



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 ROBOTIZOVANÉ VÝROBNÍ BUŇKY

VIRTUAL COMMISSIONING OF THE ROBOTIC PRODUCTION CELL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Bařka

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jakub Brařina

BRNO 2021

Zadaní diplomové práce

Ústav:	Ústav výrobních strojů, systémů a robotiky
Student:	Bc. Tomáš Bařka
Studijní program:	Strojní inženýrství
Studijní obor:	Výrobní stroje, systémy a roboty
Vedoucí práce:	Ing. Jakub Brařina
Akademický rok:	2020/21

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 robotizované výrobní buňky

Stručná charakteristika problematiky úkolu:

V rámci řešení této diplomové práce je student seznámen s problematikou virtuálního zprovoznění robotických 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 PLC programu.
Tvorba HMI.
Virtuální zprovoznění zadané výrobní buňky.
Vyhodnocení dosažených výsledků.
Doporučení pro praxi.

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 978-8-2144-828-5.

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 2020/21

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Diplomová práce se zabývá virtuálním zprovozněním robotizovaného pracoviště určeného ke gravírování deskových materiálů a jejich následného balení. V teoretické části jsou shrnuty poznatky přispívající k rozvoji virtuálního zprovoznění a popsána zařízení nacházející se v zadané robotizované buňce. V praktické části je následně přiblížen postup sestavení simulačního modelu v softwaru Process Simulate, tvorba PLC programu pomocí vývojového prostředí TIA Portal a v neposlední řadě tvorba vizualizace HMI panelu. Na závěr bylo provedeno samotné virtuální zprovoznění pracoviště následované dodatečnou úpravou a validací robotického programu.

ABSTRACT

The diploma thesis deals with the virtual commissioning of robotic workplace designed to engrave board materials and their subsequent packaging. The summary of the knowledge that contributes to the development of the virtual commissioning as well as description of each component of the given robotic cell, are described in the theoretical part. In the practical part are described procedures such as assembling of the simulation model in Process Simulate software, creating the PLC program in integrated development environment TIA Portal or creating visualization for HMI panel. In the end, the actual commissioning of the workplace was performed, followed by additional modification and validation of the robotic program.

KLÍČOVÁ SLOVA

Virtuální zprovoznění, simulace robotického pracoviště, PLC programování, Siemens, Process Simulate, PLCsim Advanced, TIA Portal, průmyslové roboty ABB

KEYWORDS

Virtual commissioning, simulation of a robotic workplace, PLC programming, Siemens, Process Simulate, PLCsim Advanced, TIA Portal, ABB industrial robots

BIBLIOGRAFICKÁ CITACE

BAŤKA, Tomáš. *Virtuální zprovoznění robotizované výrobní buňky*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/132993>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Ing. Jakub Bražina.

PODĚKOVÁNÍ

Tímto bych rád poděkoval Ing. Jakubu Bražinovi za cenné rady a trpělivost během vypracovávání této závěrečné práce. Také bych rád poděkoval své rodině za velkou podporu během celého 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 18.5.2021

.....

Tomáš Bařka

OBSAH

1	ÚVOD	15
2	MOTIVACE	17
3	PŘEHLED SOUČASNÉHO STAVU POZNÁNÍ	19
3.1.	Průmysl 4.0.....	19
3.2.	Robotizace	20
3.2.	Virtuální zprovoznění.....	20
3.1.1	Porovnání frame-worku tradičního a virtuálního zprovoznění.....	20
3.1.2	Exponenciální růst nákladů	21
3.1.3	HIL vs. SIL	22
3.2	Simulační programy	22
3.2.1	Simulační softwary výrobců robotů	24
3.2.2	Univerzální simulační software	25
3.2.3	PLC	27
3.2.4	Komunikační rozhraní pro virtuální zprovoznění	28
3.3	Analýza zadaného pracoviště.....	30
3.3.1	Průmyslové roboty ABB IRB 1200	31
3.3.2	Lineární pojezd Güdel TMF-1	33
3.3.3	Pneumatická sponkovačka BeA Autotec 246 s doplňovací stanicí	33
3.3.4	Pneumatické uchopovací prvky SMC	34
3.3.5	CNC router	35
3.3.6	Pásový dopravník Alutec KK	36
3.3.7	Vertikální zásobník s přechycením.....	36
3.4	Funkční bezpečnost.....	37
3.4.1	Hierarchie norem EU	37
3.4.2	Úroveň zabezpečení.....	38
3.4.3	Normy pro robotizovaná pracoviště	38
3.4.4	Bezpečnostní prvky v rámci zadané robotizované buňky	38
4	PRAKTICKÁ ČÁST	41
4.1	Tvorba simulačního modelu	41
4.1.1	Nastavení pohybů (kinematika).....	42
4.1.2	Nastavení dat pro tvorbu drah robotů	42
4.1.3	Tvorba robotických operací	45
4.1.4	Materiálový tok.....	48
4.1.5	Signály robotu.....	48
4.1.6	Senzory	49
4.1.7	Dopravníky	50
4.1.8	Logické bloky	50
4.2	Tvorba PLC programu	52
4.2.1	OB Main – Hlavní rutina	52
4.2.2	FB Robot Control – řízení robotu z PLC.....	52
4.2.3	FB Cell Control – řízení robotizované buňky	54
4.2.4	FB Program Flow – sekvence výrobního programu.....	55
4.2.5	FB Reloading Station – Doplňovací stanice.....	58
4.2.6	FailSafe PLC program	59
4.3	Tvorba HMI	62
4.3.1	Rozložení prvků vizualizace.....	62

4.3.2	Ovládací prvky	63
4.3.3	Automatické otvírání vyskakovacích oken přes Visual Basic skript	63
4.4	Virtuální zprovoznění	65
4.5	Úpravy a ověření robotických programů v RobotStudios	67
5	ZHODNOCENÍ A DISKUZE	71
6	ZÁVĚR.....	73
7	SEZNAM POUŽITÝCH ZDROJŮ	75
8	SEZNAM ZKRATEK A OBRÁZKŮ	79
8.1	Seznam zkratk.....	79
8.2	Seznam obrázků.....	79

1 ÚVOD

V posledních letech se výrobní podniky potýkají s nedostatkem personálu. Tento jev ještě více zdůraznila současná pandemická krize, kvůli které se zrychlil rostoucí trend automatizace a robotizace. Ty právě snižují závislost výrobních systémů na lidských zdrojích a dokážou tím zajistit kontinuitu výroby i při nečekaných výpadcích zaměstnanců. Další velkou motivací k automatizaci je zvýšení efektivity a udržení konkurenceschopnosti, což umožňuje firmám zachovat výrobu ve své oblasti působení bez nutnosti přesouvat ji do lokalit s levnou pracovní silou.

Ovšem proces kompletní integrace takových systémů je často zdlouhavý a ekonomicky náročný, což právě ulehčuje moderní přístup s využitím virtuálního zprovoznění. Díky němu je možné paralelně již při výrobě nebo ještě před jejím započítím otestovat simulaci mechanických 3D modelů a tak včas odhalit chyby v konstrukci. A hlavně odladit řídicí programy ještě před zhotovením reálného modelu, což značně zkracuje dobu a snižuje náklady potřebné na realizaci.

Virtuální zprovoznění nepomáhá pouze při vývoji automatizovaného pracoviště, ale také během jeho celého životního cyklu. Na vzniklém simulačním modelu je možné testovat a optimalizovat nové výrobní postupy bez nutnosti odstávky reálného provozu, což zkracuje vedlejší výrobní časy, a tak snižuje celkové výrobní náklady.

Hlavním cílem této diplomové práce je vytvořit simulační model robotizované buňky na výrobu gravírovaných deskových materiálů. Celá buňka bude virtuálně zprovozněna včetně návrhu ovládacího HMI panelu. Po zprovoznění proběhne odladění PLC programu.

2 MOTIVACE

Hlavní motivací pro výběr diplomové práce s tématem virtuálního zprovoznění bylo prohloubit své znalosti a zkušenosti v oblasti PLC programování a programování průmyslových robotů, protože bych se rád touto tematikou zabýval i v budoucí praxi. Právě díky napojení simulačního modelu na řídicí PLC program (virtuální zprovoznění) má student jedinečnou možnost mít zpětnou vazbu komplexního celku, jakým robotizovaná buňka je, a pochopit tak vazby mezi jednotlivými komponenty řízení.

Navíc se techniky oživování i provozování virtualizovaných modelů stále více rozvíjejí a stávají oblíbenějšími nejen mezi integrátory výrobních systémů. Tudiž v nich osobně vidím velký potenciál.

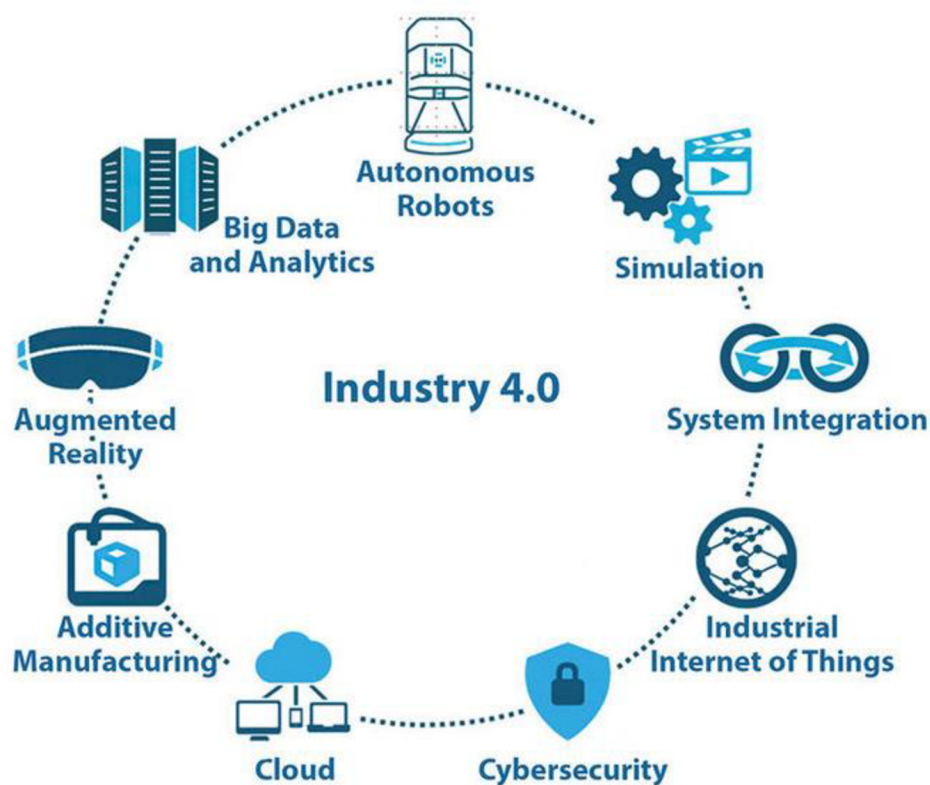
V neposlední řadě bylo motivací k napsání této závěrečné práce prostudování možných funkčních modelů virtuálního zprovoznění s použitím různých softwarů a obeznámení se tak s ekonomicky dostupnějšími variantami softwarových balíčků.

3 PŘEHLED SOUČASNÉHO STAVU POZNÁNÍ

3.1. Průmysl 4.0

V poslední době je v oblasti výroby často skloňován termín průmysl 4.0, kterým je označována 4. průmyslová revoluce. Tento termín byl poprvé použit v roce 2011 na veletrhu v německém Hannoveru a vznikl na popud německé vlády při analýze dopadu moderních technologií na průmysl [1]. Průmysl 4.0 se od svého prvního uvedení postupným vývojem formoval až se jeho definice ustálila na 9 základních pilířích (viz Obr. 1), kterými jsou: systémová integrace, autonomní roboty, internet věcí (IoT), Big Data, Cloud computing, 3D tisk, rozšířená realita (AR), simulace a kybernetická bezpečnost. Jeho hlavním cílem je dosáhnout tzv. chytré výroby (angl. Smart factory), která právě díky výše zmíněným pilířům této filozofie umožňuje propojit všechny jednotlivé agendy podniku, jako je např. logistika, správa budov, engineering, plánování, optimalizace atd. To umožňuje řídit celou firmu na základě aktuálních digitálních dat a dělat rychlejší a také přesnější rozhodnutí. [2]

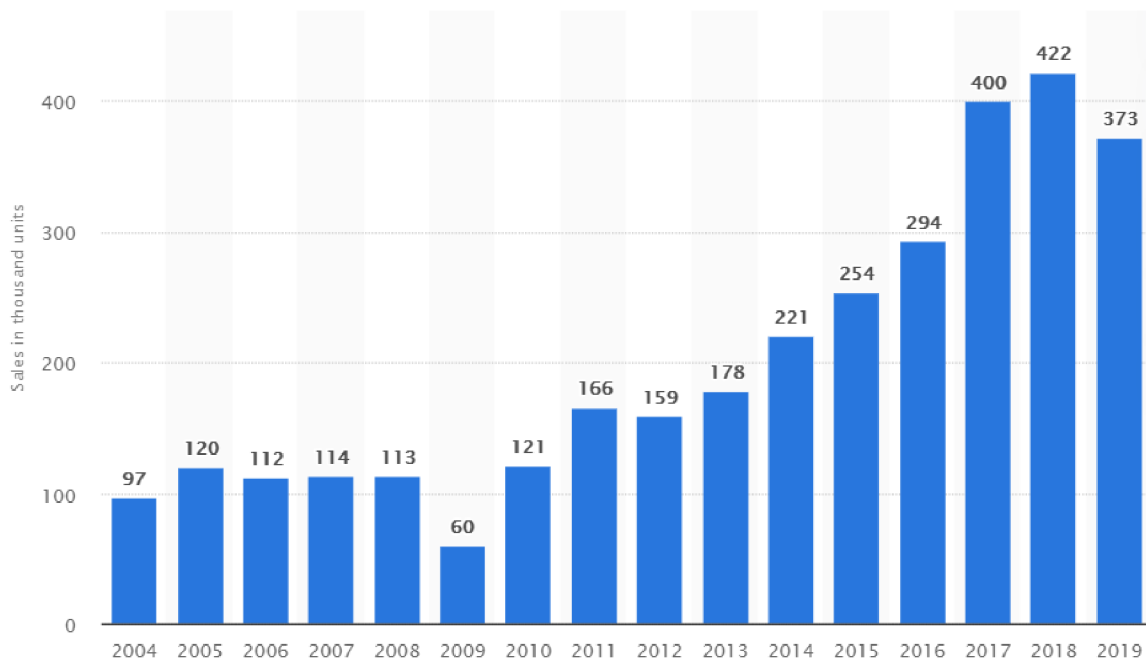
Chytrá výroba se tedy mimo jiné skládá z virtuálních modelů výrobních systémů, které budou sbírat data ze strojů osazených chytrými senzory. Tím bude souběžně s reálným výrobním systémem existovat jeho digitální kopie, která díky aktuálním datům ze senzorů bude věrohodně simulovat jeho chování. Tento model označujeme jako digitální dvojče, které nám přináší velký benefit při testování, optimalizaci výroby a predikci údržby i servisních zákroků, což má za následek snížení prostojů a nákladů na výrobu, a tak se zvyšuje celková efektivita výroby.



Obr. 1) 9 základních pilířů Průmyslu 4.0 [3]

3.2. Robotizace

Ačkoli od udělení prvního patentu na průmyslový robot Unimate v roce 1961 inženýru Georgi Devolovi uplynulo více než 60 let, trh průmyslových robotů zažívá největší rozvoj až v poslední dekádě. Jak ukazují celosvětové statistiky organizace Statista (viz Obr. 2), od poslední ekonomické recese v roce 2009 počet instalovaných robotů výrazně rostl krom malého poklesu v roce 2012 a 2019. [4] [5]



Obr. 2) Graf počtu celosvětově instalovaných průmyslových robotů podle roku [5]

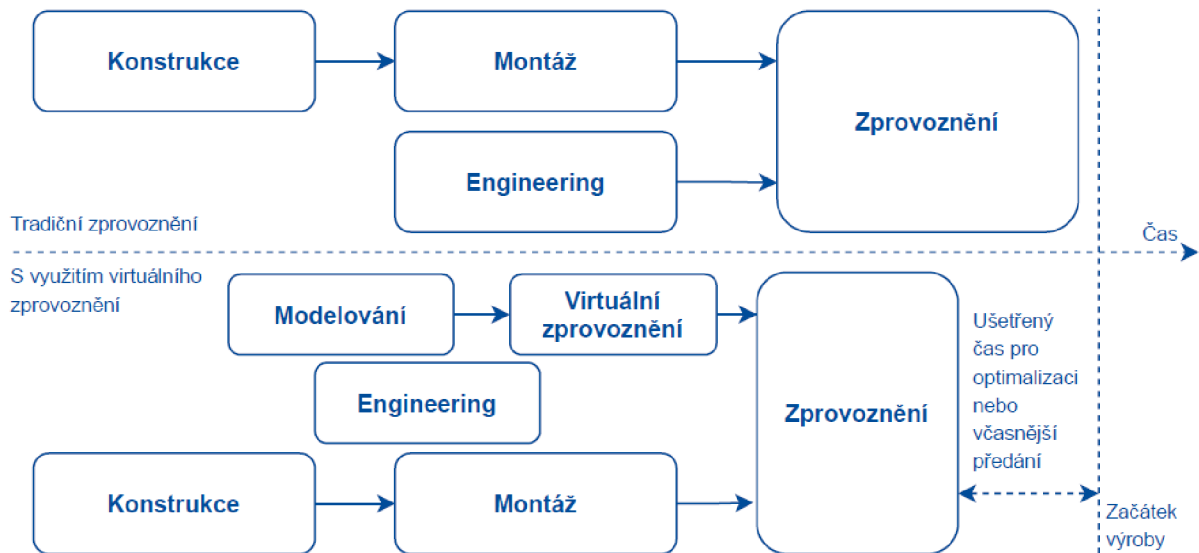
Za tímto růstem stojí mimo jiné i rychlé zavádění prvků průmyslu 4.0 do výroby a rozvoj simulačních programů, které umožňují rychlejší vývoj robotizovaných pracovišť, off-line programování robotů, simulaci komplexnějších výrobních celků a přinášejí další výhody.

Průmyslové roboty se stávají stále inteligentnějšími. Osazují se širokou škálou senzorů či prvky strojového vidění, díky kterým mají možnost interagovat se svým okolím. Na základě změny vstupních dat ze senzorů jsou schopny adekvátně přizpůsobit řídicí programy bez zásahu člověka a do určité míry se umějí rozhodovat samostatně. Takové roboty pak můžeme označit za autonomní. Navíc se při jejich programování začíná využívat umělá inteligence, díky které se při svém provozu učí a neustále zlepšují své rozhodovací algoritmy. S rozšiřujícími se schopnostmi robotů se stále rozrůstá pole aplikací, ve kterých je můžeme použít, což je jeden z dalších důvodů, který stojí za nárůstem prodaných kusů. [6]

3.2. Virtuální zprovoznění

3.1.1 Porovnání frame-worku tradičního a virtuálního zprovoznění

Právě díky většímu zájmu o množství instalovaných automatizovaných buněk a s nárůstem požadavků na jejich komplexnost se začala více prosazovat technologie virtuálního zprovoznění, která oproti tradičnímu přístupu umožňuje oživit výrobní systém ještě před zhotovením fyzického modelu, jak je znázorněno na Obr. 3., a umožňuje tak odladit chyby v PLC programu s dostatečným předstihem.

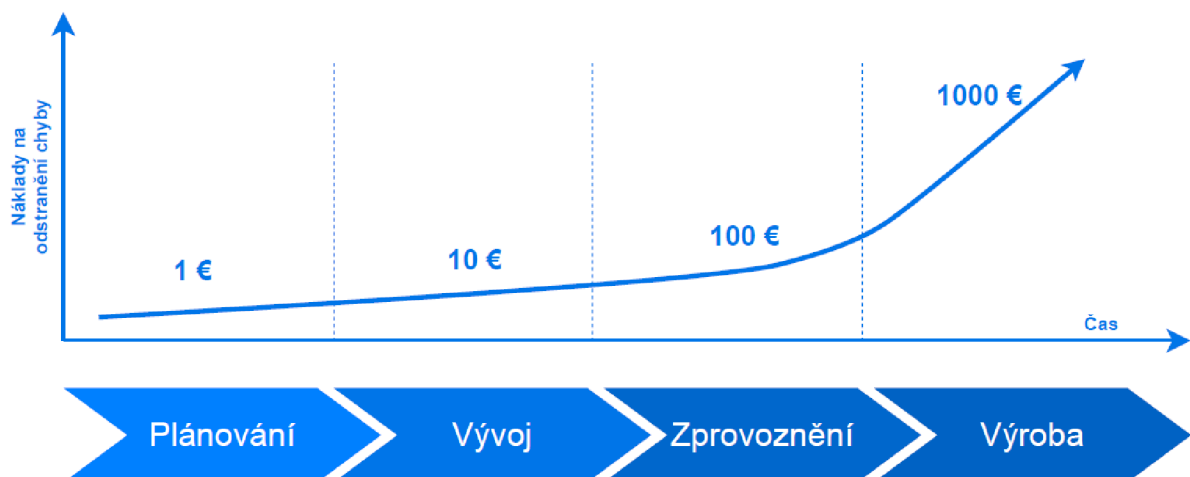


Obr. 3) Srovnání postupu tradičního a virtuálního zprovoznění [7]

Tento moderní přístup výrazně zkracuje celkový čas reálného uvádění výrobní technologie do provozu na místě instalace. Díky virtuálnímu zprovoznění probíhá odstranění chyb a optimalizace řídicího programu (PLC) na digitálním modelu, a tak neriskujeme poškození skutečných strojů. Programátoři nejsou vystaveni takovému stresu při realizaci zakázky, protože mají prostor otestovat své programy před montáží u zákazníka. Navíc díky této možnosti mohou detailněji propracovat řídicí programy, což přímo zvyšuje jejich kvalitu.

3.1.2 Exponenciální růst nákladů

V závislosti na tom, v které etapě životního cyklu výrobku je odhalena chyba, se odvíjí náklady potřebné na její odstranění. Tyto náklady však rostou exponenciálně s časem (viz Obr. 4). Z tohoto důvodu jsou právě simulační programy na vzestupu a přikládá se jim velký význam. Pokud je už ve fázi návrhu rozpoznána malá chyba, zamezí se vysokým nákladům, které by vznikly, kdyby daná chyba byla odhalena po uvedení výroby do chodu a musel by se její provoz odstavit. [8]



Obr. 4) Graf závislosti nákladů pro odstranění chyby na etapě odhalení v projektu [9]

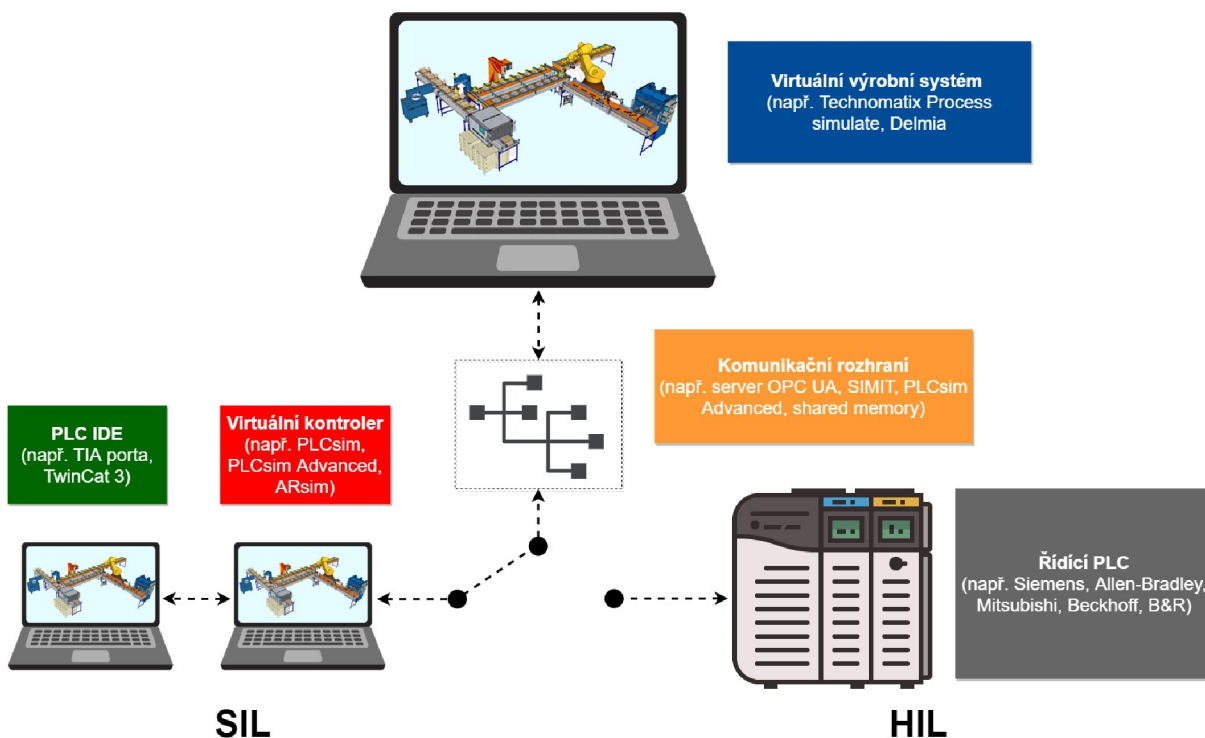
3.1.3 HIL vs. SIL

U virtuálního zprovoznění se nejčastěji používají simulační techniky Hardware-in-the-Loop (HIL) a Software-in-the-Loop (SIL). Simulační schémata lze vidět na Obr. 5.

SIL model nepoužívá žádný fyzický hardware. Celá simulace probíhá ve virtuálním prostředí na počítači. Chování hardwaru řídicí jednotky je tedy nutné co nejrealističtěji napodobit. Tady se dostáváme k pojmům simulace a emulace. Simulace je méně přesná technika, kdy chování a vlastnosti reálného PLC jsou naprogramovány ve vysokoúrovňových jazycích, proto chování procesoru simulované jednotky není shodné s reálnou jednotkou. Naproti tomu emulace je programována na úrovni jazyku symbolických instrukcí, častěji známých pod pojmem assembly, díky čemuž je chování řídicí jednotky věrohodnější. Ale i při použití emulátoru není chování tak přesné jako na reálném hardwaru. [10] [11]

Na druhé straně model HIL využívá reálného hardwaru řídicího systému, do kterého se nahraje zkompileovaný program, který řídí digitální mechanický model simulovaný v připojeném PC. Nevýhodou této techniky je větší časová náročnost zprovoznění a hlavně je potřeba mít reálný hardware k dispozici. [10]

V praxi se v první fázi často používá ožívování metodou SIL a po dostatečném ověření programu i vybraného hardwaru je zprovozněno řízení metodou HIL.

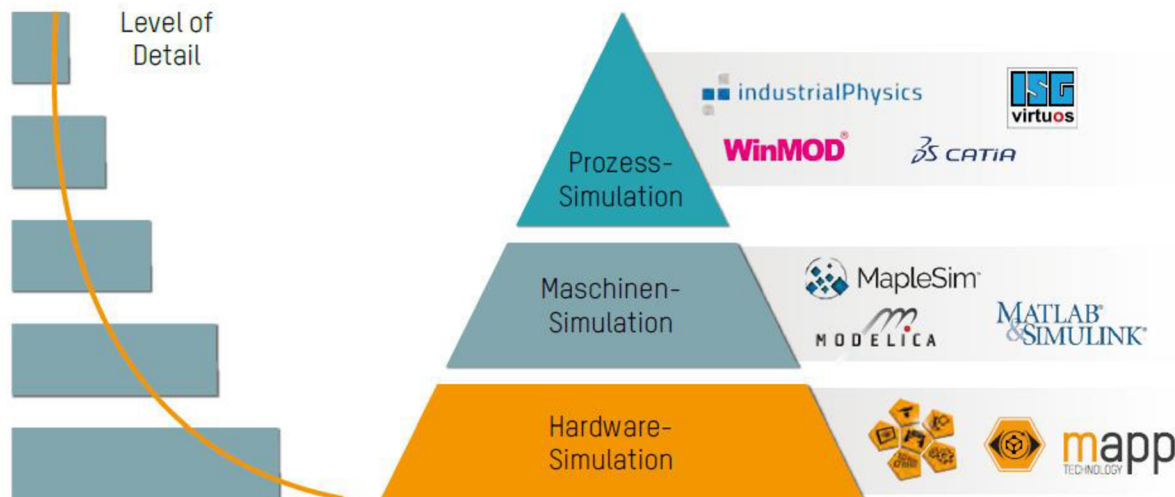


Obr. 5) Srovnání schémat simulačních modelů SIL a HIL

3.2 Simulační programy

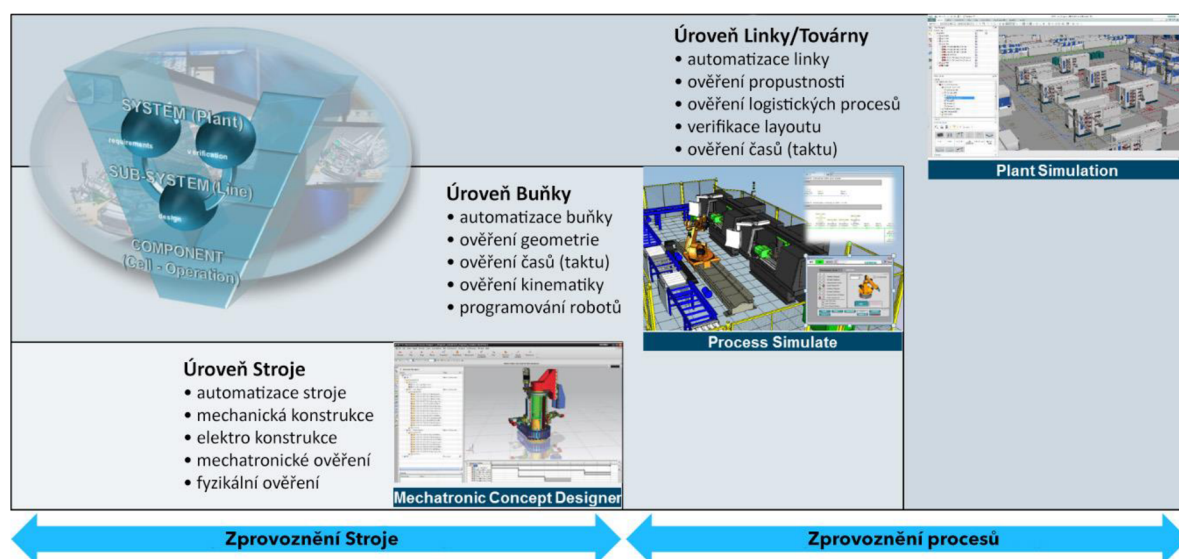
Aby mohl být systém virtuálně zprovozněn, je potřeba využít různých simulačních programů, které budou spolu schopny komunikovat a zastoupí jeden nebo více uzlů simulačního schématu vyobrazených na předchozím Obr. 5. Protože rozvoj virtuálního zprovoznění získává na oblibě napříč obory, tak také vzniká velké množství programů určených k tomuto účelu, které jsou často specializovány na dané aplikace.

