



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

Implementace řídicího systému robota COMAU

Bakalářská práce

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Autor práce: **Vít Benda**

Vedoucí práce: Ing. Leoš Beran, Ph.D.

Konzultant: Ing. Martin Diblík, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Vít Benda**
Osobní číslo: **M14000101**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronické informační a řídicí systémy**
Název tématu: **Implementace řídicího systému robota COMAU**
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s řídicím systémem robota COMAU.
2. Seznamte se s možnostmi implementace knihovny MpRoboX.
3. Vytvořte SW pro ovládání základních pohybů robota COMAU.
4. Vytvořený SW přehledně zdokumentujte.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **30–40 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] JOHN, Kharl-Heinz; TIEGELKAMP, Michael. IEC 61131-3 Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision - Making Aids. 2nd Ed.
- [2] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. 1. díl. Praha: BEN, 1999. ISBN 80-8605-58-9
- [3] NOVOTNÝ, František a Marcel HORÁK. Konstrukce robotů. Vydání 1. Liberec: Technická univerzita v Liberci, 2015. ISBN 978-80-7494-216-7.

Vedoucí bakalářské práce: **Ing. Leoš Beran, Ph.D.**

Ústav mechatroniky a technické informatiky

Konzultant bakalářské práce: **Ing. Martin Diblík, Ph.D.**

Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: **10. října 2017**

Termín odevzdání bakalářské práce: **14. května 2018**

prof. Ing. Zdeněk Plíva, Ph.D.
děkan



doc. Ing. Milan Kolář, CSc.
vedoucí ústavu



V Liberci dne 10. října 2017

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

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

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

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 22.12.2018

Podpis: *Benda*

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu práce Ing. Leoši Beranovi Ph.D. za poskytnutí znalostí a rad potřebných k vytvoření předmětné práce. Poděkování patří také školitelům Ing. Petru Hojnému a Ing. Viktoru Masárovi z firmy B&R. A nakonec odborníkům z firmy B&R, Ing. Ladislavu Robovi, Ing. Martinu Markvartovi a Ing. Tomáši Matuchovi za pomoc s oživením robota.

Abstrakt

Bakalářská práce seznamuje s aplikací pro naprogramování základních pohybů robota RACER 3 od firmy COMAU. Jsou zde popsány použité prvky hardwaru, jako jsou průmyslový počítač, servozsilovač, napěťový zdroj, displej, komunikace a samotný robot. Také seznamuje s použitými programovacími jazyky v aplikaci. Jedná se o textový jazyk ST a grafický jazyk Ladder diagram. V práci byla využita revoluční Mapp technologie, která zrychluje celý proces programování. Je zde stručný popis postupu vytvoření aplikace ve vývojovém prostředí Automation studio a také možnost vizualizace robota v programu B&R Robotic Visualization. Zároveň je popsána i implementace knihovny MpRoboX.

Klíčová slova: Hardware, Software, Mapp technologie.

Abstract

This bachelor's thesis focuses on an application designed to program basic movements of COMAU company's robot RACER 3. The thesis provides a description of used hardware components such as an industrial computer, a servo driver, a power supply unit, a display, communication components, and the robot itself. Further, the paper deals with programming languages that were employed for creating the application, namely a textual programming language ST (Structured Text) and a graphical programming language Ladder Diagram. A revolutionary Mapp Technology was employed as well, making the entire programming process faster. The thesis also includes a brief description of the development process of the application in the software development environment Automation Studio. Moreover, it discusses the option of visualization of the robot in the B&R Robotic Visualization program. Finally, it is concerned with the implementation of MpRoboX library as well.

Keywords: Hardware, Software, Mapp technology.

Obsah

Prohlášení.....	4
Poděkování	5
Abstrakt.....	6
Abstract	6
Obsah.....	7
Seznam použitých obrázků	9
Seznam použitých zkratek	10
Úvod.....	11
1 Hardware.....	12
1.1 Průmyslové PC	12
1.2 Servozesilovač.....	13
1.3 Displej	14
1.4 Napájecí zdroj.....	15
1.5 Komunikace.....	16
1.6 Robot	17
1.7 Safety	20
2 Software.....	20
2.1 Strukturovaný text (ST)	21
2.2 Ladder diagram.....	22
2.3 Vytvořené tasky v projektu.....	22
2.4 Nahrané programy v APC910	28
3 Vývojové prostředí.....	29
3.1 Hlavní části.....	30
3.2 Diagnostické nástroje.....	33
3.3 Task.....	35
4 Mapp Technologie	35

4.1	MpAxisBasic	36
4.2	MpRoboArm6Axis.....	37
5	Postup vytvoření projektu v AS.....	38
5.1	Knihovny.....	40
5.2	Knihovna MpRoboX	40
5.3	Ladění aplikace	40
6	B&R Robotic Visualization	40
	Závěr	42
	Použité zdroje.....	43

Seznam použitých obrázků

Obr. 1 – HW konfigurace	12
Obr. 2 – APC 910 [22]	13
Obr. 3 – ACOPOS P3 [20]	14
Obr. 4 – Automation panel 5000	14
Obr. 5 – Napájecí zdroj [21]	15
Obr. 6 – Komunikační model Ethernet Powerlink	16
Obr. 7 – RACER 3 [13]	17
Obr. 8 – Pracovní prostor robota [13]	18
Obr. 9 – Umístění os a motorů v robotu [13]	19
Obr. 10 – Funkční blok MpAxisBasic v Ladder diagramu	23
Obr. 11 – Funkční blok MpRoboArm6Axis v Ladder diagramu	23
Obr. 12 – Vývojový diagram pro pohyb jedné osy	25
Obr. 13 – Vývojový diagram pro pohyb robota	26
Obr. 14 – Control error	27
Obr. 15 – Transport position	28
Obr. 16 – Home position	29
Obr. 17 – Hlavní části Automation studia	30
Obr. 18 – Struktura Test okna	34
Obr. 19 – MpAxisBasic [3]	36
Obr. 20 – MpRoboArm6Axis [3]	37
Obr. 21 – B&R Robotic visualization [4]	41

Seznam použitých zkratek

CNC – (Computer Numerical Control) počítačem řízený obráběcí stroj

PCI – je dle [8] sběrnice, využívá paralelní přenos dat

PCI Express – je dle [8] sběrnice, která používá sériový přenos dat a umožňuje zvyšování frekvence, na které sběrnice pracuje

SDL – (Simple DirectMedia Layer) je dle [6] multiplatformní multimediální knihovna poskytující nízko-úrovňový přístup na audio, klávesnici, joystick, 2D a 3D počítačovou grafiku

SDL3 – (Smart Display Link 3) dle [15] přenáší všechny komunikační kanály mezi PC a panelem prostřednictvím standardního ethernetového kabelu

DVI – (Digital Visual Interface) je dle [7] rozhraní pro propojení videozařízení s počítačem

PC – (Personal computer) osobní počítač

PLC – (Programable Logical Controller) programovatelný logický automat je číslicový elektronický systém navržený pro použití v průmyslovém prostředí, a to za účelem řízení různých typů výrobních strojů a procesů

PWM – (Pulse Width Modulation) pulzně šířková modulace

IEC – je dle [14] mezinárodní elektrotechnická komise, která vypracovává a publikuje mezinárodní normy v elektrotechnice

B&R – firma zabývající se oborem průmyslové automatizace

AS – (Automation studio) vývojové prostředí od firmy B&R

ST – programovací jazyk využívaný například k řízení PLC

CPU – centrální procesorová jednotka

Úvod

Cílem této bakalářské práce bylo použít již zrealizovaný hardware pro řízení robota, vytvořit vhodný software pro ovládání jeho základních funkcí, vytvořený program odzkoušet, odladit a přehledně popsat. Účelem práce bylo spojit a oživit robota RACER 3 od firmy COMAU s hardwarem od firmy B&R. Využitý software je také od firmy B&R, jmenovitě se jedná o vývojové prostředí Automation Studio. Celá aplikace byla vytvořena pouze pro výstavní účely robota.

Práce je rozdělena do několika částí. První část je hardwarová, kde je uveden popis použitého hardwaru k této aplikaci. Softwarová část se věnuje použitým jazykům, vytvořeným programům v projektu a nahraným programům do APC910. Další části popisují použité vývojové prostředí a Mapp technologie. V posledních dvou částech práce je popsán stručný postup vytvoření projektu v Automation studiu a program B&R Robotic Visualization pro simulaci pohybů robota.

1 Hardware

Veškeré použité komponenty hardwaru, kromě robota, jsou od firmy B&R. Jde o průmyslový počítač, servozesilovač, safety logiku, panel pro ovládání, komunikaci a napěťový zdroj. Jednotlivé komponenty jsou popsány níže. Na Obr. 1 uvádím HW konfiguraci pro tuto aplikaci. Na obrázku chybí jen napěťový zdroj a samotný robot.



Obr. 1 – HW konfigurace

1.1 Průmyslové PC

Počítač je dle [11] v informatice zařízení a výpočetní technika, která zpracovává data pomocí předem vytvořeného programu. Skládá se z hardwaru a softwaru. Hardware obsahuje všechny fyzické prvky počítače. Software tvoří operační systém, veškeré programy a algoritmy v počítači.

Pokud chceme z PC udělat PLC, tak musíme dle [10] přidat PCI-karty, softwarové a hardwarové ovladače a jádro operačního systému pracující v reálném čase. Naopak některé PLC mají zabudované PC a ke správnému chodu stačí připojit klávesnici a myš. Průmyslový počítač, je v podstatě malý typ obyčejného PC, pouze má v sobě software, který pracuje v reálném čase (real-time). Průmyslové počítače, se využívají v průmyslu, a to v podstatě ve všech jeho odvětvích. Dále se využívají v dopravě, armádě a lékařství. Základní charakteristikou průmyslového počítače je vysoký výkon a konstrukce se zřetelem na odolnost a spolehlivost. Klade se důraz na odolnost proti prachu, vlhkosti, teplotě a vibracím.

APC 910



Obr. 2 – APC 910 [22]

Na Obr. 2 je APC 910 a dle [5] se využívá v průmyslových aplikacích k řízení a vizualizaci průmyslových procesů i k provozování dalších softwarových aplikací. Jeho výhodami jsou velký výpočetní výkon, značná odolnost, způsob chlazení, spolehlivost a dlouhá životnost. Vzduch je zde cirkulován přes chladičí žebra, to znamená, že není použit žádný větrací zařízení, což zajišťuje snazší a rychlejší údržbu.

APC je dvou slotový a má dvou jádrový procesor Intel core i5 3610ME 2.7GHz, který zajišťuje vysoký výpočetní výkon, grafickou kartu Intel HD Graphics 4000, operační paměť DDR3 4096MB. Díky paměti SRAM 2 MB je velmi rychlý. Obsahuje také kartu pro komunikaci se servozsilovačem, pomocí protokolu POWERLINK a dva gigabitové ethernetové porty pro komunikaci s uživatelským PC. Pro komunikaci mezi základní deskou a perifériemi jsou použity sběrnice PCI a PCI express.

1.2 Servozesilovač

Servozesilovač obsahuje dle [1] prostředky pro napájení motoru, řízení a ovládání elektrického pohonu pomocí nadřazeného systému PLC, v tomto případě PC. Obsahuje také rychlostní a polohový regulátor, regulátory pro každou fázi a střídač k napájení motoru. K napájení synchronního motoru se využívá PWM. Jeho hlavní funkcí je převod síťového napájecího napětí na napětí vhodné pro napájení motoru pomocí amplitudy a frekvence.

