

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

Rukavice se senzory pohybu

Vojtěch Vrba

Vedoucí: prof. Ing. Jan Holub, Ph.D.
Obor: Kybernetika a robotika
Studijní program: Kybernetika a robotika
Květen 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vrba** Jméno: **Vojtěch** Osobní číslo: **474409**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Rukavice se senzory pohybu

Název bakalářské práce anglicky:

Gloves Equipped with Motion Sensors

Pokyny pro vypracování:

Navrhněte a realizujte senzorický systém pro sběr dat, popisujících pohyb článků prstů lidské ruky. Systém musí obsahovat 6-DOF senzory, např. ST LSM6DSL a vhodný mikroprocesor pro zpracování dat, dále může obsahovat další přídavné senzory (magnetometr, tlakoměr apod.) a modul pro bezdrátový přenos dat (např. Bluetooth). Funkčnost demonstруйте s využitím jednoduchého SW pro vizualizaci dat.

Seznam doporučené literatury:

- [1] Niazmand, Khalil & Tonn, Karin & Kalaras, Anastasios & Fietzek, Urban & Mehrkens, Jan-Hinnerk & Lueth, Tim. (2011): Quantitative evaluation of Parkinson's disease using sensor based smart glove. Proceedings - IEEE Symposium on Computer-Based Medical Systems. 1-8. 10.1109/CBMS.2011.5999113.
[2] J. Connolly, J. Condell, B. O'Flynn, J. T. Sanchez and P. Gardiner: "IMU Sensor-Based Electronic Goniometric Glove for Clinical Finger Movement Analysis," in IEEE Sensors Journal, vol. 18, no. 3, pp. 1273-1281, 1 Feb.1, 2018. doi: 10.1109/JSEN.2017.2776262
[3] O'Connor TF, Fach ME, Miller R, Root SE, Mercier PP, Lipomi DJ (2017): The Language of Glove: Wireless gesture decoder with low-power and stretchable hybrid electronics. PLoS ONE 12(7): e0179766. <https://doi.org/10.1371/journal.pone.0179766>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

prof. Ing. Jan Holub, Ph.D., katedra měření FEL

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

Datum zadání bakalářské práce: **11.02.2020**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce:

do konce letního semestru 2020/2021

prof. Ing. Jan Holub, Ph.D.
podpis vedoucí(ho) práce

_____ podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

_____ Datum převzetí zadání

_____ Podpis studenta

Poděkování

V první řadě bych chtěl poděkovat vedoucímu práce, panu prof. Ing. Janu Holubovi, Ph.D., za odborné vedení v průběhu celé práce, konzultace a ochotu.

Speciální poděkování bych chtěl vyjádřit zejména

panu Ing. Oldřichu Šimkovi z firmy PragoBoard s. r. o. za nezištnou výrobu ohebných plošných spojů dle návrhů,

panu Tomáši Pokornému z firmy STMicroelectronics za nezištné zajištění kitů i vzorků senzorů a modulů,

paní Ing. Ivaně Beshajové Pelikánové, Ph.D., z Laboratoře vývoje a realizace FEL ČVUT za osazení všech desek senzorů.

Stejně velké poděkování patří i mým blízkým a mé rodině za veškerou podporu, které se mi od nich při studiu na ČVUT dostává.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 22. května 2020

Abstrakt

Bakalářská práce se zabývá vývojem rukavic osazených senzory pohybu. Výsledkem celé práce by mělo být funkční řešení včetně prostorové vizualizace naměřených ohybů prstů, tedy gestikulace rukou uživatele rukavic. Čtenář je nejprve seznámen s teorií komunikace pomocí protokolu Bluetooth Low Energy. V další kapitole jsou vysvětleny jednotlivé aspekty hardwarového návrhu plošných spojů. Další část práce se zabývá softwarovou implementací aplikace pro programovatelný BLE modul umístěný na rukavicích. Pak je uveden základ matematické teorie kvaternionů ve spojení s rotacemi a orientacemi v prostoru, včetně vysvětlení odvození algoritmu pro určení orientace. Dále je vysvětlen způsob implementace vizualizačních skriptů v jazyce Python. V posledních částech práce je uvedeno několik závěrů získaných ze zkoušky funkčnosti rukavic, včetně autorových námětů pro další vývoj.

Klíčová slova: snímání gest rukou, rukavice, pohybové senzory

Vedoucí: prof. Ing. Jan Holub, Ph.D.

Abstract

This bachelor thesis focuses on the development of gloves equipped with motion sensors. The result of this work is supposed to be a functional device including 3D visualisation of measured bending of all fingers, so hand gestures can be determined. First of all, the theory of Bluetooth Low Energy devices is presented. The next chapter contains an explanation of hardware design details. Then the software implementation for the programmable BLE module is presented. Explanation of the theory of quaternion representation of the rotation and orientation in space for Madgwick's AHRS algorithm can be also found in this work. Python language has been chosen for implementing the visualisation. The thesis is completed with achieved results and ideas for further development.

Keywords: gesture monitoring, gloves, IMU sensors, Bluetooth Low Energy

Title translation: Gloves Equipped with Motion Sensors

Obsah

1 Úvod	1	4.3.7 Struktura paketů	30
2 Základy Bluetooth Low Energy	3	4.4 Nahrávání programu do modulu	31
2.1 Historický vývoj	3	5 Zpracování a vizualizace dat	33
2.2 BLE protokol	3	5.1 Orientace a rotace	33
2.2.1 Fyzická vrstva	4	5.2 Použití kvaternionů pro popis orientace	33
2.2.2 Spojová vrstva	5	5.3 Madgwickův algoritmus	37
2.2.3 L2CAP	6	5.3.1 Orientace z gyroskopických dat	37
2.2.4 SMP & GAP	6	5.3.2 Orientace v homogenních polích	38
2.2.5 ATT & GATT	7	5.3.3 Optimalizační algoritmus	38
2.2.6 Aplikační vrstva	8	5.3.4 Bloková schémata algoritmu	38
2.3 Adresy zařízení	8	5.3.5 Implementace	40
2.4 Základní omezení	9	5.4 Jiné algoritmy	41
2.4.1 Nejvyšší datová propustnost	9	5.4.1 Mahonyho algoritmus	41
2.4.2 Maximální dosah	10	5.4.2 Kalmanův filtr	41
3 Zapojení pro rukavice	13	5.5 Vizualizace	41
3.1 Požadavky	13	5.5.1 Výběr programovacího jazyka	41
3.2 Napájecí obvod	13	5.5.2 Třírozměrný model ruky	42
3.3 Senzory	15	5.5.3 Způsob implementace	42
3.3.1 Pohybové senzory	15	6 Zkouška zařízení	45
3.3.2 Magnetometr	16	6.1 Ověření funkce multiplexoru	45
3.3.3 Barometrický senzor	17	6.2 Vizualizace nezpracovaných dat	46
3.4 Bluetooth modul	17	6.3 Vizualizace vypočtených úhlů	47
3.5 Komunikace s PC	20	6.4 Prostorová vizualizace	50
3.6 Připojení vícera I2C sběrnic	21	6.5 Měření spotřeby	51
3.7 Indikační LED diody	22	7 Diskuze	53
3.8 Konektory	23	7.1 Objevené chyby a nedostatky	53
4 Aplikace pro modul	25	7.2 Další vývoj	54
4.1 Vývojové prostředí	25	8 Závěr	55
4.2 Knihovny	25	A Schémata zapojení	57
4.3 Implementace aplikace	26	B Seznam součástek	65
4.3.1 Způsob implementace	26	C Návrh DPS	69
4.3.2 Nastavení BLE stacku	26	D Obsah přiloženého CD	73
4.3.3 Použité periferie modulu	26	E Bibliografie	75
4.3.4 Přístup k senzorům	27		
4.3.5 Funkce indikačních LED diod	29		
4.3.6 Nastavení GAP & GATT	30		

Obrázky

2.1 Frekvence jednotlivých kanálů používaných v BLE	4
2.2 Časování datového přenosu v BLE v5.0	9
3.1 Vybraný Li-Po akumulátor	13
3.2 Zapojení nabíjecího obvodu MCP73811	14
3.3 Zapojení napěťového regulátoru LTC1844	14
3.4 Zkušební desky obou pohybových senzorů	15
3.5 Zapojení senzoru LSM6DSL umístěného na hřbetu ruky	16
3.6 Zapojení senzoru LIS2MDL umístěného na hřbetu ruky	17
3.7 Zapojení senzoru LPS22HB umístěného na hřbetu ruky	17
3.8 Návrhy adaptorů pro komunikační moduly	18
3.9 Osazené adaptory s komunikačními moduly	19
3.10 Zapojení Bluetooth modulu BlueNRG-M2SA	20
3.11 Zapojení USB-UART převodníku PL2303SA	21
3.12 Zapojení multiplexoru/demultiplexoru	22
3.13 Zapojení indikačních LED diod	22
3.14 Konektory pro akumulátor a programátor	23
3.15 Konektory pro plošné spoje na prstech	23
4.1 Nahrávání programu v aplikaci BlueNRG-1 ST-Link Utility	32
5.1 Vyznačení os gyroskopu vůči rámu senzoru LSM6DSL	37
5.2 Vyznačení os vůči rámcům senzorů	39
5.3 Blokové schéma Madgwickova algoritmu pro 6-DOF senzory	40
5.4 Blokové schéma Madgwickova algoritmu pro 9-DOF senzory	40
5.5 Vykreslený model lidské ruky v programu 3D prohlížeč (součást Windows 10)	42
6.1 Zaslání identifikačního požadavku po lince I2C	45
6.2 Odpověď na identifikační požadavek po lince I2C	46
6.3 Měření intervalů mezi pakety ...	46
6.4 Histogram pro měření akcelerometrem	47
6.5 Histogram pro měření gyroskopem.....	47
6.6 Histogram pro měření magnetometrem	47
6.7 Časový průběh vypočteného úhlu natočení rukavice okolo osy X	48
6.8 Časový průběh vypočteného úhlu natočení rukavice okolo osy Y	48
6.9 Časový průběh vypočteného úhlu natočení rukavice okolo osy Z	48
6.10 Fotografie navlečené rukavice .	50
6.11 Snímek vizualizace	51
6.12 Zapojení obvodu pro měření proudu	51
6.13 Časový průběh napájecího proudu po přivedení napájení	52
6.14 Časový průběh napájecího proudu při připojení BLE klienta ..	52
A.1 Schéma zapojení pro palec	57
A.2 Schéma zapojení pro ukazováček a prsteníček	58
A.3 Schéma zapojení pro prostředníček a malíček	59
A.4 Schéma zapojení pro ruku - napájecí obvody	60
A.5 Schéma zapojení pro ruku - konektory	61
A.6 Schéma zapojení pro ruku - senzory	62

