

Digitální dvojče paletových posuvných regálů

Bc. Adam Gajdošík

Diplomová práce
2024



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2023/2024

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Bc. Adam Gajdošík
Osobní číslo: A23566
Studijní program: N0714A150006 Automatické řízení a informatika v průmyslu 4.0
Forma studia: Kombinovaná
Téma práce: Digitální dvojče paletových posuvných regálů
Téma práce anglicky: Digital Twin of Mobile Pallet Racks

Zásady pro vypracování

1. Vypracujte literární rešerši na dané téma.
2. Připravte v prostředí RobotStudio podklady pro vytvoření digitálního dvojčete daného problému.
3. Navrhněte a zimplementujte komunikační protokol OPC UA mezi vytvořeným dvojčetem a simulovaným PLC B&R připraveným firmou SSI Schäfer.
4. Implementujte logiku výměny signálů a vykonávání příkazů z PLC se všemi potřebnými zpětnými vazbami k simulování kompletního regálového systému.
5. Vytvořte modulární komponenty pro automatické sestavení regálového systému podle parametrů z konfiguračního souboru.
6. Aplikaci otestujte a vytvořte testovací postup pro rychlou kontrolu funkčnosti celého systému.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. RobotStudio Operating Manual. 3HAC032104-001. ABB, 2023.
2. SEGOVIA, Mariana a GARCIA-ALFARO, Joaquin. Design, Modeling and Implementation of Digital Twins. Online. Sensors. 2022, roč. 22, č. 14. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s22145396>.
3. BARRICELLI, Barbara Rita; CASIRAGHI, Elena a FOGLI, Daniela. A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. Online. IEEE Access. 2019, roč. 7, s. 167653-167671. ISSN 2169-3536. Dostupné z: <https://doi.org/10.1109/ACCESS.2019.2953499>.
4. DRAHOS, Peter; KUCERA, Erik; HAFFNER, Oto a KLIMO, Ivan. Trends in industrial communication and OPC UA. Online. In: 2018 Cybernetics & Informatics (K&I). IEEE, 2018, s. 1-5. ISBN 978-1-5386-4419-5. Dostupné z: <https://doi.org/10.1109/CYBERI.2018.8337560>.
5. HENSEL, Stephan; GRAUBE, Markus; URBAS, Leon; HEINZERLING, Till a OPPELT, Mathias. Co-simulation with OPC UA. Online. In: 2016 IEEE 14th International Conference on Industrial Informatics (INDIN). IEEE, 2016, s. 20-25. ISBN 978-1-5090-2870-2. Dostupné z: <https://doi.org/10.1109/INDIN.2016.7819127>.

Vedoucí diplomové práce: **Ing. Ľuboš Spaček, Ph.D.**
Ústav řízení procesů

Datum zadání diplomové práce: **8. prosince 2023**
Termín odevzdání diplomové práce: **27. května 2024**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 8. prosince 2023

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 02.05.2024

Bc. Adam Gajdošík v.r.
.....
podpis studenta

ABSTRAKT

V práci je vytvořeno digitální dvojče paletových posuvných regálů pro společnost SSI SCHÄFER v programovém prostředí RobotStudio s využitím komunikačního protokolu OPC UA pro simulované PLC. Práce začíná literární rešerší na dané téma, následuje hlavním cílem práce, což je vytvořit digitální dvojče regálového systému s logikou výměny signálů a vykonávání příkazů z PLC se zpětnými vazbami. Na závěr je vytvořen testovací postup pro rychlou kontrolu funkčnosti celého systému.

Klíčová slova: RobotStudio, Posuvný regálový systém, PLC, OPC UA

ABSTRACT

The thesis presents the creation of a digital twin of pallet racking systems for SSI SCHÄFER within the RobotStudio programming environment, utilizing OPC UA communication protocol for simulated PLCs. It commences with a literature review on the topic followed by the primary objective of developing the digital twin of the racking system with signal exchange logic and command execution from PLCs with appropriate feedback. Finally, a testing procedure is devised for rapid functionality checks of the entire system.

Keywords: RobotStudio, pallet racking systém, PLC, OPC UA

Poděkování

Chtěl bych velmi poděkovat panu Ing. Ľubošovi Spačkovi Ph.D. za trpělivost při vedení této práce, jeho připomínky a odborné rady. Dále bych chtěl velmi poděkovat panu Ing. Ondřeji Sochorovi za odborné připomínky, rady a znalosti při tvorbě této práce. Velké poděkování patří taktěž mé rodině a přátelům za velkou podporu po dobu celého studia.

Motto

“Engineering is not only study of 45 subjects but it is moral studies of intellectual life.”

--Prakhar Srivastav--

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 DIGITÁLNÍ DVOJČE	10
1.1 ZÁKLADNÍ TYPY DIGITÁLNÍHO DVOJČETE.....	10
1.2 POUŽITÍ DIGITÁLNÍCH DVOJČAT.....	12
2 POSUVNÉ REGÁLY	14
2.1 ROZDĚLENÍ POSUVNÝCH REGÁLŮ.....	14
3 OPC UA	15
3.1 PRINCIP OPC UA.....	15
3.1.1 Model OPC UA.....	16
3.1.2 OPC UA služby.....	17
3.2 PŘÍKLADY ARCHITEKTURY OPC UA KLIENT – SERVER.....	21
4 ROBOTSTUDIO	23
4.1 SMART COMPONENTY.....	23
4.1.1 Vlastnosti.....	24
5 PLC	26
5.1 CYKLUS PLC.....	26
5.2 SOUČÁSTI PLC.....	27
5.3 PROGRAMOVÁNÍ PLC.....	28
II PRAKTICKÁ ČÁST	31
6 REGÁLOVÝ SYSTÉM OD SPOLEČNOSTI SSI SCHÄFER	32
6.1 POPIS.....	32
6.1.1 Klíčový spínač provozních režimů (NOUZOVÁ JÍZDA).....	33
6.1.2 Obslužná jednotka.....	34
6.1.3 Obslužný panel.....	34
6.2 BEZPEČNOSTNÍ SVĚTELNÉ ZÁVORY A SENZORY.....	35
6.2.1 Přístupová světelná závora.....	35
6.2.2 Bezpečnostní nožní světelná závora.....	36
6.2.3 Koncový spínač.....	36
7 PŮVODNÍ STAV	37
8 LOGIKA A FUNKCE REGÁLOVÉHO SYSTÉMU	38
8.1 OVLÁDÁNÍ.....	38
8.2 POHYB.....	39
8.3 BEZPEČNOST.....	42
9 SESTAVENÍ REGÁLOVÉHO SYSTÉMU	43
9.1 3D MODEL Y.....	43
9.2 SMART COMPONENTY PRO GENEROVÁNÍ REGÁLOVÉHO SYSTÉMU.....	44
9.3 SMART COMPONENTY PRO PŘEPÍNÁNÍ REŽIMŮ.....	45
10 SIMULACE VOZÍKU	48

10.1	SENZORY VOZÍKU	48
10.2	TLAČÍTKA.....	51
10.3	POHYB.....	54
10.4	PRŮBĚH OTEVÍRÁNÍ ULIČKY	58
11	PROPOJENÍ S PLC.....	59
	ZÁVĚR	61
	SEZNAM POUŽITÉ LITERATURY.....	62
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	65
	SEZNAM OBRÁZKŮ	66
	SEZNAM PŘÍLOH.....	70

ÚVOD

Digitální dvojčata se jistě stávají nedílnou součástí při vývoji či návrhu produktů nebo celých systémů. V dnešní době se běžně vytváří digitální dvojčata jakož to součást vývoje zejména v průmyslu a logistice. Využití digitálních dvojčat překračuje pouhé replikování fyzických procesů nebo produktů v digitálním prostoru. Jedná se o modely, které sbírají data a umožňují simulace, analýzy a optimalizace v reálném čase.

Jedním z klíčových prvků úspěchu digitálních dvojčat je integrace s širokým spektrem technologií, internetu věcí či rozšířené reality. Díky digitálním dvojčatům lze testovat různé scénáře bez rizika poškození, předvídat možné poruchy nebo optimalizovat výkon reálných zařízení či výrobních linek.

Výsledkem práce je návrh digitálního dvojčete posuvných paletových regálů, které vyrábí společnost SSI SCHÄFER. Digitální dvojče bylo vytvořeno v programovém prostředí RobotStudio od společnosti ABB a je ovládáno skrze PLC simulátor, který dodala firma SSI SCHÄFER.

Teoretická část práce se v kapitole 1 zabývá definicí, základními typy a použitím digitálních dvojčat. Kapitola 2 definuje posuvné regály a jejich rozdělení. Kapitola 3 se zabývá architekturou OPC UA, vysvětluje princip komunikace, definuje model a služby a na závěr jsou zde popsány příklady zapojení či architektury mezi serverem a klientem. Kapitola 4 popisuje RobotStudio, jakožto simulační programové prostředí, kde bylo digitální dvojče vytvořeno, a Smart Componenty, ze kterých byla složena celá logika a interakce s digitálním prostředím. Kapitola 5 se poté zabývá PLC, kde popisuje obecný logický programovatelný automat, princip cyklu PLC, z čeho se skládá a jak se dá PLC naprogramovat.

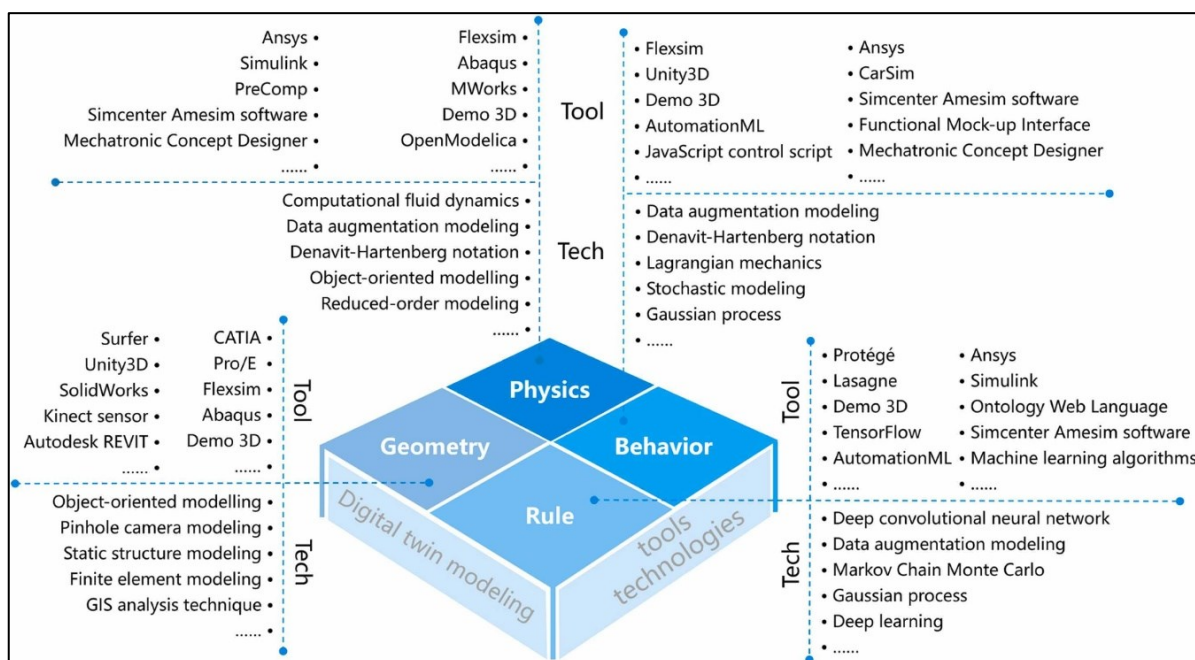
Praktická část práce začíná od kapitoly 6 popisem regálového systému od společnosti SSI SCHÄFER, kde lze nalézt popis klíčových částí a chování regálového systému. Kapitola 7 se zabývá původním prostředkům pro simulaci regálového systému. Následuje kapitola 8, kde je popsána logika a funkce regálového systému. Kapitola 9 popisuje jakým způsobem se automaticky vytvoří dle parametrů regálový systém v simulačním prostředí. Další Kapitola 10 se zabývá popisem Smart Componentu vozíku, kde jsou obsaženy všechny důležité senzory, simulované frekvenční měniče a pohyb v simulačním prostředí. Praktickou část uzavírá kapitola 11, která se věnuje komunikaci mezi PLC a RobotStudiem skrze OPC UA protokol.

I. TEORETICKÁ ČÁST

1 DIGITÁLNÍ DVOJČE

Digitální dvojčata jsou virtuální reprezentace fyzických objektů, systémů nebo procesů. Používají se, jako most mezi reálným světem a digitálním prostředím. Tyto dvojčata se využívají například v průmyslu, zdravotnictví, městském plánování a energetice. Jsou založena na datové integraci, analýze a prediktivních schopnostech. Jejich vývoj je spojen s pokroky v IoT (Internet of Things), analýze dat a simulačních technologiích. První zmínka konceptu digitálního dvojčete byla představena Michaelem Grievesem v roce 2002. [1][2][3]

Digitální dvojče je virtuální reprezentací objektu nebo systému, která zahrnuje celý jeho životní cyklus, je aktualizována na základě dat v reálném čase a využívá simulace, strojového učení a logického úsudku k podpoře rozhodování. [1]



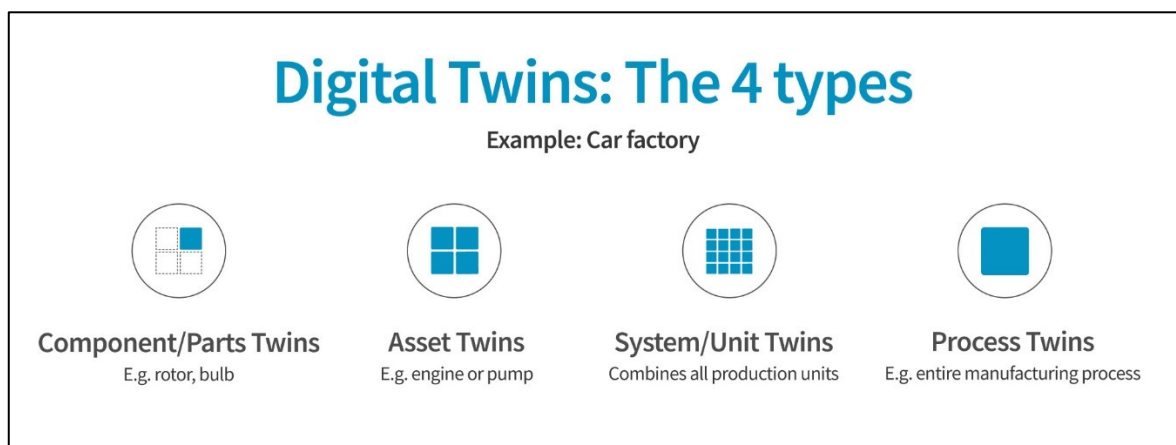
Obrázek 1: Rámec technologií a nástrojů v konstrukci modelů [4]

1.1 Základní typy digitálního dvojčete

Existují různé způsoby, jak kategorizovat digitální dvojčata, nejběžnější typy a metody jsou:

- **Komponentní neboli částečná dvojčata:** Jedná se o digitální repliky jednotlivých součástí v systému nebo produktu, které poskytují možnosti pro zajištění odolnosti v reálném použití. Pomocí těchto digitálních modelů mohou návrháři a inženýři simulovat dynamické scénáře, se kterými se mohou jednotlivé díly během používání setkat. [5][6]

- **Dvojčata aktiv neboli dvojčata produktů:** Tento typ poskytuje jedinečný způsob, jak vizualizovat provoz fyzických produktů a jejich různých součástí. Díky virtuálním inkarnacím reálných produktů umožňují dvojčata aktiv odborníkům v mnoha oborech, jako inženýrství či údržba, získat přehled o tom, jak systémy fungují společně. [5][6]
- **Systémová dvojčata:** Poskytují virtuální reprezentaci systémů složených z mnoha vzájemně propojených produktů. Modelováním jednotlivých produktů a jejich interakcí jako součásti komplexního systému je možné identifikovat příležitosti pro zvýšení efektivity a produktivity. Systémová dvojčata fungují jako objektív, jehož prostřednictvím mohou organizace získat přehled o vztazích mezi produkty a poptávkou zákazníků, což jim umožňuje optimalizovat procesy pro dosažení maximální výkonnosti. [5][6]
- **Procesní dvojčata:** Pomocí procesních dvojčat lze modelovat všechny úrovně operací, od velkých systémů až po malé detaily. Podniky tak mohou získat prediktivní náhled na potenciální problémy, vypracovat doporučení a akční plány na základě aktuálních vzorců nebo předpokládaných scénářů a optimalizovat výrobní modely pro zajištění maximální efektivity. [5][6]



Obrázek 2: Typy digitálních dvojčat [7]

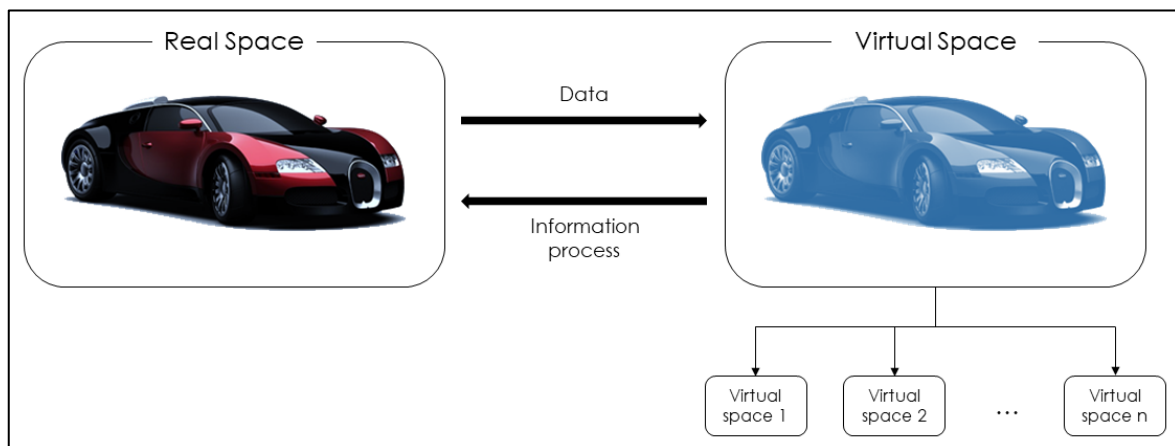
1.2 Použití digitálních dvojčat

Počáteční nasazení digitálních dvojčat bylo většinou zaměřeno na návrh a údržbu vysoce hodnotného a fyzicky velkého zařízení, jako jsou letadla, raketoplány, budovy, mosty a elektrárny, kde mechanické selhání může být ohrožením života nebo způsobit finanční ztráty, které převyšují významné náklady a úsilí spojené s vývojem digitálního dvojčete. [5][6]

Následující odvětví vykazují největší aktivitu v plánování nebo nasazení digitálních dvojčat:

- **Výroba:** Průmyslový svět je široce uznáván jako průkopník v používání digitálních dvojčat. Výrobci již několik let vytvářejí digitální dvojčata dílů, výrobků a systémů a čím dál více nasazují procesní dvojčata, která modelují výrobní procesy a někdy i celé továrny. [5][6]
- **Veřejné služby a energetika:** Dnes již společnosti používají digitální dvojčata k návrhu, sledování a údržbě elektráren, elektrických sítí, přenosu a spotřeby. Technologie by také mohla pomoci zlepšit účinnost obnovitelných energetických systémů, jako jsou solární instalace a větrné elektrárny, jejichž produkce je méně předvídatelná než u elektráren spalujících fosilní paliva. [5][6]
- **Zdravotnictví:** Digitální dvojčata postavená na elektronických zdravotnických záznamech, lékařských obrazech, sekvenování genomu a dalších lékařských informacích by mohla usnadnit poskytovatelům diagnostiku nemocí a doporučit léčby porovnáním digitálního dvojčete pacienta s pacienty s podobnými profily. Lékařské testování by mohlo být efektivnější tím, že by se vyhnulo rizikům spojeným s použitím skutečných pacientů. [5][6]
- **Urbanistické plánování a výstavba:** Digitální dvojčata se používají při navrhování velkých budov a ropných plošin na moři. Je možné rozšířit stavební dvojčata tak, aby zahrnovala i čtvrti a města se zaměřením na infrastrukturu. Velká Británie se dokonce pokusila vyvinout národní digitální dvojče, ale v roce 2022 tuto pětiletou iniciativu ukončila. Digitální dvojčata hrají roli také v iniciativách chytrých měst, jejichž cílem je digitálně propojit infrastrukturu, často prostřednictvím internetu věcí, a na data aplikovat umělou inteligenci a analytiku, což mimo jiné zefektivňuje dopravu a šetří energii. [5][6]
- **Automobilový průmysl:** Digitální dvojčata hrají svou obvyklou roli ve fázi návrhu výrobku, ale také v pozdějších fázích životního cyklu vozidla, jako je například servis. Výrobci automobilů využívají digitální dvojčata také k zefektivnění montážních

závodů. Očekává se, že digitální dvojčata sníží počet masivních svolávacích akcí, protože umožní analyzovat jedinečné dvojče každého vozidla z hlediska závad. [5][6]



