



STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

VÝUKOVÝ ROBOTICKÝ MODEL LIDSKÉ RUKY S BEZDRÁTOVÝMI ŘÍDICÍMI MODULY

AUTOR	Martina Hanusová
ŠKOLA	Gymnázium a Střední průmyslová škola elektrotechniky a informatiky, Frenštát pod Radhoštěm, příspěvková organizace
KRAJ	Moravskoslezský
ROČNÍK	Tercie, obor osmileté gymnázium
OBOR	10. Elektrotechnika, elektronika a telekomunikace



STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

VÝUKOVÝ ROBOTICKÝ MODEL LIDSKÉ RUKY S BEZDRÁTOVÝMI ŘÍDICÍMI MODULY

EDUCATIONAL ROBOTIC MODEL OF A HUMAN HAND WITH WIRELESS CONTROL MODULES

AUTOR	Martina Hanusová
ŠKOLA	Gymnázium a Střední průmyslová škola elektrotechniky a informatiky, Frenštát pod Radhoštěm, příspěvková organizace
KRAJ	Moravskoslezský
ŠKOLITEL	Ing. Marcel Hanus, Ph.D.
OBOR	10. Elektrotechnika, elektronika a telekomunikace

Prohlášení

Prohlašuji, že svou práci na téma *Výukový robotický model lidské ruky s bezdrátovými řídicími moduly* jsem vypracovala samostatně pod vedením Ing. Marcela Hanuse, Ph.D. a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Dále prohlašuji, že tištěná i elektronická verze práce SOČ jsou shodné a nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a změně některých zákonů (autorský zákon) v platném znění.

V Kunčicích pod Ondřejníkem dne 20. 3. 2019

Podpis:



Gymnázium a Střední průmyslová škola elektrotechniky a informatiky,
Frenštát pod Radhoštěm, příspěvková organizace



Poděkování

Děkuji svému otci Marcelovi za mé vedení v této práci, podnětné připomínky, nekonečnou trpělivost a za pomoc při nastavení registrů akcelerometrů. Dále děkuji svému bratrovi Michalovi za nápady a tipy k programování webserveru a mamince Šárce za pomoc se šitím rukavice s akcelerometry.

V neposlední řadě děkuji svým učitelům: panu Ing. Liboru Otáhalíkovi za skvělý kroužek Internetu věcí s ESP32 a panu Mgr. Richardu Štěpánovi za skvělý kroužek robotiky, kteří a které mě k této práci přivedli.

Nakonec děkuji autorům šablony SOČ Lucii Vaškeové, Jaroslavu Páralovi a Romanu Beránkovi, která mi usnadnila psaní této práce.

Anotace

Tato práce navazuje na mou loňskou SOČ s názvem *Univerzální řídicí systém pro ovládání výukových robotických manipulátorů*, jejíž cílem bylo vytvořit univerzální, levný, dostupný a snadno obsluhovatelný řídicí systém pro ovládání servomotorů, bipolárních krokových motorů a senzorů desítek různých typů výukových robotických manipulátorů, které si jde snadno vytisknout na 3D tiskárnách z návodů na internetu.

V letošní práci jsem se zaměřila na výběr a úpravu konstrukce jediného typu robotického manipulátoru – pohyblivého modelu lidské ruky vytisknutelného na 3D tiskárně, který by měl být pro výuku na školách dostatečně atraktivní a zároveň levný a snadno sestavitelný i opravitelný. Pro řízení robotické ruky jsem vybrala mikrokontrolér ESP32, který umožňuje její bezdrátové ovládání přes WiFi i Bluetooth, včetně zapojení do sítě Internetu věcí (IoT). Robotická ruka je tak ovládatelná přes svůj webservice z externího počítače, mobilu nebo tabletu, případně napřímo přes Bluetooth mobilní aplikaci, stejně dobře k ní lze ale připojit i širokou řadu různých bezdrátových řídicích modulů s mikrokontroléry ESP32.

Jako příklad jednoduchého bezdrátového řídicího modulu jsem v této práci zvolila rukavici s pěti akcelerometry, které sledují polohu prstů lidské ruky a přenášejí jejich signál bezdrátově na robotickou ruku, která tak pohyb prstů kopíruje. K nabíjení Li-Pol akumulátoru napájející rukavici jsem sestavila jednoduchý nabíjecí obvod včetně krabičky pro jeho uchycení.

Dalšími řídicími moduly, o které mám v plánu tuto výukovou sadu v budoucnu rozšířit, jsou například moduly ovládající pohyb robotické ruky pomocí kontrakce svalů, mimiky, hlasu, mozkových impulzů nebo pokročilých mikrovlnných technologií (Google Soli).

Popsaná snadno sestavitelná a opravitelná výuková pomůcka s výrobními náklady okolo 3000 Kč je dle mého názoru schopna zatraktivnit školní výuku v několika studijních oborech - robotice, mechatronice, programování mikrokontrolérů, senzorů, síťových a mobilních aplikací a 3D tisku. Práce obsahuje kompletní návod na stavbu robotické ruky, rukavice s pěti akcelerometry a nabíječky lithiových akumulátorů, včetně dokumentace jejich mechanických částí, elektronického hardware a software. Návod je dokumentován v „open hardware + open software“ podobě a publikován na veřejném git serveru gitlab.com/MartinaH/SOC2019, který bude průběžně aktualizován o novinky a vylepšení.

Klíčová slova

robotická ruka, senzorická rukavice, ESP32, Bluetooth, WiFi, IoT. webservice, akcelerometry, 3D tisk

Annotation

This project is a follow-up of my last year's project called *Universal control system for educational robotic manipulators*, the aim of which was to develop a universal, cheap, available and easily manageable control system for servo motors, bipolar stepper motors and sensors for dozens different types of educational robotic manipulators printable on 3D printers using instructions published on the internet.

This year I focused my work on selection and design modification of a single type of a robotic manipulator – a moveable model of a human hand printable on a 3D printer, which should be attractive enough for school education, while being cheap, easy to assemble and easy to repair. I chose ESP32 microcontroller as a base for its control system, which enables its wireless control via WiFi and Bluetooth, including its connection to the Internet of Things (IoT). The robotic hand can be controlled via its webserver from an external computer, a mobile phone or a tablet, via Bluetooth point-to-point communication with mobile device and allows to connect a wide range of wireless control modules with ESP32 microcontrollers to it.

As an example of such a wireless control module, I made a glove with five accelerometers, which detect position of fingers of a human hand and transfer their signal wirelessly to the robotic hand, which copies the fingers movements. For charging the Li-Pol battery on the glove I made a simple charging circuit including a box for its fixing.

In my following work I plan to expand this educational set with other wireless control modules, allowing to control the robotic hand movement with signals from muscle contractions, mimics, voice, brain impulses or advanced microwave technologies (Google Soli).

The easy-to-assemble-and-repair educational kit described in this project can be made for about 3000 CZK and, in my opinion, it has a potential to make a school education more attractive in several subjects – robotics, mechatronics, 3D printing, programming of microcontrollers, sensors, network and mobile applications. This project contains complete instructions for making the robotic hand, the glove with five accelerometers and the Li-Pol battery charger, including documentation of its mechanical parts, electronic hardware and software. The instructions are published as “open hardware + software” on a public git server gitlab.com/MartinaH/SOC2019, which will be continuously updated for the design upgrades and improvements.

Keywords

robotic hand, sensoric glove, ESP32, Bluetooth, WiFi, webserver, accelerometers, 3D print

Obsah

ÚVOD.....	10
1 ESP32 JAKO UNIVERZÁLNÍ IOT PLATFORMA	12
2 ROBOTICKÁ RUKA.....	14
2.1 MECHANICKÁ KONSTRUKCE ROBOTICKÉ RUKY.....	14
2.1.1 <i>Původní díly robotické ruky konstruktéra grosrc</i>	14
2.1.2 <i>Vlastní díly nakreslené v programu SketchUp</i>	15
2.1.3 <i>Návod na sestavení mechanického modelu robotické ruky</i>	17
2.2 ELEKTRONICKÝ HARDWARE ROBOTICKÉ RUKY.....	27
2.2.1 <i>Deska plošného spoje</i>	28
2.2.2 <i>Návod na kompletaci robotické ruky s řídicí elektronikou</i>	30
2.3 SOFTWARE PRO KOMUNIKACI S BEZDRÁTOVÝMI MODULY	35
2.4 SOFTWARE PRO WEBSERVER	39
2.5 SOFTWARE PRO MOBILNÍ BLUETOOTH APLIKACI	42
2.5.1 <i>Výpis programu pro Bluetooth komunikace s Pocket Code v ESP32</i>	43
2.5.2 <i>Výpis programu pro mobilní aplikaci v Pocket Code</i>	44
3 RUKAVICE S AKCELEROMETRY	47
3.1 HLAVNÍ DESKA PRO RUKAVICI.....	48
3.2 PŘEVODNÍ DESKY S AKCELEROMETRY A FPC KONEKTOREM	51
3.3 NABÍJEČKA LI-POL AKUMULÁTORŮ	52
3.3.1 <i>Mechanický hardware nabíječky</i>	52
3.3.2 <i>Elektronika nabíječky</i>	53
3.4 NÁVOD NA STAVBU RUKAVICE S AKCELEROMETRY	55
3.4.1 <i>Potřebné komponenty</i>	55
3.4.2 <i>Krok 1: Provrtání DPS</i>	56
3.4.3 <i>Krok 2: Přišití kapes pro akcelerometry</i>	56
3.4.4 <i>Krok 3: Přišití hlavní desky k rukavici a propojení s akcelerometry</i>	57
3.4.5 <i>Krok 4: Programování rukavice</i>	58
3.5 SOFTWARE PRO ESP32 NA RUKAVICI S AKCELEROMETRY	59
ZÁVĚR	65
SEZNAM POUŽITÝCH ODKAZŮ	67
SEZNAM OBRÁZKŮ	68
SEZNAM TABULEK.....	70
SEZNAM ZKRATEK.....	71
SEZNAM PŘÍLOH	71

Úvod

Robotizace průmyslové výroby pokračuje rychlejším tempem, než jsme si uměli před několika lety představit. Robotické manipulátory postupně vytlačují lidskou pracovní sílu ve všech výrobních operacích, které vyžadují opakované činnosti, při manipulaci s těžkými předměty, při rizikových činnostech nebo úkonech vyžadujících velkou manipulační přesnost a zvládají to mnohem rychleji, stabilněji a precizněji, než by to byl schopen zvládnout člověk, navíc bez nároků na mzdu a pravidelný odpočinek.

Se zrychlujícím se tempem robotizace, zvyšováním spolehlivosti a snižováním výrobní ceny robotů lze předpokládat, že všechny výrobní profese provádějící opakované a jednoznačně definovatelné mechanické operace budou nahrazeny roboty.

Česká republika jako průmyslová země, jejíž rozložení průmyslu je z velké části závislé na relativně jednoduchých montážních operacích (automobilový průmysl, montáž elektrotechnických a elektronických výrobků, strojná výroba), je podle některých studií^{[1][2]} zemí s nejvyšším ohrožením pracovních míst robotizací v Evropě.

Na druhou stranu je u nás nedostatek kvalifikovaných pracovníků schopných tyto průmyslové roboty vyvíjet, programovat, udržovat a opravovat a tento nedostatek bude logicky s postupujícím tempem robotizace dále narůstat, stejně jako nároky na rekvalifikaci nově nepotřebné pracovní síly a české odborné školství bohužel v tomto stále zaostává za požadavky současné i budoucí praxe.

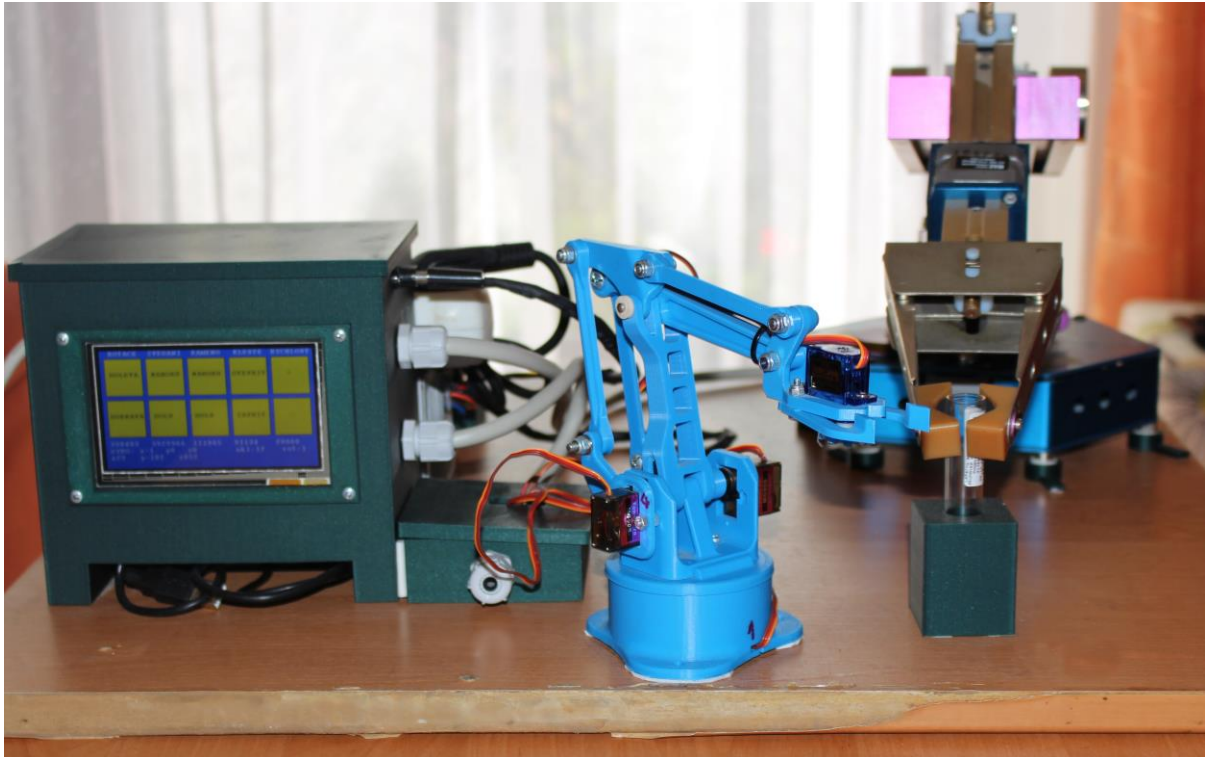
Proto je dle mého názoru důležité, aby se studenti středních odborných škol, stejně jako studenti rekvalifikačních kurzů pro dospělé, mohli ve své výuce co nejpodrobněji seznámit se základy robotiky, mechatroniky, 3D tisku, programování řídicích jednotek, síťových i mobilních aplikací, pokud možno na dostatečně atraktivních a jednoduchých výukových pomůckách.

K výuce je sice možné použít komerčně dostupné manipulátory, používané přímo ve výrobních linkách, tyto jsou však velmi drahé a většině škol chybí prostředky k jejich pořízení, nebo se mohou obávat o jejich poškození při výuce a s tím spojených nákladných oprav rušících jejich rozpočet, proto je raději studentům nedávají do ruky.

Protože podle mých zkušeností je nejefektivnějším způsobem výuky robotiky možnost si robotický výrobek sama sestavit, pochopit jeho konstrukci a postupně jej oživit elektronikou a programovacími funkcemi, měly by mít tyto výukové pomůcky k dispozici všichni žáci ve třídě, což vyžaduje, aby tyto výukové pomůcky byly levné a snadno vyrobitelné i opravitelné přímo ve školách. Zároveň by tyto pomůcky měly být pro studenty dostatečně atraktivní, aby sami měli zájem o jejich studium a hlubší zkoumání a nebyly to jen nudné statické modely.

Proto jsem se v rámci loňské SOČ rozhodla sestavit univerzální řídicí systém, který by byl schopen ovládat většinu z několika desítek typů různých robotických manipulátorů, které si jde v ceně několika stovek korun vytisknout na 3D tiskárnách z volně dostupných návodů na internetu (např. z www.thingiverse.com) a stejným způsobem levně nahradit i případně poškozené díly. Podle mých poznatků z různých studentských soutěží (např. Networking Academy Games) už jsou 3D tiskárny zavedeny ve velké části českých odborných škol, řada žáků je má doma nebo je možno objednat levný tisk v síti majitelů 3D tiskáren (www.3Dhubs.com).

Univerzální řídicí systém byl založený na vývojové desce DISCO-F746NG s barevným dotykovým displejem, čtyřech vývojových deskách s moderními motorovými drivery powerSTEP01 a umožňoval nejen uživatelsky pohodlné ovládání jedné robotické ruky se čtyřmi bipolárními krokovými motory, ale měl i dostatek volných signálů k připojení dalšího robotického manipulátoru se čtyřmi servomotory, pásového dopravníku s optickou závorou pro možnost sestavení složitější aplikace vzájemně spolupracujících robotů, reproduktoru a až několika stovek senzorů a dalších mikrokontrolérů komunikujících po I2C sběrnici umožňujících takřka nekonečné větvení řídicího systému na další součásti robotických linek. Řídicí systém bylo možno dále ovládat bezdrátově z mobilních zařízení pomocí Bluetooth komunikace. Materiálové náklady na sestavení navrženého řídicího systému, včetně osazených desek plošných spojů a krabiček vytištěných na 3D tiskárně, nepřekračují 3000 Kč bez DPH.



Obrázek 1: Foto výsledné sestavy univerzálního řídicího systému se dvěma robotickými manipulátory

Letos jsem se rozhodla zaměřit svou práci na jediný typ levného a jednoduchého robotického manipulátoru – uživatelsky atraktivní robotický model lidské ruky ovládaný pomocí bezdrátových technologií (WiFi, Bluetooth), který by byl použitelný i k výuce síťových technologií (IoT - internet věcí) a byl by do budoucna snadno rozšiřitelný o různé bezdrátové řídicí moduly, včetně pokročilých technologií, jako je ovládání pohybem těla, kontrakcí svalů, hlasem, mimikou či mozkovými impulzy, které by tuto pomůcku dělaly pro studenty zajímavější, než jsou stávající výukové pomůcky.

1 ESP32 jako univerzální IoT platforma

Pro levné a snadno rozšiřitelné bezdrátové ovládání robotické ruky i jejich řídicích modulů jsem vybrala moderní mikrokontrolér ESP32. Mikrokontrolér ESP32 je vyráběn čínskou firmou Espressif Systems (www.espressif.com) a od svého uvedení na trh v roce 2016 se stal velice populárním jak mezi profesionály, tak mezi hobby uživateli – spolu se svým předchůdcem ESP8266 jich bylo vyrobeno více než 100 miliónů kusů a jejich aplikace jsou častou náplní webů s hobby návody a tutoriály, jako jsou například www.instructables.com.

Atraktivitu ESP32 způsobuje souhrn několika faktorů^{[3][4]}:

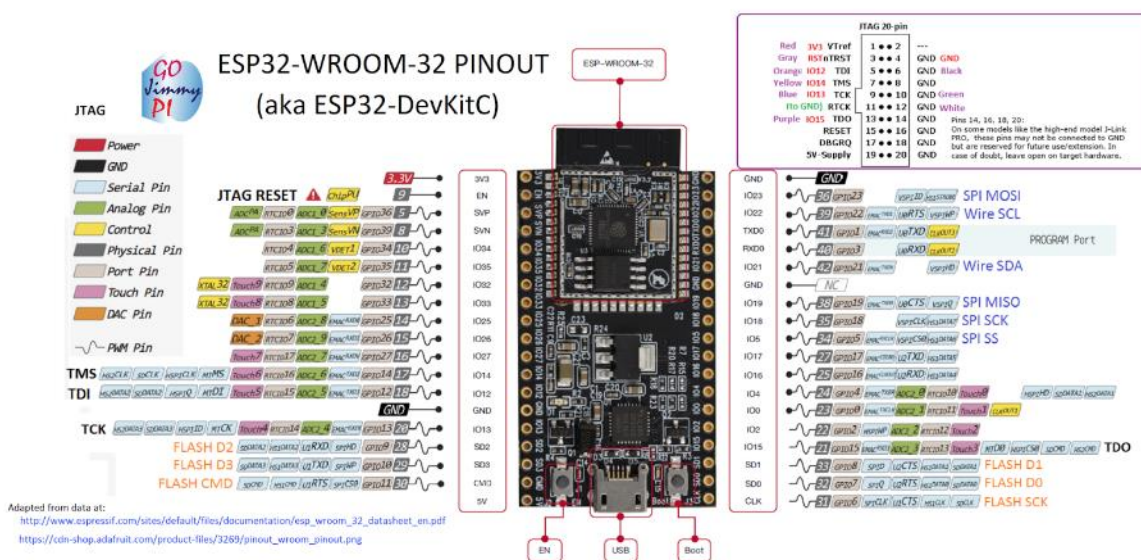
- velice výkonný dvoujádrový procesor pracující na frekvenci až 240 MHz;
- až 34 GPIO pinů se softwarově nastavitelnými funkcemi (analogový vstup, kapacitní dotek, Hallův senzor, interní teploměr, PWM, CAN, SPI, I2C, I2S, UART, ethernet);
- kompletní radiofrekvenční transceiver modulovaný pro vysílání a přijímání signálů v pásmu 2,4 GHz protokoly WiFi a Bluetooth 4.0, s dosahem až několik stovek metrů v otevřeném prostoru;
- velké množství dostupných modulů (např. Wrover, Wroom), obsahující kromě vlastního čipu ESP32 i externí SRAM a flash paměti o velikosti až několik MB, pasivní součástky, integrovanou anténu nebo konektor na externí anténu, vše na plošném spoji s roztečemi pájecích bodů 1,27 mm zvládnutelnými k pájení většinou hobby uživateli, v ceně okolo 100 – 150 Kč (cena samotného čipu ESP32 je menší než 100 Kč);
- dostupné vývojové moduly (např. DevKit), které obsahují integrovaný převodník USB/UART a výstupní pinovou řadu se standardní roztečí 2,54 mm a jsou tedy programovatelné přímo z USB portu počítače a zapojitelné do různých typů nepájivých polí;
- Zpracované vývojové rozhraní ESP-IDF, převedené do podoby funkcí kompatibilních s funkcemi prostředí Arduino IDE;
- Rozsáhlá a ochotná komunita uživatelů dostupná na různých fórech s publikovanými přehlednými řešeními nejčastějších problémů (např. na esp32.net).

Z těchto důvodů je dle mého názoru ESP32 velmi užitečnou náhradou široce zavedených modulů řady Arduino, které předčí nejen svými uživatelskými parametry, ale i pořizovací cenou (viz Tabulku 1 níže). Uživatel Arduina může na platformu ESP32 přejít velmi snadno, protože funkce a příkazy používané v Arduinu IDE jsou použitelné i k programování ESP32 v témže vývojovém prostředí.

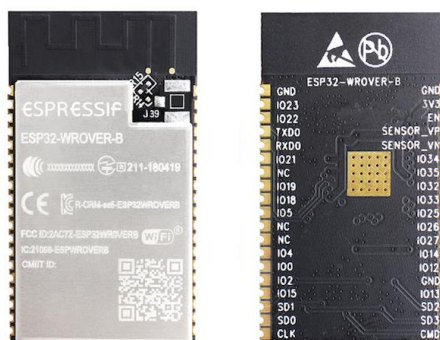
V této práci jsem použila dva typy modulů ESP32: Pro robotickou ruku snadněji programovatelný ESP32-DevKitC (obrázek 2), s pinovými řadami a možností programování přes USB a pro senzorickou rukavici ESP32 Wrover (obrázek 3), menší modul vhodnější pro pájení do miniaturnějších desek plošných spojů.

