VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

VIRTUÁLNÍ ZPROVOZNĚNÍ ROBOTIZOVANÉHO OBRÁBĚCÍHO PRACOVIŠTĚ

VIRTUAL COMMISSIONING OF THE MACHINING WORKPLACE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. Tomáš Valach

AUTHOR

VEDOUCÍ PRÁCE — Ing. Jakub Bražina

SUPERVISOR

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Student:	**Bc. Tomáš Valach**
Studijní program:	Výrobní stroje, systémy a roboty
Studijní obor:	bez specializace
Vedoucí práce:	**Ing. Jakub Bražina**
Akademický rok:	2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Virtuální zprovoznění robotizovaného obráběcího pracoviště

Stručná charakteristika problematiky úkolu:

V rámci řešení této diplomové práce je student seznámen s problematikou virtuálního zprovoznění robotizovaných pracovišť. Dochází zde k syntéze znalosti z průmyslové robotiky, automatizace a programování. Jako hlavní cíl je považováno virtuálně zprovozněné pracoviště na některé z nabízených platforem, kde je hlavním přínosem prohloubení a zdokumentování znalostí potřebných k virtuálnímu zprovoznění.

Cíle diplomové práce:

Rešerše dané problematiky.
Tvorba simulačního modelu.
Tvorba řídícího PLC a HMI.
Virtuální zprovoznění a zhodnocení výsledků.

Seznam doporučené literatury:

SICILIANO, Bruno a Oussama KHATIB. Springer Handbook of Robotics. Berlin: Springer, 2008. ISBN 978-3-540-23957-4.

KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. Brno: Vutium, 2016. ISBN 978-80-2144-828-5.

NOF, Shimon. Springer Handbook of Automation. Berlin: Springer, 2009. ISBN 978-3-540-78830-0.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

ABSTRAKT

Tato práce se zabývá virtuálním zprovozněním robotického obrábění v již existujícím robotickém pracovišti na Ústavu výrobních strojů, systémů a robotiky VUT v Brně. V teoretické části je popsána problematika virtuálního zprovoznění, využití robotů při obrábění a software použitelný pro virtuální zprovoznění. V praktické části je popsán postup při návrhu CAM operace, tvorba 3d modelu pracoviště a návrh řídícího PLC a HMI. Pro samotnou realizaci byly využity programy Fusion 360, Robot studio 2021, TwinCAT a OPC UA.

ABSTRACT

This diploma thesis deals with virtual commissioning of production system for robotic milling, which is located in the laboratories of the Institute of Production Machines, Systems and Robotics at the BUT. In the theoretical part of this thesis is described the issue of virtual commissioning, uses of robot systems in milling operations and software usable for virtual commissioning. Practical part of the thesis contains description to creating CAM operations, the creation of 3D model of workplace and PLC program and its HMI. Programs Fusion 360, Robot studio 2021, TwinCAT and OPC UA were used for the actual implementation.

KLÍČOVÁ SLOVA

Virtuální zprovoznění, Robotické obrábění, Simulace robotického pracoviště, Fusion 360, CAM, Robot Studio, UPC UA

KEYWORDS

Virtual commissioning, Robotic milling, Robotic workplace simulation, Fusion 360, CAM, Robot Studio, UPC UA

BIBLIOGRAFICKÁ CITACE

VALACH, Tomáš. Virtuální zprovoznění robotizovaného obráběcího pracoviště [online]. Brno, 2022 [cit. 2022-02-19]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/139864. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky, 82 s., Vedoucí práce Jakub Bražina.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce Ing. Jakubovi Bražinovi, za odbornou pomoc a další cenné rady při zpracování této diplomové práce. Dále děkuji svým spolužákům a známým za jejich rady a konstruktivní kritiku. Avšak hlavní dík si zaslouží moji nejbližší přátelé a moje rodina, za jejich podporu během celých 5ti let mého studia.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Jakuba Bražiny a s použitím literatury uvedené v seznamu.

V Brně dne 20.05.2022 ……

Bc. Valach Tomáš

OBSAH

1 ÚVOD 15

2 MOTIVACE 17

3 PŘEHLED SOUČASNÉHO STAVU POZNÁNÍ 19

3.1 Virtuální zprovoznění 19

3.1.1 Online programování robota 19

3.1.2 Offline programování robota 20

3.1.3 Robotické obrábění 20

3.2 Simulační programy používané pro VZ 21

3.2.1 CAM software 21

3.2.2 Tvorba modelu pracoviště 22

3.2.3 Programování PLC 24

3.2.4 Programovací jazyky pro PLC 25

3.2.5 Propojení PLC a modelu 28

3.3 Výběr softwaru 28

3.4 Bezpečnost frekvenčních měničů 29

3.4.1 Performance Level – obecně 29

3.4.2 PL Level – frekvenční měnič 30

3.4.3 SAFETY funkce frekvenčního měniče 30

4 ANALÝZA ROBOTICKÉHO PRACOVIŠTĚ ÚVSSR 33

4.1 ABB IRB 4400/60 33

4.2 Řídicí systém robotu ABB IRC5 34

4.3 Pojezd ABB IRBT 4003 34

4.4 Nástroj robotu (vřeteno) – Hiteco QD-1F 4/12 24 I30 NC CB 35

4.5 Odsavač KH – 3121 a skládaný filtr 36

4.6 Frekvenční měnič 36

4.7 Ostatní vybavení buňky 37

4.8 Komunikace v rámci pracoviště 38

5 VZ. OBRÁBĚCÍHO PRACOVIŠTĚ 39

5.1 Tvorba modelu + CAM programu 39

5.2 Virtuální model 41

5.3 PLC 45

5.3.1 GVL 45

5.3.2 PLC kód – Vřeteno 45

5.3.3 PLC kód – Robot 47

5.3.4 PLC kód – Měnič 49

5.3.5 PLC kód – Simulace 50

5.4 HMI – Human Machine Interface 51

5.5 Propojení programů 52

5.5.1 Twincat 53

5.5.2 OPC UA Gateway 53

5.5.3 RS 54

6 ZHODNOCENÍ A DISKUZE 55

7 ZÁVĚR 57

8 SEZNAM POUŽITÝCH ZDROJŮ 59

9 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK 63
9.1 Seznam zkratek 63
9.2 Seznam tabulek 64
9.3 Seznam rovnic 64
9.4 Seznam obrázků 64
10 SEZNAM PŘÍLOH 67
PŘÍLOHA 1 – PLC PROGRAM 69
PŘÍLOHA 2 – SETUP SHEET 79

1 ÚVOD

O automatizaci/robotizaci se podniky zajímají již řadu let, protože umožnuje nahradit člověka, jak na fyzicky náročných pozicích jako je manipulace s těžkými břemeny, či pracovat s malými díly s rychlostí a přesností, kterým zaměstnanec nemůže konkurovat. Tím se firmě zvedá efektivita výroby a s tím i spjatá konkurenceschopnost.

Avšak samotná integrace této automatizace do již běžící výrobní linky je časově i finančně náročná, což může mít za následek odstávky a finanční ztráty. Tyto problémy řeší virtuální zprovoznění, kdy návrh pracoviště probíhá pouze na počítači a samotná výrobní linka může pracovat do té doby, než je celý projekt nachystán a odladěn do posledního detailu.

Samotnou integrací však životní cyklus virtuálního modelu nekončí. Nadále jej lze využívat jako vizualizaci dat z provozu a monitorovat stavy jeho jednotlivých prvků.

Tato diplomová práce se bude dále zabývat předem popsaným virtuálním zprovozněním. V rešeršní části jsou popsány základní postupy zprovozňování, výhody obrábění robotem a programy nejběžněji využívané pro tento úkol. V praktické části je po jednotlivých krocích popsán proces virtuálního zprovoznění obráběcí buňky pro robotické pracoviště ÚVSSR.

2 MOTIVACE

Mým důvodem pro výběr této diplomové práce pro mě byl můj osobní zájem o vše týkající se robotizace, automatizace a celkově tvorby řídících PLC programů.

Hlavním důvodem k tvorbě této práce však bylo zavedení tohoto pracoviště do dílen ÚVSSR. Pracoviště je již v současné době funkční, avšak doposud stále chyběl kompletní model pro snadnější návrh nových pracovních operací. Takže i přesto že se jedná o teoretickou práci, tj. výstupem není hmatatelný produkt, ale programy robota, modely, PLC a simulace, v případě dostatečné časové rezervy bude možno ke konci celou moji práci vyzkoušet na již existujícím pracovišti a ověřit si tak i správnost celého obrábění.

Dále k rozhodnutí přispěla touha otestovat technologii virtuálního zprovoznění v praxi, a to bez využití jakéhokoliv hardwaru a využít tak tyto znalosti v budoucnu při hledání či vykonávání zaměstnání.

Také si nelze nevšimnout, že samotné virtuální zprovoznění nabírá na oblibě mezi integrátory a programátory robotických buněk, protože se jedná o nástroj, který jim umožnuje dosahovat požadovaných výsledků v kratším čase a zároveň vytváří základ pro tvorbu digitálního dvojčete a získávat tak i zpětně real-time data z provozu.

3 PŘEHLED SOUČASNÉHO STAVU POZNÁNÍ

3.1 Virtuální zprovoznění

Virtuální zprovoznění je virtuální reprezentace fyzického produktu nebo procesu, která se používá k pochopení a predikci výkonnostních charakteristik fyzického protějšku. Digitální zprovoznění se používá v průběhu životního cyklu produktu k simulaci a optimalizaci produktu a výrobního systému před investicí do fyzických prototypů.

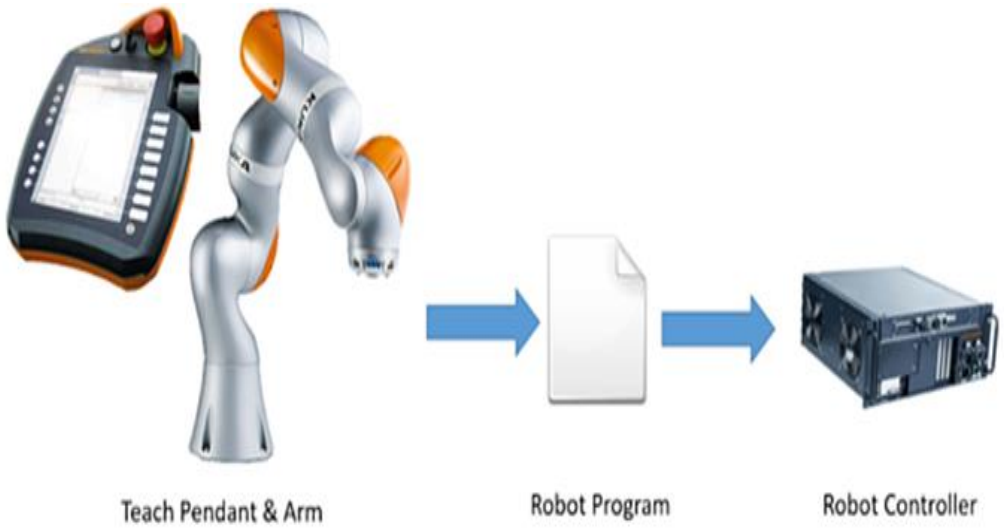
Využitím virtuálního zprovoznění produkčního systému a produktu je nyní možné výrazně zkrátit dobu potřebnou k nastavení a ověření robotického systému s integrovaným viděním a strojovým učením. Je možné tak dosáhnout robustních a spolehlivých výsledků rychleji a za mnohem nižší náklady.

Ve virtuálním prostředí se skutečný robot, díly a kamery nahrazují virtuálními. Namísto utrácení spousty času a prostředků na nastavování zařízení, pořizování mnoha snímků a ruční přidávání poznámek je nyní možné tak činit snadno a automaticky ve virtuálním prostředí.

Dalším krokem je přechod z virtuálního prostředí na fyzické zařízení – kde je skutečné zařízení nastaveno a zprovozněno. Výjimku tvoří algoritmy strojového učení, které mohou vyžadovat nějaké doplňující školení s obrázky pořízenými ze skutečné kamery. Protože je však algoritmus strojového učení již předem natrénován v digitálním verzi, bude k dosažení přesného a robustního výsledku vyžadováno podstatně méně skutečných vzorových obrázků, čímž se sníží čas fyzického uvedení do provozu, potřebné zdroje a čas na přepracování. [1]

3.1.1 Online programování robota

Online programování se stává populární metodou v malých firmách. Některé systémy totiž programování robota velmi usnadňují i ne-programátorům. Schéma principu online programování lze vidět viz. Obrázek 1.



Obrázek 1 Proces online programování [2]

Ve většině situací se setkáváme se dvěma různými typy online programování:

Teach Pendant programování

Teach pendant je zařízení, které se zapojí přímo do robota. Pomocí jeho rozhraní se přesunuje robot do požadovaných pozic a zaznamenává se každý pohyb. Některé Teach pendanty se používají snadněji než jiné. Ty s textovým vstupem umožňují používat programovací jazyk výrobce. Ty s grafickým vstupem mají tendenci se snadněji používat.

Vedení nebo ruční navádění

Robot je fyzicky veden po pracovním prostoru, je přesouván na požadovaná místa a zaznamenávají se body nebo přesná cesta. To má výhodu v tom, že je to velmi intuitivní, ale má tu nevýhodu, že je tato metoda méně přesná a často vyžaduje další senzory. [3]

3.1.2 Offline programování robota

Obrázek 2 Proces offline programování [2]

Offline programování se stává v průmyslové robotice zásadní změnou. Umožňuje snížit prostoje, urychlit integraci robota a neustále zlepšovat program robota, aniž by to mělo dopad na produktivitu. Ukázka principu viz. Obrázek 2. Ve většině situací je možné se setkat se dvěma různými typy offline programování:

Textové programování

Jedná se o tradiční metodu programování. Program je napsán v offline v textovém editoru a poté jej stáhnut do robota, až bude připraven. Pokud je využíván programovací jazyk výrobce, je pravděpodobné, že pomocí této metody získáte přístup k více funkcím robota. Má však nevýhodu, že vyžaduje mnohem více ladění a zkušeností s obsluhou než jiné programovací metody.

Grafické offline programování

Jedná se o programování robota, jako by se pohybovalo skutečným robotem, ale ve skutečnosti pohybujete jeho simulací. Po dokončení a odladění programu jej můžete nahrát do robota. Je-li software dobrý, tato metoda kombinuje to nejlepší z ostatních metod programování s dalšími výhodami offline metody. Při výběru nejlepšího softwaru však musí být programátor opatrný. Ne všechny programy jsou si adekvátní a neumožnují stejnou funkcionalitu. [3]

3.1.3 Robotické obrábění

Jednou z hlavních výhod používání robotického systému oproti tradičnímu CNC stroji je to, že robotický systém může vést k výrazně nižším investičním nákladům na vybavení než ty spojené s velkým CNC strojem. Kromě toho instalovaný robotický systém pro nakládání a vykládání materiálu snižuje náklady na pracovní sílu. Instalace dvou nebo více robotů navíc

umožňuje integrovat více robotů do výrobního procesu. Například integrací jednoho robota s obráběcím nástrojem a dalšího s efektorem pro nakládání a vykládání může výrobce dosáhnout vyšší produktivity. Navíc dosáhnout výrazného snížení nákladů ve srovnání se stejnou aplikací prováděnou CNC strojem a obsluhou.

Další věc, ve které bude mít robot navrch oproti konvenčním metodám je pracovní prostor. I když u řady CNC strojů jsou výjimky, obecně se dá říct, že jejich pracovní plocha je relativně malá. Oproti tomu středně velký průmyslový robot může využívat okolo 8 m^3 pracovního prostoru a toto číslo dále roste s přidáním externí osy v podobě pojezdu.

Poslední vlastností, ve kterém má robot navrch je produktivita. CNC pracují vždy na jednom konkrétním úkolu, složeného z frézování, soustružení, vrtání atd. Robot může dělat všechny tyto věci a další na různých pracovištích, přičemž mezi nimi může volně střídat během svého programu, a to s nižší pořizovací cenou. Příklad reálného využití je vidět viz. Obrázek 3.

Obrázek 3 Ukázka použití robotického obrábění [4]

Tyto výhody ale mají i svou cenu a roboty nemůžou se tradičnímu CNC stroji vyrovnat v přesnosti a tuhosti celého systému. Tuhost obráběcího stroje silně ovlivňuje jeho přesnost. Nástroj s nízkou tuhostí se pohne, když narazí na tvrdý materiál. To způsobí nepřesný řez. Roboty mají obvykle mnohem menší tuhost než tradiční obráběcí stroje. Robotické obrábění si snadno poradí s měkkými materiály, jako je pěna, dřevo, plast hliník atd. Tvrdší materiály, jako je ocel nebo titan, však mohou způsobit vůli v motorech robota. [4] [5]

3.2 Simulační programy používané pro VZ

V této části jsou stručně popsány programy, které se v praxi pro VZ používají. Bude zde popsán software používaný pro tvorbu CAM operací, simulace pracoviště, tvorbu PLC a finálního propojení těchto částí.

3.2.1 CAM software

Autodesk inventor

Autodesk Inventor je jedním z nejrozšířenější 3D CAD modelářů a jednotné a ucelené pracovní prostředí s moderním vzhledem, které se zdokonaluje. Všechny funkce jsou dopodrobna vyobrazeny i s popisující grafikou v nápovědě a celý systém je nativně v češtině. Do Inventoru je plně integrován i modul Autodesk Inventor CAM, který pracuje na 64-bitové struktuře, takže je schopen využít maximum zdrojů z dostupné pracovní stanice. Další možností jak nadále zvyšovat výkon je technologie Distributed CAM umožňující zpřístupnit výpočetní výkon dalších počítačů v lokální síti bez potřeby zakoupení dalších licencí. To umožnuje výpočty pro obrábění velmi složitých 3D dílců a atypicky forem dále zrychlit. Další urychlení prací umožnuje funkce pro hromadné změny řezných podmínek a parametrů v již hotové

technologií, anebo například simulátor NC-kódu s 3D vykreslením dráhy, který je již součástí licence tohoto CAD/CAM systému. [6] [7]

Fusion 360

Oproti Inventoru je Fusion 360 novější přírůstek do softwarové řady firmy Autodesk. Oproti němu se liší převážně tím, že je cloudově založený a po uživatelské stránce je přívětivější na používání. Hlavní rozdíly se nachází právě až v možnostech CAM operací. Oproti základnímu soustružení a obrábění které zvládá i Inventor, Fusion 360 umožnuje využívat i modernější technologie výroby jako je například aditivní výroba, tj. 3d tisk. Další možností jsou i laserové řezání nebo následná kontrola součástí měřením skrze sondy. Po vytvoření nebo importu dané geometrie je možné vytvořit dráhy nástroje s velkým množstvím uživatelsky nastavitelných parametrů.

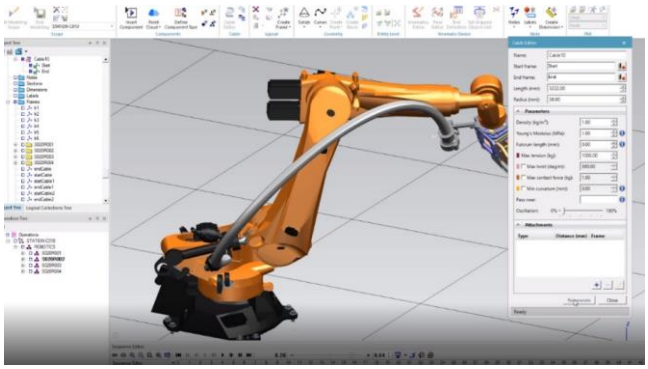
Dráhy nástroje z CAM programu je potřeba převést do jazyka řídícího systému obráběcího stroje. Takovýto překladač se nazývá postprocesor. Fusion nabízí v základní verzi softwaru rozsáhlou knihovnu těchto postprocesorů. Ty navíc fungují na bázi open source, což umožnuje nahrání vlastního postprocesoru, díky kterým se možnosti programu dále rozšiřují. [8]

3.2.2 Tvorba modelu pracoviště

Process Simulate

Process Simulate se dá rozdělit na 2 hlavní prvky Process Simulate Human a Process Simulate Robotics. První z nich se zaměřuje na simulace operací obsluhy (kroky, přesun materiálu, Pick&Place operace). Analýzy těchto operací včas zjistí potencionální rizika a díky tomu je možné optimalizovat ergonomičnost daného pracoviště, očekávanou zátěž samotné obsluhy, nebo celkovou bezpečnost pracoviště.

