



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## ROBOTICKÉ ZAKLÁDÁNÍ DO OBRÁBĚCÍHO STROJE

ROBOTIC INSERTION INTO A MACHINE TOOL

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Matěj Pála

### VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Petr Pivoňka, CSc.

BRNO 2019



# Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**  
Ústav automatizace a měřicí techniky

**Student:** Bc. Matěj Pála

**ID:** 173718

**Ročník:** 2

**Akademický rok:** 2018/19

## NÁZEV TÉMATU:

### Robotické zakládání do obráběcího stroje

#### POKyny PRO VYPRACOVÁNÍ:

1. Zpracujte rešerši týkající se možnosti vzájemné komunikace a ovládání robota - PLC.
2. Seznamte se s řídicími prostředky společnosti B&R Automation a vývojovým prostředím Automation Studio.
3. Navrhnete modulární koncepci softwarového řešení robotického zakládání obrobku do obráběcího stroje.
4. Implementujte řídicí algoritmus do řídicího systému PLC.
5. Doplněte existující vizualizaci dotykového panelu.
6. Ověřte funkčnost na reálném pracovišti.

#### DOPORUČENÁ LITERATURA:

[1] Firemní literatura B&R: Automation Studio V4

[2] Firemní literatura B&R: Produkty pro řízení pohybu a vizualizace společnosti B&R

**Termín zadání:** 4.2.2019

**Termín odevzdání:** 13.5.2019

**Vedoucí práce:** prof. Ing. Petr Pivoňka, CSc.

**Konzultant:** Ing. Tomáš Matucha, PhD.

**doc. Ing. Václav Jirsík, CSc.**  
*předseda oborové rady*

#### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Diplomová práce se zabývá problematikou zakládání a přesunu různých obrobků a profilů. V první části je uvedeno rozdělení robotů, popis jejich kinematiky i samotného zakládání. Druhá část práce se zabývá analýzou a rešerší možností komunikace řídicí jednotky B&R s robotem firmy ABB, včetně popisu zařízení na pracovišti. Dále je představen návrh koncepce programu řízení robota a také způsob řízení komunikace mezi jednotkami. Na konci práce je uvedeno testování a ověřování zmíněného konceptu.

## **Klíčová slova**

Robotické zakládání, B&R APC, ABB Robot, B&R Automation, Automation Studio V4

## **Abstract**

Semestral thesis deals with the process of an insertion and movement of different workpieces and profiles. In the first, theoretical, part, the definition of types of robots and description of their kinematics are presented. The second part deals with the analysis and research of the possibilities of communication between the B&R control unit and the ABB robot, including a description of the equipment in the workplace. In addition, the concept of the robot control program and mainly the management of communication between units are presented. At the end of the work is the description of testing and verification of the concept.

## **Key words**

Robotic insertion ,B&R APC, ABB Robot, B&R Automation, Automation Studio V4

## **Bibliografická citace**

PÁLA, M. *Robotické zakládání do obráběcího stroje*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2019. 52 s. Vedoucí semestrální práce prof. Ing. Petr Pivoňka, CSc.

## **Prohlášení**

„Prohlašuji, že svou závěrečnou práci na téma Robotické zakládání do obráběcího stroje jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne **12. května 2019**

.....

podpis autora

## **Poděkování**

Tímto děkuji vedoucímu mé diplomové práce Prof. Ing. Petru Pivoňkovi, CSc. za účinnou metodickou, pedagogickou pomoc a konzultace při zpracování mé diplomové práce. Dále děkuji firmě B+R automatizace, spol. s.r.o. v Brně, jmenovitě zejména Ing. Tomáši Matuchovi Ph.D. za účinnou odbornou pomoc a další cenné rady při zpracování této práce. V neposlední řadě děkuji také firmě ABB v Praze Vestci, konkrétně panu Tomáši Brojírovi za odbornou pomoc a konzultace.

V Brně dne **12. května 2019**

.....  
podpis autora

# Obsah

1	Úvod.....	11
2	Průmyslový robot.....	12
2.1	Klasifikace průmyslových robotů .....	12
2.2	Kinematika průmyslových robotů.....	13
2.3	Řízení a Učení pohybu průmyslových robotů.....	14
2.4	Souřadný systém robotů se sériovou kinematikou.....	14
2.5	Robotické zakládání .....	14
3	Konfigurace pracoviště .....	16
3.1	Obráběcí centrum .....	16
3.2	Měřicí stanice .....	17
3.3	Robot.....	18
3.4	Zásobník.....	18
4	Komunikace .....	19
4.1	Hardwarové prostředky .....	19
4.1.1	Průmyslové PC .....	19
4.1.2	Řídicí jednotka robota.....	20
4.1.3	Porovnání komunikačních modulů .....	21
4.2	Výběr způsobu komunikace .....	21
4.2.1	DeviceNet .....	22
4.2.2	PROFIBUS DP .....	22
4.2.3	Ethernet/IP .....	22
4.2.4	Profinet.....	22
4.2.5	REST.....	23
4.3	Výsledná použitá sběrnice.....	23
5	Návrh koncepce .....	25
5.1	Dosavadní stav .....	25
5.2	Cíle práce .....	25
5.3	Možnosti pohybu robota .....	25
5.4	Struktura programu .....	26
5.5	Programové vybavení robota .....	26
5.6	Basic Modul a jeho teoretický návrh .....	27
5.6.1	Datové struktury komunikace .....	27
5.6.2	Řízení komunikace .....	29
5.6.3	Startovací sekvence robota .....	29
5.7	Hierarchie užití funkčních bloků.....	32
6	Realizace.....	33
6.1	Prvotní návrh algoritmu .....	33
6.2	Rozdíl realizace oproti návrhu .....	33

6.3	Změna použité sběrnice.....	34
6.4	Řízení robota .....	35
6.4.1	Řízení komunikace – RBComm FBK .....	35
6.4.2	Řízení startovací sekvence robota – ABBRobotCtrl FBK.....	36
6.4.3	Modul generování programového modulu RAPID kódu robota – ABBRobotPrgGen FBK .....	37
6.4.4	Hlavní modul řízení robota – FBK ABBRobotBasic .....	38
6.5	Učení pozic.....	38
6.6	Spouštění programu .....	39
6.7	Vizualizace.....	39
6.8	Diagnostika robota .....	40
7	Ověřování.....	42
7.1	Komunikace .....	42
7.2	Simulace.....	42
7.3	Řízení robota.....	43
7.4	Zakládání do svěráku .....	44
7.5	Chybové stavy robota.....	45
7.6	Užití manuálního ovládání .....	45
7.7	Web Services a REST Api .....	45
8	Závěr .....	47
	Literatura.....	48
	Seznam symbolů, veličin a zkratk.....	51
	Seznam příloh .....	52

## Seznam obrázků

Obr. 3.1: Vertikální obráběcí centrum MCFV 1680 firmy TAJMAC-ZPS [15].....	16
Obr. 3.2: Měřicí stanice Renishaw Equator 300 se sondou SP 225 [15].....	17
Obr. 3.3: Náhled stacionárního robota ABB IRB 1600 [12] .....	18
Obr. 4.1: Náhled průmyslového PC APC910 se dvěma sloty [11].....	19
Obr. 4.2: Náhled rozvaděčové skříně řídicí jednotky ABB robota IRC5 [13] .....	20
Obr. 5.1: Nová koncepce platformy.....	26
Obr. 5.2: Basic Modul – model komunikační struktury mezi APC a IRC5 (Aktuální data z robota) .....	27
Obr. 5.3: Basic Modul – model komunikační struktury mezi APC a IRC5 (Příkazy statusy robota).....	28
Obr. 5.4: Stavový automat startovací sekvence robota.....	30
Obr. 5.5: Stavové automaty běžící paralelně mimo startovací sekvenci robota .....	31
Obr. 5.6: Základní koncepce programu .....	32
Obr. 6.1: Finální koncepce ABBRobotBasic funkčního bloku .....	34
Obr. 6.2: Funkční blok řízení komunikace RBComm včetně popisu vstupů/výstupů ...	35
Obr. 6.3: Funkční blok řízení startovací sekvence robota ABBRobotCtrl včetně popisu vstupů/výstupů .....	36
Obr. 6.4: Funkční blok pro generování programového modulu s RAPID kódem robota .....	37
Obr. 6.5: Hlavní funkční blok řízení robota ABBRobotBasic včetně popisu vstupů/výstupů .....	38
Obr. 6.6: Náhled datové struktury na pTeaching vstupu ABBRobotBasic FBK .....	39
Obr. 6.7: Náhled části stránky pro spouštění modulu programu robota .....	40
Obr. 6.8: Datová struktura obsahující veškerá diagnostická data o robotu[17].....	41
Obr. 7.1: Náhled budoucího pracoviště (digitální dvojče) v RS.....	43
Obr. 7.2: Náhled robota IRB1200, na kterém byl koncept testován.....	44

## Seznam tabulek

Tab. 4.1: Porovnání podporovaných možností komunikace pro obě řídicí jednotky (APC, IRC5)[12][14] .....	21
---	----

# 1 ÚVOD

V současnosti se v průmyslu, ať už výrobě či obrábění, rozšiřuje možnost použití průmyslových robotů a manipulátorů. To je dáno hlavně požadavkem na zvýšení efektivity výroby a s tím související vyšší produktivitou. Také s ohledem na zachování, či zvýšení kvality výroby a výsledné výrobní ceně. Tyto technologie však musí být nejprve otestovány na různých pracovištích, nebo ve výzkumných centrech, než mohou být uvedeny na trh. Touto problematikou se, stejně jako mnoho dalších firem, zabývá také firma B+R Automatizace s.r.o.

Jedná se o přesun a zakládání různých obrobků do obráběcího stroje v jisté nejmenované firmě. V současnosti je zakládání obrobků a jejich přesun realizován robotem firmy Comau, řízeným výhradně řídicí jednotkou firmy B+R. Na základě požadavku zákazníka je v plánu nahradit Comau robota robotem firmy ABB s vlastní řídicí jednotkou.

Tato práce si klade za cíl vytvořit univerzální modulární platformu pro komunikaci a řízení ABB robota B&R řídicím systémem. Výsledkem by měla být možnost použití jakéhokoli ABB robota pro zmíněnou aplikaci, resp. jedním z hlavních cílů práce je vytvořit univerzální modul řízení ABB robota a integrovat jej do B&R systému. V první řadě je však potřeba provést rešerši a zvolit vhodnou komunikační sběrnici. Bude tedy provedena dostatečně podrobná analýza možností komunikace obou řídicích systémů, neboť volbou použité sběrnice bude pravděpodobně ovlivněn také vývoj již zmíněné integrační platformy. Práce se bude zabývat návrhem řízení komunikace po zvolené sběrnici, včetně programového vybavení. Z požadavků dále vyplývá potřebná analýza způsobu zakládání a řízení pohybu robota. Důležitým faktorem bude také možnost učít robota novým trajektoriím (či pouze bodům) přímo z B&R systému. V práci bude tedy analyzována aktuální koncepce a navrženo nové, modulární řešení. S tím souvisí také stávající vizualizace, která bude pravděpodobně upravena, či rozšířena o další možnosti práce s robotem, potažmo pracovištěm. Nová koncepce bude na zmíněném pracovišti také otestována.

Dále, v souvislosti s Průmyslem 4.0, se práce zaměří na možnost diagnostiky robota, resp. sběrem dat z robota, jejich zpracováním a také možnostmi předání těchto dat nadřazeným systémům. V neposlední řadě bude v práci rovněž analyzován vliv digitálního dvojčete na vývoj celé platformy a jeho využití pro testování.

## 2 PRŮMYSLOVÝ ROBOT

Pro pochopení použití průmyslového robotu je nejprve potřeba nastínit problematiku z oblasti průmyslové robotiky. Slovo *Robot*, které roku 1920 použil poprvé Karel Čapek ve svém dramatu R.U.R., v technické praxi nahradila fráze *Průmyslový robot*. Ten je dle normy ISO 8373 definován jako:

*„Automaticky řízený, reprogramovatelný, víceúčelový manipulátor, programovatelný ve třech nebo více osách, který může být buď pevný, nebo mobilní pro použití v průmyslových automatizačních aplikacích“.*

Vzhledem ke zmíněné definici Průmyslového robotu je také potřeba objasnit pojem manipulátor. Z historického hlediska byly hlavně ve strojírenství k manipulaci s různými objekty používány právě manipulátory. Ty lze dnes chápat jako zařízení bez řídicího systému (nulová úroveň inteligence), cyklicky vykonávající daný pohyb. Také je potřeba si uvědomit, že slovním spojením *manipulátor robotu*, rozumíme mechanickou část robotu. Manipulátory však společně s průmyslovými roboty patří do jedné skupiny, a v technické literatuře jsou také tak označovány, tedy *PraM – Průmyslové roboty a Manipulátory*.

### 2.1 Klasifikace průmyslových robotů

Průmyslové roboty (a manipulátory) je možno klasifikovat z mnoha hledisek [3]. Koncepčně lze rozlišovat 2 základní skupiny.

- **Jednouúčelové** – Jedná se o zařízení s omezenými možnostmi pohybu, kde úroveň (složitost) řízení odpovídá dané aplikaci. Pověštinou se jedná o podavače, či manipulátory.
- **Univerzální** – Tato skupina se vyznačuje možností přizpůsobení různým technologiím (ať už výrobním, či obráběcím).

Univerzální skupinu průmyslových robotů lze dále členit:

#### **Synchronní manipulátory**

Také nazývány teleoperátory jsou manipulátory ovládané člověkem na dálku. Jejich řízení provádí obsluha či operátor. Mechanické ústrojí má často složitou konstrukci a používají se hlavně v prostředích pro člověka nebezpečných.

## Programovatelné s pevným programem

Tyto roboty vykonávají jednoduché předem naprogramované operace. Tyto operace v podstatě tvoří sekvence činností, které jsou stále stejné a po celou dobu vykonávání operace se nemění. Jedná se o tzv. jednoduché průmyslové roboty a jsou vesměs používány pro přemísťování předmětů, zdvih apod.

## Adaptivní roboty

Jak už název napovídá, jedná se o roboty schopné se přizpůsobovat změně okolí. Tyto roboty přes zabudovanou zpětnou vazbu dokáží reagovat na změnu sledované veličiny takovým způsobem, aby ji přivedly zpět do původního stavu. Jedním z příkladů je vyhodnocení reakčních sil např. při dotyku hlavy robota (chlapadla) s překážkou.

## Kognitivní roboty

Kognitivní roboty jsou autonomní (inteligentní) roboty, které jsou díky použití umělé inteligence schopny samostatně řešit zadané úlohy, stejně jako se přizpůsobovat změně podmínek. Zadáání cílů činnosti robota ovšem stále zůstává na člověku, neboť vždy je na počátku nutno definovat činnosti, které bude robot vykonávat. Charakteristickými rysy kognitivních robotů jsou rozpoznávání a orientace v prostředí, či hlasová komunikace (většinou s operátorem).

## 2.2 Kinematika průmyslových robotů

Kinematika se zabývá sledováním polohy, rychlosti, obecně pohybem. V případě stacionárních robotů je potřeba vymezit pojem *kinematická dvojice*. Jedná se o dva vzájemně spojené členy (např.: ramena robota), které se vůči sobě pohybují. Kinematiku stacionárních robotů lze rozdělit do 3 skupin.

První je **sériová** kinematika. Jedná se o roboty s otevřeným kinematickým řetězcem. Jeho členy (části) jsou tedy spojeny pouze jednou kinematickou dvojicí. Z takto propojených dvojic se pak jednotlivé roboty skládají.

Dalším typem kinematiky je kinematika **paralelní**. Ta se objevuje jak u stacionárních tak i mobilních robotů. Princip spočívá v uzavřeném kinematickém řetězci, tudíž vždy nejméně 2 kinematické dvojice jsou spojeny s platformou (zemí, základnou).

