

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení



Diplomová práce

Simulace výukových sestav strojů s využitím NX Mechatronics Concept Designer

2018

Daniel Kalina

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kalina** Jméno: **Daniel** Osobní číslo: **419943**
Fakulta/ústav: **Fakulta strojní**
Zadávající 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:

Simulace výukových sestav strojů s využitím NX Mechatronics Concept Designer

Název diplomové práce anglicky:

Machine Teaching Assembly Simulations Using the NX Mechatronic Concept Designer

Pokyny pro vypracování:

Popis tématu: Cílem diplomové práce je realizovat virtuální Software-in-the-loop (SIL) modely čtyř různých výukových sestav strojů, které bude možné využít pro návrh a testy řídicích programů na PLC Siemens Simatic S7-1500.; Osnova práce: - Zpracujte možnosti SIL simulace pro danou řadu PLC v NX Mechatronics Concept Designer nebo jiném vhodném prostředí. - Zvolte jednu výukovou sestavu a zpracujte PLC program pro její řízení. Pro zvolenou sestavu připravte virtuální model v NX Mechatronics Concept Designer a realizujte simulaci s PLC řízením. - Připravte virtuální modely pro zbývající tři výukové sestavy strojů. - Zpracujte popis postupu tvorby simulačních modelů a uživatelskou dokumentaci.; 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.; [2] Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic: dokument TXV 003 21.01 [online]. Desáté vydání. 2007 [cit. 2017-04-03]. Dostupné z: http://www.tecomat.m/wpimages/other/DOCS/cze/TXV00321_01_Mosaic_ProgIEC_cz.pdf

Jméno a pracoviště vedoucí(ho) diplomové práce:

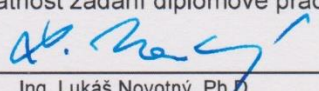
Ing. Lukáš Novotný, Ph.D., ústav výrobních strojů a zařízení FS

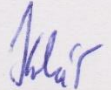
Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

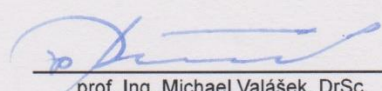
Datum zadání diplomové práce: **12.04.2018**

Termín odevzdání diplomové práce: **13.07.2018**

Platnost zadání diplomové práce: **30.09.2018**

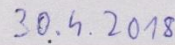

Ing. Lukáš Novotný, Ph.D.
podpis vedoucí(ho) práce

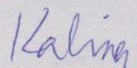

Ing. Petr Kolář, Ph.D.
podpis vedoucí(ho) ústavu/katedry


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 bakalářskou práci vypracoval samostatně a že jsem uvedl v přiloženém 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

.....
podpis

Poděkování

Tímto bych chtěl poděkovat Ing. Lukáši Novotnému, Ph.D. za odborné vedení a pomoc během vypracování celé práce. Dále bych chtěl poděkovat mé rodině za podporu během studií.

Anotace

Autor:	Daniel Kalina
Název DP:	Simulace výukových sestav strojů s využitím NX Mechatronics Concept Designer
Rozsah práce:	61 str., 49 obr., 5 tab.
Školní rok vyhotovení:	2018
Škola:	ČVUT – Fakulta strojní
Ústav:	Ú12135 – Ústav výrobních strojů a zařízení
Vedoucí diplomové práce:	Ing. Lukáš Novotný, Ph.D.
Zadavatel:	ČVUT – Fakulta strojní
Využití:	Programovatelné virtuální modely stavebnic Fischertechnik určené pro výuku předmětu PLC řízení výrobních strojů a zařízení
Klíčová slova:	PLC, automatizace, simulace, virtuální zprovoznění
Anotace:	Tato práce se zabývá virtuálním zprovozněním čtyř výukových stavebnic od firmy Fischertechnik pomocí metody Software in the Loop. Součástí práce je tvorba programovatelných virtuálních modelů zmíněných stavebnic a tvorba plného řídicího programu na jednu vybranou stavebnici.

Annotation

Author: Daniel Kalina

Title of masters dissertation: Simulation of teaching machine assemblies using NX Mechatronics Concept Designer

Extent: 61 p., 49 fig., 5 tab.

Academic year: 2018

University: CTU – Faculty of Mechanical Engineering

Department: Ú12135 – Department of Production Machines

Supervisor: Ing. Lukáš Novotný, Ph.D.

Submitter of Theme: CTU – Faculty of Mechanical Engineering

Application: Programmable virtual models of machine assemblies by Fischertechnik designated as teaching tool in subject of PLC control of production machines

Key words: PLC, automation, simulation, virtual commissioning

Annotation: This thesis is concerned with virtual commissioning of four machine assemblies by Fischertechnik using Software in the Loop method. The thesis contains creation of virtual models of said assemblies and solution of controlling program for one chosen assembly.

Obsah

1. Úvod.....	9
1.1. Cíle diplomové práce.....	9
2. Přehled problematiky virtuálního zprovoznění.....	10
2.1. Reálné zprovoznění	10
2.2. Virtuální zprovoznění	11
2.3. Metody virtuálního zprovoznění	13
2.3.1. Hardware in Loop.....	13
2.3.2. Software in Loop.....	13
2.4. Digitální dvojče	14
2.5. Programy pro tvorbu virtuálního modelu	16
2.5.1. NX Mechatronics concept designer.....	16
2.5.2. Tecnomatix Process Simulate.....	17
2.5.3. Software od ostatních výrobců	17
2.6. Virtuální PLC	18
2.7. Možnosti propojení virtuálního modelu s virtuálním PLC.....	19
2.7.1. OPC Server.....	19
2.7.2. Přímé propojení NX MCD s PLCSim Advanced.....	20
2.7.3. SIMIT	20
3. Virtuální zprovoznění stavebnic Fischertechnik.....	21
3.1. Tvorba virtuálního modelu	22
3.1.1. Funkční prvky automatizovaného skladu	23
3.1.2. Tvorba 3D modelu.....	25
3.1.3. Simulace pohonů, senzorů a enkodérů	26
3.1.4. Tvorba signálů pro virtuální PLC.....	28
3.2. Tvorba PLC programu	31
3.2.1. Zadáání pro tvorbu PLC.....	32



3.2.2.	Struktura řídicího programu	32
3.2.3.	Automatický režim	33
3.2.4.	Manuální režim.....	37
3.2.5.	Reference manipulátoru.....	37
3.2.6.	Ovládací panel.....	38
3.2.7.	Virtualizace PLC	41
3.3.	Propojení virtuálního modelu a PLC	43
3.3.1.	Testování simulace	45
3.4.	Tvorba virtuálních modelů ostatních stavebnic	48
3.4.1.	Třídící linka	48
3.4.2.	Robot	51
3.4.3.	Obráběcí pracoviště s pecí.....	53
4.	Závěr	56
	Seznamy	57
	Seznam použité literatury	57
	Seznam obrázků	58
	Seznam tabulek.....	60
	Seznam použitých zkratk.....	60
	Seznam příloh.....	60

1. Úvod

Tématem této práce je prostudování možností metody „Software in the loop“ a její následná aplikace na výukových sestavách automatizovaných stavebnic firmy Fischertechnik.

Při standardním zprovoznění automatizovaných strojů a zařízení nastává obvykle mnoho neočekávaných potíží, které prodlužují dobu uvedení do provozu a zvyšují náklady. Hlavními vlivy vzniku těchto potíží jsou neočekávané kolize a špatně navržený řídicí software. Důvodem chybného řídicího softwaru je skutečnost, že programátor PLC může program plně ověřit až po fyzickém zapojení technologie. Řešení nabízí simulační metoda „Software in the loop“. Jejím principem je vytvoření virtuálního dvojčete stroje a jeho řídicího PLC. Na vzniklém modelu může programátor PLC ověřovat řídicí program již při samotném procesu návrhu konstrukce.

V první části práce budou popsány metody virtuálního zprovoznění. Dále bude následovat popis programu Mechatronics concept designer a jeho možností pro potřeby simulace. Výstupem práce bude virtuální zprovoznění automatizovaných stavebnic firmy Fischertechnik.

1.1. Cíle diplomové práce

Cílem této práce je vytvořit programovatelné virtuální modely stavebnic pro výuku předmětu *PLC řízení strojů a zařízení*. K tomu bude zapotřebí:

- Tvorba virtuálních modelů stavebnic v programu Mechatronics Concept Designer
- Tvorba řídicího programu pro PLC jedné vybrané stavebnice
- Test řídicího programu na virtuálním modelu a reálné stavebnici
- Dokumentace tvorby modelů, řídicího programu a jejich propojení

2. Přehled problematiky virtuálního zprovoznění

V dnešní době se ve většině výrobních odvětvích používá automatizační technika. Automatizace procesů zvyšuje produktivitu, kvalitu a bezpečnost. Zároveň snižuje potřebný počet zaměstnanců. Nicméně se zvyšující se automatizací rostou požadavky na řízení těchto procesů. Tím roste čas testování a zprovoznění celého zařízení.

2.1. Reálné zprovoznění

Tradiční vývoj automatizovaného zařízení probíhá v několika oddělených procesech: návrh zařízení, návrh mechanické konstrukce, návrh elektrických rozvodů a návrh řídicího softwaru. Jednotlivé procesy na sebe postupně navazují a programátor má možnost otestovat navržený řídicí program až na hotovém zařízení.

Jak je vidět na (Obr. 1), zprovoznění zařízení zabere 15-20% času z celého projektu. Z toho až 90% času zprovoznění trvá rozjetí kontrolních systémů, například PLC. Ze samotného zprovoznění kontrolních a řídicích systémů trvá 70% času odstraňování nalezených chyb v řídicím softwaru. Tyto chyby mohou způsobit, kromě časového skluzu, mechanické poškození zprovozněného zařízení. Z těchto důvodů je tedy žádoucí, aby byl řídicí program testován v dřívějších fázích vývoje. [1]



Obr. 1: Podíl zprovoznění na délce projektu [1]

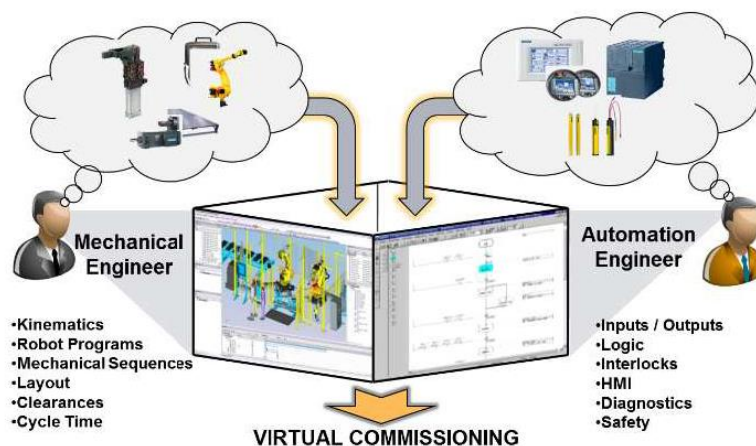
Dalším problémem tradičního zprovoznění je pozdní detekce problému, který vznikl v dřívějších fázích konstrukce zařízení. Dle Obr. 2 je zřejmé, že po zprovoznění jsou náklady na opravu detekovaných problémů značně nákladnější než ve fázi plánování. [1]



Obr. 2: Cena nápravy problému v závislosti na fázi projektu [1]

2.2. Virtuální zprovoznění

Výše zmíněné problémy řeší virtuální zprovoznění. Principem je tvorba digitálního dvojčete automatizovaného zařízení (viz Kapitola 2.4), které je následně propojeno s řídicím systémem, PLC. Programátor řídicího programu má pak k dispozici programovatelný model vytvořený konstruktérem, na kterém může testovat řídicí program a správnost zvolené konstrukce již v samotné fázi vývoje, a při samotném zprovoznění mít program odladěný a připravený k použití. Zamezí se tak ohrožení životů, mechanickému poškození a dalším prostojům vzniklými při tradičním zprovoznění. [2]



Obr. 3: Spolupráce konstruktéra a programátora na virtuálním modelu [2]

Projekt virtuálního zprovoznění zahrnuje tři hlavní faktory:

- 1) Mechatronický model zařízení včetně pohonů, snímačů a jiných funkcionalit
- 2) Řídící program včetně vstupních a výstupních signálů
- 3) Propojení mechatronického modelu s řídicím programem

Řídící program použitý pro virtuální zprovoznění by měl být totožný s programem pro reálné oživení. Mechatronický model by měl co nejdříve simulovat chování reálného zařízení. Z hlediska pohonů a snímačů by měl být model identický s reálným zařízením. Aby bylo možné dosažení takto přesného modelu, jsou zapotřebí následující data:

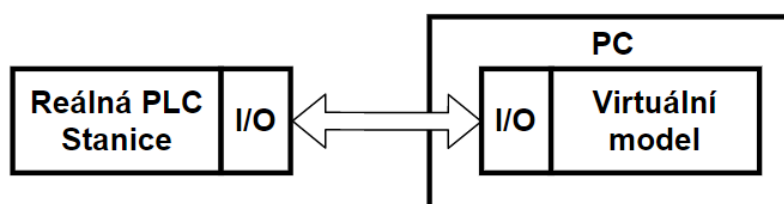
- 1) 3D modely všech jednotlivých součástí náležících do automatizovaného zařízení včetně jejich geometrií, kinematik a elektrických rozvodů
- 2) Rozložení automatizovaného zařízení včetně přesného umístění jednotlivých součástí
- 3) Trasa výrobku napříč zařízením včetně jednotlivých operací
- 4) Zmapování vstupních a výstupních signálů z jednotlivých součástí zařízení
- 5) Komunikační prostředí pro přenos dat mezi modelem a řídicím programem [3]

2.3. Metody virtuálního zprovoznění

V dnešní době jsou používány dvě metody virtuálního zprovoznění, Software in Loop, která je tématem této práce, a Hardware in Loop. Obě metody pracují s virtuálním modelem automatizovaného zařízení, na kterém je testován řídicí program. Stejně tak je výsledkem obou metod simulace v reálném čase. Virtuální model simuluje všechny procesy a generuje tak signály vstupů a výstupů, které jsou následně konvertovány do digitální a analogové podoby a přeneseny do řídicího programu.

2.3.1. Hardware in Loop

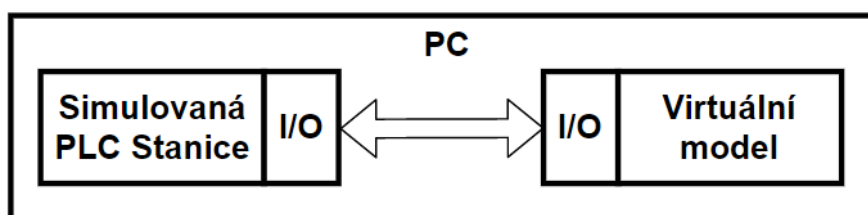
Jak je řečeno v úvodu kapitoly, automatizované zařízení je převedeno do virtuální podoby. K řízení tohoto modelu je následně použita reálná PLC stanice. Tato metoda dovoluje programátorovi ověřovat více komplexní řídicí programy a přibližuje se svým výsledkem realitě více, než metoda Software in Loop. Nevýhodami této metody je potřeba PLC stanice a přítomnost vstupů a výstupů na počítači s běžícím virtuálním modelem. [3]



Obr. 4: Schéma metody Hardware in Loop

2.3.2. Software in Loop

Princip této metody je založen na simulaci virtuálního zařízení i řídicí PLC stanice. Programátor má tak možnost na jednom počítači otestovat řídicí program bez potřeby dalšího hardwaru. Celý projekt lze jednoduše nahrát na jakýkoliv počítač s potřebným softwarem a výkonem. Z tohoto důvodu byla tato metoda vybrána pro virtuální zprovoznění stavebnic Fischertechnik, jelikož cílem této práce je tvorba programovatelných virtuálních modelů jednotlivých stavebnic bez potřeby dalšího hardwaru, jako jsou například PLC stanice nebo počítače s fyzickými vstupy a výstupy. [3]



Obr. 5: Schéma metody Software in Loop

2.4. Digitální dvojče

Jak již bylo zmíněno v kapitole 2.2, virtuální model reálného zařízení se často nazývá digitální dvojče. Nicméně je nutné si uvědomit, že tento pojem nezahrnuje pouze virtuální model. Koncept digitálního dvojčete byl představen již v roce 2003. V následujícím desetiletí zájem o toto téma prudce vzrostl.

Koncept digitálního dvojčete se skládá ze tří hlavních částí:

- 1) Fyzický model v reálném prostoru
- 2) Virtuální model ve virtuálním prostoru
- 3) Propojení dat a informací, které svazuje fyzický a virtuální model dohromady [4]

Firma Siemens dále definuje digitální dvojče těmito body:

- Digitální dvojče není pouze 3D model
- Digitální dvojče může kontrolovat reálnou věc, kterou reprezentuje
- Digitální dvojče má svojí identitu a kontext, podle kterého pracuje
- Digitální dvojče zahrnuje analýzy, predikce a řešení algoritmů [5]

Za deset let od představení tohoto konceptu vzrostlo množství a přesnost dat a informací, které mezi produkty lze přenášet. U virtuálního modelu lze simulovat chování za různých stavů a hodnotit tak jeho výkonnost. Lze tyto modely zjednodušit do bodu, kdy se shoduje pouze žádaná geometrie či jiné charakteristiky a drasticky tak zmenšit jejich velikost a zrychlit výpočetní rychlost simulací. Takto odlehčené modely dovolují simulovat složité komplexní systémy a jejich chování v reálném čase s přijatelným výpočetním výkonem. U fyzického modelu lze snímat různé hodnoty ze snímačů a tyto hodnoty následně implementovat do virtuálního modelu pro zpřesnění simulace. [4]

Digitální dvojče poskytuje tři důležité znalosti:

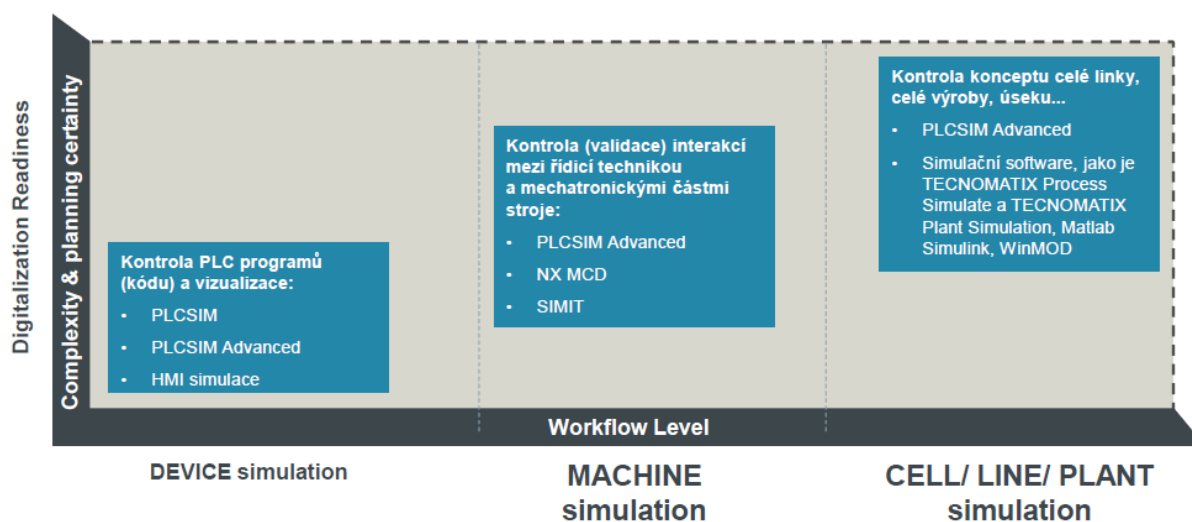
- 1) Konceptualizace
- 2) Porovnání
- 3) Kolaborace

Pod konceptualizací je myšlena možnost pozorovat pohyb výrobků či jiných předmětů v reálném zařízení na virtuálním modelu, který je propojen se zařízením v reálném čase. Místo tabulek s údaji o výrobku lze výrobek najít ve virtuálním modelu pomocí jeho čísla a pozorovat jeho stav, popřípadě trendové linie, které mohou indikovat blížící se problém. Pod porovnáním

je myšlen rozdíl mezi žádaným a reálným stavem. Ve virtuálním modelu obvykle vše probíhá naprosto ideálně, například přesnost polohování. Reálné zařízení se tak ale nechová. Proto lze zavést do virtuálního modelu určité kompenzace, které následně docílí co nejlepšího chodu reálného zařízení. Kolaborací je myšlen fakt, že virtuální model zařízení může využívat kdokoliiv kdekoliiv na světě. Zodpovědní zaměstnanci tak mohou sledovat stavy automatizovaných linek, aniž by u nich fyzicky byli, a detekovat závady na dálku. Případně opravy daných závad aplikovat na linky v jiných částech světa. [4]

2.5. Programy pro tvorbu virtuálního modelu

Od roku 2003, kdy byl koncept digitálního dvojčete představen, došlo k velkému vývoji softwarových programů, které dovolují tvorbu a ovládání virtuálních modelů reálných zařízení. Předním výrobcem simulačního softwaru je firma Siemens se svými produkty NX Mechatronics concept designer a Tecnomatix Process Simulate.