Druhá část se zabývá samotnou linkou, a to jak simulací např. obráběcích center, tak robotické obsluhy. Jedná se o simulační nástroj pro 3D zprovoznění celých výrobních linek. Po navržení simulačního modelu, lze snadno vytvářet jednotlivé signály pro řízení a následné propojení s PLC, kontrolovat dosah použitých robotů. Do základní knihovny lze kromě pár předem předpřipravených robotů nahrát i vlastní modely a následně si sám vytvořit kinematické vazby, případně použít modely přímo dodané výrobcem. Samotný program je nachystán na operace typu Pick&Place, montážní úkoly, svařování či obráběcí operace. Takto vytvořený program lze následně nahrát i do kontroleru robota. Prostředí programu lze vidět viz. Obrázek 4.



Obrázek 4 Prostředí programu Process Simulate [9]

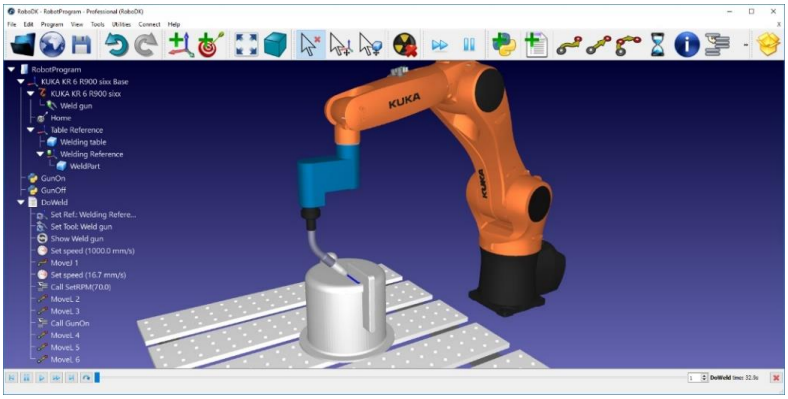
Důležité je zmínit i fakt že Process Simulate je multiplatformní, což znamená že s jedním softwarem je možné obsluhovat většinu robotů na trhu, případně si vytvořit vlastní kinematickou reprezentaci robota. Další nespornou výhodou tohoto programu je možnost vytváření časování jednotlivých operací, díky čemu lze navrhovat i celé linky metodami jako JIT („Just In Time“). I proto se program hodí spíše na projektování celých továren než malých buněk. [9]

RoboDK

RoboDK je vysoce všestrannou vývojovou platformou pro off-line programování a simulaci průmyslových robotů, která podporuje mnoho průmyslových robotů od předních výrobců, jako jsou ABB, KUKA, Yaskawa, Adept a mnoho dalších.

Prostředí 3D simulace nabízí mnoho funkcí. Uživatel může přehrávat program a vizualizovat každý aspekt chování robota, software generuje upozornění, když jsou detekovány singularity robota nebo možné kolize, umožňuje uživateli vizualizovat pracovní prostor robota, optimalizovat pohyby a vyhnout se tak poškození. Uživatel si může při vytváření pohybů zvolit libovolnou referenční rovinu a také může mít přehled o celém technologickém procesu a podle toho programovat, což je užitečná funkce, když je prostředí složitější, kde je spousta rušivých prvků nebo dokonce více robotů. V současné době se knihovna modelů CAD skládá z více než 200 průmyslových robotů, nástrojů a externích os, ze kterých je možno si vybrat. Uživatelé si také mohou navrhnout své vlastní nástroje a přizpůsobit konfiguraci zvoleného robota. Prostředí programu lze vidět viz. Obrázek 5.

V RoboDK lze programy vytvářet buď vizuálně pomocí integrovaného 3D simulačního prostředí, nebo psaním skriptů v Pythonu. Bez ohledu na zvolenou metodu může software generovat programy v jazycích specifických pro roboty – tj. ABB používá RAPID, KUKA používá KRL a tak dále – takže je lze snadno nahrát do skutečných kontrolerů. [10] [11]



Obrázek 5 Prostředí programu RoboDK [10]

RobotStudio

RobotStudio je PC aplikace pro modelování, offline programování a simulaci robotických buněk. Umožňuje uživateli pracovat s off-line ovladačem, virtuálním ovladačem IRC5, který běží lokálně na vlastním PC. RobotStudio používá přesnou kopii skutečného softwaru, který je používán reálnými roboty během výroby. To umožňuje provádět velice realistické simulace pomocí skutečných programů robota a konfigurací souborů, které se používají v praxi.

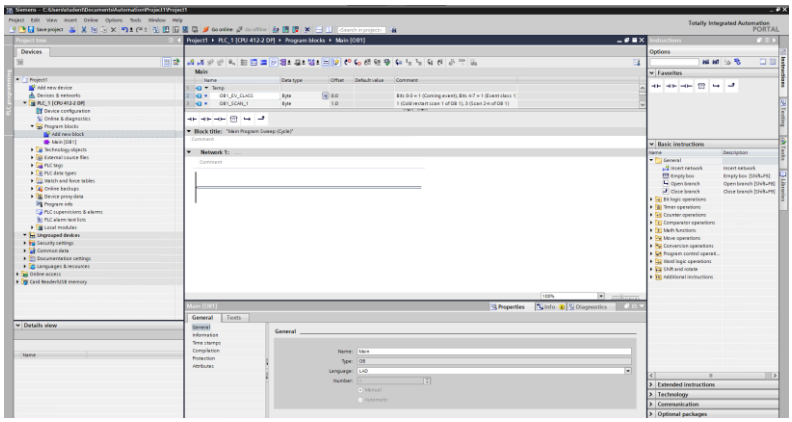
RobotStudio také umožňuje práci se skutečným fyzickým kontrolérem IRC5, v online režimu. Pokud není připojen žádný skutečný kontrolér, nebo pokud je připojen virtuální kontrolér, RobotStudio je v režimu offline.

Software RobotStudio obsahuje funkce pro programování více robotů a více řídicích jednotek a také umožňuje pracovat s různými formáty CAD. Několik specifických programovacích funkcí – AutoPath a AutoReach – umožňuje programátorovi robota ušetřit čas a námahu automatickým generováním drah robota a rozložení pracovních buněk. K dispozici je také možnost využití simulace fyzikálních vlastností jednotlivých těles a tím posunout digitálně zprovozněné pracoviště na vyšší úroveň. [12]

3.2.3 Programování PLC

TIA portal

Program TIA portal je charakteristickým svým propojením jednotlivých dílčích oblastí v rámci jednoho prostředí – hlavně pak projektování řídicích systémů, komunikační techniky, diagnostiky, funkční i informační bezpečnosti a robustnosti přirozeně podporuje stroj či strojní celek po celý jeho životní cyklus. Prostředí TIA portálu je vidět viz. Obrázek 6.



Obrázek 6 Prostředí programu TIA portál

Program TIA Portal poskytuje jednotné prostředí pro vývoj aplikačních programů pro řídicí systémy tvořené programovatelnými automaty (PLC) a decentralizovanými periferiemi (stanicemi I/O). TIA Portal umožnuje i vývoj operátorských rozhraní pro stroje a zařízení s použitím ovládacích panelů (HMI – Human Machine Interface)

V současné době se jedná o nejuniverzálnější a uživatelsky nejpřívětivější vývojové prostředí pro průmyslové automatizační aplikace. TIA Portal má dvě verze. Simatic Step 7 Basic, je určen k programování řídících systémů S7-1200 a Step 7 Professional slouží k programování všech PLC ze současné řady SIMATIC S7-1500, S7-1200, S7-400, S7-300. [13]

TwinCAT

TwinCAT (zkratka pro The Windows Control Automation Technology) je program, který firma Beckhoff představila v roce 1986 určený jako nástroj pro tvorbu automatizačních systémů. Současný stav výpočetní techniky, díky kterému přístup k CPU s vyšším počtem jader umožnuje rozložit požadované úkoly vždy na dedikované jádro pracující samostatně. Pro každé jádro používané vývojovým prostředím lze nastavit maximální zatížení i základní čas a tím i možné doby cyklu odděleně.

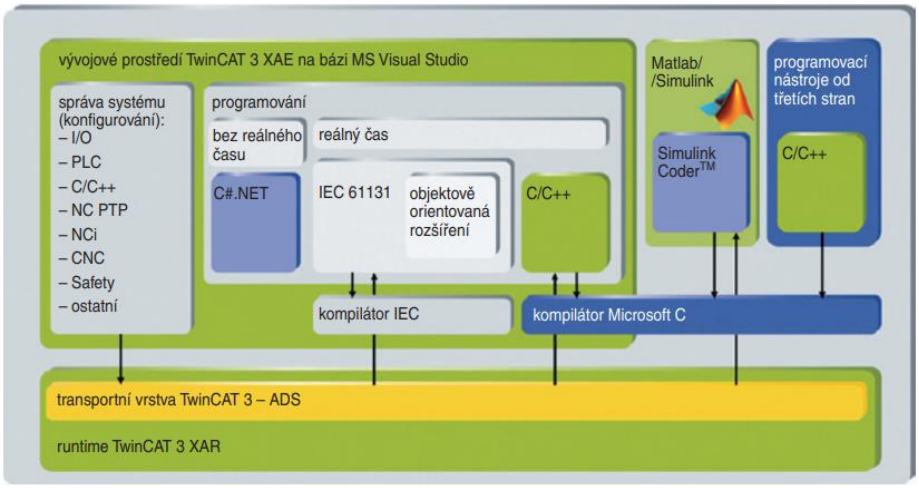
Program obsahuje 4 základní prostředí:

- PLC
- NC PTP (NC Point-to-Point, řízení pohybu os z bodu do bodu)
- NCi (NC interpolation, řízení pohybu os s použitím interpolace),
- Plnohodnotné řízení CNC nebo řízení robotů

Struktura TwinCAT 3 je tvořena těmito základními částmi:

- Vývojovým prostředí (eXetended Automation Engineering – XAE)
- Runtime (eXetended Automation Runtime – XAR)

Twincat běží na Protokolu ADS což je transportní vrstva. Ten byl navržen pro výměnu dat mezi různými prvky softwaru. Protokol nabízí svobodně volit mezi jinými nástroji pro komunikaci s ostatními zařízeními. To znamená, že v síťovém systému jsou všechny data přístupná z libovolného požadovaného bodu. Struktura programu je vidět viz. Obrázek 7.



Obrázek 7 Struktura vývojového prostředí TwinCAT 3 (XAE) [14]

Stažení základního vývojového prostředí programu je zcela zdarma. Plné runtime prostředí se všemi různými doplňky si lze stáhnout a spustit po dobu jednoho týdne. Jakmile týden uběhne, stačí vyplnit CAPTCHA kontrolu a je možno pokračovat v práci další týden. To umožňuje testovat a hodnotit (a dokonce vyvinout celý produkt) před jeho zakoupením. Takže veškeré licence, které se pro reálnou implementaci dokupují, je možné předem vyzkoušet před učiněním finančních závazků.

Navíc, oproti jiným softwarům podobného druhu je Twincat „relativně“ hardwarově nenáročný, takže jej lze bez problémů spustit i na starším noteboku. To je výhodné například i pro studenty či osoby které se zajímají o problematiku programování PLC. Mohou si pořídit software a učit se v něm či pracovat i bez větších finančních závazků. [14]

3.2.4 Programovací jazyky pro PLC

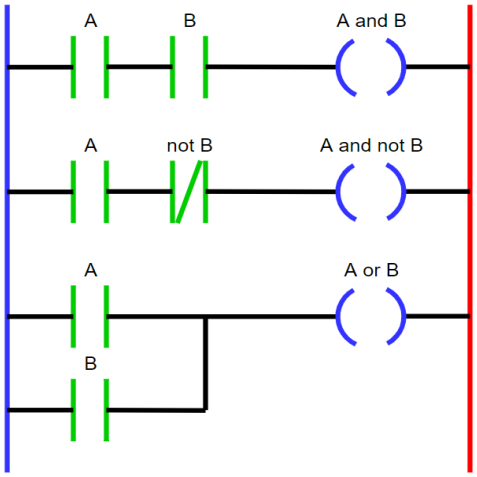
V této části jsou stručně popsány základní programovací jazyky používané pro PLC definované evropskou normou IEC EN 61131-3.

LD – Ladder Diagram – reléové schéma

Ladder Diagram je grafický jazyk, určený pro programování automatických systémů, a je užíván již od druhé světové války. Do dnešních dob je nejstarším a stále jedním z nejpopulárnějších jazyků, které jsou určeny k programování automatických systémů.

Kombinační logika LD vzniká spojením jednoho nebo více prvků vstupu v sériovém nebo paralelním zapojení a výstupu, kterým může být relé, čítač/časovač a jiné instrukce pro externí aplikaci.

V praxi se jim také přezdívá „žebříková“ schémata, protože se podobají žebříku se dvěma vertikálními kolejnicemi (napájecí napětí) a tolika „příčkami“ (horizontálními čarami), kolik má reprezentovat řídicích obvodů. Ukázkový příklad je viz. Obrázek 8. [15] [16]

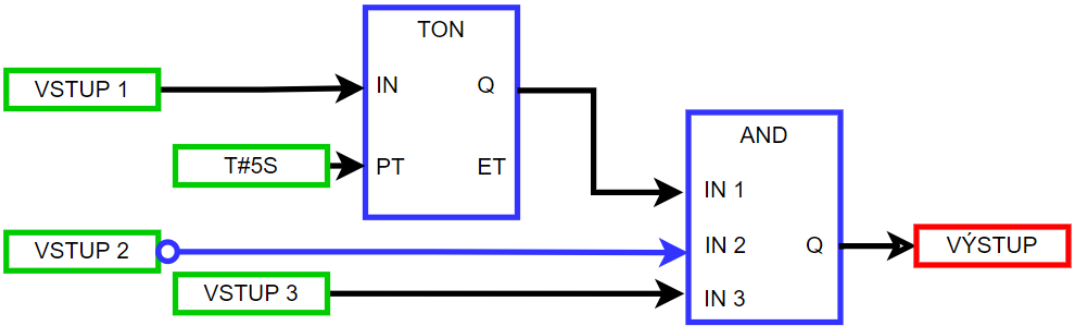


Obrázek 8 Ukázka kódu – LD

FBD – Function block diagram – funkční bloky

Funkční blokový diagram je grafické znázornění funkčního procesu prostřednictvím bloků a diagramů, které je pro čtenáře snazší pochopit a interpretovat. FBD pomáhá určit funkci mezi výstupními proměnnými a vstupními proměnnými pomocí sady základních bloků a diagramů, které jsou spojeny šipkami známými jako „spojení“.

Funkční blokový diagram může vytvořit vztahy mezi jednou nebo více než jednou proměnnou (jak vstupem, tak výstupem), aby bylo možné jednoduše porozumět funkčním procesům zarovnaných v systému. Ukázkový příklad je viz. Obrázek 9. [17]

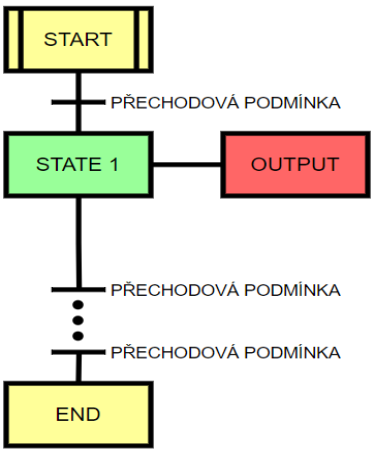


Obrázek 9 Ukázka kódu – FBD

SCF – Sequential Function Chart – sekvenční programování

Sekvenční řízení zahrnuje zpracování sekvenčních a paralelních operací, které jsou diskrétní z hlediska času nebo událostí, zejména pro řízení složitých procesních sekvencí. Programovací jazyk SFC usnadňuje vizualizaci a navrhování složitých sekvenčních systémů. V tomto programovacím jazyce jsou operace popsány jako samostatné kroky, které jsou postupně propojeny.

Když je spuštěn SFC program, má několik kroků, které následují jeden po druhém. Když jeden krok skončí, automaticky začne další. Tento automatický postup je možné regulovat přechodovými podmínkami. Termíny „kroky“ a „stav“ představují stejný koncept v programování SFC. Ukázkový příklad je viz. Obrázek 10. [18]



Obrázek 10 Ukázka kódu – SCF

ST – Structured text – strukturovaný text

Structured Text nebo jen ST je založen na tradičních programovacích jazycích jako Python nebo Java a značně se jim i podobá. Ve většině případů je možné vyvinout projekt programování PLC se strukturovaným textem bez použití programovacího softwaru PLC. Protože je založen na textu, je možné svůj projekt napsat do jednoduchého textového souboru a vložit jej poté do projektu PLC, až bude připraven.

Vzhledem k tomu, že strukturovaný text je ze své definice textový, znamená to, že nemá grafické rozhraní náročné na paměť, což vytváří menší nároky na paměť procesoru a tím i nižší náklady. Ukázkový příklad je viz. Obrázek 11. [19]

```
gvl.robot_motors_ON:=TRUE;
IF gvl.robot_motors_ON_state=TRUE THEN
    gvl.nState_automat:=31;
END_IF
```

Obrázek 11 Ukázka kódu – ST

Syntaxe strukturovaného textu se řídí některými základními pravidly:

- Všechny příkazy ve strukturovaném textu budou končit středníkem podle standardu IEC EN 61131-3
- Podmínky se uzavírají příkazem End_If.
- Mezery a tabulátory nejsou vyžadovány, ale pro čitelnost kódu jsou doporučeny

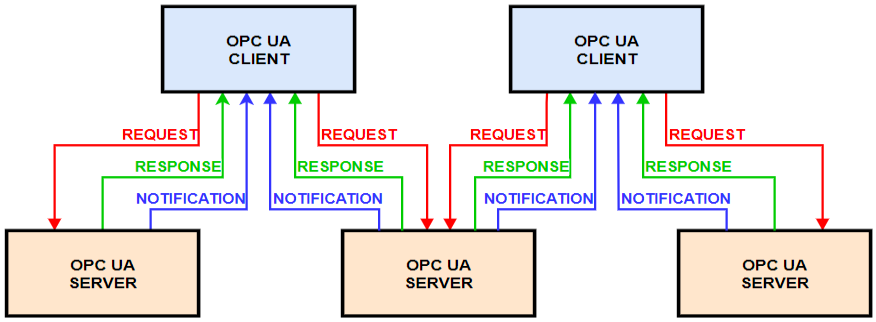
3.2.5 Propojení PLC a modelu

PLCSIM Advanced

Tento software propojuje virtuální stroj buď s virtuálním PLC, nebo s hardwarem v reálném světě, aby usnadnil komplexní simulace funkcí ovládání strojů během konfigurace. PLCSIM Advanced umožňuje uživatelům vytvořit digitální repliku CPU S7-1500 nebo ET200SP pro ověření standardního, safety či pohybového kódu bez potřeby fyzického hardwaru. Tento proces používání virtuálního řadiče pro ověřování automatizačního kódu s plnou viditelností toho, jak se stroj bude chovat, se často nazývá „software ve smyčce“. Virtuální kontrolér také poskytuje prostředek pro testování komunikace, modulů zabezpečených proti selhání, diagnostiky procesů a funkcí webového serveru. [20]

OPC UA

OPC UA (zkratka pro Open Platform Communications United Architecture) je standard pro výměnu dat pro průmyslovou komunikaci (komunikace mezi stroji nebo PC se strojem). Tento standard otevřeného rozhraní je nezávislý na výrobci nebo dodavateli systému aplikace, na programovacím jazyce, ve kterém byl příslušný software naprogramován, a na operačním systému, na kterém aplikace běží. Největší rozdíl oproti předchozím verzím spočívá v tom, že strojová data lze nejen přenášet, ale také je popisovat strojově čitelným způsobem. OPC UA poskytuje přístup k široké škále dat ve vertikálním i horizontálním směru. Spektrum sahá od komponent OPC UA přímo integrovaných do zařízení a řídicích jednotek nebo strojů a systémů až po takzvané brány a agregační servery. Princip komunikace ukazuje Obrázek 12.



