

Řízení virtuální výrobní linky za využití Factory I/O a TIA Portal

Ing. David Kundera

Bakalářská 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 BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Ing. David Kundera**
Osobní číslo: **A21194**
Studijní program: **B0714A150006 Aplikovaná informatika v průmyslové automatizaci**
Specializace: **Průmyslová automatizace**
Forma studia: **Kombinovaná**
Téma práce: **Řízení virtuální výrobní linky za využití Factory I/O a TIA Portal**
Téma práce anglicky: **Control of a Virtual Production Line Using Factory I/O and TIA Portal**

Zásady pro vypracování

1. Vypracujte literární rešerši na téma programovatelné automaty a jejich použití s 3D simulátorem Factory I/O.
2. Vyberte vhodné komponenty (senzory, akční členy, dopravníky, stanice, objekty, PLC) pro návrh virtuální výrobní linky.
3. Navrhněte a zrealizujte virtuální výrobní linku demonstrující výrobní operace, transport materiálu a jeho skladování.
4. Vytvořte programové vybavení pro řídicí PLC Siemens v TIA Portal.
5. Simulačně ověřte funkci navržené výrobní linky a vyhodnoťte dosažené výsledky.

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

Seznam doporučené literatury:

1. MARTINÁSKOVÁ, Marie a ŠMEJKAL, Ladislav. *Řízení programovatelnými automaty*. Praha: ČVUT, 1998. ISBN 8001017664.
2. PETRUZELLA, Frank D. *Programmable Logic Controllers*. 4th ed. New York: McGraw-Hill Companies, 2010. ISBN 978-0-07-351088-0.
3. REAL GAMES. *FACTORY I/O Manual* [online]. 2006-2023 [citováno 2023-11-10]. Dostupné z: <https://docs.factoryio.com>.
4. SIEMENS. *Programming Guideline for S7-1200/1500* [online]. V1.6. Siemens AG, 2018 [citováno 2023-11-10]. Dostupné z: <https://support.industry.siemens.com/cs/document/90885040/programming-guideline-for-s7-1200-s7-1500?dti=0&lc=en-CZ>.
5. ŠMEJKAL, Ladislav a MARTINÁSKOVÁ, Marie. *PLC a automatizace. 1. díl, Základní pojmy, úvod do programování*. Praha: BEN – technická literatura, 1999. ISBN 8086056589.

Vedoucí bakalářské práce: **Ing. Petr Dostálek, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **8. prosince 2023**

Termín odevzdání bakalářské 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 bakalářské 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 bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou 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 – bakalářskou 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 bakalářské 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 bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské 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 bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

David Kundera, v. r.
podpis studenta

ABSTRAKT

Bakalářská práce se zabývá návrhem virtuální výrobní linky demonstrující výrobní operace (obrábění, montáž, balení), transport materiálu a jeho skladování za využití programového vybavení Factory I/O a TIA Portal. V teoretické části je zpracována rešerše na téma programovatelné automaty a jejich použití s 3D simulátorem Factory I/O. Součástí rešerše je popis prostředí Factory I/O a TIA Portal, návod na vytvoření projektu v TIA Portal a samotné spuštění simulace.

V praktické části byly vybrány scény Production Line, Assembler, Automated Warehouse a komponenty linky v programu Factory I/O, následně je vybráno CPU 1513-1 PN a jsou popsány nejdůležitější body logiky programu. Na závěr je simulačně ověřeno, že dle navrženého programu dojde k naskladnění na všechny skladové pozice bez kolize před i po samotném naskladnění.

Klíčová slova: PLC, TIA Portal, SIMATIC S7-1500, Factory I/O, virtuální výrobní linka

ABSTRACT

The Bachelor thesis is focused on creation of virtual production line demonstrating production operations (machining, assembly, packaging), material transport and storage using Factory I/O and TIA Portal. The theoretical part focuses on PLC and their use with 3D Factory I/O simulator. Part of the research is a description of Factory I/O environment and TIA Portal, instructions for creating a project in TIA Portal and starting simulation itself.

In the practical part, Production Line, Assembler, Automated Warehouse and line components are first selected in Factory I/O, then CPU 1513-1 PN is selected and the most important parts of the program are described. Finally, it is verified by simulation that according to the proposed program, all warehouse positions will be stocked without collision before and after the actual stocking.

Keywords: PLC, TIA Portal, SIMATIC S7-1500, Factory I/O, virtual production line

Na tomto místě bych rád poděkoval Ing. Petru Dostálkovi, Ph.D. za odborné vedení, cenné rady a čas strávený nad touto prací.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PROGRAMOVATELNÉ AUTOMATY	11
1.1 CHARAKTERISTIKA PLC	11
1.2 STRUKTURA PLC	12
1.2.1 Napájení	13
1.2.2 CPU	13
1.2.3 Komunikační jednotky	14
1.2.4 Digitální a analogové vstupy/výstupy	14
1.3 TYPY PLC	15
1.3.1 Mikro PLC	15
1.3.2 Kompaktní PLC	15
1.3.3 Modulární PLC.....	16
1.3.4 Mini, Piko/Nano PLC.....	16
1.4 PRINCIP FUNKCE.....	16
1.5 PROGRAMOVACÍ JAZYKY PRO PLC.....	19
1.5.1 Žebříčkový diagram	19
1.5.2 Diagram funkčních bloků.....	20
1.5.3 Strukturovaný text.....	22
1.5.4 Seznam instrukcí	22
1.5.5 Sekvenční funkční diagram.....	22
2 PLC SIEMENS	24
2.1 SIMATIC S7-1500	24
2.2 TIA PORTAL.....	25
2.2.1 Programovací bloky	27
2.2.2 Datové typy	31
2.3 PROJEKT V TIA PORTAL.....	33
2.3.1 Vytvoření projektu	35
2.3.2 HW konfigurace	35
2.3.3 Vložení programového bloku.....	37
2.3.4 Kompilace projektu	41
2.3.5 Nahrání projektu do PLCSIM	42
2.3.6 Online kontrola funkčnosti.....	45
3 FACTORY I/O	46
3.1 CHARAKTERISTIKA SOFTWARE	46
3.2 PROSTŘEDÍ	47
3.3 NABÍZENÉ EDICE	50
3.4 ALTERNATIVY	55
3.4.1 EasyPLC.....	55
II PRAKTICKÁ ČÁST	58
4 NÁVRH VIRTUÁLNÍ VÝROBNÍ LINKY VE FACTORY I/O	59

4.1	KONFIGURACE.....	60
4.2	ROZDĚLENÍ PRACOVÍŠŤ A SOUČÁSTÍ VÝROBNÍ LINKY	61
4.2.1	CNC.....	61
4.2.2	Dopravníky.....	63
4.2.3	Montáž (Pick & place 1)	64
4.2.4	Balení (Pick & place 2)	65
4.2.5	Sklad.....	66
5	PROGRAMOVÁNÍ V TIA PORTAL.....	68
5.1	CHARAKTERISTIKA PROGRAMU	69
5.2	VYTVOŘENÍ PROJEKTU	69
5.3	ZVOLENÍ HW	69
5.4	POPIS PROGRAMU PRO JEDNOTLIVÁ PRACOVÍŠŤE	70
5.4.1	CNC.....	70
5.4.2	Dopravníky.....	71
5.4.3	Montáž (Pick & place 1)	76
5.4.4	Balení (Pick & place 2)	82
5.4.5	Sklad.....	83
6	SIMULACE	86
6.1	KOMPILACE PROGRAMU A NAHRÁNÍ DO S7-PLCSIM ADVANCED 5.0	86
6.2	TESTOVÁNÍ PROGRAMU	86
6.2.1	Naskladnění na všech 54 skladových pozic	87
6.2.2	Bezkolizní stav	89
	ZÁVĚR	93
	SEZNAM POUŽITÉ LITERATURY.....	95
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	99
	SEZNAM OBRÁZKŮ	100
	SEZNAM TABULEK.....	103
	SEZNAM PŘÍLOH.....	104

ÚVOD

Vzhledem k tomu, že je v dnešní době stále nutné zeštíhlovat výrobní procesy a snižovat náklady s výrobou spojené, je pro firmy důležité automatizovat procesy. K automatizaci procesů jsou potřeba programátoři, kteří jsou schopni vytvořit řídicí program dle požadavků zákazníků. Tato bakalářská práce vznikla za tím účelem, aby ukázala jednu z možností, jak mohou studenti případně zaměstnanci, kteří běžně s automatizací do styku nepřichází, začít s tvorbou programů pro PLC z pohodlí domova, což jim umožňuje virtuální simulátor PLC, který v této práci nahradí úlohu fyzického PLC. Dalším přínosem využití virtuálního simulátoru PLC může být snížení nákladů na vývoj a předcházení havarijních stavů v reálném prostředí.

Teoretická část bakalářské práce je věnována programovatelným automatům, PLC Siemens, TIA Portalu, Factory I/O a alternativnímu softwaru EasyPLC. V teoretické části je popsáno prostředí prvních dvou zmíněných programů. Dále je v teoretické části popsán postup, jak vytvořit projekt v TIA Portal, kde zvolit hardware, jak vložit programový blok a spustit svůj první virtuální projekt, který bude spuštěn na simulátoru PLC.

V praktické části jsou definovány cíle řízení virtuální linky, které vychází z bodů zadání bakalářské práce. Následně je popsán výběr scén a komponent pro vytvoření virtuální výrobní linky, které jsou v práci použity. Při návrhu virtuální linky bylo postupováno dle zadání bakalářské práce, tzn. že se linka skládá z pracoviště CNC, montáže, balení, dopravníků a skladu. Virtuální výrobní linka je navržena ve Factory I/O v2.5.2 – Ultimate Edition. Další část praktické části je věnována popisu programového vybavení, které bylo tvořeno v TIA Portal V18, přičemž jsou vždy popsány nejdůležitější části programu. Praktická část je doplněna ilustrativními obrázky, aby bylo zřejmé, jak jednotlivé části linky včetně samotné linky vypadají. Závěr praktické části je věnován testování programu a vyhodnocení dosažených cílů.

I. TEORETICKÁ ČÁST

1 PROGRAMOVATELNÉ AUTOMATY

První kapitola bakalářské práce se zabývá charakteristikou programovatelných automatů, jejich strukturou, funkcí, typy PLC, se kterými je možné se v praxi setkat a jazyky, které se používají pro jejich programování.

1.1 Charakteristika PLC

Dle Šmejkal a Martináskové je PLC (Programmable Logic Controller) uživatelsky programovatelný řídicí systém, který je přizpůsoben pro řízení průmyslových a technologických procesů velmi často specializovaný na logické úlohy. [1]

Petruzella uvádí, že PLC je digitální počítač, který slouží k řízení strojů, který byl oproti osobnímu počítači navržen tak, aby dokázal fungovat ve výrobním prostředí, přičemž je vybaven speciálními vstupy/výstupy a řídicím programovacím jazykem. Aktuálně jsou PLC nejpoužívanější řídicí technologií používanou v průmyslovém prostředí. PLC je navrženo pro připojení více vstupů/výstupů, je odolné vůči teplotě, rušení z elektrické rozvodné sítě, vibracím a nárazům. [4]

PLC jsou používány ve výrobních podnicích k řízení montážních linek, dopravníků, svařovacích robotů, robotů pro lakování nebo balení. V potravinářském a nápojovém průmyslu jde pak o řízení procesu míchání materiálu, vaření a naplňování. Za účelem dosažení odpovídající kvality produktu, lze skrze PLC monitorovat a řídit teplotu, tlak a další technologické veličiny. Dále jsou PLC využívány např. ve farmaceutickém průmyslu (míchání a balení léků) a energetickém průmyslu (systémy výroby a distribuce energie jako jsou turbíny, generátory a transformátory). [19]

PLC nahradilo původně používanou reléovou logiku používanou k řešení úloh logického řízení. V dnešní době však obstarávají úlohy regulačního typu, úlohy zaměřené na monitorování procesu a analogová měření. [2]

Výhodou PLC oproti mechanickým technologiím, které nahradily, je to, že zabírají méně místa, provádějí složitější úkoly a jsou více přizpůsobitelné. Mezi další výhody patří nepřetržitý provoz bez údržby a digitalizace výroby. První PLC vyrobené pro General Motors Dickem Morleyem v roce 1969 fungovalo bez přerušení 20 let. [20]

PLC je průmyslový počítač, který nabízí vysoký stupeň spolehlivosti s možností vykonávání programu bez přerušení tj. 24/7/365. PLC jsou cenově dostupnější alternativou oproti reléovým systémům od dob, kdy se začaly stále častěji používat tranzistory. Na počátku byly PLC jednoduché a lehce nastavitelné. Se zvyšováním složitosti výroby v podnicích došlo k jejich vývoji, což vedlo k tomu, že nyní je k programování potřeba osoby, která ví, jak vytvořit, optimalizovat a modifikovat program dle potřeb zákazníků. [32]

Šmejkal a Martinásková jako výhody PLC uvádí rychlost realizace systému, což znamená, že uživatel nemusí vyvíjet technické vybavení, ale jen si navolí vhodnou sestavu modulů PLC, vytvoří projekt, napíše a odladí program. Mezi další výhody řadí spolehlivost, odolnost, diagnostiku, snadnou přizpůsobitelnost změnám zadání a schopnost komunikace s různými systémy a zařízeními v podřízené úrovni v tzv. poli, do kterého spadají senzory, měřicí zařízení, akční členy. Dále je PLC schopné komunikovat se systémy a zařízeními na souřadné úrovni (ostatní programovatelné automaty nebo jiné řídicí systémy) a se systémy nadřízenými. [2]

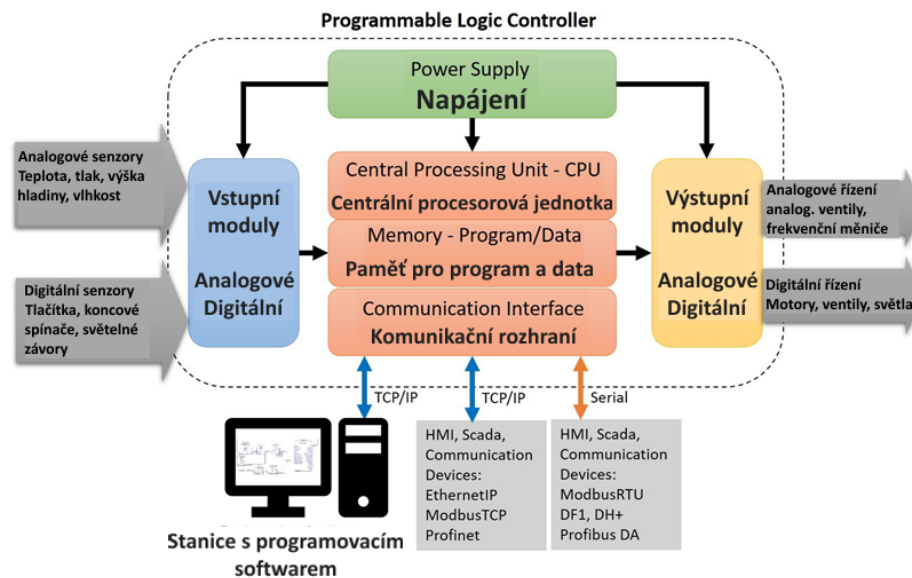
PLC jsou oblíbenou volbou pro průmyslovou automatizaci a řídicí aplikace, protože v mnoha ohledech překonávají tradiční řídicí systémy. Mezi výhody PLC patří snadná programovatelnost a modifikace programu, spolehlivost, škálovatelnost (možnosti rozšíření a aktualizace reagující na měnící se požadavky), bezpečnost, vzdálený přístup a záznam dat. [19]

1.2 Struktura PLC

Části, ze kterých se standardně PLC skládá jsou centrální procesorová jednotka, systémová paměť, uživatelská paměť, vstupní a výstupní jednotky, které slouží k připojení řízeného systému a soubor komunikačních jednotek, který slouží ke komunikaci se souřadnými i nadřazenými řídicími systémy. Jednotky jsou mezi sebou propojeny systémovou sběrnici sloužící ke komunikaci mezi nimi. Uživatelský program může být napsán ve vícero programovacích jazycích. Po jeho překladu je program uložen v uživatelské paměti. Program se skládá z posloupnosti instrukcí, která je procesorem vykonávána cyklicky. [2]

Šmejkal a Martinásková do konstrukčního a elektrického provedení programovatelných automatů zahrnují pouzdro, základní modul, napájení, CPU, binární vstupní a výstupní jednotky, kombinované jednotky binárních vstupů a výstupů, analogové vstupní a výstupní jednotky. [2]

Podle Petruzelly může typické PLC být rozděleno na centrální procesorovou jednotku (CPU), vstupy/výstupy, napájení a programovací zařízení. [4]



Obrázek 1. Základní schéma PLC [35]

1.2.1 Napájení

Napájení primárně slouží k zajišťování chodu PLC komponentů, přičemž většina PLC je napájeno buď 24 V stejnosměrného napětí nebo 240 V střídavého napětí. Některé z PLC mají napájecí modul zvlášť. Systémový napájecí zdroj slouží také k monitorování a regulaci dodávaného napětí, přičemž je schopen upozornit CPU, pokud je něco v nepořádku. Za účelem poskytování ochrany pro ostatní PLC komponenty je zdroj napájení schopen spolehlivé funkce s 10-15% změnami síťového napětí. [9]

1.2.2 CPU

Jádrem programovatelného automatu je centrální procesorová jednotka (CPU), která určuje jeho výkonnost. Centrální procesorová jednotka bývá jednocpu nebo víceprocesorová. [2]

CPU je hlavní částí PLC, která je označovaná za jeho „mozek“, protože řídí a zpracovává všechny jeho operace. CPU se skládá z aritmeticko-logické jednotky, paměti pro program, proces, vnitřních časovačů a čítačů. [9]

Operační rychlost CPU je posuzována podle tzv. doby cyklu. Doba cyklu je doba zpracování 1000 (1k) logických instrukcí, která se řádově pohybuje od desítek milisekund až k desítkám milisekund. Výrobci pro daný typ automatu nabízí výběr CPU lišících se rychlostí, velikostí paměťového prostoru a cenou. [2]

Centrální jednotka vykonává uživatelský program a systémové služby, dále se stará o komunikaci s vlastními i vzdálenými moduly, s nadřazeným systémem a programovacím přístrojem. CPU obsahuje mikroprocesor, mikrořadič nebo specializovaný řadič a paměťový prostor, který se dále dělí na:

- paměť pro uložení uživatelského programu, datových bloků a tabulek – nemění se při vykonávání programu,
- operační paměť (zápisník) – obsahuje uživatelské registry, čítače a časovače, obrazy vstupů a výstupů, komunikační, časové a další systémové proměnné (systémové registry) a její obsah se mění při vykonávání uživatelského a systémového programu.

[1]

Jako další dělení paměti, která tvoří paměťový prostor CPU, uvádí Šmejkal a Martinásková paměť uživatelskou, systémovou a paměť dat. V uživatelské paměti je uložen uživatelský program. Uživatelská paměť je většinou typu EEPROM o kapacitě desítek kB až jednotek MB u modulárních PLC. Kompaktní programovatelné automaty jsou vybaveny uživatelskou pamětí o velikosti desítek kB. Systémová paměť je typu FLASH a je v ní umístěn systémový program. Paměť dat je typu RAM a nachází se v ní uživatelské registry, zápisníkové registry, čítače, časovače, a často také vyrovnávací registry pro obrazy vstupů a výstupů. [2]

1.2.3 Komunikační jednotky

Komunikační moduly umožňují programovatelným automatům komunikovat se vzdálenými moduly vstupů/výstupů, se souřadnými a nadřazenými systémy, s podsystémy, s panely pro operátory označovanými zkratkou HMI (Human Machine Interface) příp. s jinými inteligentními přístroji, počítači a jejich sítěmi. Takováto spojení jsou označovaná za distribuovaný systém. [1]

1.2.4 Digitální a analogové vstupy/výstupy

Digitální vstupní jednotky nabývají dvou hodnot, co se výstupního signálu týče. Patří sem např. tlačítka, přepínače, koncové spínače, senzory doteku nebo přiblížení dvouhodnotové senzory tlaku, hladiny, teploty apod. Vstupní binární modul má na starost ochranu všech

konkrétních vstupů před potenciálním zničením od chybného napětí či přepětí, dále odfiltrování krátkodobých rušivých pulzů a signalizaci stavu vstupů skrze LED. [2]

Za účelem ochrany digitálních vstupů a výstupů je modul galvanicky oddělen od centrální jednotky. Toto oddělení slouží k potlačení rušivých signálů do systému ze vstupních a výstupních svorek. [1]

K digitálním výstupním jednotkám připojujeme akční členy, jejichž vstupní signál může nabývat dvou hodnot. Příkladem výstupních digitálních jednotek jsou cívky relé, solenoidových ventilů, elektromagneticky ovládaných pneumatických nebo hydraulických rozvaděčů, elektromagnetických spojek, optická či akustická signalizační technika apod. [2]

Petruzella uvádí, že typické analogové vstupy/výstupy nabývají hodnot 0-20 mA, 4-20 mA nebo 0-10 V. O převod analogového signálu na digitální se stará A/D převodník, k opačnému převodu slouží D/A převodník. [4]

Analogové vstupní a výstupní jednotky pracují se spojitými veličinami. K analogovým vstupům lze připojit například snímače teploty, snímače vlhkosti, tlaku, síly, hladiny, rychlosti ale i většinu inteligentních přístrojů s analogovými výstupy, nebo třeba měřené napětí či výstup z potenciometru. [2]

1.3 Typy PLC

1.3.1 Mikro PLC

Jedná se o nejmenší a nejlevnější kompaktní PLC systémy s pevnou sestavou vstupů a výstupů, které jsou obvykle jen binární. Tyto PLC jsou díky své nízké ceně, malým rozměrům a kompaktním provedením spotřebním materiálem. Mikro PLC není možné dodatečně rozšiřovat. Nejčastěji jsou používány pro realizaci logické výbavy jednoduchých strojů a mechanismů, která byla dříve řešena pomocí relé, což přináší úsporu na dalších ovládacích prvcích, jakými jsou např. relé, stykače, časové relé a programátory. [1]

1.3.2 Kompaktní PLC

Kompaktní PLC oproti Mikro PLC nabízejí omezenou možnost nakonfigurování. K základnímu modulu je možné připojit jeden nebo několik přídatných modulů, které mají pevnou kombinaci vstupů a výstupů, jedná se např. o modul s 8 binárními vstupy a 8 binárními vý-

stupy (tranzistorovými nebo reléovými), modul rychlých čítačů, analogový vstupní nebo výstupní modul nebo modul regulátoru. Některé z kompaktních systémů nabízejí navíc vnitřní modulárnost. [1]

1.3.3 Modulární PLC

U modulárních PLC má každá součást nebo modul nezávislé karty, kterými jsou např. CPU karta, karta s analogovými vstupy, karta s digitálními výstupy, komunikační karta apod. Výhodou je možnost jednotlivé karty snadno vložit nebo vyjmout. [37]

Modulární PLC nabízí škálovatelný systém, díky kterému je možné pokrýt více požadavků na automatizaci. Vzhledem k tomu, že všechny komponenty jsou na sobě nezávislé, je nutné manuálně připojit napájecí zdroj, komunikační modul a další moduly k CPU. Modulární PLC jsou často montovány do racku (seskupení přístrojů), ve kterém je umístěn komunikační modul, díky kterému jsou všechna připojení centralizovaná. [40]

1.3.4 Mini, Piko/Nano PLC

Výrobci PLC navrhli menší PLC pro využití v méně rozsáhlých aplikacích. Tyto PLC je možné dělit dle velikosti na Mini, Piko, Nano a výše již zmíněné Mikro PLC. Mini PLC obvykle nabízí 128 a 512 vstupů/výstupů. Mikro PLC mají k dispozici 15 až 128 vstupů/výstupů a bývají používány ve velmi malých automatizačních nebo řídicích systémech např. zábavních atrakcích. Piko/Nano PLC jsou vybaveny méně než 15 vstupy/výstupy. Jsou velmi jednoduché a často jsou používány pro PLC trénink. [31]

1.4 Princip funkce

Princip, jakým pracuje programovatelný automat, se dělí do čtyř fází: čtení vstupů, vykonávání programu, zápis na výstupy a režie. PLC monitoruje vstupy z připojených zařízení. Stav z těchto vstupů následně vyhodnocuje dle programu, na základě vyhodnocení dojde k nastavení výstupů připojených zařízení. PLC pracují s digitálními a analogovými vstupy/výstupy, přičemž digitální vstup/výstup se chová jako standardní světelný spínač, který má dva stavy – vypnuto/zapnuto. Analogový vstup/výstup se chovají jako stmívač, který může nabývat jakéhokoliv stavu mezi vypnuto/zapnuto. PLC pracuje se dvěma typy vstupních dat, první z nich jsou generovány stroji nebo senzory, druhé jsou generovány lidským operátorem používajícím HMI nebo SCADA systém. Mezi vstupy, se kterými přichází do styku lidský

operátor, patří např. tlačítka, senzory z klávesnice, dotykové obrazovky, dálkového ovladače nebo čtečky karet. [18]

Dle Šmejkal a Martináskové je PLC program posloupnost instrukcí a příkazů jazyka. Zpravidla je PLC program aktivován cyklickým vykonáváním programové smyčky. Výhodou oproti jiným programovatelným systémům je to, že systémový program se po vykonání poslední instrukce uživatelského programu postará o návrat na začátek programu. Systém je schopen hlásit i dlouhodobé setrvání programu v programové smyčce, které je fatální chybou hlášenou systémem jako „překročení doby cyklu. [1]

PLC program je vykonáván cyklicky, prvně proběhne skenování stavu připojených vstupních zařízení. Poté dojde k vykonání samotného uživatelského programu, kdy PLC využije informace o stavu vstupních zařízení, na základě nichž dle uživatelského programu změní stav signálu u odpovídajících výstupních zařízení. V rámci fáze režie proběhne úklidový krok, kdy PLC provede interní diagnostiku bezpečnosti, aby ověřilo, že je vše v normálních provozních podmínkách, když je proces dokončen, dojde k restartu cyklu a PLC začne znovu kontrolou vstupů. [18]

Šmejkal a Martinásková pro PLC TECOMAT uvádí, že vždy po vykonání poslední instrukce uživatelského programu je předáno řízení systémovému programu, který provede tzv. otočku cyklu. V ní nejprve aktualizuje hodnoty výstupů a vstupů: hodnoty uložené dosud v paměti jako obrazy výstupů (registry Y) přepíše do registrů výstupních modulů a hodnoty ze vstupních modulů okopíruje do paměťových obrazů vstupů (registr X). Dále aktualizuje časové údaje pro časovače a systémové registry, ošetří komunikaci a provede ještě řadu režijních úkonů. Po otočce cyklu je opět předáno řízení první instrukci uživatelského programu (např. P0). [1]

Siemens uvádí, že v režimu RUN během skenovacího cyklu dojde k zapsání výstupů, načtení vstupů, vykonání uživatelského programu, aktualizaci komunikačních modulů, reakci na uživatelská přerušení a komunikační žádosti, vše zmíněné provádí CPU. Komunikační požadavky jsou zpracovávány pravidelně během skenování. Zmíněné akce jsou vyjma uživatelských přerušení obsluhovány pravidelně a v sekvenčním pořadí. Povolená uživatelská přerušení jsou obsluhována podle priority v pořadí, ve kterém k nim dochází. Přerušení mohou nastat kdykoliv během skenování. Při přerušení procesor načte vstupy, vykoná organizační blok a zapíše výstupy za použití přidruženého oddílu obrazu procesu, je-li to vhodné.

Pokud nedojde k dokončení cyklu skenování v maximální době cyklu, což zaručuje systém, dojde k vygenerování události časové chyby. [15]

Na začátku každého cyklu skenování dojde k načtení aktuálních hodnot digitálních a analogových výstupů z obrazu procesu, které jsou následně zapsány do fyzických výstupů CPU, systémového bloku a SM modulů (digitální nebo analogové vstupní, výstupní příp. vstupně/výstupní přídavné moduly), které jsou nakonfigurovány pro automatickou aktualizaci vstupů a výstupů. Při přístupu instrukce k fyzickému výstupu dojde k aktualizaci obrazu procesu výstupů a fyzických výstupů. Poté jsou načteny aktuální hodnoty digitálních a analogových vstupů z CPU, systémového bloku a SM modulů, následně dochází k zápisu těchto hodnot do obrazu procesu. Při přistoupení instrukce k fyzickému vstupu nedochází k aktualizaci obrazu procesu, fyzický vstup je však aktualizován. Nakonec dojde k vykonání instrukcí uživatelského programu od první až po poslední instrukci, což obnáší vykonání všech organizačních bloků programu a všech jejich přidružených funkcí a funkčních bloků. Organizační bloky se vykonávají dle jejich přiděleného čísla od nejnižšího po nejvyšší. [15]

Po vykonání STARTUP rutiny, která zahrnuje vyčištění obrazu procesu, inicializaci vstupů s poslední hodnotou, přerušení je vykonán organizační blok OB 100. Tento blok je vykonán procesorem před organizačním blokem OB 1. Dále jsou před OB 1 načteny stavy fyzických vstupů, které jsou uloženy v oblasti systémové paměti CPU a distribuovaných vstupech a výstupech. Této paměti se říká obraz procesu. Před spuštěním hlavního programu jsou tedy stavy fyzických vstupů přeneseny do obrazu procesu. Následně je procesorem vykonán hlavní program skrze OB 1. V tomto hlavním OB mohou být volány funkce a funkční bloky a můžeme skrze něj pracovat se systémovou pamětí pomocí paměťových příznaků, časovačů a čítačů. Funkce a funkční bloky, které nejsou volány hlavním organizačním blokem, nebudou provedeny, protože cyklus skenování je zaměřen pouze na něj a na to, co obsahuje. Po dokončení hlavního organizačního bloku jsou zkopírovány stavy výstupů do obrazu procesu. Po vykonání uživatelského programu je cyklus ukončen a stavy výstupů z obrazu procesu jsou zapsány na výstup fyzických modulů. [13]

Hlavní výhodou toho, že CPU přistupuje k obrazu procesu a ne přímo k fyzickým vstupním a výstupním modulům je ta, že CPU má konzistentní obraz procesu signálů během vykonání jednoho cyklu programu. Dojde-li ke změně stavu signálu na vstupním modulu při vykonávání programu, stav signálu v obrazu procesu je zachován, dokud nedojde k jeho aktualizaci v dalším cyklu. [13]

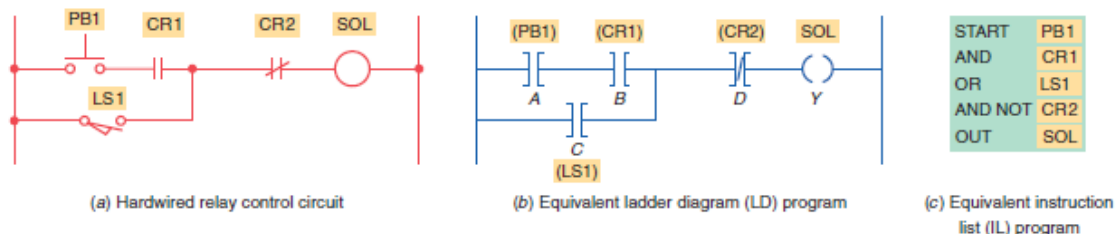
1.5 Programovací jazyky pro PLC

Mezinárodní elektrotechnická komise (IEC) 61131 udává 5 programovacích jazyků. Těmito jazyky jsou žebříčkový diagram (LD), diagram funkčních bloků (FBD), strukturovaný text (ST), seznam instrukcí (IL) a sekvenční funkční diagram (SFC). [20]

1.5.1 Žebříčkový diagram

Žebříčkový diagram je grafický programovací jazyk, který má podobnou formu, která je využívána pro kreslení schémat reléových a kontaktních prvků. Hlavní rozdíl je v symbolech pro kontakty a cívky, které jsou ze semigrafických důvodů zjednodušené. [1]

Dle Petruzelly se jedná o nejčastěji používaný jazyk. Tento jazyk napodobuje logiku relé a preferují ho programátoři, kteří upřednostňují zadávání řídicích pokynů skrze reléové kontakty, cívky a dalších funkcí jako jsou například instrukční bloky. Adresování vstupů a výstupů se liší výrobce od výrobce. Obr. č. 2 zobrazuje porovnání žebříčkového diagramu (LD) a seznamu instrukcí (IL). Na obrázku vlevo je znázorněn klasický obvod reléových kontaktů, uprostřed je pak ekvivalentní logický obvod naprogramovaný v žebříčkovém diagramu. Na obrázku vpravo je stejný obvod naprogramován v seznamu instrukcí, který se skládá z instrukcí odkazujících na základní funkce logických hradel AND, OR a NOT. [4]



Obrázek 2. Ekvivalent zapojení v reléové logice, LD a IL [4]

Mezi výhody žebříčkového diagramu patří:

- Jednoduchá implementace a hledání chyb – díky vizualizaci je patrný stav většiny proměnných.
- Modulární design – každá příčka je samostatnou podmínkou, kterou lze přidat nebo odebrat dle potřeby programátora.