Základní hledisko, dle kterého můžeme simulační nástroje dělit, je úroveň detailů simulovaného prvku. S vyšší úrovní se snižuje detail jednotlivých komponent a přibývají interakce mezi nimi. Na Obr. 6 je znázorněn příklad výše popsaného schématu s vyjmenovanými simulačními programy, které podporuje vývojové prostředí PLC jednotek společnosti B&R Automation.



Obr. 6) Kompatibilní simulační programy s PLC IDE společnosti B&R [12]

Podobné dělení lze pozorovat i u jedné z největších společností zabývajících se virtuálním zprovozněním firmy Siemens, která ve svém balíku digitální továrny Tecnomatix nabízí široké portfolio programů a pokrývá tak všechny úrovně virtuálního zprovoznění (viz Obr. 7). Na nejnižší úrovni s nejvyšším stupněm detailů je Mechatronics Concept Designer, což je rozšíření 3D modeláře NX, a je primárně určené pro softwarové oživení strojů. O úroveň výš stojí software Process Simulate, který je primárně určen pro zprovoznění výrobních buněk s využitím robotů. Nejvýše v hierarchii je software Plant Simulation, který řeší automatizaci celé výrobní linky nebo továrny a procesy s tím spojené jako je logistika, propustnost celého systému atd. [13]



Obr. 7) Portfolio produktů Siemens pro virtuální zprovoznění [13]

V další části se zaměřím na softwary vhodné pro virtuální zprovoznění robotických výrobních buněk, protože danou problematikou se zabývám i v praktické části této práce a některé z nich detailněji rozeberu v následující podkapitole. Robotické simulační programy dále můžeme rozdělit na programy vyvíjené výrobcí robotů, ve kterých je možno simulovat pouze roboty dané značky. Výhodou těchto programů je velice přesná emulace řídicích jednotek robotů, která se blíží reálnému kontroléru. Nevýhodou je přirozeně omezení výběru robotů pouze na jedinou značku. Druhou skupinou jsou univerzální programy, ve kterých můžeme programovat a simulovat chování téměř všech značek průmyslových robotů.

3.2.1 Simulační softwary výrobců robotů

RobotStudio

Jedná se o simulační program společnosti ABB, který je určen pro off-line programování robotů a díky postupnému rozšiřování funkcí v poslední době je již možné v tomto programu realizovat kompletní virtuální zprovoznění s využitím produktů ABB. Navíc od roku 2019 je možné napojit program na software SIMIT, který slouží primárně jako komunikační rozhraní s řídicími jednotkami firmy Siemens, ale také umožňuje komunikaci pomocí standardizovaného rozhraní OPC UA, a je tak možné napojit externí řízení různých výrobců.

Do RobotStudia je možné importovat 3D geometrii ve vybraných generických formátech, nebo využít Mechanism Modeler a vymodelovat si vlastní dopravníky, nástroje nebo koncové efekty. Při tvorbě realistických simulací se využívá tzv. Inteligentních (Smart) komponent, ve kterých definujeme řízení pomocí signálů a vazeb mezi nimi. Při tvorbě robotických drah je možné využít funkci AutoPath, která dráhy vygeneruje dle vybrané geometrie. Pokročilou funkcí řídicích kontrolérů ABB je funkce TrueMove, díky které robot přesně dodržuje naprogramovanou dráhu bez ohledu na požadované rychlosti TCP. Při využití funkce MultiMove je možné pomocí jedné řídicí jednotky ovládat více robotů zároveň.

Pro zobrazení virtuálního Teach Pendantu slouží funkce FlexPendant, která je věrohodnou emulací reálného panelu a nabízí i možnost pomocí ScreenMakeru vytvářet aplikace a obrazovky přímo pro ovládací panely.

K softwaru je možné dokoupit volitelné softwarové balíčky PowerPac pro specifické aplikace, například lakování, svařování, řezání, obrábění. [14]

RoboGuide

K dalším rozšířeným výrobcům průmyslových robotů bezesporu patří japonský výrobce Fanuc. Simulační program tohoto výrobce určený pro off-line programování svých robotů nese název RoboGuide. Nejnovější verze tohoto softwaru již mohou komunikovat s dalšími programy pomocí OPC, proto je možné připojit externí řízení téměř libovolného výrobce PLC.

Sám výrobce v propagačním listu zdůrazňuje jednoduchost používání softwaru RobotGuide, která je postavena na předpřipravených aplikačních nástrojích pro typické robotické operace: ChamferingPro, HandlingPRO, OLPCPRO, PaintPRO, PalletPRO, iRPickPRO, WeldPRO. K programu je možné dokoupit i následující opce pro rozšíření jeho funkcí.

- **AutoPlace** – automaticky vypočte nejvýhodnější polohu robota pro dosažení nejlepší doby cyklu a výkonu
- **Duty** – spočítá energetickou náročnost a výkon robota
- **Vision** – umožňuje simulovat aplikace pro 2D i 3D strojové vidění

- **Spray** – určená opce pro vizualizaci sprejování krytů na dílech
- **Coord** – umožňuje koordinaci a podporu více skupin robotů
- **Cable** – pokročilá simulace kabelů připojených k robotu nebo dalším zařízením [15]

Další softwary výrobců robotů

- Kuka - KukaSim
- Stäubli - Robotics Suite
- Yaskawa Motoman - MotoSim
- Kawasaki - K-ROSET

3.2.2 Univerzální simulační software

Tecnomatix Process Simulate

Jak již bylo zmíněno, Process Simulate patří do balíčku digitální továrny Tecnomatix, kterou vlastní koncern Siemens, a je primárně určen pro simulaci procesů na úrovni výrobní buňky s využitím robotů. Tato část softwaru se označuje přívlastkem Robotics. Velkou výhodou oproti konkurenci je možnost dokoupení balíčku Process Simulate Human, díky kterému můžeme testovat pracovní ergonomii lidí, například i při interakci s roboty nebo ovládání výrobní buňky. V základní konfiguraci program simuluje řídicí jednotky robotů zabudovaným ovladačem, takže časy nemusí přesně odpovídat reálným kontrolérům. Tento problém lze eliminovat, jestliže dokoupíme tzv. RCS (Real-Time Control System) modulů jednotlivých výrobců robotů. [16]

Do programu vkládáme 3D modely pouze ve formátu JT. Každý model před vložením musíme nadefinovat z nabídky komponent, ať už se jedná o vlastní konstrukci gripperu na robota, nebo pouze pracovní stůl buňky. Toto definování vkládaných modelů je nutné provést, aby program věděl, jak má na daný model dále nahlížet. Abychom mohli řídit mechanismy komponent, což může být právě koncový efektor robotu, musíme sestavit pohyb v Kinematic Editoru. Následně je možné skládat operace sekvenčně z předdefinované nabídky operací a řídit celý proces časově, nebo přepnout do Line Simulation módu, ve kterém je celý proces řízen událostně v závislosti na externích signálech. U událostně řízeného módu je nutno vytvořit tzv. materiálový tok, který určuje sekvenci jednotlivých operací, kterými musí výrobek v rámci jednoho výrobního cyklu projít.

Další konkurenční výhodou je kompatibilita celého řešení digitální továrny Tecnomatix, kdy při využívání více produktů z tohoto portfolia je možno data z jednotlivých programů mezi sebou sdílet. V programu Process Designer můžeme navrhnout schéma celé buňky, které se následně importuje do programu Process Simulate, kde se návrh detailně rozpracuje do realističtější podoby, a po ověření všech operací a taktu celé buňky je možné data sdílet do programu Plant Simulation pro řízení procesů v rámci celé továrny. Ucelenost řešení firmy Siemens završuje možnost přímého napojení simulačních programů na vlastní řídicí jednotky přes PLCsim Advanced, ale nabízí se i možnost využít konektivity standardizovaným protokolem OPC UA a zprovoznit tak externí řízení i jiných výrobců PLC jednotek. [17]

RF::Suite

RF::Suite je modulární softwarový balíček německé společnosti EKS InTec. V poslední době se stává stále populárnější a daří se mu získávat i velké zákazníky, jakými jsou například automobilky. Celý systém je postaven na myšlence, že při tvorbě nového projektu si integrátor

této technologie vybírá jednotlivé moduly s příslušnými funkcemi, které v dané aplikaci potřebuje. Další velkou výhodou je otevřenost tohoto řešení pro jednoduchou komunikaci s kontroléry různých výrobců robotů a s předními výrobci PLC.

Konkrétní moduly RF::Suite jsou rozděleny podle funkce do následujících 4 skupin:

- **Validace dat:**

- RF::Max² - tvorba a správa projektů virtuálního zprovoznění a tvorba digitálních stínů
- RF::RobCheck - jednoduchý nástroj pro kontrolu robotických programů a jejich konfigurací
- RF::EGEdit - úprava 3D geometrie modelů, nástroje pro tvorbu i úpravu kinematik, jednoduché měření vzdáleností
- RF::TIAExporter - nástroj určený pro rychlý export dat pro virtuální zprovoznění pomocí softwaru TIA Portal firmy Siemens
- RF::HMI - nástroj pro vizualizaci dat a pro tvorbu uživatelských rozhraní, připravené konektory pro komunikace pomocí shared memory a software WinMod

- **Virtuální zprovoznění:**

- RF::YAMS - modul pro vizualizaci 3D modelů geometrie a pro jejich následnou simulaci na základě materiálového toku
- RF::RobSim - pružný kompilátor umožňující testování robotických programů nebo jejich nahrání z reálného kontroleru, nabízí knihovnu kinematiky robotů předních výrobců, ale je také možné vytvořit kinematiku vlastní
- RF::ViPer - moderní nástroj pro simulaci periferií, obsahuje bloky a elementy, které je možné programovat dle standardu IEC 61131-3
- RF::FSBox - nabízí simulaci komunikace pomocí Profinet a umožňuje simulovat více instancí PLC najednou

- **Nástroje pro připojení:**

- RF::RobotConnect - umožňuje propojit simulaci s virtuálním nebo reálným kontrolérem robotu
- RF::PLCConnect - umožňuje propojit simulaci s virtuálním i reálným PLC
- RF::DS - konektor pro shared memory umožňující komunikaci mezi různými PC díky vytvoření serveru, který spravuje (posílá data)

- **Analytické nástroje:**

- RF::Analyser - nástroj pro analýzu signálů ze všech připojených externích jednotek slouží primárně k hledání chyb
- RF::Recorder - umožňuje nahrávat a následně analyzovat interní procesy, největší uplatnění najde při optimalizaci procesů digitální továrny

[18]

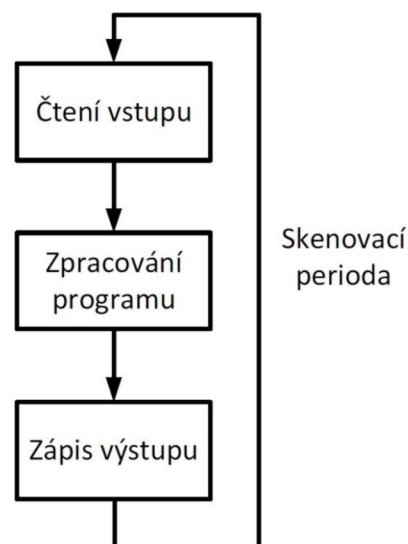
Další univerzální softwary pro simulaci robotických operací

- *Delmia V5 Robotic (Dassault Systèmes)*
- Visual Components
- FASTSUITE
- Emulate3D (Rockwell Automation)

Pro praktickou část byl zvolen program Tecnomatix Process Simulate, hlavně pro svou univerzálnost, a díky tomu, že pro studenty Ústavu výrobních strojů, systémů a robotiky na Fakultě strojního inženýrství VUT je tento program k dispozici.

3.2.3 PLC

Programovatelný logický automat (angl. Programmable logical controller) je řídicí systém určený pro řízení průmyslových strojů a technologických procesů. Původně bylo PLC navrženo pro řízení čistě logických úloh, konkrétně pro nahrazení tzv. reléové logiky, kdy pomocí spínacích relé byla vytvořena struktura programu pro řízení procesů, při kterých byl algoritmus tvořen schématem zapojení, což kladlo velké nároky na kabeláž, a navíc byly podobné systémy málo flexibilní, protože pro změnu logiky bylo nutné přepojit posloupnost relé. Díky PLC je možné tyto logické úlohy měnit flexibilně změnou uživatelského programu, kdy při nutné změně stačí upravit a přehrát stávající kód. [19]



Obr. 8) Skenovací perioda PLC [19]

Jak napovídá slovo automat v názvu, cyklus PLC jednotek probíhá v automatickém cyklu, kterému říkáme skenovací perioda (viz Obr. 8). Nejdříve řídicí jednotka načte vstupy a zapíše je, následně dle vstupů vykoná uživatelský program a zapíše hodnoty na výstup. Skenovací perioda má přesně stanovený časový blok, ve kterém musí proběhnout. Pokud jednotka z výkonnostního nebo jakéhokoliv hlediska není schopna tuto dobu dodržet, automat se vypne a zahlásí chybovou hlášku. Tato vlastnost časové determinovanosti je velice důležitá, protože PLC často ovládá reálný hardware, který by se v důsledku špatného vyhodnocení uživatelského programu mohl zničit nebo dokonce někoho zranit.

Protože využití PLC v průmyslu je dominantní, je na trhu také velký počet výrobců. Dnes je pomocí těchto jednotek možné řídit většinu procesů včetně spojitého řízení, řízení motorů apod.

Jednotliví výrobci PLC zpravidla nabízejí softwarové vybavení určené pro programování svých automatů, kterému se říká vývojové prostředí (angl. IDE – Integrated Development Environment). Univerzální přístup programování nezávisle na výrobci je zaručen díky standardu IEC 61131-3, ve kterém je popsáno 5 základních programovacích jazyků určených pro tvorbu uživatelských programů. [20]

V praktické části budu pro virtuální zprovoznění používat PLC firmy Siemens s jejich PLC IDE s názvem TIA Portal V16. Hlavním důvodem, proč jsem zvolil značku Siemens, je její největší zastoupení v průmyslových aplikacích. A proto si myslím, že osvojení práce v daném softwaru a znalost jejich modelových řad nejvíce využiji v budoucí praxi.

3.2.4 Komunikační rozhraní pro virtuální zprovoznění

Poslední podmínkou pro dosažení virtuálního zprovoznění je nutnost propojit simulační program, ve kterém je mechanický model výrobního systému, s emulátorem PLC, který je často součástí daného PLC IDE. Možnosti volby této komunikace se odvíjejí od použitých programů. V následujícím přehledu se pokusím popsat univerzální přístupy pro danou komunikaci, ale také softwary určené výhradně k propojení produktů firmy Siemens, jelikož jsem v předchozím textu vybral její zástupce pro praktickou část této práce.

OPC

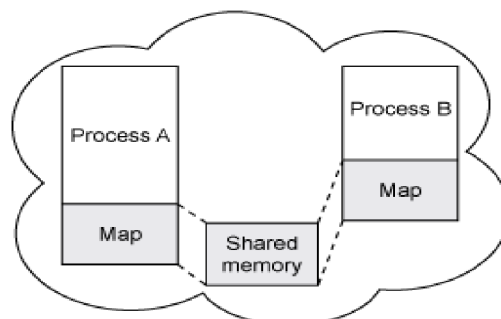
OPC je průmyslový komunikační protokol, který vznikl v roce 1996 za účelem zlepšení komunikační kompatibility mezi HMI/SCADA systémy s PLC v oblasti automatického řízení. Za jeho vznikem a následným vývojem stojí sdružení s názvem OPC Foundation.

První verze protokolu byla omezena pouze na operační systém Windows. Akronym OPC původně vznikl ze slovního spojení OLE (object linking and embedding) for Process Control. Dnes tuto vývojově nejstarší verzi protokolu nazýváme termínem OPC Classic.

S postupným rozvojem průmyslové komunikace se zvýšily nároky na bezpečnost a flexibilitu datových modelů komunikačních rozhraní. Proto byla představena nová verze OPC UA, kde celý akronym znamená Open Platform Communication Unified Architecture. Jak již název napovídá, nová verze protokolu není závislá na určitém operačním systému. Navíc celá architektura stojí na tzv. servisně orientované architektuře (SOA) využívající ethernetovou komunikaci klient-server. Díky velké flexibilitě našel tento protokol široké uplatnění napříč aplikacemi od nejnižší úrovně komunikace mezi senzory až do nejvyšší úrovně nadřazených systémů typu ERP. [21]

Shared Memory

Jedná se o termín z informatiky, kdy část operační paměti (RAM) je poskytována více procesům, jak je zachyceno na Obr. 9. V případě virtuálního zprovoznění k datům ve sdílené paměti přistupují dva simulační programy, které tak mohou oboustranně sdílet data a aktualizovat je v závislosti na daném stavu. [22]



Obr. 9) Schéma sdílené paměti (Shared memory) RAM [22]

WinMOD

WinMod je modulární software, který je na trhu již od roku 1995. V programu je možné simulovat celá zařízení a věrohodně tak modelovat jejich chování. Mezi taková zařízení mohou patřit typicky elektromotory nebo dopravníky, které je jednoduché do programu implementovat díky obsáhlé knihovně **Fact Lib E-motors**. V takovém případě pak WinMod komunikuje jak s emulátorem daného výrobce PLC, ve kterém probíhá logika nadřazeného PLC, tak se simulačním programem, ve kterém je virtuální model celého pracoviště. V tomto případě tedy program zastupuje uzel komunikačního rozhraní v rámci celé simulace.

Výrobce ale nabízí i rozšíření **SimLine**, ve kterém je možné sestavit celý simulační model bez použití softwarů třetích stran. Proto je možné považovat WinMod za plnohodnotný simulační software. [23]

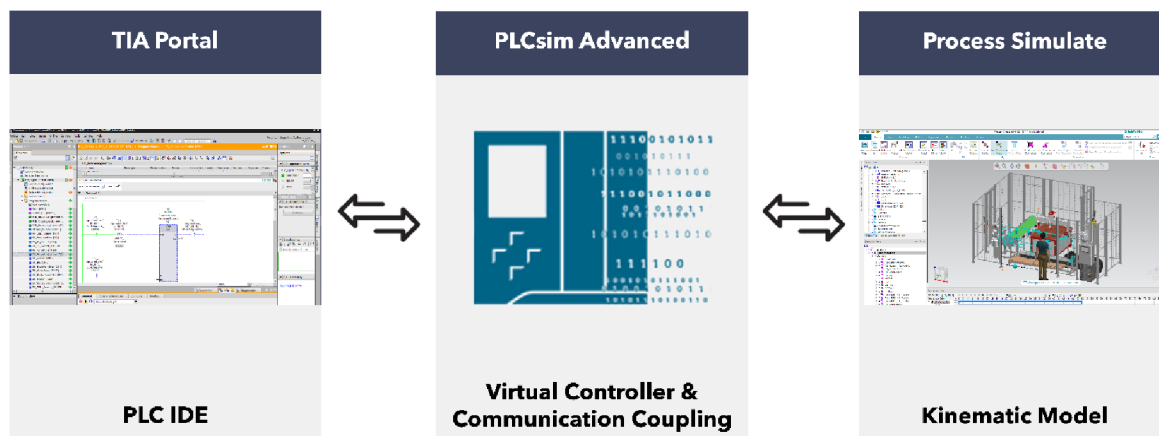
SIMIT

SIMIT je komplexní simulační platforma firmy Siemens, která nabízí velké množství komunikačních rozhraní od standardních konektorů pro komunikaci přes protokol OPC nebo Shared Memory až po konektory svých vlastních softwarů jako jsou PLCSIM, Mechatronics Concept designer a další. Navíc nabízí možnost nastavit komunikaci s hardwarovou jednotkou SIMIT Unit, která slouží jako hardwarové rozhraní mezi reálným PLC a softwarem SIMIT, a zjednodušuje zapojení konfigurace simulačního modelu Hardware-in-the-Loop. [24]

PLCsim Advanced

Vývojově se jedná o nástupce softwaru PLCsim, který slouží pro simulaci automatů firmy Siemens. Na rozdíl od svého předchůdce je PLCsim Advance emulátor řídicích jednotek, díky čemuž poskytuje reálnější zpětnou vazbu zvoleného hardwaru. Mezi další výhody patří možnost přímé komunikace se simulačními programy stejného výrobce nebo schopnost využít virtuálního ethernetového adaptéru, který si program při spuštění své instance vytvoří a díky tomu, je možno nastavit ethernetovou komunikaci s libovolnými zařízeními v lokální síti. K nevýhodám patří hlavně omezení na PLC řady S7-1500 a ET200SP. Navíc tento program na rozdíl od PLCsim není součástí vývojového prostředí TIA Portal, takže je nutné jej dodatečně dokoupit.

Pro praktickou část této práce jsem na pozici komunikačního uzlu vybral poslední zmiňovaný program PLCsim Advanced právě kvůli tomu, že jsem již v přechozí části vybral PLC značky Siemens a simulační program Process Simulate stejného výrobce. Proto se tato volba pro nastavení komunikace jeví jako nejefektivnější a nabízí i relativně širokou škálu možností pro budoucí rozvoj. Výsledné simulační schéma je představeno na Obr. 10.

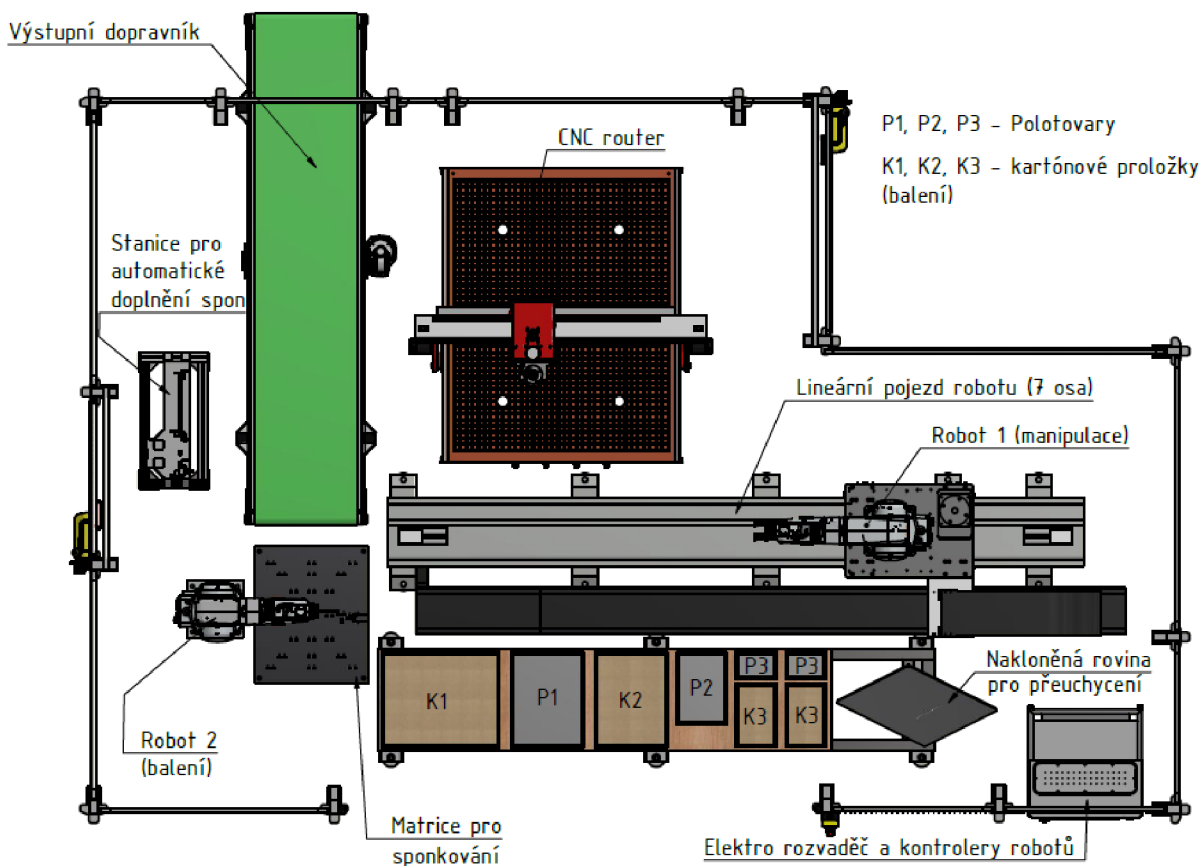


Obr. 10) Schéma simulačního modelu virtuálního zprovoznění pro praktickou část

3.3 Analýza zadaného pracoviště

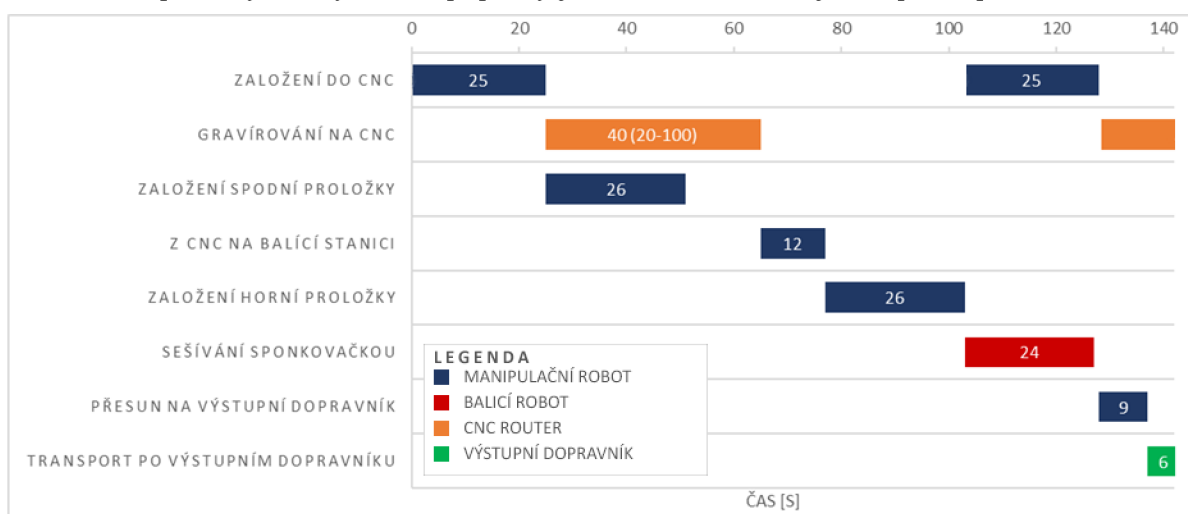
Virtuální zprovoznění bude provedeno na zadané robotizované buňce. Účelem zadaného výrobního systému je obsluhovat CNC router, který bude gravírovat deskové polotovary 3 různých velikostí. Po obrobení má probíhat také balení finálních produktů, kdy po zabalení dochází k jejich transportu z výrobního systému pomocí výstupního dopravníku. Pro lepší představu je celá buňka zobrazena z ptáčí perspektivy, včetně popisků na Obr. 11.