Obrázek 12 Komunikace OPC UA

Komunikace mezi serverem a klientem je založená na navazování spojení, mapování kontaktů, zabezpečení komunikace a přeurčených datových typech. Každý klient v síti je nadefinován tak, že má pouze omezený počet dotazů, na které se může ptát, a každý server má předem definované odpovědi. [21]

3.3 Výběr softwaru

Praktická část diplomové práce bude provedena v těchto programech na základě zmíněných důvodů:

- Fusion 360 – Podpora postprocesoru pro ABB roboty
- Robot studio – Vhodný pro systémy s roboty od ABB
- Twincat – Osobní preference
- OPC UA – Osobní zkušenost s nastavením serveru

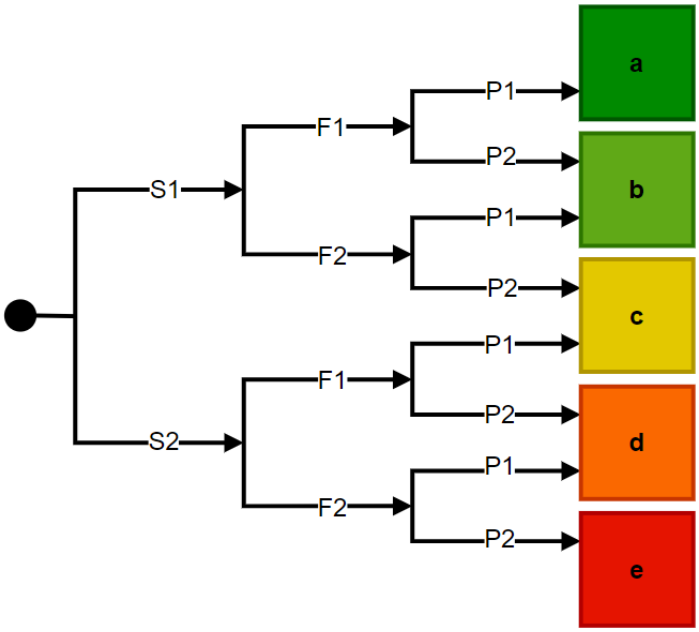
3.4 Bezpečnost frekvenčních měničů

Ke zhodnocení bezpečnosti frekvenčních měničů, a dalších částí řídícího obvodu, využíváme několik ukazatelů, jako například Performance Level (PL) nebo Safety Integrity Levels (SIL).

3.4.1 Performance Level – obecně

Úroveň výkonnosti (PL) je hodnota používaná k definování schopnosti částí řídicích systémů souvisejících s bezpečností vykonávat bezpečnostní funkci za předvídatelných podmínek. Na druhé straně požadovaná úroveň výkonu (PLr) určuje minimum, aby bylo dosaženo požadovaného snížení rizika pro každou bezpečnostní funkci. Proto musí být výkonnostní úroveň (PL) bezpečnostních částí řídicího systému stejná nebo vyšší než požadovaná výkonnostní úroveň (PLr).

Čím větší riziko, tím vyšší požadavky na řídicí systém. Příspěvek spolehlivosti a struktury se může lišit v závislosti na použité technologii. Úroveň každé nebezpečné situace je klasifikována do pěti stupňů od „a“ do „e“. S úrovní „a" je příspěvek řídicí funkce ke snížení rizika nízký, s PL „e" je vysoký. Rizikový graf lze použít k určení požadované úrovně výkonnosti (PL r) pro popsanou bezpečnostní funkci. Graf je vyobrazen viz. Obrázek 13 .



Obrázek 13 Performance Level – úrovně

V grafu se pohybujeme na základě tří parametrů:

- Závažnost zranění (S)
  - S1 = Lehké (normálně reverzibilní) poranění
  - S2 = Vážné (normálně nevratné) zranění, včetně smrti
- Frekvence a/nebo vystavení nebezpečí (F)
  - F1 = Zřídka až méně často a/nebo doba expozice je krátká
  - F2 = Časté až nepřetržité a/nebo doba expozice je dlouhá
- Možnost vyhnout se nebezpečí nebo omezit poškození (P)
  - P1 = Možné za určitých podmínek
  - P2 = Sotva možné

Každou úroveň PL lze vyjádřit číselnou hodnotou PFHd neboli pravděpodobností nebezpečného selhání za hodinu . Tato hodnota se pohybuje mezi 0.01% až 0.000001%. Přesné hodnoty pro jednotlivé úrovně jsou vypsány v Tabulka 1. [22] [23]

Tabulka 1 Performance Level – PFHd

Performance Level (PL)	Probability of Dangerous Failure per Hour (PFHd) 1/h
a	$\geq 10^{-5}\ a\ < 10^{-4}\ \Rightarrow\ \langle\ 0.001\%\ až\ 0.01\%\rangle$
b	$\geq 3 * 10^{-6}\ a\ < 10^{-5}\ \Rightarrow\ \langle\ 0.0003\%\ až\ 0.001\%\rangle$
c	$\geq 10^{-6}\ a\ < 3 * 10^{-6}\ \Rightarrow\ \langle 0.0001\ až\ 0.0003\%\rangle$
d	$\geq 10^{-7}\ a\ < 10^{-6}\ \Rightarrow\ \langle 0.00001\%\ až\ 0.0001\%\rangle$
e	$\geq 10^{-8}\ a\ < 10^{-7}\ \Rightarrow\ \langle 0.000001\%\ až\ 0.00001\%\rangle$

3.4.2 PL Level – frekvenční měnič

Frekvenční měniče nejčastěji spadají do kategorií „c“ až „e“ v závislosti na jejich konečném použití. Kategorii PL je u měničů možné změnit jejich zapojením do obvodu společně s kombinací relé a stykačů. Obvody vyžadované pro dosáhnutí jednotlivých kategorií bývají vyznačené zpravidla v manuálu daného stykače.

Při použití frekvenčního měniče jako pohon vřetene u obráběcí operace, odpovídá PL kategorii „e“:

- Závažnost zranění: S2 = Vážné (normálně nevratné) zranění, včetně smrti
- Frekvence a/nebo vystavení nebezpečí: F2 = Časté až nepřetržité a/nebo doba expozice je dlouhá
- Možnost vyhnout se nebezpečí nebo omezit poškození: P2 = Sotva možné

Porucha na tomto zařízení by mohla způsobit nežádoucí roztočení vřetene s frézou, což může skončit závažným zraněním až smrtí.

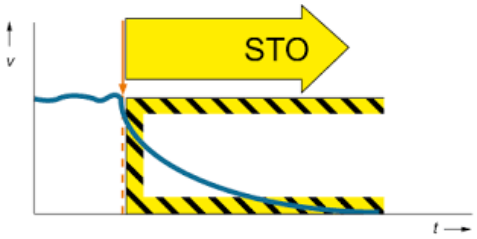
3.4.3 SAFETY funkce frekvenčního měniče

Pohony odolné proti poruchám jsou charakteristické svými integrovanými bezpečnostními funkcemi, které poskytují uživateli možné reakce na události související s bezpečností. Bezpečnostní integrované funkce dostupné v pohonech jsou popsány dále.

Funkční bezpečnost všech funkcí splňuje požadavky definované v části 5-2 mezinárodní normy IEC 61800 pro systémy pohonu s proměnnou rychlostí. Funkce SBT a SP přesahují rozsah stanovený v IEC 61800-5-2. [24]

Safe Torque Off (STO)

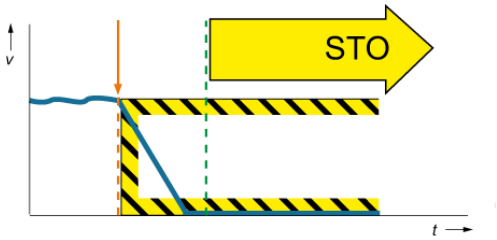
Funkce STO zajišťuje, že na motor nemůže působit žádný točivý moment tím, že blokuje elektrické signály z výkonových zařízení do motoru. Jako funkce nouzového zastavení může STO okamžitě snížit energii generující točivý moment a umožnit motoru úplné zastavení pomocí přirozené setrvačnosti a tření od zátěže. Motor by v tomto případě byl považován, že je ve stavu „volný chod“. Graf průběhu rychlosti na čase je vidět viz. Obrázek 14.



Obrázek 14 Safety funkce – STO [24]

Safe Stop 1 (SS1)

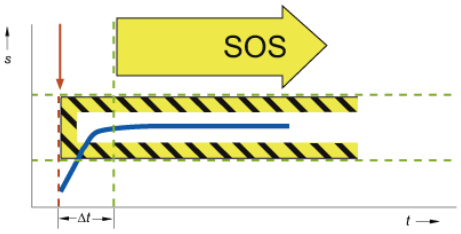
Používá řízený náběh (zpomalení) k bezpečnému zastavení motoru a poté aktivuje funkci STO. SS1 umožňuje velmi rychlé zastavení systému s vysokou setrvačností. Graf průběhu rychlosti na čase je vidět viz. Obrázek 15



Obrázek 15 Safety funkce – SS1 [24]

Safe Operating Stop (SOS)

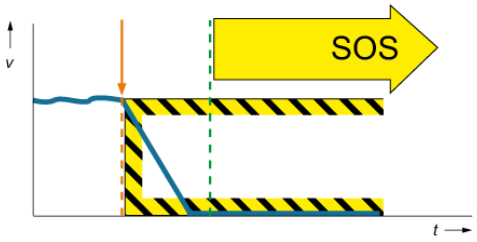
Monitoruje zastavenou polohu motoru a také sleduje odchylky od stanoveného rozsahu. Je to alternativa k STO, ale na rozdíl od STO nemusí motor úplně zastavit vyvíjení točivého momentu. Místo toho měnič zůstává v řízení polohy, drží svou polohu a je monitorován, aby detekoval nulovou rychlost. Graf průběhu polohy na čase je vidět viz. Obrázek 16.



Obrázek 16 Safety funkce – SOS [24]

Safe Stop 2 (SS2)

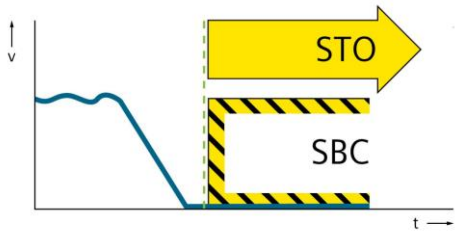
Stejně jako SS1 tato funkce používá řízené zpomalení k bezpečnému zastavení motoru, ale jakmile se motor zastaví, aktivuje funkci SOS (spíše než STO). Graf průběhu rychlosti na čase je vidět viz. Obrázek 17.



Obrázek 17 Safety funkce – SS2 [24]

Safe brake Control (SBC)

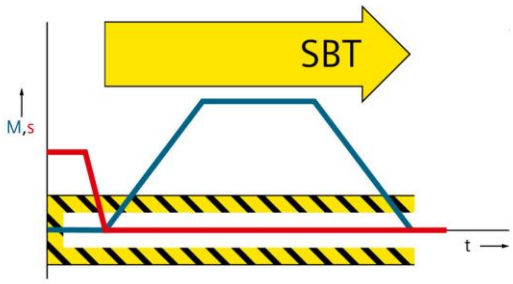
Bezpečně ovládá externí přídržnou brzdu a je vždy aktivována ve spojení s STO. SBC se obvykle používá, když je pohon vypnutý a na motor působí zatížení, jako je tomu často u vertikálních zařízení, jako jsou jeřáby a výtahy. Graf průběhu rychlosti na čase je vidět viz. Obrázek 18.



Obrázek 18 Safety funkce – SBC [24]

Safe Brake Test (SBT)

Automaticky testuje brzdu, aby se ujistil, že poskytuje požadovaný brzdný účinek. Graf průběhu momentu a polohy na čase je vidět viz. Obrázek 19.



Obrázek 19 Safety funkce – SBT [24]

Další SAFETY funkce

- Funkce pro bezpečné sledování pohybu pohonu
  - Safely-Limited Speed (SLS)
  - Safe Speed Monitor (SSM)
  - Safe Direction (SDI)
  - Safely-Limited Acceleration (SLA)
- Funkce pro bezpečné sledování polohy pohonu
  - Safely-Limited Position (SLP)
  - Safe Position (SP)
  - Safe Cam (SCA)

4 ANALÝZA ROBOTICKÉHO PRACOVIŠTĚ ÚVSSR

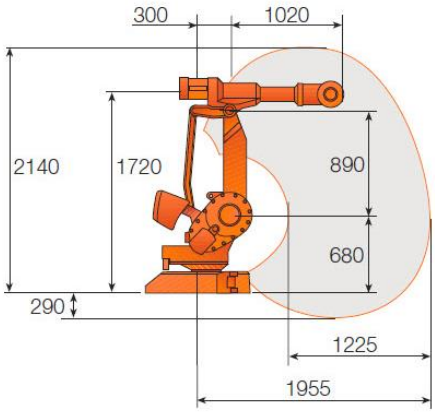
Pro účely této práce bylo rozšířeno již existující robotické pracoviště, které se nachází v prostorách dílny Ústavu výrobních strojů, systému a robotiky na FSI VUT. Samotné pracoviště v jeho reálné podobě je možné vidět viz. Obrázek 20.

Obrázek 20 Robotické pracoviště ÚVSSR

Jednotlivé prvky této buňky jsou popsány v další části práce.

4.1 ABB IRB 4400/60

ABB IRB 4400/60 je 6osý průmyslový robot s primárním určením pro výrobní průmysl. Dle výrobce se hodí hlavně na operace typu řezání, stříkání, broušení a manipulace či obsluha strojů, ale s průměrnou polohovou přesností 0,3 mm (maximum 0,7 mm) lze použít i pro obrábění, převážně hrubovací operace. Náčrt robota včetně rozměrů je zobrazen viz. Obrázek 21.



Obrázek 21 Průmyslový robot ABB IRB 4400/ 60 + pracovní obálka [36]

Koncový efektor může mít až 60 kg a lze s ním obrábět v celkové vzdálenosti až 1,96 m, kdy tato vzdálenost se dá zvětšit použitím externí osy (pojezdu) viz. kapitola 4.3.

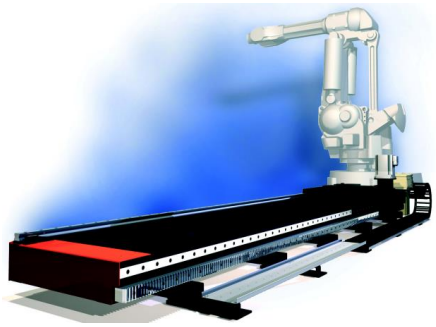
4.2 Řídicí systém robotu ABB IRC5

Řídící jednotka ABB IRC5 je již pátou generací, která v sobě kombinuje řízení pohybů, flexibilitu a bezpečnost. Produktivitu zvyšuje pokročilá diagnostika, rychlé vyšetřování poruch a monitorování stavu robota v reálném čase. Kromě toho má schopnost provádět všechny své úkoly s maximální účinností díky pokročilému dynamickému modelování. To umožňuje mít předvídatelný a vysoce výkonný robot bez nutnosti ladění jeho programu. Tento systém je také prakticky bezúdržbový. Jednou z možností ovládání řídícího systému je dodávaný FlexPendant, který je spárován s IRC5, a používá programovací jazyk RAPID a má barevnou dotykovou obrazovku a 3D joystick. Jazyk RAPID se snadno používá a je univerzálním jazykem, který podporuje strukturované programy, pokročilé funkce. Ukázka řídíci jednotky je viz. Obrázek 22. [25]

Obrázek 22 Řídící systém robotu ABB IRC5 a FlexPendant [25]

4.3 Pojezd ABB IRBT 4003

Pojezd ABB IRBT 4003 rozšiřuje pracovní prostor robota o 7 externí osu. Jeho plných vlastností využijeme až po kompletní integraci s řídícím systémem, což se týká jak dynamiky i přesnosti. V pracovišti ÚVSSR je instalovaný pojezd s využitelnou délkou 4,7 m, avšak vzhledem k layoutu celého pracoviště a umístění pracovního prostoru pro obrábění, nebude tento pojezd v rámci DP aktivně využíván při řízení robota, protože robot bude po celou dobu operace v krajní poloze pojezdu. Pojezd je vyobrazen viz Obrázek 23. [26]



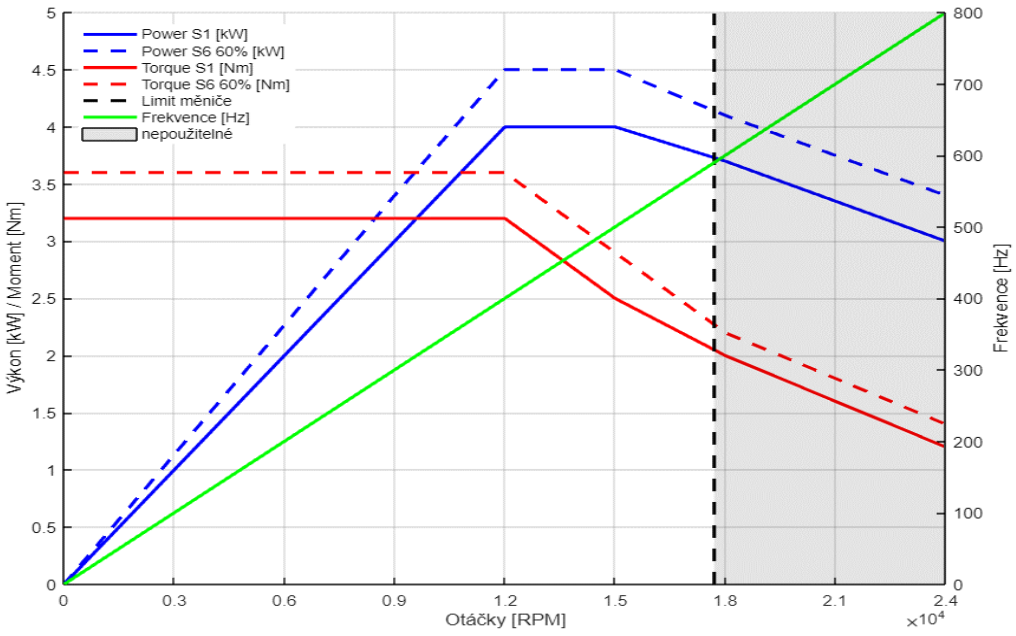
Obrázek 23 Pojezd ABB IRBT 4003 [26]

4.4 Nástroj robotu (vřeteno) – Hiteco QD-1F 4/12 24 I30 NC CB

Pro samotné obrábění byl robot vybaven 4pólovým asynchronním elektrickým vřetenem od firmy Hiteco o nominálním výkonu 4 kW (4,5 kW v režimu S6). Samotné vřeteno je schopno dosáhnout až 24 000 otáček (při nominálním zatížení 12 000 RPM), což je dostatečné i pro většinu dokončovacích operací. Pro tento typ vřetene je doporučeno zpracovávat nekovové materiály jako je dřevo, plast či pokročilé materiály jako polymery nebo kompozitní materiály. I proto bude výsledný model navržen z extrudovaného polystyrenu. Vřeteno je zobrazeno viz. Obrázek 24.

Obrázek 24 Vřeteno Hiteco QD-1F 4/12 24 I30 NC CB [27]

Na Obrázek 25 lze vidět závislost výkonu a momentu na otáčkách vřetene, která byla získána z manuálu pro toto zařízení. Do této závislosti byla podle tabulky parametrů přidána křivka odpovídající vstupní frekvenci a byla vyznačena zóna, ve které se vřeteno nebude moci provozovat kvůli limitacím frekvenčního měniče viz kapitola 4.6. [27]



Obrázek 25 Charakteristika výkonu, momentu a frekvence na RPM

Z grafu vyplývá, že frekvenční měnič omezí rozsah otáček vřetene o cca 27%, což ale nemá vlil na maximální dosažitelný výkon a moment, který je dosáhnut při otáčkách 12 000 RPM.

4.5 Odsavač KH – 3121 a skládaný filtr

Poslední problém, nad kterým je nutné se pozastavit, je způsob odstranění prachových částic produkovaných během obrábění. Pro tento účel je k buňce připojen průmyslový odsavač KH-3121. Díky sacímu výkonu 1,8 KPa je schopen efektivně odsávat částice z prostoru o objemu až 2,8m^3. I když je pracovní prostor z části otevřený, konfigurace kdy ústí sací trubice je umístěno v těsné blízkosti samotného obrobku, je sací výkon adekvátní pro samotnou operaci. [28]

V základu je odsavač prodáván s látkovým filtračním pytlem, ale pro toto použití byl pořízen přídavný kartušový filtr KH-F3121, který zvedá účinnost filtrace na 99,9 % a díky externě ovládaným stíracím kartáčům se značně usnadní i jeho údržba. Samotný odsavač i kartušový filtr je možno vidět viz. Obrázek 26. [29]

Obrázek 26 Odsavač KH - 3121 a) Základní konfigurace b) Přídavný filtr [28] [29]

Hadice od odsavače je k robotu připevněná napevno skrze přípravek na vřeteni. To omezuje automatizaci pracoviště, protože v současném stavu je nutno vyměnit efektor robota a manuálně připojit hadici.

4.6 Frekvenční měnič

Vzhledem k pořizovací ceně měničů byl prozatím použit v pracovišti měnič neadekvátních parametrů, který byl zakoupen již dříve kvůli jiné aplikaci. Měnič nemá ani dostatečný výkon, ani požadovanou bezpečnost. V rámci této práce jsem tedy volil přijatelnou náhradu do budoucna.

Měnič byl volen na základě několika základních parametrů uvedených v Tabulka 2.

Tabulka 2 Minimální požadavky na frekvenční měnič

Výkon	4 kW
Počet fází	3
Vstupní napájení	380 V
Vstupní frekvence	50 Hz
Výstupní frekvence	> 400 Hz
Komunikační sběrnice	PROFINET
Bezpečnostní úroveň dle PL	e
Bezpečnostní funkce	SS1 a lepší

Pro tuto aplikaci byl zvolen frekvenční měnič značky Danfoss produktové řady VLT® HVAC Drive FC 102 s označením 134L2281 vyobrazený viz. Obrázek 27 [30].

