



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název: Robotická ruka
Student: Ilia Popov
Vedoucí: Ing. Miroslav Skrbek, Ph.D.
Studijní program: Informatika
Studijní obor: Počítačové inženýrství
Katedra: Katedra číslicového návrhu
Platnost zadání: Do konce zimního semestru 2020/21

Pokyny pro vypracování

Seznamte se s modelem lidské ruky, který je schopný elektronicky ovládat všech pět prstů.

Prozkoumejte jeho mechanické možnosti formou jednoduchých testů.

Navrhňte a realizujte programové vybavení, které bude schopné v robotické ruce uchopit předmět a držet jej.

Určete, jaké typy předmětů půjde uchopit. Dále propojte ruku se svalovým senzorem, případně odporovými ohybovými senzory a demonstруйте ovládání robotické ruky pohybem lidské paže.

Vytvořte demo aplikaci.

Vyvinuté programové vybavení řádně zdokumentujte.

Rozsah práce upřesněte po dohodě s vedoucím práce.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Hana Kubátová, CSc.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 14. února 2019



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

Bakalářská práce

Robotická ruka

Katedra číslicového návrhu

Vedoucí práce: Ing. Miroslav Skrbek, Ph.D.

4. června 2020

Poděkování

Chtěl bych poděkovat panu Ing. Miroslavu Skrbkovi, Ph.D. za vedení, trpělivost a praktickou pomoc v rámci této bakalářské práce. Dále děkuji Bc. Petrovi Gulovi za technickou pomoc v laboratoři, své kamarádce Janě za rady v české gramatice a své rodině za podporu a pomoc při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mé práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 4. června 2020

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2020 Ilija Popov. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Popov, Ilija. *Robotická ruka*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Cílem této práce bylo prozkoumat model robotické ruky od společnosti YouBionic a následně navrhnout a implementovat programové vybavení pro ovládání.

V práci je podrobně popsán původní produkt, který byl pak modernizován jak hardwarově tak i softwarově. Pro ovládání modelu pohybem paže uživatele byla vytvořena rukavice se senzory ohybu.

Také bylo v práci implementováno programové řešení pro řídicí jednotku robotické ruky a vytvořena PC aplikace v jazyce C pro ovládání a demonstraci možností modelu.

Klíčová slova Robotická ruka, YouBionic Hand, Arduino, servomotor, Ohybový senzor, programové vybavení

Abstract

The goal of this thesis was to study a model of a robotic hand from the company YouBionic and then to design and implement software to control this product.

This thesis in detail describes the original product, which was then upgraded in terms of both hardware and software. A glove with flex sensors was created to control the model by moving the user's arm.

Software solution involved creating script for the control unit of the robotic hand and PC application in C language to control and demonstrate its capabilities.

Keywords Robotic hand, YouBionic Hand, Arduino, servo, humanoid, Flex sensor, software

Obsah

Úvod	1
1 Cíl práce	3
2 Rešerše	5
2.1 YouBionic Hand	5
2.2 Lidská ruka	12
3 Testování možností modelu	15
4 Analýza a návrh	17
4.1 Polohy prstů robotické ruky	17
4.2 Komunikace mezi deskami Arduino	18
4.3 Řízení modelu	20
4.4 Připojení modelu a PC	22
4.5 Ovládání modelu pomocí senzoru	22
4.6 Aplikace	23
5 Realizace	25
5.1 Připojení a komunikace mezi komponenty	25
5.2 Rukavice	28
5.3 Řídicí programy	30
5.4 Knihovna	34
5.5 Aplikace	35
6 Testování řešení	39
6.1 Testování robotické ruky	39
6.2 Testování aplikací a rukavice	40
Závěr	41

Literatura	43
A Uživatelská příručka	45
A.1 Připojení komponentů	45
A.2 Funkce knihovny	47
B Fotografie z testování	49
B.1 Testování možností modelu	49
B.2 Testování řešení	51
C Seznam použitých zkratk	55
D Obsah příloženého CD	57

Seznam obrázků

2.1	Youbionic Hand. Převzato z [1].	6
2.2	Originální schéma zapojení. Převzato z [1].	7
2.3	Arduino IDE	8
2.4	Arduino Micro	9
4.1	Komunikace komponent robotické ruky	18
4.2	SPI komunikace	19
4.3	ICSP	19
4.4	Řízení modelu	21
4.5	Regulace modelu	22
4.6	Navrhované schéma modelu	23
4.7	První obrazovka aplikace	23
5.1	Port potenciometrů	25
5.2	Formát příkazu	26
5.3	Schéma zapojení komponentů	27
5.4	Sestavený model ruky	28
5.5	Rukavice	28
5.6	Připojení senzoru ohybu	29
5.7	Schéma zapojení rukavice	30
5.8	Program pro hlavní řídicí jednotku robotické ruky	32
5.9	Program pro externí desku	33
5.10	Program pro ovládací rukavice	34
5.11	Obrazovka ovládání přes rukavice	36
5.12	Obrazovka manuálního ovládání	37
5.13	Obrazovka přednastavených stavů	38
A.1	Výstupy servomotorů	45
A.2	Připojení desek	46
A.3	Připojení robotické ruky	47

Seznam tabulek

2.1	Technické informace	9
2.2	Specifikace servopohonu	10
2.3	Specifikace pinů	10
2.4	Konfigurace pinů	11
3.1	Výsledky testů	16
4.1	Umístění linek SPI	19
5.1	Připojení potenciometrů	25
5.2	Slave odesílání	26
5.3	Připojení prstů rukavice	29
5.4	Zakladní funkce knihovny	35

Úvod

Robotika je dnes jednou z nejrychleji rostoucích oblastí. Součástí této oblasti je vývoj robotických ruk, které se v poslední době aktivně rozvíjejí. V podstatě jakékoli zařízení používané k pohybu objektů lze označit za robotickou ruku. Výhodou podobných zařízení je vysoká přesnost, proto se robotické ruce aktivně používají v průmyslu.

Další oblast je medicína. Robotické ruce se používá v chirurgii a jako protézy lidských končetin. Ve vývoji robotických ruk došlo k obrovskému pokroku díky rozvoji 3D tisku, který mnohokrát zlevnil výrobu podobných projektů.

Mechanický model je pouze jednou částí těchto projektů, druhou částí je software. Model musíme naprogramovat, abychom jej mohli podle potřeby ovládat. V případě modelu vyrobeného na základě lidské končetiny by pohyby modelů měly odpovídat pohybům lidské ruky.

Hlavním cílem této práce je prozkoumat model lidské ruky od firmy YouBionic, navrhnout a realizovat programové vybavení, pro ovládání tohoto zařízení. Toto téma jsem si zvolil, protože je to pro mě příležitost rozšířit své znalosti o robotice a programování. Projekt YouBionic Hand navíc neobsahuje plně vybavené řídicí řešení a je vhodný pro rozšíření. Díky použití Arduino pro řízení modelu je projekt otevřen pro implementace vlastních softwarových řešení.

V této práci se zabývám analýzou modelu robotické ruky, úpravou hardwarové části projektu, realizací ovládání modelu pohybem ruky uživatele a implementací demo aplikace, která dokáže ovládat celý model.

Cíl práce

Cílem této bakalářské práce je navrhnout a implementovat funkční programové vybavení pro ovládání robotické ruky. Dalším požadavkem je pomoci tohoto vybavení prokázat ovládání modelu pohybem lidské paže.

Práce se zaměřuje na analýzu modelu lidské ruky od společnosti YouBionic, která je schopna elektronicky ovládat všech pět prstů. Cílem je seznámit se s modelem a popsat jeho jednotlivé části. Cílem teoretické části je také prozkoumat pohyby lidské ruky pro jejich následnou implementaci do modelu.

Cílem praktické části práce je prozkoumat mechanické možnosti robotické ruky formou jednoduchých testů. Na základě testů provedených na tomto modelu a výsledku rešeršní části navrhnout změny v původním modelu a softwarové řešení pro řízení modelu. Následně implementovat změny v modelu a vytvořit aplikaci pro ovládání modelu.

Rešerše

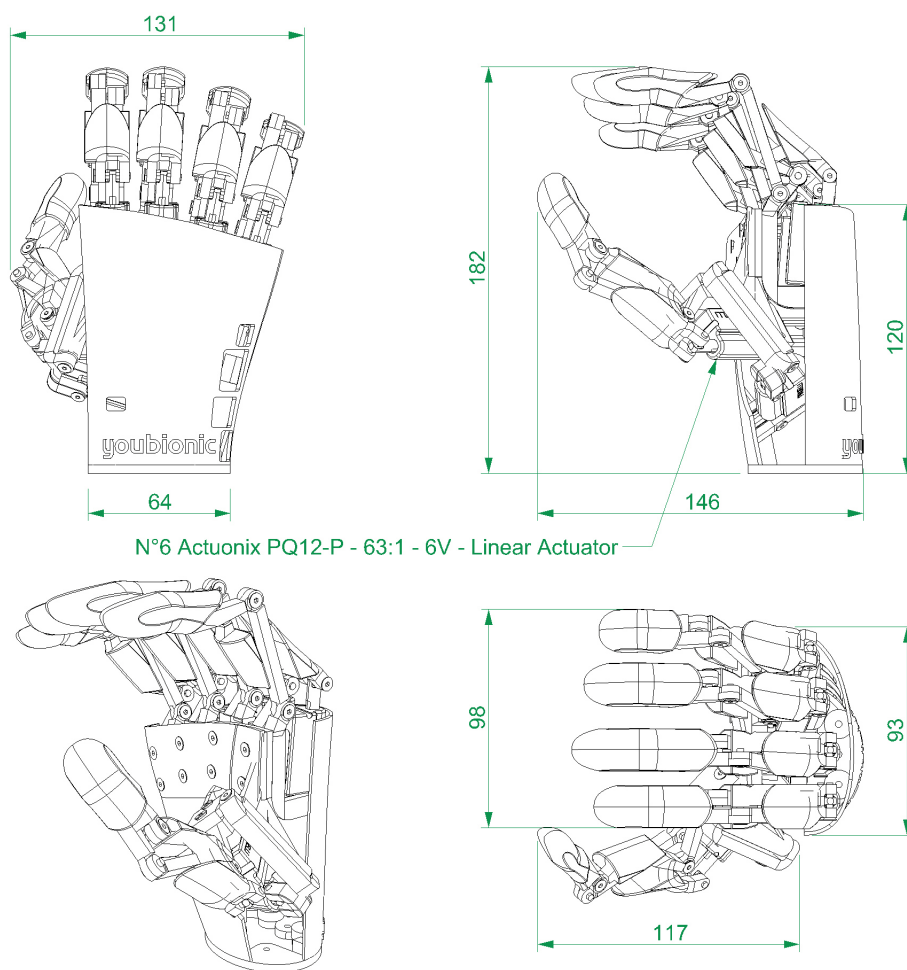
Následující část popisuje model lidské ruky od firmy YouBionic a její součásti. Provedeno testování možností modelu a prozkoumaný způsoby uchopení lidské ruky.

2.1 YouBionic Hand

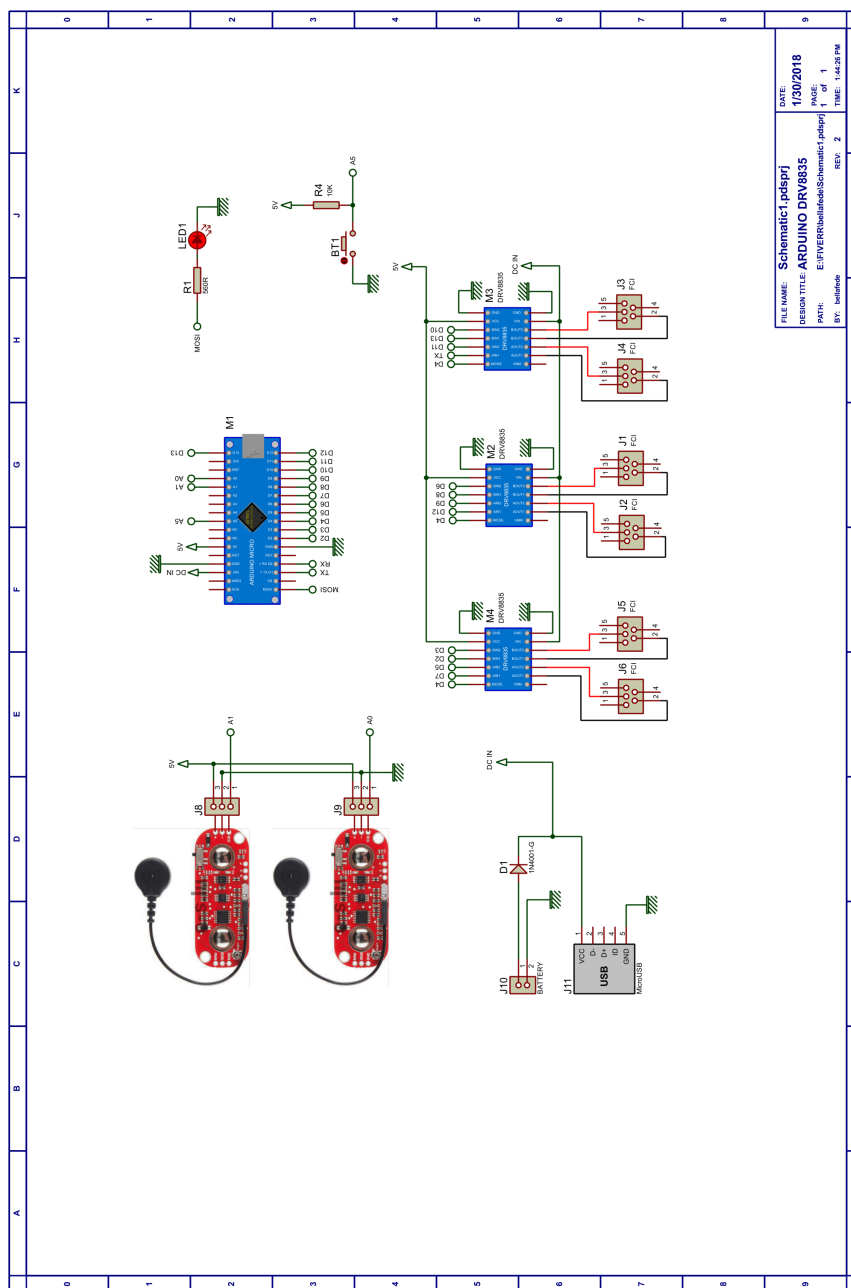
Youbionic hand (2.1) – vytištěná na 3D tiskárně bionická ruka od firmy YouBionic. Tato robotická ruka je jedna z nejlevnějších na trhu. Navíc modelem ovládá jednodeskový počítač Arduino, který je jednoduchý k programování. Kombinace těchto faktorů dělá projekt Youbionic hand optimálním řešením pro výuku a integrace do jiných projektů.