Poslední, **hybridní** kinematika, obsahuje, jak otevřený, tak uzavřený kinematický řetězec. Nejvíce je používána u mobilních robotů.

Robot, použit na pracovišti, využívá sériové kinematiky.

## 2.3 Řízení a Učení pohybu průmyslových robotů

Řízení průmyslových robotů z hlediska způsobu pohybu lze rozdělit na dva druhy:

- **Bodové řízení (Point to Point)** – Jedná se řízení od bodu k bodu (dále jen PTP). Využití tohoto řízení lze najít zejména v aplikacích, jako jsou svařování či obsluha obráběcích strojů. Tedy v aplikacích, kde je nezbytně nutné aby mezi jednotlivými požadovanými body operačního prostoru, do nichž robot zasahuje, nebyla funkční závislost.
- **Dráhové řízení (Continuous path)** – Tento způsob se používá zejména v případech, kdy je robot, resp. jeho pohyb funkčně vázán k technologickému procesu. Například lakování či spojité svařování.

Dále je potřeba definovat pojem učení pohybu robota, zvané také „*play-back*“. [4] Vzhledem k velké variabilitě ekvivalentních překladů je potřeba učení robotů přesně specifikovat:

- **Zprostředkované** – Zde je použit tzv. programovací panel. (Teachpendant, Flexpendant). Díky tlačítkům na něm umístěným lze robota navést do dané pozice (požadovaných bodů v operačním prostoru) a uložit danou pozici, případně lze i nastavit spojení uložených bodů po požadované dráze.
- **Bezprostřední** – Robot resp. jeho hlava, chapadlo je vedeno po požadované dráze. Celý tento cyklus je zapamatován (uložen) a následně pouze cyklicky spouštěn.

## 2.4 Souřadný systém robotů se sériovou kinematikou

Roboty obecně pracují s vícero souřadnými systémy. Jsou tedy potřeba transformace mezi zmíněnými systémy. Většinou se jedná kartézský souřadný systém světa – robota, samotného. Následně souřadný systém vztahující se k aktuálnímu natočení polohy hlavy nástroje a v neposlední řadě také souřadný systém vztahující se k začátku trajektorie.

## 2.5 Robotické zakládání

Zakládáním se rozumí přemísťování materiálu mezi několika místy, tzv. „pick & place“ manipulace. V praxi často robot (pomocí chapadla, úchopového mechanismu) manipuluje s předměty stejné velikosti a překládá je mezi předem definovanými místy a také definovaným pohybem. Jedná se o robota se sériovou kinematikou. Dále důležitým faktorem, charakterizujícím tuto práci, a který je potřeba zmínit, je fakt že v tomto případě je potřeba obrobek založit např. pod různým úhlem, přičemž se často také mění

způsob uchycení (z boku, se shora). Je zde určitá variabilita, kterou je nutno brát v potaz. Obráběcí stroj slouží totiž jako testovací zařízení pro různé obrobky.

V tomto případě je obrobek zakládán buď do zásobníku, měřicí stanice nebo obráběcího stroje, dále nazvány pracovní stanice.

### 3 KONFIGURACE PRACOVIŠTĚ

Pro zakládání obrobku do obráběcího stroje je potřeba popsat pracoviště a jednotlivá stanoviště. Kromě obráběcího stroje je na pracovišti také zásobník pro jednotlivé obrobky a měřicí stanice. Přemísťování s obrobku je prováděno robotem firmy ABB.

#### 3.1 Obráběcí centrum

Na pracovišti je umístěno vertikální obráběcí centrum MCFV 1680 firmy TAJMAC-ZPS [15] (dále jen OC) zajišťující třískové obrábění v jednotlivých osách. Uvnitř obráběcího centra je na pracovním stole v podélném směru umístěn svěrák pro uchycení obrobku. Svěrák je ovládán servomotorem, který je připojen k servozesilovači a ten je ovládán APC.



**Obr. 3.1:** Vertikální obráběcí centrum MCFV 1680 firmy TAJMAC-ZPS [15]

### 3.2 Měřicí stanice

Pro ověření správného opracování obrobku je na pracovišti umístěn univerzální kontrolní systém Equator 300 EH firmy Renishaw [16]. Ve stanici je umístěna skenovací tříosá analogová sonda SP25. Robot vždy založí referenční obrobek. Ten je ve stanici oskenován a dle něj jsou následně měřeny veškeré ostatní obrobky, které jsou do měřicí stanice vloženy. Jedná se o měřicí programy ve stanici již nahrané, z APC jsou tyto programy pouze spouštěny.



**Obr. 3.2:** Měřicí stanice Renishaw Equator 300 se sondou SP 225 [15]

### 3.3 Robot

Firma ABB na trhu nabízí nespočet průmyslových robotů pro různé aplikace. Pro účel zakládání obrobku do obráběcího stroje a s ohledem na technologické požadavky byl instalován víceúčelový robot IRB 1600 [12]. Robot splňuje požadavky na něj kladené, neboť materiál, jež jím bude přenášen, nebude těžší než 10 kg.



Obr. 3.3: Náhled stacionárního robota ABB IRB 1600 [12]

### 3.4 Zásobník

Součástí pracoviště je také zásobník, do kterého budou jednotlivé obrobky zakládány. Tento zásobník byl vyroben dle požadavků zákazníka. Je rozdělen na tři části, tedy na část pro hotové obrobky, část pro neopracovaný materiál a část pro průběžné odkládání obrobků během procesu obrábění.

## 4 KOMUNIKACE

### 4.1 Hardwarové prostředky

Pro zmíněnou technologii zakládání je oproti Comau robotu (zde stačí jedna řídicí jednotka) je při použití robota firmy ABB potřeba dvou řídicích jednotek. Jedna pro řízení pohybu robota (přepočet transformací, atd.) a druhá pro řízení samotné technologie, resp. aplikace. V této podkapitole jsou popsána zařízení, mezi nimiž probíhá komunikace a která se zprostředkovávají řízení zakládání. Jedná se o průmyslové PC firmy B&R a řídicí jednotku robota firmy ABB.

#### 4.1.1 Průmyslové PC

Firma B&R na trhu nabízí PLC s různým rozsahem výkonu. Pro náročnější aplikace v průmyslu jsou použity průmyslové počítače, dle názvosloví B&R: „Automation PC“ (dále jen APC). Pro již zmíněnou aplikaci se dle poměru cena/výkon jeví jako nejvhodnější APC 910 a to z důvodu možného nárůstu požadovaného výkonu (složitost aplikace). Škálovatelnost výkonu jednotlivých modelů usnadňuje výběr vhodného HW, neboť zmíněný model 910 je nabízen v mnoha různých variantách. Všechna CPU jsou na platformě Intel Core i procesorů 6. generace. S ohledem na přepokládané vytížení procesoru byla zvolena platforma Intel Core i5 3610ME 2.7 GHz - Dual core s L3MB cache s 8 GB DDR3 RAM pamětí.



**Obr. 4.1:** Náhled průmyslového PC APC910 se dvěma sloty [11]

Ve zvolené konfiguraci má APC 910 port pro sériovou komunikaci (RS232C), dva ethernetové porty (1000BASE-T) a dva volné sloty pro karty umožňující komunikaci přes sériovou linku (RS232/422/485), CAN sběrnici či POWERLINK. Případně lze

využit X20 sběrnici pro připojení Fieldbus kontroléru (CAN, PROFINET, DeviceNet, PROFIBUS). Popis jednotlivých slotů lze nalézt v [1].

## 4.1.2 Řídicí jednotka robota

Součástí každého robota je i jeho řídicí jednotka. Pověštinou také s ručním panelem pro jeho ovládání. Firma ABB ke svým robotům poskytuje řídicí jednotky IRC5. Jedná se již o pátou generaci těchto řídicích jednotek s mnoha funkcionalitami[13]. V tomto případě se jedná o variantu rozvaděčové skříně s tlačítky na přední straně skříně. Jsou zde tlačítka např. pro zapnutí robota a pro nouzové zastavení, jak lze vidět na **Obr. 3.2**. K jednotce je také připojen ruční ovládací panel FlexPendant s displejem a tlačítky umožňující manuální ovládání robota.



**Obr. 4.2:** Náhled rozvaděčové skříně řídicí jednotky ABB robota IRC5 [13]

Komunikace s IRC5 jednotkou je možná po sériové lince (RS 232C/RS 422), pomocí ethernetu (100BASE-TX) a také přes Fieldbus sběrnice (DeviceNet, PROFINET, PROFIBUS DP, Ethernet/IP). Uvnitř již zmíněné IRC5 je integrováno ABB PLC AC500 [14] zajišťující např.: přepočítání kinematiky pohybu robota a celkový běh systému robota.

### 4.1.3 Porovnání komunikačních modulů

Obě zmíněné řídicí jednotky mají možnost komunikace s jinými zařízeními po sériové lince, ethernetu či Fieldbus sběrnicích. V **Tab. 4.1** jsou uvedeny veškeré komunikační možnosti obou řídicích jednotek, tedy APC a IRC5.

**Tab. 4.1:** Porovnání podporovaných možností komunikace pro obě řídicí jednotky (APC, IRC5)[12][14]

Komunikační médium, sběrnice		APC	IRC5
Sériová linka	RS 232C	✓	✓
	RS 422	✓	✓
	RS 485	✓	✗
Ethernet	100 Mbit/s	✓	✓
	1000 Mbit/s	✓	✗
Fieldbus	Ethernet POWERLINK	✓	✗
	CAN	✓	✗
	DeviceNet	✓	✓
	Profinet	✓	✓
	PROFIBUS DP	✓	✓
	Modbus	✓	✗

Jak lze vyčíst z **Tab. 4.1**, pro komunikaci není možno využít B&R hojně využívaný Ethernet POWERLINK, neboť ABB nemá na své straně podporu této průmyslové sběrnice. Odpadá také možnost komunikace skrze Modbus, který není IRC5 podporován. Výběr a podrobnější porovnání zmíněných sběrnic je uvedeno dále.

## 4.2 Výběr způsobu komunikace

Výsledný způsob komunikace (výběr sběrnice) vychází z faktů uvedených v předchozí kapitole. Komunikace po sériové lince je sice možná, ale vzhledem k jejímu stáří a také s ohledem na skutečnost, že je dnes na trhu mnoho průmyslových sběrnic převyšující možnosti sériové linky (rychlost, maximální délka vodičů), není způsob komunikace skrz tuto sběrnici relevantní.

Jako vhodný způsob komunikace na základě **Tab. 4.1** je možno zvolit mezi komunikací po Ethernetu, či Fieldbus sběrnicích, v tomto případě DeviceNet sběrnici, Profinetu, či PROFIBUSU. Po vyloučení možností použít pro komunikaci obou řídicích jednotek sériovou linku, zbývají dle **Tab. 4.1** 4 možnosti. Dále jsou tedy popsány zbývající sběrnice a komunikační protokoly. Jsou zde uvedeny výhody a nevýhody každé zmíněné komunikační možnosti.

### 4.2.1 DeviceNet

Jedná se o standard vytvořený americkou firmou Allen Bradley (dnes součástí Rockwell Automation). DeviceNet sběrnice vznikla definicí aplikačního protokolu na již existující sběrnici CAN, tedy přidání aplikační vrstvy ISO/OSI modelu k fyzické a linkové vrstvě definované CAN protokolem. Výhodou DeviceNet sítě je možnost připojení až 64 účastníků. Přenosové rychlosti se pohybují v rámci 125/250/500 kb/s, přičemž maximální hodnota může být dosažena pouze při délce kabelu do 100 m. Při použití plochého kabelu až do 380m. [6]

### 4.2.2 PROFIBUS DP

Profibus (Process fieldbus) sběrnice, vyvinutá s podporou několika německých firem, implementuje obdobně jako DeviceNet pouze 1., 2., 7. vrstvu ISO/OSI modelu. Fyzickým médiem je v tomto případě stíněná kroucená dvoulinka mající rozhraní RS 485 s nestandardně definovanými parametry a maximální délkou do 1200 m (bez opakovačů). Druhá, linková vrstva definuje přístup k síti jako hybridní verzi IEEE 802.4 Token Bus, což je vlastně použití deterministické přístupové metody na kruhové topologii.

Profibus DP je nejjednodušší a nejrozšířenější variantou Profibusu. Je určen především pro rychlou komunikaci typu Master-Slave. Při použití světlovodiče je schopen dosáhnout přenosové rychlosti až 12 Mb/s. V praxi se Profibus DP využívá především pro ovládání jednoduchých pohonů a dalších akčních členů. Pro komunikaci na vyšší úrovni s velkou nabídkou služeb pro práci s daty či alarmy je určena varianta Profibus FMS.

### 4.2.3 Ethernet/IP

V průmyslu se také čím dál častěji používá ethernet. Je to dáno hlavně dnes již možným zajištěním determinismu, vysokou přenosovou rychlostí a také hardwarem. Např. stíněná vedení, konektory splňující IP67 atd. Jednou z variant průmyslového ethernetu je „Ethernet/IP (Industrial Protocol), vytvořen pod záštitou organizace ODVA (Rockwell), který využívá v 7. vrstvě ISO/OSI modelu CIP protokol původně vytvořený pro DeviceNet sběrnici. Zmíněný protokol využívá komunikaci typu „*producer-consumer*“. Dále Ethernet/IP definuje použití explicitních a implicitních zpráv využívajících protokolů UDP a TCP [6]. Velkou výhodou tohoto protokolu je možnost kompatibility s Ethernet/TCP/IP sítí. Další výhodou je také rychlost všech variant průmyslového ethernetu zejména pak rychlosti 1 Gbit/s a vyšších.

### 4.2.4 Profinet

V případě Profinetu se také jedná o jednu z variant průmyslového ethernetu vyvinutého organizací PNO pod záštitou firmy SIEMENS. Cílem vývoje Profinetu bylo

sloučit výhody Ethernetu a PROFIBUS sběrnice. Ve výsledku tedy probíhá komunikace ve 2 módech. Prvním módem je Profinet I/O. Ten je využíván hlavně pro komunikaci a obsluhu I/O jednotek. Zde je potřeba zmínit třetí verze Profinetu zvanou IRT (Isochronous Real Time). Ta je určena především pro úlohy, u kterých jsou definovány přesné požadavky na synchronizaci a dodržení doby odezvy druhé strany.

Naopak zprávy bez požadavků na přenos v reálném čase jsou zasílány pomocí TCP/IP (UDP) protokolu. Jedná se o druhý mód označovaný jako Profinet CBA (Component Based Automation).

#### **4.2.5 REST**

K již zmíněnému ethernetu je potřeba se vrátit, protože vzhledem k požadavkům průmyslu 4.0, kdy je potřeba posílat velká data skrze velké sítě, je dnes mnoho výrobních zařízení připojených do ethernetových sítí. A pro vzdálený přístup k datům lze použít i standardizované rozhraní, konkrétně REST. Tento koncept distribuované architektury, navržený Royem Fieldingem v jeho dizertační práci[8], využívá 4 základní http služby. Zde je však potřeba zmínit možnou variabilitu definice samotné architektury, která může být také definována pomocí XML a přitom se plně neřídí REST principy. Proto veškeré systémy využívající plně principy Fieldingovy architektury jsou označovány jako RESTful. Na základě zmíněného vyplývá, že přístup k datům je možný také pomocí webového prohlížeče. Postačujícím identifikátorem je URL adresa, která odkazuje na daný objekt – data. Tímto je splněna jedna z podmínek definující průmysl 4.0, kdy je definován vzdálený přístup k datům, resp. stavu výrobního stroje, či jiného manipulačního zařízení.

