

Řízení a kontrola výrobního procesu linky FMS-200

Diplomová práce

Vedoucí práce:

Dr. Ing. Radovan Kukla

Bc. Michal Malý

Brno 2015

Zde bych rád poděkoval vedoucímu mé práce Dr. Ing. Radovanu Kuklovi za umožnění vypracování této práce, cenné rady, vstřícný přístup a odborné vedení při zpracovávání.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Řízení a kontrola výrobního procesu linky FMS-200** vypracoval/a samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou Směrnicí o zveřejňování vysokoškolských závěrečných prací.

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 22. května 2015

Abstract

Malý, M. Monitoring and control of the production process lines FMS-200. Graduation thesis. Brno, 2015.

The aim of this thesis is to analyse the machine line FMS-200 made by the company SMC, its control and monitoring. The analysis includes communication between every module of the line and the industrial robot MELFA. Based on this knowledge control programs and applications will be created which will be able to monitor and control the given process. Control programs for each module were created in STEP7 environment from Siemens. OPC server technology was used for communication between the modules and the robot. Control Web application has been made for monitoring and controlling of each module. The result of this work can be applied to other production processes, combining various production processes and control elements.

Keywords

STEP7, MELFA, Control Web, OPC, communication, controls, monitoring.

Abstrakt

Malý, M. Řízení a kontrola výrobního procesu linky FMS-200. Diplomová práce. Brno, 2015.

Cílem práce je analyzovat strojní linku FMS-200 vyrobenou firmou SMC, její řízení a kontrolu. Dále analyzovat a navrhnout komunikaci mezi moduly linky a robotem MELFA. Na základě získaných poznatků vytvořit řídicí programy a aplikace, pomocí které bude možné celý proces ovládat a monitorovat. Řídicí programy modulů byly vytvořeny v prostředí STEP7 od firmy Siemens. Pro komunikaci mezi jednotlivými moduly a robotem byla použita technologie OPC server. Pro vizualizaci a řízení jednotlivých modulů výrobní linky byla vytvořena aplikace v programu Control Web. Výsledky této práce je možné aplikovat i na jiné výrobní procesy, kombinující různé výrobní procesy a kontrolní prvky.

Klíčová slova

STEP7, MELFA, Control Web, OPC, komunikace, řízení, kontrola.

Obsah

1	Úvod a cíl práce	15
1.1	Úvod	15
1.2	Cíl práce	15
2	Analýza komponent výrobního procesu	17
2.1	Manipulační linka FMS-200	17
2.1.1	Modul FMS-202.....	19
2.1.2	Modul FMS-205.....	20
2.1.3	Modul FMS-206.....	21
2.2	Senzorický systém linky FMS-200	22
2.2.1	Mikrospínač	22
2.2.2	Lineární potenciometr	23
2.2.3	Induktivní senzor	23
2.2.4	Kapacitní senzor.....	24
2.2.5	Fotoelektrický senzor.....	25
2.2.6	Lineární enkodér.....	26
2.3	Pneumatický systém linky FMS-200	26
2.3.1	Médium a jeho transport	27
2.3.2	Výroba média	27
2.3.3	Pneumatické pohony	28
2.3.4	Ventily	29
2.4	Programovatelný logický automat.....	30
2.4.1	Komponenty PLC.....	30
2.4.2	Cyklus PLC	31
2.4.3	Programovací jazyk PLC	31
2.5	Robotický manipulátor MELFA	32
2.5.1	Úvod do problematiky robotických manipulátorů.....	32
2.5.2	Kinematika manipulátorů.....	34
2.5.3	Mitsubishi Melfa RV-2AJ	35

2.5.4	Absolutní snímače polohy	36
2.5.5	Program RT-Toolbox.....	37
2.5.6	ImagingLab Robotics Library	37
2.5.7	Ovládací panel robota.....	38
3	Řídicí programy pro moduly linky	40
3.1	Simatic S7-300	40
3.1.1	Simatic S7 bloky.....	41
3.1.2	Simatic S7 proměnné.....	41
3.1.3	Simatic S7 programovací jazyky	41
3.2	Řídicí programy linky FMS-200	42
3.2.1	Požadavky na řídicí programy	42
3.2.2	Příprava pro vytváření řídicích programů	42
3.2.3	Hardwarová konfigurace modulů	43
3.2.4	Řídicí program pro modul FMS-202	44
3.2.5	Řídicí program pro modul FMS-205.....	46
3.2.6	Řídicí program pro modul FMS-206	48
3.3	Řídicí program pro robota MELFA v Labview.....	48
3.3.1	Program pro získání aktuálních souřadnic koncového efektoru ...	50
3.3.2	Program pro přesun koncového efektoru na zadané souřadnice ..	50
3.3.3	Program pro ovládání sevření nebo rozevření koncového efektoru	51
4	Komunikace mezi komponentami linky	53
4.1	OPC architektura	53
4.1.1	OPC klient	54
4.1.2	OPC server	54
4.2	Siemens S7 Simatic.Net OPC Server	55
4.2.1	Inicializace a nastavení S7 Simatic.Net OPC serveru	55
4.2.2	PC Station Configuration Editor	57
4.2.3	Přístup k hodnotám OPC serveru.....	58
4.3	MatrikonOPC Simulation Server	59
4.3.1	MatrikonOPC Explorer.....	61

4.4	Alternativní OPC servery	61
4.5	Monitorované hodnoty	62
4.5.1	Hodnoty pro modul FMS-202	62
4.5.2	Hodnoty pro modul FMS-205	63
4.5.3	Hodnoty pro modul FMS-206	64
4.5.4	Hodnoty pro robota MELFA.....	65
4.6	Připojení klientů k OPC serveru	66
4.6.1	Připojení Control Webu k OPC serveru.....	66
4.6.2	Připojení MatrikonOPC Exploreru k OPC severu	68
4.6.3	Připojení Labview k OPC serveru	71
5	Vizualizace komponent v prostředí Control Webu	75
5.1	Vývojové prostředí Control Web	75
5.2	Vizualizace komponent linky FMS-200	76
5.2.1	Panel pro modul FMS-202	78
5.2.2	Panel pro modul FMS-205	80
5.2.3	Panel pro modul FMS-206	81
5.2.4	Panel pro robota MELFA.....	82
6	Závěr	85
7	Literatura	88
A	Funkční schémata modulů linky FMS-200	91
B	Obsah přiloženého CD	94

Seznam obrázků

Obr. 1	Linka FMS-200 od firmy SMC (SMC International Training, 2015)	18
Obr. 2	Modul FMS-202 od firmy SMC (SMC International Training, 2015)	19
Obr. 3	Modul FMS-205 od firmy SMC (SMC International Training, 2015)	20
Obr. 4	Modul FMS-206 od firmy SMC (SMC International Training, 2015)	21
Obr. 5	Ukázka mikropsínače	23
Obr. 6	Ukázka lineárního potenciometru	23
Obr. 7	Ukázka indukčního senzoru	24
Obr. 8	Ukázka kapacitního senzoru	25
Obr. 9	Řídicí jednotka CR1-571	36
Obr. 10	Funkční paleta pro robota Mitsubishi v prostředí Labview (ImagingLab, 2015)	38
Obr. 11	Ovládací panel robota	39
Obr. 12	Ukázka nastavení připojení pro robota MELFA v ImagingLab Robotics Library v řídicím panelu Labview	49
Obr. 13	Ukázka blokového diagramu funkce pro získání pozice robota MELFA v kartézském souřadném systému	50
Obr. 14	Ukázka blokového diagramu procedury pro pohyb robota MELFA na zadané kartézské souřadnice	51
Obr. 15	Ukázka blokového diagramu pro ovládání sevření nebo rozevření koncového efektoru robota MELFA	52
Obr. 16	Ukázka OPC architektury klient – server (Foxon, 2014)	54
Obr. 17	Hardwarová konfigurace PC station	56
Obr. 18	Zobrazení síťového propojení všech našich komponent	56

Obr. 19	Zobrazení výsledného propojení komponent pomocí S7 connection	57
Obr. 20	Nastavený PC Station Configuration Editor	58
Obr. 21	MatrikonOPC Simulation Server	60
Obr. 22	Vytváření aliasu na MatrikonOPC Simulation Serveru	61
Obr. 23	Výběr OPC serveru v konfiguračním nástroji OPC ovladače	67
Obr. 24	Tabulka kanálů v konfiguračním nástroji OPC ovladače	68
Obr. 25	MatrikonOPC Explorer	69
Obr. 26	Přidání aliasů do MatrikonOPC Exploreru	70
Obr. 27	Změna hodnoty atributu v MatrikonOPC Exploreru	71
Obr. 28	DataSocket paleta v programu Labview	72
Obr. 29	Čtení hodnot z OPC serveru a předání hodnot proceduře pro ovládání MELFY	73
Obr. 30	Zápis hodnot obdržených z MELFY na OPC server	74
Obr. 31	Ukázka přidání zařízení reprezentující OPC server v prostředí Control Webu	77
Obr. 32	Ukázka panelu pro vizualizaci a ovládání modulu FMS-202 v prostředí Control Webu	80
Obr. 33	Ukázka panelu pro vizualizaci a ovládání modulu FMS-205 v prostředí Control Webu	81
Obr. 34	Ukázka panelu pro vizualizaci a ovládání modulu FMS-206 v prostředí Control Webu	82
Obr. 35	Ukázka panelu pro vizualizaci a ovládání robota MELFA v prostředí Control Webu	84
Obr. 36	Funkční schéma modulu FMS-202	91
Obr. 37	Funkční schéma modulu FMS-205	92
Obr. 38	Funční schéma modulu FMS-206	93

Seznam tabulek

Tab. 1	IP adresy jednotlivých modulů linky FMS-200	43
Tab. 2	Vybavení modulu FMS-202	44
Tab. 3	Vybavení modulu FMS-205	44
Tab. 4	Vybavení modulu FMS-206	44
Tab. 5	Ukázka řetězců pro přístup k hodnotám OPC serveru	59
Tab. 6	Ukázka výrobců OPC serverů pro komunikaci se Siemens Simatic S7	62
Tab. 7	OPC atributy pro modul FMS-202	63
Tab. 8	OPC atributy pro modul FMS-205	64
Tab. 9	OPC atributy pro modul FMS-206	65
Tab. 10	OPC atributy pro robota MELFA	66

1 Úvod a cíl práce

1.1 Úvod

V dnešní době se u výrobních procesů velice často setkáváme s velkou rozmanitostí použitých technologických procesů, výrobních postupů atd. To sebou přináší různé výhody ve formě snížených výrobních nákladů, zvýšení kvality výsledného produktu, atd. Monitorování jednotlivých stanic výrobního procesu není nic náročného. Často máme k dispozici řešení přímo od výrobce, nebo si můžeme monitorování pořídit či vytvořit sami. To platí i v případech kdy používáme různé typy či druhy stanic od stejného výrobce.

V případě, kdy máme v rámci našeho procesu k dispozici stanice od různých výrobců, tak se možnost monitorovat celý výrobní proces výrazně komplikuje. Většina výrobců si svoje technologie a postupy chrání a přístup k nim poskytují pouze skrze licence. Z toho vyplývá, že pro monitorování celého výrobního procesu si musíme zajistit či pořídit přístup ke všem jednotlivým stanicím.

Od momentu, kdy máme k dispozici způsob, jak se připojit ke všem stanicím výrobního procesu můžeme začít vytvářet monitorování a řízení celého výrobního procesu, tak aby vyhovovalo našim potřebám. Můžeme si vytvořit program, který bude zahrnovat celý proces od začátku až do konce, nebo si můžeme vytvořit program, který bude zahrnovat pouze kritickou část výrobního procesu, popřípadě pouze jednotlivé pracovní stanice či moduly.

Výsledkem je vytvoření monitorování a řízení výrobního procesu přesně na míru našim potřebám. Díky tomu můžeme zajistit efektivní řízení výrobního procesu. Zároveň vytvoříme automatizovaný výrobní proces, který odstraní možnost vzniku chyb způsobených lidskou obsluhou, a zároveň zvýšíme bezpečnost na pracovišti, jelikož obsluha bude mít za úkol vykonávat jen údržbu, kontrolu a provádět jen nezbytné úkony vyžadující lidskou interakci.

1.2 Cíl práce

Cílem práce je vytvořit řídicí programy pro řídicí jednotky programovatelných logických automatů jednotlivých modulů manipulační linky FMS-200 od firmy SMC. Při vytváření těchto programů je potřeba zohlednit možné zapojení do společné výrobní linky, která bude obsahovat kromě jednotlivých modulů i jiné komponenty s rozdílným způsobem řízení.

Z důvodů realizace společného zapojení do výrobní linky, práce bude obsahovat analýzu a návrh komunikace mezi jednotlivými stanicemi či moduly výrobní linky.

Na základě těchto kroků bude vytvořena vizualizace jednotlivých stanic výrobní linky či výrobního procesu v programu Control Web, kde bude možné monitorovat a řídit jednotlivé stanice nebo celý výrobní proces.

Poznatky získané a vytvořené v této práci by měli posloužit jako podklad či návod pro vytváření různých výrobních procesů, nebo jako základní kámen pro další rozšiřování a vylepšování celého procesu.

2 Analýza komponent výrobního procesu

2.1 Manipulační linka FMS-200

Manipulační linka FMS-200 je vyráběna firmou SMC (Sintered Metal Corporation). Linka je určena primárně jako výukový nástroj, ale dá se i nasadit v reálném pracovním prostředí. Za tímto účelem je linka vybavena patřičnými technologiemi, které zaručují její jednoduchost a bezpečnost při nasazení k výuce nebo provozu.

Hlavní vlastností linky je její vysoká modulárnost. Linka se může skládat až z deseti modulů, které spolu spolupracují, ale zároveň jsou schopny pracovat samostatně, což nám umožní začít s jedním nebo několika moduly a postupem času je můžeme doplnit. Linka může obsahovat stanice:

- Modul pro podávání těles, do kterých budeme následně vkládat další objekty (FMS-201)
- Modul pro výběr a vkládání ložisek (FMS-202)
- Modul obsahující hydraulický lis, který má za úkol zatlačit ložisko do tělesa (FMS-203)
- Modul pro výběr a vkládání os do připraveného tělesa s ložiskem (FMS-204)
- Modul pro výběr a vkládání krytů do tělesa (FMS-205)
- Modul pro vkládání šroubků do připraveného tělesa (FMS-206)
- Modul pro robotické šroubování šroubků v tělese (FMS-207)
- Modul pro ukládání hotových produktů linky (FMS-208)
- Modul obsahující pec pro sušení obarvených těles (FMS-209)
- Modul zajišťující výstupní kontrolu zhotovených těles (FMS-210)

Díky rozmanitosti činností, které může linka provádět, jeví se jako ideální nástroj pro školení, protože nám umožní provádět četné činnosti:

- Výběr materiálu na základě jeho vlastností (výška, barva, typ materiálu, atd.)
- Doprava materiálu různými způsoby (uchopování, otáčení, posouvání, atd.)
- Využívá různé průmyslové technologie (senzory, robotiku, komunikaci, atd.)
- Využívá kontrolní panely

Jednotlivé moduly jsou umístěny na hliníkových stolech s kolečky a osazeny jednoduchým kontrolním panelem. Díky tomu je zajištěna stabilita a flexibilita při umístění modulu či linky v rámci výrobního procesu. Manipulační linka pro svůj chod využívá pneumatický systém. To sebou nese četné výhody a jeho nasazení

pro výukové účely navíc zajistí bezpečnost jak pro obsluhu, tak i samotnou linku. Pro řízení svých činností jsou linky osazeny PLC systémy od různých výrobců:

- Mitsubishi PLC
- Siemens PLC
- Omron PLC
- Allen Bradley PLC
- Schneider PLC
- Lze moduly pořídít i bez PLC

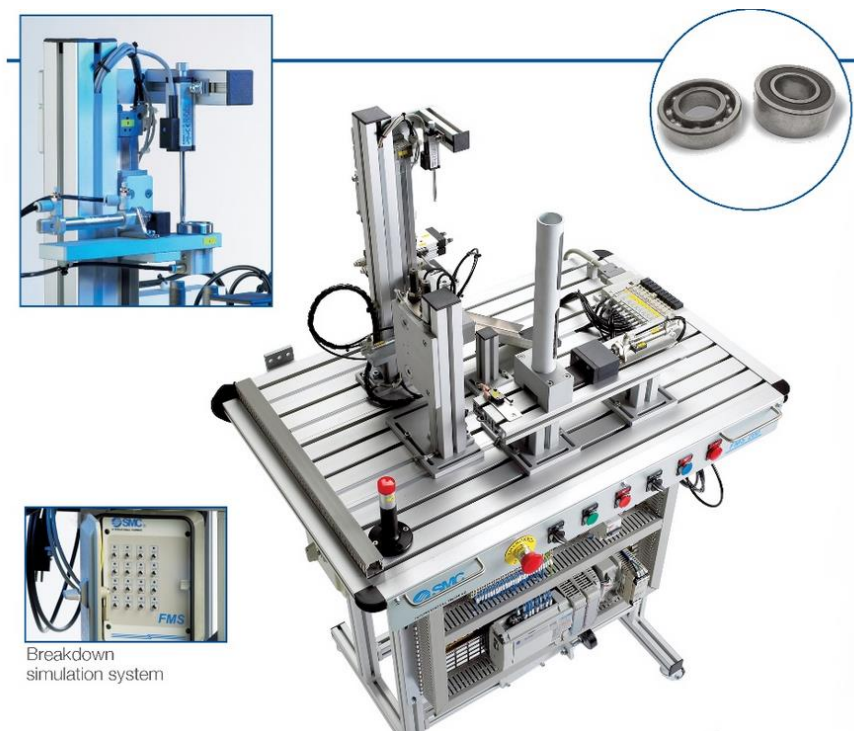


Obr. 1 Linka FMS-200 od firmy SMC (SMC International Training, 2015)

Každý modul je navíc vybaven systémem pro simulaci poruch, který umí simulovat až 16 různých problémů. Díky tomu lze provádět výuku analýzy a odstraňování poruch, které mohou nastat během výrobního procesu. (SMC International Training, 2015)

2.1.1 Modul FMS-202

Účelem toho modulu je zajistit podání ložisek a jejich následný transport do připraveného tělesa. Modul je zároveň vybaven systémem pro výběr ložiska na základě jeho vlastností, což nám dává možnost při vytváření řídicího programu zajistit podání jen takových ložisek, která vyhovují našim potřebám.



Obr. 2 Modul FMS-202 od firmy SMC (SMC International Training, 2015)

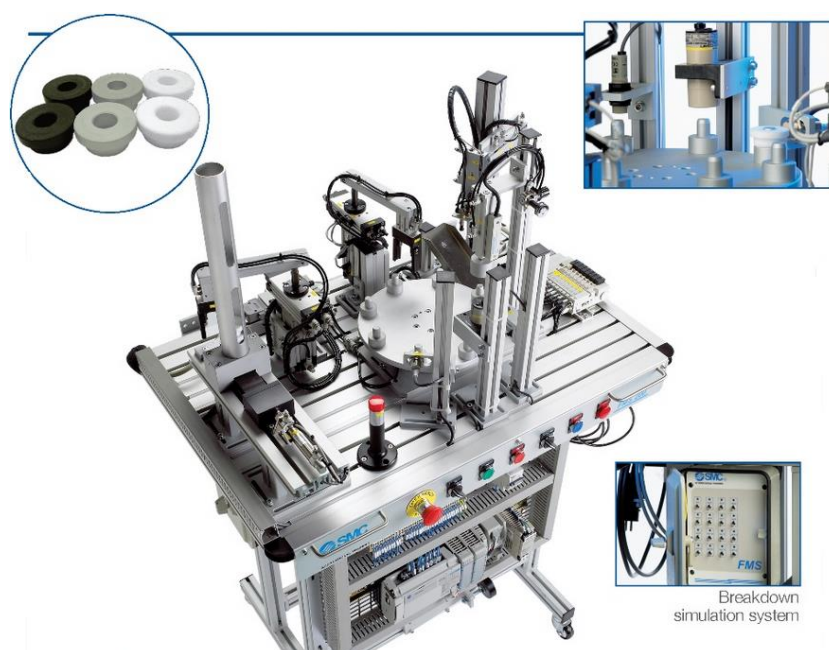
Náš modul obsahuje:

- PLC systém SIMATIC S7 od firmy Siemens, který obsahuje S7 CPU, 15 digitálních vstupů, 13 digitálních výstupů a jeden analogový vstup
- Obsahuje systém pro simulaci poruch (až 16 možných poruch)
- Sensorický systém obsahuje lineární potenciometr a mikrospínač
- Kontrolní panel s tlačítky
- Pneumatický systém řízený pomocí ventilů

Modul má za úkol zajistit podání ložiska z vertikálního zásobníku, pomocí systému přenašečů a uchopovačů jej dopravit k lineárnímu potenciometru, který změří výšku ložiska. Pokud má ložisko požadovanou výšku je dále pomocí přenašečů vloženo do připraveného tělesa, jinak je vyloučeno. (SMC International Training, 2015)

2.1.2 Modul FMS-205

Cílem modulu FMS-205 je zajistit podání krytů ze zásobníku a jejich následný transport do připraveného tělesa. Zároveň je vybaven senzorickým systémem, který dokáže zjistit vlastnosti podaných krytů. Tohoto můžeme využít při vytváření řídicího programu a zajistit si tak vložení pouze krytů splňujících kritéria, která si sami stanovíme.



Obr. 3 Modul FMS-205 od firmy SMC (SMC International Training, 2015)

Náš modul obsahuje:

- PLC systém SIMATIC S7 od firmy Siemens, který obsahuje S7 CPU, 24 digitálních vstupů, 46 digitálních výstupů a analogové počítadlo
- Obsahuje systém pro simulaci poruch (až 16 možných poruch)
- Sensorický systém obsahuje mikrospínač, indukční senzor, kapacitní senzor, fotoelektrický senzor, vakuový tlakový senzor a lineární enkodér
- Kontrolní panel s tlačítky
- Pneumatický systém řízený pomocí ventilů

Modul má za úkol podání krytu z vertikálního zásobníku a jeho dopravu pomocí přenašečů na otočný stůl. Otáčením stolu je zajištěno projetí krytu systémem senzorů, které jsou schopny zjistit vlastnosti materiálu krytu. Pokud odpovídá naším zvoleným kritériím, tak je pomocí přenašečů vložen do připraveného tělesa, jinak je pomocí vakuové přísavky vyloučen. (SMC International Training, 2015)

2.1.3 Modul FMS-206

Cílem modulu FMS-206 je zajistit podání šroubků ze zásobníku a jejich následné dopravení do připraveného tělesa. Tento modul nedokáže rozlišovat žádné vlastnosti podaných šroubků. Pouze zajišťuje jejich podání a dopravu do tělesa.



Obr. 4 Modul FMS-206 od firmy SMC (SMC International Training, 2015)

Náš modul obsahuje:

- PLC systém SIMATIC S7 od firmy Siemens, který obsahuje S7 CPU, 13 digitálních vstupů, 9 digitálních výstupů
- Obsahuje systém pro simulaci poruch (až 16 možných poruch)
- Sensorický systém obsahuje pouze jednoduchý optický senzor (optická závora)
- Kontrolní panel s tlačítky
- Pneumatický systém řízený pomocí ventilů

Modul má za úkol podání šroubku z vertikálního zásobníku do připraveného přepravníku obsahující jednoduchou optickou bránu pro detekci úspěšného podání šroubku. Poté pomocí přenašečů zajistí podání šroubku do připraveného tělesa. Modul nemá žádné možnosti otáčení nebo manipulace s tělesem, proto je nutné zajistit tuto činnost pomocí nějakého externího mechanismu. (SMC International Training, 2015)

2.2 Senzorický systém linky FMS-200

Každý modul je vybaven senzorickým systémem, který je potřebný pro získávání nebo zajištění informací potřebných pro řídicí jednotku. Díky tomuto systému můžeme získat informace:

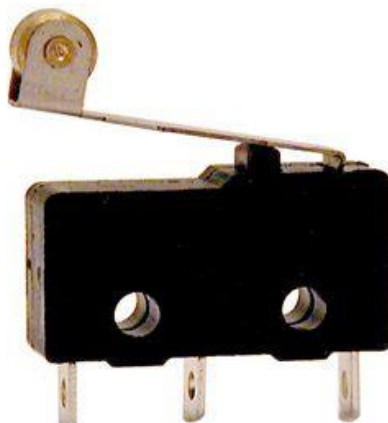
- Výška tělesa
- Barva materiálu
- Typ materiálu
- Detekce tělesa
- A jiné vlastnosti

2.2.1 Mikrospínač

Mikrospínač je druhem jednoduchého elektrického spínače, který je běžně nasazován v mnoha odvětvích průmyslu (automobilový, lékařský, strojírenský, atd.). Důvody jejich častého výskytu jsou:

- Jednoduchost
- Spolehlivost
- Nízké výrobní náklady

Princip funkce tohoto spínače je velice jednoduchý. Spínač má tři kontaktní plochy, reprezentující uzemnění, vstupní a výstupní kontakt. Dále obsahuje ve většině případu plastové tělo s mechanismy pro upevnění a jednoduché kontaktní rameno s pružinou (Obr. 5). Uvnitř spínače jsou vstupní a výstupní kontakty rozpojeny. Působením tlaku na kontaktní rameno dojde ke stlačení pružiny, což má za následek spojení vstupního a výstupního kontaktu. (Engineers Garage, 2012)



Obr. 5 Ukázka mikrosvítače

2.2.2 Lineární potenciometr

Potenciometry jsou elektronické součástky sloužící pro změny hodnot odporu elektrického obvodu. Změna odporu je vyvolána za pomoci posuvníku. Podle druhu posuvníku rozlišujeme:

- **Tahový (lineární) potenciometr** – druh potenciometru, u kterého se posuvník pohybuje po přímé dráze.
- **Rotační (otočný) potenciometr** – druh potenciometru, u kterého se posuvník otáčí okolo osy.

Změnou odporu v obvodu se změní samotné vlastnosti takového obvodu. Na základě této skutečnosti můžeme v automatizaci například měřit ujetou dráhu, velikost nějakého tělesa, stupeň otočení a jiné. (IHS Engineering360, 2015)



Obr. 6 Ukázka lineárního potenciometru

2.2.3 Induktivní senzor

Induktivní senzory jsou velice jednoduché a efektivní senzory, které dokáží detekovat metalické (vodivé) předměty (v určitých případech i jiné, ovšem s relativně zkrácenou vzdáleností) na určitou vzdálenost bez toho, aby se jich jakkoliv dotknuly.

Senzory obsahují cívku a oscilátor, které vytváří v okolí senzoru magnetické pole. V momentu, kdy do magnetického pole vstoupí předmět, dojde k utlumení amplitudy oscilátoru. Takovéto změny amplitudy jsou zaznamenávány obvodem. Pokud tato změna přesáhne stanovenou hranici obvodu, dojde k změně výstupního napětí senzoru, což signalizuje prezenci tělesa ve snímané oblasti senzoru. Vzdálenost, na jakou dokáže takovýto senzor fungovat, záleží na prostředí, ve kterém se nachází, jeho tvaru, velikosti a vlastnostech samotného tělesa. (Fargo Controls, 2013)



Obr. 7 Ukázka indukčního senzoru

2.2.4 Kapacitní senzor

Kapacitní senzory jsou svou stavbou a funkcí velice podobné indukčním senzorům. Hlavní rozdíl je ovšem v tom, že dokáží detekovat i nevodivé předměty. Hlavním principem funkce tohoto senzoru je změna jeho kapacity na základě přítomnosti tělesa v jeho blízkosti.

