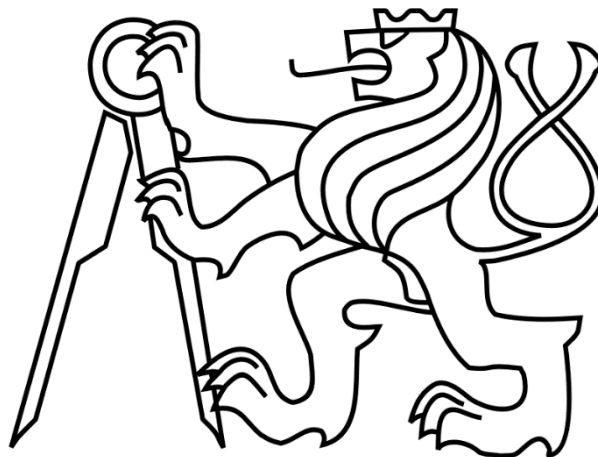


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení



Diplomová práce

Příprava a zprovoznění virtuálních pracovišť pro výuku PLC řízení

2022

Bc. Oskar Lamač

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Lamač** Jméno: **Oskar** Osobní číslo: **465391**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav výrobních strojů a zařízení**
Studijní program: **Strojní inženýrství**
Studijní obor: **Výrobní stroje a zařízení**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Příprava a zprovoznění virtuálních pracovišť pro výuku PLC řízení

Název diplomové práce anglicky:

Preparation and commissioning of virtual workstations for teaching PLC control

Pokyny pro vypracování:

Popis tématu: Cílem práce je zmapovat možnosti současných softwarů, které jsou využitelné pro virtuální zprovoznění PLC. Ve zvoleném software bude zpracováno několik virtuálních pracovišť pro využití ve výuce PLC řízení.;
Osnova práce: Zpracujte rešerši softwarů využitelných pro virtuální zprovoznění PLC včetně jejich komunikačních možností. Pro nejhodnější řešení zpracujte přehled jeho/jejich funkcionality. Ve vybraném SW zpracujte minimálně 4 virtuální modely pracovišť pro využití ve výuce PLC řízení. Minimálně jednu z virtuálních úloh zpracujte jako kompletní demo včetně PLC programu. Součástí všech vytvořených úloh bude uživatelská dokumentace.; Rozsah textové části: 60-80 stran;

Seznam doporučené literatury:

[1] BERGER, Hans. Automating with SIMATIC S7-1500 Configuring, Programming, Motion Control and Security inside TIA Portal. Erlangen: PUBLICIS, 2013. ISBN 9783895784040.;

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Lukáš Novotný, Ph.D. ústav výrobních strojů a zařízení FS

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

Datum zadání diplomové práce: **29.03.2022**

Termín odevzdání diplomové práce: **25.07.2022**

Platnost zadání diplomové práce: **30.09.2022**

Ing. Lukáš Novotný, Ph.D.
podpis vedoucí(ho) práce

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

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Prohlášení

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

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

V Praze dne 22.7.2022

.....

podpis

Poděkování

Rád bych tímto poděkoval svému vedoucímu diplomové práce panu Ing. Lukáši Novotnému, Ph.D. za odborné vedení v průběhu celé diplomové práce, cenné rady, věcné připomínky a velmi vstřícné jednání. Dále chci vyjádřit dík kolegovi Ing. Adolfu Valáškovvi za odborné a cenné rady.

Nakonec děkuji i svým rodičům za jejich vytrvalou podporu a trpělivost v průběhu celého studia.

Anotace

Autor:	Bc. Oskar Lamač
Název DP:	Příprava a zprovoznění virtuálních pracovišť pro výuku PLC řízení
Rozsah práce:	88 stran, 10 tabulek, 75 obrázků
Školní rok vyhotovení:	2021/2022
Škola:	České vysoké učení technické v Praze, Fakulta strojní
Ústav:	Ú12135 – Ústav výrobních strojů a zařízení
Vedoucí DP:	Ing. Lukáš Novotný, Ph.D.
Zadavatel:	ČVUT FS
Využití:	Možné využití ve výuce základů PLC programování
Klíčová slova:	virtuální zprovoznění, PLC program, PLC řízení, virtuální model, digitální dvojče, průmyslová komunikace
Anotace:	Diplomová práce se zabývá přípravou a zprovozněním virtuálních pracovišť pro výuku PLC řízení. Tvorba virtuálních pracovišť probíhá v softwaru pro virtuální zprovoznění, který vychází z rešerše. Jedno z virtuálních pracovišť je zpracováno jako kompletní demo včetně PLC programu. Hlavním výstupem práce jsou čtyři <i>state-of-the-art</i> virtuální modely výukových zařízení – digitální dvojčata programovatelných stavebnic Fischertechnik – včetně detailní uživatelské dokumentace.

Abstract

Author:	Bc. Oskar Lamač
Title of the diploma thesis:	Preparation and commissioning of virtual workstations for teaching PLC control
Extent:	88 pages, 10 tables, 75 figures
Academic year:	2021/2022
University:	CTU in Prague, Faculty of Mechanical Engineering
Department:	Ú12135 – Department of Production Machines and Equipment
Supervisor:	Ing. Lukáš Novotný, Ph.D.
Submitter:	CTU FME
Application:	Possible implementation in teaching the basics of PLC programming
Key words:	virtual commissioning, PLC program, PLC control, virtual model, digital twin, industrial communication
Abstract:	Diploma thesis deals with the preparation and commissioning of virtual workstations for teaching PLC control. The virtual workstations are created in a software which is based on research. One of the virtual workstations is made as a complete demo including a PLC program. The main output of the thesis is four state-of-the-art virtual models of teaching devices – digital twins of Fischertechnik programmable kits – including detailed user documentation.

Obsah

1	Úvod	11
2	Cíle práce	12
3	Stav problematiky	13
3.1	Virtuální zprovoznění	13
3.2	Virtuální zprovoznění ve výuce	14
3.3	Nástroje pro virtuální zprovoznění	14
3.3.1	NX Mechatronics Concept Designer	14
3.3.2	Tecnomatix Plant Simulation	15
3.3.3	DELMIA	15
3.3.4	MapleSim	15
3.3.5	Emulate3D	16
3.3.6	FastSuite	16
3.3.7	Simumatik	16
3.3.8	CIROS Studio	17
3.3.9	FE.Screen-Sim	17
3.3.10	RobotStudio	17
3.3.11	Factory I/O	18
3.3.12	Visual Components	18
3.4	Výběr ze dvou hlavních kandidátů	18
3.4.1	Bližší pohled na Factory I/O	18
3.4.2	Bližší pohled na Visual Components	20
3.4.3	Srovnání Factory I/O a Visual Components	22
3.4.4	Shrnutí	23
3.5	Virtuální PLC	23
3.6	Komunikační protokoly	23
3.6.1	Průmyslový Ethernet	23
3.6.2	Průmyslová komunikace v reálném čase	25
3.7	Přehled vybraných standardů průmyslového Ethernetu	26
3.7.1	EtherCAT	27
3.7.2	EtherNetIP	27
3.7.3	Ethernet Powerlink	27
3.7.4	Modbus/TCP	28
3.7.5	Profinet	28
3.7.6	OPC UA	29
3.8	Programovací nástroje automatizačních systémů	29
3.8.1	Siemens Totally Integrated Automation (TIA) Portal	30
3.8.2	TECO Mosaic	30
3.8.3	Beckhoff TwinCAT 3	31
3.9	Výstupy řešerše	32
3.9.1	Vybraný SW nástroj pro virtuální zprovoznění	32
3.9.2	Zvolený komunikační protokol	32
3.9.3	Podporování výrobci PLC, korespondující automatizační SW	32
4	Vlastní řešení	33
4.1	Výuka PLC řízení	33

4.2	Propojení vizualizačního softwaru s PLC	34
4.2.1	Propojení Visual Components se Siemens – OPC UA.....	34
4.2.2	Propojení Visual Components s Beckhoff – OPC UA.....	42
4.3	Tvorba modelu Dopravníku	45
4.3.1	Tvorba geometrie	45
4.3.2	Návrh ovládacích prvků virtuálního modelu	45
4.3.3	Oživení geometrie	46
4.3.4	Tvorba I/O pro propojení a řízení PLC	47
4.3.5	Dílčí shrnutí	49
4.4	Tvorba virtuálního modelu FT č.1: Robot.....	50
4.4.1	Tvorba geometrie	50
4.4.2	Návrh ovládacích prvků virtuálního modelu	50
4.4.3	Oživení geometrie	51
4.4.4	Kolize kinematiky	56
4.4.5	Tvorba I/O pro propojení a řízení PLC	57
4.4.6	Demo PLC projekt modelu Robotu.....	58
4.4.7	Dílčí shrnutí	59
4.5	Tvorba virtuálního modelu FT č.2: Regalový zakladač.....	60
4.5.1	Tvorba geometrie	60
4.5.2	Návrh ovládacích prvků virtuálního modelu	60
4.5.3	Oživení geometrie	61
4.5.4	Kolize kinematiky	66
4.5.5	Tvorba I/O pro propojení a řízení PLC	67
4.5.6	Dílčí shrnutí	68
4.6	Tvorba virtuálního modelu FT č.3: Třídíčka.....	69
4.6.1	Tvorba geometrie	69
4.6.2	Návrh ovládacích prvků virtuálního modelu	69
4.6.3	Oživení geometrie	70
4.6.4	Tvorba I/O pro propojení a řízení PLC	73
4.6.5	Dílčí shrnutí	74
4.7	Tvorba virtuálního modelu FT č.4: Multiprocesní stanice	75
4.7.1	Tvorba geometrie	75
4.7.2	Návrh ovládacích prvků virtuálního modelu	75
4.7.3	Oživení geometrie	76
4.7.4	Kolize kinematiky	78
4.7.5	Tvorba I/O pro propojení a řízení PLC	79
4.7.6	Dílčí shrnutí	80
5	Závěr.....	81
	Seznamy	83
	Seznam použité literatury	83
	Seznam tabulek	86
	Seznam obrázků	86
	Seznam použitého softwaru.....	87
	Seznam příloh	88

Seznam použitých zkratk

ASIC	Australian Securities & Investments Commission
CAD	Computer-Aided Design
CBA	Component Based Automation
CIP	Common Industrial Protocol
CNC	Computer Numerical Control
COM/DCOM	Component Object Model / Distributed Component Object Model
CPU	Central Processing Unit
DB	Data Block
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
ECAD	Electrical Computer-Aided Design
EHB	Elektrohängebahn
EtherCAT	Ethernet for Control Automation Technology
FB	Function Block
FBD	Function Block Diagram
FC	Function
FIO	Factory I/O
FPS	Frames Per Second
FT	Fischertechnik
FTP	File Transfer Protocol
FUN	Function
FUP	Funktionsplan
HD	High Definition
HDD	Hard Disk Drive
HIL	Hardware-in-the-Loop
HMI	Human-Machine Interface
HTTP	HyperText Transfer Protocol
HW	Hardware
I/O	Input/Output
IANA	Internet Assigned Numbers Authority
ID	Identification
IDA	International Development Association
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IPC	Industrial Personal Computer
IRT	Isochronous Real Time
ISO	International Organization for Standardization
IT	Information Technology
LD	Ladder Diagram
LED	Light-Emitting Diode
MAC	Media Access Control
MCAD	Mechanical Computer-Aided Design
MCD	Mechatronics Concept Designer
OB	Organization Block
ODVA	Open DeviceNet Vendor Association
OPC-DA	Open Platform Communications-Data Access

OPC-UA	Open Platform Communications-Unified Architecture
OSI	Open Systems Interconnection
PC	Personal Computer
PG/PC	Programmiergeräte/Personal Computer
PLC	Programmable Logic Controller
PNO	PROFIBUS Nutzerorganisation
POU	Program Organization Unit
PROG	Program
PS	Plant Simulation
PTP	Precision Time Protocol
RAM	Random Access Memory
RF	Reference Model
RS	Reset-Set
RT	Real-Time
RTT	Rotace-Translace-Translace
SCNM	Slot Communication Network Management
SIL	Software-in-the-Loop
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SP1	Service Pack 1
SQL	Structured Query Language
SSE2	Streaming SIMD Extensions 2
ST	Structured Text
SW	Software
TCP/IP	Transmission Control Protocol/Internet Protocol
TIA	Totally Integrated Automation
TMC	Trace Message Control
TTT	Translace-Translace-Translace
UDP	User Datagram Protocol
UR	Universal Robot
URL	Uniform Resource Locator
USB	Universal Serial Bus
VC	Visual Components
VR	Virtual Reality
XAE	eXtended Automation Engineering
XML	Extensible Markup Language

1 Úvod

Aktuálními trendy ve výrobním průmyslu – také nazývaným Průmysl 4.0 – jsou rychlejší vývoj novinek, co nejrychlejší zahájení výroby, snižování nákladů, avšak za současně zvyšující se kvality a flexibility výroby. Velice výhodně toho lze dosáhnout za pomoci virtuálního zprovoznění digitálních dvojčat strojů a výrobních linek. [1]

Tématem této práce je prozkoumání současných možností virtuálního zprovoznění a tvorba virtuálních pracovišť využitím softwarů jako např. Siemens NX MCD nebo Factory I/O, a to za konkrétním účelem zlepšení kvality výuky PLC řízení oproti stávajícímu řešení.

Výhodou virtuálního zprovoznění může být např. zkrácení celkového času realizace zařízení. Lze jej tak absolvovat např. i před dokončením montáže pracoviště. Navíc v případě nějaké chyby nedojde k poškození nákladné a zbrusu nové strojní techniky, což opět šetří čas i peníze. Ovšem pro dosažení tohoto konečného výsledku je zapotřebí vytvořit všechny příslušné mezikroky pro zajištění bezpečného, bezproblémového, responsivního a robustního řízení provozu pracoviště.

Tato práce zahrnuje zpracování rešerše softwarů využitelných pro virtuální zprovoznění PLC, včetně jejich komunikačních možností. Následovat bude tvorba několika virtuálních pracovišť ve vybraném softwaru, který vzejde z rešerše, a to včetně uživatelské dokumentace.

Výsledkem této práce by měly být *state-of-the-art* (nejmodernější) virtuální modely výukových zařízení – digitální dvojčata programovatelných stavebnic Fischertechnik. Tyto virtuální modely budou, mimo jiné, srozumitelně zdokumentovány, budou věrněji vystihovat realitu, budou responsivní, budou mít velmi dobré grafické podání, budou připojitelné z různých PLC od různých výrobců – a to i na dálku (on-line výuka). Zároveň studenty vytvořený PLC program na virtuálních modelech bude přenositelný i na fyzická zařízení.

2 Cíle práce

Hlavním cílem této práce je zmapování možností současných softwarů, které jsou využitelné pro virtuální zprovoznění PLC. Ve zvoleném softwaru bude následovat zpracování několika programovatelných virtuálních modelů pro využití ve výuce PLC řízení.

Dílčími cíli práce jsou:

- Zpracování rešerše softwarů využitelných pro virtuální zprovoznění PLC, včetně jejich komunikačních možností
- Zpracování přehledu funkcionality nejvhodnějšího řešení
- Ve vybraném softwaru zpracování min. 4 modelů virtuálních pracovišť pro využití ve výuce PLC řízení
- Zpracování minimálně jednoho z virtuálních pracovišť jako kompletní demo včetně PLC programu
- Vytvoření uživatelské dokumentace ke všem vytvořeným pracovištím

3 Stav problematiky

Následující kapitoly blíže popisují hlavní uzly potřebné pro vlastní řešení práce – softwarové nástroje pro virtuální zprovoznění a vizualizaci, komunikační protokoly a softwarové nástroje pro PLC programování automatizačních systémů. Na konci kap. 3 následuje shrnutí s vyvozenými závěry pro vlastní řešení.

3.1 Virtuální zprovoznění

Virtuálním zprovozněním je myšleno simulování, testování a optimalizace modelu fyzického výrobního systému v prostředí zahrnujícím jak hardware (HW), tak software (SW). Vlastnosti takového modelu jsou blízké reálnému stroji, a to včetně kinematiky, PLC programu i komunikace s řídicím systémem.

Hlavní výhodou virtuálního zprovoznění zařízení oproti fyzickému zprovoznění je zkrácení celkového času realizace zařízení. Značnou část programování lze řešit již paralelně s výrobou a montáží zařízení. Nutno podotknout, že virtuálním zprovozněním je možné se vyhnout např. poškození úplně nové linky při výskytu nějakého kritického problému, viz Obr. 1. Proto je velice výhodné mít možnost udělat toto prvotní nastavení virtuálně pomocí simulace zařízení (Obr. 2) za použití PC, a případně i virtuálních PLC. Samozřejmě není nutné dělat jen a pouze prvotní nastavení pro rozběhnutí linky. Virtuální simulace lze využít např. i k vyzkoušení nových změn v programu linky a následného nasazení do již běžícího provozu nebo i k monitorování běžící linky, případně pro školení operátorů. Tím vším virtuální zprovoznění nabízí kromě úspory času i možnou úsporu financí, nicméně je nutno brát v potaz také cenu tvorby virtuálního modelu a cenu samotného SW.



Obr. 1 – Náklady na nápravu chyb a problémů v závislosti na časovém okamžiku ve výrobě (hrubé plánování, detailní plánování, zprovoznění a produkce) [2]

Kategorie

Virtuální zprovoznění se provádí dvěma hlavními způsoby, a to tzv. *Software-in-the-loop* (SIL) nebo *Hardware-in-the-loop* (HIL). HIL simuluje digitální dvojče v PC, ale řízení probíhá reálným řídicím systémem, PLC stanicí. Kdežto SIL simuluje všechno, včetně řídicího systému, v prostředí PC. [1], [3]



Obr. 2 – Příklad virtuální výrobní linky v prostředí Factory I/O [4]

3.2 Virtuální zprovoznění ve výuce

Kromě výše zmíněných příkladů z průmyslu lze virtuální zprovoznění využít i pro účely výuky, např. PLC programování, kde si studenti mohou vytvořený PLC program rovnou otestovat a ladit. Vše se tak děje na jedné pracovní stanici (PC) a studentům to umožňuje pracovat i mimo hodiny výuky. V neposlední řadě to výuku činí více progresivní, aby držela krok s digitální dobou. Virtuálním zprovozněním se tak studenti mohou maximálně přiblížit fyzickému zprovoznění, aniž by se museli střídat u omezeného počtu fyzických zařízení.

V současné době se pro účely virtuálního zprovoznění při výuce používá škála softwaru od Siemens, tj. TIA Portal pro programování, PLCSIM Advanced pro emulaci reálného PLC a NX Mechatronics Concept Designer pro vizualizaci. [5] Zájmem této práce je posunout výukové prostředky směrem k lepší grafice vizualizace a rozšířit výběr kompatibilních PLC od dalších výrobců jako např. Beckhoff nebo TECO.

3.3 Nástroje pro virtuální zprovoznění

V této kapitole jsou stručné přehledy aktuálních vizualizačních SW pro virtuální zprovoznění, jejich podporované typy komunikace, vlastnosti a případně výhody a nevýhody.

3.3.1 NX Mechatronics Concept Designer



Mechatronics Concept Designer, dále jen MCD, je SW modul v NX od Siemens. V MCD jsou integrovány jak prvky pro mechanický design (MCAD), tak i elektrický design (ECAD) a automatizaci, případně mechatroniku. Umožňuje návrh a simulaci kinematických pohybů, virtuálního zprovoznění strojů a zařízení včetně pohonů, spínačů a sensorů. Disponuje mimo jiné následujícími funkcemi:

- Ověřování řídicích programů PLC s vizualizací
- Simulování chování stroje v reálném čase včetně PLC, CNC nebo sensorů
- Přímá integrace z CAD modelu v NX do MCD
- Snadné propojení s virtuálním PLC přes S7-PLCSIM od Siemens
- Propojení přes OPC protokol s externím HW

Tento SW je využíván jako současné řešení pro účely výuky PLC programování pro ověřování řídicích programů PLC. Výhodou MCD je úzká integrace mezi SW od Siemens, ale lze ho propojit s externími zařízeními pomocí protokolu OPC. [6]

3.3.2 Tecnomatix Plant Simulation



Plant Simulation, dále jen PS, je SW modul Tecnomatix opět od Siemens. PS umožňuje ladit řídicí automatizační logiku a PLC programy ve virtuálním prostředí. Na rozdíl od MCD, který je zaměřený spíše na konkrétní stroje, je PS zaměřen více na celé výrobní linky, a proto mimo jiné obsahuje funkce jako:

- Eliminování zpoždění (*bottleneck*) a zlepšení propustnosti linky
- Optimalizování spotřeby energie
- Testování automatizace, materiálových toků a řídicích programů PLC
- Podpora SIL i HIL simulace
- Snadné propojení s virtuálním PLC přes S7-PLCSIM od Siemens
- Propojení přes OPC protokol s externím HW
- Podpora jakéhokoliv PLC [7]

3.3.3 DELMIA

DELMIA je platforma spadající pod 3DEXperience (3DS) od Dassault Systèmes. K dispozici je velké množství modulů, především se zaměřením na automobilový průmysl, přičemž této práci je nejbližší Virtual Commissioning Expert, který nabízí mimo jiné následující funkce:

- Ověřování řídicích programů PLC s vizualizací
- Propojení přes OPC protokol
- Modelování sensorů
- Úzké integrování s CATIA Control Build
- Práce s PLC od různých výrobců (Siemens, Schneider, Omron, Rockwell Automation atd.)

SW klade důraz především na virtuální testování robotů, takže podporuje standard RRS-II. Disponuje údajně snadným mapováním I/O (vstupů/výstupů) a podporuje i HIL simulaci pro připojení reálných PLC. [8], [9]

3.3.4 MapleSim

MapleSim je SW společnosti MapleSoft. SW kromě vizualizace disponuje také např. pokročilým matematickým jádrem aplikace (*enginem*), návrhem řízení, pokročilou hydraulikou, nebo modelováním pneumatických systémů, pneumatik, baterií, přenosu tepla a hnacího ústrojí automobilů. Zaměřen je tedy z velké části na automobilový průmysl, ale nabízí mimo jiné následující funkce pro virtuální zprovoznění:

- Ověřování řídicích programů PLC s vizualizací
- Umožňuje simulace a analýzu kinematiky i dynamiky
- Používán především s PLC od B&R, Beckhoff a Rockwell Automation
- Zvládá HIL i SIL simulace v reálném čase
- Může běžet na serveru i PC s Windows 10, Linux nebo MacOS
- Potřebuje povolenou vnitřní komunikaci TCP/IP [10]

3.3.5 Emulate3D



Emulate3D je SW platforma od Rockwell Automation. Vedle virtuálního zprovoznění formou testování PLC programu umožňuje i analýzu toku materiálů nebo kinematiky modelu. Nabízí také vestavěné knihovny s hotovými modely zařízení pro automatizaci jako roboty, dopravníky, palety, třídičky a další. Obsahuje také knihovnu pro rychlou tvorbu PLC kódu pomocí kontaktních schémat (*Ladder Diagram*) nebo seznamu instrukcí (*Instruction List*). Mimo jiné nabízí následující funkce:

- Ověřování řídicích programů PLC s vizualizací
- Podpora importu až osmnácti CAD formátů
- Možnost připojení PLC od Siemens, Allen-Bradley, Mitsubishi pomocí jejich nativních protokolů
- Možnost připojení dalších PLC přes Modbus, OPC UA, EtherCAT, Profinet, Ethernet/IP atd.
- Může běžet na PC s Windows 10 [11], [12]

3.3.6 FastSuite **fastSUITE**

FastSuite je simulační produkt od CENIT určený k virtuálnímu zprovoznění. Kromě obecného virtuálního zprovoznění nabízí moduly konkrétně zaměřené např. na automatizaci automobilového průmyslu, leteckého průmyslu nebo svařování, řezání laserem, lakování, pískování, vroubkování, vrtání atd. Dále kromě návrhu a zprovoznění výrobního systému umožňuje také monitorování systému za účelem analýzy dat ze sensorů, predikování chyb a plánování údržby. Mimo jiné nabízí následující funkce:

- Ověřování řídicích programů PLC s vizualizací
- Propojení přes OPC-UA protokol s externím HW
- Integrace v reálném čase skrze *fieldbus*
- Může běžet na PC s Windows 7/10
- Podpora importu až dvaceti osmi CAD formátů
- Podpora SIL i HIL simulace
- Generování pohybů robotu a obráběcího programu přímo z CAD [13]

3.3.7 Simumatik



Simumatik je nezávislá simulační platforma zaměřená na virtuální zprovoznění. Od ostatních zástupců se liší hlavně tím, že je používán pouze skrze server (cloud). Ten může být jak lokální, tak vzdálený. Obsahuje nástroje pro hydrauliku, pneumatiku a elektriku a mimo jiné nabízí následující funkce:

- Ověřování řídicích programů PLC s vizualizací
- Může běžet na PC s Windows a MacOS nebo v prohlížeči Chrome, Firefox nebo Edge
- Propojení přes OPC-UA protokol s externím HW
- Licence se liší počtem možných připojených účastníků a kapacitou cloudového úložiště
- Nahrávání nebo tvorba modelů přes XML kód [15]

3.3.8 CIROS Studio



CIROS Studio je platforma pro vytváření simulačních modelů od Festo. Zahrnuje simulování, modelování a programování. Vedle verze Studio existuje i verze Education, která nemá možnost tvořit vlastní 3D modely. Verze Studio má navíc k dispozici připravené modely – přes 25 pracovních buněk s roboty a přes 30 sestav z oblasti automatizace a mimo jiné nabízí následující funkce:

- Ověřování řídicích programů PLC s vizualizací
- Úzká integrace se S7-PLCSIM a TIA Portal
- Může běžet na PC s Windows
- Kompatibilní s VR (používá OpenVR) za použití HTC VIVE
- Podpora importu až čtyř externích CAD formátů
- Podporuje paralelní simulace robotů/PLC
- Propojení přes OPC protokol s externím HW
- Podpora několika jazyků k programování robotů (Mitsubishi, KUKA, ABB, Stäubli) [16]

3.3.9 FE.Screen-Sim



FE.Screen-Sim je softwarové řešení pro simulaci a virtuální zprovoznění systémů od F.EE. Uvádí možnost nasazení až 100 000 elementů, 16 řadičů, 700 pohonů a cca 150 000 vstupů v jedné simulaci. SW se dělí na tzv. Core, který obsahuje komunikační rozhraní pro PLC, SQL, robotiku a další subsystemy. Pak následují volitelné moduly jako např. CAD Importer, sklad, záznam simulace, animace člověka, EHB, VR prohlížeč, tvorba logiky a další knihovny. Mimo jiné nabízí následující funkce:

- Ověřování řídicích programů PLC s vizualizací
- Podpora PLC od Siemens (S7), Allen-Bradley, Rockwell, Beckhoff, Fanuc
- Rozhraní pro MATLAB/Simulink, WinMOD atd.
- Podpora importu více jak pěti CAD formátů
- Tvorba logiky pomocí FUP a C#
- Automatické přiřazení signálů I/O z PLC
- Může běžet na PC s Windows [17]

3.3.10 RobotStudio RobotStudio®

RobotStudio je SW od ABB, především pro simulaci, programování a zprovoznění robotů od ABB. SW disponuje např. funkcí AutoPath a AutoConfiguration pro automatické generování trajektorie na základě geometrie vloženého CAD modelu a automatické generování konfigurace robotu na základě trajektorie. Dále také detekce kolize a velkou řadu dalších funkcí pro ladění robotu a také nabízí následující:

- Ověřování řídicích programů PLC s vizualizací
- Může běžet na PC s Windows 10
- Podpora importu až šesti CAD formátů [18]

3.3.11 Factory I/O



Factory I/O je předposlední zástupce z tohoto přehledu. Z důvodu většího zájmu o tento SW, je blíže popsán v kap. 3.4.1 a následně porovnán s Visual Components.

3.3.12 Visual Components



Visual Components je poslední zástupce z tohoto přehledu. Z důvodu většího zájmu o tento SW, je blíže popsán v kap. 3.4.2 a následně porovnán s Factory I/O.

3.4 Výběr ze dvou hlavních kandidátů

Následující kapitoly obsahují přehledy a porovnání dvou hlavních kandidátů pro výběr vhodného SW k vlastnímu řešení. Z přehledů v kap. 3.3 softwarů VC a FIO nejlépe splňují požadovanou funkcionalitu, kvalitní grafické podání, spolehlivou podporu a nabízejí možnosti nákupu licencí – ne pouze pronájem.

3.4.1 Bližší pohled na Factory I/O

Factory I/O, dále také zkráceně FIO, je simulační SW 3D továrny pro výuku automatizace a virtuálního zprovoznění od RealGames. Je navržen tak, aby se snadno používal, což také zahrnuje výběr z více než 20 modelových sestav nebo možnost si vytvořit vlastní uspořádání scény z knihovny nejběžnějších komponent (Obr. 3), kde většina modelů obsahuje digitální i analogové I/O (vstupy/výstupy). Připojení I/O na propojené PLC je vcelku přímočaré, stačí myši přetáhnout příslušný vstup nebo výstup na I/O PLC v sekci nastavení driverů.

Grafické provedení je na velmi vysoké úrovni s kvalitními texturami a realistickým stínováním a věrohodnými zvukovými efekty. Kameru lze ovládat za pomoci myši i klávesnice.

Co se týká podpory PLC, je možné si vybrat přímo edici podle výrobce PLC – Siemens, Allen-Bradley nebo Modbus & OPC a Automgen – nebo případně edici Ultimate, která obsahuje všechny možnosti komunikace. Edice Ultimate pak obsahuje následující drivery I/O k propojení jak virtuálních, tak reálných PLC:

- Advantech – USB 4750, USB 4704
- Allen-Bradley – Logix5000, Micro800, MicroLogix, SLC 5/05
- Automgen Server
- Control I/O (TCP/IP server)
- MHJ – WinPLC-Engine, WinSPS-S7
- Modbus TCP/IP – Client, Server
- OPC DA/UA – Client
- Siemens – LOGO!, S7-200/300/400, S7-1200/1500, S7-PLCSIM

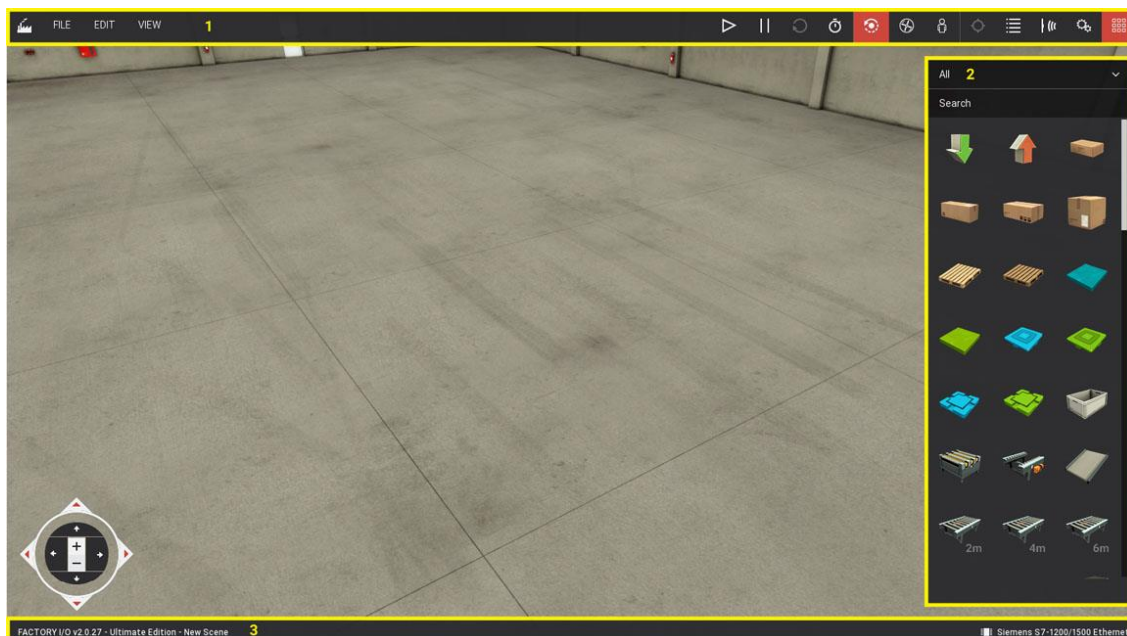
Systémovými požadavky jsou:

- Windows 7 SP1 nebo vyšší
- Procesor podporující balík instrukcí SSE2
- Grafická karta NVIDIA (2006 dále), AMD (2006 dále) nebo Intel (2012 dále)



Obr. 3 – Příklady komponent z knihovny prostředí Factory I/O [4]

Uživatelské rozhraní je poměrně minimalistické, viz Obr. 4. V části 1 se nachází nástrojová lišta obsahující tlačítka ovládání simulace (run/pause/reset, rychlost simulace), ovládání kamery, zobrazování tagů sensorů a pohonů a dále zobrazení Palety knihovny. Obsahuje také standardní kaskádová menu Soubor, Edit, Zobrazení s dalšími méně častěji používanými funkcemi. V zóně 2 se nachází již zmíněná Paleta knihovny, kde jsou zobrazeny veškeré dostupné díly a součásti pro vytvoření/úpravu testované scény, které lze pouhým “drag & drop“ vložit do scény. Oblast 3 slouží k zobrazení stavového řádku s verzí softwaru, názvu scény a vybraného driveru pro komunikaci, společně s indikátorem nízkého výkonu – pokud klesne update simulace pod 15 FPS (snímků za sekundu).

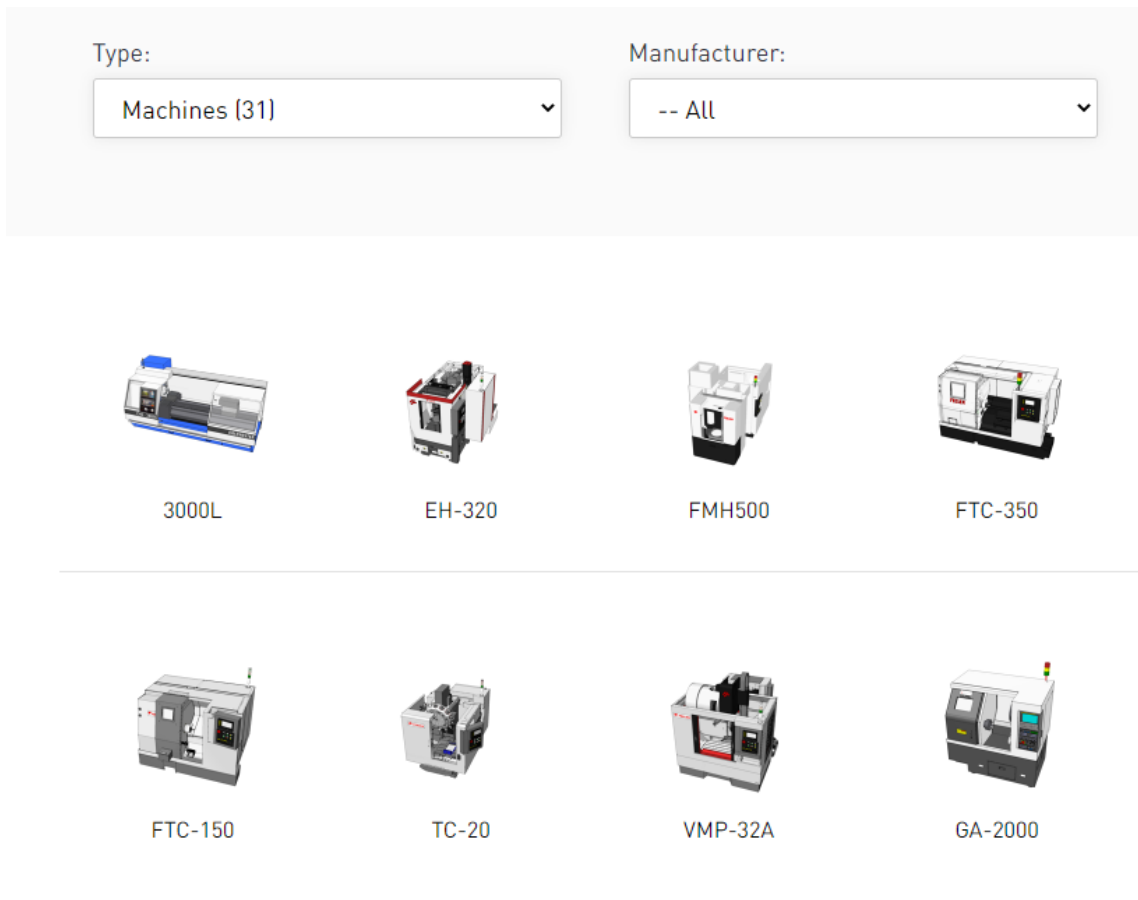


Obr. 4 – Uživatelské rozhraní prostředí Factory I/O [4]

Detailní popis ovládání a nastavení FIO se nachází pohodlně přímo na stránkách softwaru v sekci dokumentace: docs.factoryio.com. [4]

3.4.2 Bližší pohled na Visual Components

Visual Components, dále také zkráceně VC, je nezávislý simulační program zaměřený na virtuální zprovoznění. Nabízí řešení pro automobilový průmysl, elektroniku, farmaceutický průmysl, průmyslovou automatizaci, paletizaci, potravinářský průmysl atd. K dispozici je knihovna s předpřipravenými prvky, tzv. eCatalog (Obr. 5), ve kterém se nachází přes 2500 vymodelovaných součástí, napříč různými kategoriemi. Navíc eCatalog také obsahuje 35 již vymodelovaných demo sestav. Kromě možnosti použít tyto hotové součásti, je k dispozici také přímo v rozhraní CAD editor, kde lze vytvořit vlastní součást, anebo je možné taktéž importovat již externě vytvořený CAD model – podporuje import až třiceti tří formátů.