### **4.3 Výsledná použitá sběrnice**

V přechodných podkapitolách byly představeny jednotlivé sběrnice umožňující komunikaci mezi APC a IRC5. O první zmíněné sběrnici, DeviceNetu, lze dnes také říci, že je považována za „mrtvou“. Totiž v porovnání s dalším zmíněnými sběrnici má DeviceNet velice nízkou rychlost přenosu dat. To je hlavní důvod, proč není tato sběrnice pro zmíněnou aplikaci použita.

Další v pořadí je Profibus DP. Tu je dnes, stejně jako předchozí možnost, považovat za zastaralou. Jedním z důvodů, proč tuto sběrnici nepoužít je fakt, že pro komunikaci zmíněných řídicích jednotek by spíše vyhovovala varianta Profibus FMS, která však není ani jednou řídicí jednotkou podporována. Hlavním důvodem, podobně jako u DeviceNetu je však přenosová rychlost. Profibus DP s maximálním přenosem 12MB/s má s 16 připojenými zařízeními dobu cyklu 2 ms. Oproti tomu Ethernet o přenosové rychlosti 100Mb/s má dobu cyklu pod 1 ms. S tím souvisí 2x vyšší produktivita Ethernetu.

Zbývá tedy průmyslový ethernet a jeho 2 varianty. Jako nejlepší možné řešení z hlediska rychlosti je použít Profinet IRT, který má pro 1 Gbit/s cyklus sítě 31μs a obecně je rychlejší než Ethernet/IP. Zde je však problém s IRC5 řídicí jednotkou, která

dle [12] nemá podporu Gigabitového ethernetu. Na rychlosti 100 Mb/s je tedy cyklus sítě zmíněného Profinetu 250  $\mu$ s. Zajištění determinismu v čase menším než 1ms a operačním cyklu řídicích jednotek v rozsahu 10-100 ms splňuje také Ethernet/IP. Dalším poznatkem je také původní řešení a to rychlost komunikace APC s Comau robotem. Ta probíhá jako half duplex přes Ethernet POWERLINK o rychlosti 100Mbit/s.

Shrnutím všeho řečeného je jasnou volbou Profinet. Je rychlejší než Ethernet/IP a vzhledem k definovaným požadavkům aplikace je determinismus postačující.

## **Simulace**

Je potřeba také zmínit fakt, že pro testování digitálního dvojčete je nezbytně nutná simulace celého technologického procesu. Jedná se tedy o použití simulace za pomoci vývojového prostředí ABB RobotStudio (dále jen RS) [19] a B&R Automation Studia (dále jen AS). V RS je spuštěn virtuální kontrolér robota, stejně tak v AS je za pomoci ARSim spuštěno virtuální APC. Komunikace mezi těmito vývojovými prostředími (inženýrskými platformami) je možná pouze pomocí TCP/IP komunikace realizovanou programově v obou prostředích. Na B&R straně tedy díky knihovně AsTCP a na ABB straně taktéž za použití knihoven. . Pro použití simulace je tedy nezbytné komunikovat mezi zařízeními pomocí TCP/IP.

Výsledkem je tedy možnost použití Profinetu v případě komunikace s HW a TCP/IP, je-li použita simulace. Tento fakt je však v rozporu s hlavní myšlenkou zmíněného konceptu, kde je požadována modularita. Totiž aby bylo možno snadně a jednoduše přecházet mezi simulací a reálným HW. V konečném důsledku je tedy TCP/IP komunikace hlavní možnou variantou. Podrobnější rozbor je uveden v **podkapitole 6.3**.

## 5 NÁVRH KONCEPCE

Hlavní myšlenkou návrhu nové koncepce je modularita resp. nezávislost daného technologického procesu na způsobu řízení a komunikace s robotem. Způsobem řízení je myšleno, jaká data jsou v APC zpracovávána a jaká data jsou robotu (IRC5) posílána.

### 5.1 Dosavadní stav

Stávající koncepce programu je koncipována konkrétně na již zmíněné pracoviště. Pro řízení pohybu robota je použita B&R *mapp komponenta* "OpenRobotics". Ta se stará mimo jiné také o přepočítání kinematiky robota. Comau robot je pomocí Ethernet Powerlink připojen přímo k APC. Stejně tak je pomocí ethernetu (zde TCP/IP) připojena k APC také měřicí stanice a obráběcí centrum.

Z pohledu návrhu SW nelze o stávající architekturu celé aplikace (programu) hovořit o modularitě. Program běžící v CPU APC je koncipován takovým způsobem, že kupříkladu zakládání do zásobníku a měřicí stanice je provedeno naprosto rozdílným způsobem. Stávající aplikace je sice plně funkční, je ale řekněme „ušita“ na míru aktuálnímu pracovišti s přesně danou konfigurací, resp. umístěním zařízení zmíněných v **kap. 3** na pracovišti.

### 5.2 Cíle práce

Cílem je vytvořit univerzální modul pro řízení jakéhokoliv ABB robota, včetně aplikace zakládání do stroje. Díky možnosti použití tohoto modulu jako součásti B&R integrační platformy bude ve výsledku dosaženo možné modularity resp. nezávislosti řídicí jednotky robota na způsobu provedení aplikace. IRC5 bude pouze získávat data, konkrétně jakým směrem a jak se má robot pohybovat. Aplikace technologického procesu bude realizována na APC.

### 5.3 Možnosti pohybu robota

Robota je možno řídit, jak v automatickém, tak manuálním režimu. Manuálním režimem je myšlen tzv. *jogging*, kdy se robot pohybuje pouze v dané ose, či úhlu (TCP) nebo je v pohybu pouze daný kloub robota (JOINT).

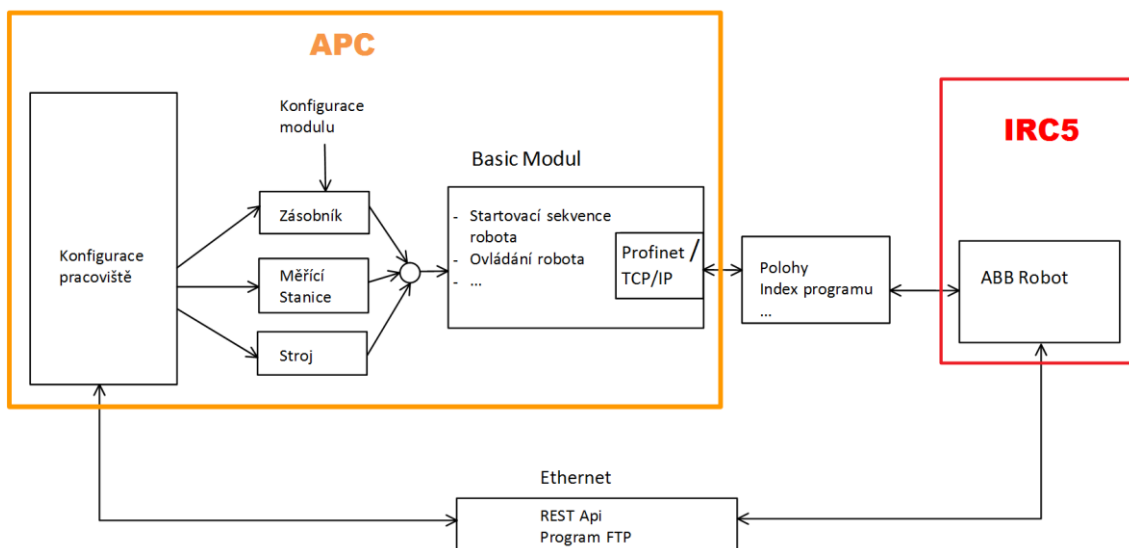
V automatickém režimu se nabízí mnoho dalších způsobů pohybu robota. Jedním z nich je i tzv. *handling* s předem nadefinovanou trasou. Jedná se o RAPID [55] program, který je do robota nahrán a následně spouštěn z APC. *Cyklické zasílání pozic* je dalším typem pohybu robota. V tomto případě je robotu cyklicky zasílána cílová pozice, která je neustále přepisována. Jedná se tedy o jakousi formu *jogging* pohybu, s tím rozdílem, že jednotlivé pozice jsou velmi blízko sobě a přechod mezi nimi je hladký.

Je potřeba se také zmínit ne o možnosti, ale způsobu pohybu, který může být aplikován při všech zmíněných možnostech, a to o *force control*. Jedná se o pohyb se zpětnou tenzometrickou vazbou. Prakticky to znamená, že pokud robot narazí na odpor (rukou zatlačím na hlavu), robot uhne. Snaží se snížit hodnotu tlaku působící na hlavu, nástroj.

## 5.4 Struktura programu

Základní koncepce platformy (její teoretický návrh) se skládá z 3 částí. První a základní je samotný robot se svou řídicí jednotkou zajišťující překlad obrobku mezi 3 místy (stanicemi): obráběcím strojem, měřicí stanicí a zásobníkem. Druhou částí je tzv. *Basic modul* (viz dále) běžící na straně APC, zajišťující komunikaci s robotem (IRC5). Třetí částí je konfigurace pracoviště včetně všech konfigurací jednotlivých modulů (dalších zařízení na pracovišti). Jedná se již o HMI, požadavky zadané obsluhou pracoviště, či operátorem jsou zde zpracovávány. Basic modulu jsou následně předána data ze všech 3 stanic a ten s nimi pracuje. Obě, jak druhá, tak i třetí část jsou prováděny na APC.

Na straně APC je provedeno řízení celého technologického procesu, IRC5 má na starosti pouze komunikaci s APC a zajištění pohybu robota. Na robota je tedy potřeba pohlížet jako na mechanický nástroj s vlastní logikou přepočtu transformací pro pohyb a zpracováním příkazů. Vše ostatní je potřeba řešit na straně APC, tak jak je to uvedeno na **Obr. 5.1**.



**Obr. 5.1:** Nová koncepce platformy

## 5.5 Programové vybavení robota

Dále definovaný „*Basic modul*“ je vytvořen také na straně robota. Robot, resp. IRC5 jednotka je programována ve vývojovém prostředí RobotStudio [18], které je také produktem firmy ABB. Programové vybavení robota odpovídá Basic modulu, ale tato

část (resp. naprogramování robota) byla prováděna firmou ABB v na jaře a dále bude prováděna v průběhu roku 2019.

Pozn.: V případě příkazů robot vždy reaguje na hranu.

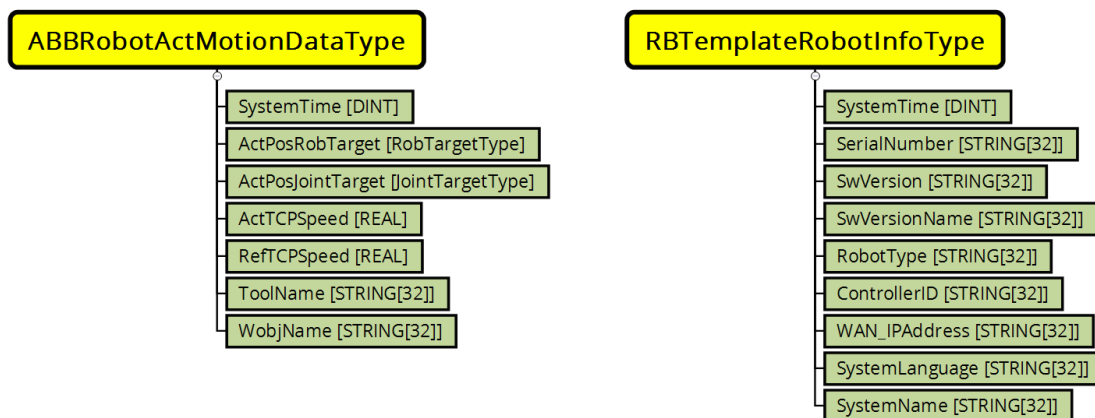
## 5.6 Basic Modul a jeho teoretický návrh

Samotný Basic modul (BM) slouží jako rozhraní mezi APC a IRC5 po zvoleném médiu. Jedná se o přenos dat, ve kterých jsou obsaženy jak systémové příkazy robota (zapnout motory, atd.) tak i status o dosažení požadované polohy, pozice v daných souřadnicových systémech jednotlivých stanic, či status chyby atd. Basic modul je nutno chápat jako „*black box*“ s požadavky dosažení cílové polohy robotem, či způsobu pohybu robota na vstupu a výslednými souřadnicemi na výstupu. Výstupem je také myšlena sekvence příkazů, po které provede robot požadované operace. Dále se Basic modul se stará o zpracování dat uložených v komunikační struktuře (**Obr. 5.2**, **Obr. 5.3**) a jejich vyhodnocení. Data **z robota** jsou vyhodnocena a zasílána do HMI (např.: zobrazení aktuální pozice robota na panelu ve vizualizaci). A naopak data **určená robotu** jsou zpracována a uložena do komunikační struktury, která je cyklicky zasílána z APC na IRC5 a naopak. Celou koncepci BM zajišťuje několik funkčních bloků (dále jen FBK) vytvořených v rámci této práce, viz dále.

Samotná realizace Basic Modulu je popsána v kapitole 6.

### 5.6.1 Datové struktury komunikace

Zde je uveden popis datových struktur zasílaných mezi APC a IRC5, viz **Obr. 5.2** a **Obr. 5.3** Kompletní seznam proměnných, včetně jejich popisu, je uveden v **příloze 1**.



**Obr. 5.2:** Basic Modul – model komunikační struktury mezi APC a IRC5 (Aktuální data z robota)

Na předcházejícím obrázku jsou uvedena data, která budou po vybraném médiu zasílána. V tomto případě se jedná o 2 struktury dat. Datová struktura *ABBRobotActMotionDataType* (na obrázku vlevo) obsahuje aktuální data o pozici robota jak v TCP, tak JOINT souřadnicích včetně rychlosti a další parametrů. Online je také zasílán název aktuálně používaného nástroje a workObjectu. Vpravo na zmíněném obrázku je pak uvedena *RBTemplateRobotInfoType* struktura obsahující dodatečné informace o robotu, jako sériové číslo atd.



**Obr. 5.3:** Basic Modul – model komunikační struktury mezi APC a IRC5 (Příkazy statusy robota)

Hlavní dvě datové struktury uchovávající data potřebná pro řízení robota se nacházejí na **Obr. 5.3**. Vlevo je vyobrazena *RBTemplateRobotSystemCmdType* datová struktura obsahující veškeré příkazy robotu včetně dodatečných parametrů. Naopak datová struktura *RBTemplateRobotSystemStatusType* obsahuje veškeré statusy zasílány z robota. Jednoduše lze tato data popsat následovně:

- **Příkazy robotu** – Jedná se o příkazy „zapnout motory, spustit program“ atd. Veškeré tyto příkazy jsou v robotu zpracovávány a následně lze jejich zpětnou vazbu v komunikaci pozorovat ve struktuře se statusy robota.

- **Statusy robota** – Ať už se jedná o status typu „robot v pohybu, motory zapnuty“, stejně jako u většiny příkazů se jedná přímo o systémové proměnné robota.

### 5.6.2 Řízení komunikace

Řízením komunikace je myšleno zasilání datových struktur popsaných v předcházející podkapitole. Počítá se s využitím funkčních bloků obsažených v AsTcp knihovně v AS. Předpokladem je cyklické zasilání zmíněných datových struktur, přičemž TCP serverem bude vždy IRC5. APC se bude v roli klienta vždy připojovat na IRC5, nikoliv naopak.

### 5.6.3 Startovací sekvence robota

Startovací sekvenci robota nejlépe vystihuje stavový automat na **Obr. 5.4**. V první řadě je potřeba navázat či ověřit komunikaci mezi oběma řídicími jednotkami. Následně program čeká, dokud není robot v automatu. Pokud je robot v manuálním režimu, zmíněný stavový automat se neuplatní. Pokud je vše v pořádku, chyba nenastala a komunikace probíhá, pak je program přepnut do „*Ready*“ stavu, kde buď čeká na externí (např.: z vizualizace) příkaz pro sepnutí motorů, nebo rovnou přechází do „*Active*“ stavu na základě statusu o zapnutých motorech. Poté je možné již spustit program. Samozřejmě jsou ošetřeny i možné chybové stavy včetně kolize robota, či stisku tlačítka nouzového zastavení na pracovišti. Zároveň je také možno zapínat a vypínat bezpečnou rychlost na robotu.