ACOPOS P3



Obr. 3 – ACOPOS P3 [20]

Na Obr. 3 je ACOPOS P3 a slouží pro řízení 3 pohonů jednotlivých os v robotu, v tomto případě je použit 2 x, první ACOPOS má navíc oproti druhému ACOPOSU ovládání brzd a ovládá osy 1,3 a 5, druhý ovládá 2, 4, a 6 osu. Umožňuje velmi rychlé a spolehlivé řízení každé osy zvlášť. Díky encodéru v motoru (což může být například inkrementální rotační snímač) zjistíme přesnou polohu a rychlost motoru nebo mechaniky. Minimální vzorkovací čas má 50 μ s. Má také integrovanou safety logiku, která je popsána v sekci Safety. Hlavní důvod výběru tohoto servozesilovače byl, že má možnost ovládat 3 osy najednou, což ušetří místo v rozvaděči a počet kabelů. Další důvod je vyzkoušení zcela nového servozesilovače na aplikaci s robotem.

1.3 Displej



Obr. 4 – Automation panel 5000

Jedná se o Automation Panel 5000 a je na Obr. 4. Je to 21,5 palcový fullHD displej s rozlišením 1920x1080 a s možností ovládní pomocí multi-touch. Multi-touch systémy mají své výhody, například: intuitivní přetáčení, přiblížení, posouvání gest a gesta pomocí obou rukou, které zabraňují neúmyslnému spuštění kritických operací. Panel má otočné rameno, to znamená, že je velmi flexibilní. Panel obsahuje ochranu IP65, dále také možnost zamčení displeje pomocí klíče, bezpečnostní total stop tlačítko, dva přepínače a dvě další tlačítka, červené a zelené. Veškeré komponenty na panelu mohou sloužit k jakýmkoli potřebám určeným uživatelem. Jeho hlavní výhody jsou dobrá flexibilita, možnost ovládní v rukavicích a připojení pro DVI, SDL a SDL3. Dotykový displej u aplikace je použit jako ovladač.

1.4 Napájecí zdroj



Obr. 5 – Napájecí zdroj [21]

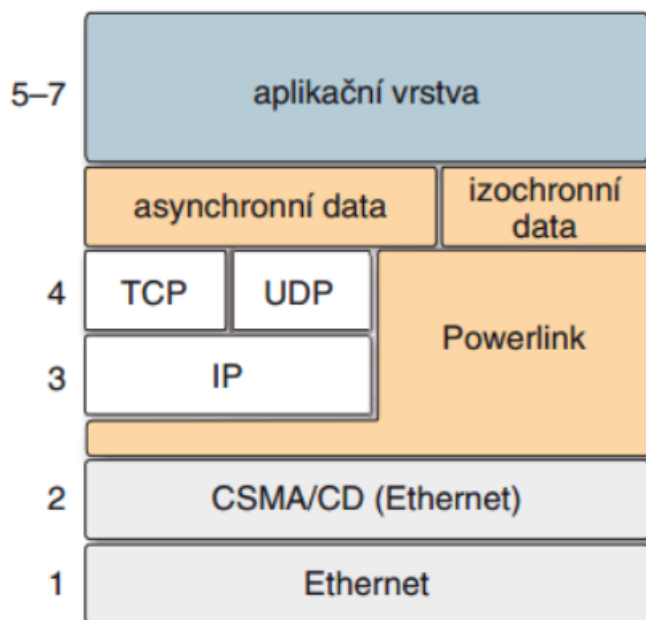
Jedná se o jedno fázový napěťový zdroj, viz Obr. 5. Zdroj převádí vstupní střídavé napětí na stejnosměrné napětí až s výkonem 240 W při 24 V s 10 A. Poskytuje regulované, nastavitelné výstupní napětí. Rozsah vstupního napětí je 100 až 240VAC. Má také ochranné funkce, jako jsou ochrana výstupu proti nepřetržitému zkratu, otevřenému obvodu a přetížení. Využívá se pro udržení konstantního výstupního napětí. Tento zdroj slouží k napájení celého řídicího systému.

1.5 Komunikace

Je nutná rychlá komunikace mezi uživatelským počítačem, servozsilovači, displejem a samotnými motory. K této aplikaci se využily sítě Ethernet a Powerlink. Detailnější informace o Powerlinku jsou uvedeny níže.

Ethernet slouží v této aplikaci pro komunikaci mezi počítačem APC910 a naším uživatelským počítačem, ze kterého můžeme kopírovat, vytvářet programy nebo přímo ovládat robota pomocí vývojového prostředí.

Ethernet Powerlink



Obr. 6 – Komunikační model Ethernet Powerlink

Ethernet Powerlink (EPL) je dle [3] deterministický real-time protokol, založen na standardu Ethernetu. Přenosová rychlost je 100 Mb/s a poskytuje maximální výkon s dobou cyklu až 100 μ s. EPL umožňuje využití sítě s jakoukoli topologií. Na Obr. 6 vidíme jeho komunikační model. Z něj je patrné, že zatímco přenosy časově nekritických dat se uskutečňují protokoly TCP/UDP/IP, přenosy časově kritických dat (izochronní přenos) probíhají mezi linkovou a aplikační vrstvou speciálním protokolem dle standardu IEEE 1588.

Ethernet Powerlink je vhodný pro aplikace, kde je potřeba rychle a synchronně řídit více servopohonů, vstupy a výstupy, vizualizaci a výměnu dat mezi programovatelnými automaty. V této aplikaci slouží pro komunikaci mezi počítačem APC910, servozesilovači a safety modulem.

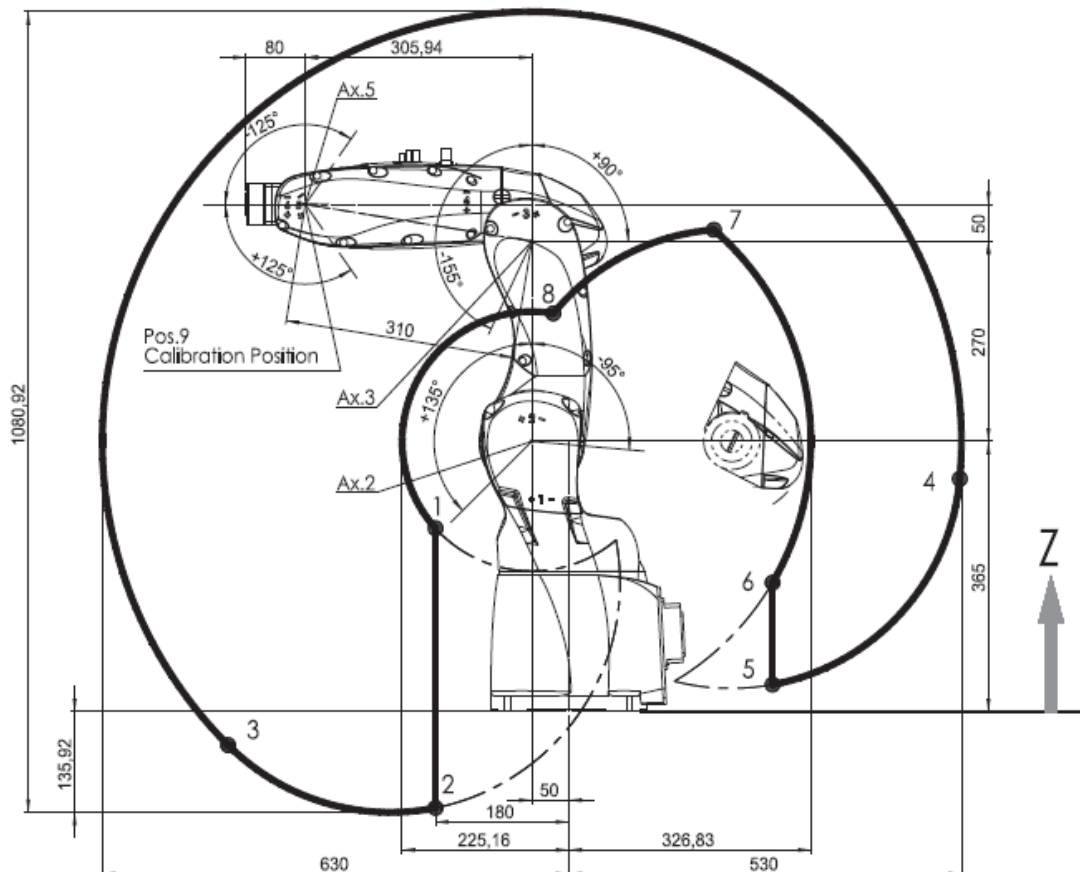
1.6 Robot



Obr. 7 – RACER 3 [13]

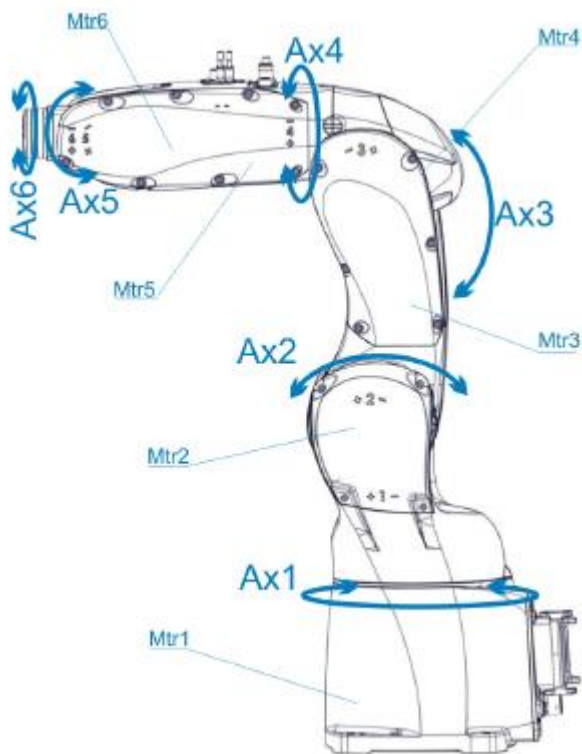
Jedná se o 6-ti osého angulárního robota RACER 3 dle [13] viz Obr. 7. Maximální zatížení na přírubě robota jsou 3 kg. Robot váží 30 kg a má ochranu IP54, která dle [12] zajišťuje ochranu proti prachu a dotyku drátem živých částí. Robot má konektory pro signály z encodéru, signály pro řízení brzd a motorů, vstupní a výstupní signály a možnost připojení ethernetu. Je zde také možnost připojení tlakového vzduchu, pomocí kterého, můžeme například využít přísavku nebo jiný efektor pro chycení tělesa. Efektor je dle [18] výkonným subsystémem robota. Je to například hlavice, pracovní hlavice, chapadlo, koncový efektor.

Rychlost a možnosti rotace jsou u každé osy různé. Z Obr. 8 vidíme, že robot má velkou pracovní plochu a možnost pohybu v záporných hodnotách osy Z.



