

# Konstrukce a ovládání animatronické ruky

Jakub Trefil

---

Bakalářská práce  
2017



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2016/2017

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub Trefil**  
Osobní číslo: **A14147**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Informační a řídicí technologie**  
Forma studia: **prezenční**

Téma práce: **Konstrukce a ovládání animatronické ruky**  
Téma anglicky: **The Construction and Control of an Animatronic Hand**

Zásady pro vypracování:

1. Provedte návrh animatronické ruky.
2. Vyberte vhodné senzory a další elektronické prvky, které budou součástí rukavice pro ovládání.
3. Navrhněte elektronické prvky pro řízení animatronické ruky.
4. Uvedený návrh hardwarově realizujte.
5. Vytvořte program, který bude zpracovávat data ze senzorů rukavice a vyhodnocovat zásah do animatronické ruky.
6. Realizujte bezdrátovou komunikaci mezi rukavicí a animatronickou rukou.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. BARR, Michael a Anthony J MASSA. Programming embedded systems. 2nd ed. Sebastopol: O'Reilly, 2006, xxi, 301 s. ISBN 978-0-596-00983-0.
2. CATSOULIS, John. Designing Embedded Hardware. Sebastopol: O'Reilly Media, 2005. ISBN 978-0-596-00755-3.
3. MANN, Burkhard. C pro mikrokontroléry: ANSI-C, kompilátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky. Praha: BEN, 2003. ISBN 80-730-0077-6.
4. NOVÁK, Petr. Mobilní roboty - pohony, senzory, řízení, Praha: BEN - technická literatura, 2005. ISBN/EAN 80-7300-141-1 / 9788073001414.
5. PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1.

Vedoucí bakalářské práce: **Ing. Petr Navrátil, Ph.D.**

Ústav řízení procesů

Datum zadání bakalářské práce: **24. února 2017**

Termín odevzdání bakalářské práce: **24. května 2017**

Ve Zlíně dne 24. února 2017



doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynalčeny (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 24. 5. 2017

  
.....  
podpis diplomanta

## **ABSTRAKT**

Cílem bakalářské práce je vytvoření plně funkčního modelu animatronické ruky, která bude dálkově ovládaná pomocí speciální rukavice. Teoretická část práce má za úkol seznámit čtenáře s problematikou animatroniky a jejím spojením s robotikou. Dále jsou zde popsány principy a technologie, které úzce souvisí s praktickou částí. Praktická část obsahuje celý postup tvorby animatronické ruky od navrhování 3D modelů a elektroniky, přes programování, až po fyzickou realizaci projektu.

Klíčová slova: animatronická ruka, Arduino, mikropočítač, servomotor, programování

## **ABSTRACT**

The main goal of this bachelor's thesis is creation of fully functional animatronic hand, which will be controlled by special glove. The theoretical part of thesis is trying to familiarize reader with animatronics and its connection to robotics. It also contains description of principles and technologies that are closely related to the practical part. The practical part contains the entire process of creating animatronic hand from designing 3D models and electronics, through programming, to the physical implementation of the project.

Keywords: animatronic hand, Arduino, microcontroller, servomotor, programming

Rád bych poděkoval vedoucímu mé bakalářské práce panu Ing. Petru Navrátilovi, Ph.D. za poskytnuté konzultace a cenné rady. Mé poděkování patří také rodině a přátelům za podporu a trpělivost, kterou mi poskytovali po celou dobu psaní této práce.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 ANIMATRONIKA</b> .....	<b>11</b>
1.1 ANIMATRONIKA A FILM.....	11
<b>2 SERVOMOTOR</b> .....	<b>13</b>
2.1 PRINCIP ČINNOSTI SERVOMOTORU.....	13
2.2 ŘÍZENÍ SERVOMOTORU.....	14
2.2.1 PWM.....	14
2.2.2 PWM pro servomotor.....	14
<b>3 SENZORY</b> .....	<b>16</b>
3.1 AUTONOMNÍ ANIMATRONICKÉ MODEL Y.....	16
3.2 POUŽÍVANÉ SENZORY V ANIMATRONICE.....	16
3.2.1 Taktilní senzor.....	16
3.2.2 IR detektor překážek.....	17
3.2.3 Sonar.....	18
3.2.4 Bolometr.....	19
3.2.5 Akcelerometr – MEMS senzor.....	19
3.3 OVLADAČE.....	20
<b>4 MIKROPOČÍTAČE</b> .....	<b>21</b>
4.1 PROCESORY TYPU RISC A CISC.....	21
4.2 VSTUPY A VÝSTUPY MIKROPOČÍTAČE.....	22
4.2.1 Digitální vstupy a výstupy.....	22
4.2.2 Analogové vstupy.....	22
4.2.2.1 A/D převodník.....	22
4.2.3 Analogové výstupy.....	23
4.3 I2C KOMUNIKACE.....	24
4.3.1 Princip funkce.....	24
4.4 ARDUINO.....	25
<b>5 POUŽITÉ PROGRAMY</b> .....	<b>26</b>
5.1 AUTODESK INVENTOR.....	26
5.2 SLIC3R.....	26
5.3 FRITZING.....	26
5.4 ARDUINO IDE.....	26
<b>II PRAKTICKÁ ČÁST</b> .....	<b>27</b>
<b>6 NÁVRHANIMATRONICKÉ RUKY</b> .....	<b>28</b>
6.1 VÝCHOZÍ PŘEDSTAVA.....	28
6.2 NÁVRH PRSTŮ.....	30
6.3 NÁVRH DLANĚ A ZÁPĚSTÍ.....	32
6.4 NÁVRH STATICKÉ ZÁKLADNY.....	33
6.5 NÁVRH PROSTORU PRO ELEKTRONIKU, NAPÁJENÍ A CELKOVÝ MODEL.....	34
<b>7 NÁVRH ELEKTRONIKY</b> .....	<b>36</b>

7.1	ELEKTRONICKÉ SOUČÁSTI ROBOTICKÉ RUKY .....	36
7.1.1	Vývojová platforma .....	36
7.1.2	Servopohony .....	37
7.1.3	Napájení .....	39
7.1.4	Nepájivé kontaktní pole .....	40
7.1.5	Schéma zapojení.....	41
7.2	ELEKTRONICKÉ SOUČÁSTI RUKAVICE PRO OVLÁDÁNÍ .....	42
7.2.1	Vývojová platforma .....	42
7.2.2	Senzory pro snímání ohybu prstů.....	42
7.2.3	Senzor pro snímání rotace zápěstí.....	43
7.2.4	Ostatní součástky .....	44
7.2.5	Schéma zapojení.....	45
<b>8</b>	<b>BEZDRÁTOVÁ KOMUNIKACE .....</b>	<b>47</b>
8.1	VYSÍLAČ.....	47
8.1.1	Anténa .....	48
8.1.2	Schéma propojení vysílače s Arduinem .....	48
8.2	PŘIJÍMAČ .....	48
8.2.1	Schéma propojení přijímače s Arduinem .....	49
<b>9</b>	<b>PROGRAM.....</b>	<b>50</b>
9.1	KÓD PROGRAMU RUKAVICE.....	50
9.2	KÓD PROGRAMU ANIMATRONICKÉ RUKY .....	53
<b>10</b>	<b>FYZICKÁ REALIZACE PROJEKTU .....</b>	<b>54</b>
10.1	3D TISK.....	54
10.1.1	Tiskárna.....	54
10.1.2	Materiál .....	55
10.2	ANIMATRONICKÁ RUKA .....	55
10.3	RUKAVICE PRO OVLÁDÁNÍ.....	58
10.3.1	Kalibrace .....	59
	<b>ZÁVĚR .....</b>	<b>61</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>62</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>64</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>65</b>
	<b>SEZNAM TABULEK.....</b>	<b>67</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>68</b>



## ÚVOD

Cílem bakalářské práce je zhotovení animatronické ruky a popsání celého výrobního procesu. Bakalářskou práci na dané téma jsem si zvolil z toho důvodu, že bych v budoucnu chtěl vytvořit větší projekt, který bude z této bakalářské práce dále čerpat. Jedná se o projekt vytvoření protetické ruky. Tato práce je příležitostí k seznámení se s různými vývojovými platformami nebo senzorickými systémy a k načerpání zkušeností pro podobné projekty. Celá práce je rozdělena do dvou částí.

Teoretická část se skládá z pěti kapitol. V první z nich je čtenář seznámen s pojmem animatronika. Druhá kapitola popisuje využití servomotoru u podobných úloh. Je zde popsáno fungování servomotoru a způsob jeho řízení. Kapitola 3 je věnována senzorům, které nacházejí využití v animatronice. V této kapitole je také objasněn princip autonomního modelu. Čtvrtá kapitola se zabývá pojmem mikropočítač. Obsahuje základní informace o mikropočítačích. Jsou zde popsány vstupy/výstupy mikropočítače a komunikace I2C, která je využita v praktické části při komunikaci s perifériemi. Poslední kapitola teoretické části zmiňuje programy, které byly zapotřebí k vytvoření celé animatronické ruky.

Praktická část je rozdělena do pěti kapitol a každá z nich popisuje úsek výrobního procesu. Kapitola 6 je zaměřena na návrh animatronické ruky pomocí 3D modelovacího softwaru. Nachází se zde také původní vize, podle které byl model vytvářen. V kapitole 7 jsou vybírány vhodné elektronické prvky jak pro animatronickou ruku, tak pro speciální rukavici pro ovládání. Kapitola také ukazuje zapojení jednotlivých elektronických prvků pomocí elektro schémat. Osmá kapitola je zaměřena na problematiku bezdrátového přenosu. Jsou zde vybrány vhodné moduly pro komunikaci a jejich připojení k mikropočítači. Kapitola 9 začíná obecným algoritmem pro rozpohybování animatronické ruky a je věnována programovému vybavení. Nachází se zde popis nejdůležitějších částí kódu. V poslední kapitole se dostáváme k fyzické realizaci animatronické ruky i rukavice pro ovládání. Vidíme zde využití 3D tisku a dalších úprav ke zhotovení celého modelu. Poslední kapitola také obsahuje fotografie výsledného provedení.

## **I. TEORETICKÁ ČÁST**

## 1 ANIMATRONIKA

Animatronika se zabývá využitím robotiky k simulaci života u neživých věcí. V jednoduchých případech se jedná pouze o rozpohybování předmětu. Nejsložitější animatronické modely mohou být autonomní. Jejich chování je tedy spontánní a mění se v závislosti na okolních podmínkách. K vytvoření autonomního modelu se využívá řady senzorů, akčních členů a složité výpočetní programy. Animatronika vychází z oborů jako je anatomie, mechatronika (informatika, elektronika, mechanika) a modelářství. Může sloužit k nahrazení živých osob nebo zvířat v případech, kdy by mohlo dojít k vážnému zranění. Velkou oblastí animatroniky je napodobení vyhynulých nebo neexistujících druhů zvířat či rostlin – prehistorická nebo mytologická. Hlavní využití nachází animatronika ve filmu a zábavním průmyslu (např. zábavní parky, hračky). Pro tvorbu velkých a těžkých modelů se využívají hydraulické pohony. Naopak u malých modelů nebo hraček se využívá výhradně elektrických pohonů. [1] [3]

### 1.1 Animatronika a film

Animatronické modely jsou známy hlavně z filmů Stevena Spielberga - Čelisti, E.T. – Mímozemšťan nebo Jurský Park. Největší vytvořený model se objevil ve filmu Jurský Park 3. Jedná se o 13,3 m dlouhý model vážící 11 tun. Veškeré pohyby byly zajištěny hydraulickými prvky. Model fungoval i za deště a na vývoji se podílelo 30 lidí. K ovládní bylo zapotřebí 8 operátorů, každý z nich dálkově ovládal část modelu:

- Pohyb těla a hlavy
- Pohyb jazyku
- Pohyb očí a víček
- Pohyb předních končetin
- Posun modelu na připravené trati dopředu a dozadu
- Ovládní dechu
- Pohyb ocasu
- Zvyšování a snižování těžiště těla [2]



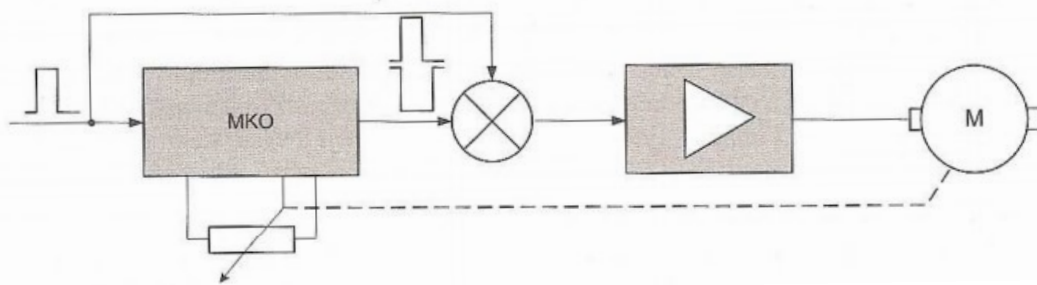
*Obrázek 1 – Rozpracovaný model Spinosaura pro film Jurský Park 3 [2]*

## 2 SERVOMOTOR

V animatronice se často vyskytují elektromechanické akční členy. Jejich úkolem je upravit polohu mechanické části podle potřeby. Úprava polohy může být uskutečněna pomocí systému táhel, kladek, ozubených kol nebo je v některých případech spojen akční člen přímo k mechanické konstrukci. K účelům změny polohy slouží stejnosměrné motory, krokové motory nebo servomotory. Stejnosměrné motory ani krokové motory však nemají údaj o své poloze. Musí se proto využívat dalších senzorů, kterými požadovaný údaj získáme (např. enkodéry). V tomto směru se jeví jako výhodné použití servopohonů, u kterých lze nastavit přesnou polohu natočení pomocí řídicího signálu.

### 2.1 Princip činnosti servomotoru

Servopohon se skládá z několika částí. První z nich je malý stejnosměrný motor. Druhou část tvoří potenciometr a poslední částí je řídicí obvod. Stejnosměrný motor je přes převody (několik ozubených kol) spojen s potenciometrem. Rotace stejnosměrného motoru způsobuje změnu odporu potenciometru. Vstupní řídicí signál spouští monostabilní klopný obvod, ten generuje impuls odpovídající momentální poloze potenciometru (serva) s opačnou polaritou než vstupní signál. Oba impulzy se odečtou a výsledkem je řídicí signál, který je zesílen a ovládá rotaci stejnosměrného motoru. Pokud mají oba impulzy stejnou délku, řídicí signál bude mít nulovou hodnotu, což odpovídá dosažení požadované hodnoty natočení. Akce se velmi rychle opakuje s periodou 20 ms. [4] [6]



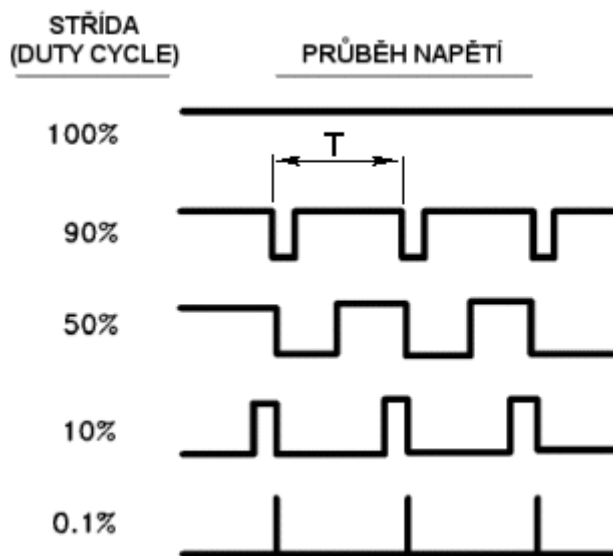
Obrázek 2 – Blokové schéma vnitřní elektroniky modelářského serva [6]

Servopohony umožňují rotaci v rozmezích 180°, 270° nebo dokonce 360°. Můžou být mechanicky upraveny na kontinuální rotaci – odstranění mechanického dorazu a změna potenciometru za odporový dělič. [6]

## 2.2 Řízení servomotoru

### 2.2.1 PWM

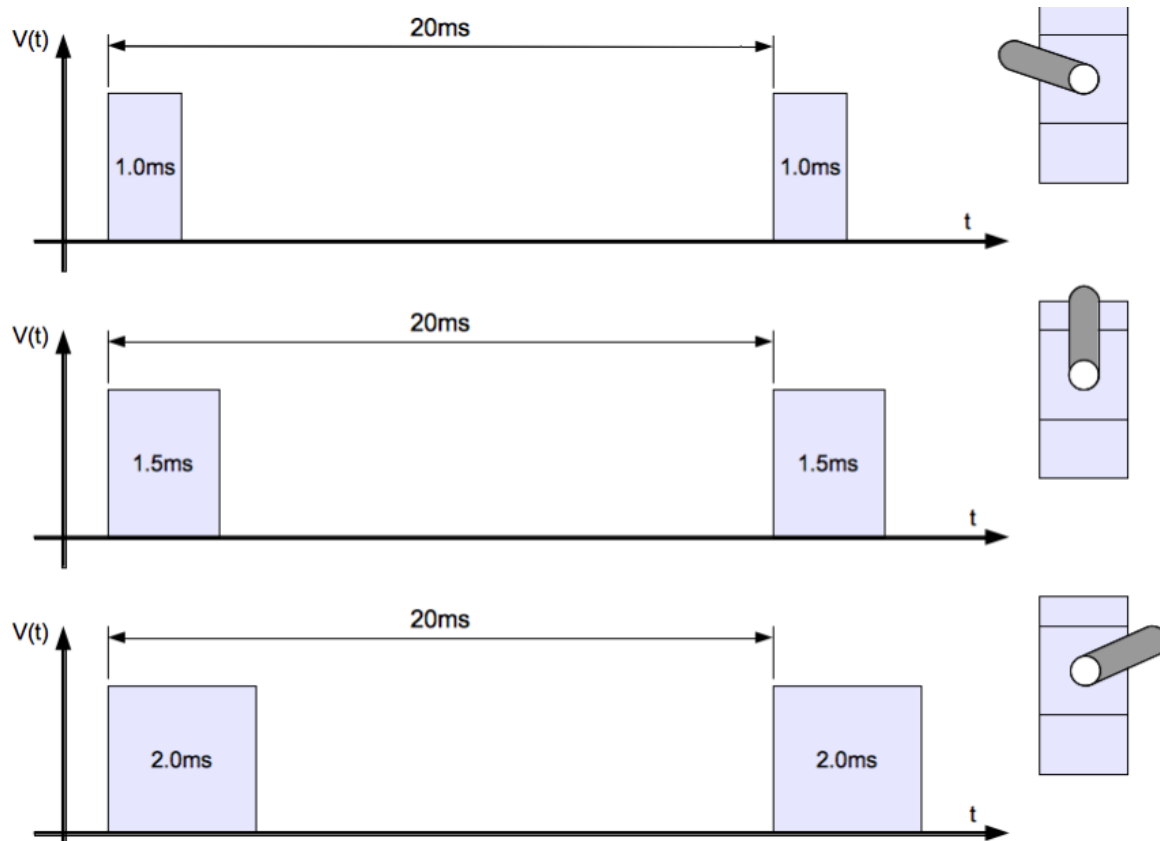
Řízení servomotoru se provádí pomocí PWM, neboli pulzně-šířkové modulace (Pulse-Width Modulation). Jedná se o signál, který střídá hodnotu log. 0 a log. 1 (0 a 100%). Představuje zapnutí/vypnutí zdroje napětí. Poměr mezi délkou pulzu na úrovni 1 a délkou pulzu na úrovni 0 se nazývá střída. Výkon dodávaný do zátěže je dán střídou. Pokud se signál nachází celou periodu ve stavu log. 1, poté je střída rovna 100 % a znamená 100 % výkon do zátěže. Pokud je signál nachází polovinu času periody v log. 1 a zbytek log. 0, střída je 50 %. PWM je ukázáno na následujícím obrázku. [9]



Obrázek 3 – Princip pulzně-šířkové modulace [9]

### 2.2.2 PWM pro servomotor

Při řízení servomotoru se generuje pulzní signál s periodou 20 ms a proměnnou šířkou pulzu 1-2 ms. Střída takového signálu se pohybuje mezi 5 % až 10%. Pro pulzy o délce 1 ms se servomotor nachází v počáteční poloze ( $0^\circ$  natočení). Pro délku pulzu 1,5 ms se nachází servo ve středové poloze ( $90^\circ$  natočení). Pro délku pulzu 2 ms je servo v koncové poloze ( $180^\circ$  natočení). Pro různé druhy servomotorů se délky řídicího signálu mohou lišit. Ke generování podobných signálů lze jednoduše využít mikropočítače. [6]



Obrázek 4 – PWM pro řízení servomotoru [4]

### 3 SENZORY

Senzory jsou důležitou součástí každého animatronického nebo robotického modelu. V robotice zastupují senzory lidské smysly. Čím lépe je senzorický systém navrhnout a čím více údajů dokáže snímat, tím více se zvyšuje schopnost robota reagovat na okolní prostředí a přizpůsobit se mu. Senzory umožňují získávat informace nejen o okolí robota, ale také o samotném robotu. Z tohoto pohledu se dělí na externí a interní. Interní senzory poskytují informaci o teplotě, poloze mechanických částí, rovnováze nebo stavu baterií robota. Externí senzory slouží pro získávání informací o blízkých objektech, složení okolního vzduchu, tepelné vyzařování z prostředí apod.