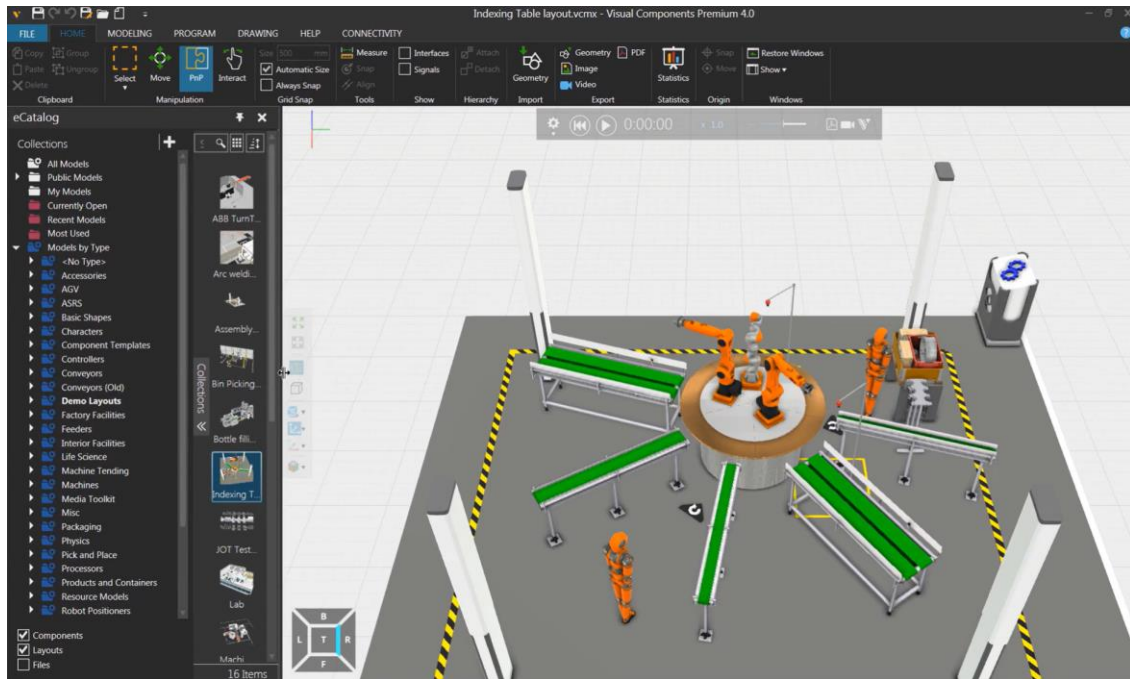


Obr. 5 – Příklad komponent z eCatalogu prostředí Visual Components [14]

Mimo to je k dispozici na webu rozsáhlá sekce Academy, kde jsou podrobně popsána, nebo za pomoci videí vysvětlena, různá řešení a nastavení týkající se VC. Vedle Academy je možné další informace získat během pár kliknutí také na oficiálním blogu nebo fóru.

Dalším komfortním prvkem je tzv. Process Modeling, který po rozmístění modelů simulované továrny dokáže sám vytvořit ovládací programy všech strojů a robotů (eventuelně i obsluhy). Uživatel musí pouze vyznačit tok materiálu/polotovaru/obrobku – odkud kam putuje.

Grafická reprezentace simulace je na velmi dobré úrovni (Obr. 6), i když se nejedná o realistickou 3D renderovanou grafiku, ale pouze o stínované a texturované CAD modely. Nejen grafická reprezentace umožňuje poměrně rozsáhlé nastavení zobrazení a dalších funkcí.



Obr. 6 – Uživatelské rozhraní prostředí Visual Components [14]

Všechny verze disponují možností připojit PLC skrze univerzální protokol OPC UA, Premium verze navíc nabízí následující funkce:

- Podpora S7-PLCSIM
- Připojení WinMOD a SIMIT
- Nativní připojení pro roboty Fanuc
- VRC připojení pro UR (Universal Robot) a Stäubli
- Doplňkové funkce pro programování robotů (CAD data pro dráhy robotu, vizualizace lakování)
- Vizualizace ve VR [14]

Systémovými požadavky (minimální) jsou:

- Ekvivalent procesoru Intel i5-3xxx
- 8GB RAM
- HDD s 3GB volného místa
- Grafická karta Intel HD Graphics 4400 (integrována)
- Rozlišení displeje alespoň 1280x1024
- Myš se třemi tlačítky (L, R, prostřední)
- 64-bitový operační systém Windows 8.1 nebo Windows 10
- Další požadavky (např. na licenční server) jsou k dispozici zde: <https://www.visualcomponents.com/system-requirements/>

Detailní popis ovládání a nastavení Visual Components se nachází pohodlně přímo v sekci Academy na webu: academy.visualcomponents.com. [14]

3.4.3 Srovnání Factory I/O a Visual Components

V této kapitole se nachází stručné porovnání dvou softwarových nástrojů pro virtuální zprovoznění, které vzešly z rešerše. Porovnání je provedeno především z pohledu virtuálního zprovoznění pro výuku PLC programování, přestože softwary mohou pokrýt daleko větší spektrum využití. Porovnávají jsou nejvyšší licence softwarů – FIO Ultimate a VC Premium.

Podpora PLC výrobců (možnosti komunikace)

FIO nabízí početně více možností komunikace; podporuje nativní připojení pro Siemens a Allen-Bradley, a navíc i dva univerzální komunikační protokoly Modbus TCP/IP a OPC UA. VC na druhé straně podporuje nativně pouze Siemens a univerzální pouze OPC UA.

Prakticky nám ale pro potřeby této práce stačí jeden univerzální protokol, kterým by bylo možné propojit všechny potřebné výrobce PLC. Těmi jsou Siemens, Beckhoff a TECO. Siemens a Beckhoff je možné vcelku snadno propojit přes OPC UA, nicméně připojení PLC od TECO je složitější. OPC UA momentálně oficiálně nepodporuje vůbec a pomocí Modbus se nejedná o stejně přímočarý úkol, jako při použití OPC UA.

Lehce navrch má tedy v tomto případě FIO.

Práce se softwarem z pohledu uživatele

Tento odstavce má za cíl přiblížit složitost práce se SW. Což je vedle uvedených výčtů parametrů a funkcí SW obtížněji kvantifikovatelné, nicméně taktéž důležité. Tato sekce je hodnocena autorem této práce a může tak převládat subjektivní pohled na věc.

V ohledu složitosti práce se SW je FIO vytvořeno ve smyslu co největší jednoduchosti. Nicméně pro účely této práce je to spíše na škodu. Sice je pracovní prostředí velice přehledné, ale chybí jakékoliv popisky u ovládacích prvků, které by začátečníkům přiblížily, co jaké tlačítko dělá. To samé v případě nastavování I/O u modelových komponent. Dále při importu tagů z OPC serveru úplně chybí možnost si manuálně vybrat konkrétní tagy z PLC programu – místo toho se nahrají úplně všechny, a to vstupy a výstupy vůbec nesouvisející s nahraným PLC programem. Je zde pouze možnost tagy vyfiltrovat podle názvu – což by nevadilo, pokud začínáme úplně nový projekt, a tudíž jsme již seznámeni s tímto principem importu tagů, ale pokud chceme nahrát a otestovat již existující PLC program, který nebyl pro FIO takto přizpůsoben, import je zbytečně obtížný.

VC prakticky ve všech ohledech volí směr opačný – co největší možnosti nastavení a úprav přímo v softwaru pro virtuální zprovoznění. Oproti FIO sice rozhraní zabírá více prostoru obrazovky, ale umožňuje tím vyšší úroveň přizpůsobení a nabízí více možností při tvorbě virtuálních modelů. V případě zmíněného importu tagů z OPC serveru lze pracovat s jasně srozumitelnou strukturou tagů. Ve výsledku více přehlednou než v případě až přílišné jednoduchosti FIO.

Grafická reprezentace

Grafická reprezentace je na první pohled lepší v případě FIO, které disponuje grafikou renderovanou ve 3D enginu s důkladným stínováním, odlesky a kvalitními texturami. Grafika v případě VC nedosahuje takové fotorealistické kvality, ale na testování PLC programů je reprezentace plně dostačující a zároveň krokem kupředu oproti stávajícímu řešení (Siemens NX MCD). Navíc oproti FIO obsahuje VC možnost si reprezentaci různě upravit nebo přepnout

(stínování, odlesky, zobrazení drátových modelů apod.), např. pro snížení potřebného výpočetního výkonu.

Tvorba virtuálních sestav

Oba softwary disponují připravenými modely i celými připravenými sestavami. Nicméně poměr připravených modelů jasně hovoří ve prospěch VC (2500+ modelů) oproti FIO (pouze desítky modelů). Kromě toho nabízí VC tvorbu nebo import vlastních CAD modelů přímo v prostředí VC, což je poměrně zásadní funkce pro tvorbu virtuálních modelů založených na fyzickém zařízení, které nemusí být reprezentováno v knihovně komponent. VC také umožňuje jak importovanou geometrii, tak i předpřipravené modely dále programovat pomocí Python skriptů.

3.4.4 Shrnutí

Přestože obě řešení jsou schopna plně zastoupit žádanou funkci při virtuálním zprovoznění testováním PLC programů, tak bez ohledu na licencování a cenu vychází Visual Components jako vhodnější řešení. Factory I/O sice na první pohled zaujme svou jednoduchostí a kvalitní grafickou reprezentací, ale jednoduchost je v některých ohledech spíše na škodu. Největší nevýhodou FIO je pro řešení této práce nemožnost importovat vlastní CAD geometrii a z toho vyplývající závislost pouze na vestavěné knihovně komponent. Visual Components je komplexnější řešení, které obsahuje moduly přesahující testování PLC programování a bylo by možné ho využít i např. při výuce programování robotů. Zároveň jeho důraz na rozsáhlou „nastavitelnost“ všeho přes rozhraní, grafickou reprezentaci, způsobů připojení až po import tagů z OPC serveru spíše usnadňuje práci při virtuálním zprovoznění. V neposlední řadě je výhodou VC i přesah do průmyslové praxe, kde se komerčně využívá.

3.5 Virtuální PLC

Z hlediska funkce existují dva typy virtuálních PLC: simulované a emulované. Simulované slouží pouze pro stažení a testování uživatelského programu, jedná se pouze o SW – což teoreticky stačí na většinu aplikací. Ve skutečnosti se ale jedná pouze o malou část toho, co reprezentuje skutečné PLC. “Téměř emulované“ PLC se dokáže např. připojit pomocí jiného Ethernetového rozhraní a komunikovat i směrem “ven“ např. s reálným PLC/HMI, tj. kromě komunikace s PC (*softbus*) může využít např. protokol TCP/IP. Příkladem jsou PLCSIM (simulované PLC) a PLCSIM Advanced (velice blízko emulovanému PLC) od Siemens. [19]

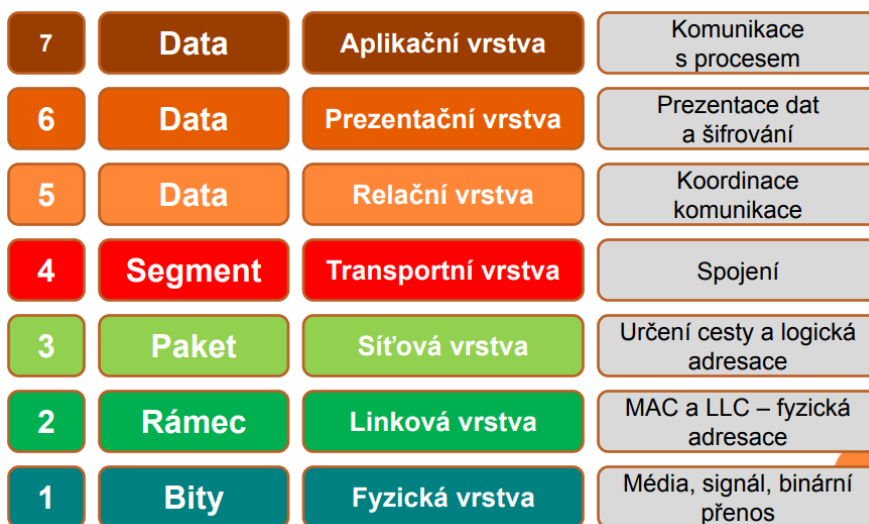
3.6 Komunikační protokoly

Tato kapitola popisuje podrobněji komunikační protokoly se zaměřením na průmyslový Ethernet a komunikaci v reálném čase. Z pohledu této práce se jedná o jeden z hlavních pilířů – propojení rozhraní na PLC programování s (virtuálním) PLC a posléze i s vizualizačním SW pro virtuální zprovoznění.

3.6.1 Průmyslový Ethernet

Průmyslový Ethernet je upravená verze Ethernetu pro vysokorychlostní komunikaci mezi automatizačními systémy v reálném čase. Nejedná se o nový standard, ale spíše znovuobjevení

Ethernetu oborem průmyslové automatizace s příchodem vlny IT v období 80. let, resp. 90. let minulého století. Ethernet jako takový je komunikační standard uznávaný standardizačními organizacemi IEEE i ISO. A to přesto, že nesledoval sedmivrstvý referenční model otevřené komunikace ISO/OSI (Obr. 7), doporučený právě organizací ISO. V průběhu minulých dekad se přenosové rychlosti dostaly od 1 Mb/s na 400 Gb/s pomocí běžné MAC (*Media Access Control*) specifikace. V dnešní době se rychlosti průmyslového Ethernetu pohybují od 10 Mb/s do 1 Gb/s, přičemž nejčastěji se vyskytuje rychlost 100 Mb/s [20], [21], [22], [23]

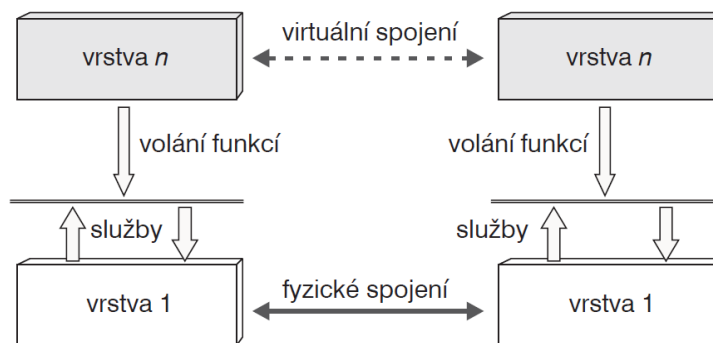


Obr. 7 – Schéma vrstev architektury komunikačního kanálu – ISO/OSI [20]

Ethernet TCP/IP

Na Ethernetu TCP/IP, ze kterého vychází průmyslový Ethernet, lze názorně ukázat základní principy jeho fungování.

Ethernet TCP/IP se mapuje do prvních čtyř vrstev RM ISO/OSI (Obr. 7), tj. fyzické a spojové (linkové) vrstvy, síťové vrstvy (protokoly IP) a transportní vrstvy (protokoly TCP). Spojení mezi účastníky se virtuálně odehrává na každé vrstvě, ale fyzické spojení a přenos dat je pouze na vrstvě první. Účastníci tak např. pomocí poslední vrstvy 7 (aplikace na PC) komunikují bez znalostí funkcí zbylých vrstev. Vrstvy mají dosah pouze na jednu nejbližší vyšší a jednu nejbližší nižší vrstvu. Každá vrstva poskytuje služby vrstvě nejbližší vyšší na základě jejího požadavku (volání funkce), viz Obr. 8. Další podrobnosti lze dohledat v [24].

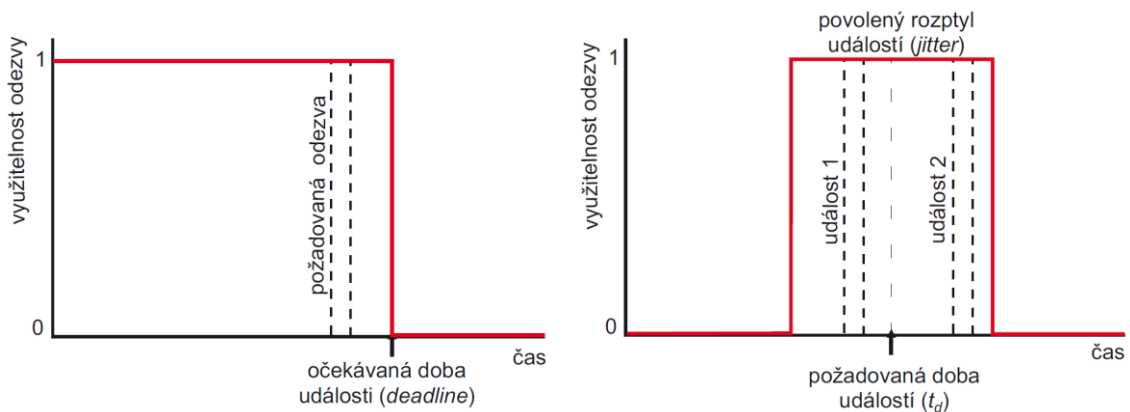


Obr. 8 – Přenos zpráv mezi vrstvami modelu OSI [24]

3.6.2 Průmyslová komunikace v reálném čase

Nezbytným předpokladem pro činnost řídicích systémů v reálném čase je determinismus komunikace. Nicméně zmiňovaný Ethernet TCP/IP je nedeterministický, a proto je nutné ho upravit a doplnit, aby bylo dosaženo příslušného stupně determinismu. [25]

Pod pojmem reálný čas jsou schovány dvě vlastnosti: současnost a včasnost. **Současnost** jako schopnost korelovaně zpracovávat data více vstupů najednou nebo na Obr. 9 jako povolené toleranční časové pásmo v okolí jmenovitého okamžiku, ve kterém se musí uskutečnit obě události vstupující do korelace. **Včasnost** poté jako požadovanou dobu odezvy řídicího systému – která je v průmyslové automatizaci mnohem důležitější než samotná rychlost výpočtu – nebo na Obr. 9 jako interval, ve kterém musí řídicí systém zajistit reakci na událost. [25]



Obr. 9 – Včasnost (vlevo) a současnost (vpravo) [25]

Vyšších stupňů determinismu lze dosáhnout např. již zmíněným použitím prepínačů (*switch*), plně duplexním přenosem, vyšší přenosovou rychlostí, použitím protokolu UDP místo TCP, segmentací sítě, synchronizací metodou PTP nebo použitím prioritních slotů. [25]

Při použití desetinásobně **vyšší přenosové rychlosti** se zkracuje doba potřebná na přenos jednoho paketu na desetinu. [25]

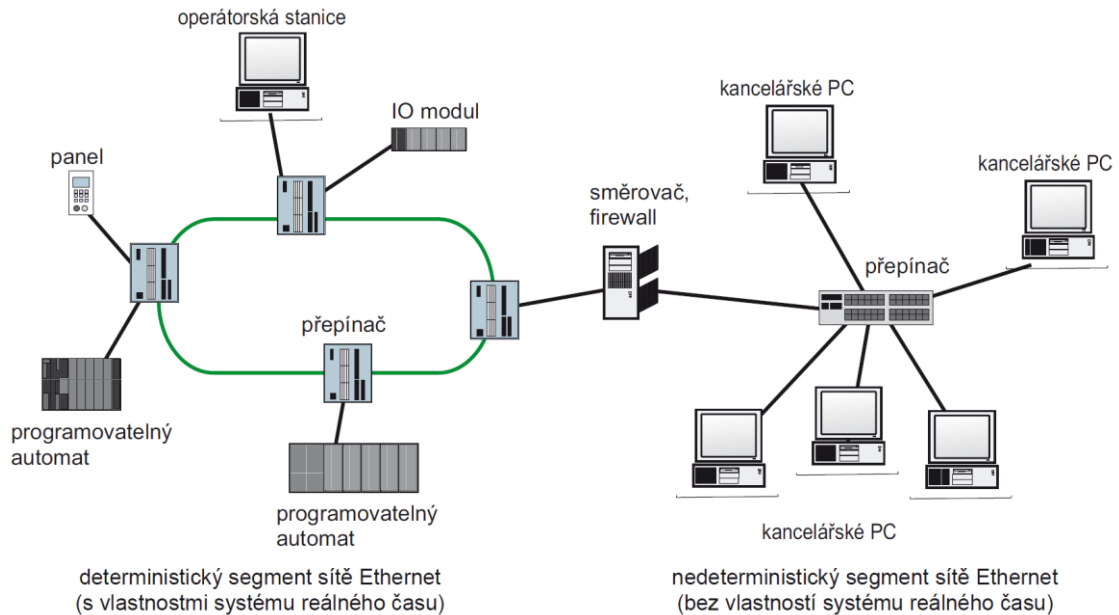
Protokol UDP na rozdíl od TCP posílá navzájem nezávislá data, nezná potvrzování zpráv ani ustavení a ukončení spojení. Při chybě přenosu tak data mohou být ihned přenesena v dalším cyklu, kdežto TCP při poruše opakuje přenos tak dlouho, dokud nejsou data správně přijata – pro jeho rychlost a jednoduchost může být v průmyslu upřednostňován právě UDP. [25]

Při použití **prioritních slotů** je v protokolu spojové vrstvy v síti Ethernet umístěno pole s názvem *tag*, které nese informaci o prioritě. Toho dokážou využít prepínače a směrovat příslušné časově prioritní zprávy do příslušných segmentů. [25]

Segmentování sítě se využívá pro oddělení deterministické části sítě, ve které se používá např. menší počet prepínačů. Protože čím více prepínačů je v cestě signálu, tím je větší zpoždění (*ping*). Příklad segmentování sítě je na Obr. 10, kde rozhraní mezi deterministickou částí a nedeterministickou částí musí být přesně řízeno prepínačem, případně lze také oddělit směrovačem a firewallem. [25]

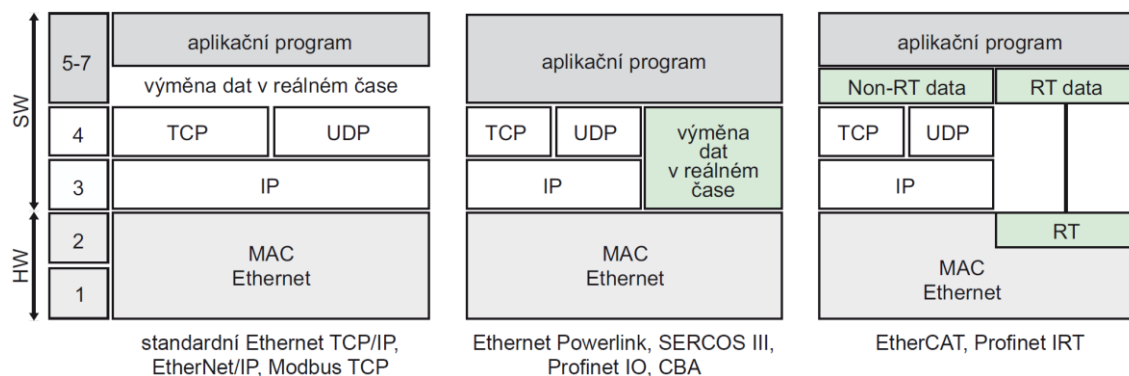
Pro velkou třídu deterministických úloh, jako je např. řízení pohonů (poloha a pohyb) nelze dosáhnout synchronizace (*jitter*), Obr. 9, použitím pouze variant Ethernetu TCP/UDP/IP. V případě průmyslových sběrnic vyvinutých speciálně pro řízení v reálném čase se synchronizace řeší jednoznačným plánováním času předání zpráv jednotlivými účastníky. Systém je tak deterministický, tj. lze přesně určit okamžik provedení akcí. V případě řízení sběrnice způsobem

master-slave (sběrnice kategorie *fieldbus*) je zcela pod kontrolou synchronní chod řídicích systémů, např. řídicích jednotek pohonů os obráběcích strojů. Jelikož je v případě protokolu Ethernet systém zatížen širokým tolerančním pásmem (*jitter*) v okolí požadovaného okamžiku, je systém nedeterministický. Jedním ze způsobů zavedení synchronizace je použití metody PTP (*Precision Time Protocol*) podle IEEE 1588. Principem této metody synchronizace je distribuce hodin reálného času, kterou lze dosáhnout lepší synchronizace než např. u zmíněné sběrnice *fieldbus*. [25]



Obr. 10 – Segmentování sítě Ethernet [25]

Na Obr. 11 jsou architektury variant průmyslového Ethernetu, kde nalevo jsou varianty používající standardní protokoly TCP/UDP/IP a determinismu dosahují kombinací výše zmíněných prostředků. Napravo jsou pak zástupci, u nichž nekritické zprávy jdou skrze TCP/UDP/IP a časově kritické zprávy TCP/UDP/IP obcházejí buď softwarově nebo hardwarově. [25]



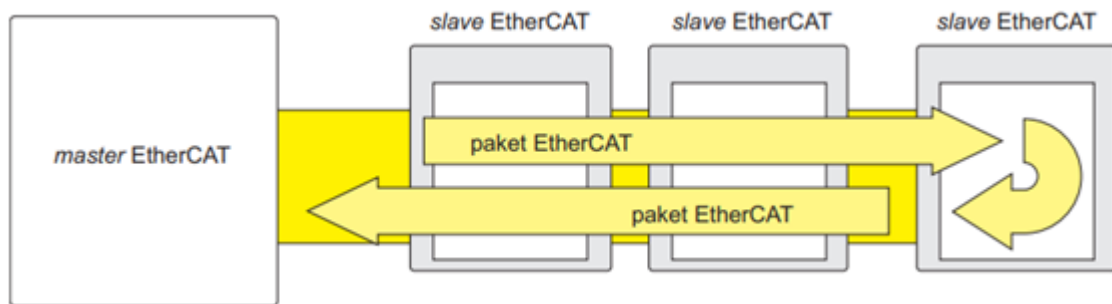
Obr. 11 – Architektury variant průmyslového Ethernetu (RT – Real-Time, reálný čas) [25]

3.7 Přehled vybraných standardů průmyslového Ethernetu

Tato kapitola obsahuje přehled vybraných standardů průmyslového Ethernetu, které lze využít k fyzickému i virtuálnímu zprovoznění.

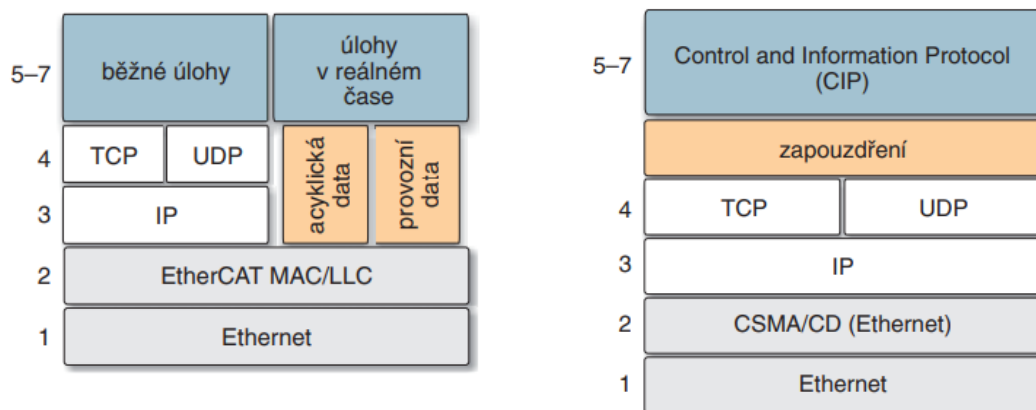
3.7.1 EtherCAT

Standard EtherCAT (*Ethernet for Control Automation Technology*) je modifikace Ethernetu s důrazem na rychlý přenos dat s krátkým komunikačním cyklem (řízení v reálném čase) od firmy Beckhoff (Obr. 13). K dosažení vysokého výkonu zcela nahrazuje přístupovou vrstvu MAC standardního Ethernetu. Využívá principu *master/slave*, kde *master* zařízení (např. řídicí systém) posílá ethernetové rámce do jednotek *slave* (např. PLC) a tento rámec prochází posléze všemi jednotkami *slave*, viz Obr. 12. Tyto jednotky musejí mít speciální HW (ASIC), který dokáže zpracovávat rámce za běhu.



Obr. 12 – Průchod paketu sítě EtherCAT [26]

EtherCAT obsahuje vestavěnou diagnostiku, umožňuje uspořádání do téměř všech typů topologie s redundancí a je kompatibilní se standardním Ethernetem. Také dosahuje velmi vysokého stupně determinismu, což je výhodné např. při řízení pohonů. [26], [27]



Obr. 13 – Komunikační modely EtherCAT (vlevo) a EtherNet/IP (vpravo) [28]

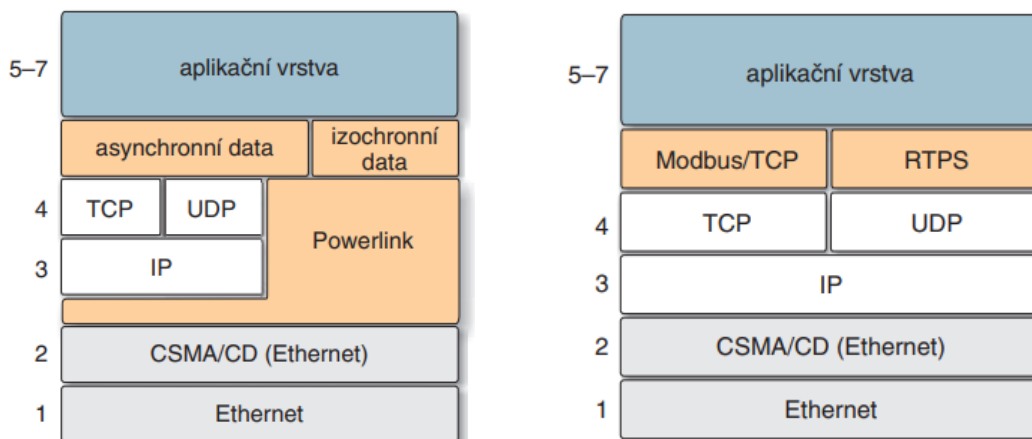
3.7.2 EtherNetIP

EtherNetIP je rozšíření Ethernetu asociací ODVA (Obr. 13). Použita je metoda uceleného přenosu dat *producer-consumer* (najednou více odběratelům). Necyklická data jsou přenášena zabezpečeným TCP protokolem a časově kritická protokolem UDP. Jednotná aplikační vrstva také umožňuje propojení I/O protokolem CIP. [26], [27]

3.7.3 Ethernet Powerlink

Ethernet Powerlink je velmi rozšířeným standardem od firmy B&R (Obr. 14). Vychází ze standardu Ethernet, nepotřebuje žádný speciální HW a má velmi dobré vlastnosti reálného času. Nejkratší doba cyklu (perioda) odpovídá 200 μ s, ve které může být obslouženo až osm tzv.

rychlých zařízení a další tzv. pomalá zařízení. Při periodě 1 ms zvládne obsloužit až 30 rychlých zařízení. Rychlá zařízení mohou vysílat v každém taktu sběrnice, kdežto z pomalých může v každém taktu vyslat vždy právě jen jedno. Přijímat mohou všechna zařízení, což se nazývá *peer-to-peer* (přímá) komunikace. Díky řídicí metodě SCNM (*Slot Communication Network Management*) má v daném okamžiku k přenosovým datům přístup pouze jedno zařízení, a tak nemůže docházet ke kolizím a je zajištěn striktně deterministický přenos. [27], [29]



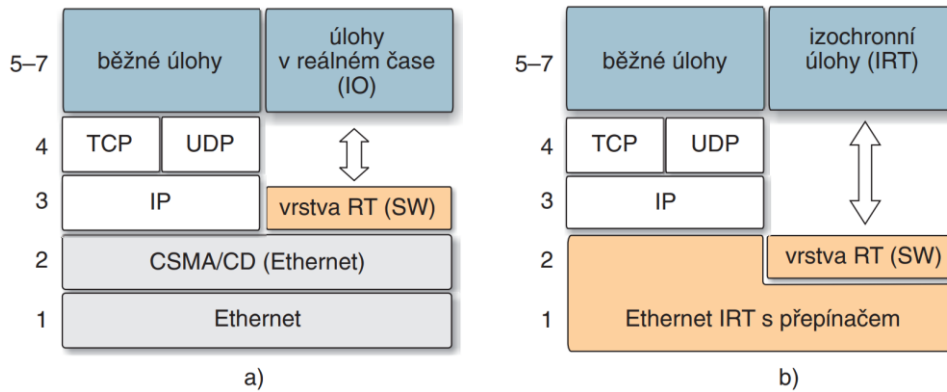
Obr. 14 – Komunikační modely Ethernet Powerlink (vlevo) a Modbus/TCP (vpravo) [28]

3.7.4 Modbus/TCP

Modbus/TCP (Obr. 14) vychází z historicky vůbec nejrozšířenějšího komunikačního protokolu automatizace Modbus od firmy Modicon (později AEG a Schneider Electric). Jeho specifikace je volně dostupná a od jeho uvedení nedošlo k žádné změně. U Modbus/TCP probíhá komunikace principem klient/server, resp. *master/slave*. Umožňuje přenášet dat po různých sběrnících a sítích (RS-232, RS-485, Ethernet TCP/IP atd.). Z toho vyplývá velice snadné nasazení do systémů podporující protokol TCP/IP. Jedná se o rychlou, efektivní a jednu z nejrozšířenějších průmyslových komunikací, a to i přesto, že zdaleka nedosahuje nejlepších parametrů. [27], [28]

3.7.5 Profinet

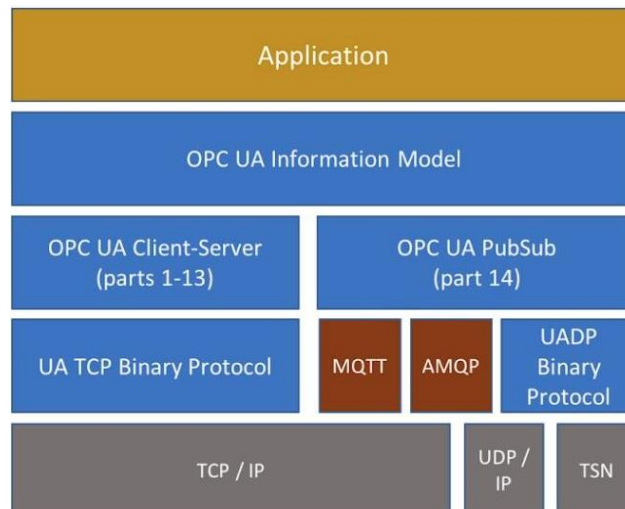
Profinet byl vyvinut organizací PNO za přispění Siemens (Obr. 15). Jeho architektura je koncipována jako modulární, takže lze funkce do jisté míry přizpůsobit dané úloze. Nyní je jedním z předních standardů automatizace, také díky snadné integraci do systémů *fieldbus*. Komunikace probíhá ve dvou módech, kde prvním módem je Profinet IO, určený k přenosu v reálném čase a k izochronnímu přenosu. Druhým módem je Profinet CBA, který přenáší zprávy prostřednictvím TCP/IP bez požadavku na doručení v reálném čase. Pro úlohy v reálném čase s tvrdými požadavky na dodržení doby odezvy a synchronizace se používá Profinet verze V3 neboli Profinet IRT, který překlenuje vrstvy TCP/IP pomocí tzv. *HW bypass*. [28], [29]



Obr. 15 – Komunikační model Profinet IO (a) a Profinet IRT (b) [28]

3.7.6 OPC UA

OPC UA je nástupce původní specifikace OPC, založené na technologii COM/DCOM od firmy Microsoft (Obr. 16). Původní OPC tak fungovalo pouze na Windows a novější varianta UA (*Unified Architecture*) proto využívá standardů jako TCP/IP, HTTP a SOAP, což umožňuje použití i mimo Windows (iOS, Android, Linux atd.) nebo přímo zabudování do PLC automatu (formou serveru). Poté se stačí externě připojit k serveru jako OPC klient a lze rovnou pozorovat data z PLC. Může nastat ale situace, kdy vestavěný OPC server přestane zvládat dotazy od velkého množství OPC klientů. [30], [31]



Obr. 16 – Komunikační architektura OPC UA [32]

3.8 Programovací nástroje automatizačních systémů

Dále zkráceně automatizační SW, který je zapotřebí pro vytváření samotných automatizačních programů. Spolu s množstvím dalších automatizačních funkcí jsou tyto SW nabízeny několika výrobci. Dále jsou ve vazbě k této práci přiblíženi vybraní zástupci od Siemens, Beckhoff a TECO.