Obr. 8 – Pracovní prostor robota [13]

Na Obr. 9 je vidět, kde se nachází jednotlivé osy a motory v robotu. Jsou v něm také vidět směry rotace dané osy. Osu 1 lze rotovat od -170° do 170° , rychlost je až $440^\circ/\text{s}$. Osu 2 lze rotovat od -95° do 135° , rychlost je až $450^\circ/\text{s}$. Osu 3 lze rotovat od -155° do 90° , rychlost je až $500^\circ/\text{s}$. Osu 4 lze rotovat od -200° do 200° , rychlost je až $600^\circ/\text{s}$. Osu 5 lze rotovat od -125° do 125° , rychlost je až $600^\circ/\text{s}$. Osu 6 lze rotovat od -2700° do 2700° , rychlost je až $900^\circ/\text{s}$.



Obr. 9 – Umístění os a motorů v robotu [13]

Robot využívá ve všech šesti osách AC Brushless motor. Každý motor má i integrovanou brzdu a enkodér (senzor polohy). U první a druhé osy je totožný motor 1.27NM a 1,8 kg. U třetí osy je motor 0.64NM a 1,4kg. U páté a šesté osy je motor 0.32NM a 0,9 kg.

Brushless AC motor

Je to dle [9] elektronicky komutovaný střídavý motor. Někdy se také nazývá střídavý bezkomutátorový motor. Je to kvalitativně vyšší typ bezkartáčových motorů. Může se chovat i jako stejnosměrný motor s cizím buzením.

Motor obsahuje statorové trojfázové vinutí a rotor s permanentními magnety. Stator je napájen harmonickým průběhem proudu, který vytvoří točivé magnetické pole. Permanentní magnety v rotoru také vytvoří točivé magnetické pole. Pro řízení motoru je nutné sledovat polohu rotoru a transformátor souřadnic místo elektronického komutátoru. Mezi výhody tohoto motoru patří vysoká dynamika, vysoká statická přesnost, přesná synchronizace rychlosti otáčení ve skupinových pohonech a výkon v rozsahu stovek W a desítek kW. Nevýhodou je vysoká cena motoru.

1.7 Safety

Cílem safety je ochrana lidí, výrobků a samotných strojů při nějaké neočekávané události. Například robot má svůj pracovní prostor, do kterého když vnikne nějaká cizí osoba nebo předmět, tak právě úkolem safety je zastavit pohyb robota, nebo jen snížit rychlost, aby nedošlo k nějaké újmě na zdraví osoby nebo poškození robota. Bezpečnostní systémy lze naprogramovat podle návrhu zákazníka.

Použité safety moduly v aplikaci od firmy B&R jsou X20SL8101 (SafeLOGIC CPU), X20SI4100 (Safe digital inputs modul) a SX20SO4110 (Safe digital outputs modul). CPU Modul je vybaven funkcí SafeLOGIC, která umožňuje bezpečně provádět aplikace navržené v SafeDESIGNER (softwarový nástroj pro návrh bezpečnostních systémů). Nyní vám popíši, jak je safety v této aplikaci řešeno.

Nouzové stop tlačítko (Emergency stop) je připojeno dvoukanálově na digitální vstupy 1 a 2 modulu X20SI4100. Při každém opětovném nahození Emergency stop je potřeba resetovat safety systém, jinak safety systém nebude funkční. Reset tlačítko je připojeno jednocanálově na digitální vstup 3 modulu X20SI4100. V safety projektu je použit pouze jeden funkční blok SF_EmergencyStop, který monitoruje Emergency stop tlačítko. Pokud Emergency stop tlačítko není stisknuto, tak výstup funkčního bloku je aktivní. Tento signál je vyveden na digitální výstupy 1 a 3 modulu X20SO4110, kde jsou připojeny Enable vstupy od Acoposů P3, které řídí motory robota. Pokud tedy Emergency stop tlačítko je stisknuté, Enable vstup na Acoposu je shozen, a tudíž dojde k vypnutí kontrolérů. Následkem toho se robot zastaví (aktivují se brzdy) a dále s robotem není možno manipulovat. Pro opětovnou manipulaci s robotem je potřeba nahodit Emergency stop tlačítko a poté zmáčknout Reset tlačítko.

2 Software

K vytvoření softwaru pro tuto aplikaci jsem využil 2 programovací jazyky. Pro nadefinování mapp komponent (popsáno v sekci Mapp Technologie) jsem použil jazyk Ladder diagram a to hlavně pro jeho jednoduchost a rychlost. Jazyk ST

(strukturovaný text) pro algoritmus na kontrolu chyb, stavů robota a chybové hlášení. Níže popisují nahrané programy v PLC, vytvořené Tasky (Task je popsán v sekci Task), dle [16] Ladder diagram, dle [2] jazyk ST a datové typy dle normy IEC61131-3.

2.1 Strukturovaný text (ST)

Jedná se o vyšší textový jazyk dle standardu IEC 61131-3. Tento jazyk má své předky v jazycích Ada, Pascal a C. V jazyku ST se využívá také Booleova algebra. Jazyk nabízí rychlé a jednoduché vytváření aplikací například v automatizačním průmyslu. Program plyne postupně po řádcích. To znamená, že nejdříve se vykoná příkaz na prvním řádku, pak druhý, třetí až do posledního. Je zde možné vytváření cyklů, časovačů, logických a aritmetických operací, volání akcí a využívání funkčních bloků.

Samotný jazyk můžeme rozdělit na dvě hlavní části a to deklarační a výkonnou. V deklarační části definujeme proměnné, které pak používáme ve výkonné části. Zejména to jsou proměnné nutné pro správný chod výkonné části. Můžeme v ní definovat název proměnné, jakého je typu například INT, BOOL, TON atd., popis proměnné, ale třeba i výchozí stav proměnné. Ve výkonné části definujeme daný algoritmus, který chceme, aby program vykonával. Výkonná část se může rozdělit na tři části. První z nich je Initní část. Ta se vykoná jako první při spuštění PLC a v ní nastavujeme proměnné nebo algoritmy, které buď chceme anebo nám stačí, aby se vykonaly pouze jednou a to po zavolání vybraného programu. Druhá část se nazývá cyklická. To je část programu, kterou chceme například vykonávat opakovaně v cyklu, dokud program neskončí. Poslední část je Exitní a to je část, která se jako Initní, vykoná pouze jednou, jen se vykoná až po uvedení PLC do režimu STOP. Tato část se moc nevyužívá, protože pokud chceme nastavit nějaké proměnné či algoritmy po ukončení programu, můžeme uvedené využít vlastně v Initní části, která se provede před cyklickou částí.

Je potřeba jako i u jiných programovacích jazyků dodržet danou syntaxi psaní a snažit se psát co nejvíce přehledně a srozumitelně, aby program pochopil jiný programátor, ale i vy, když ho otevřete za 10 let, a chcete z něho čerpat nebo ho upravovat. Program se poté dá využít i u následujících aplikací, což samozřejmě šetří

drahý čas, peníze ale i spolehlivost. Tímto tématem se zabývám v níže uvedeném tématu Mapp Technologie.

2.2 Ladder diagram

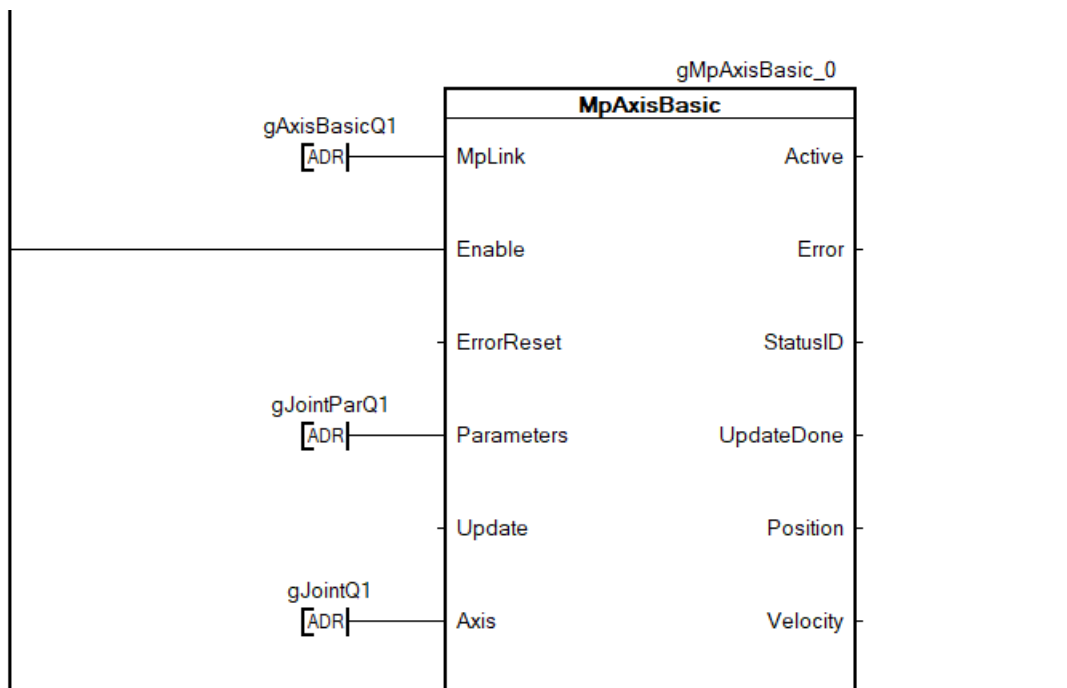
Ladder diagram je grafický programovací jazyk dle normy IEC 61131-3. Má podobnosti s elektrickými obvody. Je určený pro programování automatických systémů a je jedním z nejstarších a nejvyužívanějších programovacích jazyků. Nabízí jednoduché a rychlé programování, použití analogových a digitálních vstupů/výstupů, funkčních bloků, nástroje pro diagnostiku, přerušení a skákání v programu. Já jsem ho využil v této práci na definici mapp komponent v tasku. Hlavní důvod byl v jeho jednoduchosti a rychlosti vytvoření. Co je to task, je popsáno níže v sekci Task.

2.3 Vytvořené tasky v projektu

Tasky jsou vytvořené ve dvou jazycích. Task RobotCtrl je napsán v Ladder diagramu. Task ErrorCheck je napsán ve strukturovaném textu. Oba tasky obsahuje projekt přiložený v příloze. Níže popisují jejich stručný popis.