#### 3.1 Autonomní animatronické modely

Pokud model obsahuje dostatečné množství senzorů a vhodný řídicí program, který je schopen s měřenými daty pracovat, může se model pohybovat a pracovat samostatně. U takových modelů není potřeba vnějšího řízení člověkem. Jedná se o modely s pevně daným programem nebo o samoučící se modely s prvky umělé inteligence.

#### 3.2 Používané senzory v animatronicce

##### 3.2.1 Taktilní senzor

Jedná se o prvek, který je schopen snímat informaci o dotyku s prvkem z vnějšího prostředí a převádět ji na elektrický signál. Bývají uskupeny do matic nebo podobných útvarů a vytvářejí taktilní plochu. Slouží k ochraně robota, uchopení předmětu s požadovanou silou nebo měření souřadnic předmětu v prostoru.

Výhody:

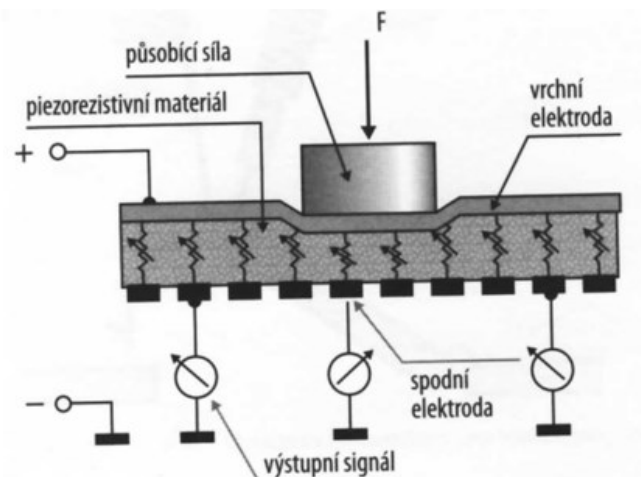
- Malé rozměry, nízká hmotnost
- Nezávislost na okolních vlivech v porovnání s jinými senzory
- Levná realizace
- Možnost práce v reálném čase

Nevýhody:

- Omezená rozlišovací schopnost
- Opotřebení



Nejjednodušším taktilním senzorem jsou kontaktní snímače (tlačítka). Fungují na změně logické úrovně při sepnutí/rozepnutí obvodu při kontaktu s překážkou. Další provedení využívají např. piezoelektrického jevu – působením síly na piezokrystal vzniká na elektrodách náboj, elastomery – mění svůj odpor v závislosti na působící síle, tenzometrický jev – změna délky vodiče vyvolá změnu jeho odporu. [6] [10]

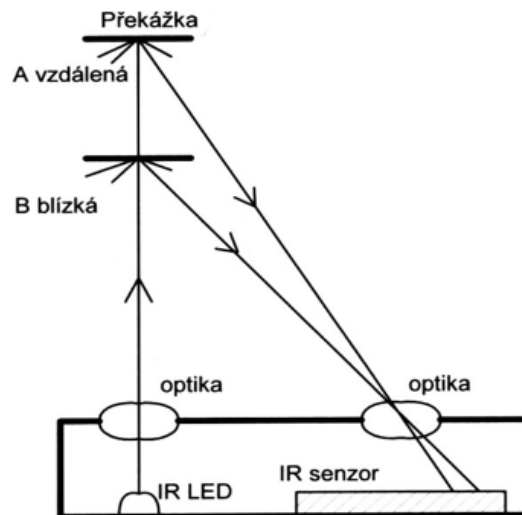


Obrázek 5 – Taktilní senzor s využitím piezoelektrického jevu [6]

### 3.2.2 IR detektor překážek

Slouží hlavně k detekci překážek v blízkosti robotu. V závislosti na prostředí detekuje překážku ve vzdálenosti desítek centimetrů. Skládá se ze dvou částí. První část je LED dioda, která vysílá do okolí infračervené záření (vlnová délka kolem 800-1000 nm). Druhá část je tvořena fototranzistorem citlivým na vyzařované infračervené záření, který přijímá světlo odražené od překážky.

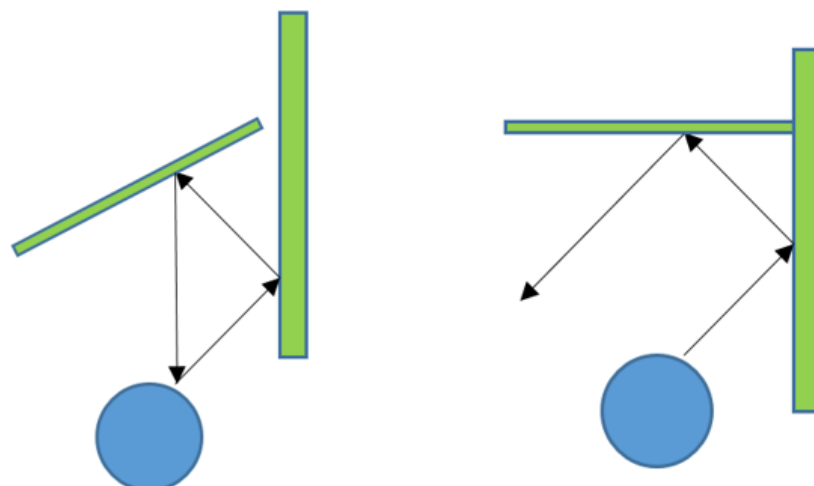
IR senzory poskytují dva způsoby detekce. Jednodušším z nich je detekce překážky do určité vzdálenosti. Senzor nám vrací hodnotu log. 1 pokud se překážka nachází do určité vzdálenosti, neposkytuje však přesnou vzdálenost. Složitější IR senzory jsou schopny určit přesnou vzdálenost od 10 do 80 cm. Jeden ze způsobů detekce je měření velikosti úhlu dopadu infračerveného záření. Vyslaný paprsek se odráží od překážky a přes optiku dopadá na fototranzistor (může být i fotodiody nebo jiný fotocitlivý prvek). Úhel dopadu je závislý na vzdálenosti překážky. [6] [10]



Obrázek 6 – Princip funkce IR detektoru [10]

### 3.2.3 Sonar

Sonar slouží k detekci překážek na větší vzdálenosti než u IR detektoru. Princip spočívá ve vyslání zvukové vlny (ultrazvuk) a následném přijímání odrazů od překážek (echa). Vzdálenost překážky je poté určena pomocí rozdílu času mezi vyslaným a přijatým signálem při znalosti rychlosti šíření zvuku v daném prostředí. Sonar se může potýkat s nepřesnostmi ve formě falešných ech – echo z předchozích vyslaných impulzů. Měření je ovlivněno vlhkostí vzduchu nebo teplotou. [10]



Obrázek 7 – Příklad chybného měření pomocí sonaru, vlevo nepřesná naměřená vzdálenost, vpravo nedetekuje sonar překážku [10]

### 3.2.4 Bolometr

Senzor pro bezdotykové měření teploty. Každý předmět vyzařuje infračervené záření, které může být detekováno. Na základě intenzity infračerveného záření lze určit teplotu předmětu. Bolometr se skládá z několika částí. První z nich je absorpční materiál, na který dopadá přes optiku infračervené záření a ohřívá jej. Druhá část je tvořena odporovým meandrem, který při změně teploty mění svůj odpor. Teplota je poté vyhodnocena na základě změny odporu.

Výhody:

- Rychlé, možnost práce v reálném čase
- Velká teplotní citlivost
- Vysoká spektrální citlivost
- Lze měřit teploty nad 1500 °C
- Malý rozměr
- Neovlivňuje měřený objekt

Nevýhody:

- Vysoká cena
- V některých případech nutné chlazení

Bolometry lze uspořádat do matice a získat obraz s rozlišením 320x240 pixelů. Takto uspořádané matice poté slouží jako termovize nebo detekce výbuchu a ohně. [10]

### 3.2.5 Akcelerometr – MEMS senzor

Senzor pro snímání naklápění, zrychlení, vibrací nebo mechanických rázů. Jeden z hlavních senzorů pro udržování stability. Princip senzoru je založen na měnící se vzdálenosti elektrod kondenzátoru v závislosti na působícím zrychlení. Vzdálenost elektrod má poté vliv na kapacitu C. Jedná se o tříelektrodový vzduchový kondenzátor. Prostřední elektroda je pohyblivá (pohybuje se v závislosti na zrychlení). MEMS (mikro-elektro-mechanické systémy) integrují více elektroniky na křemíkovém substrátu. Obsahuje integrované obvody, senzory a další mechanické prvky nebo akční členy. Elektronika zpracovává informace ze senzorů, a buď je posílá na další vyhodnocování, nebo je ihned zpracovává a dává pokyny akčním členům na čipu. [10]

### 3.3 Ovladače

K ovládání animatronických modelů se často využívá řízení pomocí operátorů (někdy označováni jako loutkaři). Operátor je člověk, který má za úkol věrohodně napodobit chování modelu za pomoci ovladače nebo jiného ovládacího systému (např. celý oblek, ovládací rukavice). Ovladače jsou mechanicky navrženy tak, aby operátorovi co nejvíce ulehčovaly práci s animatronickým modelem. Může se jednat o zařízení, které operátor drží v ruce a pomocí pák ovládá model (podobné jako ovladač k herní konzoli). Nejpropracovanější ovladače jsou zhotoveny jako celé obleky, které jsou osazeny mnoha senzory. Pro snímání pohybu operátora slouží v mnoha případech obyčejné potenciometry (otočné, lineární) zapojené jako děliče napětí. Ke konstrukci ovladačů lze použít i senzory z předchozí kapitoly jako jsou taktilní senzory nebo akcelerometry.

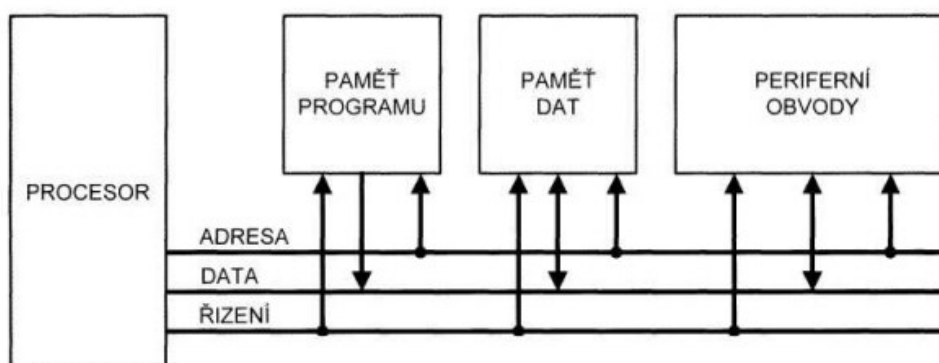


*Obrázek 8 – Operátor s ovladačem pro řízení animatronického modelu [2]*

## 4 MIKROPOČÍTAČE

Mikropočítač by se dal označit za „spojku“ celého animatronického modelu z hlediska elektroniky. Na jedné straně se mikropočítač stará o zpracování dat, které poskytují senzory, a na straně druhé dává pokyny pro akční členy. Všechny informace jsou vyhodnocovány pomocí řídicího programu.

Mikropočítačem se rozumí procesor, paměť a další vstupní/výstupní obvody obsažené na jediném čipu. K propojení všech jednotek slouží sběrnice. Její činnost je řízena převážně procesorem. Adresová sběrnice slouží k rozlišení jednotlivých zařízení připojených na sběrnici a k rozlišení jednotlivých míst jejich paměti. Šířka adresové sběrnice určuje maximální počet možných adres. Datová sběrnice slouží k přenosu dat. Šířka datové sběrnice je zpravidla násobek osmi. Veškerá komunikace je poté zprostředkována pomocí řídicích signálů, které jsou generovány procesorem. [5] [7]



Obrázek 9 – Základní schéma mikropočítače [5]

Většina dnešních mikropočítačů je postavena na Von Neumannově architektuře. Jedná se o počítač s jedním adresovým prostorem, kde existují jen 2 základní sběrnicové cykly – čtení a zápis. Paměť je zde společná jak pro instrukce, tak pro data.

### 4.1 Procesory typu RISC a CISC

CISC (Complex Instruction Set Computing) je počítač se složitým souborem instrukcí. Jedná se o procesory, které poskytují velké množství složitých instrukcí. V dobách Assembleru usnadňovaly programátorům práci. Je zde snaha o minimalizaci počtu instrukcí v programu a zmenšení množství paměti, které zabere. Procesor ovšem tyto složité instrukce zpracovává delší dobu. Naopak RISC (Reduced Instruction Set Computing) procesory dávají přednost menšímu množství základních instrukcí, které je

procesor schopen zpracovat velmi rychle a které jsou často používané. Složitější instrukce se tedy rozkládají na jednodušší, které je procesor schopen vykonat.

## 4.2 Vstupy a výstupy mikropočítače

### 4.2.1 Digitální vstupy a výstupy

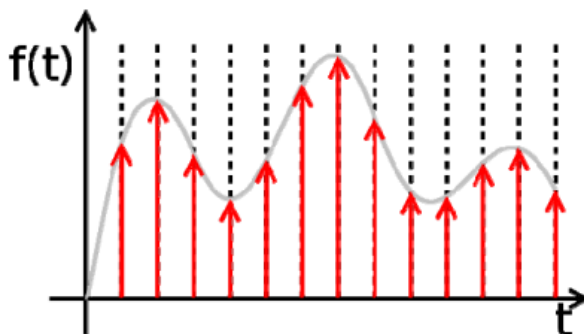
Označovány také jako GPIO – General Purpose Inputs Outputs. K těmto vstupům lze připojit libovolné zařízení, které lze ovládat změnou napěťové úrovně. Digitální vstupy a výstupy nabývají pouze dvou hodnot log. 0 a 1. Jako vstup zde mohou sloužit např. tlačítka (stav vypnuto/zapnuto). Pomocí digitálních výstupů pak můžeme řídit např. signalizaci LED diodou. Digitální výstupy jsou omezeny proudem, který jsou schopny dodat do zátěže. Maximální proud se pohybuje kolem 25 mA.

### 4.2.2 Analogové vstupy

Na rozdíl od digitálních vstupů jsou analogové vstupy schopny rozlišovat velké množství hodnot. Na vstupním analogovém pinu mikropočítače se může objevit napětí spojitě od 0-5 V. Výpočetní technika ale nemá možnost pracovat se spojitým signálem. Proto se používají A/D převodníky, které vstupní spojitý signál převedou na digitální (nespojitý v hodnotách a čase). Mikropočítače obsahují jeden A/D převodník a několik analogových vstupů, které jsou na A/D převodník přivedeny přes multiplexer. Výběr pinu pro převod hodnoty se provádí programově. Pokud by měl každý pin vlastní A/D převodník, celá konstrukce mikropočítače by se výrazně prodražila.

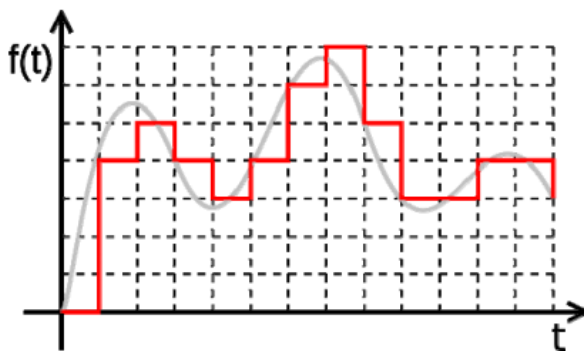
#### 4.2.2.1 A/D převodník

Převod analogového na digitální signál probíhá ve dvou základních krocích. Prvním z nich je vzorkování. Vzorkování slouží pro diskretizaci signálu v čase. Probíhá periodické měření v čase, kde získáme posloupnost hodnot v určitých časových okamžicích. Signál ovšem pořád nabývá nekonečně mnoha hodnot.



Obrázek 10 – Princip vzorkování [11]

Proto je potřeba diskretizovat signál i ve funkčních hodnotách. K tomu slouží kvantování. Kvantování přiřazuje každou navzorkovanou hodnotu do jedné z možných úrovní, které A/D převodník umí rozlišit. Pro A/D převodník s rozlišením 10 bitů se jedná o rozdělení vstupního signálu 0-5 V do 1024 ( $2^{10}$ ) hodnot.