Obr. 6: Portfolio firmy Siemens pro simulační software [12]

2.5.1. NX Mechatronics concept designer

Mechatronics concept designer, dále jen MCD, je modul programu NX. Může tak naplno využívat CAD možnosti tohoto programu a vytvořené modely následně implementovat do tvorby simulace. Další výhodou součástí programu NX je možnost využití modulu pro tvorbu elektrických rozvodů. Při detailní tvorbě virtuálního modelu tak mohou spolupracovat konstruktéři, kteří tvoří vzhled, kinematiku a rozložení stroje, elektroinženýři, kteří volí rozmístění sensorů a pohonů, a programátoři, kteří určují chování stroje na základě řídicího kódu. Hotový virtuální model lze v MCD řídit pomocí časově závislých operací, které na sebe navazují, nebo pomocí řízení založeného na vzniklých událostech. První metoda se obvykle využívá v prvních fázích tvorby modelu, kdy je potřeba ověřit vhodnost zvolené kinematiky bez potřeby řídicího kódu. Druhá metoda je následně použita v dalších fázích, kdy již dochází i k verifikaci samotného řídicího kódu. MCD je svojí strukturou zaměřen na jednotlivé stroje z automatizovaných linek. Stavebnice Fischertechnik, které jsou předmětem této práce, právě tyto stroje představují a z toho důvodu byl MCD zvolen pro tvorbu virtuálních modelů. [6]

2.5.2. Tecnomatix Process Simulate

Podobně jako MCD, i program Process Simulate je součástí produktu Tecnomatix. Stejně jako MCD má integrovaný 3D modelovací nástroj. Hlavním rozdílem od MCD je jeho zaměření. Process Simulate se soustředí na automatizované linky jako celky. V rámci programu se tak nachází moduly jako Process Simulate Assembly, který cílí na optimalizaci montážních operací, Process Simulate Robotics, který slouží k programování a vizualizaci robotů, nebo také Process Simulate Human, který slouží k ověření ergonomických podmínek linky. [7]

2.5.3. Software od ostatních výrobců

Dalším vhodným programem pro virtuální zprovoznění je program Emulate 3D od stejnojmenné firmy. Mezi hlavními výhodami tohoto programu výrobce zmiňuje knihovnu QuickStart, ve které má uživatel možnost vybrat si z množství 3D modelů standardizovaného zařízení používaného v automatizaci jako jsou dopravníkové pásy, řadiče, palety a další. Další výhodou je integrovaný modul pro programování virtuálního PLC v jazyce kontaktních schémát nebo seznamu instrukcí. Jedná se o levnější variantu, která najde své uplatnění v menších aplikacích, kde nebude kladen takový důraz na složitost ovládání. [8]

Mezi další zástupce se řadí program Visual Components, který své uplatnění najde zejména v automobilovém průmyslu díky schopnosti simulace robotizovaného nanášení laku na dílce a měření nanesené vrstvy.

2.6. Virtuální PLC

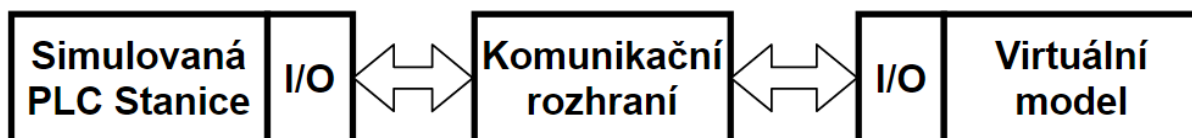
„PLC neboli Programmable Logic Controller je číslicově pracující elektronický systém konstruovaný pro použití v průmyslovém prostředí, využívající programovatelnou paměť pro interní ukládání uživatelsky orientovaných instrukcí pro provádění specifických funkcí (logických, sekvenčních, časovacích, čítacích, komunikačních a organizačních) za účelem řízení strojů či procesů, a to prostřednictvím digitálních nebo analogových vstupů a výstupů.“ [9]

Funkcí virtuálního PLC je simulace jeho reálného protějšku. Reálné PLC není zapotřebí. Programátor může průběžně do určité míry ověřovat funkčnost řídicího programu bez potřeby reálné sestavy.

K sestavení a naprogramování virtuálního PLC bude v této práci použit program TIA Portal V14. Jedná se o software od firmy Siemens. Označení TIA je zkratkou pro Totally Integrated Automation. Jak již název napovídá, tento program integruje kontroléry, HMI, řízení pohonů, snímače, a další, pod jedno programovací prostředí. K samotné simulaci bude použit program PLCSim Advanced, rovněž od firmy Siemens. Tento software je určen k simulaci PLC řady S7-1500, která je použita k řízení stavebnic Fischertechnik, které jsou předmětem této práce. Simulace bude probíhat na počítači, na kterém je zmíněný software instalován a žádný další hardware nebude zapotřebí. Kromě samotné simulace běhu programu lze měnit operační mód CPU mezi režimy RUN, PAUSE a STOP. V případě, že virtuální PLC není propojeno s řízenou jednotkou, lze měnit hodnoty vstupů a výstupů PLC manuálně v programu TIA Portal.

2.7. Možnosti propojení virtuálního modelu s virtuálním PLC

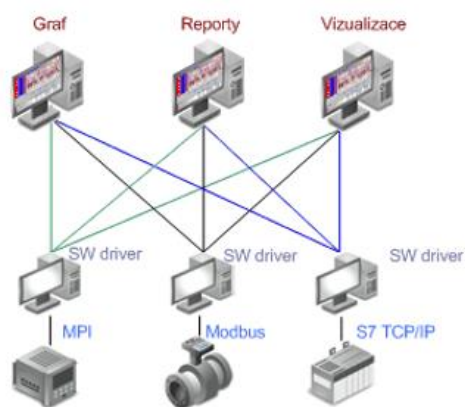
Aby mohl virtuální model přijímat a vysílat signály do virtuálního PLC, musí mezi nimi být zavedeno komunikační rozhraní (Obr. 7). V této kapitole budou popsány využívané možnosti jak toto rozhraní uskutečnit mezi zmíněným softwarem NX MCD a PLCSim Advanced od firmy Siemens.



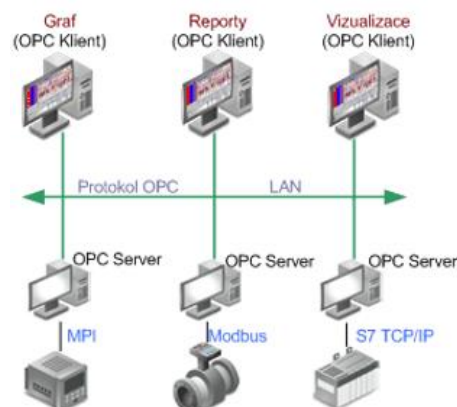
Obr. 7: Schéma propojení virtuálního PLC s modelem

2.7.1. OPC Server

OPC neboli Open Process Control je komunikační protokol, jehož cílem je vytvořit jednotné komunikační rozhraní mezi hardware a software produkty průmyslové automatizace. Dříve byl potřeba speciální ovladač pro každé hardware zařízení připojené k počítači. Častým problémem se stávalo vzájemné ovlivňování komunikace při instalaci více ovladačů. Docházelo také k nekompatibilitě s daným operačním systémem. [10]



Obr. 8: Propojení bez OPC serveru [10]



Obr. 9: Propojení s OPC serverem [10]

Použitím OPC serveru se jediným komunikačním rozhraním stane právě OPC. Jeho komunikační protokol je založen na architektuře Klient – Server. OPC klient je softwarový program. V případě této práce se bude jednat o program Siemens NX Mechatronics concept designer, který přijímá data z OPC serveru a převádí je do simulace v závislosti na použitém

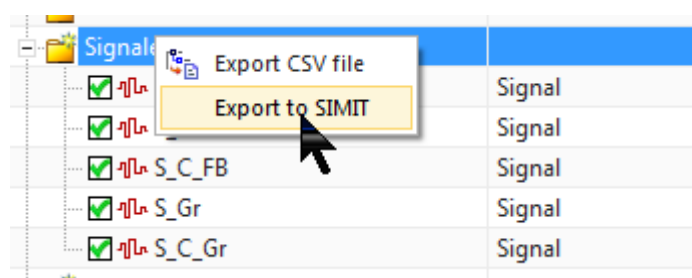
programu. OPC server je také softwarový program, který komunikuje s připojeným zařízením, v případě této práce se bude jednat o program PLCSim Advanced. Získaná data posílá OPC klientovi. Program NX MCD byl do své desáté verze schopen přijímat signály pouze z OPC DA serveru. Od jedenácté verze dokáže již přijímat signály i z OPC UA serveru. Zatímco OPC DA server je založen na technologii COM/DCOM od firmy Microsoft, a tedy funguje pouze na OS Windows, jeho novější verze OPC UA může fungovat i na jiných platformách. Může být například zabudován i do PLC automatu. [10]

2.7.2. Přímé propojení NX MCD s PLCSim Advanced

Na konci října roku 2017 byla uvolněna dvanáctá verze programu Siemens NX, která s sebou přinesla inovaci modulu MCD. V této verzi je možné propojit program NX MCD přímo s programem PLCSim Advanced, který byl popsán v předchozí kapitole.

2.7.3. SIMIT

SIMIT je program od firmy Siemens, který dovoluje simulovat automatizované technologie pro systémy a zařízení. Hlavní uplatnění tohoto programu se nachází ve virtuálním zprovoznění průmyslových procesů. Nicméně jedna z funkcí tohoto programu je také simulace virtuálního PLC. Program použitý pro tvorbu virtuálního modelu Siemens NX Mechatronics concept designer dovoluje exportovat vytvořené signály přímo do programu SIMIT, čímž se vytvoří požadované propojení. Nicméně na straně virtuálního PLC dovoluje program SIMIT pouze propojení se starší verzí programu PLCSim, která v této práci nebude použita.



Obr. 10: Export signálů z NX MCD do SIMIT

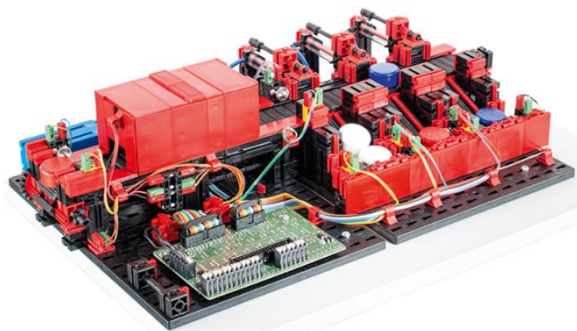
3. Virtuální zprovoznění stavebnic Fischertechnik

Následující kapitola se věnuje části vlastní tvorby práce. Jak již bylo řečeno v úvodu, cílem této práce je vytvořit programovatelné virtuální modely stavebnic pro výuku předmětu *PLC řízení strojů a zařízení*. Výstupem práce by tedy měly být čtyři zhotovené virtuální modely vytvořené na základě stavebnic od firmy Fischertechnik, které se dosud využívaly k výuce zmíněného předmětu. Tyto modely by měly mít zhotovené propojení s připravenými PLC programy, do kterých studenti předmětu budou následně tvořit své verze řídicího kódu. Pro jednu vybranou stavebnici by měl být vytvořen kompletní řídicí program. Pro splnění zadání práce bude zapotřebí:

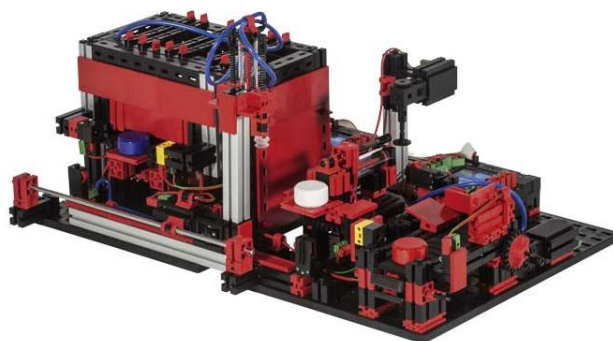
- Tvorba virtuálních modelů stavebnic v programu Mechatronics Concept Designer
- Tvorba řídicího programu pro PLC jedné vybrané stavebnice
- Test řídicího programu na virtuálním modelu a jedné vybrané stavebnici
- Dokumentace tvorby modelů, řídicího programu a jejich propojení

3.1. Tvorba virtuálního modelu

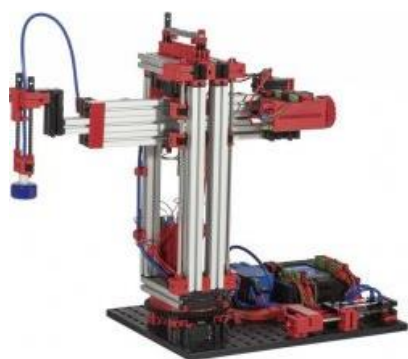
V rámci předmětu PLC řízení výrobních strojů a zařízení jsou k výuce programování PLC využívány čtyři stavebnice od firmy Fischertechnik. První stavebnicí je třídící linka, která za pomoci dopravníkového pásu, světelných závor a pneumatických válců třídí obrobky do třech stanišť (Obr. 11). Druhou stavebnicí je obráběcí stanoviště s pecí, které manipuluje s obrobky pomocí dopravníku a manipulátoru s přísavkou (Obr. 12). Třetí stavebnicí je robot s cylindrickou kinematickou strukturou, který manipuluje s obrobky pomocí přísavky (Obr. 13). Pro detailní popis virtuálního zprovoznění byla vybrána čtvrtá stavebnice s názvem Automated High-Bay Warehouse. Jedná se o model automatizovaného skladu, kdy je paleta s obrobkem po přistavení dopravníkem automaticky založena do pozice ve skladu pomocí manipulátoru (Obr. 14).



Obr. 11: Třídící linka



Obr. 12: Obráběcí stanoviště s pecí



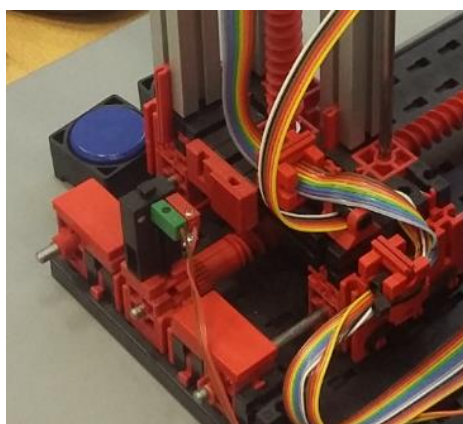
Obr. 13: Robot



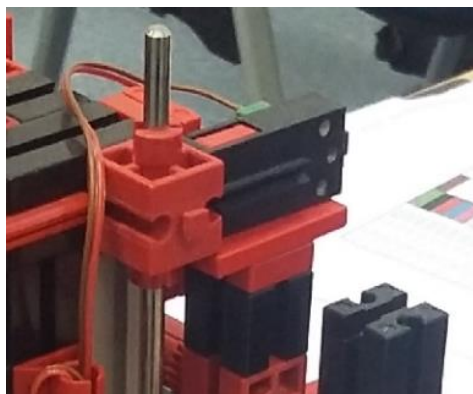
Obr. 14: Automatizovaný sklad [13]

3.1.1. Funkční prvky automatizovaného skladu

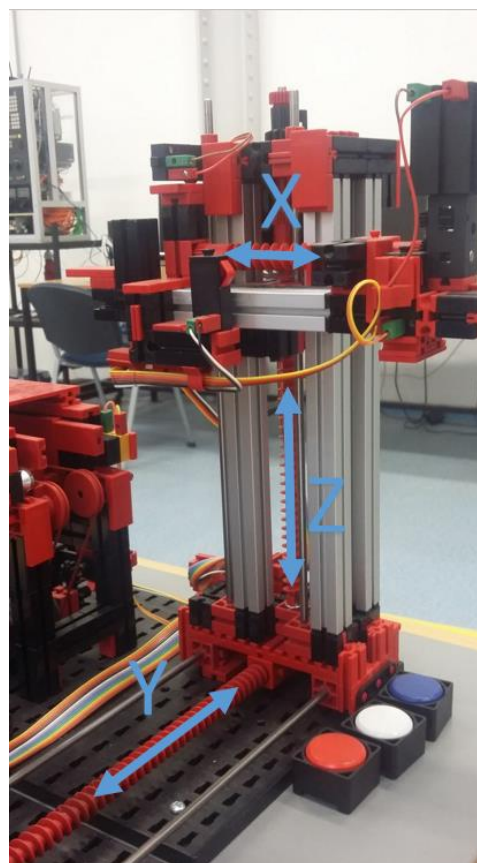
Hlavním funkčním prvkem je manipulátor sloužící k manipulaci s paletami mezi skladem a dopravníkem (Obr. 17). Manipulátor vykonává translační pohyb ve třech osách, které byly pojmenovány X, Y a Z. Osy Y a Z jsou poháněny pomocí stejnosměrných motorů s enkodéry, čímž je umožněno libovolné polohování těchto os. Regulace rychlosti motorů možná ovšem není. Dále je na obou osách jeden koncový snímač (Obr. 15 a Obr. 16), který obvykle slouží jako referenční snímač k vynulování enkodérů. Osa X slouží pouze k nabírání palety a proto není přesné odměřování polohy třeba. Tato osa je tedy bez enkodéru a je řízena pouze ve dvou koncových polohách pomocí koncových snímačů (Obr. 18).



