětlení obecných principů) bude věnována následující podkapitola.

## 2.4 Programování robotů

V této podkapitole budou vysvětleny základní principy programování průmyslových robotů. Většina velkých výrobců průmyslových robotů má své vlastní programovací jazyky pro své roboty. Příkladem může být KAREL od společnosti FANUC, KRL firmy KUKA, či URScript společnosti Universal Robots.

Způsoby programování robotů lze rozdělit do dvou základních kategorií, a to online a offline programování. Principem offline programování je práce se softwarem, který umožňuje vytvořit digitální model pracoviště včetně robota, jehož je cíl naprogramovat. Programů pro tuto práci je nepřeberné množství (některé jsou univerzální, některé specializované od konkrétních výrobců). V těchto programech lze navrhovat pracoviště či ověřovat jeho parametry, jako je dosah robota, proveditelnost jednotlivých operací a to bez nutnosti přítomnosti robota, či fyzické stavby pracoviště. Jednou z možností offline programování je využití obecných programů pro simulaci procesů, jako jsou například Delmia Robotics nebo Technomatix Process Simulate, které umožňují do navrhovaného pracoviště umístit robota podporovaného daným programem. Výhodou této kategorie programů je jejich univerzálnost. Dále se k programování mohou využít softwary vyvíjené samotnými výrobci robotů, tedy například FANUC Roboguide nebo ABB Robot Studio. Primární výhodou těchto specializovaných softwarů je schopnost programovat robota pomocí virtuálního teach-pendantu tak, jako by byl programován fyzický robot. Následně je pak možné tento hotový kód nahrát do controlleru skutečného robota [1, 9].

Opakem offline programování je programování online. Online znamená, že programátor se nachází fyzicky u robota, kterého programuje. Tento proces probíhá pomocí teach-pendantu, který je rozhraním mezi controllerem robota a člověkem. Výhodou je, že je ihned vidět výsledek naprogramovaného kódu [1, 9]. Ukázka teach-pendantu společnosti FANUC je vidět na Obrázku č. 32 níže.



Obrázek 32: Teach-pendant od společnosti FANUC [31]

Velkou částí kódu jsou pohybové akce robota, tedy přejetí z jednoho místa do druhého v určitém čase (určitou rychlostí) a po určité trajektorii. Následovat bude popis základních pohybů, které robot může vykonávat [1].

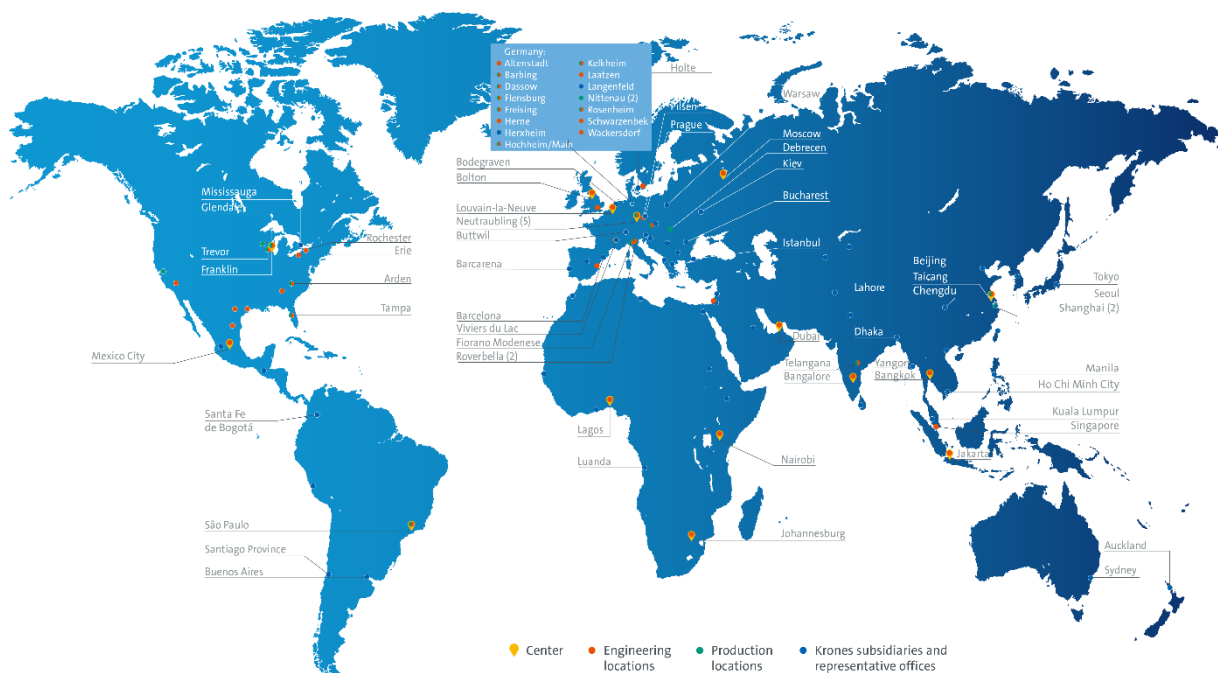
1. Obecný pohyb – Při obecném pohybu mezi dvěma body se robot pohybuje po předem neznámé obecné křivce s cílem zajistit co nejrychlejší a nejplynulejší pohyb. Obecně však neplatí, že tato dráha bude nejkratší. Právě proto se využívá při pomocných pohybech, kde nehrozí možnost kolize, a tedy konkrétní dráha je irelevantní.
2. Lineární pohyb – Při lineárním pohybu se robot pohybuje konstantní rychlostí po přímce.
3. Kruhový pohyb – Robot se v tomto pohybu pohybuje konstantní rychlostí po oblouku kruhu, kdy dráha je dána počátečním a koncovým bodem spolu s bodem mezi nimi vymezující oblouk.

### 3 Analýza současného stavu

Tato kapitola se věnuje základní charakteristice pracoviště, na kterém bude zpracována praktická část následné diplomové práce. Protože tato práce vznikla ve spolupráci s firmou Krones AG, bude ještě před popisem pracoviště tato společnost stručně představena.

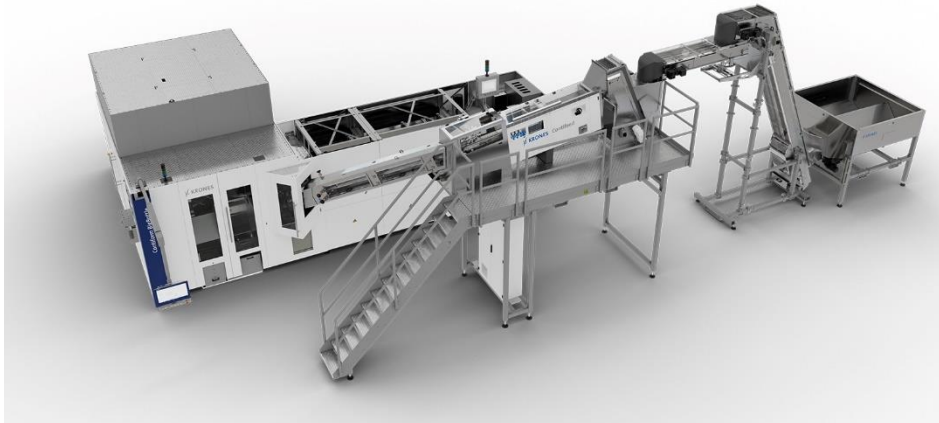
#### 3.1 Charakteristika společnosti Krones AG

Společnost Krones AG je německý výrobce linek a strojů pro výrobu, plnění a balení nápojů a tekutých potravin do PET, skleněných lahví a nápojových plechovek. Hlavním sídlem společnosti je město Neutraubling nedaleko německého Regensburgu. Pobočky firmy a se nacházejí po celém světě (viz Obrázek č. 33), například v Mexiku, Nigérii či Brazílii [33].



Obrázek 33: Mapa poboček společnosti Krones AG [34]

Kromě samotných poboček je možné najít dceřiné společnosti, které se podílejí na zpracování řešení požadavků, například firmu Konplan s. r. o se sídlem v Plzni v České republice, která se zaměřuje na oblast engineeringu. Krones AG je dodavatelem komplexních řešení, tedy od implementace jednotlivých strojů po návrh celých výrobních hal. Společnost nabízí široké portfolio strojů, které jsou součástí výrobních linek, od tvorby lahví, zpracovatelské technologie (vaření, míchání, sterilizace apod.), plnění a etiketování lahví až po balicí technologie. Firemní motto „Solutions beyond tomorrow“ („řešení za hranicí zítřka“) souvisí s hlavními cíli společnosti, kterými jsou co nejvyšší efektivita a spolehlivost výroby, minimalizace odpadu a snižování uhlíkové stopy. Na Obrázku č. 34 je možné vidět jeden z produktů společnosti. Jedná se o stroj Contiform, který se skládá z vyfukovací stanice (na obrázku vlevo), topného modulu (na obrázku uprostřed v zadní části) a zásobovacího modulu pro preformy (na obrázku uprostřed v přední části a vpravo) [33].



Obrázek 34: Stroj Contiform od společnosti Krones AG [35]

### 3.2 Popis pracoviště

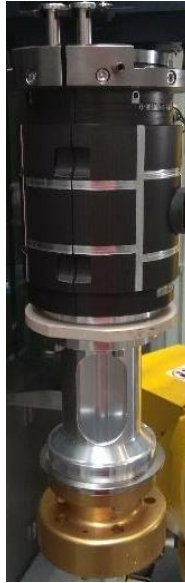
Diplomová práce se zabývá oblastí robotiky a bude zpracována na průmyslovém robotu, který je součástí jednoho stroje výrobní linky, konkrétně u stroje zvaného vyfukovací stanice (anglicky Blowing machine, německy Blasmachine).

Úkolem vyfukovací stanice je vytvořit (vyfouknout) plastovou lahev. Do stroje vstupují tzv. preformy (Obrázek č. 35). Preforma je předem vytvarovaná plastová část daných vlastností.



Obrázek 35: Plastové preforma

V první fázi výroby dochází k jejímu rovnoměrnému zahřátí a poté je takto zahřátá preforma vložena do forem na vyfouknutí (Obrázek č. 36). Každá z těchto forem je umístěna v obalu, který je připevněn ke stroji viz Obrázek č. 37. Formy jsou zevnitř konstruované tak, aby vyfouklá lahev odpovídala požadovanému tvaru a rozměru. Po vložení preformy do formy je pomocí proudu vzduchu preforma vyfouknuta do hotové plastové lahve.



Obrázek 36: Forma na vyfouknutí



Obrázek 37: Obal formy připevněný ke stroji

Pokud je potřeba v rámci produkce přejít na jiný typ lahve (například odlišného objemu, tvaru nebo rozměrů), je také potřeba vyměnit příslušné formy, které budou tvarované pro nový typ lahve. Stroj obsahuje různé množství forem v závislosti na daném modelu stroje. V době vypracování praktické části byl používán stroj, který obsahoval 28 forem. Výměna tolika forem může být časově náročná, proto se v případě, že bude výměna probíhat častěji, nabízí možnost proces výměny automatizovat.

Je patrné, že pracoviště, na kterém byla zpracována tato práce, má tento proces automatizovaný pomocí robotického manipulátoru. Konkrétně se jedná o model robota Fanuc M-710iC/50S. Na Obrázku č. 38 lze vidět robota, kterak přenáší formu.



Obrázek 38: Robotický manipulátor Fanuc M-710iC/50S nesoucí formu

Fanuc M-710iC/50S je víceúčelový šestiosý průmyslový robot charakteristický svým kratším ramenem (rameno J3) oproti ostatním robotům stejné řady. Jeho krátké rameno zajišťuje vysokou opakovatelnost ( $\pm 0,04$  mm), ovšem za cenu menšího maximálního dosahu, který je výrobcem stanoven na 1 359 mm. Výrobce také stanovuje jeho maximální užitečné zatížení, a to na 50 kg. Jeho hmotnost je 545 kg [18]. Pro účely aplikace je robot vybaven speciálním koncovým efektořem, tzv. gripper (německy Greifer), který umožňuje manipulaci s formami (viz. Obrázek č. 38 výše).

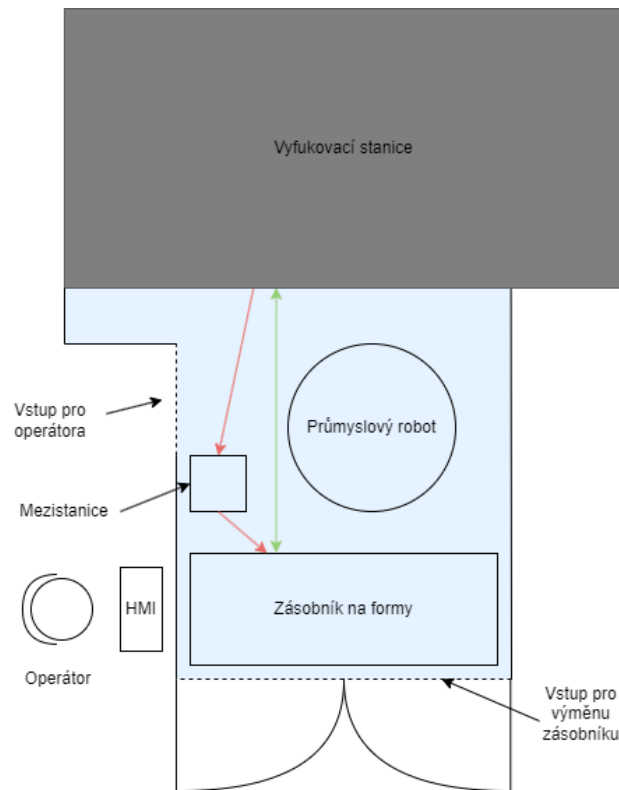
Úkolem robota je vyjmout veškeré formy z vyfukovací stanice, přemístit je do přistaveného zásobníku a z tohoto zásobníku vložit nové formy do stroje. Tato výměna probíhá ve stylu „kus za kus“. Zásobník je univerzální a obsahuje vždy 36 míst na formy. V případě, že se ve stroji nachází stejný počet forem jako míst v zásobníku, je potřeba mít ještě „mezistanici“. Do této mezistanice je vložena první vyjmutá forma, čímž se uvolní místo ve stroji, na které je možné umístit novou formu ze zásobníku. Na pracovišti, kde byla práce testována, se nachází 28 forem, jak již bylo zmíněno. Proto zde není žádná mezistanice potřeba. To se poté projeví i při programování, kdy je nutné s touto skutečností pracovat. Celé pracoviště je možné vidět na Obrázku č. 39.



Obrázek 39: Snímek pracoviště

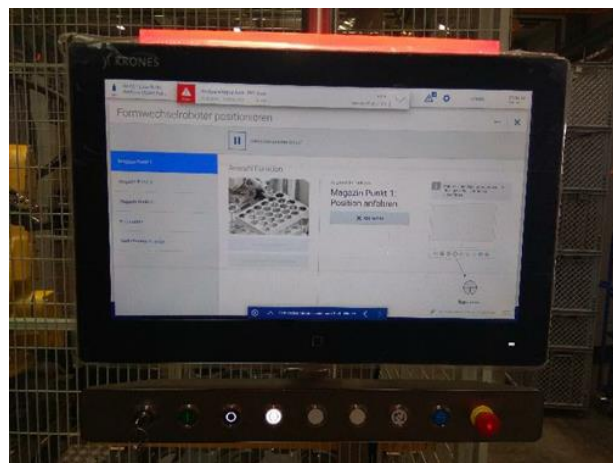


Schéma pracoviště je vidět na Obrázku č. 40 níže. Červené šipky slouží pro reprezentaci pohybu první formy ze stroje přes mezistanici (pokud by stroj obsahoval 36 forem) do zásobníku. Zelená šipka značí pohyby forem přímo ze stroje do zásobníku a naopak. Modrá zóna na tomto schématu označuje oblast uzavřenou v bezpečnostní ocelové kleci.



Obrázek 40: Schéma pracoviště

Řídicí systém se skládá ze tří různých částí. Hlavním řídicím prvkem je PLC, které řídí celý proces. Obsahuje jak kompletní řízení vyfukovací stanice, tak i „obsahu“ robota. Robot má naprogramované vlastní funkce ve svém controlleru v programovacím jazyku Fanuc KAREL a lze jej ovládat pomocí teach pendantu. Celý proces je pak možné monitorovat a obsluhovat pomocí HMI - Human Machine Interface (Obrázek č. 41). V tomto provedení se nachází pod dotykovým tabletem sada tlačítek pro obecnou interakci (tedy například červené tlačítko pro nouzové vypnutí napravo) a dále nad tabletem podlouhlá stavová LED dioda sloužící pro znázornění aktuálního stavu stroje (na obrázku svítí dioda červeně, což znamená chybu či poruchu stroje).