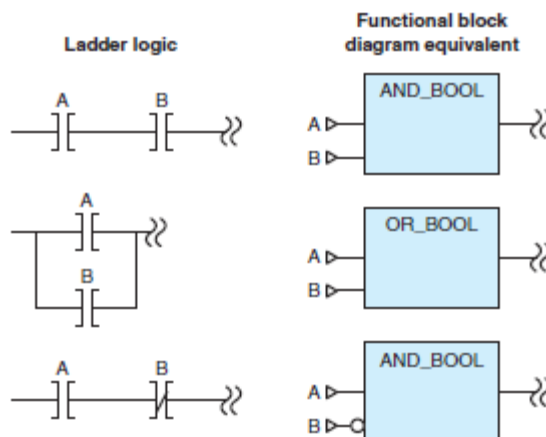
- Odolnost a soudržnost - možnost implementace široké škály funkcí, vysoká míra standardizace nenabízí uživateli vysokou flexibilitu, což vede ke konzistentnímu kódu mezi různými implementacemi.

Nevýhody pak jsou:

- Nízká intuitivita pro ty, kdo mají zkušenosti s programováním v C, C++, Javě nebo Pythonu a nemají elektrotechnické vzdělání.
- Pomalé nasazení - dáno vizuální logikou, kdy programátorovi trvá déle, než vytvoří potřebnou logiku, navíc programátora částečně zdržuje i přetahování prvků.
- Moderní teorie řízení – náročnost implementace pro PID, řízení toku, analogové senzory a zpětnovazební smyčky. [5]

1.5.2 Diagram funkčních bloků

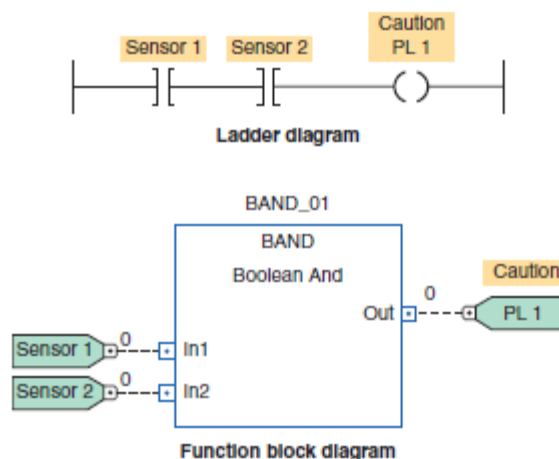
Při programování jazykem funkčních bloků jsou využity instrukce, které jsou naprogramovány jako vzájemně propojené bloky, což je pro programátora vizuálně na monitoru patrné. Mezi typické příklady patří logické funkce, časovače a čítače. Layout funkčních bloků je podobný elektrickým a elektronickým blokovým schémátům používaným pro zjednodušení složitějších systémů funkčních bloků. Diagram funkčních bloků stojí na datovém toku od vstupů přes funkční bloky nebo instrukce k výstupům. Tento programovací jazyk si získává svou oblibu, protože oproti jazyku žebříčkového diagramu a reléovým schémátům nabízí programátorovi grafické zobrazení, na jehož pozadí jsou již předdefinované algoritmy tzn. že programátor v podstatě jen doplní potřebné informace. [4]



Obrázek 3. Ekvivalent zapojení v žebříčkové logice a FBD [4]

Jazyk funkčních bloků je grafický jazyk, v němž jsou základní logické operace popisovány obdélníkovými značkami. Výška značky je přizpůsobována automaticky podle počtu vstupů. Své značky mají i ucelené funkční bloky mezi které patří např. čítače, časovače, posuvné registry, paměťové členy, ale i aritmetické a paralelní logické instrukce. Jazyk je intuitivní pro uživatele, kteří jsou zvyklí na kreslení logických schémat pro řízení s integrovanými obvody. [1]

Obrázek č. 4 zobrazuje příklad využití žebříčkového diagramu (nahore) a diagramu funkčních bloků (dole) se stejným logickým výstupem. V příkladu jsou dva senzory zapojené v sérii a cílem je rozsvítit výstražnou žárovku PL 1. K rozsvícení žárovky dojde, pokud jsou oba kontakty senzorů sepnuté. U žebříčkového diagramu leží oba vstupní senzory a výstražné světlo mezi příčkami. Funkční blok obsahuje logickou booleanovskou funkci AND se dvěma vstupy (Sensor 1 a 2) a jedním výstupem (PL 1). [4]



Obrázek 4. Ekvivalent zapojení LD a FBD [4]

Mezi výhody programování v diagramu funkčních bloků patří:

- uživatelská přívětivost a jednoduchost při vytváření libovolného rozložení,
- vhodnost použití pro komplexní projekty – oproti žebříčkovému diagramu, kde je nutné použít více příček, zde stačí stejného výsledku dosáhnout popsáním jedné stránky.

Flexibilita s sebou nese nevýhodu v tom, že je těžké jazyk standardizovat, aby měl každý programátor stejný přístup. Dále je snadné se ztratit ve všech těch listech při tvorbě komplexního programu. [5]

1.5.3 Strukturovaný text

Strukturovaný text se podobá programování v jazyce C nebo assembleru. Při programování jsou zadávány řádky kódu, které se vykonávají sekvenčně, vyhodnocují specifické funkce, logické výrazy a posílají signál na výstupy PLC. Přejít na tento jazyk je jednodušší pro uživatele, kteří mají zkušenosti s programováním např. v C, C++, Javě nebo Pythonu. [5]

Mezi výhody programování ve strukturovaném textu patří následující:

- intuitivita – spočívá v jednodušším přechodu z IT prostředí,
- komplexnost – oproti jiným programovacím jazykům nabízí větší flexibilitu,
- přenositelnost – díky standardizaci je možné jej přenášet mezi různými PLC systémy.

Řešení problémů je naopak podstatně složitější oproti žebříčkovému diagramu, což je zapříčiněno nižší mírou vizualizace. Další z nevýhod je vyšší míra flexibility, která s sebou nese nutnost standardizace, kdy uživatelé musí používat osvědčené postupy. [5]

1.5.4 Seznam instrukcí

„Seznam instrukcí je obdobou assembleru u počítačů a je také strojově orientován. To znamená, že každé instrukci PLC systému odpovídá stejně pojmenovaný příkaz jazyka. Jazyky mnemokódů poskytují i obvyklý „asemblerový komfort“, tj. aparát symbolického označení návěstí pro cíle skoků a volání, symbolická jména pro číselné hodnoty, pro pojmenování vstupních, výstupních a vnitřních proměnných a jiných objektů programu (datových bloků a tabulek, struktur a jejich prvků), pro automatické přidělování paměti pro uživatelské registry a pro jiné datové objekty, pro jejich inicializaci (zadání počátečního obsahu), pro zadávání číselných hodnot v různých číselných soustavách.“ [1]

Mezi výhody patří vysoká míra standardizace, která přináší pevnou strukturu instrukcí. Tím, že je jazyk zaměřený na instrukce více než na datový tok, je snazší si udělat představu o tom, jak jsou v programu data zpracovávána. Nevýhodou je nedostupnost na některých PLC platformách. [5]

1.5.5 Sekvenční funkční diagram

Petruzella uvádí, že tento programovací jazyk je podobný vývojovému diagramu procesu, který je jím programován. Jazyk je navržen pro programování složitějších procesů. Program lze rozdělit do jednotlivých kroků s více operacemi, které probíhají v paralelních větvích. [4]

Dle Šmejkal a Martináskové je tento jazyk velmi názorný, dále podporuje systémový přístup k programování. Programátorovi tento jazyk pomáhá v tom, aby netvořil chaoticky uspořádané programy, tím, že jej nutí zamyslet se nad podstatou problému. [1]

Sekvenční funkční diagram exceluje v aplikacích řízení chemických procesů a bývá často používán v kombinaci se strukturovaným jazykem. Nevýhodou je náročnost použití na procesy, které nemají sekvenční charakter. [5]

2 PLC SIEMENS

Nadnárodní společnost Siemens byla založena v roce 1847 a působí v mnoha různých odvětvích jako je např. energetika, zdravotnictví, doprava a průmyslová automatizace. Jedná se o jednu z nejužší a nejdržitelnějších společností, která dodává inovativní produkty. [8]

Společnost nabízí řadu produktů průmyslové automatizace, jako jsou např. programovatelné automaty, HMI, frekvenční měniče a průmyslové komunikační sítě. Dále společnost nabízí softwarová řešení jako je například TIA (Totally Integrated Automation) Portal, který slouží jako inženýrský rámec integrující všechny softwarové nástroje pro automatizace do jedné platformy. Siemens nabízí dvě hlavní rodiny PLC, kterými jsou Simatic S5 a Simatic S7. Simatic S5 je starší a již se nevyrábí. Řada S7 nabízí širokou škálu procesorů o různém výkonu, spolehlivost, robustnost, flexibilitu a schopnost splnit různé automatizační požadavky. Procesory se dále dělí do těchto podrodin: S7-1200, S7-1500, S7-300, S7-400. [8]

2.1 SIMATIC S7-1500

Siemens na svých webových stránkách označuje SIMATIC S7-1500 pravděpodobně za nejrychlejší řídicí systém na světě, který svým uživatelům pomáhá dosáhnout maximální produktivity a kvality výrobních procesů díky výkonné systémové sběrnici, rozhraní pro Profinet, rychlé odezvě systému a rychlosti zpracování příkazu v procesoru až 1 nanosekund. Siemens nabízí různé typy procesorových jednotek pro jeho různé výkonnostní třídy. Dále pro systémy nabízí širokou nabídku signálních modulů pro vstupy a výstupy, technologických modulů pro speciální technologické funkce, centralizovaných a decentralizovaných komunikačních modulů. [28]

Simatic S7-1500 je vysoce výkonné PLC pro střední až velké aplikace. S7-1500 přichází s pokročilými funkcemi zaměřenými na bezpečnost, řízení pohybu a zabezpečení. Simatic S7-1500 nabízí řadu modelů, které se liší ve výbavě a vhodnosti pro využití v požadovaných aplikacích. Přehled modelů řady S7-1500 zahrnuje:

- Standardní CPU Simatic S7-1500 – nabízí vysokorychlostní zpracování a pokročilé komunikační možnosti (Profinet, Profibus a průmyslový Ethernet). Tato řada nabízí například tyto verze: CPU 1511-1 PN, CPU 1513-1 PN, CPU 1515-2 PN a CPU 1518-4 PN.
- Simatic S7-1500 s integrovanou funkční bezpečností.

- Simatic S7-1500 pokročilé kontroléry – řízení pohybu, vysokorychlostní čítání, pokročilé komunikační možnosti.
- Simatic S7-1500 T-CPU – nabízí rozšířené funkce řízení pohybu (kinematické funkce, funkce pro řazení apod.).
- Simatic S7-1500 TM NPU – zaměřeno pro aplikace strojového učení a umělé inteligence. [8]



Obrázek 5. SIMATIC S7-1500 CPU
1511-1 PN [33]

2.2 TIA Portal

TIA Portal je software a balíček nástrojů navržených společností Siemens, jehož cílem je integrace více vývojových nástrojů pro oblast automatizace. Toho je dosaženo sjednocením a přemodelováním již existujícího software, mezi nějž patří Simatic Step 7, Simatic WinCC a Sinamics Starter. Prostředí uživateli umožňuje možnost programování, vývoje, konfigurace PLC od Siemens, HMI a frekvenčních měničů. [26]

„Totally Integrated Automation Portal (TIA Portal) propojuje celou řadu inženýrských nástrojů do jedné platformy, která se snadno používá. Obsahuje podporu digitalizace pracovních postupů od digitálního návrhu přes integrovaný inženýring až po transparentní procesy a analytickou práci s daty během všech fází, kterými projekt prochází.“ [39]

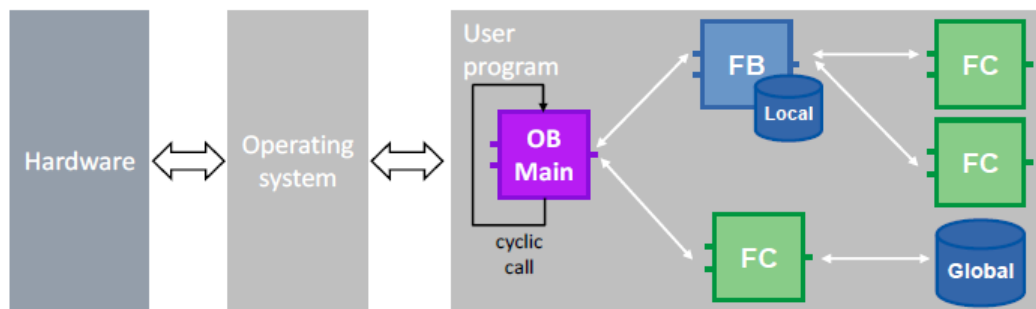
Zástupce českého Siemens uvádí, že TIA V18 je obdobně jako předchozí verze řešením na nové problémy dnešní doby jako je např. kybernetická bezpečnost, dále se součástí oboru průmyslové automatizace stává umělá inteligence, digitální dvojčata, průmyslová metaversa apod. Dále TIA V18 nabízí oproti předchozí verzi více funkcí pro týmovou spolupráci, což umožňuje, že na projektech může současně pracovat více lidí. TIA V18 řeší vzrůstající požadavky na integraci robotů a ostatních komplexních kinematických zařízení tím, že přináší jednoduché a ucelené projektování 5D/6D kinematiky. TIA V18 nově podporuje redundanci úrovně R1 tzn. že přichází s maximální dostupností aplikací. [29]

Na obrázku č. 6 jsou zobrazeny programovací jazyky, které jsou podporované pro programování v TIA PORTAL pro řady PLC SIEMENS S7-1200 a S7-1500. Každý z jazyků má své výhody a může být používán flexibilně v závislosti na aplikaci. Každý blok v uživatelském programu může být naprogramován v jiném jazyce. [3]

Programming language	S7-1200	S7-1500
Ladder diagram (LAD)	yes	yes
Function block diagram (FBD)	yes	yes
Structured Control Language (SCL)	yes	yes
Graph	no	yes
Statement list (STL)	no	yes

Obrázek 6. Přehled podporovaných programovacích jazyků [3]

Kontroléry SIMATIC se skládají z operačního systému a uživatelského programu. Úkolem operačního systému je organizace všech funkcí a sekvencí kontroléru, které nejsou spojeny se specifickým řídicím úkolem. Jedná se například o řízení restartu, aktualizaci obrazu procesu, volání uživatelského programu, paměťový management a vypořádání se s chybami. Operační systém je integrovanou částí kontroléru. Uživatelský program se skládá z bloků, které jsou potřebné pro zpracování konkrétního úkolu. Uživatelský program obsahuje programové bloky, které jsou nahrané do kontroléru. Uživatelský program je vykonáván cyklicky a Hlavní cyklus „Main“ organizačního bloku se vytvoří ve složce programových bloků potom, co je zvolen kontrolér v TIA Portal. Organizační blok je zpracováván cyklicky v nekonečné smyčce. [3]



Obrázek 7. Operační systém a uživatelský program [3]

2.2.1 Programovací bloky

V TIA Portal je možné k programování použít následující typy bloků:

- Organizační bloky (OB)
- Funkční bloky (FB)
- Funkce (FC)
- Datové bloky (DB)

Siemens jako výhody použití různých bloků zmíněných výše považuje možnost vytvoření programu s dobrou a jasnou strukturou. Tyto funkční jednotky pak může programátor znovu použít v rámci projektu případně při tvorbě jiných projektů. Opakované použití již vytvořených bloků v rámci firmy může vést k rychlejšímu odhalení chyb. Co se programování týká, doporučuje Siemens strukturovat úkoly automatizace a funkční celky rozdělit do menších celků, až dokud programátor nedostane funkci, která může být použita v jiných aplikacích s jinými parametry. [3]

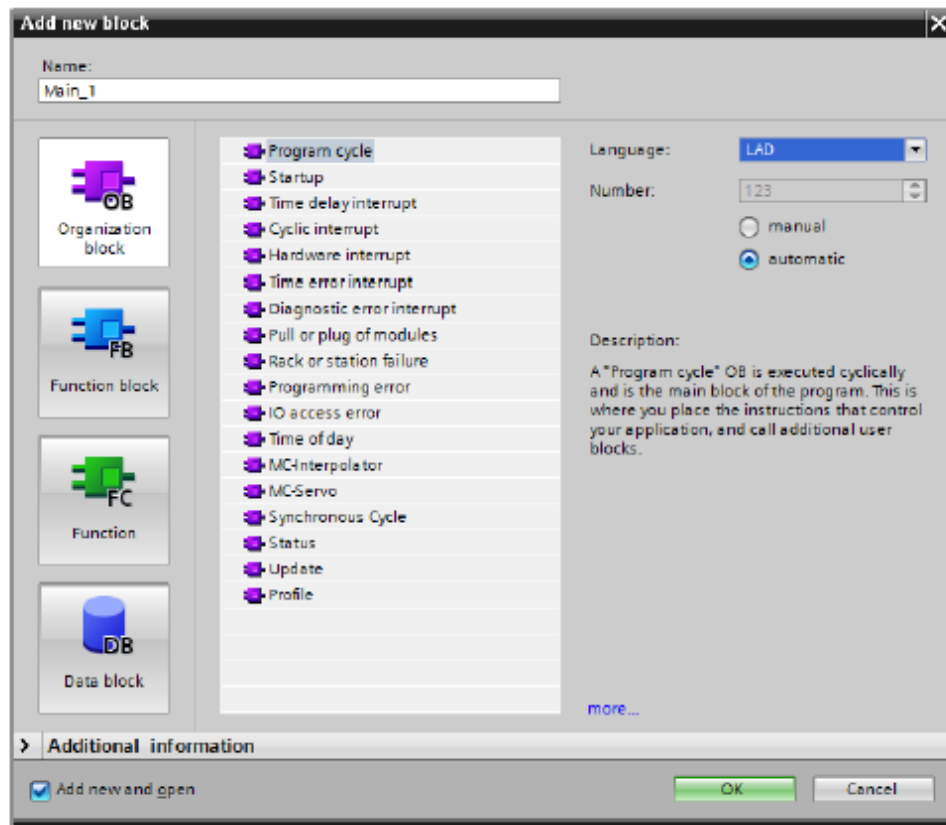
Organizační bloky (OB)

Organizační bloky slouží k vytvoření struktury programu a jsou rozhraním mezi operačním systémem a uživatelským programem. Organizační bloky jsou řízeny událostmi. Události, jako je např. diagnostické přerušení nebo časový interval vedou k tomu, že procesor vykoná organizační blok. Programový cyklus organizačního bloku obsahuje hlavní program a může obsahovat i více než jeden cyklus programu organizačního bloku v uživatelském programu. Během RUN módu je cyklus programu organizačního bloku vykonáván s nejnižší prioritou a může být přerušeno všemi ostatními typy zpracování programu. Startup organizačního bloku nepřerušuje vykonávání cyklu programu organizačního bloku, protože jej procesor vykoná před přechodem do RUN módu. [6]

Mezi přerušení, která mohou přerušit vykonávání organizačního bloku a nemusí být volána z hlavního organizačního bloku patří:

- Cyklické přerušení – je například přerušení zařazené mezi program v žebříčkové logice, kdy dojde po uplynutí časového intervalu k přerušení hlavního programu organizačního bloku. Poté následuje návrat k vykonávání hlavního programu tam, kde skončil před přerušením. Nejčastěji se používá s PID regulátory. Výchozí čas přerušení pro PLC Siemens S7 je 100 ms.
- Hardwarová chyba – jedná se o přerušení, ke kterému dojde, když procesor detekuje jakoukoliv chybu v signálním modulu.
- Přerušení časového zpoždění – přerušení, které se vykoná na základě určité denní doby.
- Start-up přerušení. [38]

Mnoho aplikací je tvořeno uživatelským programem, který je obsažen v jednom programovém cyklu organizačního bloku. Ten je procesorem vykonáván cyklicky. Uživatel může ve svém programu použít více organizačních bloků pro vykonání specifických funkcí jako je například řízení přerušení a chyb nebo vykonávání specifické části programu v předem definovaných časových intervalech. [6]



Obrázek 8. Přidání nového bloku (OB) v TIA Portal [3]

Funkční bloky (FB)

Funkční bloky jsou bloky kódu, jejichž vstupní, výstupní, vstupně-výstupní parametry jsou ukládány do instancí datových bloků a tedy zůstávají dostupné i po provedení bloku, jsou označovány jako „bloky s pamětí“. S funkcemi mají společné to, že podprogramy, které obsahují, jsou volány, když je funkční blok zavolán jiným blokem kódu. Funkční blok může být zavolán několikrát v různých částech programu, při jeho volání mohou formální parametry zůstat bez adresy. Při volání funkčního bloku se vyvolá jeho instance. Pro každou instanci funkčního bloku je nutné vytvořit datový blok instance obsahující specifické hodnoty formálních parametrů deklarovaných ve funkčním bloku. Data instance funkčního bloku mohou být ukládána ve vlastní instanci datového bloku nebo v instanci datového bloku, který ho volá. [7]

Funkční blok, je blok kódu využívající instanci datového bloku pro své parametry a statická data. Funkční blok nabízí variabilní paměť nacházející se v instanci datového bloku. Tato instance poskytuje paměťový blok přidružený k této instanci nebo volání funkčního bloku a jsou zde ukládána data po dokončení funkčního bloku. Programátor může přidružit různé instance datového bloku k různým voláním funkčního bloku. Při volání funkčního bloku a

instance datového bloku procesor vykoná program daného funkčního bloku a uloží parametry bloku a statická lokální data v instanci datového bloku. Jakmile je ukončeno provádění funkčního bloku, procesor se vrátí do bloku kódu, kde k jeho volání došlo a v instanci datového bloku jsou uchovány hodnoty pro instanci daného funkčního bloku. Zmíněné hodnoty jsou dostupné pro následná volání funkčního bloku ve stejném nebo dalším skenování. [6]

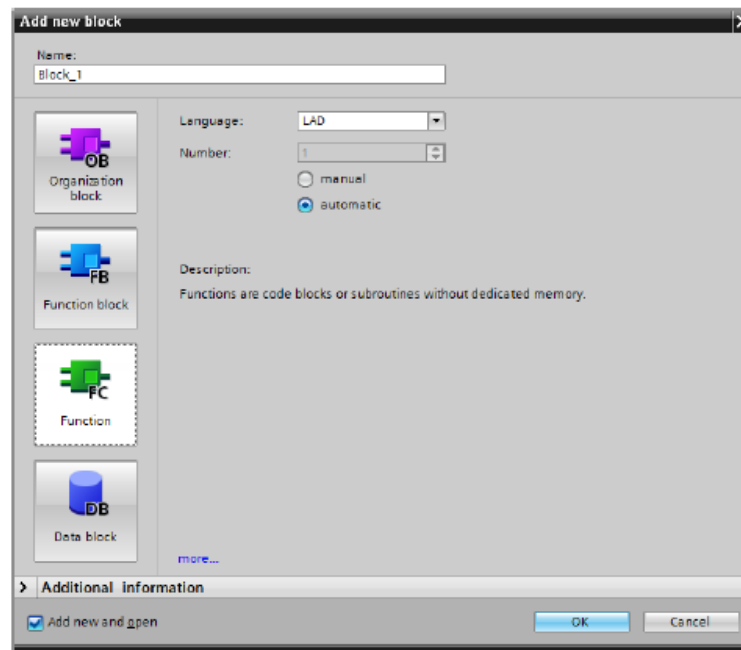
Funkce (FC)

Funkce je blok kódu, díky kterému můžeme přenášet parametry v uživatelském programu. Funkce je vhodné používat ve složitějších a často se opakujících konstruktech, což usnadňuje programování. Funkce nemají instanci datového bloku, kde by mohly uchovat hodnotu parametru. Při volání funkce musí mít všechny její formální vstupní, výstupní a vstupně-výstupní parametry přiřazenou aktuální hodnotu. Funkce mohou využívat globální datové bloky, ve kterých mohou trvale ukládat data. Program, který funkce obsahuje, je vykonán v případě, že je funkce volána jiným blokem kódu. Funkce se nejčastěji používají např. pro matematické funkce, kdy potřebujeme vrátit hodnotu funkce do bloku, kterým je volána, dále pro technologické funkce, které jsou zaměřeny na řízení pomocí bitových logických operací. [7]

Tabulka 1. Rozdíl mezi FC a FB [7]

Funkce (FC)	Funkční blok (FB)
Nemá dedikovanou paměť	Má dedikovanou paměť
Parametry nelze ponechat prázdné	Parametry lze ponechat bez adresy
Nevyžaduje instanci datového bloku	Potřebuje instanci datového bloku

Funkce jsou bloky bez cyklických datových uložišť, což je jeden z důvodů, proč nemohou být hodnoty parametrů bloku uloženy do dalšího volání, z toho plyne, že hodnoty parametrů bloku musí obsahovat aktuální parametry, když jsou volány. [3]



Obrázek 9. Přidání FC v TIA Portal

Siemens uvádí, že pokud chceme trvale uložit data funkce, je nutné použít dostupné globální datové bloky funkcí. Funkce může mít více výstupů a její hodnota může být znovu použita ve strukturovaném řídicím jazyce ve vzorci. Dále doporučuje používat funkce pro často se vyskytující části programu, které jsou volány několikrát v různých částech uživatelského programu. [3]

Datové bloky (DB)

Datové bloky uživatel vytváří za účelem ukládání dat pro bloky kódu uživatelského programu. Všechny programové bloky uživatelského programu mají přístup k datům v globálních datových blocích. Instance datových bloků slouží pro ukládání dat konkrétních funkčních bloků, přesto k těmto datům může přistupovat libovolný blok programu. Po vykonání programu využívajícího tyto datové bloky nedojde ke smazání dat v nich uložených. [6]

2.2.2 Datové typy

Na obrázcích je přehled základních datových bloků a datových skupin pro S7-300/400 and S7-1200/1500. Siemens doporučuje používat datové typy podle PLC, na kterém program běží. [3]

	Description	S7-300/400	S7-1200	S7-1500
Bit data types	<ul style="list-style-type: none"> • BOOL • BYTE • WORD • DWORD 	yes	yes	yes
	<ul style="list-style-type: none"> • LWORD 	no	no	yes
Character type	<ul style="list-style-type: none"> • CHAR (8 bit) 	yes	yes	yes
Numerical data types	<ul style="list-style-type: none"> • INT (16 bit) • DINT (32 bit) • REAL (32 bit) 	yes	yes	yes
	<ul style="list-style-type: none"> • SINT (8 bit) • USINT (8 bit) • UINT (16 bit) • UDINT (32 bit) • LREAL (64 bit) 	no	yes	yes
	<ul style="list-style-type: none"> • LINT (64 bit) • ULINT (64 bit) 	no	no	yes
Time types	<ul style="list-style-type: none"> • TIME • DATE • TIME_OF_DAY 	yes	yes	yes
	<ul style="list-style-type: none"> • S5TIME 	yes	no	yes
	<ul style="list-style-type: none"> • LTIME • L_TIME_OF_DAY 	no	no	yes

Obrázek 10. Základní datové typy korespondující s EN 61131-3 [3]

	Description	S7-300/400	S7-1200	S7-1500
Time types	<ul style="list-style-type: none"> • DT (DATE_AND_TIME) 	yes	no	yes
	<ul style="list-style-type: none"> • DTL 	no	yes	yes
	<ul style="list-style-type: none"> • LDT (L_DATE_AND_TIME) 	no	no	yes
Character type	<ul style="list-style-type: none"> • STRING 	yes	yes	yes
Feld	<ul style="list-style-type: none"> • ARRAY 	yes	yes	yes
Structure	<ul style="list-style-type: none"> • STRUCT 	yes	yes	yes

Obrázek 11. Skupiny dat vytvořené z jiných datových typů [3]

	Description	S7-300/400	S7-1200	S7-1500
Pointer	<ul style="list-style-type: none"> • POINTER • ANY 	yes	no	yes ¹⁾
	<ul style="list-style-type: none"> • VARIANT 	no	yes	yes
Blocks	<ul style="list-style-type: none"> • TIMER • COUNTER 	yes	yes ²⁾	yes
	<ul style="list-style-type: none"> • BLOCK_FB • BLOCK_FC 	yes	no	yes
	<ul style="list-style-type: none"> • BLOCK_DB • BLOCK_SDB 	yes	no	no
	<ul style="list-style-type: none"> • VOID 	yes	yes	yes
PLC data types	<ul style="list-style-type: none"> • PLC DATA TYPE 	yes	yes	yes

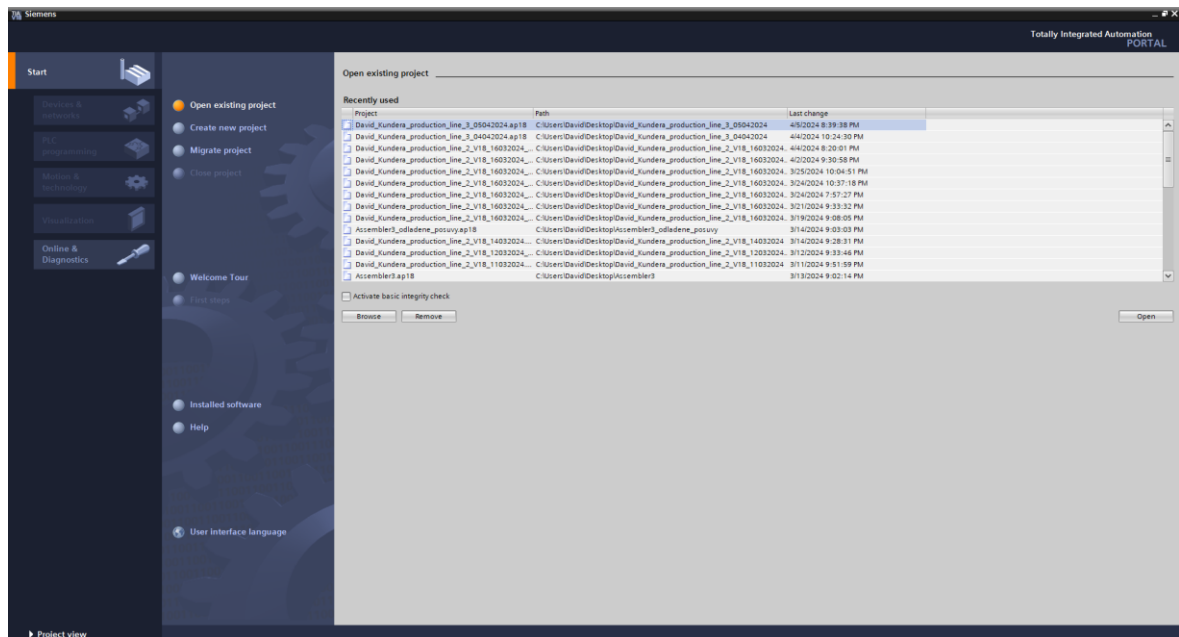
¹⁾ For optimized access, only symbolic addressing is possible

²⁾ For S7-1200/1500 the TIMER and COUNTER data type is represented by IEC_TIMER and IEC_Counter.