Obrázek 3: Model digitálního dvojčete [8]

2 POSUVNÉ REGÁLY

Posuvné regály jsou systémy, ve kterých jsou regály umístěny na podvozcích nebo mobilních základnách vedených po pevných kolejničích na podlaze. Jsou navrženy tak, aby minimalizovaly potřebu prostoru mezi jednotlivými regály. Místo toho se regály posouvají po kolejničích, což umožňuje uložení více materiálu na stejné ploše. [9][10]



Obrázek 4: Posuvné paletové regály [9]

2.1 Rozdělení posuvných regálů

Posuvné regály lze dělit podle způsobu posunu:

- **Posuvné regály s manuálním posuvem:** Regály se posouvají ručně nebo pomocí mechanických systémů, jako jsou kliky nebo řetězy. Tento typ je vhodný pro menší systémy s nižší frekvencí manipulace a nízkou hmotností. [11]



Obrázek 5: Posuvný regál s manuálním posuvem [11]

- **Posuvné regály s automatickým posuvem:** Regály se posouvají pomocí elektromotorů. Tento typ je vhodný pro frekventovanou manipulaci s těžkými předměty. [9][10]

3 OPC UA

OPC Unified Architecture (UA), vydána v roce 2006, je platformově nezávislá architektura orientovaná na služby, která integruje všechny funkce jednotlivých specifikací OPC Classic do jednoho rozšiřitelného rámce. Jedná se o průmyslový M2M (machine-to-machine) komunikační standard. Uživatelé (např. systémoví integrátoři) mohou díky OPC začlenit do svých projektů hardware a software různých výrobců bez ohledu na komunikační rozhraní těchto komponent. Jedinou podmínkou je existence OPC rozhraní pro obě strany neboli existence příslušného OPC serveru pro použitý hardware a rozhraní OPC klient u použitého software. [12][13][14][15]

Na rozdíl od původní verze OPC, která je založena na technologii COM/DCOM od firmy Microsoft (a tudíž funguje pouze pod OS Windows), jde o technologii založenou na obecných komunikačních standardech jako jsou TCP/IP, HTTP a SOAP. To znamená, že OPC UA může být provozován na různých platformách, nejen na Windows. Komunikaci OPC UA je možné integrovat do PLC automatů a dalších zařízení. Na rozdíl od OPC Classic, který definuje přístup k procesním datům (OPC DA), alarmům (OPC AE) a historickým datům (OPC HDA) odděleně, nové OPC UA nedefinuje tyto specifické přístupy, ale pouze formát předávaných zpráv. To znamená, že jeden standard OPC UA umožňuje přenos procesních dat, alarmů a historických dat. Objektová orientace, rozšiřitelnost a bezproblémový přístup k datům, instancí i metadatům jej činí velmi vhodným pro flexibilní produkční prostředí. [13][14][16]

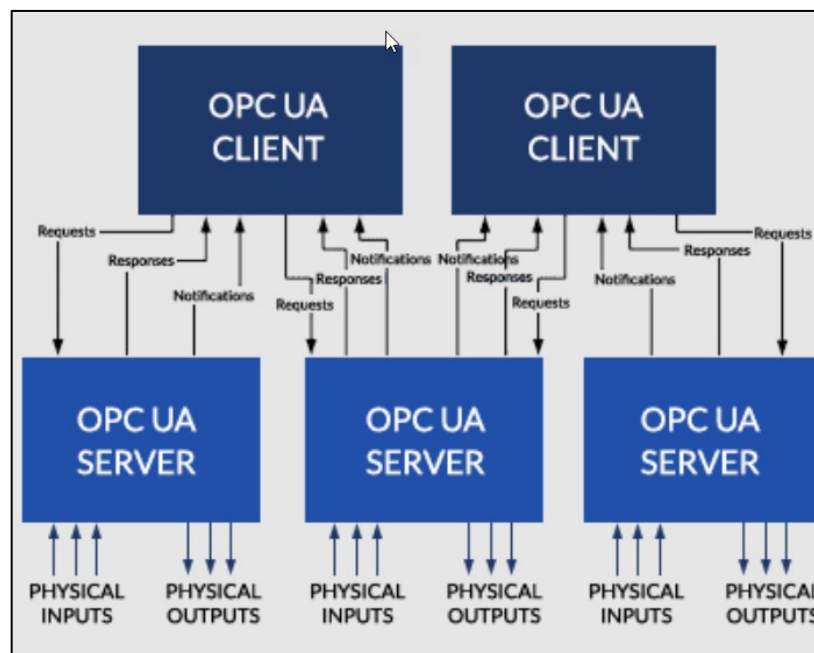


Obrázek 6: OPC UA logo [17]

3.1 Princip OPC UA

Specifikace OPC UA je založená na přenosu dat mezi klientem a serverem, který je založen na mapování a navazování spojení, zabezpečení komunikace a struktury odeslaných dat. Protokol OPC UA je specifikován jako architektura orientovaná na služby (SOA), což znamená, že jsou definovány služby, na které se klient může dotazovat a server na každý dotaz reaguje odpovídající odpovědí. Služby, které server poskytuje, tvoří abstraktní komunikační model.

Po navázání fyzického spojení se pomocí služeb vytvoří a udržuje zabezpečený kanál (SecuredChannel) a relace (Session). Pro veškerou komunikaci je nutné mít aktivní zabezpečený kanál, pro dotazování klienta na služby serveru je nutné mít vytvořenou relaci. Obě tyto části komunikace jsou často označovány jako komunikační zásobník (Stack). Popis adresního prostoru se provádí pomocí uzlů. [17][18][19]



Obrázek 7: Princip komunikace OPC UA [18]

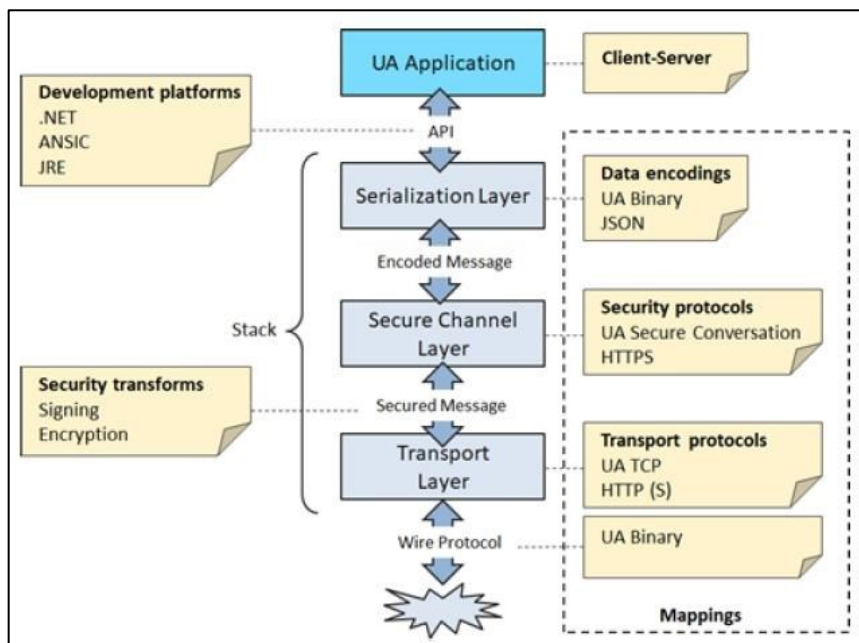
3.1.1 Model OPC UA

Komunikaci v rámci OPC UA zajišťuje komunikační model/zásobník, který se skládá z transportní, komunikační a aplikační vrstvy:

- **Transportní vrstva (Transport Layer):** Obstarává samotné odesílání a příjem zpráv (paketů), tedy vytváří a obsluhuje komunikační protokol. Používá šifrovací a ověřovací mechanismy k zajištění bezpečnosti zprávy poslané po síti, aby nemohla být zpráva přečtena ani modifikována třetí stranou. Při navazování spojení je transportní vrstva vytvořena ihned po připojení. OPC UA podporuje protokoly: HTTP/SOAP, HTTPS a TCP/IP. [17]
- **Komunikační vrstva (Secure Channel + Serialization Layer):** Představuje zabezpečený kanál (SecuredChannel) mezi serverem a klientem, který musí být vytvořen ihned po navázání komunikace. Způsob vytvoření kanálu závisí na použitém komunikačním protokolu. Pro více připojení je třeba použít více kanálů, na každém kanále

lze identifikovat, kterému partneru patří. Po vytvoření zabezpečeného kanálu je vytvořen identifikátor kanálu (Secure-ChannelId) a bezpečnostní známka (Security-Token), kterými se dále kanál prokazuje. Zatímco identifikátor je trvalý, známka má omezenou životnost a je třeba ji v pravidelných intervalech obměňovat. Po vypršení životnosti poslední známky se spojení ukončí. [17]

- **Aplikační vrstva (Application Layer):** Představuje relace (Session), které slouží k provádění volání a zpracování služeb. Relace se používá pro vnitřní identifikaci komunikace a může poskytovat také autorizaci. Při vytváření relace klient předá serveru přihlašovací údaje, a po jejich ověření server označí klienta jako specifického uživatele s příslušnými oprávněními pro provedení určitých akcí. OPC UA neupřesňuje formát uživatelů, pouze specifikuje způsob předání přihlašovacích údajů. Každý zabezpečený kanál může obsahovat maximálně jednu relaci, která musí být vytvořena a následně aktivována. Aktivací se relace spojí s aktivním zabezpečeným kanálem. Relace se ukončí po uplynutí předem stanovené doby nečinnosti. Pokud klient nepošílá žádné požadavky, musí pravidelně zasílat dotazy, aby udržel relaci aktivní. [17]



Obrázek 8: OPC UA model [17]

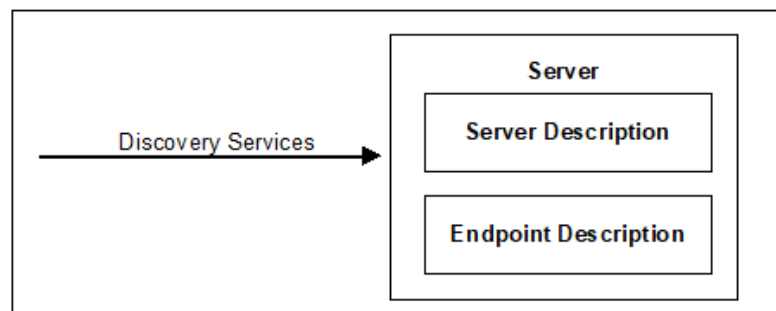
3.1.2 OPC UA služby

Komunikace mezi klientem a serverem v prostředí OPC UA je realizována výhradně pomocí volání a zpracování služeb (Services), které se starají o řízení jednotlivých funkcí serveru

OPC UA. Dotazy i odpovědi jsou doprovázeny společnými hlavičkami, které umožňují klientovi například nastavit požadované informace, jež má server vrátit. [20]

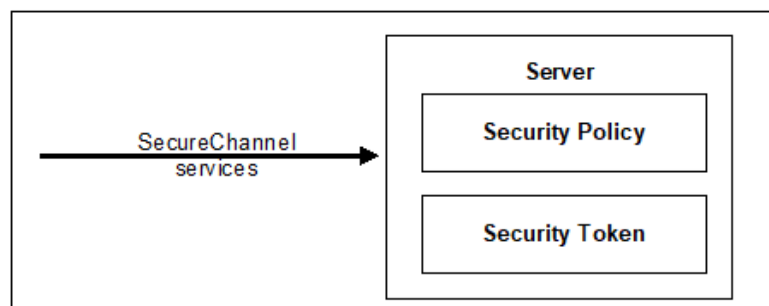
Server v odpovědích specifikuje stavový kód provedení požadavku, který indikuje, zda byla služba úspěšně vykonána. Všechny stavové kódy jsou definovány standardem OPC UA a rozdělují se do kategorií dobrých (Good), což signalizuje úspěšné provedení služby, nejistých (Uncertain) a špatných (Bad), které naznačují selhání volání služby. Každá služba stanovuje možné stavové kódy, které může vrátit. Chyby se mohou týkat například přístupových práv, stavu serveru, stavu cílového objektu nebo nesprávně provedených akcí. Nejistý výsledek může nastat, pokud alespoň jedna z částečných akcí byla ukončena chybou. [20]

- **Sada služeb Discovery** definuje služby, které umožňují klientovi hledat koncové body implementované serverem a číst konfiguraci zabezpečení pro každý z těchto koncových bodů. [20]



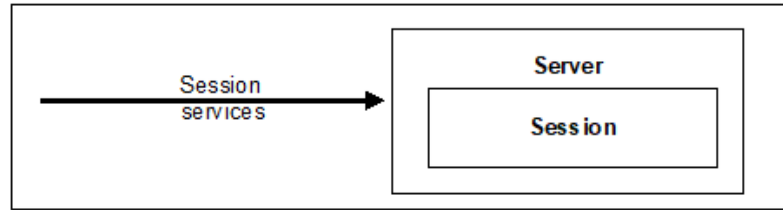
Obrázek 9: Služby zjišťování [20]

- **Sada služeb SecureChannel** definuje služby, které umožňují klientovi vytvořit komunikační kanál pro zajištění důvěrnosti a integrity zpráv vyměňovaných se serverem. [20]



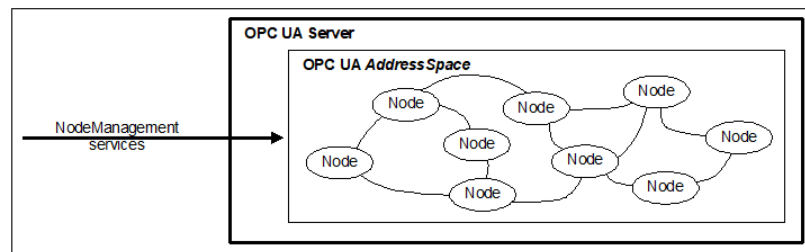
Obrázek 10: Služby SecureChannel [20]

- **Sada služeb Session** definuje služby, které klientovi umožňují ověřit uživatele, jehož jménem jedná, a spravovat relace. [20]



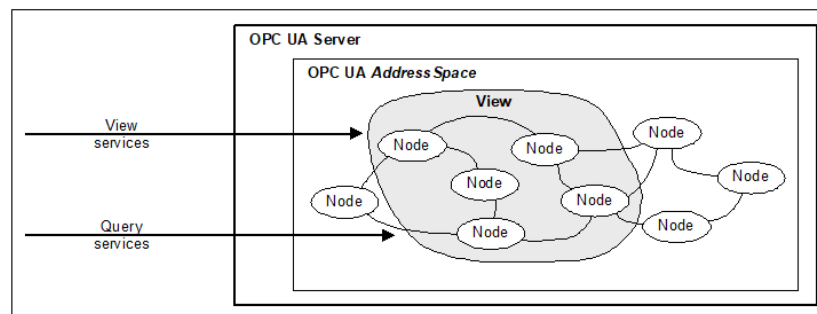
Obrázek 11: Služby relací [20]

- **Sada služeb NodeManagement** definuje služby, které klientovi umožňují přidávat, upravovat a odstraňovat uzly v adresním prostoru. [20]



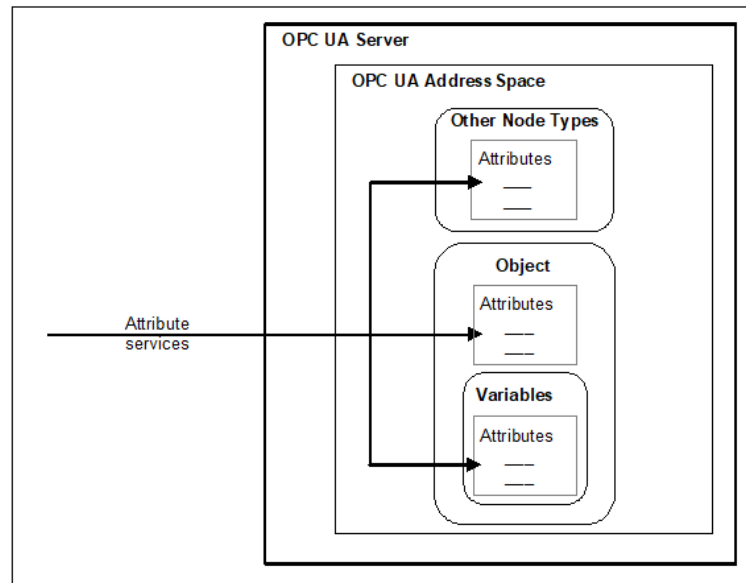
Obrázek 12: Služby NodeManagement [20]

- **Sada služeb View** definuje služby, které klientům umožňují procházet adresní prostor nebo jeho podmnožiny zvané zobrazení. **Sada služeb Query** umožňuje klientům získat podmnožinu dat z prostoru AddressSpace nebo View. [20]



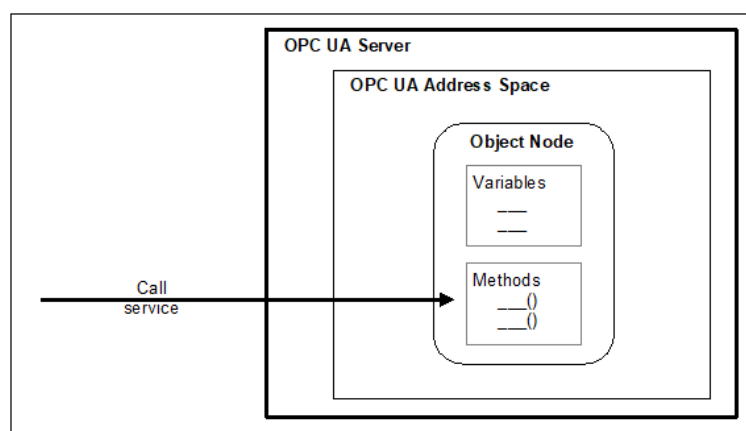
Obrázek 13: Sada služeb View a Query [20]

- **Sada služeb Attribute** definuje služby, které klientům umožňují číst a zapisovat atributy uzlů, včetně jejich historických hodnot. Protože hodnota proměnné je modelována jako atribut, umožňují tyto služby klientům číst a zapisovat hodnoty proměnných. [20]



Obrázek 14: Sada atributových služeb [20]

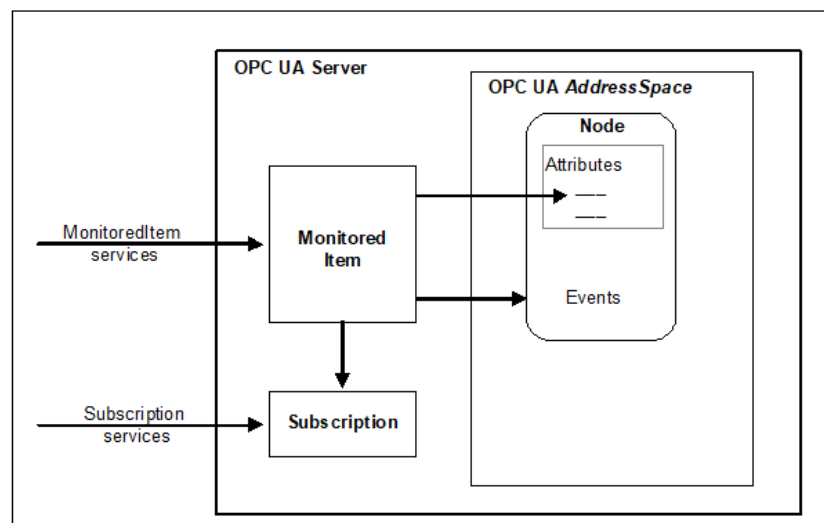
- **Sada služeb Call** jsou služby, které umožňují klientům volat funkce. Funkce se po zavolání spustí až do konce. Mohou být volány se vstupními parametry specifickými pro metodu a mohou vracet výstupní parametry specifické pro funkci. [20]



Obrázek 15: Služby Metod [20]

- **Sada služeb MonitoredItem** a **sada služeb Subscription** se společně používají k odběru uzlů v adresním prostoru OPC UA. Sada služeb MonitoredItem Service Set definuje služby, které klientům umožňují vytvářet, upravovat a odstraňovat MonitoredItems používané ke sledování změn hodnot atributů a událostí objektů. Tato oznámení jsou řazena do fronty pro přenos klientovi prostřednictvím odběrů. Sada služeb Subscription definuje služby, které umožňují klientům vytvářet, upravovat a odstraňovat Subscriptions (Odběry), které zasílají klientovi oznámení generovaná

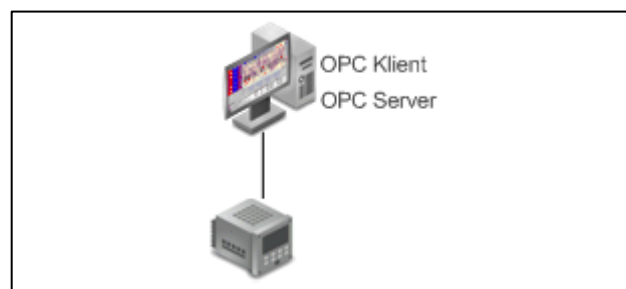
sledovanými položkami. Služby odběru také zajišťují obnovu Klienta po zmeškaných Zprávách a selháních komunikace. [20]