Výsledné obrobky mají sloužit jako reklamní nebo výstražné cedule. Hlavním materiálem používaným k jejich výrobě je sendvičový materiál, který se skládá z okrajových vrstev hliníku o tloušťce 0,12 - 0,5 mm a ze střední vrstvy z PVC. Celková tloušťka standardizovaných rozměrů je 2, 3 a 4 mm. Tento materiál je znám pod svým obchodním názvem Dibond. Protože se ale jedná o reklamní prostředky, byl stanoven požadavek na vyšší flexibilitu systému, aby bylo možné rychle měnit rozměry polotovarů a případně i jejich materiál. Kvůli tomu je zásobník v buňce navržen pouze formou plošiny určené pro stohování různých polotovarů podle rozměru, které odebírá manipulační robot rozšířený o sedmou osu formou lineárního pojezdu za účelem zvětšení jeho pracovního prostoru. Protože se jedná o deskový materiál s malou hmotností, tak je koncový efektor robotu složen pouze ze 4 kusů aktivních pneumatických přísavek. Při prvním uchopení robotem není zaručena přesná lokace polotovaru a pro další operaci, kterou je CNC gravírování, je potřeba zaručit přesnou polohu. Proto robot polotovar znovu přechytné pomocí nakloněné roviny s přesnou aretací polohy. To znamená, že při najetí s polotovarem nad plošinu jej uvolní a ten díky gravitaci sklouzne do přesného místa, a tak robot při opětovném upnutí daného materiálu má již zaručeně přesnou pozici úchyty. Následně je tedy možné s dostatečnou přesností založit polotovar na pneumatický stůl CNC routeru, který může začít obrábět.



Obr. 11) Zadaná robotizovaná buňka (pohled z ptáčí perspektivy)

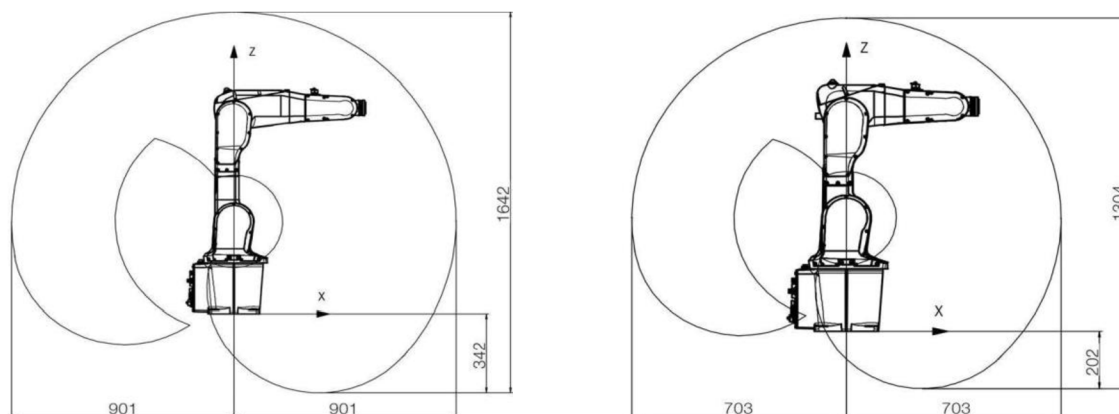
Druhou významnou operací vykonávanou v rámci celé buňky je balení již obrobek, které se řeší pomocí spojení dvou kartonových proložek pneumatickou sponkovačkou, kterou je osazen druhý robot. Samotné spojení probíhá na speciálně připraveném ocelovém podkladu. Na něj nejprve první robot založí spodní kartonovou proložku. Po dokončení gravírovací operace na spodní vrstvu lepenky umístí obrobek a následně na něj uloží horní kartonovou proložku. Pak začne operace sešívání, kterou provádí druhý robot. Po dokončení této procedury manipulační robot přemístí již zabalený produkt na výstupní dopravník, který je spuštěn po dokončení této manipulační procedury signálem z nadřazeného PLC. Pro lepší přehlednost a pochopení posloupnosti jednotlivých operací jsem vypracoval Ganttův digram pro výrobu jednoho kusu produktu (viz Obr. 12). Časové úseky operací jsou barevně odlišeny podle zařízení, které je vykonávají. Gravírování má proměnlivou délku v závislosti na velikosti polotovaru a daném technologickém postupu. Při pracovním proceduře balicího robotu začne manipulační robot vykonávat operace následujícího výrobku. Hlavní komponenty buňky budou popsány jednotlivě v následujících podkapitolách.



Obr. 12) Ganttův diagram operací pro výrobu jednice

3.3.1 Průmyslové roboty ABB IRB 1200

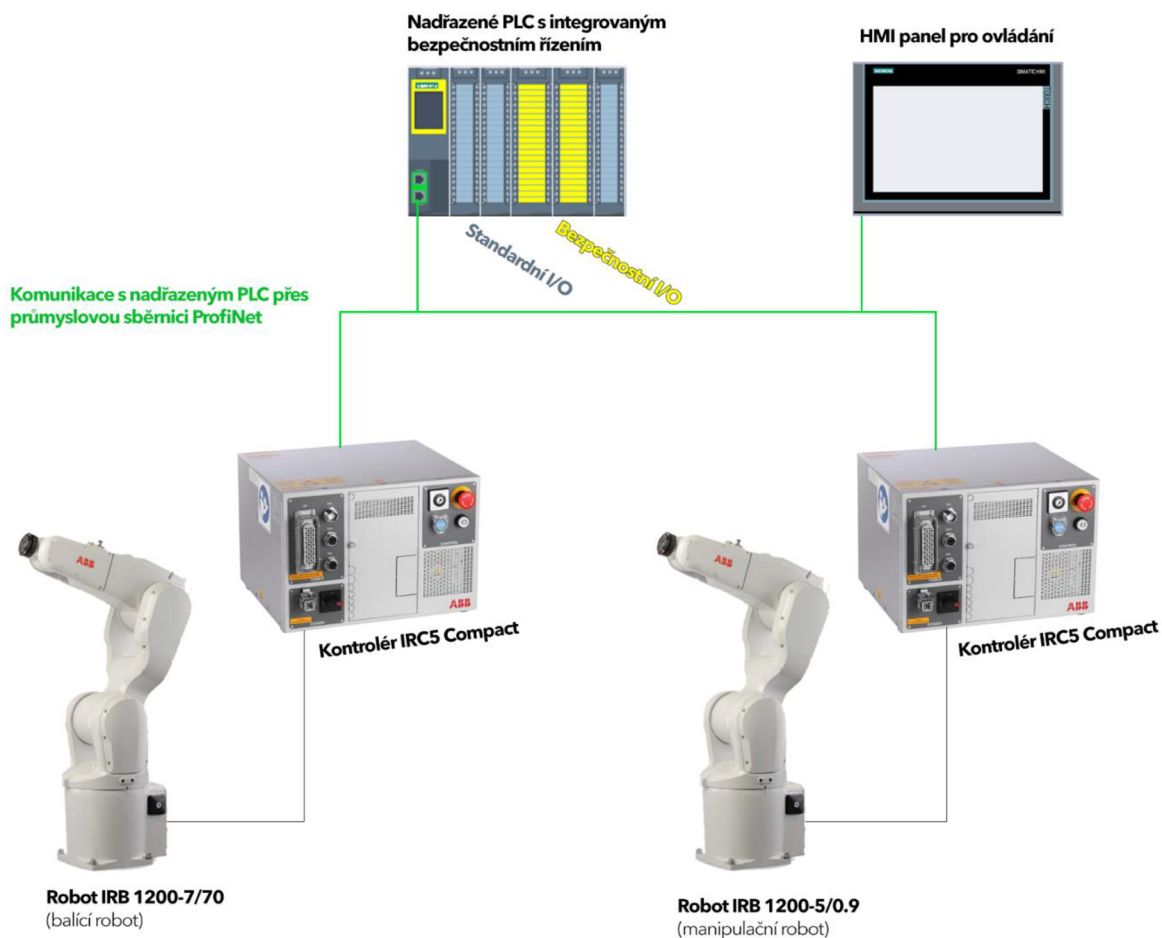
Oba použité roboty jsou od společnosti ABB z modelové řady IRB 1200, liší se pouze dosahem, který může být 0,7 m nebo 0,9 m, a maximální nosností, která je 5 kg nebo 7 kg a logicky je menší u robotu s větším dosahem. Dle těchto parametrů řadíme tyto roboty do kategorie malých průmyslových robotů. I přes to díky velké pracovní obálce, která se nachází v těsné blízkosti robotu, zvládnou pokrýt širokou škálu různých aplikací.



Obr. 13) Pracovní obálka robotu IRB 1200-5/0.9 (vlevo) a robotu IRB 1200-7/0.7 (vpravo)

Řízení robotů je zajištěno pomocí kontroléru IRC5, který je postupně nahrazován u většiny modelových řad robotů ABB novou jednotkou OmniCore. To je ale plánováno u řady 1200 až na podzim roku 2021, proto byl v této práci použit řídicí systém IRC5.

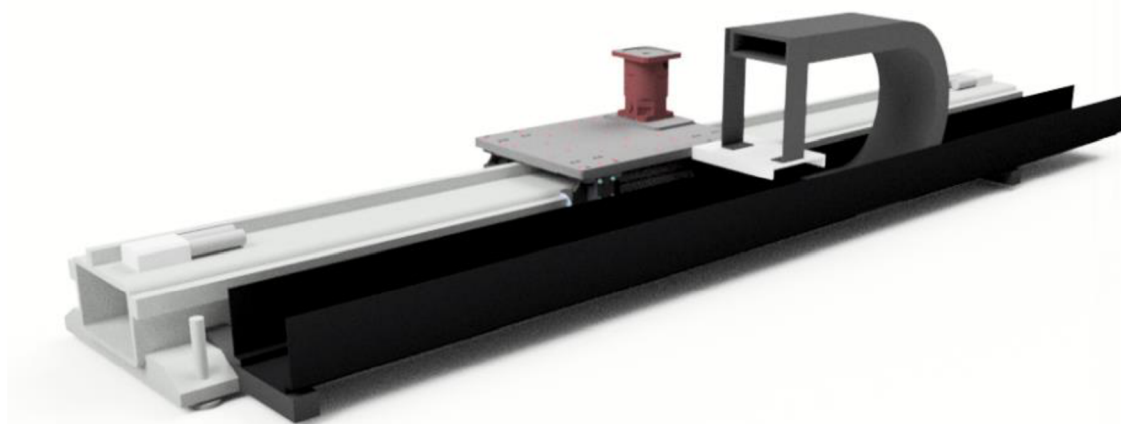
Pro řízení dvou robotů se nabízejí dvě varianty konfigurace. Buď je možné koupit jednotlivé kontroléry pro každý robot zvlášť, kdy u menších robotů je možné použít tzv. verzi Compact, což je zmenšená a také ekonomicky výhodnější varianta. Nebo lze koupit pouze jeden plnohodnotný kontrolér IRC5 a k němu dokoupit rozšiřující Drive Modul pro řízení výkonové části dalšího robotu. Díky tomu lze ovládat a programovat oba roboty z jedné řídicí části. IRC5 totiž disponuje tzv. funkcí MultiMove, proto je možné jedním kontrolérem ovládat až čtyři roboty. Díky této funkci je možné jednoduše vytvářet komplexní dráhy při kooperaci obou robotů. Navíc v této konfiguraci s jedním kontrolérem při pořízení rozšiřujících opcí, jakou je třeba připojení na průmyslovou sběrnici Profinet, se platí vybrané doplňky pouze jednou. Cena obou konfigurací je v tomto případě téměř totožná, ale mírně ekonomicky příznivěji vychází použití klasické jednotky IRC5 s Drive Modulem. Ale u výrobní buňky z praktické části této práce byla použita druhá varianta konfigurace s využitím dvou kontrolérů IRC5 Compact. Hlavním argumentem této volby je flexibilita vůči budoucím změnám. Například po demontáži buňky je tak možné využít roboty samostatně na různých pracovištích, protože nejsou závislé na společném řízení. [25] Schéma řízení robotů a komunikace s nadřazeným PLC je naznačeno na Obr. 14.



Obr. 14) Schéma řízení celé buňky z praktické části

3.3.2 Lineární pojezd Güdel TMF-1

Přestože firma ABB nabízí lineární pojedy pro své průmyslové roboty, je jejich nabídka externích os omezená a je zaměřená na větší roboty. Proto byl v buňce použit pojezd firmy Güdel, která se na podobná zařízení pro roboty zaměřuje. Kvůli malé hmotnosti robotu IRB 1200-5/0.9 (56 kg), který bude na pojezd upnut, byl použit nejmenší model z řady TMF (Track Motion Flour). Tato řada je určena pro transfer robotu lehce nad úroveň podlahy na rozdíl od portálových jednotek, u kterých je základna robotu instalována do vyvýšené úrovně. U nejmenšího modelu TMF-1 není možné vést energetický řetěz středem pojezdu a nabízí se pouze varianta jeho umístění z boku (viz Obr. 15). Vybraný model je dodáván v délce 3–12 m. V buňce je použita varianta s délkou 3 m. Pro možnost integrace řízení motoru pojezdu do řídicího systému IRC5 a synchronního ovládání 7. osy robotu je nutné osadit pojezd pohonnou jednotkou firmy ABB. [26]



Obr. 15) 3D model lineární jednotky Güdel TMF-1

3.3.3 Pneumatická sponkovačka BeA Autotec 246 s doplňovací stanicí

Balení produktu bude probíhat pomocí pneumatické sponkovačky německé společnosti BeA, která se zabývá technologií sešívání pomocí spon pro průmyslové aplikace. Konkrétně se jedná o model 246 řady Autotec, která je určena pro plně automatizované provozy. Hmotnost sponkovačky je 3 kg a disponuje zásobníkem pro 180 ks spon. Navíc je možné všechny modely této řady doplnit o automatickou doplňovací stanicí, která je určena pro aplikace se šestiosými roboty (viz Obr. 16). Zásobník doplňovací stanice má kapacitu až 30 000 spon a výrazně tak prodlouží dobu mezi nutným zásahem obsluhy pro doplnění spon. Pro spolehlivé spojení dvou kartonových proložek pomocí spon je nutné v místě spojů vytvořit na podkladu matici se správnou geometrií, díky které se konce spon zahnou a vytvoří spolehlivý spoj. Matici dodává sám výrobce a její povrch je teplotně upravován kalením za účelem zpevnění materiálu a prodloužení jeho životnosti, protože při spojování působí ostré konce spon přímo na povrch matrice. [27]



Obr. 16) Spunkovačka Autotec 246 s automatickou doplňovací stanicí firmy BeA [27]

3.3.4 Pneumatické uchopovací prvky SMC

Uchopení deskových polotovarů a kartonových proložek při robotické manipulaci zajišťují aktivní vakuové prvky firmy SMC. Konkrétně jsou použity 4 kusy tzv. bezotiskových přísavek o průměru 25 mm řady ZP2. Při výběru byl proveden orientační výpočet pro správnou volbu průměru pneumatických přísavek pro danou aplikaci. Při manipulaci s polotovarem o maximálním navrhovaném rozměru s maximální možnou celkovou tloušťkou 4 mm a tloušťkou hliníkových stěn 0,5 mm je hmotnost produktu včetně balicího materiálu přibližně 800 g. Hodnoty dalších parametrů použitých ve výpočtu jsou uvedeny v části následující části s výpočty. Zvolená varianta byla konzultována s obchodními zástupci společnosti SMC, kdy byl výběr potvrzen a byly k němu doplněny další nutné pneumatické prvky. Ale protože deska po gravírování bude mít lehce nespojitý povrch, tak samotná technická podpora SMC navrhl, že při realizaci projektu zapůjčí různé přísavky na testování. Protože cílem této práce nebyl konstrukční návrh gripperu, proto byla tato navržená varianta dostačující.

Výpočet průměru přísavky

- m – Max. hmotnost: 0,8 kg
- a – Max. zrychlení robotu (hodnota při nouzovém zastavení): 36 ms^{-2}
- f – Koeficient tření: 0,5
- g – Tíhové zrychlení: $9,8 \text{ ms}^{-2}$
- k – Koeficient bezpečnosti: 1,5
- n – Počet použitých přísavek: 4
- Δp – Relativní rozdíl tlaku vakua: - 0,7 baru

Teoretická uchopovací síla:

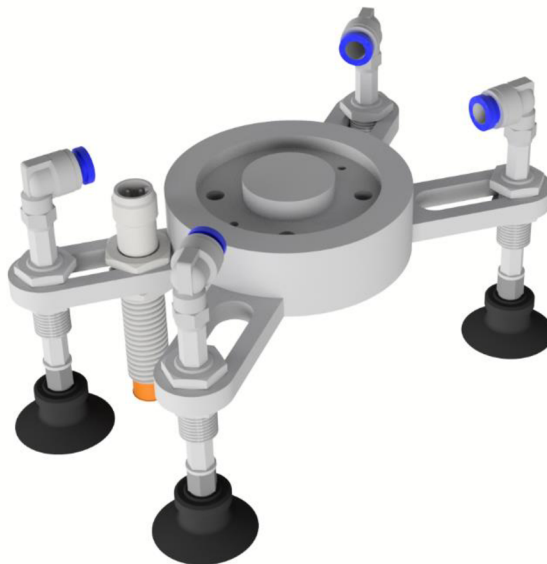
$$F_h = m \cdot \left(g + \frac{a}{f} \right) \cdot k = 0,8 \cdot \left(9,81 + \frac{36}{0,5} \right) \cdot 1,5 = 98,2 \text{ N} \quad (1)$$

Teoretický průměr jedné přísavky:

$$d = \sqrt{\frac{F_h \cdot 4}{\pi \cdot |\Delta p| \cdot n}} = \sqrt{\frac{F_h \cdot 4}{\pi \cdot |-0,7 \cdot 10^5| \cdot 4}} = 0,0211 \cong 21 \text{ mm} \quad (2)$$

Přísavky jsou namontovány na adaptér robotu tak, aby tvořily obdélníkové pole (viz Obr. 17). Každá z nich je navíc připevněna na odpružený držák se zdvihem 10 mm, který se při najíždění robotu nad materiál při prvním styku začne zasouvat, dokud kapacitní senzor, kterým je konstrukce adaptéru osazena, nedetekuje materiál. Při detekci materiálu robot zastaví a řídicí

systém robotu vyše signál pro otevření ventilu pneumatického rozvodu koncového efektoru, kterým spustí tvorba vakuum. Zdroj vakua zajišťuje ejektor s tlakovým čidlem a 2 PNP výstupy. Snímač tlaku tak měří aktuální hodnotu tlaku v ejektoru a díky digitálním výstupům je možné hlídat 2 nastavené hladiny reprezentující stav uchopení a upuštění manipulovaného předmětu, o kterých dostává zpětnou vazbu řídicí systém. [28]



Obr. 17) 3D model použitého gripperu s přísavkami firmy SMC

3.3.5 CNC router

Gravírování cedulí zajišťuje tříosý CNC router vlastní konstrukce (viz Obr. 18) s pracovním prostorem 800x800x200mm, což je více než dostačující pro maximální velikost desek, které se budou na stroji v rámci robotizované buňky vyrábět. Důležitým prvkem pro plně automatizovaný provoz je vakuový stůl pro upínání deskových obrobků.



Obr. 18) CNC frézovací router s vakuovým upínacím stolem

3.3.6 Pásový dopravník Alutec KK

Výstupním zařízením celé buňky je pásový dopravník sestavený z hliníkového konstrukčního systému Alutec KK s osovou roztečí válců 2200 mm a šířkou pásu 550 mm (viz Obr. 19), který je osazen asynchronním motorem Ampo, který zajišťuje dostatečnou dynamiku pohybu s cílenou rychlostí dopravníku 300 mm/s. [29]

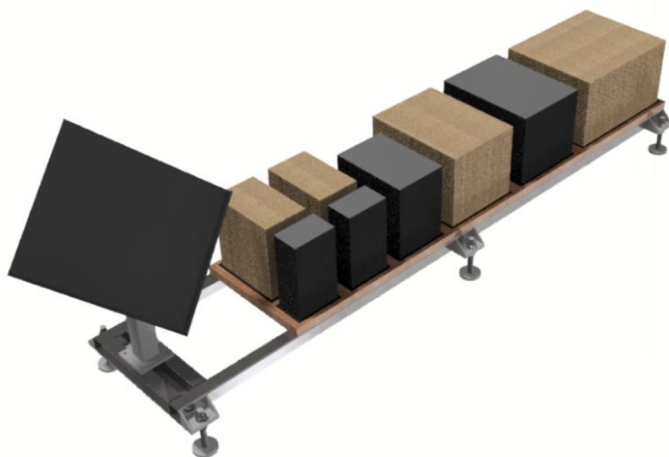


Obr. 19) 3D model pásového dopravníku Alutec KK

3.3.7 Vertikální zásobník s přechycením

Zásobování deskovými polotovary je prováděno obsluhou manuálně, ta je při nedostatku materiálu buď vyzvána systémem k jeho doplnění, nebo může operátor stisknutím tlačítka na panelu HMI vystavit požadavek pro vstup do buňky. Tak se celý výrobní proces zastaví na konci dalšího cyklu a odblokují se bezpečnostní prvky a obsluha má možnost doplnit zásobník.

Jak jsem již zmiňoval v popisu funkce celé buňky (viz kapitola 3.3), zásobník je koncipován co nejjednodušeji pro dosažení vysoké flexibility. To z toho důvodu, aby bylo možné rychle změnit rozměry gravírovaných cedulí. Proto slouží zásobník jako jednoduché úložiště, v němž se desky skládají na sebe na určenou pozici, která je naznačena na pokládací desce pomocí pásky, aby bylo jasné kam polotovary umisťovat. Tato zjednodušená koncepce nezaručí přesnou polohu materiálu. K tomu slouží stanice pro přechycení, kdy robot najede s deskovým materiálem nad nakloněnou rovinu, uvolní materiál, který díky gravitaci sklouzne do rohu plošiny a zastaví se v přesné pozici o ohnutý lem. K ověření, že skluz proběhl podle předpokladu, slouží detekční senzor. Pokud senzor vyšle kladný signál, robot znovu uchopí desku již na přesně specifikované pozici a může s ní manipulovat dál v rámci buňky.



Obr. 20) 3D model zásobníku se základnou pro přechycení polotovarů robotem

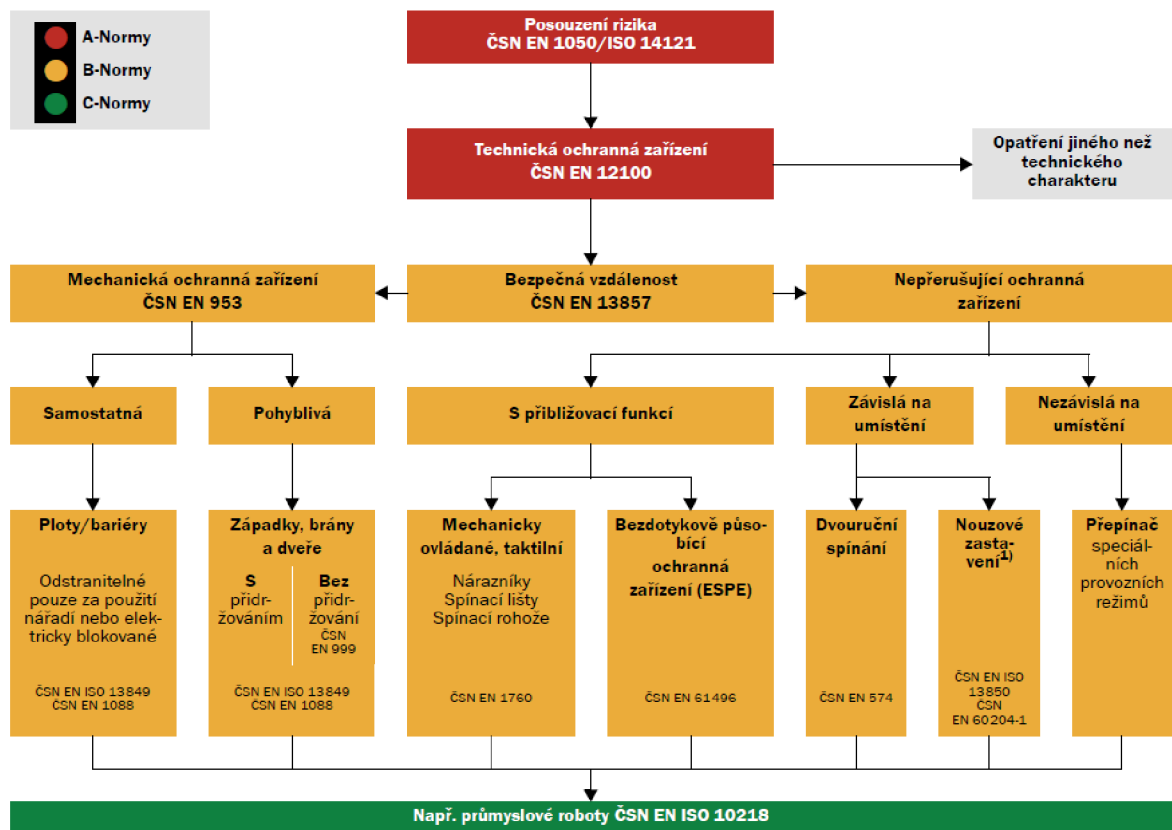
3.4 Funkční bezpečnost

Před uvedením jakéhokoliv stroje, v našem případě automatizovaného pracoviště do provozu, musí být zajištěna dostatečná bezpečnost. Proto je nutné provést tzv. analýzu rizik. Tento proces posuzování možných nebezpečí úrazu je dle nařízení legislativy povinné zdokumentovat, zmínit dosažené výsledky a pokud je to možné, je potřeba tato rizika eliminovat, nebo alespoň snížit jejich následky. Posouzení rizik je ale obsáhlý proces a přesahuje rámec této práce. Proto v této podkapitole pouze přiblížím základní principy, jak se u takového rozboru postupuje, a zmíním některé konkrétní normy, které se aplikují při posuzování robotizovaných pracovišť. Nakonec zmíním bezpečnostní prvky, které jsou využity v rámci buňky zpracovávané v praktické části a na které bude potřeba brát ohled z hlediska tvorby řídicího programu. [30]

3.4.1 Hierarchie norem EU

Při samotném posuzování rizik se vychází ze standardů popsanych v příslušných normách. Evropské normy jsou rozděleny do tří úrovní a jejich převzetí je pro Českou republiku jako člena EU povinné. Normy typu A se označují jako Základní bezpečnostní normy a v našem případě obsahují úvod do problematiky bezpečnosti, která se dá aplikovat na všechny stroje. Jako Skupinové bezpečnostní normy nazýváme druhou úroveň, která je označována jako typ B. Normy typu B se dále dělí na normy B1, které se zabývají strojem z pohledu bezpečnosti neovlivnitelné obsluhou, a normy B2, které řeší již konkrétní bezpečnostní ovládací prvky. Nejspecifikovanější skupinou norem jsou normy typu C, které pojednávají o bezpečnostních požadavcích na konkrétní stroje. Pokud si normy v něčem rozporují, mají přednost ty z kategorie C. Příklad výše popsané hierarchie je vidět na obrázku Obr. 21, kde je popsána struktura bezpečnostních nároků pro aplikace s průmyslovými roboty. [31]

Možnosti při výběru ochranného zařízení a příslušné normy



Obr. 21) Grafická reprezentace rozdělení norem typu A, B a C [32]

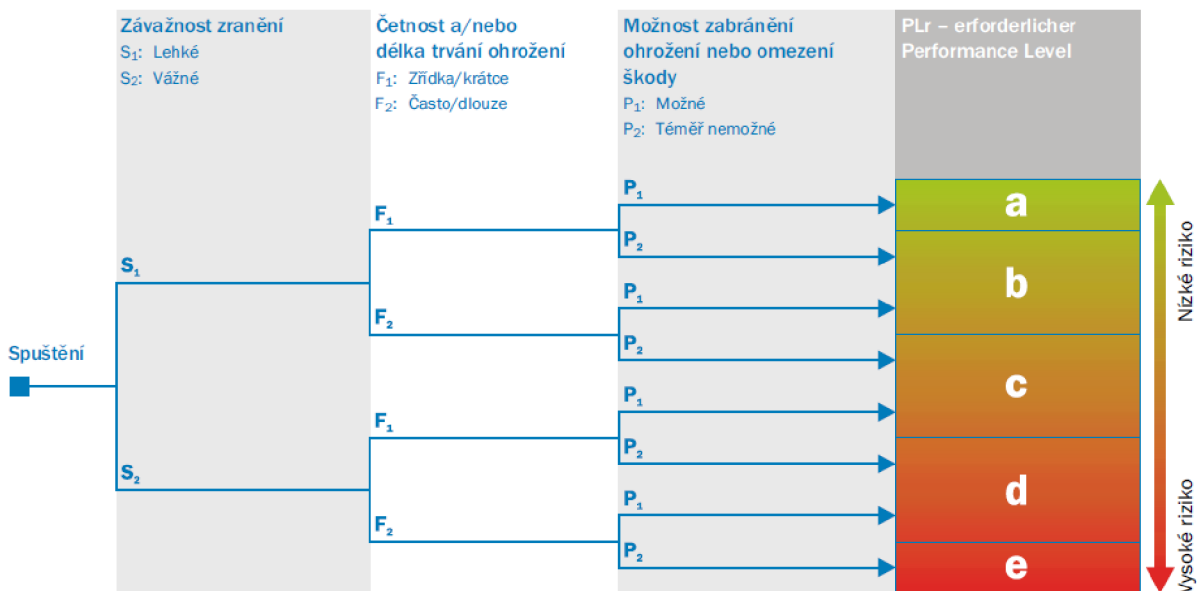
3.4.2 Úroveň zabezpečení

Při identifikaci nebezpečí v rámci analýzy je nutné každému zjištěnému riziku přidělit úroveň potřebnou pro jeho zabezpečení, která závisí jednak na závažnosti zranění, četnosti a délce trvání ohrožení a také na tom, zda je obsluha stroje schopna danému nebezpečí zabránit. Pro stanovení úrovně zabezpečení se používají tzv. rizikové grafy. Výstupem těchto grafických pomůcek je Performance Level, což je minimální požadovaný stupeň ochrany, který stanovíme dle výše zmíněných tří kritérií posuzovaného rizika (viz Obr. 22). [33]

3.4.3 Normy pro robotizovaná pracoviště

Pro stanovení bezpečnosti robotické buňky by měly být zohledněny minimálně následující normy:

- **ČSN EN ISO 12100** – Jedná se o základní normu (typ A) k dosažení bezpečnosti u strojních zařízení.
- **ČSN EN ISO 10218** – Specifikuje bezpečnost pro průmyslové roboty, kdy její první část (ČSN EN ISO 10218-1) popisuje samotnou konstrukci průmyslových robotů a druhá část (ČSN EN ISO 10218-2) blíže specifikuje jejich systémy a integraci.
- **ČSN EN ISO 13850** – Detailně rozebírá zásady funkce nouzového zastavení.
- **ČSN EN ISO 13855** – Určuje místo bezpečnostních zařízení v závislosti na rychlosti přiblížení lidského těla.
- **ČSN EN ISO 13857** – Stanovuje bezpečné vzdálenosti k zamezení dosahu nebezpečných míst horních a dolních končetin.
- **ČSN EN ISO 13849-1** – Popisuje bezpečnostní části ovládacích systémů. [31]

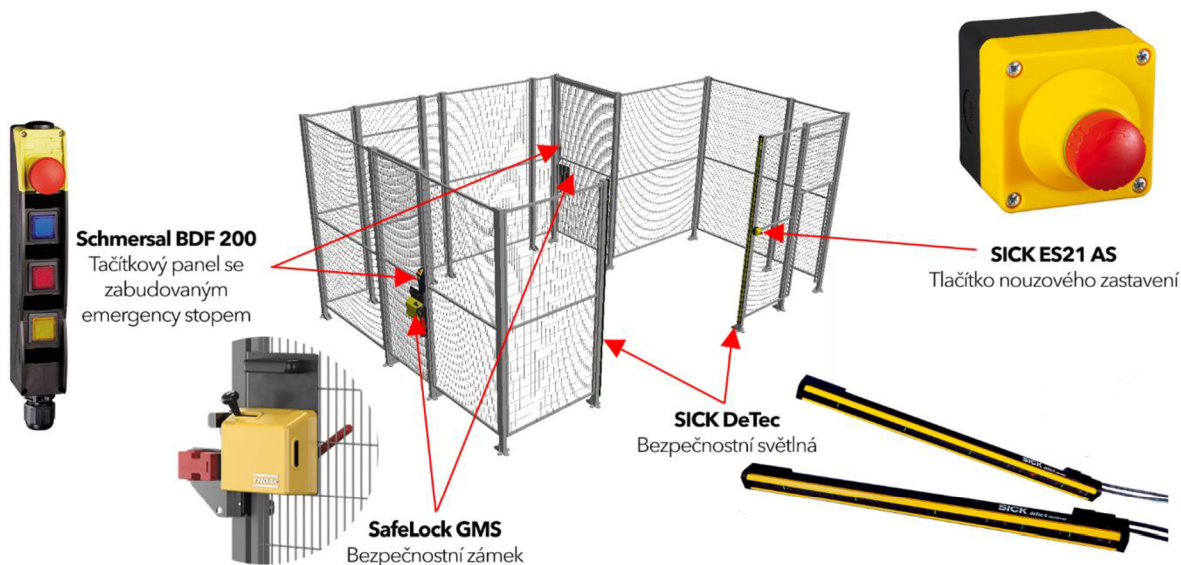


Obr. 22) Rizikový graf z normy ČSN EN ISO 13849-1 [33]

3.4.4 Bezpečnostní prvky v rámci zadané robotizované buňky

Vstupu do pracovního prostoru buňky je primárně zamezeno drátěným oplocením Troax ST 20 s velikostí otvorů 20x100 mm a celkovou výškou 2200 mm. Plot má otvor pro jednoduchý přístup k robotickým kontrolérům a elektrorozvaděči, do kterého je zabudovaný i HMI panel.