Obrázek 12. Typy parametrů pro transfer mezi bloky [3]

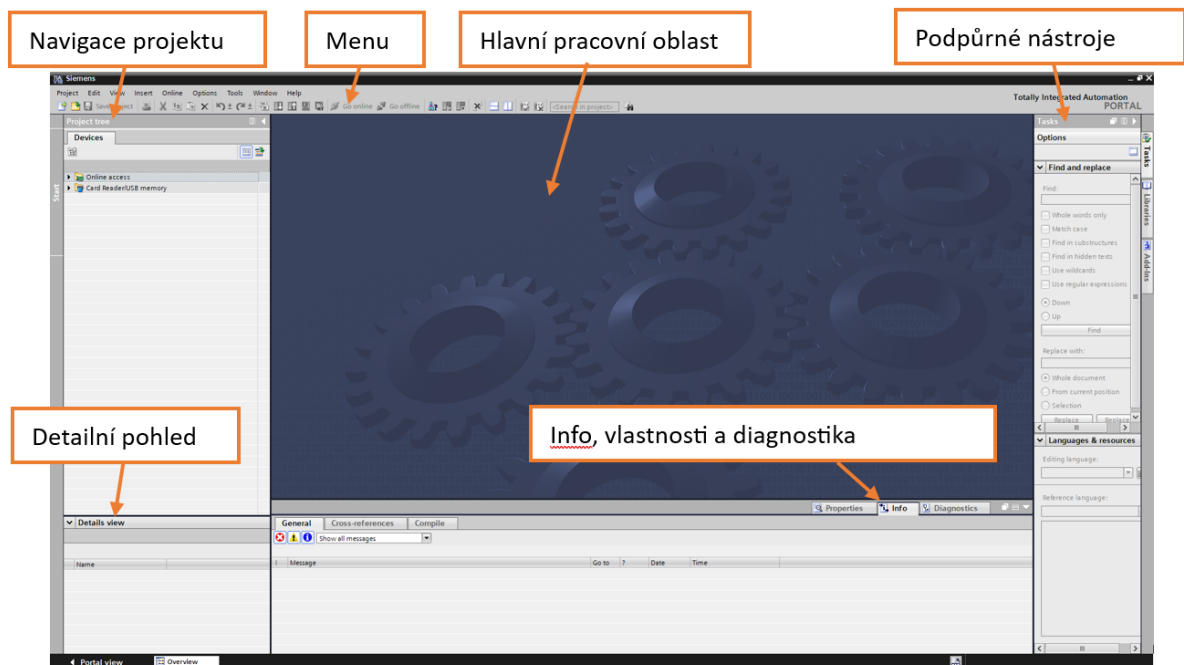
2.3 Projekt v TIA Portal

Vzhledem k tomu, že programované vybavení PLC je vytvořeno v programu TIA Portal V18, budou následující vizualizace právě z tohoto prostředí. Na obrázku č. 13 je zobrazeno výchozí zobrazení – Portal view, které se zobrazí po spuštění programu. V něm uživatel vidí seznam nedávno otevřených projektů. Dále v něm může vytvořit nový projekt nebo migrovat projekt vytvořený v předešlých verzích TIA Portal. Na obrázku vlevo dole se pak uživatel může přepnout do projektového zobrazení – Project view, kde je taktéž možné vytvořit, otevřít nebo migrovat projekt. V projektovém zobrazení pak uživatel vybírá PLC, karty, vytváří samotný program, kompiluje jej, odesílá do PLC, sleduje jeho vykonávání a provádí jeho úpravy.



Obrázek 13. Portal view v TIA Portal V18

Na obrázku č. 14 je zobrazeno projektové zobrazení. Tohle zobrazení nabízí menu s tlačítky (horní lišta) a dále se dělí do následujících 5 oblastí: navigace projektu, hlavní pracovní oblast, podpůrné nástroje, detailní pohled, podokno vlastností, diagnostiky a informací.



Obrázek 14. Project view v TIA Portal V18

2.3.1 Vytvoření projektu

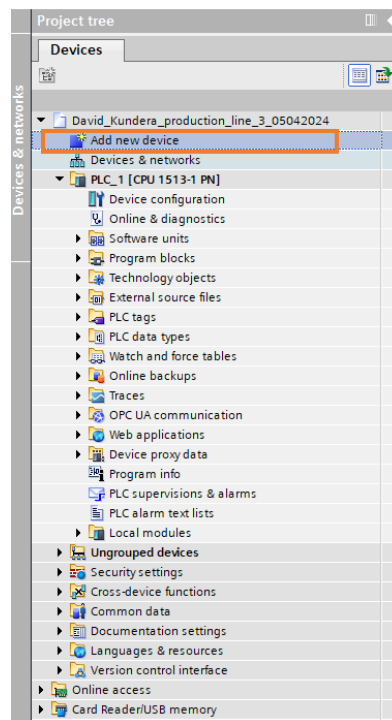
Pro vytvoření nového projektu uživatel klikne na „Create new project“, poté zvolí název projektu, cestu pro uložení, autora příp. komentář k projektu. Následně klikne na tlačítko „Create“ viz obrázek č. 15.



Obrázek 15. Vytvoření nového projektu

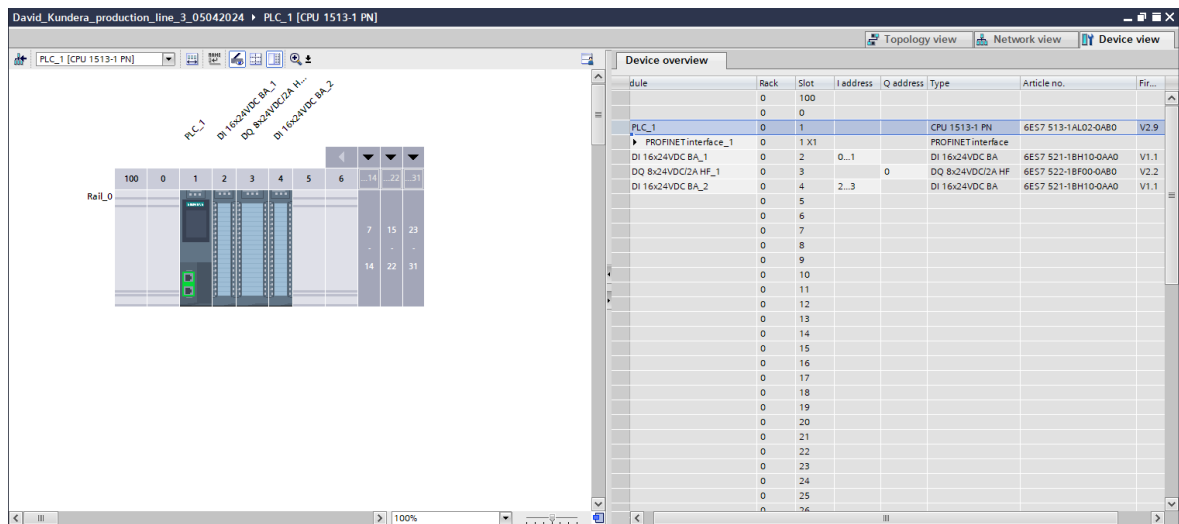
2.3.2 HW konfigurace

Pro přidání zařízení uživatel klikne na „Add new device“, kde si zvolí PLC a příslušné moduly viz obrázek č. 16.



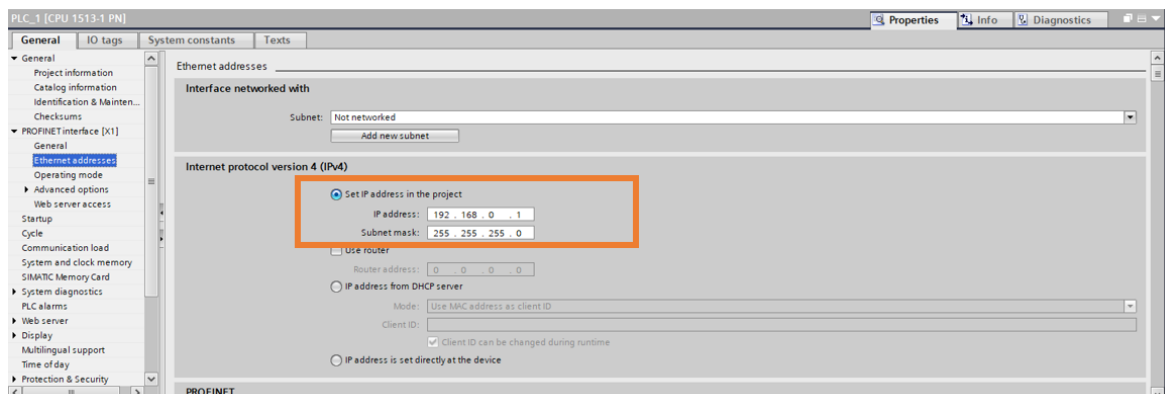
Obrázek 16. Přidání nového zařízení

V tomto konkrétním případě bylo zvoleno CPU 1513-1 PN s katalogovým číslem 6ES7 513-1AL02-0AB0. Dále dva moduly s digitálními vstupy DI 16x24VDC BA s katalogovým číslem 6ES7 521-1BH10-0AA0 a jeden modul s digitálními výstupy DQ 8x24VDC/2A HF s katalogovým číslem 6ES7 522-1BF00-0AB0 viz obrázek č. 17.



Obrázek 17. Přehled vybraných zařízení

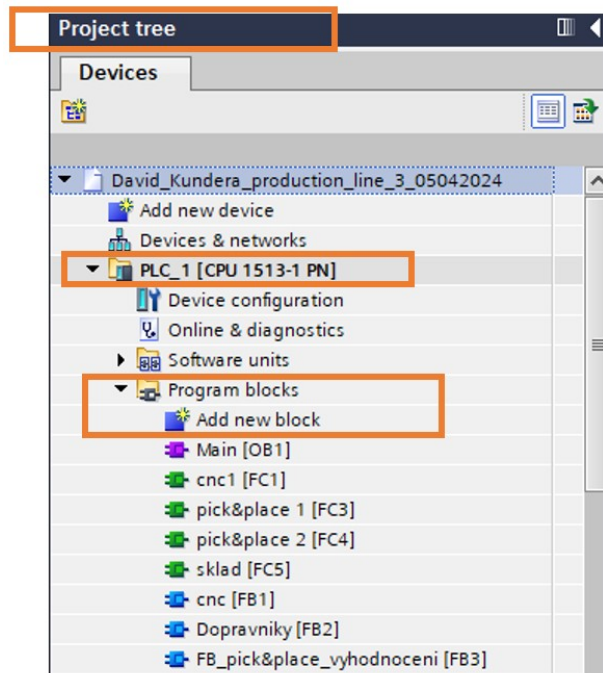
Pro připojení k PLC je nutné zvolit jeho IP adresu. V přehledu příslušných zařízení uživatel zvolí dané PLC a v podokně General -> Profinet interface [X1] -> Ethernet addresses. Adresu uživatel volí dle rozsahu a dostupnosti adres v dané síti, v níž bude PLC i s počítačem připojené. V tomto případě bude zvolena defaultní IP adresa 192.168.0.1, protože v bakalářské práci bude použit simulátor PLC nikoliv reálné fyzické PLC viz obrázek č. 18. PLC simulátor ke svému chodu nevyžaduje přidání karet se vstupy a výstupy.



Obrázek 18. Nastavení IP adresy

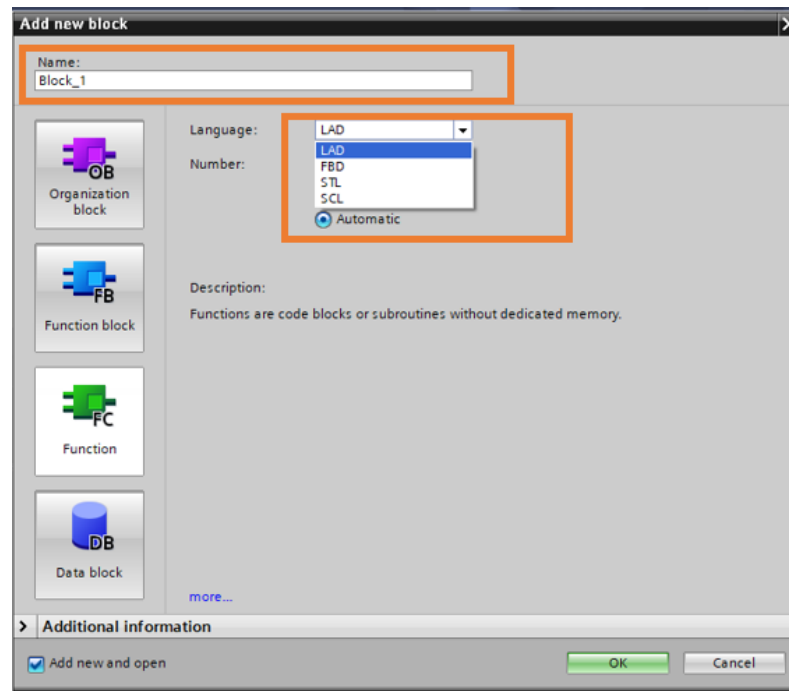
2.3.3 Vložení programového bloku

Programový blok se vkládá v „Project tree“, kdy po zvolení PLC uživatel klikne na záložku „Add new block“, poté si vybere jednu z možností – OB, FB, FC, DB) viz obrázek č.19. V programu je již OB s názvem Main. Do něj uživatel vloží přetažením do hlavní pracovní oblasti organizační bloky, které chce, aby byly volány a prováděny.



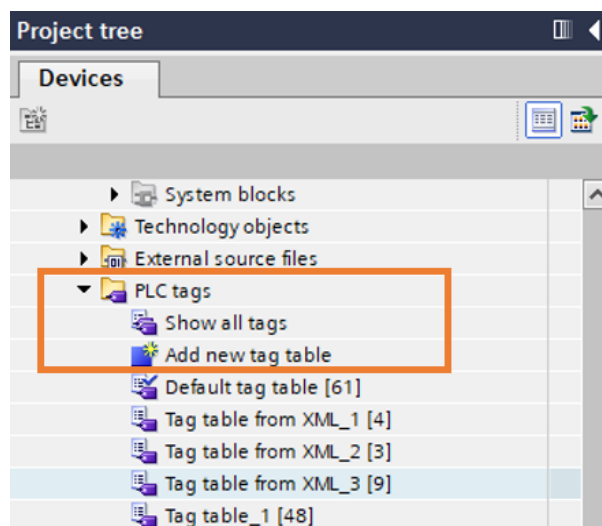
Obrázek 19. Přidání nového bloku

Na obrázku č. 20 uživatel provede výběr programového bloku, zvolí jeho název a programovací jazyk a potvrdí tlačítkem „OK“. Zvolený programovací jazyk uživatele nelimituje tím způsobem, že by při programování daného programového bloku nemohl využít i další z jazyků.



Obrázek 20. Výběr programového bloku

Po kliknutí na tlačítko „OK“ by měl uživatel vytvořit vstupy a výstupy, se kterými bude v programu pracovat a následně může již uživatel začít psát program. Vstupy a výstupy přidá uživatel po kliknutí na tlačítko „Add new tag table“ viz obrázek č. 21.



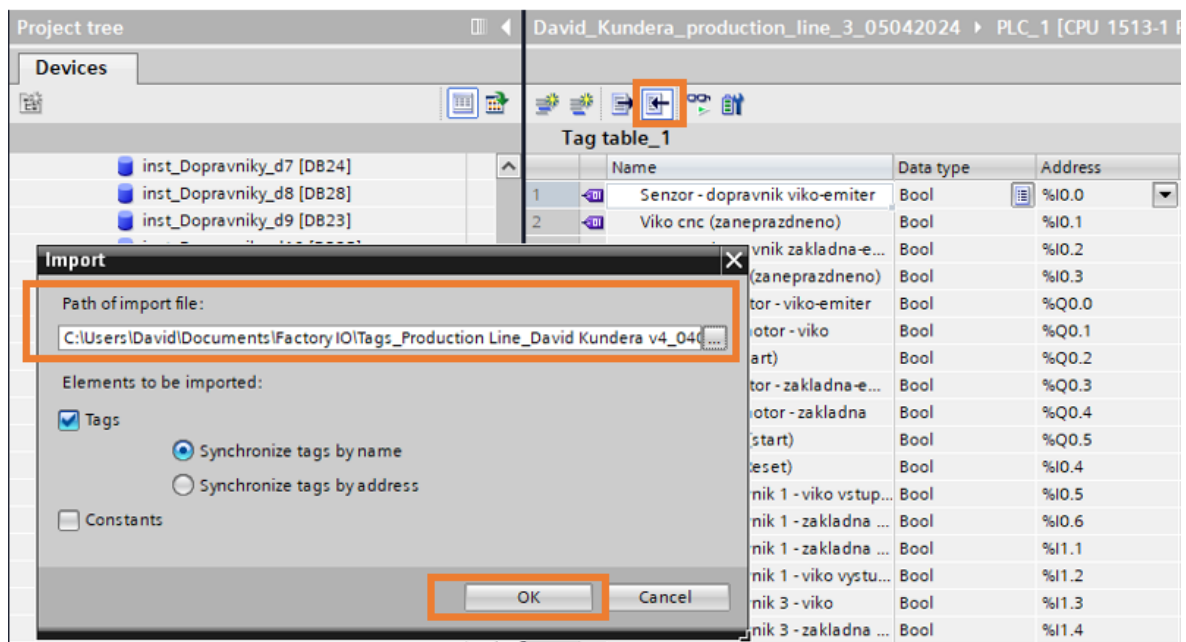
Obrázek 21. Přidání vstupů a výstupů

Zvolené tagy si uživatel pojmenuje, ideálně se název snaží spojit s daným pracovištěm, aby z něj bylo jasné, k čemu se váže. To uživateli při psaní programu pomůže v rychlejší orientaci. Na obrázku č. 22 je tagová tabulka, jak je patrné, uživatel k tagu přiřadí datový typ, adresu případně další vlastnosti.

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervision	Comment
1	Senzor - dopravnik viko-emiter	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	Viko cnc (zaneprazdnenno)	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	Senzor - dopravnik zakladna-e...	Bool	%I0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	Zakladna cnc (zaneprazdnenno)	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	Dopravnik motor - viko-emiter	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	Dopravnik 1 motor - viko	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Lids center (start)	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	Dopravnik motor - zakladna-e...	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	Dopravnik 1 motor - zakladna	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	Bases center (start)	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	FACTORY I/O (Reset)	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

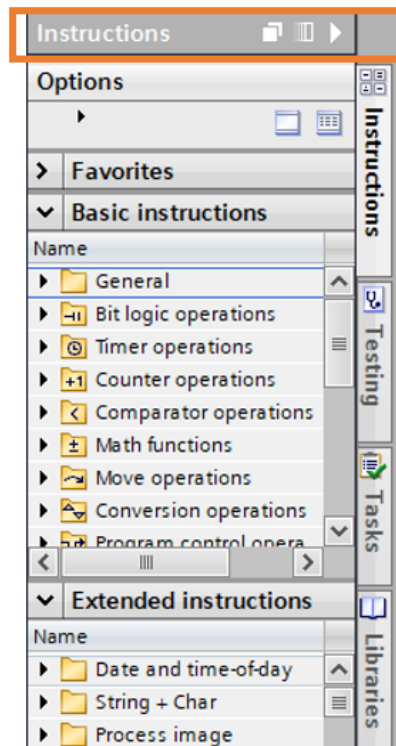
Obrázek 22. Tabulka s tagy

Na obrázku č. 23 je znázorněno, jakým způsobem lze do TIA Portal naimportovat tagy z Factory I/O.



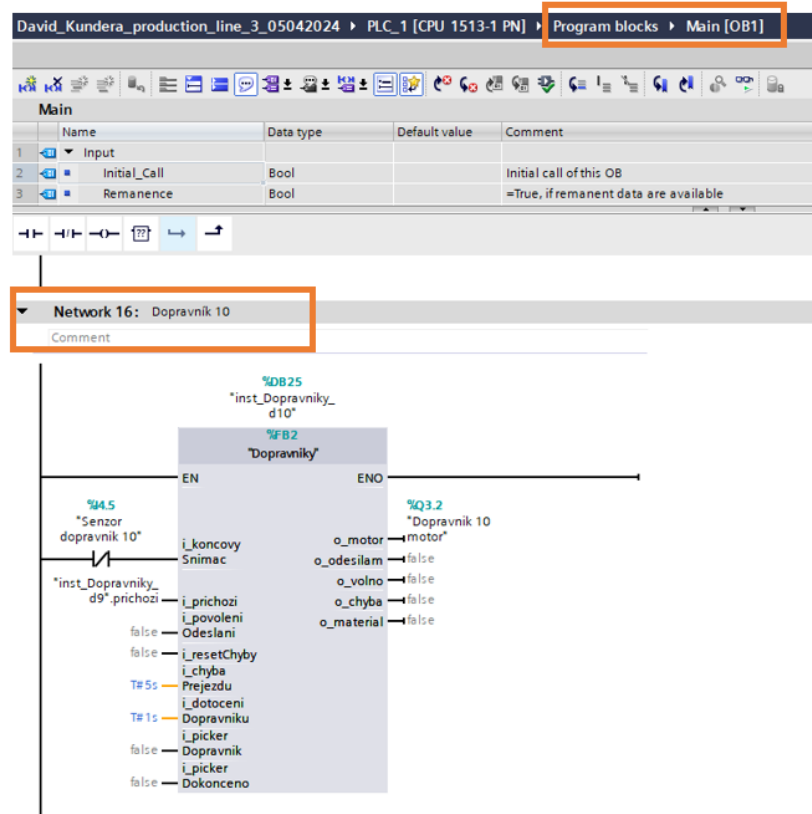
Obrázek 23. Nahrání vstupů a výstupů z Factory I/O do TIA Portal

Program nabízí sadu základních i rozšířených instrukcí viz obrázek č. 24, které uživatel může používat. Do záložky „Favorites“ se ukládají nejčastěji používané instrukce.



Obrázek 24. Nabídka instrukcí

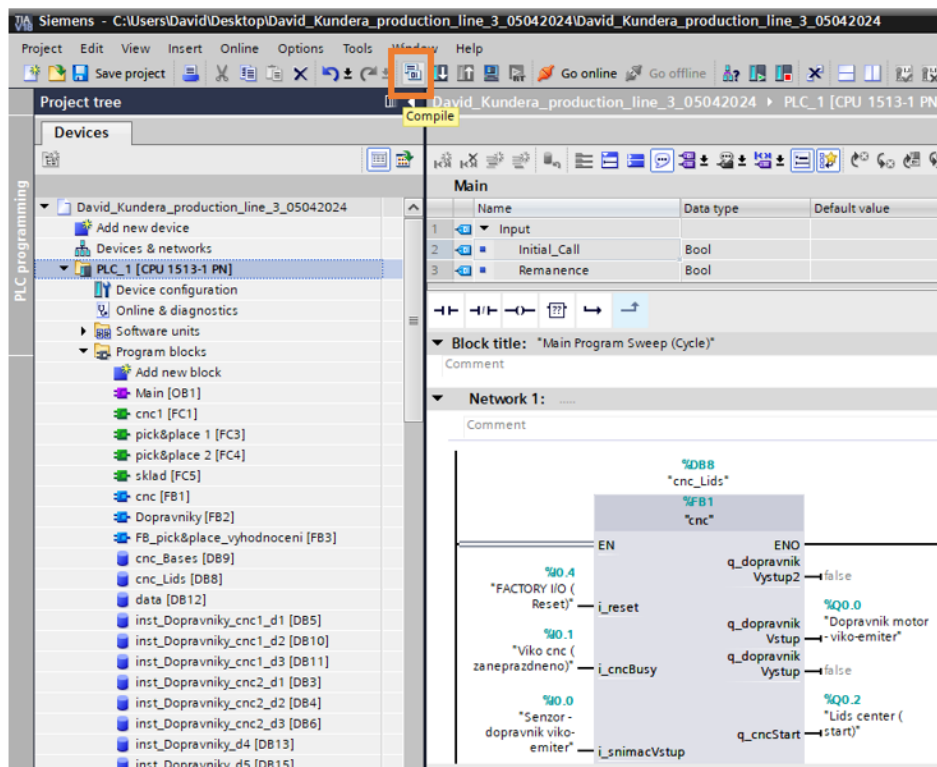
Jakmile uživatel vytvoří programový blok, například na obrázku č. 25 je zobrazena instance funkčního bloku Dopravníku, která vznikla přetažením funkčního bloku Dopravníky [FB2] ze záložky přetažením do MAIN [OB1], může přejít ke kompilaci programu a jeho nahrání do PLC. Uživatel si může každý network programového bloku (OB, FB, FC) pojmenovat, například níže je Network 10 pojmenován Dopravník 10.



Obrázek 25. Pojmenování networku

2.3.4 Kompilace projektu

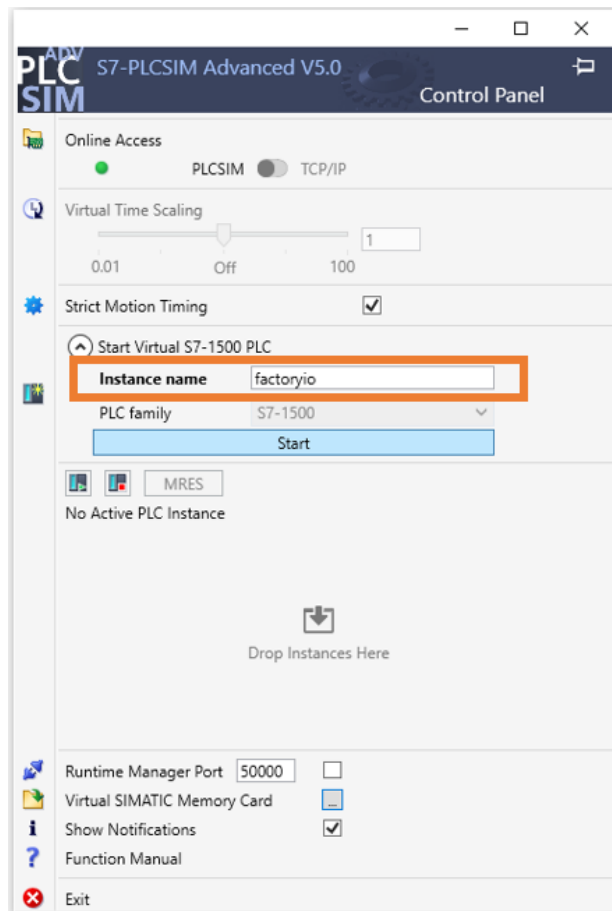
Jakmile má uživatel program vytvořený následuje jeho kompilace, kdy je nutné kliknout na tlačítko „Compile“ viz obrázek č. 26 v liště menu. Následně se provede hardwarová i softwarová kompilace.



Obrázek 26. Kompilace projektu

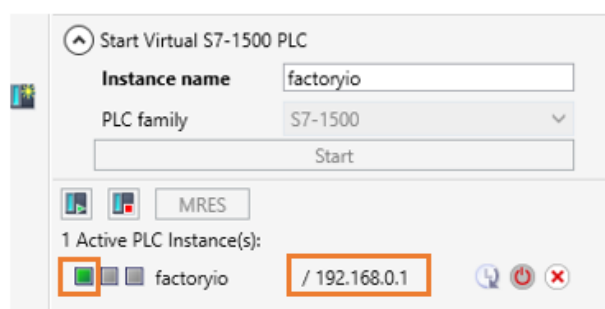
2.3.5 Nahrání projektu do PLCSIM

Před samotným nahráním programu do PLC je nutné spustit simulátor PLC. K tomuto účelu bude použit program PLCSIM Advanced V5.0. Pro vytvoření instance reprezentující PLC do pole „Instance name“ je nutné napsat „factoryio“, v poli „PLC family“ pak vybrat S7-1500 (jedno z řady CPU, které simulátor podporuje), protože program poběží na PLC této řady a kliknout na tlačítko „start“ viz obrázek č. 27.



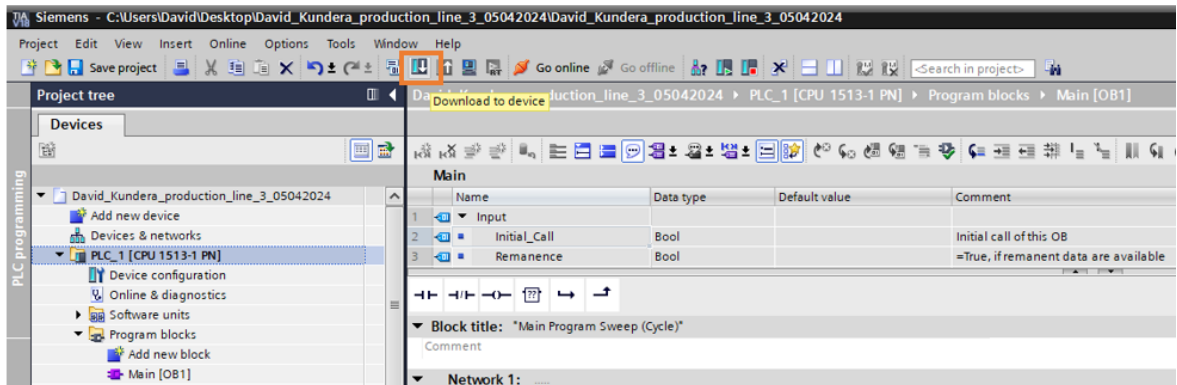
Obrázek 27. Spuštění PLCSIM Advanced V5.0

Po chvíli dojde ke spuštění simulátoru a jak je patrné z obrázku č. 28, v PLCSIM se rozsvítí zelená kontrolka, která signalizuje, že simulátor běží, je v něm nahráný program a je v RUN módu. Pokud by v simulátoru nebyl nahráný žádný program, tj. program by byl nahráván poprvé nebo by byla vymazána paměť simulátoru „Virtual SIMATIC Memory card“ viz obrázek č. 27, kontrolka by svítila žlutě a PLC by bylo ve STOP módu až do doby, než by do něj byl v TIA Portal nahrán program.



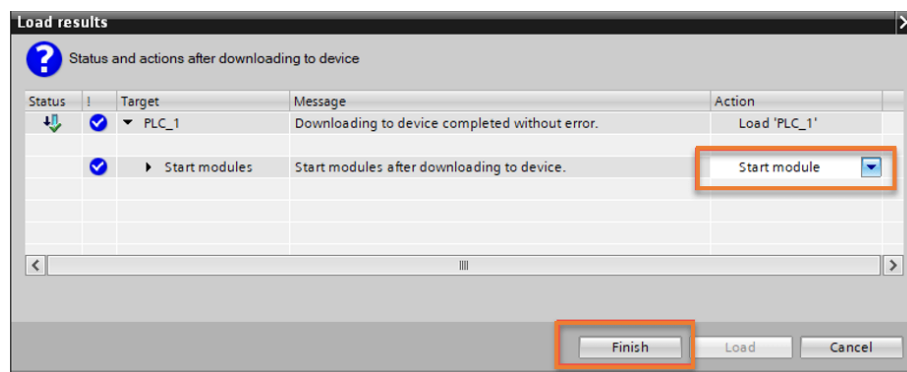
Obrázek 28. Spuštěný PLCSIM Advanced V5.0

Nyní se uživatel může vrátit do TIA Portal a nahrát program do PLC simulátoru kliknutím na tlačítko „Download to device“ viz obrázek č. 29.