Tabulka 1: Srovnání ESP32 s Arduino UNO

	Arduino UNO ^[5]	ESP32 ^[4]
cena	636 Kč	226 Kč
počet GPIO	20	34
PWM	6	34
SPI	1x	4x
I2C	1x	2x
I2S	ne	2x
UART	1x	3x
CAN 2.0	ne	ano
WiFi	ne	ano
Bluetooth	ne	ano
Ethernet MAC	ne	ano
SD/eMMC	ne	host + slave
DAC	ne	2x 8bit
ADC kanály	6	18
Rozlišení ADC	10 bit	12 bit
dotykové senzory	ne	10x
frekvence oscilátoru	16 MHz	240 MHz
SRAM paměť	2 kB	520 KB + až 16 MB externí
Flash paměť	32 kB	448 kB + až 16 MB externí



Obrázek 2: Schéma pinů vývojové desky ESP32 DevKit^[6]



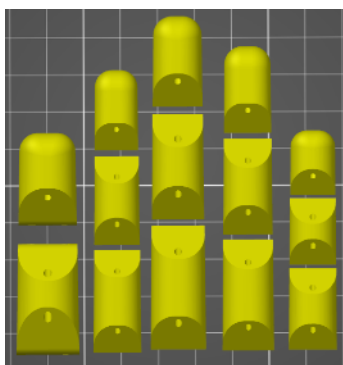
Obrázek 3: Foto ESP32 Wrover^[7]

2 Robotická ruka

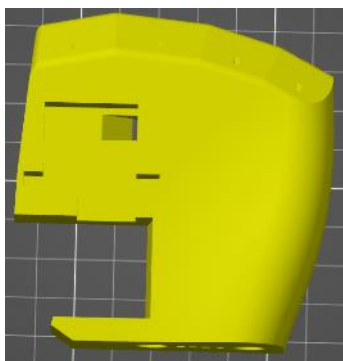
2.1 Mechanická konstrukce robotické ruky

Jako nejvhodnější model robotické ruky, dostatečně jednoduchý k výrobě i sestavení, jsem si vybrala model Humanoid Robotic Hand^[8] konstruktéra grossrc, publikovaný na webu Thingiverse s příznivou licencí CC-BY-NC 3.0, který jsem doplnila vlastními díly nakreslenými v programu SketchUp pro možnost upevnění levnějších servomotorů MG90S a řídicí desky s ESP32.

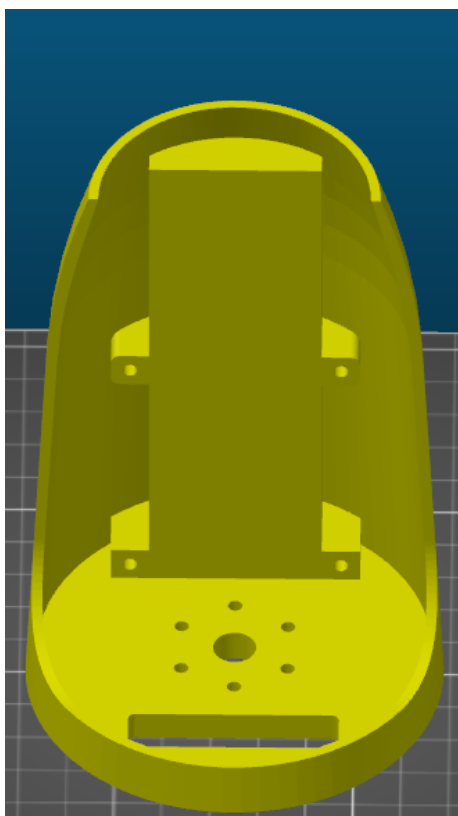
2.1.1 Původní díly robotické ruky konstruktéra grossrc



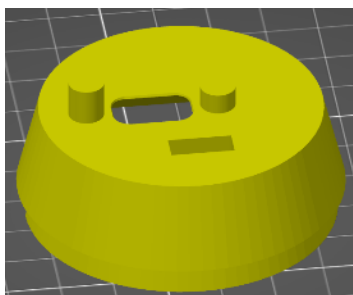
Obrázek 4: Modely prstů robotické ruky



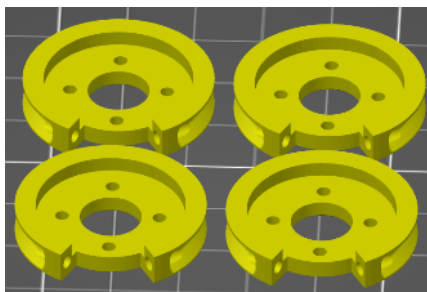
Obrázek 5: Model dlaně robotické ruky



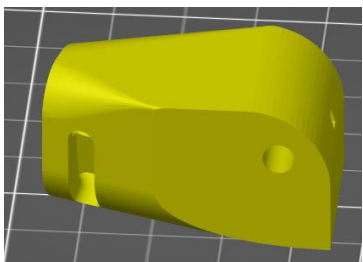
Obrázek 6: Model předloktí robotické ruky



Obrázek 7: Model zápěstí robotické ruky



Obrázek 8: Model navijáků silonu

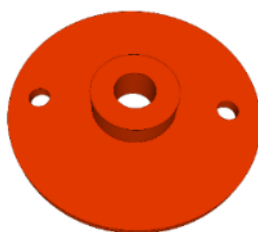


Obrázek 9: Model palce robotické ruky

2.1.2 Vlastní díly nakreslené v programu SketchUp



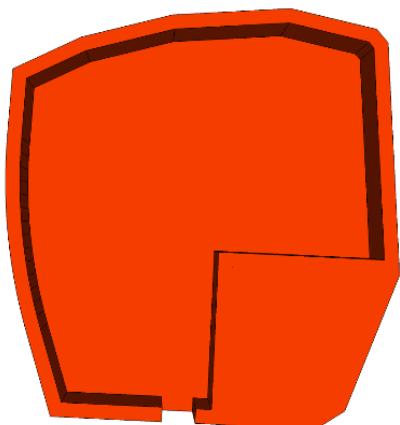
Obrázek 12: Model držáku elektroniky



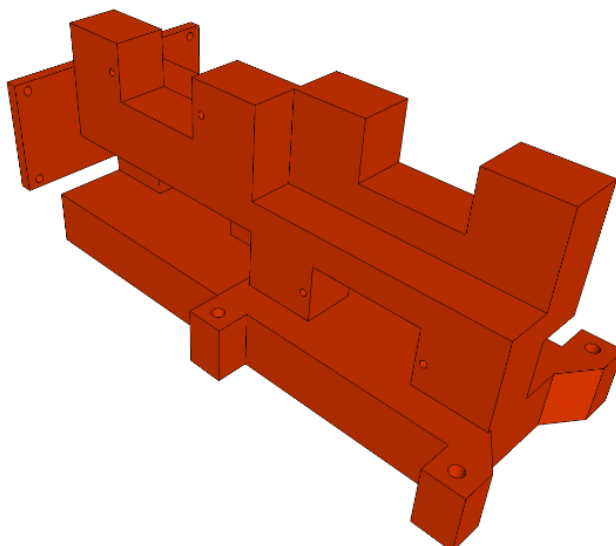
Obrázek 10: Model kolečka na otáčení předloktí



Obrázek 11: Model koleček na servomotory



Obrázek 13: Model krytky hřbetu ruky



Obrázek 14: Model držáku servomotorů

Díly nakreslené ve SketchUpu jsem vyexportovala do formátu STL a spolu s originálními díly konstruktéra grossrc dle předchozí kapitoly je vytiskla na 3D tiskárně Průša i3 z materiálu PLA měděné barvy, s použitím programu Sli3er ve standardním nastavení tiskových podmínek. Jednotlivé STL modely i soubor SketchUp s upravenými modely jsou obsaženy v příloze této práce.

Tabulka 2: Tabulka tisku 3D dílů modelu robotické ruky

Díl	Číslo obrázku	Podpěry	Doba tisku [h:m:s]	Hmotnost [g]	Cena struny [Kč]
prsty	4	NE	3:34:41	36,34	19,95
dlaň	5	NE	4:22:44	56,08	30,79
zápěstí	7	ANO	1:42:43	20,76	11,40
předloktí	6	NE	6:37:18	96,12	52,77
navijáky silonu	8	NE	0:43:03	6,74	3,70
palec	9	ANO	1:08:52	11,13	6,11
sloupek pro PCB	12	ANO	18:53:46	259,26	142,33
držák servomotorů	14	ANO	5:23:54	67,51	37,06
kolečka na serva	11	NE	0:26:41	4,46	2,45
kolečko na otáčení předloktí	10	NE	0:12:19	1,87	1,03
zadní krytka hřbetu ruky	13	NE	1:54:58	26,68	14,65
Celkem			45:00:59	586,95	322

2.1.3 Návod na sestavení mechanického modelu robotické ruky

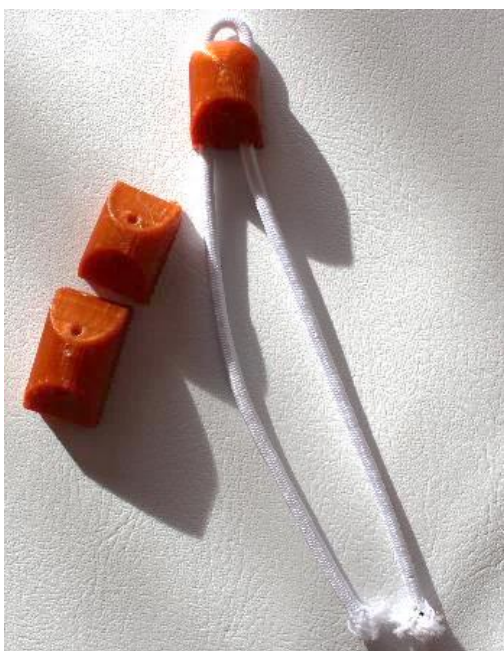
2.1.3.1 Potřebné komponenty



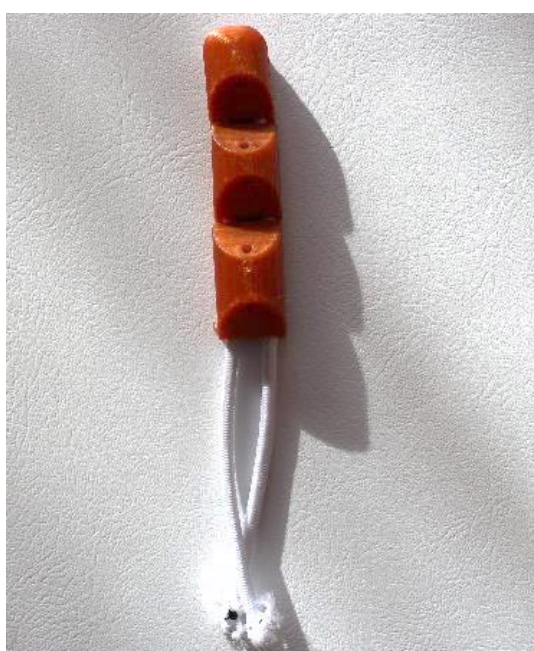
Obrázek 15: Potřebné komponenty ke stavbě mechanického modelu robotické ruky

2.1.3.2 Krok 1: Složení jednotlivých prstů

Ustříhnete kousek pruženky a prostrčte ji otvory v jednotlivých člancích prstů, jak vidíte na obrázcích níže.



Obrázek 16: Navlečení článků prstu na pruženku



Obrázek 17: Navlečené články prstu na pruženku

Jakmile budete mít navlečen celý prst, provlečte oba konce pruženky otvory v dlani a konce zavažte na uzel. To samé proved'te se zbývajícími prsty a ustříhnete konce pruženky.



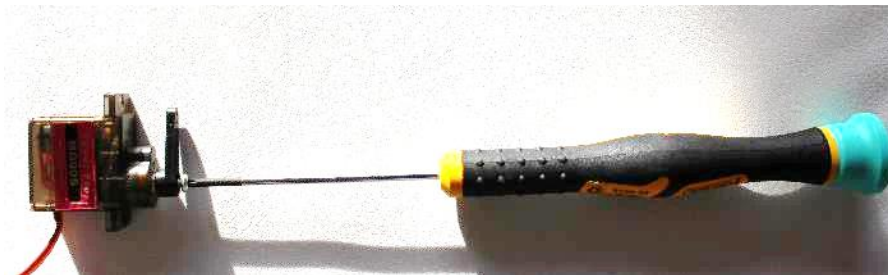
Obrázek 18: Připevnění malíčku k dlani



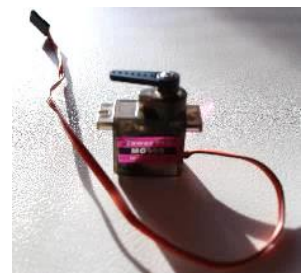
Obrázek 19: Připevnění zbylých prstů k dlani

2.1.3.3 Krok 2: Připevnění palce ke dlani

Přišroubujte k servomotoru MG90S unašeč (ujistěte se, že je servo ve správné krajní poloze, aby servomotor byl schopen po montáži pohybovat palcem).



Obrázek 20: Přišroubování unašeče k servomotoru



Obrázek 21: Servomotor s unašečem

Zhruba ve $\frac{3}{4}$ unašeč ustříhnete.



Obrázek 22: Ustřížení části unašeče



Obrázek 23: Servomotor s ustříženou částí unašeče

Vteřinovým lepidlem potřete otvor na palci a vložte do něj unašeč, jak vidíte na obrázcích.

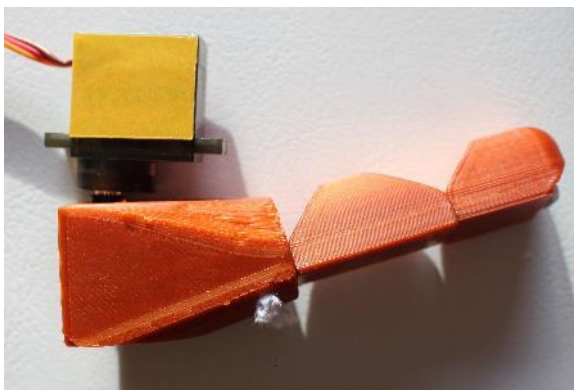


Obrázek 24: Nanesení lepidla do otvoru na palci

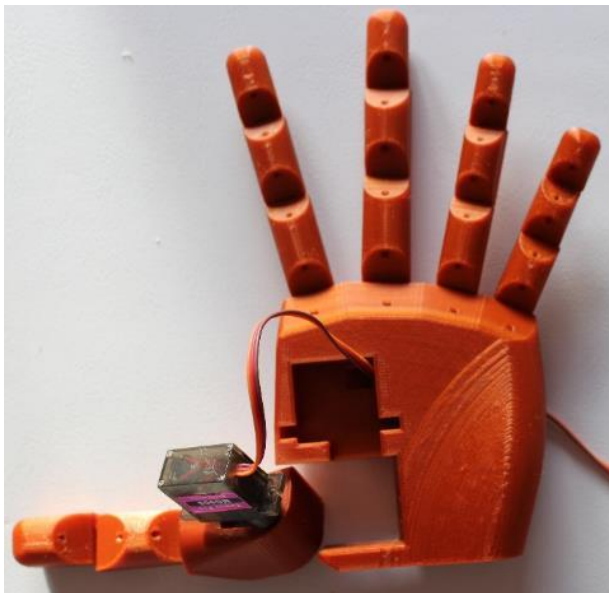


Obrázek 25: Vložení servomotoru do otvoru na palci

Servomotor přilepte tenkou oboustrannou lepicí páskou k dlani.



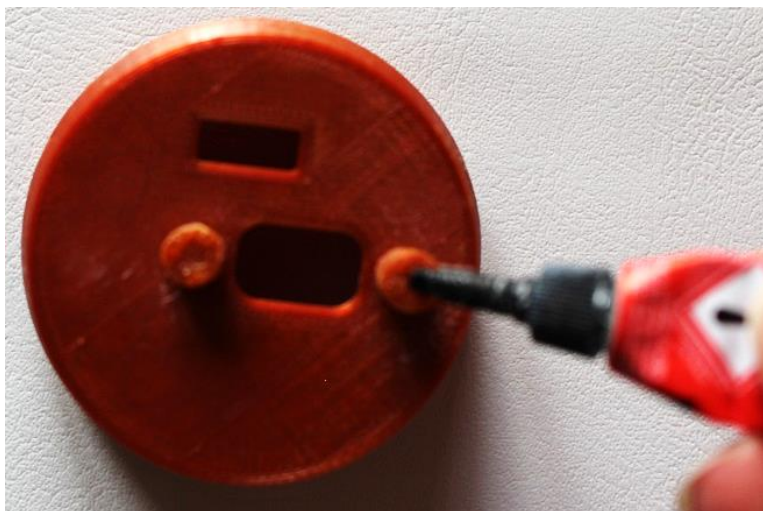
Obrázek 26 a Obrázek 27: Přilepení oboustranné lepicí pásky na servomotor



Obrázek 28 a Obrázek 29: Vložení servomotoru do otvoru v dlani

2.1.3.4 Krok 3: Složení předloktí

Vteřinovým lepidlem potřete výstupky na zápětním dílu a zasuňte je do již složené horní části ruky.

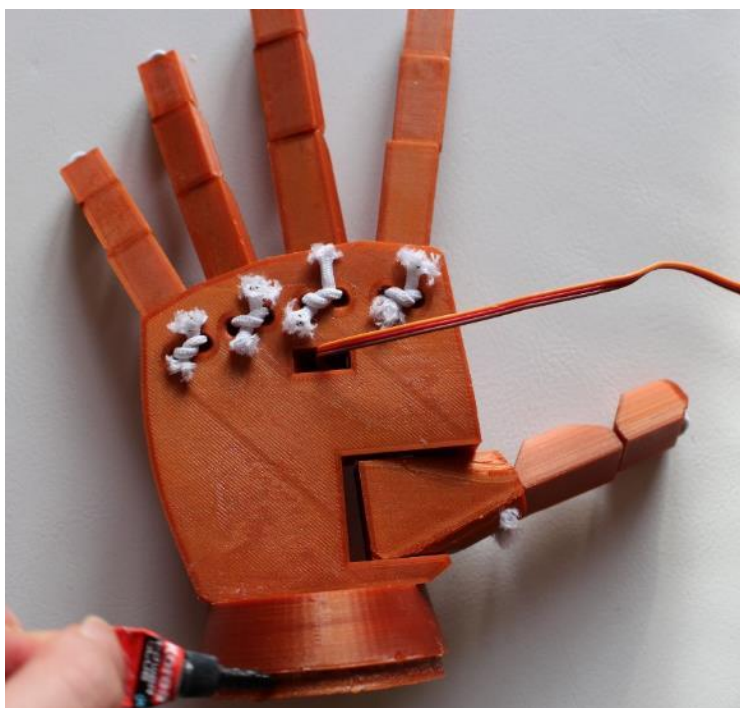


Obrázek 30: Nanesení lepidla na výstupky zápěstí



Obrázek 31: Přilepení zápěstí k dlani

Po zaschnutí lepidla potřete lepidlem i spodní okraj zápětního dílu a přilepte k předloktnímu dílu, jak vidíte na následujících obrázcích.



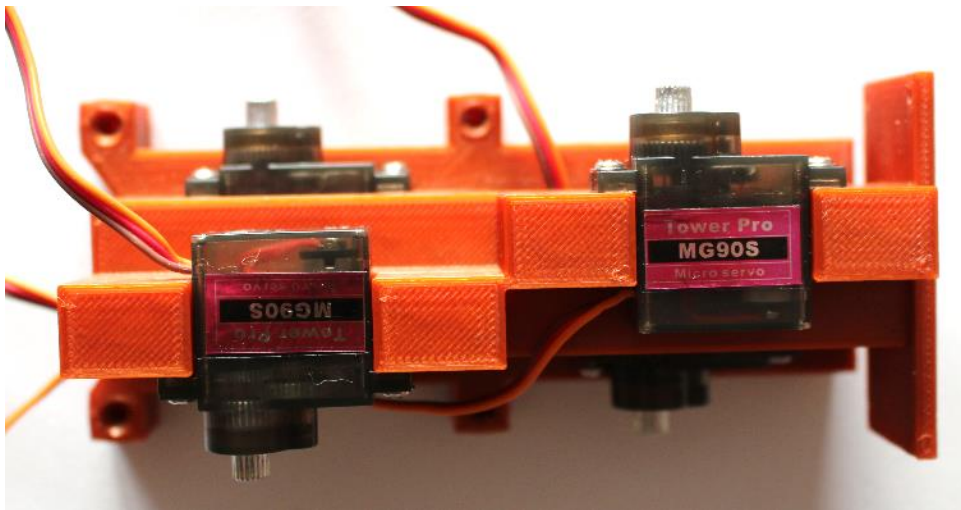
Obrázek 32: Nanesení lepidla na spodní okraj zápěstí



Obrázek 33: Přilepení zápěstí k předloktí

2.1.3.5 Krok 4: Zkompletování servomotorů

Přišroubujte servomotory MG90S do držáku, jak vidíte na tomto obrázku.



Obrázek 34: Přišroubování servomotorů k držáku

Kolečka na servomotory potřete vteřinovým lepidlem a vložte do navijáků silonu

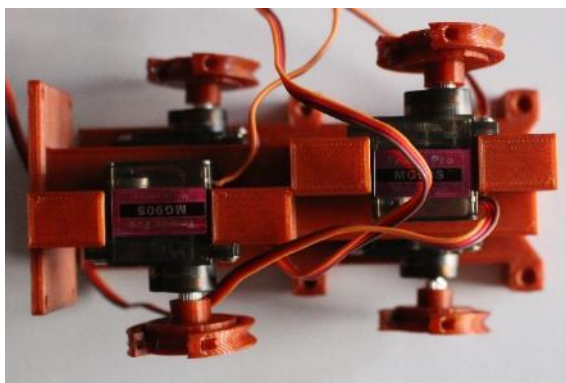


Obrázek 35: Nanesení lepidla na kolečko pro servomotor



Obrázek 36: Přilepení koleček k navijákům silonu

Kolečka s navijáky pak nasadíte natěsno na hřídele servomotorů.

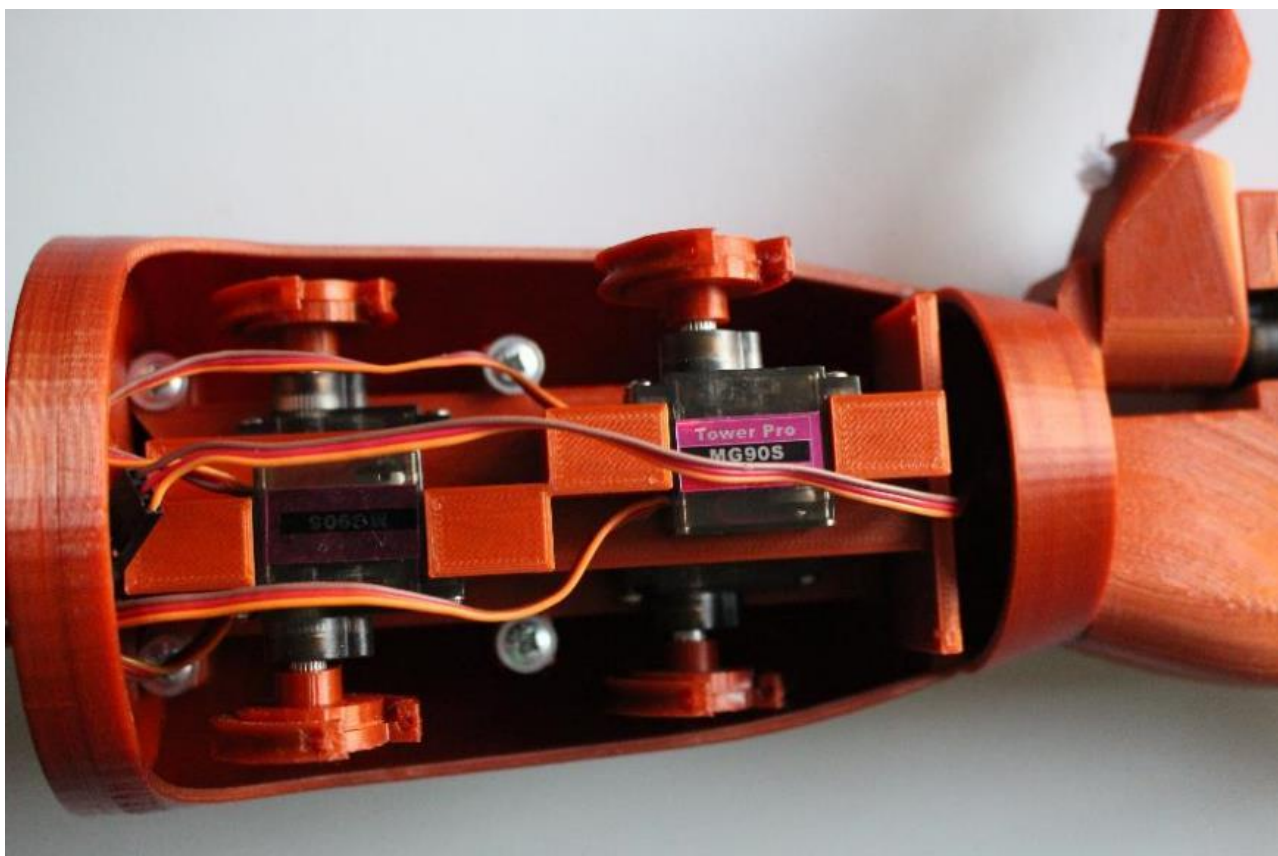


Obrázek 37: Nasazení koleček s navijáky na servomotory



Obrázek 38: Kolečka nasazená na servomotorech

Celou sestavu připevněte k předloktí pomocí čtyř vrtů M4x25.



Obrázek 39: Přišroubování držáku servomotorů k předloktí

2.1.3.6 Krok 5: Propojení prstů se servomotory

Z rybářského vlasce ustříhnete pět zhruba 35cm kousků a udělejte na konci uzly.