Kapacita je schopnost objektu či vodiče uchovat si elektrický náboj. Senzor obsahuje dvě elektrody, mezi kterými vzniká elektrické pole s elektrickým potenciálem. Dalo by se říct, že tak vznikne kondenzátor s určitou kapacitou. V momentu, kdy se ke snímací plošce přiblíží těleso, dojde ke změně kapacity kondenzátoru. Tato změna je zaznamenána oscilátorem, který je na kondenzátor napojený. Tato změna je zaregistrována kontrolním obvodem, který tuto změnu vyhodnotí. Když změna přesáhne hranici obvodu, dojde ke změně výstupního napětí senzoru. (Lionprecision, 2012)



Obr. 8 Ukázka kapacitního senzoru

2.2.5 Fotoelektrický senzor

Fotoelektrický senzor je jeden z dalších bezkontaktních senzorů. Tento druh senzoru využívá principů emitace a detekce světelného paprsku. Tento parsek může být buď viditelný, nebo infračervený. Podle způsobu detekce světelného paprsku rozlišujeme:

- **Optické závory** – senzor se skládá z přijímače a vysílače. Ty jsou umístěny naproti sobě tak, aby vysílaný paprsek dopadal na přijímač. Detekce tělesa je realizována přerušením vysílaného paprsku, což má za následek změny výstupního signálu senzoru. Tento druh senzoru je využíván pouze pro detekci přítomnosti těles.
- **Retro reflexivní senzory** – senzor se skládá z přijímače, vysílače a odrazky. Vysílač a přijímač jsou uloženy přímo naproti odrazce. Vysílač posílá paprsek na odrazku a ta jej odrazí na přijímač. V momentu vložení tělesa před odrazku dojde k přerušení paprsku, což má za následek změny výstupního signálu senzoru. Tento druh senzoru je využíván pouze pro detekci přítomnosti těles.
- **Difúzní senzory** – senzor obsahuje pouze přijímač a vysílač. Vysílaný paprsek je detekován jen v případě, že dojde k jeho odražení od tělesa, které se dostane před vysílač. Tento druh senzoru je schopen detekovat barvu tělesa a jeho reflexivitu.

Výhody těchto senzorů:

- Schopnost zachytit jakékoliv těleso
- Rychlost detekce těles
- Schopnost zachytit tělesa na velkou vzdálenost
- Umožňují identifikovat vlastnosti těles
- Přesnost s jakou jsou schopny detekovat tělesa (SensorCentral.com, 2014)

2.2.6 Lineární enkodér

Lineární enkodér je druh snímače specializovaný na zjišťování polohy na základě nastaveného měřítka. Celý princip funkce je založený na snímání pozice na měřítku. Na základě pozice vrací senzor údaje o pozici ve formě signálů:

- **Analogový signál** – pro výstup využívá kosinového a sinového signálu, který lze jednoduše detekovat pomocí osciloskopu. Tyto signály jsou často vysílány separátně, což pomáhá redukovat rušení zapříččeného elektromagnetickou emisí.
- **Digitální signál** – výstupem je signál, který vzniká interpolací kosinového a sinového signálu.

Enkodéry mají výstupní kanály, jejichž hodnoty signálů jsou následně převedeny na obdélníkové signály, jenž jsou oproti sobě posunuté o devadesát elektrických stupňů.

Snímače můžeme rozdělit na:

- **Inkrementální** – změna polohy je detekována na základě posunu senzoru o pevně stanovený krok. Pro výpočet polohy je důležité znát počáteční stav, na základě kterého se vypočítává hodnota posunu.
- **Absolutní** – změna polohy je vracena ve formě kódu, který reprezentuje polohu v rámci rozlišení snímače. (Walcher, 1994, s. 105-110)

2.3 Pneumatický systém linky FMS-200

Linka FMS-200 využívá systém pneumatických prvků řízených pomocí programovatelného automatu. Jako médium je využíváno stlačeného vzduchu. To sebou přináší výhody:

- Lehká údržba celého systému
- Bezpečnost pro obsluhu i samotné vybavení linky
- Vhodnost pro nasazení do výroky
- Lze využít četné způsoby řízení takového systému
- Mobilita systému
- Jednoduchost systému

Při nasazení pneumatického systému musíme zajistit:

- Těsnost celého systému
- Tlak uvnitř systému nikdy nesmí klesnout nebo překročit stanovené meze
- Podle nasazení zdroje média se můžeme setkat se vznikem hluku
- Musíme zajistit eliminaci výskytu nečistot v médiu

2.3.1 Médium a jeho transport

V rámci pneumatického systému si můžeme zvolit různé formy plynů jako média. Nejčastěji se využívá stlačený atmosférický vzduch. Lze využít i různé vzácné plyny, ale ty jsou využity jen vzácně, jelikož jejich výroba, úprava a nasazení sebou často nese výrazně vyšší náklady a požadavky na údržbu.

Výroba stlačeného vzduchu je jednoduchá a nenákladná. Ovšem atmosférický vzduch kromě plynných složek obsahuje vlhkost a nečistoty. Vlhkost ve vzduchu závisí na jeho teplotě, tlaku a prostředí, ve kterém se nachází. Nečistoty se ve vzduchu vyskytují ve formě částic o různé velikosti. Ta se pohybuje v rozmezí mikrometrů až několika milimetrů. Kromě částic může atmosférický vzduch obsahovat i mikroorganismy, jako jsou bakterie, viry, spory, atd.

Transport média je řešen systémem potrubí. To může být tvořeno systémem hadic a trubic, které mohou být vyrobeny z různého materiálu. Rychlost, objem a tlak protečeného vzduchu je v potrubí ovlivněno jeho průměrem. Proto je důležité zajistit použití potrubí o stejném průměru, popřípadě do systému vložit prvky, které umožní regulaci průtoku a tlaku tak, aby bylo dosaženo ve všech částech systému stejného průtoku a hlavně tlaku. Rozvodný systém zároveň musí vykazovat vysoký stupeň těsnosti. Únik média ze systému nejen, že zvyšuje provozní náklady systému, ale zároveň může negativně ovlivnit funkčnost celého systému.

2.3.2 Výroba média

Výroba stlačeného vzduchu jako média je jednoduchou záležitostí. Můžeme si jej nechat dopravit na místo ve formě naplněného zásobníku, anebo si jej můžeme sami vyrábět pomocí kompresoru.

Kompresor je zařízení určené k výrobě stlačeného atmosférického vzduchu. Jako pohon využívá elektrický nebo spalovací motor. Ten pohání ústrojí určené pro nasávání a následné stlačování vzduchu. Podle použitého ústrojí rozlišujeme kompresory:

- **Objemové** – jsou založené na principu zmenšování objemu nasátého vzduchu
 - **Pístové** – vzduch je nasán do pracovní komory pístu, který je přes klikovou hřídel připojen k pohonu. Pohybem pístu a otvíráním a zavíráním vstupního a výstupního ventilu je zajištěno nasávání vzduchu a jeho stlačování. Stlačený vzduch je uchovávan v vzdušníku. Při výrobě vzduchu dochází ke vzniku tepla, proto je důležité zajistit chlazení kompresoru.
 - **Membránové** – V pracovní komoře kompresoru se nachází membrána. Membrána je pomocí ojnice připojena ke klikové hřídeli, která je napojena na pohon kompresoru. Pohybem membrány nahoru a dolů dochází k nasátí vzduchu do pracovní části kompresoru a jeho následné vytlačení do vzdušníku.

- **Lamelové** – jsou druhem rotačních objemových kompresorů. V pracovním prostoru kompresoru (válcového tvaru) jsou excentricky uložené lopatky, které se vlivem odstředivých sil vysouvají. Tím dochází ke zmenšování pracovního prostoru, což má za následek stlačování nasávaného vzduchu. Při nasávání vzduchu je nutné zajistit vstřikování oleje, který slouží pro mazání a chlazení kompresoru.
- **Proudové** – zajišťují nasátí vzduchu s vysokou rychlostí, který je poté ve vzdušníku stlačován. Stlačování funguje na principu převodu kinetické energie vzduchu na tlakovou. Nasátí vzduchu je umožněno pomocí lopatkového kola nebo vrtule.

Vzdušníky jsou nádoby o různém objemu, které slouží pro uchovávání stlačeného vzduchu. Zároveň v systému slouží jako vyrovnávací nádoby, které pomáhají regulovat tlakové výkyvy způsobené během systému.

Při výrobě se do média pravidelně dostávají nečistoty a vlhkost. Ty se tam mohou dostat nejen z prostředí, ale i ze samotného kompresoru. Tím jsou myšleny prachové částice, vodní páry, oleje sloužící ke chlazení kompresů, atd. Proto je nutné zajistit umístění filtračních prvků do systému, které dokáží tyto nečistoty zachytit. K odstranění vlhkosti se stlačený vzduch musí ochladit a popřípadě i vysušit. (Vančura, 2007)

2.3.3 Pneumatické pohony

Princip funkce pneumatických pohonů je založen na převodu tlakové síly přiváděného média na mechanickou práci pohonu. Proto jsou pneumatické pohony vybaveny prvky, nejčastěji písty, které jsou schopny takového přenosu. Pohony můžeme rozlišit na:

- **Válcové pohony** – nejčastěji vykonávají lineární přímočarý pohyb. Uvnitř válce se nachází píst, který je tlakem média posouván do konečné pozice. Podle způsobu návratu pístu rozlišujeme:
 - Jednočinné válce – pro návrat pístu do výchozí pozice využívají mechanických prvků jako například pružin, elastických prvků a jiné.
 - Dvojčinné válce – pro návrat do výchozí pozice je přiveden stlačený vzduch z opačné strany válce.
- **Kyvné pohony** – pohony upravené tak, že je přiváděný tlak převáděn na pohyb okolo osy s přesně vymezenými dorazy. To znamená, že můžeme dosáhnout natočení o určitý úhel.
- **Rotační pohony** – někdy se jim taky říká pneumatické motory. Přiváděné médium způsobuje rotaci kolem osy. U této rotace můžeme regulovat otáčky motoru, výkon motoru, směr rotace a točivý moment.

Nasazení pneumatických pohonů sebou nese řadu výhod:

- Jemnost pohybu
- Nízká údržba
- Rychlost pohybu
- Nízký hluk
- Vysoká odolnost

Mezi nevýhody lze zmínit:

- Nerovnoměrný pohyb při nízkých rychlostech
- Polohování v jiných než krajních pozicích
- Někdy může být provoz dražší než elektrické varianty (Vančura, 2007)

2.3.4 Ventily

Ventily jsou prvky pneumatického systému, které mají za úkol řídit a regulovat směr, tlak a proudění média v systému. Podle způsobu ovládání můžeme ventily rozdělit na:

- **Manuální** – ovládání ventilů je zajištěno pomocí páček, tlačítek a spínačů, které ovládá obsluha.
- **Mechanické** – k ovládání těchto ventilů jsou použité pružiny, péra, ramena a kladky.
- **Tlakové** – ventily jsou ovládány přivedením tlaku na vstup.
- **Elektrické** – ventily jsou ovládány přivedením napětí na elektromagnet. Ten může být buď s jedním, nebo se dvěma vinutími.
- **Kombinované** – ventily jsou ovládány kombinací výše vypsanych možností ovládání. Například ventil ovládaný elektromagnetem s pružinou.

Ventily dále můžeme dělit podle jejich konstrukce:

- **Cestné ventily** – tlakové médium je přes ventil vedeno systémem cestiček, pomocí kterých uzavírá, otvírá a mění jeho proudění.
- **Sedlové ventily** – tento typ ventilu je relativně jednoduchý. Zajišťuje pouze průchod tlakového média. Uvnitř ventilu se nachází pouze uzavírající těleso (kulička, kužel, atd.) umístěné na pružině, která zajišťuje návrat do výchozí pozice ventilu.
- **Šoupátkové ventily** – na průchozím kanálu pro médium ve ventilu se nachází kolmo uložený rotační disk s průřezy, pomocí kterých je regulován průtok média ventilem.
- **Logické ventily** – propouští vzduch ventilem na základě přítomnosti tlaků na vstupních kanálech. K tomu využívá logické funkce AND nebo OR. (Vančura, 2007)

2.4 Programovatelný logický automat

Programovatelný logický automat (dále jen PLC¹) je druh počítače určený pro automatizaci výrobních procesů a strojů nasazovaných v průmyslovém prostředí. Z toho vyplývá, že u jejich vývoje došlo k uzpůsobení softwaru i hardwaru takovým způsobem, aby bylo možné přijímat a vysílat signály potřebné pro monitorování a řízení technologických procesů.

PLC byly dříve reprezentovány systémem jednoduchých relé systémů, které tvořili logiku řídicí daný proces nebo stroj. Se zvyšující se složitostí a náročností řízených procesů či strojů bylo nutné vytvářet složité reléové logické systémy, které reprezentovaly automatizaci. Tyto systémy pak dosahovali velkých rozměrů, jejich údržba byla složitá a případné změny se prováděly velice obtížně. To zapříčinilo nutnost najít náhradu, která by tyto nedostatky odstranila. To umožnil rozvoj miniaturizace na poli počítačových systémů. Jednoduché reléové systémy začali nahrazovat mikroprocesory, které byly schopny zpracovávat signály a zároveň umožnily pružně reagovat na změny. K tomu bylo pouze potřeba upravit či vytvořit nový řídicí program.

Při řízení a monitorování technologických procesů musí být PLC schopnost zpracovávat velké množství signálů. Rozlišujeme dva druhy signálů:

- **Binární signály** – někdy bývají označovány jako digitální signály. Jsou to jednoduché signály, které reprezentují hodnoty 1 (*true*) a 0 (*false*). Mohou například signalizovat stav otevřeno/zavřeno prvků procesu atd.
- **Analogové signály** – složitější signály, nejčastěji nesoucí informace o vlastnostech procesu či stroje jako je například teplota, vlhkost atd. Tyto signály musí být prvně zpracovány a převedeny pomocí A/D převodníků, aby s nimi mohlo PLC pracovat.

Kromě signálů souvisejících s technologickým procesem musí PLC zpracovávat i interní signály, které souvisí s řídicí logikou PLC. Ta je řešena na základě nahrazeného řídicího programu v paměti. (Rehg, 2009)

2.4.1 Komponenty PLC

PLC systémy jsou složeny z:

- **Central Processing Unit (CPU)** – tvoří samotné jádro PLC. Jeho úkolem je provádět řízení na základě uloženého řídicího programu, který je prováděn v pravidelných cyklech. Kromě toho aktualizuje stav vstupů a výstupů uložených v paměti, provádí komunikaci mezi jednotlivými prvky a spravuje paměť PLC.
- **Paměť PLC** – je tvořena operační pamětí RAM, která slouží pro uchovávání stavů periférií a vnitřních stavů závislých na řídicím programu. Ten je uložen v sekundární paměti, která je vyhrazena čistě pro něj.

¹ Přeloženo z anglického Programmable logic Controller

- **Komunikační rozhraní** – rozhraní určené pro komunikaci mezi jednotlivými prvky PLC a okolními systémy.
- **Napájecí zdroj** – dodává napájení pro komponenty PLC systému.
- **Periferie** – moduly určené pro obsluhu a zpracovávání vstupních a výstupních signálů. Podle druhu signálů, jaké tyto moduly zpracovávají, rozlišujeme digitální a analogové periferie.

Na základě složení PLC systému rozlišujeme dva druhy PLC:

- **Kompaktní PLC** – je takové provedení PLC, které je tvořeno jedním modulem. Tento obsahuje samotné CPU, vstupní a výstupní periferie a napájecí zdroj. Kompaktní PLC mají omezený počet vstupů a výstupů. Možnosti rozšíření jsou omezené. Výhodou takového řešení je rychlost přístupu k perifériím a nízké pořizovací náklady.
- **Modulární PLC** – takovéto PLC si pomocí modulů můžeme sestavit takzvaně na míru. Moduly jsou propojené pomocí speciální sběrnice. Při sestavování modulárního PLC je potřeba zajistit přítomnost modulu obsahující CPU a napájecí zdroj. Periferní moduly lze volně kombinovat a popřípadě je lze kdykoliv rozšířit. Modulární PLC mají vyšší pořizovací náklady a díky nutnosti propojení přes speciální sběrnici může dojít ke zvýšení doby přístupu k perifériím. Výhodou je ovšem možnost řídit rozsáhlé a složité technologické procesy a stroje. (Rehg, 2009)

2.4.2 Cyklus PLC

PLC provádí svoji činnost ve stanoveném cyklu. Jeho rozsah je v desítkách až stovkách milisekund. Ten je buď pevně stanoven, nebo má uživatel možnost jej nastavit. Cyklus obnáší:

- Zpracování vnitřních instrukcí a operací uložených v CPU. Ty reprezentují různé diagnostiky, kontroly, komunikační protokoly, atd.
- Poté PLC zapíše do paměti stav všech vnitřních a vnějších vstupních periférií. Této části paměti se říká *Process Input Image (PII)*.
- Poté začne CPU zpracovávat uložený řídicí program. Ten čte řádek po řádku, od shora dolů. Během zpracování postupně ukládá do paměti výstupní signály vyplývající z řídicího programu. Této části paměti se říká *Process Output Image (POI)*.
- Po dokončení zpracování řídicího programu PLC aktualizuje stav výstupních periférií na základě hodnot uložených v paměti. (Rehg, 2009)

2.4.3 Programovací jazyk PLC

PLC používá speciální formy programovacích jazyků pro vytváření řídicích programů. Většina výrobců si tyto jazyky upravuje podle sebe, proto i když vypadají

stejně, nejsou přenosné a fungují pouze na daném PLC systému. Jazyky můžeme rozdělit do čtyř typů:

- **Instruction list jazyk** – nízko úroňový jazyk připomínající assembler. Jazyk obsahuje návěští pro skoky a cíle, symbolická jména pro číselné hodnoty a jejich samotnou hodnotu.
- **Jazyk kontaktních schémat** – jazyk je reprezentovaný grafickými schématy reprezentující reléový a spínací systém.
- **Jazyk logických schémat** – jazyk reprezentovaný základními logickými operacemi. Ty jsou reprezentovány pomocí příslušných značek.
- **Jazyk strukturovaného textu** – jazyk podobný vyšším programovacím jazykům jako je například Pascal, jazyk C, atd.
- **Grafické jazyky** – speciální druh jazyků reprezentovaných různými grafickými prvky jako například grafy. (Rehg, 2009)

2.5 Robotický manipulátor MELFA

2.5.1 Úvod do problematiky robotických manipulátorů

V mnoha průmyslových procesech a oblastech se vyskytují části, které je potřeba vykonávat rychle, přesně a opakovaně. Dříve tyto činnosti musela vykonávat lidská obsluha. To ovšem sebou neslo zvýšené náklady a zároveň tato často opakovaná činnost neúměrně zatěžovala její vykonavatele. Proto bylo zapotřebí zajistit vykonávání takovýchto monotónních činností automatizovaně. Výsledkem byl vznik automatizovaných robotických manipulátorů (někdy mohou být označovány jako roboti), které dokázaly tuto činnost vykonávat rychle, efektivně a s vysokou přesností.

Podle činnosti můžeme roboty rozdělit na:

- **Teleoperátory** – princip funkce těchto robotů je založený na dálkovém přenosu pohybu lidského operátora. Často využívané jsou v medicíně či astronautice.
- **Jednoúčelové manipulátory** – reprezentovány roboty, které vykonávají přesně stanovenou činnost. Změna této činnosti není většinou možná, nebo zahrnuje výrazný zásah do samotné konstrukce robota. Často zastoupené různými dopravníky, balicími stroji, atd.
- **Univerzální manipulátory** – jsou to roboti, kteří dokáží vykonávat více funkcí. K tomu je zapotřebí jen upravit jejich řídicí program, popřípadě menší zásah do jeho konstrukce. Ten nejčastěji obnáší jen výměnu nástroje, se kterým robot pracuje.
- **Adaptivní roboty** – tito roboti jsou vybaveny vyšším stupněm řízení. To jim umožňuje vykonávat složité technologické procesy a svoji činnost mohou

upravit či přizpůsobit na základě změn v pracovním prostředí. K tomu jsou často vybaveny senzorickým systémem.

- **Kognitivní roboty** – tento typ robota je vybaven umělou inteligencí. Ta mu umožňuje vykonávat rozhodnutí a reagovat pružně na změny ve výrobním procesu bez nutnosti zásahu lidské obsluhy.

Manipulátory se skládají z několika částí:

- **Robotická základna** – základna manipulátoru, která je připevněná k nějaké pevné podložce. Podložka musí zaručit stabilitu základny, zamezit nechtěným pohybům a musí být schopna odolat váze manipulátoru a případného tělesa, se kterým robot pracuje. Tato základna je nejčastěji považována za výchozí bod souřadného systému manipulátoru.
- **Rameno** – manipulátor může obsahovat jedno nebo více ramen. Tato ramena vykonávají pohyb samotného manipulátoru.
- **Klouby** – zajišťují propojení mezi rameny robota a zároveň vymezují rozsah pohybu ramen. Klouby mohou být pasivní, které pouze vymezují rozsah pohybů mezi dvěma rameny. Aktivní klouby, které zajišťují přesný pohyb ramen. Aktivní klouby jsou řešené pomocí:
 - Lineární pohony – pohony vykonávající posuvný (lineární) pohyb. Nejčastěji jsou používány elektromagnetické pohony, ale mohou být i piezoelektrické, pneumatické, hydraulické, atd. Pomocí nich lze například prodlužovat nebo zkracovat délku robotického ramena.
 - Rotační pohony – pohony produkující rotační pohyb. Pomocí těchto pohonů je zajištěná rotace ramen manipulátoru. Nejčastěji jsou využívány elektrické enkodéry, pomocí nichž lze zajistit přesné natočení ramen.

V případě robotických manipulátorů dále rozlišujeme základní pojmy:

- **Stupeň volnosti** – nám říká počet parametrů, pomocí kterých můžeme popsat polohu bodu nebo tělesa v souřadném systému. Parametry jsou myšleny rotace a translace.
- **Koncový efektor** – rameno manipulátoru určené pro upevnění nástroje nebo pro vykonávání činnosti manipulátoru.
- **Aktuátory manipulátoru** – jsou to zdroje pohybu manipulátoru. Často bývají integrovány do kloubů manipulátoru, proto takovéto klouby označujeme jako aktivní.
- **Výchozí poloha manipulátoru** – jsou to takové souřadnice manipulátoru, ve kterých manipulátor zaujímá výchozí polohu.
- **Pracovní prostor manipulátoru** – prostor, ve kterém se může pohybovat koncový efektor manipulátoru.

- **Přesnost manipulátoru** – je hodnota odchylky, která je dána vzdáleností současné polohy koncového efektoru od požadované polohy efektoru. (Goubaj, 2012)

2.5.2 Kinematika manipulátorů

Cílem kinematiky manipulátorů je za pomoci jednoduchých rotačních či lineárních pohybů dosáhnout určité polohy v daném souřadném či prostorovém systému. Ta je závislá na rotaci, zrychlení a poloze jednotlivých částí manipulátoru. Toho dosáhneme vhodným nastavením akuatorů manipulátoru.

Poloha manipulátoru je tedy popis vzájemné translace a rotace jednotlivých ramen manipulátoru vůči sobě. Každý pevný bod na rameni manipulátoru má svůj souřadný systém. Proto translace a rotace vždy popisuje otočení a posunutí mezi dvěma souřadnými systémy:

- Rotace kolem osy x o úhel α mezi dvěma souřadnými systémy je popsána pomocí matice:

$$R(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (1)$$

- Rotace kolem osy y o úhel β mezi dvěma souřadnými systémy je popsána pomocí matice:

$$R(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2)$$

- Rotace kolem osy z o úhel γ mezi dvěma souřadnými systémy je popsána pomocí matice:

$$R(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

- Translace mezi dvěma souřadnými systémy v ose x (Px), v ose y (Py) a v ose z (Pz) je popsána pomocí matice:

$$R(\alpha) = \begin{bmatrix} 1 & 0 & 0 & Px \\ 0 & 1 & 0 & Py \\ 0 & 0 & 1 & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Polohové závislosti mezi souřadnicemi jednotlivých kloubů manipulátoru a polohy koncového efektoru v souřadném systému pracovního prostoru lze řešit pomocí dvou úloh:

- **Přímé kinematické úlohy** – známe polohu koncového efektoru, na základě které dopočítáváme kloubové souřadnice.
- **Inverzní kinematické úlohy** – u inverzní úlohy neznáme polohu koncového efektoru. Tu dopočítáváme na základě kloubových souřadnic.

2.5.3 Mitsubishi Melfa RV-2AJ

Robot je vyráběn japonskou firmou Mitsubishi a je určen pro vykonávání přesných operací v relativně malém pracovním prostoru. Robot je štíhlé konstrukce, díky které je relativně lehký a nemá problém vykonávat i činnosti ve stísněných prostorech. Robot je schopen pracovat s tělesy nepřesahující hmotnost dvou kilogramů. Robot umožňuje instalaci různých nástrojů, pomocí kterých může následně provádět různé uchopovací, manipulační či montážní operace v rámci výrobního procesu.

Robot je vybaven pohony se střídavými servomotory s integrovanými číslíkovými snímači absolutní polohy. Díky těmto pohonům je zajištěna vysoká přesnost a rychlost pohybů vykonávaných robotem. Díky absolutním snímačům polohy je robot schopen pokračovat v pohybu i v případě přerušení. Díky použitým pohonům je dosaženo vysoké spolehlivosti a nízkých nároků na údržbu.

