



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**NÁVRH FUNKČNÍHO BLOKU PRO POKROČILÉ
ŘÍZENÍ KOLABORATIVNÍCH ROBOTŮ UNIVERSAL
ROBOTS S VYUŽITÍM SIEMENS TIA PORTAL**

DESIGN OF A FUNCTION BLOCK FOR ADVANCED CONTROL OF UR COLLABORATIVE ROBOTS
USING THE SIEMENS TIA PORTAL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Rusnák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. et Ing. Stanislav Lang Ph.D.

BRNO 2025

Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	Bc. Filip Rusnák
Studijní program:	Aplikovaná informatika a řízení
Studijní obor:	bez specializace
Vedoucí práce:	Ing. et Ing. Stanislav Lang, Ph.D.
Akademický rok:	2024/25

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh funkčního bloku pro pokročilé řízení kolaborativních robotů Universal Robots s využitím Siemens TIA Portal

Stručná charakteristika problematiky úkolu:

Závěrečná práce se zaměří na návrh funkčního bloku pro pokročilé řízení kolaborativních robotů společnosti Universal Robots za použití nástroje TIA Portal společnosti Siemens. Předmětem práce bude vytvoření modulárního řešení, které umožní efektivní řízení a sběr dat s využitím průmyslového protokolu PROFINET. Kromě samotného návrhu funkčního bloku se práce zaměří i na vytvoření rozhraní člověk–stroj pro intuitivní ovládání kolaborativního robota UR10e. Rozhraní člověk–stroj bude realizováno jako webová aplikace, která poskytne uživatelům možnost provádět úpravy parametrů, zpracovávat a řešit chybová hlášení, monitorovat stav systému a vizualizovat data v reálném čase. Pro zajištění bezpečné a efektivní výměny dat mezi řídicím systémem a uživatelským rozhraním bude využita technologie OPC UA (Open Platform Communications Unified Architecture). Validace navrženého řešení funkčního bloku bude provedena na kolaborativním robotu UR10e ze série E od společnosti Universal Robots, a to jak v simulačním prostředí, tak v reálných provozních podmínkách.

Závěrečná práce bude realizována ve spolupráci s firmou Intemac Solutions s.r.o., která poskytne potřebné technické vybavení a odborné konzultace k řešení problematice.

Cíle diplomové práce:

- 1) Provedení rešerše a samostudium v oblasti kolaborativní robotiky. Zejména:
Nastudování vlastností kolaborativních robotů, jejich způsobů řízení a konkrétních možností využití v průmyslové automatizaci.
Prozkoumání oblasti vertikální systémové integrace, včetně průmyslových komunikačních protokolů (např. PROFINET, OPC UA).
Prozkoumání možností využití rozhraní člověk–stroj pro efektivní řízení a monitorování kolaborativních robotů.
- 2) Návrh funkčního bloku pro pokročilé řízení kolaborativních robotů Universal Robots pomocí nástroje TIA Portal od společnosti Siemens.
- 3) Návrh rozhraní člověk–stroj pro intuitivní ovládání kolaborativního robota UR10e s využitím webových technologií a komunikačního protokolu OPC UA pro bezpečnou výměnu dat.
- 4) Validace funkčního bloku bude provedena na robotu UR10e ze série E od Universal Robots, jak v simulaci, tak v reálných podmínkách.

Seznam doporučené literatury:

- [1] KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016. ISBN 978-80-214-4828-5.
- [2] SICILIANO, Bruno a KHATIB, Oussama, ed. Springer handbook of robotics. 2nd edition. Berlin: Springer, 2016. ISBN 978-3-319-32550-7.
- [3] LYNCH, Kevin M. a Frank C. PARK. Modern robotics: mechanics, planning, and control. Cambridge, United Kingdom: Cambridge University Press, 2017. ISBN 978-1-107-15630-2.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2024/25

V Brně, dne

L. S.

doc. Ing. Zdeněk Hadaš, Ph.D.
ředitel ústavu

doc. Ing. Jiří Hlinka, Ph.D.
děkan fakulty

ABSTRAKT

Táto diplomová práca sa zaoberá návrhom a implementáciou funkčného bloku pre pokročilé riadenie kolaboratívneho robota Universal Robots UR10e s využitím programovacieho prostredia Siemens TIA Portal. Práca sa sústreďuje na vytvorenie flexibilného a modulárneho riešenia, ktoré umožňuje bezpečnú a efektívnu integráciu robota do priemyselného prostredia s dôrazom na štandardizovanú komunikačnú zbernicu PROFINET. Hlavná časť práce je venovaná návrhu a vývoju funkčného bloku v prostredí TIA Portal, vrátane jeho štruktúry, stavového automatu a podmienok pre bezpečnú prevádzku. Blok zahŕňa vstupné aj výstupné signály, a je schopný reagovať na chyby v reálnom čase. V práci je ďalej popis vytvoreného webového rozhrania HMI pomocou Python Flask, HTML a CSS, s využitím komunikačného protokolu OPC UA, ktoré umožňuje jednoduché ovládanie a vizualizáciu údajov na rôznych zariadeniach. Výsledná časť práce je zameraná na overenie navrhnutého funkčného bloku, a to ako formou simulačného nástroja s využitím Unity3D a PLCSIM Advanced, tak na reálnom hardvéri.

ABSTRACT

This diploma thesis provides the design and implementation of a functional block for advanced control of the collaborative robot Universal Robots UR10e using the Siemens TIA Portal programming environment. It focuses on the development of a flexible and modular solution that enables safe and efficient integration of the robot into an industrial environment with an emphasis on the standardized PROFINET communication bus. The main part of the thesis is dedicated to the design and development of the functional block in the TIA Portal environment, including its structure, state machine and conditions for safe operation. The block includes both input and output signals, and is able to respond to errors in real time. The thesis also describes the created HMI web interface using Python Flask, HTML and CSS, using the OPC UA communication protocol, which allows easy control and visualization of data on various devices. The final part of the thesis is focused on verifying the description of the functional block, both in the form of a simulation tool using Unity3D and PLCSIM Advanced, and on real needs.

KLÚČOVÉ SLOVÁ

Siemens, TIA Portal, kolaboratívna robotika, rozhranie človek-stroj, programovateľný logický automat, OPC UA, PROFINET, systémová integrácia

KEYWORDS

Siemens, TIA Portal, collaborative robotics, human-machine interface, programmable logic controller, OPC UA, PROFINET, system integration

The logo consists of the lowercase letters 'ai' in white, set against a red square background.

ÚSTAV AUTOMATIZACE
A INFORMATIKY



2025

BIBLIOGRAFICKÁ CITÁCIA

RUSNÁK, Filip. Návrh funkčního bloku pro pokročilé řízení kolaborativních robotů Universal Robots s využitím Siemens TIA Portal. Diplomová práce. Stanislav LANG (vedoucí práce). Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2025.

POĎAKOVANIE

Veľká vďaka patrí Ing. Romanovi Parákovi, Ph.D., za odborné konzultácie, inšpiráciu a cenné rady, ktoré mali významný vplyv na kvalitu a výsledky mojej práce. Taktiež by som chcel poďakovať svojmu vedúcemu práce Ing. et Ing. Stanislavovi Langovi, Ph.D., za jeho dôveru a podnetné rady. V neposlednom rade patrí poďakovanie mojej rodine a priateľke, bez ktorých by toto štúdium nebolo možné.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 20. 5. 2025

.....

Filip Rusnák

OBSAH

1	ÚVOD	9
2	PREHLAD SÚČASNÉHO STAVU POZNANIA.....	11
2.1	Kolaboratívna robotika	11
2.1.1	Definície a charakteristika	11
2.1.2	Vyžitie v praxi.....	12
2.1.3	Bezpečnosť, regulácie a štandardy.....	13
2.2	Riadenie kolaboratívnych robotov	14
2.2.1	Ovládací pane.....	14
2.2.2	Ručné vedenie	15
2.2.3	Systémová Integrácia s Programovateľným Logickým Automation (PLC)	15
2.2.4	Skriptovacie jazyky a API.....	16
2.2.5	Ďalšie formy programovania	17
2.3	Programovateľný logický automat.....	17
2.3.1	Hardvérové komponenty.....	18
2.3.2	Softvér a programovanie PLC.....	18
2.3.3	Priemyslové komunikačné protokoly.....	20
2.3.4	Použitia a aplikácie PLC	20
2.4	Rozhranie človek - stroj	21
2.4.1	Súčasnú požiadavky priemyslu	21
3	FUNKČNÝ BLOK V TIA PORTAL.....	23
3.1	Výsledná architektúra ovládania	23
3.2	Totally Integrated Automation Portal	24
3.3	Požiadavky pre FB	25
3.4	Architektúra riešenia	27
3.4.1	Použitý softvér	27
3.4.2	Štruktúra riešenia FB	28
3.4.3	Konfigurácia komunikácie PLC s UR robotom.....	31
3.4.4	Štruktúra programových blokov v TIA Portal	33
3.4.5	Podmienky tvorenia UR programu	33
3.4.6	Ovládanie robota	35
4	WEBOVÉ HMI	38
4.1	Python	38
4.1.1	Moduly a knižnice použité pre vývoj aplikácie	39
4.1.2	PLC – HMI komunikácia	40
4.2	HTML, CSS a JavaScript.....	40
4.3	Login stránka.....	42
4.4	Dekorátory	42
4.5	Domovská stránka.....	43

4.6	Sidebar	44
4.7	Robot stránka	45
4.8	Kamera stránka	46
4.9	Error stránka.....	46
5	EXPERIMENTÁLNE OVEROVANIE	48
5.1	Testovanie funkčného bloku v simulácii	48
5.1.1	Unity.....	48
5.1.2	Testovanie v Unity3D	48
5.2	Testovanie funkčného bloku na reálnom hardvéri	49
5.2.1	Použitý hardvér	49
5.2.2	Prvý testovací program – overenie objektov.....	50
5.2.3	Druhý testovací program – vytvorenie datasetu	52
5.2.4	Tretí testovací program – <i>pick and place</i>	53
6	ZÁVER.....	55
	ZOZNAM POUŽITEJ LITERATÚRY	57
	ZOZNAM SKRATIEK.....	66
	ZOZNAM OBRÁZKOV.....	67
	ZOZNAM VÝPISOV KÓDOV.....	68
	ZOZNAM TABULIEK.....	69
	ZOZNAM PRÍLOH	70

1 ÚVOD

V poslednom desaťročí je zaznamenávaný výrazný technologický pokrok v oblasti priemyselnej automatizácie, pričom osobitná pozornosť je venovaná rozvoju kolaboratívnej robotiky. Na rozdiel od klasických priemyselných robotov, ktoré bývajú oddelované od ľudskej pracovnej sily, sú kolaboratívne roboty (koboti) navrhované tak, aby umožňovali bezpečnú spoluprácu človeka a stroja v spoločnom pracovnom priestore. Tento vývoj je považovaný za významný krok smerom k flexibilnejšej, adaptívnejšej a efektívnejšej výrobe. Spoločnosť Universal Robots je zaradovaná medzi popredných výrobcov kolaboratívnych robotov, ktorých zariadenia sú charakterizované jednoduchosťou programovania, pokročilými bezpečnostnými funkciami a možnosťou integrácie do rôznych automatizačných systémov [1]. V moderných výrobných závodoch býva čoraz častejšie požadované prepojenie robotických systémov s nadriadenými riadiacimi jednotkami, akými sú programovateľné logické automaty (PLC). Tieto jednotky sú využívané na zabezpečenie komplexných logických nadväzností a celkového riadenia výrobných procesov. Medzi aktuálne trendy v oblasti priemyselnej komunikácie patrí používanie štandardu OPC UA (Open Platform Communications Unified Architecture), prostredníctvom ktorého je umožňovaná spoľahlivá a bezpečná výmena údajov medzi rôznymi priemyselnými zariadeniami nezávisle od ich výrobcu. Prepojenie robota Universal Robots a PLC značky Siemens s využitím komunikačnej zbernice PROFINET a protokolu OPC UA predstavuje možnosť na vytvorenie univerzálneho a opakovane použiteľného softvérového riešenia.

V rámci tejto diplomovej práce je navrhovaný a implementovaný funkčný blok v prostredí Siemens TIA Portal, prostredníctvom ktorého je umožnené pokročilé riadenie robotov Universal Robots. V priemyselnej praxi býva často narážané na problém neefektívnej integrácie robotických systémov do existujúcich riadiacich architektúr, predovšetkým v prípadoch, kde je vyžadovaná vyššia úroveň flexibility, škálovateľnosti a bezpečnosti. Bežne dostupné riešenia nebývajú schopné poskytnúť dostatočnú mieru konfigurovateľnosti, alebo býva ich použitie obmedzené na špecifické značky a uzavreté softvérové platformy. Navrhovaný funkčný blok je preto vytváraný ako univerzálny a modulárny základ, ktorým je umožňovaná integrácia robota UR s PLC systémom Siemens prostredníctvom protokolu OPC UA bez potreby dodatočných licencií alebo špecializovaných nástrojov. Implementáciou stavového automatu v prostredí TIA Portal je dosahovaná zjednodušená programovateľnosť a zároveň aj zvýšená transparentnosť vykonávaných operácií. Riešenie je ďalej rozširované o podporu diagnostiky chybových stavov a bezpečnostného zastavenia, čím je zvyšovaná bezpečnosť celého systému. Súčasne je vytváraná webová HMI aplikácia, ktorou je zabezpečené užívateľsky prívetivé vzdialené ovládanie robota prostredníctvom webového prehliadača. Aplikácia je navrhovaná s použitím *frameworku* Flask v kombinácii s HTML, CSS a JavaScriptom, pričom je kladený dôraz na jednoduchosť obsluhy, prehľadnú vizualizáciu a možnosť rýchleho zásahu v prípade potreby údržby alebo diagnostiky. Týmto spôsobom je zabezpečovaná reakcia na aktuálne potreby moderného priemyslu, v ktorom je kladený dôraz na interoperabilitu, jednoduchosť používania a bezpečnosť.

Diplomová práce začíná teoretickou částí, v ktorej je uvedený prehľad informácií o kolaboratívnej robotike, jej praktickom využití a dostupných metódach riadenia. Ďalej je rozoberaná problematika programovateľných logických automatov (PLC), vrátane ich hardvérovej a softvérovej výbavy, podporovaných komunikačných protokolov a konceptov rozhrania HMI. Následne je predstavený samotný funkčný blok vytvorený v prostredí TIA Portal, vrátane požiadaviek na jeho implementáciu, architektúry riešenia, konfigurácie komunikácie medzi PLC a UR robotom, štruktúry programových blokov, ako aj logiky riadenia robota. V ďalšej kapitole je popísané webové HMI rozhranie, jeho funkcionality a vizuálne spracovanie. Záverečná časť práce je venovaná experimentálnemu overeniu funkčného bloku, a to jednak prostredníctvom simulácie v Unity3D a PLCSIM Advanced, ako aj pri testovaní na reálnom hardvéri pomocou viacerých overovacích programov.

2 PREHĽAD SÚČASNÉHO STAVU POZNANIA

2.1 Kolaboratívna robotika

Robotické technológie transformovali výrobný priemysel už začiatkom šesťdesiatych rokov minulého storočia [2]. Priemyselné roboty sú dnes do značnej miery naprogramované na konkrétne dané úlohy a nie sú schopné odhaliť chyby v procese alebo reagovať na podnety od pracovníkov a prostredia. V súčasnosti sú potrebné adaptívne riešenia, ktoré poskytujú rýchle reakcie, reštrukturalizáciu, preplánovanie, prípadne čiastočné doladenie. Úplná autonómia nie je otázkou blízkej budúcnosti vzhľadom na komplexnosť problematiky. Nejedná sa len o už spomínané reakcie na podnety a odhalenie chýb, ale aj o učenie sa od ľudí, dbanie na bezpečnosť, adaptáciu na nové prostredie a vytváranie prirodzených reakcií [3]. Okrem autonómie sa čo raz viac dbá na zníženie zložitosti programovania a jednoduchosť prepojenia rôznych druhov zariadení. Jeden z možných technologických posunov smerom k dosiahnutiu týchto cieľov je kolaboratívna robotika.

2.1.1 Definície a charakteristika

Pre pochopenie rozdielu kolaboratívnych robotov od klasických priemyselných je potrebné chápať obidva druhy robotiky. Priemyselné roboty vnímame ako veľké a ťažké zariadenia pozostávajúce z mechanických a elektronických súčiastok, ktoré sú samostatne ovládateľné a programovateľné. Dokážu vykonávať zložitú sériu činností, pričom je ich aplikácia najčastejšia pri úlohách, ktoré sú pre ľudí nebezpečné alebo náročné. Úlohy sú väčšinou špecifické v izolovanom pracovnom priestore a tým pádom sú bez zásahu ľudskej pracovnej sily.

Kolaboratívne roboty sú roboty navrhnuté tak, aby mohli zdieľať pracovný priestor s ľuďmi a pracovali spolu v súlade. Práve preto sa nazývajú aj ako koboty. V porovnaní s klasickými priemyselnými robotmi sú oveľa ľahšie a mobilnejšie, vďaka čomu je ich presun v továrni alebo sklade jednoduchší. Flexibilita je ich ďalšie nezanedbateľné pozitívum. Jedného robota je preto možné aplikovať na viacero úloh. Bezpečne a efektívne pracovať s ľuďmi umožňuje aj ich ľahká programovateľnosť [4].

Prednosti kolaboratívnych robotov veľmi dobre zapadajú do slabín ľudských faktorov vo výrobnom priemysle. Zatiaľ čo človek je zručný, kreatívny a dokáže sa rozhodovať, robot tieto kvality nemá. Robot avšak dopĺňa človeka o jeho silu, presnosť, produktivitu a výdrž. V minulosti sa roboty používali len na automatizáciu rozsiahlych úloh, zatiaľ čo menšie, menej podstatné úlohy, boli ignorované a vykonávajú sa aj naďalej manuálne. Práve na takéto úlohy je ideálna aplikácia kolaboratívnej robotiky. Automatizácia aj menších úloh vedie k celkovej optimalizácii a zrýchleniu procesu výroby. Ďalší prípad použitia je možnosť vykonávania, alebo čiastočná pomoc pri nebezpečných, monotónnych, či ľuďom nepríjemných úlohách. V minulosti nastával odpor pracovníkov k robotom, ktorý bol spôsobený strachom o stratu pracovnej pozície. Tento problém avšak nenastáva v prípade kolaboratívnej robotiky, pretože je zrejma ich spolupráca, prídanie hodnoty a nie nahradenie zamestnanca [5].

Podľa spoločnosti Interact Analysis by mal do roku 2027 trh s kolaboratívnou robotikou dosiahnuť hodnotu 5,6 miliardy dolárov [6]. Je predpokladané, že trh s kobotmi bude nadalej narastať a rapídne si získa veľkú časť podielu trhu s robotikou. Vplýva na to viacero faktorov, ako napríklad nedostatok kvalifikovaných zamestnancov, rastúce náklady na pracovnú silu, väčšia potreba flexibility v automatizácií, vyšší sortiment produktov s kratšími cyklami, ale aj prísnejšie požiadavky na rýchlejšiu návratnosť investícií a ziskovosť.

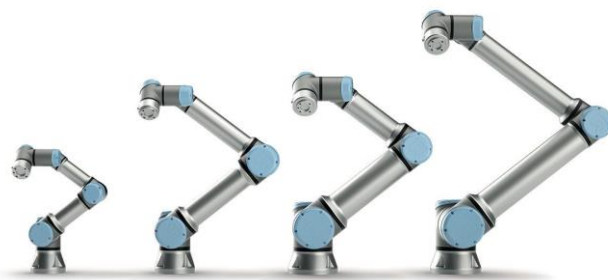
Vďaka menším počiatocným nákladom, menším rozmerom, rýchlejšej finančnej návratnosti v porovnaní s klasickými robotmi boli častou voľbou pre malé a stredné podniky. Očakáva sa zavádzanie kolaboratívnych robotov u mnohých OEM výrobcov najmä v automobilovom a elektronickom priemysle. V blízkej budúcnosti je pravdepodobné predstavenie viacerých nových výrobcov, čo bude viesť k vyššej konkurencie a následnému zníženiu cien. [7]

2.1.2 Vyžitie v praxi

Na kolaboratívne roboty je možné v praxi naraziť čoraz častejšie. V automobilovom priemysle sa používajú na úlohy ako sú skrutkovanie, zváranie, lepenie či kontrola kvality. Príkladom môže byť spoločnosť Volkswagen, ktorá ich používa pri skladaní motora v Nemecku alebo závod BMW v Južnej Karolíne, kde tieto roboty kolaborujú pri finálnej montáži dverí. Ďalším príkladom je Mercedes a jeho pilotný projekt, kde sa montujú zadné nápravy za použitia robotov s inteligentnými snímačmi pre krútiaci moment [8].

V prípade kombinácie kolaboratívnej robotiky s agrikultúrou a potravinárskym priemyslom sa najčastejšie jedná o *pick and place*, triedenie, balenie alebo paletizáciu [9, 10]. V týchto prípadoch je primárny cieľ zníženie manuálnej práce, a ako pozitívum je ich nepretržitá prevádzka. Ďalšie výhody sú konzistencia, presnosť, a taktiež splnenie hygienických noriem. V prípade výberu vhodného koncového efektoru, robot dokáže manipulovať aj s mäkkými potravinami bez ich poškodenia [11].

Koboty sú obzvlášť vhodné pre malé a stredné podniky, ktoré často nemajú financie na aplikáciu veľkých klasických robotov. Kolaboratívne roboty majú menšie počiatocné náklady, rýchlu integráciu a nepotrebujú vysoko vyškolených pracovníkov k ich obsluhu. Ich intuitívne rozhranie umožňuje rýchlu adaptáciu pre širší okruh ľudí. Ich mobilita a flexibilita umožňuje rýchle presúvanie a ľahké preprogramovanie medzi rôznymi úlohami [12]. Spoločnosti ktoré vyrábajú takéto roboty sú napríklad KUKA, Universal Robots, FANUC, ABB, Rethink Robotics alebo Dobot Robotics.



Obr. 1: Roboty Universal Robots [1]

2.1.3 Bezpečnosť, regulácie a štandardy

Hlavným dôvodom pre nesprávne použitie je chybné porozumenie samotných pojmov využívaných v kolaboratívnej robotike. Tá sa ešte aj v súčasnosti stretáva s nedostatočnými znalosťami certifikácií a noriem. Aj napriek tomu, že kolaboratívna robotika je v priemysle už viac než desaťročie, stále dochádza k nesprávnym predstavám ohľadom noriem a požiadaviek. Využitie kolaboratívnych robotov má rôznorodé kvalifikácie, no je potrebné štandardizovať ich uplatnenie na tieto predpisy [13]:

1. Roboty a robotické zariadenia: Bezpečnostné požiadavky na priemyselné roboty ISO 10218
2. Roboty a robotické zariadenia - Kolaboratívne roboty ISO/TS 15066

Norma ISO 10218 je rozdelená na dve časti. Prvá časť (ISO 10218-1) je pre výrobcov robotov, kde sú špecifikované bezpečnostné požiadavky na samotného robota, ako sú mechanický návrh, vstavané bezpečnostné funkcie, požiadavky na núdzové zastavenia alebo obmedzenia pohybu a rýchlosti. Druhá časť (ISO 10218-2) je pre systémových integrátorov a koncových používateľov. Zameriava na celkovú robotickú bunku, a teda aj bezpečnostné ploty, kryty, blokovania, postupy hodnotenia rizík, bezpečné vzdialenosti a bezpečnosť interakcie operátora. V prípade integrácie robota je potrebné postupovať v súlade s týmito predpismi, aby výrobná linka nepredstavovala žiadne riziko. [14]

ISO/TS 15066 dopĺňa normu ISO 10218 o špecifické pravidlá pre kolaboratívnu prevádzku. Poskytuje podrobné pokyny, ako implementovať a overiť bezpečnú spoluprácu medzi človekom a kolaboratívnym robotom, kde nie sú prítomné žiadne ploty ani fyzické bariéry. Obsahuje 4 režimy kolaboratívnej prevádzky:

1. Obmedzenie výkonu a sily – v prípade kolízie robot nespôsobí škodu.
2. Bezpečnostne monitorované zastavenie – robot sa úplne zastaví v prípade, že človek vstúpi do kolaboratívneho priestoru.
3. Monitorovanie rýchlosti a odstupu – robot spomalí alebo zastaví v prípade, že sa človek priblíži do určitej vzdialenosti a následne po opustení tejto vzdialenosti sa vráti na pôvodnú rýchlosť.
4. Navádzanie rukou – fyzické navádzanie ramena robota človekom pre naučenie polohy alebo dráhy, pričom snímače sily detekujú ľudský vstup a následne ho zaznamenávajú.

Norma ISO/TS 15066 taktiež definuje limity sily a tlaku, ktoré sú prijateľné počas kontaktu medzi človekom a robotom a zahŕňa tabuľky špecifických limitov. Obsahom sú aj požiadavky na posúdenie rizika, zdôraznenia tvaru a materiálu nástroja, environmentálne riziká a limity rýchlosti. V norme sú uvedené aj návody pre testovanie a meranie bezpečnosti senzorov sily, kolíznych testov a simulačných či fyzikálnych testov. [15]

Vzhľadom na tieto normy majú kolaboratívne roboty obmedzené rýchlosti, maximálnu hmotnosť koncového efektoru, a teda je obmedzená aj hmotnosť manipulovaného objektu. V prípade firmy Universal Robots roboty dokážu uniesť hmotnosť od 3 do 35 kilogramov avšak toto číslo sa znižuje v závislosti na vzdialenosti ťažiska [16]. Môže sa zdať, že sa jedná o vysoké hmotnosti, ale v prípade klasických robotov napríklad roboty ABB dosahujú maximálne zaťaženie až 800kg [17].

