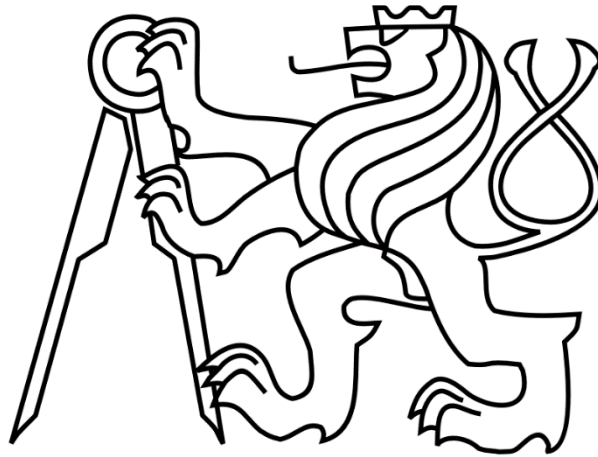


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení



# Diplomová práce

Implementace výroby na robotické montážní buňce

2022

Bc. Lukáš Jílek

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Jílek** Jméno: **Lukáš** Osobní číslo: **467430**  
Fakulta/ústav: **Fakulta strojní**  
Zadávající katedra/ústav: **Ústav výrobních strojů a zařízení**  
Studijní program: **Průmysl 4.0**  
Studijní obor: **bez oboru**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Implementace výroby na robotické montážní buňce**

Název diplomové práce anglicky:

**Production implementation on a robotic assembly cell**

Pokyny pro vypracování:

Popis tématu: Hlavním cílem práce je implementace zadaného výrobního postupu modelu dálkově ovládaného vozidla na připravenou robotickou buňku. Robotická buňka se skládá ze 3 robotů (SCARA, šestiosý angulární, dvouramenný kolaborativní) a dopravníku. Práce je zaměřena na vlastní návrh implementace, přípravu virtuálního modelu linky, virtuální zprovoznění včetně simulace a konstrukční návrh dílčích component nezbytných pro manipulační a montážní operace.; Osnova práce: 1. Rešerše v oblasti programování použitých robotů a řídicích systémů, rešerše v oblasti nezbytných automatizačních component 2. Návrh implementace výrobního postupu na výrobní linku 3. Příprava virtuálního modelu linky 4. Virtuální zprovoznění včetně přípravy nezbytných programů 5. Simulační testování výroby; Rozsah grafické části: schéma pracoviště, virtuální model pracoviště, modely uchopovačů a dalších přípravků; Rozsah textové části: 60-80 stran

Seznam doporučené literatury:

HOUŠA, Jaromír. Stavba výrobních strojů II. Dot. Praha: České vysoké učení technické, 1994.; Firemní dokumentace ABB [online]. Dostupné z: <https://new.abb.com/cz/>; Firemní dokumentace B&R Industrial Automation [online]. Dostupné z: <https://www.br-automation.com/cs/>;

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Jiří Švéda, Ph.D. ústav výrobních strojů a zařízení FS**

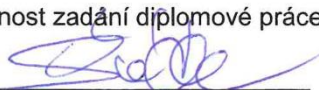
Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

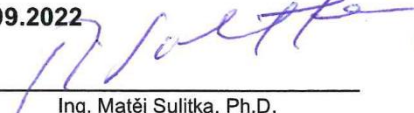
**Ing. Ondřej Novák katedra řídicí techniky FEL**


Datum zadání diplomové práce: **29.03.2022**

Termín odevzdání diplomové práce: **25.07.2022**

Platnost zadání diplomové práce: **30.09.2022**

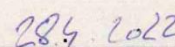
  
Ing. Jiří Švéda, Ph.D.  
podpis vedoucí(ho) práce


  
Ing. Matěj Sulítka, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

  
prof. Ing. Michael Valášek, DrSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

  
Datum převzetí zadání

  
Podpis studenta

---

## Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a že jsem uvedl v příloženém seznamu veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací, vydaným ČVUT v Praze 1. 7. 2009.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 9. 9. 2022

.....

Bc. Lukáš Jílek

---

## Poděkování

Poděkování bych chtěl věnovat lidem, díky kterým jsem se dostal až do fáze, kdy mám tu čest psát tato slova. Jedná se o mou rodinu. Je to rodina, která mi umožnila nejen studovat, ale také stála vždy při mně, a to za jakékoliv situace. V průběhu studia se k rodině přidala ještě jedna osoba, a to má láska, která svou empatií, humorem a podporou mojí motivaci ještě více pozvedla. Ač se jedná jen o pár řádků, děkuji Vám všem za vše.

Poděkování si rovněž zaslouží můj vedoucí práce Ing. Jiří Švéda Ph.D. a Ing. Pavel Burget Ph.D., který mi umožnil pracovat na tomto tématu. Dále děkuji kolegům z Testbedu pro Průmysl 4.0, kterými jsou Bc. Serhii Voronov a Lukáš Kornel Gajdár za skvělý design elektroniky.

---

## Anotace

Autor:	Bc. Lukáš Jílek
Název diplomové práce:	Implementace výroby na robotické montážní buňce
Rozsah práce:	89 Str., 82 obr., 10 tab.
Akademický rok:	2022
Škola:	ČVUT – Fakulta strojní
Ústav:	Ú12135 – ÚSTAV výrobních strojů a zařízení
Vedoucí:	Ing. Jiří Švéda, Ph.D.
Konzultant:	Ing. Ondřej Novák
Zadavatel:	ČVUT – Fakulta strojní
Klíčová slova:	Návrh montáže, Automatizované montážní pracoviště, Kolaborativní robot, PLC řízení, Povelování robotů z PLC, RobotStudio, AutomationStudio, ROBOTemplate, ACOPOStrak, CIIRCtrack
Anotace:	Tato diplomová práce je zaměřena na návrh předmětu montáže a jeho následnou implementaci do výrobní buňky v CIIRC. Pro realizaci této implementace bylo potřeba získat základní poznatek o simulačních nástrojích a jejich použití v praxi. V rešeršní části je dále probírané téma robotiky, a to především s ohledem na ty roboty, které se nachází v robotické buňce, které se diplomová práce týká. Na téma robotiky navazuje rešerše o řízení a programování robotů. V rámci vlastní práce byl proveden návrh předmětu montáže a výrobní postup s ohledem na dispozice montážní buňky. Po návrzích následuje již samotná práce se softwary pro simulaci robotické buňky, jejichž cílem je virtuální zprovoznění.

---

## Anotation

Author:	Bc. Lukáš Jílek
Title of diploma thesis:	Production implementation on a robotic assembly cell
Extent:	89 p., 82 fig., 10 tab.
Academic year:	2022
University:	CTU – Faculty of Mechanical Engineering
Department:	Ú12135 – Department of Production Machines and Equipment
Supervisor:	Ing. Jiří Švéda, Ph.D.
Consultant:	Ing. Ondřej Novák
Submitter of the theme:	CTU – Faculty of Mechanical Engineering
Key words:	Assembly design, Automated assembly workplace, Collaborative robot, PLC control, robots commanding from PLC, RobotStudio, AutomationStudio, ROBOTemplate, ACOPOStrak, CIIRCtrack
Anotation:	<p>This thesis focuses on the design of an assembly object and its subsequent implementation in a production cell at CIIRC. To realize this implementation, it was necessary to gain a basic understanding of simulation tools and their use in practice. In the research part, the topic of robotics is further discussed, especially with respect to those robots that are located in the robotics cell that the thesis is concerned with. The topic of robotics is followed by research on their control and programming. Within the actual thesis, the design of the subject assembly and the manufacturing process has been carried out with respect to the layout of the assembly cell. After the designs, the actual work with the software for the virtual commissioning of the robotic cell is already following.</p>

## Obsah

1	Úvod.....	4
2	Průmyslové roboty a jejich řízení .....	6
2.1	Simulační nástroje.....	6
2.2	Průmyslové roboty.....	9
2.3	Programování a řízení robotů .....	18
3	Robotická buňka CIIRCtrack.....	26
3.1	Popis pracoviště .....	26
3.2	Roboty pracoviště .....	26
3.3	Nástroje robotů.....	28
3.4	Dopravník ACOPOStrak.....	31
4	Předmět automatizované montáže .....	32
4.1	Popis předmětu.....	34
4.2	Montážní postup.....	37
4.3	Přípravky .....	40
4.4	Návrh uchopovacích prvků .....	44
5	Simulační model .....	49
5.1	Robotická linka.....	50
5.2	Import robotů .....	50
5.3	Modely objektů.....	50
5.4	Příprava mechanismů .....	51
5.5	Příprava předmětů pro montáž .....	51
5.6	Tvorba pracovních objektů .....	52
5.7	Instalace kontroléru .....	54
6	Řídící programy.....	55
6.1	Roboty.....	55
6.2	SmartComponents .....	59
6.3	PLC.....	66
7	Simulační testování.....	76

Simulace.....	76
Výstupní data.....	78
8 Závěr .....	80
Seznamy.....	82
Seznam použité literatury.....	82
Seznam použitých softwarů.....	86
Seznam obrázků.....	86
Seznam tabulek.....	88
Seznam příloh .....	89



## Seznam použitých zkratk

ACIS	Geometrický formát 3D dat
CAD	Computer aided design – počítačem podporované projektování
HMI	Human machine interface – uživatelské rozhraní
MoveJ	Move joint – pohyb robotu přes klouby
MoveL	Move linearly – lineární pohyb robotu
NIOSH	National Institute for Occupational Safety & Health – Národní institut pracovní bezpečnosti a zdraví
OLP	Off-line Programming – Off-line programování
OPC UA	Open Platform Communications – Unified Architecture – otevřený komunikační standard
OWAS	Owako Working posture Assessment System – systém analýzy pracovní polohy Ovako
PLC	Programmable logic controller – programovatelný logický kontrolér
RCS	Robotic Controller Simulation – simulovaný robotický kontrolér
RRS	Realistic Robot Simulation – Simulovaný reálný robot
TCP	Tool center point – středový bod nástroje robotu
VRC	Virtual Robot Controller – virtuální robotický kontrolér
Wobj	Work object – Souřadný systém pracovního objektu

## 1 Úvod

Téma práce bylo zvoleno na základě možnosti pracovat na nově vzniklé robotické buňce v CIIRC, která bude v práci zmiňována také pod jménem CIIRCtrack. Pro tuto buňku byla potřeba navrhnout předmět montáže v rámci plánovaného projektu distribuované výroby. Část montáže tohoto předmětu probíhá právě v této robotické buňce. Před fyzickou realizací však bylo potřeba provést virtuální zprovoznění robotické linky, aby byla ověřena realizovatelnost a softwarová architektura. To vyžadovalo propojení dvou softwarů, a to RobotStudio a AutomationStudio. Vzhledem k tomu, že pracoviště je ve skutečnosti řízené celé z PLC, byla i tato práce zaměřena na stejný způsob řízení.

Cílem diplomové práce bylo provést návrh předmětu montáže s výrobním postupem, na základě kterého byla provedena příprava robotické linky, programování robotů a PLC pro její řízení, aby mohla být navržena montáž implementována za cílem vzniku virtuálního zprovoznění. Před zahájením vlastní práce bylo ovšem potřebné provést rešerši v několika oblastech týkajících se tématu, aby byly nalezeny nástroje a postup pro řešení práce. Protože je téma zaměřeno na simulace a robotiku, byla i na tato témata provedena rešerše. Nejdříve byla provedena rešerše, která se týkala simulačních nástrojů. K těmto simulačním nástrojům bylo dohledáno, jaký je jejich význam a použití v praxi a v čem se jednotlivé nástroje odlišují. Po vyjasnění základů simulačních nástrojů následuje kapitola, která je zaměřena na roboty. V rámci této kapitoly jsou ukázány různé kinematiky robotů, a to zejména těch, které se v robotické buňce nachází. V kapitole je možné se dozvědět jejich základní odlišnosti s výhodami a nevýhodami a také jaké je v praxi jejich nejčastější využití. Téma robotiky je následováno kapitolou, která je již zaměřená na programování robotiky a způsoby jejího řízení. Během této kapitoly bylo nejdříve potřeba vyjasnit základní terminologie programování, po které přijdou na řadu způsoby programování. Zde jsou popsány rozdíly mezi jednotlivými způsoby a také jejich opodstatnění, kdy jaký způsob použít. Kapitola je zakončena metodami řízení robotiky, která má za cíl poukázat na význam těchto metod a jejich odlišnosti.

Po rešeršní části již přichází na řadu kapitola, která je zaměřena na samotnou robotickou buňku, kde je psáno o technologickém vybavení robotické buňky. Kapitola je následována vlastním řešením a vypracováním tématu.

Vlastní práce byla zahájena návrhem předmětu montáže, kterým bylo auto na dálkové ovládání, jehož velká část byla vyrobena s pomocí 3D tisku a zakázkových komponent. Tento návrh byl proveden tak, aby montáž auta mohla být zcela automatizována a realizována na více robotických pracovištích, z čehož část montáže bude prováděna právě na robotické buňce CIIRCtrack. Toho bylo dosaženo zjednodušením předmětu montáže s využitím zakomponovaných montážních mechanismů a zjednodušením elektroniky, která musí být zcela bez kabelů. Po návrhu předmětu montáže následoval návrh výrobního postupu pro montážní buňku. Dle montážního postupu byly následně provedeny návrhy přípravků a uchopovacích prvků.

Po návrhové části následovala část programovací, která byla zahájena přípravou virtuálního modelu sloužícího pro simulaci montáže v softwaru RobotStudio. Příprava zahrnovala importování robotiky a jejich pozicování v robotické buňce. U robotů byla provedena instalace robotického kontroléru a jeho příprava pro komunikaci s PLC. Následujícím krokem byly importy již samotných modelů z návrhové části práce, sloužících pro realizaci montáže, a to předmětu montáže a předmětů s výrobou spojených. Tato geometrie byla rozmístěna na základě montážního postupu po pracovišti, během čehož byla provedena i příprava souřadných systémů které budou později sloužit k programování robotiky. Po přípravách následovalo programování robotiky, jejichž trajektorie byla tvořena po úsecích, kvůli pozdějšímu řízení z PLC, které obstarává řízení a logiku pracoviště. Pro virtuální pracoviště bylo provedeno doprogramování dílčích logických úloh s pomocí doplňku, které v softwaru nejsou obsaženy. S pomocí tohoto doplňku bylo provedeno programování dopravníku. Zároveň ve virtuálním modelu pracoviště nebude použit nástroj pro simulaci fyzikálních procesů, kvůli potenciálnímu využití pro digitální dvojče. Nástroj simulace fyziky by mohl vnášet nepředvídatelné jevy do digitálního dvojčete. Proto posledním krokem bylo s pomocí doplňku provedeno doprogramování veškerých potřebných fyzikálních jevů.

Závěrečná část práce byla orientována na práci v AutomationStudio, kde probíhalo programování virtuálního PLC pro řízení virtuálního robotického pracoviště vytvořeného v softwaru RobotStudio. Zde bylo nejdříve provedeno vysvětlení deklarovaných proměnných a uvedení návrhu architektury řízení, které bylo rozděleno do tří vrstev tak, aby řídicí vrstva byla schopna fungovat odděleně a nezasahovala nijak do vrstev nižších. Během této kapitoly byly vysvětleny účely jednotlivých vrstev a jejich obsah, kde bylo zároveň i uvedeno, jak je nastavena komunikace při použití více než jednoho robotického kontroléru. Na závěr bylo provedeno simulační testování na virtuálním modelu a identifikace produktivity linky.

## 2 Průmyslové roboty a jejich řízení

Dnes jsou brány automatizované výrobní procesy jako samozřejmost, protože mnohdy bez automatizace podnik není konkurenceschopný. Automatizací je docíleno, že jsou odstraněny výrobní chyby, jsou sníženy výrobní časy a zároveň je navýšena i výrobní kapacita. Tento automatizovaný systém je však nutné správně navrhnout. Jak si tým, který výrobní linku navrhoval, může být jistý, že zvolil správné řešení rozložení pracoviště a vše správně naprogramoval? Jak řešení mohou obhájit společnosti, které linku dodávají? K ověření a zodpovězení otázek slouží simulační nástroje.

### 2.1 Simulační nástroje

Jedná se o matematickou reprezentaci komplexního průmyslového systému, který umožňuje zlepšit skutečný průmyslový systém na základě výstupních dat. Využívá se při programování či zkoušení různých optimalizací, které by při aplikaci na skutečné lince byli drahé a riskantní.

Simulační nástroje se používají při takzvaném off-line programování. Výhodou tohoto typu programování je, že při programování není nutné mít fyzicky k dispozici programované zařízení. Programování může tedy probíhat nezávisle na reálných komponentech pracoviště. Pokud je zařízení již někde implementováno, není nutné jej odstavovat a omezovat celý výrobní proces. Tehdy, kdy je program připraven, jednoduše se do zařízení stáhne a odladí. U robotů například polohovou korekcí. Právým opakem off-line programování je on-line programování, které vyžaduje fyzickou přítomnost programovaného zařízení. Zařízení je tak omezeno ve své produkční činnosti.

S pomocí simulačních nástrojů si lze věrohodně ověřit navržený software, aniž by byl ohrožen reálný hardware. Jsou zde odstraněny chyby v kódu, které v průběhu programování vznikly. Dojde-li ke kolizi v simulačním prostředí, nestojí to žádné finanční ani časové prostředky na opravu. Mezi další časté chyby, které se mohou objevit při absenci simulačních modelů patří:

- Linka zabírá nadbytečný prostor
- Robot nedosáhne na požadovanou pozici
- Kolize robotů
- Zabezpečení ve formě plotů, které omezuje produkci
- Neoptimalizované pohyby robotů

Volba simulačního softwaru je závislá na programované technologii, aplikaci, diferenci výrobců zařízení a jejich počtu, které budou programovány. Rozhodujícím faktorem je také to, jaké výstupy jsou od simulace očekávány. Proto simulační nástroje lze rozdělit následovně:

- Simulační prostředí výrobce dané technologie
- Komplexní simulační prostředí

### *Simulační prostředí výrobce dané technologie*

Někteří výrobci disponují vlastními simulačními nástroji, které jsou vyvíjeny přímo na jejich technologii. Přínosem je odladěný software s veškerými funkcemi k jejich technologii, avšak s tím limitem, že mimo technologii výrobce je software nemožný použít.

### Roboty

Takový problém se týká především robotů. Každý výrobce má vlastní programovací jazyk pro své kontroléry<sup>1</sup> robotů. Tomuto programovacímu jazyku je přizpůsobené i programovací prostředí, tedy simulační nástroj. Problém tedy nastává v tom, že specializované simulační prostředí neumí a ani neumožňuje komunikaci s kontroléry jiných výrobců. Takto specializovaný software způsobuje problém zejména během rozvoje linky, kdy je buď nutnost zůstat u stejného výrobce, či nutnost přejít na jiný simulační nástroj a vše přeprogramovat. Podrobněji o způsobech řízení robotů bude pojednáno v následující podkapitole „Řízení robotu“.

### PLC

Jednotlivá programovací prostředí významných výrobců PLC disponují již v základu či ve formě doplňku schopností simulovat PLC. Jedná se však jen o zprovoznění programu ve virtuálním prostředí bez vizuální zpětné vazby. Výrobců, kteří dokážou nabídnout možnost vizualizace programu PLC, není mnoho. Zpravidla je potřeba použít software třetích stran, kdy je použit CAD software či rovnou komplexní simulační nástroj pro virtuální zprovoznění, které jsou však cenově velmi nákladné. Mezi výrobce PLC, kteří jsou schopni poskytnout možnost simulace programu formou virtuálního zprovoznění PLC, patří například Siemens či B&R.

**Siemens** – Při volbě PLC od společnosti Siemens je programování realizováno v softwaru zvaném TIA Portal. K tomu je nutné mít software PLCsim Advanced pro simulování virtuálního PLC. Do toho lze následně nahrávat vytvořené programy. Díky tomuto simulovanému PLC jsme schopni provést virtuální zprovoznění programu na simulačním modelu s předem definovanou kinematikou. K simulování lze použít například CAD software Siemens NX s doplňkem Mechatronics concept designer či Process Simulate od téhož výrobce. Tento software bude popsán podrobněji v podkapitole „Komplexní simulační nástroje“.

**B&R** – Programování PLC od společnosti B&R je realizováno v softwaru AutomationStudio, kde pro simulaci virtuálního PLC poslouží již integrovaný ARsim. Do ARsim jsou nahrávány vytvořené programy. Jako software pro jednoduché virtuální zprovoznění poslouží B&R SceneViewer, který však nedisponuje simulací fyziky. V případě požadavku detailnějších simulací je zde nutné použít software třetí strany zvaný IndustriaPhysics, která však se společností B&R spolupracuje. Software je tedy přizpůsoben.

---

<sup>1</sup> Jedná se o volný překlad do češtiny z anglického jazyka - Robot controller, tedy robotická řídicí jednotka.

### *Komplexní simulační nástroje*

V praxi je běžné, že se na pracovišti nachází více různých výrobců například robotů. Proto vzniklo universální programovací prostředí, která bariéru různorodosti robotů bourají. Tato prostředí umožňují kombinovat různé výrobce dohromady. Komplexní simulační nástroje také zpravidla slouží pro virtuální zprovoznění celých výrobních linek, kdy není řešeno dílčí zařízení, ale je řešeno celé pracoviště. Tedy propojení více robotů s PLC a jinými technologiemi na pracovišti. Jedná se o softwary, které už umožňují vyladit takt výrobní linky, ověřit tok materiálu, výrobní kapacitu, montáže, lakování apod.

V některých softwarech je možné ověřovat i ergonomii pracoviště pro dělníky na základě světově používaných analýz pro odstranění rizika zranění jako je například OWAS, Lower back či NIOSH.

Tyto softwary dnes už poměrně často podporují i virtuální realitu, která otevírá dveře novému způsobu výcviku operátorů a dělníků přímo na výrobní lince v simulačním prostředí. Lze tak školit na základní zacházení s technologií či dokonce na nepředvídatelné situace, havárie a jejich následné řešení.

Mezi nejznámější softwary virtuálního zprovoznění patří Siemens Process Simulate. Jedná se o komplexní řešení pro ověření navržené výroby v 3D prostředí. Lze tu simulovat jak techniku, tak i lidskou pracovní sílu.

Co se týče programování, tak v těchto simulačních nástrojích je možné programovat roboty a nikoliv PLC. Programování PLC je ponecháno originálnímu softwaru výrobce. Lze jej však emulovat či napojit přes OPC UA. Přístupy programování robotů v komplexních simulačních nástrojích jsou následující (viz Obrázek 1).



*Obrázek 1 –Obsah kapitoly - programovací prvky*

**VRC** – Při tvorbě programu pro roboty lze využít takzvaný VRC (Virtual robot controller). Jedná se o kontrolér, který je možný použít na všechny roboty. Obsahuje základní příkazy a parametry, které je možné modifikovat. Využít jej lze například tehdy, kdy není jasné, jaký robot bude v konečném projektu použit. Kontrolér je velmi přesný v porovnání se skutečným kontrolérem robotů. Cílové body pohybových rutin virtuálního a skutečného kontroléru jsou stejné, avšak trajektorie mezi těmito body je odlišná. Do softwaru je možné instalovat doplňkové balíčky různých výrobců robotů, které simulují reálný kontrolér. Jedná se o takzvané RCS moduly.

**RCS** – (Robot Controller Simulation) moduly poskytují spolehlivá data pro přesné plánování trajektorie robotu a jeho inverzní kinematiky. Jelikož je známa charakteristika a limity daného robotu, lze přesně vyřešit natočení kloubů pro toto plánování. Znamená to, že trajektorie mezi definovanými body pohybové rutiny nebude již nepředvídatelná či *proměnlivá*.

**OLP** – Balíček OLP (Off-line Programming) umožňuje simulování Teach Pendantu k robotu. Co je Teach Pendant je k dohledání v podkapitole „Teach Pendant programování“. Poskytuje možnost zacházet ve virtuálním prostředí s robotem jako se skutečným a vzniká tak **RRS** (Realistic Robot Simulation). To znamená, že OLP poskytuje i aplikaci pro nahrávání a stahování programů robotu, přesně jako u skutečného zařízení. [1] [2] [3] [4] [5]

## 2.2 Průmyslové roboty

Na trhu je dnes velmi široký sortiment robotů (viz Obrázek 1). Stačí si jen zvolit. Ale jak poznat, že se jedná o správně zvolený robot, když se odlišují v tolika parametrech? Tato kapitola bude zaměřena právě na tyto odlišnosti robotů a jak jejich odlišnost využít ve prospěch pracoviště.

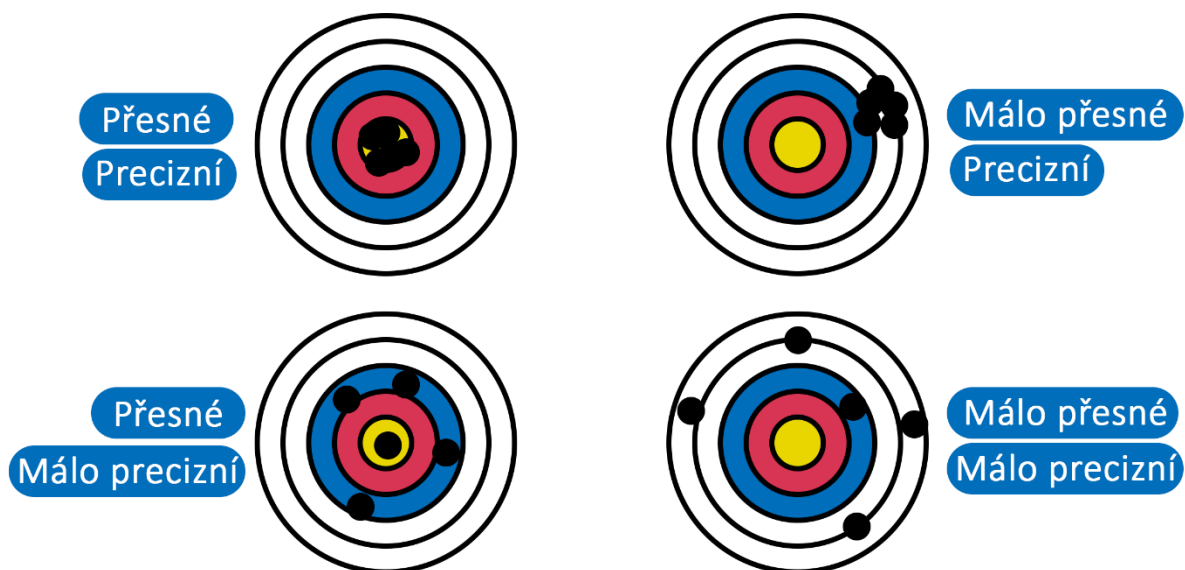


Obrázek 2- Obsah kapitoly - Roboty

Ovšem ještě před samotnými popisy robotů budou vysvětleny dva termíny, které v práci budou používány. Jedná se o termín „přesnost“ a „opakovatelnost“. Jde o velmi podstatné termíny, které v praxi není radno zaměnit. K vysvětlení poslouží jednoduchý obrázek terče (viz Obrázek 3).

**Přesnost** – Cílem je dostat body co nejbližší do středu terče. Se zmenšující se vzdáleností od středu narůstá přesnost zařízení, v tomto případě robotu. Analogicky čím větší vzdálenost od středu, tím menší přesnost robot má.

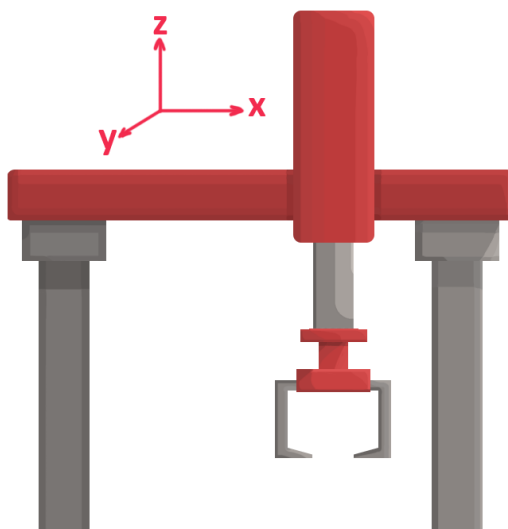
**Opakovatelnost (preciznost)** – Lze ji chápat jako shluk či rozptýl bodů. Čím menší rozptýlení body mají, tím větší opakovatelnost robot má.



Obrázek 3 - Přesnost a preciznost

### Kartézský robot

Jedná se o robot, který se pohybuje v kartézském souřadnicovém systému tří os X, Y a Z translačně (viz Obrázek 4). Jednotlivé pohybové osy mají mezi sebou pravý úhel. Proto je pracovní obálka tohoto robotu ve tvaru čtverce či kváдру.



Obrázek 4 - Kartézský robot

### Výhody

Je to robot s největší tuhostí právě kvůli jeho konstrukci. Každá z os je totiž podepřena na obou koncích. Kvůli jejich tuhosti jsou schopny dosahovat velkých nosností v celé délce osy přičemž dosahují i dobré přesnosti a opakovatelnosti. Konstrukce robotu rovněž umožňuje vykonávat lineární pohyby. Kartézské roboty je zároveň poměrně snadné naprogramovat.



### Nevýhody

Nevýhodou těchto robotů je zejména náročnost na prostor pro umístění. Část prostoru, který robot zabere se nevyužije jako pracovní.

### Aplikace

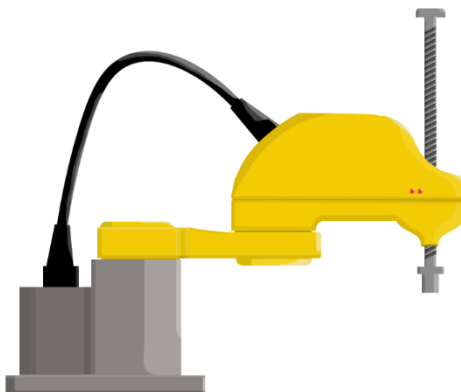
Pokud je na umístění robotu dostatek prostoru, jeho aplikace nemá žádné omezení. Mohou být použity jak na velké manipulace přes celou halu, tak mohou být použity i při drobných montážích.

Nejčastější aplikace jsou:

- Pick and Place
- Montáže
- Zásobování strojů či obrábění

### SCARA

Jedná se o čtyřosý robot. Má tři rotační osy a jednu translační. Název robotu vznikl zkrácením původního názvu „Selective Compliance Assembly Robot Arm“ (viz Obrázek 5). Pohyby jsou velice podobné lidské ruce. Disponují ramenem, loktem a zápěstím. Jedná se o robot cylindrický s cylindrickou pracovní obálkou. Rotační osy kloubů jsou rovnoběžné s osou Z. SCARA roboty jsou tak poddajné v rovině X-Y, ale tuhé v ose Z, což je jejich velkým plusem.



Obrázek 5 - SCARA robot

### Výhody

SCARA roboty dosahují velkých rychlostí pohybu. Ve své podstatě jsou rychlejší než šestiosé roboty. Díky pozici rotačních os, které jsou rovnoběžné s osou „Z“, je robot ve vertikálním směru tuhý. Tím robot dosahuje velmi dobré přesnosti a opakovatelnosti. Přesnost, které jsou roboty schopné dosáhnout, je menší než 10 mikronů.

### Nevýhody

Ve srovnání s kartézským robotem jsou roboty SCARA tuhé o něco méně. SCARA roboty mají malou nosnost, pokud jsou srovnány například s šestiosými roboty či kartézským.

Co může působit problematicky, to je pracovní obálka robotu, která je obtížně kontrolovatelná. Zároveň jsou roboty vysoké, to znamená, že vyžadují dostatek prostoru nad sebou.

Robot má zároveň problémy s flexibilitou a zručností kvůli své cylindrické pracovní obálce. Nehodí se například pro aplikace, kde robot musí sledovat 3D konturu nějakého výrobku.

#### Aplikace

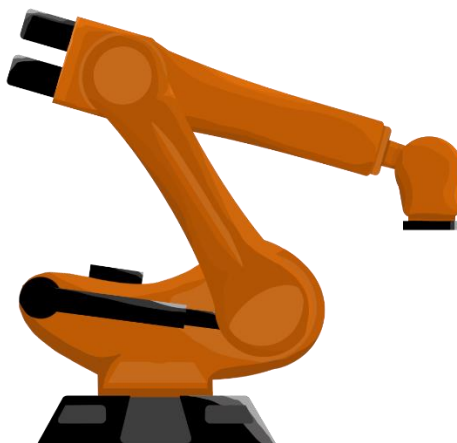
Jsou velice vhodné pro montáže, kde je vyžadována velká rychlost pohybu například v automobilovém průmyslu či elektronice. Jsou proto označovány jako nejrychlejší a nejlevnější roboty pro aplikace vyžadující vysokou rychlost. [6] [7] [8] [9]

Nejčastější aplikace jsou:

- Pick and Place
- Pájení
- Drobné montáže

#### Angulární roboty

Tyto roboty jsou svou kinematikou podobné lidské ruce (viz Obrázek 6). Obvykle mají šest rotačních os. Pracovní obálka u těchto robotů je sférická.