3.8.1 Siemens Totally Integrated Automation (TIA) Portal

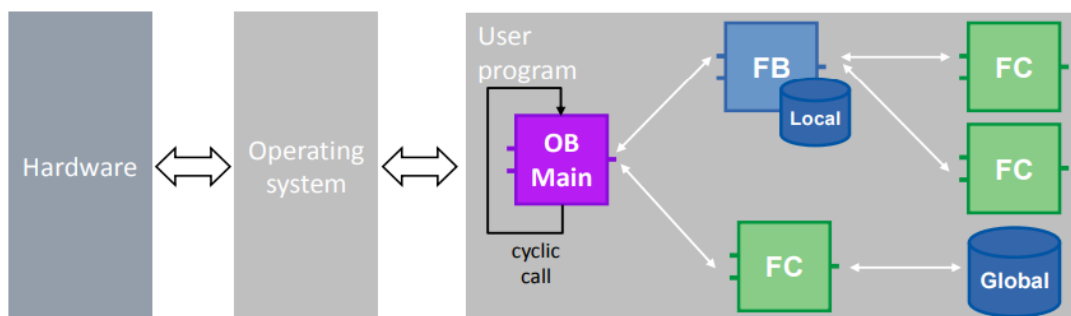
Prvním zástupcem je automatizační SW od značky Siemens. Pouhé vypsání jeho funkcí by rozsahem dalo možná na celou další závěrečnou práci, takže je zde shrnut jen základní princip programování PLC, který je důležitý pro vlastní řešení. PLC programování se řídí podle normy IEC61131-3, ale místy se odlišuje názvoslovím.

Struktura tvorby PLC programů je v případě TIA Portalu zajištěna tzv. bloky (Obr. 17):

- *Organizační blok (OB)*
- *Funkční blok (FB)*
- *Datový blok (DB)* – není součástí normy IEC61131-3
- *Blok funkce (FC)*

TIA Portal podporuje několik programovacích jazyků dle normy IEC61131-3:

- *Ladder Diagram (LD)* – reléové schéma
- *Function Block Diagram (FBD)* – jazyk funkčních bloků
- *Structured Text (ST)* – strukturovaný text



Obr. 17 – Struktura programování v Siemens TIA Portal [34]

OB tvoří rozhraní mezi operačním systémem a samotným uživatelským programem a většinou pak obsahuje hlavní cyklický program (Main) a zároveň tak může volat další bloky. OB si pak volá samotný operační systém PLC.

FB pracují nejlépe ve spojení s DB, kde FB cyklicky generuje data a ta ukládá do DB, kde jsou uložena jako trvalé hodnoty do opětovného přemazání. Nejčastěji se FB používají k tvorbě podprogramů k lepšímu strukturování uživatelského programu – FB je možné volat na několika místech uživatelského programu, a proto je vhodný pro často se opakující příkazové smyčky.

DB může být použit také samostatně jako úložiště dat, např. pro definování ovládacích prvků HMI rozhraní, které je pak možné snadno volat v OB/FB. Při definici dat je možné si navolit datový typ (Bool, Int, Word, Time, Real atd.) a počáteční hodnotu.

FC by se dalo zjednodušeně popsat jako FB bez cyklického ukládání dat.

TIA Portal také spojuje funkci tvorby a simulace HMI rozhraní pro vytvořený uživatelský program. [33], [34]

3.8.2 TECO Mosaic

Obdobně jako u TIA Portal od Siemens, je v této podkapitole popsán jen základní princip programování PLC, který se řídí normou IEC61131-3.

Základním stavebním blokem programu je tzv. *Programová Organizační Jednotka*, zkráceně POU. Každá tato POU může volat další POU a při tom zároveň předávat jeden nebo více parametrů. Dále existují tři základní typy POU:

- *Program (PROG)* – odpovídá **OB** u Siemens
- *Funkční blok (FB)*
- *Funkce (FUN)*

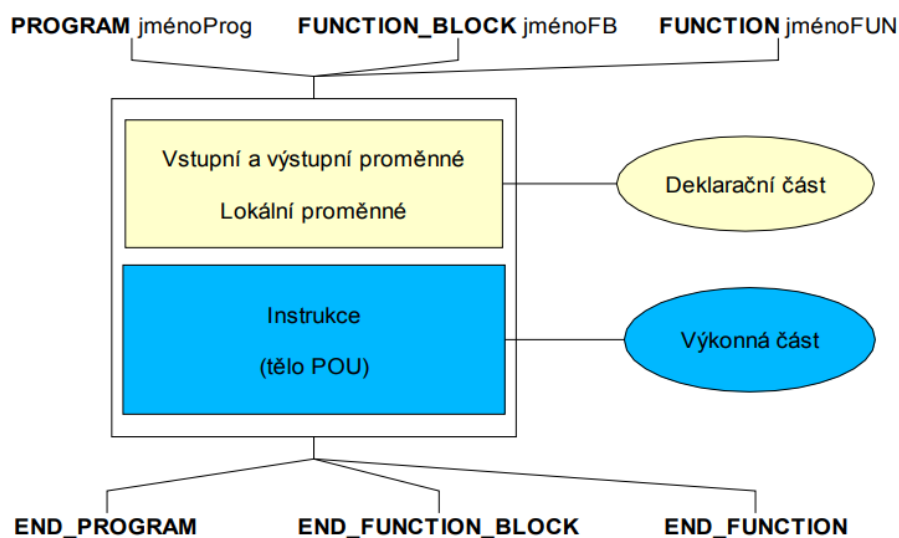
Nejjednodušším typem POU je *funkce*, která se stejnými vstupními parametry vrací pouze jeden stejný výsledek.

Dalším typem je *funkční blok*, který si na rozdíl od *funkce* dokáže zapamatovat některé hodnoty proměnných z předchozího volání. Tím pádem může vracet i více než jeden výsledek.

Vrcholovou programovou jednotkou je *program*. PLC pak může zpracovávat více programů v určitých periodách a s určitými prioritami, vše podle definic spouštění programů v jazyce ST.

Každá POU má dvě části: *deklarační* a *výkonnou*, viz Obr. 18. V části deklarace se definují potřebné proměnné a ve výkonné části se už vykonává vlastní algoritmus.

Dále, obdobně jako TIA Portal, umožňuje tvorbu HMI rozhraní pomocí nástroje *WebMaker*. [35]



Obr. 18 – Základní struktura POU [35]

3.8.3 Beckhoff TwinCAT 3

Obdobně jako u TIA Portal od Siemens, je v této podkapitole popsán jen základní princip programování PLC, který se řídí normou IEC61131-3.

Struktura PLC programování v případě TwinCAT využívá tzv. programovací objekty (*Programming Objects*):

- *Object POU*
- *Object Action*
- *Object Transition*
- *Object Method*
- *Object Property*

Základním programovacím objektem je opět POU (*Programová Organizační Jednotka*), která obsahuje tři základní typy – *Program*, *Funkční blok* a *Funkci* – stejně jako v případě TECO Mosaic (Obr. 18) a v souladu s normou IEC61131-3.

Action lze použít k implementaci dalšího kódu, a to i v jazyce, který se liší od základní implementace – tj. FB nebo PROG, pod kterým je akce vložena.

Transition se používá pro definování podmínky, za které se má aktivovat následující krok.

Method obsahuje sekvenci příkazů, ale na rozdíl od FUN není nezávislou POU, a proto musí být přiřazena k FB nebo PROG.

Property je rozšíření normy IEC61131-3 a je určená k objektově orientovanému programování. Skládá se z přístupových metod *Get* a *Set*. [36]

3.9 Výstupy řešerše

Na závěr jsou shrnuty výstupy řešerše. Cílem bylo najít vhodný SW nástroj pro přípravu virtuálního zprovoznění k účelům výuky PLC řízení. K tomu se váže volba komunikačního protokolu a podpora širšího spektra výrobců PLC.

3.9.1 Vybraný SW nástroj pro virtuální zprovoznění

Na základě výstupů z kap. 3.1 až 3.4, byl pro tvorbu vlastního řešení vybrán SW Visual Components. Ten lze mimo jiné využít v rámci virtuálního zprovoznění jako vizualizaci při testování PLC programu.

3.9.2 Zvolený komunikační protokol

K volbě komunikačního protokolu došlo z výčtu v kap. 3.7. Nejvhodnějším řešením je univerzální protokol, který podporuje velká škála výrobců PLC a zároveň je kompatibilní s vybraným SW Visual Components z předchozí kapitoly. Tím je protokol OPC UA.

3.9.3 Podporování výrobců PLC, korespondující automatizační SW

OPC UA je čím dál více podporovaný standard a spektrum výrobců PLC, kteří jej podporují je široké a bude se do budoucna patrně rozrůstat. Ve vlastním řešení bude VC využito jako virtuální model připojený k virtuálnímu PLC za pomoci OPC UA. V principu lze VC využít i jako virtuální model připojený k fyzickému PLC přes OPC UA. Z uvedených příkladů v kap. 3.8 je možné použít virtuální PLC od Siemens, který podporuje OPC UA skrze PLCSIM Advanced. Beckhoff je softwarové PLC, které běží na standardním HW a lze jej pro komunikaci po OPC UA také použít. TECO bohužel v době vzniku této práce stále nepodporuje nativní komunikaci přes OPC UA, tudíž pro vlastní řešení nebude použito.

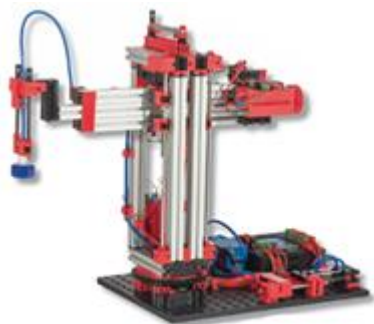
4 Vlastní řešení

V této kapitole bude podrobněji popsáno vlastní řešení, sestávající se primárně z tvorby několika virtuálních modelů využitelných pro výuku PLC řízení. Bude vytvořen jeden jednodušší model dopravníku, který bude sloužit k úvodnímu seznámení studentů s PLC programováním a virtuálním zprovozněním. Dále budou vytvořeny čtyři programovatelné virtuální modely, odpovídající čtyřem fyzickým stavebnicím od firmy Fischertechnik, které budou studenti virtuálně zprovožňovat v rámci semestrálního projektu.

Nejdříve bude čtenář seznámen s předlohou virtuálních modelů. Poté bude pokračovat popis propojení prostředí virtuálních modelů s PLC. Následovat bude popis tvorby virtuálních modelů. Nakonec budou provedena shrnutí a závěry, společně s uvedením seznamů a příloh.

4.1 Výuka PLC řízení

V rámci výuky PLC řízení studenti vypracovávají semestrální úlohu, kde má každý student za úkol zprovoznit (naprogramovat) jednu ze čtyř přidělených fyzických programovatelných stavebnic od firmy Fischertechnik (FT), viz Obr. 19 až Obr. 22. Vedle samotných stavebnic jsou k dispozici i čtyři fyzické PLC sestavy obsahující PLC, potřebné moduly pro I/O (vstupy a výstupy), HMI obrazovku, tlačítko CentralStop, napájení, ethernetové připojení k PC a další nezbytné součásti. Celá PLC sestava se připojí ke stavebnici pomocí jednoho datového kabelu, který obsahuje i napájení. Jelikož se jedná pouze o základní kurz PLC řízení, není součástí výuky (ani stavebnic) bezpečnost, programování safety atp. Je to kapacitně nad možnosti kurzu a nezabývá se tím ani tato práce.



Obr. 19 – Stavebnice FT č.1: Robotická manipulace dílců (400x350 mm)



Obr. 20 – Stavebnice FT č.2: Regálový zakladač dílů (700x400 mm)



Obr. 21 – Stavebnice FT č.3: Multiprocesní stanice s tepelným zpracováním (500x350 mm)



Obr. 22 – Stavebnice FT č.4: Třídící linka s rozpoznáním barev (500x350 mm)

Pro tvorbu a ladění PLC programu je velmi výhodné mít možnost si PLC program poměrně často testovat a snáze tak postupovat kupředu v řešení úlohy. Nicméně, stavebnice jsou pouze

čtyři a studentů bývá i několik desítek, tudíž by v jeden moment každý student neměl možnost být fyzicky připojen ke stavebnici a odlaďovat PLC program. I z tohoto důvodu se již několik let využívá virtuální zprovoznění pomocí Siemens MCD, jak bylo zmíněno v kap. 3.2.

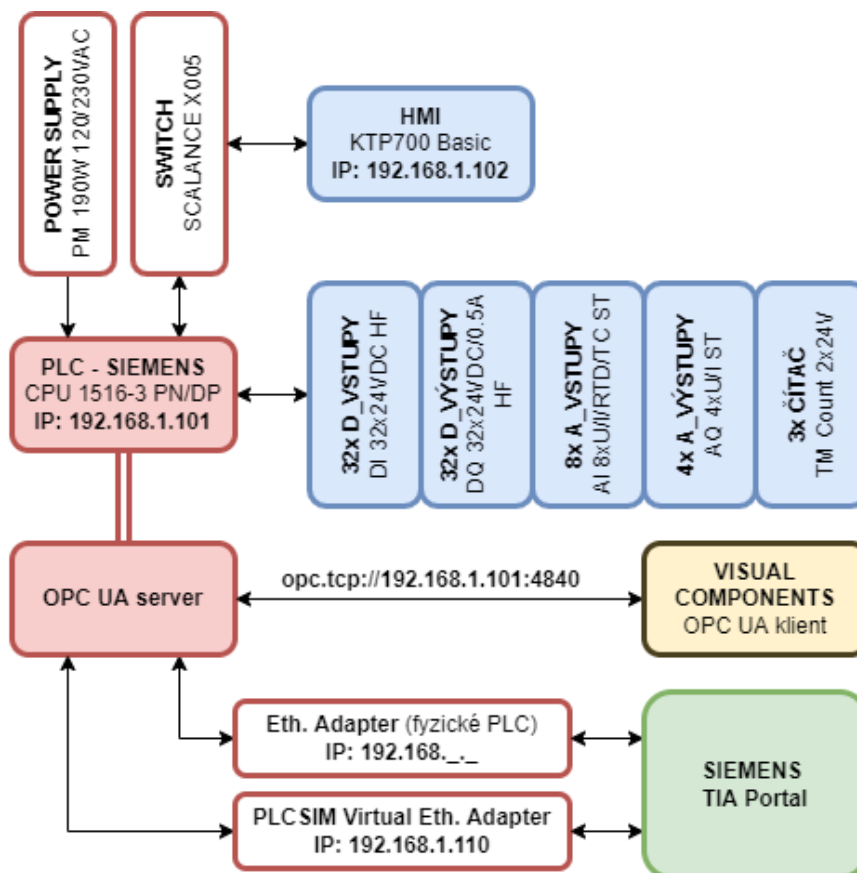
V rámci této práce bude ke každému ze čtyř modelů vytvořen virtuální model, který splňuje nutné vlastnosti pro zajištění plné funkčnosti, z hlediska PLC řízení, v porovnání s fyzickou stavebnicí. Bude kopírovat fyzické rozměry stavebnic, kinematickou strukturu, I/O a zjednodušenou geometrii. Každý ze studentů tak ve finále bude mít k dispozici svůj vlastní virtuální model stavebnice (pracoviště) ve VC pro odlaďování PLC programu. Po odlaďení PLC programu na virtuálním modelu každý student PLC program vyzkouší i na fyzické stavebnici a případně provede pouze drobné úpravy, aby mu PLC program spolehlivě fungoval i s fyzickou stavebnicí – např. hodnoty časovačů, prodlevy nebo koeficienty pro odečty hodnot z enkodérů.

4.2 Propojení vizualizačního softwaru s PLC

V této kapitole je blíže popsáno nezbytné propojení vizualizačního programu pro virtuální zprovoznění s virtuálním PLC. Pro všechna propojení je použit univerzální komunikační protokol OPC UA.

4.2.1 Propojení Visual Components se Siemens – OPC UA

Pro plnohodnotné propojení VC s produkty od Siemens je potřeba provést několik nastavení jak v prostředí pro tvorbu PLC programu (TIA Portal), tak v samotném virtuálním PLCSIM Advanced, a taktéž ve Windows a samotném Visual Components.



Obr. 23 – Blokové schéma propojení VC s PLC Siemens

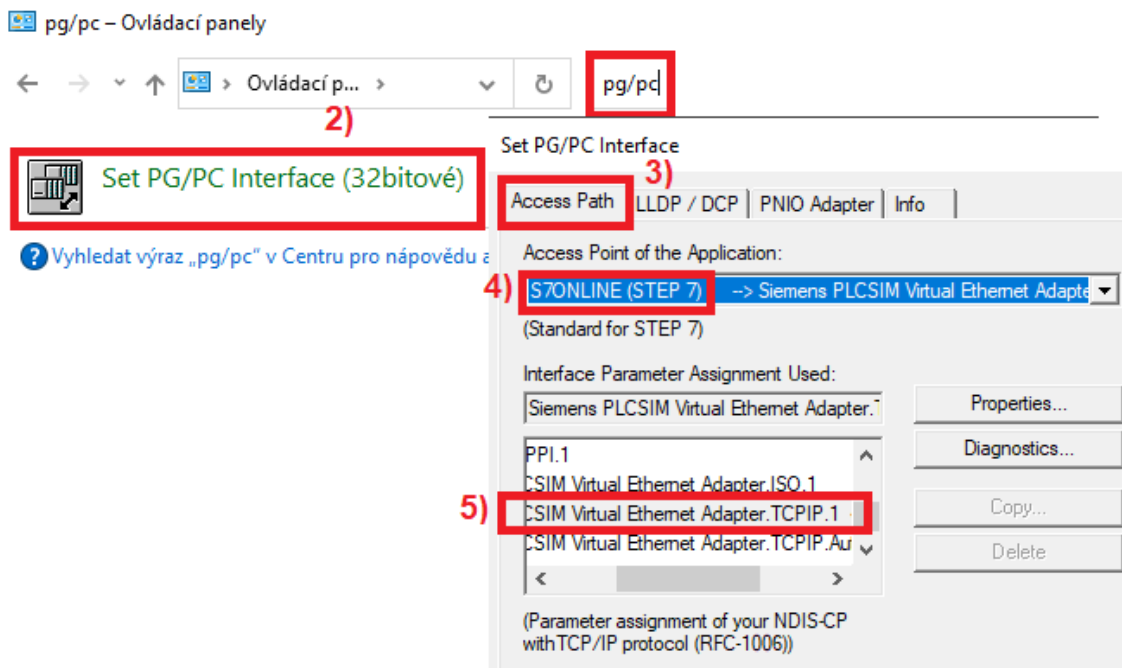
Na Obr. 23 je schéma propojení VC s PLC pomocí OPC UA společně s konfigurací PLC, které odpovídá i fyzickým PLC sestavám. Níže je výčet potřebného SW i s použitou verzí:

- Visual Components Premium 4.4 (nebo vyšší)
- TIA Portal V15.1
- PLCSIM Advanced V2.0 SP1

Windows

Pro možnost komunikace OPC serveru PLCSIM Advanced s Virtual Ethernet Adapter od Siemens, je třeba nejprve změnit přístupový bod aplikace rozhraní PG/PC v ovládacích panelech, viz Obr. 24:

- 1) Otevřít *Ovládací panely*
- 2) Vyhledat a otevřít „PG/PC“
- 3) Vybrat záložku *Access Path*
- 4) Zkontrolovat, že se nastavuje *S7ONLINE (STEP 7)*
- 5) Přepnout na „*Siemens PLCSIM Virtual Ethernet Adapter.TCPIP.1*“

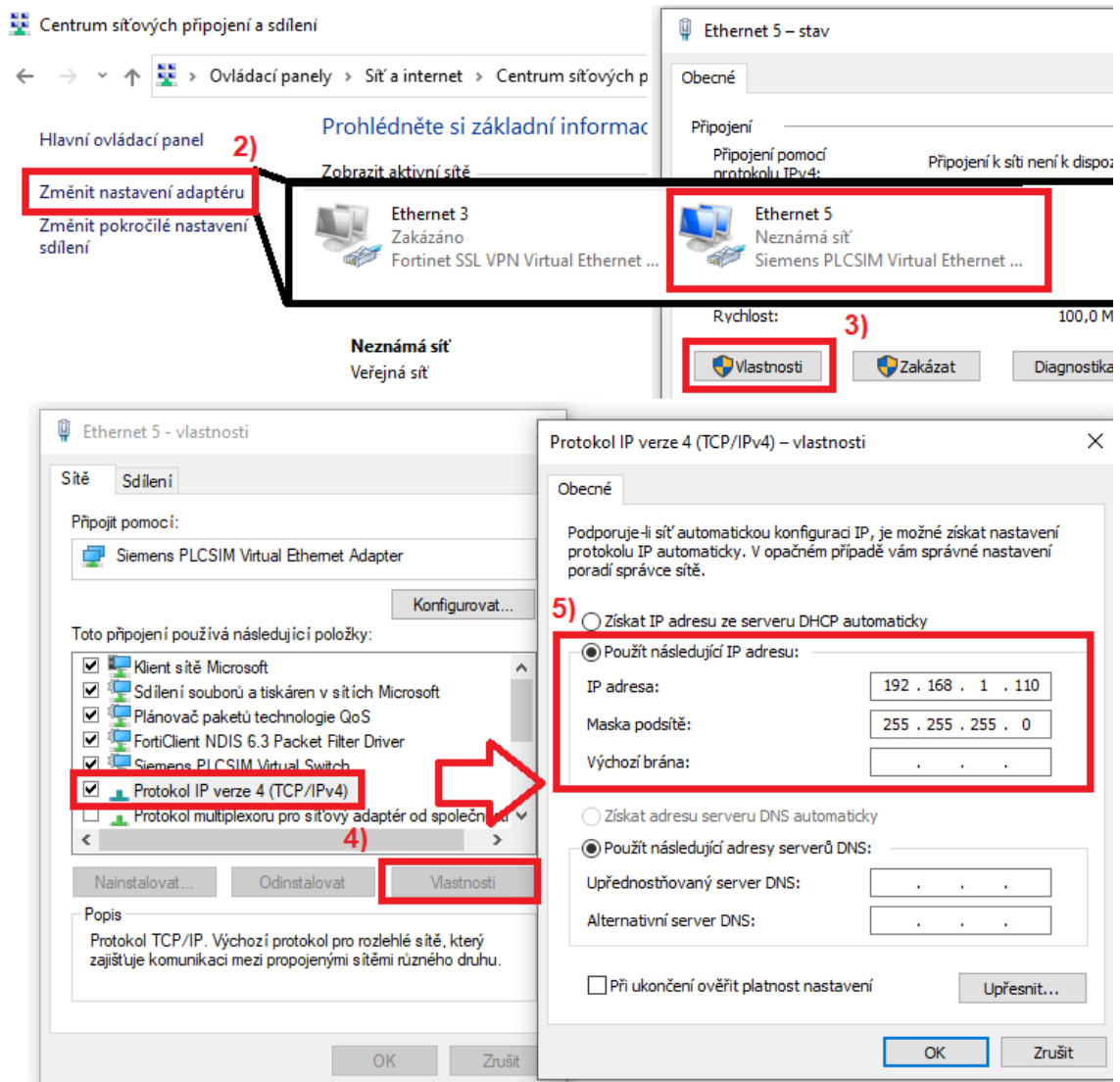


Obr. 24 – Konfigurace rozhraní PG/PC

Další konfigurací v prostředí Windows je nastavení pevné IP adresy právě tohoto virtuálního adaptéru z předchozího kroku. Postup, viz Obr. 25, je následovný:

- 1) Otevřít *Centrum síťových připojení a sdílení*
- 2) Pokud není zobrazena neznámá síť virtuálního adaptéru, je nutno adaptér nejprve přes pravé tl. *Povolit* skrze *Změnit nastavení adaptéru*
- 3) Poté na příslušný ethernetový adaptér (v tomto případě *Ethernet 5*) kliknout a zvolit *Vlastnosti*
- 4) Vybrat *Protokol IP verze 4 (TCP/IPv4)* a opět zvolit *Vlastnosti*

- 5) Zaškrtnout *Použít následující IP adresu* a vyplnit příslušnou pevnou IP adresu (např. „192.168.1.110“) a masku podsítě („255.255.255.0“) – **!!POZOR!! IP adresa MUSÍ být jiná než adresa virtuálního PLC (viz Obr. 29)**

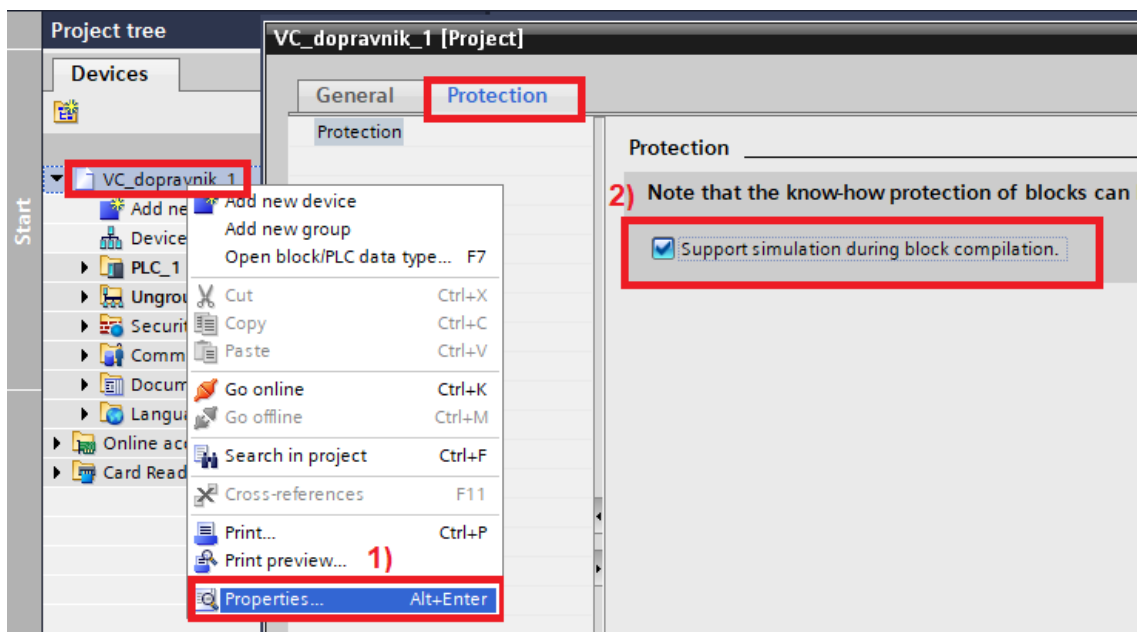


Obr. 25 – Nastavení pevné IP adresy virtuálního adaptéru

TIA Portal

Po spuštění a přípravě PLC programu v prostředí TIA Portal je před odesláním na virtuální PLC třeba udělat několik nastavení. Prvním z nich je v nastavení vlastností projektu, viz Obr. 26:

- 1) Pravým tl. kliknout na příslušný projekt a zvolit *Properties*
- 2) V záložce *Protection* povolit „*Support simulation during block compilation*“



Obr. 26 – Nastavení projektu v TIA Portal

Jako další krok je nastavení zvoleného PLC a OPC serveru, v tomto případě *CPU 1516-3 PN/DP*, viz Obr. 27:

- 1) Pravým tl. kliknout na příslušné PLC a zvolit *Properties*
- 2) V záložce *General* po najetí na *Profinet interface XI* lze zvolit IP adresu virtuálního PLC, jehož instance je posléze spuštěna v *PLCSIM Advanced*
- 3) Dále je třeba zvolit *Protection & Security* a zaškrtnout *Full access (no protection)* společně s *Permit access with PUT/GET communication from remote partner*, aby mohla simulace ve VC kromě čtení dat z PLC data také zapisovat zpět na PLC
- 4) Dále pod *OPC UA* → *Server* povolit *Activate OPC UA server* a zároveň je možné vidět i adresu OPC serveru, která bude třeba uvést ve VC v jednom z dalších kroků
- 5) Nakonec je třeba pod *Runtime licences* → *OPC UA* přiřadit potřebnou licenci; v tomto případě ze seznamu vybrat licenci „*medium*“

PROFINET interface [X1]

Ethernet addresses

Interface networked with

Subnet: Not networked

Add new subnet

2)

IP protocol

Set IP address in the project

IP address: 192 . 168 . 1 . 101

Subnet mask: 255 . 255 . 255 . 0

Use router

Router address: 0 . 0 . 0 . 0

Protection & Security

Access level

Select the access level for the PLC.

Access level	HMI	Read	Write
3) <input checked="" type="radio"/> Full access (no protection)	✓	✓	✓
<input type="radio"/> Read access	✓	✓	
<input type="radio"/> HMI access	✓		
<input type="radio"/> No access (complete protection)			

Connection mechanisms

3) Permit access with PUT/GET communication from remote partner

Server

General

Accessibility of the server

4) Activate OPC UA server

Server addresses

Address
opc.tcp://192.168.1.101:4840
opc.tcp://192.168.0.101:4840

OPC UA

Runtime licenses

Type of required license: SIMATIC OPC UA S7-1500 medium

Type of purchased license: SIMATIC OPC UA S7-1500 medium

No license

SIMATIC OPC UA S7-1500 small

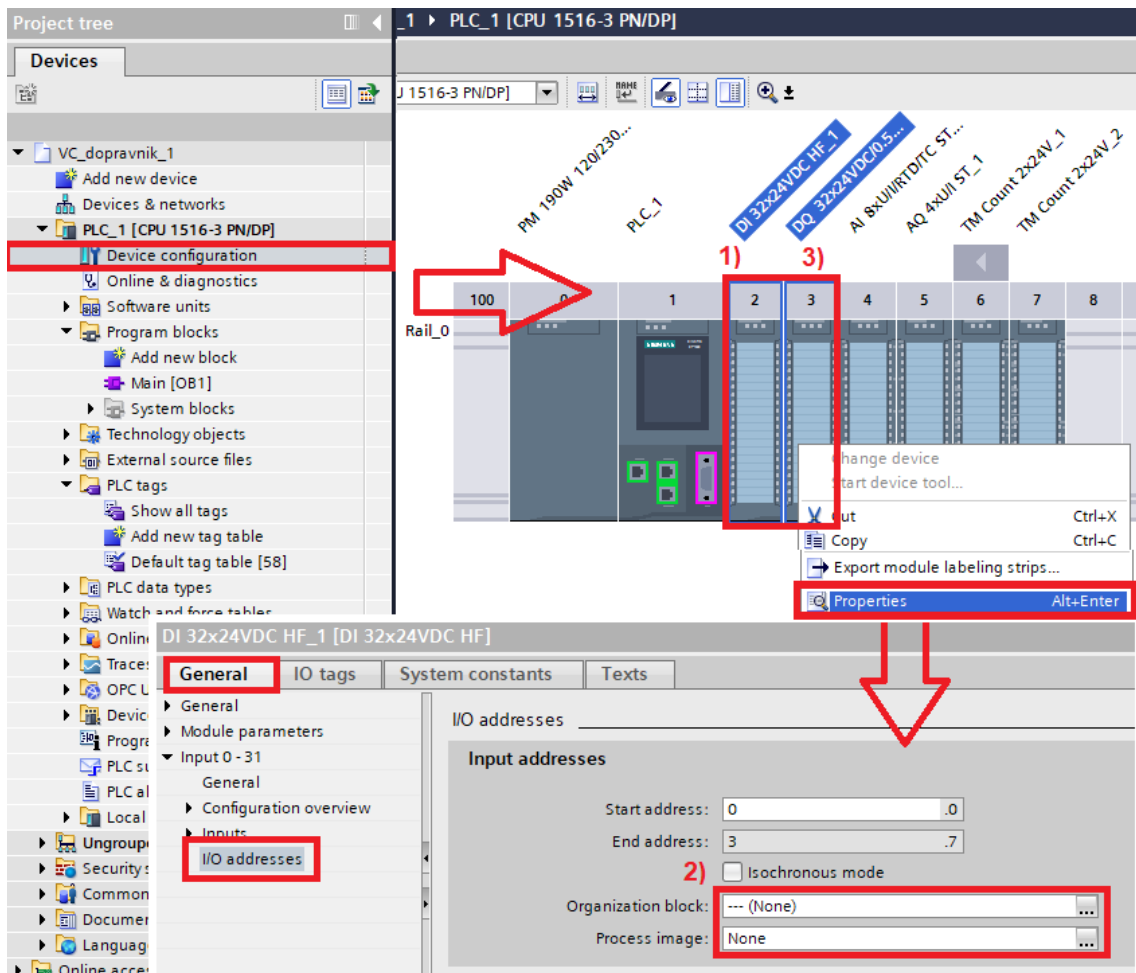
5) **SIMATIC OPC UA S7-1500 medium**

SIMATIC OPC UA S7-1500 large

Obr. 27 – Nastavení PLC v TIA Portal

Posledním krokem v nastavení TIA Portal je správné nastavení modulů vstupů a výstupů. Konkrétně se jedná o *Process image*, který musí být neaktivní, aby fyzické I/O nepřepisovaly hodnoty zapsané simulací ve VC. To se provede následovně, viz Obr. 28:

- 1) Vybrat *Device configuration* a kliknout pravým tl. nejdříve na modul vstupů (*DI 32x24VDC HF_1*)
- 2) V záložce *General* zvolit *I/O addresses* pod *Input 0-31* a nastavit *Organization block* a *Process image* na „None“
- 3) Stejný postup se opakuje i pro modul výstupů (*DQ 32x24VDC/0.5A HF_1*), případně pro další moduly

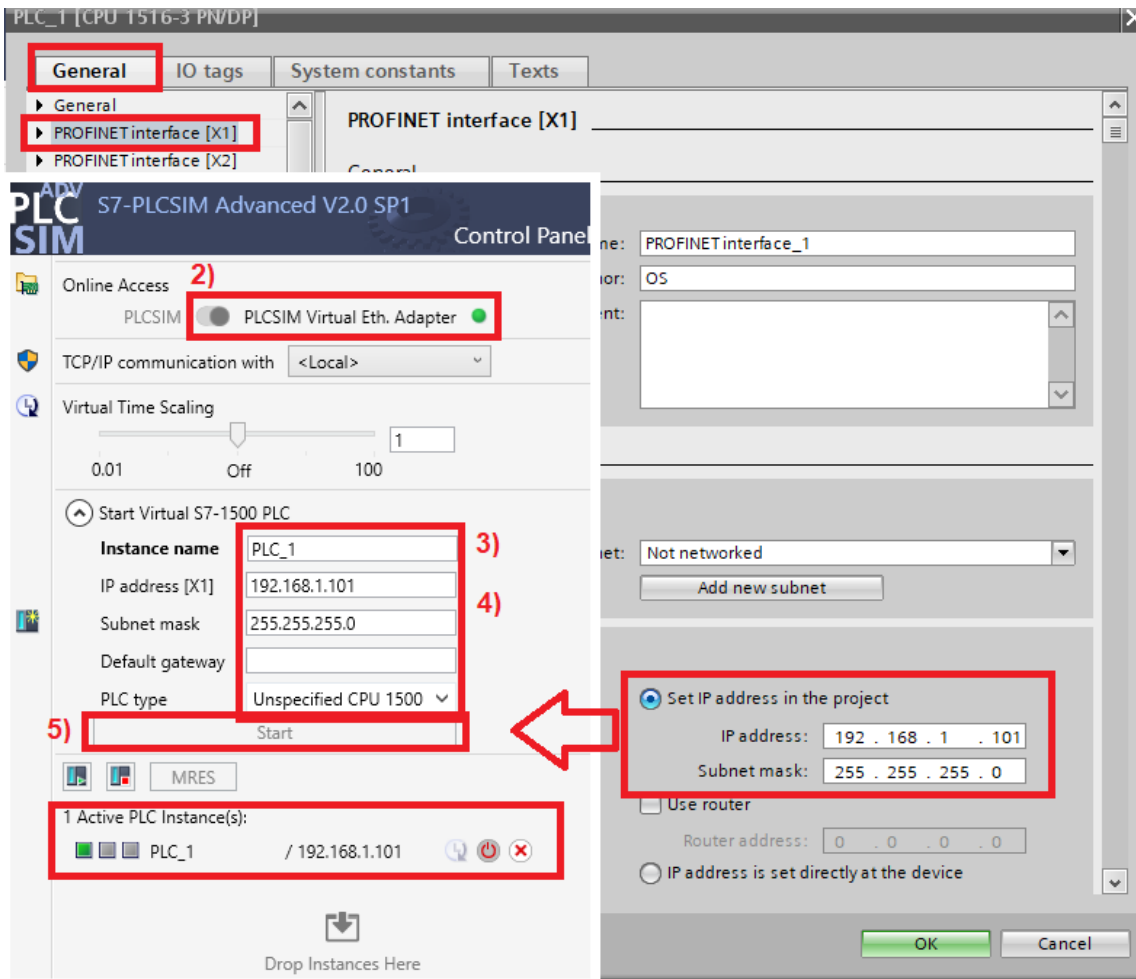


Obr. 28 – Nastavení I/O v TIA Portal

PLCSIM Advanced

V samotném SW pro virtuální PLC od Siemens je třeba provést následující kroky (Obr. 29):