Řízení robota je zajištěno pomocí dodávané řídicí jednotky CR1-571 (Obr. 9). Ta obsahuje výkonný 64 bitový procesor architektury RISC² obsahující DSP³ mikroprocesor. Díky tomuto procesoru zvládne jednotka zpracovávat až 32 paralelních úloh. To robotu umožňuje řídit pohyb jednotlivých ramen a zároveň zpracovávat a vysílat data přes komunikační rozhraní, různé výpočetní úlohy atd. Jednotka je pro komunikaci standardně vybavena komunikačním rozhraním RS-232. Komunikační možnosti je možno rozšířit pomocí přídatných komunikačních karet, které umožní komunikaci pomocí *TCP/IP protokolů*, *CC-Link* technologií vyvinutou firmou Mitsubishi Electric. Tato technologie byla navržena pro rychlou výměnu dat mezi robotem a programovatelným automatem. (AUTOMA, 2002)

² Reduced Instruction Set Computing

³ Digitální signálový procesor



Obr. 9 Řídicí jednotka CR1-571

2.5.4 Absolutní snímače polohy

Tento druh snímačů nám umožňuje okamžitě vrátit hodnotu pozice bez nutnosti znalosti předchozí pozice snímače. K tomu využívají různých fyzikálních jevů:

- **Odporové snímače** – poloha je vracena ve formě hodnoty odporu. Tento typ snímače je často řešen pomocí lineárního potenciometru.
- **Optické snímače polohy** – poloha je vypočítána pomocí světelného paprsku. Poloha je vypočítána buď na základě vzdálenosti, nebo natočení. U vzdálenosti se pomocí triangulace vypočítává vzdálenost odraženého paprsku. U natočení prochází paprsek poloprůsvitným kotoučem, který ovlivňuje průchod paprsku kotoučem. Tato změna je detekována pomocí snímačů na druhé straně.
- **Akustické snímače polohy** – tyto snímače obsahují vysílač a přijímač. Vysílač vysílá ultrazvukový paprsek a přijímač na základě vlnové délky, fáze a doby paprsku vyhodnocuje polohu.
- **Indukční snímače polohy** – princip těchto snímačů je založený na magnetickém obvodu, kde je jeho indukčnost přímo úměrná počtu závitů cívky. Na základě tohoto jevu je vypočítávána poloha.
- **Kapacitní snímače polohy** – měření polohy je závislé na kapacitě kondenzátoru. Jeho hodnota kapacity je závislá na vzdálenosti a ploše elektrod. (Pikula, 2009)

2.5.5 Program RT-Toolbox

Počítačový program RT-Toolbox je základní program dodávaný společně s robotem. Tento program má za úkol zajistit základní ovládání, údržbu, vytváření programů pro řízení manipulátoru a úpravu registrů robota.

Aby bylo možné vůbec pracovat s robotem je potřeba nastavit připojení mezi počítačovou stanicí s nainstalovaným programem a řídicí jednotkou. K tomu je využito komunikační rozhraní *RS-232*.

Mezi servisní úkoly, které lze provádět pomocí RT-Toolboxu je čtení a nastavení čítače doby běhu robota, úprava výchozích parametrů robota, nastavení výchozí polohy, nastavení nulového bodu souřadného systému, posunutí výchozí polohy souřadného systému, atd.

Vývojové prostředí nabízí *off-line* a *online* část. V *off-line* části můžeme provádět vývoj programů pro řízení a provádět jejich simulaci, abychom ověřily správnost jejich funkce. V *online* části můžeme sledovat průběh vykonávaného programu, ovládat jednotlivé části manipulátory, sledovat informace o stavu manipulátoru, můžeme si zobrazit obsah jednotlivých registrů robota, zobrazit si telemetrická data robota z příslušných senzorů, atd.

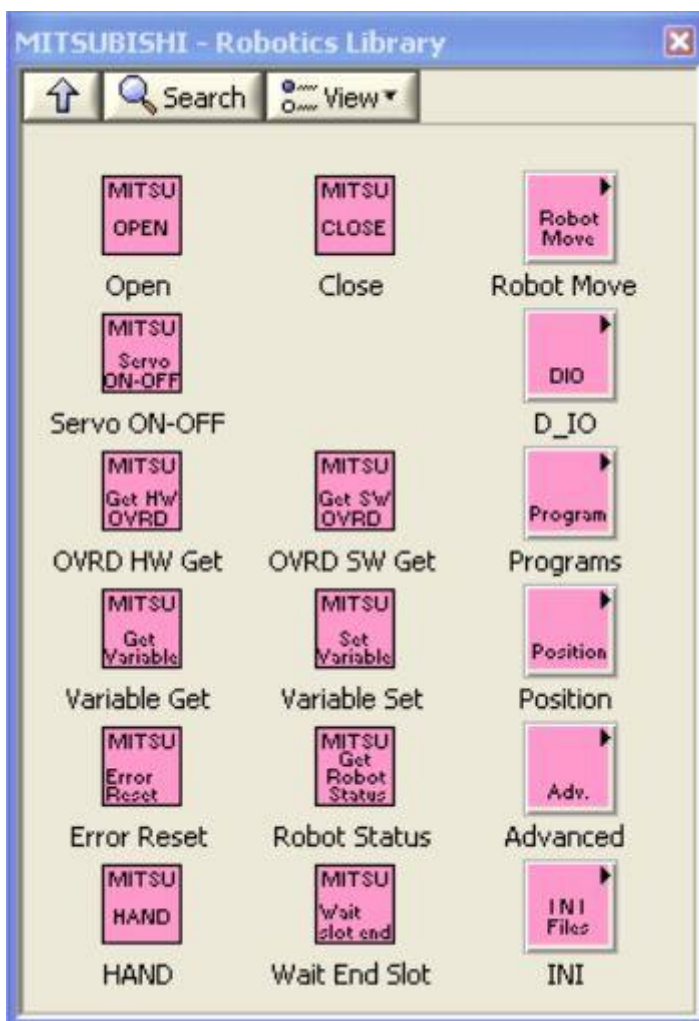
Program neobsahuje rozsáhlejší dokumentaci, proto je jeho nasazení a pochopení obtížnější. Mezi další výrazné nedostatky programu patří absence možnosti jakékoliv vnější komunikace.

2.5.6 ImagingLab Robotics Library

Tato knihovna je vyvíjena italskou firmou ImagingLab srl. Tato firma se specializuje na vývoj v oblasti robotiky a strojového vidění. ImagingLab Robotics Library je vyvinutá knihovna instrukcí pro program National Instruments (zkráceně NI) Labview. Knihovna umožňuje jednoduše ovládat a spravovat roboty od firem jako jsou MITSUBISHI, DENSO a KUKA.

Pomocí této knihovny můžeme tedy jednoduše integrovat ovládání a monitorování robotů do aplikací vyvíjených v prostředí Labview. Díky sloučení ovládání, monitorování a HMI⁴ je zajištěno jednoduché a rychlé vytváření aplikací pracujících s roboty. Knihovna podporuje různé typy robotů od zmíněných výrobců. Při vytváření aplikací musíme akorát zadat typ robota, jeho stupeň volnosti a typ komunikace s robotem, který budeme používat. Poté můžeme z funkční palety (Obr. 10) Labview vybírat obsažené prvky a sestavit z nich naši aplikaci. Výhodou pro využití Labview prostředí pro ovládání robota, je možnost kombinovat různé systémy a lze jednoduše zajistit vnější komunikaci. (ImagingLab, 2015)

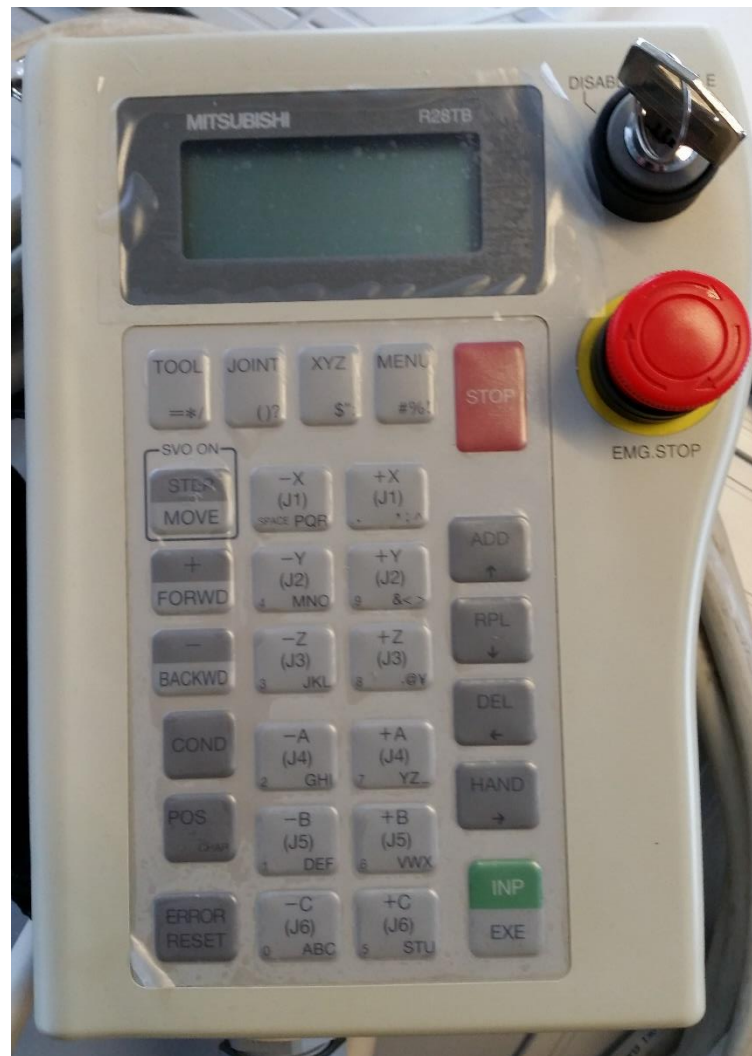
⁴ Human Machine Interface



Obr. 10 Funkční paleta pro robota Mitsubishi v prostředí Labview (ImagingLab, 2015)

2.5.7 Ovládací panel robota

Součástí řídicí jednotky robota je přenosný ovládací panel, který je pomocí kabelu připojený k jednotce. Ovládací panel umožňuje manuální řízení robota v reálném čase. Pohyb robota je možný buď přímým zadáním souřadnic X, Y, Z, na které má robot najet nebo nastavení úhlu natočení jednotlivých kloubů (J1 – J6) manipulatoru. Kromě pohybu je možné pomocí panelu provádět jednoduchou údržbu a nastavení jednotlivých registrů robota.



Obr. 11 Ovládací panel robota

Aby bylo možné provádět jakékoliv operace pomocí panelu, je potřeba nastavit klíč na řídicí jednotce do pozice *teach* a klíč na panelu do pozice *enable*. Poté je potřeba pomocí ruky zajistit stisknutí tzv. přepínače mrtvého muže. Ten funguje jako ochranný prvek. Bez jeho sepnutí není možné provádět žádné akce pomocí ovládacího panelu, nebo dojde k přerušení aktuální činnosti.

3 Řídicí programy pro moduly linky

3.1 Simatic S7-300

Simatic S7-300 je modulární PLC systém vyvíjený firmou Siemens. Tento univerzální systém umožňuje jednoduchou efektivní instalaci a rozsáhlé možnosti rozšíření pomocí modulů, které umožňují přizpůsobit systém přesně na míru našim potřebám.

CPU modul řady S7-300 nabízí celkem sedm standartních CPU, pět bezpečnostních CPU a tři technologická CPU. CPU mají k dispozici velký výkon, který jim umožňuje dosahovat krátkých PLC cyklů. Standardně je podporováno až osm modulů, ale podle potřeby je pomocí expanzních jednotek zajištěno připojení až dalších třiceti dvou jednotek. Mezi přídatné moduly můžeme počítat:

- **Signální moduly** – tyto moduly zajišťují rozhraní mezi CPU modulem a procesem, který řídí. Tyto moduly mohou mít k dispozici osm až šedesát čtyři digitálních kanálů, nebo dva až osm analogových kanálů.
- **Speciální moduly** – tyto moduly bývají někdy označovány jako simulační moduly, které jsou vybaveny přepínači umožňující simulovat běh procesu.
- **Funkční moduly** – tyto inteligentní moduly jsou specializované na vykonávání specifických úkolů, jako například rychlé počítání impulzů, PID regulátory, měření, kontrola pohybu, atd. Výhodou použití těchto modulů je odlehčení práce, kterou by jinak musel vykonávat CPU modul.

Simatic S7-300 navíc nabízí mnohé komunikační protokoly:

- **Ethernet protokol** (*S7, PROFINET, IE-Industrial Ethernet*) – protokol využívá *TCP/IP protokol*, který se vyznačuje rychlostí přenosu a nízkou latencí.
- **Profibus** – pro tuto komunikaci je potřeba mít v PC nainstalovanou speciální *Profibus* kartu. Pro komunikaci využívá *RS-485(IS)* rozhraní, *MBP*⁵ nebo optické vlákna.
- **MPI komunikace** – je to derivát Profibus komunikace založený na komunikačním rozhraní *RS-485*.
- **PPI komunikace** – tato komunikace je využíváný u nižších řad S7 PLC systémů, které ke komunikaci využívá standartní sériové rozhraní.
- **Sériová komunikace** – u některých Simatic S7 PLC systémů lze pro komunikaci využít *RS-232* rozhraní. (Siemens, 2014)

⁵ Manchester Coding and Bus Powered

3.1.1 Simatic S7 bloky

Řídicí programy pro Simatic S7 (dále jen S7) se zakládají ze specifických bloků. Ty rozlišujeme na:

- **Systémové bloky** – hlavní bloky obsahující specifické instrukce a procedury (diagnostika, přerušení, inicializace registrů, atd.) nutné pro chod celého PLC systému.
- **Organizační bloky (OB)** – tyto bloky jsou základním stavebním kamenem řídicích programů. Tyto určují běh programu, vykonávání kroků nutných při spuštění programu, zajištění přerušení, atd.
- **Funkce (FC)** – tyto bloky slouží pro vykonání specifické funkce s návratovou hodnotou. Proměnné použité v tomto bloku jsou globální.
- **Funkční bloky (FB)** – stejné jako FC, ale rozdíl spočívá v tom, že proměnné v tomto bloku jsou lokální.
- **Datové bloky (DB)** – tyto bloky slouží pro organizaci a úložiště proměnných, které potřebujeme použít v řídicím programu.

Díky využití rozdělení řídicího programu do výše zmíněných bloků je zajištěno efektivní a přehledné vytvoření programu. (Berger, 2012)

3.1.2 Simatic S7 proměnné

S7 obsahuje řadu proměnných, které jsou uloženy v operační paměti obsažené v CPU modulu. Každá proměnná obsahuje:

- **Identifikátor** – říká nám, o jakou proměnnou se jedná. Rozlišujeme:
 - Input memory (I) – tyto proměnné reprezentují všechny vstupní digitální signály.
 - Output memory (O) – tyto proměnné reprezentují všechny výstupní digitální výstupy.
 - Peripheral memory (PI/PQ) – tyto proměnné reprezentují přímo hodnoty na jednotlivých vstupních či výstupních modulech. V paměti je pouze uložena adresa proměnné, ale hodnota je čtena přímo z modulu, proto čtení či zápis trvá déle než u proměnných I a Q.
- **Číslo** – určuje, odkud jsou signály čteny nebo kam jsou odesílány. Číslo obsahuje byt a bit oddělený tečkou (např. 124.5).
- **Datový typ** – určují, jakou strukturu pro uložení v paměti proměnné využívají. Jako datové typy využíváme například integer, word, real, S5Time, Char, atd. (Berger, 2012)

3.1.3 Simatic S7 programovací jazyky

S7 umožňuje vytvářet řídicí programy v následujících jazycích:

- **Ladder** – grafický programovací jazyk reprezentující kontaktní schémata. Tato schémata jsou reprezentována pomocí kontaktů a značek reprezentujících specifické funkce, řazených zleva doprava, vytvářející logiku.
- **Function Block Diagram** – grafický programovací jazyk reprezentující bloková schémata, jako je tomu například u elektrických obvodů. Tyto bloky mají vstupní signály, které jsou zpracovávány uvnitř funkce a ta je převádí na výstupní signály.
- **Instruction List** – tento programovací jazyk je reprezentovaný seznamem instrukcí, které jsou zpracovávány řádek po řádku. Každý řádek musí obsahovat návěští, operátor doplněný modifikátorem a operandem.
- **Structured Control Language** – tento jazyk je určený pro vytváření komplexních algoritmů, jako je tomu u vyšších programovacích jazyků (Pascal, C++, atd.).
- **S7-Graph** – grafický programovací jazyk, který je reprezentovaný posloupností kroků, které se musí vykonat. Tyto kroky obsahují podmínky, popřípadě funkce. Kroky mohou být rozděleny do současných nebo paralelních stromů. (Berger, 2012)

3.2 Řídicí programy linky FMS-200

Pro vytvoření řídicích programů pro linku FMS-200 bylo použito vývojové prostředí TIA Portal v11 od firmy Siemens. Toto vývojové prostředí díky integraci prostředí Simatic Step7 a Simatic WinCC umožňuje vytvářet programy pro řízení a monitorování automatizačních procesů a zařízení.

3.2.1 Požadavky na řídicí programy

Námi vytvořené řídicí programy musí splňovat následující požadavky:

- Vytvořené programy musí být plně automatizované, tak aby obsluha musela jen pomocí tlačítka spustit běh linky
- Musí zaručit bezpečnost obsluhy a samotného vybavení linky
- Vytvoření bodů, které umožní přesněji monitorovat běh linky
- Zohlednit komunikaci mezi jednotlivými moduly linky FMS-200
- Umožnit vizualizaci linky pomocí softwaru třetích stran

3.2.2 Příprava pro vytváření řídicích programů

Než začneme vytvářet samotné řídicí programy, musíme se prvně seznámit s funkcí linky, pro kterou je chceme vytvářet. K tomu nám pomůže dokumentace dodávaná společně s linkou. V dokumentaci najdeme:

- Krátký popis vybavení linky (senzory, atd.)

- Elektrické schéma připojení jednotlivých částí vybavení linky do vstupních a výstupních kanálů periferních modulů PLC systému
- Schéma funkce dané linky

Na základě funkčního schéma linky dodávané výrobcem si objasníme, jak fungují jednotlivé části linky. Na základě tohoto zjištění si poté vytvoříme vlastní funkční schéma linky, které odpovídá našim požadavkům na řízení a zároveň je zajištěna funkčnost linky.

Dalším nutným krokem je ověření správného zapojení vstupních a výstupních signálů linky do PLC systému. K tomu nám poslouží schéma elektrického zapojení. Na základě tohoto schématu si pomocí indikátorů umístěných na PLC modulu ověříme, zdali kanál opravdu odpovídá elektrickému schématu zapojení. U kanálů, které nejsou aktivní, si musíme pomoci vytvořením jednoduchých PLC programů, které nám například pomocí stisku tlačítka na ovládacím panelu linky aktivují příslušný kanál.

Posledním krokem je zvolit typ komunikace. Pro komunikaci byla vybrána technologie průmyslového ethernetu, který umožní rychlou komunikaci jak mezi samotnými moduly tak i případnými zařízeními třetích stran. Každý CPU modul má již z výroby přiřazenou IP adresu, kterou si musíme zjistit. Kdyby ji neměl, musely bychom využít jiný způsob komunikace, díky kterému bychom se dostali do nastavení PLC modulu, kde bychom ji musely nastavit. Ke zjištění IP adresy jednotlivých modulů je vývojové prostředí TIA Portal vybaveno tlačítkem „Scan For Accessible Devices“, které nám zjistí všechna PLC zařízení, která se nachází v naší síti. Na základě zjištěných adres přiřadíme adresu a masku naší PC stanici, kterou potřebujeme pro následnou komunikaci (viz níže).

Tab. 1 IP adresy jednotlivých modulů linky FMS-200

Modul	IP adresa	Maska sítě
FMS-202	130.130.130.2	255.255.255.0
FMS-205	130.130.130.5	255.255.255.0
FMS-206	130.130.130.6	255.255.255.0
PC stanice	130.130.130.100	255.255.255.0

3.2.3 Hardwarová konfigurace modulů

Součástí každého řídicího programu je nutná hardwarová konfigurace, která musí přesně odpovídat sestavení PLC systému, kterým je linka vybavena, aby mohla fungovat komunikace mezi CPU modulem a jednotlivými moduly tvořící PLC systém. U každého modulu je nutné dodržet:

- O jaký modul se jedná
- Přesný typ daného modulu
- Specifikaci modulu

V následujících tabulkách jsou uvedené PLC moduly použité na jednotlivých modulech linky FMS-200.

Tab. 2 Vybavení modulu FMS-202

Modul	Typ modulu	Specifikace
CPU	CPU315-2 PN/DP	315-2EH14-0ABO
Digitální vstup	DI 16xDC24V	321-1BH02-0AAO
Digitální výstup	DO 16xDC24v/0.5A	322-1BH01-0AAO
Analogový výstup	AI 2x12BIT	331-7KB02-0ABO

Tab. 3 Vybavení modulu FMS-205

Modul	Typ modulu	Specifikace
CPU	CPU315-2 PN/DP	315-2EH14-0ABO
Digitální vstup	DI 16xDC24V	321-1BH02-0AAO
Digitální vstup	DI 16xDC24V	321-1BH02-0AAO
Digitální výstup	DO 16xDC24v/0.5A	322-1BH01-0AAO
Counter modul	FM 305-1	350-1AH03-0AE0

Tab. 4 Vybavení modulu FMS-206

Modul	Typ modulu	Specifikace
CPU	CPU315-2 PN/DP	315-2EH14-0ABO
Digitální vstup	DI 16xDC24V	321-1BH02-0AAO
Digitální výstup	DO 16xDC24v/0.5A	322-1BH01-0AAO

U CPU modulů je dále nutné v jejich vlastnostech nastavit IP adresu, kterou jsme si zjistili v přípravě (Tab. 1). Bez tohoto kroku by nebylo možné komunikovat s moduly.

U vstupních a výstupních modulů je pouze nutné nastavit rozhraní kanálů, které budeme používat v našich programech. Můžeme si určit vlastní rozsah, nebo můžeme dodržet rozsah, který je použit v elektrickém schématu zapojení kanálů v příslušné dokumentaci.

3.2.4 Řídicí program pro modul FMS-202

Pro vytváření programu ve vývojovém prostředí Simatic Step7 byl využit jazyk LAD. Podkladem pro řídicí program se stalo vytvořené funkční schéma (Obr. 36). Program se skládá z hlavního organizačního bloku, ve kterém jsou umístěny funkce. Program je spuštěn pomocí tlačítka *Start* a může být kdykoliv pomocí tlačítka *Reset* přerušen.

Main (OB1) je hlavní organizační blok, který je zpracováván PLC systémem. Tento blok má za úkol pouze zajistit volání funkcí, které reprezentují jednotlivé funkční části linky.

Default_position (FC1) je funkce, která má za úkol zajistit signalizaci, že se manipulátory nenachází ve výchozí pozici. To je důležité u manipulátorů, které jsou poháněny pomocí dvojčinných pneumatických válců. Tato situace nastane tehdy, kdy samotný manipulátor zůstane v jiné než krajní poloze. To by mělo za následek, že nedojde k sepnutí ani jednoho krajního spínače, který signalizuje polohu manipulátoru řídicímu systému, což naruší funkci celé linky.

Feed (FC2) je funkce, která obstarává proceduru podávání ložisek ze zásobníku. Aby mohlo proběhnout podání ložiska ze zásobníku, je zapotřebí vyjetí *podavače ložisek A*. Úspěšné podání ložiska je následně detekováno mikropsínačem, který je umístěn přímo naproti *podavači A*. Pokud po aktivaci podavače nedojde do jedné a půl sekundy k sepnutí mikropsínače, dojde k rozsvícení červeného *FM světla*, které signalizuje obsluze prázdný zásobník, nebo že došlo k zaseknutí ložiska v zásobníku. Pokud dojde k úspěšné detekci podaného ložiska, je aktivován *kontrolní bod I.*, která signalizuje připravenost link pro další část její funkce.

Transfer1 (FC3) je funkce, která má za úkol zajistit přenesení podaného ložiska ze zásobníku na plošinu, na které se posléze provede měření výšky ložiska. Podmínkou pro vykonání této funkce je nutná aktivace *kontrolního bodu I.* Tato procedura je vykonávána pomocí *přenašeče B*, na kterém je umístěn *uchopovač C*. Pokud dojde k uchycení podaného ložiska, je aktivován *kontrolní bod II.* Po přenesení ložiska na *plošinu D* a jeho zachycení pomocí *vyjíždějícího tělesa F*, dojde k aktivaci *kontrolního bodu III.* Aby mohla plošina *D* vykonávat svou činnost, musí *přenašeč B* odjed do své prostřední pozice. Vykonání toho kroku nám signalizuje *kontrolní bod IV.*

Measurement (FC4) je funkce, která má za úkol změřit výšku ložiska. Funkce se začne provádět pouze v momentu, kdy dojde k aktivaci *kontrolního bodu IV.* Tento bod signalizuje umístění a zajištění ložiska na *plošině D*. Tato plošina poté vyjede nahoru, kde se nachází analogový lineární potenciometr, který je napojený na analogový vstupní modul linky. Hodnota odporu potenciometru nám signalizuje velikost ložiska na plošině. Linka pracuje se dvěma druhy ložisek, proto je možné rozlišit velikost pomocí hraniční hodnoty:

- Pokud je hodnota menší než hraniční hodnota, je ložisko označené jako *malé*
- Pokud je hodnota větší než hraniční hodnota, je ložisko označené jako *velké*

V momentu, kdy dojde k označení velikosti ložiska, je aktivován *kontrolní bod pro měření*. Následně dojde ke sjetí plošiny dolů. Mezitím systém vyhodnotí, zdali má ložisko požadovanou velikost. To je zaručeno pomocí *přepínače Auto/Man*:

- Pokud je přepínač v pozici *auto*, akceptuje linka pouze malá ložiska
- Pokud je přepínač v pozici *man*, akceptuje linka pouze velká ložiska

V momentu, kdy systém rozhodl, zdali ložisko vyhovuje nebo ne a *plošina D* se nachází dole je aktivován *kontrolní bod V.*, který signalizuje připravenost ložiska pro poslední část manipulace.

Transfer2 (FC5) je poslední funkce, která buď vyloučí ložisko, nebo zajistí jeho vložení do připraveného tělesa. Pokud bylo ve *FC4* rozhodnuto o vyloučení tělesa, dojde pomocí *ejektoru E* k vyloučení ložiska a linka již dále nepokračuje v činnosti. To je signalizováno pomocí *kontrolního bodu pro vyloučení*. V opačném případě dojde pomocí *přenašeče H*, na kterém je umístěn *uchopovač I*, k uchycení ložiska na plošinu. Tento krok je signalizován pomocí *kontrolního bodu VI*. Po tomto bodu přeneseme přenašeč ložisko nad připravené těleso, což je signalizováno pomocí *kontrolního bodu VII*. Poté je ložisko uvolněno a vloženo do tělesa. Tento poslední krok je signalizován *kontrolním bodem pro vložení*.

3.2.5 Řídicí program pro modul FMS-205

Pro vytváření programu ve vývojovém prostředí Simatic Step7 byl využit jazyk LAD. Podkladem pro řídicí program se stalo vytvořené funkční schéma (Obr. 37). Program se skládá z hlavního organizačního bloku, ve kterém jsou umístěny funkce. Program je spuštěn pomocí *tlačítka Start* a může být kdykoliv pomocí *tlačítka Reset* přerušen.

Main (OB1) je hlavní organizační blok, který je zpracováván PLC systémem. Tento blok má za úkol pouze zajistit volání funkcí, které reprezentují jednotlivé funkční části linky.

Feed (FC1) je funkce, která má za úkol zajistit podání krytu z vertikálního zásobníku. Po stisknutí *tlačítka Start* dojde k vytažení krytu ze zásobníku pomocí *podavače K* a vynulování čítače pro otočky stolu (viz níže). Naproti podavači se nachází mikropínač, který signalizuje úspěšné podání krytu ze zásobníku. Pokud nedojde k detekci krytu do jedné a půl vteřiny, je pomocí červeného *FM světla* signalizováno obsluze zaseknutí krytu v zásobníku nebo jeho prázdnota. Pokud dojde k detekci krytu je aktivován *kontrolní bod I*.

