

Univerzita Jana Evangelisty Purkyně
v Ústí nad Labem
Přírodovědecká fakulta



Návrh a realizace řízení robotického ramene
pro výukové účely

BAKALÁŘSKÁ PRÁCE

Vypracoval: Lukáš Bílek

Vedoucí práce: Ing. Petr Haberzettl

Studijní program: Aplikovaná informatika

Studijní obor: Informační systémy

ÚSTÍ NAD LABEM 2023

Namísto žlutých stránek vložte digitálně podepsané zadání kvalifikační práce poskytnuté vedoucím katedry.

Zadání musí zaujímat právě dvě strany.

Zadání je nutno vložit jako PDF pomocí některého nástroje, který umožňuje editaci dokumentů (se zachováním elektronického podpisu).

V Linuxe lze například použít příkaz pdftk.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a použil jen pramenů, které cituji a uvádím v přiloženém seznamu literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., ve znění zákona č. 81/2005 Sb., autorský zákon, zejména se skutečností, že Univerzita Jana Evangelisty Purkyně v Ústí nad Labem má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Jana Evangelisty Purkyně v Ústí nad Labem oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladu, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

V Ústí nad Labem dne 24. dubna 2023

Podpis:

Děkuji vedoucímu práce Ing. Petrovi Haberzettlovi
za neocenitelné rady a pomoc při tvorbě bakalářské práce.
Dále děkuji Ing. Vítu Vaněčkovi za praktické rady
ohledně kinematiky robotů.

Abstrakt:

Tato práce se zaměřuje na návrh a realizaci řídicího systému pro ovládání robotického ramene Tinkerkit Braccio s využitím platformy Arduino Uno. Součástí je i realizace fyzického modelu robotického pracoviště s hardwarovými prvky. V teoretické části práce je nejprve popsána základní teorie robotických manipulátorů a důležité pojmy z robotiky. Dále jsou popsány základy kinematiky robotů a pohybové struktury. Nakonec je představen použitý hardware. Praktická část práce se pak zabývá řešením inverzní kinematické úlohy ramene s pěti rotačními klouby, řešením linearizace pohybu ramene a implementací těchto řešení do ovládacích funkcí robotu. Dále jsou popsány algoritmy a uživatelské instrukce, které tyto algoritmy využívají. Z algoritmů a instrukcí jsou dále sestaveny demonstrační úlohy. Práce implementuje i vlastní vyvinutý hardware (3D tištěný světelný maják) a bezpečnostní prvek (nouzové tlačítko). Nakonec jsou k robotickému pracovišti vytvořeny manuály pro obsluhu a popis demonstračních úloh.

Klíčová slova: Robotické rameno, Arduino Uno, Inverzní kinematická úloha, Linearita pohybu, Model robotického pracoviště

DESIGN AND IMPLEMENTATION OF ROBOTIC ARM CONTROL FOR TEACHING PURPOSES

Abstract: The aim of this thesis is to design and develop a control system for controlling the Tinkerkit Braccio robotic arm using the Arduino Uno board. It also includes the created model of the robotic workstation with hardware elements. In the theoretical part of the thesis, the basic theory of robotic manipulators and important concepts of robotics are first described. Then the basics of robot kinematics and motion structures are described. Finally, the hardware used is introduced. The practical part of the thesis presents the solution of the inverse kinematic of a robotic arm with five rotary joints, then the solution of the linear arm motion is presented, and the implementation of these solutions in the robot control functions is shown. Furthermore, the algorithms and user instructions are described. Sample tasks are also created from the developed instructions. The thesis also implements self-developed hardware (3D printed light beacon) and a safety feature (emergency stop button). Lastly, operational manuals for the robotic workstation are created.

Keywords: Robotic arm, Arduino Uno, Inverse kinematics, Linear motion, Robotic arm workstation model

Obsah

Úvod	13
TEORETICKÁ ČÁST	14
1. Automatizace a robotizace	15
1.1. Roboty a manipulátory	15
1.2. Průmyslové roboty	17
1.3. Pohony robotů	18
1.3.1. Elektrické pohony	18
1.3.2. Pneumatické a hydraulické pohony	19
1.4. Kinematika robotů	20
1.4.1. Přímá a inverzní kinematická úloha	21
2. Použitý hardware	23
2.1. Arduino UNO	23
2.1.1. Mikrokontroler Atmel	23
2.1.2. Vstupně-výstupní porty a převodníky	23
2.1.3. Napájení	24
2.2. Robotické rameno Tinkerkit Braccio	26
2.3. Servomotory	27
2.3.1. Princip činnosti	27
2.3.2. Řízení servomotoru	28
2.4. Servodriver	29
PRAKTICKÁ ČÁST	31
3. Pracovní deska	33
4. Řídicí systém	35
4.1. Linearita pohybu	35
4.2. Inverzní kinematika	38
4.2.1. Výpočet polohy bodů	38
4.2.2. Výpočet úhlu q_0	44
4.2.3. Výpočet úhlů $q_1 - q_3$	45
4.2.4. Převod úhlů do Braccio systému	48

4.3. Funkce firmwaru	48
4.3.1. G10 a G11 - Ovládání gripperu	49
4.3.2. G92 - Nastavení aktuální pozice os	50
4.3.3. G28 - Autohome	51
4.3.4. G90 - Lineární pohyb IKU	51
4.3.5. G101 – G105 - Ukázkové úlohy	54
5. Přídavné periferie	57
5.1. Světelný maják	57
5.1.1. Návrh konstrukce	57
5.1.2. Elektronické zapojení	58
5.1.3. Řízení funkce majáku	59
5.2. Tlačítko CENTRAL–STOP	60
5.2.1. Elektronické zapojení	60
5.2.2. Řízení funkce tlačítka	60
6. Závěr	63
A. Externí přílohy	69

Úvod

Robotika je oblast, která se v posledních letech rychle rozvíjí a nabízí stále více možností využití v průmyslu, medicíně a mnohdy už i ve všedním životě. Robotická ramena jsou významnou součástí robotických systémů a jejich použití se rozšiřuje do mnoha různých odvětví. Se stále větším rozmachem robotiky je tak logickým krokem i její popularizace a zařazení do výuky. Robotické systémy v sobě skloubí znalosti z různých odvětví, od mechaniky a elektroniky přes materiálové inženýrství až po matematiku, fyziku a další. Na takových systémech lze prakticky demonstrovat vyučované teoretické znalosti.

Cílem této práce je vytvořit řídicí systém pro ovládání robotického ramene Tinkerkit Braccio na open-source platformě Arduino pro výukové účely. Zároveň je součástí práce i navržení a vyrobení celého simulovaného robotického pracoviště se souřadnicovým systémem a signalizačními a bezpečnostními prvky. Součástí je uživatelský manuál pro obsluhu ramene. Model robotického pracoviště je vybavený instrukční sadou a komunikací skrz USB sběrnici a je ho tak možné jednoduše řídit každým notebookem či počítačem vybaveným tímto rozhraním a konzolovou aplikací. Díky intuitivnímu ovládání, lehké manipulaci a uživatelské podpoře v podobě manuálu je model pracoviště ideální pro prezentaci široké veřejnosti na veletrzích či jiných veřejných akcích. Zároveň slouží jako ideální demonstrační pomůcka pro výuku elektroniky a robotiky.

Práce je členěna do dvou základních celků, teoretické a praktické části. Teoretická část popisuje základní znalosti robotiky, pohonů robotů a pohybu v prostoru (kinematiky). Popisuje také hardware, který byl využit při realizaci práce. Praktická část pak popisuje návrh a realizaci samotného řídicího systému a jeho algoritmů. Součástí je i řešení inverzní kinematické úlohy robotu, linearizace pohybu a implementace těchto řešení do řídicího systému robotu. Z vytvořených algoritmů a instrukcí jsou sestaveny demonstrační úlohy, které v praxi ukazují robotické rameno v průběhu různých pracovních úkonů, jako je například přesun nákladu mezi stanovišti či stavba věže. Praktická část dále popisuje návrh a realizaci signalizačního světelného majáku vyrobeného technologií 3D tisku a zároveň, tak jako v reálném provozu, implementaci bezpečnostních prvků ve formě nouzového tlačítka. V neposlední řadě jsou pak součástí bakalářské práce manuály pro obsluhu celého pracoviště. Součástí manuálů je i detailní popis k přípravě jednotlivých demonstračních úloh a jejich průběh, aby osoba ovládající robotické rameno dokázala připravit požadovaný scénář pro každou demonstrační úlohu.

TEORETICKÁ ČÁST

Teoretická část bakalářské práce uvádí čtenáře do problematiky průmyslové výroby, průmyslových robotů a manipulátorů ve výrobním procesu a průmyslové automatizace. Představuje také konkrétní použitý hardware, jež je následně využit v části praktické. Informace zde dostupné se opírají o uvedené citované zdroje a mé vlastní znalosti z předchozího studia Automatizace a počítačových aplikací elektrotechnického programu na střední průmyslové škole.

1. Automatizace a robotizace

Světová populace má stále rostoucí trend a spolu s ní se tak zvyšuje tlak na produktivitu a kvalitu výroby. Stále zvyšující se poptávku po výrobcích ovšem není možné uspokojit pouze lidskou pracovní silou. Lidská práce je pomalá, finančně nákladná a mnohdy časově neefektivní. Monotónní pracovní úkony také neodpovídají lidskému naturelu a silná konkurence na trhu navíc nutí výrobce pružně reagovat na měnící se trendy. Důležitým aspektem je samozřejmě i ekonomika podniků a s ní související snaha o snižování výrobních nákladů a maximalizování zisků.

Všechny tyto aspekty naplňuje automatizace výrobních procesů. Ta, ať již částečná či úplná, nahrazuje člověka ve výrobním procesu všude tam, kde je člověk omezujícím faktorem nebo pracuje v nebezpečném prostředí, které ohrožuje jeho zdraví. Zavádění tzv. robotizovaných pracovišť má za následek zvýšení produktivity práce, snížení výrobních nákladů, úsporu pracovních sil, snížení zdravotních rizik a v neposlední řadě zvýšení kultury lidské práce. [1] V rámci těchto automatizovaných pracovišť či celých výrobních linek jsou často používány i manipulátory a roboty. [2]

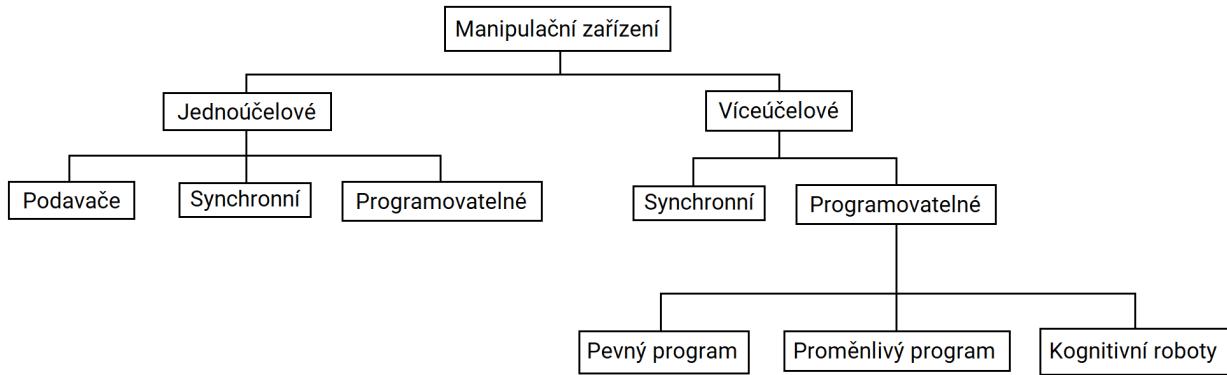
Automatizaci nelze brát pouze jako vlastní jeden stroj. Jedná se o komplexní proces zahrnující dopravu materiálu, polotovarů a výrobků, kontrolu a měření, výměnu nástrojů apod. Samotné řešení automatizace výrobních procesů vždy závisí na konkrétním typu výroby, využívané prostředky se totiž velmi liší. Pro automatizaci kusové a malosériové výroby se použijí převážně pružné automatizační výrobní systémy. Jsou to takové systémy, které nabízí snadnou výměnu programu s malými náklady. Naopak pro automatizaci velkosériové a hromadné výroby se použijí systémy tzv. tvrdé automatizace. Jsou to takové systémy, u kterých je program automatického technického zařízení vyroben přímo na míru vyráběnému výrobku a program nelze změnit vůbec nebo jen velmi nákladně. Takové výrobní automaty sice nejsou flexibilní, nabízí ale nejvyšší efektivitu výroby. Vedle samotného technického řešení je ovšem také nutné zohlednit i další hlediska, jako je například rychlosť dodávky konkrétního požadovaného systému, přesnost výroby konkrétního systému, prostorovou náročnost technického vybavení, energetickou náročnost, další výdaje spojené s údržbou atd. [2, 3]

1.1. Roboty a manipulátory

Robotizace průmyslové výroby je významným prvkem při automatizaci výroby. Manipulátory a roboty zastoupí člověka v monotónních a fyzicky náročných pracovních úkonech. Tím zvedají

kvalitu práce a umožňují lidem zaměřit se na důležitější pracovní úkony. Taková manipulační zařízení jsou navíc univerzální a kromě snížení provozních nákladů a zvýšení bezpečnosti a produktivity výroby jsou zároveň natolik univerzální, že je možné je bez velkých výdajů použít zrovna tam, kde jsou potřeba.[2]

Průmyslové roboty a manipulátory lze dle funkce, možnosti řízení, míry autonomnosti atd. rozdělit do kategorií uvedených v obr. 1.1.



Obrázek 1.1.: Rozdělení robotů a manipulátorů [2]

Obecně lze roboty a manipulátory rozdělit na 2 základní skupiny: - jednoúčelové - víceúčelové.

Jednoúčelové roboty a manipulátory jsou charakteristické omezenými pohybovými možnostmi, které jsou přímo na míru přizpůsobeny danému pracovnímu úkonu. Úroveň řízení taktéž odpovídá dané aplikaci, je na míru přizpůsobena danému úkonu. Konstrukční provedení a použité pohonních jednotek jsou použity v souladu s použitou technologií u obsluhovaného zařízení, aby byla zajištěna vzájemná kompatibilita. Víceúčelové roboty a manipulátory jsou pak přesným opakem a vyznačují se univerzálním použitím a schopností přizpůsobit se různým používaným technologiím. Mimo toto základní rozdělení lze roboty a manipulátory rozdělit i podle jejich typů. [2]

Podavače - Jedná se o nejjednodušší jednoúčelový manipulátor vytvořený pro jeden typ pracovní operace. Pracují společně s ovládaným strojem jako jedno zařízení a jejich funkcí je například podávat materiál ke zpracování, manipulovat s vyráběným polotovarem a nebo přemístit obroběný či již vyrobený kus mimo stroj.

Synchronní teleoperátory - Tato zařízení slouží jako "prodloužené ruce" pracovníka, ovládá ho tedy člověk a ne stroj. Používá se všude tam, kde je třeba snížit fyzickou námahu daného pracovního úkonu, pracovní prostředí vyžaduje přísne hygienické normy nebo je pracovní prostředí pro člověka toxicke apod.

Programovatelné - Programovatelná zařízení jsou řízena řídící jednotkou s pevným či proměnlivým programem. Pevný program se nemění během činnosti a stále dokola opakuje jednu a tu samou činnost. Proměnlivý program dokáže reagovat na vnější vlivy a podle nich přepínat chod programu nebo přenastavovat jeho parametry, dokáže se adaptovat aktuální situaci.[2]

Kognitivní roboty - Tyto roboty dokáží na základě vnímání okolního světa skrz sadu senzorů reagovat na své prostředí, řešit nepředvídané události v prostředí, samostatně vykonávat činnosti v prostředí atd. Je vybaven schopností "myslet". [4]

1.2. Průmyslové roboty

Průmyslový robot je zařízení mnohem složitější než manipulátor. Robot je schopný vykonávat autonomní interakce s prostředím kolem něj dle instrukcí, které mu zadá člověk ať už přímo a nebo ve formě programu. V tomto prostředí robot dokáže manipulovat s předměty a nebo se v něm sám pohybovat a vykonávat nějakou výrobní operaci, např. svařování, lakování, lepení atd. [2]



Obrázek 1.2.: Průmyslové roboty KUKA [5]

Průmyslové roboty lze obecně dělit do 5 generací.

Nultá generace robotů zpravidla nemá žádnou zpětnou vazbu. Robot se tedy pohybuje v prostředí pouze dle předem naprogramovaného programu. Případné změny v prostředí pracovní oblasti nebo poruchy vedou k zablokování dalšího pracovního kroku a centrálního zastavení zařízení. Potom je nutná interakce obsluhy či údržby.

První generace robotů již disponuje lehkými prvky zpětné vazby. Robot je tedy schopný na případné změny reagovat přepínáním mezi několika podprogramy. Tyto podprogramy jsou ovšem předem naprogramovány člověkem.

Druhá generace robotů je schopná vybrat z předem naprogramovaných programů optimální program. Robot je schopný optimalizace výroby podle zadaných kritérií

Třetí generace robotů je schopná se učit. Dokáže samostatně tvořit program na základě zkušeností, které nabyl v provozu. Na začátku robot přijme pouze cíl činnosti a způsob, jakým k tomuto cíli robot dojde, je ponechán na inteligenci řídícího systému robotu. Systém sám program vytvoří a robot dle něj pak požadovanou úlohu vykoná.

Čtvrtá generace robotů je již čistě autonomní. Roboty mají schopnost sociální interakce, chovají se podobně jako člověk a samostatně si volí, jaké cíle práce budou vykonávat a jakým způsobem.

V dnešní době je možné se v průmyslových a výrobních závodech setkat s robota nulté a první generace, jejich příkladem je série průmyslových robotů KUKA na obr. 1.2. Robota dalších generací jsou cenově velmi nákladné a spolu s nimi je nutné výrobní linku vybavit také odpovídajícím množstvím senzorové techniky, která umožní robota potřebnou zpětnou vazbu a úroveň rozpoznání pracovní oblasti robota.[2]

1.3. Pohony robotů

Klíčovou vlastností robotů je pohyb v prostoru. Pro dosažení pohybu robota využíváme pohony - zařízení, která přeměňují vstupní energii (elektrickou, pneumatickou, hydraulickou) na mechanický pohyb. Samotný pohon je složen z motoru, zařízení pro ovládání vstupní energie a spojovacím blokem. Vstupní energie nejdříve projde ovládacím členem a podle potřeby se reguluje. Následně je v motoru přeměněna na energii pohybovou a vystoupí ven přes spojovací blok, díky kterému se pohybová energie přenese na výstup pohybové jednotky. Podle potřeby se mezi výstup z motoru a spojovací blok zařazují i převodové bloky, převodovky. Ty mohou být mechanické, magnetické nebo elektrické.[2]

Na pohony robotů jsou kladené nejrůznější požadavky. Pohyb by například měl být plynulý. Díky plynulému pohybu je při pohybu robota potřeba menší úchopná síla pro držení přenášeného předmětu. Absence rázů v průběhu pohybu také prodlužuje životnost konstrukce robota, dochází k menšímu mechanickému namáhání jednotlivých částí.