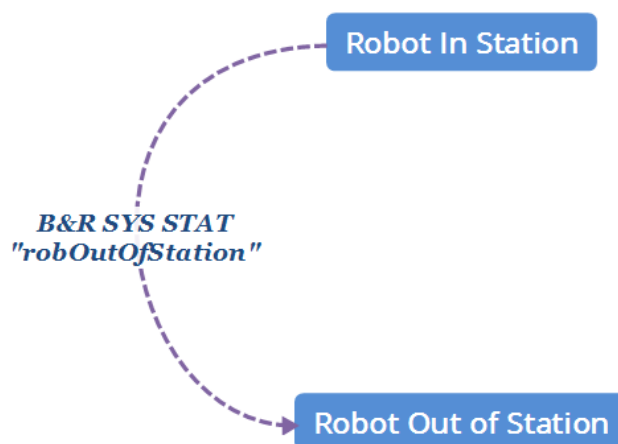
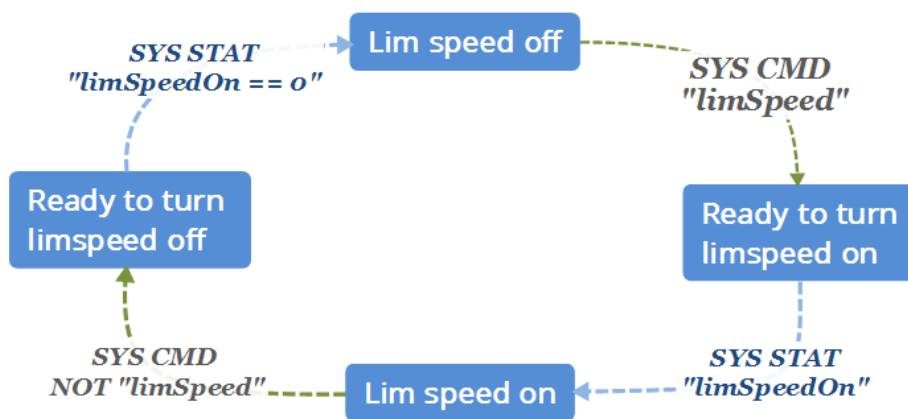
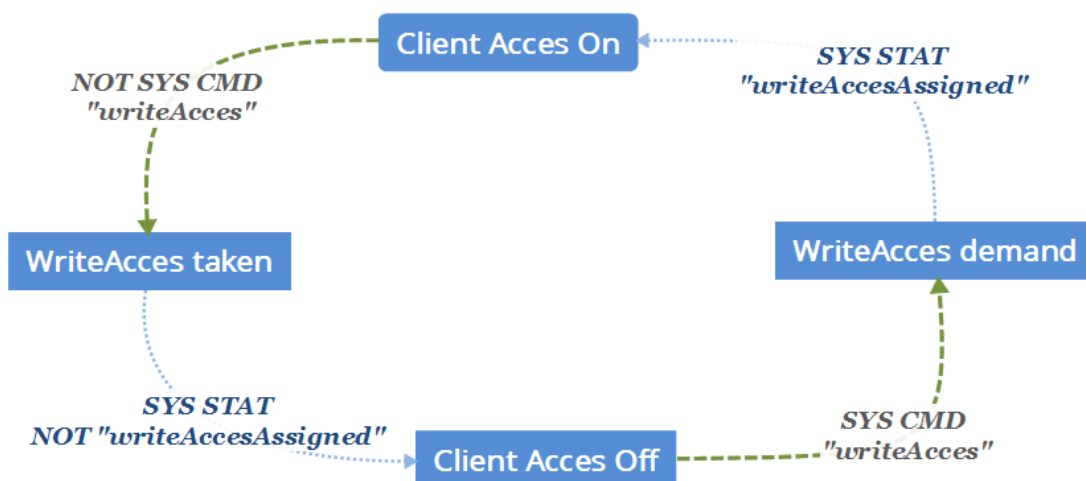
### Vzdálený přístup k robotu

Pokud je robot v automatickém režimu, přístupová (zápisová) práva jsou automaticky přidělena připojenému APC. Přístupovými právy je myšleno, že pokud je například v datové struktuře s příkazy roboty sepnut nějaký příkaz (např.: zapnout motory), je kontrolérem robota ignorován, neboť APC nemá právě tato přístupová práva. Z FlexPendantu u robota je možno tyto zapisovací práva APC „násilně“ odebrat, stejně tak se jich APC může samo vzdát. To znamená, že v programu lze nastavit hodnotu pomocí proměnné v datové struktuře obsahující příkazy robotu, na jejímž základě jsou zápisová práva přidělena.

### Program pro pohyb robota

Zmíněný program pro pohyb robota musí být do robota nějakým způsobem nahrán. Na B&R straně je tedy na základě uložených bodů vytvořen a následně pomocí FTP nahrán (nebo externě, také pomocí FTP) RAPID kód robota obsahující **programový modul** (ten tvoří samotný kód). Zmíněný modul obsahuje, dle ABB názvosloví, určitý počet **procedur**, ve kterých jsou uvedeny **instrukce** pro pohyb robota.

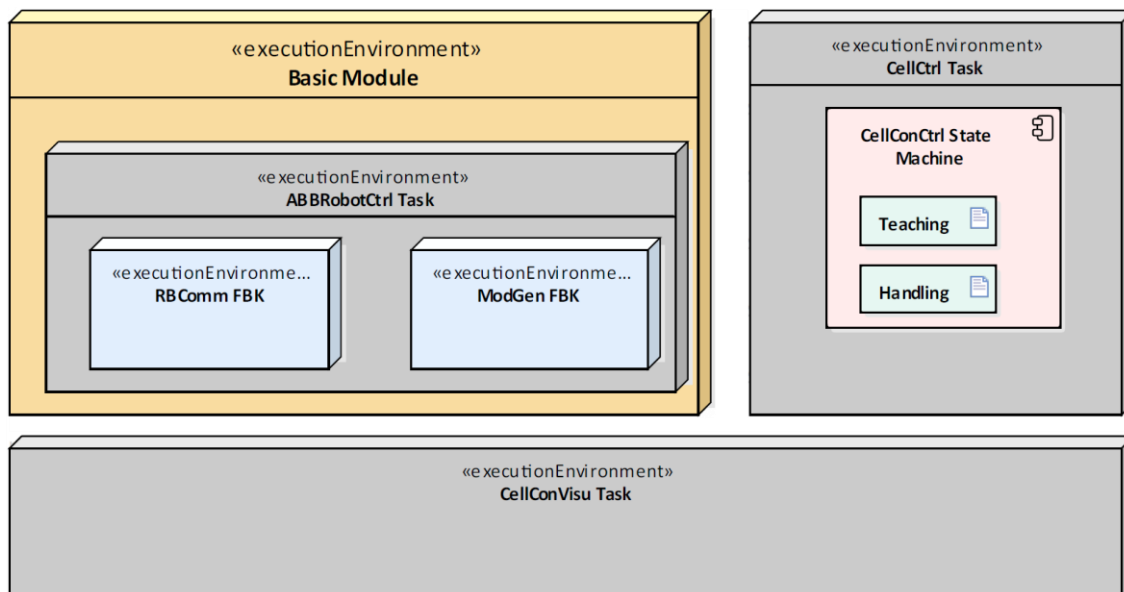




Obr. 5.5: Stavové automaty běžící paralelně mimo startovací sekvenci robota

## 5.7 Hierarchie užití funkčních bloků

Koncepci celého programu lze chápat jako určitou stupňovitou hierarchii užití funkčních bloků. Daný FBK zajišťující určitou funkcionalitu je dále ve větším měřítku použit jiným FBK, který je zase instancí v nadřazeném FBK. Nejvyšší funkční blok se tedy nazývá Basic Modul a ten tedy komunikuje s robotem. Instance jednotlivých funkčních bloků jsou pak následně použity v cyklických „tascích“, kde je již implementováno řízení celého pracoviště. Zmíněnou hierarchii lze vidět na **Obr. 5.6**.



**Obr. 5.6:** Základní koncepce programu

Na **Obr. 5.6** lze také vidět 2 tasky. Jedná se o programy běžící v, dle B&R terminologie, cyklických třídách, tascích, ve kterých jsou vykonávány. Lze nastavit maximální dobu cyklu včetně tolerance překročení doby cyklu atd. [1]. Task vpravo na obrázku – *CellCtrl* zajišťuje chod celého pracoviště – je zde hlavní stavový automat. Také jsou zde zmíněny dva režimy chodu robota popsané v přecházející kapitole. V případě *CellConVisu* se jedná o task již existující ve stávající aplikaci. Zmíněný task má za úkol předávání dat vizualizaci (viz kapitola 6.7).

## 6 REALIZACE

### 6.1 Prvotní návrh algoritmu

Zde je namístě zmínit prvotní návrh algoritmu řízení ABB robota., jež byl realizován při použití Profinetu. Program pracující s proměnnými ve *gABBRobotCtrl* struktuře popsané v **příloze 2**, je v AS přiřazen jedné z osmi taskových tříd cyklicky běžících na CPU APC. Z nutnosti rychlého čtení dat byla zvolena první tasková třída s nejvyšší prioritou, dobou cyklu 2 ms a nulovou tolerancí překročení délky cyklu [1].

Algoritmus zpracování uložených dat je rozdělen na 2 části.

- V inicializační části je při každém startu robot přepnut do manuálního režimu. Tím je ošetřeno možné nežádoucí chování robota způsobené vypnutím napájení v průběhu pohybu. Také je nastaven příznak signalizující, že APC je připraveno posílat data.
- V cyklické části jsou nejprve přečteny všechny vstupy. Následně je kontrolována chyba na robotu a chyba programu. Pokud chyba nenastala a je zde požadavek na automatický režim robota, je robot přepnut do automatického režimu a čeká na další příkazy. Poté již následuje sekvence příkazů a jejich zpětná vazba (opět jsou kontrolovány proměnné v *gABBRobotCtrl* struktuře). Nejprve jsou spuštěny motory a program čeká na sepnutí proměnné pro start pohybu (ať už handling, cyklické zasílání pozic, či program). Nakonec program čeká na konec cyklu (dokončení pohybu a vrácení do *home* pozice), přičemž je ale stále kontrolována proměnná uchováující informaci o chybě na robotu.

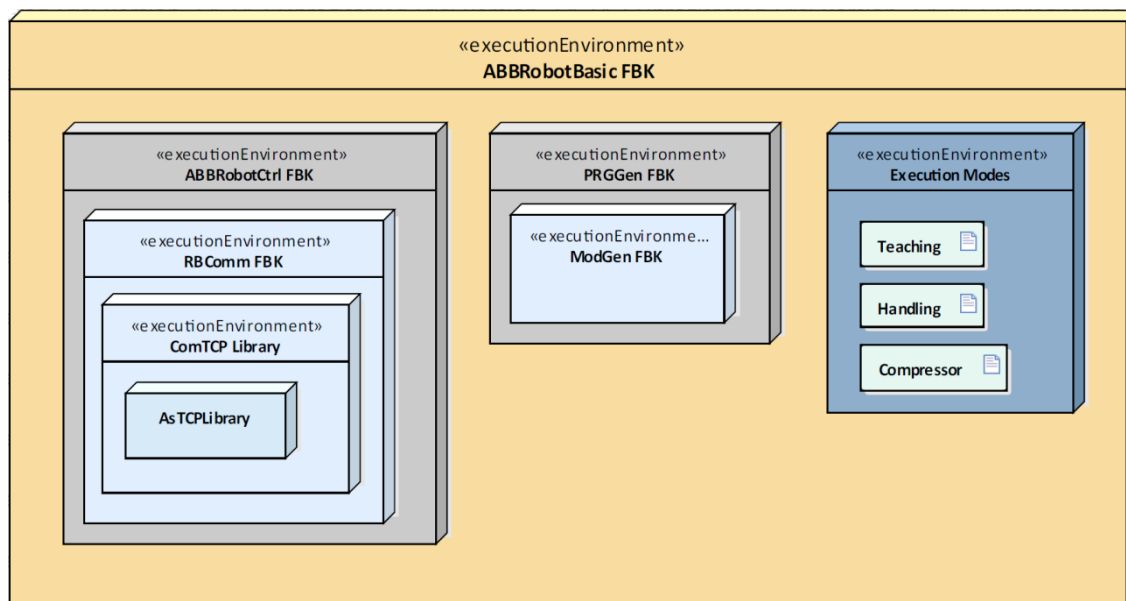
Výše popsany algoritmus ve výsledné koncepci nakonec použit nebyl, přispěl však k dalšímu rozšíření koncepce a hlavně bylo možno díky této koncepci plně pochopit spouštění a „chování“ robota.

### 6.2 Rozdíl realizace oproti návrhu

Samotná realizace zmíněné platformy se ve veliké míře liší od jejího návrhu popsáno v předcházející kapitole. Je to jednak dáno změnou způsobu komunikace, tak také startovací sekvencí robota, kterou musí ABB robot projít, aby mohl být v chodu.

#### Úprava Basic Modulu

Původní koncepci „vnořování funkčních bloků“, bylo potřeba z hlediska přehlednosti a funkcionality upravit. Funkční blok BasicModulu byl přejmenován na **ABBRobotBasic**. Jeho koncepce, resp. samotné zanořování jednotlivých funkčních bloků bylo upraveno takovým způsobem, aby bylo dosaženo přehlednosti a efektivnějšího zpracování dat, viz **Obr. 6.1**.



**Obr. 6.1:** Finální koncepce ABBRobotBasic funkčního bloku

### Význam konfigurace pracoviště (SW)

V **podkapitole 5.4** je uvedena konfigurace pracoviště jako jedna ze 3 základních částí tvořících koncepci programu. Vzhledem k veliké různorodosti jednotlivých možností se jeví jako vhodné omezit zmíněnou aplikaci na jeden jediný přístup. Jedná se o 1 stavový automat řídicí zmíněnou aplikaci. Přičemž je mimo něj cyklicky volána instance **ABBRobotBasic** FBK.

### 6.3 Změna použité sběrnice

V průběhu realizace zmíněné platformy bylo potřeba řešit specifické problémy, na jejichž základě byla provedena náhrada Profinetu za „klasickou“ TCP/IP komunikaci. Jsou to tedy:

- **Potřeba simulovat pracoviště** – Jedním z hlavních aspektů, které bylo potřeba brát v potaz je, jak již bylo zmíněno v úvodu, že celé pracoviště se nachází v testovacím středisku, tudíž robot nebyl vždy k dispozici. Důsledkem, jak se ukázalo, samozřejmě v souladu s průmyslem 4.0, bylo potřeba vytvořit **digitální dvojče** celého pracoviště. To bylo zákazníkem ve spolupráci s ABB a B&R provedeno v prostředí ABB RobotStudio. Díky tomu, že zde běží virtuální kontrolér robota, bylo možno pracovat s robotem i v simulaci. Dále vzhledem k tomu, že komunikaci mezi oběma řídicími jednotkami je realizována programově, bylo použito TCP/IP právě pro komunikaci mezi oběma vývojovými prostředími v simulaci. Je tedy možno mít HW na jedné straně a simulaci na druhé, či naopak.

