



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ŘÍZENÍ A VIZUALIZACE VIRTUÁLNÍ VÝROBNÍ LINKY

CONTROL AND VISUALISATION OF A VIRTUAL PRODUCTION LINE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Dominik Viater

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Radek Štohl, Ph.D.

BRNO 2023

Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Dominik Viater

ID: 204765

Ročník: 2

Akademický rok: 2022/23

NÁZEV TÉMATU:

Řízení a vizualizace virtuální výrobní linky

POKYNY PRO VYPRACOVÁNÍ:

- 1) Seznamte se a popište vývojový systém Factory I/O.
- 2) Navrhnete a realizujete demonstrační úlohu řízení a vizualizace virtuální výrobní linky.
- 3) Vytvořte potřebné softwarové vybavení pro PLC řady LOGIX.
- 4) Vytvořte potřebnou vizualizaci na HMI panel podle standardu ANSI/ISA-101.01-2015.
- 5) Ověřte své řešení.

DOPORUČENÁ LITERATURA:

Logix 5000 Controllers Tasks, Programs, and Routines. Programming Manual. Rockwell Automation. 1756-PM005H-EN-P. 2018.

ANSI/ISA-101.01-2015, Human Machine Interfaces for Process Automation Systems.

FactoryTalk Alarms and Events System Configuration Guide. Rockwell Automation. FTAE-RM001O-EN-E. 2022.

Termín zadání: 6.2.2023

Termín odevzdání: 17.5.2023

Vedoucí práce: Ing. Radek Štohl, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Táto diplomová práca sa zameriava na riadenie virtuálnej výrobnéj linky pomocou programu FACTORY I/O a PLC. Úvod práce sa venuje popisu programu FACTORY I/O a jeho možností úpravy scény a manipulácie s dielmi. Ďalej sa zaoberá návrhom a implementáciou virtuálnej výrobnéj linky, pre ktorú bolo vytvorené softvérové vybavenie pre riadenie PLC a vizualizácia na HMI paneli. Cieľom práce je poskytnúť nástroj na lepšie porozumenie a ovládanie PLC prostredníctvom virtuálnych výrobných liniek.

KĽÚČOVÉ SLOVÁ

Factory I/O, PLC, LOGIX, HMI, simulácia, virtuálna výrobná linka, stavový diagram, ANSI/ISA-101.01-2015, demonštračná úloha, sklad, zakladač paliet, zakladač balíkov, palety, dopravník

ABSTRACT

This thesis focuses on the control of a virtual production line using FACTORY I/O and PLC. The introduction of the thesis is devoted to the description of the FACTORY I/O program and its scene editing and part manipulation capabilities. It then discusses the design and implementation of a virtual production line, for which PLC control software and HMI panel visualization were developed. The aim of the thesis is to provide a tool to better understand and control PLCs through virtual production lines.

KEYWORDS

Factory I/O, PLC, LOGIX, HMI, simulation, virtual production line, state diagram, ANSI/ISA-101.01-2015, demonstration task, warehouse, pallet stacker, bale stacker, pallets, conveyor

VIATER, Dominik. *Řízení a vizualizace virtuální výrobní linky*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2023, 63 s. Diplomová práce. Vedúci práce: Ing. Radek Štohl, Ph.D.

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora:	Bc. Dominik Viater
VUT ID autora:	204765
Typ práce:	Diplomová práca
Akademický rok:	2022/23
Téma záverečnej práce:	Řízení a vizualizace virtuální výrobní linky

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podpisuje iba v tlačenej verzii.

POĎAKOVANIE

Chcel by som poďakovať svojmu vedúcemu diplomovej práce Ing. Radkovi Štoholvi, PhD. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy pre túto prácu.

Obsah

Úvod	12
1 FACTORY I/O	13
1.1 Popis programu	14
1.1.1 Panel s nástrojmi	14
1.1.2 Paletové okno	15
1.1.3 Stavový panel	15
1.2 Navigácia	16
1.2.1 Orbitálna kamera	17
1.2.2 Fly kamera	17
1.2.3 First Person kamera	18
1.2.4 Okno s kamerami	18
1.2.5 Gizmo kamera	18
1.2.6 Sledovanie dielov	19
1.3 Vytváranie scény	20
1.4 Úprava scény	20
1.5 Spustenie simulácie	23
1.6 Ovládanie simulácie	24
1.7 Ovládanie scény pomocou PLC	26
1.8 Užitočné skratky	29
2 Návrh úlohy	31
2.1 Zakladače a sklady	32
2.2 Zakladač balíkov	33
2.3 Bezpečnosť	34
2.4 Nakonfigurovanie komunikácie	35
3 Riadiaci softvér	39
3.1 Hlavný program	39
3.2 Generovanie paliet	40
3.3 Vykladanie paliet	42
3.4 Zakončenie vykladania	43
3.5 Zakladanie balíkov	44
3.6 Simulácia	45
4 HMI	47
4.1 Štandard ANSI/ISA-101.01-2015	47
4.2 Navigačná schéma medzi obrazovkami	50

4.3	Prepojenie obrazoviek v mojom návrhu	51
4.3.1	Hlavná obrazovka	52
4.3.2	Prvý sklad	53
4.3.3	Zakladač balíkov	54
4.3.4	Nastavenie pohľadov	55
5	Overenie riešenia	59
	Záver	61
	Literatúra	62

Zoznam obrázkov

1.1	Typický vzhľad fabriky [1]	13
1.2	Tabuľka so symbolmi	14
1.3	Paletové okno	15
1.4	Stavový panel	16
1.5	Typy kamier	16
1.6	Orbitálna kamera	17
1.7	First Person kamera	18
1.8	Okno s kamerami	19
1.9	Gizmo kamera	19
1.10	Dostupné scény	20
1.11	Príklad zlého vloženia	21
1.12	Výber dielov na úpravu	21
1.13	Vertikálny presun dielov	22
1.14	Kontextové menu	23
1.15	Úprava rýchlosti simulácie	24
1.16	Zobrazenie označení	25
1.17	Ovládanie simulácie [14]	25
1.18	Nastavenie ovládačov	26
1.19	Nastavenie ovládačov pomocou stavového riadku	27
1.20	Dostupné ovládače	27
1.21	Konfigurácia	28
1.22	Konfiguračný panel	28
1.23	Konfiguračný panel	29
1.24	Export konfigurácie	29
2.1	Návrh linky	31
2.2	Ukážka pozícií skladu	32
2.3	Spojenie vykladania	33
2.4	Balíky uložené na paletu	34
2.5	Bezpečnostné prvky	34
2.6	RSLinx Classic	35
2.7	Konfigurácia	36
2.8	Overenie komunikácie	38
3.1	Ukážka hlavného programu	40
3.2	Stavový diagram pre generovanie paliet	41
3.3	Ukážka kódu zaplnenia	42
3.4	Stavový diagram pre vykladanie paliet	43
3.5	Stavový diagram pre zakončenie vykladania	44

3.6 Stavový diagram pre zakladanie balíkov	45
3.7 Stavový diagram pre zakladanie balíkov	46
4.1 Usporiadanie HMI	48
4.2 Usporiadanie objektov v HMI	49
4.3 Tabulka odporúčaného prístupu a navigácie	51
4.4 Diagram vzťahov	52
4.5 Hlavná obrazovka	53
4.6 Prvý sklad	54
4.7 Zakladač balíkov	55
4.8 Nastavenie pohľadu	56
4.9 Pravá časť prvého skladu	57
4.10 Zobrazenie palety	58
5.1 Pripojený hardvér	59
5.2 Pripojený hardvér	60

Zoznam tabuliek

1.1	Možné dátové typy 13	24
1.2	Skratky	30
2.1	Premenné typu BOOL	37
2.2	Premenné typu INT	37

Úvod

Medzi najrozšírenejšie metódy ovládania liniek v súčasnosti patrí PLC. Preto je výhodné, aby vznikali nástroje na lepšie pochopenie práce s nimi. Jedným z týchto nástrojov na výuku je aj FACTORY I/O. Tento program slúži na vytváranie virtuálnych výrobných liniek a ich následné ovládanie pomocou riadiacich ovládačov, ako je napríklad PLC.

Cieľom tejto diplomovej práce je riadenie virtuálnej výrobnéj linky. V úvode práca popisuje program FACTORY I/O. Zaoberá sa možnými spôsobmi upravovania scény, pridávaním, otáčaním dielov a podobne. Ďalej sa sústreďí na návrh a realizáciu virtuálnej výrobnéj linky. Vytvorím softvérové vybavenie potrebné na riadenie PLC spolu s vizualizáciou pre HMI panel.

V prvej kapitole bude predstavený program FACTORY I/O, v ktorom bude navrhovaná virtuálna výrobná linka. Budú popísané všetky časti programu a ich účel. Taktiež budú opísané všetky možnosti navigácie v scéne kvôli ich špecifickému využitiu. Budú predstavené aj možnosti vytvárania a úpravy scén potrebné na prácu s programom FACTORY I/O. Popísané bude aj spustenie a následné ovládanie simulácie v reálnom čase. Pre ovládanie scény pomocou PLC bude ukázaný postup na spojenie FACTORY I/O s externým riadiacim ovládačom.

V druhej kapitole bude predstretý návrh laboratórnej úlohy. Laboratórna úloha bude pozostávať z automatizovaného skladu s prvkami výroby. Úlohou bude automaticky ukladať a vykladať palety do regálov. Pri vykladaní bude potrebné vyriešiť prioritu vykladania. Predstavím návrh výrobného prvku zakladača balíkov. Ďalšou súčasťou návrhu bude zabezpečenie virtuálnej výrobnéj linky a opíšem aj nakonfigurovanie komunikácie.

V tretej kapitole popíšem riadiaci softvér. Opíšem jeho rozdelenie na časti a vysvetlím, ako som ich používal a prečo. Pridám aj stavové diagramy pre lepšie pochopenie fungovania programu.

V štvrtej kapitole predstavím štandard ANSI/ISA-101.01-2015. Popíšem jeho hlavné myšlienky a prvky, ktoré som použil v mojom návrhu HMI panelu. Ukážem aj prepojenie medzi obrazovkami, každú z nich opíšem a zobrazím pre dobre pochopenie. Poslednou kapitolou je piata kapitola. V tejto kapitole overím funkčnosť návrhu.

1 FACTORY I/O

FACTORY I/O je program na vytváranie 3D simulácií továrne. Jeho hlavným účelom je výuka automatizačných technológií, ale môže byť použitý aj pri mnohých iných aplikáciách. Môžeme napríklad vytvoriť náhľad fabriky pre lepšiu predstavu a otestovať funkčnosť chodu linky pred jej spustením do prevádzky. **FACTORY I/O** je navrhnutý tak, aby sa dal ľahko používať. Umožňuje rýchlo vybudovať virtuálnu továreň pomocou prvkov inšpirovaných skutočnou továrňou. Tieto prvky môžu pozostávať z viacerých druhov ako napríklad: dopravníkov, snímačov a škatúl. Obsahuje aj mnoho scén inšpirovaných typickými priemyselnými aplikáciami, ako sú napríklad výrobné linky s robotickou rukou alebo nádoby na ohrev tekutiny. Jeden z príkladov je možné vidieť na obrázku 1.1.

Najbežnejším scenárom je použitie **FACTORY I/O** ako školiacej platformy PLC, pretože PLC sú najviac používané riadiace jednotky v priemyselných aplikáciách. Môže sa však použiť aj s mikrokontrolérmi, SoftPLC, Modbus a mnohými ďalšími technológiami. [1]
















Obr. 1.1: Typický vzhľad fabriky [1]

1.1 Popis programu

Program FACTORY I/O má užívateľské rozhranie, ktoré sa skladá z viacerých častí. Hlavné okno programu je zobrazené na obrázku 1.1 vyššie. Môžeme na ňom vidieť štyri hlavné oblasti. Prvou a zároveň najväčšou je tá v strede, kde sa nachádza aktuálna virtuálna výrobná linka. Pre ovládanie tejto časti sú potrebné ďalšie tri a to sú:

1.1.1 Panel s nástrojmi

Jedná sa o hornú lištu, v ktorej sú všetky dôležité nastavenia ovládania. V ľavej časti je uvítacie menu, file, edit a view. Pomocou tlačidla Uvítacie menu je možné prísť k dokumentácii, tutoriálom, scénam a aktualizáciám. Tlačidlo Súbor(File) slúži na otvorenie nového projektu alebo už vytvoreného projektu. Jeho hlavnou úlohou je však prístup k Nastaveniam a Správe ovládačov(Driverov). Nastavovanie ovládačov je dôležité pri prepojení simulácie s reálnym PLC, a preto ho budem popisovať v podkapitole Prepojenia Factory I/O a PLC. Úprava(Edit) slúži na úpravu vykonaných zmien a Náhľad(View) na zobrazenie potrebných vecí v editore. V pravej časti sú umiestnené potrebné tlačidlá na ovládanie simulácie a scény. Je tam napríklad tlačidlo na spustenie simulácie, pozastavenie simulácie, prepínanie kamier, ale aj tlačidlá na zobrazenie paletového okna. Tlačidlo na spustenie simulácie slúži na prepínanie z módu editora do simulácie. Pre lepšiu orientáciu som v nasledujúcej tabuľke popísal význam jednotlivých symbolov.

Symbol	Popis	Symbol	Popis	Symbol	Popis
	Uvítacie menu		Orbitálna kamera		Označenie senzora
	Spustiť/upraviť		Fly kamera		Označenie akčných členov
	Pauza		First person kamera		Paleta
	Reset		Sledovanie dielu		
	Rýchlosť prehrávania		Okno s kamerami		

Obr. 1.2: Tabuľka so symbolmi

1.1.2 Paletové okno

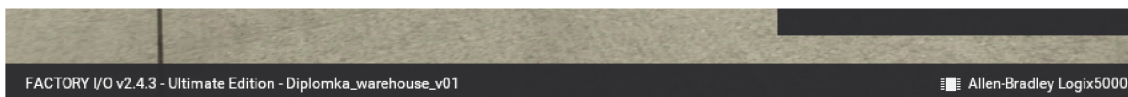
Je okno, ktoré sa nachádza vpravo. Toto okno je spúšťané z hlavnej lišty (Panel s nástrojmi), ako už bolo spomenuté v predchádzajúcej podkapitole, a to vpravo hore. Nájde sa v ňom už spomínané bežné prvky továrne: dopravníky, bedne, snímače a dokonca aj kamery. V hornej časti je možné selektovať produkty podľa kategórií, pri výbere kategórie Senzory budú zobrazené iba senzory. Z toho menu môžeme následne vyberať a používať potrebné prvky v simulácii. Pre názornú ukážku výberu kategórie snímačov a vzhľad paletového okna vid. obrázok 1.3. [2]



Obr. 1.3: Paletové okno

1.1.3 Stavový panel

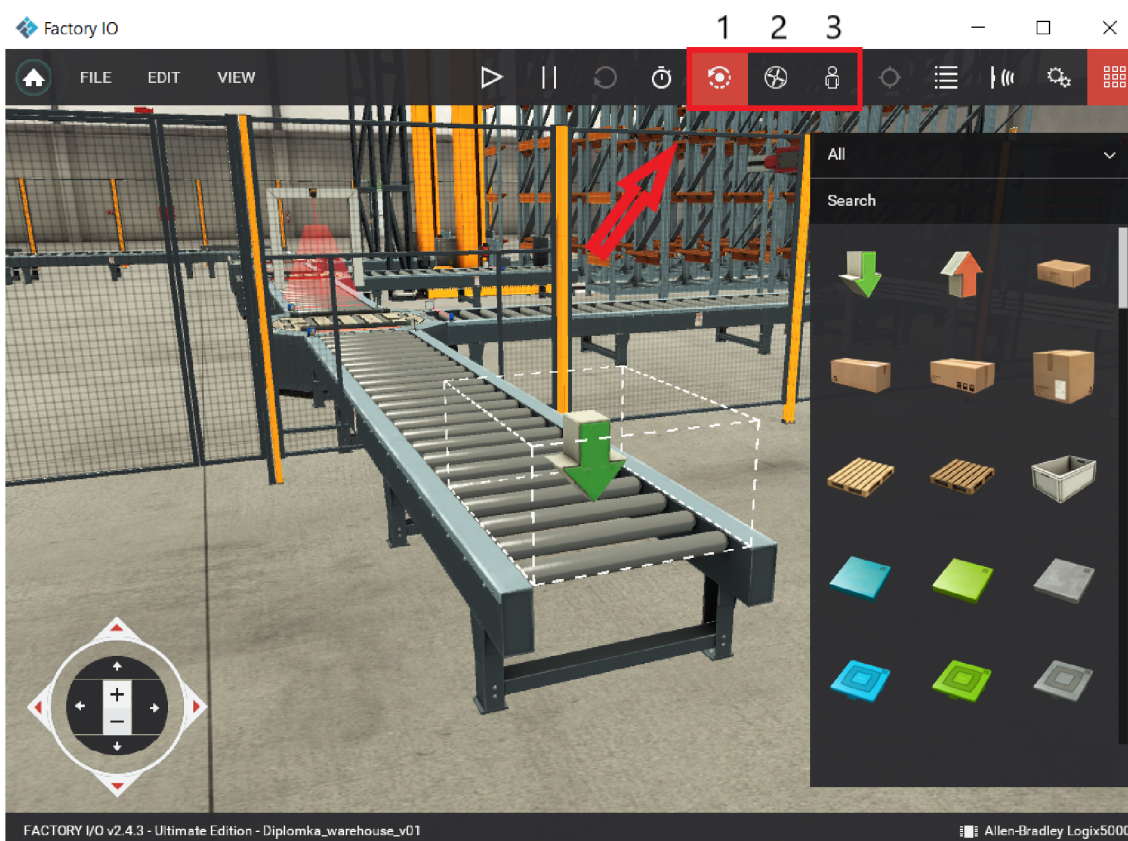
Jedná sa o dolnú lištu, v ktorej sa nachádzajú informácie o aktuálnom projekte. V ľavom rohu je názov projektu, ktorý je práve otvorený. V pravom rohu zase názov Ovládača (Drivera), s ktorým sa bude výrobná linka ovládať.



Obr. 1.4: Stavový panel

1.2 Navigácia

Navigácia v FACTORY I/O je možná pomocou kamier. Kamery sú kľúčovým prvkom vo FACTORY I/O. Pri každej akcii je dôležité poznať účel kamery, ktorá sa práve využíva. Okrem hlavného využitia pre navigáciu v 3D priestore sa tiež používajú pri úpravách scén, vytváraní scén, interakcii s dielmi a pod. Bolo vytvorených viac druhov kamier, aby bolo možné vyťažiť maximum z FACTORY I/O. Veľmi dôležité je aj, aby sa užívateľ cítil komfortne pri ich používaní. Avšak treba mať na pamäti aj fakt, že tieto kamery majú svoj primárny účel.^[3]

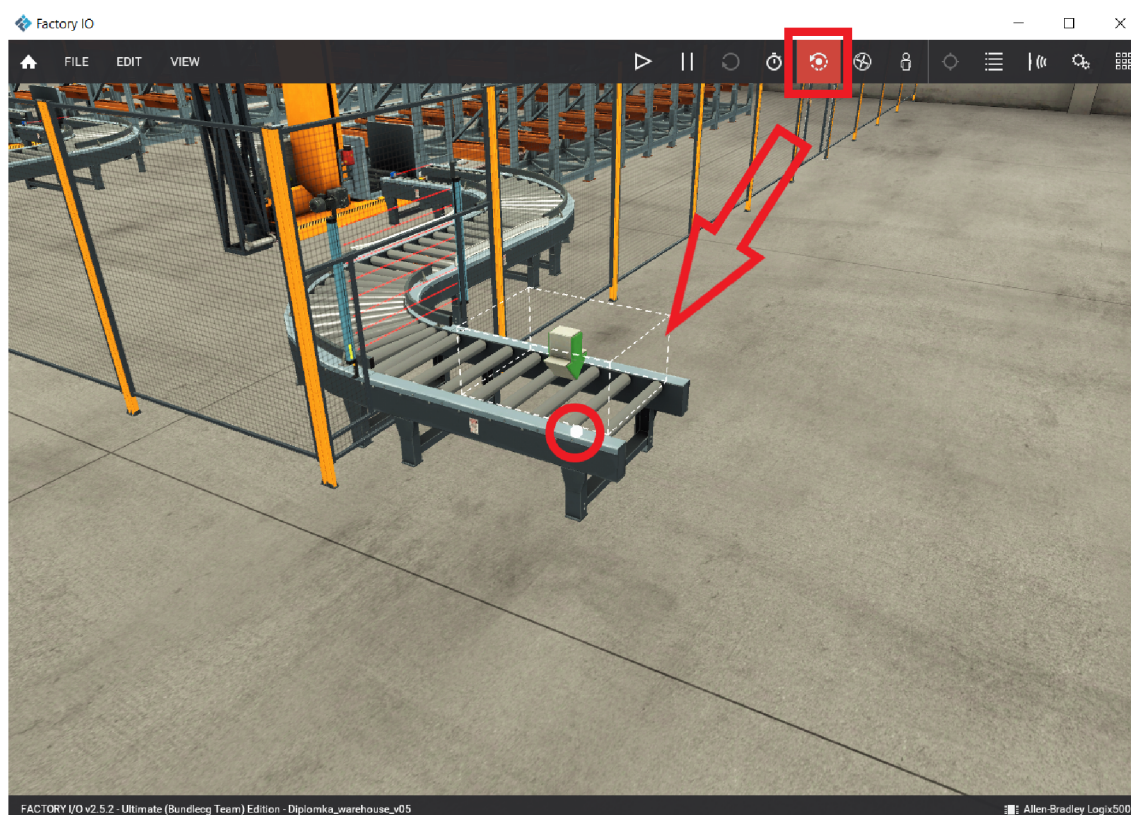


Obr. 1.5: Typy kamier

1.2.1 Orbitálna kamera

Orbitálna kamera je predvolená kamera a jej hlavným účelom je vytváranie scény. Toto je jediná kamera, ktorá umožňuje pohybovať sa aj cez diely bez toho, aby ste do nich narazili.

