

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: B0715A270013 – Strojní inženýrství
Studijní specializace: Průmyslové inženýrství a management

BAKALÁŘSKÁ PRÁCE

Programování průmyslového robota s využitím rozšířené reality

Autor: Jan ŠVESTKA
Vedoucí práce: Ing. Jakub MÜLLER

Akademický rok 2023/2024

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta strojní

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan ŠVESTKA**
Osobní číslo: **S23B0260P**
Studijní program: **B0715A270013 Strojní inženýrství**
Specializace: **Průmyslové inženýrství a management**
Téma práce: **Programování průmyslového robota s využitím rozšířené reality**
Zadávací katedra: **Katedra průmyslového inženýrství a managementu**

Zásady pro vypracování

- Průmyslová robotika
- Rozšířená realita v průmyslu
- Analýza současného stavu
- Návrh řešení
- Vyhodnocení

Rozsah bakalářské práce: **30 – 50**
Rozsah grafických prací: **–**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. NOVOTNÝ, František. Úvod do automatizace a robotizace ve strojírenství. Liberec: Technická univerzita v Liberci, 2020. ISBN 978-80-7494-545-8
2. FORD, Martin. Roboti nastupují: automatizace, umělá inteligence a hrozba budoucnosti bez práce. Praha: Rybka Publishers, 2017. ISBN 978-80-87950-46-3.
3. CAI, Zixing. *Robotics: From Manipulator to Mobilebot*. Singapore: World Scientific Publishing Company, 2022. ISBN 978-9811253461.
4. KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. Brno: Vysoké učení technické v Brně, 2016. ISBN 978-80-214-4828-5.
5. CHLEBNÝ, Jan a kol. *Automatizace a automatizační technika: Prostředky automatizační techniky*. Brno: Computer Press, 2014. ISBN 978-80-251-3747-5.

Vedoucí bakalářské práce: **Ing. Jakub Müller**
Katedra průmyslového inženýrství a managementu

Konzultant bakalářské práce: **Doc. Ing. Petr Hořejší, Ph.D.**
Katedra průmyslového inženýrství a managementu

Datum zadání bakalářské práce: **16. října 2023**
Termín odevzdání bakalářské práce: **24. května 2024**

L.S.

Doc. Ing. Vladimír Duchek, Ph.D.
děkan

Doc. Ing. Michal Šimon, Ph.D.
vedoucí katedry

Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

V Plzni dne:

.....

podpis autora

Poděkování

Rád bych vyjádřil upřímné poděkování Ing. Jakubovi Müllerovi za jeho cenné rady, odborné vedení a okamžitou podporu poskytnutou během psaní této bakalářské práce. Dále bych chtěl poděkovat doc. Ing. Petrovi Hořejšímu, Ph.D., za jeho konzultace a podnětné připomínky, které přispěly k vylepšení obsahu práce, a týmu Robo@FIT za jejich podporu a přístupnost během celé implementace.

ANOTAČNÍ LIST BAKALÁŘSKÉ PRÁCE

AUTOR	Příjmení Švestka	Jméno Jan	
STUDIJNÍ PROGRAM	B0715A270013 Strojní inženýrství		
VEDOUCÍ PRÁCE	Příjmení (včetně titulů) Ing. Müller	Jméno Jakub	
PRACOVISŤE	ZČU - FST – KPV		
DRUH PRÁCE	DIPLOMOVÁ	BAKALÁŘSKÁ	Nehodící se škrtněte
NÁZEV PRÁCE	Programování průmyslového robota s využitím rozšířené reality		

FAKULTA	strojní	KATEDRA	KPV	ROK ODEVZD.	2024
----------------	---------	----------------	-----	--------------------	------

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	68	TEXTOVÁ ČÁST	64	GRAFICKÁ ČÁST	4
---------------	----	---------------------	----	----------------------	---

<p style="text-align: center;">STRUČNÝ POPIS (MAX 10 ŘÁDEK)</p> <p>ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY</p>	<p>Bakalářská práce se zaměřuje na využití rozšířené reality v průmyslové robotice s hlavním cílem implementace vývojové platformy ARCOR2 u kolaborativního robota od společnosti Fanuc. Rešerše práce se zabývá detailním popisem oblastí robotiky, průmyslových robotů, rozšířené reality a jejich aplikací v průmyslu. V praktické části práce je pak podrobně analyzován proces implementace platformy ARCOR2 a tvorba Python programu pro přesouvání objektů. V závěru práce jsou shrnuty dosažené výsledky a diskutovány přínosy a možnosti dalšího rozvoje.</p>
<p style="text-align: center;">KLÍČOVÁ SLOVA</p> <p style="text-align: center;">ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE</p>	<p style="text-align: center;">programování, kolaborativní robot, průmyslová robotika, rozšířená realita, implementace, automatizace</p>

SUMMARY OF BACHELOR SHEET

AUTHOR	Surname Švestka	Name Jan		
STUDY PROGRAMME	B0715A270013 Mechanical Engineering			
SUPERVISOR	Surname (Inclusive of Degrees) Ing. Müller		Name Jakub	
INSTITUTION	ZČU - FST - KPV			
TYPE OF WORK	DIPLOMA	BACHELOR	Delete when not applicable	
TITLE OF THE WORK	Programming an industrial robot using augmented reality			

FACULTY	Mechanical Engineering	DEPARTMENT	KPV	SUBMITTED IN	2024
----------------	------------------------	-------------------	-----	---------------------	------

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	68	TEXT PART	64	GRAPHICAL PART	4
----------------	----	------------------	----	-----------------------	---

BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS	<p>The bachelor's thesis focuses on the use of augmented reality in industrial robotics with the main goal of implementing the ARCOR2 development platform for a collaborative robot from Fanuc. The research part of the thesis provides a detailed description of the areas of robotics, industrial robots, augmented reality, and their applications in the industry. In the practical part of the thesis, the process of implementing the ARCOR2 platform and the creation of a Python program for moving objects are analyzed in detail. The conclusion of the thesis summarizes the achieved results and discusses the benefits and possibilities for further development.</p>
KEYWORDS	<p>programming, collaborative robot, industrial robotics, augmented reality, implementation, automation</p>

Obsah

Seznam obrázků	9
Seznam tabulek	10
Úvod	12
1 Robotika	13
1.1 Historie	14
1.2 Vývoj robotiky v průmyslu	15
1.3 Využití robotů.....	16
1.4 Průmyslové roboty a jejich systémy	18
1.5 Rozdělení průmyslových robotů.....	22
2 Rozšířená realita	26
2.1 Historie rozšířené reality	27
2.2 Vývoj rozšířené reality	27
2.3 Zařízení pro rozšířenou realitu	29
2.4 Využití AR v průmyslu	30
2.5 Využití AR v robotice	32
3 ARCOR2	34
3.1 Architektura.....	34
3.2 Docker	35
3.3 Uživatelské prostředí AREditoru	35
4 Ovládání kolaborativního robota.....	37
4.1 Pracoviště	37
4.2 Použití ARCOR2 pro ovládání kolaborativního robota	38
4.2.1 Virtuální zkouška	39
4.2.2 Nahrání knihovny fanucpy do řídicí jednotky	41
4.2.3 Praktická zkouška ovládání robota	46
4.2.4 Proces implementace a zhodnocení.....	50
4.3 Použití Python programu pro ovládání kolaborativního robota	51
4.3.1 Popis experimentu	51
4.3.2 Popis programu.....	51
4.3.3 Testování programu.....	55
4.3.4 Možnosti dalšího rozvoje a optimalizace programu.....	57
Závěr.....	58
Seznam použitých zdrojů	60
Seznam příloh.....	64

Seznam obrázků

Obrázek 1-1 - Robotický vysavač[6]	13
Obrázek 1-2 - Popis robota Elsie bez ochranného štítu[8].....	14
Obrázek 1-3 - Časová osa implementace průmyslových robotů[11]	15
Obrázek 1-4 - Každoroční instalace průmyslových robotů 2017-2022 a 2023*-2026*[12]....	16
Obrázek 1-5 - Chirurgický robot Da Vinci Xi[14].....	16
Obrázek 1-6 - Autonomní doručovací robot X1 PRO[15].....	17
Obrázek 1-7 - Taktický robot TALON 5[16].....	17
Obrázek 1-8 - Robotická ponorka AQUANAUT[17].....	18
Obrázek 1-9 - Blokové schéma řízení průmyslového robota [19]	18
Obrázek 1-10 - Popis průmyslového robota Kuka[20]	19
Obrázek 1-11 - Dělení průmyslových robotů dle geometrie pracovního prostoru[21].....	20
Obrázek 1-12 - Konec robotického ramena s rozdělením efektorů[21; 22].....	20
Obrázek 1-13 - Tablet Teach Pendant s popisem[24]	22
Obrázek 1-14 - Kloubový robot ABB IRB 8700[25].....	23
Obrázek 1-15 – Delta robot IRB 390 [25].....	23
Obrázek 1-16 - SCARA robot IRB 910SC[25].....	24
Obrázek 1-17 - Paletizační robot IRB 460 [25]	24
Obrázek 1-18 - Lakovací robot IRB 5510[25]	25
Obrázek 1-19 - Kolaborativní robot YuMi IRB 14000 [25]	25
Obrázek 2-1 - Využití rozšířené reality v praxi[27]	26
Obrázek 2-2 - Pokrok v rozšířené realitě [28].....	27
Obrázek 2-3 - Prostředí Apple Vision Pro[29].....	28
Obrázek 2-4 - Výrobní linka přes aplikaci Stages v Apple Vision Pro[30].....	28
Obrázek 2-5 - AR Marker[48].....	29
Obrázek 2-6 - LightGuide's Projected AR[32]	30
Obrázek 2-7 - Využití AR automobilkou BMW[34]	31
Obrázek 2-8 - Sledování a řízení testovacího zařízení od AV ENGINEERING[37]	32
Obrázek 2-9 - AR aplikace RobotStudio® AR Viewer ve společnosti Volvo[39].....	33
Obrázek 2-10 - Ovládání kolaborativního robota s Kinetic AR[40].....	33
Obrázek 3-1 - Blokové schéma systému ARCOR2[43].....	34
Obrázek 3-2 - Hlavní obrazovka AREditoru[42]	35
Obrázek 3-3 - Hlavní pracovní prostředí AREditoru	36
Obrázek 4-1 - Kolaborativní robot Fanuc CR-7iA/L	37
Obrázek 4-2 - Řídicí jednotka R-30iB Mate Plus	38

Obrázek 4-3 - Fanuc iPendant-Touch	38
Obrázek 4-4 - Docker fit-demo ARServer	39
Obrázek 4-5 - Připojení k serveru AREditor.....	39
Obrázek 4-6 - Kalibrační krychle.....	40
Obrázek 4-7 - Objekt DobotMagician.....	40
Obrázek 4-8 - Posun robota při virtuální zkoušce.....	41
Obrázek 4-9 - Komunikační protokol mezi Python a řídicím systémem robota FANUC[46]	42
Obrázek 4-10 - Modul R632 - KAREL.....	42
Obrázek 4-11 - Modul R648 - User Socket Messaging	42
Obrázek 4-12 - Pendant nastavení Host Comm	43
Obrázek 4-13 - Pendant nastavení Variables	44
Obrázek 4-14 - Pendant nastavení Tags	44
Obrázek 4-15 - Pendant nahrání MAPPDK	45
Obrázek 4-16 - Ověření připojení ping a ARP	46
Obrázek 4-17 - Docker fanuc-demo ARServer.....	47
Obrázek 4-18 - Model fanuc_1_r_mate200id7_1.....	48
Obrázek 4-19 - Chybové hlášení "Catastrophic system error"	48
Obrázek 4-20 - Výpis příkazového řádku při chybě	49
Obrázek 4-21 - Pendant chyba INTP-311	49
Obrázek 4-22 - Pozice puků pro Python program.....	51
Obrázek 4-23 - Připojení k robotu.....	52
Obrázek 4-24 - Definice pozic pro jednotlivé barvy.....	52
Obrázek 4-25 - Funkce pro ovládání robota.....	53
Obrázek 4-26 - Uživatelský vstup programu	54
Obrázek 4-27 - Metoda pro zadání počtu puků jednotlivých barev	54
Obrázek 4-28 - Spuštění hlavního programu	55
Obrázek 4-29 - Pozice po uchopení puku	56
Obrázek 4-30 - Pozice pro uvolnění puku.....	56

Seznam tabulek

Tabulka 1 - Časový průběh implementace ARCOR2	50
---	----

Seznam zkratk

LiDAR	Light detection and ranging
m	Metr
kg	Kilogram
ms	Milisekunda
IP	Internetový protokol
AR	Rozšířená realita
VR	Virtuální realita
HDM	Head mounted display
SAR	Spatial augmented reality
REST API	Representational state transfer application programming interface
HTTP	Hypertext transfer protocol
LED	Light emitting diode
CAD	Computer aided design
ARP	Address resolution protocol
ARCOR2	Augmented reality collaborative robot
ZČU	Západočeská univerzita v Plzni

Úvod

V průmyslových odvětvích se v posledních letech stává stále důležitější využití moderních technologií ke zvyšování efektivity a produktivity. Jednou z inovativních oblastí je spojení průmyslového robotického systému s rozšířenou realitou, která umožňuje intuitivnější programování a správu pracovišť.

Cílem této bakalářské práce je implementace a následné využití mobilní rozšířené reality vývojové platformy ARCOR2 pro zjednodušené programování kolaborativního robota od společnosti Fanuc, který je součástí Katedry průmyslového inženýrství a managementu na Fakultě strojní Západočeské univerzity v Plzni. Platformu ARCOR2 vyvinul tým Robo@FIT z Fakulty informačních technologií Vysokého učení technického v Brně. Kromě samotné implementace je záměrem této práce i ověření funkčnosti a spolehlivosti platformy ARCOR2 v reálných pracovních podmínkách a identifikace možných výzev a problémů při implementaci a používání této technologie.

První část rešerše obsahuje různé definice robotiky, její historii, vývoj a pohled na využití robotů v různých oblastech, jako je zdravotnictví, výzkum a další. Dále jsou rozebrány konkrétní aspekty průmyslových robotů, včetně jejich pohybového systému, koncových efektorů, pohonu a řídicího systému. Z katalogu výrobce průmyslových robotů, společnosti ABB, je provedeno jejich rozdělení a základní popis každého z nich.

Další kapitola se věnuje rozšířené realitě, její historii, vývoji a inovativním brýlím pro smíšenou realitu Vision Pro od společnosti Apple. Pro podání uceleného obrazu rozšířené reality jsou rozebrány potřebné zařízení. Závěr rešerše je zaměřen na reálné využití rozšířené reality v oblasti průmyslu a robotiky.

V úvodu praktické části jsou představeny klíčové principy a architektura platformy ARCOR2, včetně detailního popisu využití nástroje Docker pro spuštění serveru. Dále se zaměřuje na analýzu hlavní obrazovky a editoru mobilní aplikace AREditor, která slouží pro interakci s roboty a umožňuje uživatelům intuitivně programovat jejich pohyby a úkoly. Práce pokračuje implementací platformy ARCOR2 na pracovišti, které je podrobně popsáno spolu s jeho vybavením. V následující části je provedena virtuální zkouška modelu robota, následovaná nahráním knihovny fanucpy do řídicí jednotky robota a praktickou zkouškou s reálným kolaborativním robotem od společnosti Fanuc.

1 Robotika

Robotika je věda, která spojuje inženýrství a informatiku, jejímž cílem je vytvořit chytré stroje, které jsou schopny asistovat lidem s ulehčováním a zvyšováním produktivity práce v různých činnostech a prostředích. Zjednodušeně řečeno, zabývá se návrhem, výrobou a provozem robotů, které jsou navrženy a naprogramovány tak, aby efektivně plnily úkoly, které by pro člověka mohly být obtížné, nebezpečné, nebo opakující se.[1]

Definice dle knihy *The Robotics Primer*: „*Robotika je studium robotů, jejich schopnosti autonomního a účelného vnímání a působení ve fyzickém světě.*“ (překlad vlastní, Maja J., 2007, s. 5)[2]

Definice dle článku v *IEEE Robotics & Automation Magazine*: Robotika je interdisciplinární obor, který se zaměřuje na vývoj a konstrukci systémů, které spojují elektrické a mechanické prvky s počítačovou technologií.[3]

Roboty jsou všude kolem nás, ale každý si pod nimi představí něco jiného. Někteří vidí roboty z filmů, kteří mají snahu o zničení světa a jsou vyspělejší než lidstvo, ale ve srovnání s lidskými schopnostmi jsou však roboty stále ve stádiu raného vývoje. Jsou nepostradatelní, zejména na místech, kde by práce byla pro lidi nebezpečná, a představují nejen budoucnost výroby, ale i způsob, jakým pracujeme a žijeme. Definovat robota může být obtížné, ačkoli existují některé společné rysy, jako je programovatelnost, multifunkčnost, schopnost manipulace s objekty a provádění konkrétních výrobních úkonů. Mezinárodní organizace pro standardizaci definovala robota v normě ISO 8373 jako „*automaticky řízený, opětovně programovatelný, víceúčelový manipulátor pro činnost ve třech nebo více osách, který může být buď upevněn na místě nebo mobilní k užití v průmyslových automatických aplikacích.*“ (Volná a Kotyba, 2013, s. 97) [4; 5]

Dobře známé a široce využívané zařízení, které demonstruje praktické využití robotů v běžném životě je robotický vysavač, který je navržen k autonomnímu úklidu domácnosti. Využívá senzory a algoritmy k mapování a navigaci v prostoru a dokáže čistit podlahy bez naší přímé asistence. Na obrázku 1-1 lze pozorovat robotický vysavač se třemi LiDAR senzory. Díky těmto sensorům si vysavač vytváří detailní 3D mapu prostoru bytu, která mu umožňuje efektivní navigování, a zabezpečuje tak účinné čištění. [6]



Obrázek 1-1 - Robotický vysavač[6]

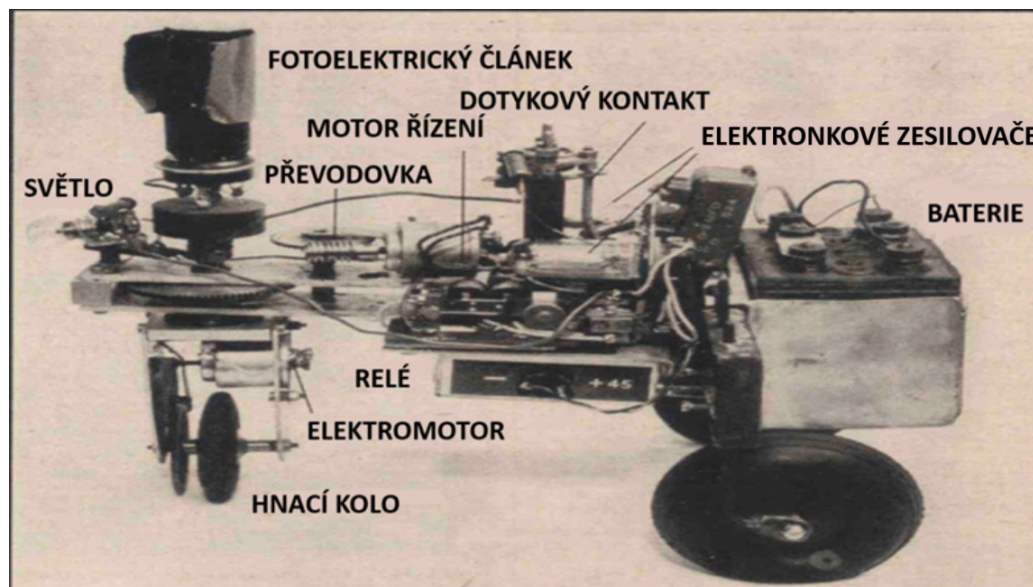
V následující podkapitole se podíváme na významné okamžiky a osobnosti v historii robotiky.

1.1 Historie

Historie robotiky sahá do minulosti a zahrnuje několik klíčových momentů a milníků, které formovaly tuto oblast. První zmínky o humanoidních strojích se objevují ve starověkém Řecku a Egyptě. Řečtí vědci, jako Archytas z Tarentu, vytvářeli první mechanické zařízení, zatímco v Egyptě byly nalezeny sochy s pohyblivými částmi. Mezi další osobnosti patří například Aristoteles s jeho úvahami nad nástroji, které by nahradily otroky. Později přišel Henry Ford s inovativními myšlenkami v oblasti automatizace průmyslové výroby, a také Leonardo da Vinci, který vytvořil koncept mechanického rytíře, který by mohl být považován za předchůdce moderních robotů. Slovo robot, odvozené od slova robota, prvně použil ve svém divadelním dramatu R.U.R český spisovatel Karel Čapek. Díky jeho geniální tvorbě a přínosu v oblasti umělé inteligence a automatizace se tento termín stal světově používaným. Dalším důležitým bodem byl příchod Isaaca Asimova, který poprvé použil termín robotika a definoval Tři zákony robotiky, které zní takto:

- „1. Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby člověku bylo ublíženo.*
- 2. Robot musí uposlechnout příkazů člověka, kromě případů, kdy tyto příkazy jsou v rozporu s prvním zákonem.*
- 3. Robot musí chránit sám sebe před zničením, kromě případů, kdy tato ochrana je v rozporu s prvním nebo druhým zákonem.“[7]*

V průběhu druhé poloviny 20. století došlo k důležitým událostem, které ovlivnily vývoj autonomní robotiky. V 50. letech vznikly první elektrické mobilní roboty jménem Elmer a Elsie (Obrázek 1-2), u kterých neurovědec W. Grey Walter zkoumal možnost řídit složité chování a volbu pomocí minimálního počtu mozkových buněk s použitím pouze dvou smyslů, zraku a hmatu. Tyto roboty byly vybaveny světelnými a dotykovými senzory, které jim umožňovaly navigovat v prostředí, reagovat na světelné podněty a vnímat překážky dotykem.