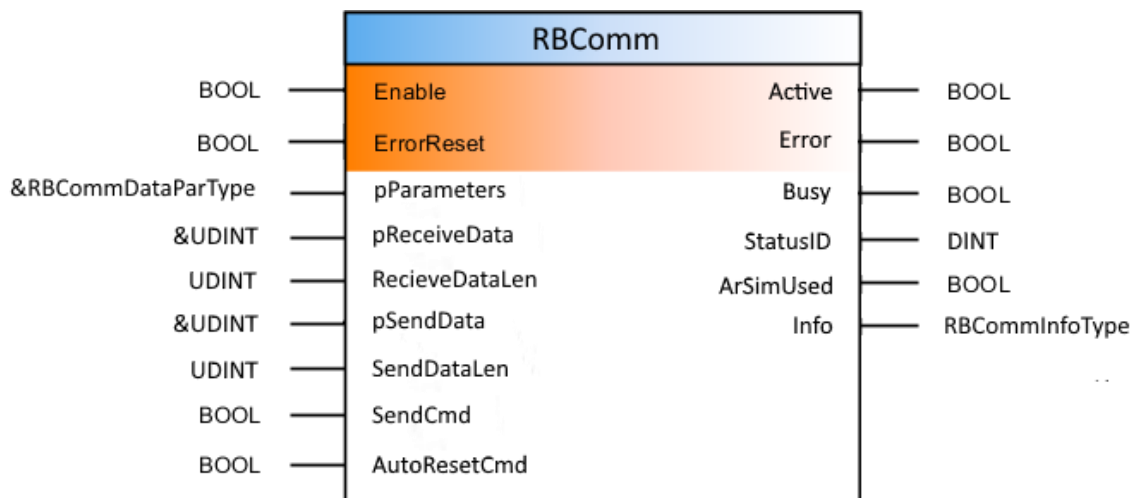
- **Ceny Profinet karet** – Dalším, neméně důležitým aspektem se ukázala být cena Profinet karty na B&R straně. Tento aspekt také přispěl k volbě právě TCP/IP.
- **Možnost využití FTP** – Použití FTP protokolu se v průběhu testování ukázalo velice užitečné. A to hlavně v době, kdy ještě nebyl hotov funkční blok pro vytváření RAPID kódu.

## 6.4 Řízení robota

Dle návrhu, uvedeného v kapitole 5, byly realizovány 4 hlavní funkční bloky realizující řízení ABB robota z B&R strany. Zmíněné funkční bloky realizují jednotlivé operace a v průběhu vývoje byly několikrát upravovány.

### 6.4.1 Řízení komunikace – RBComm FBK

Pro řízení a navázání komunikace byl vytvořen funkční blok (dále jen FBK) řízení komunikace *RBComm*. Tento FBK má na starosti navázání komunikaci a cyklické zasílání dat na uvedených na vstupu. Byly tedy vytvořeny instance pro každou ze 4 zmíněných datových struktur (viz podkapitola 5.6.1) vyjma *RBTemplateRobotInfoType* datové struktury. Ta je zasílána pouze při navázání komunikace nebo na základě požadavku ze strany APC.



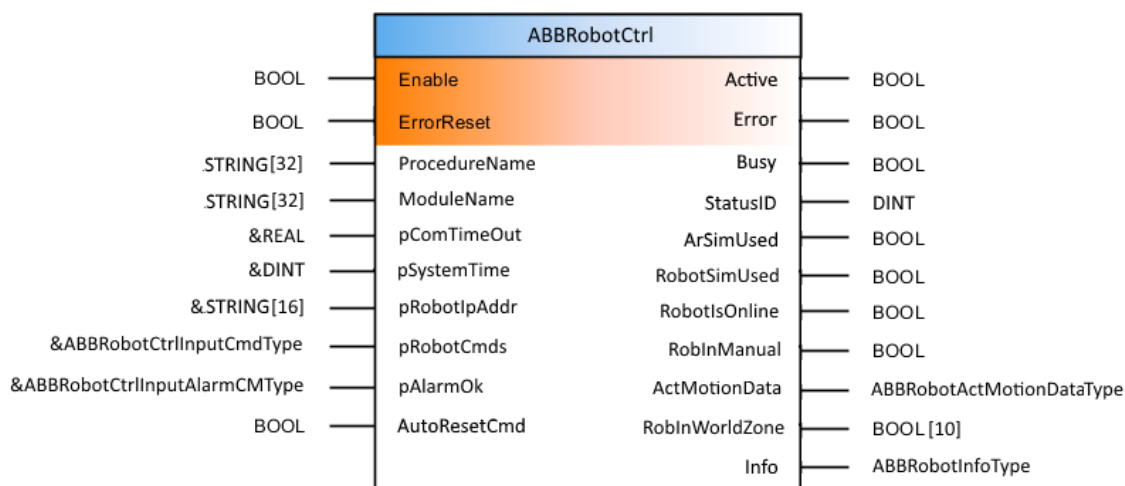
**Obr. 6.2:** Funkční blok řízení komunikace RBComm včetně popisu vstupů/výstupů

Zmíněný funkční blok využívá další funkční bloky již hotové a v rámci B&R vytvořené knihovny COMTcp. V ní je zapouzdřena další knihovna AsTCP zajišťující sestavení spojení. [1].

## 6.4.2 Řízení startovací sekvence robota – ABBRobotCtrl FBK

Řízení startovací sekvence robota realizuje funkční blok ABBRobotCtrl využívající RBComm FBK pro přenos všech čtyř datových struktur. Tento FBK s nezbytnými úpravami implementuje stavový automat uvedený na **Obr. 5.4**. Aby byla zajištěna dostatečná prodleva přepnutí stavu po vykonání daného příkazu, bylo využito v robotu již implementované proměnné *SysInputBusy* v *RBTemplateRobotSystemStatusType* datové struktuře. Po vykonání příkazu je hodnota této proměnné přepnuta kontrolérem robota do stavu logické 1. Jakmile je příkaz kontrolérem akceptován a jím vyvolaná operace dokončena, je hodnota této proměnné nastavena na log. 0. V APC (ABBRobotCtrl FBK) je tedy kontrolována hodnota této proměnné (proto nutnost využití RBComm uvnitř ABBRobotCtrl) a až po jejím přepnutí do log. 0 je resetován daný příkaz. Zmíněnou funkcionalitu zajišťuje další FBK ABBRobotSysWait vytvořen v rámci vývoje ABBRobotCtrl funkčního bloku.

Sekvence příkazů, jež jsou na vstupu tohoto FBK je přesně dána a nelze ji měnit. Pokud by např.: byl vynucen příkaz pro spuštění programu a motory by nebyly nahozeny, je tento příkaz ignorován a na výstupu FBK je uveden „warning“ (odpovídající hodnota na *StatusID* výstupu) o příchodu příkazu v nesprávný čas.

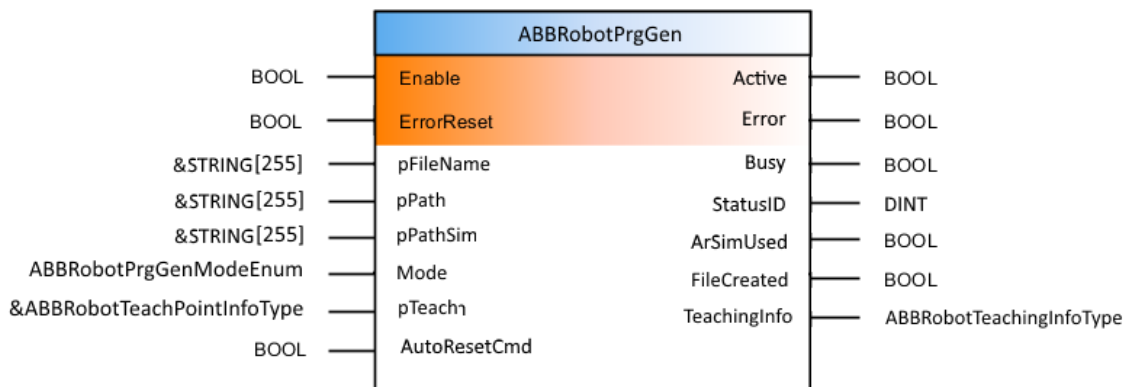


**Obr. 6.3:** Funkční blok řízení startovací sekvence robota ABBRobotCtrl včetně popisu vstupů/výstupů

Na vstupu *pAlarmOk* FBK uvedeného na **Obr. 6.5** je uvedena struktura *ABBRobotCtrlInputAlarmCMTType*. Jedná se o datovou strukturu, ve které jsou proměnné vztahující se k potvrzení alarmů, resp. chybových stavů na robotu. Ty je vždy potřeba potvrdit, aby mohla být daná chyba vy resetována. Děje se tak ve stavu „*System Err Reset State*“ uvedeném ve stavovém automatu na **Obr. 5.4**. Bližší popis resetu chyby robota je uveden v **příloze 3**.

### 6.4.3 Modul generování programového modulu RAPID kódu robota – ABBRobotPrgGen FBK

Hlavním požadavkem na tuto platformu je plné řízení ABB robota APC B&R. To znamená, že i RAPID programy pro robota [20] musí být vytvářeny na B&R straně. Dále v souvislosti s využitím TCP/IP komunikace a FTP protokolu pro „nahrání“ programu robota do jeho řídicí jednotky byl vytvořen další funkční blok zajišťující vytvoření \*.mod souboru (dle ABB názvosloví programového modulu)[20], ve kterém je uveden RAPID program pro pohyb robota, viz podkapitola 5.6.3 – Program pro pohyb robota. ABBRobotPrgGen funkční blok vytváří programový modul robota, konkrétně aktuálně ukládané pozice při učení jsou uloženy do tzv. *MoveInstructionList* datové struktury, která je přesnou kopií systémové datové struktury užívanou ABB při učení robota. Z parametrů a hodnot v této datové struktuře (aktuální pozice, parametry nástroje,...) jsou následně vytvořeny instrukce pro pohyb robota, jež jsou součástí dané procedury v programovém modulu. Je tedy vytvořen celý soubor \*.mod a uložen na FileDevice [1] – většinou na USER partition na CFast kartě.



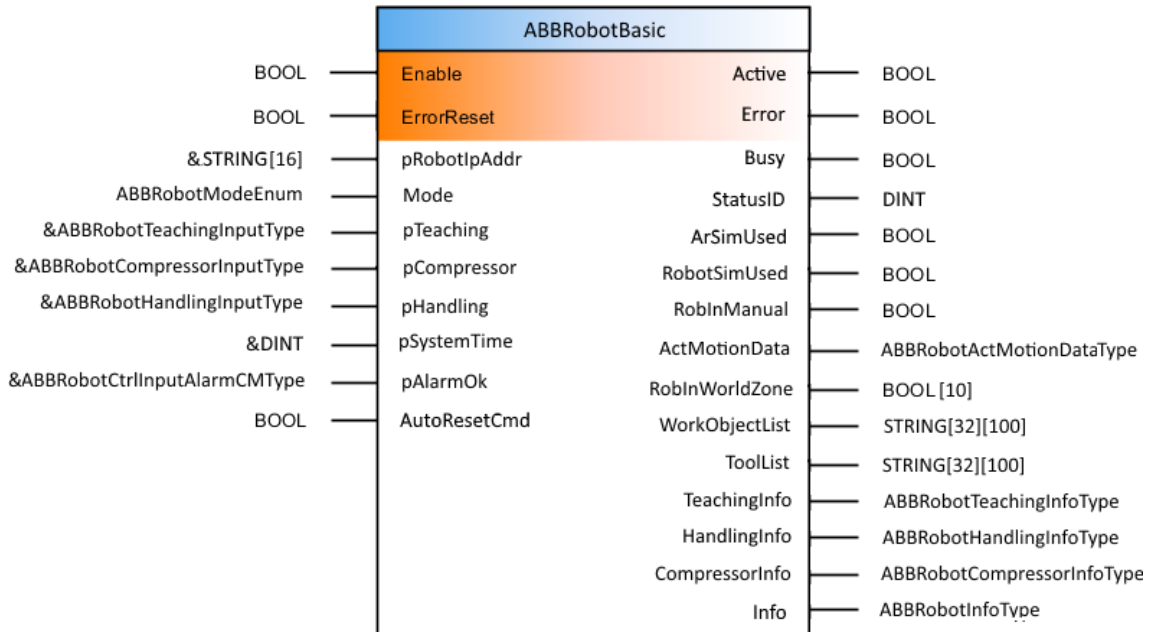
**Obr. 6.4:** Funkční blok pro generování programového modulu s RAPID kódem robota

Aktuální koncepce umožňuje pouze vytvoření jedné procedury v rámci jednoho modulu. Pro aplikaci zakládání je to naprosto postačující. V budoucnu se však předpokládá užití platformy řízení robota i v jiných aplikacích. Je tedy velice pravděpodobné, že stávající kód tohoto FBK bude rozšířen o zmíněnou možnost více použití více procedur v rámci jednoho programového modulu robota. Tomu je již také uzpůsoben stavový automat spuštění programu, jehož náhled je uveden v **příloze 3** na **Obr. P3.1**.

Struktury uvedené na vstupu jsou pospány v kapitole 6.5.

## 6.4.4 Hlavní modul řízení robota – FBK ABBRobotBasic

Tento funkční blok dle **Obr. 6.1** zajišťuje kompletní řízení ABB robota včetně všech možností způsobu pohybu robota (kapitola 5.3).

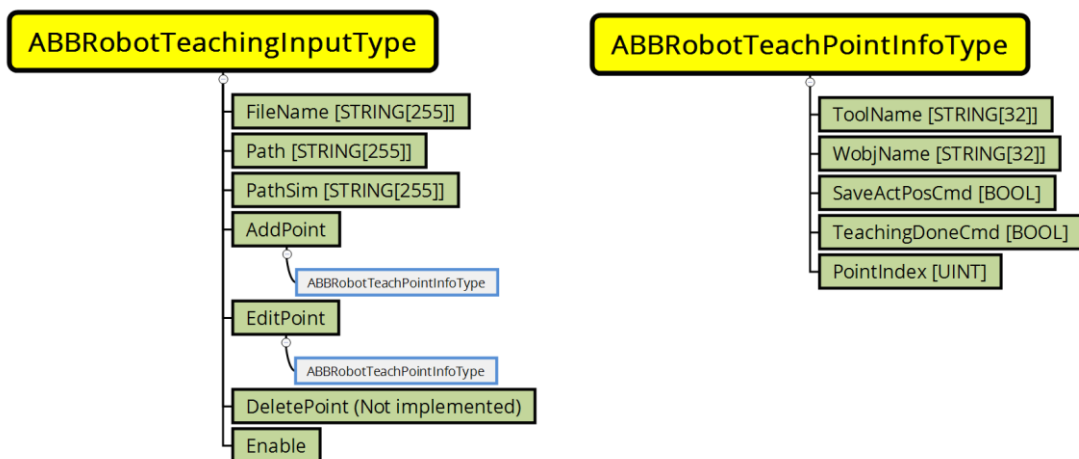


**Obr. 6.5:** Hlavní funkční blok řízení robota ABBRobotBasic včetně popisu vstupů/výstupů

Jednoduše lze popsat jeho hlavní funkcionalitu následovně. Vstupem *Mode* je jasně určeno, v jakém módu bude zmíněný FBK pracovat. Následně jsou zkontrolována vstupní data pro daný mód a program dle proměnných v těchto strukturách (většinou příkazy) vykonává danou sekvenci příkazů. Tento FBK využívá dle **Obr. 6.1** jak ABBRobotCtrl, tak i ABBRobotPrgGen funkční bloky, mezi kterými předává interní data. Instance toho funkčního bloku je pak použita v *CellCtrl* tasku.