Funguje tak, že sa otáča okolo bodu záujmu, ktorý je označený bielou bodkou, viď obrázok 1.6. Tento bod definujeme dvojitým kliknutím ľavým tlačidlom myši na diel. Ak je už bod definovaný, je možné sa okolo neho pohybovať. Pohyb vykonávame takým spôsobom, že stlačíme pravé tlačidlo myši a ťaháme. Výhodou pri definovaní toho bodu je, že pri pridaní dielu z paletového okna sa pridá do úrovne bodu a nemusíme ho vertikálne posúvať. Neplatí to však pri dieloch, ktoré sú typicky na zemi ako napríklad dopravníky alebo stanice. [4]



Obr. 1.6: Orbitálna kamera

1.2.2 Fly kamera

Fly kamera slúži na voľný pohyb v 3D priestore. Tento typ kamery koluduje s dielami scény, ale snímače ho nerozpoznajú. Táto kamera je jednoduchšia na ovládanie po priestore pre mnoho užívateľov, pretože je viac intuitívna a je možné sa s ňou stretnúť pri niektorých hrách. [5]

1.2.3 First Person kamera

First Person kamera sprostredkováva pohľad osoby vysokej 1,8 m na fabriku. Simuluje aj jej pohyb vo virtuálnej továrni a tým pádom naráža do dielov. Ak je v pôvodných nastaveniach, tak ju nevidia senzory. Názornú ukážku môžete vidieť na obrázku 1.7. [6]



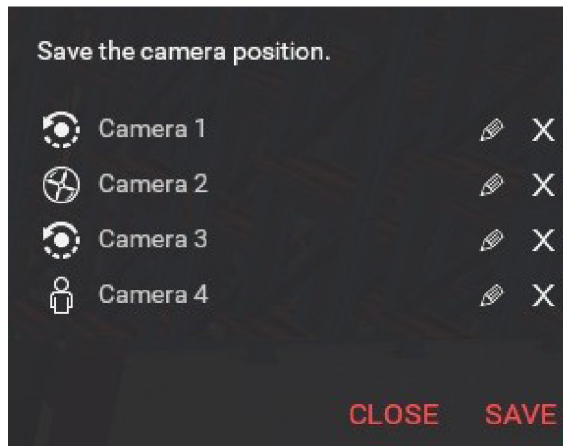
Obr. 1.7: First Person kamera

1.2.4 Okno s kamerami

Okno s kamerami slúži na uloženie aktuálnej polohy kamery. Pomocou tohto okna môžeme ľahko pristupovať k predtým uloženému pohľadu a jednoducho zmeniť perspektívu. Toto okno je spúšťané z Panela s nástrojmi. Je to ikonka, ktorá je štvrtou vpravo hore. Pre prepínanie sa medzi uloženými kamerami je potrebné dvakrát kliknúť ľavým tlačidlom myši alebo stlačiť Page Up / Page Down. Uložené pozície je možné kedykoľvek upraviť a vymazať. Okno s kamerami je na nasledujúcom obrázku 1.8. [7]

1.2.5 Gizmo kamera

Táto kamera ponúka väčšinu z už spomínaných funkcií v podkapitole 1.2. Gizmo kamera je na obrázku 1.9 a skladá sa z troch vrstiev. Prvá vonkajšia slúži na posúvanie po scéne. Druhá na natočenie scény okolo bodu záujmu a tretia na priblíženie



Obr. 1.8: Okno s kamerami

a vzdialenie. Na sprístupnenie kamery je potrebné na Paneli s nástrojmi vo View povoliť jej zobrazenie. [8]



Obr. 1.9: Gizmo kamera

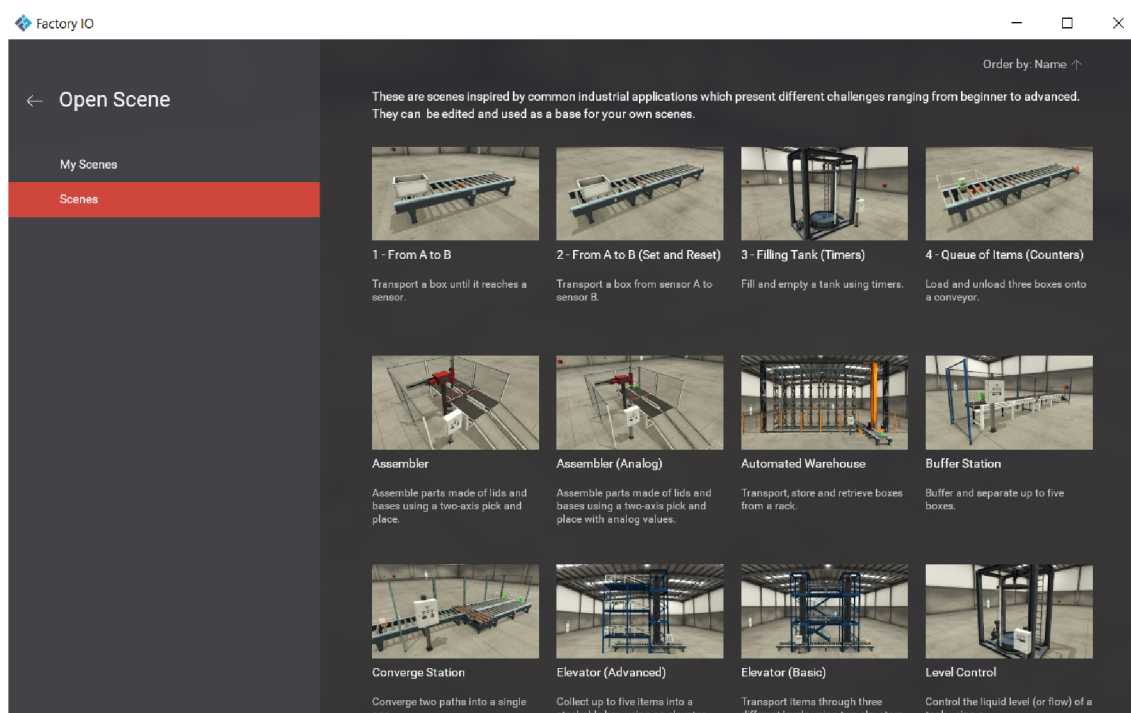
1.2.6 Sledovanie dielov

Tento nástroj umožňuje orbitálnej kamere sledovať diel. Je to skvelý nástroj na sledovanie sledu udalostí a kontrolu možných porúch. Ak chceme začať sledovať diel, vyberieme na Paneli s nástrojmi ikonu Sledovať diel. Následne klikneme ľavým tlačidlom myši na diel, ktorý chceme sledovať. Ak chceme zastaviť sledovanie dielu, dvakrát klikneme ľavým tlačidlom myši kdekoľvek v scéne alebo stlačíme Esc. [9]

1.3 Vytváranie scény

Pri vytváraní scény vo Factory I/O máme dve možnosti a to vytvoriť ju od začiatku alebo použiť už predpripravené scény. Tieto scény inšpirované typickými priemyselnými aplikáciami sú rôznej náročnosti, a to od začiatočníckych až po pokročilé úrovne obtiažnosti. Úroveň obtiažnosti nie je však presne definovaná, pretože závisí od náročnosti požadovaného riadiaceho algoritmu. [10]

Predpripravené scény je možné nájsť na Paneli s nástrojmi, a to dvoma spôsobmi. Prvý je pomocou ikony Uvítacieho menu a následne z výberu zvolením scény. Druhý je prístup cez súbor a stlačením možnosti otvoriť. Pri oboch možnostiach sa zobrazia moje scény a typické scény, ako je možné vidieť na obrázku 1.10.



Obr. 1.10: Dostupné scény

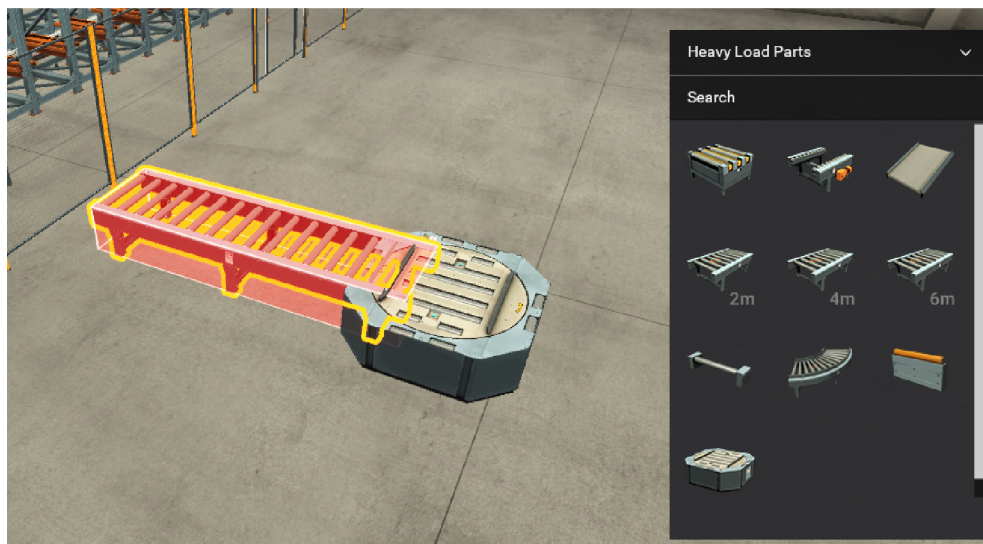
Pre vytvorenie scény od začiatku sú taktiež dve možnosti. Prvá je pomocou Uvítacieho menu na Paneli s nástrojmi a zvolením možnosti nový. Druhá cez tlačidlo Súbor na Paneli s nástrojmi a zvolením nový.

1.4 Úprava scény

Pri úprave scény je možné vykonávať nasledujúce úkony:

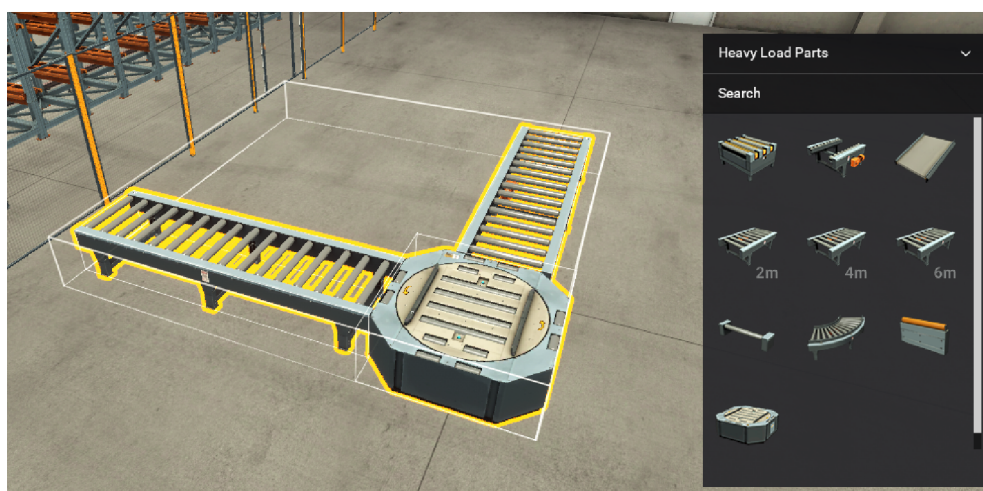
- **Pridávanie** dielov sa vykonáva pretiahnutím z paletového okna do 3D prostredia scény. Je možné pridať novú časť duplikovaním existujúcej a to tak,

že vyberieme diel, stlačíme a podržíme Alt a presunieme duplikovanú časť na nové miesto. Tento postup je možný buď na rovnakej scéne alebo aj medzi rôznymi scénami. Pri zlom umiestnení budú diely svietiť na červeno. Ak nedôjde k úprave, diely sa odstránia. Príklad takéhoto vloženia je na obrázku 1.11. [11]



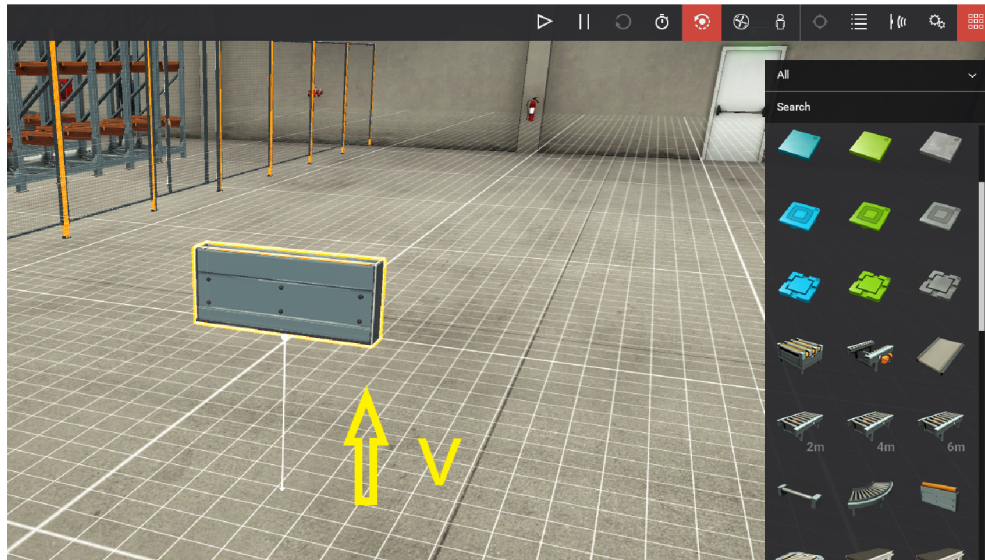
Obr. 1.11: Príklad zlého vloženia

- **Výber** jedného dielu sa vykonáva pomocou kliknutia ľavým tlačidlom myši. Pri výbere viacerých naraz sa používa kliknutie ľavým tlačidlom myši na pozadie scény a nakreslenie obdĺžnika pretínajúceho všetky diely, ktoré majú byť zahrnuté do výberu. Je možné aj následné pridanie podržaní klávesu Ctrl. Takýto výber je na obrázku 1.12. [11]



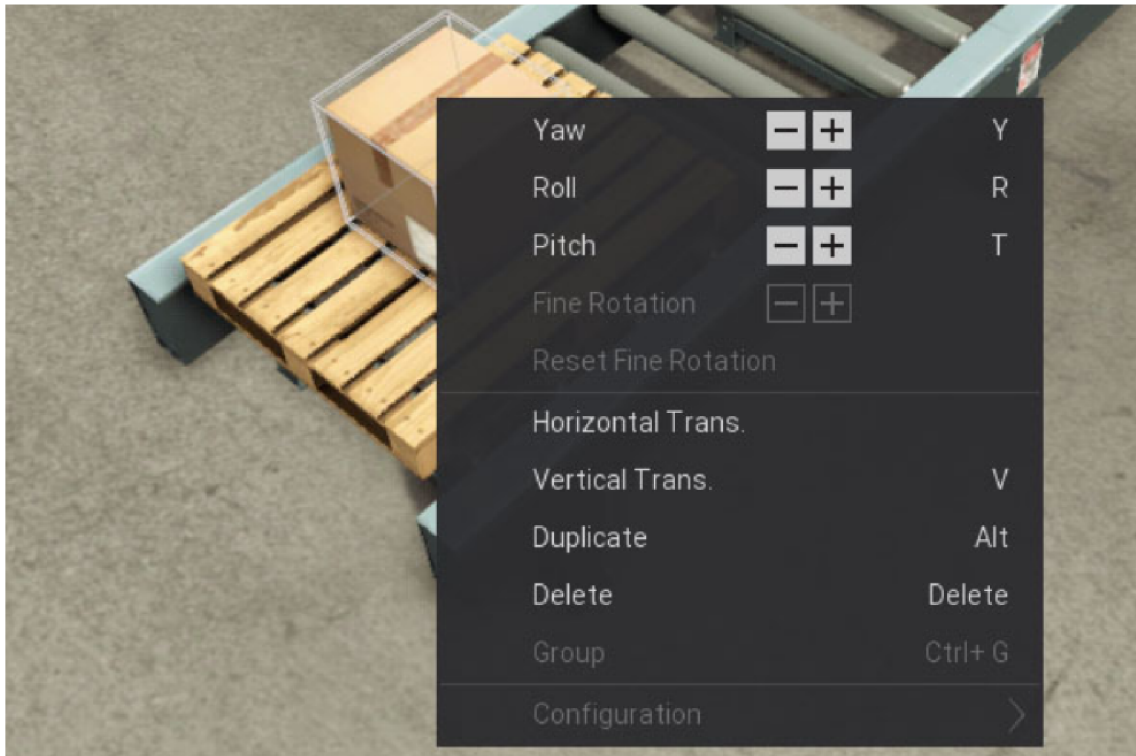
Obr. 1.12: Výber dielov na úpravu

- **Vymazanie** výberom a stlačením tlačidla Del.
- **Presúvanie** Na presúvanie dielov je potrebné kliknúť ľavým tlačidlom a držať ho. Takto sa hýbu diely horizontálne. Na vertikálny presun je potrebné stlačiť tlačidlo V, viď obrázok 1.13.



Obr. 1.13: Vertikálny presun dielov

- **Rotácia** dielov sa vykonáva stlačením Y na rotáciu okolo hornej osi, R na rolovanie a T na naklonenie. Väčšina dielov sa môže otočiť iba o 90°, ako napríklad dopravníky. [11]
- **Zoskupovanie** Diely môžu byť zoskupené, aby bolo možné jednoducho vykonávať viacero operácií naraz. Vyberieme diely, ktoré chceme zoskupiť, a stlačíme Ctrl+G, aby sme ich zoskupili. Jednotlivú časť skupiny je možné vybrať podržaním Ctrl a jej výberom. Ak chceme zrušiť zoskupenie všetkých častí, vyberieme skupinu a stlačíme Ctrl+G. [11]
- **Konfigurácia** Niektoré časti obsahujú rôzne konfigurácie, ktoré je možné vybrať z kontextového menu. Dostupné konfigurácie sa zvyčajne týkajú toho, ako diely fungujú alebo vizuálnej reprezentácie akou je napríklad farba. [11]
- **Kontextové menu** Väčšinu predchádzajúcich príkazov je možné vykonať prostredníctvom kontextového menu. Kliknutím pravého tlačidla myši na výber zobrazíme kontextové menu, ktoré je na obrázku 1.14. [11]



Obr. 1.14: Kontextové menu

[11]

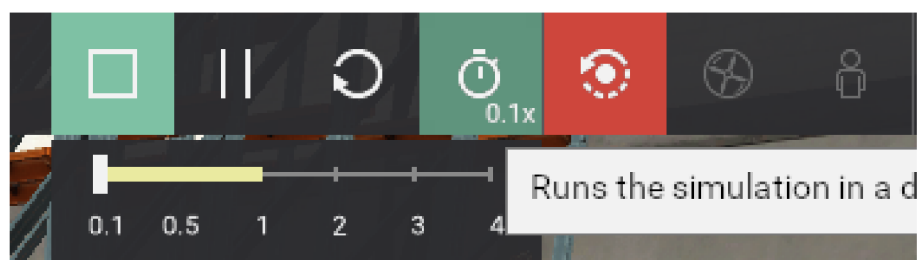
1.5 Spustenie simulácie

Na spustenie simulácie scény v reálnom čase slúži tlačidlo spustenia, ktoré bolo spomenuté v podkapitole Panel s nástrojmi. Simuláciu môžeme kedykoľvek zastaviť a znovu spustiť. Pozastavenie je výhodné, pretože je možné skontrolovať stav dielov a snímačov v simulácií, čo uľahčí prácu pri ladení.

Simuláciu scény je taktiež možné spustiť spomalene alebo zrýchlene. Pri spomalenom režime je možné jednoduchšie sledovať proces a správanie sa jednotlivých dielov a snímačov. Spomalený režim je výhodný najmä pri scénach, ktoré majú rýchlo sa pohybujúce časti. Pri zrýchlenej simulácii scény môžeme overiť logiku ovládača za kratší čas alebo je tiež výhodné využiť ho, keď sa proces opakuje za účelom zisku dát. Pri tomto režime je však potrebné dať si pozor na rýchlo sa pohybujúce diely, pretože môže dôjsť k tomu, že snímač ho nestihne detegovať. [12]

Pre zmenu rýchlosti behu simulácie je potrebné stlačiť tlačidlo vedľa orbitálnej kamery vľavo (viď. obrázok 1.15) a následne vybrať požadovanú rýchlosť.

V simulácii je možná aj interakcia s predmetmi. Interakcia však nie je možná so všetkými. Ak kurzorom myši ukážeme na predmet a ten sa zmení na ruku, môžeme ho kliknutím uchopiť a posunúť. Stlačením klávesu Shift obmedzíme pohyb vo



Obr. 1.15: Úprava rýchlosti simulácie

vodorovnej roviny a bude sa presúvať vertikálne.

1.6 Ovládanie simulácie

Ako už bolo spomenuté v úvode tejto kapitoly, simulácie scén je možné ovládať viacerými spôsobmi, ako napríklad mikrokontrolérmi alebo PLC. Predtým, ako tak urobíme, je výhodné otestovať scénu manuálne, aby všetko fungovalo podľa predstáv.

Na manuálne overenie scény sa využívajú znalosti označení(tagov) dielov simulácie. Každý diel má aspoň jedno označenie pozostávajúce z názvu a hodnoty. Je výhodné označenia pomenovať krátko a výstižne. Toto pomenovanie je dôležité preto, lebo sa bude používať pri mapovaní aktuátorov a snímačov na zariadenie, ktorým bude ovládaná linka. Označenia môžu obsahovať tri rôzne typy údajov [13]:

Tab. 1.1: Možné dátové typy [13]

Dátový typ	Veľkosť v bytoch	IEC 61131-3 ekvivalent
Boolean	1	BOOL
Float	4	REAL
Integer	4	DINT

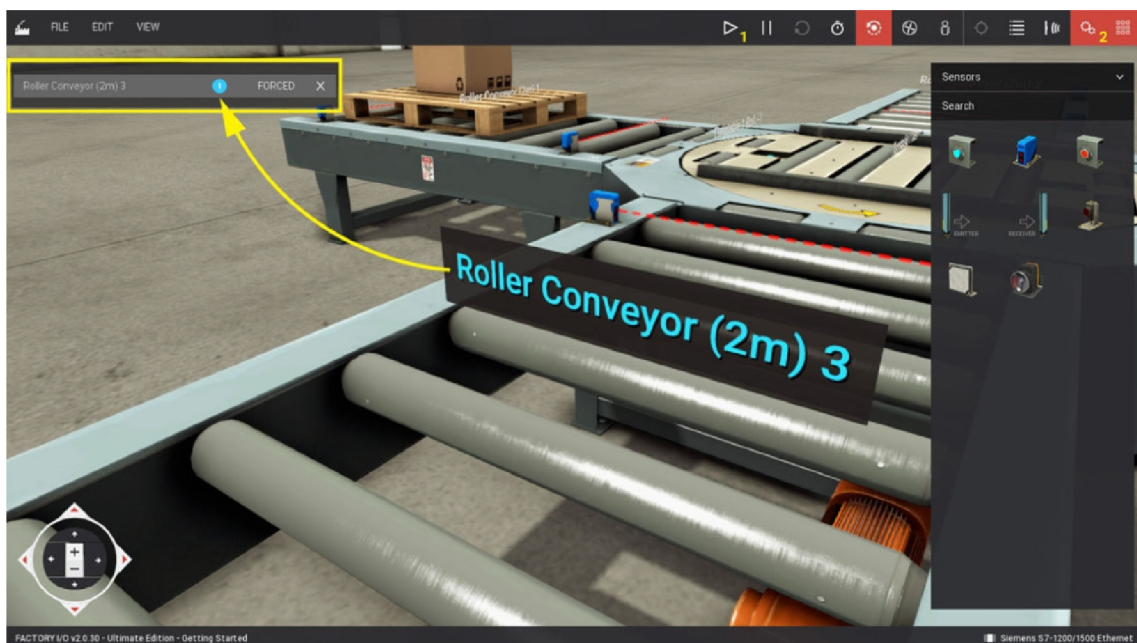
Označenia sensorov a akčných členov môžeme zobrazit alebo skryt kliknutím na príslušné ikony na Paneli s nástrojmi, ako je možné vidieť na obrázku 1.16. Kliknutím na ľavé tlačidlo sa zobrazia, respektíve skryjú označenia sensorov a kliknutím na pravé tlačidlo sa zase skryjú/zobrazia označenia akčných členov.