Důležitou vlastností je také vysoká přesnost polohování. Robot totiž musí cílovou polohu dosáhnout s co největší přesností. Pomůže k tomu například zapojení zpětné vazby a tedy schopnost pohonu detekovat, zda se již nachází v požadované poloze.

Polohová tuhost je také důležitá, rozumí se tím schopnost pohonu po dosažení koncové polohy tuto polohu udržet. Zároveň musí pohon po dosažení polohy odolat do určité míry působení vnějších sil. Různé motory mají různé tuhosti. Vysokou tuhostí disponují např. hydraulické motory, které využívají zablokování hydraulické kapaliny v motoru. Naproti tomu minimální tuhostí disponují elektromotory a pneumatické motory, v jejich případě je nutné zařadit do pohonu např. brzdu.

V neposlední řadě je kladen velký důraz na minimální hmotnost a rozměry pohonu. Hmotnost pohonu ovlivňuje celkovou hmotnost robota. Jsou-li pohony navíc umístěny přímo na pohyblivých částech robota, ovlivňuje jejich hmotnost přímo dynamické chování a energetickou náročnost celé konstrukce.[2]

1.3.1. Elektrické pohony

Elektromotory, ať už stejnosměrné nebo střídavé, patří mezi nejpoužívanější typy pohonů pro robota střední nosnosti. Jejich výhodou je snadná dostupnost elektrické energie a jednoduché vedení od zdroje energie k motoru. Vodič elektrické energie je uživatelsky mnohem přívětivější

než pneumatická či hydraulická hadice. Elektromotor navíc není náročný na údržbu a jeho provoz je čistý. Výhodou je také nižší hlučnost, menší nároky na chlazení a nižší pořizovací, provozní a udržovací náklady. Nevýhodou je naopak požadované kvalitní provedení motoru a nebezpečí úrazu elektrickým proudem.

Elektromotory obecně rozdělujeme do dvou základních skupin: stejnosměrné a střídavé. Kromě typu napájení motory dělíme také podle typu výstupní pohybové energie na motory s rotačním výstupem a s přímočarým výstupem. Příklad rotačního elektromotoru je na obr. 1.3 [2].



Obrázek 1.3.: Elektrické krokové motory s rotačním výstupem [6]

1.3.2. Pneumatické a hydraulické pohony

Pneumatické a hydraulické pohony spojuje jeden typ napájecího média – tekutina (obr. 1.4).

Hydraulické pohony využívají jako provozní médium minerální oleje. Hodí se pro použití v zařízeních větších výkonů, nevadí totiž jejich přetížení. Jejich výhodou je velká tuhost, možnost plynulé regulace a velká účinnost. Nevýhodou je naopak obtížné dosažení vyšších pohybových rychlostí, závislost viskozity kapaliny na teplotě a možná hořlavost některých provozních kapalin. Nutností je také olej k pohonu přivést a odvést zpět ke zdroji, pro každý pohon tak jsou třeba dvě hadice.[2]

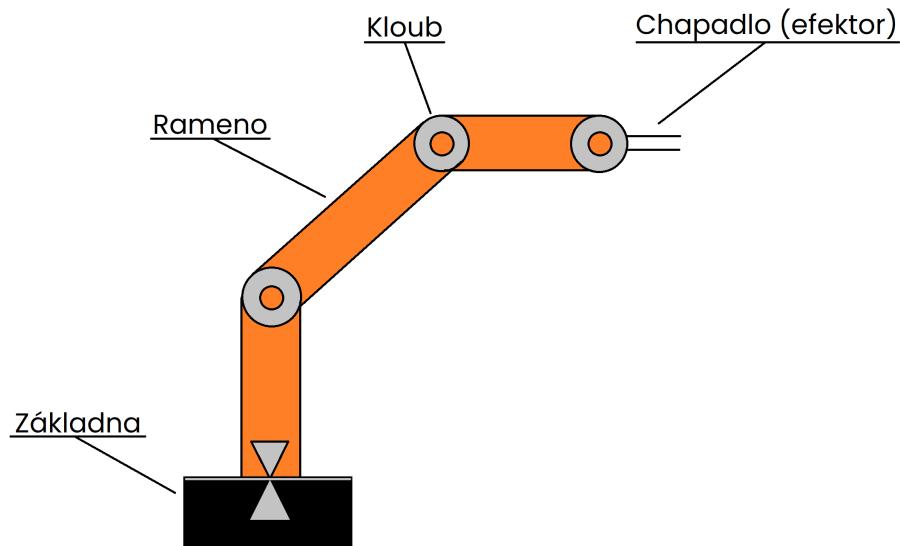


Obrázek 1.4.: Pneumatické motory s přímočarým výstupem [7]

Pneumatické pohony využívají jako provozní médium stlačený vzduch. Ten je nutné před použitím upravit. Vzduch je nutné pomocí filtrů zbavit prachových a jiných částic a následně i vlhkosti. K tomu lze využít kondenzační, membránové nebo adsorpční sušičky.[8] Mezi výhody pneumatických pohonů patří možnost připojení na centrální rozvod stlačeného vzduchu, kdy není potřeba zpětný odvod vzduchu zpět ke zdroji. Dále je výhodou rychlosť pohybu pohonů a možnost činnosti ve velkém tepelném rozsahu a v provozech, kde hrozí nebezpečí výbuchu nebo vznícení. Naopak nevýhodou je obtížné udržení plynulého pohybu při malých rychlostech, složité mazání pohyblivých prvků a nutnost výroby a úpravy stlačeného vzduchu.[2]

1.4. Kinematika robotů

Pohyb robotu musí být jasně definovaný a pro jeho popis se využívá právě kinematiky. Kinematika popisuje pohyb robotu, tedy jeho trajektorii v pracovním prostoru. Využívá při tom v praxi zavedených pojmuů popsaných [9] na obr. 1.5



Obrázek 1.5.: Části robotu

Rameno je pevná část robotu. Jedná se o tu část robotu, která leží mezi dvěma klouby.

Kloub je část robotu, která spojuje dvě ramena a umožňuje jejich volný nebo řízený pohyb.

Chapadlo (efektor) je koncová část robotu. Díky ní robot dokáže uchopit předmět a přenést ho nebo vykonat požadovaný pracovní úkol (svařovací hlavice, lepící hlavice atd.)

Základna je část robotu pevně spojená se zemí.

Kinematická dvojice je složena ze dvou ramen a kloubu, který je spojuje.[9]

Dalším, důležitým parametrem je tzv. **počet stupňů volnosti**. Jedná se o nezávislý počet pohybů, které může robot vykonat. Rozdělují se na hlavní polohovací systém a vedlejší orientační systém. Na obr. 1.5 je uveden příklad robotu se 4 stupni volnosti. V praxi se nejčastěji používají roboty se 6 stupni volnosti. První 3 stupně zajišťují pohyb v hlavním polohovacím systému a zbylé 3 stupně ve vedlejším orientačním systému.

Stupně volnosti a s nimi související klouby můžeme dále rozdělit na rotační a translační. Rotační kloub umožňuje otáčení ramene kolem osy kloubu. Translační kloub umožňuje přímočarý posuv (translaci) po ose kloubu. Vhodnou kombinací výše zmíněných kloubů **R** (rotační) a **T** (translační) lze sestavit tzv. **kinematický řetězec**. Ten popisuje tipy jednotlivých členů a kinematických dvojic. Jejich příkladem jsou:

Válcová (cylindrická) struktura - RTT - tímto řetězcem lze popsat standardní stavební jeřáb. První osa je rotační, umožňuje jeřábu otáčet se kolem své osy a dosáhnout tak různých míst po staveništi. Druhá osa je translační, jedná se o pojazd pod ramenem jeřábu, na kterém visí

lano s hákem. Třetí osou, také translační, je pak pohyb samotného háku nahoru a dolů pomocí navijáku.[9]

Pravoúhlá (kartézská) struktura - TTT - tímto řetězcem lze popsat portálové či mostové jeřáby ve výrobních halách. První osa je translační, umožňuje pohyb celého jeřábu po výrobní hale tam a zpět. Druhá osa je také translační, umožňuje posun pojazdu pod ramenem jeřábu, na kterém visí lano s hákem. Třetí osa je také translační a jedná se o samotný pohyb lana s hákem nahoru a dolů pomocí navijáku.



Obrázek 1.6.: Portálový jeřáb [10]

Angulární struktura - RRR - tento řetězec je typický pro robotická ramena, jeho příklad je popsán na obr. 1.6. První osa je rotační, umožňuje otáčení robota kolem základny. Druhá osa je také rotační, první rameno je schopno se otočit kolem své osy. Třetí osa je také rotační a umožňuje druhému ramenu se otočit kolem své osy.[9]

1.4.1. Přímá a inverzní kinematická úloha

Pro pohyb robotů v rámci pracovního prostoru se využívá kinematických úloh. Ty řeší popis trajektorie robotu z počátečního do cílového bodu, rychlosť a zrychlení. V rámci kinematiky robotů se používají dva souřadnicové systémy:

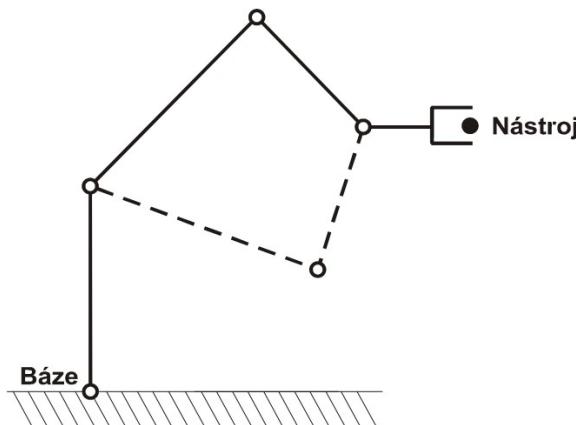
Kloubové souřadnice - souřadnice ve formátu $(\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_n)$ vyjadřují natočení jednotlivých rotačních kloubů ve stupních. Souřadnice ve formátu $(t_0, t_1, t_2, \dots, t_n)$ pak vyjadřují posun translačních kloubů v milimetrech.

Kartézské souřadnice - souřadnice udávající informaci o poloze bodu v prostoru ve formátu (x, y, z) .[2]

Právě převod mezi těmito dvěma soustavami řeší kinematické úlohy.

Přímá kinematická úloha řeší problém, kdy jsou známé kloubové souřadnice a je potřeba z nich určit koncovou polohu efektoru. Přímá úloha je plně řešitelná a její řešení není nijak zvlášť výpočetně náročné. Pro konkrétní kloubové souřadnice existuje vždy pouze jedna výsledná poloha ramene. Lze využít například popis ramene podle Denavit–Hartenbergovi notace, výsledkem je soustava matic a jejich vyřešení vede k určení konečné pozice efektoru. Tento typ úloh ovšem není tak obvyklý.

Inverzní kinematická úloha řeší opačný a mnohem častější problém. Jsou známé konečné kartézské souřadnice efektoru a je na algoritmu, aby spočítal kloubové souřadnice. Tento typ úloh je výrazně obtížnější než úlohy přímé kinematiky a neexistuje žádný jednoduchý algoritmus, který by úlohu uměl vyřešit. V případě inverzní úlohy stojí za zmínku také skutečnost, že úloha může mít jedno řešení, nekonečně mnoho řešení a nebo vyřešit vůbec nelze. V případě řešení úlohy pro koncový bod zadaný mimo pracovní prostor je výsledek jasný, úloha nebude mít řešení. Ovšem ani to, že se zadaný koncový bod nachází v pracovním prostoru, není záruka řešitelnosti dané úlohy. Robot se totiž musí řídit také kloubovými omezeními, která chrání jeho vlastní strukturu před kolizemi rámů do sebe sama. Pro jeden koncový bod totiž existuje nekonečně mnoho způsobů, jak se může rameno natočit, aby daného bodu dosáhlo (viz obr. 1.7) a některé z nich mohou zahrnovat i kolizi s tělem robotu. Je pak na algoritmu, aby takové kolizní varianty natočení eliminoval. Inverzní kinematická úloha dokonce ani nemusí mít analytické řešení. Pokud se tak stane, je nutné přikročit k použití numerických metod, které jsou ovšem velmi výpočetně náročné a zatěžují řídící systém robotu.[11]



Obrázek 1.7.: Ukázka více řešení inverzní úlohy [11]

2. Použitý hardware

Tato kapitola bakalářské práce představuje konkrétní hardware, který byl využit při praktické realizaci práce. Konkrétně se jedná o samotné robotické rameno, řídící systém Arduino a servodriver.

2.1. Arduino UNO

Arduino je celosvětově úspěšný projekt jednodeskových mikropočítačů. Jedná se o prototypovou desku obsahující mikrokontroler, elektroniku potřebnou k provozu mikrokontroleru, napěťové regulační obvody, ochranné obvody pro digitální a analogové vstupy a výstupy, rozhraní pro USB komunikaci a tak dále. Jde tedy o ideální "all in one" zařízení nejen pro domácí projekty. Komponenty jsou popsány na obr. 2.1.

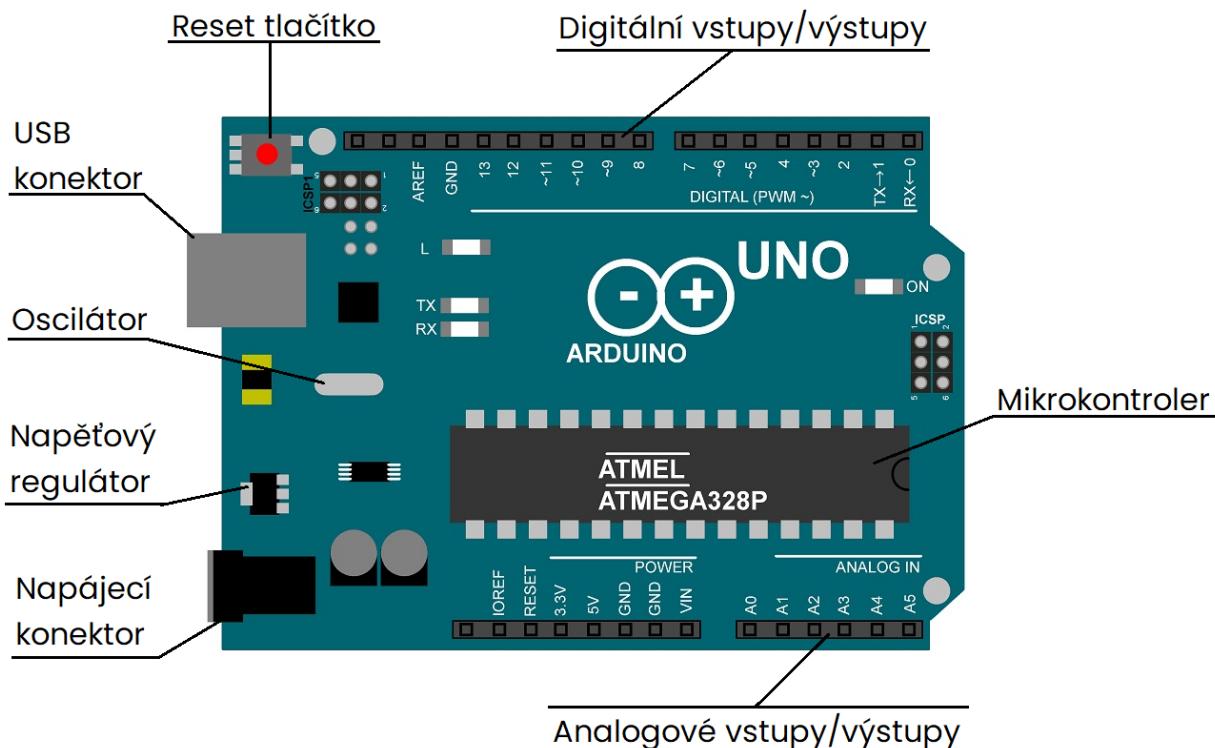
Další výhodou je otevřenost celé platformy, ta je totiž "open source". Na internetu jsou proto dostupné kompletní podklady a schémata zapojení a deska je tak plně modifikovatelná dle potřeb konkrétní aplikace. Arduino má i svůj programovací jazyk *Wiring* založený na jazyce *C++*. Na internetu je v tomto jazyce k dispozici nepřeberné množství návodů i knihoven, které uživatel může využít a implementovat ve svém projektu. [12]

2.1.1. Mikrokontroler Atmel

Srdcem vývojové desky Arduino je mikrokontroler od firmy Atmel. Arduino Uno verze R3 používá mikrokontroler AVR ATMEGA 328P (obr. 2.2). Mikrokontroler vykonává instrukce programu, který zkompilovaný a uložený v jeho paměti. Poskytuje také přístup k *analogově-digitálním převodníkům* (umožňují číst hodnotu elektrického napětí přivedeného na analogové vstupy), k vstupně-výstupním portům, komunikačním sběrnicím *SPI* a *I2C* atd. Chod mikrokontroleru zajišťuje hlavní oscilátor o frekvenci až 20 MHz, který poskytuje clock signál pro chod programu. Všechny vstupy a výstupy z portů mikrokontroleru jsou pak vyvedeny do PIN lišt po obvodu desky, (viz obr. 2.1), aby byly co nejlépe dostupné a daly se k nim připojit vodiče bez nutnosti pájení.[12]

2.1.2. Vstupně-výstupní porty a převodníky

Vstupně-výstupní porty mikrokontroleru slouží pro interakci s vnějším světem. Porty jsou jednotlivě adresovatelné a lze je softwarově nastavit jako vstup nebo výstup. Napěťové hladiny



Obrázek 2.1.: Schéma Arduino Uno, obrázek z [13] doplněn o vlastní popisky



Obrázek 2.2.: AVR ATMEGA 328P [14]

digitálních portů odpovídají binární logice. Logická "0" se zapíše jako 0V a logická "1" jako 5V. Mimo digitálních portů jsou k dispozici i analogové porty, které využívají analogově-digitální převodník a dokáží číst hodnotu elektrického napětí od 0 do 5 voltů. Kromě těchto portů jsou dostupné i speciální porty umožňující pulsně-šířkovou modulaci, externí přerušení, komunikační protokoly (I2C, SPI) atd.[12]

2.1.3. Napájení

Arduino Uno disponuje napájením přes USB port a přes externí napěťový regulátor. USB napájení přijímá stabilní napětí 5V. Arduino lze ovšem napájet i z externího zdroje, k tomu lze využít buď napájecí konektor nebo pin V_{in} po obvodu desky. Obě tato místa jsou vedena do externího napěťového regulátoru umístěného na desce, který na vstupu akceptuje 6V - 20V (doporučeně 7V - 12V) a na výstupu vyrábí stabilních 5V pro chod mikrokontroleru.