2.2 Riadenie kolaboratívnych robotov

Existujú dva základné druhy riadenia pohybu robota. Prvý je manipulácia s jednotlivými kĺbmi nazývané aj riadenie kĺbového priestoru. Druhý je upravovanie polohy a orientácie stredového bodu nástroja robota. V závislosti od aplikácie môže používateľ potrebovať presné riadenie dráhy alebo iba jednoduchý pohyb z bodu do bodu. Kolaboratívne roboty sú navrhnuté s užívateľsky prívetivými riadiacimi systémami, ktoré umožňujú efektívnu interakciu s nimi a odborným programátorom, ale aj technickým operátorom, ktorý nedokáže programovať. Na rozdiel od tradičných priemyselných robotov, ktoré často vyžadujú rozsiahle znalosti programovania, koloboty ponúkajú viacero prístupných ovládacích rozhraní. [18]

2.2.1 Ovládací pane

Ovládací panel (*Teach Pendant*, TP) je ručné zariadenie, ktoré obsahuje základné, alebo aj pokročilé operácie pre ovládanie robota. Dokáže spustiť a zastaviť program, nastaviť polohu robota alebo núdzovo zastaviť. Obsahuje tlačidlá, joystick, alebo displej, ktorý môže byť dotykový, ako je to v prípade robotov od firmy Universal Robots. [19] Joystick umožňuje pohyb v šiestich osiach, tlačidlá sú predprogramované na danú funkciu alebo môžu byť programovateľné či už na funkcie robota alebo jeho koncového efektoru. Pomocou displeja vieme prijímať a monitorovať údaje z robota, ako napríklad aktuálny stav každej osi, zatiaľ čo vstupmi vieme tvoriť a upravovať program robota. Program je zoznam pracovných úloh robota, ktorý sa nazýva aj JOB súbor a vytvára sa pomocou offline programovania [20]. TP dokáže tento program skompilovať a odhaliť prípadné chyby.

Existuje viacero typov TP. Najrozšírenejšia je káblková verzia, pripájaná prostredníctvom fyzického prepojenia, ktorá umožňuje dostatočnú flexibilitu pri zadávaní požiadaviek, ale niekedy môže byť obmedzujúca. Pevné TP sú inštalované priamo na robote a nemožno ich nijako upravovať. Druhým typom je bezdrôtový TP, ktorý funguje na základe rádiových signálov vďaka čomu poskytuje lepšiu mobilitu. Jednou z nevýhod tohto typu TP je možnosť dochádzaniu k oneskoreniam alebo rušeniam signálu. Ďalšou z možností je TP vo forme softvéru pre počítače, ktorý sa zvyčajne nachádza na priemyselnom počítači v blízkosti robota. K tomuto systému môže byť pripojená aj dotyková obrazovka, ktorá poskytuje flexibilné a prispôsobiteľné rozhranie. TP založené na tejto forme však bývajú drahšie a zložitejšie na používanie než klasické.

Využívanie TP môže viesť k zvýšeniu produktivity, keďže jeho ovládacie prvky častokrát umožňujú manuálne umiestnenie robota do požadovanej polohy. Rozhranie je zvyčajne intuitívne a jednoduché. Jasne označené tlačidlá a joystick uľahčujú používanie zamestnancom. Na TP sa zvyčajne nachádza aj tlačidlo núdzového zastavenia. V súčasnosti majú TP široké uplatnenie. Používajú sa v továrenském priemysle, kde sú nimi ovládané roboty pri zváraní, lakovaní alebo montáži. S TP sa vieme stretnúť ale aj v operačných sálach, kde zohrávajú kľúčovú úlohu pri chirurgických zákrokoch, kde je potreba vysokej presnosti a precíznosti. [21]



Obr. 2: Teach pendanty [22]

2.2.2 Ručné vedenie

Jednou z hlavných funkcií kolaboratívnych robotov je ručné vedenie (*Hand-guiding*). Vďaka nemu je programovanie intuitívne a jednoduché. Táto funkcia taktiež značne urýchľuje proces presného polohovania koncového efektora, čo vyžadujú mnohé aplikácie ako napríklad paletizovanie, zváranie alebo manipulácia s ťažkými objektmi [23–25]. Ručné vedenie robota umožňuje aj neskúseným používateľom interagovať a programovať robota intuitívnejším spôsobom v porovnaní s tradičným TP. Ľudia môžu fyzicky interagovať s robotickým ramenom tak, že manuálne vedú robota do požadovaných polôh, zatiaľ čo sa zaznamenáva konfigurácia robota [26]. Asistencia robota prostredníctvom ručného vedenia sa používa aj pri kooperatívnych úlohách pomocou nekonvenčného ovládania v režime posuvu, aby sa zaručili bezpečné referenčné parametre stanovené úlohou a nedošlo tak k zrážke s objektmi. Ručné vedenie je štandardnou funkciou mnohých kolaboratívnych robotov, ale predstavuje obmedzenia v prípade, že ide o presnosť, správnosť alebo intuitívnosť používania. Pre úlohy vyžadujúce presné polohovanie robota je TP stále primárnou voľbou pre väčšinu priemyselných aplikácií. Používanie TP však obmedzuje intuitívnosť učebného procesu, je časovo náročné a vyžaduje znalosti o špecifickom programovacom jazyku robota [27]. Okrem toho, používanie TP na polohovanie koncového efektora vyžaduje vizualizáciu rôznych referenčných osí, ktorej chýba intuitívnosť a môže byť mäťúca, keď sa človek pohybuje okolo robota [28]. Taktiež môže dôjsť ku kolíziám, ktoré môžu spôsobiť nežiadúce poškodenie robota, koncového efektora alebo iného objektu. [29]

2.2.3 Systémová Integrácia s Programovateľným Logickým Automation (PLC)

Programovateľné logické automaty (PLC) sú široko používané v priemyselnej automatizácii. Návrh technologických liniek, či iných digitálnych riadiacich systémov je často založený na využití PLC. Takéto zariadenia ponúkajú širokú škálu funkcií, ktoré je možné organizovať do blokov nezávisle alebo do hierarchických štruktúr. Integrácia kolaboratívnych robotov s PLC umožňuje synchronizované operácie v rámci väčších automatizačných systémov. V roku 2024 integrátor Siemens oznámil spoluprácu s dvomi najväčšími výrobcami kolaboratívnych robotov – Universal Robots a JAKA [30]. Komunikácia medzi robotom a PLC zariadením

môže prebiehať pomocou komunikačnej zbernice PROFINET alebo komunikačného protokolu OPC UA [31]. Programovateľnými logickými automatmi sa bude práca zaoberať v kapitole 2.3.

Pre dosiahnutie flexibilnejšej a adaptívnejšej integrácie robotov sú vytvárané integračné platformy, ktoré uľahčujú opätovné použitie kódu v rôznych aplikáciách s malými úpravami. Využitím týchto frameworkov sa zefektívňuje a zjednodušuje proces vývoja. Príkladom je ROBOTemplate od spoločnosti ABB, ktorý hovorí o až o 30 % úspore času [32]. Jedna z nevýhod je limitácia použitého hardvéru. Softvér ROBOTemplate vznikol ako kolaborácia firmy ABB a B&R [33] a preto disponuje len robotmi ABB a PLC zariadeniami od firmy B&R. Ďalšia možnosť je knižnica robotov SIMATIC Robot Library od spoločnosti Siemens. Aplikácia SIMATIC Robot Integrator umožňuje zahrnutie robotov s využitím štandardného rozhrania príkazov (*Standard Robot Command Interface, SRCI*) [34] vďaka čomu umožňuje kompatibilitu so širokou škálou robotov od výrobcov ako ABB, AUBO, COMAU, FAIRINO, FANUC, JAKA, KUKA, Universal Robots, YASKAVA a mnoho ďalších [35]. Nevýhodou využitia SIMATIC Robot Library sú licencie. Limitujúcim faktorom nie je len cena, ale aj podmienka verzie TIA Portal, ktorá musí byť minimálne 16 [36].

2.2.4 Skriptovacie jazyky a API

Niektoré kolaboratívne roboty podporujú využívanie skriptovacích jazykov. V prípade použitia týchto jazykov má používateľ väčšiu flexibilitu a v prípade programátorov sa jedná o príjemnejšie prostredie. Program je možné napísať bez nutnosti nachádzať sa pri robotovi a následne len nahráť, otestovať a doladiť program na mieste aplikácie. Toto môže urýchliť celú aplikáciu robotickej bunky, pretože sa na programe môže začať pracovať už pred dodaním a inštaláciou robota.

Firma Universal Robots poskytuje vlastný jazyk URScript [37] ale taktiež podporujú Real-Time Data Exchange (RTDE) knižnicu na ovládanie ich kobotov pomocou externých aplikácií napísaných v jazyku Python alebo C++ [38]. Vo všetkých troch prípadoch sa jedná o jazyky, vďaka ktorým vieme plnohodnotne ovládať a monitorovať robota. Koboty KUKA disponujú ovládateľnosťou pomocou jazyka KRL. Jedná sa o licencovaný programovací jazyk podobný zastaranému jazyku Pascal [39]. Okrem jazyka KRL pre kolaboratívne roboty KUKA existuje toolbox v programovacom jazyku MATLAB [40]. Tento toolbox obsahuje funkcie pre robotické manipulátory, ako sú homogénne transformácie, priama a inverzná kinematika, priama a inverzná dynamika a generovanie trajektórií. Ďalšou z výhod využitia tohto riešenia je možnosť rozšírenia softvéru o ďalšie funkcie pomocou iných toolboxov podporovaných programom MATLAB [41]. TMflow je prostredie založené na grafickom programovaní kobotov TM. Umožňuje kombinovanie grafického programovania a skriptov, čo zjednodušuje celkové ovládanie a programovanie [42].

RoboDK ponúka API, ktoré podporuje viacero programovacích jazykov, ako sú Python, C++ a MATLAB. API umožňuje simuláciu a offline programovanie viac než 700 druhov robotov od viac než 50 výrobcov vrátane ABB, Universal Robots, FANUC, Yaskawa a mnoho iných [43]. Offline programovanie, vývoj a testovanie je veľkou výhodou z hľadiska času, financií, a taktiež bezpečnosti. Takéto riešenie je obzvlášť užitočné na automatizáciu

opakujúcich sa úloh, či optimalizáciu trás robotov [44]. Okrem RoboDK ponúka podobné možnosti aj OCTOPUZ. Podporuje robotov od mnohých značiek, dokáže automaticky detegovať error robotov, optimalizovať programy vzhľadom na čas, simulovať komplexné 3D systémy a vie aj vygenerovať kód pre viaceré roboty [45].

2.2.5 Ďalšie formy programovania

Hlasové ovládanie sa čoraz viac používa v každodennom živote, či už ide o inteligentnú domácnosť, hlasové ovládanie mobilného telefónu alebo hlasové ovládanie funkcií v aute. Či alebo kedy sa hlasové ovládanie dostane plnohodnotne do priemyslu je zatiaľ otáznе [46]. V súčasnosti sa takéto ovládanie nachádza primárne v laboratóriách, ako napríklad systém real-time ovládania robota pomocou gestikulácie [47] alebo použitie kinectu pre monitorovanie rýchlosti a odstupe [48]. Gestikulácia môže stroju naznačovať ďalšie akcie, ktoré sa majú vykonať. Pridaním hlasových povelov môžeme tvrdiť, že spôsob komunikácie so strojmi sa priblíži komunikácii používanej v každodenných medziľudských vzťahoch. Vzhľadom na toto uvažovanie, bude používanie gest a hlasových povelov uľahčenie pre programovanie a ovládanie veľmi zložitých zariadení aj pre laikov v robotike.

2.3 Programovateľný logický automat

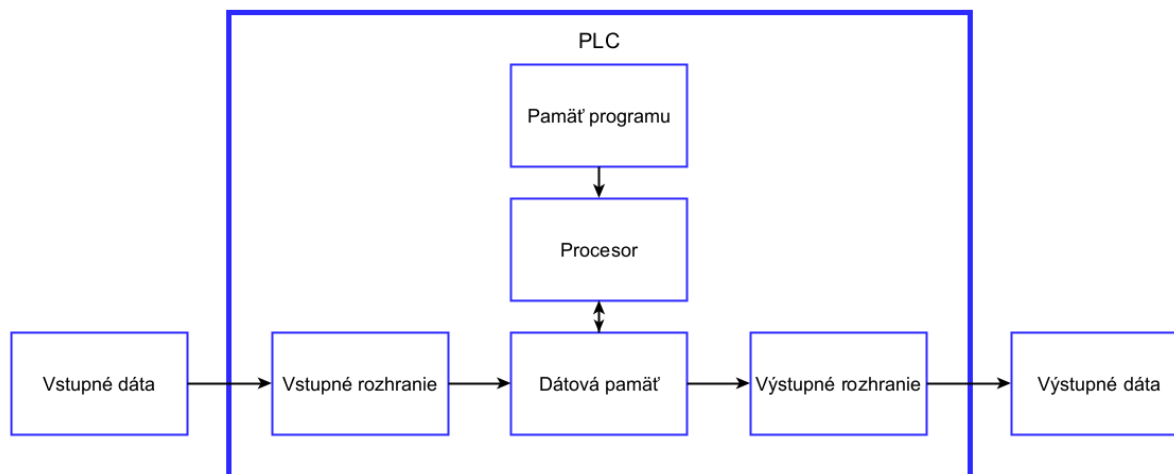
PLC vznikli koncom 60. rokov 20. storočia v automobilovom priemysle v USA a boli navrhnuté ako náhrada za logické relé systémy [49]. Tieto PLC boli naprogramované v takzvanej rebríkovej logike, ktorá sa značne podobá schematickému diagramu relé logiky. Jedná sa o špeciálnu formu riadiaceho systému, založenú na mikroprocesore. Aplikácia a využitie PLC má výhodu fyzickej ochrany pred nežiadúcimi faktormi, ako sú najčastejšie prach, teplo a chlad. Nevýhodou používania mikrokontrolérov bola potreba návrhu prídavnej elektroniky pre jednotlivé aplikácie, čo sa častokrát vyplatilo len pre veľké sériové výroby. Pri zariadeniach PLC tento problém nenastáva, keďže je ich softvér opakovateľne modifikovateľný a následne môžu byť znovu použiteľné. Nie je teda potrebné vymeniť samotné hardvérové komponenty meniť v závislosti od aplikácie ako je to v prípade použitia mikrokontroléra. Práve vďaka možnosti ich preprogramovaniu ich považujeme za špeciálne počítače na priemyselné použitie [50]. Príklad PLC zariadení je možné vidieť na obrázku 3.



Obr. 3: PLC od firmy Siemens [51]

2.3.1 Hardvérové komponenty

Veľkou výhodou PLC je ich modularita a možnosť pretransformovania ich hardvérovej zostavy v závislosti na požiadavkách konkrétnej aplikácie. Každá zostava obsahuje napájací zdroj, ktorý transformuje 230 V sieťové napätie na jednosmerné 24 V napätie. Toto 24 V DC napätie je následne privádzané do ostatných komponentov. Procesorový modul je hlavná súčiastka, v ktorej sa nachádza procesor, a taktiež pamäť RAM a ROM. Je v nej uložený program, kde prebiehajú všetky výpočty a procesy. Rovnako podstatné k celkovej funkcionalite sú moduly. Môže sa jednať o vstupné, výstupné, analógové a aj digitálne typy. Sú do nich pripojené káble spájajúce koncové zariadenia, ako sú napríklad senzory, svetlá, motory, oscilátory, dopravníky, ale aj komplexné stroje ako frézky, píly, brúsky či roboty. Bus adaptéry alebo zbernice, prípadne iné komunikačné moduly môžu byť taktiež zapojené v prípade komplexnejších aplikácií. Je možné aj zapojenie a integrácia rôznych obrazoviek alebo displejov, vďaka ktorým je možné ovládať určitú funkcionalitu PLC, a tiež zobrazovať dáta v reálnom čase, bez potreby pripojeného počítača. [52] Hardvérové komponenty sú zobrazené na obrázku 4.



Obr. 4: Schéma hardvéru PLC [53]

2.3.2 Softvér a programovanie PLC

PLC automaty sa dajú programovať pomocou viacerých programovacích jazykov, keďže spĺňajú tretiu časť normy IEC 61131. Táto norma sa zaoberá základnou softvérovou architektúrou a programovacími jazykmi riadiaceho programu v rámci PLC. Definuje dva textové a tri grafické štandardy programovacích jazykov:

1. Štruktúrovaný text (ST – Structured text)
2. Zoznam inštrukcií (IL – Instruction list)
3. Rebríkový diagram (LD – Ladder diagram)
4. Funkčný blokový diagram (FBD – Function block diagram)
5. Sekvenčný funkčný diagram (SFC - Sequential function chart)

Štruktúrovaný text je textový programovací jazyk vysokej úrovni podobajúci sa Pascalu, na ktorom je aj založený [54]. Jeho ľahká čitateľnosť môže byť výhodou pre programátorov a naopak

negatívom pre elektrikárov, ktorým viac vyhovujú grafické jazyky. Zoznam inštrukcií je taktiež textový jazyk, ktorý sa podobá na assembler. V súčasnosti je považovaný za zastaraný, kvôli jeho obmedzenej čitateľnosti, podpore a jeho nevhodnosť pre zložitú logiku [55]. Rebríkový diagram je grafický jazyk, napodobňujúci logických diagramov elektrických relé. Chýba mu flexibilita a možnosť pokročilého spracovania údajov, ktoré sú čoraz potrebnejšie v súčasných aplikáciách [56]. Funkčný blokový diagram a sekvenčný funkčný diagram sú taktiež grafické programovacie jazyky pričom každý blok reprezentuje špecifickú funkciu. Ich vizuálna štruktúra uľahčuje pochopenie interakcií signálov avšak pri zložitých aplikáciách neponúka flexibilitu ako štruktúrovaný text [57, 58].

Okrem programovacích jazykov norma definuje aj dátové typy:

1. Boolean / stringbit – skupina pre hodnoty 0/1
 - BOOL, BYTE, WORD, DWORD, LWORD
2. Signed integer
 - INT, SINT, DINT, LINT
3. Unsigned integer
 - UINT, USINT, UDINT, ULINT
4. Floating point (real)
 - REAL, LREAL
5. Time, duration, date and character string
 - TIME, DATE, TIME_OF_DAY, DATE_AND_TIME, STRING

Písmená znamenajú nasledovné: D – Double, L – Long, S – Short, U – Unsigned. Norma definuje aj generické dátové typy. Existujú pre jednoduchšie zapisovanie a následnú prácu s hodnotami. Nereprezentujú len jeden konkrétny druh hodnoty, ale podporujú zápis viacerých hodnôt pod jeden generický typ. Pravidlá sú nasledovné:

1. ANY_BIT dokáže pracovať s nasledovnými dátovými typmi:
 - BOOL, BYTE, WORD, DWORD, LWORD
2. ANY_DATE
 - DATE, TIME_OF_DAY, DATE_AND_TIME
3. ANY_STRING
 - STRING, WSTRING

V prípade generického dátového typu ANY_MAGNITUDE sa nachádzajú ešte podkategórie, ktoré sa ďalej členia nasledovne:

- ANY_MAGNITUDE
 - TIME
 - ANY_NUM
 - ANY_INT – obsahujú INT, SINT, DINT, LINT, UINT, USINT, UDINT, ULINT
 - ANY_REAL – obsahujú REAL, LREAL [55]

Programovacie prostredia slúžia ako programy, ktorými vieme konfigurovať, programovať a nastavovať PLC zariadenia. Najpopulárnejšie sú TIA Portal od Siemensu, RSLogix alebo Studio 5000 od Rockwellu alebo CX-Programmer od Omronu. Pre vývoj bol

použitý TIA Portal, ktorý je bližšie opísaný v kapitole 3.1. RSLogix alebo Studio 5000 sú rozhrania používajúce sa pre programovanie Rockwell PLC zariadení. Často sa používa v odvetviach ako je automobilový alebo potravinársky priemysel v Severnej Amerike [59].

2.3.3 Priemyslové komunikačné protokoly

Moderné PLC nie sú izolované systémy, ale sú súčasťou priemyselných sietí, ktoré umožňujú komunikáciu na vyššej úrovni. Táto komunikácia prebieha medzi rôznymi PLC, akčnými členmi, HMI, SCADA systémami či MES systémami. Spoľahlivá komunikácia v reálnom čase je nevyhnutná pre efektívne fungovanie automatizovanej výroby súčasnosti. Pre takéto fungovanie sú potrebné komunikačné protokoly ako napríklad PROFINET, OPC UA, EtherNet/IP alebo Modbus.

PROFINET je komunikačný štandard pre automatizáciu PROFIBUS a PROFINET International. Priemyselná zbernica PROFIBUS podľa smernice IEC 61158 podporuje mnoho možností implementácie výmeny dát medzi masterom zbernice a pripojenými slave zariadeniami. Od najjednoduchšieho slave zariadenia, ktoré obsluhuje iba niekoľko vstupných/výstupných kanálov, až po inteligentné slave zariadenie, ktoré spracováva úlohy prieskumného, môže master PROFIBUS vykonávať výmenu dát. Práve preto sa môže zariadenie s pripojením PROFIBUS optimálne prispôsobiť príslušnej automatizačnej úlohe [60]. Výhodou použitia komunikačného protokolu PROFINET je mnoho. Jednoduchosť a flexibilita použitia zabezpečuje jeho kompatibilitu s ethernetovým káblom podľa štandardu IEEE. Keďže aplikácia prebieha vždy cez rovnaký druh kábla, komunikácia je vždy škálovateľná, nezávislá na rozsiahlosti úlohy. Vďaka integrovanej komunikácii PROFINET spĺňa širokú škálu požiadaviek od dátovo náročného parametrizovania až po prenos dát v reálnom čase. Medzi výhody patrí aj globálne rozšírená dokumentácia, podpora, školenia a poradenstvo [61].

Špecifikácia OPC UA (*Open Platform Communications Unified Architecture*) definuje flexibilnú architektúru pre komunikáciu medzi klientmi a servermi. V tejto architektúre klientske a serverové aplikácie interagujú pomocou štandardizovaného rozhrania OPC UA Client/Server API, ktoré izoluje logiku aplikácie od komunikačnej vrstvy. Toto oddelenie umožňuje komunikačnému zásobníku OPC UA konvertovať volania API na správy a naopak, čo umožňuje plynulú výmenu údajov cez rôzne transportné technológie. Architektúra je navrhnutá tak, aby bola nezávislá od technológie, čím sa zabezpečuje dlhodobá prispôsobivosť budúcim komunikačným štandardom [62]. OPC UA dokáže pracovať na počítačoch, tabletoch, mikrokontroléroch a aj na Cloude [63]. Celkovo OPC UA poskytuje robustnú, škálovateľnú a štandardizovanú metódu pre bezpečnú a reálnu komunikáciu priemyselných dát.

2.3.4 Použitia a aplikácie PLC

PLC sú jednou z hlavných súčastí automatických systémov v priemysle [64]. Nachádzajú sa v chemickom, procesnom a výrobnom priemysle, kde je potrebné efektívne a spoľahlivé sekvenčné riadenie [65]. Aplikáciu PLC zariadení vieme chápať ako priame riadenie procesu, kde sa na základe regulátorov a monitorovacích zariadení riadia akčné členy. Vo výrobnom

priemysle, kde sa produkty vyrábajú zo surovín pomocou strojov alebo robotických zariadení, patria medzi hlavné oblasti napríklad textilný a odevný priemysel, výroba papiera, skla, keramiky, ako aj potravinársky a nápojový priemysel. Moderné výrobné systémy čoraz častejšie integrujú automatizačné technológie do každej fázy výrobného procesu. Tieto fázy zahŕňajú manipuláciu s materiálmi, samotné obrábanie, montážne operácie, ako aj záverečnú kontrolu kvality a balenie výrobkov. Vďaka využitiu priemyselných robotov a sofistikovaného riadenia výrobných procesov sa automatizácia stáva nielen efektívnejšou, ale aj výrazne flexibilnejšou. V prípade závodov, ktoré pracujú na princípe chemických procesov, ako sú farmaceutický, petrochemický alebo cementársky priemysel, predstavuje automatizácia kľúčové riešenie pre zabezpečenie presného, spoľahlivého a kvalitného riadenia rôznych fyzikálnych veličín počas celého výrobného cyklu [66].

Závod Volkswagenu v Zwickau používa PLC od spoločnosti Siemens pre riadenie procesu pri výrobe elektrických vozidiel ako napríklad ID.3, Audi Q4 e-tron alebo Cupra Born [67, 68]. Siemens využila aj spoločnosť Patti Engineering pri vyvíjaní linky pre zákazníka z farmaceutického priemyslu. Nový riadiaci systém dokázal počítať produkty, odstarňovať vadné kusy a umožňoval jednoduchšiu údržbu [69]. Spoločnosť Maverick International optimalizovala svoje baliace procesy integráciou PLC systémov od spoločnosti Omron. Toto riešenie zlepšilo kvalitu utesnenia a znížilo množstvo odpadu [70].