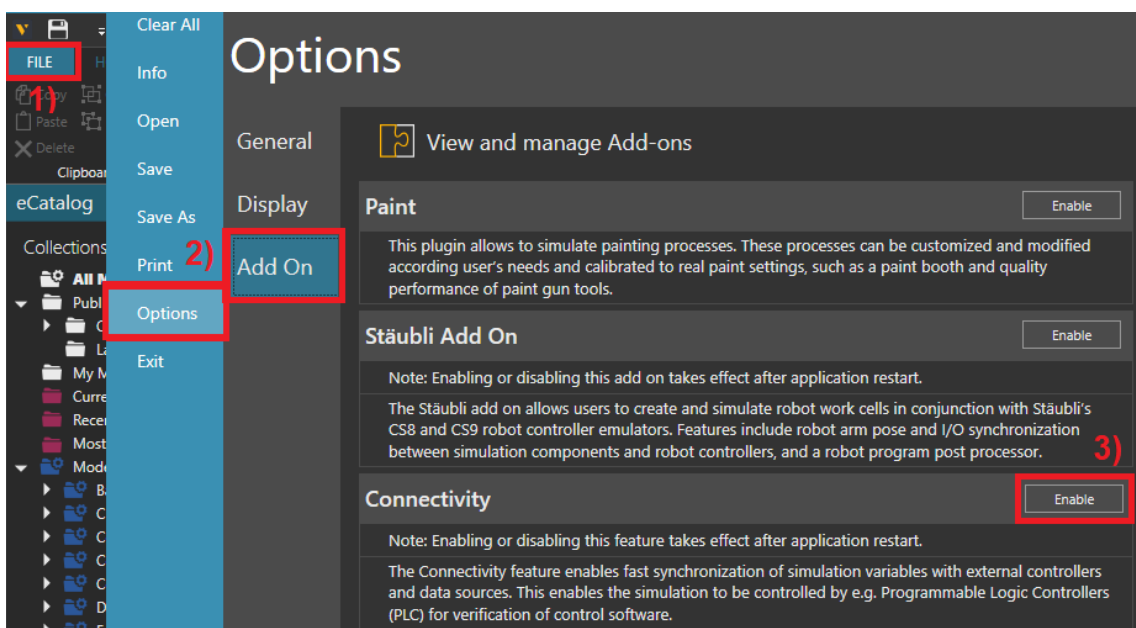
- 1) Spustit PLCSIM Advanced V2.0 SP1
- 2) Přepnout *Online Access* na *PLCSIM Virtual Adapter*
- 3) Poté zvolit *jméno Instance* PLC dle pojmenování PLC v TIA („PLC_1“)
- 4) Následně napsat *IP adresu Profinet interface X1*, který bude použit („192.168.1.101“) a také defaultní masku podsítě („255.255.255.0“)
- 5) Spustit virtuální PLC tlačítkem *Start*



Obr. 29 – Spuštění virtuálního PLC v PLCSIM Advanced

Visual Components

Jako krok č.1 je třeba ve VC povolit záložku *Connectivity* v nastavení, viz Obr. 30:

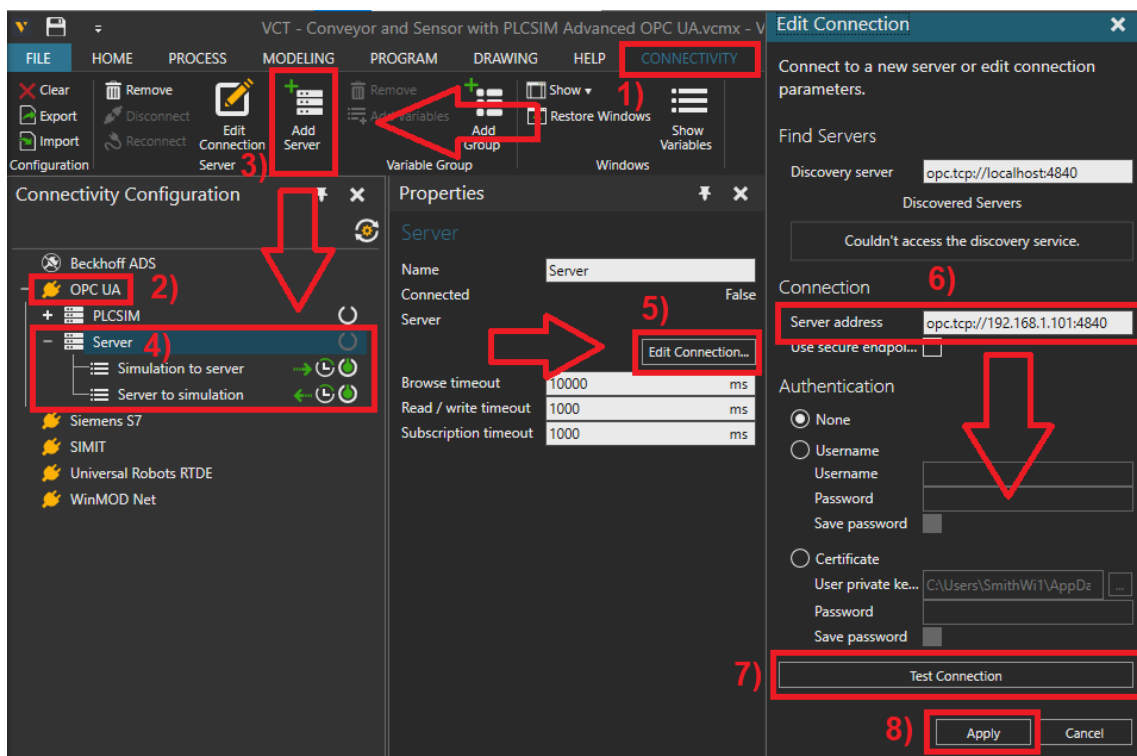


Obr. 30 – Povolení záložky Connectivity

- 1) Přepnout do záložky *File*
- 2) Zvolit *Options* a následně *Add On*
- 3) Povolit záložku *Connectivity* skrze „*Enable*“

Krokem č.2 je přidání příslušného serveru již v samotné záložce *Connectivity* a nastavením připojení serveru, v tomto případě jde o nastavení OPC UA, viz Obr. 31:

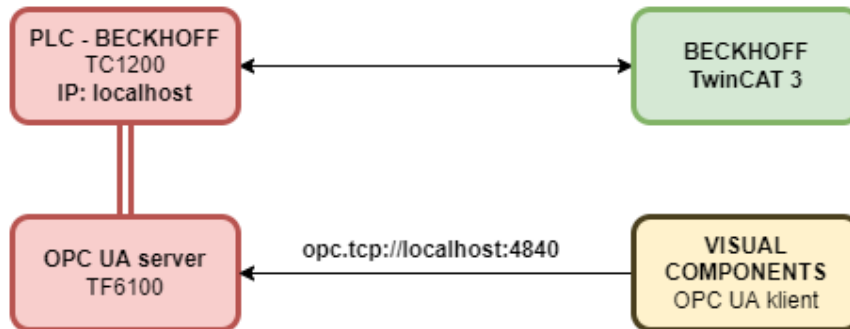
- 1) Přepnout do záložky *Connectivity*
- 2) Vybrat *OPC UA*
- 3) Kliknout na *Add Server*
- 4) Vytvořený server vybrat kliknutím na *Server* (název lze změnit)
- 5) Pokračovat stiskem *Edit Connection*
- 6) Vyplnit IP do *Server address* spuštěné instance PLC včetně portu – podle nastavení v TIA Portal, viz Obr. 27
- 7) Otestovat připojení stiskem *Test Connection*
- 8) Pokud je *Success*, tak vše uložit stiskem *Apply*



Obr. 31 – Vytvoření spojení VC a OPC UA serveru (Siemens)

4.2.2 Propojení Visual Components s Beckhoff – OPC UA

Pro plnohodnotné propojení VC s produkty od Beckhoff je potřeba provést několik nastavení jak v prostředí pro tvorbu PLC programu (TwinCAT), tak v dalších podpůrných SW od Beckhoff a v samotném Visual Components.



Obr. 32 – Blokové schéma propojení VC s PLC Beckhoff

Na Obr. 32 je schéma propojení VC s PLC pomocí OPC UA. Níže je výčet potřebného SW i s použitou verzí:

- Visual Components Premium 4.4 (nebo vyšší)
- TwinCAT 3.1 eXtended Automation Engineering (XAE) 4024.10 (nebo vyšší)
- TF6100 TwinCAT 3 OPC UA Server 4.4.53.0
- TF6100 TwinCAT 3 OPC UA Configurator 4.4.73.0

TwinCAT 3

Jako první po otevření nebo založení PLC projektu v prostředí TwinCAT je třeba konfigurovat PLC proměnné pro přístup OPC UA. To je provedeno přidáním speciálního *attribute* **před každou zvolenou proměnnou**, viz Obr. 33:

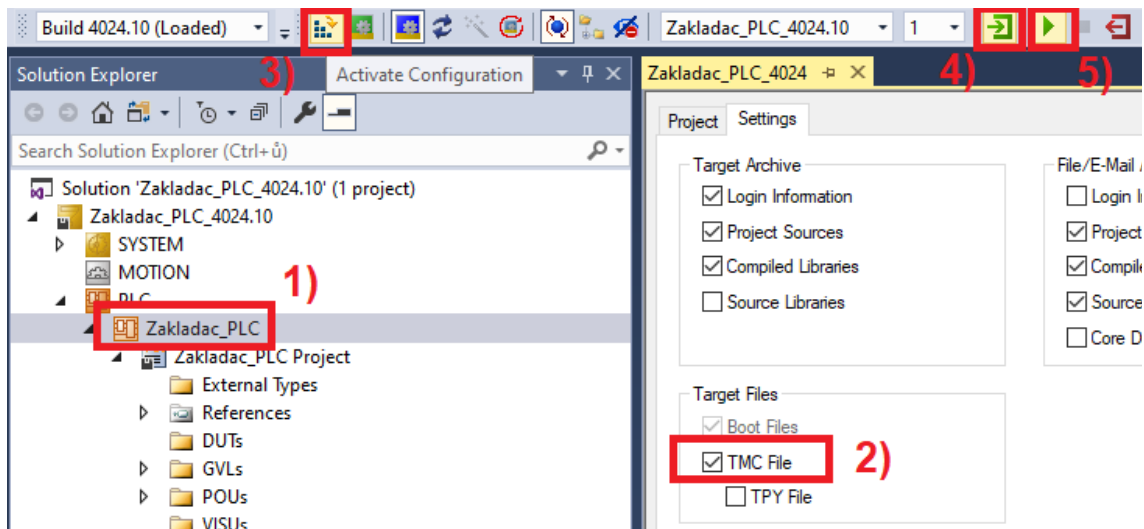
```
{attribute 'OPC.UA.DA' := '1'}
bVariable : BOOL;
```

Obr. 33 – Definování PLC proměnných pro OPC UA

Dalším krokem je aktivace stahování TMC souboru symbolů a provedení následujících kroků, viz Obr. 34:

- 1) Zvolit aktuální PLC projekt
- 2) Aktivovat *TMC File* v *Settings*
- 3) Stisknout *Activate Configuration* a potvrdit první i druhé dialogové okno stiskem *Ok*
- 4) Po nahrání do PLC stisknout *Login* a potvrdit vytvoření aplikace *Port_851* stiskem *Yes*
- 5) Poté stisknout *Start* pro spuštění PLC

Tímto se nahraje konfigurace na PLC a spustí se server OPC UA.

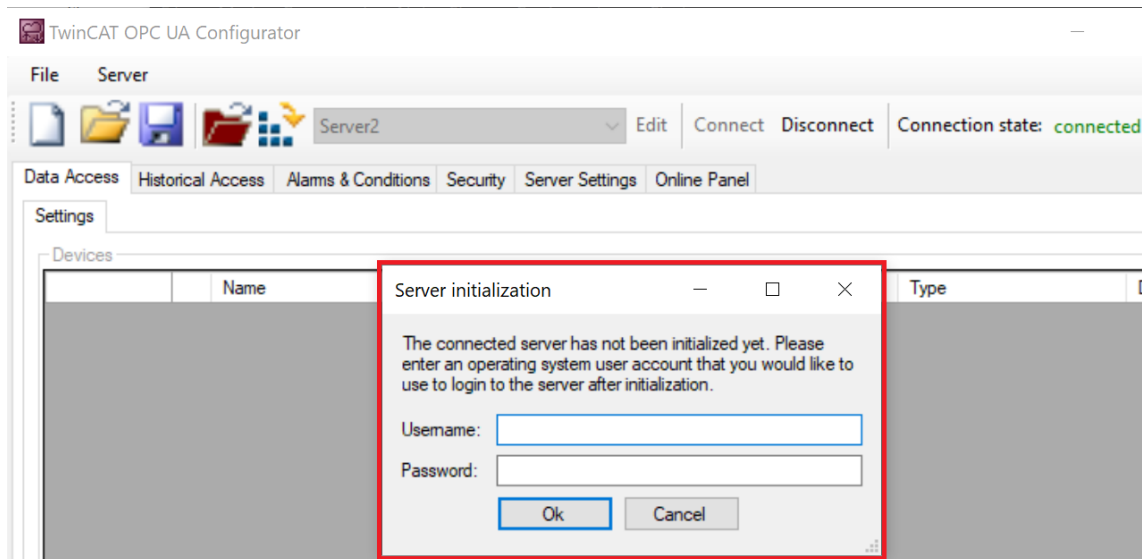


Obr. 34 – Konfigurace PLC Beckhoff pro OPC UA

TF6100 OPC UA Configurator

Od verze 4.4.0 TwinCAT OPC UA serveru je třeba provést prvotní inicializaci serveru, která je založena na tzv. *Trust On First Use*. Až potom budou k dispozici všechny funkce OPC UA serveru. Inicializace se provede následujícími kroky, viz Obr. 35:

- 1) Spustit TF6100 OPC UA Configurator
- 2) Po připojení se k serveru konfigurační nástroj automaticky rozpozná, pokud server není inicializovaný a vyskočí dialogové okno
- 3) Do okna je potřeba vyplnit **uživatelské jméno a heslo**, pod kterým bude posléze server potřeba ověřovat při pokusech o připojení – jak zde v konfigurační nástroji, tak poté i **ve Visual Components**
- 4) Po vyplnění a stisku *Ok* je server inicializovaný, automaticky se restartuje a budou k dispozici všechny funkce

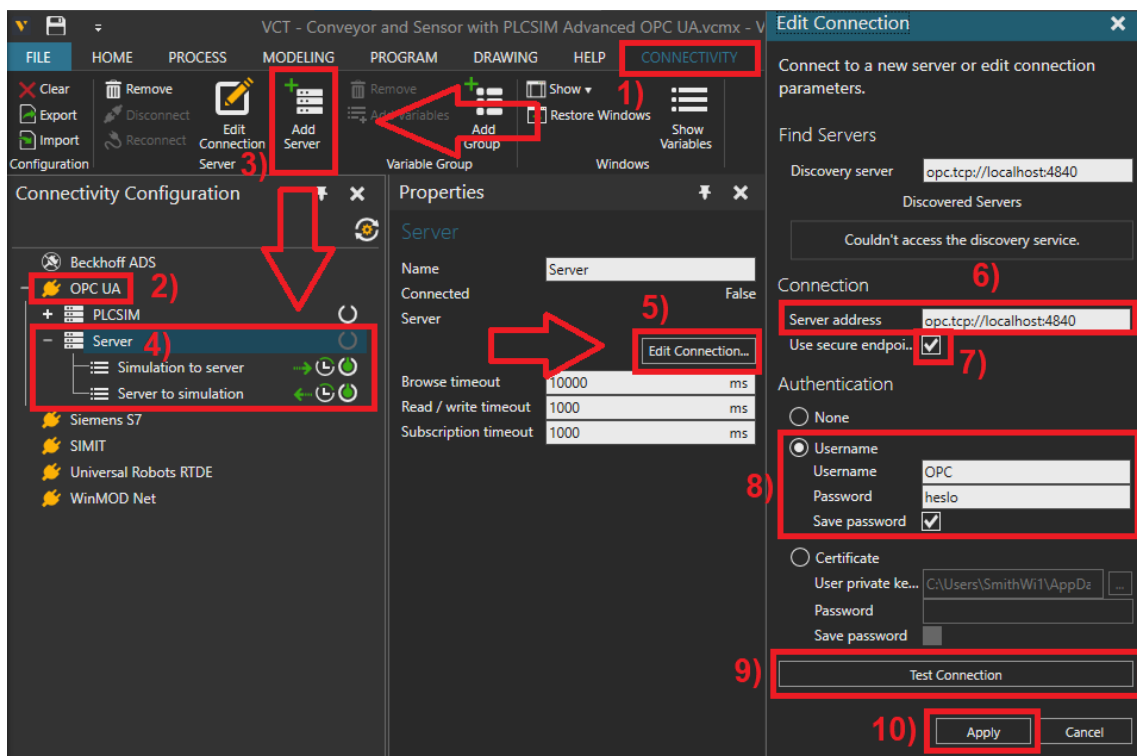


Obr. 35 – Inicializace OPC UA serveru

Visual Components

Pokud po spuštění VC není k dispozici záložka Connectivity, je potřeba ji nejdříve povolit (Obr. 30), viz kap 4.2.1. Poté následuje přidání serveru obdobně jako tomu bylo v kap. 4.2.1 v případě Siemensu. Je potřeba provést následující kroky, viz Obr. 36:

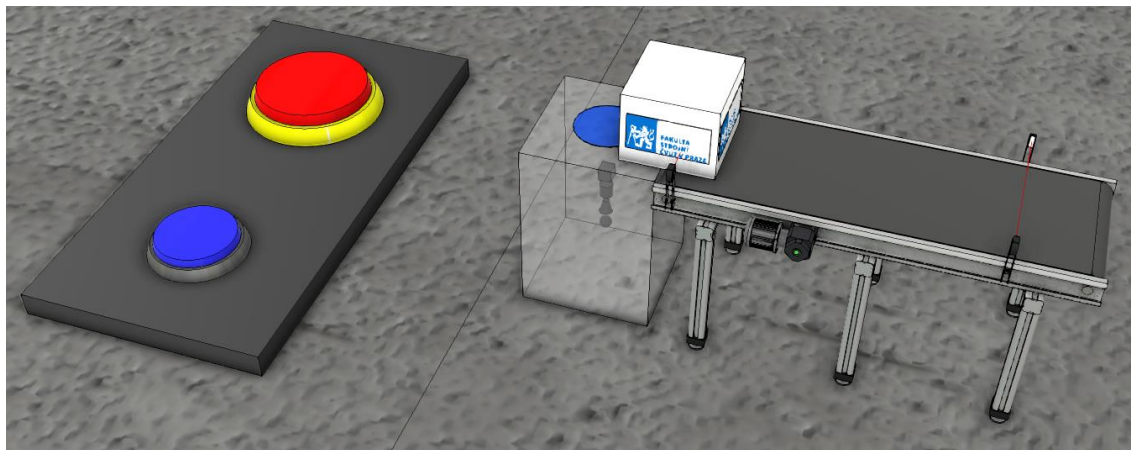
- 1) Přepnout do záložky *Connectivity*
- 2) Vybrat *OPC UA*
- 3) Kliknout na *Add Server*
- 4) Vytvořený server vybrat kliknutím na *Server* (název lze změnit)
- 5) Pokračovat stiskem *Edit Connection*
- 6) Vyplnit IP do *Server address* spuštěné instance PLC včetně portu – v případě Beckhoff URL OPC serveru obsahuje přímo název IPC, na kterém lokální instance běží, tudíž bylo vyplněno univerzální URL: ***opc.tcp://localhost:4840***
- 7) Dále je potřeba zaškrtnout *Use secure endpoint*
- 8) V části *Authentication* zvolit *Username* a vyplnit údaje použité při inicializaci serveru (Obr. 35)
- 9) Otestovat připojení stiskem *Test Connection*
- 10) Pokud je *Success*, tak vše uložit stiskem ***Apply***



Obr. 36 – Vytvoření spojení VC a OPC UA serveru (Beckhoff)

4.3 Tvorba modelu Dopravníku

Vedle tvorby hlavních virtuálních modelů pro studentský semestrální projekt byl také vytvořen jednoduchý pomocný model pro účely výuky, viz Obr. 37. Lze na něm názorně demonstrovat základní principy a funkce PLC programování, jako např. RS klopný obvod, časovač, vysílaný signál na pohon dopravníku nebo přijímaný signál z optických závor.



Obr. 37 – Pomocný virtuální programovatelný model pro účely názorné výuky

4.3.1 Tvorba geometrie

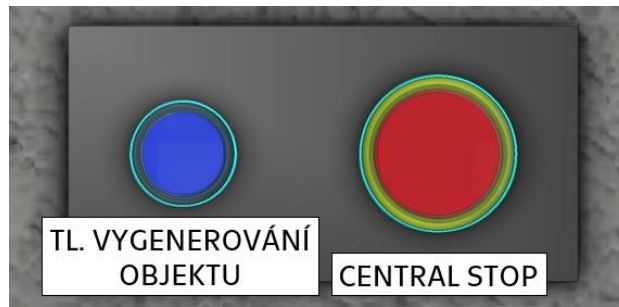
V tomto případě bylo použito několik prvků přímo z dostupné knihovny *eCatalog* ve VC. Kromě samotné geometrie obsahují také již některé připravené funkce jako signály, rozpohybované osy, cesty pro vedení objektů atd. Jsou jimi:

- *ShapeFeeder* – slouží pro generování zvolených předmětů
- *Visual Components Box* – objekt zvolený pro generování
- *Conveyor* – jeden ze základních pásových dopravníků
- *Motor* – pohon dopravníku
- *Conveyor Sensor* – optická brána

4.3.2 Návrh ovládacích prvků virtuálního modelu

Vedle externího řízení pomocí PLC je při běhu simulace možné virtuální model ovládat přímo také dvěma aktivními tlačítky, viz Obr. 38.

- **CentralStop** má signál vyvedený přímo do I/O připojeného PLC. Tlačítko je nutné po stisku uvést do původní polohy opětovným stiskem – obdobně jako tomu je ve skutečnosti (tlačítko s aretací)
- **Tlačítko pro generování objektu** je signálem spojeno se *ShapeFeederem* a po jeho stisku dojde k vygenerování příslušného objektu. Při opětovném stisku dojde k vygenerování dalšího kusu.



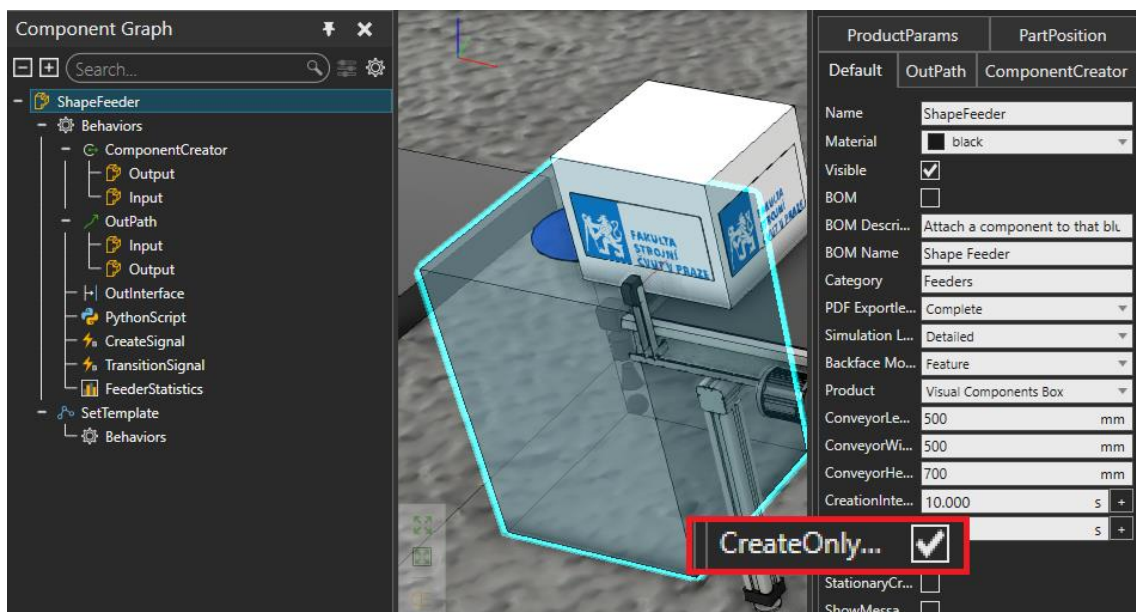
Obr. 38 – Dvě ovládací tlačítka pracoviště

4.3.3 Oživení geometrie

Jak již bylo uvedeno v předchozí kapitole, prvky z knihovny již obsahují některé funkce pro virtuální zprovoznění, takže byly provedeny jen drobné úpravy:

ShapeFeeder

Generování objektů bylo nastaveno na manuální po přivedení externího True signálu, viz Obr. 39.



Obr. 39 – Nastavení podmínky pro generování objektu po přivedení True signálu

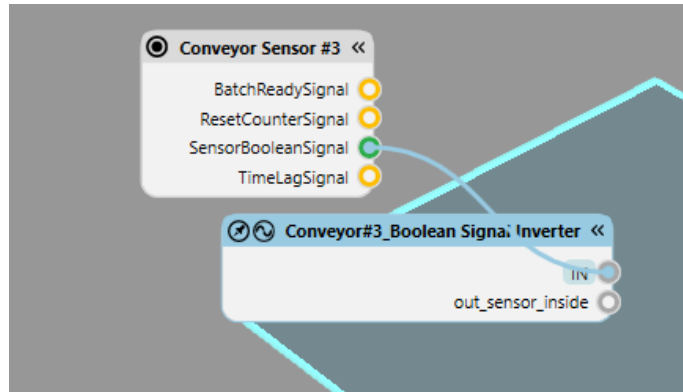
HMI Button

V případě **CentralStopu** bylo třeba převrátit hodnotu signálu při stisku tlačítka. Ve výchozím provedení tlačítko v nestisknutém stavu vysílá signál False a po stisknutí vysílá signál True – až do dalšího stisku (funguje jako tlačítko s aretací). Nicméně v běžném provozu plní CentralStop funkci opačnou – při jeho stisku dojde k přerušení obvodu a vyslání signálu False. V příslušném Python skriptu bylo upraveno několik řádků kódu z celkových cca 60 řádků kódu.

V případě **tlačítka pro generování objektu** bylo naopak potřeba změnit charakteristiku přepínače na tlačítko, které se po stisku vrátí opět do výchozího stavu – tj. signál True vysílá pouze při stisku tlačítkem myši. V příslušném Python skriptu bylo upraveno několik řádků kódu z celkových cca 60 řádků kódu.

Optické závory

Optické závory byly použity jako *Raycast Sensor* s vyvedeným Booleovským signálem. Nicméně bylo zapotřebí otočit hodnoty signálů, jelikož ve výchozím stavu VC vysílají hodnotu False a při průchodu obrobku se změni hodnota na True, což je přesně naopak s chováním optických závor u stavebnice FT. Z důvodu složitosti zásahu do komplexního Python skriptu pro ovládání závor byla z knihovny přidána komponenta *Boolean Signal Inverter*, která slouží jako převodník, jehož výstupem je již převrácený signál, viz Obr. 40.



Obr. 40 – Konfigurace Boolean Signal Inverter

4.3.4 Tvorba I/O pro propojení a řízení PLC

V Tab. 1 a Tab. 2 následuje přehled zvolených signálů pro vstup a výstup (I/O) pro PLC řízení modelu Dopravníku:

Tab. 1 – Zvolené signály pro PLC vstupy (simulace → PLC)

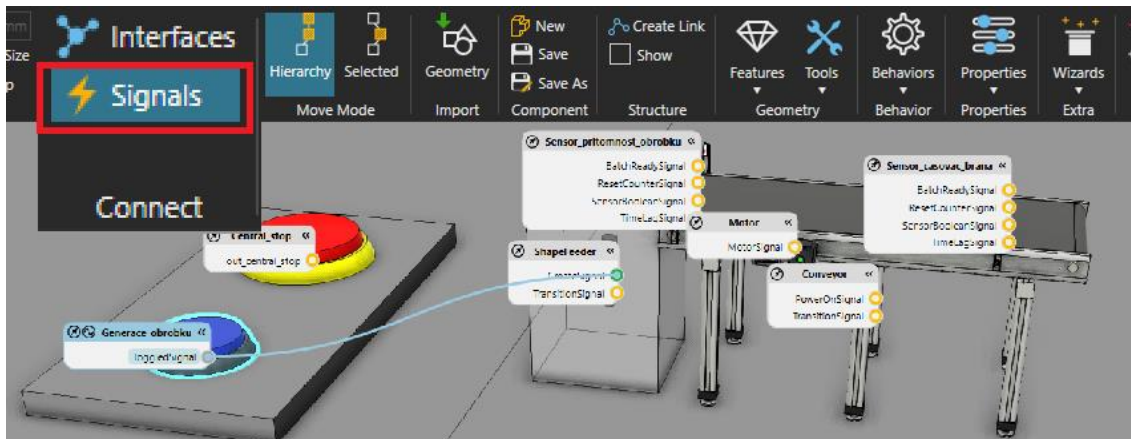
Datový typ	Název signálu VC	Název PLC proměnné
Bool	out_TransitionSignal	di_objekt_na_dopravniku
Bool	out_sensor_inside	di_sensor_inside
Bool	out_sensor_outside	di_sensor_outside
Bool	out_central_stop	di_central_stop

Tab. 2 – Zvolené signály pro PLC výstupy (PLC → simulace)

Datový typ	Název signálu VC	Název PLC proměnné
Bool	in_MotorSignal	do_motor

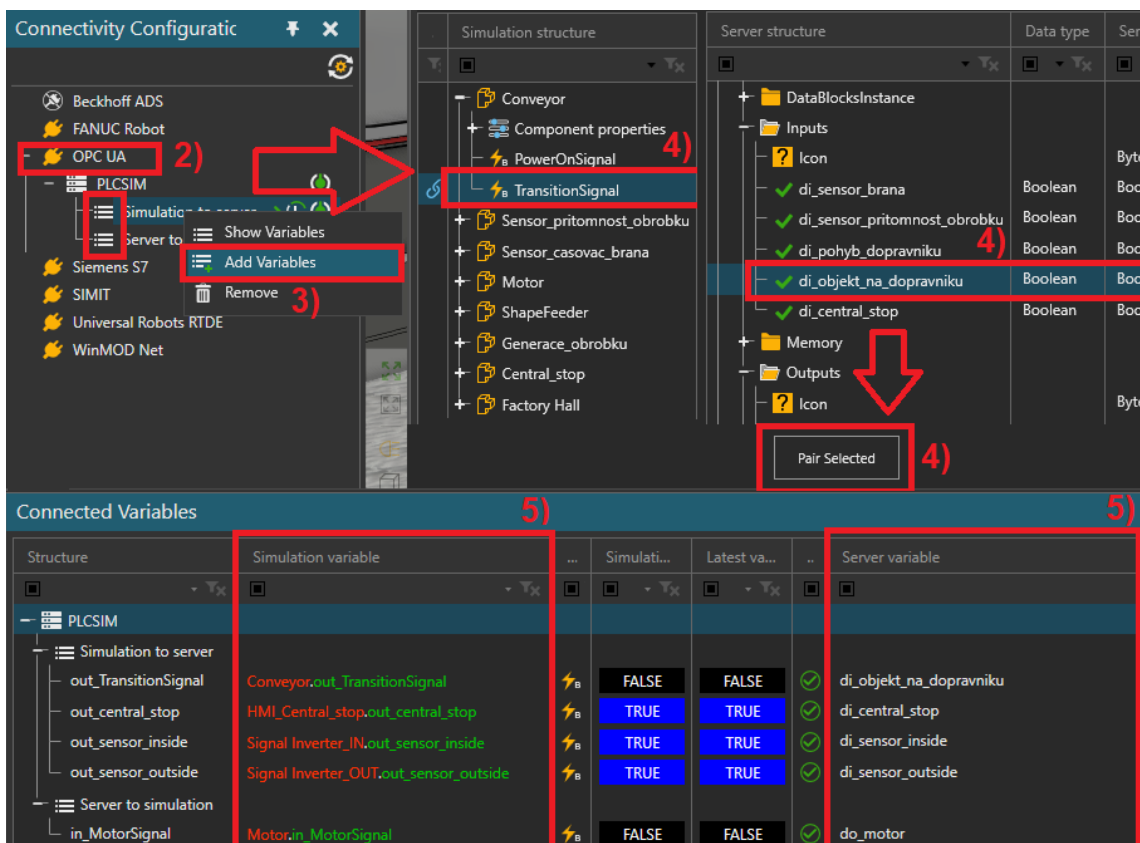
Každá z komponent virtuálního modelu, která je nějakým způsobem aktivní při simulaci, disponuje signály. Signály mohou být pouze interní (pro komunikaci v rámci VC) nebo i externě viditelné, tj. lze je spárovat s proměnnými na serveru, komunikujícím s PLC – v tomto případě server OPC UA. U tohoto pomocného modelu byly použity již připravené signály komponent z knihovny.

Pro zobrazení všech disponibilních signálů na scéně stačí stisknout tlačítko *Signals* v přihrádce *Connect*, která se vyskytuje na většině záložek horní lišty (např. *Modeling*, *Program* atd.), viz Obr. 41, a vybrat jednu z komponent obsahující signál. Při zcela zastavené (ne pozastavené) či resetované simulaci je možné jednoduše myší propojovat signály mezi sebou a určit tak, jestli se budou ovlivňovat i mezi sebou.