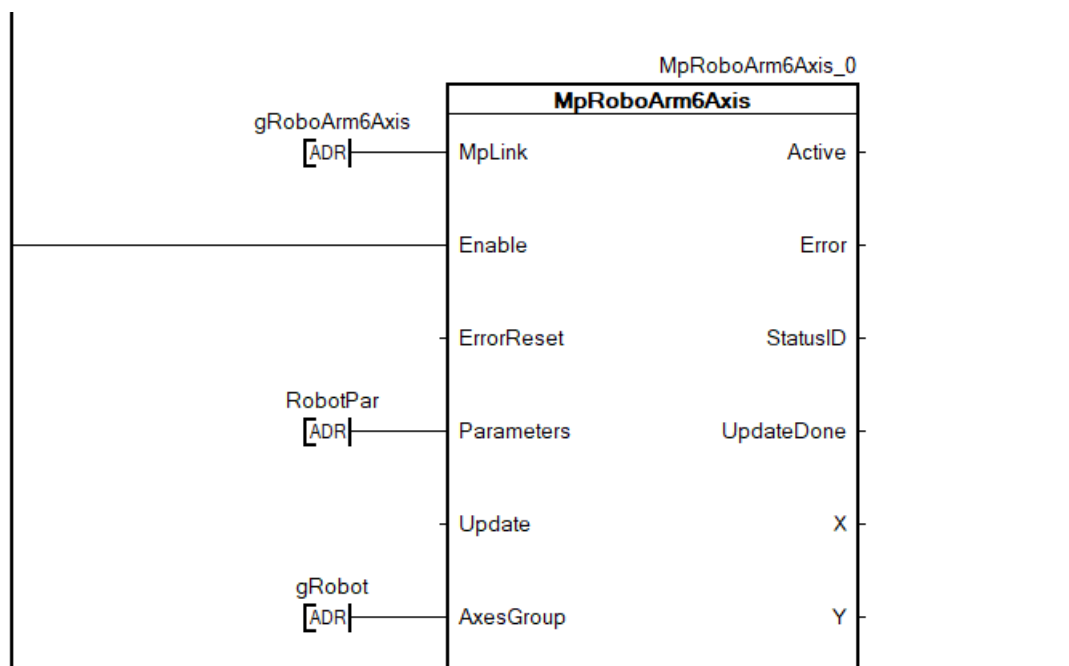
2.3.1 RobotCtrl

Na Obr. 10 a Obr. 11 nevidíme celé funkční bloky, ale jen její vrchní potřebnou část. Zbylé vstupy a výstupy se po přidání nijak neupravovaly a nejsou tedy důležité. Funkční blok MpAxisBasic je v tasku obsažen 6x, a to pro každou osu zvlášť, ale já se budu věnovat pouze jednomu, protože u ostatních je to stejné, jen pro jinou osu. MpRoboArm6Axis stačí využít jednou, protože díky němu se dá ovládat všech 6 os najednou. Celý task je tedy jen 7 funkčních bloků, které mají přiřazené nějaké vstupy popsané níže. Abyste mohli pracovat s funkčními bloky, bylo nutné u každého bloku zapnout enable (napájení).



Obr. 10 – Funkční blok MpAxisBasic v Ladder diagramu

Dále musíme u funkčního bloku MpAxisBasic viz Obr. 10 přiřadit ke vstupu Axis adresový vstup vybrané osy. Na vstup MpLink je přiřazena cesta k dané mapp komponentě, v tomto případě nazvané gAxisBasicQ1. Aby funkční blok měl nahané stejné parametry jako vybraná osa, je ke vstupu Parameters přiřazena cesta k parametrům dané osy. Toto je nutné udělat pro všech 6 funkčních bloků MpAxisBasic.



Obr. 11 – Funkční blok MpRoboArm6Axis v Ladder diagramu

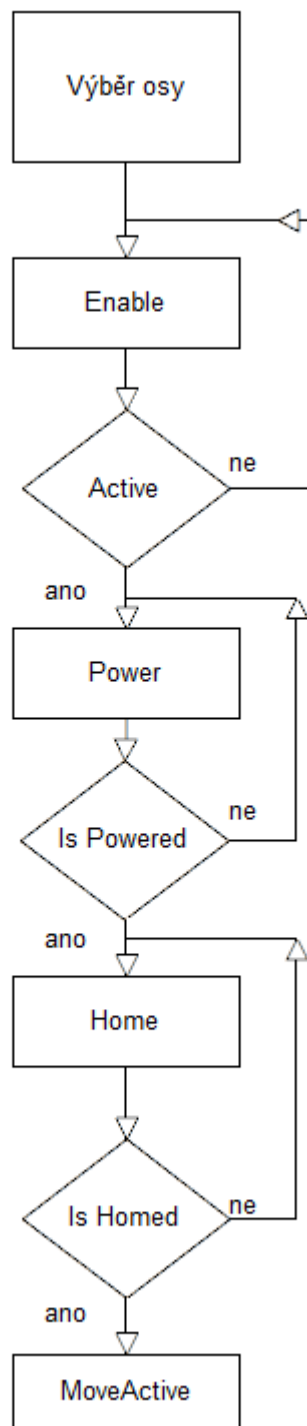
Na Obr. 11 vidíme funkční blok pro Mapp komponentu MpRoboArm6Axis. Přiřazení ke vstupům je velmi podobné jako u MpAxisBasic. K MpLink je přiřazena cesta k mapp komponentě, opět jsou ke vstupu Parameters přiřazeny parametry a také je nutné přiřadit dané osy ke vstupu AxesGroup.

Vývojové diagramy

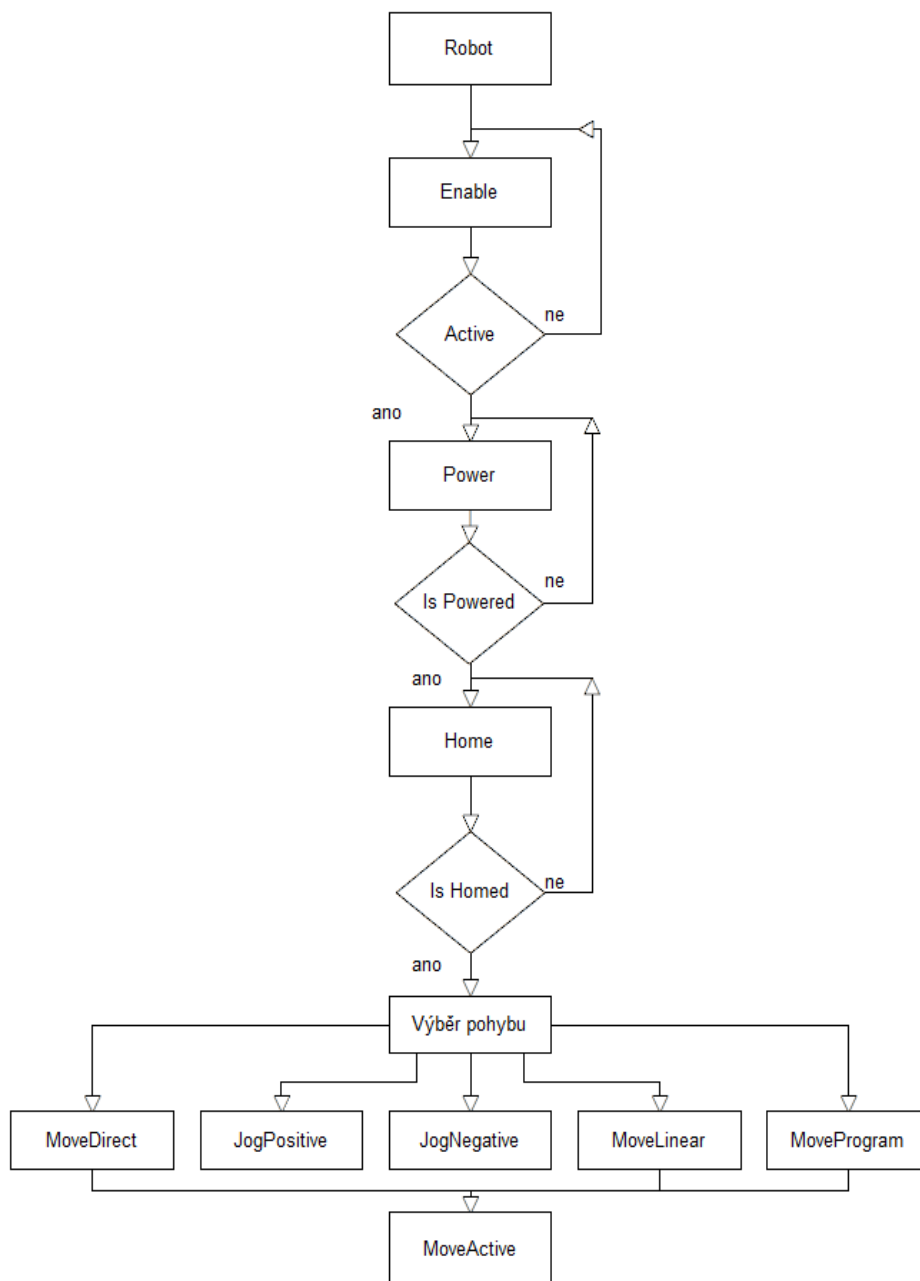
Vývojový diagram je grafické znázornění chování systému popsaného v krocích. Jsou tu znázorněné a popsané 2 diagramy. První je pro pohyb jedné osy. Druhý je pro pohyb robota jako celek, kdy se můžou hýbat buď všechny osy, anebo jen některé. Uvedené postupy jsou možné, jen pokud nevznikne žádná chyba v kterémkoli kroku. Postup při chybě je popsán v sekci Error. Samozřejmě je možné zastavit pohyb robota pomocí vstupu Stop, který je na obou mapp komponentách. Pokud je robot zastaven vstupem Stop je výstup Stopped nastaven na jedničku. Níže se věnuji oběma zmíněným diagramům.

Na Obr. 12 vidíme chování systému, když chceme pohnout osou robota. První krok je vybrat patřičnou osu, kterou chceme pohnout. Poté je nutné zapnout Enable, což je přivedení napájení ke vstupu funkčního bloku u mapp komponenty MpAxisBasic pro danou osu. Pokud je výstup Active jednička, můžeme zapnout střídač na kontroléru dané osy pomocí vstupu Power. Pokud je výstup PowerOn jednička je zapnut kontrolér. Dalším krokem je uvedení osy do základní polohy. To umožňuje vstup Home. Pokud je tedy výstup IsHomed jednička, už nám zbývá jediné a to vybrat daný pohyb. Zda se osa pohybuje, značí výstup MoveActive.

Postup pro pohyb robota je podobný jako pro pohyb jedné osy viz Obr. 13. Pro pohyb robota slouží mapp komponenta MpRoboArm6Axis. Rozdíl je například možnost předem vytvořený program nebo spouštět pohyb postupně po jednotlivých krocích, pomocí vstupu SingleStep. Pokud se robot pohybuje je opět výstup MoveActive v jedničce.



Obr. 12 – Vývojový diagram pro pohyb jedné osy



Obr. 13 – Vývojový diagram pro pohyb robota

2.3.2 ErrorCheck

Tento task kontroluje stavy robota, zastavení veškerých pohybů při vzniklé chybě nebo po stisknutí stop tlačítka. Kontroluje napájení robota, dokončení pohybu robota a resetuje spuštěný program, pokud je robot v chybovém stavu. Například na Obr. 14 je kus kódu, který zaručuje kontrolu zastavení všech možných typů pohybů robota, pokud se nachází v chybovém stavu. To znamená, pokud se nachází funkční blok MpRoboArm6Axis v chybě, tak se robot zastaví, a to při jakémkoli pohybu.

```
//control error
IF (MpRoboArm6Axis_0.Error) THEN
    MpRoboArm6Axis_0.Continue := 0;
    MpRoboArm6Axis_0.MoveProgram := 0;
    MpRoboArm6Axis_0.MoveDirect := 0;
    MpRoboArm6Axis_0.MoveLinear := 0;
    MpRoboArm6Axis_0.JogNegative := 0;
    MpRoboArm6Axis_0.JogPositive := 0;
    MpRoboArm6Axis_0.Interrupt := 0;
    MoveProgram := 0;
END_IF
```

Obr. 14 – Control error

Task obsahuje také kódy, které přímo nesouvisí s mojí prací, ale v celé aplikaci jsou nutné pro správnou funkci s displejem. Tyto kódy psal kolega, který měl za úkol vizualizaci na displeji.