Obrázek 6 - Kloubový robot

#### Výhody

Jsou to nejflexibilnější roboty ze všech. Tyto roboty mohou být silné a schopné dosahovat jak velké rychlosti, tak i velké nosnosti. Díky své flexibilitě a obratnosti mohou dosáhnout na obtížně přístupná místa.

#### Nevýhody

Tyto roboty vyžadují robustní řídicí systém. Zároveň jsou náročné na pracovní prostor kvůli bezpečnosti. Jejich pohyb je rychlý a pro člověka nepředvídatelný. Je proto nutné pro roboty

udělat oddělené a oplocené pracoviště. Oproti ostatním robotům jsou pomalejší a ve srovnání se SCARA roboty jsou až dvakrát méně přesné.

#### Aplikace

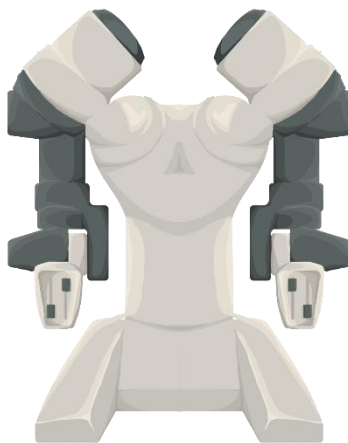
Skrze počet os jsou roboty velice flexibilní a jelikož se díky tomu vlastnostmi blíží lidské ruce, nejsou téměř ani meze pro jejich aplikaci. [9] [10] [11]

Nejčastější aplikace jsou:

- Pick and Place
- Sváření
- Lakování
- Inspekce
- Nanášení lepidla
- Srážení hran
- 3D tisk

#### Kolaborativní *robot*

Tyto roboty (viz Obrázek 7) jsou téměř stejné, jako roboty průmyslové kloubové. Hlavní rozdíl, kterým se tyto roboty odlišují, je bezpečnost. A velice často kolaborativní roboty mají ještě o jednu osu více.



Obrázek 7 - Kolaborativní robot

Kolaborativní roboty, bývají nazývané také jako koboty. Umožňují spolupráci s člověkem v bezprostřední blízkosti. Aby tuto spolupráci bylo možné realizovat, roboty musely projít značnou modernizací, a to včetně jejich řídicího systému.

### Bezpečnost

Aby byl robot označen za kolaborativní, nesmí hrozit žádné riziko poranění operátora. Tyto bezpečnostní prvky pro spolupráci robot-člověk lze rozdělit do čtyř kategorií.

**Vnitřní senzory** – Robot obsahuje senzory kroutícího momentu v kloubech, které pomáhají robotu analyzovat, zda došlo k překročení limitu či nikoliv. Nutno podotknout, že senzory musí počítat

i se zatížením senzoru od procesu, například při manipulaci s objektem. V případě překročení limitu dojde k zastavení pohybu. Záleží na naprogramované logice robota.

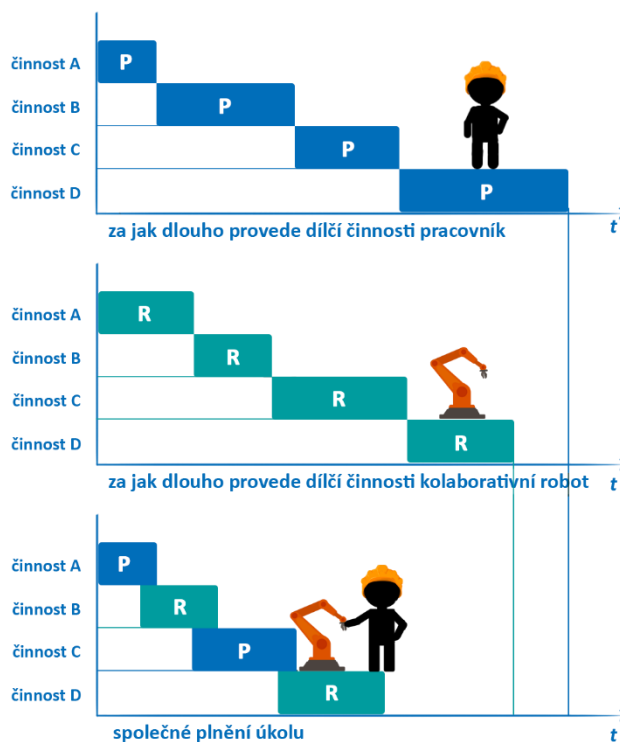
**Vnější senzory** – *Nejedná se zatím o příliš častou aplikaci, jelikož jde o záležitost ve vývojovém stádiu.* Prakticky se však jedná o robotickou kůži, která s pomocí kapacitních senzorů detekuje přítomnost vodivého či nevodivého předmětu. Tato detekce probíhá již na dálku před samotným kontaktem. Robot je tedy schopný „cítit“ a předejít tak kolizi.

**Design robotu** – *Tvary robotu jsou co nejvíce oblé, a v místech kloubů je vytvořené jakési odsazení, aby nemohlo dojít ke skřípnutí končetiny člověka.* Materiály, které se pro kolaborativní robotiku používají, jsou velice často plasty, z důvodu snížení hmotnosti ramene a polstrování rizikových částí pro změkčení kolize s člověkem, která je v případě kolaborace nevyhnutelná.

**Design procesu spolupráce** – Jedná se o nejdůležitější kategorii z celé bezpečnosti. Bezpečný nesmí být jen robot, ale i proces, do kterého je robot zapojen. Jedním z hlavních bezpečnostních prvků, kterými lze zajistit bezpečnější interakci, je rychlost pohybu robotu. Ta je stanovená v technické specifikaci. Zároveň kolaborativní robot nesmí používat ostré nástroje a manipulovat s ostrými předměty.

### Výhody

Jednou z hlavních předností kolaborativní robotiky je již samotné programování, které lze provést s pomocí ručního navádění robotického ramene bez nutnosti psaní kódu. Kolaborativní roboty ve výrobě dokážou zásadně snížit náročnost některých činností, kdy operátor a robot vzájemně využívají svých předností. Operátor zastane činnosti, kde je vyžadována flexibilita, kobot zastane pro změnu činnosti, které jsou neergonomické a repetitivní či náročné na pozornost. Jedná se takzvanou skill based task sharing aplikaci (viz Obrázek 8). Kobot rovněž není potřeba zavírat do klecí a zajišťovat nadbytečné bezpečnostní prvky. Může tedy pracovat samostatně mezi zaměstnanci a konat si svou práci. [12] [13] [10]



Obrázek 8 - Skill based Task sharing

### Nevýhody

Koboty mají obvykle daleko nižší nosnost a několikanásobně vyšší cenu, než mají průmyslové roboty stejné velikosti. Další nevýhodou je skutečnost, že tvorba kolaborativního pracoviště není dodnes řádně zdokumentována.

### Aplikace

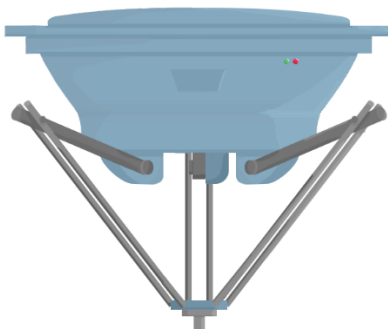
Kolaborativní robot prakticky nemá omezení. Může být použit i jako robot rehabilitační.

Nejčastější aplikace jsou:

- Pick and Place
- Nýtování
- Šroubování
- Obsluha stroje
- Nános lepidla
- Výstupní kontrola

### *Delta robot*

Roboty, které se montují nad pracoviště a svým vzezřením připomínají pavouka (viz Obrázek 9). Je to robot se třemi rameny, která jsou do tvaru paralelogramu. Disponuje třemi až čtyřmi osami v závislosti na tom, zda mají rotační zápěstí či nikoliv. Pracovní obálka těchto robotů je kuželovitého tvaru.



Obrázek 9 - Delta robot

### Výhody

Umístění motorů u těchto robotů je v základně. Tím se z Delta robotu stává nejrychlejší typ robotu vůbec, protože ramena a koncový efektor jsou lehká a nenesou žádnou nadbytečnou hmotu jako je to například u kloubových robotů. Roboty jsou také velmi přesné a flexibilní.

### Nevýhody

Delta roboty mají kvůli své konstrukci malý dosah a malou nosnost. Zároveň je nutné brát v potaz, jak vysoko bude robot umístěn, jelikož čím níže robot sáhne, tím menší vzdálenosti je schopný udělat do strany.

### Aplikace

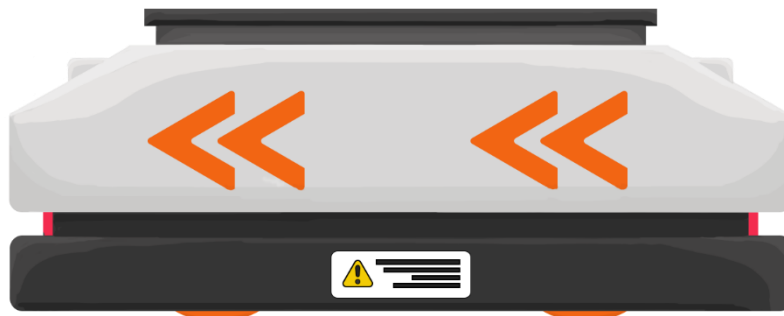
Určení těchto robotů je především pro aplikace, kde je vyžadována velká rychlost pohybu a přesnost. Avšak půjde-li o manipulaci, musí se jednat o díly s nízkou hmotností. [14] [10] [10]

Nejčastější aplikace jsou:

- Pick and Place
- Třídění
- Drobné montáže
- Balení

### *Autonomní mobilní robot (AMR)*

Pojízdný paletový vozík, který se umí pohybovat prostorem autonomně (viz Obrázek 10). Jedná se o následovníky AGV (Automated Guided Vehlices). V čem se tyto mobilní vozíky odlišují? Lze si udělat jednoduché srovnání.



Obrázek 10 - AMR

Návštěvník města volí mezi tramvají a taxíkem. Tramvaj představuje AGV a taxi AMR. Tramvaj návštěvníka odveze pouze poblíž místa, kam se návštěvník přeje dostat. Zároveň není schopna vyhnout se jakékoliv složité situaci, která může po cestě nastat, například koloně. Na rozdíl od toho se taxi dokáže dopravním neduhům vyhnout a návštěvníka odveze přímo až na žádané místo.

AGV se pohybují na základě předem vystavěné tratě, kterou představuje drát zabudovaný v zemi. K tomu bylo vozidlo opatřeno senzorem a jednoduchou logikou, aby v případě hrozby kolize zastavilo. Jak již z logiky věci vyplývá, zabudovaný drát v zemi nepředstavuje příliš flexibilní ani levné řešení.

AMR jsou postaveny oproti AGV na zcela jiné technologii. AMR se dokáže dynamicky navigovat s pomocí předem zmapovaného prostředí a samo si dokáže naplánovat trasu tak, aby se na místo určení dostalo co nejdříve. Zároveň se už jedná o chytré zařízení, které dokáže rozpoznat překážky a vyhnout se kolizi tak, že buď upraví svou trasu, nebo na okamžik zastaví. Svou práci AMR dokáže vykonat bez ohledu na to, jak je okolí rušné. [15] [16] [10]

### Aplikace

AMR slouží především pro dopravu materiálů.

Nejčastější aplikace jsou:

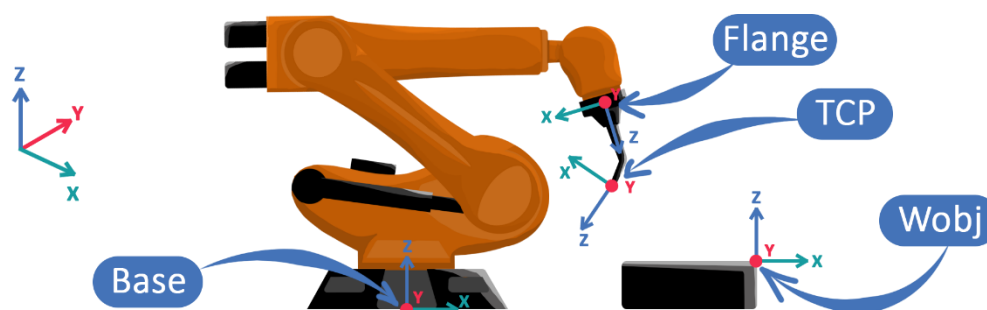
- Přeprava ve skladu
- Zásobování robotických buněk

### 2.3 Programování a řízení robotů

Tato kapitola (viz Obrázek 12) bude zaměřena na přístupy, jakými lze robot naprogramovat a jaké jsou případné odlišnosti mezi nimi. Další podkapitola bude následně zaměřena na způsoby řízení robotů. Ovšem nejprve bude proveden stručný úvod k souřadným systémům a kinematice robotů po kterém bude následovat podkapitola o vzniku trajektorií.

#### Souřadné systémy

Na začátek je vhodné uvést, jaké souřadné systémy robot má (viz Obrázek 11). Jelikož od těchto souřadných systémů se i následně odvíjí naprogramovaná trajektorie robotu a její modularita.



Obrázek 11 - Souřadné systémy robotu, ve kterých lze programovat trajektorii

Níže jsou popsány jednotlivé souřadné systémy:

**Base** – Jde o výchozí souřadný systém robotu, který se nachází v jeho patě či základně. Tento souřadný systém je dán výrobcem. Pokud není nastaven jiný souřadný systém programátorem, jsou veškeré programované trajektorie přiřazeny k tomuto souřadnému systému.

**Flange** – Souřadný systém na konci robotického ramene, který je aktivní tehdy, kdy robot nemá namontovaný či připnutý nástroj. Na základě tohoto souřadného systému je posléze vypočítávána trajektorie. Souřadný systém je opět dán výrobcem a nelze jej měnit.

**TCP** – Tento souřadný systém je volen programátorem. Představuje „nulu“ či střed nástroje. „TCP“ nahrazuje souřadný systém „Flange“.

**Wobj** – Jedná se o souřadný systém, který může nahrazovat „Base“. Více o významu tohoto souřadného systému se lze dočíst v podkapitole „Trajektorie robotu – Pracovní objekt“.

Nyní, když jsou popsány základní souřadné systémy robotu, lze se přesunout k řešením, jakými lze tvorby trajektorie dosáhnout, a to buď dopřednou či inverzní kinematikou.



### *Kinematika robotu*

Kinematika se zabývá pohybem těles s více stupni volnosti. Neuvažuje se jejich hmotnost a ani momenty setrvačnosti. Pro tento případ je zde psáno o robotickém ramenu. Jde v podstatě o matematické řešení, které má za cíl vypočítat parametry natočení kloubů či TCP na základě konfigurace robotu.

**dopředná kinematika** – Tento model pracuje se znalostí natočení všech kloubů robotu s jejichž pomocí je schopna dopočítat polohu a natočení koncového bodu (Flange, TCP), vůči zvolenému počátku (Base, Wobj).

**Inverzní kinematika** – Model, který vychází ze znalosti polohy bodu robotu, kterým může být bod koncový (Flange, TCP), ale může to být i bod kloubu robotu. S pomocí této informace následně řeší polohy všech zbývajících částí robotu tak, aby bylo požadované polohy skutečně dosaženo. Pro řízení se jedná o významnější model a veškeré tvořené trajektorie v následujících kapitolách z něj vycházejí.

Po stručném úvodu následuje samotná tvorba trajektorie robotu a její optimalizační parametry.



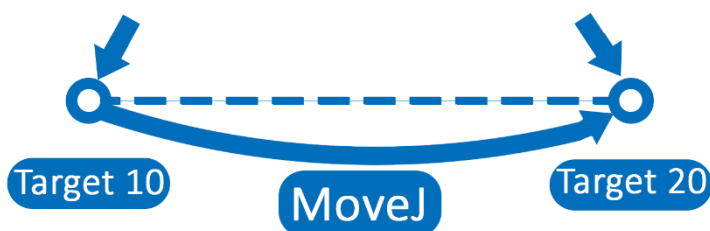
Obrázek 12 – Obsahu kapitoly - Programování a řízení robotů

### *Trajektorie robotu*

Trajektorie robotu vzniká s pomocí tvorby bodů v prostoru, také zvané jako „Targets“, o jejichž způsobu vzniku bude psáno v následující kapitole. Pospojováním bodů vytvořených v prostoru následně vzniká trajektorie robotu. Ta je sledována robotickým ramenem, které má na svém konci nadefinovaný bod zvaný jako TCP, neboli „Tool Center Point“. Tento bod představuje střed nástroje, který je volen programátorem.

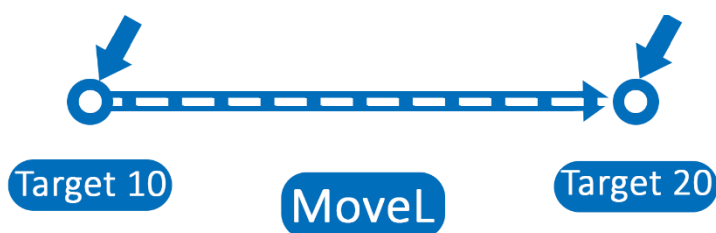
Tvar trajektorie a způsob pohybu, který robot bude při sledování trajektorie vykonávat, lze ovlivnit následujícími parametry, které budou popsány. Samozřejmostí je možnost nastavení rychlosti pohybu, která v bodech není zahrnuta.

**MoveJ** – Jedná se o parametr, který ovlivňuje způsob dosažení dalšího bodu po trajektorii. Tento pohyb je vykonán přes klouby robotu a představuje nejrychlejší možné dosažení dalšího bodu. Během toho, pohyb TCP nemusí sledovat přímku. Cílového bodu dosáhnou všechny osy robotu ve stejný čas (viz Obrázek 13).



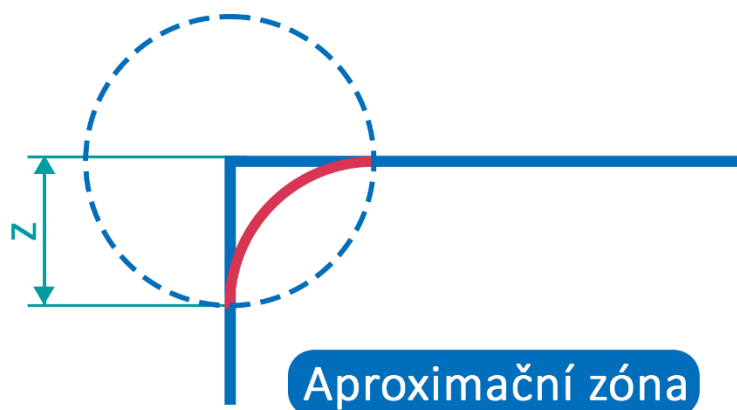
Obrázek 13 – MoveJ, šipka představuje nástroj robotu..

**MoveL** – Jedná se o lineární polohování TCP, kdy cílového bodu je dosaženo určitou rychlostí (viz Obrázek 14).



Obrázek 14 – MoveL, šipka představuje nástroj robotu.

**Aproximační zóny** – Jedná se o vyhlazování trajektorie v místě bodů, s pomocí kterých byla trajektorie vytvořena. S rostoucím parametrem zóny dochází k růstu zaoblení trajektorie, ale zároveň také k oddalování trajektorie od definovaného bodu v prostoru. (viz Obrázek 15).



Obrázek 15 - Aproximační zóna trajektorie

**Konfigurace robotu** – Jedná se parametr, kterým je definováno, jakým způsobem bude robot dosahovat daného bodu trajektorie. Konfigurace šestiosého robotu je definována s pomocí čtyř celých čísel, která mohou být pozitivní a negativní. Tyto hodnoty indikují, v jakém kvadrantu plné otáčky jsou umístěny významné osy robotu.

Kvadranty jsou číslovány od 0 pro kladný směr, který je proti směru hodinových ručiček a -1 pro záporné, které jsou ve směru hodinových ručiček. Pro příklad význam hodnot této konfigurace [0 -1 2 1] je následující.

První číslo „0“ definuje osu jedna někde v kvadrantu od 0 do 90 stupňů rotace. Druhé číslo „-1“ definuje čtvrtou osu od 0 do -90 stupňů. Třetí číslo „2“ definuje šestou osu někde mezi 180 a 270 stupni. Čtvrté číslo udává polohu virtuální osy x pro specifikaci středu zápěstí k ostatním osám.

**Pracovní objekt** – Nazývá se v angličtině jako „workobject“. Jedná se o souřadný systém, který určuje polohu dílce, regálu, pracovního stolu či například místa na dopravníku, kam budou díly zakládány. K tomuto souřadnému systému lze poté přidávat vytvořené body trajektorie.

Pokud například robot odebírá díly z regálu, který je k robotu dopravován s pomocí AMR, nebude poloha regálu vždy stejná. Bude-li regál nadefinován jako workobject, robot si může rychlými kalibračními dotyky regálu přeorientovat naprogramované body trajektorie a bude tak vykompenzována chyba změny polohy regálu.

Po stručném úvodu ke vzniku a úpravám naprogramované trajektorie, je na čase přesunout se ke způsobům, jak ji lze vytvářet.

**Programování robotu** – Přístupů, jak naprogramovat robot, je více. Některé programování je snadnější, jiné zase obtížnější. Volba způsobu programování ovšem není závislá na obtížnosti, ale také na podmínkách, za kterých programování probíhá. Rozhodující faktory mohou být rozděleny do tří následujících skupin:

- Obtížnost programování
- Komplexnost programované aplikace
- Snížení doby prostoje



Obrázek 16 – Obsah kapitoly – Způsoby programování robotů

V první řadě je nutná úvaha, zda bude programování probíhat on-line, nebo off-line (viz Obrázek 16). Off-line programování bude voleno tehdy, kdy bude cíleno na snížení doby prostoje výstavby linky a jejího uvedení do provozu. Někdy je však pro změnu vyžadována přítomnost programátora u robotu, proto bude zvoleno programování on-line. Avšak většinou se jedná o kombinaci off-line programování a on-line programování, kdy je potřeba naprogramované pohyby přes off-line programování odladit přímo u robotu. V následujících podkapitolách budou popsány způsoby on-line programování a off-line programování.

### *On-line programování*

On-line programování je používáno v malých podnicích. Aplikace, které by vyžadovali složité programování s mnoha řádky kódu, mohou být tímto způsobem poměrně rychle zajištěny. Záleží také na softwarové vybavenosti daného zařízení a obtížnosti aplikace.

**Teach Pendant programování** – Teach Pendant či pro roboty ABB FlexPendant je zařízení, které je napřímo zapojené do kontroléru robota. Vizuálně vypadají jako dotykové tablety s širokým kabelem (viz Obrázek 17). Jsou to zařízení, která jsou součástí vybavení téměř každého robota. Teach Pendantem je možné robotem polohovat a najíždět s ním do žádaných pozic, které jsou následně ukládány. Robot se tak učí pohybovou rutinu. Naučenou pohybovou rutinu si je možné pustit, odkrokovat a odladit. Skrze Teach Pendant je možné nahlížet i do kódu a provádět jeho editace. To umožňuje vložení pokročilé logiky do pohybové rutiny bez nutnosti editace programu skrze počítač.



Obrázek 17 - FlexPendant zařízení (pro robot IRB1300)

Použití tohoto způsobu programování je vhodné pro tvorbu jednodušších pohybů, u kterých není požadavkem sledovat robotem nějaké kontury obrobku nebo tvarové plochy.

Nevýhodou tohoto programování je nutnost odstavení robota a nastavení ho do režimu učení, což může způsobit prostoje. Zároveň se nejedná o způsob, který by se obešel bez programátorských znalostí a znalosti programovacího jazyka daného robota, který je jiný pro

každého z výrobců. Rovněž je potřeba dbát na bezpečnost a obezřetnost u takového způsobu programování.

**Ruční navádění** – Jedná se o nejnovější způsob programování, který nevyžaduje téměř žádné programátorské zkušenosti. Takový způsob programování je možný pouze u kolaborativních robotů. Při tomto programování je robotické rameno ručně polohováno v rámci pracovního prostoru, aby se robot naučil trajektorii pohybu. Malé kolaborativní roboty totiž mohou deaktivovat brzdy v kloubech a díky nízké hmotnosti rameno nespadne.

Tento způsob programování je velice intuitivní, rychlý, ale také méně přesný ve srovnání s jakýmkoliv jiným způsobem programování. Zároveň se nejedná o způsob programování, které by bylo standartně součástí každého robotu.

#### *Off-line programování*

Zásadní výhodou je snížení časových prostojů, jelikož program může vznikat paralelně s výstavbou pracoviště. Hodí se především pro aplikace velkých robotických pracovišť.

**Textové programování** – Tradiční metoda programování, kdy jsou psány jednotlivé řádky kódu ručně do textového editoru a následně staženy do robotu. Toto programování umožňuje přístup ke všem funkcím robotu, avšak nevýhodou tohoto programování je nutná znalost programovacího jazyka robotu a programování vůbec. Zároveň při ručním psaní kódu vznikají chyby, které je potřeba odladit ideálně v simulátoru robotu od výrobce.

**Grafické programování** – Programování robotu probíhá v simulačním prostředí, kde se simulovaný robot chová jako skutečný. O této problematice se pojednávalo v kapitole „**Chyba! Nenašel jsem zdroj odkazů.**“. Zde si je možné vytvářet souřadné systémy a vůči těmto systémům si posléze vytvářet body nástroje. Rovněž je zde možné nastavit sledování nějaké kontury apod. Jedná se v podstatě o kombinaci ručního navádění, programování a virtuální Teach Pendant. [17] [18] [19] [20] [21]

#### *Řízení robotu*

Při implementaci robotu na pracoviště je jedním ze zásadních rozhodnutí, jakým způsobem bude robot řízen, jelikož existují tři přístupy (viz Obrázek 18).



Obrázek 18 – Obsah kapitoly – Metody řízení robotu

I přesto, že robotické kontroléry obsahují veškeré pokročilé funkce robotu jako bezpečnost, detekce kolize či dokonce integrované vidění, nejedná se o příliš flexibilní řešení při implementaci

do velkých robotických pracovišť. Problematické zejména bývá, že s každým jiným výrobcem robotu přichází i jiný programovací jazyk. Oproti tomu řízení robotu přes PLC má tu výhodu, že je zcela postačující znát programovací jazyk PLC.

Mnoho inženýrů a techniků ve výrobě PLC rozumí. Tudíž si dokážou poradit s robotickým systémem řízeným na bázi PLC, aniž by byla nutná znalost programovacího jazyka robotického kontroléru.

Toto řízení zároveň usnadňuje práci při slučování robotu s automatizovaným pracovištěm. Nerobotické systémy automatizovaného pracoviště jsou obvykle řízeny přes PLC. Řízení robotu s pomocí stejného systému odstraní problematické rozhraní mezi dříve oddělenými řídicími systémy. Použitím jednoho řídicího systému pro vše znamená i sníženou náročnost na údržbu, aktualizace a řešení případných chyb.

Tento způsob řízení umožňuje daleko větší kompatibilitu, protože robot na automatickém pracovišti může být klidně vyměněn za robot jiného výrobce, aniž by se musel přepisovat program robotu.

Programování robotu přes PLC funguje jako jakýkoliv jiný program v PLC a to přes funkční bloky. Jedná se tedy o další způsob, jak robot naprogramovat, ačkoliv se již neovládá přes kontrolér robotu. Využití takového řízení má především potenciál na aplikace typu manipulace s materiálem jako je Pick and Place, kdy robot nepožaduje použití speciálních funkcí, které se nachází v kontroléru a jedná se o snadno programovatelný pohyb s pomocí funkčních bloků a souřadnic. [22] [23] [24]

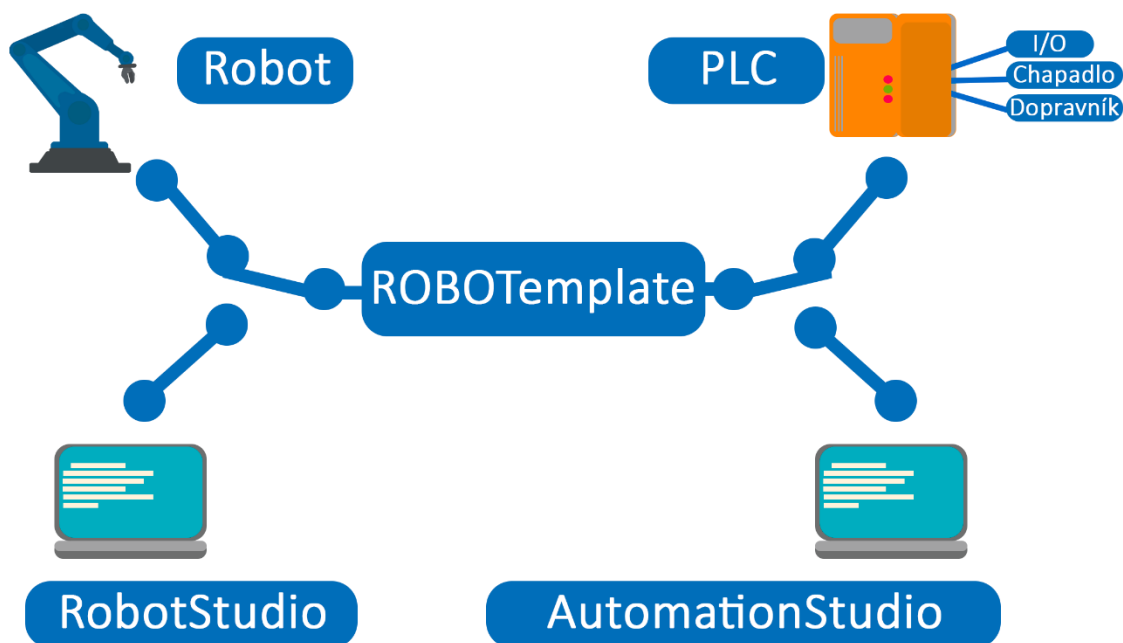
Další variantou je ovládání robotického kontroléru skrze PLC. Robot je naprogramován klasickým způsobem, který byl popsán výše v podkapitole „Programování robotu“, čímž vzniknou pohybové rutiny. Tyto pohybové rutiny jsou poté volány z PLC. Ovšem, aby tato komunikace byla možná, musí k tomu být doinstalována knihovna. A to jak do kontroléru robotu, tak i do PLC. Tímto způsobem je řešena komunikace mezi roboty ABB a PLC systémy B&R. Knihovna se nazývá ROBOTemplate.

#### *ROBOTemplate*

Tato knihovna slouží k navázání komunikace mezi roboty a PLC systémy (viz Obrázek 19). Smysl spočívá v tom, že robot obsahuje pouze pohybové instrukce a PLC řídí veškerou logiku. Lze tak kontrolovat linku z jednoho místa. Zároveň knihovna umožňuje editaci poloh robotu z PLC a vyčítat i případné chyby robotu, které lze zobrazovat například na HMI panelu.

S použitím této knihovny zároveň nedochází ke ztrátě funkcionalit robotu, jako je například detekce kolize či vyčítání aktuálního zatížení robotu. Ve své podstatě vše, co dokáže robot zobrazovat přes kontrolér, lze s pomocí této knihovny číst a ovládat přes funkční blok.

V předchozích kapitolách byla provedena rešerše zásadních oblastí, kterých se diplomová práce týká za cílem nalezení postupu a řešení práce. Nyní když je rešeršní část dokončena, lze se přesunout k vlastní práci tématu, která bude zahájena kapitolou týkající se popisu robotické buňky CIIRCtrack a jejího vybavení.



Obrázek 19 -Funkce ROBOTemplate

Mezi hlavní výhody patří také to, že vše jde simulovat s propojením ABB RobotStudio a B&R Automation studio, díky čemuž lze udělat i digitální dvojče pracoviště, na které bude zaměřena i tato diplomová práce. [25] [26]

### 3 Robotická buňka CIIRCtrack

Tato kapitola se bude věnovat již samotné robotické buňce v CIIRC a jejímu vybavení, které je v rámci diplomové práce důležité. Nejdříve bude popsána robotická buňka (viz Obrázek 20) a její účel, načež bude následovat popis robotů, kteří jsou na pracovišti, včetně jejich specifikací a nástrojů. Poslední část, která bude popsána v této kapitole, bude dopravník ACOPOStrak.



Obrázek 20 - Robotická buňka CIIRCtrack

#### 3.1 Popis pracoviště

Jedná se o robotickou buňku, která obsahuje řídicí systém PLC od společnosti B&R a roboty od společnosti ABB. Na tomto pracovišti byla využita i knihovna ROBOTemplate pro komunikaci mezi PLC a roboty, o které bylo pojednáno v podkapitole „Řízení robotu“.