Obrázek 16: Služby MonitoredItem a Subscription [20]

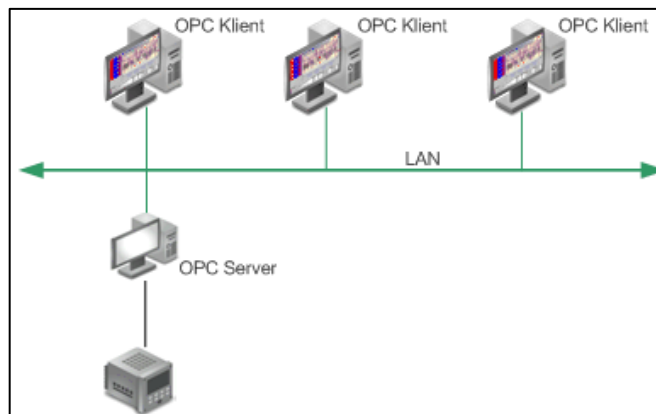
3.2 Příklady architektury OPC UA klient – server

- **Aplikace na lokální PC stanici:** Oba programy (server i klient) jsou nainstalovány na stejném počítači. Toto nastavení je typické pro aplikace s jedním účelem, jako je například výrobní stroj a podobně. Pokud je to potřeba, tento počítač lze připojit k síti Ethernet a rozšířit systém o další OPC UA servery/klienty. [21]



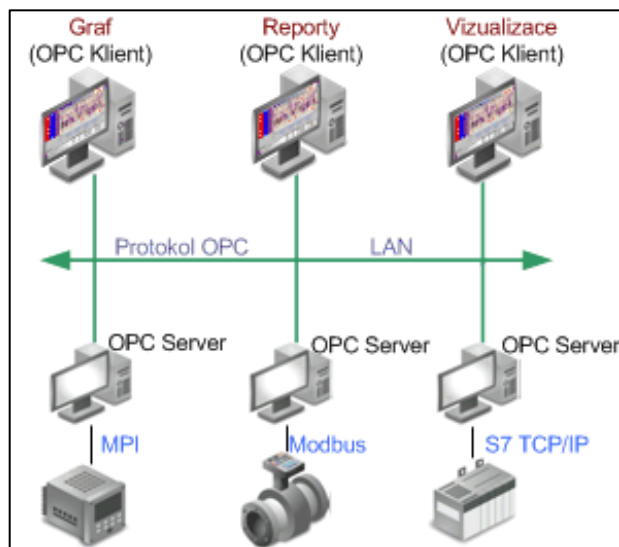
Obrázek 17: Aplikace na lokální PC stanici [21]

- **Aplikace v rámci sítě Ethernet:** OPC UA server a OPC UA klient jsou nainstalovány odděleně na různých počítačích. Tento konfigurační setup je často využíván v projektech, kde je možné propojit hardwarová zařízení s počítačem pouze pomocí sériového kabelu (například RS-232) a podobně. Počítač je umístěn v blízkosti zařízení, avšak je připojen do sítě Ethernet, což umožňuje sledování stavu zařízení například z počítače ve vaší kanceláři. Tento systém rovněž nabízí možnost snadného přidání dalších OPC UA serverů a OPC UA klientů. [21]



Obrázek 18: Aplikace v rámci sítě Ethernet [21]

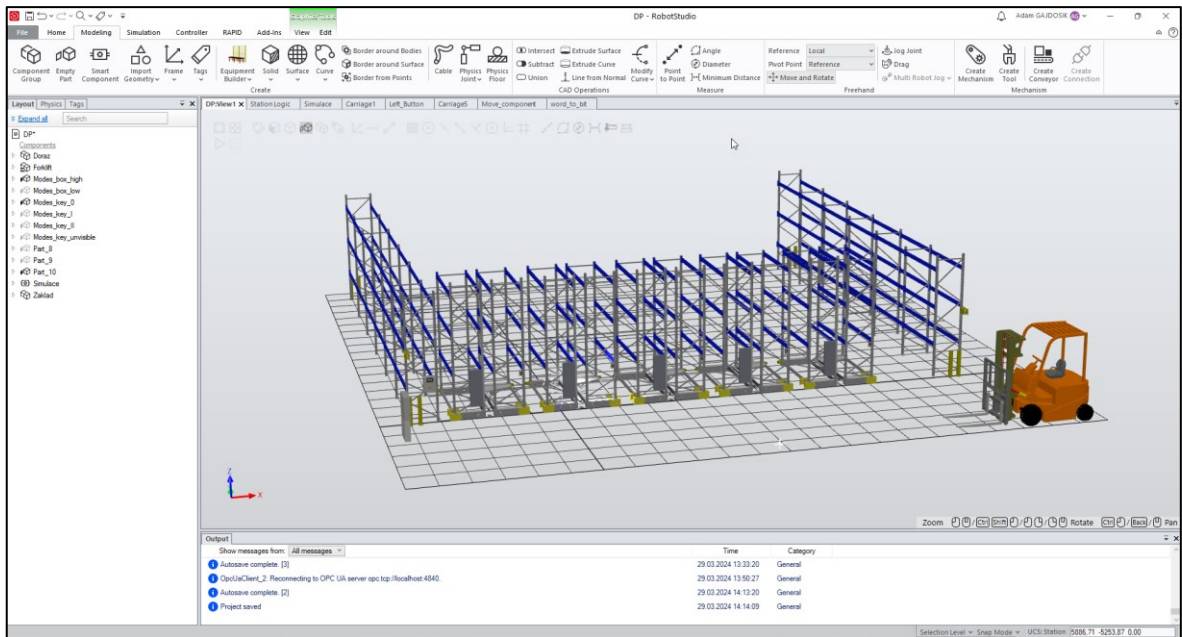
- **Rozsáhlé aplikace:** V rámci projektu jsou data čtena z více OPC UA serverů a zpracovávána v několika klientských PC stanicích. Tento typ systému je obvyklý zejména v průmyslových odvětvích, kde operátoři sledují na svých počítačích celé výrobní linky a podobně. Navíc vedení monitoruje stav výroby, plánovači sledují plnění výrobního plánu, kvalita dbá na dodržování kvality výroby a další. Běžně se v takových projektech setkáváme s až deseti OPC UA servery, přibližně stejným počtem klientských stanic a stovkami (až tisíci) komunikovaných veličin. [21]



Obrázek 19: Rozsáhlá aplikace OPC UA [21]

4 ROBOTSTUDIO

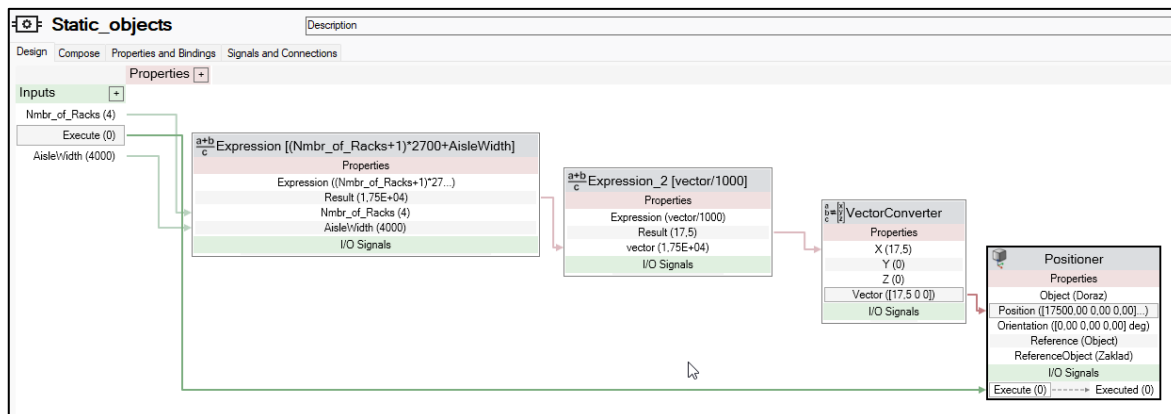
RobotStudio je počítačová aplikace pro modelování, offline i online programování a simulaci robotů od společnosti ABB. RobotStudio umožňuje uživatelům vytvářet, editovat, optimalizovat a simulovat pohyby a operace nejen průmyslových robotů bez nutnosti zastavení výroby. [22][23]



Obrázek 20: Prostředí RobotStudia

4.1 Smart Componenty

Smart Componenty jsou objekty aplikace RobotStudio s vestavěnými vlastnostmi a logikou pro simulaci komponent, které nejsou součástí virtuálního řadiče. RobotStudio ve výchozím nastavení nabízí sadu základních Smart Componentů pro základní pohyb, signální logiku, aritmetiku, parametrické modelování, senzory atd. Z těchto Componentů lze vytvořit uživatelsky definované Componenty se složitějšími vlastnostmi. Příkladem může být pohyb chlapad, pohyb objektů, logika, komunikace atd. [23]

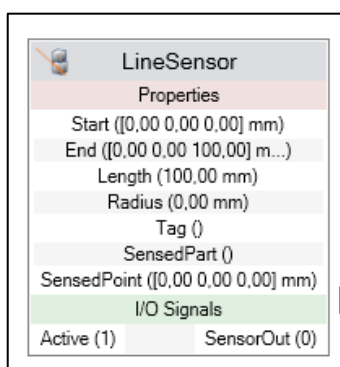


Obrázek 21: Smart Componenty

4.1.1 Vlastnosti

Vlastnosti Smart Componentů představují soubor různých parametrů, které definují Smart Componentu. Vlastnosti jsou důležité při návrhu složitých Smart Component z těch základních. Příkladem může být Smart Componenta Line Sensor, která zjišťuje, zda nějaký objekt protíná přímku mezi dvěma body. [23]

Vlastnosti Line Sensoru jsou počáteční bod, koncový bod a snímaný objekt. Počáteční a koncový bod zde určují vzdálenost, ve které je Line Sensor v aktivním stavu. Snímaný objekt poté určí, který objekt byl detekován., například krabice, která se pohybuje na pásovém dopravníku. Jednoduše si lze představit, že LineSensor virtuálně nahrazuje reálný optický snímač. [23]



Obrázek 22: LineSensor

Vazby propojují hodnotu jedné vlastnosti s hodnotou jiné vlastnosti. Tedy propojují informace z jednoho Smart Componentu do druhého. [23]

Signály jsou oddělenou vlastností Smart Componentů, které mají hodnotu, typ a směr (vstup/výstup). Tyto signály jsou obdobou vstupně-výstupních signálů na řídicí jednotce robota či na PLC. [23]

5 PLC

Programovatelný logický počítač, zkráceně PLC, představuje odolné počítačové zařízení určené pro průmyslovou automatizaci. PLC jsou programovatelné počítače, které se používají k řízení elektromechanických procesů, obvykle v průmyslových oblastech. Jsou také nazývány průmyslové počítače, což popisuje jejich hlavní funkci jako specializované průmyslové výpočetní jednotky. [24][25][26]

PLC monitorují stav vstupního zařízení, jako jsou signály z vypínače, a rozhodují o dalším stavu výstupního zařízení, například zapnutí nebo vypnutí světla a jsou také používány k přenosu informací ze zařízení v továrnách nebo z odlehlých míst do centralizovaných aplikací, často běžících na PC. PLC jsou také běžně používány k monitorování a reportování zařízení, diagnostikování chyb v hardwarových zařízeních, jako jsou průmyslové stroje a nástroje, a k provádění událostí na zařízení. [24][25][26]

5.1 Cyklus PLC

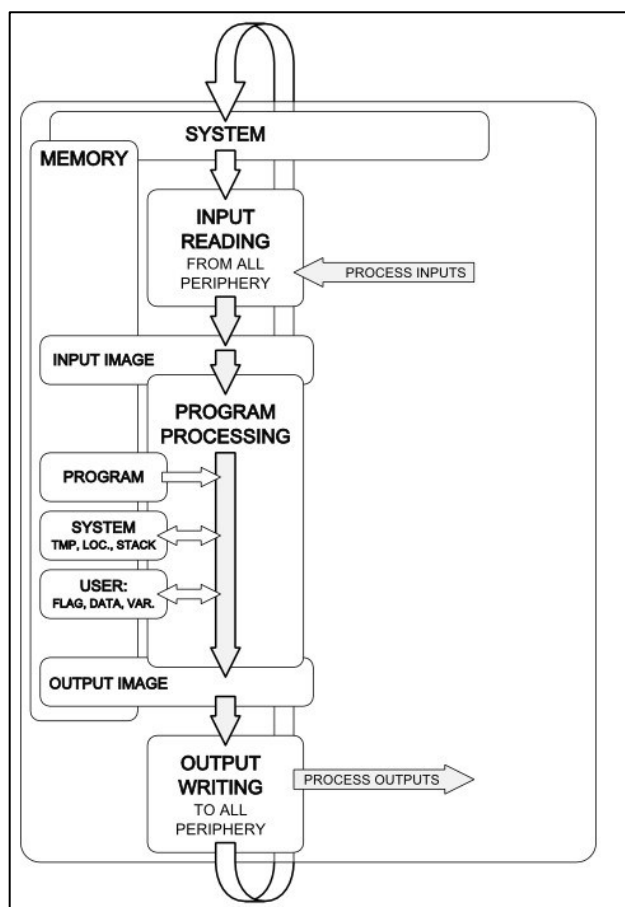
Prvním krokem na začátku každého cyklu je provádění systémových operací v CPU, které uživatel nemá možnost ovlivnit. Tento proces zahrnuje interní kontrolu, komunikaci s programátorem, manipulaci s programovými bloky a další úkony. [27]

Dalším krokem je synchronní načtení stavu všech dostupných vstupních periferních signálů a jejich uložení do paměti označované jako obraz procesních vstupů (PII - Process Input Image). Pokud se v programu odkazuje na nějaký vstupní signál, jeho hodnota je převzata právě z tohoto obrazu vstupů. I když se ukládání stavu vstupů může zdát jako zbytečná komplikace, důvodem pro toto opatření je zajistit jednoznačnost stavu vstupních signálů během jednoho cyklu programu, zejména u digitálních vstupů. Načítání stavu vstupů probíhá tzv. vzorkováním, aby se zabránilo nejednoznačným stavům a výsledkům. [27]

Po načtení stavů vstupních signálů následuje zpracování programu uloženého v paměťové oblasti vyhrazené pro program. Program je zpracováván shora dolů podle toho, jak byl napsán a sestaven programátorem. Pro chod programu se využívá pomocná paměť pro ukládání dočasných výsledků, paměť pro lokální proměnné platné pouze po určitou dobu a zásobník hloubky vnoření pro volání podprogramů a obnovení registrů. [27]

Během zpracování programu se generují výstupní signály pro řízení procesu podle algoritmu. Stav signálů se nezapisují přímo do výstupních periférií, ale ukládají se do paměti

označené jako obraz procesních výstupů (POI - Process Output Image). Po dokončení programu je požadovaný stav výstupů synchronně zapsán na fyzické výstupy periferií. [27]



Obrázek 23: Základní cyklus PLC [27]

5.2 Součásti PLC

CPU (Centrální procesorová jednotka): Stejně jako v počítači je i v PLC mozem procesor. CPU má dva provozní režimy: režim programování a režim chodu. V programovacím režimu CPU stahuje logiku ve formě programovacích instrukcí vytvořených uživatelem na PC. V režimu běhu CPU provádí logiku. [28][29][30]

CPU řídí všechny operace PLC podle programovacích instrukcí uložených v paměti. Systém řídicí sběrnice přenáší informace do a z CPU. [28][29][30]

Soubor vstupních a výstupních jednotek: Tyto jednotky jsou určeny pro komunikaci s okolím jako např. regulační ventily, motory, tlačítka apod. Existují dva druhy I/O (Input/Output) jednotek – Analogové a digitální. [29][30][31]

Analogové I/O jednotky převádí vstupní spojité signály na diskrétní číselné a naopak. Digitální I/O jednotky zpracovávají dvoustavové – nespojitý signály na vstupu a tyto stavy generují i na výstupu. [29][30][31]

Paměti: Paměť se v PLC skládá ze systémové a uživatelské:

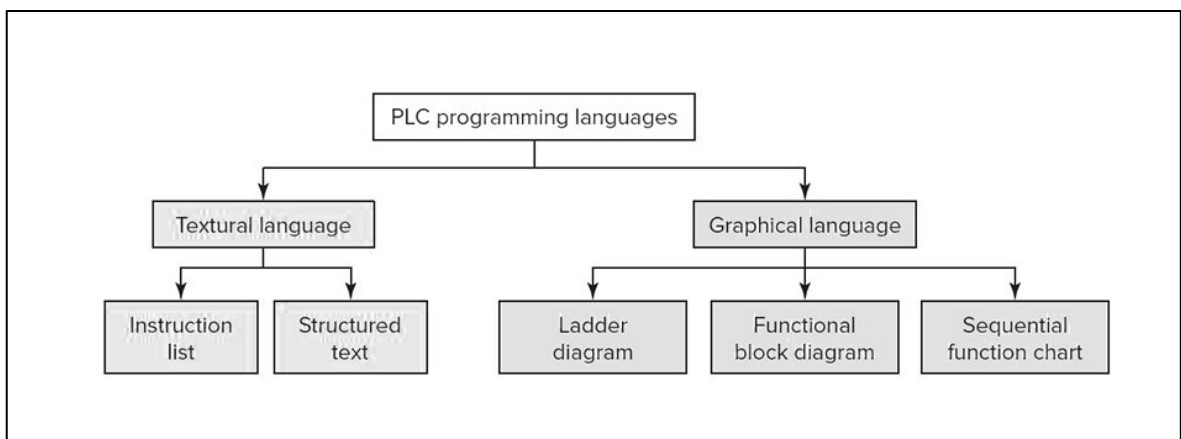
Systémová paměť je dále rozdělena na Operační paměť, Obrazy vstupů, Obrazy výstupů, Uživatelské registry a Systémové registry. [29][31]

Uživatelská paměť je dále rozdělena na Uživatelské procesy (program), Uživatelská data (proměnné), Uživatelské tabulky a Konfigurační konstanty. [28]

Soubor komunikačních jednotek: Komunikační moduly se dodávají v široké škále průmyslových standardů. Umožňují PLC komunikovat mezi sebou a se systémy, které řídí. [29][31]

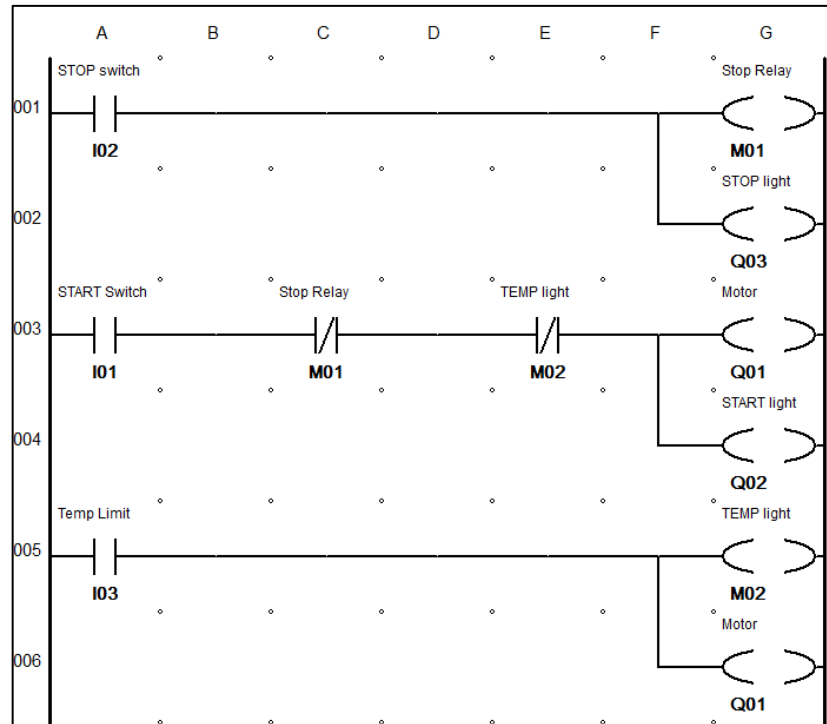
5.3 Programování PLC

PLC program může být zapsán v různých programovacích jazycích v souladu s normou IEC 61 131-3.