Základem modelu je plastové pouzdro vyrobené ve tvaru lidské ruky. Ke každému prstu ruky připojen jeden servomotor pro realizace pohybu a jeden servomotor pro otáčení palce. Na zadní straně dlaně umístěna deska s Arduinem na kterou jsou připojeny všechny komponenty ruky. Součástí modelů a jejich funkce jsou popsány v podkapitole 2.1.1. Projekt navíc obsahuje schéma připojení pro všechny komponenty (2.2) a základní softwarové řešení pro ovládání ruky. Analýza a testování původního programového řešení jsou v podkapitole 2.1.2.

2. REŠERŠE



Obrázek 2.1: Youbionic Hand. Převzato z [1].



Obrázek 2.2: Originální schéma zapojení. Převzato z [1].

2.1.1 Komponenty

Tato podkapitola podrobně popisuje hlavní součásti projektu YouBionic Hand, jejich možnosti, vlastnosti a funkce v modelu.

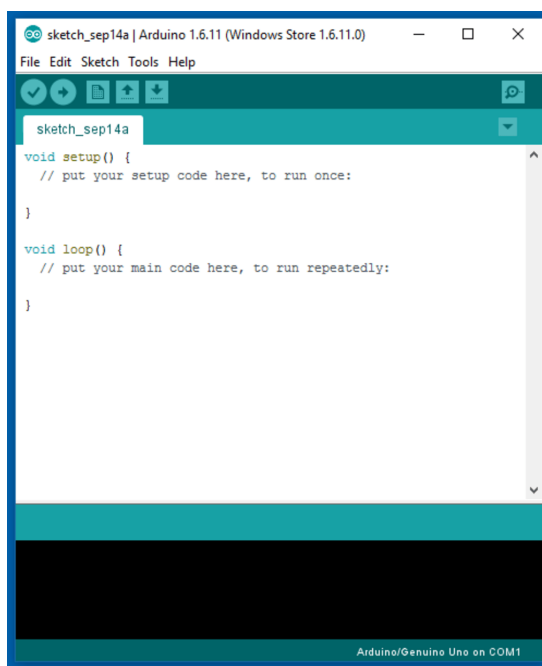
Hlavní řídicí jednotkou projektu je Arduino Micro, které je k dispozici pro

2. REŠERŠE

programování. Servomotory které provádí fyzické pohyby rukou jsou připojeny k hlavnímu ovladači. Servomotory jsou připojeny k Arduino prostřednictvím dvoumotorových budičů, které umožňují řídit servomotory. Všechny jednotky modelu jsou připojena k jedné desce, k dispozici je pouze port USB pro programování Arduino a několik výstupních pinů pro připojení externích senzorů. To znamená, že jakékoli změny v původním schématu budou vyžadovat pájení součástí a změny v pouzdru zařízení.

2.1.1.1 Arduino Micro

Arduino je otevřená elektronická platforma. Výhodou platformy je flexibilní software a hardware Arduino, díky kterému je platforma jednoduché konfigurovatelná pro různé projekty. Základním principem zařízení je zpracování vstupních signálů (např. tlakový senzor, tlačítko, atd.). Deska vstupů je pak schopná zpracovávat a ovládat různé výstupy (rozsvícení LED, spuštění motoru, zobrazení textu na obrazovku, atd.). K programování desek Arduino se používá software Arduino IDE (Vývojové prostředí) (Obrázek 2.3). Programovacím jazykem pro desky Arduino je Wiring, postavený na C a C++. Software je k dispozici na adrese *arduino.cc*. Programy se zapisují na desku pomocí kabelu USB, přes který je deska připojena k počítači.



Obrázek 2.3: Arduino IDE

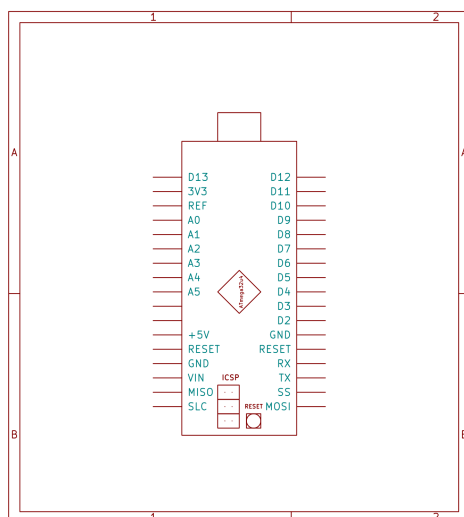
V modelu robotické ruky byl použit ovladač Arduino Micro. „*Arduino Micro obsahuje 20 digitálních I/O pinů (z nichž 7 je možné použít jako PWM*

(Pulzně Šířková Modulace) výstup a 12 jako analogový vstup), 16MHz krystalový oscilátor, micro USB konektor, ICSP (In-Circuit Serial Programming) čtečku a resetovací tlačítko“[2]. V Tabulce 2.1 jsou uvedeny specifikace desky Arduino Micro.

Tabulka 2.1: Technické informace

Mikroprocesor	ATmega32U4
Provozní napětí (logická úroveň)	5 V
Vstupní napětí (doporučeno)	7-12 V
Vstupní napětí (maximální limit)	6-20 V
Počet digitálních I/O pinů	20 pinů, z toho 7 s PWM
Počet analogových vstupů	12 pinů
Proudové zatížení na 1 pin	20 mA
Flash paměť	32 KB
SRAM	2,5 KB
EEPROM	1 KB
Rychlost hodin	16 MHz

Arduino Micro (2.4) je hlavním řídicím mikrokontrolérem pro celý projekt. Díky malé velikosti zařízení a dostatečnému množství vstupu je tato verze řídicí desky optimálním řešením pro robotickou ruku. Tento ovladač spojuje a řídí všechny části modelu, řídí servomotory a komunikuje s externími jednotkami (senzory, počítač, atd.). V původním projektu Arduino Micro obsahuje několik volných, přístupových pinů, které lze použít k rozšíření funkčnosti modelu.



Obrázek 2.4: Arduino Micro

2.1.1.2 Servomotor

Pro mechanické pohyby všech pěti prstů robotické ruky se používají servomotory. „*Servo motory slouží pro nastavení určité polohy ovládaného mechanismu a následné držení v této poloze. Stejnoseměrné servo motory se využívají například pro ovládání robotické paže nebo pro nastavení kormidla u leteckých modelů. Jejich hlavní výhodou je malý rozměr a malá hmotnost s relativně velkou silou*“[3].

Projekt Youbionic Hand obsahuje šest servopohonů typu PQ12-63-6-P od společnosti Actuonix Motion Devices Inc. Jeden servopohon pohon pro každý prst a jeden pro otáčení palce.

Servopohony PQ12 jsou pohybové přístroje s polohovou zpětnou vazbou pro řízení polohy. Pohony lze ovládat pomocí stejnosměrného napětí, aby se pohon prodloužil. Změna polarity provede sevržení servopohonu.[4]. Technické vlastnosti servomotoru použitého v modelu jsou uvedeny v Tabulce 2.2. Použité servopohony obsahují potenciometr polohové zpětné vazby. Potenciometr neobsahuje zabudovaný budič, ale poskytuje analogovou zpětnou vazbu. Tuto zpětnou vazbu lze použít k sledování aktuální polohy servopohonu, což je užitečné v modelu robotické ruky pro ovládání polohy prstu. Připojení pinů potenciometru je uvedeno v Tabulce 2.3. V počátečním provedení není popsán potenciometr připojen, což neumožňuje řídit stav servomotoru.

Tabulka 2.3: Specifikace pinů

PIN	Význam
Pin 1	Zpětná vazba potenciometru negativní referenční kolejnice
Pin 2	Napájení pohonu
Pin 3	Napájení pohonu
Pin 4	Zpětná vazba potenciometru pozitivní referenční kolejnice
Pin 5	Potenciometr zpětné vazby

Tabulka 2.2: Specifikace servopohonu

Vlastnosti	Hodnota
Převodový poměr	63
Napětí	6

2.1.1.3 Budiče pro motory

Servomotory použité v modelu neobsahují integrované ovladače. Projekt proto používá ovladače DRV8835 od společnosti Texas Instruments. Ke každému regulátoru jsou připojeny dva pohony. V rámci projektů používané řídicí rozhraní ovladače je PHASE/ENABLE. Rozhraní je zvoleno prostřednictvím pinu Mode. V tomto rozhraní jeden pin je určený pro směr otáčení (AIN1/BIN1) a druhý pin pro rychlost (AIN2/BIN2)[5]. Pro ovládání dvou servomotorů jsou ke každému regulátoru připojeny čtyři výstupní signály Arduino (dva pro každý motor). Tato konfigurace řídicí jednotky servomotoru neobsahuje přiřazenou zpětnou vazbu z pohonu, tj. v původním modelu není možné nastavit servomotor (nebo prst robotické ruky) do konkrétní polohy.

2.1.2 Programové řešení

Projekt YouBionic Hand obsahuje základní řešení pro řízení modelu pomocí svalového senzoru MyoWare.

Svalový senzor měří aktivaci svalů elektrickým potenciálem. Senzor MyoWare má navíc analogový výstup v rozsahu 0-1023[6]. Díky této funkci je tento senzor snadno kompatibilní s Arduino. Připojením tohoto senzoru k Arduino v ruce můžeme zkontrolovat původní řešení pro ovládání modelu. Robotická ruka reaguje na stisknutí a uvolnění ruky uživatele. Na základě pohybu, všechny prsty v modelu se otevírají/zavírají. V tomto řešení není implementováno ovládání různých prstů zvlášť a otáčení palce.

Výrobce poskytuje kód pro své řešení. Je to jediná softwarová část projektů, která v době psaní práce je k dispozici. Softwarovou součástí projektu je skript napsaný pro Arduino Micro. Můžeme analyzovat tuto část projektu, testováním původního programu.

Arduino řídí servomotory prostřednictvím výstupních signálů připojených k budičům servopohonů. K ovládání každého servomotoru se používají dva výstupní signály. První označuje směr otáčení pohonu. Druhý signál nastavuje rychlost otáčení. Počáteční konfigurace pinů jsou uvedeny v Tabulce 2.4. V kódu také se konfiguruje budiče servopohonů. Budiče jsou nastaveny do režimu PHASE/ENABLE. Pro ovládání dvou pohonů jedním komponentem. Pro řízení motorů v kódu aplikace jsou použity funkce z Ukázky 2.1, kde *direction* je směr motoru (LOW - otevírání, HIGH - zavírání), *speed* je rychlost otáčení.