Obrázek 40: Rybářský vlasce



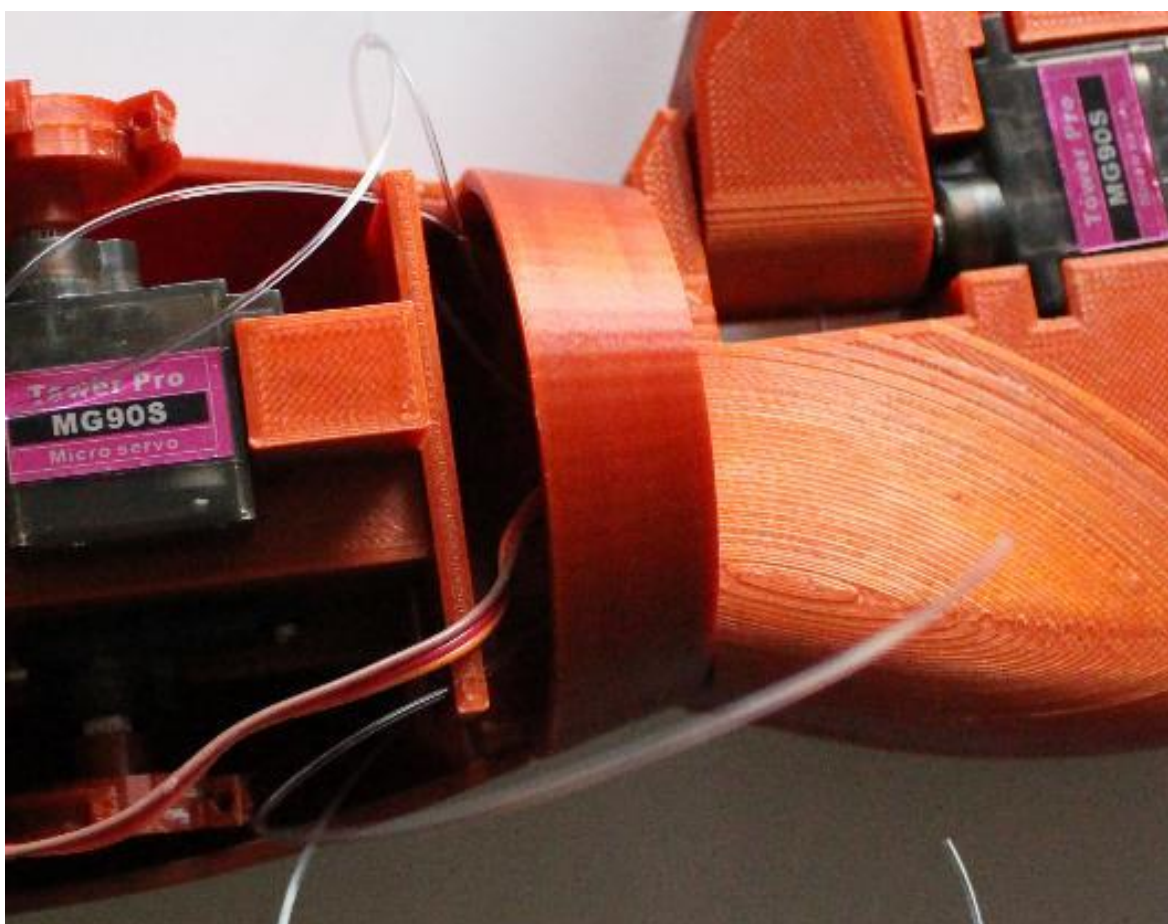
Obrázek 41: Kousky rybářského vlasce s uzly na konci

Vlasec provlečte skrz jednotlivé prsty



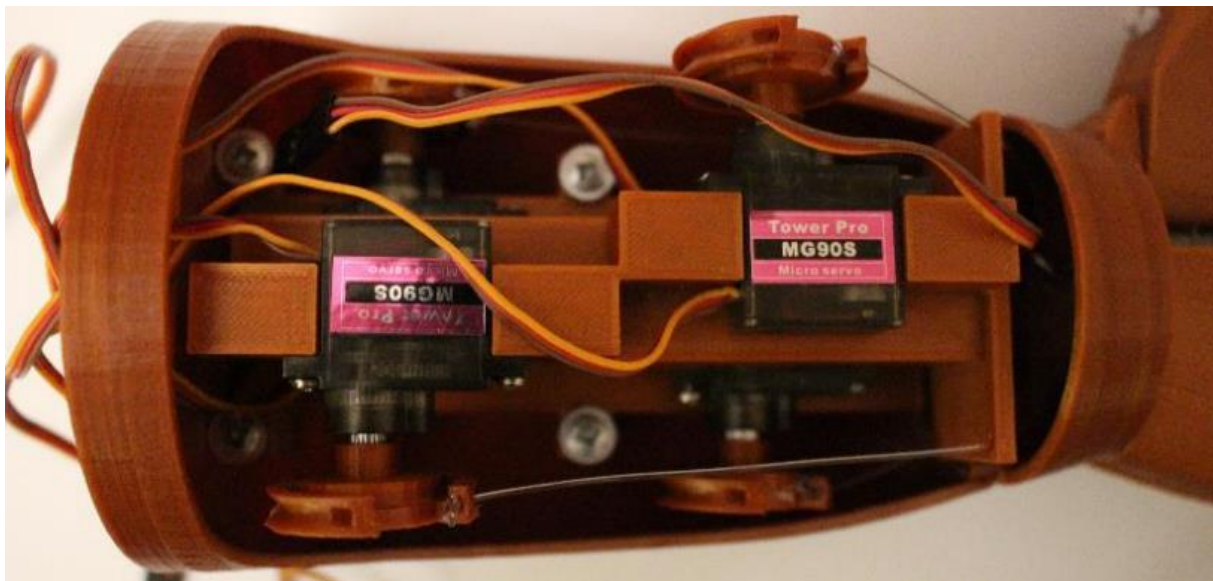
Obrázek 42, Obrázek 43 a Obrázek 44 : Provlečení silonu články prstů

a poté i skrz otvory v držáku servomotorů:



Obrázek 45 : Provlečení silonu otvory v držáku servomotorů

Nakonec vlasec provlečte i skrz otvory v navijácích upevněných na servomotoru a na konci udělejte uzly.



Obrázek 46 : Provlečení silonu navijáky a zauzlování na konci

2.1.3.7 Krok 6: Přilepení zadní krytky na hřbet ruky

Oboustrannou lepicí páskou oblepte celý horní okraj krytky na hřbet ruky.



Obrázek 47 : Zadní krytka na hřbet



Obrázek 48 : Oblepení okraje krytky oboustrannou lepicí páskou

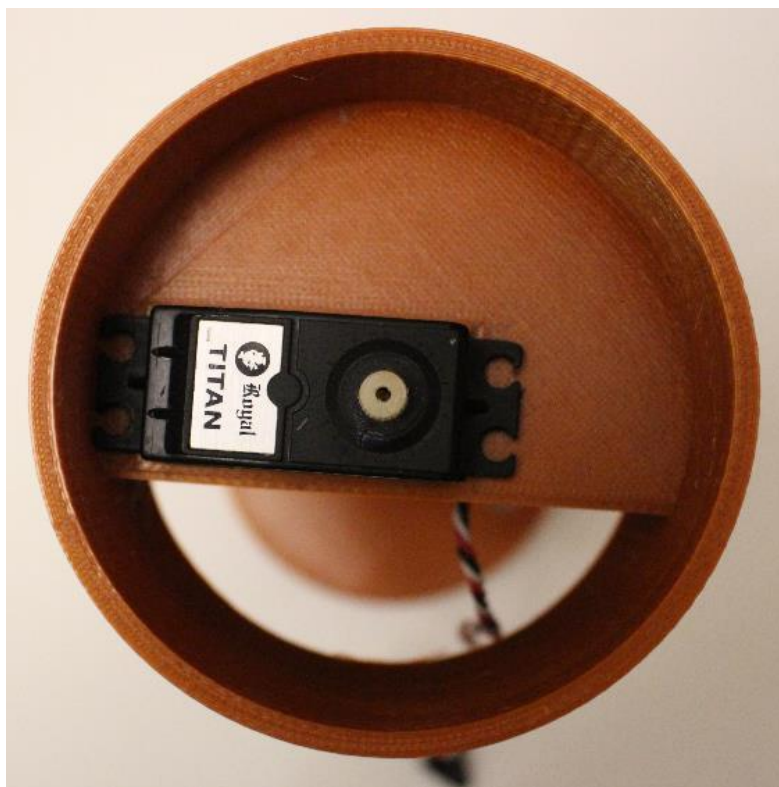
Poté přilepte celou krytku na hřbetní část robotické ruky.



Obrázek 49 : Pohled na robotickou ruku zezadu

2.1.3.8 Krok 7: Sestavení otočného zápěstí

Do spodního sloupku pro DPS vložte servomotor MG995 nebo rozměrově ekvivalentní typ.



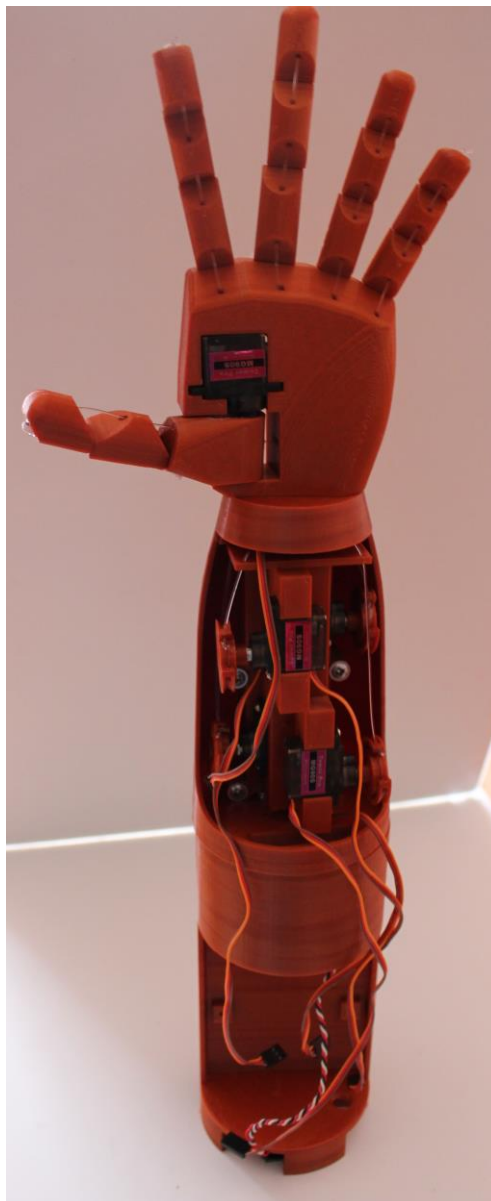
Obrázek 50 : Vložení servomotoru do sloupku pro DPS

Přišroubujte kolečko pro otáčení předloktím k předloketnímu dílu dvěma šroubky M3x12.



Obrázek 51 : Přišroubování kolečka pro otáčení předloktím k předloketnímu dílu

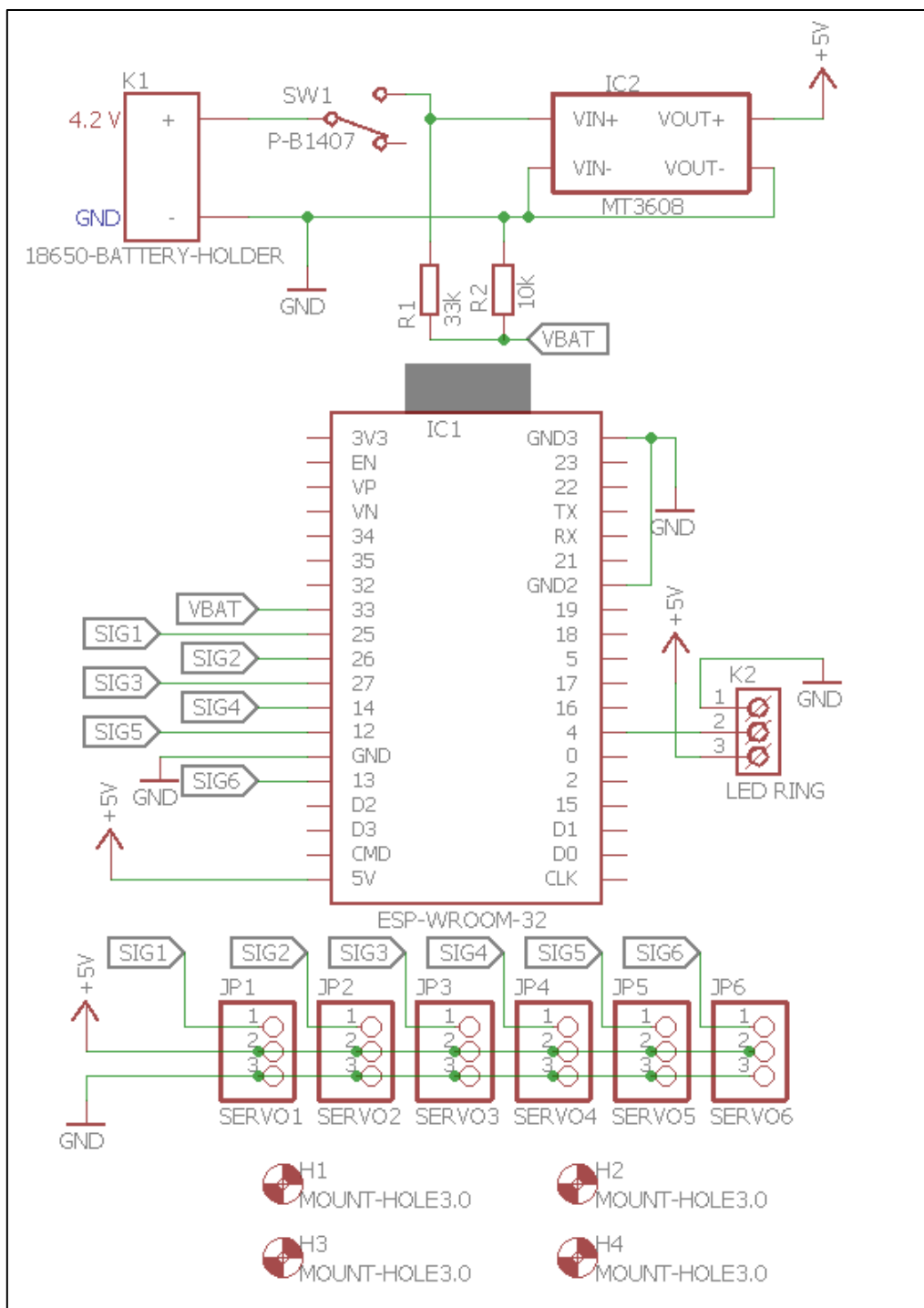
Kolečko pak zasuňte na hřídel servomotoru umístěného ve sloupku.



Obrázek 52: Nasazení horního dílu robotické ruky na spodní sloupek pro elektroniku

2.2 Elektronický hardware robotické ruky

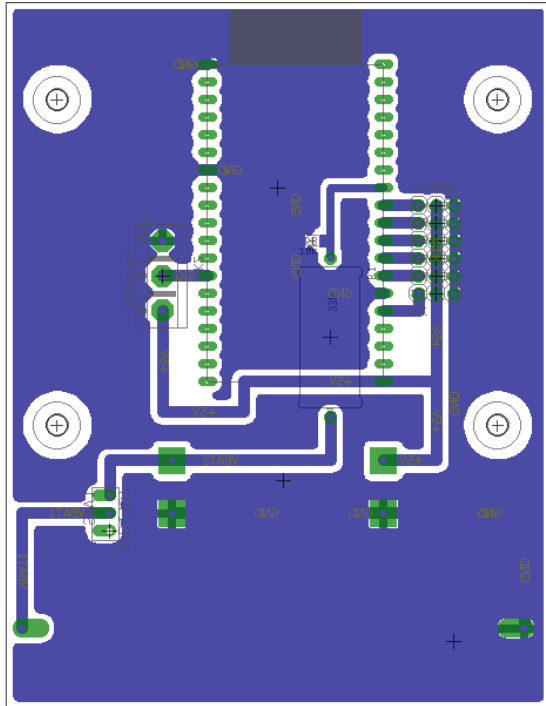
Řídicí systém robotické ruky je založen na mikrokontroléru ESP32 v podobě jeho vývojové desky ESP32-DevKitC, která pomocí PWM signálu ovládá šest servomotorů ruky a současně programuje i programovatelný kroužek (ring) s 16 RGB LED s integrovaným driverem WS2812, který slouží k vizuální indikaci stavu baterie a může být použit pro různé barevné efekty. Řídicí systém je napájen z jediného lithiového akumulátoru typu 18650 (3,6V / 2200 mAh), jehož napětí je převedeno na potřebných 5V pomocí modulu step-up měniče MT3608. Stav akumulátoru je měřen odporovým děličem zavedeným na analogový vstup ESP32.



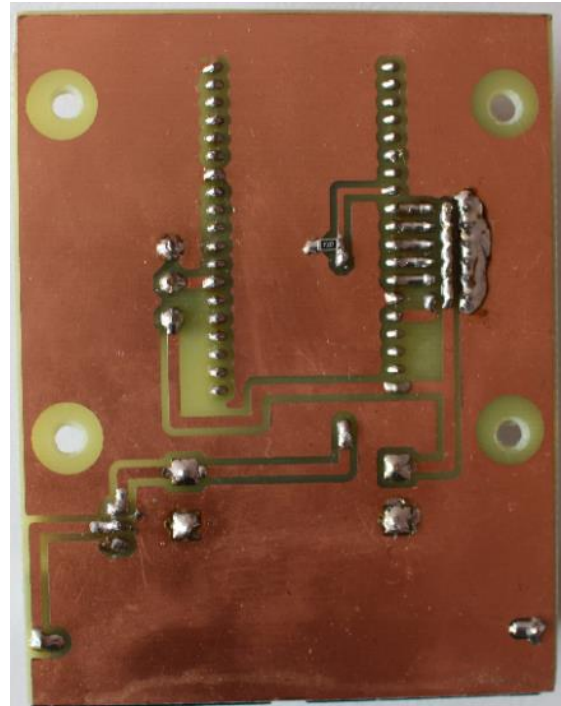
Obrázek 53: Schéma zapojení řídicího systému robotické ruky

2.2.1 Deska plošného spoje

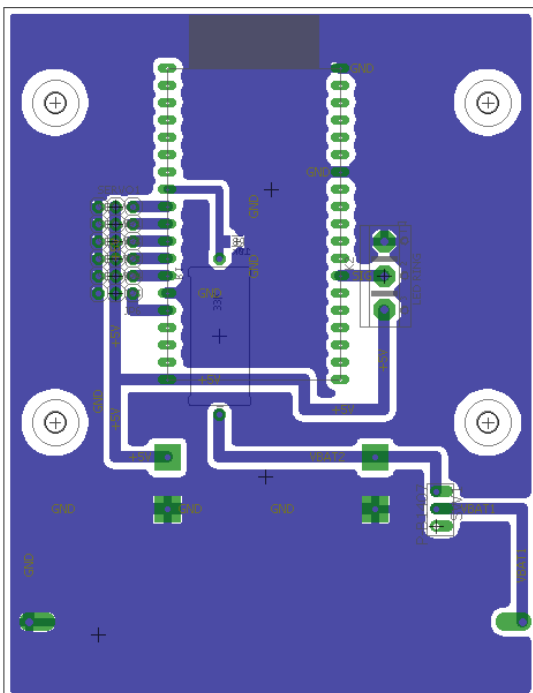
Pro řídicí systém robotické ruky byla vyrobena jednovrstvá deska plošného spoje (DPS), klasickou cestou s osvětlením fotocitlivé vrstvy UV lampou, vyvoláním v 1% roztoku hydroxidu sodného a vypleváním v 20% roztoku persíranu sodného. Schéma plošného spoje bylo nakresleno v programu Eagle CAD 7.7 a je obsaženo v elektronické příloze této práce.



Obrázek 54: Pájecí schéma DPS robotické ruky
– strana spojů



Obrázek 55: Foto DPS robotické ruky
– strana spojů



Obrázek 56: Pájecí schéma DPS robotické ruky
– strana součástek



Obrázek 57: Foto DPS robotické ruky
– strana součástek

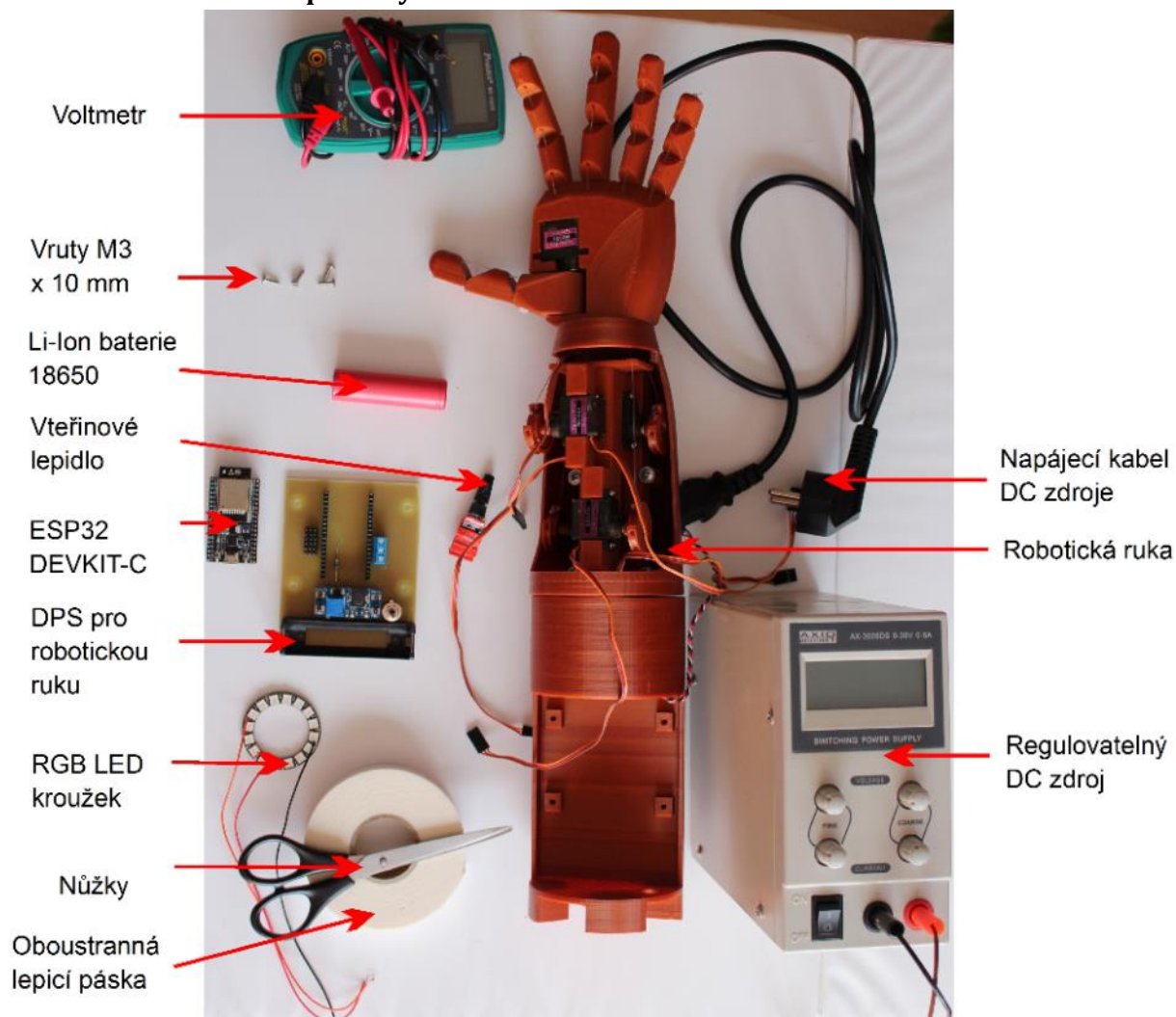
Tabulka 3: Tabulka elektronických součástí robotické ruky

Kód	Typ	Parametry	Funkce	Cena [Kč]	Zdroj
IC1	vývojová deska	ESP32 DEVKIT-C	mikrokontrolér řídícího systému	226	TME ^[9]
IC2	step up měnič	MT3608	převádí napětí akumulátoru na 5V	28	Laskarduino ^[10]
JP1 – JP6	pinové řady	3x1	připojení servomotorů	11	TME ^[9]
K1	držák baterie	pro 18650	připojení Li-Ion akumulátoru	21	TME ^[9]
K2	svorkovnice	ARK500/3	připojení RGB LED kroužku	6	TME ^[9]
R1	rezistor	33 kΩ THT	odporový dělič	0,5	TME ^[9]
R2	rezistor	10 kΩ SMD 1206	odporový dělič	0,2	TME ^[9]
SW1	vypínač	páčkový 3piny e = 5,08 mm	vypínač napájení	12	TME ^[9]
	akumulátor	Li-Ion 18650	napájení obvodu	134	TME ^[9]
	DPS s fotocitlivou vrstvou 160x100 mm	FR4 100x160/3500	deska plošného spoje	58	TME ^[9]
	Kroužek 16 x RGB LED WS2812	POLOLU – 2537	signalizace stavu baterie	319	TME ^[9]
	servomotory (5x)	MG90S	pohyb prstů	550	Aukro ^[11]
	servomotor	MG995	otáčení předloktím	159	Aukro ^[11]
Celkem:				1525	

K nabíjení Li-ion akumulátoru typu 18650 je dále vhodné pořídit nabíječku, a to buď některý typ dostupný v e-shopech za cenu okolo 15 Kč (např. TP4056^[12]) nebo vyrobit druhý kus nabíječky popsané v kapitole 3.5 a použít jeho programovací rezistor s hodnou 2 kΩ pro rychlejší nabíjení.