Obrázek 41: Human Machine Interface na pracovišti

### 3.3 Učení robotu

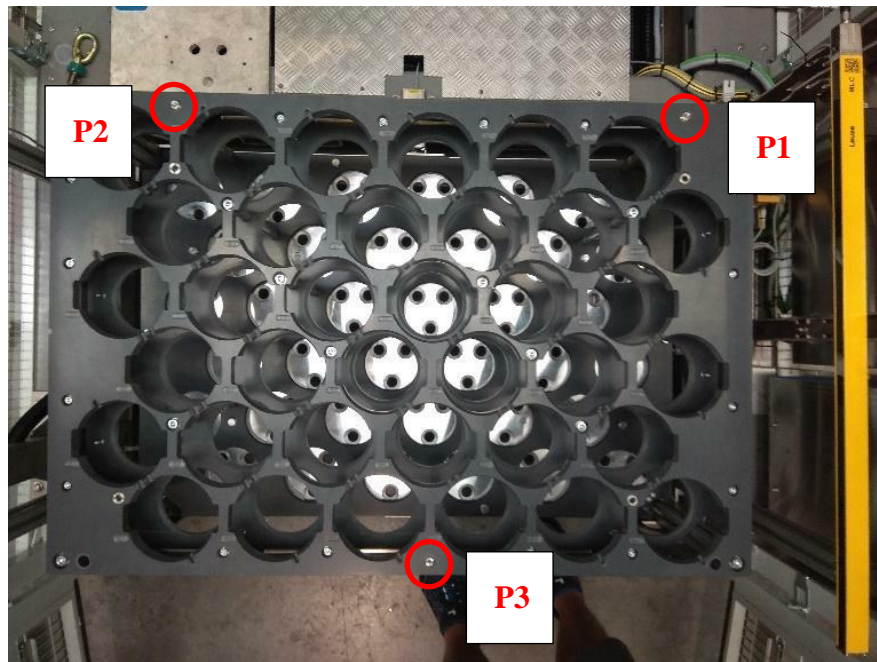
Robot na tomto pracovišti není vybaven kamerou umožňující strojové vidění. Díky této kameře by robot mohl samostatně rozpoznávat polohu forem. Protože ale kamera na pracovišti není k dispozici, je potřeba provádět tzv. učení robotu. Učení musí být provedeno pokaždé, když je robot poprvé instalován, nebo pokud dojde na pracovišti ke změně (například pokud se vymění zásobník). Principem učení je ruční napolohování robota do několika pozic a v těchto pozicích uložení aktuálních souřadnic polohy robota do PLC.

Prvním krokem ještě před samotným učením robota je přimontování speciálního dílu, tzv. cylindru (německy Zylinder). Cylindr se upevní na koncový efektor robotu (tzv. gripper) a zároveň na jednotlivé body k naučení. Cílem učení každého bodu je polohování robota tak, aby se špička cylindru na gripperu robota co nejpřesněji dotkla se špičkou cylindru na daném bodě (Obrázek č. 42).



Obrázek 42: Dotyk špičky cylindru robota se špičkou cylindru daného bodu

Celkem je nutné robota naučit pět bodů. Tři z těchto bodů se nacházejí na zásobníku. Tyto body budou dále v textu označeny jako „Magazine P1“, „Magazine P2“ a „Magazine P3“. Poloha těchto bodů na zásobníku je znázorněna na Obrázku č. 43. K obrázku je důležité dodat, že horní strana zásobníku na obrázku je bližší strana k robotu. Pokud by se na pracovišti nacházela mezistanice (dále označena „Interstation“), nacházel by se na ní další bod učení. Posledním bodem je forma (její vrchní strana) uvnitř stroje, která bude dále zmiňována jako „BlowingStation“. Zde je důležité zmínit, že stroj je synchronizovaný s robotem při výměně forem. To znamená, že jakmile robot umístí novou formu, stroj se pootočí a napolohuje další formu k výměně přesně na stejné místo, jako byla forma předchozí. Tento proces je nezbytný, neboť postačí naučit robota polohu pouze jedné formy.



Obrázek 43: Zásobník forem

V rámci vývoje je snaha soustředit řízení co nejvíce do PLC. Proto po napolohování robota do správné pozice proběhne uložení souřadnic pouze do PLC. Do controlleru robotu se tím pádem nově naučený bod neuloží a bude potřeba, aby byly nové souřadnice předány z PLC.

Aktuálním nedostatkem byla nemožnost volat funkce pro učení robota z PLC. Pokud byla potřeba robota učit, bylo nutné tak učinit především s pomocí pendantu. Nejprve se zavolala daná funkce, která zajistila pohyb robota do předem definované polohy. Tato se nacházela v širším okolí bodu, jenž je třeba naučit. Následně bylo nutné robota napolohovat tak, aby se cylindry dotýkaly.

Cílem budoucí diplomové práce bude tento proces vylepšit zapojením PLC a HMI. Volat funkce z PLC může být vhodné v případě, že byl například vyměněn zásobník na pracovišti a je potřeba pouze ověřit, zda-li jsou již naučené souřadnice shodné se souřadnicemi nového zásobníku. Pokud by se souřadnice shodovaly, nemusel by se robot znovu učit. Zároveň by nebylo potřeba ovládat robota pomocí pendantu, což by celkově urychlilo celý proces výměny zásobníku. V navazující diplomové práci bude popsáno, jaká rozšíření budou provedena, aby byla tato funkcionální možná. Obecně bude pro zajištění funkcionality potřeba rozšířit program PLC, dále funkce v rámci controlleru robota a také bude vytvořen nový screen pro HMI.

### 3.4 Cíl a popis programu

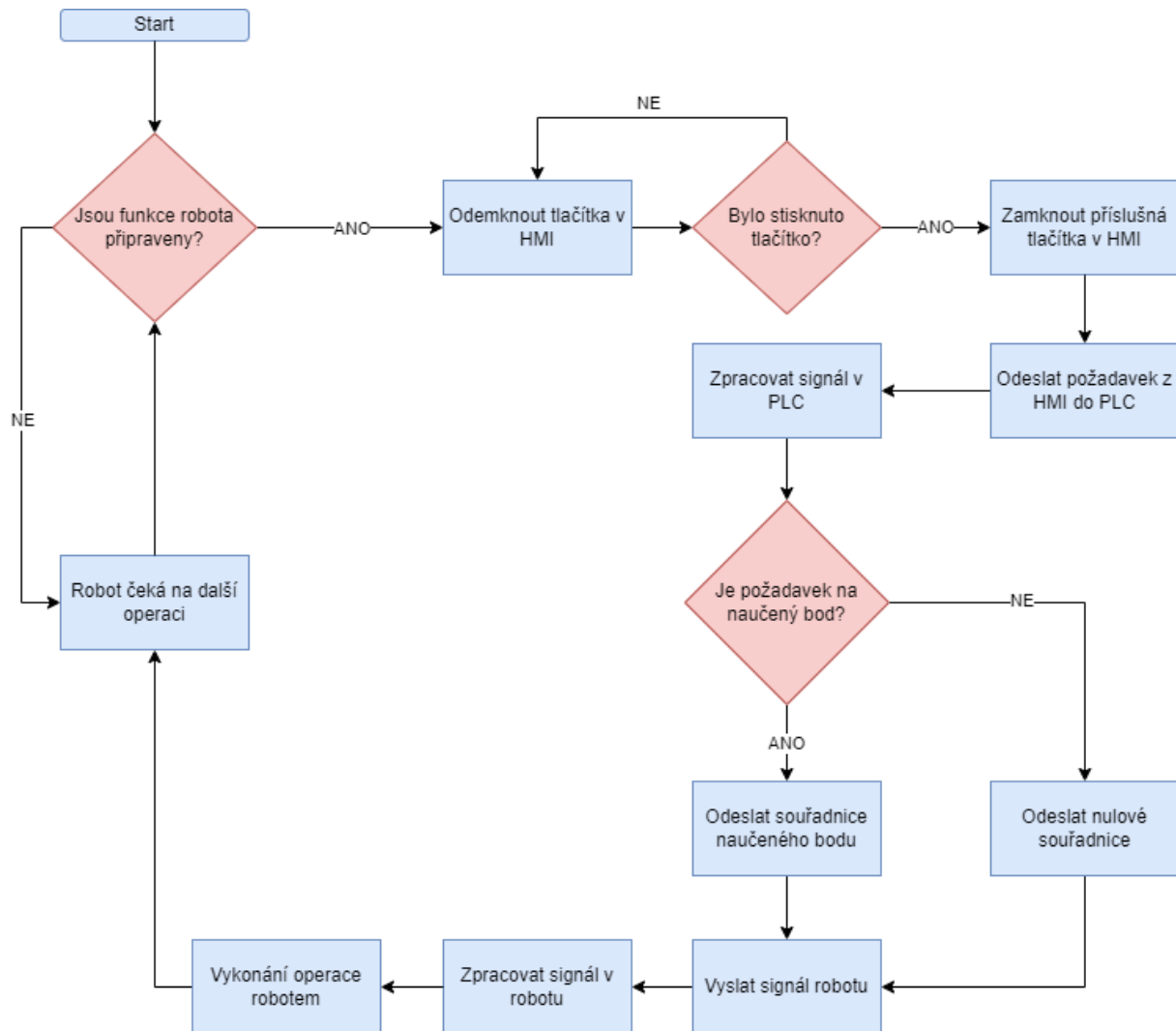
Než práce přejde k popisu vytvořených programů, je vhodné nejprve zmínit cíl zpracované úlohy a obecně sled operací, který bude úloha vykonávat.

Jak již bylo zmíněno v semestrálním projektu, cílem práce je automatizovat proces učení robota. Před zpracováním této práce byl sled operací v rámci učení robota následující. Nejprve bylo třeba zavolat funkci robota pro učení pomocí teach pendantu. Tato funkce automaticky polohovala robota do pozice, která se nacházela v blízkosti bodu, který se měl robot naučit. Bod byl zvolen tak, aby byl dostatečně blízko bodu učení, ale zároveň byla jistota, že robot cestou ani v daném bodě nebude kolidovat s žádným jiným objektem. Poté se manuálně pomocí teach pendantu napolohoval do správné polohy pro učení, aby došlo k doteku cylindrů (viz semestrální projekt). Poté operátor uložil pomocí tlačítka na HMI souřadnice robota do PLC. Takto se naučily všechny potřebné body a robot mohl být uveden do provozu.

Pokud se na pracovišti něco změnilo, například byl vyměněn zásobník, bylo potřeba zkontrolovat, zda-li byl například zmíněný zásobník umístěn shodně s předchozím, či nikoliv. Pokud by byl umístěn jinak (v řádu milimetrů), bylo by třeba robota znovu učít. Dosud jediný způsob ověření této skutečnosti bylo podívat se pomocí HMI do paměti PLC a manuálně napolohovat robota pomocí teach pendentu na uložené souřadnice a zjistit odchylku od poslední změny.

Cílem práce tedy bylo tento proces zjednodušit za využití PLC a HMI. Operátor by tedy mohl zavolat funkce robota pomocí grafického rozhraní. Zároveň přibude možnost automaticky napolohovat robota do naučených pozic. V ideálním případě, kdy provedené změny neovlivnily naučené body (například nový zásobník byl umístěn přesně), je možné proces ověření provést bez použití teach pendentu. V případě změny na pracovišti dojde přesto k urychlení celého procesu, protože bude možné robota polohovat do předem naučené polohy, která bude v těsnější blízkosti bodu, který bude třeba učít.

Nyní práce přejde k popisu postupu operací, které bude muset program vykonat, aby splnil požadovanou funkcionalitu. Pro grafické znázornění byl vytvořen vývojový diagram, který je možné vidět na obrázku č. 44 níže.



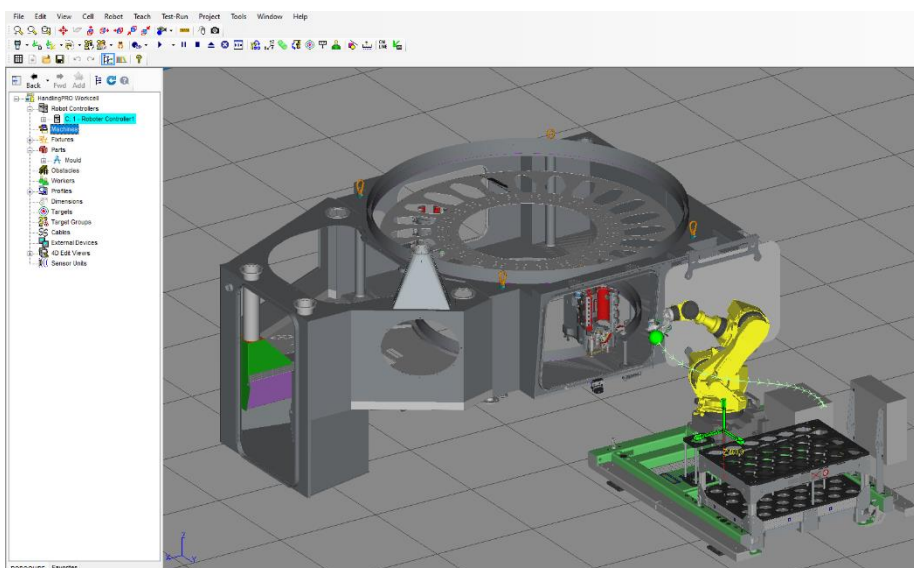
Obrázek 44: Vývojový diagram práce

Protože běh navrženého programu je spojen s během celého pracoviště, začíná program spolu se zapnutím stroje. Prvním krokem po zapnutí stroje (včetně robota) je ověření, zdali jsou funkce robota (tedy i robot sám) připraveny. Pokud ano, je možné odemknout příslušná tlačítka

na HMI (od spuštění jsou tlačítka zamčena). V tuto chvíli se čeká na akci operátora v podobě stisknutí tlačítka a tím zavoláním funkce robota. Pokud byla tato akce provedena, zamknou se příslušná tlačítka, aby nebylo možné akci robota přerušit voláním jiné funkce. Stisknutím tlačítka se zároveň odešle signál z HMI do PLC, kde bude signál zpracován (viz kapitola PLC programování). Zde z hlediska běhu programu je potřebné zjistit, zdali byl požadavek na již naučený bod, či na předem definovaný bod. V prvním případě, se odešlou souřadnice požadovaného naučeného bodu. V opačném případě se pak odešlou nulové souřadnice (viz následující kapitola). Následně se vyšle signál robota, který bude obsahovat jak souřadnice bodu, tak číslo funkce, kterou má robot vykonat. Následně se poslaný signál z PLC zpracuje v robota, který následně operaci vykoná. Protože program probíhá v nekonečné smyčce, tak pokud byla funkce v pořádku dokončena, nebo přerušena, proběhne znovu ověření, zdali je robot připraven na další operaci, čímž se program dostane do počátečního stavu celé smyčky. Běh programu jako takový končí spolu s vypnutím stroje. V rámci praktické části pak byly všechny tyto kroky naprogramovány a následující kapitoly budou popisovat jednotlivé postupy a principy spolu s jednotlivými snímky provedených změn a rozšíření. Konkrétně v následující kapitole se práce zaměří na programování samotného průmyslového robota, který výměnu forem fyzicky provádí.

## 4 Programování průmyslového robota

Prvním úkolem bylo rozšířit stávající funkce pro učení robota. K programování těchto rozšíření byl využit software Fanuc Robotics Roboguide. S ohledem na to, že tento software je vyvíjen společností Fanuc, je možnost programovat robota pomocí virtuálního teach pendantu, který je totožný se skutečným. To mimo urychluje implementaci nových verzí, kdy je třeba pouze nahrát obsah virtuálního pendantu do fyzického pendantu připojenému ke skutečnému robotu. V softwaru Roboguide má společnost již vytvořený model pracoviště, který je možné vidět na obrázku č. 45 níže.



Obrázek 45: Model pracoviště v SW Roboguide

V rámci programování robota bylo potřeba udělat několik změn ve funkcích a zároveň přidat některé registry nutné pro zajištění funkcionality.