Pracoviště se skládá ze tří robotů a dopravníku ACOPOStrak. Robotická buňka má být v budoucnu napojena na nadřazený systém, který bude komunikovat jak řízení linky, tak i její zásobování. Proto je robotická buňka vybavena elektrickou bránou, která bude umožňovat vjezd AMR se zásobami pro výrobu. Zásoby bude dopravovat mezi roboty dopravník ACOPOStrak s pomocí lineárních motorů, které budou sloužit jako nosič.

#### 3.2 Roboty pracoviště

Robotické pracoviště je vybaveno třemi roboty, kteří jsou od sebe velmi odlišní. Na pracovišti se nachází šestiosý robot IRB1300, poté robot SCARA, který má označení 910SC, a v poslední řadě dvouramenný kolaborativní robot YuMi označován jako IRB14000.



### *IRB1300*

Jedná se o klasický šestiosý robot (viz Obrázek 21), jehož úkolem na pracovišti je zejména vykládání a nakládání zásob na dopravník ze zásobovacího regálu a opačně. Tuto úlohu může vykonávat jako jediný z robotické buňky, jelikož se nachází v blízkosti místa, které je určené pro odložení regálu se zásobami.



*Obrázek 21 -Robot IRB1300*

Dosah robotu je 1,4 m a dokáže manipulovat s hmotností až 7 kg. Jedná se tedy o prodlouženou verzi robotu IRB1300. Tento robot dokáže dosáhnout polohové přesnosti až 0,02 mm s opakovatelností 0,03 mm. [27]

### *910SC – SCARA*

Nejrychlejší z robotů na pracovišti je robot SCARA (viz Obrázek 22). Úlohy tohoto robotu budou zejména montáže, které vyžadují velkou přesnost a jsou realizovány pouze ve vertikálním směru.



*Obrázek 22 - Robot SCARA IRB910SC*

SCARA má dosah 0,65 m, přičemž dosahuje nosnosti až 6 kg. Díky kinematice robotu dokáže dosáhnout polohové přesnosti až 0,01 mm s opakovatelností 0,01 mm. [28]

#### *IRB14000 – YuMi*

Dvouramenný kolaborativní robot se sedmi osami (viz Obrázek 23). Robot na pracovišti může pracovat jak v kolaborativním, tak automatickém režimu. Jeho uplatnění bude zejména u drobných montáží.



*Obrázek 23 - Kolaborativní robot YuMi IRB14000*

Dosah ramene kolaborativního robotu je 0,559 m, přičemž je nosnost ramene 0,5 kg. Zde je ovšem nutné brát v potaz, že minimálně polovina nosnosti ramene je spotřebována na nástroj. Robot je schopný dosáhnout polohové přesnosti až 0,02 mm s opakovatelností 0,02 mm. [29]

### 3.3 Nástroje robotů

Na pracovišti je možnost automatické výměny nástroje. Je jím vybaven robot IRB1300 a robot 910SC. Každý z robotů má čtyři pozice pro nástroje. Jedná se o pneumatické výměníky od výrobce Zimmer (viz Obrázek 24). Robot YuMi možnost výměny nástroje nemá, avšak je vybavený kolaborativními grippery.



Obrázek 24 - Volná část výměníku (označení. WWR40L-B), ke kterému lze připevnit nástroj

### Nástroje s výměníky

Výměník WWR40F-B sestává ze dvou částí, a to fixní části, která je připevněna na flange robotu, a volné části ke které je připevněn nástroj. Spojení těchto částí je řešeno přes zámkové elementy. Tyto elementy jsou vytlačovány za účelem uzamknutí volné části výměníku. Působí na ně tlak, který vyvíjí dvojčinný pneumatický válec s pružinou. Ta zároveň zabraňuje samovolnému odpadnutí nástroje v případě selhání pneumatiky. Stojan s výměníky lze spatřit na (viz Obrázek 25).

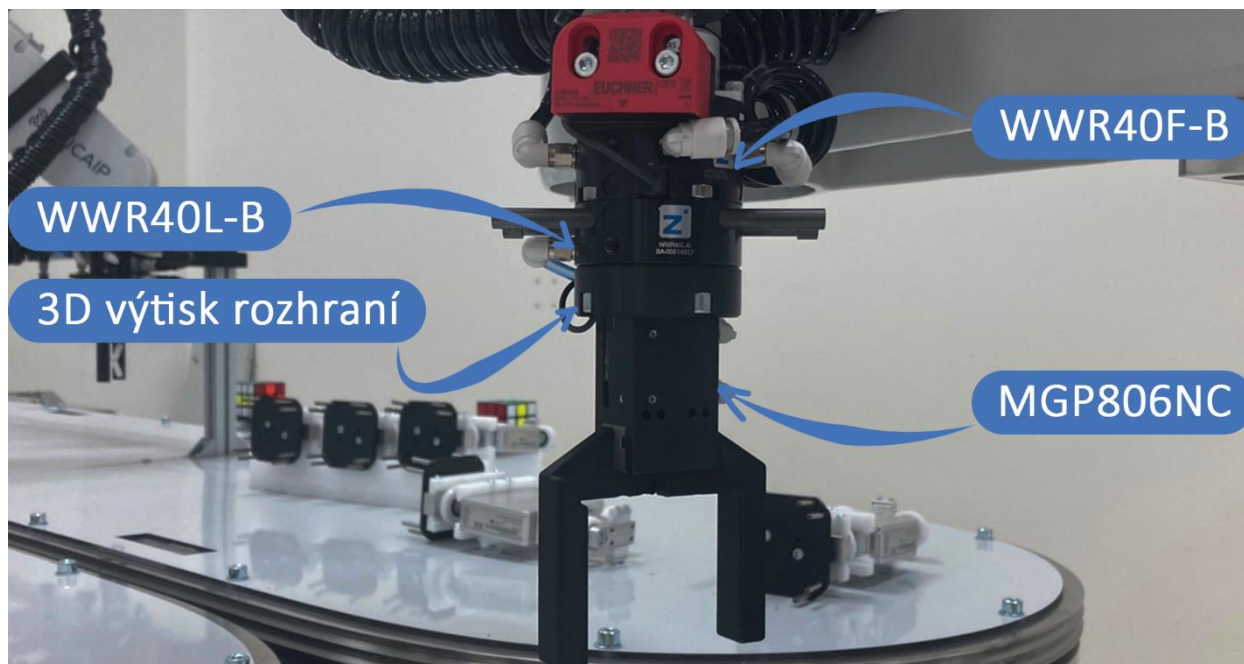


Obrázek 25 - Stojan s výměníky nástrojů pro robot IRB1300

Tyto výměníky umožňují manipulaci až s 20kg zátěží a dosahují velké opakovatelné přesnosti, která v případě spojení výměníku s nástrojem dosahuje v ose Z 0,01 mm a v osách X, Y 0,02 mm.

Na výměníky lze namontovat buď pneumatické anebo elektricky řízené nástroje s pomocí energetického elementu, který je nasazen na fixní a volnou část výměníku.

K pneumatickému výměníku jsou připevněny pneumatické grippery MGP806NC (viz Obrázek 26). Pneumatický gripper je ve výchozí poloze sevřen díky pružině. Smysl tohoto řešení spočívá v tom, aby v případě ztráty tlaku nedošlo k rozevření čelistí a ztrátě manipulované součásti. Síla, kterou dokáže gripper svírat, činí 140 N.



Obrázek 26 - MGP806NC u robotu IRB910SC

Aby čelisti byly rozevřeny, je potřeba přivést tlak o velikosti 4 barů. Celkový zdvih gripperu poté činí pouze 12 mm. Díky takto krátké dráze je rychlost rozevírání čelistí pouze 0,04 sekund. [30] [31] [32] [33] [34] [35]

#### *Kolaborativní gripper*

Robot YuMi je vybaven kolaborativními grippery zvanými Smart Gripper (viz Obrázek 27). Tyto Grippery váží 215 g bez prstů. Gripper je polohován s pomocí servopohonů, jehož celkové rozevření čelistí činí 50 mm. Gripper umožňuje manipulovat s předměty o hmotnosti 270 g při síle sevření 20 N. [29]



Obrázek 27 - Smart Gripper

### 3.4 Dopravník ACOPOStrak

Poslední důležitou komponentou pracoviště je dopravník ACOPOStrak (viz Obrázek 28), který není zajímavý jen svým vzhledem, ale i svou funkcionalitou. Jedná se dopravník, který se skládá z tratě a lineárních pohonů nazývaných také jako shuttles. Lineární pohony mají tu schopnost, že se dokáží v provozu dynamicky přepínat mezi jednotlivými tratěmi, aniž by došlo k jejich zpomalení. To umožňuje vytvořit flexibilní řízení výroby.

Trať je poskládána z různých segmentů, což umožňuje sestavovat tratě různých tvarů a rozměrů. Na tomto pracovišti byly zvoleny oválové tvary. Dopravník je řízen s pomocí Motion knihovny přes PLC od B&R.

Jelikož se jedná o trať s lineárními pohony, přináší to tu výhodu, že je možné sledovat polohu jednotlivých vozíků. To umožňuje adaptivní řízení a přepínání mezi jednotlivými tratěmi na základě požadavků výroby. Lineární pohony mohou dosáhnout rychlosti až 5 m/s o maximálním možném zatížení 2 kg. Vozíky lze spatřit položené na obrázku (viz Obrázek 28). [36]

Ted', když byly představeny klíčové prvky robotické buňky, lze se uchýlit již k návrhové části práce, a to návrhu předmětu montáže a potřebných komponent, které jsou potřebné pro jeho implementaci do výrobní buňky.



Obrázek 28 - Dopravník ACOPOStrak

## 4 Předmět automatizované montáže

Součástí diplomové práce je návrh předmětu montáže, kterým bylo zvoleno auto na dálkové ovládání. U něj je předpoklad ba dokonce i požadavek, aby bylo smontovatelné výhradně s pomocí robotiky. Cílem tohoto návrhu je zároveň vytvořit takový model, který by mohl v budoucnu demonstrovat distribuovanou výrobu v laboratoři Testbed pro průmysl 4.0 v CIIRC. Proto bylo zapotřebí brát zřetel na smontovatelnost, jednoduchost a jednoznačnost. Předpokladem pro vznik vozu je výroba s pomocí 3D tisku a v tuto chvíli, kdy práce vznikala se jedná o první tisknutelný prototyp (viz Obrázek 29), který je již druhou iterací celého návrhu. První iteraci lze spatřit (viz Obrázek 30 - První iterace prototypování auta na dálkové ovládání) Avšak níže popsané přípravy jsou první iterací. A s ohledem na vyvíjející se výrobní postup a požadavky napříč robotickými buňkami se zcela jistě budou měnit.

Jak již bylo zmíněno, cílem navrhování je vytvořit takový předmět montáže, který bude demonstrovat distribuovanou výrobu. To s sebou přináší několik zásadních požadavků, které je potřeba zvážit:

Mezi tyto požadavky patří:

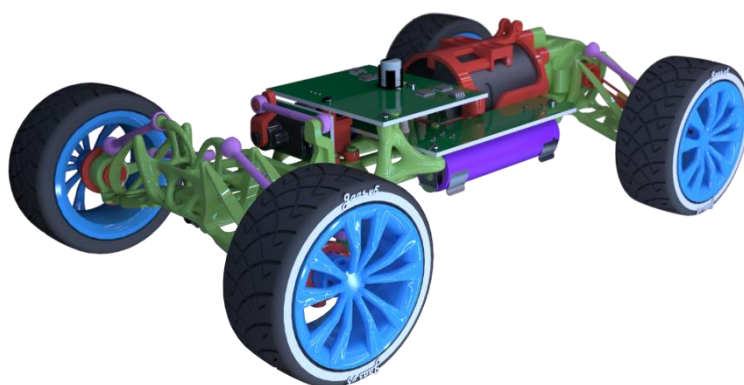
- Demonstrace použití 3D tiskárny pro sestavy, které jsou tištěné již v jednom kuse tzv. „print in place“, a možnosti tisku obtížných dílů
- Použití poddajných mechanismů
- Model musí obsahovat šroubové spoje
- Musí být co nejjednodušší pro složení, tudíž u elektroniky je nutné vyvarovat se kabeláží
- Do montáže musí být zapojeno co nejvíce robotů napříč laboratoří Testbed pro Průmysl 4.0
- Díky propojení 3D tisku s elektronikou vznikne funkční model
- Cena modelu by měla být co nejnižší
- Model by měl být opětovně snadno rozebíratelný pro opakovanou demonstraci
- „Fun factor“ výsledného prototypu musí být co nejvyšší.

Důležitými parametry výsledného produktu jsou také rozměry. Délka vozu je 253 mm a šíře činí 163 mm.

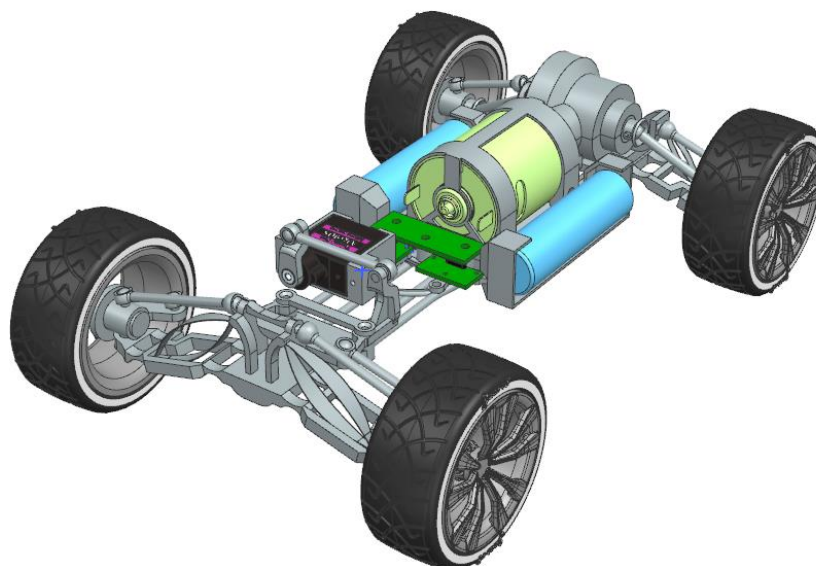
Návrh prototypu je cílen na využití spojovacích mechanismů, které můžeme v angličtině nalézt pod názvem „snap-fit design“. Díky tomuto mechanismu vznikne samonosný produkt, u kterého lze zajistit po dobu montáže a přesunu výrobku (či jeho částí) mezi jednotlivými pracovišti soudržnost, aniž by bylo potřeba používat pomocné prvky ve formě upínek, či svěráků, které přidávají během montážního procesu minimálně krok navíc. Šroubováním se v poslední fázi montážního procesu zajistí trvanlivé spojení k sobě připojených dílů.

Uchycení elektroniky je řešené podobným způsobem jako výtisky auta. U elektroniky byl potřeba vymyslet takový design, který efektivně zastoupí kabeláž. Tímto řešením jsou elektronické

kontakty, které jsou přichyceny k PCB deskám neboli „tišťáku“. Design elektroniky je dílem kolegů z distribuovaných systémů v CIIRC. Zjednodušeně řečeno: kontakty propojují jednotlivé desky, na jejichž koncích se vždy nachází nějaká řízená komponenta. Může se jednat například o motor. Hlavní roli zde pak bude hrát celková tuhost konstrukce a přítlačná síla kontaktů, která bude po dobu životnosti mechanismu neměnná. Pokud bude síla a tuhost konstrukce malá, může dojít k přerušení spoje. Naopak bude-li síla velká, mechanismus se poškodí. Je zde daleko více problémů, které je potřeba řešit jako je například i bezpečnost, ochrana apod.



Obrázek 29 – Výsledný (zatím poslední prototyp) auta na dálkové ovládání



Obrázek 30 - První iterace prototypování auta na dálkové ovládání

#### 4.1 Popis předmětu

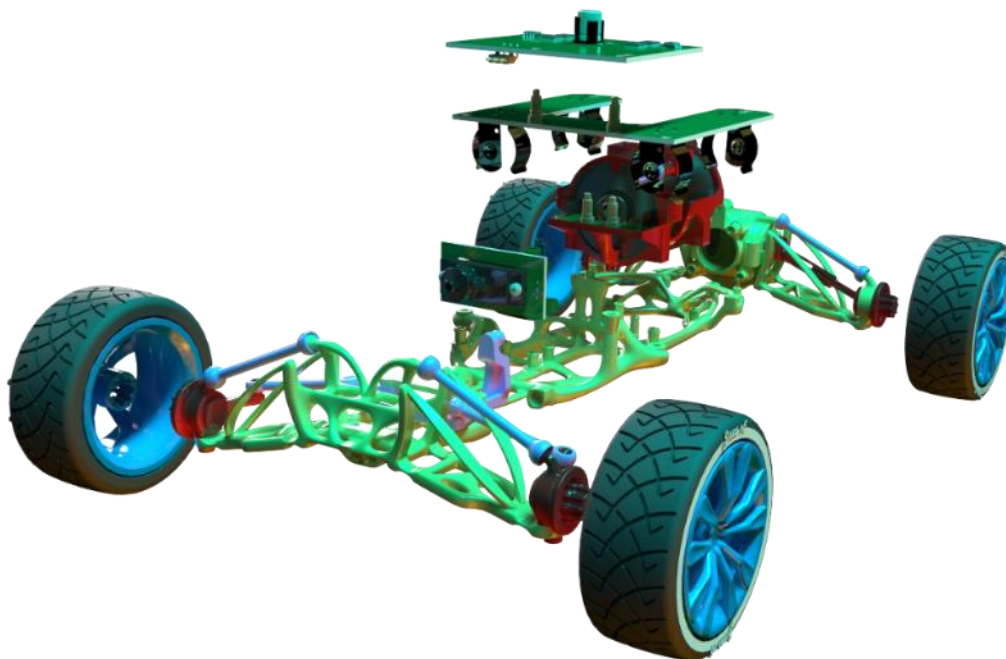
Prototyp auta se skládá jak z dílů nakoupených, tak z dílů vyrobených v laboratořích Testbedu. A to většinou vyrobených s pomocí 3D tiskáren. V tabulce níže (viz Tabulka 1) jsou vypsané díly, které budou pro realizaci prototypu auta potřeba.

Tabulka 1 - Kompletní kusovník

<b>díl</b>	<b>Počet</b>	<b>původ</b>	
<b>Rám</b>	<b>1</b>	3D tisk PA12	
<b>Přední náprava</b>	<b>1</b>	3D tisk	PA12
<b>Zadní náprava</b>	<b>1</b>	3D tisk	PA12
<b>Disky</b>	<b>4</b>	3D tisk	PA12
<b>Motor - domek</b>	<b>1</b>	3D Tisk	PA12
<b>Servo - domek</b>	<b>1</b>	3D Tisk	PA12
<b>Servo ramínko</b>	<b>1</b>	3D Tisk	PA12
<b>Pastorek</b>	<b>1</b>	3D tisk	PA12 či 17H4
<b>Diferenciál velké kolo</b>	<b>1</b>	3D tisk	PA12 či 17H4
<b>Diferenciál planeta</b>	<b>1</b>	3D tisk	PA12 či 17H4
<b>Diferenciál oběžná kola</b>	<b>2</b>	3D tisk	PA12 či 17H4
<b>Pneumatiky</b>	<b>4</b>	3D tisk TPU	TPU
<b>Servomotor</b>	<b>1</b>	Zakoupený	-
<b>Motor</b>	<b>1</b>	Zakoupený	-
<b>PCB deska - motor</b>	<b>1</b>	Zakoupené	-
<b>PCB deska - servo</b>	<b>1</b>	Zakoupené	-
<b>PCB deska - driver</b>	<b>1</b>	Zakoupené	-
<b>PCB baterie</b>	<b>1</b>	Zakoupené	-
<b>Šroub - kola</b>	<b>4</b>	Zakoupené	-
<b>Šroub – nápravy</b>		Zakoupené	-
<b>Šroub - PCB</b>		Zakoupené	-
<b>Šroub - diferenciál</b>		Zakoupené	-
<b>Ložiska diferenciálu</b>	<b>2</b>	Zakoupené	-

V závislosti na úspěšnosti prototypování je také počítáno s variantou, kdy bude diferenciál vytištěn jako jeden celek. Dojde tak k velkému zjednodušení celé sestavy a teoreticky tak může splynout zadní náprava s diferenciálem v jeden celek. Díky tomu by opadla i drobná montáž diferenciálu a vymizel by tak jeden montážní krok.





*Obrázek 31 - Rozložený prototyp auta na dálkové ovládní*

Předchozí zmíněná tabulka poukazuje na celkovou skladbu vozidla. V následující popsané tabulce (viz Tabulka 2) jsou vstupy, se kterými se bude v robotické buňce pracovat. Pracuje se zde zejména s podsestavami. Na obrázku (viz Obrázek 31) lze vidět rozložené vozidlo.

Pro tyto díly je potřeba s ohledem na robotické operace vhodně navrhnout přípravky. Samotná obtížnost montáže se odvíjí od návrhu přípravků, které by během montáže neměly překážet, ale naopak by měly montáž usnadňovat. Avšak návrhu přípravků by měl předcházet výrobní postup, který stanoví hranice a požadavky.

*Tabulka 2 - Kusovník*

Označení	Předmět	Kusů	Náhled
A1	Rám	1	
A2	Přední náprava	1	
E1	PCB deska - Baterie	1	
E2	PCB deska - Driver	1	
A3	Zadní náprava- podsestava	1	
B1	Kola - podsestava	4	
M1	Motor- podsestava	1	
M2	Servo- podsestava	1	

## 4.2 Montážní postup

Pro prototyp auta byl vytvořen i obecný montážní postup, který je k vidění na konci této podkapitoly (viz Tabulka 3). Montážní postup je zatím navržen pouze pro montáž jednoho auta v robotické buňce. Důvody, proč tomu tak je, budou vysvětleny v podkapitole „Dopravník“. V robotické buňce budou probíhat operace Pick&Place. To znamená, že robot odebírá z místa A díly, které posléze vloží na místo B. Před návrhem přípravků je potřeba mít dobře rozmyšlené způsoby montáže a rozložení dílčích operací mezi jednotlivé roboty. Kinematiky robotů v buňce jsou totiž velice odlišné. Montážní proces byl proto navržen s ohledem na dispozice robotů. Vzhledem k tomu, že roboty IRB1300 a 910SC mají k dispozici i výměníky nástrojů až se čtyřmi pozicemi, je možné provádět vícero operací na jednom robotickém pracovišti.

Robot IRB1300 má ve srovnání s roboty na pracovišti velký dosah, proto jeho hlavní úlohou bude zejména zásobování linky dílci, ale také montáž nápravy.

Robot 910SC – Kinematika robotu umožňuje dosahovat velkých rychlostí a kvůli existenci pouze vertikálních os dokáže robot dosahovat i velké přesnosti a opakovatelnosti. Proto úloha, která je robotu přidělena, je montáž elektronických komponent.

Robot YuMi Má k dispozici dvě robotická ramena, která lze pro urychlení montážní činnosti použít na osazování náprav koly.

Robot IRB1300 bude mít za úkol vyndávání přípravků z regálu (který přiveze AMR do robotické buňky) na shuttle dopravníku (dále jen vozík) s pomocí nástroje neboli gripperu, který bude mít prsty navržené pro tento účel. Nástroj bude dále označován jako T1.

Poté, co budou všechny vozíky naloženy, dojde k jejich přesunu na definovaná stanoviště po definované trati oválů. Prvním definovaným stanovištěm je montáž elektroniky na rám vozidla s pomocí robotu 910SC, který je také jinak nazýván SCARA robot, a druhým stanovištěm je robot YuMi.

Robot 910SC na prvním stanovišti bude nejdříve přendávat z vozíku s elektronikou motory na vozík s rámem vozidla. To bude realizováno nástrojem označovaným jako T1 pro robot 910SC. Posléze dojde k výměně nástrojů, aby mohla proběhnout montáž PCB desek s pomocí nástroje T2. Poté, co dojde k dokončení montáže, robot vymění svůj nástroj na T1 a vyčká na dokončení činnosti ostatních pracovišť.

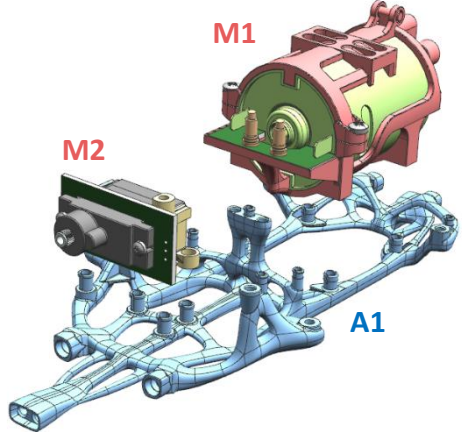
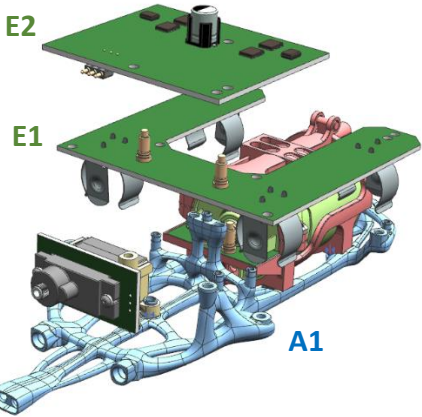
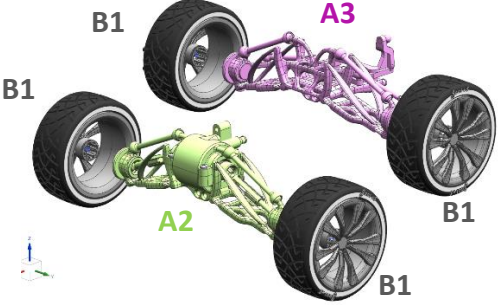
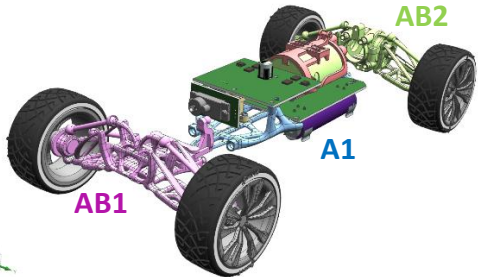
Na druhém stanovišti Robot YuMi nejdříve převezme z vozíku přípravek s díly a položí jej na pracovní plochu před sebe. Zde totiž dosáhne oběma rameny do pracovního prostoru. Robot provede montáž kol oběma rameny současně. Jakmile je tato činnost dokončena, přendá opět přípravek zpět na vozík a vyčkává na další pokyny.

Vozíky se dají do pohybu tehdy, kdy byla dokončena veškerá montážní činnost a vrací se zpět na startovní pozici k robotu IRB1300. Tento robot si již předtím, než vozíky vyrazily na montážní stanoviště, vyměnili nástroj, který slouží k manipulaci s nápravami vozidla. Tento nástroj je označen jako T2. Po příjezdu vozíků robot nejdříve odjistí zámek u přípravku s nápravou, aby mohl zahájit jejich odběr. Nápravy poté vkládá na přípravek s paletou základního rámu.

Poslední fází je odběr přípravků z dopravníku, které jsou vráceny zpět do regálu. To vše je provedeno robotem IRB1300, který pro tento účel musí zaměnit svůj nástroj zpět na T1. Od této chvíle je celý cyklus opakován.

Nyní, když byla vytvořena rozvaha nad výrobním postupem, lze přejít k samotnému návrhu přípravků a uchopovacích prvků.

Tabulka 3 – Obecný montážní list

Číslo operace	Popis činnosti	Náhled
10	Uložení dílu (M1) a (M2) na zacvakávací kolíky (A1).	
20	Uložení dílu (E1) a (E2) na zacvakávací kolíky (A1).	
30	Díly (B1) jsou nasazeny na díly (A2) a (A3). Vzniknou tak podsestavy (AB1) a (AB2)	
40	Podsestava (AB1) je nasazena na přední část dílu (A1). Podsestava (AB2) je nasazena na zadní část dílu (A1).	

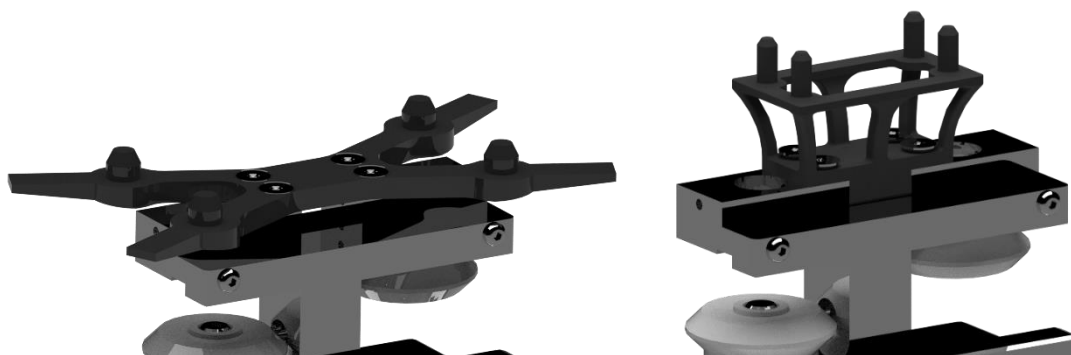
### 4.3 Přípravky

Moderní dopravník ACOPOStrak umožňuje vytvořit takový návrh přípravků, které budou moci cestovat na dopravníku s díly. Tyto přípravky budou dále nazývány jako paletky. Nebudou zbytečně přemísťovány díly po jednom kusu na dopravník, ale lze je vložit všechny najednou na vozík, který slouží jako transportní element na dopravníku (dále jen „trati“). Zároveň může být na trati realizována i montáž, je-li to v možnostech dosahu robotu.

S touto úvahou bylo dosaženo takového návrhu, který požaduje pro realizaci montáže tři vozíky: Vozík pro transport paletky se základním rámem, vozík pro transport paletky elektroniky a vozík pro transport paletky náprav a pneumatik. Tímto způsobem dojde k vhodné distribuci dílů mezi jednotlivá pracoviště se vším, co roboty potřebují pro zahájení své činnosti.

Nutno dodat, že bylo také uvažováno řešení, kdy by všechny potřebné díly pro montáž byly narovnané na jednu paletku, která by byla rozebíratelná. Ovšem toto řešení působilo problematicky zejména v rámci stability vozíku a jeho nákladu, ale také v technické a konstrukční složitosti, která ve fázi prototypování není žádoucí.

Ukládání paletek na vozíky bylo vyřešeno s pomocí rámečku s kuželovými trny, který se přišroubuje k vozíku (viz Obrázek 32). Rámeček pro nosič základního rámu je ovšem odlišný od zásobovacího rámečku. Důvod k tomu je zcela prostý. U tratě je stanovena minimální výška, ve které mohou být uloženy vozíkem nesené dílce, aby nedošlo ke kolizi mezi tratí a přípravkem. Jelikož paletky se zásobami jsou uvažovány jako ploché, mohou tak být posazeny níže. Na rozdíl od toho paletka s rámem potřebuje dodatečný prostor pod sebou kvůli montáži nápravy a samotné charakteristice paletky, která bude popsána dále.



Obrázek 32 - Lože pro přípravky vlevo – zásoby, vpravo rám

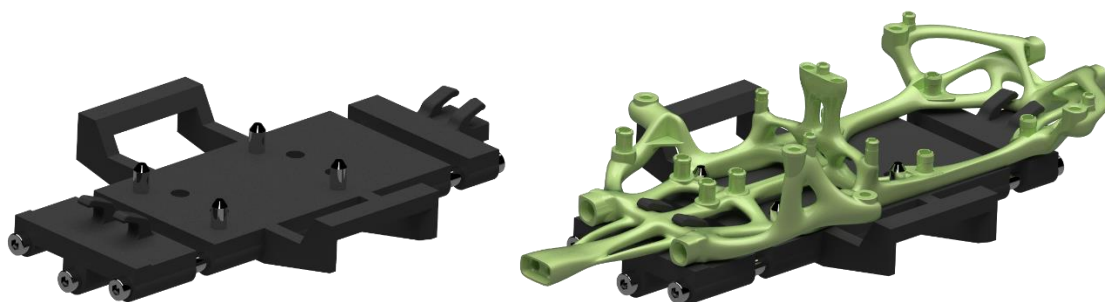
Zcela jistě by šel vytvořit universální rámeček pro všechny paletky, ale skutečnost, že vozík s rámem kolem sebe potřebuje více místa byla bohužel odhalena, až když bylo vše navrženo. Bylo tedy jednodušší předělat jeden rámeček než vše ostatní a rámeček. Avšak tento nedostatek chod linky nijak neovlivní, zejména ve fázi prototypování.