Obrázek 1-2 - Popis robota Elsie bez ochranného štítu[8]

V 60. letech byl vyvinut robot Shakey, který již využíval umělou inteligenci a zahrnoval dotykové senzory i kameru k plánování svých pohybů. Tento krok směrem k interakci s okolím a vnímání prostředí byl klíčový, aby otevřel cestu pro další vývoj autonomních robotů, kteří by

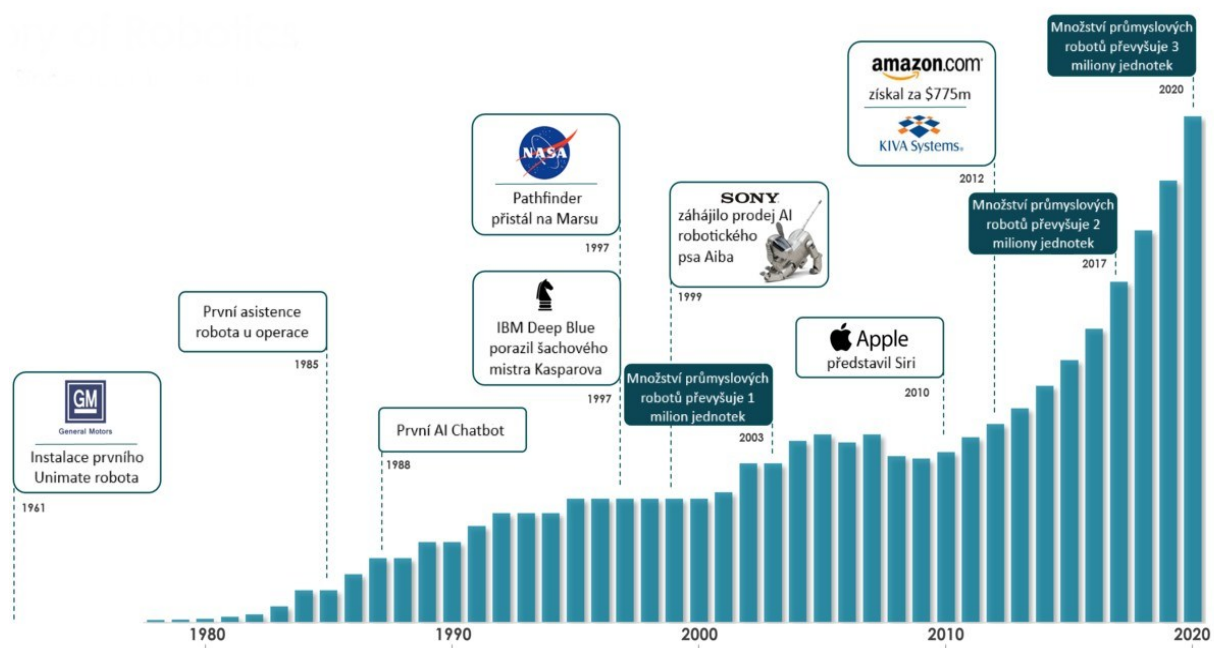
mohli být využiti v různých oblastech, od průmyslové výroby po průzkum vesmíru a pomoc ve zdravotnictví. [9; 10]

Tato podkapitola o historii robotiky poskytuje základní přehled o tom, jak se robotika vyvinula od svých počátků a jakým způsobem se formovala. Další se bude zabývat průmyslovou robotikou a její vývojem.

1.2 Vývoj robotiky v průmyslu

První stacionární průmyslový robot jménem Unimate byl navrhnout americkým vynálezcem Georgem Devolem, a v roce 1961 byl nasazen ve společnosti General Motors. Jednalo se o programovatelné, elektricky ovládané a hydraulicky poháněné rameno, které sloužilo k přesouvání těžkých odlitků a bodovému svařování na montážní lince. Po tomto průlomovém okamžiku začal rychlý a nezadržitelný nárůst použití průmyslových robotů po celém světě. Robotika se stala zásadní součástí moderního průmyslu, který byl schopen díky robotům dosáhnout vyšší efektivity, kvality a bezpečnosti výroby.[4]

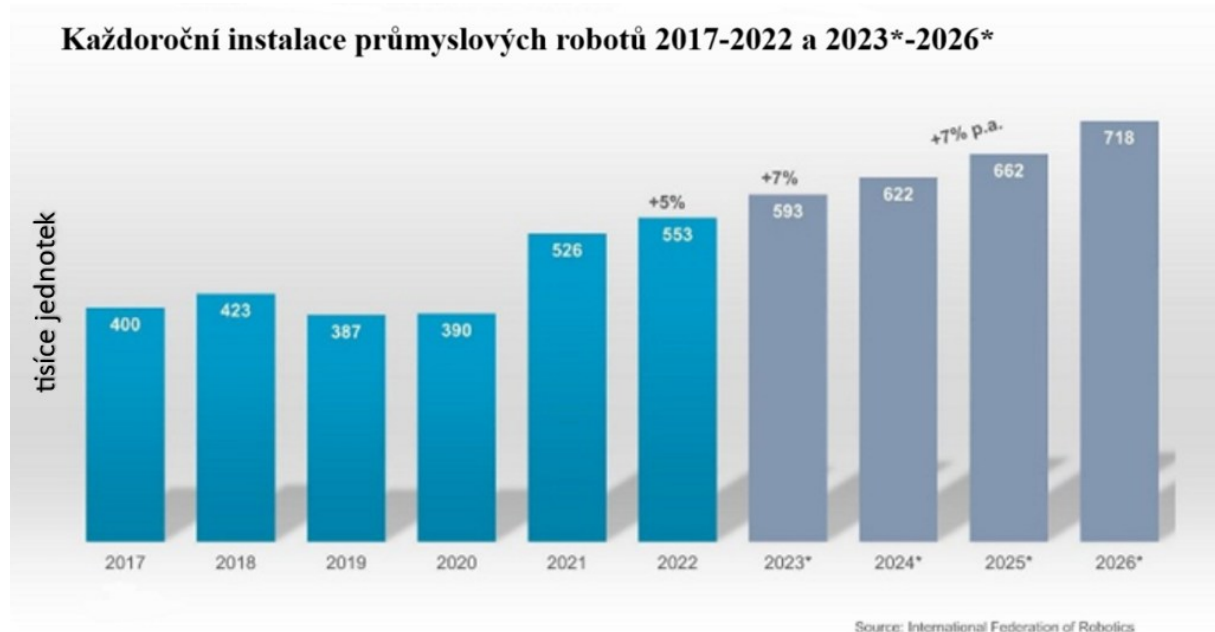
Graf na obrázku 1-3 znázorňuje počet instalovaných průmyslových robotů do výroby na celém světě od prvního nasazení robota Unimate v roce 1961 až po rok 2020. V období od roku 2010 dochází k exponenciálnímu nárůstu počtu robotů, přičemž podle Mezinárodní federace robotiky (IFR) v roce 2020 dosáhl počet 3,035 milionů jednotek.



Obrázek 1-3 - Časová osa implementace průmyslových robotů[11]

Podle zprávy World Robotics bylo v roce 2021 zaznamenáno nainstalování 526 000 průmyslových robotů ve firmách po celém světě. V roce 2022 se tento počet ještě zvýšil na 553,052 jednotek. To představuje meziroční růst o 5 % (Obrázek 1-4). Regionálně byla 73 % těchto nově nasazených robotů umístěna v Asii, zatímco Evropa přispěla 15 % a Amerika 10 %. Tyto čísla jasně ukazují, že průmyslová robotika je v současné době na vzestupu a tvoří důležitou součást moderní výroby a automatizace. S ohledem na nadcházející období lze předpokládat, že tento trend bude pokračovat i do budoucna, a to se stále rostoucím tempem s meziročním růstem 7 % a vyšším. V roce 2024 by mělo dojít k překonání hranice 600 000

instalovaných robotů ročně. Roboty se stávají nezbytnými nástroji v průmyslu a jejich využití bude pravděpodobně stále více rozšiřováno do dalších odvětví a regionů.[12]



Obrázek 1-4 - Každoroční instalace průmyslových robotů 2017-2022 a 2023*-2026*[12]

V další podkapitole bude přibliženo využití robotů v různých odvětvích a konkrétních aplikacích.

1.3 Využití robotů

Roboty mají širokou škálu využití ve více odvětvích. Zde jsou některé z hlavních oblastí, kde se roboty uplatňují:[13]

- **Průmyslové roboty:** Jsou často používáni v továrnách a výrobních zařízeních. Jejich popisem a rozdělením se bude věnováno v dalších částech práce.
- **Lékařská robotika:** Roboty jsou využívány ve zdravotnictví pro chirurgické zákroky, diagnostiku a léčbu. Chirurgické roboty umožňují lékařům provádět přesné a minimálně invazivní operace (Obrázek 1-5).



Obrázek 1-5 - Chirurgický robot Da Vinci Xi[14]

- **Služební robotika:** Roboty jsou nasazovány veřejnosti pro různé úkoly, jako je úklid domácnosti, doručování zásilek, obsluha restaurací a hotelů a další služby (Obrázek 1-6).



Obrázek 1-6 - Autonomní doručovací robot X1 PRO[15]

- **Vojenská robotika:** Vojenské roboty jsou využívány pro průzkum, záchranu, odstraňování nástražných min, bezpilotní bojové letouny a další aplikace v oblasti obrany (Obrázek 1-7).



Obrázek 1-7 - Taktický robot TALON 5[16]

- **Výzkum a průzkum:** Roboty jsou používány pro průzkum těžko dostupných nebo nebezpečných prostředí, včetně hloubkových oceánských výprav, průzkumu vesmíru a výzkumu katastrofických scénářů (Obrázek 1-8).



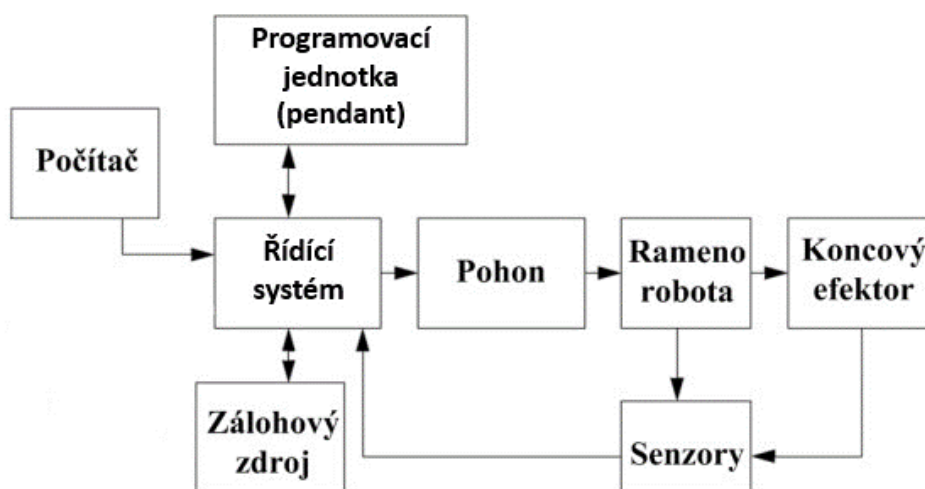
Obrázek 1-8 - Robotická ponorka AQUANAUT[17]

Zatímco byla představena široká škála využití robotů, průmyslové roboty představují specifickou kategorii, která je klíčová pro automatizaci v průmyslu. Jejich schopnost provádět různorodé úkoly v průmyslovém prostředí zajišťuje nejen efektivitu, ale i flexibilitu v rámci výrobních procesů. V následující části budou popsány konkrétní aspekty průmyslových robotů, včetně jejich pohybového systému, koncových efektorů, pohonu a řídicího systému.

1.4 Průmyslové roboty a jejich systémy

Průmyslový robot je programovatelné zařízení s robotickou rukou umístěné stacionárně či na pojezdu, které je navrženo pro autonomní provedení fyzických úkolů v průmyslovém prostředí. Tato zařízení jsou schopna vykonávat různorodé operace, jako jsou manipulace s objekty, svařování, malování, montáž a další činnosti výrobního procesu. Jejich programovatelnost a flexibilita v nejméně třech pohybových osách umožňuje rychlé přizpůsobení se novým úkolům a změnám výrobního procesu.[18]

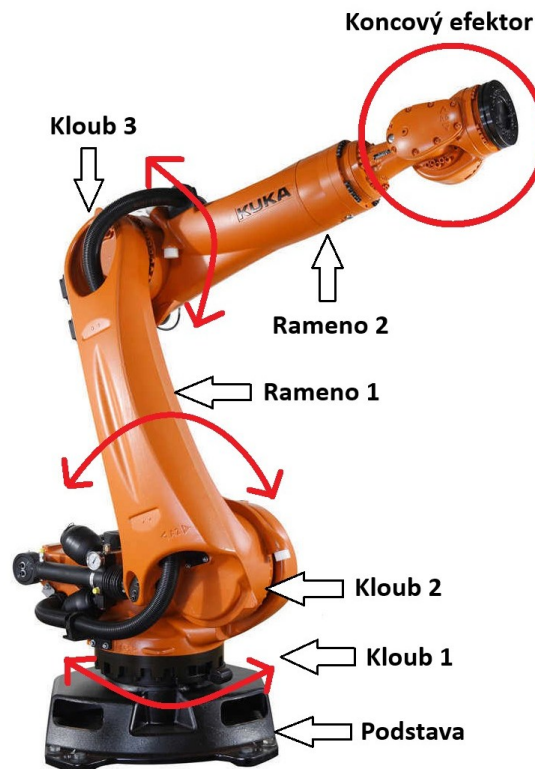
Roboty nabízejí širokou škálu schopností a konfigurací v rámci svých dostupných možností. Struktura robotického systému zahrnuje několik klíčových podsystémů. Mezi ně patří pohybový systém, koncový efektor, pohon a řídicí systém s programovací jednotkou (Obrázek 1-9).[4]



Obrázek 1-9 - Blokové schéma řízení průmyslového robota [19]

Pohybový systém

Obsahuje sérii mechanických spojů, ramen a kloubů navržených pro pohyb v různých směrech a přenosu sil s cílem ovládat koncový efektor pro zacházení s objekty. (Obrázek 1-10)

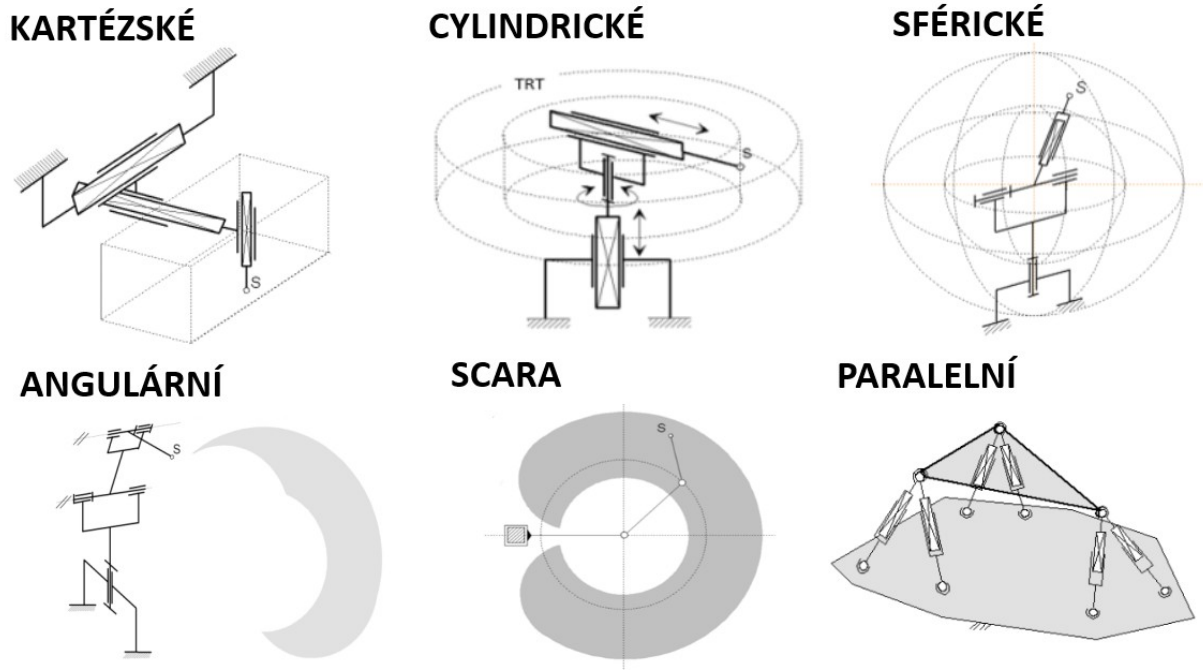


Obrázek 1-10 - Popis průmyslového robota Kuka[20]

Kinematika manipulačních mechanismů je odvozena z otevřených prostorových kinematických řetězců a vázána prostorovými kinematickými dvojicemi. Pohyblivost kloubů a jejich vzájemný pohyb jsou definovány počtem stupňů volnosti i . Počet těchto stupňů volnosti odpovídá počtu nezávislých posuvů a rotací, které mohou členy kinematické dvojice vzájemně vykonávat. Průmyslové roboty obvykle mají $4^\circ - 6^\circ$ (v případě pojezdu 7°) volnosti. Tato struktura je důležitá pro definování flexibility a schopnosti pohybu manipulátoru v pracovním prostoru.

Podle struktur kinematických řetězců pro pohybový systém dělíme průmyslové roboty na (Obrázek 1-11):[21]

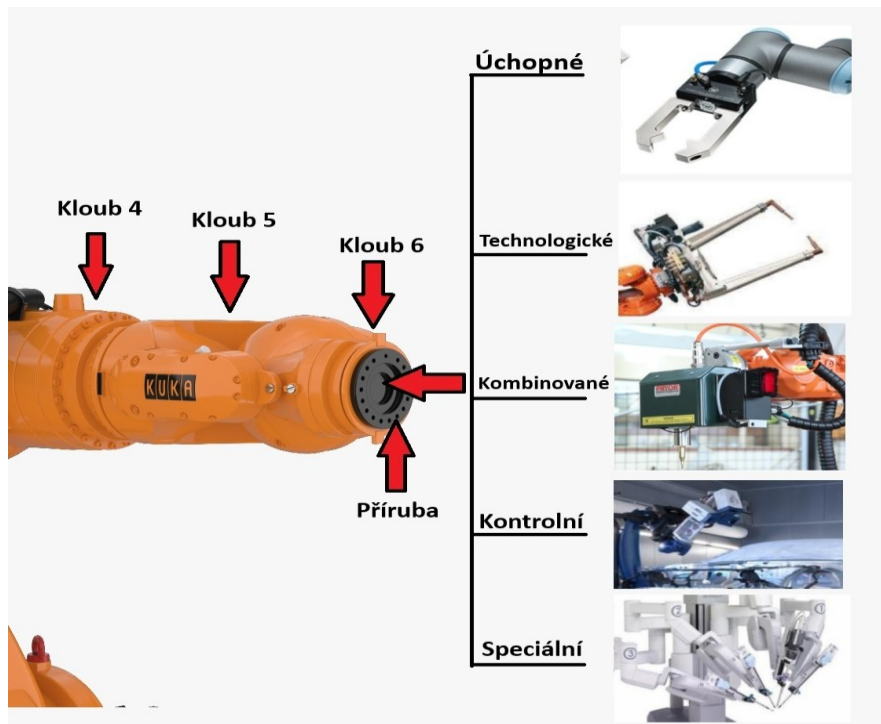
- **Kartézské (TTT)** – 3 na sebe kolmé posuvné pohyby, pracovním prostorem je hranol, vysoká tuhost a přesnost polohování.
- **Cylindrické (RTT)** – 1 posuvný vertikální a horizontální a 1 rotační pohyb v ose rotace, pracovním prostorem je válcový prstenec.
- **Sférické (RRT)** – 2 na sebe osami kolmé rotační a 1 posuvný pohyb, pracovním prostorem je kulový vrchlík, prvně použito u robota Unimate.
- **Angulární (RRR)** – 3 rotační osy (rotace kolem svislé osy a zbývající osy jsou vodorovné a rovnoběžné), pracovní prostor má širokou variabilitu (většinou anuloid).
- **SCARA** – 2 rotační a 1 posuvný pohyb (pohybové osy jsou svislé a vzájemně rovnoběžné), velká rychlost a přesné polohování.
- **Paralelní** – Vzniknou spojením výstupních prvků systému s rámem prostřednictvím několika tradičních sériových řetězců.



Obrázek 1-11 - Dělení průmyslových robotů dle geometrie pracovního prostoru[21]

Koncový efektor

Koncový efektor je přídavné zařízení na konci robotického ramene uchycené za pomoci příruby (Obrázek 1-12). Hlavice slouží k interakci s okolím, uchycení manipulační součásti a změně její polohy a orientace v prostoru. Koncový efektor může mít různé formy a být navržen specificky pro konkrétní aplikace.



Obrázek 1-12 - Konec robotického ramene s rozdělením efektorů[21; 22]

Rozdělení hlavic dle aplikačního nasazení (Obrázek 1-12):[21]

- **Úchopné** – Používají se k uchopování a držení objektů. Mají různé konstrukce a mohou být navrženy pro specifické tvary nebo velikosti objektů. Využívají silové či tvarové principy uchopení a držení objektu.
- **Technologické** – Hlavice tvořené přímo nástrojem, které se používají k provedení konkrétních technologických úkolů v průběhu výrobního procesu. Nejčastějšími druhy operací jsou dělení, obrábění a spojování materiálu, nanášení ochranných látek a speciální aplikace v různých odvětvích.
- **Kombinované** – Tyto hlavice kombinují funkci úchopné hlavice s jiným specializovaným příslušenstvím podle konkrétní potřeby. Umožňují provádět různorodé úkoly bez nutnosti výměny celého efektoru.
- **Kontrolní** – Hlavice disponují senzory pro identifikaci kvalitativních parametrů součástí, tj. měřicími zařízeními. Často se využívá 2D a 3D kamer a 3D scannerů.
- **Speciální** – Jiné než výše zmíněné, například napodobující lidskou ruku.