Obr. 15: Koncový snímač na ose Y

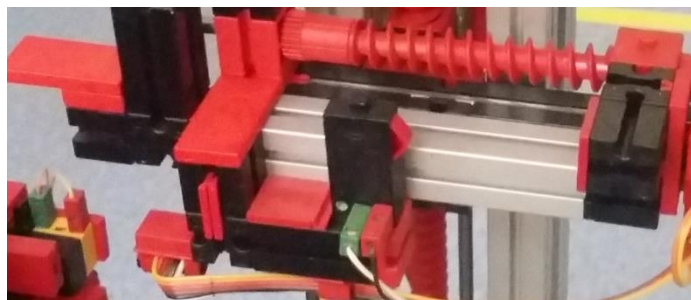


Obr. 16: Koncový snímač na ose Z

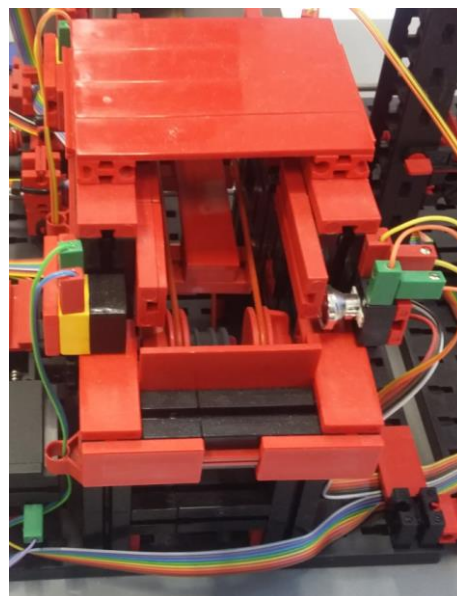


Obr. 17: Manipulátor

Druhým funkčním prvkem je dopravník. Ten slouží k přepravě palety z místa vložení obsluhou do místa, ve kterém je paleta nabrána manipulátorem. Dopravník je stejně jako osa X řízen dvěma snímači v koncových polohách nakládky a vykládky. V tomto případě se jedná o světelné závory (Obr. 19). Posledním funkčním blokem je samotný sklad, který plní pouze pasivní funkci. Nabízí devět stanovišť pro uložení palety s obrobkem.



Obr. 18: Koncové snímače na ose X



Obr. 19: Světelná závora na dopravníku

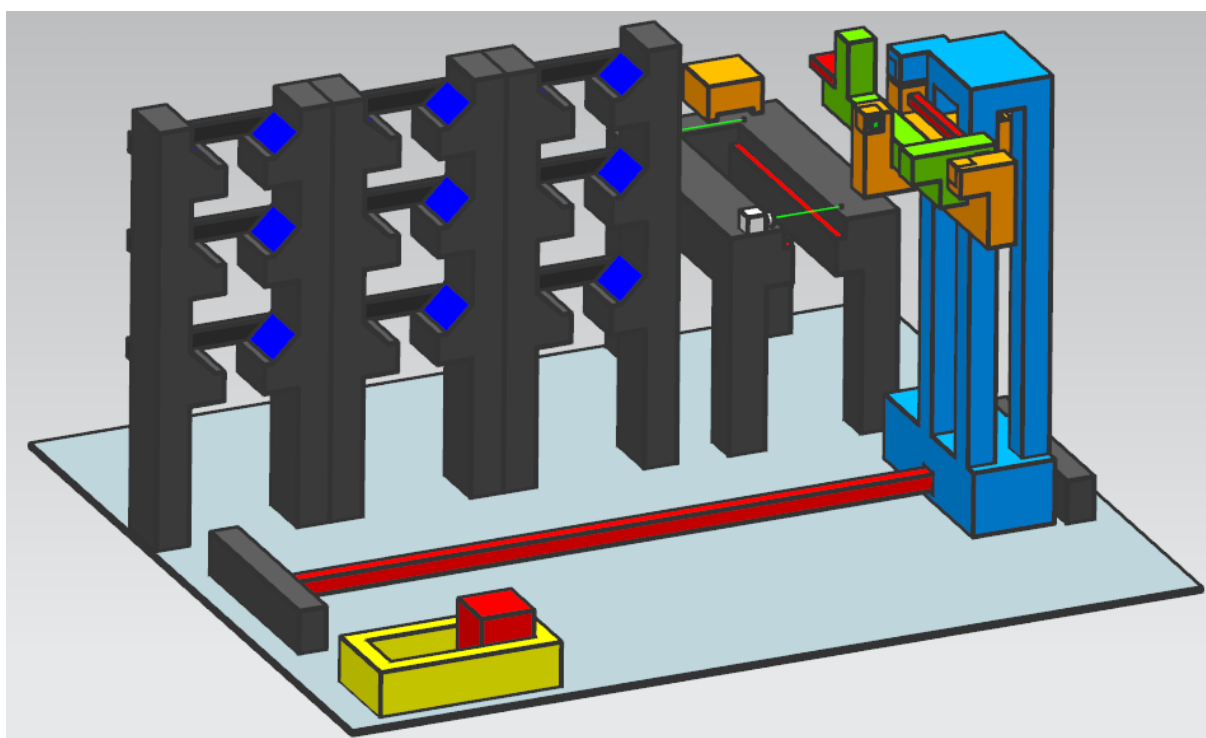
V následující tabulce (Tabulka 1) je souhrnný seznam všech vstupů a výstupů automatizovaného skladu.

Digitální vstupy a výstupy PLC	Popis	Název proměnné
do_0.0	Motor dopravníku, směr ven	do_conveyor_outside
do_0.1	Motor dopravníku, směr dovnitř	do_conveyor_inside
do_0.2	Motor osy Y, směr dozadu	do_y_minus
do_0.3	Motor osy Y, směr dopředu	do_y_plus
do_0.4	Motor osy Z, směr nahoru	do_z_plus
do_0.5	Motor osy Z, směr dolů	do_z_minus
do_0.6	Motor osy X, směr dopředu	do_x_plus
do_0.7	Motor osy X, směr dozadu	do_x_minus
di_0.0	Koncový snímač osy Y, směrem dozadu	di_reference_y
di_0.1	Světelná závora dopravníku, vevnitř	di_light_barrier_inside
di_0.2	Světelná závora dopravníku, venku	di_light_barrier_outside
di_0.3	Koncový snímač osy Z, směrem nahoru	di_reference_z
di_0.6	Koncový snímač osy X, vepředu	di_x_front
di_0.7	Koncový snímač osy X, vzadu	di_x_back
di_1.1	Tlačítko TOTAL STOP	di_total_stop

Tabulka 1: Souhrn vstupů a výstupů automatizovaného skladu

3.1.2. Tvorba 3D modelu

Tvorba modelu probíhala v CAD modulu programu NX 12.0 od firmy Siemens. Pomocí posuvného měřítka byly naměřeny všechny charakteristické kóty, aby byla zachována proporcionalita mezi modelem a stavebnicí. Jednotlivé funkční geometrie byly následně zjednodušeny z důvodu zrychlení běhu následných simulací. Důraz byl kladen na zachování rozměrů pohybových os manipulátoru a dopravníku a na rozmístění jednotlivých snímačů.



Obr. 20: 3D model automatizovaného skladu

Následně byl 3D model spuštěn v modulu MCD. Prvním krokem bylo přiřazení fyzických vlastností jednotlivým prvkům modelu. K tomu byla použita funkce *Rigid body*. Tato funkce přiřadí vybranému objektu vlastnosti tuhého tělesa. V dialogu funkce lze zvolit hmotnost objektu, umístění těžiště a momenty setrvačnosti v jednotlivých osách. Bylo zvoleno automatické přiřazení těchto hodnot, jelikož předmětem modelu je testování PLC programu a ne sledování dynamických vlastností. Výsledkem byla tabulka tuhých těles (Obr. 21), ke kterým bylo nutné přiřazení kolizních vlastností, které definují vzájemnou interakci jednotlivých těles. K tomu byla použita funkce *Collision body*. V dialogu této funkce je nutné vybrat všechny povrchy tělesa, kterým mají být přiřazeny kolizní vlastnosti. Dále je zde

možnost volby tvaru kolizní vrstvy. Byla vybrána možnost mesh, která nejméně kopíruje vybrané povrchy.

Basic Physics		
+ ✓	dopravník	Rigid Body
+ ✓	obrobek	Rigid Body
+ ✓	osa_x	Rigid Body
+ ✓	osa_y	Rigid Body
+ ✓	osa_y_spodek	Rigid Body
+ ✓	osa_z	Rigid Body
+ ✓	sklad	Rigid Body
+ ✓	total_stop	Rigid Body
+ ✓	total_stop_2	Rigid Body

Obr. 21: Tuhá tělesa v 3D modelu

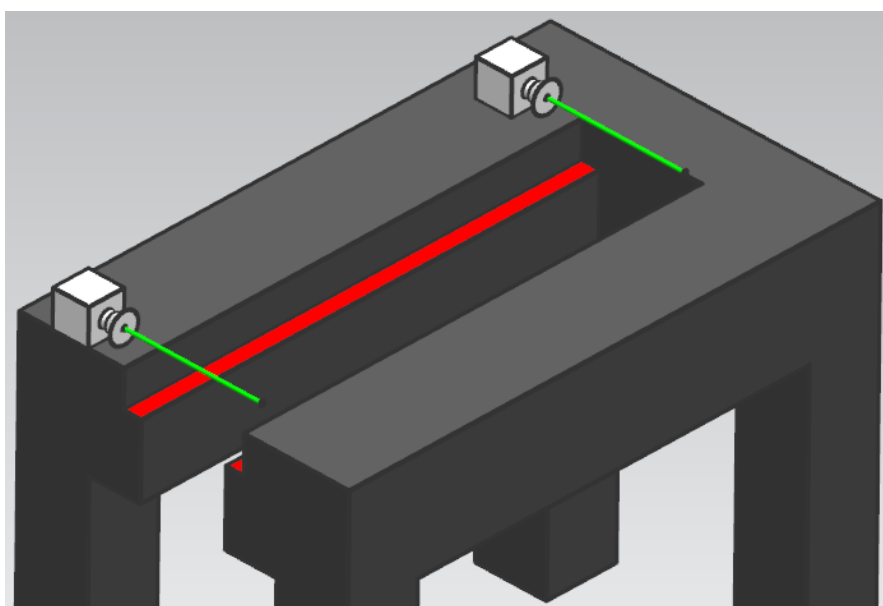
Dalším krokem byla definice pohybových vazeb jednotlivých prvků. Pro sklad, dopravník, spodní část osy Y a pro vodící drážku tlačítka total_stop byla použita funkce *Fixed joint*, která vybraný prvek pevně zavazbí na zvoleném místě. Pro pohyblivé prvky os X, Y a Z a pro tlačítko total_stop byla vybrána funkce *Sliding joint*. V dialogu této funkce je nutné vybrat jedno těleso, které má vykonávat posuvný pohyb, a druhé těleso, vůči kterému se první těleso bude pohybovat. Pro vložení modelu obrobku do probíhající simulace byla použita funkce *Object source*.

3.1.3. Simulace pohonů, senzorů a enkodérů

Simulace pohonů, senzorů a enkodérů probíhala v modulu MCD. Pro posuvný pohyb dopravníku byla použita funkce *Transport surface*, která z vybraného povrchu učiní povrch, který přesouvá objekty na něj položené ve zvoleném směru se zvolenou rychlostí. V tomto případě byla zvolena posuvová rychlost 0, jelikož samotný smysl a rychlost posuvu dopravníku budou řízeny funkcí *Speed control*. Byly vytvořeny dvě funkce *Speed control*, které byly propojeny s funkcí *Transport surface* definující dopravníkový pás. První funkce s názvem “dopravník_dovnitř” měla zadanou rychlost posuvu na 100 mm/s, druhá funkce s názvem “dopravník_ven” měla zadanou rychlost posuvu na -100 mm/s. Stejným způsobem byly vytvořeny funkce *Speed control* pro pohybové osy manipulátoru, které byly propojeny s funkcemi *Sliding joint* (viz. 3.1.2) pro jednotlivé osy. Pro ovládání osy X byly vytvořeny funkce s názvy “x_plus” a “x_minus” jejichž hodnoty rychlostí posuvů jsou -6,5 mm/s a 6,5 mm/s. Pro ovládání osy Y byly vytvořeny funkce s názvy “y_plus” a “y_minus” jejichž hodnoty

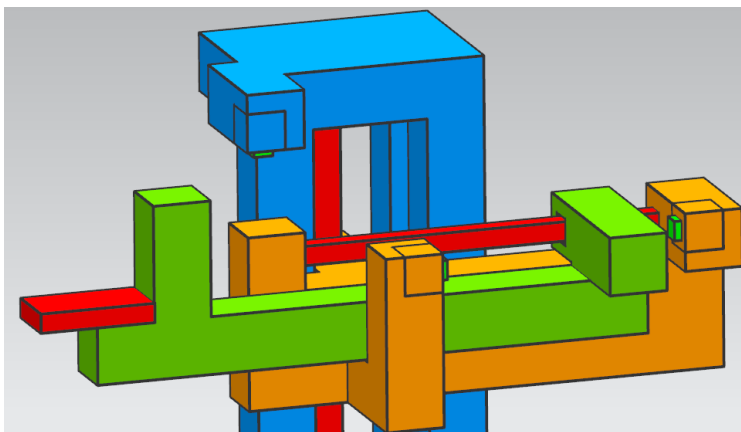
rychlostí posuvů jsou 16 mm/s a -16 mm/s. Pro ovládání osy Z byly vytvořeny funkce s názvy “z_plus“ a “z_minus“ jejichž hodnoty rychlostí posuvů jsou 32 mm/s a -32 mm/s.

Pro světelné závory na dopravníku byla vytvořena součást “svetelne_cidlo“, která byla dvakrát umístěna na dopravník (Obr. 22). K detekci protnutí zeleného paprsku jiným tělesem byla použita funkce *Collision sensor*. Byly tak vytvořeny kolizní senzory s názvy “senzor_ven“ a “senzor_dovnitř“, které při protnutí jiným tělesem změni hodnotu z FALSE na TRUE. Jelikož se kolizní senzory dotýkají dopravníku, bylo zapotřebí použití funkce *Prevent collision*, která mezi dvěma vybranými tělesy přeruší jakoukoliv kolizi.



Obr. 22: Umístění světelných závor na dopravníku

Stejně jako pro světelné závory, i pro koncové snímače na osách manipulátoru byla vytvořena součást “koncove_cidlo“. Ta byla zavazbena do všech pozic popsaných v kapitole 3.1.1 (viz. Obr. 23 a Obr. 24). Pro jednotlivé senzory byla rovněž použita funkce *Collision sensor*. Snímače na ose X nesou názvy “koncovy_snimac_x_plus“ a “koncovy_snimac_x_minus“. Snímače na osách Y a Z jsou pojmenovány “reference_y“ a “reference_z“. Jelikož jsou modely sensorů vnořené do modelu manipulátoru, bylo i zde zapotřebí použití funkce *Prevent collision*.

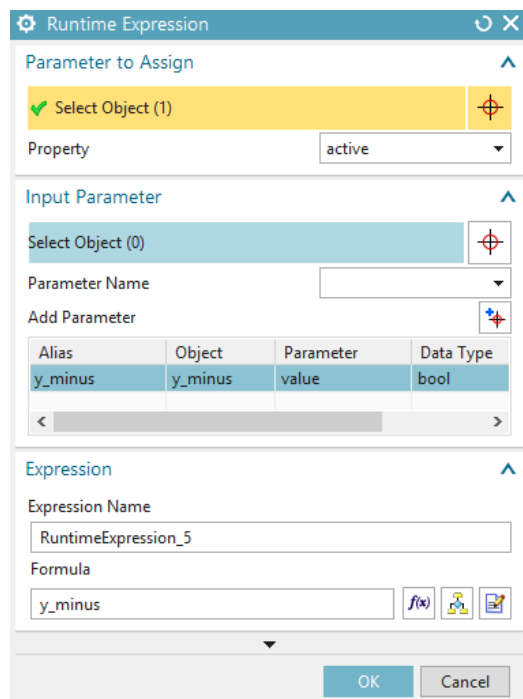


Obr. 24: Koncový snímač na ose Y

K simulaci enkodérů na osách manipulátoru Y a Z byla použita funkce *Position sensor*. Vytvořené funkce s názvy “encoder_y“ a “encoder_z“ byly propojeny s funkcemi *Sliding joint* příslušných os a měří hodnoty posuvu těles na těchto osách. Jelikož má být hodnota enkodérů nulová v případě sepnutí koncových snímačů, musí být osy Y a Z v počátku simulace v dotyku se zmíněnými snímači.

3.1.4. Tvorba signálů pro virtuální PLC

Tvorba signálů pro virtuální PLC probíhala rovněž v modulu MCD. Signály byly vytvořeny pomocí funkce *Signal adapter*. V dialogu této funkce se vygenerují požadované signály, přiřadí se jim datový typ a určí se, zda se jedná o vstup nebo výstup z MCD. Takto vytvořené signály je následně nutné propojit s příslušnými funkcemi řídicími funkční prvky skladu pomocí karty *Runtime Expression*, ve které lze definovat chování modelu při běhu simulace. V případě modelu automatizovaného skladu budou pro signály vybrány funkce vytvořené v kapitole 3.1.3. Signály propojené s funkcemi *Speed control* budou sloužit k ovládní pohonu os manipulátoru a dopravníkového pásu. Signály propojené s touto funkcí dokáží ovládat její aktivitu, hodnotu rychlosti, zrychlení a jerku, dále se zde dají simulovat stavy zatížení v osách X, Y a Z. Pro potřebu PLC programu byla vybrána možnost ovládní aktivity vybrané funkce. Pro manipulátor byla vytvořena funkce *Signal adapter* s názvem “manipulator“ a v ní vytvořeny signály pro jednotlivé osy. Pro dopravník byla rovněž vytvořena tato funkce s názvem “dopravník“ obsahující signály pro ovládní pohybu pásu. Na obrázku (Obr. 25) je zobrazeno propojení signálu “y_minus“, který byl vybrán jako *Parameter to Assign* s funkcí *Speed control* “y_minus“, která byla vybrána jako *Input parameter*. Do řádku *Formula* lze zapsat základní logické operace typu AND, OR, NOT. Zápis typu y_minus značí, že funkce *Speed control* “y_minus“ bude aktivní pouze v případě, kdy signál “y_minus“ bude nabývat hodnoty TRUE.