Obrázek 11 – Signál po vzorkování a kvantování [11]

#### 4.2.3 Analogové výstupy

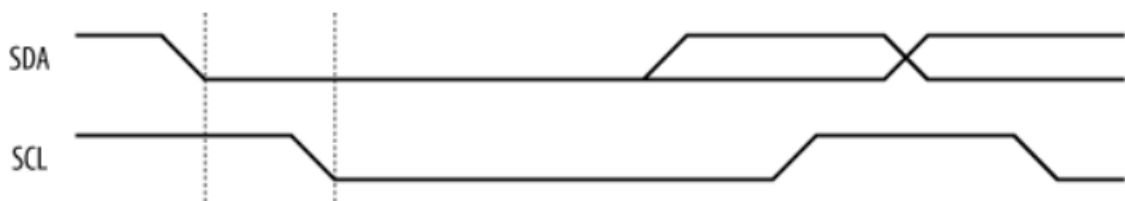
Jsou u mikropočítačů realizovány pomocí D/A převodníků. Ten vytváří spojitý signál z digitálního pomocí tvarovače. Mikropočítače bývají D/A převodníky vybaveny jen zřídka. Částečné zastoupení analogového výstupu pak tvoří piny s možností PWM. Pomocí PWM lze ovládat výkon do zátěže. Lze tak řídit výkon topného tělesa nebo otáčky stejnosměrného motoru.

### 4.3 I2C komunikace

Jedná se o způsob připojení a komunikace mezi mikropočítačem a periferiemi. Využívá se zde dvou vodičů. Jedním z nich je SCL – synchronizační, druhým je SDA – datový. U obou vodičů je možný dvousměrný přenos. Ke sběrnici může být připojeno více zařízení. Zařízení lze libovolně odpojovat a připojovat bez omezení funkčnosti ostatních členů komunikace. Zařízení jsou vybírána podle adresy (každé má unikátní), která je součástí zprávy (první část zprávy). Každé zařízení může pracovat jako vysílač (master) nebo přijímač (slave). [7] [8]

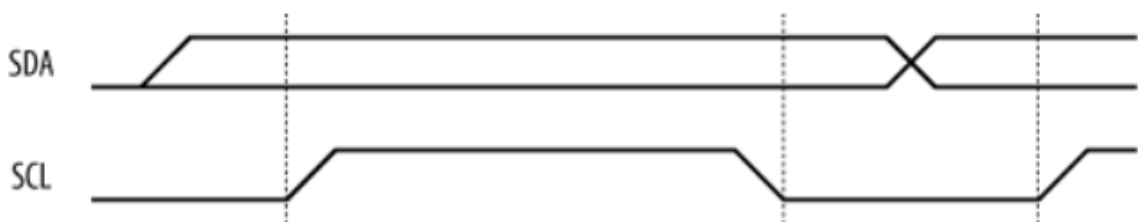
#### 4.3.1 Princip funkce

Pokud se nepřenáší žádná data, SDA i SCL jsou ve stavu HIGH (log. 1). Start komunikace je dán přechodem SDA na log. 0 a v zápětí přechodem SCL na log. 0.



Obrázek 12 – Začátek komunikace I2C [7]

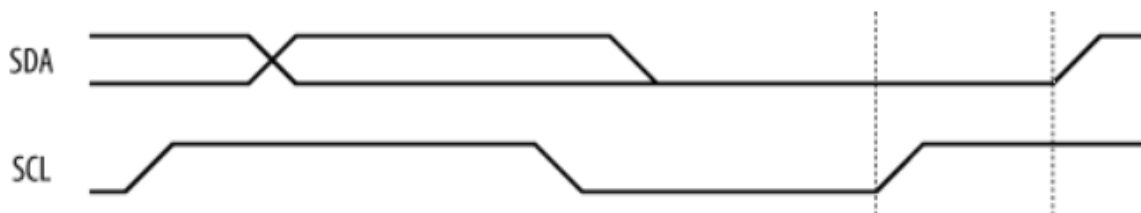
Každý přenesený bit je z vodiče SDA vyčten s náběžnou hranou SCL. SDA musí zůstat na stejné logické úrovni, dokud SCL nepřejde na log. 0. Poté se změní SDA pro další bit a akce se opakuje. Zpráva je členěna na osmibitové slabiky, ale jejich počet je libovolný.



Obrázek 13 – Vyčtení bitu z vodiče SDA pomocí náběžné hrany SCL [7]

Přenos je ukončen při změně SCL na log. 1 a poté SDA na log. 1. Každý odeslaný byte je potvrzován přijímačem.





Obrázek 14 – Ukončení komunikace I2C [7]

#### 4.4 Arduino

Jedná se o open-source platformu. Firma poskytuje zdrojové kódy i plošné spoje v případě zájmu vlastní výroby. Původně vzniklo jako levné řešení pro studenty technologií. Program je psán v jazyce Arduino, které je založeno na jazyce Wiring. Programování je podobné C/C++ a probíhá ve vývojovém prostředí Arduino IDE. Pro práci s Arduinem existuje celá řada modulů, nazývaných Shiedly, které rozšiřují schopnosti samotné vývojové desky. Může se jednat o Wifi shield (umožňuje připojení k internetu), motor shield (ovládání stejnosměrných, krokových motorů) nebo GPS shield (zjišťování údajů o poloze). Arduino je založeno na mikrokontrolerech ATmega. [12] [13]

Tabulka 1 – Druhy Arduino desek a jejich parametry, převzato z [13]

Arduino deska	Mikrokontrolér	Flash paměť KiB	Digitální I/O piny	PWM	Analogové vstupy	Analogové výstupy
Uno R3	ATmega328P	32	14	6	6-8	0
Nano	(ATmega168 před R3) / ATmega328	16/32	14	6	8	0
Mega2560	ATmega2560	256	54	16	16	0
Arduino Due	AT91SAM3X8E	512	54	12	12	2
LilyPad Arduino	ATmega168V / ATmega328V	16	14	6	6	0

## 5 POUŽITÉ PROGRAMY

### 5.1 Autodesk Inventor

Program pro vytváření 3D modelů, výkresové dokumentace nebo animací v prostředí Windows. Základem konstrukcí jsou součásti (parts), jejichž geometrie je odvozena od 2D náčrtů. Součásti mohou být spojovány pomocí různých vazeb do sestav. Při změnách kót se aktualizuje celá sestava i výkresová dokumentace.

### 5.2 Slic3r

Program pro konverzi 3D modelu na posloupnost instrukcí 3D tiskárny. Funkce spočívá ve vodorovných řezech modelem a zjišťování rozměrů 3D modelu v daném řezu. Program umožňuje generovat gcode, disponuje opravou vadných 3D modelů. Dále umožňuje rozdělení modelu na více samostatných dílů. Umožňuje nastavit výšku vrstvy, procenta výplně, způsoby vyplňování, automaticky generuje podpory při tisku a další. Program je zdarma ke stažení.

### 5.3 Fritzing

Jedná se o open source program. Používá se pro navrhování elektrických obvodů. Umožňuje vygenerování elektroschématu nebo desky plošného spoje. Obsahuje širokou databázi elektrických a elektronických součástek (např. senzory). Nechybí zde ani vývojové kity různých firem jako Arduino nebo Raspberry Pi.

### 5.4 Arduino IDE

Vývojové prostředí pro platformu Arduino, ve kterém lze psát, kompilovat a nahrávat program do mikropočítače. Podporuje různé Arduino desky. Pro jazyk C/C++ je zvýrazněná syntaxe. Volně dostupný. Jedná se o velmi přehledný program napsaný v jazyce Java. Obsahuje ukázky programů pro práci s dostupnými moduly pro Arduino a má několik vestavěných knihoven. Podporuje sériovou komunikaci s mikropočítačem.

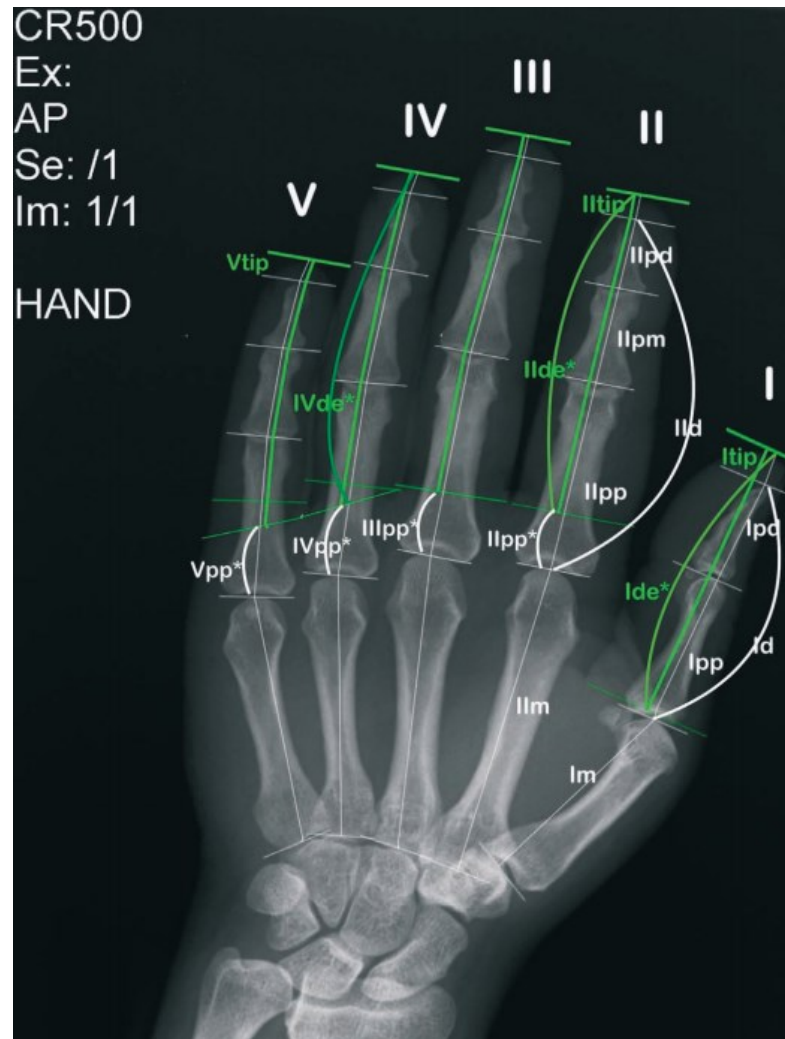
## **II. PRAKTICKÁ ČÁST**

## 6 NÁVRHANIMATRONICKÉ RUKY

### 6.1 Výchozí představa

Na začátku vývoje muselo být rozhodnuto, jaké pohyby by měla robotická ruka umět vykonávat. Bylo zvoleno několik základních pohybů – sevření (a rozevření) každého prstu zvlášť a otáčení zápěstím kolem své osy o  $180^\circ$ . Při návrhu muselo být zohledněno také to, jaký princip bude sloužit k rozpohybování jednotlivých částí. Byl vybrán tradiční mechanismus, který se běžně u animatronických rukou používá. Tvoří jej servopohony, které svým otáčením navíjejí silon (např. na řemenici). Silon je připevněn pouze na navíjecím mechanismu a na konečcích prstů. Při navíjení silonu dochází ke zkracování vnitřní vzdálenosti mezi jednotlivými články prstu a prst se tímto zavírá. Podobně probíhá opětovné otevírání prstu. Rotace zápěstí byla realizována pomocí dvou stejných ozubených kol. Jedno je pevně spojené s rukou a druhé na servopohonu. Ruka (chápejme dlaň s prsty) je uložena v základně tak, aby byl možný otočný pohyb kolem osy zápěstí. Otáčením servopohonu poté dochází k otáčení ruky v poměru 1:1.

Před samotným modelováním částí, které tvoří robotickou ruku, musela být jasná představa také o designu. Jedná se především o část dlaně a prstů. Jako nejvěrohodnější řešení se jevílo zachování samotné anatomie lidské ruky, především délky jednotlivých článků a kostí (obrázek 15, tabulka 2).



Obrázek 15 – Rentgenový snímek ruky s popisem jednotlivých kostí, I – palec, II – ukazováček, III – prostředníček, IV – prsteníček, V – malíček; tip – jemná tkáň špičky prstu; pd – distal phalanx; pm – medial phalanx; pp – proximal phalanx; m – metacarpal [14]

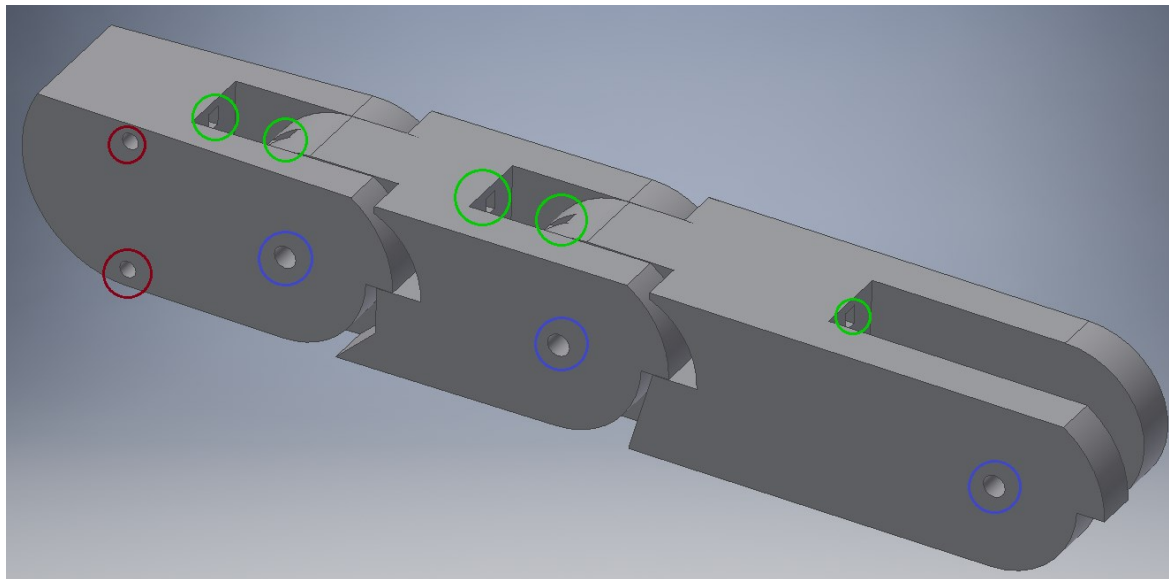
Tabulka 2 - Rozměry kostí v dlani a prstech, I – palec, II – ukazováček, III – prostředníček, IV – prsteníček, V – malíček [14]

Označení prstu	Koneček prstu – měkká tkáň [mm]	pd – distal phalanx [mm]	pm – medial phalanx [mm]	pp – proximal phalanx [mm]	m – metacarpal [mm]
I	5,67	21,67	-	31,57	46,22
II	3,84	15,82	22,38	39,78	68,12
III	3,95	17,40	26,33	44,63	64,60
IV	3,95	17,30	25,65	41,37	58,00
V	3,73	15,96	18,11	32,74	53,69

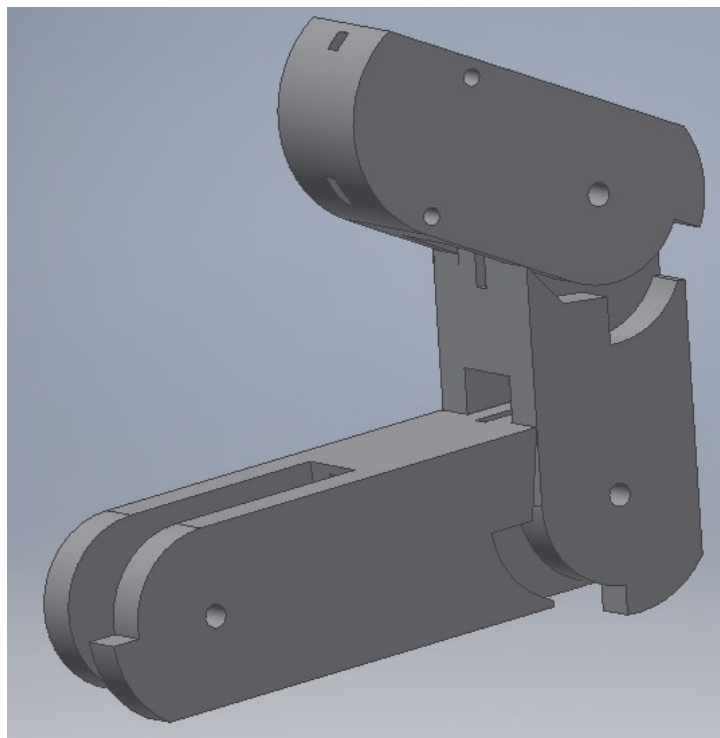
## 6.2 Návrh prstů

Všechny modely byly tvořeny v programu Autodesk Inventor 2016. Prsty jsou tvořeny třemi články (kromě palce - 2 články), které je možno spojit pomocí kolíků. Jejich délky odpovídají rozměrům z předcházejícího obrázku. Dále jsou zde otvory sloužící k provlečení silonu a jeho připevnění na konci prstu. Každý prst má také mechanické zarážky, které nedovolují překročit určitou polohu. Při otevírání nepřekročí polohu nataženého prstu a při svírání jsou vymodelovány zarážky tak, aby odpovídaly skutečnému sevření lidské ruky.

Na obr. 17 jsou červeně označeny otvory, ve kterých bude vyvrtán závit M2. Do tohoto závitu je možno našroubovat červík, který zajistí silon proti stěně na druhé straně. Tímto způsobem se silon upevní na konci každého prstu. Zeleně jsou zde vyznačeny otvory pro provlečení silonu (vidíme pouze horní stranu prstu, podobné otvory se nacházejí na spodní straně, viz obr. 18) a modré označení ukazuje na díry pro kolíky (představují zde klouby, středy otáčení).

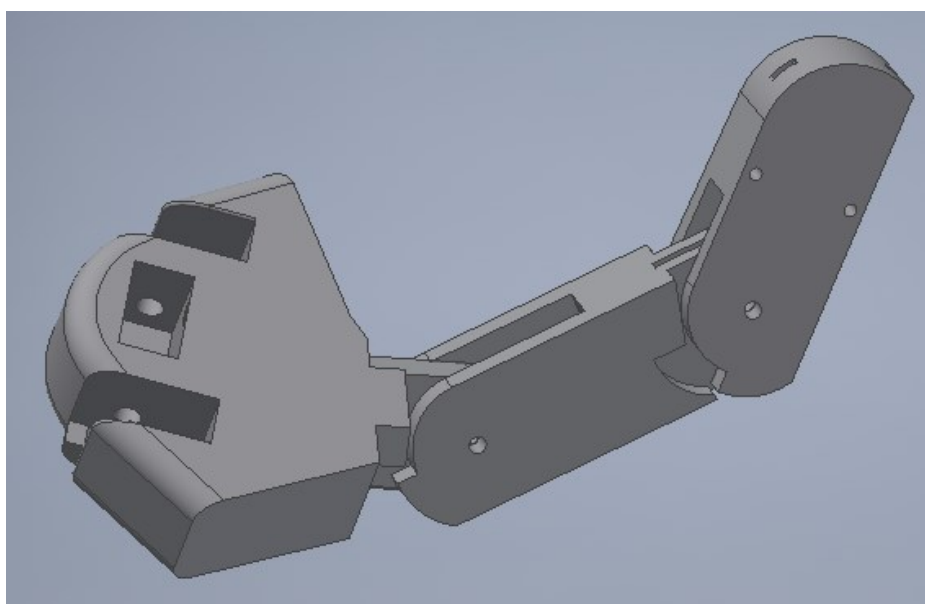


Obrázek 16 – Model prstu (ukazováček) v otevřeném stavu



*Obrázek 17 – Model prstu (ukazováček) v sevřeném stavu*

Palec je pak tvořen dvěma články a mezikusem. Mezikus představuje spojení mezi palcem a dlaní, který je schopen také měnit svou polohu vůči dlani a jeho specifický tvar podporuje úchop předmětů a dodává ruce na přirozenosti.