A.7 Schéma zapojení pro ruku - BLE modul a multiplexor	63
C.1 Návrh DPS pro palec	69
C.2 Návrh DPS pro ukazováček	69
C.3 Návrh DPS pro prostředníček ..	69
C.4 Návrh DPS pro prsteníček	69
C.5 Návrh DPS pro malíček	69
C.6 Návrh DPS pro levou a pravou ruku	70
C.7 Fotografie vyrobené DPS pro pravou ruku	71

Tabulky

2.1 Model BLE protokolu od nejvyšší vrstvy po nejnižší	4
2.2 Struktura BLE paketu (spojová vrstva)	5
2.3 Vzdálenosti od vysílače s útlumem od 50 do 110 dB	10
3.1 Srovnání parametrů pohybových senzorů	16
3.2 Srovnání parametrů programovatelných modulů	18
4.1 Pořadí senzorů dle přístupu	29
4.2 Struktura datového paketu	31
4.3 Popis debugovacího konektoru CN2 na kitu STM32F4-Discovery .	32
B.1 Součástky pro DPS na palec	65
B.2 Součástky pro DPS na ostatních prstech	65
B.3 Součástky pro napájecí obvod na hřbetu ruky	66
B.4 Součástky pro zbytek obvodu na hřbetu ruky	67
D.1 Obsah složek na přiloženém CD	73

Kapitola 1

Úvod

Cílem této bakalářské práce byl kompletní návrh a konstrukce rukavic osazených vhodným počtem pohybových senzorů společně s komunikačním modulem, implementace jeho řídicího programu, a implementace vybraných algoritmů pro zpracování naměřených dat, včetně jednoduché aplikace určené pro třírozměrnou vizualizaci gestikulace podle orientace článků prstů ruky uvnitř rukavic.

Snímáním gestikulace rukou se zabývá mnoho různých prací a jsou vybudovány různé přístupy k řešení této problematiky. Cílem těchto prací bývá využití takto získaných dat pro virtuální realitu (např. v [1]), rozpoznávání a překlad znakového jazyka neslyšících do psaného nebo mluveného slova v reálném čase (např. v [2]), ovládání různých zařízení bez nutnosti přímého kontaktu (např. v [3]), zhodnocení zdravotního stavu pacientů (např. s roztroušenou sklerózou nebo Parkinsonovou chorobou, např. v [4], [5]) aj. Zevrubné srovnání různých komerčních i nekomerčních řešení lze najít např. v článku [6].

Uvádím zde zjednodušené rozdělení existujících řešení podle způsobu monitorování lidské ruky:

1. **Měření mechanického ohybu** vodivého materiálu umístěného na prstech. Tento způsob umožňuje velmi levnou realizaci celého zařízení, jelikož konstrukce ohybových senzorů se skládá ze snadno dostupných materiálů (např. polyimidové odporové fólie, pokovených textilií běžně používaných v kapacitních dotykových displejích aj.). Přesnost měření s vlastnoručně vyrobeným senzorem se pak odvíjí jak od dodržení přesného výrobního postupu, až po kvalitu měření změny nominálního odporu nebo kapacity). Tímto se zabývají například články [7], [8], [1] nebo [9].
2. **Zpracování vizuálních dat** získaných z kamery zaznamenávající pohyb rukou. Nespornou výhodou je opět snadná dostupnost vhodných kamer, není obvykle nutné konstruovat vlastní zařízení, a tím pádem ani uživatelé nejsou omežováni nošením sensorických zařízení. Nevýhodou je předpoklad dobrých vizuálních podmínek (kvalitní záběr na obě ruky). Vizuální data poté bývají zpracována za pomoci strojového

učení. Tento přístup prezentují např. články [3], [10].

- 3. Použitím pohybových senzorů** (angl. *inertial measurement unit*, IMU) umístěných na požadovaných částech ruky. Kvůli komerčním cenám senzorů se pak realizace obvykle pohybuje v řádu tisíců Kč. V člancích [4], [5] lze najít informace o možných implementacích tohoto přístupu. V druhém z nich se zapojení nejvíce podobá mému, uvedenému v této práci, s umístěním pohybových senzorů na každém článku prstů lidské ruky.

Kapitola 2

Základy Bluetooth Low Energy

2.1 Historický vývoj

Specifikace Bluetooth Low Energy (také Bluetooth Smart, dále jen BLE) je postupně vyvíjena od roku 2001, kdy byl firmou Nokia převzat základní koncept „standardní“ Bluetooth technologie (přesněji Bluetooth Basic Rate/Enhanced Data Rate protokolu, BR/EDR), s cílem výrazně snížit energetické nároky této technologie a její cenu, a dostat ji tak do sféry přenosných a IoT zařízení, která potřebují vysílat a přijímat pouze malé množství dat v krátkém dosahu. V roce 2010 byla uveřejněna specifikace BLE v4.0, která se v moderních dotykových telefonech začala objevovat v průběhu následujících let.

Roku 2016 byla uveřejněna nejnovější verze, specifikace BLE v5.0 (komerčně nazývané pouze Bluetooth 5). Výrazné rozdíly přišly v podobě čtyřnásobného dosahu (díky integraci výkonnějších vysílačů a zlepšení parametrů fyzické vrstvy), dvojnásobné rychlosti (zkrácením časových intervalů v době přenosu, konkrétně zvýšením modulační rychlosti z 1 Mb/s na 2 Mb/s) a až osminásobného počtu datových bajtů, které je možné zaslat v jediném paketu.

Je nutné poznamenat, že technologie BLE není zpětně kompatibilní s původní technologií BR/EDR. Využívá totiž jiných (jednodušších) modulačních technik a i softwarové implementace jsou velmi odlišné. Obě technologie ale využívají stejné frekvenční ISM pásmo 2450 MHz (podobně jako např. IEEE 802.11/Wi-Fi). [11]

2.2 BLE protokol

BLE protokol (BLE stack) lze dekomponovat do jednotlivých vrstev, které odpovídají některým vrstvám referenčního ISO/OSI modelu. Ty je možné rozdělit do tří hlavních skupin, které jsou:

- **Software** (též Application), tj. uživatelská aplikace, která komunikuje s firmwarem (odpovídá aplikační vrstvě)

- **Firmware** (také Host), což jsou horní vrstvy modelu (bez aplikační vrstvy)
- **Hardware** (také Controller), která se skládá ze spojové a fyzické vrstvy modelu

Pro komunikace firmwaru s hardwarem je poskytován protokol Host Controller Interface (HCI). Běžně mohou být tyto tři části rozděleny mezi 2 integrované obvody nebo mohou být všechny v jednom programovatelném modulu (System-on-Chip, SoC), čehož jsem využil v této aplikaci.

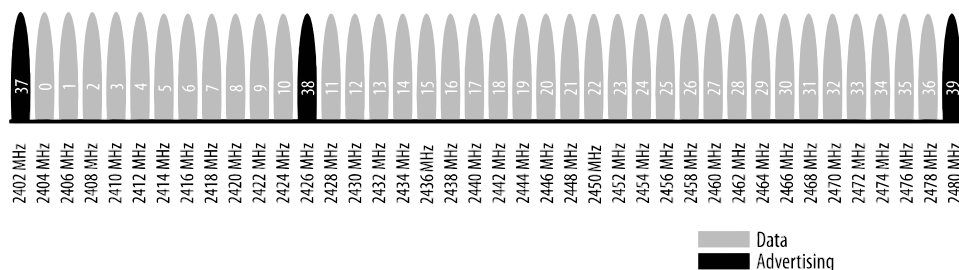
Aplikační vrstva		Software (Application)
GAP	GATT	Firmware (Host)
SMP	ATT	
L2CAP		
Spojová vrstva		Hardware (Controller)
Fyzická vrstva		

Tabulka 2.1: Model BLE protokolu od nejvyšší vrstvy po nejnižší

2.2.1 Fyzická vrstva

Fyzickou vrstvou BLE stacku (označovanou jako BLE PHY) tvoří samotný vysílač a přijímač elektromagnetických vln, modulátor a demodulátor signálu.

Frekvenční spektrum je rozděleno do celkem 40 kanálů, jak ukazuje obrázek 2.1 (převzato z [11], str. 17). Tři z těchto kanálů jsou využity pro tzv. *advertising* (neaktivní komunikace), kdy jeden síťový uzel (angl. *broadcaster* nebo *advertiser*) vysílá všem okolním uzlům (angl. *observers* nebo *scanners*) informace o svém nastavení (identifikačních údajích, profilech, službách atd.), sloužící k následujícímu navázání aktivního spojení s některým konkrétním uzlem (angl. *slave*). Zbývající kanály pak slouží pro aktivní obousměrnou komunikaci a posílání dat mezi *master* a *slave* uzly.