Pohon pohybových jednotek

Průmyslové roboty využívají k pohonu transformaci primární energie na mechanický pohyb. Části pohonu jsou motor, blok pro ovládání energie do motoru a spojovací blok. Přenos pohybu z motoru na pohyblivou část pohybové jednotky může probíhat přímo nebo přes transformační blok. Konstrukce pohonů využívá mechanické, magnetické nebo elektrické převody, přičemž ozubené převody s novými typy harmonických převodovek dominují. Významným hlediskem pro funkčnost je plynulý rozběh, vysoká přesnost polohování, dostatečná polohová tuhost, vhodné prostorové uspořádání, minimální hmotnost a rozměry. Tyto požadavky slouží k dosažení klidného a přesného průběhu manipulační činnosti s ohledem na setrvačné síly a rychlost pohybu. Plynulý bezrázový chod a vysoká přesnost polohování jsou důležité pro bezpečnost, úchopnou sílu a životnost zařízení.

Pohon pohybových jednotek, jako je kloub, rameno a koncový efektor, může být uspořádán buď odděleně, kde každá jednotka má vlastní pohon s nezávislým motorem nebo společně, kde pohony vycházejí z jednoho motoru.

Existují tři modifikace společného pohonu:

- **Paralelní uspořádání:** Pohon dvou nebo více pohybových jednotek vychází z jednoho společného motoru. Rozvětvení za motorem je řešeno transformačním blokem nebo prostřednictvím ovládacích bloků.
- **Sériové uspořádání:** Pohyb každé jednotky je odvozen od pohybu předchozí jednotky.
- **Semiparalelní uspořádání:** Kombinace společného sériového a paralelního pohonu.

Tři hlavní zdroje pohonu jsou:[21; 23]

- **Elektrický pohon:** K pohonu využívá asynchronních, stejnosměrných, bezkartáčových, krokových a lineárních motorů a patří v poslední době k nejpoužívanějším. Má nižší výkon než ostatní typy, ale vyniká v přesnosti a opakovatelnosti. Výhodou jsou nízké pořizovací, provozní a udržovací náklady, nízká hlučnost a menší nároky na chlazení a prostor.
- **Hydraulický pohon:** Je mechanicky jednodušší než elektrický pohon a využívá se u robotů pro těžkou práci, jelikož disponuje velkou silou a rychlostí. Pracovním médiem je minerální olej a pro fungování je potřeba samostatný energetický blok. Vynikají díky své velké tuhosti, plynulému chodu a schopnosti dosáhnout i nízkých rychlostí pohybů bez použití převodů.

- **Pneumatický pohon:** Využívá stlačeného vzduchu pro potřeby malých až středních sil a rychlých pohybů s vysokou frekvencí. Jeho výhodou je jednoduchá konstrukce a dobrá dostupnost stlačeného vzduchu, spolu s pozitivními vlastnostmi, jako je efektivní skladování, minimální vliv na životní prostředí a bezpečnost. Pro bezpečnost zůstávají motory při přetížení nehybné po neomezenou dobu.

Řídicí systém

Řídicí systém řídí a koordinuje pohyby a operace robota. Tento systém zahrnuje hardware a software, který pracuje společně k tomu, aby robot vykonával specifické úkoly. Mozek robota je mikroprocesor, který zpracovává informace z programovací jednotky a senzorů a rozhoduje o tom, jaké akce má robot podniknout. Senzory jsou zařízení, které sbírají data z okolního prostředí či stavu robota. Programovací jednotka neboli pendant slouží k naprogramování aplikace pro pohyb robota. [19]

Kolaborativní roboty řady CRX společnosti Fanuc nabízí jednoduché programování skrze tablet (Obrázek 1-13). Tablet Teach Pendant disponuje 10palcovou obrazovkou a aplikací s intuitivním drag & drop programováním, která zjednodušuje proces vytváření programů a umožňuje rychlou a efektivní konfiguraci robota. Stačí jednoduše přetáhnout programovací pokyny na časovou osu společně s manuálním nastavením ramene do požadované polohy. V průběhu programování je možné pozorovat pohyb ramene ve 3D pohledu. Držák tabletu obsahuje tlačítko nouzového zastavení a třípolohový aktivační spínač.[24]



Obrázek 1-13 - Tablet Teach Pendant s popisem[24]

1.5 Rozdělení průmyslových robotů

Mezi přední společnosti v automatizaci a výrobě průmyslových robotů patří ABB, KUKA, Fanuc a Yaskawa. Každá z těchto firem přináší do průmyslového sektoru inovativní a výkonné robotické systémy. Tato podkapitola se zaměřuje na katalog průmyslových robotů společnosti ABB a jeho hlavní rozdělení: [25]

Kloubové roboty

Šestiosé kloubové roboty je možné využít například pro manipulaci s materiálem, montáž, obsluhu strojů, bodové a obloukové svařování, řezání, testování, kontrolu, dávkování, broušení a leštění. Na výběr je z 30 vysoce přesných modelů s nosností od 1,5 kg do 800 kg a dosahem od 0,37 m do 4,2 m. Některé modely excelují v odolnosti a spolehlivosti, čímž jsou ideální pro

náročné průmyslové aplikace. Kde je kladen důraz na hygienu a čistotu, jsou k dispozici roboty navržené s ohledem na hygienické normy. Štíhlý design a nízká hmotnost pak zase představují skvělou volbu pro ty, kteří chtějí snížit energetickou náročnost svých výrobních procesů. Největším a nejvýkonnějším robotem v nabídce je IRB 8700 (Obrázek 1-14).



Obrázek 1-14 - Kloubový robot ABB IRB 8700[25]

Delta roboty

Delta roboty jsou speciální typ průmyslových robotů navrženy pro rychlé a přesné operace v oblastech jako sbírání, balení či manipulace s objekty (Obrázek 1-15). Charakterizuje je paralelní kinematika, kde tři a více lineárních pohonů pohání pohyblivý efektor. Na výběr je z 3 modelů a liší se především nosností (1,5 - 15 kg) a dosahem (1,13 - 1,6 m).



Obrázek 1-15 – Delta robot IRB 390 [25]

SCARA roboty

Scara roboty jsou navrženy pro aplikace, které vyžadují rychlé, opakované a přesné horizontální pohyby i v omezeném prostoru (Obrázek 1-16). Využívají se především při

manipulaci s materiálem, kde je potřeba jeho zakládání, odebírání či přemísťování. Na výběr je z 3 modelů a liší se především montáží, nosností (3 - 12 kg) a dosahem (0,35 - 1,05 m).



Obrázek 1-16 - SCARA robot IRB 910SC[25]

Paletizační roboty

Paletizační roboty slouží k automatické manipulaci a uspořádání předmětů na palety (Obrázek 1-17). Mohou pracovat s různými typy materiálů, od krabic až po těžší nebo nepravidelně tvarované předměty. Na výběr je z 3 modelů a liší se především nosností (110 - 450 kg) a dosahem (2,4 - 3,18 m).



Obrázek 1-17 - Paletizační robot IRB 460 [25]

Lakovací roboty

Pro zvýšení produktivity a úrovně kvality provozu lakovny jsou v nabídce 6, 7 osé a portálové lakovací roboty (Obrázek 1-18). V automobilovém průmyslu je možné využít robotický otevírač dveří, který usnadní vnik ramene do karoserie automobilu. Na výběr je ze 4 modelů a liší se především rychlostí lakování, dosahem (1,20 - 2,97 m), nosností (7 - 13 kg) a způsobem montáže.



Obrázek 1-18 - Lakovací robot IRB 5510[25]

Kolaborativní roboty

Kolaborativní robot, zkráceně cobot, je druh průmyslového robota navrženého tak, aby mohl bezpečně a efektivně spolupracovat s lidmi ve sdíleném pracovním prostoru bez fyzických bariér nebo oplocení. Dokážou pracovat nepřetržitě a opakovat monotónní úkony, čímž uvolňují lidský potenciál pro kreativnější úkoly a snižují potřebu lidské práce ve stereotypních činnostech. U spolupráce robota s člověkem je nutné dbát důraz na zvýšenou bezpečnost, aby nedošlo k úrazu. Kolaborativní roboty jsou vybaveny senzory, které při střetu s objektem přeruší aktuální činnost robota, a dojde k jeho zastavení. Vyznačují se jednoduchým programováním, kdy je možné rameno manuálně provést trasou a řídicí systém během toho snímá průběh dráhy a souřadnice si ukládá do paměti.

Na výběr je z 5 modelů a liší se především nosností (0,5 – 12 kg), dosahem (0,475 – 1,52 m) a počty os (6,7 nebo 14). YuMi je jediný robot se dvěma pažemi z katalogu ABB (Obrázek 1-19).



Obrázek 1-19 - Kolaborativní robot YuMi IRB 14000 [25]

V následující kapitole se zaměříme na další inovativní technologii, která mění náš pohled na pracovní prostředí, a to rozšířenou realitu. Přiblížíme si její fungování, historii, vývoj a využití v průmyslových aplikacích.

2 Rozšířená realita

V současném digitálním věku se technologie stávají stále neodmyslitelnější součástí našeho každodenního života. Jednou z inovativních oblastí, která výrazně mění způsob, jakým vnímáme a interagujeme se světem kolem sebe, je rozšířená realita. Rozšířená realita je technologie, která v reálném čase přidává virtuální informace a objekty do fyzického prostředí, a vytváří tím obohacený a rozšířený zážitek (Obrázek 2-1). [26]



Obrázek 2-1 - Využití rozšířené reality v praxi[27]

Definice rozšířené reality dle společnosti Microsoft zní následovně: „Rozšířená realita je vylepšená, interaktivní verze reálného prostředí, která se vytváří pomocí digitálních vizuálních prvků, zvuků a dalších smyslových podnětů prostřednictvím holografické technologie. AR zahrnuje tři vlastnosti: kombinaci digitálního a fyzického světa, interakce prováděné v reálném čase a přesnou 3D identifikaci virtuálních a skutečných objektů.“[27]

S AR se v dnešní době nejběžněji můžeme setkat skrze aplikace v mobilních telefonech. Ať už se jedná o hry, kde můžeme přes displej lovit v reálném světě nejrůznější příšery, nebo aplikace od společností prodávající nábytek, které umožňují vyzkoušení výrobků v interiéru ještě před jeho zakoupením. Rozšířená realita dále nachází široké uplatnění v různých odvětvích. V oblasti obchodu a marketingu může poskytovat interaktivní reklamy a umožňovat zákazníkům zkusit produkty virtuálně před nákupem. Pro vzdělávání se využívá k obohacení učebních materiálů a simulacím pro lepší porozumění studentů. Ve zdravotnictví může sloužit k chirurgické navigaci a tréninku pacientů a v kultuře přináší interaktivní prvky do uměleckých děl. Rozšířená realita má mnohostranné využití a přináší inovativní a interaktivní zážitky do různých sfér našeho každodenního života.

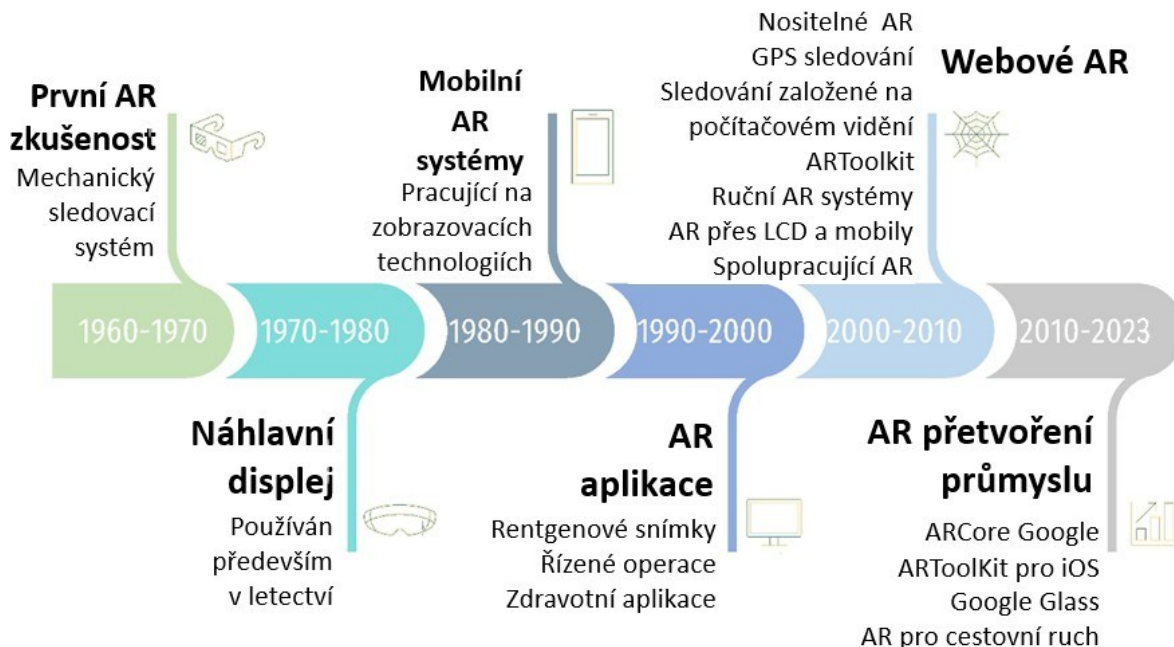
Vzhledem k tomu, že se v této práci zaměřujeme na rozšířenou realitu, je důležité rozlišit ji od další inovativní technologie známé jako virtuální realita. Rozšířená realita, jak již bylo zmíněno, spojuje digitální a fyzický svět. Přidává virtuální prvky do reálného prostředí a umožňuje uživatelům interagovat s oběma světy současně. Na rozdíl od toho virtuální realita vytváří úplně odlišný prostor, do kterého uživatele zcela pohltí, odstavuje je od okolní reality a přenáší je do plně virtuálního prostoru.[26]

2.1 Historie rozšířené reality

Vznik rozšířené reality sahá až do 50. let, kdy Morton Heilig vytvořil Sensorama. Přístroj, který měl působit na všechny divákovy smysly a tím ho vtáhnout do dění. Vůbec první HMD, (Head Mounted Display) neboli náhlavní displej, vynalezl v roce 1968 vědec Ivan Sutherland. Jednalo se o revoluční zařízení využívající rozšířenou realitu, které bylo později inovováno a využíváno především v letectví. Po Sutherlandově průlomu se technologie rozšířené reality začala rychle vyvíjet. V roce 1975 Myron Krueger vytvořil Videoplace, místnost umožňující uživatelům interakci s virtuálními objekty. Následně, ve stejném roce, Tom Caudell a David Mizell z Boeing při práci na leteckých montážích vytvořili termín "Augmented Reality".

2.2 Vývoj rozšířené reality

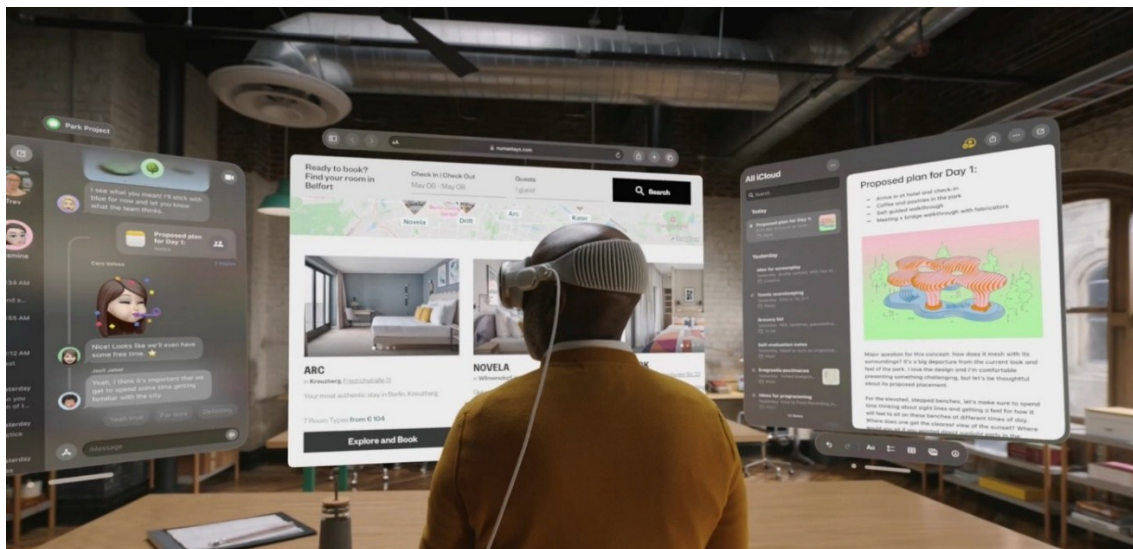
S postupem času se AR stala klíčovou technologií v různých odvětvích, včetně letectví, lékařství a mobilních aplikací. V roce 2000 byla vyvinuta první venkovní mobilní AR hra, ARQuake, a během 5 let byly vyvinuty kamery schopné analyzovat fyzické prostředí v reálném čase, to přispělo k integraci virtuálních objektů s realitou v AR systémech. Průběh vývoje jednotlivých období AR od počátku do roku 2023 je možné pozorovat na obrázku 2-2.[26]



Obrázek 2-2 - Pokrok v rozšířené realitě [28]

V současné době patří mezi nejnovější výrobky v oblasti smíšené reality, kombinující AR a VR, brýle Vision Pro od společnosti Apple (Obrázek 2-3). Apple Vision Pro představuje revoluční prostorový počítač, který integruje digitální obsah s fyzickým světem, umožňuje interakci pomocí očí, rukou a hlasu a nabízí trojrozměrné uživatelské rozhraní s dvěma ultra-vysokorozlišovacími micro-OLED displeji, disponujícími dohromady 23 miliony pixely. Pro lepší představu, 4K televize má 8,3 milionu pixelů, což je skoro třikrát méně. S ohledem na velikost displejů poštovní známky poskytuje Vision Pro opravdu vynikající vizuální zážitek. Pro snímání obrazu, pohybu a zvuku je využíváno dvanáct kamer, pět pohybových senzorů a šest mikrofónů. Pro sledování očí jsou k dispozici sada LED a infračervených kamer, které promítají neviditelné světelné vzory do každého oka, umožňující používat aplikace pouhým pohledem. Zajímavou vlastností je také technologie EyeSight, která promítá vaše oči na vnější

displej, informující okolí, zda právě používáte aplikace ve fyzickém světě nebo jste plně ponořeni do virtuální reality.[29]



Obrázek 2-3 - Prostředí Apple Vision Pro[29]

Aplikace pro Vision Pro jsou stále ve stádiu raného vývoje a Apple poskytuje developerům veškerou podporu. Na úvodní prezentaci headsetu byly představeny 4 aplikace, které prezentují možnosti využití:

- **Complete HeartX** obsahuje hyperrealistický 3D model srdce a obohacuje lékařské vzdělání propojením anatomie a fyziologie s reálnými diagnostickými a léčebnými scénáři jako je fibrilace komor.
- **Djay** nabízí uživatelům autentický zážitek z mixování hudby i bez skutečného mixážního pultu. Lze jednoduše přidávat efekty pohledem nebo gestem ruky a tvořit vlastní hudbu v reálném čase.
- **JigSpace** a **Stages** slouží k inovativnímu představování nápadů a produktů. JigSpace umožňuje efektivní spolupráci týmů v oblasti marketingu, prodeje a produkce pomocí CAD souborů. Stages využívá Apple Vision Pro k přenášení interaktivního 3D obsahu do reálného světa. Umožňuje všem zúčastněným stranám pracovat současně a spolupracovat na rozhodování o návrzích a dalších krocích (Obrázek 2-4). Tato inovace otevírá nové možnosti spolupráce mezi odděleními a lokalitami. [30]



Obrázek 2-4 - Výrobní linka přes aplikaci Stages v Apple Vision Pro[30]

2.3 Zařízení pro rozšířenou realitu

Pro zobrazení kombinace reálného světa s virtuální vrstvou grafických informací či objektů využívá AR počítače a zobrazovacího, vstupního a sledovacího zařízení: [26]

Zobrazovací zařízení

Zobrazovací zařízení lze rozdělit do tří hlavních typů: náhlavní displej (HDM), ruční displeje a prostorová rozšířená realita (SAR). HDM je zařízení s displejem, které je nošeno na hlavě a promítá digitální obraz do zorného pole uživatele. Obraz může být promítán do jednoho či obou očí. Existují varianty nazývané video-see-through, kde na displeji dochází k promítání záznamu z kamer s již upravenou realitou, a optical-see-through, kde se využívá nakloněné průhledné čočky pro pohled na reálný svět a odrazu grafických informací do oka. Ruční displeje využívají video-see-through k překrytí grafiky a reálného prostředí nasnímaného kamerou. Patří mezi ně například mobilní telefony a tablety. SAR využívá video-projektorů, hologramů a dalších technologií k přímému zobrazování grafických informací na fyzické objekty.

Vstupní zařízení

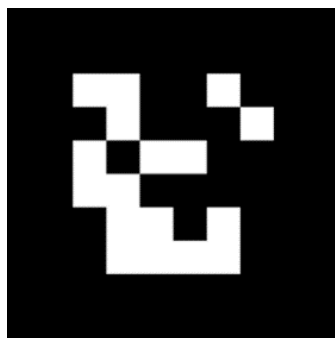
Vstupní zařízení slouží uživateli pro interakci a manipulaci s virtuálním prostředím a prvky. K nejrozšířenějším příslušenstvím patří pohybové ovladače, které umožňují uživatelům sledovat a reagovat na pohyby ve virtuálním světě. Haptické rukavice dodávají uživatelům hmatový zážitek při interakci s virtuálními objekty. Zařízení pro sledování očí umožňují přesné zaměřování a ovládání pomocí pohybu očí. Dokonce i chytré telefony mohou sloužit jako vstupní zařízení, když jsou využívány ve spojení s aplikacemi a senzory pro rozšířenou realitu.