2.2.2 Návod na kompletaci robotické ruky s řídicí elektronikou

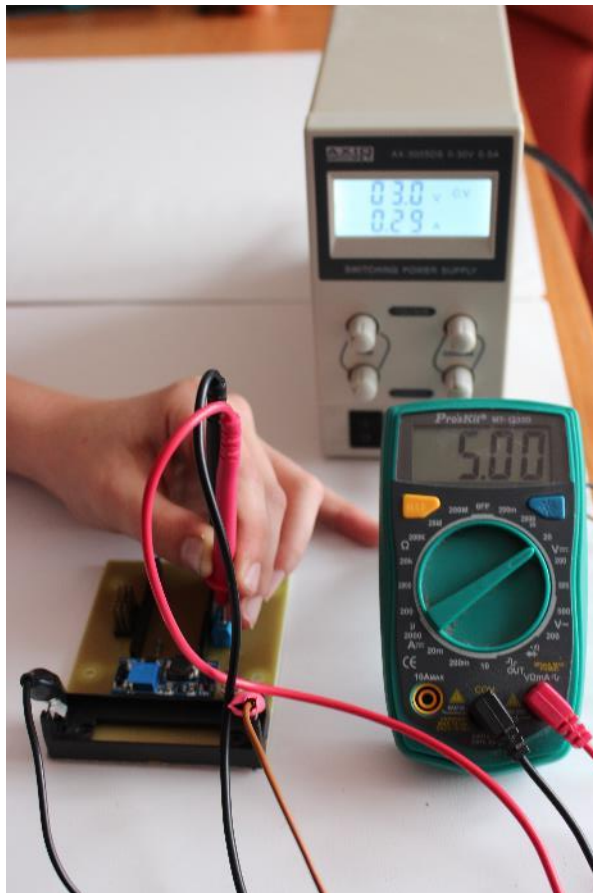
2.2.2.1 Potřebné komponenty



Obrázek 58: Potřebné komponenty ke kompletaci robotické ruky s elektronikou

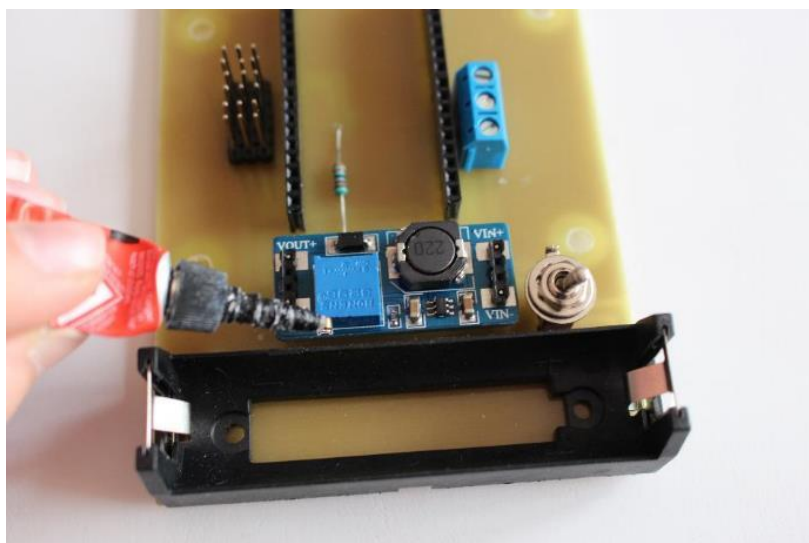
2.2.2.2 Krok 1: Nastavení step up měniče na 5V

Nastavte na laboratorním zdroji napětí 3 V a připněte jeho výstupní svorky k držáku baterie ve správné polaritě (**POZOR: NEZAPOJUJTE ZATÍM MIKROKONTROLÉR !!!**). Měřte na výstupu ze stabilizátoru napětí a otáčejte potenciometrem na modulu měniče, dokud nenaměříte napětí 5 V.



Obrázek 59: Nastavení step up měniče na 5V

Zakápněte potenciometr vteřinovým lepidlem, aby nedošlo ke změně výstupní hodnoty při manipulaci s DPS.



Obrázek 60: Zakápnutí potenciometru lepidlem

2.2.2.3 Krok 2: Vložení baterie a mikrokontroléru

Do patice na DPS vložte ESP32 a do držáku baterie vložte baterii, jak vidíte na obrázku:



Obrázek 61: DPS robotické ruky bez ESP32



Obrázek 62: DPS robotické ruky s ESP32

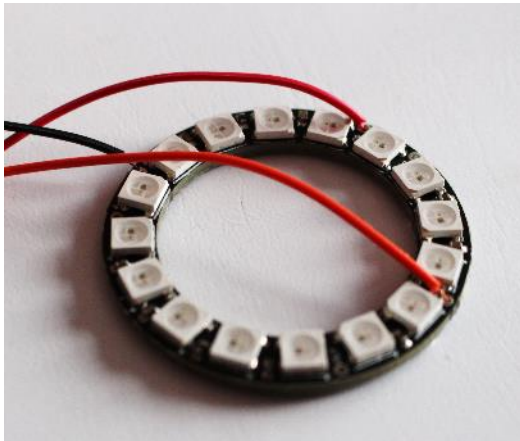
2.2.2.4 Krok 3: Montáž DPS do držáku ruky a připojení RGB LED kroužku

Pomocí vrutů M3x10 přišroubujte DPS do spodního sloupku robotické ruky a zapojte kabely servomotorů do pinových řad



Obrázek 63: DPS přišroubovaná k robotické ruce

Připájejte na RGB LED kroužek přívodní dráty (černý na GND, červený na 5V a např. oranžový na OUT) a oboustrannou lepicí páskou kroužek přilepte zespod na místo ve dně robotické ruky.

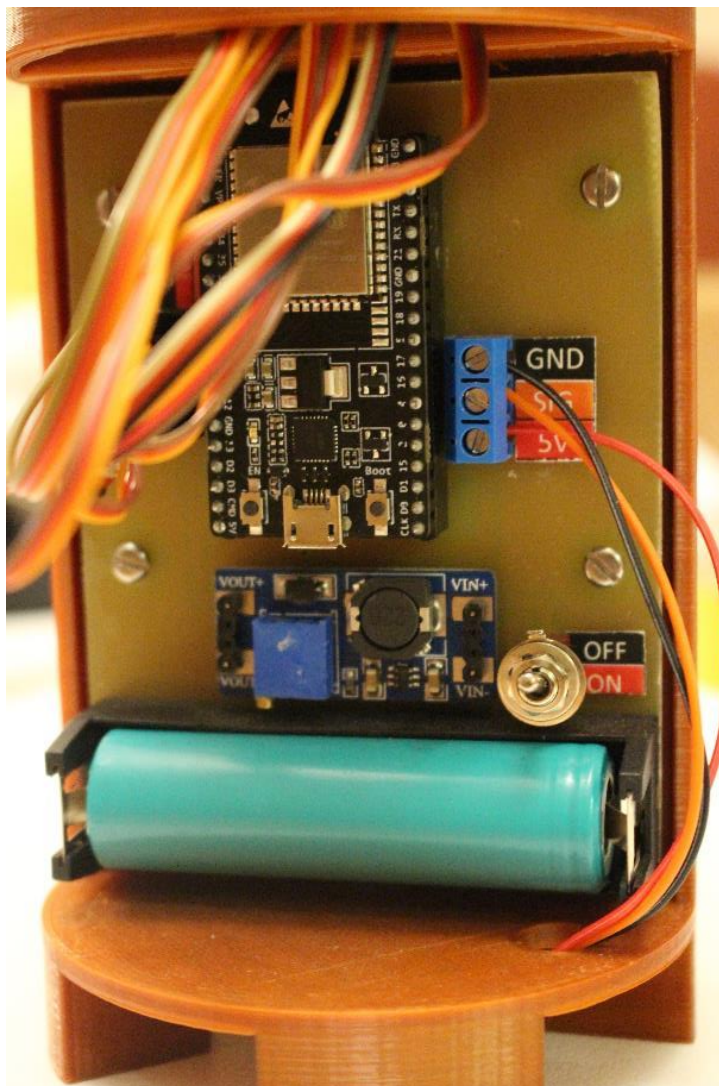


Obrázek 64: RGB LED kroužek



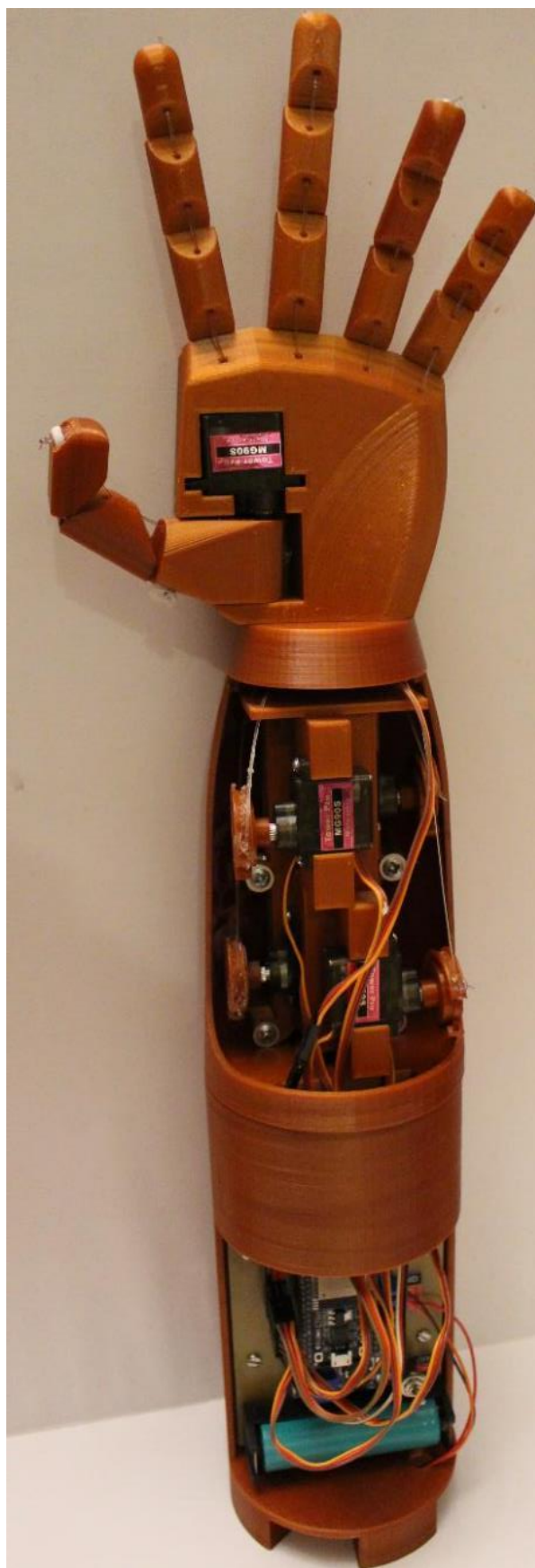
Obrázek 65: RGB LED kroužek přilepený zespod robotické ruky

Nakonec dráty RGB LED kroužku zapojte do svorkovnice na PCB:



Obrázek 66: Pohled na DPS ve sloupku robotické ruky

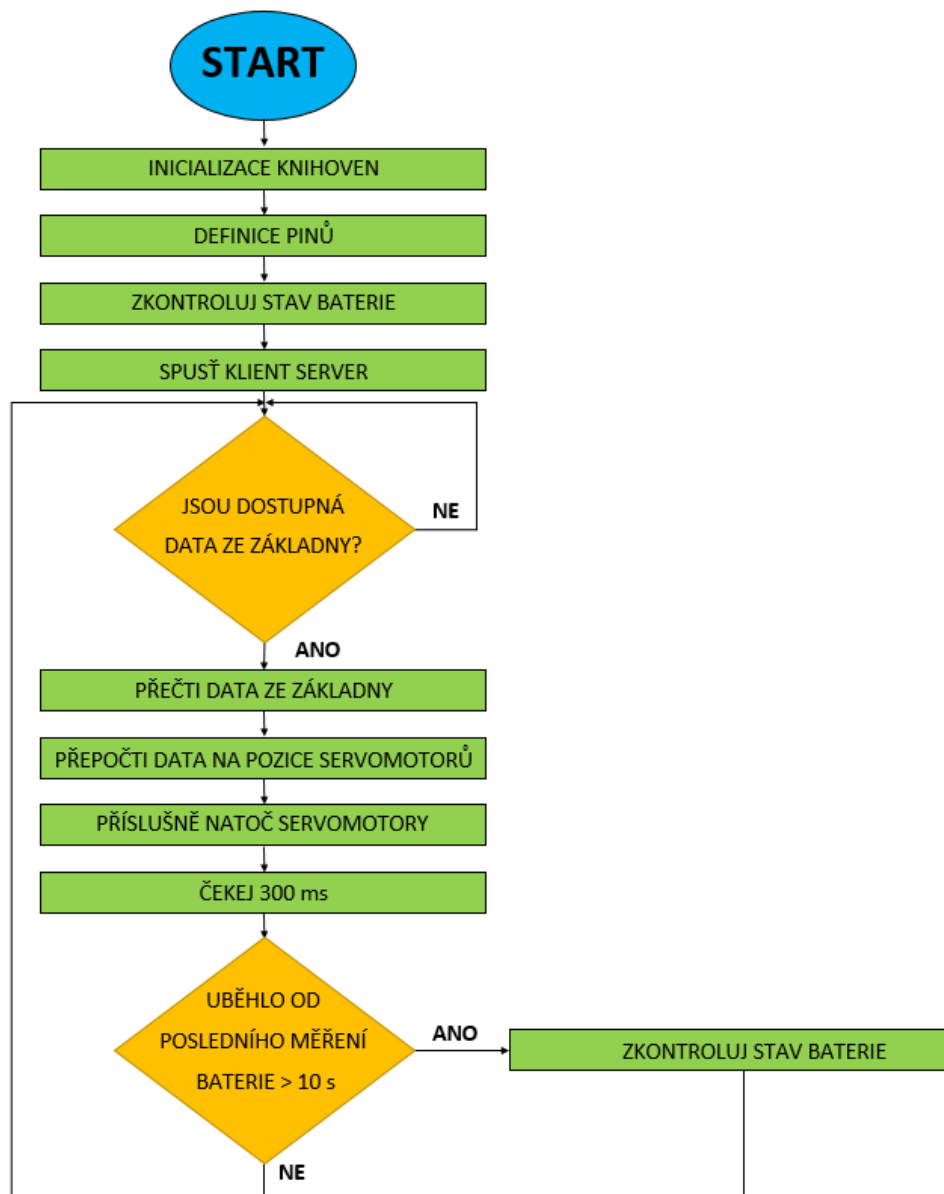
A hardware máte hotov!



Obrázek 67: Sestavená robotická ruka

2.3 Software pro komunikaci s bezdrátovými moduly

Software pro ESP32 sloužící ke komunikaci s bezdrátovými řídicími moduly je psán ve vývojovém prostředí PlatformIO^[13] s podporou Arduino IDE v jazyce C. Komunikace mezi ESP32 na robotické ruce a ESP32 v řídicím modulu (např. sensorické rukavici) vychází z příkladu publikovaného na webu Instructables od makera Gyalu1^[14], který pomocí komunikace typu access point – station propojil dvě ESP8266. Využila jsem tohoto příkladu pro ovládání robotické ruky pomocí rukavice s akcelerometry, v němž ESP32 na rukavici s akcelerometry je základna (station) a ESP32 na modelu robotické ruky je přístupový bod (access point – AP), který se k základně připojuje. AP čte data ze základny, která potom přepočítává na pozici servomotorů, podle které následně servomotory příslušně natočí. Program každých deset vteřin zkontroluje stav baterie, který je poté signalizován barvou RGB LED kroužku (zelená, zelenomodrá, modrá, bílá, růžová, fialová, červená). V programu byly dále použity standardní Arduino knihovny WiFi.h, AdafruitNeopixel.h a ESP32Servo.h. Průběh programu je shrnut na následujícím algoritmu:



Obrázek 68: Algoritmus programu robotické ruky

```

1  /*
2  |   Name: Robotic Hand      Author: MartinaH      Date: March 2019
3  |   This program is based on example of accespoint - station communication from Gyalul:
4  |   https://www.instructables.com/id/Accesspoint-Station-Communication-Between-Two-ESP8/
5  |   Others sources: internal Arduino libraries: AdafruitNeopixel.h, WiFi.h, ESP32Servo.h
6  | */
7
8  // Internal Arduino libraries
9  #include <Arduino.h>
10 #include <Adafruit_NeoPixel.h>
11 #include <WiFi.h>
12 #include <ESP32Servo.h>
13
14 // Making servo objects
15 Servo thumb;
16 Servo indexFinger;
17 Servo middleFinger;
18 Servo ringFinger;
19 Servo smallFinger;
20 Servo wrist;
21
22 #define VBAT 33
23
24 WiFiServer server(80);
25 Adafruit_NeoPixel ring = Adafruit_NeoPixel(16, 4, NEO_GRB + NEO_KHZ800);
26 IPAddress IP(192,168,4,15);
27 IPAddress mask (255, 255, 255, 0);
28
29 void CheckBattery();
30
31 void setup() {
32 |   thumb.attach(25);
33 |   indexFinger.attach(26);
34 |   middleFinger.attach(27);
35 |   ringFinger.attach(14);
36 |   smallFinger.attach(12);
37 |   wrist.attach(13);
38
39 |   pinMode(VBAT, INPUT);
40
41 |   Serial.begin(115200);
42 |   ring.begin();
43
44 |   CheckBattery();
45
46 |   // Starting client server
47 |   WiFi.mode(WIFI_AP);
48 |   WiFi.softAP("Wemos_AP", "Wemos_comm");
49 |   WiFi.softAPConfig(IP, IP, mask);
50 |   server.begin();
51
52 |   Serial.println();
53 |   Serial.println("Robotic Hand");
54 |   Serial.println("Server started.");
55 |   Serial.print("IP: ");      Serial.println(WiFi.softAPIP());
56 |   Serial.print("MAC:");      Serial.println(WiFi.softAPmacAddress());
57 | }
58
59 long Millis, currentTime = 0;
60

```

```

61 void loop() {
62   WiFiClient client = server.available();
63   if (!client) {return;}
64   // Reading data from the station
65   String request = client.readStringUntil('\r');
66   String request2 = client.readStringUntil('\r');
67   String request3 = client.readStringUntil('\r');
68   String request4 = client.readStringUntil('\r');
69   String request5 = client.readStringUntil('\r');
70   String request6 = client.readStringUntil('\r');
71   String request7 = client.readStringUntil('\r');
72   String request8 = client.readStringUntil('\r');
73
74   // Printing data on serial monitor
75   Serial.println("*****");
76   Serial.println("From the station: ");
77   Serial.println("1." + request);
78   Serial.println("2." + request2);
79   Serial.println("3." + request3);
80   Serial.println("4." + request4);
81   Serial.println("5." + request5);
82   Serial.println("6." + request6);
83
84   client.flush();
85
86   // Converting the accelerometer and gyroscope positions to the servo positions
87   int pos1 = map(request.toInt(), -900, 900, 0, 180);
88   int pos2 = map(request2.toInt(), -900, 900, 0, 180);
89   int pos3 = map(request3.toInt(), -900, 900, 0, 180);
90   int pos4 = map(request4.toInt(), -900, 900, 0, 180);
91   int pos5 = map(request5.toInt(), -900, 900, 0, 180);
92   int pos6 = map(request6.toInt(), -200, 2800, 0, 180);

```

```

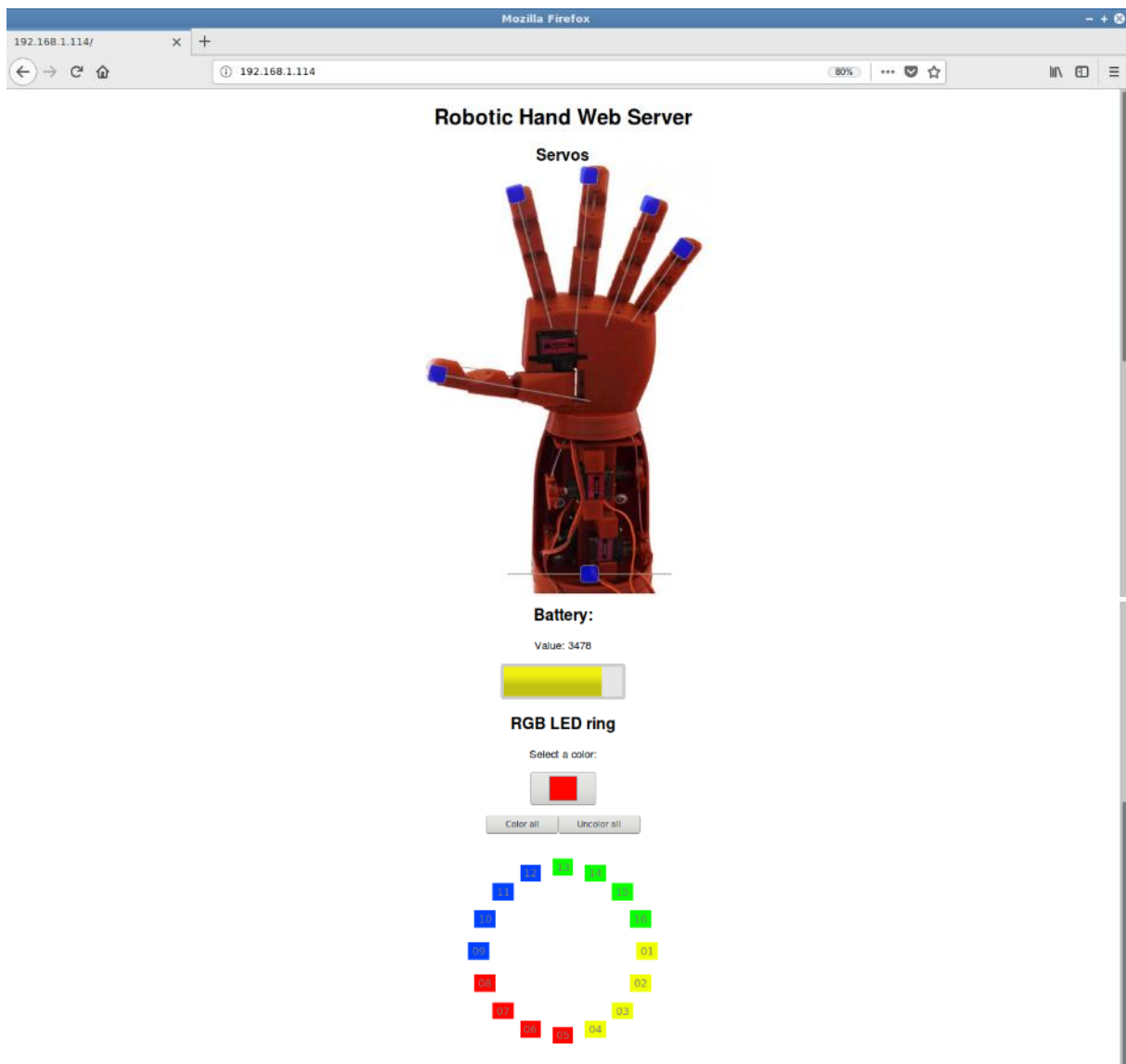
89   int pos3 = map(request3.toInt(), -900, 900, 0, 180);
90   int pos4 = map(request4.toInt(), -900, 900, 0, 180);
91   int pos5 = map(request5.toInt(), -900, 900, 0, 180);
92   int pos6 = map(request6.toInt(), -200, 2800, 0, 180);
93
94   // Sending positions to servos
95   thumb.write(pos1);
96   indexFinger.write(pos2);
97   middleFinger.write(pos3);
98   ringFinger.write(pos4);
99   smallFinger.write(pos5);
100  wrist.write(pos6);
101
102  delay(300);
103
104  // Checking the battery state every minute
105  Millis = millis();
106  if((Millis - currentTime) > 60000)
107  {
108      currentTime = Millis;
109      CheckBattery();
110  }
111
112 }
113
114 void CheckBattery()
115 {
116     int red = 0, green = 0, blue = 0;
117     int battery = 0;
118
119     for(int i = 0; i < 20; i++)
120     {
121         battery += analogRead(33);
122     }
123     battery /= 20; // Average of 20 values
124     battery *= 4.07; // Conversion ADC to mV
125
126     // The RGB LED ring color corresponds to the battery voltage
127     if(battery < 3000) {red = 255; green = 0; blue = 0;}
128     else if(battery < 3200) {red = 200; green = 0; blue = 255;}
129     else if(battery < 3400) {red = 255; green = 0; blue = 150;}
130     else if(battery < 3600) {red = 255; green = 255; blue = 255;}
131     else if(battery < 3800) {red = 0; green = 0; blue = 255;}
132     else if(battery < 4000) {red = 0; green = 255; blue = 255;}
133     else {red = 0; green = 255; blue = 0;}
134
135     for(int i=0;i<=16;i++)
136     {
137         ring.setPixelColor(i, ring.Color(red, green, blue));
138         ring.show();
139     }
140     Serial.printf("Battery: %i mV\n", battery);
141 }

```

Obrázek 69: Výpis programu pro robotickou ruku pro komunikaci s bezdrátovými moduly

2.4 Software pro webserver

Software pro ESP32 vytvářející webserver pro ovládání robotické ruky z počítače, mobilu či tabletu je psán ve vývojovém prostředí PlatformIO s podporou Arduino IDE v jazyce C. Webová stránka je psána v jazycích HTML, CSS a JavaScript a následně konvertována do .cpp souboru. Pro získávání dat z webu a posílání dat na web byl použit příklad z PathArgServer^[15] z githubu espressif. V programu byly dále převzaty Arduino knihovny WiFi.h, WiFiClient.h, WebServer.h, ESPmDNS.h, AdafruitNeopixel.h a ESP32Servo.h. Webserver pomocí šesti posuvníků umožňuje ovládání servomotorů na modelu robotické ruky, dále ukazuje stav baterie a obsluhuje RGB LED kroužek. Barva každé LED v kroužku je nastavitelná individuálně pomocí RGB vzorkovníku barev. Níže je uveden výpis hlavního zdrojového kódu main.cpp. Soubor s webovou stránkou je příliš velký pro publikování a je uvedený v příloze této práce a na mém gitlabu.