Obrázek 24: PLC programovací jazyky – IEC 61 131 – 3 [32]

LD (Ladder Logic): Ladder Logic je grafický programovací jazyk, který znázorňuje logické vazby mezi vstupy a výstupy pomocí symbolů a diagramů. Jedná se o nejrozšířenější programovací jazyk pro PLC systémy, který je obzvláště vhodný pro diskrétní logické řídicí aplikace. [32][33][34]



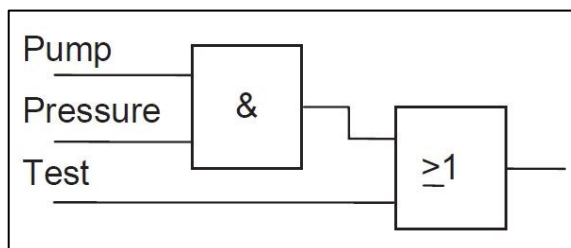
Obrázek 25: Ladder Logic [32]

ST (Structured Text): Strukturovaný text, stejně jako Pascal nebo C, je vysokoúrovňový programovací jazyk. Pracuje v textovém formátu a je nejvhodnější pro složité výpočty a práci s daty. [32][34]

```
Example:
Buzzer:=PushA OR ( PushB AND (NOT PushC) );
```

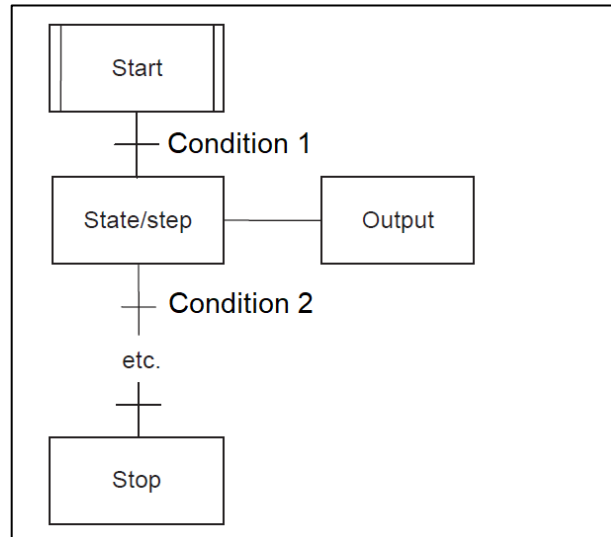
Obrázek 26: Structured Text [32]

FBD (Function Block Diagram): Function block diagram je grafický programovací jazyk, který používá bloky k reprezentaci funkcí a jejich vztahů. Nejlépe se hodí pro složité aplikace logického řízení. [32][34]



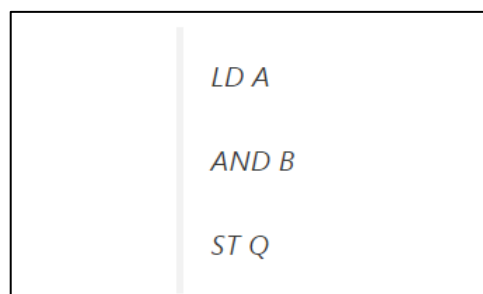
Obrázek 27: Function Block Diagram [32]

SFC (Sequential Function Chart): Sequential function chart je grafický programovací jazyk, který znázorňuje řídicí sekvence s kroky a přechody. Nejlépe se hodí pro sofistikované řídicí aplikace vyžadující sekvence událostí. [32][34]



Obrázek 28: Sequential Function Chart [32]

IL (Instruction List): Instruction List je nízkoúrovňový programovací jazyk, který reprezentuje instrukce strojového kódu v textové podobě. Nejlépe se hodí pro aplikace, které vyžadují rychlé zpracování. [32][34]



Obrázek 29: Instruction List [32]

II. PRAKTICKÁ ČÁST

6 REGÁLOVÝ SYSTÉM OD SPOLEČNOSTI SSI SCHÄFER

Posuvný regál od společnosti SSI SCHÄFER je určen pro průmyslové nebo obchodní použití za normálních provozních podmínek v zavřených prostorech a suchém prostředí a slouží k uskladnění, skladování a k odebírání zboží. [9]

Posuvná regálová zařízení pro velká zatížení **nesmějí** být zatížena například [9]:

- Nezabalené rychle se kazící zboží
- Hořlavé zboží
- Agresivní prostředky

a **nesmí** se používat [9]:

- V prostředí ohroženém explozí
- V agresivním ovzduší
- Pokud není dohodnuto s SSI SCHÄFER jinak, v oblastech se zrychlením povrchu, které překračuje stanovené hodnoty

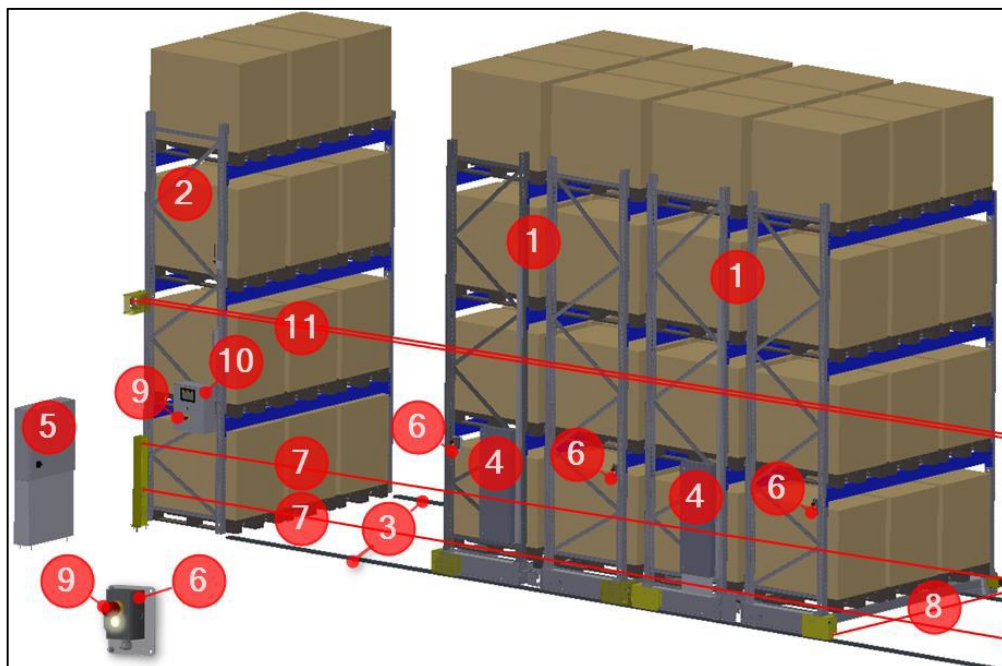
6.1 Popis

Zařízení na Obrázek 30 se skládá z posuvných regálů(1) a pevného regálu/regálů (2). Regály jsou namontovány na vozík a pohybují se pomocí elektromotorů na kolejkách (3).

Každý posuvný regál disponuje ovládním vozíku (4). Na jeden hlavní rozvaděč (5) může být připojeno až 32 posuvných regálů (33 chodeb), každý s ovládním vozíků.

Obsluha regálového systému probíhá buď manuálně na zařízení, pomocí dálkového ovládní nebo plně automaticky pomocí nadřazeného systému správy skladu (LSV – Local Speed Vault). Při obsluze zařízení se otevře chodba, která se nachází vedle obslužné jednotky (6). Na prvním vozíku se nachází vždy vlevo a vpravo obslužná jednotka (6), u každého dalšího vozíku se nachází obslužná jednotka (6) vždy vpravo na podpěře. [9]

Bezpečnost zajišťují přístupové světelné závory (7) před regálem (a na všech přístupných místech), bezpečnostní nožní světelné závory (8) v oblasti chodidel posuvný regálový vozík (1), jakožto i tlačítko NOUZOVÉHO ZASTAVENÍ (9) na každé obslužné jednotce (6) a pod HMI panelem (10). [9]



Obrázek 30: Dílčí pohled na zařízení [9]

6.1.1 Klíčový spínač provozních režimů (NOUZOVÁ JÍZDA)

Klíčový spínač provozních režimů je manuální zařízení pro nouzový provoz k použití při blokadě zařízení nebo ve zvláštním provozu. [9]

Provozní režimy:

MANUAL: Nouzová jízda

0: VYP.

AUTO: Normální provoz (automatický provoz)



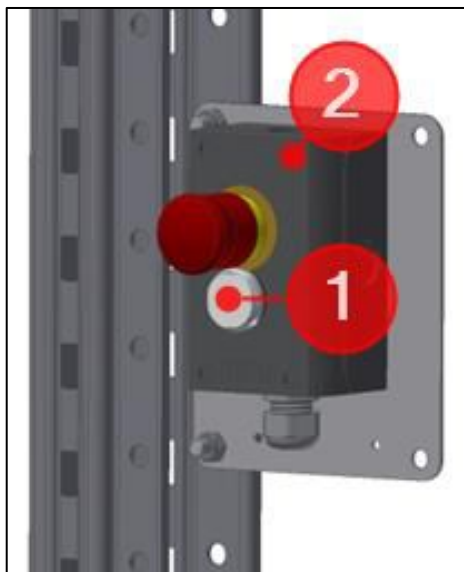
Obrázek 31: Klíčový spínač provozních režimů [9]

6.1.2 Obslužná jednotka

Obrázek 32 ukazuje obslužnou jednotku, kde stisknutím obslužného tlačítka (1) na obslužné jednotce (2) se chodba po kontrole uvolní. Opětovným stisknutím se udělí příkaz ke startu (vozik vyjede). V případě poruchy se stisknutím ovládacího tlačítka potvrdí porucha. [9]

Světelné signály:

- Pomalé blikání (2x za sekundu). => varovná signalizace za jízdy
- Rychlé blikání (4x za sekundu). => porucha na příslušném vozíku.
- Tlačítko svítí (ZAP). => chodba uvolněna, posuvný regál je připraven k jízdě
- Tlačítko nesvítí (VYP). => Chodba nebyla uvolněna. => Spuštěna bezpečnostní funkce.
 - Spuštění pomocí kontroly přístupu. => Všechna tlačítka jsou VYP.
 - Spuštění pomocí NOUZOVÉHO ZASTAVENÍ nebo bezpečnostní nožní světelné závory. => Tlačítko, na kterém je spuštěna bezpečnostní funkce. bliká 4x za sekundu. Všechna ostatní tlačítka jsou VYP.



Obrázek 32: Obslužná jednotka [9]

6.1.3 Obslužný panel

Přes obslužný panel se zobrazují provozní stavy a chybová hlášení. Kromě toho může být zařízení obsluhováno přes obslužný panel a parametry ovládání lze nastavit. [9]



Obrázek 33: Obslužný panel [9]

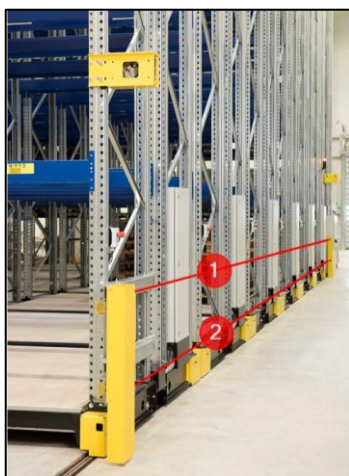
6.2 Bezpečnostní světelné závory a senzory

6.2.1 Přístupová světelná závora

Přístupové světelné závory (1) a (2) slouží ke sledování přístupu k regálovým chodbám. Jsou instalovány ve výšce 900 mm (1) a 400 mm (2).

U odstavených zařízení a na bezpečnostních světelných závorech namontovaných (jezdících) na regálu musí paletová místa, vedle kterých jsou namontovány bezpečnostní světelné závory, zůstat volná z důvodů nedostatku místa. [9]

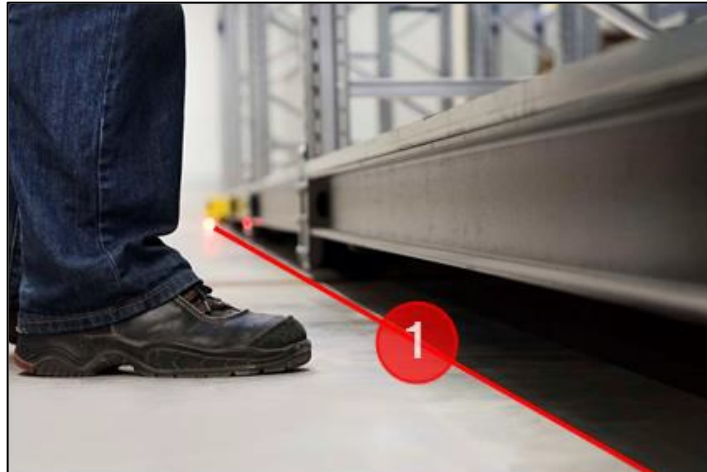
- Při přerušení přístupové světelné závory poté, co se otevřela chodba, vychází ovládnání z toho, že se v otevřené chodbě nachází osoba. => Zařízení se zastaví.
- Pokud se přístupové světelné závory přeruší v nějakém jiném okamžiku (např. během jízdy regálu), spustí se nouzové zastavení. => Zařízení se zastaví.



Obrázek 34: Přístupové světelné závory [9]

6.2.2 Bezpečnostní nožní světelná závora

Bezpečnostní nožní světelná závora (1) sleduje každý posuvný regálový vozík ve směru chodu. Pokud se v normálním provozu (automatickém provozu) v uzavírající se chodbě nachází nějaká osoba nebo předmět, spustí systém nouzové zastavení a vozík zastaví.



Obrázek 35: Bezpečnostní nožní světelná závora [9]

6.2.3 Koncový spínač

Na posuvném regálovém vozíku jsou připevněny světelné senzory (1), které při zadané vzdálenosti k dalšímu vozíku zahájí brzdění a zastavení.

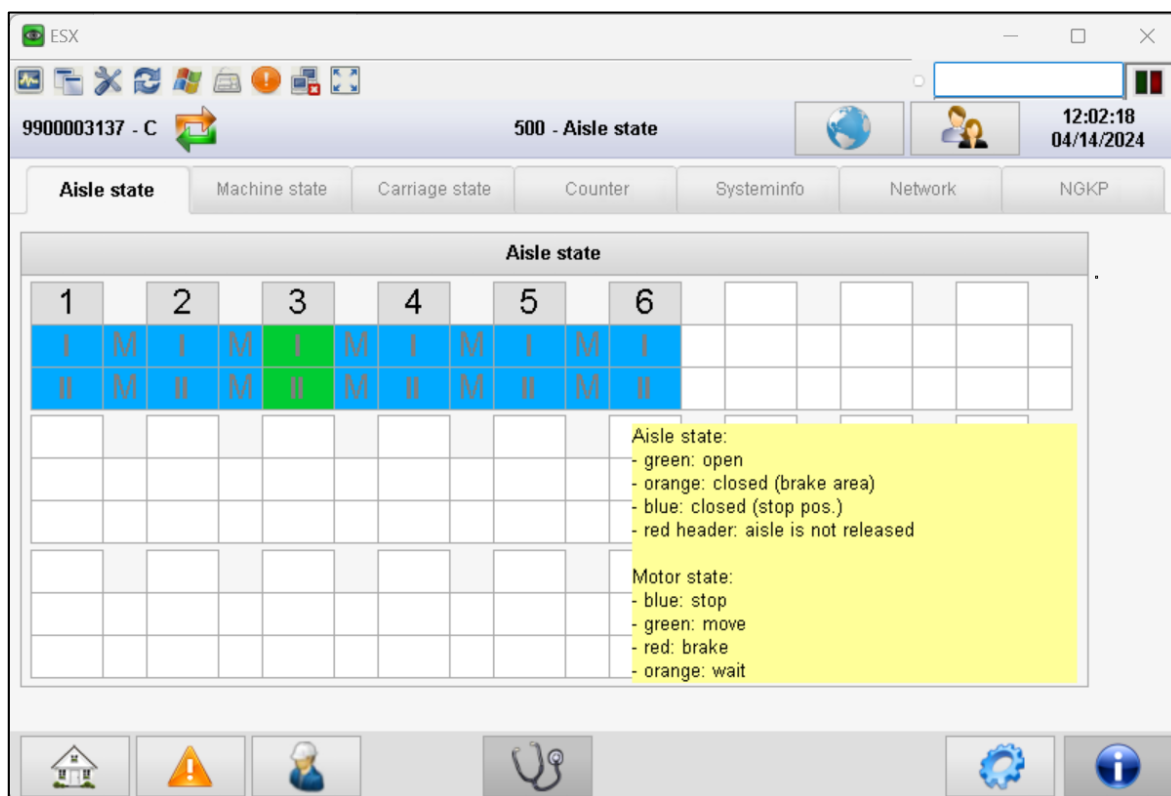


Obrázek 36: Koncový spínač [9]

7 PŮVODNÍ STAV

Původním prostředkem pro simulování posuvných paletových regálů měla firma SSI SCHÄFER pouze samotný PLC simulátor, na jehož HMI se lze připojit. Následně se lze z hlavní nabídky dostat do Diagnostiky, kde lze vidět aktuální konfiguraci (počty vozíků a chodeb), v jakém jsou vozíky stavu (stop, pohyb, brždění, čekání) a v jakém stavu jsou jednotlivé uličky (otevřeno, zavřeno-brždění, zavřeno-stop, ulička není uvolněna).

Jiný prostředek pro testování, vizualizaci a vývoj firma SSI SCHÄFER neměla.



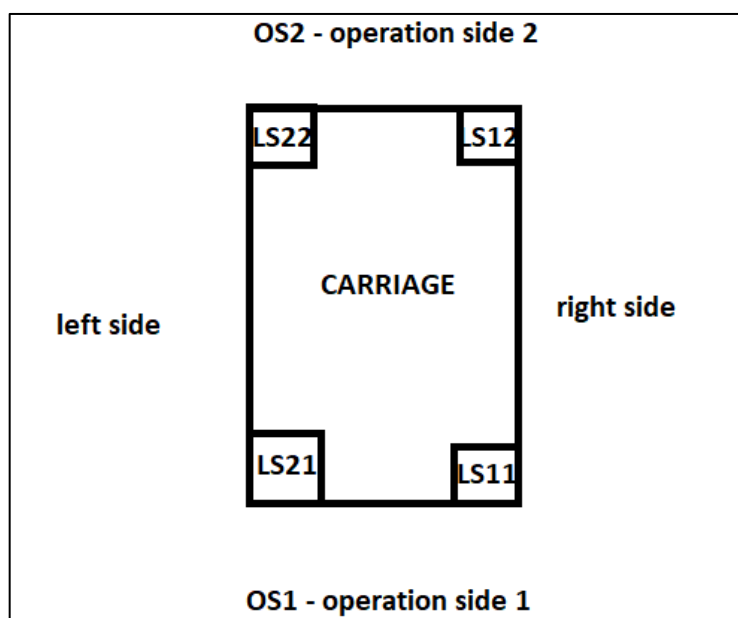
Obrázek 37: Diagnostika na HMI panelu PLC

8 LOGIKA A FUNKCE REGÁLOVÉHO SYSTÉMU

Pro vytvoření digitálního dvojčete posuvného regálového systému poskytla firma SSI SCHÄFER všechny potřebné podklady. V těchto podkladech byl i tabulkový soubor, kde byly popsány veškeré potřebné vstupy a výstupy PLC. Soubor je rozdělen do několika sekcí.

8.1 Ovládání

Ovládání regálového systému se skládá z pohledu vstupů z tlačítek a sensorů. Tato sekce se v PLC jmenuje CCM. Vozík je rozdělen na části, podle kterých jsou pojmenovány a přiřazeny senzory.



Obrázek 38: Rozdělení vozíku

Součástí je vstupní proměnná o velikosti 16 bitů, která se nazývá `ai_int_fromSL_CCC_Sts`. V této proměnné se promítají stavy jednotlivých sensorů na vozíku.

Další vstupní a výstupní proměnné se týkají tlačítek, které jsou umístěny na vozíku. Tedy na prvním a posledním vozíku jsou dvě tlačítka a na ostatních vozících je jen jedno. Vstupní proměnná nese informaci stisknutí tlačítka a výstupní proměnná ovládá podsvícení tlačítka.

Tabulka 1: Popis vstupů a výstupů tlačítek na vozíku

Inputs	Description
di_bol_CE11_BtnOpenAisle	operation Button OS1 right
di_bol_CE12_BtnOpenAisle	operation Button OS2 right
di_bol_CE21_BtnOpenAisle	operation Button OS1 left
di_bol_CE22_BtnOpenAisle	operation Button OS2 left
Outputs	Description
do_bol_CE11_LED	Button backlighting
do_bol_CE12_LED	Button backlighting
do_bol_CE21_LED	Button backlighting
do_bol_CE22_LED	Button backlighting

8.2 Pohyb

Druhá část souboru DRV určuje vstupy a výstupy frekvenčních měničů. Důležitou částí je vždy odezva frekvenčního měniče na příkazy z PLC. Proto je důležité porozumět sekvencím pro pohyb a spuštění systému z PLC.

Jako výstup z PLC jsou proměnné SetSpeed_LFRD_rpm, která udává otáčky, které má mít motor a CtrlWord_CMDD, která udává příkazy frekvenčnímu měniči.