Load to plate (FC2) je funkce, která po signalizaci úspěšného podání ložiska ze zásobníku zajistí pomocí *podavače A*, na kterém je umístěn *uchopovač C*, zajistí přenesení krytu na otočný stůl. Umístění krytu na stůl je signalizováno pomocí *kontrolního bodu II*.

Plate (FC3) je funkce, která má za úkol otáčení stolu do pozic pro měření, vyloučení nebo vložení do tělesa. Aby bylo možné otočit stolem bylo prvně nutné pochopit princip funkčnosti systému pro otáčení. Aby bylo možné otočit stolem o jednu pozici, je nutné zachycení stolu k otočnému systému pomocí vysunutí *cy-lindru E*, který propojí uchopení stolu k *otočnému systému D*. Aktivací toho systému je zajištěno otočení stolu o jednu pozici. Posun o jednu pozici zajištěn pevnou zarážkou, do které otočný systém narazí. Zde bohužel chybí jakákoliv signalizace, která by naznačovala, že došlo k posunutí stolu o jednu pozici. Proto byl vytvořen jednoduchý časovač, který po jedné vteřině signalizuje, že došlo k otočení o jednu pozici. Následně se musí zasunout *cylindr E* a následně se může *otočný mechanismus D* vrátit do původní pozice a časovač se resetuje. Následně bylo nutné vytvořit si vlastní počítadlo pro otočky stolu. Na základě toho počítadla je zaručeno, že stůl se posune do potřebných pozic:

- Nad třetí pozicí je umístěn *kapacitní snímač*, který nám ověří výskyt krytu na stole. Detekce je signalizována aktivací příslušného vstupního signálu.
- Nad čtvrtou pozicí je umístěn *indukční snímač*, který nám ověří, zdali je kryt z kovu nebo plastu. Detekce je signalizována aktivací příslušného vstupního signálu.
- Nad pátou pozicí je *fotoelektrický snímač*, který má pomocí fyzického přepínače nastavenou pevně barvu, kterou má zachytit. Detekce je signalizována aktivací příslušného vstupního signálu.
- Šestá pozice je určena pro vyloučení krytu
- Sedmá pozice je určena pro vložení krytu do připraveného tělesa

Po tom co kryt projde přes čtvrtou pozici, rozhodne systém na základě pozice *přepínače Auto/Man*, zdali má kryt požadované vlastnosti:

- V pozici *Auto* linka přijímá stříbrné kovové kryty a ostatní vylučuje
- V pozici *Manual* linka přijímá plastové kryty jakékoliv barvy a ostatní vylučuje

Na základě vyhodnocení vlastností krytu se nastaví konečná pozice stolu:

- Pokud je kryt schválen, stůl se zastaví na sedmé pozici, kde je posléze signalizována *připravenost pro přenos do tělesa*
- Pokud je kryt zamítnut, stůl se zastaví na šesté pozici, kde je posléze signalizována *připravenost pro vyloučení krytu*

Evacuate (FC4) je funkce, která má za úkol zajistit vyloučení krytu, který nebyl schválen. Pokud se nastaví šestá pozice stolu jako konečná, dojde k aktivaci *kontrolního bodu pro vyloučení*. V momentu kdy se stůl zastaví v šesté pozici, dojde k aktivaci *kontrolního bodu pro signalizaci stolu v pozici pro vyloučení*. V tuto chvíli najede *vyhazovač F*, na kterém je umístěna *vakuová přísavka V*, nad stůl s umístěným krytem. Pomocí této přísavky je kryt uchycen a přenašeč jej přenesse na pozici, kde jej přísavka uvolní. Vyloučení krytu je signalizováno pomocí *kontrolního bodu pro vyloučení ložiska*. Linka poté už dál nepracuje.

Unload (FC5) je funkce, která má za úkol zajistit vložení schváleného krytu do připraveného tělesa. Pokud dojde k nastavení sedmé pozice jako konečné, je aktivován *kontrolní bod pro vložení do tělesa*. V momentu kdy se stůl nachází v sedmé pozici, dojde k aktivaci *kontrolního bodu pro signalizaci stolu v pozici pro přenos*. Přenos do tělesa je zajištěn pomocí *přenašeče I* a *ukladače H*, na kterém je umístěn *uchopovač J*. Vložení do tělesa je signalizováno aktivací *kontrolního bodu pro přenos*.

Data_block_1 (DB1) je datový blok, do kterého se ukládají informace o pozici stolu a data potřebná pro naše vytvořené počítadlo.

3.2.6 Řídicí program pro modul FMS-206

Pro vytváření programu ve vývojovém prostředí Simatic Step7 byl využit jazyk LAD. Podkladem pro řídicí program se stalo vytvořené funkční schéma (Obr. 38). Program se skládá z hlavního organizačního bloku, ve kterém jsou umístěné funkce. Program je spuštěn pomocí *tlačítka Start* a může být kdykoliv pomocí *tlačítka Reset* přerušen.

Main (OB1) je hlavní organizační blok, který je zpracováván PLC systémem. Tento blok má za úkol pouze zajistit volání funkcí, které reprezentují jednotlivé funkční části linky.

Feed (FC1) je funkce, která má zajistit podání šroubků z vertikálního zásobníku. To je zajištěno pomocí *podavače D*, který je složený ze dvou západek, které jsou připojené na protichůdné válcové pohony. Předtím než můžeme podat šroubek ze zásobníku, musíme zajistit, že je *přenašeč E* ve výchozí pozici pod zásobníkem, jelikož je na něm umístěn detektor pro prezenci šroubku. Pokud je ve výchozí pozici, dojde k aktivaci *kontrolního bodu I*. Když je *přenašeč E* ve výchozí pozici a je stisknuto tlačítko *Start*, dojde k podání šroubku ze zásobníku. Pokud nedojde k detekci šroubku do jedné vteřiny od aktivace podavače šroubku ze zásobníku, dojde k rozsvícení červeného *FM světla*. To signalizuje obsluze, že je zásobník prázdný nebo došlo k zaseknutí šroubku v zásobníku. Při úspěšné detekci je aktivován *kontrolní bod II*. Následně je potřeba ještě ověřit zdali je podavač *A* ve výchozí pozici, popřípadě zajistit jeho najetí do této pozice.

Move (FC2) je funkce, která po úspěšném podání šroubku zajistí jeho přenos do připraveného tělesa. Tento proces spočívá v najetí *přenašeče E* s podaným šroubkem pod *přenašeč A*. V momentu kdy se nachází *přenašeč E* pod *přenašečem A* dojde k aktivaci *kontrolního bodu III.*, který signalizuje připravenost pro přenos do tělesa. Pomocí *podavače B* a *uchopovače C* umístěným na *přenašeči A* je zajištěno uchycení a vyzvednutí šroubku z *podavače E*. Poté kdy najede *přenašeč A* nad připravené těleso, je aktivován *kontrolní bod IV.*, který signalizuje připravenost pro vložení do tělesa. Poté se provede vložení šroubku do tělesa a je aktivován *finální bod* signalizující úspěšné provedení celé operace linky.

3.3 Řídicí program pro robota MELFA v Labview

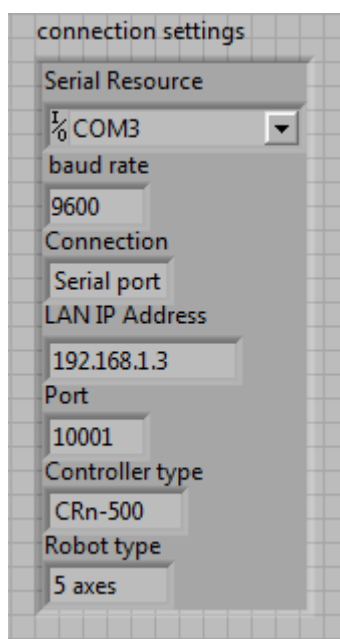
Vývojové prostředí Labview se skládá z blokového diagramu a čelního panelu. Čelní panel je určen pro vytvoření vizualizace a ovládání námi vytvářeného přístroje. Blokový diagram obsahuje zdrojový kód, který tvoří funkční logiku přístroje.

Aby bylo možné vytvořit základní vizualizační program pro obsluhu robota MELFA, je potřeba zajistit získání dat o aktuální pozici koncového efektoru. Kromě aktuálních souřadnic pozice koncového efektoru bude potřeba předat koncovému efektoru souřadnice pozice, na kterou chceme, aby se přemístil. Pro komunikaci s OPC serverem jsme využily modul DataSocket v programu Labview (kapitola 4.6.3). Tyto programy získávají data nebo předávají data funkci nebo

proceduře, která je v podstatě další program vytvoření v prostředí Labview, který má nastavené vstupy nebo výstupy, odkud čte nebo posílá data.

Pro tyto programy využijeme modul ImagingLab Robotics Library, který je reprezentován v programu Labview příslušnou paletou. Z této palety vybereme základní prvky, které musí obsahovat každý program:

- **Prvek na otevření spojení s robotem** – tento prvek má na vstupu nastavení spojení s robotem (Obr. 12) a k němu příslušný identifikátor robota, se kterým pracujeme. U nastavení spojení musíme zvolit port pro komunikaci, rychlost datového toku, druh spojení, případně IP adresu a port robota, typ ovládacího kontroléru a počet stupňů volnosti robota.
- **Prvek na uzavření spojení s robotem** – po vykonání obsahu řídicího programu je nutné zajistit ukončení spojení s robotem.
- **Prvek na ovládání servomotorů robota** – aby bylo možné získat různá data, nebo vykonat pohyb robota je vždy nutné prvně zapnout servomotory, jinak nebude vykonán pohyb nebo nebude možné získat potřebná data. Zapnutí nebo vypnutí zajistíme přiřazením proměnné *Open/Close* prvku.



Obr. 12 Ukázka nastavení připojení pro robota MELFA v ImagingLab Robotics Library v řídicím panelu Labview

Všechny prvky ovládání robota je nutné propojit stejným identifikátorem, aby byla zajištěna jejich návaznost pro řízení příslušného robota. Díky tomuto můžeme například využít řídicí program pro více robotů.

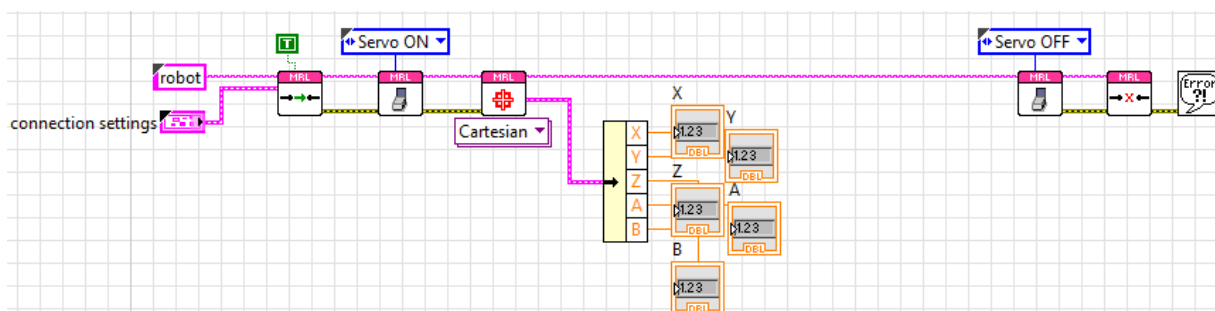
3.3.1 Program pro získání aktuálních souřadnic koncového efektoru

Pokud chceme zjistit aktuální pozici koncového efektoru robota, musíme vytvořit nový Labview program, který bude fungovat jako funkce, která bude vracet následující hodnoty:

- Pozici na osách x, y a z koncového efektoru v souřadném systému
- Úhel A značící natočení koncového efektoru kolem osy x
- Úhel B značící natočení koncového efektoru kolem osy y

Do blokového diagramu programu (Obr. 13) musíme kromě výše zmíněných prvků zajišťující nastavení spojení a identifikátoru robota, otevření komunikace a zapnutí servomotorů, přidat ještě jeden prvek vracející pozici koncového efektoru v souřadném systému, který si u tohoto prvku zvolíme. Pro náš program byl vybrán kartézský souřadný systém. Tento prvek nám vrátí hodnoty v datovém poli. Na jednotlivé prvky tohoto pole musíme napojit indikátory, které se nám zobrazí na řídicím panelu, odkud můžeme tyto indikátory nastavit jako výstupní hodnoty. Tím je zajištěno možnost použití programu jako funkce, která bude mít výstupní hodnoty signalizující polohu koncového efektoru ve zvoleném souřadném systému.

Následně musíme ještě pomocí prvku pro ovládání servomotorů zajistit jejich vypnutí a nakonec přidat prvek, který uzavře komunikaci s robotem.



Obr. 13 Ukázka blokového diagramu funkce pro získání pozice robota MELFA v kartézském souřadném systému

3.3.2 Program pro přesun koncového efektoru na zadané souřadnice

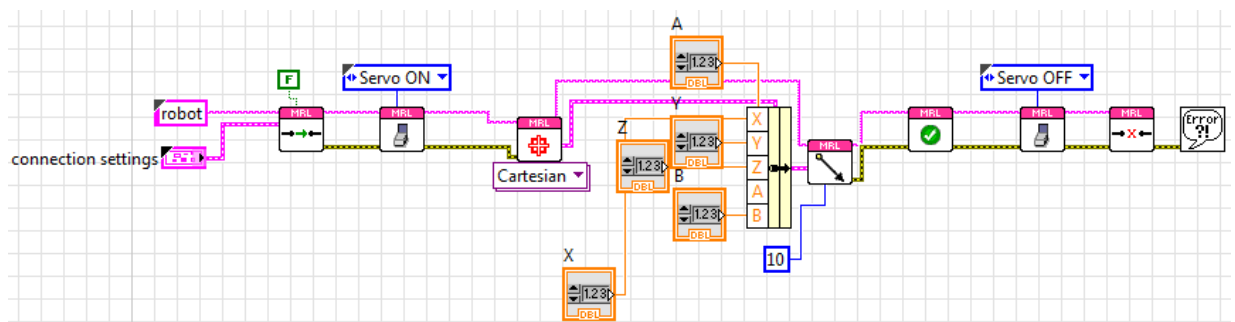
Pro zajištění pohybu koncového efektoru robota MELFA na pozici v souřadném systému potřebujeme vytvořit řídicí program v Labview, který bude fungovat jako procedura, které na základě předaných parametrů zajistí pohyb efektoru na pozici. Jako parametry bude procedura vyžadovat:

- Pozici na osách x, y a z kam se má koncový efektor robota přemístit
- Úhel A, který má koncový efektor na cílových souřadnicích svírat vůči ose x

- Úhel B, která má koncový efektor na cílových souřadnicích svírat vůči ose y

Do blokového diagramu musíme tedy kromě výše zmíněných nutných prvků přidat prvek pro získání souřadného systému, který zde využijeme pouze jako zdroj prázdného datového pole pro kartézský souřadný systém. Každému prvku datového pole přiřadíme opět indikátor, který se nám zobrazí v řídicím panelu programu Labview. Zde tyto indikátory nastavíme jako vstupní parametry, díky kterým tento program můžeme využít v jiných jako proceduru s parametry.

Poté co máme pole naplněné hodnotami, zajistíme propojení tohoto pole na prvek zajišťující pohyb robota. Ten zpracuje vstupní souřadnice a zajistí pohyb robota na tyto souřadnice. Na výstup prvku pro pohyb je nutné zajistit jeho napojení na další prvek, který zajišťuje signalizaci dokončení provedení předešlého prvku. Na tento prvek navážeme prvek pro ovládání servomotorů, který zajistí jejich vypnutí a nakonec prvek pro ukončení komunikace s robotem.

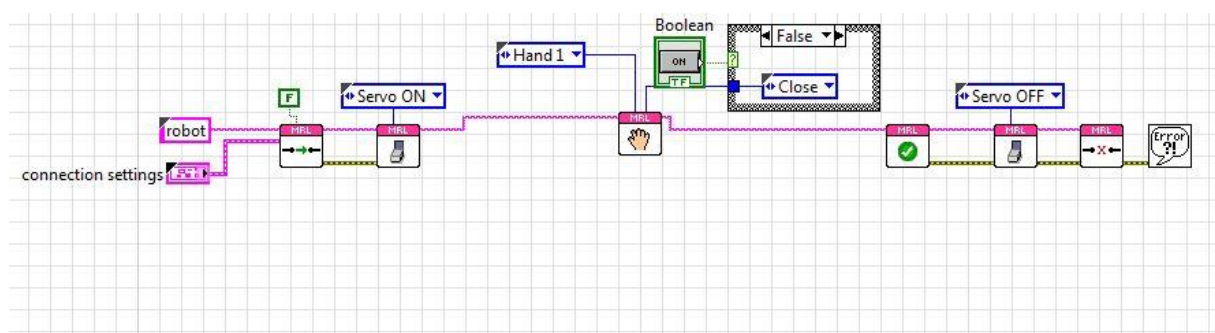


Obr. 14 Ukázka blokového diagramu procedury pro pohyb robota MELFA na zadané kartézské souřadnice

3.3.3 Program pro ovládání sevření nebo rozevření koncového efektoru

Koncový efektor u robota MELFA umožňuje umístění tzv. nástroje, který zajistí, že robot bude schopen vykonávat různé manipulační nebo montážní procedury. Tyto nástroje, které robot MELFA používá, mají uložené hodnoty v registrech robota. Ty obsahují údaje o jejich rozměrech. Protože nástroje můžeme volně měnit, bylo by velice nepraktické ručně přepočítávat souřadnice koncového efektoru. Proto je důležité robotovi říci, který nástroj používáme, a řídicí jednotka robota na základě údajů z registru automaticky přepočítá souřadnice koncového efektoru, kterým je v tu chvíli náš zvolený nástroj.

Program, který nám umožní ovládat sevření nebo rozevření koncového efektoru, musí kromě prvků pro ovládání servopohonů robota, prvku pro čekání na vykonání akce, obsahovat prvek pro obsluhu nástroje (Obr. 15). Tento prvek potřebuje na vstupu zadat číslo nástroje, se kterým robot pracuje. Následně už jen stačí pomocí indikátoru nastavit otevření nebo zavření nástroje. Tento indikátor si můžeme umístit do *switch* struktury, pomocí které můžeme měnit vstup tohoto indikátoru na základě stavu proměnné na OPC serveru.



Obr. 15 Ukázka blokového diagramu pro ovládání sevření nebo rozevření koncového efektoru robota MELFA

4 Komunikace mezi komponentami linky

Abychom mohly řídit a monitorovat výrobní linku FMS-200, musíme zajistit komunikaci mezi jednotlivými komponentami. Linka se skládá z komponent od různých výrobců, proto je zapotřebí zvolit komunikační technologii, která umožňuje komunikaci mezi různými druhy zařízení. Proto jsme pro naši komunikaci zvolily OPC server, který díky možnosti instalace ovladačů dokáže komunikovat s jakoukoliv komponentou, pro kterou existuje zmíněný ovladač, a máme k dispozici potřebné komunikační technologie.

4.1 OPC architektura

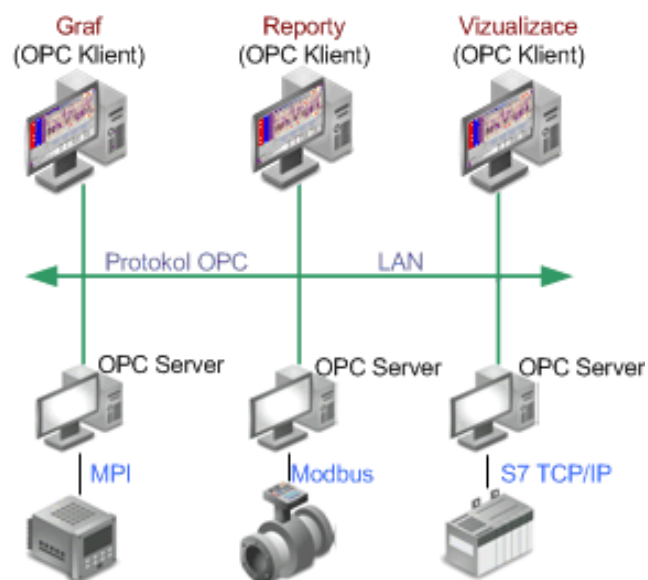
OPC server je druh komunikační technologie určený pro automatizační procesy a technologie. Je spravován a vyvíjen organizací OPC Foundation. Tato organizace byla založena v roce 1990. Jejím hlavním cílem je vytvořit a spravovat specifikace, pomocí kterých bude zajištěna jednotnost při aplikaci a vývoji OPC technologie v automatizačních procesech a technologiích. OPC zajišťuje soulad se stanovenými specifikacemi a provádí jejich ověřování pomocí stanovených zkoušek a postupů. Organizace také zajišťuje úzkou spolupráci s vedoucími průmyslovými subjekty na poli průmyslové automatizace. (OPCFoundation, 2015)

OLE for Process Control (OPC) je specifický komunikační protokol s cílem zajistit jednotné komunikační rozhraní mezi průmyslovým automatizačním hardwarem a softwarem určeným pro jejich řízení a monitorování. Tento protokol je založen na architektuře *klient – server* a je volně dostupný komukoliv bez nutnosti platby licenčních poplatků. Nasazení OPC komunikace nevyžaduje instalaci žádných specifických softwarových nebo hardwarových prvků do stávajících průmyslových strojů nebo technologií. Pro komunikaci s prvky je využíván již stávající druh komunikace. Jakákoliv nutnost instalace ovladačů probíhá na straně OPC serveru. (Foxon, 2014)

Technologicky OPC standart vychází z Microsoft COM technologie. COM je objektově orientovaný a určený pro komunikaci mezi softwarovými komponentami. Výhodou této technologie je, že může být nasazena v jiném prostředí, než ve kterém byla vytvořena. Každý objekt je zodpovědný za svoje vytvoření a zničení. Zároveň obsahuje rozhraní, přes který k němu mohou přistupovat vnější objekty. Tím je zajištěno použití objektů bez potřebné znalosti jejich vnitřní struktury. Technologie COM podporuje rozsáhlý počet programovacích jazyků. Z COM technologie vychází:

- DCOM
- OLE Automation
- ActiveX
- COM+

- A další



Obr. 16 Ukázka OPC architektury klient – server (Foxon, 2014)

4.1.1 OPC klient

Jako OPC klient bývá často označován software, který je schopný přijímat, případně zasílat, data dostupná na OPC serveru. Tato data jsou nadále zpracovávána a reprezentována uživateli. OPC klient se může připojit k více OPC serverům najednou. (Foxon, 2014)

4.1.2 OPC server

OPC server je software, který obsahuje nutné ovladače pro komunikaci se zařízeními, ke kterým je server připojen. Data získaná ze zařízení jsou poté převedena do OPC formátu a uložena na serveru. Tato data jsou podle potřeby aktualizována. Kromě čtení těchto dat, je OPC server schopen provádět i zápis. Server umožňuje typy komunikace:

- **Synchronní komunikace se zařízením** – čtení a zápis je dokončen, až když jsou data přenesena do nebo ze zařízení. Tzn. komunikace je blokující.
- **Synchronní komunikace s vyrovnávací pamětí** – klientům jsou data předávána z vlastní vyrovnávací paměti, ta musí být periodicky obnovována.
- **Asynchronní komunikace** – druh neblokující komunikace. Není potřeba čekat na potvrzení čtení nebo zápisu.
- **Periodická komunikace** – data jsou periodicky čtena do vyrovnávací paměti nebo zapisována do zařízení. (Moravské přístroje, 2005)

K OPC serveru se může naráz připojit více různých klientů. Zároveň není problém zařadit do stejné sítě více OPC serverů, což by bylo zapotřebí v případech, že potřebujeme nasadit různé OPC kvůli podpoře ovladačů pro připojení k různému hardwaru.

OPC servery umožňují vzdálený přístup i klientům umístěným v síti, ve které se nachází. To je umožněno použitím DCOM (Distributed COM) technologie. DCOM vychází z technologie COM a jejím zaměřením je komunikace mezi softwarovými komponentami rozmístěným v síti.

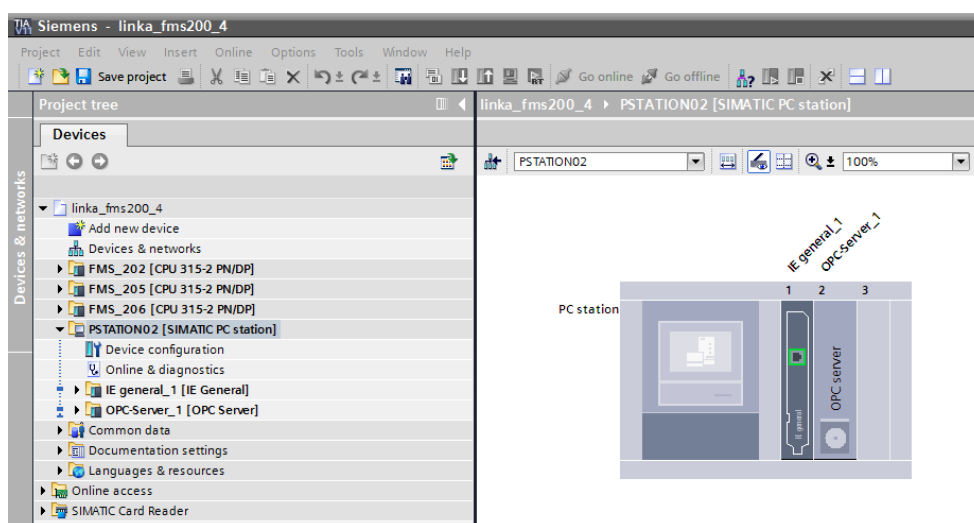
4.2 Siemens S7 Simatic.Net OPC Server

Součástí Siemens Simatic S7 PLC systému je i WinCC. WinCC byl navržen a slouží pro vytváření HMI rozhraní. WinCC umožňuje pomocí různých typů komunikačních protokolů a technologií vytvářet vizualizace a řízení průmyslových automatizačních procesů a technologií. Součástí protokolů pro komunikaci je i zmíněný Simatic.Net OPC server, který obsahuje všechny potřebné ovladače a protokoly pro komunikaci se všemi PLC systémy řady Simatic S7.