Obr. 1.16: Zobrazenie označení

[13]

Po kliknutí na názov objektu, ktorý chceme ovládať, sa nám zobrazí jeho označenie vľavo hore a môžeme zmeniť jeho hodnotu. Označeniam senzorov a akčných členov možno vynútiť hodnotu kliknutím myši na tlačidlo označenia, posuvníka alebo vstupného poľa (čo sa líši od typu dát, ktoré obsahuje). Týmto postupom sa prepíše hodnota načítaná z ovládača a snímač deteguje položku aj keď tam nie je. Touto zmenou je možné ovládať scénu manuálne a tým suplovať úlohu ovládača. Taktiež je možné vyvolať chybu a testovať chybové stavy. Je možná aj zmena názvu označenia a aj vymazanie všetkých ukotvených označení a to tak, že klikneme na Zobrazit > Vymazať ukotvené štítky. [13]



Obr. 1.17: Ovládanie simulácie [14]

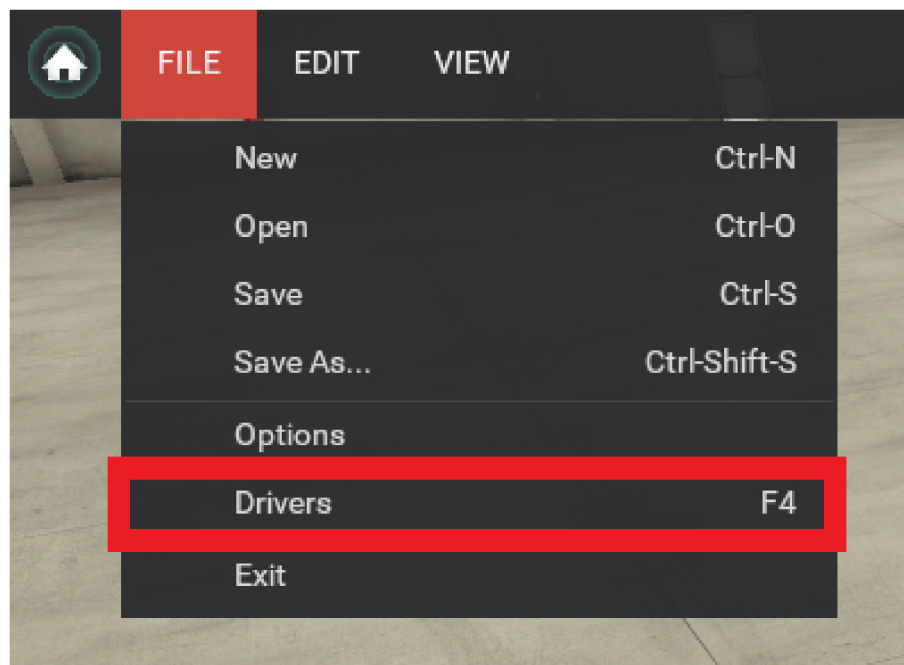
Každá scéna obsahuje štyri vstavané označenia (senzory), ktoré môže ovládač použiť na získanie relevantných údajov o simulácii a to:

- Factory I/O (Paused), ktorá je v logickej jednotke, ak je simulácia pozastavená
- Factory I/O (Reset), ktorá je v logickej jednotke(počas jednej sekundy), ak je simulácia resetovaná
- Factory I/O (Running), ktorá je v logickej jednotke, ak je simulácia spustená
- Factory I/O (Time Scale), ktorá zobrazuje, aký je nastavený čas simulácie (či je zrýchlená alebo spomalená)

Existujú aj označenia(aktuátory), ktoré ovládajú simuláciu. Majú tie isté názvy ako označenia(senzory). Rozdiel je v tom, že neslúžia na čítanie aktuálneho stavu, ale na zapísanie. Ak nastavíme označenie Factory I/O (Paused), simulácia sa zastaví. [13]

1.7 Ovládanie scény pomocou PLC

Na ovládanie scény pomocou PLC sa využíva znalosť I/O ovládačov(driverov). Je to vstavaná funkcia Factory I/O, ktorá sa používa na komunikáciu s externým riadiacim zariadením. Vo Factory I/O je veľa I/O ovládačov, ako napríklad od firmy Allen-Bradley alebo Siemens. Po výbere konkrétneho ovládača, ktorý chceme použiť, je potrebné ho nakonfigurovať. Je to nevyhnutné pre správnu komunikáciu. Pre zobrazenie dostupných ovládačov klikneme na Súbor > Ovládače alebo pomocou klávesy (F4). Obidve možnosti môžeme vidieť na obrázku 1.18. [15]



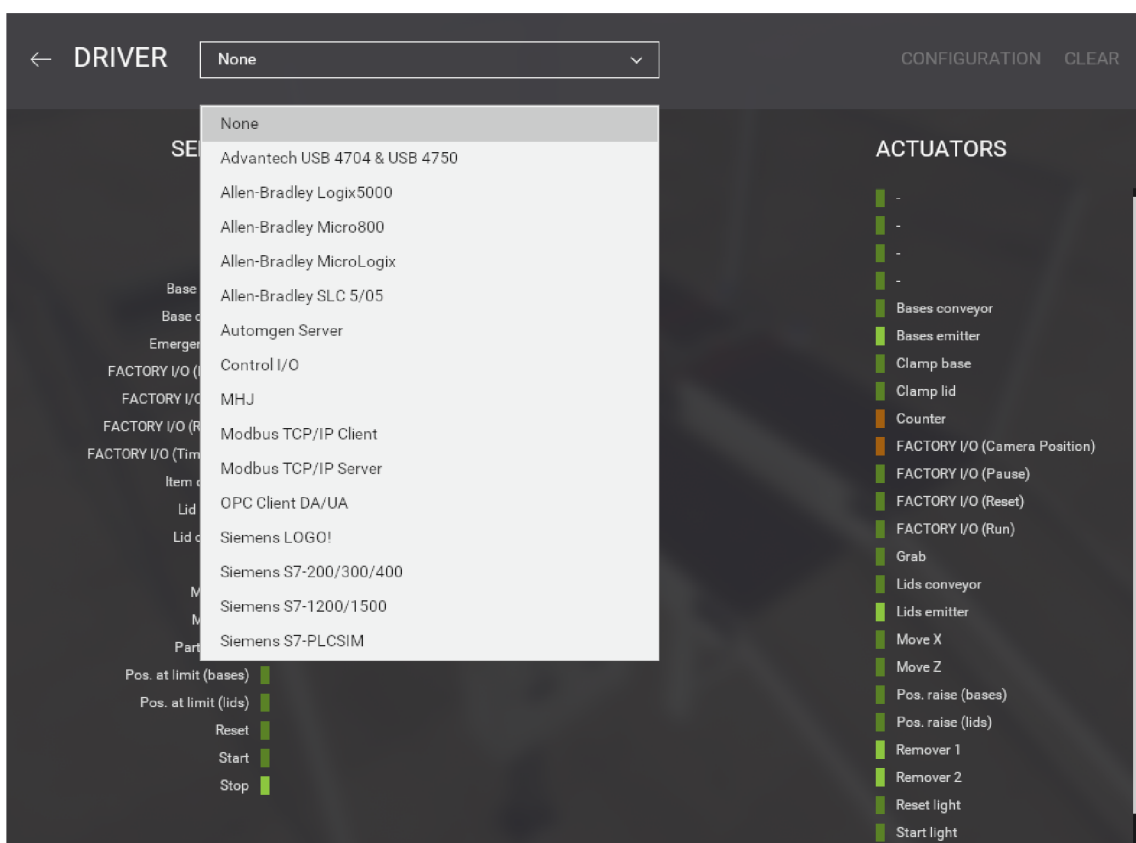
Obr. 1.18: Nastavenie ovládačov

Prípadne kliknutím ľavého tlačidla myši na štítok ovládača zobrazený v stavovom riadku, viď orázok 1.19.



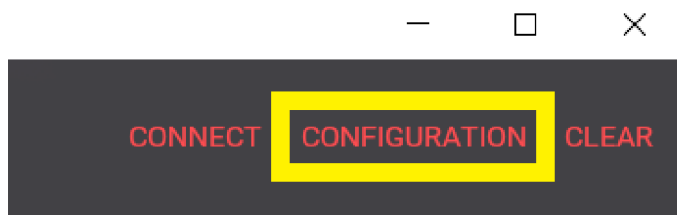
Obr. 1.19: Nastavenie ovládačov pomocou stavového riadku

V ponuke ovládača(driver) je možné vyberať ovládač zo zoznamu. V zozname (viď obrázok 1.20) sú viaceré druhy ovládačov. Ovládače od firmy Allen-bredley a Siemens(okrem PLCSIM) využívajú pripojenie pomocou Ethernetu. Automgen využíva komunikáciu cez TCP/IP server. MHJ je rozhranie pre softvér WinPLC-Engine a WinSPS-S7. Control I/O je rozhranie priamo navrhnuté na komunikáciu Factory I/O a SoftPLC. [15]



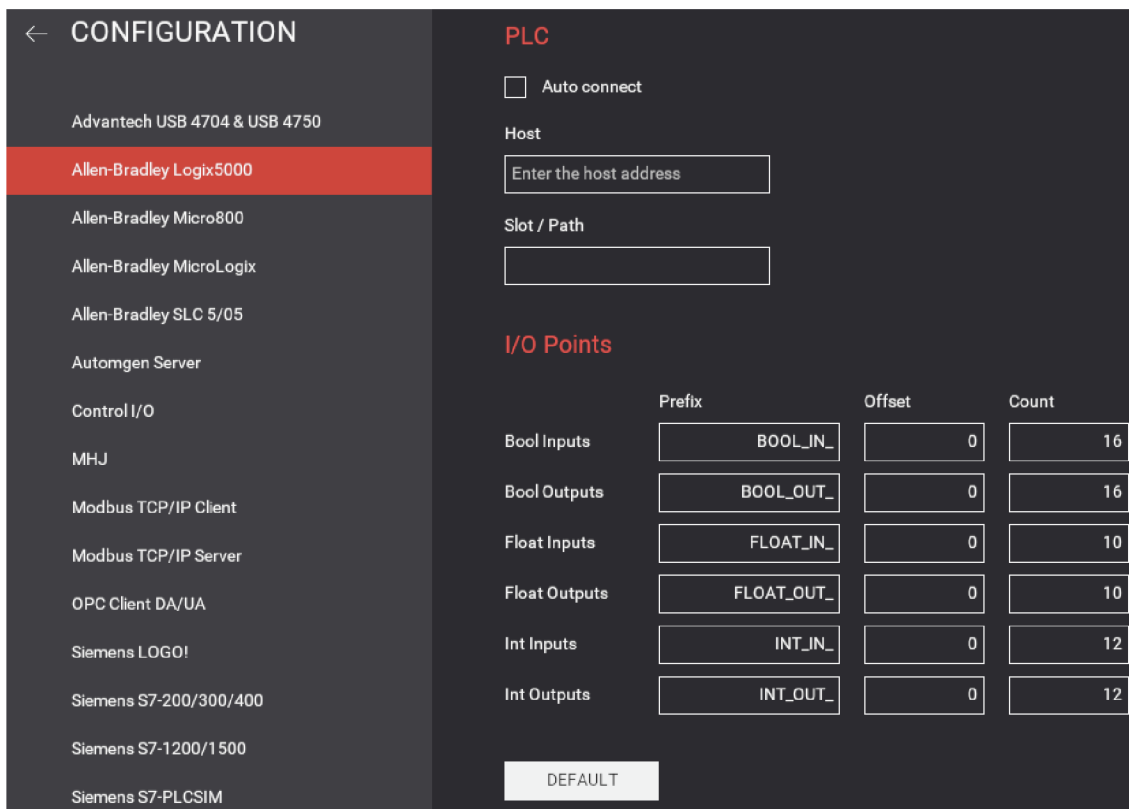
Obr. 1.20: Dostupné ovládače

Po výbere Ovládača, ktorý budeme využívať na ovládacie scény, nasleduje konfigurácia. K nej je možné pristúpiť pomocou tlačidla v hornom rohu, viď obrázok 1.21.



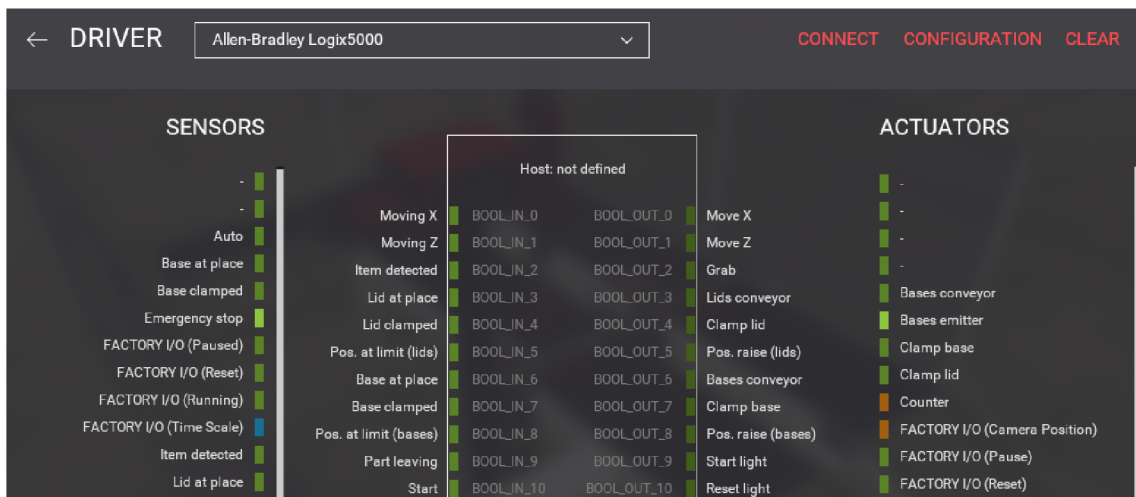
Obr. 1.21: Konfigurácia

Kliknutím myši na položku Konfigurácia, otvoríme konfiguračný panel, ktorý je na obrázku 1.22. V tomto paneli sa nastavuje komunikácia s ovládacím systémom. Taktiež sa tu nastavuje počet a typ premenných, s ktorými budeme riadiť scénu.



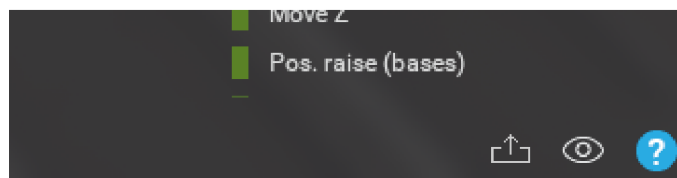
Obr. 1.22: Konfiguračný panel

Keď je ovládač správne nakonfigurovaný, môžeme priradiť označenia (tagy) snímačov a ovládačov, ktoré chceme ovládať (viď. obrázok 1.23). Komunikáciu s PLC začneme kliknutím myši na tlačidlo spojenie (CONNECT).



Obr. 1.23: Konfiguračný panel

V tejto karte je aj možnosť exportu mapovaných označení a vstupno-výstupných bodov ovládača scény kliknutím na obrázok tlačidla vpravo dole. Údaje sa exportujú ako súbor csv pre všetky podporované ovládače okrem Siemensu, ktoré sa exportujú ako xml kompatibilný s TIA Portal. Táto funkcia je implementovaná v ovládačoch Allen-Bradley, Automgen, MHJ, Modbus a Siemens. Tlačidlo exportu sa nachádza v pravom dolnom horu, viď obrázok 1.24 [15]



Obr. 1.24: Export konfigurácie

1.8 Užitočné skratky

Pri práci s FACTORY I/O je výhodné používať klávesové skratky. Uľahčujú a urýchľujú prácu s FACTORY I/O. Zoznam najpoužívanejších je v nasledujúcej tabulke.

Tab. 1.2: Skratky

Klávesová skratka	Popis funkčnosti
CTRL+N	Vytvorí novú scénu
CTRL+O	Otvorí okno Otvoriť scénu
CTRL+S	Uloží scénu
CTRL+SHIFT+S	Otvorí okno Uložiť scénu
CTRL+Z	Vráti akciu späť
CTRL+Y	Vrátenú akciu urobí znovu
CTRL+G	Zoskupí prvky
Page UP/Page DOWN	Prepínanie medzi uloženými pohľadmi
Ľavé tlačidlo	Posuv vľavo
Pravé tlačidlo	Posuv vpravo
F4	Otvorí okno Konfigurácia ovládača
F12	Urobí snímku obrazovky

2 Návrh úlohy

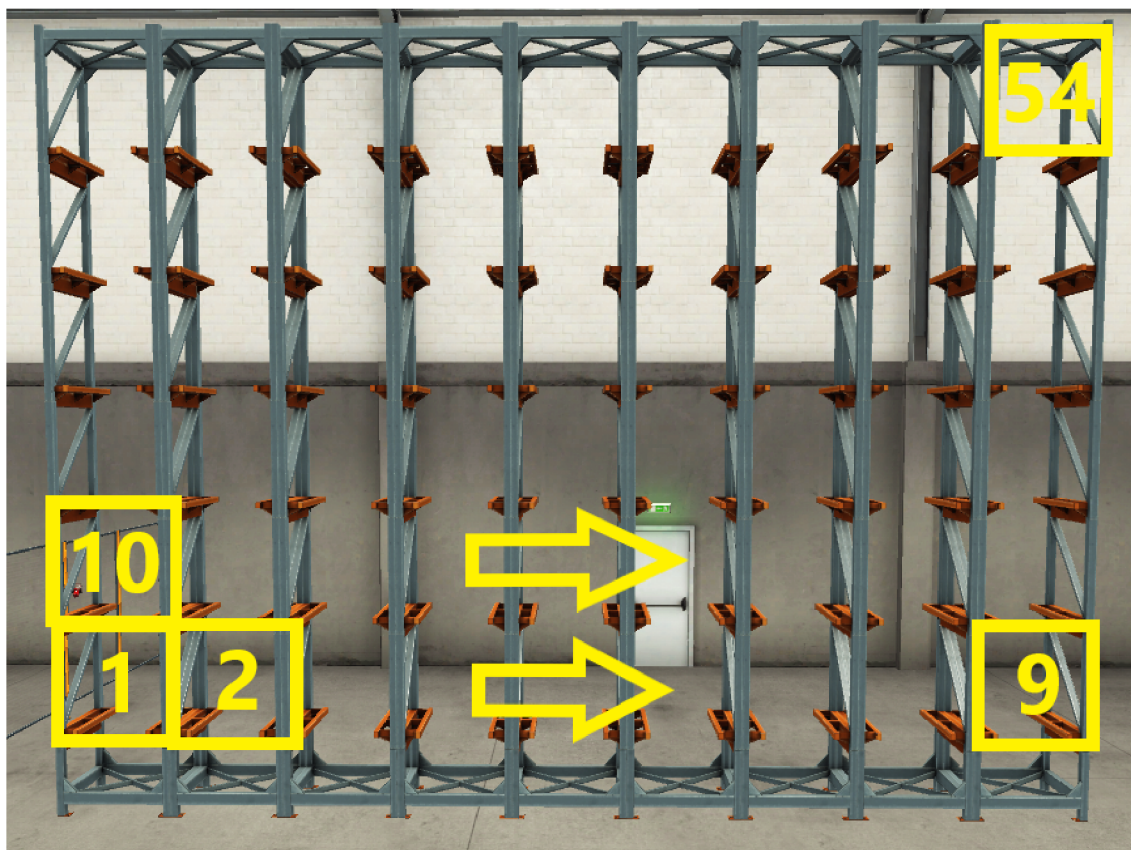
Ako som spomínal v úvode predchádzajúcej kapitoly, vo Factory I/O sa dajú scény vytvárať dvoma spôsobmi. Rozhodol som sa vytvoriť scénu úplne od začiatku a pridať čo najviac zaujímavých prvkov, no stále nevyrobiť niečo príliš komplikované. Vytvoril som návrh automatického skladu s prvkami výroby. Tento návrh je na obrázku 2.1. Sú v ňom dva sklady s regálmi vpravo a vľavo. Nachádzajú sa tam aj dva zakladače palet, ktoré vedú uložiť paletu aj vpravo aj vľavo. Taktiež je tam zakladač balíkov, ktorý ukladá balíky na prázdnu paletu. Sú tam aj dopravníky a dokonca aj bezpečnostné prvky. Linku som navrhol tak, aby sa do skladu dalo zakladať a aj z neho vykladať. Výrobnú linku je navrhnutá tak, aby bola čo najefektívnejšia a teda použiteľná pre prax. Zakladače do skladu a zakladač balíkov fungujú súbežne, teda nedochádza k zbytočnému čakaniu. Taktiež vykladanie z prvého a druhého skladu funguje súbežne.



Obr. 2.1: Návrh linky

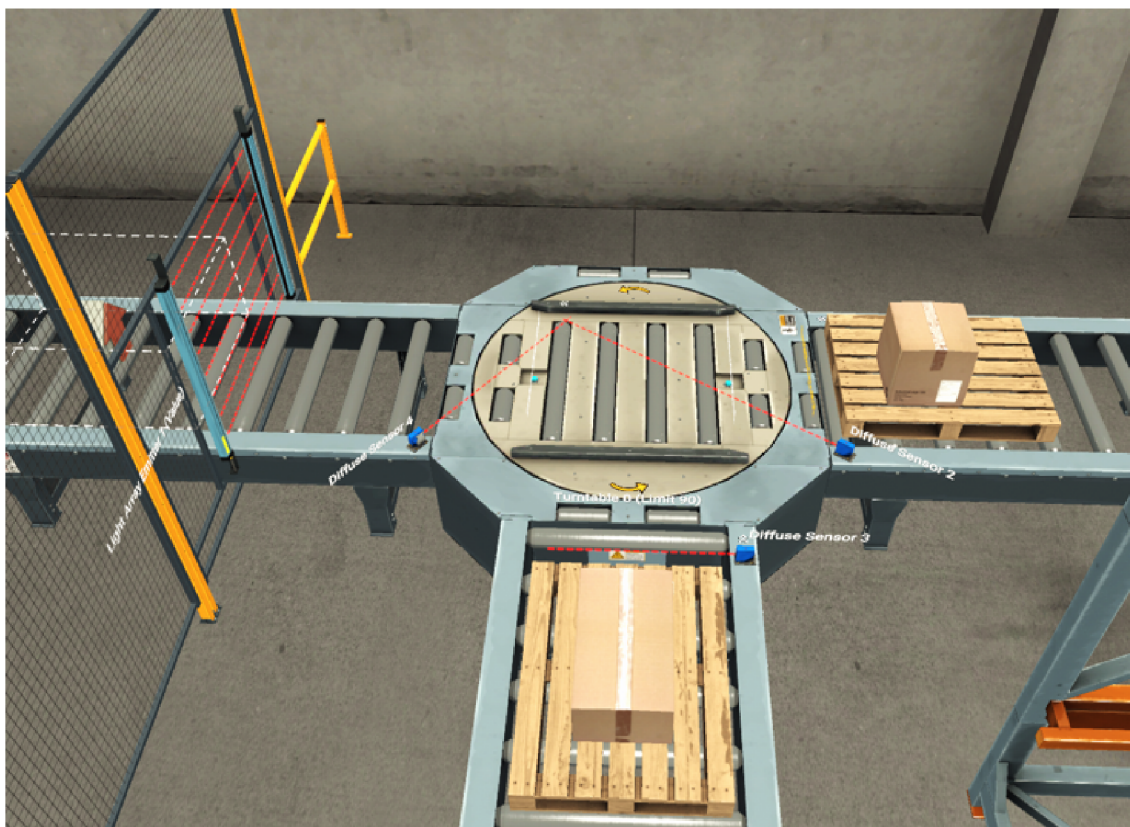
2.1 Zakladače a sklady

Pri zakladačoch sa primárne zakladá do regálov. Palety sa začnú ukladať do pravej časti skladu na pozíciu 1. Pozícia 1 sa, pri pohľade sprava, nachádza v ľavom dolnom rohu. Najskôr sa zaplní prvý riadok skladu, následne sa pokračuje na riadok vyššie, viď obrázok 2.2 nižšie. Po zaplnení všetkých pozícií v pravej časti (pozícia 54) sa pozícia nastaví zase na pozíciu 1 a palety sa začnú ukladať do ľavej časti skladu takým istým spôsobom.