Zároveň jsou na desce umístěny piny 5V a 3V3 (3.3V), ze kterých lze vyvést stabilní napájecí napětí pro napájení externích komponent či obvodů. Je ovšem nutné dbát na maximální proudové

zatížení, aby nedošlo k poškození napěťových obvodů.[12]

2.2. Robotické rameno Tinkerkit Braccio

Po důkladném průzkumu trhu bylo pro realizaci bakalářské práce vybráno robotické rameno Tinkerkit Braccio na obr. 2.3. To se svou stavbou a parametry nejvíce přibližuje průmyslovým standardům nejpoužívanějších ramen v průmyslu a výrobě, nabízí poutavý design a zpracování vhodné pro demonstraci pro veřejnost a nabízí dobrý poměr cena-výkon. Informace v této kapitole vycházejí z oficiálního návodu pro Tinkerkit Braccio.



Obrázek 2.3.: Robotické rameno Tinkerkit Braccio [15]

Rameno Tinkerkit je robotické rameno se 6 stupni volnosti angulární struktury RRR. Celý set (viz obr. 2.4) obsahuje:

- otočnou základnu 71.5 mm
- 2 ramena 125 mm a 1 rameno s efektorem 160 mm
- 4 servomotory SR 431 a 2 servomotory SR 311
- desku se servodrivery
- napájecí adaptér 5V 5A
- příslušenství (šroubky, podložky, šroubovák atd.)



Obrázek 2.4.: Obsah balení Tinkerkit Braccio [16]

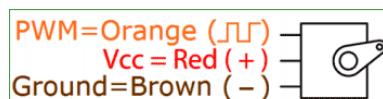
2.3. Servomotory

Servomotor je elektricky rotační pohon, viz obr. 2.5. Jeho výhodou je možnost přesného nastavení požadovaného úhlu natočení osy. Běžné servomotory jsou schopné pohybovat se jen v určitém pracovním rozsahu, typicky od 0° do 180° . Je to proto, že na výstupní hřídel servomotoru je připojen potenciometr, který detekuje úhel natočení a předává tuto informaci řídící jednotce servomotoru. Potenciometry jsou běžně omezené v dráze pohybu, pracují na principu pohybu jezdce po odporové dráze. Čím je vzdálenost jezdce od začátku dráhy větší, tím větší je hodnota elektrického odporu.[12]



Obrázek 2.5.: Servomotor [17]

Servomotory existují jak se stejnosměrným elektromotorem (DC), tak i střídavým elektromotorem (AC). Pro potřeby této práce je dále vysvětlen princip servomotoru se stejnosměrným elektromotorem. Hlavním parametrem je kg/cm . Tento parametr udává, jakou váhu dokáže servomotor zdvihnout v určité vzdálenosti od osy servomotoru. Čím dál od osy zátěž působí, tím menší zátěž servomotor dokáže zdvihnout (princip páky). Nastavení přesné polohy natočení se provádí přivedením pulsně-šírkové modulace na datový vodič servomotoru (viz obr. 2.6). [18].



Obrázek 2.6.: Zapojení servomotoru [18]

2.3.1. Princip činnosti

Servomotor sestává ze 3 funkčních celků:

- Elektromotor
- Senzor
- Zpětnovazební systém

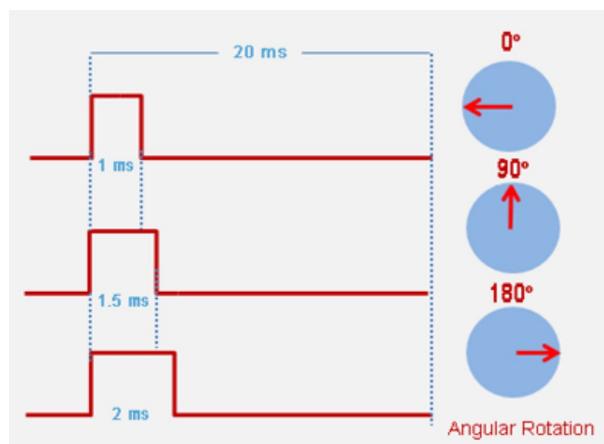
Jedná se o uzavřený systém (někdy též uzavřená regulační smyčka), který využívá zpětnou vazbu na výstupu servomotoru k ovládání pohybu a požadované polohy servomotoru. Zpětnovazební systém porovná požadovaný signál a aktuální signál a rozdíl těchto dvou signálů pak slouží jako vstupní signál pro servomotor. Servomotor se tedy snaží upravovat svou polohu natočení do té doby, dokud je požadovaný signál a aktuální signál rozdílný.

Servomotor fyzicky obsahuje samotný elektromotor, potenciometr, mechanické převody a ovládací obvod. Na samotném začátku je výstup z elektromotoru veden přes mechanické převody. Zde se sníží otáčky motoru a zvýší točivý moment (přidá sílu). Počáteční poloha osy servomotoru je taková, při které na výstupním potenciometru nevzniká žádný signál. Není-li připojen signál požadované polohy, nevzniká žádný rozdílový signál a servomotor se nehýbe.

Při připojení požadovaného signálu na vstup servomotoru servomotor tento signál vezme a porovná ho v zesilovači chyby se signálem aktuální polohy z potenciometru. Signál chyby pak slouží jako vstupní signál pro motor a servomotor se dá do pohybu. Jak se osa servomotoru otáčí, otáčí se i potenciometr na výstupu a mění signál aktuální polohy natočení. Zesilovač chyby neustále tyto dva signály porovnává a upravuje rozdílový signál chyby, probíhá neustálá regulace. V momentě, kdy servomotor dosáhne požadované polohy, výstupní signál z potenciometru je totožný s požadovaným signálem, nevzniká žádný rozdílový signál a rotace se zastaví. Servomotor úspěšně dosáhl požadovaného natočení.[18]

2.3.2. Řízení servomotoru

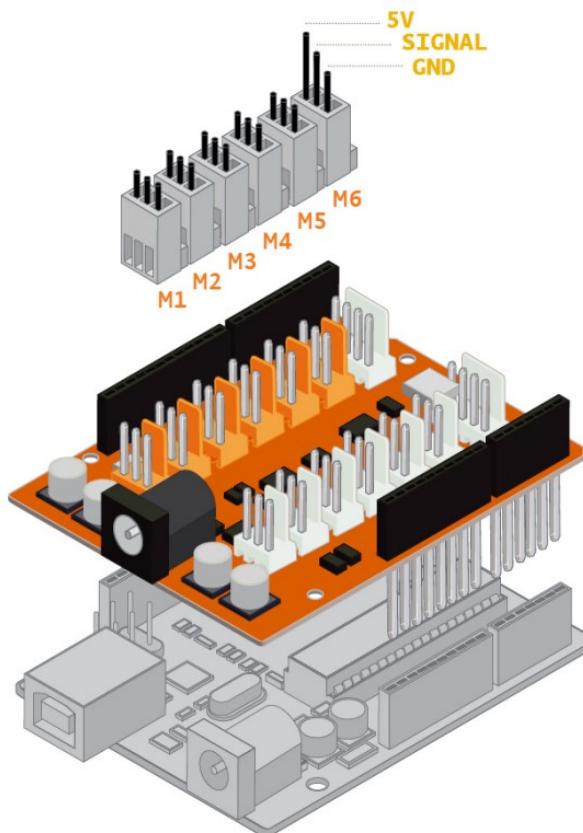
Servotmory mají 3 výstupní vodiče. Dva vodiče slouží jako napájení (+ a - u DC, L a N u AC) a zbyvající vodič je datový pro příjem řídícího signálu. Servotmory jsou řízeny pomocí *Pulsně-šířkové modulace*. Jedná se o speciální typ signálu, kde se mění šířka pulsu. Princip je názorně ukázán na obr. 2.7. Šířka pulsu udává úhel natočení servomotoru. V případě z obr. 2.7 se servomotor při šířce pulsu 1 ms a periodě signálu 20 ms natočí do polohy 0°, při stejné periodě a šířce pulsu 2 ms se servomotor natočí do polohy 180°. Je tedy zřejmé, že perioda čtení signálu je konstantní, mění se pouze šířka pulsu (tj. doba, kdy je hodnota signálu logická "1"- 5 V). [18]



Obrázek 2.7.: Pulsně-šířková modulace a úhel natočení servomotoru [18]

2.4. Servodriver

Servomotor při provozu odebírá poměrně hodně elektrického proudu. V případě, že by byl servomotor připojen přímo na výstupní pin portu mikrokontroleru, došlo by k proudovému přetížení a zničení mikrokontroleru. Z tohoto důvodu je třeba použít desku se servodrivery (obr. 2.8), které poskytují servomotorům externí zdroj napájení s dostatečnou proudovou zatížitelností. V případě ramene Tinkerkit Braccio je externí zdroj schopný při 5V dodat až 4A, tj. 20W výkonu.



Obrázek 2.8.: Servodriver Braccio servoshield, obrázek z oficálního manuálu Tinkerkit Braccio

Kromě externího napájení pro servomotory disponuje Braccio servoshield také funkcí *Soft-Start*. Po zapnutí zdroje napájení servodriver po dobu osmi sekund hlídá případnou kolizi ramene s překážkou

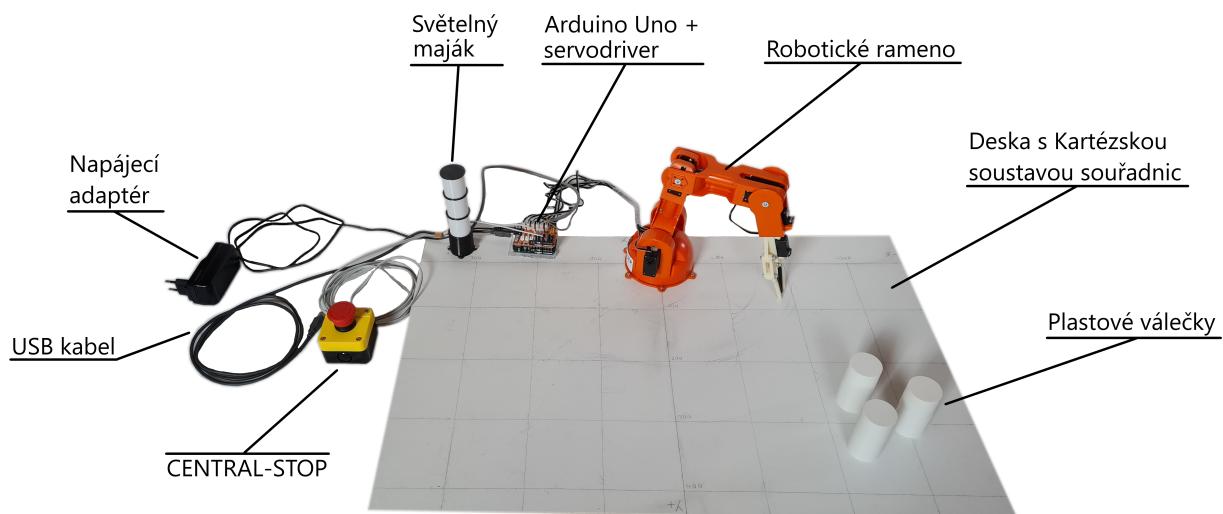
Informace v kapitole 2.4 vycházejí z oficiálního manuálu Tinkerkit Braccio.

PRAKTICKÁ ČÁST

Praktická část bakalářské práce se zabývá vlastní realizací řídicího systému robotického ramene a aplikací kinematických principů. Popisuje vývoj a funkce vlastního firmwaru a vyrobených přídavných periferií.

3. Pracovní deska

Součástí praktické části bakalářské práce není pouze řídicí systém. Byl vytvořen model robotického pracoviště, které se skládá z následujících částí popsaných na obr. 3.1.



Obrázek 3.1.: Robotické stanoviště

Základem je pracovní deska. Na ní je nanesen Kartézský souřadnicový systém, který definuje pracovní oblast robotu. Souřadnicový systém slouží pro navigaci robotu, operátor je dle něj schopný zadat přesně definovanou pozici, ke které má robot následně vypočítat kinematickou úlohu (více v kapitole 4.2) a případně se na ni následně přesunout. Robotické pracoviště je definováno rozsahy:

- osa X: $< -400; 400 >$ [mm]
- osa Y: $< 0; 400 >$ [mm]
- osa Z: $< 0; 520 >$ [mm]

Reálná pracovní oblast je ovšem značně menší, což je dáno omezeným dosahem samotného ramene. Dále se na pracovišti nachází 3D vytiskný světelný maják, který je podrobněji popsán v kapitole 5.1. Implementován je i bezpečnostní prvek ve formě CENTRAL-STOP tlačítka, které je podrobněji popsáno v kapitole 5.2. Zbytek pracoviště tvoří nezbytná kabeláž pro napájení a řízení. Na pracovišti se pro robotické úlohy používají plastové válečky (také 3D tištěné), které simulují náklad pro transport.

4. Řídicí systém

Řídicí systém robotického ramene (firmware) je vytvořen v programovacím jazyce *Wiring* založeném na syntaxi jazyka *C++*. K vývoji lze použít oficiální vývojové prostředí Arduino IDE, v této bakalářské práci bylo ovšem využito vývojové prostředí Microsoft Visual Studio Code s rozšířením *Platformio*. Toto vývojové prostředí lépe zvýrazňuje syntaxi a poskytuje lepší orientaci v delším kódu. Rozšíření *Platformio* poskytuje lehkou instalaci knihoven z online repozitáře a komunikaci s vývojovou deskou Arduino přímo z prostředí Visual Studio Code.

Firmware integruje veškeré periferie, řeší výpočetní části kinematiky a linearity pohybu a zajišťuje komunikaci s ramenem a operátorem. Řízení ramene je navrženo tak, že vlastní ovládání probíhá přes USB sběrnici. Stačí tedy jakýkoliv počítač s USB rozhraním, terminálovou aplikací a znalostí funkcí (Gcode). Obecně chod hlavního programu popisuje diagram na obr. 4.1.

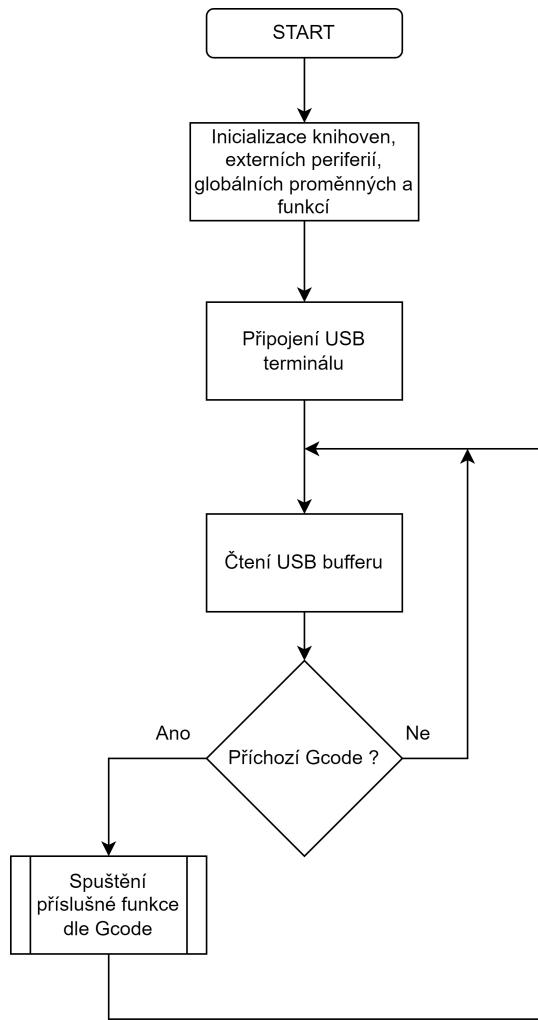
Po zapnutí napájení a připojení pracoviště skrz USB sběrnici se provede inicializační sekvence. Načtou se všechny knihovny, provede se připojení externích periferií a připojení servomotorů samotného ramene. Zároveň Servoshield osm sekund po bootu chrání případné kolize ramene funkcí Soft-Start (popsané v kapitole 2.4). Rameno se nastaví do startovní polohy (vzprímená kolmo vzhůru), dojde k nastavení světelného majáku na světlou zelenou barvu (stav "Stand-by"), které indikuje uživateli připravenost systému, a vyšle se zpráva přes USB sběrnici do terminálu informující uživatele o verzi firmwaru a instrukcích. Nakonec je odeslána výzva pro zadání požadované funkce (Gcode).

Rameno v tomto stavu čeká na příchozí instrukci a je-li platná, vyzve případně uživatele k zadání dalších požadovaných parametrů (souřadnice, úhly) a následně instrukci provede. Není-li instrukce platná, odešle se zpráva o chybně zadaném požadavku a výzva k zadání platné instrukce.

4.1. Linearita pohybu

Pohyb ramene by měl být co možná nejplynulejší. Jak již bylo dříve vysvětleno v kapitole 1.3, plynulý pohyb snižuje potřebnou úchopovou sílu a celkově snižuje mechanické zatížení a opotřebení celé konstrukce ramene, na které nepůsobí pohybové rázy.