### 4.1 Setup funkce

V první řadě bylo nutné upravit samotné funkce pro učení robota. V rámci této práce budou dále tyto funkce nazývané také „teach“ funkce. Pro každý bod učení existuje vlastní teach funkce. Před rozšířením fungovaly funkce pro jednotlivé body tak, že měly v sobě definované souřadnice bodu, který se nacházel poblíž bodu, jež byl potřeba naučit. Robot do daných pozic najel, a posléze operátor manuálně nastavil robota do správné polohy, kterou následně uložil do PLC. Rozšíření těchto funkcí spočívalo v možnosti rozhodnout se, zda-li má robot najet do předem definované pozice nebo do již naučené pozice, a to na základě požadavku z PLC. K tomu bylo třeba vytvořit registr. Konkrétně byl vytvořen registr č. 66 s názvem „Teach Mode“. Způsob nastavení hodnoty registru bude popsán níže. Na obrázku č. 46 je možné vidět úpravu kódu, která byla v rámci práce provedena.

```
27: IF (R[66:Teach Mode]=1) THEN
28:J PR[66:Teach PLC] 100% CNT100
29: ELSE
30:J PR[61:Teach Magazine 1] 100%
: CNT100
31: ENDIF
```

Obrázek 46: Rozšíření teach funkce

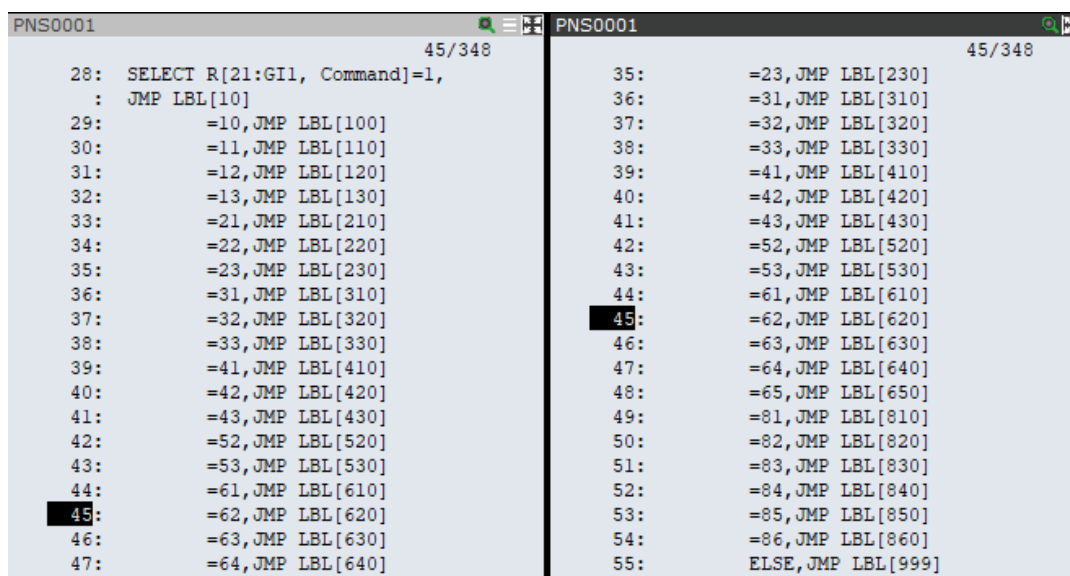
V rámci práce byl také vytvořen polohový registr č. 66 s názvem „Teach PLC“, ve kterém jsou uloženy souřadnice poslané z PLC a odpovídají naučeným souřadnicím zvoleného bodu z fáze učení. V polohových registrech 61 až 65 jsou pak uloženy předem vytvořené souřadnice

pro jednotlivé učící body. Výše popsané změny byly provedeny zároveň také ve zbylých teach funkcích.

V rámci práce vznikl také požadavek na zajištění stejné funkcionality (tedy zavolání funkce z PLC) pro funkci, která uvede robota do přepravní polohy (anglicky „Delivery position“, německy „Auslieferungs Position“). Zde však nebyl požadavek na učení robota, jelikož přepravní poloha je uložena v controlleru robota. Spolu s teach funkcemi bude tato skupina funkcí dále nazývána „setup funkce“.

## 4.2 Funkce PNS0001

Další funkcí, která byla upravena, byla funkce PNS0001. Tato funkce slouží jako řídicí prvek controlleru robota. Jejím úkolem je volat ostatní funkce na základě požadavků z PLC, a proto běží po celou dobu běhu stroje. Z PLC přicházejí požadavky v podobě čísel příkazů (anglicky „Command“), které jsou definované v PLC stejně (se shodnou hodnotou) jako v controlleru robota. Prvním krokem úpravy tedy bylo přidání čísel příkazů. Volba čísel byla libovolná z dosud neobsazených pozic. První část funkce řídicí další průběh je možné vidět na obrázku č. 47 (z důvodu přehlednosti je pendant nastaven na dva displeje, aby bylo možné zobrazit na jednom obrázku vše potřebné).



```
PNS0001 45/348
28: SELECT R[21:GI1, Command]=1,
   : JMP LBL[10]
29:     =10, JMP LBL[100]
30:     =11, JMP LBL[110]
31:     =12, JMP LBL[120]
32:     =13, JMP LBL[130]
33:     =21, JMP LBL[210]
34:     =22, JMP LBL[220]
35:     =23, JMP LBL[230]
36:     =31, JMP LBL[310]
37:     =32, JMP LBL[320]
38:     =33, JMP LBL[330]
39:     =41, JMP LBL[410]
40:     =42, JMP LBL[420]
41:     =43, JMP LBL[430]
42:     =52, JMP LBL[520]
43:     =53, JMP LBL[530]
44:     =61, JMP LBL[610]
45:     =62, JMP LBL[620]
46:     =63, JMP LBL[630]
47:     =64, JMP LBL[640]

PNS0001 45/348
35:     =23, JMP LBL[230]
36:     =31, JMP LBL[310]
37:     =32, JMP LBL[320]
38:     =33, JMP LBL[330]
39:     =41, JMP LBL[410]
40:     =42, JMP LBL[420]
41:     =43, JMP LBL[430]
42:     =52, JMP LBL[520]
43:     =53, JMP LBL[530]
44:     =61, JMP LBL[610]
45:     =62, JMP LBL[620]
46:     =63, JMP LBL[630]
47:     =64, JMP LBL[640]
48:     =65, JMP LBL[650]
49:     =81, JMP LBL[810]
50:     =82, JMP LBL[820]
51:     =83, JMP LBL[830]
52:     =84, JMP LBL[840]
53:     =85, JMP LBL[850]
54:     =86, JMP LBL[860]
55:     ELSE, JMP LBL[999]
```

Obrázek 47: Funkce PNS0001 – výběr funkce

Z obrázku je patrné, že na základě hodnoty registru 21 bude pokračovat program s vykonáním příkazů za labelem uvedeným na řádce splněné podmínky, tedy například pro command 81 bude aplikován „skok“ na label 810. Pro setup funkce byly zvoleny příkazy 81 až 86 a odpovídající labely 810, 820, 830, 840, 850 a 860.

Na následujícím obrázku č. 48 je zobrazen celý příkaz (konkrétně pro setup funkci pro bod „Magazine P1“). Příkaz začíná odpovídajícím labelem, na který se program přesune na základě první části funkce. Poté zde dochází k uložení souřadnic zaslaných z PLC do výše uvedeného polohového registru 66. Souřadnice se nacházejí v registrech 22 až 27 a odpovídají souřadnicím ve smyslu os X, Y a Z, a rotaci kolem jednotlivých os. Do registrů jsou souřadnice namapovány z I/O rozhraní s PLC. Hodnoty poloh jsou dále vyděleny 10, protože hodnoty přicházející z PLC jsou s desetinná čísla (například 100,1). Controller robota pracuje s celými čísly, a proto by se hodnoty do controlleru uložily jako celé číslo o řád vyšší (tedy například 1001). To by způsobilo najetí robota do nesprávné (nepřesné) pozice, a zároveň by mohlo dojít i ke kolizi robota s předměty v jeho okolí (ochranná klec, zásobník, stroj,...). Souřadnice by

také mohly být mimo dosah robota a pokud by se robot o toto vyjetí pokusil, došlo by k chybě (zablokování robota), která je nutná řešit manuálně pomocí teach pendantu.

Následuje rozhodnutí, na jakou hodnotu se nastaví registr 66. Rozhodovací algoritmus funguje na základě příchozích hodnot z PLC. Pokud jsou všechny souřadnice X, Y a Z rovny nule, nastaví se hodnota registru 66 na hodnotu 0, tedy robot najede na předem definované souřadnice. V opačném případě se nastaví registr na hodnotu 1 a robot najede na naučené souřadnice. Poté se jen zavolá konkrétní setup funkce (na obrázku pro bod „Magazine P1“), nastaví se hodnota registru zpět na nulu a program přeskočí na label č. 1000, kdy bude robot čekat na další příkaz z PLC.

```

PNS0001 240/337
229: !+-----+
230: !COMMAND 81
231: !TEACH_MAGAZINE_P1
232: !+-----+
233: LBL[810]
234: PR[66,1:Teach PLC]=
: R[22:GI2, Value 1, X]
235: PR[66,2:Teach PLC]=
: R[23:GI3, Value 2, Y]
236: PR[66,3:Teach PLC]=
: R[24:GI4, Value 3, Z]
237: PR[66,4:Teach PLC]=
: R[25:GI5, Value 4, W]
238: PR[66,5:Teach PLC]=
: R[26:GI6, Value 5, P]
239: PR[66,6:Teach PLC]=
: R[27:GI7, Value 6, R]
240: IF (PR[66,1:Teach PLC]=0 AND
: PR[66,2:Teach PLC]=0 AND
: PR[66,3:Teach PLC]=0) THEN
241: R[66:Teach Mode]=0

PNS0001 236/337
235: PR[66,2:Teach PLC]=
: R[23:GI3, Value 2, Y]
236: PR[66,3:Teach PLC]=
: R[24:GI4, Value 3, Z]
237: PR[66,4:Teach PLC]=
: R[25:GI5, Value 4, W]
238: PR[66,5:Teach PLC]=
: R[26:GI6, Value 5, P]
239: PR[66,6:Teach PLC]=
: R[27:GI7, Value 6, R]
240: IF (PR[66,1:Teach PLC]=0 AND
: PR[66,2:Teach PLC]=0 AND
: PR[66,3:Teach PLC]=0) THEN
241: R[66:Teach Mode]=0
242: ELSE
243: R[66:Teach Mode]=1
244: ENDIF
245: CALL COMMAND_ACTIVE
246: CALL TEACH_MAGAZINE_P1
247: R[66:Teach Mode]=0
248: JMP LBL[1000]
    
```

Obrázek 48: Funkce PNS0001 – teach funkce

Dalším krokem bylo přidání příkazu pro zavolání funkce „Delivery position“. Protože jsou souřadnice této polohy uloženy v controlleru, není třeba žádných I/O operací. Je ovšem nutné z bezpečnostních důvodů před zavoláním samotné funkce uvést robota do domácí polohy. Domácí poloha je narozdíl od přepravní polohy dostupná ze všech potenciálních poloh robota. Díky tomu nemůže dojít k nárazu robota do konstrukce, či vyjetí ze svého dosahu. Příkaz pro zavolání funkce „Delivery position“ je na obrázku č. 49 níže.

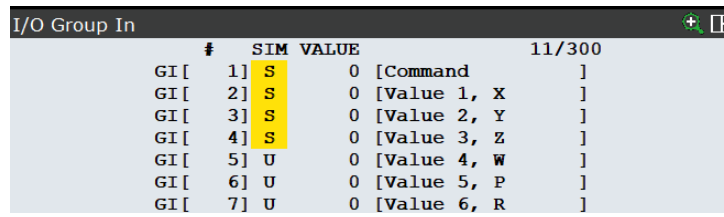
```

323: !+-----+
324: !COMMAND 86
325: !DELIVERY_POSITION
326: !+-----+
327: LBL[860]
328: CALL COMMAND_ACTIVE
329: CALL HOME
330: CALL DELIVERY_POSITION
331: JMP LBL[1000]
    
```

Obrázek 49: Funkce PNS0001 – funkce pro přepravní polohu

V rámci programu Roboguide bylo možné odzkoušet funkčnost navržených změn. Program však není nijak propojen s PLC, a proto se v rámci simulace dají příkazy ručně zadávat, stejně tak jako souřadnice. Změna se provede v rámci I/O operací v controlleru, viz Obrázek č. 50, kde se na první řádek napíše číslo příkazu (81 až 86) a poté se ručně zadají souřadnice. Po spuštění PNS0001 tak dojde ke stejnému sledu operací, jako při skutečné situaci.





I/O Group In				
#	SIM	VALUE		11/300
GI[ 1]	S	0	[Command	]
GI[ 2]	S	0	[Value 1, X	]
GI[ 3]	S	0	[Value 2, Y	]
GI[ 4]	S	0	[Value 3, Z	]
GI[ 5]	U	0	[Value 4, W	]
GI[ 6]	U	0	[Value 5, P	]
GI[ 7]	U	0	[Value 6, R	]

Obrázek 50: Vstupně výstupní rozhraní

Nyní, když byly popsány změny a rozšíření provedené v rámci průmyslového robota, přejde práce k popisu změn a rozšíření stávajícího PLC programu, které byly třeba udělat, pro zajištění požadované funkcionality.

## 5 PLC programování

Veškeré PLC programování bylo provedeno v programu TIA Portal od firmy Siemens, ve kterém je naprogramován celý stroj i obsluha robota. V rámci rozšíření funkcí bylo potřeba naprogramovat dva úkony. Zaprvé bylo nutné na základě signálu zavolat konkrétní setup funkci robota i s požadovanými souřadnicemi. Zadruhé pak vytvořit část programu, která zajistí zpracování požadavku z HMI.

Nejprve bude v této kapitole popsán první úkol, tedy zajištění zpracování požadavku z HMI a následné zavolání požadované funkce robota. Proto bylo potřeba vytvořit nové UDT a rozšířit stávající UDT o potřebné proměnné. Následně bude vysvětlen způsob přiřazení hodnot ID. Poté budou popsány networky pro setup funkce. Následně se kapitola bude věnovat řízení zpracování samotného signálu z HMI, spolu s nastavením viditelnosti tlačítek a jejich zamykáním. K tomu bylo třeba rozšířit stávající UDT a naprogramovat networky pro jednotlivé setup funkce.

### 5.1 Deklarace datových typů

První částí rozšíření funkcí bylo zavolání konkrétní setup funkce v závislosti na vstupním signálu z HMI. Za tímto účelem byl vytvořen Function Block, kdy se každá setup funkce skládá ze tří networků. Jednotlivé networky budou popsány níže. Nejprve však bylo potřeba vytvořit proměnné (v podobě UDT), které se poté využily v jednotlivých funkcích. Na obrázku č. 51 níže se deklarují datové typy pro jednotlivé funkce.

Pro jednoduchost budou pro každý bod, který se robot učí, vytvořeny dvě sady proměnných. Jedna sada proměnných bude definována pro naučený bod (dále „Teachpoint“) a druhá pro předem definované souřadnice (dále „Zeropoint“). V HMI pak budou jednotlivé sady proměnných řešeny dvěma tlačítky pro každý bod, kdy operátor vybere, na které souřadnice robot najede. Z důvodu rozsáhlosti některých networků budou obrázky pro účely dokumentace zobrazeny ve více snímcích.

SetupFunctions_UDT									
	Name	Data type	Default value	...	...	...	...	Supervision	Comment
1	ActiveID	Int	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Active ID
2	MagazineP1_Teachpoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in Magazine taught Position 1
3	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
4	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected taught point
5	MagazineP1_Zeropoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in Magazine zero Position 1
6	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
7	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected zero point
8	MagazineP2_Teachpoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in Magazine taught Position 2
9	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
10	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected taught point
11	MagazineP2_Zeropoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in Magazine zero Position 2
12	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
13	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected zero point
14	MagazineP3_Teachpoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in Magazine taught Position 3
15	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
16	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected taught point
17	MagazineP3_Zeropoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in Magazine zero Position 3
18	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
19	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected zero point
20	Interstation_Teachpoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in Interstation taught Position
21	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
22	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected taught point
23	Interstation_Zeropoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in Interstation zero Position
24	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
25	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected zero point
26	BlowingStation_Teachpoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in BlowingStation taught Position
27	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
28	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected taught point
29	BlowingStation_Zeropoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in BlowingStation zero Position
30	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
31	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected zero point
32	BlowingStation_Zeropoint	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in BlowingStation zero Position
33	DeliveryPosition	Struct		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Robot in delivery Position
34	Ready	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Ready for operation
35	Selected	Bool	0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1: Task selected delivery position

Obrázek 51: UDT pro jednotlivé setup funkce

Pro každou setup funkci jsou stanoveny dvě proměnné typu boolean.

1. Ready – pokud je hodnota 1, je funkce připravená k použití
2. Selected – pokud je hodnota 1, je daná funkce zvolena

V prvním řádku UDT si lze všimnout proměnné ActiveID, která v průběhu výkonu funkcí nese číslo (ID) dané aktivní operace.

Poté je také zapotřebí rozšířit stávající UDT, jenž obsahuje všechny používané funkce robota. Rozšíření je vidět na obrázku č. 52 níže (z důvodu ochrany dat jsou viditelné pouze provedené změny).