Obrázek 70: Webserver robotické ruky

```

1  /*
2  Name: Robotic Hand Webserver   Author: MartinaH   Date: March 2019
3  This program is based on example PathArgServer from espressif github:
4  https://github.com/espressif/arduino-esp32/blob/master/libraries/WebServer/examples/PathArgServer/PathArgServer.ino
5  Others sources: Arduino libraries: WiFi.h, WiFiClient.h, WebServer.h, ESPmDNS.h, AdafruitNeoPixel.h, ESP32Servo.h
6  */
7
8  // Internal Arduino libraries
9  #include <WiFi.h>
10 #include <WiFiClient.h>
11 #include <WebServer.h>
12 #include <ESPmDNS.h>
13 #include <Adafruit_NeoPixel.h>
14 #include <ESP32Servo.h>
15
16 // RGB LED ring initiation
17 Adafruit_NeoPixel ring = Adafruit_NeoPixel(16, 4, NEO_GRB + NEO_KHZ800);
18
19 // Replace with your SSID and password
20 const char *ssid = "-----";
21 const char *password = "-----";
22
23 extern const char *web;
24 extern const char *ruka;
25
26 // List of functions
27 void handleNotFound();
28 void WebRoot();
29 void ServoValue();
30 void RingValue();
31
32 WebServer server(80);
33
34 // Making servo objects
35 Servo thumb;
36 Servo indexFinger;
37 Servo middleFinger;
38 Servo ringFinger;
39 Servo smallFinger;
40 Servo wrist;
41
42 String battery;
43
44 void WebRoot()
45 {
46   server.send(200, "text/html", web);
47 }
48
49 // Getting values from slider bars on website
50 void ServoValue()
51 {
52   Serial.printf("Servo: %s val: %s\n", server.pathArg(0).c_str(), server.pathArg(1).c_str());
53
54   String request1 = server.pathArg(0).c_str();
55   String request2 = server.pathArg(1).c_str();
56
57   int servo = request1.toInt(); // Slider bar number
58   int pos = request2.toInt(); // Slider bar position
59
60   // Sending positions to servos
61   switch(servo)
62   {
63     case 1: thumb.write(pos); delay(20);
64             break;
65     case 2: indexFinger.write(pos); delay(20);
66             break;
67     case 3: middleFinger.write(pos); delay(20);
68             break;
69     case 4: ringFinger.write(pos); delay(20);
70             break;
71     case 5: smallFinger.write(pos); delay(20);
72             break;
73     case 6: wrist.write(pos); delay(20);
74             break;
75   }
76   server.send(200, "text/html", "Hello here");
77 }
78

```

```

79 // Getting LED number and RGB code from website
80 void RingValue()
81 {
82     Serial.printf("LED: %s red: %s green: %s blue: %s\n", server.pathArg(0).c_str(), server.pathArg(1).c_str(),
83     server.pathArg(2).c_str(), server.pathArg(3).c_str());
84
85     String request1 = server.pathArg(0).c_str();
86     String request2 = server.pathArg(1).c_str();
87     String request3 = server.pathArg(2).c_str();
88     String request4 = server.pathArg(3).c_str();
89
90     int ledNumber = request1.toInt();
91     int red = request2.toInt();
92     int green = request3.toInt();
93     int blue = request4.toInt();
94
95     // Sending values to RGB LED ring
96     ring.setPixelColor(ledNumber-1, ring.Color(red, green, blue));
97     ring.show();
98
99     server.send(200, "text/html", "Hello here");
100 }
101
102 void setup(void)
103 {
104     Serial.begin(115200);
105
106     thumb.attach(25);
107     indexFinger.attach(26);
108     middleFinger.attach(27);
109     ringFinger.attach(14);
110     smallFinger.attach(12);
111     wrist.attach(13);
112
113     WiFi.mode(WIFI_STA);
114     WiFi.begin(ssid, password);
115     Serial.println("");
116
117     // Uncoloring RGB LED ring
118     ring.begin();
119     for(int i = 0; i <=15; i++)
120     {
121         ring.setPixelColor(i, ring.Color(0, 0, 0));
122         ring.show();
123     }
124
125     // Waiting for connection
126     while (WiFi.status() != WL_CONNECTED) {
127         delay(500);
128         Serial.print(".");
129     }
130
131     Serial.println("");
132     Serial.print("Connected to ");
133     Serial.println(ssid);
134     Serial.print("IP address: ");
135     Serial.println(WiFi.localIP());
136
137     if (MDNS.begin("ruka")) {
138         Serial.println("MDNS responder started");
139     }
140
141     // Getting values from website
142     server.on("/", WebRoot);
143     server.on("/servo/{}/{}", ServoValue);
144     server.on("/ring/{}/{}/{}", RingValue);
145
146     // Sending data on website (battery state and image of robotic hand)
147     server.on("/battery", [](){ server.send(200, "text/plain", String(map(analogRead(33), 0, 1024, 0, 4200))); });
148     server.on("/ruka.jpg", [](){ server.send_P(200, "img/jpg", ruka, 43078); });
149
150     server.onNotFound(handleNotFound);
151
152     server.begin();
153     Serial.println("HTTP server started");
154 }

```

```

155
156 void loop(void) {
157     server.handleClient();
158 }
159
160 void handleNotFound()
161 {
162     String message = "File Not Found\n\n";
163     message += "URI: ";
164     message += server.uri();
165     message += "\nMethod: ";
166     message += (server.method() == HTTP_GET) ? "GET" : "POST";
167     message += "\nArguments: ";
168     message += server.args();
169     message += "\n";
170     for (uint8_t i = 0; i < server.args(); i++) {
171         message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
172     }
173     server.send(404, "text/plain", message);
174 }

```

Obrázek 71: Výpis programu pro webserver – main.cpp

2.5 Software pro mobilní Bluetooth aplikaci

Software pro ESP32 k přímému řízení robotické ruky z notebooku, mobilního telefonu nebo tabletu pomocí Bluetooth je psán ve vývojovém prostředí PlatformIO s podporou Arduino IDE v jazyce C. Mobilní aplikace pro operační systém Android je napsána v programu Pocket Code^[16], jehož tutoriály v češtině můžete zhlédnout na mém Youtube kanálu^[17]. Pro implementaci Bluetooth periferie do programu pro ESP32 jsem se inspirovala z příkladu z webu CircuitDigest^[18]. V programu byly dále převzaty Arduino knihovny BluetoothSerial.h a ESP32Servo.h. Mobilní aplikace pomocí šesti neviditelných posuvníků umožňuje ovládání servomotorů na modelu robotické ruky. Níže vidíte výpis zdrojového kódu. Mobilní aplikaci lze stáhnout z Pocket Code Sharing platformy zde: <https://share.catrob.at/pocketcode/program/82476>



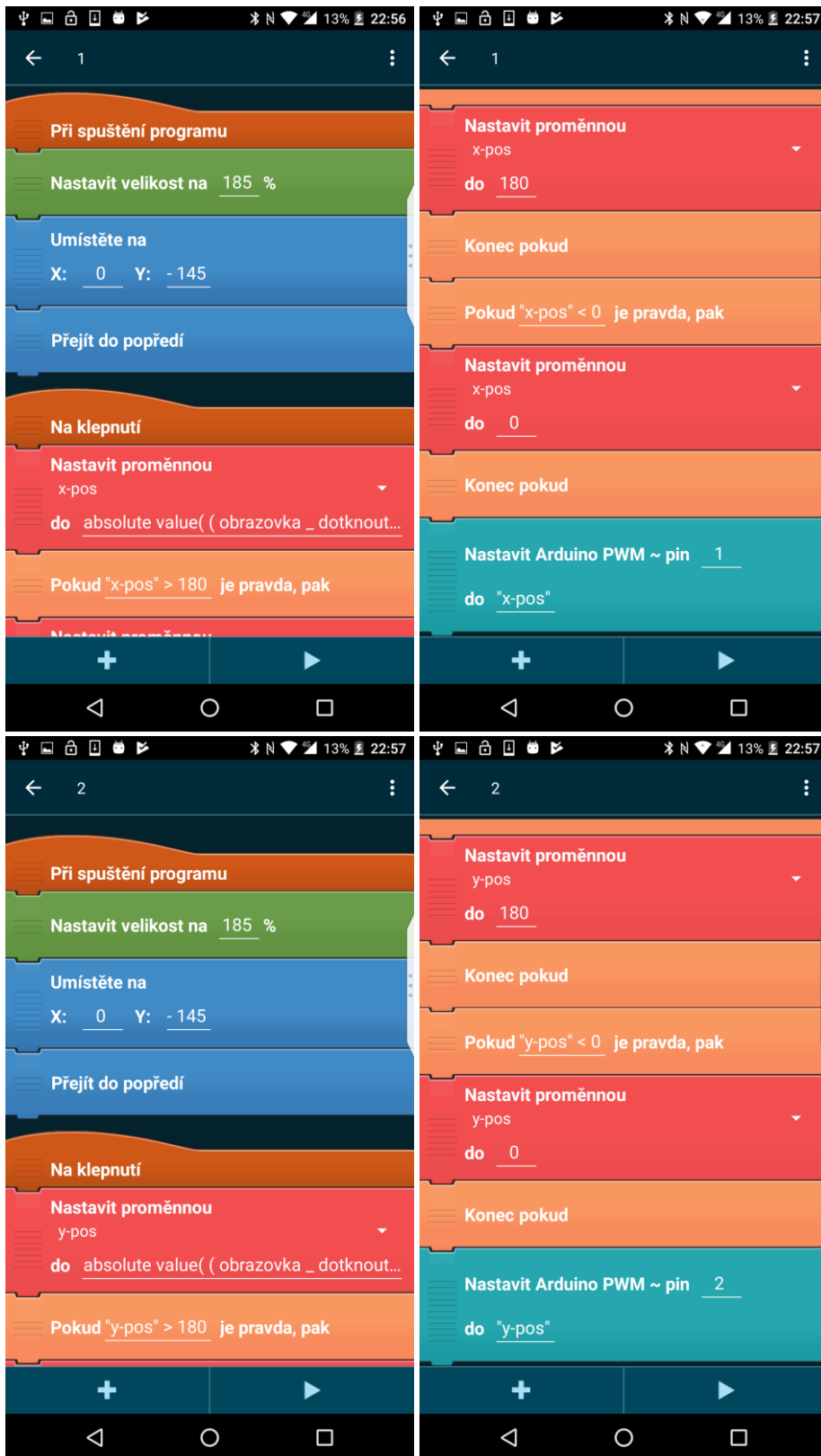
Obrázek 72: Fotografie obrazovky mobilní aplikace

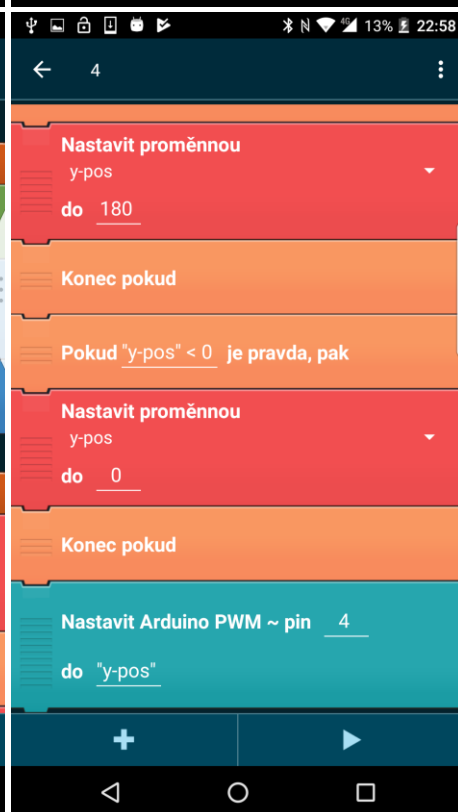
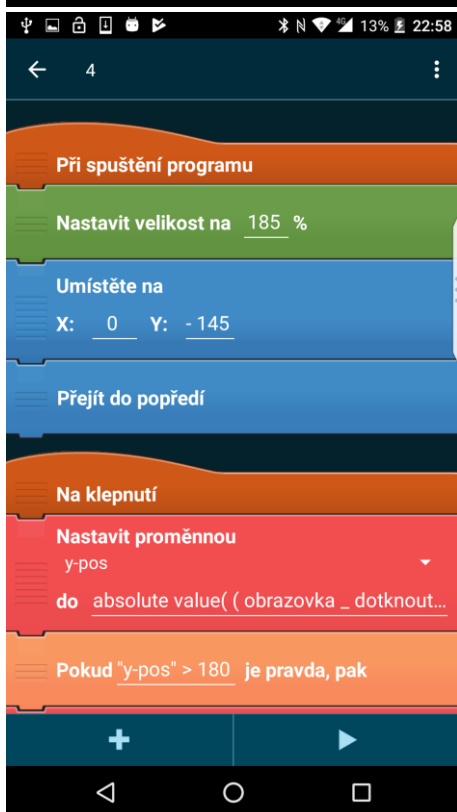
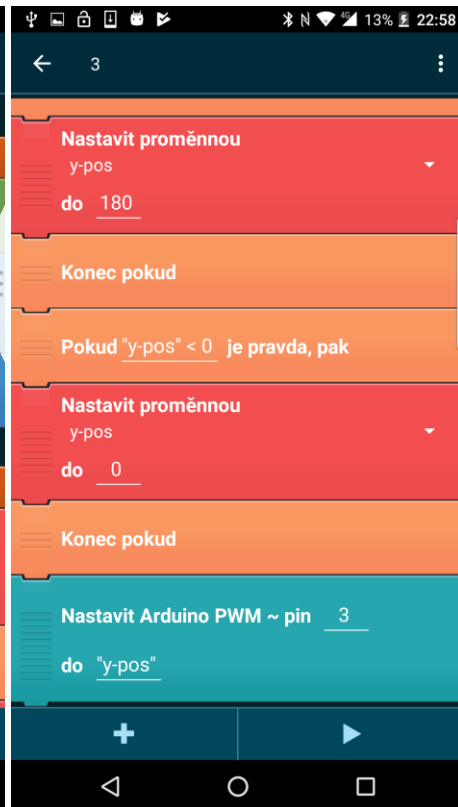
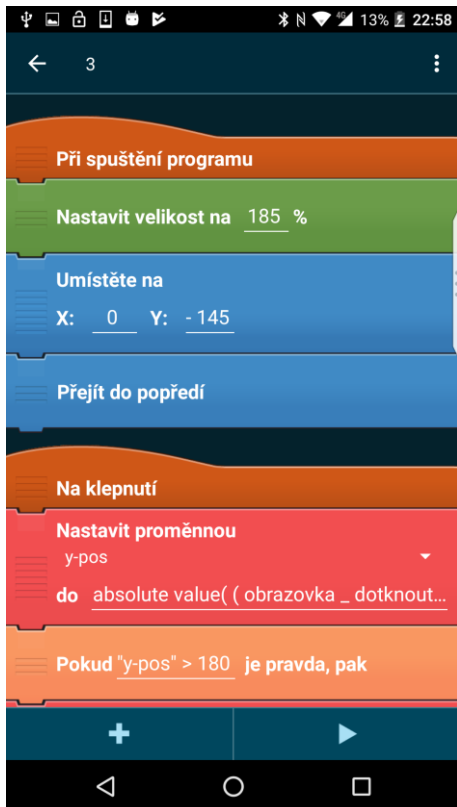
2.5.1 Výpis programu pro Bluetooth komunikace s Pocket Code v ESP32

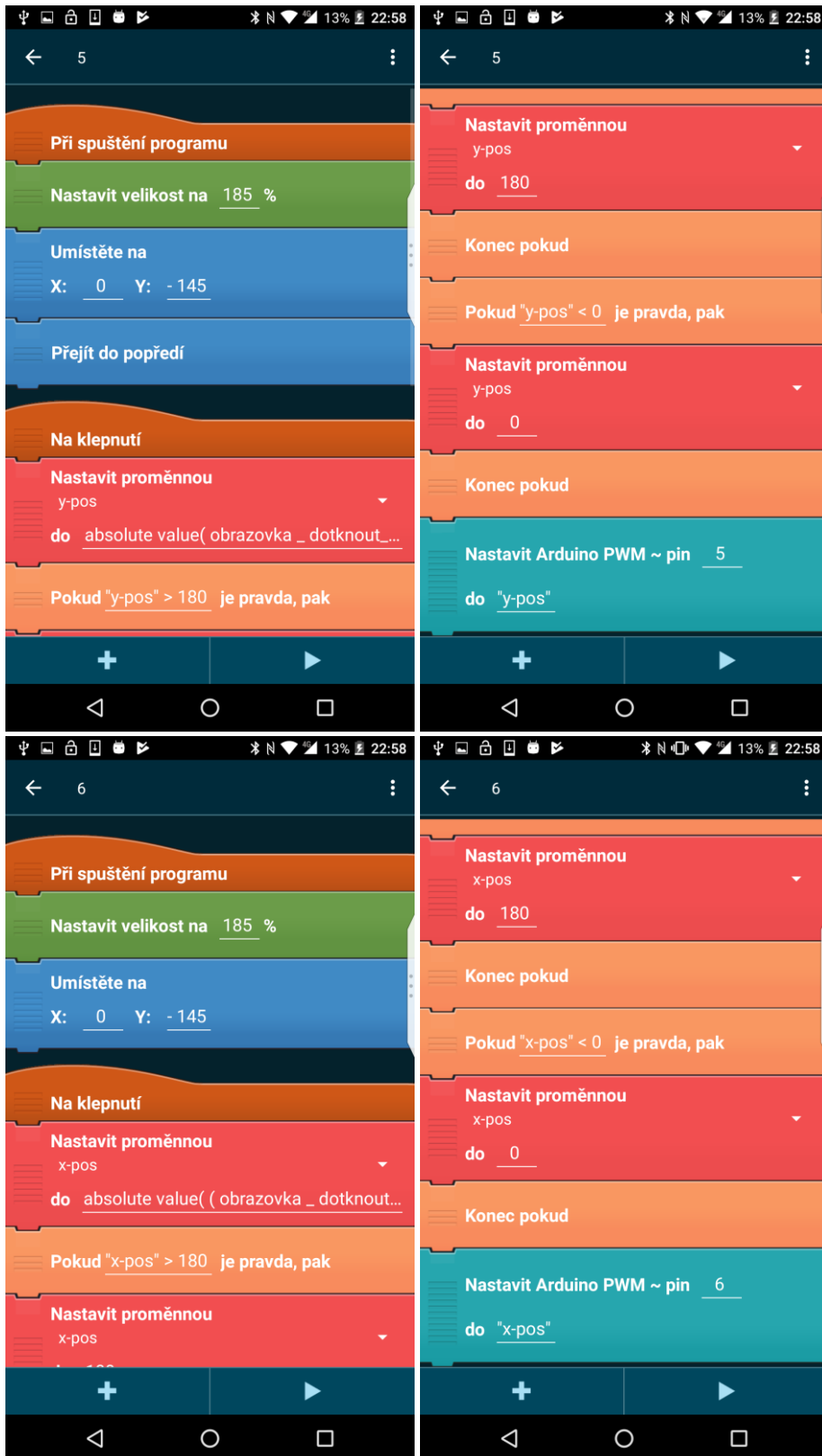
```
1  /*
2   Name: Robotic Hand BT   Author: MartinaH   Date: March 2019
3   This program is based on example of accespoint - station communication from Gyalul:
4   https://circuitdigest.com/microcontroller-projects/using-classic-bluetooth-in-esp32-and-toogle-an-led
5   Others sources: internal Arduino libraries: BluetoothSerial.h, ESP32Servo.h
6  */
7
8  // Internal Arduino libraries
9  #include <BluetoothSerial.h>
10 #include <ESP32Servo.h>
11
12 // Making servo objects
13 Servo thumb;
14 Servo indexFinger;
15 Servo middleFinger;
16 Servo ringFinger;
17 Servo smallFinger;
18 Servo wrist;
19
20 // Making Bluetooth object
21 BluetoothSerial ESP_BT;
22
23 int incoming[6];
24
25 void setup() {
26   Serial.begin(115200);
27   ESP_BT.begin("RoboRuka");
28   Serial.println("Bluetooth Device is Ready to Pair");
29
30   thumb.attach(25);
31   indexFinger.attach(26);
32   middleFinger.attach(27);
33   ringFinger.attach(14);
34   smallFinger.attach(12);
35   wrist.attach(13);
36 }
37
38 void loop() {
39
40   if (ESP_BT.available()) //Check if we receive anything from Bluetooth
41   {
42     // Reading 6 bytes
43     for(int i = 0; i < 6; i++)
44     {
45       incoming[i] = ESP_BT.read();
46       Serial.println(incoming[i]);
47     }
48
49     if (incoming[0] == 244 && incoming[1] == 1) {thumb.write(incoming[4]);}
50     if (incoming[0] == 244 && incoming[1] == 2) {ringFinger.write(incoming[4]);}
51     if (incoming[0] == 244 && incoming[1] == 3) {middleFinger.write(incoming[4]);}
52     if (incoming[0] == 244 && incoming[1] == 4) {ringFinger.write(incoming[4]);}
53     if (incoming[0] == 244 && incoming[1] == 5) {smallFinger.write(incoming[4]);}
54     if (incoming[0] == 244 && incoming[1] == 6) {wrist.write(incoming[4]);}
55   }
56   delay(100);
57 }
```

Obrázek 73: Výpis programu k ovládní přes Bluetooth pro ESP32

2.5.2 Výpis programu pro mobilní aplikaci v Pocket Code







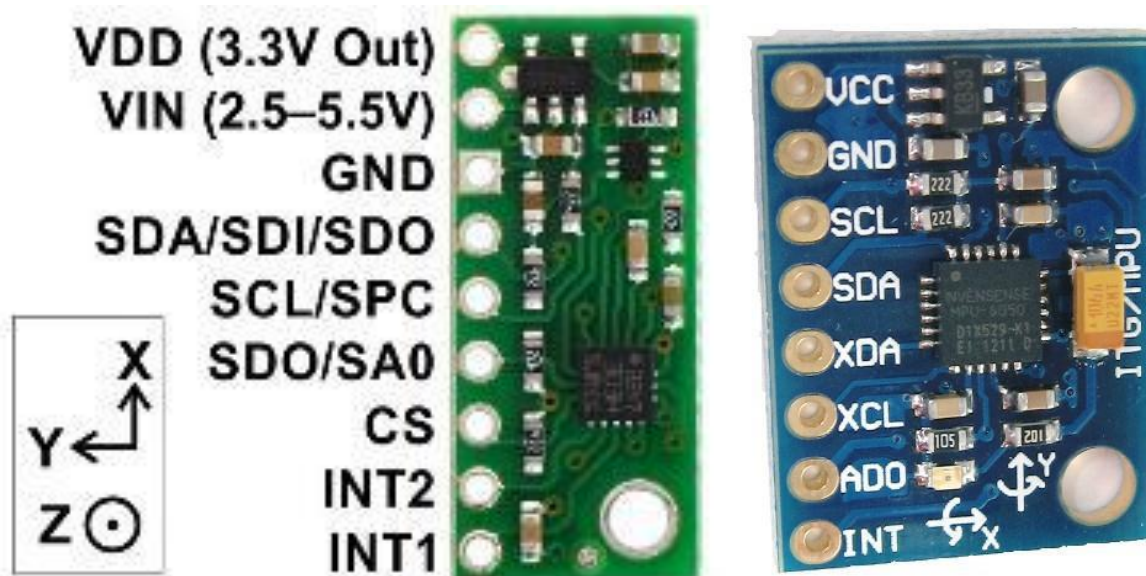
Obrázek 74: Výpis programu pro mobilní aplikaci v Pocket Code

3 Rukavice s akcelerometry

Jako jednoduchý příklad možnosti ovládní robotické ruky pomocí bezdrátových modulů jsem vyrobila rukavici s pěti moduly akcelerometrů na konečcích prstů, které snímají pohyb jednotlivých prstů. Moduly akcelerometrů jsou spojeny plochým FFC kabelem s deskou plošného spoje, obsahující napájený modul ESP32 Wrover a modul gyroskopu pro detekci otáčení zápěstím. K napájení desky slouží malý Li-Pol akumulátor (3,6 V / 250 mAh) v kombinaci s diodou 1N4007 pro ochranu proti přepólování a pro sražení příliš vysokého napětí plně nabitého akumulátoru (může dosahovat až 4,2 V). Napětí na akumulátoru je měřeno analogovým vstupem ESP32 přes odporový dělič a po zapnutí zařízení indikováno počtem bliknutí červené LED. Zapnutí napájení je signalizováno zelenou LED. Programování modulu ESP32 Wrover se provádí externím převodníkem USB/UART a programovací tlačítka (boot, reset).