Tabulka 2: Výstupy z PLC pro pohyb systému

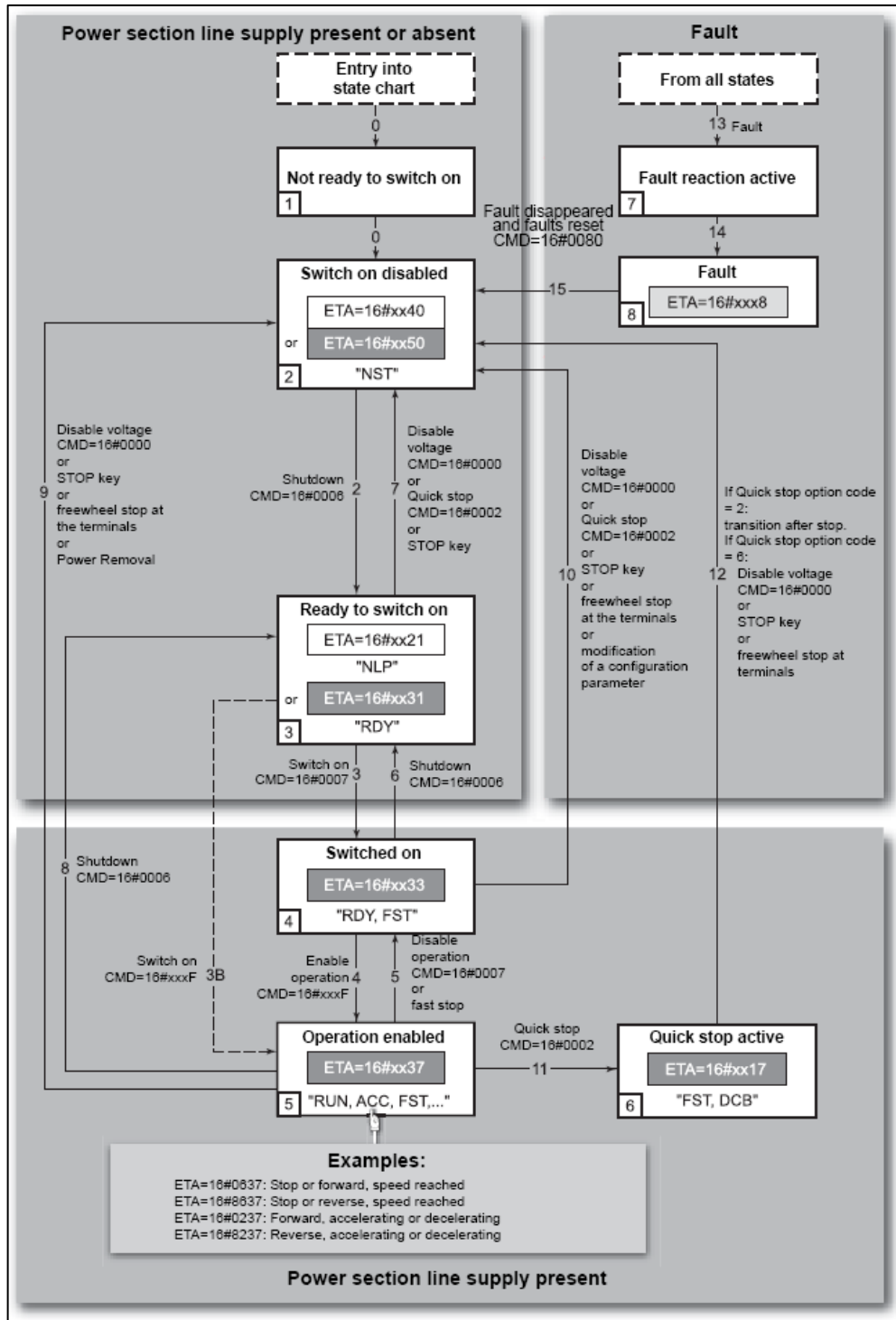
Outputs		Description
SetSpeed_LFRD_rpm		Setpoint speed
CtrlWord_CMDD	bit 0	On/Off
	bit 1	Enable voltage
	bit 2	Quick stop (always 1)
	bit 3	Enable operation
	bit 4	always 0
	bit 5	always 0
	bit 6	always 0
	bit 7	Fault reset
	bit 8	Halt
	bit 9	always 0
	bit 10	always 0
	bit 11	Set direction of rotation
	bit 12	always 0
	bit 13	always 0
	bit 14	always 0
	bit 15	always 0

Jako vstupy do PLC jsou proměnné Speed_RFRD_rpm, která udává otáčky, jaké motor posílá jako zpětnou vazbu a StatusWord_ETAD, která udává zpětnou vazbu frekvenčního měniče a jeho stav.

Tabulka 3: Vstupy z PLC pro pohyb systému

Inputs		Description
Speed_RFRD_rpm		Actual speed
StatusWord_ETAD	bit 0	Ready
	bit 1	Switched on
	bit 2	Operation enabled
	bit 3	Fault
	bit 4	Voltage enabled
	bit 5	Quick stop
	bit 6	Switch on disabled
	bit 7	always 0
	bit 8	always 0
	bit 9	always 0
	bit 10	always 0
	bit 11	always 0
	bit 12	always 0
	bit 13	always 0
	bit 14	always 0
	bit 15	direction

Následující obrázek ukazuje, jak program v PLC blokově funguje:



Obrázek 39: Sekvence provozu frekvenčních měničů

Pro zjednodušení vznikla tabulka s popisem komunikace PLC – frekvenční měnič, kde je popsán stav, příkaz z PLC a odpověď frekvenčního měniče pro provoz.

Tabulka 4: Popis komunikace mezi PLC a frekvenčním měničem

Condition / State		CMD / Action		Status Word check	
FAULT	16#xxx8	16#0080	Reset fault		
		16#090F	ERROR	16#xx50	Reset
MOVE right					
	1.	16#0007	Switch on	16#xx33	Switched on
	2.	16#000F	Enable operation	16#xx37	Operarational
	3.	11. bit = 0	Set direction to right		
	4.	LFRD_rpm > 0	Set rpm		
MOVE left					
	1.	16#0007	Switch on	16#xx33	Switched on
	2.	16#000F	Enable operation	16#xx37	Operarational
	3.	11. bit = 1	Set direction to left		
	4.	LFRD_rpm > 0	Set rpm		
Slow down					
		LFRD_rpm	Change rpm		
STOP					
	1.	8. bit = 1	Halt		
	2.	LFRD_rpm = 0	Set rpm to 0	RFRD_rpm = 0	No speed
	3.	16#0000	OFF	16#xx50	Switched off
	4.	16#0006	Shut down	16#xx31	Ready

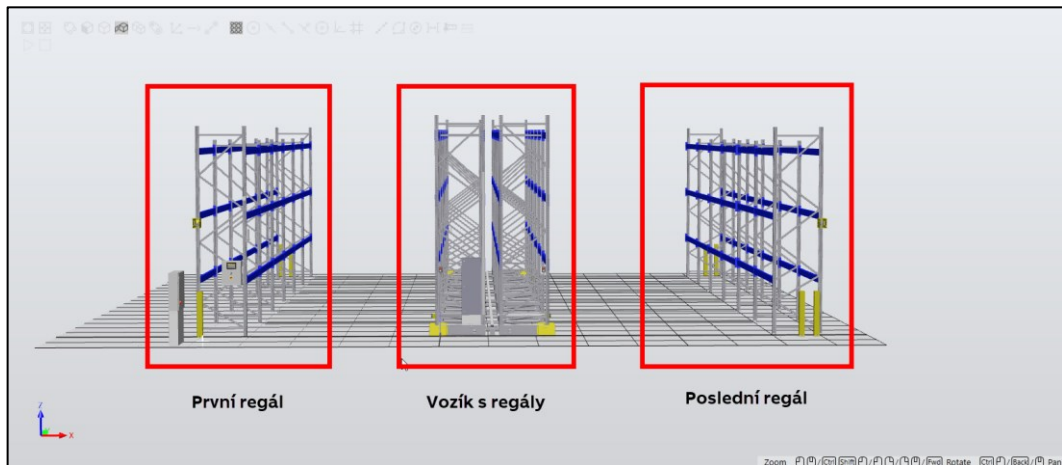
8.3 Bezpečnost

Poslední část souboru popisující vstupy a výstupy PLC je bezpečnost. Jedná se o signály sloužící k zajištění bezpečnosti zařízení. Jsou to signály světelných závor, tlačítka nouzového zastavení a signály z klíčového přepínače režimů provozu.

9 SESTAVENÍ REGÁLOVÉHO SYSTÉMU

9.1 3D modely

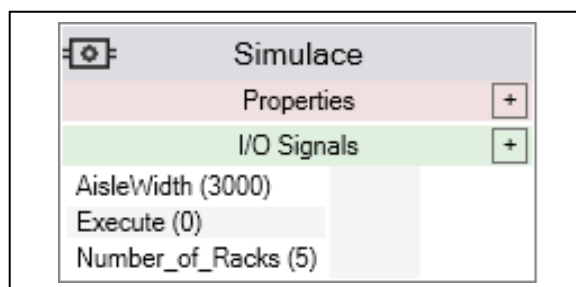
V těchto podkladech byly potřebné 3D modely, jako jeden celek. Nejdříve bylo tedy nutné tento celek rozdělit na statické části/objekty, jako je první regál a poslední regál. Dále bylo nutné z tohoto celku vybrat pohyblivé části, tedy vozík s regály.



Obrázek 40: Modely regálového systému

Regálový systém obsahuje vždy první statický regál, poslední regál a vždy minimálně jeden vozík s regály. Dále je nutné, aby se systém vždy uzpůsobil dané konfiguraci simulátoru PLC. Parametry, které udávají regálový systém jsou – **počet vozíků s regály** a **šířka uličky**. Na základě těchto parametrů je nutné z poskytnutých modelů dynamicky vygenerovat kompletní regálový systém.

Pro vygenerování kompletního regálového systému byly použity Smart Componenty v RobotStudiu. Hlavní Smart Component Simulace má tedy vstupní parametry šířku uličky a počet vozíků.

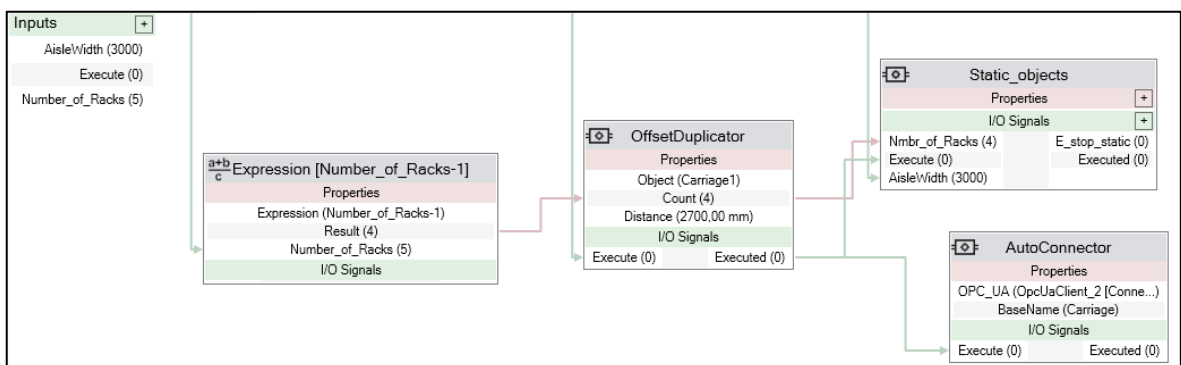


Obrázek 41: Smart Component Simulace

9.2 Smart Componenty pro generování regálového systému

Po zadání parametrů je nejdříve nutné duplikovat pohyblivé vozíky s regály jedenkrát méně, než kolik zadal uživatel, jako vstupní parametr do Smart Componentu simulace. Dále je nutné, aby se každý duplikovaný vozík s regály posunul o šířku vozíku. V případě poskytnutého modelu tato šířka činí 2700 [mm].

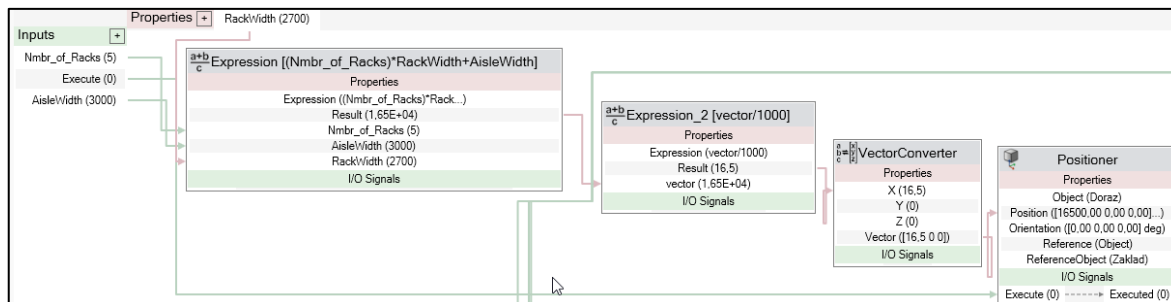
Pro duplikování vozíků byl použit Smart Component OffsetDuplicator (Obrázek 42), kde jako vstupní parametry jsou objekt, který se má duplikovat, počet a vzdálenost v ose X, po které se budou objekty duplikovat.



Obrázek 42: Smart Componenty pro generování regálového systému

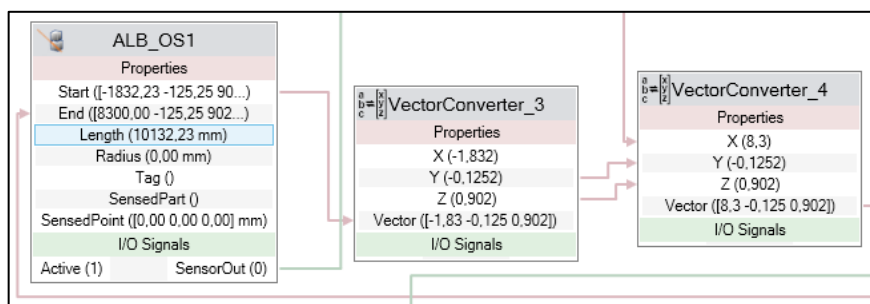
Poté, co jsou vygenerovány vozíky s regály, tak je nutné vygenerovat poslední regál a upravit vstupní světelné závory. Pro tyto operace byl vytvořen Smart Component Static_objects. Jako vstupní parametry tohoto Smart Componentu jsou šířka uličky a počet přidávaných vozíků.

V tomto Smart Componentu (Obrázek 43) se nejdříve musí vypočítat vzdálenost posunutí koncového regálu. Proto je použit Smart Component Expression, který po zadání vzorce s neznámými vygeneruje tyto neznámé, jako vstupní parametr. Vzorec v tomto případě vznikl vynásobením počtu vozíků s jejich šířkou a součtem s šířkou uličky. Výsledná hodnota se poté převedla na vektor pomocí Smart Componentu VectorConverter. Tento vektor, který obsahuje velikost posunutí po souřadnici X se následně vložil jako vstupní parametr pozice Smart Componentu Positioner, který daný objekt posune na správnou pozici s orientací. V tomto případě je objekt poslední regál.



Obrázek 43: Smart Componenty pro posunutí posledního regálu

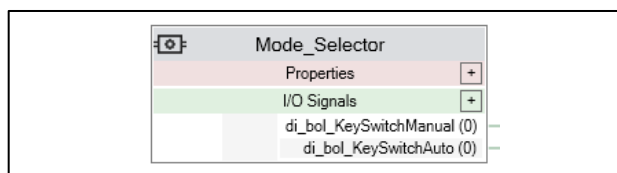
Vstupní světelné závory byly vytvořeny pomocí Smart Componentu LineSensor (Obrázek 44), který má jako vstupní parametry počáteční bod, který je statický a koncový bod, který se mění podle polohy posledního regálu. K tomu slouží Smart Component VectorConverter, kde hodnoty souřadnic Y a Z se propíší z počátečního bodu a hodnota X z polohy posledního regálu. Jako výstupem z tohoto senzoru je SensorOutput, který se nastaví na 1, pakliže je signál mezi počátečním a koncovým bodem přerušen.



Obrázek 44: Smart Componenty pro vytvoření vstupních světelných závor

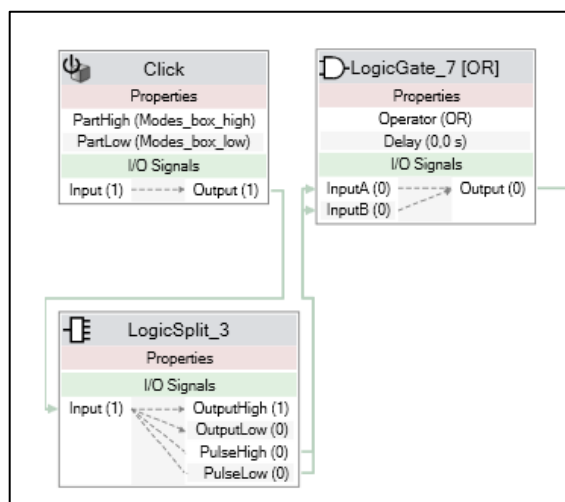
9.3 Smart Componenty pro přepínání režimů

Pro přepínání režimů byl využit model klíčového přepínače provozních režimů, který je umístěný na prvním statickém regálu. Pro přepnutí režimu byl vytvořen Smart Component Mode_selector (Obrázek 45), který má výstupní parametry signály, které nesou informaci o vybraném režimu.



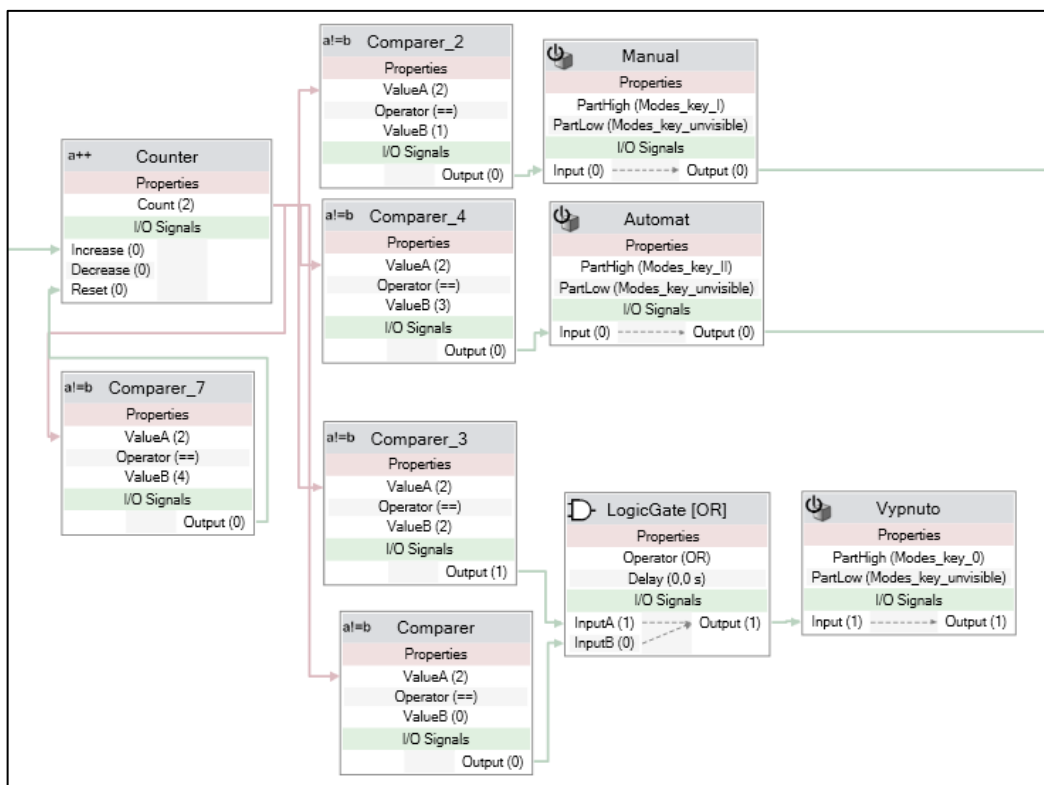
Obrázek 45: Smart Component Mode_Selector

Pro snímání kliknutí na daný objekt byl využit Smart Component GraphicSwitch (Obrázek 46), který má vstupní parametry dva objekty PartHigh a PartLow, které po kliknutí na daný objekt jeden schová a druhý zobrazí. Na tyto vstupní parametry bylo třeba namodelovat oblast pro kliknutí, která byla následně zprůhledněna. Následně byl výstup z tohoto Smart Componentu předán do Smart Componentu LogicSplit, který má jako vstupní parametr Input a výstupní parametry OutputHigh, OutputLow, PulseHigh a PulseLow. LogicSplit přebírá hodnotu Input a nastavuje OutputHigh na totožnou hodnotu a OutputLow na hodnotu inverzní. PulseHigh vyše impuls, když je Input nastaven na hodnotu 1 a PulseLow vyše impuls, když je Input nastaven na hodnotu 0.



Obrázek 46: Snímání kliknutí na přepínač
provozních režimů

Pro reálný chod klíčového spínače provozních režimů jde klíček vždy přes hodnotu 0. Pro tento jev se každé kliknutí na přepínač inkrementuje proměnná, která se následně porovnává s hodnotami, které určují, v jaké pozici by měl klíček být. Jakmile hodnota přesáhne hodnotu 3, tak se tato proměnná vyresetuje. Zobrazení klíčku na správné pozici je poté vytvořeno pomocí Smart Componentu GraphicSwitch.



Obrázek 47: Zobrazení přepínače provozních režimů ve správné pozici a orientaci



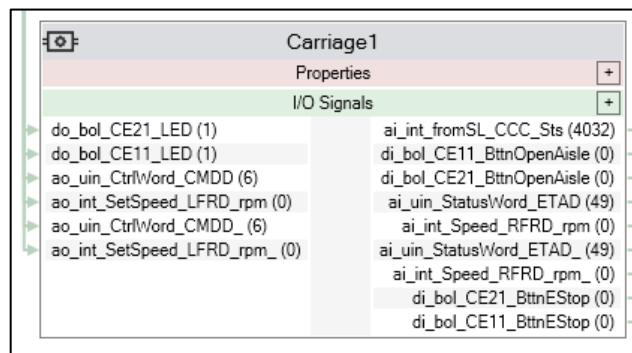
Obrázek 48: Klíčový přepínač provozních režimů

10 SIMULACE VOZÍKU

Po vyčlenění vozíku s regály z 3D modelu bylo nutné k tomuto modelu vytvořit patřičné prostředky reakce s okolím a napojením na PLC.