Sledovací zařízení

Sledovací zařízení umožňuje systému vidět a monitorovat pohyby uživatele nebo objektů v reálném světě. Jedná se především o kamery, optické senzory, GPS, akcelerometry, pevné kompasu a bezdrátové senzory.

Pro orientaci v prostoru se využívají dva hlavní způsoby trackování, markerless a marker-based techniky. Markerless techniky umožňují sledování pohybu bez použití speciálních vizuálních značek. Patří sem sensor-based metody, jako je magnetické nebo inerciální sledování, vision-based metody, které využívají kamery a senzory, a 3D structure tracking, technologie založená například na zařízeních jako Microsoft Kinect.

Na rozdíl od toho marker-based techniky pracují s umělými vizuálními značkami, jako jsou QR kódy a markery nebo 3D modely objektů (Obrázek 2-5). Fiducial tracking využívá černobílé značky, které jsou snímány kamerou k identifikaci referenčního bodu v reálném světě. Na základě této identifikace lze pak do prostoru přidat virtuální prvky. Model-based tracking pracuje s 3D modely objektů a sleduje jejich pohyb v reálném čase.



Obrázek 2-5 - AR Marker[48]

Některá zařízení využívají hybridního sledování, kde kombinují různé techniky z obou kategorií. Tímto způsobem lze dosáhnout vyšší přesnosti a řešit případné nedostatky jednotlivých sledovacích metod. Celkově tyto techniky umožňují systému vidět a interagovat s virtuálními prvky v reálném světě pro vytváření poutavých a interaktivních zážitků v rozšířené realitě.[28]

Počítač

Pro zpracování informací z kamer a senzorů, zajištění plynulého chodu aplikací rozšířené reality, vyžadují AR systémy výkonný procesor a značné množství operační paměti.[26]

2.4 Využití AR v průmyslu

Rozšířená realita hraje stále významnější roli v průmyslových odvětvích a přináší inovace a efektivitu. V této kapitole se zaměřujeme na využití AR v průmyslových procesech a na to, jak tato technologie přispívá ke zlepšení produktivity, školení a údržby.

Návodové systémy pro nové pracovníky

Implementace AR brýlí umožňuje pracovníkům získávat návodové informace přímo ve svém zorném poli. Tímto způsobem se nejen výrazně urychluje a usnadňuje plnění úkolů, ale zároveň dochází k minimalizaci možnosti chyb. Vytváření realistických simulací pracovních situací pomocí AR přispívá k lepšímu a bezpečnějšímu školení pracovníků.[31]



Obrázek 2-6 - LightGuide's Projected AR[32]

Systém od společnosti LightGuide využívá kombinaci 3D vizuálních kamer a projekčních vizuálních nápověd (Obrázek 2-6). Návod pro sestavení dílu je promítán na pracovní prostor v jednotlivých krocích, a pracovníkovi pomáhá světelným zvýrazněním k nalezení správného dílu z jednotlivých krabiček, a následně jeho umístění na odpovídající místo. Kamery snímají přesnou polohu součástí, a pokud pracovník umístí něco nesprávně, nedovolí mu pokračovat a celý krok musí provést znovu. Celé prostředí je interaktivní a pro potvrzení tlačítek (další, zpět či zrušení) stačí na dané tlačítko kliknout celou dlaní. Po dokončení operace je zobrazen celkový čas složení, který poskytuje pracovníkovi a zaměstnavateli přehled o rychlosti montáže. Tento systém poskytuje možnost školení více pracovníků bez nutnosti dozoru dalších lidí a ve výrobě zajišťuje nulové chyby.[32]

Interaktivní návody pro opravy a údržbu

Rozšířená realita umožňuje poskytovat technikům interaktivní asistenci při opravách a údržbě zařízení. Tato forma podpory nejenže snižuje čas potřebný k provedení oprav, ale zároveň minimalizuje riziko chyb, a tím snižuje náklady. Technikům jsou poskytovány vizuální a zvukové návody v reálném čase. Zvyšuje to jejich schopnost správně identifikovat nástroje a součásti nezbytné k úspěšnému dokončení úkolů.[33]

Automobilka BMW používá v mnichovské továrně softwaru Hololight Space s HDM HoloLens 2 od Microsoftu k vizualizaci vozidla a jeho součástí s pomocí CAD modelů (Obrázek 2-7). Toto umožňuje technikům pochopit dostupnost k jednotlivým dílům, překrývat skutečné geometrie s možností instalace a viditelnosti montážních bodů ve složitých systémech. Tato technologie umožňuje flexibilní a nákladově efektivní posouzení různých variant koncepcí a montážních procesů pro nadcházející sériová vozidla.[34]



Obrázek 2-7 - Využití AR automobilkou BMW[34]

Porsche využívá HDM ke vzdálené asistenci při řešení obtížných oprav, kdy je potřeba odborníka, který již nemusí k pomoci nadále cestovat. Tým technické podpory má schopnost zaznamenávat snímky obrazovky a zasílat technické zprávy a pokyny na projekční plochu brýlí, když technik pracuje na vozidle. Tato technologie dokáže zkrátit dobu řešení servisního problému až o 40 %.[35]

Návrh, sledování a optimalizace výrobních procesů

Rozšířená realita se může využívat k návrhu, monitorování výrobních linek a identifikaci potenciálních problémů (Obrázek 2-4). Zavedení AR do průmyslových procesů umožňuje rychle reagovat na změny a odchylky v reálném čase. Pracovníci mohou využívat AR k vizualizaci a analýze konfigurací a možností v reálném prostředí, což přispívá k lepšímu rozhodování a efektivitě výrobních operací.[33; 36]

Sledování stavu zařízení

Rozšířená realita poskytuje možnost sledovat stav strojů v reálném čase a rychle identifikovat potenciální problémy. Tímto způsobem lze provádět preventivní údržbu a minimalizovat riziko neplánovaných výpadků. Technici jsou schopni prostřednictvím AR získat okamžitý přehled například o výkonu zařízení a provádět potřebné kroky k udržení optimálního provozu.[33]

Společnost AV ENGINEERING vyvinula pro svůj projekt na testování životnosti brzdových válečků pro automobily software na vzdálené sledování a řízení zkušebního zařízení (Obrázek 2-8). Tento moderní nástroj umožňuje pracovníkům v reálném čase sledovat na tabletu či mobilním zařízení okamžité hodnoty sledovaných veličin a celkový přehled zařízení s jeho 3D modelem pro jeho obsluhu a servis. Důraz je kladen na uživatelskou přívětivost, a proto je vytvořen animovaný postup pro jednoduchou výměnu testovaných válečků. Tato interaktivní a intuitivní funkcionalita usnadňuje pracovníkům provádět údržbu a opravy. Tímto maximalizuje efektivitu a minimalizuje časové ztráty.[37]



Obrázek 2-8 - Sledování a řízení testovacího zařízení od AV ENGINEERING[37]

Integrace AR do průmyslových procesů přináší mnoho výhod a otevírá cestu k modernizaci a efektivitě v rámci různých odvětví. S vývojem a zdokonalováním AR technologií bude pravděpodobně docházet k ještě širšímu využití v průmyslovém prostředí. V další podkapitole se podíváme na využití rozšířené reality v robotice.

2.5 Využití AR v robotice

Rozšířená realita přináší v oblasti robotické automatizace zvýšení produktivity a zlepšení bezpečnosti pracovníků prostřednictvím vylepšených digitálních simulací a optimalizace interakce mezi člověkem a roboty. Tato inovativní technologie umožňuje výrobcům daleko lepší vizualizaci potenciálních řešení při implementaci robotů a vytváření reálných prostředí s efektivní interakcí mezi lidskými pracovníky a roboty.

V tomto kontextu poskytuje výrobce robotů ABB unikátní možnost vizualizace robotické automatizace prostřednictvím chytrých telefonů nebo tabletů s aplikací RobotStudio® AR Viewer. Tato aplikace umožňuje přenést modely robotů do skutečného prostředí pro detailní pozorování velikosti a rozsahu robota či robotické buňky před jejich implementací. Výrobky ABB lze nejen přenášet do reálného prostředí, ale také naprogramovat a optimalizovat jejich chování před samotným nasazením.[38]

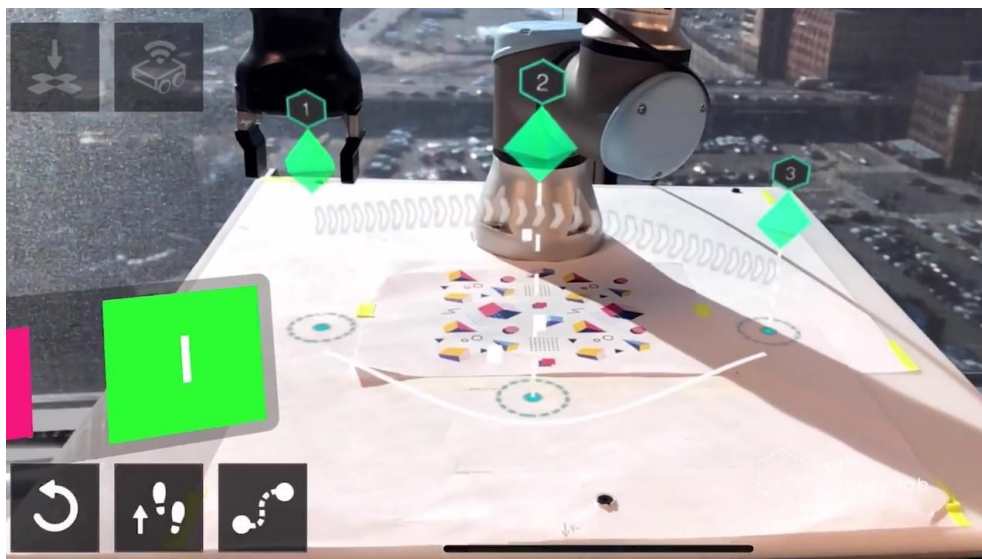
AR Viewer od ABB se již osvědčil jako klíčový nástroj pro firmy, které teprve začínají s robotickou automatizací nebo ještě nezačaly plánovat instalaci. V praxi se ukázal jako cenný pomocník při interaktivním testování modelů v průmyslovém prostředí. Pomáhá výrobcům a inženýrům lépe porozumět potenciálním řešením a snadněji implementovat robotickou

automatizaci. V nedávné době byl tento nástroj úspěšně implementován ve společnosti Volvo v Curitibě, Brazílii, kde inženýrské týmy využívají AR Viewer ke zrychlení návrhového procesu a zlepšení spolupráce při implementaci robotických řešení (Obrázek 2-9).[39]



Obrázek 2-9 - AR aplikace RobotStudio® AR Viewer ve společnosti Volvo[39]

Tým vývojářů představil na konferenci CHI v roce 2020 vývojovou platformu Kinetic AR, která umožňuje uživatelům intuitivně programovat pohyby robotických systémů ve fyzickém prostoru prostřednictvím mobilní rozšířené reality a jejím cílem je usnadnit spolupráci mezi lidmi a roboty v reálném prostředí. Ovládání robota je realizováno umístěním prostorových značek a definováním trajektorií přímo v prostoru kolem uživatele a robota (Obrázek 2-10).



Obrázek 2-10 - Ovládání kolaborativního robota s Kinetic AR[40]

Tuto technologii je možné využít k propojení a automatizaci výrobních linek s kolaborativními a mobilními roboty. Uživatelé mohou jednoduše a intuitivně definovat a optimalizovat pohyby robotů podle aktuálních potřeb výrobní linky. To přináší rychlou adaptaci na změny ve výrobním procesu a efektivní využití robotů podle konkrétních úkolů. [41]

Podobná aplikace, jako je Kinetic AR, bude využívána v praktické části bakalářské práce. Její popis a podpůrný program pro spuštění serveru bude popsán v následující kapitole.

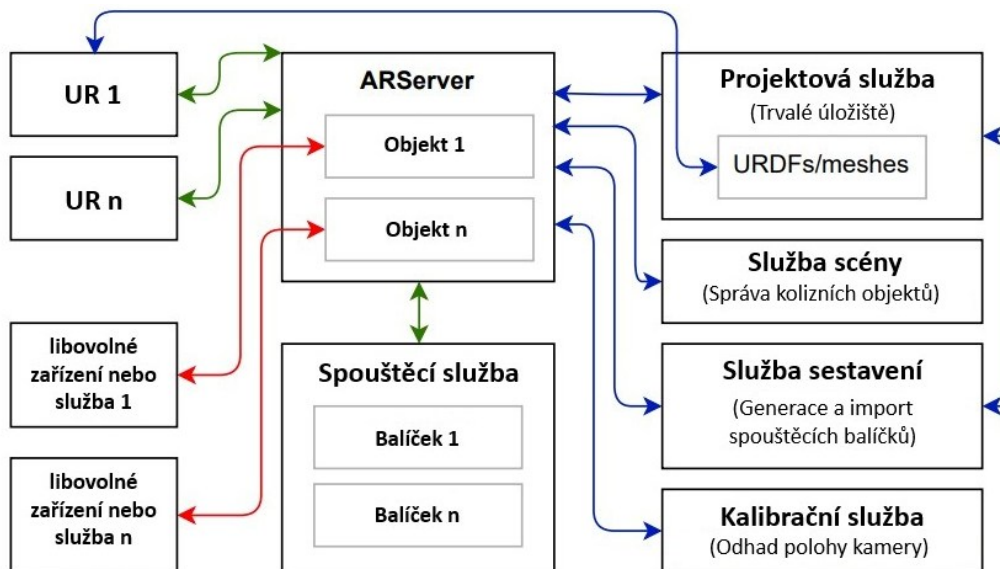
3 ARCOR2

Vývojová platforma ARCOR2, neboli Augmented Reality Collaborative Robot, slouží pro zjednodušené programování, kompletní správu pracoviště s robotem či celých výrobních linek, a využívá rozšířenou realitu k vizualizaci programových dat. Uživatelské rozhraní je možné využít pro všechny typy zařízení bez omezení na výrobce či typ robotu. Výhodou je možnost integrace nového zařízení pomocí vlastního doplňku v programovacím jazyce Python. Tento přístup eliminuje potřebu školení pro každé zařízení, a vede ke snížení nákladů a času potřebných k přípravě a provozu. Za vznikem celé platformy stojí tým Robo@FIT z Fakulty informačních technologií Vysokého učení technického v Brně.[42]

3.1 Architektura

Aplikační rámec je rozdělen na sadu služeb a uživatelské rozhraní. Hlavní službou systému je ARServer založený na programovacím jazyce Python, který zprostředkovává komunikaci mezi službami a uživatelským rozhraním. Rozhraní jsou k serveru připojena prostřednictvím počítačového komunikačního protokolu WebSocket, který zajišťuje obousměrnou komunikaci. Mezi další služby, které zajišťují chod aplikace, patří:[43]

- **Projektová služba** poskytuje trvalé úložiště dat relevantních pro pracoviště (scény, projekty, typy objektů, modely).
- **Služba scény** je zodpovědná za správu kolizních objektů.
- **Služba sestavení** vytváří pro daný projekt samostatný spustitelný balíček se sadou doplňkových tříd, např. zjednodušujících práci s akčními body.
- **Spouštěcí služba** spravuje balíčky vytvořené službou sestavení. Zajišťuje jejich spuštění, pozastavení nebo obnovení.
- **Kalibrační služba** slouží k detekci markerů, následnému výpočtu polohy kamery a vytvoření konečné 6D pózy.



Obrázek 3-1 - Blokové schéma systému ARCOR2[43]

Na obrázku 3-1 je možné pozorovat blokové schéma systému a propojení serveru s jednotlivými službami a uživatelským prostředím ve stavu, kdy jsou instance objektů již vytvořené. Zelené čáry značí spojení skrze WebSocket, modré komunikují skrze standardizované rozhraní REST API, které umožňuje přístup k datům a funkcím

prostřednictvím webového protokolu HTTP. Komunikaci mezi libovolným zařízením či další službou si může implementátor zvolit dle vhodnosti.

3.2 Docker

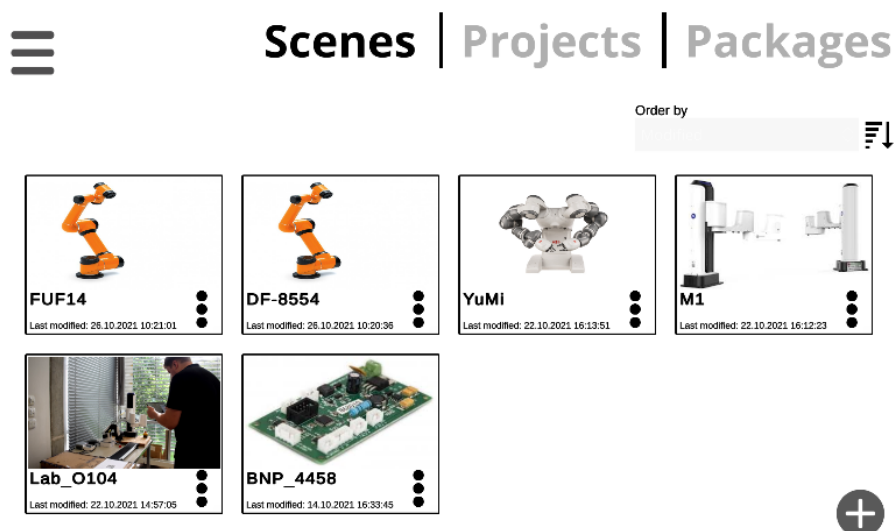
Spuštění ARServeru je nezbytné provést pomocí softwaru Docker. Docker je virtualizační platforma, která umožňuje balení aplikací do kontejnerů a spuštění těchto kontejnerů na různých operačních systémech. Jedná se o populární nástroj vývojářů a správců systémů, protože umožňuje jednoduché a konzistentní nasazení aplikací. Veškeré služby a součásti ARServeru jsou definovány v jednom konfiguračním souboru ve formátu YAML, usnadňující správu a konfiguraci serveru. Díky Dockeru není správa serveru omezena na jeden konkrétní operační systém, ale lze ARServer spouštět a provozovat na různých platformách, včetně Microsoft Windows, macOS a různých distribucí Linuxu.[44]

3.3 Uživatelské prostředí AREditoru

Pro interakci uživatele s kolaborativními roboty slouží aplikace AREditor, která je vytvořená pro mobilní zařízení s operačním systémem Android a podporou pro platformu rozšířené reality ARCore od společnosti Google. Nabízí uživatelům bez předchozích znalostí robotiky a programování možnost snadno nastavit, anotovat, programovat a modifikovat pracoviště pomocí rozšířené reality. Programování probíhá v grafické podobě a je usnadněno zobrazením prostorových vztahů přímo na scéně. V následující části bude základní ovládání aplikace více přiblíženo. [42]

Hlavní obrazovka

Po přihlášení k serveru, které bude popsáno v následující podkapitole, se zobrazí hlavní obrazovka se seznamy dostupných scén, projektů a balíčků, mezi kterými je možné přepínat skrze tlačítka v pravém horním rohu (Obrázek 3-2). V dolním pravém rohu lze tlačítkem plus jednotlivé projekty vytvářet. Úpravy, jako je přidání do oblíbených položek, duplikování, přejmenování, změna obrazu, odstranění atd., se provádějí v otevřeném kontextovém menu, které se zobrazuje po klepnutí na tři tečky.



Obrázek 3-2 - Hlavní obrazovka AREditoru[42]

V levé části obrazovky se po kliknutí na tři vodorovné čáry zobrazí navigační menu, skládající se ze tří bloků. V prvním bloku jsou prezentovány běžné informace, například název

hostitele nebo IP adresa ARServeru, verze AREditoru a ARServeru. V dalších dvou blocích jsou tlačítka pro následující funkce: zobrazení hlavní obrazovky, oblíbených projektů, oznámení z aktuální instance aplikace, protokolů a nastavení, dále zde najdeme tlačítko pro přepínání mezi manuální a automatickou kalibrací, odpojení od serveru a kompletní zavření aplikace.

Editor

Hlavní pracovní prostředí aplikace je editor pro tvorbu a editaci scén a projektů. Zde se prolíná reálný svět s modely v rozšířené realitě. Pro ovládání je k dispozici navigační menu, ovládací menu a výběrové menu (Obrázek 3-3).



Obrázek 3-3 - Hlavní pracovní prostředí AREditoru

Navigační menu obsahuje název hostitele nebo IP adresu ARServeru v horním řádku, po kterém následuje několik tlačítek: uložení protokolu, zobrazení notifikací, recalibrace modelu, odpojení od serveru a kompletní zavření aplikace.

Hlavní ovládací menu disponuje pěti hlavními tlačítky pro ovládání aplikace, mezi ně patří: oblíbené, přidat, nastavení, domov a robot. Po rozkliknutí těchto nástrojových tlačítek se zobrazí panel s tlačítky pro ovládání dané kategorie.

Výběrové menu společně s výběrovým křížem uprostřed obrazovky slouží jako hlavní interakční widget pro výběr objektů. Na horní hraně jsou zobrazena čtyři tlačítka filtru, která umožňují uživatelům z interakce vyřadit roboty, akční objekty, akční body a akce. Uprostřed nabídky je umístěn seznam objektů. Každá položka seznamu se skládá z ikony objektu následované jejím názvem. Na spodní straně má menu čtyři tabulátory měnící řazení a zobrazování objektů v nabídce: Aim, Alphabet, NoPose a BlockList. Celé výběrové či kontextové menu se může měnit dle zvolených nástrojů v ovládacím menu, například při pohybu robota se otevře menu pro jeho ovládání.

Po detailním pohledu na platformu ARCOR2 bude následující kapitola zaměřena na její implementaci v reálném pracovním prostředí.

4 Ovládání kolaborativního robota

Cílem této bakalářské práce je implementovat rozšířenou realitu ARCOR2 u kolaborativního robota, a s využitím mobilního zařízení ho ovládat. Tato kapitola se detailně zabývá procesem implementace vývojové platformy ARCOR2 na pracovišti, které je součástí Katedry průmyslového inženýrství a managementu na Fakultě strojní Západočeské univerzity v Plzni. Nejprve je popsáno samotné pracoviště s jeho vybavením, poté virtuální zkouška modelu robota, která slouží zároveň k ověření funkčnosti serveru a aplikace. Následuje nahrání balíčku fanucpy do řídicí jednotky robota Fanuc pro zajištění komunikace mezi ARserverem platformy ARCOR2 a řídicí jednotkou. Dále je provedena praktická zkouška, která se zaměřuje na pohyb robota skrze prostředí rozšířené reality aplikace AREditor.

