

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Robotický manipulátor
Dušan Vašek

Bakalářská práce
2019

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2018/2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Dušan Vašek**
Osobní číslo: **I16039**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**
Název tématu: **Robotický manipulátor**
Zadávající katedra: **Katedra elektrotechniky**

Zásady pro vypracování:

Navrhněte řídicí jednotku a mechanismus robotické ruky. Pohyb ruky bude zajištěn modelářskými servo pohony. Vytvořte program pro PC nebo smartphone, který bude s řídicí jednotkou komunikovat a předávat příkazy k ovládní ruky. Navrhněte možnosti řešení proti přetížení jednotlivých pohonů. Zařízení otestujte a ověřte jeho možnosti.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

[1] Atmel 8-bit Microcontroller with 4/8/16/32KBytes In-System Programmable Flash. ATMEL CORPORATION. [online]. [cit. 2013-10-29].

Dostupné z:

http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf

[2] VÁŇA, Vladimír. Mikrokontroléry ATMEL AVR: popis procesorů a instrukční soubor. 1. vyd. Praha: BEN - technická literatura, 2003, 335 s. ISBN 80-730-0083-0.

[3] MATRKA, Lukáš. Jednotka automatiky pro otevírání brány. Pardubice, 2014. Bakalářská práce. Univerzita Pardubice.

Vedoucí bakalářské práce:

Ing. Jiří Roleček

Katedra elektrotechniky

Datum zadání bakalářské práce:

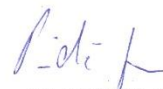
15. října 2018

Termín odevzdání bakalářské práce:

10. května 2019



Ing. Zdeněk Němec, Ph.D.
děkan



Ing. Jan Pídaníč, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2018

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice 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 Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 10. 5. 2019

Dušan Vašek

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat vedoucímu mé práce panu Ing. Jiřímu Rolečkovi za odborné konzultace a pomoc při realizaci. Také bych chtěl poděkovat panu Ing. Pavlovi Rozsívalovi za výrobu desky plošného spoje. Dále bych chtěl poděkovat mé rodině za podporu při psaní této bakalářské práce.

ANOTACE

Tato bakalářská práce je zaměřena na navrhnutí a realizaci manipulátoru. Pojednává o návrhu mechanických částí, řídicí elektroniky a počítačové aplikace. Práce obsahuje v teoretické části úvod do problematiky průmyslových manipulátoru. Praktická část se zaměřuje na návrh a konstrukci mechanických částí a řídicí elektroniky a sepsání příslušných programů.

KLÍČOVÁ SLOVA

ATmega32, roboty, manipulátor, C#, servomotory

TITLE

Robotic manipulator

ANNOTATION

This bachelor thesis is focused on design and realization of a manipulator. It deals with the design of mechanical parts, control electronics and computer applications. The thesis contains the theoretical part of the issue of industrial manipulators. The practical part focuses on the design and construction of mechanical parts and control electronics and the writing of relevant programs.

KEYWORDS

ATmega32, robots, manipulator, C#, servomotors

OBSAH

Seznam obrázků	9
Seznam tabulek	10
Úvod	12
1 Úvod do průmyslových robotů a manipulátorů	14
1.1 Důvody zařazování manipulátorů do výroby.....	16
1.2 Průmysl 4.0	18
1.3 Manipulátory v praxi	18
2 Problematika manipulátorů.....	19
2.1 Univerzální manipulátor	19
2.2 Způsoby programování manipulátorů.....	19
2.2.1 Přímé programování	20
2.2.2 Nepřímé programování	21
2.2.3 Ruční programování	21
2.3 Definice souřadnicového systému	21
2.3.1 Přímá kinematika	21
2.3.2 Nepřímá kinematika.....	21
2.4 Pohony pro manipulátory.....	21
2.5 Pohyby pohonu	22
2.6 Variabilní použití nástrojů	22
3 Praktická část.....	23
3.1 Návrh mechanizace	24
3.2 Servomotory.....	27
3.3 Ochrana proti přetížení	28
3.4 Procesor řídicí desky.....	28
3.4.1 Program procesoru	31
3.5 Řídicí počítačová aplikace	32
3.5.1 Metody pro výpočet polohy XYZ.....	34
3.6 Návrh DPS řídicí elektroniky	36

4	Praktické využití	39
4.1	Možnosti vylepšení	41
	Závěr	42
	Použitá literatura	43

SEZNAM OBRÁZKŮ

Obrázek 1 Rozdělení manipulátorů.....	14
Obrázek 2 Manipulátor od ABB [13].....	17
Obrázek 3 Navržený manipulátor	24
Obrázek 4 Základna manipulátoru	25
Obrázek 5 Horní kryt základny se servomotorem.....	25
Obrázek 6 První kloub manipulátoru	26
Obrázek 7 Druhý kloub a třetí kloub manipulátoru	26
Obrázek 8 Koncová hlava manipulátoru.....	27
Obrázek 9 Ukázka řezu 1 ramenem	27
Obrázek 10 Standartní zapojení ACS712 [8].....	28
Obrázek 11 ATmega32A [11].....	29
Obrázek 12 Možné rychlosti pro komunikaci přes USART [11]	30
Obrázek 13 Ukázka funkce FAST PWM [11]	31
Obrázek 14 Vývojový diagram programu procesoru.....	32
Obrázek 15 Ukázka PC aplikace.....	33
Obrázek 16 Pohled z boku	35
Obrázek 17 Pohled ze shora	35
Obrázek 18 Zapojení ACS712, vstupů a výstupů	37
Obrázek 19 Schématické zapojení procesoru	38
Obrázek 20 Schématické zapojení stabilizátoru	38
Obrázek 21 DPS řídicí desky	39
Obrázek 22 Vytvořený manipulátor.....	41

SEZNAM TABULEK

V bakalářské práci nejsou přiloženy žádné tabulky.

Seznam zkratek

PDF	Portable Document Format
PWM	Pulzně šířková modulace (Pulse Width Modulation)
USART	Synchronní/asynchronní sériové rozhraní
USB	Universal Serial Bus
DPS	Deska plošných spojů

ÚVOD

První známky automatizace v průmyslové výrobě můžeme objevit již ve středověku, ovšem automatizace, ve významu v jakém ji známe dnes, začala až ve dvacátém století. První průmyslový robot byl instalován v roce 1974 v továrně FANUS. Od instalace prvního robota se vývoj v oblasti automatizace výrazně zrychlil, díky čemuž mohlo být v roce 2008 instalováno ve světě 200 tisíc robotů. Průmyslové roboty mají významný podíl na růstu světových i domácích ekonomik.

Cílem mé práce je navrhnout manipulátor, který bude ovládán PC aplikací a ověřit jeho možnosti využití. Součástí práce bude i samotné sestavení manipulátoru ze součástek, které navrhnu a vytisknu na 3D tiskárně. Současně sestrojím řídicí desku, která bude obsluhovat jednotlivé pohony manipulátoru a navrhnu opatření proti přetížení. Vytvořím počítačovou aplikaci, prostřednictvím které se budou manipulátoru zadávat pokyny. Na aplikaci si ověřím možnosti využití sestaveného mechanismu.

V teoretické části se nejdříve bude zabývat v rámci první subkapitoly průmyslovými roboty a manipulátory. Zde popíšu jejich základní vlastnosti, jejich druhy a možnosti jejich využití v průmyslu. Zaměřím se i na jejich základní řazení a odůvodnění jejich nezbytnosti v dnešní průmyslové výrobě. Součástí tohoto teoretického bloku bude i zmínka o čtvrté průmyslové revoluci, pro kterou se roboti staly nepostradatelnou součástí. Druhá subkapitola bude již zaměřena na podrobnější problematiku manipulátorů a možnosti jejich programování. V kapitole nejdříve popíšu teoretické příklady, od kterých přejdu i k příkladům z praxe.

Následovat bude praktická část bakalářské práce, která bude rozdělena do tří subkapitol. V první z nich navrhnu mechanismus manipulátoru, rozkreslím všechny jednotlivé součástky, jejichž rozměry vložím do 3D softwaru pro modelaci. Výsledné modely součástek následně přenesu do softwaru obsluhující 3D tiskárnu, pomocí které zmíněné součástky postupně vytisknu. Následovat bude sestavení samotného manipulátoru a jeho umístění na pracovní desku.

Druhá subkapitola bude věnována řídicí desce. Zde navrhnu schéma zapojení řídicí desky, současně i vhodné rozložení součástek na DPS. Následně řídicí desku vyrobím a sestrojím dle mnou navrženého schématu a rozložení. Na řídicí desku použiji procesor ATmega32A, který je dle mého uvážení vhodný a dostačující pro správné fungování manipulátoru. V rámci této subkapitoly se budu zabývat také problematikou přetížení jednotlivých pohonů. Navrhnu možnosti řešení, které zabrání přetížení a následného zničení pohonů.

V rámci třetí subkapitoly se budu věnovat návržení vhodného programu pro komunikaci s řídicí deskou, pro vytvoření programu využiji programovací jazyk C#. Pro napsání kódu aplikace využiji programovací software Visual Studio od společnosti Microsoft, se kterým již mám zkušenosti.

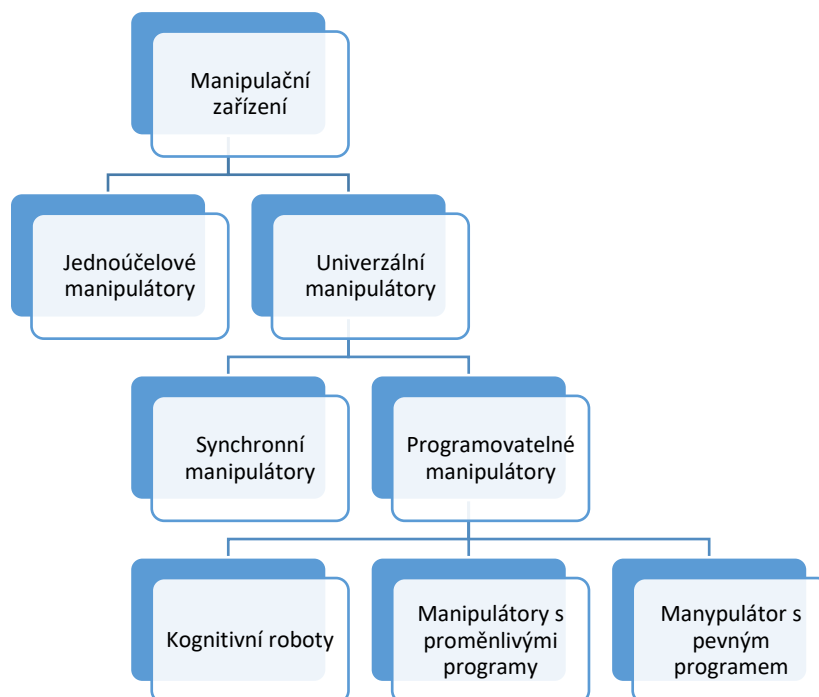
Po dokončení všech tří částí, tedy sestrojení mechanismu manipulátoru, řídicí desky a zprovoznění počítačové aplikace, vyzkouším funkčnost a možnosti manipulátoru. Na závěr zhodnotím všechny mnou navržené aspekty manipulátoru a popíšu možnosti vylepšení.

1 ÚVOD DO PRŮMYSLOVÝCH ROBOTŮ A MANIPULÁTORŮ

V rámci této kapitoly se budu věnovat úvodu do problematiky průmyslových robotů a manipulátorů. Stručně zde shrnu historii vývoje průmyslových manipulátorů a robotů a nastíním možnosti jejich členění, dále se budu věnovat výhodám zavedení manipulátoru do průmyslové výroby.