## 6.5 Učení pozic

K učení pozic, resp. pohybu robota byl uzpůsoben ABBRobotBasic FBK. Zde je využito ABBRobotPrgGen funkčního bloku, který, jak již bylo popsáno, vytváří \*.mod soubor (dále jen modul programu robota), ve kterém je kód pro pohyb robota. Pomocí FlexPendantu je v manuálním režimu robot, resp. hlava s nástrojem uvedena do polohy. Následně je ve vizualizaci na panelu (viz kapitola 6.7) stisknuto tlačítko pro uložení aktuální pozice. Po ukončení je stiskem dalšího tlačítka ukončeno učení pozic a je vygenerován již zmíněný modul programu robota. Na výstupu Basic funkčního bloku TeachingInfo jsou pak uvedena data typu kolik bodů bylo naučeno, či index aktuálního bodu a status o uložení příslušného bodu. Náhled datové struktury na vstupu ABBRobotBasic FBK je uveden na **Obr. 6.6**.



**Obr. 6.6:** Náhled datové struktury na pTeaching vstupu ABBRobotBasic FBK

Jak lze na výše uvedeném obrázku vidět, mód učení nabízí nejen možnost vytvoření daného bodu (instrukce pohybu na tento bod v proceduře modulu programu robota), ale také jeho úpravu. Je tedy možné vzít aktuální modul programu robota a upravit v něm jednotlivé, již naučené body. Smazání jednotlivých bodů není v této době (tj. květen 2019) doposud implementováno, bude tomu tak učiněno pravděpodobně v průběhu dalších měsíců. Dále je potřeba zmínit nástroj. Robot celkem může pracovat se 4 nástroji [12]. Pro každou sekvenci bodů (proceduru) je možné zvolit nástroj, se kterým bude prováděna. Stejně tak je tomu se souřadnicovými systémy. Tedy, dojde-li například k nahrazení stávajícího zásobníku pro obrobky na pracovišti zásobníkem jiným, s jinými rozměry. Je v průběhu učení vytvořen Workobject nového zásobníku. Poloha koncového bodu robota je pak při zakládání, či odebírání obrobku ze zásobníku vztažena právě k souřadnému systému tohoto zásobníku – workObjectu zásobníku.

## 6.6 Spouštění programu

Programem robota je myšlen již několikrát zmíněn modul programu pohybu robota. Program robota je vždy vytvořen na disku APC. Robot, resp. IRC5 se vždy pouze připojí na FTP server běžící na APC a stáhne si aktuální program. Ten je v robotu zaveden a následně je možno jej spustit pouze pomocí příkazu z APC. K tomuto účelu slouží datová struktura na vstupu ABBRobotBasic FBK *ABBRobotBasicHandlingInputType*. Důležitými parametry jsou v tomto případě název modulu, procedury a příkazy pro jejich spuštění viz **Obr. P3.1** v **příloze 3**.

## 6.7 Vizualizace

Součástí B&R integrační platformy je také uživatelské prostředí HMI. To již bylo v rámci původního projektu vytvořeno a plně funkční. S novým konceptem aplikace, resp. řízení robota však byla potřeba tuto vizualizaci upravit. Celá vizualizace je

zprovozněna na panelu uchyceném na stojanu umístěném na pracovišti. Jedná se o 21” multitouch panel 5AP5230 s FULL HD rozlišením [1]. Jedná se o vizualizaci ve webovém rozhraní B&R mappView. Spuštěnou pomocí webového prohlížeče Windows Chrome. Na APC tedy běží paralelně jak OS (v tomto případě Win10 embedded) tak B&R Automation Runtime. Chod obou systémů zajišťuje B&R Hypervisor [1].

Pro účely aplikace byly tedy vytvořeny stránky umožňující jak spuštění programu v robotu, tak také učení robota. Na **Obr. 6.7** je vyobrazen náhled stránky pro spuštění modulu programu robota. Pro použití v simulaci (a tudíž testování na menším displeji) byla vizualizace z důvodu přehlednosti také upravena.

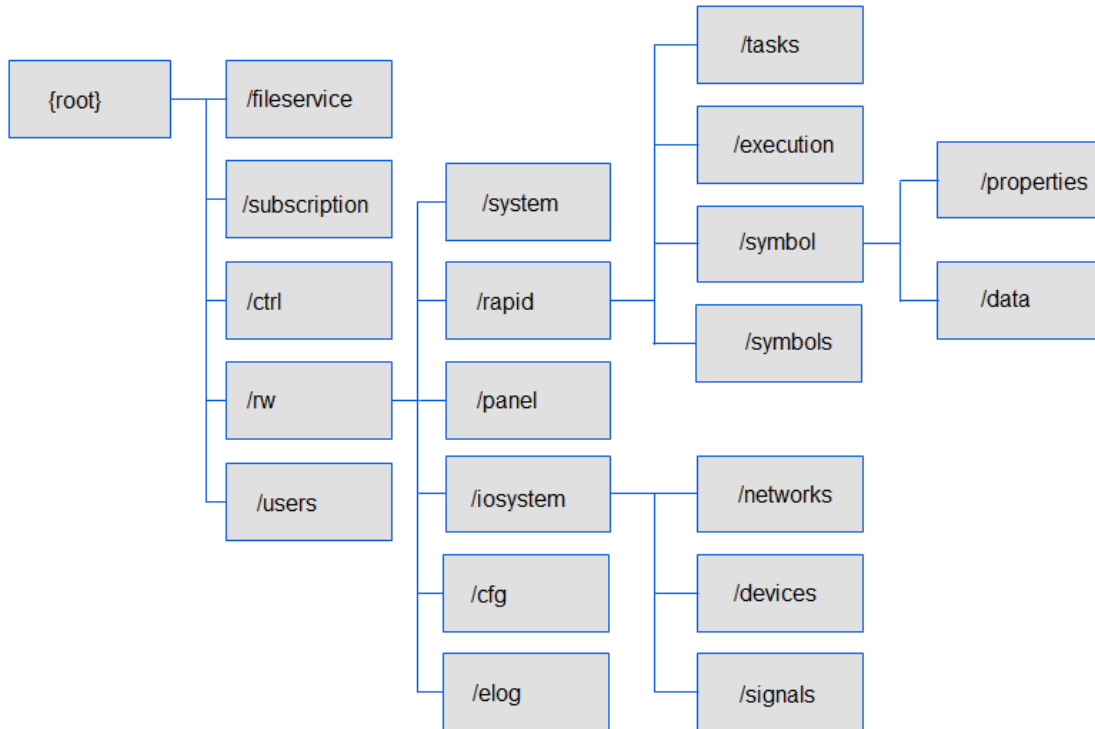


**Obr. 6.7:** Náhled části stránky pro spuštění modulu programu robota

## 6.8 Diagnostika robota

Pro diagnostiku je potřeba zasílat data v určitých časových intervalech. Diagnostikou je v tomto případě myšleno zpracování dat z robota, např.: doba spuštění

motoru klouby, jeho aktuální spotřeba elektrické energie atd. Předzpracování těchto dat je provedeno již na straně robota. Firma ABB poskytuje ke svým robotům službu „Robot Web Services“, běžící na IRC5, která zpracovává veškerá diagnostická data. [17].



**Obr. 6.8:** Datová struktura obsahující veškerá diagnostická data o robotu[17]

Vzhledem k požadavku na cyklické zasilání těchto dat (nejedná se ale o real-time ve smyslu sběrnice) nelze použít vybranou sběrnici a to hlavně z důvodu velkého objemu již zmíněných dat. Vhodnou volbou (vzhledem k řečenému) je využití standardizované komunikace skrze webové rozhraní REST API [7].

Dalším důvodem je také fakt, že tato data nejsou uchovávána v APC, ani s nimi (v omezené míře) není pracováno, jsou pouze zasilána do nadřazených systémů, např.: [17]. A vzhledem k již zmíněnému požadavku, že APC je nadřazeno IRC5 a robotu, je upuštěno od možnosti připojovat se přímo na robota. (byť pro účely testování je to stále možné).

# 7 OVĚŘOVÁNÍ

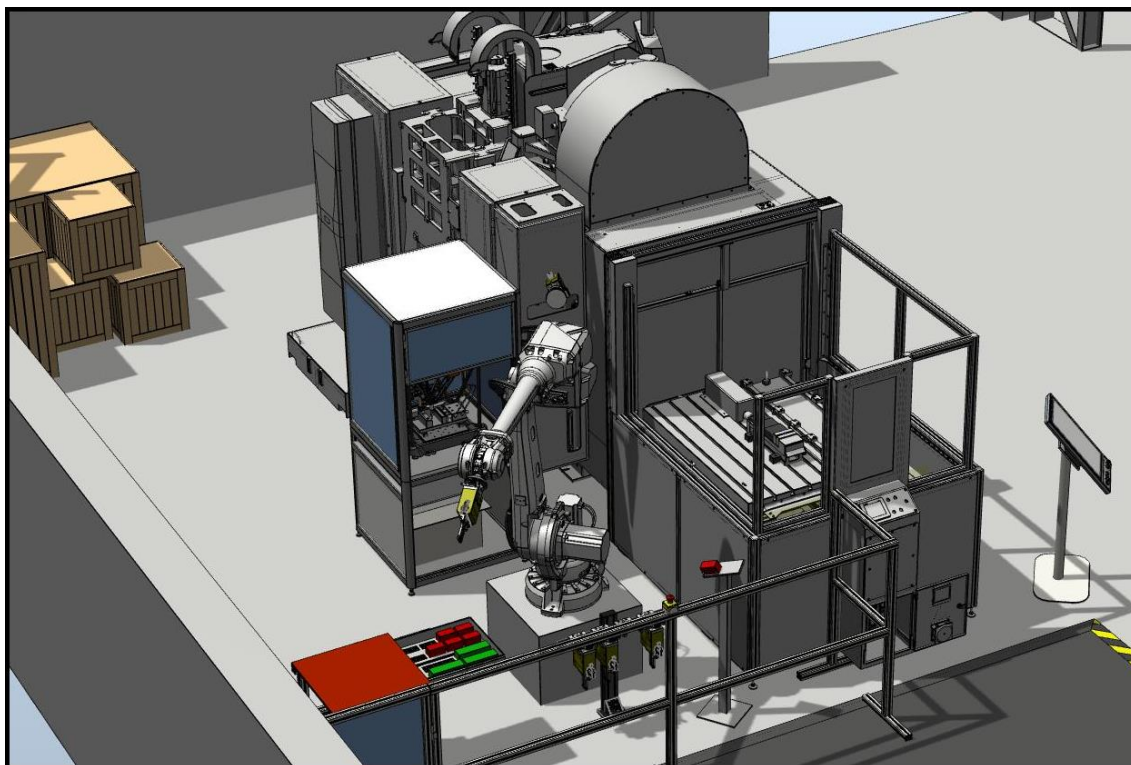
V následující kapitole jsou popsány jednotlivé fáze a části testování a ověřování nové koncepce programu. Byla provedena rešerše a testování komunikace.

## 7.1 Komunikace

Testováním komunikace je myšleno ověření funkčnosti funkčního bloku pro komunikaci s robotem RBComm. Zde při cyklickém zasílání nastal problém se zahlcením komunikace. Jak již bylo řečeno v případě této TCP/IP komunikace je robot vždy TCP serverem, na kterého se APC připojuje. V kontroléru robota byl tedy ABB vytvořen program pro zmíněné cyklické zasílání dat a zpracování příkazů z B&R strany (obdoba BasicModulu na ABB straně). Vzhledem k tomu, že kontrolér robota cyklicky „hrne“ data popsaná v datových strukturách komunikace, nastal problém při hodnotě doby cyklu zasílání těchto dat 1 ms. RBComm byl v první fázi testování implementován takovým způsobem, že před odesláním dat robotu vždy provedl ping druhé strany (knihovna AsICMP [1]). S tím souvisí důsledek zmíněné doby cyklu. Totiž zpětná vazba po příkazu ping od robota ne vždy došla. Totožný problém nastal i v simulaci, kdy i po použití příkazů ping přímo z Windows OS došlo vždy 50% paketů. Byla tedy zvýšena hodnota doby cyklus zasílání dat od robota na hodnotu 2 ms. Zde ale opět nastal totožný problém při použití příkazu ping. Smyslem jeho použití bylo otestování, zda druhá strana (kontrolér robota) je také v chodu. Řešením se ukázalo být použití časových značek v datových strukturách komunikace. A ověření, zdali je druhá strana v chodu, bylo přesunuto do funkčního bloku řízení robota ABBRobotCtrl. Zde jsou vyhodnocovány časové značky a v případě velkého rozdílu je použit příkaz ping pro ověření, zdali komunikace probíhá. Výsledná hodnota doby cyklu zasílání dat jsou tedy 2 ms. Je to dáno také tím, že program pro obsluhu robota ve kterém je volán ABBRobotBasic běží na APC v první taskové třídě (tedy nejvyšší priorita) s dobou cyklu taktéž 2ms a nulovou tolerancí překročení doby cyklu.

## 7.2 Simulace

Simulace byla prováděna v RobotStudios. (kapitola 4.3 – Simulace). Samotné testování SW popsaného v předcházející kapitole bylo však (pokud to bylo možno – viz dále) povětšinou testováno na reálném HW. Jednalo se však o jiný model robota, konkrétně model IRB 1200 [21], případně IRB 4600 [22] a to také v případě simulace. Na Obr. 7.1 je uveden náhled pracoviště v prostředí RS, ovšem s jiným OC, konkrétně MFCV 1060 [23].



**Obr. 7.1:** Náhled budoucího pracoviště (digitální dvojče) v RS

## Komunikace

Vzhledem k tomu, že komunikace je realizována programově na obou stranách, bylo dosaženo totožného způsobu zpracování dat jak pro HW, tak i simulaci. Pouze se tedy změnila IP adresa na vstupu ABBRobotBasic funkčního bloku.

Jedním z problémů testování v simulaci se stalo použití FTP pro přenos modulu programu robota. FTP server na B&R straně není totiž v ARSim (Simulace runtime APC) implementován. Byla tedy vytvořena sdílená složka, do které byl modul programu robota ukládán a odkud si jej virtuální kontrolér robota stahoval a zaváděl.

## 7.3 Řízení robota

Ověřování také spočívalo v otestování funkčnosti ABBRobotBasic funkčního bloku včetně ostatních zmíněných v kapitole 6.4. V první řadě bylo provedeno testování komunikace již popsané v kapitole 7.1. Po otestování funkčnosti RBComm funkčního bloku došlo k testování ABBRobotCtrl funkčního bloku. Zde nastal problém se zpětnou vazbou od robota po zpracování příkazů. Funkční blok ABBRobotSysWait zmíněný v podkapitole 6.4.2. byl původně implementován jako funkce. V průběhu ověřování funkčnosti ABBRobotCtrl FBK bylo zjištěno, že logicky po zpracování různých příkazů robotu trvá odezvy vždy různou dobu. ABBRobotSysWait implementován jako funkce byl tedy po jejím zavolání nefunkční. Bylo potřeba udržet informaci o tom že systémová proměnná robota *SysInputBusy* změnila hodnotu. Řešení ve smyslu čekání v této funkci pomocí nekonečné smyčky bylo ihned zavrženo. APC totiž na tento status

při čeká někdy i 1 s, tudíž by došlo k překročení doby cyklu na APC a to by se restartovalo a přešlo do SERVICE režimu.

Bylo tedy otestováno učení robota pozicím, včetně vygenerování modulu a jeho následné spuštění, včetně řešení chybových stavů, viz dále. Také bylo otestováno zapínání bezpečné rychlosti robota a ověření přístupových práv.

## 7.4 Zakládání do svěráku

Poslední a neméně důležitá část této práce, konkrétně zakládání do obráběcího stroje, resp. do svěráku v tomto stroji nebylo možno prakticky otestovat. Důvodem je (mimo jiné důvody uvedené v podkapitole 6.3) fakt, že robot nebyl ve zmíněném testovacím středisku vždy přítomen (vlivem dohod mezi investorem a dodavatelem). Jedinou alternativou tedy bylo testování robota bez obráběcího stroje. Výsledek této práce byl tedy zejména v posledních fázích (kdy už byl odladěn SW v simulaci) testován v školicím středisku pro obrábění v Praze Vestci. Zde mi byl zapůjčen robot IRB1200, na kterém bylo možno testovat zmíněný koncept, viz



**Obr. 7.2:** Náhled robota IRB1200, na kterém byl koncept testován

## 7.5 Chybové stavy robota

V této kapitole je popsáno řešení chybových stavů robota, které byly otestovány.