V neposlední řadě je potřeba u všech palet zajistit stejný tvar uchopení, který zajistí bezproblémovou a jednoznačnou manipulaci. Byl zvolen obdélníkový tvar uch se zkosenými hranami. Na každé z palet musí být rozteč uch identická, jelikož gripper, kterým je robot vybaven, nemá příliš velkou schopnost sevření/otevření čelistí, činí pouhých 12 mm. To značně omezuje variabilitu návrhu. Návrhy prstů budou podrobněji popsány v podkapitole „Návrh uchopovacích prvků“.

Všechny prototypy přípravků a uchopovacích prvků jsou navrženy pro výrobu s pomocí 3D tisku. Jedná se tak o nejrychlejší a nejlevnější způsob prototypování a ověření funkčnosti.

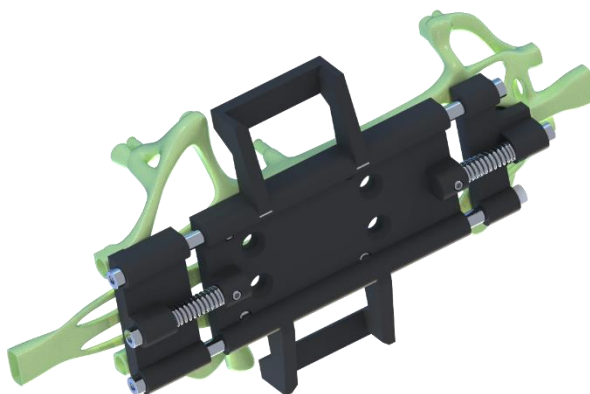
#### *Paletka pro základní rám*

Tato paletka slouží k uchycení základního rámu vozidla na dálkové ovládání (viz Obrázek 33). Jedná se o elementární prvek celé montáže, jelikož rám zůstane k paletce přichycen od samého začátku až do konce montážního procesu. Tedy i mimo robotickou buňku ABB. V robotické buňce CIIRCtrack bude probíhat montáž elektroniky a montáž přední a zadní nápravy.



Obrázek 33 - Paletka pro základní rám

Paletka má opěrné kolíky, které definují přesnou polohu rámu vozidla, které ovšem nefixují rám, aby v průběhu montáže nedošlo k jeho pohybu. Proto byla paletka navržena jakožto rozpínající se svěrák s háčky, které kopírují tvar rámu. Rozpínací sílu tvoří tlačné pružiny (viz Obrázek 34), kde každá z pružin působí na rám silou 45 N.

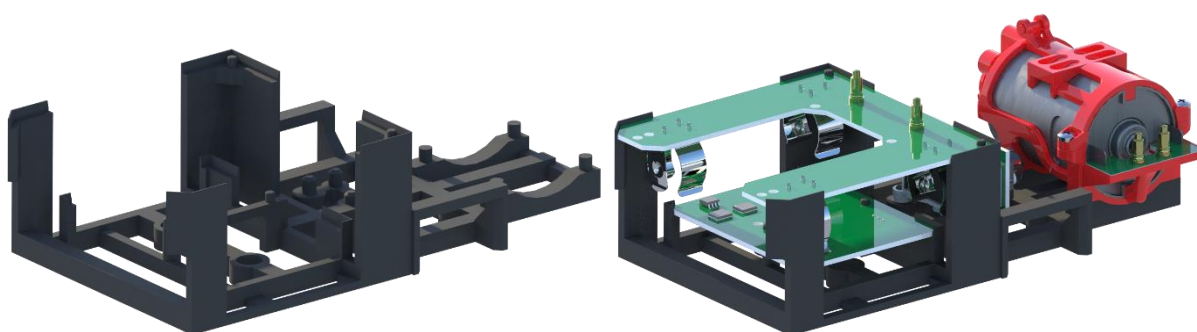


Obrázek 34 - Detail na rozpínací systém paletky

Tato paletka byla navržena s ohledem na možnost automatizovaného zakládání rámu do paletky a jeho opětovného vyjmutí. Svěrák stačí z každé strany stlačit o 6 mm, aby došlo k úplnému uvolnění rámu.

#### *Paletka pro elektroniku*

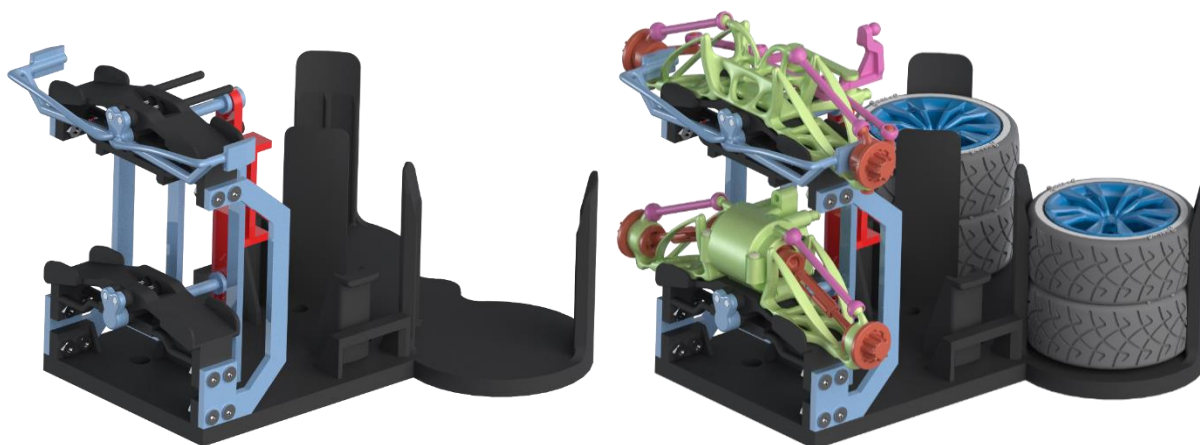
Tato paletka slouží jako nosič dílů potřebných pro montáž elektroniky pro robot 910SC, kterou bude servomotor, motor, PCB deska driveru a baterie (viz Obrázek 35). Jedná se tedy o mobilní odběrové místo, které se po odebrání všech komponent vrátí na výchozí stanoviště. Tam bude prázdná paletka vyměněna za plnou.



Obrázek 35 - Paletka pro převoz elektroniky

#### *Paletka pro přední a zadní osu*

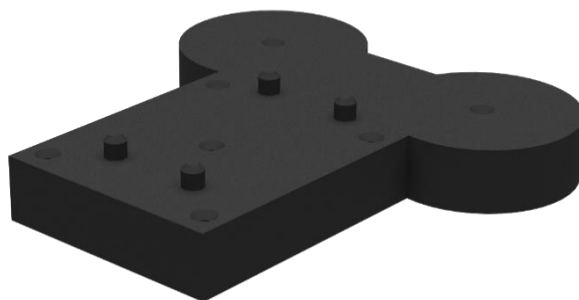
Jedná se o univerzální paletku, která slouží zároveň jako nosič dílů, ale zrovna tak i jako montážní stůl (viz Obrázek 36). Tím dojde k redukci času, který by za jiných okolností spotřebovala nadbytečná manipulace.



Obrázek 36 - Paletka pro přepravu pneumatik a os vozidla



Tato paletka přepravuje kola, přední a zadní nápravu. Nápravy jsou v přípravku fixovány kvůli požadavku montáže kol. Jelikož se jedná o zrcadlově identickou montáž, bude ji provádět robot YuMi, který má dvě robotická ramena. Paletku z vozíku bude přemísťovat před sebe, aby na paletku dosáhl oběma rameny. Paletku YuMi položí na přípravek (viz Obrázek 37), který zajistí opakovatelnost umístění a zároveň i fixaci.



Obrázek 37 - Podložka pod paletku pro robot YuMi

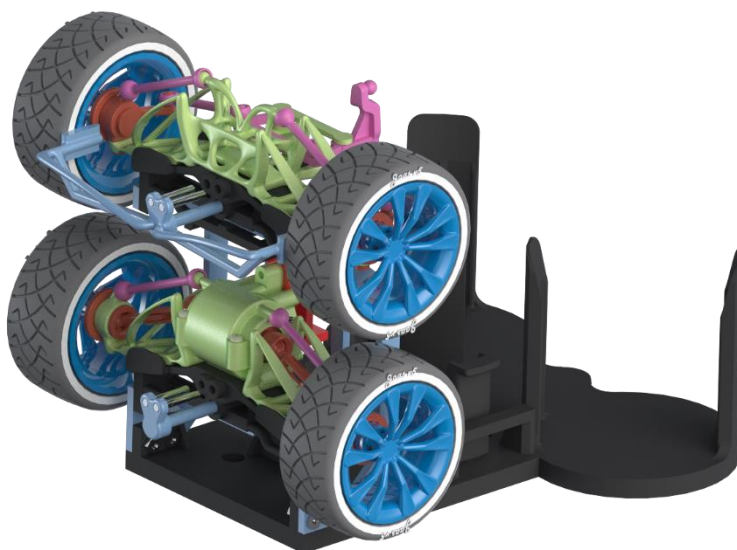
Vzhledem k tomu, že YuMi bude paletku přemísťovat z vozíku, nesmí paletka svou hmotností (a to včetně nákladu) přesáhnout 250 g. Tato hmotnost představuje maximální nosnost robotického ramene. Většina prvků je vyrobena z materiálu PA12, proto redukce hmotnosti pod tuto hranici nečinila problém. Paletku rovněž musí být robot schopen uchopit stejnými prsty, kterými bude provádět montáž kol, proto byla paletka opatřena úchopem, který slouží výhradně pro robota YuMi.

Z paletky budou však v další fázi odebírány nápravy robotem IRB1300. Proto je zde navržen takový mechanismus (viz Obrázek 38), který dokáže odjistit robot jednoduchým pohybem. V prvním kroku robot nadzvedne zámek směrem nahoru a ve druhém na něj mírně zatlačí ve vodorovném směru. Tím dojde k uvolnění náprav (viz Obrázek 39).



Obrázek 38 - Mechanismus pro odemčení os na zadní části paletky

U této paletky bylo nejdříve také pracováno s variantou, že by nápravy byly rovnány vedle sebe, ale z hlediska rozložení a velikosti paletky se to později ukázalo jako neefektivní řešení. Odběr z paletky robotem IRB1300 by byl o něco složitější. Návrh prototypu paletky lze nalézt v příloze.



Obrázek 39 - Uvolněné osy

#### 4.4 Návrh uchopovacích prvků

Uchopovací prvky, nebo také jinak řečeno prsty, byly pro každý z robotů navrženy na základě jejich úlohy. Tím bude zaručeno přesné a jednoznačné uchopení manipulovaného předmětu.

V robotické buňce se typově nacházejí dva druhy paralelních gripperů. Jeden typ je pro roboty IRB1300 a 910SC, které jsou vybaveny pneumatickým gripperem Zimmer MP806NC a Smart Gripper, kterým jsou vybaveny robotická ramena robota YuMi. Gripperery byly popsán v rešeršní části práce.

U pneumatického gripperu Zimmer je potřeba pracovat s malou schopností rozevření paralelních prstů. Je to jeden z důvodů, proč každý z robotů s tímto gripperem je vybaven dvěma nástroji. Nástroje byly navrženy na základě co největší podobnosti součástí, aby bylo možné s vybaveným nástrojem provést co možná nejvíce montážních úkonů.

Naopak tomu je u robota YuMi, který je vybaven Smart Gripperem. Tento gripper disponuje výrazně větším sevřením, a dokonce i schopností polohování prstů, která u pneumatického gripperu není možná. Limitem je zde nízká nosnost robotického ramene, která je díky hmotnosti samotného Smart gripperu snížena na polovinu a to 250 g.

V následujících podkapitolách budou popsány jednotlivé návrhy prstů pro každý z robotů.

##### *Uchopovací prvky robotu IRB1300 pro nástroj T1*

Tento robot bude provádět manipulaci paletek, pro které byl navržen nástroj označovaný jako T1. Tento nástroj (viz Obrázek 40), nebo přesněji uchopovací prsty, mají tvar přizpůsoben pro uchopení paletek.

Prsty mají zkosené hrany, aby v případě nepřesného polohování či nepřesného zastavení vozíku došlo ke vklouznutí do otvoru uchopovacích uch. Zároveň to usnadní programovací práci při zacházení se skutečným robotem.



Obrázek 40 - Prsty pro robot IRB1300 s ukázkou uchopení paletky

Hrana, která vytváří podpěru pod uchopenou paletou, musí být dostatečně široká, ale ne tak, aby působila jako překážka při najíždění robota do uchopovací polohy. Žádný z robotů nemá nekonečnou přesnost a mohlo by tak docházet ke kolizi s paletkou. Samozřejmostí jsou i vůle, které zabrání zadrhání mezi prsty a uchopovanou paletkou.

#### *Uchopovací prvky robotu IRB1300 pro nástroj T2*

Další nástroj, který má robot ve výbavě, je označován jako T2. Tento nástroj (viz Obrázek 41) slouží k odemykání náprav a jejich manipulaci. Na prsty lze pohlížet jako na vidle, které se pro uchopení součástí sevrou.



Obrázek 41 - Nástroj T2 pro IRB1300

Manipulované součásti jsou docela tvarově složité a k tomu i tvarově odlišné. Dále je kolem nich prostorové omezení pro manipulaci kvůli jejich uložení. Byla zde hledána tvarová podobnost,

kteřá by umožnila stejné uchopení u obou součástí. Podobnost byla nalezena u ramen a středové části nápravy. Z tohoto důvodu jsou prsty ve tvaru vidličky.

Tento tvar je zejména vhodný při montáži nápravy na rám vozidla na dálkové ovládní. Náprava nemá kam uhnout a během montáže je podepírána z přední a zadní strany.

#### *Uchopovací prvky robotu 910SC pro nástroj T1*

Tento nástroj (viz Obrázek 42) je určen k manipulaci s elektronikou, která bude k robotu přepravena na paletce. Návrh prstů, ale i návrh manipulovaných součástí, byl vzájemně přizpůsoben, aby nebylo potřeba pro každý z manipulovaných motorů vytvářet nástroj zvlášť. U obou manipulovaných dílů byla vytvořena pojistná hrana proti vyklouznutí a přizpůsobená šíře dílu, aby byly z pohledu manipulace identické.



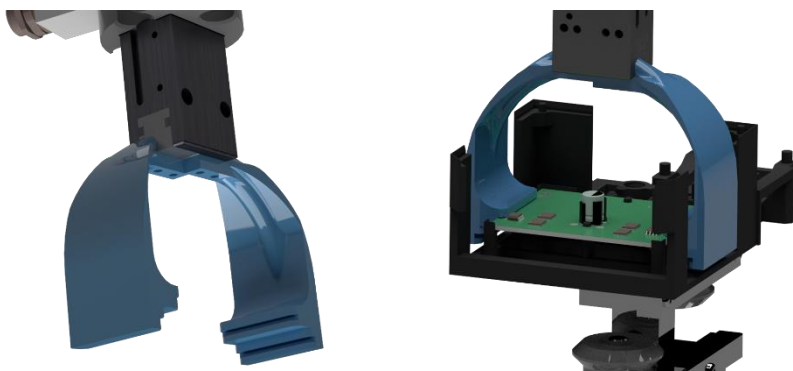
*Obrázek 42 - Nástroj T1 pro robot 910SC*

Oba manipulované díly kvůli jejich požadavku na pevnější ukotvení potřebují větší sílu na přicvaknutí k rámu. Z tohoto důvodu byla u jednoho z prstů vytvořena rozšiřující plocha, která pomůže lepší distribuci přitlačné síly od robotu.

#### *Uchopovací prvky robotu 910SC pro nástroj T2*

Pro manipulaci PCB desek byly navrženy prsty, které dokáží desku vzít ze stran (viz Obrázek 43). Desky, kterými je manipulováno, mají různé rozměry. Opět, aby nebylo pro každou z desek nutné navrhovat speciální prsty, tak tomu byly prsty přizpůsobeny. Rozteč prstů byla navržena na základě největší desky. Od tohoto rozměru byl poté navržen schod, který kompenzuje velikostní rozdíl desky.

Prsty v sobě mají drážku, do které zapadne PCB deska. Tím je deska jištěna ze všech stran a nemůže tedy vyklouznout ani se vzpříčit. Zároveň rozteč mezi drážkou velké desky a drážkou malé desky nemůže být příliš velká, aby prsty nekolidovaly při odběru menší desky s paletkou.

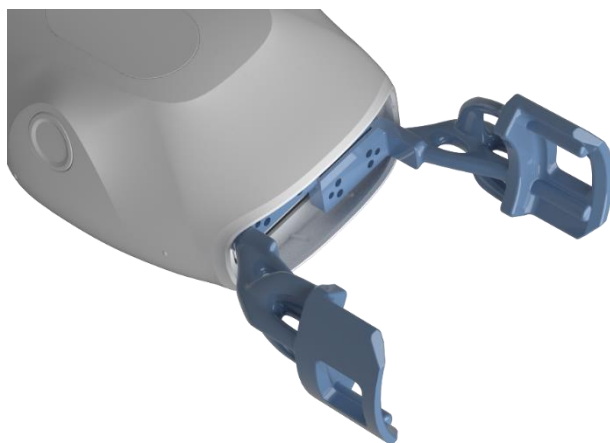


Obrázek 43 - Nástroj T2 pro robot 910SC

#### *Uchopovací prvky robotu YuMi*

Prsty jsou přizpůsobeny tvaru manipulovaného předmětu (viz Obrázek 44), kterým jsou kola auta na dálkové ovládní. Důležité při návrhu bylo, aby kolo nemělo tendenci vyklouznout díky svému tvaru mezi prsty či aby se při montáži nepříčilo.

To vše se dalo vyřešit opěrnou hranou, miskovitým tvarem a zajištěním dvoubodového styku kola s pneumatikou na každém z prstů. Prsty byly navrženy poměrně krátké, respektive jejich uchopovací část, aby nedocházelo během manipulace ke kolizi a také k odběru více než jedné pneumatiky.



Obrázek 44 - Prsty pro uchopení pneumatik pro robot YuMi

Důležité je dbát na hmotnost prstů, která by vedla k dalšímu snížení nosnosti a obratnosti ramene. Vzhledem k typu montáže a hmotnosti manipulovaných předmětů není nutné vytvářet nadbytečně masivní prsty.

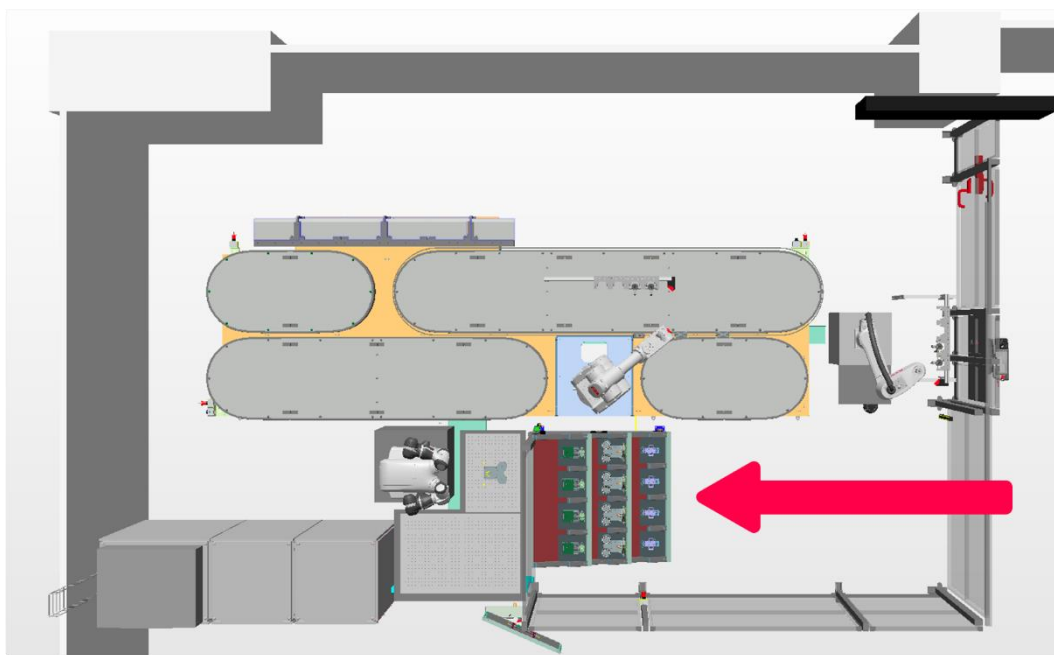
Robot YuMi je robotem kolaborativním a zrovna tak i jeho smart grippery, kterými je vybaven, jsou certifikovány jako kolaborativní. Z tohoto důvodu bylo dbáno na to, že jsou prsty zahlazené a bez ostrých hran, které by v případě kolize s člověkem během kolaborativní montáže mohly vést ke způsobení nechtěného úrazu.

V této kapitole byly provedeny návrhy veškerých přípravků a uchopovacích prvků, které budou následně během montáže používány. Vše bylo navrhováno za účelem produkce s pomocí 3D tisku, který umožňuje vytvářet snadno a rychle složité tvary s poměrně velkou přesností, která je zcela dostatečná pro manipulaci. Lze dokonce v případě úspěšného designu počítat i s velkou životností výtisků, a to zejména při použití technologie Binder Jetting a materiálu PA12, kdy vzniká homogenní struktura výrobku.

Teď, když jsou všechny návrhové a modelovací fáze dokončené, je načase přesunout se k další kapitole, jejíž obsah se bude zabývat přípravou simulačního modelu robotické buňky.

## 5 Simulační model

Tato kapitola se bude zabývat přípravou simulačního modelu robotické linky v softwaru RobotStudio od ABB. Výsledkem bude virtuální prostředí, které bude vizuálně zrcadlit prostředí skutečné robotické linky (viz Obrázek 45). Model bude sloužit k odhalení potencionálních hrozeb a nedostatků, které by za normálních okolností byly odhaleny až během výstavby a zprovoznění skutečné robotické buňky. Díky tomuto modelu lze otestovat, zda robot například dosáhne do pracovního prostoru či zda někde nehrozí kolize s okolním prostředím. Vizualizaci výstupu příprav lze spatřit na obrázku (. Obrázek 49).



Obrázek 45 - Schéma pracoviště. Červená šipka označuje vjezd pro AMR.

Pro vznik simulačního modelu je jednou z nejdůležitějších částí import 3D dat, která zpravidla vznikají v nějakém CAD softwaru. Zde je potřeba modely vyexportovat ve formátu ACIS, který je poté možné naimportovat do RobotStudio.

Pro zjednodušení práce je lepší exportovat hotové sestavy z CAD modelu. Vždy je také dobré provést v CAD softwaru nadefinování počátku sestavy či dílu, jelikož tento počátek je posléze naimportován se součásti i do RobotStudio. Počátek součástí lze měnit, ale je to zbytečná práce. Důvod, proč je lepší importovat sestavy a hotová pracoviště z CAD softwaru, je prostý. V CAD softwaru je daleko jednodušší provádět manipulaci a polohování jednotlivých objektů než v softwaru RobotStudio, který je určen primárně pro simulace, nikoliv pro modelování a sestavování.

### 5.1 Robotická linka

Protože návrh a konstrukce robotické linky byly objednány na zakázku, byly před výrobou vypracovány i CAD modely, které byly použity pro vytvoření simulačního modelu. Sestava modelu linky byla tedy od společnosti ABB na vyžádání obdržena.

Prvky, které bylo potřeba k robotické lince domodelovat a dodat, byly stojany na výměníky nástrojů pro roboty IRB1300 a 910SC, které byly navrženy z hliníkových profilů, které bylo potřeba napolohovat v RobotStudio.

### 5.2 Import robotů

Roboty byly vybírány z knihovny robotů v softwaru RobotStudio, se kterými si je rovnou možné naimportovat i simulovaný kontrolér. Robot se tak bude chovat jako skutečný. Nejen co se softwaru týče, ale také fyzikálních vlastností, které lze ze simulace později vyčíst.

Z knihovny byly vybrány stejné roboty, které se nachází na robotické lince. Ke každému z robotů byl rovnou naimportován i příslušný kontrolér, se kterým bude pracováno později v podkapitole „Instalace kontroléru“.

### 5.3 Modely objektů

Dalším krokem byl import vlastních modelů, kterými jsou přípravky, prsty či předmět montáže. Ale také modely předmětů, které bylo nutné zakoupit. Mezi nimi jsou pneumatické výměníky či pneumatický nástroj Zimmer. U všech modelů bylo provedeno v CAD softwaru nastavení nulového počátku.

Při exportování modelů bylo potřeba postupovat obezřetně u modelů, které jsou z více částí a nepůsobí vůči sobě staticky, jako například u gripperu s výměníkem, kde byl postup následující: V CAD softwaru byla vytvořena sestava kompletního výměníku s gripperem. Počátek sestavy byl definován ve středu flange výměníku. Jako první díl byl vyexportován výměník, dále bylo vyexportováno tělo gripperu a v poslední řadě prst levý s paralelním úchytem gripperu, a to samé pro pravý prst s paralelním úchytem.

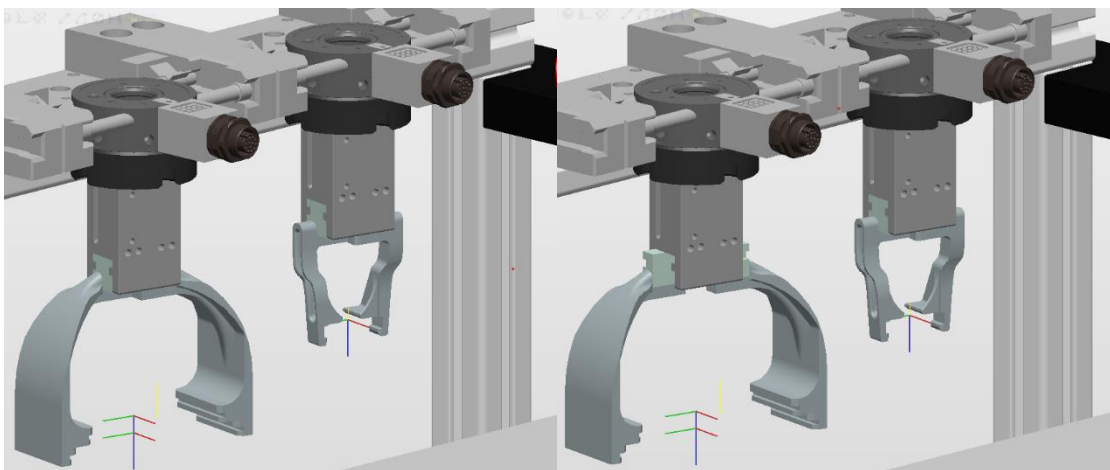
Dále už stačí pouze vyměňovat v CAD softwaru prsty a ty z této sestavy exportovat. Když se zvolí tento postup, je poté daleko jednodušší vytvořit mechanismus, který je popsán v podkapitole „Příprava mechanismů“, jelikož jak bylo zmíněno, tak s exportovanými daty je exportován i počátek modelu či sestavy. Prsty jsou tak rovnou naimportovány do RobotStudio přímo do těla gripperu. Dále, když bude v simulaci prováděno automatizované připínání nástroje, nástroj se připíná vždy svým nulovým bodem do souřadnicového systému flange robotu. Flange výměníku je pro simulaci pouhý připnutý model na konec robotu, se kterým dokáže počítat jen v rámci kolizí, ale to, že na konec tohoto modelu přijde ještě nějaký nástroj, netuší. Pokud bychom importovali nástroj bez tohoto odsazení, došlo by k prolnutí flange výměníku a připnutého nástroje, jelikož by nebylo počítáno s původním odsazením. Avšak dá se to dodatečně řešit posunem počátku nástroje v RobotStudio.



V poslední řadě modely, které bylo potřeba naimportovat, byly Smart gripperly pro robot YuMi. Tyto modely lze dohledat přímo v knihovně ABB v rámci RobotStudio.

#### 5.4 Příprava mechanismů

Jedná se zejména o přípravu gripperů, aby se chovaly jako skutečné. Tomu předchází ještě získání informace o TCP z CAD softwaru, kde jej lze snadno stanovit a změřit jeho konkrétní polohu, ideálně za přítomnosti součásti, se kterou bude v simulaci manipulovat. Jedná se o podstatné parametry, které je nutné v rámci vytváření nástroje vyplnit. TCP určuje střed nástroje. S pomocí tohoto parametru lze robot snadno polohovat při manipulaci s díly. Zároveň je to bod, skrze který robot vypočítává svou trajektorii. Kdyby TCP zůstal nulový, mohlo by dojít ke kolizi robotu s prostředím, protože by nepočítal s nástrojem při výpočtu trajektorie.



Obrázek 46 - Vytvořený mechanismus pro Gripperly IRB910SC, který umožňuje fungování gripperů

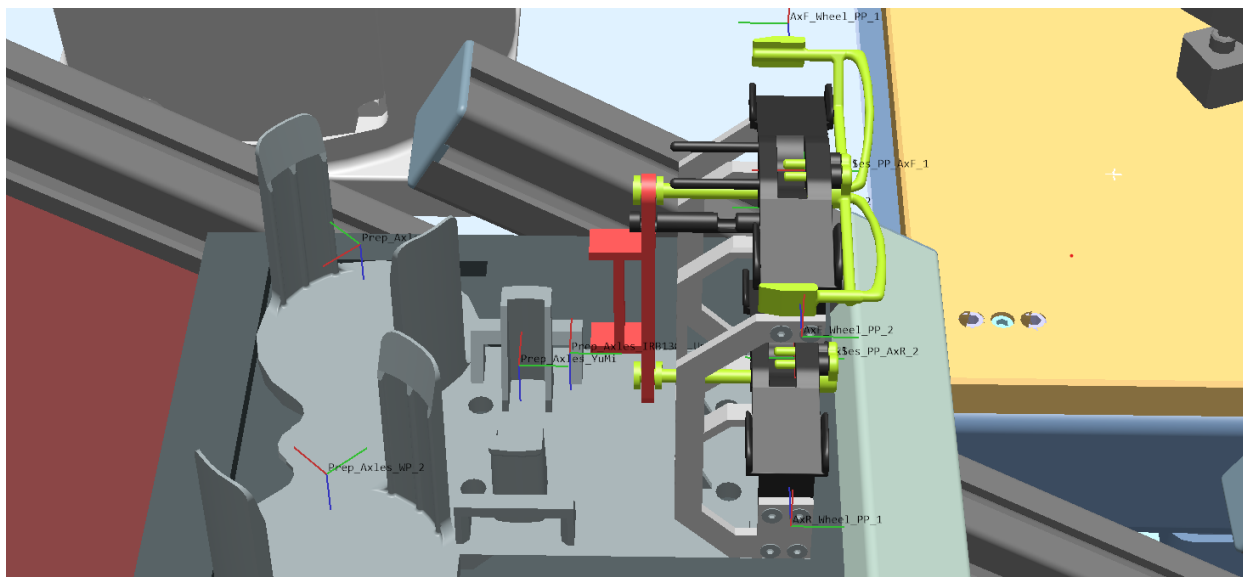
Funkcionalita gripperu je nastavována přes záložku „*Create mechanism*“. Nastavuje se zde rozevírání a svírání čelistí ve stanovém rozsahu a čase, nadefinování TCP a všech 3D dat, které jsou součástí Gripperu. Výstupem tohoto nastavení je naprogramovaný objekt, který software vnímá jako nástroj pro robot. Po jeho připnutí k robotu lze u robotu nastavit TCP nástroje.

#### 5.5 Příprava předmětů pro montáž

I zde proběhla nejdříve příprava v softwaru CAD, ale tentokrát v rámci odměřování souřadných systémů, které byly u modelů, přípravků a montovaných součástí nastavovaných na základě TCP nástroje. Jedná se o efektivní postup, jak i v případě tvarově složitých součástí zajistit přesné polohování manipulovaných dílů.

Postup byl následující k uvažované součásti, jako je v tomto případě motor. Ten byl vložen do sestavy gripperu, u kterého byl již nadefinovaný TCP. Gripper byl napolohován tak, jak je předpokládáno, že bude součást uchopována. Jelikož u modelu motoru byl předem nastavený počátek ve spodní noze pouzdra, od tohoto počátku lze odměřit vzdálenosti jednotlivých rovin TCP. Zároveň byly vytvořeny souřadné systémy míst, kam budou součásti roboty odkládány, ať už v rámci sestavování modelu auta, tak v rámci přípravků či vozíků.