Obrázek 2.1: Frekvence jednotlivých kanálů používaných v BLE

Pro modulaci binárních dat se používá modulace GFSK (Gaussian Frequency

Shift Keying), kterou používají i některé další bezdrátové standardy. Modulační rychlost je v BLE v5.0 fixní a to 2 Mb/s.

■ 2.2.2 Spojová vrstva

Spojová vrstva komunikuje přímo s vrstvou fyzickou, a skládá se obvykle z jednoúčelově naprogramovaných součástí. Nejdůležitější funkce, které má na starost, jsou:

- Tvorbu paketů a časování bezdrátové komunikace
- Generování a ověřování CRC
- *Data whitening* (transformování dat do vektoru bílého šumu)
- Generování náhodných čísel
- Šifrování AES

Také se zde udržují informace o tom, jak probíhá komunikace s ostatními uzly Bluetooth sítě. Podle situace probíhá aktivní (*data connection*) nebo neaktivní (*advertising*) komunikace.

V tabulce 2.2 je vyznačena struktura běžného BLE paketu (angl. *protocol data unit*, PDU). V BLE komunikaci jsou všechny bity vysílány v pořadí LSB-first.

Úvodní bajt(y)	Přístupová adresa	Hlavička	Data	MIC	CRC
1 až 2 B	4 B	2 B	0 až 251 B	4 B	3 B

Tabulka 2.2: Struktura BLE paketu (spojová vrstva)

- **Úvodní bajty** (angl. *preamble*) slouží k označení, zda je použita modulační rychlost 1 Mb/s (pak je zde pouze 1 úvodní bajt) a nebo 2 Mb/s (pak je přidáván druhý úvodní bajt). Bity v těchto bajtech mají pouze 2 možné hodnoty, ve kterých se střídají obě možné hodnoty bitů, tedy hodnoty začínají buď nulou nebo jedničkou, podle prvního bitu přístupové adresy. Z této sekvence bitů je schopen přijímač automaticky určit zisk, na který se musí naladit.
- **Přístupová adresa** (angl. *access address*) slouží k odlišení více BLE zařízení vysílající na stejném kanálu, tedy je volena náhodně při startu zařízení a je jedinečná pro každé připojené zařízení.
- **Hlavička** (angl. *PDU header*) se skládá ze dvou bajtů. První bajt má svou vlastní bitovou strukturu, která liší podle toho, o jaký typ paketu se jedná (*advertising*, *direct connection*, *non-connectable* a několik dalších). Druhý bajt obsahuje číslo označující počet datových bajtů, v módu *advertising* je to nejvýše 31 B, v aktivní komunikaci nejvýše 251 B.

- **Data** (angl. *PDU payload*) jsou cílová data, která jsou odesílána a přijímána.
- **MIC** (*Message Integrity Check*, PDU MIC) je přidáváno při použití šifrované komunikace.
- **CRC** (*Cyclic Redundancy Check*) je přidáno na konec paketu, a slouží k odhalení chyb lichých násobností, případně dvojnásobných a čtyřnásobných chyb. Generující polynom má tvar $x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$, je tedy zvoleno 24bitové CRC.

■ 2.2.3 L2CAP

L2CAP (Logical Link Control and Adaptation Protocol) propojuje spojovou vrstvu s vyššími vrstvami přes sériové rozhraní HCI (Host-Controller Interface). Má dvě hlavní funkce:

- Formátování dat z vyšších vrstev do rámců a jejich předávání spojové vrstvě (a obráceně)
- Rozdělování (a spojování) dat po maximálních počtech bajtů pro tvorbu paketů

■ 2.2.4 SMP & GAP

SMP (Security Management Protocol) je soubor bezpečnostních algoritmů umožňující šifrovanou komunikaci mezi jednotlivými uzly sítě, indentifikaci uzlů a skrývání/odkrývání veřejných Bluetooth adres pro ovládání zjistitelnosti cizími zařízeními.

Bezpečnostní postupy používané SMP jsou:

- **Párování**, sloužící ke generování dočasného klíče pro zajištění zašifrované komunikace
- **Spojování** (angl. *bonding*), což je párování následované generováním a výměnou permanentních klíčů
- **Obnovení šifrování** (angl. *encryption re-establishment*) následuje po spojování dvou zařízení, a definuje, jakým způsobem mají být uložené permanentní klíče používány pro obnovení zašifrovaného spojení

Existují tři základní párovací algoritmy, které se liší způsobem generování krátkodobého klíče (angl. *short-term key*, STK) a tak mohou poskytovat ochranu proti útočnickům:

- **Nechráněný postup** (angl. *Just Works*), tedy STK je vygenerován na obou stranách na základě paketů, které jsou veřejně vyměněny, tzn. tento způsob neposkytuje žádnou ochranu pro útokům typu MITM (*man-in-the-middle*).

- **Předání PINu** (angl. *Passkey Display*) je způsobem, jak zabránit MITM útokům. První strana vygeneruje číselný kód, který musí druhá strana zadat, aby došlo ke spárování obou zařízení. Jedná se o nejběžnější postup.
- **Mimopásmový postup** (angl. *Out of Band, OOB*) funguje na principu předávání dodatečných dat jinou technologií než BLE, např. pomocí NFC. Také poskytuje ochranu proti MITM útokům.

GAP (Generic Access Profile) definuje, jakým způsobem spolu mají zařízení komunikovat na úrovni mimo samotný BLE protokol, jedná se o nejvyšší ovládací vrstvu BLE protokolu navazující na SMP. Konkrétně specifikuje, jak zařízení mají provádět postup hledání dostupných zařízení, navazovat zabezpečené spojení, určuje jejich role v topologii celé sítě aj.

■ 2.2.5 ATT & GATT

ATT (Attribution Protocol) je protokol založený na jednoduchém konceptu klient/server, kde **klient** je zařízení, které si vyžádává data od serveru a **server** poskytuje tato data klientům. V BLE může být každé zařízení klientem, serverem nebo obojím zároveň. Tento protokol je deterministický, vždy je možné odeslat pouze jeden požadavek na server/klienta současně, a další odeslat až po obdržení odpovědi.

Každý server obsahuje data v podobě **atributů**. Každý atribut má vlastní 16bitové madlo (angl. *handle*), univerzálně jedinečný identifikátor (angl. *universally unique identifier, UUID*), přístupová práva (angl. *permissions*) a hodnotu. UUID specifikuje typ a povahu hodnoty.

Pokud chce klient číst nebo zapisovat atributy serveru, zašle serveru požadavek k zapisování či čtení včetně madla. Server odpovídá klientovi buď přímo hodnotou, nebo oznámením (angl. *acknowledgement*).

GATT (Generic Attribute Profile) vychází z ATT a přidává hierarchické uspořádání. Definuje, jak jsou mezi aplikacemi data organizována a vyměňována.

Součástí hierarchie jsou **obecné datové objekty**, které mohou být používány velkou škálou aplikačních profilů (které se nazývají angl. *GATT-based profiles*). Data jsou součástí tzv. **služeb** (angl. *services*), které se skládají z jedné nebo více **charakteristik** (angl. *characteristics*). Každou charakteristiku lze považovat za sjednocení uživatelských dat a metadat (obsahující informace popisující hodnoty uložené v datech, např. viditelný název, jednotky aj.).

Charakteristiky lze také chápat jako komunikační kanály v každé BLE aplikaci. V každém tomto kanálu můžeme měnit vysílanou hodnotu (tj. hodnotu charakteristiky), povolovat a zakazovat některým účastníkům komunikace čtení nebo zápis této hodnoty na základě nastavení dané služby pro danou charakteristiku, celý kanál šifrovat aj. Jeden z možných aplikačních profilů

- **Náhodná adresa** (angl. *random address*) může být předprogramována v každém Bluetooth zařízení, nebo může být dynamicky generována.

Toto základní rozdělení adres dále GAP upravuje na tři různé kategorie.

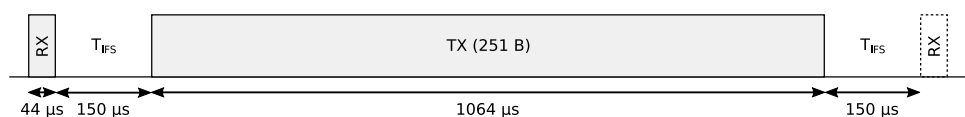
- **Statická adresa** (angl. *static address*) může sloužit k nahrazení veřejné adresy zařízení. Může být náhodně vygenerovaná při každém startu zařízení anebo fixní po celou dobu.
- **Nejistitelná soukromá adresa** (angl. *non-resolvable private address*) není běžně používána, je generována náhodně a obvykle je dočasná.
- **Zjistitelná soukromá adresa** (angl. *resolvable private address*) je základním kamenem zabezpečení komunikace v BLE. Je generována z klíče pro zjišťování identity (angl. *identity resolving key*, IRK) a náhodného čísla, a může se měnit mnohokrát za běhu zařízení, aby nemohlo být zařízení sledováno a identifikováno cizím zařízením.

2.4 Základní omezení

2.4.1 Nejvyšší datová propustnost

Ve specifikaci BLE v5.0 je uvedena maximální modulační rychlost 2 Mb/s. Tato hodnota je v praxi nižší, kvůli vlivu obousměrné komunikace, omezením procesoru a vysílače, i omezením v implementaci software pro řídicí mikroprocesor.

Na obrázku 2.2 je naznačeno časování střídání komunikace mezi dvěma uzly. Pro zjištění nejvyšší datové propustnosti vysílače je část pro přijímání dat nejmenší možná, a část pro vysílání dat největší možná. Vysílání každého z obou uzlů je odděleno mezirámcovou mezerou T_{IFS} (*inter-frame spacing*), která je stálá a rovna 150 μs .