### Kolize robota

Řešení kolizí robota bylo zahrnuto již v návrhu pospaném v kapitole 5. V případě, že dojde ke kolizi robota, robot automaticky zastaví. A nastaví status, že došlo ke kolizi (viz **Tab. P1.2** v **příloze 1**). Příkladem, může být případ, který se stal během testování ABBRobotBasic funkčního bloku. Špatné uchycení kabeláže na IRB1200 způsobilo, že robot se robot při pohybu (byl spuštěn program) nástrojem zachytil o právě zmíněnou kabeláž a vyhodnotil kolizi. V tomto případě přešel ABBRobotBasic do chyby. Z B&R strany pak bylo potřeba potvrdit odstranění chyby, resp. kolize. V případě, že by na reálném pracovišti došlo ke kolizi, je potřeba robota přepnout do manuálního režimu a pousnout jej z místa kolize. Poté přepnout zpět do automatického režimu a potvrdit z B&R strany, že robot již není v kolizi.

### Bezpečnostní řetězec

Z důvodů zmíněných v přecházející kapitole nebyla možnost otestovat celé pracoviště. Byla však simulována chyba porušení bezpečnostního řetězce. Robot je tedy v chybovém stavu tak dlouho, dokud není bezpečnostní řetězec opět zapnut. Opět je potřeba potvrdit chybu z HMI.

## 7.6 Užití manuálního ovládání

Manuální ovládání robota je využito hlavně při učení bodů a pohybu robota, také pro servis, či řešení případných kolizí atd. Pro již několikrát zmíněnou koncepci se nepředpokládá ovládání pohybu robota v manuálním režimu z B&R strany. Pro tyto účely slouží FlexPendant na robotu. Je ale možné, že v budoucnu nastane požadavek pro manuální ovládání robota přímo z APC (např.: z vizualizace na panelu). V tomto případě bude pravděpodobně nutné rozšíření stavového automatu startovací sekvence robota o další stavy. Charakter ABBRobotCtrl funkčního bloku však zůstane zachován. Koncepce se ale také pravděpodobně rozšíří o další datovou strukturu s názvem řekněme *RBTemplateMovementCmdType*. Tato datová struktura bude obsahovat příkazy pro pohyb robota v jednotlivých osách a další parametry, jež se nastavují přímo na FlexPendantu. Ve výsledku se tedy může jednat o SW obdobu FlexPendantu (resp. příkazů z něj nastavitelných), která byla již zmíněna v prvotním návrhu algoritmu řízení robota, viz příloha 2.

## 7.7 Web Services a REST Api

Díky podpoře Web services na robotu (kap.:6.8) byla vytvořena aplikace pro získání diagnostických dat z robota (task běžící v CPU APC). Tato aplikace běžící na straně B&R (APC) zajišťuje sestavení TCP komunikace mezi řídicími jednotkami a následně

je z APC na IRC5 použit „http request“ pro již zmíněná požadovaná data. Na APC byla vytvořena datová struktura odpovídající struktuře na **Obr. 6.8**, do které jsou získaná data ukládána. Pro samotnou komunikaci byly použity knihovny z prostředí Automation Studio 4.4.5. Konkrétně knihovny AsHttp a AsTcp. [1].

V první fázi, testování TCP komunikace, byla použita AsTcp knihovna, obsahující funkční bloky pro sestavení komunikace a připojení klienta. V další fázi testování byla použita knihovna AsHttp, zapouzdřující funkční bloky knihovny AsTcp. Její použití se však ukázalo jako nedostatečné, neboť služba Web Services na robotu vyžaduje pro přihlášení *Rest klienta* také uživatelské jméno a heslo [17]. Tato funkcionality není v AsHttp knihovně implementována, knihovna dokáže sestavit TCP komunikaci a vyslat (přijmout) http zprávu bez použití zmíněných přihlašovacích údajů. Možnou „cestou“ může být poslední FBK z AsTcp knihovny a to *TcpOpenSsl()*. Z časových důvodů (testováno jako poslední část) a také z toho důvodu, že získání diagnostických dat z roboty není prioritou a hlavním tématem této práce, zůstala tato část rozpracována.

## 8 ZÁVĚR

Dle zadání byla provedena rešerše řídicích jednotek obou stran komunikace včetně jejich komunikačních možností. Dále byla provedena analýza jednotlivých komunikačních sběrnic. Z dostupných možností byla nakonec vybrána komunikace pomocí TCP/IP. (kapitola 4). Jednak z důvodu možnosti využití digitálního dvojčete ve vývojovém prostředí ABB RobotStudio. Důležitým faktem v rozhodování byla také cena HW, konkrétně B&R Profinet modulů, která byla nakonec rozhodujícím faktorem výběru způsobu komunikace.

Následně byla představena nová koncepce programu pro komunikaci s robotem. Kapitola 5.6 se zabývá částí této koncepce a to návrhem a řízením komunikace. Byly vytvořeny datové struktury pro komunikaci mezi řídicími jednotkami. Kompletní seznam proměnných tvořících tyto struktury je uveden v **příloze 1**. Poté byly dle návrhu koncepce platformy vytvořeny funkční bloky pro řízení robota. Výsledkem je tedy jeden funkční blok ABBRobotBasic umožňující řízení robota a to včetně spouštění programu, či učení robota novým pozicím. S ohledem na složitost realizace zmíněného funkčního bloku využívá tento FBK dílčí funkční bloky uvedené v kapitole 6.

Po realizaci způsobu řízení robota byla také nutná úprava vizualizace. Byly vytvořeny nové stránky umožňující popsané řízení robota přímo z B&R integrační platformy.

Poslední kapitola se zabývá ověřováním. Bylo provedeno testování komunikace mezi APC a IRC5. Výše zmíněné datové struktury jsou cyklicky zasílány mezi oběma členy komunikace. Problém nastal při cyklickém zasílání o s dobou cyklu 1 ms, kdy byla komunikace zahlcena. Byla tedy provedena úprava a robot s APC spolu komunikují s dobou cyklu 2 ms viz kapitola 7.1. Dále bylo provedeno testování řízení robota vytvořením instance FBK ABBRobotBasic. Podařilo se jak spustit program v robotu, tak jej i učít novým pozicím, tedy bylo dosaženo hlavního požadovaného cíle této práce. Problém však nastal při ověřování způsobu zakládání, které nemohlo být provedeno, jak z časových důvodů (vývoj obdoby Basic modulu na ABB straně zabral víc času, než bylo plánováno), tak i z důvodů popsaných v kapitole 7.4.

S ohledem na potřeby a definice průmyslu 4.0 byla pro diagnostiku dat z robota na straně řídicí jednotky APC vytvořena aplikace. Tato aplikace komunikující skrze standardizované rozhraní REST Api zajišťuje sběr diagnostických dat z robota. Při testování ale nastal problém s přihlašovacími údaji při použití zmíněných knihoven pro komunikaci, jak je popsáno v kapitole 7.7. Jedním z možných řešení je úprava hlavičky „*http requestu*“, kde by byly potřebné přihlašovací údaje uvedeny. Dalším z možných řešení vytvoření kompletně nové knihovny zahrnující v sobě použití již zmíněných přihlašovacích údajů Tyto možnosti však nebyly ověřeny jak časových důvodů, tak i proto, že se nejedná o hlavní cíl této práce.

# Literatura

- [1] B&R Industrial Automation GmbH. Produkty, Software, B&R Automation Studio Software Help, Ver. 4.4.4.64. [cit. 2018-12-16].
- [2] SKAŘUPA J. Průmyslové roboty a manipulátory: učební text [online]. Ostrava: VŠB TU, 2007. 260 s. ISBN 978 80 248 1522 0  
Dostupné z: [www.elearn.vsb.cz/archived/FS/PRM/Text/Skripta\\_PRaM.pdf](http://www.elearn.vsb.cz/archived/FS/PRM/Text/Skripta_PRaM.pdf)
- [3] RUMÍŠEK P. Automatizace: Roboty a manipulátory [online]. Brno: FSI, VUT, 2003. 31s  
Dostupné z:  
[http://ust.fme.vutbr.cz/tvareni/img/opory/emm\\_mechanizace\\_a\\_automatizace\\_roboty\\_rumisek.pdf](http://ust.fme.vutbr.cz/tvareni/img/opory/emm_mechanizace_a_automatizace_roboty_rumisek.pdf)
- [4] KOLÍBAL Z. Průmyslové roboty I: Konstrukce průmyslových robotů a manipulátorů. [skriptum]. Brno: FSI, VUT, 1993. 189 s
- [5] ŠOLC F., ŽALUD L. Robotika. [skriptum]. Brno: UAMT, FEKT VUT, 2006. 144 s
- [6] ZEŽULKA F., FIEDLER P., VAŇOUS P., CACH P. Průmyslové komunikační sítě. [skriptum]. Brno: UAMT, FEI, VUT, 2000. 72 s
- [7] ZEŽULKA, F., HYNČICA O. Průmyslový Ethernet VII: Přehled současných standardů. Automa: časopis pro automatizační techniku. Praha: FCC Public, 2008, roč. 14, č. 2, s. 26-29.
- [8] F. ZEŽULKA. Prostředky průmyslové automatizace, Brno: VUTIUM, 2004. 176 s
- [9] FIELDING, R. T. Chapter 5: Representational State Transfer (REST), Architectural Styles and the Design of Network-based Software Architectures [online]. Irvine USA: University of California. [cit. 2018-12-27]  
Dostupné z:  
[https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
- [10] Ethernet POWERLINK Standardization Group. Technology. Ethernet-powerlink.org [online]. [cit. 2018-12-06].  
Dostupné z <https://www.ethernet-powerlink.org/index.php?id=11&L=846>

- [11] B&R Industrial Automation GmbH. Produkty. Br-automation.com [online].  
[cit. 2018-12-16]  
Dostupné z: <https://www.br-automation.com/cs/produkty/industrial-pcs/automation-pc-910/>
- [12] ABB s.r.o. Produkty, ABB Robotika. New.abb.com [online]. [cit. 2018-12-16]  
Dostupné z: <https://new.abb.com/products/robotics/cs/prumyslove-roboty/irb-1600>
- [13] ABB s.r.o. Produkty, ABB Robotika. New.abb.com [online]. [cit. 2018-12-23]  
Dostupné z: <https://new.abb.com/products/robotics/controllers/irc5>
- [14] ABB s.r.o. Produkty, PLC. New.abb.com [online]. [cit. 2018-12-26]  
Dostupné z: <https://new.abb.com/plc/programmable-logic-controllers-plcs>
- [15] TAJMAC-ZPS, a. s. Výrobní program, Vertikální obráběcí centra. Tajmac-zps.cz [online]. [cit. 2018-12-26]  
Dostupné z: <https://www.tajmac-zps.cz/mcfv-1680>
- [16] Renishaw s.r.o. Výrobky, Kontrolní systém Equator. Renishaw.cz [online]. [cit. 2018-12-26]  
Dostupné z: <https://www.renishaw.cz/cs/kontrolni-system-equator--12595>
- [17] ABB s.r.o. Produkty, ABB Robotika, Vývojové centrum. New.abb.com [online]. [cit. 2018-12-27]  
Dostupné z: <http://developercenter.robotstudio.com/webservice>
- [18] B&R Industrial Automation GmbH. Produkty. Br-automation.com [online]. [cit. 2018-12-29]  
Dostupné z: <https://www.br-automation.com/cs/produkty/process-control-systems/>
- [19] ABB s.r.o. Produkty, ABB Robotika. New.abb.com [online]. [cit. 2018-12-31]  
Dostupné z: <https://new.abb.com/products/robotics/robotstudio>
- [20] ABB s.r.o. Produkty, ABB Robotika. New.abb.com [online]. [cit. 2019-05-03]  
Dostupné z:  
[https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual RAPID 3HAC16581-1\\_revJ\\_en.pdf](https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual%20RAPID%203HAC16581-1_revJ_en.pdf)

- [21] ABB s.r.o. Produkty, ABB Robotika. New.abb.com [online]. [cit. 2019-05-10]  
Dostupné z: <https://new.abb.com/products/robotics/cs/prumyslove-roboty/irb-1200>
- [22] ABB s.r.o. Produkty, ABB Robotika. New.abb.com [online]. [cit. 2019-05-10]  
Dostupné z: <https://new.abb.com/products/robotics/cs/prumyslove-roboty/irb-4600>
- [23] TAJMAC-ZPS, a. s. Výrobní program, Vertikální obráběcí centra. Tajmac-zps.cz [online]. [cit. 2018-12-26]  
Dostupné z: <https://www.tajmac-zps.cz/mcfv-1060>

# Seznam symbolů, veličin a zkratk

FEKT	-	Fakulta elektrotechniky a komunikačních technologií
VUT	-	Vysoké učení technické v Brně
B&R	-	Bernecker + Rainer
ABB	-	Asea Brown Boveri
AS	-	Automation Studio
APC	-	Automation PC
PLC	-	Programmable Logic Controller
CPU	-	Central processing unit
TCP/IP	-	Transmission Control Protocol/Internet Protocol
IRC	-	Industrial Robot Controller
OC	-	Obráběcí centrum
BM	-	Basic Modul ( <b>5.6</b> )
HW	-	Hardware
SW	-	Software
FBK	-	Funkční blok

# Seznam příloh

Příloha 1: Datové struktury komunikace Basic Modulu.

Příloha 2: Prvotní návrh komunikační struktury Basic Modulu

Příloha 3: Startovací sekvence robota – popis jednotlivých stavů

# PŘÍLOHA 1

## Datové struktury komunikace Basic Modulu: (součást knihovny RBComm)

V této příloze je uveden kompletní seznam proměnných tvořících komunikační strukturu Basic modulu, jejíž hlavní části byly pospány v **podkapitole 5.6.1**. U každé proměnné je uveden datový typ včetně popisu (pokud sám nevyplývá s názvu).

