

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra měření

## Robotická ruka pro mobilního robota

**Jan Škvrna**

Vedoucí: Ing. Vojtěch Petrucha, Ph.D.  
Obor: Kybernetika a Robotika  
Květen 2021



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Škvrna** Jméno: **Jan** Osobní číslo: **483569**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra měření**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Robotická ruka pro mobilního robota**

Název bakalářské práce anglicky:

**Robotic arm for a mobile robot**

Pokyny pro vypracování:

- Vzhledem k únosnosti a dalším parametrům mobilního robota zvolte vhodnou mechanickou konstrukci robotické ruky.
- Navrhněte řídicí jednotku připojitelnou na sběrnici CAN, včetně senzorové výbavy, která bude umožňovat (zpětnovazební) řízení robotické ruky.
- Implementujte firmware pro základní uživatelské ovládání robotické ruky.
- Konstrukci otestujte a pokuste se ji začlenit do systému mobilního robota.

Seznam doporučené literatury:

- [1] Reza N. Jazar: "Kinematics, Dynamics, and Control", 2nd Edition, Springer 2010 ISBN 978-1-4419-1750-8
- [2] Franklin, Powell, Emami-Naeini: "Feedback Control of Dynamics Systems". Prentice Hall, USA
- [3] Ahmet Bindal: "Electronics for Embedded Systems", Springer International Publishing, Switzerland 2017

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Vojtěch Petrucha, Ph.D., 13138**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **13.01.2021**

Termín odevzdání bakalářské práce: \_\_\_\_\_

Platnost zadání bakalářské práce:

**do konce letního semestru 2021/2022**

Ing. Vojtěch Petrucha, Ph.D.  
podpis vedoucí(ho) práce

Podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

**3.5.2021**

Datum převzetí zadání

Podpis studenta



## Poděkování

Děkuji vedoucímu práce, panu Ing. Vojtěchu Petruchovi, Ph.D., za všechny rady a skvělé vedení této práce. Dále bych chtěl poděkovat mé rodině za podporu během studia.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 21. května 2021

## Abstrakt

Obsahem této práce je sestavení a zprovoznění robotické ruky určené pro mobilní outdoorové vozítko. V práci jsou porovnány mechanické konstrukce od různých výrobců a jsou zde popsány úpravy, které zvýšily robustnost konstrukce a umožnily integraci ruky do vozítka.

V rámci této práce jsou navrženy dvě desky plošných spojů pro řízení ruky a jsou zde popsány jednotlivé komponenty. Pro jednu desku je navržen firmware splňující všechny zadané úkoly v oblastech řízení a komunikace ruky. Závěrem bylo provedeno testování celého systému.

**Klíčová slova:** robotická ruka, outdoorové vozítko, rover

**Vedoucí:** Ing. Vojtěch Petrucha, Ph.D.

## Abstract

The content of this work is to build a robotic arm for a mobile outdoor vehicle. The work contains a comparison of different structures from multiple manufacturers, and it describes modifications made to increase robustness of the structure and integrate the arm to the vehicle.

Within this work, two different printed circuit boards are designed for driving the arm, and all components on the boards are described. There is a custom firmware for one board that fulfills all the tasks in communication and driving the arm. In the end, the whole system was tested.

**Keywords:** robotic arm, outdoor vehicle, rover

**Title translation:** Robotic arm for a mobile robot

## Obsah

<b>1 Úvod</b>	<b>1</b>	4.1.5 Senzor úhlu natočení . . . . .	15
<b>2 Koncepce řešení</b>	<b>3</b>	4.1.6 Hallův senzor . . . . .	16
<b>3 Mechanická část ruky</b>	<b>5</b>	4.1.7 Mikrokontroler . . . . .	17
3.1 Dostupné řešení . . . . .	5	4.1.8 Vstupní ochrana . . . . .	18
3.1.1 Mirobot . . . . .	5	4.1.9 Regulátory . . . . .	19
3.1.2 KAUDA . . . . .	6	4.2 Desky plošných spojů . . . . .	20
3.1.3 uStepper robot arm 4 . . . . .	7	<b>5 Pokročilé algoritmy pro řízení</b>	<b>23</b>
3.1.4 Srovnání dostupných řešení . . . . .	7	<b>6 Řídicí Firmware</b>	<b>27</b>
3.2 Výběr nejvhodnějšího řešení . . . . .	8	6.1 Zpracování vstupů . . . . .	28
3.3 Úprava ruky . . . . .	8	6.1.1 Hallovy senzory . . . . .	28
<b>4 Řídicí elektronika</b>	<b>11</b>	6.1.2 Senzory úhlu natočení . . . . .	28
4.1 Výběr komponent . . . . .	12	6.1.3 Komunikace přes USB . . . . .	29
4.1.1 Driver krokových motorů . . . . .	12	6.2 Komunikace přes CAN . . . . .	29
4.1.2 Ovládání servomotorů . . . . .	13	6.3 Zpracování zpráv . . . . .	30
4.1.3 USB . . . . .	14	6.4 Zpracování výstupů . . . . .	31
4.1.4 CAN . . . . .	15	6.4.1 Servomotory . . . . .	31
		6.4.2 Krokové motory . . . . .	32

6.4.3 Komunikace se senzory úhlu .	34
6.4.4 Ochrany ruky .....	35
6.5 Kalibrace .....	36
6.6 Možnosti řízení ruky .....	36
6.6.1 Windows aplikace .....	36
6.6.2 CAN .....	37
<b>7 Testování</b>	<b>39</b>
<b>8 Závěr</b>	<b>41</b>
<b>A Literatura</b>	<b>43</b>
<b>B Desky plošných spojů</b>	<b>45</b>



## Obrázky

1.1 Robot Spot s rukou [4] . . . . .	1	4.6 Připojení konektoru CAN . . . . .	15
2.1 Popis ovládacích os . . . . .	4	4.7 Připojení modulu CAN . . . . .	15
3.1 Mirobot od výrobce WLKATA . . . . .	6	4.8 Magnet na hřídeli . . . . .	16
3.2 KAUDA od Giovanni Lerda . . . . .	6	4.9 Zapojení senzoru AS5601 . . . . .	16
3.3 uStepper robot arm 4 . . . . .	7	4.10 Zapojení Hallových senzorů a průběh výstupu senzoru [5] . . . . .	16
3.4 Originální základna . . . . .	9	4.11 Zapojení mikrokontroleru část 1. . . . .	17
3.5 Upravená základna . . . . .	9	4.12 Zapojení mikrokontroleru část 2. . . . .	18
3.6 Ruka na roveru . . . . .	10	4.13 Zapojení mikrokontroleru část 3. . . . .	18
3.7 Implementace Hallových senzorů a magnetů do konstrukce . . . . .	10	4.14 Vstupní ochrana . . . . .	19
4.1 Blokové schéma řídicí elektroniky . . . . .	11	4.15 Regulátor napětí 7V . . . . .	19
4.2 Porovnání řízení s a bez microstepping [12] . . . . .	12	4.16 Regulátor napětí 5V . . . . .	20
4.3 Schéma zapojení L6470 . . . . .	13	4.17 Regulátor napětí 3,3V . . . . .	20
4.4 Schéma zapojení ovládání servomotorů . . . . .	14	4.18 Signálové vrstvy . . . . .	21
4.5 Připojení konektoru USB . . . . .	14	4.19 Druhá zemnicí vrstva a třetí napájecí vrstva . . . . .	21
		4.20 Návrh slave desky . . . . .	22
		4.21 Osazená master a slave deska . . . . .	22
		5.1 Velikosti ramen a body . . . . .	23

5.2 Výpočet bodu B .....	24	7.1 Průběh proudu při amplitudě výstupního napětí 16 %.....	40
5.3 Výpočet bodu B .....	24	7.2 Průběh proudu při amplitudě výstupního napětí 32 a 24 % .....	40
5.4 Ukázka kódu .....	25		
6.1 ukázka nastavení pinů mikrokontroleru a nastavení časovače s PWM výstupem v STM32CubeMX [15] .....	27	B.1 První, signálová vrstva .....	46
6.2 Kvadrurní výstup senzoru úhlu při rozběhu motoru .....	29	B.2 Druhá, zemnicí vrstva.....	47
6.3 Přenos bytu přes UART .....	29	B.3 Třetí, napájecí vrstva .....	48
6.4 Přenos zprávy typu data přes USB .....	31	B.4 Čtvrtá, signálová vrstva .....	49
6.5 Příkaz set parameter .....	32		
6.6 Příkaz run .....	33		
6.7 status registr .....	33		
6.8 Příkaz run, hodinový signál je modrý, data červené a chipselect žlutý .....	34		
6.9 Detail přenosu, hodinový signál je modrý, data červené a chipselect žlutý .....	34		
6.10 ABN registr .....	35		
6.11 Aplikace .....	36		

## Tabulky

3.1 Srovnání konstrukcí .....	7
6.1 Přijímané zprávy .....	30
6.2 Příkaz data write .....	35
6.3 Ovládání aplikace .....	37



# Kapitola 1

## Úvod

Klon Sawppy rovera je šestikolové outdoorové dálkově ovládané vozítko sestavené na Katedře měření Fakulty elektrotechnické ČVUT v Praze. Outdoorové rovery mají zpravidla více účelů než pouhý pohyb. Z tohoto důvodu je cílem této práce doplnit Sawppy rover o robotickou ruku, která bude dálkově řízená. S rukou bude rover schopen manipulovat s předměty. Ruka musí být robustní, ale zároveň lehká, napájená z baterie a schopná reagovat na příkazy od rovera pomocí sběrnice. Ruka musí nabízet alespoň 5 stupňů volnosti, a to kvůli nutnosti univerzálního použití. Pro intuitivnější ovládání bude implementována inverzní kinematická transformace, která zaručí ovládání pomocí os X, Y a rotace kolem základny. Na robotické ruce nebude chybět kalibrace a různé typy ochrany. Ruku bude možné ovládat i bez rovera, a to za pomoci připojení USB a použití navržené aplikace pro operační systém Windows. Rover s rukou budou sloužit na prezentační akce Fakulty.



Obrázek 1.1: Robot Spot s rukou [4]



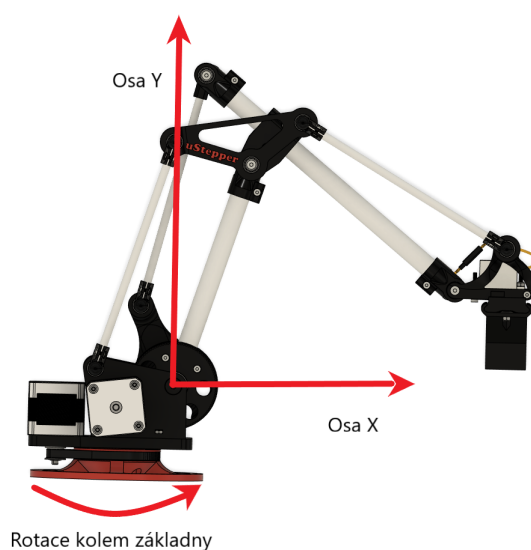
## Kapitola 2

### Koncepce řešení

Vzhledem k tomu, že je tato práce zpracovávána na Fakultě elektrotechnické, není nutné navrhovat novou mechanickou konstrukci. Na trhu je dostatek již hotových konstrukcí. Některé jsou open source, tudíž jediný výdaj jsou náklady na pořízení a vytištění součástek. Ve třetí kapitole jsou popsáni kandidáti, kteří byli vybráni na základě ceny, dostupnosti, omezené hmotnosti a počtu stupňů volnosti. Hmotnost je omezená na přibližně 1.5 kg, protože větší váha by udělala rover nestabilní. Konstrukce musí být robustní, protože se bude pohybovat na roveru a také s ní bude na prezentačních dnech zacházeno neodborně. Proto musí mít ruka dostatek ochran, aby se nepoškodila.

Ve čtvrté kapitole je popsán návrh řídicí elektroniky ruky. Spoustu dostupných mechanických řešení je možné zakoupit s již hotovou řídicí elektronikou. Bohužel ji nelze upravovat a možnost programování je omezená. Z tohoto důvodu je v této práci proveden návrh nové řídicí elektroniky. Nový návrh poskytl dostatek svobody pro implementaci všech žádaných funkcionalit. Řídicí elektronika umožňuje řízení všech motorů, přidání ochran a také možnost připojení přes různé sběrnice, tak aby ruka mohla komunikovat s roverem. V této kapitole jsou podrobně popsány všechny zvolené komponenty.

Pátá kapitola je věnována inverzní kinematické transformaci. Řídit robotickou ruku nastavováním úhlů motorů by bylo velmi nepraktické a obsluhovat ruku by dokázal jen zkušený člověk. Z tohoto důvodu by bylo vhodné řídit ruku pomocí souřadnic X a Y chapadla a rotací kolem základny, které jsou zobrazeny na obrázku 2.1. Inverzní kinematická transformace vypočítá žádané úhly na základě zvolené polohy chapadla. V kapitole je popsán postup výpočtu úhlů a převod na polohu sensorů úhlu.



**Obrázek 2.1:** Popis ovládacích os

V šesté kapitole je rozebrán řídicí firmware, který je nahraný v mikrokontroleru. Ve firmwaru je nutné zpracovat všechny vstupy jako jsou Hallovy senzory a senzory úhlu natočení. Firmware musí umožňovat komunikaci skrz UART s USB modulem a pomocí CAN sběrnice s roverem, dále komunikovat s drivery krokových motorů pomocí sběrnice SPI. Mikrokontroler musí být schopen vypočítat inverzní kinematickou transformaci a ovládat krokové motory a modelářské servomotory. Firmware v sobě bude mít implementovanou kalibraci, protože nejsou použity absolutní senzory úhlu.

V sedmé kapitole je popsáno testování a měření spotřeby ruky, která je důležitým parametrem, protože je rover napájen z baterie.



## Kapitola 3

### Mechanická část ruky

#### 3.1 Dostupné řešení

Při výběru kandidátů byl kladen důraz na robustnost, váhu, počet stupňů volnosti a také dostupnost a cenu. Níže jsou uvedeni tři vhodní kandidáti.

##### 3.1.1 Mirobot

Mirobot je první a nejkvalitnější mechanické řešení robotické ruky. Robotická ruka se jmenuje Mirobot[18] a vyrábí ji čínská společnost WLKATA. Ruka byla představena roku 2020 v crowdfundingové kampani na stránce [www.kickstarter.com](http://www.kickstarter.com), kde žádala uživatele, aby přispěli na vývoj ruky tím, že si ji zakoupí. Bylo vybráno přes 330 000 USD, což je desetinásobek žádané částky. Ruka má 6 stupňů volnosti, používá pouze jeden servomotor, a to na chapadle. Ve všech ostatních kloubech používá krokové motory, což jí přidává na robustnosti. Výrobce uvádí přesnost pohybu chapadla 0.2 mm. Ruka je převážně vyrobená z plastu, váží 1.5 kg a unese maximálně 400 gramů. Výrobce ji nyní prodává za 1 500 USD se vším příslušenstvím, jako jsou motory, řídicí deska a firmware, který je opensource a lze si ho upravovat. Výrobce nabízí spoustu dokupitelných chapadel.