Hlavním důvodem všech průmyslových revolucí a vynálezů byl neustálý vývoj člověka, který se snažil usnadnit si práci, snažil se o přesnější, dokonalejší, rychlejší a efektivnější výrobu. Tento vývoj lidstvu zaručil neustálý posun kupředu. Každá průmyslová revoluce přinesla své, a ta čtvrtá stvořila roboty, aby mohli nahradit lidi ve výrobním procesu, ale cesta to nebyla jednoduchá. Dříve než se mohlo začít mluvit o robotech, kteří by připomínali člověka, musely vzniknout automaty, funkčně úzce specializované manipulátory, ze kterých se postupně vyvíjeli univerzálnější manipulátory, od kterých se došlo až ke kognitivním robotům. Významnou událostí pro vývoj manipulátoru znamenal vynález číslicového řízení v polovině 20. století. Tento vynález pomohl realizovat představy o automatické výrobě. [1]

Jako nejvýhodnější náhradu za člověka se ukázaly být právě průmyslové roboty a manipulátory. Průmyslové roboty a manipulátory jsou navrženy tak, aby mohly vykonávat pohyby a manipulace. Tyto stroje se rozdělují podle svých charakteristických a konstrukčních vlastností a dle toho jsou řazeny do tříd. [1] [2]



Obrázek 1 Rozdělení manipulátorů

Schéma číslo 1 představuje rozdělení manipulačních zařízení. Nejdříve je členění na jed-
nouúčelové a univerzální manipulátory. Jednouúčelové manipulátory jsou roboty určené k pořád
se opakující činnosti. Tedy mohou sloužit jako podavače, přidržovače. Nemusí se vždy jednat
pouze o jednoduchý stroj, v některých případech i mechanicky velmi složitý. [1]

Univerzální manipulátory se dále dělí na synchronní a programovatelné manipulátory, kde
synchronní manipulátor pracuje synchronně s uživatelem. Tento stroj tedy stínuje pohyby uží-
vatele přes konzoli nebo „loutku“ (přístroj hodně podobný manipulátoru, prostřednictvím kte-
rého se předávají informace o pohybu, nemá akční prvky sloužící k pohybu). Tyto manipulátory
se také označují jako „master-slave“, tedy „master“(uživatel) dává příkazy „slave“ a ten je oka-
mžitě vykonává. Příkladem jsou roboty, které fungují jen s koexistencí uživatele. [1] [6]

Programovatelné manipulátory konečně dělíme na manipulátory s pevným programem,
s proměnlivými programy a kognitivní roboty. V případě manipulátoru s pevným programem
jsme nuceni dle potřeby výroby měnit kompletní program, naopak při využití manipulátorů
s proměnlivými programy, můžeme libovolně pomocí konzole měnit koeficienty a vnitřní na-
stavení, což umožňuje flexibilnější použití průmyslových robotů. Výhodou kognitivních robotů
je pak umělá inteligence, která jim umožňuje adaptivně se přizpůsobovat danému úkolu.
V rámci své práce se budu podrobněji zabývat právě manipulátorem s proměnlivým progra-
mem, tedy průmyslovým robotem. [1] [6]

Kromě členění dle schématu č. 1 lze posuzovat manipulátory a roboty dle následujících
aspektů:

- Morfologie robota (architektura) - Jakým stylem je řešená kinematická struktura robota. [1]
- Počet stupňů volnosti – Je odvozeno od počtu pohyblivých kinematických prvků. [1]
- Velikost a hmotnost robota – Zde jsou posuzovány rozměry a váha robota. Robot by měl být co nejlehčí při zachování potřebné pevnosti a tuhosti. [1]
- Velikost obsluhovaného prostoru – Zaleží na konstrukci, délce ramen, stupňů volnosti a kinematické struktuře. [1]
- Hmotnost břemene – Je jeden z parametrů, podle kterých se konstruuje a přizpůsobují ostatní prvky. Určuje se podle toho, jakou váhu uzvedne koncový prvek (efector). [1]
- Dosahovaná přesnost – Je velmi důležitý aspekt. Dosahovaná přesnost se porovnává buď se zatížením koncového prvku nebo bez. Dále je to, s jakou přesností

dokáže opakovat pohyby opět jak se zatížením, tak bez. Toto nejvíc řeší softwarová část, která upravuje rychlosti podle zatížení, a tím zvyšují přesnost. [1]

- Rychlost pohybu – Pověšinou je upravitelné. Záleží, jakou práci robot bude vykonávat. Hledá se zde kompromis, aby byla zachována přesnost a úkon netrval zbytečně dlouho. [1]
- Způsob pohybu – Zde záleží na programátorovi, jaký pohyb bude robotu umožněn. [1]
- Druh servomotoru – Zde už záleží na vnitřním řešení robota. Používané pohony jsou mechanické, pneumatické, hydraulické, elektrické a kombinované. [1]
- Způsob odměřování – Obdobně jako o druhu servomotoru. [1]
- Způsob a rozsah vnímání – Záleží, jakými senzory je manipulátor vybaven a jak je schopný data se senzorů zpracovat. [1]
- Způsob řízení a komunikace s okolím – Závisí na aplikaci a jejím propojení s hardwarem. [1]
- Autonomnost robotů – Čím větší je možnost vnímání, tím více se může robot rozhodovat sám. Příkladem jsou kognitivní roboti. [1]

1.1 Důvody zařazování manipulátorů do výroby

Jak jsem již zmiňoval v úvodu, hlavním důvodem zařazování manipulátorů do výroby je nahrazení lidského kapitálu ve výrobním procesu. Příčinou robotizace byl stále zvyšující se tlak na rychlejší, přesnější a kvalitnější výrobu, a těmto požadavkům lidský kapitál již nebyl schopný vyhovět.

Hlavní nevýhodou lidského kapitálu oproti manipulátorům je především vyšší chybovost, a také omezená doba výroby. Manipulátory budou svou práci vykonávat dokola, neúnavně, dokud jim někdo nedá pokyn přestat. Kdežto v případě lidského kapitálu jsme omezeni jejich pracovní dobou a navíc když k tomu přičteme dnešní problémy s nízkou nezaměstnaností, která může způsobit výluky ve výrobě, máme další výhody pro manipulátory. [2]

V souvislosti s lidským faktorem musíme taktéž zmínit mzdové náklady, které právě díky nízké nezaměstnanosti v posledních letech výrazně vzrostly, a nelze předpokládat jejich snížení. Naopak náklady na manipulátory lze považovat za konstantní, hlavní vliv na jejich růst má pouze cena energií. Kdežto vliv na růst minimální mzdy může být i politický apod. [2]

Zavedení manipulátorů do výroby se nesníží pouze mzdové náklady výrobních pracovníků, ale umožní i ušetření nákladů na administrativu spojenou s výrobními zaměstnanci (např.: náklady na personalistiku a řízení lidských zdrojů, mzdová účetní, kontroloři apod.)

Je nesporné, že zavedením manipulátorů se průmyslová výroba zefektivnila. Manipulátory jsou navíc mnohonásobně přesnější, což je v kritických místech technologických postupů náchylných na přesnost a rychlost velmi výhodné. Tímto se kritická místa výroby dostávají pod velkou kontrolu a snižuje se podíl vadných výrobků. Snížením vadných výrobků se snižují i náklady na materiál.

Nevýhodu zavádění manipulátorů je možné spatřovat ve vysokých počátečních investicích, kdy pořizovací cena robotů, úprava pracoviště, nákup technologií a najmutí specializovaných zaměstnanců na programování a údržbu je zásadní položkou v rozpočtu každého podniku. Nicméně během několik let se tato počáteční investice vrátí.



Obrázek 2 Manipulátor od ABB [13]

1.2 Průmysl 4.0

Smysl průmyslu 4.0 spočívá v tom, že výroba ze samostatných poloautomatických linek se přesune na plně automatizovanou výrobu a průběžně optimalizované výrobní prostředí. Měly by tak vzniknout nové kybernetické fyzikální systémy, vzniknou propojováním výrobních zařízení. Tyto systémy budou základem pro inteligentní továrny. Továrny mají být schopny autonomní výměny informací, a vyvolání příslušných reakcí na tyto informace. V těchto továrnách tak budou vznikat přesně identifikované produkty, o kterých bude známa celá jejich historie, a to od materiálu po smontování. [2]

Průmysl 4.0 lze shrnout:

- „Výrobní procesy jsou optimalizované v rámci celého hodnotového řetězce díky vertikálně i horizontálně integrovanému IT systému.“ [2]
- „Izolované výrobní jednotky jsou nahrazeny plně automatizovanými a vzájemně propojenými výrobními linkami.“ [2]
- „Flexibilní výrobní procesy umožňující efektivní výrobu i malých výrobních dávek.“ [2]
- „Vzájemně komunikující roboty, výrobní zařízení a výrobky činí do jisté míry autonomní rozhodnutí v reálném čase a tím zvyšují svoji flexibilitu a efektivitu výrobního procesu.“ [2]
- „Výrobní prostředí se samo optimalizuje a konfiguruje.“ [2]

1.3 Manipulátory v praxi

V současné době na trhu dominují dvě firmy zabývající se technologiemi zmíněnými v předchozím textu. Jedná se o společnosti KUKA a ABB. Společnost KUKA působí na trhu už více než sto let, společnost byla založena v roce 1898 v Augsburgu (Německo), jako acetylenový závod na výrobu osvětlení. Svou historii jako průkopník robotiky začala společnost psát v roce 1973, kdy vyvinula FAMULA (prvního průmyslového robota). Roku 2007 se následně zapsala do Guinnessovy knihy rekordů prostřednictvím svého v té době největšího průmyslového robota. Tato společnost se díky své minulosti stala nedílnou součástí a budoucností pro průmysl 4.0. Druhou ale výrazně mladší společností podnikající v této oblasti je společnost ABB, která má pobočky po celém světě, nicméně její hlavní sídlo je v Curychu (Švýcarsko), tato společnost klade důraz především na výzkumné a vývojové činnosti. V praxi se objevují manipulátory ve velkých počtech na specializovaných robotizovaných pracovištích, kde vykonávají činnosti jako lepení, sváření, nebo lakování. [7] [12] [13]

2 PROBLEMATIKA MANIPULÁTORŮ

V této kapitole přiblížím problematiku manipulátorů a nastíním její řešení z praxe. U manipulátoru se setkáme s mnoha úskalími, které pro správný chod a bezproblémové řízení je potřeba řešit. Patří sem zejména vyřešení vhodného výběru pohonu, řízení a kinematiky. Při nedbalosti manipulátor nebude schopen plnit příkazy a bude mít různá omezení. U špatného výběru pohonů nemusí mít manipulátor schopnost unést těžší břemeno nebo vlastní nástroj, v horším případě nemusí unést ani svojí vlastní váhu. Špatně spočítané vzdálenosti a úhly zase znemožní správně vypočítat trajektorii, tedy nástroj nikdy nezaujme požadovanou polohu.