4.1 Pracoviště

Pracoviště využívané v praktické části Průmysl 4.0 disponuje kolaborativním robotem, podavačem puků, samotnými puků různých barev a červenými boxy pro umístění puků (Obrázek 4-1). Další vybavení laboratoře je využíváno pro jiné experimenty.



Obrázek 4-1 - Kolaborativní robot Fanuc CR-7iA/L

Kolaborativní robot Fanuc CR-7iA/L je ideálním řešením pro bezpečnou spolupráci s lidskými operátory. Tato šestiosá prodloužená verze modelu poskytuje dosah až 911 mm s nosností 7 kg. Umožňuje manipulaci a montáž malých dílů s vysokou přesností. Díky stupni krytí IP67 a certifikaci UV ISO 10218-1:2011 je zajištěna bezpečnost spolupráce s člověkem bez potřeby ochranné klece. Robot může být vybaven širokou škálou senzorů, včetně FANUC Vision Sensor a FANUC 3D Area Sensor. Toto vybavení mu umožňuje lokalizovat a rozpoznat objekty s vysokou úrovní přesnosti a efektivity. Robot je vybaven řídicí jednotkou R-30iB Mate Plus, která s více než 250 softwarovými funkcemi zajišťuje výkonnost a bezpečnost robota (Obrázek 4-2). Dále disponuje rychlou konektivitou přes ethernetovou síť, která bude v této práci využita k propojení zařízení.



Obrázek 4-2 - Řídicí jednotka R-30iB Mate Plus

K řídicí jednotce je připojen Fanuc iPendant-Touch (Obrázek 4-3), který spojuje uživatelsky přívětivé rozhraní s vysokým výkonem. Zařízení kombinuje klasická tlačítka s dotykovou obrazovkou. Klasická tlačítka poskytují přesnost při ovládání a umožňují snadné naladění do konkrétní pozice. Na monitoru lze zobrazit 3D grafiku a rozdělit displej do dvou oken pro zvýšení efektivity práce. [45]



Obrázek 4-3 - Fanuc iPendant-Touch

4.2 Použití ARCOR2 pro ovládání kolaborativního robota

V této podkapitole bude popsán proces implementace vývojové platformy ARCOR2 na pracoviště, který zahrnuje provedení virtuální zkoušky aplikace AREditor s tabletem, nahrání serveru do řídicí jednotky robota, a následné propojení ARCOR2 s kolaborativním robotem společně se zkouškou jeho ovládání.

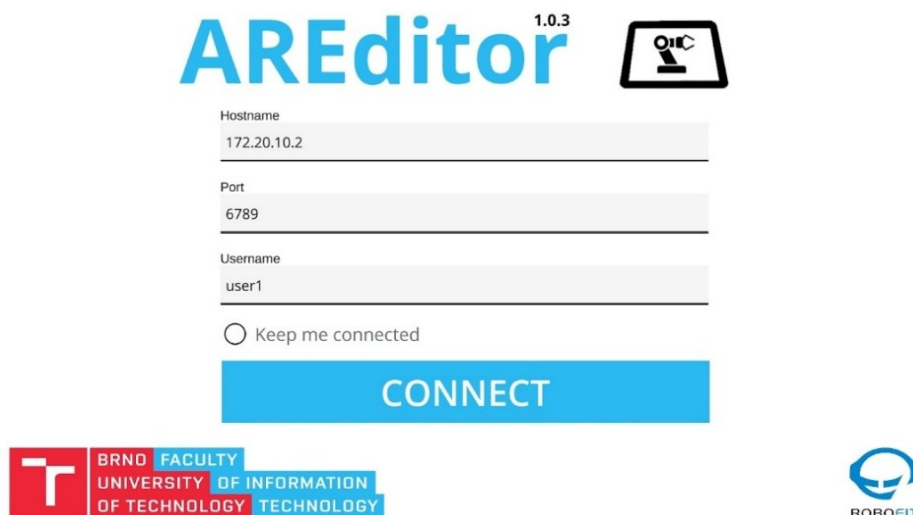
4.2.1 Virtuální zkouška

Z důvodu ověření funkčnosti serveru a aplikace byla nejprve provedena virtuální zkouška, kde byl 3D model kolaborativního robota zobrazen v reálném prostředí, a byla ověřena možnost jeho ovládní. Ke správnému fungování aplikace je potřeba vytvořit server, ke kterému se uživatelské prostředí přes IP adresu připojí. Zapnutí serveru probíhá přes nástroj Docker, a proces je následující. V první řadě je potřeba spustit desktopovou aplikaci Dockeru, dále otevřít v příkazovém řádku adresář se soubory serveru, konkrétně složku fit-demo, která poskytuje YML konfigurační soubor, který lze otevřít příkazem „docker compose up“. Po stažení potřebných dat je server společně se službami a dalšími obrazy spuštěn. Jeho stav a další informace lze pozorovat přímo v aplikaci (Obrázek 4-4). Verze serveru je pro tuto zkoušku zvolena 1.1.1.

Name	Image	Status ↑	CPU (%)	Port(s)
fanuc-demo		Running (9/10)	3.96%	
scene 89ad7cc3195c	arcor2/arcor2_scene:1.0.0	Running	1.73%	5013:5013
fanuc-demo-calibration-1 284b0f52cee7	arcor2/arcor2_calibration:1.1.0	Running	0.69%	5014:5014
asset 1cd587f0d88e	registry.gitlab.com/kinalisoft/test-it-off/asset:1.0.0	Running	0.07%	10040:10040
execution ee57b3a365f0	arcor2/arcor2_execution:1.2.0	Running	0%	
robot d49e7fe5f056	arcor2/arcor2_fanuc:0.3.0	Running	0.01%	5027:5027
project a56671e8e0e4	registry.gitlab.com/kinalisoft/test-it-off/project:1.0.1	Running	0.03%	10000:10000 Show all ports (2)
build e583a69e4da7	arcor2/arcor2_build:1.3.2	Running	1.41%	5008:5008
arserver bfd1a1e9b4c	arcor2/arcor2_arserver:1.1.1	Running	0.02%	6789:6789 Show all ports (2)
nginx 77eea0004bfb	nginx:1.23.3	Running	0%	6790:80
upload-object-types 268347f6b4c7	arcor2/arcor2_fanuc_upload_object_types:0.3.1	Exited	0%	

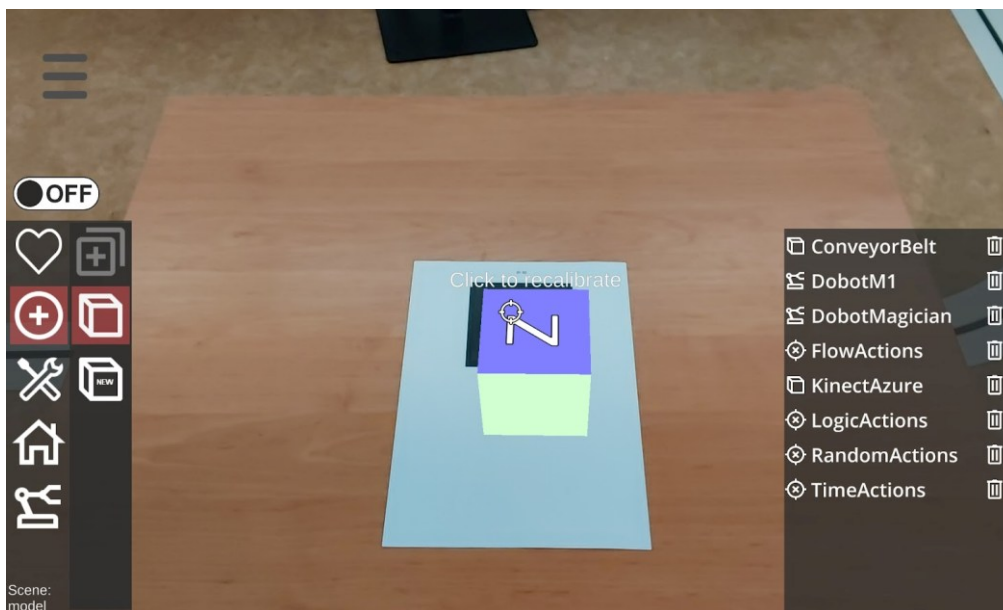
Obrázek 4-4 - Docker fit-demo ARServer

Pro připojení k serveru přes aplikaci AREditor je potřeba zadat IP adresu a port ke komunikaci (Obrázek 4-5). IPv4 adresu je možné zjistit přes příkazový řádek příkazem ipconfig, port je daný v kódu serveru a zobrazuje se v seznamu obrazů v Dockeru. Verze aplikace je 1.0.3.



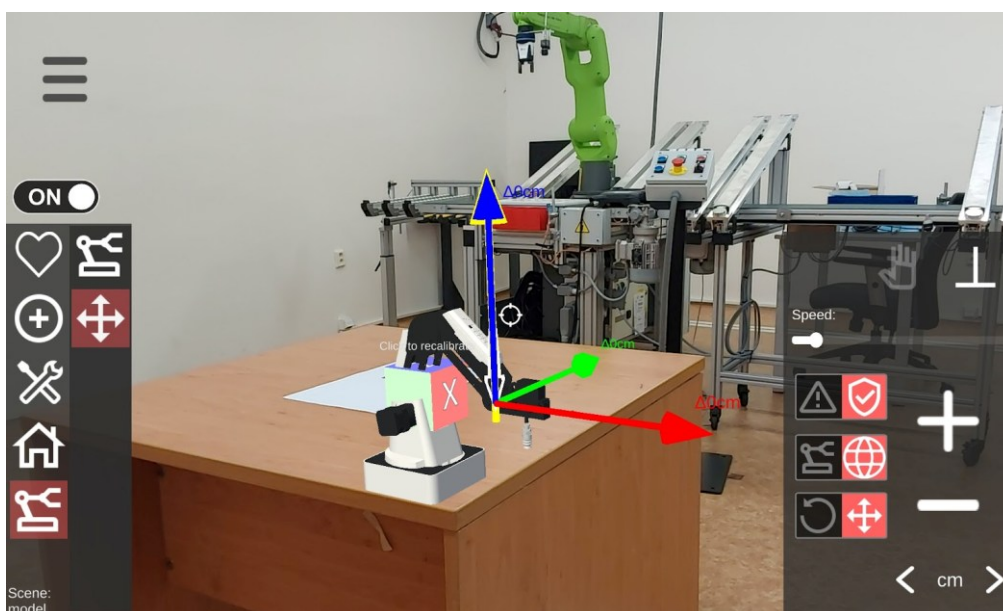
Obrázek 4-5 - Připojení k serveru AREditor

V dalším kroku byla v hlavní obrazovce vytvořena scéna, a po jejím zobrazení je nutné provést kalibraci. Pro kalibraci je potřeba pomalu pohybovat tabletem po pracovní ploše. Pokud knihovna sledování ARFoundation zaznamená alespoň jednu vodorovnou rovinu a detekuje více než jeden bod, automaticky aktivuje sledování, a přechází do kalibračního režimu. Pro úspěšnou kalibraci je potřeba namířit kameru tabletu na viditelnou kalibrační značku neboli marker. Kalibrace poté proběhne automaticky, a na značce se zobrazí kalibrační krychle (Obrázek 4-6). Automatické překalibrování je nastaveno na hodnotu 120 sekund z důvodu eliminace nežádoucích přesunů krychle v prostoru, případně lze kalibraci provést v navigačním menu manuálně.



Obrázek 4-6 - Kalibrační krychle

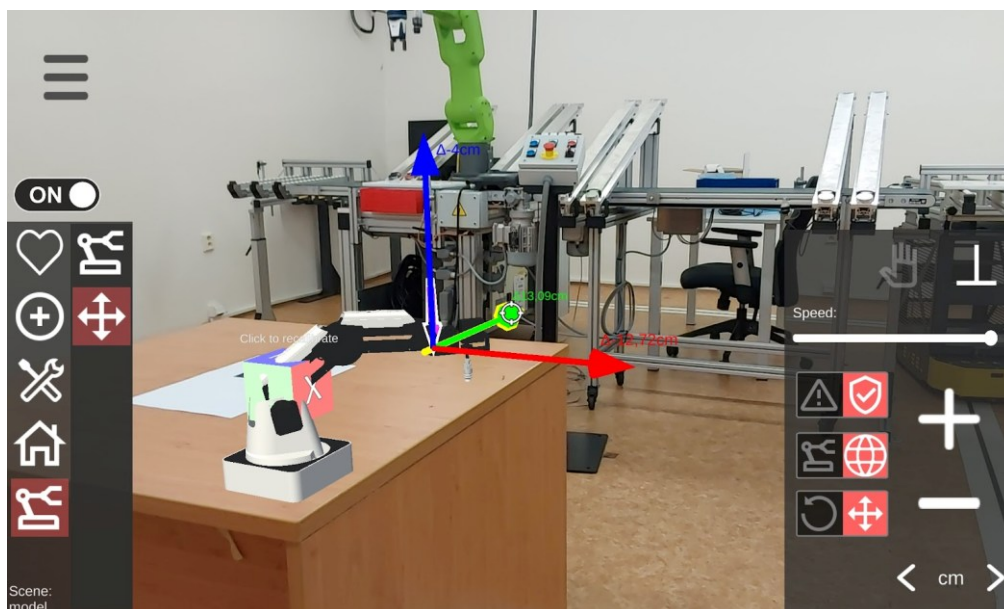
Načtení modelu robota je provedeno skrze kontextové menu v pravém dolním rohu, kde je zvolen objekt DobotMagician. Robot je následně zobrazen na kalibrační krychli. Pro ovládání je nutné se přepnout do online režimu, aby došlo k odemknutí os robota. Dále zvolit v ovládacím menu kategorii robota s krokovacím menu (Obrázek 4-7).



Obrázek 4-7 - Objekt DobotMagician

V kontextovém menu lze nyní nalézt tlačítka + a -, která slouží pro posuv ramene ve zvolené ose. Výběr osy je zvolen prostřednictvím výběrového kříže a délku posuvu lze měnit šipkami pod těmito tlačítky. V horní části menu se nachází ikona dlaně, která odemkne polohu ramene a umožní manuální pohyb. Tlačítko kolmosti slouží k zarovnání koncového efektoru kolmo k zemi. Pro upravení rychlosti pohybu koncového efektoru je zde rychlostní posuvník. Na levé straně nabídky naleznete tři spínače. První slouží k přepínání mezi režimem bezpečného a nebezpečného pohybu. Bezpečný pohyb znamená, že robot nedovolí žádný pohyb, který by mohl vést k možné kolizi. Druhý spínač umožňuje přepínat mezi globálními a lokálními transformačními systémy. Poslední tlačítko určuje, zda se koncový efektor pohne ve vybrané ose posunem nebo rotací.

Během virtuální zkoušky bylo rameno robota úspěšně posunuto ve všech osách. Porovnáním kalibrační pozice a základní polohy zobrazené na obrázku 4-7 s výslednou polohou na obrázku 4-8 je patrné, že v ose X došlo k zápornému posunu o 12,72 cm, v ose Y kladnému posunu o 13,09 cm a v ose Z opět k zápornému posunu o 4 cm v porovnání s původními nulovými souřadnicemi.



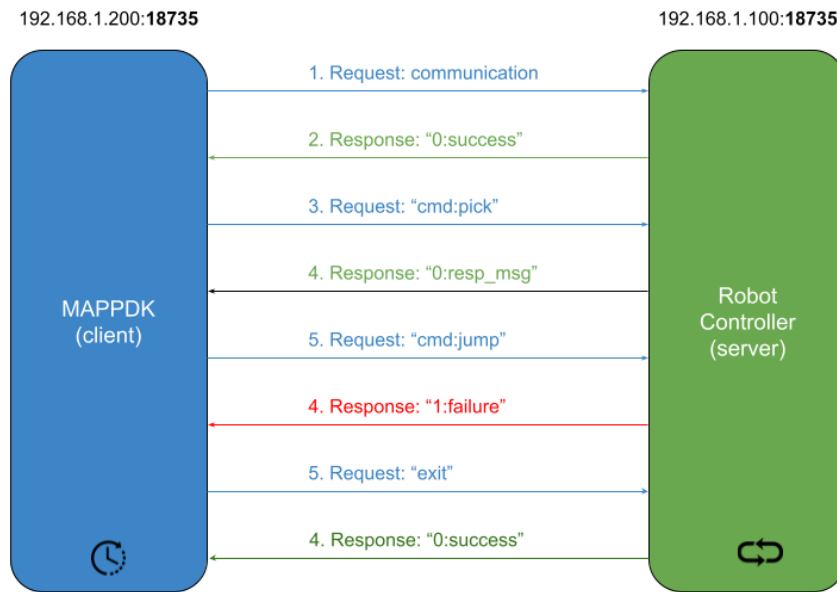
Obrázek 4-8 - Posun robota při virtuální zkoušce

Experiment úspěšně ověřil funkčnost serveru a aplikace pro ovládání 3D modelu kolaborativního robota v reálném prostředí. Virtuální zkouška, která zahrnovala spuštění serveru prostřednictvím nástroje Docker a připojení aplikace AREditor, proběhla bez významných komplikací. Kalibrace robota byla provedena rychle a efektivně s automatickým přechodem do kalibračního režimu poté, co byla detekována vodorovná rovina a kalibrační značka. Během ovládání robota byla pozorována přesná a spolehlivá reakce na zadávané příkazy. Výsledky experimentu ukázaly, že robot byl úspěšně ovládán ve všech osách, s přesným sledováním pohybu a minimálními odchylkami od očekávaných výsledků.

4.2.2 Nahrání knihovny fanucpy do řídicí jednotky

Pro komunikaci mezi ARserverem platformy ARCOR2, využívající programovací jazyk Python, a řídicí jednotkou kolaborativního robota Fanuc byla potřeba do pendantu jednotky nahrát knihovnu fanucpy. Bez této knihovny by nebyl robot schopný přijímat příkazy a komunikovat skrze aplikaci AREditor. Tento projekt vyvinuli v Institutu pro pokročilou výrobu na Univerzitě Nottinghamu jako součást vzdělávací sítě digitální výroby. Na obrázku

4-9 lze pozorovat komunikační protokol v jazyce Python mezi modulem MAPPPDK a pendantem robota.



Obrázek 4-9 - Komunikační protokol mezi Python a řídicím systémem robota FANUC[46]

Implementace knihovny fanucpy byla provedena dle těchto kroků:

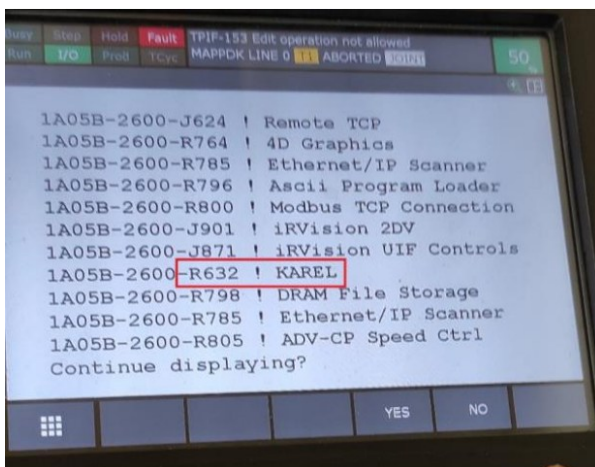
Ověření systémové kompatibility

Nejprve bylo potřeba ověřit, zda jsou v řídicí jednotce povoleny následující moduly:

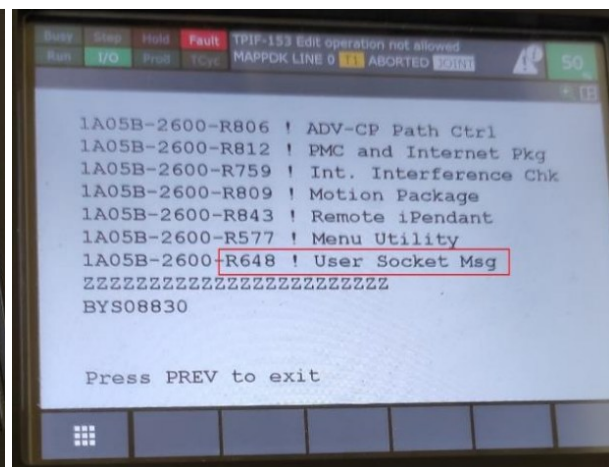
- R632 - KAREL (programovací jazyk pro vytváření vlastních funkcí a úloh pro roboty Fanuc)
- R648 - User Socket Messaging (umožňuje komunikaci prostřednictvím síťových zpráv)

V pendantu bylo potřeba zkontrolovat tuto skutečnost dle následujícího postupu:

1. MENU → NEXT → STATUS → Version ID
2. Kliknuto na ORDER FI
3. Ověřeno, zda jsou moduly R632 KAREL a R648 User Socket Msg v seznamu (Obrázek 4-10, Obrázek 4-11)



Obrázek 4-10 - Modul R632 - KAREL



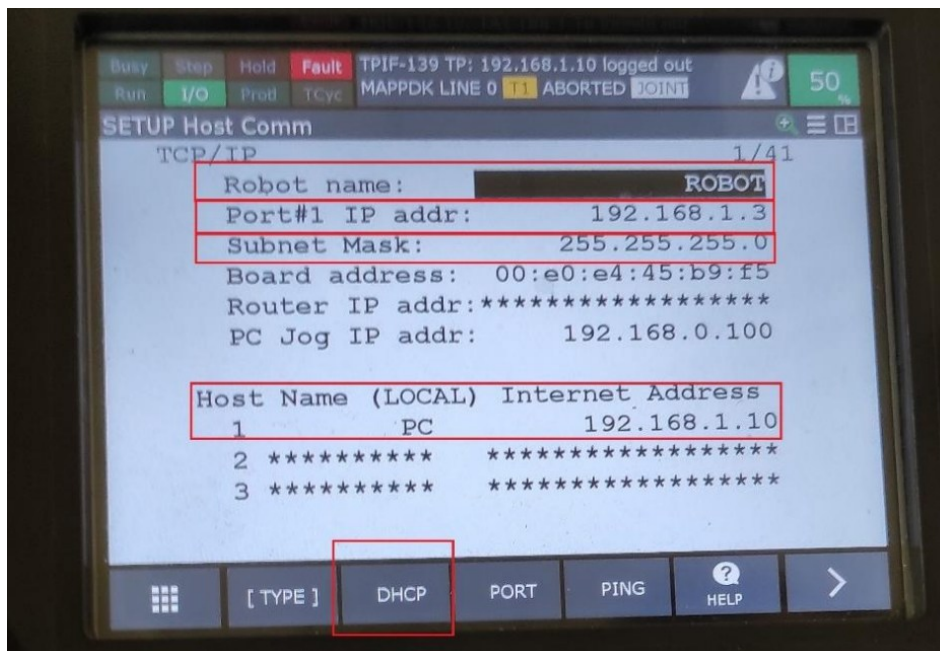
Obrázek 4-11 - Modul R648 - User Socket Messaging

Nastavení připojení

Počítač byl připojen ethernetovým kabelem k robotu, ručně byla nastavena ve vlastnostech síťového připojení počítače IP adresa a maska podsítě. Ověření správnosti nastavení bylo provedeno příkazem „ipconfig“ v příkazovém řádku.