10	SetupFunctions	Struct						Setup functions
11	BlowingStation_Teachpoint	Bool	FALSE					1: Request move Robot to BlowingStation teached po
12	BlowingStation_Zeropoint	Bool	FALSE					1: Request move Robot to BlowingStation zero point
13	Interstation_Teachpoint	Bool	FALSE					1: Request move Robot to Interstation teached point
14	Interstation_Zeropoint	Bool	FALSE					1: Request move Robot to Interstation zero point
15	Magazine_P3_Teachpoint	Bool	FALSE					1: Request move Robot to Magazine teached P3
16	Magazine_P3_Zeropoint	Bool	FALSE					1: Request move Robot to Magazine zero P3
17	Magazine_P2_Teachpoint	Bool	FALSE					1: Request move Robot to Magazine teached P2
18	Magazine_P2_Zeropoint	Bool	FALSE					1: Request move Robot to Magazine zero P2
19	Magazine_P1_Teachpoint	Bool	FALSE					1: Request move Robot to Magazine teached P1
20	Magazine_P1_Zeropoint	Bool	FALSE					1: Request move Robot to Magazine zero P1
21	DeliveryPosition	Bool	FALSE					1: Request move Robot to delivery position

Obrázek 52: Rozšíření stávajícího UDT robota

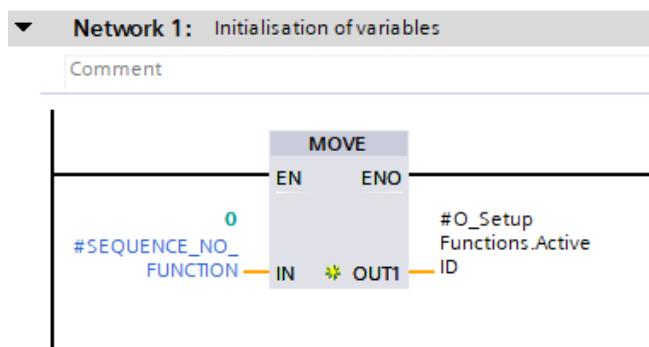
## 5.2 Inicializace proměnné „ActiveID“

Celý Function Block začíná inicializací proměnné „ActiveID“. Pro účely této práce mohou z pohledu přiřazení jednotlivých ID nastat 4 situace. Pokud není zvolena žádná funkce, přiřadí se ID hodnota rovna 0. Hodnoty 51, respektive 52 se přiřadí v případě, že z PLC přijde požadavek na pohyb robota do naučeného, respektive předem definovaného bodu. Čtvrtou možností je požadavek na přepravní polohu, který nese ID hodnotu 55. Veškeré výše zmíněné situace jsou definovány pomocí konstant uvedených na obrázku č. 53 níže.

32	SEQUENCE_NO_FUNCTION	Int	0					Sequence 0: no function
33	SEQUENCE_SETUP_TEACH_POSITION	Int	51					Sequence 51: Setup teach Position
34	SEQUENCE_SETUP_ZERO_POSITION	Int	52					Sequence 52: Setup zero Position
35	SEQUENCE_SETUP_DELIVERY_POSITION	Int	55					Sequence 55: Setup Delivery position

Obrázek 53: Hodnoty ID pro jednotlivé funkce

Celý Program blok začíná nastavením proměnné „ActiveID“ na hodnotu nula, tedy není zvolena žádná funkce. Network je vidět na obrázku č. 54 níže.

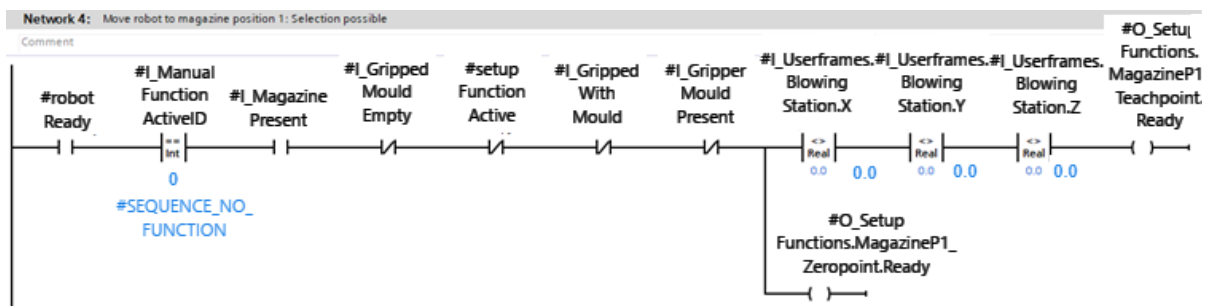


Obrázek 54: Network pro iniciaci ID proměnné

Každá jednotlivá setup funkce obsahuje 3 networky, které budou popsány v následujících podkapitolách. Pro všechny jednotlivé setup funkce jsou networky ve velké míře stejné, proto bude postup ukázán na setup funkci pro bod „Magazine P1“. V případě odlišností networků ostatních funkcí, budou tyto rozdíly také uvedeny.

## 5.3 Network „Selection possible“

První network setup funkce je „Selection possible“, ve kterém je cílem nastavení bitu „ready“. Network je vidět na obrázku č. 55 níže.

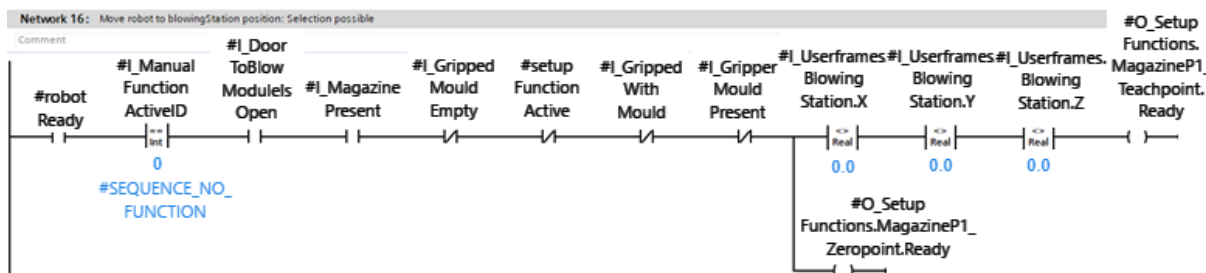


Obrázek 55: Network „Selection possible“ pro teach funkci „Magazine P1“

Aby byla funkce připravena k použití, musí být splněny následující podmínky:

1. robotReady – hodnota musí být 1, tj. robot je připraven k vykonání příkazu
2. I\_ManualFuctionsActiveID – Zde se jedná o manuální funkce robota, které sdílí stejnou proměnnou pro uchování ID hodnoty dané funkce. Pokud je hodnota rovna nula, pak není zvolena žádná setup ani manuální funkce.
3. I\_MagazinePresent – hodnota musí být 1, tj. zásobník je přistaven na pracovišti
4. setupFunctionActive – hodnota musí být 0, tj. žádná jiná funkce není aktivní
5. I\_GrippedMouldEmpty, I\_GrippedWithMould a I\_GrippedMouldPresent – hodnota musí být 0, tj. robot „nedrží“ žádnou formu
6. I\_Userframes.Magazine[1].X/Y/Z – v této proměnné jsou uloženy souřadnice naučeného bodu (v tomto případě bodu Magazine P1). Pokud jsou hodnoty nulové, není žádný bod naučen. Jinými slovy budou tyto podmínky splněné, jestliže byl již daný bod naučen. Pokud nebyl doposud žádný bod naučen, je však při splnění výše uvedených podmínek možné využít funkci pro polohování robota do předem definovaných souřadnic.

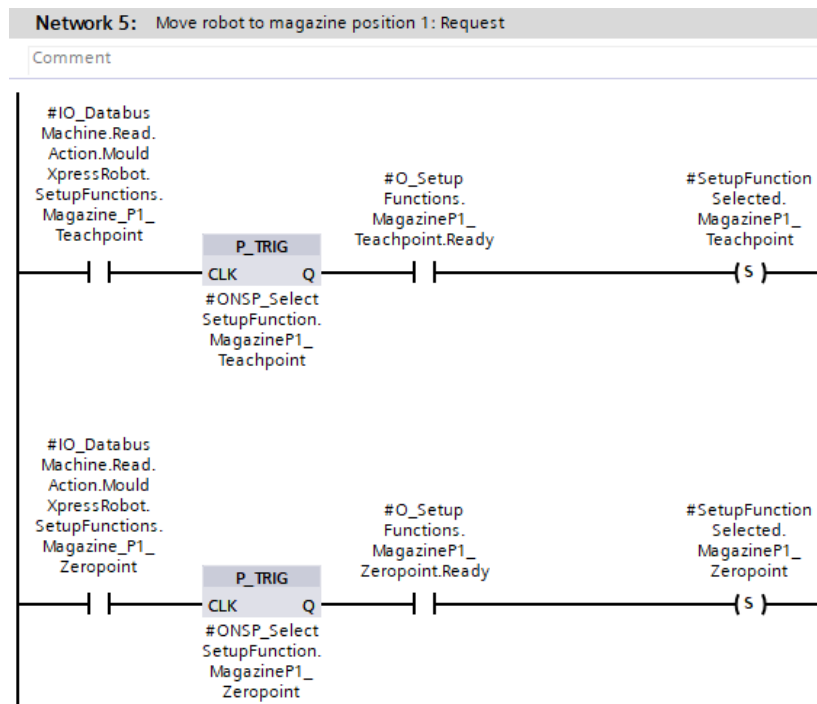
Podmínky z bodu 6. nejsou součástí networku pro „Delivery position“, neboť se zde nejedná o teach funkci, a tedy není potřeba žádný bod učit. Pro bod „Blowing station“ musí být zároveň splněna podmínka „I\_DoorToBlowModuleIsOpen“, neboli dveře mezi strojem a robotem musí být otevřeny. Network pro „Blowing station“ je vidět na obrázku č. 56 níže. Z důvodu čitelnosti byly texty upraveny mimo program TIA Portal.



Obrázek 56: Network „Selection possible“ pro teach funkci „Blowing station“

## 5.4 Network „Request“

Druhý network zajišťuje zpracování signálu z HMI a aktivuje interní proměnnou. Pomocí té je následně možné rozhodnout, které souřadnice mají být odeslány. Obrázek networku „Request“ č. 57 můžeme vidět níže.



Obrázek 57: Network „Request“

Aby byl nastaven bit pro výběr souřadnic („SetupFunctionSelected.MagazineP1\_...“) je potřeba následující:

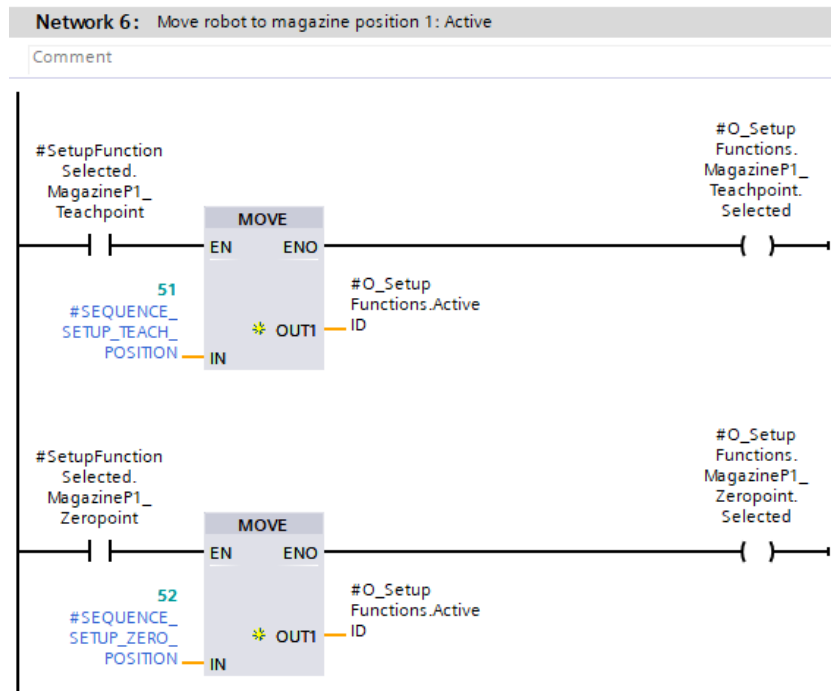
1. IO\_Databus.Machine.Read.Action.MouldXpressRobot\_SetupFunctions.Magazine\_P1-Teachpoint/Zeropoint – hodnota bitu musí být 1, tj. operátor stiskl konkrétní tlačítko pro zvolenou funkci. (Nastavení tohoto bitu bude popsáno dále).
2. ONSP\_SelectSetupFunction.MagazineP1\_Teachpoint/Zeropoint – dává informaci o stisknutí tlačítka (při stisknutí tlačítka dojde ke změně hodnoty bitu výše zmíněného, čímž v jednom cyklu běhu PLC dojde k propuštění signálu)
3. O\_SetupFunctions.MagazineP1\_Teachpoint/Zeropoint.Ready – hodnota bitu musí být 1 (bit je nastaven v přechodím networku)

Pro funkci „Delivery position“ se žádné souřadnice neposílají, proto není třeba dvou větví, nýbrž postačí jedna.

## 5.5 Network „Active“

V této podkapitole bude popsán poslední network s názvem „Active“. Cílem tohoto networku je zavolat setup funkci uloženou v robotu a zajistit odeslání správných souřadnic dle volby operátora. Z důvodu přehlednosti bude network rozdělen na více obrázků (č. 58, č. 60 a č. 61).

V první části zobrazené na obrázku č. 58 dochází k volbě setup funkce („O\_SetupFunctions.MagazineP1\_...“) na základě parametru nastaveného v předchozím networku („SetupFunction-Selected.MagazineP1\_...“). Na základě stejné podmínky bude přiřazena hodnota proměnné „ActiveID“.



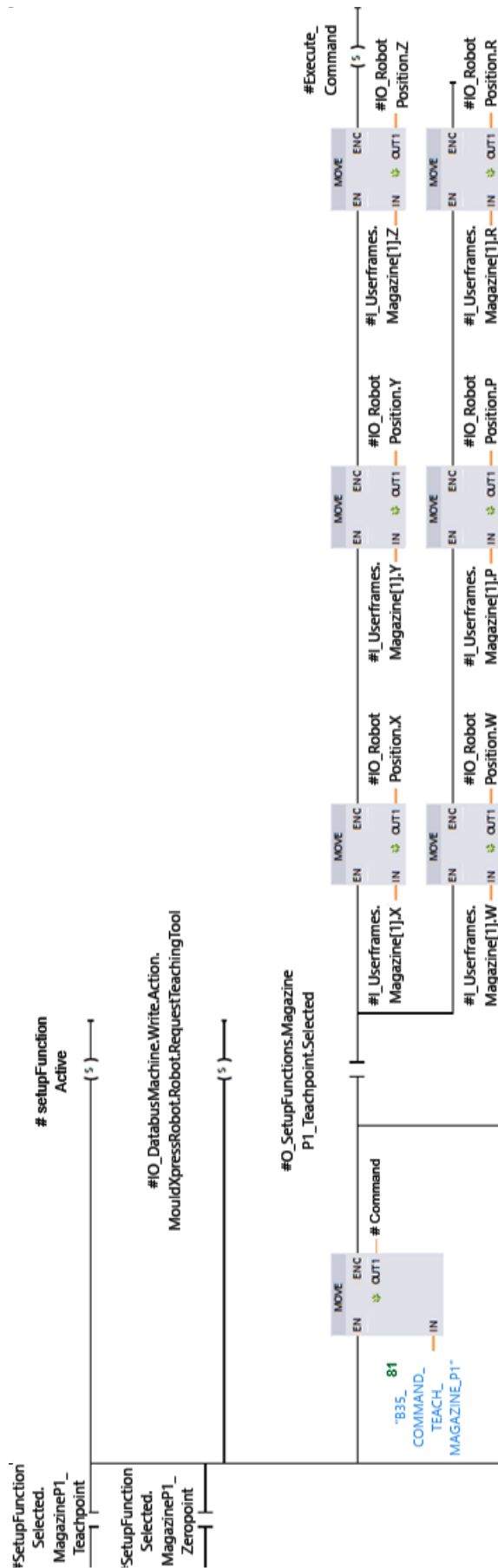
Obrázek 58: Network Active – první část

V druhé části networku „Active“ se nejprve na základě hodnoty jednoho z bitů na obrázku č. 58 nastaví hodnota bitu „setupFunctionActive“ na číslo 1, čímž se zamezí zavolání jiných setup funkcí. V této části networku také dochází k zavolání setup funkce. To je dosaženo pomocí nastavení hodnoty proměnné „Command“, do které je uloženo číslo příkazu definované v příkazech. Definiční lze vidět na obrázku č. 59. Zde je nutné podotknout, že čísla příkazů musí odpovídat číslům příkazů v controlleru robota, viz obrázek č. 47 v podkapitole Funkce PNS0001.