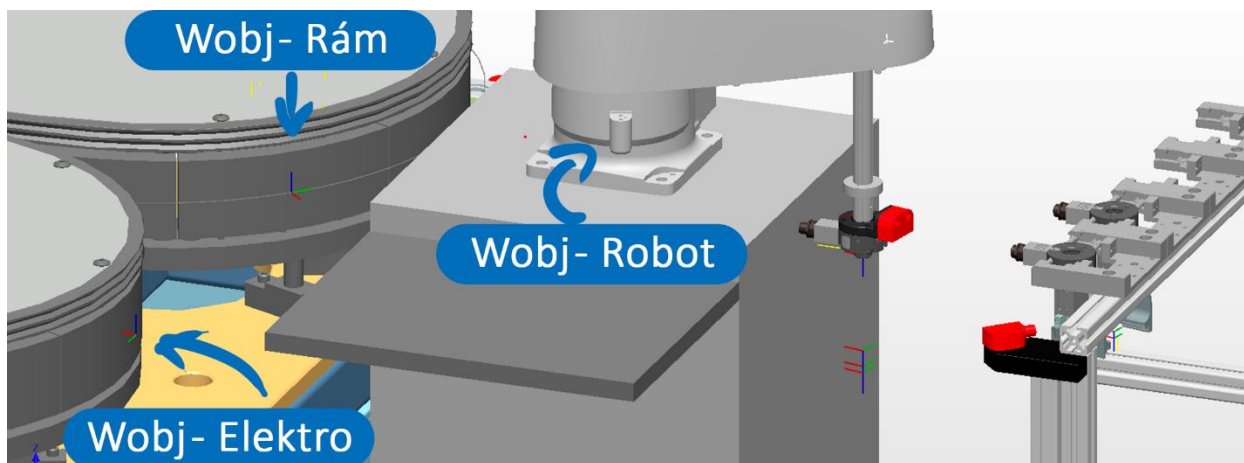
Výsledkem tohoto počínání byly postupně vytvořené učební modely, s pomocí kterých bude později probíhat programování trajektorií robotů. Vše bude vysvětleno v kapitole „**Chyba! Nenašel zdroj odkazů.**“ (viz Obrázek 47).



Obrázek 47 - Vytvořený model se souřadnými systémy pro odběry různými nástroji. Jedná se o systémy, které cestují s paletkou.

## 5.6 Tvorba pracovních objektů

Pracovní objekty, ke kterým budou v pozdější fázi programování robotů přiřazovány body trajektorie, byly vytvořeny na klíčových místech. Pracovní objekty jsou pro každý robot jiné. Příklad pracovních objektů lze spatřit na obrázku (viz. Obrázek 48).



Obrázek 48 - Wobj pro robot SCARA

#### IRB1300

- Základní pracovní objekt (v nulovém bodě robotu, systém si jej vytvoří samostatně)
- Zásobovací regál
- Vozík na paletku na – Nápravy
- Vozík na paletku na – Elektroniku
- Vozík na paletku na – Rám

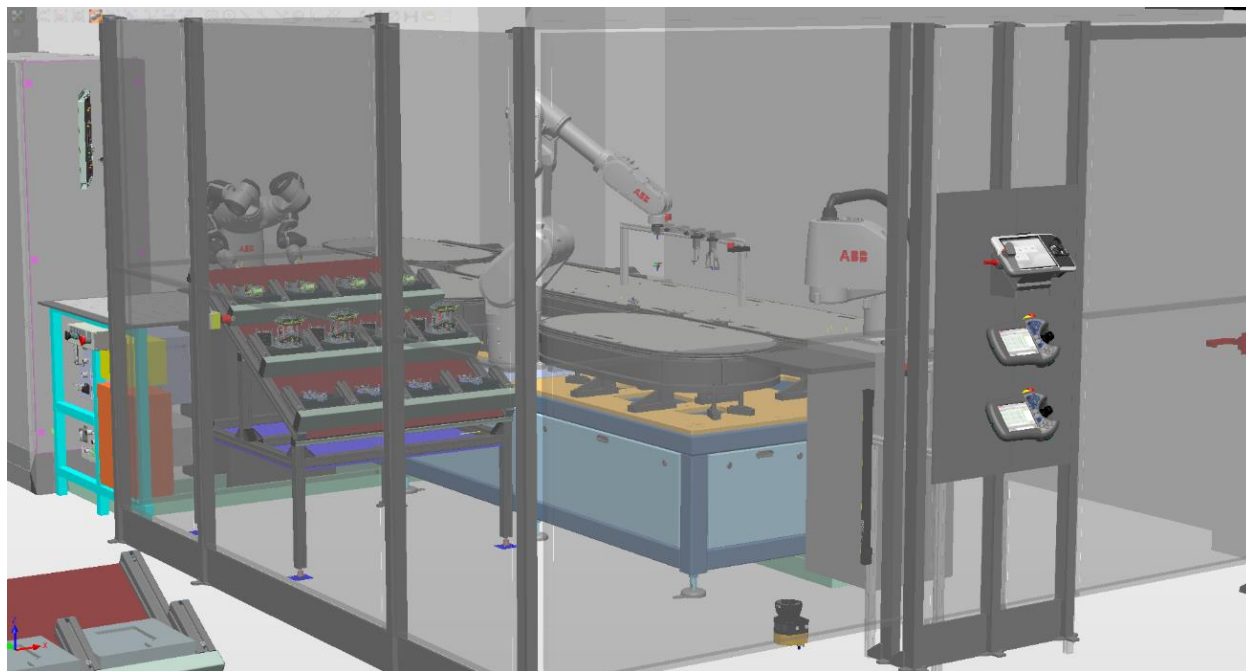
#### IRB14000 YuMi

- Základní pracovní objekt (v nulovém bodě robotu, systém si jej vytvoří samostatně pro každé rameno zvlášť)
- Vozík na paletku na – Nápravy (Levé rameno)
- Montážní prostor (Pro pravé i levé rameno zvlášť)

#### 910SC SCARA

- Základní pracovní objekt (v nulovém bodě robotu, systém si jej vytvoří samostatně)
- Zásobovací regál
- Vozík na paletku na – Nápravy
- Vozík na paletku na – Elektroniku
- Vozík na paletku na – Rám

Zcela jistě by se dal vytvořit pracovní objekt pro stojany s výměníky, ale u těch je předpokladem, že jejich pozice bude neměnná.



Obrázek 49 Výsledek příprav simulačního modelu

## 5.7 Instalace kontroléru

Důležitou částí příprav je také příprava robotického kontroléru, ze kterého je řízen robot. Jak již bylo zmíněno, k simulovanému kontroléru lze přistupovat jako skutečnému. Do kontroléru je tedy možné doinstalovávat aktualizace či pomocné knihovny. Jedním z doplňků, který bylo u každého z robotických kontrolérů nutno doinstalovat, je ROBOTemplate, který poslouží v pozdější fázi programování jako most mezi robotickým kontrolérem a simulovaným PLC.

Nyní, když jsou dokončeny veškeré přípravné činnosti pro tvorbu simulačního modelu pracoviště, je možné přesunout se do další části, která se již týká samotného programování robotické buňky.

## 6 Řídící programy

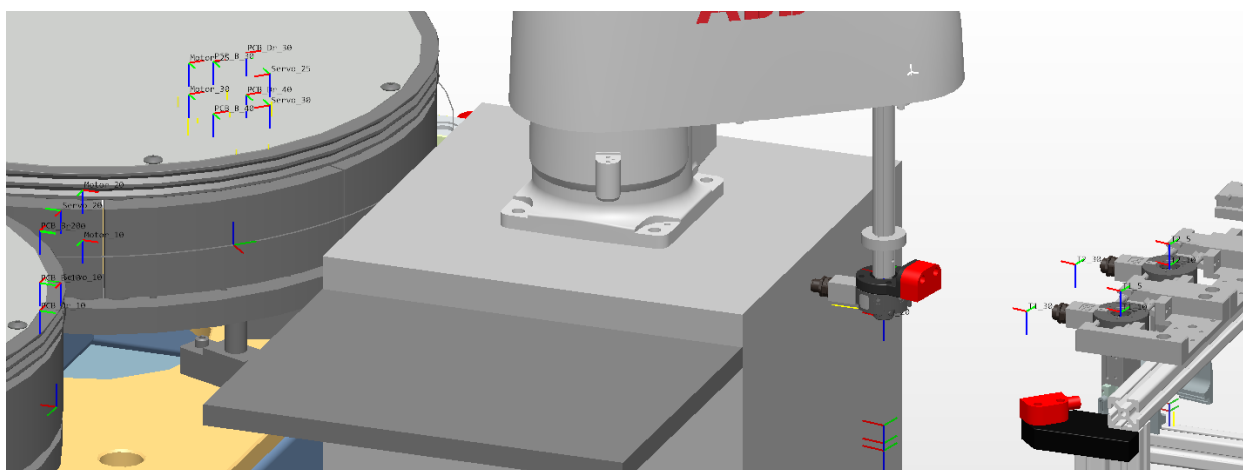
Kapitola je věnována kompletní problematice programování robotické buňky. Nejdříve bude věnována pozornost programování dráhy robotů, po kterých bude následovat doprogramování logiky prvků, které se na pracovišti nacházejí přes takzvané Smart components. Poslední podkapitola bude věnována řízení PLC, u kterého je cílem vytvořit virtuální zprovoznění linky a ověření funkčnosti navrženého algoritmu.

### 6.1 Roboty

Tato podkapitola je věnována vytváření trajektorií robotů a popisu způsobu, s pomocí něhož bylo trajektorií dosaženo. Jak bylo popsáno v rešeršní části práce „Trajektorie robotu“, vzniku trajektorií robotu lze docílit mnoha metodami programování. Výsledkem všech metod je vznik množiny bodů v prostoru, které společně vytvářejí trajektorii TCP.

#### *Tvorba bodů v prostoru*

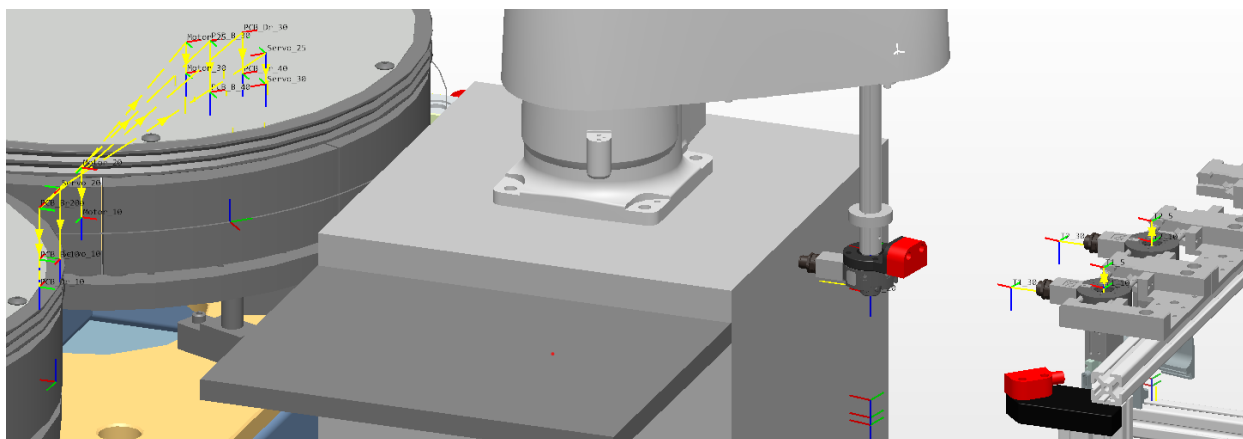
V předchozí podkapitole byla zmínka o učebních předmětech s různými souřadnými systémy. Tyto souřadnicové systémy lze považovat za koncové body pro manipulaci vytvářené trajektorie. Zajišťují nám místo uchopení nástroje a jeho orientaci při manipulaci. Ovšem tyto koncové body jsou pro vznik korektní trajektorie operace Pick&Place málo. Pokud by byla trajektorie vytvořena jen s pomocí dvou bodů, robot by mezi body putoval nejkratší možnou cestou (pokud by se nacházel v jednom z bodů), která by zcela jistě vedla skrze přípravky a okolní prostředí. Proto byl od každého koncového bodu vytvořen minimálně jeden další bod, který je odsazený a poslouží jako bod příjezdový. Nachází se v takové vzdálenosti od koncového bodu, aby bylo možné zajistit bezpečný příjezd robotu s nástrojem. Vytvořené body pro robot SCARA lze vidět na obrázku (viz Obrázek 50).



Obrázek 50 - Vytvořené body v prostoru vázající se k wobj. Z těchto bodů bude následně vytvořena trajektorie

### Tvorba trajektorie

Trajektorie byly vytvořeny s pomocí chronologického uskupení souvisejících bodů. Tím dojde v softwaru RobotStudio k vytvoření trajektorie. Trajektorie však nebyly vytvořeny jako jedna souvislá křivka v prostoru. Každá manipulační operace byla rozdělena do několika samostatných trajektorií kvůli řízení z PLC, odkud budou trajektorie volány zrovna tak, jako nastavení sevření gripperu.



Obrázek 51 - Vytvořené trajektorie pro montáž

Vychází to z charakteristiky komunikace mezi PLC a Kontrolérem robotu. V PLC jsou volány chronologicky trajektorie. Do volané trajektorie není možné bez vytváření obtížné logiky nic vpisovat. Ovšem, kde lze provádět změny proměnných, je právě na začátku a na konci vykonávané trajektorie.

Celková trajektorie každé operace je tedy rozdělena na další tři, které jsou:

- Příjezdová trajektorie
- Retrakce s doručovací trajektorií
- Retrakce

Po vykonání retrakce přichází na řadu další operace. Příklad kódu a nastavení je vidět na obrázku (viz. Obrázek 52).



Trajektorie v hierarchii  
RobotStudio

```

26 PROC Battery1_Pick()
27   MoveJ PCB_B_20,v800,fine,PCB_TCP_1\Wobj:=WO_SH;
28   MoveL PCB_B_10,v150,fine,PCB_TCP_1\Wobj:=WO_SH;
29 ENDPROC
30 PROC Battery2_Deliver()
31   MoveL PCB_B_20,v150,z100,PCB_TCP_1\Wobj:=WO_SH;
32   MoveJ PCB_B_30,v500,z50,PCB_TCP_1\Wobj:=WO_FRAME;
33   MoveL PCB_B_40,v100,fine,PCB_TCP_1\Wobj:=WO_FRAME;
34 ENDPROC
35 PROC Battery3_Retract()
36   MoveL PCB_B_30,v300,fine,PCB_TCP_1\Wobj:=WO_FRAME;
37 ENDPROC
38 PROC Driver1_Pick()
39   MoveJ PCB_Dr_20,v800,fine,PCB_TCP_1\Wobj:=WO_SH;
40   MoveL PCB_Dr_10,v150,fine,PCB_TCP_2\Wobj:=WO_SH;
41 ENDPROC

```

Jednotlivé pohybové procedury

Obrázek 52 - Rozdělení pohybových procedur jako příprava pro řízení z PLC

### Nastavení trajektorie

Každý jednotlivý pohyb mezi body trajektorie je ještě nutno nakonfigurovat s pomocí konfiguračních parametrů.

- Rychlost pohybu
- Typ dráhy MoveJ, MoveL
- Zóny dráhy
- Konfigurace robotu
- Volba TCP

Lineární pohyb „MoveL“ byl zejména nastaven u odběrů a retrakcí. To s malou rychlostí a co nejmenší zónou dráhy, tedy minimálním zaoblením.

Pohyb přes klouby „MoveJ“ byl nastaven zase mezi přejezdy, kde je mnohdy přejezd s pomocí „MoveL“ pro robot nerealizovatelný, a to zejména u sedmiosého robotu YuMi. U těchto pohybů byly nastavené vyšší rychlosti a větší zóny.

Na špatnou konfiguraci robotického ramene velice často RobotStudio upozorní samo, takže pokud nevyskočilo upozornění či konfigurace robotu nebyla vyloženě nežádoucí, nemělo smysl ji zkoušet cíleně měnit.

TCP byla volena na základě nástroje, který byl na danou operaci potřeba. Aby bylo možné mezi nástroji volit, musel být každý nástroj alespoň jednou připnutý k robotu. Tím došlo k vytvoření informace TCP pro programovaný robot. Dále je tato informace podstatná pro generování kódu. Kontrola, zda je všude použit správný TCP žádaného nástroje, je velice důležitá, jinak se program nebude chovat tak, jak bylo zamýšleno, a pravděpodobně bude robot kolidovat.

### Přiřazení k pracovním objektům wobj

V posledním kroku bylo vykonáno rozdělení vytvořených bodů mezi jednotlivé pracovní objekty na základě jejich souvislosti s danými objekty. Doplňkové body, které byly vytvořeny jen kvůli optimalizaci trajektorie, zůstaly ponechány základnímu pracovnímu objektu u každého

z robotů. Bylo to provedeno kvůli parametrizaci a také případné optimalizaci polohy objektu, čímž dojde k pohybu vytvořené množiny bodů.

### Generování kódu

Po dokončení veškerých příprav a nastavení byl vygenerován kód v jazyce RAPID pro každý robot. Příklad kódu lze spatřit na obrázku (viz Obrázek 53). Avšak ještě před generováním kódu byly vytvořeny dva moduly v Kontroléru robotů. Jeden modul, jehož obsah budou tvořit trajektorie a druhý modul pro body tvořící onu trajektorii.

V Generátoru bylo poté nutné manuálně přiřadit každé trajektorii a bodu modul do kterého bude kód následně generován. Zároveň generátor tvoří i kód pro pracovní objekty a nástroje.

Kód trajektorie, který byl vygenerován do zvoleného modulu bude v pozdější fázi práce využit pro řízení PLC.

```

26 PROC Battery1_Pick()
27   MoveJ PCB_B_20,v800,fine,PCB_TCP_1\Wobj:=WO_SH;
28   MoveL PCB_B_10,v150,fine,PCB_TCP_1\Wobj:=WO_SH;
29 ENDPROC
30 PROC Battery2_Deliver()
31   MoveL PCB_B_20,v150,z100,PCB_TCP_1\Wobj:=WO_SH;
32   MoveJ PCB_B_30,v500,z50,PCB_TCP_1\Wobj:=WO_FRAME;
33   MoveL PCB_B_40,v100,fine,PCB_TCP_1\Wobj:=WO_FRAME;
34 ENDPROC
35 PROC Battery3_Retract()
36   MoveL PCB_B_30,v300,fine,PCB_TCP_1\Wobj:=WO_FRAME;
37 ENDPROC
38 PROC Driver1_Pick()
39   MoveJ PCB_Dr_20,v800,fine,PCB_TCP_1\Wobj:=WO_SH;
40   MoveL PCB_Dr_10,v150,fine,PCB_TCP_2\Wobj:=WO_SH;
41 ENDPROC

```

### Jednotlivé pohybové procedury

```

10 CONST robtarget PCB_B_20:=[0.047172439,41.999020235,184.155690542],[0,-0.707106781,0.707106781,0],[-1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
11 CONST robtarget PCB_B_10:=[0.047172439,41.999020235,134.155690542],[0,-0.707106781,0.707106781,0],[-1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
12 CONST robtarget PCB_B_30:=[-1.071125006,-19.301890528,178.599690542],[0,0.70710678,0.707106782,0],[-1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
13 CONST robtarget PCB_B_40:=[-1.071125006,-19.301890528,128.599690542],[0,0.70710678,0.707106782,0],[-1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];

```

### Body trajektorie

Obrázek 53 - Vygenerovaný kód pro robot IRB910SC. Jedná se o program pro odběr baterie

### Příprava robotu pro řízení z PLC

Posledním krokem, který je potřeba v RobotStudio provést v rámci robotů, je nutnost vytvoření hlavního modulu v kontroléru robotu, který bude sloužit pro komunikaci s PLC systémem.

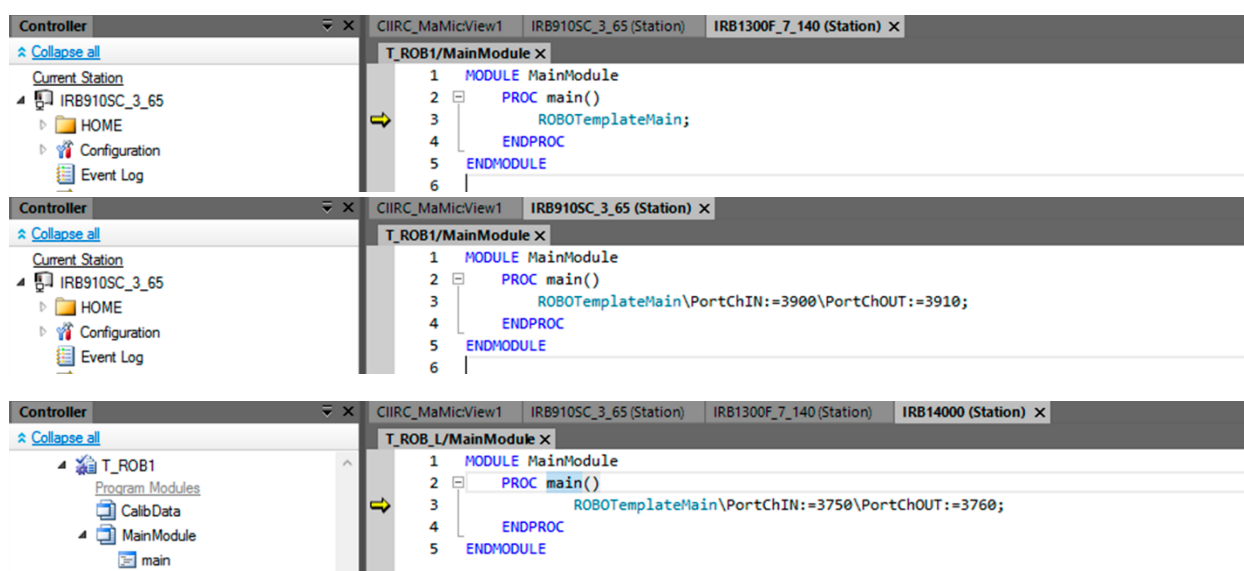
Modul byl pojmenován jako „MainModule“. Následně do něj byl dopsán program, který vykonává komunikaci s PLC. Je nutné pojmenovat ho „main“. Ve vytvořeném programu je potřeba přiřadit individuálně pro každý robot volitelný parametr portu, skrze který bude posléze probíhat komunikace mezi kontrolérem robotu a PLC. Pro každý robot je potřeba vytvořit minimálně dva unikátní porty. Tím bude možné roboty v Automation Studiu později odlišit v rámci komunikace a ovládání. U jednoho z robotů lze nevyplnit žádný parametr a tím pádem jsou nastaveny základní hodnoty portů.



Po vytvoření komunikačního programu bylo potřeba simulaci alespoň jednou spustit, aby došlo k uložení. Ukládání by mělo být prováděno ideálně pouze s jedním robotem povoleným v simulaci.

I přesto, že robot YuMi má dvě ramena, stačí vytvořit pro obě ramena identický modul se stejnými porty. Je to dáno tím, že skrze kontrolér IRC5 lze komunikovat až se čtyřmi roboty. Jak budou od sebe ramena odlišena bude probíráno v podkapitole PLC řízení konkrétně v části „Příprava“.

Posledním krokem v kontrolérech robotů bylo ujistit se, že program pointer je nastaven na čtení programu „main“. Příprava komunikace pro roboty je k vidění níže na obrázku (viz Obrázek 54).



Obrázek 54 - Nastavení komunikace pro: IRB1300, IRB910SC, YuMi L/R

## 6.2 SmartComponents

Tyto programovatelné komponenty posloužily v simulaci pro tvorbu mechaniky a jevů, které ve skutečnosti obstarává fyzika či jiné řídicí systémy. To umožňuje simulaci doplnit o předměty a děje, které v RobotStudio chybí, jako je třeba dopravník ACOPOStrak. Výhodou těchto komponent je, že RobotStudio umožňuje ovládání komponenty přes PLC. Veškeré prvky, které byly naprogramovány budou k přečtení níže.

### *Dopravník*

Jedna z největších a nejdůležitějších částí, která se na pracovišti nachází je dopravník ACOPOStrak. RobotStudio v tuto chvíli neumožňuje přímou simulaci tohoto dopravníku. Aby bylo možné vytvořit digitální dvojče dopravníku, vyžadovalo by to množství času a práce pro vytvoření externího skriptu, který by byl jako doplňková komponenta do SmartComponents.

Další překážkou bylo, že algoritmus řízení dopravníku není příliš jednoduchou záležitostí a jelikož v tuto chvíli nebylo ani možné zrcadlit opravdový dopravník, jehož logika je řízena z PLC, bylo rozhodnuto, že dopravník bude naprogramován celý jako jedna komponenta s předem definovanou tratí pohybu vozíků. Do této komponenty budou přivedeny vstupy a vyvedeny výstupy pro jednoduché povelování z PLC. Dopravník bude tedy fungovat jen pro účely simulace a nebude mít vlastnosti, které má skutečný dopravník, jako je schopnost dynamického přepínání mezi ovály. Proto bylo potřeba simulaci podvolit tomuto stavu a na pracovišti se pracuje zatím jen se třemi vozíky.

#### Návrh dopravníku

Celý koncept je vizuálně jednoduchý, ale poměrně obtížný na vytvoření. Principiálně je to objekt, který sleduje vytvořenou dráhu. Ta byla vytvořena v RobotStudio s pomocí čar a kružnic po kterých se poté bude vozík pohybovat. Čáry jsou orientované tudíž nezáleží na tom, jak je čára vytvořena. V prvním vytvořeném bodě čáry totiž dojde zároveň i k tvorbě souřadnicového systému, který už není možné později změnit. Zároveň čáry musejí být v RobotStudio uchovány jednotlivě. Kdyby byly sloučeny, může dojít k problémům s orientací objektu, který se po dráze bude pohybovat. V bodě, kde má vozík zastavit na montáž je křivka přerušena, aby byl pevně stanoven konec dráhy. Místo zastavení se dá také vyřešit senzorem, ale RobotStudio má svou frekvenci vzorkování. Čili místo zastavení je poté proměnlivé v závislosti na rychlosti pohybu vozíku. Aby nebylo nutné řešit parametrizování pracovních objektů, který by vždy zajistil posun bodů trajektorie robotu na místo zastavení, byla zvolena jednodušší cesta pro účely simulace. Výsledek tvorby trajektorie vozíků pro dopravník je k vidění na obrázku (viz Obrázek 55).



Obrázek 55 - Trajektorie dopravníku

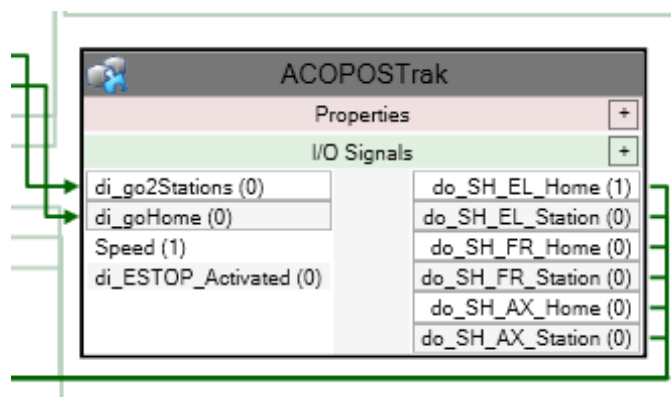
Tímto způsobem a s těmito zásadami byla pro každý z vozíků vytvořena dráha na místo zastavení ve stanici u robotů a dráha, která opět vozík vrátí zpět na počátek. Tedy vcelku šest různých drah.

Komponenta dopravníku má pouze čtyři vstupní signály, kterými je ovládán a 6 signálů, které udávají informaci o poloze vozíku jako výstup (viz Tabulka 4). Tato informace je posléze využita pro zahájení montáže roboty.

Tabulka 4 - Vstupy a výstupy dopravníku

Vstup	Výstup
Odjezd do stanice	Vozík s elektronikou - Doma
Odjezd domů	Vozík s elektronikou - Stanice
Rychlost	Vozík s rámem - Doma
Nouzové zastavení	Vozík s rámem - Stanice
	Vozík s osou - Doma
	Vozík s osou - Stanice

Komponenta pro řízení vozíků sestává ze tří úrovní. Uvnitř hlavního bloku pro řízení dopravníku jsou totiž další komponenty, kterými jsou již řízeny jednotlivé vozíky. Na této úrovni probíhalo ladění. O úroveň níže je pak samotná logika sledování vytvořených čar vozíkem. Tyto komponenty jsou pak připnuty ke stejnojmennému vozíku. Logika dopravníku je k vidění na obrázku (viz Obrázek 56).

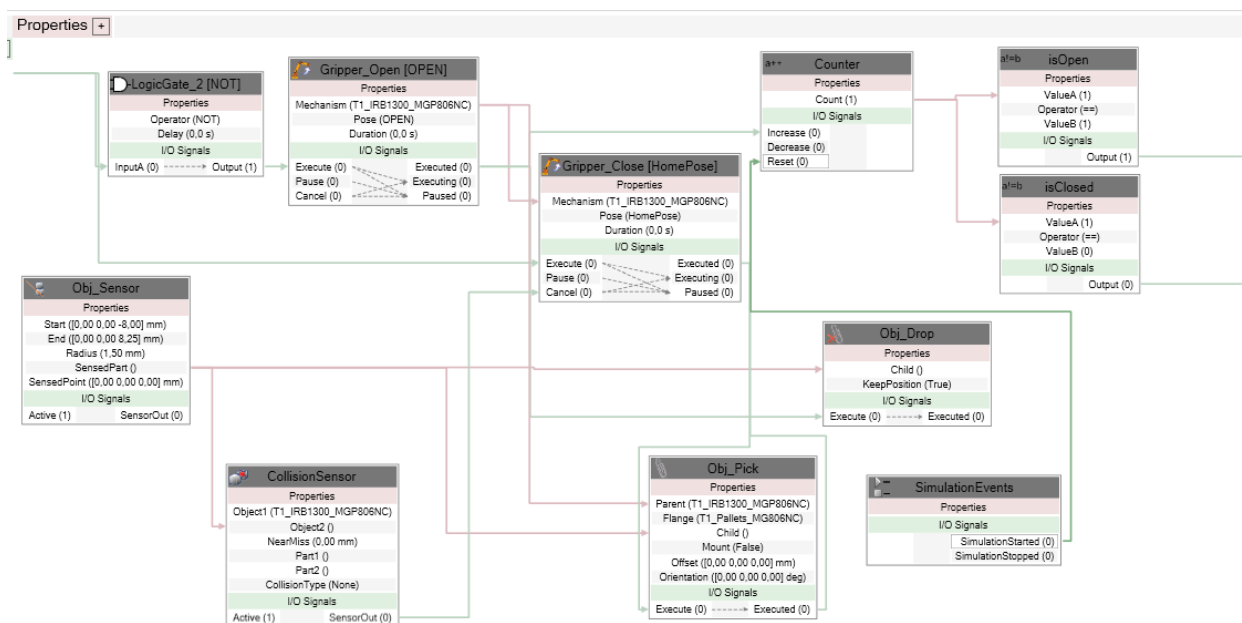


Obrázek 56 - Nejvyšší vrstva ovládání dopravníku

### Ovládání Gripperu

Logika, která předchází robotům s automatickými výměníky a robotem YuMi je trochu odlišná, avšak jádro kontroly gripperu zůstává stejné.

Naprogramované schéma (viz Obrázek 57) se zabývá dvěma věcmi a to svíráním/rozevíráním a detekcí objektu, která následně zajistí manipulaci s předmětem. V simulačním prostředí není totiž nastavena fyzika, která by zajišťovala manipulaci. Tato možnost tam je, jen nebyla využita, protože linka je chystána na tvorbu digitálního dvojčete.



Obrázek 57 - Ovládání Gripperu

Ze schématu je patrné, že logika polohování gripperu je nastavena z maxima do minima. Aby v případě různě velkých součástí nedošlo k prolnutí prstů s uchopeným předmětem, byla potřeba nastavit detekce kolize. Na základě této detekce dojde i k aktivaci signálu „Obj\_Pick“, který virtuálně připne k nástroji robotu. Předmět však nelze uchopit bez senzoru, který je zde pojmenován jako „Obj\_Senzor“. Senzory byly vyvedeny u každého robotu z TCP nástroje tak, aby byl cílený předmět vždy senzorem zaznamenán.