2.4 Rozhranie človek - stroj

Rozhranie človek-stroj (*Human-Machine Interface*, HMI) je kľúčovým prvkom v priemyselnej automatizácii a slúži, ako prepojenie medzi ľudskými operátormi a strojmi. Jeho primárnou úlohou je poskytovať rozhranie, prostredníctvom ktorého môžu operátori ovládať, monitorovať a interagovať s automatizovanými systémami. Historicky HMI začínali, ako jednoduché hardvérové panely s fyzickými tlačidlami a informatívnymi svetlami. Vyvinuli sa do kombinácií obrazoviek a tlačidiel, ktoré v súčasnosti môžu byť reprezentované grafickými dotykovými obrazovkami, prípadne web stránkami, ktoré ponúkajú väčšiu flexibilitu a funkčnosť [71].

HMI v súčasnosti nezobrazujú len údaje, ale pomáhajú aj s diagnostikou, alarmami a analýzami. Ich funkcionálnosť sa líši od koncového cieľového používateľa. Pre operátorov robotov sa HMI musia zamerať na vykonávanie úloh – umožňujú jednoduchý prístup k výberu programu, riadeniu cyklov a monitorovaniu systému. V prípade programátorov či personálu údržby sa jedná o prístup k hlbším vrstvám údajov, ako je diagnostika I/O, protokoly chýb a možnosti plnohodnotného manuálneho ovládania. Toto rozlíšenie si vyžaduje prístup založený na rolách a návrh viacvrstvového rozhrania pre použiteľnosť a bezpečnosť.

2.4.1 Súčasná požiadavka priemyslu

Moderné HMI rozhrania musia spĺňať rôzne priemyselné štandardy a smernice, ako napríklad ISA101 pre vysokovýkonné HMI rozhrania a normy IEC, ako napríklad IEC 60204-1 pre bezpečnosť. Tieto smernice kladú dôraz na prehľadnosť, zníženie preplnenosti obrazovky a zobrazenia zamerané na proces. Ďalšou kritickou požiadavkou je responzivnosť, pretože

oneskorenia v spätnej väzbe z HMI môžu viesť k nebezpečným podmienkam alebo prestávkam vo výrobe. Veľmi dôležitá je bezpečnosť proti poruchám zaručená intuitívnymi prvkami, ako núdzové zastavenie, bezpečnostné blokovania a alarmové signály. Podstatnú úlohu zohráva aj prístupnosť, ktorá zabezpečuje, že HMI spĺňa rôzne potreby používateľov, ako je jazyková podpora a nastavenie vizuálneho kontrastu. Do budúcnosti sa štandard ISA101 má v pláne zameriavať aj na rozhranie HMI pre mobilné zariadenia, ktoré sú čoraz častejšie v technickom priemysle. [72–74]

Veľká časť HMI inštalovaných v dnešných závodoch je z hľadiska dizajnu zastaralá a môže zvýšiť pravdepodobnosť incidentu. Tradičná grafika so svojimi často kreatívnymi farebnými paletami často sťažuje operátorom rozlíšiť kritické informácie, ktoré je potrebné okamžite riešiť, od bežného šumu v pozadí správne prebiehajúceho procesu. Aj keď vizuálne zobrazenia procesov môžu priniesť značné výhody, samotný návrh HMI si vyžaduje dôkladné zváženie toho, akým spôsobom, v akom čase a na akom mieste sa údaje na displeji prezentujú. Výzvou zostáva aj správne využívanie farieb, pretože sa čoraz častejšie poukazuje na to, že klasické HMI systémy často ignorujú problémy ako farbosleposť a iné obmedzenia spojené so spracovaním vizuálnych podnetov u používateľov [75].

Použitie webových technológií ako sú HTML, CSS alebo JavaScript, v priemyselných systémoch HMI prináša niekoľko výhod. Ponúkajú nezávislosť od platformy, čo umožňuje spustenie HMI na akomkoľvek zariadení s prehliadačom a prístupom na sieť. To otvára možnosti pre vzdialené monitorovanie a mobilnú dostupnosť. Frameworky ako Node-RED a Flask umožňujú rýchly vývoj vlastných HMI. Node-RED poskytuje rýchle vytváranie dashboardov, zatiaľ čo Flask umožňuje väčšiu kontrolu nad dizajnom a logikou pomocou Pythonu. Tieto nástroje sa dajú ľahko integrovať s priemyselnými systémami prostredníctvom protokolov ako OPC UA alebo MQTT [76, 77]. Kompatibilita s mobilnými zariadeniami umožňuje operátorom a technikom vzdialený prístup k HMI, prijímať alarmy a dokonca ovládať určité funkcie v reálnom čase. Táto flexibilita zlepšuje prevádzkovú efektívnosť a odozvu.



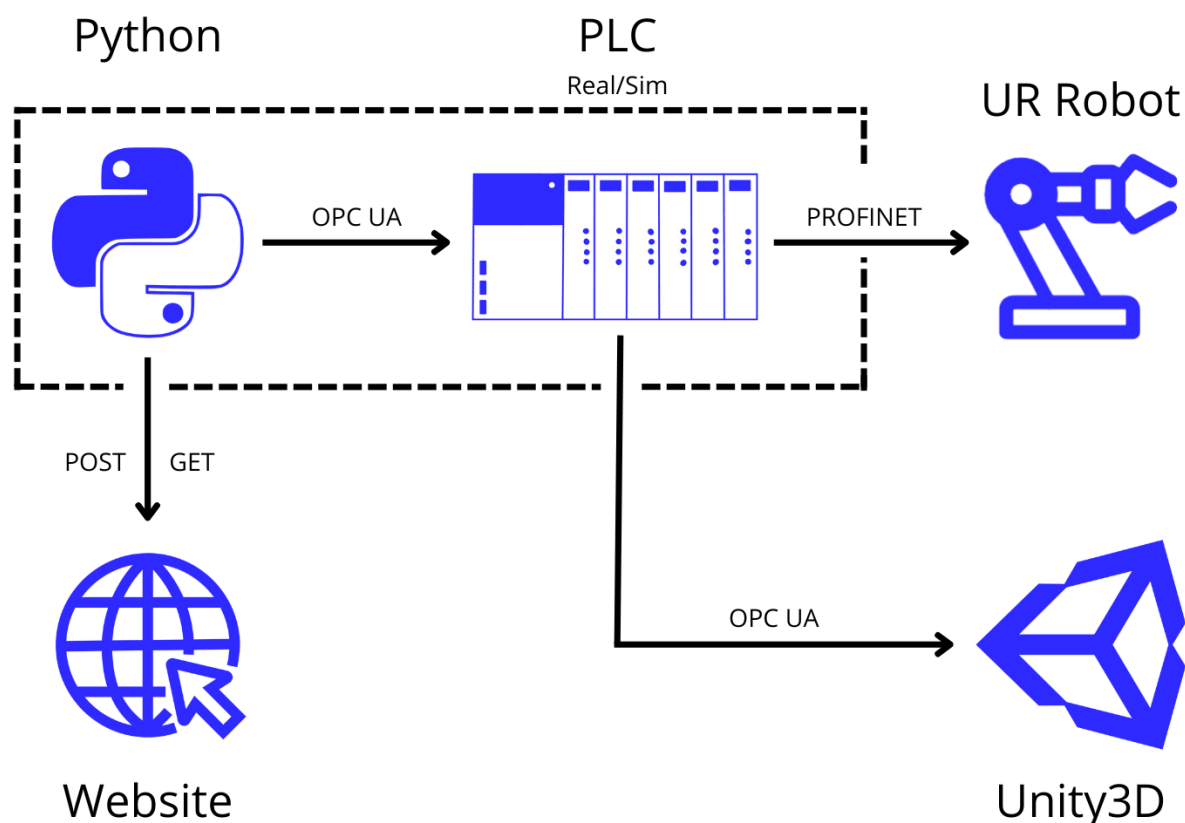
Obr. 5: Človek ovládajúci stroj pomocou HMI [78]

3 FUNKČNÝ BLOK V TIA PORTAL

Kapitola je zameraná na vytvorenie funkčného bloku, ktorý bude základ pre pokročilé riadenie kolaboratívnych robotov Universal Robots pomocou nástroja TIA Portal od spoločnosti Siemens. Výsledné riešenie je modulárne, umožňuje efektívnu výmenu dát s využitím protokolu PROFINET.

3.1 Výsledná architektúra ovládania

Funkčný blok bol navrhnutý univerzálne pre viacero možností ovládania robota. Základnou možnosťou je použitie samotného PLC, programových blokov funkčného bloku a vlastných programových blokov. Rozšírenie zariadenia o vstupové, výstupové či modbus karty je bezproblémové až žiaduce. Ďalšou možnosťou je ovládanie pomocou opcua v kombinácii s Python programom, ktorý umožňuje nielen väčšiu výpočtovú kapacitu ale aj množstvo knižníc. Nadstavba vývoja lokálneho alebo webového rozhrania je taktiež možná. Výhodou FB je jeho možnosť pracovať s reálnym robotom, ale aj v simulácií. Priestor pre škálovateľnosť aplikácie FB je veľký. Výsledná architektúra ovládania je zobrazená na obrázku 6.



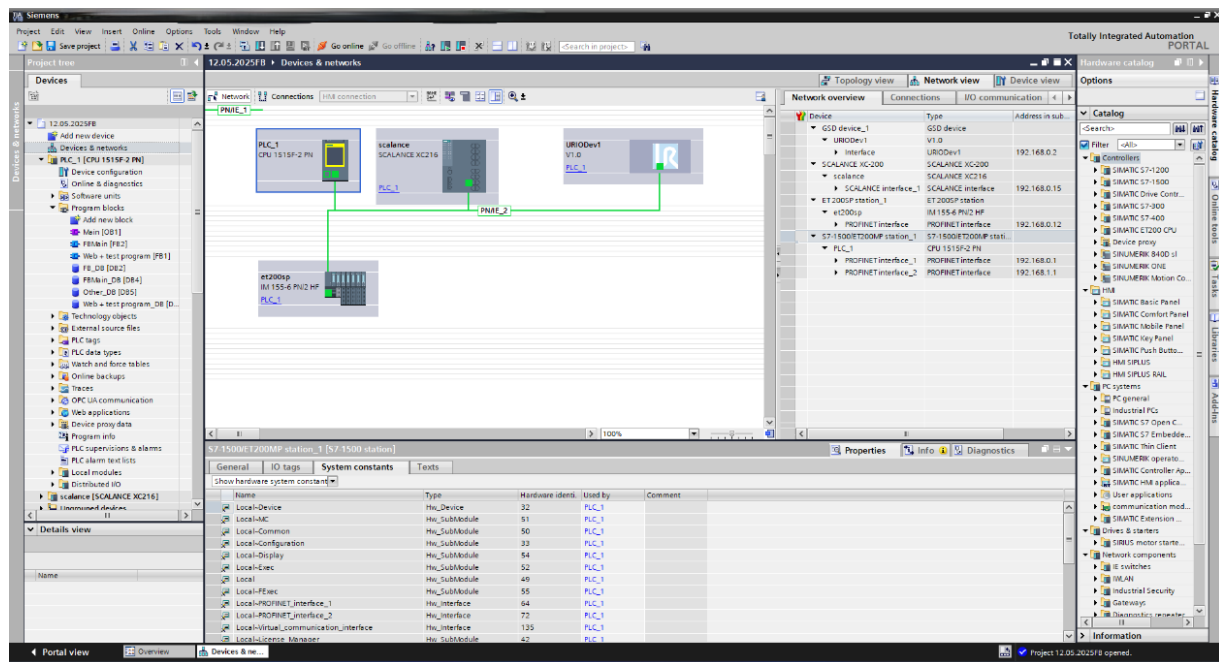
Obr. 6: Architektúra ovládania

3.2 Totally Integrated Automation Portal

Pre konfiguráciu, programovanie a prácu s PLC zariadeniami je potrebný adekvátny softvér. Pre PLC od značky Siemens sa používa softvérový balík Totally Integrated Automation Portal (TIA Portal), ktorý je možné vidieť na obrázku číslo 7. Používa sa hlavne v projektoch priemyselnej automatizácie na vývoj, simuláciu a nasadzovanie aplikácií, ktoré riadia stroje, výrobné linky a robotické systémy [79]. Jeho základné funkcie sú nasledovné:

- Programovanie Siemens PLC
 - Rady S7-1500, S7-1200, S7-400 a S7-300
 - Programovacie jazyky: Ladder Logic, Function Block Diagram, Statement List, Structured Text a GRAPH
- HMI
 - Návrh pomocou integrovaného WinCC
 - Vytváranie obrazovky, tlačidiel, alarmov a ich prepojenie s premennými
- Networking
 - Konfigurácia sietí PROFINET, OPC UA či Ethernet
 - Grafické zobrazenie pripojených zariadení (PLC, HMI, karty, switche, UR IO modul, pohony)
- Diagnostika a monitorovanie
 - Integrované diagnostické nástroje pre detekciu zariadení na sieti
 - Nástroje na ladenie a monitorovanie systému a hodnôt v reálnom čase
- Simulácia
 - Prepojenie s PLCSIM na simuláciu PLC
 - Simulácia HMI
- Modulárna štruktúra projektu
 - Podporuje vytváranie modulárnych a opakovane použiteľných blokov kódu, UDT (user defined datatypes – užívateľom definovaných typov) a globálnych knižníc [80]

TIA Portal má široké využitie a aplikácie ako aj samotné Siemens PLC zariadenia. V rámci celosvetového trhu sú PLC od značky Siemens na prvom mieste a ich podiel je až 30 % [81]. Ich použitie na priemyselnú automatizáciu sú napríklad dopravníkové pásy, baliace stroje, miešacie zariadenia, ale aj riadenie komplexných výrobných liniek. Voľba Siemens série PLC závisí od konkrétnej aplikácie. PLC z najvyššej série S7-1500 sú momentálne najnovšie, najvýkonnejšie a vysoko škálovateľné pričom ponúkajú taktiež pokročilé funkcie ako integrácia s IoT. Nižšie série S7-1200, S7-400 a S7-300 sú taktiež aplikované vo veľkej miere a sú dostatočne výkonné na riadenie jednoduchších procesov. V minulosti sa vyrábala aj séria S7-200, ktorá bola známa svojou kompaktnosťou. V súčasnosti sa s ňou stretne len zriedka, najčastejšie vo vzdelávacích zariadeniach a malých automatizačných projektoch [82].



Obr. 7: Rozhranie programu TIA Portal

3.3 Požiadavky pre FB

Pre správne fungovanie funkčného bloku je potrebné definovanie vstupných, výstupných a taktiež vnútorných požiadaviek. Vstupné a výstupné požiadavky sú zobrazené v tabuľkách 1 a 2.

Tab. 1: Vstupné požiadavky pre FB

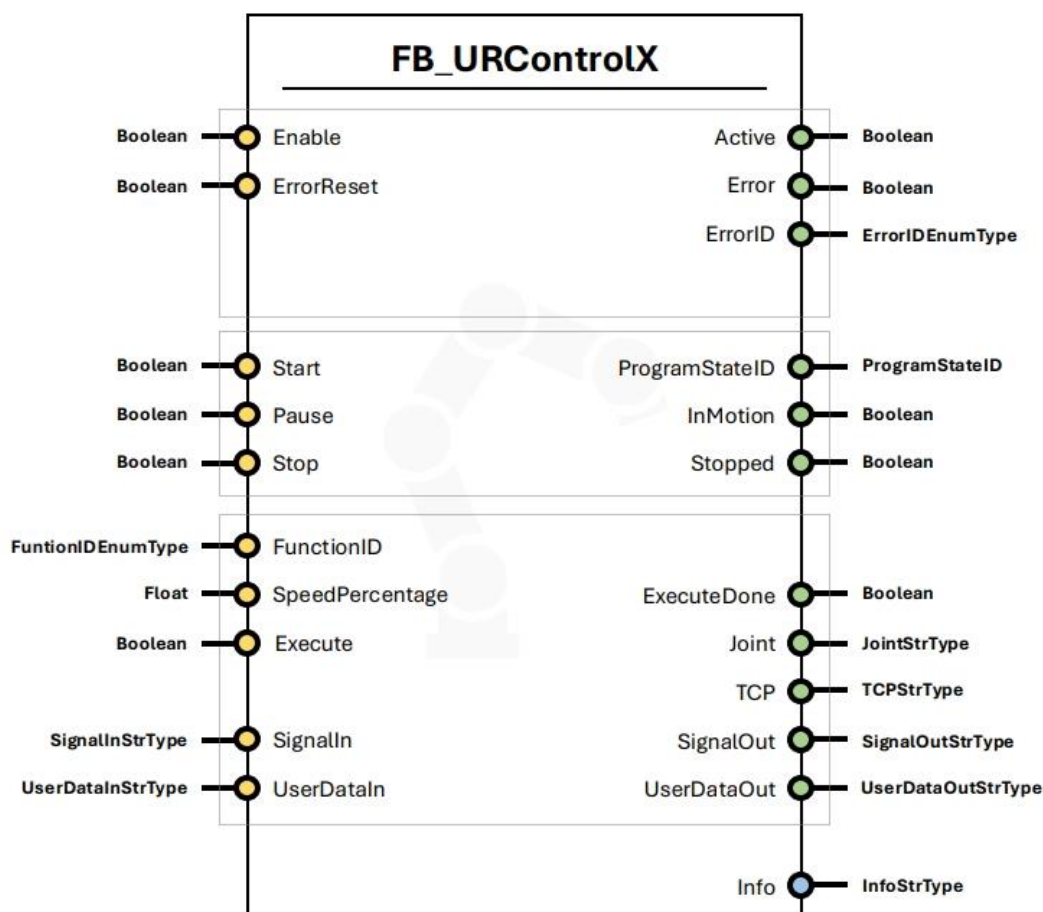
Signál	Dátový typ	Stručný popis
Enable	Bool	pri prevedení stavu na True sa aktivuje funkčný blok; v prípade, že sa hodnota rovná False dôjde k resetovaniu všetkých premenných
ErrorReset	Bool	príkaz na resetovanie chyby indikované výstupným signálom funkčného bloku Error a identifikátorom ErrorID
Start	Bool	príkaz k spusteniu štart v prípade predošlého zastavenia
Pause	Bool	príkaz k prerušeniu programu
Stop	Bool	príkaz k zastaveniu programu
FunctionID	Int	identifikačné číslo funkcie, ktorá sa spustí s využitím príkazu Execute
SpeedPercentage	Real	percento pre bežiaci program bez porušenia bezpečnostných limitov
Execute	Bool	príkaz ku spusteniu funkcie so zvoleným identifikačným číslom

UserDataIn	UserDataType	ovládanie užívateľských signálov nápomocným k modulárnejšej realizácii funkcií; obsahuje 32 adries pre hodnoty bool, 16 adries pre hodnoty int a 16 adries pre hodnoty real
------------	--------------	---

Tab. 2: Výstupné požiadavky pre FB

Signál	Dátový typ	Stručný popis
Active	Bool	identifikácia úspešného nadviazania spojenia s robotom zároveň s aktiváciou FB
Error	Bool	identifikácia prítomnosti erroru
ErrorID	Int	stav 0 znamená žiadny error, stav 1 Safe stop a stav 2 Emergency stop
ProgramStateID	String	môže nadobúdať stavy Offline, Idle, Started, Paused, Stopped alebo Error
InMotion	Bool	identifikácia o pohybe robota
Stopped	Bool	identifikácia o zastavení robota
ExecuteDone	Bool	identifikácia, či sa spustená funkcia (FunctionID) dokončila v poriadku
Joint	UR_3_T2O_Joints	čítanie dát o polohách, rýchlostiach, prúdoch a teplotách jednotlivých kĺbov
TCP	UR_4_T2O_TCP	čítanie dát o polohe, rýchlosti, sile a momente TCP stredového bodu nástroja
UserDataOut	UserDataType	čítanie užívateľských signálov nápomocným k modulárnejšej realizácii funkcií; obsahuje 32 adries pre hodnoty bool, 16 adries pre hodnoty int a 16 adries pre hodnoty real
Info	InfoType	zobrazenie predošlého stavu

Medzi vnútorné požiadavky patrí napríklad error handling. V prípade, že robot narazí a dostane sa tak do Safe Stop-u funkčný blok reaguje adekvátne, zaznamená zmenu a tiež oznamuje error. Okrem tohto prípadu, môže nastať aj Emergency Stop, ktorý je v momente stlačenia Emergency Stop tlačidla na teach pendante robota. Aj v tomto prípade funkčný blok reaguje a oznamuje chybu. Jeho štruktúru je možné vidieť na obázku 8.



Obr. 8: Vizualizácia štruktúry funkčného bloku

3.4 Architektúra riešenia

3.4.1 Použitý softvér

Ako softvér bol použitý Siemens TIA Portal V18, ktorý bol opísaný v kapitole 3.1. Okrem tohto programu bol využitý od spoločnosti Siemens aj S7-PLCSIM Advanced V5.0 pre simuláciu PLC. Priebeh simulácie je opísaný v kapitole 5.1. Funkčný blok bol napísaný v štruktúrovanom texte. Štruktúrovaný text, ako bolo už spomínané v kapitole 2.3.2, je jeden z jazykov podporovaných normou IEC 61131-3, určenou pre programovateľné logické automaty (PLC). Dátové typy spĺňané normou IEC 61131-3 a teda aj štruktúrovaným textom boli taktiež opísané v kapitole 2.3.2. Tento jazyk je vyššej úrovne, ktorý je založený na programovacom jazyku Pascal. Štruktúrovaný text podporuje riadiace štruktúry ako IF – THEN, CASE, WHILE, FOR, REPEAT a definovanie funkcií pomocou FUNCTION [53]. Výhody štruktúrovaného textu sú napríklad moderná syntax, výkonnosť a kompaktnosť v porovnaní s ostatnými programovacími jazykmi používanými v programe TIA Portal.

3.4.2 Štruktúra riešenia FB

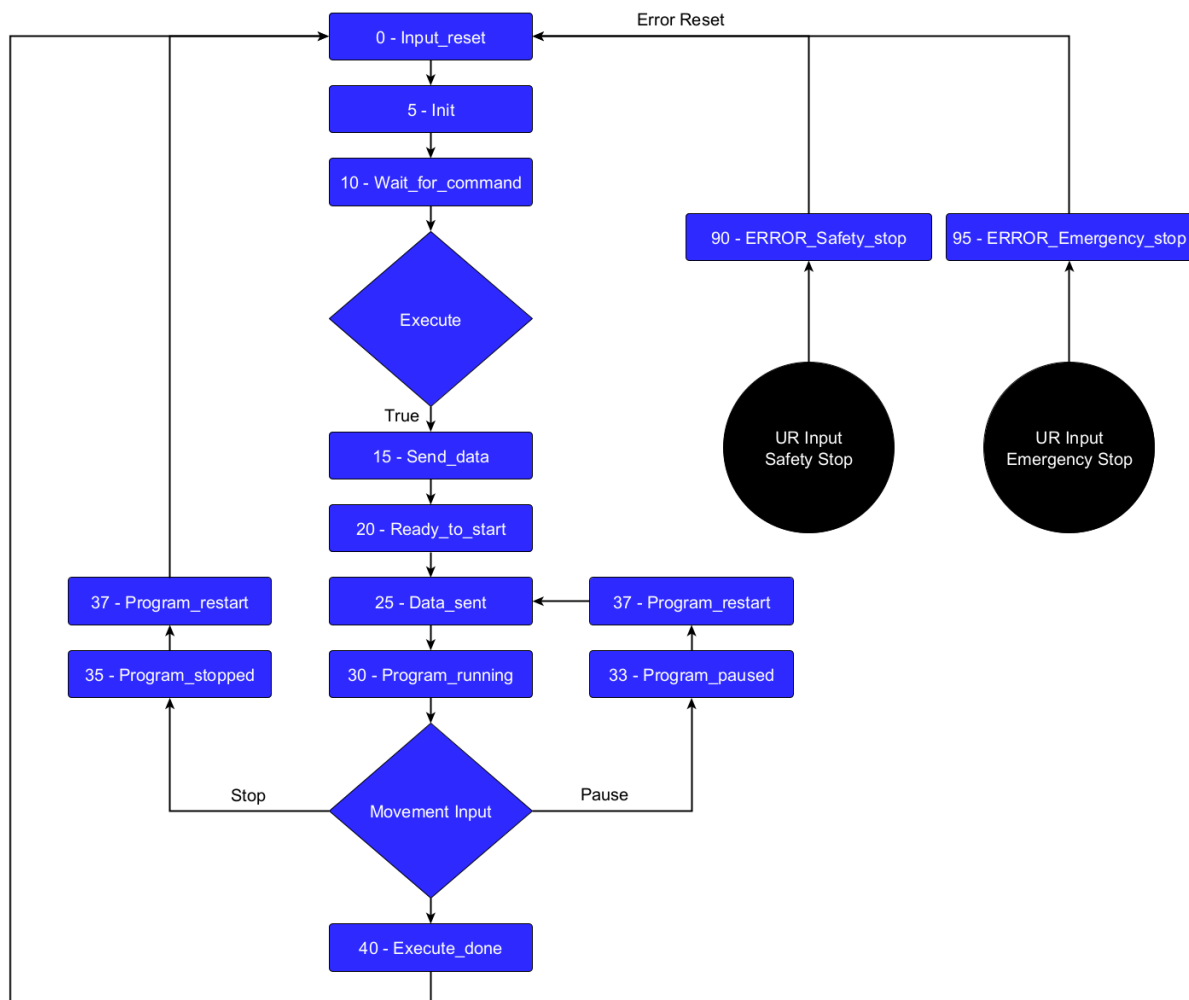
Funkčný blok bol napísaný v jazyku štruktúrovaný text a jeho jadro, ktoré komunikuje a ovláda robota bolo postavené na stavovom automate CASE, ktorý pozostáva zo stavov zobrazených v tabuľke číslo 3.