Tabulka 2.4: Konfigurace pinů

Proměnná	PIN	Ovládaný prst
pinDirA	2	Ukazovác
pinPwmA	3	
pinDirB	7	Prostředník
pinPwmB	5	
pinDirC	8	Prsteník
pinPwmC	6	
pinDirD	12	Malíček
pinPwmD	9	
pinDirE	13	Palec
pinPwmE	10	
pinDirF	1	Palcové otáčení
pinPwmF	11	

Ukázka kódu 2.1: Funkce pro ovládání motoru

```
digitalWrite( pinDir,direction );  
analogWrite( pinPwm,speed );
```

Procesy otevírání a zavírání celé ruky jsou řízeny pomocí vstupního signálu ze svalového senzoru. Svalový senzor připojen na pin A1 a čtení signálů z tohoto pinu realizované pomocí funkce: `analogRead(pin)`. Pro správné zpracování signálů se provádí fáze kalibrace senzoru, fáze kalibrace vyžaduje od uživatele používat svaly, na které je senzor přiřazen. Fáze kalibrace trvá 10 vteřin, výsledkem této fáze jsou minimální a maximální hodnoty naměřené snímačem. Hlavní funkcí programu je cyklus, ve kterém aplikace sleduje aktivitu svalů uživatele a na základě výstupu snímače provádí pro otevření a zavření robotické ruky.

YouBionic Hand nemá knihovnu ani dokumentaci popisující funkci pro ovládání modelu. Analyzovány v této podkapitole program je částečně popsán pomocí komentářů. Ten program je jediné softwarové řešení, které společnost YouBionic poskytuje pro svůj výrobek. Toto řešení je použitelné pro testování možností modelu a části kódu mohou být použity pro návrh softwarového vybavení v rámci práce.

2.2 Lidská ruka

Předmětem bakalářské práce je robotická ruka vytvořená ve tvaru lidské ruky. Mechanické pohyby modelu proto musí odpovídat pohybu lidské ruky. Jedním z cílů práce je implementace zachycení objektů. Po prostudování způsobů uchopení předmětu lidskou rukou můžeme realizovat podobný pohyb v modelu. Na základě těchto informací a charakteristik modelu můžeme zjistit, jaké formy pohybu lze v práci použít.

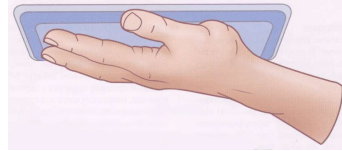
Metody zachycení a držení objektu jsou rozděleny do tří hlavních skupin: zachycení(2.5a), zachycení pomocí gravitace(2.5b), zachycení spolu s akcí(2.5c). V této části již můžeme říci, že konstrukce robotické ruky použité v práci není vhodná pro třetí skupinu zachycení. Protože zachycení s akcí vyžaduje použití svalů prstů a puky, které nejsou v modelu implementovány. Například otáčení prstu a pohyb pouze části prstu. Druhá skupina zachycení nevyžaduje složité řešení pro ovládání modelu, v podstatě jde o podporu objektu. Proto v této práci budeme věnovat pozornost první skupině. Skupina zachycení je rozdělena do tří podskupin[7]:

- prstové - zachycení je provedeno pouze pomocí prstů.
- dlaňové - při uchopení se používají dlaň a prsty.
- středové - úchyty vytvářejí symetrii kolem podélné osy, která se obvykle shoduje s osou předloktí.

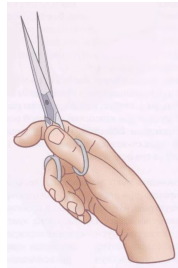
Každá z těchto podskupin obsahuje úchop, který lze realizovat v použité robotické ruce. Návrh a implementace těchto technik zachycení je cílem praktické části práce.



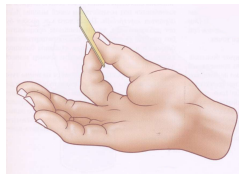
(a) Zachycení. Převzato z [7].



(b) Zachycení pomocí gravitace. Převzato z [7].



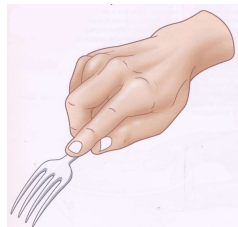
(c) Zachycení spolu s akcí. Převzato z [7].



(a) Prstové zachycení. Převzato z [7].



(b) Dlaňové zachycení. Převzato z [7].



(c) Středové zachycení. Převzato z [7].

Testování možností modelu

V této kapitole otestujeme model robotické ruky Youbionic. Musíme identifikovat objekty, které model dokáže zachytit a držet. Pro testování provedeme změny v původním řešení řízení modelu od výrobce. Změníme princip ovládání otevírání a zavírání ruky, pomocí svalového senzoru namísto svalového senzoru bude model cyklicky otevírat a zavírat ruku. Pro různé typy objektů jsou určeny různé počáteční stavy otevřené ruky pro implementaci různých typů uchopení. Tyto typy úchopů jsou popsány v předchozí kapitole. Program pro testování je v elektronické příloze k práci. Vybereme osm předmětů různých tvarů, materiálů a hmotností. Vybrané objekty umožňují testovat různé typy zachycení a lidé je také denně používají. Všechny položky váží méně než 0,6 kg pro testování, aniž by došlo k poškození modelu. Výsledky testů jsou podrobně popsány v Tabulce 3.1.

Výsledky zkoušek ukazují, že model je vhodný pro zachycení válcových předmětů a není vhodný pro práci s malými předměty. Ve robotické ruce vybrané pro tuto práci není pohyb určité části prstu a rotace zamýšlen. Model umožňuje pohyb pouze celého prstu. Tato omezení znemožňují přesnou práci (např. psaní), ale model nedokáže s přesností uchopit malé objekty.

Tabulka 3.1: Výsledky testů

Předmět	Popis	Velikosti , cm	Hmotnost , g	Způsob zachycení	Výsledek
Tužka	-	16 x 0,7	8	prstové/středové	Ruka může držet tužku mezi palcem a ukazováčkem. A nebo uchopit tužku, aby byla vhodná pro psaní. Nicméně, síla stisku není vhodná pro psaní.
Sklenici	válcové prázdné sklo	14 x 9	240	dlaňové	Ruka je schopna sklenici zachytit a držet.
Hrnek	prázdný hrnek	10 x 8	160	prstové	Ruka může hrnek držet za rukojeť.
Knihy	vázaná knihy	19 x 13 x 2	350	prstové	Ruka nemůže manipulovat se stránkami knihy. Ale může chytit a držet celou knihu.
Kladivo	-	25 x 3	470	dlaňové	Ruka může uchopit kladivo za rukojeť, ale síla uchopení je pro pevné uchopení nedostatečná. Rukojeť kladiva vyklouzla z ruky.
Plastová láhev	plná plastová láhev 0,5l	23 x 8	522	dlaňové	Ruka může chytit láhev a držet ji
Míč	gumový míč	9 x 9 x 9	41	dlaňové	Ruka dokáže uchopit míč nebo jiný kulový předmět.
Papír	několikrát složený list papíru	10 x 5	5	prstové	Ruka může držet papír mezi palcem a ukazováčkem.

3. TESTOVÁNÍ MOŽNOSTÍ MODELU

Analýza a návrh

Tato kapitola je věnována analýze a vývoji řešení. V kapitola jsou popsány technologie zvolené pro dosažení cílů této práce a jsou navrženy změny v modelu a řešení pro ovládání robotické ruky. Jsou navržena funkčnost aplikace pro demonstraci možnosti robotické ruky a uživatelské rozhraní této aplikace.

4.1 Polohy prstů robotické ruky

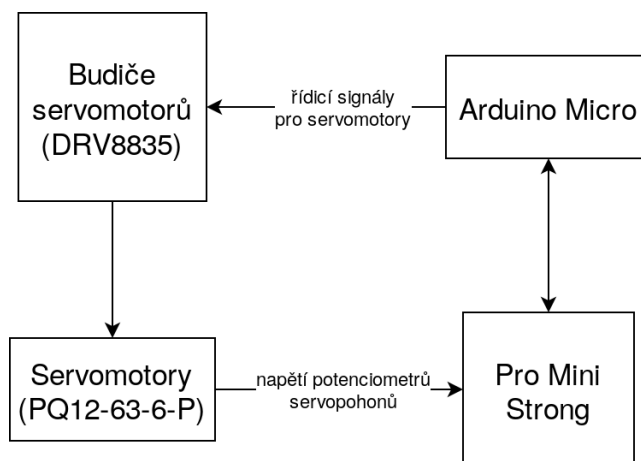
Hlavním cílem práce je ovládání robotické ruky. Pro co nejpřesnější ovládání modelu potřebujeme vědět aktuální polohu všech prstů. Pro tyto účely, PQ12 pohony použité v modelu mají potenciometr zpětné vazby. Servopohony nemají vestavěné budiče a potenciometr zpětné vazby nebyl použit v původním modelu. Ze studie v teoretické části práce, dokumentaci pohonu, víme, že výstup potenciometru je umístěn na PIN1 servopohonu a je analogový signál. Můžeme tento signál připojit k Arduino a sledovat výstupní napětí potenciometrů. Výstupní napětí potenciometrů je úměrné poloze pohonu, ke kterému je připojeno. Poloha pohonu v modelu manipulátoru odpovídá poloze prstu, který řídí servomotor[8].

Prvním praktickým úkolem práce je připojit zpětnou vazbu servopohonů k modelu, aby bylo možné řídit pohony v závislosti na jejich poloze. Robotická ruka má šest servopohonů, takže potřebujeme šest volných kontaktů na ovládací desce. Arduino Micro, který řídí model, neobsahuje dostatek volných pinů, které lze použít k připojení zpětné vazby z servopohonu. Zvolené řešení je přidat další ovládací prvek a nastavit spojení mezi původní deskou a deskou periferní. Toto řešení bylo zvoleno z několika důvodů:

- toto řešení vyžaduje minimální změny v konstrukce původního modelu
- nedostatek volných pinů na původní desce
- možnost přidání dalších senzorů a jiných vstupů a výstupů do celého projektu

Pro tyto účely byla zvolena deska „DM Pro Mini Strong“, která je plně kompatibilní s Arduino. Díky kompatibilitě a malé konstrukce tato deska je optimálním řešením pro integrace do modelu a splnění cílů práce.

Na Obrázku 4.1 je navržen nový diagram modelu. Výstupy potenciometru servomotoru jsou připojeny k desce Pro Mini Strong, která komunikuje s Arduino Micro. Arduino Micro na základě zpětné vazby potenciometru nastavuje polohu servopohonů.



Obrázek 4.1: Komunikace komponent robotické ruky

4.2 Komunikace mezi deskami Arduino

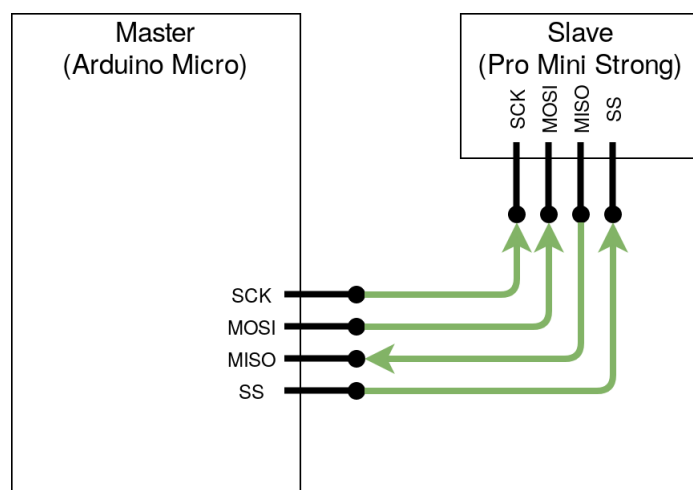
Externí deska Pro Mini Strong, ke které připojíme výstupy servopohonů, je použita v navrhovaném řešení ovládní robotické ruky. Pro přístup k těmto výstupům musíme připojit vybranou desku k hlavní řídicí desce modelu (Arduino micro).

Existuje několik různých řešení pro komunikaci mezi deskami Arduino. Většina komunikačních řešení (např. Standard Serial communication, I2C serial communication) používá výstupní piny Arduino, které jsou již v původním modelu použity nebo jsou nepřístupné. Zvoleným řešením je komunikace prostřednictvím SPI (Serial Peripheral Interface). SPI je synchronní sériový datový protokol používaný mikrokontroléry pro komunikaci s jedním nebo více periferními zařízeními rychle na krátké vzdálenosti. V původním návrhu modelu jsou výstupy pro komunikaci přes SPI k dispozici bez jakýchkoli změn v projektu. SPI má linky MISO, MOSI, SS a CLK[9].

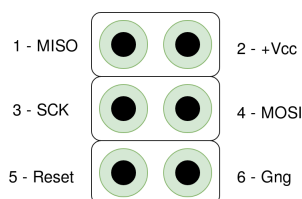
- MISO (Master in Slave Out) - Slave linka pro odesílání dat do Master.
- MOSI (Master Out Slave In) - Master linka pro odesílání dat do periferie.

- SCK (Serial Clock) - Hodinové impulsy synchronizující přenos dat generovaný Masterem.
- SS (Slave Select) - Master může pomocí této linky vybrat konkrétní zařízení.