2.3.3 Error

Pokud nastane chyba, je nutné chybu potvrdit a pokračovat v daném kroku znova. Uvedu stručný příklad. Chceme například hýbat jednou osou v pozitivním směru. Výstupy dané Mapp komponenty MpAxisBasic jsou jednička a to výstupy Active, PowerOn a IsHomed. Zvolíme pohyb v pozitivním směru, ale robot se nám dostane do chyby. Musíme tedy si přečíst v Automation studiu nebo na displeji o jakou chybu se jedná a snažit se ji vyřešit. Vzniklá chyba může být například dojetí osy do jejího limitu. To znamená, že robot už daným směrem nemůže pokračovat například z mechanického hlediska. Pokud ji máme vyřešenou, je nutné chybu potvrdit pomocí vstupu ErrorReset. Poté je možné opět zvolit pohyb v pozitivním směru. Tímto je problém vyřešen a daná osa se nám pohybuje v pozitivním směru. Pokud nastane

znovu chyba, je možné že jste patřičnou chybu nevyřešili správně a je nutné se jí znovu zabývat.

2.4 Nahrané programy v APC910

Programy lze nahrát do PLC pomocí FTP protokolu. FTP (File Transfer Protokol) je dle [19] protokol, pro přenos souborů mezi počítači pomocí počítačové sítě. Je nezávislý na typu operačního systému. V PLC jsou nahrané 3 programy a jsou ve dvou jazycích a to v jazyku ST a v G-kódu. Důvodem bylo vyzkoušení si psaní v obou jazycích. G- kód nicméně není vhodný pro programování robota, ale je vyvinut na programování CNC strojů.

2.4.1 Transport position

```
%001 (CncProgram)

N001 G126 50
N100 G100 Q5 -119 Q4 90
N200 G100 Q3 175
N300 G100 Q1 -90
N400 G100 Q2 -88

N9998 G126 0
N9999 M30
```

Obr. 15 – Transport position

Program pomocí, kterého dostaneme robota do transportní pozice. Je to poloha robota, kdy se složí do co nejmenší pozice vhodné pro jeho transport. Program je psaný v G-kódu viz Obr. 15.

Číslo N značí číslování řádků. Program plyne od nejmenšího čísla k největšímu. Číslování řádků není úplně postupné. Je to kvůli možnosti dopsání dalších funkcí do programu, bez nutnosti přepisování řádků u funkcí.

- G126 - G – funkce, která zaobluje spojení dvou přímek. Uvedené číslo je radius zaoblání. Funkce platí, dokud není anulována, viz řádek N9998
- G100 - G – funkce, pro PTP (point to point) pohyb (pohyb, kdy osa/y se pohnou z aktuálního bodu do zadaného bodu, v co nejkratším čase)

- M30 - M – funkce, která ukončuje program

2.4.2 Home position

```
VAR CONSTANT
  StartPos : MXAxisTargetType := (JointAxis:=[0, 10, -10, 0, 70, 0]);
END_VAR
PROGRAM _MAIN
  MoveA(StartPos, Vel:=MX_VEL_F10000);
END_PROGRAM
```

Obr. 16 – Home position

Program, je na uvedení robota do výchozí pozice viz Obr. 8. Program vidíme na Obr. 16. Je psaný v jazyce ST. Na druhém řádku programu, je vytvořená proměnná StartPos typu MXAcisTargetType, který umožňuje pohyb se všemi osami robota najednou. Používá se s příkazem MoveA. V hranatých závorkách jsou uvedeny pozice, na které se osy mají pohnout. V příkazu MoveA je uvedená vybraná proměnná a předem nadefinovaná rychlost pohybu.

2.4.3 Racer Dance

Program je psaný v G-kódu. Jelikož je program dlouhý, tak je přiložen v příloze spolu s videem, na kterém je tento program spuštěný. Program je vhodný pro odzkoušení pohybů všech os robota. Je důležité zjistit, zda je robot dostatečně pevně přidělán na svém podstavci a to při jakékoli možné rychlosti, která je dána limitem rychlostí jednotlivých motorů v robotu, což tento program splňuje. V programu jsou použity stejné funkce jako u transportní pozice a to G126, G100 a M30, Proto není potřeba další komentář.

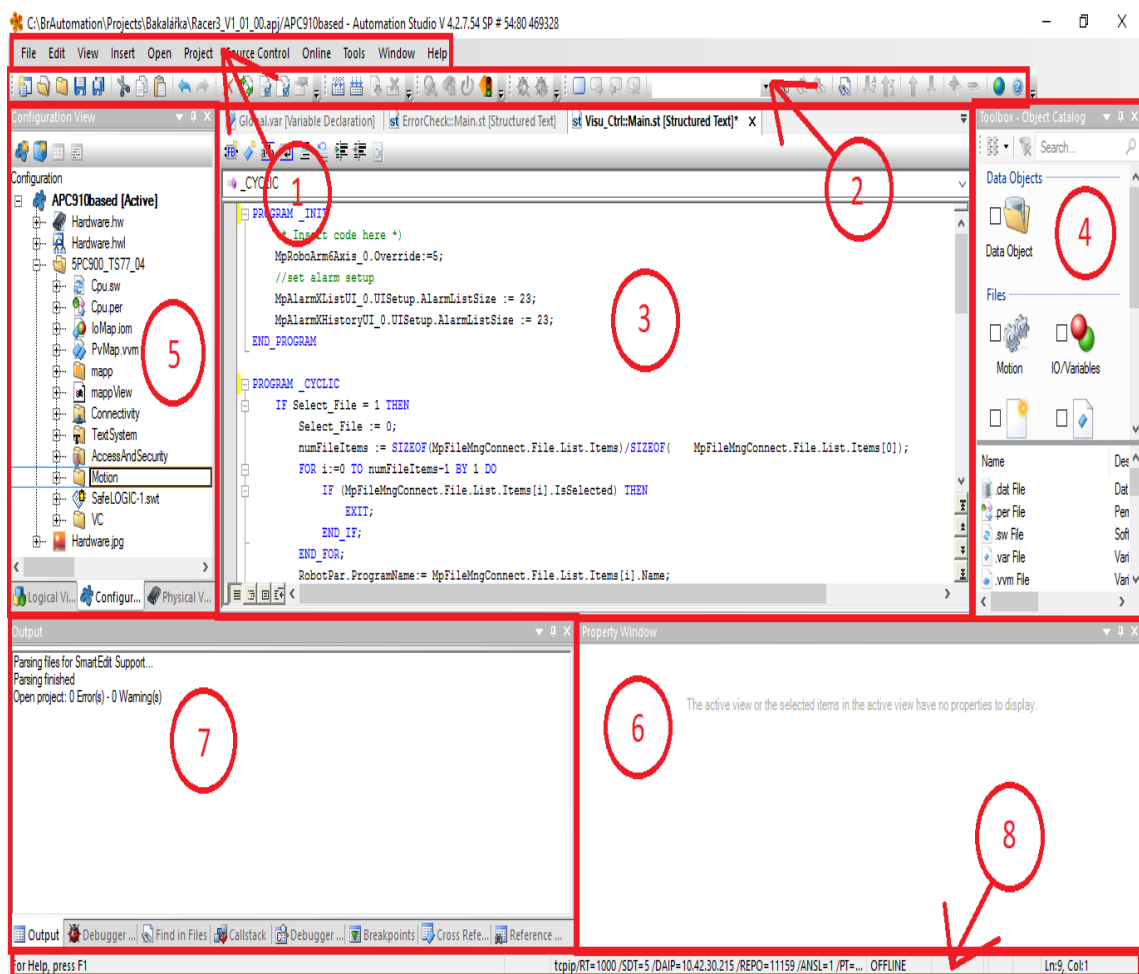
3 Vývojové prostředí

Jedná se o vývojové prostředí Automation studio od firmy B&R. Umožňuje vytváření různých průmyslových aplikací v oblasti strojího inženýrství, jsou to například aplikace s využitím CNC strojů nebo robotů. Je zde možnost vytvářet programy obsahující samotný algoritmus k aplikaci a to v různých programovacích jazycích i v jednom projektu, vyzkoušet si aplikaci díky simulaci a nastavovat parametry k aplikaci. Jeho výhody jsou hlavně rychlost a jednoduchost při vytváření aplikací.

Veškeré možnosti prostředí jsou podrobně popsány v helpu Automation studia. To nejdůležitější je popsáno níže dle [3].

3.1 Hlavní části

Vývojové prostředí lze rozdělit na 8 hlavních částí, které vidíme na Obr. 17. Jedná se o Main menu, Toolbar, Toolbox, Properties, Program workspace, Project explorer, Status bar a Output window.



Obr. 17 – Hlavní části Automation studia

1) Main menu

Tato část má mnoho využití. Zde uvedu jen ty, co si myslím, že jsou nejdůležitější a základní. Používá se pro ukládání, kopírování, otevírání, vytváření projektů a celkové nastavení zobrazení prostředí. Můžeme zde otevřít help a najít informace o

prostředí. Je to například verze vývojového prostředí či licence. Pro otevření některých oken se dá jako v mnoha prostředích či programech použít klávesové zkratky, které nám urychlují práci.

2) Toolbar

Slouží v podstatě jen pro urychlení aplikací. Veškeré možnosti v tomto řádku jsou také v hlavním menu prostředí.

3) Program workspace

Je to v podstatě jakási plocha, která slouží pro samotné vytváření a upravování otevřených programů či konfigurací. Mezi otevřenými okny lze vybírat pomocí horní lišty na této ploše. Podle typu otevřeného okna se nám též zobrazují jeho nástroje. Například v okně pro vytváření strukturovaného textu je v nástrojích přidání funkčního bloku, proměnných, ale také možnost využití komentáře v programu.

4) Toolbox

Zde můžeme přidávat veškeré programy nabízené ve vybraných jazycích, jako jsou Strukturovaný text, Ladder diagram či jazyk C. Zde jsem uvedl jen tři z mnoha nabízených typů programovacích jazyků. Dalšími možnostmi jsou přidání knihoven, složek, grafů, veškerých tabulek pro motion, poznámky, vizualizace, objekty, mapp komponenty, všechny komponenty hardwarové konfigurace, funkční bloky nebo také příklady knihoven a motion programů. Těch možností je spousta a já zde vybral ty nejdůležitější a nutné pro uvedenou aplikaci. Nabízené možnosti se mění vzhledem k tomu, na jakou záložku a typ položky máte vybraný v Project exploreru. Pokud jste na záložce Physical View, tak se vám budou nabízet hardwarové prvky. Po přidání je pak uvidíte v samotné části Project exploreru. V panelu je i možnost vyhledávání podle označení hardwarového prvku, které používá firma B&R. Tomuto označení se zde věnovat nebudu.

5) Project explorer

Díky této části prostředí můžeme otvírat a zavírat příslušná okna, která se nám zobrazí na pracovní ploše. Zde na všech záložkách vidíte, co vlastně ten příslušný

projekt obsahuje. Mohou to být veškeré vámi vytvořené programy, složky obsahující tabulky nutné pro správný chod aplikace, hardwarové a softwarové konfigurace a vizualizace.

Obsahuje 3 záložky a to Logical View, Configuration View a Physical View. Na každé z nich můžete využívat jiné možnosti. Ty popíšete v následujícím textu.