Za účelem doplňování polotovarů a balicího materiálu do stohovacího zásobníku není oplocení zcela uzavřeno. Aby obsluha nevstoupila do pracoviště tímto průchozím místem, nachází se zde bezpečnostní světelné závory Sick DeTec s rozlišením paprsku 30 mm a výškou ochranného pole 2100 mm. Pro regulérní vstup do pracoviště slouží dvojce křídlové dveře opatřené zámkem SafeLock. Tento bezpečnostní zámek je opatřen snímači, které detekují, zda jsou dveře otevřené, nebo zavřené. Signál z těchto snímačů se posílá do nadřazeného bezpečnostního PLC, které nedovolí buňku spustit v automatickém režimu, protože při stavu s otevřenými dveřmi není splněna požadovaná bezpečnost. Pokud ale buňka není v provozu, je bezpečnostní zámek odemčen a obsluha může vstoupit do pracoviště. Aby při vstupu do nebezpečných prostor nedošlo k zavření dveří a opětovnému spuštění systému v automatickém režimu, je dveřní systém osazen mechanickým zámkem, který zamezí samovolnému zavření dveří. Navíc je z vnitřní strany další červená bezpečnostní klika, kterou je možné otevřít dveře za jakékoliv situace a z nebezpečného prostoru se dostat ven. U obou dveří je tlačítkový panel sloužící ke znovuspuštění celé buňky po dokončení servisního zákroku (vstupu do buňky), aby obsluha mohla spustit výrobu hned po opuštění pracoviště bez přístupu k HMI panelu, který je od obou dveří vzdálený. Součástí tohoto tlačítkového panelu je hlavně i E-Stop, který slouží u dveří jako bezpečnostní doplněk. Tlačítko nouzového zastavení je instalováno u vstupu pro doplnění polotovarů, kde se nachází i světelná závora. Zde je tlačítko samostatně bez dalších ovládacích prvků, protože nedaleko tohoto místa má obsluha k dispozici již zmiňovaný HMI panel celého řídicího systému. Popis a umístění bezpečnostních prvků znázorňuje Obr. 23.



Obr. 23) Popis bezpečnostních prvků použitých v zadané výrobní buňce

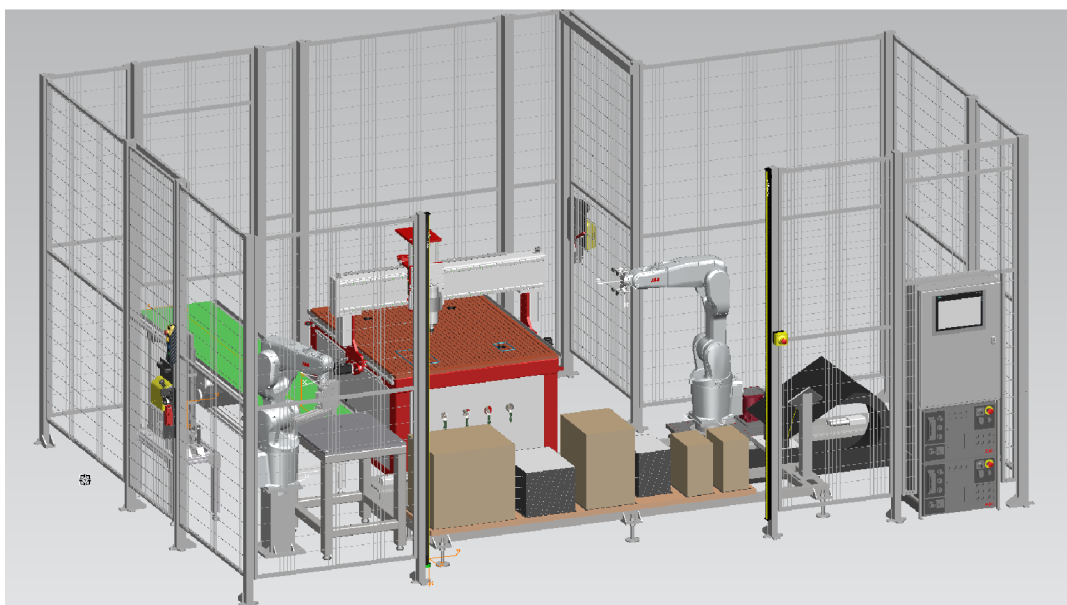
4 PRAKTICKÁ ČÁST

V praktické části jsem s využitím programů vybraných v předchozích kapitolách postupoval za účelem virtuálního zprovoznění. To znamená, že v první řadě jsem musel sestavit simulační model tak, aby co nejvíce odpovídal realitě. Po přípravě modelu a tvorbě robotických operací jsem mohl začít tvořit program nadřazeného PLC. Při samotném programování jsem již propojil 3D model pracoviště a virtuální programovatelný automat tak, aby bylo možné ladit program již po částech kódu. Můžeme tak říct, že řízení již v této pasáži bylo částečně virtuálně zprovozněno. Po tvorbě hlavní struktury programu byl vytvořen ovládací panel dotykové obrazovky, který bývá často označován pojmem HMI. Pak následovala dlouhá část úprav a ladění, aby chování buňky odpovídalo požadavkům. Nad rámec práce byla vytvořena také programová část tzv. Safety PLC, které řídí bezpečnostní prvky a musí splňovat speciální normy a předpisy. Přestože se v praktické části jednotlivé úkony výrazně prolínaly, tak za účelem lepší čitelnosti budu popisovat jednotlivé procesy tvorby tak, aby odpovídaly logickému řazení kroků nutných k vytvoření virtuálního zprovoznění.

4.1 Tvorba simulačního modelu

Kompletní sestavení simulačního modelu zadané buňky v softwaru Process Simulate (dále jen PS) zahrnovalo velký počet dílčích kroků. Abych se neopakoval a text zbytečně nenabýval na rozsahu, pokusím se zmínit jen ty nejdůležitější typizované postupy, které jsem použil.

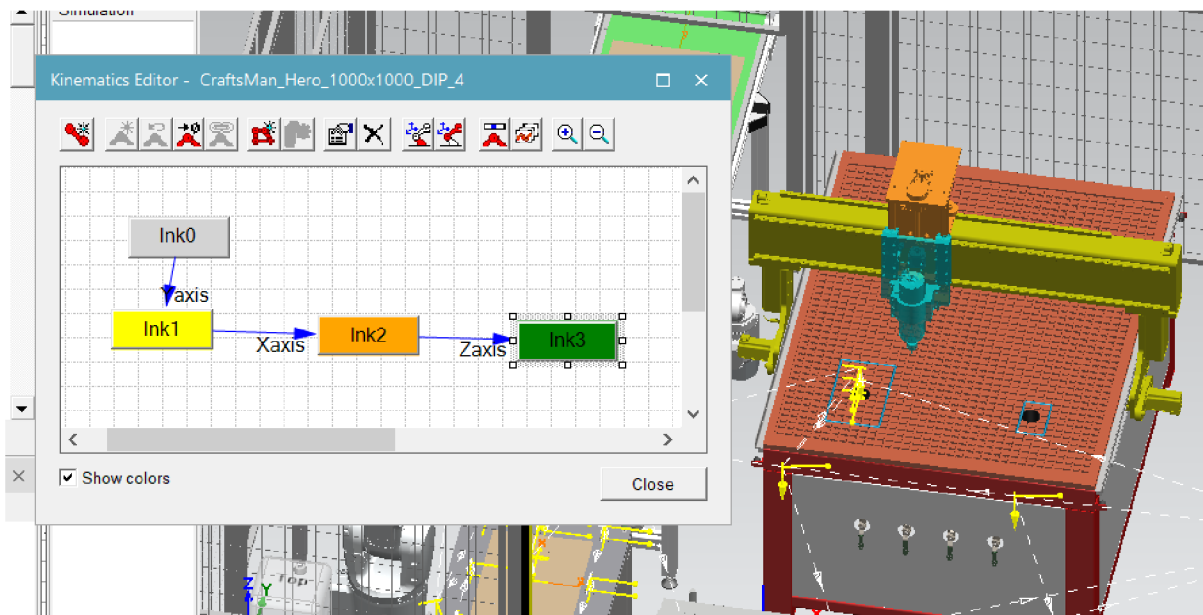
V první řadě jsem sestavil navržený layout buňky. 3D modely je možné do PS importovat pouze ve formátu JT. Většina výrobců nakupovaných komponent jako jsou například dopravníky, pneumatické prvky apod, nabízí své produkty ke stažení nejčastěji ve formátu STEP a jemu podobných, ale formát JT to bývá jen výjimečně. Abych získal všechny modely pracoviště v tomto kompatibilním formátu použil jsem program Autodesk Inventor, který umožňuje exportovat modely v různých formátech. Navíc jsem pomocí tohoto 3D modeláře upravoval a vytvářel vlastní komponenty, kterými byly gripper robotu, zásobník polotovarů nebo CNC router. Buňka po výsledném sestavení v PS je zobrazena na Obr. 24.



Obr. 24) Výsledný simulační model po sestavení v PS

4.1.1 Nastavení pohybů (kinematika)

Pokud chceme jednotlivé komponenty rozhybat, tak je nejprve nutné daný pohyb nastavit v „Kinematics Editoru“. Pro úpravu dané komponenty je nutné ji vždy nejdříve dostat do tzv. rozmodelovaného stavu. To se provede tak, že se vybere pomocí tlačítka myši a v záložce „Modeling“ se zvolí „Set Modeling Scope“. Následně je možné po otevření kinematického editoru tvořit a upravovat definice pohybů. V rámci této práce jsem takto sestavoval například pohyb pro CNC router. Jelikož tento stroj pracuje v kartézské soustavě souřadnic, tak byl zvolen translační pohyb pro všechny osy. Kinematika stroje je sériová, tak i jednotlivé vazby jsou řazeny za sebou (viz Obr. 25).



Obr. 25) Definice pohybu CNC routeru v Kinematics Editoru v PS

Podobným způsobem jsem nastavil například i otvírání dveří v modelu oplocení, kde jsem na rozdíl využil vazby rotační, protože se jedná o křídlové dveře s panty. Nejdůležitějším pohybovým prvkem v rámci buňky jsou samozřejmě průmyslové roboty. Ty byly staženy v již připravené verzi pro PS včetně nastavených kinematik ze stránek výrobce. [34]

4.1.2 Nastavení dat pro tvorbu drah robotů

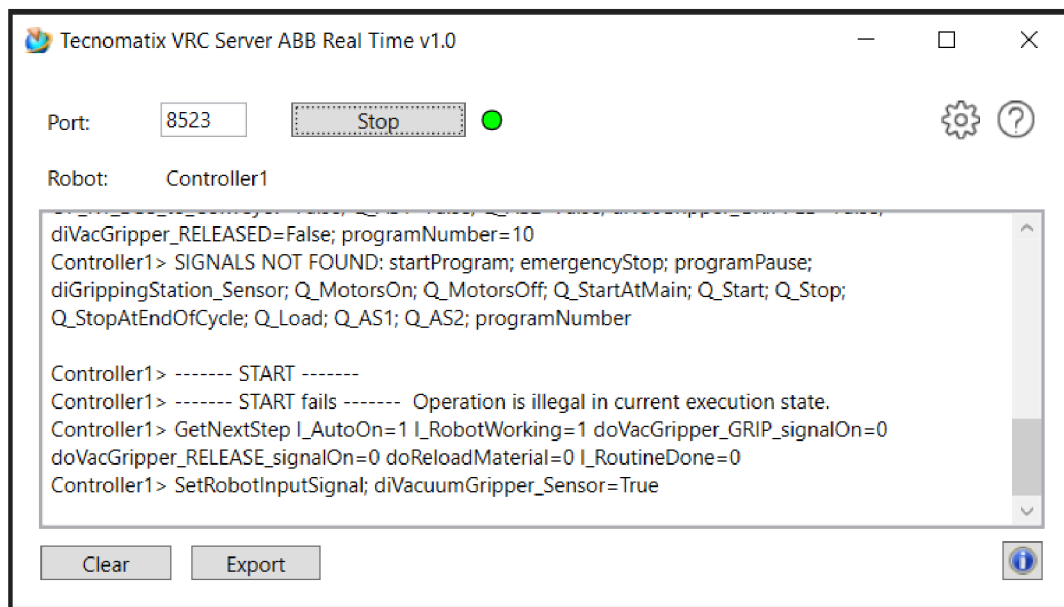
Nastavení kontrolérů robotů

Hlavní přidanou hodnotou off-line programování robotů je příprava pohybových drah a programu robotu ještě před jeho oživením. PS umožňuje vygenerovat tento program pro většinu výrobců robotů. Protože jazyky, podle kterých se roboty řídí, používá každý výrobce vlastní, je nutné mít doinstalované balíčky v rámci PS, aby získaný kód pro řídicí systém robotu odpovídal syntaxi jazyku výrobce. Pokud daný doplněk máte již nainstalovaný, je možné po otevření „Controller Settings“ přiřadit všem robotům z dané studie odpovídající verzi kontroléru.

O řízení simulace robotu na straně PS se můžou starat různé typy virtuálních kontrolérů. Já jsem v práci použil základní variantu „Motion Planner“ (dále jen MOP). Ale je možné využít další dvě varianty, které se svým chováním více blíží reálnému kontroléru.

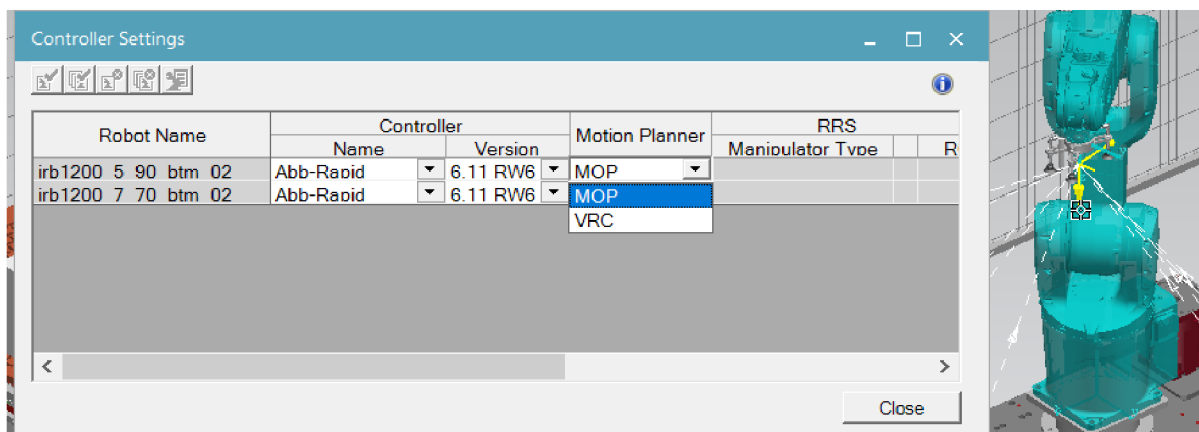
Virtual robot controller (VRC) – Jedná se o novinku od verze 16. Řízení simulace robotu je řešeno napojením na spuštěný kontrolér v simulačním programu od daného výrobce robotů. V mém případě se jednalo o program RobotStudio firmy ABB.

Pro spojení programů RobotStudio a PS je potřeba doinstalovat VRC ABB Server, který je možné bezplatně stáhnout ze stránek Siemens v části podpory, na kterou je možné se přihlásit pouze s platnou licencí. Ale po mailové komunikaci s obchodními zástupci firmy Axiom Tech mi potřebný software byl přeposlán. Ke zprovoznění musíme mít tedy otevřeny všechny 3 softwary zároveň. Na straně RobotStudia stačí, aby virtuální kontrolér byl spuštěný, což je detekováno zelenou barvou v pravé spodní části informační lišty. VRC ABB Server spustíme tlačítkem „Start“ v dialogovém okně (viz Obr. 26), které otevřeme spuštěním souboru „TecnomatixVrcServerAbbRealTime.exe“ v instalační složce programu.



Obr. 26) Dialogové okno softwaru VRC ABB Server pro propojení PS s RobotStudiem

Na straně PS je potřeba zajistit 3 věci. Nejprve je nutné vybrat variantu VRC z nabídky nastavení kontroléru (viz Obr. 27). Dále se název komponenty robotu musí jmenovat, přesně jako se jmenuje virtuální kontrolér v RobotStudiu. Připojení PS k VRC serveru je vhodné ověřit v nabídce „Robot Setup > VRC Connection“, kde je možné změnit i port komunikace serveru. Na závěr je nutné mít PS v režimu Line Simulation, protože pouze v tomto módu VRC komunikace funguje. Následně se klasickým spuštěním simulace robot připojí ke kontroléru v RobotStudiu a začne vykonávat operace v závislosti na jeho robotickém programu. Přenos robotických signálů mezi oběma programy probíhá v pravidelných intervalech a signály se párují podle svého názvu.



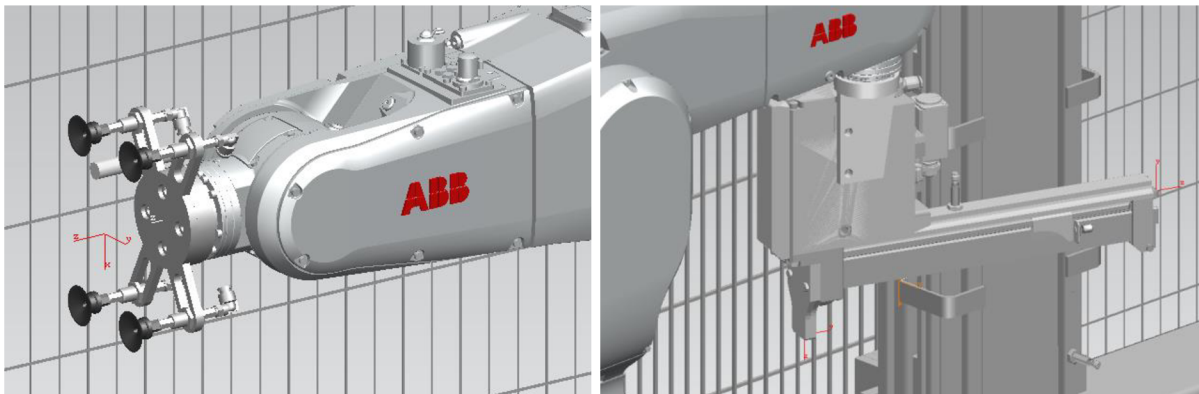
Obr. 27) Nastavení verze kontroléru robotu pro simulaci v PS

Tento typ řízení kontrolérů jsem použil při úpravě a testování robotických programů získaných z PS. Osobně v tomto způsobu řízení vidím velký potenciál, protože při správném nastavení signálů je možné eliminovat úpravy na straně PS pro realistické řízení z nadřazeného PLC, které jsou nutné při použití základní varianty řízení kontrolérů MOP, která byla používána i v této práci.

Realistic Controller Simulation (RCS) – Doposud se jednalo o jedinou možnost dosažení přesné simulace robotu. RCS modul obsahuje virtuální ovladač skutečného robotu vyvíjený přímo daným výrobcem robotů. Pro jeho použití je nutné zakoupit nákladnou licenci.

Definice souřadných systémů pro práci s robotem

Po přiřazení verze kontroléru jednotlivým robotům je možné v nastavení každého z nich definovat souřadné systémy potřebné pro tvorbu robotických operací. Primárně se jedná o Tool Data, ve kterých je definován souřadný systém TCP (Tool Center Point) koncového efektoru robotu. Umístění těchto systémů je znázorněno na Obr. 28, kdy balicí robot má definovány dva souřadné systémy nástroje, protože má pracovní TCP pro operaci sešívání a TCP určené pro doplňování zásobníku.



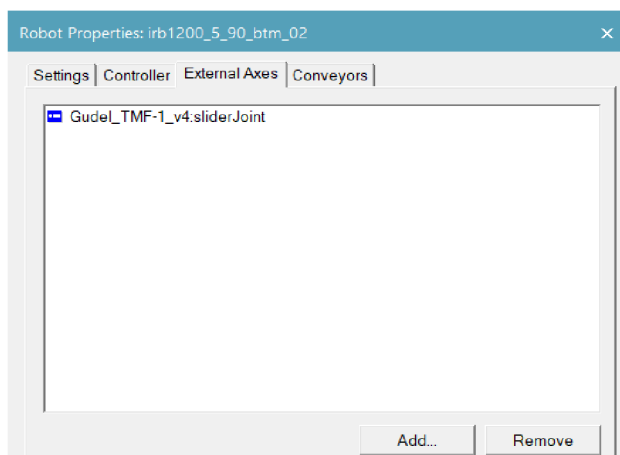
Obr. 28) Souřadné systémy nástroje manipulačního (vlevo) a balicího robotu (vpravo)

Mezi další souřadné systémy používané při programování robotů ABB jsou tzv. Wobj. Jejich použití na rozdíl od Tool Dat není povinné, ale výrazně ulehčuje oživení robotů na místě realizace a případné úpravy v layoutu pracoviště. Principiálně jsou Wobj ekvivalentem souřadných systému obrobku (WCS) u CNC strojů. S jejich pomocí se definuje počátek a orientace pracovní plochy, kterou může být třeba nakloněná rovina pro přechycení v této práci. Následně je možné body robotických drah, které jsou definovány v závislosti na této pracovní ploše, přiřadit k danému Wobj. Díky tomu stačí při změně polohy pracovní plochy v rámci celé buňky pouze znovu přeměřit souřadný systém Wobj, a protože vztah mezi ním a jednotlivými body trajektorie robotu zůstává stejný, celá pracovní dráha se posune s přemístěním Wobj.

V této práci jsem se snažil využít co nejvíce těchto pracovních souřadných systémů. Aby zaměření počátečního bodu v praxi probíhalo co nejjednodušeji, vybral jsem ostré rohy pracovních objektů. Pro manipulační robot se konkrétně jednalo o Wobj vztahované k zásobníku materiálu, CNC routeru, dopravníku a sponkovací matrice. U balicího robotu se jednalo o automatickou doplňovací stanici a také o sponkovací matici, která ale měla počátek pracovního souřadného systému v jiné části oproti manipulačnímu robotu tak, aby se zaměřovací bod nacházel v jeho dosahu.

Definice externí osy robotu:

Pro napojení 7. osy manipulačního robotu bylo nutné nejdřív vytvořit pohyb lineárního pojezdu pomocí „Kinematics editoru“, jak je popsáno v kapitole 4.1.1. Zde byla tvorba vazeb značně zjednodušená, protože zařízení koná lineární pohyb pouze v jedné ose. Po upnutí manipulačního robotu na posuvný jezdec sedmé osy přes příkaz „Attach“ bylo ještě potřeba řídicí systém nastavit tak, aby mohl ovládat pohyb pojezdu a bylo tak možné sedmou osu řídit z robotu. To se provede v záložce „Robot Properties > External Axes“ přiřazením dané komponenty s pohybovou vazbou (viz Obr. 29).



Obr. 29) Přiřazení rozšiřující osy robotu

4.1.3 Tvorba robotických operací

Před samotnou tvorbou trajektorií jednotlivých robotických operací jsem si nejdřív vytvořil další pomocné souřadné systémy pro lokace bodů Pick and Place jednotlivých manipulací. Po prvotním sestavení drah následovala jejich optimalizace včetně přiřazení správných atributů jako jsou Tool Data, Wobj, Speed Data (rychlosti pohybů), Zone Data (přesnost najetí do bodu trajektorie) a přiřazení OLP příkazů (viz Obr. 30). Za zkratkou OLP se skrývá off-line programování a těmito příkazy se v PS přiřazuje další funkcionalita k jednotlivým bodům dráhy robotu. Typicky se jedná o příkazy pro sevření čelistí gripperu, čekání jak časové, tak i na změnu signálu a mnohé další. Počet nativních OLP příkazů se s každou novou verzí rozšiřuje. Stejně tak po přiřazení kontroléru robotu (viz 4.1.2) se nabídka těchto příkazů rozšíří o další instrukce daného jazyku robotů, v mém případě se jednalo o jazyk Rapid.

Paths & Locations	OLP Commands	Conf...	Config	Moti...	Spee...	Tool Data	Wobj Data	Zone...	Ext1
[-] R1_DB1_to_CNC									
[-] DB1toCNC_10		✓	0,0,0,0	MoveJ	v2000	VacuumG...	wobj_zaso...	z100	1970 mm
[-] DB1toCNC_20		✓	1,0,0,0	MoveL	v1500	VacuumG...	wobj_zaso...	z40	1970 mm
[-] DB1toCNC_30_p...	# Destination Sucti	✓	1,0,0,0	MoveL	v100	VacuumG...	wobj_zaso...	fine	1970 mm
[-] DB1toCNC_20		✓	1,0,0,0	MoveL	v300	VacuumG...	wobj_zaso...	z10	1970 mm

Obr. 30) Příklad tvorby drah pohybů robotu v Path Editoru

OLP Commands

Z již zmíněných OLP příkazů jsem konkrétně využil příkazy „PulseDO“ pro vystavení signálu pro tvorbu vakua pneumatických přísavek gripperu prvního robotu a podobně signál pro jejich zavzdušnění při uvolnění manipulovaného materiálu. U obou případů vystavení toho signálu je použita hned druhá instrukce „WaitDI“, která jak název napovídá, čeká, až digitální vstup nabude definované hodnoty. V tomto případě se čeká na vstupní hodnoty z tlakového senzoru ejektory, který sleduje odpovídající hladiny tlaku vzduchu. I u druhého robotu je vystavován signál pomocí instrukce „PulseDO“ pro řízení pneumatického obvodu sešívací pistole. Ale zde není řešena zpětná vazba, takže místo čekání na vstupní signál je použita instrukce „WaitTime“ po dobu 0,5 s. Podobně jsou tyto robotické signály v rámci celé buňky použity například i u nakloněné roviny sloužící pro přechycení, kde senzor detekuje skluz

deskových materiálů do spodní polohy. Manipulační robot zde kvůli použití funkce „*WaitDI*“ čeká na náběžnou hranu signálu z tohoto senzoru.