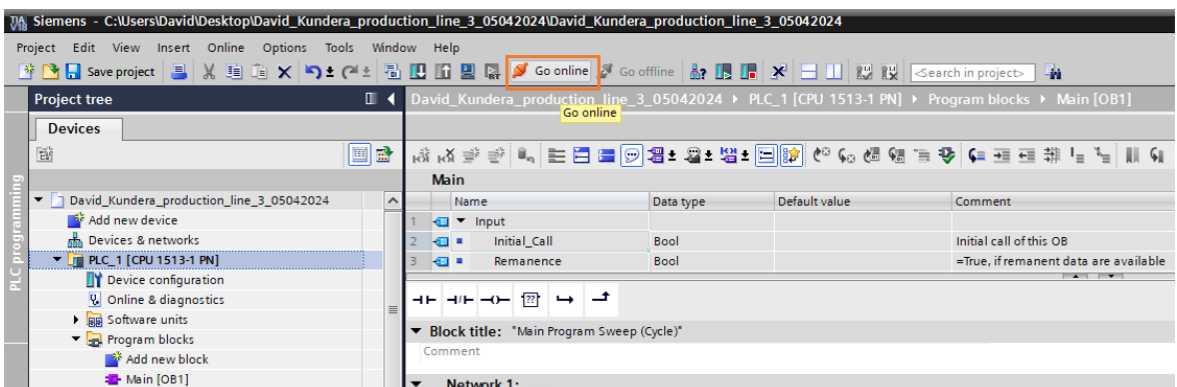
Obrázek 29. Nahrání programu do PLC

Nyní je nutné v rozklikávací nabídce vybrat „Start module“ a následně kliknout na tlačítko „Finish“ viz obrázek č. 30.



Obrázek 30. Nahrání programu do PLC - pokračování

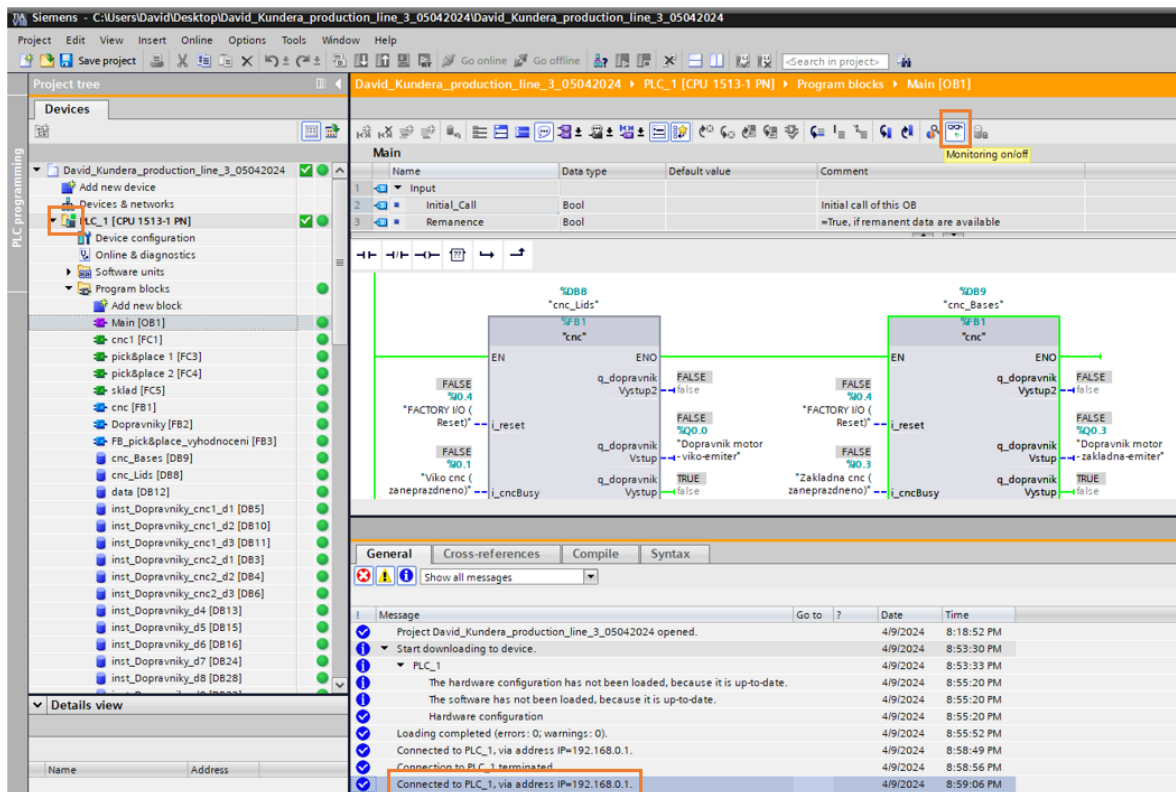
Po kliknutí na tlačítko „Go online“ viz obrázek č. 31 dojde k připojení k simulátoru, což znázorňuje zelená kontrolka viz obrázek č. 32.



Obrázek 31. Spuštění online režimu

2.3.6 Online kontrola funkčnosti

Přechod do online režimu je možné realizovat i přes ikonku „Monitoring on/off“ viz obrázek č. 32. Před kliknutím na ikonku je nutné v „Project tree“ vybrat programový blok, jehož vykonávání chce programátor sledovat a pak už jen stačí kliknout na zmíněnou ikonku s brýlemi. Kontrola funkčnosti programátorovi dává zpětnou vazbu, co se vykonávání programu týče a také, zda je offline verze projektu shodná s projektem nahraným v PLC.



Obrázek 32. Online kontrola funkčnosti

3 FACTORY I/O

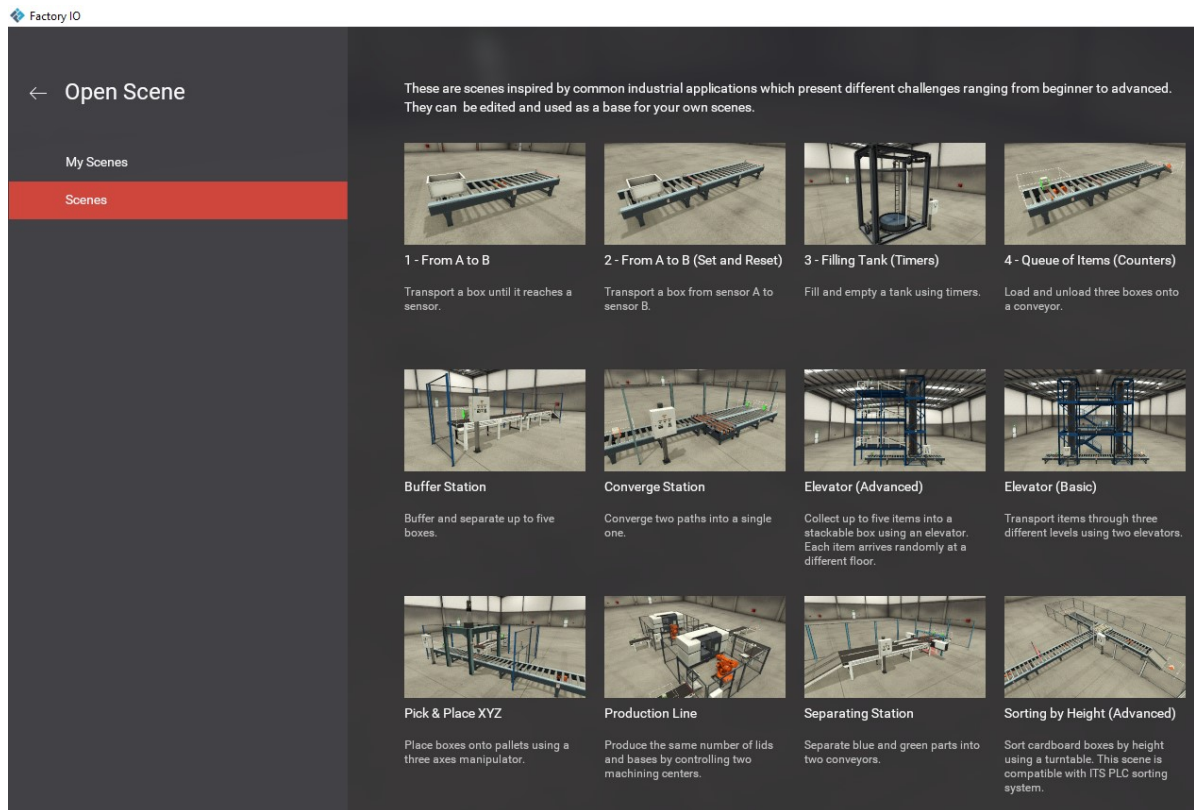
Třetí kapitola se zaměřuje na charakteristiku 3D simulačního softwaru Factory I/O, popisu jeho prostředí, možnostem připojení k PLC, nabídkou jednotlivých edic a alternativou v podobě EasyPLC.

3.1 Charakteristika softwaru

Factory I/O je 3D simulační software od společnosti Real Games, který slouží k výuce automatizačních technologií. Cílem návrhu tohoto softwaru bylo, aby umožňoval snadnou použitelnost a rychlost, co se budování virtuální továrny týče. Software nabízí celou řadu průmyslových dílů a scén inspirovaných reálným výrobním prostředím. Jednotlivé scény jsou rozděleny dle úrovně obtížnosti od začátečníků až po pokročilé. Nejčastěji je Factory I/O používáno jako platforma pro školení PLC z důvodu jejich vysokého zastoupení v průmyslové praxi. Factory I/O je možné použít také s mikrokontroléry, SoftPLC, Modbus a dalšími technologiemi. [24]

Factory I/O je flexibilní prostředí, které nabízí bohaté knihovny komponent a scénáře, díky kterým si mohou studenti snadno sestavit rozsáhlé kombinace scén a komponent, naprogramovat je, ověřit jejich funkčnost a případně je opravit. Některé verze podporují propojení simulace se skutečnými kontroléry. [23]

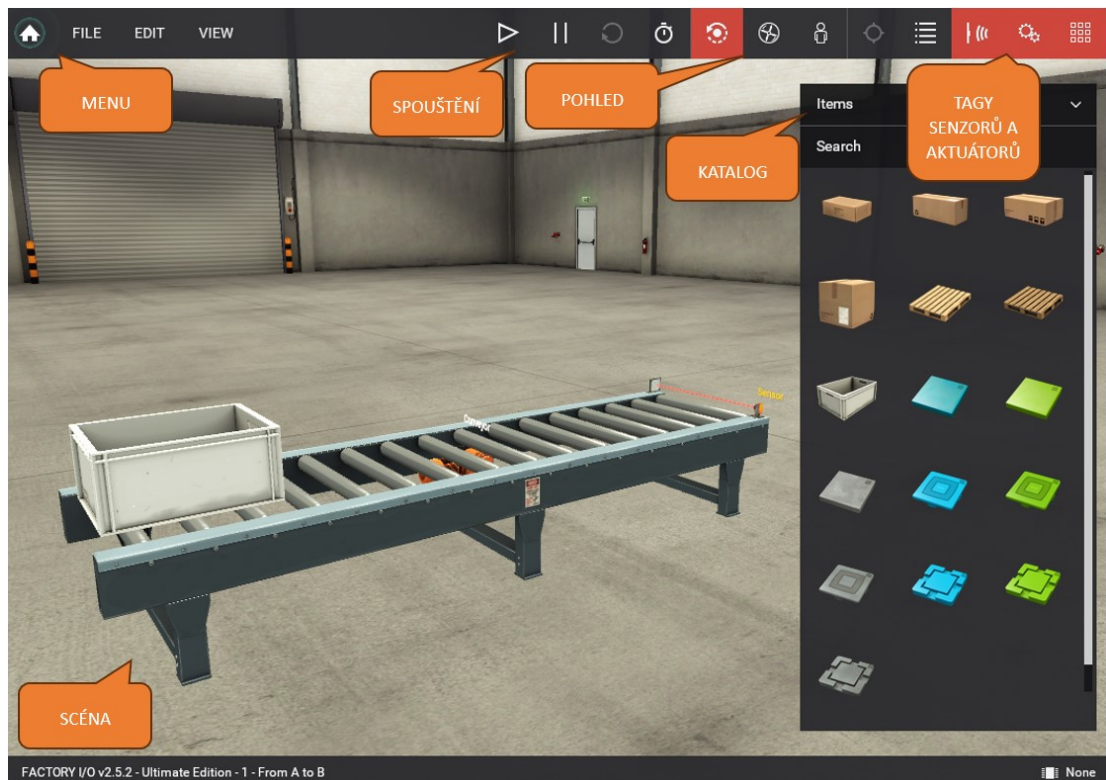
Factory I/O nabízí přes 20 scén, které jsou inspirované reálným výrobním prostředím. Uživatel může scénu použít, tak jak je, nebo si ji přetvořit podle sebe. Scény standardně obsahují předměty jako jsou například díly, palety, obalové jednotky apod., senzory, aktuátory, stanice, dopravníky a mnoho jiných. Scény neobsahují řídicí program, ten si uživatel musí vytvořit např. v TIA Portal. Některé ze scén jsou zobrazeny na obrázku č. 33.



Obrázek 33. Nabídka scén z Factory I/O

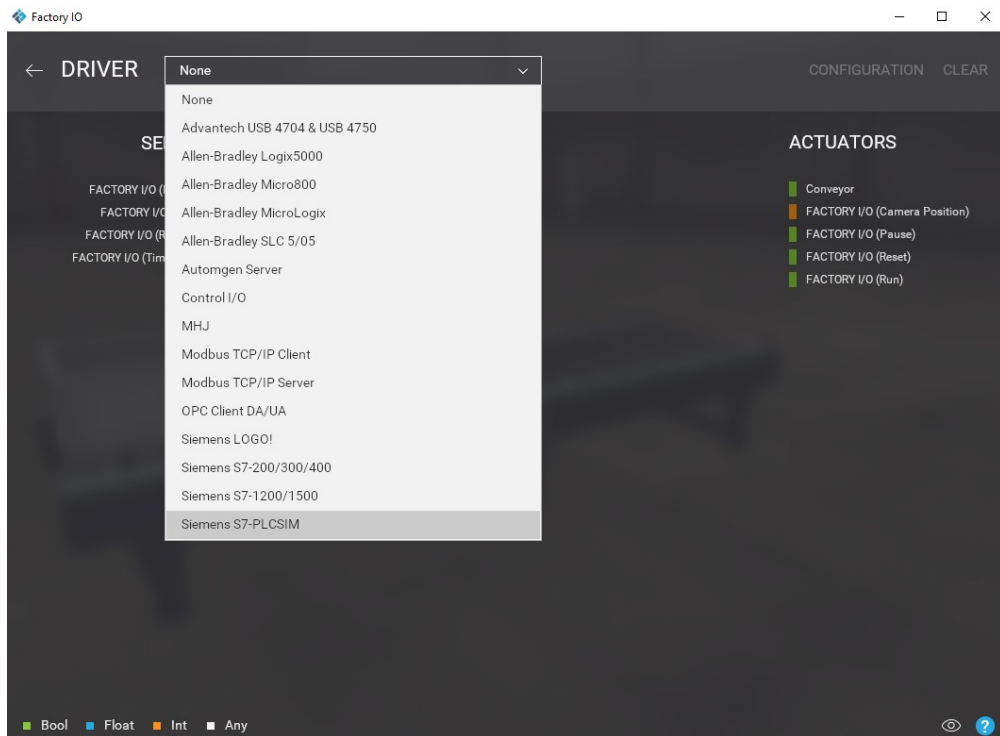
3.2 Prostředí

Pro zobrazení prostředí Factory I/O byla použita jedna z tréninkových scén, jejímž cílem je dostat přepravní bednu z bodu A do bodu B. V horní liště se nachází možnost zobrazení/skrytí tagů pro senzory a aktuátory, pohledy (pohled z první osoby, orbit, létající kamera), paleta s výběrem předmětů (úplně vpravo), tlačítka související se spuštěním simulace (přepínač mezi během a editací scény, pauza, reset), možnosti zobrazení, editace, soubor (ukládání, vytvoření nové scény, otevření uložené scény, nastavení, ovladače a uvítací okno (symbol domečku)). Scéna je umístěna do objektu výrobní haly, který je na začátku prázdná. Po kliknutí pravým tlačítkem na předmět lze nastavit možnosti jeho manipulace (vertikální, horizontální), natočení, typ signálu, zkopírovat, popř. jej smazat.



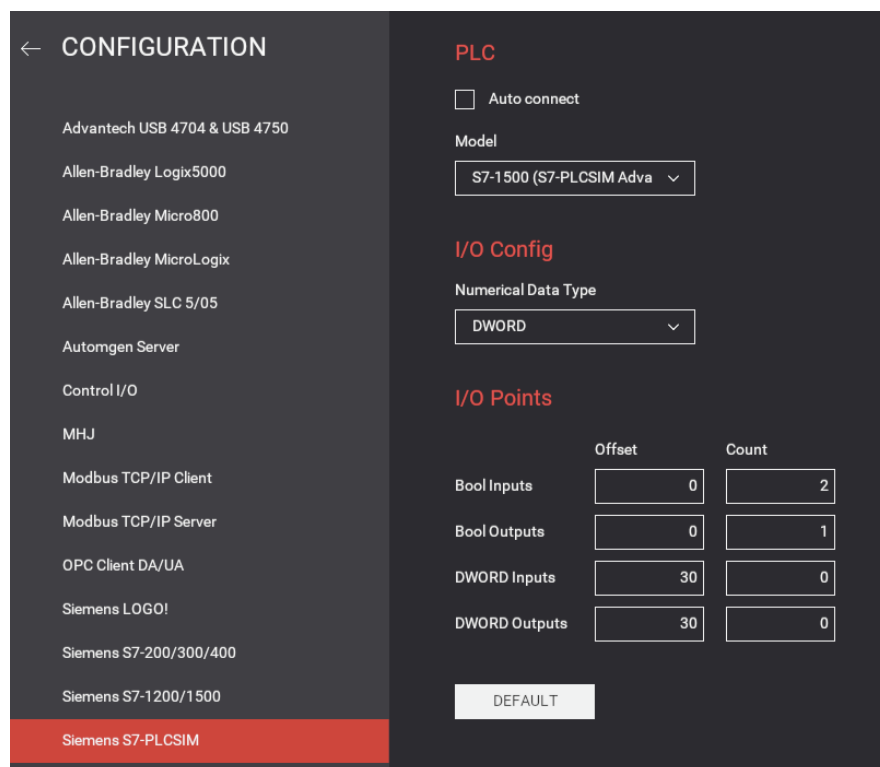
Obrázek 34. Popis prostředí Factory I/O

Vstupní/výstupní ovladače jsou vestavěnou funkcí Factory I/O, která je zodpovědná za komunikaci s externím kontrolérem. Factory I/O nabízí ovladače viz obrázek č. 35, přičemž každý z nich může být použit se specifickou technologií tzn. uživatel vybírá ovladač podle použitého kontroléru. Po výběru a nakonfigurování ovladače viz obrázek č. 35 a 36, může uživatel namapovat tagy jednotlivých senzorů a aktuátorů na zvolený ovladač viz obrázek č. 37. [25]



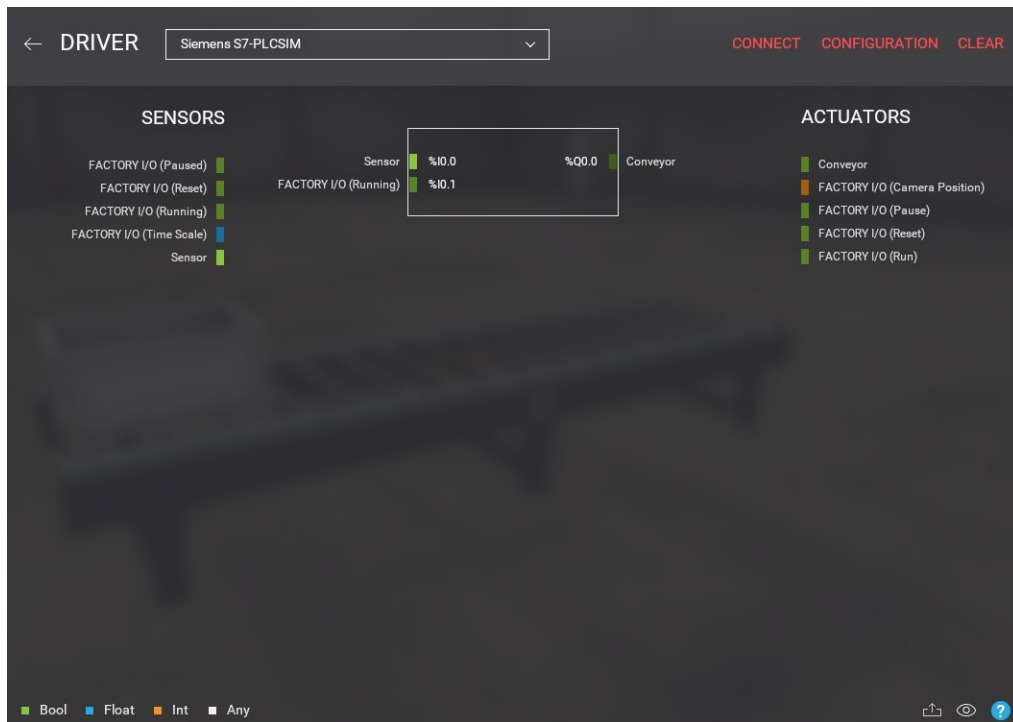
Obrázek 35. Výběr řady PLC ve Factory I/O

Program umožňuje export jednotlivých vstupů a výstupů ve formátu .xml, který je možné nahrát do softwaru, ve kterém bude probíhat samotné programování.



Obrázek 36. Nastavení množství vstupů a výstupů ve Factory I/O

Na obrázku č. 36 si uživatel může nastavit množství vstupů a výstupů. Maximální množství digitálních vstupů a výstupů je 256. Analogových vstupů a výstupů může uživatel použít až 64.



Obrázek 37. Přiřazení senzorů a aktuátorů ve Factory I/O

3.3 Nabízené edice

Obrázek č. 39 zobrazuje edice, které je možné si pořídit. Factory I/O nabízí celkem 7 edic a 1 zkušební verzi, která je zdarma. Nejdražší je edice Ultimate, která stojí 278 euro ročně. Factory I/O nabízí možnost licenci platit měsíčně nebo jednorázově (zahrnuje aktualizace, bezplatnou uživatelskou podporu a přístup ke všem verzím Factory I/O 2.x). Obrázek č. 40 zobrazuje, co jednotlivé edice nabízí a v čem se oproti ostatním liší. [11]

The screenshot displays eight different editions of Factory I/O software, each with its own set of features and pricing. The editions are arranged in two rows of four. Each edition card includes a title, a list of key features, the annual price, and a 'See more' link. The 'Free trial' edition is highlighted with a black border.

Obrázek 38. Nabízené edice Factory I/O [11]

	Starter	Allen-Bradley	Siemens	Modbus & OPC	MHJ	Automgen	Ultimate
All parts	✓	✓	✓	✓	✓	✓	✓
All scenes	✓	✓	✓	✓	✓	✓	✓
Digital & analog I/O	✓	✓	✓	✓	✓	✓	✓
SDK							✓
Control I/O (integrated softPLC)	✓						✓
Allen-Bradley Ethernet connection		✓					✓
Siemens Ethernet connection			✓				✓
Modbus TCP server & client				✓			✓
OPC Client DA & UA				✓			✓
WinSPS-S7 & Grafcet-Studio					✓		✓
Automgen						✓	✓
Advantech USB 4750 & 4704 wiring							✓

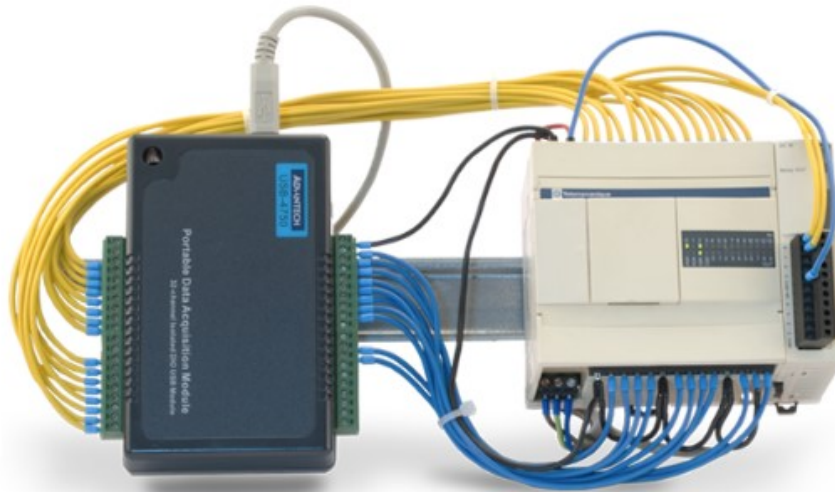
Obrázek 39. Porovnání nabízených edic Factory I/O [22]

„Factory I/O používá ovladače k pro rozhraní s PLC, SoftPLC, PLC simulátory , Modbus a mnoha dalšími technologiemi. Každá edice obsahuje balíček ovladačů pro konkrétní technologii (např. edice Allen-Bradley, edice Siemens, ...).“ [12]

Každá z edic uvádí seznam kompatibilních modelů PLC a požadovaného softwaru, který může být použit s Factory I/O, pro PLC Siemens viz obrázek č. 41. Při potřebě připojit PLC, které na seznamu není, je nutné použít I/O USB komunikační modul případně Advantech USB 4704. Při této volbě musí uživatel zakoupit edici Ultimate. [14]

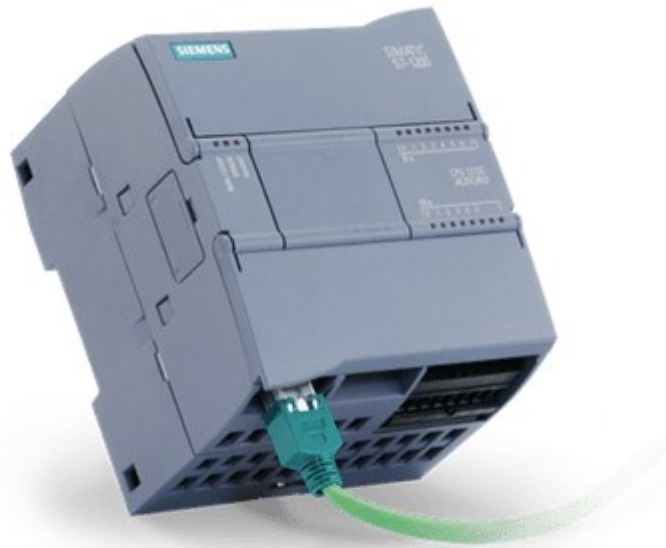
LOGO!	S7-200 / 300 / 400	S7-1200 / 1500	S7-PLCSIM
<ul style="list-style-type: none"> ✓ LOGO! 0BA7 or 0BA8 ✓ LOGO! Soft Comfort Software 	<ul style="list-style-type: none"> ✓ S7-200/200 SMART/300/400 ✓ STEP 7-Micro/WIN Software or STEP 7 Professional (TIA Portal) 	<ul style="list-style-type: none"> ✓ S7-1200 / 1500 ✓ STEP 7 (TIA Portal) 	<ul style="list-style-type: none"> ✓ S7-PLCSIM (V5 / 13 / 14 / 15 / 16) ✓ STEP 7-Micro/WIN Software or STEP 7 (TIA Portal)

Obrázek 40. Seznam kompatibilních modelů PLC a požadovaného softwaru [14]



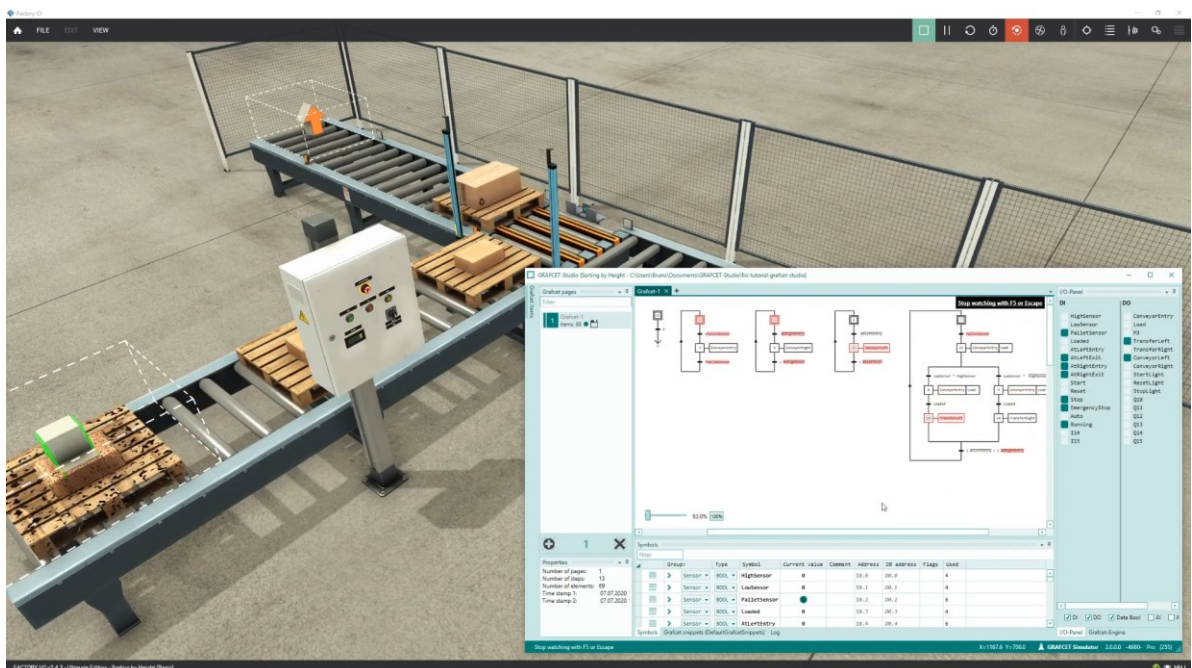
Obrázek 41. Izolovaný digitální I/O modul Advantech USB 4750 připojený k PLC [17]

K PLC značky Allen-Bradley a Siemens je možné se připojit skrze Ethernet. K PLC, které nejsou na seznamu kompatibility, je nutné dokoupit modul Advantech USB 4704 nebo 4750. Mezi výhody Factory I/O patří: velikost jen 600 MB, jedna licence v sobě zahrnuje 20 scén, které mohou být ihned použity, možnost učit celou třídu současně, široká paleta značek PLC, se kterými spolupracuje, studenti mohou experimentovat v podstatě bez rizika, že by zničili strojní zařízení. [12]



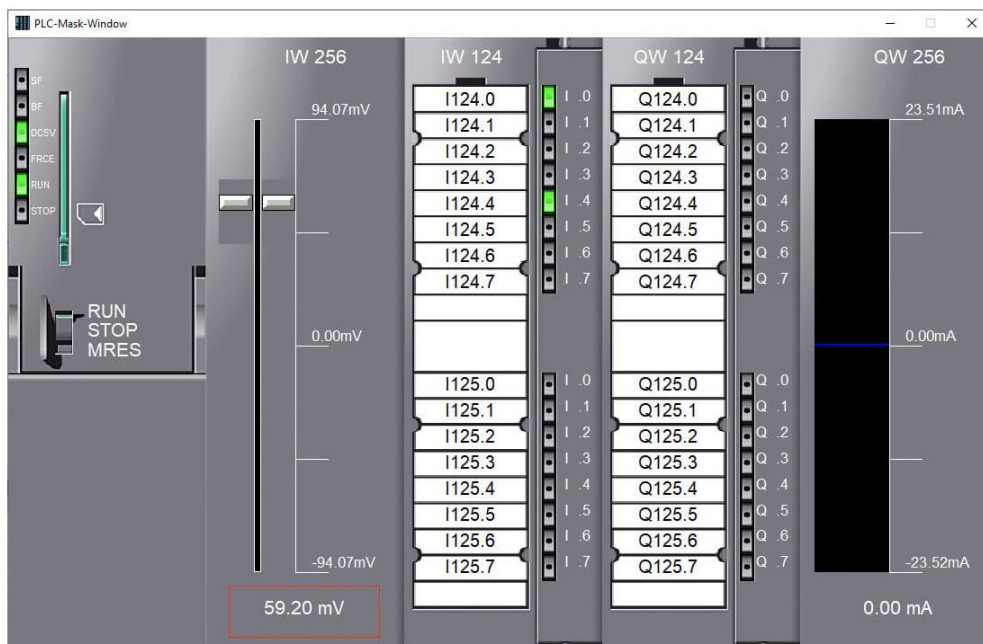
Obrázek 42. Připojení PLC Simatic S7-1200 přes Ethernet [17]

Pro studenty, kteří nemají k dispozici reálné PLC a software např. TIA Portal, je ideální zainvestovat do edice Starter nebo edice MHJ. Obě zmíněné edice stojí 40 euro ročně a obsahují v sobě PLC simulátor tzn. není potřeba mít k dispozici fyzické PLC. Edice MHJ nabízí možnost programování ve WinSPS-S7 a Graftec studiu viz obrázek č. 44.