Obrázek 4.2 ukazuje standardní schéma připojení přes SPI. Tabulka 4.1 ukazuje umístění komunikačních linek SPI na použitých deskách. V navrhovaném řešení jsou pouze dvě zařízení, takže není nutné používat linku SS. Zbývající linky potřebné pro komunikaci mezi deskami jsou umístěny na rozhraní ICSP (Obrázek 4.3), které se používá k programování desky prostřednictvím sériové linky. Toto rozhraní je snadno přístupné a přes toto rozhraní bude externí deska poháněná hlavní řídicí jednotkou. Z těchto důvodů byl zvolen navrhovaný způsob připojení.



Obrázek 4.2: SPI komunikace



Obrázek 4.3: ICSP

SPI Linka	Arduino Micro	DM Pro Mini Strong
MOSI	10 nebo ICSP-4	15 nebo ICSP-4
MISO	11 nebo ICSP-1	16 nebo ICSP-1
SCK	9 nebo ICSP-3	17 nebo ICSP-3
SS	8	14

Tabulka 4.1: Umístění linek SPI

Pro implementaci komunikace pomocí SPI použijeme knihovnu `SPI.H` pro Arduino. Arduino Micro bude nastaveno do režimu Master a DM Pro Mini Strong do režimu Slave. Na straně Master můžeme použít funkci `SPI.transfer` (`MasterSend`), jejíž prostřednictvím budeme odesílat data do zařízení Slave a současně vyvolávat přerušování na Slave zařízení. Implementujeme zpracování přerušování tak, aby Slave předal data Masteru na základě získaných dat.

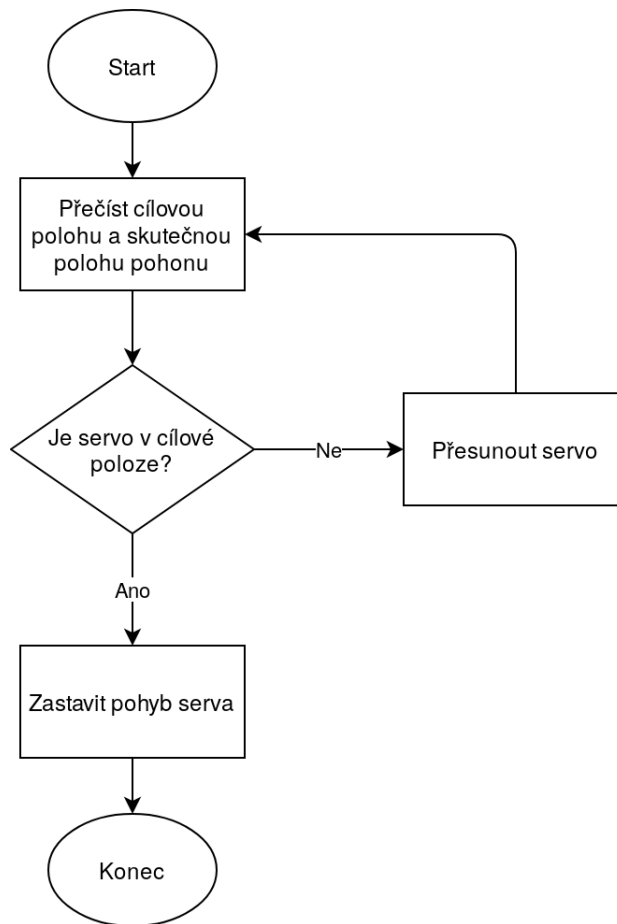
4.3 Řízení modelu

Správné řízení modelu robotické je jedním z cílů této práce. Proto je potřeba zvolit řídicí systém, který bude použit pro ovládání modelu. Řídicí systémy jsou rozděleny do dvou obecných kategorií: systémy s otevřenou smyčkou a uzavřenou smyčkou. Rozdíl je v regulační akci, která odpovídá za výstup systému. Principy těchto systémů jsou popsány v knize „Feedback and control systems“ [10].

V původním řešení je vztah mezi vstupem a výstupem přímý a není ovlivněn vnějšími poruchami. Vstupem do systému je signál ze svalového senzoru, na jehož základě se provádí otevírání a zavírání robotické ruky. Tento typ systému s otevřenou smyčkou je nedostatečný pro kompletní nezávislé ovládání modelu. Pro uchopení předmětu v robotické ruce, je třeba zastavit motory, aby nedošlo k poškození modelu nebo předmětu. Proto je nutné navrhnout systém, který bude reagovat na překážky v pohybu prstů modelu.

Pro ovládání servomotoru platforma Arduino obsahuje knihovnu `Servo.h`. Problém s vybraným modelem je v tom, že budič používaný pro servopohony nepodporuje instalaci pohonu do určité polohy a je nekompatibilní s knihovnou. Připojením zpětné vazby servopohonu k hlavní řídicí jednotce modelu můžeme sledovat změny polohy prstu robotické ruky, ale kvůli vnějšímu připojení potenciometru pohonu nemůžeme pouze nastavit požadovanou polohu servomotoru. Navrhované řešení je cyklus pro nastavení modelu do předem určené polohy na základě aktuální polohy servomotoru. Obrázek 4.4 schematicky znázorňuje navrhované řešení.

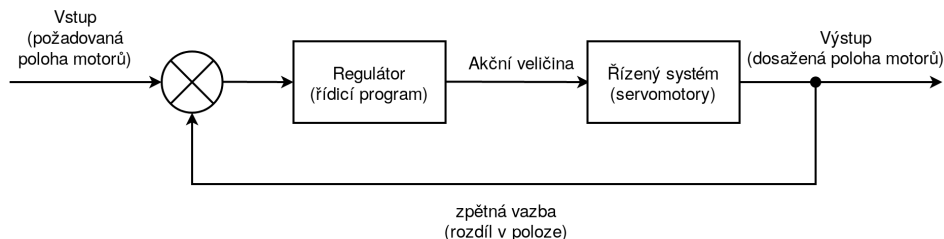
Toto řešení, na rozdíl od původního, nastavuje všechny prsty robotické ruky do požadované polohy. Což zvyšuje funkčnost mnohokrát. Dalším cílem je kontrolovat celý proces zavírání ruky a zastavit každý z prstů včas tak, aby objekt, který chceme uchopit nebo aby náš model nebyl v procesu zavírání poškozen. Zároveň musíme zastavit tuto ruku, aby mohla držet předmět.



Obrázek 4.4: Řízení modelu

Možným řešením je časové sledování stavu všech servomotorů v ruce. Myšlenka této metody je sledovat polohu každého servomotoru v procesu zavírání nebo otevírání ruky, poté aplikace na základě naměřených výsledků rozhodne, zda zastaví servomotory. Zpočátku víme, původní polohu všech prstů, když je ruka volně otevřena a zavřena (bez předmětu). Vložíme-li objekt do ruky a začneme uzavírat můžeme sledovat změnu v poloze motoru, pak ve chvíli, kdy ruka zasáhne objekt poloha se přestane měnit. Program bude zpracovávat změny a zastaví proces uzavírání ruky. Výhodou tohoto řešení je jednoduchá hardwarová realizace v rámci tohoto projektu, absence potřeby přidávání dalších čidel do projektu a minimální změny v původním modelu. Nevýhody jsou nízká přesnost ovládání, nemůžeme udržet moc jemný předmět.

Základem navrhovaného algoritmu je metoda zpětné vazebné smyčky. Navržené řešení je znázorněno na Obrázku 4.5. Kombinace těchto řešení umožňuje upravit polohu všech prstů pro různé typy úchopů a regulovat zavírání ruky, aby se objekt zachytil.



Obrázek 4.5: Regulace modelu

4.4 Připojení modelu a PC

Dalším problémem je spojení mezi PC a robotickou rukou. Pro vytvoření aplikace na PC je nutné v modelu implementovat přenos dat do Arduino. Zvoleným řešením je sériová komunikace pomocí kabelu USB.

Všechny desky Arduino obsahují alespoň jeden sériový port (UART nebo USART) určený pro komunikaci s PC nebo jiným zařízením. Arduino navíc obsahuje řadu předem implementovaných funkcí pro čtení (`Serial.print()`) a zápis (`Serial.read()`) přes sériový port. Tyto funkce budou použity v programu Arduino Micro, který získá polohu robotické ruky z aplikace na PC a provede model o požadovaného stavu. V PC aplikaci pro ovládání modelu použijeme knihovnu `termios.h`[11], pomocí které určíme vstupní/výstupní rozhraní pro komunikaci s Arduino Micro.

4.5 Ovládání modelu pomocí senzoru

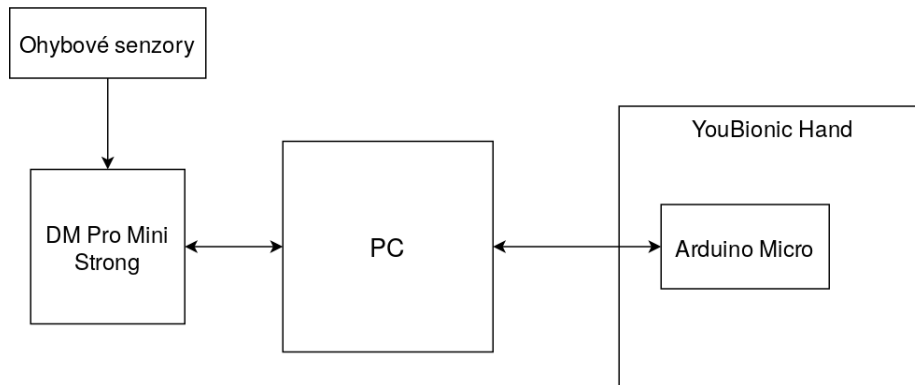
V původním projektu byl použit svalový senzor ke snímání pohybu paže uživatele. Nevýhodou tohoto řešení je, že pomocí svalových senzorů není možné zjistit přesnou polohu paže. Proto k řešení problému „*ovládání robotické ruky pohybem lidské paže*“ je vybrán senzor ohybu.

Senzor ohybu nebo ohybový senzor je senzor, který měří velikost průhybu nebo ohybu. Princip činnosti těchto senzorů je polymerní inkoust, který se nanáší na jednu stranu senzoru. Tento inkoust obsahuje vodivé částice. Když se snímač ohýbá vodivé částice se pohybují dále od sebe a zvyšují odpor[12].

Obvykle je senzor přilepen k povrchu a odpor snímacího prvku se mění ohýbáním povrchu. Jako povrch pro tento snímač bude použit prst uživatele, tím pádem výstupní hodnota senzorů bude reprezentovat polohu prstů uživatele. Na základě této hodnoty aplikace bude řídit servomotory, aby robotická ruka dosáhla stejné pozice jako ruka uživatele.

Navrhovaným řešením pro ovládání modelu pomocí senzoru je ovladač ve tvaru rukavice. Senzory ohybu budou nalepeny na každý prst rukavice. Řídící jednotkou ovladače bude deska DM Pro Mini Strong. K této desce bude připojeno pět snímačů ohybu. Řídící jednotka předá hodnoty senzoru do

robotické ruky přes PC. Obrázek 4.6 ukazuje blokové schéma celého návrhu projektu.

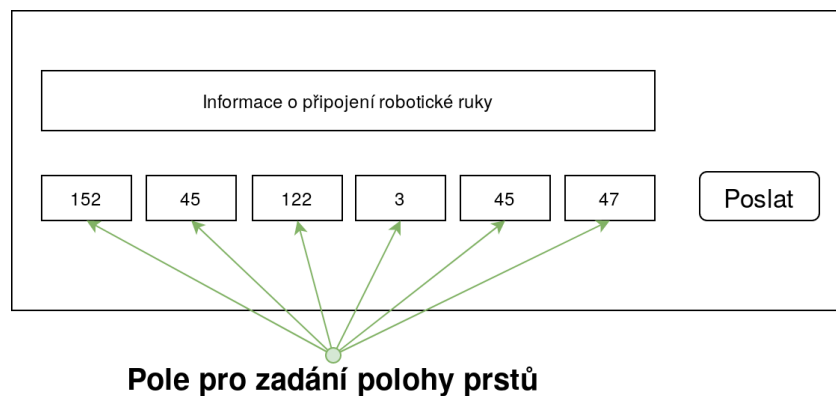


Obrázek 4.6: Navrhované schéma modelu

4.6 Aplikace

Demo aplikace by měla ukázat možnosti celého modelu. Kromě toho aplikace spojí řídicí rukavici s robotickou rukou. Pomocí aplikace bude uživatel moci připojit komponenty v rozhraní aplikace, ovládat modelem robotické ruky a sledovat aktuální polohu všech prstů.

Aplikace by měla obsahovat dvě základní obrazovky. První pro ruční ovládní robotické ruky. Uživatel zadá polohu všech prstů. Poté aplikace odešle tyto polohy do ovladače robotické ruky, který převede model do požadované polohy. Obrázek 4.7 ukazuje wireframe první obrazovky.



Obrázek 4.7: První obrazovka aplikace

Druhá obrazovka je určena k ovládní modelu pomocí rukavice. Uživatel připojí řídicí rukavice k počítači a stiskne tlačítko pro přenos dat z rukavice do robotické ruky.