Obr. 41 – Zobrazení signálů scény

Posledním krokem je propojení vybraných signálů s korespondujícími proměnnými na PLC serveru – v tomto případě serveru OPC UA. Z toho také vyplývá, že lze párovat proměnné a signály pouze s připojeným OPC serverem, na kterém je již nahrán PLC program se všemi proměnnými, které chceme spárovat. Nicméně po odpojení OPC serveru hodnoty zůstanou spárovány a při příštím opětovném připojení není třeba nic opakovat – pouze se ve VC připojit k OPC serveru (viz kap. 4.2.1 a Obr. 31). Párování signálů simulace s PLC proměnnými probíhá následovně, viz Obr. 42:



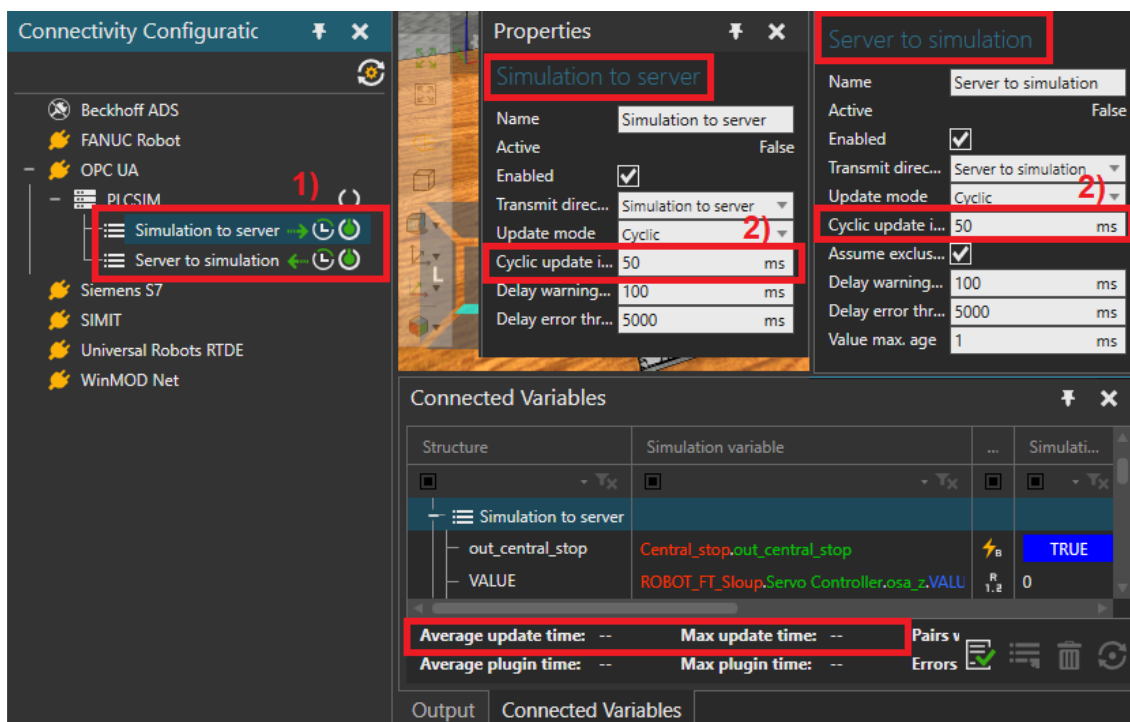
Obr. 42 – Propojení a zobrazení propojených proměnných mezi VC a PLC

- 1) V horní liště je třeba se přepnout do záložky *Connectivity* (viz Obr. 31)
- 2) V levém panelu (*Connectivity Configuration*) se přepnout do OPC UA a rozbalit všechna kaskádová menu

- 3) Na seznamu proměnných přes menu pravým tl. zvolit *Add Variables*
- 4) V nově otevřeném okně vlevo zvolit příslušný signál a k němu odpovídající proměnnou na OPC serveru a potvrdit tl. *Pair Selected*
- 5) Dole v rozhraní je nakonec možné vidět všechny aktuálně spárované proměnné, jejich datový typ i aktuální hodnoty při běhu simulace

Takt komunikačních cyklů ve VC

Takty komunikačních cyklů lze upravovat v záložce *Connectivity* vybráním typu komunikace v levém panelu (simulace → server nebo server → simulace), viz Obr. 43. Poté stačí změnit hodnotu *Cyclic update interval*, přičemž u modelu Dopravníku jsou oba směry komunikace nastaveny na výchozích 50 ms. V dolní části rozhraní lze také při běhu simulace kontrolovat aktuální průměrný a maximální *update time*.



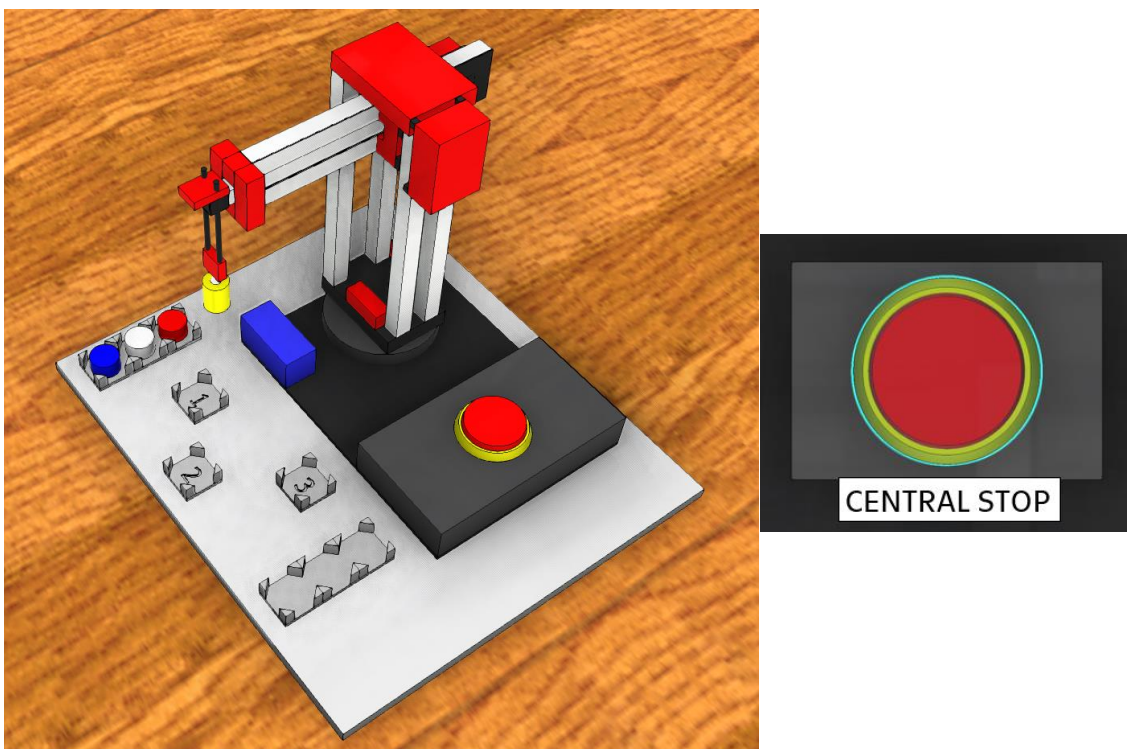
Obr. 43 – Konfigurace taktu komunikačních cyklů

4.3.5 Dílčí shrnutí

Byl vytvořen model dopravníku, na kterém je možné v rámci výuky názorně ukázat základní principy PLC programování. Také na něm byly v předchozích kapitolách vysvětleny základní principy tvorby virtuálních modelů v prostředí VC. Součástí vytvořeného modelu je taktéž připravený demo PLC projekt včetně HW konfigurace s definovanými proměnnými viz Tab. 1 a Tab. 2. Tento PLC projekt je společně s virtuálním modelem součástí XII. přílohy této práce. V rámci práce byl také vytvořen detailní uživatelský manuál, který je součástí I. přílohy této práce.

4.4 Tvorba virtuálního modelu FT č.1: Robot

Prvním z virtuálních modelů stavebnice FT je Robot (Obr. 44). Jedná se o tříosý (RTT) robot zakončený přísavkou. Robot má za úkol zakládání a odebrání obrobků do strojů, které jsou na pracovišti reprezentovány paletkami s čísly 1, 2, 3. Všechny osy robotu jsou poháněny mechanickým převodem, disponují svým vlastním odečítáním pozice inkrementálními enkodéry a jedním koncovým spínačem určeným pro referenci. Přísavka je poháněna pneumaticky za použití kompresoru (modrý kvádr na Obr. 44), který je nutné také řídit skrze PLC. Výše zmíněné snímače a aktory jsou řízené PLC signály, které jsou přehledně uvedeny v kap. 4.4.5 v Tab. 3, resp. Tab. 4.



Obr. 44 – Virtuální model FT: Robot (vlevo), ovládací tlačítko Robotu (vpravo)

4.4.1 Tvorba geometrie

Geometricky jednodušší části virtuálního modelu Robotu byly vytvořeny přímo ve VC, pomocí vestavěného modulu pro modelování (záložka *Modeling*), ve spolupráci s technickou podporou distributora VC – firmy SoliCAD. Tvarově složitější části za pomoci CAD softwaru Autodesk Inventor, které byly posléze importovány v nativním formátu, jež VC podporuje. Nejedná se o do detailu přesnou kopii reálné stavebnice – protože to ani není cílem – nýbrž z hlediska PLC řízení o funkčně identické dvojčce se zjednodušenou geometrií. Byly použity i následující připravené prvky z knihovny:

- *HMI Button* – ovládací tlačítko

4.4.2 Návrh ovládacích prvků virtuálního modelu

Obdobně jako model Dopravníku (kap. 4.3.2) obsahuje i model Robotu aktivní tlačítko přímo na scéně připravené k interakci – **CentralStop** (Obr. 44).

4.4.3 Oživení geometrie

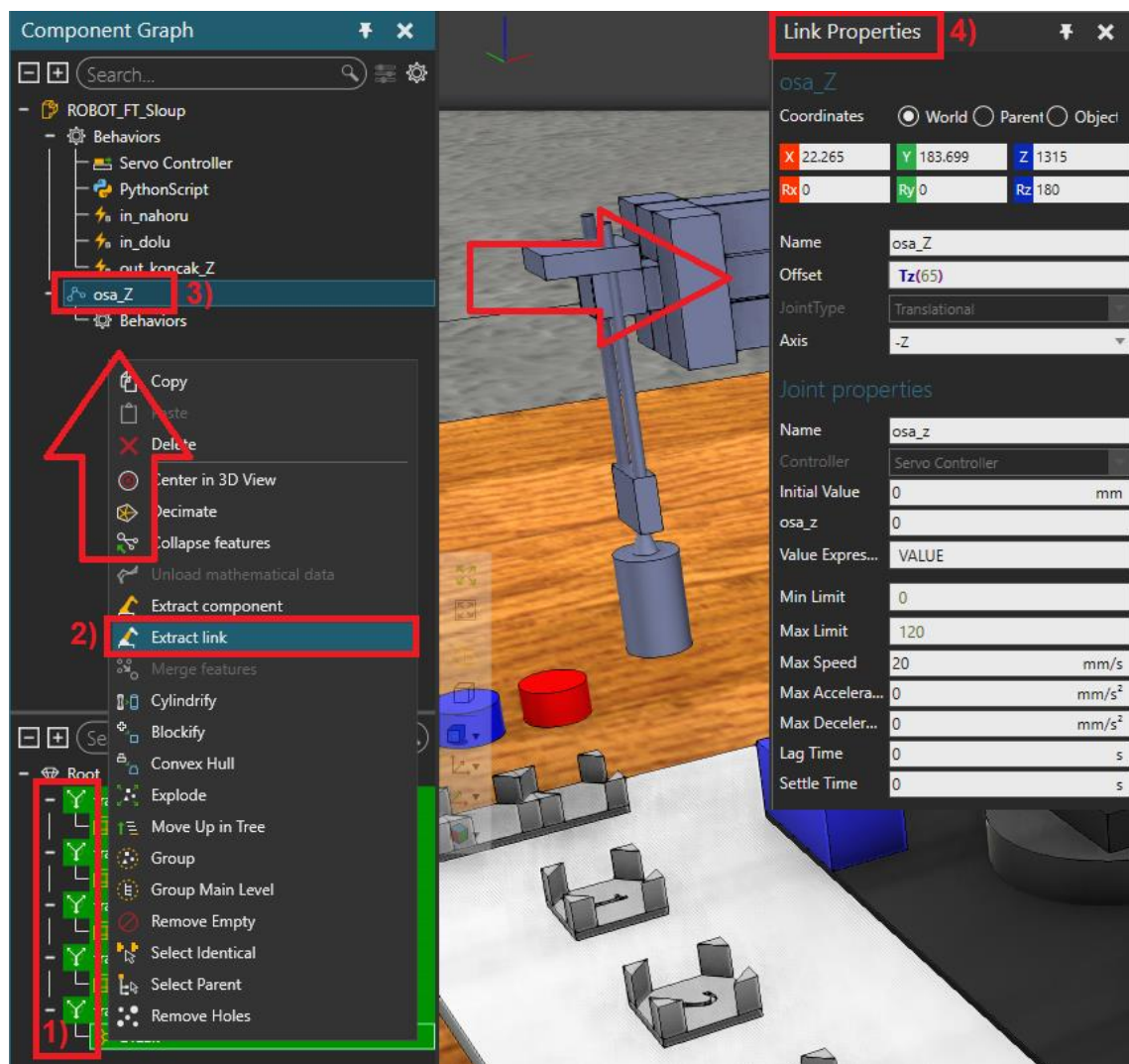
Na rozdíl od modelu Dopravníku, kde byly použity prvky téměř výhradně z knihovny, v případě modelu Robotu byla většina geometrie ručně modelována, a tudíž ji bylo třeba patřičně rozpohybovat (oživit):

Osy Robotu

Nejprve je třeba si ujasnit, co by měly osy Robotu zvládat:

- Translační nebo rotační pohyb řízený PLC, odpovídající stavebnici FT
- Možnost upravit zdvihy a rychlost pohybu, aby odpovídaly stavebnici FT
- Každá osa by měla mít inkrementální odečítání polohy s konfigurovatelným nulovým (počátečním) bodem zdvihu a startovací pozicí osy při zapnutí simulace
- Každá osa by měla obsahovat po jednom koncovém snímači v nulovém zdvihu

VC disponuje v módu (záložce) *Modeling* funkcí pro vytvoření příslušných lineárních nebo rotačních vazeb mezi dílčími částmi geometrie. Je výhodné, pokud jsou např. osy, které je třeba takto rozpohybovat, vymodelovány jako zvláštní díl. Je dobré proto na tento fakt myslet již při tvorbě geometrie např. v CADu.

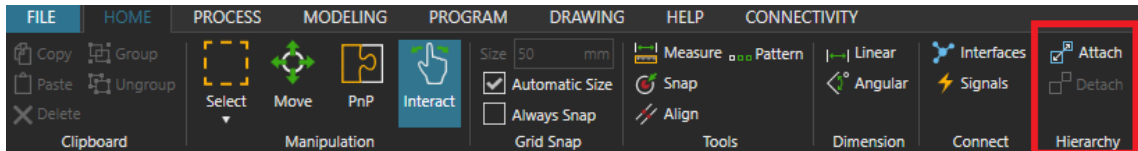


Obr. 45 – Vytvoření tzv. Linku

Následuje vytvoření tzv. *Linku*, který slouží pro definování geometrie, která má konat nějaký pohyb (akci/funkci) v rámci komponenty, a tento *Link* je pak možné konfigurovat bez ovlivnění zbytku komponenty, viz Obr. 45:

- 1) Výběr částí geometrie, ze kterých bude vytvořen *Link*
- 2) Pravým tl. na vybrané geometrii zvolit možnost *Extract link*

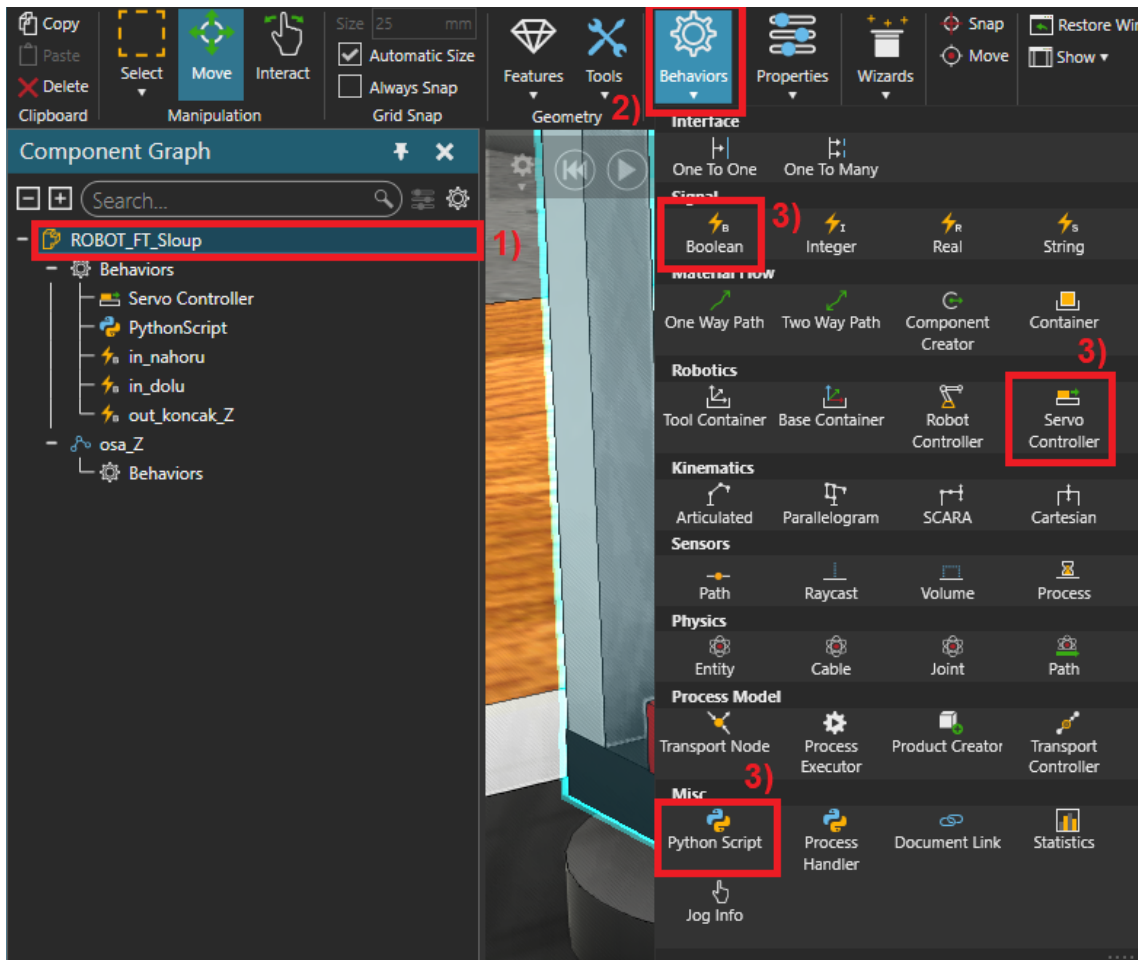
Pokud byla geometrie importována zvlášť po jednotlivých komponentách nebo byly komponenty tvořeny přímo zvlášť v modeláři VC (případ Robotu), musí se provést spojení geometrií, aby se spojené komponenty při pohybu navzájem následovaly. Slouží k tomu tlačítko *Attach* v přihrádce *Hierarchy* v záložce *Home*, viz Obr. 46.



Obr. 46 – Připojení geometrie pomocí funkce *Attach*

Po tomto rozdělení geometrie je možné přejít k rozpohybování. K tomu je zapotřebí ke každému *Linku* přidat z přihrádky *Behaviors* (Obr. 47):

- Vstupní a výstupní signály, které bude třeba k řízení PLC
- *ServoController*, který umožňuje konfiguraci většiny náležitostí týkající se pohybu osy
- Python skript, ve kterém je nutné definovat samotnou logiku pro řízení pohybu os



Obr. 47 – Konfigurace os pomocí *Behaviors*

Dále je třeba nastavení parametrů dané vazby přímo ve vytvořeném *Linku*, kde je nutné konfigurovat (viz Obr. 45, kroky 3 a 4):

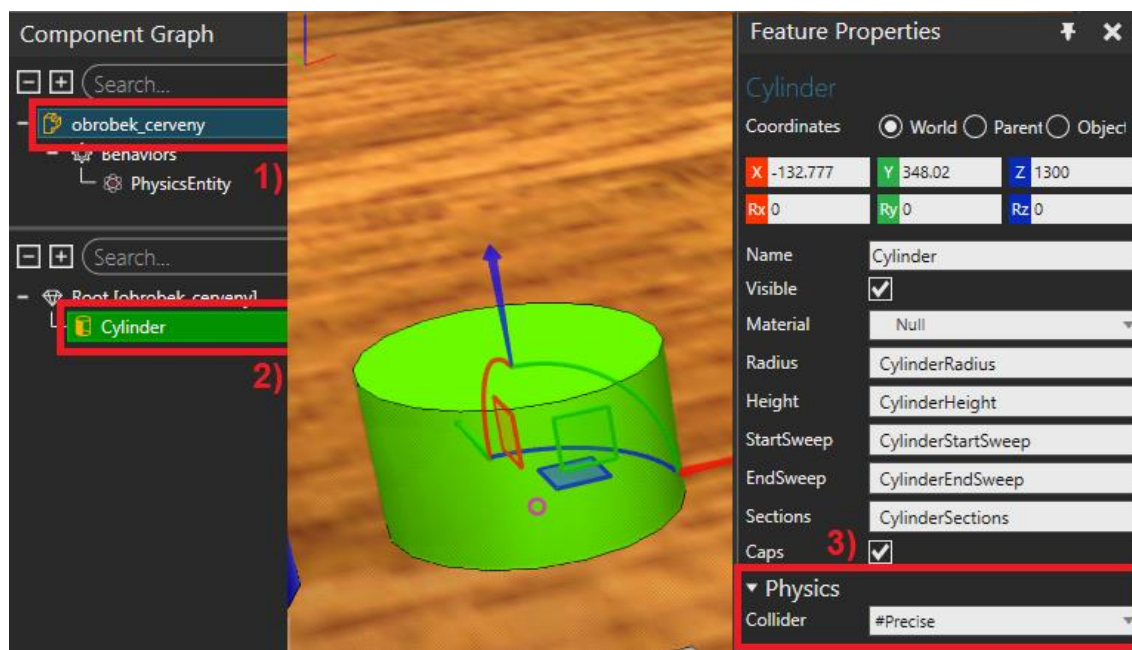
- Typ pohybu (translace/rotace)
- Směr pohybu osy (ve zvoleném souřadném systému)
- Připojení *ServoControlleru*
- Limitní hodnoty dráhy (zdvih osy)
- Rychlost pohybu osy apod.

Na závěr bylo třeba doplnit logiku k pohybu os v Python skriptu – v tomto případě za asistence technické podpory ze SoliCAD. Nejdříve je třeba si do skriptu importovat potřebné hodnoty z připravené osy a uložit je do proměnných pro další použití ve skriptu. Poté, velmi zjednodušeně, proběhne definice akcí (logiky), které mají nastat po přivedení signálu na danou osu (funkce *OnSignal*). A dále definice akcí, které mají nastat při spuštění simulace (funkce *OnRun*) – jedná se o hlavní funkci/smyčku skriptu. Zároveň je možné si definovat i svoje vlastní funkce nebo využít jiných nativních funkcí VC. U translačních os bylo definováno rozlišení pohybu desetina milimetru, u rotačních os rozlišení pohybu desetina stupně, což bylo otestováno jako více než dostatečné rozlišení pro přesné polohování modelu. V Python skriptu bylo pro každou pohybovou osu vytvořeno celkem cca 100 řádků kódu. Kód obsahuje i logiku pro simulaci inkrementálních enkodérů, která je blíže popsána v kap. 4.4.5.

Obrobky se simulovanou fyzikou

Aby obrobky odpovídaly co nejvíce realitě, byla na ně aplikována *PhysicsEntity* ve variantě *In Physics*, tj. kompletní simulování fyziky. Při upuštění obrobku z přísavky tak obrobek padá na podložku ve směru gravitace a interaguje s dalšími obrobky, potažmo objekty s *PhysicalEntity*.

Drobnou úpravou oproti výchozím hodnotám byla změna tzv. *Collideru* (nastavení kolizní obálky obrobku) na *Precise*, aby lépe kopíroval celý válec a více tak odpovídal realitě (viz Obr. 48). V opačném případě lze *Collider* zjednodušit i na *Box*, pokud je třeba.



Obr. 48 – Konfigurace Collideru obrobků

Přísavka s kompresorem

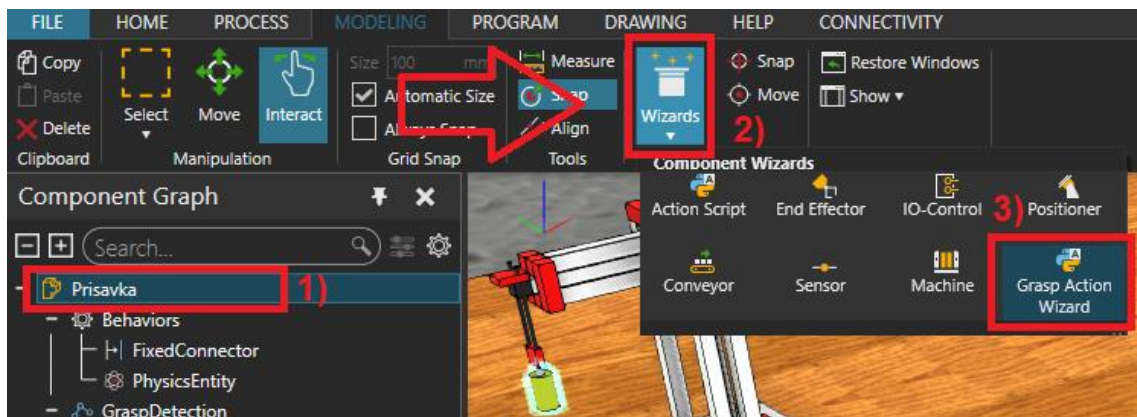
Dalším z hlavních funkčních pilířů Robotu vedle pohybujících se os je zakončení přísavkou, která by měla v rámci možností simulovat realitu. K tomu je do simulace zapotřebí doplnit taktéž kompresor, který je reálně nezbytný pro funkci přísavky – nicméně ten je simulován pouze přidáním signálem do podmínky pro aktivaci přísavky, takže je popsán až v kap. 4.4.5.

Po vytvoření *Linku* je třeba zajistit, aby se geometrie uchopovaného objektu, v tomto případě obrobku, ve správný moment „přichytila“ ke geometrii koncového efektoru – přísavky. Tato problematika je poměrně složitá:

- Za běžící simulace je zapotřebí nadřazený uzel (*Parent node*) uchopovaného předmětu se simulovanou fyzikou připojit ke geometrii přísavky, potažmo jejímu nadřazenému uzlu
- Při upuštění je zase potřeba nadřazený uzel odpojit od uzlu přísavky, aby spolu geometrie nadále nebyly spojené
- Zároveň při uchopení změnit simulovanou fyziku objektu na kinematickou vazbu
- Při upuštění je zase nutné vrátit zpět simulovanou fyziku, aby objekt po upuštění dopadl na podložku
- Také je potřeba definovat uchopovací prostor, který tak vytvoří toleranční pásmo, ve kterém je možné objekt uchytit

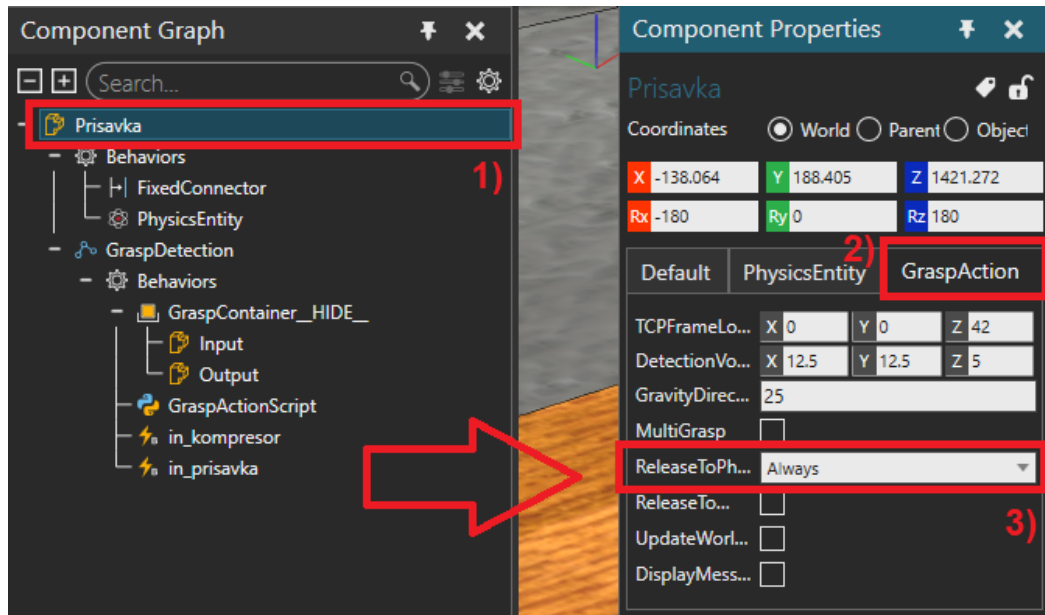
Nicméně komunita VC v průběhu let vytvořila funkci, která vše podstatně zjednodušuje – avšak musí se pro tvorbu modelu do VC importovat externě. Jedná o tzv. *Grasp Action Wizard*, který lze po importu přidat na náš vytvořený *Link*, viz Obr. 49. Tento balíček obsahuje konfigurovaný:

- Python skript se základní logikou pro přísavku
- Booleovský signál pro aktivaci přísavky
- *Container* (příhrádka *Behaviors*) – k definování uchopovacího prostoru



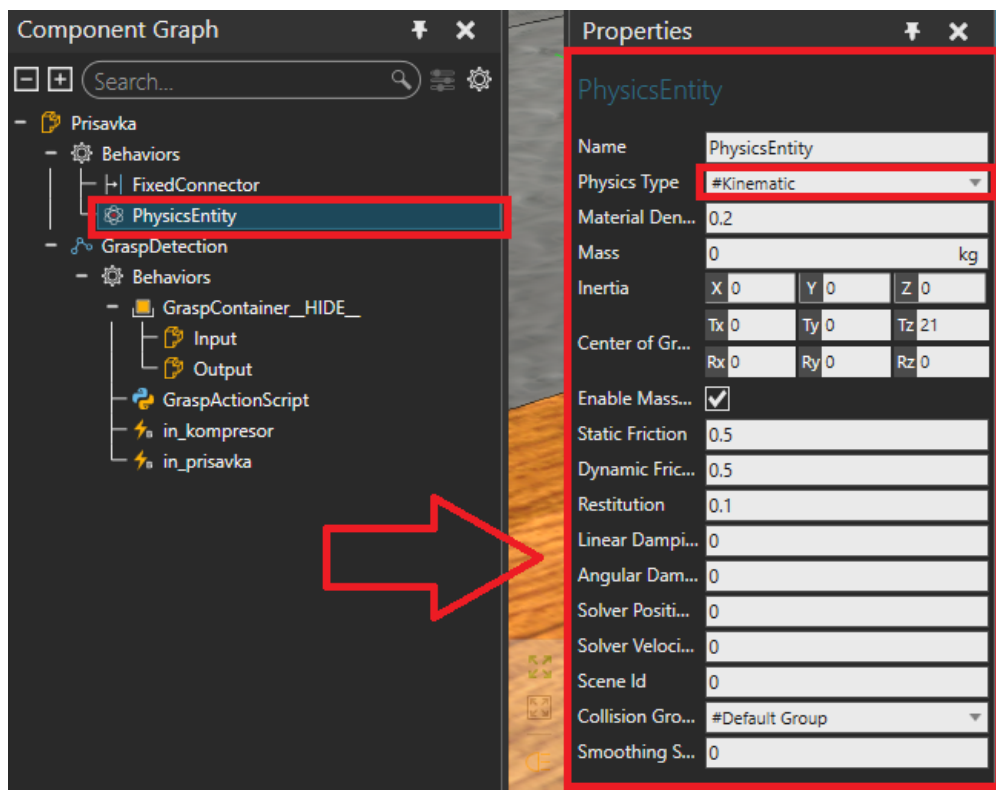
Obr. 49 – Přidání balíčku *Grasp Action Wizard*

Po přidání balíčku je možné již v *Component Properties* přísavky (Obr. 50) nastavovat základní vlastnosti úchopu, jako lokace a velikost detekčního prostoru, směr gravitace, ale také velmi důležitá vlastnost pro tento případ – volba upuštění obrobku vždy do režimu simulované fyziky (*ReleaseToPhysics* → *Always*). Obrobek tak po upuštění dopadne na nejbližší geometrii s kinematickou nebo fyzikální vazbou ve směru nastavené gravitace.



Obr. 50 – Konfigurace přísavky

Komponentám, *Linkům* nebo i jednotlivým plochám lze totiž z příhrádky *Behaviors* přiřadit tzv. *PhysicsEntity* (Obr. 51). Jednou z variant je kompletní fyzikální simulace (*In Physics*) se svým těžištěm, hustotou, objemem, koeficientem tření, elasticitou atd. Další variantou je kinematická vazba (*Kinematic*) zajišťující interakci s fyzikálními objekty – vytvoří fyzikální obálku geometrie, ale s pevnou vazbou k hlavnímu členu (*Parent node*). Na rozdíl od vazby *In Physics* na objekty nepůsobí gravitace ani dynamika z jiných fyzikálních objektů. Proto byla přidána kinematická vazba na všechny části geometrie, o které je žádoucí, aby se obrobky se simulovanou fyzikou zastavily a nepropadly skrz.



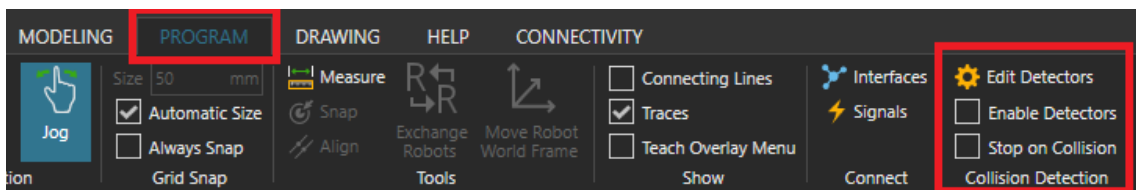
Obr. 51 – Konfigurace PhysicsEntity

HMI Button

Pro simulování chování skutečného CentralStopu byla provedena stejná inverze signálu jako v případě modelu Dopravníku, viz kap. 4.3.3.

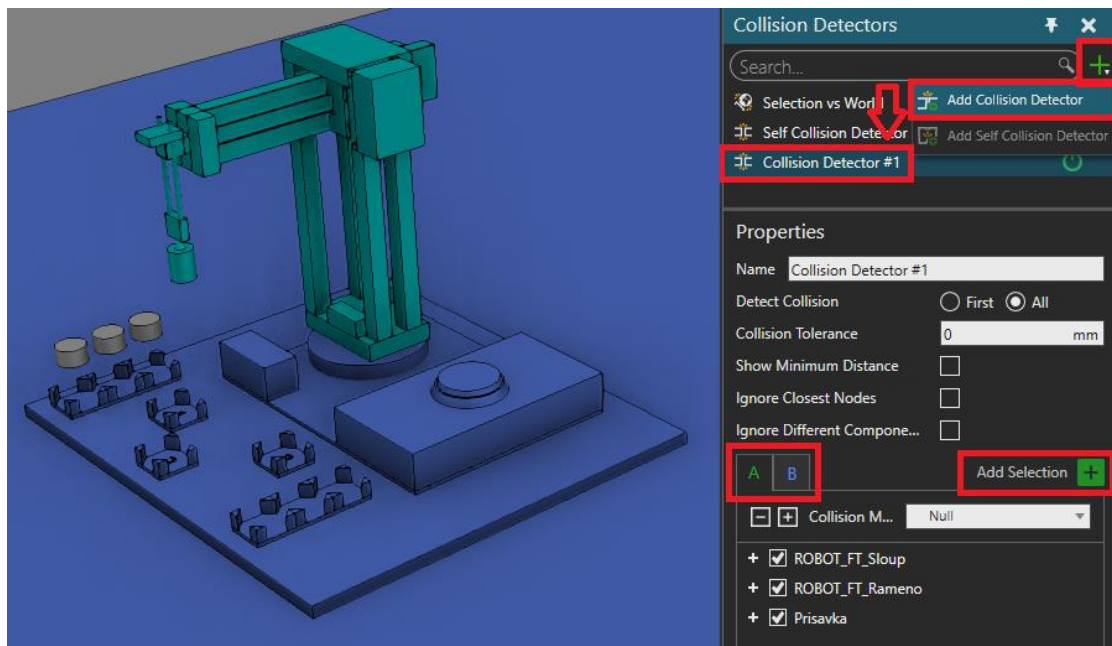
4.4.4 Kolize kinematiky

V případě modelu Robotu může při simulaci dojít ke kolizi geometrie. VC k tomu má upravenou funkci v záložce *Program* v přihrádce *Collision Detection* (viz Obr. 52). Je možné aktivovat buď pouze detektory kolize, které vizuálně (změnou barvy) upozorňují obsluhu na kolidující komponenty nebo aktivovat pozastavení simulace při detekování kolize – hodnoty PLC proměnných v momentě kolize zůstávají uloženy. Je tak možné provést diagnostiku, za jakých okolností ke kolizi došlo.