Jeho parametry jsou uvedeny níže:

- 4,0 KW
- Třífázový
- 380–480 V AC
- Výstupní frekvence 0–590 Hz
- IP20
- RFI filtr třídy A1/B (C1)
- S funkcí Safe Stop
- Grafický ovládací panel
- Bez lakování
- Bez doplňků napájení
- MCA-120 ProfiNet SRT
- Konstrukční velikost: A2
- Bezpečnost dle PL: c až e (dle zapojení)

Obrázek 27 Frekvenční měnič VLT® HVAC Drive FC 102 - 134L2281 [30]

4.7 Ostatní vybavení buňky

Další části výrobní buňky nejsou v této práci aktivně používány, ale jsou součástí layoutu pracoviště, a tudíž se s nimi musí počítat při tvorbě 3D modelu a robotického programu, aby v budoucnu nedocházelo ke kolizím.

Jedná se hlavně o:

- CNC soustruh Kovosvit SPM 16
- CNC vertikální frézka Kovosvit MCB 754 QUICK
- Automatická měřící stanice
- Čistící stanice na obrobky
- Zásobník na koncové efektory
- Doplňky (stoly, počítače, monitory,…)

4.8 Komunikace v rámci pracoviště

V rámci práce je ke komunikaci využito serveru OPC UA, ale reálná zařízení spolu komunikují skrze komunikační protokol PROFINET.

PROFINET je součástí normy IEC 61158 a je založený na mezinárodním standardu ethernet (IEEE 802.3). PROFINET pro svou funkci používá odstupňovanou komunikační architekturu založenou na Ethernetu. Rozlišuje se na tři základní typy:

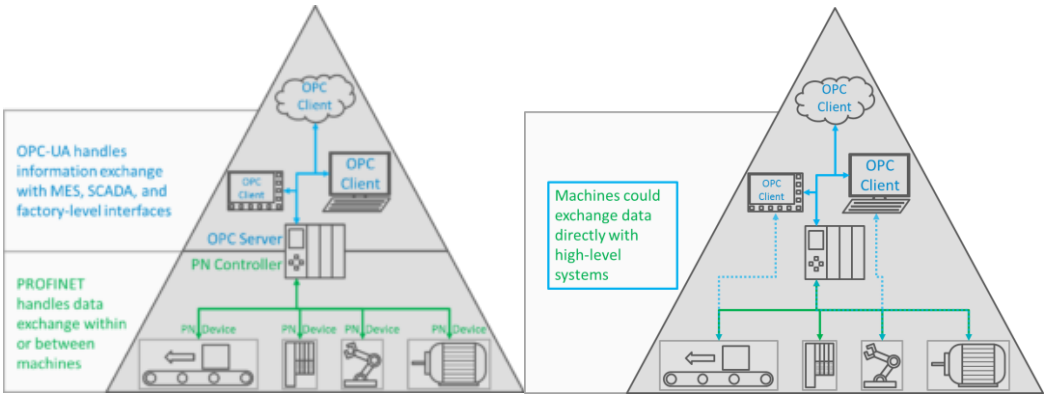
- standardní komunikace (TCP/IP) – přenos obecných informací
- komunikace pro reálný čas (RT) – důležitá data jako ovládání pohonů a safety
- izochronní reálný čas (IRT) – velký objem dat potřebný v dlouhých časových intervalech

PROFINET i OPC UA si vyměňují stejné typy dat, ale dělají to dvěma velmi odlišnými způsoby. PROFINET si vyměňuje časově kritická data v přesně řízeném formátu, který je navržen tak, aby usnadnil rychlé zpracování, minimalizoval zatížení procesoru a umožnil běh řídicích smyček s relativně malými časovými konstantami (~1 ms) po síti. OPC UA si však může vyměňovat téměř jakýkoli typ informací. Byl od základu navržen tak, aby umožňoval flexibilní komunikaci na úkor pomalejšího zpracování, vyšší latence a větší zátěže procesoru.

Historicky měly tyto dva protokoly dvě odlišné role. PROFINET se obvykle používá pro datovou komunikaci v reálném čase mezi provozními zařízeními a místními ovladači. Naopak OPC UA se obvykle používá ke komunikaci mezi těmito kontroléry.

Tento jasný rozdíl však začíná mizet, kdy zařízení PROFINET implementují své vlastní servery OPC UA a řídicí jednotky PROFINET implementují kromě serverů také klienty OPC UA. Propojení z HMI nebo nadřazeného systému přímo do vnějších zařízení je možné a jasná hierarchie automatizačního systému začíná být matoucí [31] [32].

Proto by ve výsledku zaměnění jednoho protokolu za druhý nedělalo velké problémy.



Obrázek 28 Hierarchie OPC UA a PROFINET a)Stará b)Nová [31]

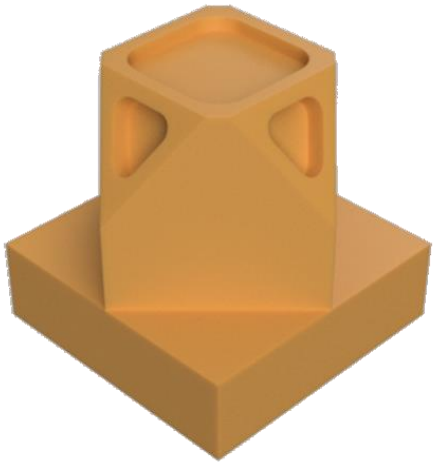
5 VZ. OBRÁBĚCÍHO PRACOVIŠTĚ

Samotné virtuální zprovoznění je rozděleno na 5 samostatných úkolů, které jsou popsány v následujících kapitolách.

5.1 Tvorba modelu + CAM programu

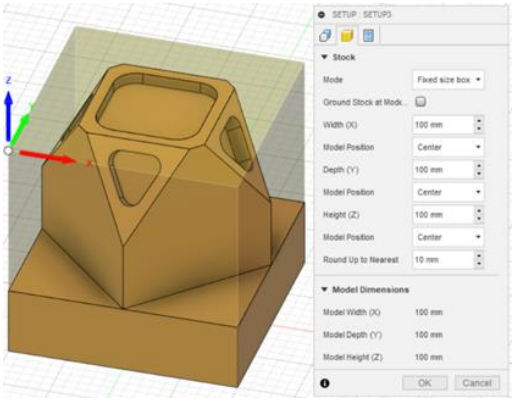
Prvním krokem bylo vytvoření modelu samotného budoucího obrobku. Cílem bylo vymyslet tvarově jednoduchý model, který by byl zároveň dostatečně složitý pro výrobu na konvenčních strojích.

Konečná verze modelu je hranol na podstavci se sraženými hranami pod úhlem 45° a kapsami směřujícími v 5 různých směrech viz. Obrázek 29. Tyto kapsy znemožňují, aby model byl obroben na klasických 3-osých strojích, bez složitého upínání nebo tvorby speciálních přípravků. Na spodní straně je ponecháno 30mm určených pro upínání do svěráku.



Obrázek 29 Výsledný model

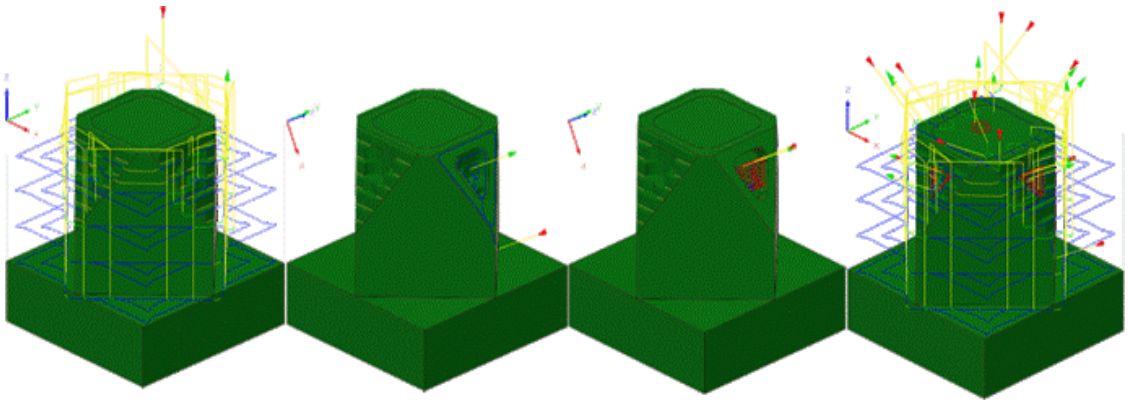
Dalším krokem bylo vytvoření CAM programu. Zde se musí v záložce setup jako první definovat typ operace, pro tento případ obrábění („milling“), nastavit rozměry polotovaru na 100x100x100mm a přesunout souřadnicovou soustavu defaultně umístěnou ve středu vrchní plochy na roh polotovaru, kde pro nás v pozdější fázi bude jednoduší této systém zaměřit robotem. Samotná nastavení karty setup je zobrazeno viz. Obrázek 30.



Obrázek 30 CAM – Setup

Pro tvorbu CAM programu byly použity 3 různé obráběcí operace. Jako první proběhne hrubovací operace „3D adaptive clearing“ která se hodí především pro odstraňování velkého množství materiálu za krátký čas. Tím vznikne hrubý polotovar nachystaný na dokončování.

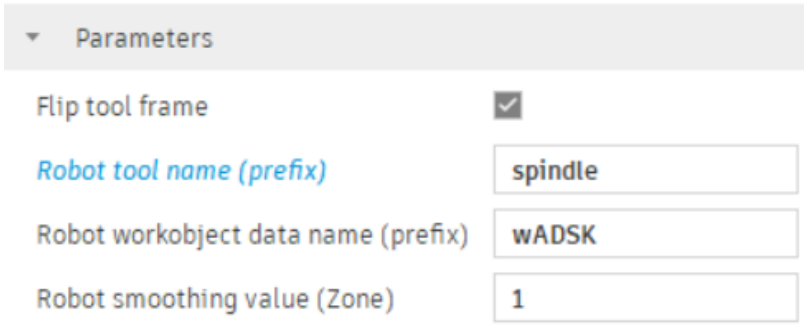
Následně proběhne operace „2D face“ která zarovná veškeré nedokonalosti („schody“) zanechané po hrubování na sražených čelech a vytvoří se kapsa operací „3D pocket“. Tyto 2 operace se následně zopakují pro všechny 4 zbylé plochy na modelu. Jednotlivé trasy operací je možné vidět níže viz. Obrázek 31.



Obrázek 31 CAM operace – a) 3D adaptive clearing b) 2D face c) 3D pocket d) všechny operace

V této fázi je možné vygenerovat G-Code který přečte většina CNC strojů, avšak roboty značky ABB používají jazyk RAPID, proto je nutné doinstalovat do Fusion 360 postprocessor, který umožní vytvořený CAM program vyexportovat jako sérii příkazů v jazyce RAPID. [33]

V samotném postprocesoru je hned několik možností nastavení, jako jsou parametry pro svařování, výchozí stav efektoru, tolerance nebo výchozí konfigurace robota, avšak pro tento případ mohou všechny zůstat v defaultních hodnotách. Jediné na co je třeba dávat pozor, je směr osy Z a prefixy pro „workobject data name“ a „tool name“. Pokud by tyto hodnoty neodpovídaly modelu, Robot Studio by nemohlo správně fungovat. Nastavení parametrů je vidět viz. Obrázek 32.



Obrázek 32 Postprocessor ABB – Parameters

Pro obsluhu pracoviště generuje Fusion 360 i Setup Sheet, na kterém jsou shrnuty všechny důležité informace o počtu nástrojů, počtu operací, velikosti polotovaru, celkovém nastavení řezných rychlostí, otáček apod. Obsluha pracoviště má pak možnost přímo nahlédnout do této dokumentace a vše potřebné nachystat a zkontrolovat předem. Ukázka Setup Sheetu lze vidět viz. Obrázek 33 a v plné délce je k nahlédnutí v Příloze 2.

Setup Sheet for Program 1001

Job Description: Setup3
Document Path: test v6

Setup

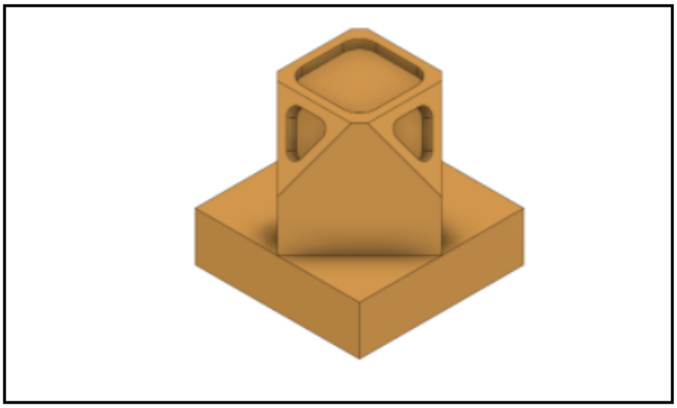
WCS: #1

Stock:
DX: 100mm
DY: 100mm
DZ: 100mm

Part:
DX: 100mm
DY: 100mm
DZ: 100mm

Stock Lower in WCS #1:
X: 0mm
Y: 0mm
Z: -100mm

Stock Upper in WCS #1:
X: 100mm
Y: 100mm
Z: 0mm

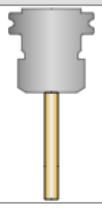


Total

Number Of Operations: 11
Number Of Tools: 1
Tools: T2
Maximum Feedrate: 5000.74mm/min
Maximum Spindle Speed: 8000rpm
Cutting Distance: 8251.61mm
Rapid Distance: 5404.55mm
Estimated Cycle Time: 3m:26s

Tools

T2 D2 L2
Type: flat end mill
Diameter: 10mm
Length: 80mm
Flutes: 3
Description: 10mm Flat Endmill
Maximum Feed: 5000.74mm/min
Maximum Spindle Speed: 8000rpm
Cutting Distance: 8251.61mm
Rapid Distance: 5404.55mm
Estimated Cycle Time: 3m:11s (92.7%)
Holder: Maritool CAT40-ER32-2.35
Vendor: Maritool
Product: CAT40-ER32-2.35

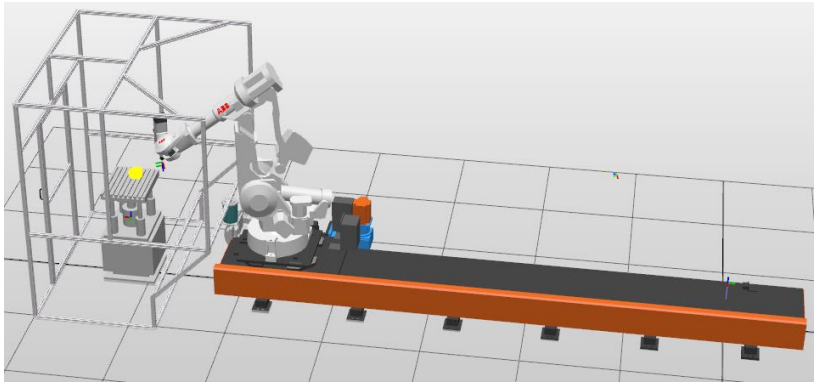


Obrázek 33 Setup Sheet

5.2 Virtuální model

Prvním krokem tvorby virtuálního modelu v Robot Studiu je vytvoření virtuálního kontroléru, bez čehož program neumožňuje další práci. Alternativou je připojit se na reálný hardware a pracovat přímo na něm, ale pro odlaďování je práce s virtuální verzí výhodnější.

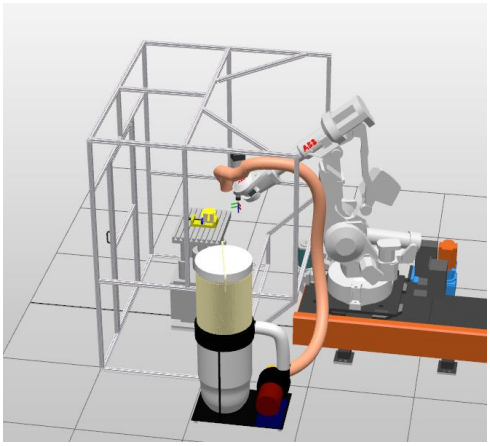
Dále je na řadě import modelů jednotlivých prvků. RS nativně obsahuje knihovny od firmy ABB, takže je možno modely pro robota, pojezd a kontrolér vložit bez větších problémů. Další prvky jako jsou unikátní držáky nástrojů, měřící stanice, či ochranná bariéra pro obrábění nejsou volně dostupné, protože se jedná buď o vlastní výrobu ÚVSSR nebo produkty dalších diplomových či bakalářských prací. Ochranná bariéra pro obrábění společně s modely soustruhu, frézy a vřetene byly dodány vedoucím práce. Původní stav modelu je vidět viz. Obrázek 34.



Obrázek 34 Základ 3D modelu – robot, pojezd, buňka

Protože ne všechny modely komponent linky jsou volně dostupné ke stažení na stránkách výrobců, bylo nutné si některé reprezentace modelů vytvořit vlastní. Bylo nutné vytvořit 3d reprezentaci odsavače prachových částic a čisticí stanici. Pro tvorbu těchto reprezentací byl použit program Autodesk Inventor 2022.

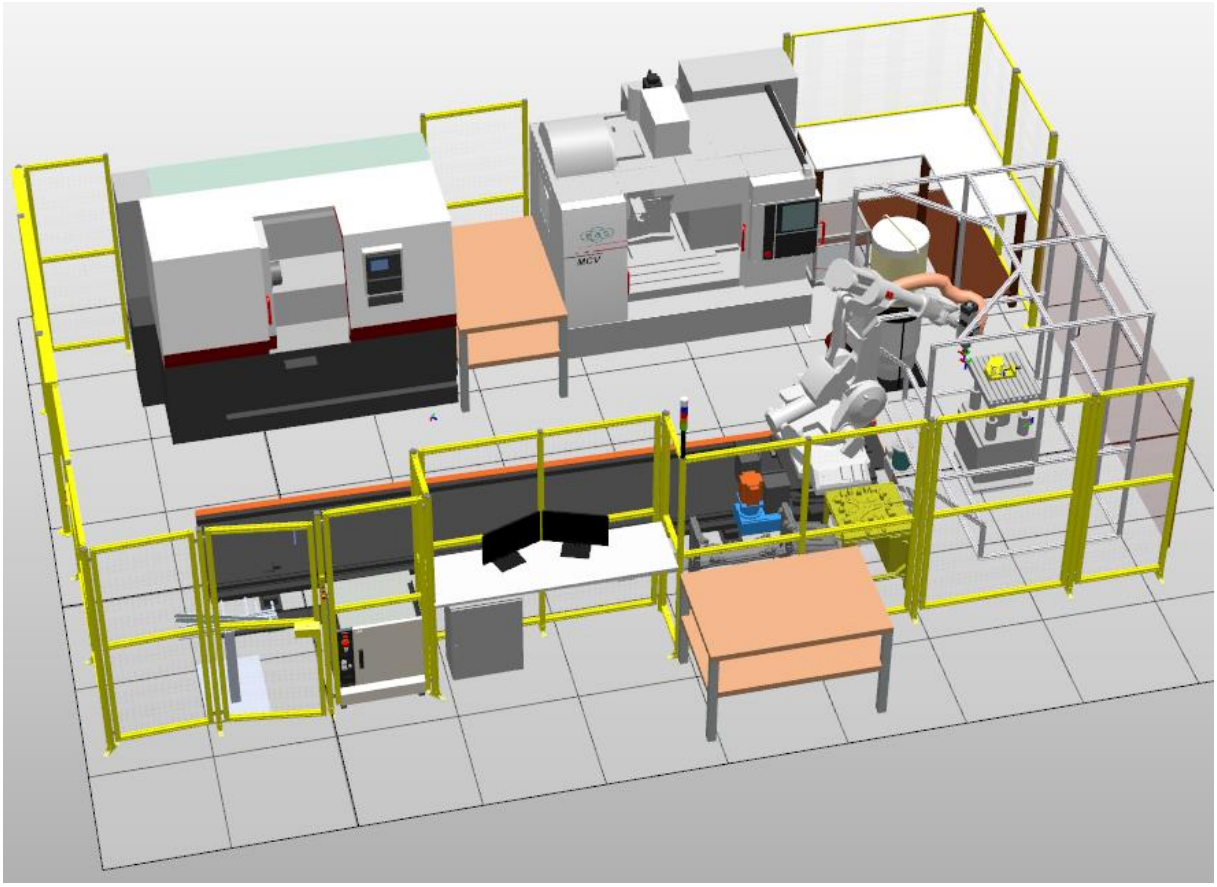
Dále byl z vlastních modelů vložen samotný polotovar v upínacím zařízení (reprezentace libovolného svěráku). Ten mohl být umístěn na náhodné místo v rámci pracovní plochy, protože jeho přesnou pozici bude stejně následně potřeba zaměřit robotem skrze workobject. Stav modelu odpovídající této fázi tvorby modelu je vidět viz. Obrázek 35.