Základní algoritmus pohybu ramene dostupný v knihovně Braccio pracuje na principu průběžné inkrementace či dekrementace úhlů o 1° pro každý jednotlivý kloub samostatně do té doby, dokud kloub nedosáhne požadované polohy. Takový pohyb ovšem není lineární, nýbrž sekavý a neplynulý. Nastávají tak situace, kdy například jeden kloub se potřebuje otočit o 10° , druhý o 30° a



Obrázek 4.1.: Vývojový diagram hlavního programu

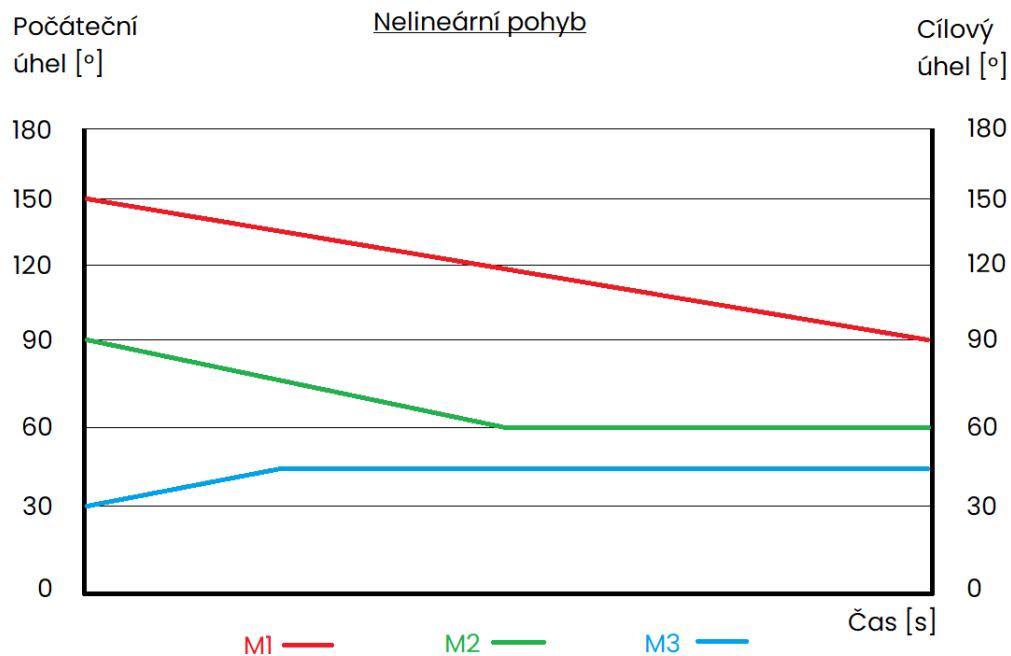
třetí o 90°. Z povahy algoritmu vyplývá, že kloub s nejkratší dráhou, tj. 10°, ukončí svůj pohyb nejdříve, zbylé motory se ovšem stále pohybují, jejich dráha je totiž mnohem delší. Pohyb je tak sekavý a plný pohybových rázů. Tento jev nejlépe popisuje graf na obr. 4.2.

Hlavním problémem tedy je, že základní pohybový algoritmus nezohledňuje délky drah jednotlivých servopohonů. Algoritmus byl tedy značně vylepšen. Nový lineární algoritmus pracuje následovně:

- Algoritmus přijme požadované úhly q_0, q_1, q_2, q_3, q_4 a aktuálně nastavené úhly $step_q_0, step_q_1, step_q_2, step_q_3$ a $step_q_4$
- Proběhne kontrola, zda jsou požadované úhly v pracovní oblasti daných servomotorů. Je to z toho důvodu, aby nedošlo k případné kolizi ramene se sebou samým. Je-li úhel mimo pracovní oblast, přepíše se požadovaná hodnota úhlu na krajní hodnotu intervalu, který původní hodnota překročila.
- Algoritmus určí, který motor musí urazit nejdelší vzdálenost. Proběhne tedy výpočet 4.1

$$\delta q_n = |q_n - step_q_n| \quad n \in \langle 0; 4 \rangle \quad (4.1)$$

a následně je určena hodnota nejdelší dráhy



Obrázek 4.2.: Nelineární pohyb ramene

$$\delta_{max} = \delta_{q_n} \quad (4.2)$$

kde n je index největší hodnoty δ_{q_n}

- Dále jsou určeny inkrementy/dekrementy jednotlivých motorů, viz 4.3.

$$part_{q_n} = \frac{\delta_{q_n}}{\delta_{max}} \quad n \in \{0; 4\} \quad (4.3)$$

Tyto inkrementy/dekrementy ($part_{q_n}$) jsou vypočítány tak, aby jejich počet byl u všech motorů stejný. Algoritmus tím docílí toho, že každý motor se bude pohybovat stejný počet cyklů a tedy, že všechny motory se budou pohybovat stejně dlouhou dobu a jejich pohyb skončí ve stejnou chvíli.

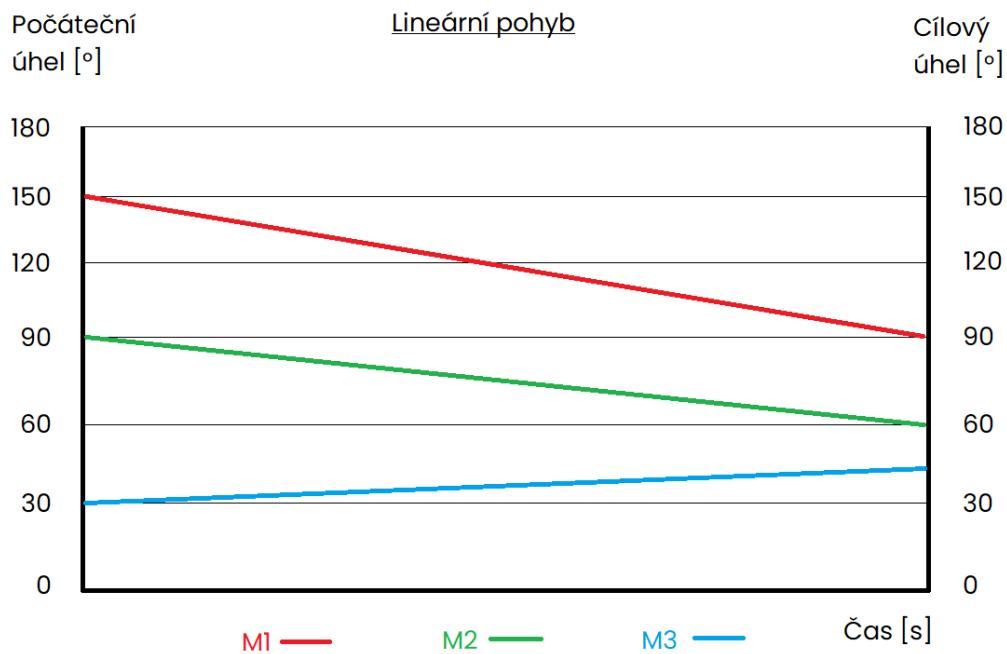
- Algoritmus nyní spustí cyklus s délkou δ_{max} a v každém cyklu každému motoru n přičte ($step_{q_n} < q_n$) nebo odečte ($step_{q_n} > q_n$) příslušný $part_{q_n}$

$$step_{q_n} = step_{q_n} \pm part_{q_n} \quad n \in \{0; 4\} \quad (4.4)$$

Na konci cyklu zapíše algoritmus aktuální hodnotu $step_{q_n}$ příslušnému servomotoru M_n a cyklus se znova opakuje, dokud počet proběhlých cyklů není roven δ_{max} .

Průběh algoritmu je plynulý a bez pohybových rázů. Pohyb motorů dle nového algoritmu ukazuje graf na obr. 4.3.

Praktická aplikace algoritmu včetně vývojového diagramu je podrobně popsána v kapitole 4.3.2.



Obrázek 4.3.: Lineární pohyb ramene

4.2. Inverzní kinematika

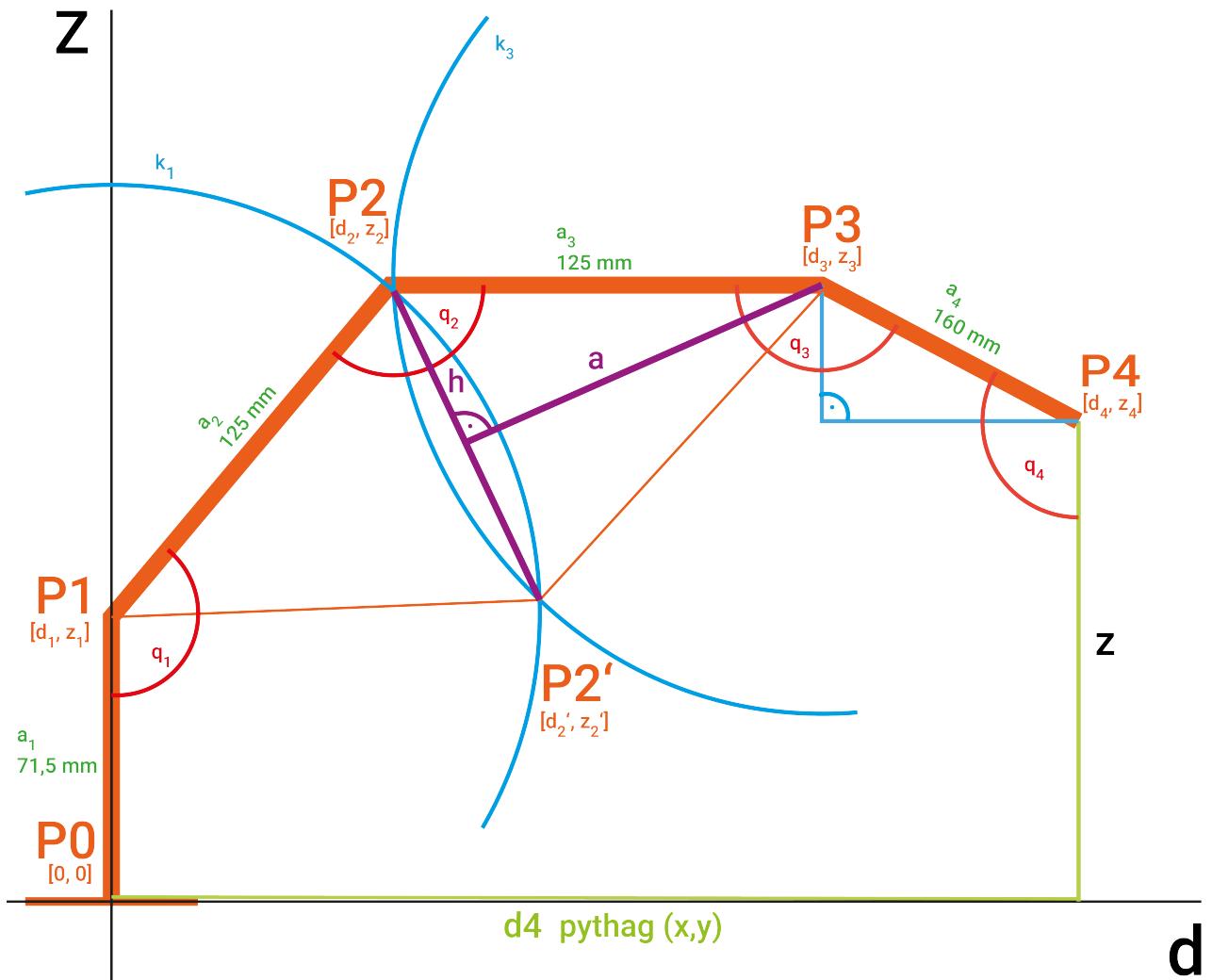
Schopnost robota orientovat se v prostoru je klíčovou vlastností. Pro operátora je ovšem mnohem výhodnější zadávat pozice v rámci pracovní oblasti ve formě Kartézských souřadnic, než ručně zadávat natočení každého kloubu robota zvlášť. Určení polohy pomocí souřadnic jednoznačně určuje požadované místo v prostoru a není třeba složitě počítat či odhadovat natočení jednotlivých kloubů manuálně.

Inverzní kinematická úloha byla vytvořena a naprogramována tak, že jejím vstupem jsou souřadnice bodu v prostoru pracovní oblasti ve formátu (X, Y, Z). Algoritmus přijme tyto vstupní parametry a s jejich pomocí vypočítá pro danou pozici jednotlivé kloubové úhly. Přitom kontroluje, aby výsledné vypočtené úhly nevedly ke kolizi ramene se sebou samým a aby výsledné úhly byly platné v pracovním rozsahu jednotlivých servomotorů.

Matematická část algoritmu je popsána schématy na obr. 4.5 a obr. 4.4.

4.2.1. Výpočet polohy bodů

Pro výpočet jednotlivých úhlů q_n je klíčové nejdříve vědět polohy všech kloubů P_n . Po výpočtu poloh všech kloubů pak lze úhly ramen mezi nimi dopočítat. Algoritmus pracuje ve 2D prostoru, jak je znázorněno na obr. 4.4, třetí dimenze není pro výpočet úhlů kloubů potřebná. Části algoritmu jsou konkrétně popsány v následujících kapitolách.



Obrázek 4.4.: Matematický popis ramene (boční pohled)

Body P_0 a P_1

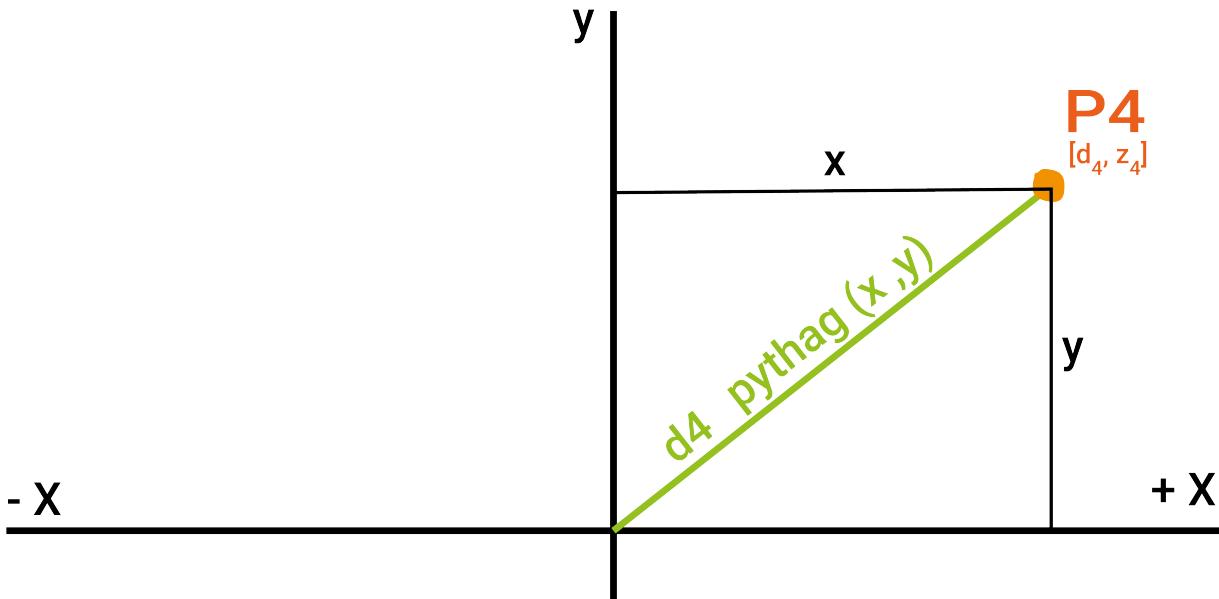
Pozice těchto dvou bodů jsou jako jediné známé ihned na začátku výpočtu. Dle obr. 4.4 se bod P_0 nachází v počátku souřadnic, leží tedy na pozici $[0, 0]$.

Bod P_1 je také známý, jeho pozici získáme pouze posunem z souřadnice bodu P_0 po ose z o délku základny a_1 (tj. 71,5 mm). Bod P_1 leží na pozici $[0, z_1]$, kde $z_1 = 71,5$.

Výpočet bodu P_4

Z obr. 4.4 je zřejmé, že základním předpokladem pro výpočet je vzdálenost d_4 od počátku souřadnic $[0, 0]$ do bodu P_4 . Tuto vzdálenost konkrétněji ukazuje schéma na obr. 4.5, kde je znázorněn pohled z hora na pracovní desku.

Pro výpočet je využita *Pythagorova věta*, jedná se o přeponu trojúhelníku. Vzhledem k tomu, že vzdálenosti x a y jsou známé (zadává je přímo operátor), lze je dosadit do vztahu

Obrázek 4.5.: Výpočet přepony d_4 z x a y

$$d_4 = \sqrt{x^2 + y^2} \quad (4.5)$$

Je-li vzdálenost d_4 známá, lze určit polohu bodu P_4 . Bod P_4 leží na souřadnicích $[d_4, z_4]$ kde z_4 je vzdálenost od pracovní desky k určenému bodu (operátorem zadaná souřadnice z).

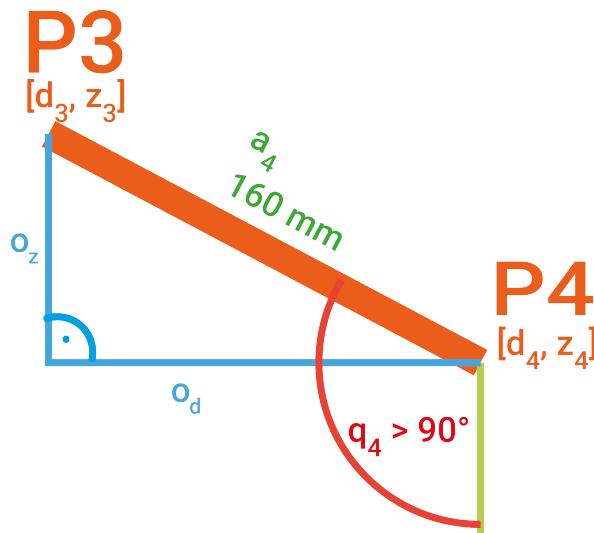
Výpočet bodu P_3

Je-li bod P_4 známý, bod P_3 lze dopočítat posunem bodu P_4 po osách d a z . Výpočty jednotlivých posunů se ovšem liší a závisí na velikosti úhlu natočení efektoru q_4 . Úhel q_4 je známý, pohybuje se v rozsahu od 180° do 0° a dodává ho sám algoritmus. V každém běhu cyklu dodá úhel q_4 a pokud pro daný úhel natočení efektoru není nalezena platná poloha, sníží úhel q_4 o 1° a výpočet opakuje. Začne tedy na 180° , pak případně sníží úhel efektoru na 179° , pak případně na 178° , 177° atd. Dojde-li algoritmus k úhlu 0° a stále nebyla nalezena platná pozice, algoritmus skončí a vyhodnotí nedosažitelnou zadанou pozici. V závislosti na velikosti úhlu q_4 mohou nastat 3 případy:

- $q_4 > 90^\circ$
- $q_4 = 90^\circ$
- $q_4 < 90^\circ$

Úhel $q_4 > 90^\circ$

V případě, že úhel q_4 je větší než 90° (obr. 4.6), využije algoritmus pro výpočet d_3 a z_3 pravidel *sinus* a *cosinus*, jelikož se jedná o výpočet odvesen pravoúhlého trojúhelníku. Přepona a_4 je známá,

Obrázek 4.6.: Úhel efektoru $q_4 > 90^\circ$

jedná se o rameno robotu o délce 160 mm. Je ovšem nutné od úhlu q_4 odečíst 90° , protože výpočet pracuje pouze s částí úhlu q_4 uvnitř trojúhelníku.