Všechny potřebné prvky simulace digitálního dvojčete musely být škálovatelné. Z tohoto důvodu byly veškeré prvky implementovány do jednoho Smart Componentu s názvem Carriage (Obrázek 49).

Tento Smart Component má vstupní parametry podsvícení tlačítek, ovládací proměnné pro frekvenční měniče vozíku a žádané otáčky motoru. Výstupní parametry jsou poté proměnná nesoucí informaci o stavu senzorů, stavy tlačítek pro ovládání vozíku, stavy frekvenčních měničů, rychlosti otáček motorů a stavy tlačítek nouzového zastavení. Všechny tyto parametry jsou pojmenovány dle konfiguračního souboru a jsou připojeny s PLC skrz OPC UA protokol.

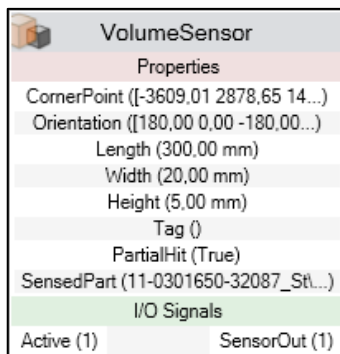


Obrázek 49: Smart Component Carriage

10.1 Senzory vozíku

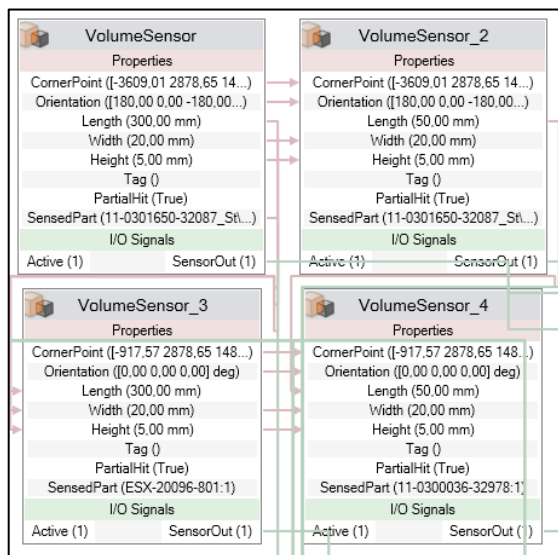
Jako senzory vozíku byly myšleny:

- **Senzory přiblížení**, které sepnou výstup, když se vozík dostane 300 [mm] k překážce. Senzory jsou určeny pro zjištění, jestli se již vozík dostal k jinému vozíku nebo k prvnímu či poslednímu regálu, tedy od této chvíle by vozík měl začít zpomalovat. Tedy tyto senzory jsou dva. Každý na jedné straně vozíku. Senzory přiblížení byly realizovány pomocí Smart Componentu VolumeSensor, který vytvoří kvádr a když nějaký objekt protne tento kvádr tak sensor sepnou výstup na hodnotu 1. Vstupní parametry tohoto Smart Componentu jsou pozice a orientace rohového bodu, a rozměry kvádru. Jako výstup je logická hodnota, jeli nějaký objekt v kvádru a o jaký objekt se jedná.



Obrázek 50: Smart Component VolumeSensor

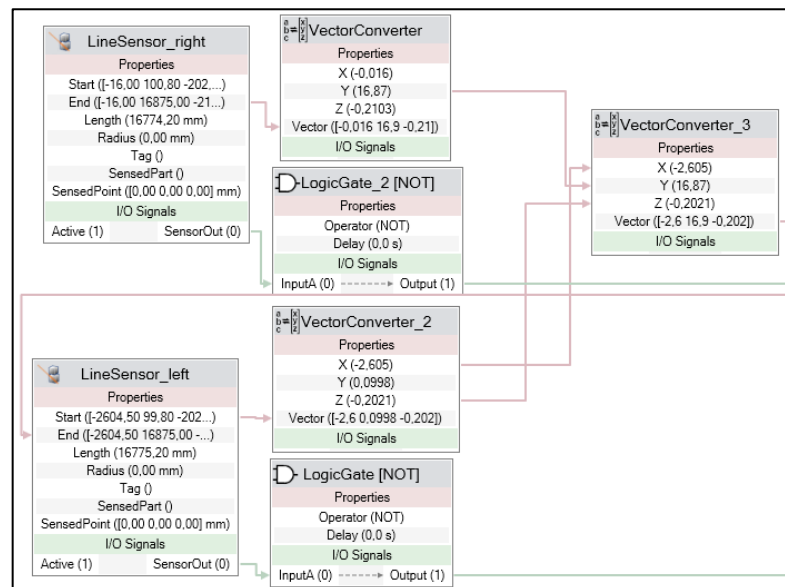
- **Senzory dorazu**, které sepnou výstupní parametr SensorOut, když se vozík dostane 50 [mm] k překážce. Senzory slouží pro zjištění, jestli je vozík v těsné vzdálenosti jiného vozíku nebo prvního či posledního regálu. Když sensor sepne na hodnotu 1 znamená to, že na straně, kde je sensor sepnut je ulička uzavřena. Senzory dorazu byly realizovány taktéž Smart Componenty Volume Sensor, jen s rozdílným parametrem Length, kde je 50 [mm]. Ostatní parametry byly zkopírovány ze sensoru přiblížení.



Obrázek 51: Smart Componenty sensorů přiblížení a dorazů

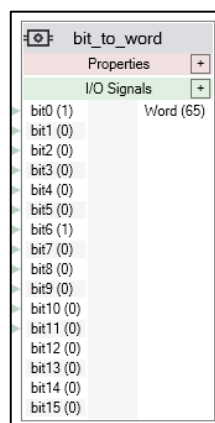
- **Senzory nožních světelných závor**, které mají hodnotu 1, když v uličce není překážka. Světelné nožní závory byly realizovány pomocí Smart Componentů LineSensor tak, jako ostatní světelné závory. Jako vstupní parametr pro koncový bod byl spojen pomocí Smart Componentu VectorConverter, kdy pro levý sensor byla

použita stejná délka jako pro pravý. Jelikož samotný sensor má výstupní hodnotu 1, když v uličce překážka je, bylo nutné tento výstup znegovat logickým hradlem NOT.



Obrázek 52: Sensory nožních světelných závor

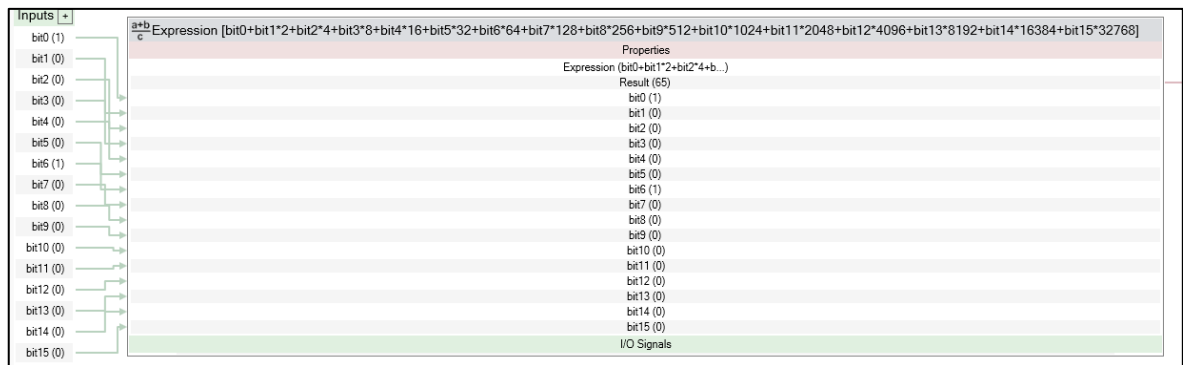
Následně se všechny výstupy ze sensorů musejí převést na jedno šestnácti bitové číslo. Pro tento přepočít byl vytvořen Smart Component bit_to_word, který má jako vstupní parametry 16 bitů a jako výstup jedno šestnácti bitové číslo.



Obrázek 53:

Smart Component bit_to_word

Převod do dekadické soustavy byl vytvořen pomocí Smart Componentu Expression (Obrázek 54). Rovnice pro převod z binární do dekadické soustavy byla sestavena násobením jednotlivých bitů od konce číslem vždy dvakrát větším než předchozím číslem počínaje jedničkou.



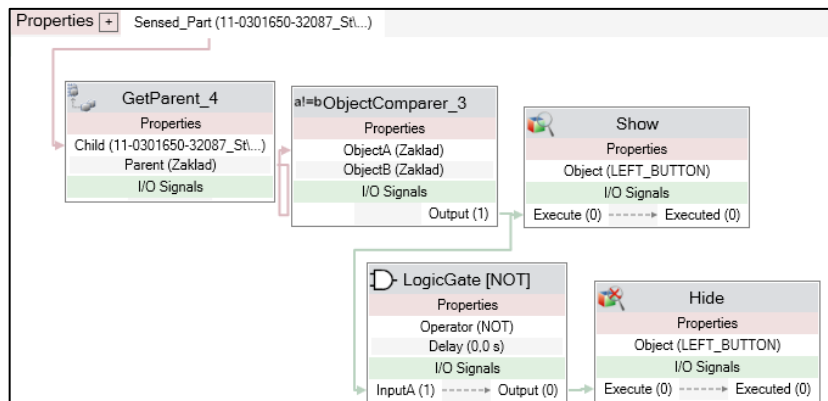
Obrázek 54: Převod z binárního čísla na dekadické

10.2 Tlačítka

Každý vozík obsahuje sekci, kde je podsvícené tlačítko pro ovládání a tlačítko nouzového zastavení. První vozík obsahuje tyto sekce dvě. Pro kopírování vozíků bylo tedy nutné dynamicky levou část s tlačítky schovat pro ostatní zkopírované vozíky.

Nejdříve bylo nutné zjistit, jaká část či model se nachází po levé straně vozíku. Tato informace byla již obsažena senzorem pro přiblížení či doraz ze Smart Componentu VolumeSensor, který obsahuje výstupní parametr SensedPart.

Byl tedy vytvořen Smart Component Left_Buttons, který má vstupní parametr Sensed_Part, který se předal ze sensoru přiblížení na levé straně vozíku. V tomto Smart Componentu se poté tato část modelu předala Smart Componentu GetParent, který má jako výstupní parametr nadřazenou část. Následně se tato nadřazená část porovná s modelem základního či prvního regálu pomocí Smart Componentu ObjectComparer. Pokud je výstup z tohoto porovnání jedna, tak se levá část s tlačítky objeví. Pro operaci schování a objevení modelu ve stanici RobotStudia jsou Smart Componenty Show a Hide, které mají jako vstupní parametr daný objekt a příkaz pro vykonání dané funkce Execute.



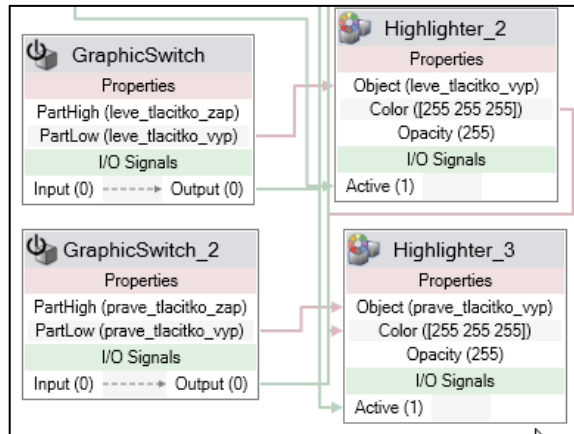
Obrázek 55: Smart Component Left_buttons

Podsvícená tlačítka byla vytvořena pomocí dvou Smart Componentů. Reakce na stisk tlačítka byla vytvořena pomocí Smart Componentu GraphicSwitch, který má dva vstupní parametry PartHigh a PartLow a jeden výstupní parametr Output. Na základě kliknutí na objekt schová objekt PartLow a ukáže objekt PartHigh. Pokud je viditelný objekt PartHigh tak se nastaví Output na hodnotu 1. Z tohoto důvodu musí existovat dva modely na totožné pozici, kde jeden model má odlišnou barvu pro imitaci stisknutí tlačítka.

Další vlastnost ovládacího tlačítka je jeho podsvícení. To bylo vytvořeno pomocí Smart Componentu Highlighter, který má jako vstupní parametry Object, barvu a neprůhlednost.

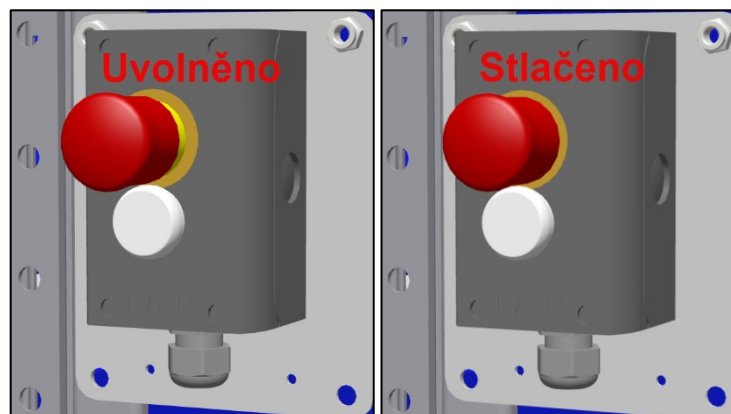


Obrázek 56: Ovládací tlačítka vozíku



Obrázek 57: Smart Componenty ovládacích tlačítek vozíku

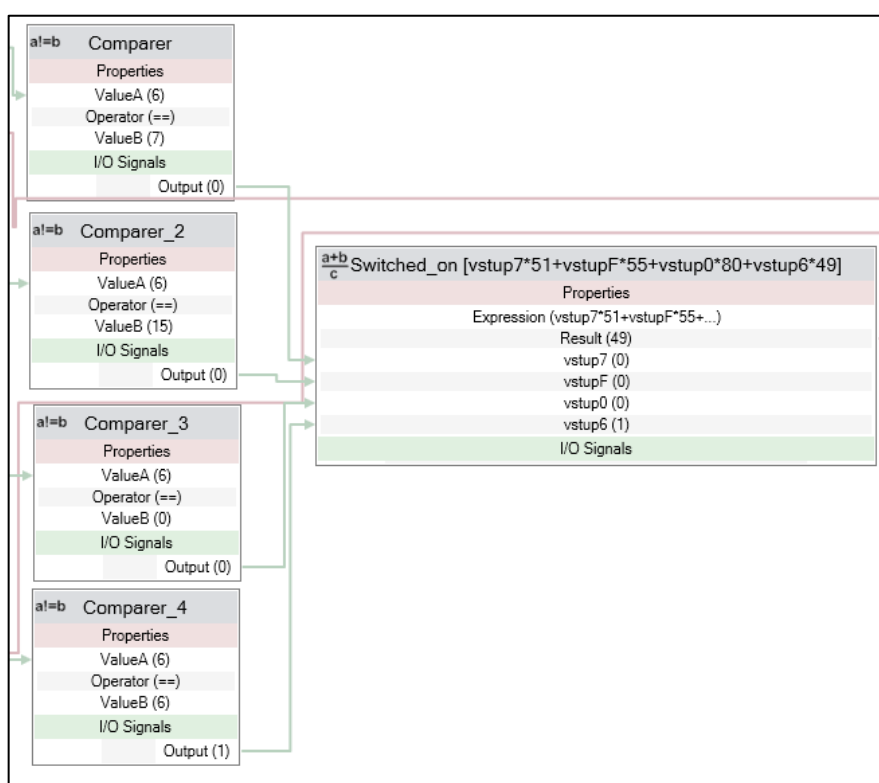
Tlačítka nouzového zastavení byla realizována taktéž pomocí Smart Componentů GraphicSwitch. Tudiž bylo potřeba vytvořit totožný model, který by imitoval stlačené tlačítko nouzového zastavení.



Obrázek 58: Tlačítka nouzového zastavení

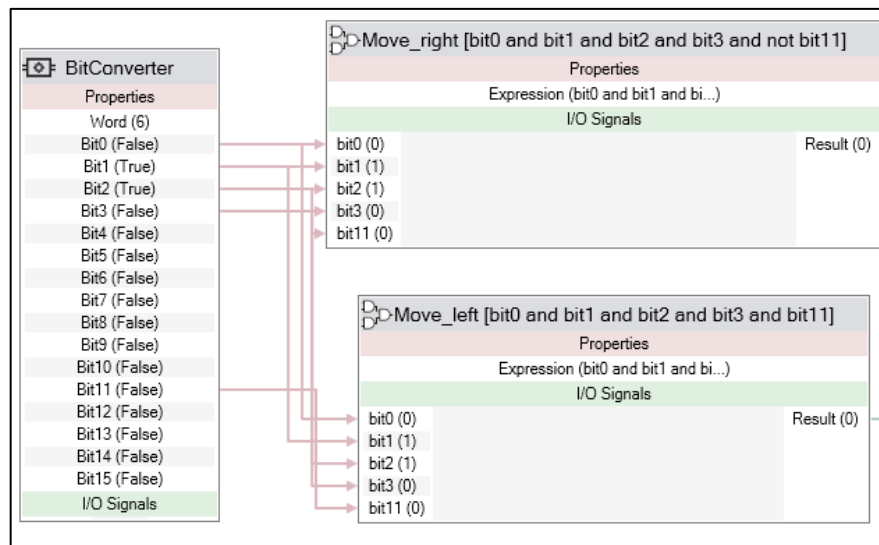
10.3 Pohyb

Pohyb vozíku zajišťují dva frekvenční měniče. Nejdříve bylo nutné nasimulovat správnou odezvu stavů frekvenčních měničů na příkazy z PLC dle konfiguračního souboru poskytnutého firmou SSI. To bylo vytvořeno pomocí Smart Componentů Comparer, který porovnává příkazy z PLC a očekávané hodnoty dle konfiguračního souboru. Jako výstup je pouze hodnota 1, když se vstupy shodují. Bylo tedy nutné ještě tyto hodnoty změnit na požadované hodnoty odezvy. K tomuto byl vytvořen Smart Component Expression, kde byla vytvořena rovnice ze součtu součinů, kde výsledkem je správná odezva na příkaz z PLC.



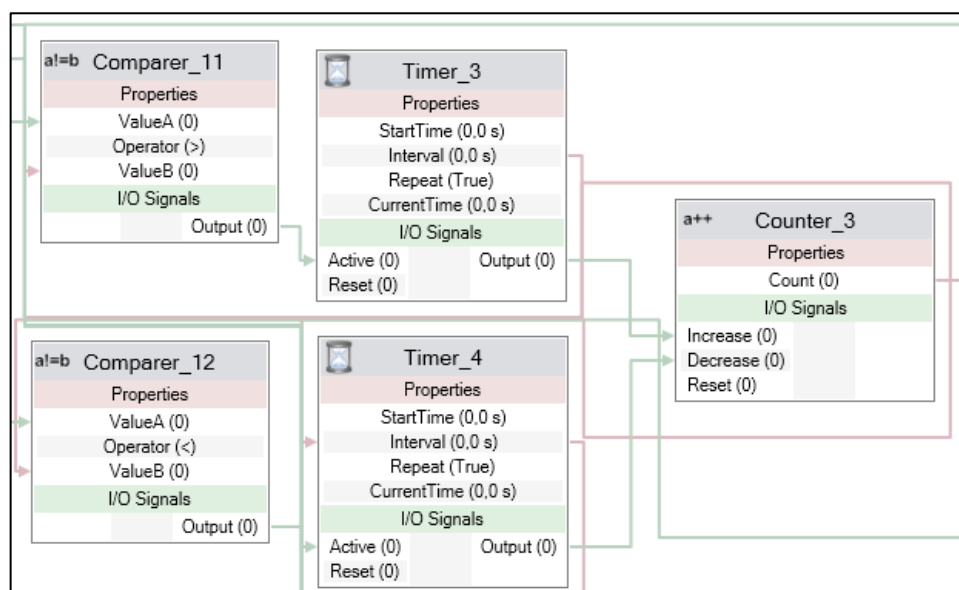
Obrázek 59: Odezva simulovaných frekvenčních měničů

Dále bylo nutné zjistit, co daný příkaz z PLC znamená. Podle konfiguračního souboru bylo zjištěno, že se jedná o přirozené číslo. Pro určení příkazu bylo nutné toto přirozené číslo převést na číslo binární. K tomuto byl použit Smart Component BitConverter, který převádí dekadické číslo na jednotlivé bity. Poté se dle konfiguračního souboru zjistily stavy, při kterých je vozík v pohybu. Následně se dle stavu jednotlivých bitů vytvořily logické rovnice pomocí Smart Componentů LogicExpression pro pohyb doleva a doprava.