2.1 Univerzální manipulátor

Je to automatický stroj, který plní předem definovaný program ze své paměti, nebo je přímo ovládaný z uživatelského prostředí. Tento stroj plně nahrazuje člověka při jednoduchých i složitých technologických postupech. Univerzální manipulátor většinou kolaboruje buď s dalším univerzálním robotem, nebo s jednoúčelovým manipulátorem, který slouží na upevnění předmětu pro potřeby univerzálního manipulátoru. [1]

Skládá se z podstavy, soustavy ramen a koncového prvku. Podle počtu kloubů mezi jednotlivými rameny určujeme počet stupňů volnosti. To znamená, jak se dá všelijak naklonit koncový prvek manipulátoru vůči základně. Koncový prvek je hlava s nástrojem buď pevně usazeným, nebo vyměnitelným. Nástroje se vyměňují ručně pomocí operátora výroby nebo sám manipulátor ze zásobníku s nástroji. [1]

Manipulátor povětšinou provádí jednu činnost podle nastaveného programu, avšak tento program se může i s ostatními parametry (např. nový nástroj v zásobníku) měnit v uživatelském prostředí. Možnosti uživatelského prostředí se liší na základě požadavku provozovatele a návrhu výrobce. [1]

2.2 Způsoby programování manipulátorů

Možnost programování u univerzálních manipulátorů je podstatná záležitost. Je to jeden z aspektů, jež je odlišují od jednoúčelových manipulátorů. Jedná se o data, podle kterých se manipulátor orientuje a vykonává zadané příkazy.

Programovacích jazyků existují spousty a vždy se tento jazyk odvíjí od daného výrobce. Například výrobce KUKA používá u svých manipulátorů programovací jazyk KRL a výrobce ABB používá jazyk RAPID. Tyto jazyky jsou sice odlišné ale v závěru založené jsou založené na stejném principu. [1] [5]

Příklady instrukcí RAPID

- Robtarget – definuje polohu robota v prostoru [1]
- Wobjdata – definuje polohu souřadnicového systému [1]
- Tooldata – definuje používaný nástroj a souřadnicový systém [1]
- Move() – definuje pohyb [1]

Programování tedy probíhá skrz programovací jazyky. Obsluha programuje různými způsoby. Nejpoužívanějšími metodami v dnešní praxi jsou programování online a off-line. Při metodě online obsluha programuje manipulátor na místě, buď pomocí počítače nebo za pomoci ručního ovládacího panelu. Na ovládacím panelu je nejčastěji umístěn displej pro lehčí orientaci, klávesnice a řada tlačítek s různými funkcemi opět pro usnadnění a urychlení programování. [1] [5]

Metoda off-line je založena na existenci softwaru s 3D virtuální simulací konkrétního robotizovaného pracoviště. Tato metoda dovoluje definovat dráhy a pohyby robota, výsledný soubor se pak může jen nahrát do příslušného programovacího jazyka. Výhodou této metody je, že v průběhu vytváření programu nemusíte mít příslušného robota k dispozici. V dnešní době se projevuje snaha o propojení obou metod a využití jejich dílčích vlastností. [1] [4]

Samotné programování dráhy manipulátorů se obecně dělí na tři typy.

2.2.1 Přímé programování

Přímé programování se provádí snímáním pohybu jednotlivých ramen, kterými uživatel pohybuje do potřebných míst. Toto programování se ještě dělí na dva typy.

V případě metody „Teach-in“ uživatel pomocí ovládací konzole navede robota na požadované místo a toto místo robotovi zapíše do paměti. Dá se tím dosáhnout velké přesnosti. Tato metoda nedefinuje dráhu robota, a proto uživatel kromě požadovaných bodů ukládá i pomocné body, robot pak jen najíždí na pomocné body, anebo si svoji dráhu se znalostí pomocných a koncových bodů aproximuje. [1] [4]

Metoda „Play-Back“ snímá kromě průběhu drah i rychlost pohybu. Díky tomu je tato metoda vhodná na operace, u kterých je potřeba velká přesnost a optimální rychlost. Hodí se například tehdy, kdy je nutné kopírovat nějaký rozličný povrch nebo v místech se špatným přístupem. [1] [4]

2.2.2 Nepřímé programování

Při tomto jsou programovány křivky, které robot používá jako svoji dráhu. Dosahuje se velké přesnosti a robot přesně kopíruje daný objekt. S přesností souvisí obtížnější vypočítání křivek. [1]

2.2.3 Ruční programování

Přes uživatelskou aplikaci se ručně zadávají hodnoty. Hodnoty mohou být ve formátu koncových souřadnic v kartézských souřadnicích nebo v úhlech natočení jednotlivých servomotorů. Robot se poté řídí těmito hodnotami. [1]

2.3 Definice souřadnicového systému

Definice souřadnicového systému se odvíjí od způsobu programování manipulátoru. Tyto definice jsou nejvíce potřeba u ručního programování, kdy uživatel musí mít přehled o přesné poloze koncového bodu neboli efektoru. K tomu je potřeba jednoduchý souřadnicový systém. Nejužívanější je kartézský. V tomto systému je lehká orientace a uživatel vždy vidí koncový bod, ať zadává přímo koncový bod nebo úhly natočení servomotorů. Ovšem při zadání koncového bodu se neví nic o úhlech natočení jednotlivých servomotorů a naopak u zadávání natočení servomotorů se neví nic o poloze koncového bodu v kartézské soustavě souřadnic. Tuto problematiku řeší přímá a nepřímá kinematika. [1] [4]

2.3.1 Přímá kinematika

Tato metoda spočívá ve znalosti polohy koncového bodu a jeho orientaci. Z těchto znalostí se vypočítávají vztahy mezi jednotlivými členy systému. Výpočty jsou jednoduché a jsou založené na goniometrických funkcích. Může se stát, že existuje více cest k cíli a je na programu, kterou cestu zvolí. Ne vždy to může být optimální řešení. [1]

2.3.2 Nepřímá kinematika

Nepřímá kinematika má za úkol ze znalosti úhlového natočení jednotlivých ramen spočítat polohu koncového bodu. Čím více je stupňů volnosti, tím je výpočet složitější. [1]

2.4 Pohony pro manipulátory

Pro manipulátory se dají optimálně využívat krokové motory, servomotory nebo jejich kombinace. Při výběru správného pohonu se musí zejména dbát na moment síly, které musí

pohony generovat, přičemž nesmí dojít k dlouhodobému přetěžování. Dále se musí hledět na hmotnost, rozměry a v neposlední řadě na cenu. Nesmí se přehlédnout ani umístění pohonu.

Hmotnost pohonu se nesmí zanedbat z důvodu navýšení váhy celého manipulátoru, kde se musí počítat s tím, že první kloubový pohon musí mít takový moment síly, aby se dokázal jemně otočit i s nataženými rameny manipulátoru. Pohon taky musí rameno udržet v ustálené poloze. [1] [7]

2.5 Pohyby pohonu

Řízení pohonu probíhá buď na základě přímých příkazů z aplikace, a to jednotlivým pohonům nebo jsou ovládány skrz už vypočtené dráhy z výpočtu z kinematiky. Mnohdy je ale důležité kromě koncových bodů řešit i dráhu hlavy manipulátoru, třeba z důvodu střetu s předměty. [1]

Základní přehled pohybů.

- Obecný pohyb – U tohoto pohybu zná manipulátor jen dva koncové body. Mezi těmito body přejeđe nejrychlejší možnou cestou. Cesta ale určitě není nejkratší a je popsána náhodnou křivkou, která vyjde až poté, co robot daný pohyb udělá. [1]
- Lineární pohyb – U tohoto pohybu jsou také známy jen dva koncové body. Ovšem matematickými výpočty se upraví křivka na lineární pohyb. Zde se najde nejkratší cesta na úkor rychlosti. Uživatel má alespoň základní přehled o tom, kde se manipulátor bude pohybovat. [1]
- Kruhový pohyb – Tento pohyb si nevystačí pouze se dvěma koncovými body, ale musí se mu také zadat pomocný bod, který vymezuje kruhový pohyb mezi nimi. [1]

S těmito pohyby se v praxi nevystačí. Pro mnoho praktických úloh jsou nedostatečné. Kolektivními vlastnostmi jsou jeden počáteční bod, koncový a v případě kruhového pohybu jeden pomocný bod. Když najedou do koncového bodu, jejich pohyb se zastaví a následně pokračují na další instrukci. To je v mnoha případech nežádoucí. Řešením je zde aproximace, která dráhu tvoří plynulejší a rychlejší.

2.6 Variabilní použití nástrojů

Jak bylo již zmíněno, manipulátor tvoří kromě pohonů, základny a jednotlivých ramen také koncový prvek efektor. Tento efektor odpovídá úloze, pro kterou byl daný manipulátor zkonstruován. Efektor plní většinou tyto typy úloh:

- Vkládání objektů
- Mezioperační manipulace
- Technologické operace
- Kontrolní operace
- Speciální práce

Z technologického hlediska se efekторы dělí na

- Technologické
- Manipulační
- Kombinované
- Speciální

Na manipulátorech může být efektor pevně přidělán, ručně vyměnitelný nebo obsahuje zásobník, kde si manipulátor automaticky vyměňuje nástroje. [1]

S použitím zásobníku je potřeba speciální efektor s uchopovacími vlastnostmi, ať už jsou mechanické, magnetické, podtlakové nebo speciální. Mnohdy je taky podstatné zajistit kalibraci těchto nástrojů, hlavně u těch obráběcích. [1]

3 PRAKTICKÁ ČÁST

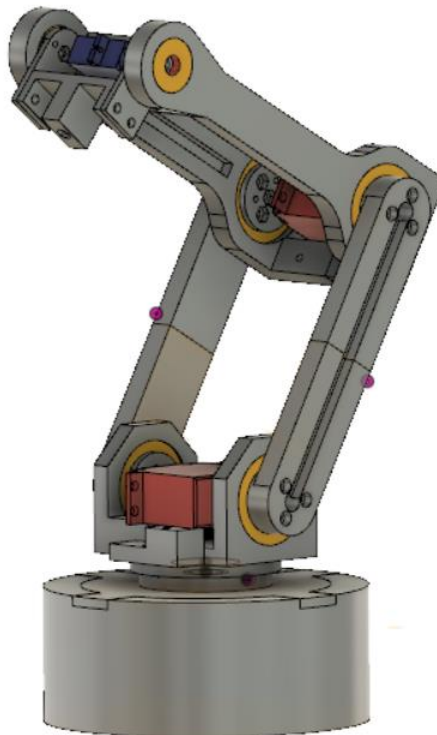
Tato část práce se soustředí na její hlavní cíl, a to navrhnout a sestrojít univerzální manipulátor. Tento manipulátor by se měl dále řadit do třídy asynchronních manipulátorů. V jednotlivých podkapitolách budou řešeny už probrané problematiky na sestrojovaný manipulátor.

Počáteční návrh je manipulátor se čtyřmi stupni volnosti. První stupeň bude servomotor umístěný v základně, který bude zajišťovat rotační pohyb zbytku mechanismu. Následující stupně budou vždy servomotory umístěné v klouby. Klouby budou mezi sebou spojeny rameny. Na posledním rameni bude jednoduchý systém na vyměnitelnou hlavu pro ruční výměnu nástrojů. Na ověření vlastností manipulátoru bude zkonstruována efektor s osazenou obyčejnou tužkou. Manipulátor se bude ovládat přes počítačovou aplikaci napsanou v jazyce C#. Tato aplikace bude generovat instrukce pro řídicí desku. Program na řídicí desce tyto instrukce bude muset spravovat a podle toho nastavit manipulátor do požadovaného rozpoložení. Bude se jednat o manipulátor s proměnlivým programem. Při konstrukci manipulátoru budu dbát hlavně na jednoduchou konstrukci, která se bude dít snadno smontovat a dále upravovat pro budoucí vylepšení.