Jako akcelerometry na konečcích prstů je možné použít jakékoliv moduly akcelerometrů schopné komunikovat s mikrokontrolérem po SPI sběrnici. Ke stavbě svého modelu jsem použila moduly Pololu – 2127^[19] obsahující čip LSM303D, schopný měřit akceleraci ve všech třech osách v rozsazích ± 2 g, ± 4 g, ± 8 g a ± 16 g (pro reálné měření se používá rozsah ± 2 g), použít jde nicméně i levnější typy modulů akcelerometrů, pokud se příslušně upraví připojovací deska, která převádí signál z pinových řad akcelerometru na FPC konektor.

Pro detekci rotace zápěstí je použit modul 3osého gyroskopu (kombinovaný s akcelerometrem) MPU6050, který komunikuje s mikrokontrolérem ESP32 po I2C sběrnici a je připájen přímo k DPS s ESP32. MPU6050 je schopen detekovat rychlosti otáčení v rozsazích ± 250 , ± 500 , ± 1000 , ± 2000 °/sec, s rozlišením 16 bitů. Podobně jako v případě modulu akcelerometrů Pololu – 2127 je i zde možno použít jiného modulu s 3osým gyroskopem komunikujícím po I2C, může však být nutné změnit zapojení DPS podle rozložení pinů použitého modulu.

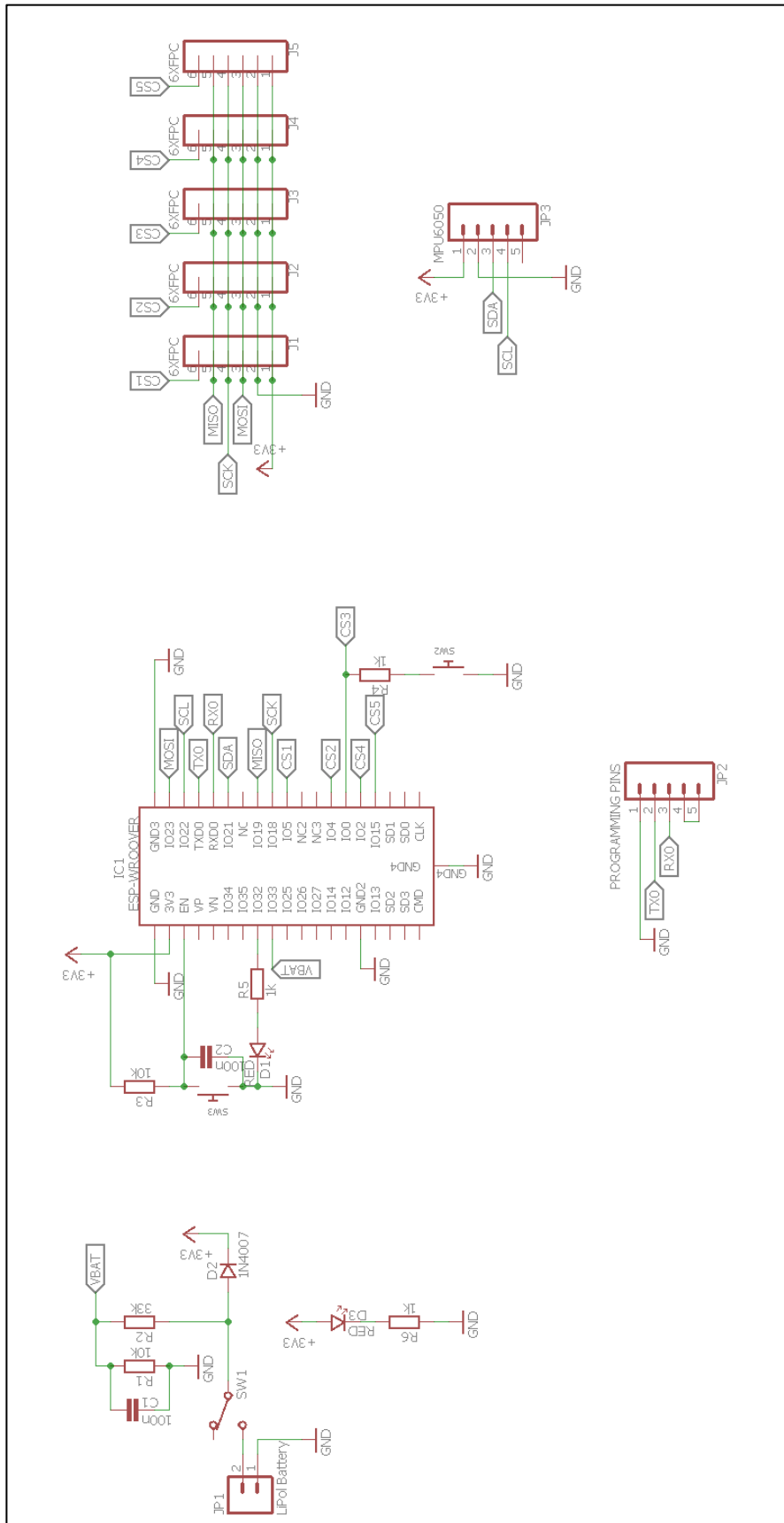


Obrázek 75: Akcelerometr Pololu – 2127^[20]

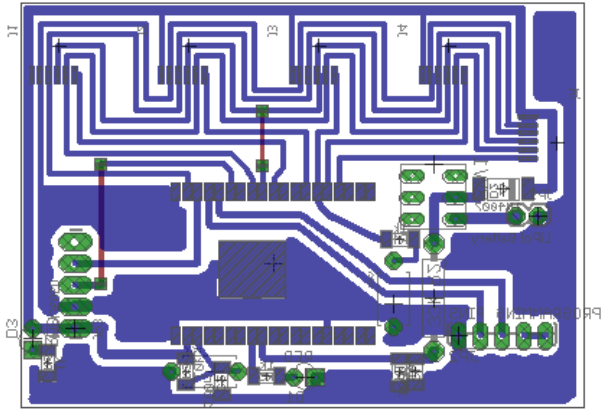
Obrázek 76: Modul gyroskopu s MPU6050^[21]

Pro ovládní robotické ruky pomocí rukavice s akcelerometry bylo zhotoveno 7 desek plošných spojů: 1 hlavní deska, 5 převodních desek s akcelerometry a 1 deska pro nabíječku Li-Pol baterie. Ke spojení mezi hlavní deskou a převodními deskami s akcelerometry slouží ohebné 6vodičové FFC pásky s roztečí 1 mm a délkou 15 cm, zapojované do FPC konektorů na deskách.

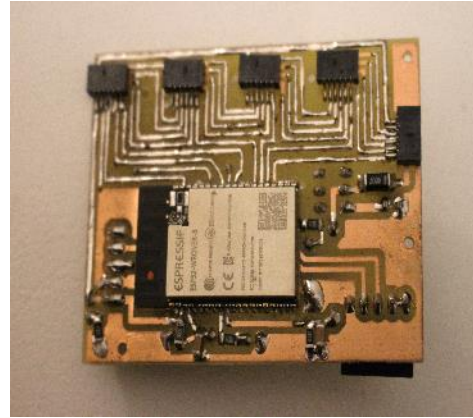
3.1 Hlavní deska pro rukavici



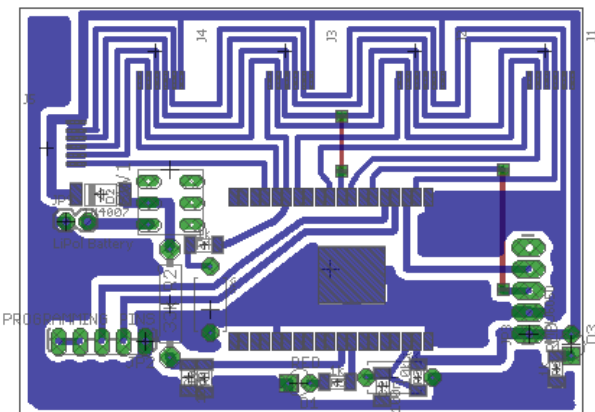
Obrázek 77: Schéma zapojení hlavní desky rukavice



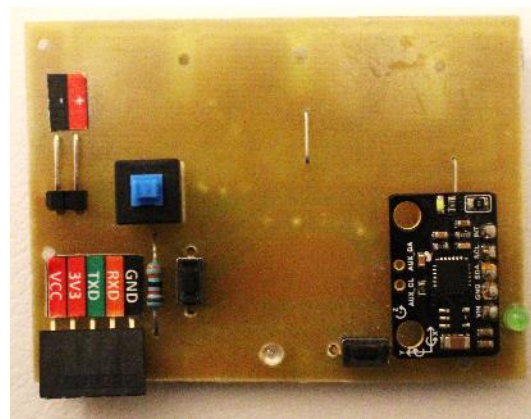
Obrázek 78: Pájecí schéma DPS hlavní desky rukavice – strana spojů



Obrázek 79: Foto DPS hlavní desky rukavice – strana spojů



Obrázek 80: Pájecí schéma DPS hlavní desky rukavice – strana součástek



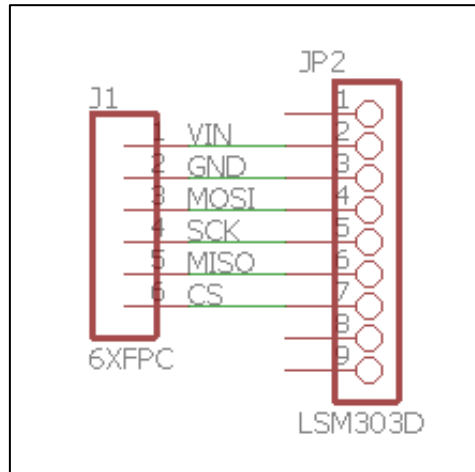
Obrázek 81: Foto DPS hlavní desky rukavice – strana součástek

Tabulka 4: Tabulka součástek hlavní DPS rukavice

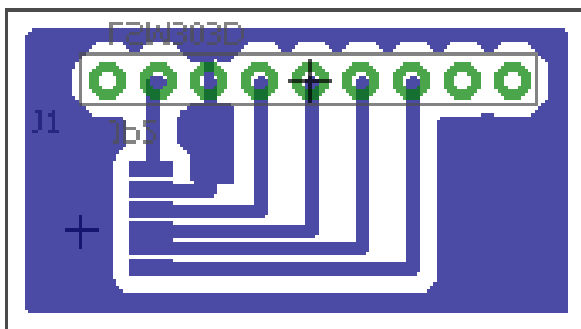
Kód	Typ	Parametry	Funkce	Cena [Kč]	Zdroj
C1	kondenzátor	100 nF SMD 1206	RC filtr	1	TME ^[9]
C2	kondenzátor	100 nF SMD 1206	utlumení proti zákmitům tlačítka	1	TME ^[9]
D1	LED dioda	červená 3 mm	indikace baterie	3	TME ^[9]
D2	dioda	1N4007 (alternativně S1A-13-F), SMA	ochrana proti přepólování a přepětí	2,6	TME ^[9]
D3	LED dioda	zelená 3 mm	indikace napájení	3	TME ^[9]
IC1	mikrokontrolér	ESP32 Wrover	řídící modul	110	TME ^[9]
IC2	3D akcelerometr a gyroskop	MPU6050	snímání rotace zápěstí	69	Laskarduino ^[10]
J1 - J5	FPC konektory	CONNFLY DS1020-06-06BT1	připojení akcelerometrů	20	TME ^[9]
JP1	zahnuté piny	2x1	připojení baterie	3	TME ^[9]
JP2	dutinky	5x1	připojení programovacího převodníku USB/UART	3	TME ^[9]
R1	rezistor	10k SMD 1206	odporový dělič	0,16	TME ^[9]
R2	rezistor	33k THT	odporový dělič	0,48	TME ^[9]
R3	rezistor	10k SMD 1206	pull - up pro reset tlačítko	0,16	TME ^[9]
R4	rezistor	1k SMD 1206	ochrana bootovacího vstupu	0,3	TME ^[9]
R5	rezistor	1k SMD 1206	omezení proudu na LED	0,3	TME ^[9]
R6	rezistor	1k SMD 1206	omezení proudu na LED	0,3	TME ^[9]
SW1	vypínač	tlačítkový	spínač napájení	7	TME ^[9]
SW2	tlačítko	NINGI TACT-34N-F	boot tlačítko	1,23	TME ^[9]
SW3	tlačítko	NINGI TACT-34N-F	reset tlačítko	1,23	TME ^[9]
	Li-Pol baterie	LP601730	napájení obvodu	74	TME ^[9]
	3D Akcelerometr a gyroskop	MPU6050	detekce rotace zápěstí	89	TME ^[9]
Celkem:				390	

3.2 Převodní desky s akcelerometry a FPC konektorem

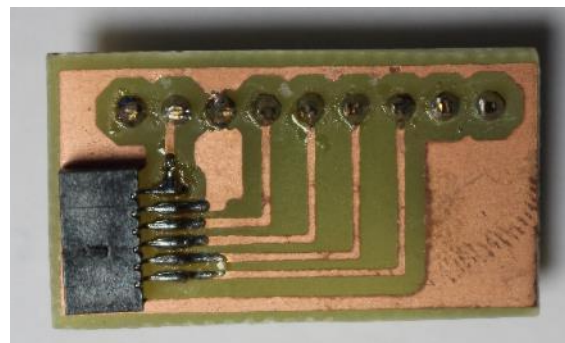
Převodní desky slouží k připojení modulů s akcelerometry k FPC konektorům pro ploché FFC kabely rozvádějící SPI komunikaci z hlavní desky rukavice.



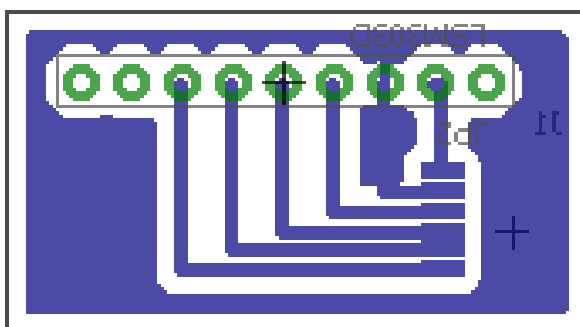
Obrázek 82: Schéma zapojení desky s akcelerometry



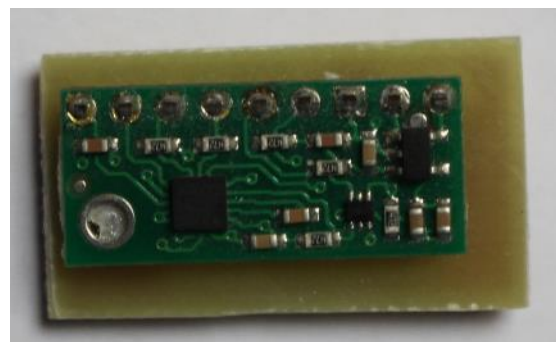
Obrázek 83: Pájecí schéma desky s akcelerometry
– strana spojů



Obrázek 84: Foto desky s akcelerometry
– strana spojů



Obrázek 85: Pájecí schéma desky s akcelerometry
– strana součástek



Obrázek 86: Foto desky s akcelerometry
– strana součástek

Tabulka 5: Tabulka součástek převodních desek s akcelerometry

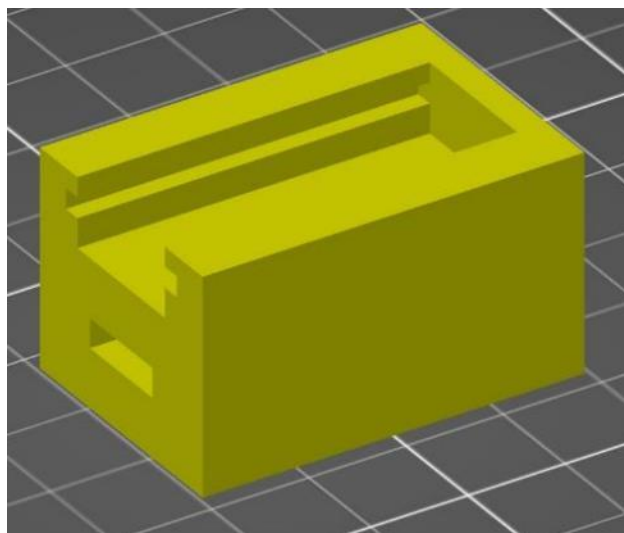
Kód	Typ	Parametry	Funkce	Cena [Kč]	Zdroj
5xJ1	FPC konektor	CONNFLY DS1020-06-06BT1	Připojení akcelerometrů na hlavní desku rukavice	20	TME ^[9]
5xIC1	akcelerometr	LSM303D	snímání polohy prstů	860	TME ^[9]
	5xFPC kabel	MOLEX 98267-0211	Připojení akcelerometrů na hlavní desku rukavice	140	TME ^[9]
Celkem:				1020	

3.3 Nabíječka Li-Pol akumulátorů

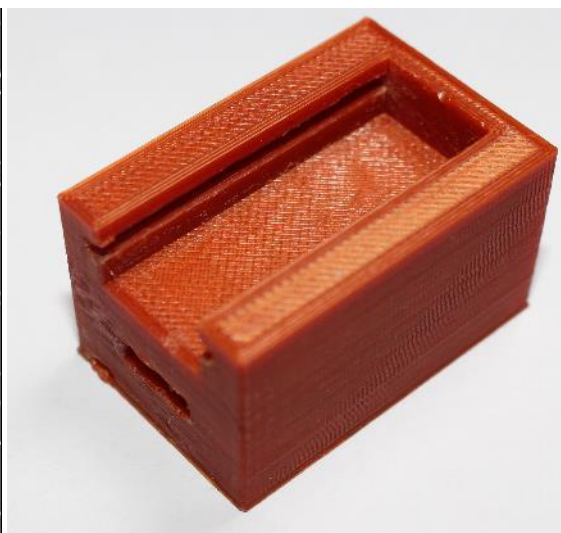
K nabíjení malých Li-Pol akumulátorů, jako jsou použity k napájení rukavice (250 mAh), nelze použít běžné nabíjecí moduly pro Li-ion akumulátory typu 18650, jako byly zmíněny v kapitole 2.2.1, protože většinou mají příliš vysoké nabíjecí proudy (0,5 – 1 A), které by malý Li-Pol akumulátor mohly poškodit. Proto jsem se rozhodla sestavit vlastní nabíječku Li-Pol akumulátorů s omezením nabíjecího proudu na 100 mA.

3.3.1 Mechanický hardware nabíječky

Pro úchyt plošného spoje a zasunutí nabíjeného akumulátoru jsem nakreslila v programu SketchUp držák a následně vytiskla na 3D tiskárně Prusa i3 z materiálu PLA.



Obrázek 87: Model držáku nabíječky



Obrázek 88: Foto držáku nabíječky

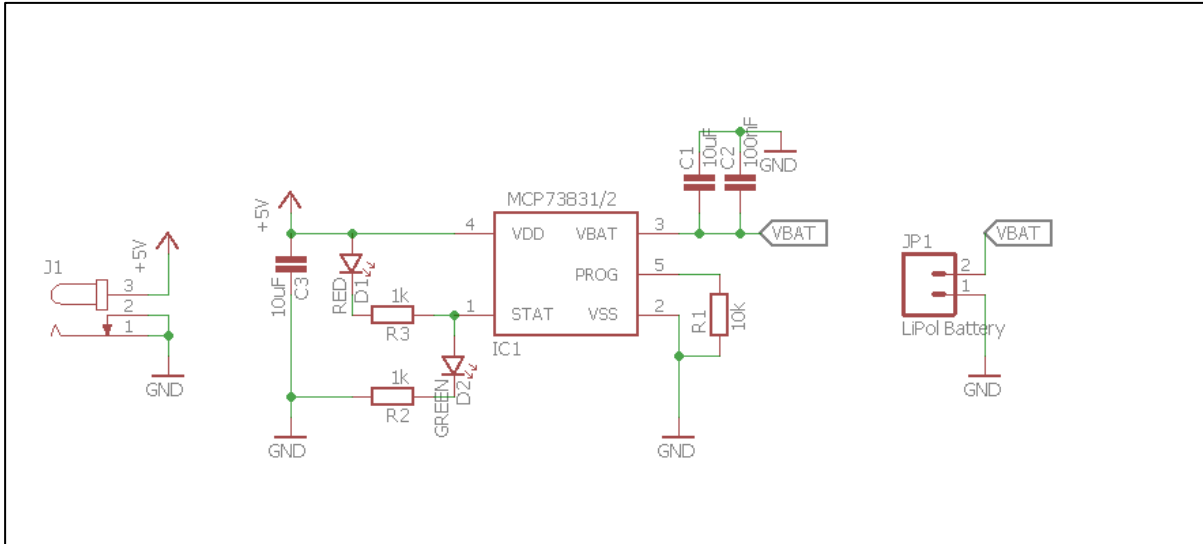
Tabulka 6: Tabulka tisku 3D modelu krabičky

Díl	Podpěry	Doba tisku	Hmotnost [g]	Cena struny [Kč]
držák pro nabíječku Li-Pol	ANO	1:13:48	13,14	7,21

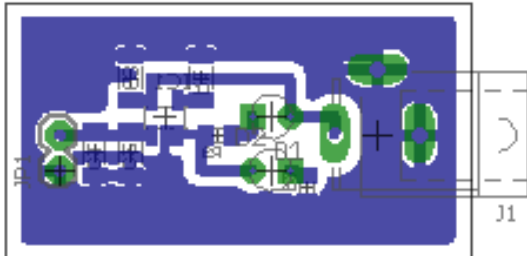
3.3.2 Elektronika nabíječky

Pro nabíjení Li-Pol baterie byla vyrobena jednovrstvá deska plošného spoje s nabíjecím čipem MCP73831, zapojená podle datasheetu^[23]

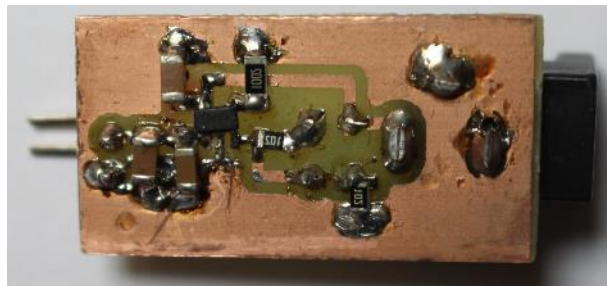
Rezistor R1 určuje nabíjecí proud, hodnota 10 kΩ odpovídá nabíjecímu proudu 100 mA, při její změně na 2 kΩ lze nabíjet maximálním proudem 500 mA (použitelné pro Li-ion akumulátor typu 18650 napájející robotickou ruku).