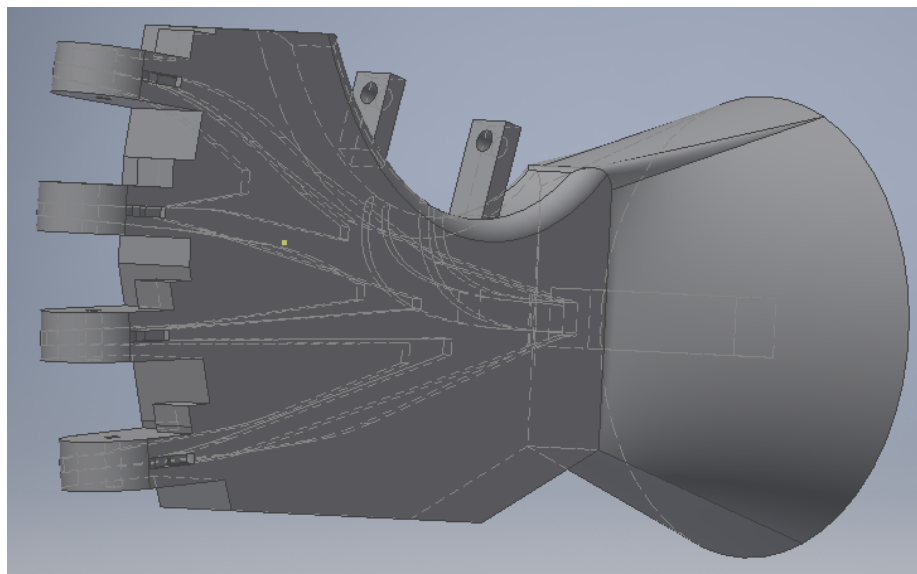


*Obrázek 18 – Model palce a mezikusu v obecné poloze*

### 6.3 Návrh dlaně a zápěstí

Při modelování dlaně bylo nutno vzít v potaz, že dlaň musí umožnit připevnění jednotlivých prstů, a také zachovat otvory, kterými bude provlečen silon. Celý model byl komplikován faktem, že silony musí z dlaně směrem k základně vycházet co nejbližší středu otáčení, aby nedocházelo k jejich zkracování (případně prodlužování) při otáčení zápěstím. Pokud by nebyla dodržena tato podmínka, docházelo by k nechtěnému svírání, případně rozevírání prstů při rotaci zápěstí. Celá dlaň je napevno spojena (jako jeden model) se zápěstím, které přechází z obdélníkového tvaru do kruhového. Spojený celek se otáčí kolem své osy. Proto je v zápěstí také vytvořen čtvercový otvor, ve kterém je uložena hřídel, na jejímž konci se nachází již zmiňované ozubené kolo. V této hřídeli je také otvor umožňující provlečení silonů. Čtvercový tvar otvoru zde zabraňuje vzájemnému pohybu hřídele a zápěstí při rotaci. Dále se na modelu dlaně nachází podobné zarážky jako u prstů. Zarážky zabraňují prstům překročit krajní polohy. Pro rozevřený prst se jedná o polohu rovnoběžnou s hřbetem dlaně. Naopak při maximálním sevření je kladen důraz na co nejpřirozenější polohu prstů. Na dlani byly také vymodelovány dva úchyty pro přišroubování palce (mezikusu).

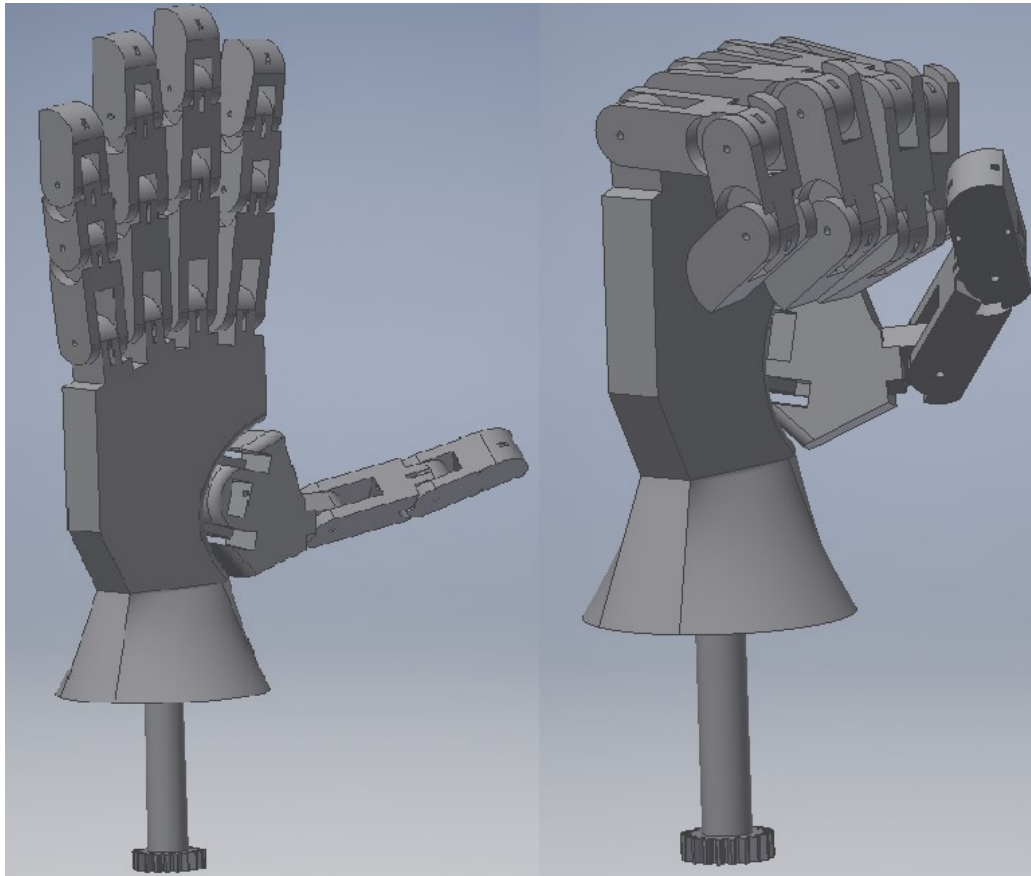
Na obr. 20 vidíme vytvořený model s kanálky pro silon, které se sbíhají ve středu otáčení a otvor přechází do čtvercového tvaru, ve kterém bude napevno natlačena část hřídele.



Obrázek 19 – Model dlaně se zápěstím, čárkovaně vyznačené skryté hrany kanálků uvnitř dlaně



Celou dosavadní sestavu je možné vidět na následujícím obrázku.



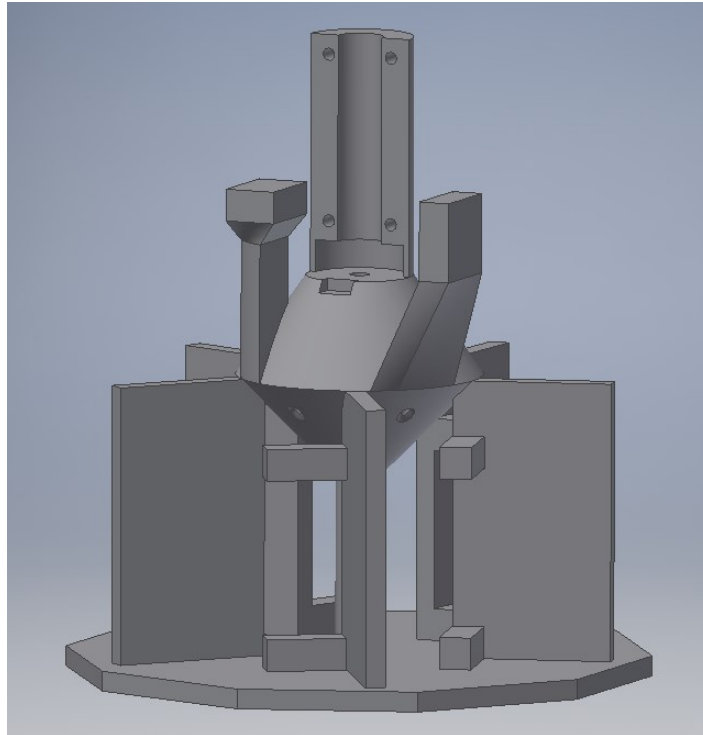
Obrázek 20 – Modely pohyblivé části, nalevo s rozevřenými prsty, napravo se sevřenými

#### 6.4 Návrh statické základny

Jak plyne z předchozích modelů, bylo zapotřebí vytvořit takovou část, která by sloužila jako pevná podstava (základna), ve které se může naše pohyblivá část otáčet. Zároveň musí základna splňovat několik požadavků. Jednou ze základních podmínek bylo vytvořit vhodné uspořádání všech šesti servopohonů, které jsou právě v základně upevněny. Kladl se zde důraz na minimalizaci délky silonu. Jeho nadměrná délka by komplikovala funkčnost celé animatronické ruky.

Základna se skládá z několika oblastí. První z nich je oblast s válcovou plochou, která slouží pro uložení hřídele. Nachází se zde 4 díry pro šrouby, pomocí nichž bude přišroubován protikus, ve kterém jsou 4 závity, a tím obepne hřídel. Hned vedle této oblasti se nachází uložení prvního servopohonu. Uložení je navrženo tak, aby ozubené kolo na

hřídeli doléhalo právě na ozubené kolo připevněné na tomto servopohonu. Druhá oblast se poté skládá z otvorů, kterými prochází silony od servopohonů k otvoru v hřídeli. Poslední oblast slouží k upevnění zbylých pěti servopohonů.



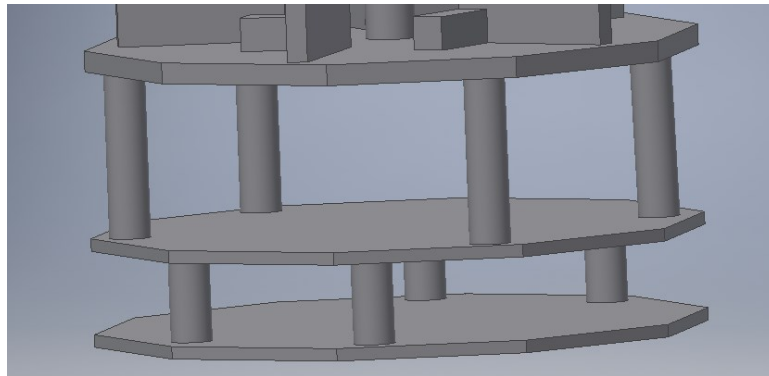
Obrázek 21 – Model základny

## 6.5 Návrh prostoru pro elektroniku, napájení a celkový model

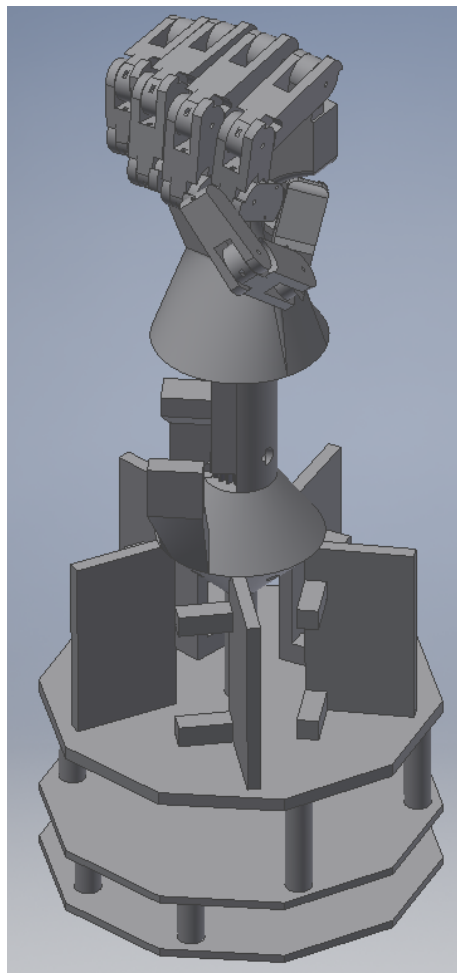
Při vytváření bylo jasné, že ruka bude řízena mikropočítačem za použití elektronických součástek a nepájivého pole. Dále bylo známo napájení servopohonů, které tvořily dvě pouzdra, každé po 4xAA bateriích a jedna 9V baterie pro napájení mikropočítače. Všechny součástky bylo nutno umístit k základně.

Prostor byl namodelován s využitím tvaru podstavy základny. Nová deska (totožný tvar jako podstava) slouží k upevnění součástek tak, že z jedné strany (vrchní, směr k základně) se nachází mikropočítač a nepájivé pole, a na druhé (spodní) straně jsou umístěny baterie. Deska je se základnou spojena tak, že jsou zde umístěny čtyři distanční sloupky (40 mm) mezi základnu a desku, které jsou přišroubovány k základně i k desce. Deska s elektronikou a bateriemi je odnímatelná při odšroubování těchto šroubů. Aby baterie netvořily dno celé základny, byla vytvořena ještě jedna deska stejného tvaru, která je přichycena stejným způsobem pomocí distančních sloupků (20 mm) k desce

s elektronikou. K bateriím se tedy dostaneme odšroubováním poslední desky. Takto vytvořený komplet je již finální verzí modelu.



*Obrázek 22 – Detail modelu základny, vzniklý prostor pro baterie a elektroniku*



*Obrázek 23 – Kompletní model robotické ruky bez krytu*

## 7 NÁVRH ELEKTRONIKY

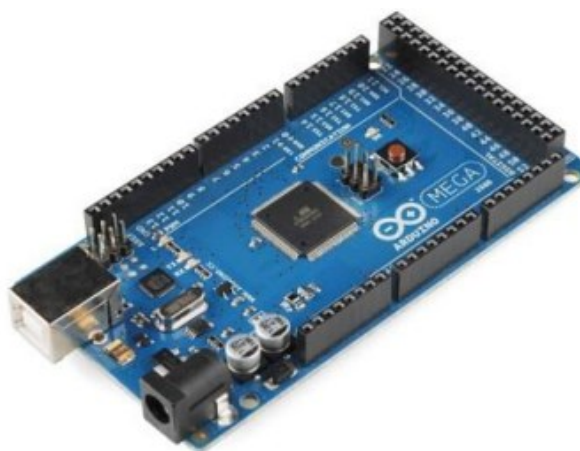
### 7.1 Elektronické součásti robotické ruky

Základem animatronické ruky je mikropočítač, který řídí šest servomotorů. Mikropočítač je napájen pomocí 9V baterie. Napájení servomotorů obslouží 8 AA baterií, rozdělených po čtveřicích zapojených v sérii.

#### 7.1.1 Vývojová platforma

Jako „srdce“ celé robotické ruky byl zvolen mikropočítač Arduino Mega 2560. Jedná se o jeden z mála mikropočítačů poskytující dostatečné množství vstupů a výstupů pro případné další vylepšování animatronické ruky v budoucnosti.

U mikropočítače využíváme hlavně PWM výstupy, které slouží k ovládání servopohonů. V našem případě mikropočítač zpracovává přijatá data z přijímače (viz dále kapitola 8 Bezdrátová komunikace) a na základě těchto dat upravuje polohu jednotlivých servopohonů. Pro nahrání programu do mikropočítače lze využít USB konektor.



Obrázek 24 – Vývojový kit Arduino Mega 2560 [15]

Tabulka 3 – Parametry kitu Arduino Mega 2560 [15]

Mikrokontrolér	Atmega2560
Provozní napětí	5 V
Vstupní napětí (doporučené)	7-12 V
Vstupní napětí (limit)	6-20 V
Digitální I/O piny	54 (z toho 15 poskytuje PWM výstup)
Analogové vstupy	16
DC proud I/O pinu	20 mA

FLASH paměť	256 KB (z toho 8 KB používá zavaděč)
SRAM	8 KB
EEPROM	4 KB
Frekvence mikroprocesoru	16 MHz
Rozměry	101,52x53,3 x 15,1 mm
Hmotnost	37 g

### 7.1.2 Servopohony

Z dostupných servopohonů, vhodných k mikropočítačům, byl zvolen model firmy Futaba, označení Futaba S3003 Servo. Toto servo se hojně využívá v modelářství. Hlavními oblastmi jsou RC auta, letadla a lodě.



Obrázek 25 – Servopohon Futaba S3003 [16]

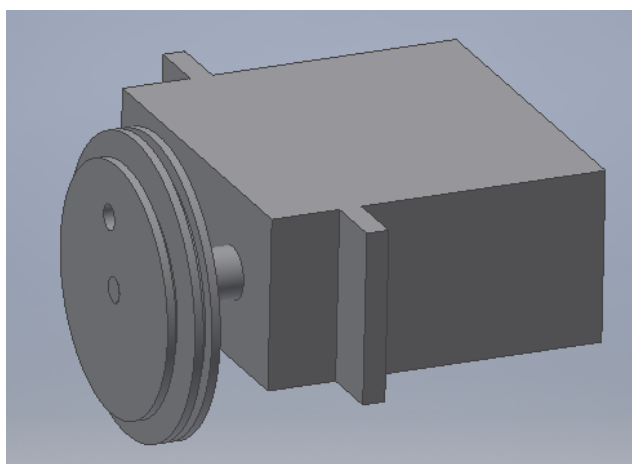
Tabulka 4 – Parametry servopohonu Futaba S3003 [16]

Řízení	Analogové
Vstupní napětí	4,8 – 6 V
Rychlost otáčení bez zátěže	0,23s/60° při 4,8 V 0,19s/60° při 6 V
Moment	3,2kg/cm při 4,8 V 4,1kg/cm při 6 V
Amplituda PWM	3-5 V
Možnost modifikace na kontinuální rotaci	Ano
Směr rotace s rostoucím PWM	Proti hodinovým ručičkám
Typ motoru	Stejnoseměrný
Podpora výstupní hřídele	Plastové ložisko

Materiál převodů	Nylon
Rozměry	40 x 20 x 36 mm
Hmotnost	37 g

Ovládání úhlu natočení serva je řízeno PWM piny z mikropočítače. Každý servopohon je osazen řemenicí, ve které je upevněn druhý konec silonu. Řemenice se skládá ze dvou různých průměrů. Na jednom průměru je upevněn silon pro svírání prstu, a na druhém průměru silon pro rozevírání prstu. Průměr řemenic pak ovlivňuje délku namotání silonu při otočení serva o určitý úhel. Potřebné délky namotání silonu se liší pro každý prst a pro každý pohyb (svírání/rozevírání).

Obr. 27 ukazuje model řemenice osazenou na servopohonu. Otvor ve středu řemenice slouží pro uchycení k servopohonu šroubem. Druhý otvor poté slouží k upevnění silonů.