Obr. 2.2: Ukážka pozícií skladu

Ak príde požiadavka na vyloženie zo skladu a zakladač dokončil svoju činnosť (uložil paletu do skladu), môže dôjsť k vyloženiu. Pri vyložení zo skladu som určil priority vyloženia prvého skladu, na obrázku 2.1 vpravo. Ako som už spomenul, urobil som tak z dôvodu predpokladu častejšieho vykladania z prvého skladu. Keď však dôjde ku vyloženiu palety aj z druhého skladu, paleta sa na križovatke zastaví a počká na uvoľnenie otočného dopravníka. Po uvoľnení sa vyloženie z prvého skladu zastaví a vyloží paleta z druhého skladu. Na zistenie, či je otočný dopravník obsadený a paleta z druhého skladu je pri ňom, som použil tri optické snímače. Pre bližšie zobrazenie otočného dopravníka a riešenie priority, viď obrázok 2.2.



Obr. 2.3: Spojenie vykladania

2.2 Zakladač balíkov

Zakladač balíkov som navrhol tak, že sa v prvom rade balíky najskôr založia vertikálne a v druhom rade sa otočia o 90 stupňov a ukladajú sa horizontálne. Zakladač balíkov je koncipovaný týmto spôsobom preto, aby balíky pri preprave lepšie držali pokope. Na nasledujúcom obrázku 2.3 môžete detailnejšie vidieť spomínaný návrh. Po uložení druhej vrstvy sa paleta pošle do druhého skladu a ukladajú sa nové balíky na novú paletu. Súčasťou paletizačného stroja je aj ovládací panel, ktorý je potrebný na jeho ovládanie. V mojom návrhu som použil tri tlačidlá, a to konkrétne tlačidlo Štart, Stop a E-Stop. Tlačidlá Štart a Stop sú využívané na spustenie a zastavenie paletizačného stroja, nie však celej výrobnéj linky. Nachádzajú sa aj vo vizualizácii. Tlačidlo E-Stop bude detailnejšie opísané v nasledujúcej podkapitole.



Obr. 2.4: Balíky uložené na paletu

2.3 Bezpečnosť

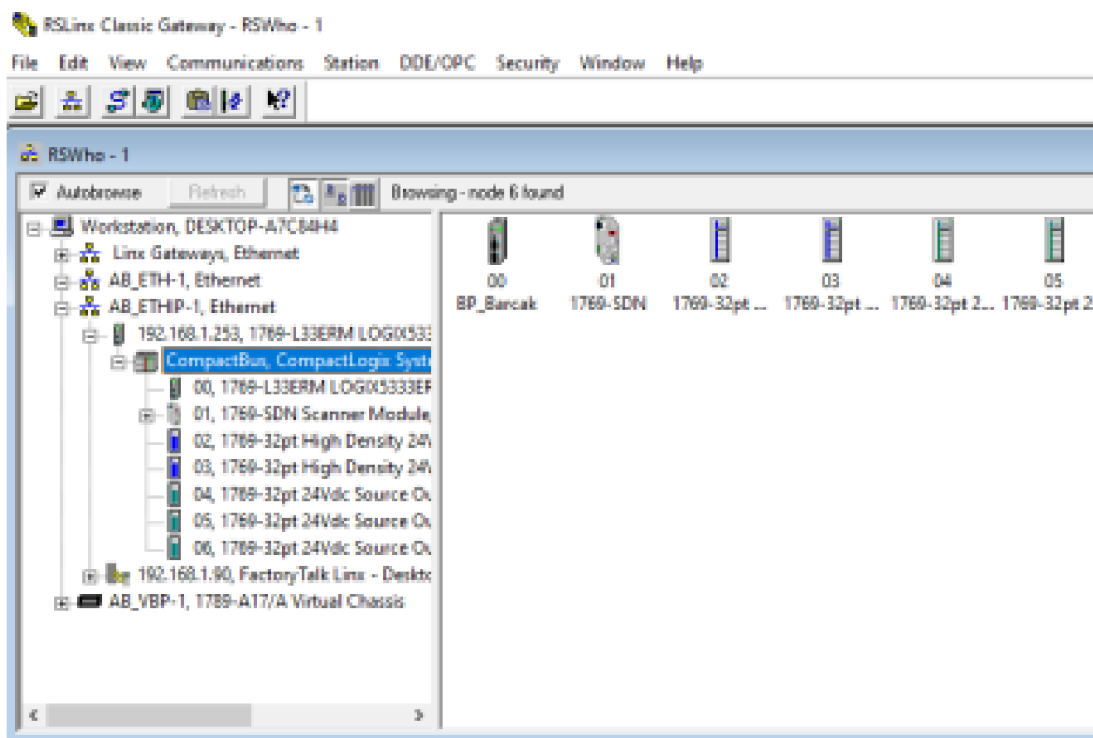
Vo výrobnjej linke som navrhol aj prvky základnej bezpečnosti. Výrobná linka je zabezpečená kliečkou s troma vstupnými dverami, ktoré umožňujú vstup do vnútra linky z každej strany. Ak dôjde k vstupu osoby do priestoru linky počas činnosti ktorejkoľvek pohyblivej časti linky, linka sa zastaví a následne je potrebné opustiť priestor, zavrieť dvere a potvrdiť bezpečnosť. Súčasťou paletizačného stroja je aj E-Stop, pretože pri návrhu som počítal s jedným operátorom, ktorý bude dozeráť na chod celej linky. Operátor po stlačení E-Stopu zastaví paletizačný stroj a predíde tak rizikovej udalosti. Na obrázku 2.5 môžete vidieť bezpečnostné prvky, ktoré navrhujem použiť.



Obr. 2.5: Bezpečnostné prvky

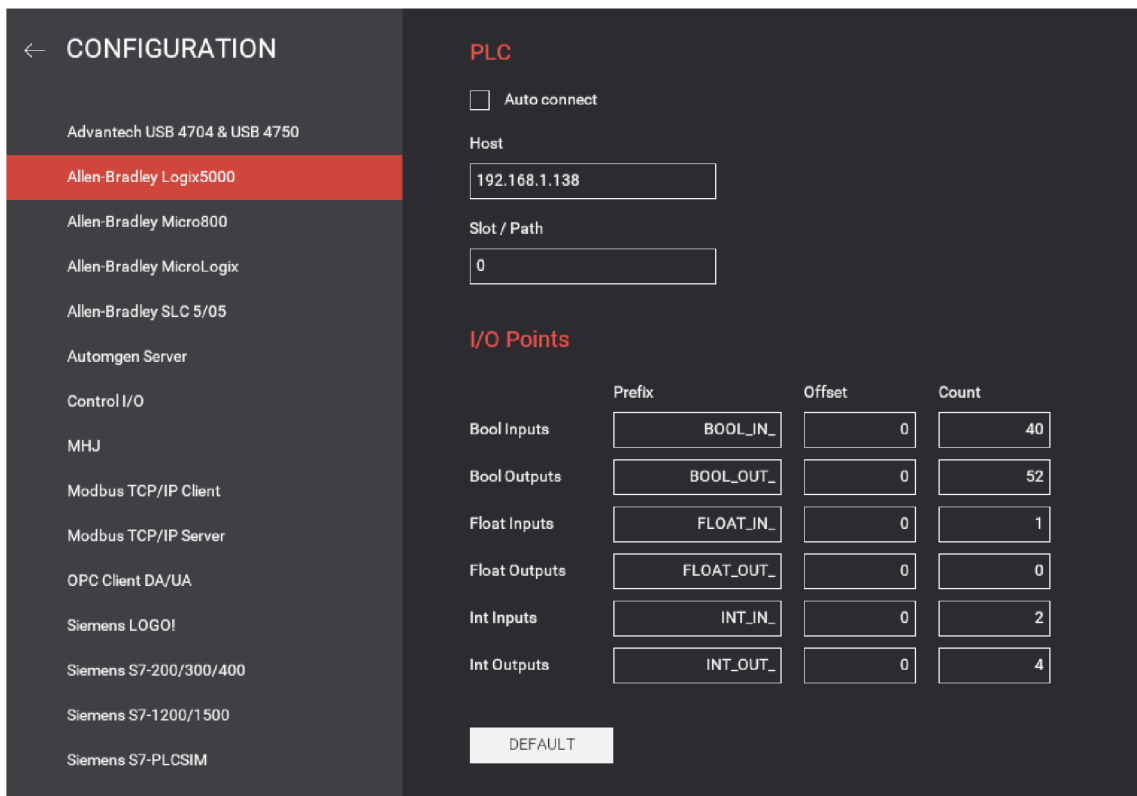
2.4 Nakonfigurovanie komunikácie

Pri nastavovaní komunikácie medzi Factory I/O a PLC existuje viacero postupov. Líšia sa v závislosti od vybraného riadiaceho modulu. V mojom prípade mi bolo poskytnuté PLC od firmy Rockwell Automation, a to konkrétne Compactlogix 1769-L33ERM. Pri tomto type PLC je nutné ako prvé spustiť aplikáciu RSLinx Classic a vyhľadať PLC, ktorým bude linka riadená. Na obrázku 2.6 je program RSLinx, v ktorom som vyhľadal PLC. V aplikácii RSLinx je možné vidieť všetky dostupné PLC, ich moduly a všetky potrebné informácie ako napríklad verzie firmware a podobne.



Obr. 2.6: RSLinx Classic

Po nájdení PLC (ktorým budeme ovládať linku) je potrebné zistiť jeho IP adresu a Slot. Tieto údaje je potrebné zadať v aplikácii Factory I/O a postupovať podľa podkapitoly 1.7. Kliknutím na Súbor > Ovládače (F4), vybrať možnosť Allein-Bradley Logix5000. V konfigurácii je nutné okrem IP adresy PLC a Slotu napísať aj počet potrebných premenných potrebných na ovládanie linky. Ukážku môžete vidieť na obrázku 2.7. Počet premenných odpovedá počtu použitých senzorov a aktuátorov, ktorými je simulácia ovládaná. Po pridaní senzoru alebo aktuátora je potrebné tento počet zmeniť.



Obr. 2.7: Konfigurácia

Následne je potrebné otvoriť vývojové prostredie pre PLC, čo je v mojom prípade Studio 5000. V programe som si pridal do controller tagov všetky potrebné premenné na ovládanie výrobnnej linky. Využíval som dva druhy vstupných a výstupných premenných, a to konkrétne typu BOOL a INT. V mojom prípade som musel pridať premenné s tým istým názvom ako v Factory I/O. Pridal som teda 40 premenných BOOL_IN a 2 INT_IN s indexami od 0 po n-1. Rovnako som postupoval aj pri výstupných premenných vid' obrázok 2.8. Po pridaní premenných v tomto tvare som pridal aj ich aliasy pod názvom, ktorý súvisel s ich činnosťou. Tento postup som zvolil preto, aby programovanie prebiehalo jednoduchšie.

Tab. 2.1: Premenné typu BOOL

Vstupné	Výstupné	Vstupné	Výstupné
BOOL_IN_0	BOOL_OUT_0	BOOL_IN_27	BOOL_OUT_27
BOOL_IN_1	BOOL_OUT_1	BOOL_IN_28	BOOL_OUT_28
BOOL_IN_2	BOOL_OUT_2	BOOL_IN_29	BOOL_OUT_29
BOOL_IN_3	BOOL_OUT_3	BOOL_IN_30	BOOL_OUT_30
BOOL_IN_4	BOOL_OUT_4	BOOL_IN_31	BOOL_OUT_31
BOOL_IN_5	BOOL_OUT_5	BOOL_IN_32	BOOL_OUT_32
BOOL_IN_6	BOOL_OUT_6	BOOL_IN_33	BOOL_OUT_33
BOOL_IN_7	BOOL_OUT_7	BOOL_IN_34	BOOL_OUT_34
BOOL_IN_8	BOOL_OUT_8	BOOL_IN_35	BOOL_OUT_35
BOOL_IN_9	BOOL_OUT_9	BOOL_IN_36	BOOL_OUT_36
BOOL_IN_10	BOOL_OUT_10	BOOL_IN_37	BOOL_OUT_37
BOOL_IN_11	BOOL_OUT_11	BOOL_IN_38	BOOL_OUT_38
BOOL_IN_12	BOOL_OUT_12	BOOL_IN_39	BOOL_OUT_39
BOOL_IN_13	BOOL_OUT_13		BOOL_OUT_40
BOOL_IN_14	BOOL_OUT_14		BOOL_OUT_41
BOOL_IN_15	BOOL_OUT_15		BOOL_OUT_42
BOOL_IN_16	BOOL_OUT_16		BOOL_OUT_43
BOOL_IN_17	BOOL_OUT_17		BOOL_OUT_44
BOOL_IN_18	BOOL_OUT_18		BOOL_OUT_45
BOOL_IN_19	BOOL_OUT_19		BOOL_OUT_46
BOOL_IN_20	BOOL_OUT_20		BOOL_OUT_47
BOOL_IN_21	BOOL_OUT_21		BOOL_OUT_48
BOOL_IN_22	BOOL_OUT_22		BOOL_OUT_49
BOOL_IN_23	BOOL_OUT_23		BOOL_OUT_50
BOOL_IN_24	BOOL_OUT_24		BOOL_OUT_51
BOOL_IN_25	BOOL_OUT_25		BOOL_OUT_52
BOOL_IN_26	BOOL_OUT_26		

Tab. 2.2: Premenné typu INT

Vstupné	Výstupné
INT_IN_0	INT_OUT_0
INT_IN_1	INT_OUT_1
	INT_OUT_2
	INT_OUT_3

Posledným krokom je pridanie označení (tagov) na konkrétne pozície. Sensory sa pridávajú na ľavú časť kontroléru a aktuátoru na pravú časť kontroléru. Ukážku môžete vidieť na nasledujúcom obrázku 2.9. Po splnení týchto krokov je komunikácia nastavená a je možné prepojiť Factory I/O s PLC pomocou tlačidla Connect.



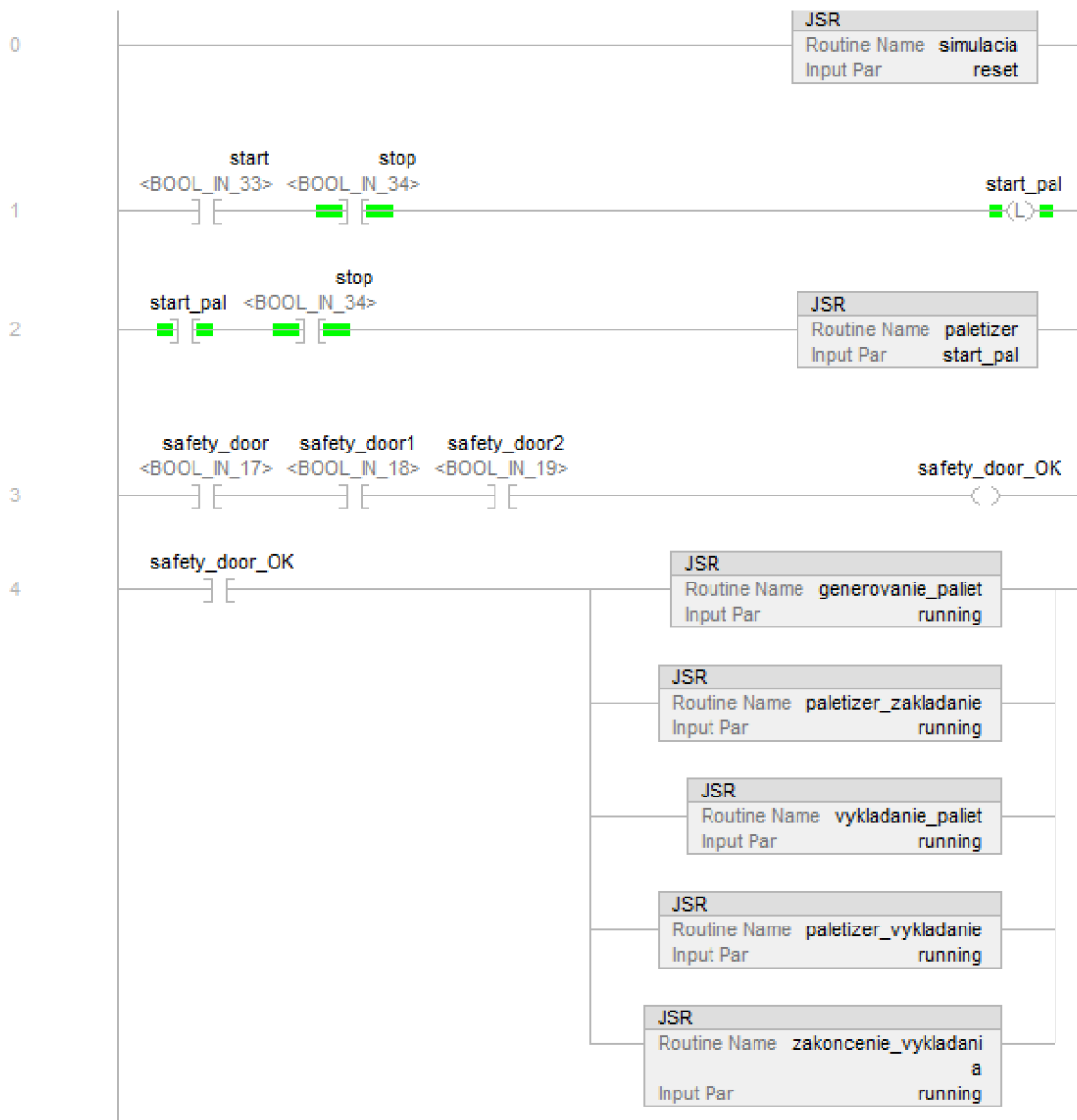
Obr. 2.8: Overenie komunikácie

3 Riadiaci softvér

V predchádzajúcej kapitole som popísal návrh virtuálnej výrobnéj linky a hardvér, ktorým bude ovládaná. V tejto kapitole podrobnejšie popíšem riadiaci softvér, ktorým linku ovládam. V studio 5000 je možné si vybrať z viacerých dostupných programovacích jazykov, a to konkrétne: Ladder Logic, Function Block Diagram, Structured Text a Sequential Function Chart. V mojej práci som sa rozhodol použiť kombináciu jazykov ST(Structured Text) a Ladder Logic. Pre dosiahnutie čo najprehľadnejšieho riadenia som sa rozhodol program rozdeliť do viacerých podprogramov. Pre každú časť linky je vytvorený vlastný podprogram. Vytvoril som aj jeden podprogram navyše, ktorý slúži na resetovanie premenných a riadenie simulácie. Všetky tieto podprogramy mi umožňujú plne automatizovať prevádzku linky a zabezpečiť jej bezproblémové fungovanie. V tejto kapitole sa budem zaoberať podrobnejšie prácou s týmito podprogramami a spôsobom, ako sú integrované do celkového softvéru pre ovládanie linky.

3.1 Hlavný program

Ako som už spomínal, riadiaci softvér je rozdelený do viacerých podprogramov, ktoré sú volané z hlavného programu. Tento program je ako jediný napísaný v jazyku Ladder a jeho ukážku môžete vidieť na obrázku 3.1 nižšie. Jeho hlavnou úlohou je to, aby volal podprogramy. Tieto podprogramy sú však volané iba vtedy, keď sú splnené podmienky, ako napr. bezpečnosť alebo požiadavka na spustenie. V prvom riadku je podprogram simulácia, pomocou ktorého je možný reset premenných a zobrazenie premenných pre vizualizáciu. Ako jediný je volaný vždy. V ďalších dvoch riadkoch som zabezpečil to, aby zakladanie balíkov bolo spustené, ak je požiadavka na štart (pomocou tlačidla na paneli) a nie je prerušený signál tlačidlom stop (taktiež na ovládacom paneli). Volanie je možné prerušiť, ak zadáme požiadavku na zastavenie. Ostatné podprogramy ovládajú linku, ktorá je vo vnútri bezpečnostnej kľetky. Do tejto kľetky sa dá vstúpiť iba cez tri bezpečnostné dvere a teda volanie týchto podprogramov je prerušené iba keď sú otvorené tieto dvere. Podrobnejšie tieto podprogramy rozoberiem v nasledujúcich podkapitolách.

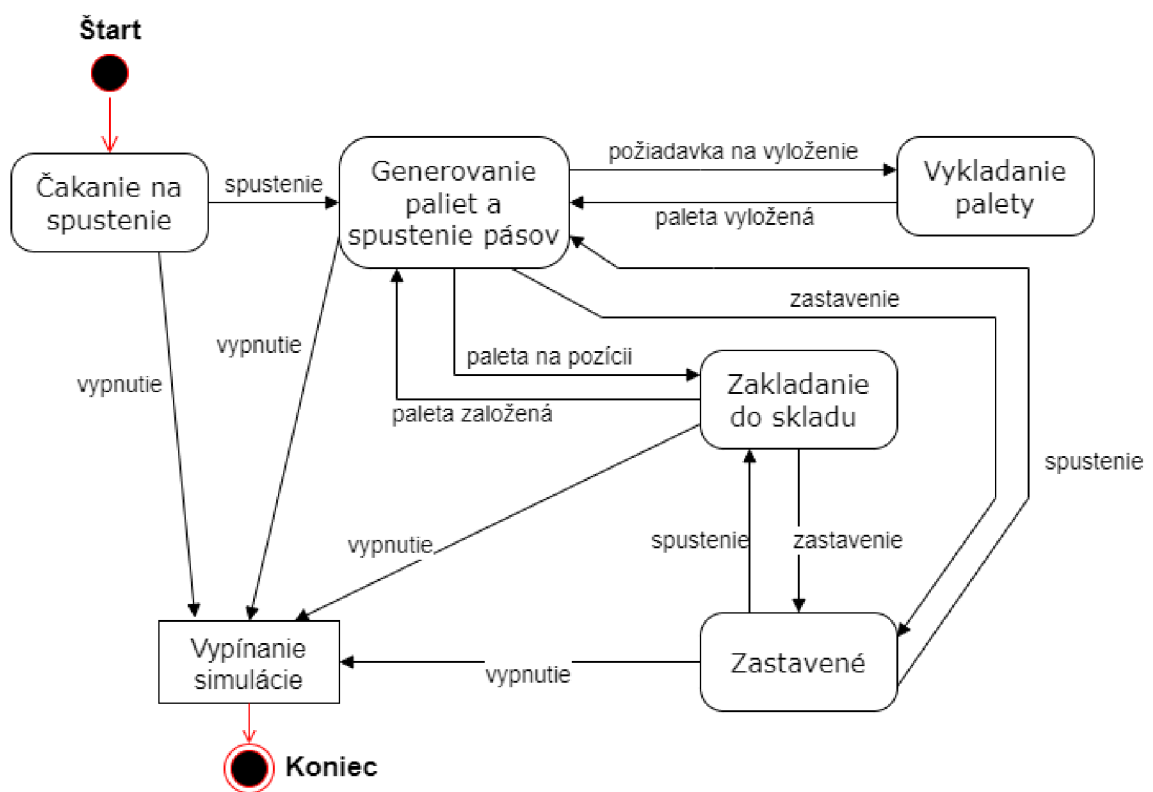


Obr. 3.1: Ukážka hlavného programu

3.2 Generovanie paliet

Tento podprogram je veľmi podobný s podprogramom "paletizer zakladanie", preto opíšem iba jeden. V tejto časti podprogramu sa generujú palety a zakladajú sa do pravého skladu. Po spustení simulácie musí prísť požiadavka na spustenie linky, následne sa automaticky spustí generovanie paliet a dopravníky, ktoré presunú tieto palety ku zakladaču paliet do skladu. Ak senzor zaznamená, že paleta je pri sklade, dopravníky sa zastavia a začne proces zakladania palety. Palety sú ukladané do skladu od prvej pozície po poslednú. Pre presnejší popis zakladania viď kapitolu 2.1. Po uložení do skladu a vrátení zakladača na počiatočnú pozíciu sa následne spustia dopravníky. Súbežne zakladač čaká na požiadavku založenia do skladu alebo

na požiadavku vyloženia prvej palety. Ak príde požiadavka založenia, paleta bude uložená do skladu. Ak však príde požiadavka vyloženia, podprogram prechádza do módu čakania. Následne je vykonávaný podprogram vykladania zo skladu, ktorý popíšem neskôr. Po dokončení vykladania podprogram generovania paliet prejde z módu čakania do normálneho režimu. Tento podprogram je však vykonávaný iba vtedy, ak nie je porušená bezpečnosť. Bezpečnosť je porušená vtedy, ak sú otvorené bezpečnostné dvere. Po ich otvorení sa zastavia všetky pohyblivé časti linky a preruší sa vykonávanie. Na opätovné spustenie je potrebné opustiť priestor, zatvoriť dvere a potvrdiť bezpečnosť linky tlačidlom, ktoré nie je možné potvrdiť z vnútra linky. Celý proces generovania a zakladania som názorne zobrazil v nasledujúcom stavovom diagrame na obrázku 3.2.