Pro výpočet odvěsny o_d je využit vztah

$$o_d = a_4 * \cos(q_4 - 90^\circ) \quad (4.6)$$

a pro výpočet odvěsny o_z pak vztah

$$o_z = a_4 * \sin(q_4 - 90^\circ) \quad (4.7)$$

Po výpočtu délky obou odvěsen lze posunem bodu P_4 dosáhnout polohy bodu P_3 . Souřadnice bodu P_3 tedy budou vypadat

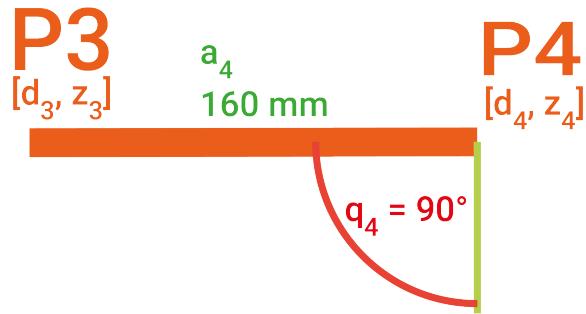
$$P_3 = [d_3, z_3] \quad d_3 = d_4 - o_d \quad z_3 = z_4 + o_z \quad (4.8)$$

Úhel $q_4 = 90^\circ$

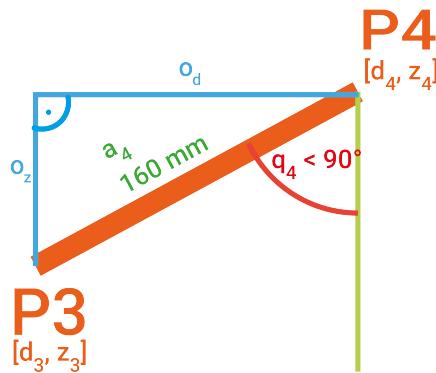
Speciální případ, kdy se úhel q_4 rovná 90° (obr. 4.7) nastává pouze jednou za chod algoritmu. Jedná se o nejsnazší výpočet, protože není potřeba počítat žádné odvěsny, posun bodu je zřejmý a souřadnice se posunou pouze o délku ramene a_4 .

Souřadnice bodu P_3 budou tedy vypadat

$$P_3 = [d_3, z_3] \quad d_3 = d_4 - a_4 \quad z_3 = z_4 \quad (4.9)$$

Obrázek 4.7.: Úhel efektoru $q_4 = 90^\circ$

Úhel $q_4 < 90^\circ$

Obrázek 4.8.: Úhel efektoru $q_4 < 90^\circ$

Posledním případem je úhel efektoru menší než 90° (obr. 4.8). Algoritmus opět využije pravidel *sinus* a *cosinus* a vypočítá odvěsny pro posun.

Pro výpočet odvěsny o_d je využit vztah

$$o_d = a_4 * \sin(q_4) \quad (4.10)$$

a pro výpočet odvěsny o_z pak vztah

$$o_z = a_4 * \cos(q_4) \quad (4.11)$$

Po výpočtu délky obou odvěsen lze posunem bodu P_4 dosáhnout polohy bodu P_3 . Souřadnice bodu P_3 tedy budou vypadat

$$P_3 = [d_3, z_3] \quad d_3 = d_4 - o_d \quad z_3 = z_4 - o_z \quad (4.12)$$

Výpočet bodu P₂

Jak již bylo řečeno v kapitole 1.4.1, inverzní kinematická úloha může mít nekonečně mnoho řešení. Algoritmus navržený v této bakalářské práci je ovšem sestaven takovým způsobem, že pro každý úhel natočení efektoru q₄ vypočítává a následně testuje pouze dvě polohy ramene (dva možné způsoby, jak daného bodu při daném úhlu q₄ dosáhnout). Toto nejen že snižuje náročnost celého řešení kinematické úlohy, ale hlavně snižuje výpočetní náročnost a tudíž snižuje zátěž na řídicí mikroprocesor.

Základem pro výpočet jsou známé pozice bodů P₁ a P₃, ty v tuto chvíli algoritmus již spočítal. Následuje tedy výpočet bodu P₂. Ve skutečnosti ovšem proběhne výpočet dvou bodů, hlavního bodu P₂ a alternativního bodu P'₂. Situaci nejlépe popisuje diagram na obr. 4.9. Algoritmus hledá průniky kružnic k₁ a k₃ o poloměru a₂ = a₃ = 125 mm.

Při výpočtu mohou nastat 3 situace

- 1 průnik - existuje pouze jedna poloha ramene a to taková, že úhel q₂ = 90°
- 2 průniky - nejčastější případ, algoritmus prověří obě možnosti a vybere první platnou polohu
- žádný průnik - pro zadaný bod neexistuje platná poloha robotu, úhel q₂ = NULL a algoritmus se posune k snížení úhlu q₄ o 1° a výpočet inverzní kinematické úlohy začne od začátku

Pro výpočet je klíčový vektor $\vec{v}_{[P1,P3]}$, přesněji jeho velikost. Vektor $\vec{v}_{[P1,P3]}$ lze určit

$$\vec{v}_{[P1,P3]} = (d_3 - d_1, z_3 - z_1) \quad (4.13)$$

velikost vektoru pak lze určit

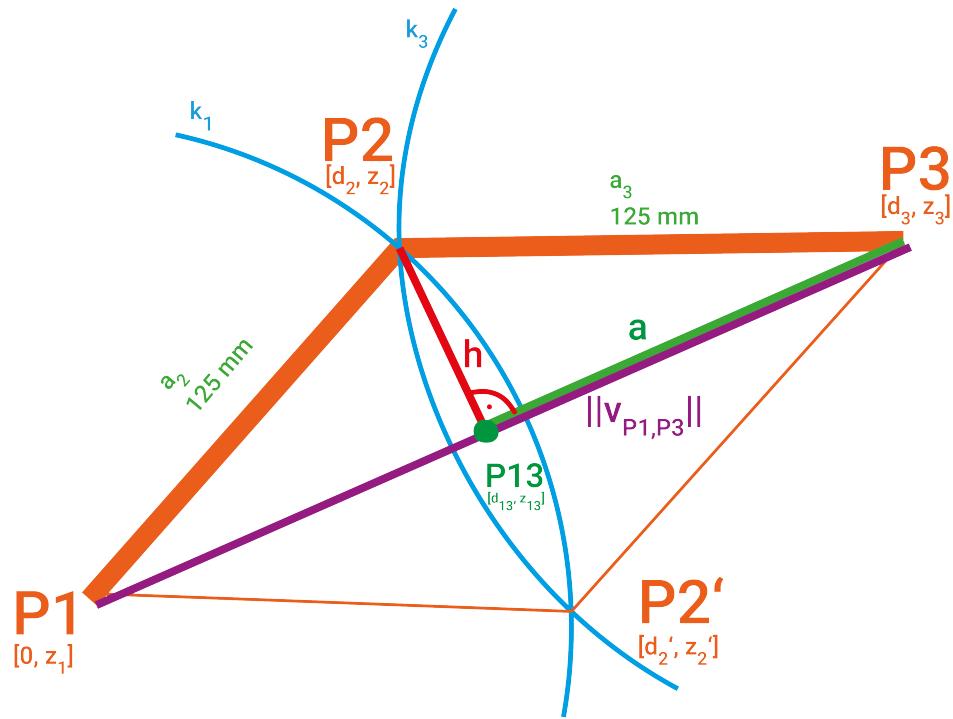
$$\|\vec{v}_{[P1,P3]}\| = \sqrt{(d_3 - d_1)^2 + (z_3 - z_1)^2} \quad (4.14)$$

Vzhledem k tomu, že a₂ = a₃ = 125 mm, jedná se o rovnoramenný trojúhelník, kde velikosti základny i ramen jsou nyní známé. Ze základny teď jasně lze určit velikost a

$$a = \frac{\|\vec{v}_{[P1,P3]}\|}{2} \quad (4.15)$$

Ve vzniklém pravoúhlém trojúhelníku lze pomocí Pythagorovy věty určit velikost výšky h

$$h = \sqrt{a_3^2 - a^2} \quad (4.16)$$



Obrázek 4.9.: Výpočet bodů P_2 a P_2'

Následně lze určit polohu pomocného bodu P_{13} obecně

$$P_{13} = P_1 + a * \frac{\vec{v}_{[P_1, P_3]}}{\| \vec{v}_{[P_1, P_3]} \|} \quad (4.17)$$

a jeho složky konkrétně

$$P_{13} = [d_{13}, z_{13}] \quad d_{13} = d_1 + a * \frac{d_3 - d_1}{\| \vec{v}_{[P_1, P_3]} \|} \quad z_{13} = z_1 + a * \frac{z_3 - z_1}{\| \vec{v}_{[P_1, P_3]} \|} \quad (4.18)$$

Ze získaného pomocného bodu P_{13} již lze dopočítat průniky kružnic k_1 a k_3 , body P_2 a P'_2

Bod P_2 lze dopočítat

$$P_2 = [d_2, z_2] \quad d_2 = d_{13} + h * \frac{z_3 - z_1}{\| \vec{v}_{[P_1, P_3]} \|} \quad z_2 = z_{13} - h * \frac{d_3 - d_1}{\| \vec{v}_{[P_1, P_3]} \|} \quad (4.19)$$

a obdobně lze dopočítat i P'_2

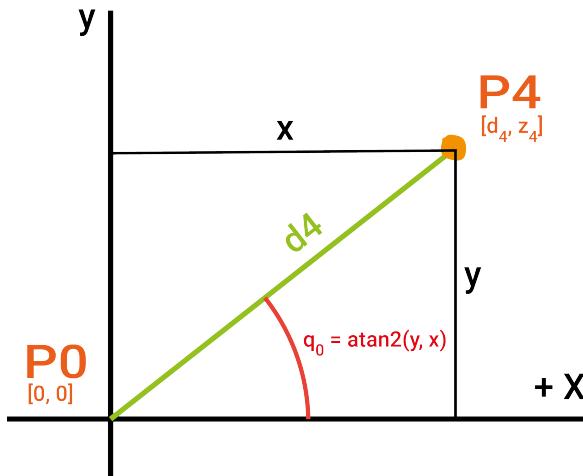
$$P'_2 = [d'_2, z'_2] \quad d'_2 = d_{13} - h * \frac{z_3 - z_1}{\| \vec{v}_{[P_1, P_3]} \|} \quad z'_2 = z_{13} + h * \frac{d_3 - d_1}{\| \vec{v}_{[P_1, P_3]} \|} \quad (4.20)$$

4.2.2. Výpočet úhlu q_0

V tuto chvíli algoritmus úspěšně nalezl alespoň jeden průnik kružnic k_1 a k_3 a dokázal tak dopočítat pozice všech bodů. Robotické rameno ovšem pro dosažení pozice potřebuje znát úhly natočení

jednotlivých ramen, tudíž úhly svírané mezi rameny $a_1 - a_4$ v bodech (kloubech) $P_1 - P_4$. Bod P_0 tvoří výjimku, úhel natočení celého ramene q_0 je možné vypočítat přímo ze zadaných souřadnic, přesněji složek x a y .

Úhel q_0 udává natočení celého ramene kolem osy z . Pracovní rozsah servomotoru je plný, tj $0^\circ - 180^\circ$. Na obr. 4.10 je úhel vyznačen, jedná se o pohled na desku shora.



Obrázek 4.10.: Výpočet úhlu q_0

Výpočet úhlu q_0 zajišťuje funkce *arcus tangens 2*. Jedná se o speciální druh funkce *arcus tangens*, která usnadňuje převod z Kartézských souřadnic na polární a při výpočtech zohledňuje kvadrant, ve kterém se převáděný bod nachází. Úhel q_0 lze vypočítat rovnicí 4.21 a není třeba ho dál převádět do Braccio úhlového systému.

$$q_0 = \text{arctg}2(y, x) \quad (4.21)$$

4.2.3. Výpočet úhlů $q_1 - q_3$

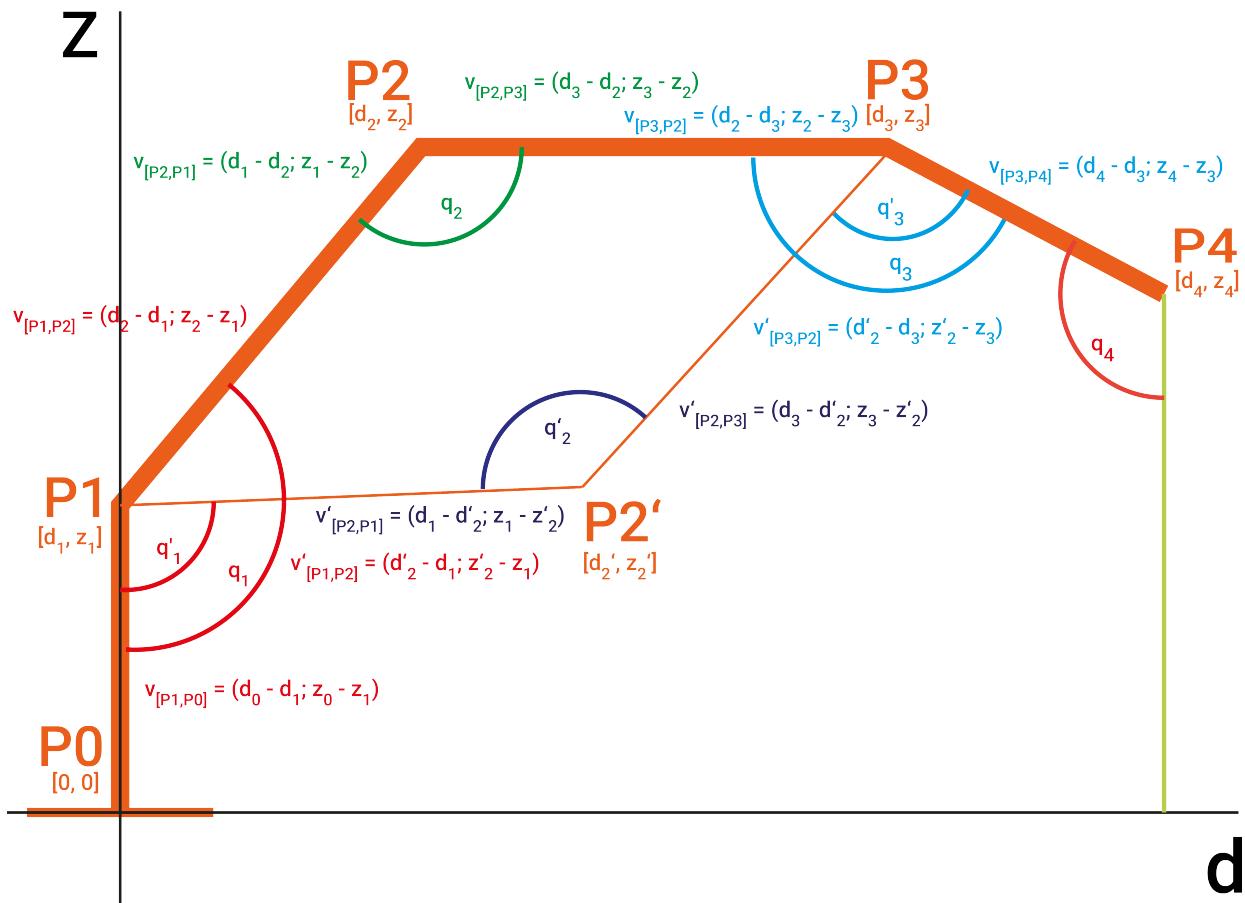
Jsou-li známé všechny pozice bodů $P_1 - P_4$, pak je možné vypočítat všechny potřebné vektory mezi těmito body. Obecně lze vektor mezi dvěma body určit vztahem

$$v_{ij} = P_j - P_i \quad (4.22)$$

kde v_{ij} je požadovaný vektor, P_j je vzdálenější bod a P_i je bližší bod. Následně lze využít vzorec pro odchylku dvou vektorů a s jeho pomocí spočítat úhly mezi těmito vektory. Úhly q_0 a q_4 jsou již známé, princip výpočtu zbylých úhlů $q_1 - q_3$ popisuje obr. 4.11.

Algoritmus využívá matematického vzorce

$$\cos \varphi = \frac{u * v}{|u| * |v|} \quad (4.23)$$

Obrázek 4.11.: Výpočet úhlů q_1-q_3

kde u a v jsou dva nenulové vektory. Pro využití výpočtu je však nutné místo $\cos \varphi$ počítat s funkcí $\arccos \varphi$, aby výstupem výpočtu byly stupně daného úhlu.

Výpočet q_3 a q'_3

Vzhledem k tomu, že kinematická úloha uvažuje dvě možné polohy ramena pro jeden zadaný bod koncového efektoru i výpočty úhlů pro jednotlivé body budou dva. Úhel q_3 reprezentuje hlavní polohu ramene a úhel q'_3 reprezentuje alternativní polohu ramene pro případ, že by zadaný bod v pracovním prostoru robotu nebyl hlavní polohou dosažitelný.