Obrázek 35 3D model – odsavač + polotovar

Další součástí modelu linky jsou bezpečnostní prvky. Pro jejich tvorbu byl využit Addon pro RS „Equipment Builder“, který umožnuje lehce generovat síťované oplocení. Před samotnou tvorbou bylo ale nutné umístit vše ostatní na určenou pozici, protože po vygenerování oplocení je velice obtížné provádět změny a ve většině případů je potřeba s jejich generováním začít od začátku. Pro jejich pozicování byl zvolen pojezd jako referenční prvek. Polohy ostatních prvků byly odměřeny od hran referenčního prvku a následně zkontrolovány vzájemné pozice mezi sebou.

Tato kontrola pozice probíhala jak na reálném zařízení, tak z dalších modelů jiných prací využívajících linku ÚVSSR v různých stavech rozpracování. Po úspěšném vytvoření oplocení je možné do jeho okolí umístit další příslušenství, které v modelu splňuje pouze vizuální účel a neplní v modelu žádnou funkci k chodu linky. Jedná se například o stoly, monitory, vizualizace laserových bran a ovladače bezpečnostního okruhu. Finální verze modelu je vidět viz. Obrázek 36.



Obrázek 36 3D model – Kompletní

V této fázi práce je 3D model kompletní a může se přejit k tvorbě samotného programu robota. Jako první je nutné opravit defaultní pozici TCP vřetene a upravit vizualizaci frézy, aby odpovídala požadovanému nástroji. Tato CAM operace je nastavená pro čelovou frézu Ø10mm, délka 100mm (80mm přesah pod kleštinou upínače). Dále je možné naimportovat programové moduly vygenerované dříve postprocesorem. Fusion 360 generuje jak main modul, tak soubory .mod pro každou operaci a je důležité zkontrolovat, že byly nahrány všechny bez chyb. Po úspěšném importu bylo zapotřebí MAIN modul upravit tak, aby splňoval požadovanou funkcionalitu celého pracoviště. V současném stavu po spuštění simulace se celé obrábění spustí automaticky bez ohledu na stav/příkazy PLC.

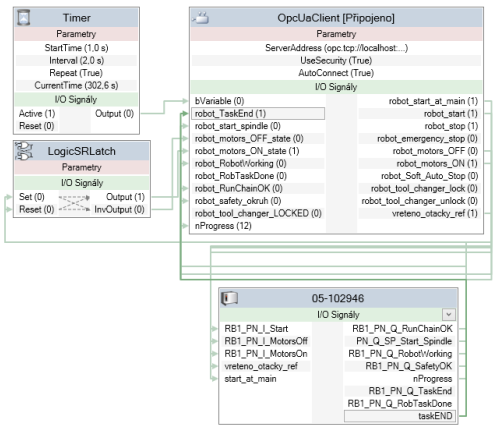
Provedené změny jsou zvýrazněny níže:

```
PROC main()
WaitDI start_at_main,1;
   ! Acceleration and jerk/ramp (percentage)
   AccSet 20,20;
   ! Configurations
   ConfJ\On;
   ConfL\Off;
```

```
    ! Load Tool
    ! Tool_Change spindle;
MoveL [ [-0.981307289,111.366794991,20.000007005],[-0.000000014,
-0.258819025,0.965925832,-0.000000043],[-1,0,-,0],
[4650,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z1,spindle\WObj:=wADSK1;
    !Spindle On/Speed
    !Spindle_ON 8000;
SetDO PN_Q_SP_Start_Spindle,1;
WaitDI vreteno_otacky_ref,1;
 SetGO nProgress,0;
    pAdaptive2;
SetGO nProgress, 1;
    pFlat1;
SetGO nProgress, 2;
.
.
ENDPROC
```

Účel těchto změn je synchronizace simulací RS a PLC. RS při startu simulace automaticky startuje program MAIN, avšak v realitě se robot spustí až po příkazu PLC. Pro simulaci tohoto chování první řádek MAIN programu pozastavuje program, dokud nezíská pokyn ke spuštění od PLC. Dále najede robotem nad workobject polotovaru do bezpečné vzdálenosti, pošle příkaz k roztočení vřetene a vyčká na potvrzení od PLC, že měnič dosáhl přednastavenou referenční hodnotu frekvence (v PLC programu simulováno 2s zpožděním odezvy) a až tehdy zahájí proces obrábění. Jako poslední byl přidán příkaz pro přepínání hodnoty proměnné nProgress pro vizualizaci postupu operace.

Poslední krok v rámci RS je vytvořit logiku stanice. Zde se propojí signály z kontroleru do UPC UA (jeho tvorba a nastavení je vidět v kapitole 5.5.3). Dále je zde vidět simulace odezvy stavu motorů. RS při zaslání příkazu na vypnutí motorů zastaví simulaci, což by v rámci VZ bylo nežádoucí, proto byly příkazy Motor ON/OFF k odezvě Motors State ON/OFF připojeny napřímo skrze RS Latch a nejdou do kontroleru. Časovač slouží pouze ke snadné kontrole spojení s OPC serverem. Zapojení logiky stanice lze vidět viz. Obrázek 37. Nepřehlednost tohoto zapojení je způsobená tím, že RS neumožnuje manuálně přesouvat spojení signálů mezi jednotlivými bloky, ale vždy volí nejkratší možnou trasu.



Obrázek 37 Logika Stanice

5.3 PLC

Pro přehlednost byl kód pro PLC rozdělen na jednotlivé bloky, které jsou popsány v dalších kapitolách. V této kapitole budou pouze ukázky tohoto kódu a stručný popis funkce, v celé jeho délce ho lze nalézt v Příloze 1.

5.3.1 GVL

Pro správnou funkčnost komunikace je prvně zapotřebí vytvořit Global Variable List , dále jen GVL, ve kterém jsou definovány všechny proměnné, které se budou následně posílat mezi PLC a kontrolérem robota. Zbytek kódu PLC se průběžně na tento seznam odkazuje, proto je vhodné při pojmenovávání těchto proměnných zvolit jasný systém, aby nedocházelo k omylům a chybám díky prohození signálů v důsledku nejasných názvů. V rámci PLC je využíváno 3 druhů signálů. BOOL proměnné říkají stav části zařízení formou 1 a 0 (true, false), INT jsou 16-bitové proměnné určené ke sledování postupu obrábění a určení polohy v CASE struktuře řízení robota a datový typ WORD je využíván měničem kdy každý bit zajišťuje jiný příkaz, nebo udává stav měniče v reálném čase. Seznam těchto proměnných je vidět viz. Obrázek 38.

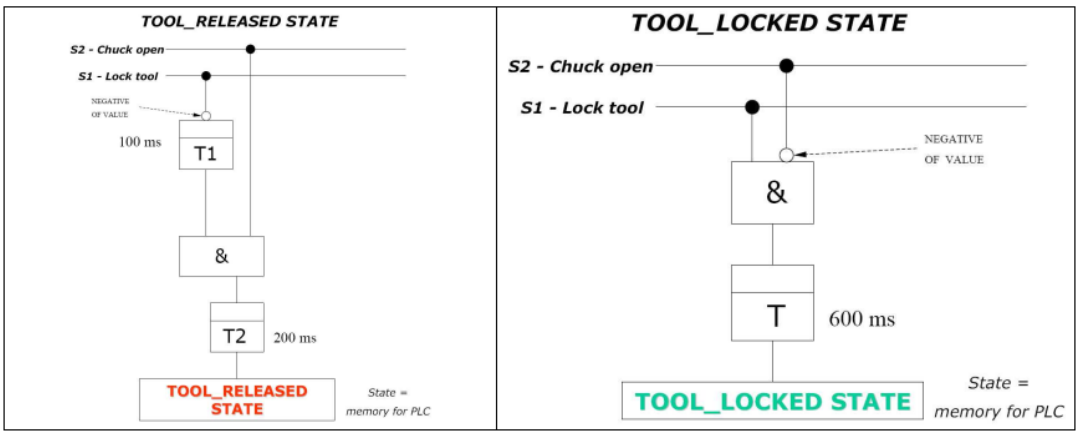
```
VAR_GLOBAL
    {attribute 'OPC.UA.DA' := '1'}

    bVariable                       :BOOL;  //testovací proměná pro ověření spojení s RS
    nState_automat                  :INT;   //pozice kodu v hlavním automatu
//vřeteno
    vreteno_S1                      :BOOL;  //tool corectly locked
    vreteno_S2                      :BOOL;  //tool holder open
    vreteno_S3                      :BOOL;  //spindle stopped
    vreteno_motor_thermal_protection :BOOL; //motor overheating
    vreteno_TC_BUTTON               :bool;  //manual tool change button
    vreteno_vzduchovy_ventil        :BOOL;  //1-unlock 0-lock
//měnič
    menic_CTW                       :WORD;  //control word
    menic_CTW_SPEED                 :WORD:=16#4000; //control word speed -100% -16384; 0% 0; 100% 16384;
    menic_STW                       :WORD;  //status word
//robot výstupy
    robot_start_spindle             :BOOL;  // robot chce zapnout vřeteno
    robot_Soft_Auto_Stop            :BOOL;  //soft stop
    robot_tool_changer_lock         :BOOL;  //zamknout hlavu
    robot_tool_changer_unlock       :BOOL;  //odemknout hlavu
    robot_emergency_stop            :BOOL;  //emergency stop
    robot_start                     :BOOL;  //spust robota
    robot_start_at_main             :BOOL;  //spustit robota v MAIN programu
    robot_stop                      :BOOL;  //zastavit úlohu robota
    robot_motors_ON                 :BOOL;  //zapnout motory
    robot_motors_OFF                :BOOL;  //vypnout motory
//robot výstupy
    robot_motors_ON_state           :BOOL;  //ON stav motorů
    robot_tool_changer_LOCKED       :BOOL;  //Shunk locked
    robot_motors_OFF_state          :BOOL;  //OFF stav motorů
    robot_RobotWorking              :BOOL;  //robot pracuje
    robot_RobTaskDone               :BOOL;  //informace o dokončení úlohy
    robot_RunChainOK                :BOOL;  //robot pripraven k provozu
    robot_TaskEnd                   :BOOL;  //informace o dokončení úlohy
    robot_safety_okruh              :BOOL;  //bool proměná ze safety okruhu bunky
    nProgress                       :INT;   // INTEAGER proměná pro sledování progresu obrábění
END_VAR
```

Obrázek 38 GVL – Seznam proměnných

5.3.2 PLC kód – Vřeteno

Vřeteno má zabudované pouze 4 senzory, které posílají signály o stavu otevření/zavření držáku nástrojů, správnosti zamčení nástroje, senzor detekující pulzy při otáčení (4 pulzy / otáčku) a senzor teploty vřetene. Dále je na vřetenu umístěno manuální tlačítko pro výměnu nástroje, které lze také připojit k PLC.

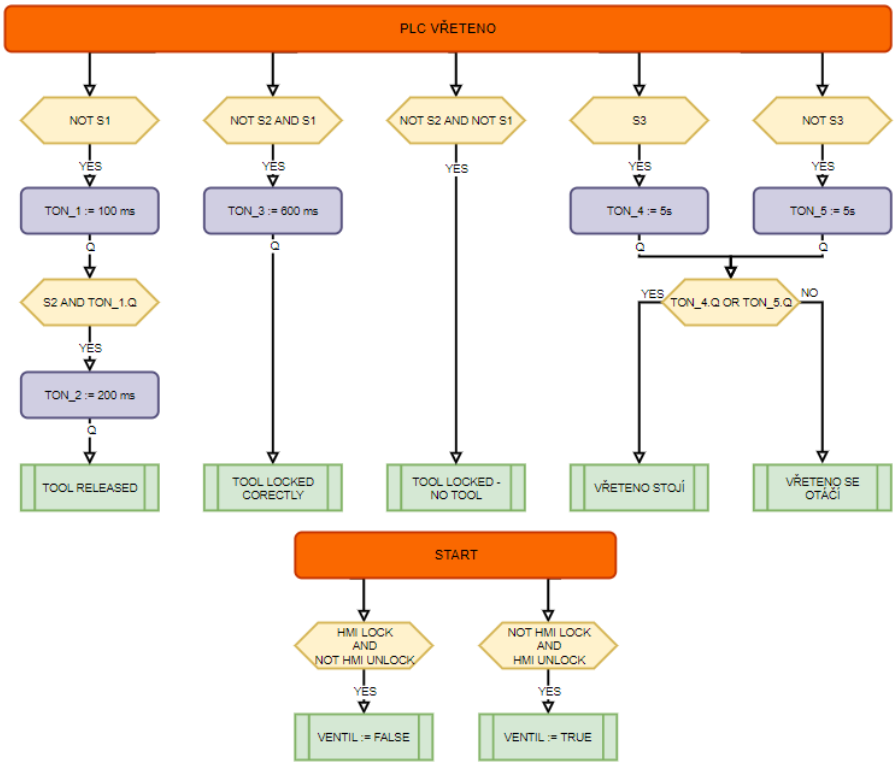
První krok je vyřešit stav upnutí nástroje ve vřeteni. Pro tento účel má výrobce v manuálu diagram zapojení signálu pro tuto indikaci, který lze vidět viz. Obrázek 39. [27]



Obrázek 39 Funkční stavy vřetene

Druhá část tohoto kódu hlídá otáčky vřetene. Ten kontroluje signál S3, který vyšle 4 pulzy na otočku. Kód pak pouze hlídá náběžné a padající hrany a v případě, že po určitý čas žádnou hranu nezaznamená, vyhodnotí vřeteno jako ustálené. Pro tento účel byly použity dva samostatné TON časovače nastavené na 5 vteřin.

Celá logika signálů vřetene je vidět viz. Obrázek 40.



Obrázek 40 Vývojový diagram – vřeteno

Níže je pak ukázka větvě diagramu kontrolující otáčky vřetene po přepsání do strukturovaného textu:

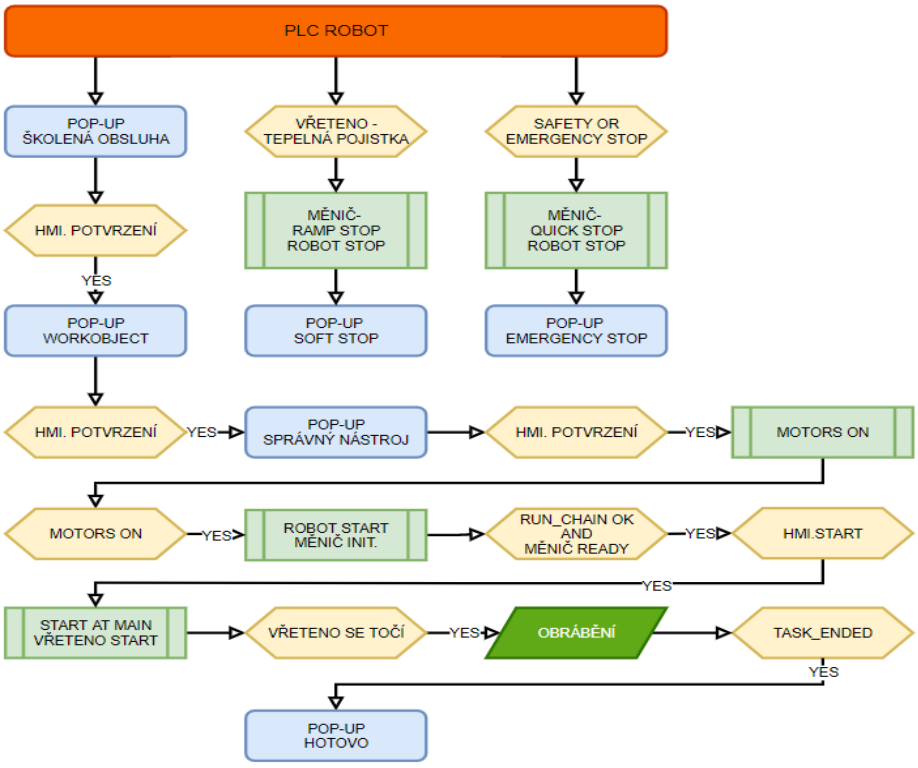
```
//otáčky
    TON_otacky1(IN:= S3 , PT:= T#5S, Q=> , ET=> );
    TON_otacky2(IN:= NOT S3, PT:=T#5S , Q=> , ET=> );
    IF TON_otacky1.Q OR TON_otacky2.Q THEN
        b_vreteno_stoji:=TRUE;
    ELSE
        b_vreteno_stoji:=FALSE;
    END_IF
```

5.3.3 PLC kód – Robot

Tato část je psaná jako jedna velká CASE struktura, která na základě proměnné gvl.nState_automat přepíná kroky, které se mají provádět. Stavy 0, 10, 20 jsou určeny pouze pro vyskakovací upozornění, které mají za účel upozornit obsluhu na možné problémy jako je špatně nastavený workobject nebo špatná fréza. Následně stavy 30, 31 jsou určeny k bezpečnému uvedení robota a frekvenčního měniče do provozu a stav 32 spustí celou obráběcí operaci do chodu. Pak už jen stav 33 čeká na signál o gvl.robot_TaskEnd a upozorní obsluhu vyskakovacím oknem o ukončení celé operace.

Dále jsou v této části jednoduché podmínky nadřazené celé operaci, které hlídají safety okruh stanice a tepelný senzor na vřetenu. Při splnění kterékoliv z těchto podmínek okamžitě zastavují robota a odstavují vřeteno/měnič. V závislosti na závažnosti je toto zastavení rozděleno na závažné, kdy dojde ke kompletnímu odstavení pohonů a resetu celého pracoviště a nezávažné, kdy dojde pouze k pozastavení pohonů.

Celá logika programu řízení pracoviště je vidět na vývojovém diagramu vyobrazeném viz. Obrázek 41.



Obrázek 41 Vývojový diagram – Robot

Níže je pak ukázka kódu po přepsání do strukturovaného textu:

```
CASE gvl.nState_automat OF
    0: //POPUP – obsluhovat může pouze školená obsluha
        b_inv_hotovo:=TRUE;
        b_inv_stop:=FALSE;
        b_inv_stop:=TRUE;
        b_inv_obsluha:=FALSE;
    30: //zapnu motory
        GVL.robot_emergency_stop:=FALSE;
        b_inv_nastroj:=TRUE;
        b_inv_safety:=TRUE;
        b_inv_stop:=TRUE;
        gvl.robot_motors_ON:=TRUE;
        IF gvl.robot_motors_ON_state=TRUE THEN
            gvl.nState_automat:=31;
        END_IF
    31: // inicializace pracoviště
        gvl.robot_start:=TRUE;      //nahoď robota
        gvl.menic_CTW:=16#047F; //Init CTW
        t1(IN:=gvl.robot_start , PT:= T#1S, Q=> , ET=> );
        IF gvl.robot_RunChainOK:=TRUE AND gvl.menic_STW.2 AND t1.Q THEN
            gvl.nState_automat:=32;
        END_IF
    32:  //start at main     ; HMI tlačítko spustí operaci
        IF b_start_operaci THEN
            gvl.robot_stop:=FALSE;
            gvl.robot_start_at_main:=TRUE;
            gvl.nState_automat:=33;
        END_IF
    .
    .
    .
    END_CASE
```

5.3.4 PLC kód – Měnič

Poslední část PLC programu potřebná pro funkci reálného pracoviště je určená k ovládání frekvenčního měniče. PLC s měničem komunikuje skrze sběrnici ProfiNET a využívá k tomu 3 proměnných typu WORD. První je Control Word, dále jen CTW, to posílá PLC měniči a dává mu tím instrukce, jak se má zachovat. Zpětnou vazbu dostává PLC skrze Status Word, dále jen STW. Opět se jedná o 16-bitový signál typu WORD, kdy každý bit předává informaci o jiném parametru stavu měniče.