Custom OLP Commands

Off-line robotické příkazy v rámci PS mají 3 úrovně nebo lépe řečeno vrstvy, které mohou ovlivňovat. Když se nad tím zamyslíme, tak se jedná vcelku o logickou věc. Použití „*OLP commands*“ chceme jednak vypsát v „*path editoru*“, kde se jednotlivé příkazy přiřazují k bodům trajektorii (viz Obr. 30). V druhé řadě chceme ovlivnit simulaci. A v neposlední řadě chceme docílit toho, aby se propsaly i do robotického programu při jeho stažení ve správné syntaxi jazyka kontroléru tak, aby nebylo potřeba kód před jeho reálným zprovozněním již upravovat.

Velkou přidanou hodnotou PS je to, že nabízí možnost si tyto příkazy doprogramovat vlastní pomocí jazyka XML. To znamená, že je možné paletu základních OLP instrukcí rozšířit dle vlastní potřeby. Kostra XML kódu příkazu odpovídá výše popsaným 3 vrstvám, které jsou v kódu přímo popsány výrazy: „*UI Layer*“, „*Simulation Layer*“, „*Download Layer*“ (viz Obr. 31).

Pro účely virtuálního zprovoznění odpovídající reálnému uvedení do provozu byl získán XML soubor s pracovním názvem „*Program Call*“, ve kterém jsou naprogramovány 3 vlastní obsáhlé OLP příkazy. Každý z nich definuje typizovanou robotickou proceduru:

- **Main** – Tato hlavní procedura spouští inicializaci, a pokud je robot v automatickém režimu, tak volá stavový automat robotu, ve kterém mohou být volány jednotlivé pracovní operace.
- **Init** – Jedná se o inicializační proceduru, ve které jsou resetovány signály „*RobotWorking*“ a „*TaskEnded*“ a případně zde mohou být resetovány i další signály, protože XML kód je na to připravený.
- **Program Flow** – V této proceduře je sestavena dlouhá IF struktura, ve které se kontroluje, zda není z nadřazeného PLC volána nějaká pracovní operace. Pokud je nějaká operace ve skenovací periodě zavolána, tak se nejprve signál „*RobotWorking*“ nastaví na hodnotu „*TRUE*“. Po skončení celé pracovní operace se přenastaví zase na hodnotu „*FALSE*“ a zároveň se po dobu 1 s nastaví další uživatelský signál „*Task Ended*“ na hodnotu „*TRUE*“.

```

<RobotController Name="Abb-Rapid">
  <!--Robotic parameters definition-->
  <RoboticParams>
  </RoboticParams>
  <!--OLP-->
  <OlpCommands>
    <Command Name="Example_OLP_Name">
      <RoboticParamRef>
      </RoboticParamRef>
      <Layers>
        <!--UI Layer is layer where is specified how OLP command in OLP window will look like-->
        <UILayer>
        </UILayer>
        <!--Simulation layer is used to provide commands for PS simulation-->
        <SimulationLayer>
        </SimulationLayer>
        <!--Download layer provides correct Rapid code for robot when PS program
        is downloaded to robot-->
        <DownloadLayer>
        </DownloadLayer>
      </Layers>
    </Command>
  </OlpCommands>
  <!--This part of code is used to add command into PS OLP commands offer-->
  <OlpDialogs>
  </OlpDialogs>
</RobotController>
    
```

Obr. 31) Obecná struktura kódu pro tvorbu vlastního OLP příkazu v jazyku XML

Výše zmíněné programové procedury je nutné zavolat v samostatných robotických operacích a při sestavování programu robotu v PS je potřeba je dát v právě tomto pořadí zcela na začátek, jak je naznačeno na Obr. 32.

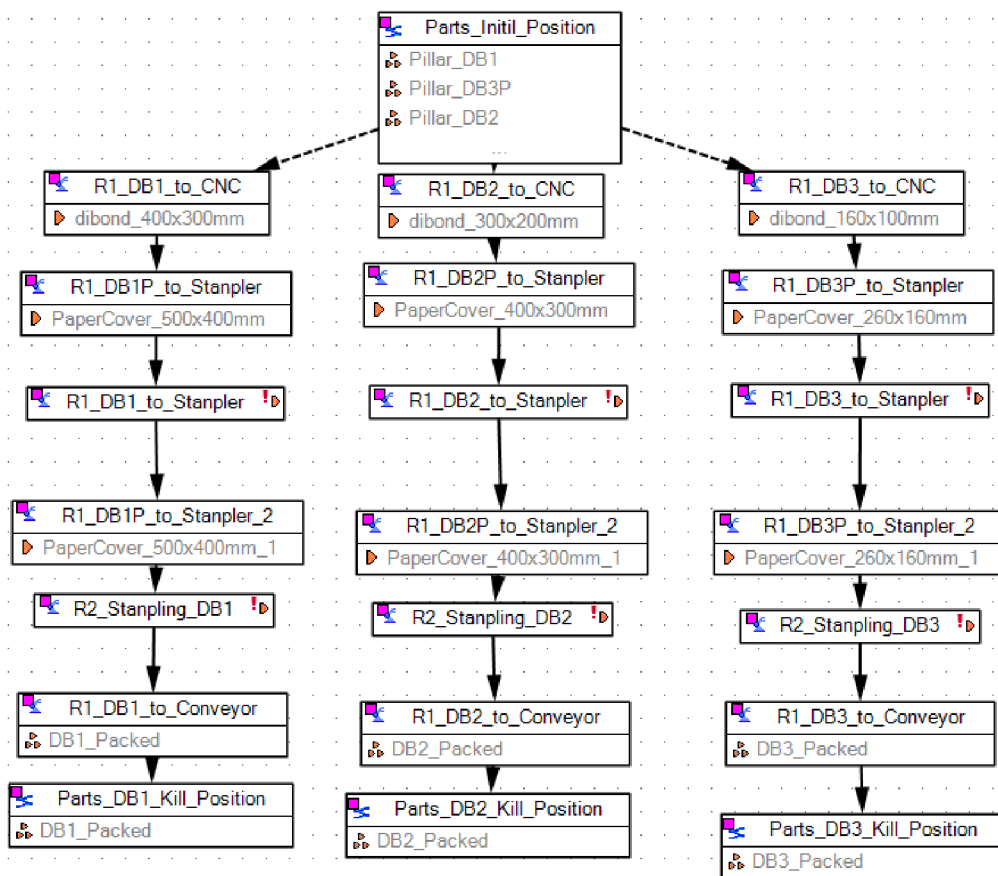
Paths & Locations	OLP Commands
<ul style="list-style-type: none"> [-] Main_R1 <ul style="list-style-type: none"> R1_MainModule R1_Init R1_ProgramFlow R1_DB1_to_CNC R1_DB1P_to_Stanpler R1_DB1_to_Stanpler R1_DB1_to_Conveyor R1_DB1P_to_Stanpler_2 R1_DB2_to_CNC R1_DB2P_to_Stanpler R1_DB2_to_Stanpler R1_DB2P_to_Stanpler_2 R1_DB2_to_Conveyor R1_DB3_to_CNC R1_DB3P_to_Stanpler R1_DB3_to_Stanpler R1_DB3P_to_Stanpler_2 R1_DB3_to_Conveyor 	<ul style="list-style-type: none"> R1_Init; R1_ProgramFlow; # WaitTime 1 SignalsInitialization; # If OP_R1_DB1_to_CNC = high Then SetI_RobotWorkin

Obr. 32) Příklad sestaveného programu z jednotlivých operací manipulačního robotu

Přestože má uživatel možnost tvořit vlastní instrukce simulace robotů, je omezen pouze na příkazy, které jsou do programu PS implementovány. Snažil jsem se v rámci práce vytvořit vlastní příkaz, který by ekvivalentně napodobil pokročilou instrukci „SearchL“ z jazyku Rapid, ale kvůli omezené nabídce příkazu se mi ji nepodařilo simulovat. Z toho důvodu bude nutné před reálným zprovozněním vygenerovaný robotický program dodatečně upravit. Toto téma blíže rozeberu v poslední kapitole (4.5) této práce.

4.1.4 Materiálový tok

Při událostně řízené simulaci, která se v prostředí PS jmenuje „*Line Simulation*“, se produkty generují a zanikají na základě tzv. materiálového toku. Nutno podotknout, že bez jeho nastavení nebude simulace fungovat. Vznik součástí je reprezentován tzv. „*appearance*“, což můžeme volně přeložit do češtiny jako zjev/vzhled. Tento vzhled součásti je možné generovat na začátku operací. Pokud se generují ty stejné součásti v rámci na sebe navazujících operací v materiálovém toku, tak předchozí vždy zmizí, aby v rámci jedné posloupnosti operací nevznikla daná součást vícekrát. Díky tomu můžeme třeba složením více částí v jednom výrobním cyklu vytvořit sestavu, s kterou je dále operováno jako s jednotným dílem, což jsem použil i při programování zadané buňky. Protože cílem tohoto výrobního systému je vyrábět 3 různé velikosti, využil jsem při sestavování toku materiálu paralelní struktury (viz Obr. 33).



Obr. 33) Materiálový tok pro výrobu 3 velikostí gravírovaných desek

4.1.5 Signály robotu

Již při popisování OLP příkazů v kapitole 4.1.3 jsem se zmínil o signálech, kterými je možné řídit robotické operace. V rámci sekce „*OLP Commands*“ je možné používat pouze signály přiřazené konkrétnímu robotu. Tyto signály se vytváří v části „*Robot Signals*“. Příklad tabulky robotických signálů balicího robotu je na Obr. 34. Prvních 8 z nich jsou nativní signály z PS pro řízení simulace, ale tyto signály nekorrespondují s reálným řízením robotů ABB. Proto bylo nutné vytvořit systémové signály dle příručky kontroléru IRC5, kterými je robot ovládán z nadřazeného PLC, a naprogramovat jim funkci v logickém bloku, což popisují v kapitole 4.1.8. Jedná se o signály, které mají v názvu značku PN, což je zkratka pro Profinet, po kterém spolu kontroléry robotů a PLC komunikují, jak bylo uvedeno na Obr. 14. Další skupinou jsou signály vyvolávající jednotlivé operace. A v rámci příkladu tohoto konkrétního robotu jsou

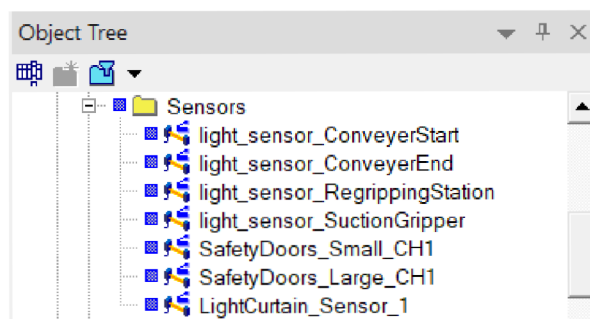
ještě řešeny signály pro ovládání koncového efektoru. V tomto případě se jedná o signál pro vystřelení spony z pneumatické pistole a o signál z automatické doplňovací stanice, na který robot musí čekat, než je proces doplňování dokončen. Podobným způsobem je v rámci práce nastaven i druhý robot.

PLC Signal Name	Robot Signal Name	I/O	Signal Function	HW ...
irb1200_7_70_btm_02_startProgram	startProgram	Q	Starting Program	BOOL
irb1200_7_70_btm_02_programNumber	programNumber	Q	Program Number	BYTE
irb1200_7_70_btm_02_emergencyStop	emergencyStop	Q	Program Emergency ...	BOOL
irb1200_7_70_btm_02_programPause	programPause	Q	Program Pause	BOOL
irb1200_7_70_btm_02_programEnded	programEnded	I	Ending Program	BOOL
irb1200_7_70_btm_02_mirrorProgramNumber	mirrorProgramNumber	I	Mirror Program Number	BYTE
irb1200_7_70_btm_02_errorProgramNumber	errorProgramNumber	I	Error Program Number	BOOL
irb1200_7_70_btm_02_robotReady	robotReady	I	Robot Ready	BOOL
irb1200_7_70_PN_Q_MotorsOn	Q_MotorsOn	Q		BOOL
irb1200_7_70_PN_Q_MotorsOff	Q_MotorsOff	Q		BOOL
irb1200_7_70_PN_Q_Start	Q_Start	Q		BOOL
irb1200_7_70_PN_Q_StartAtMain	Q_StartAtMain	Q		BOOL
irb1200_7_70_PN_Q_Stop	Q_Stop	Q		BOOL
irb1200_7_70_PN_Q_StopAtEndOfCycle	Q_StopAtEndOfCycle	Q		BOOL
OP_R2_Stanpling_DB1	OP_R2_Stanpling_DB1	Q		BOOL
OP_R2_Stanpling_DB2	OP_R2_Stanpling_DB2	Q		BOOL
OP_R2_Stanpling_DB3	OP_R2_Stanpling_DB3	Q		BOOL
OP_R2_StanplerReloading	OP_R2_StanplerRelo...	Q		BOOL
OP_R2_Go2Home	OP_R2_Go2Home	Q		BOOL
irb1200_7_70_Q_AS1	Q_AS1	Q		BOOL
irb1200_7_70_Q_AS2	Q_AS2	Q		BOOL
irb1200_7_70_PN_I_RunChainOK	I_RunChainOK	I		BOOL
irb1200_7_70_PN_I_AutoOn	I_AutoOn	I		BOOL
irb1200_7_70_PN_I_CycleOn	I_CycleOn	I		BOOL
irb1200_7_70_PN_I_MotorsOnState	I_MotorsOnState	I		BOOL
irb1200_7_70_PN_I_MotorsOffState	I_MotorsOffState	I		BOOL
irb1200_7_70_PN_I_RobotWorking	I_RobotWorking	I		BOOL
irb1200_7_70_PN_I_RoutineEnded	I_RoutineEnded	I		BOOL
PN_I_MagazineReloaded	I_MagazineReloaded	I		BOOL
Local_Q_Tool_BeA_Stanpler_SHOOT	Local_Q_Tool_BeA...	I		BOOL
irb1200_7_70_btm_02_at_HOME	HOME	I	Pose Signal	BOOL
irb1200_7_70_btm_02_at_Homed	Homed	I	Pose Signal	BOOL

Obr. 34) Všechny signály balicího robotu z praktické části

4.1.6 Senzory

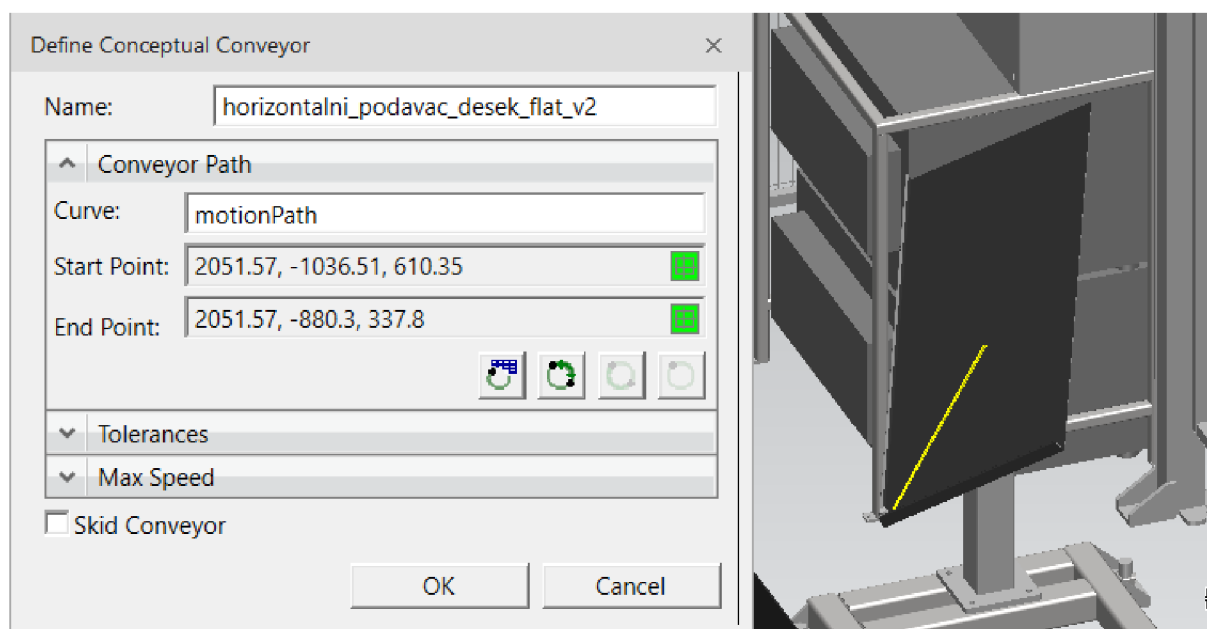
Když se bavíme o signálech, tak nesmíme zapomenout na senzory. V rámci buňky jsem jich použil hned několik. Všechny jsou typu „Photoelectric Sensor“ z nabídky PS (viz Obr. 35). Konkrétně se jedná o senzory na začátku a konci výstupního dopravníku, na gripperu s přísavkami, na spodní části plošiny určené pro přechycení a taky spínače detekující, zda je splněna bezpečnost. To znamená, zda jsou zavřené dveře a není narušena oblast světelné závory. Signály z těchto snímačů pak byly použity v logických blocích popsaných v kapitole 4.1.8.



Obr. 35) Použité senzory v rámci simulace

4.1.7 Dopravníky

Komponentám, které jsou při importování nadefinovány jako dopravník, je možné v PS jednoduše přiřadit funkci dopravování. Za tímto účelem je nejprve nutné udělat křivku, která slouží jako trajektorie cesty. Pak již stačí vybrat danou komponentu a zvolit „Define Conveyor“ v záložce „Control“. Já jsem logicky tuto funkci použil u výstupního dopravníku, ale navíc i u nakloněné plošiny, aby součásti sklouzly do nejnižšího bodu. Původně jsem zamýšlel k tomuto účelu použít novou funkci „Physics“ dostupnou od verze 16, kde je možné nasimulovat gravitaci a další základní fyzikální chování, ale protože se zatím jedná o BETA verzi, tak jsem při jejím použití objevil velké nedostatky, mezi něž patří například výrazné zpomalení práce s PS. Proto jsem nadefinoval nakloněnou plošinu také jako dopravník (viz Obr. 36), který je v simulaci neustále v provozu a není ho tedy nutné spouštět z nadřazeného řízení. Ještě bych dodal, že aby byly součásti po dopravníku transportovány, je nutné je definovat jako „Conveyable Part“. Pro možnost řídit výstupní dopravník z nadřazeného PLC jsem vytvořil z jeho komponenty logický blok, kdy se po jeho vygenerování automaticky vytvoří signály pro spuštění a zastavení pohybu jeho pásu, které jsem následně propojil na nadřazené řízení.



Obr. 36) Nastavení dopravníku u stanice pro přechycení polotovarů

4.1.8 Logické bloky

V přechodím textu jsem již několikrát zmínil logické bloky, které v PS slouží k tvorbě vnitřní logiky při simulaci. Nejhodnotnější a nejobsáhlejší logický blok, který jsem použil, byl ten pro simulaci řízení robota. V kapitole 4.1.5 jsem popisoval systémové signály kontroléru IRC5, kterými je robot řízen z PLC. Ale aby signály reagovaly jako v realitě i při simulaci, a tedy i při virtuálním zprovoznění, bylo nutné jejich odezvu naprogramovat v rámci logického bloku. Za tímto účelem byl získán předpřipravený blok, který jsem rozšířil a upravil o další funkce pro použití v této práci. Systém IRC5 obsahuje celkem 24 systémových vstupů a 32 systémových výstupů. Ale pro bezpečné a robustní řízení stačí použít jen některé z nich. Já jsem použil následující:

- **Systémové vstupy:** StartAtMain, Start, Stop, StopAtEndOfCycle, MotorsOn, MotorsOff
- **Systémové výstupy:** RunchainOK, AutoOn, CycleOn, MotorsOnState, MotorsOffState

Navíc jsem v logickém bloku využil některé nativní robotické signály z PS, uživatelské signály „RobotWorking“ a „TaskEnded“, které jsou vystavovány při volání jednotlivých operací, a signál „AutoStop“. Ten je při fyzickém zprovoznění propojen napřímo vodiči z bezpečnostního PLC dvoukanalově na svorkovnici robotického kontroléru. O této části se zmiňuji v kapitole věnované bezpečnostnímu PLC programu (4.2.6). V rámci těchto bloků jsem navíc řešil i logiku signálů ze sensorů, s kterými daný robot interagoval. Příklad vstupních a výstupních signálů, které byly v rámci programování funkčních bloků využity, je znázorněn na Obr. 37.

Name	Type	Connected Signals
RB_I_StartAtMain	BOOL	irb1200_5_90_PN_Q_StartAtMain
RB_I_MotorsOn	BOOL	irb1200_5_90_PN_Q_MotorsOn
RB_I_MotorsOff	BOOL	irb1200_5_90_PN_Q_MotorsOff
RB_I_LoadProgram	BOOL	irb1200_5_90_PN_Q_Load
RB_I_StartRoutine	BOOL	irb1200_5_90_PN_Q_Start
RB_I_StopPauseRoutine	BOOL	irb1200_5_90_PN_Q_Stop
PS_I_ProgramEnded	BOOL	irb1200_5_90_btm_02_programEnded
RB_I_AutoSwitch	BOOL	AutoSwitchOn
PS_I_Sensor_RegripStat	BOOL	light_sensor_RegrippingStation
PS_I_Sensor_VacuumCup	BOOL	light_sensor
RB_I_RobotWorking	BOOL	irb1200_5_90_PN_I_RobotWorking
RB_I_StopAtEndOfCycle	BOOL	irb1200_5_90_PN_Q_StopAtEndOfCycle
RB_I_AS1	BOOL	irb1200_5_90_Q_AS1

Name	Type	Connected Signals
RB_Q_AutoOn	BOOL	irb1200_5_90_PN_I_AutoOn
RB_Q_RunChainOK	BOOL	irb1200_5_90_PN_I_RunChainOK
RB_Q_CycleOn	BOOL	irb1200_5_90_PN_I_CycleOn
RB_Q_MotorsOnState	BOOL	irb1200_5_90_PN_I_MotorsOnState
RB_Q_MotorsOffState	BOOL	irb1200_5_90_PN_I_MotorsOffState
PS_Q_StartRobotProgram	BOOL	irb1200_5_90_btm_02_startProgram
PS_Q_LoadProgramNumber	BYTE	irb1200_5_90_btm_02_programNumber
PS_Q_Sensor_RegrippingStation	BOOL	bQ_PN_Sensor_RegrippingStation
PS_Q_Sensor_VacuumCup	BOOL	local_IO_deVacuumGripper_Sensor
PS_Q_PauseRoutine	BOOL	irb1200_5_90_btm_02_programPause
PS_Q_VacGripper_RELEASED	BOOL	local_IO_diVacGripper_RELEASED
PS_Q_VacGripper_GRIP	BOOL	local_IO_diVacGripper_GRIPPED

Obr. 37) Vstupní (vlevo) a výstupní (vpravo) signály logického bloku robotů ABB

Při sestavování logiky v bloku je možné využít vnitřních proměnných tzv. parametrů (viz Obr. 38). Samotné kódování je ale velice omezené. Obsahuje pouze základní funkce typu Set-Reset, Falling/Rising Edge, Timer On, Timer Off a několik dalších. Proto vývojáři přidali do verze 16 možnost programovat logiku jednotlivých komponent pomocí SCL editoru, kde je možné využít standardizovaný jazyk Strukturovaný text dle normy IEC 61131-3. Pro podobnou komplexní logiku bych tuto variantu doporučil. Ale sám jsem tuto novinku objevil až v pozdější části, tak jsem ji neměl příležitost využít.

Name	Type	Index	Description
AutoOn_P	BOOL	1	
RunChainOK_P	BOOL	2	
MotorsOnState_P	BOOL	3	
ProgramLoaded_P	BOOL	4	
CycleOn_P	BOOL	5	
PauseProgram_P	BOOL	6	
StopAtEndOfCycle_P	BOOL	7	

Value Expression

```
SR((RE(RB_I_StartAtMain) AND MotorsOnState_P) OR (RE(RB_I_StartRoutine) AND MotorsOnState_P),RB_I_StopPauseRoutine OR RB_I_MotorsOff OR (StopAtEndOfCycle_P AND RE(PS_I_ProgramEnded))) OR FE(AutoOn_P))
```

Obr. 38) Ukázka vnitřních proměnných tzv. parametrů logického bloku řízení robotů

4.2 Tvorba PLC programu

Při programování PLC jsem primárně používal jazyk SCL (Structured Control Language), což je textový jazyk firmy Siemens, který vyhovuje standardům Strukturovaného textu (ST) definovaného normou IEC 61131-3. Při sestavování logiky většiny funkčních bloků jsem využil filozofie stavového automatu, proto mi volba SCL přišla jako nejvhodnější varianta. U bloků s jednodušší logikou jsem použil grafický jazyk Ladder Diagram, který je také definován výše zmíněnou normou.

4.2.1 OB Main – Hlavní rutina

Organizační blok „*Main*“ je automaticky volán při spuštění PLC. Konkrétně se jedná o tzv. „*Program Cycle*“ organizační blok, což znamená, že je automaticky volán v každé skenovací periodě PLC. Při jeho sestavování jsem ve struktuře nevytvářel žádnou logiku a pouze volám další vytvořené uživatelské bloky a funkce, které popíšu v následujících podkapitolách s využitím vývojových diagramů pro lepší názornost algoritmizace jednotlivých úloh.

4.2.2 FB Robot Control – řízení robotu z PLC

První funkční blok, který jsem naprogramoval, byl pro řízení robotu (viz Obr. 39), ve kterém řeším jeho ovládání a volání procedur z nadřazeného PLC. Tato část kódu je potřeba pro dosažení dostatečné robustnosti, aby se nestalo to, že zavolám určitou robotickou operaci nebo proceduru, ale robot bude stále ještě vykonávat nějakou předchozí. To by mohlo vyvolat nežádoucí stav. Proto jsem při sestavování tohoto bloku využil již zmiňovanou logiku stavového automatu, kdy na základě vstupních podmínek mohu vyvolat pouze určitý stav.

Na začátku kódu se kontroluje, zda má robot splněnou bezpečnostní podmínku „*RunChainOK*“ a zároveň je přepnutý do automatického režimu, což je detekováno systémovým výstupem „*AutoOn*“. Tato podmínka je kontrolována bezpodmínečně při každé skenovací periodě. Pokud jsou tyto dvě nutné podmínky splněny, tak je v další části ověřováno, zda je robotický cyklus již spuštěný. Pokud zatím není spuštěný, tak mohu volat procedury pro zapnutí nebo vypnutí motorů, nebo je možné spustit vykonávání smyčky robotického programu od jeho začátku nebo z místa, kde byl v předchozí části pozastaven. To nás přivádí k části programu, kdy jsou splněny podmínky, že tento cyklus již probíhá. Zde funkční blok umožňuje volat proceduru pro okamžité zastavení nebo pro zastavení po dokončení jednoho právě probíhajícího cyklu. Ale hlavně v této části kódu mohu volat jednotlivé pracovní operace, pokud zrovna robot nějakou již nevykonává. Číslo operace je vybráno na základě struktury SWITCH-CASE dle hodnoty proměnné „*R_opNum*“ a celkový počet případů odpovídá počtu robotických operací, které má robot naprogramované. Ve vývojovém diagramu na Obr. 39 je to naznačeno pomocí čárkované šipky a rozhodovacího symbolu s textem „*CASE:n*“.

4.2.3 FB Cell Control – řízení robotizované buňky

Funkční blok „*Cell_Control*“ má na starost řízení celé buňky a všech možných stavů, které by mohly nastat. Kód je rozdělen do 3 logických částí, jak je znázorněno pomocí šrafovaných obdélníků na Obr. 40. První část kódu běží bezpodmínečně a kontroluje, jestli nebyly porušeny některé základní podmínky 4 stavů buňky, které jsou důležité pro její řízení. Konkrétně se jedná o stav, zda jsou nahozeny pohony, jenž je reprezentován proměnnou „*CELL_IsReady*“. Druhý z nich kontroluje, pokud je buňka v provozu a roboty jsou v automatickém cyklu. A zbylé dvě podmínky zjišťují, zda není potřeba doplnit automatickou stanici se sponami nebo polotovary v zásobníku.