Obrázek 43. Graftec studio [27]

WinSPS-S7 je kompletní programovací systém pro S7-300®, S7-400® a kompatibilní např. s PLC (VIPA/YASKAWA S7-PLC). Součástí je softwarové PLC, které slouží ke zjednodušení simulace programů S7. Digitální a analogové vstupy a výstupy jsou monitorovány díky masce PLC viz obrázek č. 45. Programy lze vytvářet v seznamu instrukcí, funkčním blokovém schématu a žebříčkovém jazyce. K dispozici je možnost zmíněné jazyky kombinovat v jednom bloku. [39]



Obrázek 44. Maska PLC [39]

Edice Starter v sobě zahrnuje simulátor PLC, více než 20 scén z průmyslu, funkční bloky, které zahrnují nejběžnější funkce se kterými přichází reálné PLC, mezi které patří časovače, čítače, logické a aritmetické operátory. [16]



Obrázek 45. Editor kontroléru – edice Starter [16]

3.4 Alternativy

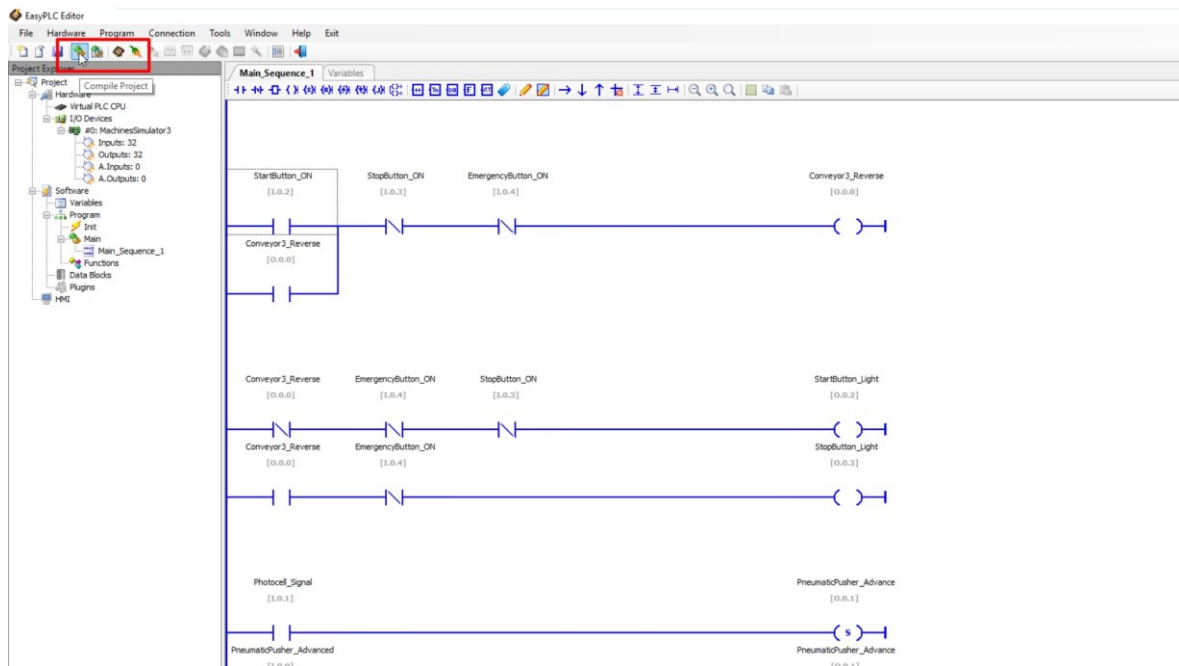
3.4.1 EasyPLC

EasyPLC je simulační a školící software pro realistickou 3D simulaci průmyslových strojů. Díky EasyPLC je možné vytvářet vlastní scénáře pro své digitální dvojče nebo je možné použít předpřipravené scény a poté jej připojit s automatizačním softwarem jako je např. TIA Portal, Rockwell Studio 500, Schneider, CodeSys nebo ostatní systémy akceptující OPC DA/UA nebo Modbus protokoly. EasyPLC podporuje následující programovací jazyky: žebříčkový diagram, sekvenční diagram (používající Graftec), funkční blok a instrukční list. K programování sekvencí je také možné používat jazyk C#, který umožňuje přístup ke všem prvkům PLC (vstupy, výstupy, proměnné, čítače, časovače, datové bloky a další), dále také k ovládacím prvkům HMI. Program zahrnuje následující typy proměnných: boolean, integer, long, float, double, string, čítače a časovače (standardní, on-delay, off-delay) a konfigurovatelné datové bloky. [10]



Obrázek 46. Machine simulátor [10]

Součástí EasyPLC je snadno použitelný nástroj Machine simulátor viz obrázek č. 47, díky kterému může uživatel vytvářet jednoduché, středně obtížné nebo složité mechanické systémy se sestavami mechanických elektrických a elektronických komponent. Uživatel tyto systémy může programovat v EasyPLC nebo ve svém oblíbeném softwaru (Siemens TIA Portal, Rockwell Studio 500, CodeSys, Schneider, Festo, TwinCAT, Mitsubishi,...). Uživatel může vytvářet své vlastní mechanismy. Software je dodáván s více než 50 předpřipravenými stroji. Dále je možné programovat reálné fyzické systémy ve virtuálním prostředí, což může zabránit škodám na reálných systémech. [21]



Obrázek 47. EasyPLC software [10]

Softwarová sada EasyPLC nevyžaduje použití hardwaru ani žádný jiný PLC software vyjma programovacího softwaru EasyPLC, který je možno použít se všemi hlavními značkami PLC (Rockwell Automation, Siemens, CodeSys). EasyPLC je možné používat na počítačích s operačním systémem Windows 11, Windows 10, s pracovní stanicí VMware V15 a vyšší. Počítač by měl mít minimálně 8 GB operační paměť a 64-bitový operační systém. [41]

V eshopu je možné za 85 euro koupit samostatnou licenci, která zahrnuje EasyPLC software, virtuální PLC, HMI systém a Machine Simulator. Dále je možné zakoupit USB licenci, Tutorial Manager, Machines Simulator VR, USB licenci s kartou rozhraní. Všechny licence jsou doživotní. [30]

II. PRAKTICKÁ ČÁST

4 NÁVRH VIRTUÁLNÍ VÝROBNÍ LINKY VE FACTORY I/O

Při návrhu virtuální výrobní linky bylo postupováno tak, aby byly splněny všechny zadané cíle bakalářské práce. Praktická část se zaměřuje na vhodný výběr komponent virtuální výrobní linky, tak aby bylo možné nasimulovat běžné výrobní procesy, kterými jsou v tomto konkrétním případě CNC obrábění, montáž, balení, přeprava materiálu a jeho skladování. V kapitole č. 5 bude navrženo programové vybavení pro řídicí PLC Siemens S7-1500, které bude simulačně ověřeno v kapitole č. 6.

Cíle řízení virtuální linky, které budou v poslední kapitole bakalářské práce ověřeny, jsou konkrétně následující:

- Naskladnění všech 54 skladových pozic hotovými sestavami, které jsou uloženy v bedně na paletě, na výstup procesu (sklad).
- Před samotným naskladněním na všech 54 pozic i po něm na pracovištích a dopravnících před skladem nedojde ke kolizi tj. surový materiál, polotovary, sestavy před a po uložení do bedny, která je umístěna na paletě, budou čekat na dopravníku (u CNC i na vstupní platformě) do té doby, dokud se ve skladu neuvolní místo (funkce vyskladňování není součástí bakalářské práce).

Pro vyšší přehlednost je přiložena tabulka č. 2, ve které jsou zobrazeny procesní časy jednotlivých pracovišť, která jsou součástí virtuální výrobní linky. Přeprava objektů od začátku po konec dopravníku je ovlivněna rychlostí dopravníku 0,6 m/s, a proto přesun objektů po 2 m dlouhém dopravníku trvá přibližně 3,4 s a po 4 m dopravníku pak 6,7 s. Procesní čas naskladnění závisí na vzdálenosti skladové pozice vůči nakládacím vidlicím. Naskladnění na nejvzdálenější pozici tj. pozice č. 54 trvá přibližně 28 s. Naskladnění první sestavy od začátku spuštění simulace tj. od doby, kdy vstupní materiál projde celým procesem a hotová sestava je naskladněna na skladovou pozici č. 1 zabere 1 minutu a 54 sekund.

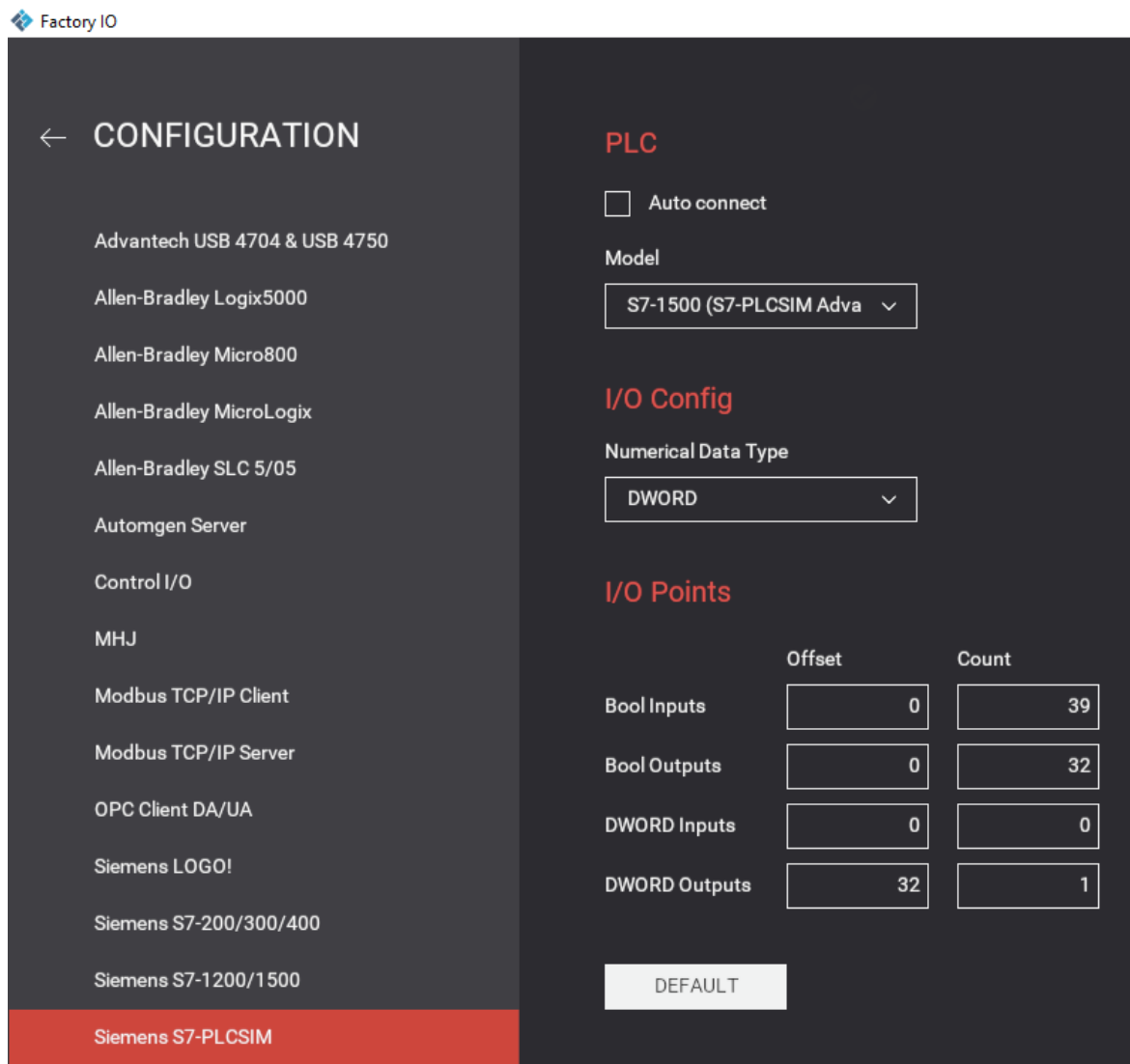
Tabulka 2. Procesní časy pracovišť

Součásti	Činnost	Operační čas (s)
CNC obrábění - základna	Výroba spodního dílu - základny	31 s
CNC obrábění - víko	Výroba horního dílu - víka	34 s
Dopravníky	Přeprava objektů po dopravníku	3,4 s - 6,7 s
Montáž	Smontování víka a základny 2-osým manipulátorem	10,5 s
Balení	Uložení sestavy 2-osým manipulátorem do bedny	9,4 s
Sklad	Naskladnění na volnou skladovou pozici	18,5 s – 28 s

Tvorbu virtuální výrobní linky v programu Factory I/O v2. 5. 2 – Ultimate Edition a programování v TIA Portal V18 bylo prováděno postupně. Prvně byla z nabídky pracovišť, které Factory I/O nabízí, vybrána scéna s názvem „Production line“. Scéna byla upravena do podoby viz obrázek č. 49. Poté byl k danému pracovišti vytvořen program včetně jeho odzkoušení. Dalším krokem bylo naprogramování dopravníků, výběr montážního pracoviště (Pick & place 1), kdy byla vybrána scéna „Assembler“ z nabídky Factory I/O, její úprava a naprogramování a ověření funkčnosti. Dalším krokem byl návrh dopravníků vedoucích k balicímu pracovišti (Pick & place 2) a skladu, návrh balicího pracoviště, programování a ověření funkčnosti. Posledním krokem byl návrh skladu, kdy byla použita scéna „Automated warehouse“. Scéna byla upravena, byl k ní vytvořen program a odzkoušen. Během tvorby programu bylo nejvíce času věnováno samotnému programování a ladění programu. Ladění programu probíhalo takovým způsobem, že byla spuštěna simulace a bylo sledováno, co program dělá, a to jak ve Factory I/O tak v TIA Portal V18. Pro snadnější pochopení funkčnosti programu budou následující kapitoly doplněny o obrázky z Factory I/O a TIA Portal V18.

4.1 Konfigurace

Vzhledem k tomu, že simulace bude spuštěna na virtuálním PLC a S7-PLCSIM Advanced V5.0 podporuje řadu S7-1500 od Siemens, tak bylo ve Factory I/O jako „Model“ zvoleno S7-1500 (S7-PLCSIM Advanced). Množství a typ vstupů a výstupů je zobrazeno na obrázku č. 48. Přehled vstupů a výstupů včetně adresy a názvu je uveden v příloze P I.



Obrázek 48. Konfigurace

4.2 Rozdělení pracovišť a součástí výrobní linky

4.2.1 CNC

Jako pracoviště CNC na začátku virtuální výrobní linky byla použita scéna z Factory I/O s názvem „Production Line“. Tato scéna v sobě obnáší níže popsané a také zobrazené na obrázku č. 49. Scéna zahrnuje 2 CNC zařízení, přičemž jedno produkuje základny a druhé pak víka (lze nastavit přímo ve Factory I/O při přepnutí aktuátoru):

- *Zakladna cnc (vyrabi),*
- *Viko cnc (vyrabi).*

Pracoviště CNC zahrnuje vstupní dopravník o délce 2 m, který je poháněn motorem. Na vstupu tohoto dopravníku je generátor vstupního surového materiálu, u kterého bylo ve Factory I/O nastaveno generování nového materiálu náhodně každé 2-3 sekundy po tom, co předchozí materiál opustí vyznačenou zónu. Na druhém konci dopravníku se nachází retro-reflexní senzor.



Obrázek 49. CNC pracoviště

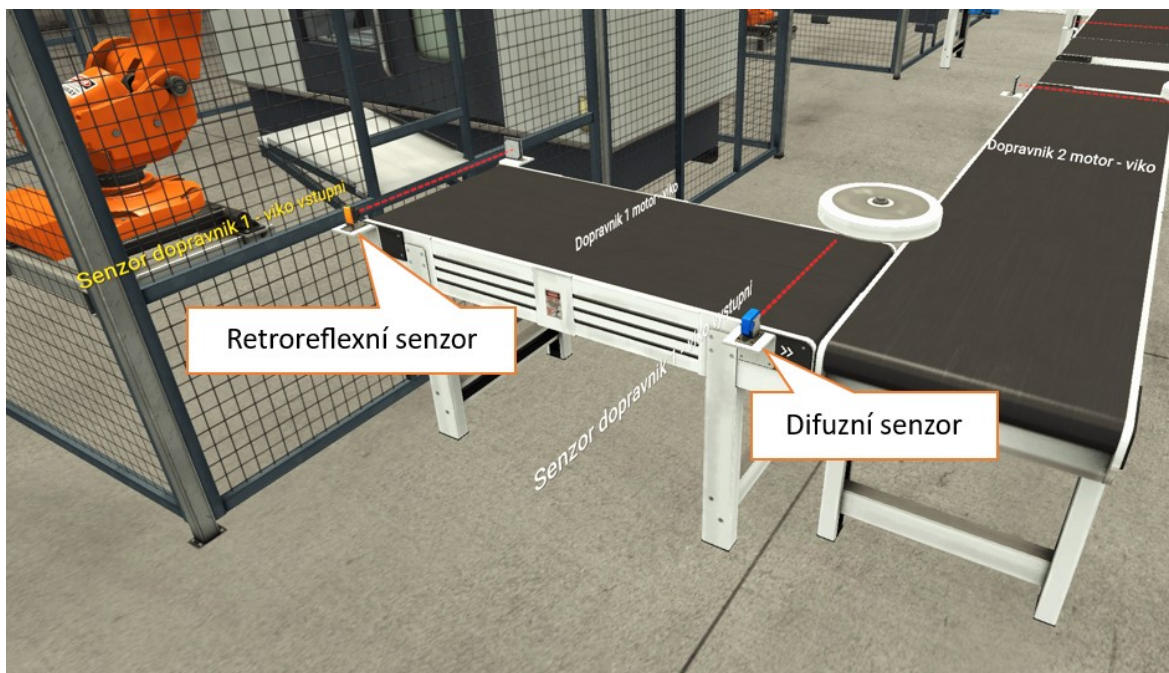
Za dopravníkem se nachází rameno s CNC zařízením. Délka obrábění nelze v programu nastavit. Dále je vývojáři Factory I/O definováno, že po aktivaci pracoviště CNC, což signalizují senzory:

- *Viko cnc (zaneprazdнено),*
- *Zakladna cnc (zaneprazdнено),*
- a oranžové signální světlo,

jsou provedeny následující kroky:

- přesun ramena robota pro vstupní materiál,
- uchopení materiálu,
- založení materiálu do CNC,
- obrobení materiálu,
- vytažení a položení víka/základny na výstupní platformu,
- návrat ramena robota do výchozí pozice.

Aktivní obrábění, je signalizováno senzory – *Viko cnc (progres)* a *Zakladna cnc (progres)*. Za pracovištěm CNC se nachází výstupní dopravník se vstupním retroreflexním a výstupním difuzním senzorem, který je v následující kapitole programován skrze Dopravníky [FB2]. Retroreflexní senzor je rozpínací a difuzní senzor naopak spínací. Oba senzory jsou zobrazeny na obrázku č. 50.



Obrázek 50. Retroreflexní a difuzní senzor

4.2.2 Dopravníky

Virtuální výrobní linka se skládá celkem z 10 pásových dopravníků a 7 válečkových dopravníků, z nichž 2 jsou nakládací a jsou fyzicky propojeny přímo se skladem. Válečkové dopravníky, kromě těch nakládacích, mají délku 4 m. Pásové dopravníky byly vybrány v délce 2 a 4 m. Všechny dopravníky mají motor k jejich pohánění, který je jejich neoddělitelnou součástí. Část linky, po které bude přepravován vstupní materiál, polotovary a smontovaný díl, je tvořena právě pásovými dopravníky. Pro druhou část byly použity válečkové dopravníky, které jsou nižší a plynule navazují na sklad.

Aby bylo možné dopravníky řídit, je nutné je osadit senzory, které dávají programátorovi informace potřebné právě pro jejich řízení. Jako senzory byly použity retroreflexní a difuzní senzory. *Dopravník 1 motor – viko*, *Dopravník 1 motor – základna* a *Dopravník 3 motor – základna* jsou vybaveny vstupním a výstupním senzorem. Ostatní dopravníky jsou vybaveny pouze jedním senzorem.

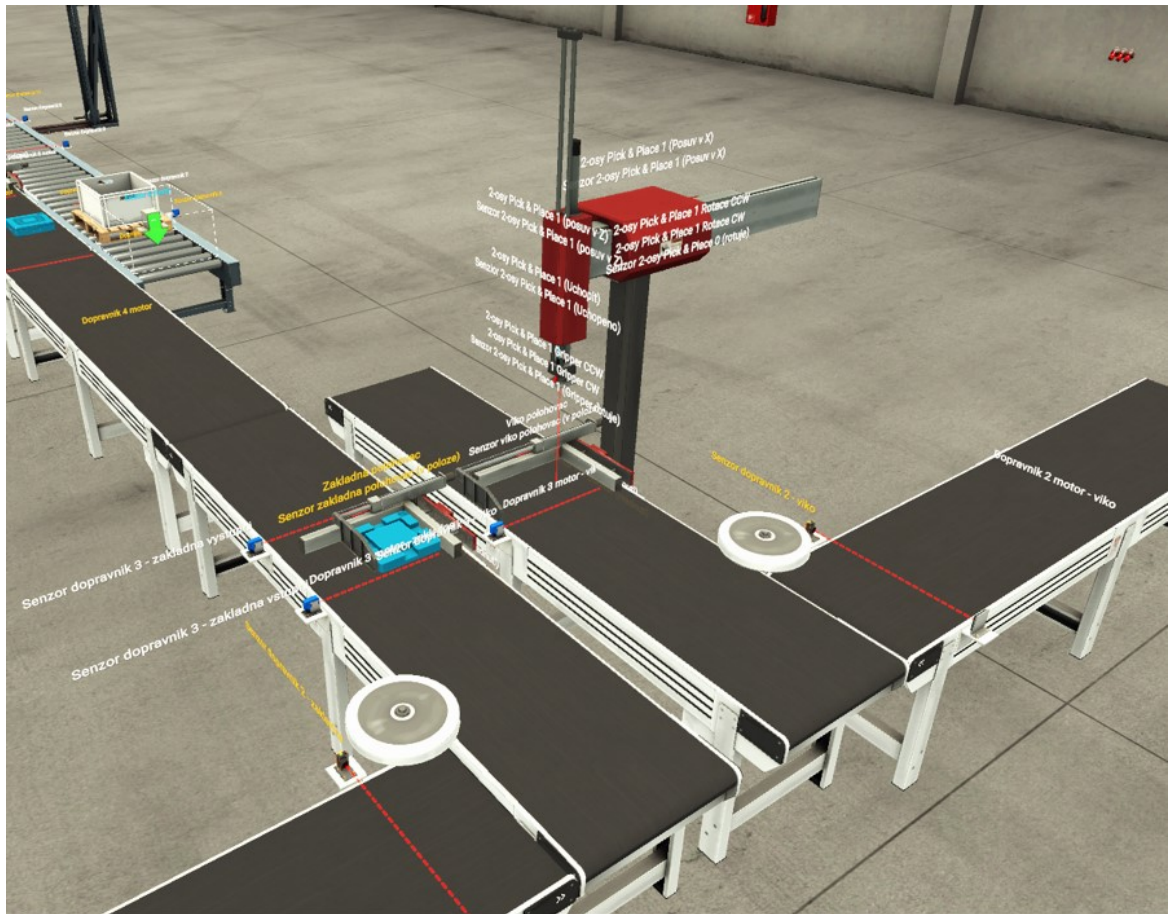
Motory a senzory dopravníků byly pojmenovány takovým způsobem, aby byla dodržena přehlednost pro následnou tvorbu programu. Vstupní dopravníky před CNC stanicí jsou pojmenovány *Dopravník motor - základna-emiter* a *Dopravník motor - viko-emiter*.

Senzory byly pojmenovány následovně: *Senzor - dopravník viko-emiter* a *Senzor - dopravník základna-emiter*. Z toho plyne, že pro senzor bylo v podstatě vždy jen přidáno slovo „senzor“ před samotný název a odebráno slovo „motor“. Pojmenování všech senzorů a aktuátorů je zobrazeno v příloze P I.

4.2.3 Montáž (Pick & place 1)

Pro pracoviště montáže byla z nabídky scén vybrána scéna s názvem „Assembler“. Tato scéna byla upravena, konkrétně z ní byly odstraněny 2 generátory vstupních dílů, protože jejich funkci bude obstarávat pracoviště CNC, které bude díly posílat právě k montážnímu pracovišti. Dále byly odstraněny skluzové dopravníky a výstupní „Removery“, protože smontovaný celek bude pokračovat po dopravnících dál k pracovišti balení. Scéna tedy obsahuje manipulátor „Two-Axis Pick & Place“, 2 dopravníky, na nichž jsou umístěny 3 difuzní senzory a 2 polohovače, které slouží k zarovnání polotovaru do montážní polohy, a také jako zarážka, která brání dílu v pohybu po dopravníku směrem dopředu.

Manipulátor se může otáčet kolem vlastní osy (po a proti směru hodinových ručiček), konat posuv v ose Z (dolů a zpět nahoru) a v ose X (dopředu a zpět dozadu do výchozí pozice). Manipulátor je vybaven gripperem, který se stará o úchop a jenž může rotovat po a proti směru hodinových ručiček. Manipulátor je vybaven senzory, aby bylo možné jej řídit. Pracoviště montáže je zobrazeno na obrázku č. 51.



Obrázek 51. Montáž (Pick & place 1)

4.2.4 Balení (Pick & place 2)

Pracoviště, na kterém probíhá uložení dílu do obalové jednotky (bedna s paletou), částečně vychází z pracoviště montáže. Dopravník, na který dojde již smontovaný díl, je obstarán difuzním senzorem, polohovačem a manipulátorem „Two-Axis Pick & Place“. Před pracoviště byly umístěny 4 válečkové dopravníky. Na první z nich byl umístěn generátor bedny s paletou, který bude po jejich opuštění vyznačené zóny náhodně generovat každých 6-7 sekund novou bednu s paletou, do níž bude manipulátorem uložen smontovaný díl. Na druhý dopravník tj. Dopravník č. 7 byl umístěn difuzní senzor přímo doprostřed. Pracoviště balení je zobrazeno na obrázku č. 52.



Obrázek 52. Balení (Pick & place 2)

4.2.5 Sklad

Jako sklad byla zvolena scéna s názvem „Automated warehouse“. Scéna nabízí 3 skladové regály s kolejovým stohovacím jeřábem, které celkem nabízí:

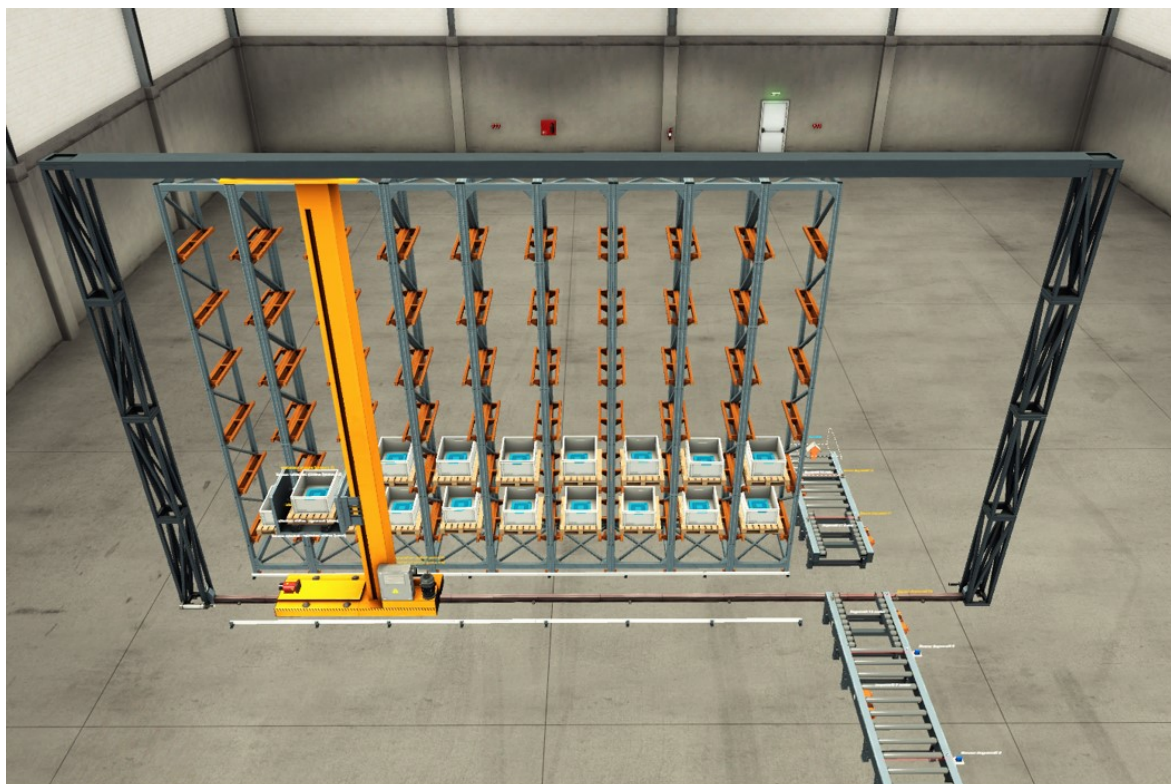
- 54 pozic k zaskladnění,
- 1 pozici, ve které se nakládací plošina zastaví na aktuální pozici (tj. pozice 0 – datový typ integer),
- a 1 výchozí pozici, do které se nakládací plošina vrátí, když je jí zadána pozice 55 a vyšší (platí pro datový typ integer) viz obrázek č. 53.

Scéna dále obsahuje nakládací plošinu se 2 vidlicemi, které se starají o nabrání a vyskladnění, 2 válečkové a 2 válečkové nakládací dopravníky s motorem a retroreflexními senzory, generátor palet s objekty, který byl smazán a „Remover“, který odebírá materiál z procesu. Součástí scény je i oplocení, které bylo smazáno a ovládací panel, který nebyl využit. Nakládací plošina s vidlicemi se může díky motorům pohybovat v ose Z a X, což indikují i její

senzory. Vidlice se mohou z výchozí pozice vysouvat vlevo (nakládka ze vstupního dopravníku), anebo vpravo (zaskladnění do regálu příp. vyskladnění ze skladu/vyskladnění na výstupní dopravník). Sklad je zobrazen na obrázku č. 54.