**Obrázek 3.1:** Mirobot od výrobce WLKATA

### ■ 3.1.2 KAUDA

Dalším kandidátem je robotická ruka KAUDA[8] navržená Giovannim Lerdem. Jedná se o opensource projekt. Ruka se dá sestavit z běžně dostupných materiálů a ze součástek vytištěných na 3D tiskárně. Autor dává všechny potřebné soubory k vytištění na svoji webovou stránku a jsou volně přístupné. Je zde uveden i podrobný návod, jak ruku sestavit a také rozpis všech potřebných součástek. O ruce není příliš informací, ale má 5 stupňů volnosti, ovládají ji tři krokové motory a dva servomotory. Váží přibližně 3kg. Ruku není možné koupit. Autor nabízí na svých stránkách několik různých chapadel.



**Obrázek 3.2:** KAUDA od Giovanni Lerda

### 3.1.3 uStepper robot arm 4

Posledním vhodným kandidátem je uStepper robot arm 4[16] od firmy uStepper. Ruka se dá přímo od výrobce koupit s motory a řídicí elektronikou za 455 EUR. K ruce je dostupný podrobný návod a také návrhy součástek, které je možné vytisknout na 3D tiskárně a sestavit si ruku svépomocí. Ruka má 4 stupně volnosti a používá tři krokové a jeden servomotor. Jedná se o jednoduchou konstrukci, ve které byla dosažena malá hmotnost díky použití hliníkových tyčí. Váží přibližně 1.5 kg.



Obrázek 3.3: uStepper robot arm 4

### 3.1.4 Srovnání dostupných řešení

Tabulka 3.1: Srovnání konstrukcí

	Mirobot	Kaada	uStepper
počet stupňů volnosti	6	5	4
váha [kg]	1.5	3	1.5
motory	5x krok. + 1x servo	3x krok. + 2x servo	3x krok. + 1 servo
Dostupnost a cena	1540 USD	Opensource	Opensource/455 EUR

## 3.2 Výběr nejvhodnějšího řešení

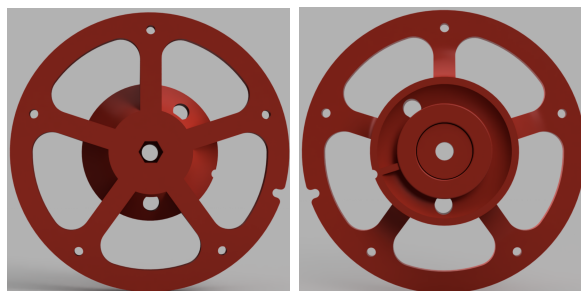
Při výběru nejlepšího kandidáta rozhodovalo několik faktorů. První a nejdůležitější byla hmotnost. Bylo otestováno, jak se rover chová při položení přibližně 1.5 kg a 3 kg závaží. Při 3 kg se začal naklánět, tudíž je nutné zvolit ruku vážící ideálně 1.5 kg. Z tohoto důvodu byla vyřazena robotická ruka KAUDA. Používá převážně tištěné díly, které jsou bohužel příliš těžké.

Dalším faktorem byl počet stupňů volnosti a velikost pracovního prostoru. V tomto vyniká Mirobot. Nevýhodou u této ruky ale je, že není možné si ji vytisknout a sestavit. Kupovat celý set za 1500 USD, bylo nejenom zbytečné, ale také příliš drahé. Byla potřeba pouze konstrukce a motory. Elektroniku bylo nutné doplnit o CAN sběrnici a o přídatnou ochranu v podobě Halloových senzorů umístěných na mezních úhlech kloubů. Nebylo by také možné navrhnout vlastní firmware. Výrobce byl kontaktován, ale bohužel nebylo možné od něj zakoupit pouze mechanickou část.

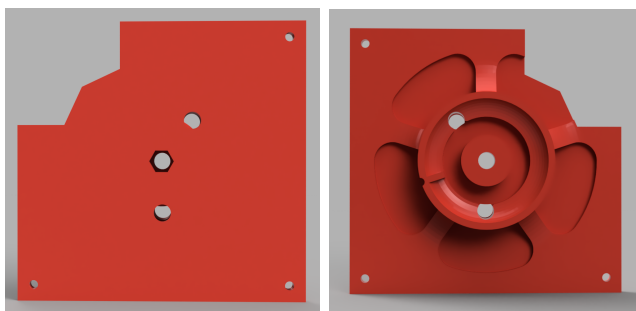
Jediným vhodným kandidátem je uStepper. Byly objednány všechny potřebné součástky a zároveň se nechaly vytisknout všechny potřebné díly na 3D tiskárně. Poté byly nařezány hliníkové tyče a robotická ruka byla sestavena. 3D tisk nebyl přesný a bylo potřeba vybrousit všechny díry na ložiska a matice.

## 3.3 Úprava ruky

Vybraná konstrukce není ideální, a proto bylo nutné udělat 3 úpravy. První úprava se týká základny. Základna je v originálním návrhu kulatá, a jelikož je potřeba připevnit ruku k roveru, je vhodnější, aby byla základna čtvercová. Ruka bude přichycena na roh roveru. Rover je sestaven z hliníkových profilů a v rozích jsou 3D tištěné spojky. Spojky jsou o něco vyšší než hliníkové profily, a proto bylo nutné navrhnout výřez do podstavy. Díky této úpravě sedí ruka na roveru perfektně. Velikost podstavy je 10x10 cm, vysoká je 5 mm. V místech, kde se v originálním návrhu nenachází žádná výplň, je přidáno 2 mm výplně pro vyšší tuhost.



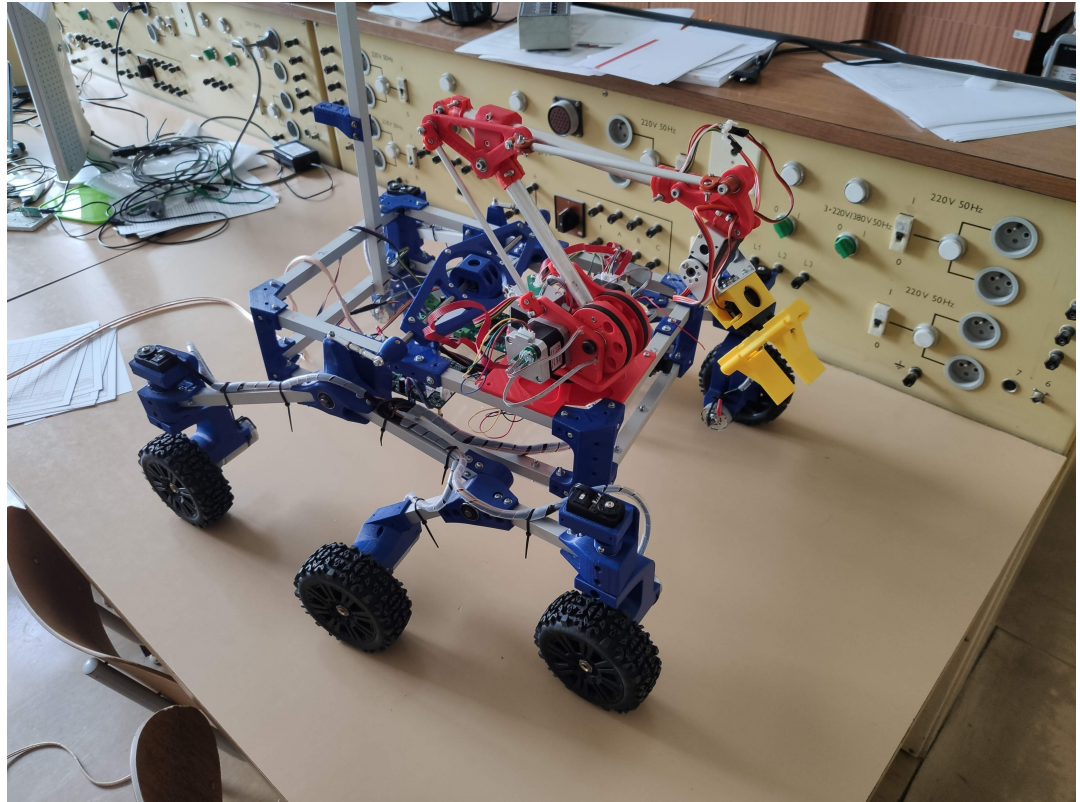
**Obrázek 3.4:** Originální základna



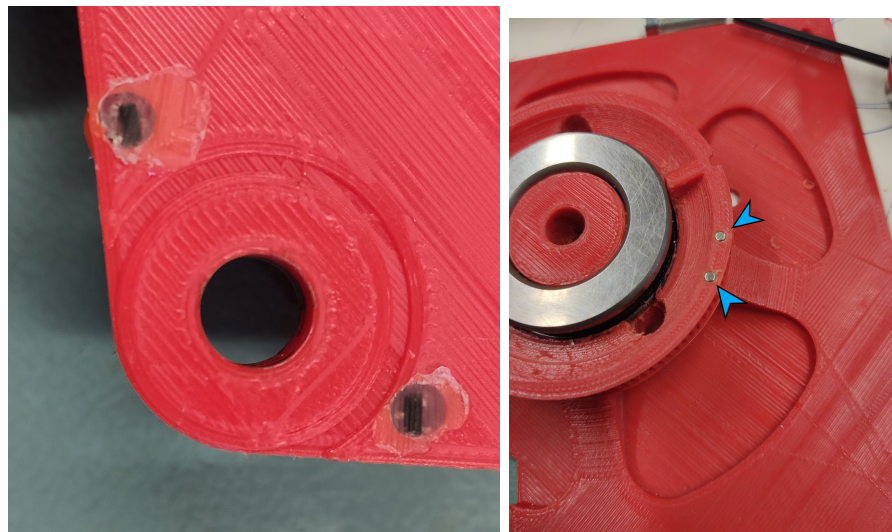
**Obrázek 3.5:** Upravená základna

Protože má ruka pouze 4 stupně volnosti a chapadlo je neustále drženo ve vodorovné pozici, což není praktické, bylo vhodné přidat další stupeň volnosti, který umožní hýbat s chapadlem tak, aby mohlo objekt zvedat z různých úhlů a nikoliv pouze ze shora. Na místo, kde je v originálním návrhu přímo uchyceno chapadlo, byl přidělán servomotor a na tento servomotor byly navrženy 2 díly na uchycení chapadla. Úpravy byly prováděny v programu Fusion 360 od firmy Autodesk a ruka s nimi váží o 400 gramů více.

Dále byla na každý krokový motor přidána malá deska plošných spojů, která obsahuje senzor úhlu natočení a konektory na Hallovy senzory. Tato koncepce vychází z řídicí elektroniky dodávané k ruce. Součástky na desce a deska taková jsou ale jiné a v originálním návrhu se nevyskytují Hallovy senzory. Ty budou přidělané na konstrukci takovým způsobem, aby zastavily motor v případě, že by byl úhel natočení kloubu mimo pracovní prostor a další pohyb by mohl poškodit konstrukci.



Obrázek 3.6: Ruka na roveru

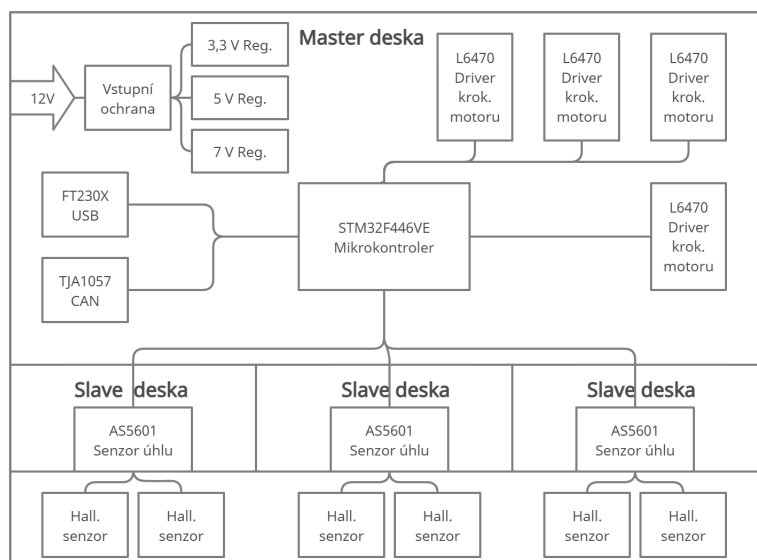


Obrázek 3.7: Implementace Hallových senzorů a magnetů do konstrukce

## Kapitola 4

### Řídicí elektronika

Hlavní řídicí deska se nazývá master deska. Její základ tvoří mikrokontroler. Obsahuje drivery pro krokové motory a také výstupy na ovládání servomotorů. Deska je napájena z baterie v roveru složené z tří li-ion článků v sérii. Komunikace s deskou je možná pomocí USB a CAN sběrnice. K této desce budou připojeny 3 slave desky. Tyto slave desky jsou umístěny na každém krokovém motoru. Snímají úhel natočení hřídele a ke každé desce jsou připojeny 2 Hallovy senzory. Slave desky jsou napájeny přímo z master desky. Schéma i deska plošných spojů byly navrženy programem KiCAD.



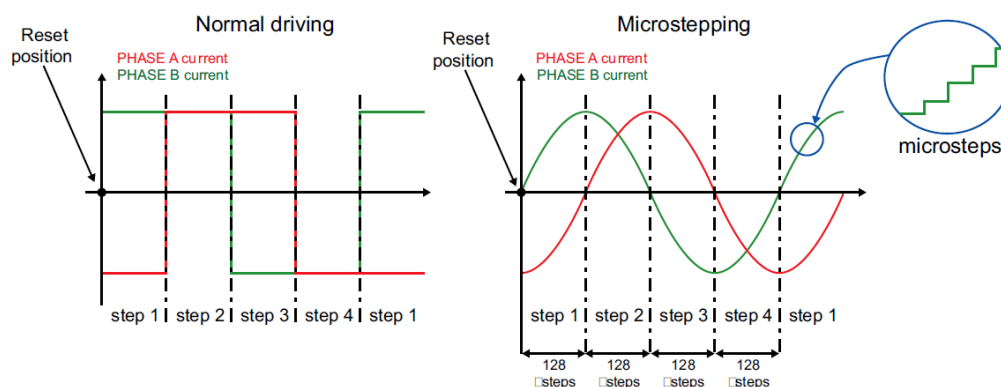
Obrázek 4.1: Blokové schéma řídicí elektroniky

## 4.1 Výběr komponent

### 4.1.1 Driver krokových motorů

Driver bipolárních krokových motorů byl zvolen L6470[12] od firmy STMicroelectronics. Driver je prověřen z předchozích projektů Katedry měření. Umožňuje napájení z 12 V a připojení separátního napájení pro logiku komunikace. Maximální stálý výstupní proud jsou 3 A, což je dostatek pro krokové motory 17HS4401 od firmy Hanpose, které vyžadují proud 1,5 A. Driver obsahuje plný H-můstek, a aby bylo možné spínat N MOSFET je zapotřebí externí zapojení s nábojovou pumpou (Pin CP), které vytváří napětí o velikosti 20 V (Pin VBOOT). V zapojení není použitý externí oscilátor. Pin SW umožňuje rychlé provedení pokynu, zde ale není použit z důvodu dostatečné rychlosti komunikace. Komunikace s driverem probíhá pomocí sběrnice SPI až s rychlostí 5 Mbit/s a driver je schopný detekovat a posílat chyby, jako je přehřívání, nadproud na výstupu, podpětí atd. Pin FLAG značí, že došlo k nějaké chybě. Integrovaný AD převodník zde není použit. Driver má implementovanou metodu microstepping. Metoda umožňuje rozdělit 1 krok motoru do více mikrokroků. Díky této metodě je chod motoru plynulejší a způsobuje méně vibrací. Umožňuje také vyšší přesnost. Microstepping je zde implementován tak, že místo obdelníkového průběhu má výstup tvar podobný sinusovému průběhu tvořeným pomocí PWM. Microstepping je znázorněn na obrázku 4.2.