Druhá část kódu probíhá pouze v případě, pokud je splněna celková bezpečnost a neprobíhá žádná změna stavu řízení buňky. To je reprezentováno proměnnou „*nFB_StateNum*“, která musí být rovna nule. Pokud rovna nule není, tak to znamená, že stavový automat vykonává nějakou sekvenci v rámci řízení buňky, což může být například povel „*Start*“, který vyvolá nahození motorů všech zařízení v rámci pracoviště včetně inicializačního procesu, ve kterém resetuje hlavní proměnné starající se o řízení, a hlavně se nastaví výrobní program, který je pak možné spustit příkazem pro automatický režim. Takže v této druhé části jsou naprogramovány podmínky pro volání základních ovládacích prvků z HMI panelu, jako je již zmíněný „*Start*“, „*Stop*“, „*Pause*“ a „*Continue*“. Příkazem „*Stop*“ se přeruší nekonečný výrobní cyklus, kontrolka na HMI panelu začne hlásit stav „*Cell is Stopping*“ a dokončí výrobu právě rozpracovaných produktů. Až jsou všechny rozpracované polotovary dokončeny, tak se vypne robotický cyklus a pohony všech zařízení v rámci buňky. Příkazem „*Pause*“ se robotický cyklus pouze pozastaví a je možné ho opět spustit ze stejného místa příkazem „*Continue*“. Pro vstup do buňky, ale z bezpečnostního hlediska nestačí pouze zastavit cyklické opakování programu, dokonce nestačí ani vypnout motory robotů, ale je nutné blokovat spuštění automatického režimu. To je zahrnuto v sekvenci „*Enter the Cell*“ a „*Servis Mode*“, které po shození motorů aktivují navíc „*Auto Stop*“ a deaktivují bezpečnostní prvky, čímž umožní vstoupit obsluze do prostor buňky. Rozdíl mezi těmito dvěma vypínacími sekvencemi je v tom, že při spuštění servisního módu je výrobní proces pozastaven okamžitě, aby bylo možné neprodleně provést servisní zásah. Ale v případě volby klasického vstupu je nejprve dokončen právě probíhající cyklus. V rámci ošetření možných stavů jsem implementoval ještě další 3 ovládací sekvence. Jedna z nich je automatické spuštění vypnutí buňky na konci cyklu, pokud je zjištěn nedostatek spon v doplňovací stanici i v pneumatické sponkovačce. Při této vypínací sekvenci je na HMI obrazovce vypsána hláška vyzývající obsluhu pro doplnění spon. Když snímač v doplňovací stanici detekuje, že spony byly doplněny, tak se stav buňky přepne a umožní obsluze znovu spustit výrobu. Podobný princip s vyskakovacími okny s odpovídajícími hláškami o stávajícím stavu je použit i při vstupu do buňky, servisním módu, ale i při nečekaném spadnutí okruhu bezpečnostních prvků nebo v případě, kdy manipulační robot zaznamená, že je v zásobníku nedostatek materiálu v závislosti na aktuální souřadnici osy Z bodu, při kterém proběhlo poslední odebrání stohovaného polotovaru. Tento princip je více rozebrán v kapitole 4.5, ve které popisují dodatečnou úpravu robotického programu a implementaci pokročilých instrukcí. Protože se jedná o jiné stavy s různými hláškami, tak jsou použity jiné sekvence stavového automatu řízení buňky v tomto funkčním bloku.

Poslední část kódu vykonává již zmíněný stavový automat. Také se zde resetují proměnné pro řízení buňky, pokud není splněna bezpečnost, protože i v tomto stavu je možné klikat na tlačítka HMI panelu, což by vyvolalo nechtěný stav po znovuspuštění celé buňky.

na hodnotu 1 a „*NextPartSize*“ na hodnotu 2. Tyto hodnoty reprezentují právě velikosti jednotlivých desek. Číslo 1 představuje největší rozměr a číslo 3 ten nejmenší.

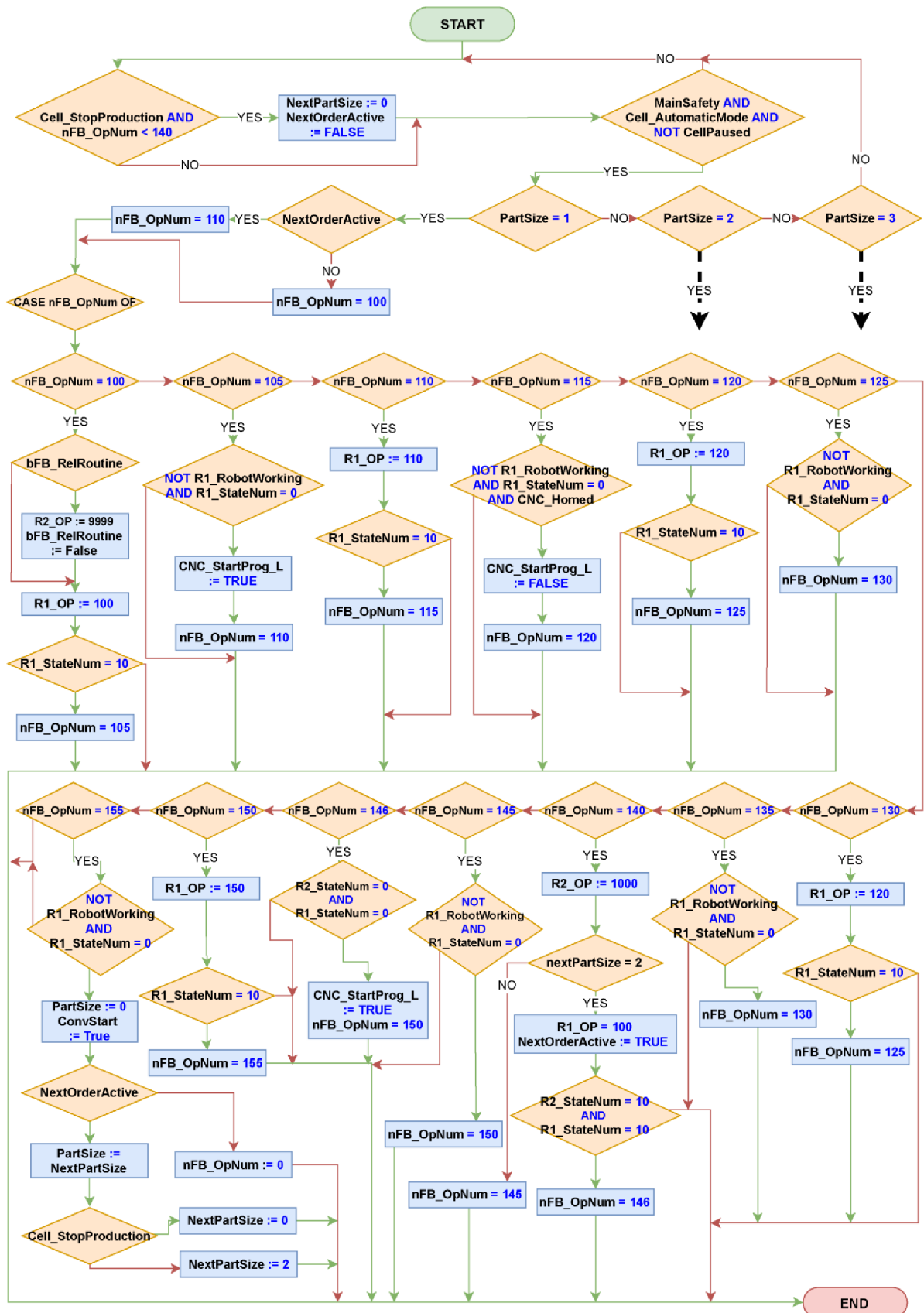
V první části kódu je kontrolována podmínka, zda nebyl vystaven požadavek na zastavení výroby. To znamená, pokud obsluha stiskla tlačítko „*Stop*“, tak se v rámci vypínací sekvence v bloku popsaném v předchozí podkapitole (obr. Obr. 40) proměnná „*Cell_StopProduction*“ změní na hodnotu „*TRUE*“. Zároveň pokud manipulační robot ještě nezačal výrobu další součásti podle výrobní sekvence, tak se hodnota „*NextPartSize*“ změní na 0. To znamená, že následující součást se již nezačne vyrábět. Zároveň se nastaví hodnota „*NextOrderIsActive*“ na „*False*“, která je dále použita při větvení algoritmu.

Druhá podmínka kontroluje, zda je splněna celková bezpečnost, jestli je buňka v automatickém režimu a jestli nebyla buňka pozastavena. Když jsou všechny tyto nutné podmínky splněny, tak může začít výrobní program. Volba konkrétní výrobní sekvence se vybírá dle aktuální hodnoty proměnné „*PartSize*“. Podmínky a celá sekvence jsou u všech výrobních programů totožné, pouze se liší v číslech volaných robotických operací odpovídajících dané velikosti desek. Proto je na vývojovém diagramu na Obr. 41 znázorněna pouze posloupnost pro výrobu největších polotovarů, protože by se části grafy opakovaly.

Po volbě výrobní sekvence se jako první kontroluje, jestli už první operace nebyla vyhotovena v rámci předchozího výrobního programu. Protože při operaci sponkování, kterou vykonává balicí robot, mezitím manipulační robot zakládá deskový polotovar na CNC router, aby zbytečně nečekal a byl co nejvíce vytížen. Proto se musí na začátku každého výrobního programu kontrolovat, zda tato první operace v materiálovém toku nebyla již provedena, ať se neprovádí duplicitně. Dle této podmínky je volán první stav v rámci stavového automatu výroby pomocí proměnné „*bFB_OpNum*“.

Při prvním spuštění je tedy volán první stav, ve kterém PLC kontroluje, jestli není nutné doplnit pneumatickou sponkovačku, protože právě v této části je vhodné tuto doplňovací proceduru provést, protože je nejmenší pravděpodobnost, že se roboty střetnou. Pak je volána ona operace pro založení polotovaru do CNC. V rámci všech stavů v tomto nekonečném automatu vždy po volání nějaké operace následuje podmínka, zda byla operace spuštěna kontrolou proměnné z funkčního bloku řízení robotou „*R_StateNum*“. Pokud se nachází ve stavu hodnoty 10, tak robot právě vykonává nějakou operaci. Následně vždy po této podmínce v následujícím stavu je kontrolováno, zda robot již operaci dokončil. Zde je použita redundantní podmínka, ve které se kontroluje, zda robot nevykonává nějakou operaci. Ve stavu 105 po skončení první robotické operace je volán výrobní program CNC routeru. Proto je ve stavu 115 potřeba zkontrolovat, zda byl CNC program již dokončen a zda je CNC stroj již v domovské pozici. Důležitá změna v rozhodovacích podmínkách je až ve stavu 140, kdy robot 2 vykonává operaci balení. Pokud je hodnota „*NextPartSize*“ 2, což by měla být jediná hodnota, která při výrobě velikosti 1 v této proměnné kromě 0 může nastat, tak manipulační robot začne vykonávat zakládací operaci destičky střední velikosti na CNC router a zároveň je volán stav 146 místo stavu s hodnotou 145. Ten je naopak volán, pokud podmínka s hodnotou 2 není splněna, což znamená, že je buňka ve stavu vypínání výroby. V posledním stavu, který je reprezentován hodnotou 155, je kontrolováno, zda manipulační robot dokončil operaci uložení produktu na výstupní dopravník, který je následně spuštěn. Pak probíhá důležitá část přepsání proměnných definujících následující výrobní program. Pokud již manipulační robot začal výrobní sekvenci následující součásti, což znamená, že hodnota proměnné „*NextOrderIsActive*“ je kladná, tak je ještě znovu ověřeno, zda nebyl stanoven požadavek na zastavení výroby. Pokud jsou všechny podmínky splněny, tak se číslo následující výrobní části vloží do proměnné

„PartSize“ a na její pozici se запиše nové číslo následující části, která je v tomto případě číslo 3. Protože po dokončení výroby střední velikosti následuje výroba nejmenší gravírované desky.

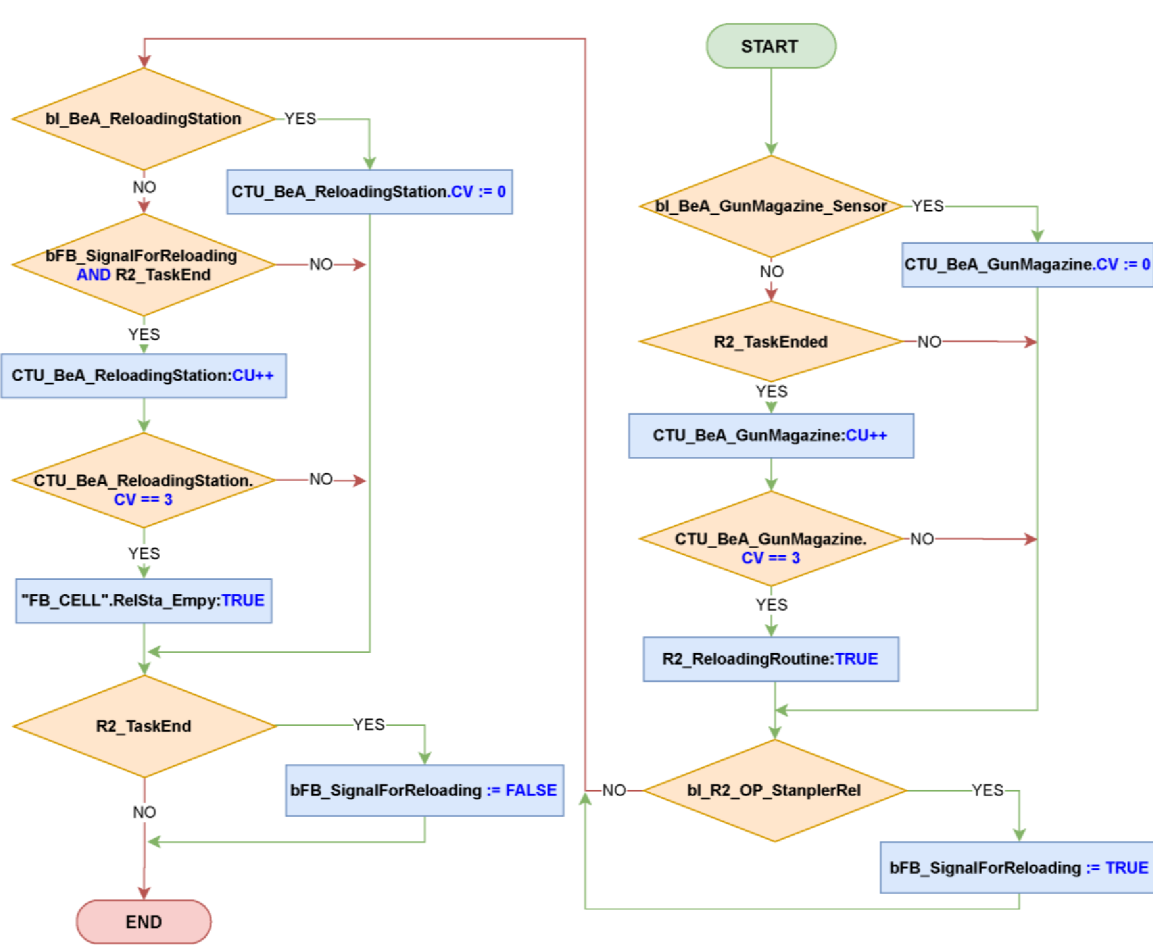


Obr. 41) Vývojový diagram funkčního bloku řízení výrobního programu – Program Flow

4.2.5 FB Reloading Station – Doplnovací stanice

Funkční blok Doplnovací stanice (viz Obr. 42) je určen pro neustálou kontrolu stavu zásobníku spon v automatické doplňovací stanici a druhého zásobníku v pneumatické sponkovačce, kterou je osazen balicí robot. Oba z nich jsou osazeny indukčními snímači, které kontrolují určitou minimální hladinu spon. Pokud zásobárna spon klesne pod tuto hladinu, tak u zásobníku na robotu je ještě dostatek spon pro sešití 3 kusů produktu a stejně tak po detekci nedostatku materiálu v doplňovací stanici je možné ještě celý magazín sponkovačky doplnit celkem třikrát.

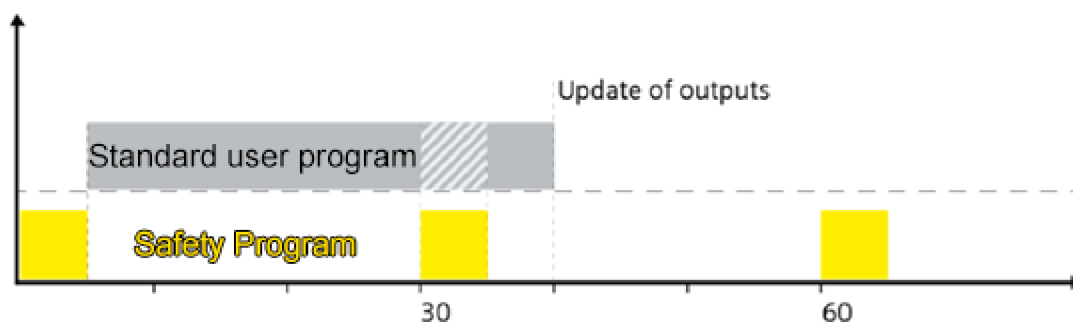
Proto jsem v programu použil čítače typu CTU. Konkrétně program nejprve kontroluje vstup ze senzoru zásobníku sponkovačky „*bl_BeA_GunMagazine_Sensor*“. Pokud je tento signál „*TRUE*“, tak není problém s nedostatkem spon a čítač „*CTU_BeA_GunMagazine*“ je resetován a tok programu skočí do další části. Ale pokud je jeho hodnota „*FALSE*“ a zároveň robot 2 dokončí pracovní operaci, tak čítač přičte kladnou hodnotu a po dosažení čítače 3 se proměnná „*R2_ReloadingRoutine*“ změní na kladnou hodnotu. Tuto hodnotu čte funkční blok „*Program Flow*“, který při nejbližší vhodné příležitosti v rámci výroby zavolá proceduru robotu 2 pro doplnění spon. Po jejím zavolání se změní hodnota lokální proměnné „*bFB_SignalForReloading*“ na „*TRUE*“, což při splnění ekvivalentních podmínek jako u prvního čítače přičte hodnotu i u čítače s názvem „*CTU_BeA_ReloadingStation*“. Ten počítá celkový počet cyklů doplnění spon ve stavu, kdy signál ze snímače automatické stanice detekuje, že zásoba spon klesla pod minimální hranici. Opět po dosažení hodnoty 3 je vystaven signál, který zastaví buňku a vystaví na HMI patřičnou hlášku. Na úplném konci je lokální proměnná „*bFB_SignalForReloading*“ nastavena na defaultní hodnotu.



Obr. 42) Vývojový diagram funkčního bloku pro volání procedur pro doplnění spon

4.2.6 FailSafe PLC program

Nad rámec práce jsem vytvořil zjednodušený Safety PLC program pro řízení funkční bezpečnosti. Firma Siemens nabízí možnost volby PLC, která jsou certifikovaná pro tento typ řízení a je tak možné v rámci jedné řídicí jednotky řešit jak uživatelský, tak i bezpečnostní program. Obě části řízení jsou vyhodnocovány nezávisle na sobě a vždy má větší prioritu část obsahující bezpečnost. To znamená, že bezpečnostní program se vyhodnocuje periodicky podle intervalu, který je ve výchozím stavu nastaven na 150 ms. Tento interval je možné změnit v nastavení PLC v záložce „Fail-Safe“, kde je také možné vypnout všechny funkcionality spojené s bezpečností, pokud chceme řídicí jednotku používat pouze v normálním režimu. Když se začne vyhodnocovat bezpečnostní program, tak je přerušen cyklus (angl. cyclic interrupt) uživatelského programu, jak je naznačeno na Obr. 43.



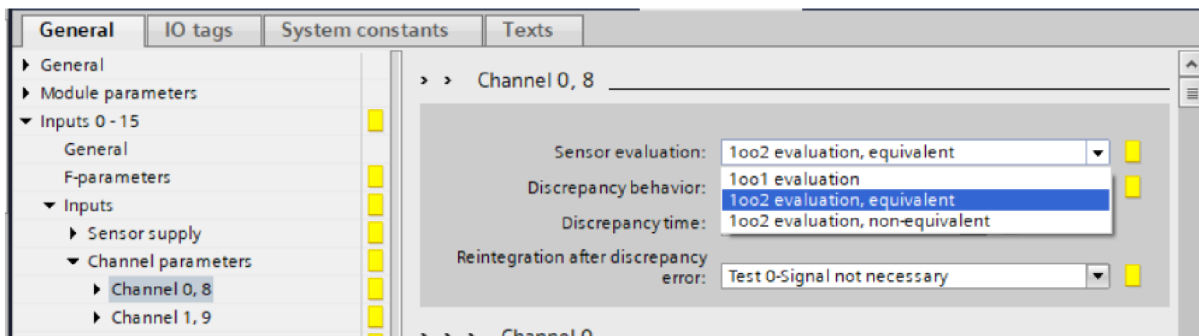
Obr. 43) Prioritní vyhodnocování safety programu [35]

Základní nastavení

V hardwarové konfiguraci projektu jsem použil model PLC 1513F-1 PN, kdy písmeno F v názvu právě označuje jednotku certifikovanou pro použití v bezpečnosti. Pro vstupy z bezpečnostních prvků zmíněných v kapitole 3.4.4 byla přidána do hardwarové konfigurace rozšiřující karta s digitálními vstupy, která svůj účel pro aplikaci v bezpečnosti opět indikuje žlutým zbarvením. Stejně tak byla přidána karta s digitálními výstupy primárně určená pro propojení s bezpečnostními vstupy kontrolérů IRC5, které slouží pro odstavení robotů z automatického režimu.

Řídicí signály z bezpečnostních prvků, jako je typicky třeba E-stop, jsou zpravidla zdvojené kvůli zvýšené úrovni bezpečnosti. To znamená, že tyto zdvojené vodiče jsou zapojeny jako samostatné vstupy do PLC, které jsou spolu spárovány v rámci příslušných kanálů. Vstupy v rámci kanálu jsou neustále porovnávány, a když jejich hodnoty nejsou shodné pro definovaný čas, který bývá v řádu milisekund, tak to PLC vyhodnotí jako chybu. V základním nastavení při takovém stavu přiřadí daným vstupům hodnotu 0 a celý kanál deaktivuje, takže se přestanou vyhodnocovat příslušné vstupy. Kanál je možné aktivovat teprve manuálně potvrzovacím příkazem. Aktivace kanálu je možné také nechat provádět automaticky PLC, ale já jsem v práci použil manuální způsob mimo jiné proto, že se jedná o nejbezpečnější variantu.

U každého vstupního kanálu je možné vybrat ze 3 variant vyhodnocování vstupů. Signál zdvojených signálů z bezpečnostních prvků totiž může být stejný nebo inverzní. K tomu slouží výběr variant nastavení „equivalent“ a „non-equivalent“ (viz Obr. 44). Pro případ, kdy bezpečnostní senzory mají pouze jeden signálový vodič, je určená varianta „1ool“ (angl. one out of one). Následně se spárované vstupy v rámci kanálu vyhodnocují nezávisle na sobě. Na Obr. 45 je uvedeno adresování zdvojených bezpečnostních vstupů použitých v této práci.



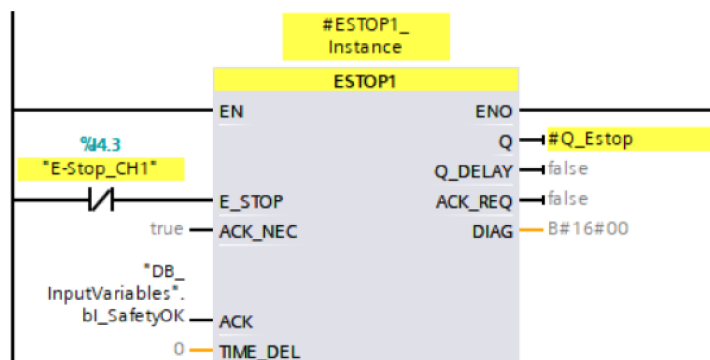
Obr. 44) Nastavení jednotlivých kanálů na kartě bezpečnostních vstupů

F-DI 16x24V DC_1 [F-DI 16x24V DC]						
Properties						
General						
Name	Type	Address	Tag table	Comment		
SafeDoors_Small_CH1	Bool	%I4.0	SAFETY			
SafeDoors_Large_CH1	Bool	%I4.1	SAFETY			
LightCurtain_CH1	Bool	%I4.2	SAFETY			
E-Stop_CH1	Bool	%I4.3	SAFETY			
	Bool	%I4.4				
	Bool	%I4.5				
	Bool	%I4.6				
	Bool	%I4.7				
SafeDoors_Small_CH2	Bool	%I5.0	SAFETY			
SafeDoors_Large_CH2	Bool	%I5.1	SAFETY			
LightCurtain_CH2	Bool	%I5.2	SAFETY			
E-Stop_CH2	Bool	%I5.3	SAFETY			

Obr. 45) Použití zdvojené vstupy bezpečnostních prvků

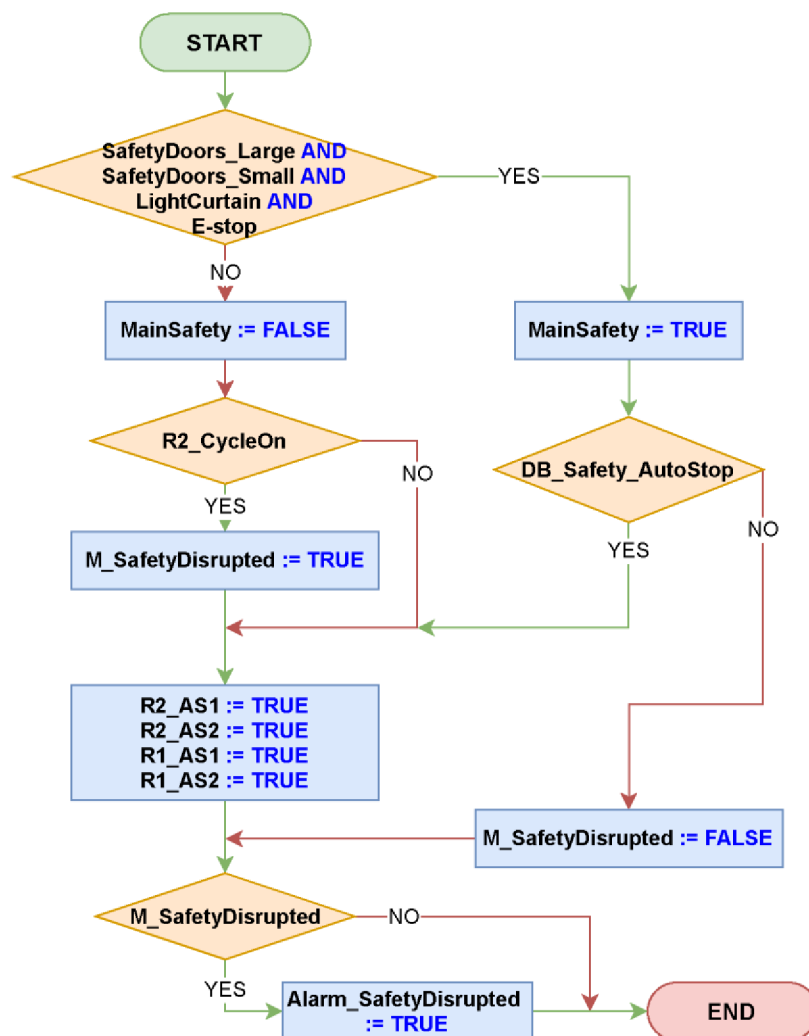
FB Main Safety

Pro tvorbu Safety PLC programu jsou připraveny speciální bezpečnostní bloky, které také splňují příslušné certifikace a požadavky spojené s funkční bezpečností, které jsou převážně popsány v normě IEC 62 061. Bloky jsou navrženy účelně, aby programátoři mohli vytvářet co nejjednodušší algoritmy a vyvarovali se tak chybám při sestavování logiky řízení bezpečnosti. Při tvorbě programu byly primárně použity tyto certifikované bloky pro bezpečnostní dveře, E-stop a světelnou závoru. Příklad sestaveného bloku E-stop z kódu je uveden na Obr. 46. Všechny tyto bezpečnostní prvky je dle regulí při opětovném přivedení pozitivního signálu navíc potřeba potvrdit, k tomu slouží vstup do bloku „ACK“. Výstup „Q“ je napojený na vnitřní statickou proměnnou, s kterou je dále pracováno v další části programu.



Obr. 46) Sestavený blok E-stopu s NC kontaktem

Funkční blok „*Main Safety*“ (viz Obr. 47) v první části porovnává již zmíněné výstupní vnitřní statické proměnné z výše popsaných bloků. Pokud jsou všechny splněny, tak je také splněný celý bezpečnostní okruh a do proměnné „*MainSafety*“, která je použita při ověřování bezpečnostní podmínky v uživatelském programu, se zapíše hodnota „*TRUE*“. Při splnění bezpečnosti je možné také požádat o bezpečný vstup do buňky. Při stanovení takového požadavku (viz kapitola 4.2.3) se v rámci vypínací sekvence pracovního cyklu buňky musí spolehlivě zastavit také automatický cyklus robotů. To je provedeno pomocí proměnné „*DB_Safety_AutoStop*“, která je nastavena právě v tomto vypínacím cyklu v bloku „*Cell Control*“. Následně v bloku funkční bezpečnosti při načtení kladné hodnoty této proměnné se digitální výstupy napojené na kontroléry robotů nastaví na hodnoty „*TRUE*“, čímž je blokováno přepnutí robotu do automatického cyklu. Automatický režim je samozřejmě blokováno i v případě, kdy není splněný bezpečnostní okruh. Navíc při náhlém shoení bezpečnosti, což je detekováno tak, že robotický cyklus je právě v provozu, tak vnitřní proměnná „*SafetyDisrupted*“ se změní na hodnotu 1, čímž se zobrazí na HMI panelu příslušné okno s hláškou upozorňující obsluhu na neobvyklý stav.