Tab. 3: Popis jednotlivých stavov v hlavnom stavovom automate riadiaceho funkčného bloku

Stav	Popis stavu
Input_reset	Všetky premenné okrem nastavenej rýchlosti sa zresetujú a program čaká na nastavenie premennej Enable na hodnotu True, následne sa case dostane do stavu Init.
Init	Časovač TON, ktorý trvá pól sekundy a spúšťa sa automaticky po tom, čo sa program dostane do tohto stavu. Po dokončení časovaču, sa overuje pripojenie robota a jeho napájanie pomocou bool hodnoty premennej "URI".State.Robot."PW: Is power on". V prípade hodnoty True sa stavový automat dostane do stavu W_for_cmd.
W_for_cmd	Stav, v ktorom program čaká na nastavenie funkcie a následné spustenie. Je to východzí stav po aktivovaní celého funkčného bloku. Stav sa posunie do stavu Send_data v prípade, že je nastavená FunctionID na hodnotu inú ako je 0 a zároveň je nastavená hodnota Execute na True.
Send_data	Nastáva nastavenie všetkých ovládacích premenných z lokálnych na výstupné. Jednotlivé prepojenia sú popísané v tabuľkách č. 1-3. Okrem týchto ovládacích premenných, ktoré sú konfigurovateľné používateľom, sa nastaví aj FunctionID z lokálnej na výstupnú a taktiež premenná Execute_done_reset na hodnotu False. Následne po odoslaní dát sa spúšťa druhý timer o rovnakej dĺžke ako prvý a po jeho dokončení sa stavový automat dostane do stavu Ready_to_start.
Ready_to_start	Premenná StartUR sa nastaví na hodnotu True a spustí sa ďalší časovač. Po dokončení časovaču sa stavový automat dostáva do stavu Data_send.
Data_sent	Program čaká na informáciu od robota o tom, že začal vykonávať funkciu. Následne sa presunie do stavu Program_running.
Program_running	Stavový automat čaká na spätnú väzbu od robota, ktorý oznámi vykonanie funkcie – nastaví sa hodnota ExecuteDone na True a následne sa stavový automat dostáva do stavu Execute_Done. V prípade pozastavenia alebo zastavenia funkcie sa case dostane do adekvátneho stavu – Program_paused alebo Program_stopped.

Program_paused	Môže nastať len pri pozastavení programu pri vykonávaní funkcie robotom a má len jednu úlohu ako hovoriť aj názov. Čaká na znovu spustenie programu a následne nastaví hodnotu case na Program_restart.
Program_stopped	Má analogickú funkcionálnosť ako Program_paused, avšak v prípade znovu spustenia sa zmení priebeh v Program_restart.
Program_restart	Rozhoduje o následnej hodnote case vzhľadom na predošlý stav. V prípade, že má premenná Pause hodnotu True a teda predošlý stav bol Program_paused sa hodnota stavového automatu nastaví na Data_sent. Naopak ak predošlý stav bol Stop, case sa nastavuje na hodnotu Input_reset.
Execute_done	Premenné Execute a StartUR sa nastavujú na False, ExecuteDoneReset na True a následne v prípade, že má premenná ExecuteDone hodnotu False, stavový automat prejde do stavu Input_reset a celý proces sa môže zopakovať.
ERROR_safety_stop	Nastaví Enable na False a po tom, čo premenná ErrorReset nadobude hodnotu True nastaví case na InputReset.
ERROR_emergency_stop	Funguje nalogicky ako ERROR_safety_stop.

Vo všetkých stavoch sa nachádza podmienka, ktorá nastaví case na Input_reset v prípade, že sa hodnota Enable zmení na False. Okrem stavového automatu sa v programe nachádzajú aj podmienky pre zistenie safety a emergency zastavenia a taktiež prepočet rýchlosti z percent na metre za sekundu. Keďže PLC má obnovovaciu frekvenciu 1,2 až 2 ms, frekvencia robota je 2 ms a nemajú synchronizovaný systémový čas, je potrebné zamedziť nesprávnemu načítaniu alebo zápisu dát, preto program obsahuje už spomínaný časovač. Priradenia premenných pre zápis a čítanie hodnôt kontinuálne mimo stavového automatu je možné vidieť v tabuľkách číslo 1 až 3. Vývojový diagram priebehu programu je zobrazený na obrázku číslo 9.



Obr. 9: Vývojový diagram priebehu FB

Tab. 4: Adresy pre kontinuálne zapisované hodnoty

Lokálna adresa PLC	Výstupná adresa PLC
"FB_DB".Start	"URO"."Reg 2".Bits.Register[0]
"FB_DB".Pause	"URO"."Reg 2".Bits.Register[1]
"FB_DB".Stop	"URO"."Reg 2".Bits.Register[2]
"FB_DB".ExecuteDoneReset	"URO"."Reg 2".Bits.Register[3]
"FB_DB".StartUR	"URO"."Reg 2".Bits.Register[4]
"FB_DB".SpeedActual	"URO"."Reg 2".Floats.Register[4]

Tab. 5: Adresy pre kontinuálne čítané hodnoty vrátane informácií o kĺboch a TCP

Lokálna adresa PLC	Vstupná adresa PLC
"FB_DB".FeedbackFunctionDone	"URI".Bits.Register[32]
"FB_DB".FeedbackProgramRunning	"URI".Bits.Register[33]
"FB_DB".Joint."Joint position [rad]"[0-5]	"URI".Joints."Joint position [rad]"[0-5]
"FB_DB".Joint."Joint velocity [rad/s]"[0-5]	"URI".Joints."Joint velocity [rad/s]"[0-5]
"FB_DB".Joint."Joint current [A]"[0-5]	"URI".Joints."Joint current [A]"[0-5]

"FB_DB".Joint."Joint temperature [°C]"[0-5]	"URI".Joints."Joint temperature [°C]"[0-5]
"FB_DB".Joint."Joint mode"[0-5]	"URI".Joints."Joint mode"[0-5]
"FB_DB".Joint.Reserved_9	"URI".Joints.Reserved_9
"FB_DB".TCP."TCP Position (X,Y,Z) [m]"[0-2]	"URI".TCP."TCP Position (X,Y,Z) [m]"[0-2]
"FB_DB".TCP."TCP Position (RX,RY,RZ) [rad]"[0-2]	"URI".TCP."TCP Position (RX,RY,RZ) [rad]"[0-2]
"FB_DB".TCP."TCP velocity (X,Y,Z) [m/s]"[0-2]	"URI".TCP."TCP velocity (X,Y,Z) [m/s]"[0-2]
"FB_DB".TCP."TCP velocity (RX,RY,RZ) [rad/s]"[0-2]	"URI".TCP."TCP velocity (RX,RY,RZ) [rad/s]"[0-2]
"FB_DB".TCP."TCP force (X,Y,Z) [N]"[0-2]	"URI".TCP."TCP force (X,Y,Z) [N]"[0-2]
"FB_DB".TCP."TCP torque (X,Y,Z) [Nm]"[0-2]	"URI".TCP."TCP torque (X,Y,Z) [Nm]"[0-2]
"FB_DB".TCP."TCP force scalar [N]"	"URI".TCP."TCP force scalar [N]"

Tab. 6: Adresy pre kontinuálne čítané hodnoty Bool, Int a Real

Lokálna adresa PLC	Vstupná adresa PLC
"FB_DB".UserDataOutput."Bool".Register[0-31]	"URI".Bits.Register[0-31]
"FB_DB".UserDataOutput."Int".Register[0-15]	"URI".Ints.Register[0-15]
"FB_DB".UserDataOutput."Real".Register[0-15]	"URI".Floats.Register[0-15]

V tabuľkách 4 až 6 je možné vidieť adresy premenných, ktoré sú vyhradené pre používateľa na ich následné využitie v UR funkciách.

Tab. 7: Adresy pre zapisované hodnoty Bool

Lokálna adresa PLC pre hodnoty Bool	Výstupná adresa PLC pre hodnoty Bool
"FB_DB".UserDataInput."Bool".Register[0-31]	"URO"."Reg 1".Bits.Register[0-31]

Tab. 8: Adresy pre zapisované hodnoty Int

Lokálna adresa PLC pre hodnoty Int	Výstupná adresa PLC pre hodnoty Int
"FB_DB".UserDataInput."Int".Register[0-11]	"URO"."Reg 1".Ints.Register[0-11]
"FB_DB".UserDataInput."Int".Register[12]	"URO"."Reg 2".Ints.Register[0-3]

Tab. 9: Adresy pre zapisované hodnoty Real

Lokálna adresa PLC pre hodnoty Real	Výstupná adresa PLC pre hodnoty Real
"FB_DB".UserDataInput."Real".Register[0-11]	"URO"."Reg 1".Floats.Register[0-11]
"FB_DB".UserDataInput."Real".Register[12]	"URO"."Reg 2".Floats.Register[0-3]

3.4.3 Konfigurácia komunikácie PLC s UR robotom

Na komunikáciu medzi programovateľným automatom a robotom od firmy Universal Robots bol použitý PROFINET, ktorý je popísaný v kapitole 2.3.3. Pre úspešné nastavenie PROFINET komunikácie treba mať PLC firmvér verzie 4.0 alebo vyšší a prebieha nasledovne.

- Na teach pendante je potrebné v karte inštalácia, podkarte Fieldbus, povoliť *PROFINET IO Device* a následne uložiť inštaláciu. Následne by sa mal zmeniť sivý oznamovací krúžok na žltý, a taktiež by sa mal zmeniť stav z *Disabled* na *Running*.
- V programe TIA Portal sa na ľavej strane nachádza Project view, v ktorom sa nachádza karta *Online access*. Tu je potrebné zvoliť možnosť *Update accessible devices*.
- Pre úspešnú identifikáciu robota, ktorého chceme spojiť s PLC je potrebné priradenie ID adres a mien. V karte pripojeného robota je potrebné otvoriť *Online & diagnostics*, následne rozkliknúť *Functions & Assign IP adress*. V tejto karte je potreba zadať IP adresu, masku siete a následne potvrdiť tlačidlom *Assign IP adress*. Následne je taktiež nutné rozkliknúť *Diagnostics, Assign name*, zadať požadované meno a potvrdiť ho.
- Nasleduje import GSD súboru od Universal Robots. Je možné tento súbor nájsť na ich stránkach. Po stiahnutí je potrebné v karte *Options* zvoliť možnosť *Manage general station description files (GSD)*, zvoliť stiahnutý súbor a potvrdiť inštaláciu.
- Pre správne prepojenie je potrebné v záložke *Devices & networks* lokalizovať PLC, UR I/O zariadenie, a prepojiť ich pomocou ich portov označených zeleným štvorcikom. Pre úspešné prepojenie stačí kliknúť a podržať jeden z nich, následne premiestniť kurzor na druhý a pustiť.
- Pridanie všetkých modulov UR I/O zariadenia nenastáva automaticky, ale je potrebné ich pridať manuálne. Po rozkliknutí zariadenia v *Network view* sa moduly zobrazia v katalógu na pravej strane. Dvojklikom jednotlivých modulov sa pridajú. Bez týchto modulov sa nedá pristupovať k vstupným a výstupným adresám robota.
- Ďalej treba rozkliknúť UR I/O zariadenie v *Device view*, kartu *Ethernet Adresses* a vybrať *Use IP protocol*. Nastaviť rovnakú IP adresu, zrušiť výber *Generate PROFINET device name automatically* a zadať rovnaké meno, aké sa zadalo online zariadeniu v treťom bode.
- V *External source files* je potrebné dvojkliknúť *Add new external file* pre otvorenie *UR_datastruct.udt* taktiež stiahnutého zo stránok Universal Robots. Následne je potreba importovania dátových typov a zvolenie *Generate block from source*. Potom je možné vytvoriť vo vlastnom dátovom bloku tagy, ktorým sa priradí dátový typ *UR_T2O* a *UR_O2T*. Po vytvorení tagov s týmito dátovými typmi sa pod obidvomi automaticky vytvoria všetky vstupné a výstupné adresy. Dátový typ *UR_T2O* reprezentuje vstup do PLC a teda výstup z robota pričom obsahuje mnoho informácií ako napríklad stav a safety robota, rýchlosti, zrýchlenia, prúdy a teploty jednotlivých kĺbov, pozíciu rýchlosť a silu TCP bodu, ale aj výstupné hodnoty Bool, Int a Float. Analogicky dátový typ *UR_O2T* reprezentuje výstupy z PLC čiže vstupné adresy do robota a disponuje adresami pre dátové typy Bool, Int a Float.
- Nasleduje nahratie programu do PLC, overenie spojenia medzi blokmi, tagmi a modulmi. V prípade úspešného postupu a spojenia sa tieto prepojenia sfarbia na zeleno, ako aj informatívny krúžok na teach pendante robota. [83]

3.4.4 Štruktúra programových blokov v TIA Portal

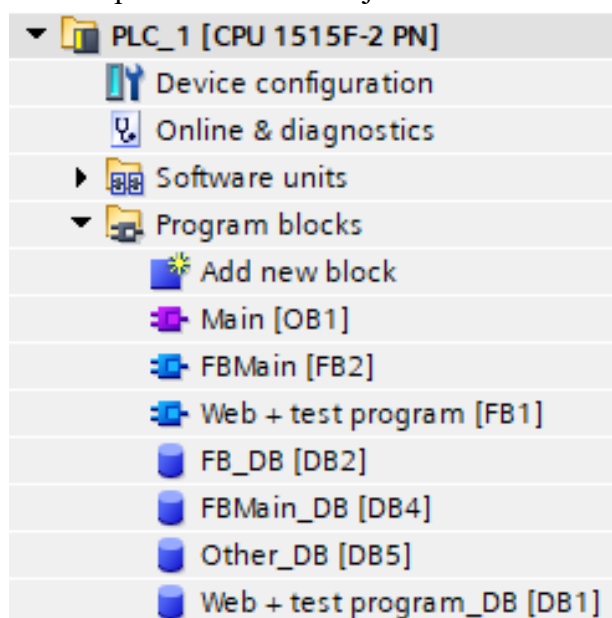
Funkčný blok bol tvorený modulárne s cieľom možného importovania programových blokov. Pre správne fungovanie je potrebné do projektu v TIA Portal importovať nasledovné programové bloky:

- FBMain – Program funkčného bloku
- FBMainDB – Databáza slúžiaca k vnútorným procesom ako je case a časovač
- FB_DB – Databáza všetkých ostatných potrebných premenných

Ovládanie pomocou webového HMI zabezpečujú bloky:

- Web + test program
- Web + test program_DB
- Other_DB

Nevyhnutné je pridanie FBMain a Web + test program do programového bloku Main na jednu zo sietí. V prípade, že postup nebude dodržaný, funkčný blok nebude pracovať správne, prípadne vôbec. Zobrazenie v aplikácii TIA Portal je na obrázku 10.



Obr. 10: Štruktúra programových blokov v TIA Portal

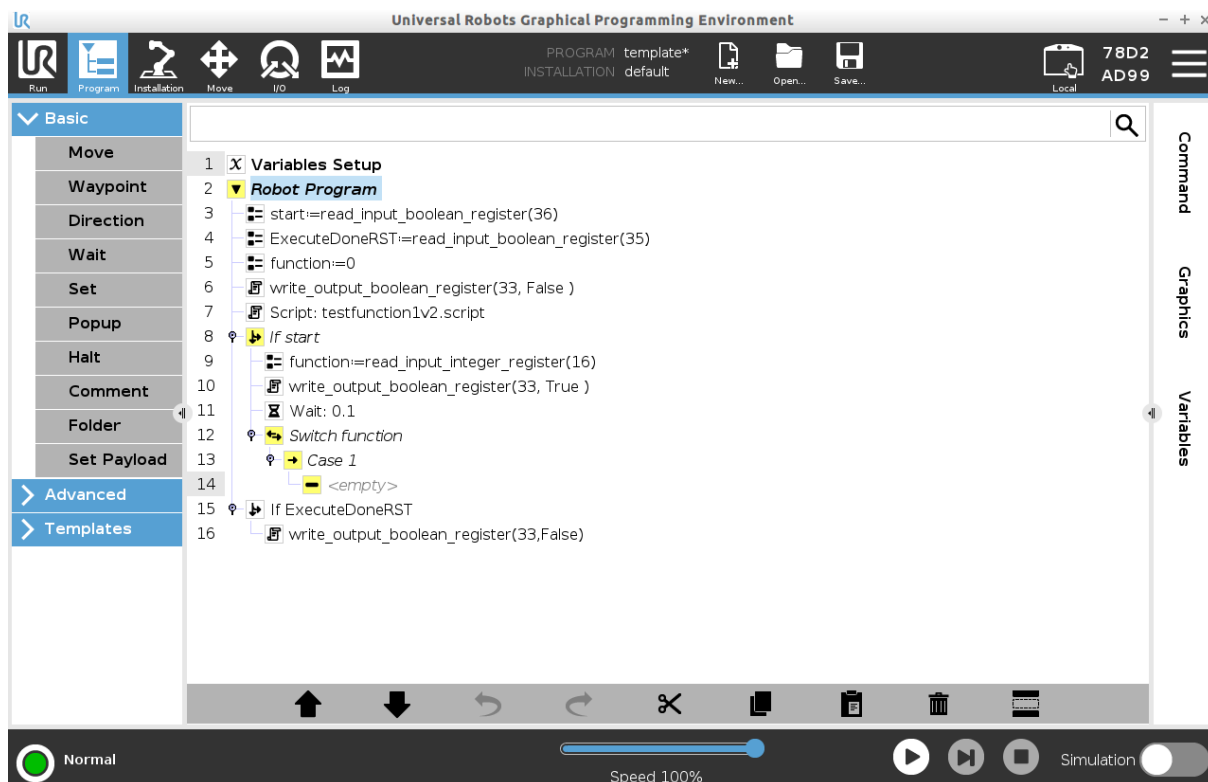
3.4.5 Podmienky tvorenia UR programu

Tvorenie UR programu, ktorý bude správne spolupracovať s funkčným blokom môže mať dva postupy. Prvý postup je len rozširovanie programu do predvyplnenej šablóny o vlastné funkcie a ich následné volanie. Táto šablóna sa dá nahrat' do teach pendantu pomocou USB kľúča. Druhá možnosť je manuálne zakomponovanie podmienok pre spoluprácu. Podmienky sú nasledovné:

- Vytvorenie premennej function a následné priradenie hodnoty 0
- Čítanie a zápis hodnoty Bool na registri 36 (napríklad do premennej start)
- Čítanie a zápis hodnoty Bool na registri 35 (napríklad do premennej ExecuteDoneRST)
- Zápis outputu Bool na registri 33 o hodnote False
- Pridanie skriptu

- Vytvorenie podmienky IF start
 - Zápis hodnoty Int z registru 16 do premenej function
 - Zápis outputu Bool na registri 33 o hodnote True
 - Zastavenie programu na 0,1 sekundy
 - Podmienka Switch s vlastnými funkciami
- Vytvorenie podmienky IF
 - Zápis outputu Bool na registri 33 o hodnote True

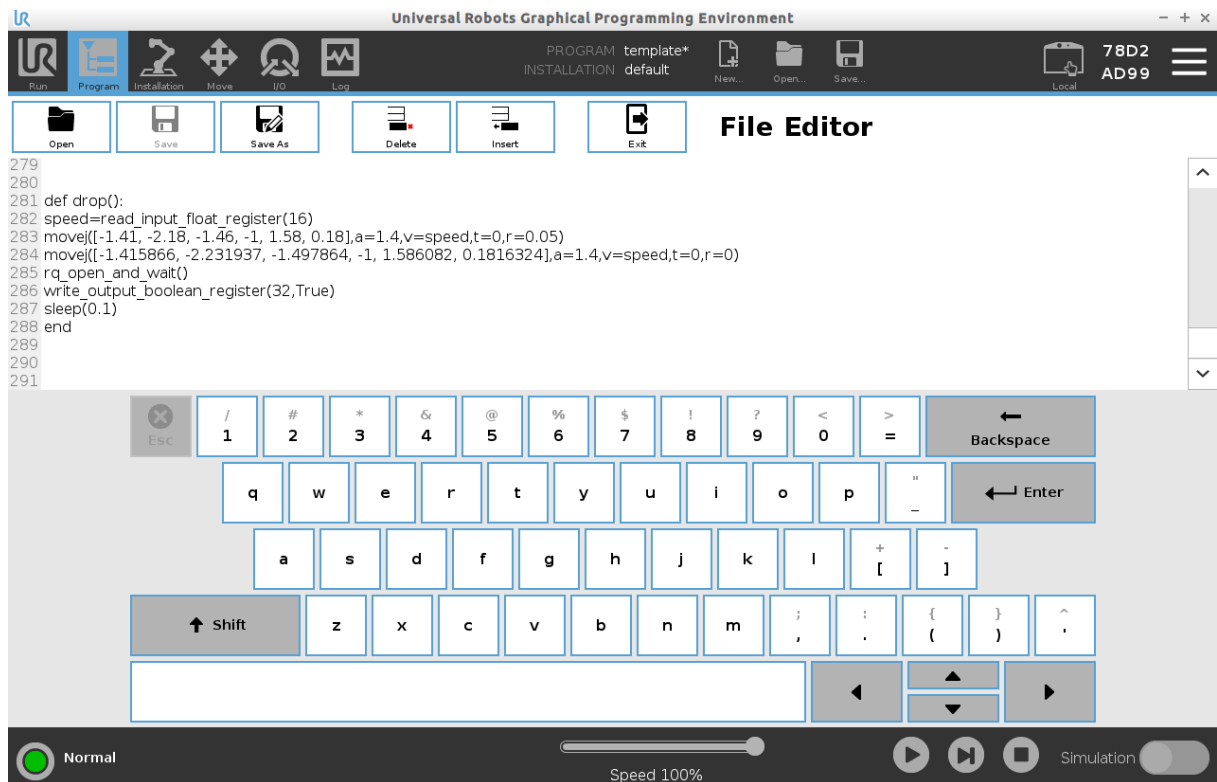
V obidvoch prípadoch nastane rovnaký východzí program, ktorý je možné vidieť na obrázku 11.



Obr. 11: Šablóna UR programu

Vytvorenie function je potrebné, pretože URScript nedokáže dynamicky vytvárať premenné v priebehu programu a následne by nebol možný prechod do danej funkcie automatu case. Hodnota na registri 36 určuje, kedy je možné začať vykonávať funkciu a hodnota na registri 35 poskytuje spätnú väzbu ohľadom ukončenia case automatu v PLC. Zápis Bool na registri 33 naopak zabezpečuje spätnú väzbu o vykonaní funkcie pre PLC. Zastavenie programu tu plní rovnakú funkciu ako časovač v PLC.

Po tomto bode nastáva úprava kódu pre individuálne použitie. Spúšťané funkcie sa môžu buď písať priamo do jednotlivých stavov ako kombinácia viacerých riadkov pohybov a akcií, alebo je možné pridanie funkcie, ktorá je už definovaná v skripte pred podmienkou IF. V obidvoch prípadoch je potrebné na konci každej funkcie nastaviť výstupnú hodnotu Bool registra 33 na True a následne uspať robota na 0,1 sekundy. Tieto dve funkcie zabezpečujú spätnú väzbu pre PLC, bez ktorej by nebolo možné spustenie ďalšej funkcie. Príklad takejto funkcie je možný vidieť na obrázku 12.



Obr. 12: Příklad UR funkcie

3.4.6 Ovládanie robota

Východzí stav PLC programu je Input_reset, z ktorého sa dostane po aktivácii funkčného bloku pomocou premennej Enable. Po aktivácii sa presunie do stavu Init, kde sa nastaví premenná Start na hodnotu True, spustí sa časovač a program čaká na spätnú odozvu od robota, pričom kontroluje jeho napájanie. Premenná Start je monitorovaná robotom a odozva v prípade Start = True je spustenie UR programu. V prípade, že sa UR program úspešne spustil, stavový automat na PLC prechádza do nasledujúceho stavu Wait_for_command. V tomto stave zotrvá až do zadania FunctionID a následného spustenia pomocou premennej Execute, po ktorom sa presúva do stavu Send_data. Ak sa nenastaví FunctionID, ale nastaví sa hodnota Execute na True, robot nastaví premennú Execute naspäť na False a zotrvá v stave Wait_for_command. V stave Send_data sa najprv nastaví hodnota ExecuteDoneReset na False, pričom sa jedná o potvrdenie spätnej väzby pre robota a následne sa všetky premenné patriace pod UserDataInput nakopírujú z lokálnych na výstupné. Ako posledné sa v tomto stave odošle FunctionID a spustí sa časovač. Po jeho ukončení sa stavový automat presúva do stavu Ready_to_start, kde sa nastaví hodnota premennej Start_UR na True, spustí sa ďalší časovač a následne sa automat presunie do stavu Data_sent. V stave Data_sent program čaká na spätnú väzbu od robota, či sa program dostal do vnútornej IF podmienky. Ak je podmienka FeedbackProgramRunning splnená, program sa dostáva do stavu Program_running.