Obrázek 60: Určení stavů pro pohyb vozíku

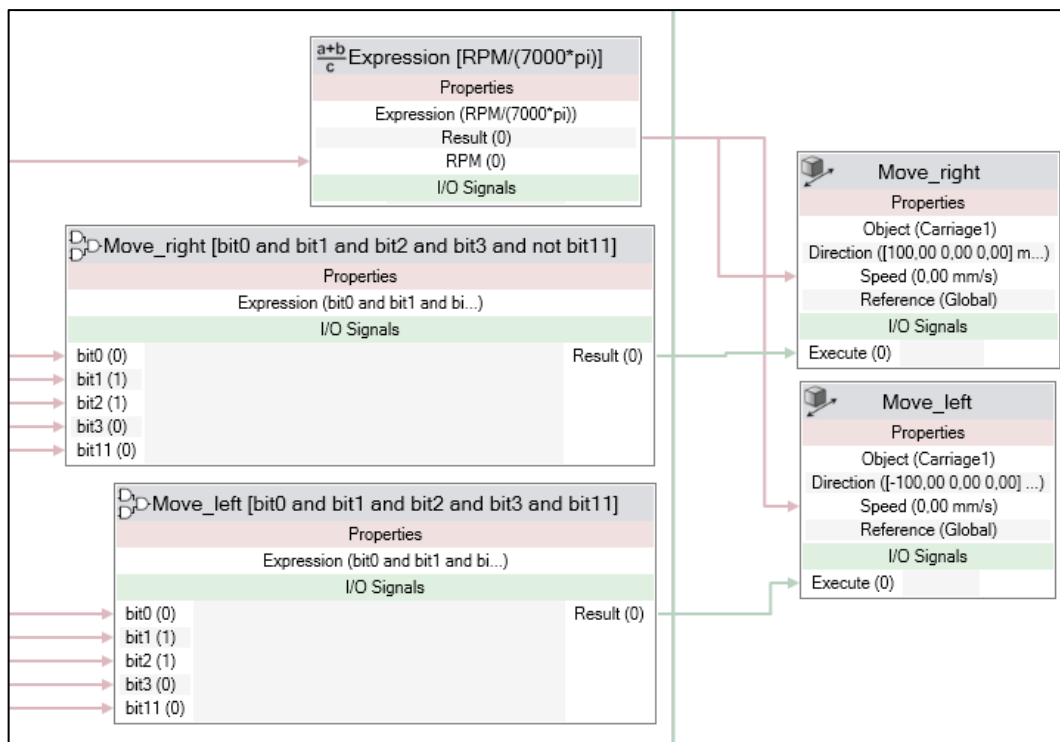
Pro nasimulování akcelerace vozíku, tedy postupné zvyšování rychlosti, byla hodnota pro rychlost otáček nejdříve porovnána Smart Componentem s aktuální rychlostí otáček. Pokud rychlost otáček z PLC je vyšší než aktuální hodnota, tak se začne přes časovač inkrementovat hodnota do té doby, než se dostane na požadovanou hodnotu. Časovač posílá pulzy co 0,024 [s] do vstupu pro inkrementaci hodnoty Smart Componentu Counter, který má jako vstupy Increase, Decrease a reset. Když je hodnota rychlosti otáček menší, než požadovaná hodnota z PLC tak se přes časovač posílají pulzy do vstupu pro dekrementaci. Výsledná hodnota se poté posílá do PLC jako zpětná vazba o rychlosti otáček.



Obrázek 61: Smart Componenty pro simulaci akcelerace vozíku

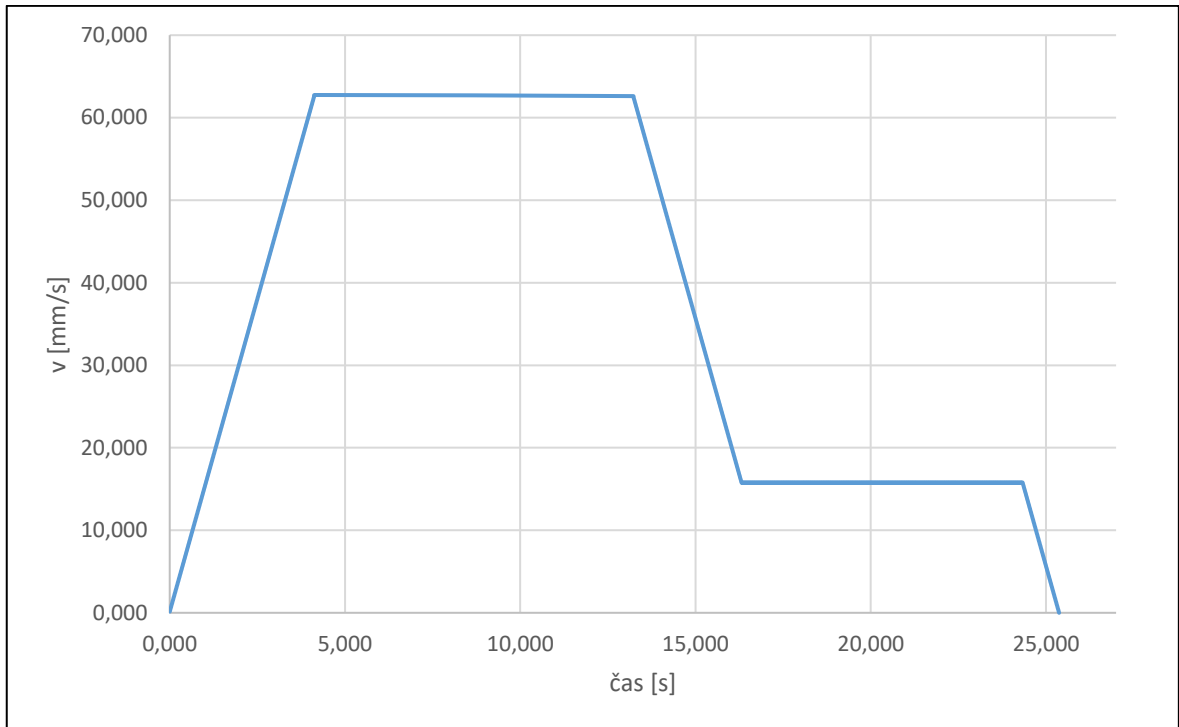
Výsledné otáčky motoru je dále nutné podělit převodem, což bylo vytvořeno pomocí Smart Componentu Expression, kdy vstupní parametr otáčky byly poděleny převodovým poměrem tak, aby výsledná rychlost dosáhla reálné rychlosti vozíku což je přibližně 65 [mm/s].

Samotný pohyb byl vytvořen pomocí Smart Componentu LinearMover, který má vstupní parametry objekt, který se má pohybovat, směr, ve kterém se má objekt pohybovat a rychlost, kterou se má objekt pohybovat.



Obrázek 62: Realizace pohybu vozíku

V programovém prostředí RobotStudio je možné nahrávat průběhy různých veličin. Pro ověření akcelerace vozíku a zobrazení průběhu rychlosti byla tato hodnota zaznamenána.



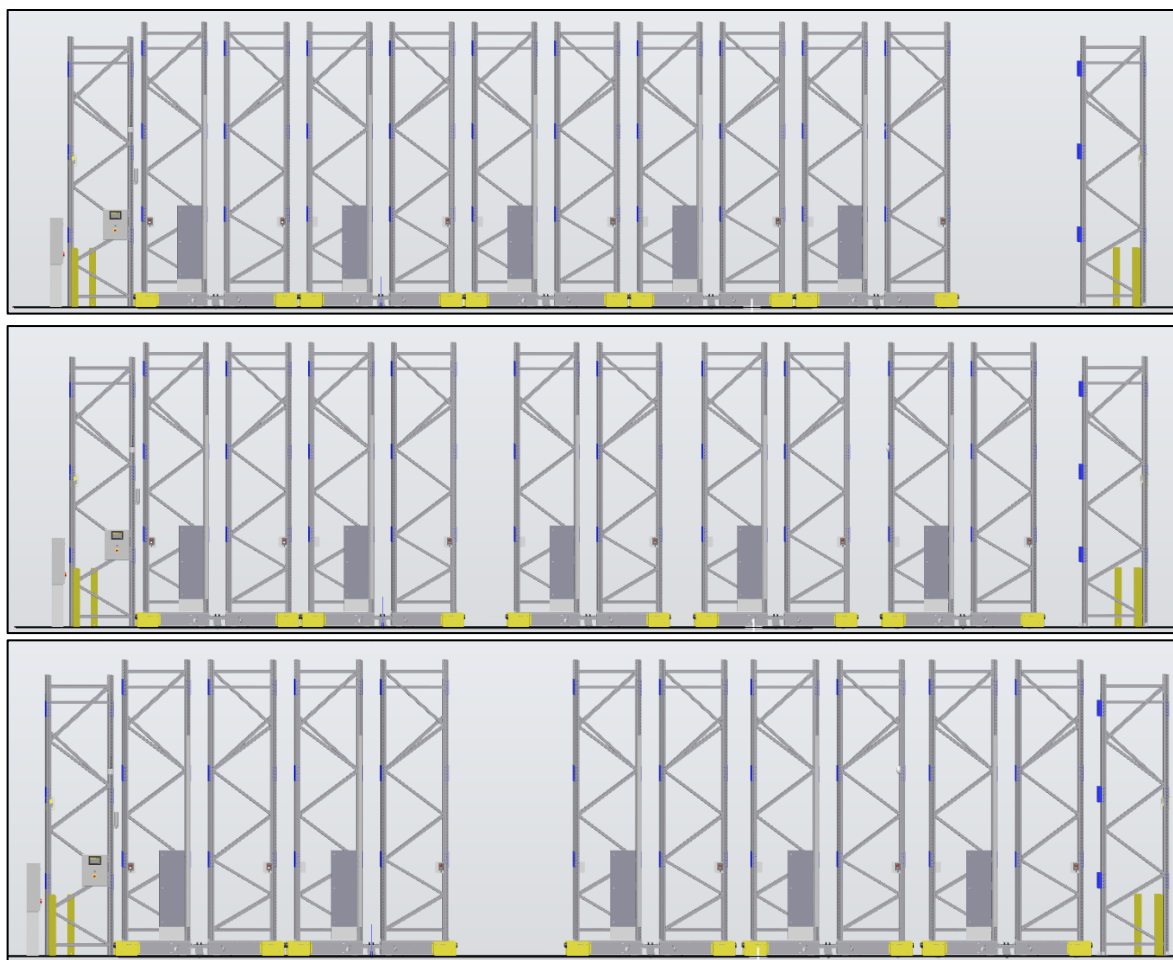
Obrázek 63: Průběh rychlosti vozíku

Na obrázku lze vidět, že nárůst rychlosti je postupný. Po dosažení potřebné rychlosti se vozík pohybuje konstantní rychlostí do doby, než je vzdálen od překážky (jiného regálu) 300 [mm]. V tu chvíli vozík začne zpomalovat. Poté jede vozík zpomalenou rychlostí až do doby, než se snímač dorazu nastaví na hodnotu 1. Poté vozík zastaví.

Tento Smart Component se z pohledu PLC chová jako reálný frekvenční měnič a plně vykonává jeho příkazy, tj. pohyb samotný vykonává tento Smart Component, ale rozhodování o tom, kdy se systém začne pohybovat a jakým směrem, určuje PLC.

10.4 Průběh otevírání uličky

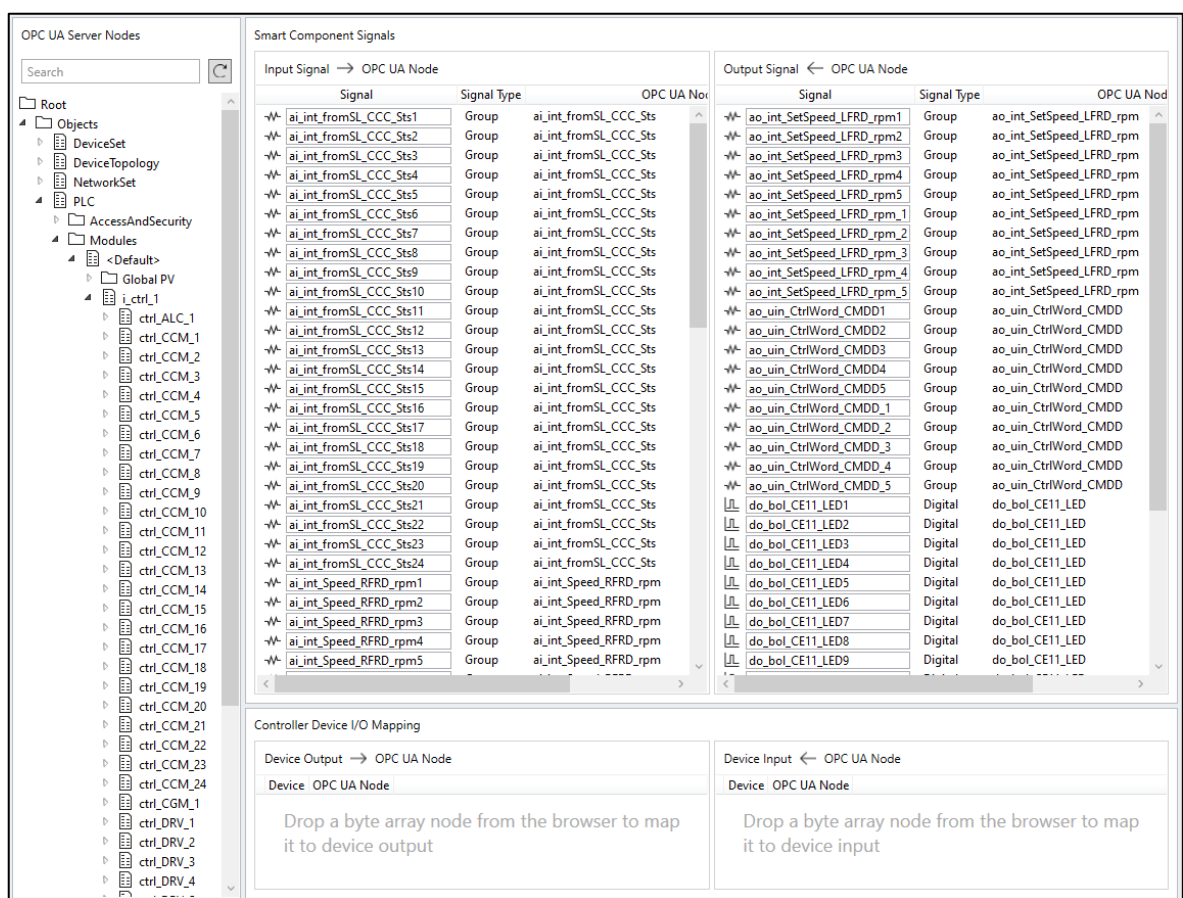
Pro ukázkou lze vidět na Obrázek 64 průběh otevírání třetí uličky v systému, který má pět vozíků a uličku o šířce 2000 [mm].



Obrázek 64: Průběh otevírání uličky regálového systému

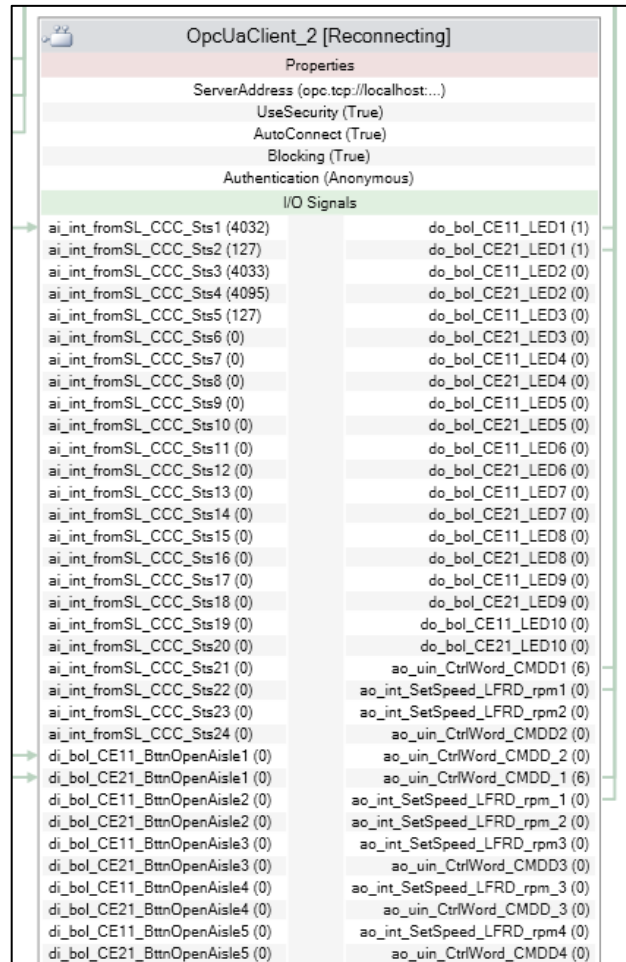
11 PROPOJENÍ S PLC

Propojení digitálního dvojčete v RobotStudiu a virtuálního PLC, které je spuštěno na pozadí počítače a které dodala společnost SSI SCHÄFER, je vytvořeno skrze OPC UA protokol. V RobotStudiu existuje pro tento účel Smart Component OpcUaClient, který se dokáže automaticky napojit na PLC. Smart Component lze poté jednoduše konfigurovat tím, že se ve stromu proměnných, které připravila firma SSI SCHÄFER, vyberou všechny potřebné signály. Tyto signály byly následně pojmenovány podle konfiguračního souboru s číslem, pro který vozík mají být přiřazeny.



Obrázek 65: Konfigurace OPC UA klienta

Po dokončení konfigurace OPC UA klienta se Smart Component automaticky změní a zobrazí se nakonfigurované vstupy a výstupy.



Obrázek 66: Ukázka vstupu a výstupu Smart Componentu UpcUaClient

ZÁVĚR

V této práci se podařilo vytvořit digitální dvojče paletových posuvných regálů od společnosti SSI SCHÄFER, které lze jednoduše propojit s PLC skrze komunikační protokol OPC UA.

Při vytváření digitálního dvojčete byl kladen důraz na zprovoznění všech ovládacích a důležitých částí, které jsou spojeny s PLC a dle kterých PLC řídí reálný paletový posuvný regál. Tím se dosáhla plná virtualizace regálového systému a z pohledu PLC se tak jedná o jeho kompletní digitální dvojče.

Dále byl zimplementován komunikační protokol OPC UA mezi vytvořeným digitálním dvojčetem v programovém prostředí RobotStudio a PLC dodaném společností SSI SCHÄFER.

Po nastudování všech patřičných signálů a funkcí systému z konfiguračního souboru, který dodala společnost SSI SCHÄFER, byla vytvořena logika výměny signálů a vykonávání příkazů z PLC.

Další důležitou částí bylo vytvoření Smart Componentu, který na základě dat z konfiguračního souboru, což je počet regálů a šířka uličky, dokáže automaticky sestavit celý regálový systém.

V poslední části byl vytvořen testovací krokový postup pro spuštění a rychlou kontrolu funkčnosti celého systému navrženého v programovém prostředí RobotStudio, který je součástí přílohy P1.

Celková simulace paletového posuvného regálu, tedy jeho digitální dvojče, byla řádně otestována a během testování nevykazovala žádné nedostatky či chyby.