Výpočet hlavního úhlu q_3 využívá matematický vzorec 4.23 s použitím \arccos místo \cos . Úhel lze vypočítat se znalostí vektorů $\vec{v}_{[P3,P4]}(d; z)$ a $\vec{v}_{[P3,P2]}(d; z)$ rovnici

$$q_3 = \arccos \left(\frac{\vec{v}_{[P3,P4]}[d] * \vec{v}_{[P3,P2]}[d] + \vec{v}_{[P3,P4]}[z] * \vec{v}_{[P3,P2]}[z]}{\sqrt{\vec{v}_{[P3,P4]}[d]^2 + \vec{v}_{[P3,P4]}[z]^2} * \sqrt{\vec{v}_{[P3,P2]}[d]^2 + \vec{v}_{[P3,P2]}[z]^2}} \right) \quad (4.24)$$

Alternativní úhel q'_3 lze pak obdobně vypočítat

$$q'_3 = \arccos \left(\frac{\vec{v}_{[P3,P4]}[d] * \vec{v}'_{[P3,P2]}[d] + \vec{v}_{[P3,P4]}[z] * \vec{v}'_{[P3,P2]}[z]}{\sqrt{\vec{v}_{[P3,P4]}[d]^2 + \vec{v}_{[P3,P4]}[z]^2} * \sqrt{\vec{v}'_{[P3,P2]}[d]^2 + \vec{v}'_{[P3,P2]}[z]^2}} \right) \quad (4.25)$$

Výpočet \mathbf{q}_2 a \mathbf{q}'_2

Stejně jako bod P_3 má 2 úhly, i bodu P_2 náleží úhel hlavní (q_2) a alternativní (q'_2). Výpočet opět využívá rovnici 4.23. Úhel q_2 lze vypočítat

$$q_2 = \arccos \left(\frac{\vec{v}_{[P2,P3]}[d] * \vec{v}_{[P2,P1]}[d] + \vec{v}_{[P2,P3]}[z] * \vec{v}_{[P2,P1]}[z]}{\sqrt{\vec{v}_{[P2,P3]}[d]^2 + \vec{v}_{[P2,P3]}[z]^2} * \sqrt{\vec{v}_{[P2,P1]}[d]^2 + \vec{v}_{[P2,P1]}[z]^2}} \right) \quad (4.26)$$

Obdobně alternativní úhel q'_2 lze vypočítat

$$q'_2 = \arccos \left(\frac{\vec{v}'_{[P2,P3]}[d] * \vec{v}'_{[P2,P1]}[d] + \vec{v}'_{[P2,P3]}[z] * \vec{v}'_{[P2,P1]}[z]}{\sqrt{\vec{v}'_{[P2,P3]}[d]^2 + \vec{v}'_{[P2,P3]}[z]^2} * \sqrt{\vec{v}'_{[P2,P1]}[d]^2 + \vec{v}'_{[P2,P1]}[z]^2}} \right) \quad (4.27)$$

Výpočet \mathbf{q}_1 a \mathbf{q}'_1

Bodu P_1 také náleží 2 úhly. Hlavní úhel q_1 lze vypočítat

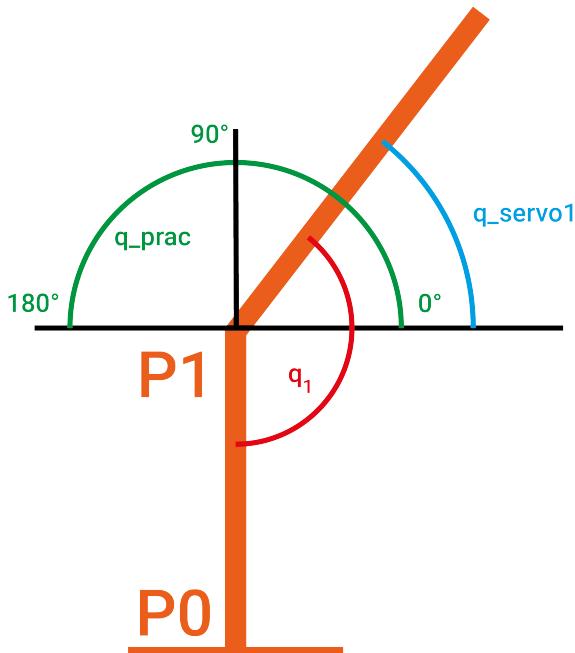
$$q_1 = \arccos \left(\frac{\vec{v}_{[P1,P2]}[d] * \vec{v}_{[P1,P0]}[d] + \vec{v}_{[P1,P2]}[z] * \vec{v}_{[P1,P0]}[z]}{\sqrt{\vec{v}_{[P1,P2]}[d]^2 + \vec{v}_{[P1,P2]}[z]^2} * \sqrt{\vec{v}_{[P1,P0]}[d]^2 + \vec{v}_{[P1,P0]}[z]^2}} \right) \quad (4.28)$$

Obdobně úhel q'_1 lze vypočítat

$$q'_1 = \arccos \left(\frac{\vec{v}'_{[P1,P2]}[d] * \vec{v}'_{[P1,P0]}[d] + \vec{v}'_{[P1,P2]}[z] * \vec{v}'_{[P1,P0]}[z]}{\sqrt{\vec{v}'_{[P1,P2]}[d]^2 + \vec{v}'_{[P1,P2]}[z]^2} * \sqrt{\vec{v}'_{[P1,P0]}[d]^2 + \vec{v}'_{[P1,P0]}[z]^2}} \right) \quad (4.29)$$

4.2.4. Převod úhlů do Braccio systému

Algoritmus s využitím vektorové matematiky úspěšně získal všechny požadované úhly natočení kloubů $q_0 - q_3$. Takto vypočítané úhly ovšem nejsou finální, nelze je zapsat do servomotorů. Problém popisuje obr. 4.12.



Obrázek 4.12.: Úhlové systémy

Jak je z obr. 4.12 patrné, úhlové systémy se liší. Algoritmus sice dokáže spočítat úhel svíraný dvěma rameny (v tomto případě q_1), ale servomotor ovládající kloub P_1 pracuje v rozdílném rozsahu a z rozdílné počáteční polohy. Servomotor má pracovní rozsah definovaný pouze od 0° do 180° (q_{prac}) a jeho počátek (tj. 0°) začíná v 90° skutečné hodnoty úhlu q_1 . Požaduje tedy úhel q_{servo1} .

Pro získání správného úhlu pro servomotor (převod úhlu do *Braccio systému* zmíněného v kapitole 4.2.2) je tak potřeba od všech úhlů $q_1 - q_3$ odečíst 90° , tedy

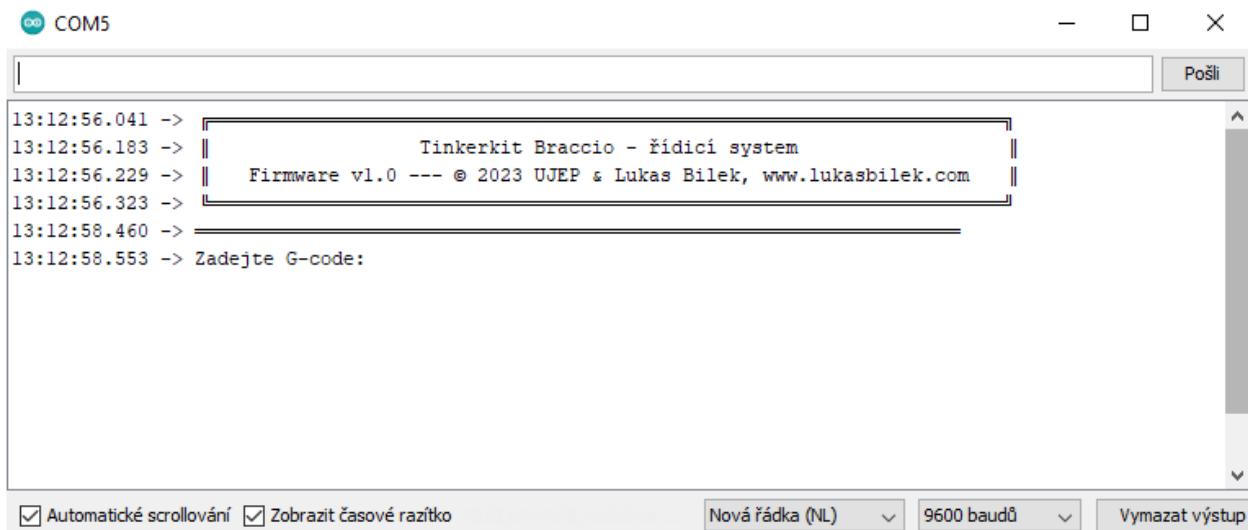
$$q_{servo_n} = q_n - 90^\circ \quad (4.30)$$

Převod úhlů je poslední částí algoritmu inverzní kinematické úlohy. Vypočítané a převedené úhly jsou vráceny zpět volající funkci programu a ta je následně odešle do jednotlivých servomotorů.

4.3. Funkce firmwaru

Řídicí systém robotického ramena implementuje industriální standard *G-code*. Jedná se o speciální „jazyk“, přesněji spíš sadu instrukcí, začínajících písmenem **G** nebo **M** a následované číslem označujícím konkrétní instrukci. Součástí G-code instrukce mohou být i další dodatečné parametry jako úhly, souřadnice a další. Používá se především pro CNC stroje a jiné průmyslové stroje.

Robotické rameno je tedy možné pomocí těchto instrukcí řídit. Po připojení USB kabelu k počítači, spuštění terminálu a navázání sériové komunikace lze ramenu odesílat G-kódy a ovládat ho v reálném čase(viz obr. 4.13). Konkrétní instrukce, kterými robotické rameno disponuje, jsou podrobně rozepsány v následujících kapitolách.



Obrázek 4.13.: Terminál řídicího systému

Ovládací instrukce *G* [číslo] využívají další interní pomocné funkce (algoritmy), které jsou v rámci G instrukcí volány. Tyto ovšem nejsou v bakalářské práci podrobně popisovány z důvodu rozsahu celkové práce a nižší důležitosti. Kompletní zdrojový kód včetně interních pomocných funkcí doplněných o komentáře je dostupný na úložišti GitHub, viz kapitola A *Externí přílohy*. Důležité principy a algoritmy jsou však v práci podrobně popsány.

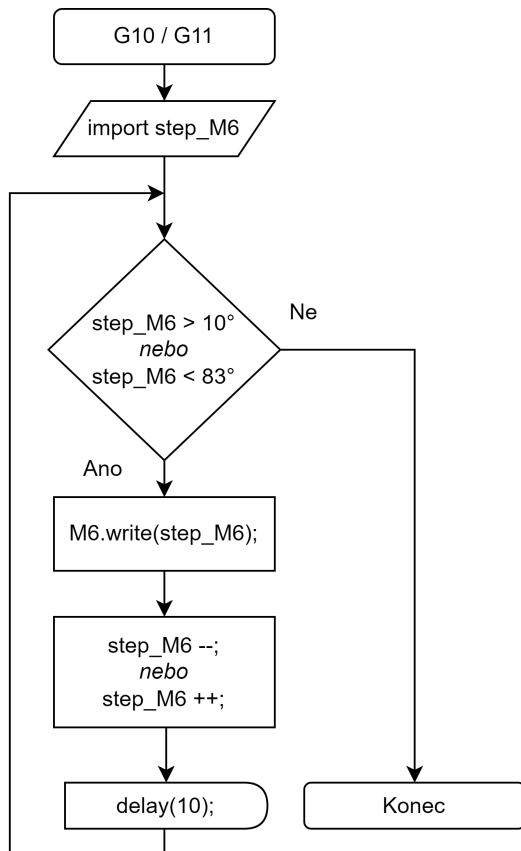
4.3.1. G10 a G11 - Ovládání gripperu

Funkce *G10* a *G11* ovládají koncový effektor robotického ramene, v tomto případě gripper na uchopení předmětů. Instrukce *G10* gripper ramene otevře a instrukce *G11* ho zavře.

Průběh algoritmu

Funkce instrukce je popsána diagramem na obr. 4.14. Po zavolání instrukce *G10* či *G11* algoritmus přistoupí k proměnné reprezentující aktuální hodnotu úhlu natočení servomotoru M_6 . Pro zavření gripperu je dle oficiálního manuálu Tinkerkit Braccio určena hodnota natočení M_6 rovna 83° , pro otevření gripperu je hodnota M_6 10° . Také pohyb samotného gripperu by měl být co nejvíce plynulý. Pokud by se provedl pouze zápis hodnoty úhlu do servomotoru M_6 , gripper by se skokově otevřel či zavřel a vytvářel nežádoucí pohybové rázy a znemožňoval jemné uchopení předmětů. Algoritmus tedy místo zápisu na přímo spustí cyklus *while*. Podmínka cyklu testuje, zda aktuální hodnota natočení dosahuje požadované hodnoty. Pokud tomu tak není, algoritmus zapíše aktuální hodnotu úhlu do motoru M_6 instrukcí $M6.write(step_M6);$, kde $M6$ reprezentuje

připojený servomotor využívající oficiální Servo knihovnu Arduino a *step_M6* reprezentuje aktuální hodnotu úhlu M6. Po zápisu aktuální hodnoty je tato hodnota zvýšena či snížena o 1° podle toho, zda se gripper otevírá nebo zavírá. Dále je volán příkaz *delay(10)*; který určuje rychlosť otevírání nebo zavírání gripperu změnou délky čekání. Cyklus je tímto u konce a probíhá do té doby, dokud aktuální hodnota úhlu nedosáhne požadované hodnoty pro otevření či zavření gripperu.



Obrázek 4.14.: Vývojový diagram G10/G11

4.3.2. G92 - Nastavení aktuální pozice os

Funkce *G92* přímo nastavuje aktuální pozice os kloubů robotu. Jedná se tedy o přímé nastavení jednotlivých úhlů natočení servomotorů $q_0 - q_3$ a úhel natočení gripperu ve vertikální ose servomotoru M_5 . Neprobíhají zde žádné výpočty a nepracuje se se souřadnicovým systémem. Instrukci lze zavolat jak z konzole, tak i z dalších instrukcí. Instrukce implementuje algoritmus pro lineární pohyb popsany v kapitole 4.1.

Průběh algoritmu

Funkce instrukce je popsána diagramem na obr. 4.15. Po zavolání instrukce se importují aktuální hodnoty natočení úhlů $q_0 - q_3$. Algoritmus se skrz terminál doptá na dodatečné úhly $q_0 - q_3$ a

rychlosť pohybu *step_delay*, úhel natočení vertikální osy gripperu předpokládáme konstantních 90° (kolmo s pracovní plochou pro možnost uchopit předmět). Tyto úhly postupně uloží do lokálních proměnných *q0_* – *q3_*. Následně potvrší přijetí úhlů výpisem zprávy a přijatých úhlů do konzole. Algoritmus dále otestuje, zda jsou zadané úhly v pracovních rozsazích daných servomotorů. Není-li úhel uvnitř rozsahu daného servomotoru, je hodnota úhlu nahrazena hraniční hodnotou intervalu pracovních úhlů takovou, kterou úhel překročil. Kontrola probíhá z toho důvodu, aby robotické rameno nenařazilo některým z rámén samo do sebe, některé klouby se sice mohou otočit do plných 180° nebo 0° , hrozilo by ovšem jejich poškození.

Algoritmus nyní spustí již dříve popsaný algoritmus pro linearizaci pohybu z kapitoly 4.1. Určí tedy servomotor s nejdelší dráhou pohybu a podle nich vypočítá inkrementy/dekrementy ostatních servomotorů. Poté se spustí cyklus *while*, jehož podmínkou je pomocná řídicí proměnná typu bool. V cyklu se testuje každý motor zvlášť. U každého motoru se testuje podmínka, zda již dosáhl své požadované polohy a pokud ne, dle požadavku se přičte/odečte příslušný inkrement/dekrement daného úhlu. provede se zápis úhlu do servomotoru instrukcí *Mn.write(step_Mn)*; kde *M_n* je konkrétní servomotor a *step_M_n* je příslušný aktuální úhel natočení tohoto motoru. Po úhlových výpočtech a zápisech je kontrolován stav tlačítka CENTRAL–STOP a v případě aktivace tlačítka je vyvoláno přerušení a obslužný algoritmus (více v kapitole 5.2). Následně dojde k finální kontrole úhlů. Jsou-li všechny aktuální úhly rovny požadovaným, cyklus končí. Pokud se úhly nerovnají, následuje instrukce *delay(step_delay)*; kde *step_delay* představuje zpoždění v ms. Cyklus se následně opakuje do té doby, dokud úhly nedosáhnou svých požadovaných pozic.

4.3.3. G28 - Autohome

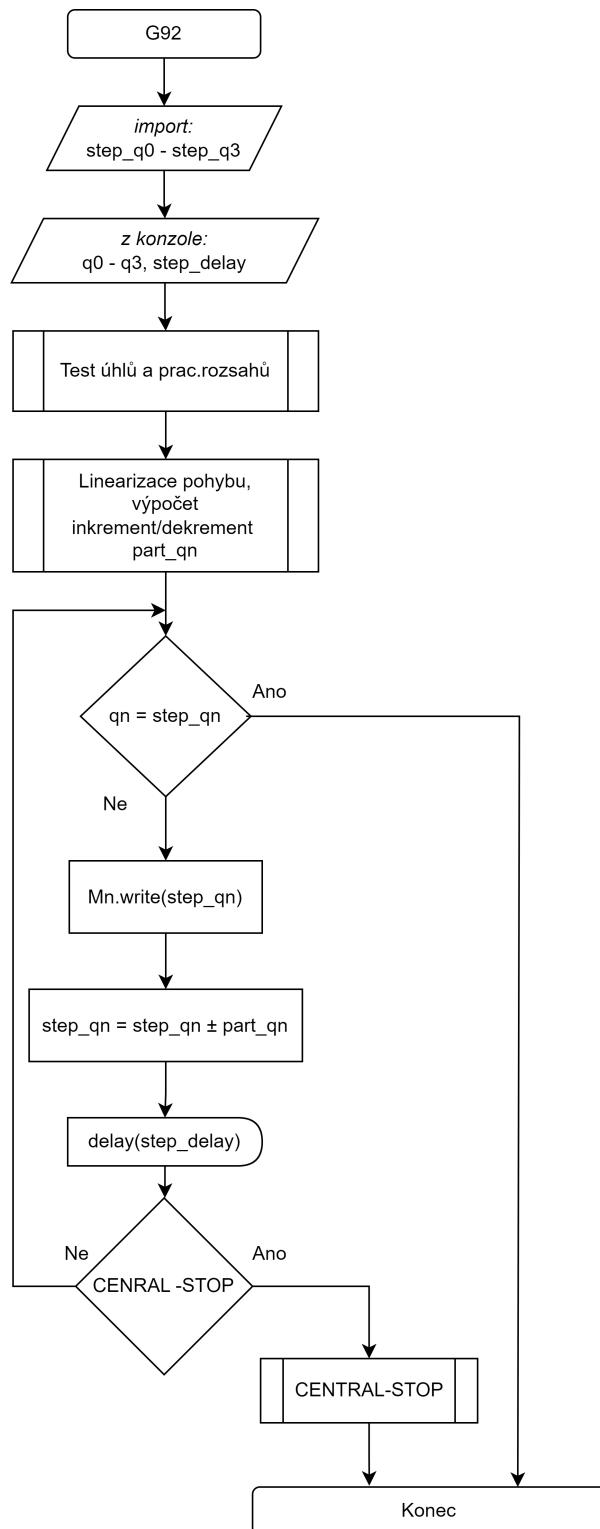
Funkce *G28* nastavuje polohu robotického ramene do základní (domovské) polohy. V této poloze rameno nijak nezasahuje do pracovní oblasti a je připraveno pro příjem instrukce. Každá úloha končí přesunem ramene do domovské polohy. *G28* se také vykoná při každém zapnutí ramene.