Pro snadnější implementaci aplikace a vývoj projektu v budoucnu je nutné napsat knihovnu pro ovládání modelu. Tato knihovna umožní práci s modelem v dalších projektech a umožní rozšíření existujících řešení.

K implementaci aplikace byla vybrán framework **GTK**. Je to otevřený nástroj pro vytváření grafických uživatelských rozhraní. Zvolený programovací jazyk pro vytvoření aplikace je **C**.

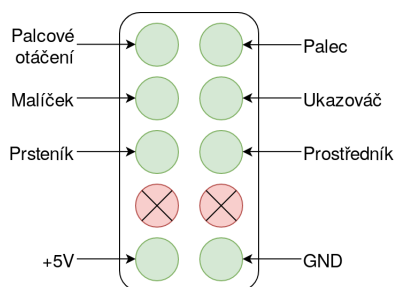
K implementaci grafického rozhraní bude použit program **Glade**[13]. To je aplikace pro vytváření grafického rozhraní na základě knihovny **GTK**.

Realizace

Tato kapitola popisuje implementaci řešení navržených v předchozí kapitole. Podrobně vysvětleny hardwarové změny v modelu pro jednodušší rozšíření projektu do budoucnosti. Představena aplikace pro demonstraci možnosti modelu. Jsou popsány implementované programy pro komponenty projektu.

5.1 Připojení a komunikace mezi komponenty

Pro přístup k potenciometrům v servomotorech byla původní deska přepájena a zpětná vazba servomotoru byla vyvedena na externí port (Obrázek 5.1). K tomuto portu je připojena deska DM Pro Mini Strong pomocí drátů. Tabulka 5.1 ukazuje implementované připojení potenciometrů servomotorů k řídicí desce.



Obrázek 5.1: Port potenciometrů

Potenciometr	Pin
Ukazováč	A0
Prostředník	A1
Prsteník	A2
Malíček	A3
Palec	A4
Palcové otáčení	A5

Tabulka 5.1: Připojení potenciometrů

Komunikace mezi ovladačem robotické ruky a externí deskou realizovaná prostřednictvím knihovny `SPI.h`. Funkce `SPIgetServoPositions()` byla implementována na straně Master, která používá funkci `transferAndWait()` k dodání hodnoty potenciometru zpětné vazby servomotoru z externí desky. Pomocí funkce `SPI.transfer()` Master vysílá číslo požadovaného potenciometru a vyvolá přerušení na straně slave. Implementována obsluha přerušení, ve které

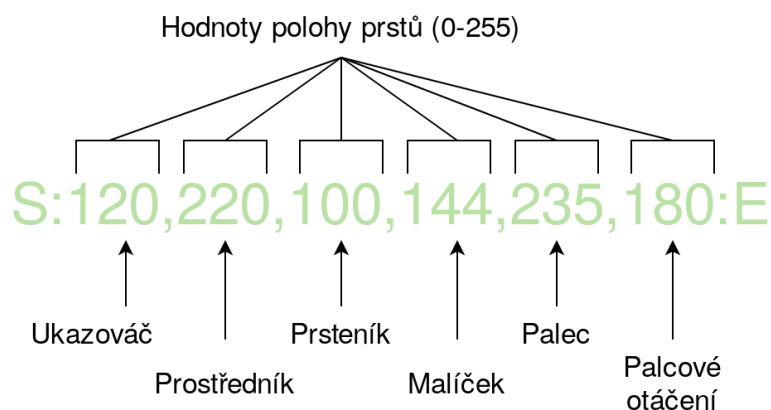
5. REALIZACE

Slave na základě přijatých dat předává hodnoty vstupních signálů z potenciometrů servomotorů. Tabulka 5.2 ukazuje vztah mezi přijatou a odeslanou hodnotou na zařízení Slave.

Přijatá hodnota	Odeslaná hodnota
1	hodnota na A0
2	hodnota na A1
3	hodnota na A2
4	hodnota na A3
5	hodnota na A4
6	hodnota na A5

Tabulka 5.2: Slave odesílání

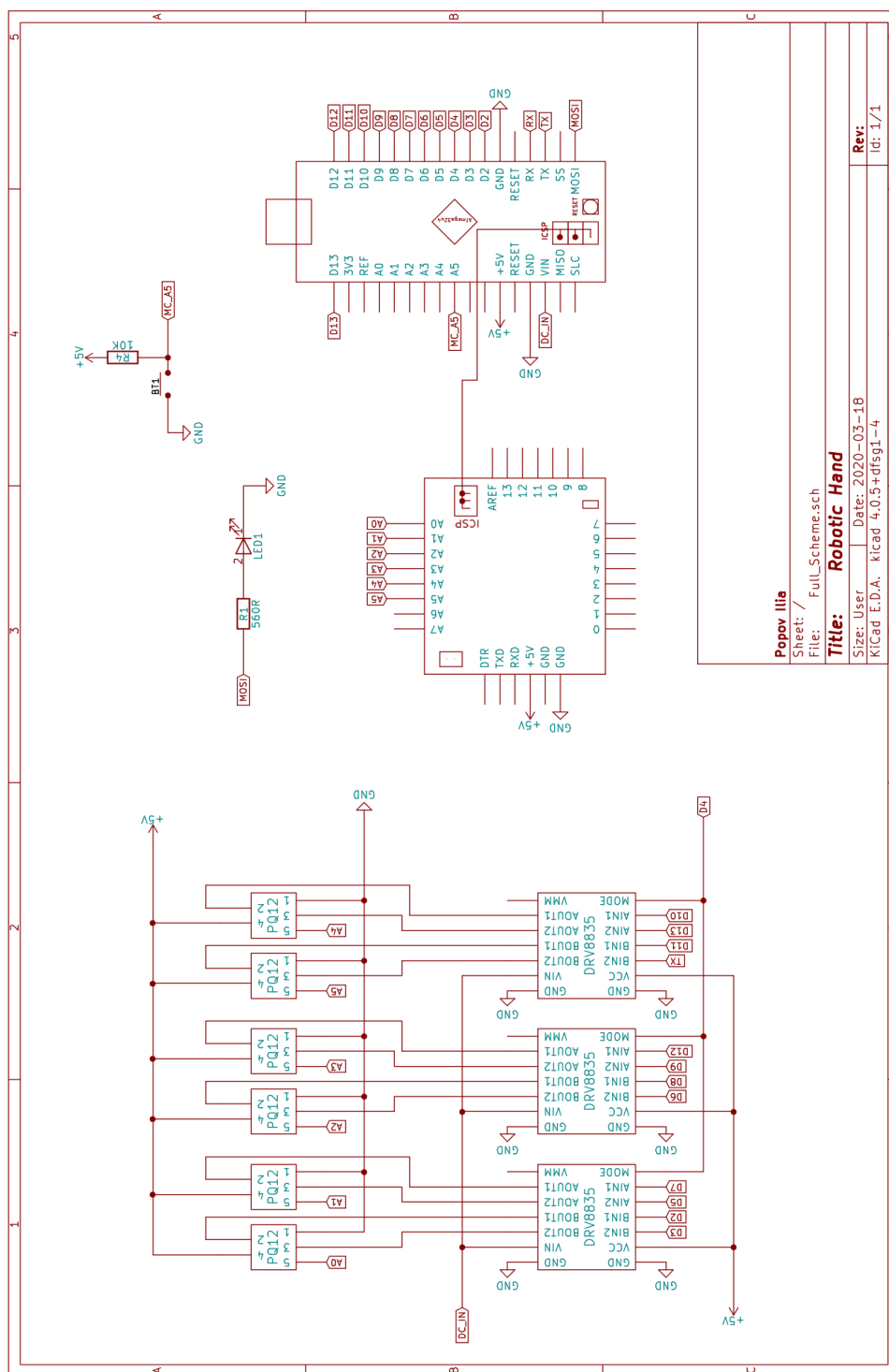
Řídicí jednotka robotické ruky čte požadovanou polohu ze sériového portu. Pak přivádí model do požadovaného stavu. Obrázek 5.2 ukazuje implementovaný formát, ve kterém ovladač robotické ruky přijímá informace o požadované poloze. Po zastavení všech servomotorů pošle řídicí jednotka aktuální polohy pohonů. Stejný formát byl použit pro příjem a odeslání pozice motoru.



Obrázek 5.2: Formát příkazu

Pro ilustraci změn provedených v modelu a pro demonstraci propojení všech komponent, bylo vytvořeno schéma připojení komponentů v projektu. Schéma znázorněné na Obrázku 5.3 a přidané k elektrickým přílohám díla.

5.1. Připojení a komunikace mezi komponenty



Obrázek 5.3: Schéma zapojení komponentů

5. REALIZACE

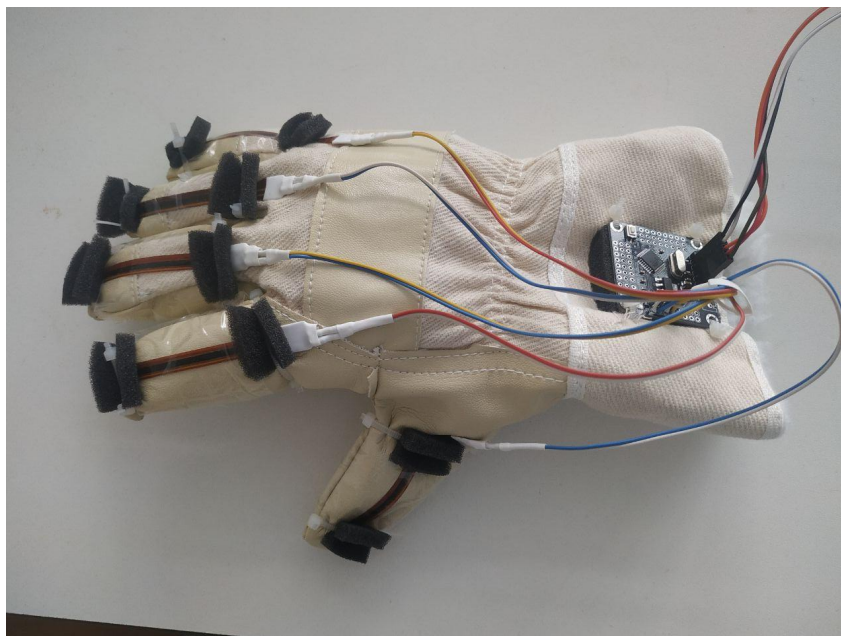
Po provedení všech změn v návrhu byl model znovu sestaven. Konečný výsledek je znázorněn na Obrázku 5.4.



Obrázek 5.4: Sestavený model ruky

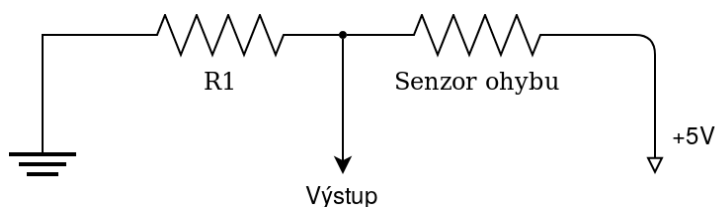
5.2 Rukavice

V rámci této práce byl navržen a implementován řadič, ve tvaru rukavice(5.5), pro ovládání robotické ruky ohýbáním paže uživatele. Senzory ohybu byly přilepeny na každý prst pracovní rukavice.



Obrázek 5.5: Rukavice

Senzory jsou propojeny do ovládací desky přes odpor způsobem znázorněným na Obrázku 5.6.



Obrázek 5.6: Připojení senzoru ohybu

Optimální hodnota odporu se rovná 47K Ω a byla spočítána z výrazu pro výstupní napětí této schématy: $V_{out} = \frac{V^+ R_1}{1 + \frac{R_1}{R_2}}$. Tato hodnota má největší rozsah výstupní hodnoty při ohýbání senzoru. Uvedený odpor byl použit při implementaci.