B35_COMMAND_TEACH_MAGAZINE_P1	Int	81	Command: Teach magazine P1
B35_COMMAND_TEACH_MAGAZINE_P2	Int	82	Command: Teach magazine P2
B35_COMMAND_TEACH_MAGAZINE_P3	Int	83	Command: Teach magazine P3
B35_COMMAND_TEACH_INTERSTATION	Int	84	Command: Teach interstation
B35_COMMAND_TEACH_BLOWINGSTATION	Int	85	Command: Teach blowingStation
B35_COMMAND_DELIVERY_POSITION	Int	86	Command: Delivery position

Obrázek 59: Označení jednotlivých příkazů

Druhou část networku „Active“ je možné vidět na obrázku č. 60 a z důvodu čitelnosti je následující obrázek otočen o 90 stupňů a texty byly upraveny mimo program TIA Portal.



Obrázek 60: Network Active – druhá část



V robotu je vždy naprogramována pouze jedna funkce pro učící bod. Z tohoto důvodu je i v PLC nutný pouze jeden příkaz. Dvě funkce jsou již zapotřebí ve chvíli, kdy má PLC poslat správné souřadnice. Proto na základě podmínky nastavené v networku výše („O\_SetupFunctions.MagazineP1...“), budou poslány do robota buď naučené souřadnice (v případě, že je splněna podmínka pro Teachpoint) nebo nulové souřadnice (v případě, že je splněna podmínka pro Zeropoint). Větev s posíláním nulových souřadnic lze vidět na obrázku č. 18 níže. Následně je v obou případech nastaven bit „Execute\_Command“ na hodnotu 1, čímž se odešle požadavek do controlleru robota. Nulové souřadnice jsou posílány z důvodu způsobu naprogramování volání setup funkcí v rámci funkce PNS0001, viz Obrázek č. 48.

Poslední část networku Active je možné vidět na obrázku č. 61 níže. Opět z důvodu čitelnosti je obrázek otočen o 90 stupňů a texty byly upraveny mimo program TIA Portal.



V této části dochází k odeslání nulových souřadnic a aktivování daného příkazu. Dále je zde provedeno ukončení setup funkce, kdy je potřeba vrátit hodnotu proměnné „Execute\_Command“ na hodnotu 0. Zároveň se na hodnotu nula nastaví i interní proměnné „SetupFunctionSelected...“, které zamezí v dalším cyklu aktivaci networku “Active”. Pro ukončení musí být splněna jedna z následujících podmínek:

1. IO\_DatabusMachine.Read.Action... – tato podmínka bude splněna ve chvíli, kdy operátor stiskne tlačítko pro okamžité ukončení operace robota
2. robot.Aborted – zde bude podmínka splněna v případě, že robot vynutí ukončení operace např. z důvodu poruchy
3. robot.Done a I\_RobotMotion – tato podmínka je splněna v případě, kdy robot dokončil svoji operaci a již není v pohybu

Tímto byl dokončen úkol zavolání konkrétní setup funkce. V další části kapitoly se tedy bude práce zabývat druhým úkolem, a to zpracování a mapování signálů z HMI a řízení HMI v podobě zamykání a odemykání tlačítek a nastavení jejich viditelnosti.

## 5.6 Komunikace s HMI

V této kapitole se bude práce věnovat zpracování komunikace mezi PLC a HMI. V principu je nutné zpracovat požadavek z HMI (stisknutí tlačítka) a zajistit spuštění správné funkce odpovídající požadavku. Aby bylo možné zpracovat signály z HMI, je nejprve potřeba definovat jak vstupní, tak výstupní signály v PLC, které se později propojí se signály s HMI (více v kapitole Programování HMI). Vstupní signály je možné vidět na obrázku č. 62 níže.

HMI_SetupFunctions_I_UDT								
	Name	Data type	Default value	...	...	...	Supervision	Comment
1	Topic_Active	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1: Topic is active
2	MagazineP1_Teachpoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to Magazine teached P1
3	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
4	MagazineP1_Zeropoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to Magazine zero P1
5	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
6	MagazineP2_Teachpoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to Magazine teached P2
7	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
8	MagazineP2_Zeropoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to Magazine zero P2
9	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
10	MagazineP3_Teachpoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to Magazine teached P3
11	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
12	MagazineP3_Zeropoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to Magazine zero P3
13	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
14	Interstation_Teachpoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to teached Interstation point
15	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
16	Interstation_Zeropoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to zero Interstation point
17	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
18	BlowingStation_Teachpoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to teached BlowingStation point
19	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
20	BlowingStation_Zeropoint	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to zero BlowingStation point
21	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value
22	DeliveryPosition	*I_PLCKeyButton_UDT*		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Move to Delivery position
23	RequestValue	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Value

Obrázek 62: UDT pro vstupní signály

V rámci vstupních signálů musely být vytvořeny opět dvě proměnné pro každý bod učení, protože v HMI poté budou dvě tlačítka. Každé tlačítko obsahuje jeden vstupní signál, a to „RequestValue“, který posílá informaci o stisknutí tlačítka z HMI do PLC. Dále je zde také proměnná „TopicActive“, která slouží pro identifikaci, který screen v rámci HMI je právě aktivní.

Výstupní signály jsou vidět na obrázku č. 63 níže (z důvodu přehlednosti nejsou „rozbaleny“ poslední signály).

HMI_SetupFunctions_O_UDT									
	Name	Data type	De...	...	...	...	...	Supervision	Comment
1	RobotState	Int	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0:=NotReady;1:=T1;2:=NotHomed;3:=R
2	MagazineP1_Zeropoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to Magazine zero P1
3	ResponseValue	Bool	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		value
4	Lock	DInt	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Control is locked
5	Visibility	Bool	true	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Control is visible
6	MagazineP1_Teachpoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to Magazine teached P1
7	ResponseValue	Bool	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		value
8	Lock	DInt	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Control is locked
9	Visibility	Bool	true	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Control is visible
10	MagazineP2_Teachpoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to Magazine teached P2
11	ResponseValue	Bool	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		value
12	Lock	DInt	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Control is locked
13	Visibility	Bool	true	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Control is visible
14	MagazineP2_Zeropoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to Magazine zero P2
15	ResponseValue	Bool	false	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		value
16	Lock	DInt	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Control is locked
17	Visibility	Bool	true	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Control is visible
18	MagazineP3_Teachpoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to Magazine teached P3
19	MagazineP3_Zeropoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to Magazine zero P3
20	Interstation_Teachpoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to teached Interstation point
21	Interstation_Zeropoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to zero Interstation point
22	BlowingStation_Teachpoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to teached BlowingStation point
23	BlowingStation_Zeropoint	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to zero BlowingStation point
24	DeliveryPosition	*O_PLCKeyButton_LockInfo_UDT*		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Move to Delivery position

Obrázek 63: UDT pro výstupní signály

Pro každé tlačítko je nutné definovat tři proměnné:

1. ResponseValue – tato proměnná posílá informaci o stisknutí tlačítka z PLC do HMI
2. Lock – tato proměnná slouží pro nastavení, zda bude možné tlačítko stisknout či nikoliv
3. Visibility – tato proměnná určuje, zda-li bude tlačítko viditelné či nikoliv

Dále je zde použita proměnná „RobotState“, která slouží pro získání informace o stavu robota v HMI.

V tuto chvíli byly vytvořeny všechny potřebné proměnné a je možné přejít na programování logiky. V následujících částech bude popsáno, jak byly namapované vstupy a výstupy, následně bude představen samotný network pro jednotlivé setup funkce a poté bude také vysvětleno přiřazení hodnoty ID jednotlivých funkcí.

### Mapování vstupů a výstupů

Zatím nejsou výše vytvořené proměnné nijak propojené s HMI, proto je nutné namapovat vstupy a výstupy, čímž se zajistí, že signály z HMI se uloží do výše uvedených proměnných. Naopak nastavené proměnné v PLC pak budou namapovány na signály do HMI. Mapování vstupů a výstupů probíhá v jedné FC (Function) a konkrétní její upravenou část můžeme vidět na obrázcích č. 64 a 65 níže.

```
"B35_121_OperateAndMonitorTheMachine_ITF".I_HMI.SetupFunctions := "PlasTMouldXpressRobot_HMI".SetupFunctions.I;
```

Obrázek 64: Mapování vstupních signálů

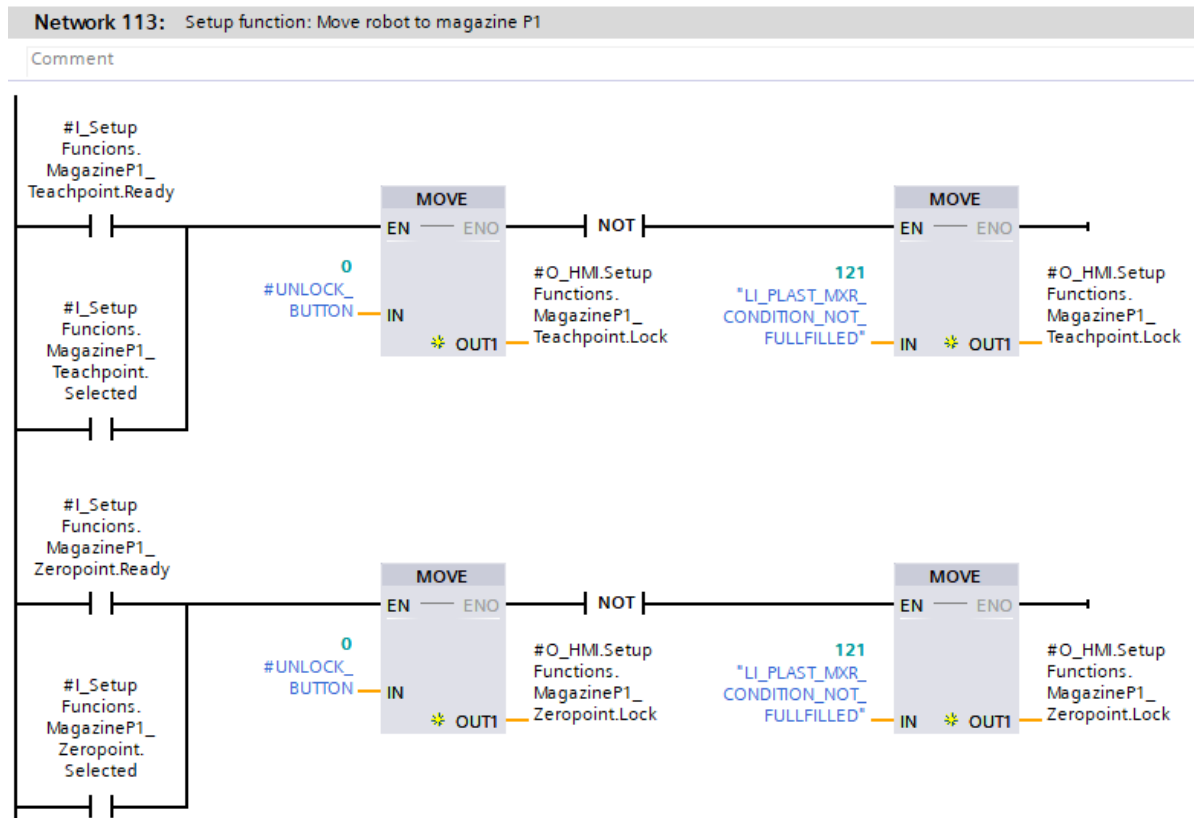
```
"PlasTMouldXpressRobot_HMI".SetupFunctions.O := "B35_121_OperateAndMonitorTheMachine_ITF".O_HMI.SetupFunctions;
```

Obrázek 65: Mapování výstupních signálů

Signály, které si vyměňují HMI a PLC, jsou uloženy v DataBlock „PlasTMouldExpressRobot\_HMI.I/O“ a proměnné používané v PLC jsou uloženy v DB „B36\_121\_OperateAndMonitorTheMachine\_ITF.I/O...“.

### Network pro setup funkce

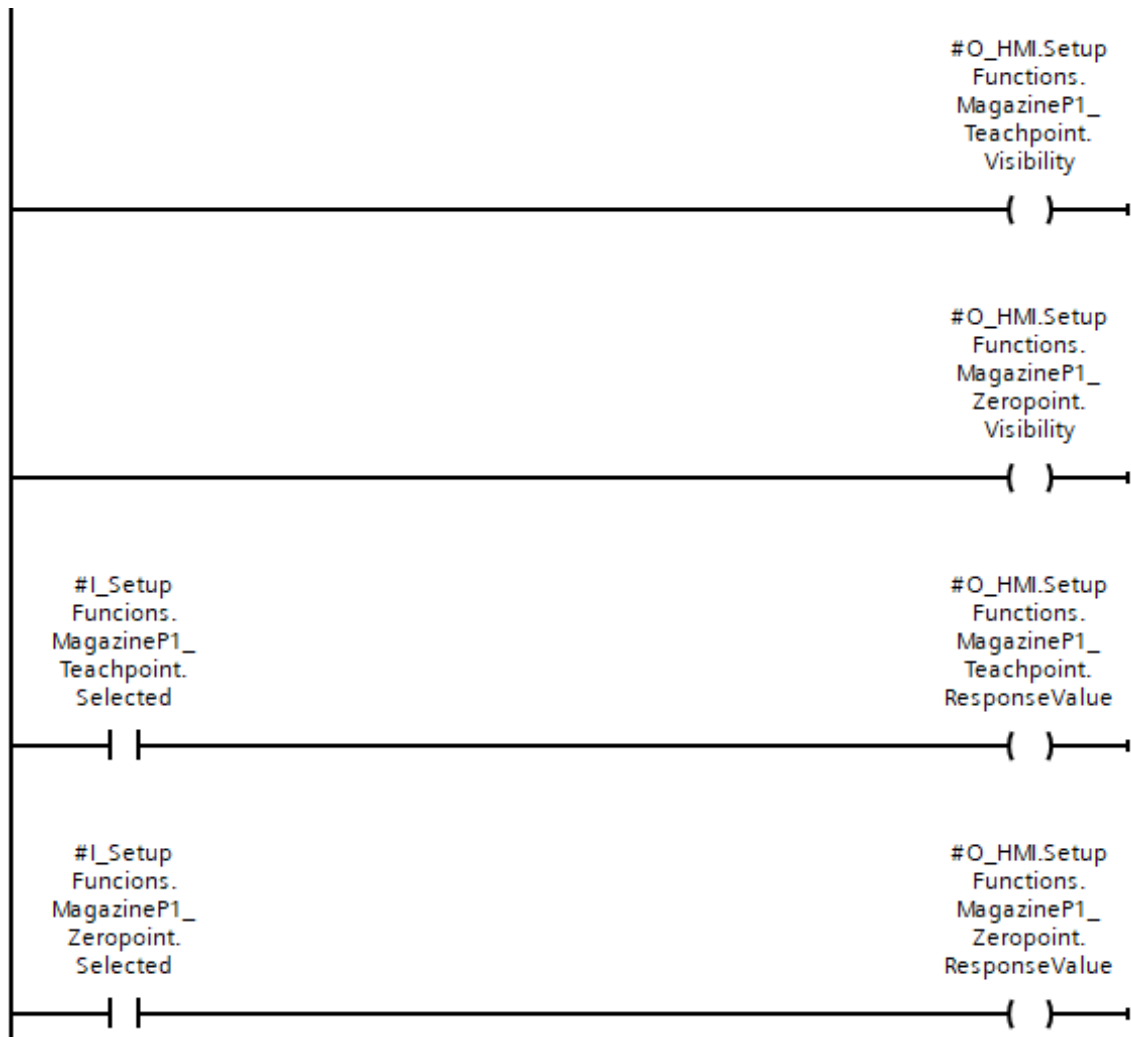
Na následujících řádcích bude představen samotný network pro zpracování signálu. Z důvodu rozsáhlosti networku bude opět rozdělen na více obrázků (č. 66 až č. 68).