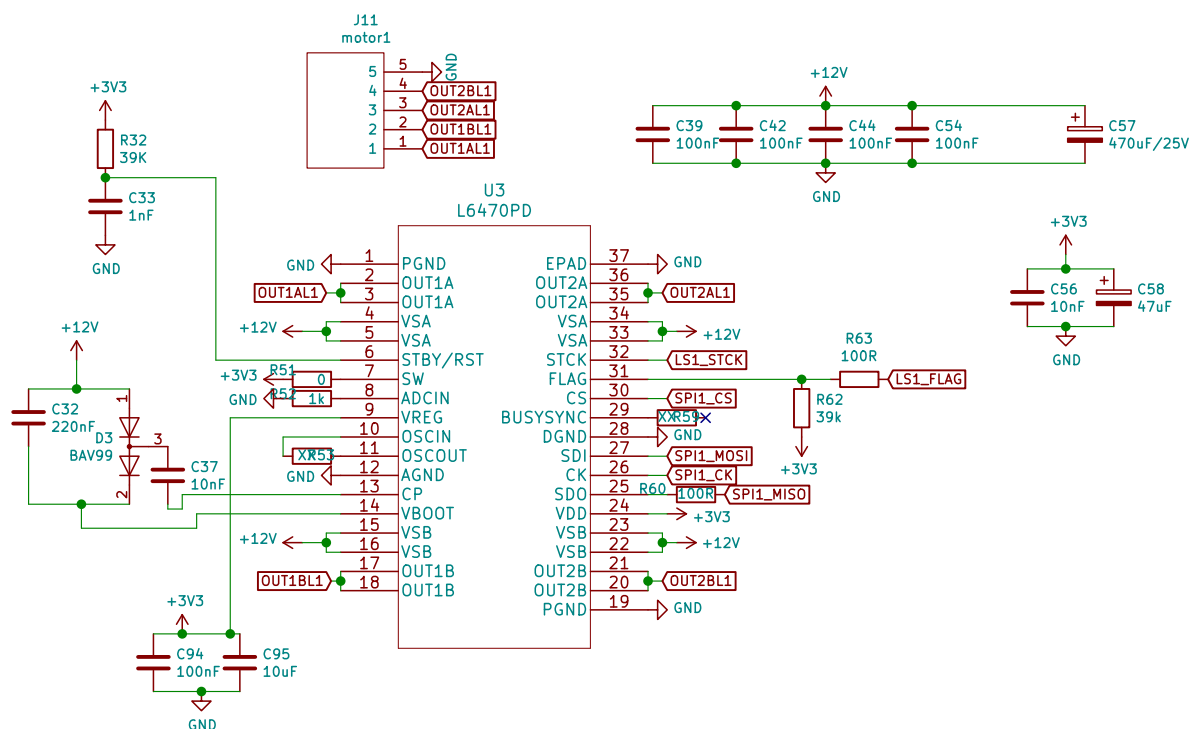
Driver nabízí až 128 mikrokroků.



Obrázek 4.2: Porovnání řízení s a bez microstepping [12]

Na desce jsou vytvořeny spoje pro 4 drivery krokové motor, ale jsou osazeny celkem 3 tyto drivery v 32 pinovém provedení s thermal pady. Neosazený driver slouží jako rezerva pro možnost vylepšení ruky. Schéma zapojení je na obrázku 4.3. Schéma vychází z doporučeného zapojení v datasheetu.

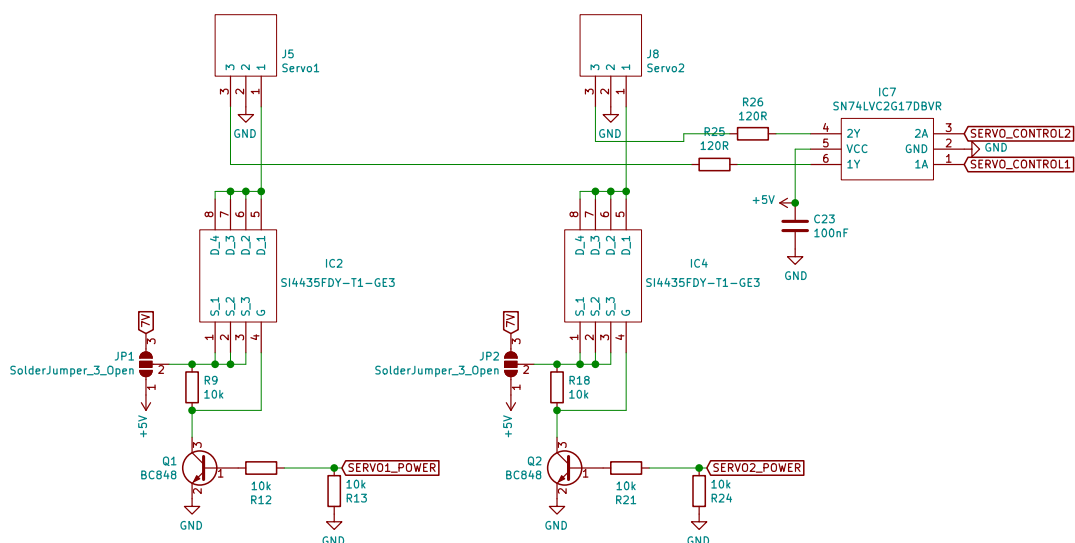




Obrázek 4.3: Schéma zapojení L6470

#### 4.1.2 Ovládání servomotorů

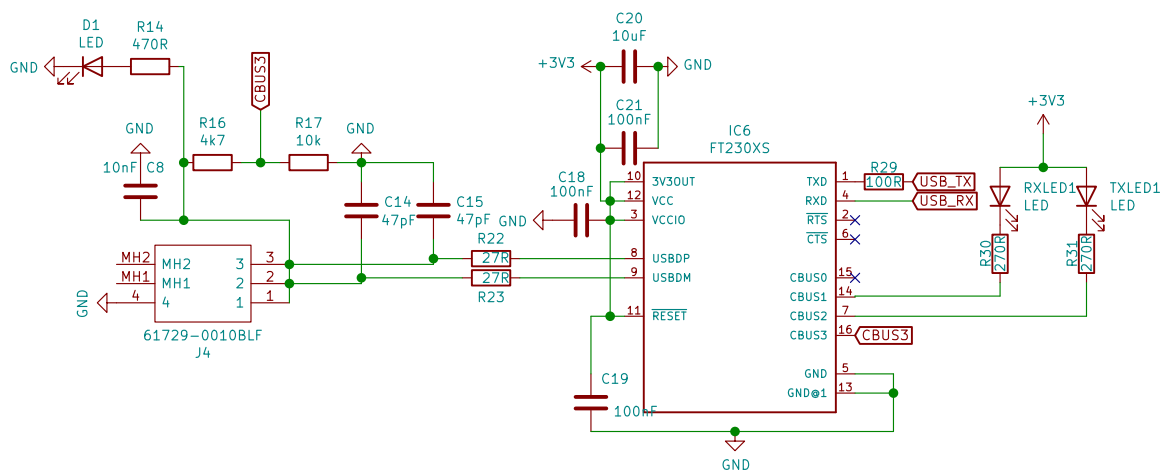
PWM signál pro ovládání servomotorů je generován pomocí časovačů v mikrokontroleru. Pro každý motor je jeden časovač. Byly použity 2 digitální servomotory. První, který ovládá chapadlo, je PDI-1171MG od firmy JX a je napájen z 5 V. Druhý, který hýbe chapadlem, je RDS3135MG od neznámého výrobce a je napájen ze 7 V. Jelikož je jeden servomotor napájen ze 7 V, mezi výstup časovačů a konektorů byl přidán budič, který zvýší maximální napětí ovládacího signálu z 3,3 V na 5 V. Tento budič má 2 nezávislé vstupy a výstupy. Napájení je možné vypnout skrz dva P-MOSFET, které jsou spínány z mikrokontroleru přes NPN tranzistory. Je tedy možné pro úsporu energie vypnout oba servomotory. Schéma zapojení je na obrázku 4.4.



Obrázek 4.4: Schéma zapojení ovládní servomotorů

### 4.1.3 USB

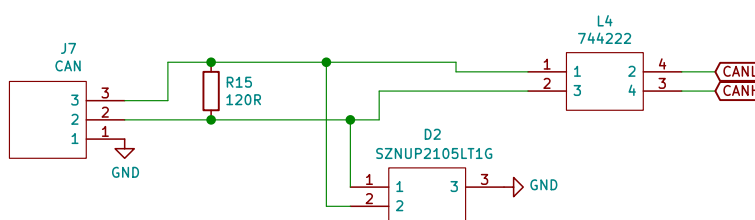
Pro komunikaci pomocí USB byl zvolen modul FT230X[3] od firmy FTDI-Chips. Modul převádí USB na serial UART, který je připojen na mikrokontroler. Modul je napájen z 5 V a je zapojen tak, že se napájí z desky nikoliv z USB a je v úsporném režimu, dokud se nepřipojí USB zařízení. Převádí rychlostí až 3 Mbaud. Pro logiku je připojeno 3,3 V napájení. U modulu jsou 3 LED, které signalizují připojení USB zařízení, odeslání a přijímání zpráv. USB konektor je zvolen typu B pro jeho robustnost. Schéma zapojení modulu vychází z datasheetu a je znázorněno na obrázku 4.5.



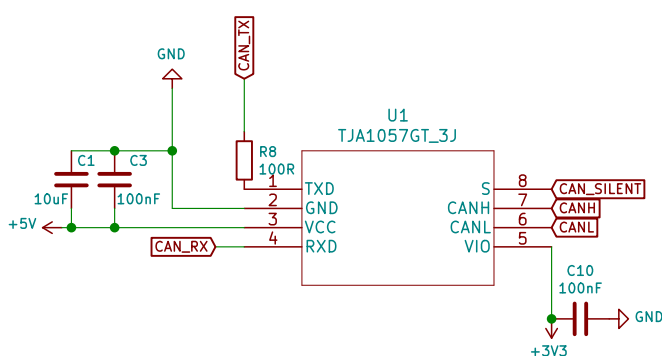
Obrázek 4.5: Připojení konektoru USB

### 4.1.4 CAN

Pro komunikaci pomocí CAN sběrnice byl zvolen TJA1057GT[10] od firmy NXP. Tento integrovaný obvod umožňuje připojení CAN kontroleru v mikrokontroleru do fyzické vrstvy CAN sběrnice. Mezi konektorem a vstupem do modulu je filtr souhlasného rušení 744222 od firmy Wurth a také dioda pro ochranu proti ESD SZNUP2105L[11] od výrobce ON semiconductor. Schéma zapojení vychází z datasheetů a je znázorněno na obrázcích 4.6 a 4.7.



Obrázek 4.6: Připojení konektoru CAN

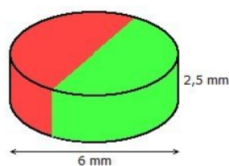


Obrázek 4.7: Připojení modulu CAN

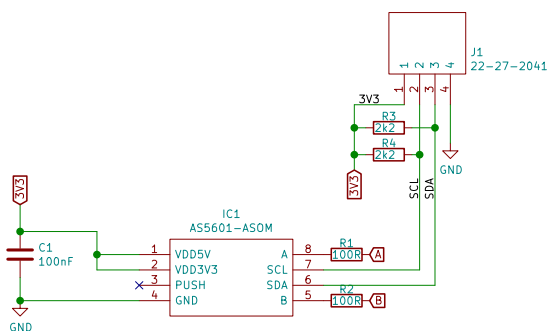
### 4.1.5 Senzor úhlu natočení

Jako senzor úhlu natočení byl zvolen AS5601[1] od firmy AMS. Jedná se o magnetický bezkontaktní absolutní senzor úhlu napájený z 3,3 V umístěný na slave desce. Jeho výstupem je kvadrurní signál, který je zapojen a zpracován mikrokontrolerem. Senzor je schopen komunikovat skrz sběrnici  $I^2C$ . Na senzoru je možné nastavit rozlišení až 2 048 bodů na jednu otáčku magnetu. Aby senzor fungoval, musí být magnet připevněný na hřídeli diametrálně magnetizovaný, viz obrázek 4.8. Pin Push umožňuje signalizovat přiblížení magnetu k senzoru. Jelikož je vzdálenost mezi motorem a magnetem konstantní, nelze tuto funkci využít.

Schéma zapojení je dle datasheetu senzoru a je uvedeno na obrázku 4.9.