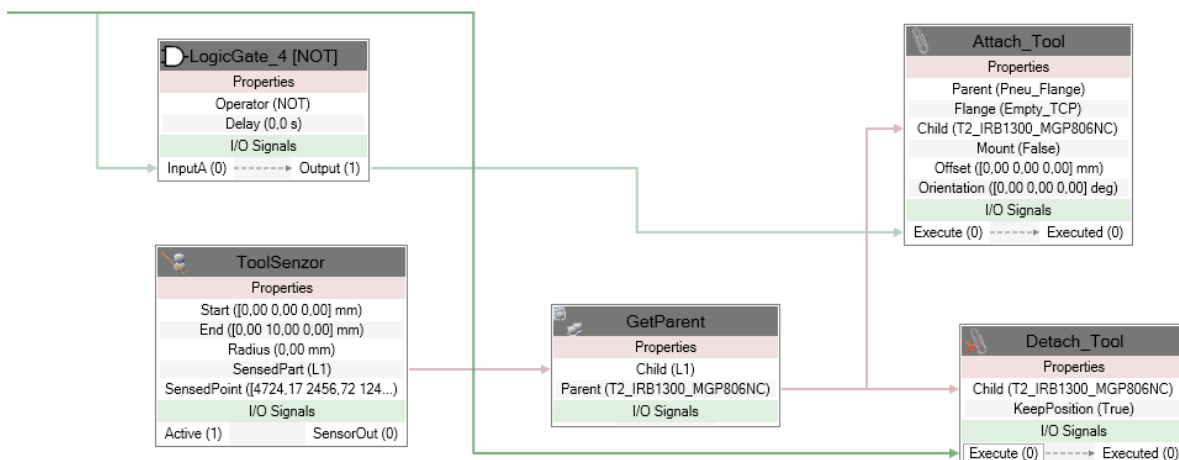
Počítadlo na konci schématu je logika pro výstup, aby bylo známo, zda je Gripper již otevřen, či nikoliv. Vstupy a výstupy pro tento blok jsou k vidění v tabulce níže (viz Tabulka 5).

Tabulka 5 - Vstupy a výstupy pro ovládání gripperu

Vstup	Výstup
Gripper - zavřít	Gripper je otevřen
	Gripper je zavřen

### Výměník nástrojů

Tato komponenta (viz Obrázek 58) byla naprogramována pro schopnost připnout či odepnout nástroj k robotu na základě vstupního signálu. S pomocí senzoru je detekován nástroj a ve stanovenou chvíli s pomocí vstupu je nástroj připnut. Výměník není naprogramován na základě chování skutečného výměníku. Aby logika odpovídala skutečnosti, byla by o něco složitější. Vstupy a výstupy potřebné pro ovládání výměníku (viz Tabulka 6).

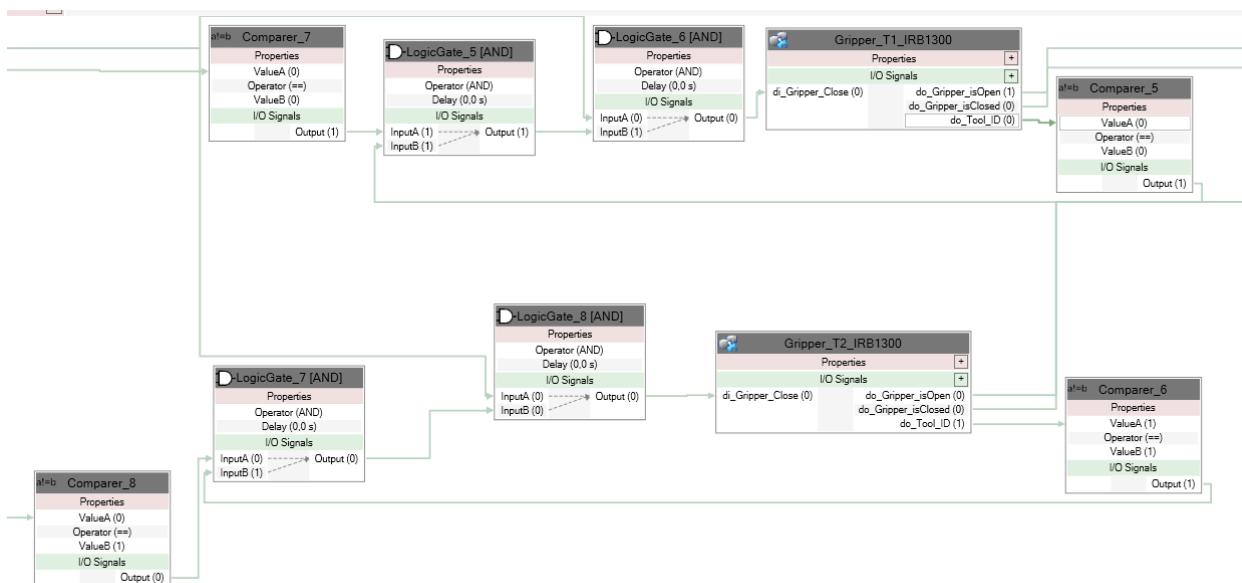


Obrázek 58 - Logika bloku pro výměnu nástroje

Tabulka 6 - Vstupy a výstupy pro ovládání výměníku

Vstup	Výstup
Uvolnit nástroj	

Aby během simulace nedocházelo k otevírání a zavírání gripperů, které nejsou momentálně připnuty k robotu, byla vytvořena komponenta s logikou, která tomuto jevu brání. Ve své podstatě kontroluje, zda připnutý gripper podle ID odeslaného z PLC, odpovídá ID gripperu připnutému v simulaci. Tento blok (viz Obrázek 59) je nadřazený komponentě pro ovládání gripperu. Jsou zde tedy vyvedeny stejné vstupy a výstupy jako byly u komponenty pro ovládání gripperu. Vstupy a výstupy pro tento blok jsou v tabulce (viz Tabulka 7).



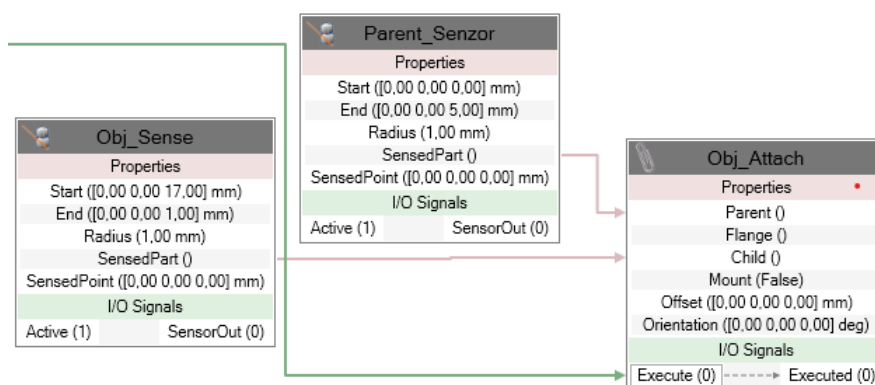
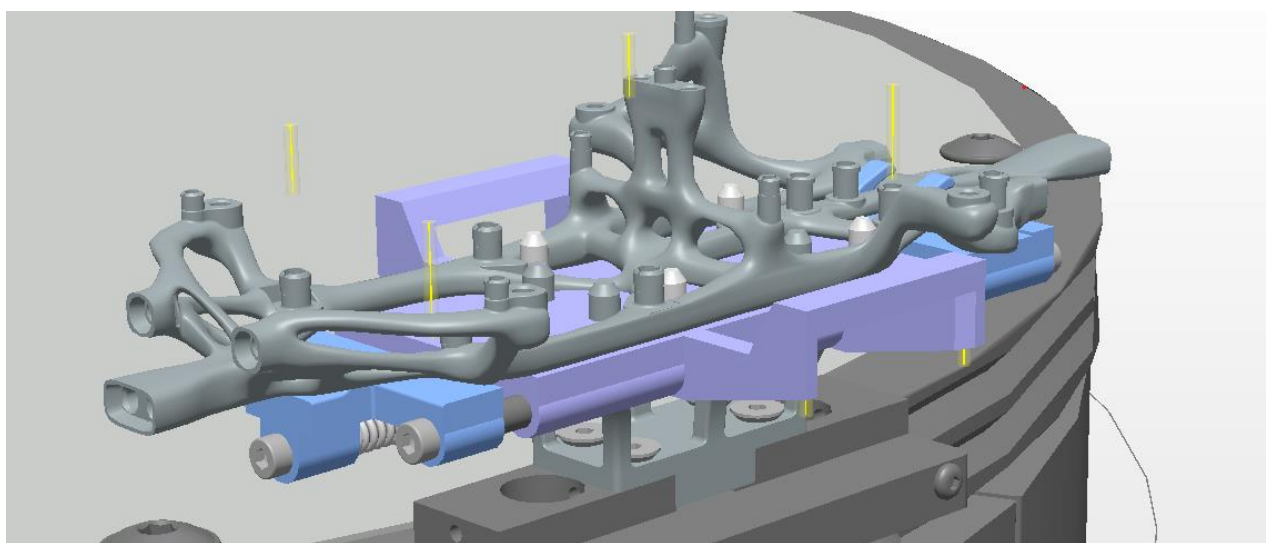
Obrázek 59 - Logika volby nástroje

*Tabulka 7 - Logický člen pro ovládání zvoleného nástroje*

Vstup	Výstup
ID požadovaného nástroje	Gripper je otevřen
Gripper - zavřít	Gripper je zavřen

### Logika položení

I přesto, že se jedná o věc ve skutečném světě samozřejmou, zde je potřeba fungování simulovaného světa doprogramovat. Odkládací logika je řešena přes senzory (viz Obrázek 61), které jsou položeny v místech, kde chceme objekty detekovat. Detekovaný objekt je poté na základě vstupního signálu připnut k žádanému objektu. Žádaný objekt (dále jen rodič) je v některých případech napevno zadán jako parametr a v některých případech je opět nejdříve detekován senzorem. Logika je k vidění (viz Obrázek 60).


*Obrázek 60 - Logika uchycení detekované součásti k detekovanému tělesu*

*Obrázek 61 - Žluté čáry představují senzory, které detekují součást k uchycení k rámu*

Případy, kdy byl rodič zadáván napevno, bylo u neměnných rodičů. To platí u dopravníku, kde se vozíky nemění a zůstávají stejné. Naopak, kde se rodič vždy mění, je při jakékoliv montáži, kde proměnného rodiče tvoří paletka s rámem či osy náprav. Dochází totiž k zaměňování palet po dokončení předchozí montáže a přechodu na montáž novou.

Tato logika byla implementována do každého montážního bodu linky.

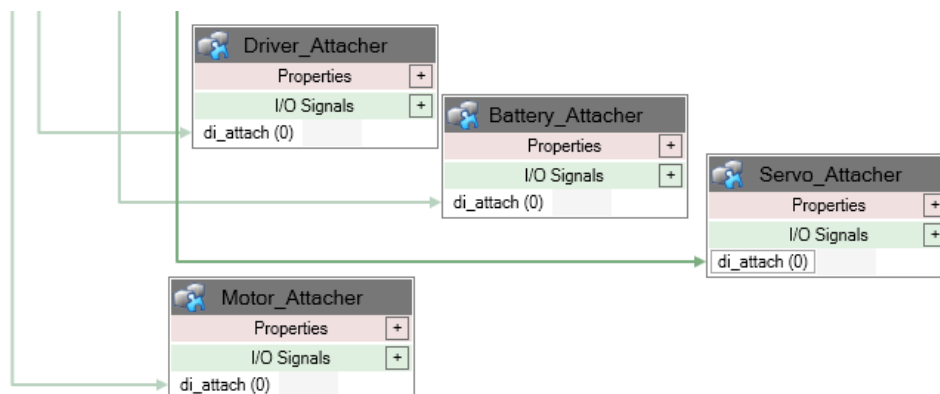
- Položení na vozíky
- Montáž elektroniky
- Montáž kol na nápravy
- Montáž náprav na rám

Vstupní signál lze považovat za pouhý digitální signál (viz Tabulka 8), který je řízen jen v rámci simulace, nikoliv tak, aby byl řízen skrze PLC.

Tabulka 8 - Vstupy a výstupy pro ovládání uchycení komponent k sestavě

Vstup	Výstup
přípnout	

Komponenta byla v simulačním modelu uplatněna jak jako samostatná, tak i jako komponenta podřazená, a to například u dopravníku, kde je vstupní signál vyvolán uvedením dopravníku do pohybu. Hromadně volaný blok pro uchycení je k vidění (viz Obrázek 62).



Obrázek 62 - Hromadně volaný blok pro uchycení elektronických komponent k rámu

### Doplňky

V práci bylo naprogramováno i funkční AMR a automatické otevírání brány robotické buňky. Jedno AMR sloužilo pro import a druhé pro export hotových sestav. Simulační model by tak teoreticky mohl běžet do nekonečna, jelikož v RobotStudio byla doprogramována i komponenta pro generování nových plných regálů a mazání odvezených prázdných regálů.

Komunikace AMR byla vyřešena interně v rámci RobotStudio, jelikož bylo předpokládáno, že AMR budou ve skutečnosti řízena z nadřazeného systému. V simulaci RobotStudio byla nachystána trasa AMR a komunikace mezi nimi, aby nedošlo ke kolizi či nevjela obě AMR do robotické buňky. Dále AMR komunikovali s linkou, aby bylo vyřešeno zásobování včas. Bohužel, z časových důvodů zůstal tento koncept rozpracovaný a do finálního simulačního modelu nebyl zatím implementován.

Bohužel se jedná o velmi rozsáhlou část, kterou nelze v práci rozepsat a vysvětlit s pomocí screenshotů, jelikož by práce nakonec byla nepřehledná a díky malým rozměrům obrázků mezi texty i nesrozumitelná. Proto budou dílčí bloky uvedeny ještě v příloze, kde bude vidět i celková logika pracoviště.

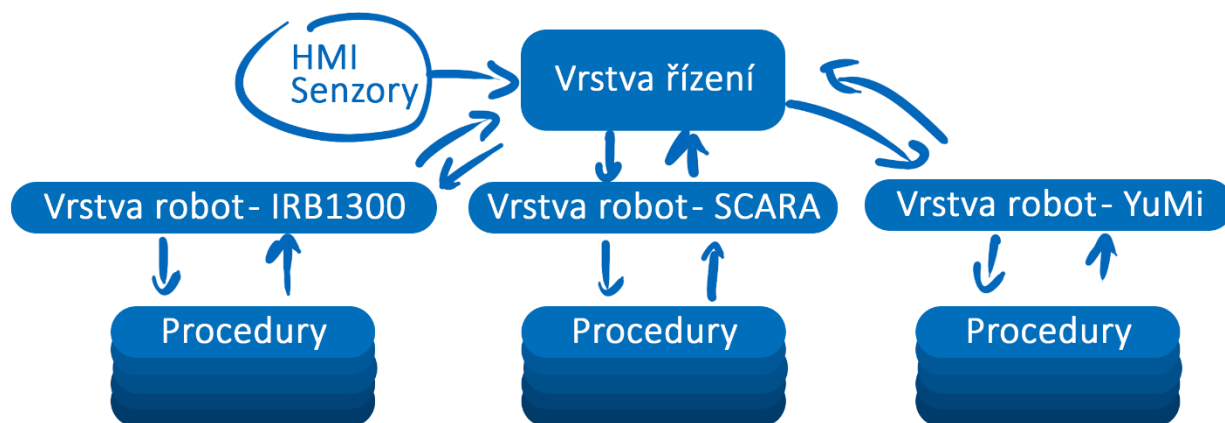
Nyní je simulační prostředí v rámci RobotStudio kompletní. Je tedy na čase přejít k poslednímu kroku, a to řízení celé linky skrze PLC.

### 6.3 PLC

Tato kapitola se věnuje části řízení celého robotického pracoviště, které bude jako pracoviště skutečně řízené celé z PLC. Tato kapitola povede k cílovému výstupu práce, kterým je virtuální zprovoznění linky. V předchozích kapitolách byly provedeny veškeré přípravy, a to od návrhu předmětu montáže až po nachystání virtuálního pracoviště s pohybovými rutinami. Nyní je cílem naprogramovat zbývající logiku pracoviště.

#### *Návrh algoritmu řízení*

Cílem je vytvořit takový kód, který bude dobře čitelný a přehledný. K tomu napomůže tvorba vlastního programu pro každý z robotů, aby bylo možné oddělit řízení pracoviště od řízení robotu. Pod programem robotu bude možné nalézt volání jednotlivých procedur z kontroléru robotu.

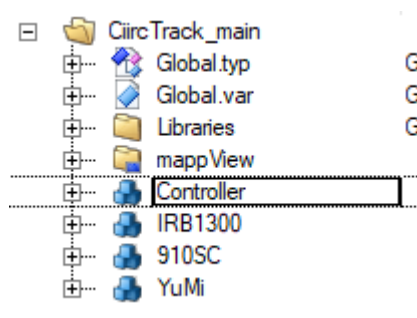


Obrázek 63 - Schéma softwarové architektury pro řízení pracoviště

Řízení pracoviště bylo navrženo tak, aby byl kód přehledný, čitelný a modulární. Proto byla struktura řízení rozdělena do několika vrstev, a to řídicí vrstvy, kde probíhají veškeré změny, vrstvy robotů, která slouží pro nastavení komunikace s robotickým kontrolérem a pro výpis



pohybových procedur robota a v té nejnižší vrstvě jsou již jednotlivé pohybové procedury. Jednotlivé vrstvy budou rozepsány v následujících podkapitolách, které budou zaměřeny na účel a obsah těchto vrstev. Schéma lze vidět na obrázku (viz Obrázek 63). Na obrázku (viz Obrázek 64) jsou vidět vytvořené vrstvy.



Obrázek 64 - Programové vrstvy. Programy s názvy robotů pod sebou skrývají akce. Dále každá z vrstev má své proměnné, cyklický, inicializační a ukončovací část programu.

Než se však přistoupí k samotným vrstvám, je potřeba provést popis globálních proměnných a struktur, které byly vytvořeny, jelikož se jedná o zásadní část pro tvorbu algoritmu řízení pracoviště.

Proměnné, které se vytvoří u programu jsou vždy lokální, nelze je tedy používat v ostatních programech či jak je zde nazýváno, vrstvách. Globální proměnné (viz Obrázek 65) mají tu výhodu, že je lze volat napříč všemi vrstvami. Proto zde byly deklarovány proměnné, které slouží jako zpětná vazba o aktuálním stavu robotu z vrstev nižších do vrstvy řídicí, či pro nastavení žádaného stavu v nižší vrstvě. Je však nutné dodržet konvenci a proměnné měnit vždy jen z jednoho místa. Zde je to odlišeno tak, že pokud proměnná začíná malým písmenem, jde o proměnnou, která je měněna jen v nižších vrstvách. Má-li proměnná písmeno velké, jedná se o proměnnou, která slouží pro řízení vrstev nižších.

Name	Type	Constant	Retain	Replicable	Value
←-PROCEDURE STATUS-→					
done_IRB1300	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
done_SCARA	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
done_YuMi_R	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
done_YuMi_L	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
←-ROBOT STATUS-→					
Error_IRB1300	DINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Error_SCARA	DINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Error_YuMi_R	DINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Error_YuMi_L	DINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
←-ROBOT RESET-→					
Error_Reset_IRB1300	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Error_Reset_SCARA	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Error_Reset_YuMi	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Error_Reset_YuMi_R	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Error_Reset_YuMi_L	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
←-ROBOT PROGRAM-→					
PrgSet_IRB1300	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0
PrgSet_SCARA	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0
PrgSet_YuMi_R	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0
PrgSet_YuMi_L	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0
←-WorkObj offset-→					
StepOver	UINT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0

Obrázek 65 - Globální proměnné pro řízení pracoviště

**Procedure status** – Tyto proměnné slouží pro komunikaci z robotické vrstvy do řídicí vrstvy. Udávají informaci o aktuálním stavu robotu, zda je připraven pro další činnost. Této informace je využito u stavového automatu, který se nachází v řídicí vrstvě.

**Robot reset** – Proměnná sloužící pro reset chyby robotického kontroléru, a to zejména při startu robotické linky. Musí být nastavena jako globální proměnná, jelikož veškeré proměnné pro komunikaci s robotickým kontrolérem jsou nastaveny až v nižší vrstvě.

**Robot program** – Určené pro spouštění dalších pohybových procedur na základě aktuálního stavu stavového automatu.

**Workobject offset** – Tato proměnná slouží pro posouvání robotického ramene IRB1300 do další řady s díly, které se nachází v regálu. Děje se to vždy po vyložení složeného auta z dopravníku. Proměnná je měněna tedy na konci stavového automatu a je komunikována až do nejnižší vrstvy, a to pohybové procedury robotu.

**Struktury** – Dalším globálním prvkem jsou struktury (viz Obrázek 66). Ty slouží zejména pro organizaci programu a jeho konzistentnost. Obsahem struktur jsou proměnné, které jsou pro všechny roboty stejné. Nebylo tedy potřeba pro každý robot vytvářet ty stejné proměnné znovu. V proměnných robotické vrstvy byla nedefinována proměnná se jménem robotu, u které byl zvolen vytvořený datový typ struktury. Skrze deklarované jméno lze poté přistupovat k jednotlivým instancím proměnných, které tvoří obsah struktury. Do struktury

„Robot\_Structure“ byl rovnou zahrnut i funkční blok z knihovny ROBOTemplate, který je pojmenován jako „RBTCtrl“. Tento funkční blok umožňuje komunikaci a řízení robotu z PLC.

Robot_Structure						
RBTCtrl	RBTemplateRobo...	<input type="checkbox"/>	<input checked="" type="checkbox"/>			
ProcedureStep	USINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0	
PrgStep_Old	USINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0	
Start	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		TRUE	
Busy	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		TRUE	
Tool_ID	BYTE	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0	
Tool_Release	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		TRUE	
Tool_Close	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		TRUE	
Timer_Structure			<input checked="" type="checkbox"/>			
Start	BOOL	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0	
forTime	TIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>		T#500ms	

Obrázek 66 - Globální struktury

Nyní, když byly deklarovány důležité prvky řízení, lze se začít zabývat tvorbou jednotlivých vrstev. První podkapitola bude věnována vrstvě robotu, která je však úzce spjatá i s vrstvou procedur. Proto budou tyto dvě podkapitoly sloučeny do jedné.

### Vrstvy robotů

Jedná se o vrstvu či program, který slouží pro komunikaci s robotickým kontrolérem. To znamená, že vše, co je potřeba od robotu vykonat, či o robotu vědět bude prováděno zde, a to i volání procedur robotu, což je nejnižší vrstva. Vše, co bude obsahem této podkapitoly bylo potřeba provést pro každý z robotů.

### Příprava

Prvním krokem byla deklarace vytvořené datové struktury (viz Obrázek 67), které bylo přidělené jméno na základě řízeného robotu. Tato proměnná pod sebou skrývá i funkční blok z knihovny ROBOTemplate, který byl přidán při tvorbě datové struktury. Skrze tento funkční blok bude ve své podstatě řízena celá robotická vrstva. Aby docházelo ke komunikaci mezi deklarovaným funkčním blokem a žádaným robotem při řízení více jak jednoho robotu, je potřeba nastavit komunikaci pro každý deklarovaný funkční blok. Jelikož tato vrstva je samostatně běžící program na procesoru PLC systému, je i zde inicializační, cyklická a ukončovací část programu. Proto bylo využito inicializační vrstvy k nastavení komunikačních parametrů (viz Obrázek 68).

Name	Type	& Reference	Constant	Retain	Replicable	Value
IRB1300	Robot_Structure	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Delay	TON	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Timer	Timer_Structure	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Obrázek 67 - Deklarace lokálních proměnných pro robot IRB1300. Proměnná IRB1300 představuje deklaraci vytvořené struktury v předchozím kroku. Skrze tuto strukturu lze nyní přistupovat k ROBOTemplate knihovně.

Jedná se o přidělení unikátních portů virtuálnímu kontroléru robotu a nastavení komunikačního kanálu, který musí odpovídat kanálu zadanému do kontroléru robotu v RobotStudio, jako tomu bylo v kapitole „Příprava robotu pro řízení z PLC“. Tento krok vede ke schopnosti navázat komunikaci s roboty a odlišit je tak od sebe. Toho všeho bylo dosaženo s pomocí deklarovaného funkčního bloku ROBOTemplate. Posledním důležitým krokem bylo vložit do cyklické části vrstvy načítání funkčního bloku, aby mohlo dojít obousměrnému propisování změn. Ukončovací část programu lze vidět na obrázku (viz Obrázek 69).

```
PROGRAM _INIT
(* ----- client (PLC) ports -----
(* For IRB1300 controller *)
IRB1300.RBTCtrl.Parameters.PortClient[0] := 12010;
IRB1300.RBTCtrl.Parameters.PortClient[1] := 12020;
IRB1300.RBTCtrl.Parameters.PortClient[2] := 12030;

(* ----- server (robot controller) ports -----
(* set ports for IRB1300 instance of RobotCtrl *)
IRB1300.RBTCtrl.Parameters.PortServer[0] := 3500;
IRB1300.RBTCtrl.Parameters.PortServer[1] := 3510;
IRB1300.RBTCtrl.Parameters.PortServer[2] := 3520;

IRB1300.RBTCtrl(Enable := TRUE);
END_PROGRAM
```

Obrázek 68 - Inicializační část kódu pro robot IRB1300. Zde probíhá nastavení komunikaci s virtuálním kontrolérem.

```
PROGRAM _EXIT
IRB1300.RBTCtrl(Enable := FALSE);
END_PROGRAM
```

Obrázek 69 - Ukončovací část programu pro případ, kdy je PLC vypínáno

Jelikož je to jediná vrstva, která má deklarovaný funkční blok ROBOTemplate, jsou odsud volány i pohybové procedury robotu. Jsou to naprogramované trajektorie uvnitř robotického kontroléru. Pohybové procedury byly vytvořeny přes takzvaný „Action File“ (dále jen akce). Akce na rozdíl od funkcí nemusejí mít vstupní parametry. Lze tak zachovat přehledný a jednoduchý kód bez nadbytečného vyplňování funkce. Co je však obsahem této akce? Jak bylo zmíněno v podkapitole „Tvorba trajektorie“, jednotlivé manipulační operace byly rozděleny do několika dílčích trajektorií, a to na základě požadavku sevření gripperu. V akcích jsou dílčí trajektorie volány podle jména trajektorie v RobotStudio chronologicky. Tam, kde došlo k přerušení trajektorie ať už kvůli sevření gripperu či výměníku, lze doplnit požadovanou funkci, která je ovládána přes PLC. Jelikož každá požadovaná akce má nějaké mírné opoždění, jako je například sevření gripperu, byl každý požadavek doplněn o časování, které brání akci přejít do dalšího stavu před uplynutím stanoveného času. Akce obsahují tedy veškerou logiku, která je nutná pro dokončení jejich

činnosti. A protože se vždy jedná o specifickou činnost, jako je sevření gripperu či sepnutí výměníku, u kterých je jasným předpokladem, kdy se mají u dané procedury vykonat, nebylo důvodem vytahovat tato ovládání a jejich proměnné do vyšších vrstev.

Jednotlivé akce jejichž obsahem jsou zkompletované pohybové procedury s logikou řízení uchopovacích prvků, jsou vypsány do jednoduchého bloku *CASE*<sup>2</sup> (do češtiny lze přeložit jako podmíněné větvení), který je možné chápat jako list nahraných procedur robotu. Akce jsou poté z tohoto „listu“ volány z řídicí vrstvy s pomocí globální proměnné „*PrgSet\_robotName*“, kterou lze chápat jako ID zkompletované procedury. Příklad procedury lze vidět níže (viz Obrázek 70).

```

ACTION T1_Take:

CASE IRB1300.ProcedureStep OF

0:
done_IRB1300 := FALSE;
IRB1300.ProcedureStep := 1;

1: (* start procedure from loaded module *)
IF IRB1300.RBTCtrl.RobotStatus[0].ProcDone THEN

    IRB1300.RBTCtrl.RobotInput[0].StartProcedure := FALSE;
    IRB1300.Tool_ID := 1;
    IRB1300.Tool_Release := FALSE;
    IRB1300.ProcedureStep := 2;

ELSE
(* set speed override to 100 % *)
IRB1300.RBTCtrl.RobotInput[0].SpeedOverride := 100;
IRB1300.RBTCtrl.RobotInput[0].ProcedureName := 'T1_1_Take_Approach';
IRB1300.RBTCtrl.RobotInput[0].StartProcedure := TRUE;
END_IF;

2:

IF NOT Timer.Start THEN
    Timer.Start := TRUE;
    IRB1300.Tool_Close := TRUE;
ELSIF Delay.Q THEN
    Timer.Start := FALSE;
    IRB1300.ProcedureStep := 3;
END_IF;

3: (* start procedure from loaded module *)
IF IRB1300.RBTCtrl.RobotStatus[0].ProcDone THEN
    IRB1300.RBTCtrl.RobotInput[0].StartProcedure := FALSE;
    IRB1300.ProcedureStep := 4;
ELSE
(* set speed override to 100 % *)
IRB1300.RBTCtrl.RobotInput[0].SpeedOverride := 100;
IRB1300.RBTCtrl.RobotInput[0].ProcedureName := 'T1_2_Take_Exit';
IRB1300.RBTCtrl.RobotInput[0].StartProcedure := TRUE;
END_IF;

4:
IRB1300.Tool_Close := FALSE;
PrgSet_IRB1300 := 0;

END_CASE;

```

Obrázek 70 - Příklad akce pro volání pohybových procedur uvnitř kontroléru robotu. Z Příkladu je patrné, že je do akce zakomponována i logika.

<sup>2</sup> Jedná se o vícecestné větvení. Proměnná za *CASE* je postupně srovnávána s hodnotami jednotlivých případů. Pod každým případem může být napsán nějaký kód. Při shodě hodnoty za *CASE* s hodnotou případu dojde ke čtení obsahu.

Na začátku každé volané akce dojde k resetování globální proměnné „*Done\_robotName := false*“, která dává zpětnou vazbu do řídicí vrstvy. Tím je sděleno stavovému automatu, že akce ještě nebyla dokonána. Akcí je poté postupně krokováno změnou lokální proměnné „*PrgSet\_robotName*“, která je s každou další dokončenou procedurou zvýšena. Na konci akce je vykonáno resetování „*PrgSet\_robotName*“, které navrátí v robotické vrstvě stav robotu do nulového stavu „*Done\_robotName := TRUE*“. Tím dojde k zabránění opakovanému načítání předchozí akce. Celý kód pro cyklickou komunikaci s robotem lze vidět na obrázku (viz Obrázek 71).

```

PROGRAM _CYCLIC

  IRB1300.RBTCtrl();
  Delay(IN := Timer.Start, PT := Timer.forTime);
  IRB1300.Busy := NOT done_IRB1300;
  IRB1300.RBTCtrl.ErrorReset := Error_Reset_IRB1300;
  Error_IRB1300 := IRB1300.RBTCtrl.RobotStatus[0].ErrorCode;

  IF IRB1300.RBTCtrl.Error OR IRB1300.RBTCtrl.RobotStatus[0].Error THEN
    PrgSet_IRB1300 := 255;
  END_IF;

  CASE PrgSet_IRB1300 OF
    0:
      //Waiting
      IRB1300.ProcedureStep := 0;
      done_IRB1300 := TRUE;
    10:
      T1_Take;
    15:
      Pick_Electro;
    20:
      Pick_Axles;
    25:
      Pick_Frame;
    30:
      T1_Return;
    35:
      T2_Take;
    40:
      Front_Axle;
    45:
      Rear_Axle;
    50:
      T2_Return;
    55:
      Return_Frame;
    60:
      Return_Axles;
    65:
      Return_Electro;
    255:
      brsmemset(ADR(IRB1300.RBTCtrl.ControllerInput), 0, SIZEOF(IRB1300.RBTCtrl.ControllerInput));
      brsmemset(ADR(IRB1300.RBTCtrl.RobotInput[0]), 0, SIZEOF(IRB1300.RBTCtrl.RobotInput[0]));
  END_CASE;

END_PROGRAM

```

Obrázek 71 - Cyklický program pro robot IRB1300

### Řídicí vrstva

Poslední částí softwaru je řídicí vrstva, kterou lze chápat jako jednoduchý stavový automat (viz. Obrázek 73). Nejdříve bylo potřeba definovat proměnné, které lze vidět na obrázku (viz Obrázek 72). Na základě získaných informací z linky podniká naprogramované kroky. V této vrstvě jsou nalezeny čtyři druhy CASE. Ač se to může zdát složité, tak opak je

pravdou. Tři CASE jsou pro volání pohybových procedur s pomocí „PrgSet\_robotName“ a jeden CASE je již stavový diagram.

Jelikož řídicí vrstva nemá deklarovaný funkční blok ROBOTemplate, jsou zde vytvořeny CASE pro řízení robotů (viz Obrázek 74). Zdá se to být nadbytečné a zcela jistě by se dali zahrnout přímo do vrstvy robotu, ale cílem bylo vytvořit vrstvy co možná nejméně propletené. Proto jsou v řídicím programu vypsané CASE pro volání ID pohybových procedur robotu. Mají však tu odlišnost, že umožňují sekvenční procházení mezi jednotlivými procedurami na základě hodnoty „PRG\_RobotName“, která je vždy navyšována o jednotku. Pokud je hodnota na konci procedury rovna „1“, stavový automat bude pokračovat na další. Pokud je hodnota rovna „0“, tak stavový automat zastaví čtení další procedury a vrátí robot do stavu čekajícího na další příkaz, tedy stavu nulového. Předchozí hodnota této proměnné je vždy memorována do proměnné „step\_mem\_RobotName“. Proto při požadavku dalšího spuštění ze stavového automatu stačí „PRG\_RobotName“ porovnat „1“. Nastane tak stav, který zjišťuje, zda neprobíhá čtení procedur. Na základě rozhodnutí s pomocí jednoduché rovnice sepne další sekvenci příkazů pro vykonání procedur.