3.1 Návrh mechanizace

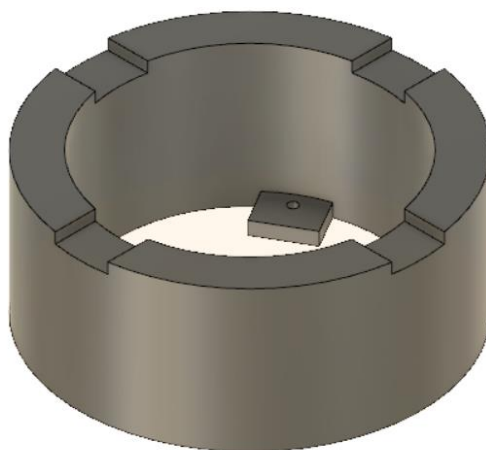
Návrh mechanizace byl vytvořen v softwaru Fusion360 od společnosti AutoCad. Jedná se o cloudovou 3D modelovací aplikaci. Určený je pro vytváření prototypů od vytváření základních modelů až pro složité součásti složené z mnoha modelů. Dále má skvěle propracované exportování pro potřeby 3D tisku, kde se dají nastavovat základní vlastnosti. Software obsahuje i integrované nástroje pro přípravu výroby a nástroje pro simulace.

Manipulátor byl navržen tak, aby byl lehký a snadno vyrobitelný. Všechny díly včetně servomotoru a ložisek byly vymodelovány v programu Fusion360. Vymodelované díly servomotoru a ložisek byly použity pro modelování ostatních dílů manipulátorů. Tento postup velice urychlil vývoj celého manipulátoru a zamezil vzniku nepřesností. Díly určené na výrobu byly vytisknuty na 3D tiskárně obdobné jako Original Prusa I3 MK3S. Usazení ložisek se provádělo lisováním do nahřátých plastových dílů, což zcela zabránilo vzniku vůle.



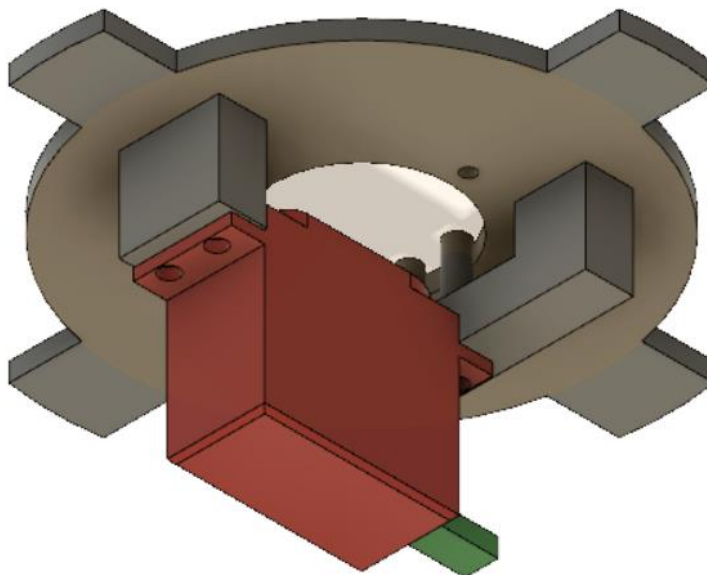
Obrázek 3 Navržený manipulátor

Na obrázku je vidět navržený manipulátor, který obsahuje všechny konstrukční prvky, včetně servomotorů a ložisek. Jediné, co v modelu chybí, je kabelový svazek, který bude procházet skoro celým manipulátorem. Ovšem při modelování na něj nebylo zapomenuto a je pro něj připraveno vhodné vedení.

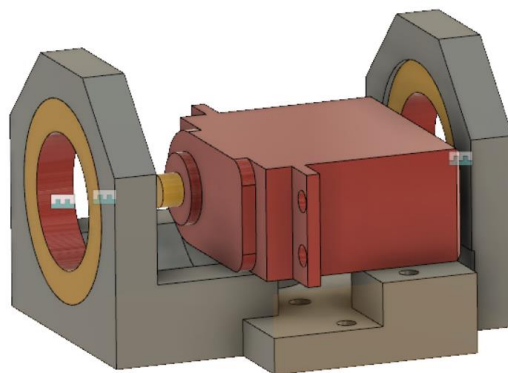


Obrázek 4 Základna manipulátoru

Základna se upevňuje na pracovní stůl. Uvnitř základny se nachází servomotor přimontovaný na vrchní kryt, který obstarává rotační pohyb celého manipulátoru vůči rovině. Mezi základnou a prvním kloubem je umístěno ložisko. Skrz základnu prochází kabelový svazek z prvního kloubu do řídicí desky.

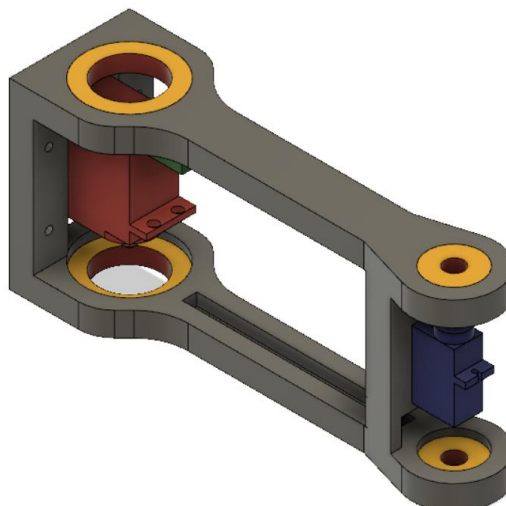


Obrázek 5 Horní kryt základny se servomotorem



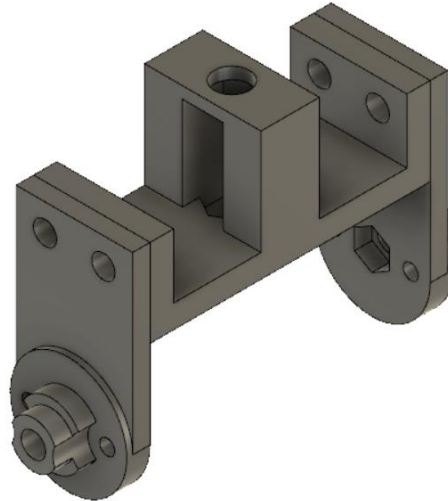
Obrázek 6 První kloub manipulátoru

Na obrázku je zobrazen první kloub manipulátoru s osazením servomotorem a ložisky. Pod servomotorem je umístěn otvor pro kabelový svazek, který vede do základny, ten pak pokračuje skrz ložisko do drážky v rameni a následně do druhého kloubu.



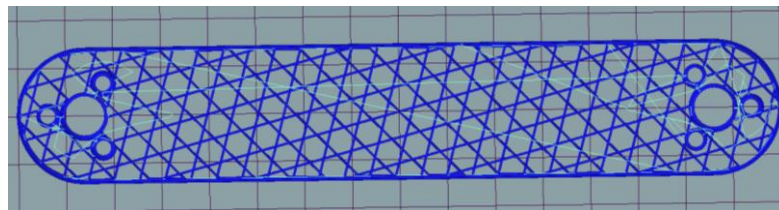
Obrázek 7 Druhý kloub a třetí kloub manipulátoru

Tento obrázek ukazuje osazení druhého a třetího kloubu servomotory a ložisky. Kabelový svazek zde projde opět ložiskem a až ke druhému servu. Pro snazší výrobu byla součást rozdělena na dvě nezávislé části a spojena součástkou, která plní i vlastnost aretace servomotoru. Tato součástka je pouze obdélník se dvěma širšími otvory, zde nebude zobrazena.



Obrázek 8 Koncová hlava manipulátoru

Celý manipulátor je vytisknutý z materiálu PLA. Jedná se o biologicky rozložitelný plast vyrobený z kukuřice, cukrové třtiny nebo brambor. Je to poměrně pevný materiál, který vydrží teploty do 70°C, poté tento plast začne ztrácet svoji pevnost a měkne. Všechny součásti byly tisknuty na stejné nastavení. Výška vrstvy 0,4mm s tloušťkou stěny 2 perimetry (tzn. kolik vláken tvoří tloušťku stěny) a dvacetiprocentní vnitřní vyplň.



Obrázek 9 Ukázka řezu 1 ramenem

Na obrázku je vidět řez prvním ramenem. Řez je veden ve čtvrté vrstvě a můžeme na něm vidět vnitřní systém podpor, který zaručuje, že se součástky nezlomí nebo jinak nezničí.

3.2 Servomotory

Pro pohon byly vybrány modelářské servomotory MG995 a SG90. Oba servomotory mají úhel natočení 180° a oba se ovládají PWM signálem generovaným mikroprocesorem. Úhel se nastavuje posláním PWM pulzu o určité délce a to 0,5ms až 2,5ms, kde 0.5ms znamená nulový úhel a 2,5ms znamená plný úhel, tedy 180°. [5]

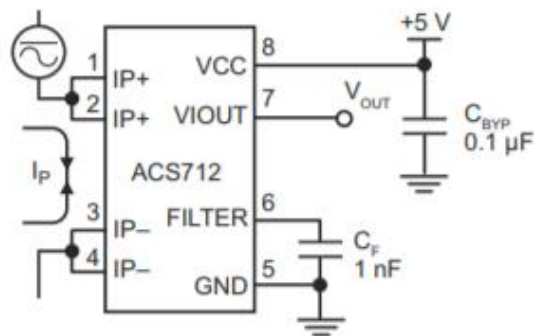
Servomotor MG995 má mít při 4,8V moment 9,4Kg při 6,6V má mít až 11Kg, což je dostatečný moment pro hlavní klouby manipulátoru.

Servomotor SG90 má moment okolo 2,5Kg a váhu 15g. Tento servomotor se ideálně hodí na použití v posledním kloubu manipulátoru, je lehký a moment síly má dost velký na to, aby mohlo manipulovat s posledním ramenem na kterém bude umístěna koncová hlava.

3.3 Ochrana proti přetížení

Ochrana proti přetížení je řešena pomocí sledování velikosti proudu, který jde do jednotlivých servomotorů a následně se tento proud vyhodnocuje. Tímto postupem by se zabránilo zničení jednotlivých komponent manipulátoru.

Pro tento návrh jsem zvolil integrovaný obvod ACS712, který má v sobě integrovaný Halův senzor.