Obr. 25: Runtime expression pro signal "y_minus"

Signály propojené s funkcemi *Collision sensor* budou do PLC vysílat informaci o spuštění koncových snímačů na osách manipulátoru, světelných závor na dopravníku, nebo stlačení tlačítka TOTAL STOP. Signály propojené s funkcemi *Position sensor* budou do PLC vysílat číselnou hodnotu posuvu os Y a Z manipulátoru. Pro signály snímačů a enkodérů byla vytvořena funkce *Signal adapter* s názvem "senzory". U signálů světelných bran a tlačítka TOTAL_STOP byla v *Runtime expression* vytvořena negace pomocí záporného znaménka v zápisu řádku *Formula*, jelikož je tato logika i v předlohové stavebnici. Aby výsledná hodnota signálů z enkodérů odpovídala realitě, bylo zapotřebí hodnotu naměřenou v modelu, která odpovídá velikosti posuvu, přepočíst v řádku *Formula*. K přepočtu bylo zapotřebí přeměření stoupání pohybového šroubu os Y a Z a zjištění kolik reálný enkodér vyše impulzů za jednu otáčku. Výsledná rovnice v řádku *Formula* byla ve tvaru (1), kde hodnota 4,4 značí stoupání pohybového šroubu v milimetrech a 127,8 počet vyslaných pulzů za otáčku reálným enkodérem. Stejný přepočet byl použit i pro signal "encoder_z". [11]

$$\text{encoder_ysignál} = \frac{\text{encoder_ySenzor} * 1000 * 127,8}{4,4} \quad (1)$$

Všechny vytvořené signály jsou datového typu Bool, kromě signálů pro enkodéry, které jsou datového typu DInt (Obr. 26).

Signals	
<input checked="" type="checkbox"/>  dopravnik	Signal Adapter
<input checked="" type="checkbox"/>  dopravnik_dovnitř	Signal
<input checked="" type="checkbox"/>  dopravnik_ven	Signal
<input checked="" type="checkbox"/>  manipulator	Signal Adapter
<input checked="" type="checkbox"/>  x_minus	Signal
<input checked="" type="checkbox"/>  x_plus	Signal
<input checked="" type="checkbox"/>  y_minus	Signal
<input checked="" type="checkbox"/>  y_plus	Signal
<input checked="" type="checkbox"/>  z_minus	Signal
<input checked="" type="checkbox"/>  z_plus	Signal
<input checked="" type="checkbox"/>  senzory	Signal Adapter
<input checked="" type="checkbox"/>  encoder_y	Signal
<input checked="" type="checkbox"/>  encoder_z	Signal
<input checked="" type="checkbox"/>  koncak_x_minus	Signal
<input checked="" type="checkbox"/>  koncak_x_plus	Signal
<input checked="" type="checkbox"/>  reference_y	Signal
<input checked="" type="checkbox"/>  reference_z	Signal
<input checked="" type="checkbox"/>  senzor_dovnitř	Signal
<input checked="" type="checkbox"/>  senzor_ven	Signal
<input checked="" type="checkbox"/>  total_stop	Signal

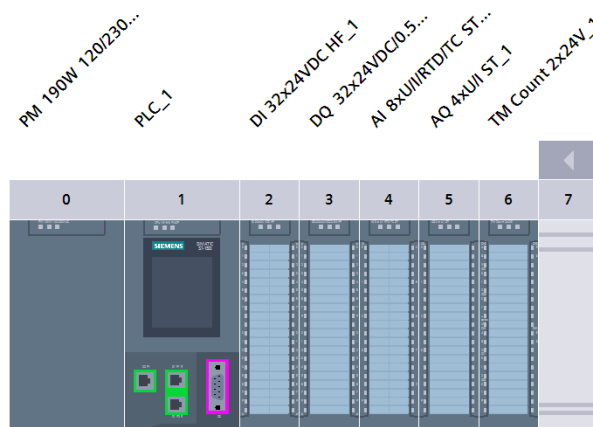
Obr. 26: Vytvořené signály pro PLC

3.2. Tvorba PLC programu

Stavebnice Fischertechnik jsou v rámci předmětu PLC řízení výrobních strojů a zařízení řízeny pomocí modulárního programovatelného automatu řady S7-1500 s HMI Panely KTP700 Basic PN od firmy Siemens. Základními součástmi všech PLC sestav jsou:

- zdroj PM180W 120/230VA
- procesorová jednotka CPU 1516-3 PN/DP
- digitální vstupy DI 32x24VDC HF
- digitální výstupy DQ 32x24VDC/0.5A HF
- analogové vstupy AI 8xU/I/RTD/TC ST
- analogové výstupy AQ 4xU/I ST
- v závislosti na řízené stavebnici čítací moduly TM Count 2x24V
- externí tlačítko TOTAL STOP

Tlačítko TOTAL STOP je vedeno pouze jako jeden ze vstupů do PLC. Součástí sestavy není Safety PLC, ani žádný jiný bezpečnostní obvod. Důvodem je určení PLC sestavy pro výuku PLC programování, kdy téma “Safety“ je již nad rámec praktické výuky semestrálního kurzu i nad rámec této práce. Virtualizace zmíněné sestavy bude provedena v programu TIA Portal V14 (Obr. 27). Jedná se o program, který je určený ke konfiguraci jednotlivých modulů sestav, k napsání řídicího programu, a také ke konfiguraci HMI panelu. Následná funkčnost výsledného virtuálního PLC bude ověřena pomocí programu S7-PLCSIM Advanced V1.0 SP1. Tento program vytvoří virtuální instanci PLC S7-1500 do které lze nahrát vytvořený řídicí program v TIA Portalu. Výstupem takto vytvořeného virtuálního PLC budou tagy vstupů a výstupů, které bude zapotřebí propojit s odpovídajícími vstupy a výstupy ve virtuálním modelu.



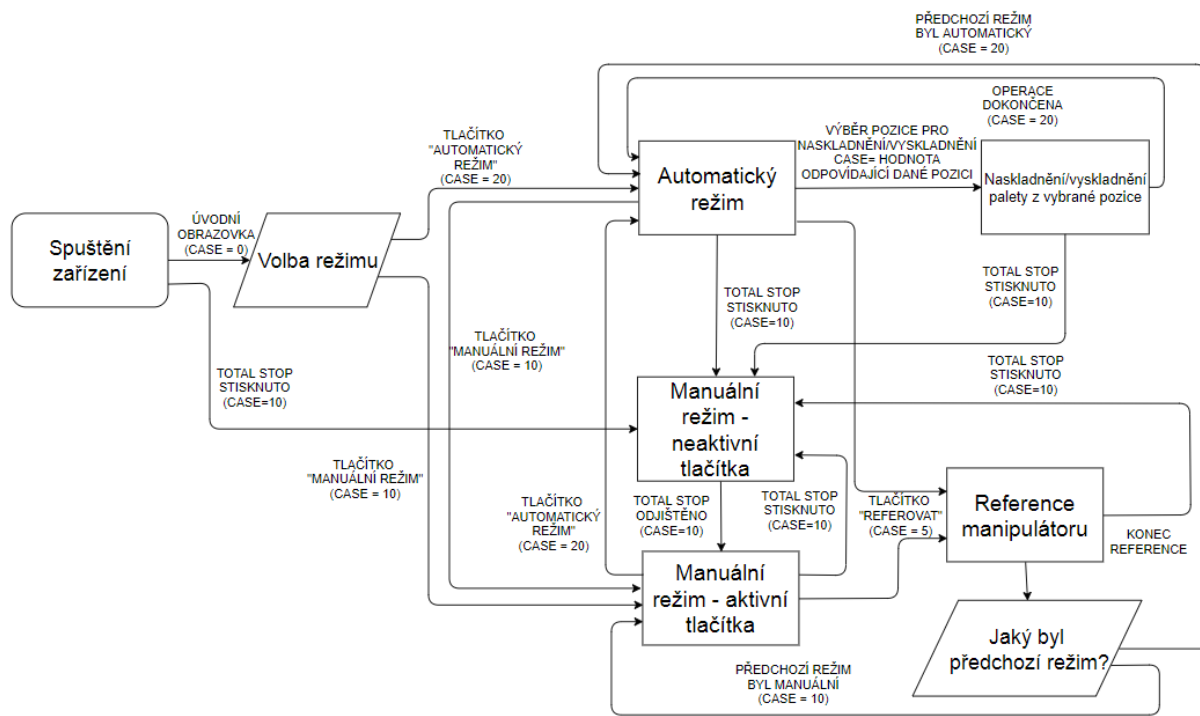
Obr. 27: Konfigurace virtuální sestavy v programu TIA Portal

3.2.1. Zadání pro tvorbu PLC

Zadání pro tvorbu PLC pro automatizovaný sklad bylo navrženo zpracovatelem této diplomové práce. Cílem je vytvořit program pro automatický a manuální režim automatizovaného skladu. Funkcí automatického režimu by mělo být automatické založení palety z dopravníku do pozice ve skladě po stisknutí příslušného tlačítka obsluhou. Funkcí manuálního režimu by mělo být dostupné ovládání všech pohybových prvků automatizovaného skladu pomocí HMI. Do obou režimů by také mělo být zakomponováno tlačítko TOTAL STOP, které by mělo přerušit právě probíhající děje a přepnout systém do manuálního režimu, a příkaz k referenci manipulátoru, který zajistí přejezd manipulátoru do krajních poloh. Reference manipulátoru je nutná z důvodu použití inkrementálních enkodérů na osách Y a Z.

3.2.2. Struktura řídicího programu

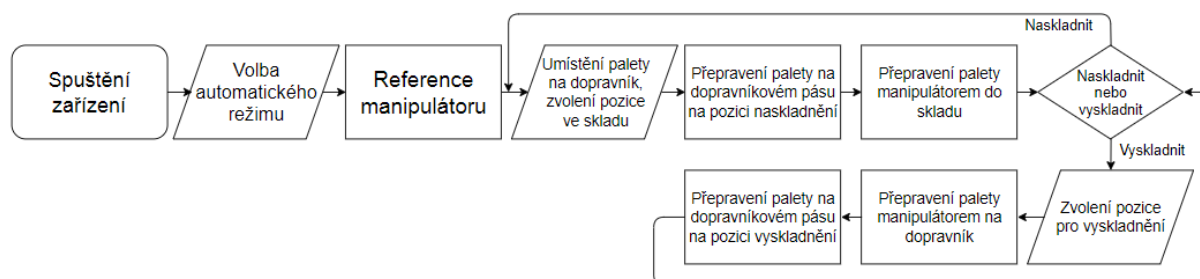
Celý program je uložen v jednom organizačním bloku OB1 s názvem "Main". Organizační blok tvoří rozhraní mezi operačním systémem a programem, který je systémem cyklicky volán. Základem organizačního bloku je funkce CASE, která provádí pouze určité instrukce v organizačním bloku v závislosti na její hodnotě typu INTEGER. Hodnota funkce CASE se mění v závislosti na splnění určené podmínky, jako je například zmáčknutí tlačítka na ovládacím panelu nebo změny hodnoty čidla na řízené sestavě. Tyto podmínky jsou realizovány pomocí funkce IF. V úvodu programu jsou zavedené pomocné RS bloky s názvy "reference", "automat" a "manual", které slouží k přepínání hodnoty funkce CASE, respektive k přecházení mezi automatickým a manuálním režimem nebo k puštění reference manipulátoru (Obr. 28). Dále následuje druhá sada pomocných RS bloků, které jsou spojeny s tlačítky na ovládacím panelu. Tato sada bude blíže popsána v kapitole 3.2.6. Dále již následuje funkce CASE, která, jak již bylo naznačeno, přepíná program mezi automatickým a manuálním režimem. V počátečním stavu 0 jsou deklarovány hodnoty všech výstupů jako FALSE, čímž dojde k zamezení samovolného rozjetí některého z pohonů po spuštění programu. Poté následují čtyři funkce IF, kdy první funkce v případě zmáčknutí tlačítka TOTAL STOP přehodí program do manuálního režimu. Druhá funkce spustí referenci manipulátoru v případě, že proměnná "reference" nabyde hodnoty TRUE. Třetí funkce přehodí program do automatického režimu v případě, že proměnná "automat" nabyde hodnoty TRUE. Čtvrtá funkce přehodí program do manuálního režimu v případě, že proměnná "manual" nabyde hodnoty TRUE. Dále již následují části programu pro jednotlivé režimy, které budou popsány v samostatných kapitolách.



Obr. 28: Základní stavy PLC

3.2.3. Automatický režim

Z požadavků plyne, že automatický režim musí po vložení palety na dopravník a po zvolení pozice skladu automaticky zadat příkazy dopravníku a manipulátoru, díky kterým dojde k přemístění palety do požadované pozice. V případě již založené palety musí po zadání příkazu dojít k jejímu automatickému vyskladnění. Celá situace je zobrazena na následujícím stavovém diagramu (Obr. 29).



Obr. 29: Stavový diagram automatického režimu

Dle Obr. 28 odpovídá části kódu pro automatický režim funkce CASE o hodnotě 20. Zde je stejně jako v hodnotě 0 nejdřív deklarována hodnota FALSE pro všechny výstupy. Dále následují stejné funkce IF, jako ve stavu 0, pro zapnutí referování, přepnutí do manuálního režimu, případně pro případ zmáčknutí tlačítka TOTAL STOP. Za nimi je umístěno osmnáct funkcí IF, devět pro volbu pozic naskladnění a devět pro volbu pozice vyskladnění. Každá

z těchto funkcí IF, po splnění podmínky zmáčknutí tlačítka příslušné pozice na ovládacím panelu a nespouštěném tlačítku TOTAL STOP, změni hodnotu funkce CASE na příslušné číslo odpovídající části programu pro požadované naskladnění, respektive vyskladnění (Tabulka 2).

Hodnota funkce CASE	Účel dané části programu
25	Naskladnění palety do pozice „1“
30	Naskladnění palety do pozice „2“
35	Naskladnění palety do pozice „3“
40	Naskladnění palety do pozice „4“
45	Naskladnění palety do pozice „5“
50	Naskladnění palety do pozice „6“
55	Naskladnění palety do pozice „7“
60	Naskladnění palety do pozice „8“
65	Naskladnění palety do pozice „9“
70	Vyskladnění palety z pozice „1“
75	Vyskladnění palety z pozice „2“
80	Vyskladnění palety z pozice „3“
85	Vyskladnění palety z pozice „4“
90	Vyskladnění palety z pozice „5“
95	Vyskladnění palety z pozice „6“
100	Vyskladnění palety z pozice „7“
105	Vyskladnění palety z pozice „8“
110	Vyskladnění palety z pozice „9“

Tabulka 2: Hodnoty funkce CASE pro operace naskladnění/vyskladnění palety z určené pozice

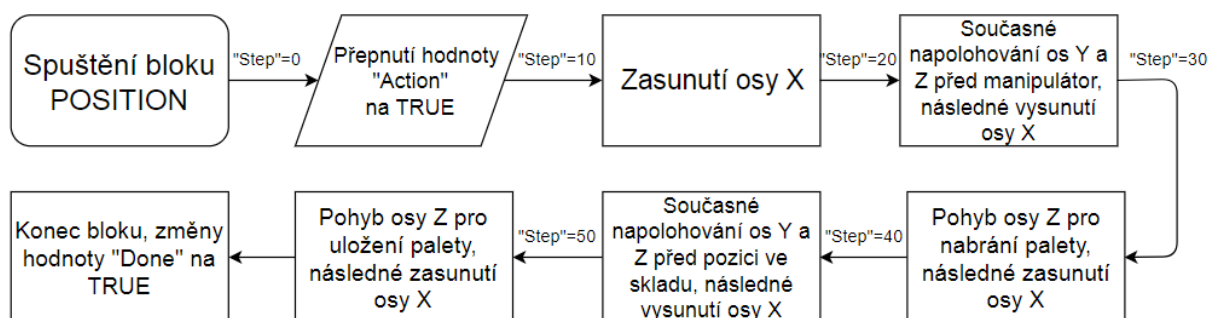
V této části se vyskytuje RS blok pro ovládání dopravníkového pásu a instanční datový blok funkčního bloku s názvem “Position“ pro ovládání manipulátoru. Do funkčního bloku lze psát, stejně jako do organizačního bloku, části programu. Navíc zde lze definovat lokální proměnné. Z funkčního bloku lze následně vytvořit jeho instanci pomocí datového bloku. Tuto instanci je možné následně vyvolat v organizačním bloku. Část programu, která je v instanci obsažena, se v organizačním bloku nezobrazí. Je pouze nutné přiřadit k lokálním proměnným hodnoty. Jeden funkční blok tak může být vykonáván několikrát nezávisle na sobě.

Základem funkčního bloku “Position“ je další funkční blok Automatic_move_Y_Z (Obr. 30), který je do bloku “Position“ vložen rovněž pomocí instance. Funkcí bloku Automatic_move_Y_Z je pohyb osy Y nebo Z na požadovanou pozici. Jako vstupy jsou definovány tři proměnné typu DInt s názvy “target“, “actual“, “deviation“ a jedna proměnná typu Bool s názvem “action“. Jako výstupy jsou definovány tři proměnné typu Bool s názvy “dir_plus“, “dir_minus“ a “done“, jedna proměnná typu DInt s názvem “difference“ a jedna proměnná typu Int s názvem “state“. Hodnota proměnné “target“ je požadovaný počet pulzů načtených enkodérem v pohonu osy Y nebo Z, který odpovídá požadované poloze. Hodnota proměnné “actual“ odpovídá aktuálnímu počtu pulzů načtených enkodérem. Po odečtení těchto dvou hodnot vyjde odchylka požadované polohy, která je definována jako hodnota proměnné “difference“. V programu následuje funkce CASE, která používá definovaný výstup “state“ jako hodnotu aktuálního stavu. V případě, že hodnota vstupu “action“ nabyde hodnoty TRUE, dojde k přepnutí do následujícího stavu. Zde dojde k vyhodnocení hodnoty proměnné “difference“. V případě, že je větší než 1, proměnná “dir_plus“ nabyde hodnoty TRUE. Když bude menší než -1, nabyde hodnoty TRUE proměnná “dir_minus“. Důvodem tohoto vyhodnocení je spuštění správného směru pohonu v závislosti na zvoleném souřadnicovém systému. Zvolený interval definuje toleranci nepřesnosti najetí na polohu. Pro definici jeho velikosti je použita proměnná “deviation“. V organizačním bloku “Main“ byla nastavena hodnota této proměnné 5. V případě, kdy byla hodnota proměnné “difference“ porovnávaná s 0, docházelo k nežádoucím jevům na pohonu vlivem nepřesnosti enkodéru. Po dosažení požadované polohy přejde funkce CASE do dalšího stavu, ve kterém proměnné “dir_plus“ a “dir_minus“ nabydou hodnoty FALSE a “done“ hodnoty TRUE.

```
1 (*Základní funkční blok pro pohyb v ose Y nebo Z na požadovanou hodnotu*)
2 #difference := #target - #actual; //Definice rozdílu cílové a aktuální hodnoty z enkodéru
3 CASE #state OF
4 0: #done := FALSE; //Počáteční stav funkčního bloku
5   IF #action THEN //Zmáčknutí tlačítka pro založení/vyložení palety
6     #state := 5;
7   END_IF;
8 5: //Polohování na požadovanou pozici
9   IF #difference > #deviation THEN //Porovnávání rozdílu s intervalem, definující přípustnou
10      //odchylku od žádané hodnoty
11     #dir_plus := TRUE; //Aktivace příslušného pohonu v závislosti
12     //na znaménku rozdílu
13   ELSIF #difference < -#deviation THEN //Porovnávání rozdílu s intervalem, definující přípustnou
14      //odchylku od žádané hodnoty
15     #dir_minus := TRUE; //Aktivace příslušného pohonu v závislosti
16     //na znaménku rozdílu
17   ELSE
18     #state := 10;
19   END_IF;
20 10: //Konec polohování
21   #dir_plus := FALSE; //Vypnutí pohonů po dosažení cílové hodnoty
22   //v definovaném intervalu
23   #dir_minus := FALSE;
24   #done := TRUE;
25   IF NOT #action THEN
26     #state := 0;
27   END_IF;
28 END_CASE;
```

Obr. 30: Funkční blok Automatic_move_Y_Z

Takto definovaný funkční blok je, jak již bylo řečeno, v bloku POSITION (Obr. 31) vyvoláván pomocí instancí s názvy “Automatic_move_Y_DB“ a “Automatic_move_Z_DB“, pro pohyb os Y a Z. Mezi vstupy jsou proměnné typu DInt pro nastavení požadovaných posunů, dále dvě proměnné typu Bool pro ovládání osy X, které budou následně propojeny s koncovými snímači na této ose, a jedna proměnná, rovněž typu Bool, která uvede blok v činnost. Mezi výstupy je sada proměnných typu Bool, která bude následně propojena s jednotlivými pohony manipulátoru. Dále jsou zde pomocné proměnné typu Int a Bool pro řízení programu. Průběh celého bloku POSITION, který je vyobrazen na stavovém diagramu (Obr. 31), je rovněž řešen pomocí funkcí CASE a IF.