4.2.1 Inicializace a nastavení S7 Simatic.Net OPC serveru

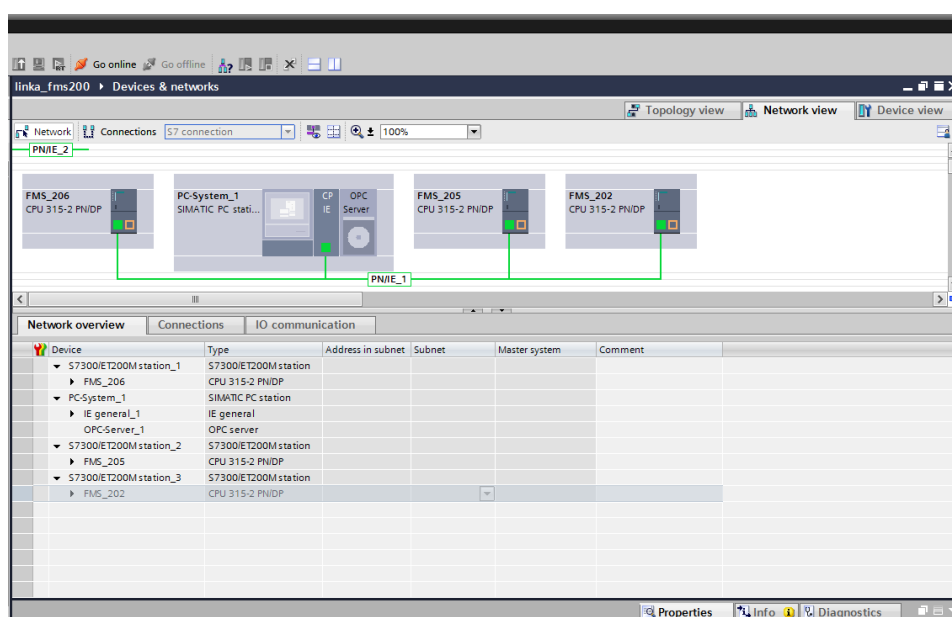
Poté, co máme úspěšně nastavenou hardwarovou konfiguraci a vytvořené řídicí programy našich PLC systémů, můžeme přistoupit k nastavení a inicializaci OPC připojení. Jako první krok je nejdůležitější do naší sítě v programu TIA Portal přidat nové zařízení. Toto zařízení reprezentuje naši počítačovou stanici, na které poběží OPC server. Proto v nabídce nových zařízení vybere, které chceme přidat do našeho projektu, PC station.

V hardwarové konfiguraci poté musíme do PC stanice přidat modul reprezentující síťovou kartu a modul reprezentující OPC server (Obr. 17). Dalším velice důležitým krokem je zadat do nastavení modulu reprezentující síťovou kartu IP adresu našeho počítače. Bez toho kroku by nebyla možná komunikace mezi OPC serverem, klienty a PLC systémy. V modulu síťové karty je tedy potřeba zadat naši zvolenou IP adresu pro PC stanici, kterou jsme zároveň zadaly do nastavení síťového připojení v našem operačním systému.



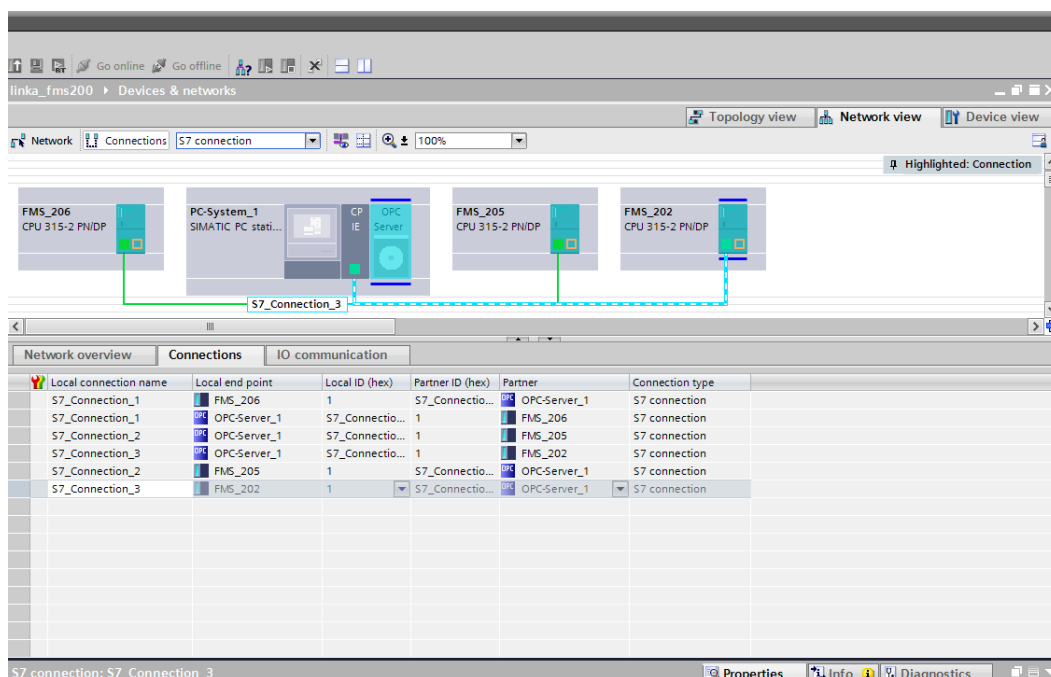
Obr. 17 Hardwarová konfigurace PC station

Poté, co jsme nastavily hardwarovou konfiguraci PC station, musíme ještě zajistit přejmenování tohoto zařízení tak, aby se jeho jméno shodovalo se jménem, které nese Station Configuration Editor (viz níže). Poté se přepneme do záložky zařízení a sítě, kde bychom měli vidět propojení našich PLC systému a PC station (Obr. 18).



Obr. 18 Zobrazení síťového propojení všech našich komponent

Nyní musíme mezi jednotlivými PLC systémy a PC Station nastavit S7 connection (Obr. 19). Toto připojení je určené pro OPC komunikaci. To provedeme výběrem položky connection, poté vedle v rolovacím menu vybereme S7 connection. Nyní už nám zbývá pomocí myši propojit každé PLC zařízení s PC station.



Obr. 19 Zobrazení výsledného propojení komponent pomocí S7 connection

Výsledkem jsou tři S7 connection:

- *S7_Connection_1* – zajišťuje propojení mezi linkou FMS-206 (linka na vkládání ložisek) a OPC serverem umístěným na PC Station
- *S7_Connection_2* – zajišťuje propojení mezi linkou FMS-205 (linka na vkládání krytů) a OPC serverem umístěným na PC Station
- *S7_Connection_3* – zajišťuje propojení mezi linkou FMS-202 (linka na vkládání šroubků) a OPC serverem umístěným na PC Station

Toto rozdělení připojení je důležité pro následný přístup k jednotlivým hodnotám a atributům příslušných linek na OPC serveru. Poté už nám zbývá jen v nastavení jednotlivého připojení zajistit, aby bylo zatrženo aktivní vytvoření připojení. Zároveň si zde můžeme ověřit nastavení připojení a popřípadě upravit jeho hodnoty. Následně nastavení PC station vykompilujeme, abychom zjistili případné chyby v nastavení. Když máme úspěšně nastavení PC stanice v TIA Portalu, můžeme přistoupit na nastavení PC Station Configuration.

4.2.2 PC Station Configuration Editor

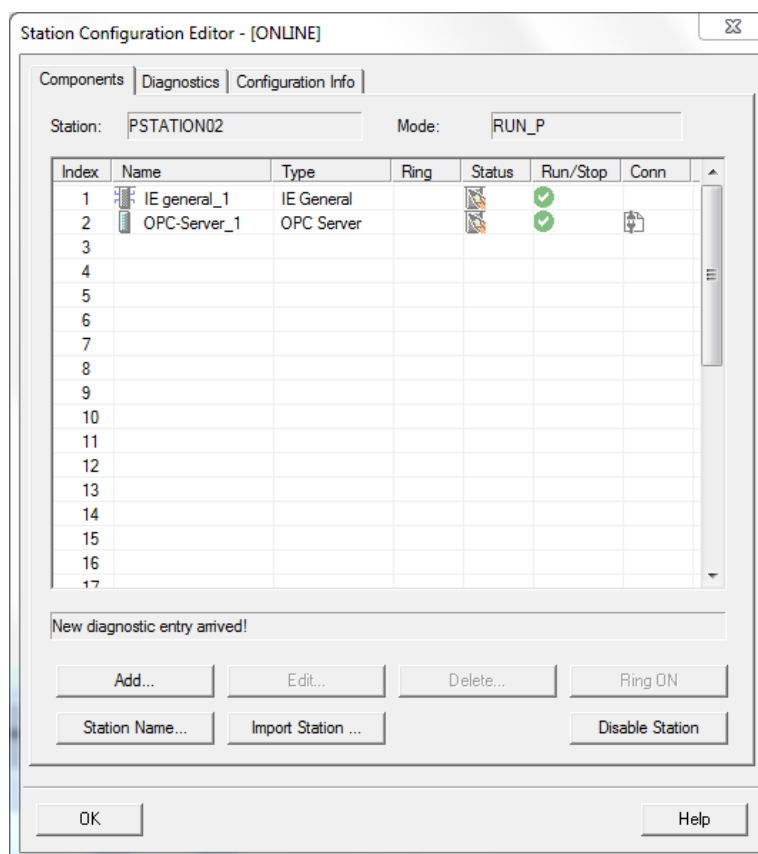
PC Station Configuration Editor je uživatelské konfigurační rozhraní určené pro nastavení naší PC stanice, která obsahuje důležité komunikační komponenty. PC stanice je brána jako další SIMATIC S7 zařízení. Proto je důležité zajistit vytvoření jeho konfigurace a nastavení v softwaru, který využíváme pro vytváření řídicích programů pro S7 PLC systémy. Toto nastavení, stejně jako u PLC systémů, je důležité vykompilovat a nahrát do zařízení.

V editoru nalezneme pojmenování stanice. Toto jméno musíme zadat i jako pojmenování naší PC Station v TIA Portal, ve kterém vytváříme řídicí programy pro PLC a konfiguraci PC Station. Bez tohoto kroku nebude možné následně naši vykompilovanou konfiguraci v programu TIA Portal nahrát do PC Station.

Kromě pojmenování a nastavení stavu běhu stanice, zde máme možnost i přidávat moduly reprezentující softwarové a hardwarové vybavení naší stanice. Opět musíme zajistit, aby stanice obsahovala moduly, které používáme a konfiguruje v programu TIA Portal. To zajistíme přidáním modulů:

- *Modul IE General* reprezentující naši síťovou kartu v PC stanici
- *Modul OPC-Server* reprezentující námi zvolený typ komunikace

Nastavení těchto komponent je zařízeno nahráním konfigurace z TIA Portal. V momentu, kdy máme všechny tyto kroky splněné (Obr. 20), tak můžeme stanici přepnout do módu běhu. Od tohoto momentu máme k dispozici funkční OPC komunikaci mezi linkami a OPC serverem umístěným na naší PC stanici.



Obr. 20 Nastavený PC Station Configuration Editor

4.2.3 Přístup k hodnotám OPC serveru

Námi vytvořený a nastavený Simatic.Net OPC server nám umožní přístup ke všem datům našich nakonfigurovaných PLC systémů:

- Všechny digitální vstupy (např. I124.5)
- Všechny digitální výstupy (např. Q125.2)
- Všechny hodnoty uložené v paměti (např. M1.0)
- Všechny hodnoty atributů v datovém bloku (např. DB1.DB4)

Všechny tyto hodnoty jsou na Simatic.Net OPC serveru uloženy ve složce „\S7:“. Z toho důvodu je nutné u klientů, které se k našemu OPC serveru připojují, vybrat jen ty hodnoty atributů, ke kterým potřebujeme mít přístup. Aby klient mohl číst tyto hodnoty, je potřeba vytvořit řetězec (Tab. 5):

- Obsahující složku, ze které chceme číst
- Jméno připojení, které nám říká z jakého PLC systému chceme brát hodnoty
- Jaký typ proměnné chceme číst
- A symbolický bit proměnné uložené v paměti

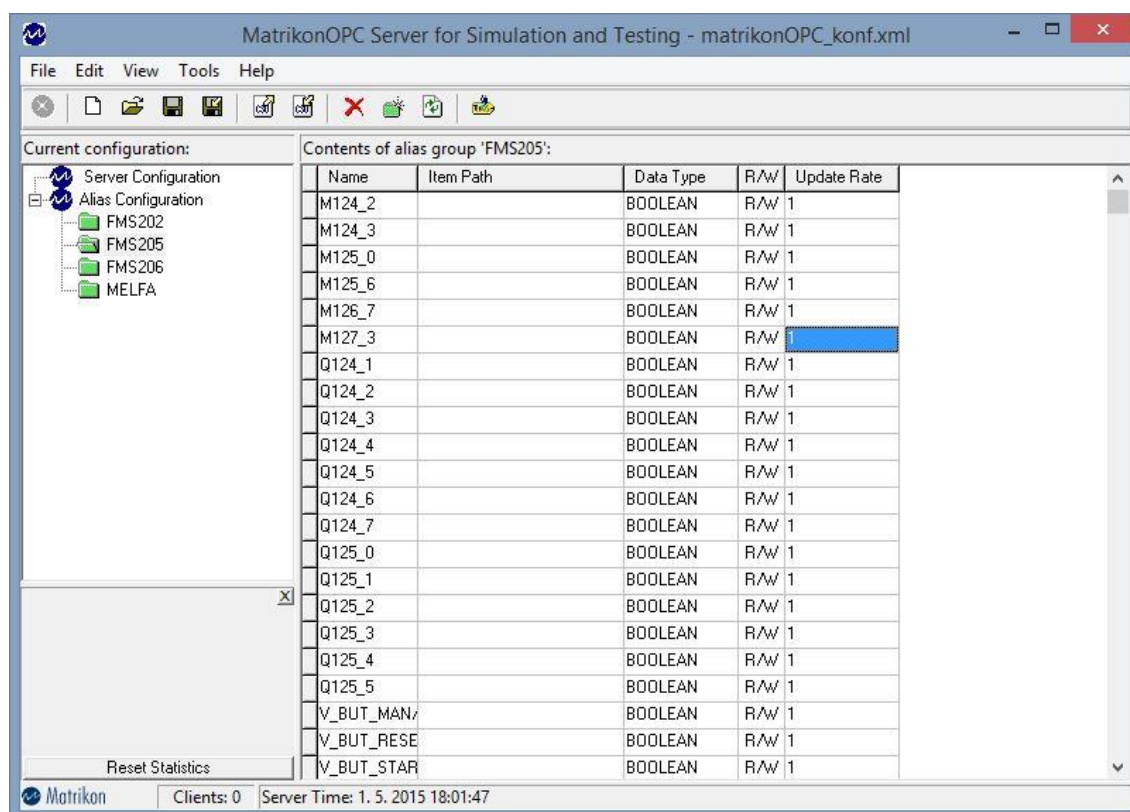
Tab. 5 Ukázka řetězců pro přístup k hodnotám OPC serveru

Řetězec	Popis
S7:[S7_Connection_1]IX124.5	Přístup k hodnotě digitálního vstupu v paměti s hodnotou bytu 124 a bitu 5 modulu linky FMS-206
S7:[S7_Connection_2]QX125.0	Přístup k hodnotě digitálního výstupu v paměti s hodnotou bytu 125 a bitu 0 modulu linky FMS-205
S7:[S7_Connection_3]MX1.0	Přístup k hodnotě v paměti s hodnotou bytu 1 a bitu 0 modulu linky FMS-202
S7:[S7_Connection_1]DB1, X0.5	Přístup k hodnotě atributu uloženém v datovém bloku s hodnotou bytu 0 a bitu 5 modulu linky FMS-206

4.3 MatrikonOPC Simulation Server

Při vytváření OPC komunikace je vždy rozumné zajistit odzkoušení námi navržené komunikace. Díky tomuto kroku se vyhneme možným chybám a kritickým situacím, které by mohly zavinit chybu nebo poškození námi řízeného procesu nebo přístroje. Navíc můžeme provádět změny nebo vytvářet nové komunikační postupy bez nutnosti zastavit již běžící proces nebo stroj.

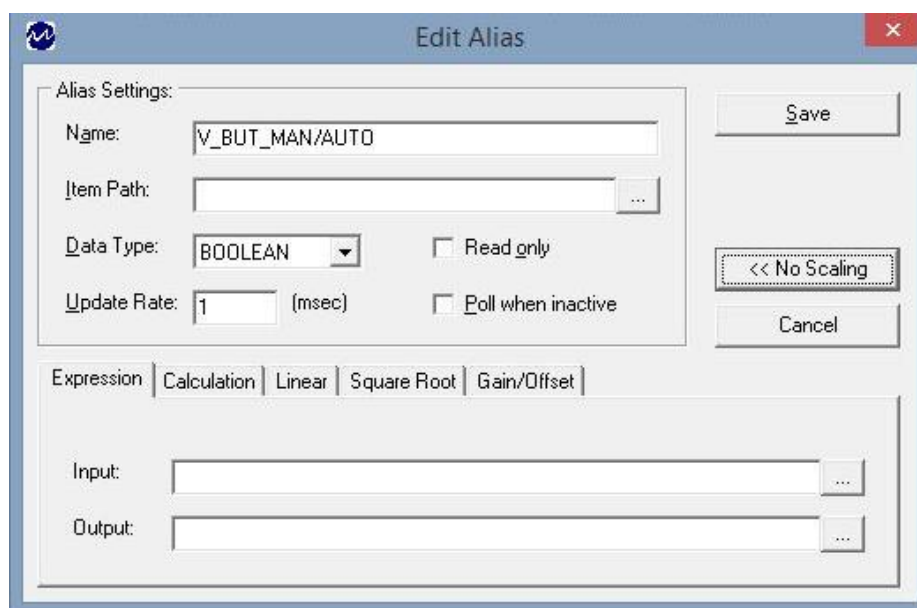
MatrikonOPC Simulation Server je volně dostupný OPC server, který neobsahuje žádné ovladače pro připojení ke konkrétním přístrojům. Jeho cílem je jen simulace běhu virtuálního OPC serveru. Díky tomu si můžeme odzkoušet připojení klientů k serveru, simulovat si běh našeho průmyslového procesu nebo stroje tak, abychom zajistily bezchybné nasazení do reálného provozu.



Obr. 21 MatrikonOPC Simulation Server

Server nám umožňuje simulovat náhodné hodnoty nebo konkrétní hodnoty atributů, které budeme potřebovat otestovat pro náš provoz. Tyto hodnoty si můžeme pomocí aliasů pojmenovat tak, abychom měli lepší přehled a odpovídaly našim hodnotám, které budeme v reálném prostředí využívat. Aliasy je možné dále kvůli přehlednosti organizovat do skupin. U jednotlivých aliasů můžeme nastavovat (Obr. 22):

- Jméno aliasu
- Cestu k objektu – v tomto případě se jedná o odkaz na typ proměnné uložené na OPC serveru, kterou chceme simulovat
- Typ proměnné (např. integer, boolean, string, atd.)
- Zda má být proměnná pouze pro čtení
- Interval aktualizace proměnné
- Hodnoty pro škálování, pokud chceme vytvářet složitější simulované hodnoty



Obr. 22 Vytváření aliasu na MatrikonOPC Simulation Serveru

Další výhodou tohoto simulačního serveru je naprostá absence nutnosti provádět jakoukoliv konfiguraci samotného serveru. Stačí pouze provést instalaci serveru a můžeme začít simulovat a testovat náš proces či přístroj.

4.3.1 MatrikonOPC Explorer

Firma Matrikon nabízí ke svému simulačnímu serveru i program Explorer. Tento program je určený k prohlížení hodnot OPC serverů. Kromě toho ale zároveň umožňuje přidávat hodnoty do OPC serverů. Výhodou přidávání hodnot přes program Explorer je možnost využít přidání atributu s hodnotou generovanou pomocí generátoru, u kterého můžeme nastavovat:

- Rozsah hodnot proměnných
- Hodnotu inkrementu proměnné
- Počet opakování
- Případně text, který má proměnná obsahovat

4.4 Alternativní OPC servery

Jak již bylo zmíněno, pro komunikaci mezi Siemens moduly nám stačí jakýkoliv OPC server, který obsahuje ovladače pro komunikaci se SIMATIC S7 PLC moduly. Na trhu jsou mnohé OPC servery (Tab. 6), které mají podporu pro S7 systémy.

Tab. 6 Ukázka výrobců OPC serverů pro komunikaci se Siemens Simatic S7

OPC server	Výrobce	Podporované PLC
Deltalogic s7/s5 OPC server	Deltalogic	Simatic S7-200, S7-300/400, S7-1200
KEPServerEX OPC	Kepware	Simatic S7-200, S7-300/400, S7-1200
S7/S5 OPC Server	Softing	Simatic S7-200, S7-300/400, S7-1200, C7 a S5
Matrikon S7 PLC OPC Server	Matrikon Inc.	Simatic S7-200, S7-300/400, S7-1200
VIPA OPC Server	VIPA	Simatic S7-300/400, S7-1200
S7/S5 OPC Server	Softing	Simatic S7-200, S7-300/400, S7-1200
OPC Web Studio	Resolvica	Simatic S7

4.5 Monitorované hodnoty

U modulů potřebujeme pro vizualizaci zajistit přístup k atributům, které jsou nutné pro ovládání a monitorování. U těchto atributů potřebujeme mít možnost provádět jak čtení, tak i zápis.

4.5.1 Hodnoty pro modul FMS-202

Pro vizualizaci a řízení modulu FMS-202 potřebujeme následující atributy (Tab. 7):

Tab. 7 OPC atributy pro modul FMS-202

Atribut	Přístup	Popis
I124.0	Read/Write	Tlačítko Start
I124.3	Read/Write	Tlačítko Reset
I124.2	Read only	Přepínač Auto/Man
Q124.1	Read only	Vyjetí podavače A
Q124.2	Read only	Pohyb podavače B dopředu
Q124.3	Read only	Pohyb podavače B dozadu
Q124.4	Read only	Rozevření uchopovače C
Q124.5	Read only	Pohyb plošiny D nahoru
Q124.6	Read only	Pohyb plošiny D dolů
Q124.7	Read only	Vyjetí ejektoru E
Q125.0	Read only	Vyjetí tělesa F
Q125.1	Read only	Pohyb vkladače G
Q125.2	Read only	Pohyb vkladače H dopředu
Q125.3	Read only	Rozevření uchopovače I
M124.5	Read only	Signalizace prázdného zásobníku
M124.7	Read only	Kontrolní bod I.
M125.4	Read only	Kontrolní bod II.
M125.6	Read only	Kontrolní bod III.
M126.1	Read only	Kontrolní bod IV.
M126.3	Read only	Kontrolní bod provedení měření
M126.6	Read only	Velké ložisko
M126.7	Read only	Malé ložisko
M127.5	Read only	Kontrolní bod V.
M128.1	Read only	Vyloučení ložiska z linky
M128.5	Read only	Kontrolní bod VI.
M128.7	Read only	Kontrolní bod VII.
M129.2	Read only	Vložení ložiska do tělesa

4.5.2 Hodnoty pro modul FMS-205

Pro vizualizaci a řízení modulu FMS-205 potřebujeme následující atributy (Tab. 8):

Tab. 8 OPC atributy pro modul FMS-205

Atribut	Přístup	Popis
I124.2	Read/Write	Tlačítko Start
I124.5	Read/Write	Tlačítko Reset
I124.4	Read/Write	Přepínač Auto/Man
Q124.1	Read only	Sjetí podavače A dolů
Q124.2	Read only	Pohyb přenašeče B dopředu
Q124.3	Read only	Sevření uchopovače C
Q124.4	Read only	Otáčení stolu D
Q124.5	Read only	Vysunutí cylindru E
Q124.6	Read only	Vyjetí ejektoru F
Q124.7	Read only	Zajetí ejektoru F
Q125.0	Read only	Pohyb vyhazovače G dolů
Q125.1	Read only	Sepnutí vakuové přísavky V
Q125.2	Read only	Pohyb vkladače H dolů
Q125.3	Read only	Pohyb vkladače I dopředu
Q125.4	Read only	Sevření uchopovače J
Q125.5	Read only	Vyjetí podavače krytů K
M124.2	Read only	Signalizace prázdného zásobníku
M124.3	Read only	Kontrolní bod I.
M125.0	Read only	Kontrolní bod II.
M125.6	Read only	Kontrolní bod III.
M126.7	Read only	Signalizace vyloučení krytu z linky
M127.3	Read only	Signalizace schválení krytu

4.5.3 Hodnoty pro modul FMS-206

Pro vizualizaci a řízení modulu FMS-206 potřebujeme následující atributy (Tab. 9):

Tab. 9 OPC atributy pro modul FMS-206

Atribut	Přístup	Popis
I124.0	Read/Write	Virtuální tlačítko Start
I124.3	Read/Write	Virtuální tlačítko Reset
Q124.1	Read only	Pohyb přenašeče A dopředu
Q124.2	Read only	Pohyb přenašeče A dozadu
Q124.3	Read only	Pohyb přenašeče B dolů
Q124.4	Read only	Sevření uchopovače C
Q124.5	Read only	Pohyb podavače D dopředu
Q124.6	Read only	Pohyb přenašeče A dopředu
Q124.7	Read only	Pohyb přenašeče E dozadu
M124.1	Read only	Kontrolní bod I.
M124.3	Read only	Signalizace prázdného zásobníku
M124.5	Read only	Kontrolní bod II.
M125.3	Read only	Kontrolní bod III.
M126.4	Read only	Kontrolní bod IV.
M126.6	Read only	Signalizace dokončení celé operace

4.5.4 Hodnoty pro robota MELFA

Pro monitorování robota MELFA nám postačí pouze znát aktuální souřadnice koncového efektoru robota. Pro samotné ovládání bude potřebovat předat robotovi souřadnice, na které se má koncový efektor přemístit. Společně s těmito hodnotami budeme potřebovat ještě jeden atribut, který bude signalizovat robotovi, zdali má sevřít nebo rozevřít svůj koncový efektor (Tab. 10).