- Logical View – Zde vidíme složku daného projektu a najdeme vše, co tento projekt obsahuje jako například veškeré tasky, knihovny, globální proměnné a datové typy, lokální proměnné a datové typy. Může zde být i textový soubor pro nějaké poznámky, vizualizace a přidané složky obsahující tabulky například pro přidaný CNC systém
- Configuration View – Uvedl bych jen to nejdůležitější, co zde můžeme najít. Jedná se o vybranou konfiguraci a její složky obsahující například všechny mapp tabulky nebo software pro safety. Můžeme zde také přidat hardwarovou konfiguraci. To znamená, že v projektu může být víc konfigurací, ale jen k jedné můžeme být připojeni. Připojení konfigurace děláme pomocí dvojkliku na danou konfiguraci
- Physical View – najdeme tu tabulku, kde vidíme veškeré komponenty hardwaru. Pro uvedenou aplikaci například APC910, ACOPOS P3, motory, displej, jednotlivé sloty APC910. Vidíme zde i co je k čemu připojené, jaké sloty má daná komponenta. Můžeme zde také přidat nebo nahradit hardwarový modul, otevřít System Designer, ve kterém se zobrazuje graficky naše hardwarová konfigurace. Lze také otevřít konfiguraci vybraného hardwarového prvku

6) Properties

Okno nám zobrazuje veškeré vlastnosti námi vybraného prvku hardwaru.

7) Output window

V tomto okně je více záložek, ale ta nejdůležitější je záložka Output. Zde se zobrazují veškeré vzniklé chyby a varování při kompilování programu. Pokud nastane nějaká chyba, tak se zobrazí červeně. Varování jsou zelené barvy. Samotné barevné rozdíly slouží pro zrychlení hledání a rozlišení chyb, varování nebo úspěšné kompilaci. Více o kompilaci je v sekci Postup vytvoření projektu v AS.

8) Status bar

To nejdůležitější zde, je v jakém operačním stavu Runtimu se zrovna nacházíme, jeho verzi, a k jaké IP adrese jsme připojeni.

3.2 Diagnostické nástroje

Díky těmto nástrojům lokalizujeme problémy a chyby vzniklé v projektu. Ladíme celý chod aplikace a testujeme jednotlivé osy, aby fungovala pokud možno bez chyb a co nejrychleji.

Je možnost také porovnat software, hardware a Automation komponenty. Uvedu příklad. Přijdete k lince na výrobu krabic. Aplikaci jste nezprovoznili a nemáte k němu ani projekt. Musíte si založit nový projekt a vytvořit hardwarovou konfiguraci. Pokud jsme připojeni k PLC, jsme v režimu RUN a zvolíte porovnání hardwaru, tak vám vyskočí dvě okna. V prvním okně se nám zobrazí hardwarové komponenty, které jsme si přidali do hardwarové konfigurace v našem projektu. Ve druhém jsou naopak komponenty, které jsou obsaženy fyzicky v dané aplikaci. A tímto nástrojem je můžeme porovnat. Zjistíte, jaké komponenty vám chybí, jejich označení i na jaké adrese jsou zapojené k PLC.

Watch

Pomocí tohoto diagnostického nástroje můžeme online odzkoušet, sledovat a ladit aplikaci nebo funkci vybraného programu či funkčního bloku, aby fungovala tak jak potřebujeme. Je zde možnost přiřazovat proměnným jejich hodnoty. Spustit lze nad celou aplikací i nad samostatnými tasky v aplikaci, podle toho se nám nabízí možnost přidání proměnných. Do okna si můžeme přidat veškeré funkční bloky a proměnné jakéhokoli typu. Výhodou je, že si můžeme přidat pouze proměnné, které chceme odzkoušet a vybrané proměnné z daného funkčního bloku nebo mapp komponenty.

Trace

Trace lze využít pro zaznamenávání hodnot zvolených proměnných v čase. Hodnoty proměnných se v pravidelných intervalech čtou a zapisují do paměti. Data se dají uložit v několika formátech, například i jako tabulku v Excelu.

Test

The screenshot displays the gAxis03.mdc software interface with four numbered callouts:

- 1**: Command Interface tree, showing a sequence of steps like Preparation, Digital Inputs, Controller, Initialize, Switch on, Switch off, Homing, Basis Movement, Additive, Positive, Negative, Stop, Limits, Service Interface, and Motor Simulation. The 'Exclusive mode' is highlighted at the bottom.
- 2**: Structure table showing parameters for the axis. The 's' parameter is highlighted with a value of 1000.
- 3**: Watch table showing the current state of the PLCopen system, including parameters like network.init, network.service.data_text, network.service.request_par_id, network.service.response_par_id, monitor.s, monitor.v, message.count.error, message.count.warning, message.record.par_id, message.record.number, and message.record.info.
- 4**: Trace window showing two graphs. The top graph is titled 'gAxis03 ... CTRL Position controller: Set speed, 10/28/11, 13:31:51' and shows velocity (Units/s) over time. The bottom graph is titled 'gAxis03 ... CTRL Position controller: Actual position, 10/28/11, 13:31:51' and shows position (Units) over time.

Obr. 18 – Struktura Test okna

Test obsahuje 4 okna, které vidíme na Obr. 18.

- 1) Příkazové okno
- 2) Okno s parametry osy
- 3) Watch
- 4) Trace

V testu se dá nasimulovat a graficky znázornit například rozjždění motoru, které zde stručně a krátce popíši. Nejdříve je nutné nastavit parametry osy, například rychlost, limity osy nebo délku pohybu. Poté se musí v příkazovém okně nahodit napájení kontroléru a použít příkaz homing. Po provedení těchto dvou příkazů,

můžeme konečně vybrat příkaz pro pohyb. Je zde klasická nabídka pohybů například absolutní, pozitivní a negativní. Stav osy si můžeme zkontrolovat v okně Watch. Zde také vidíme, zda vznikla nějaká chyba například překročení limitu osy. Pokud chceme zobrazit nějaký parametr v tracu musíme v příkazovém okně zaškrtnout Trace enabled.

Logger

Logger je diagnostický nástroj pro zobrazení chyb při samotném běhu aplikace nebo při kompilaci. Pro vyřešení dané chyby je nejvýhodnější, protože obsahuje i historii chyb a podrobnější popis chyb.

SDM (Systém Diagnostic Manager)

Využívá se k diagnostice řadiče pomocí webového prohlížeče. Je nutné mít připojení k cílovému systému.

3.3 Task

Task je algoritmus, který nejčastěji řeší jeden úkol v aplikaci. Je možné ho psát ve více programovacích jazycích podle vaší potřeby (Strukturovaný text, Ladder diagram, Functional block diagram, Instruction list, Sequential function chart, Continues function chart atd.). Může řešit i celou aplikaci, ale u těch složitějších to není vhodné kvůli přehlednosti programu.

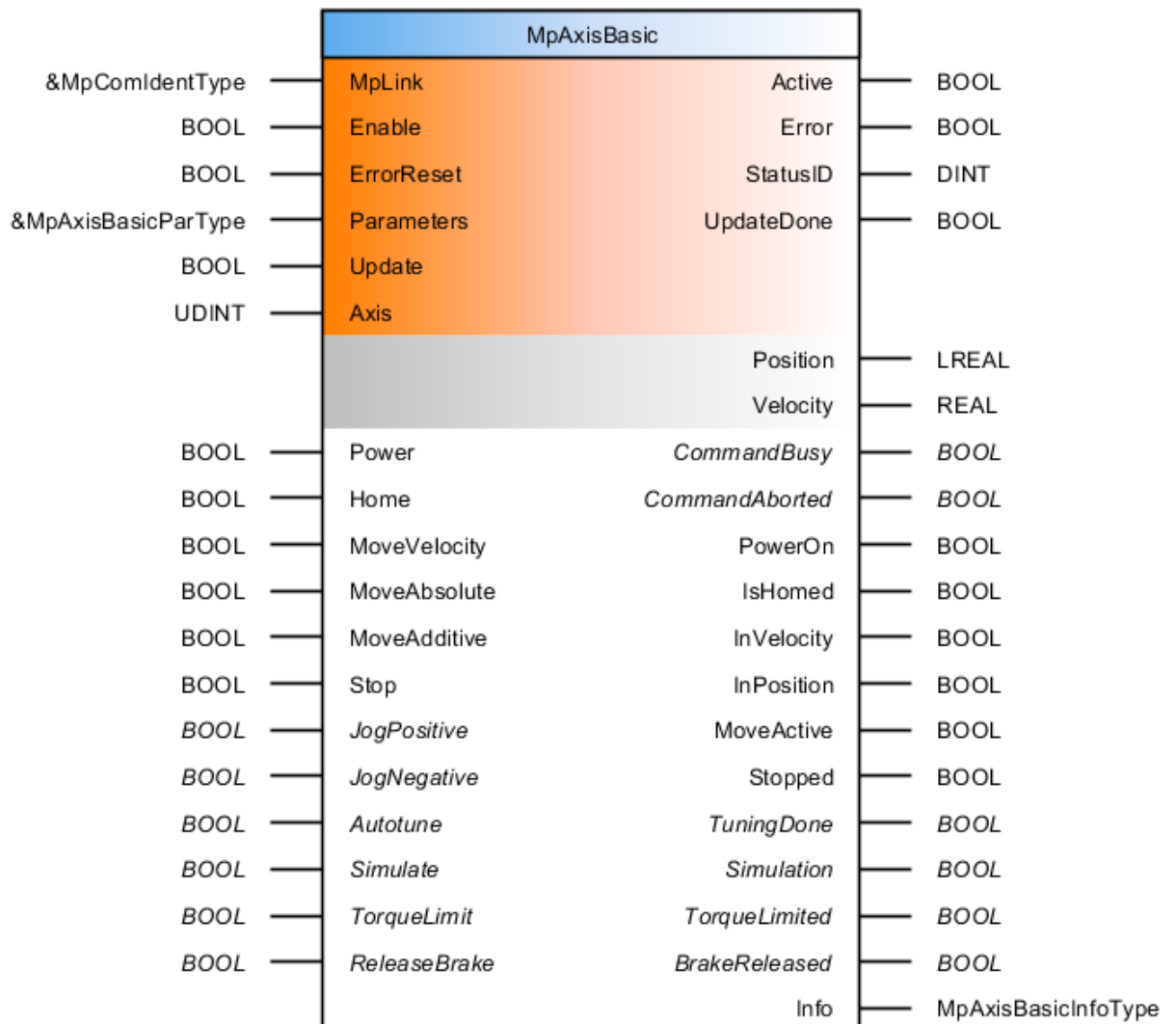
4 Mapp Technologie

Mapp technologie je dle [17] revoluce softwaru od společnosti B&R. Díky mapp technologii je vývoj aplikací snadnější a hlavně rychlejší až o 67 %. Zvyšuje spolehlivost strojů a snižuje nároky na údržbu. Vývojáři díky ní nemusí softwarově řešit základní funkce, které se často opakují, jako jsou například: ovládání napájení kontroléru, chybové hlášení, ukládání parametrů a receptur.