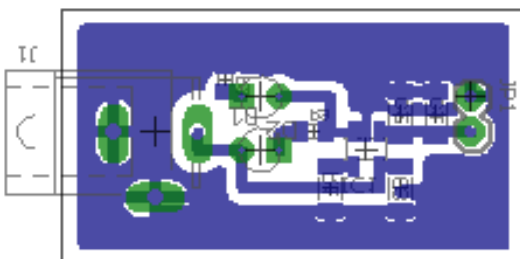
Obrázek 89: Schéma zapojení nabíječky Li-Pol baterií



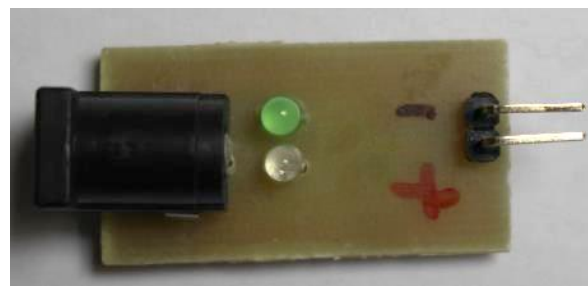
Obrázek 90: Pájecí schéma DPS nabíječky
– strana spojů



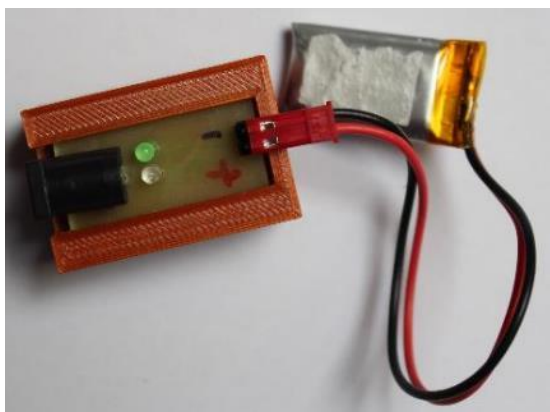
Obrázek 91: Foto DPS nabíječky
– strana spojů



Obrázek 92: Pájecí schéma DPS nabíječky
– strana součástek



Obrázek 93: Foto DPS nabíječky
– strana součástek



Obrázek 94: Foto nabíječky Li-Pol



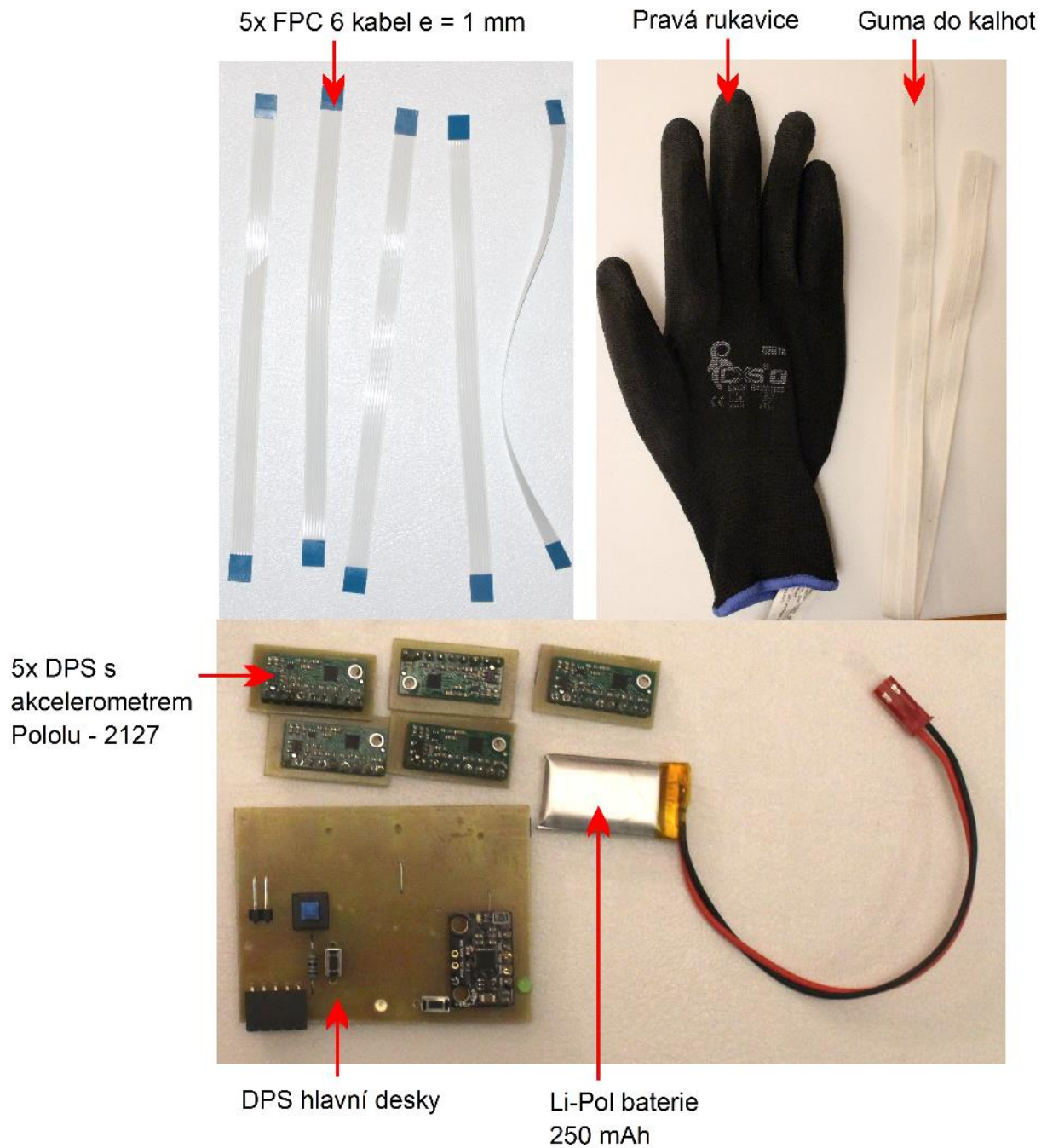
Obrázek 95: Foto nabíječky s vloženým akumulátorem

Tabulka 7: Tabulka součástí nabíječky Li-Pol baterií

Kód	Typ	Parametry	Funkce	Cena [Kč]	Zdroj
C1	kondenzátor	10 μ F SMD 1206 X7R	filtrace výstupu z nabíjecího čipu	2	TME ^[9]
C2	kondenzátor	100 nF SMD 1206 X7R	filtrace výstupů z nabíjecího čipu	1	TME ^[9]
C3	kondenzátor	10 μ F SMD 1206 X7R	filtrace napájení čipu	2	TME ^[9]
D1	LED dioda	červená 3 mm	indikace stavu nabíjení	3	TME ^[9]
D2	LED dioda	zelená 3 mm	indikace stavu nabíjení	3	TME ^[9]
IC1	integrováný obvod	MCP73831	nabíjení baterie	14	TME ^[9]
J1	jack konektor	NIGINI PC-GK2.5	připojení napájení 5V	3	TME ^[9]
JP1	zahnuté piny	2x1	připojení baterie	3	TME ^[9]
R1	rezistor	10 k Ω SMD 1206	nastavení nabíjecího proudu na 100 mA	0,3	TME ^[9]
R2	rezistor	1 k Ω SMD 1206	omezení proudu LED	0,3	TME ^[9]
R3	rezistor	1 k Ω SMD 1206	omezení proudu LED	0,3	TME ^[9]
Celkem:				32	

3.4 Návod na stavbu rukavice s akcelerometry

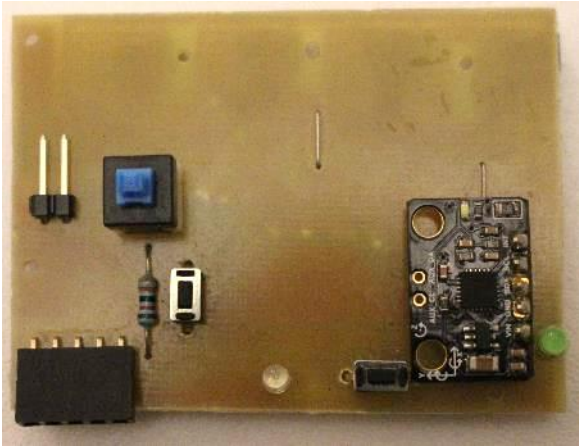
3.4.1 Potřebné komponenty



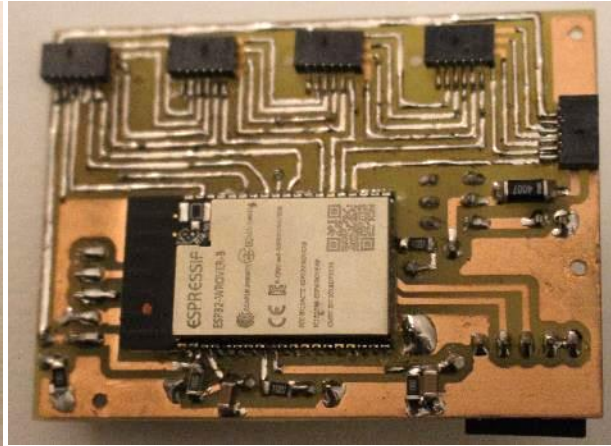
Obrázek 96: Potřebné komponenty ke stavbě rukavice

3.4.2 Krok 1: Provrtání DPS

V místech mimo signálové trasy provrtejte v DPS 4 – 5 otvorů průměru 2 mm pro zašití DPS do rukavice.



Obrázek 97: Provrtaný DPS – strana součástek



Obrázek 98: Provrtaný DPS – strana spojů

3.4.3 Krok 2: Přišití kapes pro akcelerometry

Na jednotlivé prsty rukavice přišijte gumu do kalhot, tak aby vytvořila kapsy jako na obrázku:



Obrázek 99: Pravá rukavice s gumou do kalhot



Obrázek 100: Rukavice s přišitými kapsami na akcelerometry

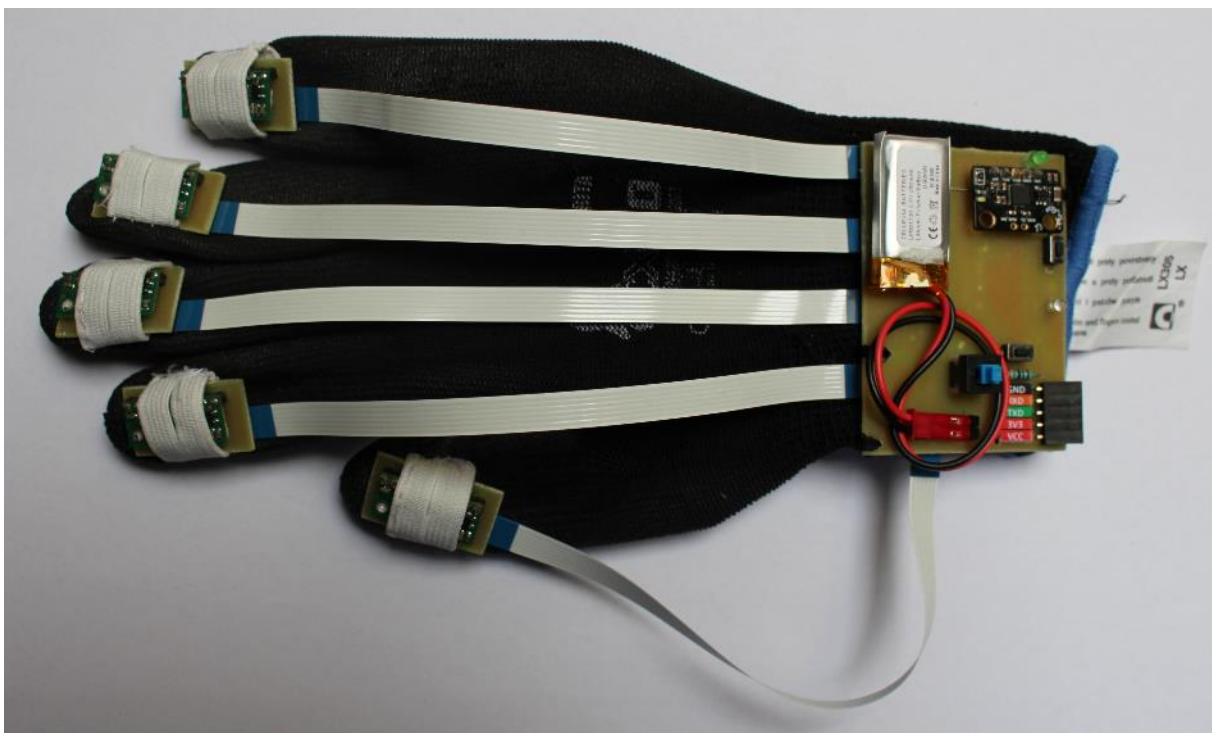
Poté do kapes vložte DPS s akcelerometry:



Obrázek 101: Rukavice s vloženými akcelerometry

3.4.4 Krok 3: Přišití hlavní desky k rukavici a propojení s akcelerometry

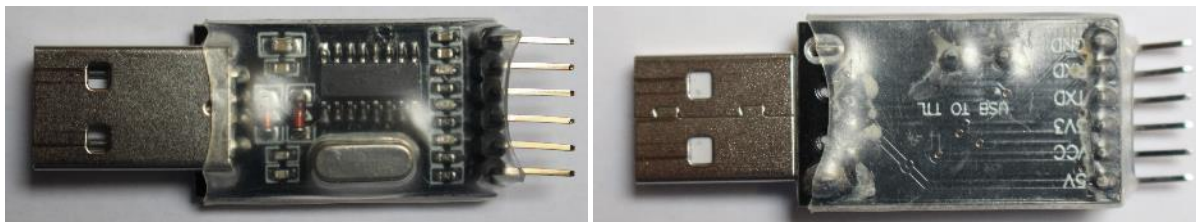
Přišijte hlavní desku k rukavici a propojte ji s DPS s akcelerometry pomocí FFC kabelů:



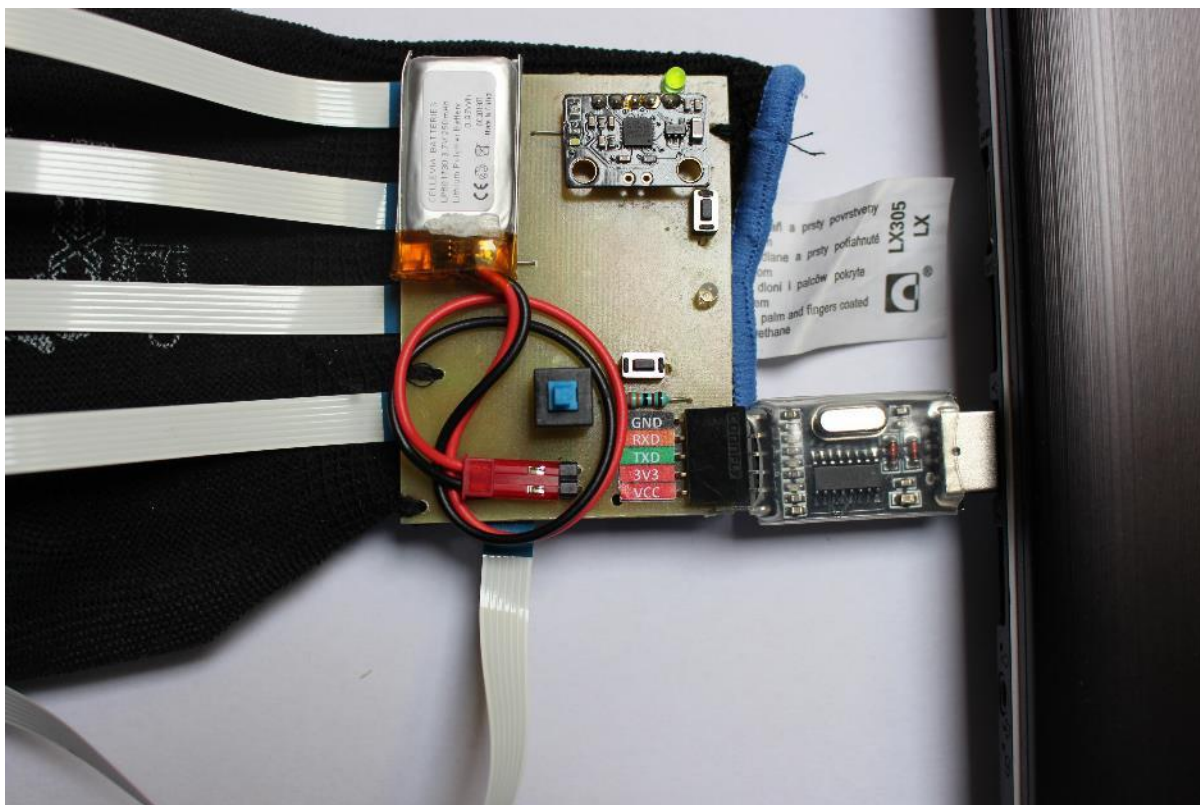
Obrázek 102: Rukavice s přišitým DPS a zapojenými akcelerometry

3.4.5 Krok 4: Programování rukavice

Pro programování rukavice z počítače použijte USB na UART převodník



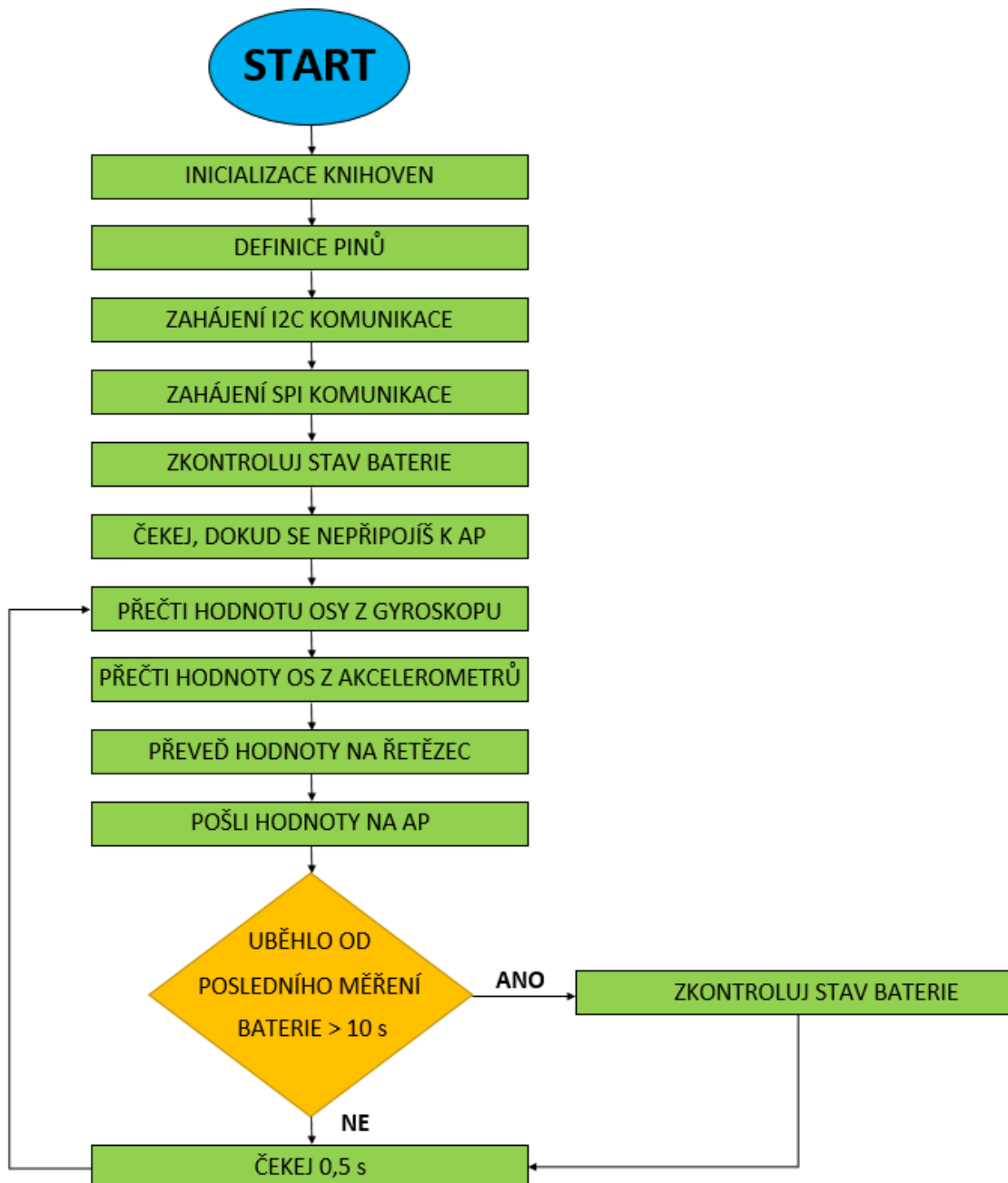
Obrázek 103 a Obrázek 104: UART na USB převodník



Obrázek 105: Rukavice s akcelerometry připojená k notebooku

3.5 Software pro ESP32 na rukavici s akcelerometry

Mikrokontrolér ESP32 nastavený na WiFi režim station (základna) průběžně načítá data z pěti akcelerometrů na prstech rukavice (po SPI) a z gyroskopu (po I2C) a každých 0,5 sekundy je posílá je na ESP32 na robotické ruce pracující v režimu AP (přístupový bod). Mikrokontrolér na rukavici měří po zapnutí napájení a potom každých deset sekund napětí na baterii, které je následně signalizováno počtem bliknutí červené LED (0 - 6). Podrobnější popis programu je znázorněn na algoritmu níže. V programu jsem využila interní Arduino knihovny SPI.h, WiFi.h a Wire.h.