**Tab. P1.1:** Seznam proměnných uvedených ve struktuře s příkazy robotu

<b>RBTemplateRobotSystemCmdType (Only Out - from PLC/APC to Robot)</b>		
<b>Název proměnné</b>	<b>Datový typ</b>	<b>Popis</b>
SystemTime	DINT*	Systémový čas PLC/APC
MotOn	BOOL	Příkaz pro zapnutí motorů
MotOff	BOOL	Příkaz pro vypnutí motorů
PrgStart	BOOL	Program Start - spuštění programu z aktuální pozice - spuštění PLC programu
PrgStop	BOOL	Program Stop - zastavení programu a řízené zastavení robota na trajektorii
PrgStartAtMain	BOOL	Program Start At Main - spuštění programu od začátku
MotOnAndPrgStart	BOOL	Motory On a Program Start - zapnutí pohonů a spuštění programu
EmStopStateReset	BOOL	Reset EmStop - reset EmStop okruhu po nouzovém zastavení
ErrReset	BOOL	Reset chyby - reset chyby na robotu
Interrupt	BOOL	Vyvolá přerušení aktuálního programu a spustí TRAP Routine
LimSpeed	BOOL	Omezí rychlost robota
PPAtMain	BOOL	Nastaví PP na Main (PP - Program Pointer v RAPID kódu)
QuickStop	BOOL	Okamžité zastavení robota
SMConfirm	BOOL	Potvrzení přerušení vyvolané řadičem SM
SimMode	BOOL	Systém přejde do simulačního režimu
SoftStop	BOOL	Stop programu na aktuální pozici, Řízené zastavení robota (mimo plánovanou trajektorii)
StopAtEndCycle	BOOL	Zastavení robota na konci programového cyklu

<b>Název proměnné</b>	<b>Datový typ</b>	<b>Popis</b>
StopAtEndInstr	BOOL	Zastavení robota po dokončení aktuálně prováděné instrukce
Restart	BOOL	Provedení "warm" restartu systému robota
WriteAcces	BOOL	Přidělení zapisovacích práv vzdálenému systému
ModulLoad	BOOL	Příkaz pro nahrání modulu s programem do robota
ModuleName	STRING[32]	Název modulu obsahující program pro robota
ProcStart	BOOL	Příkaz pro spuštění procedury na robotu
ProcName	STRING[32]	Název procedury v nahraném modulu

\* Double Integer (32 bit)

**Tab. P1.2:** Seznam proměnných uvedených ve struktuře se statusy robota

<b>RBTemplateRobotSystemStatusType (In - from Robot to PLC/APC (Diagnostics - robot status - system variables))</b>		
<b>Název proměnné</b>	<b>Datový typ</b>	<b>Popis</b>
SystemTime	DINT	Systémový čas z poslední command zprávy
MotorsOn	BOOL	Motors ON - motory jsou zapnuty
MotorsOff	BOOL	Motors OFF - motory jsou vypnuty
ExecutingMotionTask	BOOL	Program běží - program robota běží
ProdExeError	BOOL	Robot Error - došlo k chybě při automatickém provozu robota
ExeError	BOOL	Robot Error - došlo k chybě při vykonávání programu
AutomatOn	BOOL	Robot je v automatickém režimu
EmStop	BOOL	Robot je v nouzovém režimu
MotSupTriggered	BOOL	Detekce kolize robota - systém vyhodnotil kolizi robota
CycleOn	BOOL	Program běží - program v jakémkoliv Tasku běží
LimSpeedOn	BOOL	Limitace rychlosti zapnuta
MechUnitNotMove	BOOL	Robot není v pohybu
MotSupOn	BOOL	Detekce kolize robota - aktivní
RunChainOk	BOOL	Bezpečnostní řetězec robota je uzavřený
SimModeActive	BOOL	Simulační mód je aktivní
WriteAccesAssigned	BOOL	Zapisovací práva do systému přiděleny vzdálenému klientu
SysInputBusy	BOOL	Systém Input is Busy - Systém zpracovává poslední příkaz
SafeMoveViolation	SINT*	Safe Move violation
ExecutionLevel	BOOL	0 = Base level, 1 = Trap routine, 2 = Service level
ModuleLoaded	BOOL	Zvolený modul načten
ProcExecuting	BOOL	Procedura je vykonávána
ProcNotInActModule	BOOL	V aktuálním modulu se zvolena procedura nenachází
RobInWorldZone	BOOL[10]	Robot je v pracovní stanici

\* Short Integer (8 bit)

**Tab. P1.3:** Seznam proměnných uvedených ve struktuře s aktuálními „Motion“ daty robota

<b>ABBRobotActMotionDataType (In - from Robot to PLC/APC)</b>		
<b>Název proměnné</b>	<b>Datový typ</b>	<b>Popis</b>
SystemTime	DINT	Systémový čas z poslední command zprávy
ActPosRobTarget	RobTargetType	Struktura obsahující koncový bot robota a externích os v TCP souřadnicích
ActPosJointTarget	JointTargetType	Struktura obsahující koncový bot robota a externích os v JOINT hodnotách
ActTCPSpeed	REAL	Aktuální rychlost [mm/s]
RefTCPSpeed	REAL	Referenční rychlost [mm/s]
ToolName	STRING[32]	Název aktuálně používaného nástroje
WobjName	STRING[32]	Název aktuálně užitého workObjectu

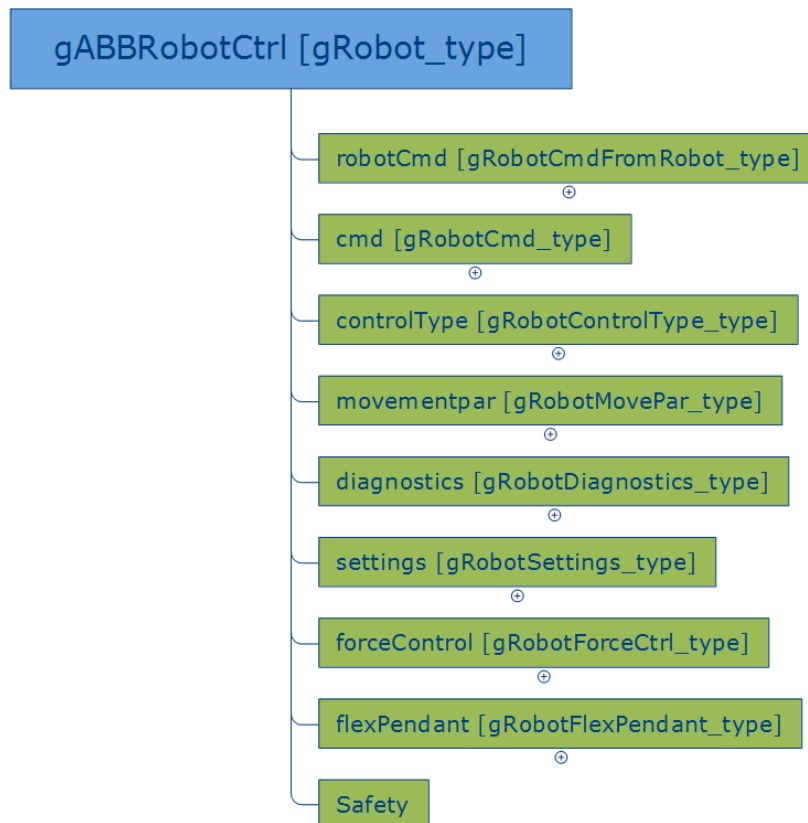
**Tab. P1.4:** Seznam proměnných uvedených ve struktuře s informacemi o robotu

<b>RBTemplateRobotInfoType (In - from Robot to PLC/APC)</b>		
<b>Název proměnné</b>	<b>Datový typ</b>	<b>Popis</b>
SystemTime	DINT	Systémový čas z poslední command zprávy
SerialNumber	STRING[32]	Sériové číslo robota
SwVersion	STRING[32]	Verze software na robotu
SwVersionName	STRING[32]	Název verze softwaru na robotu
RobotType	STRING[32]	Typ robota
ControllerID	STRING[32]	ID řídicí jednotky robota
WAN_IPAddress	STRING[32]	WAN IP adresa řídicí jednotky robota
SystemLanguage	STRING[3]	Jazyk systému na robotu (ENG, CZE, ...)
SystemName	STRING[32]	Název systému

## PŘÍLOHA 2

### Prvotní návrh komunikační struktury Basic Modulu

V této příloze je uveden bližší popis jednotlivých částí tvořících prvotní komunikační strukturu Basic modulu, jejíž popis je uveden v kapitole 6.1.



**Obr. P2.1:** Basic Modul – model komunikační struktury mezi APC a IRC5

Dále je uveden popis jednotlivých podstruktur:

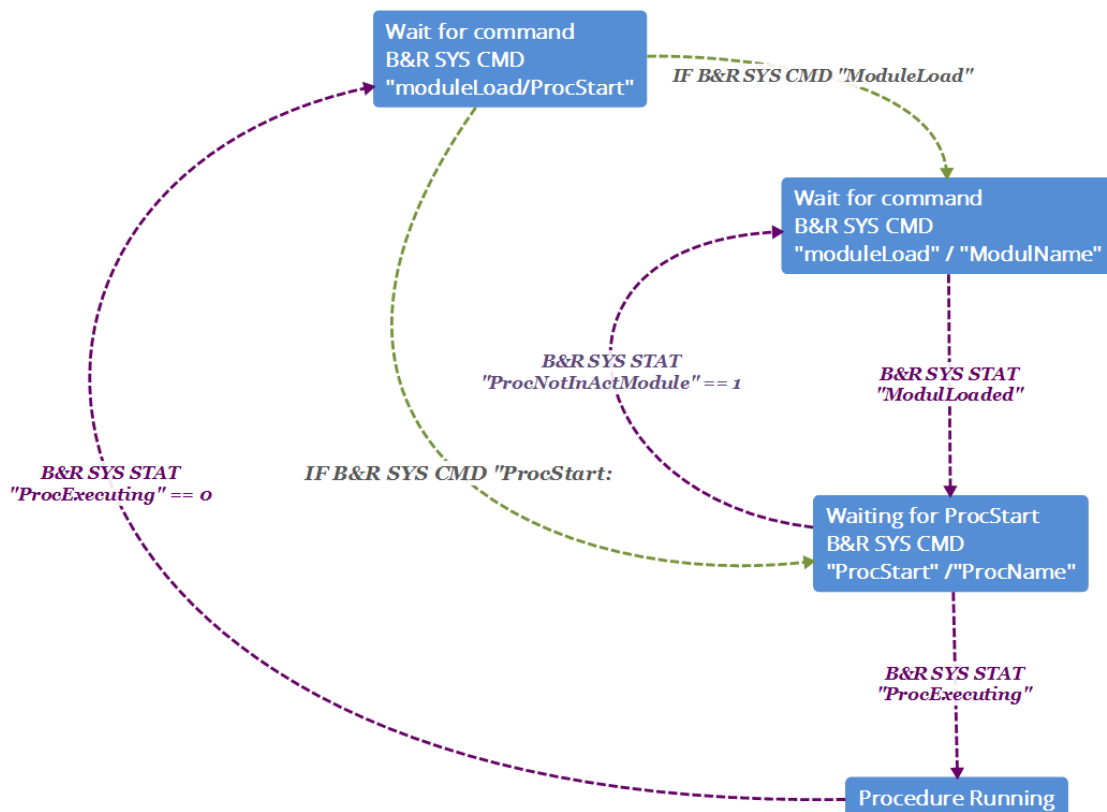
- „robotCmd“ struktura obsahuje příkazy definované na IRC5, které jsou na straně APC vykonávány. Tyto příkazy jsou použity, zejména v případě učení robota jednotlivým souřadnicovým systémům. Je zde nutná zpětná vazba od APC, že daný souřadný systém byl uložen, viz dále.
- Oproti tomu „cmd“ struktura obsahuje jednotlivé příkazy robotu (zapnout motory, spustit program, uložit nástroj, atd.). Velká část těchto příkazů je provedena během inicializace robota.

- Velice důležitou, hlavně z pohledu modularity, je „*controlType*“ struktura. Zde jsou definovány příkazy pro jednotlivé pohyby, včetně cílových pozic, jak v TCP, tak i JOINT systému.
- V „*movementpar*“ struktuře jsou uvedeny jednotlivé hodnoty souřadných systémů a také je zde definován typ pohybu (handling, cyklické zasílání pozic, jogging). Vždy je vybrán pouze jeden typ pohybu. V případě volby programu musí být veškeré spouštěné programy nahrány do robota, před samotným spuštěním Basic modulu, jinak robot zahlásí chybu. Na straně robota je pouze kontrolováno, který typ pohybu má být spuštěn a v případě programu, již zmíněná kontrola chybějícího programu.
- V „*diagnostics*“ jsou uvedeny aktuální parametry robota. Součástí je podstruktura „*robotStatus*“, která slouží jako zpětná vazba k příkazům v „*cmd*“ struktuře. Dále je zde aktuální poloha robota v TCP souřadnicích, včetně maximálních limitů v jednotlivých osách pro robot. Lze zde také kontrolovat, který typ pohybu je na robotu právě aktivní. Celá struktura je vlastně zpětnou vazbou z robota (IRC5).
- Struktura „*settings*“ uchovává data, resp. jednotlivé souřadnicové systémy, jež byly manuálně definovány (naučeny) použitím FlexPendantu.
- „*forceControl*“ – jedná se o pohyb s tenzometrickou zpětnou vazbou. Ten je zde možno zapnout a definovat jeho parametry.
- „*flexPendant*“ struktura je „ozrcadlením“ proměnných definovaných na IRC5, na které jsou namapovány vstupy a výstupy (jak analogové, tak i digitální) FlexPendantu.
- Struktura „*Safety*“ slouží pro kontrolu všech bezpečných stavů. (funkční bezpečnost)

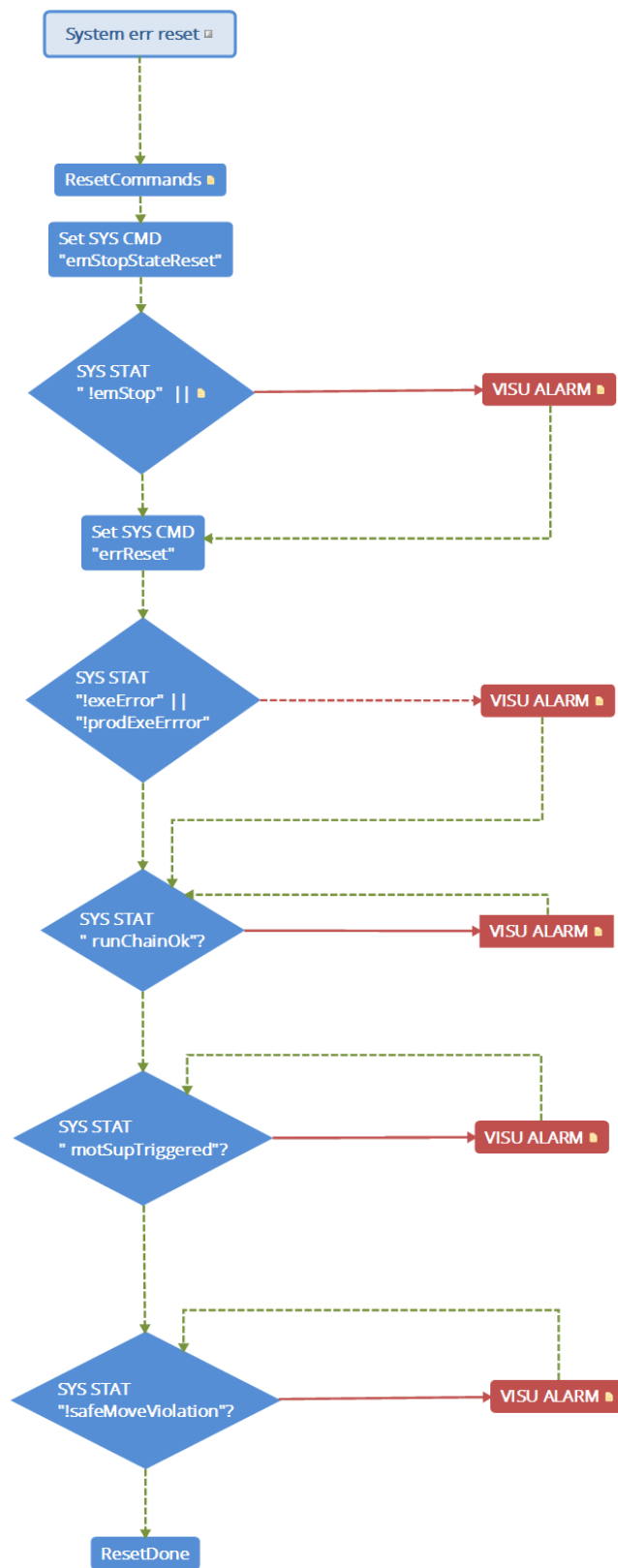
# PŘÍLOHA 3

## Startovací sekvence robota – popis jednotlivých stavů

V této příloze jsou uvedeny sekvence stavů uvnitř hlavních stavů stavového automatu startovací sekvence robota.



**Obr. P3. 1:** Stavový automat spouštění procedur v programu robota ve stavu „Program Running“ stavového automatu startovací sekvence robota



Obr. P3. 2: Stavový automat resetu chyby na robotu v programu robota ve stavu „Err Reset State“ stavového automatu startovací sekvence robota