Obrázek 2.2: Časování datového přenosu v BLE v5.0

Přijímaný paket (RX) je prázdný, tedy neobsahuje žádné datové bajty ani MIC (vizte strukturu paketu 2.2). Proto je doba nutná pro přijetí takového paketu rovna

$$T_{RX} = \frac{8 \cdot N_B(RX)}{MR} = \frac{8 \cdot (2 + 4 + 2 + 3)}{2 \cdot 10^6} = 44 \mu s \quad (2.1)$$

kde MR značí modulační rychlost.

Vysílaný paket (TX) obsahuje nejvyšší možný počet datových bajtů (251 B) včetně MIC (4 B). Doba nutná pro vysílání takového paketu je rovna

$$T_{TX} = \frac{8 \cdot N_B(TX)}{MR} = \frac{8 \cdot (2 + 4 + 2 + 251 + 4 + 3)}{2 \cdot 10^6} = 1064 \mu s \quad (2.2)$$

Jelikož v jediném paketu může být maximálně 251 datových bajtů, můžeme maximální datovou propustnost (*data rate*, DR) spočítat jako

$$\begin{aligned} DR &= \frac{8 \cdot N_B(\text{data})}{T_{RX} + T_{IFS} + T_{TX} + T_{IFS}} \\ &= \frac{8 \cdot 251 \cdot 10^6}{44 + 150 + 1064 + 150} \approx 1426 \text{ kb/s} \end{aligned} \quad (2.3)$$

Hodnota datové propustnosti se zvýšila 1.7krát od předchozí specifikace BLE v4.2.

2.4.2 Maximální dosah

Se specifikací BLE v5.0 přišlo zvýšení maximálního vysílacího výkonu z +10 dBm na +20 dBm, minimální výkon zůstal -20 dBm (zařízení nemohou být neviditelná pro ostatní blízka zařízení).

Maximální dosah určuje útlum volného prostoru (angl. *free-space path loss*, FSPL), neboli jak moc se sníží výkon rádiového signálu mezi anténami vysílače a přijímače. Aproximací útlumu pro BLE zařízení využívající běžné izotropní antény, s předpokladem ideálních vysílacích/přijímacích systémů (bez vnitřních ztrát), je rovnice

$$FSPL = 40 + 25 \log_{10}(d) \quad (2.4)$$

kde d je vzdálenost mezi vysílačem a přijímačem v metrech, jednotky útlumu jsou dB. [13]

Útlumům od 50 dB do 110 dB odpovídají tyto vzdálenosti:

Útlum [dB]	Vzdálenost [m]
50	2.5
60	6.3
70	16
80	40
90	100
100	250
110	630

Tabulka 2.3: Vzdálenosti od vysílače s útlumem od 50 do 110 dB

Pokud je známa závislost útlumu na vzdálenosti, lze při požadavku na dosah jednoduše určit požadovaný vysílací výkon a citlivost přijímače. Např. pro vysílací výkon -20 dBm, citlivost přijímače 70 dBm a útlum 50 dB je dosah 2.5 metru. Naopak pro vysílací výkon $+10$ dBm, citlivost přijímače 90 dBm a útlum 100 dB je dosah 250 metrů.

Kapitola 3

Zapojení pro rukavice

3.1 Požadavky

Zde jsem shrnul základní požadavky na fungování rukavic:

- Co nejnižší proudová spotřeba s možností napájení z Li-Po akumulátoru
- Možnost komunikace s PC přes USB kabel (např. UART rozhraní)
- Sběr dat o pohybu prstů i celé ruky s dostatečnou přesností a rychlostí
- Zasílání dat v reálném čase do připojeného zařízení dle standardu BLE
- Malé a dostatečně flexibilní plošné spoje

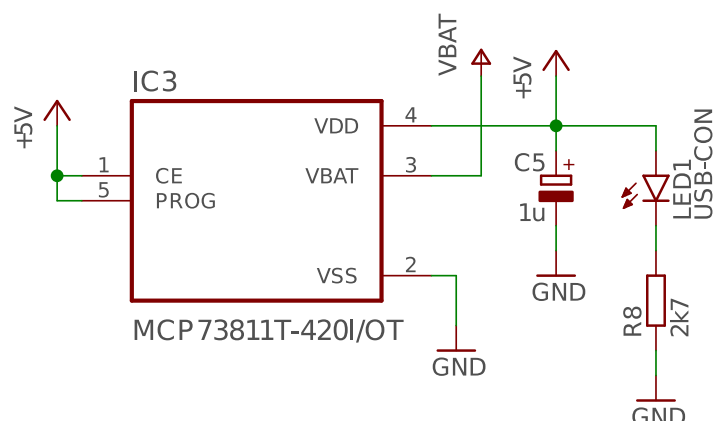
3.2 Napájecí obvod

Pro napájení celého obvodu jsem vybral jednočlánekový Li-Po akumulátor s nominální hodnotou napětí 3.7 V a kapacitou 700 mAh.



Obrázek 3.1: Vybraný Li-Po akumulátor

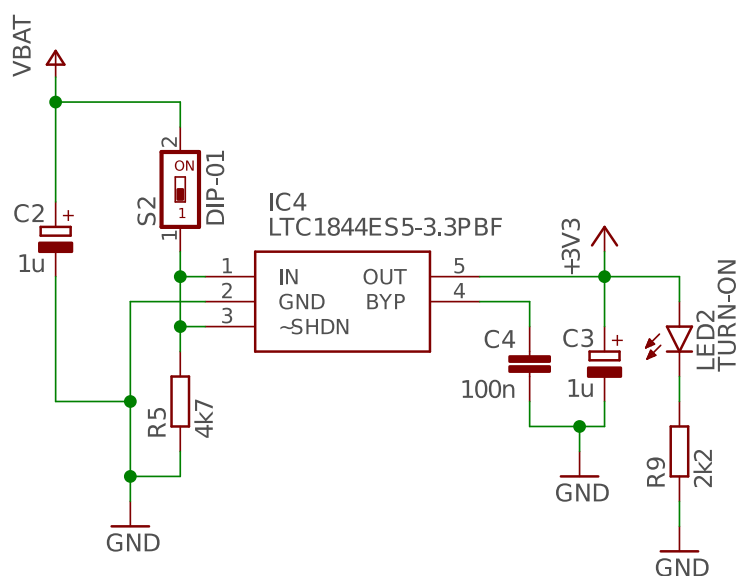
Vybral jsem integrovaný obvod MCP73811 od firmy Microchip, který umožňuje nabíjení akumulátoru přímo z 5V linky USB konektoru, přičemž omezuje nabíjecí proud na nejvýše 450 mA a výstupní napětí na 4 V. [14]



Obrázek 3.2: Zapojení nabíjecího obvodu MCP73811

K napájecímu pinu VDD je připojena LED dioda pro indikaci připojení USB konektoru.

Pro napájení většiny integrovaných obvodů je nutné stabilizované napětí 3.3 V. To zaručí napěťový regulátor (LDO) LTC1844 od firmy Analog Devices. Má velmi nízký úbytek napětí, typicky 90 mV při odběru 150 mA. Akumulátor tedy může být téměř na spodní hranici bezpečné hodnoty napětí, zhruba 3.4 V, a rukavice stále budou mít k dispozici potřebné napájecí napětí. [15]



Obrázek 3.3: Zapojení napěťového regulátoru LTC1844

Pro možnost vypnutí přívodu napájení do rukavic je na nich umístěn dvoupolohový vypínač S2. Dále je zde umístěna LED dioda pro indikaci

přívodu napájení do rukavic.

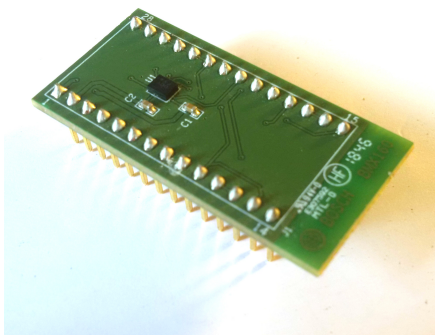
Oba výše zmíněné integrované obvody jsou zapojeny do série, aby po připojení USB konektoru bylo možné akumulátor zároveň nabíjet, a zároveň aby poskytoval napětí pro fungování celých rukavic. Rukavice lze také napájet pouze z USB konektoru při odpojení akumulátoru.

3.3 Senzory

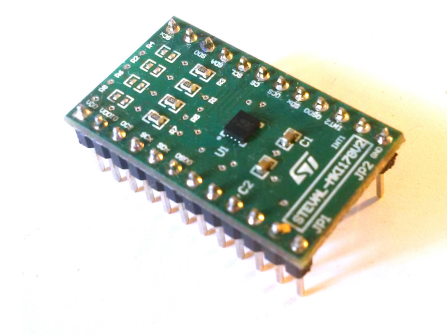
3.3.1 Pohybové senzory

Pro snímání pohybu bylo nutné vybrat takový senzor, který bude poskytovat informace o ohybu článků prstů a pohybu celé ruky s dostatečnou přesností a bez nutnosti opakované kalibrace. Zároveň byla požadována co nejmenší proudová spotřeba, jednoduchá komunikace se senzorem (ideálně I2C), malé pouzdro, a také hrála roli cena senzoru.

Otestoval jsem dva senzory od dvou různých výrobců: Bosch BMX160 a ST LSM6DSL, s využitím oficiálních zkušebních desek od obou výrobců.