Obr. 52 – Aktivace detektoru kolizí

Dále je třeba konfigurovat, které komponenty jsou zahrnuty do kolize, a to skrze tlačítko *Edit Detectors* na Obr. 52. Nejprve je třeba vybrat pod symbolem „plus“ *Add Collision Detector* a poté příslušné komponenty rozdělit do zeleně zbarvené kategorie A a do modře zbarvené kategorie B. Výběr lze provést zvolením příslušné kategorie a kliknutím na *Add Selection* pro přidání komponent (viz Obr. 53). Do kategorie A byly vybrány všechny komponenty, které při simulaci konají nějaký pohyb (a tudíž mohou kolidovat). Do kategorie B byly přidány všechny statické komponenty scény. Důležité je, ani do jedné kategorie nepřidávat manipulované obrobky, protože by hlásily kolizi při každém zvednutí nebo dopadu na podložku. Navíc se objekty s kompletně simulovanou fyzikou (obrobky) při běhu simulace ani nemohou dostat skrze jiné součásti s kinematickou vazbou.



Obr. 53 – Konfigurace detektoru kolizí Robotu

4.4.5 Tvorba I/O pro propojení a řízení PLC

V Tab. 3 a Tab. 4 následuje přehled zvolených signálů pro vstup a výstup (I/O) pro PLC řízení modelu Robotu:

Tab. 3 – Zvolené signály pro PLC vstupy (simulace → PLC)

Datový typ	Název signálu VC	Název PLC proměnné
Real	out_ekoder_Z	High_Speed_Counter_1.MeasuredValue
Real	out_ekoder_Y	High_Speed_Counter_2.MeasuredValue
Real	out_ekoder_X	High_Speed_Counter_3.MeasuredValue
Bool	out_koncak_Z	di_snimac_osa_z
Bool	out_koncak_Y	di_snimac_osa_y
Bool	out_koncak_X	di_snimac_osa_x
Bool	out_central_stop	di_total_stop

Tab. 4 – Zvolené signály pro PLC výstupy (PLC → simulace)

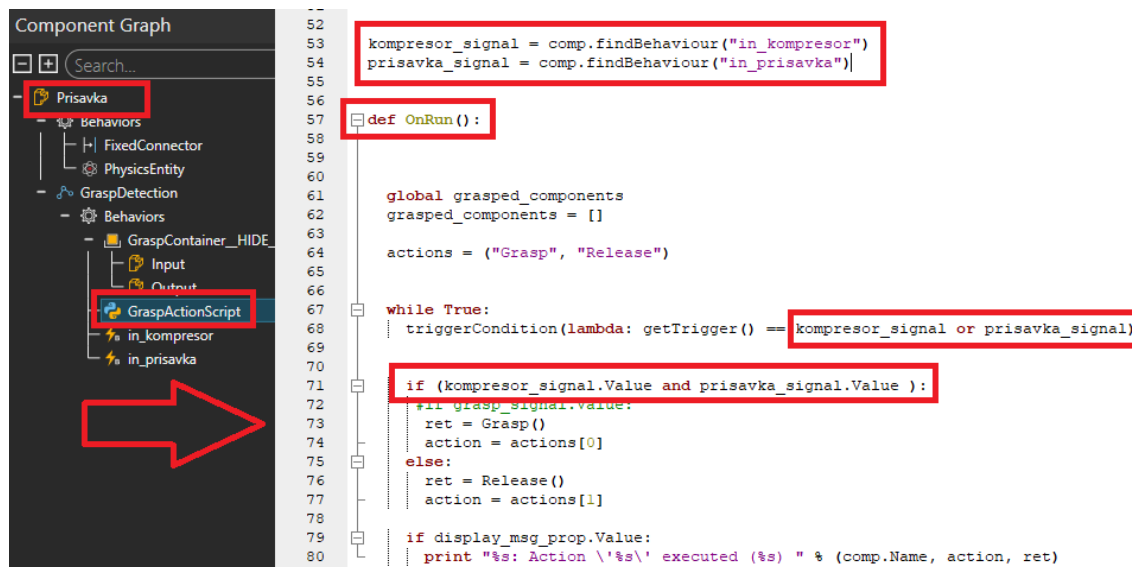
Datový typ	Název signálu VC	Název PLC proměnné
Bool	in_nahoru	do_osa_z_nahoru
Bool	in_dolu	do_osa_z_dolu
Bool	in_zasunout	do_osa_y_vzad
Bool	in_vysunout	do_osa_y_vpřed
Bool	in_rotace_minus	do_osa_x_smer
Bool	in_rotace_plus	do_osa_x_protismer
Bool	in_kompresor	do_kompresor
Bool	in_prisavka	do_vacuum

Všechny signály, s výjimkou CentralStopu a přísavky, bylo třeba vytvořit manuálně a zároveň v Python skriptu definovat a zajistit jejich propojení se zbytkem modelu.

Ve výchozím stavu poskytuje *ServoController* jednotlivých os odečítání pozice absolutně. Stavebnice FT nicméně disponuje inkrementálními enkodéry, a proto byla pomocí Python skriptu vytvořena logika pro simulaci inkrementálního enkodéru. V případě, že je zapnutá simulace, vezme se aktuální hodnota pozice osy (vestavěného absolutního enkodéru) a odečte se od ní startovací pozice (v *ServoControlleru* tzv. *initial value*), která odpovídá pozici osy při zapnutí simulace. Po zapnutí simulace tedy vždy bude inkrementální enkodér ukazovat hodnotu „0“, bez ohledu na to, v jaké pozici se bude osa nacházet při zapnutí simulace. Pro translační pohyb je hodnota v [mm] a pro rotační pohyb ve [°].

Signály koncových snímačů a pohybu os se váží na kap. 4.4.3, kde byly přidány do modelu Robotu. V Python skriptu bylo následně třeba definovat, aby se osa při příchozím signálu o hodnotě True dala do pohybu a v pohybu setrvala až do příchozího signálu o hodnotě False, kdy se zastaví. Zároveň bylo definováno vyslání signálu koncového snímače o hodnotě True při příjezdu osy k její nulové pozici – s tolerančním pásmem o velikosti rozlišení polohování, viz kap. 4.4.3.

Následně bylo třeba upravit Python skript přísavky, aby byla přísavka aktivována pouze v případě, že je aktivovaný i kompresor – tak jak je tomu ve skutečnosti. V případě, že je jeden ze signálů False, objekt se nechopí nebo případně upustí. Do skriptu bylo třeba načíst a uložit do proměnné signál kompresoru a dále upravit *triggerCondition* a další podmínku ve funkci *OnRun*, viz Obr. 54. Upraveno bylo několik řádků kódu z celkových cca 290 řádků.



Obr. 54 – Úprava skriptu pro použití přísavky a kompresoru

Finální propojení signálů s proměnnými z OPC serveru proběhlo ve stejném duchu jako u modelu Dopravníku v kap. 4.3.4.

Takt komunikačních cyklů ve VC

Takt komunikačního cyklu pro směr simulace → PLC byl zrychlen na 10 ms. Tato úprava byla provedena kvůli rychlejšímu odečítání hodnot enkodérů – jedná se o stěžejní prvek tohoto modelu z hlediska PLC řízení. Takt opačného směru komunikace PLC → simulace byl ponechán na výchozí hodnotě 50 ms. V dolní části rozhraní lze také při běhu simulace kontrolovat aktuální průměrný a maximální *update time*, viz kap. 4.3.4.

4.4.6 Demo PLC projekt modelu Robotu

V rámci zadání práce byl vytvořen program pro PLC řízení pracoviště pro virtuální model Robotu. K realizaci byl použit software Siemens TIA Portal V15.1 a jazyk strukturovaného textu dle normy IEC61131-3. Zde je princip demo PLC projektu stručně popsán, přičemž detailní popis s návodem je součástí VI. přílohy této práce.

PLC program umožňuje ovládat Robot v ručním a automatickém režimu pomocí HMI obrazovek. Ruční režim umožňuje volně pohybovat s osami Robotu a ovládat všechny jeho dostupné prvky. Zahrnuty byly i bezpečnostní prvky, včetně tlačítka CentralStop na každé obrazovce, a softwarově byly ošetřeny všechny závažné chybové stavy. Před ruční manipulací i automatickým provozem se Robot nejdříve musí referovat, tj. najet všemi osami na koncové spínače pro nastavení správného odečítání enkodérů. Před referováním lze s osami ručně pohybovat pouze ve směru ke koncovým spínačům, aby se zamezilo případné kolizi Robotu s okolní zástavbou předtím, než jsou správně nastaveny enkodéry. Přepínač CentralStop je ve výchozí pozici (po prvotním spuštění) nastaven jako sepnutý, tj. aktivní. Při interakci s ním

(v modelu) se obvod rozpojí a okamžitě zastaví pohyby všech os, avšak ponechá zapnutý kompresor a přísavku, aby případně nedošlo k neplánovanému upuštění polotovaru. Po přepnutí přepínače opět do aktivní polohy, zůstane model zastavený až do stisku tlačítka POKRAČOVAT V PROVOZU na HMI panelu – stejný postup následuje také v případě stisku tlačítka CENTRAL STOP na HMI panelu.

Pro spuštění samotné automatizace je potřeba zadat pořadí pracovišť a příslušné časy simulující délku výroby na jednotlivých pracovištích. Po spuštění automatizace začne Robot přemísťovat polotovary ze vstupního zásobníku na určená pracoviště a na konci jsou zhotovené obrobky umístěny na výstupní zásobník.

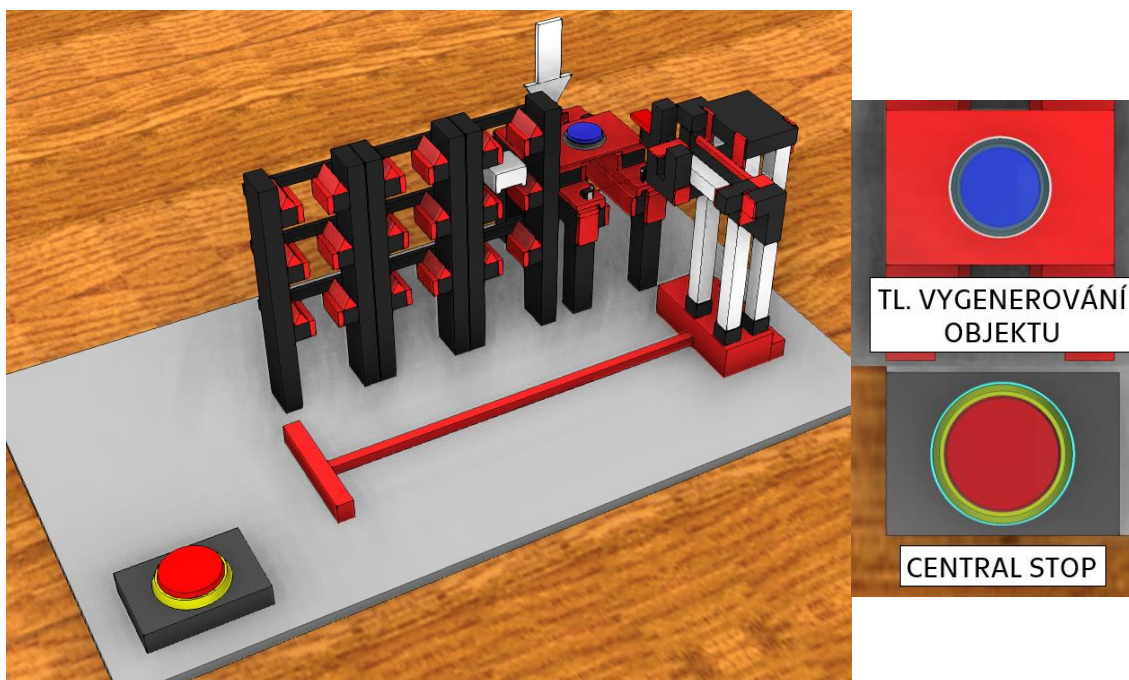
4.4.7 Dílčí shrnutí

Byl vytvořen model Robotu, který je možné využít pro virtuální zprovoznění fyzické stavebnice FT. Také na něm byly v předchozích kapitolách vysvětleny důležité principy tvorby virtuálních modelů v prostředí VC. Součástí vytvořeného modelu je též připravený prázdný PLC projekt – pouze s HW konfigurací a s definovanými proměnnými viz Tab. 3 a Tab. 4. Tento PLC projekt je společně s virtuálním modelem součástí VIII. přílohy této práce. V rámci práce byl také vytvořen detailní uživatelský manuál, který je součástí II. přílohy této práce.

Součástí modelu Robotu je též připravený demo PLC projekt, který byl hrubě popsán v kap. 4.4.6 a podrobněji vysvětlen v VI. příloze této práce. Kompletní demo včetně PLC programu je součástí XIII. přílohy této práce.

4.5 Tvorba virtuálního modelu FT č.2: Regálový zakladač

Druhým z virtuálních modelů stavebnice FT je Regálový zakladač, zkráceně Zakladač (Obr. 55). Jedná se o tříosý (TTT) zakladač, určený pro manipulaci dílců (palet) mezi obousměrným dopravníkem a regálem s devíti pozicemi. Dopravník i všechny osy zakladače jsou poháněny mechanickým převodem. Osy Y a Z disponují svým vlastním odečítáním pozice inkrementálními enkodéry a jedním koncovým spínačem určeným pro referenci. Osa X disponuje místo enkodéru dvěma koncovými spínači, protože není potřeba řídit polohu vysouvání a zasouvání na přesnou hodnotu. Dopravník obsahuje také dvě optické závory, jednu na vstupu a druhou na výstupu. Výše zmíněné snímače a aktory jsou řízené PLC signály, které jsou přehledně uvedeny v kap. 4.5.5 v Tab. 5, resp. Tab. 6.



Obr. 55 – Virtuální model FT: Regálový zakladač (vlevo), ovládací tlačítka Zakladače (vpravo)

4.5.1 Tvorba geometrie

Celková sestava byla vymodelována v CAD softwaru Autodesk Inventor a posléze importována v nativním formátu, jež VC podporuje. Jedná se z hlediska PLC řízení o funkčně identické dvojče se zjednodušenou geometrií. Byly použity i následující připravené prvky z knihovny:

- *HMI Button* – ovládací tlačítko
- *Conveyor Sensor* – optická brána
- *Boolean Signal Inverter* – prvek pro převrácení hodnoty Bool signálu

4.5.2 Návrh ovládacích prvků virtuálního modelu

Obdobně jako model Dopravníku (kap. 4.3.2) obsahuje i model Zakladače aktivní tlačítka přímo na scéně připravené k interakci (Obr. 55):

- **CentralStop**

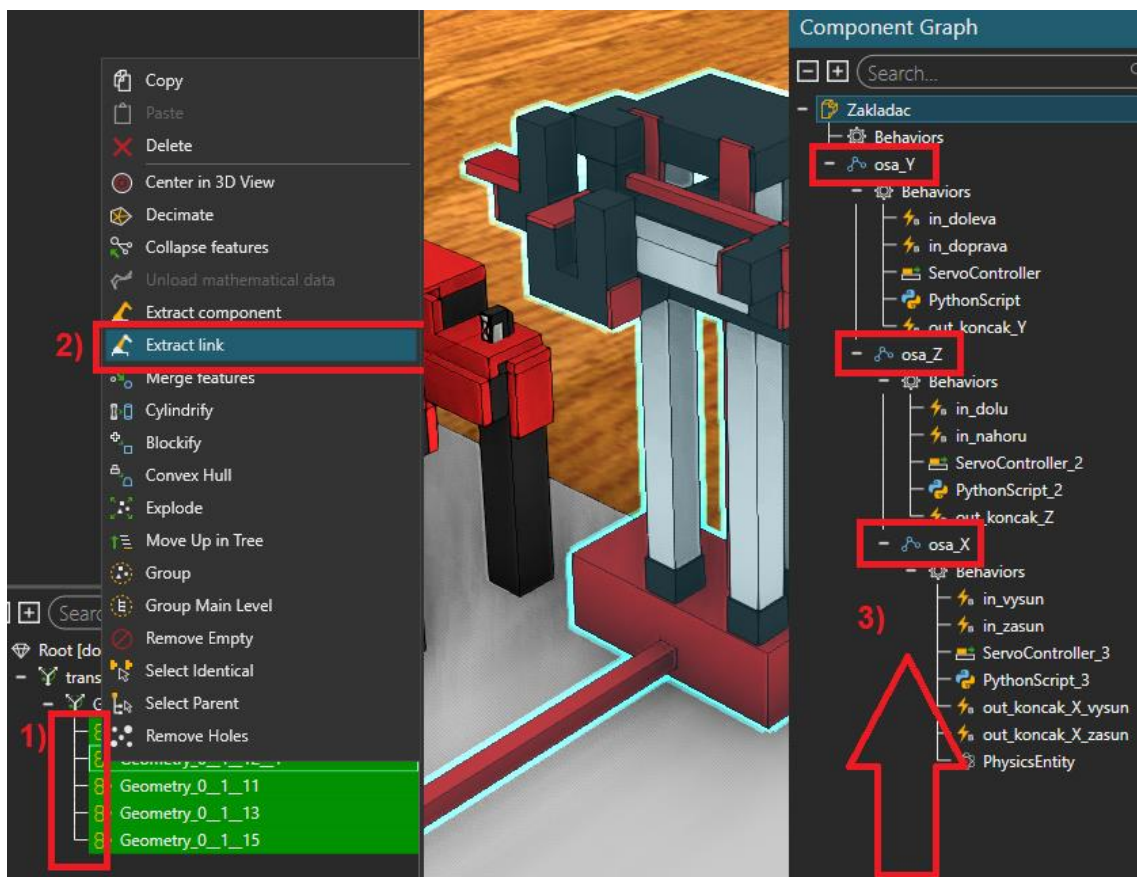
- **Tlačítko pro generování objektu** – po jeho stisku dojde k vygenerování příslušného objektu

4.5.3 Oživení geometrie

V případě Zakladače byla geometrie až na pár výjimek importovaných z knihovny VC vytvořena opět ručně, a proto bylo třeba provést její oživení v podobném duchu jako u Robotu:

Osy Zakladače

Pokud byla geometrie importována jako celková sestava z CADu (případ Zakladače), stačí vybrat dílčí komponentu sestavy, její geometrii (nebo i vícero najednou), a stiskem pravého tl. vybrat *Extract link* (Obr. 56, vlevo) – tento proces lze i řetězit a vytvořit tak několik na sebe navazujících *Linků* (Obr. 56, vpravo). Dojte tak k extrahování vybrané geometrie z nadřazené sestavy, kterou lze upravovat nezávisle na ostatních. Má to ale i tu výhodu, že nový *Link* je pořád připojený k nadřazenému *Linku*, a nakonec i k celkové sestavě. Tudiž, když se pohne první *Link* *osa_Y*, tak ji zbylé *Linky* (geometrie) – *osa_Z* a *osa_X* – následují v jejím pohybu.



Obr. 56 – Připojení geometrie pomocí funkce *Extract link*

Dále bylo postupováno analogicky jako v případě Robotu (kap. 4.4.3), tj. pro každou pohybující se komponentu přidat z *Behaviors* příslušné signály, *ServoController* a Python skript. Dále provést jejich konfigurace a vytvořit logiku pro samotný pohyb a řízení skrze signály.

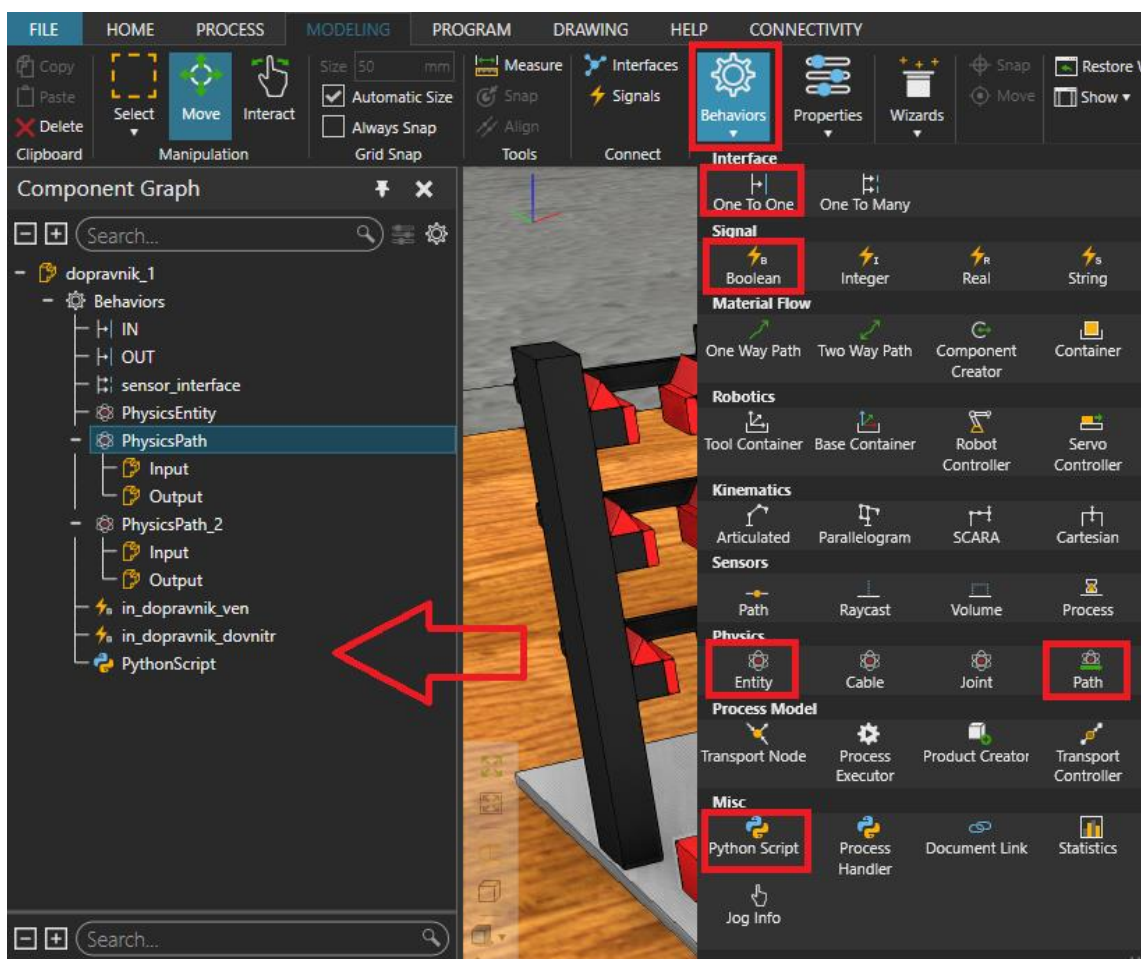
Dopravník

Na rozdíl od modelu Dopravníku byl zde dopravník modelován ručně, a proto bylo třeba celou geometrii dopravníku oživit. Nejprve je třeba uvést, že dopravník má disponovat:

- Možností řízení pohybu v obou směrech pomocí PLC
- Připojením ke generátoru palet (popsán v dalším odstavci)
- Schopností přijmout a odevzdat objekty se simulovanou fyzikou a transportovat je
- Možností konfigurovat rychlost pohybu dopravníku pro korelaci se stavebnicí FT

Po přípravě geometrie extrakcí a vytvořením *Linků* bylo třeba z *Behaviors* přidat několik prvků (Obr. 57):

- Signály pro ovládání pohybu dopravníku
- *One to One Interface* – pro možnost připojení k dalším komponentám (generátor palet)
- *PhysicsEntity* – pro definování kinematické vazby, aby palety nepropadly skrze dopravník
- *PhysicsPath* – funkce zajišťující pohyb objektů se simulovanou fyzikou po definované cestě
- Python skript – pro definici zbylé logiky

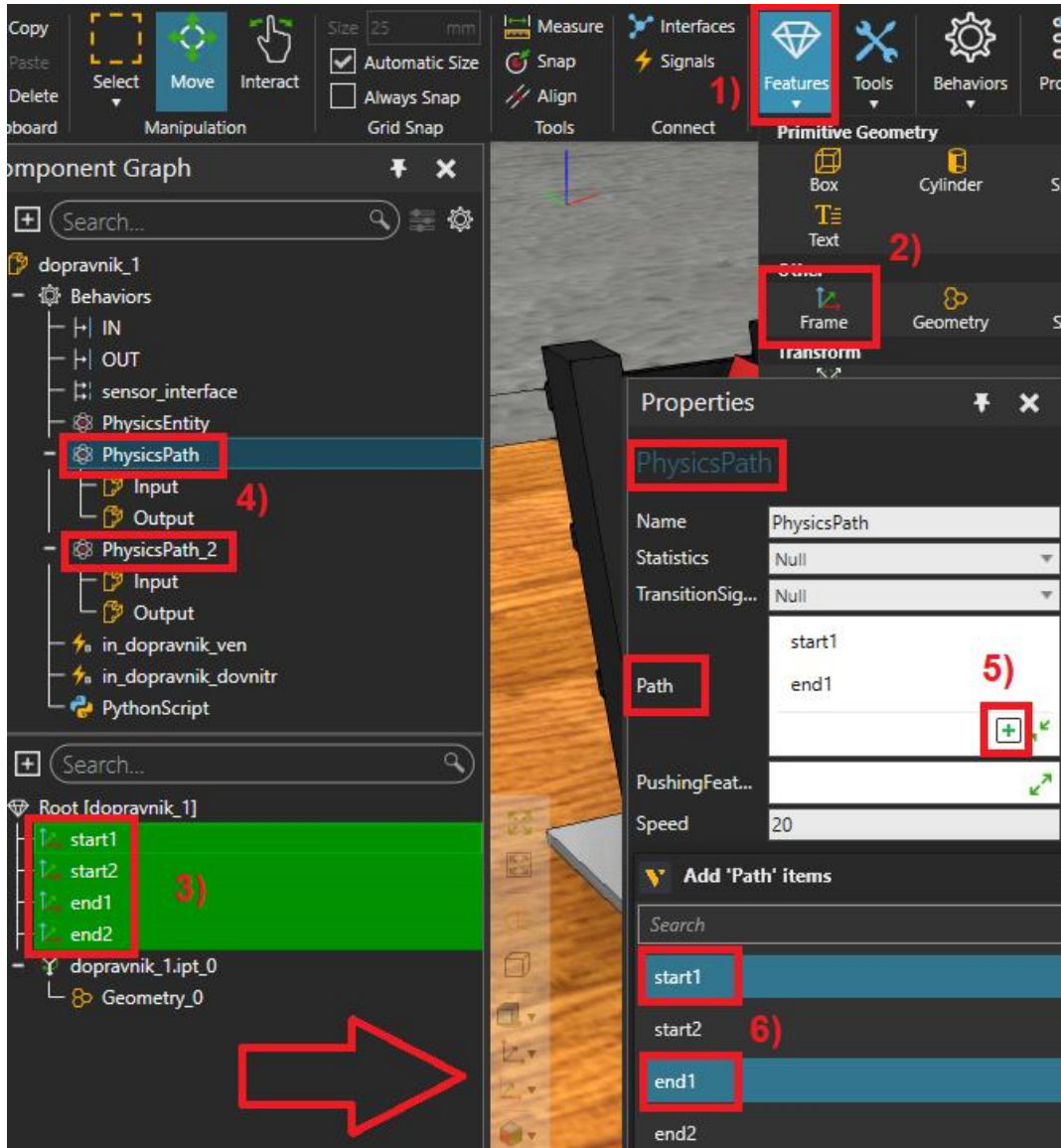


Obr. 57 – Konfigurace dopravníku pomocí Behaviors

Pro konfiguraci pohybu objektů po dopravníku za pomoci *PhysicsPath* je třeba nejprve vytvořit cestu (*Path*), po které má být pohyb vykonán. To se provede skrze umístění tzv. rámců (*Frame*) – pro přímočarý pohyb mezi dvěma body postačují dva *Framey*, pro složitější pohyby jich lze přidávat vícero (viz Obr. 58):

- 1) Vybrat příhrádku *Features*
- 2) Zvolit *Frame* a umístit ho v prostoru

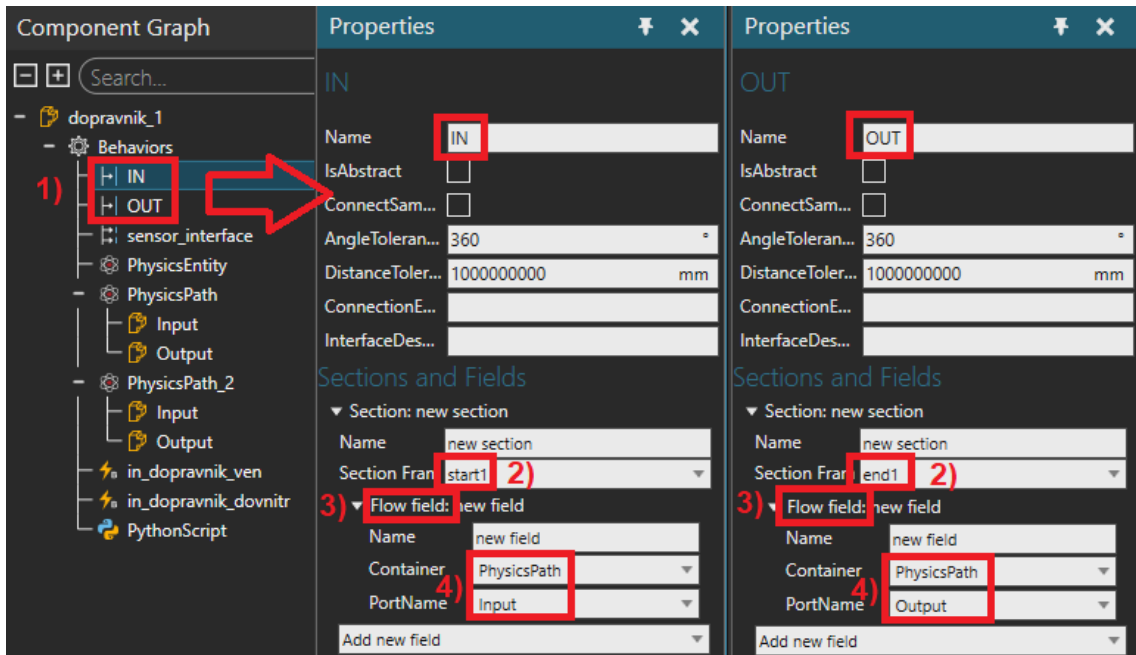
- 3) V levé dolní části obrazovky jsou přidány *Framy* přehledně vidět
 - 4) Vybrat příslušnou cestu *PhysicsPath* (v tomto případě postupně pro obě cesty)
 - 5) Stisknout symbol plus
 - 6) Vybrat příslušné *Framy*, které nyní definují počáteční a koncový bod pohybu dopravníku
- Obousměrnost dopravníku byla vyřešena vytvořením dvou cest (každá pro jeden směr), každá se svým párem *Framů* pro počátek a konec cesty.



Obr. 58 – Konfigurace *PhysicsPath*

Posledním krokem je definice vstupu a výstupu dopravníku pro připojení materiálových toků dalších komponent (generátor palet), viz Obr. 59:

- 1) Zvolit příslušný *One To One Interface*
- 2) Vybrat správný *Frame* pro určení lokace vstupu/výstupu v *Section Frame*
- 3) Přidání *Flow field* (materiálového toku)
- 4) Do *Container* vybrat příslušnou *PhysicsPath* a její port (*Input*, *Output*)

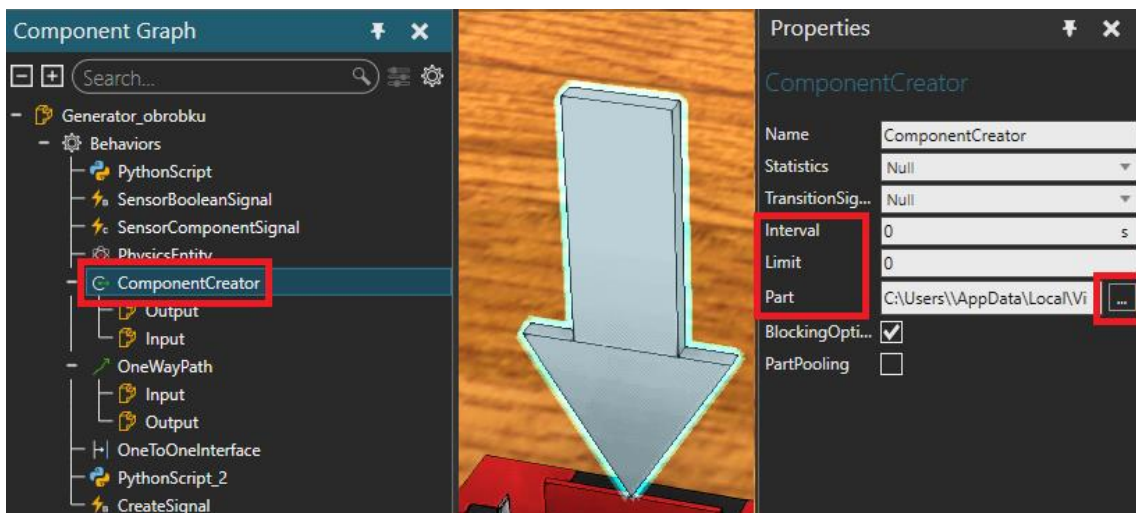


Obr. 59 – Konfigurace One to One Interface

Generátor palet

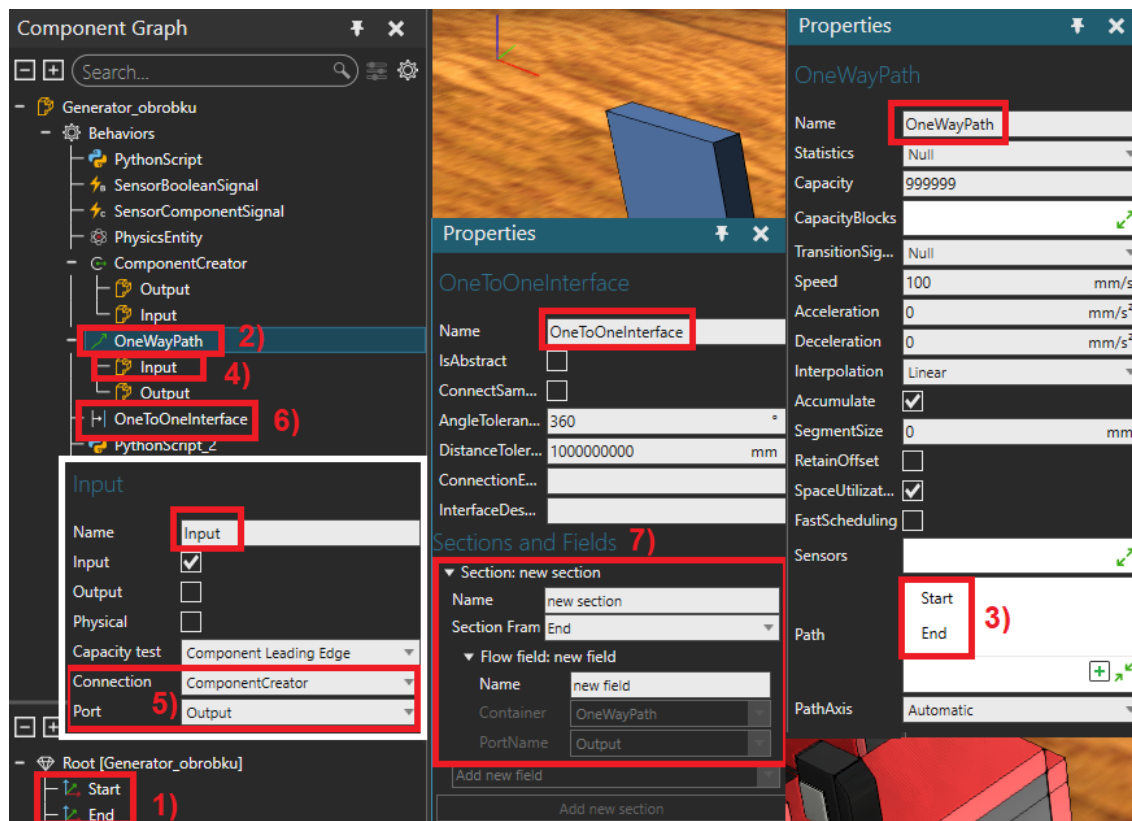
Jako náhrada za manuální ruční vkládání palet u stavebnice FT byl vytvořen generátor palet, který po stisku HMI tlačítka vygeneruje a transportuje paletu na dopravník. Generátor je reprezentován geometrií ve tvaru bílé šipky. Z *Behaviors* bylo opět třeba přidat několik prvků:

- Potřebné signály pro PLC řízení
- *PhysicsEntity* – pro definování kinematické vazby
- *OneWay Path* – funkce zajišťující pohyb objektů po definované cestě
- *One To One Interface* – pro možnost připojení k dalším komponentám
- *Component Creator* – pro generování zvoleného objektu (palety) při simulaci
- Python skript – pro definici zbylé logiky



Obr. 60 – Konfigurace Component Creator

Při konfiguraci generátoru je možné např. zvolit časový interval, při kterém se vybrané objekty budou generovat, nebo limit počtu součástí, které se vygenerují. Nicméně pro případy všech virtuálních modelů v této práci jsou tyto hodnoty nastaveny na nulu, protože je žádoucí paletu generovat jen při vyslání pokynu/signálu (Obr. 60). Dalším bodem konfigurace je již volba samotného objektu, který bude generován – stačí vybrat z uložených souborů, viz Obr. 60.