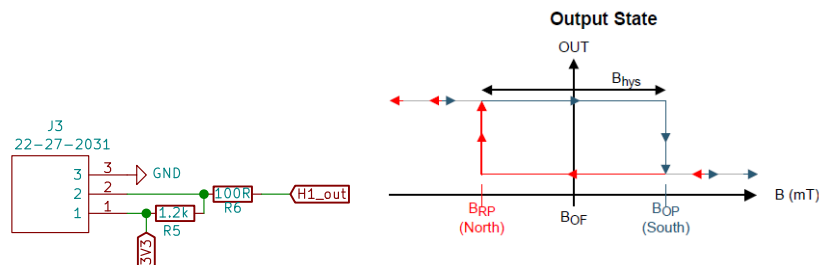
Obrázek 4.8: Magnet na hřídeli



Obrázek 4.9: Zapojení senzoru AS5601

#### 4.1.6 Hallův senzor

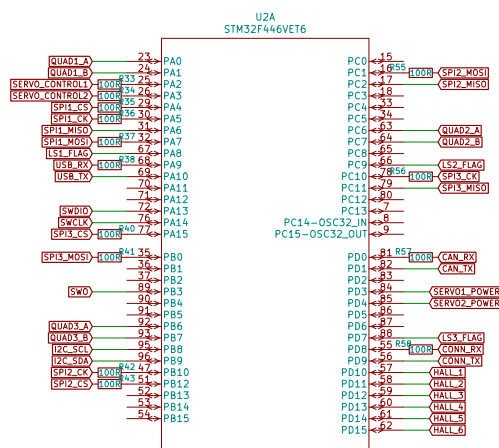
Jako Hallův senzor byl zvolen DRV5021[5] od firmy Texas Instruments. Výstup je otevřený kolektor 4.10. Mezi výstupem a napájením je pull up rezistor. Je napájen z 3,3 V a pokud se v jeho blízkosti objeví jižní magnetický pól, otevře se výstupní tranzistor a na výstupu je nulové napětí. Pokud se magnet v blízkosti senzoru nenachází, je výstup roven napájecímu napětí. Byla vybrána verze vyžadující minimálně 7 mT. Na každé slave desce se nachází 2 konektory na připojení těchto senzorů. Schéma vychází z datasheetu a je znázorněno na obrázku 4.10.



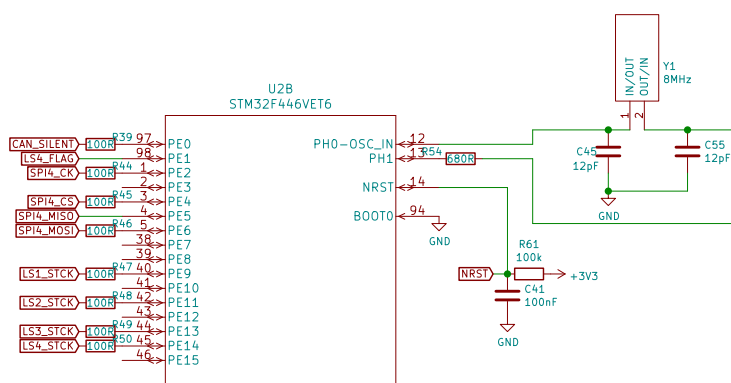
Obrázek 4.10: Zapojení Hallových senzorů a průběh výstupu senzoru [5]

### 4.1.7 Mikrokontroler

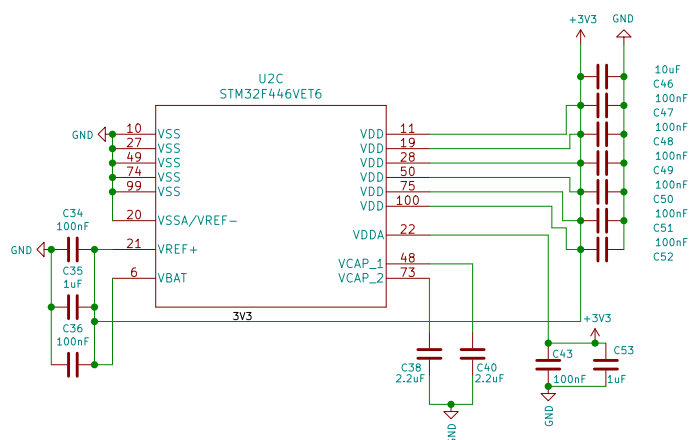
Mikrokontroler byl vybrán STM32f446VE[13] od firmy STMicroelectronics. Společnost STMicroelectronics byla upřednostňována kvůli předchozím zkušenostem s jejich mikrokontrolery. Je založen na 32-bitovém jádře Cortex M4 od společnosti ARM s frekvencí až 180 MHz. Disponuje 512 KB Flash pamětí. Je možné ho napájet z 3,3 V a umožňuje zapojení externího oscilátoru. Má tři 12-bitové AD převodníky s až 24 kanály a dva 12-bitové DA převodníky. Obsahuje 17 časovačů a až 114 vstupů/výstupů s možností přerušení. Umožňuje připojení 4  $I^2C$ , 4 USART, 2 UART, 4 SPI a 2 CAN sběrnic. Sběrnice UART je použita pro připojení USB modulu. 3 SPI sběrnice jsou použity pro drivery krokových motorů. Jedna  $I^2C$  sběrnice je použita na programování senzorů úhlu a CAN sběrnice je použita na připojení integrovaného obvodu TJA1057GT. Poskytuje tedy dostatek všech rozhraní pro řízení robotické ruky. Pouzdro má 100 pinů. Vzhledem k výrobě jednoho kusu je výkon, počet pinů a funkcionalit předdimenzován. Je zde použit 8 MHz externí krystalový oscilátor. Zapojení mikrokontroleru je znázorněno na obrázcích 4.11 4.12 4.13 a vychází z datasheetu.



Obrázek 4.11: Zapojení mikrokontroleru část 1.



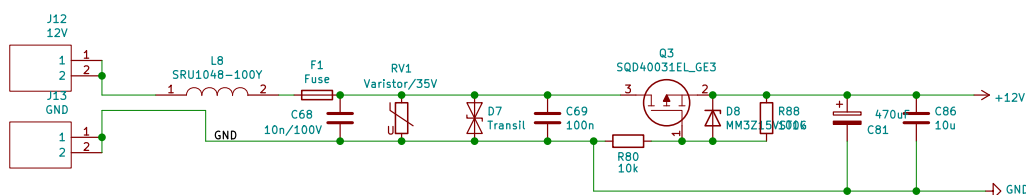
Obrázek 4.12: Zapojení mikrokontroleru část 2.



Obrázek 4.13: Zapojení mikrokontroleru část 3.

#### 4.1.8 Vstupní ochrana

Vstupní ochrana se skládá z ochrany před přepólováním vstupního napětí, přepětí a nadproudů. Jako ochrana před nadproudem je použita pojistka v pouzdře, které umožňuje snadnou výměnu. Ochrana proti přepětí je realizována pomocí obousměrného transilu a varistoru. Varistor byl zvolen MOV-14D180K[2] od výrobce Bourns. Při 18 V propouští 1 mA a při upínacím napětí 36 V propouští 10 A. Transil byl vybrán SMBJ16CA[9] od výrobce Littelfuse. Při 16 V propouští 1 mA a při upínacím napětí 26 V propouští 26 A. Ochrana proti přepólování je navržena pomocí P-MOSFET SQD40031EL[17] od společnosti Vishay, který je sepnut pouze při správné polaritě napájecího napětí. Při sepnutí má odpor 3 mΩ. Mezi gate a source je umístěna zenerova dioda na 15 V, aby ochránila před dosažením napětí mezi gate a source 20 V, což je maximální dovolené napětí. Dále jsou zde umístěny kondenzátory a cívka pro vyhlazení napěťových a proudových špiček. Schéma vychází z datasheetů a je znázorněno na obrázku 4.14.



Obrázek 4.14: Vstupní ochrana

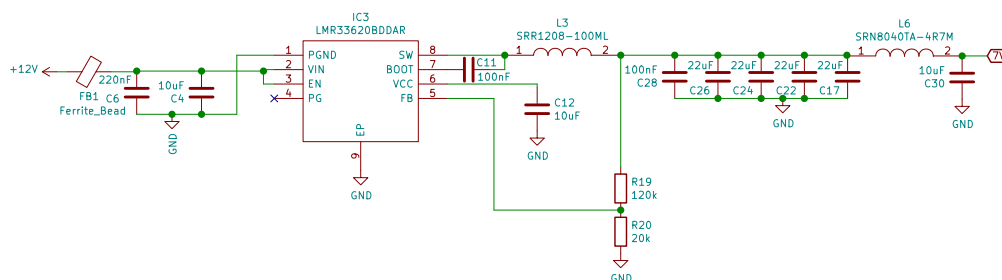
### 4.1.9 Regulátory

Komponenty zmíněné v této kapitole vyžadují 4 úrovně napájecího napětí. Pro drivery krokových motorů je využito přímo napájecí napětí master desky.

Velký servomotor pohybující s chapadlem vyžaduje 7 V. Regulátor napětí na 7 V byl zvolen LMR33620[6] od výrobce Texas instruments. Jedná se o DC-DC spínaný step-down regulátor napětí. Spínací frekvence je od 400 kHz až do 2.1 MHz a maximální výstupní proud jsou 2 A. Regulátor má účinnost až 95 % a spínací frekvenci mění podle zatížení. Před regulátor byla umístěna feritová perla pro filtrování vysokofrekvenčního rušení a za regulátor je umístěn LC filtr. Ten je přidán pro dodatečné filtrování výstupního napětí. Velikosti rezistorů ve zpětné vazbě určují velikost výstupního napětí regulátoru a byly vypočítány pomocí následujícího vzorce.

$$R_{20} = \frac{R_{19}}{\frac{V_{out}}{V_{ref}} - 1} \quad (4.1)$$

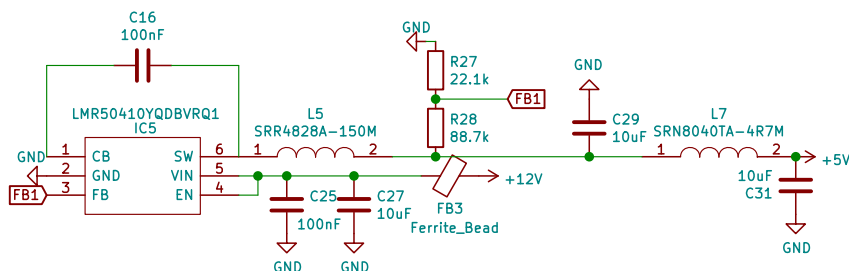
$V_{out}$  značí žádané výstupní napětí a  $V_{ref}$  je napětí na vstupu zpětné vazby, zpravidla volené jako 1 V. Označení odporů odpovídá schéma. Zapojení je na obrázku 4.15.



Obrázek 4.15: Regulátor napětí 7V

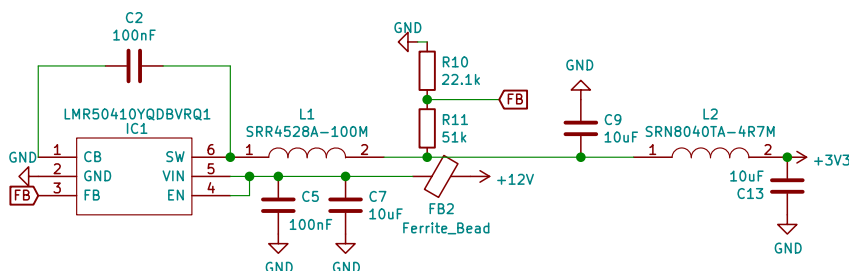
Pro napájení USB modulu a servomotoru ovládající chapadlo je potřeba 5 V napájecí napětí. Regulátor napětí byl zvolen LMR50410[7] od výrobce Texas Instruments. Jedná se o DC-DC spínaný step-down regulátor napětí. Spínací frekvence je 700 kHz a maximální výstupní proud je 1 A. Zapojení je téměř identické jako zapojení 7 V regulátoru napětí. Přidán LC filtr na

výstupu, feritová perla, kondenzátory na přívodu napájení a rezistory tvořící napěťový dělič ve zpětné vazbě. Zapojení je na obrázku 4.16.



Obrázek 4.16: Regulátor napětí 5V

Pro napájení mikrokontroleru, CAN modulu, slave desek a logických výstupů všech modulů je zapotřebí napájecí napětí o velikosti 3,3V. Regulátor napětí byl zvolen stejný regulátor jako 5 V regulátor. Jediný rozdíl je velikost rezistorů ve zpětné vazbě a velikost indukčnosti cívky v prvním LC filtru. Zapojení je na obrázku 4.17.



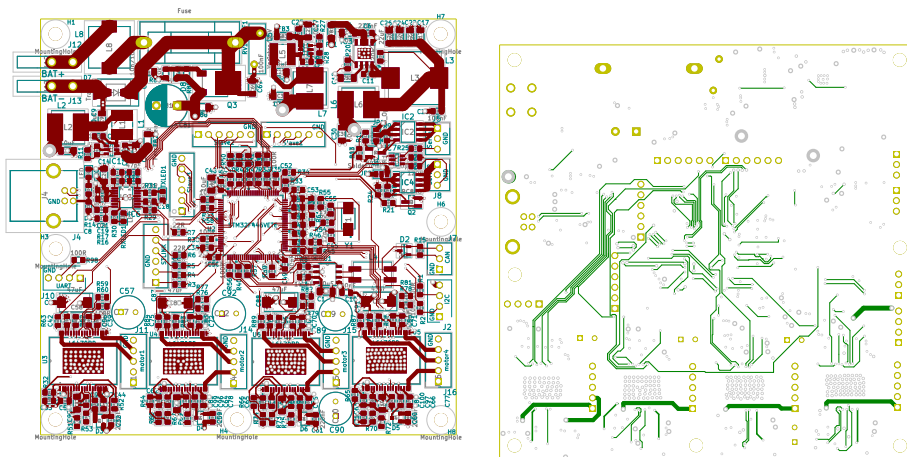
Obrázek 4.17: Regulátor napětí 3,3V

## 4.2 Desky plošných spojů

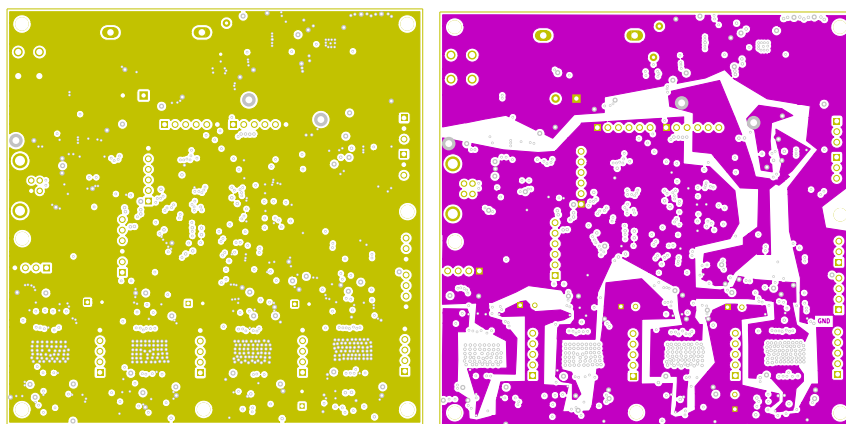
Velikost desky plošných spojů byla vybrána 10x10 cm dle nabídky výrobce a je složena ze čtyř vrstev. Na horní vrstvě jsou umístěny všechny součástky a je zde vedeno co nejvíce spojů. Druhá vrstva je celá pokryta zemí, tedy nulovým napětím. Třetí vrstva je určena na napájení. Na této vrstvě se nacházejí 4 oblasti, kterými se rozvádí napájecí napětí k součástkám. Mezi oblastmi je dostatečná izolační mezera při použití 12 V napájecího napětí. Oblasti mají dostatečnou plochu ve všech částech pro zajištění malého odporu. Na čtvrté vrstvě jsou vedeny spoje, které nebylo možné realizovat na první vrstvě. Pro přechod mezi vrstvami jsou použity prokovy o velikost 0,3 až 0,6 mm. Spoje jsou široké od 0,2 mm pro signály až po 3,2 mm pro napájecí cesty s většími proudy. Deska obsahuje konektory pro připojení napájení pomocí fastonů. Dále obsahuje konektory pro 3 krokové motory, 2 servomotory, 3



slave desky, programování mikrokontroleru, programování slave desky, USB a CAN. Jednotlivé vrstvy jsou zobrazeny na obrázcích 4.18 a 4.19. Deska byla vyrobena společností Allpcb. Detailnější obrázky jsou uvedeny v příloze.