(a) : BMX160 Shuttle Board



(b) : STEVAL-MKI178V2 (LSM6DSL)

Obrázek 3.4: Zkušební desky obou pohybových senzorů

Senzor	BMX160 [16]	LSM6DSL [17]
Výrobce	Bosch	STMicroelectronics
Počet DOF	9	6
Napětí	1.7 až 3.6 V	1.7 až 3.6 V
Spotřeba (typ.)	1.6 mA	0.65 mA
Rozlišení dat	16 bitů	16 bitů
Komunikace	I2C, SPI	I2C, SPI
Pouzdro	LGA-14	LGA-14
Cena (orientační, vč. DPH)	160 Kč	95 Kč

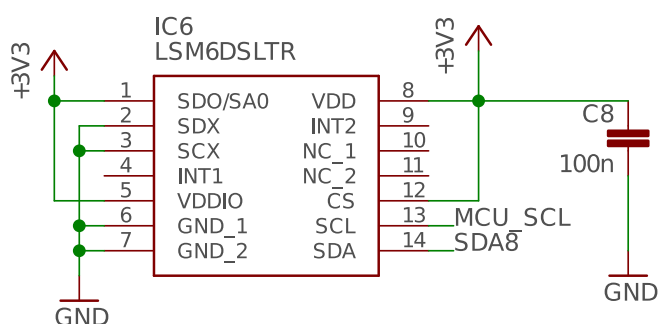
Tabulka 3.1: Srovnání parametrů pohybových senzorů

Pro tuto aplikaci jsem vybral senzor LSM6DSL od ST. Má přibližně o 1 mA nižší spotřebu než jeho konkurent od Bosch. Také postrádá magnetometr (3 stupně volnosti), který by mohl v blízkosti permanentních magnetů mohl negativně ovlivňovat měření.

Zapojení senzoru vychází přímo z oficiální dokumentace od ST [17]. Piny VDD i VDDIO jsou připojeny k napájecímu napětí +3.3 V společně s blokovacím kondenzátorem o kapacitě 100 nF, který je umístěn co nejbližší napájecímu pinu VDD. Na pin CS je přivedeno napájecí napětí pro volbu komunikace I2C.

Senzor LSM6DSL může fungovat ve 2 módech, v prvním módu funguje pouze jako slave, v druhém podporuje funkci master, tedy lze k němu připojit další senzory. V módu 1 se musí připojit piny SDX a SCX buď k napájecímu napětí, nebo k zemi.

Připojením pinu SDO/SA0 k napájecímu napětí jsem vybral pro komunikaci přes I2C nejnižší bit adresy jako 1.



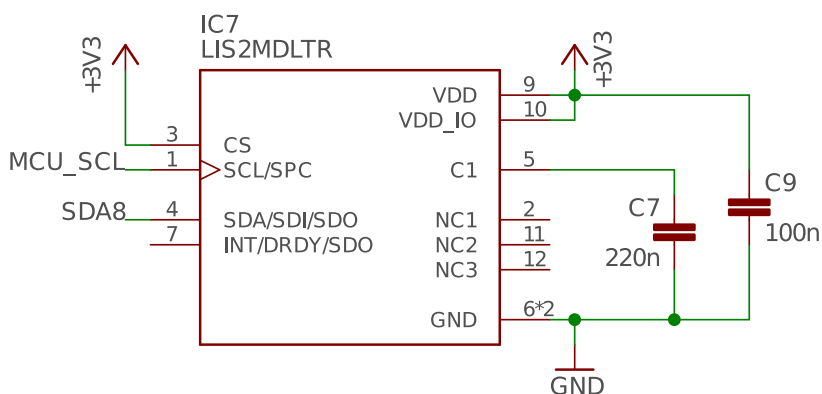
Obrázek 3.5: Zapojení senzoru LSM6DSL umístěného na hřbetu ruky

3.3.2 Magnetometr

Pohybový senzor LSM6DSL umístěný na hřbetu ruky byl doplněn o 3osý magnetometr LIS2MDL od výrobce ST. Za pomoci dat z magnetometru je

pak možné určit absolutní orientaci ruky v prostoru.

Tento senzor měří sílu magnetického pole ve všech 3 osách v rozsahu od -50 do 50 gauss s přesností 1.5 mgauss/bit a rozlišením 16 bitů. Pro komunikaci je zvolena sběrnice I2C. [18]

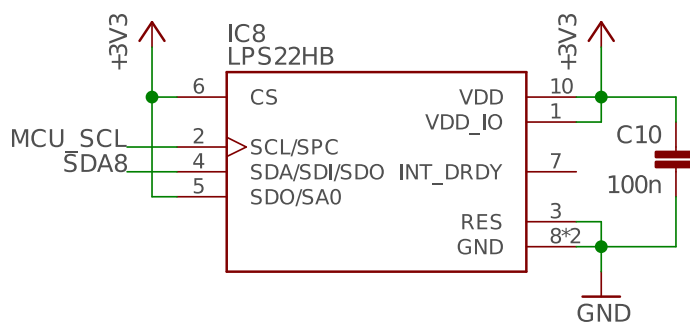


Obrázek 3.6: Zapojení senzoru LIS2MDL umístěného na hřbetu ruky

3.3.3 Barometrický senzor

Pro doplnění pohybových údajů o informace týkající se okolí (teploty a tlaku) jsem vybral absolutní barometrický senzor LPS22HB od firmy ST.

Poskytuje informace o okolním tlaku vzduchu v rozsahu 260 až 1260 hPa s citlivostí 4096 bitů/hPa s přesností ± 1 hPa (bez bodové kalibrace) a rozlišením 24 bitů. Zároveň měří okolní teplotu v rozsahu 0 až 65 °C s rozlišením 16 bitů. Data poskytuje s frekvencí volitelnou od 1 do 75 Hz, komunikace probíhá přes sběrnici I2C. Odběr proudu je zaručen pod 3 μ A. [19]



Obrázek 3.7: Zapojení senzoru LPS22HB umístěného na hřbetu ruky

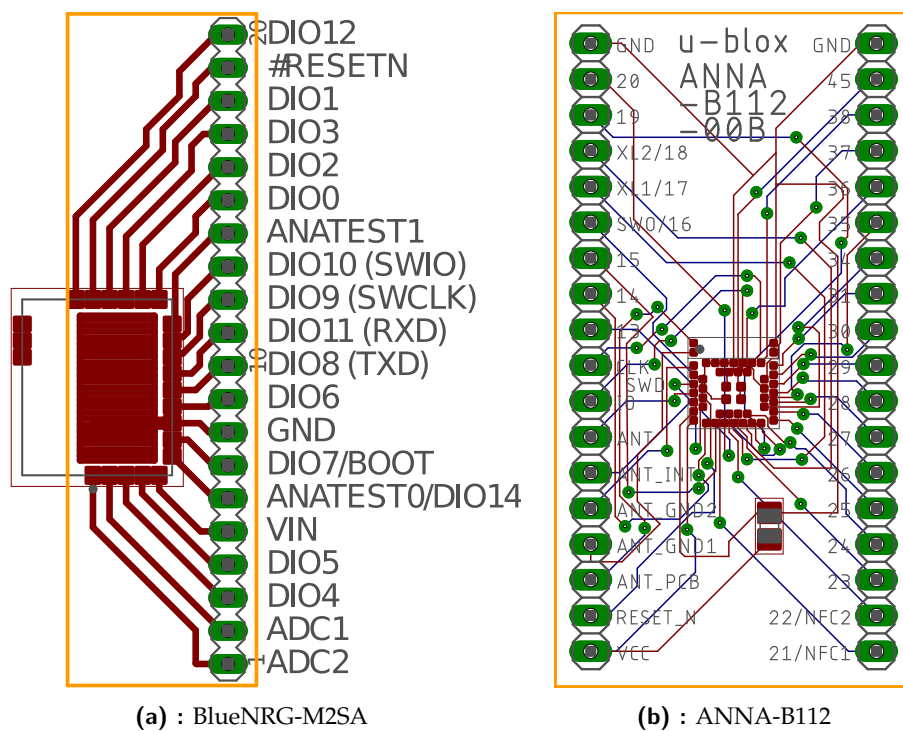
3.4 Bluetooth modul

Otestoval jsem dva programovatelné Bluetooth moduly, ST BlueNRG-M2SA a u-blox ANNA-B112-00B. Základním parametrem jejich výběru byl požadavek na přítomnost integrované Bluetooth antény.

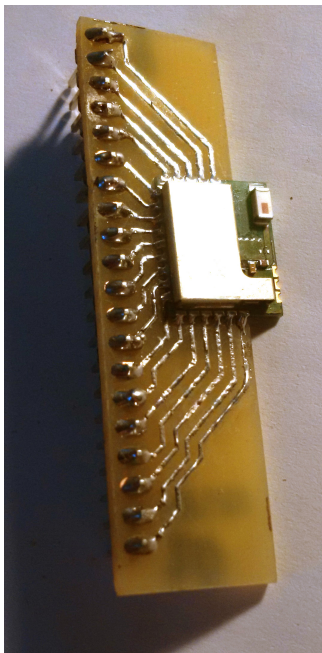
Modul	BlueNRG-M2SA [20]	ANNA-B112-00B [21]
Výrobce	STMicroelectronics	u-blox
Napětí	1.7 až 3.6 V	1.7 až 3.6 V
Spotřeba (průměr)	2.75 mA	1.8 mA
Obsažený mikroprocesor	ST BlueNRG-2	Nordic Semi nRF52832
Jádro ARM	Cortex-M0	Cortex-M4
Paměť RAM	24 kB	64 kB
Paměť Flash	256 kB	512 kB
Počet GPIO	14	25
Celkový počet pinů	23	52
Programátor (SWD)	ST-Link	J-Link
Rozměry	13.5 x 11.5 x 2 mm	6.5 x 6.5 x 1.2 mm
Cena (orientační, vč. DPH)	220 Kč	270 Kč

Tabulka 3.2: Srovnání parametrů programovatelných modulů

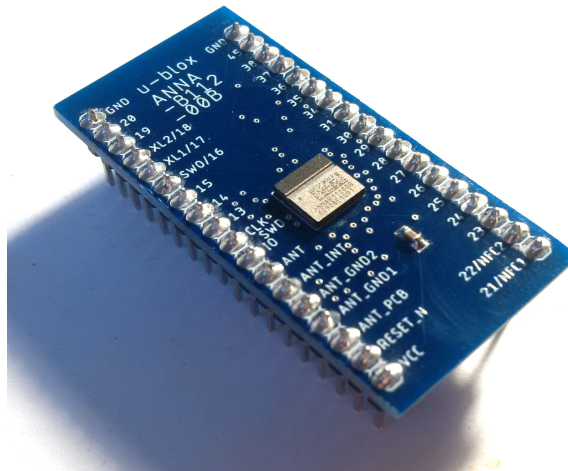
Bohužel ani jeden z výrobců modulů nenabízí hotové zkušební desky, které bych mohl použít pro testování. Z toho důvodu jsem si vyrobil vlastní adaptory do nepájivého pole, které jsem následně osadil moduly.