Tab. 10 OPC atributy pro robota MELFA

Atribut	Přístup	Popis
GET_X	Read/Write	Aktuální pozice koncového efektoru v ose x
GET_Y	Read/Write	Aktuální pozice koncového efektoru v ose y
GET_Z	Read/Write	Aktuální pozice koncového efektoru v ose z
GET_A	Read/Write	Úhel natočení koncového efektoru v ose x
GET_B	Read/Write	Úhel natočení koncového efektoru v ose y
SET_X	Read/Write	Pozice na ose x, kam se má koncový efektor přemístit
SET_Y	Read/Write	Pozice na ose y, kam se má koncový efektor přemístit
SET_Z	Read/Write	Pozice na ose z, kam se má koncový efektor přemístit
SET_A	Read/Write	Úhel k ose x, do kterého se má koncový efektor natočit
SET_B	Read/Write	Úhel k ose y, do kterého se má koncový efektor natočit
HAND_OPEN	Read/Write	Signalizace rozevření koncového efektoru

4.6 Připojení klientů k OPC serveru

4.6.1 Připojení Control Webu k OPC serveru

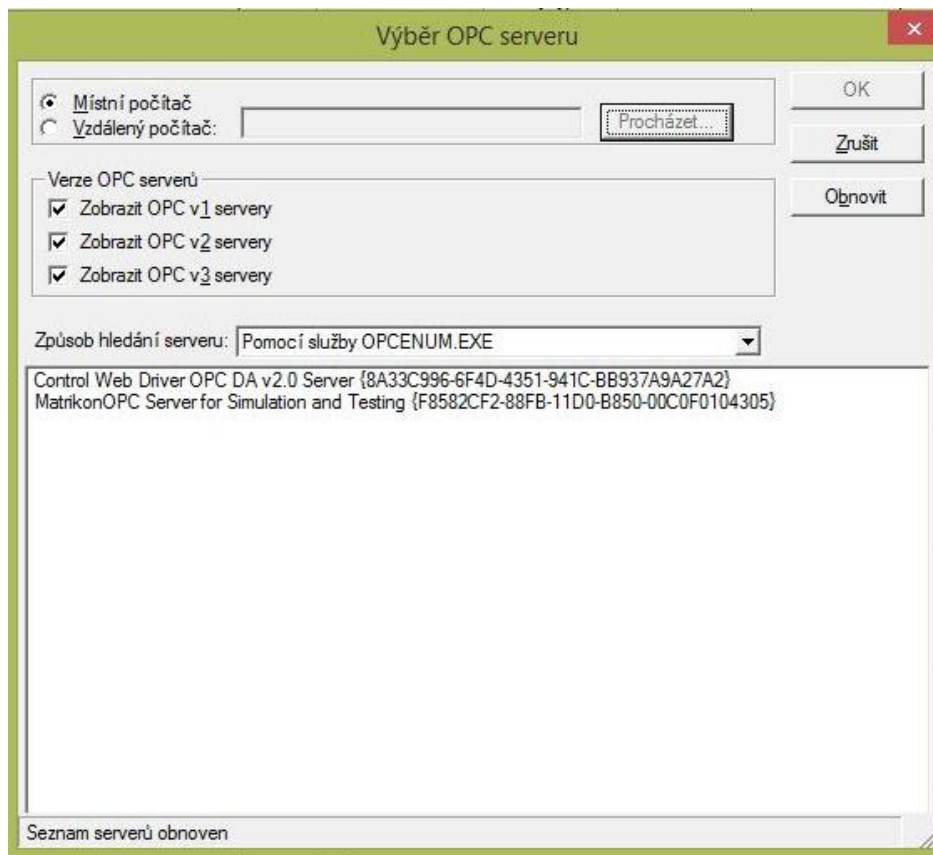
Control Web umožňuje komunikaci s jakýmkoliv OPC serverem. K tomu je potřeba, do již nainstalované verze programu Control Web, doinstalovat OPC ovladač, který nám umožní komunikovat s OPC servery třetích stran. Tento ovladač podporuje komunikaci se všemi verzemi standardů OPC serverů. V současné době jsou k dispozici tři verze standardů.

Před tím než budeme moci přímo pracovat s našimi OPC servery ve vývojovém prostředí Control Web musíme si prvně vytvořit dva konfigurační soubory odpovídající pravidlům konfiguračních *.INI* souborů:

- **Parametrický soubor (.PAR)** – tento soubor obsahuje sekci *server*, která obsahuje identifikátor serveru a případně vzdáleného serveru i jeho UNC jméno. Druhá sekce *channels* obsahuje definice kanálů, které obsahují jednotlivé atributy OPC serveru.
- **Mapovací soubor (.DMF)** – tento soubor obsahuje typ proměnné a komunikace pro jednotlivý kanál v parametrickém souboru.

Vytvoření takovýchto konfiguračních souborů je poměrně složitá a náročná operace, proto je společně s instalací programu Control Web dodáván i konfigurační nástroj OPC ovladače. Pomocí tohoto nástroje lze jednoduše v několika následujících krocích vytvořit oba konfigurační soubory, které využijeme při následné vizualizaci.

První krok obsahuje výběr OPC serveru. Při výběru serveru můžeme zvolit, zdali se chceme připojit ke vzdálenému či lokálnímu serveru. Výběr serverů si můžeme případně vyfiltrovat podle verze (Obr. 23).



Obr. 23 Výběr OPC serveru v konfiguračním nástroji OPC ovladače

Potvrzením výběru našeho OPC serveru se dostaneme do dalšího kroku. V tomto kroku si v prohlížeči můžeme rozkliknout jmenný prostor serveru, obsahující aliasy, které jsou dostupné na zvoleném OPC serveru. Atributy, se kterými chceme pracovat, musíme dvojklikem přenést do tabulky kanálů (Obr. 24). Poté, co tabulka obsahuje všechny vybrané atributy, musíme každému kanálu přiřadit číslo. Musíme zajistit, aby každý kanál měl své jedinečné číslo. To můžeme udělat ručně anebo můžeme využít systém automatického přiřazování čísel kanálům.

Poslední krokem je vygenerování konfiguračního souborů. To je zajištěno pomocí nástroje *Zapsat PAR a DMF soubory*. Tento nástroj bude fungovat pouze v případě, že všechny kanály mají přiřazené své unikátní číslo.

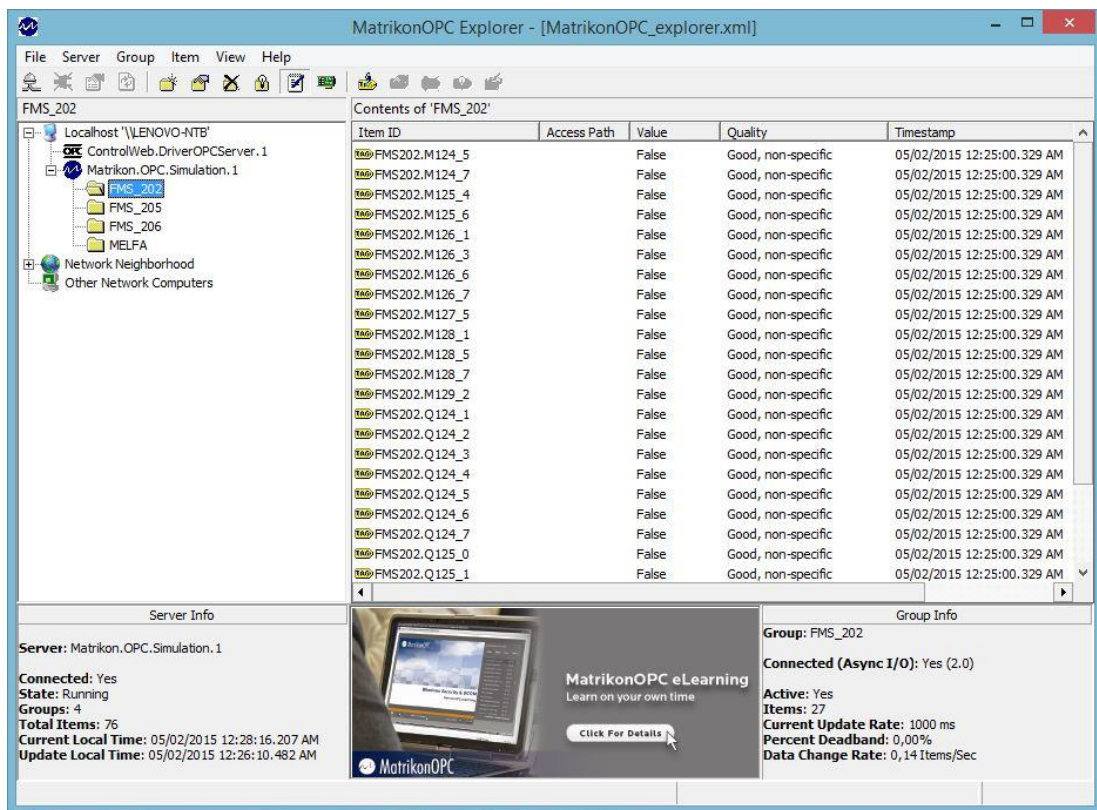
	Číslo kanálu	Datový typ	Směr	OPC identifikátor (DataID)
1	1	boolean	bidirectional	FMS202.M124_5
2	2	boolean	bidirectional	FMS202.M124_7
3	3	boolean	bidirectional	FMS202.M125_4
4	4	boolean	bidirectional	FMS202.M125_6
5	5	boolean	bidirectional	FMS202.M126_1
6	6	boolean	bidirectional	FMS202.M126_3
7	7	boolean	bidirectional	FMS202.M126_6
8	8	boolean	bidirectional	FMS202.M126_7
9	9	boolean	bidirectional	FMS202.M127_5
10	10	boolean	bidirectional	FMS202.M128_1
11	11	boolean	bidirectional	FMS202.M128_5
12	12	boolean	bidirectional	FMS202.M128_7
13	13	boolean	bidirectional	FMS202.M129_2
14	14	boolean	bidirectional	FMS202.Q124_1
15	15	boolean	bidirectional	FMS202.Q124_2
16	16	boolean	bidirectional	FMS202.Q124_3
17	17	boolean	bidirectional	FMS202.Q124_4
18	18	boolean	bidirectional	FMS202.Q124_5
19	19	boolean	bidirectional	FMS202.Q124_6
20	20	boolean	bidirectional	FMS202.Q124_7
21	21	boolean	bidirectional	FMS202.Q125_0
22	22	boolean	bidirectional	FMS202.Q125_1
23	23	boolean	bidirectional	FMS202.Q125_2
24	24	boolean	bidirectional	FMS202.Q125_3
25	25	boolean	bidirectional	FMS202.V_BUT_MAN/AUTO
26	26	boolean	bidirectional	FMS202.V_BUT_RESET
27	27	boolean	bidirectional	FMS202.V_BUT_START
28	28	boolean	bidirectional	FMS205.M124_2

Obr. 24 Tabulka kanálů v konfiguračním nástroji OPC ovladače

4.6.2 Připojení MatrikonOPC Exploreru k OPC severu

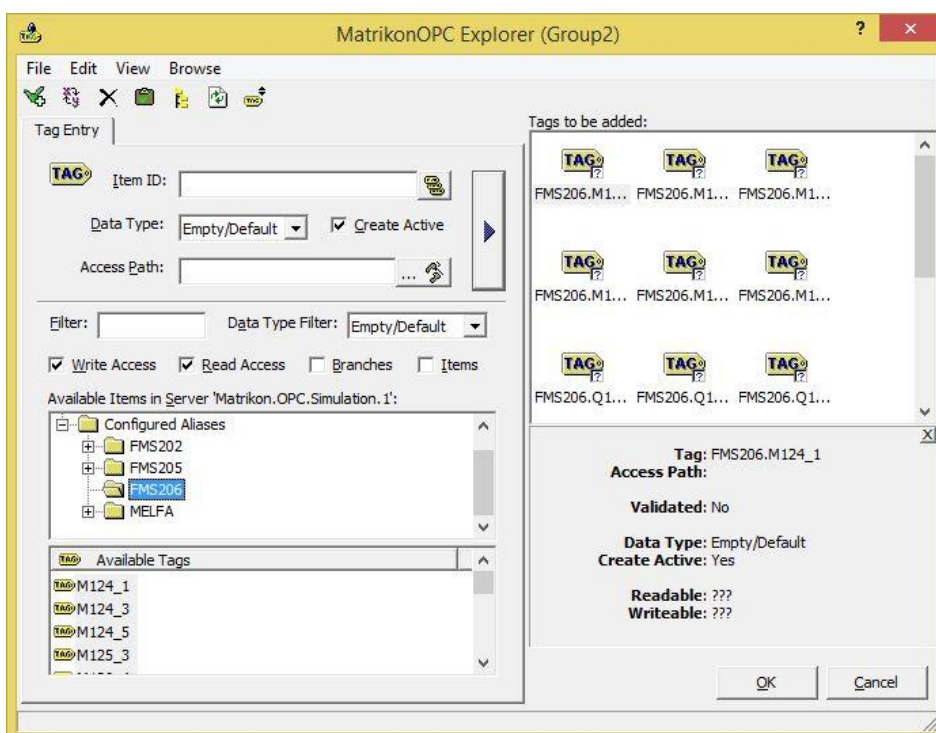
MatrikonOPC Explorer je volně dostupný klient, který obsahuje funkcionalitu pro připojení a testování OPC serverů. Pomocí tohoto Exploreru můžeme prohlížet a měnit hodnoty atributů uložených na OPC serverech.

Explorer je pro připojení k OPC serveru vybaven prohlížečem umístěným v levé části okna aplikace. Zde jsou zobrazeny všechny lokální a vzdálené OPC servery, které se Exploreru podařilo nalézt (Obr. 25).



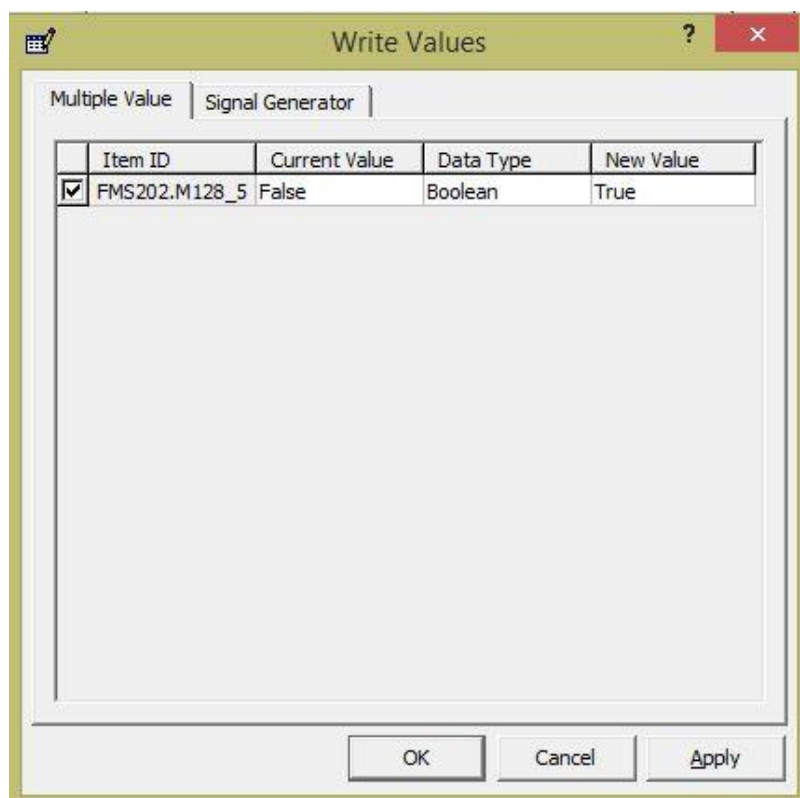
Obr. 25 MatrikonOPC Explorer

Po výběru OPC serveru si musíme do Exploreru přidat atributy, ke kterým chceme přistupovat. To provedeme pomocí tlačítka Add Tags. Poté se nám zobrazí okno, které nám umožní vytvořit skupinu aliasů. Do této skupiny můžeme přidat aliasy, které již vybraný OPC server obsahuje. Nebo můžeme vytvořit nové aliasy, které chceme uložit na vybraný OPC server (Obr. 22).



Obr. 26 Přidání aliasů do MatrikonOPC Exploreru

Poté, co máme do Exploreru přidané aliasy, se kterými chceme pracovat, můžeme monitorovat hodnotu a typ proměnné, kterou obsahují jednotlivé aliasy. Kromě monitorování můžeme tuto hodnotu také měnit (Obr. 27). Je dobré si naši vytvořenou konfiguraci uložit, abychom nemuseli po každém novém spuštění znovu vše nastavovat.

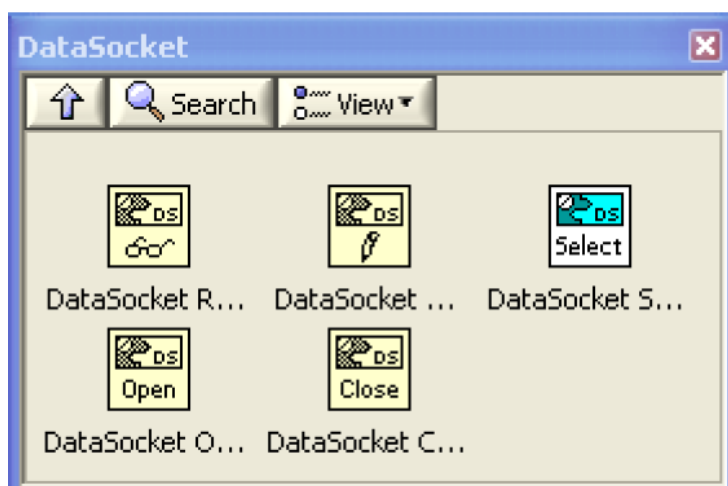


Obr. 27 Změna hodnoty atributu v MatrikonOPC Exploreru

4.6.3 Připojení Labview k OPC serveru

Při analýze možností ovládání robota MELFA bylo zjištěno, že jedinou možností pro vzdálené připojení robota do procesu linky obsahuje ImagingLab Robotics Library, která je dostupná pro vývojové prostředí Labview.

Vývojové prostředí Labview je pro komunikaci s aplikacemi třetích stran vybaveno technologií DataSocket. DataSocket je jednoduché komunikační rozhraní, které umožňuje přijímat a odesílat data různým klientům, bez nutnosti se starat o typy komunikačních protokolů nebo formátu dat. Prvky nutné pro vytvoření komunikace mezi OPC serverem a Labview aplikací najdeme v paletě DataSocket (Obr. 28).



Obr. 28 DataSocket paleta v programu Labview

Paleta obsahuje prvky:

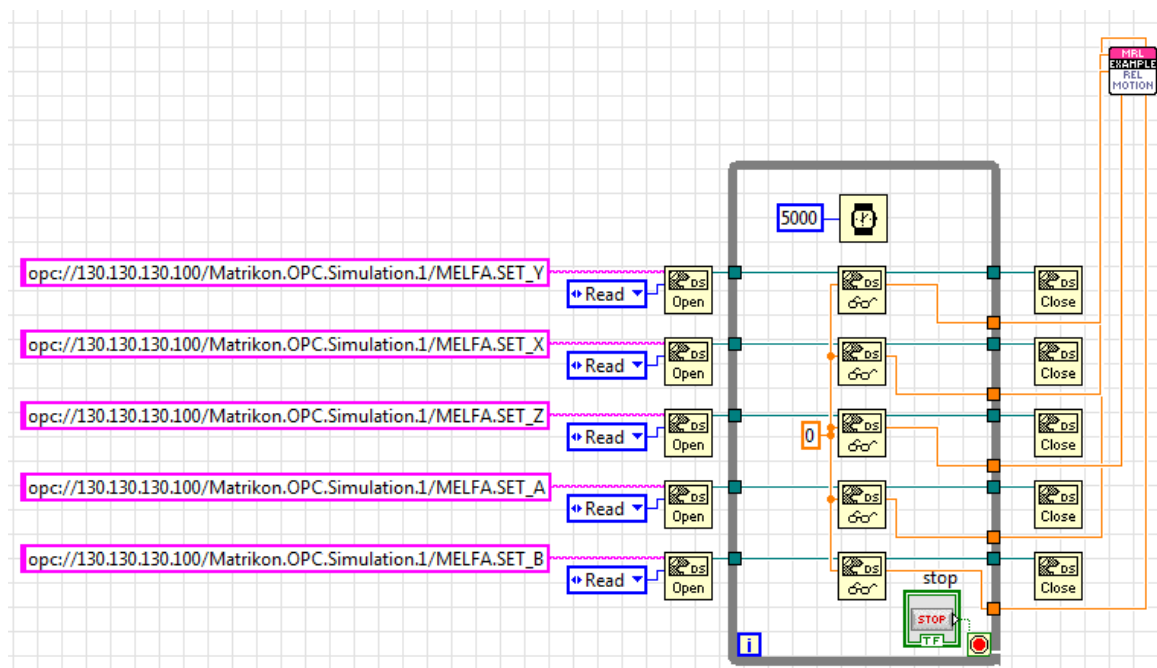
- **DataSocket Select URL** – tento prvek zobrazí prohlížeč zobrazující všechny dostupné lokální nebo vzdálené OPC servery
- **DataSocket Open** – tento prvek zajistí připojení k OPC serveru na základě URL adresy, kterou mu specifikujeme. Ta musí mít formát:

opc://název_stroje/jméno_serveru/alias

- *opc* – nám značí, o jaký typ komunikace se jedná
- *název_stroje* – nám značí jméno stroje, na kterém je náš OPC server umístěn. Ten může být reprezentován i IP adresou stroje.
- *jméno_serveru* – konkrétní pojmenování OPC serveru, ke kterému se chceme připojit
- *alias* – alias nacházející se na OPC serveru, se kterým chceme pracovat
- **DataSocket Read** – přečte obsah proměnné, kterou jsme mu předali v URL adrese
- **DataSocket Write** – zapíše hodnotu, kterou chceme zapsat na OPC server, do proměnné zadané v předané URL adrese
- **DataSocket Close** – uzavře naši komunikaci s OPC serverem

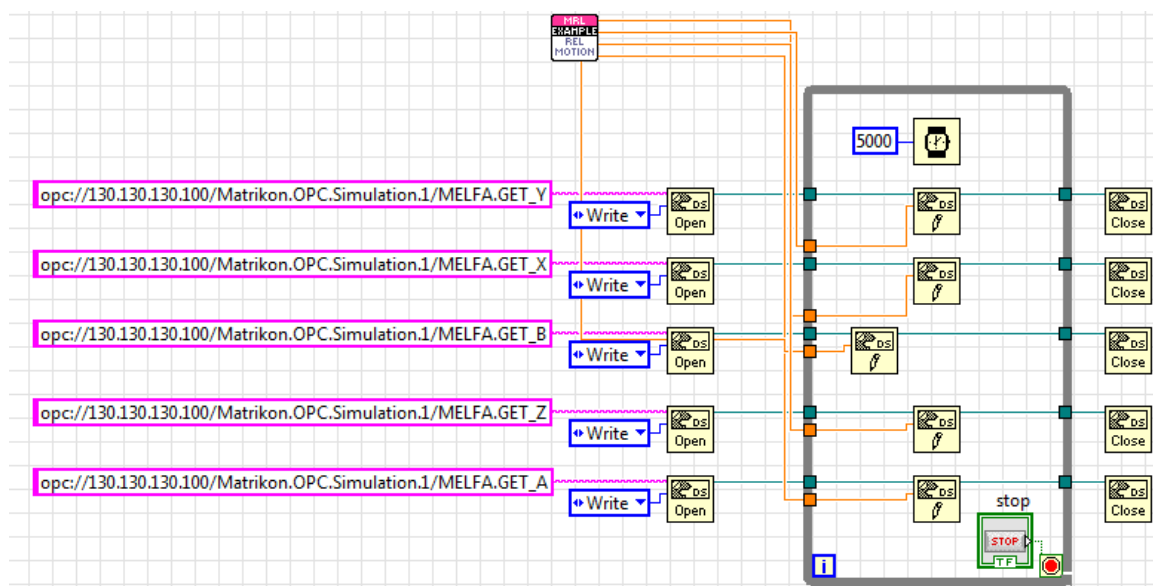
Pro čtení hodnot atributů z OPC serveru pomocí technologie DataSocket je nutné do blokového diagramu poskládat prvky pro otevření, čtení a ukončení spojení. Pokud chceme tyto prvky číst periodicky, je vhodné zajistit jejich umístění uvnitř cyklu, který zajistí periodické čtení hodnot z OPC serveru. Tyto hodnoty posléze uloží nebo odešle jiné procedury (Obr. 29). U prvku pro otevření spojení k OPC serveru musíme zadat vlastnost určující čtení a na vstup zadat URL adresu serveru. Dále je nutné u prvku pro čtení nastavit typ datové proměnné, který chceme

číst. V našem případě byl použit cyklus while, který je prováděn od spuštění periodicky, s časovým intervalem pěti sekund, do doby, dokud není ukončen tlačítkem stop umístěným na čelním panelu.



Obr. 29 Čtení hodnot z OPC serveru a předání hodnot proceduře pro ovládání MELFY

Při zápisu hodnot na OPC server musíme do blokového diagramu opět přidat prvky pro otevření komunikace, zápis hodnot a ukončení komunikace z palety DataSocket (Obr. 30). U prvku pro otevření komunikace s OPC serverem musíme zadat vlastnost určující zápis a také opět specifikovat URL adresu serveru. Prvku pro zápis pouze stačí předat atribut z funkce, která bude získávat data z robota MELFA, která chceme zapsat na opc server. Zde není nutné specifikovat žádné vlastnosti atributu. Opět jsme zde využili možnosti umístění prvku pro zápis do cyklu while s časovým intervalem pěti sekund, který bude běžet do doby, než bude ukončen stiskem tlačítka stop na čelním panelu.



Obr. 30 Zápís hodnot obdržných z MELFY na OPC server

5 Vizualizace komponent v prostředí Control Webu

5.1 Vývojové prostředí Control Web

Control Web je vývojové prostředí určené pro vizualizaci a řízení průmyslových procesů a přístrojů. Jeho nasazení je možné v mnoha oblastech působnosti. Ta zahrnuje malé až velké firmy, vytváření malých aplikací, nasazení pro výzkumné a edukační účely. (Moravské přístroje, 2010)

Control Web může vystupovat v mnoha rolích v rámci průmyslových procesů. Tím je například plnění funkce řídicí jednotky procesu nebo stroje, může zajistit propojení jednotlivých procesů nebo strojů, může modelovat nebo simulovat různé procesy nebo stroje atd. (Moravské přístroje, 2010)

Control Web bychom mohly označit jako komponentový. To vychází z podstaty vytváření aplikací v rámci vývojového prostředí. Aplikace jsou totiž složeny z virtuálních přístrojů, které již vývojové prostředí obsahuje, nebo si je ve formě knihovny můžeme do prostředí doplnit. Každý virtuální přístroj má svoje procedury, reakce na události a časovače, které můžeme v rámci přístroje využít. (Moravské přístroje, 2010)

Control Web je dostupný ve dvou verzích:

- **Vývojová verze** – tato verze je určená pro vývoj a testování aplikací. Aplikace vytvořené v tomto prostředí poté převedeme do formátu pro runtime verze.
- **Runtime verze** – tato verze je určená pro běh aplikací vytvořených ve vývojové verzi. V této verzi se nemůžeme dostat ke zdrojovému kódu aplikace ani provádět žádné změny aplikace. Pouze můžeme měnit stavy prvků, které aplikace obsahuje.