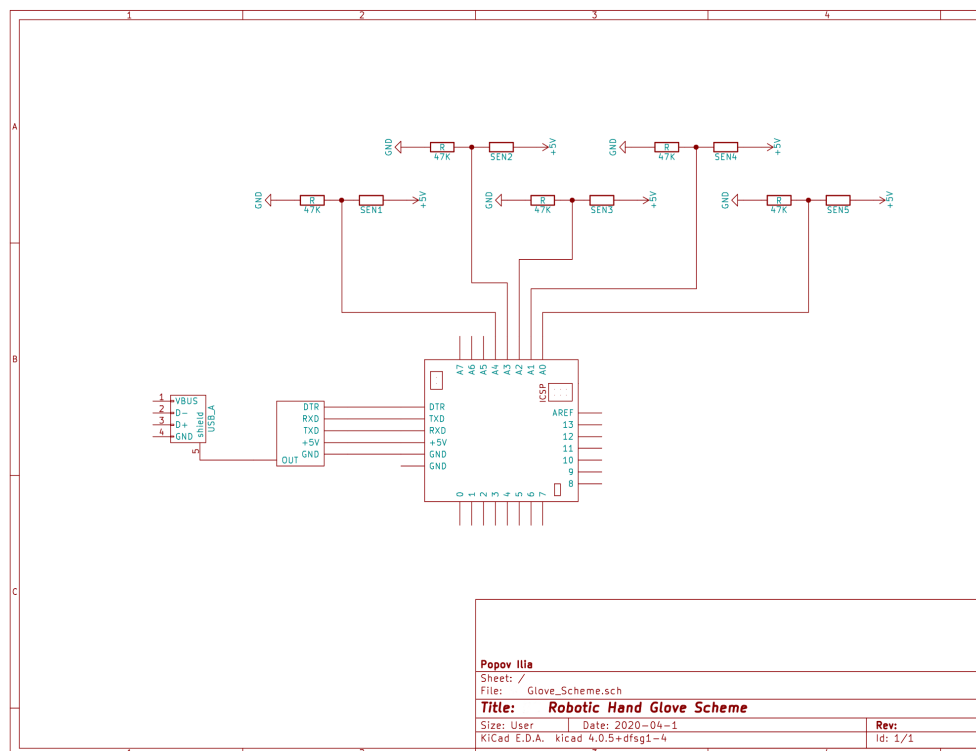
V této práci byly použity oboustranné ohybové senzory (senzor se ohýbal ve dvou směrech). Model ruky a lidská paže se ohýbají pouze v jednom směru, proto k ovladači byla připojena pouze jedna strana každého senzoru.

Realizované schéma zapojení je znázorněno na Obrázku 5.7. Řídicí jednotka zpracovává hodnoty senzorů a odesílá ve formátu používaném v ovladači robotické ruky k získání požadované polohy prstů modelu. Tabulka 5.3 ukazuje připojení prstů rukavice ke vstupním signálům řídicí jednotky.

Prst na rukavici	PIN
Ukazováč	A0
Prostředník	A1
Prsteník	A2
Malíček	A3
Palec	A4

Tabulka 5.3: Připojení prstů rukavice

5. REALIZACE



Obrázek 5.7: Schéma zapojení rukavice

5.3 Řídicí programy

Tato podkapitola popisuje implementované skripty pro všechny řídicí jednotky používané v práci. Všechny kódy popsané v této kapitole jsou součástí elektronické verze této práce a jsou popsány pomocí komentářů.

Všechny komponenty použité v projektu jsou desky Arduino nebo s nimi kompatibilní. Programy pro Arduino obsahují dvě základní funkce `setup()` a `loop()`. První se volá po zapnutí desky a používá se k inicializaci. Druhá funkce je určena pro hlavní část kódu a je volána cyklicky po celou dobu napájení desky.

5.3.1 Řídicí deska v robotické ruce

Program implementovaný pro hlavní řídicí jednotku robotické ruky, přijímá požadovanou polohu prstu robotické ruky přes sériový port ve formátu popsáném v předchozí podkapitole (Obrázek 5.2).

Program pak kontroluje správnost přijatých dat. Poloha každého prstu má rozsah pro nastavení 0-255, kde 0 - prst je zavřený a 255 - maximální otevřený. Při zadávání záporné hodnoty do polohy libovolného prstu program ponechá

vybraný prst v původní poloze. Model nebude reagovat na jiné změny formátu nebo nesprávný vstup.

Po přijetí platné zprávy pro nastavení modelu program sleduje aktuální polohu pohonu a porovná ji s požadovanou pozicí. Na základě tohoto srovnání ovladač pohybuje servopohonu a pak znovu porovnává polohy.

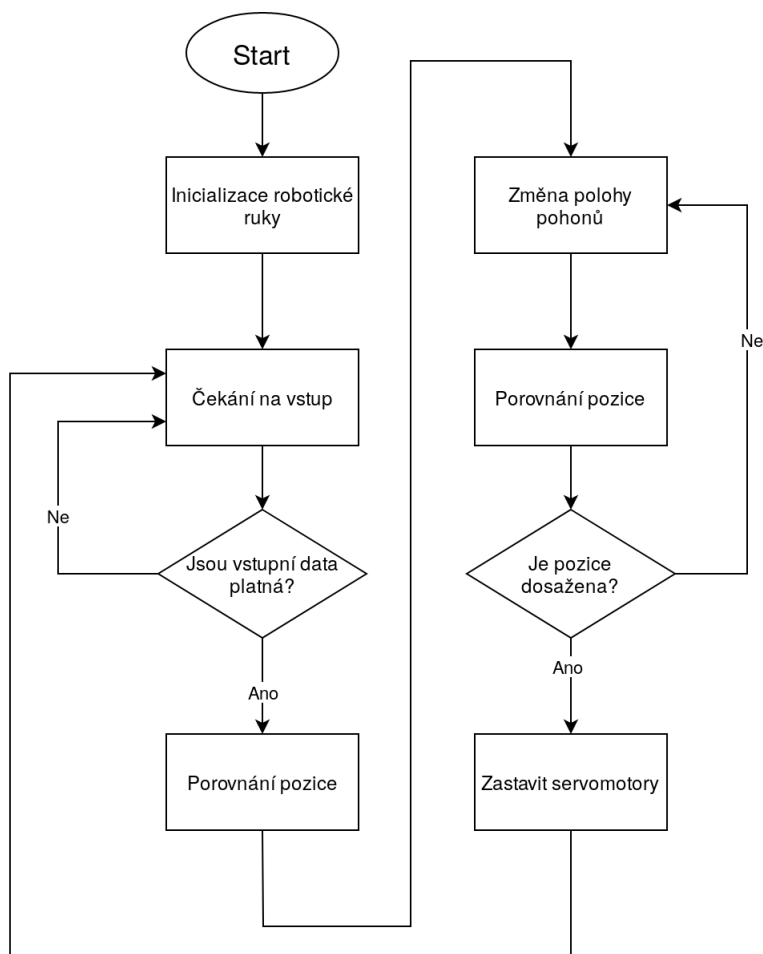
Proces porovnávání polohy a pohybu servopohonu se provádí, dokud pohon nedosáhne požadované polohy nebo nepřestane měnit svou polohu (v případě selhání pohonu, došlo k výpadku napájení pohonu, překážce v pohybu).

Po zastavení všech servomotorů se program vrátí do původního stavu, kde čeká na další vstup. Obrázek 5.8 odkaz ukazuje vývojový diagram implementovaného programu.

Pro jednoduchost použití a přehlednost v kódu aplikace byla realizována struktura `SFinger` 5.1, která reprezentuje prst modelu robotické ruky v kódu aplikace. V každé instanci této třídy jsou uloženy výstupní piny pro řízení servomotorů, číslo prstu, pozice, která má být dosažena a pro přesnost ovládání jsou uloženy dvě poslední změřené pozice servomotorů.

Ukázka kódu 5.1: Reprezentace motoru

```
struct SFinger {
    int pinDir;
    int pinPwm;
    int fingerPos;
    int fingerPrevPos;
    int fingerID;
    int targetPos;
};
```

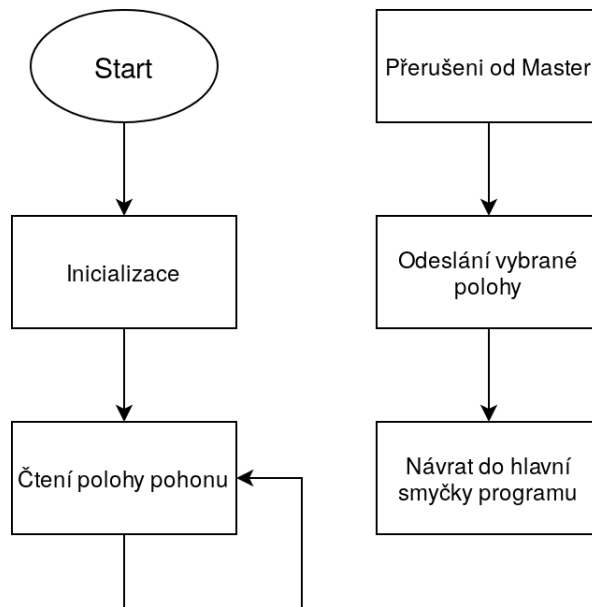


Obrázek 5.8: Program pro hlavní řídicí jednotku robotické ruky

5.3.2 Externí deska v robotické ruce

Program pro externí desku, ke které jsou připojeny zpětné vazby servopohonů, nepřetržitě čte a ukládá aktuální hodnoty potenciometrů.

Při volání přerušení ze zařízení Master, řídicí jednotka odešle jednu z uložených hodnot polohy pohonu. Funkce `SPI.transfer()` z knihovny `SPI.h`, použitá na straně Master pro předávání dat, podporuje velikost dat 8 bitů. Proto jsou naměřené a následně odeslané hodnoty přiřazeny k rozsahu 0-255. Obrázek 5.9 ukazuje vývojový diagram implementovaného programu.



Obrázek 5.9: Program pro externí desku

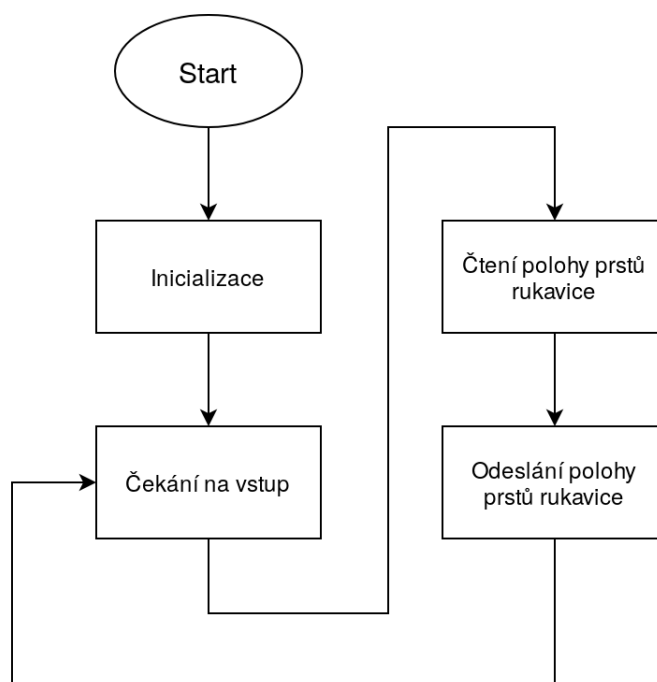
5.3.3 Řídicí jednotka rukavice

Implementace řídicí jednotky ovladače je podobná programu pro externí desku v robotické ruce. Hlavním rozdílem je způsob komunikace. Ovladač komunikuje přes sériový port.

Program čeká na požadavek na sériovém portu. Po přijetí zprávy ovladač pošle polohu všech prstů do rukavice ve formátu pro řídicí jednotku robotické ruky (Obrázek 5.2).

Protože robotické rameno nastavuje šest motorů, má-li rukavice pouze pět senzorů pro sledování polohy prstu uživatele, byla k odeslané hodnotě přidána hodnota -1 pro polohu otáčení palce. To umožňuje přenos dat přímo z rukavice do řídicí jednotky bez jakýchkoli změn za předpokladu, že poloha otáčení palce zůstává nezměněna.

Obrázek 5.10 ukazuje vývojový diagram implementovaného programu.



Obrázek 5.10: Program pro ovládací rukavice

5.4 Knihovna

Pro ovládání modelu v navržené demo aplikaci nebo v jiném PC programu, byla vytvořena knihovna `Hand.h` v jazyce C.

Knihovna obsahuje funkci pro připojení k modelu robotické ruky a dalším zařízením použitým v této práci přes sériový port. Takže knihovna obsahuje funkci ovládání modelu a přeposílání dat mezi zařízeními. Pro reprezentaci zařízení byla vytvořena struktura `SArdDev`. Ve struktuře jsou uloženy adresa zařízení a stav připojení. Knihovna také obsahuje strukturu `SSerial`, ve které jsou uložena nastavení pro komunikaci se zařízením a původní nastavení.

Popis základních funkcí knihovny uveden v Tabulce 5.4. Podrobný popis struktur a funkcí knihovny jsou uvedeny v přílohách.

Funkce	Popis
openDevice (SArdDev* device, const char* addr)	Funkce inicializuje zařízení na adrese addr. Struktura SArdDev představuje zařízení v knihovně. Vrací true při úspěšné inicializaci a false při selhání.
readFromArd (SArdDev device, char* buff, int size)	Funkce čte data ze zařízení device do pole buff.
openHand (SArdDev* device)	Funkce odešle příkaz otevření robotické ruky do zařízení device.
closeHand (SArdDev* device)	Funkce odešle příkaz uzavření robotické ruky do zařízení device.
sendStatesToHand (SArdDev* device, int states[6])	Funkce odešle příkaz k nastavení všech prstů robotické ruky na pozice uložené v poli states

Tabulka 5.4: Základní funkce knihovny

5.5 Aplikace

Na základě implementované knihovny pro řízení modelu a knihovny GTK pro implementace uživatelského rozhraní byla vytvořena aplikace pro demonstraci funkcí modelu robotické ruky. GUI aplikace vytvořené v aplikaci Glade. Popis vytvořeného rozhraní uložen v XML souborech. Vytvořena aplikace připojuje rozhraní pomocí objektu `GtkBuilder`.