Obr. 3.2: Stavový diagram pre generovanie paliet

V prípade, že sa zaplnia všetky pozície v sklade, zakladač už nemá kde dávať palety. Ak však došlo k vyloženiu paliet na niektorých pozíciách, bolo by možné ich znovu naplniť. Z tohto dôvodu som doplnil kód o funkciu, ktorá je zobrazená na obrázku 3.3. Táto funkcia je volaná, ak sa "zapln" rovná jednotke. Funkcia prehľadáva celý sklad a hľadá pozície, ktoré sú prázdne. Ak takú pozíciu nájde, opätovne spustí proces zakladania a postupuje aj v ľavej časti skladu. Táto funkcia je mimo stavového automatu.

```

141 if pozicia0>108 then
142     x:=0;
143     y:=0;
144     pozicia0:=1;
145     zapln:=1;
146 end_if;
147
148 if zapln then
149     if sklad0P[x,y]=0 then
150         stav_gen:=1;
151     else
152         x:=x+1;
153         pozicia0:=pozicia0+1;
154     end_if;
155
156     if x>8 then
157         x:=0;
158         y:=y+1;
159     end_if;
160
161     if pozicia0>54 then
162         x:=0;
163         y:=0;
164         zapln_L:=1;
165     end_if;
166 end_if;
167
168 if zapln_L then
169     zapln:=0;
170
171     if sklad0L[x,y]=0 then
172         stav_gen:=1;
173     else
174         x:=x+1;
175         pozicia0:=pozicia0+1;
176     end_if;
177
178     if x>8 then
179         x:=0;
180         y:=y+1;
181     end_if;
182 end_if;

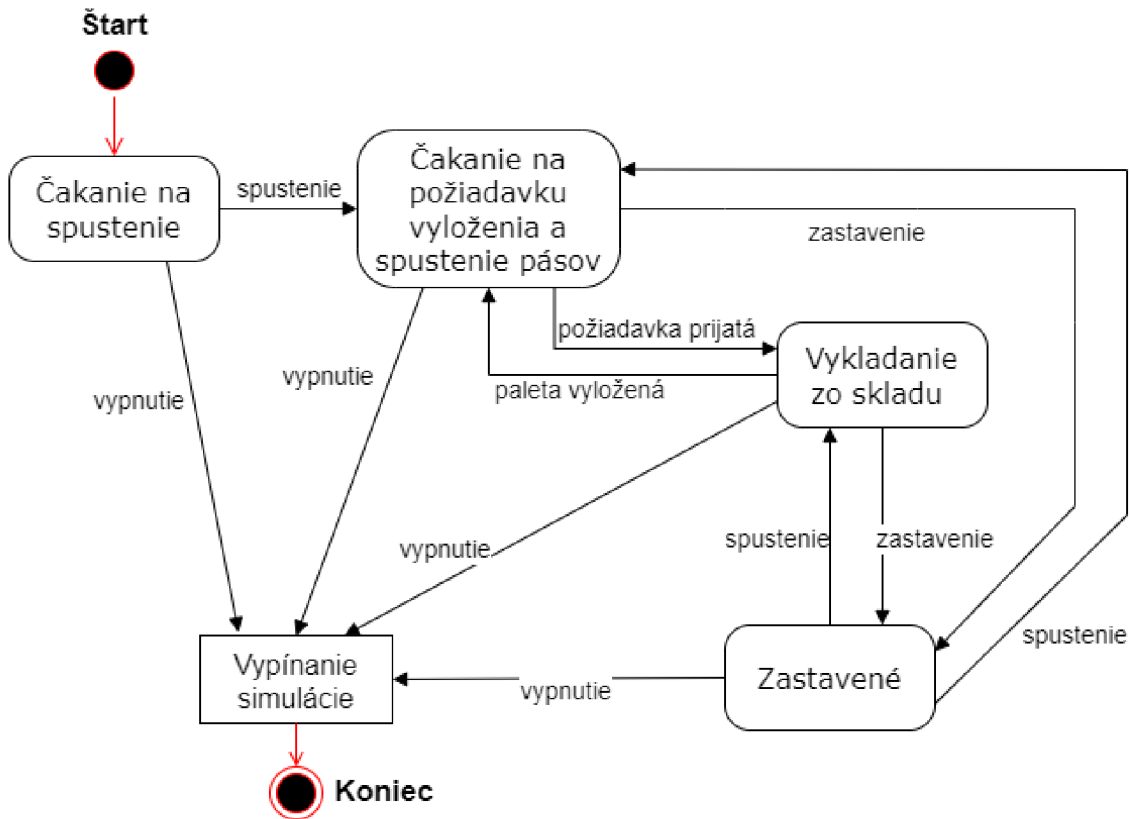
```

Obr. 3.3: Ukážka kódu zaplnenia

3.3 Vykladanie paliet

Vykladanie paliet a "paletizer vykladanie" sú podobné podprogramy, z tohto dôvodu opíšem iba jeden. Táto časť podprogramu slúži na vykladanie zo skladu. Pre vyloženie zo skladu je potrebné dať požiadavku s presnou polohou, ktorú chcete vyložiť. Požiadavka sa potvrdzuje tlačidlom enter. Pozíciu je potrebné zadať v dvojrozmernom priestore, kde súradnica osy x je vodorovná a súradnica osy y je zvislo. Ak však táto pozícia nie je zaplnená, vyloženie nebude možné a hodnoty sa vynulujú. Po zadaní požiadavky sa čaká, kým bude zakladač paliet voľný (nebude práve ukladať do skladu), následne sa paleta vyloží. Vykladať zo skladu je tiež možné iba vtedy, ak nie je porušená bezpečnosť. Tak ako pri ukladaní do skladu, je bezpečnosť porušená pri otvorení dverí. Pre opätovné spustenie je nutné opustiť priestor a potvrdiť opustenie

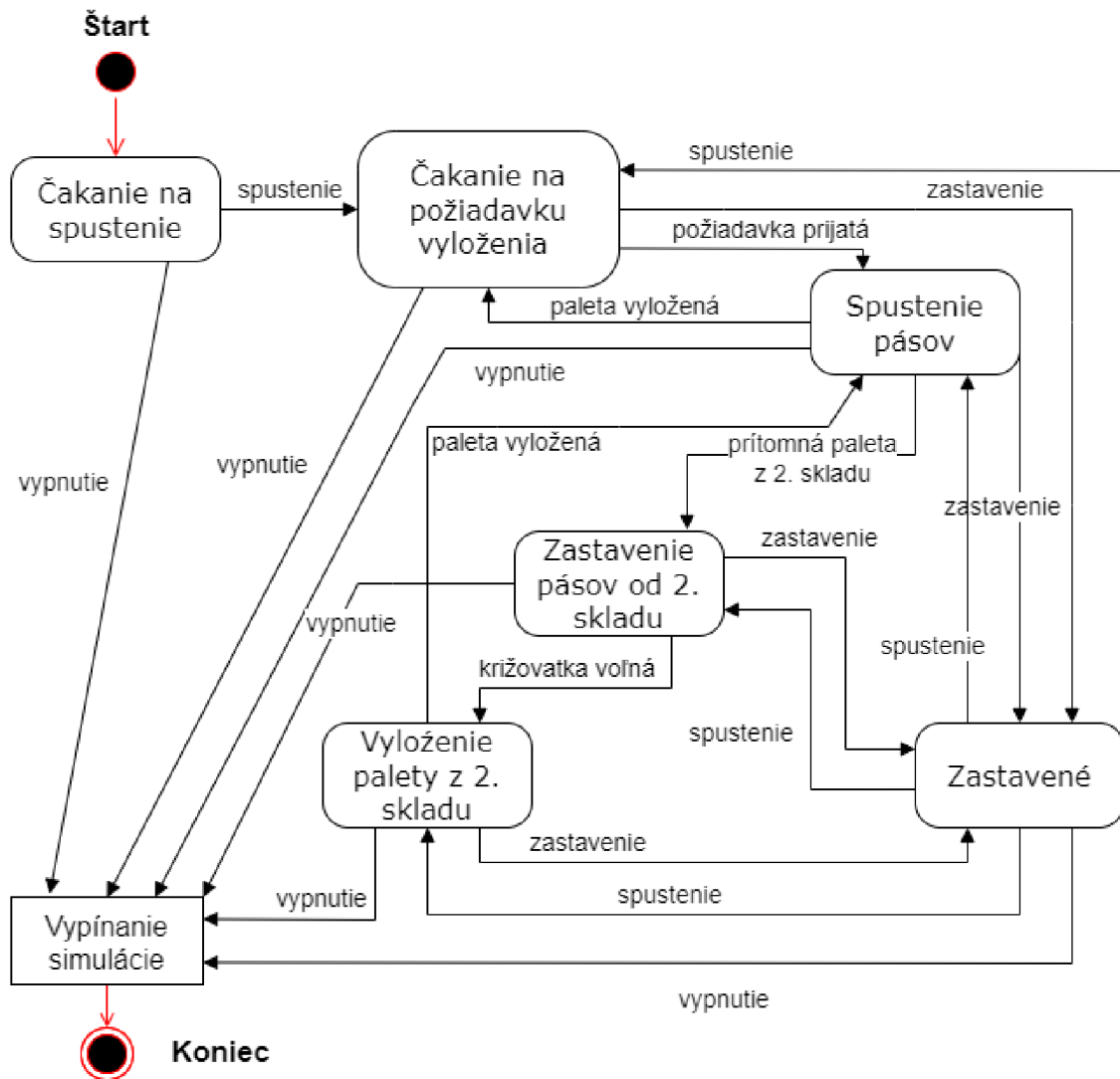
priestoru linky tlačidlom. Tento proces je zobrazený pomocou stavového diagramu na nasledujúcom obrázku 3.4.



Obr. 3.4: Stavový diagram pre vykladanie paliet

3.4 Zakončenie vykladania

Pri vytváraní návrhu virtuálnej výrobnéj linky bolo dôležité, aby bola čo najefektívnejšia. Teda, aby zbytočne nestála a požadovaná činnosť bola čo najrýchlejšia. Preto vykladanie z prvého skladu a vykladanie z druhého skladu funguje súbežne. Prioritne sa vykladá prvý sklad, pretože sa rýchlejšie zaplní. Z toho teoreticky vyplýva, že z neho bude vykladané častejšie. Keď dôjde k vyloženiu palety, spustia sa dopravníky. Následne ide paleta na koniec, kde bude vyložená. Ak však príde paleta z druhého skladu k otočnému dopravníku, dopravníky druhého skladu sa zastavia a čaká sa na uvoľnenie otočného dopravníka. Keď je dopravník voľný, otočí sa o 90 stupňov a paleta je presunutá na koniec dopravníka. Následne sa dopravník otočí naspäť, zapnú sa všetky dopravníky a pokračuje sa vo vykladaní. Tento proces som názorne zobrazil aj v stavovom diagrame na obrázku 3.5.

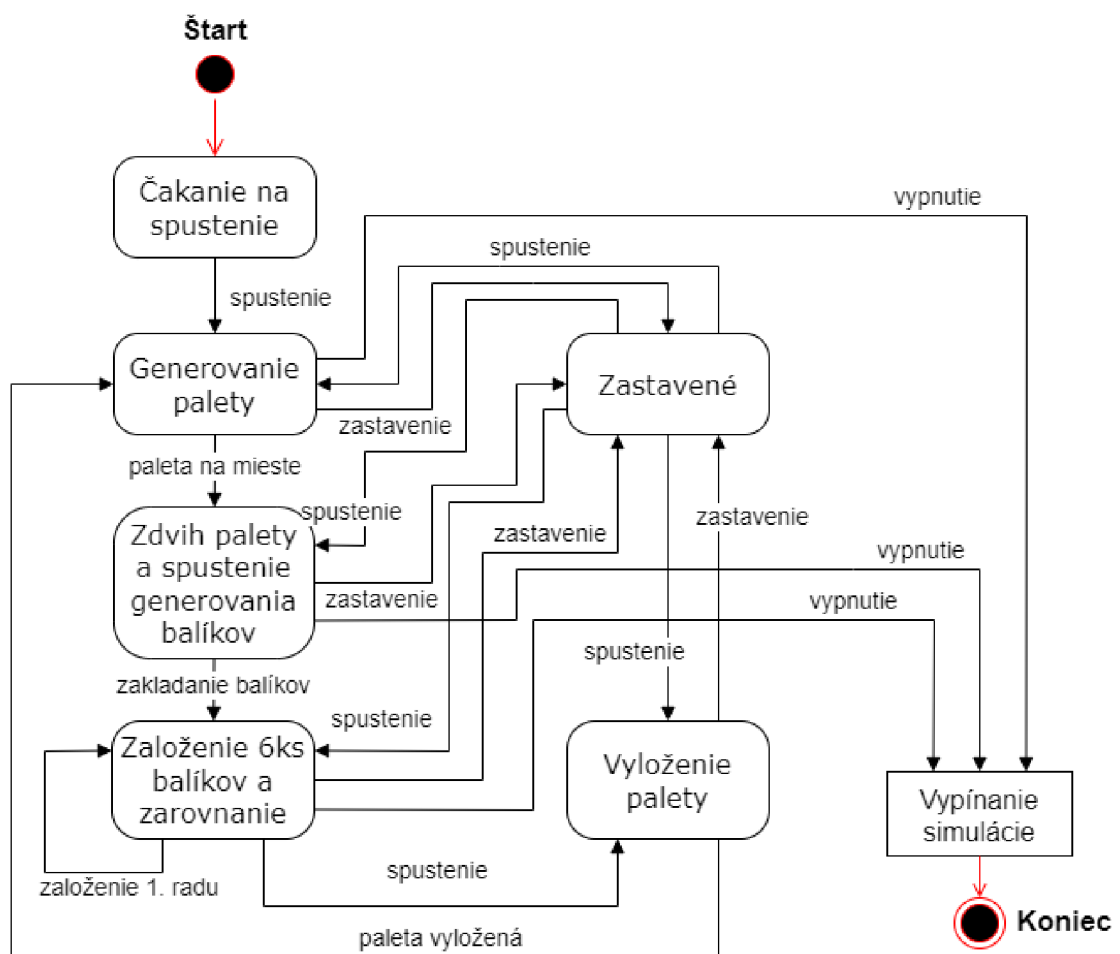


Obr. 3.5: Stavový diagram pre zakončenie vykladania

3.5 Zakladanie balíkov

Predposledným podprogramom je "paletizer". Tento podprogram riadi samostatnú časť kódu, ktorou je zakladanie balíčkov. Ako prvé sa začnú generovať palety, spustí sa dopravník na prázdne palety a reťaz, ktorá vkladá prázdne palety do výťahu. Ak je paleta vložená do výťahu, dopravník a reťaz sa zastavia. Následne sa paleta zdvihne na hornú pozíciu a začnú sa generovať balíčky, ktoré sa budú ukladať na prázdnu paletu. V prvom rade sa balíčky ukladajú vertikálne. Sú založené tri krát po dva balíčky. Následne sa urovnajú, otvorí sa brána a sú vložené na prázdnu paletu. Potom sa brána zatvorí, výťah sa posunie o jednu pozíciu nižšie a balíčky sa otočia o 90 stupňov. V druhom rade sa balíčky, otočené o 90 stupňov, zakladajú dva

krát po tri balíčky. Proces so zarovnaním sa opakuje. Po uložení druhého radu je zaplnená paleta poslaná na založenie do skladu a balíčky sa zase otočia o 90 stupňov a proces sa opakuje. Celý tento proces som zobrazil aj v stavovom diagrame na nasledujúcom obrázku 3.6. Zakladanie balíčkov nie je ovplyvňované bezpečnosťou z predchádzajúcich podprogramov. V návrhu som uvažoval o tom, že linku bude kontrolovať jeden operátor. Vzhľadom k tomu som ako jeho najvýhodnejšie miesto vykonávania práce určil pozíciu v blízkosti tejto časti linky. Operátor bude dohliadať na celý proces a ak by došlo k nežiadúcemu stavu, linku zastaví tlačidlom núdzového zastavenia (emergency stop).



Obr. 3.6: Stavový diagram pre zakladanie balíčkov

3.6 Simulácia

Posledným podprogramom je simulácia. Tento podprogram je volaný vždy a nie je potrebné splniť žiadnu bezpečnosť. Pridal som ho preto, aby ostatné podprogramy

boli prehľadnejšie a kratšie. Jeho hlavnými funkciami je reset všetkých častí a prida-
nie premenných potrebných vo vizualizácii linky. Týmito premennými sú napríklad
premenne S0 moving alebo S1 moving. Signalizujú pohyb zakladača do skladu.

```
1
2 if run AND NOT stop_sim then
3     running:=1;
4 end_if;
5
6 if NOT run OR NOT safety_door_OK then
7     //stop generovania paliet
8     if stav_gen=1 then
9         pas2_0:=0;
10        pas2_1:=0;
11        pas0CW:=0;
12        load0:=0;
13        stop_sim:=1;
14    else
15        stop_sim:=1;
16    end_if;
```

Obr. 3.7: Stavový diagram pre zakladanie balíkov

4 HMI

V tejto kapitole sa budeme venovať procesu tvorby obrazoviek pre HMI (Human Machine Interface). Budem rozoberať štandardy, podľa ktorých som postupoval pri tvorbe týchto obrazoviek. Okrem toho sa budem zaoberať aj tým, ako som vytváral a navrhoval obrazovky s ohľadom na ich funkčnosť. Ako som zabezpečoval prepojenie medzi nimi a ako som riešil ich ovládanie.

4.1 Štandard ANSI/ISA-101.01-2015

Pri návrhu obrazoviek pre HMI som sa riadil štandardom ANSI/ISA-101.01-2015 a ISA-TR101.02-2019. Štandardy sú technické správy vydávané Medzinárodnou spoločnosťou automatizácie (ISA), ktorá sa zaoberá použiteľnosťou a výkonom HMI v priemyselných aplikáciách. Správy obsahujú pokyny pre návrh a implementáciu používateľského rozhrania a poskytujú odporúčania pre zlepšenie použiteľnosti HMI. Okrem iného sa zaoberú aj otázkami ako bezpečnosť, školenie používateľov a optimalizácia work flow. Dizajn HMI podporuje primárne úlohy používateľov, ktorými sú monitorovanie a kontrola procesov. Dizajn HMI by mal minimalizovať vplyv sekundárnych úloh, ktoré môžu rozptyľovať. Takýmito úlohami môžu byť napr. navigácia displeja v rámci HMI. [17] Pri navrhovaní HMI sa berú do úvahy tieto všeobecné zásady:

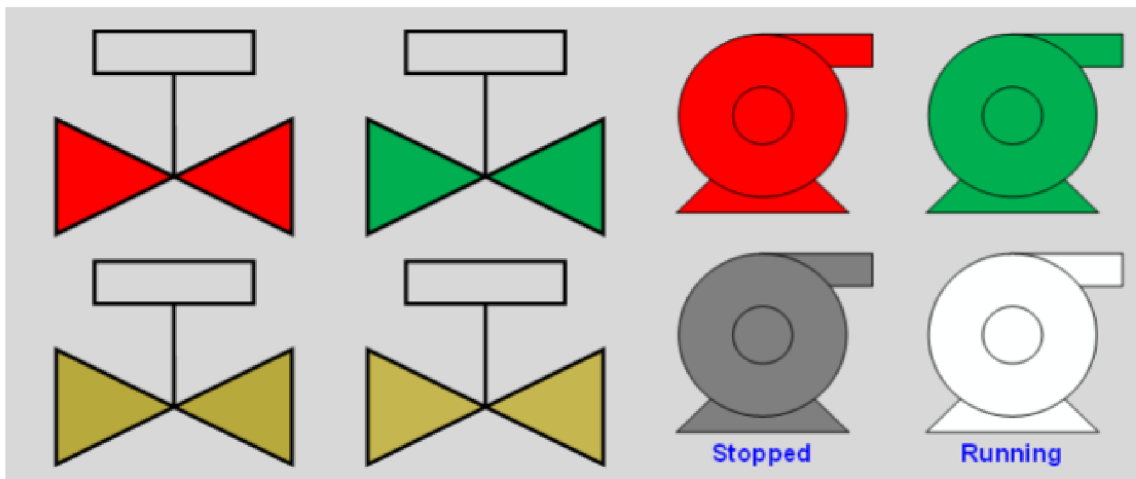
- HMI je účinný nástroj na bezpečné a efektívne riadenie procesu.
- HMI pomáha pri včasnej detekcii, diagnostike a správnej reakcii na abnormálne situácie.
- HMI je štruktúrované tak, aby pomohlo operátorom uprednostniť odozvu na veľké alebo viacnásobné simultánne poruchy systému.
- Porucha displeja alebo položiek na displeji sú operátorovi okamžite zrejmé.

HMI by malo mať konzistentný „vzhľad a dojem“ pre zobrazenie informácií a interakciu s používateľom. Funkcie v HMI by mali byť pre používateľa intuitívne a navrhnuté tak, aby podporovali úlohy súvisiace so všetkými bežne používanými režimami prevádzky. Informácie by mali byť prezentované vo formách, ktoré sú pre používateľa vhodné. Ak operátor vykoná odčítanie poľa v metroch, HMI by malo zobrazovať informácie v tomto formáte a nie v percentách alebo rozsahu.

Cieľom HMI je podporovať uvedomenie si stavu systému a procesov. Ak proces funguje správne, displej by mal byť čo najmenej rušivý. Avšak, ak sa proces odchyľi od očakávaného stavu, HMI by mala poskytovať vizuálne signály, ktoré budú zodpovedať závažnosti danej situácie. Nedostatočné povedomie o situácii bol identifikovaný

vané ako jeden z primárnych faktorov nehôd, ktoré boli spôsobené ľudskou chybou. Fixácia na určité informácie a zanedbanie ostatných môže obmedziť uvedenie o situácii. Ak sa operátor sústreďí iba na jeden problém a zanedbáva ostatné potenciálne dôležité informácie, môže to viesť k obmedzeniu povedomia o situácii. [17]

Návrh HMI by mal zahŕňať princípy, ktoré sú založené na senzorických limitoch používateľa, napr. farbosleposť. Je to nutné z dôvodu, že niektorí ľudia nemusia rozlíšiť červenú od zelenej farby. ISA-TR101.02-2019 uvádza, že až 8% mužov a 0.5% túto schopnosť nemá. Ukážku toho, ako tieto farby vidia, môžete vidieť na obrázku 4.1 vľavo. Vpravo je ukážka toho, ako by malo byť správne rozlíšený aktívny a neaktívny stav, napr. farebným kontrastom viditeľným pre všetkých alebo jednoznačným textom určujúcim stav. Taktiež je odporúčané využívať červenú na alarmy alebo pre zobrazenie neaktívneho stavu. Žltá farba sa využíva pre warningy (upozornenia) a zelená na zobrazenie aktívneho stavu.