Obr. 47) Vývojový diagram bloku starajícího se o funkční bezpečnost

4.3 Tvorba HMI

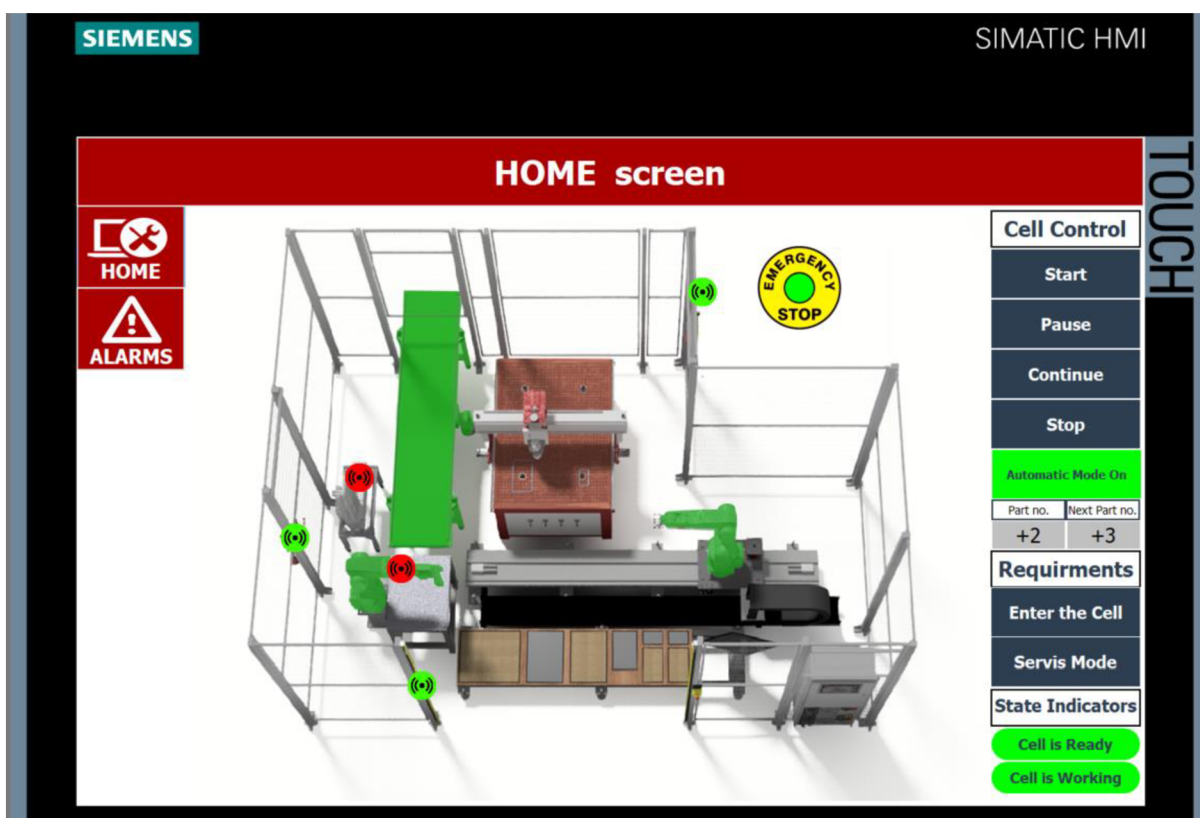
Uživatelské rozhraní pro ovládání celé buňky bylo navrženo pro dotykový panel Siemens TP1200 Comfort, který má rozlišení 1280 x 800 pixelů. Pro tvorbu jeho vizualizace byl použit program Simatic WinCC, který je integrovaný do vývojového prostředí TIA Portal.

4.3.1 Rozložení prvků vizualizace

Na Obr. 48 je zobrazena domovská stránka HMI panelu celé buňky. V levé horní části je nabídka menu, kde je možné přepínat mezi jednotlivými obrazovkami. Ty jsou v projektu aktuálně použity 2. Menu je řešeno globálním prvkem přes funkci „Screen Templates“, díky čemuž je možné dělat úpravy na jednom místě. Kromě úvodní obrazovky „Home“ s ovládáním celé buňky je v nabídce obrazovka s názvem „Alarms“, na které je umístěna tabulka s historizací neobvyklých stavů a výstražných upozornění. Nabídka menu včetně nadpisu aktuální zobrazené plochy je navržena tak, aby ji bylo možné jednoduše rozšířit dle aktuálních požadavků a případně využít dle stejného schématu i v dalších aplikacích.

Hlavní část domovské obrazovky zaujímá vizualizace celého pracoviště, ve které jsou rozmístěny důležité senzory z hlediska řízení v závislosti na jejich pozici v rámci buňky, jež indikují svůj aktuální stav barvou. Červené zbarvení indukuje stav hodnoty „False“ a naopak zelené hodnotu „True“. Protože senzory nejsou v rámci hlavního náhledu popsány, tak je možné kliknutím na jejich indikátory otevřít vyskakovací okno s jejich popisem.

Stejně tak jsou zelenou barvou zvýrazněna zařízení, která zrovna vykonávají nějakou činnost. Konkrétně se jedná o oba průmyslové roboty, CNC router a výstupní dopravník. Příklad je možné vidět na Obr. 48, kde jsou právě v provozu oba roboty a dopravník.



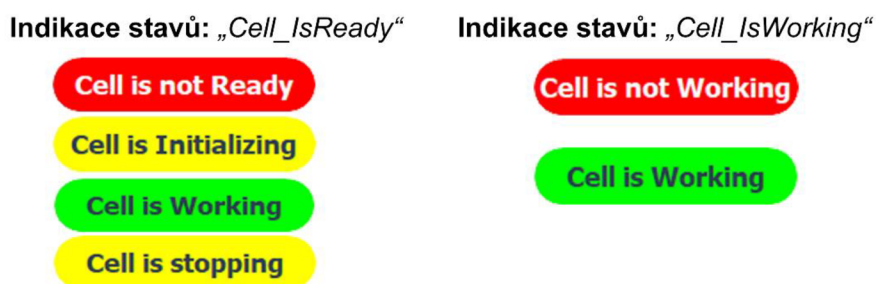
Obr. 48) Vizualizace HMI panelu domovské stránky

4.3.2 Ovládací prvky

Ovládací prvky celé výrobní buňky jsou umístěny v pravé části hlavní obrazovky. Pomocí nadpisů jsou rozděleny do 3 oddílů. První s názvem „*Cell Control*“ obsahuje základní tlačítka pro ovládání automatického režimu. Tlačítkem „*Start*“ se spustí inicializační proces pro nahození pohonných jednotek všech zařízení na pracovišti a proběhne nastavení proměnných na výchozí hodnotu včetně nastavení hodnot pro výrobu součástí dle velikosti v dialogovém okně „*Part no.*“ „*Next Part no.*“. Jsou to hodnoty, kterými je řízen funkční blok výrobní sekvence popsany v kapitole 4.2.4, a jsou na panelu uvedeny, aby obsluha měla přehled, jaký výrobní program právě probíhá. Avšak po dokončení spouštěcí procedury není buňka spuštěna do automatického režimu. K tomu slouží tlačítko „*Automatic Mode*“, které mění svůj text i barvu v závislosti na svém stavu. Ovládací prvky „*Pause*“ a „*Continue*“ pouze zastaví výrobní cyklus, ale nevypnou motory robotů. A tlačítkem „*Stop*“ se spustí vypínací rutina automatického režimu, což se po jeho bezprostředním stisknutí projeví změnou stavu indikátoru a vynulováním dialogového okna „*Next Part no.*“.

Pod nadpisem „*Requirements*“ se nachází požadavky pro zvláštní zásahy do výroby. Je to vstup do buňky („*Enter the Cell*“), kdy buňka po skončení aktuální operace zastaví svůj cyklus, vypnou se pohony, bezpečnostní část PLC spustí „*AutoStop*“ a odemknou se zámky bezpečnostních dveří. Stejný proces proběhne i při volbě tlačítka „*Servis Mode*“, ale zde proběhne zastavení buňky ihned. Je to z toho důvodu, že při servisním zásahu je požadován okamžitý vstup do buňky, když například robot neprovede nějakou operaci správně, a obsluha tak může předejít výrobě neshodného kusu přímým zásahem do pracoviště. Při stisknutí těchto tlačítek nejprve vyskočí potvrzovací obrazovka, aby nedošlo k nechtěnému zastavení výroby, při jejich neúmyslném zmáčknutí.

Ve třetí části „*State Indicators*“ se nachází pouze výstupní prvky, které informují obsluhu o aktuálním stavu, ve kterém se buňka právě nachází. Horní indikátor je navázán na proměnou „*CELL_IsReady*“ a informuje obsluhu, zda jsou všechny pohony připraveny a je dokončena inicializace, nebo byl stanoven požadavek na zastavení. Druhý prvek je spojen s proměnnou „*Cell_IsWorking*“ a dává operátorovi na vědomí, zda mají oba roboty spuštěný cyklus, takže je možné volat jednotlivé operace. Případy všech možných stavů jsou zachyceny na Obr. 49.

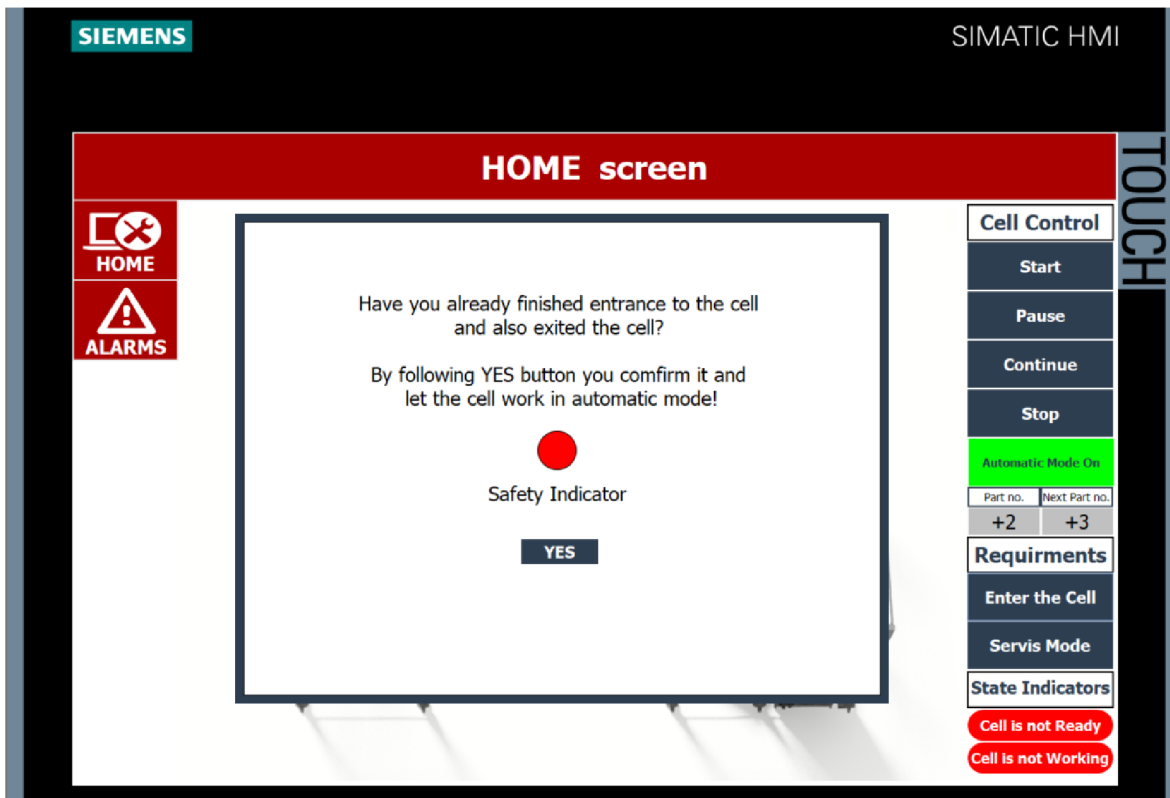


Obr. 49) Výpis všech možných stavů indikátorů režimů v buňce z vizualizace HMI panelu

4.3.3 Automatické otvírání vyskakovacích oken přes Visual Basic skript

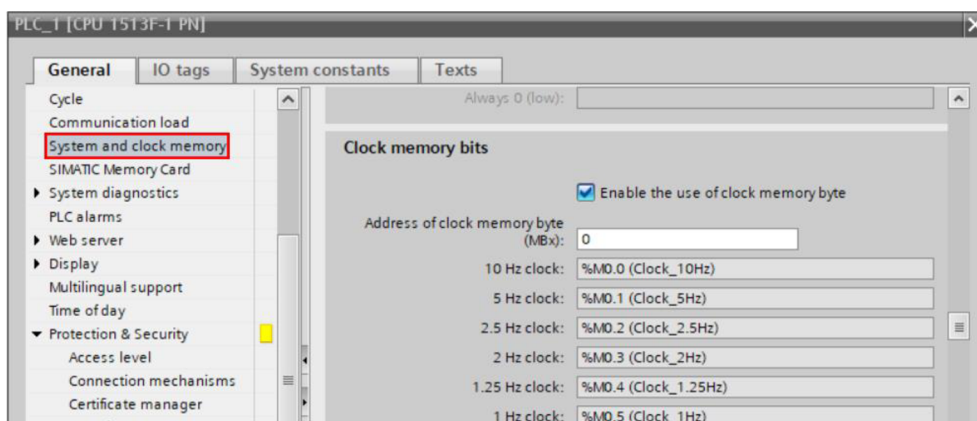
Při stavech, kdy je pozastavena výroba, se na dotykovém panelu zobrazí příslušné vyskakovací obrazovky tzv. „*Popup Screens*“. Jeden z příkladů při vstupu do buňky je ukázán na Obr. 50. Pro spuštění těchto obrazovek je použit skript v jazyku Visual Basic (dále jen VB), který automaticky zobrazí obrazovky s textem, pokud je příslušná spouštěcí proměnná na hodnotě „*TRUE*“. Důvod proč jsem použil tento způsob oproti klasickému zobrazení přes funkci

„ShowPopUpScreen“ po stisknutí tlačítka na HMI panelu je ten, že obsluha může stisknout tlačítko „Yes“ vyskakovacího okna a zavřít ho tak i ve stavu, kdy jsou dveře ještě otevřeny a není tak splněný bezpečnostní okruh. To by znamenalo, že po opravdovém dokončení vstupu by již vyskakovací okno na hlavní obrazovce nebylo a nebylo by tak možné se ze stavu právě probíhajícího vstupu dostat a spustit buňky od místa, ve kterém byla výroba přerušena. Stejným způsobem je například řešeno i oznámení obsluze o nečekaném přerušení Safety.



Obr. 50) Vyskakovací obrazovka při vystavení požadavku pro vstup do buňky

Pro opětovné spouštění VB skriptu jsem použil systémové hodiny o frekvenci 1 Hz. To znamená, že každou sekundu je kód volán. V rámci kódu se ověřuje, zda jsou proměnné, které spouští vyskakovací okna, aktivní, a pokud tomu tak je, tak se zobrazí příslušná obrazovka. Systémové hodiny je nutné povolit v nastavení PLC (viz Obr. 51). Tím se vytvoří vnitřní proměnné v podobě PLC tagů, které je pak nutné namapovat na HMI tagy, kterým lze přiřadit mimořádnou událost při změně jejich hodnoty, kterou může být právě spouštění VB skriptu.



Obr. 51) Nastavení vnitřních systémových hodin v PLC

Při tvorbě spouštěcího kódu jsem použil jednoduché struktury If-Else a předdefinované funkce „ShowPopupScreen“, která je již implementována v rámci knihovny VB (viz Obr. 52).

```

1 Sub VB_FC_ShowPopups()
2   'Popup Screen nutném zastavení při nedostatku spon v doplňovací stanici'
3   If SmartTags("DB_Alarms_bM_ReloadingStanplers_Required") Then
4     ShowPopupScreen "ReloadingStanplers_Required", 200, 100, hmiOn,hmiAnimationOff, hmiMedium
5   ElseIf Not SmartTags("DB_Alarms_bM_ReloadingStanplers_Required") Then
6     ShowPopupScreen "ReloadingStanplers_Required", 200, 100, hmiOff,hmiAnimationOff, hmiMedium
7   End If
8   'Popup Screen při probíhajícím doplňování spon a detekci senzoru zásobníku, že spony byly doplněny'
9   If SmartTags("DB_Alarms_bM_ReloadingStanplers_Finished") Then
10    ShowPopupScreen "RelStation_Reloaded", 200, 100, hmiOn,hmiAnimationOff, hmiMedium
11  End If
12  'Popup Screen při právě probíhajícím vstupu do buňky nebo servisním zásahu'
13  If SmartTags("DB_Alarms_bM_Entrance_Finished") Then
14    ShowPopupScreen "FinishedEntrance", 200, 100, hmiOn,hmiAnimationOff, hmiMedium
15  End If
16  'Popup Screen při nečekaném shození SAFETY'
17  If SmartTags("DB_Alarms_bM_SafetyUnexpectedlyDisrupted") Then
18    ShowPopupScreen "SafetyDisrupted", 200, 100, hmiOn,hmiAnimationOff, hmiMedium
19  End If
20  'Popup Screen při nutném zastavení při nedostatku polotovarů'
21  If SmartTags("DB_Alarms_bM_ReloadMaterial_Required") Then
22    ShowPopupScreen "ReloadingMaterial", 200, 100, hmiOn,hmiAnimationOff, hmiMedium
23  End If
24 End Sub

```

Obr. 52) Visual Basic skript sloužící pro spouštění vyskakovacích obrazovek

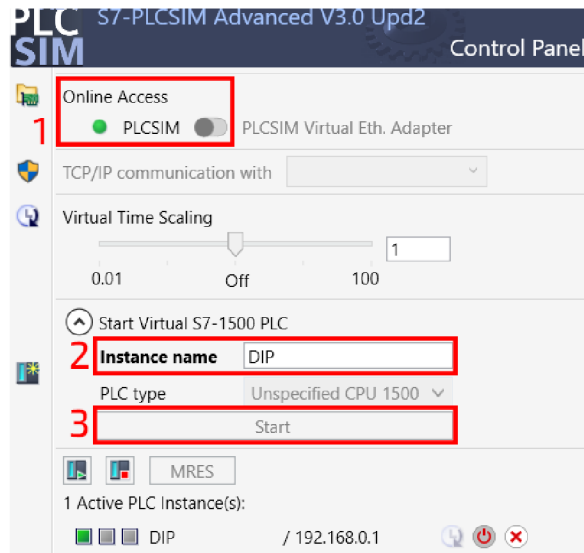
4.4 Virtuální zprovoznění

Pro samotné virtuální zprovoznění je potřeba provést několik úkonů, aby bylo možné jednotlivé softwary propojit. V první řadě se musí namapovat adresy vstupů a výstupů PLC programu na ekvivalentní proměnné v programu PS. Protože jsem použil více než 80 tagů, tak jsem na Obr. 53 uvedl jen krátký výběr na ukázkou.

Signal Name	IEC Format	Name	Address
irb1200 5 90 PN RunChainOK	IO.0	b_PN_R1_RunChainOK	%I0.0
irb1200 5 90 PN AutoOn	IO.1	b_PN_R1_AutoOn	%I0.1
irb1200 5 90 PN CycleOn	IO.2	b_PN_R1_CycleOn	%I0.2
irb1200 5 90 PN MotorsOnState	IO.3	b_PN_R1_MotorsOnState	%I0.3
irb1200 5 90 PN MotorsOffState	IO.4	b_PN_R1_MotorsOffState	%I0.4
irb1200 5 90 PN RobotWorking	IO.5	b_PN_R1_RobotWorking	%I0.5
irb1200 5 90 PN RoutineDone	IO.6	b_PN_R1_TaskEnd	%I0.6
CNC router ProgramEnded	IO.7	b_CNC_ProgramEnded	%I0.7

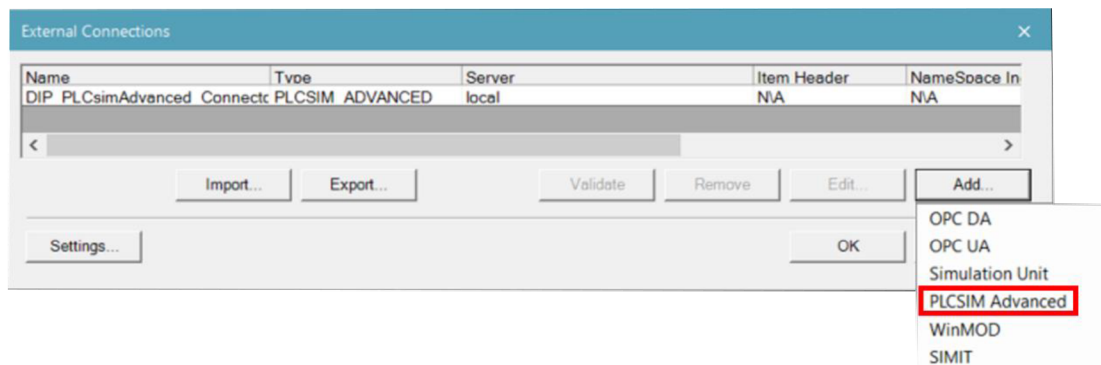
Obr. 53) Příklad mapování signálů v softwaru PS (vlevo) a TIA Portal (vpravo)

Jelikož PLCsim Advanced zastupuje v simulačním schématu komunikační rozhraní, tak v dalším kroku je potřeba nastavit právě tento program, aby bylo možné k němu následně připojit jak simulaci PLC, tak simulaci mechaniky pracoviště. Jak je naznačeno na Obr. 53, použil jsem komunikaci on-line přístupu PLCSIM. Pak už jsem jen pojmenoval název PLC instance, který může být libovolný, a spustil ji tlačítkem „Start“. Tím se vytvoří přístupná lokální IP adrese PLC jednotky.



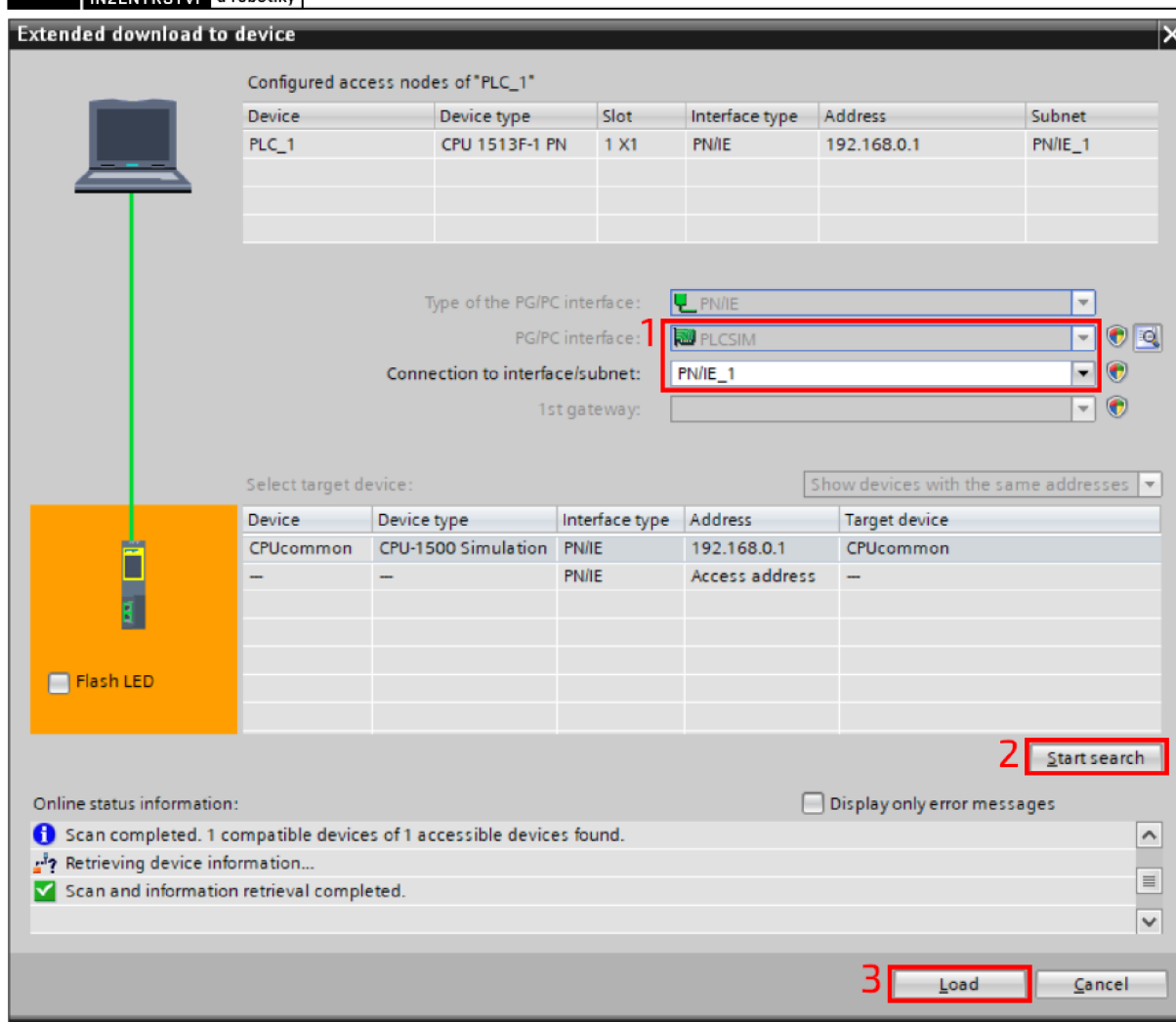
Obr. 54) Nastavení PLCsim Advanced

Na straně PS je pro připojení komunikace ve verzi 16 postup následující. V záložce „File > Options > PLC“ je nutno přepnout na externí simulaci volbou „PLC > External Connection“. Pro přímé připojení k rozhraní PLCsim Advanced je nutné otevřít okno „External Connections“, kde je možné vybrat připojení z nabídky, jak je naznačeno na Obr. 55. Pak již stačí vybrat aktuální instanci vytvořenou v PLCsim Advanced, která je v rámci sítě přístupná.



Obr. 55) Připojení komunikace pro PLCsim Advanced v PS

Na straně IDE TIA Portal je nutné v první řadě zaškrtnout pole „Support simulation during block compilation“ ve vlastnostech projektu v záložce „Protection“, bez čehož by simulace nefungovala. Následně je možné již nahrát dle klasického postupu kompletní PLC program včetně hardwarové konfigurace do emulované PLC jednotky, jak je znázorněno na Obr. 56, pokud je spuštěna instance v programu PLCsim Advanced. Bez jejího spuštění totiž není virtuální PLC jednotka k nalezení ani po prohledání lokální sítě pomocí tlačítka „Start Search“.



Obr. 56) Nastavení komunikace a stažení řídicího programu do virtuální PLC jednotky

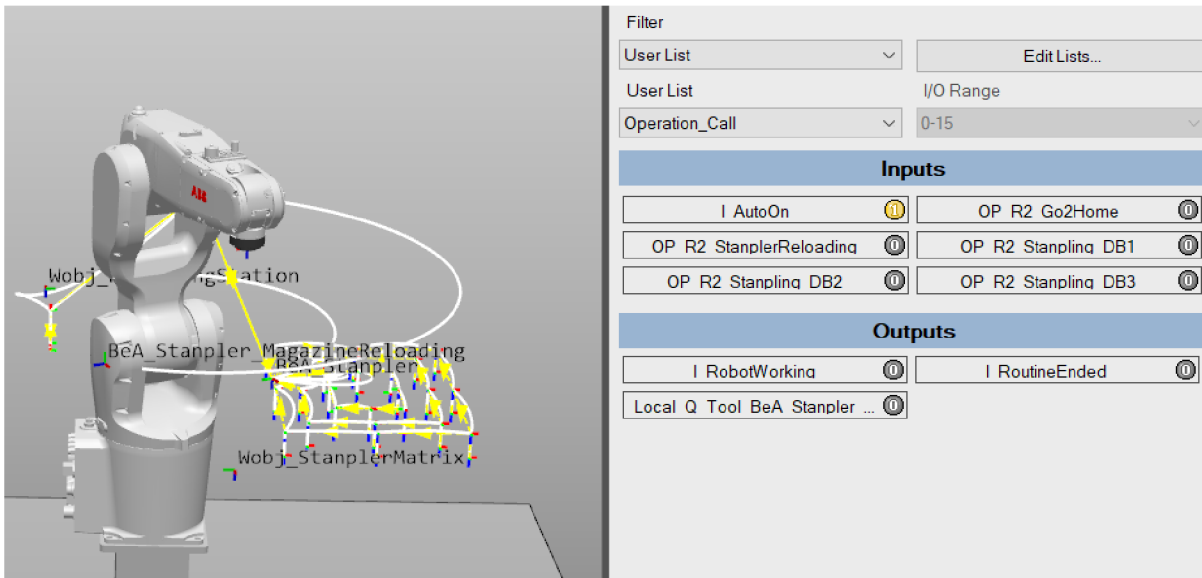
4.5 Úpravy a ověření robotických programů v RobotStudio

Získané robotické programy z PS by měly díky rozšířeným nativním OLP příkazům kontroléru ABB Rapid a doprogramováním vlastních příkazů přesně odpovídat syntaxi jazyku Rapid. Abychom to mohli říci s jistotou, je vhodné získané programy otestovat v simulačním softwaru RobotStudio firmy ABB, ve kterém simulaci řídí přesná kopie reálného robotického kontroléru.