Obrázek 10 Standartní zapojení ACS712 [8]

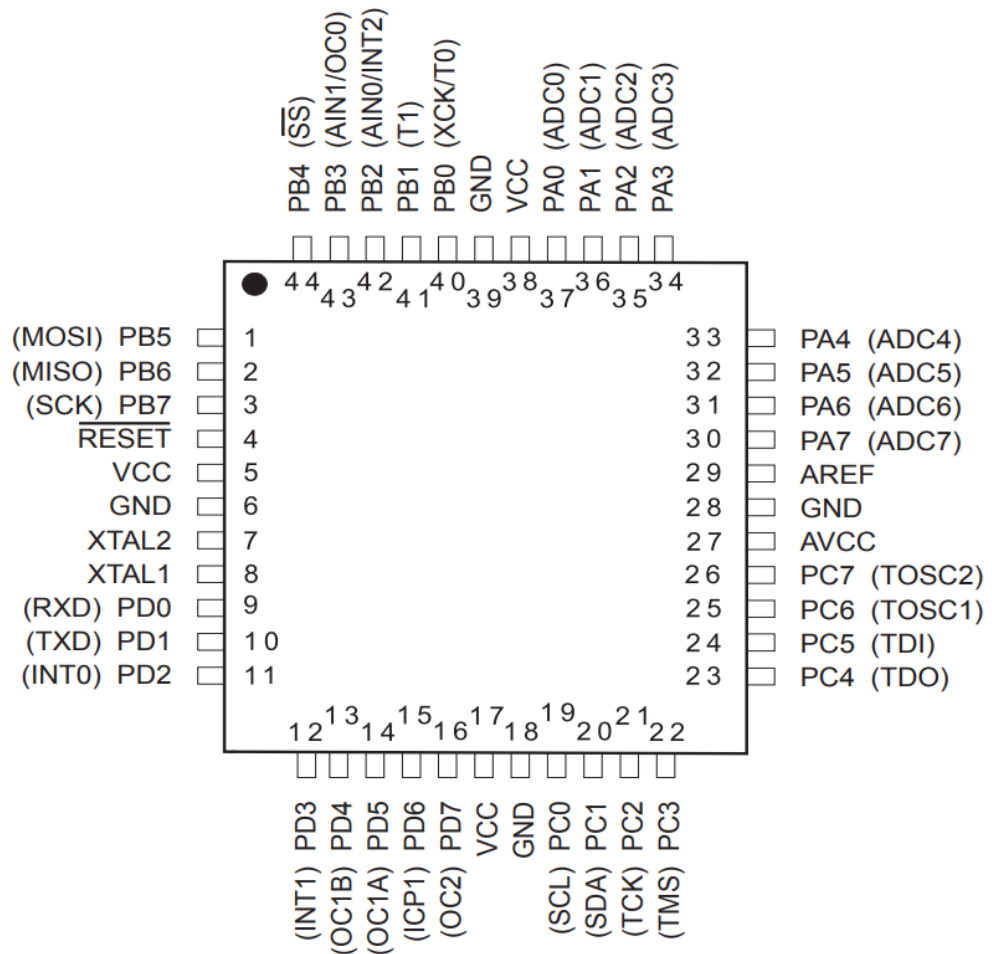
Na obrázku je vidět základní zapojení integrovaného obvodu ACS712, který je použit. Halův senzor snímá elektromagnetické pole generované procházejícím proudem mezi kontakty IP+ a IP-, velikost tohoto pole převede jako velikost napětí na kontakt Viout. Tuto analogovou hodnotu přivedeme na A/D převodník mikroprocesoru a dále zpracujeme. [8]

3.4 Procesor řídicí desky

Na řídicí desku manipulátoru byl vybrán procesor ATmega32A v zapouzdření TQFP44. Jedná se o osmi bytový procesor rodiny AVR od firmy Atmel z řady ATmega.

Tento procesor má k dispozici 32kB programovatelnou FLASH paměť, 2kB paměti SRAM a 1kB EEPROM. Pro komunikaci s ostatními zařízeními obsahuje rozhraní Inter Integrated Circuit(I2C), sériové rozhraní USART a SPI (Serial Peripheral Interface). Dále obsahuje jeden šestnácti bitový časovač a dva osmi bitové. Čtyři výstupy PWM, osm deseti bitových analog digitálních převodníků a integrovaný osmi MHz oscilátor. Umožňuje také připojení externího oscilátoru. [11]

Procesor byl vybrán kvůli tomu, že plně postačuje k obsluze všech požadovaných periférií. Tedy dokáže přes sériové rozhraní komunikovat s počítačovou aplikací, dokáže obsluhovat čtyři servomotory a obsahuje čtyři A/D převodníky ke snímání napětí s obvody ACS712.



Obrázek 11 ATmega32A [11]

Pro potřeby manipulátoru bude použit USART pro komunikaci řídicí aplikace z PC, všechny PWM výstupy na ovládání servomotory a čtveřici A/D převodníku, které budou připojeny na výstup z integrovaného obvodu ACS712. Dále bude využit interní oscilátor bez děličky. [11]

Komunikace přes USART umožňuje pracovat v duplexním režimu, může tedy najednou přijímat i odesílat data. Také nabízí možnosti volit mezi synchronním a asynchronním režimem. V synchronním režimu je třeba zvolit jaký z procesorů je Master a který Slave. Více je komunikace přes USART používána v asynchronním režimu. U konstruovaného manipulátoru jsem zvolil funkci v asynchronním a duplexním přenosu. V asynchronním režimu je možnost volit si délku bitů v rozpětí 5 až 8 bitu. Dále se musí upřesnit počet stop bitu a nastavení bitu parity. Jednotka USART umožňuje rozpoznávání chyb v komunikaci, respektive detekuje ztrátu znaku, chybu parity a rámce. Má tři možnosti přerušení. [11] [3]

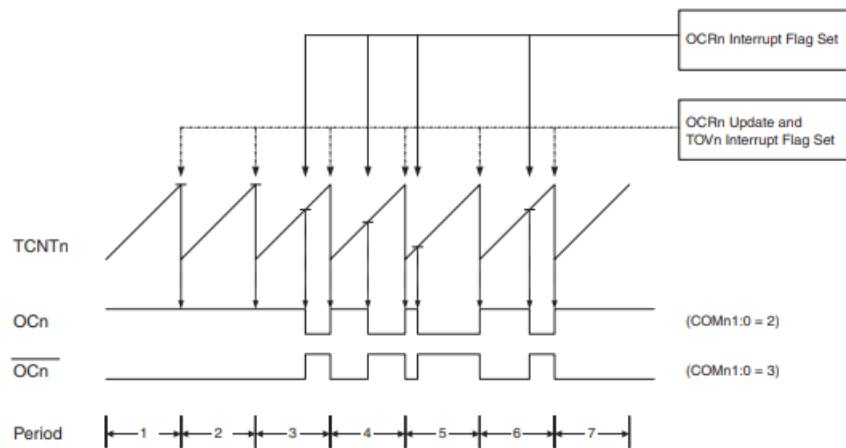
- Následně po odeslání znaku
- Následně po vyprázdnění registru
- Následně po příjmu znaků

Níže je přiložena obrázek s možnostmi výběru rychlostí komunikace při frekvenci mikroprocesoru 8Mhz.

Baud Rate (bps)	$f_{osc} = 8.0000\text{MHz}$			
	U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error
2400	207	0.2%	416	-0.1%
4800	103	0.2%	207	0.2%
9600	51	0.2%	103	0.2%
14.4k	34	-0.8%	68	0.6%
19.2k	25	0.2%	51	0.2%
28.8k	16	2.1%	34	-0.8%
38.4k	12	0.2%	25	0.2%
57.6k	8	-3.5%	16	2.1%
76.8k	6	-7.0%	12	0.2%
115.2k	3	8.5%	8	-3.5%
230.4k	1	8.5%	3	8.5%
250k	1	0.0%	3	0.0%
0.5M	0	0.0%	1	0.0%
1M	-	-	0	0.0%
Max ⁽¹⁾	0.5Mbps		1Mbps	

Obrázek 12 Možné rychlosti pro komunikaci přes USART [11]

Ke generování PWM signálu se využívá funkce čítače/časovače, ty mají čtyři módy, v němž můžou pracovat. Pro ovládání servomotoru se nejvíc hodí použití takzvaného FAST PWM modu. V tomto modu se nastaví časovač na požadovanou periodu. Při zapnutí sepne výstup a počítá. Když se napočítá do hodnoty uložené v registru OCRx výstup vypne a dál počítá do své maximální hodnoty. Úpravou hodnoty OCRx se tedy mění střída signálu a tím se mění natočení servomotoru. [11]



Obrázek 13 Ukázka funkce FAST PWM [11]

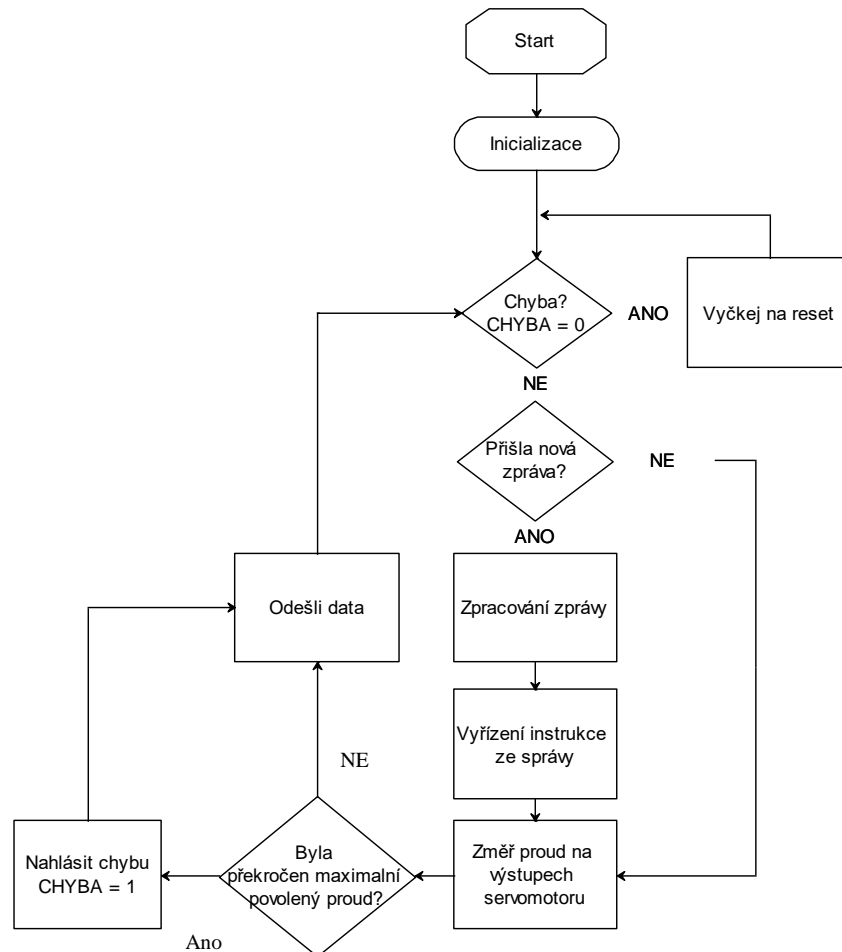
A/D převodníky se používají na měření různých elektrických veličin. Na použitém procesoru jsou s rozlišením deseti bitů. Na napájení těchto převodníků se používá pin AVCC. Na tento pin je připojeno stejné napětí jako na VCC. Je zde možno využít integrované referenční napětí 1,22V a analogovou nulu. Referenční napětí může být také použito z pinu AREF. Možnost samostatného napájení je z důvodu možného rušení tohoto napětí. Převodníky jsou typu „sample and hold“, tento typ znamená že si pamatuje naměřenou hodnotu do doby, než změří další. Tato vlastnost se hodí při vyčítání dat po měření. Převodník se tedy nemusí obsloužit hned. [11] [3]

3.4.1 Program procesoru

V této podkapitole je přiblíženo softwarové řešení procesoru. Je zde uveden přibližný vývojový diagram programu. Schéma vývojového diagramu jsem zkonstruoval tak, aby hrubě postihoval všechny důležité části programu. Nevypovídá tedy o přesném řešení programu. Pouze přibližuje, jak by měl program fungovat.