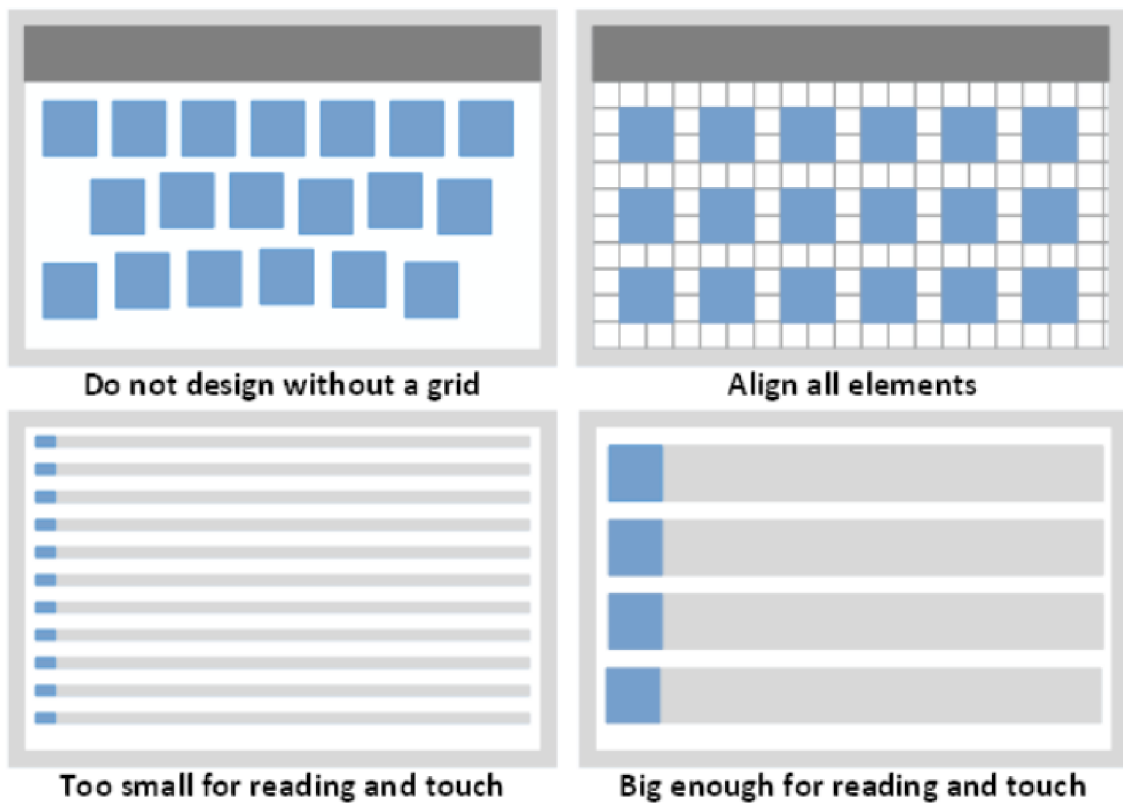


Obr. 4.1: Usporiadanie HMI

[16]

Jas obrazovky HMI by mal byť vhodný vzhľadom na okolité prostredie. Zabrániť namáhaniu očí tým, že sa vyhneme použitiu nadmerného množstva kontrastov. Pozadie by malo byť neutrálnej farby, aby sa zabezpečil kontrast a teda dobrá viditeľnosť zobrazovaných informácií. Špeciálne sa však treba vyhnúť čiernemu pozadiu.

Informačná hustota displeja by mala vychádzať z funkcie alebo účelu displeja. Je odporúčané zobrazovať iba relevantné informácie, s ktorými je možné hneď interagovať. Dôležité je aj usporiadanie. Pri návrhu je potrebné dodržiavať mriežku a neurobiť ich príliš malé, aby bolo možné čítať a ovládať prvky v HMI, viď nasledujúci obrázok 4.2.



Obr. 4.2: Usporiadanie objektov v HMI

[16]

V prípade, že požiadavky používateľa vedú k nadmernému množstvu informácií zobrazovaných na jednom displeji, je potrebné vykonať určité úpravy. Týmito úpravami môžu byť:

- využiť informácie s menšou celkovou hustotou na obrazovke, napr. zmena formátu zobrazenia alebo zjednodušenie informácií
- využiť štýl zobrazenia, ktorý umožňuje operátorovi rýchlo a ľahko identifikovať dôležité informácie a zanedbať tie, ktoré nie sú také dôležité
- poskytovanie niektorých informácií len vtedy, ak používateľ o ne požiada
- rozdelenie displeja na viac displejov, čo umožňuje rozdeliť informácie a zobrazovať ich na rôznych displejoch podľa ich významu a priority.

4.2 Navigačná schéma medzi obrazovkami

Dôležitou súčasťou dizajnu HMI je navigačná schéma. Správne navrhnutá navigácia môže zlepšiť rýchlosť a presnosť obsluhy interakcie s HMI. Základmi úspešnej navigácie sú konzistentnosť a intuitívnosť. S cieľom dosiahnuť čo najlepší výsledok, je dôležité zamerať sa na to, aby bola navigačná schéma prehľadná, logická a ľahko zrozumiteľná pre používateľov. V závislosti od požiadaviek používateľa je možné poskytnúť viacero spôsobov navigácie. Medzi bežné typy navigačných schém patria hierarchický, relačný alebo sekvenčný typ. Hierarchický dizajn je často používaný na organizáciu informácií podľa fyzickej organizácie procesu, kde sa informácie zobrazujú pomocou obrátenej stromovej štruktúry. Pri použití hierarchickej navigácie sa často definujú typy zobrazenia pre každú úroveň hierarchie. Relačný návrh displejov obsahuje viacero prepojení medzi uzlami, ktoré sú založené na rôznych vzťahoch. V prípade, že navigačná cesta závisí od typu situácie, formálna štruktúra vzťahov môže byť najefektívnejším riešením. Tento dizajn sa často uplatňuje v distribuovaných úžitkových systémoch, kde sa vyžaduje navigácia pre všetkých výrobcov, spotrebiteľov alebo kontrolu odpojenia systému. Sekvenčný dizajn usporiada displeje v určitom poradí, čo môže byť efektívne v prostredí s postupným prechodom logickou štruktúrou, ako napríklad v dávkovom procese. Avšak táto štruktúra sa môže meniť v závislosti na konkrétnych výrobkoch. [17] Pri návrhu navigačnej schémy by sme mali mali myslieť na tieto kľúčové aspekty:

- prístup k displeju by mal byť navrhnutý tak, aby minimalizoval činnosti ekvivalentné stlačeniu klávesov operátora. Pozri obrázok nižšie, na ktorom je Tabuľka 8 [17]
- nemalo by sa pristupovať k obrazovke pomocou zadania jej názvu alebo indexu
- symboly ovládajúce displeje by mali byť konzistentné a viditeľné
- voliteľné symboly na ovládanie displeja by taktiež mali byť konzistentný dizajn
- displeje by mali byť implementované v štandardizovanej obsahovej štruktúre, ktorá podporuje progresívne vystavenie podrobných informácií. Obsahová štruktúra by mala súvisieť s logickým zoskupením, ktoré je pre operátora intuitívne a podporuje funkčné vzťahy.
- HMI by malo umožňovať pracovný postup za normálnych, ale aj za abnormálnych podmienok.

Table 8 – Example access and navigation performance

Metric	Display type	Maximum access times
Access to alarm displays	Alarm summary	1 sec
	Alarm lists such as suppressed alarms	5 secs
Navigation (Note 1)	Critical displays	1-2 clicks
	Non-critical displays	3 clicks
	Alarm summary	1 click
	System diagnostics	1-2 clicks
System state changes (Note 2)	Switching operators	5 secs
	Language change	5 secs

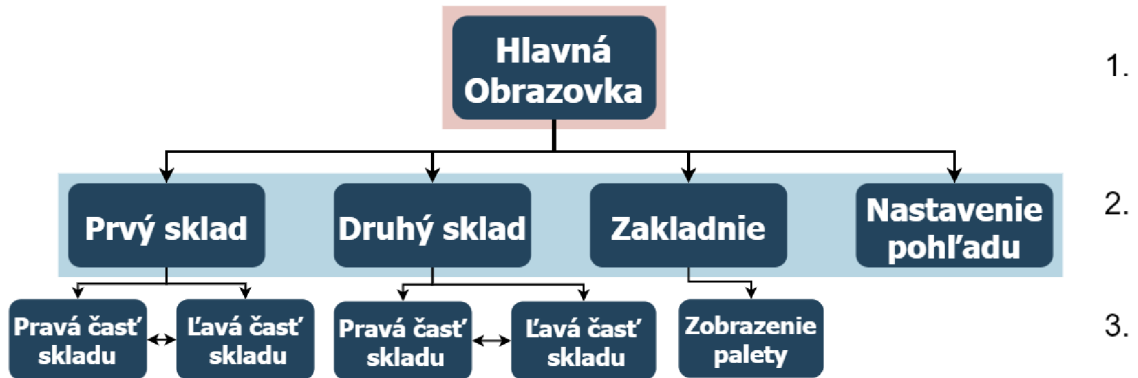
Obr. 4.3: Tabuľka odporúčaného prístupu a navigácie

[17]

4.3 Prepojenie obrazoviek v mojom návrhu

Pri návrhu navigácie medzi displejmi som sa riadil predchádzajúcou podkapitolou. Vytvoril som čo najjednoduchší návrh a zároveň som zachoval všetko potrebné na ovládanie virtuálnej výrobnéj linky. Taktiež som zakomponoval všetky odporúčania, ako napríklad farebné rozlíšenie, mriežku alebo aj kontrast medzi pozadím a prvami displeja. Návrh HMI pozostáva z viacerých úrovní displejov. V prvej úrovni sa nachádza hlavná obrazovka. V druhej je prvý sklad, druhý sklad, zakladač balíkov a nastavenia pohľadu. Vytvoril som aj tretiu úroveň pre obrazovky skladov a zakladača balíkov, nakoľko by sa do predchádzajúcich, kvôli množstvu dôležitých informácií,

nezmestili. Pri vytváraní tretej úrovne som zohľadnil metódy spomenuté v predchádzajúcej podkapitole, zredukoval informácie iba na nevyhnutné a ostatné rozdelil do ďalších obrazoviek. Konkrétne som vytvoril dve obrazovky pre každý sklad a to pravú časť a ľavú časť. Vytvoril som aj obrazovku zobrazenia palety. Pre lepšiu predstavu som vytvoril diagram, ktorý popisuje všetky vzťahy medzi obrazovkami.



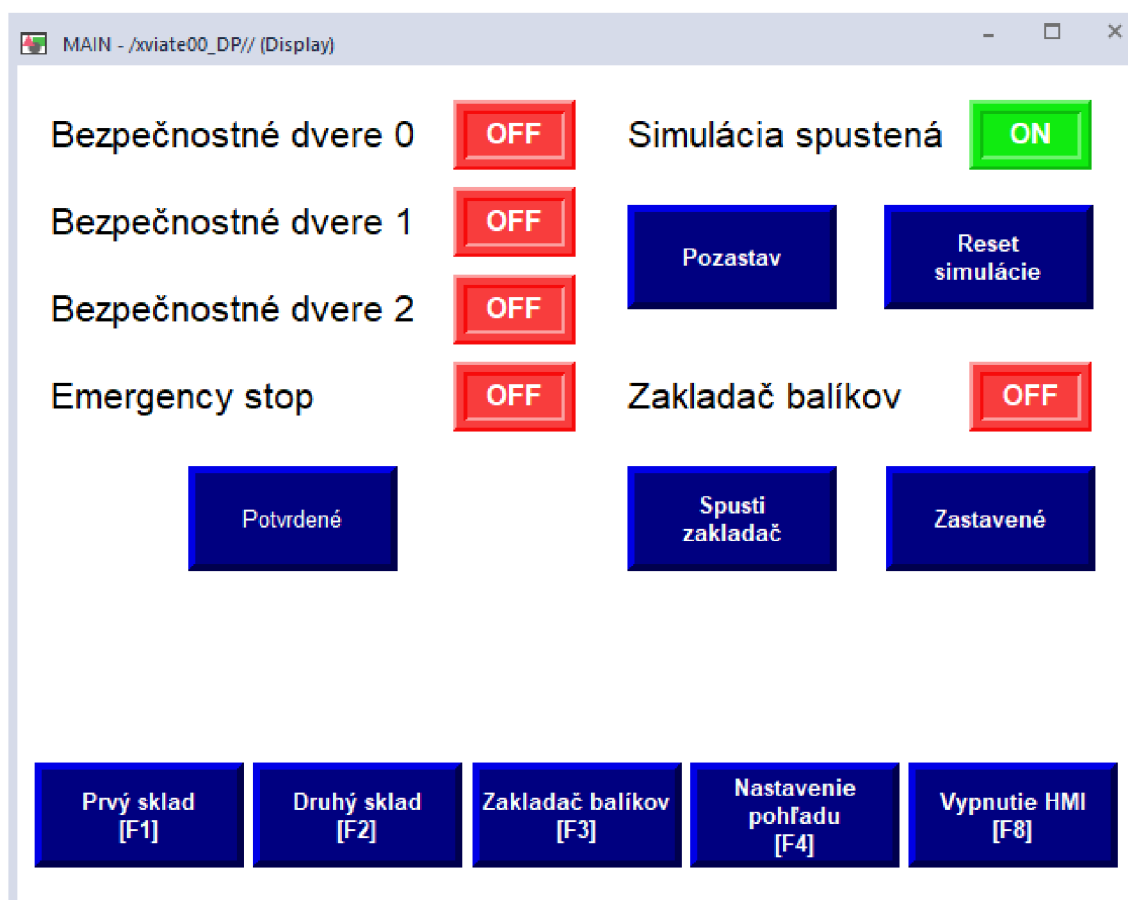
Obr. 4.4: Diagram vzťahov

Takáto štruktúra zabezpečuje to, aby som bol vždy o 1-2 kliky od hlavnej úrovne a zároveň je všetko prehľadné. Taktiež je v nej zakomponované všetko potrebné na ovládanie linky a zobrazovanie všetkých potrebných parametrov.

4.3.1 Hlavná obrazovka

Prvou obrazovkou je hlavná obrazovka. Tu sa nachádzajú všetky potrebné tlačidlá a indikátory na ovládanie virtuálnej výrobnéj linky. Naľavo sú všetky bezpečnostné prvky z môjho návrhu. Nachádza sa tam signalizácia všetkých troch bezpečnostných dverí a aj Emergency stop, ktorý slúži na zastavenie zakladača balíkov v nebezpečnom stave. Hneď pod týmito indikátormi je tlačidlo na potvrdenie bezpečnosti. Toto tlačidlo sa využíva, keď sa otvoria niektoré z bezpečnostných dverí. Keď dôjde k otvorení dverí, je možné, že osoba za sebou atvorí dvere a linka sa opätovne spustí. Môže dôjsť k zraneniu danej osoby, preto bolo potrebné pridať tlačidlo, ktorým sa potvrdzuje, že osoba nebezpečný priestor opustila a linka môže ďalej pokračovať v činnosti. Vpravo sú zase tlačidlá a indikátory na spustenie simulácie a zakladača balíkov. Vpravo hore je ako prvý indikátor pre simuláciu. Tento indikátor ukazuje, či je simulácia v danom okamihu spustená. Hneď pod ním sú tlačidlá na spustenie, respektíve zastavenie a na resetovanie simulácie. Kvôli praktickosti bol zakladač umiestnený osobitne. Indikátor signalizuje, či je zakladač spustený. Zakladač balíkov je ovládaný podobne ako simulácia nad ním, teda dvomi tlačidlami na spustenie a zastavenie. Súčasťou hlavnej obrazovky sú aj tlačidlá v spodnej časti obrazovky. Tieto tlačidlá slúžia na preklik na ostatné obrazovky, ktoré boli spomenuté v návrhu. S

týmito tlačidlami je možné interagovať aj pomocou tlačidiel priamo na displeji, ako napovedá nápis na tlačidle. Ukážku tejto obrazovky môžete vidieť na nasledujúcom obrázku 4.5.



Obr. 4.5: Hlavná obrazovka

4.3.2 Prvý sklad

Ďalšou obrazovkou je obrazovka druhej úrovne a to konkrétne obrazovka prvého skladu. Táto obrazovka je veľmi podobná obrazovke druhého a preto popíšem iba túto jednu. Táto obrazovka je rozdelená do troch segmentov. Vľavo sa nachádzajú indikátory pre zakladač paliet do skladu. Sú tu všetky potrebné indikátory zobrazenia stavu, v ktorom sa zakladač práve nachádza. Je napríklad možné vidieť, či sa hýbe alebo je aktívny zdvih a podobne. Taktiež je zobrazená pozícia nasledujúcej palety v ose x aj v ose y. Dôvodom je ukladanie paliet do skladu podľa súradníc x a y. Napravo sú dôležité indikátory, ktoré zobrazujú v akom stave je simulácia aktuálne. Stav simulácie, ktorý je zobrazovaný je vkladanie do skladu alebo vykladanie zo skladu. Taktiež môžete vidieť, či sú dopravníky v aktívnom stave. Posledným tretím je spodná lišta. V nej sa nachádzajú tlačidlá na zmenu obrazovky a hlavné

indikátory bezpečnosti. Boli pridané preto, aby bolo za každej okolnosti viditeľné, či nie je porušená bezpečnosť. Z tejto obrazovky je možné dostať sa späť na hlavnú obrazovku, vyloženie skladu alebo na obrazovku nastavenia pohľadu. Táto linka bola navrhnutá tak, aby fungovala prevažne v automatickom režime, preto som volil skôr zobrazovanie stavov ako ich menenie. V obrazovke vyloženie som pridal prvky vyloženia zo skladu, čo opíšem neskôr. Obrazovku prvého skladu môžete vidieť na nasledujúcom obrázku 4.6.

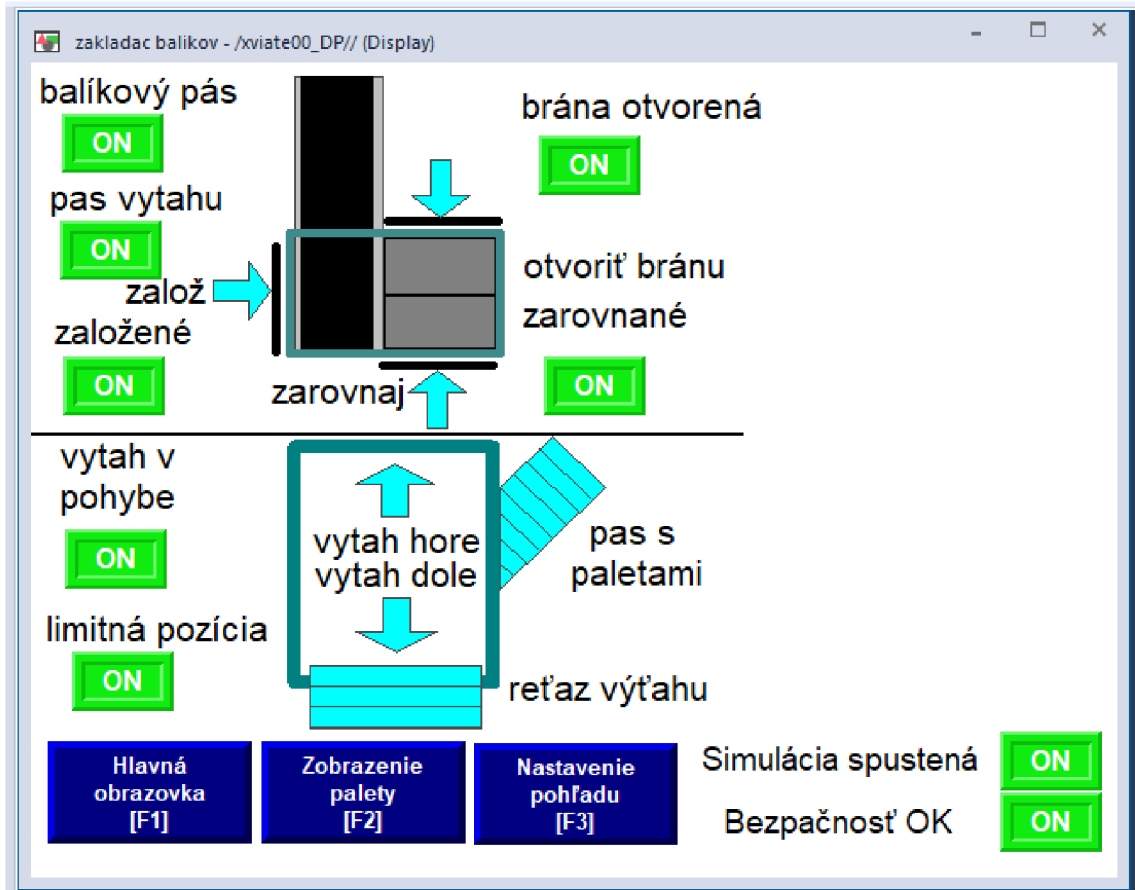


Obr. 4.6: Prvý sklad

4.3.3 Zakladač balíkov

Nasledujúcou obrazovkou druhej úrovne je obrazovka zakladača balíkov. V tejto obrazovke je navrhnutá spodná lišta, v ktorej sú tlačidlá na zmenu obrazovky a indikátory bezpečnosti. Z tejto obrazovky sa dá vrátiť na hlavnú obrazovku alebo ísť na obrazovku nastavenia pohľadu. Hlavnou časťou je zvyšok obrazovky. V hornej časti môžete vidieť pohľad na zakladač zhora. Nachádzajú sa na nej všetky indikátory využívané pre zakladanie balíkov na paletu. Napr. indikátory pre dopravníky,

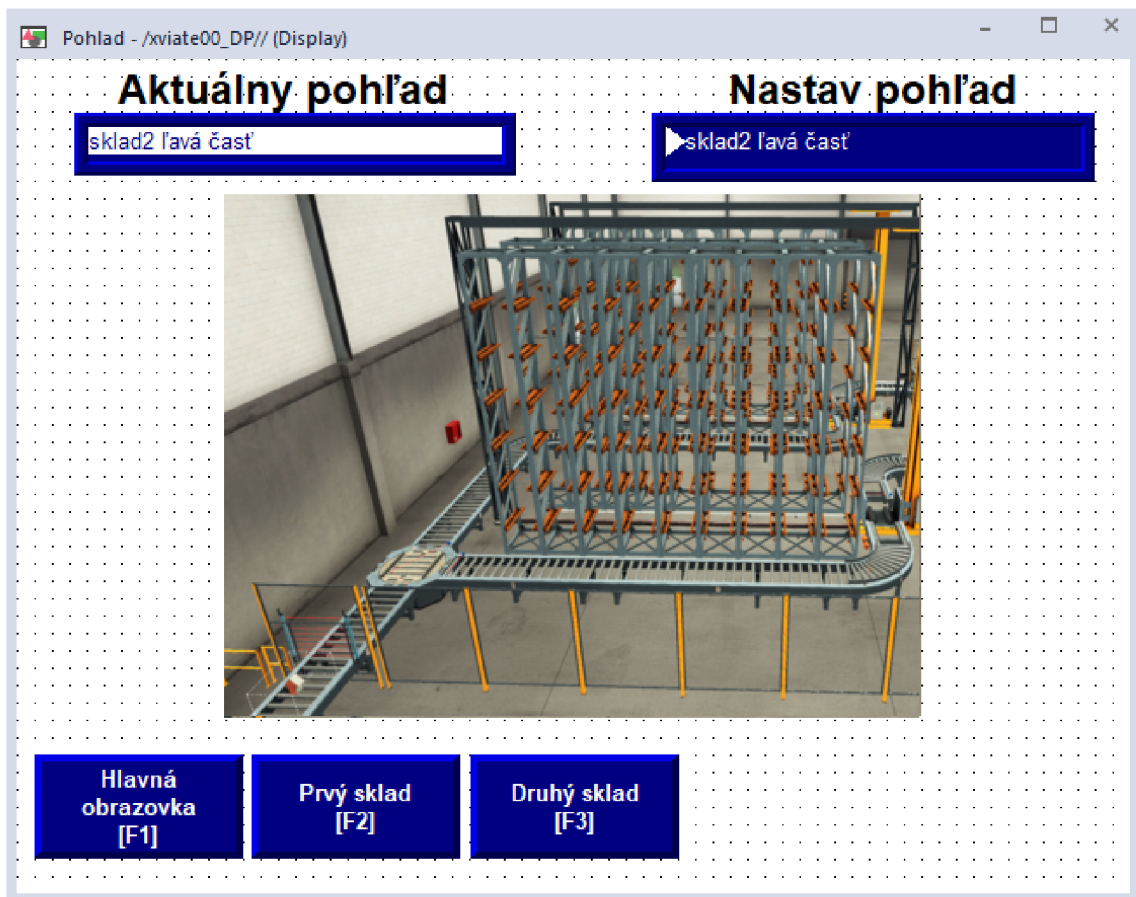
signalizácia zakladania, signalizácia zarovnania a otvorenie brány. V dolnej časti je pohľad na zakladač spredu. Vpravo je dopravník, ktorý posúva palety do výťahu. V strede je umiestnená signalizácia reťaze, ktorá posúva paletu vrámci výťahu. Taktiež sú tam aj indikátory pohybu výťahu hore alebo dole. Túto obrazovku môžete vidieť na obrázku 4.7.