Obr. 31: Stavový diagram instancí funkčního bloku Position pro naskladnění palety

Po přepnutí hodnoty proměnné “Done“ na TRUE dojde k přepnutí funkce CASE v organizačním bloku “Main“ zpět do stavu 20, ve kterém zůstane do dalších zásahů operátorem.

3.2.4. Manuální režim

Manuální režim se v operačním bloku nachází ve funkci CASE pod hodnotou 10 (Obr. 28). Mimo podmínek IF pro zreferování manipulátoru a přepnutí do automatického režimu se zde nachází příkazy pro ovládání pohonů jednotlivých os v obou směrech. Bezpečností podmínkou je stejně jako v automatickém režimu nestisknuté tlačítko TOTAL STOP. V případě, že na ose dojde k spuštění koncového snímače, signál do pohonu bude v příslušném směru přerušen a osa se zastaví.

3.2.5. Reference manipulátoru

Příkaz pro zreferování manipulátoru je v operačním bloku pod hodnotou 5. Je složen z podmínek IF, které se spustí za předpokladu zadání požadavku pro referenci manipulátoru a nestisknutém tlačítku TOTAL STOP. Jako první dojde k zasunutí osy X z důvodu vyloučení kolize s jinými objekty skladu. Po dojetí osy X na koncový snímač následuje současný pohyb os Y a Z ke svým koncovým spínačům. Sepnutím koncových snímačů v osách Y a Z bude vyslán signál do data bloků High_Speed_Counter_1 a High_Speed_Counter_2, kterými jsou zpracovávána data z enkodérů na osách Y a Z. Data blok s příponou _1 je pro osu Y a s příponou _2 pro osu Z. Aby bylo možné načítat a přepisovat data z těchto data bloků, musí být hodnoty lokálních proměnných “SwGate“ a “CaptureEnable“ nastaveny na hodnotu TRUE. Dále bude k lokální proměnné “SetCountValue“ přiřazena hodnota vstupu z koncového snímače v ose Y, respektive Z (Obr. 32). Díky tomu bude při každém sepnutí koncového snímače aktuální hodnota enkodéru přepsána na 0.

```
High_Speed_Counter_1(SwGate := TRUE,  
                    CaptureEnable := TRUE,  
                    SetCountValue := "di_reference_y");  
High_Speed_Counter_2(SwGate := TRUE,  
                    CaptureEnable := TRUE,  
                    SetCountValue := "di_reference_z");
```

Obr. 32: Data bloky zpracovávající signál z enkodérů v osách Y a Z

3.2.6. Ovládací panel

Vzhled ovládacího panelu, pomocí kterého obsluha ovládá chod programu PLC, je rovněž vytvořen v programu TIA Portal. Na úvodní obrazovce (Obr. 33) je obsluha vyzvána, zda bude programem řízena stavebnice nebo virtuální model. Důvod tohoto výběru bude popsán v kapitole 3.3. Následně je možný výběr, zda řízená sestava bude ovládána v manuálním nebo automatickém režimu.

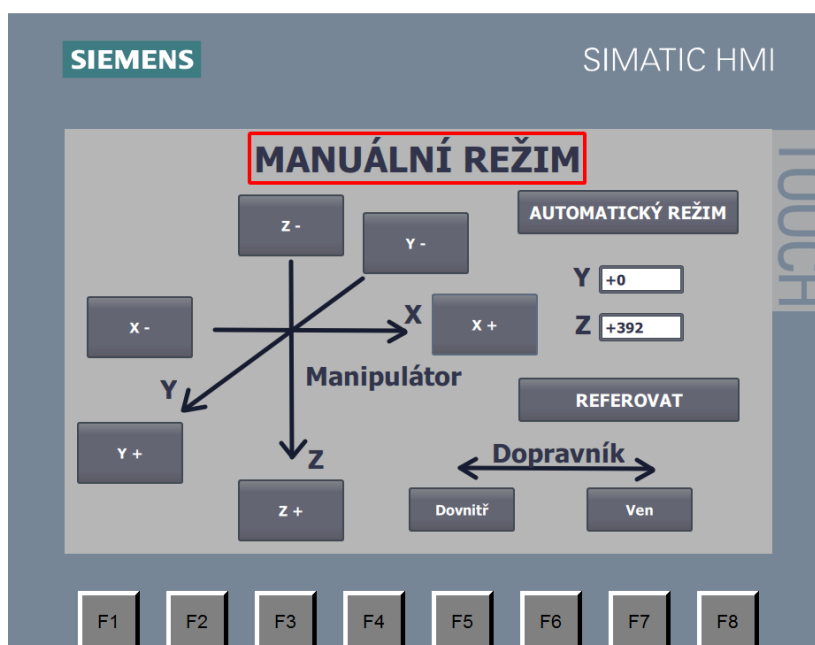


Obr. 33: Úvodní obrazovka ovládacího panelu

Tlačítko pro přepnutí mezi stavebnicí a virtuálním modelem je vytvořeno pomocí tlačítka *Switch*. V dialogu tlačítka, v záložce Events je pod možností Switch ON použita funkce SetBit pro proměnnou “Simulation“ a pod možností Switch OFF funkce ResetBit pro stejnou proměnnou. V případě stisknutí tlačítka dojde k přepnutí hodnoty proměnné “Simulation“ na TRUE a při opětovném stisknutí je hodnota vrácena na FALSE. Tlačítka pro výběr manuálního nebo automatického režimu jsou typu *Button*. V kartě Events tlačítka pro výběr manuálního režimu je v možnosti “Click“ použita funkce ActivateScreen s vybranou možností “Manual“, což je název obrazovky pro manuální režim. V možnosti “Press“ je použita funkce SetBit pro proměnnou “manual_button“ a v možnosti “Release“ funkce ResetBit pro stejnou proměnnou. Při zmáčknutí tohoto tlačítka tak dojde k přepnutí HMI do obrazovky manuálního režimu a hodnota funkce CASE v operačním bloku “Main“ je přepnuta na 10, která odpovídá části programu pro manuální režim. Na stejném principu je vytvořeno tlačítko pro Automatický

režim. Po jeho stisknutí dojde k přepnutí HMI do obrazovky automatického režimu a hodnota funkce CASE je přepnuta na 20, která odpovídá části programu pro automatický režim.

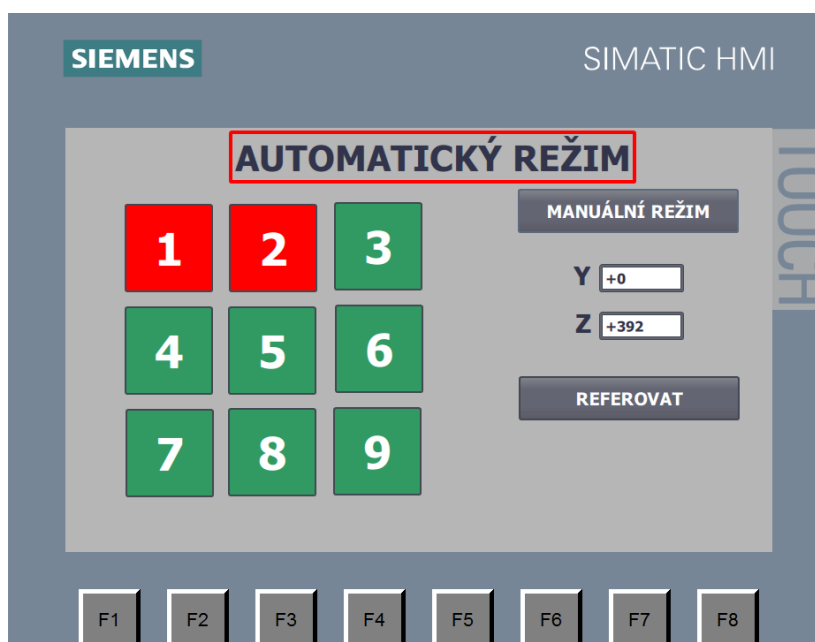
Obrazovka pro manuální režim (Obr. 34) nabízí operátorovi tlačítka typu *Button*, která jsou pomocí funkcí SetBit a ResetBit propojeny s proměnnými zodpovědnými za ovládání pohonů manipulátoru a dopravníku. Dále má operátor možnost spustit cyklus reference manipulátoru pomocí tlačítka “REFEROVAT“. Po jeho stisknutí je hodnota funkce CASE v operačním bloku “Main“ přepnuta na 5. V této části je část kódu pro zmíněnou referenci manipulátoru. Dále jsou na obrazovce dvě pole s číselnými hodnotami. Pole u písmena “Y“ zobrazuje aktuální načtenou hodnotu z enkodéru manipulátoru na ose Y a pole u písmena “Z“ ukazuje hodnotu enkodéru z osy Z. Rovněž je zde možnost pro přepnutí do automatického režimu pomocí přidaného tlačítka.



Obr. 34: Obrazovka pro manuální režim

Základem obrazovky pro automatický režim (Obr. 35) je devět tlačítek typu *Button* s označeními čísly 1 až 9, které reprezentují jednotlivé pozice ve skladu. Po úvodním spuštění PLC jsou tato tlačítka zelená, to značí, že všechny pozice ve skladu jsou volné. Po jejich stisknutí je proměnná “button_X_in“, kde X značí číslo zvolené pozice, přepnuta do hodnoty TRUE pomocí funkce SetBit. To změní hodnotu funkce CASE v operačním bloku “Main“ na číslo, kterému odpovídá příslušná část programu s data blokem funkčního bloku “Position“, který byl popsán v kapitole 3.2.3. V případě, že je v dopravníku založena paleta, proběhne automatický cyklus naskladnění palety do zvolené pozice ve skladu. Během cyklu jsou všechna

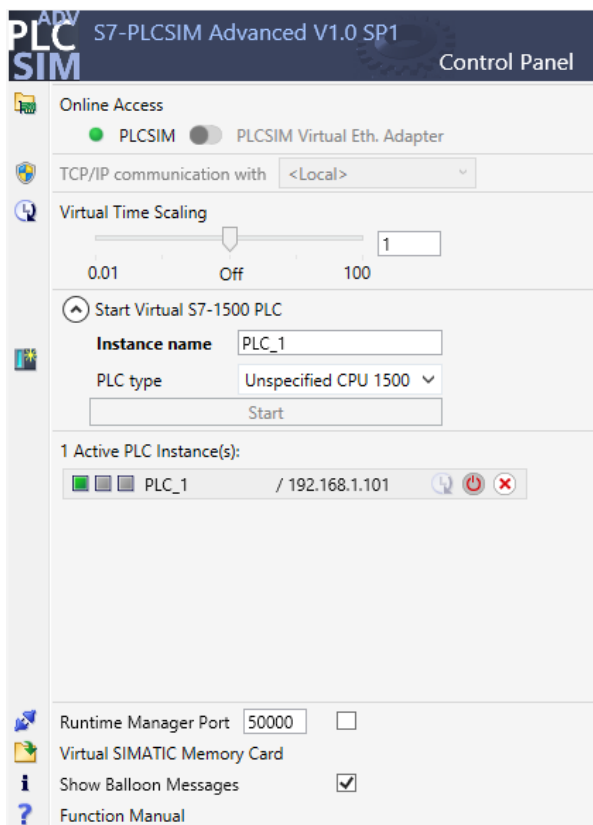
tlačítka neaktivní. Po dokončení cyklu přejde hodnota funkce CASE zpět na 20 a zelené tlačítko je zneviditelněno. Na jeho místě se objeví stejné tlačítko s červenou barvou. To je zapříčiněno propojením tlačítek s proměnnou "Button_rs_X", kde X znovu značí číslo pozice, pomocí funkce *Visibility* v záložce Animations. Proměnné "Button_rs_X" byly vygenerovány ve funkčním bloku "retentive_tags" a byly nastaveny jako retentivní. To znamená, že i po opětovném zapnutí PLC bude zachována jejich hodnota. Díky tomu zůstane zachována informace o obsazených pozicích ve skladu i po vypnutí PLC. Červené tlačítko slouží ke spuštění sekvence vyskladnění. Po provedeném vyskladnění se barva tlačítka přepne zpět na zelenou. Dále je zde, stejně jako na předchozí obrazovce, tlačítko pro referenci manipulátoru, pole s hodnotami enkodérů a tlačítko pro přepnutí do manuálního režimu.



Obr. 35: Obrázek pro automatický režim

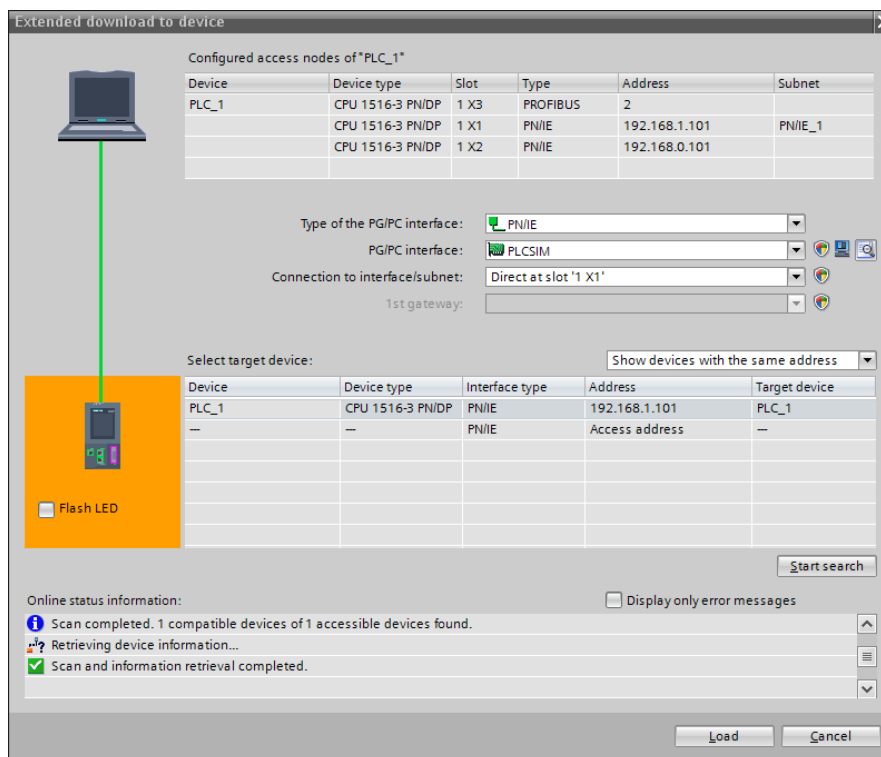
3.2.7. Virtualizace PLC

Pro virtualizaci PLC je použit program PLCSim Advanced V1.0 SP1 (Obr. 36). V dialogu tohoto programu byla u “Online Access“ vybrána možnost *PLCSIM*. Vytvořená instance byla pojmenována “PLC_1“. Ve vysouvací nabídce “PLC type“ byla vybrána možnost *Unspecified CPU 1500* a následně bylo zmáčknuto tlačítko *Start*. Následně se vytvoří instance se zadaným jménem, která reprezentuje právě vytvořené virtuální PLC.



Obr. 36: Dialog programu PLCSim Advanced

Jak bylo zmíněno v kapitole 3.2, do této instance lze nahrát řídicí program vytvořený v programu TIA Portal. Po zmáčknutí ikony s názvem “Download to device“ se zobrazí nabídka pro výběr zařízení (Obr. 37), do kterého má být program nahrán. V řádku “Type of the PG/PC interface“ byl vybrán typ sběrnice *PN/IE*, v dalším řádku “PG/PC interface“ byla jako adaptér pro připojení vybrána možnost *PLCSIM*. V posledním řádku “Connection to interface/subnet“ byla vybrána možnost *Direct at slot '1 XI'*. Následně bylo zmáčknuto tlačítko *Start search*. Po chvíli se v nabídce objeví vytvořená instance s názvem “PLC_1“ a po jejím vybrání a zmáčknutí tlačítka *load* bude řídicí program do této instance nahrán.

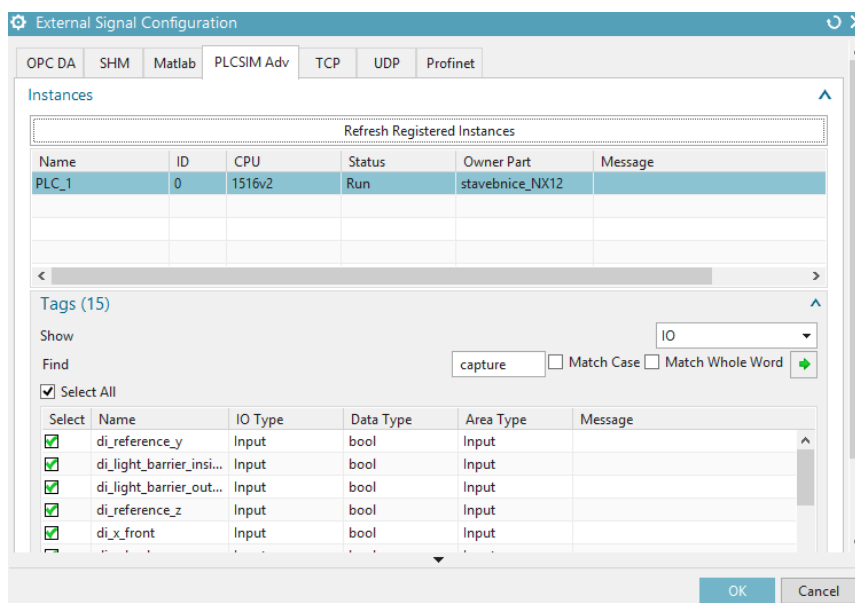


Obr. 37: Dialog pro nahrání řídicího programu do zařízení

Verze softwaru PLCSim Advanced V1.0 SP1 obsahuje chybu na základě které nelze simulovat virtuální PLC s procesorovou jednotkou CPU 1516-3 v2.1. Tuto chybu lze obejít snížením verze procesorové jednotky na v2.0.