Obr. 61 – Konfigurace Generátoru palet

Dalším krokem je konfigurace materiálového toku (Obr. 61). Opět je třeba přidat *Frame* pro určení polohy začátku a konce materiálového toku. Konec bude zároveň sloužit jako přípojka k další komponentě – dopravníku z minulého odstavce. Počátek souřadného systému generované palety poté putuje přesně podle definované cesty mezi *Framy*. Úpravou pozice *Framů* tak lze měnit trajektorii (vytvořit offset) pohybu palet, viz Obr. 61:

- 1) Lze přidat nové nebo použít stávající *Framy*
- 2) Vybrat *OneWay Path*
- 3) Pomocí symbolu plus přidat příslušné *Framy*
- 4) Pod *OneWay Path* vybrat *Input* (vstupní port)
- 5) Propojit s výstupním portem *Component Creatoru*
- 6) Vybrat *One To One Interface*
- 7) Na výstupní *Frame* dopravníku přidat *Flow field* a propojit s výstupem *OneWay Path*

Posledním krokem bylo spojení *CreateSignal* dopravníku se signálem tlačítka pro generování palety, takže po stisku tlačítka se vygeneruje zvolená paleta. Více podrobností je v kap. 4.3.4 modelu Dopravníku. Pro vygenerování palety po přivedení signálu bylo třeba doplnit několik řádků v Python skriptu, viz Obr. 62. V Python skriptu bylo vytvořeno celkem cca 10 řádků kódu.


```

1  from vcScript import *
2  #import vcMatrix
3
4  comp = GetComponent()
5  creator = comp.findBehaviour('ComponentCreator')
6
7  def OnSignal(signal):
8      if signal.Name == 'CreateSignal' and signal.Value:
9          creator.create()
10
11 def OnRun():
12     creator.Part = 'vcid:c99a2a32-6ebd-425c-9a68-46e4e2bc3022'
13

```

Obr. 62 – Konfigurace Python skriptu generátoru

Palety se simulovanou fyzikou

Tak jako v případě Robotu byla aplikována *PhysicsEntity* ve variantě *In Physics*, tj. kompletní simulování fyziky. Palety tak mohou při neopatrné manipulaci spadnout na podložku, lze je položit do regálu a také zpátky na dopravník. *Collider* musel být ručně upraven, protože *Precise* nedokázal dostatečně přesně zkopírovat tvar palety.

Regál

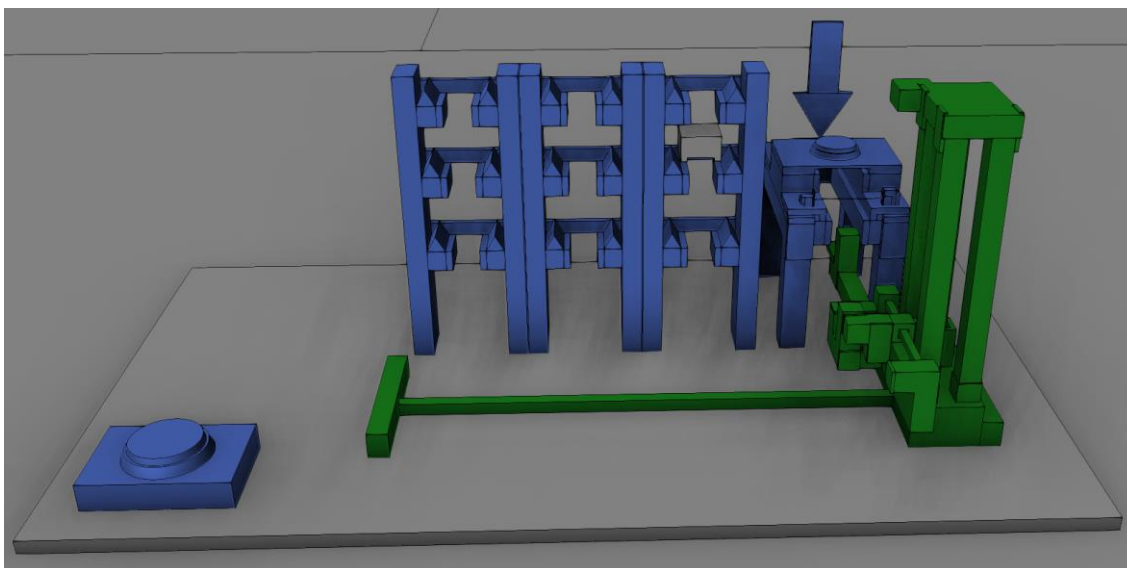
Collider geometrie regálu musel být také upraven, protože *Precise* nedokázal dostatečně přesně zkopírovat tvar. Navíc bylo nutné geometrii regálu rozdělit na dílčí jednodušší tvary, protože jinak úpravy nešly provést. Po úpravách lze již palety vkládat do regálu s dostatečnou opakovatelností.

HMI Button

Pro simulování chování skutečného CentralStopu byla provedena stejná inverze signálu jako v případě modelu Dopravníku. To samé pro tlačítko generování palety, kde proběhla změna z přepínače pamatujícího si poslední pozici na tlačítko, které se po stisku vrací do původní pozice, viz kap. 4.3.3.

4.5.4 Kolize kinematiky

Stejně jako v případě Robotu i zde může při simulaci dojít ke kolizím, a proto byl konfigurován detektor kolizí, který lze aktivovat dvěma způsoby, viz kap. 4.4.4. Na Obr. 63 je znázorněno výsledné rozdělení komponent do kolizních skupin:



Obr. 63 – Konfigurace detektoru kolizí Zakladače

4.5.5 Tvorba I/O pro propojení a řízení PLC

V Tab. 5 a Tab. 6 následuje přehled zvolených signálů pro vstup a výstup (I/O) pro PLC řízení modelu Zakladače:

Tab. 5 – Zvolené signály pro PLC vstupy (simulace → PLC)

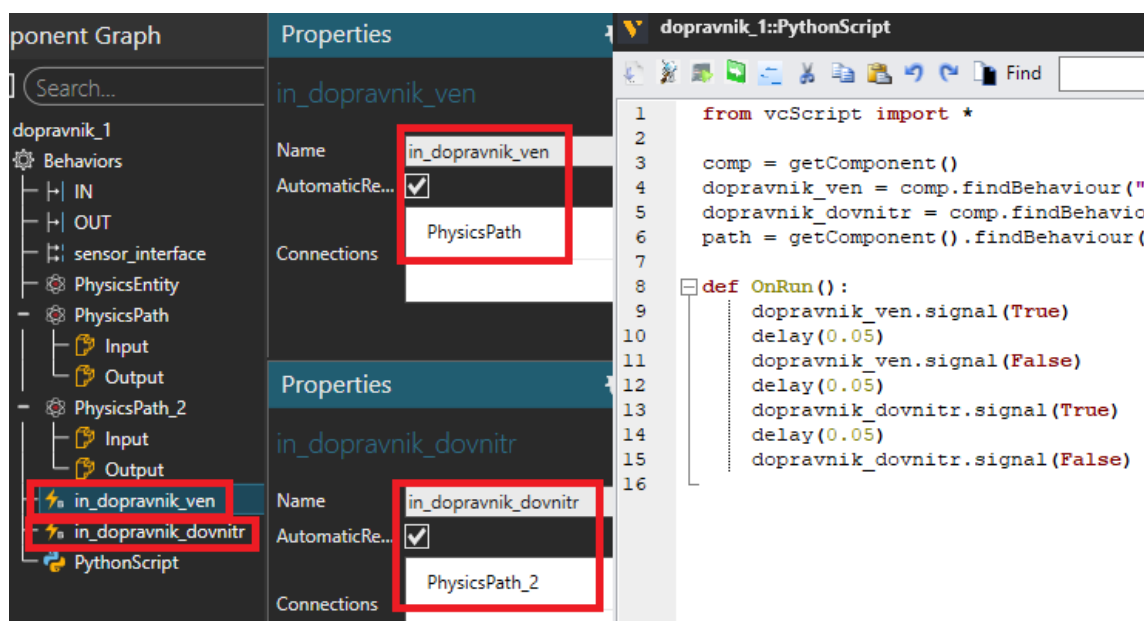
Datový typ	Název signálu VC	Název PLC proměnné
Real	out_ekoder_Z	High_Speed_Counter_2.MeasuredValue
Real	out_ekoder_Y	High_Speed_Counter_1.MeasuredValue
Bool	out_koncak_Z	di_reference_z
Bool	out_koncak_Y	di_reference_y
Bool	out_koncak_X_vysunuto	di_x_front
Bool	out_koncak_X_zasunuto	di_x_back
Bool	out_sensor_inside	di_light_barrier_inside
Bool	out_sensor_outside	di_light_barrier_outside
Bool	out_central_stop	di_total_stop

Tab. 6 – Zvolené signály pro PLC výstupy (PLC → simulace)

Datový typ	Název signálu VC	Název PLC proměnné
Bool	in_nahoru	do_z_minus
Bool	in_dolu	do_z_plus
Bool	in_zasunout	do_x_minus
Bool	in_vysunout	do_x_plus
Bool	in_doprava	do_y_plus
Bool	in_doleva	do_y_minus
Bool	in_dopravnik_ven	do_conveyor_outside
Bool	in_dopravnik_dovnitř	do_conveyor_inside

Signály tlačítka CentralStopu, tlačítka pro generování palety, společně se signály z optických závěr vychází z modelu Dopravníku (kap. 4.3.3). Hodnoty z enkodérů, signály z koncových snímačů a signály pohybu os vychází z modelu Robotu (kap. 4.5.3 a 4.5.5). S tím rozdílem, že u Zakladače jsou všechny osy translační a jedna z nich má dva koncové snímače pro obě krajní polohy nahrazující enkodér.

Signály pro pohyb dopravníku stačilo pro oba směry připojit k příslušným *PhysicsPath*, viz Obr. 64. A protože dopravník se automaticky spouští při zapnutí simulace, i když není přiveden True signál, bylo třeba doplnit několik řádků do Python skriptu pro reset obou směrů dopravníku při každém zapnutí simulace (Obr. 64). V Python skriptu bylo vytvořeno celkem cca 15 řádků kódu.



Obr. 64 – Konfigurace signálů dopravníku

Finální propojení signálů s proměnnými z OPC serveru proběhlo ve stejném duchu jako u modelu Dopravníku v kap. 4.3.4.

Takt komunikačních cyklů ve VC

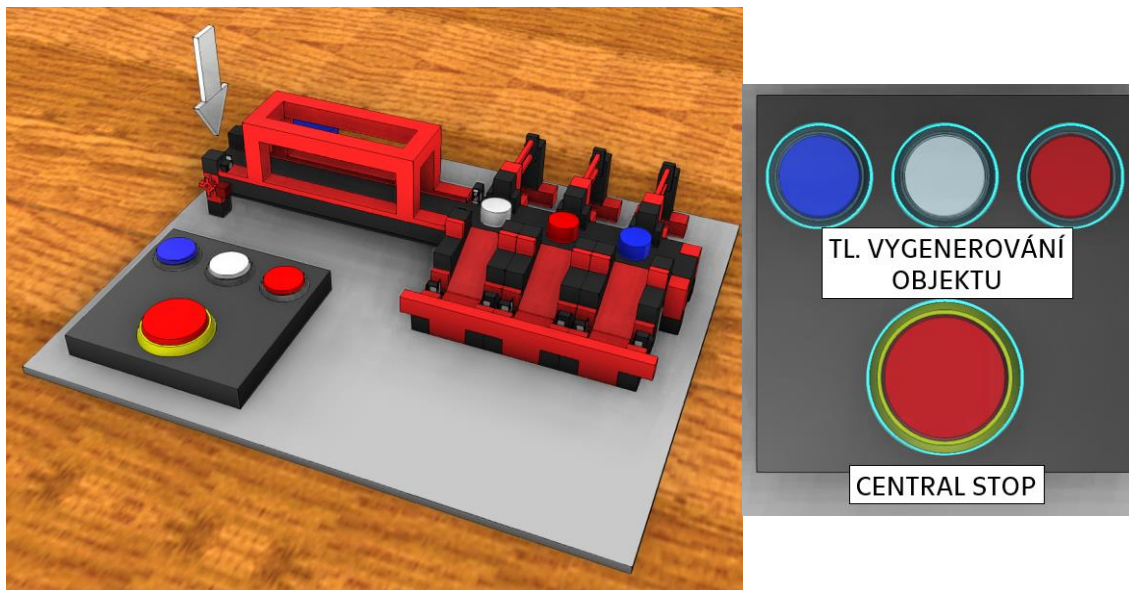
Takt komunikačního cyklu pro směr simulace → PLC byl zrychlen na 10 ms. Tato úprava byla provedena kvůli rychlejšímu odečítání hodnot enkodérů – jedná se o stěžejní prvek tohoto modelu z hlediska PLC řízení. Takt opačného směru komunikace PLC → simulace byl ponechán na výchozí hodnotě 50 ms. V dolní části rozhraní lze také při běhu simulace kontrolovat aktuální průměrný a maximální *update time*, viz kap. 4.3.4.

4.5.6 Dílčí shrnutí

Byl vytvořen model Zakladače, který je možné využít pro virtuální zprovoznění fyzické stavebnice FT. Také na něm byly v předchozích kapitolách vysvětleny důležité principy tvorby virtuálních modelů v prostředí VC. Součástí vytvořeného modelu je též připravený prázdný PLC projekt – pouze s HW konfigurací a s definovanými proměnnými viz Tab. 5 a Tab. 6. Tento PLC projekt je společně s virtuálním modelem součástí IX. přílohy této práce. V rámci práce byl také vytvořen detailní uživatelský manuál, který je součástí III. přílohy této práce.

4.6 Tvorba virtuálního modelu FT č.3: Třídíčka

Třetím z virtuálních modelů stavebnice FT je Třídíčka (Obr. 65). Jedná se o pásový dopravník s čidlem na rozpoznání barvy dílce a třemi vyhazovači, které třídí dílce do tří zásobníků. Dopravník i zásobníky jsou opatřeny optickými závory pro monitorování toku dílců Třídíčkou. Dopravník tentokrát disponuje generátorem pulzů, který funguje jako inkrementální enkodér. Vyhazovače jsou poháněny pneumaticky za použití kompresoru (modrý kvádr na Obr. 69), který je nutné také řídit skrze PLC. Výše zmíněné snímače a aktory jsou řízené PLC signály, které jsou přehledně uvedeny v kap. 4.6.4 v Tab. 7, resp. Tab. 8.



Obr. 65 – Virtuální model FT: Třídíčka (vlevo), ovládací tlačítka Třídíčky (vpravo)

4.6.1 Tvorba geometrie

Celková sestava byla vymodelována v CAD softwaru Autodesk Inventor a posléze importována v nativním formátu, jež VC podporuje. Jedná se z hlediska PLC řízení o funkčně identické dvojče se zjednodušenou geometrií. Byly použity i následující připravené prvky z knihovny:

- *HMI Button* – ovládací tlačítko
- *Conveyor Sensor* – optická brána
- *Boolean Signal Inverter* – prvek pro převrácení hodnoty Bool signálu

4.6.2 Návrh ovládacích prvků virtuálního modelu

Obdobně jako model Dopravníku (kap. 4.3.2) obsahuje i model Třídíčky aktivní tlačítka přímo na scéně připravené k interakci (Obr. 65):

- **CentralStop**
- **Tlačítka pro generování objektu** – po jejich stisku dojde k vygenerování příslušného objektu – barvy tlačítek odpovídají barvám generovaných dílců, uspořádání barev do francouzské trikolóry proběhlo nevědomě

4.6.3 Oživení geometrie

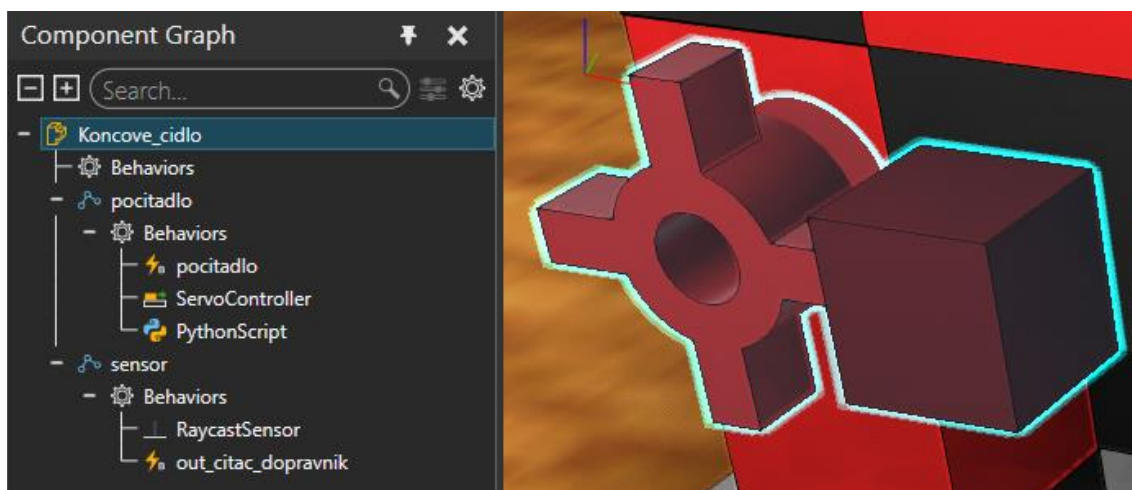
Dopravník s čítačem

Dopravník byl v tomto případě oživen úplně stejným způsobem jako v případě Zakladače, viz kap. 4.5.3, pouze s jinými rozměry, jinou rychlostí a jedním směrem pohybu.

Navíc byl přidán čítač, který obsahuje kolečko se čtyřmi výstupky po 90° a sensor (koncový snímač), který snímá průchod každého výstupku a vysílá do PLC signál krátkým pulzem. Cílem bylo co nejvíce napodobit stavebnici FT, kde je vše řešeno mechanicky přes koncový snímač.

Kolečko i snímač mají svůj vlastní *Link* (Obr. 66). Kolečko bylo oživeno jako rotační osa, která se začne točit ve směru pohybu dopravníku při přivedení True signálu, podobně jako u Robotu v kap. 4.4.3. V Python skriptu bylo vytvořeno celkem cca 30 řádků kódu. Signál je logicky spojen s pohybem dopravníku a rychlost otáčení byla odladěna tak, aby respektovala rychlost otáčení a pohybu stavebnice FT. Hodnota je nastavena na 500%/s, což odpovídá vysílání pulzů s frekvencí cca 5,6 Hz. Eventuelně by tak bylo možné tuto ryze mechanickou simulaci nahradit pouhým skriptem, který by vysílal signál s touto frekvencí.

Snímač byl oživen jako *Raycast sensor* snímající fyzické protočení výstupků kolečka.



Obr. 66 – Konfigurace čítače dopravníku

Vyhazovače s kompresorem

Vyhazovače jsou v podstatě jednočinné pneumatory konající translační pohyb. Při přivedení True signálu na pneumatorek je třeba aby se vysunul, ale na rozdíl od mechanické osy předchozích modelů, je žádoucí, aby se při změně signálu na False začal okamžitě opět zasouvat. K jeho ovládní tak stačí pouze jeden signál namísto dvou.

Vyhazovače byly opět extrahovány pomocí *Linků* a konfigurovány dle Obr. 67. Každý obsahuje:

- Signál pro aktivaci
- Signál simulující přívod z kompresoru
- Svůj *ServoController*
- *PhysicsEntity* – pro definování kinematické vazby při interakci s dílci
- Python skript – pro definici logiky pohybů

Zároveň je třeba zaručit, aby se vysunul pouze když je kompresor aktivní. Při ožívování se tak jednalo o variaci Python skriptu pro pohyb mechanických os a přísavky s kompresorem. V příslušném Python skriptu bylo upraveno několik řádků kódu na celkových cca 60 řádků. Zároveň není třeba žádných enkodérů ani koncových snímačů, protože stejně jako u stavebnice FT se pneumatory pohybují od dorazu k dorazu nastavením limitů zdvihu v *Linku*.



Obr. 67 – Konfigurace vyhozovačů

Generátor dílců

Jednou z hlavních komponent tohoto virtuálního modelu je generátor dílců, který má za úkol následující:

- Stejně jako v případě Zakladače schopnost připojit se k dopravníku a transportovat na něj vygenerovaný dílec
- Vygenerovat dílec s plně simulovanou fyzikou ve specifické barvě v závislosti na stisknutém tlačítku
- Pro každý vygenerovaný dílec zároveň náhodně vygenerovat pětimístné identifikační celé číslo z daného intervalu odpovídající stavebnici FT a číslo do každého dílce zapsat, aby jej bylo možné později přečíst na barevném sensoru

V podstatě byl použit základ z generátoru Zakladače, který byl rozšířen na celkové tři *Component Creatory* a tři korespondující signály pro připojení k tlačítkům ovládacího panelu.

Hlavní úpravy proběhly v samotném Python skriptu, který musel být kompletně přepracován. Generování náhodného čísla probíhá pomocí funkce *random* s Gaussovým rozložením pravděpodobnosti a vyloučením hodnot nacházejících se mimo požadovaný interval. Hodnota je pak z reálného čísla převedena na Integer a posléze na String. Proto, aby mohla být hodnota zapsána, se jako první musí vytvořit v generovaném dílci tzv. *Property* s datovým typem String pojmenovaná „*barva_kod*“. Poté je již možné barevný kód do dílce zapsat. Zároveň ale není možné používat vícero *Component Creatorů* s jedinou cestou *OneWay Path*, proto je nutné před

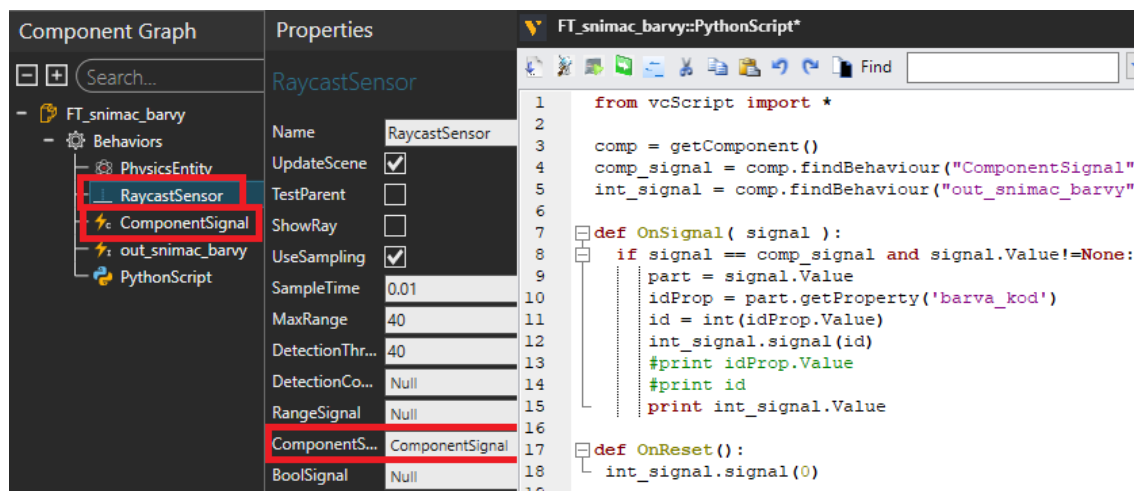
každým vygenerováním zajistit přiřazení cesty k požadovanému *Component Creatoru* na základě příchozího signálu dané barvy. V Python skriptu bylo vytvořeno celkem cca 70 řádků kódu.

Barevný sensor

Dalším krokem oživení bylo zprovoznění simulovaného barevného sensoru, který by dokázal číst barevné kódy dílců. Požadováno bylo:

- Dokázat přečíst barevný kód
- Převést barevný kód na datový typ Integer
- Přečtený a převedený barevný kód uložit do signálu typu Integer pro propojení s PLC

Ke čtení byl opět použit *Raycast sensor*, který kromě funkce binární optické závory dokáže i číst další informace o procházejícím objektu, mimo jiné i *Property*, do které je barevný kód zapsán. Zapotřebí je do *Raycast sensoru* přidat z *Behaviors* signál typu *Component* (viz Obr. 68), který funguje pouze interně ve VC – nelze ho např. propojit s externím PLC. Poté se s Python skriptem propojí *Component* signál a *Integer* signál a definuje se uložení přečtené *Property* „barva_kod“ do zavedené proměnné. Z proměnné se hodnota již může zapsat přímo do Integer signálu, který lze propojit s PLC. Tím je dosaženo maximální realističnosti při simulování stavby FT i z hlediska PLC řízení. V Python skriptu bylo vytvořeno celkem cca 20 řádků kódu.



Obr. 68 – Konfigurace barevného sensoru

Dílce se simulovanou fyzikou

Dílce, které vstupují do generátoru s již simulovanou fyzikou byly prakticky převzaty z modelu Robotu, viz kap. 4.4.3.

HMI Button

Pro simulování chování skutečného CentralStopu byla provedena stejná inverze signálu jako v případě modelu Dopravníku. To samé pro tlačítko generování dílce, kde proběhla změna z přepínače pamatujícího si poslední pozici na tlačítko, které se po stisku vrací do původní pozice, viz kap. 4.3.3. U dvou tlačítek byla změněna barva z modré na bílou, resp. červenou, aby odpovídala barvám generovaných dílců.

4.6.4 Tvorba I/O pro propojení a řízení PLC

V Tab. 7 a Tab. 8 následuje přehled zvolených signálů pro vstup a výstup (I/O) pro PLC řízení modelu Třídíčky:

Tab. 7 – Zvolené signály pro PLC vstupy (simulace → PLC)

Datový typ	Název signálu VC	Název PLC proměnné
Integer	out_snimac_barvy	ai_0_color_sensor
Bool	out_sensor_pred_komorou	di_0.0_cidlo_pred_komorou
Bool	out_sensor_za_komorou	di_0.1_cidlo_za_komorou
Bool	out_sensor_zasobnik_1	di_0.2_cidlo_zasobnik_1
Bool	out_sensor_zasobnik_2	di_0.3_cidlo_zasobnik_2
Bool	out_sensor_zasobnik_3	di_0.4_cidlo_zasobnik_3
Bool	out_citac_dopravnik	di_0.5_citac_dopavniku
Bool	out_central_stop	di_total_stop

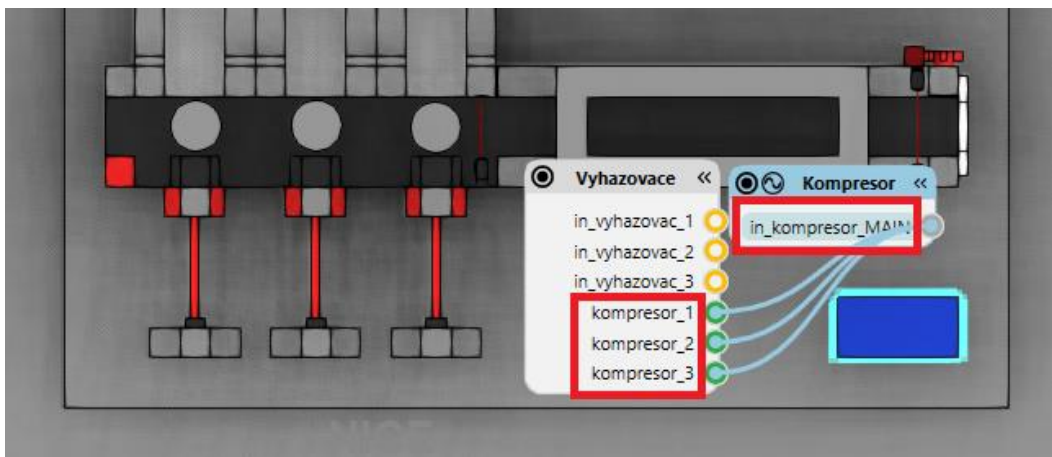
Tab. 8 – Zvolené signály pro PLC výstupy (PLC → simulace)

Datový typ	Název signálu VC	Název PLC proměnné
Bool	in_dopravnik	do_0.0_dopravnik
Bool	in_kompresor_MAIN	do_0.1_kompresor
Bool	in_vyhazovac_1	do_0.2_valec_1
Bool	in_vyhazovac_2	do_0.3_valec_2
Bool	in_vyhazovac_3	do_0.4_valec_3

Popis získání signálu typu Integer z barevného sensoru byl již popsán v předchozí kapitole 4.6.3, stejně tak jako signál typu Bool z čítače dopravníku.

Signály tlačítka CentralStopu, tlačítka pro generování dílce, společně se signály z optických závor vychází z modelu Dopravníku (kap. 4.3.3). Signál pro ovládání pohybu dopravníku byl přizpůsoben stejně jako v případě Zakladače (kap. 4.5.5).

Co se týká simulace pneumatického obvodu, je nutné v jeden moment dát signál aktivního kompresoru všem třem vyhazovačům. Proto byl na modrý kvádr, reprezentující kompresor stavebnice FT, umístěn signál pro „hlavní rozvaděč“ kompresoru a ten byl propojen s vlastním signálem kompresoru u každého vyhazovače, viz Obr. 69. V principu je tak možné řízením „hlavního rozvaděče“ řídit hodnotu zároveň u všech dílčích připojených komponent (vyhazovačů). Dílčí signály kompresorů jdou dále použít přímo v Python skriptu jednotlivých vyhazovačů.



Obr. 69 – Konfigurace signálu vyhazovačů

Finální propojení signálů s proměnnými z OPC serveru proběhlo ve stejném duchu jako u modelu Dopravníku v kap. 4.3.4.

Takt komunikačních cyklů ve VC

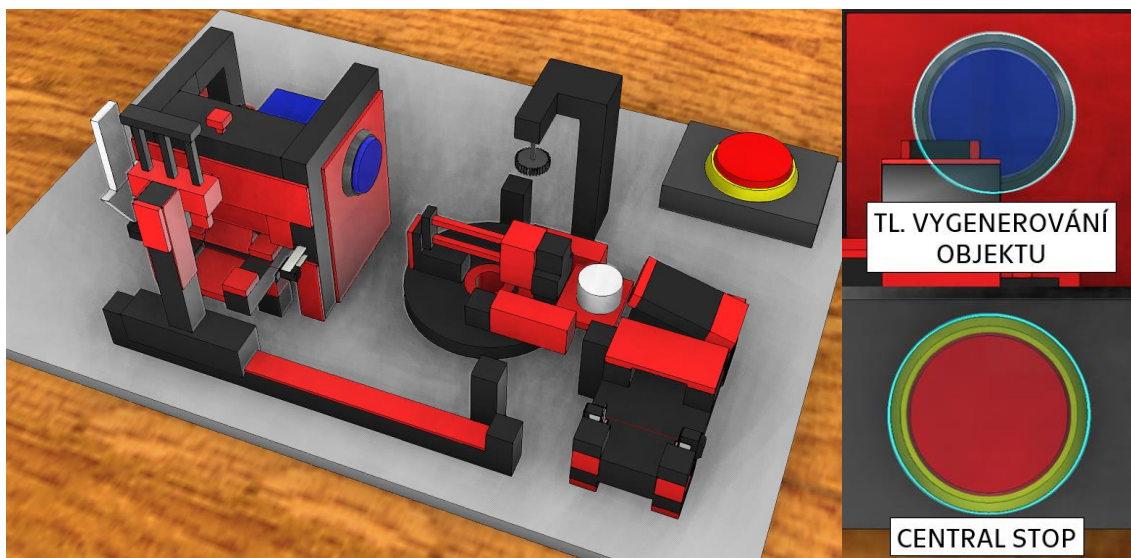
Takt komunikačního cyklu pro směr simulace → PLC byl zrychlen na 10 ms. Tato úprava byla provedena především z důvodu poměrně krátkých pulzů čítače dopravníku vysílaných rychle za sebou – odečítání pulzů je totiž stěžejním prvkem tohoto modelu z hlediska PLC řízení. Takt opačného směru komunikace PLC → simulace byl ponechán na výchozí hodnotě 50 ms. V dolní části rozhraní lze také při běhu simulace kontrolovat aktuální průměrný a maximální *update time*, viz kap. 4.3.4.

4.6.5 Dílčí shrnutí

Byl vytvořen model Třidičky, který je možné využít pro virtuální zprovoznění fyzické stavebnice FT. Také na něm byly v předchozích kapitolách vysvětleny důležité principy tvorby virtuálních modelů v prostředí VC. Součástí vytvořeného modelu je též připravený prázdný PLC projekt – pouze s HW konfigurací a s definovanými proměnnými viz Tab. 7 a Tab. 8. Tento PLC projekt je společně s virtuálním modelem součástí X. přílohy této práce. V rámci práce byl také vytvořen detailní uživatelský manuál, který je součástí IV. přílohy této práce.

4.7 Tvorba virtuálního modelu FT č.4: Multiprocesní stanice

Čtvrtým z virtuálních modelů stavebnice FT je Multiprocesní stanice, zkráceně Pec (Obr. 70). Jedná se o multiprocesní stanici kombinující pec, podavač pece, manipulátor s přísavkou, vyhazovač s otočným stolem, dopravník a další ovládané prvky. Polohovací osy manipulátoru (translace), podavače (translace) a otočného stolu (rotace) jsou řešeny mechanickým převodem s koncovými snímači (bez enkodérů). Vrata pece, úchop manipulátoru, přísavka a vyhazovač jsou řešeny pneumaticky za použití kompresoru (modrý kvádr na Obr. 71), který je nutné také řídit skrze PLC. Dopravník je jednosměrný. Dopravník a podavač mají optické závory pro určení přítomnosti obrobku. Dále je možné ovládat rotující kotouč (frézu) simulující frézovací stanici. Pec je opatřena světlem, které indikuje proces vypalování. Výše zmíněné snímače a aktory jsou řízené PLC signály, které jsou přehledně uvedeny v kap. 4.7.5 v Tab. 9, resp. Tab. 10.



Obr. 70 – Virtuální model FT: Multiprocesní stanice (vlevo), ovládací tlačítka Pece (vpravo)

4.7.1 Tvorba geometrie

Celková sestava byla vymodelována v CAD softwaru Autodesk Inventor a posléze importována v nativním formátu, jež VC podporuje. Jedná se z hlediska PLC řízení o funkčně identické dvojče se zjednodušenou geometrií. Byly použity i následující připravené prvky z knihovny:

- *HMI Button* – ovládací tlačítko
- *Conveyor Sensor* – optická brána
- *Boolean Signal Inverter* – prvek pro převrácení hodnoty Bool signálu
- *LED Indicator Light* – programovatelné světlo

4.7.2 Návrh ovládacích prvků virtuálního modelu

Obdobně jako model Dopravníku (kap. 4.3.2) obsahuje i model Pece aktivní tlačítka přímo na scéně připravené k interakci (Obr. 70):

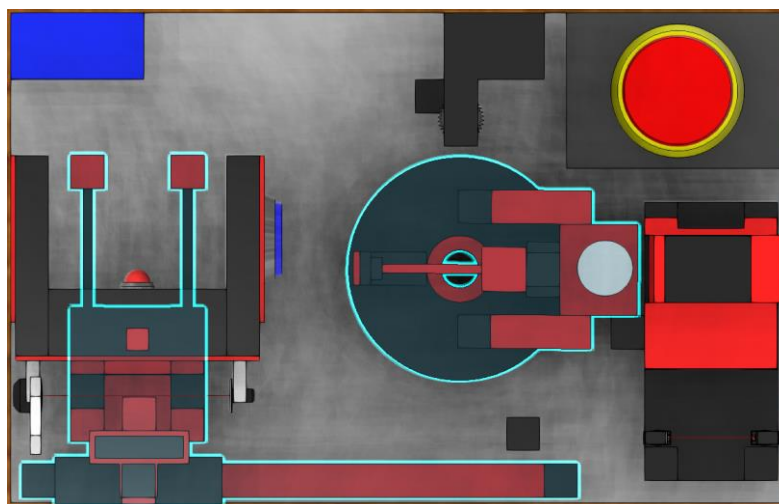
- **CentralStop**

- **Tlačítko pro generování objektu** – po jeho stisku dojde k vygenerování příslušného objektu

4.7.3 Oživení geometrie

Osy poháněné mechanickým převodem

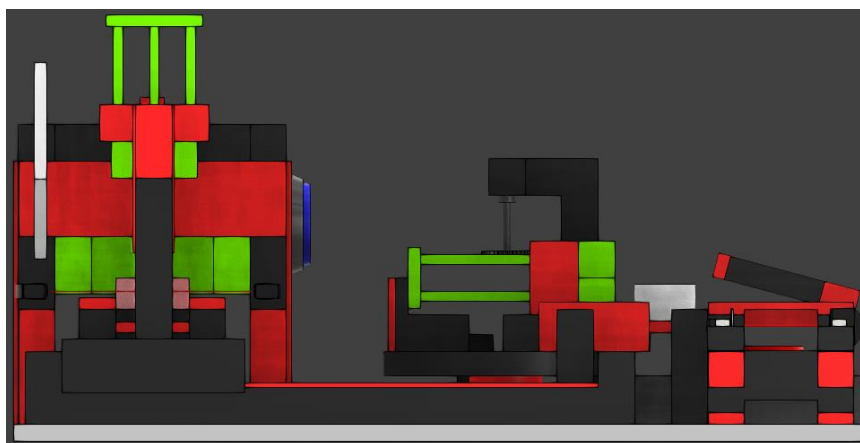
Při oživení lineární osy podavače pece, manipulátoru a rotační osy otočného stolu (modře zvýrazněno na Obr. 71) bylo postupováno analogicky jako v případě oživení os Robotu a Zakladače. Všechny tyto osy obsahují koncové snímače v obou koncových polohách. Otočný stůl navíc obsahuje třetí snímač v pozici, kdy se obrobek nachází pod frézou, aby bylo možné řídit polohu otočného stolu jako u stavebnice FT.