V rámci vývojové verze Control Webu máme pro vytváření aplikací k dispozici tři vývojová prostředí:

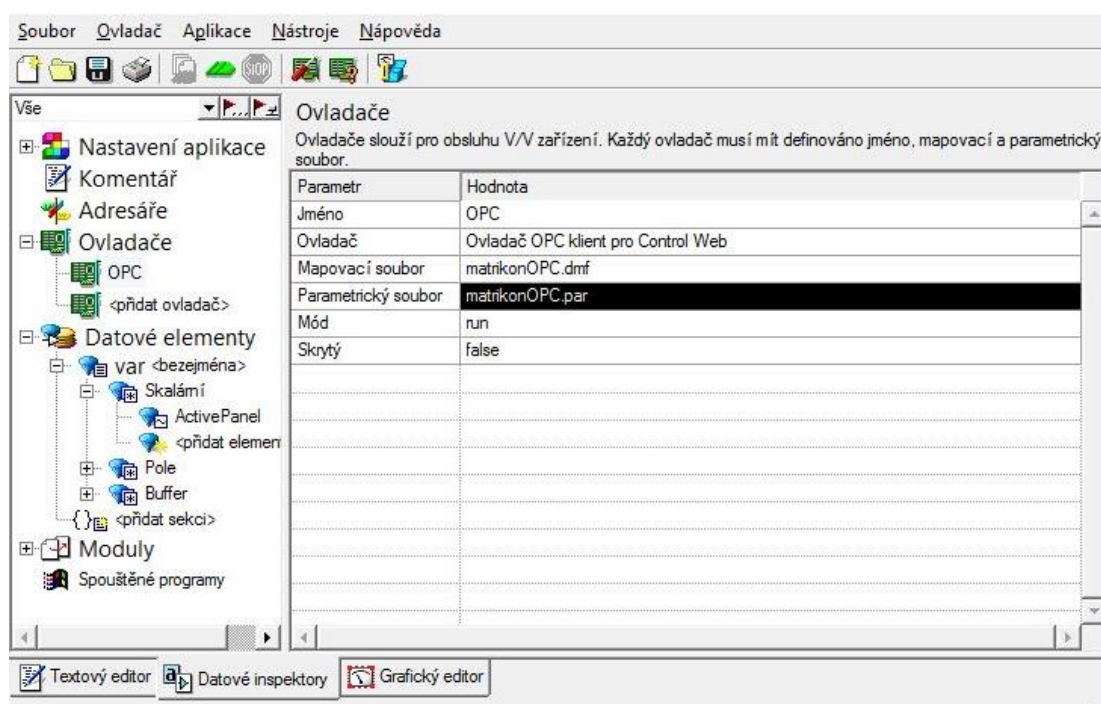
- **Grafický editor** – tento editor slouží pro vytváření vzhledu aplikace a jejího uživatelského rozhraní, pomocí kterého bude aplikace ovládána. Zároveň se zde nachází paleta přístrojů, ve které jsou obsaženy všechny přístroje, které můžeme využít v naší aplikaci. U přidaných aplikací v grafickém editoru můžeme pomocí inspektoru přístroje upravovat a nastavovat hodnoty a procedury přístroje. Zároveň zde můžeme pracovat i s přístroji, které nemají grafické rozhraní.
- **Datové inspektory (editor)** – slouží pro nastavení parametrů a globálních proměnných vytvářené aplikace. Dále se zde definují ovladače potřebné pro připojení k přístrojům.

- **Textový editor** – v tomto editoru je obsažen zdrojový kód celé aplikace. To znamená, že zde uvidíme kód všech prvků, které jsme vytvořili a přidali v grafickém nebo datovém editoru. Pokud nechceme nebo nepotřebujeme využít jiné editory, můžeme aplikaci vytvářet v textovém editoru. (Moravské přístroje, 2010)

5.2 Vizualizace komponent linky FMS-200

V rámci vizualizace procesu jednotlivých modulů musíme jako první krok zajistit inicializaci ovladače pro OPC server. K tomu musíme do vývojového prostředí Control Webu doinstalovat obecný ovladač pro OPC server. Po úspěšné instalaci se v Control Webu přepneme do prostředí datového inspektoru. Zde musíme jako nové zařízení přidat náš OPC server (Obr. 31). Pro OPC server je nutné zadat:

- Název zařízení – důležité pro přístup ke kanálům zařízení.
- Typ ovladače – zde zvolíme typ ovladače pro zařízení, se kterým chceme komunikovat, což v našem případě znamená ovladač pro OPC server.
- Mapovací soubor – tento soubor jsme si pro OPC server vygenerovaly pomocí nástroje konfigurace OPC serveru.
- Parametrický soubor – tento soubor jsme si také pro OPC server vygenerovaly pomocí nástroje konfigurace OPC serveru.
- Mód – nám říká jak v jakém módu má zařízení běžet. Můžeme nastavit buď jako run, check nebo simulate. V prvním případě běží zařízení standardním způsobem, ve druhém se pouze prověřuje stav zařízení a kontrolují se jeho kanály a ve třetím případě probíhá simulovaný běh zařízení, který vyžaduje nastavení kanálů v datovém inspektoru.
- Příznak viditelnosti – výběrem položky nastavíme, zdali má být zařízení viditelné v grafickém editoru nebo ne.



Obr. 31 Ukázka přidání zařízení reprezentující OPC server v prostředí Control Webu

Poté, co máme nakonfigurované naše zařízení reprezentující OPC server, můžeme začít v rámci naší aplikace přistupovat k atributům. Control Web pro přístup k vnějším zařízením k dispozici tzv. kanály. Atributy a typ komunikace jednotlivého kanálu je právě definovaná v mapovacím a parametrickém souboru. Díky této konfiguraci můžeme jednoduše přistupovat k hodnotám atributů OPC serveru pomocí jednoduché definice proměnné:

$$\text{Název_zařízení.číslo_kanálu} \quad (5)$$

Pro zobrazení čtyř modulů bychom mohly udělat čtyři separátní aplikace. To by ovšem v praxi bylo velmi nepraktické a neefektivní. Proto byl pro vytvoření vizualizační aplikace zvolen přístup řešený pomocí panelů, kde každý panel bude určen pro jednotlivý modul. To nám umožní z jedné aplikace ovládat jakýkoliv modul, bez nutnosti mít spuštěno více aplikací, nebo zapínat a vypínat jednotlivé aplikace.

V Control Webu je k dispozici přístroj *panel*. Tento přístroj je určen pro zobrazování či shromažďování ostatních přístrojů do jednoho objektu, který následně můžeme skrývat, minimalizovat nebo zobrazovat. Pomocí tohoto přístroje vytvoříme hlavní panel, do kterého umístíme čtyři stejně velké panely, reprezentující jednotlivé moduly a pátý panel, který umístíme na spod hlavního panelu. Tento poslední panel bude určen pro přepínání jednotlivých panelů. Abychom mohly implementovat funkčnost přepínání panelů, musíme do pátého panelu přidat přístroj *tab_switch*. Tento přístroj je určen pro přepínání panelů. Jako refe-

renci mu zadáme naše čtyři panely, které chceme přepínat. Pomocí přepínače můžeme jednoduše vybrat panel, který chceme zobrazit, a ostatní panely se v tu chvíli stanou neviditelnými. Po nastavení této konfigurace můžeme začít do jednotlivých panelů přidávat přístroje potřebné pro vizualizaci daného modulu. Tyto přístroje budou viditelné jen tehdy, pokud bude i panel, ve kterém jsou vloženy, viditelný.

V panelech budeme používat následující přístroje:

- Přístroj box – jednoduchý prvek pro vytváření grafického prostředí. Tento přístroj reprezentuje obdélník, u kterého lze pomocí parametrů nastavovat barvu, velikost a jeho pozici.
- Přístroj label – přístroj, který reprezentuje text nebo libovolné ikony. Má čistě grafický význam, jeho parametry slouží pouze pro úpravu jeho vzhledu.
- Přístroj image – slouží k zobrazení obrázku na své ploše. Pomocí parametrů mu nastavíme, jaký obrázek má zobrazovat, velikost a vzhled.
- Přístroj indicator – přístroj, který zobrazuje hodnotu logické veličiny nebo zadaného logického výrazu. Pomocí parametrů mu můžeme upravovat vzhled, velikost, popisky atd.
- Přístroj switch – slouží pro přiřazení logické hodnoty určenému datovému elementu. Pomocí parametrů mu můžeme nastavit vzhled, velikost, typ principu logiky přístroje.
- Přístroj button – tento přístroj je určený pro zobrazení textu, ikony nebo obou. Stisknutím tohoto přístroje je proveden obsah procedury onPress.
- Přístroj control – tento přístroj nám umožní nastavit hodnotu datového elementu pomocí různých grafických zobrazení.
- Přístroj meter – slouží pro zobrazení hodnoty datového elementu nebo numerického výrazu. Pomocí parametrů mu můžeme nastavit vzhled a velikost.

U některých přístrojů, nejčastěji u takových, které zobrazují hodnoty, je potřeba nastavit jejich časování. Pomocí časování se aktualizuje hodnota zobrazovaná přístrojem, či se zajistí provádění funkce daného přístroje.

5.2.1 Panel pro modul FMS-202

Pro vizualizaci modulu FMS-202 využijeme přístroje *image*, kterému nastavíme jako obrázek vyfocený modul FMS-202 s naznačenými pozicemi jednotlivých manipulátorů modulu, u kterých pomocí přístroje *indicator* budeme zobrazovat jejich pohyb. Pro indikátory opět musíme nastavit časování jednu setinu sekundy, a jako datové elementy jim přiřadíme kanály *OPC.14* až *OPC.24*.

Dále pomocí indikátoru budeme signalizovat prázdný zásobník nebo zaseknutí podání ložiska ze zásobníku. K tomu opět potřebujeme indikátor s časováním, a jako datový element mu přiřadíme kanál *OPC.1*.

Linka umí rozlišovat mezi velkými a malými ložisky. Abychom věděli, která ložiska linka aktuálně přijímá a následně vkládá do připraveného tělesa, budeme

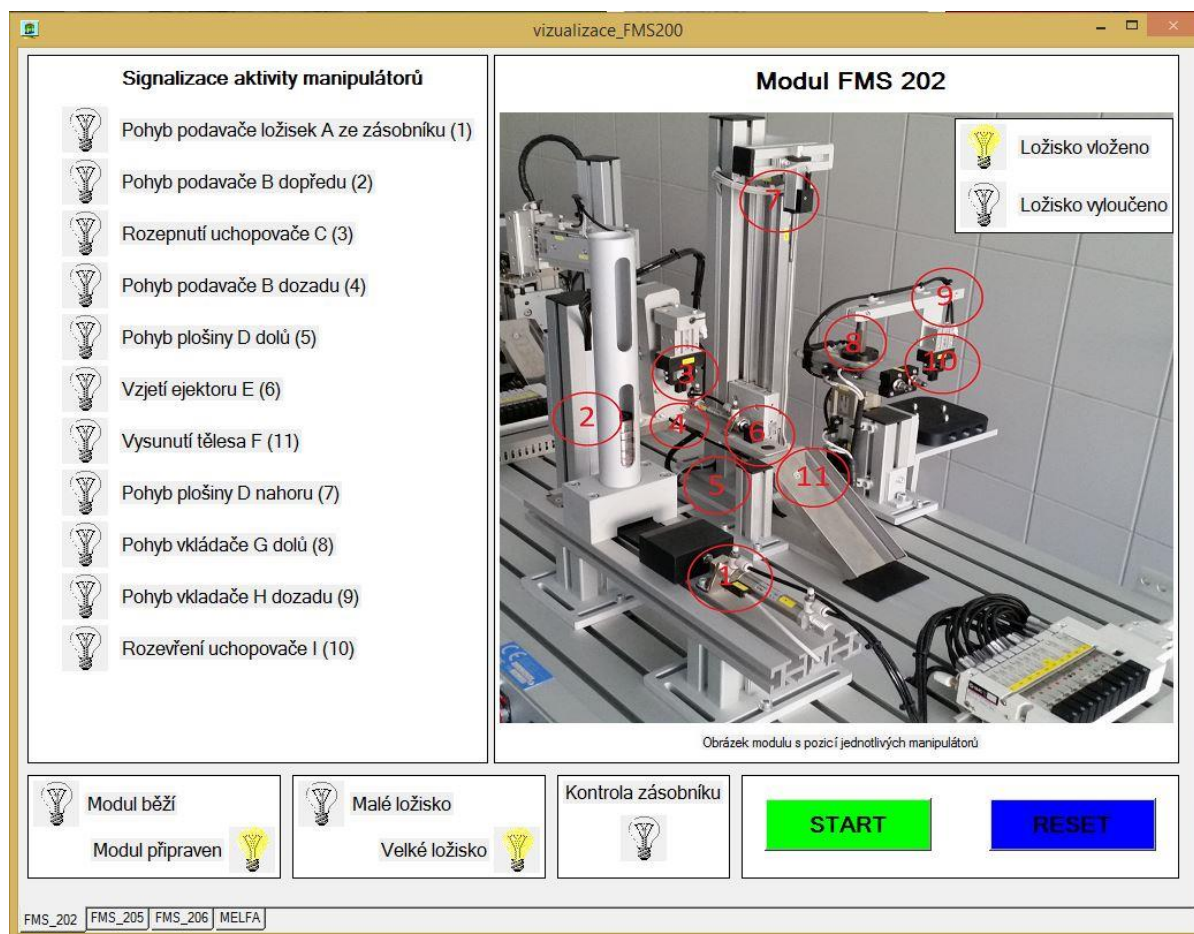
pomocí dvou indikátorů signalizovat aktuální stav. K tomu použijeme dva indikátory, kdy jeden z nich bude signalizovat stav, kdy linka přijímá malá ložiska. K tomu potřebujeme v indikátoru nastavit výraz, který bude vracet hodnotu *true*, když se bude hodnota kanálu *OPC.25* rovna stavu *false*. Druhý indikátor, který bude signalizovat stav, kdy linka přijímá velká ložiska, bude vracet hodnotu *true*, když bude výraz pro hodnotu kanálu *OPC.25* rovna stavu *true*.

Pro modul dále potřebujeme signalizovat stav, kdy linka běží. K tomu opět využijeme indikátor, který bude vyhodnocovat výraz, kdy bude aktivní kontrolní bod I. (kanál *OPC.1*) a zároveň hodnota vyloučení ložiska (kanál *OPC.10*) a vložení ložiska do tělesa (kanál *OPC.13*) neaktivní.

Kromě signalizace běhu modulu potřebujeme signalizovat stav, kdy je modul připraven provádět svou činnost. K tomu opět využijeme indikátor, který bude vyhodnocovat výraz, kdy všechny kontrolní body linky (kanály *OPC.2, 3, 4, 5, 9, 11, 12*) budou neaktivní nebo aktivní.

Jak již bylo zmíněno, tak modul umí rozlišovat ložiska podle velikosti. Proto je nutné u vizualizace zajistit zobrazení, zdali bylo ložisko vloženo nebo vyloučeno. K tomu využijeme dva indikátory, které budou signalizovat hodnotu kanálů *OPC.10* a kanálu *OPC.13*.

Abychom mohli ovládat linku, přidáme na panel dva přístroje switch, které budou reprezentovat tlačítka *START* (kanál *OPC.26*) a *RESET* (kanál *OPC.27*). U obou tlačítek musíme navíc nastavit úroveň aktivního stavu na *true* a čas aktivního stavu na tři desetiny sekundy. To nám zajistí dostatečnou časovou prodlevu, aby mohla být hodnota zpracována OPC serverem a přenesena na PLC modul.



Obr. 32 Ukázka panelu pro vizualizaci a ovládání modulu FMS-202 v prostředí Control Webu

5.2.2 Panel pro modul FMS-205

Vizualizace modulu bude obsahovat stejné přístroje jako vizualizace modulu FMS-202. U indikátorů zobrazující pohyb manipulátorů modulu FMS-205 musíme nastavit kanály *OPC.34* až *OPC.46*.

Pro signalizaci běhu musíme příslušnému indikátoru vložit výraz, kde bude aktivní kontrolní bod I. (kanál *OPC.29*) a bod pro vyloučení (kanál *OPC.32*) a vložení (kanál *OPC.33*) bude neaktivní.

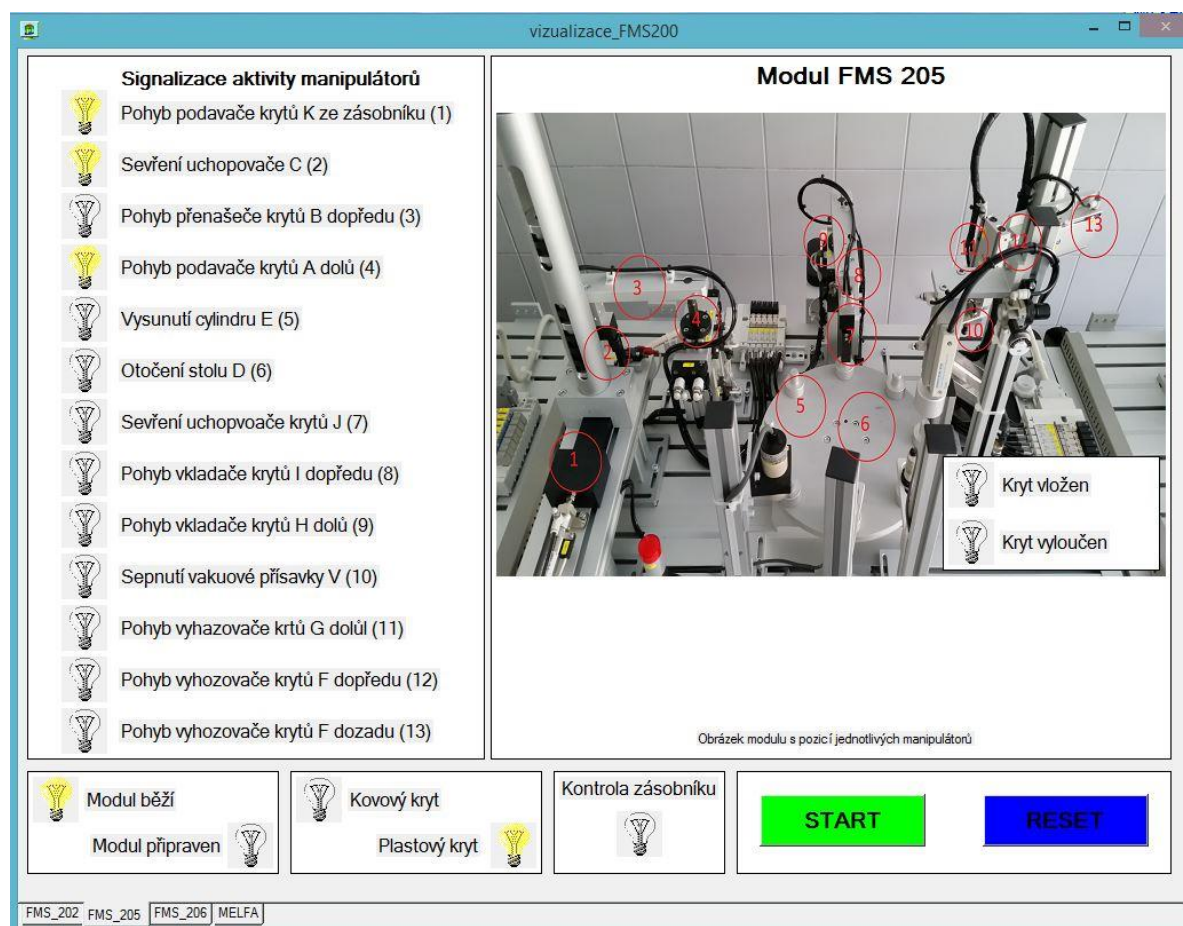
Indikátor signalizující prázdný nebo zaseknutý zásobník bude vyhodnocovat kanál *OPC.35*.

Pro signalizaci připravenosti modulu využijeme indikátor, který bude vyhodnocovat výraz kdy všechny kontrolní body (kanály *OPC.29*, *30*, *31*) a bod pro vložení (kanál *OPC.32*) a vyloučení krytu (kanál *OPC.33*) budou zároveň neaktivní, nebo jeden z nich bude aktivní.

Signalizace nastavení linky bude opět vyhodnocovat kanál *OPC.47*. Když hodnota kanálu bude *false*, rozsvítí se indikátor signalizující nastavení pro příjem kovových krytů. Když bude hodnota kanálu *true*, rozsvítí se indikátor pro signalizaci příjmu plastových krytů.

Indikátory pro signalizaci vložení nebo vyloučení krytu budou vyhodnocovat kanály *OPC.32* a *OPC.33*.

Pro tlačítka reprezentující tlačítka START (kanál *OPC.49*) a RESET (kanál *OPC.48*) použijeme stejné hodnoty pro časování a aktivaci jako u modulu FMS-202.



Obr. 33 Ukázka panelu pro vizualizaci a ovládání modulu FMS-205 v prostředí Control Webu

5.2.3 Panel pro modul FMS-206

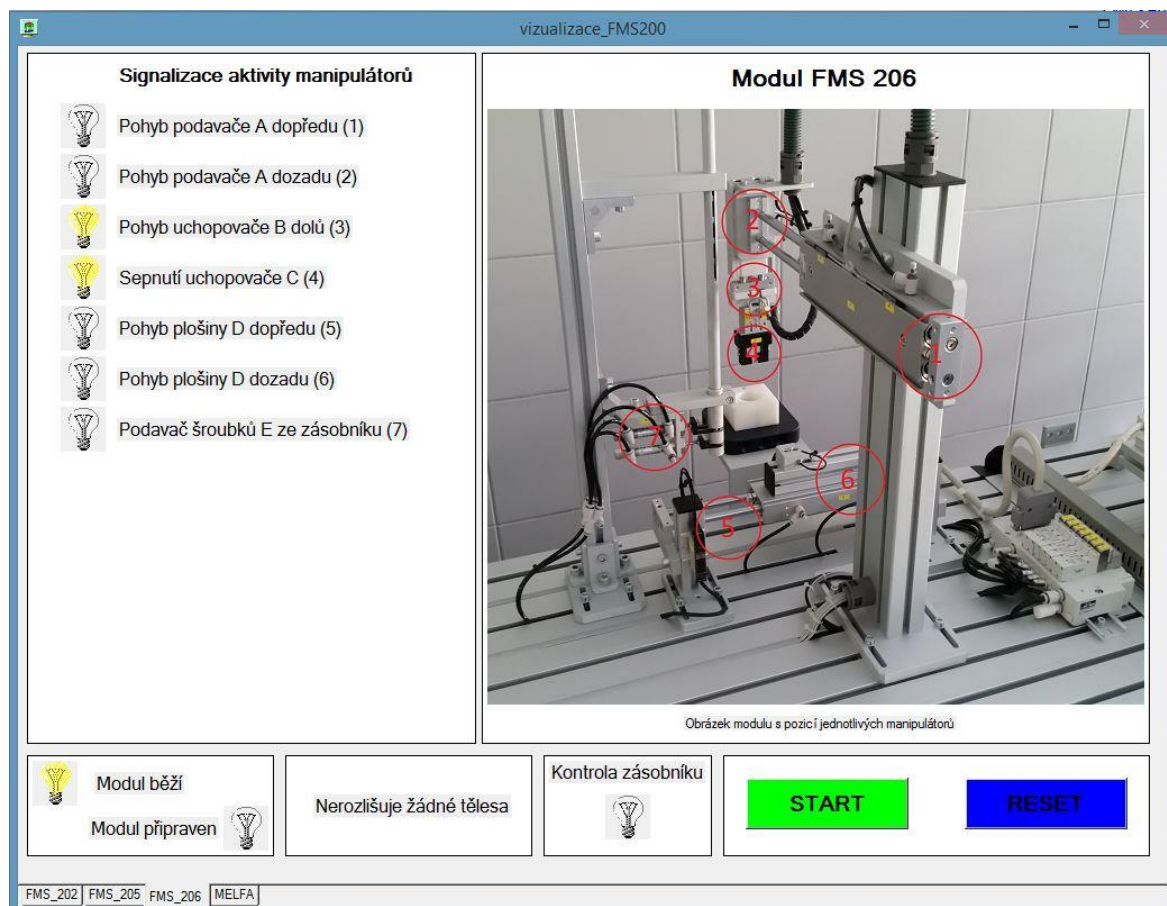
Pro vizualizaci modulu opět přidáme na panel obrázek s naznačenými místy jednotlivých manipulátorů modulu. Pro jednotlivé manipulátory přidáme indikátory, které budou vyhodnocovat datové elementy kanálů *OPC.56* až *OPC.62*.

Modul nerozlišuje žádné vlastnosti šroubků, proto potřebujeme akorát indikátor zobrazující zaseknutý nebo prázdný zásobník. K tomu potřebujeme indikátor vyhodnocující datový element kanálu *OPC.51*.

Pro signalizaci připravenosti linky potřebujeme indikátor, který bude vyhodnocovat výraz, kdy všechny kontrolní body (kanály *OPC.52*, *53*, *54*) a bod signalizující ukončení (kanál *OPC.55*) budou aktivní nebo neaktivní.

Signalizaci běhu linky bude zajišťovat indikátor vyhodnocující výraz, kdy bude aktivní podavač šroubků ze zásobníku (kanál *OPC.62*) a následně bod pro signalizaci ukončení operace (kanál *OPC.55*) bude neaktivní.

Pro ovládání použijeme opět tlačítka *START* (kanál *OPC.64*) a *RESET* (kanál *OPC.65*) se stejnými nastaveními jako u předešlých modulů.



Obr. 34 Ukázka panelu pro vizualizaci a ovládání modulu FMS-206 v prostředí Control Webu

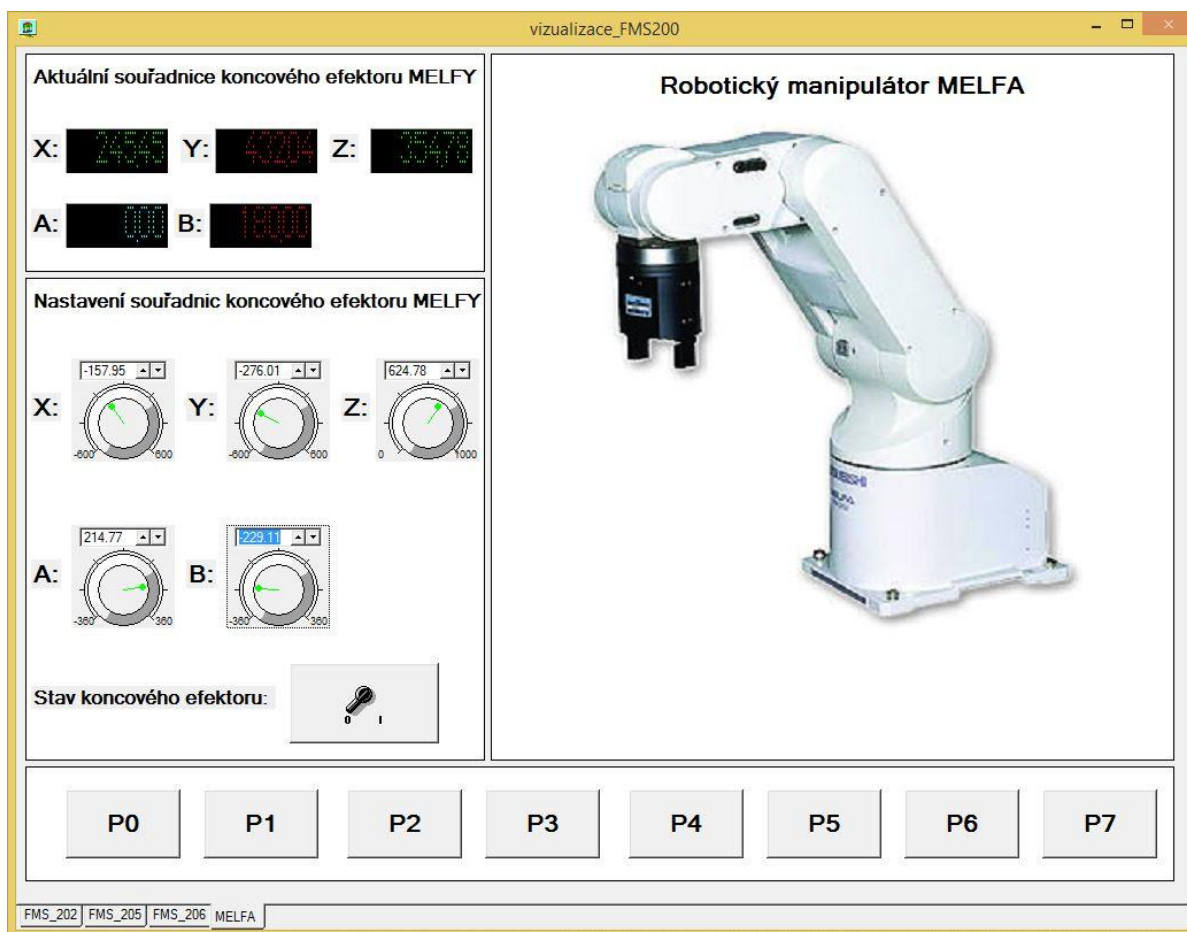
5.2.4 Panel pro robota MELFA

Tento panel pro vizualizaci robota MELFY má za úkol zobrazovat aktuální pozici koncového efektoru robota. K zobrazení aktuálních souřadnic efektoru využijeme přístroje *meter*. Celkem panel obsahuje pět takových přístrojů, aby bylo možné zobrazit hodnoty souřadnic x , y a z a také úhel natočení koncového efektoru A kolem osy x a úhel B kolem osy y . Hodnoty pro tyto přístroje získáme z kanálů *OPC.66* až *OPC.70*. Aby bylo možné zobrazovat vždy aktuální hodnotu těch kanálů, musíme také přístrojům nastavit hodnotu časování na jednu setinu sekundy.