Pro snadnější práci a docílení jednoduššího určení požadovaného CTW a očekáváné odpovědi STW byl vytvořen jednoduchý „překladač“ v programu excel. Význam jednotlivých bitů byl vyčten z manuálu měniče. Ukázka tohoto překladače je vidět viz. Obrázek 42. [27]

BIT	0	1	BIT	0/1	CTW
0	OFF 1	ON 1	0	1	ON 1
1	OFF 2	ON 2	1	1	ON 2
2	OFF 3	ON 3	2	1	ON 3
3	COASTING	NO COASTING	3	1	NO COASTING
4	QUICK STOP	RAMP	4	1	RAMP
5	HOLD FREQUEVENCY	USE RAMP	5	1	USE RAMP
6	RAMP STOP	START	6	1	START
7	NO FUNCTION	RESET	7	0	NO FUNCTION
8	JOG 1 OFF	JOG 1 ON	8	0	JOG 1 OFF
9	JOG 2 OFF	JOG 2 ON	9	0	JOG 2 OFF
10	DATA INVALID	DATA VALID	10	1	DATA VALID
11	NO FUNCTION	SLOW DOWN	11	0	NO FUNCTION
12	NO FUNCTION	CATCH UP	12	0	NO FUNCTION
13	PARAMERT SET-UP	SELECTION LSB	13	0	PARAMERT SET-UP
14	PARAMERT SET-UP	SELECTION MSB	14	0	PARAMERT SET-UP
15	NO FUNCTION	REVERSE	15	0	NO FUNCTION

HODNOTY		
2#	=	0000010001111111
10#	=	1151
16#	=	47F

BIT	0	1	BIT	0/1	STW
0	CONTRLO NOT READY	CONTROL READY	0	1	CONTROL READY
1	DRIVE NOT READY	DRIVE READY	1	1	DRIVE READY
2	COASTING	ENABLE	2	1	ENABLE
3	NO ERROR	TRIP	3	1	TRIP
4	OFF 2	ON 2	4	1	ON 2
5	OFF 3	ON 3	5	1	ON 3
6	START POSIBLE	STAR NOT POSIBLE	6	0	START POSIBLE
7	NO WARNING	WARNING	7	0	NO WARNING
8	STEED != REFERENCE	SPEED = REFERENCE	8	1	SPEED = REFERENCE
9	LOCAL OPERATION	BUS CONTROL	9	1	BUS CONTROL
10	OUT OD FREQUENCY LIMIT	FREQUENCY LIMIT OK	10	1	FREQUENCY LIMIT OK
11	NO OPERATION	IN OPERATION	11	1	IN OPERATION
12	DRIVE OK	STOPPED, AUTO START	12	0	DRIVE OK
13	VOLTAGE OK	VOLTAGE EXCEEDED	13	0	VOLTAGE OK
14	TORQUE OK	TORQUE EXCEEDED	14	0	TORQUE OK
15	TIMER OK	TIMER EXCEEDED	15	0	TIMER OK

HODNOTY		
2#	=	0000111100111111
10#	=	3903
16#	=	F3F

Obrázek 42 Dekódování a) CTW b) STW

Požadovaná výstupní frekvence je zapsaná do další 16-bitové proměnné (Main Reference Value, dále jen MRV) ve formátu procentuální hodnoty přednastavené reference v měniči. V závislosti na nastavení parametru 3-00 Reference Range se tato hodnota nastavuje buď od -100% do +100% (16#C000 do 16#4000) nebo 0% až 100% (16#0 až 16#4000)

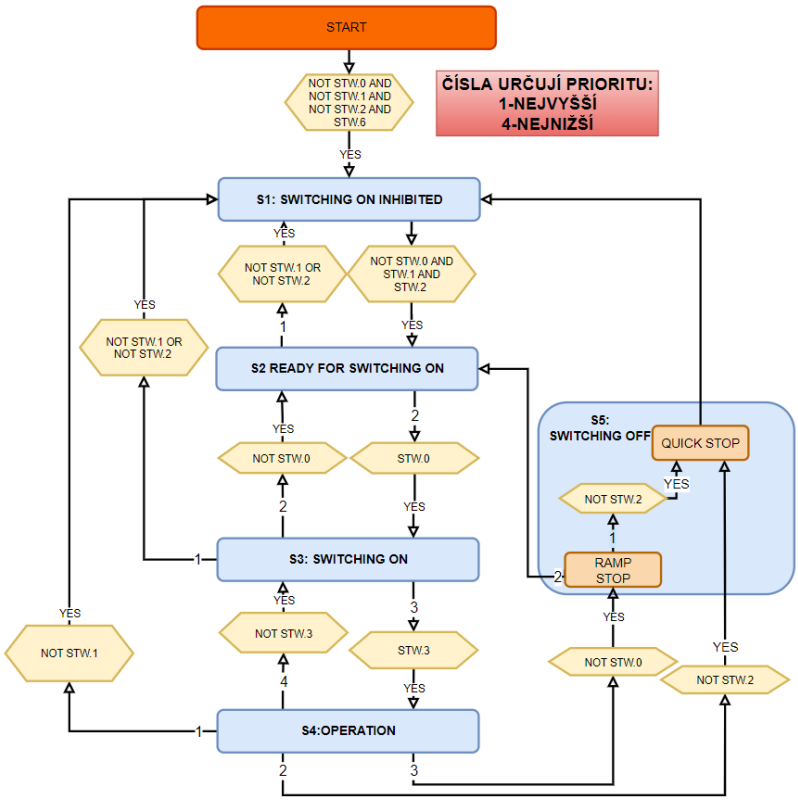
Ukázka výpočtu (Lineární interpolace):

- Parametr 3-00 Reference Range nastaven na [0] Min-Max. (16#0 až 16#4000)
- Parametr 3-02 Minimum Reference nastaven na 100 RPM.
- Parametr 3-03 Maximum Reference nastaven na 18 000 RPM
- Požadovaná hodnota 15 000 RPM

$$\begin{aligned} \mathrm{MRV} &= 16\#0 + (\text{Požadovaná hodnota} - \mathrm{Min.\,Ref.})\frac{16\#4000 - 16\#0}{\mathrm{Max.\,Ref.} - \mathrm{Min.\,Ref.}} \\ &= 16\#0 + (10\#15\,000 - 10\#100)\frac{16\#4000 - 16\#0}{10\#18000 - 10\#100} = \mathbf{16\#3546} \end{aligned} \quad (1)$$

Před samotným výpočtem je nutno převést všechna čísla do jednotné soustavy (10# desítková nebo 16# šestnáctková). Tento výpočet lze buď integrovat do samotného PLC a využívat tak rozdílnou rychlost pro různé operace, nebo jako v tomto případě, kdy je obrábění prováděno konstantní rychlostí, stačí výpočet provést pouze jednou a výsledek zavést jako konstantu pro PLC.

Samotný měnič přeskakuje mezi 5 různými stavy, SWITCHING ON INHIBITTED, OPERATION, READY FOR SWITCHING ON, SWITCHING ON, SWITCHING OFF. Mezi nimi přepíná interní logika vyobrazená na vývojovém diagramu viz. Obrázek 43.



Obrázek 43 Vývojový diagram – Měnič

Pro základní funkcionalitu měniče byly vytvořeny 2 CTW pro příkazy START a STOP (CTW pro inicializaci je definován v manuálu měniče) a jejich očekávaná odezva STW. Hodnoty pro tyto stavové proměnné lze vidět viz. Obrázek 44.

	INIT	START	STOP
CTW			
2# =	0000010001111111	0001010001111111	1000110001111111
10# =	1151	5247	35967
16# =	47F	147F	8C7F
OČEKÁVANÁ REAKCE STW			
2# =	0000011000111110	0000111100111111	0001011000101100
10# =	1598	3903	5676
16# =	63E	F3F	162C

Obrázek 44 CTW a STW pro INIT, START a STOP

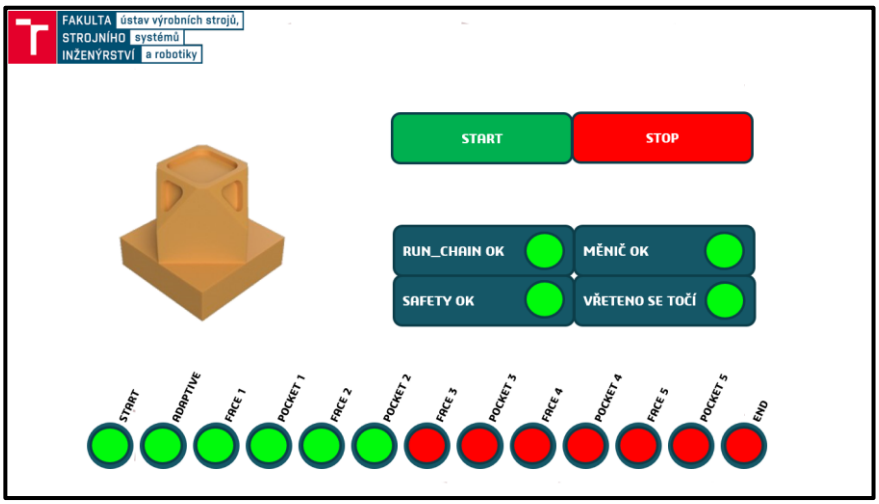
5.3.5 PLC kód – Simulace

Kvůli zaručení správného chování PLC programu i bez reálného hardwaru je nutné u některých prvků simulovat virtuální odezvu. Jedná se například o frekvenční měnič, který v současné době ještě není zakoupen, nebo zpětnou vazbu senzorů na vřeteni, které není k PLC zatím nijak připojeno. Zakomentováním této části v programu MAIN potvrdíme PLC, že byl připojen reálný hardware a simulace již není potřeba.

5.4 HMI – Human Machine Interface

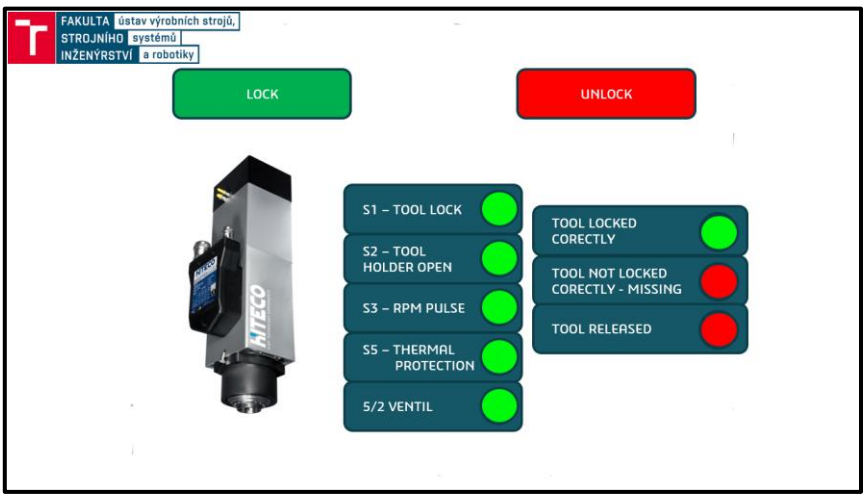
Celý program PLC je psaný s myšlenkou, že po aktivaci buňky pojede proces sám, bez zásahů obsluhy. Této myšlence odpovídají i vizualizace HMI, a proto jsou panely spíše informativní, obsahují minimum ovládacích prvků a slouží hlavně ke kontrole důležitých signálů pracoviště. Vizualizace je vytvořená pro dotykový panel CP29xx-0010 značky Beckhoff velikosti 15,6“ palců (rozlišení 1366 x 768). HMI obsahuje 4 karty, mezi kterými lze v horní části přepínat. [34]

První panel „PROCES“ slouží k samotnému spouštění výrobního procesu a obsahuje signalizaci o připravenosti robota, měniče, safety okruhu linky a informaci o otáčkách vřetene. První panel je vyobrazen viz. Obrázek 45.



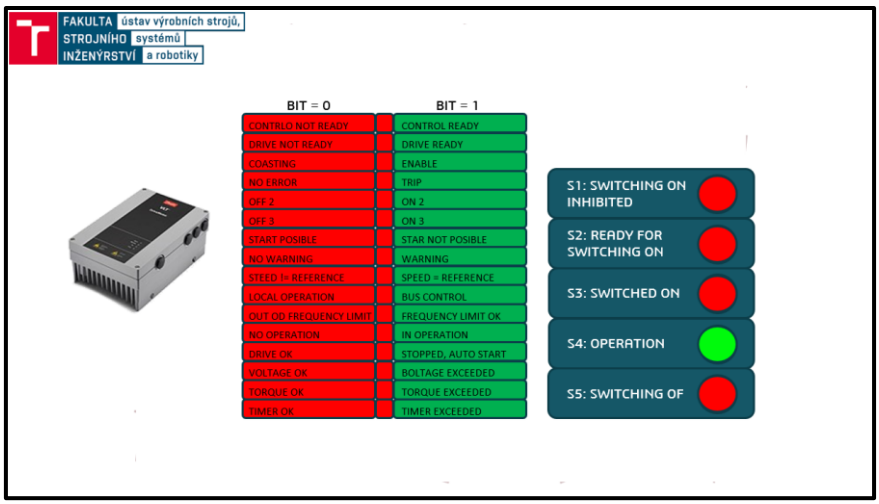
Obrázek 45 HMI – PROCES

Druhý panel ukazuje aktuální stav vřetene. Na něj jsou vyvedeny 4 výstupní signály pro kontrolu upnutí nástroje, otevření vřetene, senzor otáček, a teplotní čidlo. Na základě těchto senzorů PLC vyhodnotí stav vřetene a zobrazí ho. Dále panel obsahuje vzdálené ovládání pro upínání nástroje do vřetene a k tomu stav příslušného pneumatického ventilu. Tento panel je zobrazen viz. Obrázek 46.



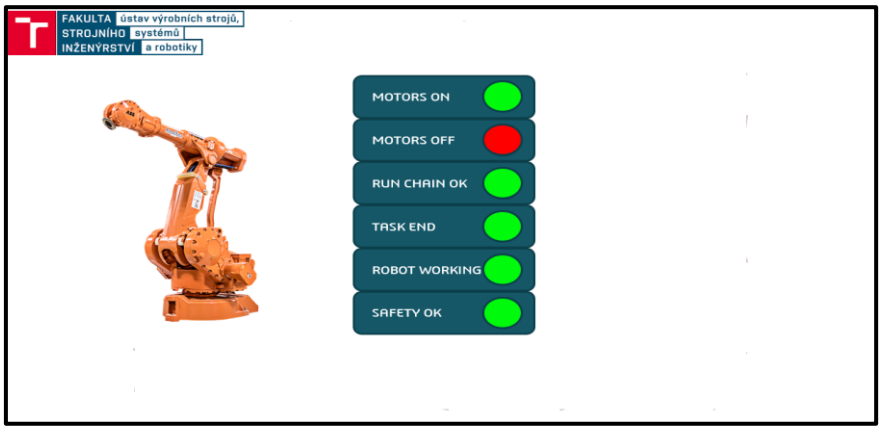
Obrázek 46 HMI – Vřeteno

Třetí karta ukazuje stav frekvenčního měniče. Krom indikace funkčních stavů S1 až S5, obsahuje tato obrazovka „dešifrování“ aktuálního STW, které hlásí měnič. Ukázka této karty je vidět viz. Obrázek 47.



Obrázek 47 HMI – Frekvenční měnič

Poslední karta je určená pro signalizaci stavu robota. K tomu je přehledně vyobrazeno několik nejdůležitějších parametrů vyčítaných z IRC5 kontroleru. Především se jedná o stav motorů, zdali je robot připraven k operaci („run chain ok“) a jestli robot v danou chvíli pracuje či už úkol dokončil („robot working“ a „task end“). Mimo jiné je zde signalizace bezpečnostního okruhu celé buňky, kdy pro přehlednost jsou všechny prvky SAFETY okruhu spojeny do jedné proměnné. Vyobrazení této karty je viz. Obrázek 48.



Obrázek 48 HMI – Robot

5.5 Propojení programů

K propojení RS a PLC (programu Twincat) bylo využito lokálního OPC UA serveru. [35]

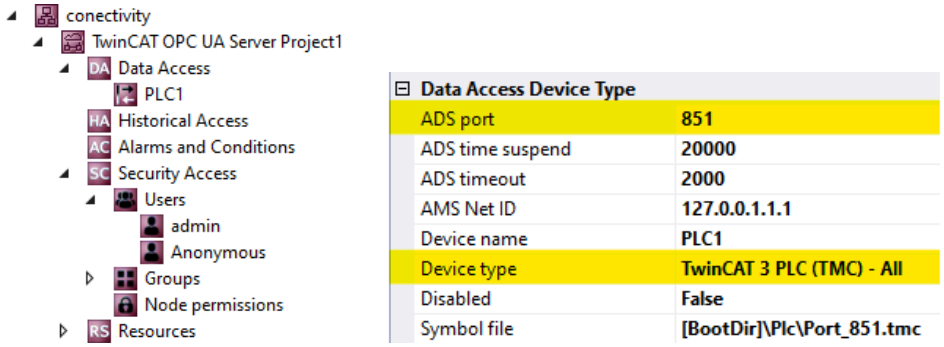
Pro samotné spuštění je potřeba naistalovat všechny položky serveru:

- Configurator
- Server
- Client
- Gateway

Samotné zprovoznění a připojení se k serveru je popsáno níže.

5.5.1 Twincat

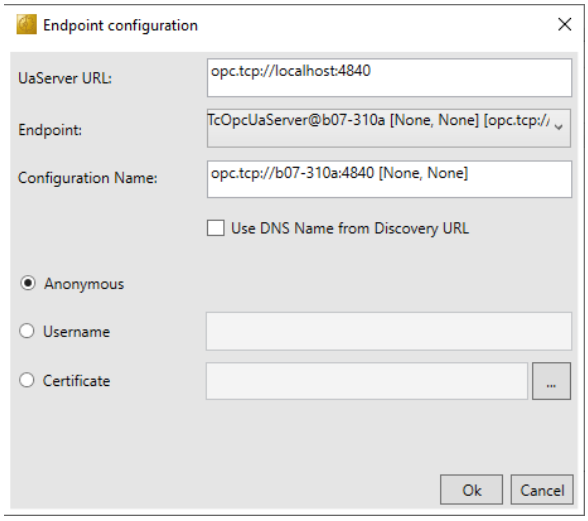
Prvním krok propojení začíná v programu Twincat vytvořením Conectivity projektu, přidáním virtuálního zařízení ve složce Data Acces a jeho konfigurací. Zde se nastavuje primárně ADS port a device type, který rozlišuje, které signály budou z PLC viditelné v OPC serveru. Pro jednoduchost celého PLC mohlo být nastaveno, aby se zobrazovaly všechny proměnné. Nastavené parametry jsou vidět viz. Obrázek 49.



Obrázek 49 a) Conectivity project b) Data Access Device Type

Další krok je vytvoření endpointu neboli serveru, ke kterému se OPC bude následně snažit připojovat. Zde prozatím zvolíme adresu lokálního počítače,, opc.tcp://localhost:4840“ a konfigurace šifrování pro tento případ zvolena None, šifrování a další zabezpečení by pouze zdržovalo při práci a neplnilo by žádný účel, protože pracujeme pouze v lokálním prostředí. Endpoint konfigurace je vidět viz. Obrázek 50.

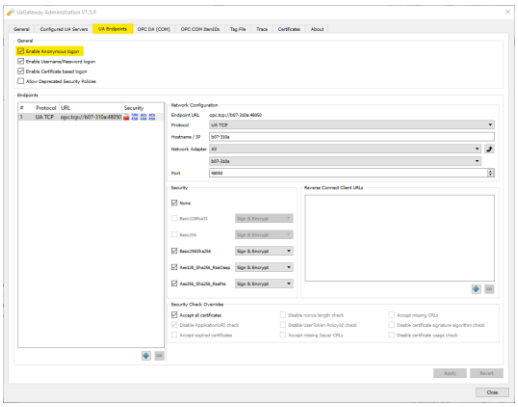
Navíc, současná verze RS neumožnuje v konfiguraci nastavení uživatelského jména a hesla, takže je nutné se vždy přihlašovat jako účet Anonymous. V případě zprovoznění na již reálný hardware je možnost adresu a zabezpečení vždy změnit.



Obrázek 50 Endpoint configuration

5.5.2 OPC UA Gateway

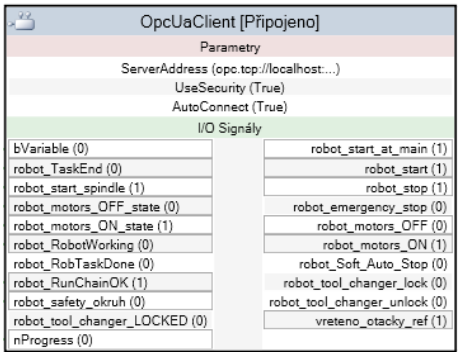
OPC Gateway je nástroj pro administrativu samotného serveru, ale v rámci této práce je využit pouze pro povolení anonymních uživatelů. Provedená změna je zvýrazněna viz. Obrázek 51.