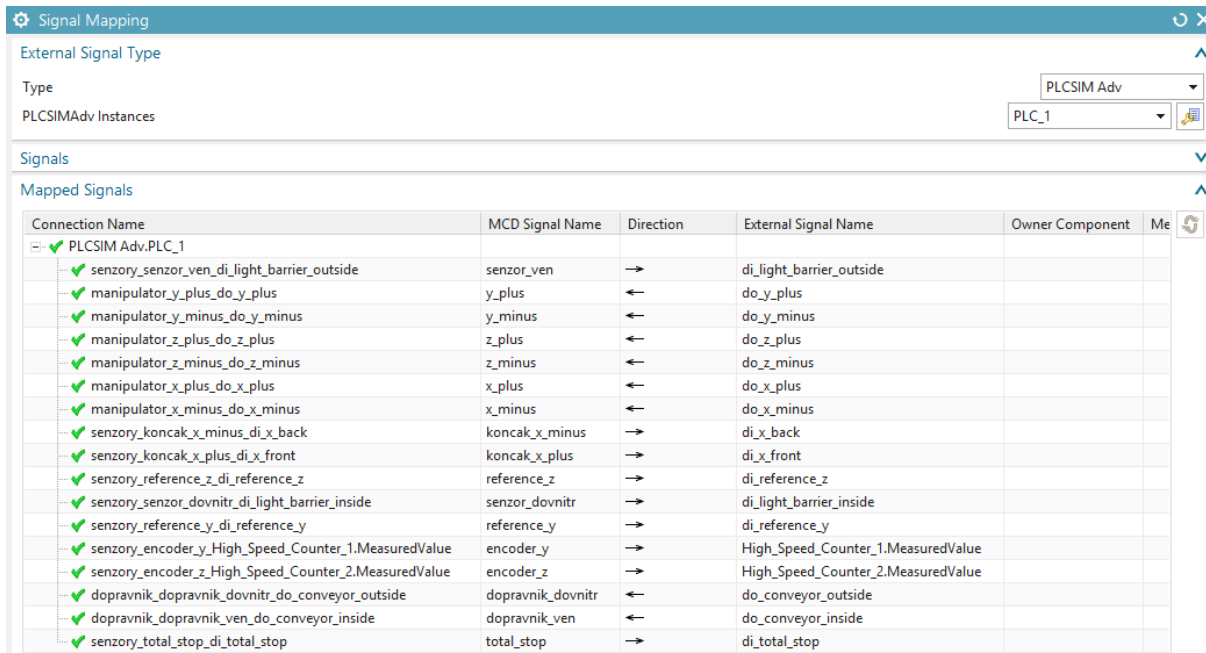
3.3. Propojení virtuálního modelu a PLC

Propojení signálů virtuálního modelu a PLC běžícího v programu PLCSim Advanced bylo provedeno v programu NX MCD. Pomocí funkce *External signal configuration* lze v kartě PLCSIM Adv vybrat instanci virtuálního PLC s názvem “PLC_1“. Po vybrání zmíněné instance se v dolní polovině dialogu funkce objeví Tagy vstupů a výstupů virtuálního PLC. Po vybrání “IOMDB“ z výsuvné nabídky řádku Show lze prohlédnout i Tagy definované uvnitř PLC programu (Obr. 38). Ve sloupci Select byly vybrány Tagy, které je zapotřebí propojit se signály z modelu.



Obr. 38: Dialog funkce *External signal configuration*

K samotnému propojení signálů byla použita funkce *Signal mapping*. V dialogu této funkce bylo z výsuvné nabídky řádku Type vybrána možnost PLCSIM Adv a jako instance “PLC_1“. V další části dialogu se na levé straně vyskytují vytvořené signály v modelu a na pravé straně importované signály z virtuálního PLC. Vybráním jednoho signálu z obou stran lze vytvořit propojení. Aby bylo možné signály propojit, musí být stejného datového typu a opačného IO typu. Proto byly signály pro ovládání aktorů vytvořeny jako Input a signály senzorů jako Output. Po následném propojení se vytvořená vazba objeví ve spodní části dialogu s názvem Mapped signals. V této části lze propojení zrušit.



Connection Name	MCD Signal Name	Direction	External Signal Name	Owner Component	Me
PLCSIM Adv.PLC_1					
✓ senzory_senzor_ven_di_light_barrier_outside	senzor_ven	→	di_light_barrier_outside		
✓ manipulator_y_plus_do_y_plus	y_plus	←	do_y_plus		
✓ manipulator_y_minus_do_y_minus	y_minus	←	do_y_minus		
✓ manipulator_z_plus_do_z_plus	z_plus	←	do_z_plus		
✓ manipulator_z_minus_do_z_minus	z_minus	←	do_z_minus		
✓ manipulator_x_plus_do_x_plus	x_plus	←	do_x_plus		
✓ manipulator_x_minus_do_x_minus	x_minus	←	do_x_minus		
✓ senzory_koncak_x_minus_di_x_back	koncak_x_minus	→	di_x_back		
✓ senzory_koncak_x_plus_di_x_front	koncak_x_plus	→	di_x_front		
✓ senzory_reference_z_di_reference_z	reference_z	→	di_reference_z		
✓ senzory_senzor_dovnitri_di_light_barrier_inside	senzor_dovnitri	→	di_light_barrier_inside		
✓ senzory_reference_y_di_reference_y	reference_y	→	di_reference_y		
✓ senzory_encoder_y_High_Speed_Counter_1.MeasuredValue	encoder_y	→	High_Speed_Counter_1.MeasuredValue		
✓ senzory_encoder_z_High_Speed_Counter_2.MeasuredValue	encoder_z	→	High_Speed_Counter_2.MeasuredValue		
✓ dopravnik_dopravnik_dovnitri_do_conveyor_outside	dopravnik_dovnitri	←	do_conveyor_outside		
✓ dopravnik_dopravnik_ven_do_conveyor_inside	dopravnik_ven	←	do_conveyor_inside		
✓ senzory_total_stop_di_total_stop	total_stop	→	di_total_stop		

Obr. 39: Propojení signálu modelu s virtuálním PLC

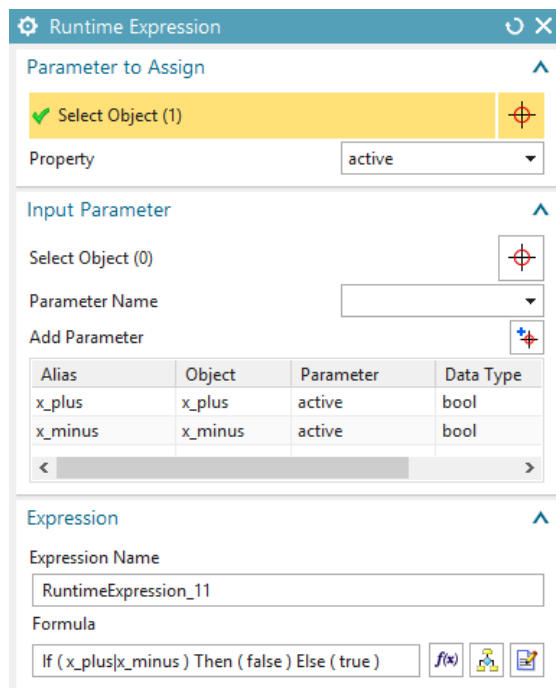
Jako problém se objevilo propojení signálu enkodéru “encoder_y“ s tagem “High_speed_counter_1.CountValue“, který zodpovídá za načtení hodnoty vysílané enkodérem na ose Y do data bloku “High_speed_counter_1“ v programu PLC. V tomto data bloku je datový typ tagu “CountValue“ definován jako DInt, nicméně v dialogu funkce *Signal mapping* se tento tag jeví jako datový typ Int. Signál “encoder_y“ je datového typu DInt a kvůli rozlišnosti datových typů nelze uskutečnit propojení. Tento problém byl vyřešen pomocí příkazu "High_Speed_Counter_1".CountValue := "High_Speed_Counter_1".MeasuredValue, který byl zadán do organizačního bloku OB1. Hodnota proměnné “CountValue“ byla přiřazena hodnotě proměnné “MeasuredValue“, která je datového typu Real. Funkce *Signal mapping* tuto proměnnou načte jako proměnnou typu DInt. Díky tomu lze uskutečnit propojení se signálem “encoder_y“. Nicméně se jedná o řešení, kvůli kterému bylo nutné upravit řídicím program PLC. Při zpětném použití takto upraveného programu k řízení stavebnice by nedocházelo k načítání hodnot z enkodéru vlivem použitého příkazu. Proto byl použitý příkaz zakomponován do funkce IF, jejíž podmínka bude splněna v případě, že na úvodní obrazovce dotykového panelu bude vybrána možnost virtuálního modelu. Stejná situace nastala u signálu

“encoder_z“ a tagu “High_speed_counter_2.CountValue“. Řešení této situace bylo obdobné jako u předchozího případu.

3.3.1. Testování simulace

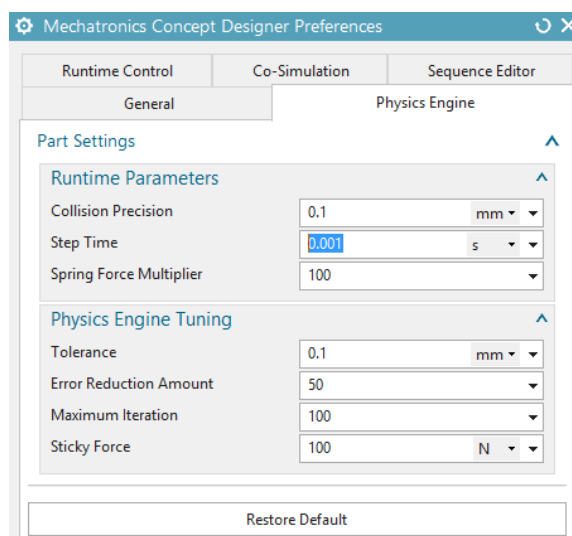
Testování simulace probíhalo na notebooku Toshiba Satellite L50-A-19N, který disponuje procesorem Intel Core i7-4700MQ o frekvenci 2.40GHz, systémovou paměť 8GB RAM, grafickou kartou NVIDIA GeForce GT 740M a operačním systémem Windows 8.1. Veškeré následující úpravy pro optimální běh simulace jsou vypovídající pouze pro počítače se srovnatelným výpočetním výkonem. Při použití počítače s nižším výpočetním výkonem nemusí být následující úpravy dostatečné.

Po spuštění virtuálního PLC s nahraným programem lze spustit simulaci v prostředí NX MCD pomocí příkazu Play. K ovládání manipulátoru je nutné spuštění simulovaného dotykového panelu v programu TIA Portal. Po přechodu do manuálního režimu bylo otestováno ovládání jednotlivých pohybových os. Jako problém se objevil setrvačný pohyb os, který nastal vlivem absence jakéhokoliv tření. K zamezení tohoto jevu byla vytvořena třetí funkce *Speed control* pro pohybové osy manipulátoru a dopravníkový pás s příponou “_zero“. Velikost rychlosti posuvu byla nastavena na 0 mm/s. Pro tyto funkce byly následně vytvořeny podmínky aktivity v kartě *Runtime expression*. Dialog podmínky pro funkci Speed control s názvem “x_zero“ je zobrazen na Obr. 40. Jako hlavní parametr byla zvolena funkce “x_zero“ a jako vstupní parametry byly vybrány funkce “x_plus“ a “x_minus“. V řádku *Formula* bylo napsáno samotné znění podmínky. Funkce “x_zero“ bude aktivní pouze v případě, že ani jedná z funkcí “x_plus“ a “x_minus“ aktivní nebude. Tímto řešením je zajištěno zastavení pohybu ovládané osy X díky zaktivování přidané funkce, která má nastavenou nulovou posuvovou rychlost. Toto řešení bylo po ověření implementováno i na ostatní pohybové osy.



Obr. 40: Dialog Runtime expression funkce "x_zero"

Jako dalším problém se ukázalo zpoždování simulace oproti reálnému času, na který bylo nastaveno její škálování. Vlivem toho byla rychlost výsledných posuvů zpomalená úměrně zpoždění simulace. V případě použití časových funkcí v PLC programu by tak simulace ztrácela výpovědní hodnotu. Jako možné řešení se jeví vhodné nastavení proměnné *Step time* v dialogu *Mechatronics Concept Designer Preferences*. Zmíněná proměnná se dá definovat jako perioda aktualizace simulace. V původním nastavení dosahovala hodnoty 0,001s (Obr. 41).



Obr. 41: Dialog funkce Mechatronics Concept Designer Preferences

Nová hodnota byla nastavena na 0,005s. Simulace se během ověřování vhodnosti nastavené hodnoty přestala zpožďovat, nicméně při bližším zkoumání se objevili nové problémy. Vlivem

větší časové periody začalo docházet k prolnutí pohybujících se objektů, které by se normálně o sebe zastavili díky nastaveným kolizním vlastnostem. Dále docházelo k rozkmitání pohybových os řízenými enkodéry. To bylo způsobeno snahou řídicího programu dostat se do požadovaného intervalu hodnot načtených z enkodéru, popsáným v kapitole 3.2.3. Po několika testech byla výsledná hodnota proměnné *Step time* nastavena na hodnotu 0,002. K zamezení kmitání při polohování pohybových os musel být interval rozdílu od požadované polohy zvětšen z (-5,5) na (-15,15). Tímto krokem se značně zlepšila dynamika a průběh simulace, nicméně stále docházelo k mírnému zpoždění. Tento problém definitivně vyřešilo zaškrtnutí možnosti *Use Time Synchronization* u možnosti *PLCSIM Advanced* v kartě *Co-Simulation* dialogu *Mechatronics Concept Designer Preferences*. Zaškrtnutí této možnosti MCD porovná časy simulací modelu a virtuálního PLC a synchronizuje. Po otestování takto nastavené simulace pomocí PLC programu s použitou časovou funkcí byla ověřena vhodnost tohoto nastavení.

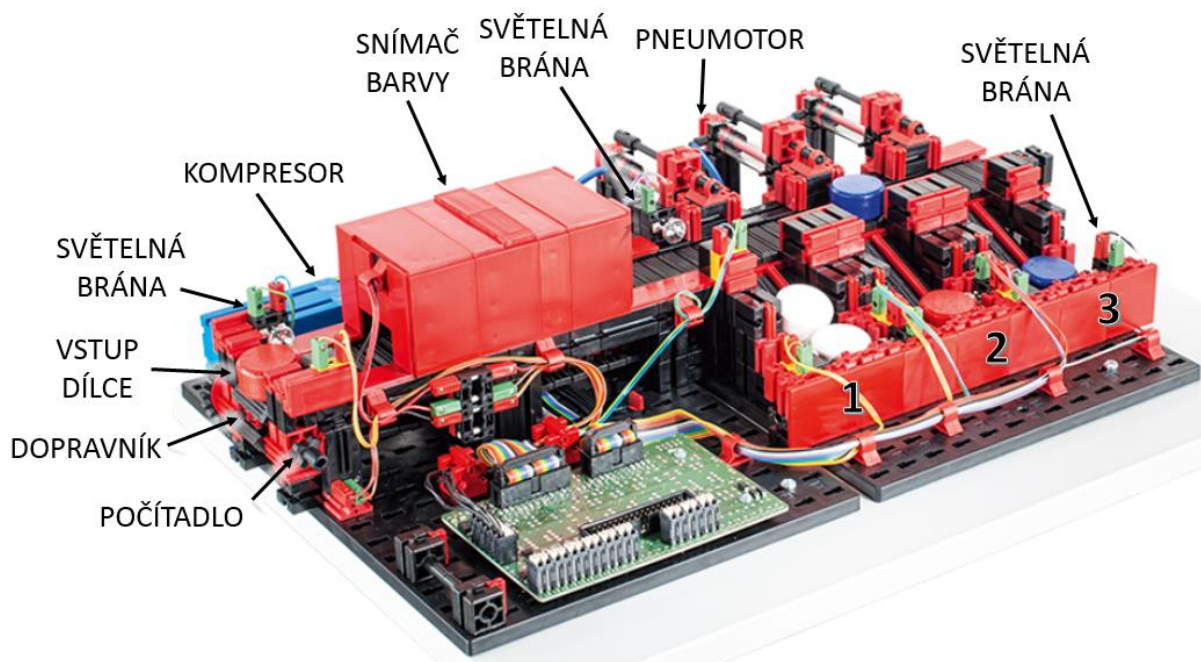
Po vyřešení výše zmíněných problémů byl již průběh simulace bezproblémový a chování modelu během simulace bylo totožné s chováním reálné situace. Bylo ověřeno ovládání pohybových os manipulátoru a dopravníku v rámci manuálního režimu, dále byla provedena úspěšná reference manipulátoru. V rámci automatického režimu bylo provedeno naskladnění a vyskladnění do všech pozic ve skladu. Dále bylo testováno tlačítko TOTAL STOP, které bylo stisknuto během výkonu operace naskladňování. Model manipulátoru se dle předpokladu zastavil a program byl přepnut do manuálního režimu, odkud bylo po odjištění tlačítka TOTAL STOP možné hýbat s manipulátorem.

3.4. Tvorba virtuálních modelů ostatních stavebnic

Dle zadání byly vytvořeny virtuální modely třech zbývajících stavebnic, které byly zmíněny v kapitole 3.1. Pro tyto modely již nebyl tvořen kompletní řídicí program. Byl vytvořen pouze základní PLC program, ve kterém byly zvoleny komponenty popsané v kapitole 3.2. Dále zde byly definovány vstupy a výstupy dle předlohy reálných stavebnic, které byly po virtualizaci PLC s takto vytvořeným programem propojeny s vytvořenými signály ze stavebnic. Student předmětu *PLC řízení výrobních strojů a zařízení* si tak při výuce pustí pouze program TIA Portal s předpřipraveným programem pro danou stavebnici a následně se může plně soustředit na samotné programování a verifikaci řídicích programů jejich nahráním do virtuálního PLC vytvořeném v programu PLCSim Advanced a následným simulováním virtuálního modelu v programu NX12. Pro usnadnění procesu byly vytvořeny uživatelské návody popisující potřebné kroky k virtuálnímu zprovoznění vytvořených modelů. Součástí návodu je také popis simulovaných prvků, které se liší od reálné předlohy, pro lepší pochopení modelu.