Zatiaľ čo prebieha tento proces na PLC, program v robotovi je nasledovný. Cyklicky číta hodnotu Start, ExecuteDoneRST a definuje premennú Function. Okrem toho je tu potrebné mať zakomponovaný skript, ktorý obsahuje konkrétne funkcie, ktoré následne ovládajú jednotlivé pohyby robota. V prípade, že sa hodnota Start zmení na True, program sa presunie do vnútornej

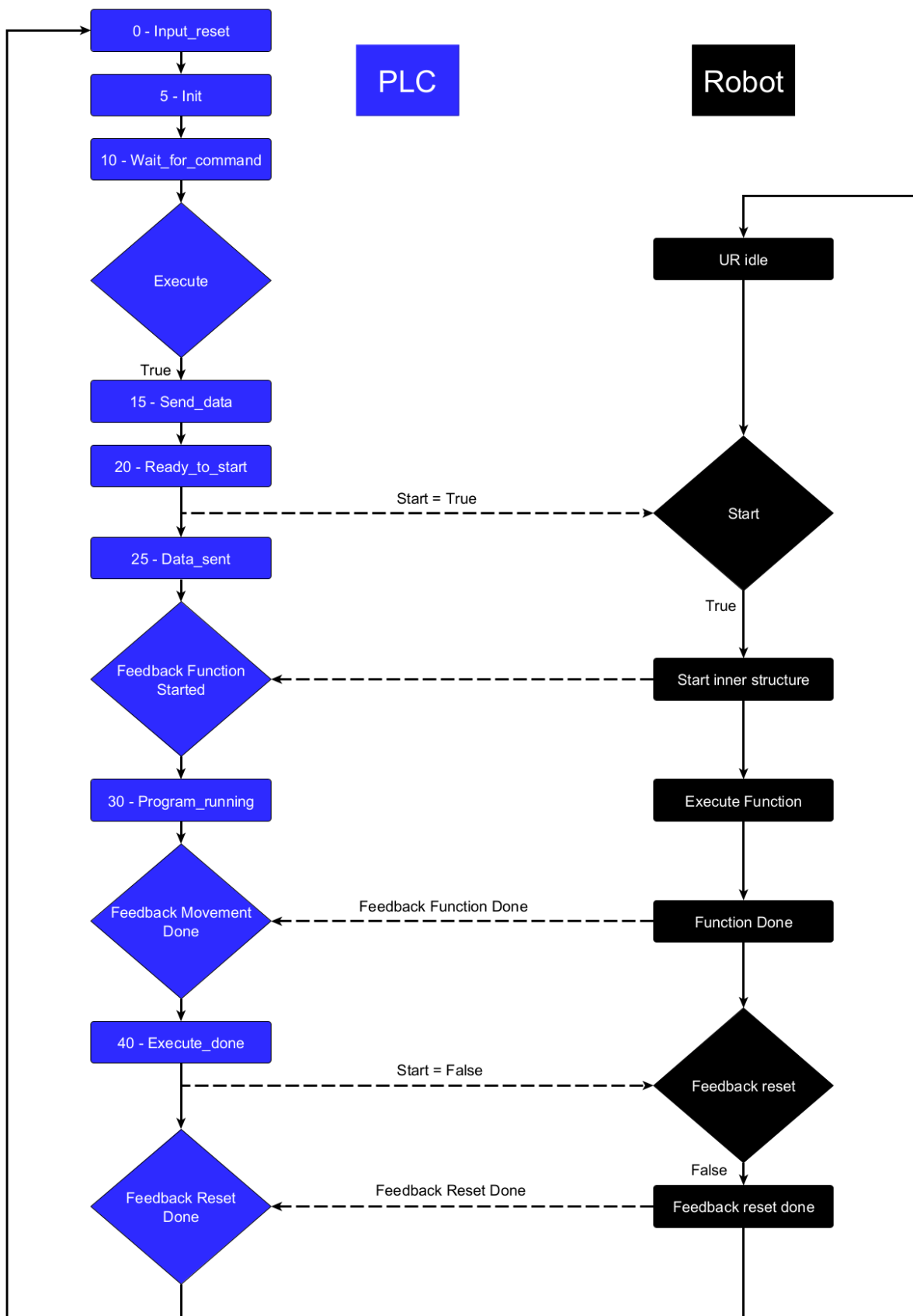
podmienky IF, prečíta hodnotu Function a zapíše hodnotu Bool adresy 32 na True, ktorá je čítaná PLC programom. Následne program počká 0.1 sekundy. Nasleduje vykonanie zadanej funkcie ktoré sa vykonáva pomocou Switch funkcie. Podľa hodnoty Function sa spustí jeden zo stavov, obsahujúci už spomínanú funkciu. Funkcia nie je nijak limitovaná, dokážeme v nej čítať jednotlivé registre, ktoré sme odoslali v stave Send_data na PLC. Pre správne fugovanie je avšak potrebné na koniec funkcie pridať príkaz write_output_boolean_register(32, True), čím poskytujeme spätnú väzbu pre PLC o úspešnom dokonaní funkcie a taktiež príkaz sleep(0.1), ktorý zabezpečí dostatok času na prečítanie spätnej väzby. Po vykonaní funkcie sa program vráti do hlavného cyklu.

V stave Program_running PLC čaká na informáciu od robota o ukončení vykonávania funkcie. Jedná sa o premennú ktorá monitoruje už spomínanú bool adresu číslo 32. V momente, keď adresa nadobudne hodnotu True sa stavový automat presunie do Execute_done. Nastaví premenné Execute a StartUR na False a ExecuteDoneReset na True. Následne čaká na opätovnú spätnú väzbu od robota, ktorý potvrdzuje úspešnú komunikáciu.

Robot po načítaní hodnoty ExecuteDoneRST True nastaví bool adresu 32 na False. Týmto krokom je ukončený proces vykonania funkcie zo strany robota, celý program sa cyklicky spúšťa znova a opätovne čaká kým premenná Start nadobudne hodnotu True.

PLC po zistení FeedbackFunctionDone (adresa bool 32) hodnoty True presunie automat do počiatočného stavu Input_reset. Vzhľadom na predpoklad, že v priebehu nezmeníme hodnotu Enable na False a robot sa neodpojí, stavový automat prechádza do stavu Init a následne do Wait_for_command, v ktorom zotrvá až do spustenia ďalšej funkcie.

Okrem tohto neprerušeného spustenia funkcie je možné funkciu počas vykonávania pozastaviť pomocou Pause alebo zastaviť pomocou premennej Stop. V prípade nastavenia hodnoty Pause na True v priebehu procesu funkcie na UR robotovi, sa celý program pozastaví a spustí až po následnom manuálnom nastavení premennej Start na hodnotu true. Premenná Stop nám zabezpečí zastavenie vykonávanej funkcie a po opätovnom znovuspustení sa zvyšok funkcie nevykoná a program sa dostane do stavu Wait_for_command. Ak sa nastaví hodnota Enable na False v ktoromkoľvek okamžiku, robot sa zastaví a funkčný blok sa deaktivuje. Po opätovnej aktivácii sa dostane do stavu Wait_for_command a zvyšok funkcie sa nevykoná. Schéma popisujúca celkové ovládanie robota cez PLC je zobrazená na obrázku číslo 13.



Obr. 13: Vývojový diagram ovládania UR robota cez PLC

4 WEBOVÉ HMI

Intuitívne ovládanie kolaboratívneho robota UR10e s využitím komunikačného protokolu OPC UA pre bezpečnú výmenu dát zabezpečuje webové rozhranie. Okrem ovládania kolaboratívneho robota, webová stránka slúži aj na zobrazovanie aktuálnych informácií robota ako sú vlastnosti klbov, errorry a zobrazovanie fotografií, ktoré boli vytvorené pripojenou kamerou Basler.

4.1 Python

Python je výkonný programovací jazyk, ktorý ponúka pokročilé dátové štruktúry na vysokej úrovni a zároveň poskytuje prehľadný a účinný spôsob, ako využívať objektovo orientované programovanie. Vďaka čistej syntaxi, dynamickému typovaniu a interpretovanému spracovaniu je Python skvelou voľbou pre skriptovanie aj rýchly vývoj aplikácií naprieč rôznymi platformami a oblasťami použitia [84]. Na programovacom jazyku Python sa začalo pracovať koncom roka 1989 a následne po dvoch rokoch práce v roku 1991 bol zverejnený Guidom van Rossumom pri jeho pôsobení v národnom výskumnom ústave pre matematiku a informatiku v Holandsku. Rossum sa snažil vytvoriť nástupcu pre jazyk ABC, pretože nebol spokojný s jeho schopnosťou rozvíjať sa. Pôvodne sa malo jednáť o jazyk špecifický pre operačný systém Amoeba, no zovšeobecnený jazyk dával väčší zmysel a tým sa napokon aj stal [85]. V priebehu rokov si prešiel mnohými verziami a aktuálne najnovšia je 3.13.3 [86]. Jednou z najsilnejších vlastností Pythonu je jeho rozsiahla štandardná knižnica a komunita. Podporuje viacero programovacích metodík vrátane procedurálneho, objektovo orientovaného a funkcionálneho programovania. Python sa široko používa v rôznych oblastiach, ako je vývoj webových stránok, analýza dát, umelá inteligencia, strojové učenie, vedecké výpočty, automatizácia a skriptovanie. Populárne *frameworky* ako Django a Flask robia vývoj webových stránok efektívnym a škálovateľným, zatiaľ čo knižnice ako NumPy, pandas, TensorFlow, PyTorch alebo LiteLLM sa používajú v aplikáciách dátovej vedy a umelej inteligencie. Flexibilita Pythonu ho tiež robí ideálnym pre rýchle prototypovanie a vývoj. Vďaka svojej popularite a širokému použitiu je Python podporovaný na väčšine platforiem ako je Windows, Mac OS či Linux a dobre sa integruje s inými jazykmi a nástrojmi [87]. Stal sa jedným z najžiadanejších programovacích jazykov na svete [88] a často sa odporúča ako prvý jazyk pre začínajúcich programátorov pre jeho čistú syntax a čitateľnosť kódu.

V robotike má jazyk Python významné miesto pre jeho *open-source frameworky* ako napríklad Freeopcua, ktorý je popísaný v nasledujúcej podkapitole, alebo ROS (Robot Operating System). ROS ponúka kolekciu knižníc a nástrojov na vytváranie robotických aplikácií na viacerých platformách. Má modulárnu architektúru s funkciami rozdelenými na menšie opakovateľne použiteľné zložky označované, ako *uzly*. Taktiež poskytuje hardvérovú abstrakciu, ktorá umožňuje fungovanie kódu na rôznych zariadeniach bez nutnosti zmien [89].

4.1.1 Moduly a knižnice použité pre vývoj aplikácie

Flask je populárny mikro-webový *framework* založený na jazyku Python, ktorý je používaný na vytváranie webových aplikácií, API a služieb. Je klasifikovaný ako *mikro-framework*, pretože štandardne neobsahuje určité funkcie, ako sú vrstvy abstrakcie databázy alebo overovanie formulárov. Namiesto toho poskytuje základné prvky pre vytváranie webových aplikácií a umožňuje vývojárom integrovať ďalšie funkcie podľa potreby. Flask je postavený na dvoch hlavných komponentoch: Werkzeug, ktorý poskytuje sadu nástrojov WSGI (*Web Server Gateway Interface*), a Jinja2, šablónovací *engine* na vykresľovanie HTML stránok [77]. Flask je všeobecne cenený pre svoju jednoduchosť, flexibilitu a jednoduché používanie. Umožňuje rýchly vývoj s minimálnym nastavením a je ideálny pre malé až stredne veľké projekty alebo na prototypovanie. Jednoduchú aplikáciu Flask je možné vytvoriť len v niekoľkých riadkoch kódu, no *framework* je dostatočne výkonný na to, aby podporoval aj zložitejšie aplikácie. Flask podporuje *URL*, spracovanie požiadaviek, *RESTful API* a správu relácií a možno ho ľahko rozšíriť o množstvo doplnkov pre úlohy, ako je autentifikácia používateľov, integrácia databázy (napr. SQLAlchemy) a spracovanie formulárov [90]. Jednou z najväčších výhod Flasku je jeho komunita a ekosystém. K dispozícii je rozsiahla dokumentácia a množstvo návodov a príkladov. Flask sa často používa v kombinácii s *front-end* technológiami, ako sú HTML/CSS/JavaScript [91], a možno ho nasadiť na rôzne platformy alebo lokálne pre interné nástroje. Jeho minimalistický prístup dáva vývojárom plnú kontrolu nad štruktúrou aplikácie.

Functools je štandardný knižný modul v jazyku Python, čo znamená, že je vstavaný a nie je potrebné ho inštalovať samostatne. Poskytuje sadu funkcií vyššieho rádu a nástrojov na prácu s funkciami a volateľnými objektmi, vďaka čomu je obzvlášť užitočný vo funkcionálnom programovaní a pri písaní dekorátorov. Tento modul zlepšuje použiteľnosť, modularitu a výkon funkcií, najmä v prípadoch, ktoré vyžadujú memorizáciu, čiastočné použitie funkcií alebo zachovanie metadát. Functools.wraps je dekorátor, ktorý pomáha pri písaní vlastných dekorátorov a zároveň zachováva názov pôvodnej funkcie, dokumentačný reťazec a ďalšie metadáta. Bez neho by obalenie funkcie mohlo viesť k strate cenných ladiacích a dokumentačných informácií. Stručne povedané, functools je ľahký, ale výkonný nástroj, ktorý má dôležitú úlohu v pokročilom programovaní v Pythone. Hoci nie je samostatným rozšírením ako Flask, umožňuje stručnejší, udržiavateľnejší a efektívnejší kód na úrovni funkcií v rámci väčších aplikácií v Pythone. [92]

Werkzeug je komplexná sada nástrojov WSGI pre Python, ktorá je navrhnutá tak, aby pomohla vývojárom vytvárať webové aplikácie poskytovaním nástrojov nižšej úrovne. Tvorí základ frameworku Flask, ale možno ju použiť aj ako samostatnú knižnicu na vytváranie webových služieb. Je potrebné si uvedomiť, že werkzeug nie je plnohodnotný webový framework, ale sada nástrojov, ktorá poskytuje všetky stavebné bloky potrebné na jeho vybudovanie. Hrá kľúčovú úlohu v ekosystéme webového vývoja v Pythone, najmä ako *back-endový engine* pre Flask. [93, 94]

Freeopcua je *open-source* knižnica jazyka Python, ktorá implementuje protokol OPC UA (opísaný v kapitole 2.3.3). Modul opcua umožňuje programom v jazyku Python fungovať ako klienti alebo servery. Ako klient sa vaša aplikácia v jazyku Python môže pripojiť

k existujúcemu serveru OPC UA a čítať alebo zapisovať premenné v reálnom čase. Ako server môže program v jazyku Python simulovať zariadenie a sprístupniť údaje iným klientom OPC UA. Vďaka tomu je knižnica vhodná na použitie v testovacích prostrediach, priemyselných digitálnych dvojčatách alebo riadiacich systémoch založených na simulácii. Jedna z kľúčových funkcií OPC UA je bezpečná komunikácia so šifrovaním a autentifikáciou. Opcua je mimoriadne užitočná v aplikáciách priemyslu a IIoT, kde je nevyhnutná integrácia výrobných zariadení s modernými softvérovými systémami. Prepojuje spracované údaje v Pythone alebo algoritmy umelej inteligencie so systémami priemyselnej automatizácie. Často sa používa vo výskumných projektoch, prototypovaní alebo výrobných prostrediach, kde je popri priemyselných komunikačných protokoloch potrebná flexibilita, schopnosť skriptovania a analýza údajov. [95, 96]

4.1.2 PLC – HMI komunikácia

Na komunikáciu medzi pythonom a pripojeným PLC sa využíva opcua knižnica. Pre jej fungovanie je potrebné vytvorenie spojenia pomocou adresy klienta OPC UA. Následne je možné posielanie a čítanie dát, ktoré je možné vidieť na výpise kódu číslo 1. Pre konkrétne čítanie dát je využívaná metóda GET a pre zápis POST.

```
client = Client("opc.tcp://192.168.0.1:4840")

try:
    client.connect()
    robot_connected = True
except Exception as e:
    logging.error(f"Failed to connect to the robot: {e}")
    robot_connected = False

def read_value_bool(node_id):
    if robot_connected:
        client_node = client.get_node(node_id)
        client_node_value = client_node.get_value()
        return client_node_value
    else:
        print("Robot is not connected.")
        return False
```

Výpis 1: Spojenie OPC UA, zapisovanie dát

4.2 HTML, CSS a JavaScript

HTML (*HyperText Markup Language*) je štandardný jazyk používaný na vytváranie a štruktúrovanie obsahu na webe. Poskytuje základný rámec pre webové stránky definovaním prvkov, ako sú text, obrázky, odkazy a multimédiá. HTML nie je programovací jazyk, ale skôr značkovací jazyk, čo znamená, že sa používa na „označovanie“ textu pomocou tagov, vďaka

ktorým webové prehliadače, vedľa ako majú správne zobraziť obsah. HTML sa často používa v kombinácii s CSS na ovládanie vzhľadu prvkov a s JavaScriptom na pridanie interaktivity. V priebehu času sa HTML vyvíjal a prechádzal rôznymi verziami [97]. HTML podporuje aj frameworky ako sú napríklad React, Angular alebo Vue, ktoré môžu mať rozšírenú funkcionálnosť, lepšie organizovanú štruktúru, rýchlejší vývoj a prototypovanie alebo optimalizáciu pre viaceré ekosystémy [98].

HTML používa tagy, ktoré sú uzavreté v lomených zátvorkách, pre popis prvkov na stránke. Väčšina prvkov má otvárací tag a uzatvárací tag, napríklad `<p>Hello</p>`, kde `<p>` začína odsek a `</p>` ho končí. Niektoré tagy sú samouzatváracie a nepotrebujú koncový tag. Základný dokument HTML začína deklaráciou `<!DOCTYPE html>`, za ktorou nasleduje element `<html>`. Vnútri tohto elementu sa zvyčajne nachádzajú dve hlavné sekcie:

- `<head>` – Metadáta o stránke, ako napríklad názov zobrazený na karte prehliadača, odkazy na štýly (CSS) a skripty (JavaScript).
- `<body>` – Viditeľná časť webovej stránky. Obsahuje obsah, ktorý je zobrazovaný používateľom, vrátane nadpisov, odsekov, obrázkov, zoznamov, tabuliek a ďalších prvkov. [99]

CSS je jazyk štýlov používaný na ovládanie vzhľadu a rozloženia prvkov HTML na webovej stránke. Zatiaľ čo HTML definuje štruktúru webovej stránky, CSS sa používa na úpravu štýlov týchto prvkov – zmenu ich farieb, veľkostí, písma, medzier, umiestnenia a ďalších parametrov. V prípade nepoužitia CSS by webová stránka vyzerala veľmi primitívne, zobrazená ako čierny text na bielom pozadí s minimálnym formátovaním. CSS umožňuje zmeniť základnú stránku na rozloženie podľa potrieb za použitia vizuálne prítiahľivých prvkov. CSS funguje na princípe aplikovania pravidiel na prvky HTML. Pravidlo sa skladá zo selektora a deklaračného bloku. Selektor definuje, ktorá časť HTML prvkov sa bude meniť a deklaračný blok definuje zmenu daných prvkov [100]. Hlavné prvky CSS sú nasledovné:

- Úprava štýlu textu – zmena fontu, veľkosti, farby medzier či zarovnanie
- Zmena rozloženia – umiestnenie prvkov, úprava okrajov či odsadenia
- Pozadia a orámovania – obrázky, prechody, rohy, tieň
- Animácie a prechody – plynulé efekty
- Responzívny dizajn – prispôbenie webových stránok rôznym veľkostiam obrazovky

Tým, že CSS oddeľuje obsah od štýlovania, uľahčuje údržbu a aktualizáciu webových stránok. Podporuje opätovné použitie, pretože jeden súbor CSS dokáže štýlovať mnoho stránok a taktiež aj frameworky, ako napríklad Bootstrap a Tailwind CSS, ktoré pomáhajú vývojárom vytvárať atraktívne rozloženia rýchlejšie [101].

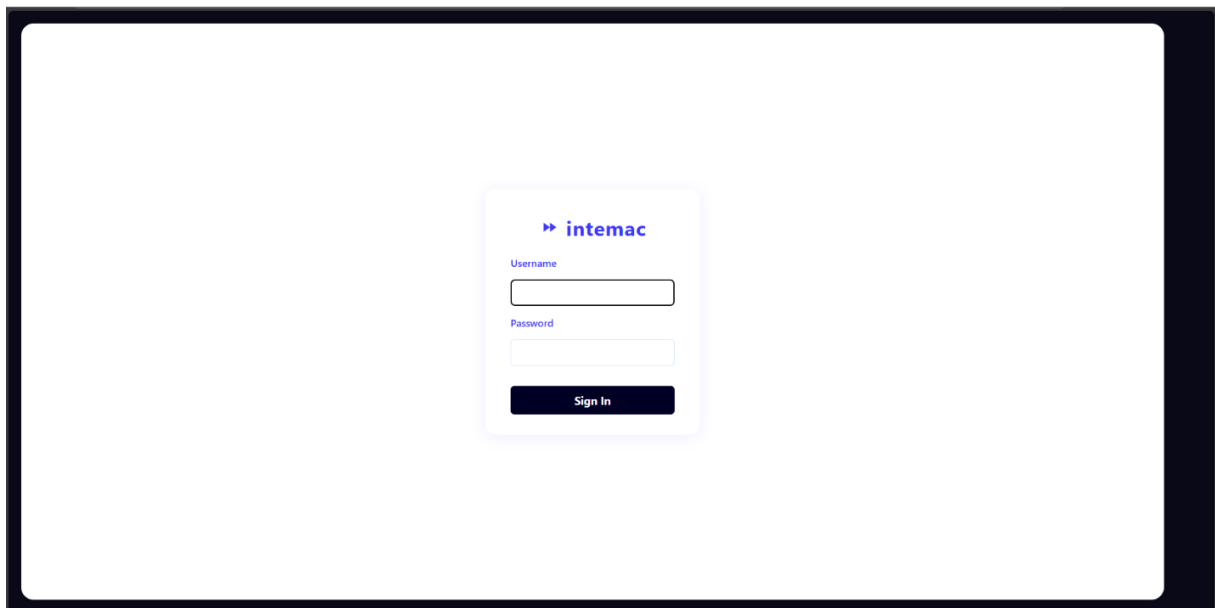
JavaScript je programovací jazyk používaný na vytvorenie interaktívnych a dynamických webových stránok. Zatiaľ čo HTML vytvára štruktúru webovej stránky a CSS ju upravuje, JavaScript pridáva konanie. Jedná sa o procesy, vzhľadom na ktoré sa na stránke menia informácie bez nutnosti jej opätovného načítania [102]. Príkladom môže byť, zobrazenie správy po stlačení tlačidla. JavaScript zaisťuje:

- Reakcie na akcie používateľa – kliknutia, písanie, rolovanie
- Zobrazovanie alebo skrývanie prvkov na stránke

- Overovanie vstupu do formulára – správny formát e-mailu
- Vytváranie animácií, či interaktívnych máp
- Načítanie a zobrazenie údajov zo servera
- Ukladanie údajov do prehliadača
- Vytváranie plnohodnotných aplikácií v kombinácii s frameworkami [103]

4.3 Login stránka

Prihlasovacia stránka je jediná html stránka, ktorá je prístupná bez prihlásenia. Aktuálne existujú dva druhy účtov, pričom každý je reprezentovaný jedným konkrétnym účtom – Admin a Spectator. Obidva účty sú zapísané v hlavnom programe app.py, pričom ich heslá sú *zahashované* pomocou funkcie `generate_password_hash`, ktorá patrí pod sadu nástrojov `werkzeug`. V prípade potreby by bolo možné vytvoriť aj ďalšie účty. Pri prihlasovaní sa overuje heslo pomocou `check_password_hash` a po úspešnom overení sa užívateľ dostane na domovskú stránku. V prípade prihlásenia pomocou účtu Admin je používateľovi sprístupnená celá webová stránka s plnou funkcionalitou. Naopak účet Spectator má limitované práva, sú mu zobrazené len oznamovacie informácie a nedokáže nič ovládať. Vizualne zobrazenie login stránky je možné vidieť na obrázku 14.