*Obrázek 26 – Názorný model servomotoru s upevněnou řemenicí*

Všechny velikosti řemenic byly určeny měřením vzniklých modelů a následným přepočtem. Bylo uvažováno, že naměřená délka navinutí silonu odpovídá  $180^\circ$  na řemenici (celý rozsah servomotoru). Délka tedy odpovídá polovině obvodu kruhu. Odtud lze určit průměr řemenice. Naši naměřenou vzdálenost můžeme označit  $x$ . Všechny průměry dodatečně vynásobíme konstantou 1,2. Hlavním důvodem je nedokonalost servopohonu, který nepokryje celých  $180^\circ$ . Poté platí:

$$2x = o \quad (1)$$

$$o = \pi d \quad (2)$$

$$d = \frac{o}{\pi} = \frac{2x}{\pi} \quad (3)$$

$$d' = d * 1,2 \quad (4)$$

Pro malíček tedy platí:

$$d' = \frac{2 * 40,22}{\pi} * 1,2 = 30,73 \text{ mm} \quad (5)$$

Stejným způsobem se dopočítaly i zbylé hodnoty.

Tabulka 5 – Naměřené a vypočítané hodnoty pro řemenice

	Potřebná délka navinutí x [mm]		Přečítaný průměr řemenic d' [mm]	
	Svírání	Rozevírání	Svírací	Rozevírací
Malíček	40,22	32,44	30,73	24,78
Prsteníček	38,5	32	29,41	24,45
Prostředníček	39,41	34,8	30,11	26,58
Ukazováček	39,46	31,78	30,15	24,28
Palec	37,25	31	28,46	23,68

### 7.1.3 Napájení

Aby byla zachována dobrá přenositelnost zařízení, bylo zvoleno napájení z baterií. Napájení kitu je zprostředkováno pomocí 9V baterie. Baterie je vyměnitelná a její typ neovlivňuje funkcionalitu celé animatronické ruky. Kapacita těchto baterií se pohybuje okolo 500 mAh. S kitem je spojena pomocí konektoru.



Obrázek 27 – Konektor pro 9V baterii [17]

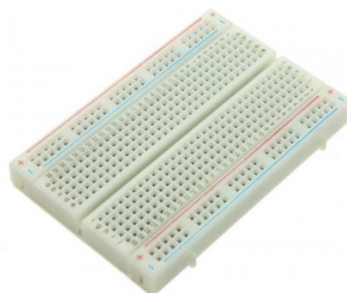
Pro napájení servomotorů není možné použít vývojový kit. Hlavním důvodem je to, že servopohony mají velké špičkové proudy. Ty můžou vystoupat až k 1 A. Za těchto okolností by mikropočítač přestal správně pracovat (mohl by se vypínat/restartovat). Správně by nepracovaly ani servopohony. Na základě předešlých poznatků bylo zvoleno napájení z AA akumulátorů. Tři servopohony jsou napájeny ze čtyř AA akumulátorů zapojených do série a zbylé tři serva z dalších čtyř AA akumulátorů. K jednoduchému zapojení baterií do série slouží pouzdro. Jako typ baterie byly zvoleny dobíjecí akumulátory GP ReCyko+ 2500mAh. Jedná se o NiMH baterie. Jejich jmenovité napětí je 1,2V a v sérii poskytnou napětí 4,8V. [18]



*Obrázek 28 – Bateriový box  
pro 4xAA baterie [19]*

#### **7.1.4 Nepájivé kontaktní pole**

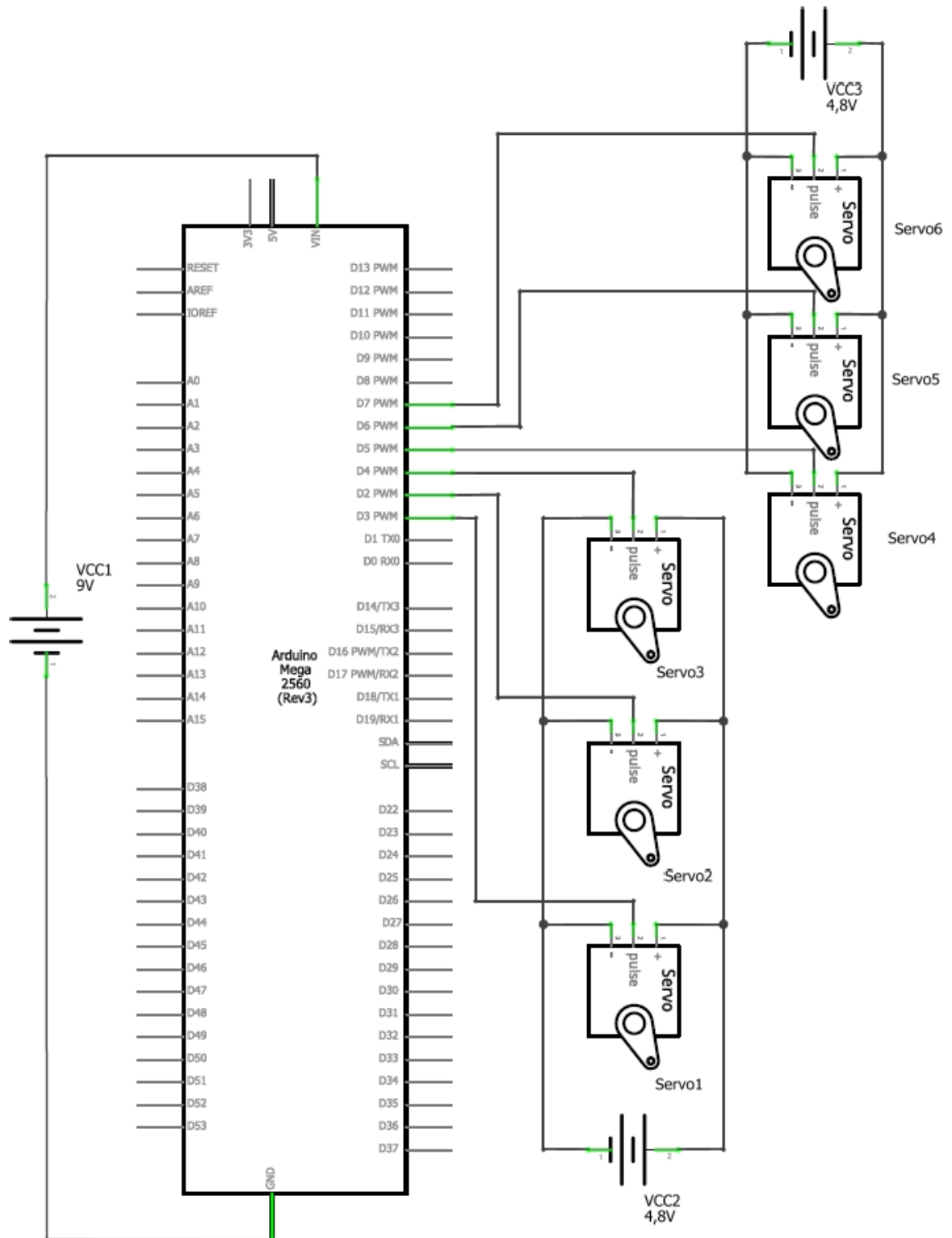
Pro propojení jednotlivých součástek bylo použito nepájivé pole. Toto pole umožňuje jednoduché zapojení součástek a zároveň podporuje libovolné rozšíření celého projektu v budoucnu. Další výhodou je možnost využití součástek i v jiném projektu, bez nutnosti mechanického zásahu do zapojení. Požité nepájivé pole má 400 pinů, z nichž 100 slouží pro napájení. Rozměry pole jsou 55 x 82 x 8,5 mm. [20]



*Obrázek 29 – Kontaktní  
nepájivé pole [20]*



7.1.5 Schéma zapojení



Obrázek 30 – Schéma zapojení mikropočítače, servopohonů a napájení, vytvořeno v programu Fritzing

Schéma zapojení je velmi jednoduché, každý servopohon je napájen z baterie a řízen pomocí PWM pinu mikropočítače. K tomuto účelu jsou využity PWM piny 2,3,4,5,6,7. Piny jsou rozděleny následovně:

- PWM pin 2 – ovládání palce
- PWM pin 3 – ovládání prsteníčku
- PWM pin 4 – ovládání prostředníčku
- PWM pin 5 – ovládání ukazováčku
- PWM pin 6 – ovládání malíčku
- PWM pin 7 – ovládání rotace zápěstí

## 7.2 Elektronické součásti rukavice pro ovládání

Pro ovládání robotické ruky je potřeba získat údaje o ohybu prstů a poté náklonu samotného zápěstí. Protože byl projekt navrhnut s bezdrátovým přenosem, bylo využito dalšího mikropočítače, který zpracovává data ze senzorů a přes bezdrátovou komunikaci zasílá informaci do mikropočítače v robotické ruce. Napájení je zajištěno z 9V baterie. Rukavice disponuje také kalibrací, která je zajištěna pomocí dvou tlačítek a následným programovým zpracováním.

### 7.2.1 Vývojová platforma

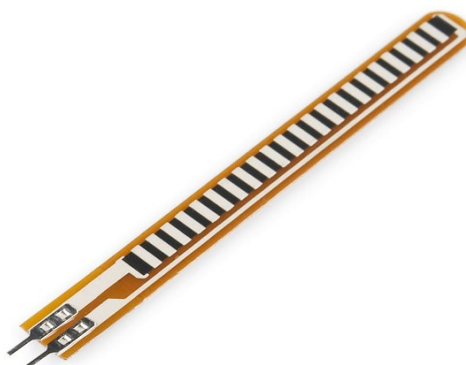
Jako základ rukavice byla zvolena opět vývojová platforma Arduino Mega 2560, jehož specifikace naleznete v tabulce 3. V této části byly využity převážně analogové vstupy mikropočítače. Analogové vstupy slouží pro přečtení hodnoty napětí na nich přiloženém. Platforma Arduino poskytuje funkce, které z analogového vstupu vracejí hodnotu 0-1023 (desetibitový převodník) se kterými se dále pracuje. Podrobněji v kapitole 9 - Program.

### 7.2.2 Sensory pro snímání ohybu prstů

Pro potřebu snímání ohybu se bohužel na trhu nenachází velké množství řešení. Nejschůdnějším z nich se jeví využití tzv. flex senzorů. Flex senzory nacházejí uplatnění v robotice, v herním průmyslu (ovladače) nebo ve zdravotnických zařízeních. Byly zvoleny senzory firmy Sparkfun s činnou délkou 56 mm (2,2“).

Princip funkce je prostý. Pokud je senzor ohýbán, roste jeho odpor. V našem případě se jedná o zapojení senzoru a rezistoru v sérii, kde je měřen úbytek napětí na předřadném

rezistoru vlivem rostoucího odporu flex senzoru. Jedná se vlastně o dělič napětí. Jako napětí zde bude sloužit 5 V poskytnuto z desky Arduina. [21]



Obrázek 31 – Flex senzor firmy Sparkfun 2,2“ [21]

Tabulka 6 – Parametry flex senzoru Sparkfun 2,2“ [21]

Životní cyklus	>1 milion ohybů
Odpor flat (původní narovnaný tvar)	10 k $\Omega$
Odpor bend (180° ohyb)	>2x odpor flat
Jmenovitý výkon	0,5 W nepřetržitý 1 W špička
Rozsah teplot	-35 °C až 80 °C
Výška senzoru	≤0,43 mm

### 7.2.3 Senzor pro snímání rotace zápěstí

Pro snímání rotace zápěstí byl zvolen modul MPU-6050. Jedná se o modul, který kombinuje tříosý akcelerometr a tříosý gyroskop. Pro komunikaci s Arduinem slouží I2C sběrnice. Z modulu lze vyčíst aktuální hodnota požadované veličiny nebo obsahuje také zásobník typu FIFO pro vyčítání všech v čase naměřených hodnot (zásobník pro 1024 hodnot).



Obrázek 32 – Modul MPU-6050 [22]

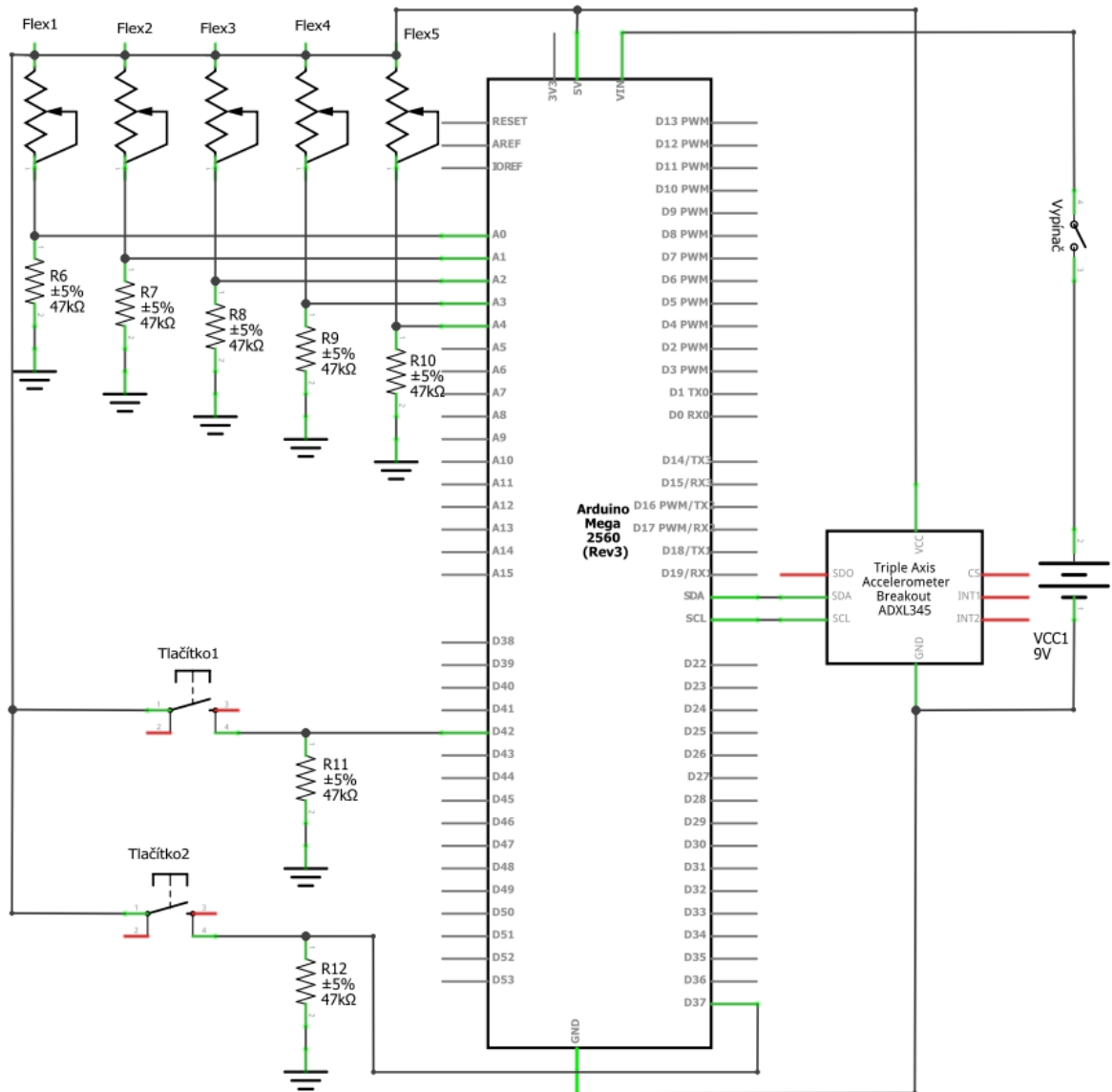
Tabulka 7 – Parametry modulu MPU-6050 [22]

Komunikační standard	I2C
Pracovní napětí	3-5 V
A/D převodník	16ti bitový
Rozsah gyroskopu	$\pm 250 \pm 500 \pm 1000 \pm 2000$ °/sec
Rozsah akcelerometru	$\pm 2 \pm 4 \pm 8 \pm 16$ g (tíhové zrychlení)
Rozměry	21 x 15 x 1,2 mm
Hmotnost	3 g (gram)

#### 7.2.4 Ostatní součástky

Další části elektrického obvodu jsou tvořeny rezistory, na kterých jsou měřeny napětí (napojeny na vstupní piny Arduina). Jejich velikost je  $47k\Omega$ . Také se zde nachází dvě tlačítka, která jsou přes pull-down rezistory spojena s Arduinem. Na základě stlačení těchto tlačítek se provádí kalibrace. Arduino je napájeno z 9V baterie, která je připojena přes vypínač. Je zde využito kontaktního nepájivého pole. Celé schéma lze vidět na obr. 34.

## 7.2.5 Schéma zapojení



Obrázek 33 – Schéma zapojení elektroniky v rukavici pro ovládání, vytvořeno v programu Fritzing

V levé horní části schématu se nachází všech pět flex senzorů (pro každý prst jeden). Sensory jsou sériově zapojeny s rezistory. Na těchto rezistorech se s ohybem flex senzoru zmenšuje napětí. Flex senzory jsou napojeny na 5 V napětí z desky Arduina a rezistory poté na zem Arduina (gnd). Napětí na rezistorech slouží jako analogové vstupy. Dále jsou využity dva digitální vstupy pro snímání tlačítek (levá dolní část). Senzor MPU-6050 (vpravo od arduino desky, ve schématu použit jiný model, zapojení ovšem stejné) komunikuje pomocí I2C, je tedy napojen na piny SDA, SCL a zároveň je napájen z desky.

Poslední část schématu tvoří napájení z baterie 9V, která je zapojena přes vypínač. Tímto můžeme vypínat elektroniku celé rukavice.

Piny jsou rozděleny následovně:

- A0 (analogový vstup) – flex senzor na malíčku
- A1 (analogový vstup) – flex senzor na ukazováčku
- A2 (analogový vstup) – flex senzor na prostředníčku
- A3 (analogový vstup) – flex senzor na prsteníčku
- A4 (analogový vstup) – flex senzor na palci
- D42 (digitální vstup) – tlačítko pro kalibraci 1
- D37 (digitální vstup) – tlačítko pro kalibraci 2
- SDA, SCL – komunikace s modulem MPU-6050

## 8 BEZDRÁTOVÁ KOMUNIKACE

Celý projekt byl navrhován s bezdrátovým ovládáním animatronické ruky. V předchozích kapitolách bylo pro jednoduchost vynecháno vše, co se této problematiky týká. Existuje celá řada možností, jak přenášet potřebná data vzduchem. V tomto projektu byla zvolena možnost rádiové komunikace. Jedná se o dvojici modulů (přijímač, vysílač) poskytující dosah až 200 m (na volném prostranství), pracující na frekvenci 433,92 MHz. Výhody modulů jsou hlavně nízká cena a jednoduchost zapojení. [23]

### 8.1 Vysílač

Vysílač se skládá z tří pinů, kde jeden slouží pro zaslání dat (Data), druhý napájení (Vcc) a třetí pro připojení země (Gnd). Na zakoupené desce se nachází také velmi jednoduchá anténa, která ovšem nemá velký dosah. Při práci s moduly se využívá knihovny VirtualWire, která poskytuje potřebné funkce pro komunikaci.