Obrázek 3.8: Návrhy adaptorů pro komunikační moduly



(a) : BlueNRG-M2SA



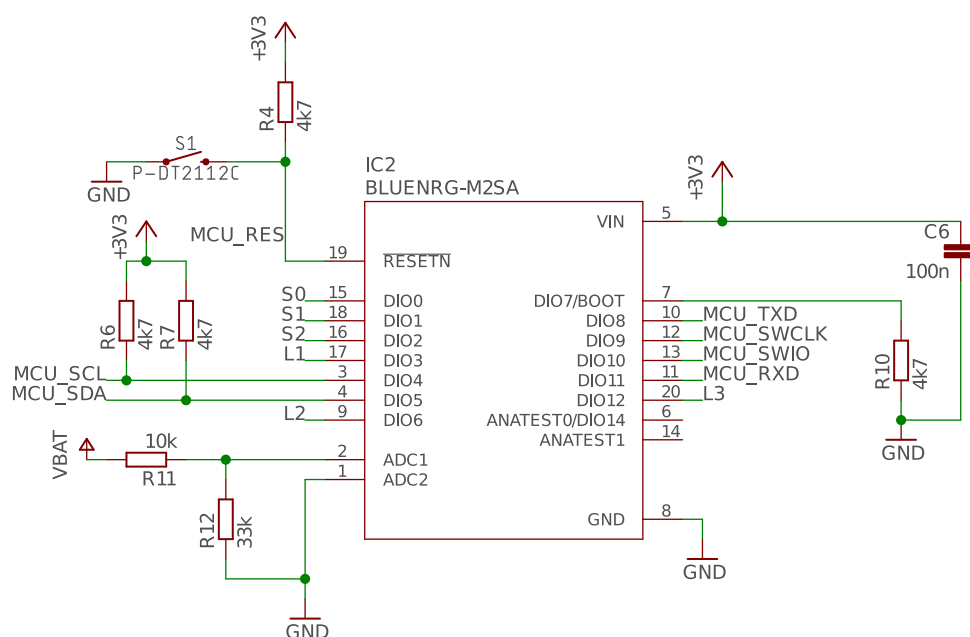
(b) : ANNA-B112

Obrázek 3.9: Osazené adaptory s komunikačními moduly

Přestože modul ANNA-B112 disponuje větším počtem pinů v menším pouzdře s podporou SWM (switch matrix pro přiřazení libovolné periferie libovolnému GPIO), větším výkonem, pamětí a má nižší spotřebu v řádu několika miliampér, rozhodl jsem se vybrat pro rukavice modul BlueNRG-M2SA.

Prvním důvodem bylo pouzdro, které je u modulu od ST jednodušší na zapájení, jelikož má méně vývodů, jejichž počet pro tento projekt dostačující, a zároveň nejsou vývody umístěné zespoda pouzdra, což vyžaduje využití složitějších pájecích technik a použití rentgenu pro ověření výsledku.

Druhým důvodem byla dokumentace a SDK. SDK od ST obsahuje více příkladů, jednodušších na kompilaci, a zdrojový kód řídicího firmwaru tak byl nakonec přehlednější. Dokumentace k obsaženému mikroprocesoru od ST je zároveň přehlednější, než dokumentace od firmy Nordic Semiconductor.



Obrázek 3.10: Zapojení Bluetooth modulu BlueNRG-M2SA

Modul je napájen z 3.3 V. Kondenzátor C6 slouží jako blokovací a má kapacitu 100 nF. Z modulu jsou vyvedeny spoje pro programování přes SWD rozhraní a spoje pro komunikaci přes UART s převodníkem USB-UART. Pin DIO7 musí být uzemněný, slouží ke spuštění bootloaderu pro programování modulu přes UART rozhraní, což jsem se rozhodl primárně nevyužívat pro nemožnost debugingu mikroprocesoru.

Na pinech DIO4 a DIO5 se nacházejí piny SCL a SDA z I2C periferie, oba jsou připojeny přes pull-up rezistory R6 a R7 o hodnotách 4.7 kΩ (modul neobsahuje vnitřní pull-up rezistory na výstupu I2C).

Resetovací pin je připojen přes pull-up rezistor R4 k napájecímu napětí a přes mikropínač S1 k zemi.

Pin ADC2 je uzemněný a k pinu ADC1 je přes napěťový dělič připojeno napětí na Li-Po akumulátoru pro sledování jeho stavu. Jelikož se nabije maximálně na 4.0 V, při nabitém akumulátoru výsledné napětí na vstupu AD převodníku bude rovno

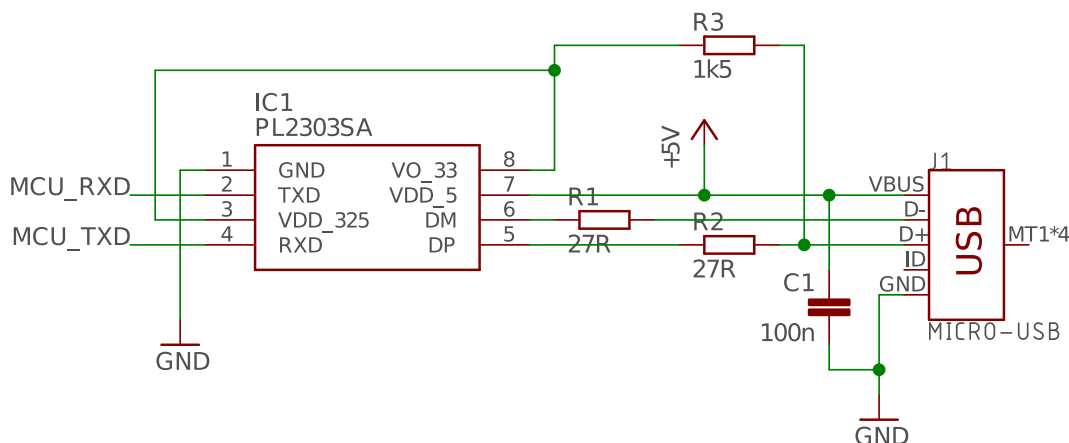
$$U_{ADC \max.} = \frac{R_{12}}{R_{11} + R_{12}} U_{bat \max.} = \frac{33}{33 + 10} \cdot 4.0 \approx 3.07 \text{ V} \quad (3.1)$$

3.5 Komunikace s PC

Pro možnost sledování logovacích zpráv v mikroprocesoru jsem zvolil rozhraní UART. Vybral jsem USB-UART převodník s dostatečně malým pouzdrům, které půjde snadno zapájet do obvodu.

Vybraný převodník Prolific PL2303SA se vyrábí v pouzdře SO-8 a jeho spotřeba je maximálně 25 mA, napájen je ovšem z 5 V po připojení USB ko-

nektoru, jinak je odpojen od napájení. Maximální přenosová rychlost je u něj omezena na 115200 bit/s, což ovšem pro sledování výpisu z programu mikroprocesoru stačí. [22]



Obrázek 3.11: Zapojení USB-UART převodníku PL2303SA

Kondenzátor C1 o hodnotě 100 nF je připojen mezi 5 V z USB a zem jako blokovací, opět co nejbližše integrovanému obvodu.

Datová linka USB D+ je připojena přes pull-up rezistor R3 o hodnotě 1.5 kΩ k napájecímu napětí 3.3 V, aby bylo zařízení správně rozpoznáno počítačem jako zařízení komunikující rychlost 12 Mb/s (full-speed).

Pro dodržení správné celkové impedance datových linek (podle oficiální USB specifikace) je dle aplikačního schématu výrobce obvodu nutné přidat za piny DM a DP rezistory o hodnotě 27 Ω do série.

V konektoru je pin ID ponechán jako plovoucí, jelikož je dle specifikace OTG (On-The-Go) zařízení poté vnímáno jako slave.

3.6 Připojení vícera I2C sběrnic

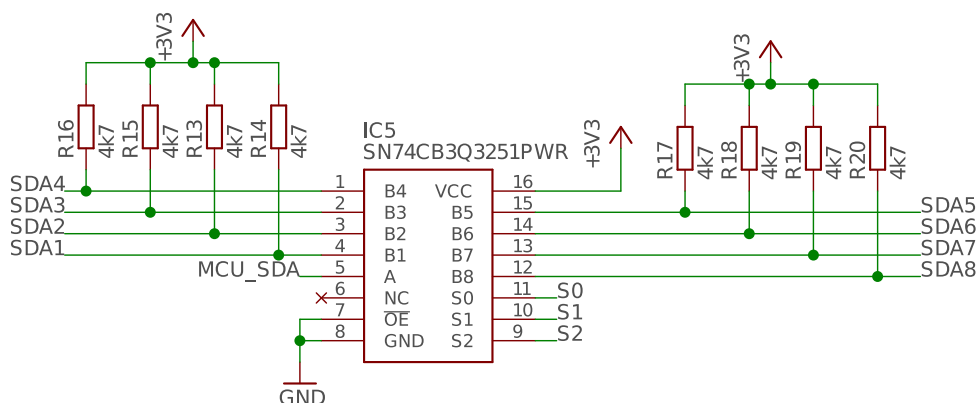
Pohybové senzory LSM6DSL mají omezení, a to volbu pouze posledního bitu jejich adresy pro I2C komunikaci, je tedy možné na jedinou sběrnicu umístit nejvýše dva senzory stejného typu, aniž by došlo ke kolizi.