-MAIN Program-							
StateMachine	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		255
Start_Delay	TON	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
StepDistance	UINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		240
BatchSize	USINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		4
Production_Done	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
Batch	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0
-MAIN Robot step control-							
PRG_SCARA	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
PRG_IRB	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
PRG_YuMi	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
step_mem_IRB	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
step_mem_SCARA	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
step_mem_YuMi	USINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		0
-Shuttles control variables-							
Shuttles_Station_FR	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
Shuttles_Home_FR	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
Shuttles_Station_AX	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
Shuttles_Home_AX	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
Shuttles_Station_EL	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
Shuttles_Home_EL	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
-Shuttles order-							
GoHome	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
Go2Station	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		FALSE
Shuttle_spd	USINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1
-Stations status-							
Finished_Assembly_Electro	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Finished_Assembly_Wheels	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Finished_Assembly_Axles	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Finished_Cell_Deload	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Finished_Cell_Load	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

Obrázek 72 - Lokální proměnné pro řídicí vrstvu

Jednotlivé sekvence jsou již spínány ze stavového automatu, ze kterého jsou i mimo jiné ovládány dopravníky a skok do dalšího sloupce v regálu nakládacím robotem IRB1300. Stavový automat lze

spustit s pomocí jednoho tlačítka na HMI panelu. Nejdříve proběhne inicializace a resetování robotických kontrolérů a proměnných. V další fázi probíhá uvedení linky do pracovního režimu, což znamená, že je provedeno nasazení nástrojů a polohování robotů do pracovních pozic. Poté již probíhá samotné čtení programu pro řízení linky. Přechod mezi jednotlivými stavy zajišťuje lokální proměnná „*StateMachine*“, která je měněna v daném případě na základě splnění nějakých podmínek, které se pokaždé odlišují. Mohou to být podmínky zastavení robotu, či příjezd vozíku se zásobami. Při tvorbě tohoto kódu ve stavovém automatu bylo cíleno na to, aby nikde nedocházelo k přepisu proměnné na sílu tou samou hodnotou. V angličtině je na to termín „variable forcing“. To bylo zajištěno okamžitým přechodem do dalšího stavu a v případě Boolean<sup>3</sup> proměnných i k jejich resetování.

```

PROGRAM _CYCLIC
//MAIN control block for entire robotic cell
CASE StateMachine OF
0: //Because of simulation, the robots have to be reset fir:
  Error_Reset_IRB1300 := TRUE;
  Error_Reset_SCARA := TRUE;
  Error_Reset_YuMi := TRUE;
  Production_Done := FALSE;

  Finished_Cell_Load := FALSE;
  Finished_Cell_Deload := FALSE;
  Finished_Assembly_Electro := FALSE;
  Finished_Assembly_Wheels := FALSE;
  Finished_Assembly_Axles := FALSE;

  //Start delay, so the robot controllers are able to rese!
  Start_Delay(IN := TRUE, PT := T#1s);
  IF Start_Delay.Q THEN
    StateMachine := 1;
  END_IF;

1: //Reset of reset...
  Error_Reset_IRB1300 := FALSE;
  Error_Reset_SCARA := FALSE;
  Error_Reset_YuMi := FALSE;
  StateMachine := 2;

2: //Cell prepare procedure for all robots
  PRG_IRB := 2;
  PRG_SCARA := 2;
  PRG_YuMi := 2;
  StateMachine := 3;

3:
  IF PRG_IRB = 0 AND PRG_SCARA = 0 AND PRG_YuMi = 0 THEN
    PRG_IRB := 1;
    Batch := Batch + 1;
    StateMachine := 4;

    Finished_Cell_Load := FALSE;
    Finished_Cell_Deload := FALSE;
    Finished_Assembly_Electro := FALSE;
    Finished_Assembly_Wheels := FALSE;
    Finished_Assembly_Axles := FALSE;

  END_IF;

4:
  IF PRG_IRB = 6 THEN
    Go2Station := TRUE;
    StateMachine := 5;
  END_IF;

5:
  Go2Station := FALSE;
  StateMachine := 6;

6:
  IF Shuttles_Station_EL AND Shuttles_Station_FR THEN
    PRG_SCARA := 1;
  END_IF;

  IF Shuttles_Station_AX THEN
    PRG_YuMi := 1;
  END_IF;

  IF PRG_SCARA = 1 AND PRG_YuMi = 1 THEN
    StateMachine := 7;
  END_IF;

7:
  IF PRG_SCARA = 0 AND PRG_YuMi = 0 THEN
    GoHome := TRUE;
    StateMachine := 8;
  END_IF;

8:
  GoHome := FALSE;
  StateMachine := 9;

9:
  IF Shuttles_Home_EL AND Shuttles_Home_FR AND Shuttles_Home_AX THEN
    PRG_IRB := 1;
    StateMachine := 10;
  END_IF;

10:
  IF PRG_IRB = 0 AND Batch < BatchSize THEN
    StepOver := StepOver + StepDistance;
    StateMachine := 3;
  ELSIF PRG_IRB = 0 AND Batch = BatchSize THEN
    StateMachine := 11;
  END_IF;

11:
  Production_Done := TRUE;

```

Obrázek 73 - Stavový automat pro řízení robotického pracoviště

Zároveň je ve stavovém automatu sledováno, kolikátá dávka je právě montována. Dávka je sledována z toho důvodu, aby program PLC mohl provést posunutí robotu o další sloupec, ale také proto, aby v případě dokončení montáže všech kusů zažádal o výměnu regálu.

<sup>3</sup> Datatyp, který má pouze dva možné stavy. A to „true“ a „false“, neboli pravda/nepravda.



```
//BLOCK FOR IRB1300 CONTROL
CASE PRG_IRB OF
0:
  //WAITING
1:
  IF done_IRB1300 THEN
    PRG_IRB := step_mem_IRB + 1;
  END_IF;
2: //CALLED ONLY ON START OF THE ROBOTIC CELL!!
  PrgSet_IRB1300 := 10; //Prepare position
  step_mem_IRB := PRG_IRB;
  PRG_IRB := 0; //NEXT
3:
  PrgSet_IRB1300 := 15; //Pick_Electro
  step_mem_IRB := PRG_IRB;
  PRG_IRB := 1;
4:
  PrgSet_IRB1300 := 20; //Pick_Axles
  step_mem_IRB := PRG_IRB;
  PRG_IRB := 1;
5:
  PrgSet_IRB1300 := 25; //Pick_Frame
  step_mem_IRB := PRG_IRB;
  PRG_IRB := 1;
6:
  PrgSet_IRB1300 := 30; //T1_Return
  step_mem_IRB := PRG_IRB;
  Finished_Cell_Load := TRUE;
  PRG_IRB := 1;
7:
  PrgSet_IRB1300 := 35; //T2_Take
  step_mem_IRB := PRG_IRB;
  PRG_IRB := 0;
8:
  PrgSet_IRB1300 := 40; //Front_Axle
  step_mem_IRB := PRG_IRB;
  PRG_IRB := 1;
```

Obrázek 74 - Část bloku v řídicím programu pro řízení robota IRB1300. S pomocí tohoto bloku jsou volány ID kompletovaných procedur z listu robotické vrstvy

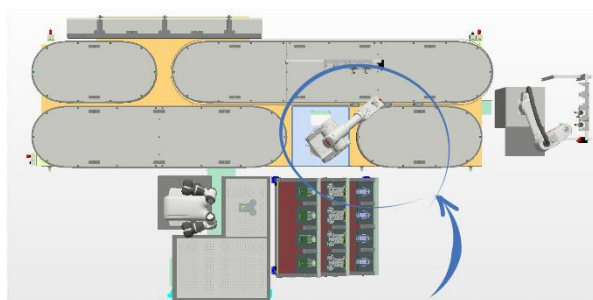
Cílem této kapitoly bylo provést provázání simulačního modelu linky v RobotStudio s virtuálním PLC. Jelikož provázání bylo úspěšné, dosáhlo se tím virtuálního zprovoznění simulačního modelu linky. Následující závěrečná podkapitola bude zaměřena na získání základních dat, které lze ze simulačního modelu čerpat.

## 7 Simulační testování

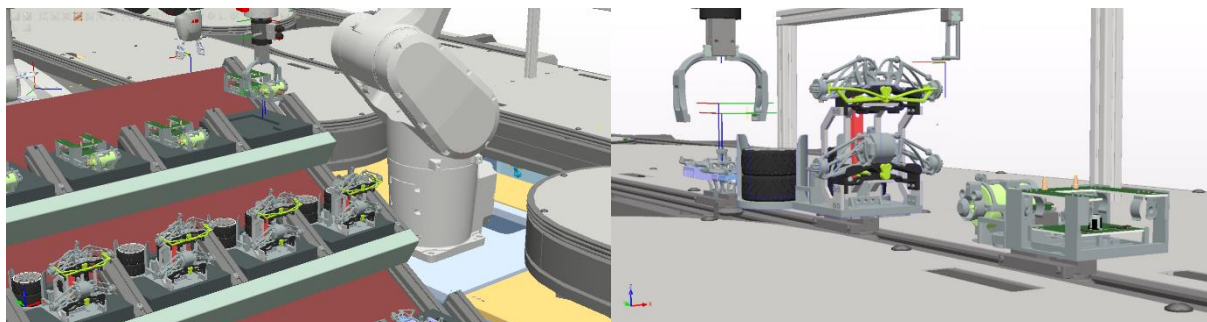
Závěrečná kapitola bude věnována již samotnému výsledku simulování. Nejdříve budou ukázány jednotlivé montážní pracoviště ze simulace, po kterých bude následovat podkapitola o simulačních výstupech, a to ve formě výrobních časů.

### Simulace

Na prvním pracovišti (viz Obrázek 75) probíhalo nakládání/vykládání paletek na dopravník (viz Obrázek 76).



Obrázek 75 - Montážní pracoviště IRB1300



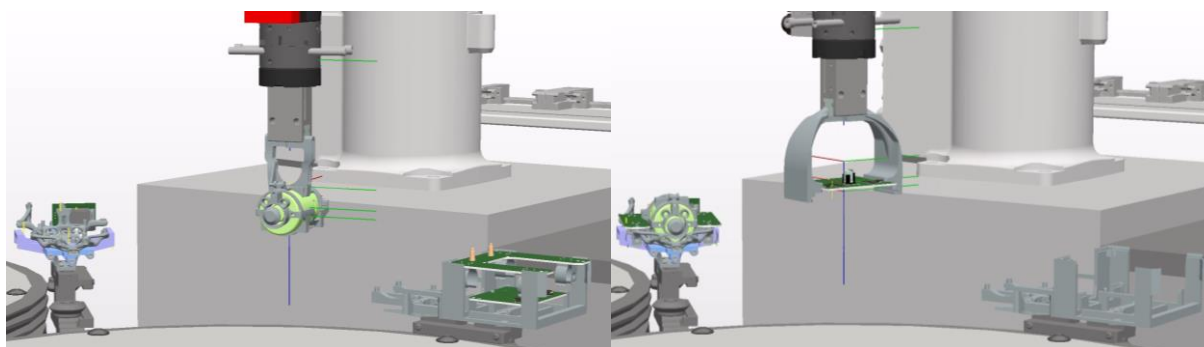
Obrázek 76 - Odběr paletek z regálu – vlevo, odložení paletek na dopravník – vpravo

Manipulace s paletkami byla realizována s pomocí nástroje T1 pro robot IRB1300. Poté, co byly všechny vozíky naloženy, došlo k rozpohybování dopravníku, který zajistí distribuci dílů mezi jednotlivá pracoviště.



Obrázek 77 - Montážní pracoviště IRB910SC

Montáž u robotu SCARA (viz Obrázek 77) začíná ihned po dopravení dílů určených k montáži. Nejdříve byla provedena montáž motorů s nástrojem T1 načež následuje montáž elektroniky s nástrojem T2 (viz Obrázek 78). Tato montáž probíhá současně s robotickým pracovištěm YuMi (viz Obrázek 79).

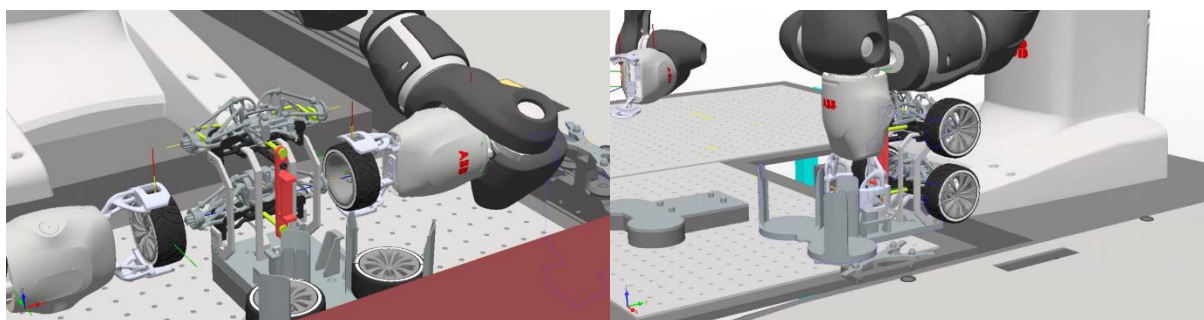


Obrázek 78 - manipulace s motory - vlevo, manipulace s elektronikou - vpravo



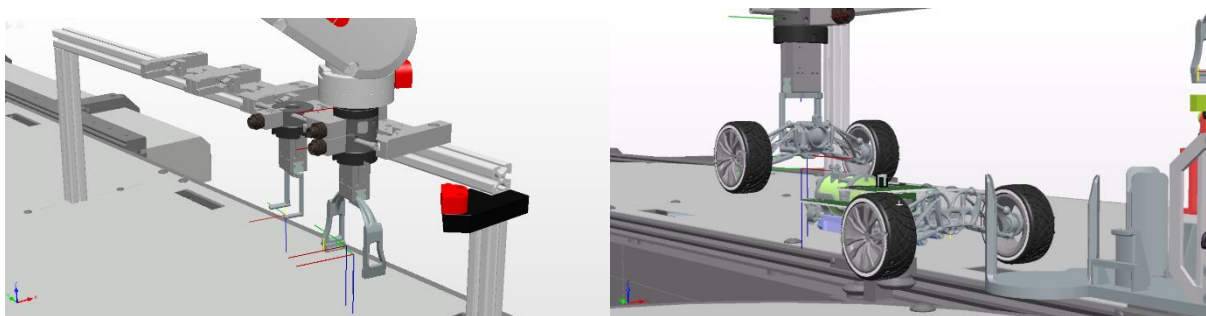
Obrázek 79 - Montážní pracoviště IRB14000

I zde ihned po příjezdu dílů ke stanovišti je proces zahájen nejdříve odběrem paletky z vozíku dopravníku. Tu YuMi umístí na přípravek před sebe a zahájí montáž (viz Obrázek 80).



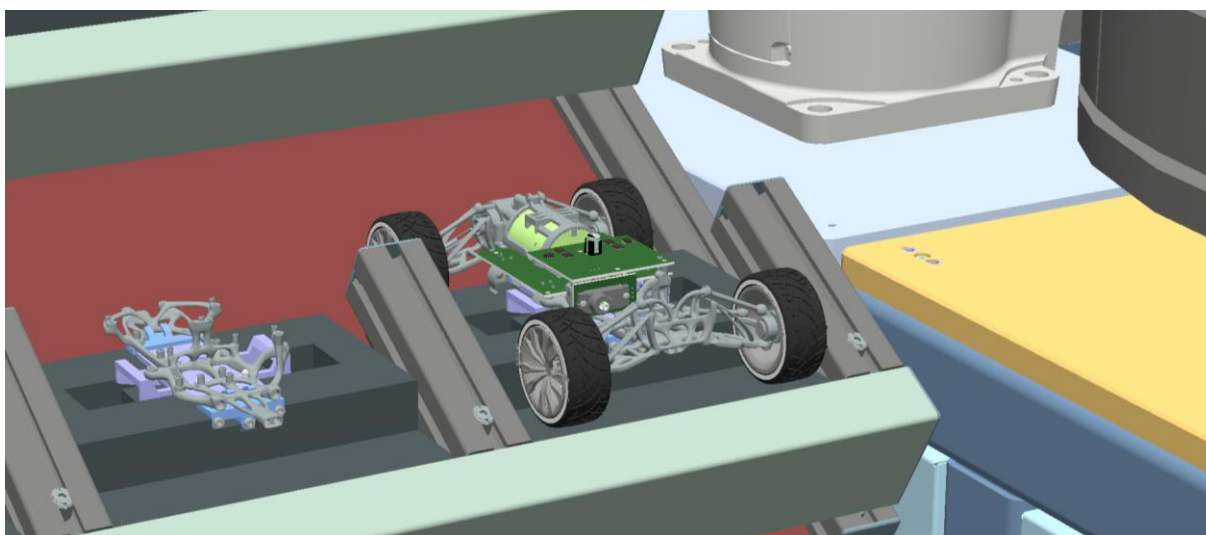
Obrázek 80 - Montáž kol na nápravu – vlevo, vrácení paletky zpět na vozík dopravníku

Poté, co jsou veškeré montážní operace dokončeny, dojde opět k přesunu vozíku do výchozího pracoviště (viz Obrázek 75) k robotu IRB1300, který s pomocí nástroje T2 provede montáž přední a zadní nápravy (viz Obrázek 81).



Obrázek 81 - Výměna nástroje – vlevo, montáž nápravy – vpravo

Po dokončení montáže robot IRB1300 provede opětovnou výměnu nástroje z T2 na T1 a zahájí vykládání palet z dopravníku, aby je mohl nahradit těmi, které ještě nepodstoupily montáž. Na obrázku (viz Obrázek 82) lze vidět již vyložené složené auto.



Obrázek 82 - Dokončená montáž auta na dálkové ovládání

### Výstupní data

Pro získání výstupních dat ze simulace je potřeba provést návrat do softwaru RobotStudio. Zde lze s pomocí integrované funkce „StopWatch“ provést měření, ať už přesunu dopravníku, či výrobního času. Simulaci je pro některé případy měření potřeba připravit s pomocí SmartComponents, se kterými tato funkce umí pracovat. Nastavuje se zde počátek měření a jeho konec na základě zdroje měření. Kterým může být:

- Změna bodu trajektorie robotu
- Na základě vstupního/výstupního signálu SmartComponents
- Zahájení a ukončení simulace

Časy, které byly během simulace naměřeny pro roboty jsou v tabulce níže (viz Tabulka 9).

*Tabulka 9 - Výrobní časy simulace*

<b>Co bylo měřeno</b>	<b>Čas [s]</b>
<b>Výrobní čas IRB1300</b>	17,2
<b>Výrobní čas 910SC</b>	24,432
<b>Výrobní čas YuMi</b>	24,504
<b>Nakládání IRB1300</b>	43,944
<b>Vykládání IRB 1300</b>	43,944
<b>Celkový výrobní čas</b>	152, 052

Byly změřeny i časy přesunů jednotlivých vozíků mezi výchozí pozicí pro naložení, která je v tabulce označena jako domov (viz. Tabulka 10) a místem, kde následně probíhá montáž, v tabulce označené jako stanice.

*Tabulka 10 - Časy pro přesuny mezi stanicemi*

<b>Vozík</b>	<b>Odkud - kam</b>	<b>Čas [s]</b>
<b>Elektronika</b>	Domov - Stanice	0,852
<b>Elektronika</b>	Stanice - Domov	2,52
<b>Rám</b>	Domov - Stanice	1,344
<b>Rám</b>	Stanice - Domov	6
<b>Osy</b>	Domov - Stanice	7,2
<b>Osy</b>	Stanice - Domov	6,156

Z tabulek je patrné, že kdyby při pohybu jednoho výrobku linkou vozíky projížděli stanicemi postupně jako u konvenčního dopravníku, výrobní čas by výrazně vzrostl, jelikož by se vždy muselo čekat na dokončení operace předchozího robotu, aby dopravník mohl popojet na další pracovní pozici. Výrobní čas by tak vycházel cca na 178 sekund. Způsobem okamžité distribuce je výrobní čas od naložení na dopravník až po vyložení zpět do regálu 152 sekund.

Potenciál rychlejší výroby je však ukryt v dopravníku ACOPOStrak, jehož využití bude v budoucnu více zkoumáno. Jde o efektivní využití oválů, které lze využít pro kratší montáže jako buffer, či linku nejdříve naplnit nepostavenými auty a poté provádět distribuci mezi jednotlivá pracoviště. Jelikož výrobní rychlosti jednotlivých pracovišť lze optimalizovat tak, aby výroba probíhala v taktu, díky jejich časové podobnosti. Optimalizačními parametry jsou zde místa, odkud budou vozíky jezdit a také rychlosti robotů.

## 8 Závěr

Cílem této diplomové práce bylo provést návrh předmětu montáže s montážním postupem, na základě kterého byla provedena příprava robotického pracoviště a implementace montáže na robotické pracoviště, a to s využitím simulačních nástrojů, kde výsledkem bylo virtuální zprovoznění. Před touto realizací bylo nejprve důležité provést rešerši zaměřenou na simulační nástroje. Jelikož téma diplomové práce je zaměřené zejména na simulaci robotického pracoviště, má kapitola za cíl přiblížit význam těchto softwarových nástrojů v průmyslu a jejich nejčastější využití s ohledem na technologii a výrobce. Po této kapitole již následuje téma robotiky, kde jsou popsány jednotlivé druhy kinematik a jejich odlišnosti, a to zejména těch robotů, které jsou součástí robotické buňky. U robotů byly popsány jejich benefity, ale zároveň také jejich nevýhody. Aby vše bylo sumarizované, byly doplněny i jejich nejčastější aplikace. Na téma robotiky již následuje programování a řízení robotiky. Zde byla nejdříve vyjasněna základní terminologie, po které již přišlo na řadu téma způsobů programování a opodstatnění těchto způsobů. Kapitola byla zakončena metodami řízení robotiky, kde byla především snaha vysvětlit odlišnosti mezi jednotlivými metodami. Závěrečná část rešeršní práce byla zaměřena již na robotickou buňku samotnou, která se zaměřovala na technologii obsaženou v této buňce a její vybavení.

Praktická část začíná návrhem předmětu montáže, kterým je autíčko na dálkové ovládání vyrobené s pomocí 3D tisku a zakázkových komponent. Návrh předmětu montáže byl prováděn se zaměřením na automatizovanou montáž, která bude prováděna na více robotických pracovištích, kdy část této montáže probíhá i v robotické buňce CIIRCtrack. Toho bylo dosaženo zakomponovanými montážními mechanismy a zjednodušením elektroniky, ze které byla odstraněna veškerá kabeláž vyžadující manuální práci operátorem. Pro tento předmět byl následně navržen montážní postup týkající se již montáže v robotické buňce CIIRCtrack. Na základě navrženého montážního postupu byly provedeny patřičné přípravy a uchopovací prvky s pomocí nichž bude montáž testována.

Po dokončení veškerých příprav byla provedena již samotná příprava virtuálního pracoviště v softwaru RobotStudio, která spočívala ve volbě robotů a jejich pozicování na pracoviště. U robotů musela být provedena také instalace kontrolérů, a to včetně příprav pro komunikaci s PLC. Poté přišel na řadu import veškerých modelů vytvořených v návrhové části práce. Po rozmístění geometrie na základě výrobního postupu byly vytvořeny pomocné souřadné systémy, které sloužily pro účely programování robotů. Pro každý z robotů byly vytvořeny trajektorie. Tyto trajektorie byly dělány po úsecích s ohledem na pozdější provázání s PLC, které bude obstarávat řízení a logiku pracoviště. Na virtuálním pracovišti byla dále doprogramována logika s pomocí doplňku obsaženém v softwaru. S pomocí tohoto doplňku byla vytvořena logika dopravníku. Pracoviště však bylo tvořeno i s ohledem na potenciální využití pro digitální dvojče, nemohlo být tedy využito nástrojů pro simulování fyziky. Proto musely být potřebné vyskytující se fyzikální procesy během simulace manuálně doprogramovány skrze výše zmíněný doplněk v RobotStudio.

Závěrečnou částí byla práce v AutomationStudio, ve kterém probíhalo programování virtuálního PLC pro řízení simulačního modelu v softwaru RobotStudio. V této kapitole byly nejdříve vysvětleny základní deklarace pro programování robotické buňky a představa o architektuře řízení, které bylo rozděleno do tří vrstev a to tak, aby nadřazená řídicí vrstva fungovala zcela odděleně. Na základě navržené architektury bylo realizováno programování dílčích vrstev, kde popis těchto vrstev začínal od střední. Ta obsahovala komunikaci s virtuálním kontrolérem robotu a zároveň této vrstvě byla podřazena vrstva pohybových procedur robotů. Tyto vrstvy byly vykonány pro všechny roboty zvlášť. Nejvyšší vrstva řídicí měla za úkol tyto střední vrstvy provázat s pomocí stavového automatu, který zároveň sloužil k povelování dopravníku. Tento stavový automat byl navržen tak, aby byl řízen vnějšími signály a zpětnou vazbou od jednotlivých robotů. Poslední část byla zaměřena na čerpání dat ve formě výrobních časů. Výsledkem této práce bylo virtuální zprovoznění automatizovaného robotického pracoviště, které bylo řízeno skrze virtuální PLC do virtuálních kontrolérů robotů.

Cíle diplomové práce byly tedy splněny, protože bylo dosaženo úspěšného návrhu předmětu pro automatizovanou montáž, který byl následně implementován do virtuální robotické buňky CIIRTrack. Zde byla provedena simulace montáže na základě navrženého výrobního postupu a algoritmu řízení.

## Seznamy

### Seznam použité literatury

- 11 MCGREGOR, Ian. Automation system simulation and virtual controls testing. In: *International society of automation* [online]. ISA, 2020 [cit. 2022-01-20]. Dostupné z: <https://www.isa.org/intech-home/2020/july-august-2020/features/automation-system-simulation-and-virtual-controls>
- [2] Modeling and simulation. In: *Br-automation* [online]. br-automation [cit. 2022-01-20]. Dostupné z: <https://www.br-automation.com/en-gb/products/software/modeling-and-simulation/>
- [3] Process Simulate manual. In: *Siemens PLM* [online]. [cit. 2022-01-20]. Dostupné z: [https://docs.sw.siemens.com/en-US/product/288782031/doc/PL20201130102000506.tecnomatix\\_eMS\\_sc.xid1015765/html/xid705353](https://docs.sw.siemens.com/en-US/product/288782031/doc/PL20201130102000506.tecnomatix_eMS_sc.xid1015765/html/xid705353)
- [4] Process Simulate Manufacturing process verification in powerful 3D environment. In: *Siemens PLM* [online]. Siemens [cit. 2022-01-20]. Dostupné z: [https://www.plm.automation.siemens.com/en\\_gb/Images/7457\\_tcm642-80351.pdf](https://www.plm.automation.siemens.com/en_gb/Images/7457_tcm642-80351.pdf)
- [5] KONOPA, Miroslav. *Simulation of Production Processes*. ČVUT Fakulta Elektrotechniky, 2013. Diplomová práce. Fakulta Elektrotechniky.
- [6] BRUMSON, Bennet. Scara vs. Cartesian Robots: Selecting the Right Type for Your Applications. In: *Automate* [online]. Automate, 2001 [cit. 2022-01-20]. Dostupné z: <https://www.automate.org/industry-insights/scara-vs-cartesian-robots-selecting-the-right-type-for-your-applications>
- [7] What is a SCARA robot?. In: *Cyan Tec* [online]. Cyan Tec [cit. 2022-01-20]. Dostupné z: <https://cyan-tec.com/laser-systems/when-to-use-a-scara-robot>
- [8] What Can You Do With a SCARA Robot?. In: *RobotDK* [online]. RobotDK, 2019 [cit. 2022-01-20]. Dostupné z: <https://robotdk.com/blog/scara-robot/>
- [9] VAUGHN, Richard. The Difference between Cartesian, Six-Axis, and SCARA Robots. In: *Machinedesign* [online]. MachineDesign, 2013 [cit. 2022-01-20]. Dostupné z: <https://www.machinedesign.com/mechanical-motion-systems/article/21831692/the-difference-between-cartesian-sixaxis-and-scara-robots>



- [10] FAIRCHILD, Mark. Types of Industrial Robots and Their Different Uses. In: *Howtorobot* [online]. howtorobot, 2021 [cit. 2022-01-20]. Dostupné z: <https://www.howtorobot.com/expert-insight/industrial-robot-types-and-their-different-uses>
- [11] HOUŠA, Jaromír. *Stavba výrobních strojů II.* 1. Praha: ČVUT, 1984.
- [12] *Definition and guidelines for collaborative workspaces* [online]. 2017, 44 [cit. 2019-12-17]. Dostupné z: <https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5b6bdb039&appId=PPGMS>
- [13] JÍLEK, Lukáš. *NÁVRH PRACOVIŠTĚ MONTÁŽE* [online]. ČVUT Fakulta Strojní, 2020 [cit. 2022-01-20]. Dostupné z: [https://dspace.cvut.cz/bitstream/handle/10467/90110/F2-BP-2020-Jilek-Lukas-BP\\_Final\\_Lukas%20Jilek.pdf?sequence=-1&isAllowed=y](https://dspace.cvut.cz/bitstream/handle/10467/90110/F2-BP-2020-Jilek-Lukas-BP_Final_Lukas%20Jilek.pdf?sequence=-1&isAllowed=y). Bakalářská práce. Fakulta Strojní. Vedoucí práce Ing. Jiří Kyncl Ph. D.
- [14] BERNIER, Catherine. Delta Robots: The Key to Increasing Manufacturing Speed. In: *Howtorobot* [online]. howtorobot, 2021 [cit. 2022-01-20]. Dostupné z: <https://www.howtorobot.com/expert-insight/delta-robots>
- [15] AMRs vs. AGVs: The Difference Between a Robot and a Guided Vehicle. In: *Fetchrobotics* [online]. fetchrobotics, 2020 [cit. 2022-01-20]. Dostupné z: <https://fetchrobotics.com/fetch-robotics-blog/amrs-vs-agvs-the-difference-between-a-robot-and-a-guided-vehicle/>
- [16] OITZMAN, Mike. What's the difference between an AMR and an AGV?. In: *Mobilerobotguide* [online]. mobilerobotguide, 2021 [cit. 2022-01-20]. Dostupné z: <https://mobilerobotguide.com/2021/08/06/whats-the-difference-between-an-amr-and-an-agv/>
- [17] OWEN-HILL, Alex. What Is the Best Way to Program a Robot?. In: *RobotDK* [online]. RobotDK, 2018 [cit. 2022-01-20]. Dostupné z: <https://robodk.com/blog/program-robot-tips/>
- [18] OWEN-HILL, Alex. The Pros and Cons of 5 Robot Programming Methods. In: *RobotDK* [online]. RobotDK, 2020 [cit. 2022-01-20]. Dostupné z: <https://robodk.com/blog/robot-programming-methods-pros-and-cons/>
- [19] OWEN-HILL, Alex. What Are the Different Programming Methods for Robots?. In: *Robotiq* [online]. robotiq, 2016 [cit. 2022-01-20]. Dostupné z: <https://blog.robotiq.com/what-are-the-different-programming-methods-for-robots>

- [20] Operating manual RobotStudio. In: *ABB* [online]. [cit. 2022-08-13]. Dostupné z: [https://library.e.abb.com/public/e5ad00148905fb58c1257b4b00523751/3HAC032104-001\\_revE\\_en.pdf](https://library.e.abb.com/public/e5ad00148905fb58c1257b4b00523751/3HAC032104-001_revE_en.pdf)
- [21] Inverse Kinematics in Robotics: What You Need to Know. In: OWEN-HILL, Alex. *RobotDK* [online]. RobotDK, 2021, s. 421-426 [cit. 2022-09-07]. Dostupné z: <https://robodk.com/blog/inverse-kinematics-in-robotics-what-you-need-to-know/>
- [22] PLC or No PLC?: Robot Controller Takes on More. In: *Automate* [online]. Automate, 2004 [cit. 2022-01-20]. Dostupné z: <https://www.automate.org/industry-insights/plc-or-no-plc-robot-controller-takes-on-more>
- [23] PLC-Based Robotic Controls Versus OEM Robotic Controls. In: *Techbriefs* [online]. techbriefs, 2015 [cit. 2022-01-20]. Dostupné z: <https://www.techbriefs.com/component/content/article/tb/supplements/mcat/features/articles/21862>
- [24] WICKS, Matt. PLC-based vs. proprietary robotic controls. In: *ControlEngineering* [online]. ControlEngineering, 2014 [cit. 2022-01-20]. Dostupné z: <https://www.controleng.com/articles/plc-based-vs-proprietary-robotic-controls/>
- [25] ROBOTemplate: Programovací interface pro PLC. In: *ABB* [online]. [cit. 2022-08-13]. Dostupné z: <https://new.abb.com/products/robotics/cs/czech-local/partneri-a-systemovi-integratori/robotemplate>
- [26] *Snadná integrace robotů do výrobních buněk díky ROBOTemplate* [online]. [cit. 2022-08-13]. Dostupné z: <https://www.ncp40.cz/aktuality/snadna-integrace-robotu-do-vyrobnich-bunek-diky-robotemplate>
- [27] Product specification: IRB 1300-7/1.4. In: *ABB* [online]. ABB, 2022, s. 61-68 [cit. 2022-08-14]. Dostupné z: <https://search.abb.com/library/Download.aspx?DocumentID=3HAC070393-001&LanguageCode=en&DocumentPartId=&Action=Launch>
- [28] Product specification: IRB 910SC. In: *ABB* [online]. ABB, 2021 [cit. 2022-08-14]. Dostupné z: <https://library.e.abb.com/public/9fc170040463477db86bda7065135457/3HAC056431%20PS%20IRB%20910SC-en.pdf>
- [29] Product specification: IRB 14000. In: *ABB* [online]. ABB, 2018 [cit. 2022-08-14]. Dostupné z: <https://library.e.abb.com/public/5f8bca51d2b541709ea5d4ef165e46ab/3HAC052982%20PS%20IRB%2014000-en.pdf?x-sign=t3HHvF5H6qsxBC83e7PHCFa18NswGcGF9Y7f99vUKTWzvvzonYm8rIB5iHb9V6FP>

- [30] TOOL CHANGERS SERIES WWR: WWR40F-B. In: *Zimmer group* [online]. [cit. 2022-08-14]. Dostupné z: <https://www.zimmer-group.com/en/technologies-components/components/handling-technology/tool-changer/pneumatic/series-wwr/products/wwr40f-b>
- [31] TOOL CHANGERS SERIES WWR: WWR40L-B. In: *Zimmer group* [online]. [cit. 2022-08-14]. Dostupné z: <https://www.zimmer-group.com/en/technologies-components/components/handling-technology/tool-changer/pneumatic/series-wwr/products/wwr40l-b>
- [32] 2-JAW PARALLEL GRIPPERS SERIES MGP800: MGP806NC. In: *Zimmer group* [online]. [cit. 2022-08-14]. Dostupné z: <https://www.zimmer-group.com/en/technologies-components/components/handling-technology/grippers/pneumatic/2-jaw-parallel-grippers/series-mgp800/products/mgp806nc>
- [33] STORAGE STATION SERIES ALSR: ALSR1-40-B. In: *Zimmer group* [online]. [cit. 2022-08-14]. Dostupné z: <https://www.zimmer-group.com/en/technologies-components/components/handling-technology/tool-changer/pneumatic/series-wwr/products/alsr1-40-b>
- [34] ENERGY ELEMENT SERIES WER1000: WER1000LSI12-00-B. In: *Zimmer group* [online]. [cit. 2022-08-14]. Dostupné z: <https://www.zimmer-group.com/en/technologies-components/components/handling-technology/energy-elements/electrical/all-energy-elements-with-electrical-transmission/products/wer1000lsi12-00-b>
- [35] ENERGY ELEMENT SERIES WER1000: WER1000FSI12-00-B. In: *Zimmer group* [online]. [cit. 2022-08-14]. Dostupné z: <https://www.zimmer-group.com/en/technologies-components/components/handling-technology/energy-elements/electrical/all-energy-elements-with-electrical-transmission/products/wer1000fsi12-00-b>
- [36] *ACOPOStrak: User's manual*. Březen 2022. B&R Strasse 1 5142 Eggelsberg Austria, 2022, 312 s.