Obrázek 34 – Modul vysílače 433,92 MHz [24]

Tabulka 8 – Parametry vysílačeho modulu [23]

Typ vysílače	XY-FST
Komunikační frekvence	315 MHz nebo 433,92 MHz
Provozní napětí	3-12 V
Provozní proud	9-40 mA
Šířka pásma	2 MHz
Vysílací výkon	25 mW pro 315 MHz, 12 V
Přenosová rychlost	<9,6 Kb/s
Rozměry	19 x 19 mm

### 8.1.1 Anténa

K modulu se doporučuje pořídit zvlášť vhodnou anténu. Lze ji zakoupit u mnoha výrobců nebo ji lze vyrobit. V tomto projektu byla anténa vytvořena pomocí měděného drátu vhodné délky, který byl připájen na desku vysílače. Stejná anténa byla umístěna i na modul vysílače.

Pro výpočet délky antény:

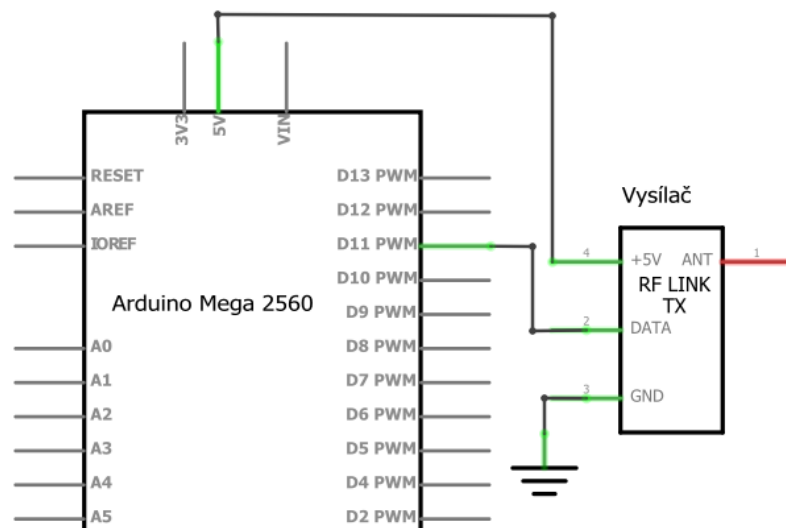
$$l = \frac{c}{4f} = \frac{299\,792\,458}{4 * 433\,920\,000} = 0,173\,m \quad (6)$$

Kde  $c$  je rychlost světla ve vakuu a  $f$  je frekvence.

Jako anténa tedy slouží drát o délce 173 mm. [25]

### 8.1.2 Schéma propojení vysílače s Arduinem

Pro připojení vysílače byl využit digitální pin D11. Vysílač je napájen z desky Arduina (5 V).



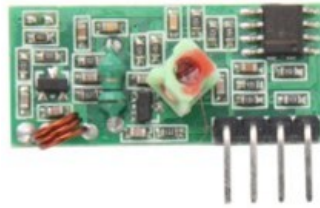
Obrázek 35 – Schéma zapojení vysílače k desce Arduina (rukavice pro ovládání)

## 8.2 Přijímač

Přijímač obsahuje čtyři piny, jejichž funkce jsou totožné jako u přijímače. Dva z nich slouží pro napájení (5V a gnd) a další dva pro data. Datové piny jsou spojeny (dva



prostřední), nezáleží tedy na tom, který propojíme s deskou Arduina. Podobně jako u vysílače je i u přijímače anténa v základu nedostačující.



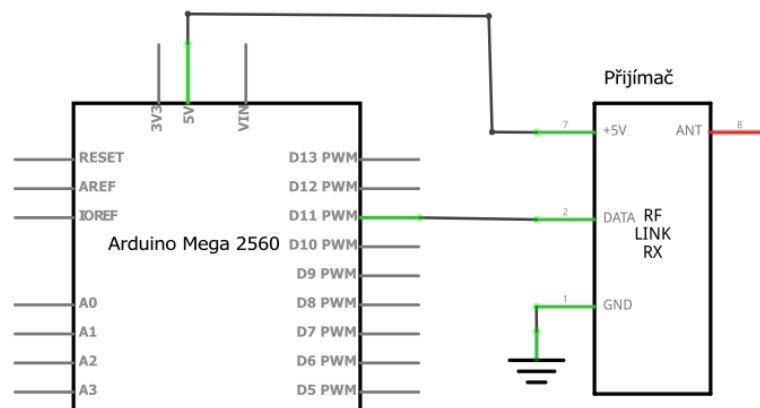
Obrázek 36 – Modul přijímače 433,92 MHz [24]

Tabulka 9 – Parametry přijímacího modulu [23]

Typ vysílače	XY-MK-5V
Komunikační frekvence	315 MHz nebo 433,92 MHz
Provozní napětí	5 V
Provozní proud	<6 mA
Přenosová rychlost	<9,6 Kb/s
Rozměry	30 x 14 mm

### 8.2.1 Schéma propojení přijímače s Arduinem

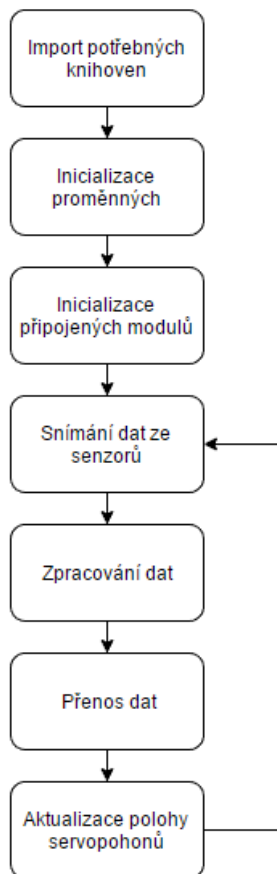
Pro připojení přijímače byl využit digitální pin D11. Modul je napájen z desky Arduina (5V).



Obrázek 37 – Schéma zapojení vysílače k desce Arduina (animatronická ruka)

## 9 PROGRAM

Nejprve byla vytvořena obecná představa o tom, jak by měl program fungovat. Situaci popisuje obr. 39. Kód byl poté rozdělen na dvě části, z nichž každá byla nahrána do příslušného mikropočítače.



Obrázek 38 – Vývojový diagram programu

### 9.1 Kód programu rukavice

Celý zdrojový kód s komentáři je k nalezení v příloze PI. V této kapitole jsou popsány nejdůležitější body programu s případnými ukázkami kódu.

Kód začíná importováním potřebných knihoven. Těmi jsou VirtualWire sloužící k bezdrátové komunikaci, I2Cdev potřebná ke komunikaci s modulem MPU6050 a samotná knihovna MPU6050 poskytující inicializaci, kalibraci a základní funkce pro práci s modulem.

Další část slouží k definování proměnných, případně struktur. Nachází se zde nastavení všech čísel pinů pod příslušné označení. Následující část kódu ukazuje vytvořenou datovou strukturu, která se přenáší pomocí bezdrátové komunikace.

```
typedef struct dataToSend // definování struktury s daty k odeslání  
{  
int Sensor1Data;  
int Sensor2Data;  
int Sensor3Data;  
int Sensor4Data;  
int Sensor5Data;  
int Sensor6Data;  
};
```

Každý program pro Arduino se pak skládá ze dvou částí. První z nich je funkce *setup()*. Tato část kódu slouží k jednorázovému provedení instrukcí v ní obsažených (např. inicializace jednotlivých zařízení). Provádí se po každém startu nebo restartu Arduina. Nachází se zde definování pinu jako vstupu nebo výstupu. Dále jsou zde nastaveny počáteční hodnoty maxima a minima pro každý senzor. Tyto hodnoty slouží k nastavení rozsahu pohybu prstů v rukavici. Čím menší interval je, tím větší citlivost pak servopohony mají. Poslední část *setup()* obsahuje inicializaci potřebných zařízení. Jedná se o nastavení a spuštění bezdrátové komunikace, a také inicializace modulu MPU6050 s nastavením offsetů.

Druhou část kódu v Arduino tvoří funkce *loop()*. Jak plyne z názvu, jedná se o smyčku, která se provádí pořád dokola pokud Arduino běží. Probíhá zde snímání hodnoty analogových vstupů, pomocí funkce *analogRead()*. Funkce vrací hodnotu 0-1023 (přepočít z hodnot 0-5 V napětí na pinu, 5 V = 1023, 0 V = 0). Hodnota je uložena do proměnné. Probíhá následné zpracování, kde se kontroluje, zda hodnota spadá do intervalu min-max. Pokud ne, hodnota se příslušně ořízne na max nebo min. Poté se hodnota promítne z intervalu min max do intervalu 0-1500 (pomocí funkce *map()*). Takto promítnuta hodnota už slouží k ovládní serva. Na konci těchto operací se provede funkce *delay(2)*, která slouží k vytvoření pauzy pro ustálení hodnot na A/D převodníku. Popsaný úsek kódu vypadá následovně:

```
analogFlex1=analogRead(flexInput1); //přečtení hodnot na analogovém vstupu (A0) a  
uložení do proměnné
```

*//zpracování hodnot (pokud dojde k překročení hranice intervalu max min, je hodnota oříznuta na max nebo min hodnotu)*

```
if(analogFlex1<flexMin1)
```

```
{
```

```
  analogFlex1=flexMin1;
```

```
}
```

```
if(analogFlex1>flexMax1)
```

```
{
```

```
  analogFlex1=flexMax1;
```

```
}
```

*servoAngle1=map(analogFlex1, flexMin1, flexMax1,0,1500); //mapování naměřené hodnoty na interval 0-1500 (slouží k ovládní serva)*

*delay(2); //pauza pro ustálení A/D převodníku*

Stejný postup se provede pro každý flex senzor. K vyčtení okamžité hodnoty z MPU6050 slouží funkce *getMotion6()*. K určení rotace nám slouží hodnota akcelerometru v ose y. Tato hodnota se s rotací zápěstí mění a lze tak hodnotu zpracovat a převést na úhel natočení serva. Hodnota z MPU je zpracována stejně jako hodnoty z flex senzorů. Nakonec se hodnoty uloží do struktury pro přenesení a dojde k odeslání.

```
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
```

*//načtení potřebných dat do struktury k přenesení*

```
data.Sensor1Data = servoAngle1;
```

```
data.Sensor2Data = servoAngle2;
```

```
data.Sensor3Data = servoAngle3;
```

```
data.Sensor4Data = servoAngle4;
```

```
data.Sensor5Data = servoAngle5;
```

```
data.Sensor6Data = servoAngle6;
```

```
vw_send((uint8_t *)&payload, sizeof(payload)); // odeslání dat
```

```
vw_wait_tx(); // čekání na dokončení odesílání
```

Konec programu slouží k provedení případné kalibrace při stisku tlačítka. Jedná se o jednoduchou úpravu intervalu min max pro každý prst. Kalibruje se tedy poloha otevřené dlaně ve vodorovné poloze (dlaň míří směrem k podlaze) a poté poloha sevřené dlaně

otočené přibližně o úhel 150° (dlaň směřuje vzhůru). Následující kód ukazuje kalibraci otevřené dlaně.

*//pokud bylo zmáčknuto tlačítko max, provede se kalibrace max, určí se nové maximální hodnoty z naměřených*

```
if(buttonStateMax == HIGH){
  flexMax1=analogRead(flexInput1);
  delay(2);
  flexMax2=analogRead(flexInput2);
  delay(2);
  flexMax3=analogRead(flexInput3);
  delay(2);
  flexMax4=analogRead(flexInput4);
  delay(2);
  flexMax5=analogRead(flexInput5);
  delay(2);
  rotateMax=ay;
  delay(2);
}
```

## 9.2 Kód programu animatronicke ruky

Celý kód je k nalezení v příloze P II. Začátek programu je podobný jako v předchozím případě, je zde nově knihovna ServoTimer2, která slouží pro práci se servomotory. Následuje vytváření proměnných a případné přiřazení pinů. Ve funkci *setup()* je ke každému servopohonu přidělen PWM pin a probíhá zde nastavení bezdrátové komunikace. Způsob přiřazení pinu k servomotoru ukazuje následující část kódu.

*//nastavení příslušných pinů k servomotorům*

```
myServo1.attach(servoPin1);
```

Ve funkci *loop()* probíhá pouze přijetí zprávy a nastavení úhlů servomotorů. Argumentem funkce *write()* u servopohonů je číslo od 750 do 2250. Při 750 se nachází servo v základní poloze. Číslo 2250 poté určuje otočení o 180°.

*vw\_wait\_rx();//čekání na příchozí zprávu*

```
vw_get_message((uint8_t *)&data, &sizeofData); //uložení příchozích dat do vytvořené struktury
```

```
myServo1.write(2250-receivedData.Sensor1Data);//nastav servo na požadovaný uhel
```

## 10 FYZICKÁ REALIZACE PROJEKTU

### 10.1 3D Tisk

Pro realizaci animatronické ruky byla zvolena metoda 3D tisku. Všechny modely byly vytvářeny s ohledem na to, aby je bylo možno vytisknout na 3D tiskárně. Nejprve proběhl převod jednotlivých modelů do formátu .stl, který je možno dále zpracovat pro 3D tisk. V programu Slic3r poté byla zvolena vhodná nastavení pro tisk a proběhl export do .gcode, který funguje jako kód pro 3D tiskárnu. Většina modelů byla tisknuta s výškou vrstvy 0,35 mm nebo 0,2 mm.

#### 10.1.1 Tiskárna

K tisku sloužil model Prusa i3 MK2. Disponuje pracovní plochou 250 x 210 x 200 mm. Obsahuje LCD displej a podporuje tisk z SD karty nebo komunikaci pomocí USB portu. Tiskárna umožňuje tisk s výškou vrstvy od 0,05 mm. Dále je vybavena softwarem pro automatickou kalibraci tiskové plochy a také kompenzací nepřesně složených os. Tiskárna obsahuje vyhřívanou podložku a 0,4 mm trysku pro 1,75 mm tiskovou strunu. Podporuje velké množství materiálů, jako jsou PLA, ABS, PET, HIPS, Nylon a další. [26]



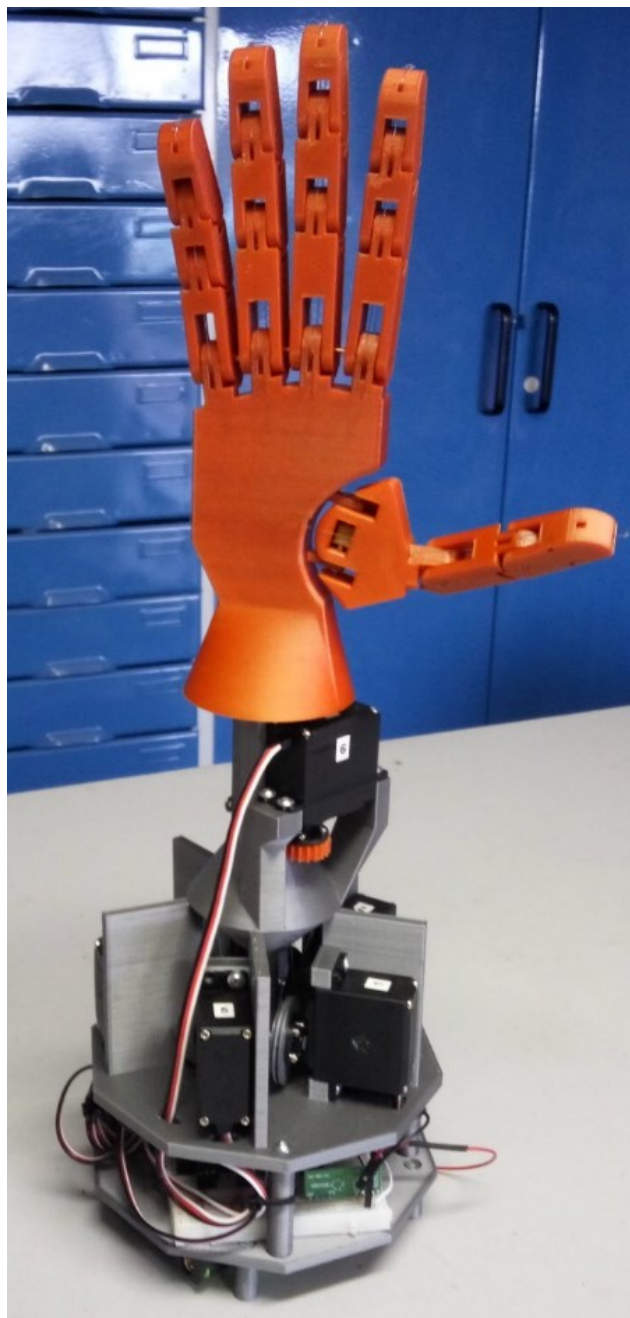
Obrázek 39 – Tiskárna Prusa i3 MK2 [26]

### 10.1.2 Materiál

Jako materiál byl zvolen plast PLA. PLA se vyrábí zpracováním z biomasy (z kukuřice, kukuřičného škrobu). Je tedy vyráběn z obnovitelných zdrojů. Dosahuje podobných vlastností jako plasty vyráběné z ropy (pružnost, tvrdost, ohebnost). Lze jej biologicky degradovat a zkompostovat, a tak nepředstavuje biologickou zátěž. Tisk probíhá okolo teploty 210°C. Materiál PLA je tvárný od teploty 60 °C. Při tisku vylučuje oproti ostatním materiálům menší množství ultrajemných částic. Mimo 3D tisku se používá v potravinářství nebo pro nemocniční zařízení. Při tisku byly použity dvě barevné varianty – měděná a stříbrná. [27] [28]

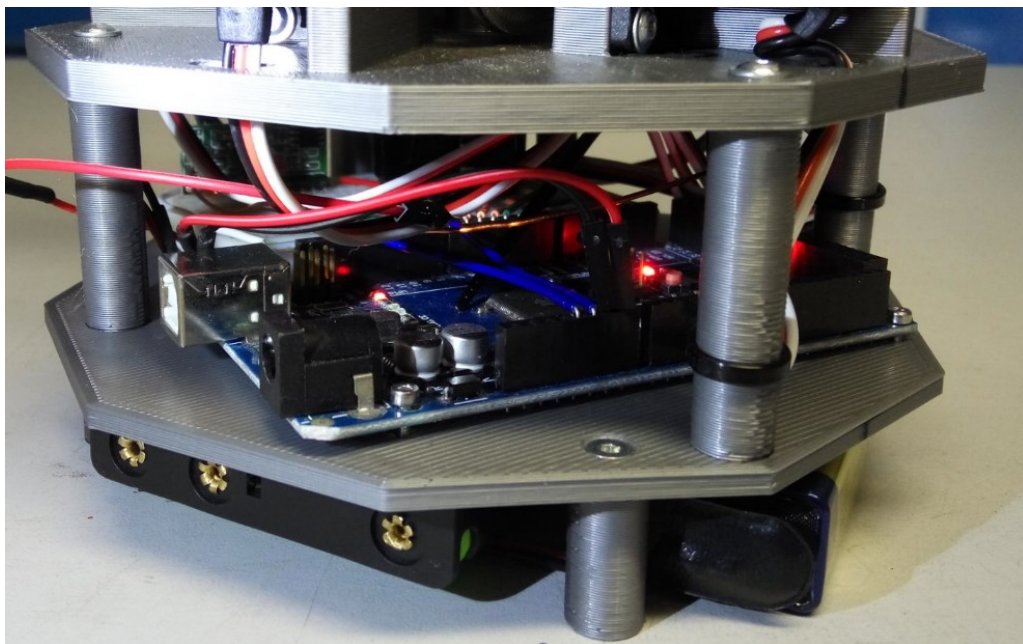
### 10.2 Animatronická ruka

Po vytisknutí byly všechny díly postupně skládány dohromady. Nejprve proběhlo sestavení prstů, jejichž články jsou spojeny pomocí kolíků (průměr 2 mm, délka 15 mm). Kromě palce byly všechny prsty k dlani připevněny pomocí měděného drátu (průměr 1 mm). Část s palcem byla k dlani přišroubována (2 šrouby M2x10). Následně proběhlo natlačení hřídele do čtvercového otvoru v zápěstí/dlani. Takto sestavená ruka byla umístěna do kruhového otvoru základny a uchycena za pomoci protikusu (4 šrouby M3x15, v protikusu se nacházejí čtyři díry, ve kterých byly vyvrtány závity M3). Následně proběhlo vyvrtání děr pro uchycení všech servopohonů (a díry pro kabeláž) a také děr pro uchycení distančních sloupků mezi základnou a podstavou pro elektroniku. Každý servopohon je přichycen k základně pomocí čtyř šroubů M2x10. Na podstavu pro elektroniku se přichytily všechny komponenty a proběhlo zapojení dle schématu.