Pro nastavení pozice, kam se má koncový efektor robota přesunout, zajistíme pomocí zadání hodnoty přes přístroj *control*. Vybraly jsme jeho grafické zobrazení *knob*, které nám umožní zadat hodnotu pomocí rotace přepínače nebo přímo zadat hodnotu do textového pole. Zároveň každému přístroji zadáme rozmezí hodnot, které pomocí tohoto přepínače můžeme předat OPC serveru. Pro úhly si vystačíme s rozmezím hodnot od mínus třiašedesáti stupňů do plus třiašedesáti stupňů a pro hodnoty x, y a z můžeme nastavit hodnoty, ve kterých chceme, aby se robot pohyboval. Zde je důležité zohlednit dosah koncového efektoru robota. Hodnoty těchto přístrojů předáme poté kanálům *OPC.72* až *OPC.77*.

Pokud nechceme zadávat hodnoty souřadnic po jedné a ručně, můžeme pomocí přístroje *button* zajistit přesun na předem definovanou pozici. Každému tlačítku reprezentující určitou pozici nastavíme hodnotu souřadnic v lokálních proměnných. A následně v proceduře *onPress* nastavíme přiřazení kanálům *OPC.72* až *OPC.77* příslušné lokální proměnné tlačítka.

Pro ovládání sevření nebo rozevření koncového efektoru využijeme přístroje *switch*. Jako datový element, který bude ovládat, mu přiřadíme kanál *OPC.71*. Zároveň mu nastavíme jako výchozí hodnotu přístroje při spuštění aktuální hodnotu daného kanálu. Přepnutím přepínače do polohy 0 zajistíme sevření efektoru a v poloze 1 dojde k rozevření efektoru.



Obr. 35 Ukázka panelu pro vizualizaci a ovládání robota MELFA v prostředí Control Webu

6 Závěr

Zajistit řízení a monitorování výrobního procesu linky FMS-200, která se skládá z modulů obsahující řídicí systémy od firmy Siemens a robotický manipulátor MELFA od firmy Mitsubishi, se projevilo ze začátku jako velice obtížné. Hlavní příčina obtížnosti spočívala v různosti způsobů řízení těchto systémů. Z toho důvodu bylo nutné zvolit takovou komunikační technologii, která by tuto překážku dokázala odstranit a zajistit efektivní způsob komunikace mezi těmito technologiemi.

Efektivní řešení tohoto problému poskytl technologie OPC. Jak již bylo zmíněno v předešlých kapitolách, pro komunikaci s jakoukoliv technologií potřebuje OPC server pouze vhodný ovladač, který dokáže komunikovat s přístrojem. Zde jsem ovšem narazil na první výrazný problém. Dostupnost OPC serverů je velká. Mnoho výrobců nabízí svá řešení této technologie. Kámen úrazu byl ovšem v dostupnosti ovladačů pro připojení k přístrojům. Výrobci si svoje technologie chrání a přístup k nim umožňují pouze skrz svůj software nebo přes licenční poplatky. To stejné platí i pro ovladače pro připojení k přístrojům pro technologii OPC. Další překážkou byl výběr OPC serveru, který umožnil připojení k S7 PLC modulům pomocí ethernet připojení a zároveň podporoval konkrétní CPU modul nacházející se na jednotlivých modulech linky.

Naštěstí byl k dispozici Simatic.Net OPC server, který je součástí Siemens WinCC implementace ve vývojovém prostředí TIA Portal. Tento OPC server poskytuje rychlou a efektivní komunikaci pro moduly Simatic S7 přístrojů. Jediným negativem je absence možnosti přidat řízení jiných přístrojů od jiných výrobců. Ovšem u OPC technologie není problém přidat a provozovat v jedné síti více OPC serverů. Toho jsem využil a použil MatrikonOPC Simulation Server, který se ukázal jako velmi efektivní nástroj, pomocí kterého můžeme simulovat a ověřit si komunikaci a připojení různých klientů.

Na základě poznatků získaných v této práci o OPC technologii je možné pro příští práce zajistit takový OPC server, který by dokázal spojit komunikaci mezi těmito technologiemi do jednoho serveru a zároveň by umožnil připojení jiných technologií a přístrojů, které by bylo potřeba přidat do linky, aby mohlo být zajištěn efektivní běh procesu a vyššího stupně automatizace.

Před tím, než jsem se mohl začít zabývat komunikační technologií, bylo nutné zajistit vytvoření řídicích programů, které zajistí automatizaci modulů, tak aby bylo od obsluhy vyžadováno jen zapnutí, přerušování běhu modulu, přepnutí rozlišování modulu nebo doplnění zásobníku, pokud se vyprázdní. Pro vytvoření programů pro PLC systémy Simatic S7 bylo použito vývojového prostředí TIA Portal. Vytvoření programů v tomto prostředí bylo přehledné a zároveň mi umožnilo skloubit jak samotné vytváření programů, tak i přípravu a následné vytvoření komunikace s nastavením samotného Simatic.Net OPC serveru. U programů bylo nutné se zaměřit na problém najetí manipulátorů do výchozích krajních pozic. U pneumatického systému je signalizace zajištěna pomocí senzorů umístěných na

krajních pozicích jednotlivých manipulátorů. Proto, když dojde k zastavení manipulátoru v jiné než krajní pozici, a tento manipulátor nemá žádný mechanismus, co zajistí jeho vrácení do krajní pozice, nelze pak určit jeho pozici. To může mít za následek výpadek funkčnosti celého modulu. Tento problém byl vyřešen sepnutím pohonu pro najetí příslušného manipulátoru do krajní pozice, pokud bylo detekováno, že se tento manipulátor nenachází ani v jedné krajní pozici. Kromě najetí do krajních pozic, musela být v programech zohledněna i detekce stavu a průběhu činnosti modulu. Toho bylo dosaženo pomocí vytvoření kontrolních bodů, které se aktivují postupně a tím signalizují průběh linky. Tyto body nám zajistí, že při vizualizaci nemusíme řešit žádné složité funkce nebo procedury co by vyhodnocovaly stav jednotlivých manipulátorů. Ve vizualizaci nám tedy stačí jen kontrolovat hodnotu těchto bodů a hned víme, v jakém stavu se modul nachází.

Pro řízení robota MELFA jsme využili ImagingLab Robotics Library modul pro vývojové prostředí Labview. Tohle bylo jediné možné řešení jak umožnit a zajistit ovládání robota společně s dalšími moduly z jediného programu. Jiné varianty skýtaly využití RT-Toolboxu nebo manuální ovládání obsluhou pomocí ovládacího panelu. Takové řešení by bylo ovšem nepřijatelné, obzvláště v případě, kdy bychom do procesu přidali ještě další jiné technologie nebo přístroje. V případě Labview není připojení k OPC technologie žádný problém. Samotný výrobce tohoto prostředí dokonce nabízí svoji vlastní implementaci OPC serveru, který má k dispozici ovladač pro připojení k Simatic S7 PLC systémům a zároveň by nám umožnil i přidat prvky pro řízení a vizualizaci robota MELFA. Ovšem tato možnost má nevýhodu v nemalé ceně poplatku za licenci takového serveru.

Vizualizace v prostředí Control Webu se ukázala jako jednoduchá a intuitivní a díky mnoha možnostem, které toho prostředí nabízí, poskytuje rozsáhlé možnosti vizualizace. Připojení k OPC serveru je ze začátku trochu matoucí a nepřehledné, ale poté co člověk přijde na to, jak správně využít nástroj pro konfiguraci připojení k OPC serveru a následný přístup k jednotlivým kanálům, se tyto problémy začnou pomalu vytrácet.

Díky použití technologie DCOM pro připojení ke vzdáleným OPC serverům se někdy můžeme setkat s neochotou připojení klientů k těmto vzdáleným serverům. To nemusí být zapříčiněno technologií samotnou, ale interakcí s jinými běžícími programy či službami na jednotlivých stanicích. Proto je dobré vypnout různé firewally a ochranné prvky, které by mohly tuto komunikaci rušit a zajistit, aby síť, pomocí které jsou jednotlivé stanice propojeny, byla bezpečná. Ovšem může se stát, že i tyto kroky nepovedou k úspěšnému navázání spojení. Proto někteří výrobci OPC serverů začínají nebo nabízí jejich vlastní implementaci komunikace, která obchází standardní DCOM komunikaci a zajistí nám jednodušší a bezproblémovou komunikaci mezi prvky OPC technologie.

Poznatky získané v této práci mohou posloužit jako základní kámen pro vytváření rozsáhlejšího automatizačního procesu, nebo zajistit jeho vylepšení. To platí hlavně v oblasti řízení robota MELFA a kontroly pracovního prostoru jednotlivých modulů nebo celé linky. Zároveň lze práci použít pro didaktické účely

nebo demonstraci řízení různých přístrojů nebo technologií sloučených v jeden proces.

7 Literatura

AUTOMA. *Nový, menší a výkonnější robot MELFA RV-2AJ* [online]. 2002 [cit. 2015-04-20]. Dostupné z: http://automa.cz/index.php?id_document=28493

BERGER, H. *Automatisieren mit SIMATIC: Controller, Software, Programmierung, Datenkommunikation, Bedienen und Beobachten*. Erlangen: Publicis Publishing, 2012. ISBN 978-3-89578-386-9.

ENGINEERS GARAGE. MATHUR, ABHIMANYU. *Insight - How Micro Switch Works* [online]. 2012 [cit. 2015-04-11]. Dostupné z: <http://www.engineersgarage.com/insight/how-micro-switch-works>

FARGO CONTROLS. *Operating principles for inductive proximity sensors* [online]. 2013 [cit. 2015-04-12]. Dostupné z: http://www.fargocontrols.com/sensors/inductive_op.html

FOXON. *Co je OPC ? OPC Server ? OPC Klient ?* [online]. 2014 [cit. 2015-04-22]. Dostupné z: <http://www.foxon.cz/cs/blogs/80-co-je-opc-opc-server-opc-klient.html>

GOUBEJ, Martin, Martin ŠVEJDA a Miloš SCHLEGEL. *Úvod od mechatroniky, robotiky a systémů řízení pohybu* [online]. Plzeň, 2012 [cit. 2015-04-20]. Dostupné z: <http://home.zcu.cz/~msvejda/URM/materialy/Uvod%20do%20mechatroniky.pdf>. Skriptum pro studenty doktorských programů v oboru automatické řízení. Západočeská univerzita v Plzni.

IHS ENGINEERING360. *Linear Potentiometers Information* [online]. 2015 [cit. 2015-04-11]. Dostupné z: http://www.globalspec.com/learnmore/sensors_transducers_detectors/linear_position_sensing/linear_potentiometers

IMAGINGLAB. *ImagingLab Robotics Library for MITSUBISHI* [online]. 2015 [cit. 2015-04-22]. Dostupné z: http://www.imaginglab.it/eng/MitsuLib_DataSheet.php

LIONPRECISION. *Capacitive Sensor Operation and Optimization* [online]. 2012 [cit. 2015-04-12]. Dostupné z: <http://www.lionprecision.com/tech-library/technotes/cap-0020-sensor-theory.html>

MORAVSKÉ PŘÍSTROJE. *OPC server pro ovladače systému Control Web* [online]. 2005 [cit. 2015-04-22]. Dostupné z: <http://www.mii.cz/art?id=214&lang=405>

MORAVSKÉ PŘÍSTROJE. *Co je Control Web* [online]. 2010 [cit. 2015-05-03]. Dostupné z: <http://www.mii.cz/art?id=380&cat=146&lang=405>

OPCF FOUNDATION. *Organization* [online]. 2015 [cit. 2015-04-22]. Dostupné z: <https://opcfoundation.org/about/opc-foundation/mission-statement/>

PIKULA, Jiří. *Návrh absolutních snímačů polohy s hallovými sondami* [online]. Brno, 2009 [cit. 2015-04-20]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_ve-rejne.php?file_id=18375. Diplomová práce. Vysoké učení technické Brno.

REHG, James A Glenn J SARTORI. *Programmable logic controllers: programmable logic controllers made easy*. 2nd ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2009, xviii, 600 p. ISBN 01-350-4881-8.

SENSORCENTRAL.COM. *Photoelectric Sensors* [online]. 2014 [cit. 2015-04-12]. Dostupné z: <http://www.sensorcentral.com/photoelectric/fiber01.php>

SIEMENS. *SIMATIC S7-300* [online]. 2014 [cit. 2015-04-22]. Dostupné z: <http://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-300/Pages/Default.aspx>

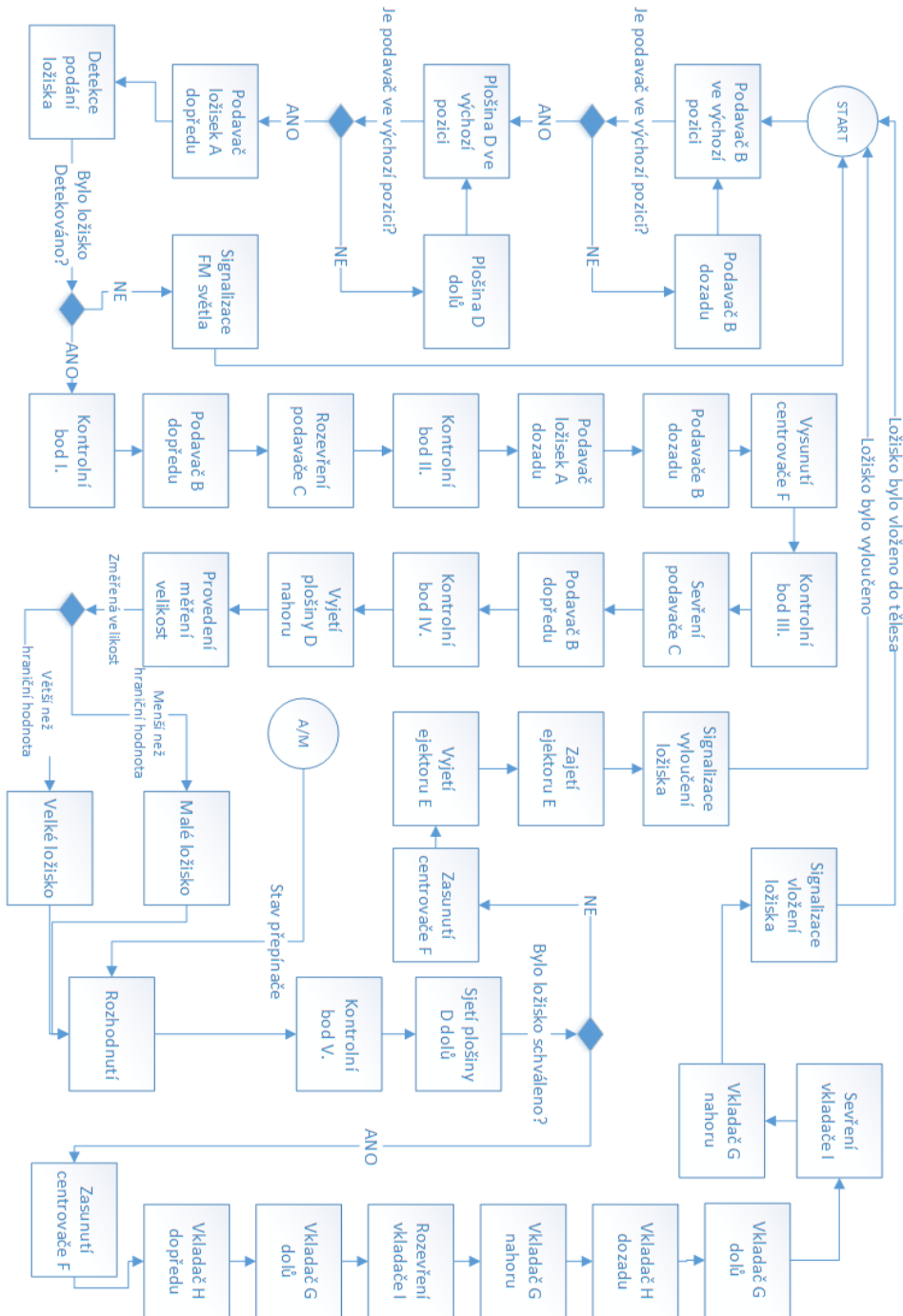
SMC INTERNATIONAL TRAINING. *FMS-200 - Flexible integrated assembling systems* [online]. 2015 [cit. 2015-04-11]. Dostupné z: <http://www.smctraining.com/webpage/indexpage/431/>

VANČURA, Tomáš. *Pneumatické akční členy a jejich řízení* [online]. Sezimovo Ústí, 2007 [cit. 2015-04-14]. Dostupné z: <http://vanto.sweb.cz/mk/Pneumatika.doc>. Výukový text. Vyšší odborná škola Střední škola Centrum odborné příprav.

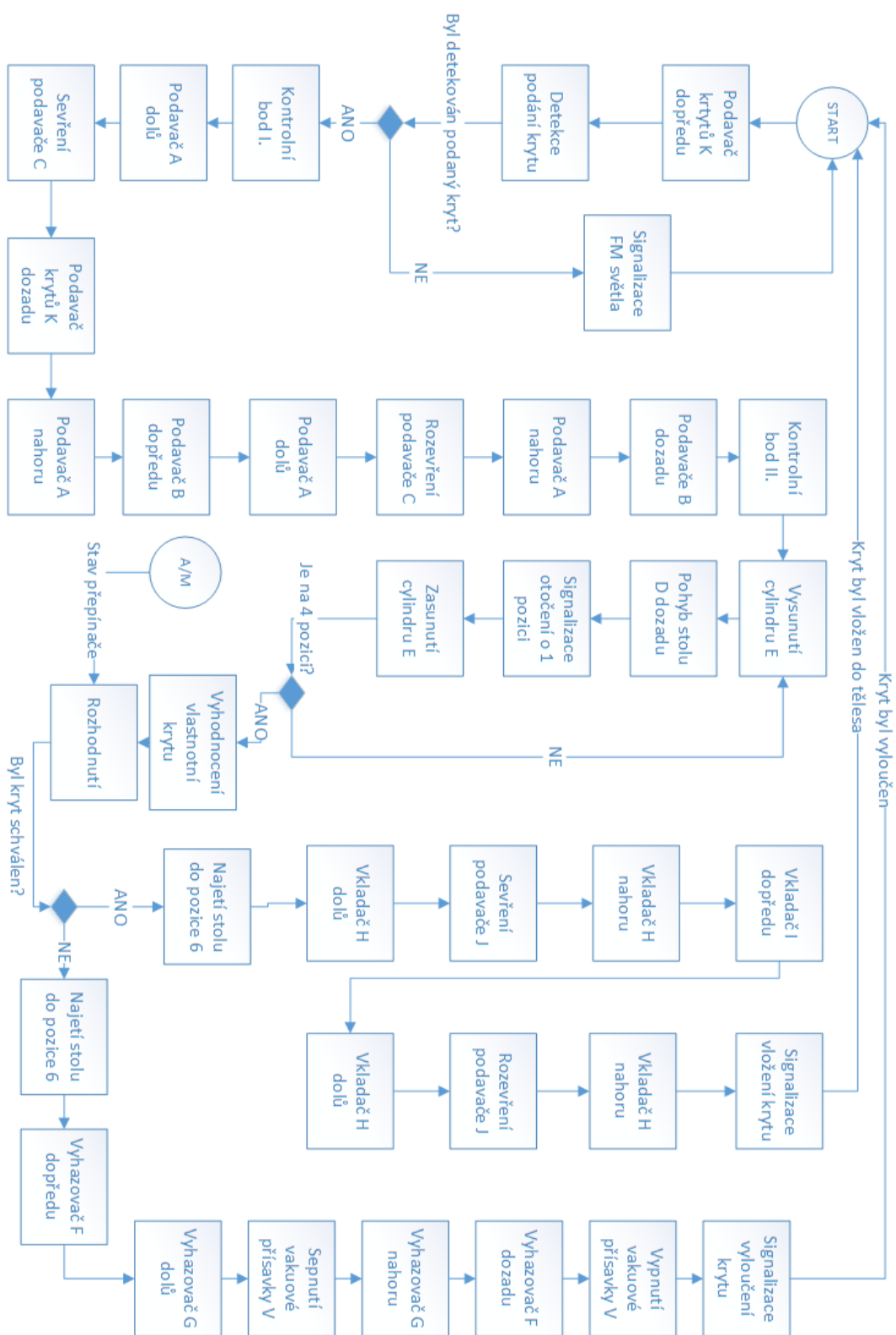
WALCHER, Hans. *Position sensing: angle and distance measurement for engineers*. New York: Butterworth-Heinemann, 1994, xi, 256 p. ISBN 07-506-1157-X.

Přílohy

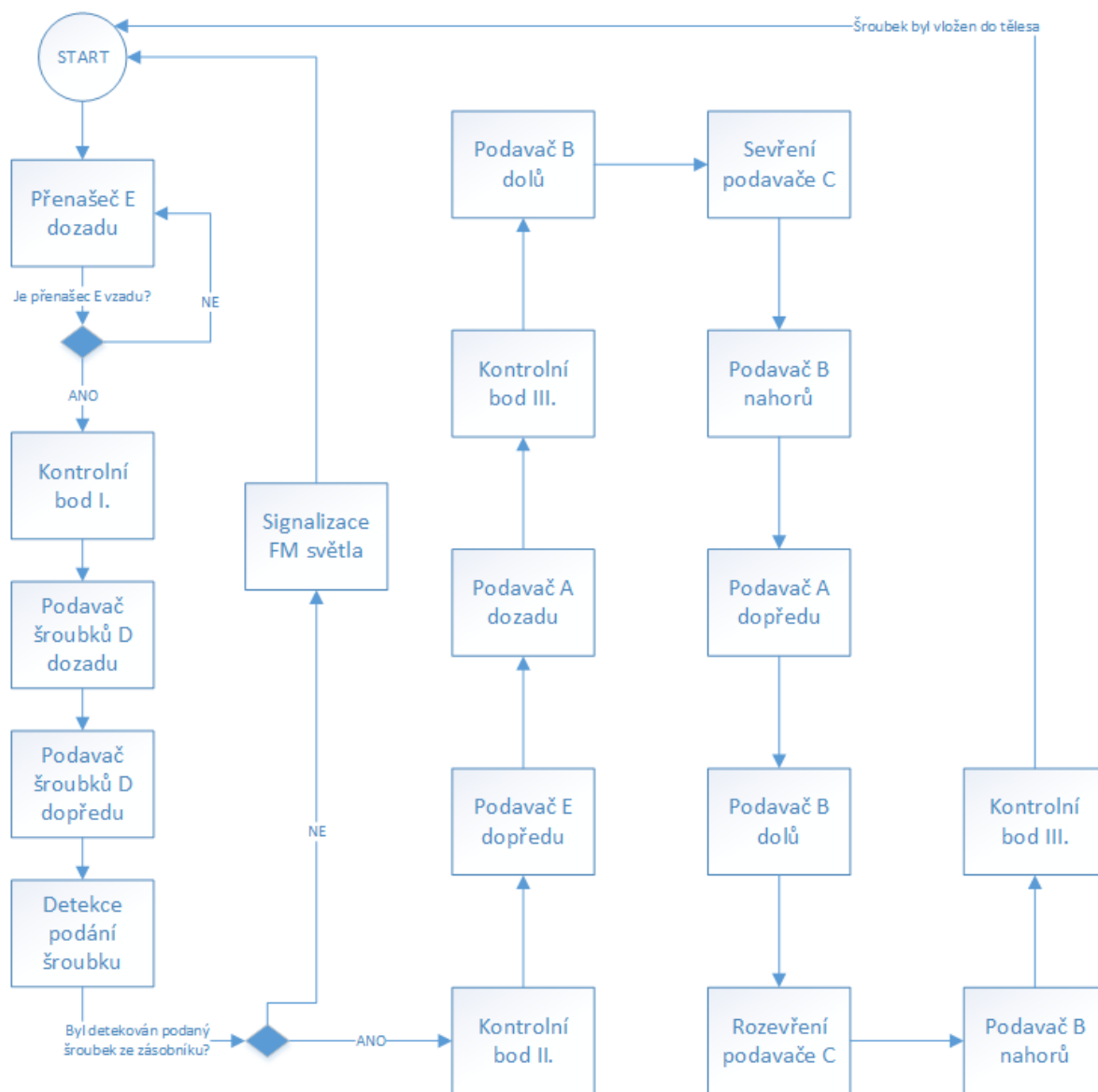
A Funkční schémata modulů linky FMS-200



Obr. 36 Funkční schéma modulu FMS-202



Obr. 37 Funkční schéma modulu FMS-205



Obr. 38 Funkční schéma modulu FMS-206

B Obsah příloženého CD

Optické médium přiložené k tištěné verzi práce obsahuje:

- Vyexportované řídicí programy pro jednotlivé moduly linky FMS-200
- Projekt TIA Portal s řídicími programy pro PLC moduly, který obsahuje i nastavený Simatic.Net OPC server
- Konfigurace pro MatrikonOPC Simulation Server
- Vyexportovaný csv soubor obsahující aliasy pro MatrikonOPC Simulation Server
- Konfigurace pro MatrikonOPC Explorer
- Parametrický a mapovací soubor OPC serveru pro Control Web
- Vizualizační programy v Control Webu pro moduly linky FMS-200
- Programy pro Labview nutné pro čtení a zápis dat z/na OPC server
- Programy pro Labview reprezentující funkci a proceduru pro ovládání a čtení dat pro robota MELFA
- Elektronická verze práce ve formátu pdf