Pro vyřešení tohoto problému jsou vždy umístěny nejvýše dva senzory na stejné sběrnicě, celkem je mikroprocesorem potřeba obsloužit 8 sběrnic. Linka SCL je pro všechny senzory umístěné na rukavicích společná, jelikož není důvod ji duplikovat. Je zde tedy 8 linek SDA1 až SDA8.

Jednou možností obsloužení takového množství I2C sběrnic s jedinou I2C periferií mikroprocesoru by bylo použít kombinaci timeru v módu Output Compare pro časování pulzů na jediné lince SCL a řízení komunikace přes softwarově implementovaný I2C protokol na 8 různých GPIO pro jednotlivé linky SDA.

Já zvolil alternativu v podobě osmikanálového multiplexoru a zároveň demultiplexoru SN74CB3Q3251 od firmy Texas Instruments. Tento integrovaný

obvod umožňuje vybrat jednu z 8 výstupních linek a připojit ji k jediné vstupní. Napájený je z 3.3 V a funkčnost je zaručena pro signály až do frekvence 500 MHz. Spotřeba obvodu je typicky 1 mA. Dalším nezbytným předpokladem je možnost obousměrné komunikace na vybrané lince, jelikož se na lince SDA střídají ve vzájemné komunikaci mikroprocesor se senzorem, což tento obvod umožňuje. [23]

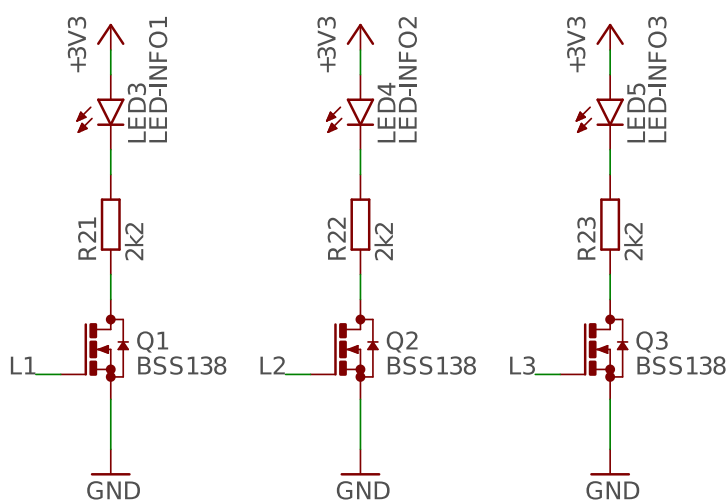


Obrázek 3.12: Zapojení multiplexoru/demultiplexoru

Všechny SDA linky musí být připojeny přes pull-up rezistor o hodnotě 4.7 k Ω k napájecímu napětí. Výběr cílové SDA linky provádí mikroprocesor nastavením logických úrovní na pinech S0, S1 a S2.

3.7 Indikační LED diody

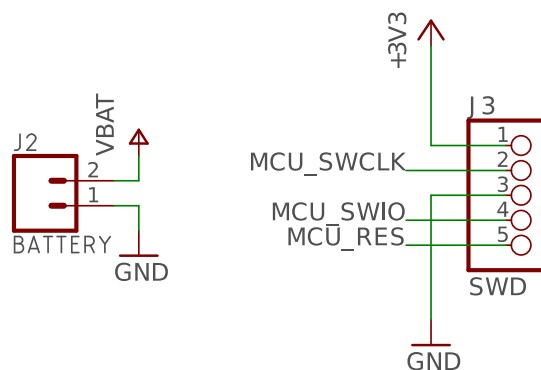
Pro indikaci libovolných stavů v mikroprocesoru jsou k němu připojeny 3 indikační LED diody přes N-kanálové MOSFET tranzistory BSS138.



Obrázek 3.13: Zapojení indikačních LED diod

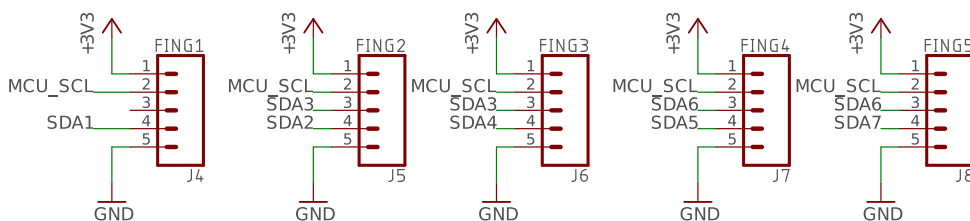
3.8 Konektory

Na rukavicích je umístěn standardní konektor od výrobce JST pro připojení akumulátoru a poté 5kolíková lišta pro připojení programátoru k SWD rozhraní.



Obrázek 3.14: Konektory pro akumulátor a programátor

Dále je zde umístěno 5 pětizdířkových dutinek pro připojení napájení a I2C linek k jednotlivým plošným spojmům nesoucím senzory umístěné nad jednotlivými články prstů.



Obrázek 3.15: Konektory pro plošné spoje na prstech

Kapitola 4

Aplikace pro modul

4.1 Vývojové prostředí

Aplikaci pro Bluetooth modul jsem se rozhodl programovat v prostředí Keil μ Vision 5. Prostředí μ Vision umožňuje vybrat mikroprocesor BlueNRG-2 obsažený ve vybraném modulu z databáze podporovaných zařízení.

4.2 Knihovny

Bohužel Keil pro modul nenabízí potřebné spouštěcí (angl. *startup*) soubory, které správně nastaví všechny nutné části mikroprocesoru pro následující bezchybný běh hlavního programu.

Výrobce modulu ovšem dává volně k dispozici Software Development Kit (SDK) pro tyto Bluetooth moduly, včetně všech potřebných knihoven - především k nastavení BLE stacku, přístupu k paměťově mapovaným perifériím a knihovnu pro šifrování. Součástí SDK je také plno příkladů a kompletních projektů pro μ Vision, ze kterých jsem vycházel při implementaci mé aplikace.

Pro funkčnost importovaných knihoven bylo nutné vhodně projekt nastavit. Pro kompilaci bylo nutné vybrat podporu MicroLIB, která na jednu stranu umožní snížení velikosti kompilovaného kódu pro ARM mikroprocesory i o více než polovinu, na stranu druhou zamezuje použití C++ a tedy i objektově orientovaného programování (OOP). Zároveň bylo nutné použít soubor nastavení (angl. *scatter file*) z SDK pro správné propojení kódů po jeho kompilaci (angl. *linking*), nikoliv automaticky generovaný soubor prostředím.

■ 4.3 Implementace aplikace

■ 4.3.1 Způsob implementace

Celou aplikaci jsem se snažil psát maximálně transparentní a zároveň snadno upravitelnou v případě potřeby, s možností na jednom místě upravovat celé nastavení BLE, např. jednoduše přidávat služby, charakteristiky. Celý koncept služeb a charakteristik (vizte kap. 2) by bylo nejvhodnější implementovat za pomoci metod OOP (především dědičnosti tříd), s vhodnými objekty pro služby, s podobjekty určenými pro jejich charakteristiky. Kvůli kompilaci s knihovnou MicroLIB jsem se však musel spokojit s omezenými možnostmi čistého jazyka C (standardu C99). Pro dodržení upravitelnosti jsem nadefinoval vhodnou inicializační strukturu pro BLE stack. Podobně jsem pro jednotnou obsluhu různých senzorů přes 8 různých SDA linek využil vhodných struktur a konstantních ukazatelů na inicializační funkce a jiné funkce senzorů, sloužící ke čtení měřených hodnot.

■ 4.3.2 Nastavení BLE stacku

Knihovna od výrobce určená pro nastavení BLE stacku obsahuje všechny funkce pro rozhraní HCI definované standardem BLE v5.0, funkce HAL (angl. *Hardware Access Layer*) pro ovládání radiového vysílače a funkce pro nastavení GAP a GATT. Popis všech funkcí této knihovny lze najít v aplikační poznámce AN5270 (určené pro použité BLE rozhraní s označením výrobce STM32WBx5).

Stack se nastavuje za pomoci inicializační struktury `BlueNRG_Stack_Initialization_t` definované v SDK. Je nutné v ní správně a vhodně definovat parametry pro výpočet alokace paměti, vycházející především z počtu všech atributů GATT serveru (např. počtu služeb, charakteristik), dále počtu povolených současných připojení k serveru, velikosti zasílaných hodnot aj.

■ 4.3.3 Použité periferie modulu

Následující periferie Bluetooth modulu jsem se rozhodl používat a implementoval pro ně základní funkce v kódu aplikace za pomoci SDK knihoven. Ostatní periferie jsem nechal odpojené od hodinového signálu, aby zbytečně nerostly energetické nároky celé aplikace.

■ SysTick

Systémový časovač SysTick (jedná se o součást jádra ARM mikroprocesoru) běží na taktovací frekvenci procesoru, tj. zde 32 MHz, a má standardně 24 bitů. Nabízí možnost vhodnou konstantou tuto frekvenci vydělit (angl. *prescaling*) a dostat tak pravidelné asynchronní přerušení procesoru (angl.

interrupt) o žádané frekvenci.