Obrázek 106: Algoritmus programu rukavice

```

1  /*
2  | Name: Robotic Glove      Author: MartinaH      Date: March 2019
3  | This program is based on example of accespoint - station communication from Gyalul:
4  | https://www.instructables.com/id/Accesspoint-Station-Communication-Between-Two-ESP8/
5  | Others sources: internal Arduino libraries: SPI.h, WiFi.h, Wire.h
6  */
7
8  // Internal Arduino libraries
9  #include <Arduino.h>
10 #include <SPI.h>
11 #include <WiFi.h>
12 #include <Wire.h>
13
14 // My libraries
15 #include "LM303D.h" // Library for LSM303D accelerometer
16 #include "main.h" // Library with ESP32 pins definitions
17
18 const int MPU_addr=0x68; // MPU6050 address
19
20 SPIClass * vspi = NULL;
21 static const int spiClk = 1000000; // 1 MHz
22
23 int16_t ACC_data, ACC_Z[5], GyZ;
24
25 char ssid[] = "Wemos_AP"; // SSID of your AP
26 char pass[] = "Wemos_comm"; // password of your AP
27
28 IPAddress server(192,168,4,15); // IP address of the AP
29 WiFiClient client;
30
31 unsigned long actualTime, preTime;
32
33 void setup()
34 {
35     // I2C initiation for MPU6050
36     Wire.begin();
37     Wire.beginTransmission(MPU_addr);
38     Wire.write(0x6B);
39     Wire.write(0);
40     Wire.endTransmission(true);
41
42     Serial.begin(115200);
43
44     // ESP32 pin directions
45     pinMode(5, OUTPUT); // CS0
46     pinMode(4, OUTPUT); // CS1
47     pinMode(0, OUTPUT); // CS2
48     pinMode(2, OUTPUT); // CS3
49     pinMode(15, OUTPUT); // CS4
50     pinMode(32, OUTPUT); // LED
51     pinMode(33, INPUT); // battery
52
53     // SPI initiation for LSM303D
54     vspi = new SPIClass(VSPI);
55     vspi->begin();
56     vspi->beginTransaction(SPISettings(spiClk, MSBFIRST, SPI_MODE3));
57
58     for(int ii = 0; ii < 2; ii++)
59     {
60         for(int i = 0; i < 5; i++)
61         {
62             CS_ON(i); vspi->transfer(CTRL1); vspi->transfer(CTRL1_EN | CTRL1_DR_50Hz); AllCS_OFF;
63             CS_ON(i); vspi->transfer(CTRL2); vspi->transfer(CTRL2_ABW_194Hz); AllCS_OFF;
64         }
65     }

```

```

67 | CheckBattery();
68 |
69 | // WiFi initiation
70 | WiFi.mode(WIFI_STA);
71 | WiFi.begin(ssid, pass);
72 | Serial.println();
73 | Serial.println("Connection to the AP");
74 |
75 | while (WiFi.status() != WL_CONNECTED) {
76 |     Serial.print(".");
77 |     delay(500);
78 | }
79 |
80 | Serial.println();
81 | Serial.println("Robotic Glove");
82 | Serial.println("Connected");
83 | Serial.print("LocalIP:"); Serial.println(WiFi.localIP());
84 | Serial.println("MAC:" + WiFi.macAddress());
85 | Serial.print("Gateway:"); Serial.println(WiFi.gatewayIP());
86 | Serial.print("AP MAC:"); Serial.println(WiFi.BSSIDstr());
87 | }
88 |
89 | void loop()
90 | {
91 |     // Reading gyro Z axis value from MPU6050
92 |     Wire.beginTransmission(MPU_addr);
93 |     Wire.write(0x47);
94 |     Wire.endTransmission(false);
95 |     Wire.requestFrom(MPU_addr,2,true);
96 |     GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
97 |
98 |     // Reading accelerometers Z axis value from LSM303D
99 |     for(int n = 0; n < 5; n++)
100 |     {
101 |         CS_ON(n); vspi->transfer(READ_REG | AUTO_INC | OUT_Z_L_A);
102 |         ACC_data = vspi->transfer16(0x00); AllCS_OFF;
103 |         ACC_Z[n] = ((uint8_t)ACC_data << 8) | (uint8_t)(ACC_data >> 8);
104 |         ACC_Z[n] *= 0.0610;
105 |     }
106 |
107 |     // Connecting to the AP
108 |     client.connect(server, 80);
109 |
110 |     // Converting data to string
111 |     String pos = String(ACC_Z[0]);
112 |     String pos2 = String(ACC_Z[1]);
113 |     String pos3 = String(ACC_Z[2]);
114 |     String pos4 = String(ACC_Z[3]);
115 |     String pos5 = String(ACC_Z[4]);
116 |     String pos6 = String(GyZ);
117 |
118 |     Serial.println("*****");
119 |     Serial.println("Data sent to the AP: ");
120 |     Serial.print(pos + "/t" + pos2 + "/t" + pos3 + "/t" + pos4 + "/t" + pos5 + "/t" + pos6 + "/t");
121 |
122 |     // Sending data to AP
123 |     client.print(pos+"\r");
124 |     client.print(pos2+"\r");
125 |     client.print(pos3+"\r");
126 |     client.print(pos4+"\r");
127 |     client.print(pos5+"\r");
128 |     client.print(pos6+"\r");
129 |     delay(500);
130 |     client.flush();
131 |     client.stop();
132 |
133 |     // Checking the battery state every minute

```



```

133 // Checking the battery state every minute
134 actualTime = millis();
135 if((actualTime - preTime) > 60000)
136 {
137     preTime = actualTime;
138     CheckBattery();
139 }
140 }
141
142 void CheckBattery()
143 {
144     int battery = 0;
145
146     for(int i = 0; i < 20; i++)
147     {
148         battery += analogRead(33);
149     }
150     battery /= 20; // Average of 20 values
151
152     int milivolts = battery * 4.07; // Conversion ADC to mV
153
154     Serial.print("Battery: ");
155     Serial.print(milivolts);
156     Serial.println(" mV");
157
158     int bat_level = 0;
159
160     if(milivolts < 3000) LED_ON;
161     else if(milivolts < 3200) bat_level = 1;
162     else if(milivolts < 3400) bat_level = 2;
163     else if(milivolts < 3600) bat_level = 3;
164     else if(milivolts < 3800) bat_level = 4;
165     else if(milivolts < 4000) bat_level = 5;
166     else bat_level = 6;
167
168     // The number of red LED flashes corresponds to the battery voltage
169     for(int i = 0; i < bat_level; i++) {LED_ON; delay(200); LED_OFF; delay(200);}
170 }
171
172 // This function turning on selected chip select
173 void CS_ON(int num)
174 {
175     switch(num)
176     {
177         case 0: CS0_ON; break;
178         case 1: CS1_ON; break;
179         case 2: CS2_ON; break;
180         case 3: CS3_ON; break;
181         case 4: CS4_ON; break;
182     }
183 }

```

Obrázek 107: Výpis programu pro rukavici – main.cpp


```

1  /*
2  Name: main.h   Author: MartinaH   Date: March 2019
3  Description: ESP32 pins definitions library
4  */
5
6  #define CS0_OFF  digitalWrite(5, HIGH)
7  #define CS0_ON  digitalWrite(5, LOW)
8
9  #define CS1_OFF  digitalWrite(4, HIGH)
10 #define CS1_ON   digitalWrite(4, LOW)
11
12 #define CS2_OFF  digitalWrite(0, HIGH)
13 #define CS2_ON   digitalWrite(0, LOW)
14
15 #define CS3_OFF  digitalWrite(2, HIGH)
16 #define CS3_ON   digitalWrite(2, LOW)
17
18 #define CS4_OFF  digitalWrite(15, HIGH)
19 #define CS4_ON   digitalWrite(15, LOW)
20
21 #define AllCS_ON {CS0_ON; CS1_ON; CS2_ON; CS3_ON; CS4_ON;}
22 #define AllCS_OFF {CS0_OFF; CS1_OFF; CS2_OFF; CS3_OFF; CS4_OFF;}
23
24 #define LED_ON   digitalWrite(32, HIGH)
25 #define LED_OFF  digitalWrite(32, LOW)
26
27 #define VBAT 33
28
29 void CheckBattery();
30 void CS_ON(int num);

```

Obrázek 108: Výpis programu pro rukavici – main.h

```

1  /*
2  Name: LSM303D.h    Author: MartinaH    Date: March 2019
3  Description: LSM303D accelerometer library
4  */
5
6  #define WHO_AM_I 0x0F
7  #define READ_REG 1<<7
8  #define AUTO_INC 1<<6
9
10 #define TEMP_OUT_L 0x05
11 #define TEMP_OUT_H 0x06
12 #define CTRL1 0x20
13 #define CTRL2 0x21
14 #define CTRL5 0x24
15 #define CTRL7 0x26
16 #define STATUS_A 0x27
17 #define OUT_X_L_A 0x28
18 #define OUT_X_H_A 0x29
19 #define OUT_Y_L_A 0x2A
20 #define OUT_Y_H_A 0x2B
21 #define OUT_Z_L_A 0x2C
22 #define OUT_Z_H_A 0x2D
23
24
25 #define CTRL1_DR_3Hz (0b0001 <<4)
26 #define CTRL1_DR_6Hz (0b0010 <<4)
27 #define CTRL1_DR_12Hz (0b0011 <<4)
28 #define CTRL1_DR_25Hz (0b0100 <<4)
29 #define CTRL1_DR_50Hz (0b0101 <<4)
30 #define CTRL1_DR_100Hz (0b0110 <<4)
31 #define CTRL1_DR_200Hz (0b0111 <<4)
32 #define CTRL1_DR_400Hz (0b1000 <<4)
33 #define CTRL1_DR_800Hz (0b1001 <<4)
34 #define CTRL1_DR_1600Hz (0b1010 <<4)
35 #define CTRL1_EN 0b111
36
37 #define CTRL2_ABW_773Hz (0b00<<6)
38 #define CTRL2_ABW_194Hz (0b01<<6)
39 #define CTRL2_ABW_326Hz (0b10<<6)
40 #define CTRL2_ABW_50Hz (0b11<<6)
41
42 #define CTRL2_AFS_2g (0b000<<3)
43 #define CTRL2_AFS_4g (0b001<<3)
44 #define CTRL2_AFS_6g (0b010<<3)
45 #define CTRL2_AFS_8g (0b011<<3)
46 #define CTRL2_AFS_16g (0b100<<3)
47
48 #define CTRL5_TEMP_EN (1<<7)
49 #define CTRL5_M_ODR (001<<2)
50 #define CTRL7_T_ONLY (1<<4)

```

Obrázek 109: Výpis programu pro rukavici – LSM303D.h

Závěr

Cílem této práce bylo sestavit levný, snadno vyrobiteľný a opraviteľný robotický model lidské ruky, vytisknutelný na 3D tiskárně, dostatečně atraktivní pro zajímavou výuku robotiky a programování na školách, vybavený softwarovými funkcemi pro bezdrátovou komunikaci a rozšiřitelný o různé bezdrátové řídicí moduly. Pro stavbu mechanického hardware byla využívána zejména technologie 3D tisku, protože v dnešní době má většina středních odborných škol k dispozici 3D tiskárnu, takže není problém tuto pomůcku vyrobit či případně upravit její design.

Domnívám se, že cíl práce se mi podařilo splnit. Vybrala jsem snadno sestavitelný, ale dostatečně věrně vypadající robotický model lidské ruky konstruktéra grossrc, publikovaný na internetu, upravila jeho mechanickou konstrukci pro možnost připojení levnějších servo motorů MG90S a úchytem pro elektroniku, nakreslila a napájela plošný spoj elektroniky založené na mikrokontroléru ESP32 umožňujícím ovládání přes WiFi a Bluetooth a naprogramovala tři vzorové aplikace pro bezdrátové řízení robotické ruky:

- ovládání z externího bezdrátového modulu (rukavice s akcelerometry);
- webservice síťově ovládatelný z počítače, mobilního telefonu nebo tabletu, zapojitelný do sítě internetu věcí (IoT), umožňující ovládání servomotorů i změnu barev RGB LED kroužku;
- ovládání point – to – point z mobilního telefonu nebo tabletu s operačním systémem Android pomocí jednoduché aplikace napsané v Pocket Code;

Model robotické ruky si jde snadno vytisknout na 3D tiskárně, sestavit dle podrobného návodu uvedeného v této práci, dle výkresové dokumentace v příloze vyrobit plošný spoj elektroniky a jako základ budoucí vlastní tvorby nahrát vzorový software.

Jako příklad externího modulu bezdrátového ovládání jsem sestavila senzorickou rukavici s akcelerometry, které snímají pohyb prstů a přenáší jejich signál prostřednictvím druhého modulu ESP32 po WiFi do robotické ruky, jejíž servomotory poté kopírují pohyb prstů s rukavicí. Stejně jako v případě robotické ruky je v práci podrobně zdokumentován postup sestavení senzorické rukavice, deska plošného spoje i základní programové vybavení. Souvisejícím výrobkem je nabíječka Li-Pol akumulátorů.

Oba výrobky (robotická ruka i senzorická rukavice) jsou vyrobiteľné s dostatečně nízkými výrobními náklady, aby je bylo možno pořídit každému studentovi v rámci výuky robotiky, mechatroniky, 3D tisku nebo programování mikrokontrolérů k docílení skutečně důkladné a zároveň atraktivní praktické výuky v těchto oborech. Kalkulovaná cena potřebných materiálových součástí (tiskové struny, elektronické komponenty) je v případě robotické ruky 1847 Kč, v případě senzorické rukavice 1020 Kč, v případě nabíječky 40 Kč, celkově tedy méně než 3000 Kč.

Software je psán ve vývojovém prostředí PlatformIO v platformě pro Arduino IDE, které je dostatečně jednoduché a výborně vysvětlené v dokumentaci a tutoriálech na webu, takže je použitelný i pro začátečníky v programování. Výukové modely by tak měly být snadno upravitelné i samotnými studenty při výuce programování mikrokontrolérů a síťových aplikací.

Ve vývoji výukového robotického modelu lidské ruky bych chtěla dále pokračovat s použitím dalších externích bezdrátových řídicích modulů pro pokročilejší sensorické ovládání člověkem, ať již je to ovládání pomocí hlasu, mimiky, svalových kontrakcí nebo mozkovými impulzy. Ráda bych rovněž oslovila firmu Google s žádostí o poskytnutí jejich novinky v bezdrátovém ovládání - mikroradaru Google Soli^[24], který dokáže velice citlivě detekovat pohyby lidské ruky.

Veškerá dokumentace projektu je publikována na veřejném git serveru www.gitlab.com/MartinaH/SOC2019, kde bude průběžně aktualizována. Po dokončení projektu bych jej chtěla zdokumentovat i v anglickém jazyce pro web www.instructables.com a pro soutěž EUCYS 2019.

Seznam použitých odkazů

1. **Studie o robotizaci: Česká televize**[online]. [Cit. 20. 3. 2019]. Dostupné z URL: <https://ct24.ceskatelevize.cz/veda/2108561-ceska-republika-je-robotizaci-ohrozena-nejvic-v-evrope-tvrdi-studie>
2. **Harnessing automation for a future that works: McKinsey & Company**[online]. [Cit. 20. 3. 2019]. Dostupné z URL: <https://www.mckinsey.com/featured-insights/digital-disruption/harnessing-automation-for-a-future-that-works>
3. **ESP32: Wikipedia**[online]. [Cit. 20. 3. 2019]. Dostupné z URL: <https://en.wikipedia.org/wiki/ESP32>
4. **Datasheet pro ESP32: Espressif**[online]. [Cit. 20. 3. 2019]. Dostupné z URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
5. **Datasheet pro Arduino UNO: Farnell**[online]. [Cit. 20. 3. 2019]. Dostupné z URL: <https://www.farnell.com/datasheets/1682209.pdf>
6. **Pinout ESP32 DevKit: Components101**[online]. [Cit. 20. 3. 2019]. Dostupné z URL: <https://components101.com/microcontrollers/esp32-devkitc>
7. **Foto ESP32 Wrover: Espressif**[online]. [Cit. 20. 3. 2019]. Dostupné z URL: https://www.espressif.com/en/media_overview/news/new-espressif-module-esp32-wrover-b?position=1&list=M_5Wl69hRnZsofhJtj9HgZQgo_EP4CzrrSo4qLD5ys
8. **Humanoid Robotic Hand: Thingiverse**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://www.thingiverse.com/thing:2269115>
9. **Eshop s elektronikou: TME**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://www.tme.eu/cz>
10. **Eshop s elektronikou: Laskarduino**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://laskarduino.cz/>
11. **Obchodní portál: Aukro**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://aukro.cz/>
12. **Nabíječka Li-Ion baterií: Laskarduino**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: https://laskarduino.cz/napajeni-zdroje/120007-59-nabijecka-li-ion-clanku-tp4056.html#/47-konektor_usb-miniusb
13. **Vývojové prostředí: PlatformIO**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://platformio.org/>
14. **AP – station komunikace: Instructables**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://www.instructables.com/id/Accesspoint-Station-Communication-Between-Two-ESP8/>
15. **Příklad webserveru: Github**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://github.com/espressif/arduino-esp32/blob/master/libraries/WebServer/examples/PathArgServer/PathArgServer.ino>
16. **Vývojové prostředí Pocket Code: Google Play**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://play.google.com/store/apps/details?id=org.catrobat.catroid&hl=cs>
17. **Pocket Code tutoriály v češtině: Youtube**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://www.youtube.com/channel/UCvYlxc7kD1IKFYwugltXMLg/videos>
18. **Příklad Bluetooth komunikace: CircuitDigest**[online]. [Cit. 21. 3. 2019]. Dostupné z URL: <https://circuitdigest.com/microcontroller-projects/using-classic-bluetooth-in-esp32-and-toogle-an-led>

19. **Akcelerometr Pololu - 2127: Pololu[online].** [Cit. 20. 3. 2019]. Dostupné z URL: <https://www.pololu.com/product/2127>
20. **Foto akcelerometru Pololu - 2127: Pololu[online].** [Cit. 20. 3. 2019]. Dostupné z URL: <https://www.pololu.com/product/2127#lightbox-picture0|4936;main-pictures>
21. **Gyroskop MPU6050: Arduino Playground[online].** [Cit. 20. 3. 2019]. Dostupné z URL: <https://playground.arduino.cc/Main/MPU-6050>
22. **Cena MPU6050: Eshop Postav Robotu[online].** [Cit. 20. 3. 2019]. Dostupné z URL: <https://www.postavrobotu.cz/Digitalni-akcelerometr-gyroskop-MPU6050-modul-3-osy-d119.htm>
23. **Datasheet pro MCP73831: Sparkfun Electronics[online].** [Cit. 20. 3. 2019]. Dostupné z URL: https://cdn.sparkfun.com/assets/learn_tutorials/6/9/5/MCP738312.pdf
24. **Mikroradar Google Soli: Google Soli[online].** [Cit. 21. 3. 2019]. Dostupné z URL: <https://atap.google.com/soli/>

Seznam obrázků

Obrázek 1: Foto výsledné sestavy univerzálního řídicího systému.....	11
Obrázek 2: Schéma pinů vývojové desky ESP32 DevKit.....	13
Obrázek 3: Foto ESP32 Wrover.....	13
Obrázek 4: Modely prstů robotické ruky.....	14
Obrázek 5: Model dlaně robotické ruky.....	15
Obrázek 6: Model předloktí robotické ruky.....	14
Obrázek 7: Model zápěstí robotické ruky.....	15
Obrázek 8: Model navijáků silonu.....	14
Obrázek 9: Model palce robotické ruky.....	15
Obrázek 10: Model kolečka na otáčení předloktí.....	15
Obrázek 11: Model koleček na servomotory.....	15
Obrázek 12: Model držáku elektroniky.....	15
Obrázek 13: Model krytky hřbetu ruky.....	16
Obrázek 14: Model držáku servomotorů.....	15
Obrázek 15: Potřebné komponenty ke stavbě mechanického modelu robotické ruky.....	17
Obrázek 16: Navlečení článků prstu na pruženku.....	18
Obrázek 17: Navlečené články prstu na pružence.....	17
Obrázek 18: Připevnění malíčku k dlani.....	19
Obrázek 19: Připevnění zbylých prstů k dlani.....	18
Obrázek 20: Přišroubování unašeče k servomotoru.....	19
Obrázek 21: Servomotor s unašečem.....	18
Obrázek 22: Ustřížení části unašeče.....	19
Obrázek 23: Servomotor s ustříženou částí unašeče.....	18
Obrázek 24: Nanesení lepidla do otvoru na palci.....	20
Obrázek 25: Vložení servomotoru do otvoru na palci.....	19
Obrázek 26 a Obrázek 27: Přilepení oboustranné lepicí pásky na servomotor.....	19
Obrázek 28 a Obrázek 29: Vložení servomotoru do otvoru v dlani.....	19

Obrázek 30: Nanesení lepidla na výstupky zápěstí	21
Obrázek 31: Přilepení zápěstí k dlani	20
Obrázek 32: Nanesení lepidla na spodní okraj zápěstí.....	21
Obrázek 33: Přilepení zápěstí k předloktí	20
Obrázek 34: Přišroubování servomotorů k držáku	21
Obrázek 35: Nanesení lepidla na kolečko pro servomotor.....	22
Obrázek 36: Přilepení koleček k navijákům silonu.....	21
Obrázek 37: Nasazení koleček s navijáky na servomotory.....	22
Obrázek 38: Kolečka nasazená na servomotorech.....	21
Obrázek 39: Přišroubování držáku servomotorů k předloktí.....	22
Obrázek 40: Rybářský vlasec	23
Obrázek 41: Kousky rybářského vlasce s uzly na konci.....	22
Obrázek 42, Obrázek 43 a Obrázek 44 : Provlečení silonu články prstů	23
Obrázek 45 : Provlečení silonu otvory v držáku servomotorů	23
Obrázek 46 : Provlečení silonu navijáky a zauzlování na konci	24
Obrázek 47 : Zadní krytka na hřbet.....	25
Obrázek 48 : Obklepení okraje krytky oboustrannou.....	24
Obrázek 49 : Pohled na robotickou ruku zezadu	25
Obrázek 50 : Vložení servomotoru do sloupku pro DPS.....	25
Obrázek 51 : Přišroubování kolečka pro otáčení předloktím k předloketnímu dílu	26
Obrázek 52: Nasazení horního dílu robotické ruky na spodní sloupek pro elektroniku	26
Obrázek 53: Schéma zapojení řídicího systému robotické ruky	27
Obrázek 54: Pájecí schéma DPS robotické ruky	29
Obrázek 55: Foto DPS robotické ruky	28
Obrázek 56: Pájecí schéma DPS robotické ruky	29
Obrázek 57: Foto DPS robotické ruky	28
Obrázek 58: Potřebné komponenty ke kompletaci robotické ruky s elektronikou	30
Obrázek 59: Nastavení step up měniče na 5V	31
Obrázek 60: Zakápnutí potenciometru lepidlem.....	31
Obrázek 61: DPS robotické ruky bez ESP32	33
Obrázek 62: DPS robotické ruky s ESP32	32
Obrázek 63: DPS přišroubovaná k robotické ruce	32
Obrázek 64: RGB LED kroužek	34
Obrázek 65: RGB LED kroužek přilepený zespod robotické ruky.....	33
Obrázek 66: Pohled na DPS ve sloupku robotické ruky.....	33
Obrázek 67: Sestavená robotická ruka	34
Obrázek 68: Algoritmus programu robotické ruky	35
Obrázek 69: Výpis programu pro robotickou ruku pro komunikaci s bezdrátovými moduly	38
Obrázek 70: Webserver robotické ruky	39
Obrázek 71: Výpis programu pro webserver – main.cpp.....	42
Obrázek 72: Fotografie obrazovky mobilní aplikace	42
Obrázek 73: Výpis programu k ovládání přes Bluetooth pro ESP32.....	43
Obrázek 74: Výpis programu pro mobilní aplikaci v Pocket Code	46
Obrázek 75: Akcelerometr Pololu – 2127	48
Obrázek 76: Modul gyroskopu s MPU6050.....	47
Obrázek 77: Schéma zapojení hlavní desky rukavice	48

Obrázek 78: Pájecí schéma DPS hlavní desky rukavice	50
Obrázek 79: Foto DPS hlavní desky rukavice.....	49
Obrázek 80: Pájecí schéma DPS hlavní desky rukavice	50
Obrázek 81: Foto DPS hlavní desky rukavice.....	49
Obrázek 82: Schéma zapojení desky s akcelerometry.....	51
Obrázek 83: Pájecí schéma desky s akcelerometry	52
Obrázek 84: Foto desky s akcelerometry.....	51
Obrázek 85: Pájecí schéma desky s akcelerometry	52
Obrázek 86: Foto desky s akcelerometry.....	51
Obrázek 87: Model držáku nabíječky	53
Obrázek 88: Foto držáku nabíječky.....	52
Obrázek 89: Schéma zapojení nabíječky Li-Pol baterií	53
Obrázek 90: Pájecí schéma DPS nabíječky	54
Obrázek 91: Foto DPS nabíječky	53
Obrázek 92: Pájecí schéma DPS nabíječky	54
Obrázek 93: Foto DPS nabíječky	53
Obrázek 94: Foto nabíječky Li-Pol.....	55
Obrázek 95: Foto nabíječky s vloženým akumulátorem.....	54
Obrázek 96: Potřebné komponenty ke stavbě rukavice.....	55
Obrázek 97: Provrtaný DPS – strana součástek	57
Obrázek 98: Provrtaný DPS – strana spojů	56
Obrázek 99: Pravá rukavice s gumou do kalhot.....	57
Obrázek 100: Rukavice s přišitými kapsami na akcelerometry	56
Obrázek 101: Rukavice s vloženými akcelerometry	57
Obrázek 102: Rukavice s přišitým DPS a zapojenými akcelerometry	57
Obrázek 103 a Obrázek 104: UART na USB převodník.....	58
Obrázek 105: Rukavice s akcelerometry připojená k notebooku.....	58
Obrázek 106: Algoritmus programu rukavice	59
Obrázek 107: Výpis programu pro rukavici – main.cpp.....	62
Obrázek 108: Výpis programu pro rukavici – main.h.....	63
Obrázek 109: Výpis programu pro rukavici – LSM303D.h.....	64

Seznam tabulek

Tabulka 1: Srovnání ESP32 s Arduino UNO	13
Tabulka 2: Tabulka tisku 3D dílů modelu robotické ruky	16
Tabulka 3: Tabulka elektronických součástek robotické ruky	29
Tabulka 4: Tabulka součástek hlavní DPS rukavice	50
Tabulka 5: Tabulka součástek převodních desek s akcelerometry.....	52
Tabulka 6: Tabulka tisku 3D modelu krabičky	52
Tabulka 7: Tabulka součástek nabíječky Li-Pol baterií.....	54

Seznam zkratek

ADC	Analogově/digitální převodník
AP	Access point (přístupový bod)
DPS	Deska plošného spoje
GPIO	Univerzální vstup/výstup
I2C	Integrované sériové rozhraní
I2S	Integrované zvukové rozhraní
PWM	Pulzně šířková modulace
SOČ	Středoškolská odborná činnost
SPI	Sériové periferní rozhraní
UART	Univerzální asynchronní přijímač-vysílač

Seznam příloh

Příloha 1. Schémata elektroniky

Příloha 2. Mechanické modely

Příloha 3. Software