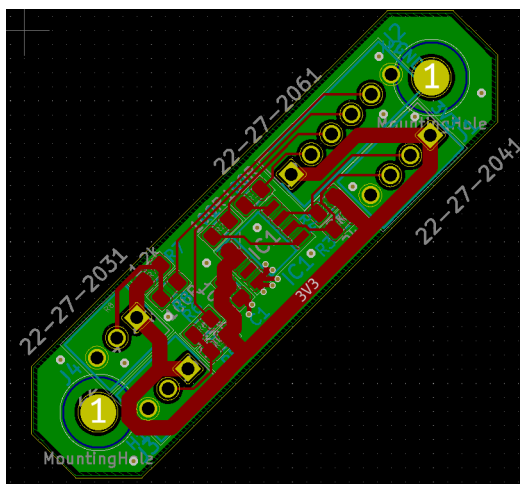


Obrázek 4.18: Signálové vrstvy



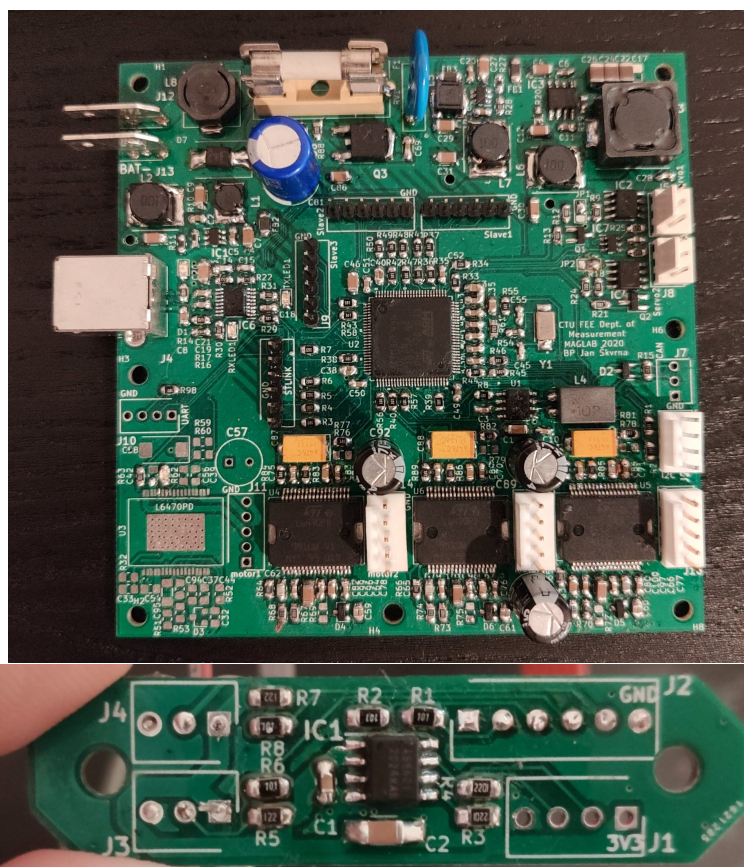
Obrázek 4.19: Druhá zemnicí vrstva a třetí napájecí vrstva

Slave deska má rozměry 5x1,4 cm. Jedná se o dvouvrstvou desku. Má zaoblené hrany a díry na šrouby jsou přesně navrženy, tak aby šla deska přidělat diagonálně na krokový motor o velikosti NEMA 17. Na první vrstvě jsou všechny komponenty a vedou se zde signálové a napájecí spoje. Druhá vrstva je celá pokrytá zemí. Při návrhu byly použity šířky obdobné jako u master desky. Slave deska má 4 konektory - 2 pro Hallové senzory, 1 pro programování senzoru úhlu a 1 na napájení a komunikaci s deskou. Slave deska je zobrazena na obrázku 4.20.



Obrázek 4.20: Návrh slave desky

Desky byly osazeny autorem práce a jsou zobrazeny na obrázcích 4.21.



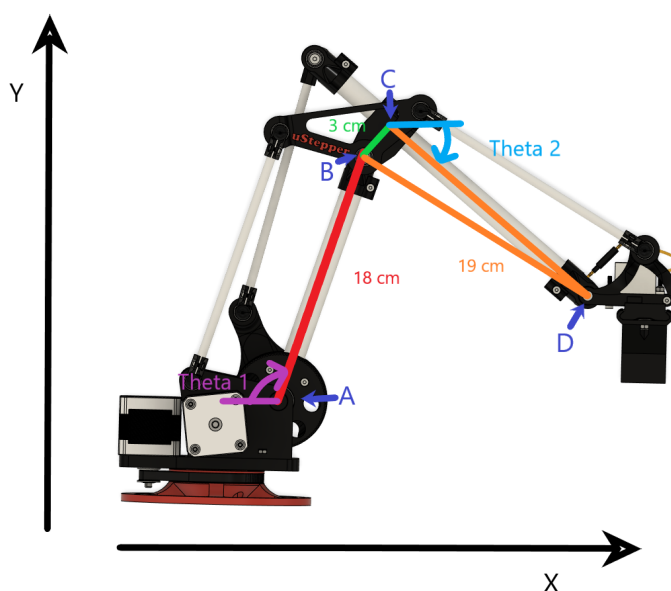
Obrázek 4.21: Osazená master a slave deska

## Kapitola 5

### Pokročilé algoritmy pro řízení

Ovládat robotickou ruku přímo pomocí úhlů natočení motorů by bylo velmi nepraktické a zcela nemožné pro nezkušeného člověka. Z tohoto důvodu se pohyb ruky ovládá pomocí tří os - osa X, osa Y a rotace kolem základny. Výpočet úhlů motorů je založen na následujících pozorování.

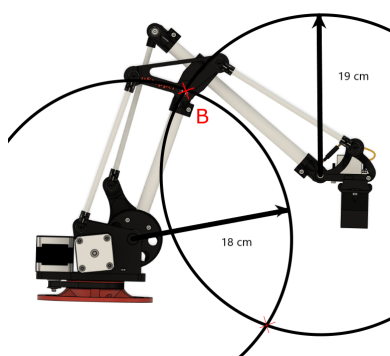
Nejprve byly změřeny velikosti ramen. Naměřené hodnoty, úhly a body, které jsou použity při výpočtu jsou znázorněny na obrázku 5.1.



Obrázek 5.1: Velikosti ramen a body

Je zadán bod A jako počátek se souřadnicemi  $[0,0]$ . Dále je zadán koncový bod D. Na základě těchto dvou informací lze vypočítat úhly  $\Theta_1$  a  $\Theta_2$ . Bod B lze vypočítat jako průsečík dvou kružnic. První kružnice má střed v bodě A a poloměr 18 cm a druhá kružnice střed v bodě D a poloměr 19 cm. Pro jednodušší výpočet je implementována funkce `circirc`, jejímž vstupem jsou středy a poloměry a výstupem jsou průsečíky. Průsečíky těchto kružnic jsou 2. Oba průsečíky jsou zobrazeny na obrázku 5.2. Druhý průsečík není přípustný a proto není využit. Nyní je znám bod B. Z bodu B lze určit úhel  $\Theta_1$ . Výpočet probíhá pomocí následujícího vzorce:

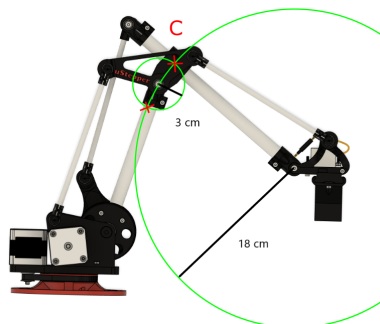
$$\Theta_1 = \text{atan2}(B_y, B_x); \quad (5.1)$$



Obrázek 5.2: Výpočet bodu B

Z bodu B lze zjistit bod C jako průsečík dvou kružnic. První kružnice má střed v bodě B a poloměr 3 cm. Druhá kružnice má střed v bodě D a poloměr 19 cm. Pomocí funkce `circirc` je vypočítán bod C. Průsečíky jsou znovu dva, ale druhý je nepřípustný. Oba jsou zobrazeny na obrázku 5.3. Body C a D jsou použity k výpočtu úhlu  $\Theta_2$  a to za pomoci následujícího vzorce.

$$\Theta_2 = -\text{atan2}(D_y - C_y, D_x - C_x) \quad (5.2)$$



Obrázek 5.3: Výpočet bodu B

Dále jsou úhly převedeny z radiánů na stupně a jsou od nich odečteny příslušné konstanty tak, aby nulový úhel odpovídal koncovému dorazu ramen. Byly naměřeny maximální a minimální úhly motorů a konstantami jsou převedeny na příkazy do motorů tak, aby odpovídaly maximálním a minimálním hodnotám sensorů úhlů. Převody na příkazy pro krokové motory odpovídají následujícím rovnicím.

$$Pozice_1 = \Theta_3 \cdot 27.5 \quad (5.3)$$

$$Pozice_2 = (\Theta_1 - 45) \cdot 27 \quad (5.4)$$

$$Pozice_3 = (\Theta_2 - 15) \cdot 26 \quad (5.5)$$

$Pozice_1$  značí rotaci kolem základny a  $\Theta_3$  je přímo přijímán z rovera či aplikace.  $Pozice_2$  značí motor měnící úhel  $\Theta_1$  a  $Pozice_3$  značí motor měnící úhel  $\Theta_2$ . Pokud je řešení mimo pracovní prostor, zachová se poslední možné řešení. Na obrázku 5.4 je zobrazen kód, který počítá inverzní kinematickou transformaci. Výstup funkce `circirc` se ukládá do globálního pole `output`. Proměnné `desiredPosition` značí žádanou pozici motoru převedenou tak, aby korespondovala s výstupem senzoru úhlu.

```

circirc(0,0,18,x,y,19);

float Ax = output[0];
float Ay = output[1];

circirc(Ax,Ay,3,x,y,19);

float Bx = output[0];
float By = output[1];

theta2 = -atan2(y-By,x-Bx)* (180.0/3.14);
theta1 = atan2(Ay,Ax) * (180.0/3.14);

/**
 * Pokud je reseni mimo nase moznosti pouzij to minule
 */

if(theta1 > 180 || theta1 < 45 || theta2 < -15 || theta2 > 120){
    theta1 = OKtheta1;
    theta2 = OKtheta2;
}
else{
    OKtheta1 = theta1;
    OKtheta2 = theta2;
}

/**
 * prepocet uhlu na chtene pozice
 */
desiredPosition2 = (theta1-45)*27;
desiredPosition3 = (theta2+15)*26;

```

**Obrázek 5.4:** Ukázka kódu

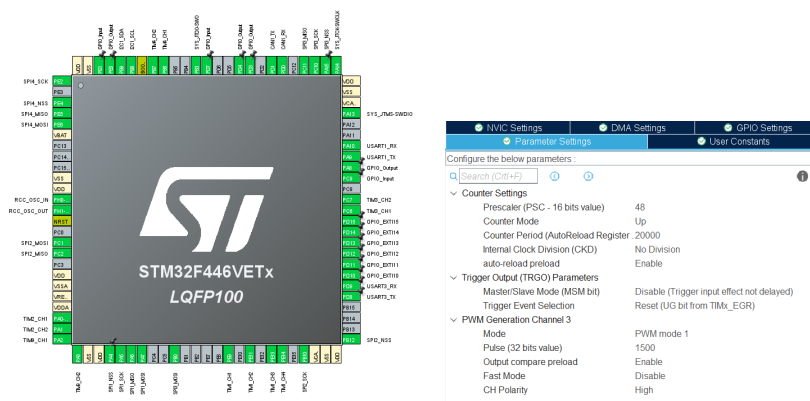
Prvotní návrh inverzní kinematické úlohy nepočítal s bodem B, ale pouze s bodem C. Protože nebral v potaz pohyb bodu C po kružnici, docházelo k nepřesnostem hlavně při pohybech, kdy se úhel mezi bodem B a C měnil. Z tohoto důvodu byla navržena stávající inverzní kinematická transformace, která je přesnější.



## Kapitola 6

### Řídicí Firmware

Řídicí firmware je napsán v programovacím jazyku C/C++ za pomoci programu STM32CubeIDE [14]. Po zapnutí desky se nejprve nainicializují všechny periferie a globální proměnné, které jsou v kódu použity. Inicializace periferií je vygenerována z programu STM32CubeMX [15], ve kterém jsou zvoleny všechny parametry použitých sběrnic, časovačů, oscilátoru a nastavení vstupních a výstupních pinů. Tato aplikace velmi ulehčuje psaní kódu. Ukázka z aplikace je na obrázku 6.1



**Obrázek 6.1:** ukázka nastavení pinů mikrokontroleru a nastavení časovače s PWM výstupem v STM32CubeMX [15]

Po inicializaci se zapnou časovače a zapne se přijímání z USB modulu. Poté probíhá nekonečný while cyklus. Ve while cyklu se nejprve vypočítají žádané příkazy pro motory a to na základě toho, jestli se jedná o běžný běh ruky nebo je spuštěná kalibrace. V případě kalibrace jsou příkazy pro motory pevně dané konstantou. V případě běžného běhu se příkazy pro motory vypočítají P

regulátorem na základě žádaného úhlu kloubu. Poté se příkazy odešlou přes sběrnici SPI do driverů krokových motorů a změní se střída PWM výstupu časovačů pro servomotory. Poté je zpracován jeden byte z fronty, kde se přijímají zprávy. Zde příkazy ve while cyklu končí a začíná se od začátku while cyklu.

Výstupy z Hallových senzorů jsou zpracovány pomocí GPIO pinů s přerušením. Výstupy senzorů úhlu jsou zpracovávány časovači. Zprávy jsou přijímány skrz přerušení a jsou přidávány do kruhové fronty.

## ■ 6.1 Zpracování vstupů

### ■ 6.1.1 Hallovy senzory

Výstupy Hallových senzorů jsou připojené na piny PD11-PD15. Každý senzor na jeden pin. Tyto piny jsou nastavené jako GPIO piny s přerušením, které je vyvoláno při náběžné či sestupné hraně. Přerušení vyvolá port D, tudíž nelze rozoznat, který pin vyvolal přerušení. Z tohoto důvodu se při zpracování přerušení vyčtou všechny aktuální hodnoty pinů a ty se uloží.