## Seznam použitých softwarů

Siemens NX 2206

ABB RobotStudio

B&R AutomationStudio

## Seznam obrázků

Obrázek 1 –Obsah kapitoly - programovací prvky

Obrázek 2- Obsah kapitoly - Roboty

Obrázek 3 - Přesnost a preciznost

Obrázek 4 - Kartézský robot

Obrázek 5 - SCARA robot

Obrázek 6 - Kloubový robot

Obrázek 7 - Kolaborativní robot

Obrázek 8 - Skill based Task sharing

Obrázek 9 - Delta robot

Obrázek 10 - AMR

Obrázek 11 - Souřadné systémy robotu, ve kterých lze programovat trajektorii

Obrázek 12 –Obsahu kapitoly - Programování a řízení robotů

Obrázek 13 – MoveJ, šipka představuje nástroj robotu..

Obrázek 14 – MoveL, šipka představuje nástroj robotu.

Obrázek 15 - Aproximační zóna trajektorie

Obrázek 16 – Obsah kapitoly – Způsoby programování robotů

Obrázek 17 - FlexPendant zařízení (pro robot IRB1300)

Obrázek 18 – Obsah kapitoly – Metody řízení robotu

Obrázek 19 -Funkce ROBOTemplate

Obrázek 20 - Robotická buňka CIIRctrack

Obrázek 21 -Robot IRB1300

Obrázek 22 - Robot SCARA IRB910SC

Obrázek 23 - Kolaborativní robot YuMi IRB14000

Obrázek 24 - Volná část výměníku (označení. WWR40L-B), ke kterému lze připevnit nástroj

Obrázek 25 - Stojan s výměníky nástrojů pro robot IRB1300

Obrázek 26 - MGP806NC u robotu IRB910SC

Obrázek 27 - Smart Gripper

Obrázek 28 - Dopravník ACOPOStrak

Obrázek 29 – Výsledný (zatím poslední prototyp) auta na dálkové ovládání

Obrázek 30 - První iterace prototypování auta na dálkové ovládání

Obrázek 31 - Rozložený prototyp auta na dálkové ovládání

- Obrázek 32 - Lože pro přípravky vlevo – zásoby, vpravo rám
- Obrázek 33 - Paletka pro základní rám
- Obrázek 34 - Detail na rozpínací systém paletky
- Obrázek 35 - Paletka pro převoz elektroniky
- Obrázek 36 - Paletka pro přepravu pneumatik a os vozidla
- Obrázek 37 - Podložka pod paletku pro robot YuMi
- Obrázek 38 - Mechanismus pro odemčení os na zadní části paletky
- Obrázek 39 - Uvolněné osy
- Obrázek 40 - Prsty pro robot IRB1300 s ukázkou uchopení paletky
- Obrázek 41 - Nástroj T2 pro robot IRB1300
- Obrázek 42 - Nástroj T1 pro robot 910SC
- Obrázek 43 - Nástroj T2 pro robot 910SC
- Obrázek 44 - Prsty pro uchopení pneumatik pro robot YuMi
- Obrázek 45 - Schéma pracoviště. Červená šipka označuje vjezd pro AMR.
- Obrázek 46 - Vytvořený mechanismus pro Gripperu IRB910SC, který umožňuje fungování gripperů
- Obrázek 47 - Vytvořený model se souřadnými systémy pro odběry různými nástroji. Jedná se o systémy, které cestují s paletkou.
- Obrázek 48 - Wobj pro robot SCARA
- Obrázek 49 Výsledek příprav simulačního modelu
- Obrázek 50 - Vytvořené body v prostoru vázající se k wobj. Z těchto bodů bude následně vytvořena trajektorie
- Obrázek 51 - Vytvořené trajektorie pro montáž
- Obrázek 52 - Rozdělení pohybových procedur jako příprava pro řízení z PLC
- Obrázek 53 - Vygenerovaný kód pro robot IRB910SC. Jedná se o program pro odběr baterie
- Obrázek 54 - Nastavení komunikace pro: IRB1300, IRB910SC, YuMi L/R
- Obrázek 55 - Trajektorie dopravníku
- Obrázek 56 - Nejvyšší vrstva ovládání dopravníku
- Obrázek 57 - Ovládání Gripperu
- Obrázek 58 - Logika bloku pro výměnu nástroje
- Obrázek 59 - Logika volby nástroje
- Obrázek 60 - Logika uchycení detekované součásti k detekovanému tělesu
- Obrázek 61 - Žluté čáry představují senzory, které detekují součást k uchycení k rámu
- Obrázek 62 - Hromadně volaný blok pro uchycení elektronických komponent k rámu
- Obrázek 63 - Schéma softwarové architektury pro řízení pracoviště
- Obrázek 64 - Programové vrstvy. Programy s názvy robotů pod sebou skrývají akce. Dále každá z vrstev má své proměnné, cyklický, inicializační a ukončovací část programu.
- Obrázek 65 - Globální proměnné pro řízení pracoviště

Obrázek 66 - Globální struktury

Obrázek 67 - Deklarace lokálních proměnných pro robot IRB1300. Proměnná IRB1300 představuje deklaraci vytvořené struktury v předchozím kroku. Skrze tuto strukturu lze nyní přistupovat k ROBOTemplate knihovně.

Obrázek 68 - Inicializační část kódu pro robot IRB1300. Zde probíhá nastavení komunikaci s virtuálním kontrolérem.

Obrázek 69 - Ukončovací část programu pro případ, kdy je PLC vypínáno

Obrázek 70 - Příklad akce pro volání pohybových procedur uvnitř kontroléru robotu. Z Příkladu je patrné, že je do akce zakomponována i logika.

Obrázek 71 - Cyklický program pro robot IRB1300

Obrázek 72 - Lokální proměnné pro řídicí vrstvu

Obrázek 73 - Stavový automat pro řízení robotického pracoviště

Obrázek 74 - Část bloku v řídicím programu pro řízení robotu IRB1300. S pomocí tohoto bloku jsou volány ID kompletovaných procedur z listu robotické vrstvy

Obrázek 75 - Montážní pracoviště IRB1300

Obrázek 76 - Odběr palet z regálu – vlevo, odložení palet na dopravník – vpravo

Obrázek 77 - Montážní pracoviště IRB910SC

Obrázek 78 - Manipulace s motory - vlevo, manipulace s elektronikou - vpravo

Obrázek 79 - Montážní pracoviště IRB14000

Obrázek 80 - Montáž kol na nápravu – vlevo, vrácení paletky zpět na vozík dopravníku

Obrázek 81 - Výměna nástroje – vlevo, montáž nápravy – vpravo

Obrázek 82 - Dokončená montáž auta na dálkové ovládání

## Seznam tabulek

Tabulka 1 - Kompletní kusovník

Tabulka 2 - Kusovník

Tabulka 3 – Obecný montážní list

Tabulka 4 - Vstupy a výstupy dopravníku

Tabulka 5 - Vstupy a výstupy pro ovládání gripperu

Tabulka 6 - Vstupy a výstupy pro ovládání výměníku

Tabulka 7 - Logický člen pro ovládání zvoleného nástroje

Tabulka 8 - Vstupy a výstupy pro ovládání uchycení komponent k sestavě

Tabulka 9 - Výrobní časy simulace

Tabulka 10 - Časy pro přesuny mezi stanicemi

## Seznam příloh

### Textové přílohy

Příloha č. 1 – Komunikace AutomationStudio a RobotStudio

### Grafické přílohy

Příloha č. 1 – Logika robotické buňky

Příloha č. 2 – Logika dopravníku ACOPOStrak

Příloha č. 3 – Prototyp paletky

Příloha č. 4 – HMI

### Elektronické přílohy

Příloha č. 1 – text diplomová práce

Příloha č. 2 – Videozáznam simulace montáže

Příloha č.3 – Projekt AutomationStudio

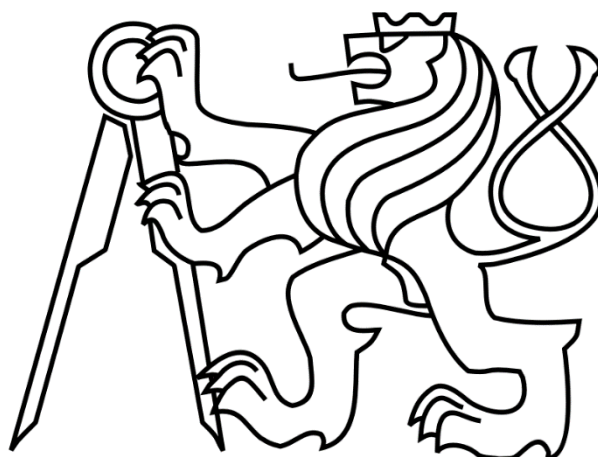
Příloha č. 4 – Projekt RobotStudio

Příloha č. 5 – Vytvořené modely

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení

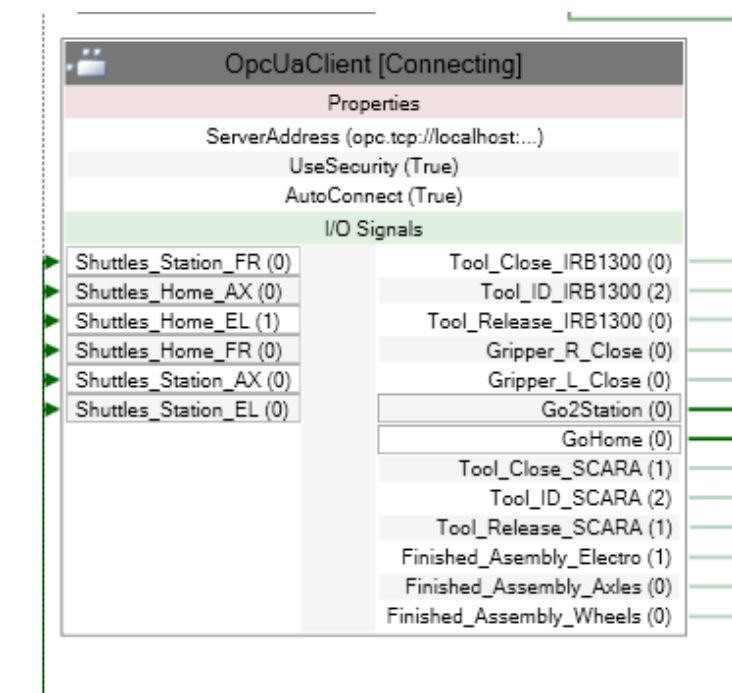


# Diplomová práce

Textové přílohy



## Příloha č. 1 – Komunikace AutomationStudio a RobotStudio



Obr 1 - Komunikační blok v RobotStudio

Proměnné vytvořené v AutomationStudio je potřeba zviditelnit pro OPC-UA komunikaci v konfiguraci (viz Obr 2). Komunikace byla nastavena přes OPC-UA i RobotStudio (viz Obr 1), kde byla přidána Smart komponenta „OpcUaClient“. Vytvořené proměnné se poté přidají do RobotStudio buď jako vstup či výstup.



The image shows the configuration interface of RobotStudio. On the left, a tree view under 'Configuration' shows various system components, with 'OpcUaM' highlighted. On the right, a table lists global variables and their data types.

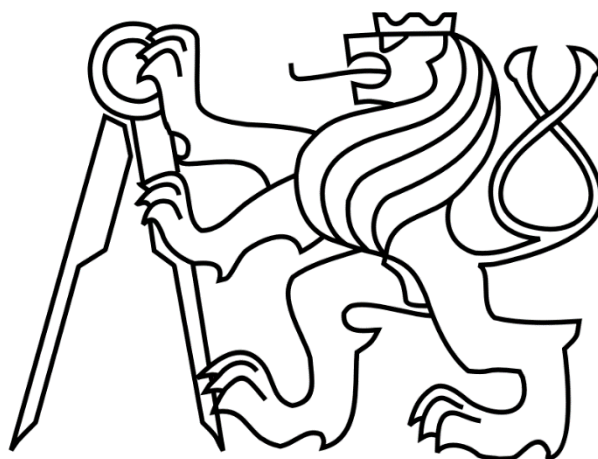
Name	Datatype
Default View	
<Default>	
Global Variables	
Controller	
Batch	USINT
BatchSize	USINT
Finished_Asembly_Electro	BOOL
Finished_Assembly_Axles	BOOL
Finished_Assembly_Wheels	BOOL
Finished_Cell_Deload	BOOL
Finished_Cell_Load	BOOL
Go2Station	BOOL
GoHome	BOOL
PRG_IRB	USINT
PRG_SCARA	USINT
PRG_YuMi	USINT
Production_Done	BOOL
Shuttle_spd	USINT
Shuttles_Home_AX	BOOL
Shuttles_Home_EL	BOOL
Shuttles_Home_FR	BOOL
Shuttles_Station_AX	BOOL
Shuttles_Station_EL	BOOL
Shuttles_Station_FR	BOOL
Start_Delay	TON
StateMachine	USINT
step_mem_IRB	USINT
step_mem_SCARA	USINT
step_mem_YuMi	USINT
StepDistance	UINT
IRB1300	
910SC	
YuMi	
TooDetache	

Obr 2 - Nastavení proměnných tak, aby je bylo možné přes OPC-UA vidět v RobotStudio

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení

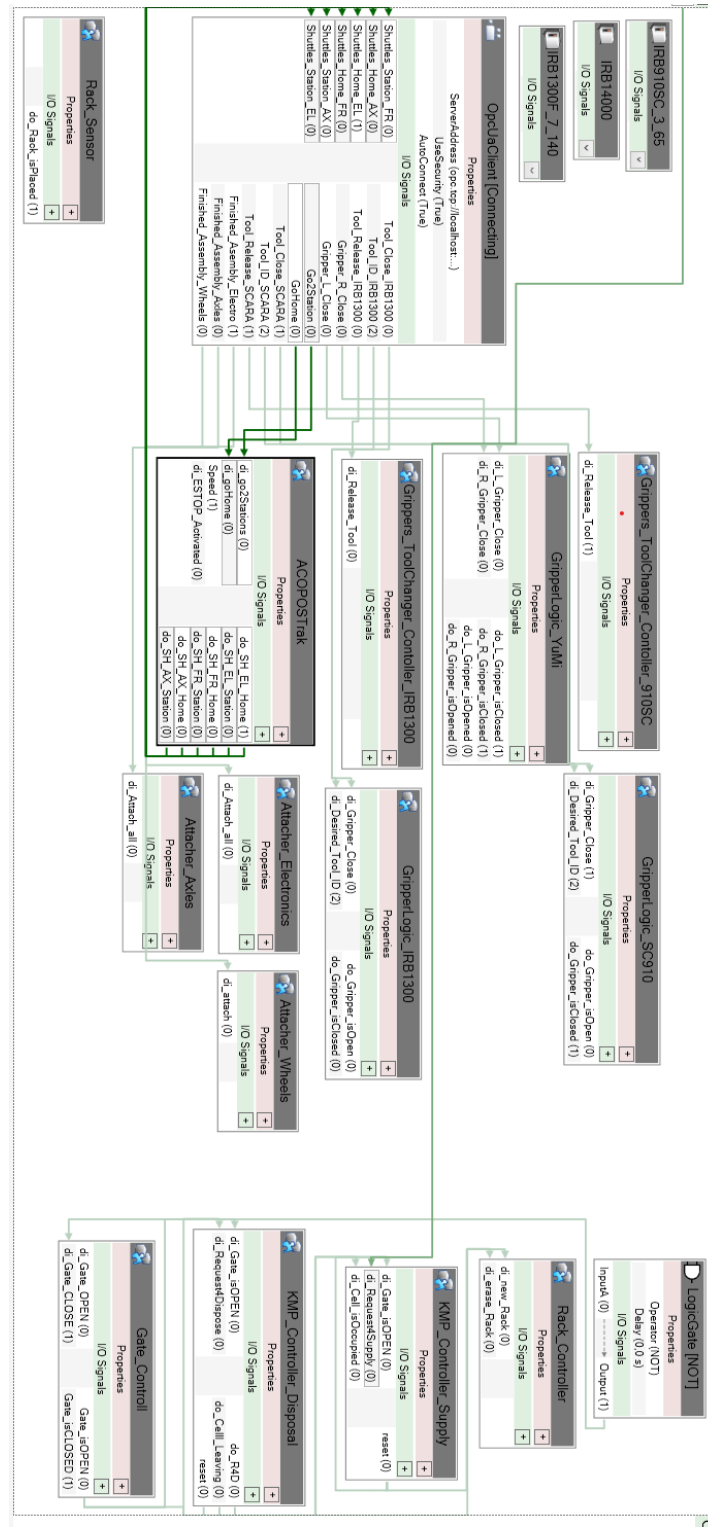


# Diplomová práce

Grafické přílohy

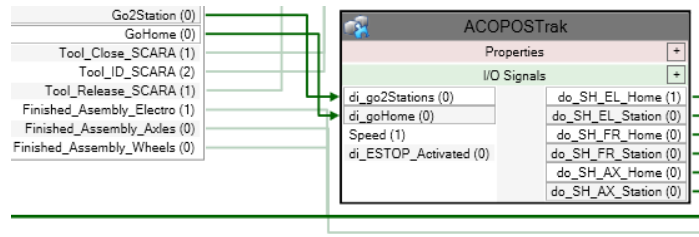


Příloha č. 1 – Logika robotické buňky

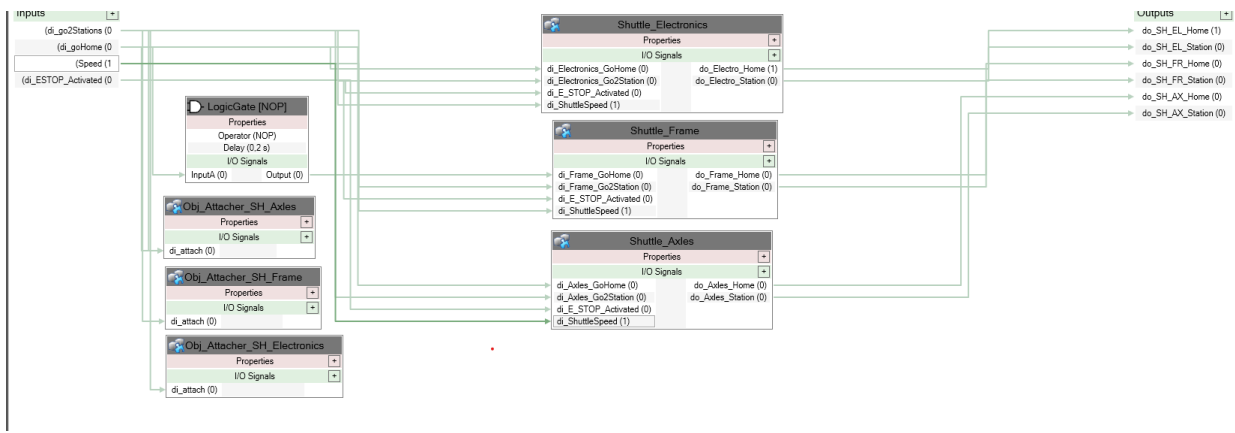


Obr 3 - Logika robotické buňky "Station logic"

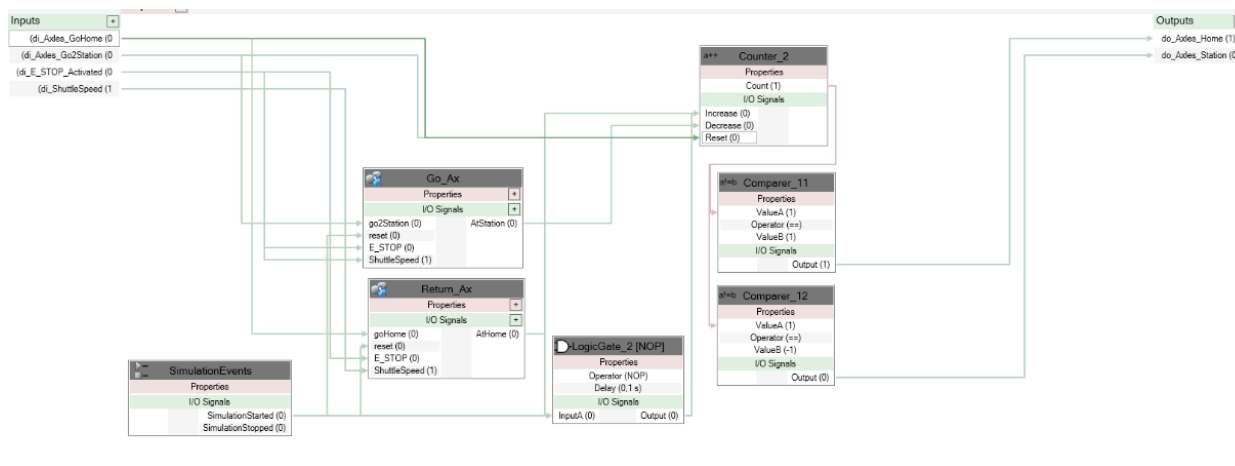
## Příloha č. 2 – Logika dopravníku ACOPOSTrak



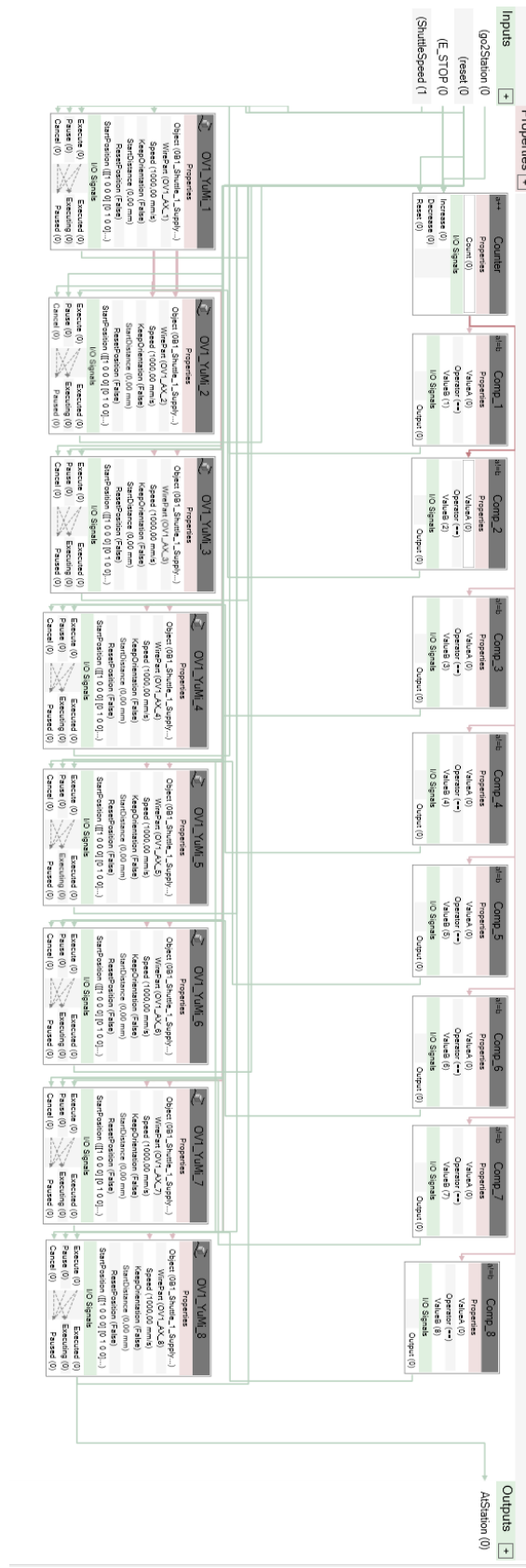
Obr 4 - Nejvyšší vrstva dopravníku



Obr 5 - Uvnitř nejvyšší vrstvy. Jsou zde bloky pro připevnění, zpoždění vyjetí vozíku a hlavně jednotlivé vozíky

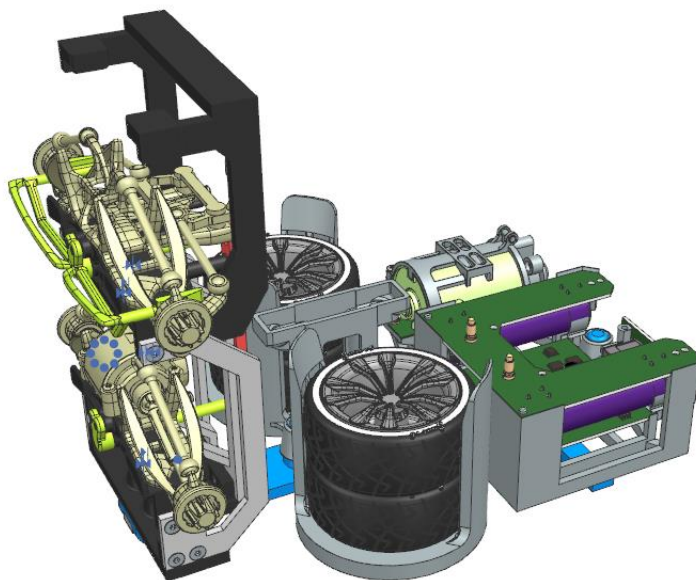


Obr 6 - Logika uvnitř vozíku pro přepravu náprav. Je zde příkaz na cestu do stanice a na cestu ze stanice. Dále je zde signalizace aktuální pozice, která je vytažena zpět do AutomationStudio přes OPC-UA.



Obr 7 - Nejnižší úroveň logiky vozíku, která slouží pro rozjezd po trati, rychlost, zastavení a čtení trati.

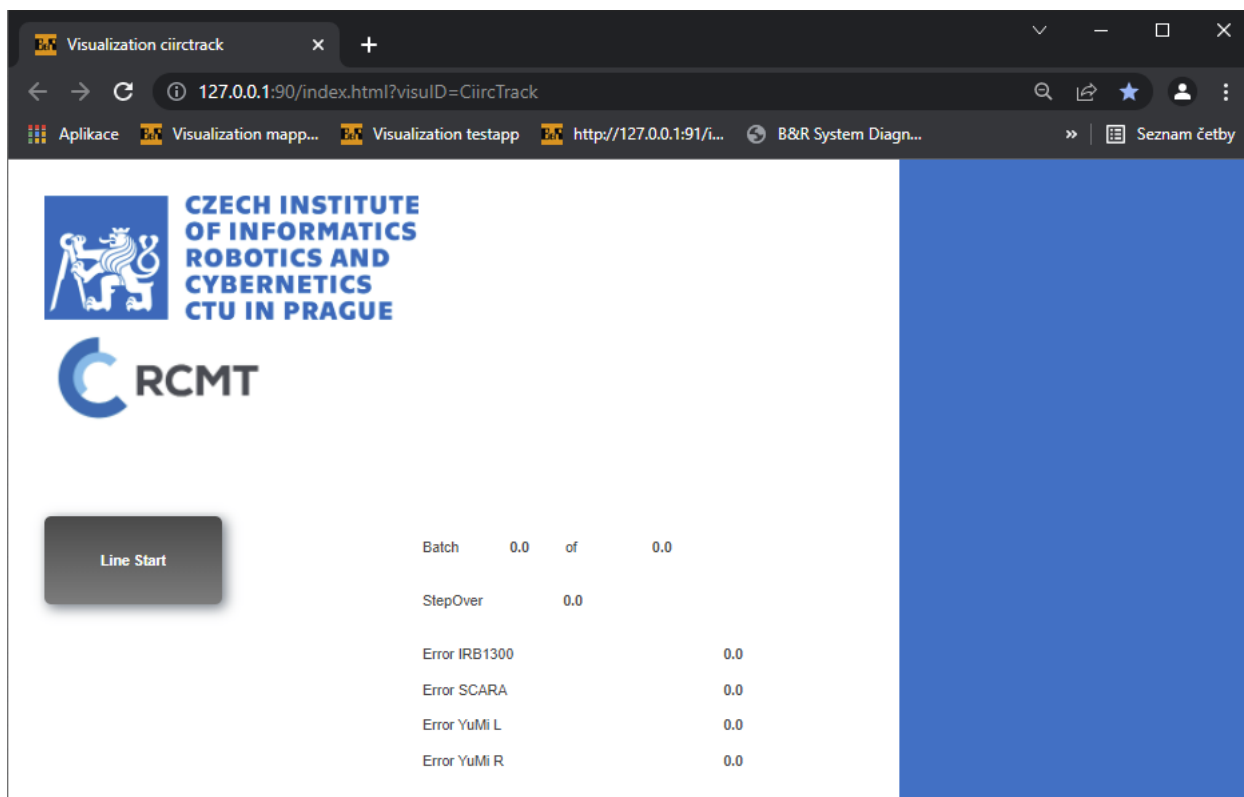
Příloha č. 3 – Prototyp přípravku



---

*Obr 8 - Prototyp dvoudílné paletky, která by dokázala nést všechny díly*

## Příloha č. 4 – HMI



Obr 9 - Jednoduchý HMI panel ve webovém prohlížeči pro spuštění robotické buňky