Obrázek 66: Network pro zpracování signálu – první část

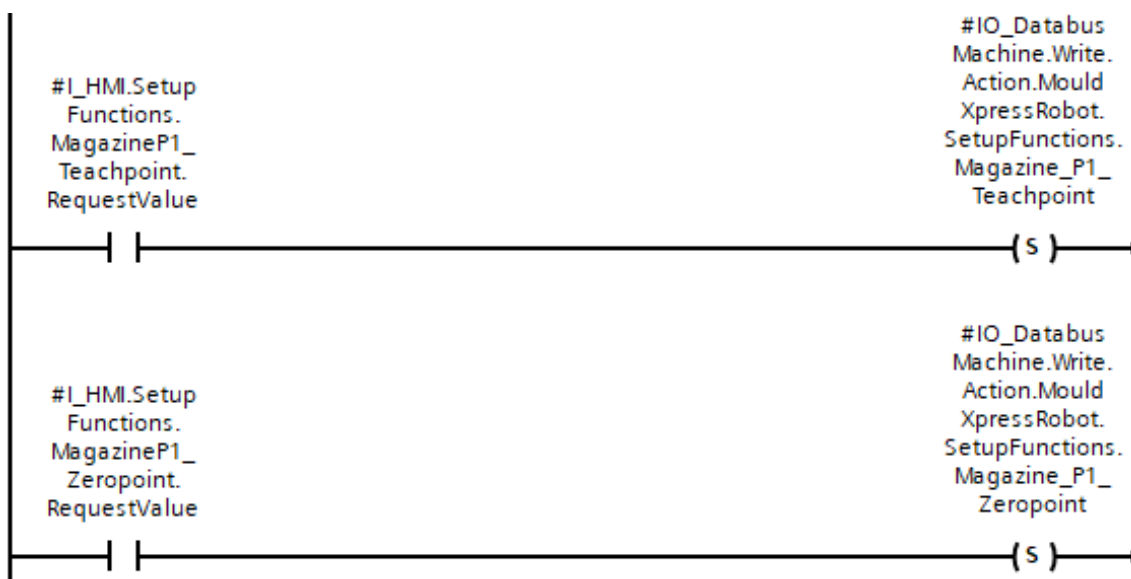
V první části networku (obrázek č. 66) je nastaveno zamknutí tlačítek na základě připravenosti funkcí („I\_SetupFunctions\_... Ready“) nebo zvolení daných funkcí (I\_SetupFunctions\_... Selected). Změna je provedena přiřazením proměnné „UNLOCK\_BUTTON“, která má nulovou hodnotu, tedy tlačítko je možné stisknout. Kromě výše zmíněných podmínek, je v rámci tohoto networku řešena situace, kdy nejsou splněny podmínky pro spuštění funkcí „LI\_PLAST\_MXR\_CONDITION\_NOT\_FULLFILLED“. Poté se přiřadí do proměnné hodnota 121, kdy nebude možné tlačítko stisknout.

Další částí je nastavení viditelnosti tlačítek. Z obrázku č. 67 je patrné, že tlačítka budou vždy viditelná. Pokud je splněna podmínka, že byla vybrána daná funkce, je nastavena hodnota bitu na hodnotu jedna pro „ResponseValue“.



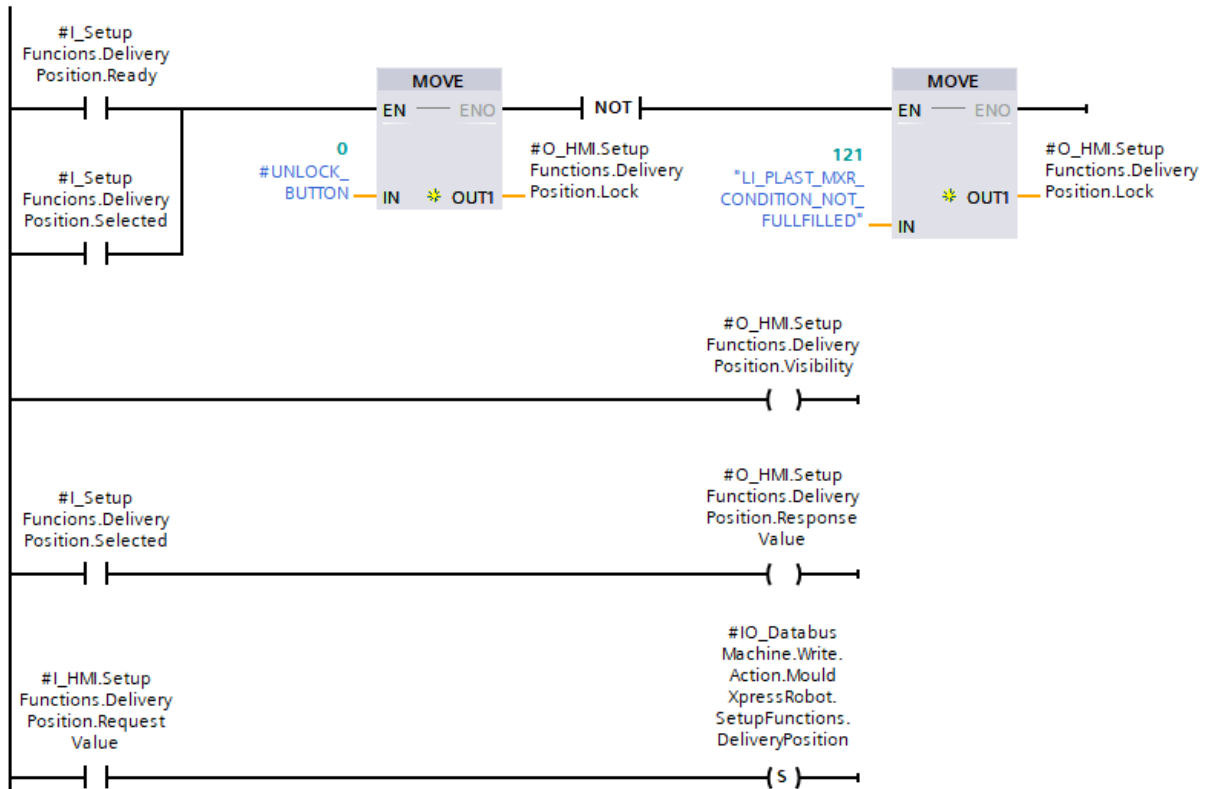
Obrázek 67: Network pro zpracování signálu – druhá část

V poslední části na obrázku č. 68 dochází k nastavení hodnoty bitu „IO\_Databus-Machine.Write.Action...“, která bude potřebná při volání setup funkcí, jelikož je nutnou podmínkou v druhém networku „Request“.



Obrázek 68: Network pro zpracování signálu – třetí část

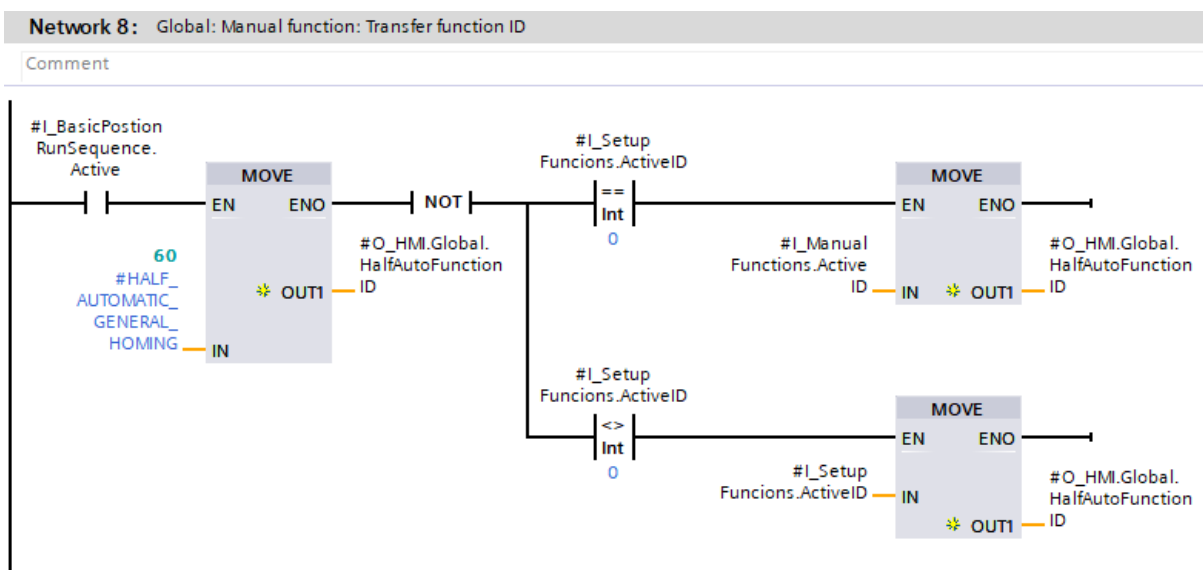
V rámci vizualizace mají všechny teach funkce stejnou strukturu. Avšak není tomu tak u setup funkce „Delivery position“. Zde není třeba rozlišovat „Teachpoint“ a „Zeropoint“, a proto je network jednodušší, viz obrázek 69 níže.



Obrázek 69: Network pro zpracování signálu pro funkci „Delivery position“

### Přiřazení hodnoty ID

Při vizualizaci v HMI bude požadováno zobrazit informaci, která funkce je právě robotem vykonávána. Proto je třeba poslat do HMI informaci v podobě ID dané funkce, kdy ID je přiřazeno v rámci networku „Active“ zmíněné v předchozí podkapitole. Network pro přenos ID je vidět na obrázku č. 70 níže.



Obrázek 70: Přiřazení hodnoty proměnné ID

Přiřazení ID funguje tak, že pokud dochází k pohybu robota do domácí polohy, na HMI se zobrazí ID dané funkce. V druhé části networku dochází k přiřazení ID zvolené funkce do HMI. Toto nastavení probíhá na základě podmínky, která vyhodnocuje hodnotu proměnné "ActiveID", která byla popsána v předchozí podkapitole. Toto řešení bylo zvoleno z důvodu, že pro různé sety funkcí (manuální a setup funkce) existují v PLC dvě různé proměnné, ve kterých je uložena aktuální hodnota ID zvolené funkce. Avšak v HMI je pro tento účel proměnná pouze jedna. Proto zde bude podmínka vyhodnocena tak, že pokud není zvolena žádná setup funkce (hodnota aktivního ID je rovna nule), uloží se do proměnné "O\_HMI.Global..." (podle které se řídí HMI) hodnota aktivní manuální funkce. Toto je platné i pro situaci, kdy není vybrána žádná manuální funkce. V opačném případě, tedy je zvolena jedna z teach funkcí, uloží se do stejné proměnné hodnota ID právě zvolené funkce. Samotná práce s touto proměnnou v rámci programování HMI bude popsána v následující kapitole.

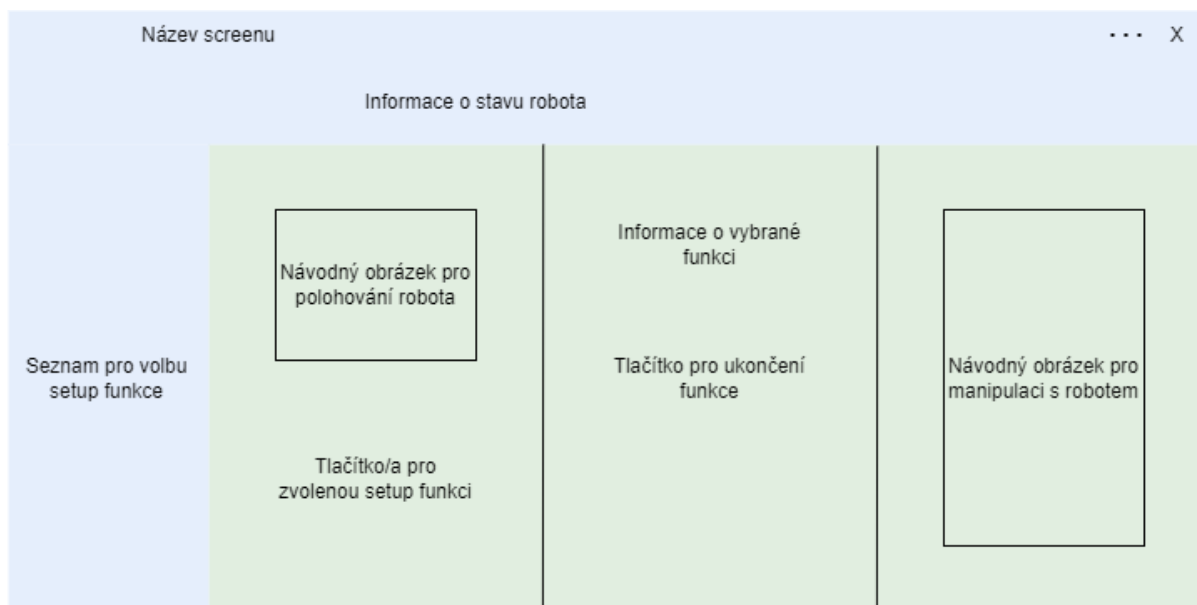


## 6 Programování HMI

Posledním úkolem bylo rozšířit HMI, přes které bude možné s robotem manipulovat. Pro návrh HMI byl použit software VisiWin, ve kterém je již vizualizace pro manipulaci se strojem a robotem vytvořena. Protože je celé HMI vytvořeno v německém jazyce, bude i jeho rozšíření, naprogramované v rámci této práce, navrženo v německém jazyce. Rozšíření HMI bude zpracováno v podobě screenu, který bude obsahovat možnost překliknutí mezi jednotlivými setup funkcemi. Na displeji budou zobrazena pouze tlačítka týkající se vybrané funkce. Pro každou teach funkci pak bude možnost stisknout dvě různá tlačítka odpovídající dvěma funkcím v PLC (Teachpoint, Zeropoint). Pro funkci „Delivery position“ bude zobrazeno pouze jedno tlačítko, pomocí kterého se robot přesune do dané polohy. Pro všechny funkce pak bude také vytvořeno tlačítko pro ukončení dané operace.

### 6.1 Návrh screenu

Pro zpracování screenu byl použit template, u kterého byly provedeny úpravy tak, aby výsledná funkcionalita odpovídala požadavkům. Název screenu je „Formwechselroboter positionieren“, neboli polohování robota na výměnu forem. Pro upřesnění se screen nachází ve stavu, kdy, již bylo stisknuto jedno z tlačítek, a tedy lze v tomto stavu demonstrovat všechny části screenu. Popis jednotlivých komponent bude uveden níže. Pro snazší orientaci bude v této práci uveden layout výsledného screenu, který je vidět na obrázku č. 71 níže.



Obrázek 71: Layout navrženého screenu

Screen se skládá ze dvou hlavních částí, hlavního (řídícího; v layoutu podbarven modře) a vedlejšího (detailního; v layoutu podbarven zeleně) screenu. Tyto screeny budou v rámci práce nazývány jako Master a Detail screen a budou podrobněji popsány v následujících podkapitolách. Detail screen je pak dále rozdělen pro zvýšení přehlednosti na tři sloupce.

Výsledný screen je pak možné vidět na obrázku č. 72 níže.