Obr. 14: Login Stránka

4.4 Dekorátory

Prístupy k jednotlivým html stránkam, a taktiež funkciám webstránky sú definované dekorátormi, ktoré zapadajú pod `functools` a ich definície sú zobrazené na výpise kódu č. 6. Ako už bolo povedané v kapitole 4.3, jediná stránka prístupná bez prihlásenia je Login. V prípade snahy o zobrazenie inej html stránky manuálnym napísaním do vyhľadávacieho

panelu je užívateľ automaticky presmerovaný na login. Podobne, ako pre jednotlivé stránky, tak vieme dekorátory používať aj pre iné funkcie, ako sú zápis a čítanie informácií z PLC, ktoré je možné vidieť na výpise kódu 2, a taktiež zobrazovanie jednotlivých tlačidiel, ktoré sú na výpise kódu číslo 3.

```
def login_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if 'username' not in session:
            return redirect(url_for('login'))
        return f(*args, **kwargs)
    return decorated_function

def admin_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if 'role' not in session or session['role'] != 'admin':
            flash('Admin access required.', 'danger')
            return jsonify({'error': 'Admin access required.'}), 403
        return f(*args, **kwargs)
    return decorated_function
```

Výpis 2: Definovanie dekorátorov

```
@app.route("/enable", methods=["POST"])
@login_required
@admin_required
def enable():
    data = request.json
    value = data.get("value", True)
    write_value_bool('ns=3;s="FB_DB"."Enable"', value)
    return jsonify({"status": "success"})
```

Výpis 3: Prístup k funkcii pre užívateľa a admin

4.5 Domovská stránka

Po prihlásení je užívateľ presmerovaný na domovskú stránku, ktorej obsah závisí od jeho práv, ktoré má pridelené podľa typu účtu. V prípade prihlásenia účtom s plnou funkcionalitou domovská stránka zobrazuje číslo stavu v stavovom automate, číslo zapísanej funkcie, nastavenú rýchlosť robota v percentách a taktiež stav pripojenia robota. Okrem zobrazenia stavov sa tu nachádzajú aj polia pre zadanie ID funkcie, rýchlosti v percentách a ich následné potvrdenie. Na stránke sú aj tlačidlá, ako Execute pre spustenie funkcie a tlačidlá Start, Pause, Stop, Enable a Disable pre celkové ovládanie funkčného bloku a programu robota. Tlačidlo Info zobrazuje stručný návod, ako ovládať robota pomocou webstránky. Pre účet s obmedzenou funkcionalitou sú zobrazené len informačné polia, zatiaľ čo ovládanie robota a funkčného bloku

nie je sprístupnené. Domovská stránka pre účet typu Admin je zobrazená na obrázku č. 15 a pre účet typu Spectator na obrázku č. 16.



Obr. 15: Stránka home po prihlásení účtom Admin



Obr. 16: Stránka home po prihlásení účtom Spectator

4.6 Sidebar

Jedná sa o postranný panel webovej stránky, ktorý má dve zobrazenia. Prvé zobrazenie je klasické pri bežnom prezeraní stránky a jeho hlavným obsahom sú ikony, ktoré slúžia ako tlačidlá pre navigáciu medzi jednotlivými html stránkami. V poradí zhora tlačidlá reprezentujú stránku home, robot, kamera, error a odhlásenie. V prípade odhlásenia sa používateľ dostane späť na prihlasovaciu stránku. Okrem týchto tlačidiel je tu aj logo firmy Intemac, ktoré slúži, ako tlačidlo pre zobrazenie rozšírenia sidebaru. Toto zobrazenie sidebaru je v pravej strane

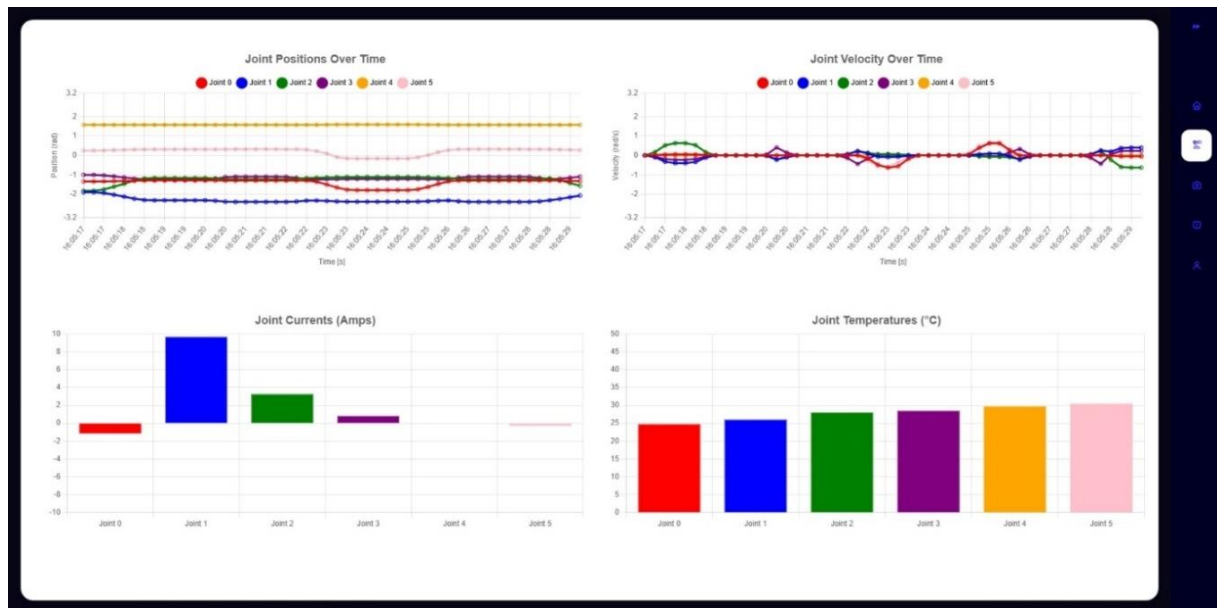
obrázku č. 16. V prípade rozšíreného zobrazenia sidebar okrem navigácie medzi stránkami zobrazuje aj aktuálny prihlásený účet, popis použitého hardvéru pre celé ovládanie robota a stav Erroru. Rozšírené zobrazenie sidebaru je na obrázku č. 17.



Obr. 17: Stránka home s otvoreným sidebarom

4.7 Robot stránka

Html stránka robot má len informačný charakter a preto zobrazuje rovnaký obsah pre obidva druhy účtov. Nachádzajú sa tu 4 grafy ktorých informácie sa automaticky obnovujú každých dvestopäťdesiat milisekúnd. Toto rozloženie je zobrazené na obrázku č. 18. Hore vľavo sa nachádza graf reprezentujúci pozície jednotlivých kĺbov v radiánoch. Po jeho pravej strane je graf predstavujúci rýchlosti jednotlivých kĺbov v radiánoch za sekundu. Aktuálne informácie v grafoch sa ukladajú na pravú stranu a grafy sa dynamicky posúvajú do ľava. Aby nedošlo k preplneniu grafov a následnej nečitateľnosti dát, grafy zobrazujú len najnovších 50 hodnôt. Staršie hodnoty sa automaticky vymažú z pamäte. V dolnej polovici sa nachádzajú dva grafy reprezentujúce aktuálnu hodnotu prúdu a teploty jednotlivých kĺboch. Graf interpretujúci prúdy kĺbov môže naberať aj záporné hodnoty v závislosti smeru natočenia kĺbov, preto je jeho nulová hodnota posunutá do stredu a nezhoduje sa s osou x.



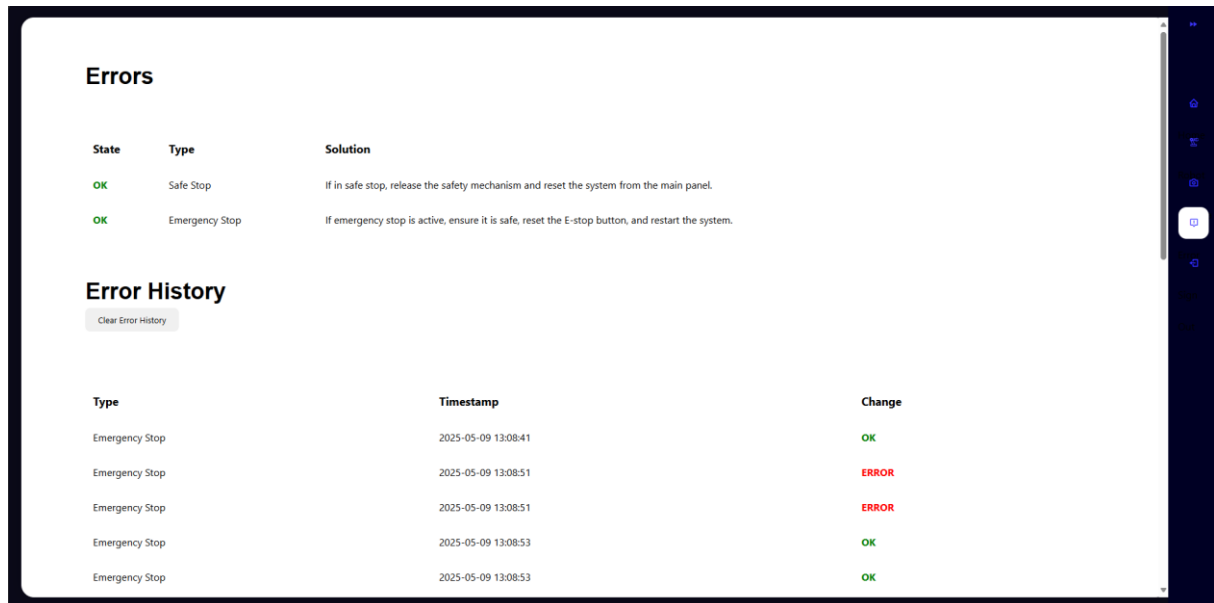
Obr. 18: Stránka robot

4.8 Kamera stránka

Stránka kamera slúži na zobrazenie fotografií a ich informácií, ktoré sú vyhotovené pre lokalizáciu objektov. Lokalizácia je využívaná pre automatickú kalibráciu robotických pracovísk a bola predmetom diplomovej práce kolegu Petra Lisníka. Opis tejto aplikácie a testovania je v kapitole 5.2.3. Na html stránke kamera sa zobrazujú už spomínané fotografie, ktoré sú načítané z lokálneho priečinku. Tento proces je periodicky načítaný každú sekundu pre zobrazenie aktuálnej fotografie. Informácie, ktoré sú vyčítané zo súboru, sú názov a čas vytvorenia fotografie.

4.9 Error stránka

Stránka error je rozdelená na dve hlavné časti. Horná časť poskytuje informácie o stavoch errorov v reálnom čase. Robot od firmy Universal Robots primárne nadobúda dva druhy errorov. Prvý error je Safe stop ktorý nastáva v prípade kolízie robota. Jedná sa o implmentáciu, základnej ochrannej funkcie kolaboratívnych robotov. Error Emergency stop oznamuje aktiváciu *emergency stop* tlačidla na *teach pendant*. V obidvoch prípadoch stránka zobrazuje stav erroru – OK alebo ERROR, a taktiež postup ako odstrániť error. Pre odstránenie Emergency stop je potrebné odistiť tlačidlo na TP, spustiť robota a následne resetovať PLC automat. Pre Safestop je postup jednoduchší, stačí povoliť pohyb robota cez TP a resetovať FB v PLC. Spodná časť stránky slúži ako logger, teda zaznamenáva zmeny jednotlivých errorov. V prípade, že sa stav jedného z errorov zmení z OK na ERROR alebo naopak, zmena sa zapíše. Okrem druhu zmeny sa zapíše aj druh erroru, a taktiež čas, kedy chyba vznikla. V prípade, že je prihlásený profil admin, je tu aj funkcionálna vymazania záznamov o erroroch v tvare tlačidla Clear Error History. Stránku z pohľadu admina je možné vidieť na obrázku číslo 19.



The screenshot displays a web interface for error management. It is divided into two main sections: 'Errors' and 'Error History'.

Errors Section:

State	Type	Solution
OK	Safe Stop	If in safe stop, release the safety mechanism and reset the system from the main panel.
OK	Emergency Stop	If emergency stop is active, ensure it is safe, reset the E-stop button, and restart the system.

Error History Section:

Clear Error History

Type	Timestamp	Change
Emergency Stop	2025-05-09 13:08:41	OK
Emergency Stop	2025-05-09 13:08:51	ERROR
Emergency Stop	2025-05-09 13:08:51	ERROR
Emergency Stop	2025-05-09 13:08:53	OK
Emergency Stop	2025-05-09 13:08:53	OK

Obr. 19: Stránka error

5 EXPERIMENTÁLNE OVEROVANIE

5.1 Testovanie funkčného bloku v simulácii

Simulácia zohráva podstatnú úlohu vo vývoji a validácii automatizačných systémov, pretože ponúka bezpečné a finančne efektívne prostredie na testovanie logiky, výmeny signálov a správania systému bez potreby fyzického hardvéru. Jednou z významných výhod simulácie je schopnosť odhaliť a vyriešiť problémy už v počiatočnej fáze vývojového procesu vďaka možnosti opakovania simulácie dookola. Tým sa znižujú možné problémy počas fyzickej aplikácie, a taktiež sa zvyšuje bezpečnosť [104]. Simulácia však prináša aj určité obmedzenia, ako je potreba presnej konfigurácie virtuálnych rozhraní, potenciálne problémy s kompatibilitou medzi rôznymi softvérovými nástrojmi a ich celková komplexnosť [105].

Počiatočná fáza testovania programu sa uskutočnila pomocou simulačnej platformy SIMIT (SIMIT SP) od spoločnosti Siemens. Cieľom bolo simulovať Funkčný blok, robot UR a ich komunikáciu vo virtuálnom prostredí pred nasadením na fyzický hardvér. Napriek viacerým pokusom o správnu konfiguráciu a mapovanie adres signálov, medzi komponentmi pretrvávali problémy s adresami. Program z aplikácie TIA Portal sa nedokázal nahráť správne, adresy nesesedeli a tým pádom simulácia nebola možná. V dôsledku toho sa rozhodlo o simuláciu a testovanie v prostredí Unity, kde bolo možné program testovať efektívnejšie a s väčšou flexibilitou. OPC UA komunikácia bola definovaná ako klient v Unity3D a v robotovi server.

5.1.1 Unity

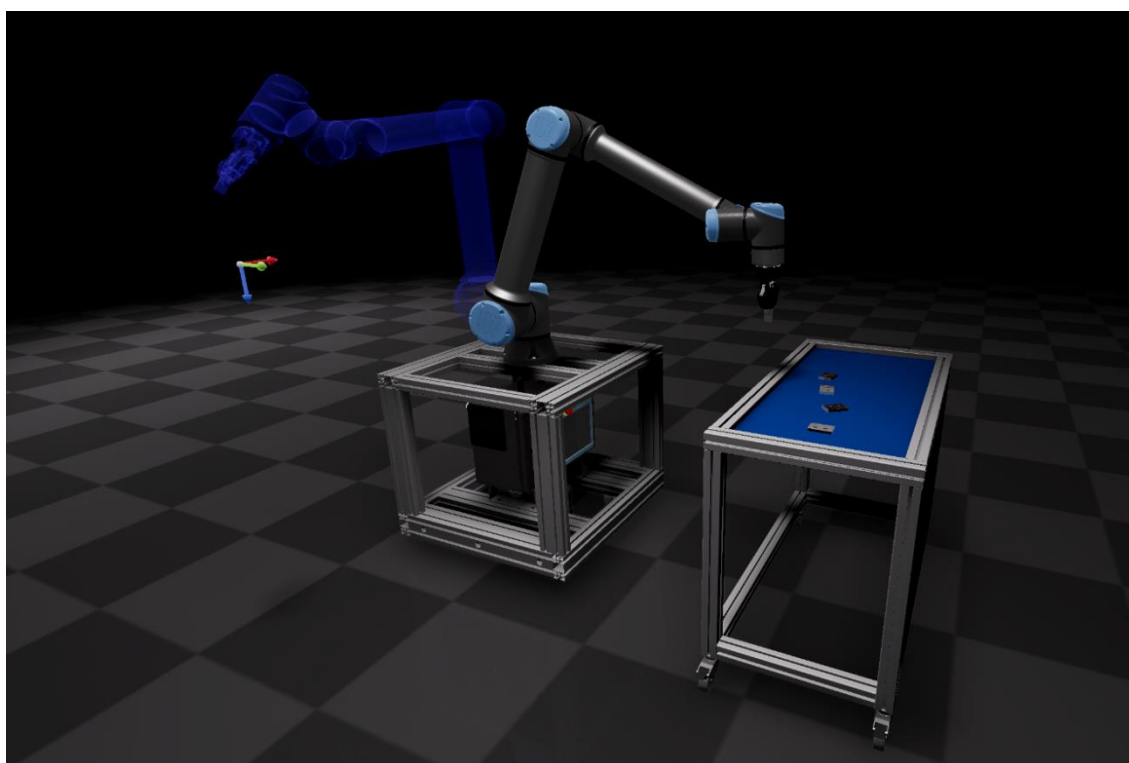
Unity je vývojová platforma primárne používaná na vytváranie 3D a 2D aplikácií, najčastejšie v oblasti vývoja hier. Vďaka svojej flexibilitě, vizuálnym možnostiam, podpore vlastného skriptovania a implementovaným OPC UA klientom je populárna aj v oblastiach, ako sú simulácie, robotika a digitálne dvojčatá [106]. Hoci jej chýba natívna podpora pre priemyselné protokoly, je možné jej integrácie pomocou knižníc tretích strán alebo vlastných komunikačných rozhraní. Poskytuje kvalitnejšiu interpretáciu fyziky vďaka enginu NVIDIA PhysX, má lepšie vizuálne efekty a umožňuje jednoduchšie prispôsobenie pri zmene typu robota alebo PLC. V kontexte automatizácie ponúka efektívny spôsob vizualizácie procesov a testovania riadiacej logiky v dynamickom virtuálnom prostredí.

5.1.2 Testovanie v Unity3D

Navrhnuté riešenie bolo otestované na sofistikovanej simulácii v nástroji Unity3D vyvinutej vo výskumnom centre INTEMAC, ktorá obsahuje funkcie pre prácu s fyzikálnymi objektmi, výpočty doprednej a inverznej kinematiky, riadenie dynamických pohybov a tvorbu programov ako pre kĺbovú, tak lineárnu sekvenciu pohybov robota, či v neposlednej rade ovládanie inteligentného koncového efektora. Vďaka čomu je možné celý návrh programu vopred pripraviť vo virtuálnom prostredí, otestovať a odstrániť neefektívne, alebo kolízne stavy, a do fyzického sveta ho preklopiť s uspokojivým výsledkom.

V rámci diplomovej práce bol v prvej fáze vytvorený prototyp, ktorý bol prezentovaný na udalosti Inspiration Day, ktorú usporiadal INTEMAC. V druhej fáze bol program rozšírený o pokročilé funkcionality demoštrujúce alternatívne riešenie k reálnemu prepojeniu Siemens PLC a kolaboratívneho robota UR10e. Navrhnuté simulačné riešenie využívalo OPC UA obojsmernú komunikáciu s PLC, ktorá umožňovala čítanie a zápis v reálnom čase. Riešenie do veľkej miery interpretovalo vlastnosti robota vo virtuálnom priestore. Vzhľadom k tejto skutočnosti bolo možné vierohodne otestovať program navrhnutý do reálneho sveta.

Táto architektúra je navrhnutá s ohľadom na jednoduchú modularizáciu, čo umožňuje rozšírenie alebo úpravu bez nutnosti výrazných zmien v už existujúcej štruktúre. Vďaka tomu je riešenie škálovateľné a prispôsobiteľné budúcim potrebám v závislosti od konkrétnej aplikácie a systém je vysoko všestranný pre priemyselné aj výskumné prostredie.



Obr. 20: Simulované stanoviisko v Unity3D

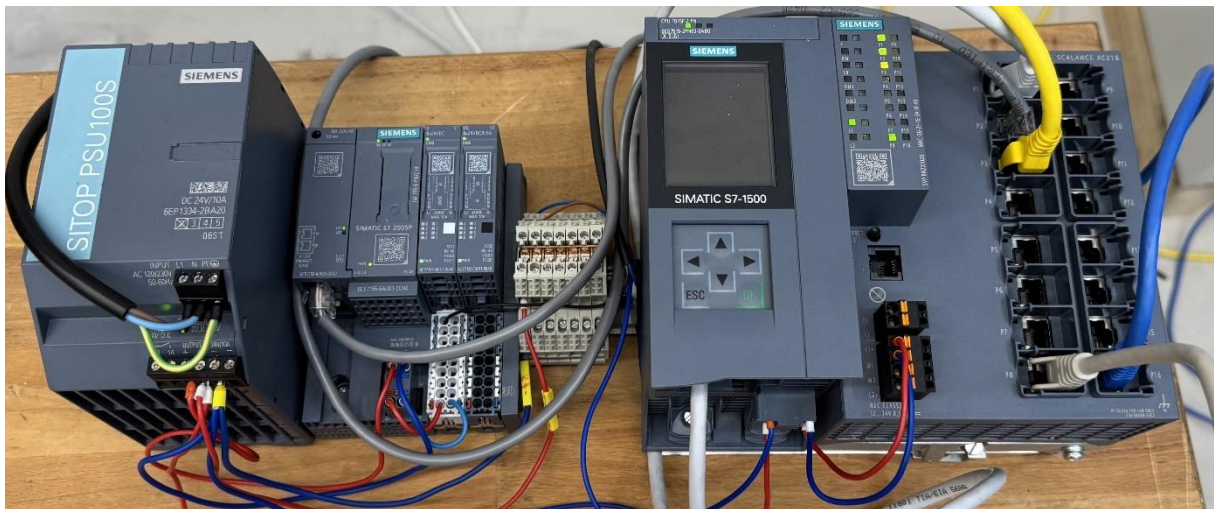
5.2 Testovanie funkčného bloku na reálnom hardvéri

5.2.1 Použitý hardvér

Pre testovanie a vývoj funkčného bloku bolo použité PLC od značky Siemens s označením SIMATIC S7-1500. Konkrétne sa jednalo o model CPU 1515F-2 PN s číslom 6ES7515-2FN03-0AB0. Toto PLC disponuje 1,5 MB pamäte pre program, 4,5 MB dátovej pamäte a rýchlosťou spracovania 6 nanosekúnd na bit. Hoci tieto parametre nie sú zaujímavé vo sfére súčasných počítačov a telefónov, medzi PLC zariadeniami je to nadpriemer. Rýchlosť tohto procesoru zaisťuje rýchle vykonávanie úloh, vďaka čomu je vhodný aj pre časovo kritické aplikácie. PLC potrebuje pamäťovú kartu SIMATIC, na ktorú sa ukladá program. Podporuje

duálne rozhranie PROFINET (RT a IRT) čo uľahčuje výmenu údajov v reálnom čase [107]. PLC je doplnené o switch Siemens SCALANCE XC216 s označením 6GK5216-0BA00-2AC2, ktorý disponuje šiestimi Ethernet portami [108]. Okrem samotného PLC je sem pripojený aj distribuovaný I/O systém Siemens SIMATIC ET 200SP s označením 6ES7155-6AU01-0CN0, ktorý je doplnený o vstupnú kartu Siemens DI 8x24VDC ST_1 s označením 6ES7 131-6BF01-0BA0 [109]. Táto karta je schopná prijímať vstupy na ôsmich adresách nula až sedem, ktoré pracujú na 24 voltoch. Je možné sem zapojiť rôzne senzory, tlačidlá alebo relé. Posledná súčiastka je výstupná karta Siemens DQ 8x24VDC/0.5A ST_1 s ozn 6ES7132-6BF01-0BA0, ktorá môže byť používaná rovnakým spôsobom ako vstupová. Sústavu súčiastok je možné vidieť na obrázku číslo 21.

Použitý robot bol model UR10e od firmy Universal robots je stredne veľký priemyselný kolaboratívny robot, ktorý disponuje dosahom 1300 mm, maximálnou nosnosťou 12,5 kg a hmotnosťou 33,5 kg. Má mnoho možných aplikácií, ako napríklad paletizácia, zváranie, montáž, dávkovanie, brúsenie či manipuláciu s materiálom [110]. Ako koncový efektor bol zvolený gripper of firmy ROBOTIQ s označením 2F-85 s váhou jedného kilogramu a maximálnou nosnosťou päť kilogramov. Disponuje maximálnym rozvretím 85 mm a silou 235 N [111]. Pre jeden z testovacích programov bol využitý optoelektronický svetelný snímač BALLUFF BOS015U s dosahom 40 až 200 mm [112]. Robota a snímač je možné vidieť na obrázku číslo 23.