Position	Actuators
	Bit 0 1 2 3 4 5
Locked	000000
1	100000
2	010000
3	001000
...	...
55 (rest)	111011

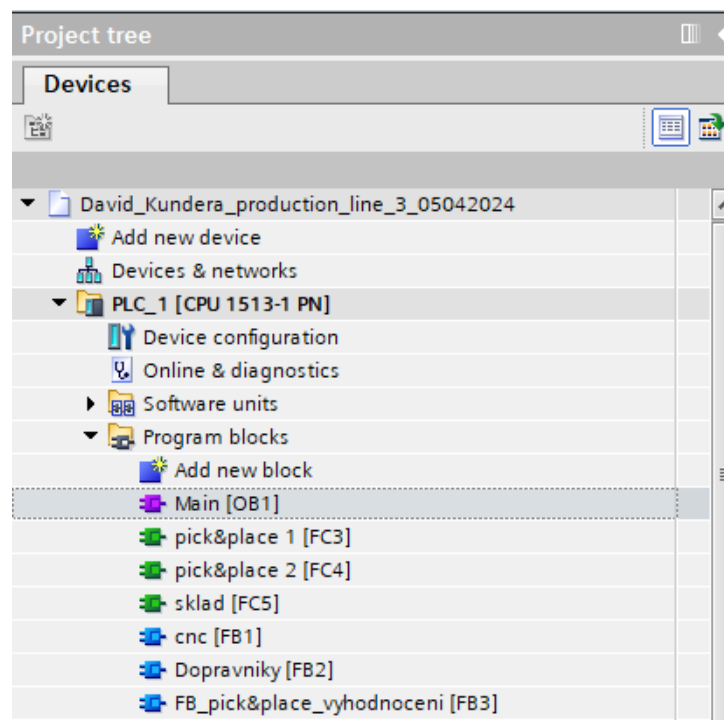
Obrázek 53. Skladové pozice [34]



Obrázek 54. Sklad

5 PROGRAMOVÁNÍ V TIA PORTAL

Při návrhu programu v TIA Portal V18 byl brán zřetel na to, že některá pracoviště a součásti výrobní linky budou použity vícekrát např. dopravníky, CNC pracoviště, pick & place (montáž a balení), a proto byly při programování použity funkční bloky a funkce. Níže bude vždy popsána programová logika jednotlivých pracovišť a součástí výrobní linky, aby bylo jasné, jakým způsobem fungují. Vše bude doplněno o obrázky, které zachycují nejpodstatnější části kódu. Na obrázku č. 55 jsou programové bloky, které byly při programování použity. Program byl tvořen takovým způsobem, aby po jeho nahrání do PLC, spuštění simulátoru a následném ověření jeho funkčnosti byly splněny cíle bakalářské práce, stanovené v předchozí kapitole.



Obrázek 55. Základní kostra programu

K použitým funkčním blokům byly po jejich přetažení do Main [OB1] následně vytvořeny instance DB. Pro CNC byl vytvořen FB, který se liší jen v přidělených vstupech a výstupech. FB pro dopravníky je vytvořen takovým způsobem, aby zohlednil mírné odlišnosti jednotlivých dopravníků, mezi něž patří typ a počet senzorů, a zdali je dopravník součástí pracoviště montáže (Pick & place 1), balení (Pick & place 2) nebo skladu. Vzhledem k odlišnostem pracovišť montáže a balení byly pro tato 2 pracoviště vytvořeny samostatné funkce (FC3 a FC4) a FB3 k vyhodnocení pozice manipulátoru v ose X a Z. Poslední funkcí je pak řízení skladu (FC5).

5.1 Charakteristika programu

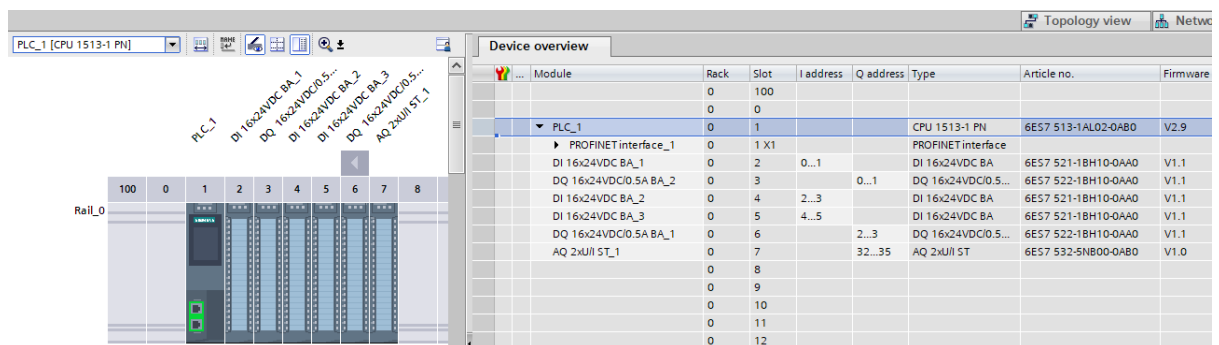
Program je tvořen v TIA Portal V18. Program zohledňuje cíle bakalářské práce, tzn. při tvorbě byla brána zřetel na to, aby byla využita požadovaná pracoviště (CNC, montáž, balení, dopravníky a sklad) a nedošlo ke kolizním stavům před a po samotném zaskladnění do všech 54 skladových pozic. To znamená, že bylo nutné navrhnout a odladit takový program, kdy jednotlivá pracoviště si mezi sebou posílají signály, aby bylo jasné, na kterém z nich se právě nachází díl a případně, zdali probíhá montáž, balení či zaskladňování. Nad rámec cílů bakalářské práce byla vytvořena funkce vyskladnění. Program je naprogramován v LD a ST.

5.2 Vytvoření projektu

Projekt je vytvořen v TIA Portal V18. Při jeho založení bylo postupováno podle kroků zmíněných v teoretické části viz kapitola 2.3.1. Po vytvoření projektu byla provedena konfigurace HW viz kapitola níže a nastavení IP adresy přesně dle kapitoly 2.3.2 z teoretické části.

5.3 Zvolení HW

Na obrázku č. 56 je zobrazeno zvolené CPU a karty s digitálními/analogovými vstupy a výstupy. Virtuální PLC simulátor ke svému chodu nepotřebuje přidání karet viz níže vyjma zvolení CPU. V tomto případě je nutné zvolit CPU řady S7-1500, protože jej podporuje PLCSIM Advanced V5.0. Pro tento účel byl zvolen CPU s číslem artiklu 6ES7 513-1AL02-0AB0. Karty se vstupy a výstupy jsou zde tedy zvoleny pro ukázkou, jak by v praxi bylo postupováno s jedinou výjimkou, a tou je karta AQ 2xU/I ST_1 s číslem artiklu 6ES7 532-5NB00-0AB0, která podporuje analogové výstupy datového typu INT, nicméně v simulaci je použit 1x analogový výstup z Factory I/O typu DINT, který Siemens nepodporuje.



Obrázek 56. Zvolené CPU a přídavné karty

5.4 Popis programu pro jednotlivá pracoviště

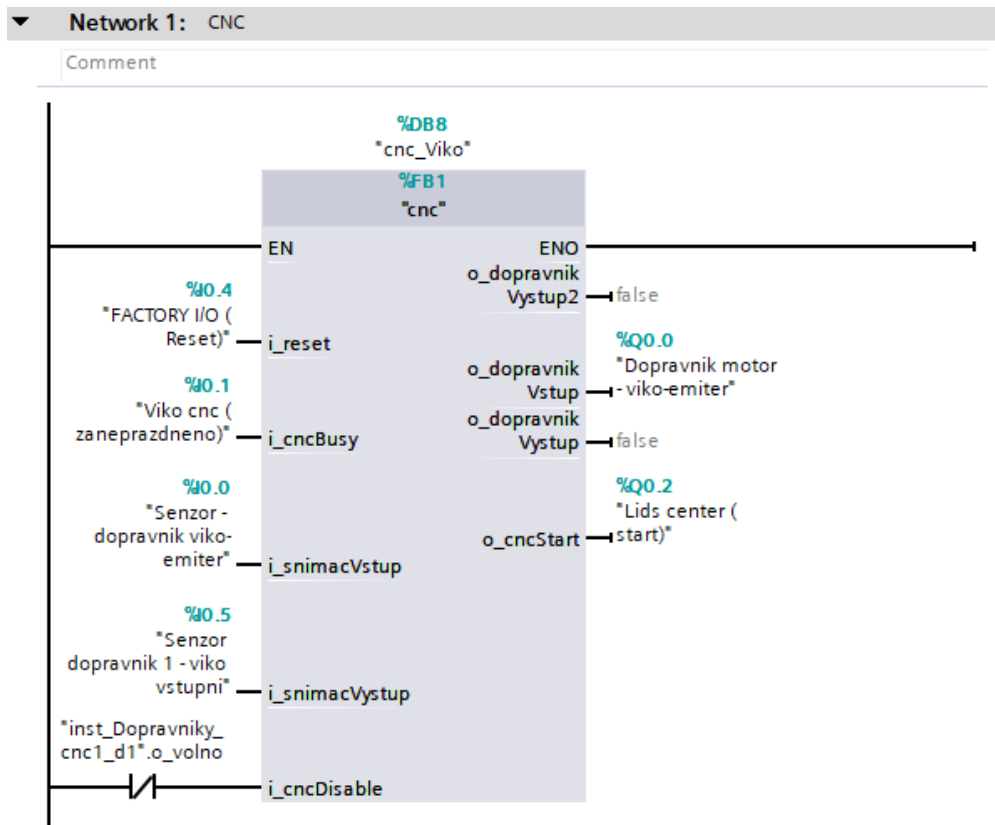
5.4.1 CNC

Na začátku virtuální výrobní linky jsou 2 totožná pracoviště s CNC zařízeními, která se liší pouze výstupním polotovarem. Z tohoto důvodu jsem pro ně v programu TIA Portal v18 vytvořil funkční blok „cnc“ [FB1]. Při přetažení tohoto FB do Main [OB1] byly vytvořeny instance DB8 a DB9. Použité proměnné ve FB pro CNC tj. FB1 jsou zobrazeny na obrázku č. 57.

Programová logika je taková, že po spuštění simulace tj. kliknutí na tlačítko "FACTORY I/O (Reset)" dojde ke spuštění vstupního dopravníku (*o_dopravnikVstup*), následně pozitivní náběžná hrana pošle po přejetí surového materiálu přes vstupní senzor (*i_snimacVstup*) signál, který spustí CNC (*o_cncStart*) a resetuje vstupní dopravník. Jakmile centrum provádí svou práci (*i_cncBusy* je v logické 1), je vyresetována proměnná *o_cncStart*. Jakmile centrum dokončí svou práci (*i_cncBusy* je v logické 0), dojde skrze negativní náběžnou hranu opět ke spuštění motoru vstupního dopravníku. CNC nezahájí svou práci, jestliže dopravník za ním nemá volno (*#o_volno* z instance DB pro FB2 Dopravníky) viz proměnná *i_cncDisable* na obrázku č. 58.

	Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Input							
2	<i>i_reset</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	<i>i_cncBusy</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	<i>i_snimacVstup</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	<i>i_snimacVystup</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	<i>i_cncDisable</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	Output							
8	<i>o_dopravnikVystup2</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	<i>o_dopravnikVstup</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	<i>o_dopravnikVystup</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	<i>o_cncStart</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	InOut							
13	<Add new>							
14	Static							
15	<i>trig1</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	<i>trig2</i>	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17	<i>timer_dopVystup</i>	TOF_TIME		Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	Temp							
19	<Add new>							
20	Constant							
21	<Add new>							

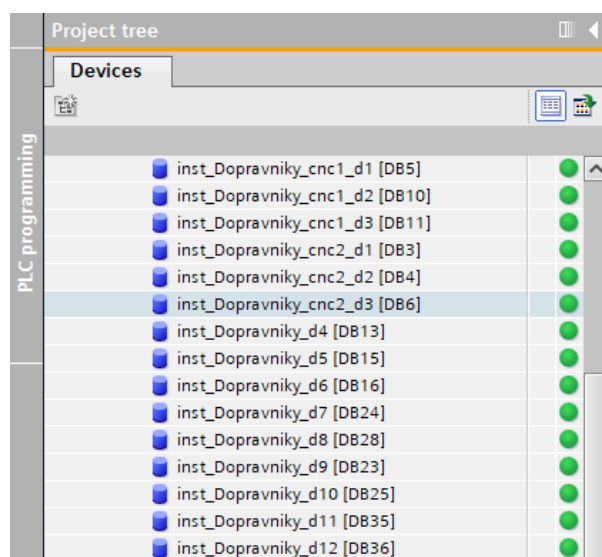
Obrázek 57. Proměnné pro FB



Obrázek 58. Přiřazení vstupů a výstupů v OB1

5.4.2 Dopravníky

Tím, že dopravníků je v projektu celkem 17, byl vytvořen pro jejich řízení FB2 „Dopravníky“, který bude vyjma 2 dopravníků, na nichž se generuje vstupní surový materiál a které jsou umístěny před pracovišti CNC, použit pro zbylých 15 dopravníků viz obrázek č. 59.



Obrázek 59: Instance DB pro dopravníky

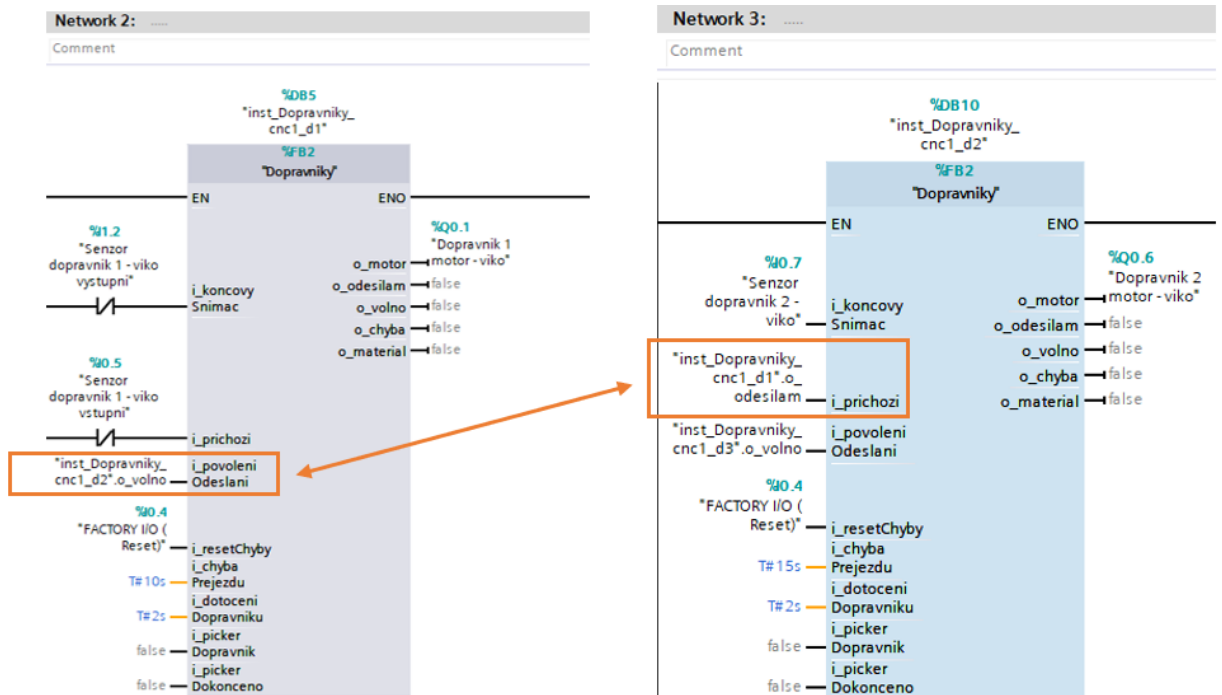
Na obrázku č. 60 jsou zobrazeny proměnné, do kterých se budou ukládat data pro řízení dopravníků.

	Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	Input							
2	i_koncovySnimac	Bool	false	Non-ret...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	i_prichozi	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	i_povoleniOdeslani	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	i_resetChyby	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	i_chybaPrejezdu	Time	T#0ms	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	i_dotoceniDopravniku	Time	T#0ms	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	i_pickerDopravnik	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	i_pickerDokonceno	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	Output							
11	o_motor	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	o_odesilam	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13	o_volno	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	o_chyba	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
15	o_material	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16	InOut							
17	<Add new>							
18	Static							
19	prichozi	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	timer_err	TON_TIME		Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	timer_prejezd	TOF_TIME		Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	Temp							
23	<Add new>							
24	Constant							
25	<Add new>							

Obrázek 60. Proměnné vytvořené pro FB2 „Dopravniky“

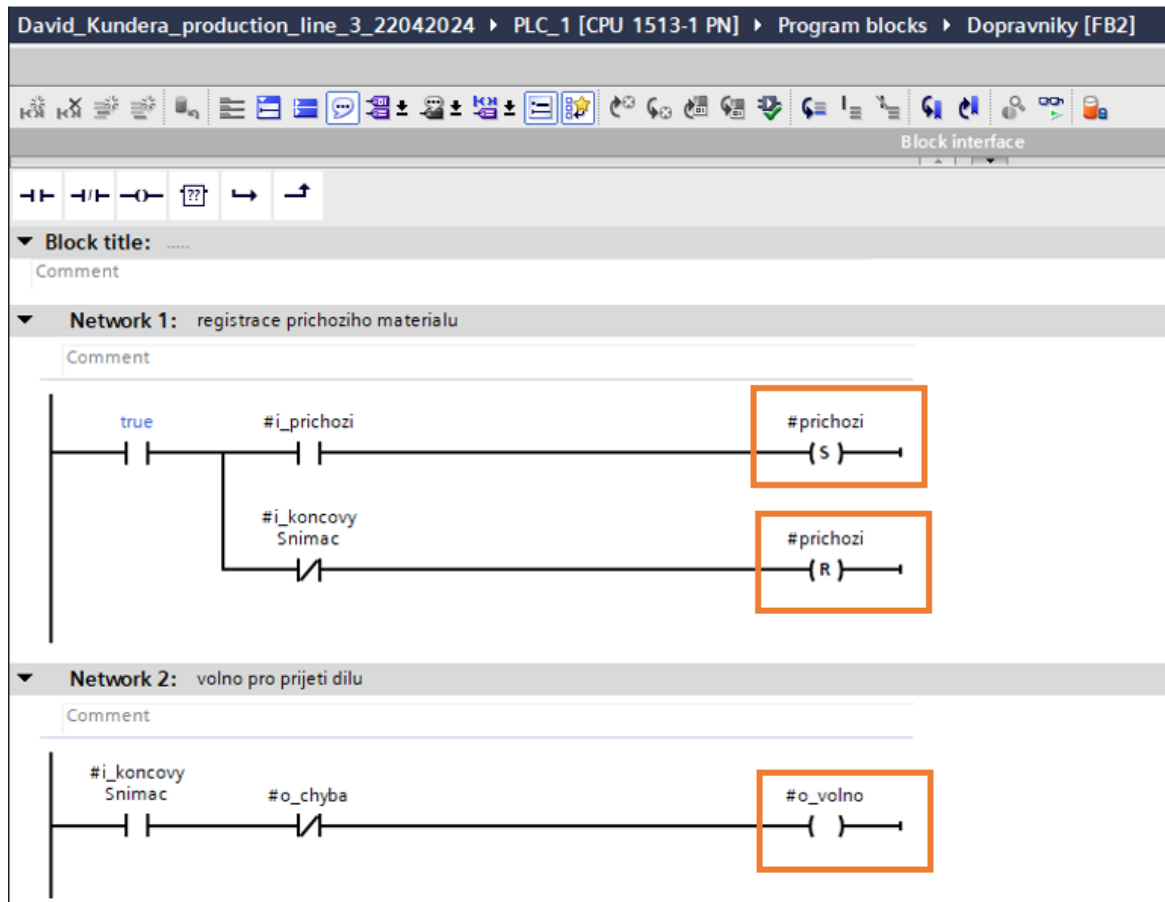
FB je vytvořen za tím účelem, aby realizoval činnosti spojené s přepravou materiálu. Podstatou FB je ta, že dopravníky si mezi sebou posílají následující signály: volno, registrace přichozího materiálu, povolení k odeslání a dále je rozlišováno, zdali se jedná o dopravník spojený s pracovišti montáže či balení. Nejpodstatnější částí FB2 „Dopravniky“ jsou popsány níže včetně příložených vizualizací.

Na obrázku č. 61 jsou zobrazeny vybrané instance DB5 a DB10 pro FB2 „Dopravniky“. Jak je z obrázku patrné, druhý dopravník prvnímu posílá informaci, zdali má volno (*o_volno*). Tím pádem pokud na koncovém senzoru má předchozí dopravník materiál a má povolení k odeslání (*i_povoleniOdeslani*), tak odesílá následujícímu dopravníku materiál.



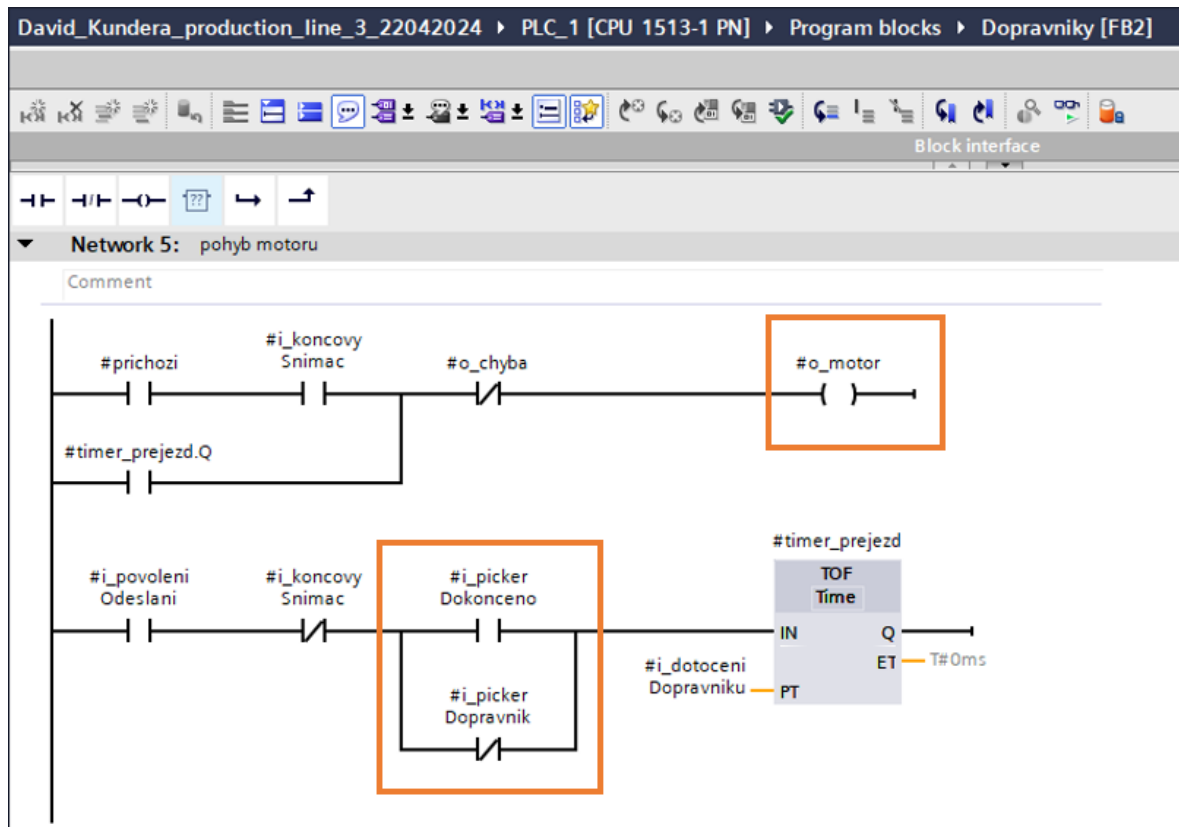
Obrázek 61. Ukázka komunikace mezi dopravníky

Pokud tedy předchozí dopravník odesílá následujícímu materiál, je nastavena pomocná proměnná *#prichozi* na následujícím dopravníku do logické 1. Jakmile materiál na dopravníku změní logický stav koncového senzoru (*#i_koncovySnimac*), dojde k resetu pomocné proměnné *#prichozi*. Dopravník hlásí, že má volno (*#o_volno*) pokud nemá materiál na svém koncovém senzoru (*i_koncovySnimac*) viz obrázek č. 62.



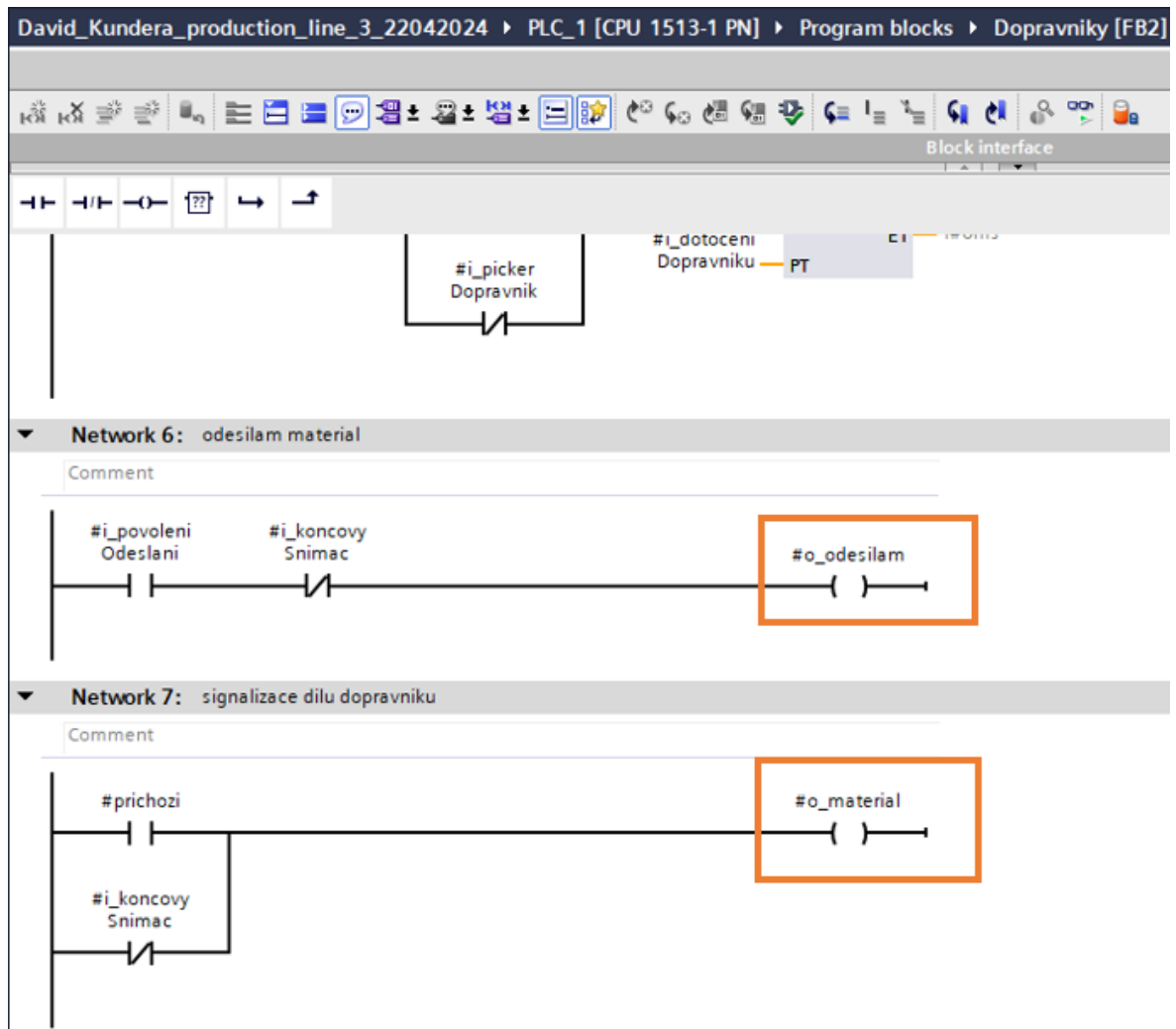
Obrázek 62. Network 1 a 2 – FB2 „Dopravniky“

Na obrázku č. 63 jsou podmínky pro spuštění motoru dopravníku ($\#o_motor$). Pokud předcházející dopravník posílá materiál ($\#prichozi$ je v logické 1) a nadcházející dopravník nemá materiál na svém koncovém senzoru nebo je $\#timer_prejezd.Q$ v logické 1, tak je spuštěn motor dopravníku.



Obrázek 63. Network 5 – FB2 „Dopravníky“

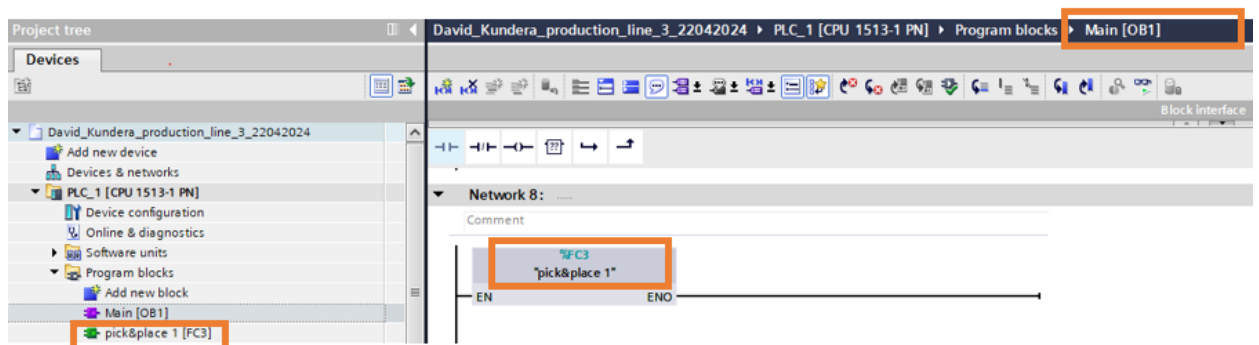
Networkem 6 viz obrázek č. 64 je poskytována následujícímu dopravníku informace o tom, že je na něj odesílán materiál (*#o_odesilam*). Network 7 dává informaci (*#o_material*), že na sobě má dopravník materiál nebo mu je právě odesílán předchozím dopravníkem. Níže na obrázku je ošetřeno spouštění motoru i pro dopravníky, které jsou součástí pracoviště montáže (Pick & place 1) a balení (Pick & place2), kdy je nutné odeslat materiál až po dokončení montáže a balení (*#i_pickerDokonceno*). Časovač TOF (*#timer_prejezd*) obstarává „dotočení“ dopravníku, které je u dopravníků 2 nebo 4 s, kdy je potřeba poslat z koncového senzoru materiál na následující dopravník.



Obrázek 64. Network 6 a 7 – FB2 „Dopravniky“

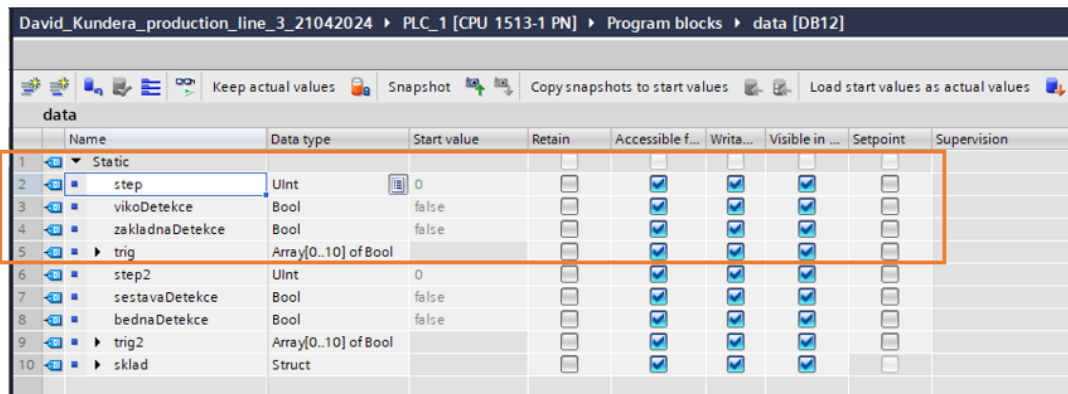
5.4.3 Montáž (Pick & place 1)

Pro montáž horního dílu „víka“ a spodního dílu „základny“ byla vytvořena funkce FC3, která byla vložena do Main [OB1] viz obrázek č. 65.



Obrázek 65. Pick & place 1 – FC3

V datech [DB12] byly vytvořeny statické proměnné pro použití ve funkci viz obrázek č. 66.



	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision
1	Static								
2	step	UInt	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	vikoDetekce	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	zakladnaDetekce	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	trig	Array[0..10] of Bool		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	step2	UInt	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	sestavaDetekce	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	bednaDetekce	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	trig2	Array[0..10] of Bool		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	sklad	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Obrázek 66. Pick & place 1 - proměnné

Dále byla vytvořen FB_pick&place_vyhodnoceni [FB3], který slouží k tomu, aby program věděl, jestli je posuv manipulátoru v ose X a Z v pracovní, nebo základní pozici viz obrázek č. 67.