Obrázek 72: Vytvořený screen

## 6.2 Screen Master

Hlavním funkčním bodem Master screenu je seznam nalevo, pomocí kterého může uživatel vybírat jednotlivé funkce. Pro tento účel bylo potřeba vytvořit interní proměnnou „SetupFunctions\_MajorJobSelected“, přes kterou je možné propojit tlačítka ze seznamu se screeny jednotlivými funkcí. Názvy jednotlivých funkcí jsou definovány v rámci uživatelských textů viz obrázek č. 73. Definici seznamu je možné vidět na obrázku č. 74 níže, kdy byla použita struktura TabControl. Překlad jednotlivých bodů je uveden v následující tabulce č. 8:

Name	Description	German (Germany) (1031)
1		Magazin Punkt 1
2		Magazin Punkt 2
4		Magazin Punkt 3
8		Zwischenstation
16		Blasstation
32		Auslieferungs Position

Obrázek 73: Definování uživatelských textů

Tabulka 8: Překlad uživatelských textů

Název německy	Název česky
Magazin Punkt 1	Bod 1 na zásobníku
Magazin Punkt 2	Bod 2 na zásobníku
Magazin Punkt 3	Bod 3 na zásobníku
Zwischenstation	Mezistanice
Blasstation	Vyfukovací stanice
Auslieferungs Position	Přepravní poloha (pozice)

Jak již bylo zmíněno na začátku praktické části, ne každé pracoviště má mezistanici pro odložení formy. To je zde vyřešeno viditelností příslušného tlačítka, tedy vidět tlačítko mezistanice bude možné pouze tehdy, má-li příslušný bit hodnotu 1. Konkrétní pracoviště, na

kterém byla práce zpracována, mezistanici neobsahuje, a proto není možné tlačítko mezistanice zvolit, jak je vidět v levé části obrázku č. 72 výše.

```
<!-- ##### Tab Area ##### -->
<hmi:TabControl Grid.Row="1" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" TabOrientation="Vertical">

    <hmi:TabItem LocalizableHeader="@${Topics.SetupFunctions.MajorJobs.1} ActiveVariableName="${SetupFunctions_MajorJobSelected}">
        <hmi:Include Margin="50,40,50,60" ViewName="SetupFunctions_Detail_View"/>
    </hmi:TabItem>

    <hmi:TabItem LocalizableHeader="@${Topics.SetupFunctions.MajorJobs.2} ActiveVariableName="${SetupFunctions_MajorJobSelected} ActiveBitNumber="1">
        <hmi:Include Margin="50,40,50,60" ViewName="SetupFunctions_Detail_View"/>
    </hmi:TabItem>

    <hmi:TabItem LocalizableHeader="@${Topics.SetupFunctions.MajorJobs.4} ActiveVariableName="${SetupFunctions_MajorJobSelected} ActiveBitNumber="2">
        <hmi:Include Margin="50,40,50,60" ViewName="SetupFunctions_Detail_View"/>
    </hmi:TabItem>

    <hmi:TabItem LocalizableHeader="@${Topics.SetupFunctions.MajorJobs.8} ActiveVariableName="${SetupFunctions_MajorJobSelected} ActiveBitNumber="3"
        Visibility="{vw:VariableBinding VariableName=${VC1$.${PlasTMouldXpressRobot_HMI.Global.O{.InterstationAvailable}}}">
        <hmi:Include Margin="50,40,50,60" ViewName="SetupFunctions_Detail_View"/>
    </hmi:TabItem>

    <hmi:TabItem LocalizableHeader="@${Topics.SetupFunctions.MajorJobs.16} ActiveVariableName="${SetupFunctions_MajorJobSelected} ActiveBitNumber="4">
        <hmi:Include Margin="50,40,50,60" ViewName="SetupFunctions_Detail_View"/>
    </hmi:TabItem>

    <hmi:TabItem LocalizableHeader="@${Topics.SetupFunctions.MajorJobs.32} ActiveVariableName="${SetupFunctions_MajorJobSelected} ActiveBitNumber="5">
        <hmi:Include Margin="50,40,50,60" ViewName="SetupFunctions_Detail_View"/>
    </hmi:TabItem>
</hmi:TabControl>
```

Obrázek 74: Struktura „TabControl“

Dalším požadavkem rozšíření funkcionality bylo, aby tento screen mohli zobrazit (a použít) pouze oprávněné osoby. Pro tyto účely slouží tzv. skupiny uživatelů, kterých existuje několik a každá skupina má jiná přístupová práva. Příkladem je operátor, který má méně práv než například administrátor. Navržené funkce slouží pouze k učení robota, což je proces, který musí provést pouze kvalifikovaný uživatel. Proto je pro tento screen nastaven přístup skupině uživatelů „Admin“. Ve VisiWin funguje nastavení práv tak, že dané právo má zvolená skupina (v tomto případě administrátor) a skupiny s vyššími pravomocemi (v tomto případě pouze uživatel skupiny „Krones“). Ostatní skupiny („Nepřihlášený uživatel“, „Operátor“ a „Expert“) nebudou mít k tomuto screenu přístup. U nastavení práv se také uvádí vykonávaná činnost, zde „Control production/process“, neboli obsluha produkce/procesu. Definování práv je možné vidět na obrázku č. 75 níže.

```
AuthorizationRight="Control production/process - Admin"
```

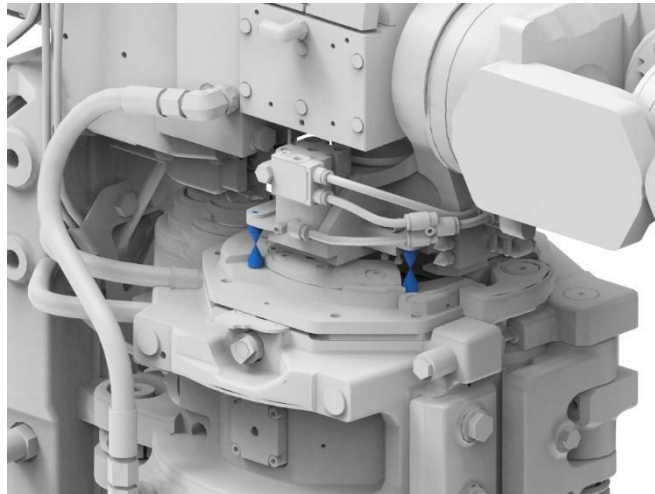
Obrázek 75: Přřazení administrátorských práv

Samotné přihlašování uživatele funguje pomocí tzv. „tokenu“ či „transponderu“. Token neboli transponder je část plastu oválného tvaru obsahující integrovaný čip s nahrenými přihlašovacími údaji vlastníka tokenu. Po přiložení tokenu na příslušné místo na HMI panelu dojde k automatickému přihlášení přiřazené skupině uživatelů (například zaměstnanec s tokenem s právy operátora se po přiložení tokenu přihlásí jako operátor).

### 6.3 Screen Detail

Detailní screen byl nazván „SetupFunction\_Detail\_View“ a skládá se ze tří sloupců. V levém sloupci je seznam tlačítek pro jednotlivé funkce. Pro seznam propojený se strukturou TabControl na Master screenu zde byla využita jednoduchá struktura „ScrollView“, která umožňuje (jak název napovídá) pomocí scrollování zobrazit více obsahu, než je možné zobrazit na obrazovce najednou. Jednotlivé prvky pak budou propojeny přes parametr „Visibility“, tedy každý prvek bude zobrazen jen tehdy, stiskne-li operátor příslušnou funkci ze struktury TabControl. Nad tlačítky se ještě nachází obrázek, který napovídá uživateli, jak robota pro daný bod napolohovat. Obrázek je zobrazen v tzv. Pictureboxu, a je opět propojen přes proměnnou „SetupFunctions\_MajorJobSelected“, kdy hodnoty odpovídají hodnotám definovaným pro

jednotlivé funkce dle obrázku č. 73. Příklad návodného obrázku pro bod Blowingstation je vidět na obrázku č. 76.



Obrázek 76: Návodný obrázek pro polohování robota [37]

Program pro funkci návodných obrázků je vidět na obrázku č. 77 níže.

```
<hmi:PictureBox VariableName="$1$SetupFunctions_MajorJobSelected" Height="263" Margin="0,0,0,16">
  <hmi:PictureBox.SymbolList>
    <hmi:SymbolState Value="0"/>
    <hmi:SymbolState Value="1" Symbol="{DynamicResource Magazine_Userframe_Point1_3}"/>
    <hmi:SymbolState Value="2" Symbol="{DynamicResource Magazine_Userframe_Point2_3}"/>
    <hmi:SymbolState Value="4" Symbol="{DynamicResource Magazine_Userframe_Point3_3}"/>
    <hmi:SymbolState Value="8" Symbol="{DynamicResource Interstation_Userframe}"/>
    <hmi:SymbolState Value="16" Symbol="{DynamicResource BlowingStation_Userframe}" />
    <hmi:SymbolState Value="32" Symbol="{DynamicResource TeachPendant_4DGraphics}"/>
  </hmi:PictureBox.SymbolList>
</hmi:PictureBox>
```

Obrázek 77: Implementace návodných obrázků

Pro implementaci každé setup funkce byl vytvořen separátní malý screen (viz obrázek č. 78), kterými se poté naplní seznam. Tento screen je možné v rámci layoutu (obrázek č. 71) najít v levém sloupci detail screenu (zelené podbarvení) dole.



Obrázek 78: Tlačítka pro ovládání robota

Každý malý screen je také propojen se seznamem v rámci Master screenu, viz obrázek č. 79, tak, aby byla viditelná pouze ta tlačítka, která odpovídají zvolené funkci.

```
<StackPanel>
  <hmi:Include ViewName="MagazineP1_View" Name="MagazineP1"
  Visibility="{vw:VariableBinding VariableName=$1$TeachFunctions_MajorJobSelected{[0]}}
  <hmi:Include ViewName="MagazineP2_View" Name="MagazineP2"
  Visibility="{vw:VariableBinding VariableName=$1$TeachFunctions_MajorJobSelected{[1]}}
  <hmi:Include ViewName="MagazineP3_View" Name="MagazineP3"
  Visibility="{vw:VariableBinding VariableName=$1$TeachFunctions_MajorJobSelected{[2]}}
  <hmi:Include ViewName="InterStation_View" Name="InterStation"
  Visibility="{vw:VariableBinding VariableName=$1$TeachFunctions_MajorJobSelected{[3]}}
  <hmi:Include ViewName="BlowingStation_View" Name="BlowingStation"
  Visibility="{vw:VariableBinding VariableName=$1$TeachFunctions_MajorJobSelected{[4]}}
</StackPanel>
```

Obrázek 79: Propojení Master a Detail screenu

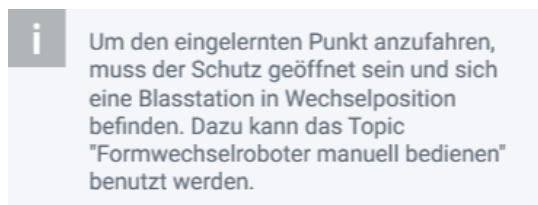
Pro teach funkce jsou k dispozici dvě tlačítka, viz obrázek 78. První tlačítko nese nápis „Hilfspunkt anfahren“, volným překladem “najat do pomocného bodu”. Tedy po stisknutí tohoto tlačítka se robot napoložuje do předem definovaného bodu. Druhým tlačítkem je „Eingelernten Punkt anfahren“, volným překladem “najat do naučeného bodu”. Po stisknutí tohoto tlačítka se robot napoložuje do již naučeného bodu.

Tlačítka jsou propojena s proměnnými v PLC, které byly definovány v rámci deklarace datových typů v „HMI\_SetupFunctions\_I\_UDT“ a „HMI\_SetupFunctions\_O\_UDT“ (viz podkapitola 3.6 Komunikace s HMI). Příklad naprogramování tlačítka pro bod Magazine P1 lze vidět na obrázku č. 80 níže.

```
<StackPanel>
  <hmi:PLCKeyButton ResponseVariableName="$VC1$. $1$PlasTMouldXpressRobot_HMI.SetupFunctions.O{.MagazineP1_Zeropoint.ResponseValue}"
  hmi:Lock.VariableName="$VC1$. $1$PlasTMouldXpressRobot_HMI.SetupFunctions.O{.MagazineP1_Zeropoint.Lock}"
  hmi:Lock.TextGroup="@§1$General.LockInfo"
  Visibility="{vw:VariableBinding VariableName=$VC1$. $1$PlasTMouldXpressRobot_HMI.SetupFunctions.O{.MagazineP1_Zeropoint.Visibility}}"
  LocalizableText="@§1$Topics.SetupFunctions.MinorJobs.52"
  RequestVariableName="$VC1$. $1$PlasTMouldXpressRobot_HMI.SetupFunctions.I{.MagazineP1_Zeropoint.RequestValue}"/>
  <hmi:PLCKeyButton ResponseVariableName="$VC1$. $1$PlasTMouldXpressRobot_HMI.SetupFunctions.O{.MagazineP1_Teachpoint.ResponseValue}"
  hmi:Lock.VariableName="$VC1$. $1$PlasTMouldXpressRobot_HMI.SetupFunctions.O{.MagazineP1_Teachpoint.Lock}"
  hmi:Lock.TextGroup="@§1$General.LockInfo"
  Visibility="{vw:VariableBinding VariableName=$VC1$. $1$PlasTMouldXpressRobot_HMI.SetupFunctions.O{.MagazineP1_Teachpoint.Visibility}}"
  LocalizableText="@§1$Topics.SetupFunctions.MinorJobs.51"
  RequestVariableName="$VC1$. $1$PlasTMouldXpressRobot_HMI.SetupFunctions.I{.MagazineP1_Teachpoint.RequestValue}"/>
</StackPanel>
```

Obrázek 80: Definování tlačítek pro ovládání robota

Screen pro bod Blowingstation je dále rozšířen o následující informační text (obrázek č. 81), jehož volný překlad je následující: „Pro najetí do naučeného bodu je potřeba, aby byly otevřené ochranné dveře a vyfukovací stanice se nacházela v pozici pro výměnu formy. K tomu je možné využít screen Formwechselroboter manuell bedienen“ (“Manuální obsluha robota pro výměnu forem”). Tento text platí pouze pro naučený bod, který se nachází nad formou (viz obrázek 76), tedy uvnitř stroje. Předem definovaný bod se nachází vně stroje, proto tento text není smysluplný.



Obrázek 81: Pomocný text pro bod „Blowing station“

Poslední setup funkce pro Delivery position obsahuje pouze jedno tlačítko (viz obrázek č. 82), a to „Position anfahren“, neboli najet na pozici. Zde se žádný bod neučí, proto jedno tlačítko dostačující.



Obrázek 82: Tlačítko pro polohování robota do přepravní pozice

Prostřední sloupec screenu je pro všechny funkce stejný. Nachází se zde jak tlačítko pro ukončení dané operace („Abbrechen“), tak informace o současně zvolené funkci. Tato informace je ve formátu „MajorJob : MinorJob“, neboli informace před dvojtečkou odpovídá zvolené funkci v rámci Master screenu a za dvojtečkou je konkrétní zvolená funkce po stisknutí tlačítka v levém sloupci detail screenu, nebo informace o tom, že žádná funkce nebyla vybrána. Oba případy lze vidět na obrázcích níže, kdy na obrázku č. 83 je uvedena situace, kdy v rámci Master screenu byla vybrána teach funkce pro bod Magazine P1, ale nebyla vybrána funkce v rámci Detail screenu. Zároveň je možné vidět, že v tuto chvíli jsou funkce připraveny.



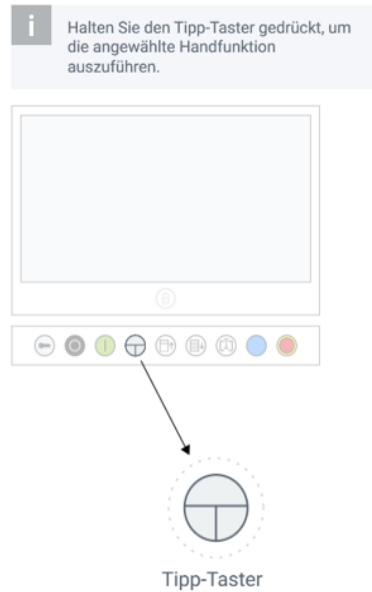
Obrázek 83: Detail screen – bez zvolené funkce

Na obrázku č. 84 byla již zvolena funkce i v rámci Detail screenu, o čemž je uživatel informován, a zároveň již není možné žádnou funkci zvolit.



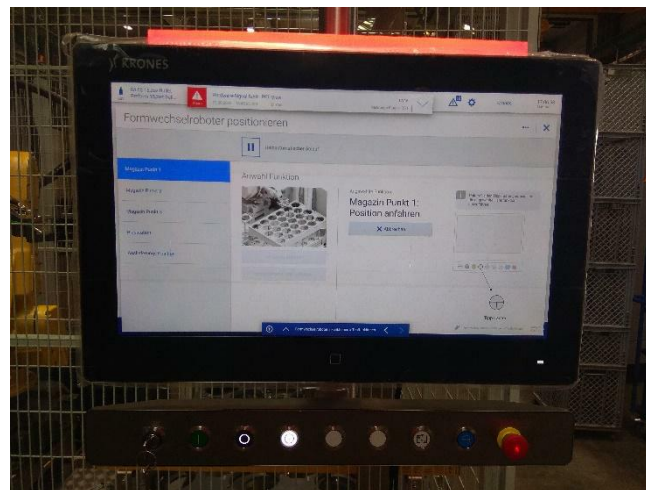
Obrázek 84: Detail screen – zvolena funkce

V posledním sloupci screenu se pak nachází nápověda pro uživatele. V souvislosti s bezpečností je třeba, aby uživatel držel stisknuté konkrétní tlačítko (viz obrázek č. 85) po celou dobu, co robot vykonává zvolenou operaci. Pokud by uživatel tlačítko pustil, proces by se zastavil.



Obrázek 85: Pomocný obrázek pro provedení akce [37]

Kromě obrázku je v rámci nápovědy i uveden informační text, který lze volně přeložit jako: „Pro vykonání zvolené funkce je třeba držet stisknuté tlačítko Tipp-Taster“. Toto tlačítko navíc ve chvíli, kdy je funkce zvolena, bliká. Což je znázorněno na obrázku č. 86 níže.



Obrázek 86: Fotografie celého HMI

## 7 Přínosy navrženého řešení

V této kapitole se bude text věnovat přínosům, které navržené a v předchozích kapitolách popsané řešení přineslo.