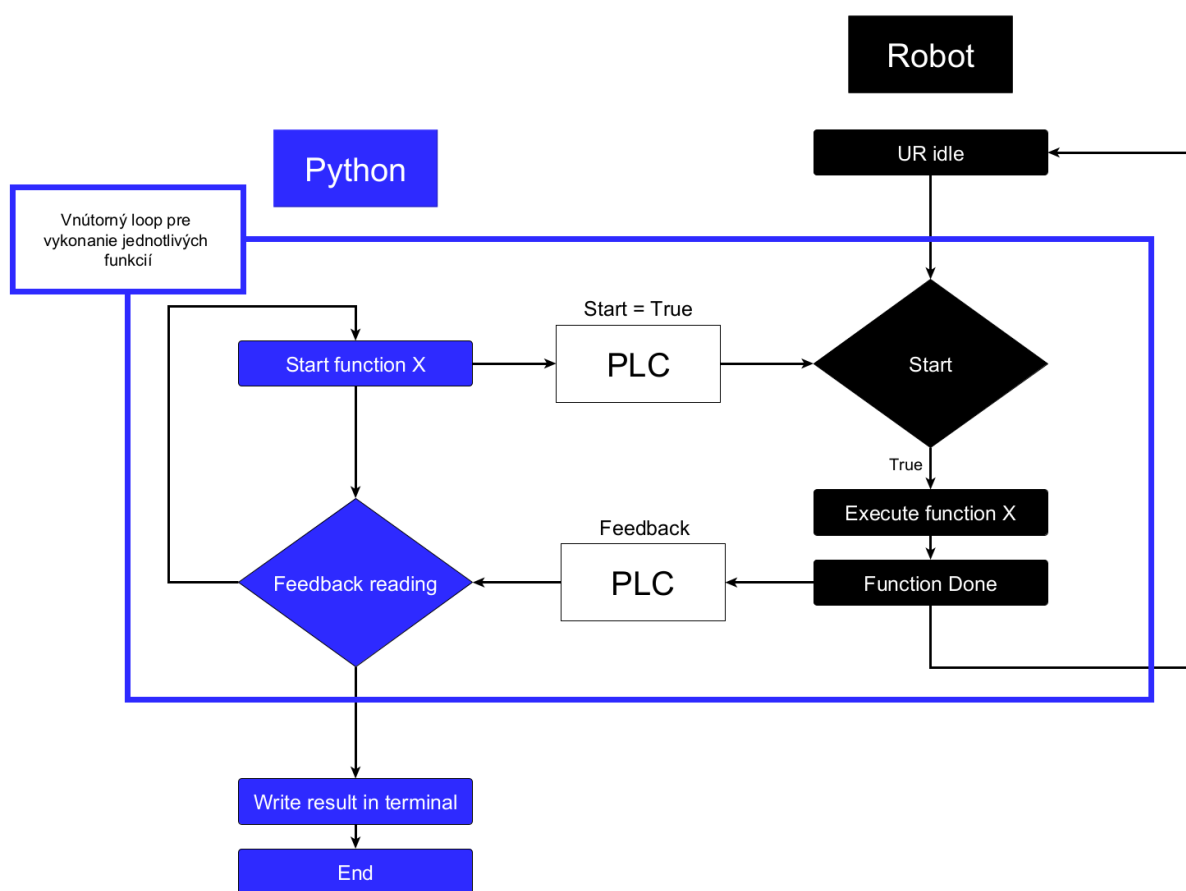


Obr. 21: Siemens PLC SIMATIC S7-1500, zdroj, karty a switch

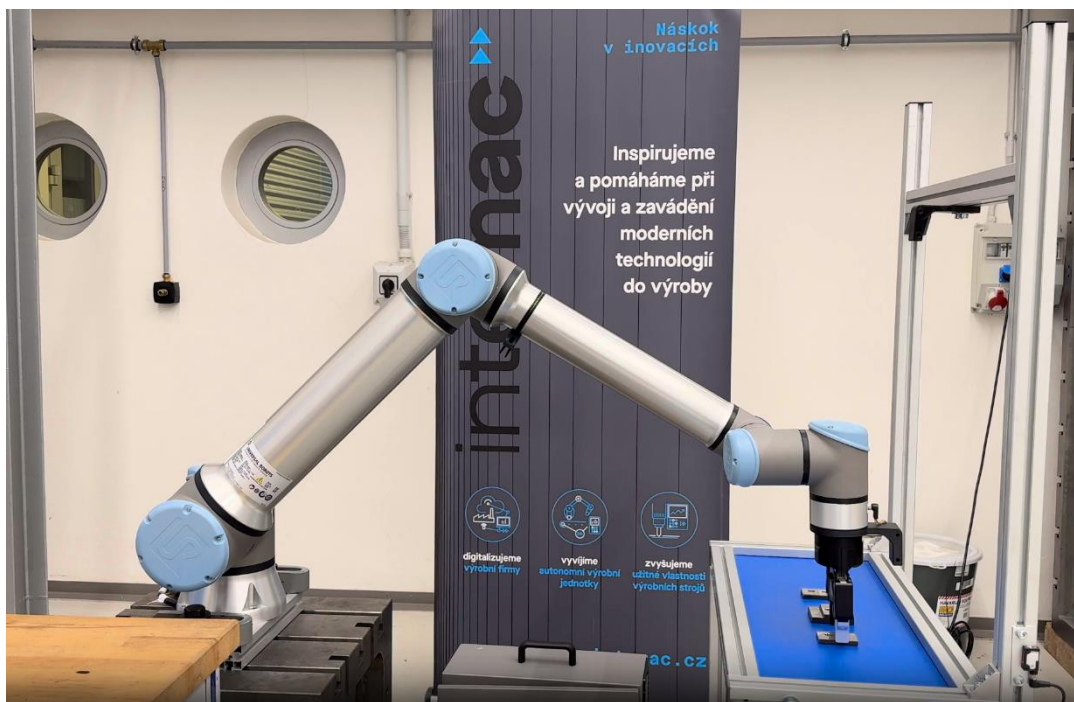
5.2.2 Prvý testovací program – overenie objektov

Jedná sa o program napísaný v jazyku Python, ktorý overuje prítomnosť a šírku objektu v gripperi. Základ programu je stavový automat, ktorý riadi kolaboratívneho robota UR pomocou knižnice opcua. Po spustení sa skript pripojí k serveru OPC UA robota a v termináli sa zobrazí jednoduché menu, kde si používateľ môže vybrať spustenie procesu pre objekt 1, 2 alebo 3. Na základe výberu program prejde do zodpovedajúcej postupnosti stavov. Každý z procesov prechádza cez viacero stavov, kde odosiela špecifické hodnoty FunctionID do PLC robota prostredníctvom adres GlobalDB.Execute a GlobalDB.FunctionID. Čaká na potvrdenie vykonania monitorovaním hodnoty FBMain_DB.Main a po vykonaní funkcie sa prejde na

d'alší krok. V programe je zakomponované aj náhodné pustenía objektu, ktoré je ovládané pomocou príkazu `random.choice([True, False])`. Po vykonaní rozhodnutia robot zastane na pozícií pred senzorom, kde sa overuje prítomnosť a číta aktuálna šírka zovretia gripperu. V prípade prítomnosti objektu pred senzorom systém pokračuje v postupnosti, a v opačnom ju ukončí. Posledným krokom stavového automatu je výpis informácie stavu o prítomnosti objektu a jeho šírke. Po dokončení všetkých úloh sa vráti do východzieho stavu, čo umožňuje d'alší výber alebo ukončenie. Vývojový diagram pre ovládanie UR robota s využitím Pythonu je na obrázku 22 a platí pre všetky tri testovacie aplikácie. Rozmiestnenie reálneho stanoviška je možné vidieť na obrázku 23.



Obr. 22: Zjednodušený vývojový diagram ovládania UR robota cez Python



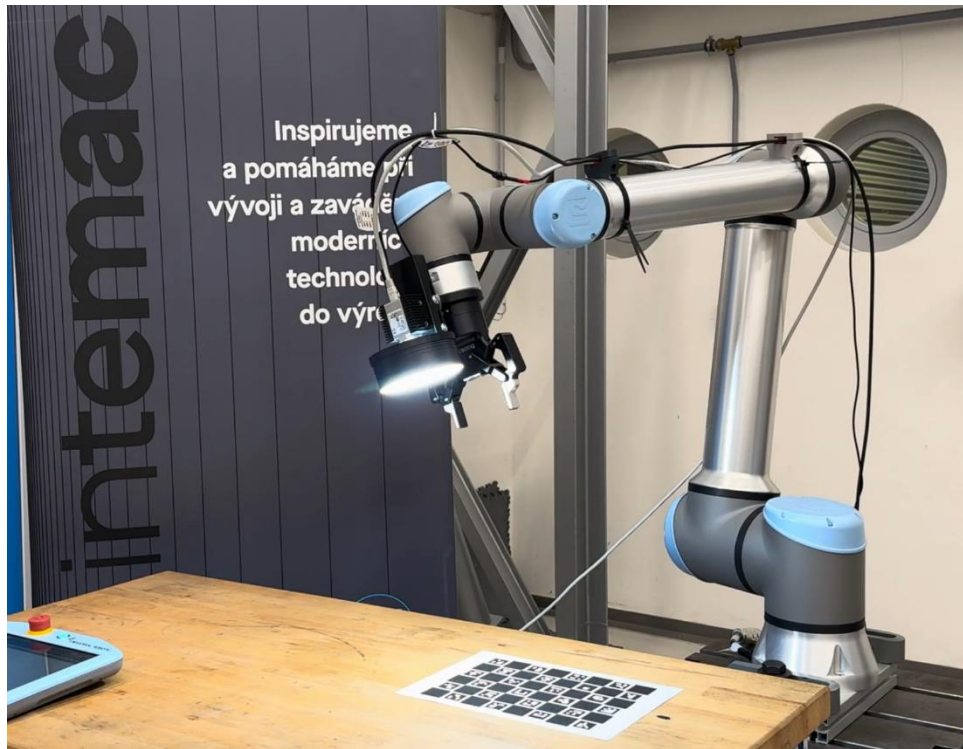
Obr. 23: Robot pri vykonávaní prvého testovacieho programu

5.2.3 Druhý testovací program – vytvorenie datasetu

Testovací program číslo dva overuje funkčný blok na vytvorení datasetu pre možné následné spracovanie Petrom Lisníkom v jeho diplomovej práci Využitie metód strojového videnia k automatickej kalibrácii robotických pracovísk. Pre vytvorenie úspešného datasetu bolo potrebné vykonať nasledovné kroky:

- Zapnutie kruhového svetla pomocou ovládania digitálnych outputov robota.
- Nastavenie robota na freeride pre manuálne navedenie robota pred kalibračnú dosku.
- Deaktivovanie freeride módu.
- Vyčítanie aktuálnej polohy všetkých kĺbov robota a taktiež jeho TCP polohy.
- Polohovanie robota pomocou moveL s využitím posielania dát cez UserDataInput.
- Vytvorenie fotografií v každej z pozícií.
- Vypnutie svetla

Po ukončení tohto procesu je dataset úspešne vytvorený v lokálnej zložke a pripravený pre výpočet. Robota pri vytváraní datasetu je zobrazený na obrázku číslo 24.



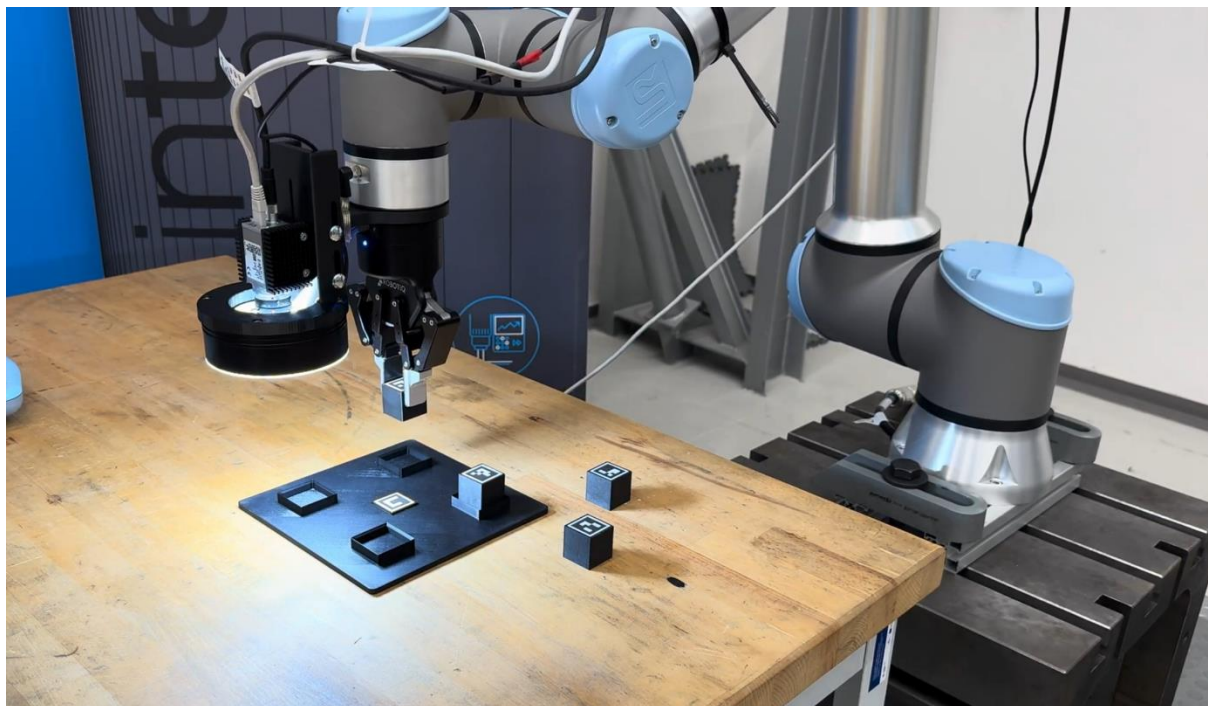
Obr. 24: Robot UR10e pri vytváraní datasetu v druhom testovacom programe

5.2.4 Tretí testovací program – *pick and place*

V treťom testovacom programe bol cieľ lokalizovať kocku a následne ju premiestniť na iné lokalizované miesto. Kocka aj finálne miesto mali na sebe kalibračné značky a ich poloha bola vypočítaná programom od Petra Lisníka z jeho diplomovej práce „Využitie metód strojového videnia k automatickej kalibrácii robotických pracovísk“. Postup pick and place aplikácie bol nasledovný:

- Zapnutie kruhového svetla pomocou ovládania digitálnych outputov robota.
- Nastavenie robota na freeride pre manuálne navedenie robota pred kalibračnú dosku.
- Deaktivovanie freeride módu.
- Vytvorenie fotky pre výpočet lokalizácie kocky a cieľového miesta.
- Vyčítanie aktuálnej polohy všetkých kĺbov robota a taktiež jeho TCP polohy.
- Pohyb robota nad polohu kocky s využitím posielania dát cez UserDataInput.
- Pohyb robota na miesto kocky s využitím posielania dát cez UserDataInput.
- Zatvorenie gripperu pre uchytanie kocky.
- Pohyb robota na lokalizované cieľové miesto.
- Otvorenie gripperu pre uvoľnenie kocky.
- Vypnutie svetla

Robot vykonal funkcie vo všetkých testovacích programoch podľa očakávania a tým sa úspešne overila správnosť funkčného bloku. Robot pri vykonávaní pick and place aplikácie je zobrazený na obrázku 25.



Obr. 25: Robot UR10e pri ukladaní náhodne rozmiestnených kociek do platformy

6 ZÁVER

V rámci tejto diplomovej práce bol navrhnutý a implementovaný funkčný blok v programovacom prostredí Siemens TIA Portal, ktorého cieľom bolo umožniť pokročilé a bezpečné riadenie kolaboratívneho robota Universal Robots UR10e prostredníctvom štandardizovaného komunikačného protokolu OPC UA. Riešenie bolo navrhnuté ako univerzálne a modulárne, so dôrazom na nezávislosť od konkrétneho výrobcu a vysokú mieru prispôbitel'nosti. Vytvorením stavového automatu s podporou diagnostiky, bezpečnostného zastavenia a možnosti parametrizácie sa dosiahla vysoká miera škálovateľnosti, transparentnosti a spoľahlivosti systému.

Navrhnutý funkčný blok je postavený na logickom stavovom automate, ktorý umožňuje jednoduché a efektívne ovládanie robota. Vstupné signály ako Enable, Start, Stop, Pause, Execute a FunctionID poskytujú operátorovi rozsiahlu kontrolu nad správaním robota. Na strane výstupov sú poskytované informácie o aktivite, výskyte chýb, potvrdeniach o stave a výsledkoch vykonaných úloh. Riešenie zároveň umožňuje prácu s užívateľskými dátami, čo zvyšuje flexibilitu pri jeho nasadení v rôznych priemyselných aplikáciách. Funkčný blok je navrhnutý tak, aby umožňoval bezpečné reakcie v prípade chybových alebo rizikových stavov, vrátane reakcie na núdzové zastavenie alebo kolíziu. Dôležitou súčasťou riešenia bola aj tvorba moderného webového HMI rozhrania. Pomocou programovacieho jazyka Python a *frameworku* Flask, v kombinácii s webovými technológiami HTML, CSS a JavaScript, bol navrhnutý a implementovaný systém pre vzdialené ovládanie robota, správu prístupov a vizualizáciu prevádzkových údajov. Rozhranie je responzívne a prístupné z rôznych zariadení, pričom rozlišuje používateľské roly admin a spectator, čím sa zvyšuje bezpečnosť interakcie. Táto časť riešenia výrazne prispieva k požiadavkám moderného priemyslu na mobilitu, vzdialenú správu a jednoduchosť ovládania.

Výsledné riešenie bolo prínosom aj pre výskumnú činnosť centra INTEMAC, kde bolo prakticky overené na simulačnej platforme v Unity3D s podporou kinematickej analýzy, detekcie kolízií a dynamického testovania pohybov. Práca priamo prispela k podpore trendu inteligentnej konektivity a položila pevné základy pre budúci výskum a rozšírenie konceptu. Vzhľadom na nezávislosť riešenia od licencovaných nástrojov výrobcov robotov je umožnené jeho rozšírenie aj na ďalšie zariadenia. Využitie otvorených štandardov, ako je OPC UA, zabezpečuje schopnosť spolupracovať s nadriadenými systémami a ľahšiu integráciu do výrobných liniek.

Počas vývoja boli identifikované aj určité technické limity na strane dostupných nástrojov. V prípade prostredia Siemens SIMIT bola zaznamenaná nekompatibilita pri načítavaní štruktúrovaných adries cez OPC UA, čo bránilo plnohodnotnej simulácii. Táto forma komunikácie nebola v čase realizácie plne odladená pre použitie s UR robotmi. Alternatívne riešenia, ako napríklad SIMATIC Robot Library, si vyžadovali platené licencie a špecifickú verziu TIA Portal, čo obmedzovalo dostupnosť. Na základe týchto skutočností bolo rozhodnuté o vytvorení vlastného riešenia, ktoré bude nezávislé, otvorené a znovu použiteľné, čím sa zároveň podporil interný výskum v oblasti nízkonákladových a otvorených systémov automatizácie.

Významným aspektem je aj plánovaná zverejniteľnosť riešenia. Funkčný blok, webové rozhranie aj súvisiaca dokumentácia budú poskytnuté pod *open-source* licenciou, čím bude umožnené nezávislé overenie, prispôsobenie pre iné typy robotov a ďalší vývoj v rámci akademickej aj priemyselnej sféry. Príkladmi rozšírenia sú pridanie podpory pre viac robotov naraz alebo rozšírenie o bezpečnostné vrstvy typu Safety PLC.

Z dlhodobého hľadiska môže byť navrhnuté riešenie využité v prostrediach, ktoré si vyžadujú otvorenú a flexibilnú integráciu robotických pracovísk bez viazanosti na konkrétne značky. Systém umožňuje rozšírenie o cloudové služby, vzdialené monitorovanie či adaptívne plánovanie výroby. Vďaka svojej modulárnej štruktúre je pripravený na integráciu do rôznych priemyselných aplikácií, pričom zostáva jednoducho nastaviteľný, rozšíriteľný a prístupný pre širšiu komunitu. Na základe získaných výsledkov možno konštatovať, že navrhnutý koncept predstavuje nový smer v oblasti riadenia kolaboratívnych robotov pomocou PLC, pričom spája výhody otvoreného vývoja, štandardizovanej komunikácie a používateľskej jednoduchosti. Riešenie má potenciál stať sa základom pre budúci vývoj v oblasti dostupnej a transparentnej automatizácie výrobných procesov.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] *Cobots* [online]. [vid. 2025-05-16]. Dostupné z: <https://www.universal-robots.com/>
- [2] NOF, Shimon Y. *Handbook of Industrial Robotics*. 1999. 2nd ed. ISBN 0-471-17783-0.
- [3] KRAGIC, Danica, Joakim GUSTAFSON, Hakan KARAOGUZ, Patric JENSFELT a Robert KRUG. *Interactive, Collaborative Robots: Challenges and Opportunities*. 2018.
- [4] ØSTERGAARD, Esben H. *The role of cobots in Industry 4.0*. 2017.
- [5] SHERWANI, F., Muhammad Mujtaba ASAD a B.S.K.K. IBRAHIM. *Collaborative Robots and Industrial Revolution 4.0 (IR 4.0)*. In: *2020 International Conference on Emerging Trends in Smart Technologies (ICETST): 2020 International Conference on Emerging Trends in Smart Technologies (ICETST)* [online]. 2020, s. 1–5 [vid. 2025-04-30]. Dostupné z: doi:10.1109/ICETST49965.2020.9080724
- [6] *Cobot Market to account for 30% of Total Robot Market by 2027 – Interact Analysis*. B.m.: Interact Analysis, UK. 5. prosinec 2019
- [7] What Are Collaborative Robots, Cobots | A3 Robotics Collaborative Robots. *Automate* [online]. [vid. 2025-05-05]. Dostupné z: <https://www.automate.org/robotics/cobots/what-are-collaborative-robots>
- [8] MÜLLER, Rainer, Matthias VETTE a Matthias SCHOLER. *Inspector Robot – A New Collaborative Testing System Designed for the Automotive Final Assembly Line*. *Procedia CIRP* [online]. 2014, **23**, 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems, 59–64. ISSN 2212-8271. Dostupné z: doi:10.1016/j.procir.2014.10.093
- [9] GROBBELAAR, Wonda, Amit VERMA a Vinod Kumar SHUKLA. *Analyzing Human Robotic Interaction in the Food Industry*. *Journal of Physics: Conference Series* [online]. 2021, **1714**(1), 012032. ISSN 1742-6596. Dostupné z: doi:10.1088/1742-6596/1714/1/012032
- [10] IVAČKO, Nikola, Ivan ČIRIĆ, Žarko COJBAŠIĆ, Maša MILOŠEVIĆ a Dušan JEVTIĆ. *Machine Learning Based Hardware Accelerated Cobot Grasping in the Food Industry*. In: *2024 11th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN): 2024 11th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)* [online]. 2024, s. 1–5 [vid. 2025-05-19]. Dostupné z: doi:10.1109/IcETRAN62308.2024.10645149
- [11] ACCORSI, R., A. TUFANO, A. GALLO, F. G. GALIZIA, G. COCCHI, M. RONZONI, A. ABBATE a R. MANZINI. *An application of collaborative robots in a food production facility*. *Procedia Manufacturing* [online]. 2019, **38**, 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing, 341–348. ISSN 2351-9789. Dostupné z: doi:10.1016/j.promfg.2020.01.044

- [12] SCHNELL, Marie. Challenges for Manufacturing SMEs in the Introduction of Collaborative Robots. In: *ResearchGate* [online]. 2024 [vid. 2025-05-01]. Dostupné z: doi:10.3233/ATDE220137
- [13] VICENTINI, Federico. Terminology in safety of collaborative robotics. *Robotics and Computer-Integrated Manufacturing* [online]. 2020, **63**, 101921. ISSN 0736-5845. Dostupné z: doi:10.1016/j.rcim.2019.101921
- [14] *ISO 10218-1:2025(en), Robotics — Safety requirements — Part 1: Industrial robots* [online]. [vid. 2025-05-01]. Dostupné z: <https://www.iso.org/obp/ui/en/#iso:std:iso:10218:-1:ed-3:v1:en>
- [15] *ISO/TS 15066:2016(en), Robots and robotic devices — Collaborative robots* [online]. [vid. 2025-05-01]. Dostupné z: <https://www.iso.org/obp/ui/en/#iso:std:iso:ts:15066:ed-1:v1:en>
- [16] *Cobots - UR portfolio* [online]. [vid. 2025-05-17]. Dostupné z: <https://www.universal-robots.com/products/>
- [17] Browse all ABB robots. *Robotics* [online]. [vid. 2025-05-17]. Dostupné z: <https://new.abb.com/products/robotics/robots>
- [18] *The Ultimate Guide to Tool Center Point (TCP) in Robotics: Defining Positioning, Orientation, and Accuracy* [online]. 4. listopad 2024 [vid. 2025-05-05]. Dostupné z: <https://learnsphere.biz/learning/tcp-robotics-positioning-orientation-accuracy/>
- [19] UNIVERSAL ROBOTS A/S. *Universal Robots e-Series User Manual*
- [20] KIM, Tae-Wan, Kyu-Yeul LEE, Jongwon KIM, Min-Jae OH a Jie Hyeung LEE. Wireless Teaching Pendant for Mobile Welding Robot in Shipyard. *IFAC Proceedings Volumes* [online]. 2008, **41**(2), 17th IFAC World Congress, 4304–4309. ISSN 1474-6670. Dostupné z: doi:10.3182/20080706-5-KR-1001.00724
- [21] *What are robotic teach pendants? Features, types & applications - Standard Bots* [online]. [vid. 2025-05-06]. Dostupné z: <https://standardbots.com/blog/what-are-robotic-teach-pendants>
- [22] *Long Live the Robot Teach Pendant | 2020-12-02 | ASSEMBLY* [online]. [vid. 2025-05-16]. Dostupné z: <https://www.assemblymag.com/articles/96022-long-live-the-robot-teach-pendant>
- [23] *Stick–slip in hand guidance of palletizing robot as collaborative robot - Yun-Ju Chuang, Ho Chang, Yin-Tung Sun, Tsing-Tshih Tsung, 2022* [online]. [vid. 2025-05-19]. Dostupné z: <https://journals.sagepub.com/doi/full/10.1177/17298806221131138>
- [24] SAFEEA, Mohammad, Richard BEAREE a Pedro NETO. End-Effector Precise Hand-Guiding for Collaborative Robots. In: Anibal OLLERO, Alberto SANFELIU, Luis MONTANO, Nuno LAU a Carlos CARDEIRA, ed. *ROBOT 2017: Third Iberian Robotics Conference* [online]. Cham: Springer International Publishing, 2018, s. 595–605. ISBN 978-3-319-70836-2. Dostupné z: doi:10.1007/978-3-319-70836-2_49