Mapp technologie obsahuje Mapp komponenty. Jsou to v podstatě funkční bloky, které mají nějaké specifické vstupy a výstupy. Pro tuto práci bylo nutné využít 2 mapp komponenty a to MpAxisBasic a MpRoboArm6Axis. U obou mapp komponent rozvedu její použití a uvedu některé vstupy a výstupy. Podrobnější popis všech

vstupů a výstupů je přiložen na CD. Přiložený dokument se jmenuje Vstupy_a_výstupy_Mapp_komponent. Po přidání mapp komponent do jakéhokoli projektu se vytvoří mappovací tabulka, kde je nutné vyplnit informace o této komponentě, jako jsou například nasměrování daných os pro předváděnou komponentu a nastavení limitů. Po přidání mapp komponenty, je nutné si vytvořit proměnnou daného typu mapp komponentu. Pro mapp komponentu MpAxisBasic se vytvoří proměnná typu MpAxisBasic. Samotný funkční blok je pak nutný deklarovat v programu a přesměrovat na danou mapp tabulku.

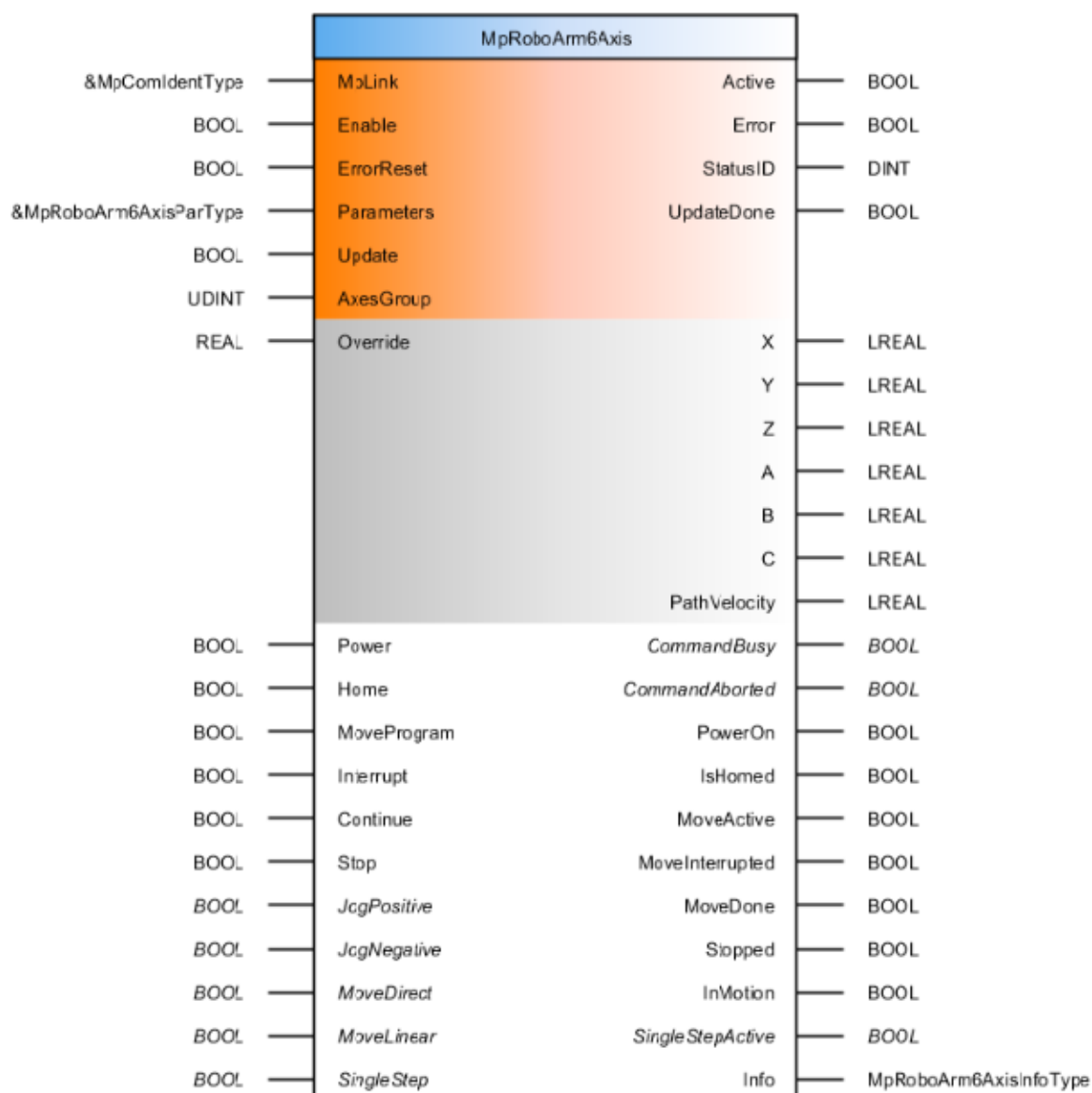
4.1 MpAxisBasic



Obr. 19 – MpAxisBasic [3]

Mapp komponenta MpAxisBasic podle [3] pomáhá s ovládáním základních funkcí jedné určité osy. Může to být například zapnutí/vypnutí funkčního bloku, ovládání napájení pro střídač motoru, nastavení veškerých parametrů jako rychlosti, pozice a jiné. Pro robot Racer 3, který má 6 os, jsem tuto komponentu použil pro každou osu zvlášť, tudíž těchto mapp komponent je v programu šest. Na Obr. 19 vidíme mapp komponentu i s jejími vstupy a výstupy.

4.2 MpRoboArm6Axis



Obr. 20 – MpRoboArm6Axis [3]

Mapp komponenta díky které, můžeme dle [3] jednoduše a rychle ovládat základní funkce celého 6-ti osého robota. Na rozdíl od mapp komponenty MpAxisBasic, zde

nemůžeme ovládat pouze jednu osu, ale buď všechny osy, anebo osy se kterými je potřeba pohnout při pohybu do daného bodu v určitých souřadnicích. Dá se zde také nahrát předem připravený program, který mapp komponenta vykoná. Program se dá například krokovat nebo zastavit. Na Obr. 20 je uvedena celá mapp komponenta se vstupy na levé straně a výstupy na straně pravé.

5 Postup vytvoření projektu v AS

V této kapitole popíši stručný postup vytvoření projektu pro aplikaci ovládání základních pohybů robota RACER 3. Je velmi důležité se seznámit nejen s vývojovým prostředím, programovacími jazyky, robotem, ale také se samotným řídicím systémem pro robota.

První věc byla založit projekt v Automation Studiu (verze 4.2.7.54), kde se na horní liště klikne na File a poté New Project. Na první stránce bylo nutné zadat název projektu a cestu, kam se má projekt ukládat. Je možnost také napsat popis projektu. Klikneme na Next. Na této stránce určujeme název a popis hardwarové konfigurace. Je nutné zaškrtnout, jakou chceme možnost vytvoření hardwaru, můžeme si vybrat mezi vytvořením nové konfigurace, použitím konfigurace z jiného projektu, anebo najít danou konfiguraci pomocí internetu. Pro tento projekt jsem zvolil vytvoření nové konfigurace. Dalšími stránkami vybíráme CPU pro daný projekt. Po přidání vhodného CPU, se nám vytvoří projekt, kde je v hardwarové konfiguraci dané CPU. Do hardwarové konfigurace k CPU, jsem přidal ještě powerlinkovou kartu, ke které jsem zapojil dva servozsilovače ACOPOS P3 a safety moduly. K CPU se ještě přidala karta pro komunikaci s displejem přes ethernet. Bylo nutné vytvořit vizualizaci pro správnou funkci ovládání na displeji. Samotná vizualizace vyžaduje přidání nových mapp komponent, programů, atd. Tímto tématem se však zabývat nebudu, jelikož to nebylo náplní mé práce. Protože jsou tato témata ale navzájem propojená, tak v projektu, který je v příloženém CD se nachází i mapp komponenty a tasky pro vizualizaci. Výslednou grafickou hardwarovou konfiguraci v Automation studiu vidíme na Obr. 1.

Dále je nutné zajistit komunikaci mezi uživatelským CPU a CPU pro robota, abych mohl ovládat robota přes AS. K tomu jsem využil ethernetový kabel. Na uživatelském počítači jsem nastavil IP adresu 10.42.30.214 a masku 255.255.255.0. Pro nastavení IP adresy pro PC robota je nutné ve Physical view kliknout pravým tlačítkem myši na ETH1 nebo ETH2, vybrat Configuration a zde nastavit mode na enter IP Address Manually a poté přepsat IP adresu na 10.42.30.215 a masku na 255.255.255.0. Mezi ETH1 a ETH2 vybíráte ten, kde ho fyzicky máte v APC910 zapojený. Pro tuto práci je vybraný port ETH1.

Pro ovládání robota je nutné přidat výše popsané Mapp komponenty a vytvořit tasky pro ovládání robota. Popis dvou použitých tasků je v sekci Vytvořené tasky v projektu. Po přidání Mapp komponent, se k nim automaticky vytvoří Mapp tabulka. V tabulkách se nastavují parametry dané Mapp komponenty. Jedná se například o maximální rychlosti nebo maximální moment hybnosti v daných osách.

Důležitou částí projektu je také vyplnění Initních a parametrických tabulek, které se samy vytvoří po přidání servozesilovačů. Initní tabulka definuje základní parametry použité při inicializaci servozesilovače, jako jsou hodnoty nastavení regulátorů, výchozí hodnoty rychlosti a zrychlení pohybů ve všech typech pohybů. Tato tabulka se nainicializuje po zapnutí kontroléru dané osy. Parametrická tabulka umožňuje přímou adresaci všech parametrů servozesilovače. Každý jednotlivý parametr je pojmenován svým unikátním číslem ID, podle kterého je možné ho identifikovat a měnit jeho hodnotu.

Aby uživatelské programy, knihovny, datové objekty a konfigurace mohli být použity v cílovém systému, jsou pomocí tlačítka build zkompileovány na takzvané B&R moduly dle [3]. Kompilace také zjišťuje provedené změny v aktivní konfiguraci od posledního nahrání do cílového systému a zobrazí vám chyby, například při nesprávné syntaxi psaní v tasku. Když vznikne při kompilaci chyba, není možné nahrát program do PLC. Naopak při varování lze program nahrát do PLC. Je nutné pak veškeré chyby a nejlépe i varování pro správný chod aplikace odstranit. Pokud už jste připojeni k cílovému PLC, tak je nutné tam nahrát program pomocí tlačítka transfer, které vám vyskočí po úspěšné kompilaci.

5.1 Knihovny

Do projektu jsou automaticky vloženy nejdůležitější knihovny. Po přidání veškerých hardwarových prvků se přidají k nim potřebné knihovny. To platí také pro mapp komponenty. Pro tento projekt bylo nutné aktualizovat knihovnu TRF-LIB a přidat knihovnu Arnc0ext. Knihovna TRF_LIB obsahuje potřebné proměnné, funkční bloky a datové typy pro transformaci souřadnic u různých kinematických typů robota. Knihovna Arnc0ext poskytuje dodatečné funkce pro interpolaci více než 4 os.

5.2 Knihovna MpRoboX

Knihovna MpRoboX je univerzální knihovna, dle [3] slouží k ovládání robota s různými kinematickými typy, lze tedy s robotem jogovat, polohovat na určitou souřadnici nebo spustit robotický program. Podporované kinematické typy jsou popsány v knihovně TRF_LIB. Knihovna obsahuje také různé struktury, datové typy a funkční bloky speciálně vytvořené pro mapp komponentu MpRoboXFlex.