Při startu se mají inicializovat všechny potřebné prvky. Poté se ověří, zda nebyl některý z motorů přetížen. Při prvním zapnutí tento stav nemůže nastat. Ovšem pokud už program proběhl a byla nahlášena chyba, program se zacyklí a čeká na povel restartu od uživatele z počítačové aplikace. Pokud tento restart přijde, chyba se vynuluje a program pokračuje v hlavní smyčce. Zde se kontroluje, jestli přišla nová zpráva neboli instrukce. Pokud ano, je instrukce následně zpracována a jsou na jejím základě vyvolány požadované změny. Po těchto změnách následuje proces změření proudu motorů. Do tohoto procesu směřuje diagram i pokud nová zpráva nepříjde. Po změření proudu se rozhodne, zda nebyl překročen maximální přípustný proud. Pokud proud překročil maximální přípustnou hranici, je vygenerována chyba, která stopne manipulátor. Při tomto stopnutí se vygeneruje zpráva pro aplikaci, která upozorní

uživatelé, že se tato chyba stala a že je požadován restart manipulátoru. Když nedojde k překročení maximálního proudu, tak se vygeneruje zpráva o aktuálním proudu jednotlivých servomotorů, která se následně odešle. Dále se vracíme na začátek, kde ověříme, zda chyba nevznikla.

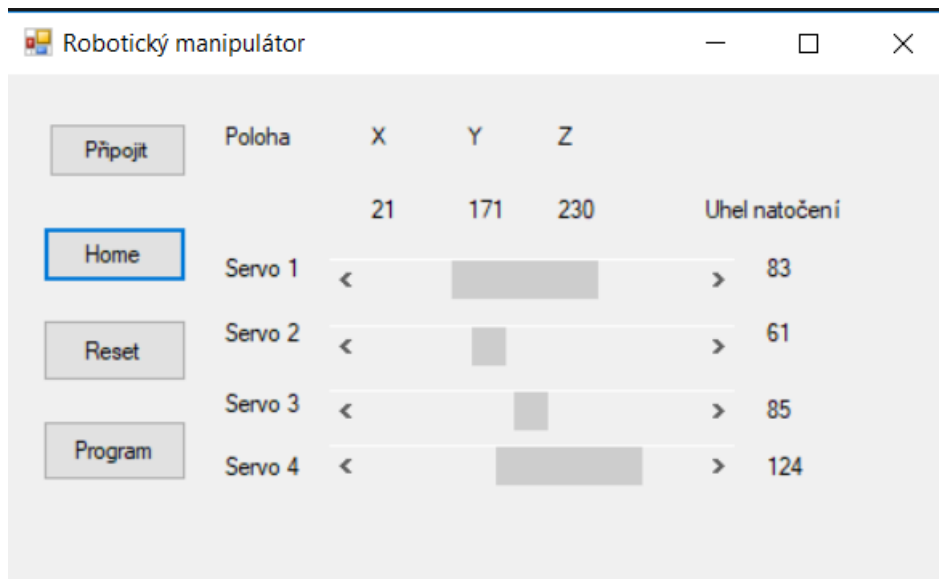


Obrázek 14 Vývojový diagram programu procesoru

Program byl konstruován podle přiloženého diagramu. V přílohách je umístěn celý kód k procesoru.

3.5 Řídící počítačová aplikace

V této kapitole přiblížím návrh a funkce řídicí počítačové aplikace. Tato aplikace je vytvořena v softwaru Microsoft Visual Studio od společnosti Microsoft v jazyce C# ve formě formulářové aplikace. Hlavním úkolem aplikace je obsluhovat komunikaci s řídicí deskou, generovat instrukce a zpracovávat přijaté údaje. Níže tedy uvedu vzhled aplikace a její nastíním její fungování.



Obrázek 15 Ukázka PC aplikace

Na obrázku můžeme vidět grafické zpracování aplikace. Tlačítko „Připojit“ obstarává navázání komunikace s řídicí deskou a zapíná časovač který vyčítá hodnoty odeslané z desky. Tlačítko „Home“ přetáčí manipulátor do stanovené domovské pozice. Tlačítko „Reset“ vyvolává instrukci, pokud přijde zpráva o chybě, tlačítko změní svoji barvu na červenou. Poslední tlačítko „Program“ přečte přiložený textový soubor s instrukcemi. Tyto instrukce přečte a odešle je do manipulátoru. Tento textový soubor lze přepisovat pouze po ručně. V horní části jsou umístěny souřadnice X, Y, Z které ukazují aktuální polohu manipulátoru na základě vypočítání přímé kinematiky. Na pravé straně jsou zobrazeny údaje o natočení jednotlivých servomotorů. Uprostřed aplikace jsou umístěny horizontální rolovací lišty. Podle těchto lišt se nastavuje natočení servomotorů. Pojmenování servomotoru například „Servo 1“ kromě jména ukazuje podle zabarvení hodnotu přetížení, čím víc je barva červenější tím více je servomotor zatěžován.

Komunikace probíhá rychlostí 9600Bds, nenastavuje se parita, je přidán jeden stop bit a počítá se s délkou zprávy 8 bitu. Toto nastavení je stejné jako nastavení USARTu v programu procesoru. Pokud by nebylo nastavení totožné, komunikace by nefungovala.

Zpráva pro odeslání se tvoří při každé změně rolovací lišty. Zpráva se vytvoří tak, že na začátek se dá počáteční písmeno, které nese informaci, o jaký servomotor se jedná. K tomuto znaku je přidána hodnota natočení. Tato hodnota je vždy trojmístná, tedy nastavíme-li například druhý servomotor na natočení 50, pak instrukce vypadá následovně: „B050“. Ta instrukce se odešle hned na zpracování.

Přijímání zpráv se provádí ve smyčce časovače. Můžou přijít dva druhy instrukce, a to buď chybová instrukce nebo pole s naměřenými hodnotami proudu. Když přijde chybová instrukce, nahlásí se chyba a jak zpracování přijmutích zpráv, tak i odesílání se zastaví a je vyžadován restart.

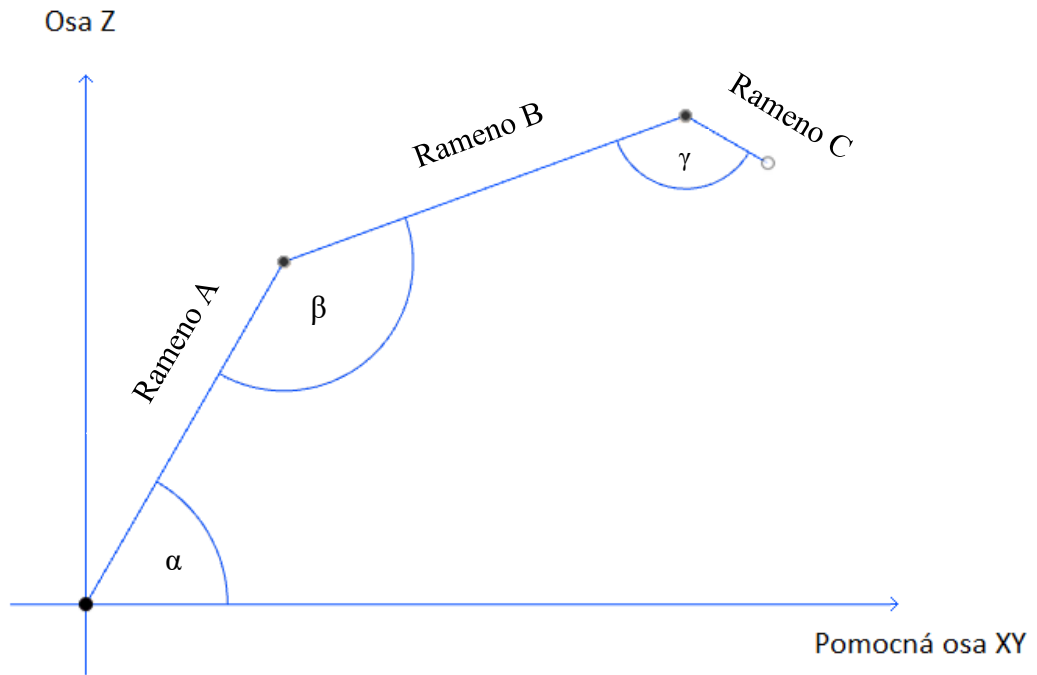
Už skoro veškeré počítače vyrobené v dnešní době neobsahují port pro sériovou komunikaci, je proto potřeba využívat převodníků. Převodník obstarává převod dat z USB na signály sériové linky. U toho manipulátoru používám modul převodníku CP2102 STC. Jedná se v podstatě o převodník TTL. Hlavním čipem je zde CP2102.

- Velikost bufferu RX 576 byte
- Velikost bufferu TX 640 byte
- Komunikační rychlost 0,3 až 1000kbps

3.5.1 Metody pro výpočet polohy XYZ

V této podkapitole přiblížím funkci přímé kinematiky a její výpočty. Vypočítané hodnoty slouží k přibližnému určení polohy. Přímá kinematika se vypočítává za pomoci goniometrických funkcí. Známý jsou úhle natočení ramen a délka ramen. Výpočty budou probíhat tak že se z délky ramene a známého úhle udělá pravoúhlý trojúhelník a spočítají se délky přilehlé a protilehlé odvěsny. Odvěsny se budou sčítat jako přírůstky v ose z a pomocné osy „xy“. Poté, co se vypočítají poslední přírůstky a jsou sečteny, dostáváme celkovou velikost v ose Z. Osy X a Y se spočítají z pomocné osy „xy“, kde je celková délka. Spočte se to opět pomocí goniometrické funkce, kde se vezme úhel natočení prvního servomotoru a se znalostí pomocné osy „xy“ se dopočítají odvěsny kde protilehlá odvěsna je délka v ose X a přilehlá odvěsna je délka v ose Y.

Graficky a početně je tato operace zobrazena níže.