Průběh algoritmu

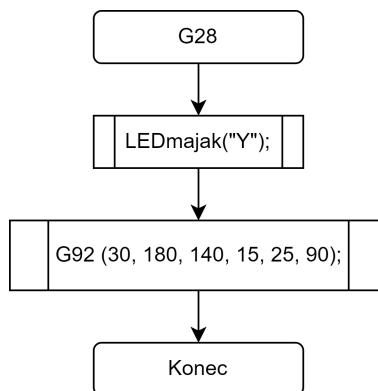
Funkce instrukce je popsána diagramem na obr. 4.16. Po zavolání instrukce se nejprve funkcií *LEDmajak("Y")*; světelný maják přestaví ze zeleného světla (stand-by) na světlo žluté (v provozu). Následně se zavolá funkce *G92* s konkrétními parametry. První parametr je rychlosť pohybu. Zbylé parametry reprezentují natočení jednotlivých servomotorů ve stupních. Funkce *G92* pak provede lineární pohyb dle popisu v kapitole 4.3.2. Po dosažení domovské polohy je funkce ukončena.

4.3.4. G90 - Lineární pohyb IKU

Funkce *G90* zajišťuje automatický přesun ramene na souřadnici v pracovním prostoru zadanou operátorem ramene, je-li tato pozice dosažitelná. Využívá k tomu řešení Inverzní kinematické úlohy



Obrázek 4.15.: Vývojový diagram G92



Obrázek 4.16.: Vývojový diagram G28

popsané v kapitole 4.2. Pohyb ramene v prostoru s využitím souřadnic je uživatelsky mnohem přívětivější, než odhadování a nastavování konkrétních úhlů manuálně pomocí funkce *G92*.

Průběh algoritmu

Funkce instrukce je popsána diagramem na obr. 4.17. Po zavolení instrukce se algoritmus doptá skrz konzoli na souřadnice bodu, do kterého se chce operátor s ramenem přesunout. Poté se funkcí *LEDmajak("Y")*; světelný maják přestaví ze zeleného světla (stand-by) na světlo žluté (v provozu). Následně proběhne test zadaných souřadnic. Kontroluje se, zda jsou zadané souřadnice uvnitř pracovního prostoru robotického ramene. Pracovní prostor je definován pevně v programu dle velikosti konkrétní pracovní desky. Dle potřeby je možné ho změnit. Je-li některá ze souřadnic mimo pracovní oblast, přepíše se požadovaná souřadnice na krajní hodnotu intervalu, který původní hodnota překročila. Je-li například požadovaná souřadnice $x = 450^\circ$ a pracovní oblast je definována $X \in <-300; 300>$, je x nastaveno na 300.

Po kontrole souřadnic je spuštěn cyklus *for* s podmínkou *for (int i = 180; i > 0; i--)*, kde i je pomocná proměnná pro cyklus a zároveň se jedná o úhel efektoru q_4 , který program testuje od 180° do 0° . S každým průchodem cyklu se i sníží o 1° . Pro každé i cyklu algoritmus spustí *Inverzní kinematickou úlohu IKU(x, y, z, i);*, kde x , y a z jsou souřadnice požadovaného bodu a i měnící se úhel efektoru q_4 .

Funkce *IKU* vypočítá úhly primární a alternativní polohy. Funkce *IKU* je podrobně popsána v kapitole 4.2. Výsledné úhly jsou následně testovány. Testuje se zejména, zda výsledná hodnota úhlů není *NaN - Not a Number*. Hodnota *NaN* reprezentuje nečíselnou hodnotu a pokud *IKU* vrátí jakýkoliv úhel primární nebo alternativní polohy *NaN*, znamená to, že pro daný úhel efektoru q_4 není nalezena platná poloha natočení robotického ramene. Následně se testuje, zda jsou jednotlivé úhly v platném rozsahu úhlů pro příslušný servomotor.

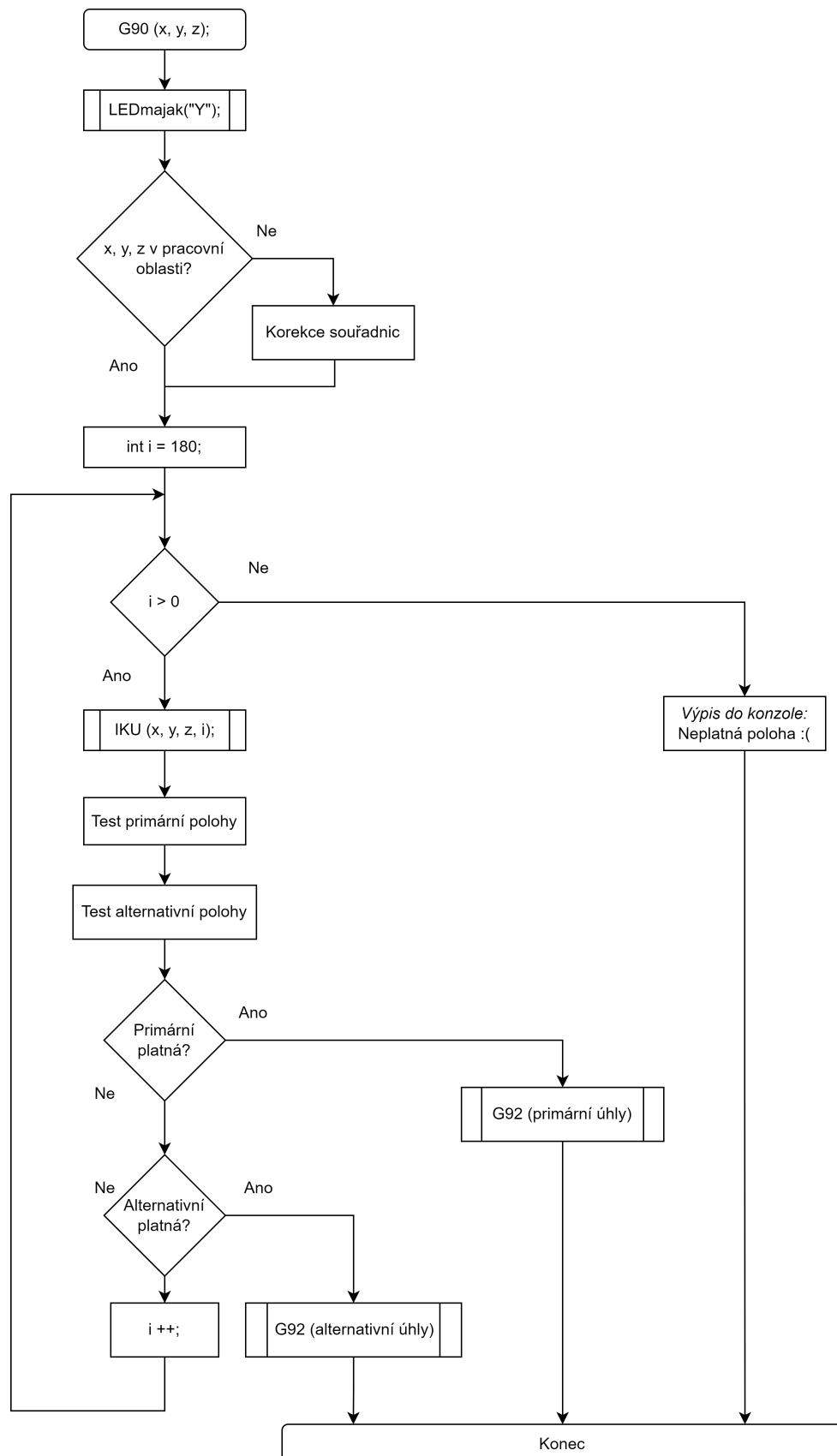
Projde-li primární nebo alternativní poloha testem na platnost úhlů, algoritmus našel platnou polohu pro zadaný bod a je zavolána funkce *G92* s vypočítanými parametry. Při tom se rozhoduje takto:

- Je-li platná primární i alternativní poloha, algoritmus volá *G92* vždy s úhly pro primární polohu
- Je-li platná pouze primární nebo pouze alternativní poloha, algoritmus volá *G92* s příslušnou platnou polohou

Pokud ani primární ani alternativní poloha nesplní test platnosti úhlů, je pomocná proměnná cyklu i (reprezentace úhlu efektoru q_4) snížena o 1 a cyklus se spustí znovu. Dojde-li opakováním během cyklu hodnota i postupně až do 0° , algoritmus nenašel pro požadovaný bod platnou polohu natočení ramene a algoritmus skončí s výpisem informativní hlášky do konzole "Neplatná poloha :(".

4.3.5. G101 – G105 - Ukázkové úlohy

Funkce *G101* a vyšší ve standardním G-kódu neexistují, byly proto využity pro spouštění ukázkových úloh, které využívají dříve popsané funkce a procedury. Jednotlivé ukázkové úlohy jsou podrobně popsány v jednotlivých manuálech pro tyto úlohy. Manuály jsou k dispozici v externí příloze bakalářské práce na platformě GitHub a slouží pro porozumění a správnou přípravu robotického pracoviště pro danou úlohu. Rameno využívá 3D vytisknuté válečky jako demonstraci přemísťovaného nákladu.



Obrázek 4.17.: Vývojový diagram G90

5. Přídavné periferie

V rámci praktické realizace bakalářské práce nebyl vytvořen pouze řídicí software, ale i hardware. Kombinací vlastního softwaru a hardwarových prvků byl vytvořen model robotického pracoviště. Pro operátora byl implementován světelný maják, který slouží jako prvek vizuální zpětné vazby. Operátor díky němu přesně ví, v jakém stavu se robotické rameno nachází. Samozřejmě nelze opomenout ani bezpečnost, ta je vždy stavěna na první místo. Z toho důvodu bylo implementováno nouzové tlačítko CENTRAL-STOP, které může operátor nebo kdokoliv jiný v případě nouze aktivovat a zastavit tak provoz ramene. Jednotlivé prvky jsou podrobněji popsány v následujících kapitolách.

V rámci bezpečnosti byla v úvahu i instalace světelné závory, která by hlídala pracovní oblast robotu a zastavila činnost robotu v případě vstupu nepovolané osoby do pracovního prostoru. Z důvodu vyšších pořizovacích nákladů však nakonec nebyla implementována.

5.1. Světelný maják

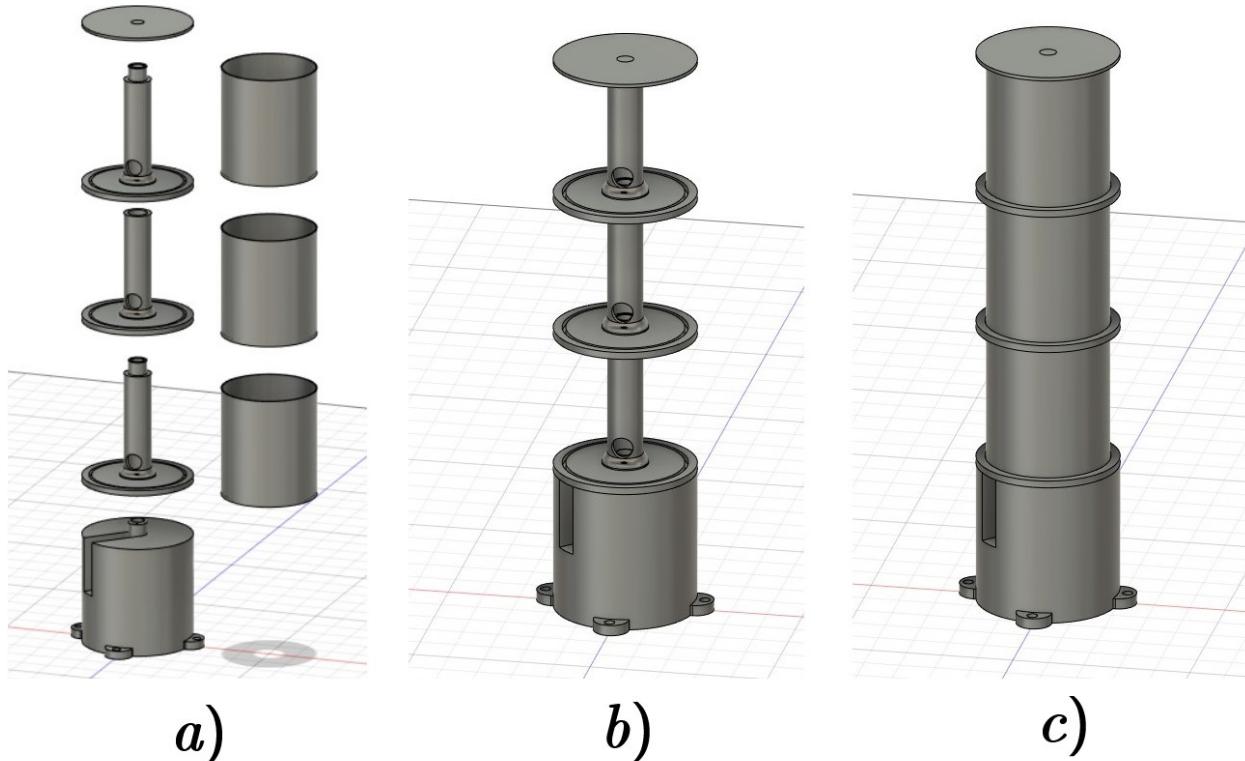
Při obsluze robotického ramene by měl operátor přesně vědět, co se s ramenem děje a v jakém stavu se nachází. Jako prvek zpětné vazby byl zvolen signalizační světelný maják, který svým barevným světlem nebo kombinací těchto světel poskytuje operátorovi zpětnou vazbu o tom, v jakém provozním stavu se rameno nachází.

Světelný maják byl navržen ve 3D CAD programu Autodesk Fusion 360 a vytisknut na 3D tiskárně z materiálu PLA. Vlastní výroba byla zvolena z důvodu nižších pořizovacích nákladů, možnosti vyrobit maják v přesně požadované velikosti a také zajištění elektrické kompatibility. Běžné světelné majáky používané v reálných výrobních provozech totiž většinou využívají napájecí napětí 24V stejnosměrných/střídavých. Externí napájecí zdroj pro robotické rameno však dodává pouze 5V a bylo by tedy nutné investovat i do nového napájecího zdroje, což by znamenalo opět vyšší pořizovací náklady.

5.1.1. Návrh konstrukce

Při návrhu konstrukce světelného majáku byl kladen důraz na co největší podobnost s běžně využívanými majáky ve výrobě, z toho důvodu byl zvolen maják se třemi světelnými prvky. Vzhledem k technologii 3D tisku FFF (*Fused Filament Fabrication*) nebylo možné celý maják vytisknout v jednom kuse a konstrukce byla navržena jako skládaná z pěti dílů, viz obr. 5.1-a).

Základním dílem je spodní základna se čtyřmi otvory pro šrouby. Maják se tak dá přimontovat přímo na pracovní desku. Skrz základnu prochází otvorem přívodní kabel, který napájí LED diody v jednotlivých "patrech" majáku. Na základně jsou nacvaknuté tři díly jednotlivých světel. Tento díl sestává ze spodního disku s drážkou a dutého sloupku uprostřed s otvorem, skrz který přichází přívodní dráty k LED diodám. Dutý sloupek je nahoře ukončen kruhovou záslepkou. Světelná "patra" majáku jsou pak kryta velmi tenkými bílými válečky - difuzory (tloušťka stěny cca 0,3mm), které skryjí instalované LED diody a rozptýlí světlo. 3D vizualizaci majáku bez difuzorů (**b**) a s difuzory (**c**) můžete vidět na obr. 5.1–**b,c**.



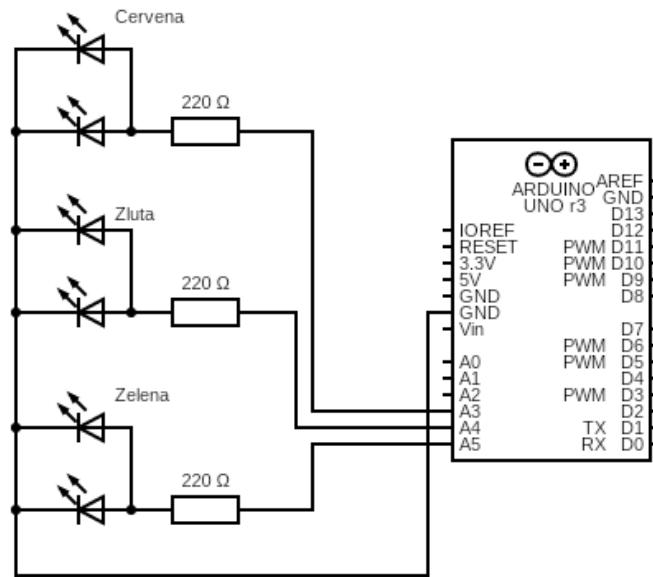
Obrázek 5.1.: 3D model světelného majáku

5.1.2. Elektronické zapojení

Světelný maják sestává ze tří světelných prvků - světel červené, žluté a zelené barvy. Pro osvětlení byly vybrány LED diody příslušných barev, dva kusy pro každou barvu. Každý pár LED diod je zapojen v paralelní kombinaci v sérii s ochranným rezistorem 220Ω . Ochranný rezistor je nutný z důvodu redukce napájecího napětí 5V.