### ■ 6.1.2 Senzory úhlu natočení

Kvadrurní výstup senzorů úhlu natočení jsou zpracovávány pomocí 3 různých časovačů. Tyto časovače jsou nastaveny v enkodérovém módu. Čítače v časovačích jsou šestnáctibitové. Časovač pracuje na frekvenci 48 MHz. Použity jsou časovače 2,4,8. Díky kvadrurnímu výstupu je rozlišeno jakým směrem se motor točí. Protože se čítač inicializuje na hodnotě 0 a při pohybu v záporném směru čítač ihned přeteče, tak se po vyčtení aktuální hodnoty čítače v časovači hodnota převede z unsigned integer na signed integer. Převod uvažuje hodnotu větší nebo rovno jak 30 000 jako zápornou ( $-65\,535 + \text{hodnota čítače}$ ) a hodnotu menší jak 30 000 jako kladnou. Tento převod můžeme použít, protože senzor na všech motorech nenasčítá při pohybu více jak 10 000 bodů. Pracovní rozsah motorů starajících se o pohyb na osách X a Y je velký přibližně 150 stupňů a pro rotaci kolem základny 360 stupňů. Jedna otáčka motoru je 2 048 bodů a motor je z převodován skrz řemeny. Experimentálně byla naměřena hodnota přibližně 10 000 pro rotaci kolem základny a 4 000 pro ostatní krokové motory. Průběh výstupu senzoru při rozběhu motoru je zobrazen na obrázku 6.2.

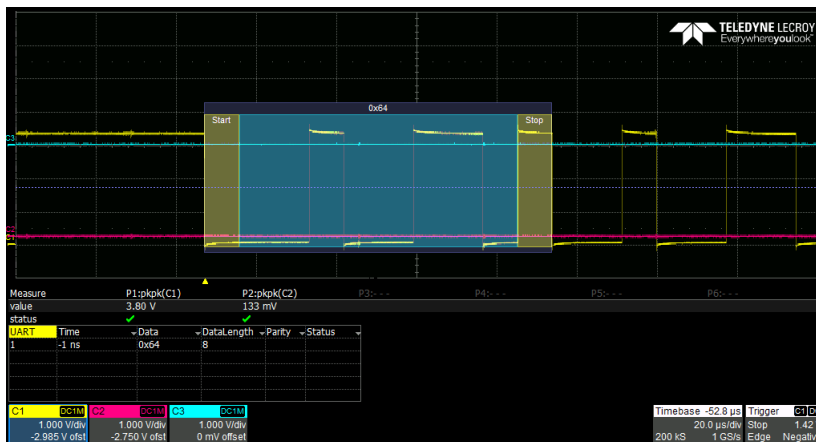




Obrázek 6.2: Kvadraturní výstup senzoru úhlu při rozběhu motoru

### 6.1.3 Komunikace přes USB

Přijímání zpráv přes USB probíhá asynchronně. Při inicializaci se zavolá funkce, která spustí přijímání zpráv přes UART s velikostí 6 bytů. Při přijetí zprávy se zavolá přerušení, které uloží zprávu do kruhové fronty a znovu zavolá funkci přijímající dalších 6 bytů. Pro příjem zpráv skrz UART slouží šestibytový buffer. BaudRate je zvolen jako 115 200 s žádnou paritou. Přenos jednoho bytu je zobrazen na obrázku 6.3.



Obrázek 6.3: Přenos bytu přes UART

## 6.2 Komunikace přes CAN

Vzhledem k nastalé situaci nebylo možné spolupracovat s autorem rovera a tudíž podpora komunikace přes CAN nebyla implementována. Komunikace

přes CAN bude probíhat obdobně jako komunikace přes USB. Budou přenášeny zprávy s žádanou polohou ruky se stejným formátem zpráv a budou se stejným způsobem ukládat do stejné kruhové fronty.

### 6.3 Zpracování zpráv

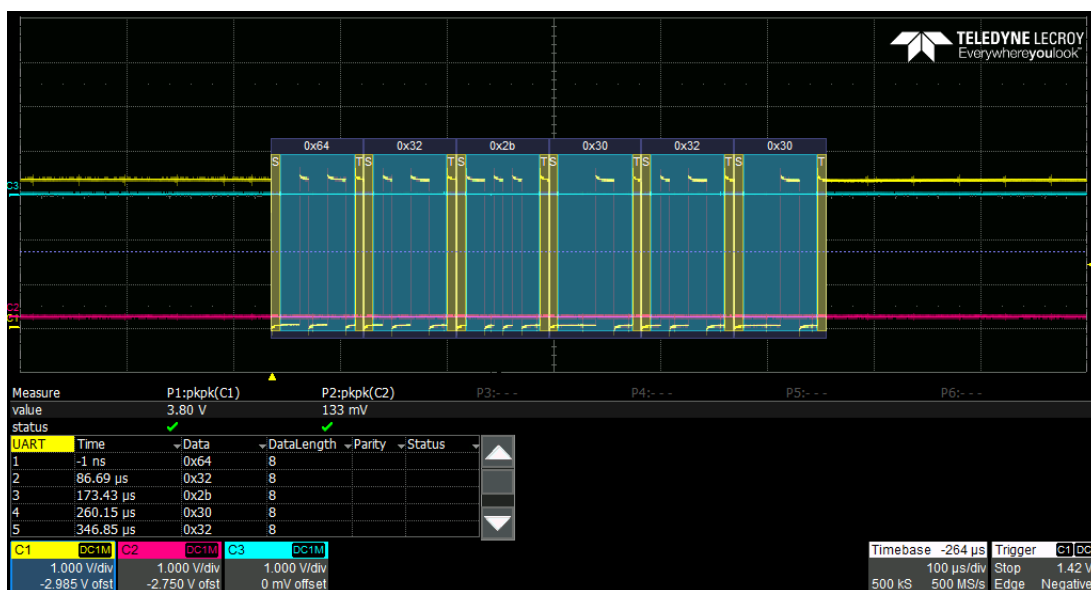
Všechny zprávy jsou přidávány do kruhové fronty o velikosti 256 bytů. Jelikož jsou všechny zprávy 6 bytové, kruhová fronta je schopná uchovat až 42 zpráv. Není potřeba volit větší kruhovou frontu, protože zpracování zpráv probíhá rychleji než jsou přijímány. Zprávy přichází primárně jako pokyn pro změnu polohy a nejsou tudíž odesílány často.

Každým projitím while cyklu je zpracován jeden byte zprávy. Ke zpracování dochází po odeslání příkazů pro drivery krokových motorů. Každá šestibytová zpráva má na prvním bytu písmeno, které značí o jaký typ zprávy se jedná. Níže je uvedena tabulka zpráv 6.1, které jsou implementovány.

Tabulka 6.1: Přijímané zprávy

Obsah zprávy	Reakce desky
s	zapne napájení pro servomotory
c	zapne kalibraci
o	vyčte a zašle jako odpověď status registry driverů krokových motorů
t	vyčte a zašle jako odpověď status registr senzoru úhlu
e	nastaví maximální rozlišení senzoru úhlu
v	zapiše nastavené hodnoty burn in příkazem do senzoru úhlu
d	<p>příjem dat má následující formát:</p> <p>1. byte 'd', 2. byte hodnota 1-5 značí číslo souřadnice (1-rotace kolem základny, 2-osa x, 3-osa z, 4-5 servomotory).</p> <p>3. byte je znaménko, 4. až 6. byte trojčíferné číslo značící hodnotu.</p>

Příklad přenosy zprávy typu data je uveden na obrázku 6.4. První byte má hodnotu 0x64, což v ASCII tabulce značí znak 'd'. Jedná se tedy o zprávu typu data. Další je byte o hodnotě 0x32, což je znak 2, tudíž se jedná o změnu na ose X. Další byte 0x2b značí znaménko +. Poslední tři byty značí žádanou hodnotu, která je zde '020'. Po přijetí této zprávy se změní žádaná hodnota na ose X na 20.



Obrázek 6.4: Přenos zprávy typu data přes USB

Pokud se nejedná o zprávu typu data, jsou všechny ostatní byty v šesti-bytové zprávě zahazeny. Jelikož osa Y může nabývat záporných hodnot, je pro zjednodušení přijímání počítáno s tím, že odeslaná hodnota má v sobě přičtenou hodnotu 100, která se při příjmu odečte. V tomto případě je vždy znaménkový byte znak '+'.

## 6.4 Zpracování výstupů

### 6.4.1 Servomotory

Servomotory jsou řízeny PWM signálem z časovačů 5 a 9. Vstupní hodinový signál do časovačů je 8 MHz a při použití čítače do hodnoty 20 000 a prescaleru 48 je dosažena výstupní frekvence 50 Hz. Toto nastavení je stejné pro oba časovače. Hodnota pulzu se kterou jsou časovače inicializovány je 1 500, což odpovídá střídě 15 %. Změna střídě probíhá skrz funkci ChangePulse, jejímž vstupem je třída časovače, číslo kanálu a délka pulzu. Pomocí makra se přepíše compare registr v časovači a tím se změní velikost pulzu. Ta značí kolik hodinových cyklů z nadefinovaných 20 000 cyklů má být výstupní hodnota logická jednička, zbylý počet hodinových cyklů je výstup v logické nule.

## 6.4.2 Krokové motory

Drivery krokových motorů komunikují skrz sběrnici SPI. Velikost přenášených rámců je 8 bitů s tím, že první se přenáší nejvíce významný bit. Rychlost komunikace je nastavena na 1.5 Mbit/s. Polarita hodinového signálu při nečinnosti je logická jednička. Fáze hodinového signálu je nastavena tak, že se bit snímá při náběžné hraně hodinového signálu. Při nečinnosti je signál chip select logická jednička. Před přenosem je změněn na logickou nulu a ponechán v tomto stavu po celou dobu přenosu. Po přenosu je znovu změněn na logickou jedničku. Po přenesení rámce je nutné počkat 800 ns, aby byl driver schopný rámec zpracovat. Jelikož HAL knihovny poskytují funkci pro uspání pouze v rozlišení milisekund byla implementována funkce `void delayus(int)`, která vyčká na základě vstupu v mikrosekundách. Příkazy jsou 1-4 bytové.

Při inicializaci se do všech driverů pošlou příkazy, které nastaví parametry akcelerace, decelerace a maximální rychlost. Hodnota, která se zapisuje, je počet kroků za hodinový tik. Akcelerace je zvolena 1 280 kroků/tik, decelerace 5 120 kroků/tik a maximální rychlost 2 560 kroků/tik. Nutno podotknout, že se jedná o tzv. mikrokroky, které dělí 1 krok motoru na 128 mikrokroků. Těmito hodnotami jsou dosaženy plynulé pohyby ruky. Použitý interní oscilátor má frekvenci 16 MHz. Příkaz na zápis hodnoty je 4 bytový a je zobrazen na obrázku 6.5.

Table 39. SetParam command structure

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	0	0	PARAM					From host
VALUE Byte 2 (if needed)								
VALUE Byte 1 (if needed)								
VALUE Byte 0								

Obrázek 6.5: Příkaz set parameter

Poloha krokových motorů je řízena pomocí P regulátoru. Nejprve se pomocí inverzní kinematické úlohy vypočítají žádané úhly motorů. Tyto úhly se poté převedou na pozici, kterou snímá senzor úhlu. Pokud je rozdíl mezi žádanou polohou a aktuální polohou menší než 5 bodů, pošle se příkaz na zastavení motoru `SoftStop`, který je 1 bytový. Driver nabízí další příkaz na zastavení se jménem `HardStop`, který nerespektuje nastavenou maximální deceleraci. Z důvodu zamezení nadměrného namáhání konstrukce je zde použit příkaz `SoftStop`. Rozdíl pěti bodů je v této aplikaci ruky zanedbatelný, protože 5 bodů značí rozdíl pro  $\Theta_1$  a  $\Theta_2$  0,04 stupně a pro rotaci kolem základny 0,02 stupně. V případě, že je rozdíl větší, zašle se do motoru 4 bytová zpráva `Run`, která je znázorněna na obrázku 6.6. Složka P regulátoru byla experimentálně určena na hodnotu 10. Díky velkému rozlišení motorů a určené aplikaci je P regulátor dostačující i k zajištění zanedbatelné odchylky. Pokud se zadaná

rychlost motoru od posledního projítí while cyklem nezměnila, příkaz Run se znovu neodesílá.

**Table 41. Run command structure**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	0	1	0	0	0	DIR	From host
X	X	X	X	SPD (Byte 2)				From host
SPD (Byte 1)								From host
SPD (Byte 0)								From host

**Obrázek 6.6:** Příkaz run

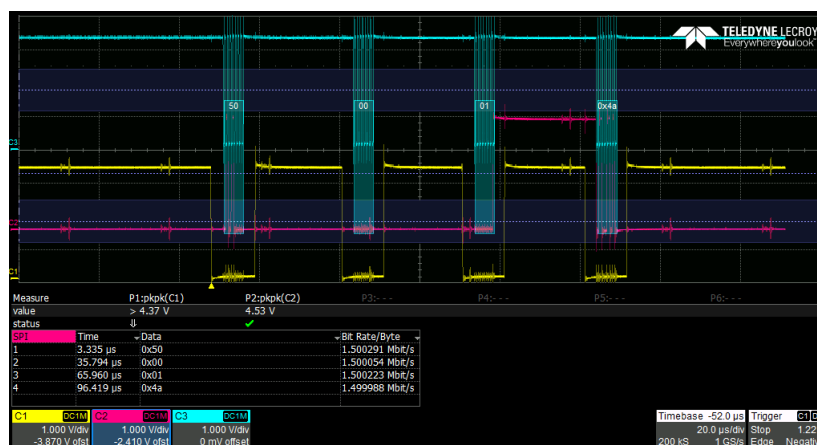
Při debugování byl použit 1 bytový příkaz GetStatus na který driver odpovídá 2 byty obsahující status registr. Obsah tohoto registru je zobrazen na obrázku 6.7. Ze status registru lze vyčíst důležité chyby, jako je ztráta kroku, nadproud, vypnutí kvůli přehřátí, varování kvůli přehřátí, podpětí, špatný příkaz, příkaz nelze provést, stav motoru a směr.