Obrázek 51 OPC UA Gateway

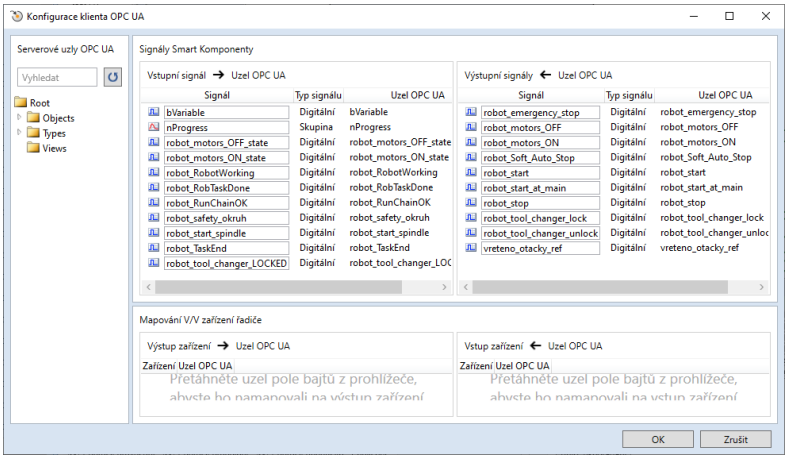
5.5.3 RS

RS od verze 2021 umožnuje připojení k OPC serverům skrze smart componentu v logice stanice. Její možnosti jsou značně limitovány, ale pro tento účel je funkcionalita dostačující. V rámci nastavení se pouze zadá cílová adresa serveru „opc.tcp://localhost:4840“ a povolí se možnost automatického připojování. Díky tomu při každém startu RS tuto akci není nutno dělat manuálně a RS se snaží připojit automaticky samo. Smart komponenta je vyobrazena viz. Obrázek 52. Cele zapojení logiky stanice včetně propojení signálu je vidět viz. Obrázek 37.



Obrázek 52 OPC UA – RS

Po úspěšném připojení už stačí jen nakonfigurovat vstupní a výstupní signály a připojit je ke korespondujícím signálům v rámci RS viz. Obrázek 53.



Obrázek 53 Propojení signálu v RS

6 ZHODNOCENÍ A DISKUZE

Diplomová práce pojednává o digitálním zprovoznění robotizovaného výrobního systému s obráběcí buňkou. Virtuální zprovoznění je sám o sobě náročný a komplexní úkol, ale jeho úspěšným zvládnutím lze dosáhnout snížení počátečních nákladů na implementaci výrobního systému.

V rámci diplomové práce byly splněny všechny požadavky udělené vedoucím práce, ale pro kompletní implementaci obráběcí buňky do již hotového pracoviště by bylo zapotřebí dodělat několik změn, aby celé pracoviště mohlo fungovat autonomně. Jednalo by se především o modifikace držáku na efektory robota pro odložení vřetene a implementace vřetene do rutiny pro výměnu efektoru. V současném stavu musí efektor vyměnit obsluha a manuálně napojit hadici k odsavači prachu.

Druhou vhodnou úpravou by byl návrh odkládací plochy pro různé frézy, aby u robota mohlo být využito automatické výměny nástroje vřetene, a tak obrábět i složitější geometrie.

Další modifikací užitečnou pro budoucí provoz pracoviště je rozšíření vizualizace o další varovné hlášky podrobněji popisující nastalé problémy či stavy na pracovišti, servisní mód a mód pro manuální ovládání celého pracoviště.

Poslední modifikací vhodnou pro přidání před reálnou implementací by bylo vytvoření části programu pro zachycení aktuální pozice v programu RAPID při neočekávaném zastavení obrábění, například při přerušení safety okruhu nebo přehřátí vřetene. Aktuální stav modelu tuto funkci neobsahuje a při nečekaném zastavení by musel dříve prošlé operace projet znovu, což by mělo za následek časové ztráty a mohlo způsobit poškození kvality povrchu obrobku.

7 ZÁVĚR

V první části práce byla rešerše na robotická pracoviště a jejich vybavení, programování robotů, výhody a nevýhody robotického obrábění a software používaný k virtuálnímu zprovoznění. V této části je také popsána bezpečnost u frekvenčních měničů a volba vhodné náhrady za aktuálně používaný měnič pro robotické pracoviště.

V praktické části byl vytvořen model pracoviště v programu Robot Studio za použití komponent získaných buď od vedoucího práce, importovaných z defaultních knihoven, nebo mnou vytvořených v modelovacím softwaru Autodesk Inventor Professional. V rámci RS byl následně vytvořen i program pro robota a upraven tak, aby správně reagoval na řídící PLC.

Hlavní částí byla tvorba programu na řízení celého pracoviště a ten byl vytvořen v programu TwinCAT s využitím nástavby TF6100 OPC UA. V rámci řešení byl vytvořen PLC projekt, Conectivity projekt a vizualizace HMI. Skrze OPC UA byla vytvořena komunikace s RS a vytvořen program spouštějící obrábění a monitorující aktuální stav signálů pracoviště a jeho komponent důležitých pro obrábění. Dále se PLC stará o komunikaci a řízení frekvenčního měniče a zajišťuje jeho odstavení z provozu v případě přerušení safety okruhu. Před tvorbou samotného programu byly veškeré požadované funkce ztvárněny ve vývojových diagramech, které jsou také součástí této práce.

Lze tedy konstatovat, že hlavní cíl práce: „Virtuální zprovoznění robotizovaného obráběcího pracoviště“, byl úspěšně splněn a vznikl tak 3D model pracoviště, řídící PLC program, program robota, vizualizace HMI a video celé simulace přiložené v příloze 1.

8 SEZNAM POUŽITÝCH ZDROJŮ

[1] Digital twins: How they can help scale up industrial robotics AI. In: *Siliconrepublic* [online]. ©2002-2022 [cit. 2022-05-06]. Dostupné z: https://www.siliconrepublic.com/machines/digital-twins-industrial-robotics

[2] The Digital Twin and Real-Time Adaptive Robot Control. In: *Energid* [online]. 2021 [cit. 2022-05-06]. Dostupné z: https://www.energid.com/blog/the-digital-twin-and-real-time-adaptive-robot-control

[3] What Is the Best Way to Program a Robot?. In: *RoboDK* [online]. 2021 [cit. 2022-05-06]. Dostupné z: https://robodk.com/blog/program-robot-tips/

[4] Machining with robots | Robot milling. In: *CNC ROBOTICS LTD* [online]. 2010 [cit. 2022-05-06]. Dostupné z: https://www.cncrobotics.co.uk/news/machining-with-robots/

[5] Can a Robot Outperform a CNC Machine for Robot Machining?. In: *RoboDK* [online]. 2021 [cit. 2022-05-06]. Dostupné z: https://robodk.com/blog/robot-machining-vs-cnc/

[6] Autodesk Inventor. In: *Inventor 3D Blog* [online]. 2019 [cit. 2022-05-07]. Dostupné z: https://www.inventor3dblog.cz/inventor/

[7] Inventor CAM: Integrovaný CAM do Inventoru zjednodušuje celý proces CNC výroby. In: *Arkance Systems CZ* [online]. 2022 [cit. 2022-05-07]. Dostupné z: https://cam.cadstudio.cz/inventor-cam/

[8] Autodesk Fusion 360. In: *Arkance Systems CZ* [online]. 2022 [cit. 2022-05-07]. Dostupné z: https://www.arkance-systems.cz/produkty/cad-cam-pro-strojirenstvi/autodesk-fusion-360

[9] Process Simulate: Manufacturing process verification in powerful 3D environment. In: *SIEMENS* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://www.plm.automation.siemens.com/en_gb/Images/7457_tcm642-80351.pdf

[10] RoboDK: An Offline Programming and 3D Simulation Software for Industrial Robots. In: *Smashingrobotics* [online]. 2021 [cit. 2022-05-06]. Dostupné z: https://www.smashingrobotics.com/robodk-industrial-robot-offline-programming-simulation-software/

[11] Robot Programs. In: *RoboDK* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://robodk.com/doc/en/Robot-Programs.html

[12] Program Off-Line with ABB RobotStudio. In: *RobotWorx* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://www.robots.com/blogs/program-off-line-with-abb-robotstudio

[13] Siemens TIA Portal – jednotné vývojové prostředí pro automatizaci v průmyslu. In: *Automa* [online]. 2016 [cit. 2022-05-06]. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/43212.pdf

[14] TwinCAT 3 – eXtended Automation Technology. In: *Automa* [online]. 2016 [cit. 2022-05-06]. Dostupné z: http://automa.cz/Aton/FileRepository/pdf_articles/53801.pdf

[15] "Ladder" Diagrams. In: *All About Circuits* [online]. [cit. 2022-05-06]. Dostupné z: https://www.allaboutcircuits.com/textbook/digital/chpt-6/ladder-diagrams/

[16] Ladder diagramy – základy. In: *PLC FATEK* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://fatek.seapraha.cz/2008/03/24/zaklladr/

[17] What is a Functional Block Diagram?. In: *Edraw* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://www.edrawsoft.com/article/what-is-functional-block-diagram.html

[18] An Overview of Sequential Function Chart (SFC) Programming. In: *Control Automation* [online]. [cit. 2022-05-06]. Dostupné z: https://control.com/technical-articles/an-overview-of-sequential-function-chart-sfc-programming/

[19] Basics of Structured Text (ST) Programming: Let's discuss the basics of PLC programming utilizing Structured Text. In: *Realpars* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://realpars.com/structured-text/

[20] Virtual Controller. In: *UNDERWOODS* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://www.ued.co.uk/manufacturers/siemens/industrial-software-/virtual-commissioning-for-machine-builders/plcsim-advanced/

[21] Průmyslová komunikace OPC UA - 1.díl - popis protokolu. In: *Automatizace.hw.cz* [online]. [cit. 2022-05-06]. Dostupné z: https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html

[22] PLr and PL (From ISO 13849-1:2006). In: *Safety knowledge* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://www.keyence.eu/ss/products/safetyknowledge/performance/level/

[23] What's the difference between PL and SIL machine safety standards?. In: *MOTION CONTROL TIPS* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://www.motioncontroltips.com/whats-the-difference-between-pl-sil-machine-safety-standards/

[24] SINAMICS Safety Integrated. In: *SIEMENS* [online]. 2022 [cit. 2022-05-06]. Dostupné z: https://new.siemens.com/global/en/products/automation/topic-areas/safety-integrated/factory-automation/offering/sinamics-safety.html

[25] IRC5: Industrial Robot Controller. In: *ABB* [online]. 2022 [cit. 2022-05-17]. Dostupné z: https://search.abb.com/library/Download.aspx?DocumentID=ROB0295EN&LanguageCode=en&DocumentPartId=&Action=Launch

[26] Product Specification: Track Motion: IRBT 6003S; IRBT 4003S. In: *DOKUMEN.TIPS* [online]. [cit. 2022-05-07]. Dostupné z: https://dokumen.tips/documents/abb-irbt-4003.html

[27] QD-1F 4/12 24 I30. In: *Hiteco* [online]. 2020 [cit. 2022-05-07]. Dostupné z: https://www.hiteco.net/en/products/electrospindles-a.t.c.c8190/powertech-210-series-4-kw.9514/qd-1f-4-12-24-i30.100148

[28] Odsavač KH - 3121. In: *Uni-max* [online]. 2022 [cit. 2022-05-07]. Dostupné z: https://www.uni-max.cz/produkty/drevoobrabeni/odsavani-prachu/odsavaci-a-filtracni-jednotky/odsavac-kh-3121

[29] Filtr skládaný pro KH - 3121. In: *Uni-max* [online]. 2022 [cit. 2022-05-07]. Dostupné z: https://www.uni-max.cz/produkty/drevoobrabeni/odsavani-prachu/filtry-a-vaky/filtry-kartusove/filtr-skladany-pro-kh-3121

[30] 134L2281: FC-102P4K0T4E20H1TGXXXXSXXXXALBXCXXXXDX. In: *Danfoss* [online]. [cit. 2022-05-07]. Dostupné z: https://store.danfoss.com/cz/cs/Frekven%C4%8Dn%C3%AD-m%C4%9Bni%C4%8De/N%C3%ADzkonap%C4%9B%C5%A5ov%C3%A9-m%C4%9Bni%C4%8De/VLT%C2%AE-HVAC-Drive-FC-102/FC-102P4K0T4E20H1TGXXXXSXXXXALBXCXXXXDX/p/134L2281

[31] OPC UA and PROFINET. In: *Profinet university* [online]. 2022 [cit. 2022-05-15]. Dostupné z: https://profinetuniversity.com/industrial-automation-ethernet/opc-ua-profinet/

[32] PROFINET Communication Channels. In: *Profinet university* [online]. 2022 [cit. 2022-05-15]. Dostupné z: https://profinetuniversity.com/profinet-basics/profinet-communication-channels/

[33] ABB Robotics - Rapid. In: *Autodesk* [online]. 2022 [cit. 2022-05-07]. Dostupné z: https://cam.autodesk.com/hsmposts?p=abb&_ga=2.20262328.124841764.1593537081-1682455771.1592242493

[34] CP29xx: Multi-touch built-in Control Panel. In: *Beckhoff* [online]. 2022 [cit. 2022-05-07]. Dostupné z: https://www.beckhoff.com/cs-cz/products/ipc/control-panels/cp29xx-cp39xx-multi-touch-control-panels/cp29xx.html

[35] TF6100: TwinCAT 3 OPC UA. In: *Beckhoff* [[online]. 2022 [cit. 2022-05-07]. Dostupné z: https://www.beckhoff.com/cs-cz/products/automation/twincat/tfxxxx-twincat-3-functions/tf6xxx-tc3-connectivity/tf6100.html

[36] Technická data pro IRB 4400. In: *ABB* [online]. 2022 [cit. 2022-05-18]. Dostupné z: https://new.abb.com/products/robotics/cs/prumyslove-roboty/irb-4400/irb-4400-technicka-data

9 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

9.1 Seznam zkratek

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CNC	Computer Numerical Control
CTW	Control Word,
FBD	Function Block Diagram
FSI	Fakulta strojního Inženýrství
GVL	Global Variable List
HMI	Human Machine Interface
I/O	Input/Output
INT	Integer
LD	Leader Diagram
Max	Maximum
Min	Minimum
MRV	Main Reference Value
NC	Numerical Control
OPC UA	Open Platform Communications United Architecture
PFHd	Probability of Dangerous Failure per Hour
PL	Performance Level
PLC	Programmable Logic Controller
PLr	Performance Level Required
RPM	Revolutions Per Minute
RS	Robot studio
SBC	Safe brake Control
SBT	Safe Brake Test
SCF	Sequential Function Chart
SIL	Safety Integrity Levels
SOS	Safe Operating Stop
SS1	Safe Stop 1
SS2	Safe Stop 2
ST	Structured Text
STO	Safe Torque Off
STW	Status Word
SW	Software
TCP	Tool center point
TON	On Delay Timer
ÚVSSR	Ústav Výrobních Strojů Systémů a Robotiky
VUT	Vysoké Učení Technické
VZ	Virtuální Zprovoznění

9.2 Seznam tabulek

Tabulka 1 Performance Level – PFHd 30
Tabulka 2 Minimální požadavky na frekvenční měnič 36

9.3 Seznam rovnic

Rovnice 1 Výpočet MRV 49

9.4 Seznam obrázků

Obrázek 1 Proces online programování [2] 19
Obrázek 2 Proces offline programování [2] 20
Obrázek 3 Ukázka použití robotického obrábění [4] 21
Obrázek 4 Prostředí programu Process Simulate [9] 22
Obrázek 5 Prostředí programu RoboDK [10] 23
Obrázek 6 Prostředí programu TIA portál 24
Obrázek 7 Struktura vývojového prostředí TwinCAT 3 (XAE) [14] 25
Obrázek 8 Ukázka kódu – LD 26
Obrázek 9 Ukázka kódu – FBD 26
Obrázek 10 Ukázka kódu – SCF 27
Obrázek 11 Ukázka kódu – ST 27
Obrázek 12 Komunikace OPC UA 28
Obrázek 13 Performance Level – úrovně 29
Obrázek 14 Safety funkce – STO [24] 31
Obrázek 15 Safety funkce – SS1 [24] 31
Obrázek 16 Safety funkce – SOS [24] 31
Obrázek 17 Safety funkce – SS2 [24] 31
Obrázek 18 Safety funkce – SBC [24] 32
Obrázek 19 Safety funkce – SBT [24] 32
Obrázek 20 Robotické pracoviště ÚVSSR 33
Obrázek 21 Průmyslový robot ABB IRB 4400/ 60 + pracovní obálka [36] 33
Obrázek 22 Řídící systém robotu ABB IRC5 a FlexPendant [25] 34
Obrázek 23 Pojezd ABB IRBT 4003 [26] 34
Obrázek 24 Vřeteno Hiteco QD-1F 4/12 24 I30 NC CB [27] 35
Obrázek 25 Charakteristika výkonu, momentu a frekvence na RPM 35
Obrázek 26 Odsavač KH - 3121 a) Základní konfigurace b) Přídavný filtr [28] [29] 36
Obrázek 27 Frekvenční měnič VLT® HVAC Drive FC 102 - 134L2281 [30] 37
Obrázek 28 Hierarchie OPC UA a PROFINET a)Stará b)Nová [31] 38
Obrázek 29 Výsledný model 39
Obrázek 30 CAM – Setup 39
Obrázek 31 CAM operace – a) 3D adaptive clearing b) 2D face c) 3D pocket d) všechny operace 40
Obrázek 32 Postprocessor ABB – Parameters 40
Obrázek 33 Setup Sheet 41
Obrázek 34 Základ 3D modelu – robot, pojezd, buňka 42
Obrázek 35 3D model – odsavač + polotovar 42
Obrázek 36 3D model – Kompletní 43

Obrázek 37 Logika Stanice 44
Obrázek 38 GVL – Seznam proměnných 45
Obrázek 39 Funkční stavy vřetene 46
Obrázek 40 Vývojový diagram – vřeteno 46
Obrázek 41 Vývojový diagram – Robot 47
Obrázek 42 Dekódování a) CTW b) STW 49
Obrázek 43 Vývojový diagram – Měnič 50
Obrázek 44 CTW a STW pro INIT, START a STOP 50
Obrázek 45 HMI – PROCES 51
Obrázek 46 HMI – Vřeteno 51
Obrázek 47 HMI – Frekvenční měnič 52
Obrázek 48 HMI – Robot 52
Obrázek 49 a) Conectivity project b) Data Access Device Type 53
Obrázek 50 Endpoint configuration 53
Obrázek 51 OPC UA Gateway 54
Obrázek 52 OPC UA – RS 54
Obrázek 53 Propojení signálu v RS 54

10 SEZNAM PŘÍLOH

Příloha 1: PLC program

Příloha 2: Setup Sheet

Příloha 3: 2022_Priloha3_Valach_Tomas_200069.mp4

PŘÍLOHA 1 – PLC PROGRAM

```
1   {attribute 'qualified_only'}
2   VAR_GLOBAL
3       {attribute 'OPC.UA.DA' := '1'}
4
5       bVariable                           : BOOL ;          //testovací proměná
    pro ověření spojení s RS
6       nState_automat                      : INT ;           //pozice kodu v
    hlavním automatu
7   //vřeteno
8       vreteno_S1                          : BOOL := TRUE ;  //tool corectly
    locked
9       vreteno_S2                          : BOOL := FALSE ; //tool holder open
10      vreteno_S3                          : BOOL ;          //spindle stopped
11      vreteno_motor_thermal_protection    : BOOL ;          //motor overheating
12      vreteno_TC_BUTTON                   : BOOL ;          //manual tool change
    button
13      vreteno_vzduchovy_ventil            : BOOL ;          //1-unlock 0-lock
14      vreteno_otacky_ref                  : BOOL ;
15  //měnič
16      menic_CTW                           : WORD ;          //control word
17      menic_CTW_SPEED                     : WORD := 16#4000 ;   //control word
    speed -100% -16384; 0% 0; 100% 16384;
18      menic_STW                           : WORD ;          //status word
19  //robot výstupy
20      robot_start_spindle                 : BOOL ;          //robot chce zapnout
    vřeteno
21      robot_Soft_Auto_Stop                : BOOL ;          //soft stop
22      robot_tool_changer_lock             : BOOL ;          //zamknout hlavu
23      robot_tool_changer_unlock           : BOOL ;          //odemknout hlavu
24      robot_emergency_stop                : BOOL ;          //emergency stop
25      robot_start                         : BOOL ;          //spust robota
26      robot_start_at_main                 : BOOL ;          //spustit robota v
    MAIN programu
27      robot_stop                          : BOOL ;          //zastavit úlohu
    robota
28      robot_motors_ON                     : BOOL ;          //zapnout motory
29      robot_motors_OFF                    : BOOL ;          //vypnout motory
30  //robot výstupy
31      robot_motors_ON_state               : BOOL ;          //ON stav motorů
32      robot_tool_changer_LOCKED           : BOOL ;          //Shunk locked
33      robot_motors_OFF_state              : BOOL ;          //OFF stav motorů
34      robot_RobotWorking                  : BOOL ;          //robot pracuje
35      robot_RobTaskDone                   : BOOL ;          //informace o
    dokončení úlohy
36      robot_RunChainOK                    : BOOL := TRUE ;  //robot pripraven k
    provozu
37      robot_TaskEnd                       : BOOL ;          //informace o
    dokončení úlohy
38      robot_safety_okruh                  : BOOL := FALSE ; //bool proměná ze
    safety okruhu bunky
39      nProgress                           : INT ;           //INTEAGER proměná
    pro sledování progresu obrábění
40  END_VAR
41
42
```