Aplikace má tři základní obrazovky pro různé způsoby ovládání modelu a obrazovku Menu pro přepínání mezi obrazovkami. Pro komunikaci mezi obrazovkami, objekty aplikace a sledování stavu aplikace byla implementována struktura `app_widgets`, která je přístupná z funkcí všech objektů aplikace. Tato struktura obsahuje odkazy na všechny objekty používané v aplikaci, uložené struktury pro připojení komponent modelu a stav aplikace a připojených komponentů. Po spuštění aplikace inicializuje všechny obrazovky, objekty GUI a konfiguruje komunikační parametry.

Všechny obrazovky obsahují pole pro zadání adresy zařízení a tlačítko připojení. Jakmile je model robotické ruky připojen, informace o stavu připojení a poloha prstů modelu budou viditelné na všech obrazovkách aplikace. Při zavření jakékoli obrazovky, program zničí všechny objekty, vrátí parametry změněné v aplikaci na výchozí nastavení a ukončí aplikaci.

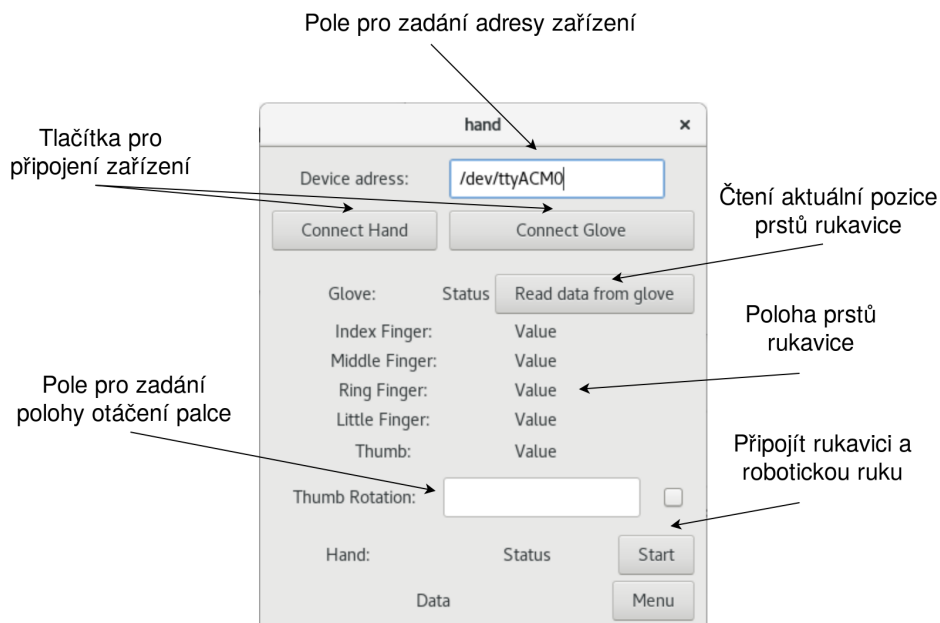
5.5.1 Obrazovka ovládání přes rukavice

První obrazovka je určena k ovládání modelu pomocí rukavice. Uživatel zde může připojit model robotické ruky a ovladač rukavice. Po připojení robotická ruka bude kopírovat aktuální polohu rukavice.

Implementovaný model ovladače ve tvaru rukavice zachycuje polohu všech prstů. Robotický model kromě nastavení polohy všech prstů má možnost nastavit polohu otáčení palce. Proto byla v aplikaci implementována funkce ručního zadávání polohy otáčení palce. Poloha otáčení palce se zadává do konkrétního pole obrazovky. Tuto funkci lze vypnout zaškrtnutím políčka vedle pole. Když je tato funkce vypnutá, poloha otáčení palce se nezmění.

V aplikaci implementována funkce čtení a zobrazení polohy prstu ovladači rukavice. Po stisknutí tlačítka `Read data from glove` aplikace zkontroluje připojení rukavice. A pak pomocí funkce `g_timeout_add_full` z knihovny GTK periodicky vyvolává implementovanou funkci čtení dat ze sériového portu, ke kterému je rukavice připojena. Přečtená poloha prstů rukavice se zobrazí na obrazovce.

Po použití tlačítka `Start`, aplikace zkontroluje připojení komponentů. Poté bude pravidelně číst polohy prstů rukavice a předávat je modelu robotické ruky.

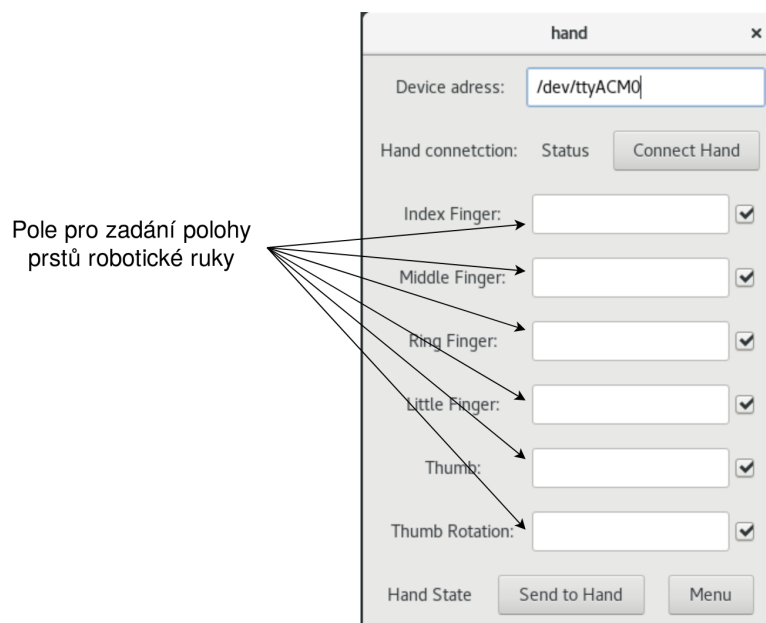


Obrázek 5.11: Obrazovka ovládání přes rukavice

5.5.2 Obrazovka manuálního ovládání

Druhá obrazovka je určena k ručnímu nastavení robotické ruky do polohy určené uživatelem. Zde může uživatel nastavit polohu všech prstů robotické ruky zvlášť. Pozice se zadávají do určitých polí na obrazovce. Je implementována funkce změny polohy pouze vybraných prstů. Prsty lze vybrat pomocí zaškrťovacích políček.

Stisknutím tlačítka **Send to Hand** aplikace kontroluje zadané pozice a přeposílá je do modelu. Pro odeslání byla použita funkce `sendStatesToHand` z implementované knihovny.

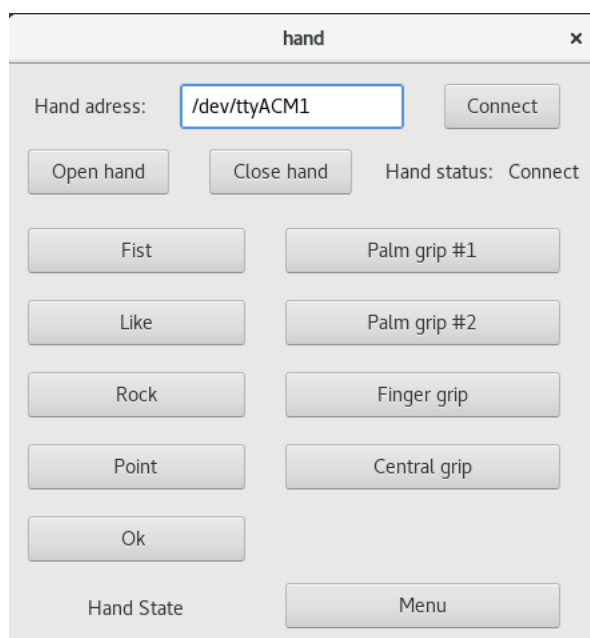


Obrázek 5.12: Obrazovka manuálního ovládání

5.5.3 Obrazovka přednastavených stavu

Poslední obrazovka obsahuje několik přednastavených pozic robotické ruky, které demonstrují možnosti modelu. Na této obrazovce jsou implementovány pohyby otevírání a zavírání ruky. Takže jsou implementovány přednastavené polohy pro různé způsoby uchopení předmětů, které byly popsány v kapitole 2.2. S těmito pozicemi se uživatel může pokusit zachytit různé objekty pomocí robotické ruky.

Pro otestování uchopení, je potřeba model nastavit do jedné z pozic a poté pomocí tlačítka **Close hand** zavřít robotickou ruku. V aplikaci je implementováno několik gestů, které jsou vhodné pro prezentaci modelu a jeho možnosti.



Obrázek 5.13: Obrazovka přednastavených stavu

Testování řešení

Na modelu YouBionic Hand byla otestována knihovna pro ovládání robotické ruky a ukázková aplikace pro demonstraci možnosti modelu. Videá ovládání modelem pomocí rukavice a uchopení předmětu robotickou rukou jsou na příloženém CD. Fotografie z testování modelu lze nalézt v příloze.

6.1 Testování robotické ruky

Pro testování realizovaného řešení pro robotickou ruku byl použit Arduino IDE a implementovaný do IDE Sériový monitor. Sériový monitor je nástroj pro komunikaci s připojením Arduino přes sériovou linku.

Testování probíhalo na základě připojení Arduino v robotické ruce k PC a následním odesláním polohy servomotoru přes Sériový monitor. Pro testování byla vytvořena série testů, ve kterých byla pozorována poloha servomotorů robotické ruky.

Testování bylo provedeno na dvou úrovních. V prvním bylo testováno dosažení modelu požadovaných poloh motoru bez překážek v pohybu. Byla testována změna pozic všech prstů samostatně a současné nastavení pozic všech prstů. Ve všech testech robotická ruka dosáhla požadovaných poloh prstů s odchylkou v poloze motorů 5-8%.

Druhý test ověřoval schopnost modelu zastavit prsty robotické ruky a chytit předmět. Pro testování byly použity předměty z kapitoly 3. Ve všech pokusech byl model schopen zastavit servomotory při překážkách v pohybu. Po zastavení motoru je v robotické ruce pevně držen předmět, který je překážkou v pohybu.

Díky implementovanému řešení byla robotická ruka při testování schopna držet tužku způsobem vhodným pro psaní. Video testu lze nalézt v příloženém CD. Fotografie z testů jsou uvedeny v příloze práce.

6.2 Testování aplikací a rukavice

Před testováním aplikace byla knihovna, implementovaná pro řízení modelu, testována samostatně. Všechny funkce knihovny fungují podle definovaných požadavků. Při komunikaci s modelem prostřednictvím knihovny model reaguje stejně jako při ovládní pomocí Arduino IDE.

Všechny aplikační funkce byly testovány a fungují podle definice. Při propojení ovladače rukavice a modelu robotického ramene aplikace realizuje řízení modelu pohybem uživatele. Zjištěným nedostatkem v řešení je rychlost reakce robotické ruky na změny polohy rukavice. Ruka kopíruje polohu se zpožděním cca 4-5 sekund. Směrem ke zlepšení je také přesnost ovladače rukavice. Implementovaný ovladač splňuje cíl ovládní modelu pomocí pohybu paže uživatele. Nicméně realizovaný model rukavici omezuje pohyby uživatele. Volné pohyby ruky mají větší flexibilitu než pohyby v rukavici. Pohodlnější a přesnější verze rukavice umožní širší rozsah pohybu a využije více možností robotické ruky.

Závěr

Cílem bakalářské práce bylo prozkoumat možnosti robotické ruky, a následně navrhnout a implementovat programové vybavení pro ovládání. Implementace byla provedena na modelu YouBionic Hand. Dalším cílem bylo realizovat ovládání modelu pohybem lidské paže, vytvořit demo aplikaci a upřesnit rozsah předmětů, které této rukou lze uchopit.

Při analýze vybraného modelu byly zjištěné nedostatky v původním projektu. Nebylo nalezeno žádné již existující vhodné řešení pro ovládání této robotické ruky. Proto byly navrženy a implementovány změny v designu původního projektu YouBionic Hand. Do projektu byla přidána a naprogramována externí periferní řídicí deska. Pro ovládání modelu byla napsána knihovna v jazyce C.

Pomocí senzorů ohybu a desky Arduino byl vytvořen ovladač ve tvaru rukavice. Také pro ovládání robotické ruky přes uživatelské rozhraní a spojení ruky s rukavicí byla implementována aplikace. Součástí aplikace je několik základních způsobů uchopení objektů kopírujících chování lidské paže.

Pro overení funkčnosti modelu a implementovaného řešení bylo provedené testování. Během testování bylo prokázáno uchopení různých objektů robotickou rukou a ovládání modelu pomocí pohybu paže uživatele, použitím ovladačí rukavici.