**Table 34. STATUS register**

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
SCK_MOD	STEP_LOSS_B	STEP_LOSS_A	OCD	TH_SD	TH_WRN	UVLO	WRONG_CMD
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NOTPERF_CMD	MOT_STATUS		DIR	SW_EVN	SW_F	BUSY	HiZ

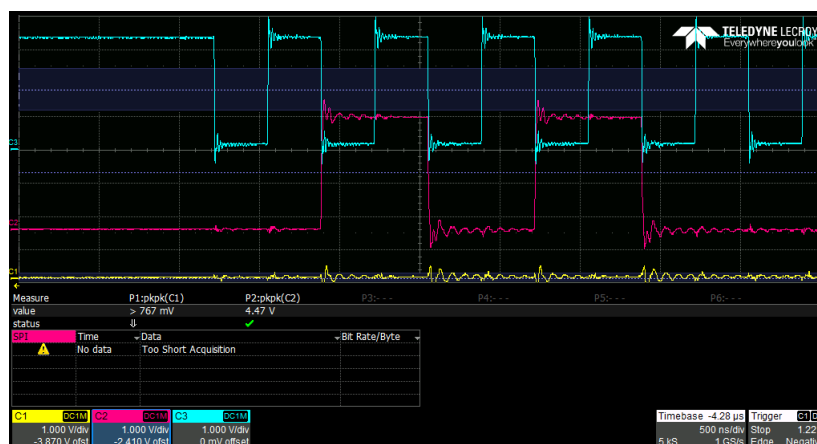
**Obrázek 6.7:** status registr

Na následujících obrázcích 6.8 a 6.9 je zobrazena komunikace přes sběrnici SPI. Jedná se o přenos příkazu Run. První byte zprávy značí typ příkazu a další 3 byty jsou data step/clock. Mezi byty je prodleva 100  $\mu$ s, aby měl driver dostatek času byty zpracovat. Modrý signál je hodinový signál, žlutý je chip select a červený jsou data.



Obrázek 6.8: Příkaz run, hodinový signál je modrý, data červené a chipselect žlutý

První byte s hodnotou 0x50 značí příkaz run, který obsahuje směr a další 3 byty značí rychlost pohybu, v tomto případě 330 steps/tick.



Obrázek 6.9: Detail přenosu, hodinový signál je modrý, data červené a chipselect žlutý

### 6.4.3 Komunikace se senzory úhlu

Pro komunikaci se senzory úhlu je použita sběrnice  $I^2C$ . Na master desce je jeden konektor pro připojení jedné slave desky za účelem naprogramování senzoru. Rychlost sběrnice je nastavena na 10 kHz. Senzor má adresu 0x36. Pro nastavení největší přesnosti je nutné zapsat hodnotu 0x0f do registru ABN. Význam registru ABN je zobrazen na obrázku 6.10.

ABN Mapping		
ABN(3:0)	3:0	Output Positions and Update Rate 0000 : 8 (61 Hz) 0001 : 16 (122 Hz) 0010 : 32 (244 Hz) 0011 : 64 (488 Hz) 0100 : 128 (976 Hz) 0101 : 256 (1.9 kHz) 0110 : 512 (3.9 kHz) 0111 : 1024 (7.8 kHz) others : 2048 (15.6 kHz)

Obrázek 6.10: ABN registr

Příkaz data write, který nastavuje ABN registr má následující obsah 6.2.

Tabulka 6.2: Příkaz data write

1. byte	2. byte	3. byte
0x36	0x09	0x0f

1. byte značí adresu zařízení, 2. byte značí adresu registru, do kterého chceme zapisovat, a 3. byte značí data, která chceme zapsat. Aby bylo toto nastavení uloženo i po odpojení napájení, je nutné zaslat příkaz BurnConf. Příkaz BurnConf spočívá v zapsání hodnoty 0x40 na adresu registru 0xff. Tento příkaz je možné zaslat pouze jednou za život senzoru. Všechny senzory byly nastaveny na nejvyšší přesnost a to 2 048 bodů na otáčku magnetu.

#### 6.4.4 Ochrany ruky

Aby se ruka nepoškodila jsou implementovány dvě ochrany. První ochrana používá Hallovy senzory. Přerušním, které vyvolá Hallův senzor, se uloží hodnota všech Hallových senzorů. Krokový motor rotující se základnou má jeden Hallův senzor. Další dva krokové motory mají každý dva Hallovy senzory. Pokud je nějaký Hallův senzor v logické nule, motor příslušící k tomuto Hallovu senzoru se zastaví. Povolen je pouze ve směru odkud rameno přijelo.

Druhá ochrana je čistě softwarová. Pokud by byl úhel osy s úhlem  $\Theta_2$  větší než je úhel osy  $\Theta_1$ , tyče by se srazily. Aby nedošlo k tomuto střetu, je žádaná poloha motorů regulována tak, aby byly od sebe vždy minimálně 3 stupně. Pokud je žádaná poloha mimo povolenou hodnotu, upraví se tak, aby byly tyče byly od sebe vzdáleny 3 stupně.

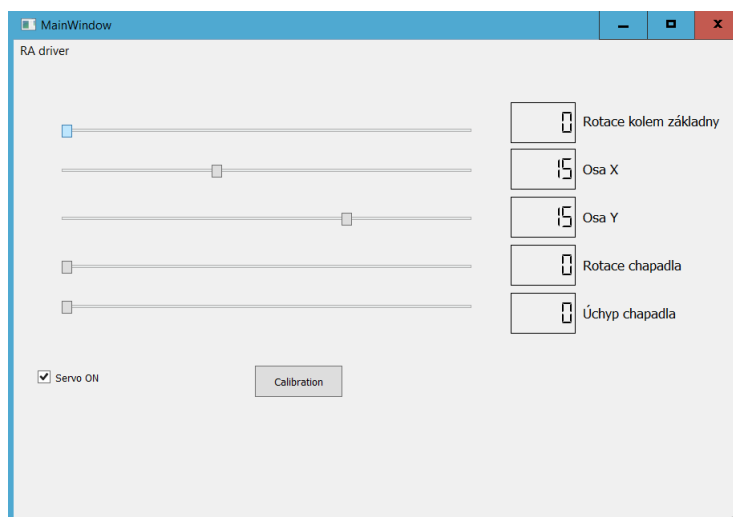
## 6.5 Kalibrace

Po spuštění master desky se spustí kalibrace. Pomalou rychlostí dojedou všechny krokové motory na doraz podle Hallových senzorů a vynulují se čítače, které snímají výstupy ze senzorů úhlu. Po provedení kalibrace se přejde znovu do poslední zvolené polohy. Po zapnutí master desky je počáteční hodnota nastavena jako 0 pro rotaci kolem základy a bod [15,15] pro chapadlo. Kalibraci je možné spustit pomocí tlačítka v aplikaci. Během kalibrace master deska přijímá zprávy. Kalibrace trvá maximálně 10 vteřin a to v závislosti v jakém úhlu jsou osy.

## 6.6 Možnosti řízení ruky

### 6.6.1 Windows aplikace

Za účelem použití robotické ruky bez rovera byla navržena aplikace pro operační systém Windows, která komunikuje s rukou pomocí USB. Aplikace má 5 sliderů pro ovládání ruky. Dále má tlačítko pro kalibraci a tlačítko pro vypnutí/zapnutí napájení servomotorů. Vzhled aplikace je zobrazen na obrázku 6.11.



Obrázek 6.11: Aplikace



Pro zjednodušení ovládání ruky lze měnit hodnoty na sliderech pomocí tlačítek na klávesnici. Výčet kláves je v tabulce 6.3.

**Tabulka 6.3:** Ovládání aplikace

+	-	slider
P	O	rotace kolem základny
D	A	osa X
W	S	osa Y
K	J	rotace chapadla
M	N	chapadlo

Odesílané zprávy odpovídají protokolu v tabulce 6.1. Z důvodu, že Qt aplikace nepošle zprávu při zavolání funkce odesílající data po sériovém portu, ale až při opuštění funkce, bylo nutné implementovat přijímání zpráv o velikosti šesti bytů na straně master desky.

## ■ 6.6.2 CAN

Zprávy z rovera přes sběrnici CAN budou primárně zprávy typu data, které budou obsahovat žádanou polohu. Žádaná poloha bude určena dvěma joysticky na dálkovém ovladači, který řídí rovera. Jeden joystick bude určen pouze pro osu X a Y, druhý joystick bude mít 2 módy. První bude pouze pro rotaci kolem základny a po přepnutí pomocí tlačítka bude možné ovládat oba servomotory.



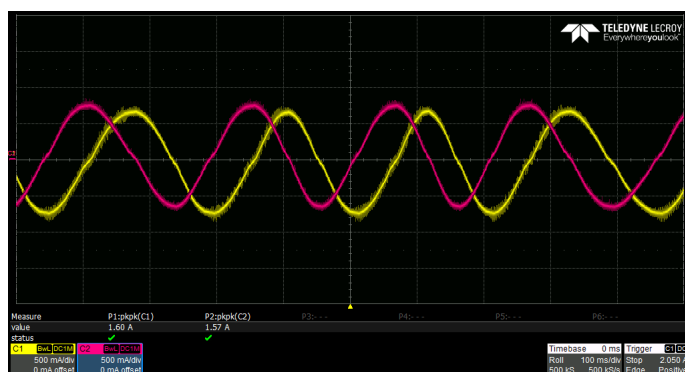
## Kapitola 7

### Testování

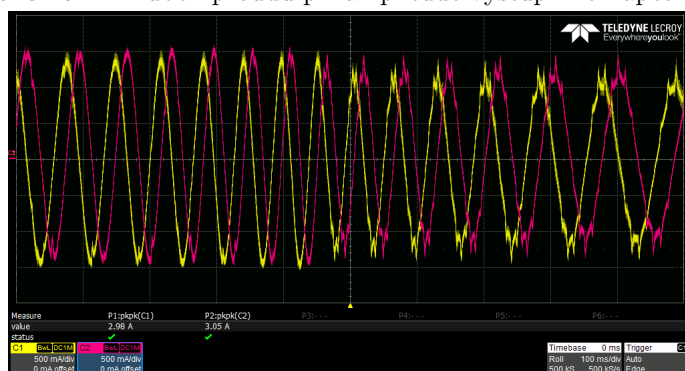
Ruka byla testována pouze pomocí aplikace. Ruka se pohybuje plynule a nevznikají žádné silné vibrace. Ruka je stabilní a i bez přišroubování na rovera se nepřevrátí. Deska ruky při napájení z 12 V má spotřebu 0,4 W, po připojení slave desek spotřeba stoupne na 0,5 W. Po připojení krokových motorů a servomotorů stoupne spotřeba na 5 W. Při testování se objevily dva problémy. Když bylo rameno ve vodorovné pozici nebylo možné zvednout rameno zpět, protože prokluzoval řemen na motoru rotující s velkou tyčí. Byly převrtány díry na uchycení motoru tak, aby byl řemen více napnutý. Touto úpravou byl problém vyřešen.

Druhý problém se týkal nedostatečného výkonu krokového motoru. Při stejné situaci, jako byl první problém, motor pohybující s velkou tyčí ztratil krok a nebylo možné dále s rukou pohybovat. Motory jsou řízeny napětím, nikoliv proudem, proto bylo nutné zvýšit amplitudu výstupů driverů krokových motorů. Tovární nastavení výstupní amplitudy je 16 % z vstupního napětí tedy přibližně 2 V. Při zvětšení amplitudy na 32 %, tedy 4 V, byl problém vyřešen. Bohužel za cenu vyšší spotřeby. Ruka měla spotřebu 17 W, což je pro rovera napájeného z baterie příliš a rover by měl kratší dojezd. Pro snížení spotřeby byla amplituda nastavena následovně. Při nečinnosti motoru je amplituda 16 %, při pohybu 24 % a při akceleraci a deceleraci 32 %. Tímto byl dosažen jak dostatečný výkon motorů, tak úspora energie. Při nečinnosti má ruka spotřebu 6 W a při činnosti 17 W.

Průběh proudu tekoucího do motoru při velikosti amplitudy výstupního napětí 16 % je zobrazen na obrázku. Velikost amplitudy proudu je 1,6 A. Na obrázku je možné vidět průběh proudu tekoucího do motoru při velikosti amplitudy výstupního napětí 32 %. V polovině přechází motor z režimu akcelerace do režimu stálé rychlosti a mění amplitudu výstupního napětí na 24 %. Velikost amplitudy proudu je 1,5 A respektive 1,25 A. Proud byl měřen pomocí proudových kleští Agilent 1146a.



Obrázek 7.1: Průběh proudu při amplitudě výstupního napětí 16 %



Obrázek 7.2: Průběh proudu při amplitudě výstupního napětí 32 a 24 %



## Kapitola 8

### Závěr

Cílem této práce bylo sestavit a zprovoznit robotickou ruku určenou pro rovera. I přes zvolení již hotové mechanické konstrukce bylo potřeba ji v několika ohledech upravit popřípadě opravit. Z tohoto důvodu jsem nemohl věnovat tolik času návrhu řídicí elektroniky či psaní firmwaru. Mechanická konstrukce je stabilní, robustní a zakomponovaná do rovera.

Úspěšně jsem navrhl a osadil desky plošných spojů, které řídí ruku. Zde se nenaskytly žádné větší problémy a nebyly zapotřebí žádné velké úpravy. Desky jsou funkční a všechny komponenty splňují svůj účel, včetně vstupních ochran.

Firmware nahraný na master desce, který jsem navrhl, umožňuje komunikaci s aplikací a dalšími komponenty na deskách. Vypočítává inverzní kinematickou úlohu pro jednoduché řízení ruky a má v sobě implementovány ochrany proti zničení ruky. Implementace komunikace přes sběrnici CAN s roverem nebyla kvůli nastalé situaci možná.

Chapadlo je schopné zvednout pouze malé a nekluzké předměty. Bylo by vhodnější použít chapadlo s větším rozevřením čelistí a také s pogumovanými čelistmi.

Ruka se pohybuje plynule a je schopná manipulovat s předměty. Celá konstrukce je stabilní a robustní. Ovládání je intuitivní. Spotřeba je při nečinnosti 6 W a při pohybu 16 W.





## Příloha A

### Literatura

- [1] AMS, *AS5601*, [https://ams.com/documents/20143/36005/AS5601\\_DS0003953-00.pdf/9a58e74f-f6d8-53eb-1fa2-ad62d9911ca4](https://ams.com/documents/20143/36005/AS5601_DS0003953-00.pdf/9a58e74f-f6d8-53eb-1fa2-ad62d9911ca4).
- [2] Bourns, *MOV-14D180K*, <https://www.bourns.com/docs/Product-Datasheets/MOV14D.pdf>.
- [3] FTDI Chip, *FT230X*, [http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS\\_FT230X.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT230X.pdf).
- [4] Boston Dynamics, *Spot*, [https://shop.bostondynamics.com/spot-arm?cclcl=en\\_US](https://shop.bostondynamics.com/spot-arm?cclcl=en_US).
- [5] Texas Instruments, *DRV5021*, [https://www.ti.com/lit/ds/symlink/drv5021.pdf?ts=1619643206516ref\\_url=https%253A%252F%252Fwww.ti.com%252Fstore%252Fti%252Fen%252Fp%252Fproduct%252F%253F](https://www.ti.com/lit/ds/symlink/drv5021.pdf?ts=1619643206516ref_url=https%253A%252F%252Fwww.ti.com%252Fstore%252Fti%252Fen%252Fp%252Fproduct%252F%253F)
- [6] ———, *LMR33620*, <https://www.ti.com/product/LMR33620>.
- [7] ———, *LMR50410*, LMR50410.
- [8] Giovanni Lerda, *KAUDA*, <https://lerdagiovanni.wixsite.com/kauda>, 2020.
- [9] Littelfuse, *SMBJ16CA*, [https://www.littelfuse.com/media/electronics/datasheets/tvs\\_diodes/littelfuse\\_tv](https://www.littelfuse.com/media/electronics/datasheets/tvs_diodes/littelfuse_tv)
- [10] NXP, *TJA1057*, <https://www.nxp.com/docs/en/data-sheet/TJA1057.pdf>.
- [11] ON Semiconductor, *SZNUP2105L*, <https://www.onsemi.com/pdf/datasheet/nup2105l-d.pdf>.
- [12] STMicroelectronics, *L6470*, <https://www.st.com/resource/en/datasheet/l6470.pdf>.
- [13] ———, *STM32F446VE*, <https://www.st.com/en/microcontrollers-microprocessors/stm32f446ve.html>.