POU: MAIN

```
1   PROGRAM MAIN
2   VAR
3   END_VAR
4
```

```
1   //inicializace poprogramů
2       POU_stavy ( ) ;
3       POU_vreteno ( ) ;
4       POU_menic ( ) ;
5       POU_simulace ( ) ;
6
```

```
PROGRAM POU_stavy
VAR
    b_inv_obsluha       : BOOL := TRUE ;  //POPUP obsluha - viditelnost
    b_inv_workobject    : BOOL := TRUE ;  //POPUP workobject - viditelnost
    b_inv_nastroj       : BOOL := TRUE ;  //POPUP nastroj - viditelnost
    b_inv_hotovo        : BOOL := TRUE ;  //POPUP hotovo - viditelnost
    b_inv_safety        : BOOL := TRUE ;  //POPUP safety - viditelnost
    b_inv_stop          : BOOL := TRUE ;  //POPUP stop - viditelnost
    b_start_operaci     : BOOL := FALSE ; //HMI start operace

    t1                  : ton ;           //zpoždění od inicializace měniče
END_VAR
```

```
//emergency stop //pokud je přerušen okruh zastav a upozorni obsluhu
IF   gvl . robot_safety_okruh  OR  gvl . robot_emergency_stop  THEN
    gvl . nState_automat := 100 ;
    b_inv_safety := FALSE ;
END_IF

CASE  gvl . nState_automat  OF
    0 :  //POPUP - obsluhovat může pouze školená obsluha
        b_inv_hotovo := TRUE ;
        b_inv_stop := FALSE ;
        b_inv_stop := TRUE ;
        b_inv_obsluha := FALSE ;
    10 :     //POPUP - nastavený správný workobject
        b_inv_obsluha := TRUE ;
        b_inv_workobject := FALSE ;
    20 :     //POPUP - správný nástroj
        b_inv_workobject := TRUE ;
        b_inv_nastroj := FALSE ;
    30 :     //zapnu motory
        GVL . robot_emergency_stop := FALSE ;
        b_inv_nastroj := TRUE ;
        b_inv_safety := TRUE ;
        b_inv_stop := TRUE ;
        gvl . robot_motors_ON := TRUE ;

        IF  gvl . robot_motors_ON_state = TRUE  THEN
            gvl . nState_automat := 31 ;
        END_IF
    31 :     // inicializace pracoviště
        gvl . robot_start := TRUE ;    //nahoď robota
        gvl . menic_CTW := 16#047F ;  //Init CTW
        t1 ( IN := gvl . robot_start  ,  PT :=  T#1S ,  Q =>  ,  ET =>  ) ;
        IF  gvl . robot_RunChainOK := TRUE  AND  gvl . menic_STW . 2  AND  t1 . Q
THEN
            gvl . nState_automat := 32 ;
        END_IF
    32 :     //start at main        ; HMI tlačítko spustí operaci
        IF  b_start_operaci  THEN
            gvl . robot_stop := FALSE ;
            gvl . robot_start_at_main := TRUE ;
            gvl . nState_automat := 33 ;
```

```
42              END_IF
43          33 :    //robot běži
44
45              //RS spusti příkaz start_spindle
46                  //zapne se měnič
47                      //PLC oznámi RS že může obrábět
48
49              IF  gvl . robot_TaskEnd  THEN
50                  gvl . robot_stop := TRUE ;          //vypne robota
51                  gvl . menic_CTW := 16#423 ;         //zastavi měnič
52                  gvl . nState_automat := 40 ;
53              END_IF
54          40 :    //POPUP - finished odeberte díl
55              b_inv_hotovo := FALSE ;
56          99 :    //softstop teplotni pojiskta vřetene
57              gvl . menic_CTW := 16#427 ;    //ramp stop
58              gvl . robot_Stop := TRUE ;  //soft stop
59              b_inv_stop := FALSE ;
60          100 : //hard stop POPUP - safety error, odstrante závadu a resetujte
        program
61              gvl . menic_CTW := 16#423 ;
62              gvl . robot_emergency_stop := FALSE ;
63              b_inv_safety := FALSE ;
64      END_CASE
65
66
```

```
PROGRAM POU_vreteno
VAR

    S1                  : BOOL ;
    S2                  : BOOL ;
    S3                  : BOOL ;
    S5                  : BOOL ;

    HMI_stav_OK         : BOOL ;
    HMI_stav_RELEASED   : BOOL ;
    HMI_stav_NO_TOOL    : BOOL ;

    TON_otacky1     : TON ;
    TON_otacky2     : TON ;
    TON_stav_1      : TON ;
    TON_stav_2      : TON ;
    TON_stav_3      : TON ;
    TON_rozjezd     : TON ;

    b_vreteno_stoji :  BOOL ;
    HMI_LOCK : BOOL ;
    HMI_UNLOCK : BOOL ;
END_VAR

```

```
//shutdown na základě přehřátí
    IF gvl.vreteno_motor_thermal_protection THEN
        gvl.nState_automat := 99 ;
    END_IF
//zkracení názvů
    S1 := gvl.vreteno_S1 ;
    S2 := gvl.vreteno_S2 ;
    S3 := gvl.vreteno_S3 ;
    S5 := gvl.vreteno_motor_thermal_protection ;
//stav vřetene
    //tool released
        TON_stav_1 (    IN := NOT S1 ,
                    PT := T#100MS ,
                    Q => ,
                    ET => ) ;
            TON_stav_2 (    IN := TON_stav_1.Q AND S2 ,
                        PT := T#200MS ,
                        Q => HMI_stav_RELEASED ,
                        ET => ) ;
    //tool lock corectly
        TON_stav_3 (    IN := NOT S2 AND S1 ,
                    PT := T#600MS ,
                    Q => HMI_stav_OK ,
                    ET => ) ;
    //tool not locked corectly - tool not present
        IF NOT S1 AND NOT S2 THEN
            HMI_stav_NO_TOOL := TRUE ;
        ELSE
            HMI_stav_NO_TOOL := FALSE ;
        END_IF
//pokud se 5s nezmění signál otáče => vřeteno stojí
```

```
32          TON_otacky1 ( IN :=  S3  ,
33                        PT :=  T#5S ,
34                        Q =>  ,
35                        ET =>  ) ;
36          TON_otacky2 ( IN :=  NOT  S3 ,
37                        PT := T#5S  ,
38                        Q =>  ,
39                        ET =>  ) ;
40          IF  TON_otacky1 . Q  OR  TON_otacky2 . Q  THEN
41              b_vreteno_stoji := TRUE ;
42          ELSE
43              b_vreteno_stoji := FALSE ;
44          END_IF
45
46          TON_rozjezd ( IN := gvl . robot_start_spindle  ,  PT :=  T#2S ,  Q =>  gvl .
        vreteno_otacky_ref ,  ET =>  ) ;
47
48      //lock tool
49          IF  HMI_LOCK     AND  NOT  HMI_UNLOCK  THEN
50              gvl . vreteno_vzduchovy_ventil    :=  TRUE ;
51              HMI_LOCK := FALSE ;
52
53              //simulace odezvy reálného vřetene
54              gvl . vreteno_S1 := TRUE ;
55              gvl . vreteno_S2 := FALSE ;
56          END_IF
57      //unlock tool
58          IF  ( gvl . vreteno_TC_BUTTON
59              OR  ( HMI_UNLOCK  AND  NOT  HMI_LOCK ) )
60              AND  b_vreteno_stoji
61              AND  ( HMI_stav_OK  OR  HMI_stav_NO_TOOL )  THEN
62                  gvl . vreteno_vzduchovy_ventil := FALSE ;
63                  HMI_UNLOCK := FALSE ;
64
65              //simulace odezvy reálného vřetene
66              gvl . vreteno_S1 := FALSE ;
67              gvl . vreteno_S2 := TRUE ;
68          END_IF
69
70
```

```
PROGRAM POU_menic
VAR
    STW : WORD ;
    nState : INT := 1 ;

    HMI_MENIC_OK :  BOOL ;
END_VAR
```

```
(*
       priorita:
             ***        :=        1
             **               :=        2
             *              :=        3
             -              :=        4
*)
STW := gvl . menic_STW ;                        //zapsání do lokální proměné
(zkrátí to kód)

IF  gvl . robot_start_spindle  THEN             //po příkazu RS roztočí
vřeteno/měnič
    gvl . menic_CTW := 16#147F ;
END_IF

CASE  nState  OF
    0 : //Init
        IF  NOT ( STW . 0
            AND  STW . 1
            AND  STW . 2 )
            AND  STW . 6  THEN
            nState := 1 ;
        END_IF
    1 : //S1: Switching ON inhibited
        IF  NOT  STW . 0
            AND  STW . 1
            AND  STW . 2  THEN                  //-
            nState := 2 ;
        END_IF
    2 : //S2: Ready for switching ON2
        IF  NOT ( STW . 1
            OR  STW . 2 )  THEN                 //*
            nState := 1 ;
        END_IF

        IF  STW . 0  THEN                   //-
            nState := 3 ;
        END_IF
    3 : //S3: Switched ON
        stw . 3 := true ;
        IF  NOT ( STW . 1
            OR  STW . 2 )  THEN                 //**
            nState := 1 ;
        END_IF

        IF  NOT  STW . 0  THEN                  //*
            nState := 2 ;
```

```
46              END_IF
47
48              IF  STW . 3  THEN                   //-
49                  nState := 4 ;
50              END_IF
51          4 : //S4: Operation
52              IF  NOT  STW . 1  THEN                  //*** coast stop
53                  nState := 1 ;
54              END_IF
55
56              IF  NOT  STW . 2  THEN                  //** quick stop
57                  gvl . menic_CTW . 4 := TRUE ;
58                  nState := 1 ;
59              END_IF
60
61              IF  NOT  STW . 0  THEN                  //* OFF
62                  nState := 5 ;
63              END_IF
64
65              IF  NOT  STW . 3  THEN                  //-
66                  nState := 3 ;
67              END_IF
68          5 : //S5: Switching OFF
69              gvl . menic_CTW . 4 := TRUE ;           //ramp stop
70
71              IF  NOT  STW . 2  THEN                  //** quick stop
72                  gvl . menic_CTW . 6 := FALSE ;
73                  nState := 1 ;
74              END_IF
75
76              nState := 5 ;                           //* pulses disabled
77
78              IF  STW . 0  THEN                   //ON
79                  nState := 4 ;
80              END_IF
81      END_CASE
82          //signalizace HMI že měnič a) je připraven na operaci b) pracuje
83      IF  nState = 3  OR  nState = 4  THEN
84          HMI_MENIC_OK := TRUE ;
85      ELSE
86          HMI_MENIC_OK := FALSE ;
87      END_IF
88
```

```
PROGRAM POU_simulace
VAR
    n_reset_6           : INT := 0 ;
    blikac_vreteno_1    : ton ;
    blikac_vreteno_2    : ton ;
END_VAR
```

```
//simulace chování vřetene
    //pokud se má točit vřeteno.. bliká senzor S3
    IF gvl.robot_start_spindle THEN
    blikac_vreteno_1 ( IN := gvl.robot_start_spindle
                          AND NOT blikac_vreteno_2.Q ,
                       PT := T#500MS ,
                       Q => ,
                       ET => ) ;
    blikac_vreteno_2 ( IN := blikac_vreteno_1.Q ,
                       PT := T#500MS ,
                       Q => gvl.vreteno_S3 ,
                       ET => ) ;
    gvl.vreteno_S3 := blikac_vreteno_1.Q ;
    END_IF
//simulace odezvy měniče STW na zadané CTW
    //init
    IF gvl.menic_CTW = 1151 THEN
        gvl.menic_STW := 1598 ;
    END_IF
    //start
    IF gvl.menic_CTW = 5247 THEN
        gvl.menic_STW := 3903 ;
    END_IF
    //stop
    IF gvl.menic_CTW = 35967 THEN
        gvl.menic_STW := 5676 ;
    END_IF

//simulace signálů vřetene
    //lokální proměné
    //umístěno přímo v POU_vřeteno
```

PŘÍLOHA 2 – SETUP SHEET

Setup Sheet for Program 1001

Job Description: Setup3
Document Path: test v6

Setup

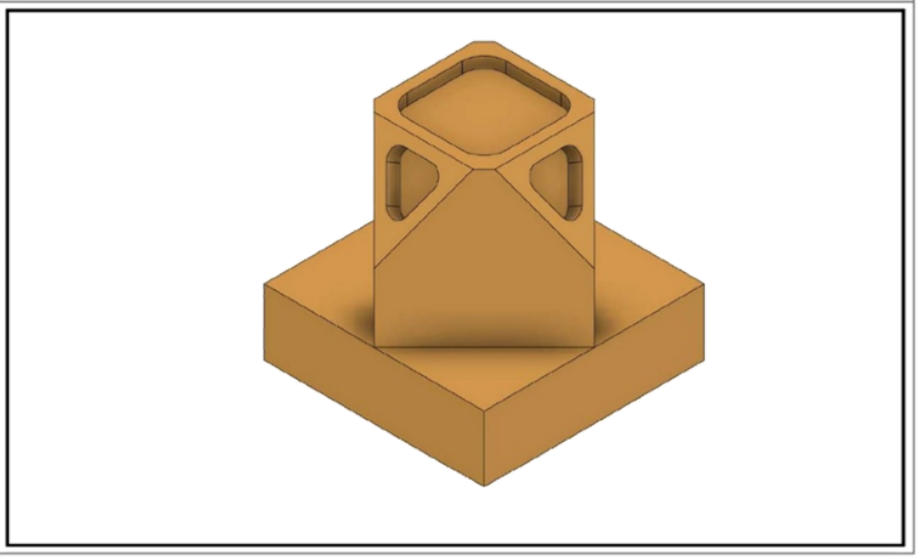
WCS: #1

Stock:
DX: 100mm
DY: 100mm
DZ: 100mm

Part:
DX: 100mm
DY: 100mm
DZ: 100mm

Stock Lower in WCS #1:
X: 0mm
Y: 0mm
Z: -100mm

Stock Upper in WCS #1:
X: 100mm
Y: 100mm
Z: 0mm



Total

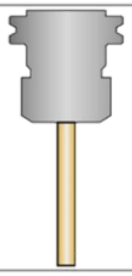
Number Of Operations: 11
Number Of Tools: 1
Tools: **T2**
Maximum Feedrate: 5000.74mm/min
Maximum Spindle Speed: 8000rpm
Cutting Distance: 8251.61mm
Rapid Distance: 5404.55mm
Estimated Cycle Time: 3m:26s

Tools

T2 D2 L2
Type: flat end mill
Diameter: 10mm
Length: 80mm
Flutes: 3
Description: 10mm Flat Endmill

Maximum Feed: 5000.74mm/min
Maximum Spindle Speed: 8000rpm
Cutting Distance: 8251.61mm
Rapid Distance: 5404.55mm
Estimated Cycle Time: 3m:11s (92.7%)

Holder: Maritool CAT40-ER32-2.35
Vendor: Maritool
Product: CAT40-ER32-2.35

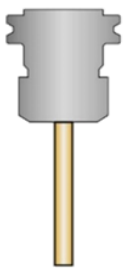


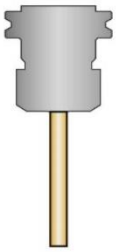
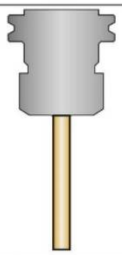
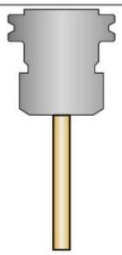
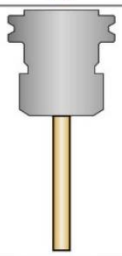
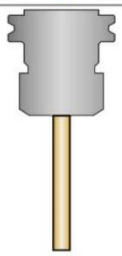
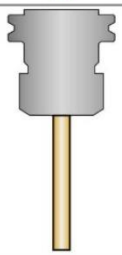
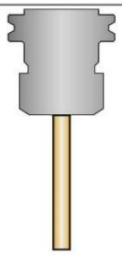
Operations

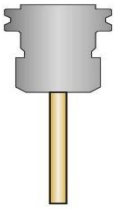
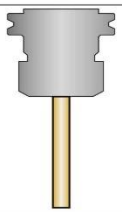
Operation 1/11
Description: Adaptive2
Strategy: Adaptive
WCS: #1
Tolerance: 0.2mm
Stock to Leave: 0mm
Maximum stepdown: 20mm
Optimal load: 9mm
Load deviation: 0.5mm

Maximum Spindle Speed: 8000rpm
Maximum Feedrate: 5000.74mm/min
Cutting Distance: 4282.43mm
Rapid Distance: 4722.94mm
Estimated Cycle Time: 1m:48s (52.5%)
Coolant: Flood

T2 D2 L2
Type: flat end mill
Diameter: 10mm
Length: 80mm
Flutes: 3
Description: 10mm Flat Endmill



Operation 2/11 DESCRIPTION: Flat1 WCS: #1 TOLERANCE: 0.01mm STOCK TO LEAVE: 0mm MAXIMUM STEPOVER: 7mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 254.7mm RAPID DISTANCE: 65.97mm ESTIMATED CYCLE TIME: 6s (2.7%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	
Operation 3/11 DESCRIPTION: Pocket1 STRATEGY: Pocket WCS: #1 TOLERANCE: 0.1mm STOCK TO LEAVE: 0mm MAXIMUM STEPDOWN: 5mm MAXIMUM STEPOVER: 6mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 441.01mm RAPID DISTANCE: 68.72mm ESTIMATED CYCLE TIME: 9s (4.4%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	
Operation 4/11 DESCRIPTION: Flat2 WCS: #1 TOLERANCE: 0.01mm STOCK TO LEAVE: 0mm MAXIMUM STEPOVER: 7mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 253.15mm RAPID DISTANCE: 65.97mm ESTIMATED CYCLE TIME: 6s (2.7%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	
Operation 5/11 DESCRIPTION: Pocket2 STRATEGY: Pocket WCS: #1 TOLERANCE: 0.1mm STOCK TO LEAVE: 0mm MAXIMUM STEPDOWN: 1mm MAXIMUM STEPOVER: 6mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 584.67mm RAPID DISTANCE: 72.92mm ESTIMATED CYCLE TIME: 12s (5.7%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	
Operation 6/11 DESCRIPTION: Flat3 WCS: #1 TOLERANCE: 0.01mm STOCK TO LEAVE: 0mm MAXIMUM STEPOVER: 7mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 253.15mm RAPID DISTANCE: 65.97mm ESTIMATED CYCLE TIME: 6s (2.7%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	
Operation 7/11 DESCRIPTION: Pocket3 STRATEGY: Pocket WCS: #1 TOLERANCE: 0.1mm STOCK TO LEAVE: 0mm MAXIMUM STEPDOWN: 1mm MAXIMUM STEPOVER: 6mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 587.52mm RAPID DISTANCE: 72.92mm ESTIMATED CYCLE TIME: 12s (5.8%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	
Operation 8/11 DESCRIPTION: Flat4 WCS: #1 TOLERANCE: 0.01mm STOCK TO LEAVE: 0mm MAXIMUM STEPOVER: 7mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 253.15mm RAPID DISTANCE: 65.97mm ESTIMATED CYCLE TIME: 6s (2.7%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	

Operation 9/11 DESCRIPTION: Pocket4 STRATEGY: Pocket WCS: #1 TOLERANCE: 0.1mm STOCK TO LEAVE: 0mm MAXIMUM STEPDOWN: 1mm MAXIMUM STEPOVER: 6mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 582.37mm RAPID DISTANCE: 72.92mm ESTIMATED CYCLE TIME: 12s (5.7%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	
Operation 10/11 DESCRIPTION: Face1 STRATEGY: Facing WCS: #1 TOLERANCE: 0.01mm MAXIMUM STEPOVER: 9.5mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 299.91mm RAPID DISTANCE: 101.23mm ESTIMATED CYCLE TIME: 7s (3.3%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	
Operation 11/11 DESCRIPTION: 2D Pocket1 STRATEGY: Pocket 2D WCS: #1 TOLERANCE: 0.1mm STOCK TO LEAVE: 0mm MAXIMUM STEPOVER: 9.5mm	MAXIMUM SPINDLE SPEED: 8000rpm MAXIMUM FEEDRATE: 3205.739mm/min CUTTING DISTANCE: 459.53mm RAPID DISTANCE: 29mm ESTIMATED CYCLE TIME: 9s (4.4%) COOLANT: Flood	**T2** D2 L2 TYPE: flat end mill DIAMETER: 10mm LENGTH: 80mm FLUTES: 3 DESCRIPTION: 10mm Flat Endmill	

Generated by Fusion 360 CAM 2.0.12888 Thursday, May 12, 2022 15:12:26