Obr. 4.7: Zakladač balíkov

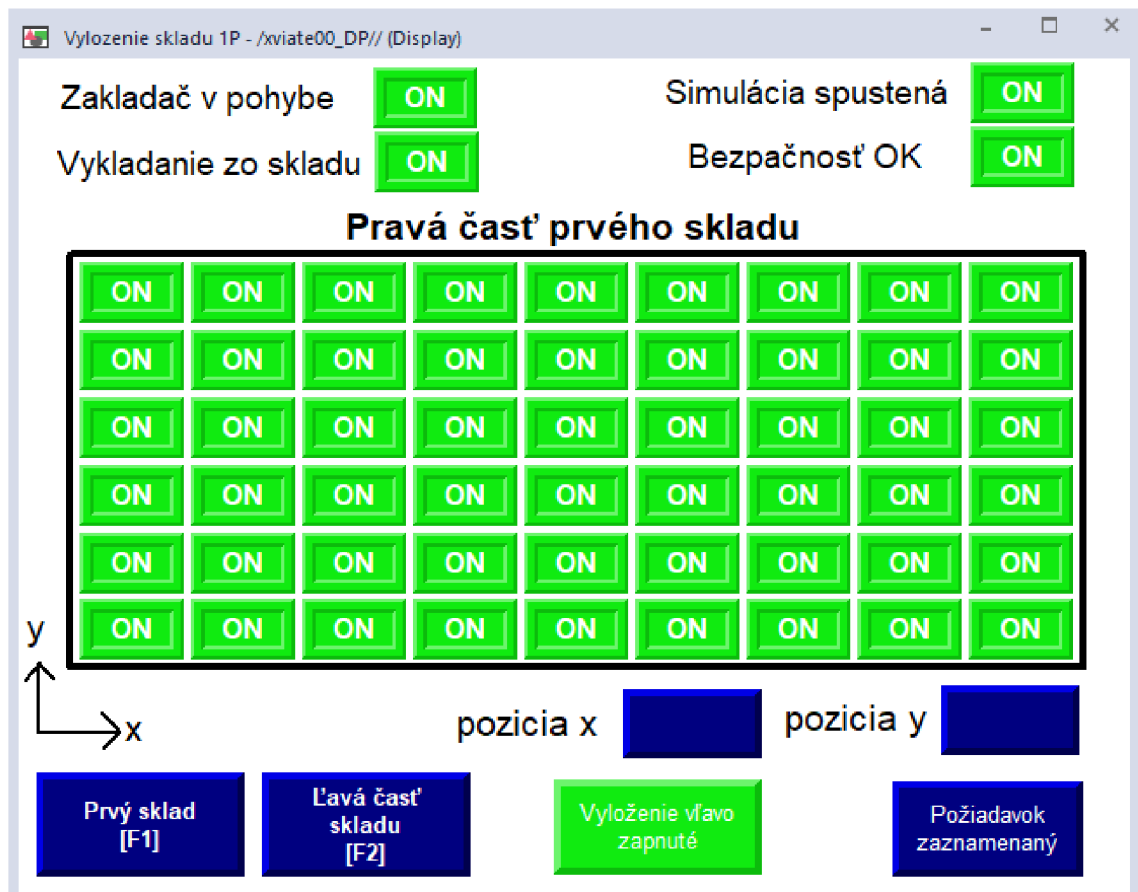
4.3.4 Nastavenie pohľadov

Poslednou obrazovkou druhej úrovne je obrazovka nastavenia pohľadov. Na túto obrazovku sa dá dostať zo všetkých obrazoviek druhej úrovne a taktiež naspäť. Bola vytvorená z dôvodu zmeny náhľadu vo Factory I/O. Pri absencii klávesnice a myši je takto zmena náhľadu stále možná pomocou dopredu uložených pozícií. Bolo ich uložených celkom sedem. Pomenovanie majú podľa, čo je na danom pohľade viditeľné. Prvým je pohľad na prvý sklad, druhým na druhý, tretím na zakladač balíkov, štvrtým je presnejší pohľad na pravú časť prvého skladu, piatym na ľavú časť toho istého skladu, šiestym na pravú časť druhého skladu a posledným siedmym na ľavú časť toho istého skladu. Táto obrazovka je na nasledujúcom obrázku 4.8.



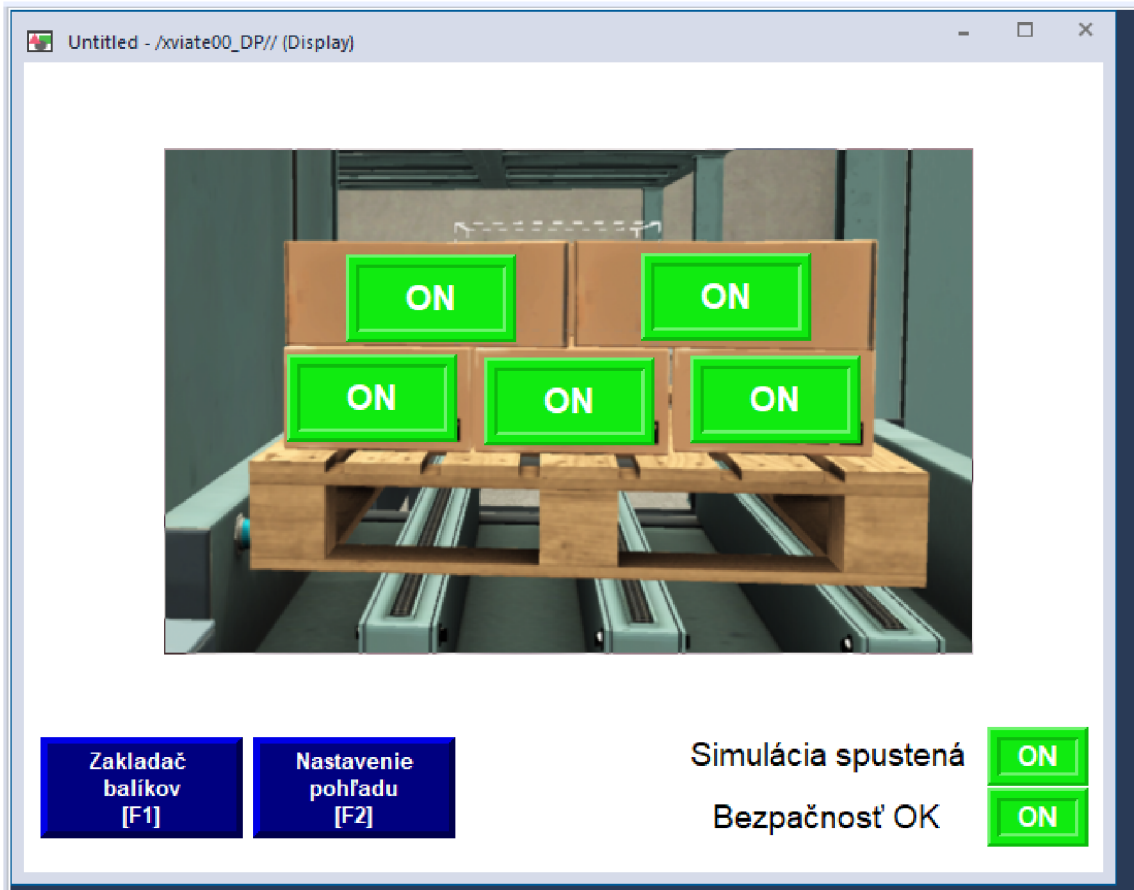
Obr. 4.8: Nastavenie pohľadu

Obrazoviek tretej úrovne je päť. Konkrétne jedna obrazovka pre "zobrazenie palety" (bližšie ju popíšem v nasledujúcom odstavci) a po dve obrazovky pre oba sklady kvôli rozdeleniu zobrazenia na pravú a ľavú časť. Pravá a ľavá časť sú takmer identické. Rovnaká situácia je aj v druhom sklade, preto popíšem iba jednu obrazovku. Na obrazovke je možný presun iba na druhú časť skladu alebo na pôvodný sklad. Týmto usporiadaním by nemalo dôjsť k zmäteniu užívateľa. Obrazovka je dôležitá z hľadiska zobrazovania všetkých pozícií v sklade. Užívateľ môže sledovať, ktoré sú obsadené a ktoré nie. Výhodou je možnosť súčasného sledovania konkrétnej pozície a zároveň jej vyloženia zo skladu. Ak chce užívateľ vyprázdniť niektorú z pozícií, zadá jeho pozíciu v súradnici osy x a y. Následne výber potvrdí tlačidlom ENTER. Ak je táto požiadavka v poriadku (nevybral prázdnu pozíciu), zobrazí sa "požiadavka zaznamenaná" čaká sa na uvoľnenie zakladača. Predvolené je vykladanie vpravo. Ak chce užívateľ vyložiť pozíciu vľavo, musí stlačiť tlačidlo Vyloženia vľavo. Toto tlačidlo sa nachádza v oboch častiach skladu. Používateľ vždy vie, či vykladá pozíciu vľavo alebo vpravo.



Obr. 4.9: Pravá časť prvého skladu

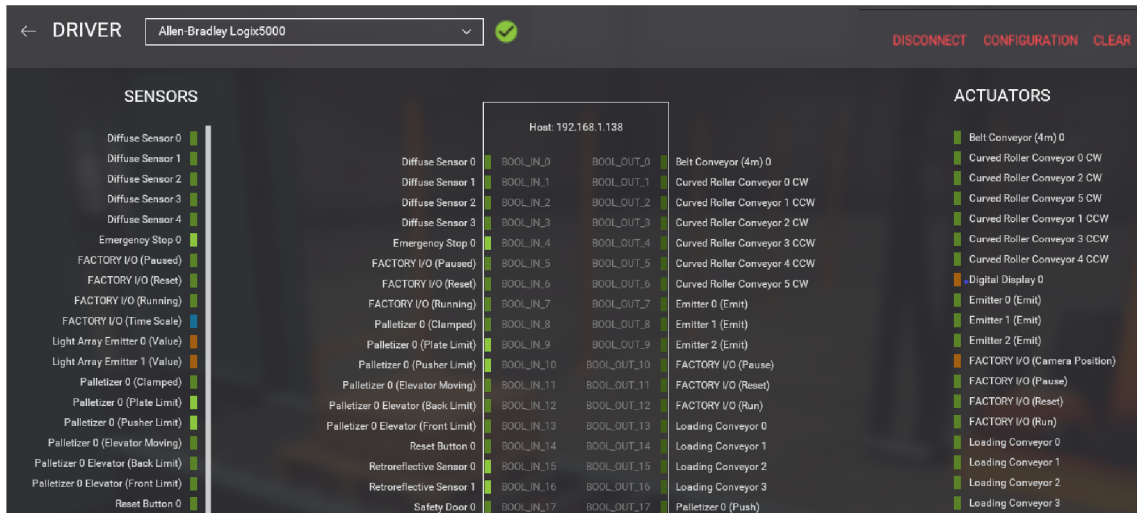
Poslednou obrazovkou tretej úrovne je obrazovka "zobrazenie palety". V spodnej časti, ako pri ostatných obrazovkách, sú tlačidlá na prepínanie medzi displejmi, indikátor bezpečnosti a indikátor stavu simulácie. Z tejto obrazovky sa dá vrátiť na zakladač balíkov alebo ísť na obrazovku nastavenia pohľadu. V hlavnej časti obrazovky (v strede) sa zobrazuje stav aktuálne založených balíkov na paletu. V prvom rade sú balíky zakladané po dva kusy. Založené sú tri krát, preto sú tak tri indikátory. V druhom rade sú balíky zakladané po tri kusy. Založenie sa uskutoční dva krát, preto sú vo vizualizácii dva indikátory. Túto obrazovku môžete vidieť na obrázku 4.10.



Obr. 4.10: Zobrazenie palety

5 Overenie riešenia

Pre overenie riešenia je potrebné najskôr nakonfigurovať komunikáciu medzi FACTORY I/O a PLC. Návrh vytvorenia tejto komunikácie som opísal v kapitole 2.4. Po pripojení PLC pomocou ethernetu a vytvorení všetkých tagov, ju bolo možné overiť, viď obrázok 5.1.



Obr. 5.1: Pripojený hardvér

Po pripojení som skontroloval softvér potrebný pre riadenie linky, ktorý bol opísaný v kapitole 3. Bol nahratý do PLC a spustená simulácia linky. Linka fungovala podľa požadovaných vlastností, zakladače skladov ukladali palety do skladu. Pri každej požiadavke na vyloženie bolo potrebné počkať, kým zakladač nebude obsadený. Taktiež som demonštroval funkčnosť bezpečnosti. Zakladač balíkov plnil palety podľa požiadaviek, viď obrázok 5.2.



Obr. 5.2: Pripojený hardvér

Záver

Cielom tejto semestrálnej práce bolo riadenie virtuálnej výrobnéj linky, navrhnutie softvéru na riadenie a následne vytvorenie HMI panela. V úvode práce som opísal program FACTORY I/O. Uviedol som, ako sa upravujú scény, pridávajú a otáčajú diely a podobne. Následne som sa zaoberal návrhom laboratórnej úlohy. Po návrhu virtuálnej výrobnéj linky som predstavil softvér potrebný na ovládanie PLC. Nakoniec som vytvoril obrazovky pre HMI panel

V prvej kapitole bol predstavený FACTORY I/O. V tomto programe bola navrhnutá virtuálna výrobná linka. Boli v nej opísané všetky štyri časti programu a ich hlavný účel. Taktiež boli vymenované všetky možnosti navigácie v scéne, ako napríklad orbitálna kamera alebo first person kamera. Venoval som sa aj možnostiam vytvárania scén a ich následnej úprave pre ďalšie použitie. Zaoberal som sa aj možnostami spustenia simulácie a to konkrétne spustenie zrýchleného a spomaleného módu simulácie. V podkapitole 1.7 som predstavil možnosť ovládania scény pomocou externého ovládacieho zariadenia, akým je PLC.

V druhej kapitole som ukázal návrh laboratórnej úlohy. Laboratórna úloha pozostávala z automatizovaných skladov a zakladača balíkov. Navrhol som zakladače skladov, tak že je možné palety ukladať a aj vykladať. Pri vykladaní som vyriešiť prioritu vykladania. Predstavil som aj návrh zakladača balíkov(výrobný prvok). Zakladač ukladá balíky na paletu najskôr horizontálne a potom vertikálne. Tento spôsob som zvolil preto, aby balíky lepšie držali pri preprave. Navrhol som aj bezpečnostné prvky, ktoré by mohli byť súčasťou reálneho skladu. Nakoniec som predstavil návrh nakonfigurovania komunikácie.

V tretej kapitole som opísal riadiaci softvér. Ukázal som jeho rozdelenie na podprogramy a popísal ich funkčnosť. Venoval som sa aj tomu, ako ovplyvňuje ich vykonávanie bezpečnosť linky a ako ich následne spustiť, ak bola bezpečnosť porušená. Súčasťou kapitoly boli aj stavové diagramy, ktoré najlepšie popisujú činnosť podprogramov.

V štvrtej kapitole som predstavil štandard ANSI/ISA-101.01-2015. Opísal som jeho hlavné myšlienky a prvky, ktoré som použil v mojom návrhu HMI panelu. Ukázal som aj návrh prepojenia medzi obrazovkami. Podrobne som opísal činnosti, ktoré sa dajú vykonávať na obrazovke.

V poslednej kapitole otestoval komunikáciu medzi FACTORY I/O a PLC. Skontroloval som funkčnosť svojho návrhu a ukázal reálne fungovanie linky.

Literatúra

- [1] Factory I/O: *About* [online][cit. 20. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com>.
- [2] Manual: *User Interface* [online][cit. 20. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/manual/user-interface/>.
- [3] Manual: *Navigation* [online][cit. 21. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/manual/navigation/>.
- [4] Getting Started: *Navigating-Orbit Camera* [online][cit. 21. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/getting-started/navigating/#1-orbit-camera>.
- [5] Getting Started: *Navigating-Fly Camera* [online][cit. 21. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/getting-started/navigating/#2-fly-camera>.
- [6] Getting Started: *Navigating-First Person Camera* [online][cit. 21. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/getting-started/navigating/#3-first-person-camera>.
- [7] Manual: *Navigation-cameras window* [online][cit. 21. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/manual/navigation/#cameras-window>.
- [8] Manual: *Navigation-camera gizmo* [online][cit. 21. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/manual/navigation/#camera-gizmo>.
- [9] Manual: *Navigation-Follow a Part* [online][cit. 21. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/manual/navigation/#cameras-window>.
- [10] Manual: *Scenes* [online][cit. 22. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/manual/scenes/>.
- [11] Manual: *Edit and Run* [online][cit. 22. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/manual/edit-and-run/#edit-mode>.
- [12] Manual: *Edit and Run* [online][cit. 22. 12. 2022]. Dostupné z URL: <https://docs.factoryio.com/manual/edit-and-run/#run-mode>.