Dále bylo potřeba nastavit IP adresu a masku podsítě i v pendantu. Zde bylo postupováno následovně:

1. **MENU** → **SETUP** → **Host Comm**
2. Kliknuto na **TCP/IP**
3. Nastavení jména robota, IP adresy, masky podsítě a hostující IP adresu zařízení, ke kterému je robot pomocí ethernet kabelu připojen (Obrázek 4-12)
 - **Robot name:** Robot (může být jakýkoliv jméno, např. Robot1, RobotA, apod.)
 - **Port#1 IP addr:** 192.168.1.3 (IPv4 adresa robota ve formátu 192.168.xxx.xxx)
 - **Subnet mask:** 255.255.255.0 (musí být shodná s maskou podsítě s hostujícím zařízením)
 - **Internet Address:** 192.168.1.10 (IPv4 adresa hostujícího zařízení)



Obrázek 4-12 - Pendant nastavení Host Comm

4. Vypnutí DHCP: **DHCP** -> **DHCP Enable: False**
5. Aktivace tohoto nastavení tlačítkem **INIT**

Nyní je řídicí jednotka propojena s počítačem. To umožňuje obousměrnou komunikaci a přenos souborů přes ethernetový kabel.

Nastavení serveru

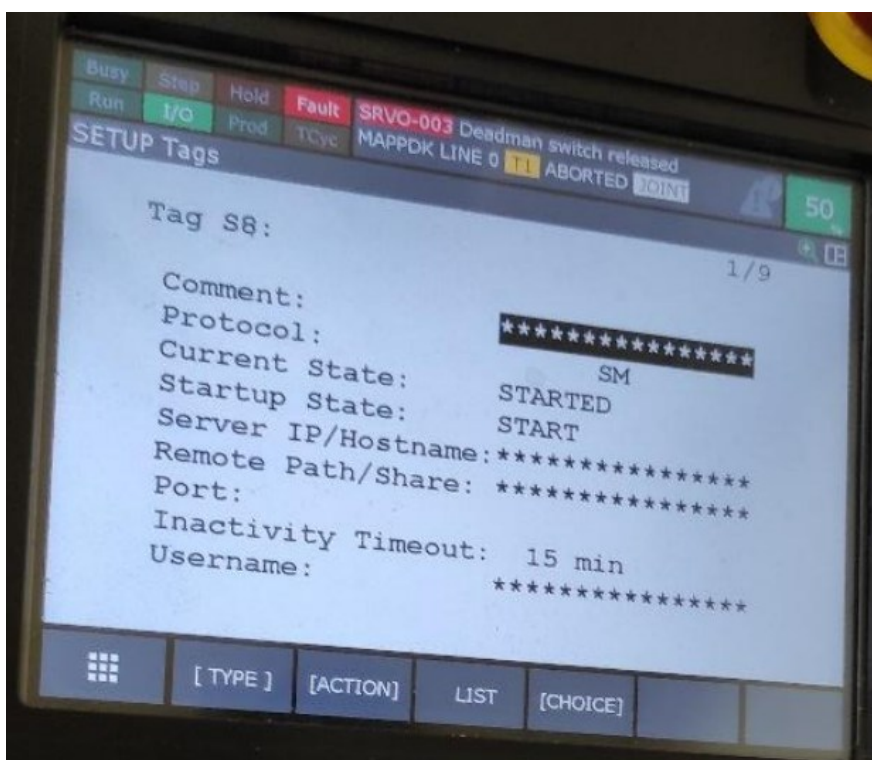
Dále bylo potřeba nastavit FANUC MAPPDK server na výchozí hodnoty, kde je označení serveru “S8:“ a port 18735.

1. **MENU** → **Next** → **SYSTEM** → **Variables**
2. Vybráno **\$HOSTS_CFG**, potvrzen výběr a zvoleno číslo **8**
3. Nastavena proměnná **\$SERVER_PORT** na **18735** (Obrázek 4-13)



Obrázek 4-13 - Pendant nastavení Variables

4. **MENU** → **SETUP** → **Host Comm**
5. **SHOW** → **Servers**, zvoleno **S8**
6. Nastaven **Protocol** na **SM**, **Startup State** na **START**, **Current State** na **STARTED** (Obrázek 4-14)



Obrázek 4-14 - Pendant nastavení Tags

7. **[ACTION]** → **DEFINE**
8. **[ACTION]** → **START**

Nastavení loggeru

Obdobně, jako byl nastaven Logger, tak i server FANUC MAPPDK ve výchozím nastavení používá označení serveru "S7:" a port 18736. Postup byl stejný jako nastavení Loggeru, a sice:

1. **MENU** → **Next** → **SYSTEM** → **Variables**
2. Vybráno **\$HOSTS_CFG**, potvrzen výběr a zvoleno číslo **7**
3. Nastavena proměnná **\$SERVER_PORT** na **18736**
4. **MENU** → **SETUP** → **Host Comm**
5. **SHOW** → **Servers**, zvoleno **S7**
6. Nastaven **Protocol** na **SM**, **Startup State** na **START**, **Current State** na **STARTED**
7. **[ACTION]** → **DEFINE**
8. **[ACTION]** → **START**

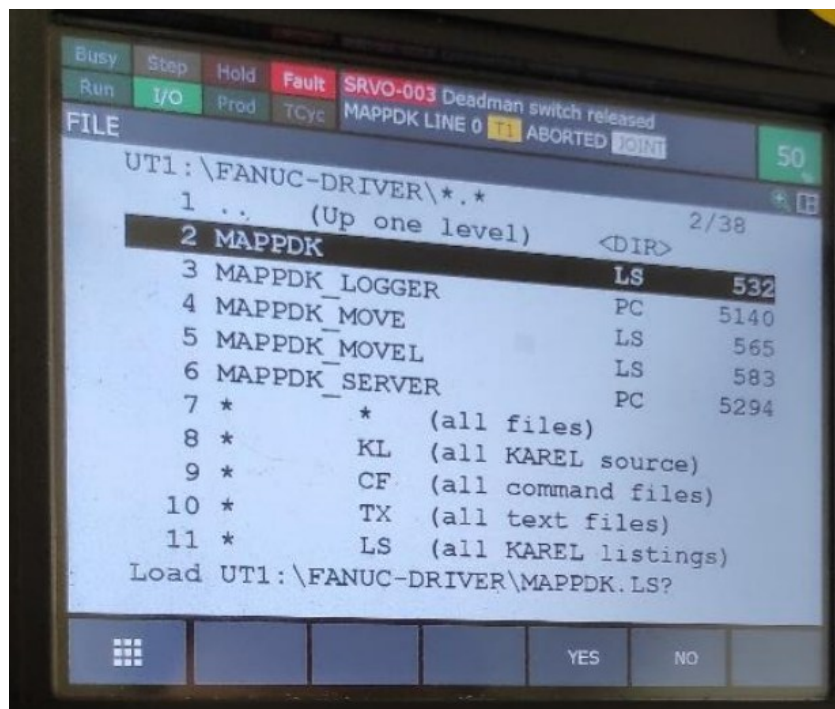
Nahrání souborů MAPPDK

V dalším kroku bylo potřeba do řídicí jednotky nahrát následující soubory skrze USB flash disk, který byl zapojen do ovladače robota:

- **mappdk.ls**: Hlavní soubor MAPPDK, který spouští prostřednictvím pendantu. Tento soubor spouští MAPPDK server a MAPPDK Logger soubory.
- **mappdk_server.pc**: MAPPDK server soubor
- **mappdk_logger.pc**: MAPPDK logger soubor
- **mappdk_move.ls**: MAPPDK soubor pro pohyb
- **mappdk_move1.ls**: MAPPDK soubor pro lineární pohyb

Postup nahrávání byl následující:

1. **MENU** → **FILE** → **File**
2. **[UTIL]** → **Set Device** a zvoleno "7 USB on TP (UT1:)"
3. Složka s nahranými soubory
4. Soubory byly nahrány příkazem **LOAD** → **YES** (Obrázek 4-15)



Obrázek 4-15 - Pendant nahrání MAPPDK

Ověření nastavení připojení

V poslední fázi bylo ověřeno správné propojení mezi robotem a počítačem skrze příkazový řádek příkazem „ping“ s IP adresou robota. Ze čtyř odeslaných paketů bylo obdrženo od všech odpověď s maximální délkou zpoždění 1 ms.

Po úspěšné implementaci knihovny fanucpy do řídicí jednotky robota Fanuc byla zajištěna komunikace s počítačem. Celý proces zahrnoval ověření kompatibility, nastavení připojení přes ethernetový kabel, nastavení IP adresy a masky podsítě, nakonec nastavení FANUC MAPDPK serveru a loggeru.

4.2.3 Praktická zkouška ovládání robota

V praktické zkoušce bylo cílem pohnout ramenem robota pouze skrze prostředí rozšířené reality tabletu aplikace AREditor. Tento úkol vyžadoval nastavení a propojení mezi tabletovým zařízením, počítačem a řídicí jednotkou robota, aby došlo k jejich správné komunikaci. Díky předchozímu nahrání knihovny fanucpy do řídicí jednotky robota by měl být robot schopen přijímat příkazy napsané v Python kódu vyslané skrze aplikaci AREditor a interagovat s prostředím rozšířené reality na tabletu.

Propojení zařízení bylo provedeno následovně. Z ethernetového výstupu řídicí jednotky robota jsme připojili kabelem počítač, a následně s využitím redukce na USB-C tablet. Pro vzájemnou komunikaci bylo nezbytné zajistit správné nastavení IP adresy a masky podsítě na obou zařízeních. Robot je již z předešlé podkapitoly nahrání fanucpy do řídicí jednotky nastaven na IP adresu 192.168.1.3 a masku podsítě 255.255.255.0. Masku podsítě byla nastavena pro počítač i tablet stejná, ale u síťové adresy bylo potřeba změnit poslední oktet. Počítač jsme nastavili na IP adresu 192.168.1.1 a tablet na 192.168.1.7.

```
C:\Users\svest>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time=1ms TTL=60
Reply from 192.168.1.3: bytes=32 time=1ms TTL=60
Reply from 192.168.1.3: bytes=32 time=1ms TTL=60
Reply from 192.168.1.3: bytes=32 time<1ms TTL=60

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Users\svest>ping 192.168.1.7

Pinging 192.168.1.7 with 32 bytes of data:
Reply from 192.168.1.7: bytes=32 time=5ms TTL=64
Reply from 192.168.1.7: bytes=32 time=1ms TTL=64
Reply from 192.168.1.7: bytes=32 time=1ms TTL=64
Reply from 192.168.1.7: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 5ms, Average = 2ms
















C:\Users\svest>arp -a

Interface: 192.168.1.1 --- 0x8
    Internet Address      Physical Address         Type
    192.168.1.3           00-e0-e4-45-b9-f5      dynamic
    192.168.1.7           00-e0-4c-02-a2-89      dynamic
```

Obrázek 4-16 - Ověření připojení ping a ARP

Ověření připojení bylo provedeno příkazem „ping“ v příkazovém řádku počítače, dále byl využit příkaz „arp-a“ pro zobrazení ARP (protokol počítačových sítí) tabulky pro získání informací o přiřazených IP adresách v lokální síti. Výsledkem bylo potvrzení, že robot i tablet jsou vzájemně viditelné v síti (Obrázek 4-16). Odezva robota na ping příkaz byla 1 ms, která naznačuje rychlou a stabilní komunikaci. Odezva tabletu byla maximálně 5 ms, což je stále velmi dobrý výsledek. Dále byla zkontrolována ARP tabulka a bylo zjištěno, že obě sítě jsou viditelné a zařízení vzájemně identifikovatelná v lokální síti.

Postup spuštění serveru přes nástroj Docker probíhal podobně jako u virtuální zkoušky. Konfigurační YML soubor docker-compose, který se nyní nachází v adresáři fanuc-demo, bylo potřeba přepsat a aktualizovat verze služeb dle souboru z virtuální zkoušky, aby ho bylo možné použít s aplikací AREditor verze 1.0.3. Bez provedení tohoto přepsání by bylo nemožné spustit server, jelikož by ihned po spuštění došlo k výskytu chybových hlášek a nedošlo by k připojení ke službám. Spuštění serveru bylo uskutečněno přes příkaz „docker compose up“ v příkazovém řádku. Jeho stav, spuštěné služby a obrazy je možné pozorovat na obrázku 4-17. Byla použita verze serveru 1.1.1.

Name	Image	Status ↑	CPU (%)	Port(s)
 fit-demo		Running (12/14)	4.81%	
 fit-demo-calibration-1 262c764aa48d	arcor2/arcor2_calibration:1.1.0	Running	0.7%	5014:5014
 kinect edb968026045	arcor2/arcor2_kinect_azure:0.7.0	Running	1.62%	5016:5016
 asset cee214d50912	registry.gitlab.com/kinalisoft/test-it-off/asset:1.0.0	Running	0%	10040:10040
 scene 315c5c70e286	arcor2/arcor2_scene:1.0.0	Running	0.75%	5013:5013
 execution ef5e71a31b07	arcor2/arcor2_execution:1.2.0	Running	0%	
 project 6ea2f808e7b1	registry.gitlab.com/kinalisoft/test-it-off/project:1.0.1	Running	0%	10000:10000 Show all ports (2)
 nginx 325258c66bf5	nginx:1.23.3	Running	0%	6790:80
 dobot-m1 8a1a3858afee	arcor2/arcor2_dobot:1.2.0	Running	0.01%	5019:5018
 dobot-magician2 1992da2683aa	arcor2/arcor2_dobot:1.2.0	Running	0.01%	5020:5018
 dobot-magician e97fb6db075b	arcor2/arcor2_dobot:1.2.0	Running	0.01%	5018:5018
 build 16aa4c4f62eb	arcor2/arcor2_build:1.3.2	Running	1.68%	5008:5008
 arserver 537d9d2f4f53	arcor2/arcor2_arserver:1.1.1	Running	0.03%	6789:6789 Show all ports (2)
 upload-object-types 83585daa69e2	arcor2/arcor2_upload_fit_demo:1.4.0	Exited	0%	
 upload-builtin-objects 05279a15da30	arcor2/arcor2_upload_builtin_objects:1.0.0	Exited	0%	

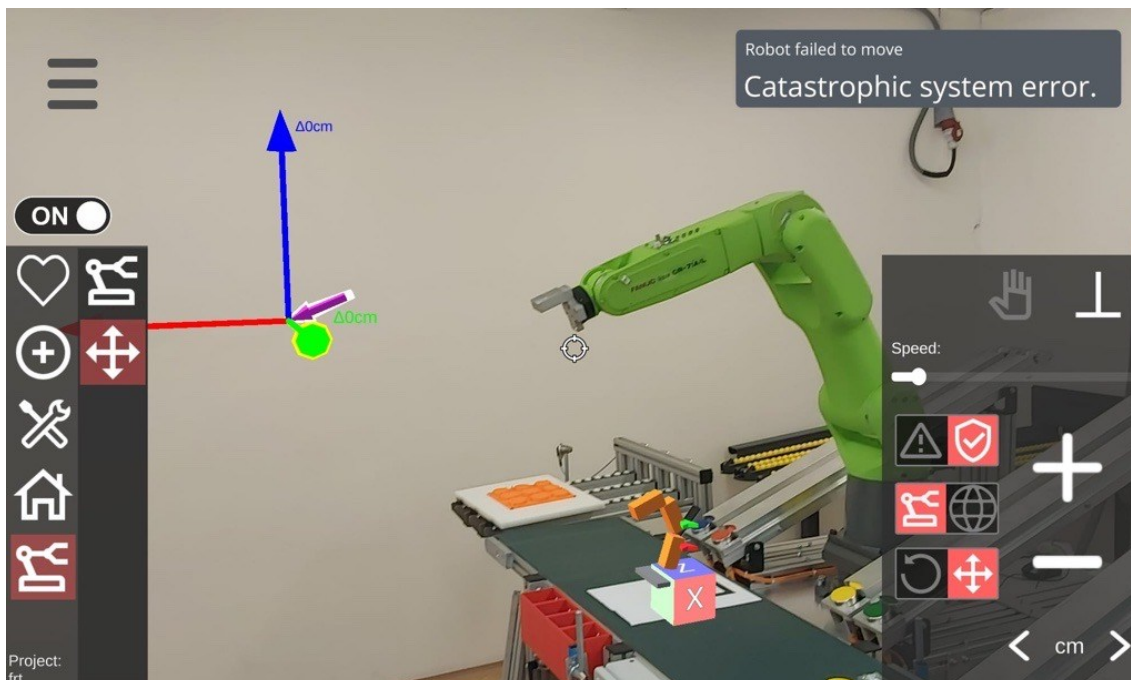
Obrázek 4-17 - Docker fanuc-demo ARServer

Pro připojení aplikace AREditor k serveru byla použita IPV4 adresa počítače, tedy 192.168.1.1 a základní port 6789. Po vytvoření scény byla úspěšně provedena kalibrace za pomoci kalibračního markeru umístěným před kolaborativním robotem. V kontextovém menu byl načten model robota s názvem fanuc_1_r_mate200id7_1, který simuluje našeho reálného Fanuc robota (Obrázek 4-18).



Obrázek 4-18 - Model fanuc_l_r_mate200id7_l

Následně se bylo potřeba pro možnost ovládní robota přepnout do projektu, a načíst již předvytvořenou scénu s kalibrací a nahaným robotem. Pro pohyb koncového efektoru je potřeba přejít do online módu, a otevřít stepovací menu. Pohyb je prováděn výběrem požadované osy a tlačítky pro posuv + a – jako u virtuální zkoušky. Avšak poté, co byl proveden posun v jedné ze souřadnicových os, se robot nepohyboval a zobrazilo se chybové hlášení "Catastrophic system error" (Obrázek 4-19).



Obrázek 4-19 - Chybové hlášení "Catastrophic system error"

V příkazovém řádku serveru bylo vypsáno následující (Obrázek 4-20):

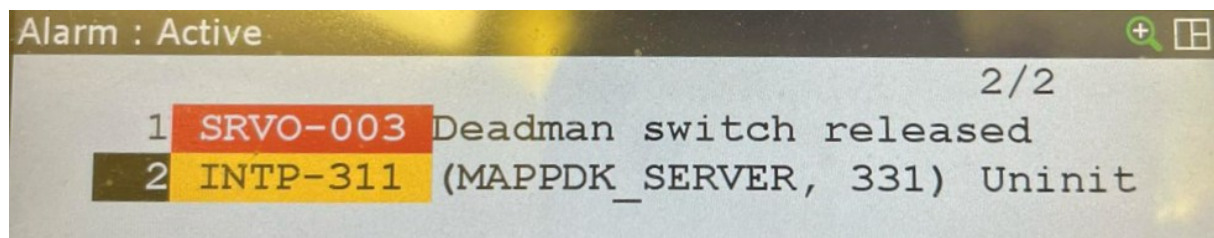
```
fanuc-demo-robot | 2024-04-03 12:25:41 INFO
Moving to Pose(position=Position(x=0.7509700000000002,
y=0.022730000000000146, z=0.5810699999999999),
orientation=Orientation(x=0.8270461171786058, y=0.05945429957030829,
z=0.5547534193178413, w=0.06861887551949225)), velocity: 5520.0, acceleration:
50.0, linear: True, safe: False

fanuc-demo-arserver | 2024-04-03 12:26:01 ERROR
Robot movement failed with: Catastrophic system error.
```

Obrázek 4-20 - Výpis příkazového řádku při chybě

Tento záznam dokumentuje událost, kdy aplikace využívající systém robota vydala příkaz k provedení pohybu na specifickou pozici s danou orientací a doprovodnými parametry pohybu, včetně rychlosti a zrychlení. Tento pokyn byl přenesen prostřednictvím serveru do robota. Nicméně, tento pokus o pohyb skončil neúspěchem a vyvolal chybu označenou jako "Katastrofická systémová chyba". Tento stav indikuje, že plánovaný pohyb robota selhal kvůli závažnému problému.