Obr. 71 – Mechanicky poháněné osy Pece

Pneumaticky poháněné osy s kompresorem

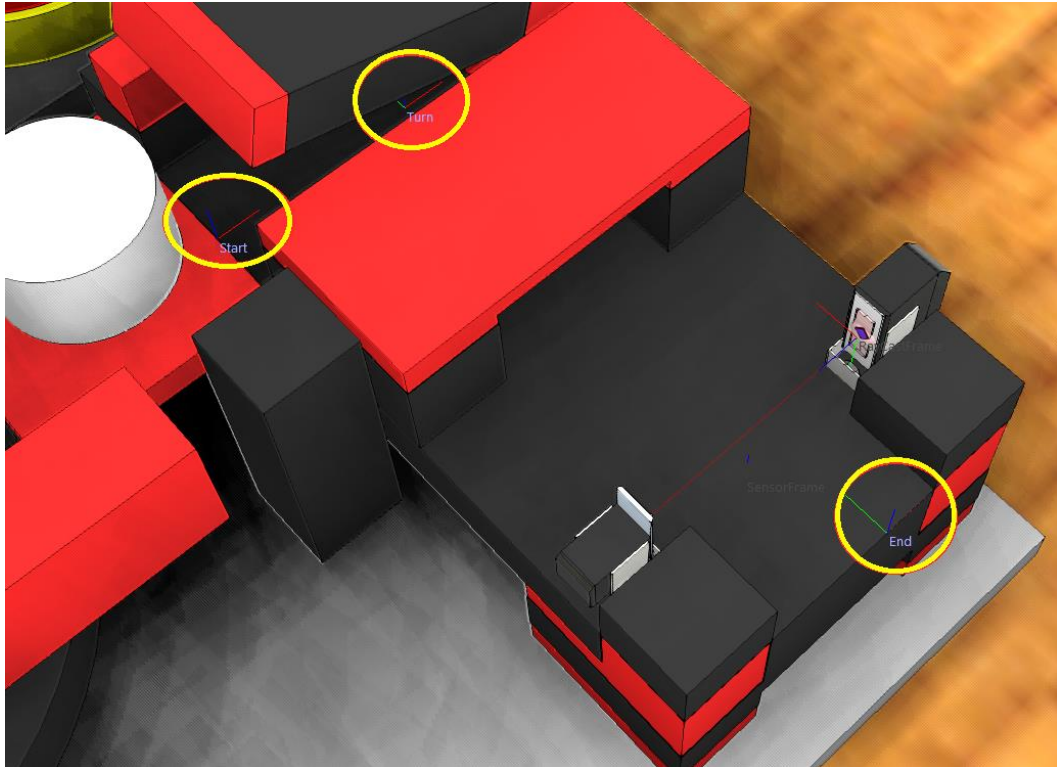
Při oživení vrat pece, uchopovače manipulátoru a vyhazovače (zeleně zvýrazněno na Obr. 72) bylo postupováno analogicky jako v případě oživení vyhazovačů Třídíčky. Všechny tyto osy disponují jedním signálem pro aktivaci a jedním signálem, který nahrazuje připojení ke kompresoru. Pro deaktivaci stačí změnit jednu z těchto hodnot na False. Kompresor je opět jako v případě Třídíčky vyřešen formou hlavního rozvaděče umístěného v modré kostce zjednodušeně reprezentující kompresor stavebnice FT.



Obr. 72 – Pneumaticky poháněné osy Pece

Dopravník

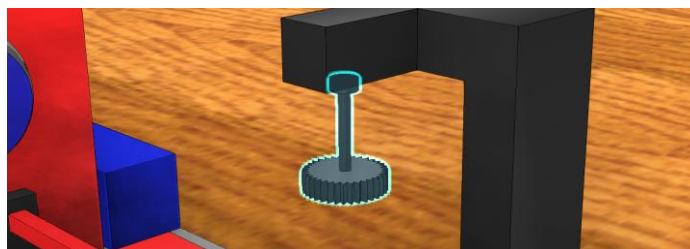
Při oživení dopravníku bylo postupováno analogicky jako v případě Trídíčky s tím rozdílem, že cesta není rovná přímka mezi dvěma *Framy*, nýbrž je tvořena třemi *Framy* umístěnými do tvaru „L“ (viz žlutě na Obr. 73). Dopravník je jednosměrný s jedním signálem pro spouštění pohybu a možnostmi nastavení rychlosti atd. jako u předchozích modelů.



Obr. 73 – Dopravník Pece

Frézovací stanice

Rotující kotouč (Obr. 74), který simuluje frézovací stanici, byl oživen obdobně jako rotační mechanická osa – disponuje však jen jedním signálem pro aktivaci pohybu, a tudíž rotuje jen jedním směrem po směru hodinových ručiček. V Python skriptu bylo vytvořeno celkem cca 30 řádků kódu.



Obr. 74 – Frézovací stanice

Generátor obrobků

Použit byl upravený generátor z modelu Zakladače, který byl rozšířen o *PhysicsPath* navazující na původní *OneWay Path*. Obrobek je totiž potřeba ze vzduchu upustit na podavač pece již se simulovanou fyzikou. Zároveň byl z důvodu nepatrně jiných rozměrů upraven offset *Framů* tvořících trajektorii putování obrobku.

Obrobky se simulovanou fyzikou

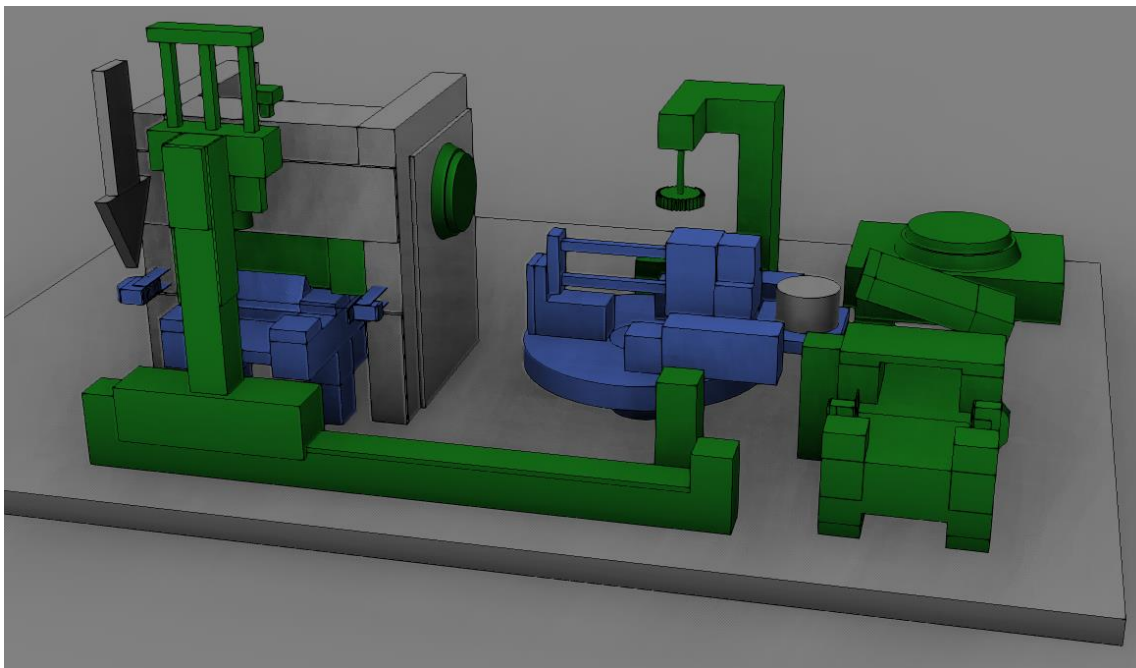
Obrobky vstupující do generátoru s již simulovanou fyzikou byly prakticky převzaty z modelu Robotu, viz kap. 4.4.3.

HMI Button

Pro simulování chování skutečného CentralStopu byla provedena stejná inverze signálu jako v případě modelu Dopravníku. To samé pro tlačítko generování obrobku, kde proběhla změna z přepínače pamatujícího si poslední pozici na tlačítko, které se po stisku vrací do původní pozice, viz kap. 4.3.3.

4.7.4 Kolize kinematiky

Stejně jako v případě Robotu i zde může při simulaci dojít ke kolizím, a proto byl konfigurován detektor kolizí, který lze aktivovat dvěma způsoby, viz kap. 4.4.4. Na Obr. 75 je znázorněno výsledné rozdělení komponent do kolizních skupin:



Obr. 75 – Konfigurace detektoru kolizí Pece

4.7.5 Tvorba I/O pro propojení a řízení PLC

V Tab. 9 a Tab. 10 následuje přehled zvolených signálů pro vstup a výstup (I/O) pro PLC řízení modelu Pece:

Tab. 9 – Zvolené signály pro PLC vstupy (simulace → PLC)

Datový typ	Název signálu VC	Název PLC proměnné
Bool	out_sensor_dopravnik	di_zavora_pas
Bool	out_sensor_pec	di_zavora_pec
Bool	out_koncak_manip_vpravo	di_koncak_uchop_ot_stul
Bool	out_koncak_manip_vlevo	di_koncak_uchop_pec
Bool	out_koncak_podavac_vysunuto	di_koncak_stul_p_ven
Bool	out_koncak_podavac_zasunuto	di_koncak_stul_p_dovnitř
Bool	out_koncak_rot_manip	di_koncak_ot_stul_uchop
Bool	out_koncak_rot_nastroj	di_ot_stul_frezka
Bool	out_koncak_rot_dopravnik	di_koncak_ot_stul_pas
Bool	out_central_stop	di_central_stop

Tab. 10 – Zvolené signály pro PLC výstupy (PLC → simulace)

Datový typ	Název signálu VC	Název PLC proměnné
Bool	in_dopravnik	do_pas
Bool	in_kompresor_MAIN	do_kompresor
Bool	in_nastroj_pohon	do_frezka
Bool	in_svetlo_pec	do_svetlo
Bool	in_rotace_plus	do_pohyb_ot_stul_pas
Bool	in_rotace_minus	do_pohyb_ot_stul_uchop
Bool	in_zasunout	do_pohyb_stul_p_dovnitř
Bool	in_vysunout	do_pohyb_stul_p_ven
Bool	in_doleva	do_pohyb_uchop_pec
Bool	in_doprava	do_pohyb_uchop_ot_stul
Bool	in_prisavka	do_podtlak
Bool	in_uchop	do_chnap
Bool	in_vyhodit	do_pist
Bool	in_vrata	do_vrata

Signál pro aktivaci frézky byl již objasněn v předchozí kap. 4.7.3.

Signály koncových snímačů byly konfigurovány analogicky jako u předchozích modelů, přičemž snímač otočného stolu v poloze u frézky byl nastaven jako aktivní pro pásmo 175° až 185° pro zajištění dostatečné reakční doby PLC po aktivaci snímače.

Signály tlačítka CentralStopu, tlačítka pro generování obrobku, společně se signály z optických závor vychází z modelu Dopravníku (kap. 4.3.3). Signály pro ovládání pohybu

dopravníku a pohybu os otočného stolu, manipulátoru a podavače byly konfigurovány analogicky jako v případě Zakladače (kap. 4.5.5).

Signál pro aktivaci kompresoru byl uzpůsoben analogicky jako v případě Třidičky (kap. 4.6.4). Tvoří tzv. hlavní rozvaděč, ke kterému se připojují dílčí signály zastupující pneumatický obvod zavedený k jednotlivým komponentám. Samotné signály pro aktivaci jednotlivých pneumaticky poháněných prvků byly vytvořeny obdobně jako v případě vyhazovačů u modelu Třidičky.

Finální propojení signálů s proměnnými z OPC serveru proběhlo ve stejném duchu jako u modelu Dopravníku v kap. 4.3.4.

Takt komunikačních cyklů ve VC

Takty komunikačních cyklů pro oba směry komunikace byly u modelu Dopravníku nastaveny na výchozích 50 ms. V dolní části rozhraní lze také při běhu simulace kontrolovat aktuální průměrný a maximální *update time*, viz kap. 4.3.4.

4.7.6 Dílčí shrnutí

Byl vytvořen model Pece, který je možné využít pro virtuální zprovoznění fyzické stavebnice FT. Také na něm byly v předchozích kapitolách vysvětleny důležité principy tvorby virtuálních modelů v prostředí VC. Součástí vytvořeného modelu je též připravený prázdný PLC projekt – pouze s HW konfigurací a s definovanými proměnnými viz Tab. 9 a Tab. 10. Tento PLC projekt je společně s virtuálním modelem součástí XI. přílohy této práce. V rámci práce byl také vytvořen detailní uživatelský manuál, který je součástí V. přílohy této práce.

5 Závěr

Tato práce se zabývala přípravou a zprovozněním virtuálních pracovišť pro výuku PLC řízení. V rešeršní části byl čtenář nejprve seznámen se stavem dané problematiky virtuálního zprovoznění. V průmyslu je hlavní výhodou virtuálního zprovoznění zkrácení celkového času realizace zařízení. Značnou část programování lze řešit již paralelně s výrobou a montáží zařízení. Ve výuce má virtuální zprovoznění tu výhodu, že se vše děje na jedné pracovní stanici (PC) a studentům to umožňuje pracovat i mimo hodiny výuky. Zároveň to studentům umožňuje se maximálně přiblížit fyzickému zprovoznění, aniž by se museli střídat u omezeného počtu fyzických zařízení. V současné době se pro účely virtuálního zprovoznění při výuce používá škála softwaru od Siemens a zájmem této práce bylo posunout výukové prostředky směrem k lepšímu grafickému podání, lepšímu vystižení reality a navýšení možností připojení z různých PLC od různých výrobců – a to i na dálku (on-line výuka). Zároveň zajistit, aby byl PLC program vytvořený studenty na propracovanějších virtuálních modelech ještě lépe přenositelný na fyzická zařízení.

Rešerše pokračovala stručnými přehledy aktuálních vizualizačních softwarů pro virtuální zprovoznění, představila jejich podporované typy komunikace, vlastností a případné výhody a nevýhody. Z těchto komerčně nabízených softwarů byli vybráni dva hlavní kandidáti, kteří splňovali nutné požadavky a nabízeli pro účely této práce (a potažmo budoucí výuky) ty nejlepší podmínky. Dále byly tyto dva softwary podrobněji popsány, a nakonec i porovnány v těch nejzásadnějších aspektech. Pro účely této práce vyšel jednoznačně lépe software Visual Components, který se i komerčně využívá v průmyslu.

Rešerše byla zakončena stručným úvodem do komunikačních protokolů se zaměřením na průmyslový Ethernet a komunikaci v reálném čase. Následoval přehled vybraných standardů průmyslového Ethernetu, které lze využít k fyzickému i virtuálnímu zprovoznění. Poté byl uveden stručný přehled struktury PLC programování v automatizačních softwarech od vybraných výrobců. Nakonec byly stanoveny výstupy rešerše důležité pro vlastní řešení.

V první části vlastního řešení byl proveden podrobný popis konfigurace propojení Visual Components s PLC použitím univerzálního komunikačního protokolu OPC UA, který vzešel z rešerše. Dále následovalo zpracování virtuálního modelu dopravníku, včetně PLC demo programu, který lze využít pro názorné ukázky základních principů PLC programování při výuce. Dále byly zpracovány čtyři virtuální modely, které z hlediska PLC řízení reprezentují čtyři fyzické stavebnice od firmy Fischertechnik používané již řadu let k výuce. Práce obsahuje potřebné informace z hlediska tvorby modelů, a i díky tomu bude případně možné modely snáze upravovat či rozšiřovat.

Prvním zpracovaným modelem byl tříosý (RTT) manipulační robot s přísavkou, pomocí kterého lze simulovat zakládání a odebírání obrobků do strojů – stroje jsou zde reprezentovány paletkami. Druhým zpracovaným modelem byl tříosý (TTT) regálový zakladač, pomocí kterého lze zakládat palety do regálu s devíti pozicemi – palety jsou k zakladači přepravovány dopravníkem. Třetím zpracovaným modelem byla dopravníková třídící linka, která za pomoci barevného sensoru třídí dílce podle barev do zásobníků. Čtvrtým zpracovaným modelem byla multiprocesní stanice s tepelným zpracováním, která simuluje tok obrobků mezi vypalovací pecí, frézou a dopravníkem.

Součástí všech virtuálních modelů je též detailní uživatelská dokumentace popisující všechny potřebné kroky k úspěšné obsluze konkrétního virtuálního modelu. Tyto uživatelské manuály jsou

součástí I. až V. elektronické přílohy této práce. CD přílohy VIII. až XI. obsahují archivy (zip) jejichž součástí jsou vedle návodů také samotné virtuální modely a prázdné PLC projekty s definovanými PLC proměnnými. Studenti tak dostanou hotový virtuální model zařízení propojený s PLC a mohou se věnovat jejich samotnému programování.

Zároveň byl jeden z modelů – model Robotu – zpracován také jako kompletní demo včetně PLC programu, na kterém je možno demonstrovat, jak by mohla vypadat jedna z úspěšných variant virtuálního zprovoznění. Toto PLC demo modelu Robotu se nachází společně s uživatelským manuálem k jeho obsluze v XIII. příloze této práce.

Hlavním výsledkem této práce jsou *state-of-the-art* virtuální modely výukových zařízení – digitální dvojčata programovatelných stavebnic Fischertechnik – vytvořené v softwaru Visual Components, který vzešel z rešerše. Tímto byl splněn hlavní cíl i všechny dílčí cíle práce.

Seznamy

Seznam použité literatury

- [1] Virtuální zprovoznění - budoucnost průmyslu. *Industry Fórum* [online]. 27.09.2018. Dostupné z: <https://www.industryforum.cz/virtualni-zprovozneni-meni-budoucnost-prumyslu>
- [2] KEIFER, J. *Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau* [online]. Saarbrücken, Germany, 2007. ISBN ISBN 978-3-930429-72-1. Dostupné také z: <https://d-nb.info/999933116/34>. PhD thesis. Universität des Saarlandes, Schriftenreihe Produktionstechnik.
- [3] DUMITRAȘCU, A., L. NAE a N. PREDINCEA. VIRTUAL COMMISSIONING AS A FINAL STEP IN DIGITAL VALIDATION OF THE ROBOTIC MANUFACTURING SYSTEMS. *Proceedings in Manufacturing Systems* [online]. 2014, 2014, **9**(4), 215-220. ISSN ISSN 2067-9238. Dostupné z: <https://www.semanticscholar.org/paper/VIRTUAL-COMMISSIONING-AS-A-FINAL-STEP-IN-DIGITAL-OF-Dumitra-Cu/8af74e9ac7825e7b311f4d26cfdead9189f8573a>
- [4] About. In: *Factory I/O* [online]. Dostupné z: <https://docs.factoryio.com/>
- [5] KALINA, Daniel. *Simulace výukových sestav strojů s využitím NX Mechatronics Concept Designer* [online]. Praha, 2018. Dostupné také z: <https://dspace.cvut.cz/handle/10467/80534>. Diplomová práce. České Vysoké Učení Technické v Praze.
- [6] Mechatronics Concept Designer. *Siemens* [online]. 2016. Dostupné z: https://www.plm.automation.siemens.com/media/global/cz/Siemens%20SW%20Mechatronics%20Concept%20Designer%20Fact%20Sheet_tcm84-87611.pdf
- [7] Tecnomatix® Plant Simulation: Plant Simulation & Throughput Optimization. *Siemens* [online]. Dostupné z: <https://www.plm.automation.siemens.com/global/en/products/manufacturing-planning/plant-simulation-throughput-optimization.html>
- [8] DELMIA Virtual Commissioning Expert: Datasheet. *3DExperience* [online]. 2019. Dostupné z: <https://www.hedraplm.com/media/1292/delmia-3dx-virtual-commissioning-expert-rcv-ds.pdf>
- [9] DELMIA Automation: Datasheet. *3DExperience* [online]. Dostupné z: https://www.tecmes.com.br/pdf/PDF_DELMIA_AUTOMATION_01.pdf
- [10] Digital Twins for Virtual Commissioning. *MapleSoft* [online]. Dostupné z: <https://www.maplesoft.com/solutions/engineering/AppAreas/Virtual-Commissioning.aspx>
- [11] Virtual Commissioning Controls Testing Emulation and Simulation. *Emulate3D* [online]. Dostupné z: <https://www.demo3d.com/>
- [12] MCGREGOR, Ian. INTRODUCTION TO EMULATE3D – EMULATION, SIMULATION, AND DEMONSTRATION. *Proceedings of the 2012 Winter Simulation*

- Conference* [online]. IEEE, 2012, (978-1-4673-4781-5/12), 10. Dostupné z: <https://www.semanticscholar.org/paper/INTRODUCTION-TO-EMULATE3D-%E2%80%93EMULATION%2C-SIMULATION%2C-Uhrmacher/16c00b4dacea34f7290e9440dc613fc6529779b>
- [13] Digital Factory Twin. *FastSuite* [online]. CENIT. Dostupné z: <https://www.fastsuite.com/solutions-products/digital-factory-twin>
- [14] Visual Components 4.3: The next generation of 3D manufacturing simulation technology. *Visual Components* [online]. Dostupné z: <https://www.visualcomponents.com/products/>
- [15] For Education: Go Digital with Simumatik. *Simumatik* [online]. Dostupné z: <https://www.simumatik.com/education/>
- [16] CIROS® Studio: Creating virtual learning environments. *Festo* [online]. 2018. Dostupné z: <https://www.festo-didactic.com/int-en/learning-systems/factory-automation-industry-4.0/focus-trending-topics-i4.0/digital-factory/factory-simulation/ciros-7.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC4xMTEwLjEwMzQxMw>
- [17] FE.Screen-Sim: planning, simulation, virtual commissioning. *F.EE* [online]. 2020. Dostupné z: https://www.fescreen-sim.de/fileadmin/fe-screen-sim/Broschueren_Flyer/F.EE_fescreen-sim_summary_data_sheet_2021_EN_01.pdf
- [18] RobotStudio®: Produktový list. *ABB* [online]. 2020. Dostupné z: <https://search.abb.com/library/Download.aspx?DocumentID=9AKK107991A4796&LanguageCode=cs&DocumentPartId=&Action=Launch>
- [19] PAUS, Philipp. TIA Portal: PLC Simulator Differences (PLCSIM Simulating Modules, PLCSIM and PLCSIM Advanced). In: *Hegamurl* [online]. YouTube, 2018, 19.3.2020. Dostupné z: <https://youtu.be/VH-tSfhQwms>
- [20] Počítačové sítě 5. přednáška standardizace počítačových sítí. In: *SlidePlayer* [online]. 2012. Dostupné z: <https://slideplayer.cz/slide/5246667/>
- [21] ZEŽULKA, František a Ondřej HYNČICA. Průmyslový Ethernet I: Historický úvod. *AUTOMA: Časopis pro automatizační techniku* [online]. **2007**(01), 41-43. Dostupné z: https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-i-historicky-uvod-2007_01_34298_2430/
- [22] *IEEE Standard for Ethernet*. 802.3-2018. New York: IEEE, 2018.
- [23] What Is the Difference Between Ethernet and Industrial Ethernet?. In: *Analog Devices* [online]. 2018. Dostupné z: <https://www.analog.com/en/technical-articles/what-is-the-difference-between-ethernet-and-industrial-ethernet.html#:~:text=What%20Is%20Industrial%20Ethernet%3F&text=With%20industrial%20Ethernet%2C%20data%20transmission,10%20Mbps%20to%201%20Gbps.&text=However%2C%20100%20Mbps%20is%20the,used%20in%20industrial%20Ethernet%20applications.>
- [24] ZEŽULKA, František a Ondřej HYNČICA. Průmyslový Ethernet II: Referenční model ISO/OSI. *AUTOMA: Časopis pro automatizační techniku* [online]. **2007**(03), 86-90 [cit.

- 2021-04-14]. Dostupné z: https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-ii-referencni-model-iso/osi-2007_03_34209_3890/
- [25] ZEZULKA, František a Ondřej HYNČICA. Průmyslový Ethernet IV: Principy průmyslového Ethernetu. *AUTOMA: Časopis pro automatizační techniku* [online]. **2007**(10), 57-60. Dostupné z: https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-iv-principy-prumysloveho-ethernetu-2007_10_34198_3258/
- [26] ZEZULKA, František a Ondřej HYNČICA. Průmyslový Ethernet IX: EtherNet/IP, EtherCAT. *AUTOMA: Časopis pro automatizační techniku* [online]. **2008**(10), 60-64. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/37910.pdf
- [27] Průmyslový Ethernet. *ElektroPrůmysl.cz: Informace ze světa průmyslu a elektrotechniky* [online]. 12. leden 2014. Dostupné z: <https://www.elektroprumysl.cz/automatizace/prumyslovy-ethernet>
- [28] ZEZULKA, František a Ondřej HYNČICA. Průmyslový Ethernet VII: Přehled současných standardů. *AUTOMA: Časopis pro automatizační techniku* [online]. **2008**(02), 26-29. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/36694.pdf
- [29] ZEZULKA, František a Ondřej HYNČICA. Průmyslový Ethernet VIII: Ethernet Powerlink, Profinet. *AUTOMA: Časopis pro automatizační techniku* [online]. **2008**(05), 62-66. Dostupné z: http://automa.cz/Aton/FileRepository/pdf_articles/37288.pdf
- [30] VOJÁČEK, Antonín. Průmyslová komunikace OPC UA - 1.díl - popis protokolu. *Automatizace.hw.cz: Rady a poslední novinky z oboru* [online]. 22. červenec 2020. Dostupné z: <https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html>
- [31] OPC UA vám zjednoduší život, ale... *Vše o průmyslu: Portál pro moderní výrobu* [online]. 9. květen 2019. Dostupné z: <https://www.vseoprumsly.cz/inspirace/firemni-novinky/opc-ua-vam-zjednodusi-zivot-ale.html>
- [32] OPC UA overview. In: *ST Wiki: STM32 MPU* [online]. 8 November 2021. Dostupné z: https://wiki.st.com/stm32mpu/wiki/OPC_UA_overview
- [33] BERGER, Hans. *Automating with SIMATIC S7-1500: Configuring, Programming and Testing with STEP 7 Professional*. Erlangen: Publicis Publishing, 2014. ISBN 9783895784040.
- [34] Programming Guideline for S7-1200/1500. In: *Siemens Industry Support* [online]. Siemens, 12/2018. Dostupné z: <https://support.industry.siemens.com/cs/ww/en/view/81318674>
- [35] Manuál jazyka ST pro PLC TECOmat. TECO Advanced Automation [online]. Kolín, listopad 2007. Dostupné z: https://www.TECOmat.cz/modules/DownloadManager/download.php?alias=txv00321_01_mosaic_progiec_cz
- [36] TwinCAT 3: Manual. In: *Beckhoff Information System* [online]. 2022-06-28. Dostupné z: https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_plc_intro/2526135947.html&id=6236151114262959556

Seznam tabulek

Tab. 1 – Zvolené signály pro PLC vstupy (simulace → PLC).....	47
Tab. 2 – Zvolené signály pro PLC výstupy (PLC → simulace).....	47
Tab. 3 – Zvolené signály pro PLC vstupy (simulace → PLC).....	57
Tab. 4 – Zvolené signály pro PLC výstupy (PLC → simulace).....	57
Tab. 5 – Zvolené signály pro PLC vstupy (simulace → PLC).....	67
Tab. 6 – Zvolené signály pro PLC výstupy (PLC → simulace).....	67
Tab. 7 – Zvolené signály pro PLC vstupy (simulace → PLC).....	73
Tab. 8 – Zvolené signály pro PLC výstupy (PLC → simulace).....	73
Tab. 9 – Zvolené signály pro PLC vstupy (simulace → PLC).....	79
Tab. 10 – Zvolené signály pro PLC výstupy (PLC → simulace).....	79

Seznam obrázků

Obr. 1 – Náklady na nápravu chyb a problémů v závislosti na časovém okamžiku ve výrobě (hrubé plánování, detailní plánování, zprovoznění a produkce) [2]	13
Obr. 2 – Příklad virtuální výrobní linky v prostředí Factory I/O [4]	14
Obr. 3 – Příklady komponent z knihovny prostředí Factory I/O [4].....	19
Obr. 4 – Uživatelské rozhraní prostředí Factory I/O [4]	19
Obr. 5 – Příklady komponent z eCatalogu prostředí Visual Components [14]	20
Obr. 6 – Uživatelské rozhraní prostředí Visual Components [14].....	21
Obr. 7 – Schéma vrstev architektury komunikačního kanálu – ISO/OSI [20]	24
Obr. 8 – Přenos zpráv mezi vrstvami modelu OSI [24]	24
Obr. 9 – Včasnost (vlevo) a současnost (vpravo) [25].....	25
Obr. 10 – Segmentování sítě Ethernet [25]	26
Obr. 11 – Architektury variant průmyslového Ethernetu (RT – Real-Time, reálný čas) [25].....	26
Obr. 12 – Průchod paketu sítě EtherCAT [26].....	27
Obr. 13 – Komunikační modely EtherCAT (vlevo) a EtherNet/IP (vpravo) [28]	27
Obr. 14 – Komunikační modely Ethernet Powerlink (vlevo) a Modbus/TCP (vpravo) [28]	28
Obr. 15 – Komunikační model Profinet IO (a) a Profinet IRT (b) [28]	29
Obr. 16 – Komunikační architektura OPC UA [32]	29
Obr. 17 – Struktura programování v Siemens TIA Portal [34]	30
Obr. 18 – Základní struktura POU [35].....	31
Obr. 19 – Stavebnice FT č.1: Robotická manipulace dílců (400x350 mm)	33
Obr. 20 – Stavebnice FT č.2: Regálový zakladač dílů (700x400 mm)	33
Obr. 21 – Stavebnice FT č.3: Multiprocesní stanice s tepelným zpracováním (500x350 mm)...33	33
Obr. 22 – Stavebnice FT č.4: Třídící linka s rozpoznáním barev (500x350 mm)	33
Obr. 23 – Blokové schéma propojení VC s PLC Siemens	34
Obr. 24 – Konfigurace rozhraní PG/PC	35
Obr. 25 – Nastavení pevné IP adresy virtuálního adaptéru	36
Obr. 26 – Nastavení projektu v TIA Portal	37
Obr. 27 – Nastavení PLC v TIA Portal	38
Obr. 28 – Nastavení I/O v TIA Portal.....	39
Obr. 29 – Spuštění virtuálního PLC v PLCSIM Advanced.....	40
Obr. 30 – Povolení záložky Connectivity	40
Obr. 31 – Vytvoření spojení VC a OPC UA serveru (Siemens).....	41
Obr. 32 – Blokové schéma propojení VC s PLC Beckhoff.....	42
Obr. 33 – Definování PLC proměnných pro OPC UA	42
Obr. 34 – Konfigurace PLC Beckhoff pro OPC UA	43
Obr. 35 – Inicializace OPC UA serveru	43

Obr. 36 – Vytvoření spojení VC a OPC UA serveru (Beckhoff)	44
Obr. 37 – Pomocný virtuální programovatelný model pro účely názorné výuky	45
Obr. 38 – Dvě ovládací tlačítka pracoviště	46
Obr. 39 – Nastavení podmínky pro generování objektu po přivedení True signálu	46
Obr. 40 – Konfigurace Boolean Signal Inverter	47
Obr. 41 – Zobrazení signálů scény	48
Obr. 42 – Propojení a zobrazení propojených proměnných mezi VC a PLC	48
Obr. 43 – Konfigurace taktu komunikačních cyklů	49
Obr. 44 – Virtuální model FT: Robot (vlevo), ovládací tlačítko Robotu (vpravo)	50
Obr. 45 – Vytvoření tzv. Linku	51
Obr. 46 – Připojení geometrie pomocí funkce Attach	52
Obr. 47 – Konfigurace os pomocí Behaviors	52
Obr. 48 – Konfigurace Collideru obrobků	53
Obr. 49 – Přidání balíčku Grasp Action Wizard	54
Obr. 50 – Konfigurace přísavky	55
Obr. 51 – Konfigurace PhysicsEntity	55
Obr. 52 – Aktivace detektoru kolizí	56
Obr. 53 – Konfigurace detektoru kolizí Robotu	56
Obr. 54 – Úprava skriptu pro použití přísavky a kompresoru	58
Obr. 55 – Virtuální model FT: Regálový zakladač (vlevo), ovládací tlačítka Zakladače (vpravo)	60
Obr. 56 – Připojení geometrie pomocí funkce Extract link	61
Obr. 57 – Konfigurace dopravníku pomocí Behaviors	62
Obr. 58 – Konfigurace PhysicsPath	63
Obr. 59 – Konfigurace One to One Interface	64
Obr. 60 – Konfigurace Component Creator	64
Obr. 61 – Konfigurace Generátoru palet	65
Obr. 62 – Konfigurace Python skriptu generátoru	66
Obr. 63 – Konfigurace detektoru kolizí Zakladače	67
Obr. 64 – Konfigurace signálů dopravníku	68
Obr. 65 – Virtuální model FT: Třidička (vlevo), ovládací tlačítka Třidičky (vpravo)	69
Obr. 66 – Konfigurace čítače dopravníku	70
Obr. 67 – Konfigurace vyhazovačů	71
Obr. 68 – Konfigurace barevného sensoru	72
Obr. 69 – Konfigurace signálu vyhazovačů	74
Obr. 70 – Virtuální model FT: Multiprocesní stanice (vlevo), ovládací tlačítka Pece (vpravo) ..	75
Obr. 71 – Mechanicky poháněné osy Pece	76
Obr. 72 – Pneumaticky poháněné osy Pece	76
Obr. 73 – Dopravník Pece	77
Obr. 74 – Frézovací stanice	77
Obr. 75 – Konfigurace detektoru kolizí Pece	78

Seznam použitého softwaru

- Autodesk Inventor Professional 2020
- Microsoft Office 365
- Visual Components Premium 4.4
- TIA Portal V15.1
- PLCSIM Advanced V2.0 SP1

- TwinCAT 3.1 eXtended Automation Engineering (XAE) 4024.10
- TF6100 TwinCAT 3 OPC UA Server 4.4.53.0
- TF6100 TwinCAT 3 OPC UA Configurator 4.4.73.0

Seznam příloh

Elektronické přílohy:

- I. Návod_SIEMENS_00_Dopravník.pdf
- II. Návod_SIEMENS_01_Robot.pdf
- III. Návod_SIEMENS_02_Zakladač.pdf
- IV. Návod_SIEMENS_03_Třídíčka.pdf
- V. Návod_SIEMENS_04_Pec.pdf
- VI. Návod_01_Robot_DEMO.pdf
- VII. DP1813_OS.pdf

CD přílohy:

Archivy v příloze obsahují kompletní dokumentaci, virtuální modely a připravené PLC projekty pro platformu Siemens a Beckhoff.

- VIII. 01_Robot.zip
- IX. 02_Zakladac.zip
- X. 03_Tridicka.zip
- XI. 04_Pec.zip
- XII. Dopravnik_PLC_demo.zip
- XIII. Robot_PLC_demo.zip
- XIV. DP1813_OS.pdf