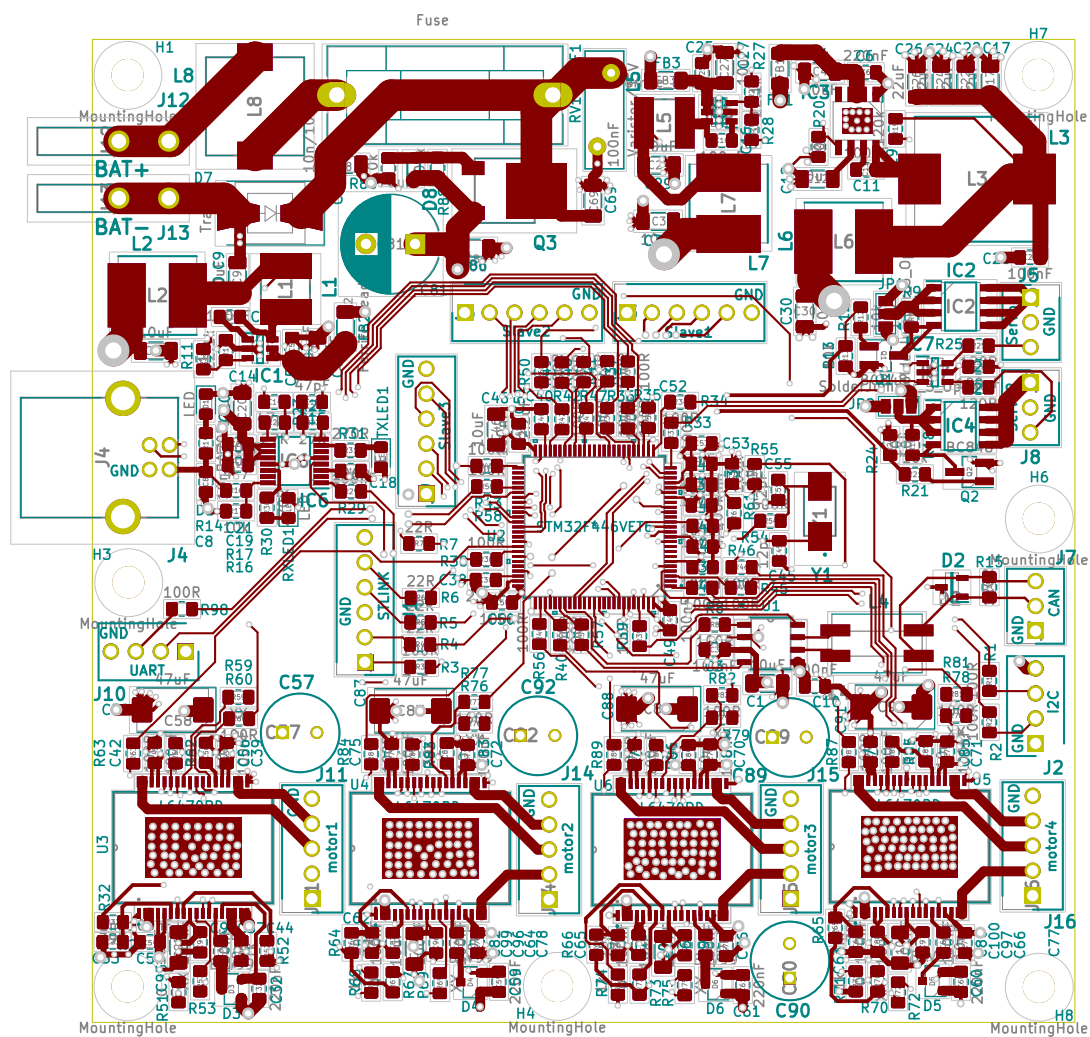
- [14] \_\_\_\_\_, *STMCubeIDE*, <https://www.st.com/en/development-tools/stm32cubeide.html>.
- [15] \_\_\_\_\_, *STMCubeMX*, <https://www.st.com/en/development-tools/stm32cubemx.html>.
- [16] uStepper, *Robotic arm 4*, <https://ustepper.com/store/robotics/34-ustepper-robot-arm-rev-4-full-kit.html>, 2020.
- [17] Vishay, *SQD40031EL*, <https://www.vishay.com/docs/76011/sqd40031el.pdf>.
- [18] WLKATA, *Mirobot*, <https://www.wlkata.com/products/wlkata-mirobot-introduction>, 2020.



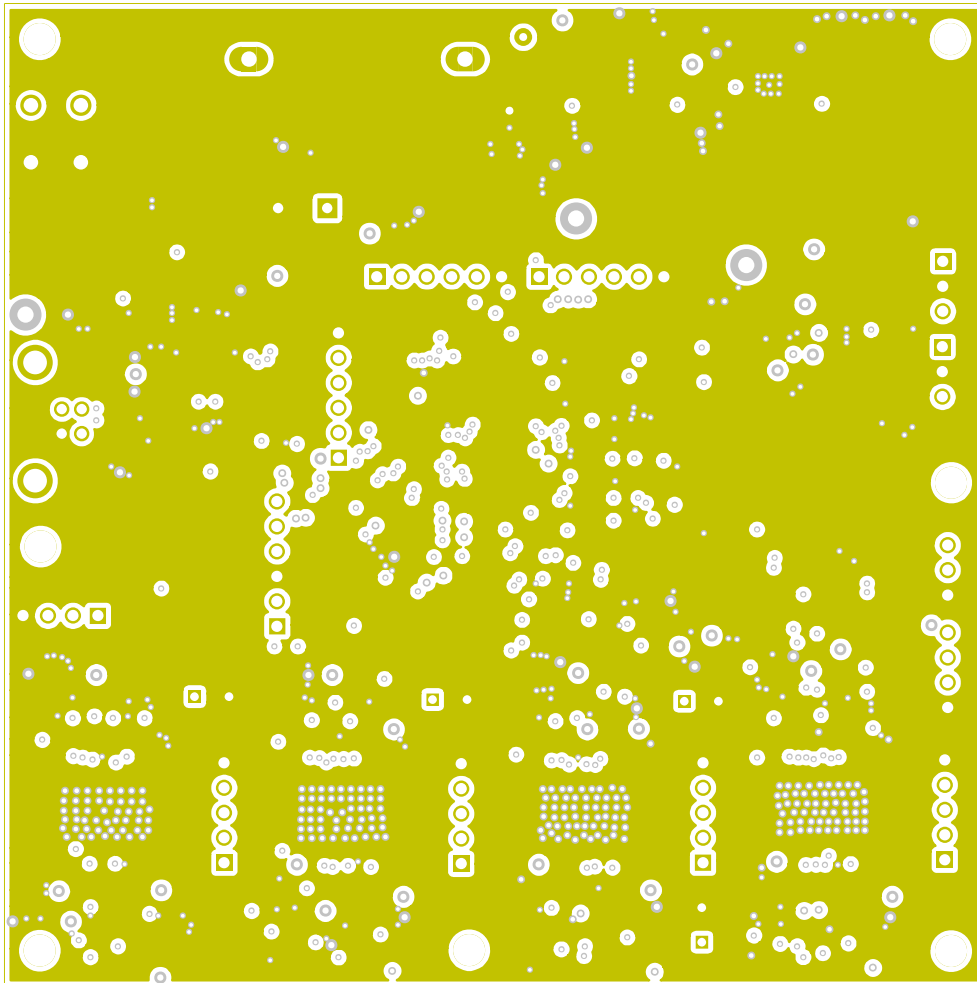


## **Příloha B**

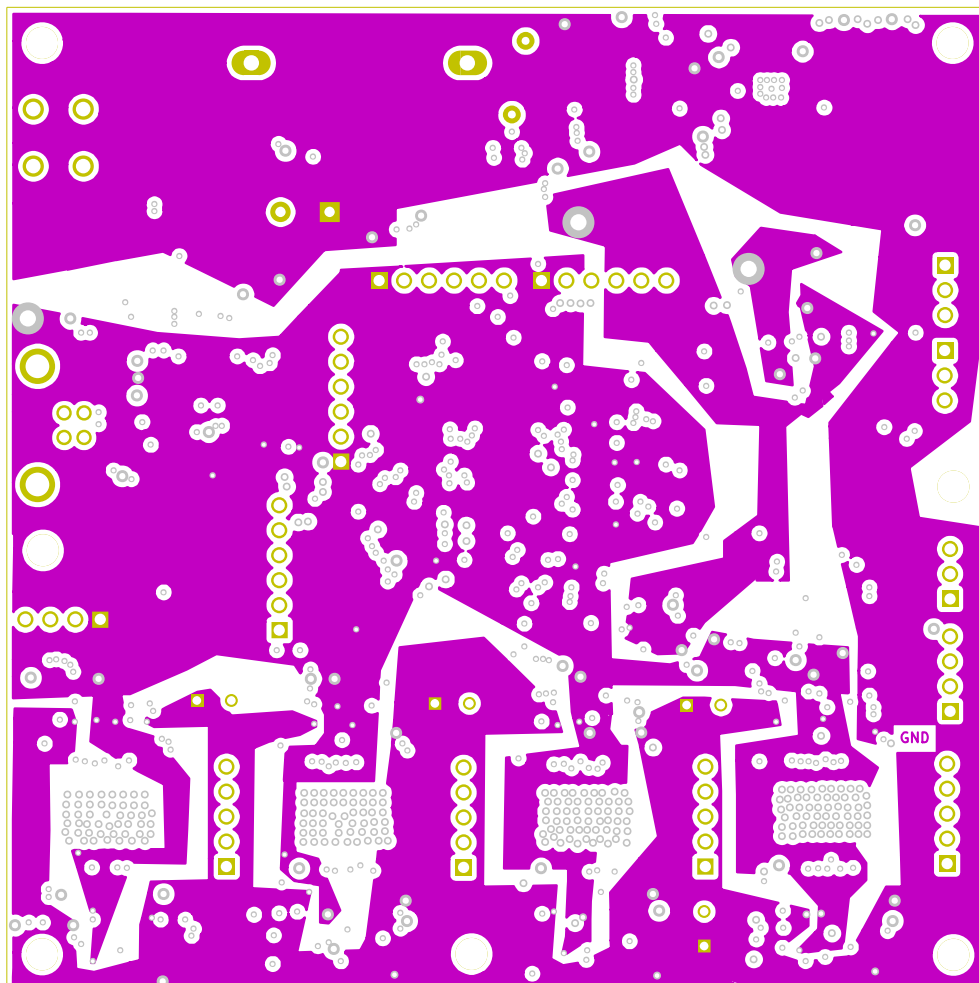
### **Desky plošných spojů**



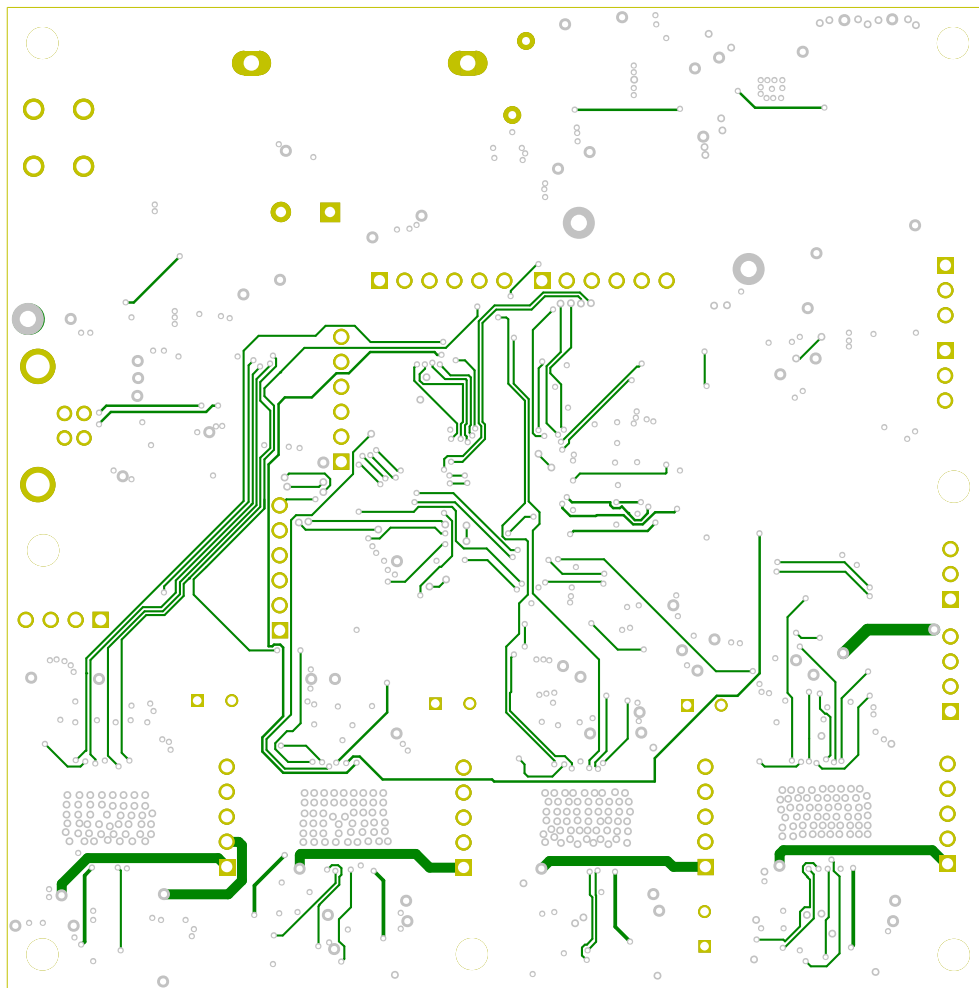
Obrázek B.1: První, signálová vrstva



**Obrázek B.2:** Druhá, zemnicí vrstva



Obrázek B.3: Třetí, napájecí vrstva



Obrázek B.4: Čtvrtá, signálová vrstva