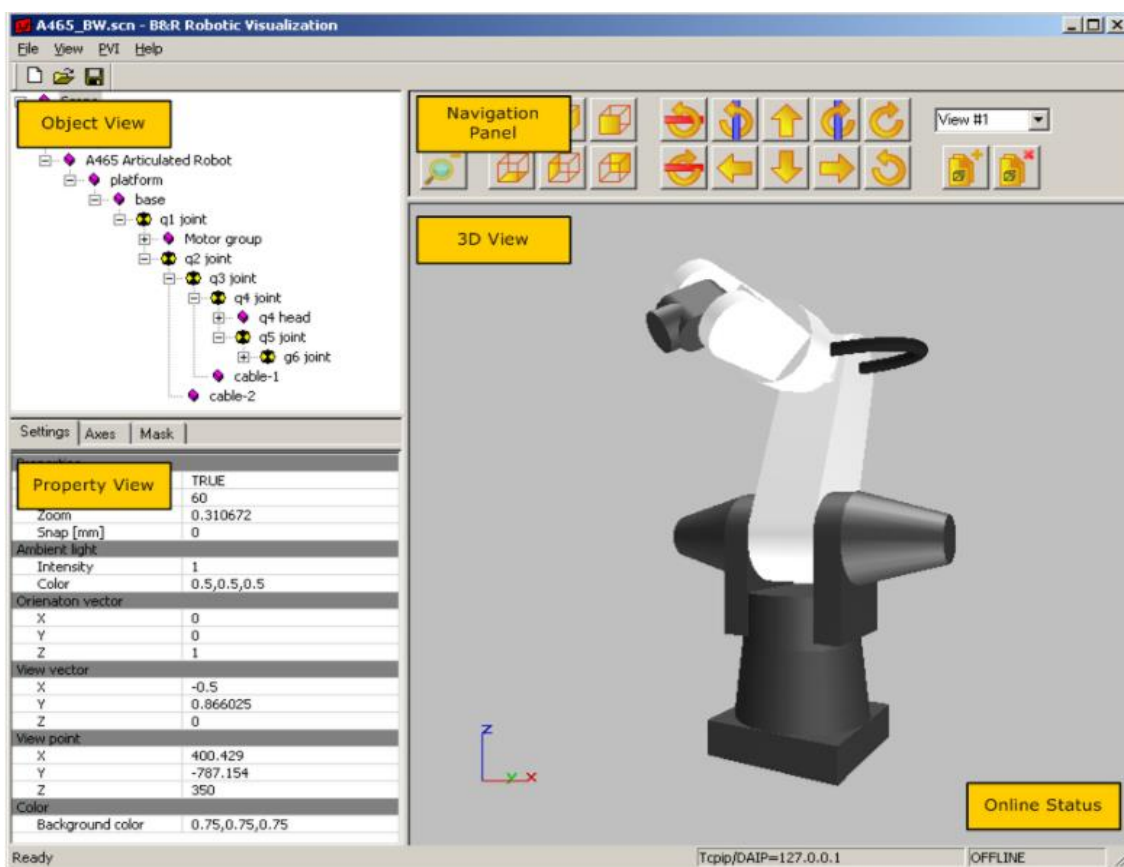
5.3 Ladění aplikace

K ladění projektu a jeho správnému chodu bez cílového systému jsem využil diagnostické nástroje watch, test a také program B&R Robotic Visualization od firmy B&R, který je popsán níže v sekci B&R Robotic Visualization. Ladění probíhalo také na místě s robotem pomocí nástroje watch a použitého ovladače (displeje).

6 B&R Robotic Visualization

Dle [4] to je program pro 3D simulování pohybů robota od firmy B&R. Je využitelný v případě, že nemáte robota fyzicky u sebe. Tím pádem je možné nasimulovat celou vytvořenou aplikaci a vyzkoušet, zda funguje vše správně. Je jasné, že když máte robot fyzicky u sebe, je to vždy nejlepší, ale jako náhrada robota, tento program postačuje. Já jsem tento nástroj využil v době, kdy jsem neměl robota fyzicky u sebe a potřeboval jsem zjistit, jak se bude robot chovat.

Z Obr. 21 vidíme, že se tento program dá rozdělit na pět částí a to Object View, Property View, 3D View, Navigation Panel a Online Status. Property View obsahuje tři záložky Settings, Axes, Mask. V záložce Settings je možné změnit pozici nebo rotaci robota na scéně, povolit zobrazení TCP souřadného systému, poslat robota do výchozí polohy či změnit jeho barvu. V Axes je možné vyzkoušet pohyby robota v reálných a TCP souřadnicích, a to jen pomocí samotného programu, není nutné být připojen k tasku. Každý přidávaný objekt má svoji ParID masku, kterou nastavujeme ve třetí záložce Mask. V Navigation Panel lze nejen měnit pozici či rotovat robotem, ale také pohybovat a rotovat se samotným souřadným systémem, měnit pohled na 3D scénu a také ukládat, vybírat a odstraňovat již uložené pohledy. V poslední řadě je možné přibližovat a oddalovat 3D scénu. Object View slouží pro vytvoření, přidání, odebrání objektů nebo nástrojů a vytvoření souřadných systémů. Po nainstalování programu už je spousta objektů vytvořena, takže je stačí jen přidat na 3D scénu. Objekty, které lze přidat mohou být vytvořeny například v programu CAD. Ve 3D View se zobrazuje samotný vybraný robot a jeho pohyby. V části Online Status vidíme, zda jsme připojeni k vybranému tasku.



Obr. 21 – B&R Robotic visualization [4]

Závěr

Výsledek práce je takový, že vytvořený software pro robota, lze ovládat pomocí dotykového displeje, který slouží jako ovladač. Robota lze ovládat ve dvou souřadných systémech. První je TCP (Tool centre point) souřadný systém. Počátek systému je vztažen ke koncovému bodu nástroje robota, například chapadla. Druhý souřadný systém je WCS (World coordinate system) a jeho počátek je vztažen k pracovnímu prostoru robota. Vytvořený algoritmus umožňuje s robotem jogovat (na ovladači vyberete s jakou osou, chcete pohybovat a dokud držíte tlačítko, tak s ní můžete hýbat v záporném nebo kladném směru, ale pouze v jednom směru), polohovat na určitou souřadnici, pohybovat s každou osou zvlášť. Lze měnit také rychlost pohybu a pouštět nahrané programy z PC. V příloze je přidán celý projekt, několik fotografií a videí z ožívování robota, nahrané programy v PC a tato práce ve formátu pdf.

Aplikace nabízí rozmanité možnosti pokračování. Na přírubu robota lze namontovat chapadlo, pomocí kterého robot uchytí různé předměty. Po uchycení předmětu, například psací potřeby, může robot na tabuli nakreslit obrázek. Lze také přidat přísavku a pomocí stlačeného vzduchu přemísťovat různé předměty. Přidáním chapadla, či přísavky se samozřejmě musí upravit i software.

Použité zdroje

- [1] Pospíchal Zbyněk. 02. 06. 2008. Realizace aplikace pro ovládání servopohonu. In: vutbr.cz [online]. [cit. 03. 08. 2017]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_ve-rejne.php?file_id=9004
- [2] Programování IEC61131-3. In: edumat.cz [online]. [cit. 10. 08. 2017]. Dostupné z: http://www.edumat.cz/texty/Programovani_IEC61131-3.pdf
- [3] Help Automation studia
- [4] Help B&R Robotic Visualization
- [5] Automation PC910. 2017. In: br-automation.com [online]. [cit. 20. 05. 2017]. Dostupné z: <https://www.br-automation.com/cs/produkty/industrial-pcs/automation-pc-910/>
- [6] Simple directMedia Layer. 03. 01. 2017. In: wikipedia.org [online]. [cit. 20. 05. 2017]. Dostupné z: https://cs.wikipedia.org/wiki/Simple_DirectMedia_Layer
- [7] Digital Visual Interface. 11. 05. 2017. In: wikipedia.org [online]. [cit. 20. 05. 2017]. Dostupné z: https://cs.wikipedia.org/wiki/Digital_Visual_Interface
- [8] PCI express. 06. 12. 2016. In: wikipedia.org [online]. [cit. 20. 05. 2017]. Dostupné z: <https://cs.wikipedia.org/wiki/PCI-Express>
- [9] RYDLO, Pavel, Řízení elektrických střídavých pohonů. V Liberci: Technická univerzita, 2006. ISBN 80-7372-117-1.
- [10] Pc vs PLC. 14. 09. 2011. In: controlengcesko.com[online]. [cit. 20. 05. 2017]. Dostupné z: <http://www.controlengcesko.com/hlavni-menu/artykuly/artykul/article/pc-vs-plc-porovnani-variant-ridiciho-sytemu/>
- [11] Počítač. 29. 04. 2017. In: wikipedia.org [online]. [cit. 20. 05. 2017]. Dostupné z: <https://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8D>
- [12] Milan Bureš. 18. 09. 2003. Tabulka krytí IP. In: elektrika.cz [online]. [cit. 20. 05. 2017]. Dostupné z: <http://elektrika.cz/data/clanky/krip030918>

- [13] Racer 3. 2017. In: comau.com [online]. [cit. 20. 05. 2017]. Dostupné z: <http://www.comau.com/EN/our-competences/robotics/robot-team/racer-3>
- [14] IEC. 6. 12. 2016. In: wikipedia.org [online]. [cit. 20. 05. 2017]. Dostupné z: https://cs.wikipedia.org/wiki/Mezin%C3%A1rodn%C3%AD_elektrotechnick%C3%A1_komise
- [15] Smart Display Link 3. 2017. In: br-automation.com [online]. [cit. 20. 05. 2017]. Dostupné z: <https://www.br-automation.com/en/products/industrial-pcs/smart-display-link-3/>
- [16] Ladder diagramy – základy. 24. 03. 2008. In: fatek.seapraha.cz [online]. [cit. 05. 08. 2017]. Dostupné z: <http://fatek.seapraha.cz/zakladr/>
- [17] Mapp technology. In: br-automation.com [online]. [cit. 12. 08. 2017]. Dostupné z: <https://www.br-automation.com/cs/technologie/mapp-technology/>
- [18] Jiří Skařupa. 2007. Průmyslové roboty a manipulátory. In: elearn.vsb.cz [online]. [cit. 20. 08. 2017]. Dostupné z: http://www.elearn.vsb.cz/archivcd/FS/PRM/Text/Skripta_PRaM.pdf
- [19] FTP. 07. 04. 2017. In: wikipedia.org [online][cit. 25. 08. 2017]. Dostupné z: https://cs.wikipedia.org/wiki/File_Transfer_Protocol
- [20] Acopos P3. 2017. In: br-automation.com [online]. [cit. 20. 05. 2017]. Dostupné z: <https://www.br-automation.com/en/company/newsletter/archiv/2016/newsletter-052016/acopos-p3/>
- [21] OPS1100.1. 2017. In: br-automation.com [online]. [cit. 20. 05. 2017]. Dostupné z: http://www.br-automation.com/downloads_br_productcatalogue/BRP44400000000000000514909/OPS1100.1-ENG_V2.11.pdf
- [22] Automation PC 910. říjen. 2016. In: br-automation.com [online]. [cit. 20. 05. 2017]. Dostupné z: http://www.br-automation.com/downloads_br_productcatalogue/BRP44400000000000000478712/MAAPC900-ENG%20V1.40.pdf