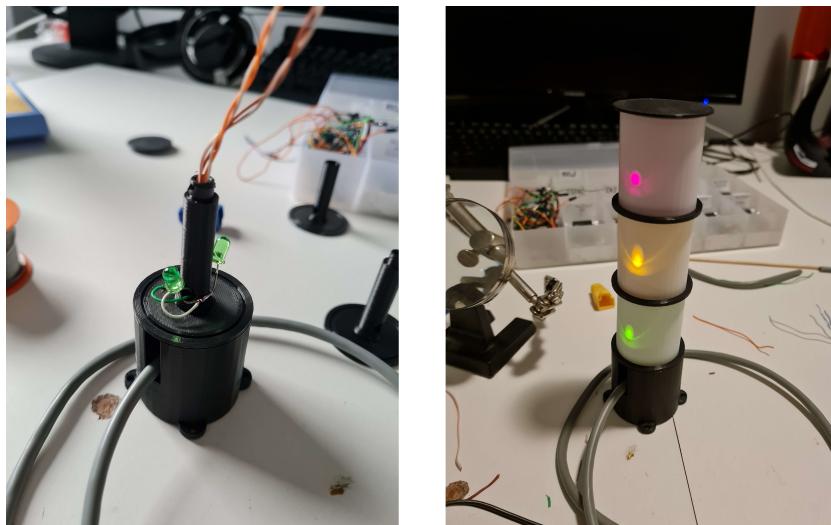
Každý světelný prvek je následně připojen na jeden z analogových výstupů řídicí jednotky Arduino. Maják je zapojen dle schématu na obr. 5.2. Pro připojení jsou využity analogové vstupy/výstupy A3 – A5 softwarově přepnuté na digitální výstupy.

Reálné zapojení jednoho ze světelných prvků je ukázáno na obr. 5.3 vlevo, zde je vidět zapojení ochranného rezistoru a dvou zelených LED diod. Vidět jsou také jednotlivé vodiče, které prochází



Obrázek 5.2.: Schéma zapojení světelného majáku

středem majáku do dalších "pater". Sestavený a rozsvícený světelný maják se pak nachází na stejném obrázku vpravo. Konkrétně zde svítí kombinace všech barev, která aktuálně signalizuje spouštěcí sekvenci celého robotického pracoviště, to je zapínání ramene, inicializaci komponent a navazování sériového spojení.



Obrázek 5.3.: Reálná podoba světelného majáku

5.1.3. Řízení funkce majáku

Jak již bylo uvedeno v minulé kapitole, světelný maják je připojen k analogovým portům vývojové desky Arduino Uno. Díky 3 světelným prvkům lze na majáku rozsvítit až 7 světelných kombinací. K dostatečné zpětné vazbě však stačí 4 kombinace rozepsané v tabulce 5.1.

Světelná kombinace	Význam
Zelená, Žlutá, Červená	Spouštění robotického pracoviště
Zelená	"Stand by"- rameno je připraveno a čeká na příchozí instrukci
Žlutá	Rameno vykonává zadanou instrukci/úlohu
Červená	CENTRAL-STOP - Nouzový stav, rameno se zastaví v aktuální poloze

Tabulka 5.1.: Světelné kombinace a jejich význam

5.2. Tlačítko CENTRAL-STOP

Bezpečnost je vždy na prvním místě. Funkčnost zařízení je sice klíčová, ale pokud je takové zařízení nebezpečné obsluze či jeho prostředí, není z praktického hlediska využitelné. Tak jako v reálných provozech jsou všechna zařízení opatřena bezpečnostními prvky, tak i model robotického pracoviště implementuje bezpečnostní prvek - nouzové tlačítko CENTRAL-STOP. Nouzové tlačítko plní funkci zastavení probíhající instrukce v případě nouze. Dojde-li k narušení pracovní oblasti, nechťkolizí ramene s objektem v pracovním prostoru, kolizi ramene s podložkou či jinému nechťkoliznému chování, má obsluha možnost toto tlačítko aktivovat a okamžitě tak zastavit aktuálně prováděný úkon.

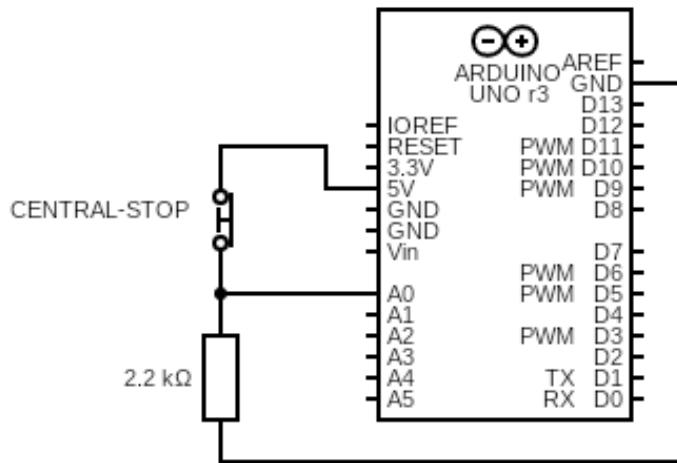
Bylo použito standardní nouzové tlačítko určené pro reálný provoz výrobce Rhenes. Tlačítko je typu **NC** (*Normally Closed*, ve výchozím stavu jsou tedy jeho kontakty sepnuté a proud může procházet skrz tlačítko). V případě zmáčknutí tlačítka kulatou houbovou hlavou dojde k rozpojení kontaktů a aretační mechanismus tlačítka ponechá ve zmáčknuté poloze, ovládaný obvod se rozpojí. Pro opětovné sepnutí je třeba pootočit hlavou tlačítka ve směru šipek, aretační mechanismus se uvolní a kontakty se opět spojí.

5.2.1. Elektronické zapojení

Standardně nouzová tlačítka přerušují přívod elektrické proudu k danému zařízení. V rámci modelu robotického pracoviště bylo nouzové tlačítko implementováno tak, že při jeho aktivaci dochází ke zrušení probíhající úlohy a zastavení ramene v aktuální poloze. Nouzové tlačítko je tedy připojeno na analogový vstup **A0**. Zapojení je popsáno schématem na obrázku 5.4.

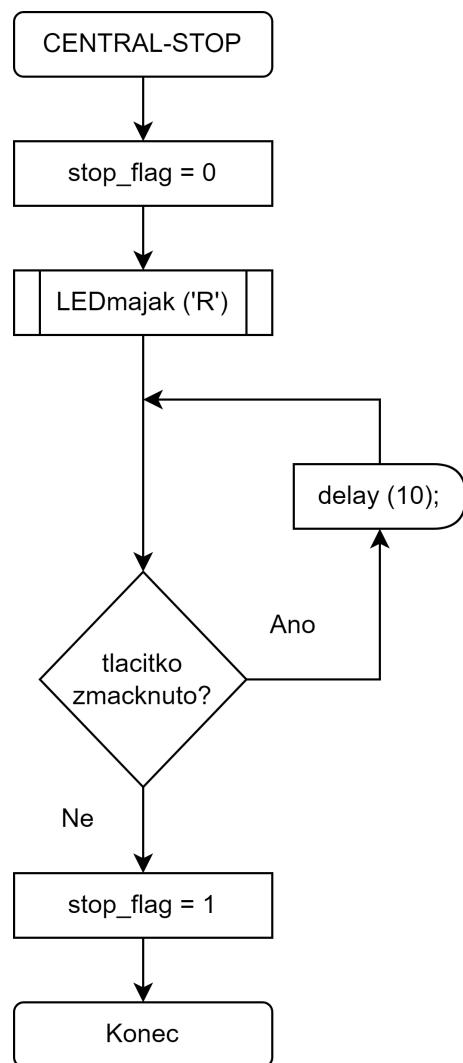
5.2.2. Řízení funkce tlačítka

Funkce **CENTRAL-STOP** je volána v pohybové funkci **G92** v případě, zjistí-li test tlačítka, že je tlačítko stisknuto a kontakty rozepnuty, viz diagram na obr. 4.15 na straně 52. Funkce G92 je součástí všech pohybových funkcí i ukázkových úloh. V okamžiku, kdy funkce zjistí, že je nouzové tlačítko stisknuto (a tedy kontakty jsou rozpojeny), dojde k okamžitému přerušení pohybu, rozsvícení červeného světla na světelném majáku a funkce spustí čekací smyčku, ve které setrvá, dokud je nouzové tlačítko stisknuto. Po uvolnění nouzového tlačítka dojde k ukončení čekací



Obrázek 5.4.: Schéma zapojení nouzového tlačítka

smyčky, zároveň dojde k nastavení pomocné proměnné `stop_flag` na hodnotu "1" a program se vrátí do původní pohybové funkce. Po návratu se otestuje proměnná `stop_flag` a pokud je tato proměnná nastavena na "1", dojde ke zrušení aktuálně probíhající operace a rameno se vrátí do stavu *Stand-by*, kde čeká na novou příchozí instrukci. Průběh funkce je znázorněn diagramem na obr. 5.5.



Obrázek 5.5.: Vývojový diagram CENTRAL-STOP

6. Závěr

Cílem práce bylo vytvořit řídící systém pro robotické rameno s pěti stupni volnosti. Řídící systém implementuje základní komunikaci jak s uživatelem, tak s robotickým ramenem a ostatním hardwarem. Zároveň implementuje navržené matematické řešení inverzní kinematické úlohy pro převod Kartézských souřadnic na jednotlivé kloubové úhly robotu, implementuje algoritmus pro lineární pohyb ramene, disponuje vlastní instrukční sadou dle standardu G-code dostupnou uživateli pro manuální řízení a je vybaven demonstračními úlohami pro názornou ukázku funkce robotických ramen.

Nad rámec softwarového řešení bylo navíc realizováno fyzické modelové pracoviště sestávající z podkladové desky s Kartézskou soustavou souřadnic. Byl vyvinut vlastní signalaizační prvek ve formě 3D tištěného světelného majáku a pracoviště bylo vybaveno také bezpečnostním prvkem ve formě nouzového tlačítka. Vzniklo tak ucelené řešení, propojení vlastního řídícího softwaru a z části i vlastního hardwaru, které je vhodné pro názornou ukázku či výuku robotiky.

V průběhu realizace práce jsem narazil i na negativum. To spočívá v již samotných pohonech robotického ramene, ale v době výběru však bohužel žádné alternativní robotické rameno v uspokojivé cenové relaci nebylo k dispozici. Použité servomotory nedosahují požadované přesnosti natočení kloubů, mají výrazné vůle a výsledky inverzní kinematické úlohy tak neodpovídají realitě tak přesně. Provoz ramene tím ovšem není nikak omezen, rameno pouze nedosahuje takové přesnosti, jako v simulaci. Ověření správnosti výsledků proběhlo na digitálním modelu v aplikaci Autodesk Inventor. V budoucnu bych pro pohony raději využil krokové motory, které disponují vyšší přesností.

I přes toto negativum je ovšem celkový koncept funkční a splňuje zadaný cíl. Práce má potenciál i pro budoucí vývoj, je zde prostor například pro přidání již dříve zmíněné optické závory jako dalšího bezpečnostního prvku, vytvoření PC aplikace s intuitivním rozhraním umožňující uživateli ovládat rameno pohodlně a ne z konzole (instrukční sada s příkazy je vytvořena), či třeba připojení zpětné vazby a nasazení regulátoru pro korekci zmíněných nepřesností. Projektu bych se rád věnoval i nadále v navazujícím magisterském studiu zde na univerzitě.

Seznam použitých zdrojů

1. *Automatizace a automatizační technika*. Vyd. 1. Praha: Computer Press, 2000. ISBN 80-7226-249-1.
2. RUMÍŠEK CSc, Doc. Ing. Pavel. AUTOMATIZACE (roboty a manipulátory). In: [b.r]. Dostupné také z: https://ust.fme.vutbr.cz/tvareni/img/opory/emm_mechanizace_a_automatizace_roboty_rumisek.pdf.
3. *Factory automation*. FANUC Czech s.r.o., [b.r]. Dostupné také z: <https://factoryautomation.cz/zakladni-pojmy-z-automatizace-32-terminu-ktere-musite-znat/>.
4. ING. VÁCLAV HLAVÁČ CSC., prof. KOGNITIVNÍ ROBOTIKA. In: Fakulta elektrotechnická ČVUT v Praze, [b.r]. Dostupné také z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/51Robotika/71KognitivniRobotika.pdf>.
5. AG, KUKA. *Průmyslové roboty firmy KUKA*. Dostupné také z: <https://www.kuka.com/cs-cz/produkty,-slu%C5%BEby/robotick%C3%A9-syst%C3%A9my/pr%C5%AFmyslov%C3%A9-roboty>.
6. A.S., M.E.Z. POHONY. *OPRAVY PRŮMYSLOVÉ AUTOMATIZACE*. Dostupné také z: <http://www.mezpohony.cz opravy/>.
7. SOLUTIONS, Stateside Industrial. *Fluid Power: Pneumatic Actuators/Air Valve Control/Fittings*. Dostupné také z: <https://statesideindustrial.com/pneumatics>.
8. TECHNOLOGIES, BEKO. *Úprava stlačeného vzduchu bez kompromisů*. Dostupné také z: https://www.beko-technologies.com/fileadmin/beko-technologies.com/CZ/brochures_cz/competencefolders/77-00057_BEKO_Overview_CS_display_01.pdf.
9. PH.D., Ing. Vladimír Smutný. *Robotika - Úvod do kinematiky (přednášky)*. Centrum strojového vnímání, ČVUT.
10. ADAMEC CRANE SYSTEMS, S.R.O. *MOSTOVÝ JEŘÁB S PORTÁLOVÝMI NÁSTAVCI*. Dostupné také z: <https://www.adamec.cz/cs/aktuality/98-mostovy-jerab-s-portalovymi-nastavci>.
11. ING. MGR. VÁCLAV ZÁDA CSC., doc. *Robotika - přednášky*. Technická univerzita v Liberci.
12. BLUM, Jeremy. *Exploring Arduino: tools and techniques for engineering wizardry*. Wiley, [2013]. ISBN 978-1-118-54936-0.

13. *Electronics Hub.* Ravi Teja, 2021. Dostupné také z: <https://www.electronicshub.org/arduino-uno-pinout/>.
14. INC., Microchip Technology. *ATmega328P.* Dostupné také z: <https://www.microchip.com/en-us/product/ATmega328P#>.
15. O.O., RS Components Sp. z. *Vyhodnocovací sada, Tinkerkit Braccio Robot, Robotické rameno.* Dostupné také z: <https://cz.rs-online.com/web/p/shieldy-pro-arduino/1113738>.
16. CONRAD ELECTRONIC ČESKÁ REPUBLIKA, s.r.o. *Arduino stavebnice robotické ruky T050000 TinkerKit Braccio Robotic Arm T050000.* Dostupné také z: <https://www.conrad.cz/p/arduino-stavebnice-roboticke-ruky-t050000-tinkerkit-braccio-robotic-arm-t050000-2338896>.
17. EXP.COM, My Electrician. *Co je to servopohon a jak to funguje.* Dostupné také z: <https://my.electricianexp.com/cs/cto-takoe-servoprivod.html>.
18. DIGEST, Circuit. *What is a Servo Motor? - Understanding the basics of Servo Motor Working.* Dostupné také z: <https://circuitdigest.com/article/servo-motor-working-and-basics>.

Seznam obrázků

1.1.	Rozdělení robotů a manipulátorů [2]	16
1.2.	Průmyslové roboty KUKA [5]	17
1.3.	Elektrické krokové motory s rotačním výstupem [6]	19
1.4.	Pneumatické motory s přímočarým výstupem [7]	19
1.5.	Části robotu	20
1.6.	Portálový jeřáb [10]	21
1.7.	Ukázka více řešení inverzní úlohy [11]	22
2.1.	Schéma Arduino Uno, obrázek z [13] doplněn o vlastní popisky	24
2.2.	AVR ATMEGA 328P [14]	24
2.3.	Robotické rameno Tinkerkit Braccio [15]	26
2.4.	Obsah balení Tinkerkit Braccio [16]	26
2.5.	Servomotor [17]	27
2.6.	Zapojení servomotoru [18]	27
2.7.	Pulsně-šířková modulace a úhel natočení servomotoru [18]	28
2.8.	Servodriver Braccio servoshield, obrázek z oficiálního manuálu Tinkerkit Braccio .	29
3.1.	Robotické stanoviště	33
4.1.	Vývojový diagram hlavního programu	36
4.2.	Nelineární pohyb ramene	37
4.3.	Lineární pohyb ramene	38
4.4.	Matematický popis ramene (boční pohled)	39
4.5.	Výpočet přepony d_4 z x a y	40
4.6.	Úhel efektoru $q_4 > 90^\circ$	41
4.7.	Úhel efektoru $q_4 = 90^\circ$	42
4.8.	Úhel efektoru $q_4 < 90^\circ$	42
4.9.	Výpočet bodů P_2 a P_2'	44
4.10.	Výpočet úhlu q_0	45
4.11.	Výpočet úhlů q_1 – q_3	46
4.12.	Úhlové systémy	48
4.13.	Terminál řídicího systému	49
4.14.	Vývojový diagram G10/G11	50
4.15.	Vývojový diagram G92	52

4.16. Vývojový diagram G28	53
4.17. Vývojový diagram G90	55
5.1. 3D model světelného majáku	58
5.2. Schéma zapojení světelného majáku	59
5.3. Reálná podoba světelného majáku	59
5.4. Schéma zapojení nouzového tlačítka	61
5.5. Vývojový diagram CENTRAL-STOP	62

A. Externí přílohy

Externí přílohy této bakalářské práce jsou umístěny na adrese:

https://github.com/LukasBlk/bakalarska_prace_UJEP.

Na úložišti GitHub naleznete:

- **elektronickou verzi této bakalářské práce**
- **manuál pro obsluhu robotického ramene**
- **manuály pro spouštění jednotlivých ukázkových úloh**
- **videa jednotlivých ukázkových úloh**
- **zdrojový kód řídicího systému v C++ (Arduino Wiring)**
- **elektronická schémata zapojení HW komponent**

Struktura GitHub repozitáře:

BP_LukasBilek.pdf	text práce v PDF
BP_LB_Braccio_manual.pdf	uživatelský manuál k ovládání ramene v PDF
BP_schema_zapojeni	elektronická schémata zapojení periferií
BP_Firmware.cpp	zdrojový kód firmwaru ramene v C++
BP_UzivatelskeRizeni.mov	ukázkové video ovládání ramene skrz terminál
BP_G101.mp4	ukázkové video úlohy G101
BP_G102.mp4	ukázkové video úlohy G102
BP_G103.mp4	ukázkové video úlohy G103
BP_G104.mp4	ukázkové video úlohy G104
BP_G105.mp4	ukázkové video úlohy G105