Ověření u robotu, který má na starost balení, proběhlo bez nutnosti zásahu do kódu a bez jediné úpravy (viz Obr. 57). Proto mohu potvrdit, že z PS je možné spolehlivě získat robotický program pro základní typy operací.

U manipulačního robotu bylo potřeba doplnit robotický kód z PS o instrukci SearchL, kterou nebylo možné v rámci PS nasimulovat (viz Obr. 58). Tuto funkci bylo nutné použít, protože souřadnice Z bodu trajektorie, ve kterém dochází k odebrání deskových polotovarů, se mění v závislosti na aktuálním počtu stohovaných polotovarů. Robot s využitím této pokročilé funkce se přibližuje k bodu, který je definován jako nejnižší bod uchopení nacházející se na plošině zásobníku, a zároveň kontroluje hodnotu vstupního signálu ze senzoru osazeného na vakuovém gripperu. Pokud senzor zaznamená přítomnost materiálu, tak ihned zastaví a zaznamená aktuální polohu do perzistentní proměnné typu „*robtarget*“ s názvem „*DBactualPos*“. Hned v dalším kroku je hodnota souřadnice osy Z této proměnné porovnávána

v podmínce, zda je menší než 3 mm, což by znamenalo, že na plošině zásobníku již není žádný materiál. Pokud je tato podmínka splněna, tak robot odjede do počáteční polohy operace, nastaví výstupní signál „doReloadMaterial“ na hodnotu „TRUE“, čímž dá nadřazenému PLC vědět, že je nutné materiál v zásobníku doplnit, a ukončí aktuální proceduru. PLC při tomto stavu zobrazí na HMI panelu příslušnou hlášku informující obsluhu o nutnosti doplnit materiál. Tato komunikace s nadřazenou řídicí jednotkou byla ověřena pomocí VRC modulu zmíněného v kapitole 4.1.2 v části Nastavení kontrolérů robotů, který umožňuje propojit virtuální kontrolér RobotStudia se simulačním modelem v PS, který již komunikuje napřímo s emulátorem PLCsim Advanced, na kterém se vykonává řídicí PLC program.



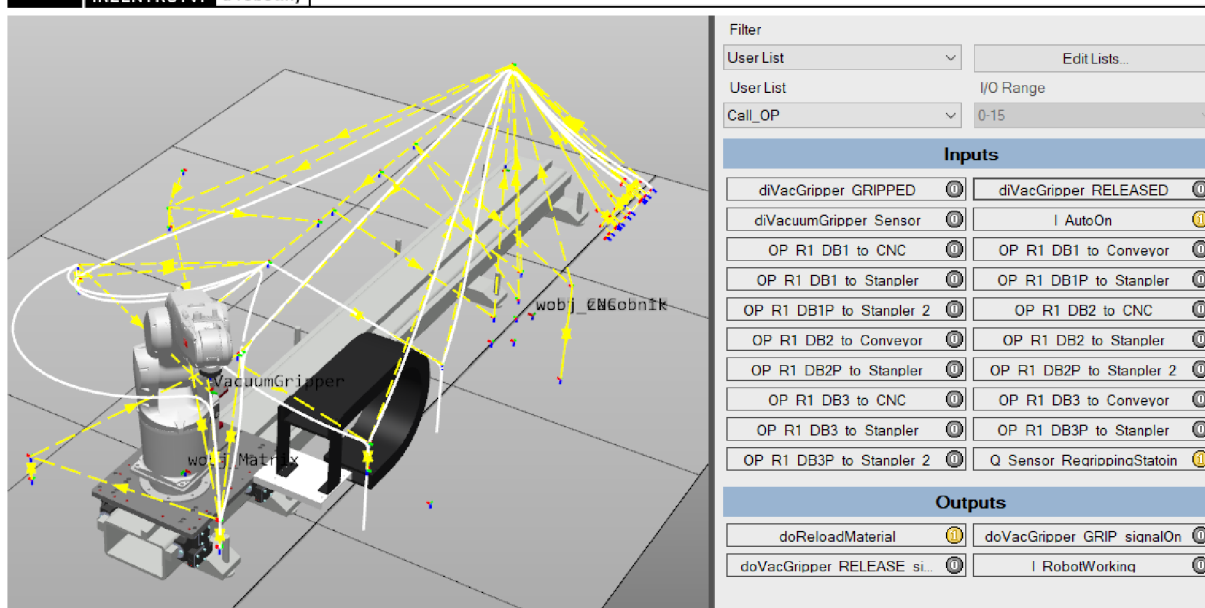
Obr. 57) Ověření robotického programu balicího robotu

```

253 PROC R1_DB1_to_CNC()
254   MoveJ DB1toCNC_10,v2000,z100,VacuumGripper\Wobj:=wobj_zasobnik;
255   MoveL DB1toCNC_20,v1500,z40,VacuumGripper\Wobj:=wobj_zasobnik;
256   !MoveL DB1toCNC_30_pick,v100,fine,VacuumGripper\Wobj:=wobj_zasobnik;
257   SearchL \STOP, diVacuumGripper_Sensor,DBActualPos,DB1toCNC_30_pick,v100,VacuumGripper\Wobj:=wobj_zasobnik;
258   IF DBActualPos.trans.z < 3 THEN
259     MoveL DB1toCNC_20,v500,z40,VacuumGripper\Wobj:=wobj_zasobnik;
260     MoveJ DB1toCNC_10,v2000,z100,VacuumGripper\Wobj:=wobj_zasobnik;
261     Set doReloadMaterial;
262     WaitTime 2;
263     Reset I_RobotWorking;
264     ExitCycle;
265   ENDF
266   PulseDO \High, \PLength:=1, doVacGripper_GRIP_signalOn;
267   WaitDI diVacGripper_GRIPPED, 1;
268   MoveL DB1toCNC_20,v300,z10,VacuumGripper\Wobj:=wobj_zasobnik;
269   MoveJ RelStat_pre,v2500,z100,VacuumGripper\Wobj:=wobj_zasobnik;
270   MoveJ DB1toCNC_40,v1000,z30,VacuumGripper\Wobj:=wobj_zasobnik;

```

Obr. 58) Ukázka doplněné instrukce SearchL v robotickém programu



Obr. 59) Ověření robotického programu manipulačního robotu

5 ZHODNOCENÍ A DISKUZE

V práci byly vypracovány všechny zadané cíle vedoucí k virtuálnímu zprovoznění. Avšak při jejich vypracovávání byly odhaleny určité nedostatky a limity současného vývoje použitých softwarů.

Největší obtíže se objevily při sestavování simulačního modelu a robotizovaných operací. Manipulační robot se při odebírání stohovaných materiálů přibližuje v jednom směru k místu uchopení pomocí pokročilé instrukce SearchL. Avšak implementace takové funkce se v programu Process Simulate i při doinstalování kontrolérů ABB Rapid v současné době nenachází. Velkou výhodou programu Process Simulate je možnost si podobné prvky doprogramovat v podobě OLP příkazů, ale i to má své omezení, kvůli kterým nebylo možné reálné chování přesně nasimulovat a vygenerovat odpovídající příkaz v robotickém programu. Proto bylo nutné získaný kód řízení robotu dodatečně upravit v editoru mimo Process Simulate.

Velkou část práce zabralo vytváření a testování odezvy systémových signálů robotu v simulaci, aby přesně odpovídalo chování kontrolérů ABB a bylo díky nim možné zprovoznit řízení nadřazeného PLC na vysoké úrovni. Tato logika se dá používat opakovaně v dalších projektech, ale nikdy nebude odpovídat chování přesné virtuální jednotky od výrobce robotů. Proto osobně vidím velký potenciál v napojení řízení robotů na simulační programy typu RobotStudio od ABB, kdy tento trend začala prosazovat firma EKS InTec, podobné ambice lze registrovat i u vývojářů Tecnomatix.

Při samotném zprovoznění jsem narazil na vysoké požadavky na výpočetní výkon. Při postupném rozšiřování projektu hlavně o PLC Safety řízení a náročné prvky v simulaci, jako je například přidání lidské obsluhy, se simulace tak zpomalila, že jsem musel některé tyto prvky eliminovat. Na to je nutné myslet při samotném zprovoznění, což si na začátku sestavování projektu nemusíme uvědomovat.

Pro reálné pracoviště by bylo nutné navíc provést analýzu rizik, při které by mohly být odhaleny některé nedostatky funkční bezpečnosti a bylo by tak potřeba některé části programu upravit. Na tento proces by měl být projekt připraven a neměl by jej příliš ovlivnit. Například se jedná o způsob zastavení buňky při nouzovém zastavení. V práci jsem totiž pro okamžité odstavení provozu obsluhoval pouze bezpečnostní vstupy „AutoStop“ robotického kontroléru IRC5, kdy pro určité případy by bylo vhodné raději použít pro odstavení buňky bezpečnostní vstupy „EmergencyStop“ a následně je v programu rozličně přiřadit pro dané případy zastavení. Tento nedostatek nepovažuji za důležitý v souvislosti se stanovenými cíle této práce, protože bezpečnostní PLC program byl vypracován nad rámec zadání a jeho sestavení by se v praxi odvíjelo od vypracované analýzy rizik

Stejně tak je virtuální pracoviště, díky jeho navržené struktuře, možné jednoduše rozšiřovat. Je připraveno pro napojení na SQL databázi s jednotlivými objednávkami od zákazníků, které by se postupně odbavovaly a samy o sobě by tak tvořily výrobní sekvenci jednotlivých kusů produktů. K tomuto účelu je možné použít například volně dostupnou S7/NET/Plus knihovnu, která slouží pro propojení jednotek Siemens S7 s aplikacemi napsanými v jazyku C#, které je možné napojit právě na SQL databázi. Podobné rozšíření by ještě více zautomatizovalo výrobu a zvýšilo možnost větší zákaznické personalizace, která se stává v poslední době trendem nejen ve výrobních procesech. Z databáze by totiž mohly být získávány detailnější informace o jednotlivých zakázkách. Při tomto propojení by bylo možné

navíc osadit výstupní dopravník značícím zařízením, které by mohlo zabalené produkty olepovat štítky s popisem objednávky a s dalšími dodatečnými informacemi.

Můžeme tedy konstatovat, že virtuální zprovoznění zadaného pracoviště bylo splněno s malým nedostatkem vygenerovaného programu manipulačního robotu, kde bylo nutné provést úpravy, po kterých byl program úspěšně otestován v programu RobotStudio. To by mělo zaručit zprovoznění i na reálném robotu. PLC program nebyl otestován na reálné jednotce, takže nemůžeme zhodnotit, na jak vysoké úrovni bylo virtuální zprovoznění provedeno. Ale jelikož řízení nadřazeného PLC fungovalo v digitální podobě při využití věrohodného emulátoru Siemens jednotek PLCsim Advanced, tak by se reálné chování hardwaru nemělo příliš lišit.

6 ZÁVĚR

Hlavním cílem této diplomové práce bylo digitální zprovoznění robotizované výrobní buňky se zadaným layoutem. Účelem výrobního systému je vyrábět gravírované desky, které mají být následně také baleny a transformovány z pracoviště ven. Dle zadaného layoutu byl sestaven 3D simulační model v programu Tecnomatix Process Simulate a vytvořeny všechny požadované operace. Následně byl vytvořen řídicí program nadřazeného PLC včetně vizualizace HMI panelu a jeho napojením na simulační model bylo provedeno virtuální zprovoznění.

V první části je popsán současný stav poznání, kdy je postupováno od obecného základu ke konkrétním detailům dané problematiky. S cílem zasadit termín digitálního zprovoznění do širšího kontextu je blíže popsán termín Průmysl 4.0 a jsou vyjmenovány jeho základní pilíře, jedním z nichž je také simulace, kterou využívá právě i virtuální zprovoznění. Následuje popis jeho hlavních výhod, popis nejpoužívanějších metod a jednotlivých komponent potřebných pro jeho dosažení. Z nich pak následuje popis hlavních představitelů, ze kterých jsou vybrány pro každou příslušenou úroveň nejvhodnější zástupci pro praktickou část. V této teoretické části je ještě popsáno pracoviště, které se bude zprovozňovat, včetně detailního popisu hlavních zařízení. A na závěr jsou uvedeny základní bezpečnostní normy, na které je potřeba při návrhu robotizovaného pracoviště brát ohled.

V praktické části je postupováno co nejvíce systematicky. Takže jsou nejprve popsány všechny typizované úkony, které bylo nutné provést pro sestavení simulačního modelu, včetně úprav nutných pro dosažení reálné odezvy pro nadřazené PLC. Na tuto část navazuje popis k vytvořenému PLC programu. Jednotlivé části kódu jsou popsány pomocí vývojových diagramů s podrobným komentářem. Nad rámec práce byl vypracován i zjednodušený program tzv. Safety PLC starajícího se o řízení bezpečnostních prvků. Na tvorbu PLC programu logicky navazuje tvorba vizualizace HMI panelu s vysvětlením jednotlivých funkcí.

V poslední části je zkráceně popsán postup propojení softwarových balíčků za účelem virtuálního zprovoznění, na který navazuje kapitola s popisem nutných úprav robotického programu vygenerovaného pomocí Tecnomatix Process Simulate a jeho následné ověření v programu RobotStudio.

Na závěr lze shrnout, že v práci byly splněny všechny stanovené cíle zadání. Simulace sice neodpovídá přesně reálnému chování zadaného výrobního systému, ale splňuje základní tezi virtuálního zprovoznění, kdy je možné odladit řídicí PLC program pouze s použitím počítače v digitální podobě do takové míry, že je možné ho následně bez velkých úprav použít na reálném zařízení. Při vypracovávání této práce jsem narazil na omezení a současné nedostatky primárně na straně simulace mechaniky výrobních systémů, které minimálně ztěžují dosažení hlavní myšlenky průmyslu 4.0, kdy budou vedle reálné výroby existovat jejich digitální ekvivalenty. Ale tato problematika jde v poslední době až neuvěřitelně rychle kupředu. Proto jsem osobně zvědavý, jak bude vypadat její budoucí vývoj, a rád bych byl jeho součástí.

7 SEZNAM POUŽITÝCH ZDROJŮ

- [1] ASOCIÁCIA DUÁLNEHO VZDELÁVANIA. Vznik priemyslu 4.0. *Priemysel 4.0* [online]. 2021 [vid. 2021-02-10]. Dostupné z: <http://priemyselstyrinula.sk/viac-o-priemysle-4-0/>
- [2] ERBOZ, Gizem. How to define industry 4.0: main pillars of industry 4.0. In: *Managerial trends in the development of enterprises in globalization era*. 2017, s. 761–767. ISBN 978-80-552-1739-0.
- [3] BIAŁY, Witold, Pavol BOŽEK, Greg GALECKI, DE GRUYTER POLAND, a WALTER DE GRUYTER. *Multidisciplinary aspects of production engineering: monograph*. Warszawa; [Berlin: Sciendo] : [Walter de Gruyter, 2019. ISBN 978-3-11-067482-8.
- [4] IFR. Robot History. *IFR International Federation of Robotics* [online]. 2021 [vid. 2021-02-11]. Dostupné z: <https://ifr.org/robot-history>
- [5] STATISTA RESEARCH DEPARTMENT. *Worldwide installations of industrial robots from 2004 to 2019* [online]. 5. únor 2021 [vid. 2021-02-10]. Dostupné z: <https://www.statista.com/statistics/264084/worldwide-sales-of-industrial-robots/>
- [6] RUI SHU, Zhang, Zhang CHANG a Zheng WEIGANG. The status and development of industrial robots. *IOP Conference Series: Materials Science and Engineering* [online]. 2018, **423**, 012051. ISSN 1757-899X. Dostupné z: doi:10.1088/1757-899X/423/1/012051
- [7] KUC, Arthur. Virtual Commissioning - Software Testing Not Just in It. *JCommerce* [online]. 13. září 2018 [vid. 2021-02-10]. Dostupné z: <https://www.jcommerce.eu/jpro/articles/virtual-commissioning-software-testing>
- [8] LIU, Zheng, Nico SUCHOLD a Christian DIEDRICH. Virtual Commissioning of Automated Systems. In: Florian KONGOLI, ed. *Automation* [online]. B.m.: InTech, 2012 [vid. 2021-02-12]. ISBN 978-953-51-0685-2. Dostupné z: doi:10.5772/45730
- [9] SIEMENS. Reduce commissioning time, cost, and risk. *Siemens Global Website* [online]. 2021 [vid. 2021-02-12]. Dostupné z: <https://new.siemens.com/global/en/company/topic-areas/digital-enterprise/discrete-industry/virtual-commissioning.html>
- [10] JACKSON, Chad. Hybrid Digital Twin for Virtual Commissioning. *Virtual Commissioning* [online]. 29. leden 2020 [vid. 2021-02-10]. Dostupné z: <https://virtualcommissioning.com/hybrid-digital-twin-for-virtual-commissioning/>
- [11] RILEY, Chris. Simulators vs. Emulators: What’s the Difference, Anyway? *Sauce Labs* [online]. 18. prosinec 2018 [vid. 2021-02-07]. Dostupné z: <https://sauce labs.com/blog/simulators-vs-emulators-whats-the-difference-anyway>
- [12] B&R INDUSTRIAL AUTOMATION. *Modeling and simulation* [online]. 2021 [vid. 2021-02-12]. Dostupné z: <https://www.br-automation.com/cs/products/software/modeling-and-simulation/>

- [13] YARAMANOGLU, Nezh. Virtual commissioning and the production process. *Siemens blog* [online]. 25. srpen 2017 [vid. 2021-02-18]. Dostupné z: <https://blogs.sw.siemens.com/thought-leadership/2017/08/25/virtual-commissioning-and-the-production-process/>
- [14] ABB. *RobotStudio produktový list* [online]. 2021 [vid. 2021-02-17]. Dostupné z: <https://search.abb.com/library/Download.aspx?DocumentID=9AKK107991A4796&LanguageCode=cs&DocumentPartId=&Action=Launch>
- [15] FANUC. *Intelligentní offline 3D simulace robotu pomocí softwaru ROBOGUIDE* [online]. 2021 [vid. 2021-03-13]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/příslušenství/roboguide>
- [16] AXIOM TECH. *Process Simulate Robotics* [online]. 2021 [vid. 2021-02-18]. Dostupné z: <https://www.axiomtech.cz/25343-tecnomatix-process-simulate-robotics>
- [17] ZÁPADOČESKÁ UNIVERZITA V PLZNI, FAKULTA STROJNÍ, KATEDRA PRŮMYSLOVÉHO INŽENÝRSTVÍ A MANAGEMENTU. *PLM* [online]. 2011 [vid. 2021-02-18]. Dostupné z: <https://www.digipod.zcu.cz/index.php/oblasti-nasazeni>
- [18] EKS INTEC. *RF::Suite* [online]. 2021 [vid. 2021-04-18]. Dostupné z: https://www.eks-intec.com/rf_en.html
- [19] MARTINÁSKOVÁ, Marie, Ladislav ŠMEJKAL, ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE, a STROJNÍ FAKULTA. *Řízení programovatelnými automaty*. Praha: Vydavatelství ČVUT, 2004. ISBN 978-80-01-02925-1.
- [20] MARTINÁSKOVÁ, Marie, Ladislav ŠMEJKAL, ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE, a STROJNÍ FAKULTA. *Řízení programovatelnými automaty III: softwarové vybavení*. Praha: Vydavatelství ČVUT, 2003. ISBN 978-80-01-02804-9.
- [21] OPC FOUNDATION. *What is OPC* [online]. 2021 [vid. 2021-02-23]. Dostupné z: <https://opcfoundation.org/>
- [22] KOLÁŘ, Petr. Operating system concepts. In: <http://www.nti.tul.cz/> [online]. Technická univerzita v Liberci. 2021. Dostupné z: <http://www.nti.tul.cz/~kolar/os/os-s.pdf>
- [23] ČEFELÍN, David. *Virtuální zprovoznění paletizačního/depaletizačního pracoviště*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Jan Vetiška.
- [24] SIEMENS. *Reality has no reset button: SIMIT - Virtual commissioning and operator training* [online]. 2018 [vid. 2021-03-14]. Dostupné z: <https://assets.new.siemens.com/siemens/assets/api/uuid:15c6f91e-136f-465c-aba7-a13f01a8de46/simit-broschuere-en.pdf>
- [25] ABB. *IRC5 Industrial Robot Controller, data sheet* [online]. 2021 [vid. 2021-02-25]. Dostupné z: <https://search.abb.com/library/Download.aspx?DocumentID=ROB0295EN&LanguageCode=en&DocumentPartId=&Action=Launch>

- [26] GÜDEL. *Linear Traversing Axis TMF* [online]. 2021 [vid. 2021-03-15]. Dostupné z: <https://www.gudel.com/products/linear-tracks-for-robots/tmf>
- [27] BEA GROUP. *BeA Autotec Catalogue* [online]. 2021. Dostupné z: https://www.bea-group.se/fileadmin/user_upload/Meta-Menue/Downloads/sv-pdf/BeA_SE_DK_NO_BeA_Autotec_Catalogue_2019.pdf
- [28] SMC CORPORATION. *Vacuum Pads* [online]. [vid. 2021-03-15]. Dostupné z: https://www.smcworld.com/products/en/s.do?ca_id=488
- [29] ALUTEC KK S.R.O. *Pásové dopravníky* [online]. 2021 [vid. 2021-03-15]. Dostupné z: <https://www.aluteckk.cz/pasove-dopravniky>
- [30] PAVELKA, Václav. *Bezpečnost robotického pracoviště*. Zlín, 2014. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky.
- [31] BALÁKOVÁ, Marie. *Bezpečnostní systémy pro pracoviště s průmyslovými roboty*. Brno, 2015. Vysoké učení technické v Brně, Fakulta strojního inženýrství.
- [32] VOJÁČEK, Antonín. *Bezpečnost strojů - 1. díl - úvod, normy, posouzení rizika. Automatizace.HW.cz* [online]. 9. říjen 2015 [vid. 2021-03-14]. Dostupné z: <https://automatizace.hw.cz/bezpecnost-stroju/bezpecnost-stroju-1-dil-normy-rizika.html>
- [33] SICK. *Guide for Safe Machinery* [online]. 2015. Dostupné z: https://cdn.sick.com/media/docs/8/78/678/special_information_guide_for_safe_machinery_en_im0014678.pdf
- [34] ABB. *IRB 1200 CAD Models* [online]. [vid. 2021-04-25]. Dostupné z: <https://new.abb.com/products/robotics/industrial-robots/irb-1200/irb-1200-cad-models>
- [35] SIEMENS. *Safety Programming Guideline for SIMATIC S7-1200/1500* [online]. 2017 [vid. 2021-04-30]. Dostupné z: https://cache.industry.siemens.com/dl/files/255/109750255/att_931677/v1/109750255_Programming-Guideline-Safety_DOC_V10_en.pdf

8 SEZNAM ZKRATEK A OBRÁZKŮ

8.1 Seznam zkratek

AR	Augmented Reality (rozšířená realita)
CNC	Computer Numerical Control
ERP	Enterprise Resource Planning
EU	Evropská unie
HIL	Hardware-in-the-Loop
IDE	Integrated Development Environment
IOT	Internet of Things
HMI	Human Machine Interface
OPC	Open Platform Communications
OPC UA	Open Platform Communications Unified Architecture
PC	Personal Computer
PLC	Programmable Logic Controller
PVC	Polyvinyl Chloride
RAM	Random Access Memory
SCADA	Supervisory Control and Data Acquisition
SIL	Software-in-the-Loop
SOA	Service-Oriented Architecture
TIA	Totally Integrated Automation

8.2 Seznam obrázků

Obr. 1) 9 základních pilířů Průmyslu 4.0 [3]	19
Obr. 2) Graf počtu celosvětově instalovaných průmyslových robotů podle roku [5]	20
Obr. 3) Srovnání postupu tradičního a virtuálního zprovoznění [7]	21
Obr. 4) Graf závislosti nákladů pro odstranění chyby na etapě odhalení v projektu [9]	21
Obr. 5) Srovnání schémat simulačních modelů SIL a HIL	22
Obr. 6) Kompatibilní simulační programy s PLC IDE společnosti B&R [12]	23
Obr. 7) Portfolio produktů Siemens pro virtuální zprovoznění [13]	23
Obr. 8) Skenovací perioda PLC [19]	27
Obr. 9) Schéma sdílené paměti (Shared memory) RAM [22]	28
Obr. 10) Schéma simulačního modelu virtuálního zprovoznění pro praktickou část	29
Obr. 11) Zadaná robotizovaná buňka (pohled z ptací perspektivy)	30
Obr. 12) Ganttův diagram operací pro výrobu jednice	31
Obr. 13) Pracovní obálka robotu IRB 1200-5/0.9 (vlevo) a robotu IRB 1200-7/0,7 (vpravo)	31
Obr. 14) Schéma řízení celé buňky z praktické části	32
Obr. 15) 3D model lineární jednotky Güdel TMF-1	33
Obr. 16) Sponkovačka Autotec 246 s automatickou doplňovací stanicí firmy BeA [27]	34

Obr. 17) 3D model použitého gripperu s přísavkami firmy SMC.....	35
Obr. 18) CNC frézovací router s vakuovým upínacím stolem	35
Obr. 19) 3D model pásového dopravníku Alutec KK	36
Obr. 20) 3D model zásobníku se základnou pro přechycení polotovarů robotem	36
Obr. 21) Grafická reprezentace rozdělení norem typu A, B a C [31].....	37
Obr. 22) Rizikový graf z normy ČSN EN ISO 13849-1 [32].....	38
Obr. 23) Popis bezpečnostních prvků použitých v zadané výrobní buňce	39
Obr. 24) Výsledný simulační model po sestavení v PS.....	41
Obr. 25) Definice pohybu CNC routeru v Kinematics Editoru v PS	42
Obr. 26) Dialogové okno softwaru VRC ABB Server pro propojení PS s RobotStudiem	43
Obr. 27) Nastavení verze kontroléru robotu pro simulaci v PS.....	43
Obr. 28) Souřadné systémy nástroje manipulačního (vlevo) a balicího robotu (vpravo).....	44
Obr. 29) Přiřazení rozšiřující osy robotu	45
Obr. 30) Příklad tvorby drah pohybů robotu v Path Editoru	45
Obr. 31) Obecná struktura kódu pro tvorbu vlastního OLP příkazu v jazyku XML.....	47
Obr. 32) Příklad sestaveného programu z jednotlivých operací manipulačního robotu	47
Obr. 33) Materiálový tok pro výrobu 3 velikostí gravírovaných desek	48
Obr. 34) Všechny signály balicího robotu z praktické části	49
Obr. 35) Použité senzory v rámci simulace	49
Obr. 36) Nastavení dopravníku u stanice pro přechycení polotovarů	50
Obr. 37) Vstupní (vlevo) a výstupní (vpravo) signály logického bloku robotů ABB	51
Obr. 38) Ukázka vnitřních proměnných tzv. parametrů logického bloku řízení robotů	51
Obr. 39) Vývojový diagram funkčního bloku pro volání robotických procedur – R_Control.53	
Obr. 40) Vývojový diagram funkčního bloku pro řízení stavů v celé buňce – Cell Control....55	
Obr. 41) Vývojový diagram funkčního bloku řízení výrobního programu – Program Flow ...57	
Obr. 42) Vývojový diagram funkčního bloku pro volání procedur pro doplnění spon.....58	
Obr. 43) Prioritní vyhodnocování safety programu [34]	59
Obr. 44) Nastavení jednotlivých kanálů na kartě bezpečnostních vstupů	60
Obr. 45) Použité zdvojené vstupy bezpečnostních prvků.....	60
Obr. 46) Sestavený blok E-stopu s NC kontaktem	60
Obr. 47) Vývojový diagram bloku starajícího se o funkční bezpečnost	61
Obr. 48) Vizualizace HMI panelu domovské stránky	62
Obr. 49) Výpis všech možných stavů indikátorů režimů v buňce z vizualizace HMI panelu..63	
Obr. 50) Vyskakovací obrazovka při vystavení pořadavku pro vstup do buňky	64
Obr. 51) Nastavení vnitřních systémových hodin v PLC	64
Obr. 52) Visual Basic skript sloužící pro spouštění vyskakovacích obrazovek	65
Obr. 53) Příklad mapování signálů v softwaru PS (vlevo) a TIA Portal (vpravo)	65
Obr. 54) Nastavení PLCsim Advanced	66
Obr. 55) Připojení komunikace pro PLCsim Advanced v PS	66
Obr. 56) Nastavení komunikace a stažení řídicího programu do virtuální PLC jednotky	67
Obr. 57) Ověření robotického programu balicího robotu.....	68
Obr. 58) Ukázka doplněné instrukce SearchL v robotickém programu	68
Obr. 59) Ověření robotického programu manipulačního robotu	69