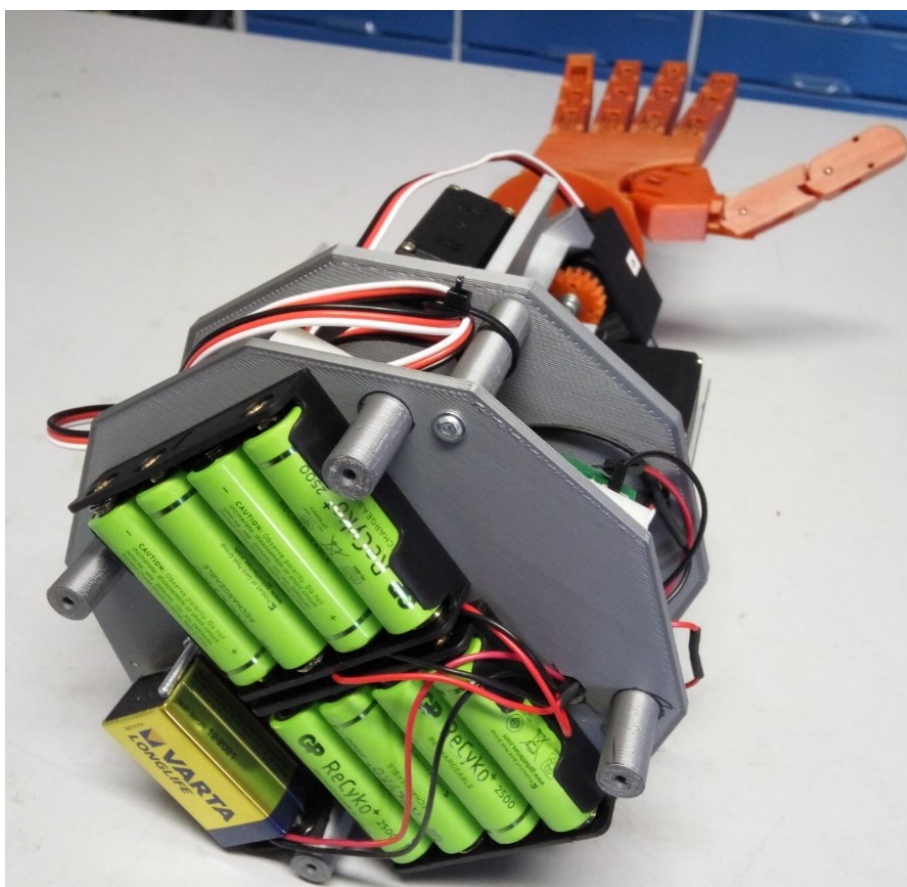


*Obrázek 40 – Výsledné provedení animatronicke ruky, bez krytu*





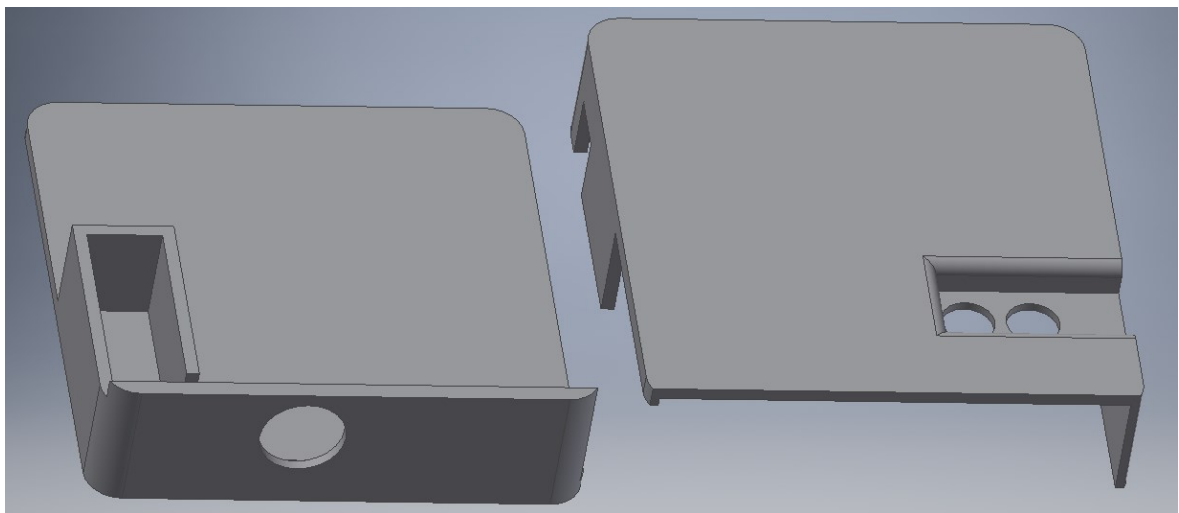
Obrázek 41 – Umístění elektroniky



Obrázek 42 – Pohled na baterie, přístupnost ze spodní strany

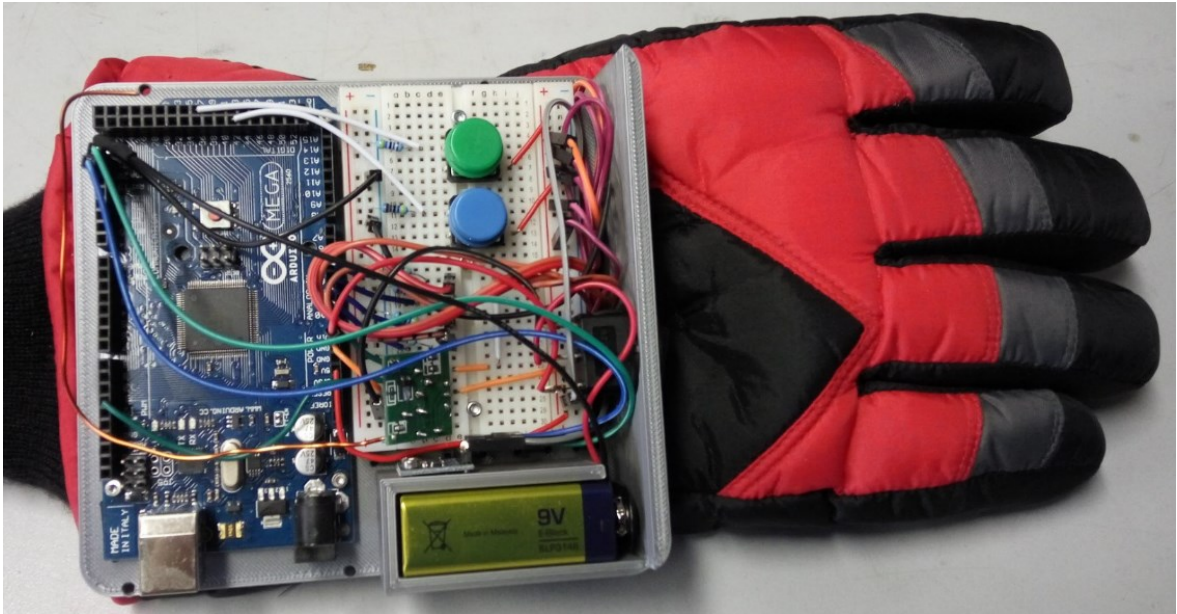
### 10.3 Rukavice pro ovládání

Při realizaci rukavice pro ovládání bylo uvažováno několik způsobů řešení. Jako nejlepším z nich (na základě dostupných materiálů) se ukázalo použití obyčejné zimní rukavice, do které byla zakomponována veškerá elektronika. Za tímto účelem byly vytvořeny další dva 3D modely, které sloužily k umístění napájení, mikropočítače a nepájivého pole na rukavici. Na obr. 44 lze vidět v levé části základnu, která obsahuje prostor pro baterii a otvor pro umístění vypínače. Pravou část obrázku tvoří kryt, ve kterém jsou otvory pro kalibrační tlačítka.



*Obrázek 43 – Model základny a krytu pro umístění rukavici*

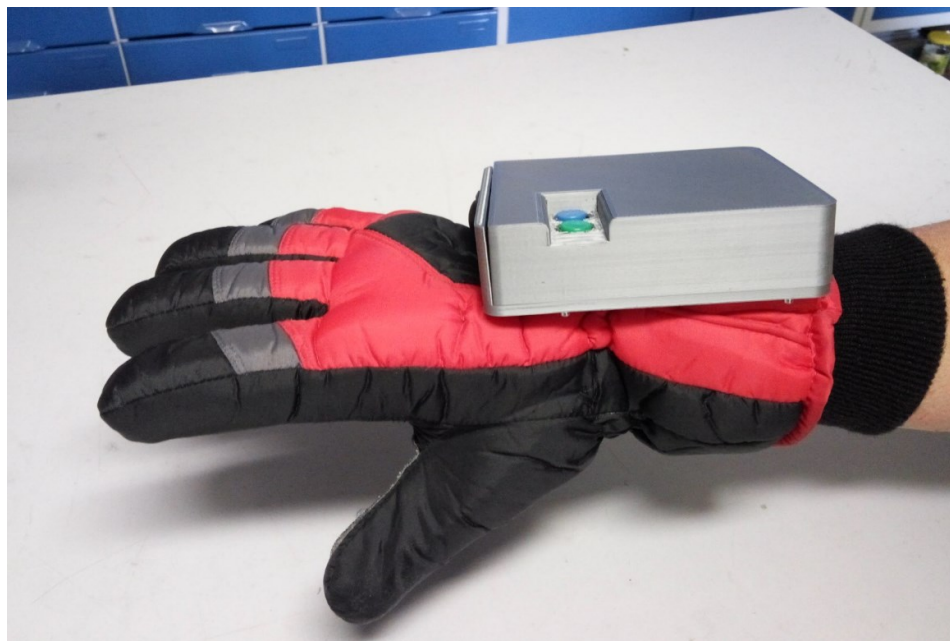
Každý flex senzor byl nejprve upevněn na pásek ocelového plechu délky 150 mm, šířky 10 mm a tloušťky 0,12mm. V plechu byly vyvrtány dírky sloužící k upevnění k rukavici. Tento pásek plechu poskytuje pokrytí většího rozsahu pohybů než samotný flex senzor. Zároveň omezuje poškození flex senzoru proti přílišnému namáhání nebo nevhodnému ohybu. V zimní rukavici se nachází několik vrstev molitanu, které obklopují prsty. Rukavice byla nastřížena u zápěstí (na hřbetu ruky) a mezi vrstvy molitanu byly směrem do prstů vsunuty jednotlivé pásy plechu s flex senzory. K rukavici byly uchyceny přišítkami (za pomoci vyvrtaných děr). Do základny se vhodně vyvrtaly díry se závity M2, které sloužily k uchycení elektroniky, a také díry pro možnost přišítky k rukavici. Dráty od senzorů byly provlečeny nastříženou částí směrem do vytištěné základny. Následně byla základna k rukavici přišita. Nakonec se zapojila elektronika dle schématu a byla zaklopena krytem.



Obrázek 44 – Pohled na otevřenou základnu rukavice

### 10.3.1 Kalibrace

Kalibrace slouží k upravení rozsahu flex senzorů. Pokud by rozsahy byly dány napevno (konstantní), mohlo by dojít k tomu, že lidé s delšími nebo kratšími prsty by měli problém s nepohodlným ovládáním animatronické ruky. Mohlo by se stát, že i přes sevřené prsty v rukavici by animatronická ruka nevykonala celý pohyb (prsty by zůstaly nedovřené). Dále je výhodou také v tom, že si můžeme libovolně nastavit rozsah pohybu prstů. Kalibrace se tedy dělí na dvě části. Kalibrace polohy 1 (zelené tlačítko) a kalibrace polohy 2 (modré tlačítko, viz obr. 45 a 46). Při poloze 1 pak budou všechny servomotory v základní poloze (otevřené prsty). Naopak při poloze 2 budou serva plně otočená (přibližně 160°, zavřené prsty).



*Obrázek 45 – Poloha ruky a prstů pro kalibraci 1*



*Obrázek 46 – Poloha ruky a prstů pro kalibraci 2*

## ZÁVĚR

Hlavním cílem této práce bylo vytvoření funkčního modelu animatronické ruky, která je řízena pomocí bezdrátové komunikace. K ovládání animatronické ruky slouží speciální rukavice.

Teoretická část sloužila čtenáři k seznámení se s animatronikou. Bylo zde vysvětleno, jak funguje jeden z používaných elektromechanických akčních členů – servomotor, který byl také použit při praktickém řešení práce. Tato část také popisovala, jakým způsobem lze servomotory řídit. Byla zde osvětlena další důležitá část animatroniky a robotiky – senzory. Na konci teoretické části byly popsány důležité funkce a vlastnosti mikropočítače ze kterých se vycházelo při řešení praktické části.

Na začátku praktické části se nachází návrh 3D modelu pro animatronickou ruku. Za pomoci programu Autodesk Inventor byly vytvořeny všechny potřebné modely k výsledné realizaci. Vytvořené modely jsou součástí bakalářské práce a lze pomocí nich vytvořit totožný projekt. Dále byla v praktické části vybrána vhodná elektronika. Jako vývojová platforma bylo zvoleno Arduino Mega 2560. Ke snímání pohybu prstů byly vybrány flex senzory a pro snímání úhlu natočení byl zvolen akcelerometr MPU 6050. Za pomoci programu Fritzing byla sestavena elektro schémata, podle kterých proběhlo zapojení. Pro bezdrátovou komunikaci byly vybrány moduly pracující na frekvenci 433 MHz. Aby moduly správně fungovaly, musela být vytvořena dvojice antén, které zvyšují spolehlivost přenosu dat a také dosah. Programy byly napsány v prostředí Arduino IDE jazykem podobným C/C++. Pro fyzickou realizaci byla zvolena technologie 3D tisku, která umožnila výrobu jinak nevyrobitelných dílů.

Při průběžném testování animatronické ruky se objevovaly problémy převážně s akumulátory. Jejich částečné vybití způsobovalo nepředvídatelné chování celého modelu. Komplikace také působily bezdrátové moduly, které bez připojené antény nefungovaly na více jak 10 cm. Problémy byly v průběhu rychle řešeny. Konečný model již nevykazuje chybové chování. Během navrhování elektro schémat proběhlo také testování elektrických obvodů pomocí simulace [circuit.io](https://circuit.io). Poté co byla ověřena správná funkce obvodu, proběhlo fyzické zapojení.

Celá animatronická ruka je nyní jednoduše vyrobitelná za pomoci schémat, programů a přiložených 3D modelů.

**SEZNAM POUŽITÉ LITERATURY**

- [1] Animatronics. *Otis.edu* [online]. [cit. 2017-05-15]. Dostupné z: <https://ospace.otis.edu/ganimatronics/Welcome>
- [2] How Animatronics Works. *Entertainment.howstuffworks.com* [online]. [cit. 2017-05-15]. Dostupné z: <http://entertainment.howstuffworks.com/animatronic1.htm>
- [3] Animatronie aneb jak oživit noční můry. *Aldebaran.cz* [online]. [cit. 2017-05-15]. Dostupné z: [http://www.aldebaran.cz/bulletin/2015\\_08\\_rob.php](http://www.aldebaran.cz/bulletin/2015_08_rob.php)
- [4] Analysis generic architecture hardware. *Sourceforge.net* [online]. [cit. 2017-05-15]. Dostupné z: <https://sourceforge.net/p/fuyard/wiki/analysis-generic-architecture-hardware/>
- [5] PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1.
- [6] NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. Vyd. 1. Praha: BEN - technická literatura, 2004, 247 s. ISBN 80-7300-141-1.
- [7] BARR, Michael a Anthony J. MASSA. Programming embedded systems: with C and GNU development tools. 2nd ed. Sebastopol: O'Reilly, 2006. ISBN 0596009836.
- [8] CATSOULIS, John. Designing Embedded Hardware. Sebastopol: O'Reilly Media, 2005. ISBN 978-0-596-00755-3.
- [9] Pulzně šířková modulace. *Dhservis.cz* [online]. [cit. 2017-05-16]. Dostupné z: <http://www.dhservis.cz/psm.htm>
- [10] HRUŠKA, František. Senzory, e-book, 2011. ISBN 978-80-7454-096-7.
- [11] Diskrétní signál. *Midas.uamt.feec.vutbr.cz* [online]. [cit. 2017-05-16]. Dostupné z: [http://midas.uamt.feec.vutbr.cz/ZVS/Exercise01/content\\_cz.php](http://midas.uamt.feec.vutbr.cz/ZVS/Exercise01/content_cz.php)
- [12] Seznámení s Arduinem. *Arduino.cz* [online]. [cit. 2017-05-16]. Dostupné z: <https://arduino.cz/seznameni-s-arduinem/>
- [13] Krátký přehled Arduino desek. *Laskarduino.cz* [online]. [cit. 2017-05-16]. Dostupné z: <http://blog.laskarduino.cz/kratky-prehled-arduino-desek/>