3.4.1. Třídící linka

Stavebnice třídící linky (Obr. 42) je tvořena dopravníkem, třemi pneumatickými motory poháněnými kompresorem a snímačem barvy. Jednou z jejích funkcí tak může být doprava dílce pomocí dopravníku k čidlu barvy, kde řídicí program zjistí jeho barvu a rozhodne, do které odkládací pozice bude pneumatickým motorem přesunut. K detekci přítomnosti dílce na dopravníku a v odkládacích pozicích slouží světelné brány.

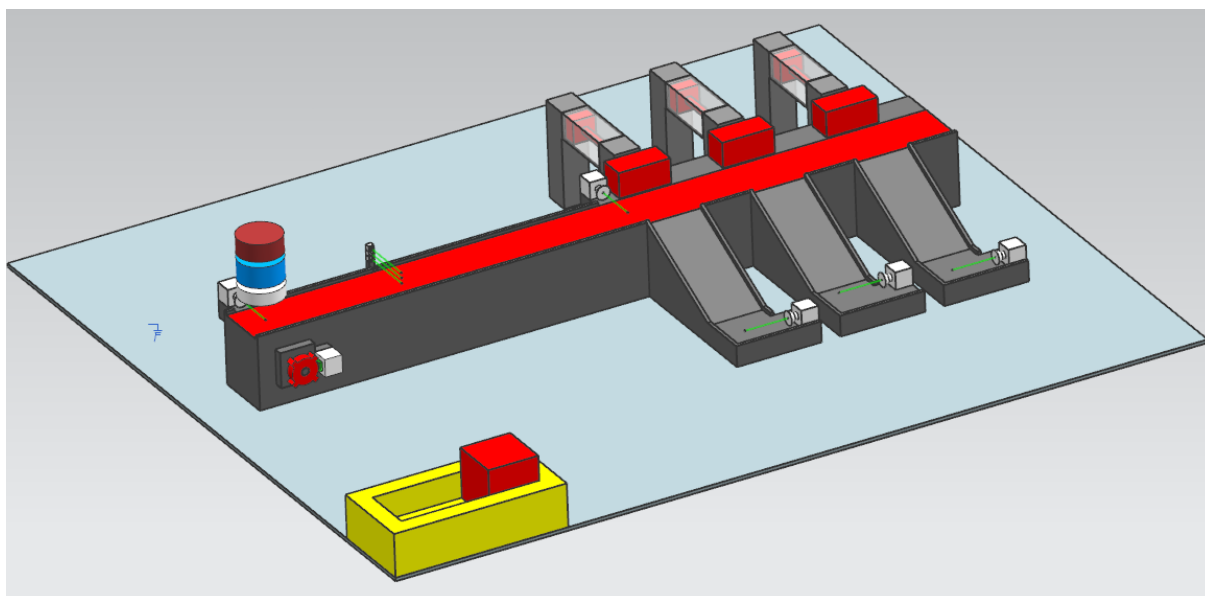


Obr. 42: Stavebnice třídící linky

Digitální vstupy a výstupy PLC	Popis	Název proměnné
do_0.0	Motor dopravníku	do_0.0_dopravnik
do_0.1	Kompresor	do_0.1_kompresor
do_0.2	Pneumatický válec 1	do_0.2_valec_1
do_0.3	Pneumatický válec 2	do_0.3_valec_2
do_0.4	Pneumatický válec 3	do_0.4_valec_3
do_0.5	Snímač barvy	ai_0_color_sensor
di_0.0	Světelná závora před komorou	di_0.0_cidlo_pred_komorou
di_0.1	Světelná závora za komorou	di_0.1_cidlo_za_komorou
di_0.2	Světelná závora zásobníku 1	di_0.2_cidlo_zasobnik_1
di_0.3	Světelná závora zásobníku 2	di_0.3_cidlo_zasobnik_2
di_0.4	Světelná závora zásobníku 3	di_0.4_cidlo_zasobnik_3
di_0.5	Čítač dopravníku	di_0.5_citac_dopravniku
di_1.1	Tlačítko TOTAL STOP	di_total_stop

Tabulka 3: Souhrn vstupů a výstupů třídící linky

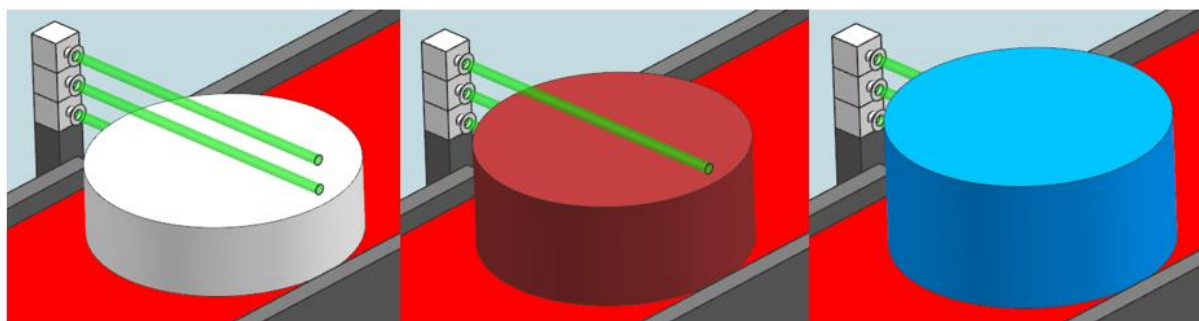
Stejně jako v případě automatizovaného skladu byl vytvořen zjednodušený 3D model v CAD modulu programu NX12 (Obr. 43). Byly zde zakomponovány všechny zmíněné funkční prvky.



Obr. 43: Virtuální model třídící linky

V modulu MCD bylo následovně definováno ovládání dopravníku, pneumatických motorů a senzorů. Definice fyzických vlastností a polohových vazeb proběhla stejným způsobem jako

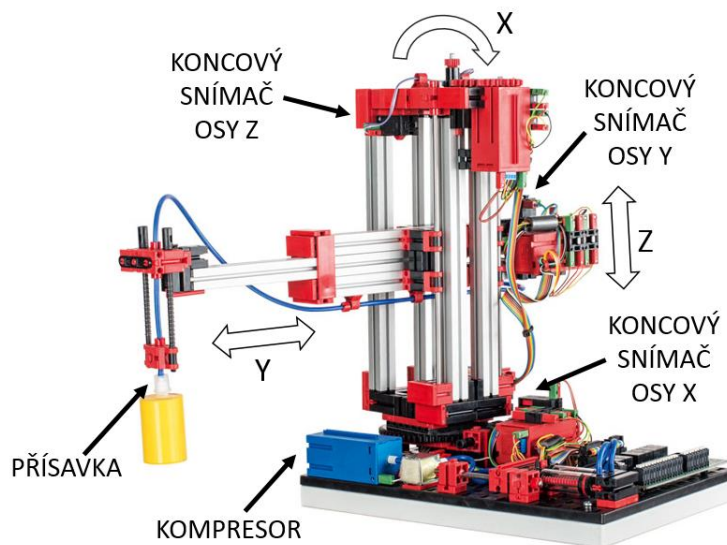
v případě automatizovaného skladu. Výsuv pneumatických motorů je řešen pomocí funkce *Speed control* s názvem “valec_X_dopredu“, kde X značí číslo válce. Jelikož se jedná o jednočinné válce a zpětný pohyb je na stavebnici řešen pomocí pružiny, bylo nutné tento jev vyřešit i v modelu. K jeho simulaci byly vytvořeny další funkce *Speed control* s názvem “valec_X_dozadu“, které jsou propojeny s funkcemi “valec_X_dopredu“ pomocí podmínek v *Runtime expression*, Ty definují aktivitu funkcí “valec_X_dozadu“ pouze v případě, že funkce pro výsuv jsou neaktivní. Pohon dopravníkového pásu a světelné závory byly vytvořeny rovněž stejným způsobem jako u automatizovaného skladu. Novým prvkem je zde snímač barvy, který je založen na principu LED, která vysílá světelný paprsek. V závislosti na barvě projíždějícího dílce se odrazí jiné množství světla zpět, které je následně změřeno. Výstupem tohoto snímače je číselná hodnota proměnné typu Int. Součástí sestavy jsou tři dílce s bílou, modrou a červenou barvou. Jelikož tuto možnost detekce modul MCD nenabízí, byla vymyšlena alternativní varianta čidla, jejíž hodnota výstupu odpovídá reálnému snímači. Zmenšený model světelné závory byl třikrát umístěn na sebe, čímž v podstatě vznikla světelná závora ve třech různých výškách. Dále byly vytvořeny tři dílce o různých výškách tak, aby bílý dílec protínal pouze spodní světelnou závora, červený protínal spodní a prostřední a modrý protínal všechny tři (Obr. 44). Světelné závory byly opět vytvořeny pomocí funkce *Collision Sensor* s názvy “senzor_bila“, “senzor_cervena“ a “senzor_modra“. Pro celkový výstup snímače byl vytvořen signál “barevne_cidlo“ typu Int. Tento signál byl pomocí funkce *Runtime expression* propojen s vytvořenými senzory tak, aby hodnota jeho výstupu byla v případě protnutí pouze spodního 17 000, v případě protnutí spodního a prostředního 20 000, a v případě protnutí všech tří 24 000. Tyto hodnoty odpovídají reálným výstupním hodnotám z barevného snímače při detekci patřičné barvy.



Obr. 44: Simulace světelného snímače

3.4.2. Robot

Stavebnici tvoří 3-osý robot s cylindrickou kinematikou (Obr. 45). Osy jsou dle výrobce označeny X, Y a Z. Osa X je rotační a osy Y a Z jsou posuvné. Všechny tři osy jsou vybaveny enkodérem, díky čemuž lze řídit jejich polohu. Zároveň je v koncových polohách všech os umístěn koncový snímač pro referenci robotu po zapnutí. Pro manipulaci s přiloženými dílci je na ose Y připevněna vakuová přísavka, která je poháněna kompresorem.

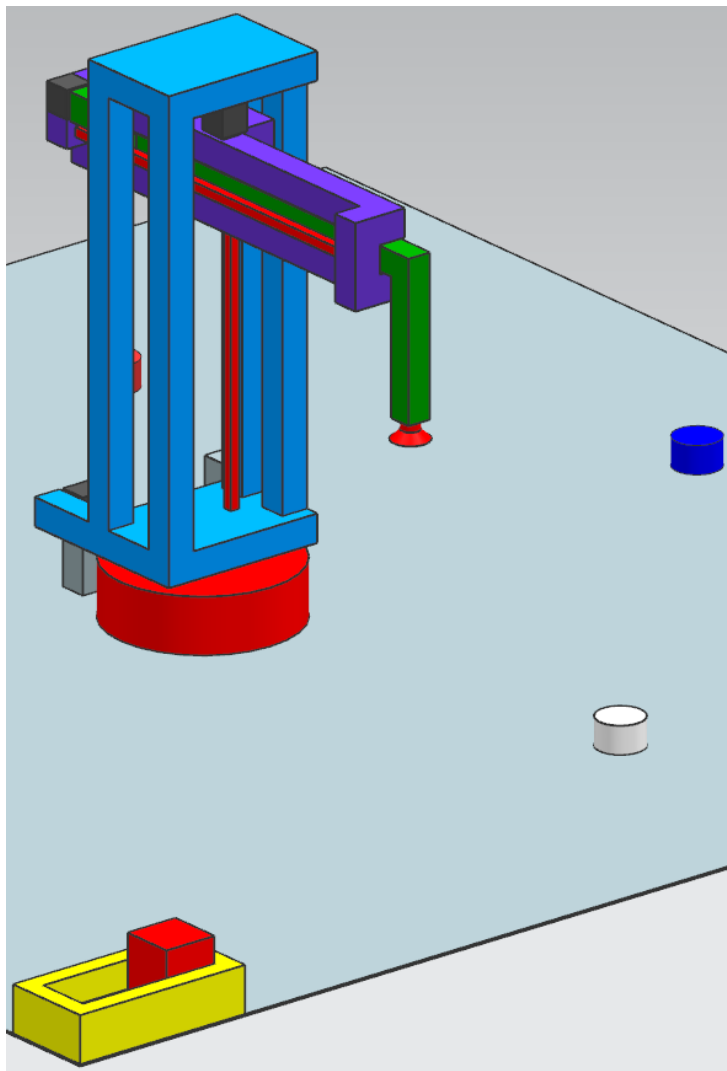


Obr. 45: Stavebnice robotu

Digitální vstupy a výstupy PLC	Popis	Název proměnné
do_0.0	Pohon osy Z nahoru	do_osa_z_nahoru
do_0.1	Pohon osy Z dolů	do_osa_z_dolu
do_0.2	Pohon osy Y vzad	do_osa_y_vzad
do_0.3	Pohon osy Y vpřed	do_osa_y_vpřed
do_0.4	Pohon osy X po směru	do_osa_x_smer
do_0.5	Pohon osy X proti směru	do_osa_x_protismer
do_0.6	Kompresor	do_kompresor
do_0.7	Vauková přísavka	do_vacuum
di_0.0	Koncový snímač osy Z ve směru nahoru	di_snimac_osa_z
di_0.1	Koncový snímač osy Y ve směru vzad	di_snimac_osa_y
di_0.2	Koncový snímač osy X v protisměru	di_snimac_osa_x
di_1.1	Tlačítko TOTAL STOP	di_total_stop

Tabulka 4: Souhrn vstupů a výstupů robotu

V CAD modulu programu NX12 byl vytvořen zjednodušený model robotu (Obr. 46). Byly zde zakomponovány všechny zmíněné funkční prvky a po pracovní ploše byly rozmístěny dílce bílé, modré a červené barvy určené k manipulaci.



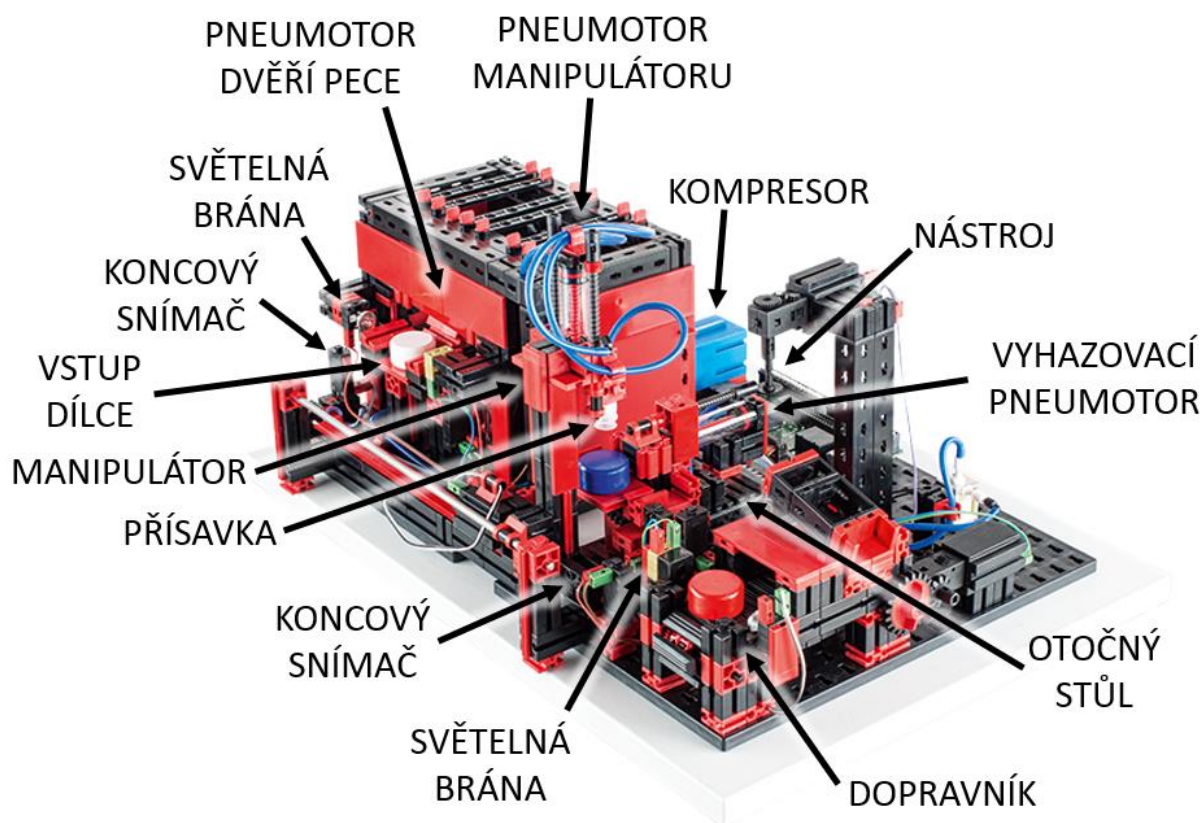
Obr. 46: Virtuální model robotu

V modulu MCD bylo následovně definováno ovládání pohybových os, vakuové přísavky a senzorů. Definice fyzických vlastností a polohových vazeb proběhla stejným způsobem jako v případě automatizovaného skladu. Novým prvkem v sestavě robotu je vakuová přísavka, která slouží k manipulaci s dílci vyvoláním podtlaku v jejich dotykové ploše. K její simulaci byly vytvořeny pomocné funkce *Fixed joint* “obrobek_X_gripper“, kde X značí barvu dílce, spojující paletu a přísavku k sobě. Na všech dílcích byla následně použita funkce *Collision sensor*, která slouží k detekci kolize dílce s přísavkou. Pomocí podmínky zadané do funkce *Runtime expression*, byla definována aktivita vytvořených fixních vazeb pouze pro případ, kdy

jsou proměnné pro pohon přísavky “gripper“ a “kompresor“ hodnoty TRUE a zároveň je spuštěn senzor funkce *Collision sensor* u daného dílce, s kterou má být manipulováno.

3.4.3. Obráběcí pracoviště s pecí

Stavebnice se skládá ze čtyř funkčních celků (Obr. 47). Prvním celkem je model pece, který disponuje jednoosým podavačem říditelným v obou směrech. Podavač je rovněž vybaven koncovými snímači v obou směrech a světelnou bránou u výstupu z pece, pro detekci přítomnosti dílce. Pec je dále vybavena žárovkou pro simulaci teplotního procesu a dveře ovládané pomocí pneumatického válce. Druhým celkem je dvouosý manipulátor s přísavkou a dvěma koncovými snímači, jehož funkcí je přesun dílce z pece k otočnému stolu, což je zároveň třetí funkční prvek. Po založení dílce se otočný stůl otočí o 180°. V této poloze je koncový snímač a model nástroje, který má simulovat obráběcí proces. Otočný stůl se dále může otočit o 90°. V této poloze je rovněž koncový snímač. Stůl je vybaven pneumatickým válcem, který slouží k vyhození dílce na poslední funkční prvek, manipulátor. Ten je vybaven světelnou bránou pro detekci dílce na konci celého procesu.