SEZNAM POUŽITÉ LITERATURY

- [1] IBM. What is a digital twin? Online. IBM. 2024, 25.2.2024. Dostupné z: <https://www.ibm.com/topics/what-is-a-digital-twin>. [cit. 2024-02-25].
- [2] SEGOVIA, Mariana a GARCIA-ALFARO, Joaquin. Design, Modeling and Implementation of Digital Twins. Online. Sensors. 2022, roč. 22, č. 14. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s22145396>.
- [3] KŘÍŽ, Lukáš. Digitální dvojčata závisí na kvalitě dat. ICT REVUE. 2021, roč. 2021, č. 11, s. 04–08.
- [4] Digital twin modeling. Journal of Manufacturing Systems [online]. 2022, 2022(7), 372-389 [cit. 2024-03-10]. Dostupné z: [doi:https://doi.org/10.1016/j.jmsy.2022.06.015](https://doi.org/10.1016/j.jmsy.2022.06.015)
- [5] MCFARLAND, Alex. What is a Digital Twin? Unite.ai [online]. 2023, 9. června, 2023 [cit. 2024-03-10]. Dostupné z: <https://www.unite.ai/what-is-a-digital-twin/>
- [6] ESSEX, David. Digital twin. Techtarget [online]. 2024, March 2024 [cit. 2024-03-10]. Dostupné z: <https://www.techtarget.com/searcherp/definition/digital-twin>
- [7] PLANK, Thomas. DIGITAL TWINS: THE 4 TYPES AND THEIR CHARACTERISTICS. Tributech [online]. 2019 [cit. 2024-03-10]. Dostupné z: <https://www.tributech.io/blog/the-4-types-of-digital-twins>
- [8] BARRICELLI, Barbara Rita; CASIRAGHI, Elena a FOGLI, Daniela. A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. Online. IEEE Access. 2019, roč. 7, s. 167653-167671. ISSN 2169-3536. Dostupné z: <https://doi.org/10.1109/ACCESS.2019.2953499>.
- [9] Posuvný regál pro velké zatížení: Návod k provozu [ESX_OM1_CS_Návod k provozu]. 2. SSI SCHAEFER, 2017.
- [10] AR RACKING. Mobile Pallet Racking (AR Mobile) [online]. 2024 [cit. 2024-03-17]. Dostupné z: <https://www.ar-racking.com/en/storage-systems/industrial-racking-systems/pallet-racking/mobile/>
- [11] JUNGHEINRICH. Dynamické skladování palet - Posuvný policový regál [online]. 2024 [cit. 2024-03-17]. Dostupné z: <https://www.jungheinrich.cz/produkty/regaly/dynamicke-skladovani-drobnych-dilu/posuvny-policovy-regal-492404>

- [12] OPC FOUNDATION. Unified Architecture. OPC FOUNDATION. OPC Foundation The Industrial Interoperability Standard [online]. 2006 [cit. 2024-04-06]. Dostupné z: <https://opcfoundation.org/about/opc-technologies/opc-ua/>
- [13] OPC FOUNDATION. What is OPC? OPC FOUNDATION. OPC Foundation The Industrial Interoperability Standard [online]. 2006 [cit. 2024-04-06]. Dostupné z: <https://opcfoundation.org/about/what-is-opc/>
- [14] OPC FOUNDATION. Classic. OPC FOUNDATION. OPC Foundation The Industrial Interoperability Standard [online]. 2006 [cit. 2024-04-06]. Dostupné z: <https://opcfoundation.org/about/opc-technologies/opc-classic/>
- [15] HENSEL, Stephan; GRAUBE, Markus; URBAS, Leon; HEINZERLING, Till a OPPELT, Mathias. Co-simulation with OPC UA. Online. In: 2016 IEEE 14th International Conference on Industrial Informatics (INDIN). IEEE, 2016, s. 20-25. ISBN 978-1-5090-2870-2. Dostupné z: <https://doi.org/10.1109/INDIN.2016.7819127>.
- [16] DRAHOS, Peter; KUCERA, Erik; HAFFNER, Oto a KLIMO, Ivan. Trends in industrial communication and OPC UA. Online. In: 2018 Cybernetics & Informatics (K&I). IEEE, 2018, s. 1-5. ISBN 978-1-5386-4419-5. Dostupné z: <https://doi.org/10.1109/CYBERI.2018.8337560>.
- [17] VOJÁČEK, Antonín. Průmyslová komunikace OPC UA - 1.díl - popis protokolu. Automatizace.hw.cz [online]. 2020, 22.07.2020 [cit. 2024-04-06]. Dostupné z: <https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html>
- [18] REAL TIME AUTOMATION. AN INTRODUCTION TO OPC UA. Online. Real Time Automation. 2019. Dostupné z: <https://www.rtautomation.com/technologies/opcua/>. [cit. 2024-04-06].
- [19] OPC ROUTER. What is OPC UA? A practical introduction. OPC ROUTER [online]. 2019[19] [cit. 2024-04-06]. Dostupné z: <https://www.opc-router.com/what-is-opc-ua/>
- [20] Service Set model. OPC FOUNDATION [online]. 2006 [cit. 2024-04-07]. Dostupné z: <https://reference.opcfoundation.org/Core/Part4/v104/docs/4.1>
- [21] FOXON. CO JE OPC? OPC SERVER? OPC KLIENT? FOXON [online]. 2013 [cit. 2024-04-06]. Dostupné z: <https://foxon.cz/blog/prakticka-teorie/159-co-je-opc-opc-server-opc-klient>

- [22] ABB. RobotStudio® Suite. ABB. ABB [online]. 2024 [cit. 2024-04-07]. Dostupné z: <https://new.abb.com/products/robotics/cs/robotstudio>
- [23] RobotStudio Operating Manual. 3HAC032104-001. ABB, 2023.
- [24] UNITRONICS. What is the definition of "PLC"? UNITRONICS [online]. 2006 [cit. 2024-04-07]. Dostupné z: <https://www.unitronicsplc.com/what-is-plc-programmable-logic-controller/>
- [25] ZOLA, Andrew. Programmable logic controller (PLC). TechTarget [online]. 2024 [cit. 2024-04-07]. Dostupné z: <https://www.techtarget.com/whatis/definition/programmed-logic-controller-PLC>
- [26] PLC. PAESSLER [online]. 2024 [cit. 2024-04-07]. Dostupné z: <https://www.paessler.com/it-explained/plc>
- [27] Cyklus PLC. PLC AUTOMATIZACE [online]. 2010 [cit. 2024-04-07]. Dostupné z: <http://www.plc-automatizace.cz/knihovna/plc/plc-cyklus.htm>
- [28] SESTAVA PLC. PLC AUTOMATIZACE [online]. 2010 [cit. 2024-04-07]. Dostupné z: <http://www.plc-automatizace.cz/knihovna/plc/plc-hw-sestava.htm>
- [29] DIVER, Brian. Components of PLCs. Industrial Automation co. [online]. 2023 [cit. 2024-04-07]. Dostupné z: <https://industrialautomationco.com/blogs/news/components-of-plcs>
- [30] Components of PLC. Instrumentationtools.com [online]. 2020 [cit. 2024-04-07]. Dostupné z: <https://instrumentationtools.com/components-of-plc/>
- [31] MORTENSON, Ted. PLC Hardware Explained. RealParts [online]. 2020 [cit. 2024-04-07]. Dostupné z: <https://www.realpars.com/blog/plc-hardware>
- [32] PHILLIPS, Rick. Basic PLC Programming – How to Program a PLC using Ladder Logic (for Beginners). Basicplc.com [online]. 2020 [cit. 2024-04-07]. Dostupné z: <https://basicplc.com/plc-programming/>
- [33] Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 5). AUTOMA. 2012, 2012(1), 29-31. ISSN 1210-9592.
- [34] What are the Basic Components of a PLC System? PLC Table [online]. 2020 [cit. 2024-04-07]. Dostupné z: <https://www.plctable.com/what-are-the-basic-components-of-a-plc-system/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

IoT	Internet of Things.
OPC UA	Open Platform Communications Unified Architecture
M2M	Machine - to - machine.
COM	Component Object Model
DCOM	Distributed Component Object Model
SOAP	Simple Object Access Protocol
PLC	Programmable Logic Controller
PII	Process Input Image
POI	Process Output Image
LSV	Local Speed Vault
HMI	Human Machine Interface

SEZNAM OBRÁZKŮ

Obrázek 1: Rámec technologií a nástrojů v konstrukci modelů [4]	10
Obrázek 2: Typy digitálních dvojčat [7]	11
Obrázek 3: Model digitálního dvojčete [8]	13
Obrázek 4: Posuvné paletové regály [9]	14
Obrázek 5: Posuvný regál s manuálním posuvem [11]	14
Obrázek 6: OPC UA logo [17]	15
Obrázek 7: Princip komunikace OPC UA [18]	16
Obrázek 8: OPC UA model [17]	17
Obrázek 9: Služby zjišťování [20]	18
Obrázek 10: Služby SecureChannel [20]	18
Obrázek 11: Služby relací [20]	19
Obrázek 12: Služby NodeManagement [20]	19
Obrázek 13: Sada služeb View a Query [20]	19
Obrázek 14: Sada atributových služeb [20]	20
Obrázek 15: Služby Metod [20]	20
Obrázek 16: Služby MonitoredItem a Subscription [20]	21
Obrázek 17: Aplikace na lokální PC stanici [21]	21
Obrázek 18: Aplikace v rámci sítě Ethernet [21]	22
Obrázek 19: Rozsáhlá aplikace OPC UA [21]	22
Obrázek 20: Prostředí RobotStudia	23
Obrázek 21: Smart Componenty	24
Obrázek 22: LineSensor	24
Obrázek 23: Základní cyklus PLC [27]	27
Obrázek 24: PLC programovací jazyky – IEC 61 131 – 3 [32]	28
Obrázek 25: Ladder Logic [32]	29
Obrázek 26: Structured Text [32]	29
Obrázek 27: Function Block Diagram [32]	29
Obrázek 28: Sequential Function Chart [32]	30
Obrázek 29: Instruction List [32]	30
Obrázek 30: Dílčí pohled na zařízení [9]	33
Obrázek 31: Klíčový spínač provozních režimů [9]	33
Obrázek 32: Obslužná jednotka [9]	34

Obrázek 33: Obslužný panel [9]	35
Obrázek 34: Přístupové světelné závory [9]	35
Obrázek 35: Bezpečnostní nožní světelná závora [9]	36
Obrázek 36: Koncový spínač [9]	36
Obrázek 37: Diagnostika na HMI panelu PLC	37
Obrázek 38: Rozdělení vozíku.....	38
Obrázek 39: Sekvence provozu frekvenčních měničů.....	41
Obrázek 40: Modely regálového systému	43
Obrázek 41: Smart Component Simulace.....	43
Obrázek 42: Smart Componenty pro generování regálového systému.....	44
Obrázek 43: Smart Componenty pro posunutí posledního regálu	45
Obrázek 44: Smart Componenty pro vytvoření vstupních světelných závor	45
Obrázek 45: Smart Component Mode_Selector	45
Obrázek 46: Snímání kliknutí na přepínač provozních režimů	46
Obrázek 47: Zobrazení přepínače provozních režimů ve správné pozici a orientaci ..	47
Obrázek 48: Klíčový přepínač provozních režimů	47
Obrázek 49: Smart Component Carriage.....	48
Obrázek 50: Smart Component VolumeSensor	49
Obrázek 51: Smart Componenty sensorů přiblížení a dorazů	49
Obrázek 52: Senzory nožních světelných závor	50
Obrázek 53: Smart Component bit_to_word	50
Obrázek 54: Převod z binárního čísla na dekadické	51
Obrázek 55: Smart Component Left_buttons	52
Obrázek 56: Ovládací tlačítka vozíku.....	52
Obrázek 57: Smart Componenty ovládacích tlačítek vozíku.....	53
Obrázek 58: Tlačítka nouzového zastavení	53
Obrázek 59: Odezva simulovaných frekvenčních měničů	54
Obrázek 60: Určení stavů pro pohyb vozíku	55
Obrázek 61: Smart Componenty pro simulaci akcelerace vozíku	55
Obrázek 62: Realizace pohybu vozíku	56
Obrázek 63: Průběh rychlosti vozíku.....	57
Obrázek 64: Průběh otevírání uličky regálového systému	58
Obrázek 65: Konfigurace OPC UA klienta	59

Obrázek 66: Ukázka vstupů a výstupu Smart Componentu UpcUaClient	60
Obrázek 67: Rozbalení projektu do RobotStudia	71
Obrázek 68: Postup pro vytvoření regálového systému – 1	72
Obrázek 69: Proměnné pro vytvoření regálového systému	72
Obrázek 70: Tlačítko Execute pro vytvoření regálového systému	73
Obrázek 71: Informace o připojení na PLC skrze OPC UA.....	73
Obrázek 72: Zapnutí simulace v RobotStudios	73
Obrázek 73: Ukázka vygenerovaného systému	74

SEZNAM TABULEK

Tabulka 1: Popis vstupů a výstupů tlačítek na vozíku.....	39
Tabulka 2: Výstupy z PLC pro pohyb systému	39
Tabulka 3: Vstupy z PLC pro pohyb systému	40
Tabulka 4: Popis komunikace mezi PLC a frekvenčním měničem	42

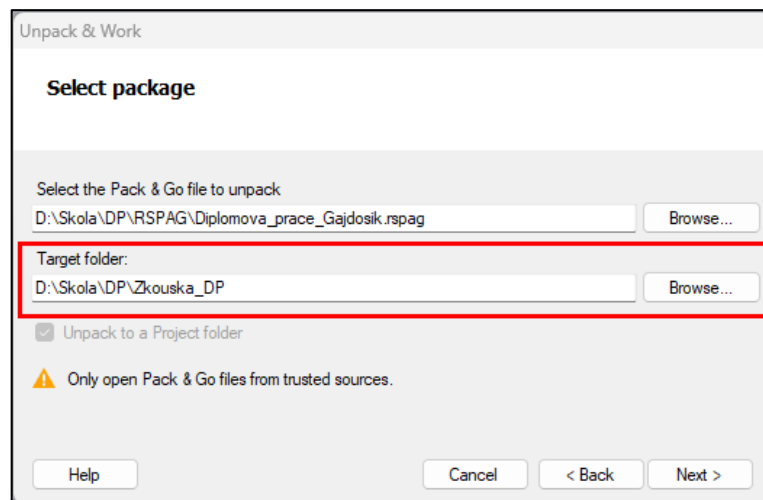
SEZNAM PŘÍLOH

- P1 Testovací postup pro ověření funkčnosti systému
- P2 Obsah CD

PŘÍLOHA P I: TESTOVACÍ POSTUP PRO OVĚŘENÍ FUNKČNOSTI SYSTÉMU

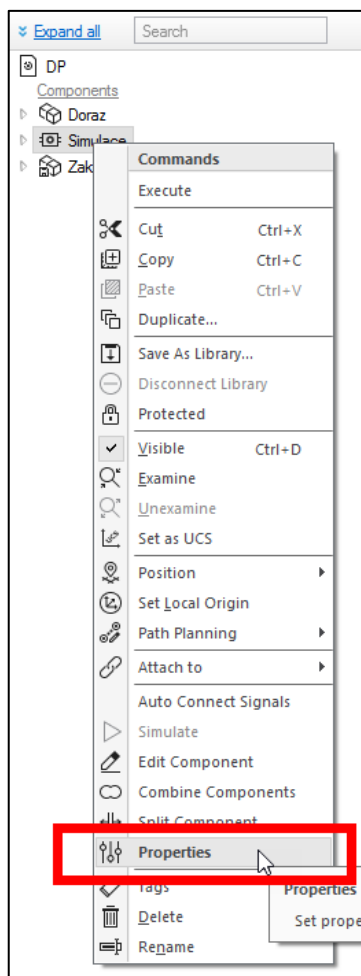
Testovací postup od nainstalování programového prostředí RobotStudio až po spuštění simulace příkazu otevření uličky:

1. Instalace programu RobotStudio, které je dostupné ke stažení na webových stránkách ABB.
2. Otevření souboru `Diplomova_prace_Gajdosik.rspag`, který je součástí této práce. Po otevření souboru Vás RobotStudio požádá, abyste vybrali cestu, kde chcete projekt rozbalit.



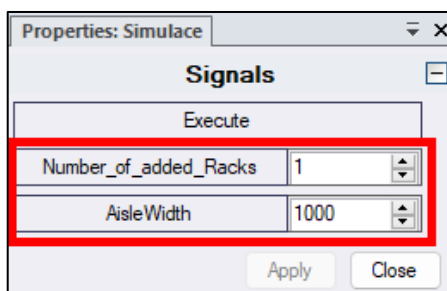
Obrázek 67: Rozbalení projektu do RobotStudia

3. Spuštění PLC simulátoru, který není součástí této práce, a který dodala firma SSI společně s podklady pro tvorbu digitálního dvojčete.
4. Nyní by měl být vidět v RobotStudiosu systém s počátečním, koncovým a jedním pohyblivým regálem. Pro vytvoření regálového systému podle konfiguračních souborů, které určují funkci PLC simulátoru je nutné tyto parametry pouze zadat do RobotStudia. Pro zadání parametrů je tedy nutné:
 - a. V levé části RobotStudia v sekci Layout pravým tlačítkem kliknout na Simulace a vybrat Properties



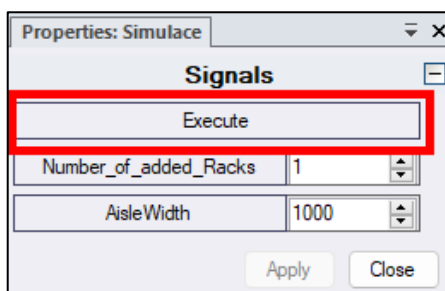
Obrázek 68: Postup pro vytvoření regálového systému – 1

- b. V levé horní části RobotStudia se zobrazí sekce s názvem Properties: Simulace, kde lze Vyplnit proměnné Number_of_Racks a AisleWidth. Proměnná Number_of_Racks udává počet pohyblivých vozíků a proměnná AisleWidth udává šířku uličky v [mm].



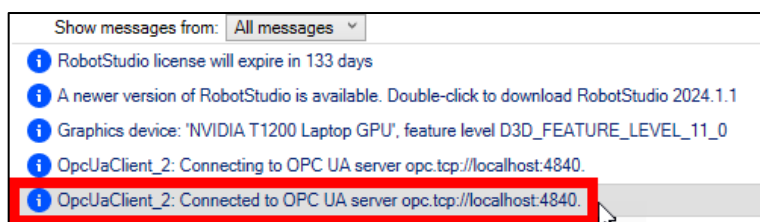
Obrázek 69: Proměnné pro vytvoření regálového systému

- c. Následně je nutné kliknout na tlačítko Execute, které se nachází nad proměnnými Number_of_Racks a AisleWidth.



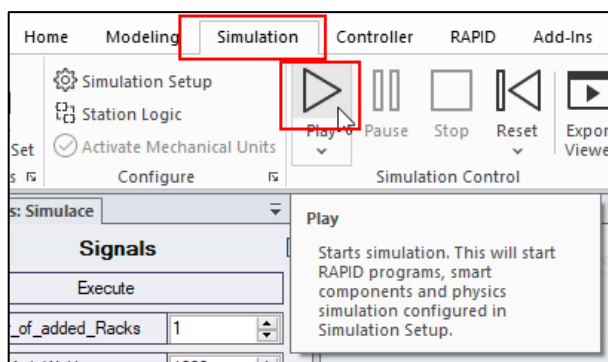
Obrázek 70: Tlačítko Execute pro vytvoření regálového systému

5. V sekci Output, která se nachází v dolní části RobotStudia by měla být poslední informace o stavu připojení OPC UA klienta na server. V této hlášce musí být informace o tom, že je klient připojen na server. Vše se provede automaticky, když je spuštěn simulátor PLC.

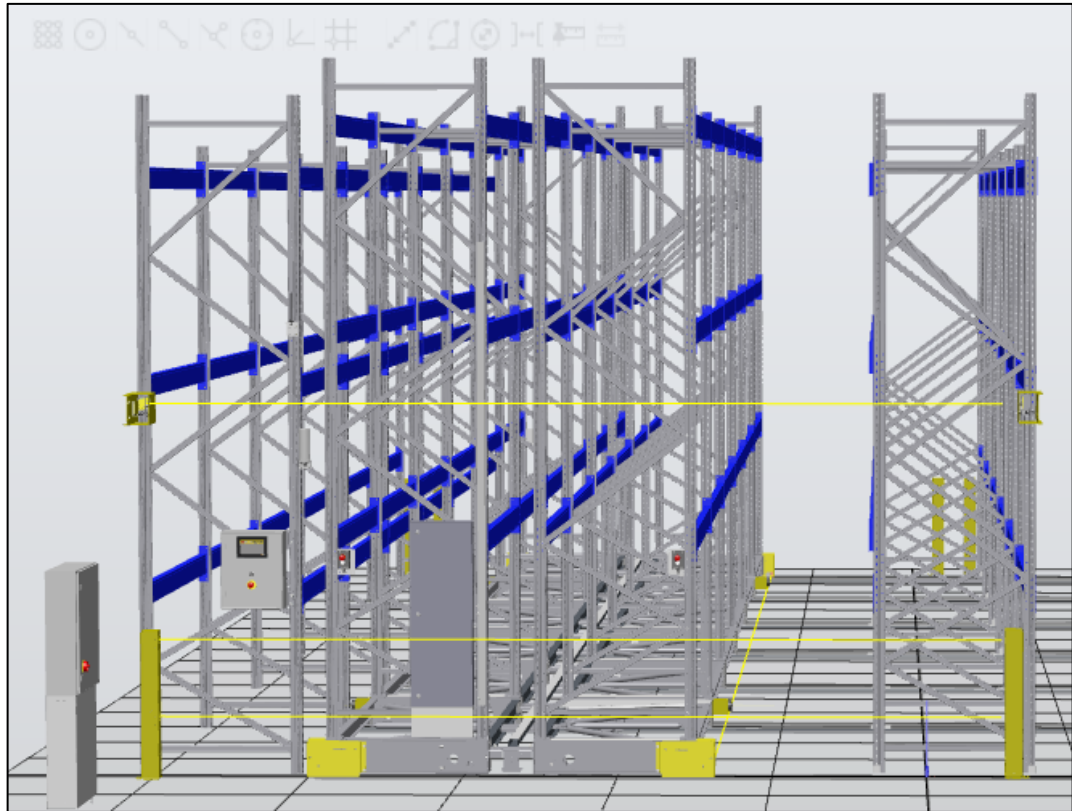


Obrázek 71: Informace o připojení na PLC skrze OPC UA

6. Poslední krok je spuštění simulace v prostředí RobotStudia. Stačí pouze vybrat v horní části sekci Simulation a následně kliknout na tlačítko play.



Obrázek 72: Zapnutí simulace v RobotStudiu



Obrázek 73: Ukázka vygenerovaného systému

PŘÍLOHA P 2: OBSAH CD

Struktura obsahu přiloženého CD:

- Adresář Text diplomové práce – obsahuje text diplomové práce ve formátu PDF/A
- Adresář Simulace – obsahuje RSPAG stanici do RobotStudia s digitálním dvojčetem paletového posuvného regálu