- [14] Proportions of Hand Segments. *Scielo.cl* [online]. [cit. 2017-05-01]. Dostupné z: <http://www.scielo.cl/pdf/ijmorphol/v28n3/art15.pdf>
- [15] Arduino Mega 2560. *Arduino.cc* [online]. [cit. 2017-05-02]. Dostupné z: <https://www.arduino.cc/en/Main/arduinoBoardMega2560>
- [16] S3003 Servo. *Servocity.com* [online]. [cit. 2017-05-02]. Dostupné z: <https://www.servocity.com/s3003-servo>
- [17] Nine-volt\_battery. *Wikipedia.org* [online]. [cit. 2017-05-02]. Dostupné z: [https://en.wikipedia.org/wiki/Nine-volt\\_battery](https://en.wikipedia.org/wiki/Nine-volt_battery)
- [18] GP Rechargeable NiMH LSD AA 2500 Series 2450 mAh. *Gpbatteries.com* [online]. [cit. 2017-05-02]. Dostupné z: [http://www.gpbatteries.com/int\\_en/2500aa](http://www.gpbatteries.com/int_en/2500aa)
- [19] Battery Holder - 4xAA Square. *Sparkfun.com* [online]. [cit. 2017-05-02]. Dostupné z: <https://www.sparkfun.com/products/552>
- [20] Nepájivé pole 400 kontaktů. *Arduino-shop.cz* [online]. [cit. 2017-05-02]. Dostupné z: <http://arduino-shop.cz/arduino-kabelaz-propoje-rozsireni/1226-nepajive-pole-400-kontaktu-1458404875.html?gclid=CKKn69G2uNACFccy0wodV5oKRw>
- [21] Flex Sensor 2.2". *Sparkfun.com* [online]. [cit. 2017-05-03]. Dostupné z: <https://www.sparkfun.com/products/10264>
- [22] MPU-6050 Accelerometer + Gyro. *Playground.arduino.cc* [online]. [cit. 2017-05-03]. Dostupné z: <http://playground.arduino.cc/Main/MPU-6050>
- [23] 433 MHz vysílač + přijímač. *Arduino-shop.cz* [online]. [cit. 2017-05-03]. Dostupné z: <http://navody.arduino-shop.cz/docs/texty/0/66/1427821401.pdf>
- [24] 433MHz Wireless Transmitter. *Miniinthebox.com* [online]. [cit. 2017-05-03]. Dostupné z: [http://www.miniinthebox.com/433mhz-wireless-transmitter-module-superregeneration-for-arduino-green\\_p411875.html](http://www.miniinthebox.com/433mhz-wireless-transmitter-module-superregeneration-for-arduino-green_p411875.html)
- [25] Antény pro malé bezdrátové systémy. *Pandatron.cz* [online]. [cit. 2017-05-03]. Dostupné z: [http://pandatron.cz/?2888&anteny\\_pro\\_male\\_bezdratove\\_systemy](http://pandatron.cz/?2888&anteny_pro_male_bezdratove_systemy)
- [26] 3D tiskárna Original Prusa i3 MK2S. *Prusa3d.cz* [online]. [cit. 2017-05-03]. Dostupné z: <http://www.prusa3d.cz/>
- [27] Informace o bioplastu PLA. *Eko-plasty.cz* [online]. [cit. 2017-05-03]. Dostupné z: <http://www.eko-plasty.cz/bioplasty-pla/>
- [28] ABS vs. PLA. *Verbatim-europe.cz* [online]. [cit. 2017-05-03]. Dostupné z: <http://www.verbatim-europe.cz/cz/3D/abs-vs-pla/?con=50>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

IR	InfraRed
PWM	Pulse width modulation
3D	Trojdimenzionální
MEMS	Microelectromechanical systems
CISC	Complex Instruction Set Computing
RISC	Reduced Instruction Set Computing
GPIO	General-purpose input/output
I2C	Inter-Integrated Circuit
SDA	Synchronous Clock
SCL	Synchronous Data
m	Metr
ms	Milisekunda
cm	Centimetr
°C	Stupeň Celsia
A/D	Analogově-digitální
mA	Miliampér
V	Volt
mm	Milimetr
MHz	Megahertz
g	gram
mAh	Miliampérhodina
KB	Kilobyte
USB	Universal Serial Bus
GND	Ground



## SEZNAM OBRÁZKŮ

Obrázek 1 – Rozpracovaný model Spinosaura pro film Jurský Park 3 [2].....	12
Obrázek 2 – Blokové schéma vnitřní elektroniky modelářského serva [6].....	13
Obrázek 3 – Princip pulzně-šířkové modulace [9] .....	14
Obrázek 4 – PWM pro řízení servomotoru [4].....	15
Obrázek 5 – Taktilní senzor s využitím piezoelektrického jevu [6].....	17
Obrázek 6 – Princip funkce IR detektoru [10].....	18
Obrázek 7 – Příklad chybného měření pomocí sonaru, vlevo nepřesná naměřená vzdálenost, vpravo nedetekuje sonar překážku [10] .....	18
Obrázek 8 – Operátor s ovladačem pro řízení animatronického modelu [2].....	20
Obrázek 9 – Základní schéma mikropočítače [5] .....	21
Obrázek 10 – Princip vzorkování [11].....	23
Obrázek 11 – Signál po vzorkování a kvantování [11] .....	23
Obrázek 12 – Začátek komunikace I2C [7].....	24
Obrázek 13 – Vyčtení bitu z vodiče SDA pomocí náběžné hrany SCL [7] .....	24
Obrázek 14 – Ukončení komunikace I2C [7] .....	25
Obrázek 15 – Rentgenový snímek ruky s popisem jednotlivých kostí, I – palec, II – ukazováček, III – prostředníček, IV – prsteníček, V – malíček; tip – jemná tkáň špičky prstu; pd – distal phalanx; pm – medial phalanx; pp – proximal phalanx; m – metacarpal [14].....	29
Obrázek 17 – Model prstu (ukazováček) v otevřeném stavu.....	30
Obrázek 18 – Model prstu (ukazováček) v sevřeném stavu .....	31
Obrázek 19 – Model palce a mezikusu v obecné poloze.....	31
Obrázek 20 – Model dlaně se zápěstím, čárkovaně vyznačené skryté hrany kanálků uvnitř dlaně.....	32
Obrázek 21 – Modely pohyblivé části, nalevo s rozevřenými prsty, napravo se sevřenými .....	33
Obrázek 22 – Model základny .....	34
Obrázek 23 – Detail modelu základny, vzniklý prostor pro baterie a elektroniku .....	35
Obrázek 24 – Kompletní model robotické ruky bez krytu .....	35
Obrázek 25 – Vývojový kit Arduino Mega 2560 [15].....	36
Obrázek 26 – Servopohon Futaba S3003 [16].....	37
Obrázek 27 – Názorný model servomotoru s upevněnou řemenicí.....	38

Obrázek 28 – Konektor pro 9V baterii [17].....	39
Obrázek 29 – Bateriový box pro 4xAA baterie [19] .....	40
Obrázek 30 – Kontaktní nepájivé pole [20].....	40
Obrázek 31 – Schéma zapojení mikropočítače, servopohonů a napájení, vytvořeno v programu Fritzing.....	41
Obrázek 32 – Flex senzor firmy Sparkfun 2,2“ [21] .....	43
Obrázek 33 –Modul MPU-6050 [22].....	44
Obrázek 34 – Schéma zapojení elektroniky v rukavici pro ovládání, vytvořeno v programu Fritzing.....	45
Obrázek 35 – Modul vysílače 433,92 MHz [24] .....	47
Obrázek 36 – Schéma zapojení vysílače k desce Arduina (rukavice pro ovládání).....	48
Obrázek 37 – Modul přijímače 433,92 MHz [24] .....	49
Obrázek 38 – Schéma zapojení vysílače k desce Arduina (animatronická ruka).....	49
Obrázek 39 – Vývojový diagram programu .....	50
Obrázek 40 – Tiskárna Prusa i3 MK2 [26].....	54
Obrázek 41 – Výsledné provedení animatronické ruky, bez krytu.....	56
Obrázek 42 – Umístění elektroniky .....	57
Obrázek 43 – Pohled na baterie, přístupnost ze spodní strany .....	57
Obrázek 44 – Model základny a krytu pro umístění rukavici .....	58
Obrázek 45 – Pohled na otevřenou základnu rukavice.....	59
Obrázek 46 – Poloha ruky a prstů pro kalibraci 1 .....	60
Obrázek 47 – Poloha ruky a prstů pro kalibraci 2 .....	60

**SEZNAM TABULEK**

Tabulka 1 – Druhy Arduino desek a jejich parametry [13] .....	25
Tabulka 2 - Rozměry kostí v dlani a prstech, I – palec, II – ukazováček, III – prostředníček, IV – prsteníček, V – malíček [14] .....	29
Tabulka 3 – Parametry kitu Arduino Mega 2560 [15].....	36
Tabulka 4 – Parametry servopohonu Futaba S3003 [16] .....	37
Tabulka 5 – Naměřené a vypočítané hodnoty pro řemenice .....	39
Tabulka 6 – Parametry flex senzoru Sparkfun 2,2“ [21] .....	43
Tabulka 7 – Parametry modulu MPU-6050 [22] .....	44
Tabulka 8 – Parametry vysílacího modulu [23].....	47
Tabulka 9 – Parametry přijímacího modulu [23].....	49

## SEZNAM PŘÍLOH

- P I Kód programu rukavice
- P II Kód programu animatronické ruky
- P III Elektronické přílohy na CD

## **PŘÍLOHA P I: KÓD PROGRAMU RUKAVICE**

```
#include <VirtualWire.h> // import knihoven
#include "I2Cdev.h"
#include "MPU6050.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    #include "Wire.h"
#endif

MPU6050 accelgyro; //vytvoření proměnné akcelerometru
int16_t ax, ay, az; //proměnné ve třech osách akcelerometr
int16_t gx, gy, gz; //proměnné ve třech osách gyroskop
int TX_PIN = 11; // určení pinu pro odesílání dat (bezdrátová komunikace)

typedef struct dataToSend // definování struktury s daty k odeslání
{
    int Sensor1Data;
    int Sensor2Data;
    int Sensor3Data;
    int Sensor4Data;
    int Sensor5Data;
    int Sensor6Data;
};

struct dataToSend data;
//určení analogových pinů
int flexInput1 = A0;
int flexInput2 = A1;
int flexInput3 = A2;
int flexInput4 = A3;
int flexInput5 = A4;
//pomocné proměnné hodnoty analogových vstupů
int analogFlex1 = 0;
int analogFlex2 = 0;
int analogFlex3 = 0;
int analogFlex4 = 0;
```

```
int analogFlex5 = 0;
//pomocné proměnné s úhlem natočení serva
int servoAngle1 = 0;
int servoAngle2 = 0;
int servoAngle3 = 0;
int servoAngle4 = 0;
int servoAngle5 = 0;
int servoAngle6 = 0;
//proměnné sloužící k aktualizování max a min hodnot (využití u kalibrace)
int flexMax1 = 0;
int flexMax2 = 0;
int flexMax3 = 0;
int flexMax4 = 0;
int flexMax5 = 0;
int rotateMin = 0;
int flexMin1 = 0;
int flexMin2 = 0;
int flexMin3 = 0;
int flexMin4 = 0;
int flexMin5 = 0;
int rotateMax = 0;
//určení pinu s tlačítka
int buttonCalibMax = 37;
int buttonCalibMin = 42;
int buttonStateMax = 0;
int buttonStateMin = 0;

void setup() {
  //nastavení pinů jako vstupů
  pinMode(flexInput1, INPUT);
  pinMode(flexInput2, INPUT);
  pinMode(flexInput3, INPUT);
  pinMode(flexInput4, INPUT);
  pinMode(flexInput5, INPUT);
```

```
pinMode(buttonCalibMax, INPUT);
pinMode(buttonCalibMin, INPUT);

//počáteční určení max a min hodnot
flexMin1 = 430;
flexMin2 = 430;
flexMin3 = 430;
flexMin4 = 430;
flexMin5 = 430;
rotateMin = -14000;
flexMax1 = 600;
flexMax2 = 600;
flexMax3 = 600;
flexMax4 = 600;
flexMax5 = 600;
rotateMax = 14000;

vw_setup(2000); // Nastaví a spustí komunikaci pomocí modulu, rychlost 2000b/s
vw_set_tx_pin(TX_PIN); // určení komunikačního pinu

// nastavení I2C sběrnice
#ifdef I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
    Wire.begin();
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
    Fastwire::setup(400, true);
#endif

accelgyro.initialize(); // inicializace modulu akcelerometru/gyroskopu

// nastavení hodnot odchylek akcelerometru/gyroskopu, lze zjistit kalibrací
accelgyro.setXGyroOffset(80);
accelgyro.setYGyroOffset(6);
accelgyro.setZGyroOffset(-13);
```

```

    accelgyro.setXAccelOffset(-4195);
    accelgyro.setYAccelOffset(1401);
    accelgyro.setZAccelOffset(1316);

}

void loop() {

    analogFlex1=analogRead(flexInput1); //přečtení hodnot na analogovém vstupu (A0) a
    uložení do proměnné
    //zpracování hodnot (pokud dojde k překročení hranice intervalu max min, je hodnota
    oříznuta na max nebo min hodnotu)
    if(analogFlex1<flexMin1)
    {
        analogFlex1=flexMin1;
    }
    if(analogFlex1>flexMax1)
    {
        analogFlex1=flexMax1;
    }
    servoAngle1=map(analogFlex1, flexMin1, flexMax1,0,1500); //mapování naměřené
    hodnoty na interval 0-1500 (slouží k ovládání serva)
    delay(2); //pauza pro ustálení A/D převodníku

    //stejný postup pro všechny flex senzory
    analogFlex2=analogRead(flexInput2);
    if(analogFlex2<flexMin2)
    {
        analogFlex2=flexMin2;
    }
    if(analogFlex2>flexMax2)
    {
        analogFlex2=flexMax2;
    }
    servoAngle2=map(analogFlex2, flexMin2, flexMax2,0,1500);
    delay(2);
}

```



```
analogFlex3=analogRead(flexInput3);
if(analogFlex3<flexMin3)
{
  analogFlex3=flexMin3;
}
if(analogFlex3>flexMax3)
{
  analogFlex3=flexMax3;
}
servoAngle3=map(analogFlex3, flexMin3, flexMax3,0,1500);
delay(2);
```

```
analogFlex4=analogRead(flexInput4);
if(analogFlex4<flexMin4)
{
  analogFlex4=flexMin4;
}
if(analogFlex4>flexMax4)
{
  analogFlex4=flexMax4;
}
servoAngle4=map(analogFlex4, flexMin4, flexMax4,0,1500);
delay(2);
```

```
analogFlex5=analogRead(flexInput5);
if(analogFlex5<flexMin5)
{
  analogFlex5=flexMin5;
}
if(analogFlex5>flexMax5)
{
```

```

    analogFlex5=flexMax5;
}
servoAngle5=map(analogFlex5, flexMin5, flexMax5,0,1500);
delay(2);

//zpracování hodnot modulu MPU6050, stejné jako u flex senzoů
if(ay<rotateMin)
{
    ay=rotateMin;
}
if(ay>rotateMax)
{
    ay=rotateMax;
}
servoAngle6=map(ay, rotateMin, rotateMax,0,1500);

//načtení potřebných dat do struktury k přenesení
data.Sensor1Data = servoAngle1;
data.Sensor2Data = servoAngle2;
data.Sensor3Data = servoAngle3;
data.Sensor4Data = servoAngle4;
data.Sensor5Data = servoAngle5;
data.Sensor6Data = servoAngle6;

vw_send((uint8_t *)&payload, sizeof(payload)); // odeslání dat
vw_wait_tx();// čekání na dokončení odesílání

buttonStateMax = digitalRead(buttonCalibMax);//kontrola stisku tlačítka pro max
buttonStateMin = digitalRead(buttonCalibMin);//kontrola stisku tlačítka pro min

//čtení okamžitých hodnot na akcelerometru/gyroskopu
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

```

*//pokud bylo zmáčknuto tlačítko max, provede se kalibrace max, určí se nové maximální hodnoty z naměřených*

```
if(buttonStateMax == HIGH){  
flexMax1=analogRead(flexInput1);  
delay(2);  
flexMax2=analogRead(flexInput2);  
delay(2);  
flexMax3=analogRead(flexInput3);  
delay(2);  
flexMax4=analogRead(flexInput4);  
delay(2);  
flexMax5=analogRead(flexInput5);  
delay(2);  
rotateMax=ay;  
delay(2);  
}
```

*//pokud bylo zmáčknuto tlačítko min, provede se kalibrace min, určí se nové minimální hodnoty z naměřených*

```
if(buttonStateMin == HIGH){  
flexMin1=analogRead(flexInput1);  
delay(2);  
flexMin2=analogRead(flexInput2);  
delay(2);  
flexMin3=analogRead(flexInput3);  
delay(2);  
flexMin4=analogRead(flexInput4);  
delay(2);  
flexMin5=analogRead(flexInput5);  
delay(2);  
rotateMin=ay;  
delay(2);  
}  
}
```

## PŘÍLOHA P II: KÓD PROGRAMU ANIMATRONICKÉ RUKY

```
#include <ServoTimer2.h> // import knihoven
#include <VirtualWire.h>

int RX_PIN = 11; // určení pinu pro přijetí dat (bezdrátová komunikace)

typedef struct dataToSend // definování struktury s daty k odeslání
{
    int  Sensor1Data;
    int  Sensor2Data;
    int  Sensor3Data;
    int  Sensor4Data;
    int  Sensor5Data;
    int  Sensor6Data;
};

struct dataToSend data;
uint8_t sizeOfData = sizeof(dataToSend); // proměnná k určení velikosti přenášené zprávy
// vytvoření proměnných servomotorů
ServoTimer2 myServo1;
ServoTimer2 myServo2;
ServoTimer2 myServo3;
ServoTimer2 myServo4;
ServoTimer2 myServo5;
ServoTimer2 myServo6;
// přiřazení čísel pinů k proměnným
int servoPin1 = 2;
int servoPin2 = 3;
int servoPin3 = 4;
int servoPin4 = 5;
int servoPin5 = 6;
int servoPin6 = 7;
// pomocné proměnné s úhly natočení serv
```

```

int servoAngle1 = 0;
int servoAngle2 = 0;
int servoAngle3 = 0;
int servoAngle4 = 0;
int servoAngle5 = 0;
int servoAngle6 = 0;

void setup() {
    //nastavení příslušných pinů k servomotorům
    myServo1.attach(servoPin1);
    myServo2.attach(servoPin2);
    myServo3.attach(servoPin3);
    myServo4.attach(servoPin4);
    myServo5.attach(servoPin5);
    myServo6.attach(servoPin6);

    vw_setup(2000); // Nastaví a spustí komunikaci pomocí modulu, rychlost 2000b/s
    vw_set_rx_pin(RX_PIN); // určení komunikačního pinu
    vw_rx_start(); //začátek komunikace

}

void loop() {
    vw_wait_rx(); //čekání na příchozí zprávu
    vw_get_message((uint8_t *)&data, &sizeOfData); //uložení příchozích dat do vytvořené
    struktury

    myServo1.write(2250-receivedData.Sensor1Data); //nastav servo na požadovaný uhel
    myServo2.write(2250-receivedData.Sensor2Data);
    myServo3.write(2250-receivedData.Sensor3Data);
    myServo4.write(2250-receivedData.Sensor4Data);
    myServo5.write(2250-receivedData.Sensor5Data);
    myServo6.write(2250-receivedData.Sensor6Data);

}

```