Hlavním přínosem práce bylo zjednodušení celého procesu učení robota. Postup učení robota byl popsán opět v kapitole Analýza současného stavu. Nevýhodou výchozího stavu byla nutnost užití teach pendantu, ve kterém bylo třeba najít a zavolat konkrétní funkci pro učení robota. S ohledem na množství funkcí v robotu nebylo efektivní funkci hledat a následně danou funkci zavolat (Standartní postup spuštění funkce pomocí teach pendantu, tedy podržet Deadman tlačítko, stisknout RESET, držet SHIFT a následně teprve stisknout tlačítko FWD pro spuštění funkce). Proto bylo užitečné tento proces zjednodušit pomocí HMI, kdy uživatel pouze vybere danou funkci v seznamu (na rozdíl od pendantu jsou na daném screenu pouze setup funkce, kdy všechny jsou dosažitelné bez nutnosti scrollování), stiskne tlačítko odpovídající funkce a podrží tlačítko „Tipp-Taster“ (stejná funkce jako Deadman). Tímto způsobem se minimalizuje počet kroků nutných k dosažení požadované funkcionality. Kromě toho není třeba interagovat s teach pendantem, což je rychlejší a zároveň uživatelsky přívětivější. Potenciálně lze tak dosáhnout snížení času na zaškolení budoucího pracovníka.

Dalším výchozím nedostatkem v rámci procesu učení robota byla nemožnost zkontrolovat, zdali již naučený bod odpovídá aktuálnímu stavu pracoviště (například po výměně zásobníku). Jedinou možností kontroly bylo manuálně robota napolohovat do pozice dle souřadnic uložených v PLC (nutné zobrazit pomocí speciálního dalšího screenu v HMI). Proto byly rozšířeny funkce robota o možnost najet do již naučených pozic (stále uložených v PLC). Tím pak bylo možné opět pomocí HMI zvolit nájezd do naučených bodů bez nutnosti interakce s teach pendantem a k tomu dalším screenem v HMI. Lze tak snadno ověřit validitu souřadnic a ušetřit čas pro další učení (pokud bude třeba).

V rámci práce byla také přidána možnost volat kromě samotných teach funkcí i funkci pro uvedení robota do přepravní pozice. Zde bylo třeba vytvořit úplně novou funkci pro tuto pozici. V rámci testování bylo navíc zjištěno, že přepravní pozice není dosažitelná ze všech pozic, kde se může robot nacházet. Proto by mohlo dojít při vůli uvést robota do této polohy ke kolizi s předměty na pracovišti. Z tohoto důvodu bylo třeba kolizi zamezit tak, že bylo na základě testování zvoleno řešení, při kterém je robot nejprve uveden do domácí polohy, která je na rozdíl od přepravní pozice dosažitelná ze všech poloh robota, aniž by byla možná kolize robota s jiným objektem. To vše opět bylo možné pomocí tlačítka v HMI bez interakce s teach pendantem.

Práce probíhala v rámci oddělení vývoje společnosti. Jak již bylo zmíněno v rámci kapitoly Analýza současného stavu, společnost Kronos AG je dodavatelem strojů a zařízení v oblasti nápojového průmyslu. Zákazníci společnosti jsou firmy, které se zabývají výrobou zboží pro koncové zákazníky. Navržené řešení tedy není pouze řešením pro jedno konkrétní pracoviště (na kterém byla práce poprvé implementována a otestována), ale pro všechna pracoviště, která budou řešit výměnu forem pomocí průmyslového robota.

Pro vyhodnocení této práce to tak znamená, že neexistují kvantifikovatelná data pro zhodnocení přínosů práce. Z pohledu průmyslového inženýrství práce optimalizovala proces učení robota, který je součástí přípravy výroby, a tedy lze přínos práce definovat jako zkrácení přípravného času. Ovšem jak již bylo naznačeno v předchozím odstavci, nelze stanovit „o kolik“ se čas zkrátil, a to z několika důvodů. V každé společnosti, kde bude řešení implementováno, bude jiná obsluha. Zároveň je možné předpokládat, že jednotlivé firmy budou mít různě stanovené časové normy pro tuto část přípravy výroby.



Dalším důvodem je modularita pracoviště. Vyfukovací stanice, tedy stroj, ve kterém se lahve vyfukují, je dodáván ve více variantách a liší se jak v počtu forem, které stroj obsahuje, tak v rychlosti výroby (například od 8 tisíc lahví za hodinu až po 100 tisíc lahví za hodinu). Kromě tohoto nelze také stanovit koncovou cenu výrobků, které se na dané lince u konkrétního zákazníka budou vyrábět.

Obecně lze tedy říci, že práce přinesla redukci interakce s teach pendantem při zachování a zároveň rozšíření stávajících funkcí. Operace týkající se setup funkcí, tedy učení robota a uvedení robota do přepravní polohy, byly přeneseny do nového screenu HMI, které je uživatelsky přívětivější a umožnilo automatizaci (redukci) některých nyní již nadbytečných kroků, čímž se proces učení optimalizoval.

## Závěr

V první kapitole věnované tématu automatizace byly představeny základy programovatelných automatů včetně jejich způsobu fungování a jejich základního rozdělení. Poté byla část textu věnována programování PLC, programovacím jazykům, datovým typům a základním logickým kombinačním funkcím. V druhé kapitole se práce věnovala oblasti robotizace se zaměřením na průmyslové roboty, které byly také mimo jiné položeny do kontextu s kolaborativní robotikou. Bylo také představeno několik výrobců průmyslových robotů.

Práce se dále zabývala analýzou současného stavu pracoviště, na kterém byla zpracována praktická část diplomové práce. Kromě samotného pracoviště byla představena společnost Kronos AG, ve které byla práce vypracována. V rámci popisu pracoviště byl představen i řídicí systém tvořený třemi základními prvky. Na závěr této kapitoly byl popsán proces učení robota. Zároveň byl vytyčen cíl diplomové práce, a to právě optimalizace a zároveň zjednodušení tohoto procesu.

V druhé části diplomové práce byly popsány jednotlivé kroky, které bylo nutné provést pro splnění požadované funkcionality. V první kapitole se práce věnovala programování robota. Zde bylo potřeba rozšířit stávající funkce o zpracování požadavku z PLC v podobě čísel volaných funkcí a souřadnic, na které má robot najet. V druhé kapitole se práce zaměřila na programování PLC. PLC slouží na pracovišti v podobě řídicího prvku, a zároveň také jako prvek propojující uživatelské rozhraní s robotem. Proto zde bylo nutné zpracovat signály přicházející z HMI. Pomocí logiky byla následně určena data, která byla posléze odeslána do controlleru robota, aby robot vykonal požadovanou funkci. Následně bylo popsáno grafické rozhraní HMI, pomocí kterého je možné snadno robota ovládat. Byl popsán výsledný screen spolu se všemi prvky, jakými byla například tlačítka, uživatelské texty, seznamy pro výběr a další.

V rámci poslední kapitoly byly shrnuty přínosy, které vyplívají z této práce. Diplomová práce splnila vytyčený cíl a zároveň byla úspěšně dokončena a připravena pro účely firmy. Na závěr by bylo vhodné zmínit, že práce probíhala v rámci oddělení vývoje, a proto výše popsané řešení není řešením pouze jednoho konkrétního pracoviště, ale všech, na kterých se bude nacházet průmyslový robot pro automatizovanou výměnu forem.

## Seznam použité literatury

- [1] KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016. ISBN 978-80-214-4828-5.
- [2] NOVOTNÝ, František. *Úvod do automatizace a robotizace ve strojírenství*. Liberec: Technická univerzita v Liberci, 2020. ISBN 978-80-7494-545-8.
- [3] CHLEBNÝ, Jan a kol. *Automatizace a automatizační technika: Prostředky automatizační techniky*. Brno: Computer Press, 2014. ISBN 978-80-251-3747-5.
- [4] CHLEBNÝ, Jan a kol. *Automatizace a automatizační technika: Systémové pojetí automatizace*. 1. vyd. Brno: Computer Press, 2012. 217 s. ISBN 978-80-251-3628-7.
- [5] Průmysl 4.0 historie [online]. [cit. 2023-11-23]. Dostupné z: [https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumysloverevoluce-ke-4\\_31001.html](https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumysloverevoluce-ke-4_31001.html).
- [6] SHERWANI, F.; ASAD, Muhammad Mujtaba a IBRAHIM, B.S.K.K. Collaborative Robots and Industrial Revolution 4.0 (IR 4.0). Online. In: *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*. IEEE, 2020, s. 1-5. ISBN 978-1-7281-7113-5. [cit. 2023-11-23]. Dostupné z: <https://doi.org/10.1109/ICETST49965.2020.9080724>.
- [7] SUJITHA, S; S, Vinod Kumar; V S, Dechamma; LIKITHA, J; R M, Prajwal et al. Study of Interfacing PLC With HMI for Industrial Applications. Online. In: *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*. IEEE, 2023, s. 343-346. ISBN 979-8-3503-4664-0. [cit. 2023-11-23]. Dostupné z: <https://doi.org/10.1109/ICEARS56392.2023.10084927>.
- [8] SAHEB, Shaik Himam; BABU, G.Satish a RAJU, N.V.Subba. Relative Kinematic Analysis of Serial and Parallel Manipulators. Online. *IOP Conference Series: Materials Science and Engineering*. 2018, roč. 455. ISSN 1757-899X. [cit. 2023-11-23]. Dostupné z: <https://doi.org/10.1088/1757-899X/455/1/012040>.
- [9] ABDELAAL, Mahmoud. A Study of Robot Control Programing for an Industrial Robotic Arm. Online. In: *2019 6th International Conference on Advanced Control Circuits and Systems (ACCS) & 2019 5th International Conference on New Paradigms in Electronics & information Technology (PEIT)*. IEEE, 2019, s. 23-28. ISBN 978-1-7281-6354-3. [cit. 2023-11-23]. Dostupné z: <https://doi.org/10.1109/ACCS-PEIT48329.2019.9062878>.
- [10] KURFESS, Thomas R. *Robotics and automation handbook*. Boca Raton: CRC Press, 2005. ISBN 0-8493-1804-1.
- [11] HULL, Tina a MINARCIN, Monika A. Considerations in Collaborative Robot System Designs and Safeguarding. Online. *SAE International Journal of Materials and Manufacturing*. 2016, roč. 9, č. 3, s. 545-551. ISSN 1946-3987. [cit. 2023-11-23]. Dostupné z: <https://doi.org/10.4271/2016-01-0340>.

- [12] Modulární PLC. In: AutomationForum.co. Online. [cit. 2023-11-23]. Dostupné z: <https://cdn.automationforum.co/uploads/2019/02/Typical-Modular-PLC.png>.
- [13] Kompaktní PLC. In: AutomationForum.co. Online. [cit. 2023-11-23]. Dostupné z: [https://assetcloud.roccommerce.net/w458-h458-cpad/\\_smcelectric/7/5/4/allen\\_bradley\\_5069\\_1306ers2.jpg](https://assetcloud.roccommerce.net/w458-h458-cpad/_smcelectric/7/5/4/allen_bradley_5069_1306ers2.jpg)
- [14] SIEMENS. *Programming Guideline for S7-1200/1500*. Online. 2018. [cit. 2023-11-23]. Dostupné z: [https://cache.industry.siemens.com/dl/files/040/90885040/att\\_970576/v1/81318674\\_Programming\\_guideline\\_DOC\\_v16\\_en.pdf](https://cache.industry.siemens.com/dl/files/040/90885040/att_970576/v1/81318674_Programming_guideline_DOC_v16_en.pdf).
- [15] FANUC. Online. [cit. 2023-11-23]. Dostupné z: <https://www.fanuc.eu/cz/cs>.
- [16] Sériový robot. In: FANUC. Online. [cit. 2023-11-23]. Dostupné z: <https://www.fanuc.eu/~media/corporate/products/robots/m20/m-20id-12l/m20id12l.png>
- [17] Paralelní robot. In: FANUC. Online. [cit. 2023-11-23]. Dostupné z: <https://www.fanuc.eu/~media/corporate/products/robots/delta/m2/m2-6h/ro-pr-m2ia6h-rgb.jpg>
- [18] FANUC M-710iC/50S. Online. [cit. 2023-11-23]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/stránka-filtru-robotů/řada-m-710/m-710ic-50s>.
- [19] Průmyslový robot FANUC. In: FANUC. Online. [cit. 2023-11-23]. Dostupné z: <https://www.fanuc.eu/~media/corporate/products/robots/m2000/m2000ia1700/fea-ro-pr-m20001700-1-1.jpg>.
- [20] ABB. Online. [cit. 2023-11-23]. Dostupné z: <https://global.abb/group/en>.
- [21] Průmyslový robot ABB. In: ABB. Online. [cit. 2023-11-23]. Dostupné z: [https://cdn.productimages.abb.com/9IBA033547\\_400x400.jpg](https://cdn.productimages.abb.com/9IBA033547_400x400.jpg).
- [22] KUKA. Online. [cit. 2023-11-23]. Dostupné z: <https://www.kuka.com>.
- [23] Průmyslový robot KUKA. In: KUKA. Online. [cit. 2023-11-23]. Dostupné z: <https://www.kuka.com/~media/kuka-corporate/images/products/robots/cta-images/kr-titan.png>.
- [24] Yaskawa. Online. [cit. 2023-11-23]. Dostupné z: <https://www.yaskawa.eu.com>.
- [25] Průmyslový robot Yaskawa. In: Yaskawa. Online. [cit. 2023-11-23]. Dostupné z: [https://www.cz.yaskawa.eu.com/Global%20Assets/Pictures/Products/Robotics/Robots/PL/12212/image-thumb\\_\\_12212\\_\\_small-headerswiper-left/PL800\\_right\\_web.webp](https://www.cz.yaskawa.eu.com/Global%20Assets/Pictures/Products/Robotics/Robots/PL/12212/image-thumb__12212__small-headerswiper-left/PL800_right_web.webp).
- [26] Stäubli. Online. [cit. 2023-11-23]. Dostupné z: <https://www.staubli.com/cz/en/corp>.

- [27] Průmyslový robot Stäubli. In: Stäubli. Online. [cit. 2023-11-23]. Dostupné z: <https://www.staubli.com/adobe/dynamicmedia/deliver/dm-aid--2685f63d-4e51-4c8b-85c3-c6a0c7801929/tx2-200-robot.png>.
- [28] Universal Robots. Online. [cit. 2023-11-23]. Dostupné z: <https://www.universal-robots.com>.
- [29] Průmyslový robot UR. In: Universal Robots. Online. [cit. 2023-11-23]. Dostupné z: [https://www.universal-robots.com/media/1824233/06\\_2022\\_ur20\\_product\\_pictures-9.jpg](https://www.universal-robots.com/media/1824233/06_2022_ur20_product_pictures-9.jpg).
- [30] Wattův odstředivý regulátor. In: Research Gate. Online. [cit. 2023-11-23]. Dostupné z: [https://www.researchgate.net/figure/Sketch-of-a-Watts-centrifugal-governor-linked-to-the-throttle-valve-Image-from\\_fig1\\_335404011](https://www.researchgate.net/figure/Sketch-of-a-Watts-centrifugal-governor-linked-to-the-throttle-valve-Image-from_fig1_335404011).
- [31] Teach-Pendant. In: FANUC America. Online. [cit. 2023-11-23]. Dostupné z: <https://www.fanucamerica.com/images/default-source/robot-options/ipendant.jpg>.
- [32] Kolaborace s kobotem. In: Universal Robots. Online. [cit. 2023-11-23]. Dostupné z: [https://a.storyblok.com/f/169662/1298x866/7f656c86f2/website-version\\_umbraco-09\\_2019\\_bw\\_industrie\\_case\\_story\\_france.jpg](https://a.storyblok.com/f/169662/1298x866/7f656c86f2/website-version_umbraco-09_2019_bw_industrie_case_story_france.jpg).
- [33] Krones AG. Online. [cit. 2023-11-23]. Dostupné z: <https://www.krones.com/en/index.php>.
- [34] Mapa poboček Krones. In: Krones AG. Online. [cit. 2023-11-23]. Dostupné z: [https://www.krones.com/media/images/230505\\_Weltkarte\\_GB\\_en\\_blau-Verlauf\\_Schrift\\_Textgrau.png](https://www.krones.com/media/images/230505_Weltkarte_GB_en_blau-Verlauf_Schrift_Textgrau.png).
- [35] Stroj Contiform. In: Krones AG. Online. [cit. 2023-11-23]. Dostupné z: [https://www.krones.com/media/images/230505\\_Weltkarte\\_GB\\_en\\_blau-Verlauf\\_Schrift\\_Textgrau.png](https://www.krones.com/media/images/230505_Weltkarte_GB_en_blau-Verlauf_Schrift_Textgrau.png).
- [36] HMI společnosti Siemens. In: Siemens. Online. [cit. 2023-11-23]. Dostupné z: <https://assets.new.siemens.com/siemens/assets/api/uuid:7938e856-c6ca-4f3a-8261-c21f814e9412/width:100/quality:low/7938e856-c6ca-4f3a-8261-c21f814e9412.webp>.
- [37] Interní databáze obrázků společnosti Krones, Krones AG, 2024