- [25] ERDEN, Mustafa Suphi a Aude BILLARD. End-point impedance measurements at human hand during interactive manual welding with robot. In: *2014 IEEE International Conference on Robotics and Automation (ICRA): 2014 IEEE International Conference on Robotics and Automation (ICRA)* [online]. Hong Kong, China: IEEE, 2014, s. 126–133 [vid. 2025-05-19]. ISBN 978-1-4799-3685-4. Dostupné z: doi:10.1109/ICRA.2014.6906599
- [26] BALASUBRAMANIAN, Ravi, Ling XU, Peter D. BROOK, Joshua R. SMITH a Yoky MATSUOKA. Physical Human Interactive Guidance: Identifying Grasping Principles From Human-Planned Grasps. *IEEE Transactions on Robotics* [online]. 2012, **28**(4), 899–910. ISSN 1552-3098, 1941-0468. Dostupné z: doi:10.1109/TRO.2012.2189498
- [27] LEE, Hoo Man a Joong Bae KIM. A Survey on Robot Teaching: Categorization and Brief Review. *Applied Mechanics and Materials* [online]. 2013, **330**, 648–656. ISSN 1662-7482. Dostupné z: doi:10.4028/www.scientific.net/AMM.330.648
- [28] KUSHIDA, Daisuke, Masatoshi NAKAMURA, Satoru GOTO a Nobuhiro KYURA. Human direct teaching of industrial articulated robot arms based on force-free control. *Artificial Life and Robotics* [online]. 2001, **5**(1), 26–32. ISSN 1614-7456. Dostupné z: doi:10.1007/BF02481317
- [29] SAFEEA, Mohammad a Pedro NETO. Precise positioning of collaborative robotic manipulators using hand-guiding. *The International Journal of Advanced Manufacturing Technology* [online]. 2022, **120**(7), 5497–5508. ISSN 1433-3015. Dostupné z: doi:10.1007/s00170-022-09107-1
- [30] MOU, Samantha. How robot vendors are targeting integrated robot control. *Control Engineering* [online]. 30. květen 2024 [vid. 2025-05-06]. Dostupné z: <https://www.controleng.com/how-robot-vendors-are-targeting-integrated-robot-control/>
- [31] OPC UA speeds up the digitalization. *siemens.com Global Website* [online]. [vid. 2025-05-06]. Dostupné z: <https://www.siemens.com/global/en/products/automation/industrial-communication/opc-ua.html>
- [32] *ROBOTemplate datasheet* [online]. [vid. 2025-05-21]. Dostupné z: <https://search.abb.com/library/Download.aspx?DocumentID=9AKK107992A4786&LanguageCode=cs&DocumentPartId=&Action=Launch>
- [33] TROJAN, Ing Tomáš. Software ROBOTemplate usnadňuje programování robotů ABB. *Strojirenstvi.cz* [online]. 7. říjen 2020 [vid. 2025-05-21]. Dostupné z: <https://www.strojirenstvi.cz/software-robotemplate-usnadnuje-programovani-robotu-abb/>
- [34] *SIMATIC Robot Library* [online]. [vid. 2025-05-21]. Dostupné z: <https://it.profibus.com/prodotti/product-guide/simatic-robot-library>
- [35] *SRCI - Robotics* [online]. [vid. 2025-05-21]. Dostupné z: <https://www.profibus.com/technologies/robotics-srci-standard-robot-command-interface>

- [36] *SIMATIC Robot Library* [online]. [vid. 2025-05-21]. Dostupné z: <https://mall.industry.siemens.com/mall/en/us/Catalog/Product/6ES78230RE003AA0>
- [37] *URScript* [online]. [vid. 2025-05-07]. Dostupné z: <https://www.universal-robots.com/developer/urscript/>
- [38] *RTDE* [online]. [vid. 2025-05-07]. Dostupné z: <https://www.universal-robots.com/developer/communication-protocol/rtd/>
- [39] BRAUMANN, Johannes a Sigrid BRELL-COKCAN. Parametric Robot Control: Integrated CAD/CAM for Architectural Design. In: *ACADIA 2011: Integration Through Computation* [online]. 2011, s. 242–251 [vid. 2025-05-07]. Dostupné z: [doi:10.52842/conf.acadia.2011.242](https://doi.org/10.52842/conf.acadia.2011.242)
- [40] *Kuka sunrise toolbox* [online]. 7. květen 2025 [vid. 2025-05-07]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/65446-kuka-sunrise-toolbox>
- [41] SAFEEA, Mohammad a Pedro NETO. *KUKA Sunrise Toolbox: Interfacing Collaborative Robots with MATLAB* [online]. B.m.: arXiv. 5. září 2017 [vid. 2025-05-07]. Dostupné z: [doi:10.48550/arXiv.1709.01438](https://doi.org/10.48550/arXiv.1709.01438). arXiv:1709.01438 [cs]
- [42] TMflow. *Techman Robot* [online]. [vid. 2025-05-07]. Dostupné z: <https://www.tm-robot.com/en/tmflow/>
- [43] *RoboDK* [online]. [vid. 2025-05-07]. Dostupné z: https://www.mathworks.com/products/connections/product_detail/roboDK.html
- [44] GARBEV, Atanas a Atanas ATANASSOV. Comparative Analysis of RoboDK and Robot Operating System for Solving Diagnostics Tasks in Off-Line Programming. In: *2020 International Conference Automatics and Informatics (ICAI): 2020 International Conference Automatics and Informatics (ICAI)* [online]. 2020, s. 1–5 [vid. 2025-05-07]. Dostupné z: [doi:10.1109/ICAI50593.2020.9311332](https://doi.org/10.1109/ICAI50593.2020.9311332)
- [45] *Robot programming and simulation software | Offline Robotic Program | OLRP | OCTOPUZ* [online]. [vid. 2025-05-07]. Dostupné z: <https://www.octopuz.com/>
- [46] JANÍČEK, M, R RUŽAROVSKÝ, K VELÍŠEK a R HOLUBEK. Analysis of voice control of a collaborative robot. *Journal of Physics: Conference Series* [online]. 2021, **1781**(1), 012025. ISSN 1742-6596. Dostupné z: [doi:10.1088/1742-6596/1781/1/012025](https://doi.org/10.1088/1742-6596/1781/1/012025)
- [47] MOHD FAUDZI, Ahmad Athif, Muaammar ALI, M. AZMAN a Zool ISMAIL. Real-time Hand Gestures System for Mobile Robots Control. *Procedia Engineering* [online]. 2012, **41**, 798–804. Dostupné z: [doi:10.1016/j.proeng.2012.07.246](https://doi.org/10.1016/j.proeng.2012.07.246)
- [48] ROSENSTRAUCH, Martin J., Tessa J. PANNEN a Jörg KRÜGER. Human robot collaboration - using kinect v2 for ISO/TS 15066 speed and separation monitoring. *Procedia CIRP* [online]. 2018, **76**, 183–186. ISSN 22128271. Dostupné z: [doi:10.1016/j.procir.2018.01.026](https://doi.org/10.1016/j.procir.2018.01.026)
- [49] PARR, E. A. *Industrial Control Handbook*. B.m.: Industrial Press Inc., 1998. 3. ISBN 978-0-8311-3085-5.

- [50] LASHIN, Maha M. Different Applications of Programmable Logic Controller (PLC). *International Journal of Computer Science, Engineering and Information Technology* [online]. 2014, 4(1), 27–32. ISSN 22313605, 22313117. Dostupné z: doi:10.5121/ijcseit.2014.4103
- [51] Need A Siemens PLC Programmer? *PLC and SCADA Programming / Automation Specialists* [online]. [vid. 2025-05-16]. Dostupné z: <https://www.outlierautomation.com/siemens-plc-programming>
- [52] *PLC Hardware Explained - RealPars* [online]. [vid. 2025-05-07]. Dostupné z: <https://www.realpars.com/blog/plc-hardware>
- [53] ROOS, Nieke. *Programming PLCs using Structured Text*. B.m.: Department of Computing Science University of Nijmegen
- [54] *Structured text* [online]. 2025 [vid. 2025-05-19]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Structured_text&oldid=1283446019
- [55] JOHN, Karl Heinz a Michael TIEGELKAMP. *IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids* [online]. Berlin, Heidelberg: Springer, 2010 [vid. 2025-05-07]. ISBN 978-3-642-12014-5. Dostupné z: doi:10.1007/978-3-642-12015-2
- [56] *Ladder logic* [online]. 2025 [vid. 2025-05-19]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Ladder_logic&oldid=1285242267
- [57] *PLC Programming With Function Block Diagrams - Technical Articles* [online]. [vid. 2025-05-19]. Dostupné z: <https://control.com/technical-articles/plc-programming-with-function-block-diagrams/>
- [58] *Sequential function chart* [online]. 2024 [vid. 2025-05-21]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Sequential_function_chart&oldid=1231864402
- [59] *Allen Bradley PLC Programming Tutorials | Training in RSLogix 5000* [online]. [vid. 2025-05-19]. Dostupné z: <https://www.solisplc.com/tutorials/plc-programming-tutorial-allen-bradley-training-in-rslogix-5000-ladder-logic-basics-for-beginners>
- [60] SIEMENS. *IEC 61158 Working with ROFIBUS-DP* [online]. duben 2003. Dostupné z: https://cache.industry.siemens.com/dl/files/306/1172306/att_39239/v1/gsd_e.pdf
- [61] PROFIBUS NUTZERORGANISATION. *PROFINET System Description* [online]. Dostupné z: <https://www.profibus.com/index.php?eID=dumpFile&t=f&f=51714&token=4ea5554cbb80a066e805a879116ead2a759c23c3>
- [62] CAVALIERI, Salvatore a Ferdinando CHIACCHIO. Analysis of OPC UA performances. *Computer Standards & Interfaces* [online]. 2013, 36(1), 165–177. ISSN 0920-5489. Dostupné z: doi:10.1016/j.csi.2013.06.004

- [63] *The ABCs of OPC UA: Everything You Need to Understand - RealPars* [online]. [vid. 2025-05-08]. Dostupné z: <https://www.realpars.com/blog/opc-ua-basics>
- [64] BAYINDIR, Ramazan a Yucel CETINCEVIZ. A water pumping control system with a programmable logic controller (PLC) and industrial wireless modules for industrial plants—An experimental setup. *ISA Transactions* [online]. 2011, **50**(2), 321–328. ISSN 0019-0578. Dostupné z: doi:10.1016/j.isatra.2010.10.006
- [65] SAAD, Nordin a M. ARROFIQ. A PLC-based modified-fuzzy controller for PWM-driven induction motor drive with constant V/Hz ratio control. *Robotics and Computer-Integrated Manufacturing* [online]. 2012, **28**(2), 95–112. ISSN 0736-5845. Dostupné z: doi:10.1016/j.rcim.2011.07.001
- [66] NAMEKAR, Swapnil Arun a Rishabh YADAV. Programmable Logic Controller (PLC) and its Applications. 2020, **6**(11).
- [67] *Siemens Supplying Automation For Electric Car Production At Volkswagen Plant - Manufacturing AUTOMATION* [online]. 9. března 2020 [vid. 2025-05-21]. Dostupné z: <https://www.automationmag.com/siemens-supplying-automation-for-electric-car-production-at-volkswagen-plant/>
- [68] Volkswagen Sachsen GmbH Zwickau Plant. *Volkswagen Newsroom* [online]. [vid. 2025-05-21]. Dostupné z: <https://www.volkswagen-newsroom.com/en/volkswagen-sachsen-gmbh-zwickau-plant-5901>
- [69] Pharma Control Improvements Using Siemens SIMATIC » Patti Engineering. *Patti Engineering* [online]. [vid. 2025-05-21]. Dostupné z: <https://www.pattiengineering.com/case-studies/pharma-control-improvements-using-siemens-simatic/>
- [70] Streamlining Packaging: Maverick Boosts Efficiency with OMRON. *Automate* [online]. [vid. 2025-05-21]. Dostupné z: <https://www.automate.org/motion-control/case-studies/maverick-international-optimizes-packaging-changeover-and-sealing-performance>
- [71] PAPCUN, Peter, Erik KAJÁTI a Jiří KOZIOREK. Human Machine Interface in Concept of Industry 4.0. In: *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA): 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)* [online]. 2018, s. 289–296 [vid. 2025-05-08]. Dostupné z: doi:10.1109/DISA.2018.8490603
- [72] ISA-101 Series of Standards. *isa.org* [online]. [vid. 2025-05-08]. Dostupné z: <https://www.isa.org/standards-and-publications/isa-standards/isa-101-standards>
- [73] PANTER, Lars, Rieke LEDER, Dennis KEISER a Michael FREITAG. Requirements for Human-Machine-Interaction Applications in Production and Logistics within Industry 5.0 – A Case Study Approach. *Procedia Computer Science* [online]. 2024, **232**, 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023), 1164–1171. ISSN 1877-0509. Dostupné z: doi:10.1016/j.procs.2024.01.114

- [74] *Group Safety Standards IEC60204-1: 2016 / USA* [online]. [vid. 2025-05-17]. Dostupné z: <https://us.idec.com/RD/safety/law/iso-iec/iec60204>
- [75] How ISA-101 Lifecycle Standard Improves Operator Effectiveness with Display Design and Lifecycle Management. *ARC Advisory Group* [online]. [vid. 2025-05-08]. Dostupné z: <https://www.arcweb.com/industry-best-practices/how-isa-101-lifecycle-standard-improves-operator-effectiveness-display>
- [76] TABAA, Mohamed, Brahim CHOURI, Safa SAADAOUI a Karim ALAMI. Industrial Communication based on Modbus and Node-RED. *Procedia Computer Science* [online]. 2018, **130**, The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops, 583–588. ISSN 1877-0509. Dostupné z: doi:10.1016/j.procs.2018.04.107
- [77] GRINBERG, Miguel. *Flask Web Development*. B.m.: O'Reilly Media, Inc., 2018. ISBN 978-1-4919-9169-5.
- [78] WIMPY. Perawatan HMI / Human-Machine Interface. *SIMENTEK News* [online]. 25. červen 2024 [vid. 2025-05-19]. Dostupné z: <https://simenteknindo.com/news/perawatan-hmi-human-machine-interface/>
- [79] HUI, Henry a Kieran MCLAUGHLIN. Investigating Current PLC Security Issues Regarding Siemens S7 Communications and TIA Portal. In: *5th International Symposium for ICS & SCADA Cyber Security Research 2018* [online]. 2018 [vid. 2025-05-08]. Dostupné z: doi:10.14236/ewic/ICS2018.8
- [80] SIEMENS. SIMATIC TIA Portal STEP 7 Basic V10.5. 2009.
- [81] *Which PLC is Mostly Used in Industry? Top PLCs Explained* [online]. 1. prosinec 2024 [vid. 2025-05-08]. Dostupné z: <https://kwoco-plc.com/most-used-plc-in-industry/>
- [82] Exploring the Diversity of Siemens PLCs: Types and Applications - ControlNexus | Siemens Industrial Control Products Specialist. <https://plcvfd.com/> [online]. [vid. 2025-05-08]. Dostupné z: <https://plcvfd.com/exploring-the-diversity-of-siemens-plcs-types-and-applications/>
- [83] *Profinet Guide - 20596* [online]. [vid. 2025-05-12]. Dostupné z: <https://www.universal-robots.com/articles/ur/interface-communication/profinet-how-to-guide-e-series/>
- [84] ROSSUM, Guido van a Fred L. DRAKE. *An introduction to Python: release 2.5*. 2. print. Bristol: Network Theory Limited, 2006. ISBN 978-0-9541617-6-7.
- [85] CHUN, Wesley. *Core Python programming*. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2007. Prentice Hall core series. ISBN 978-0-13-226993-3.
- [86] Python Release Python 3.13.3. *Python.org* [online]. [vid. 2025-05-13]. Dostupné z: <https://www.python.org/downloads/release/python-3133/>
- [87] International Journal of Advance Engineering and Research Development [online]. nedatováno [vid. 2025-05-17]. ISSN 2348-6406, 2348-4470. Dostupné z: doi:10.21090/ijaerd

- [88] Most used languages among software developers globally 2024. *Statista* [online]. [vid. 2025-05-13]. Dostupné z: <https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>
- [89] ULTRALYTICS. *ROS Quickstart* [online]. [vid. 2025-05-21]. Dostupné z: <https://docs.ultralytics.com/guides/ros-quickstart>
- [90] COPPERWAITE, Matt a Charles LEIFER. *Learning Flask Framework*. B.m.: Packt Publishing Ltd, 2015. ISBN 978-1-78398-337-7.
- [91] DWYER, Gareth. *Flask By Example*. B.m.: Packt Publishing Ltd, 2016. ISBN 978-1-78528-348-2.
- [92] functools — Higher-order functions and operations on callable objects. *Python documentation* [online]. [vid. 2025-05-13]. Dostupné z: <https://docs.python.org/3/library/functools.html>
- [93] KENNEDY, Patrick. *What is Werkzeug?* [online]. [vid. 2025-05-13]. Dostupné z: <https://testdriven.io/blog/what-is-werkzeug/>
- [94] *Werkzeug: The comprehensive WSGI web application library*. OS Independent. Python
- [95] *Why OPC UA Matters* [online]. [vid. 2025-05-13]. Dostupné z: <https://www.ni.com/en/solutions/industrial-machinery/smart-machine-control/why-opc-ua-matters.html>
- [96] *Průmyslová komunikace OPC UA - 1.díl - popis protokolu | Automatizace.HW.cz* [online]. [vid. 2025-05-13]. Dostupné z: <https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html>
- [97] MUSCIANO, Chuck a Bill KENNEDY. *HTML & XHTML: The Definitive Guide: The Definitive Guide*. B.m.: O'Reilly Media, Inc., 2002. ISBN 978-1-4493-9085-3.
- [98] Angular vs React vs Vue: Core Differences. *BrowserStack* [online]. [vid. 2025-05-14]. Dostupné z: <https://browserstack.wpengine.com/guide/angular-vs-react-vs-vue/>
- [99] MORAES, Frank. *HTML For Beginners The Easy Way: Start Learning HTML & CSS Today. HTML* [online]. [vid. 2023-05-21]. Dostupné z: <https://html.com/>
- [100] MEYER, Eric A. *CSS: The Definitive Guide: The Definitive Guide*. B.m.: O'Reilly Media, Inc., 2006. ISBN 978-1-4493-9725-8.
- [101] *Bootstrap vs. Tailwind CSS: Compare The Top CSS Frameworks* [online]. [vid. 2025-05-14]. Dostupné z: <https://strapi.io/blog/bootstrap-vs-tailwind-css-a-comparison-of-top-css-frameworks>
- [102] FLANAGAN, David. *JavaScript: The Definitive Guide*. B.m.: O'Reilly Media, Inc., 2011. ISBN 978-0-596-80552-4.

- [103] *What is JavaScript? - Learn web development / MDN* [online]. 15. duben 2025 [vid. 2025-05-14]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript
- [104] PEPPER, C., S. BALAKIRSKY a C. SCRAPPER. Robot simulation physics validation. In: *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems* [online]. New York, NY, USA: Association for Computing Machinery, 2007, s. 97–104 [vid. 2025-05-16]. PerMIS '07. ISBN 978-1-59593-854-1. Dostupné z: doi:10.1145/1660877.1660890
- [105] BARTH, Rolf, Matthias MEYER a Jan SPITZNER. Typical Pitfalls of Simulation Modeling - Lessons Learned from Armed Forces and Business. *Journal of Artificial Societies and Social Simulation*. 2011, **15**(2), 5. ISSN 1460-7425.
- [106] Digital Twins in the Machinery & Robotics Industry. *Unity* [online]. [vid. 2025-05-17]. Dostupné z: <https://unity.com/blog/digital-twins-machinery-robotics-industry>
- [107] *CPU 1515F-2 PN* [online]. [vid. 2025-05-09]. Dostupné z: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7515-2FN03-0AB0>
- [108] *SCALANCE XC-200 managed* [online]. [vid. 2025-05-09]. Dostupné z: <https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6GK5216-0BA00-2AC2>
- [109] *Digital input modules* [online]. [vid. 2025-05-09]. Dostupné z: <https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7131-6BF01-0BA0>
- [110] *UR10e Medium-sized, versatile cobot* [online]. [vid. 2025-05-15]. Dostupné z: <https://www.universal-robots.com/products/ur10e/>
- [111] *Adaptive Grippers / Robotiq* [online]. [vid. 2025-05-15]. Dostupné z: <https://robotiq.com/products/adaptive-grippers>
- [112] *BOS015U (BOS 5K-PS-RH12-S49) Difuzní snímače - BALLUFF Czech Republic* [online]. [vid. 2025-05-15]. Dostupné z: <https://www.balluff.com/cs-cz/products/BOS015U>

ZOZNAM SKRATIEK

API – Application Programming Interface
CSS – Cascading Style Sheets
FB – Funkčný Blok
GSD – General Station Description
HMI – Human Machine Interface
HTML – HyperText Markup Language
IEC – International Electrotechnical Commission
IEEE – Institute of Electrical and Electronic Engineers
IoT – Internet of Things
ISA – International Standards on Auditing
ISO – International Organization for Standardization
ISO/TS – International Organization for Standardization/Technical Specifications
KRL – KUKA Robot Language
MES – Manufacturing Execution Systems
MQTT – Message Queuing Telemetry Transport
OEM – Original Equipment Manufacturer
OPC UA – Open Platform Communications Unified Architecture
PLC – Programmable Logic Controller
PROFIBUS – Process Field Bus
PROFINET – Process Field Network
PROFINET IRT – Process Field Network Isochronous Real Time
PROFINET RT – Process Field Network Real Time
RAM – Random-access memory
ROM – Read-only memory
ROS – Robot Operating System
RTDE – Real Time Data Exchange
SCADA – Supervisory Control and Data Acquisition
TCP – Tool Center Point
TIA Portal – Totally Integrated Automation Portal
TON – Timer ON – generate on-delay
TP – Teach Pendant
UDT – User Data Type
UR – Universal Robots
URL – Uniform Resource Locator

ZOZNAM OBRÁZKOV

- Obr. 1: Roboty Universal Robots [1]
- Obr. 2: Teach pendanty [22]
- Obr. 3: PLC od firmy Siemens [51]
- Obr. 4: Schéma hardvéru PLC [53]
- Obr. 5: Človek ovládajúci stroj pomocou HMI [78]
- Obr. 6: Architektúra ovládania
- Obr. 7: Rozhranie programu TIA Portal
- Obr. 8: Vizualizácia štruktúry funkčného bloku
- Obr. 9: Vývojový diagram priebehu FB
- Obr. 10: Štruktúra programových blokov v TIA Portal
- Obr. 11: Šablóna UR programu
- Obr. 12: Príklad UR funkcie
- Obr. 13: Vývojový diagram ovládania UR robota cez PLC
- Obr. 14: Login Stránka
- Obr. 15: Stránka home po prihlásení účtom Admin
- Obr. 16: Stránka home po prihlásení účtom Spectator
- Obr. 17: Stránka home s otvoreným sidebarom
- Obr. 18: Stránka robot
- Obr. 19: Stránka error
- Obr. 20: Simulované stanovisko v Unity3D
- Obr. 21: Siemens PLC SIMATIC S7-1500, zdroj, karty a switch
- Obr. 22: Zjednodušený vývojový diagram ovládania UR robota cez Python
- Obr. 23: Robot pri vykonávaní prvého testovacieho programu
- Obr. 24: Robot UR10e pri vytváraní datasetu v druhom testovacom programe
- Obr. 25: Robot UR10e pri ukladaní náhodne rozmiestnených kociek do platformy



ZOZNAM VÝPISOV KÓDOV

Výpis 1: Spojenie OPC UA, zapisovanie dát

Výpis 2: Definovanie dekorátorov

Výpis 3: Prístup k funkcii pre užívateľa admin

ZOZNAM TABULIEK

Tab. 1: Vstupné požiadavky pre FB

Tab. 2: Výstupné požiadavky pre FB

Tab. 3: Popis stavov stavového automatu

Tab. 4: Adresy pre kontinuálne zapisované hodnoty

Tab. 5: Adresy pre kontinuálne čítané hodnoty vrátane informácií o kľúčoch a TCP

Tab. 6: Adresy pre kontinuálne čítané hodnoty Bool, Int a Real

Tab. 7: Adresy pre zapisované hodnoty Bool

Tab. 8: Adresy pre zapisované hodnoty Int

Tab. 9: Adresy pre zapisované hodnoty Real



ZOZNAM PRÍLOH

Užívateľská dokumentácia k programu.zip