Chybová hláška byla ve stejném okamžiku vypsána i v pendantu robotu, kde se zobrazila chyba "INTP-311 (MAPPDK_SERVER, 311) Uninit" (Obrázek 4-21), která značí, že byly aplikaci vyslány do řídicí jednotky neinicializované příkazy k pohybu, které nejsou v programovacím jazyce robota KAREL definovány. [47]



Obrázek 4-21 - Pendant chyba INTP-311

I přes detailní analýzu příčin této chyby a další testy se nepodařilo přesně identifikovat důvod selhání. Nastalá chyba byla konzultována s týmem Robo@FIT a odborníkem na Fanuc roboty, avšak ani jejich pomoc nepřinesla žádné zásadní poznatky. Dalším krokem k nalezení příčiny by mohlo být detailní prozkoumání zdrojového kódu platformy ARCOR2. Vzhledem k rozsahu a časové náročnosti tohoto úkolu není provedení této akce v bakalářské práci možné.

4.2.4 Proces implementace a zhodnocení

Během implementace se bylo potýkáno s řadou technických výzev, které byly postupně řešeny ve spolupráci s týmem Robo@FIT. Každá fáze přinesla svá specifická úskalí a zjištění.

První výzvou bylo úspěšné spuštění serveru a připojení aplikace AREditor. V počátcích byl využit starší mobilní telefon LG G6 s Androidem 9, u kterého docházelo k vypínání aplikace po otevření editoru. Tento problém byl vyřešen přechodem na novější tablet od značky Samsung s Androidem 13. Následně bylo potřeba použít kalibračních markerů ve velikosti 4x4, přičemž při prvních testech byla použita velikost 7x7, u kterých nedocházelo ke správné kalibraci.

Náhrání fanucpy do řídicí jednotky robota proběhlo bez komplikací. Funkčnost implementace byla ověřena jednoduchým programem pro posun ramene napsaným v Pythonu.

V praktické zkoušce nebylo možné spustit ARServer z důvodu chybně napsaného kódu v YML konfiguračním souboru fanuc-demo. Tento soubor byl ve spolupráci s týmem vývojářů přepsán, aby fungoval s aktuálními verzemi služeb platformy ARCOR2. Následně byla provedena zkouška různého propojení zařízení, aby došlo k jejich správné komunikaci. Tablet je možné propojit skrze ethernet kabel s redukcí na USB-C, kde bylo potřeba nakonfigurovat nastavení připojení tabletu, nebo přes Wi-Fi.

Tabulka 1 - Časový průběh implementace ARCOR2

20.2.	Virtuální zkouška aplikace na telefonu s verzí Androidu 9 Aplikaci se podařilo spustit a připojit k serveru. Po otevření editoru se celá aplikace vypla.
21.2.	Virtuální zkouška aplikace na telefonu se starší verzí aplikace AREditor Aplikaci se podařilo spustit a připojit k serveru. Po otevření editoru se celá aplikace vypla.
27.2.	Virtuální zkouška aplikace na tabletu s verzí Androidu 13 V aplikaci se již podařilo úspěšně otevřít editor. Chyba nastala u kalibrace, ke které nedošlo.
28.2.	Kontaktování týmu Robo@FIT Bylo potřeba použít kalibračních markerů ve velikosti 4x4. U předchozí zkoušky byla použita velikost 7x7.
1.3.	Virtuální zkouška aplikace na tabletu Zkouška proběhla dle kapitoly 4.2.1.
2.3.	Kontaktování týmu Robo@FIT Získání informací ohledně knihovny fanucpy a jejím nahrání do řídicí jednotky robota. Úprava verzí služeb v kódu souboru fanuc-demo.
25.3.	Nahrání fanucpy do řídicí jednotky robota Nahrání knihovny proběhlo dle kapitoly 4.2.2.
26.3.	Praktická zkouška Při zkoušce se nepodařilo spustit server z důvodu chybně napsaného kódu v YML souboru fanuc-demo.
27.3.	Kontaktování týmu Robo@FIT Přepsání YML souboru a ověření funkčnosti serveru.
28.3.	Praktická zkouška Při této zkoušce bylo zkoumáno, jak propojit počítač, tablet a robota.
3.4.	Praktická zkouška Zkouška proběhla dle kapitoly 4.2.3.
4.4.	Kontaktování týmu Robo@FIT Kontaktování týmu a hledání možné příčiny chyby.

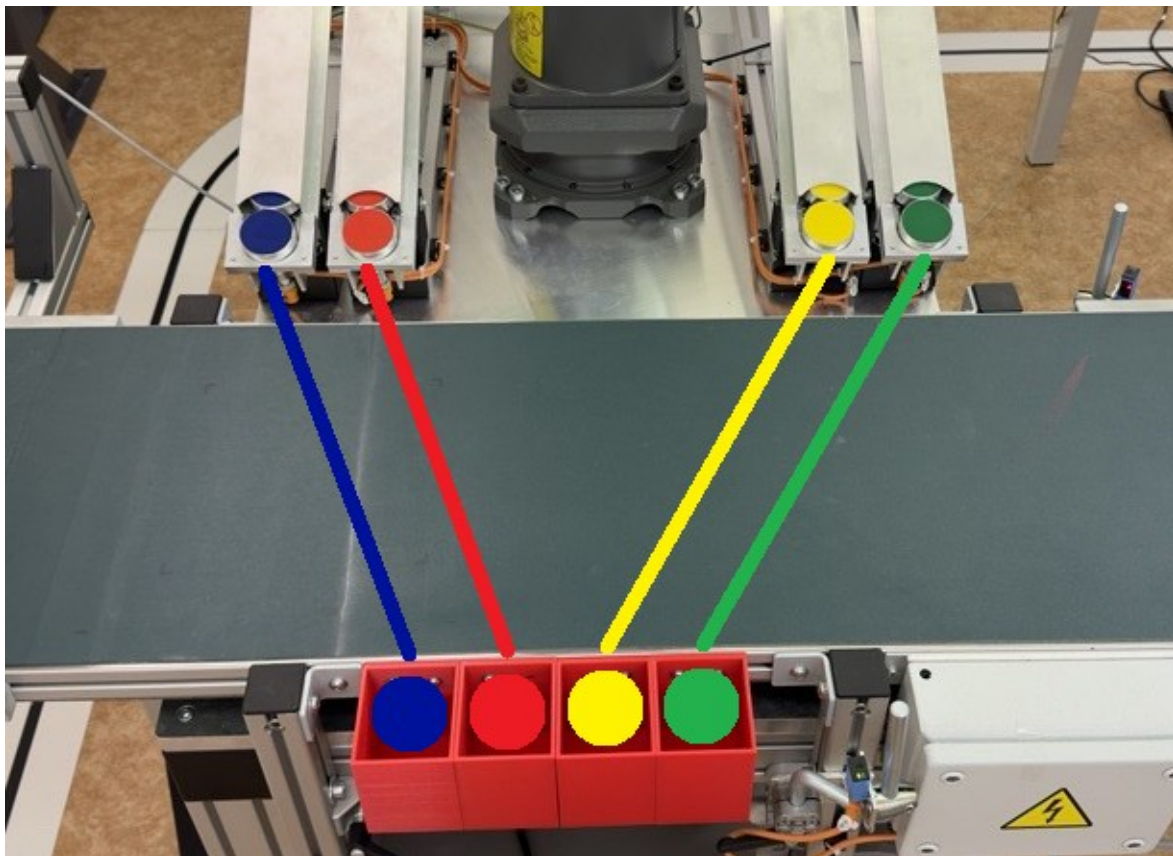
Časový průběh celé implementace společně s kontaktováním týmu Robo@FIT je shrnut v tabulce 1. Celkově lze říci, že proces implementace a zkoušení byl náročný, ale spolupráce s týmem Robo@FIT a postupné řešení vzniklých problémů vedlo k úspěšnému zajištění komunikace mezi aplikací AREditor, řídicí jednotkou robota Fanuc a prostředím rozšířené reality na tabletu. Další práce by měla směřovat k detailnímu prozkoumání zdrojového kódu platformy ARCOR2, aby bylo možné identifikovat příčinu chyby, která se projevila při praktické zkoušce ovládání robota. Po objevení této chyby by již bylo možné plně ovládat robota a začít pracovat na programu pro manipulaci s objekty. Tento krok by znamenal dosažení cíle této práce a implementace vývojové platformy ARCOR2 na pracovišti by byla úspěšná.

4.3 Použití Python programu pro ovládání kolaborativního robota

V předchozím experimentu byl popsán proces implementace vývojové platformy ARCOR2, která nedosáhla očekávaných výsledků. Toto zjištění vedlo k návrhu alternativního přístupu k ovládání robota. Nový přístup využívá existující knihovnu fanucpy, která je již nahrána do řídicí jednotky robota, a to pomocí programu napsaného v programovacím jazyce Python.

4.3.1 Popis experimentu

Experiment byl zaměřen na vytvoření programu pro ovládání kolaborativního robota za účelem manipulace s pukami. Hlavním cílem bylo vytvořit efektivní systém, který umožní přesné a spolehlivé přemístování puků z podavače do cílových pozic v souladu s požadavky uživatele.



Obrázek 4-22 - Pozice puků pro Python program

Podavač disponuje pukami následujících barev: modrá, červená, žlutá a zelená. Tyto pukové budou přesouvány robotem do boxů seřazených vedle sebe, kde jsou pozice barevně seřazené ve stejném pořadí jako v podavači (Obrázek 4-22). Uživatel na vstupu nejprve zvolí, jakou barvu puků a jejich počet požaduje, následně robot postupně uchopí, zdvihne a přemístí puk do přesně určených pozic v boxech. Každý puk je umístěn na místo odpovídající jeho barvě.

4.3.2 Popis programu

Program pro ovládání robota je napsán v jazyce Python a využívá knihovnu fanucpy pro komunikaci s řídicí jednotkou robota.

Připojení k robotu

Nejprve se program připojí k robotu pomocí knihovny fanucpy (Obrázek 4-23). Je specifikován jeho model, IP adresa, port, typ a počet digitálních výstupů pro řízení end-effectoru (chapaďla).

```
from fanucpy import Robot
import numpy as np

robot = Robot(
    robot_model="Fanuc",
    host="192.168.1.3",
    port=18735,
    ee_DO_type="RDO",
    ee_DO_num=7,
)

robot.connect()
```

Obrázek 4-23 - Připojení k robotu

Definice pozic pro jednotlivé barvy

Pro každou barvu puku jsou definovány různé pozice, které má robot dosáhnout při manipulaci. Tyto souřadnice byly zjištěny manuálním přesunutím ramene robota do potřebných umístění a zahrnují odsazené pozice pro bezpečné přiblížení a oddálení ramene, pozice pro uchopení puku a jeho puštění do boxů. Příklad jedné z pěti definovaných pozic pro jednotlivé barvy je možné pozorovat na obrázku 4-24. Čísla zobrazují natočení kloubů ramene robota.

```
positions = {
    "modrá": [-39.163, 46.794, -75.978, 22.688, -34.079, 17.703],
    "červená": [-30.026, 44.13, -81.768, 19.603, -29.268, 10.476],
    "zelená": [37.932, 45.992, -76.978, -27.129, -32.528, -13.145],
    "žlutá": [28.332, 44.048, -82.736, -24.556, -28.089, -5.173]
}
```

Obrázek 4-24 - Definice pozic pro jednotlivé barvy

Funkce pro ovládání robota

Hlavní funkcí programu je „robot_control(colors)“, která přijímá slovník obsahující počet puků pro každou zadanou barvu, a řídí pohyby robota při manipulaci s pukem podle zadaných požadavků uživatele. Pozice, do kterých se robot pohybuje, jsou pevně definovány pro každou barvu puku a jsou převzaty z definic v kódu programu. Postup manipulace je následující:

1. **Nastavení výchozí pozice:** Nejprve je robot přemístěn do výchozí pozice, aby byl připraven k manipulaci s pukem.
2. **Iterace přes zadané barvy puků:** Pro každou zadanou barvu puku se provedou následující kroky:
 - **Otevření chapaďla:** Chapaďlo je otevřeno, aby bylo připraveno k uchopení puku.
 - **Pohyb do odsazené pozice pro puk:** Rameno se přesune do pozice určené pro přiblížení. Tato pozice je definována jako offset od výchozí pozice.
 - **Pohyb do pozice pro samotný puk:** Rameno se přesune do pozice, kde je puk vybrané barvy umístěn.
 - **Uzavření chapaďla:** Chapaďlo je uzavřeno, aby uchopilo puk.

- **Pohyb robota po uchopení:** Rameno se vertikálně přesune do odsazené pozice, aby nedošlo k uvolnění puku z chapadla při kolizi s konstrukcí podavače.
 - **Pohyb do odsazené pozice pro uvolnění puku:** Po zdvihnutí je rameno přesunuto do odsazené pozice, kde má být puk spuštěn. Tato pozice je opět definována jako offset od výchozí pozice.
 - **Pohyb do cílové pozice pro uvolnění puku:** Nakonec je robot přesunut do cílové pozice, kde má být puk umístěn.
 - **Otevření chapadla:** Po umístění puku do cílové pozice je chapadlo opět otevřeno pro uvolnění puku.
3. **Návrat do výchozí pozice:** Po dokončení manipulace s pukem je robot vrácen zpět do výchozí pozice, aby byl připraven na další úlohy nebo na ukončení programu.

Celý proces manipulace s pukem je demonstrován na obrázku 4-25, kde je kód funkce „robot_control(colors)“.

```
def robot_control(colors):
    robot.move("joint", vals=[-12.811, 11.873, 1.079, -1.353, -91.074, 12.558],
              velocity=100, acceleration=100, cnt_val=0, linear=False) # Příkaz pro posun ramene
    for colors, number in colors.items():
        for i in range(number):
            robot.gripper(False) # Otevřít chapadlo
            positionsoffset = positions_offset[colors]
            robot.move("joint", vals=positionsoffset, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            positions = positions[colors]
            robot.move("joint", vals=positions, velocity=100, acceleration=100, cnt_val=0, linear=False)
            robot.gripper(True) # Uzavřít chapadlo
            positionsgrip = positions_grip[colors]
            robot.move("joint", vals=positionsgrip, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            targetpositionsoffset = target_positions_offset[colors]
            robot.move("joint", vals=targetpositionsoffset, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            targetpositions = target_positions[colors]
            robot.move("joint", vals=targetpositions, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            robot.gripper(False)
    robot.move("joint", vals=[-12.811, 11.873, 1.079, -1.353, -91.074, 12.558], velocity=100,
              acceleration=100, cnt_val=0, linear=False)
```

Obrázek 4-25 - Funkce pro ovládání robota

Metoda pro zadání počtu puků jednotlivých barev

Metoda „numbers_entry()“ je navržena pro interaktivní zadání počtu puků jednotlivých barev uživatelem. Po spuštění metoda uživatele nejprve vyzve, aby zadával barvy puků oddělené čárkou. Poté program iteruje přes tyto zadané barvy, v každé iteraci kontroluje platnost zadané barvy, a následně požaduje počet puků dané barvy od uživatele. Pokud je počet puků v platném rozsahu (0 až 5), je uložen do slovníku barvy, kde klíčem je barva puku a hodnotou je počet puků dané barvy. Pokud uživatel zadá neplatnou barvu, počet či jiný datový typ, je zobrazena chybová zpráva, a je opět umožněno zadat platnou hodnotu. Po dokončení zadávání puků jednotlivých barev je uživatel dotázán, zda chce zadat další barvy a počty puků.

Pokud odpoví "ne", cyklus se ukončí a slovník „barvy“ s počty puků je vrácen jako výstup metody. Tento slovník poté slouží jako vstup pro hlavní funkci programu, která řídí manipulaci s puky.

Na obrázku 4-26 z příkazového řádku lze pozorovat interakci mezi uživatelem a programem během zadávání barev a odpovídajících počtů puků i s ošetřenými chybně zadanými datovými vstupy.

```
Zadejte barvy puků oddělené čárkou (modrá, červená, žlutá, zelená): fialová
Zadali jste neplatnou barvu. Zadejte jednu z: modrá, červená, žlutá, zelená
Chcete zadat další počty puků? (ano/ne): ano
Zadejte barvy puků oddělené čárkou (modrá, červená, žlutá, zelená): modrá, žlutá
Zadejte počet puků pro modrá (0-5ks): 8
Zadali jste nesprávné množství puků, postupujte dle instrukcí.
Zadejte počet puků pro modrá (0-5ks): dva
Zadali jste neplatný vstup, zadejte číslo.
Zadejte počet puků pro modrá (0-5ks): 1
Zadejte počet puků pro žlutá (0-5ks): 2
Chcete zadat další počty puků? (ano/ne): ne
```

Obrázek 4-26 - Uživatelský vstup programu

Následuje obrázek 4-27 s kompletním kódem metody „numbers_entry“.

```
def numbers_entry():
    colors = {}
    while True:
        colors_input = input("Zadejte barvy puků oddělené čárkou (modrá, červená, žlutá, zelená): ")
        colors_input = colors_input.split(",")
        for colors_input in colors_input:
            colors_input = colors_input.strip().lower()
            if colors_input in positions:
                while True:
                    try:
                        number = int(input(f"Zadejte počet puků pro {colors_input} (0-5ks): "))
                        if 0 <= number <= 5:
                            colors[colors_input] = number
                            break
                    except ValueError:
                        print("Zadali jste nesprávné množství puků, postupujte dle instrukcí.")
                else:
                    print("Zadali jste neplatný vstup, zadejte číslo.")
            else:
                print("Zadali jste neplatnou barvu. Zadejte jednu z: ", ", ".join(list(positions.keys())))

        continue = input("Chcete zadat další barvy a počty puků? (ano/ne): ").lower()
        if continue != "ano":
            break
    return colors
```

Obrázek 4-27 - Metoda pro zadání počtu puků jednotlivých barev

Spuštění hlavního programu

Tato část kódu obsahuje metodu „main()“, která je zodpovědná za spuštění hlavního programu. Program neustále opakuje proces zadávání počtů puků jednotlivých barev a jejich manipulaci pomocí robota, dokud uživatel nezvolí ukončení. Metoda „main()“ nejprve volá metodu „numbers_entry()“, aby získala informace o počtech puků jednotlivých barev od uživatele. Poté předá získané informace o počtech puků jako argument do hlavní funkce programu „robot_control(colors)“, která řídí pohyby robota při manipulaci s pukem. Po dokončení manipulace s pukem je uživatel dotázán, zda chce zadat další počty puků. Pokud odpoví "ne", program se ukončí, a zobrazí se zpráva o ukončení. Pokud odpoví "ano", cyklus se opakuje a uživatel je znovu vyzván k zadání počtů puků pro další manipulaci.

Následuje blok kódu, který zajišťuje spuštění hlavního programu pouze tehdy, pokud se tento skript spouští přímo. Tímto způsobem se zajišťuje, že kód v metodě „main()“ bude proveden pouze tehdy, když je tento skript spuštěn jako samostatný program. Celý tento proces je znázorněn na obrázku 4-28.

```
def main():
    while True:
        colors = numbers_entry()

        robot_control(colors)

        continue = input("Chcete zadat další počty puků? (ano/ne): ").lower()
        if continue != "ano":
            print("Ukončuje se program.")
            break
if __name__ == "__main__":
    main()
```

Obrázek 4-28 - Spuštění hlavního programu

4.3.3 Testování programu

Testování bylo provedeno za účelem ověření správného chování a funkcionality implementovaného programu pro ovládání robota.

Počítač byl nejprve připojen k robotu pomocí ethernetového kabelu s nastavením IP adresy jako u praktické zkoušky implementace platformy ARCOR2. První spuštění programu bylo provedeno s nižší rychlostí a přidanými pauzami mezi jednotlivými pohyby robota pro bezpečnou kontrolu a rychlé zastavení robota v případě potřeby. Při této fázi testování byl identifikován a vyřešen problém s definicí pozic pro jednotlivé barvy po zdvihu, který způsoboval, že pukky vypadávaly z chapadla z důvodu nárazu do konstrukce podavače. Obrázek 4-29 zachycuje okamžik, kdy robot úspěšně uchopil modrý puk chapadlem, a vytahuje ho z podavače na odsazenou pozici.



Obrázek 4-29 - Pozice po uchopení puku

Po provedených úpravách se pokračovalo v testování, postupně byla zvyšována rychlost robota a zkracovaly se pauzy mezi pohyby. V průběhu testování byly úspěšně vyzkoušeny různé varianty počtu puků i barev. Na obrázku 4-30 je zachycen moment, kdy je již robot v pozici pro uvolnění modrého puku na určené místo v boxu.



Obrázek 4-30 - Pozice pro uvolnění puku

Po úspěšném provedení sérií testů s postupně zvyšující se rychlostí a optimalizací časových pauz mezi jednotlivými pohyby robota bylo prověřeno, že implementovaný program pro ovládání kolaborativního robota splňuje požadavky na přesnost a spolehlivost manipulace s pukem. Díky pečlivé definici pozic pro jednotlivé barvy puků bylo dosaženo bezchybného provedení manipulačních úkolů, přičemž robot spolehlivě přemísťoval puk do cílových pozic v souladu s požadavky uživatele.

4.3.4 Možnosti dalšího rozvoje a optimalizace programu

Implementace tohoto programu není závislá na složitých a obtížně pochopitelných procesech. Příkazy definovanými v programu lze snadno měnit chování robota a přizpůsobovat ho specifickým potřebám a požadavkům dané aplikace. Tímto způsobem lze program dále optimalizovat a přizpůsobit novým požadavkům nebo změnám v pracovním prostředí.

Další možnosti zlepšení programu zahrnují implementaci pokročilejších algoritmů pro optimalizaci pohybů robota, například algoritmů plánování trajektorií, které by mohly zkrátit dobu cyklu a zvýšit efektivitu manipulace. Rovněž by bylo možné rozšířit funkcionalitu programu o detekci a vyhodnocování různých situací, jako je detekce překážek nebo chybových stavů pro zvýšení spolehlivosti a bezpečnosti provozu. Pro komplexnější řízení a koordinaci v rámci výrobního prostředí by mohlo být dalším krokem propojení programu s dalšími systémy nebo zařízeními, jako jsou snímače FANUC Vision Sensor nebo FANUC 3D Area Sensor, které umožňují lokalizovat a rozpoznat objekty.