David_Kundera_production_line_3_21042024 ▶ PLC_1 [CPU 1513-1 PN] ▶ Program blocks ▶ FB_pick&place_vyhodnoceni [FB3]

Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision
1 ▶ Input								
2 ▶ povel	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3 ▶ pohyb	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4 ▶ Output								
5 ▶ zakladniPozice	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6 ▶ pracovniPozice	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7 ▶ InOut								
8 ▶ <Add new>								
9 ▶ Static								
10 ▶ trig	Array[0..5] of Bool		Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11 ▶ timer_dojezd_pracov...	TON_TIME		Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12 ▶ timer_dojezd_zakladn...	TON_TIME		Non-ret...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Comment

Network 1: Vyhodnoceni zakladni pozice

Comment

Network 2: Vyhodnoceni pracovni pozice

Comment

Obrázek 67. Network 1 a 2 – FB3

Ke zmíněnému FB3 byla vytvořena instance DB7 a DB14, aby bylo možné ukládat data viz obrázek č. 68.

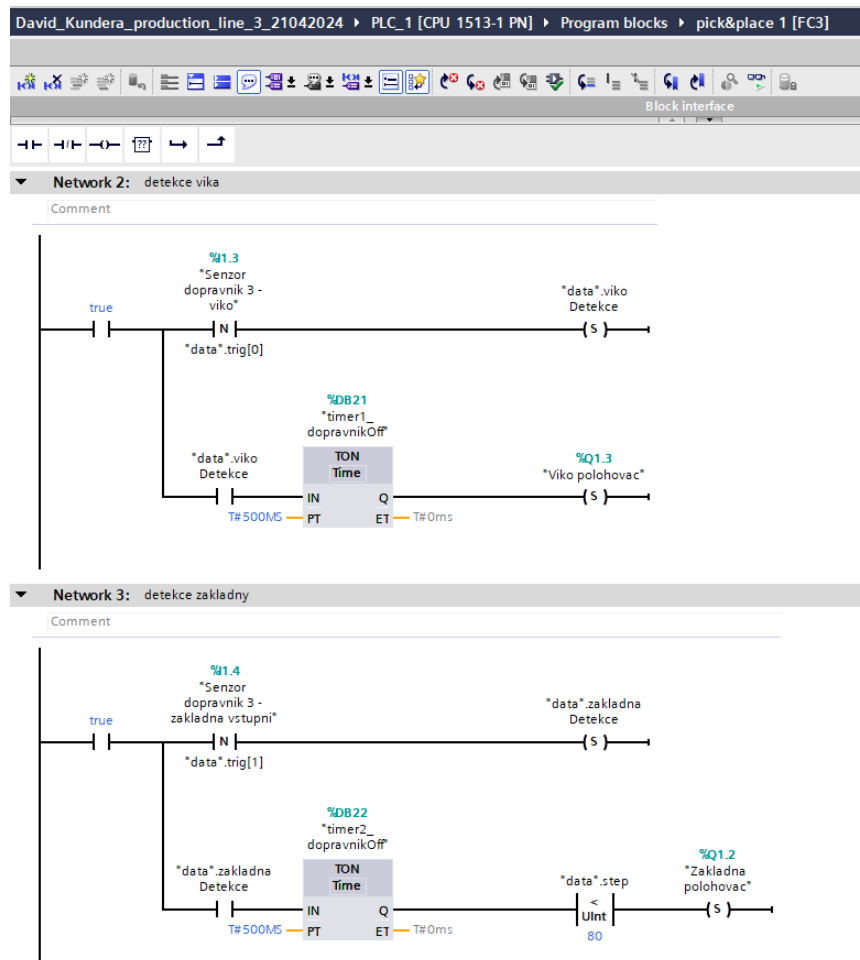
Project tree

Devices

- inst_pick&place_vyhodnoceni X_2 DB [DB18]
- inst_pick&place_vyhodnoceni X_DB [DB7]
- inst_pick&place_vyhodnoceni Z_2 DB [DB17]
- inst_pick&place_vyhodnoceni Z_DB [DB14]

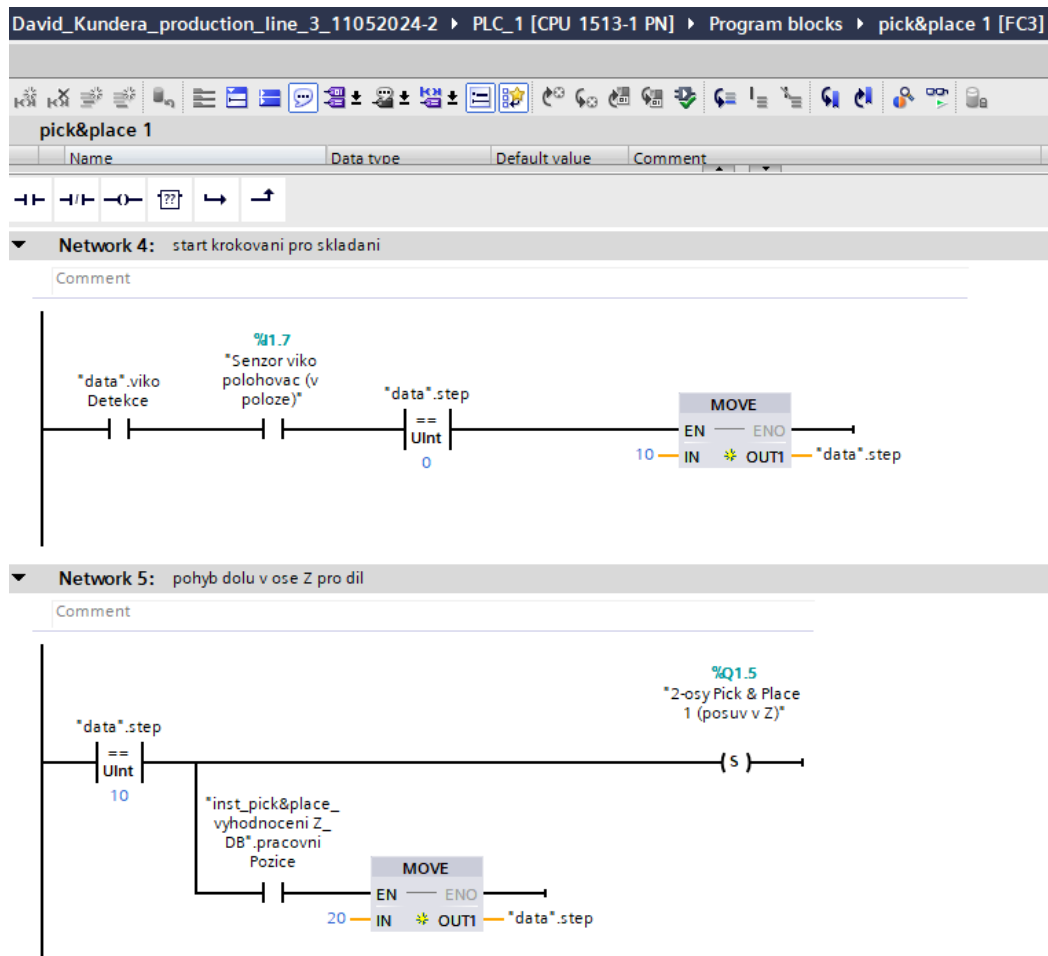
Obrázek 68. Instance DB pro Pick & place 1 a 2

Princip funkce je takový, že po detekování negativní náběžné hrany na vstupním senzoru dopravníku 3 (platí pro základnu i víko), dojde k nastavení proměnné *vikoDetekce* a *zakladnaDetekce*. Následně po 500 ms, což je dáno časovačem TON dojde k „napolohování“ obou polotovarů do montážní pozice viz obrázek č. 69.



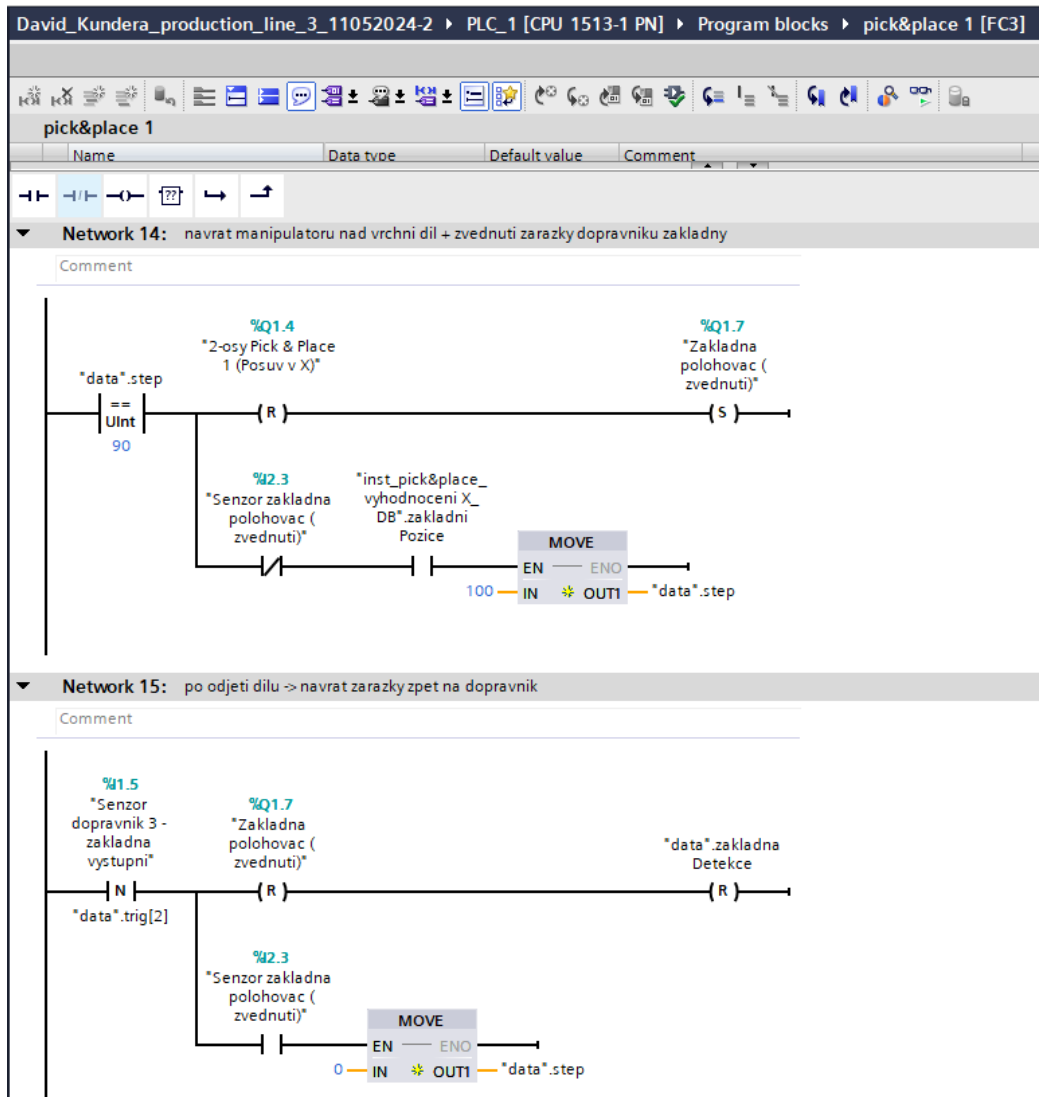
Obrázek 69. Network 2 a 3 – Pick & place 1

Následně dojde ke startu krokování skrze instrukci „Move“, přičemž hodnota kroku je ukládána v proměnné „data“.step. Ve funkci jsou kroky nastaveny od kroku 0 po krok s hodnotou 100. Z hlediska montáže dojde chronologicky k posuvu manipulátoru v ose Z dolů, uchopení dílu, resetu polohovače pro víko, resetu proměnné vikoDetekce, posuvu manipulátoru v ose Z do základní pozice, posuvu manipulátoru v ose X do pracovní pozice, posuvu manipulátoru v ose Z do pracovní pozice, resetu uchopení víka, resetu posuvu manipulátoru v ose Z, resetu polohovadla základny, resetu posuvu manipulátoru v ose X a zvednutí polohovadla, aby byla sestava připravena na opuštění dopravníku.



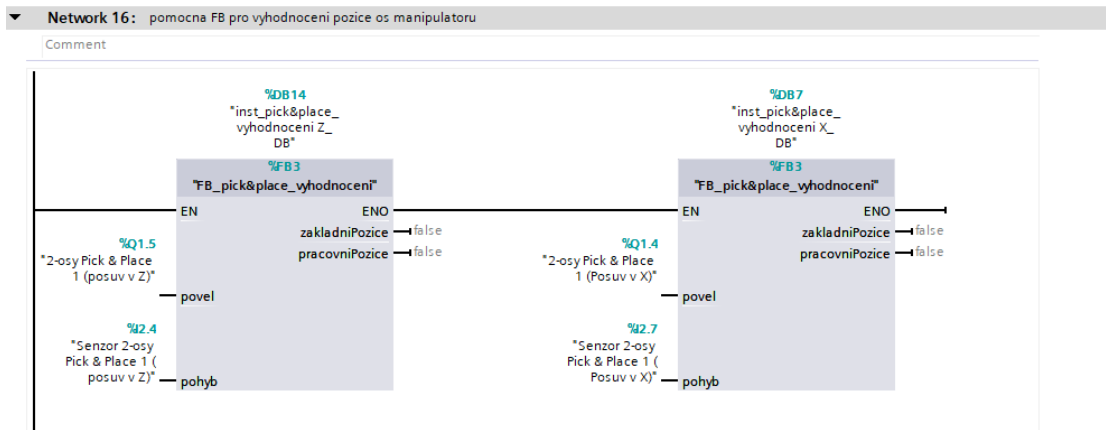
Obrázek 70: Network 4 a 5 - Pick & place 1

Po odjetí dílu z dopravníku, což je indikováno negativní hranou (*Senzor dopravník 3 – základna vystupni*), dojde k resetu polohovadla, tj. polohovadlo se vrátí v ose Z na dopravník a slouží jako zarážka, která nepustí další díl na následující dopravník. Dále dojde k resetu proměnné *zkladnaDetekce* a krokování se vrací zpět do kroku 0 viz obrázek č. 71.



Obrázek 71. Network 14 a 15 – Pick & place 1

Součástí funkce jsou i obě instance DB14 a DB8 viz obrázek níže, které poskytují data o pozici, ve které je aktuálně manipulátor, co se týče osy X a Z viz obrázek č. 72.



Obrázek 72. Network 16 – Pick & place 1

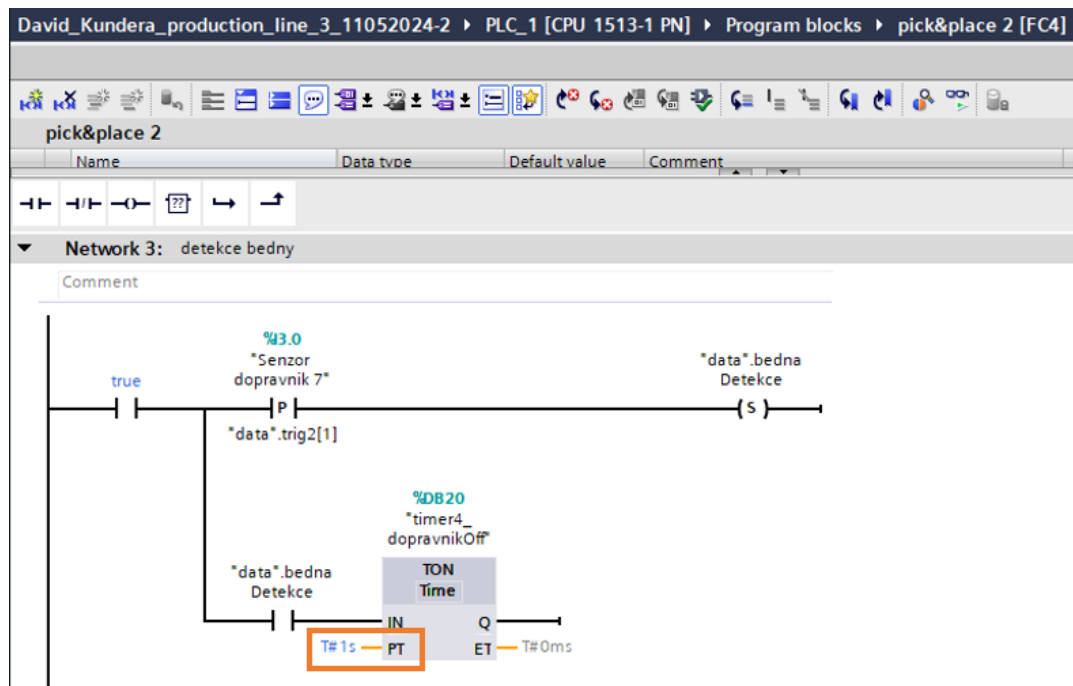
5.4.4 Balení (Pick & place 2)

Pro operaci balení byla vytvořena FC4 s názvem „pick & place 2“. Rozdíl oproti FC3 je ten, že bedna je detekována pozitivní hranou viz obrázek č. 74 a signál „*timer4_dopravnikOff*“ je dále v kódu použit pro DB 24 "inst_Dopravniky_d7". Statické proměnné pro tuto funkci jsou zobrazeny na obrázku č. 73.

	Name	Data type	Start value	Retain	Accessible f...	Write...	Visible in ...	Setpoint	Supervision
1	Static								
2	step	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	vikoDetekce	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	zakladnaDetekce	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	trig	Array[0..10] of Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	step2	UInt	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	sestavaDetekce	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	bednaDetekce	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	trig2	Array[0..10] of Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	sklad	Struct			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Obrázek 73. Pick & place 2 - proměnné

Krokování pro tuto funkci je opět od kroku s hodnotou 0 po krok s hodnotou 100. Oproti pracovišti montáže je zde pouze jedno polohovadlo a není potřeba řešit jeho posuv v ose Z nahoru a následně dolů, protože z dopravníku č. 5 již žádná sestava neodjíždí.

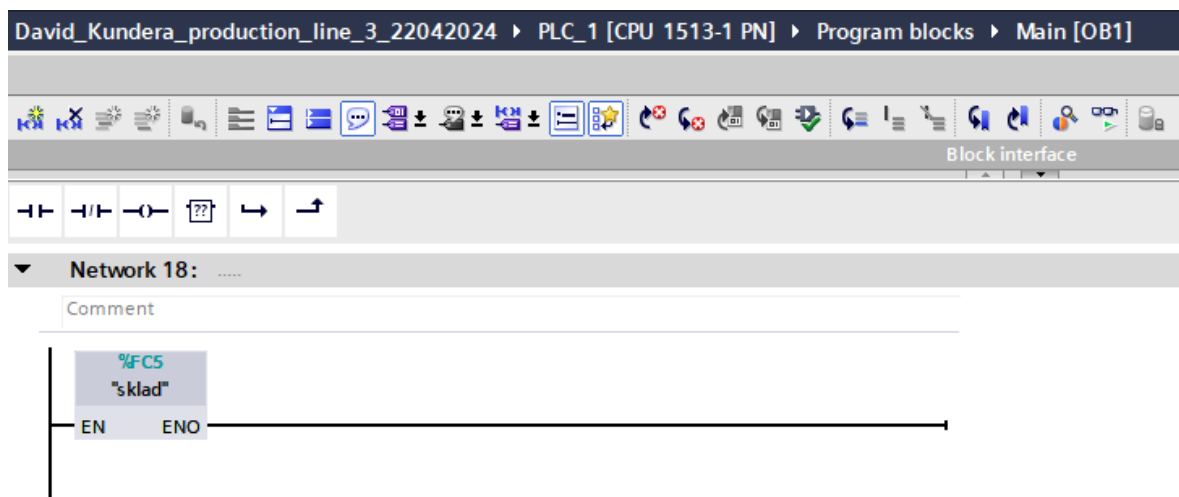


Obrázek 74. Network 3 – FC4

Funkce taktéž využívá FB3 pro vyhodnocení pracovní a základní pozice. Pro toto vyhodnocení byly vytvořeny instance DB17 a DB18 viz obrázek č. 68 výše.

5.4.5 Sklad

Pro naskladnění do skladu byla vytvořena FC5 viz obrázek č. 75. Jedním z cílů bakalářské práce je naskladnění hotových sestav na všechny skladové pozice tj. celkem 54 pozic. Nad rámec bakalářské práce byla vytvořena i možnost vyskladnění hotových sestav ze skladu. Kroky vedoucí k naložení, naskladnění a vyskladnění hotové sestavy jsou opět řešeny pomocí instrukce „MOVE“ a časovačů, podobně jako tomu je u pracoviště montáže a balení. Krokování a použití časovačů je důležité proto, aby nedošlo při obsluze skladu k nehodám.



Obrázek 75. FC5 „sklad“

Na obrázcích č. 76, 77 a 78 jsou zobrazeny proměnné použité pro FC5 „sklad“. Jak je z obrázků patrné, byly pro sklad použity i dočasné proměnné označené „temp“ a konstanty skladových pozic (*c_sklad_initPos*, *c_sklad_zaskladneni*, *c_sklad_vyskladneni*).

Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	Comment
1	Static								
2	step	UInt	0						
3	vikoDetekce	Bool	false						
4	zakladnaDetekce	Bool	false						
5	trig	Array[0..10] of Bool							
6	step2	UInt	0						
7	sestavaDetekce	Bool	false						
8	bednaDetekce	Bool	false						
9	trig2	Array[0..10] of Bool							
10	sklad	Struct							0 - zaskladnuje; 1 - vyskladnuje
11	rezim	USInt	0						
12	pozice	UInt	0						
13	poHybX_done	Bool	false						
14	poHybZ_done	Bool	false						
15	vyskladneni	Struct							
16	req	Bool	false						
17	pozice	UInt	0						
18	step	UInt	0						

Obrázek 76. Sklad proměnné

Name	Data type	Default value	Comment
1	Input		
2	<Add new>		
3	Output		
4	<Add new>		
5	InOut		
6	<Add new>		
7	Temp		
8	x	Int	
9	poziceNalezena	Bool	
10	Constant		
11	c_sklad_initPos	UInt	55
12	c_sklad_zaskladneni	USInt	0
13	c_sklad_vyskladneni	USInt	1
14	Return		
15	sklad	Void	

Obrázek 77. Sklad proměnné 2

Pro obsluhu skladu byla vytvořena proměnná typu „DT_sklad“ a řetězec k této proměnné viz obrázek č. 78. Ve zmíněné proměnné bude uloženo, zdali je na dané skladové pozici naskladněna hotová sestava či nikoliv (*obsazeno – true/false*).

David_Kundera_production_line_3_22042024 ▸ PLC_1 [CPU 1513-1 PN] ▸ Program blocks ▸ poziceSklad [DB26]									
poziceSklad									
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision
1	Static								
2	pozice	Array[0..55] ...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	pozice[0]	"DT_sklad"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	obsazeno	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	pozice[1]	"DT_sklad"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	pozice[2]	"DT_sklad"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	pozice[3]	"DT_sklad"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	pozice[4]	"DT_sklad"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	pozice[5]	"DT_sklad"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	pozice[6]	"DT_sklad"			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Obrázek 78. Sklad proměnné 3

Hlavní částí procesu naskladňování je kód napsaný v ST viz obrázek č. 79. Při žádosti o naskladnění je zapsána v „data“.sklad.step logická 1. Následně dojde k prohledání volné pozice k naskladnění, kdy je při jejím nalezení nastavena pomocná proměnná #poziceNalezena do logické 1 a je spuštěno naskladňování na volnou pozici. Pokud jsou všechny pozice obsazené, čeká se na uvolnění některé z nich.

Network 3: step 1 - zaskladneni - hledani volne pozice ve sklade

zapis cisla pozice do promenne, pokud je volna => radek 4, nastavi se pomocna promenna sklad.rezim

```

1 IF "data".sklad.step = 1 THEN
2   "data".sklad.pozice := 0;
3   FOR #x := 1 TO 54 DO
4     IF NOT "poziceSklad".pozice[#x].obsazeno THEN
5       "data".sklad.pozice := #x;
6       #poziceNalezena := true;
7       EXIT;
8     END_IF;
9   END_FOR;
10  "data".sklad.rezim := #c_sklad_zaskladneni;
11  IF #poziceNalezena THEN
12    "data".sklad.step := 10; // spusteni zaskladnovani
13  ELSE
14    "data".sklad.step := 0; //sklad je plny => cekani na volnou pozici
15  END_IF;
16 END_IF;

```

Obrázek 79. Naskladňování v ST

6 SIMULACE

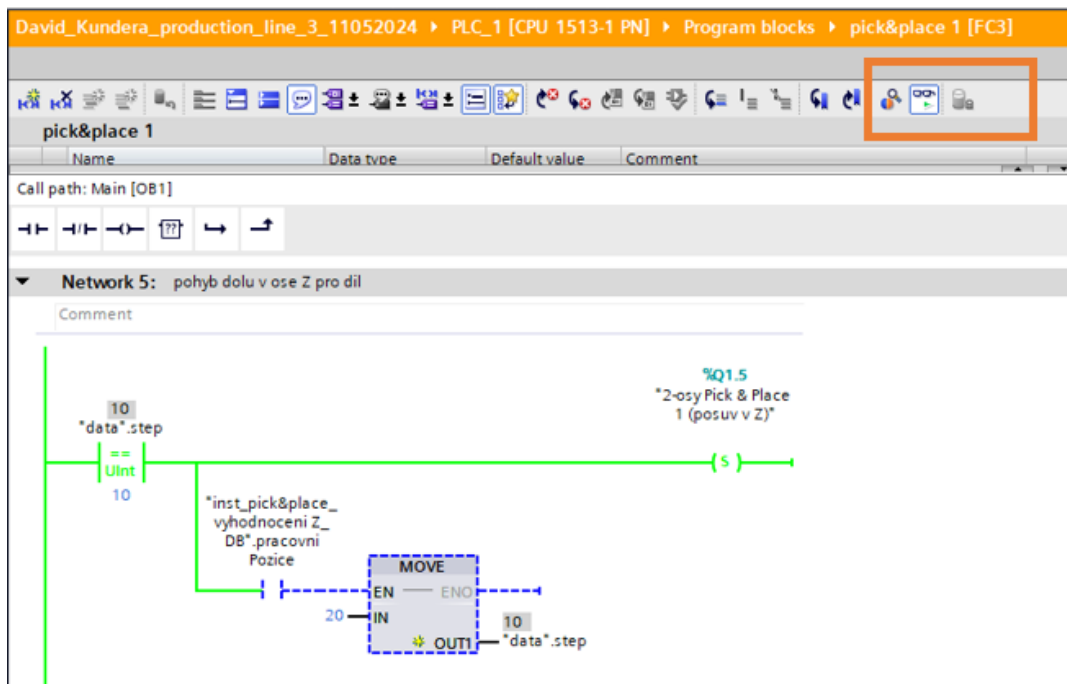
Simulace virtuální výrobní linky probíhá jak ve Factory I/O, kde lze vidět, jak probíhají jednotlivé výrobní operace, tak v TIA Portal V18, kde lze vykonávání programu pozorovat v online režimu. Celá simulace je ovládána přes tlačítka v horní liště Factory I/O (Edit/Run režim, Pauza, Restart). S tím, že po kliknutí na tlačítko „Restart“, což je i jeden ze vstupů nahraných do simulátoru s adresou %I0.4, dojde ke spuštění simulace. Níže je uvedeno zhodnocení splnění obou cílů bakalářské práce, které byly stanoveny, včetně vizualizací.

6.1 Kompilace programu a nahrání do S7-PLCSIM Advanced 5.0

Po vytvoření programu v TIA Portal byl program zkompilován a po spuštění simulátoru PLC v PLCSIM Advanced V5.0 do něj nahrán přesně takovým způsobem, jak je popsáno v kapitole č. 2.3.5.

6.2 Testování programu

Po nahrání programu do PLCSIM Advanced V5.0 byl v TIA Portal spuštěn „Run“ mód a proběhlo připojení k simulátoru ve Factory I/O. Při ladění programu bylo postupováno tak, že byla spuštěna simulace a pokud ve Factory I/O nedošlo např. k montáži víka na základnu, tak byl spuštěn online režim v TIA Portal. Následně bylo sledováno online vykonávání programu v TIA Portal viz obrázek č. 80. Program byl upravován do té doby, dokud nedošlo k odladění vedoucího ke správné funkci řídicího programu.



Obrázek 80. Testování v TIA Portal

6.2.1 Naskladnění na všech 54 skladových pozic

Jedním z cílů řízení linky je naskladnit hotové sestavy uložené v bedně, která je umístěna na paletě do všech 54 skladových pozic. Tento cíl je komplexní a pro jeho splnění je potřeba, aby byl odladěn program pro přepravu materiálu po dopravnících, obrobení, montáž a uložení sestavy do bedny, která je generována spolu s paletou. Teprve poté je možné hotovou sestavu nakládacími vidlicemi naskladnit do skladu.

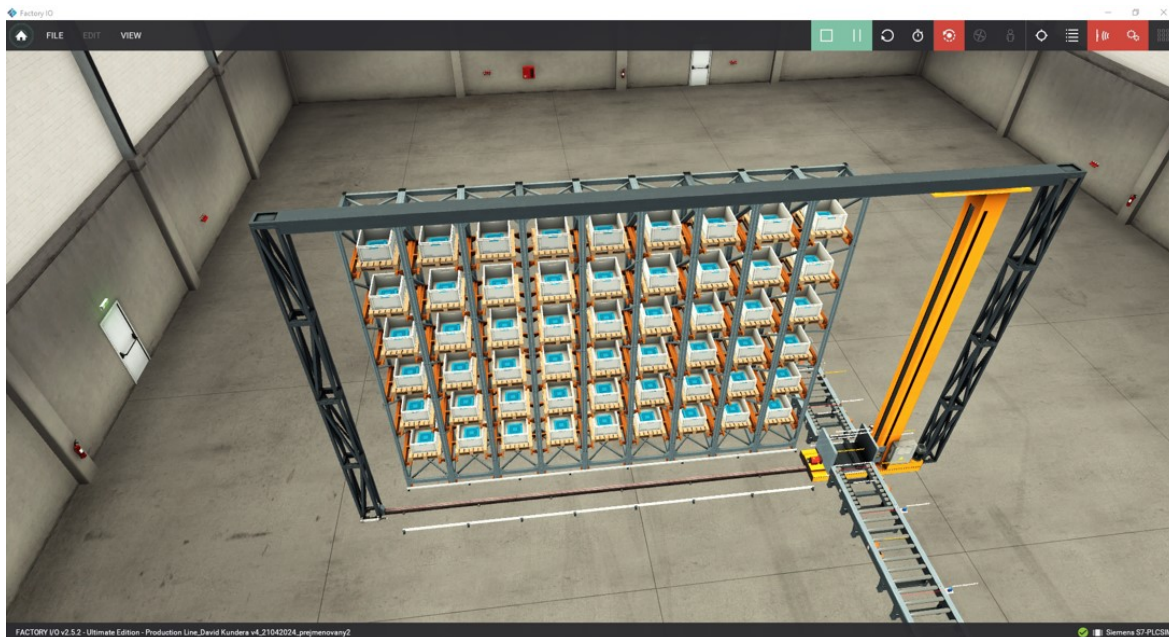
Obrázek č. 81 zobrazuje tabulku, která slouží jak k pozorování aktuálního stavu naskladnění na skladových pozicích, tak je možné skrze ni i modifikovat tzn. vepsat na danou pozici hodnotu „true“, což bude program považovat za to, že je na dané skladové pozici naskladněno. Programem došlo k naskladnění na všech 54 skladových pozic. Vzhledem k tomu, že tabulku nelze vyexportovat, je na ní zachyceno prvních 20 z nich viz sloupec „Monitor value“.