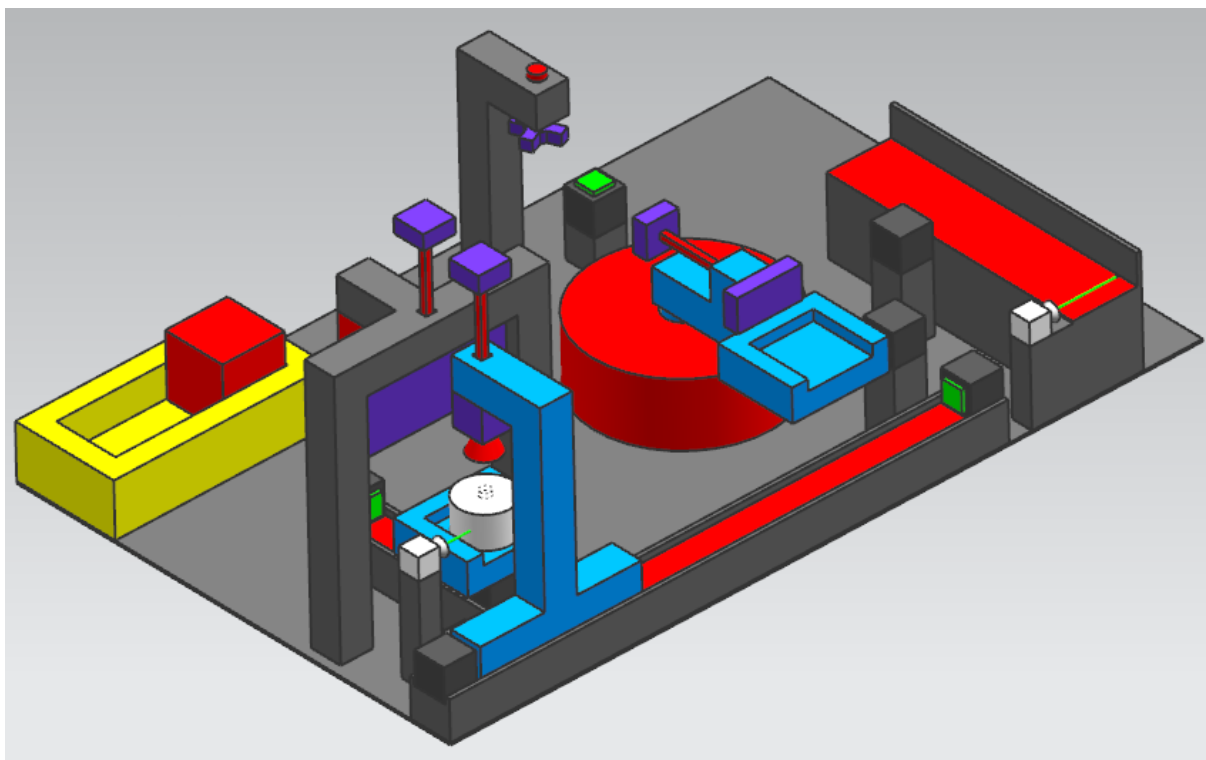


Obr. 47: Stavebnice obráběcího pracoviště s pecí

Digitální vstupy a výstupy PLC	Popis	Název proměnné
do_0.0	Pohon otočného stolu ve směru hodinových ručiček	do_table_cw
do_0.1	Pohon otočného stolu proti směru hodinových ručiček	do_table_ccw
do_0.2	Pohon dopravníku	do_belt
do_0.3	Pohon nástroje	do_motor_saw
do_0.4	Pohon podavače směrem dovnitř pece	do_feeder_in
do_0.5	Pohon podavače směrem ven z pece	do_feeder_out
do_0.6	Pohon manipulátoru doprava	do_manip_right
do_0.7	Pohon manipulátoru doleva	do_manip_left
do_1.0	Světlo v peci	do_oven_light
do_1.1	Kompresor	do_compressor
do_1.2	Přísavka	do_vacuum
do_1.3	Pneumatický válec na manipulátoru	do_lowering
do_1.4	Pneumatický válec dveří pece	do_oven_door
do_1.5	Vyhazovací pneumatický válec	do_belt_feeder
di_0.0	Koncový spínač otočného stolu v poloze u manipulátoru	di_table_ccw
di_0.1	Koncový spínač otočného stolu v poloze u pásového dopravníku	di_table_cw
di_0.2	Světelná závora na dopravníku	di_belt
di_0.3	Koncový spínač otočného stolu v poloze u nástroje	di_table_saw
di_0.4	Koncový spínač manipulátoru v poloze u pece	di_manip_left
di_0.5	Koncový spínač podavače uvnitř pece	di_feeder_in
di_0.6	Koncový spínač podavače vně pece	di_feeder_out
di_0.7	Koncový spínač manipulátoru v poloze u otočného stolu	di_manip_right
di_1.0	Světelná závora u podavače	di_manip
di_1.1	Tlačítko TOTAL STOP	di_total_stop

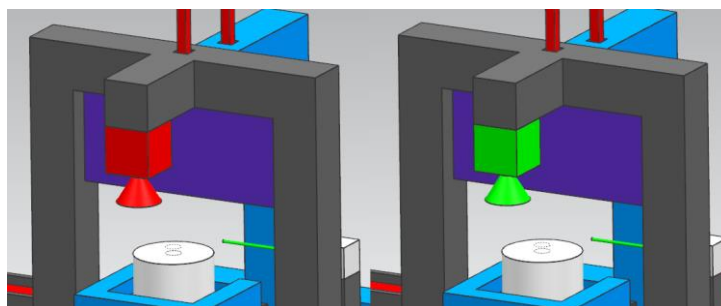
Tabulka 5: Souhrn vstupů a výstupů obráběcího pracoviště s pecí

V CAD modulu programu NX12 byl vytvořen zjednodušený model (Obr. 48). Byly zde zakomponovány všechny zmíněné funkční prvky a do podavače byl umístěn dílec.



Obr. 48: Virtuální model obráběcího pracoviště s pecí

V modulu MCD bylo následovně definováno ovládání pohybových os, vakuové přísavky a senzorů. Definice fyzických vlastností a polohových vazeb proběhla stejným způsobem jako v případě automatizovaného skladu. Novým prvkem v sestavě pracoviště je světlo v peci. Tento prvek byl ve virtuálním modelu vytvořen pomocí funkce *Display changer*. Tato funkce měnit barvu vybraného objektu. Byly vytvořeny funkce s názvem “svetlo_off“, u které byla barva součásti reprezentující světlo v peci změněna na červenou, a “svetlo_on“, u které byla stejná součást přebarvena na zeleno. Následně tyto funkce byly propojeny se signálem “svetlo“ pomocí funkce *Runtime expression* tak, aby v případě hodnoty FALSE signálu “svetlo“ byla aktivní funkce “svetlo_off“ a v případě hodnoty TRUE byla aktivní funkce “svetlo_on“ (Obr. 49).



Obr. 49: Simulace vypnutého a zapnutého světla

4. Závěr

Cílem této diplomové práce bylo vytvořit programovatelné virtuální modely čtyř stavebnic od firmy Fischertechnik za použití metody Software in the Loop. Tyto modely by měly sloužit při výuce předmětu *PLC řízení výrobních strojů a zařízení* k verifikaci navržených PLC programů bez potřeby reálné PLC sestavy nebo samotné stavebnice.

Rešeršní část se zabývá problematikou použité metody Software in the Loop. Jejím principem je virtuální zprovoznění automatizovaného systému za pomoci virtuálního modelu propojeného s virtuálním PLC. Přínosem této metody je možnost verifikace PLC programu v samotném PC bez potřeby dalšího hardwaru. V rešerši byly shrnuty možnosti programu NX Mechatronics concept designer pro tvorbu programovatelných virtuálních modelů. K tvorbě virtuálního PLC byly kladen důraz na možnosti programů TIA Portal a PLCSIM Advanced.

Jako vzor pro kompletní popis tvorby virtuálního modelu byla vybrána stavebnice automatizovaného skladu. V CAD modulu programu NX byl vytvořen zjednodušený model, na kterém byly následně definovány všechny prvky pohonů, enkodérů a dalších snímačů v modulu MCD. Výsledkem je programovatelný virtuální model automatizovaného skladu, jehož vstupy a výstupy korespondují s reálnou stavebnicí. Dle zadání byl vytvořen kompletní řídicí PLC program pro tuto stavebnici, který se sestává z *Automatického režimu* a *Manuálního režimu*. Funkcí *Automatického režimu* je automatizované naskladnění, případně vyskladnění palety. Pomocí vytvořeného modelu a PLC programu bylo uskutečněno virtuální zprovoznění, které ověřilo funkčnost modelu a PLC programu díky shodnému chování s reálnou předlohou.

Na základě dosažených znalostí při tvorbě prvního modelu byly vytvořeny programovatelné virtuální modely zbývajících tří stavebnic třídící linky, robotu a obráběcího pracoviště s pecí. Všechny vytvořené modely byly následně propojeny s prázdnými PLC programy připravenými pro aplikaci ve výuce předmětu *PLC řízení výrobních strojů a zařízení*. Součástí řešení jsou také uživatelské návody pro virtuální zprovoznění jednotlivých modelů.

Výstup této práce v podobě čtyř programovatelných virtuálních modelů s uživatelskými návody a jednoho kompletního PLC programu pro stavebnici automatizované linky splňuje veškeré stanovené cíle.

Seznamy

Seznam použité literatury

- [1] LIU, Zheng, Nico SUCHOLD a Christian DIEDRICH. *Virtual Commissioning of Automated Systems*. b.r. DOI: 10.5772/45730.
- [2] VIRTUAL COMMISSIONING AS A FINAL STEP IN DIGITAL VALIDATION OF THE ROBOTIC MANUFACTURING SYSTEMS. b.r., 2014(94). ISSN 2067-9238.
- [3] DZINIC, Jasmin a Charlie YAO. *Simulation-based verification of PLC programs*. 2013. *Master of Science Thesis in Production Engineering*. CHALMERS UNIVERSITY OF TECHNOLOGY.
- [4] GRIEVES, Michael. *Digital Twin: Manufacturing Excellence through Virtual Factory Replication* A Whitepaper by Dr. Michael Grieves This [online]. b.r. [cit. 2018-02-13]. Dostupné z: http://innovate.fit.edu/plm/documents/doc_mgr/912/1411.0_Digital_Twin_White_Paper_Dr_Grieves.pdf
- [5] Siemens PLM, NX 12.0 and the Road to the Digital Twin [online]. b.r. [cit. 2018-02-13]. Dostupné z: <https://www.engineering.com/PLMERP/ArticleID/16009/Siemens-PLM-NX-120-and-the-Road-to-the-Digital-Twin.aspx>
- [6] *Mechatronics Concept Designer* [online]. b.r. [cit. 2018-02-17]. Dostupné z: http://www.ideal-plm.ru/uEditor/files/4/NXCAD01_16_en_mcd_FS.pdf
- [7] Process Simulate [online]. b.r. [cit. 2018-02-17]. Dostupné z: https://www.plm.automation.siemens.com/en/products/tecnomatix/manufacturing-simulation/assembly/process-simulate.shtml#lightview%26url=/en_us/Images/7457_tcm1023-80351.pdf%26title=Process%26description=Process%26docType=pdf
- [8] MCGREGOR, Ian. *INTRODUCTION TO EMULATE3D – EMULATION, SIMULATION, AND DEMONSTRATION* [online]. b.r. [cit. 2018-02-17]. Dostupné z: <https://pdfs.semanticscholar.org/16c0/0b4daceea34f7290e9440dc613fc6529779b.pdf>

- [9] *PLC* [online]. b.r. [cit. 2018-03-07]. Dostupné z: http://www.e-automatizace.cz/ebooks/ridici_systemy_akcni_cleny/R_PLC.html
- [10] *CO JE OPC ? OPC SERVER ? OPC KLIENT ?* [online]. b.r. [cit. 2018-02-13]. Dostupné z: <https://www.foxon.cz/cs/blogs/80-co-je-opc-opc-server-opc-klient-.html>
- [11] *Datasheet of automated warehouse* [online]. b.r. [cit. 2018-04-23]. Dostupné z: <https://content.ugfischer.com/cbfiles/fischer/Zulassungen/ft/153422-Encodermotor9V.pdf>
- [12] KOSEK, R. *Digitalizace, projektování a nové trendy v automatizaci* [online]. b.r. [cit. 2018-02-17]. Dostupné z: <http://w5.siemens.com/web/cz/cz/corporate/portal/home/microsites/seminar-eplan/Documents/01-siemens-digitalizace-a-nove-trendy-v-automatizaci.pdf>
- [13] *Automated High-Bay Warehouse*. [online]. b.r. [cit. 2018-03-12]. Dostupné z: <https://i.ytimg.com/vi/CJmgqP0HEAo/maxresdefault.jpg>

Seznam obrázků

Obr. 1: Podíl zprovoznění na délce projektu [1]	10
Obr. 2: Cena nápravy problému v závislosti na fázi projektu [1]	11
Obr. 3: Spolupráce konstruktéra a programátora na virtuálním modelu [2]	11
Obr. 4: Schéma metody Hardware in Loop	13
Obr. 5: Schéma metody Software in Loop	13
Obr. 6: Portfolio firmy Siemens pro simulační software [12].....	16
Obr. 7: Schéma propojení virtuálního PLC s modelem	19
Obr. 8: Propojení bez OPC serveru [10]	19
Obr. 9: Propojení s OPC serverem [10]	19
Obr. 10: Export signálů z NX MCD do SIMIT	20
Obr. 11: Třídící linka.....	22
Obr. 12: Obráběcí stanoviště s pecí.....	22
Obr. 13: Robot.....	22
Obr. 14: Automatizovaný sklad [13].....	22
Obr. 15: Koncový snímač na ose Y	23
Obr. 16: Koncový snímač na ose Z	23

Obr. 17: Manipulátor.....	23
Obr. 18: Koncové snímače na ose X	24
Obr. 19: Světelná závora na dopravníku	24
Obr. 20: 3D model automatizovaného skladu	25
Obr. 21: Tuhá tělesa v 3D modelu	26
Obr. 22: Umístění světelných závor na dopravníku	27
Obr. 23: Koncové snímače na osách X a Z	27
Obr. 24: Koncový snímač na ose Y	28
Obr. 25: Runime expression pro signal "y_minus"	29
Obr. 26: Vytvořené signály pro PLC	30
Obr. 27: Konfigurace virtuální sestavy v programu TIA Portal	31
Obr. 28: Základní stavy PLC.....	33
Obr. 29: Stavový diagram automatického režimu.....	33
Obr. 30: Funkční blok Automatic_move_Y_Z	36
Obr. 31: Stavový diagram instanci funkčního bloku Position pro naskladnění palety	36
Obr. 32: Data bloky zpracovávající signál z enkodérů v osách Y a Z	37
Obr. 33: Úvodní obrazovka ovládacího panelu.....	38
Obr. 34: Obrazovka pro manuální režim.....	39
Obr. 35: Obrzovka pro automatický režim.....	40
Obr. 36: Dialog programu PLCSim Advanced	41
Obr. 37: Dialog pro nahrání řídicího programu do zařízení.....	42
Obr. 38: Dialog funkce External signal configuration	43
Obr. 39: Propojení signálu modelu s virtuálním PLC	44
Obr. 40: Dialog Runtime expression funkce "x_zero"	46
Obr. 41: Dialog funkce Mechatronics Concept Designer Preferences.....	46
Obr. 42: Stavebnice třídící linky	48
Obr. 43: Virtuální model třídící linky	49
Obr. 44: Simulace světelného snímače	50
Obr. 45: Stavebnice robotu.....	51
Obr. 46: Virtuální model robotu.....	52
Obr. 47: Stavebnice obráběcího pracoviště s pecí.....	53
Obr. 48: Virtuální model obráběcího pracoviště s pecí.....	55
Obr. 49: Simulace vypnutého a zapnutého světla	55



Seznam tabulek

<i>Tabulka 1: Souhrn vstupů a výstupů automatizovaného skladu</i>	24
<i>Tabulka 2: Hodnoty funkce CASE pro operace naskladnění/vyskladnění palety z určené pozice.....</i>	34
<i>Tabulka 3: Souhrn vstupů a výstupů třídící linky.....</i>	49
<i>Tabulka 4: Souhrn vstupů a výstupů robotu.....</i>	51
<i>Tabulka 5: Souhrn vstupů a výstupů obráběcího pracoviště s pecí</i>	54

Seznam použitých zkratk

Zkratka	Význam
AI	Analog input
AQ	Analog output
CAD	Computer aided design
CPU	Central processing unit
DI	Digital input
DInt	Double integer
DQ	Digital output
HMI	Human machine interface
Int	Integer
MCD	Mechatronics concept designer
OB	Organization block
OPC	OLE for proces control
PLC	Programmable logic controller

Seznam příloh

Textové

Příloha č.1 – Uživatelský návod virtuálního zprovoznění automatizovaného skladu

Příloha č.2 – Uživatelský návod virtuálního zprovoznění třídící linky

Příloha č.3 – Uživatelský návod virtuálního zprovoznění robotu

Příloha č.4 – Uživatelský návod virtuálního zprovoznění obráběcího pracoviště s pecí

Elektronické

Příloha č.5 – Složka “1_Automatizovany_sklad“ s uživatelským návodem, virtuálním modelem a připraveným PLC programem – Automatizovaný sklad

Příloha č.6 – Složka “2_Tridici_linka“ s uživatelským návodem, virtuálním modelem a připraveným PLC programem – Robot

Příloha č.7 – Složka “3_Robot“ s uživatelským návodem, virtuálním modelem a připraveným PLC programem – Třídící linka

Příloha č.8 – Složka “4_Pec“ s uživatelským návodem, virtuálním modelem a připraveným PLC programem – Obráběcí pracoviště s pecí

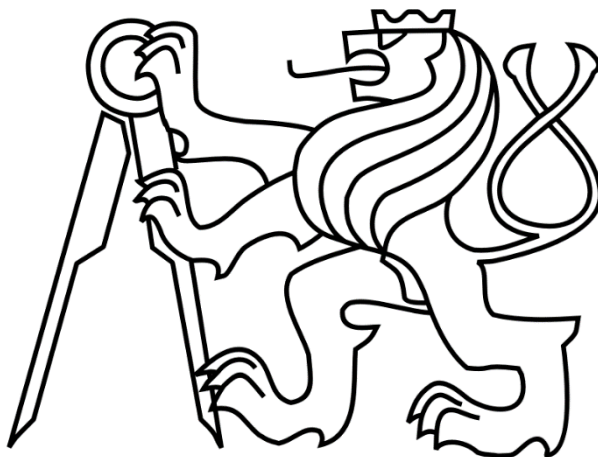
Příloha č.9 – Složka “5_PLC_pro_Automatizovany_sklad“ s řídicím programem pro PLC automatizovaného skladu

Příloha č.10 – Diplomová práce v PDF

Přílohy s čísly 5. až 10. budou z důvodu značné datové velikosti nahrány pouze na přiložené CD.

FAKULTA STROJNÍ

Ústav výrobních strojů a zařízení



Diplomová práce

Přílohy