Obrázek 16 Pohled z boku

$$PomocnaXY = \cos \alpha * RamenoA$$

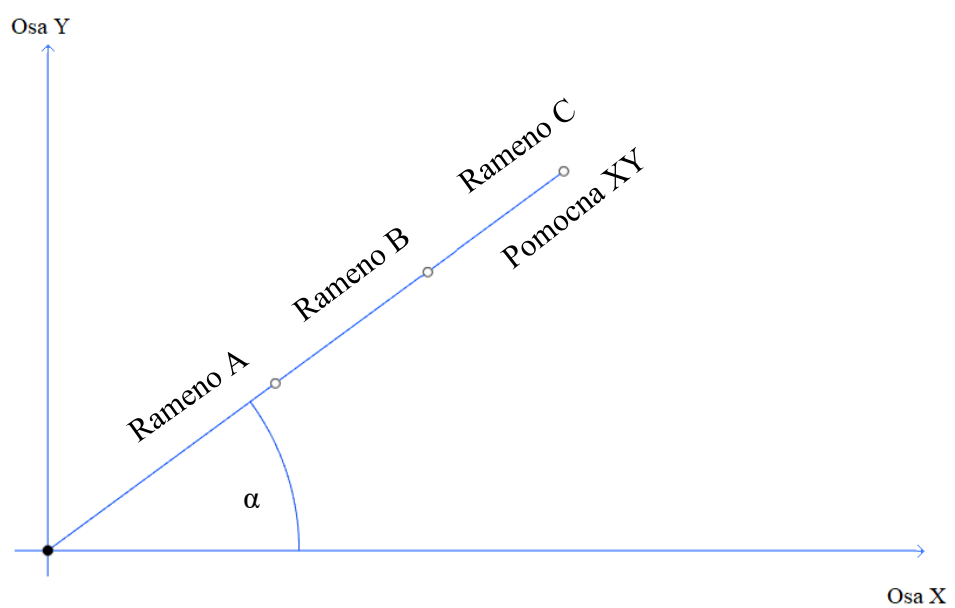
$$Z = \sin \alpha * Rameno A$$

$$PomocnaXY = \cos \beta * RamenoB$$

$$Z = \sin \beta * Rameno B$$

$$PomocnaXY = \cos \gamma * RamenoC$$

$$Z = \sin \gamma * Rameno C$$



Obrázek 17 Pohled ze shora

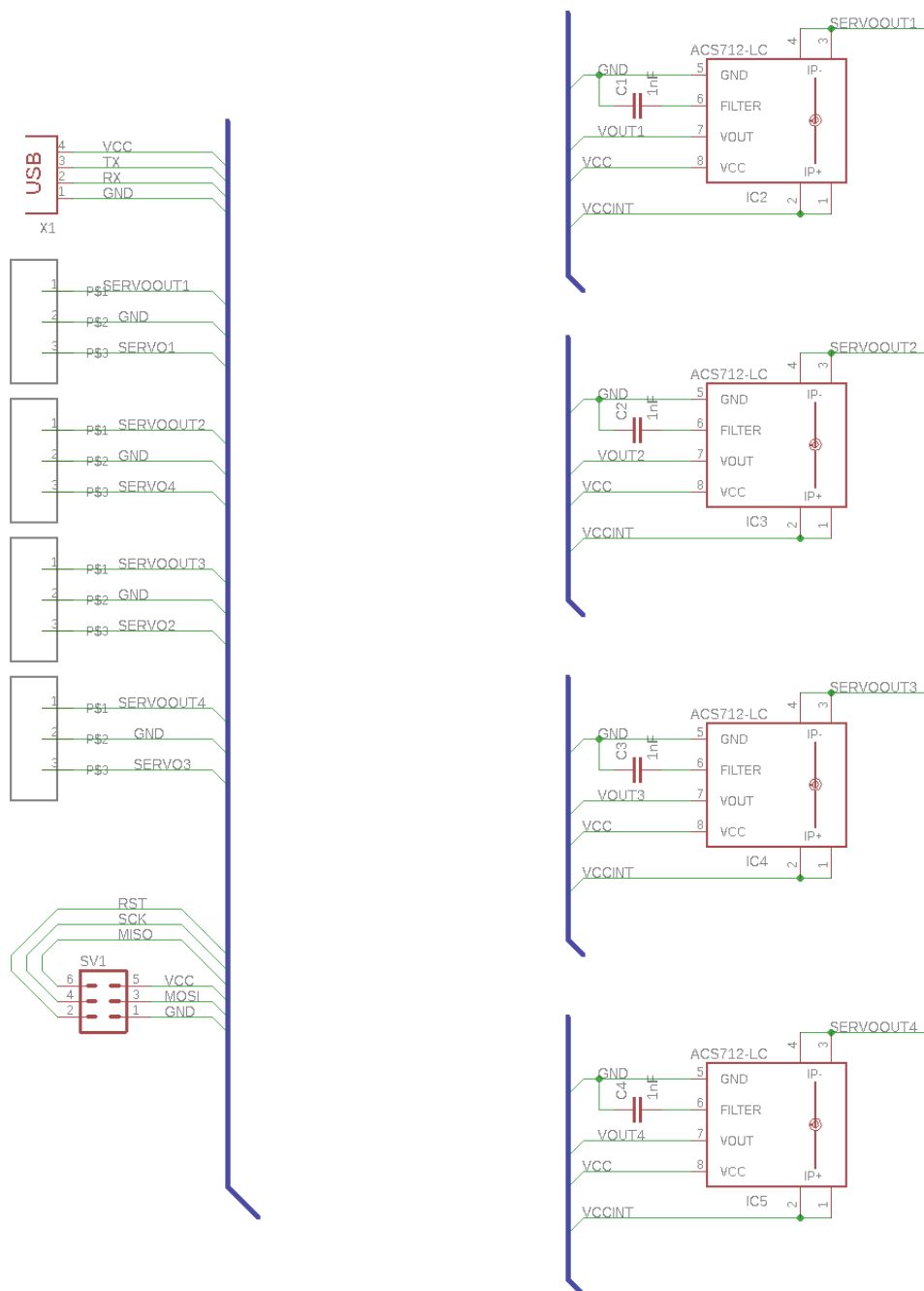
$$X = \cos \alpha * PomocnaXY$$

$$Y = \sin \alpha * PomocnaXY$$

3.6 Návrh DPS řídicí elektroniky

V této kapitole popíšu návrh desky řídicí elektroniky, na které bude umístěn procesor s programem a příslušné periférie. Tato deska je nezbytnou součástí fungování manipulátoru a zde bude probrána její schématická podoba a konečný návrh desky s rozmístěnými součástkami. Tuto desku následně vyrobím, osadím součástkami, připojím k manipulátoru a oživím. Pro schématické zapojení a následný návrh DPS byl použit software Eagle od společnosti Autodesk.

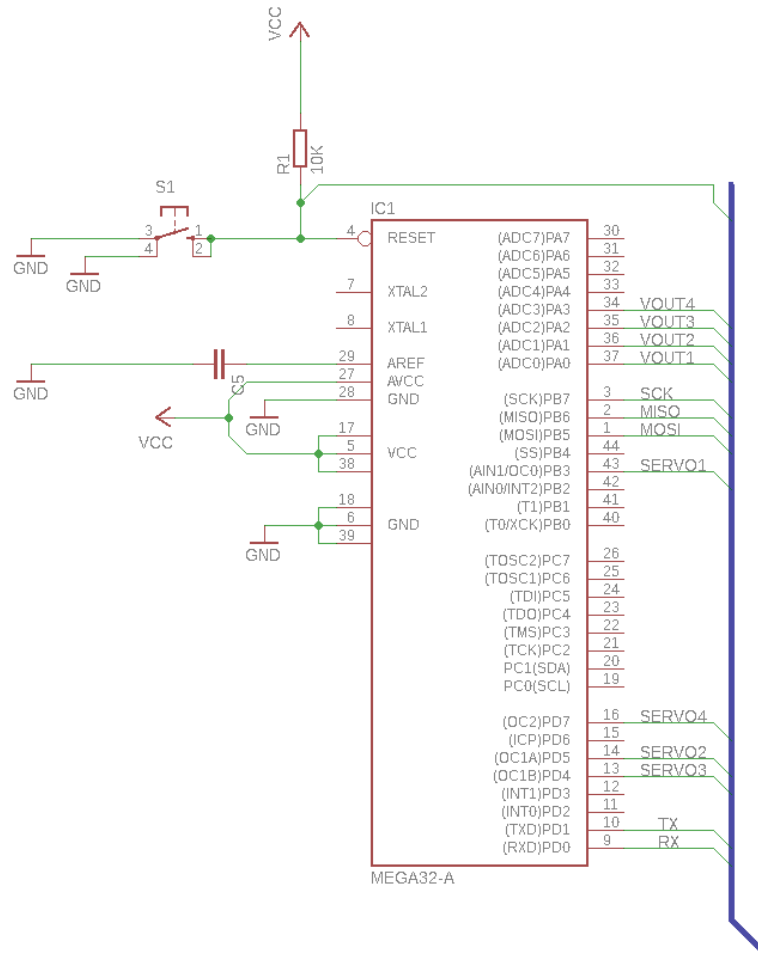
Deska plošného spoje musí pro potřeby manipulátoru obsahovat několik komponent. Těmi jsou připojovací patice pro přívod napětí, tyto patice slouží i pro připojení servomotoru k řídicí desce pro snadné odpojení motoru a následnou výměnu. Další komponenta je připojení rozhraní komunikace USART. Možností existuje několik, já jsem vybral USB typu B. Používá se u nejrůznějších zařízení, třeba u tiskáren. Pro toto použití je ideální. Do tohoto portu se bude připojovat kabel který dále povede do počítače a přes který bude komunikace probíhat. Deska bude dále obsahovat čtyři integrované obvody ACS712, tyto obvody budou zapojeny. Tyto obvody mají za úkol sledovat proud na výstupu servomotoru. Je proto připojen přes napájení k těmto servomotorům. Následný výstup ze součástky VOUT je přiveden na pin A/D převodníku na procesoru. Umístění kondenzátoru 1nF je na doporučení výrobce, vyplývá z přiložené technické dokumentace a zajišťuje lepší funkci součástky. Je zde také vidět připojení pinu pro programátor. Schématický návrh desky s těmito komponenty vypadá následovně.



Obrázek 18 Zapojení ACS712, vstupů a výstupů

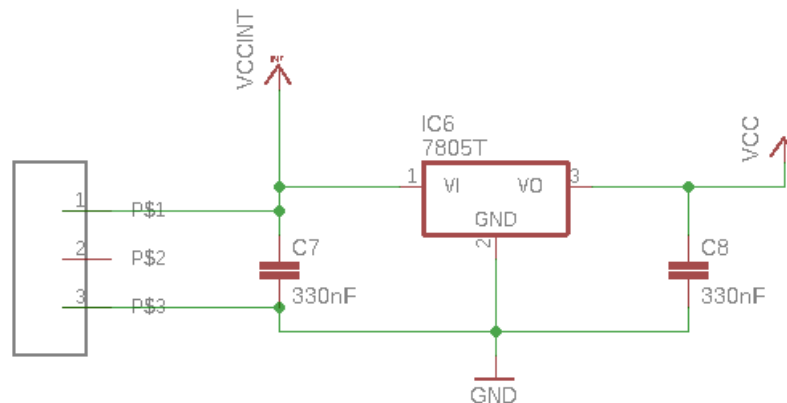
Na obrázku níže je nejdůležitější komponenta řídicí desky procesor. Součástky jako re-setovací tlačítko, rezistor a kondenzátory jsou připojeny podle doporučení výrobce. Piny PA0, PA1, PA2, a PA3 jsou připojeny na obvod ACS712, tyto vstupy mají A/D převodník, pomocí kterého se bude zpracovávat výstupní napětí. Následně jsou připojeny piny pro potřeby programátoru, aby bylo možno program nahrát a následně aby byla možnost program změnit. V dolní části obrátku je vidět připojení pinu PD0 a PD1 neboli piny RX a TX, které jsou připojeny na USB. Tyto piny slouží jako komunikační piny USARTu. Nad nimi jsou výstupy servomotorů,

respektive piny PD4, PD5, PD7 a PB3. Tyto výstupy generují PWM signál na základě vnitřních instrukcí a jsou připojeny na řídicí kontakt servomotoru.