David_Kundera_production_line_3_11052024 ▶ PLC_1 [CPU 1513-1 PN] ▶ Watch and force tables ▶ Watch table_1

i	Name	Address	Display format	Monitor value	Modify value		Comment
1	"poziceSklad".pozice[1]					<input type="checkbox"/>	
2	"poziceSklad".pozice[1].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
3	"poziceSklad".pozice[2]					<input type="checkbox"/>	
4	"poziceSklad".pozice[2].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
5	"poziceSklad".pozice[3]					<input type="checkbox"/>	
6	"poziceSklad".pozice[3].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
7	"poziceSklad".pozice[4]					<input type="checkbox"/>	
8	"poziceSklad".pozice[4].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
9	"poziceSklad".pozice[5]					<input type="checkbox"/>	
10	"poziceSklad".pozice[5].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
11	"poziceSklad".pozice[6]					<input type="checkbox"/>	
12	"poziceSklad".pozice[6].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
13	"poziceSklad".pozice[7]					<input type="checkbox"/>	
14	"poziceSklad".pozice[7].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
15	"poziceSklad".pozice[8]					<input type="checkbox"/>	
16	"poziceSklad".pozice[8].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
17	"poziceSklad".pozice[9]					<input type="checkbox"/>	
18	"poziceSklad".pozice[9].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
19	"poziceSklad".pozice[10]					<input type="checkbox"/>	
20	"poziceSklad".pozice[10].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
21	"poziceSklad".pozice[11]					<input type="checkbox"/>	
22	"poziceSklad".pozice[11].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
23	"poziceSklad".pozice[12]					<input type="checkbox"/>	
24	"poziceSklad".pozice[12].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
25	"poziceSklad".pozice[13]					<input type="checkbox"/>	
26	"poziceSklad".pozice[13].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
27	"poziceSklad".pozice[14]					<input type="checkbox"/>	
28	"poziceSklad".pozice[14].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
29	"poziceSklad".pozice[15]					<input type="checkbox"/>	
30	"poziceSklad".pozice[15].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
31	"poziceSklad".pozice[16]					<input type="checkbox"/>	
32	"poziceSklad".pozice[16].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
33	"poziceSklad".pozice[17]					<input type="checkbox"/>	
34	"poziceSklad".pozice[17].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
35	"poziceSklad".pozice[18]					<input type="checkbox"/>	
36	"poziceSklad".pozice[18].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
37	"poziceSklad".pozice[19]					<input type="checkbox"/>	
38	"poziceSklad".pozice[19].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	
39	"poziceSklad".pozice[20]					<input type="checkbox"/>	
40	"poziceSklad".pozice[20].obsazeno		Bool	TRUE	TRUE	<input checked="" type="checkbox"/>	

Obrázek 81. Online režim - naskladňování na skladové pozice

Obrázek č. 82 zobrazuje plný sklad, tzn. na všech 54 skladových pozicích je naskladněna hotová sestava, která je uložena v bedně na paletě.

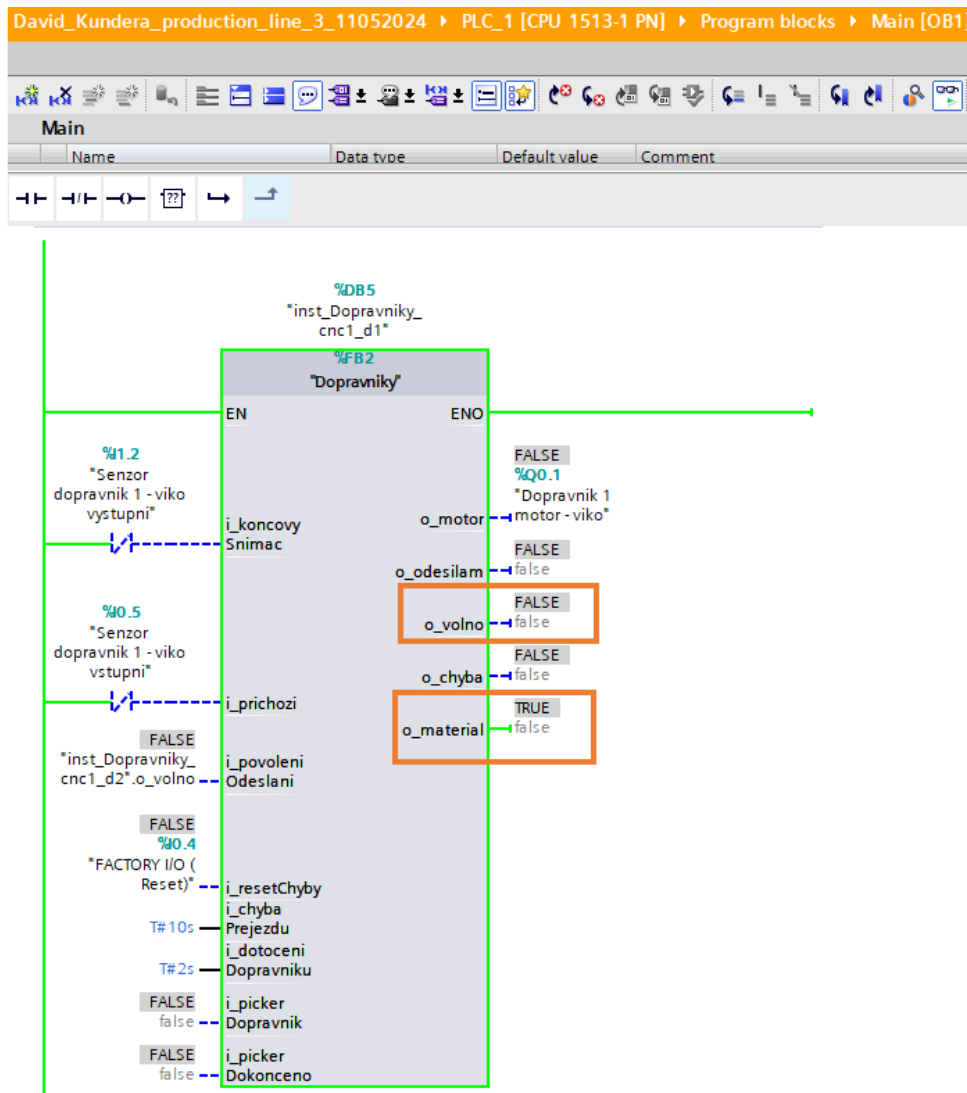


Obrázek 82. Plně obsazený sklad

6.2.2 Bezkolizní stav

Druhým cílem řízení je, aby před samotným naskladněním na všech 54 pozic i po něm na pracovištích a dopravnících před skladem nedošlo ke kolizi. Níže bude znázorněno tak, že surový materiál, polotovary, sestavy před a po uložení do bedny, která je umístěna na paletě, budou čekat na dopravníku (u CNC i na vstupní platformě) do té doby, dokud se ve skladu neuvolní místo. Jak již bylo zmíněno výše, počínaje dopravníky, které jsou na výstupu CNC pracovišť až po samotný sklad tj. celkem 13 dopravníků mezi sebou komunikuje díky vytvořeným instancím DB pro FB2 (pozn. 2 dopravníky za skladem využívají také instanci DB pro FB2).

Pokud dopravník nemá volno (proměnná $o_volno = „false“$), tak nemůže přijmout materiál z předešlého dopravníku viz obrázek č. 83. Obě pracoviště CNC nezahájí práci, pokud bude na výstupním dopravníku za nimi materiál (proměnná $o_material = „true“$). Pokud CNC nezahájí svou práci tzn. nebude proměnná $i_cncBusy = „true“$, tak vstupní dopravník před CNC nedopraví surový materiál k CNC. Pokud je sklad plný, tak nedojde k naskladnění sestavy, která čeká na koncovém senzoru dopravníku č. 10.



Obrázek 83. Instance DB5 pro FB2

Splnění výše zmíněných podmínek je pro lepší přehlednost znázorněno v tabulce č. 3, která zobrazuje, kde bude po kompletním naskladnění čekat vstupní materiál, polotovary, hotová sestava, bedna s paletou či hotová sestava umístěná v bedně na paletě. Jak je patrné, na každém z 15 dopravníků, což jsou všechny dopravníky před skladem, bude čekat objekt. Dále bude objekt i na vstupních platformách před pracovišti CNC.

Tabulka 3. Přehled čekajících objektů na lince

Místo	Čekající objekt
Dopravník - víko-emiter	ANO
Dopravník - základna-emiter	ANO
Vstupní platforma - víko	ANO
Vstupní platforma - základna	ANO
Dopravník 1 - víko	ANO
Dopravník 1 - základna	ANO
Dopravník 2 - víko	ANO
Dopravník 2 - základna	ANO
Dopravník 3 - víko	ANO
Dopravník 3 - základna	ANO
Dopravník 4	ANO
Dopravník 5	ANO
Dopravník 6	ANO
Dopravník 7	ANO
Dopravník 8	ANO
Dopravník 9	ANO
Dopravník 10	ANO

Obrázek č. 84 zobrazuje čekající objekty ve Factory I/O. Dopravníky jsou číslovány zleva doprava. Jak je z obrázku patrné, program je odladěný a nedochází k žádným kolizím při přesunu materiálu. Ke kolizním stavům nedochází ani před samotným zaplněním všech skladových pozic, což znázorňuje obrázek č. 82 výše, kdy každá hotová sestava, která je uložena ve skladu v bedně na paletě, je řádně slícovaná a naskladněná tzn. druhý cíl je také splněn. Simulační ověření obou zmíněných cílů řízení je součástí elektronické přílohy práce.



Obrázek 84. Čekající objekty ve Factory I/O

ZÁVĚR

Tématem bakalářské práce byl návrh virtuální výrobní linky demonstrující výrobní operace (obrábění, montáž, balení), transport materiálu a jeho skladování za využití programového vybavení Factory I/O a TIA Portal. Konkrétně bylo využito Factory I/O v2.5.2 – Ultimate Edition a TIA Portal V18.

V teoretické části byla zpracována literární rešerše zabývající se programovatelnými automaty, jejich použitím s 3D simulátorem Factory I/O a alternativním softwarem EasyPLC. Součástí rešerše je popis prostředí Factory I/O a TIA Portal, návod, kde vybrat hardware, jak vytvořit projekt a program v TIA Portal, konfigurace vstupů a výstupů ve Factory I/O a jejich import do TIA Portal. Závěr teoretické části je věnován spuštění simulace a online režimu, který je důležitý pro ověření funkčnosti programu.

V praktické části byly na základě bodů zadání bakalářské práce stanoveny cíle řízení virtuální linky, kterými je konkrétně naskladnit hotové sestavy uložené v bedně na paletě do 54 skladových pozic s tím, že před samotným naskladněním i po něm na pracovištích a dopravnících před skladem nedojde ke kolizi.

K tomuto účelu byly při návrhu linky ve Factory I/O vybrány scény Production Line, Assembler, Automated Warehouse, dopravníky se senzory a generátor bedny s paletou. U scén bylo upraveno rozmístění senzorů a byly z nich odebrány řídicí panely, protože simulace je řízena centrálně z ovládací lišty Factory I/O. Dále byl zvolen počet a typ vstupů a výstupů, který zahrnuje 39 digitálních vstupů, 32 digitálních výstupů a 1 analogový výstup. Dále byl vybrán řídicí systém SIMATIC S7-1500 (S7-PLCSIM Advanced), protože simulace je spuštěna na simulátoru PLC.

Následně byl vytvořen projekt v TIA Portal, bylo vybráno CPU 1513-1 PN a přídatné karty se vstupy a výstupy a vytvořen řídicí program. Program byl vytvořen za účelem splnění stanovených cílů, které vycházejí z bodů zadání bakalářské práce. Řídicí program je psán v LAD a ST. Program řídí výrobu základen a vík na pracovištích CNC, přesun objektů po dopravnících, montáž vík a základen 2-osým manipulátorem, ukládání hotové sestavy 2-osým manipulátorem do bedny a naskladnění do skladu. Při tvorbě programu byly použity FB a to konkrétně pro dopravníky, pracoviště CNC a vyhodnocení posuvu 2-osých manipulátorů. Pro pracoviště montáže, balení a sklad byly vytvořeny FC. Vytvořený program byl zkompileován a nahrán do simulátoru PLC, který je spuštěn v PLCSIM Advanced V5.0.

V závěru praktické části došlo k simulačnímu ověření stanovených cílů řízení, při němž došlo k naskladnění hotových sestav uložených v bedně na paletě do 54 skladových pozic a před ani po samotném zaplnění skladu nedošlo ke kolizi. Na základě simulačního ověření byly cíle vyhodnoceny jako splněné. Splnění cílů je mj. zajištěno tím, že dopravníky a pracoviště si mezi sebou posílají signály, zdali na sobě dopravníky mají či nemají materiál a jestli mohou materiál odeslat na další dopravník či pracoviště.

Program vytvořený v TIA Portal a projekt z Factory I/O jsou přílohou této bakalářské práce. Přínosem práce může být i to, že práce může pro studenty případně pracující, kteří běžně s automatizací do styku nepřichází, sloužit jako návod pro začátky programování PLC z pohodlí domova. Tohle jim umožňuje virtuální simulátor PLC, který v této práci nahrazuje úlohu fyzického PLC.

SEZNAM POUŽITÉ LITERATURY

- [1] ŠMEJKAL, Ladislav a MARTINÁSKOVÁ, Marie. *PLC a automatizace*. 1. díl, Základní pojmy, úvod do programování. Praha: BEN - technická literatura, 1999. ISBN 8086056589
- [2] MARTINÁSKOVÁ, Marie a ŠMEJKAL, Ladislav. *Řízení programovatelnými automaty*. Praha: ČVUT, 1998. ISBN 8001017664
- [3] SIEMENS. *Programming Guideline for S7-1200/1500* [online]. V1.6. Siemens AG, 2018 [cit. 2024-02-15]. Dostupné z: <https://support.industry.siemens.com/cs/document/90885040/programming-guideline-for-s7-1200-s7-1500?dti=0&lc=en-CZ>
- [4] PETRUZELLA, Frank D. *Programmable Logic Controllers*. 4th ed. New York: McGraw-Hill Companies, 2010. ISBN 978-0-07-351088-0
- [5] Romanov, Vladimir. TOP 5 Most Popular Types of PLC programming Languages. *Solis PLC* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://www.solisplc.com/blog/plc-programming-languages>
- [6] SIEMENS. *S7-1200 Programmable controller* [online]. Siemens AG, 2012. [cit. 2024-02-15]. Dostupné z: https://cache.industry.siemens.com/dl/files/465/36932465/att_106119/v1/s71200_system_manual_en-US_en-US.pdf
- [7] Difference between functions and function blocks in Siemens PLC programming In: *Zip Automations* [online]. 14. 9. 2022 [cit. 2024-02-15]. Dostupné z: <https://zipautomations.com/difference-between-functions-and-function-blocks-in-siemens-plc-programming/>
- [8] Salama, Mahmoud. Overview of SIEMENS PLC – S7-1500, S7-1200, S7-400, S7-300. In: *Inst Tools* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://instrumentationtools.com/overview-of-siemens-plc/>
- [9] Ashlin. What are the components of PLC? In: *AutomationForum.co* [online]. 19. 4. 2021 [cit. 2024-05-04]. Dostupné z: <https://automationforum.co/what-are-the-components-of-a-plc/> POZN. Na strane 14, pak jsem to smazal
- [10] EasyPLC Software Suite. *Nirtec* [online]. ©2023 [cit. 2024-05-04]. Dostupné z: <https://www.nirtec.com/easyplc-software-suite/>
- [11] *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://factoryio.com/>

- [12] Features. *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://factoryio.com/features>
- [13] Ghareb, Hussien. Understanding the Scan Cycle of SIEMENS PLC. In: *Inst Tools* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://instrumentationtools.com/understanding-the-scan-cycle-of-siemens-plc/#inside-cpu-module-of-siemens-plc>
- [14] Siemens Edition. *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://factoryio.com/siemens-edition>
- [15] SIMATIC S7-1200 Programmable controller. In: *SIEMENS*. [online]. 30. 4. 2014 [cit. 2024-05-04]. Dostupné z: <https://support.industry.siemens.com/cs/mdm/91696622?c=50106513547&dl=el&lc=cs-CZ>
- [16] Starter Edition. *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://factoryio.com/starter-edition>
- [17] Ultimate Edition. *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://factoryio.com/ultimate-edition>
- [18] What is a PLC? In: *Inductive automation* [online]. 24. 2. 2020 [cit. 2024-05-04]. Dostupné z: <https://inductiveautomation.com/resources/article/what-is-a-PLC>
- [19] What is PLC? *PLC Table* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://www.plctable.com/what-is-plc/>
- [20] Zola, Andrew. Programmable logic controller (PLC). In: *Tech Target* [online]. ©1999-2024 [cit. 2024-05-04]. Dostupné z: <https://www.techtarget.com/whatis/definition/programmed-logic-controller-PLC>
- [21] EasyPLC Machines Simulator. *Nirtec* [online]. ©2023 [cit. 2024-05-04]. Dostupné z: <https://www.nirtec.com/easyplc-machines-simulator/>
- [22] Editions. *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://factoryio.com/editions/>
- [23] Factory I/O? *EDU4INDUSTRY* [online]. ©2022 [cit. 2024-05-04]. Dostupné z: <https://edu4industry.com/en/produkt/factory-i-o/>
- [24] REAL GAMES. *Factory I/O Manual* [online]. 2006-2023 [cit. 2024-05-04]. Dostupné z: <https://docs.factoryio.com/>
- [25] I/O Drivers. In: *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://docs.factoryio.com/manual/drivers/>

- [26] Lins, Letácio. The Complete Practical Guide to Siemens Tia Portal Programming. In: *SolisPLC* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://www.solisplc.com/tutorials/a-practical-guide-to-siemens-tia-portal-programming>
- [27] MHJ Edition. *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://factoryio.com/mhj-edition>
- [28] Nejrychlejší řídicí systém pro automatizaci. In: *SIEMENS* [online]. ©1996-2024 [cit. 2024-05-04]. Dostupné z: <https://www.siemens.com/cz/cs/products/automation/systems/industrial/plc/simatic-s7-1500.html>
- [29] Představujeme novou verzi TIA Portal V18. In: *Visions* [online]. ©1996-2024 [cit. 2024-05-04]. Dostupné z: <https://www.visionsmag.cz/predstavujeme-novou-verzi-tia-portal-v18>
- [30] Purchase price. *Nirtec* [online]. ©2023 [cit. 2024-05-04]. Dostupné z: <https://www.nirtec.com/purchase-price/>
- [31] Phillips, Rick. What are the different types of PLC? In: *BASICPLC.COM* [online]. 8. 11. 2022 [cit. 2024-05-04]. Dostupné z: <https://basicplc.com/different-types-of-plc/>
- [32] Romanov, Vladimir. Introduction to PLC programming. *Solis PLC* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://www.solisplc.com/what-is-plc-programming>
- [33] Siemens 6ES7511-1AK02-0AB0 6ES75111AK020AB0 CPU pro PLC. In: *CONRAD* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: https://www.conrad.cz/cs/p/siemens-6es7511-1ak02-0ab0-6es75111ak020ab0-cpu-pro-plc-2263706.html?gad_source=1&gclid=Cj0KCQjwltKxBhDMARIsAG8KnqXYE-TaqYJJbnVfJhTpzUVzO-HCPit__FYom4JO_fC2K3sIZ6yEJTVVUaArlZEALw_wcB
- [34] Stacker Crane and Rack. In: *Factory I/O* [online]. ©2006-2023 [cit. 2024-05-04]. Dostupné z: <https://docs.factoryio.com/manual/parts/stations/#stacker-crane-and-rack>
- [35] SYSALA, Tomáš, 2018. PLC – obecná přednáška: Presentace v rámci předmětu Programování PLC.

- [36] TIA – plně integrovaná automatizace napříč celým podnikem. In: *SIEMENS* [online]. ©1996-2024 [cit. 2024-05-04]. Dostupné z: <https://www.siemens.com/cz/cs/products/automation/industry-software/automatizacni-software/tia-portal.html>
- [37] Types of PLC. In: *Inst Tools* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://instrumentationtools.com/types-of-plc/>
- [38] Types of Interrupt OB in Siemens Tia Portal. In: *Inst Tools* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://instrumentationtools.com/types-of-interrupt-ob-in-siemens-tia-portal/>
- [39] WinSPS-S7. *MHJ SOFTWARE* [online]. ©2024 [cit. 2024-05-04]. Dostupné z: <https://www.nirtec.com/easyplc-software-suite/>
- [40] What are 3 types of PLC? In: *Empowered Automation Solutions LLC* [online]. 30. 3. 2023 [cit. 2024-05-04]. Dostupné z: <https://www.empoweredautomation.com/what-are-the-3-types-of-plc>
- [41] *Nirtec*. [online]. ©2023 [cit. 2024-05-04]. Dostupné z: <https://www.nirtec.com/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ADC	Analog – to - Digital.
A/D	Analogově – Digitální.
CPU	Central Processing Unit.
DB	Data Block.
D/A	Digitálně – Analogový.
FB	Function Block.
FBD	Function Block Diagram.
FC	Function.
HMI	Human Machine Interface
IL	Instruction List.
LD	Ladder Diagram.
OB	Organization Block.
PLC	Programmable Logic Controller.
RAM	Random Access Memory.
ST	Structured Text.
TIA	Totally Integrated Automation.

SEZNAM OBRÁZKŮ

Obrázek 1. Základní schéma PLC [35].....	13
Obrázek 2. Ekvivalent zapojení v reléové logice, LD a IL [4].....	19
Obrázek 3. Ekvivalent zapojení v žebříčkové logice a FBD [4]	20
Obrázek 4. Ekvivalent zapojení LD a FBD [4]	21
Obrázek 5. SIMATIC S7-1500 CPU 1511-1 PN [33]	25
Obrázek 6. Přehled podporovaných programovacích jazyků [3]	26
Obrázek 7. Operační systém a uživatelský program [3].....	27
Obrázek 8. Přidání nového bloku (OB) v TIA Portal [3]	29
Obrázek 9. Přidání FC v TIA Portal	31
Obrázek 10. Základní datové typy korespondující s EN 61131-3 [3]	32
Obrázek 11. Skupiny dat vytvořené z jiných datových typů [3]	32
Obrázek 12. Typy parametrů pro transfer mezi bloky [3]	33
Obrázek 13. Portal view v TIA Portal V18	34
Obrázek 14. Project view v TIA Portal V18.....	34
Obrázek 15. Vytvoření nového projektu.....	35
Obrázek 16. Přidání nového zařízení.....	35
Obrázek 17. Přehled vybraných zařízení	36
Obrázek 18. Nastavení IP adresy	36
Obrázek 19. Přidání nového bloku	37
Obrázek 20. Výběr programového bloku	38
Obrázek 21. Přidání vstupů a výstupů	38
Obrázek 22. Tabulka s tagy	39
Obrázek 23. Nahrání vstupů a výstupů z Factory I/O do TIA Portal	39
Obrázek 24. Nabídka instrukcí	40
Obrázek 25. Pojmenování networku.....	41
Obrázek 26. Kompilace projektu	42
Obrázek 27. Spuštění PLCSIM Advanced V5.0.....	43
Obrázek 28. Spuštěný PLCSIM Advanced V5.0.....	43
Obrázek 29. Nahrání programu do PLC	44
Obrázek 30. Nahrání programu do PLC - pokračování	44
Obrázek 31. Spuštění online režimu	44
Obrázek 32. Online kontrola funkčnosti.....	45

Obrázek 33. Nabídka scén z Factory I/O	47
Obrázek 34. Popis prostředí Factory I/O	48
Obrázek 35. Výběr řady PLC ve Factory I/O	49
Obrázek 36. Nastavení množství vstupů a výstupů ve Factory I/O	49
Obrázek 37. Přiřazení senzorů a aktuátorů ve Factory I/O	50
Obrázek 38. Nabízené edice Factory I/O [11]	51
Obrázek 39. Porovnání nabízených edic Factory I/O [22]	51
Obrázek 40. Seznam kompatibilních modelů PLC a požadovaného softwaru [14] ...	52
Obrázek 41. Izolovaný digitální I/O modul Advantech USB 4750 připojený k PLC [17]	52
Obrázek 42. Připojení PLC Simatic S7-1200 přes Ethernet [17]	53
Obrázek 43. Graftec studio [27]	53
Obrázek 44. Masky PLC [39]	54
Obrázek 45. Editor kontroléru – edice Starter [16]	55
Obrázek 46. Machine simulátor [10]	56
Obrázek 47. EasyPLC software [10]	57
Obrázek 48. Konfigurace	61
Obrázek 49. CNC pracoviště	62
Obrázek 50. Retroreflexní a difuzní senzor	63
Obrázek 51. Montáž (Pick & place 1)	65
Obrázek 52. Balení (Pick & place 2)	66
Obrázek 53. Skladové pozice [34]	67
Obrázek 54. Sklad	67
Obrázek 55. Základní kostra programu	68
Obrázek 56. Zvolené CPU a přídatné karty	69
Obrázek 57. Proměnné pro FB	70
Obrázek 58. Přiřazení vstupů a výstupů v OB1	71
Obrázek 59: Instance DB pro dopravníky	71
Obrázek 60. Proměnné vytvořené pro FB2 „Dopravníky“	72
Obrázek 61. Ukázka komunikace mezi dopravníky	73
Obrázek 62. Network 1 a 2 – FB2 „Dopravníky“	74
Obrázek 63. Network 5 – FB2 „Dopravníky“	75
Obrázek 64. Network 6 a 7 – FB2 „Dopravníky“	76

Obrázek 65. Pick & place 1 – FC3	76
Obrázek 66. Pick & place 1 - proměnné	77
Obrázek 67. Network 1 a 2 – FB3	78
Obrázek 68. Instance DB pro Pick & place 1 a 2	78
Obrázek 69. Network 2 a 3 – Pick & place 1	79
Obrázek 70: Network 4 a 5 - Pick & place 1	80
Obrázek 71. Network 14 a 15 – Pick & place 1	81
Obrázek 72. Network 16 – Pick & place 1	81
Obrázek 73. Pick & place 2 - proměnné	82
Obrázek 74. Network 3 – FC4	82
Obrázek 75. FC5 „sklad“	83
Obrázek 76. Sklad proměnné	84
Obrázek 77. Sklad proměnné 2	84
Obrázek 78. Sklad proměnné 3	85
Obrázek 79. Naskladňování v ST	85
Obrázek 80. Testování v TIA Portal	87
Obrázek 81. Online režim - naskladňování na skladové pozice	88
Obrázek 82. Plně obsazený sklad	89
Obrázek 83. Instance DB5 pro FB2	90
Obrázek 84. Čekající objekty ve Factory I/O	92

SEZNAM TABULEK

Tabulka 1. Rozdíl mezi FC a FB [7].....	30
Tabulka 2. Procesní časy pracovišť	59
Tabulka 3. Přehled čekajících objektů na lince	91

SEZNAM PŘÍLOH

Příloha P I. VSTUPY A VÝSTUPY – FACTORY I/O.....	105
--	-----

PŘÍLOHA P I. VSTUPY A VÝSTUPY – FACTORY I/O

Datový typ	Adresa	Název	Datový typ	Adresa	Název
Bool	"%I0.0"	Senzor - dopravník viko-emiter	Bool	"%Q0.0"	Dopravník motor - viko-emiter
Bool	"%I0.1"	Viko cnc (zanepřazdneno)	Bool	"%Q0.1"	Dopravník 1 motor - viko
Bool	"%I0.2"	Senzor - dopravník základna-emiter	Bool	"%Q0.2"	Lids center (start)
Bool	"%I0.3"	Základna cnc (zanepřazdneno)	Bool	"%Q0.3"	Dopravník motor - základna-emiter
Bool	"%I0.4"	FACTORY I/O (Reset)	Bool	"%Q0.4"	Dopravník 1 motor - základna
Bool	"%I0.5"	Senzor dopravník 1 - viko vstupni	Bool	"%Q0.5"	Bases center (start)
Bool	"%I0.6"	Senzor dopravník 1 - základna vstupni	Bool	"%Q0.6"	Dopravník 2 motor - viko
Bool	"%I0.7"	Senzor dopravník 2 - viko	Bool	"%Q0.7"	Dopravník 2 motor - základna
Bool	"%I1.0"	Senzor dopravník 2 - základna	Bool	"%Q1.0"	Dopravník 3 motor - základna
Bool	"%I1.1"	Senzor dopravník 1 - základna vystupni	Bool	"%Q1.1"	Dopravník 3 motor - viko
Bool	"%I1.2"	Senzor dopravník 1 - viko vystupni	Bool	"%Q1.2"	Základna polohovac
Bool	"%I1.3"	Senzor dopravník 3 - viko	Bool	"%Q1.3"	Viko polohovac
Bool	"%I1.4"	Senzor dopravník 3 - základna vstupni	Bool	"%Q1.4"	2-osy Pick & Place 1 (Posuv v X)
Bool	"%I1.5"	Senzor dopravník 3 - základna vystupni	Bool	"%Q1.5"	2-osy Pick & Place 1 (posuv v Z)
Bool	"%I1.6"	Senzor základna polohovac (v poloze)	Bool	"%Q1.6"	2-osy Pick & Place 1 (Uchopit)
Bool	"%I1.7"	Senzor viko polohovac (v poloze)	Bool	"%Q1.7"	Základna polohovac (zvednuti)
Bool	"%I2.0"	Senzor 2-osy Pick & Place 1 (Uchopeno)	Bool	"%Q2.0"	Dopravník 4 motor
Bool	"%I2.1"	Senzor 2-osy Pick & Place 1 (posuv v Z)	Bool	"%Q2.1"	Dopravník 5 motor
Bool	"%I2.2"	FACTORY I/O (Running)	Bool	"%Q2.2"	Dopravník 6 motor
Bool	"%I2.3"	Senzor základna polohovac (zvednuti)	Bool	"%Q2.3"	2-osy Pick & Place 2 (posuv v X)
Bool	"%I2.4"	Senzor 2-osy Pick & Place 1 (Posuv v X)	Bool	"%Q2.4"	2-osy Pick & Place 2 (posuv v Z)
Bool	"%I2.5"	Senzor dopravník 4	Bool	"%Q2.5"	2-osy Pick & Place 2 (Uchopit)
Bool	"%I2.6"	Senzor dopravník 5	Bool	"%Q2.6"	Dopravník 7 motor
Bool	"%I2.7"	Senzor dopravník 6	Bool	"%Q2.7"	Sestava polohovac
Bool	"%I3.0"	Senzor dopravník 7	Bool	"%Q3.0"	Dopravník 8 motor
Bool	"%I3.1"	Senzor 2-osy Pick & Place 2 (posuv v X)	Bool	"%Q3.1"	Dopravník 9 motor
Bool	"%I3.2"	Senzor 2-osy Pick & Place 2 (posuv v Z)	Bool	"%Q3.2"	Dopravník 10 motor
Bool	"%I3.3"	Senzor 2-osy Pick & Place 2 (Uchopeno)	Bool	"%Q3.3"	Nakladací plosina (posuv v Z)
Bool	"%I3.4"	Senzor dopravník 8	Bool	"%Q3.4"	Nakladací vidlice - vysunout (vlevo)
Bool	"%I3.5"	Senzor sestava polohovac (v poloze)	Bool	"%Q3.5"	Nakladací vidlice - vysunout (vpravo)
Bool	"%I3.6"	Senzor dopravník 9	Bool	"%Q3.6"	Dopravník 11 motor
Bool	"%I3.7"	Senzor dopravník 10	Bool	"%Q3.7"	Dopravník 12 motor
Bool	"%I4.0"	Senzor nakladací plosina (posuv v X)	Dint	"%QD32"	Nakladací plosina - vychozi pozice 55
Bool	"%I4.1"	Senzor nakladací plosina (posuv v Z)			
Bool	"%I4.2"	Senzor vysunuti - nakladací vidlice (vlevo)			
Bool	"%I4.3"	Senzor vysunuti - nakladací vidlice (vpravo)			
Bool	"%I4.4"	Nakladací vidlice (stredova pozice)			
Bool	"%I4.5"	Senzor dopravník 11			
Bool	"%I4.6"	Senzor dopravník 12			