Navíc by bylo dobré zaměřit se na optimalizaci uživatelského rozhraní programu pro snadnější ovládání a sledování stavu manipulace s puky. Intuitivní uživatelské rozhraní by zvýšilo efektivitu práce s robotem a snížilo riziko uživatelských chyb. Vzhledem k tomu, že ARCOR2 využívá stejný programovací jazyk a příkazy pro ovládání robota, by bylo vhodné moci celý ručně napsaný kód programu přenést do aplikace AREditor, a umožnit uživatelům sledovat celý proces manipulace v rozšířené realitě a přímo v aplikaci provádět další úpravy.

Celkově lze tedy konstatovat, že i přes úspěšnou implementaci existuje stále potenciál pro další vylepšení a optimalizaci programu, která by přispěla k jeho ještě širšímu a efektivnějšímu využití v praxi.

Závěr

Cílem této bakalářské práce bylo implementovat a následně využít mobilní platformu rozšířené reality ARCOR2 pro zjednodušené programování kolaborativního robota od společnosti Fanuc, který je součástí Katedry průmyslového inženýrství a managementu na Fakultě strojní Západočeské univerzity v Plzni. Vedle samotné implementace práce také zjišťovala možné výzvy a problémy spojené s implementací a využitím této technologie, a ověřovala funkčnost ARCOR2 v reálných pracovních podmínkách.

Teoretická část práce představuje definici a historii robotiky, zkoumá její vývoj a využití v různých odvětvích. Dále analyzuje konkrétní prvky průmyslových robotů, jako je jejich pohybový systém, koncový efektor, pohon a řídicí systém. Popisuje rozdělení a charakteristiku modelů průmyslových robotů vyráběných společnostmi ABB. Další kapitola se zaměřuje na rozšířenou realitu, její historii, vývoj a aktuální trendy, včetně inovativních produktů, jako jsou brýle pro smíšenou realitu Vision Pro od společnosti Apple. Závěr rešerše se soustředí na praktické uplatnění rozšířené reality v průmyslových a robotických aplikacích.

V průběhu experimentu byla detailně analyzována architektura platformy ARCOR2, včetně klíčových principů a využití nástroje Docker pro spuštění serveru. Dále byla zkoumána mobilní aplikace AREditor pro interakci s roboty a intuitivní programování jejich pohybů. Práce pokračovala implementací platformy ARCOR2 na pracovišti, které bylo podrobně popsáno včetně jeho vybavení.

Virtuální zkouška ověřila funkčnost serveru a aplikace AREditor. Během této zkoušky byl 3D model robota zobrazen v reálném prostředí, a bylo umožněno testovat ovládání robota skrze prostředí rozšířené reality. Kalibrace robota byla provedena úspěšně a robot reagoval přesně na zadávané příkazy. Tato část experimentu potvrdila schopnost implementovaného systému pracovat v reálném prostředí.

Po úspěšné virtuální zkoušce následovalo nezbytné nahrání knihovny fanucpy do řídicí jednotky robota. Díky této implementaci je robot schopný přijímat příkazy napsané v programovacím jazyce Python, který je používán aplikací AREditor k ovládání robota. Tímto způsobem je také možné ovládat robota na pracovišti pouze s využitím programu v Pythonu.

Implementace mobilní rozšířené reality prostřednictvím platformy ARCOR2 přinesla několik technických výzev, avšak ta v praktické části se ukázala jako neřešitelná v rámci časových a zdrojových možností této práce. Praktická zkouška ovládání robota skrze prostředí rozšířené reality ukázala nedostatky v komunikaci mezi aplikací AREditor a řídicí jednotkou robota. I přes pečlivou analýzu a spolupráci s týmem Robo@FIT se nepodařilo identifikovat přesnou příčinu chyby, která vedla k neúspěšnému pohybu robota a k výskytu chybového hlášení "Catastrophic system error". Dle vypsané chyby v pendantu robota bylo zjištěno, že byly do řídicí jednotky vyslány neinicializované příkazy k pohybu, které nejsou v programovacím jazyce robota KAREL definovány, a nebylo proto možné ramenem robota pohnout.

Tato komplikace vedla k hledání alternativního přístupu pro ovládání robota. Díky již zavedené knihovně fanucpy bylo využito stejných ovládacích příkazů, které využívá aplikace AREditor, pro tvorbu programu pro manipulaci s pukem. Hlavním cílem bylo vytvořit efektivní systém, který umožní přesné a spolehlivé přemísťování puků z podavače do cílových pozic v souladu s požadavky uživatele. Samotný program pracuje tak, že nejprve uživatel zvolí barvu puků a jejich počet. Poté robot postupně uchopí, zdvihne a přemístí puk do přesně definovaných pozic v boxech, přičemž každý puk je umístěn na místo odpovídající jeho barvě. Během testování programu byla zvyšována rychlost robota a postupně se zkracovaly pauzy mezi jednotlivými pohyby. Implementovaný program byl schopen úspěšně zpracovat různé

varianty počtu puků i barev. Robot spolehlivě a přesně přemísťoval puk z podavače do cílových pozic v souladu s požadavky uživatele.

Navzdory neúspěšné implementaci je důležité zdůraznit, že proces implementace přinesl cenné poznatky a zkušenosti, které lze využít v budoucí práci na vývoji a zdokonalování platformy ARCOR2. Spolupráce s týmem Robo@FIT a postupné řešení technických výzev ukázalo schopnost adaptace a odhalilo potenciál platformy. Vzhledem k náročnosti a časovým omezením této práce není možné detailně prozkoumat zdrojový kód platformy ARCOR2 a identifikovat příčinu problému. Budoucí výzkum by měl proto směřovat k podrobnější analýze a testování platformy za účelem odhalení a odstranění technických nedostatků pro plné využití potenciálu této inovativní technologie v praxi.

Seznam použitých zdrojů

- [1] YASAR, Kinza. Robotics. *TechTarget* [online]. 2023, 12.7.2023 [cit. 2023-11-01]. Dostupné z: <https://www.techtarget.com/whatis/definition/robotics>
- [2] MATARIĆ, Maja J. *The robotics primer*. Cambridge: MIT Press, c2007. ISBN 978-026-2633-543.
- [3] BIRK, Andreas. What is Robotics? An interdisciplinary field is getting even more diverse. *IEEE Robotics & Automation Magazine* [online]. 2011, (1), 3 [cit. 2023-11-02]. Dostupné z: doi:10.1109
- [4] HUNT, V. Daniel. *Understanding Robotics*. Academic Press, 1990. ISBN 0124333699.
- [5] VOLNÁ, Eva a Martin KOTYRBA. ROBOTNIK. *Umělá inteligence*. Ostravská univerzita v Ostravě, 2013. ISBN 978-80-7464-330-9.
- [6] YANKO DESIGN [online]. [cit. 2023-11-02]. Dostupné z: <https://www.yankodesign.com/2021/03/30/this-triple-eye-lidars-robot-vacuum-cleaner-uses-advanced-ai-to-navigate-and-clean-your-home/>
- [7] ASIMOV, Isaac. *Já, robot*. Praha: Triton, c2012. Trifid (Triton). ISBN 978-80-7387-491-9.
- [8] CYBERNETICZOO.COM. *1948 – ELSIE (ELECTRO-MECHANICAL ROBOT, LIGHT SENSITIVE WITH INTERNAL AND EXTERNAL STABILITY) – W. GREY WALTER* [online]. 2009 [cit. 2023-11-03]. Dostupné z: <https://cyberneticzoo.com/cyberneticanimals/elsie-cyberneticanimals/elsie/>
- [9] ROBOTNIK. *History of robots and robotics* [online]. 2021 [cit. 2023-11-03]. Dostupné z: <https://robotnik.eu/history-of-robots-and-robotics/>
- [10] SMITHSONIAN INSTITUTION. *"Tortoise" Mobile Robot* [online]. [cit. 2023-11-03]. Dostupné z: https://americanhistory.si.edu/collections/search/object/nmah_879329
- [11] BRYSON, Mark. CAPITALX. *The Rise of Robots* [online]. [cit. 2023-11-05]. Dostupné z: <https://capitalxpartners.co/land/the-rise-of-robotics-2mrg2-m5mgy>
- [12] IFR. *World Robotics 2023 Report: Asia ahead of Europe and the Americas* [online]. 2023 [cit. 2023-11-05]. Dostupné z: <https://ifr.org/ifr-press-releases/news/world-robotics-2023-report-asia-ahead-of-europe-and-the-americas>
- [13] PRABHU, TL. *ROBOTICS ENGINEERING: Basic to Advanced Concepts of Robotics Engineering*. Nestfame Creations Pvt., 2021. B099FGWXML.
- [14] INTUITIVE. *Da Vinci Xi* [online]. [cit. 2023-11-04]. Dostupné z: <https://www.intuitive.com/en-us/products-and-services/da-vinci/xi>
- [15] BETTAROE ROBOTICS. *X1 PRO* [online]. [cit. 2023-11-04]. Dostupné z: <https://www.bettaroe.cz/x1-pro/>
- [16] URC SYSTEMS. *TALON 5* [online]. [cit. 2023-11-04]. Dostupné z: <https://www.urc-systems.cz/en/product/talon-5/>

- [17] ACKERMAN, Evan. MEET AQUANAUT, THE UNDERWATER TRANSFORMER. *IEEE Spectrum* [online]. 2019 [cit. 2023-11-05]. Dostupné z: <https://spectrum.ieee.org/meet-aquanaut-the-underwater-transformer>
- [18] SKAŘUPA, Jiří. *Průmyslové roboty a manipulátory*. Ostrava: Vysoká škola báňská - Technická univerzita, [2008]. ISBN 978-80-248-1522-0.
- [19] VITRALAB. *Automatizační a robotická technika* [online]. Košice: Leonardo da Vinci Programme LLP/LDV/TOI/2009/SK/93100530, 2001 [cit. 2023-11-12]. Dostupné z: http://www.sjf.tuke.sk/vitralab/files/vystupy/prirucka_sk_final.pdf
- [20] KAMINSKI, Joe. MRO ELECTRIC AND SUPPLY. *KUKA Robot Arms* [online]. 2022 [cit. 2023-11-10]. Dostupné z: <https://www.mroelectric.com/blog/kuka-robot-arm/>
- [21] NOVOTNÝ, František. *Úvod do automatizace a robotizace ve strojírenství*. Liberec: Technická univerzita v Liberci, 2020. ISBN 978-80-7494-545-8.
- [22] KUKA. *KR 360 FORTEC* [online]. [cit. 2023-11-10]. Dostupné z: <https://www.kuka.com/cs-cz/produkty,-slu%5%beby/robotick%3%a9-syst%3%a9my/pr%5%afmyslov%3%a9-roboty/kr-360-fortec>
- [23] RUMÍŠEK, Pavel. *AUTOMATIZACE (roboty a manipulátory)* [online]. Brno: Vysoké učení technické v Brně, 2003 [cit. 2023-11-11]. Dostupné z: https://ust.fme.vutbr.cz/tvareni/img/opory/emm_mechanizace_a_automatizace_roboty_rumisek.pdf
- [24] FANUC. *Tablet Teach Pendant for Industrial Robots* [online]. [cit. 2023-11-12]. Dostupné z: <https://www.fanucamerica.com/products/robots/controllers/tablet-teach-pendant>
- [25] ABB. *Naše nabídka* [online]. [cit. 2023-11-13]. Dostupné z: <https://new.abb.com/products/robotics/cs/roboty>
- [26] FURHT, Borko. *Handbook of Augmented Reality*. Springer New York Dordrecht Heidelberg London, c2011. ISBN 978-1-4614-0063-9.
- [27] MICROSOFT. Co je rozšířená realita, neboli AR? *Dynamics 365 Guides* [online]. [cit. 2023-11-14]. Dostupné z: <https://dynamics.microsoft.com/cs-cz/mixed-reality/guides/what-is-augmented-reality-ar/>
- [28] SYED, T.A., M.S. SIDDIQUI, H.B. ABDULLAH, S. JAN, A. NAMOUN, A. ALZAHIRANI, A. NADEEM a A.B. ALKHODRE. *ENCYCLOPEDIA. Tracking Technology in Augmented Reality* [online]. [cit. 2023-11-17]. Dostupné z: <https://encyclopedia.pub/entry/39656>
- [29] APPLE. *Apple Vision Pro* [online]. 2023 [cit. 2023-11-16]. Dostupné z: <https://www.apple.com/apple-vision-pro/>
- [30] APPLE. *Přichází vývojářské nástroje pro prostorové zážitky s Apple Vision Pro* [online]. 2023 [cit. 2023-11-16]. Dostupné z: <https://www.apple.com/cz/newsroom/2023/06/developer-tools-to-create-spatial-experiences-for-apple-vision-pro-now-available/>

- [31] RICE UNIVERSITY. *Augmented Reality Solutions in Engineering* [online]. 2022 [cit. 2023-11-24]. Dostupné z: <https://engineering.rice.edu/academics/graduate-programs/online-meml/blog/augmented-reality-in-engineering>
- [32] LIGHTGUIDE. *Manufacturing's Unexpected Solution to the Great Resignation: Digital Transformation* [online]. 2022 [cit. 2023-11-25]. Dostupné z: <https://www.lightguidesys.com/resource-center/blog/manufacturings-unexpected-solution-to-the-great-resignation-digital-transformation>
- [33] DE PACE, Francesco, Federico MANURI a Andrea SANNA. Augmented Reality in Industry 4.0. *AMERICAN JOURNAL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY* [online]. 2018, 7 [cit. 2023-11-24]. ISSN 2349-3917. Dostupné z: doi:10.21767/2349-3917.100017
- [34] HOLOLIGHT SPACE. *BMW Accelerates Concept Evaluations* [online]. 2020 [cit. 2023-11-25]. Dostupné z: <https://hololight.com/bmw-uses-ares-in-prototyping/>
- [35] PORSCHE NEWSROOM. *Augmented Reality Usage Triples in Porsche Workshops Amid COVID-19* [online]. 2020 [cit. 2023-11-25]. Dostupné z: https://newsroom.porsche.com/en_US/technology/porsche-tech-live-look-augmented-reality-usage-covid-19-20702.html
- [36] HASPINGER, Florian. HOLO-LIGHT. *How Augmented Reality Is Changing Engineering* [online]. 2020 [cit. 2023-11-24]. Dostupné z: <https://www.ien.eu/article/how-augmented-reality-is-changing-engineering/>
- [37] AV ENGINEERING. *Přehled softwarových platforem pro IoT - AR* [online]. 2023 [cit. 2023-11-25]. Dostupné z: <https://www.aveng.cz/software-services/>
- [38] ABB. *ABB offers Augmented Reality on a smartphone to simplify robot installations* [online]. 2020 [cit. 2023-11-25]. Dostupné z: <https://new.abb.com/news/detail/66541/ar-smartphone-robot-installations>
- [39] ABB. *Augmented Reality app helps Volvo unlock post-pandemic productivity* [online]. 2022 [cit. 2023-11-25]. Dostupné z: <https://new.abb.com/news/detail/89801/cstmr-augmented-reality-app-helps-volvo-unlock-post-pandemic-productivity>
- [40] PTC REALITY LAB. *Kinetic AR* [online]. 2020 [cit. 2023-11-26]. Dostupné z: <https://www.ptc.com/en/about/reality-lab/portfolio/research/kinetic-ar>
- [41] FUSTE, Anna, Ben REYNOLDS, James HOBIN a Valentin HEUN. *Kinetic AR: A Framework for Robotic Motion Systems in Spatial Computing* [online]. 8 [cit. 2023-11-26]. Dostupné z: doi:<https://doi.org/10.1145/3334480.3382814>
- [42] ROBO@FIT. *Arcor2_areditor* [online]. [2019] [cit. 2024-04-09]. Dostupné z: https://github.com/robofit/arcor2_areditor
- [43] KAPINUS, Michal, Zdeněk MATERNA, Daniel BAMBUŠEK, Vítězslav BERAN a Pavel SMRŽ. *ARCOR2: Framework for Collaborative End-User Management of Industrial Robotic Workplaces using Augmented Reality*. [online]. 2023, 27 [cit. 2024-04-09]. Dostupné z: https://www.researchgate.net/publication/371605881_ARCOR2_Framework_for_Colla

borative_End-
User_Management_of_Industrial_Robotic_Workplaces_using_Augmented_Reality

- [44] DOCKER DOCS. *Docker Compose overview* [online]. [2013] [cit. 2024-05-12]. Dostupné z: <https://docs.docker.com/compose/>
- [45] FANUC. *Kolaborativní robot CR-7iA a CR-7iA/L* [online]. [cit. 2023-11-13]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/str%C3%A1nka-filtru-robot%C5%AF/kolaborativn%C3%AD-roboty/collaborative-cr7ial>
- [46] ROBO@FIT. *Fanucpy* [online]. [2019] [cit. 2024-04-09]. Dostupné z: <https://github.com/torayeff/fanucpy>
- [47] EVERYTHING ABOUT ROBOTS!. *Fanuc INTP Alarm Codes* [online]. 2024 [cit. 2024-04-09]. Dostupné z: <https://everythingaboutrobots.com/fanuc-intp-alarm-codes/>
- [48] KHANDELWAL, Renu. MEDIUM. *Marker-based Augmented Reality using OpenCV*. [online]. 2022 [cit. 2023-11-17]. Dostupné z: <https://arshren.medium.com/marker-based-augmented-reality-using-opencv-b851b82be4dc>

Seznam příloh

Příloha 1: Zdrojový kód programu pro přesouvání puků

PŘÍLOHA č. 1

Zdrojový kód programu pro přesouvání puků

```
from fanucpy import Robot
import numpy as np

# Připojení k robotu
robot = Robot(
    robot_model="Fanuc",
    host="192.168.1.3",
    port=18735,
    ee_DO_type="RDO",
    ee_DO_num=7,
)

robot.connect()

# Definice pozic
positions_offset = {
    "modrá": [-36.677, 30.723, -57.13, 13.471, -50.102, 25.997],
    "červená": [-27.891, 26.56, -60.689, 10.467, -47.567, 19.123],
    "zelená": [35.483, 30.041, -57.728, -16.289, -48.587, -23.382],
    "žlutá": [25.988, 25.885, -61.25, -13.195, -46.414, -15.847]
}

positions = {
    "modrá": [-39.163, 46.794, -75.978, 22.688, -34.079, 17.703],
    "červená": [-30.026, 44.13, -81.768, 19.603, -29.268, 10.476],
    "zelená": [37.932, 45.992, -76.978, -27.129, -32.528, -13.145],
    "žlutá": [28.332, 44.048, -82.736, -24.556, -28.089, -5.173]
}

positions_grip = {
    "modrá": [-39.044, 34.688, -65.189, 17.872, -44.177, 23.406],
    "červená": [-30.026, 34.229, -73.385, 15.718, -37.262, 15.112],
    "zelená": [37.932, 37.491, -69.871, -22.945, -38.974, -18.295],
    "žlutá": [28.332, 35.037, -75.199, -19.91, -35.073, -10.617]
}

target_positions_offset = {
    "modrá": [-3.118, 56.993, -35.048, -0.82, -55.548, 2.931],
    "červená": [-3.118, 56.993, -35.048, -0.82, -55.548, 2.931],
    "zelená": [-3.118, 56.993, -35.048, -0.82, -55.548, 2.931],
    "žlutá": [-3.118, 56.993, -35.048, -0.82, -55.548, 2.931]
}

target_positions = {
    "modrá": [-10.718, 70.068, -39.816, -0.762, -50.867, 10.551],
    "červená": [-5.798, 68.8, -41.35, -0.853, -49.283, 5.709],
    "zelená": [4.271, 68.614, -41.587, -0.644, -49.179, 18.461],
    "žlutá": [-0.767, 68.31, -41.949, -0.931, -48.628, 0.735]
}
```

```
# Funkce pro ovládání robota
def robot_control(colors):
    robot.move("joint", vals=[-12.811, 11.873, 1.079, -1.353, -91.074, 12.558],
              velocity=100, acceleration=100, cnt_val=0, linear=False)
    for colors, number in colors.items():
        for i in range(number):
            robot.gripper(False) # Otevřít gripper
            positionsoffset = positions_offset[colors]
            robot.move("joint", vals=positionsoffset, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            positions = positions[colors]
            robot.move("joint", vals=positions, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            robot.gripper(True) # Uzavřít gripper
            positionsgrip = positions_grip[colors]
            robot.move("joint", vals= positionsgrip, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            positionsgripoffset = positions_grip_offset[colors]
            robot.move("joint", vals= positionsgripoffset, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            targetpositions = target_positions[colors]
            robot.move("joint", vals=targetpositions, velocity=100, acceleration=100, cnt_val=0,
                      linear=False)
            robot.gripper(False)
    robot.move("joint", vals=[-12.811, 11.873, 1.079, -1.353, -91.074, 12.558], velocity=100,
              acceleration=100, cnt_val=0, linear=False)
```

```
# Metoda pro zadání počtu puků jednotlivých barev
def numbers_entry():
    colors = {}
    while True:
        colors_input = input("Zadejte barvy puků oddělené čárkou (modrá, červená, žlutá, zelená): ")
        .split(",")
        for colors_input in colors_input:
            colors_input = colors_input.strip().lower()
            if colors_input in positions:
                while True:
                    try:
                        number = int(input("Zadejte počet puků pro { colors_input } (0-5ks): "))
                        if 0 <= number <= 5:
                            colors[colors_input] = number
                            break
                    except ValueError:
                        print("Zadali jste nesprávné množství puků, postupujte dle instrukcí.")
                else:
                    print("Zadali jste neplatný vstup, zadejte číslo.")
            else:
                print("Zadali jste neplatnou barvu. Zadejte jednu z: ", ", ".join(list(positions.keys())))

        continue = input("Chcete zadat další barvy a počty puků? (ano/ne): ").lower()
        if continue != "ano":
            break

    return colors

# Metoda pro spuštění hlavního programu
def main():
    while True:
        colors = colors_input()

        robot_control(colors)

        continue = input("Chcete zadat další počty puků? (ano/ne): ").lower()
        if continue != "ano":
            print("Ukončuje se program.")
            break

if __name__ == "__main__":
    main()
```