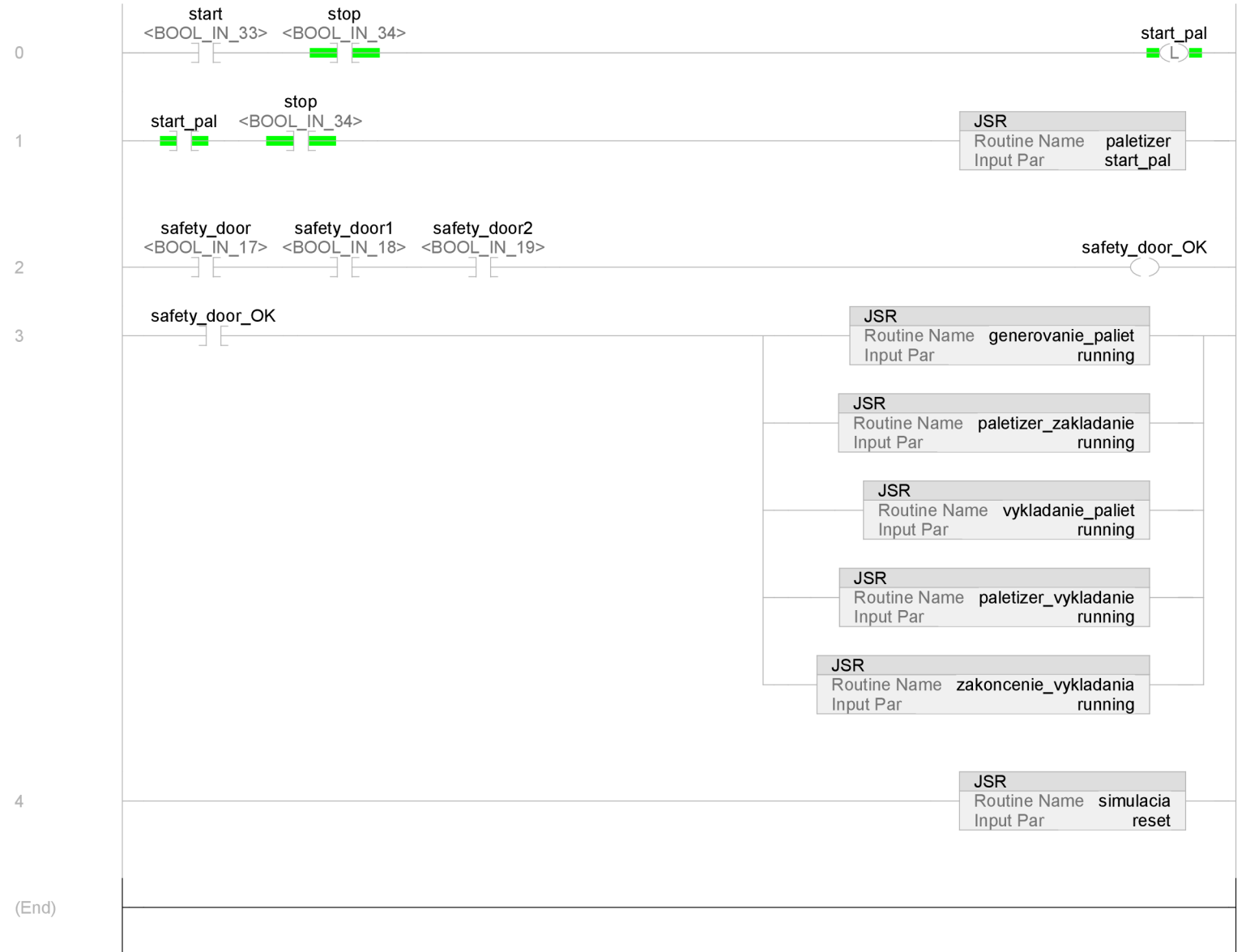
- [13] Manual: *Tags* [online][cit. 22.12.2022]. Dostupné z URL:
<<https://docs.factoryio.com/manual/tags/>>.
- [14] Getting Started: *Manually Controlling a Scene* [online][cit. 23.12.2022].
Dostupné z URL:
<[https://docs.factoryio.com/getting-started/
manually-controlling-scene/](https://docs.factoryio.com/getting-started/manually-controlling-scene/)>.
- [15] Manual: *I/O Drivers* [online][cit. 23.12.2022]. Dostupné z URL:
<<https://docs.factoryio.com/manual/drivers/>>.
- [16] ISA(2019): *ISA-TR101.02-2019, HMI Usability and Performance* [online][cit. 8.5.2023]. Dostupné z URL:
<<https://www.isa.org/products/isa-tr101-02-2019-hmi-usability-and-performance>>.
- [17] ANSI/ISA-101.01-2015, Human Machine Interfaces for Process Automation Systems. FactoryTalk Alarms and Events System Configuration Guide. Rockwell Automation. FTAE-RM001O-EN-E. 2022.


```
1
2 case stav_gen of
3
4   0:
5     if running then
6       stacker0_targ:=0;
7       stav_gen:=1;
8     end_if;
9
10  1:
11   if NOT stop_sim then
12     emit0:=1;
13     pas2_0:=1;
14     pas2_1:=1;
15     pas0CW:=1;
16     pas1CCW:=1;
17     load0:=1;
18
19     if dif_sen0 then
20       load0:=0;
21       pas0CW:=0;
22       pas2_0:=0;
23       pas2_1:=0;
24       pas1CCW:=0;
25       stacker0_R:=1;
26       stav_gen:=2;
27     end_if;
28   end_if;
29
30   if hold then
31     load0:=0;
32     pas0CW:=0;
33     pas2_0:=0;
34     pas2_1:=0;
35     pas1CCW:=0;
36     stav_gen:=20;
37   end_if;
38
39  2:
40   if stacker0_RL AND NOT stop_sim then
41     stacker0_lift:=1;
42   end_if;
43
44   if stacker0_RL AND stacker0_MZ then
45     stav_gen:=3;
46   end_if;
47  3:
48   if NOT stacker0_MZ AND NOT stop_sim then
49     stacker0_R:=0;
50     stav_gen:=4;
51   end_if;
52  4:
53   if stacker0_M AND NOT stop_sim then
54     stacker0_targ:=pozicia0;
55   end_if;
56
57   if pozicia0=55 then
58     x:=0;
59     y:=0;
```

```
60     end_if;
61
62
63     if stacker0_targ>54 then
64         stacker0_targ:= stacker0_targ - 54;
65         vlavo0:=1;
66     end_if;
67
68     if stacker0_MX then
69         stav_gen:=5;
70     end_if;
71 5:
72     if NOT stacker0_MX AND NOT stacker0_MZ AND NOT vlavo0 AND NOT stop_sim then
73         stacker0_R:=1;
74         //
75         if x<9 then
76             sklad0P[x,y]:=1;
77         else
78             x:=0;
79             y:=y+1;
80             sklad0P[x,y]:=1;
81         end_if;
82         //
83     elsif NOT stacker0_MX AND NOT stacker0_MZ AND vlavo0 AND NOT stop_sim then
84         stacker0_L:=1;
85         //
86         if x<9 then
87             sklad0L[x,y]:=1;
88         else
89             x:=0;
90             y:=y+1;
91             sklad0P[x,y]:=1;
92         end_if;
93         //
94     end_if;
95
96     if stacker0_LL OR stacker0_RL then
97         stacker0_lift:=0;
98     end_if;
99
100    if (stacker0_LL OR stacker0_RL) AND stacker0_MZ then
101        stav_gen:=6;
102    end_if;
103
104 6:
105    if NOT stacker0_MZ AND NOT stop_sim then
106        stav_gen:=7;
107    end_if;
108 7:
109    if stacker0_RL AND NOT stop_sim then
110        stacker0_R:=0;
111        stav_gen:=8;
112    elsif stacker0_LL AND NOT stop_sim then
113        stacker0_L:=0;
114        stav_gen:=8;
115    end_if;
116 8:
117    if stacker0_M AND NOT stop_sim then
118        stacker0_targ:=21474;
```

```
119     end_if;
120
121     if stacker0_MX then
122         stav_gen:=9;
123     end_if;
124 9:
125     if NOT stacker0_MX then
126
127         stacker0_targ:=0;
128         pozicia0:=pozicia0+1;
129         x:=x+1;
130         stav_gen:=1;
131     end_if;
132 20:
133     holding:=1;
134     if NOT hold then
135         holding:=0;
136         stav_gen:=1;
137     end_if;
138
139 end_case;
140
141 if pozicia0>108 then
142     x:=0;
143     y:=0;
144     pozicia0:=1;
145     zapln:=1;
146 end_if;
147
148 if zapln then
149     if sklad0P[x,y]=0 then
150         stav_gen:=1;
151     else
152         x:=x+1;
153         pozicia0:=pozicia0+1;
154     end_if;
155
156     if x>8 then
157         x:=0;
158         y:=y+1;
159     end_if;
160
161     if pozicia0>54 then
162         x:=0;
163         y:=0;
164         zapln_L:=1;
165     end_if;
166 end_if;
167
168 if zapln_L then
169     zapln:=0;
170
171     if sklad0L[x,y]=0 then
172         stav_gen:=1;
173     else
174         x:=x+1;
175         pozicia0:=pozicia0+1;
176     end_if;
177
```

```
178     if x>8 then
179         x:=0;
180         y:=y+1;
181     end_if;
182 end_if;
```



```
1 OSRI(rise);
2 OSFI(fall);
3
4 case stav_pal of
5     0:
6
7         if running then
8             stav_pal:=1;
9         end_if;
10
11     1:
12         if pal_running then
13             emitor2:=1;
14             pas4_5:=1;
15             palet_chain:=1;
16         end_if;
17
18         rise.InputBit:=0;
19         if retro0 AND palet_elev_back then
20             pas4_5:=0;
21         end_if;
22         if palet_elev_front AND pal_running then
23             pas4_5:=0;
24             palet_elev_limit:=1;
25             palet_chain:=0;
26             palet_elev_up:=1;
27             stav_pal:=2;
28         end_if;
29     2:
30         emitor1:=1;
31         if palet_push_lim AND NOT palet_ele_mov AND pal_running then
32             belt:=1;
33             palet_belt:=1;
34         end_if;
35
36         rise.InputBit:=retro1;
37         rise.EnableIn:=1;
38         if rise.OutputBit then
39             box:=box+1;
40             stav_pal:=3;
41         end_if;
42     3:
43         rise.EnableIn:=0;
44         if box<7 AND box<>3 AND box<>5 then
45             stav_pal:=2;
46         elsif box=3 OR box=5 OR box=7 then
47             stav_pal:=4;
48         elsif box>7 AND box<>11 AND box<>14 then
49             stav_pal:=2;
50         elsif box=11 OR box=14 then
51             stav_pal:=4;
52         end_if;
53     4:
54         if pal_running then
55             belt:=0;
56             palet_belt:=0;
57             palet_push:=1;
58             palet_elev_limit:=0;
59             palet_elev_up:=0;
```



```
60     end_if;
61
62     if palet_push AND NOT palet_push_lim then
63         count:=count+1;
64         stav_pal:=5;
65     end_if;
66 5:
67     if count=3 AND pal_running then
68         stav_pal:=6;
69     elsif count=5 AND pal_running then
70         stav_pal:=6;
71     end_if;
72
73     if palet_push_lim AND pal_running then
74         palet_push:=0;
75     end_if;
76
77     if NOT palet_push AND NOT palet_push_lim then
78         stav_pal:=2;
79     end_if;
80 6:
81     if palet_push_lim AND pal_running then
82         palet_clamp:=1;
83     end_if;
84
85     if palet_push_lim AND palet_clamped AND pal_running then
86         palet_open:=1;
87     end_if;
88
89     if NOT palet_PL then
90         stav_pal:=7;
91     end_if;
92 7:
93     if palet_PL AND pal_running then
94         palet_elev_down:=1;
95         palet_clamp:=0;
96         palet_push:=0;
97     end_if;
98
99     if palet_ele_mov then
100         stav_pal:=8;
101     end_if;
102
103 8:
104     if NOT palet_ele_mov AND pal_running then
105         palet_elev_down:=0;
106         palet_open:=0;
107     end_if;
108
109     if NOT palet_open AND NOT palet_PL then
110         stav_pal:=9;
111     end_if;
112 9:
113     row:=row+1;
114     if row<2 then
115         stav_pal:=10;
116     else
117         stav_pal:=11;
118     end_if;
```

```
119     10:
120         if palet_PL AND pal_running then
121             palet_turn:=1;
122             palet_beltm:=1;
123
124             fall.InputBit:=retro1;
125             fall.EnableIn:=1;
126
127             if fall.OutputBit then
128                 palet_beltm:=0;
129                 stav_pal:=2;
130             end_if;
131         end_if;
132     11:
133         if palet_PL AND pal_running then
134             box:=0;
135             count:=0;
136             row:=0;
137             palet_turn:=0;
138             emitor2:=0;
139             palet_elev_limit:=1;
140             palet_elev_down:=1;
141             if palet_ele_mov then
142                 stav_pal:=12;
143             end_if;
144         end_if;
145     12:
146         if NOT palet_ele_mov AND pal_running then
147             palet_elev_limit:=0;
148             palet_elev_down:=0;
149             palet_chain:=1;
150         end_if;
151
152         if NOT palet_elev_front then
153             stav_pal:=1;
154         end_if;
155
156 end_case;
157
```

```
1
2 case stav_pal_vyk of
3   0:
4     pozicia_x1:=0;
5     pozicia_y1:=0;
6     stacker0_targ:=0;
7     hold1:=0;
8     holding1:=0;
9     if running then
10      stav_pal_vyk:=1;
11    end_if;
12
13   1:
14     if ENTER1 then
15       if sklad1P[pozicia_x1,pozicia_y1]=1 AND NOT vlavo1_vyk then
16         hold1:=1;
17       elsif sklad1L[pozicia_x1,pozicia_y1]=1 AND vlavo1_vyk then
18         hold1:=1;
19       else
20         pozicia_x1:=0;
21         pozicia_y1:=0;
22         hold1:=0;
23       end_if;
24     end_if;
25
26     if holding1 then
27       stav_pal_vyk:=2;
28     end_if;
29
30   2:
31     if stacker1_targ = 0 AND NOT stop_sim then
32       stacker1_targ:=(pozicia_x1+1)+(pozicia_y1*9);
33     end_if;
34
35     if stacker1_MX then
36       stav_pal_vyk:=3;
37     end_if;
38
39   3:
40     if NOT stacker1_MX AND NOT stacker1_MZ AND NOT vlavo1_vyk AND NOT stop_sim then
41       stacker1_R:=1;
42     elsif NOT stacker1_MX AND NOT stacker1_MZ AND vlavo1_vyk AND NOT stop_sim then
43       stacker1_L:=1;
44     end_if;
45
46     if stacker1_LL OR stacker1_RL then
47       stacker1_lift:=1;
48     end_if;
49
50     if stacker1_LL OR stacker1_RL AND stacker1_MZ then
51       stav_pal_vyk:=4;
52     end_if;
53
54   4:
55     if NOT stacker1_MZ AND NOT stop_sim then
56       stav_pal_vyk:=5;
57     end_if;
58
59   5:
60     if NOT vlavo1_vyk AND NOT stop_sim then
61       stacker1_R:=0;
62     elsif vlavo1_vyk then
```

```
60         stacker1_L:=0;
61         stav_pal_vyk:=6;
62     end_if;
63     6:
64     if stacker1_M AND NOT stop_sim then
65         stacker1_targ:=21474;
66     end_if;
67
68     if stacker1_MX then
69         stav_pal_vyk:=7;
70     end_if;
71     7:
72     if NOT stacker1_MX AND NOT stop_sim then
73         vyloz:=1;
74
75         stacker1_targ:=0;
76         stacker1_L:=1;
77         stav_pal_vyk:=8;
78     end_if;
79     8:
80     if stacker1_LL AND NOT stop_sim then
81         stacker1_lift:=0;
82     end_if;
83
84     if stacker1_LL AND stacker1_MZ then
85         stav_pal_vyk:=9;
86     end_if;
87     9:
88     if NOT stacker1_MZ then
89         stacker1_L:=0;
90         stav_pal_vyk:=0;
91     end_if;
92     stacker0_targ:=21474;
93
94 end_case;
```

```
1
2 case stav_pal_zak of
3
4   0:
5     if running then
6       stacker1_targ:=0;
7       stav_pal_zak:=1;
8     end_if;
9
10  1:
11   if NOT stop_sim then
12     pas4_2:=1;
13     pas4CCW:=1;
14     load2:=1;
15
16     if dif_sen1 then
17       load2:=0;
18       pas4_2:=0;
19       pas4CCW:=0;
20       stacker1_R:=1;
21       stav_pal_zak:=2;
22     end_if;
23   end_if;
24
25   if hold1 then
26     load2:=0;
27     pas4_2:=0;
28     pas4CCW:=0;
29     stav_pal_zak:=20;
30   end_if;
31
32  2:
33   load2:=0;
34   if stacker1_RL AND NOT stop_sim then
35     stacker1_lift:=1;
36   end_if;
37
38   if stacker1_RL AND stacker1_MZ then
39     stav_pal_zak:=3;
40   end_if;
41  3:
42   if NOT stacker1_MZ AND NOT stop_sim then
43     stacker1_R:=0;
44     stav_pal_zak:=4;
45   end_if;
46  4:
47   if stacker1_M AND NOT stop_sim then
48     stacker1_targ:=pozicia1;
49   end_if;
50
51   if pozicia1=55 then
52     x1:=0;
53     y1:=0;
54   end_if;
55
56   if stacker1_targ>54 then
57     stacker1_targ:= stacker1_targ - 54;
58     vlav01:=1;
59   end_if;
```

```
60
61     if stacker1_MX then
62         stav_pal_zak:=5;
63     end_if;
64 5:
65     if NOT stacker1_MX AND NOT stacker1_MZ AND NOT vlavo1 AND NOT stop_sim then
66         stacker1_R:=1;
67         //
68         if x1<9 then
69             sklad1P[x1,y1]:=1;
70         else
71             x1:=0;
72             y1:=y1+1;
73             sklad0P[x1,y1]:=1;
74         end_if;
75         //
76     elsif NOT stacker1_MX AND NOT stacker1_MZ AND vlavo1 AND NOT stop_sim then
77         stacker1_L:=1;
78         //
79         if x1<9 then
80             sklad1L[x1,y1]:=1;
81         else
82             x1:=0;
83             y1:=y1+1;
84             sklad0P[x1,y1]:=1;
85         end_if;
86         //
87     end_if;
88
89     if stacker1_LL OR stacker1_RL then
90         stacker1_lift:=0;
91     end_if;
92
93     if (stacker1_LL OR stacker1_RL) AND stacker1_MZ then
94         stav_pal_zak:=6;
95     end_if;
96
97 6:
98     if NOT stacker1_MZ AND NOT stop_sim then
99         stav_pal_zak:=7;
100    end_if;
101 7:
102     if stacker1_RL AND NOT stop_sim then
103         stacker1_R:=0;
104         stav_pal_zak:=8;
105     elsif stacker1_LL AND NOT stop_sim then
106         stacker1_L:=0;
107         stav_pal_zak:=8;
108     end_if;
109 8:
110     if stacker1_M AND NOT stop_sim then
111         stacker1_targ:=21474;
112     end_if;
113
114     if stacker1_MX then
115         stav_pal_zak:=9;
116     end_if;
117 9:
118     if NOT stacker1_MX then
```



```
119
120     stacker1_targ:=0;
121     pozicia1:=pozicia1+1;
122     x1:=x1+1;
123     stav_pal_zak:=1;
124 end_if;
125 20:
126     holding1:=1;
127     if NOT hold1 then
128         holding1:=0;
129         stav_pal_zak:=1;
130     end_if;
131
132 end_case;
```

```
1
2 if run AND NOT stop_sim then
3     running:=1;
4 end_if;
5
6 if NOT run OR NOT safety_door_OK then
7     //stop generovania paliet
8     if stav_gen=1 then
9         pas2_0:=0;
10        pas2_1:=0;
11        pas0CW:=0;
12        load0:=0;
13        stop_sim:=1;
14    else
15        stop_sim:=1;
16    end_if;
17    //stop paletizer zakladanie
18    if stav_pal_zak=1 then
19        pas4_2:=0;
20        pas4CCW:=0;
21        load2:=0;
22        stop_sim:=1;
23    else
24        stop_sim:=1;
25    end_if;
26
27    //stop vykladania
28    if stav_zak=2 then
29        load1:=0;
30        pas2CW:=0;
31        pas3CCW:=0;
32        pas2_2:=0;
33        pas4_0:=0;
34        pas4_1:=0;
35        pas4_4:=0;
36        pas6_0:=0;
37        turn_roll:=0;
38        remover0:=0;
39        //vzkladanie z druheho skaldu
40        load3:=0;
41        pas5CW:=0;
42        pas4_3:=0;
43        pas6_1:=0;
44        stop_sim:=1;
45    else
46        stop_sim:=1;
47    end_if;
48 end_if;
49
50 if run AND safety_reset then
51     stop_sim:=0;
52 end_if;
53
54 if start_but AND stop_but AND Emergency_stop then
55     pal_running:=1;
56 else
57     pal_running:=0;
58 end_if;
59
```

```
60 if NOT pal_running AND stav_pal=1 then
61     pas4_5:=0;
62     palet_chain:=0;
63 elsif NOT pal_running AND stav_pal=2 then
64     palet_belt:=0;
65     belt:=0;
66 end_if;
67
68 if reset then
69     running:=0;
70     pal_running:=0;
71     //reset generovania paliet
72     pas2_0:=0;
73     pas2_1:=0;
74     pas0CW:=0;
75     pas1CCW:=0;
76     load0:=0;
77     stacker0_R:=0;
78     stacker0_L:=0;
79     stacker0_lift:=0;
80
81     stacker0_targ:=21474;
82     pozicia0:=1;
83     zapln:=0;
84     x:=0;
85     y:=0;
86     COP(pole_nul[0,0],sklad0P[0,0],54);
87     COP(pole_nul[0,0],sklad0L[0,0],54);
88     vlavo0:=0;
89     holding:=0;
90     stav_gen:=0;
91     //reset paletizer
92     emitor1:=0;
93     emitor2:=0;
94     belt:=0;
95     palet_belt:=0;
96     palet_beltm:=0;
97     pas4_5:=0;
98     palet_chain:=0;
99     palet_open:=0;
100    palet_elev_up:=0;
101    palet_elev_down:=0;
102    palet_elev_limit:=0;
103    palet_clamp:=0;
104    palet_push:=0;
105    palet_open:=0;
106    palet_turn:=0;
107    rise.EnableIn:=0;
108    palet_elev_front:=0;
109    box:=-1;
110    count:=0;
111    row:=0;
112    stav_pal:=0;
113    //reset paletizer vykladanie
114    vyloz:=0;
115    pozicia_x1:=0;
116    vlavo1_vyk:=0;
117    hold1:=0;
118    COP(pole_nul[0,0],sklad1P[0,0],54);
```

```
119     COP(pole_nul[0,0],sklad1L[0,0],54);
120     stav_pal_vyk:=0;
121     //reset paletizer zakladanie
122     pas4_2:=0;
123     pas4CCW:=0;
124     load2:=0;
125     stacker1_R:=0;
126     stacker1_L:=0;
127     stacker1_lift:=0;
128
129     stacker1_targ:=21474;
130     pozicial:=1;
131     x1:=0;
132     y1:=0;
133     vlavo1:=0;
134     holding1:=0;
135     stav_pal_zak:=0;
136     // reset vykladania paliet
137     vyloz:=0;
138     pozicia_x:=0;
139     vlavo0_vyk:=0;
140     hold:=0;
141     stav_vyk:=0;
142     //reset zakoncenie vzkladania
143     load1:=0;
144     pas2CW:=0;
145     pas3CCW:=0;
146     pas4_0:=0;
147     pas4_1:=0;
148     pas4_4:=0;
149     pas6_0:=0;
150     turn_roll:=0;
151     turn_turn:=0;
152     turn_rollm:=0;
153     stav_zak:=0;
154 end_if;
155
156 //pre visu
157 if stacker0_MZ OR stacker0_MX then
158     S0_moving:=1;
159 else
160     S0_moving:=0;
161 end_if;
162
163 if stacker1_MZ OR stacker1_MX then
164     S1_moving:=1;
165 else
166     S1_moving:=0;
167 end_if;
168
```

```
1
2 case stav_vyk of
3   0:
4     pozicia_x:=0;
5     pozicia_y:=0;
6     stacker0_targ:=0;
7     hold:=0;
8     holding:=0;
9     ENTER:=0;
10    if running then
11      stav_vyk:=1;
12    end_if;
13
14  1:
15    if ENTER then
16
17      if sklad0P[pozicia_x,pozicia_y]=1 AND NOT vlavo0_vyk then
18        hold:=1;
19      elsif sklad0L[pozicia_x,pozicia_y]=1 AND vlavo0_vyk then
20        hold:=1;
21      else
22        pozicia_x:=0;
23        pozicia_y:=0;
24        stav_vyk:=0;
25      end_if;
26    end_if;
27
28    if holding then
29      stav_vyk:=2;
30    end_if;
31  2:
32    if stacker0_targ = 0 AND NOT stop_sim then
33      stacker0_targ:=(pozicia_x+1)+(pozicia_y*9);
34    end_if;
35
36    if stacker0_MX then
37      stav_vyk:=3;
38    end_if;
39  3:
40    if NOT stacker0_MX AND NOT stacker0_MZ AND NOT vlavo0_vyk AND NOT stop_sim then
41      sklad0P[pozicia_x,pozicia_y]:=0;
42      stacker0_R:=1;
43    elsif NOT stacker0_MX AND NOT stacker0_MZ AND vlavo0_vyk AND NOT stop_sim then
44      sklad0L[pozicia_x,pozicia_y]:=0;
45      stacker0_L:=1;
46    end_if;
47
48    if stacker0_LL OR stacker0_RL then
49      stacker0_lift:=1;
50
51    end_if;
52
53    if stacker0_LL OR stacker0_RL AND stacker0_MZ then
54      stav_vyk:=4;
55    end_if;
56  4:
57    if NOT stacker0_MZ AND NOT stop_sim then
58      stav_vyk:=5;
59    end_if;
```

```
60     5:
61     if NOT vlavo0_vyk AND NOT stop_sim then
62         stacker0_R:=0;
63         stav_vyk:=6;
64     elsif vlavo0_vyk AND NOT stop_sim then
65         stacker0_L:=0;
66         stav_vyk:=6;
67     end_if;
68     6:
69     if stacker0_M AND NOT stop_sim then
70         stacker0_targ:=21474;
71     end_if;
72
73     if stacker0_MX then
74         stav_vyk:=7;
75     end_if;
76     7:
77     if NOT stacker0_MX AND NOT stop_sim then
78         vyloz:=1;
79
80         stacker0_targ:=0;
81         stacker0_L:=1;
82         stav_vyk:=8;
83     end_if;
84     8:
85     if stacker0_LL AND NOT stop_sim then
86         stacker0_lift:=0;
87     end_if;
88
89     if stacker0_LL AND stacker0_MZ then
90         stav_vyk:=9;
91     end_if;
92     9:
93     if NOT stacker0_MZ then
94         stacker0_L:=0;
95         stav_vyk:=0;
96     end_if;
97
98 end_case;
```



```
1
2 case stav_zak of
3   0:
4
5     if running then
6       stav_zak:=1;
7     end_if;
8
9   1:
10    if vyloz then
11      stav_zak:=2;
12    end_if;
13  2:
14    if NOT stop_sim then
15      //vykladanie prvy sklad
16      load1:=1;
17      pas2CW:=1;
18      pas3CCW:=1;
19      pas2_2:=1;
20      pas4_0:=1;
21      pas4_1:=1;
22      pas4_4:=1;
23      pas6_0:=1;
24      turn_roll:=1;
25      remover0:=1;
26      //vzkladanie z druheho skaldu
27      load3:=1;
28      pas5CW:=1;
29      pas4_3:=1;
30      pas6_1:=1;
31
32      if dif_sen3 then
33        load3:=0;
34        pas5CW:=0;
35        pas4_3:=0;
36        pas6_1:=0;
37        stav_zak:=3;
38      end_if;
39    end_if;
40  3:
41    if dif_sen3 AND NOT dif_sen2 AND NOT dif_sen4 AND NOT stop_sim then
42      pas2CW:=0;
43      pas3CCW:=0;
44      pas4_0:=0;
45      pas4_1:=0;
46      pas6_0:=0;
47      turn_roll:=0;
48      turn_turn:=1;
49      turn_rollm:=1;
50      stav_zak:=4;
51    end_if;
52  4:
53    if turn_L90 AND NOT stop_sim then
54      pas4_3:=1;
55    end_if;
56
57    if turn_back then
58      turn_rollm:=0;
59      pas4_3:=0;
```

```
60         turn_turn:=0;  
61     end_if;  
62  
63     if turn_L0 then  
64         turn_roll:=1;  
65         stav_zak:=2;  
66     end_if;  
67  
68 end_case;
```