Obrázek 19 Schématické zapojení procesoru

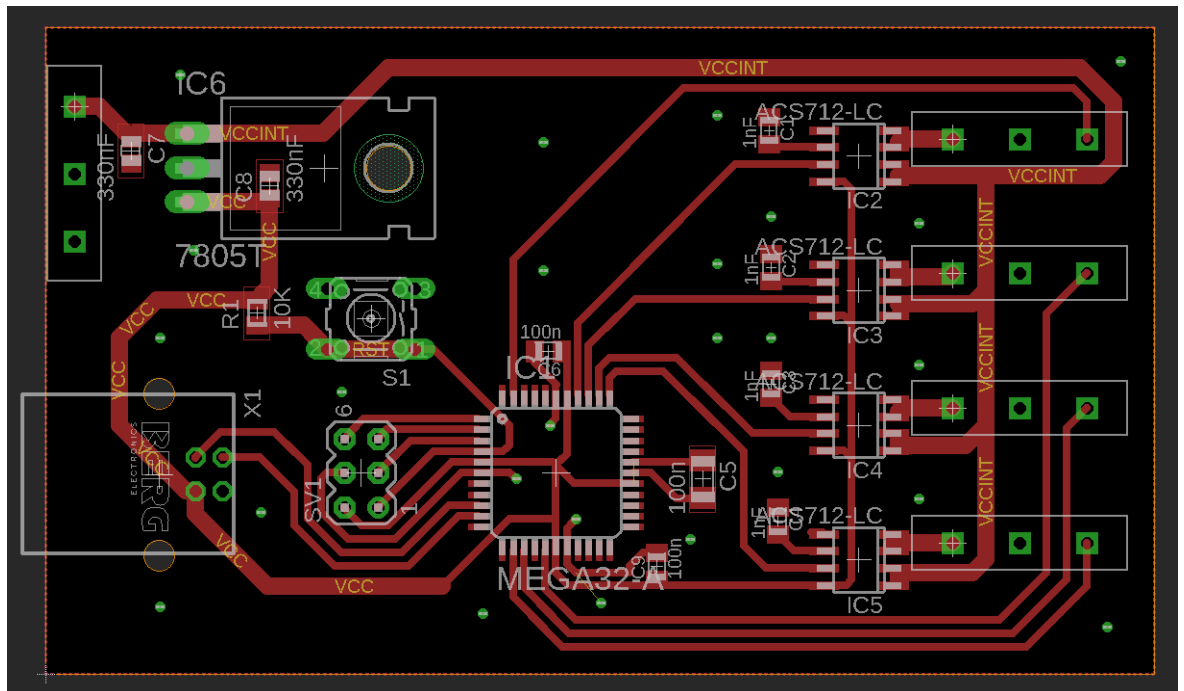
Celá DPS je napájena ze zdroje o napětí přibližně šest voltů. Napájení servomotoru kvůli zvýšení točivého momentu je z toho zdroje. Napájení mikroprocesoru a přidružených obvodů je ze stabilizátoru L7805 připojeném ke zdroji, který vstupní napětí stabilizuje na pět voltů.



Obrázek 20 Schématické zapojení stabilizátoru

Na obrázku je vidět právě zmíněný stabilizátor L7805. Přes patiči je zapojený zdroj pro DPS. Napětí VCCINT je napětí pro servomotory a VCC je stabilizované napětí pro mikroprocesor a podružné obvody.

Podle popsaného schématického zapojení jsem vytvořil desku, podle které se vytvoří výsledná řídicí elektronika. Tento návrh počítá s tím, že deska bude složena ze dvou vrstev. Na první vrstvě jsou rozvedeny všechny logické cesty a napájení. Na druhé vrstvě je řešen rozvod GND.



Obrázek 21 DPS řídicí desky

4 PRAKTICKÉ VYUŽITÍ

V této kapitole popíšu a zhodnotím několik důležitých aspektů zhotoveného manipulátoru, zhodnotím, jestli se mi podařil navrhnout manipulátor, který splňuje kritéria pro kategorii univerzální manipulátory a dále se ho pokusím podle jeho aspektů zařadit. Také popíšu návrhy zlepšení a změn, které se již v průběhu stavby začali objevovat.

Prvním z aspektů je morfologie manipulátoru. Manipulátor má konstrukci obdobnou jako většina průmyslových manipulátorů, má otočnou základnu přidělanou na pracovní desku, dále má systém kloubů s rameny kde na konci se nachází efektor. Ramena a další součásti jsou udělané symetricky a s velkou přesností, tento fakt hodně usnadnil výrobu a montáž.

Velikost a hmotnost manipulátoru se odvíjela ihned od počátku s určením, že pohony budou modelářské servomotory, které se neukázaly být efektivními pohony pro manipulátor. Snižovala se délka ramen a kladl jsem velké nároky na co nejnižší hmotnost. Díky metodě 3d tisku se hmotnost zredukovala na minimum, nejtěžší na manipulátoru jsou servomotory a ložiska.

Manipulátor má celkem čtyři pohyblivé kinematické prvky, tedy manipulátor má čtyři stupně volnosti. Tento počet není hodně a dosti omezuje univerzálnost robota pro náročnější úlohy, na druhou stranu nižší počet kinematických prvků pořád stačí na základní úlohy. Navíc čím víc je stupňů volnosti, tím se zvedá náročnost na výrobu, kde se řeší vzrůstající hmotnost jednotlivých ramen a větší zatížení motorů.

Velikost obsluhovaného prostoru se zdá být dostatečná. Natažená ramena s vodorovně natočenými motory dávají dohromady 290 mm. Zakládán se dokáže otočit o 180° . Velikost obsluhovaného prostoru je pak dána vypočítáním poloviny obsahu kruhu o poloměru 290 mm. Prostor má plochu zaokrouhlených $2\,642\text{ cm}^2$. Výpočet prostoru postrádá smysl kvůli omezení stupňů volnosti. Robot tento prostor nedokáže smysluplně využít krom některých jednoduchých úloh.

Přesnost manipulátoru není dostatečná. Je to dáno více faktory. První z nich je, že modelářské servomotory nemají velkou citlivost. Druhý je že i když má procesor všechno potřebné pro ovládání toho manipulátoru, nedokáže vygenerovat PWM signál s dosti velkým rozlišením střídny na všech svých výstupech.

Způsob řízení byl zadán již na začátku. Manipulátor má tedy na míru vytvořenou uživatelskou počítačovou aplikaci na přímé ovládání servomotorů. Komunikace této aplikace a řídicí desky je přes převodník TTL na sériovou linku. Tato aplikace je univerzální a dá se k ní doplnit mnoho funkcí.

Zkonstruovaný manipulátor se tedy může členit do univerzálních manipulátorů, přesněji do kategorie manipulátory s proměnlivým programem. Do této kategorie ho dostala možnost přiložení textový dokument s napsaným programem do kořenové složky aplikace. Tento dokument je aplikací přečten a obsažené instrukce jsou odeslány manipulátoru.



Obrázek 22 Vytvořený manipulátor

Na obrázku je zachycena poslední podoba vytvořeného manipulátoru s přidávanými protizávažími. Protizávaží bylo přidáno z důvodu přetěžování servomotoru uloženého v prvním kloubu. Toto přetěžování správně vyvolávalo chybové hlásky.

4.1 Možnosti vylepšení

Možnosti na vylepšení se v průběhu konstrukce, montáže a programování hodně množily. Zde popíšu jen možnosti, které mě více zaujaly a o kterých jsem v průběhu přemýšlel že by bylo možné je hned zapojit. Z důvodu větší časové náročnosti na programování jsem tyto možnosti opět vyloučil.

Modelářské servomotory se při praktických zkouškách manipulátoru ukázaly jako nevhodné. A to z hlediska jednak přesnosti tak krouticího momentu. Nejvíce krizovým místem se ukázal být první kloub, kde servomotor musí udržet na místě všechna ramena manipulátoru i ve vodorovné poloze. První možnost vylepšení nabízí vyměnit modelářské servomotory za kvalitnější, silnější než-li za průmyslové servomotory.

Jako další vylepšení by bylo vhodné zvolit vícebitový procesor. Osmibitové procesory sice vystačují, ale tím také končí. Velké omezení vidím v malém počtu čítačů/časovačů, které omezují přidávání dalších periférií či zvyšování pohyblivých kinematických prvků.

Závěr

Cílem mé práce bylo navrhnout manipulátor, jehož pohyb bude zajištěn modelářskými servomotory a zároveň bude ovládán PC aplikací. Součástí práce bylo dále navrhnout možnosti řešení proti přetížení jednotlivých pohonů a otestovat možnosti sestrojeného zařízení.

Nejdříve jsem vymodeloval návrh mechanismu manipulátoru a rozkreslil jednotlivé součástky v 3D softwaru pro modelaci. Rozkreslené návrhy jsem vložil do softwaru obsluhující 3D tiskárnu a vytiskl je. Následovalo sestrojení manipulátoru a jeho umístění na pracovní desku.

Po sestrojení manipulátoru jsem se věnoval řídicí desce, navrhnul jsem schéma zapojení a současně i vhodné rozložení součástek na DPS. Pro řídicí desku jsem zvolil procesor ATmega32A. Dílčím cílem mé práce bylo navrhnout možnosti řešení přetížení jednotlivých pohonů, tento problém jsem vyřešil osazením řídicí desky integrovaným obvodem ACS712, který sleduje velikost proudu servomotoru a procesor tento proud vyhodnocuje.

Nyní bylo na čase napsat vhodný program pro komunikaci s řídicí deskou. Pro vytvoření programu jsem použil programovací jazyk C#. Kód aplikace jsem napsal v programovacím softwaru Visual Studio od společnosti Microsoft.

Po splnění všech tří dílčích úkolů, tedy sestrojení mechanismu manipulátoru, oživení řídicí desky a sepsání počítačové aplikace, jsem svůj návrh vyzkoušel. Mechanismus robotické ruky správně reagoval na příkazy z počítačové aplikace. Během testů jsme si ověřili fungování navržené ochrany proti přetížení, současně jsem zjistil, že použité servomotory nezvládají vysokou zátěž. Problém se servomotory jsem vyřešil přidáním protiváhy prvního ramene. Přesto největší úskalí přístroje spatřuji v nepřesnosti přístroje, který je způsoben nedostatkem čítačů/časovačů procesoru. Proto bych navrhl v případě příštího sestrojení robotické ruky vyzkoušet jiný typ procesoru.

POUŽITÁ LITERATURA

- [1] KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016. ISBN 978-80-214-4828-5.
- [2] MAŘÍK, Vladimír. *Průmysl 4.0: výzva pro Českou republiku*. 2016. Praha: Management Press, 2016. ISBN 978-80-7261-440-0.
- [3] VÁŇA, Vladimír. *Mikrokontroléry ATMEL AVR: Popis procesoru a instrukční soubor*. Praha: BEN - technická literatura, 2003. ISBN 80-730-0083-0.
- [4] *Robotics, vision and control*. 2017. New York, NY: Springer Berlin Heidelberg, 2017. ISBN 978-3-319-54412-0.
- [5] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. 2005. Praha: BEN - technická literatura, 2005. ISBN 80-730-0141-1.
- [6] CHVÁLA, Břetislav, Robert MATIČKA a Jaroslav TALÁCKO. *Průmyslové roboty a manipulátory*. 1990. Praha: SNTL, 1990. ISBN 80-030-0361-X.
- [7] BOŽEK, Pavol. *Špecializované robotické systémy*. 2011. Ostrava: Ámos, 2011. ISBN 978-80-904766-3-9.
- [8] ACS712 [online]. [cit. 2019-05-07]. Dostupné z: <https://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf>
- [9] Cytron USBASP Programmer [online]. 2011 [cit. 2019-05-07]. Dostupné z: <http://www.hmangas.com/Electronica/Datasheets/USBasp/AVR-USBasp%20User%20Manual.pdf>
- [10] CP2102/9 [online]. [cit. 2019-05-07]. Dostupné z: <https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf>
- [11] ATmega32A [online]. [cit. 2019-05-07]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega32A-DataSheet-Complete-DS40002072A.pdf>
- [12] KUKA [online]. [cit. 2019-05-07]. Dostupné z: <https://www.kuka.com/>
- [13] ABB [online]. [cit. 2019-05-07]. Dostupné z: <https://new.abb.com>