Všechny cíle byly splněny. Výsledky této práce umožní dalším zájemcům vytvořit složitější aplikace pro projekt YouBionic Hand. Projekt lze v budoucnu rozšířit přidáním přesnější verze ovladače pro řízení modelu, rozšířením funkčnosti modelu a realizací dalších pohybů. Část práce může být také použita pro projekty s jinými verzemi robotických ruk od výrobce YouBionic.

Literatura

- [1] You Bionic: *Technical Drawing*. [cit. 2019-02-19]. Dostupné z: <https://github.com/Youbionic-com/Youbionic>
- [2] arduino.cc: *Arduino Micro [online]*. [cit. 2019-04-10]. Dostupné z: <https://store.arduino.cc/arduino-micro>
- [3] Martin, S.: *Servomotor*. [cit. 2019-02-17]. Dostupné z: <https://navody.arduino-shop.cz/arduino-projekty/servo-motor.html>
- [4] Actuonix Motion Devices Inc.: *Miniature Linear Motion Series PQ12*. [cit. 2019-04-10]. Dostupné z: <https://s3.amazonaws.com/actuonix/Actuonix+PQ12+Datasheet.pdf>
- [5] Texas Instruments: *RV8835 Dual Low-Voltage H-Bridge IC*. [cit. 2019-04-06]. Dostupné z: <https://www.pololu.com/file/0J570/drv8835.pdf>
- [6] Advancer Technologies: *Electromyography Sensor for Microcontroller Applications*. [cit. 2019-02-14]. Dostupné z: <https://cdn.sparkfun.com/assets/a/3/a/f/a/AT-04-001.pdf>
- [7] Kapandji, I. A.: *The Physiology of the Joints, Volume 1: Upper Limb*. Eksmo, 6 vydání, 2018, ISBN 978-5-699-33140-6.
- [8] Position Sensors. *Electronics Tutorials [online]*, [cit. 2020-03-01]. Dostupné z: https://www.electronics-tutorials.ws/io/io_2.html
- [9] Atmel Corporation: *AVR151: Setup and Use of the SPI*. [cit. 2020-02-07]. Dostupné z: https://ww1.microchip.com/downloads/en/AppNotes/Atmel-2585-Setup-and-Use-of-the-SPI_ApplicationNote_AVR151.pdf
- [10] Joseph J. Distefano, W. J. W., Allen R. Stubberud: *Feedback and control systems*. McGraw-Hill, druhé vydání, 1994, ISBN 978-0078427091.

LITERATURA

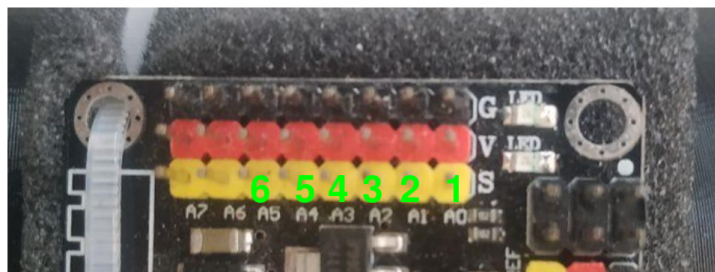
- [11] The Open Group: *termios.h*. [cit. 2019-11-20]. Dostupné z: <https://pubs.opengroup.org/onlinepubs/7908799/xsh/termios.h.html>
- [12] Flex Sensor introduction, pinouts and working. *Microcontrollerslab.com [online]*, [cit. 2019-09-07]. Dostupné z: <https://microcontrollerslab.com/flex-sensor-introduction-working/>
- [13] The Glade project: *Glade - A User Interface Designer*. [cit. 2020-04-14]. Dostupné z: <https://glade.gnome.org/>

Uživatelská příručka

Příručka obsahuje pokyny pro připojení součástí modelu robotické ruky a popis funkcí implementované knihovny pro ovládání.

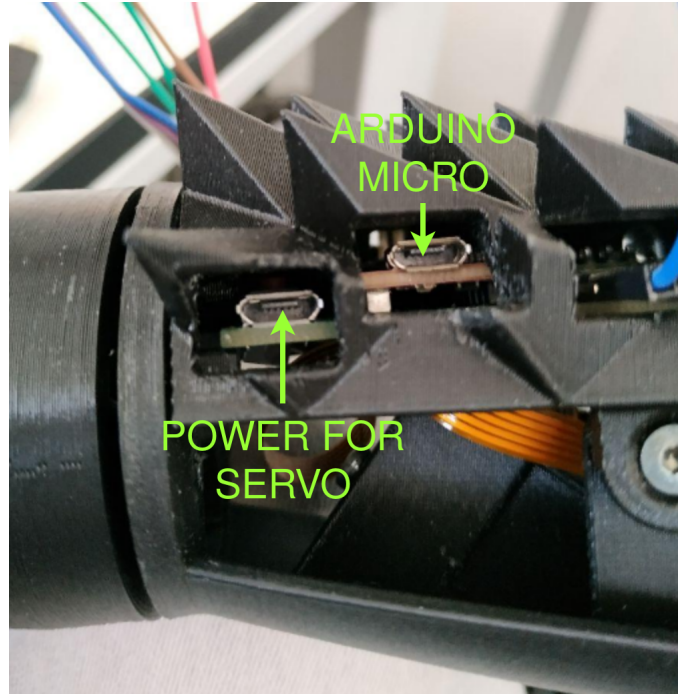
A.1 Připojení komponentů

Připojte výstupy servomotorů robotické ruky k externí desce, jak je znázorněno na Obrázku A.1.



Obrázek A.1: Výstupy servomotorů

Připojte robotickou ruku k počítači a ke zdroji energie (powerbanka)A.3.



Obrázek A.3: Připojení robotické ruky

A.2 Funkce knihovny

Knihovna je napsána v jazyce C a má následující funkce:

```
init_serial(SSerial *term)
```

Funkce provede nastavení pro komunikaci se zařízením a uloží původní nastavení. Parametr `term` je struktura pro uložení nastavení.

```
destr_serial (SSerial *term)
```

Funkce vrací původní nastavení.

```
set_io_speed (int tty_fd, SSerial *term)
```

Funkce nastavuje výstupní/vstupní přenosovou rychlost a piny RTS a DTR sériového portu. Parametr `tty_fd` určuje adresu zařízení. Parametr `term` určuje nastavení komunikace.

```
open_device (SardDev* dev, const char* addr)
```

Funkce inicializuje zařízení na adrese `addr` do struktury `dev`.

A. UŽIVATELSKÁ PŘÍRUČKA

`close_device (int tty_fd, SSerial *term)`

Funkce ukončí komunikaci se zařízením.

`read_from_device (SArdDev dev, char* buff, int size)`

Funkce přečte data odeslaná zařízením `dev` a uloží je do `buff`. Parametr `size` určuje velikost buffru pro čtení.

`read_glove_position (SArdDev dev, char* buff, int size, bool wait_glove)`

Funkce přečte aktuální polohu prstu připojené rukavice.

`write_to_device (SArdDev* dev, char* buff, int size)`

Funkce odesílá data z `buff` do zařízení `dev`.

`send_states_to_hand (SArdDev* dev, int states[6])`

Funkce posílá pozice servomotorů do robotické ruky z pole `states`.

`resend_data_between_device (SArdDev* from_device, SArdDev* to_device)`

Funkce přeposílá data ze zařízení `from_device` do zařízení `to_device`.

`open_hand (SArdDev* dev)`

Funkce otevře robotickou ruku.

`close_hand (SArdDev* dev)`

Funkce zavře robotickou ruku.

Fotografie z testování

Následující podkapitoly obsahují fotografie z testování modelu.

B.1 Testování možností modelu

Fotografie ke Kapitole 3.



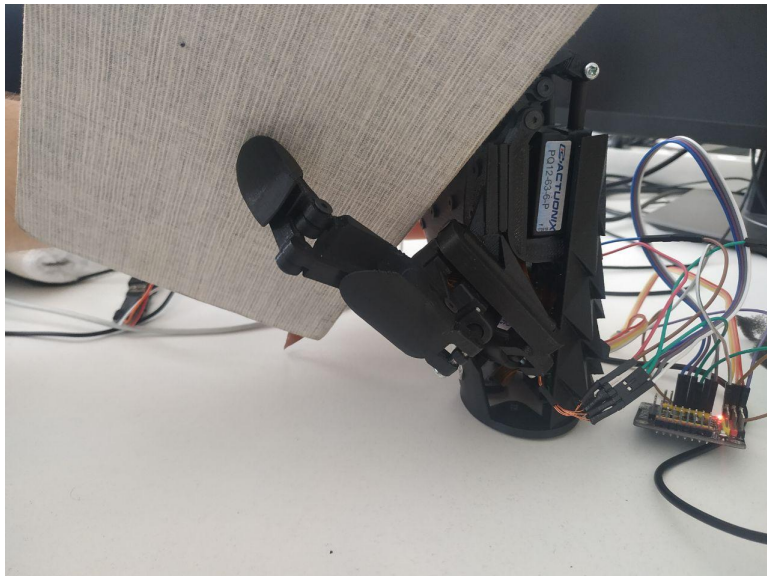
B. FOTOGRAFIE Z TESTOVÁNÍ





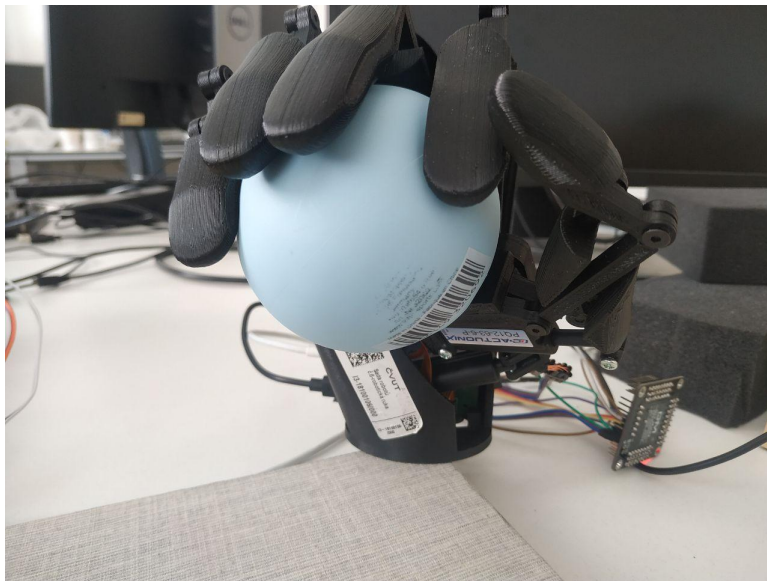
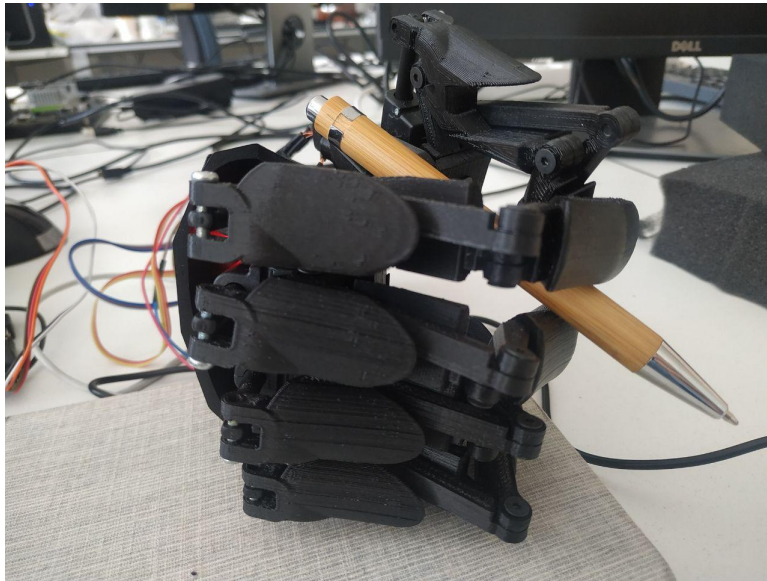
B.2 Testování řešení

Fotografie ke Kapitole 6.



B. FOTOGRAFIE Z TESTOVÁNÍ





Seznam použitých zkratk

USB Universal Serial Bus

LED Light-emitting diode

IDE Integrated Development Environment

PWM Pulzně šířková modulace

ICSP In-Circuit Serial Programming

SPI Serial Peripheral Interface

MISO Master In Slave Out

MOSI Master Out Slave In

SCLK Serial Clock

SS Slave Select

UART Universal asynchronous receiver-transmitter

USART Universal synchronous and asynchronous receiver-transmitter

XML Extensible Markup Language

GUI Graphical user interface

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
src	
├─ impl.....	zdrojové kody implementace
├─ documentation	dokumentace ke zdrojovým kódu aplikace
└─ BP_Popov_Ilia_2020	zdrojová forma práce ve formátu \LaTeX
text	text práce
└─ BP_Popov_Ilia_2020.pdf	text práce ve formátu PDF
videos.....	videa testování projektu
schemes.....	implementovaná schémata připojení