Tento časovač jsem použil pro počítání milisekund, proto taktovací frekvence byla dělena 32000 pro získání frekvence 1 kHz.

■ ADC

Analogově-číslicový převodník umístěný na čipu BlueNRG-2 nabízí 10bitový převod se vzorkovací frekvencí 1 MHz v celém rozsahu napájecího napětí (od 0 V do 3.3 V). Použil jsem ho ke snímání úrovně napětí na připojeném akumulátoru, pro indikaci jeho stavu a případné nutnosti připojení napájecího USB kabelu pro jeho nabití.

Při použití ADC periferie na modulu BlueNRG-2 jsem narazil na problém, že převod napětí nefunguje při použití ve hlavní smyčce programu, pokud je zároveň volána pravidelná aktualizace BLE stacku. Problém jsem vyřešil umístěním převodu do asynchronně volané funkce SysTick časovačem, úroveň napětí se snímá v pravidelném intervalu 1 sekundy.

■ UART

Sériový komunikační protokol UART jsem použil pro komunikaci modulu s PC za pomoci USB-UART převodníku. Protože se jedná pouze o informační (logovací) zprávy o dění uvnitř modulu, pro výběr postačila frekvence 115200 bitů/s, s délkou slova 8 bitů, bez použití paritních bitů.

■ I2C

I2C je nejdůležitějším komunikačním protokolem v této aplikaci, jelikož poskytuje přístup k datům z jednotlivých senzorů. Nejvyšší frekvence I2C podporovaná použitým modulem je 400 kHz (*fast mode*), proto jsem vybral ji.

■ 4.3.4 Přístup k senzorům

■ Inicializace

Všechny senzory jsou seřazeny v pevném pořadí určeném v kódu aplikace a uloženy v poli složeném ze struktur typu `sensor_t`, který jsem definoval jako:

```

1 typedef uint8_t (*fn_get_t)(uint8_t*);
2 typedef uint8_t (*fn_init_t)(void);
3
4 typedef struct {
5     fn_init_t* fn_init;
6     fn_get_t (*fn_get_raw)[FN_RAW_NUM];
7     fn_get_t (*fn_get_sens)[FN_SENS_NUM];
8 } fn_pointers_t;
9
```

```

10 typedef struct {
11     uint8_t i2c_bus;
12     uint8_t sensor_addr;
13     fn_pointers_t* fn_pointers;
14     uint32_t private_data [PRIVATE_DATA_4BS];
15 } sensor_t;

```

Informace o každém ze sensorů je složena z čísla SDA linky, na kterou je připojen, jeho adresy na této sběrnici, ukazatelů na funkce (inicializační, pro získání dat a pro získání nastavení senzoru - citlivosti) a části paměti určené každému senzoru, kde uchovává zpravidla informace o tom, jestli je spuštěné měření (*enabled*) a o jeho poslední zvolené rychlosti datového toku (*output data rate*, ODR).

Abych nemusel definovat všechny funkce pro zápis a čtení hodnot do a z registrů jednotlivých typů sensorů, použil jsem předem implementovaný kód veřejně dostupných knihoven **stm32duino** a odstranil objektovost (jelikož jsou napsány částečně v C++).

Z těchto knihoven jsou k dispozici inicializační funkce všech použitých sensorů, funkce pro čtení a zápis nastavení parametrů sensorů (citlivost, ODR aj.) i pro převod dat z registrů do cílových datových typů, případně jejich přepočítání ze „syrových“ (angl. *raw*) 16bitových údajů na správné údaje (32bitová reálná čísla) za využití citlivosti senzoru.

Pro každý typ senzoru jsem definoval neměnné pole ukazatelů na správné funkce. Tímto způsobem lze pro každý typ senzoru jednoduše přidávat či odebírat požadované funkce, které budou dále volány, navíc je pro všechny typy sensorů definována jednotná struktura.

■ Pořadí sensorů

V hlavním souboru aplikace je umístěno pole pro uvedení jednotlivých sensorů, které vypadá následovně:

```

1 static sensor_t ALL_SENSORS[SENSORS_NUM] = {
2     // Sensor #00, SDA1, LSM6DSL, address LOW, thumb, proximal
3     // phalanx
4     { SDA1, LSM6DSL_ACC_GYRO_I2C_ADDRESS_LOW, (fn_pointers_t*) &
5     fn_pointers_lsm6dsl },
6     ...
7 };

```

Výsledné pořadí sensorů je shrnuto v následující tabulce:

#	SDA linka	Adresa	Typ	Úmístění
0	SDA1	0xD4 (LOW)	LSM6DSL	palec, nejbližší čl.
1	SDA1	0xD6 (HIGH)	LSM6DSL	palec, špička prstu
2	SDA2	0xD4 (LOW)	LSM6DSL	ukazováček, střední čl.
3	SDA2	0xD6 (HIGH)	LSM6DSL	ukazováček, špička prstu
4	SDA3	0xD6 (HIGH)	LSM6DSL	ukazováček, nejbližší čl.
5	SDA3	0xD4 (LOW)	LSM6DSL	prostředníček, nejbližší čl.
6	SDA4	0xD4 (LOW)	LSM6DSL	prostředníček, střední čl.
7	SDA4	0xD6 (HIGH)	LSM6DSL	prostředníček, špička prstu
8	SDA5	0xD4 (LOW)	LSM6DSL	prsteníček, střední čl.
9	SDA5	0xD6 (HIGH)	LSM6DSL	prsteníček, špička prstu
10	SDA6	0xD6 (HIGH)	LSM6DSL	prsteníček, nejbližší čl.
11	SDA6	0xD4 (LOW)	LSM6DSL	malíček, nejbližší čl.
12	SDA7	0xD4 (LOW)	LSM6DSL	malíček, střední čl.
13	SDA7	0xD6 (HIGH)	LSM6DSL	malíček, špička prstu
14	SDA8	0xD6 (HIGH)	LSM6DSL	hřbet ruky
15	SDA8	0xBA (HIGH)	LPS22HB	hřbet ruky
16	SDA8	0x3C	LIS2MDL	hřbet ruky

Tabulka 4.1: Pořadí senzorů dle přístupu

V tomto stejném pořadí jsou v hlavní smyčce při inicializaci i sbírání dat ze senzorů volány odpovídající funkce po zvolení odpovídající SDA linky.

4.3.5 Funkce indikačních LED diod

Pro indikaci jsem využil pouze led diody LED3 a LED4.

Dioda LED3 (žlutá) svítí po dobu inicializace celé aplikace modulu, a zhasne po přechodu do hlavní smyčky programu. Při chybě v inicializaci spojené obvykle se špatnou paměťovou alokací dojde k resetu celé aplikace a dioda zůstane blikat, což umožňuje snažší hledání chyby v programu.

Dioda LED4 (červená) indikuje svým přerušováním blikáním chybový stav. Smyčku, která mění logický stav na pinu odpovídajícím pinu, jsem umístil mimo hlavní smyčku programu. Do tohoto stavu se program dostane při chybě při inicializaci senzorů nebo při komunikaci s nimi. Jedná se proto nejčastěji o chyby spojené se sběrnici I2C (chyba na spojích mezi modulem, senzorem či multiplexorem).

■ 4.3.6 Nastavení GAP & GATT

■ GAP

Přístupový profil této aplikace jsem nastavil tak, že je zařízení veřejně viditelné pod názvem „Glove Left“ v případě rukavice pro levou ruku, podobně „Glove Right“ pro ruku pravou. Veřejná adresa zařízení je generována náhodně při spuštění zařízení.

Zabezpečení je nastaveno na spojování klienta se serverem (*bonding*) s fixním pinem pro párování nastaveným defaultně na 123456. Nicméně pro přístup k hodnotám charakteristiky není nutné spárovat obě zařízení.

■ Charakteristika pro získání dat

Aby mohla klientská aplikace (vizualizační aplikace běžící na počítači) přijímat data, nastavil jsem pro tento účel charakteristiku, která rozesílá pravidelně všem připojeným klientům datové pakety pomocí notifikací. Čtení hodnoty charakteristiky na žádost klienta zde nemá význam, docházelo by pouze ke snížení rychlosti běhu aplikace v modulu.

Univerzální identifikátor (UUID) této charakteristiky jsem nastavil na hodnotu

1bc5d5a5-0200-36ac-e111-010000000000

■ Charakteristika pro získání citlivosti

Pro přepočítání hodnot naměřených hodnot akcelerometrem, gyroskopem a magnetometrem na odpovídající jednotky (mg, mdps a mGauss ve stejném pořadí) je nutné znát nastavenou citlivost senzorů (hodnoty v odpovídajících jednotkách příslušící nejméně významnému bitu (LSB)).

Proto jsem nadefinoval ještě druhou charakteristiku pod stejnou službou, která slouží pro získání počáteční konfigurace senzorů klientskou aplikací. Tato charakteristika má povolený způsob přístupu pouze čtením.

Univerzální identifikátor (UUID) této charakteristiky jsem nastavil na hodnotu

1bc5d5a5-0200-56bc-e111-010000000000

■ 4.3.7 Struktura paketů

■ Datové pakety

Senzory typu LSM6DSL, kterých je osazeno celkem 15, poskytují informace z akcelerometru a gyroskopu po trojicích (v osách x, y, z), každá z celkem 6 těchto hodnot je tvořena 16bitovým číslem se znaménkem. Z každého tohoto senzoru je nutné proto odeslat 12 bajtů.

Podobně senzor typu LIS2MDL poskytuje trojici hodnot pro měření z magnetometru stejného datového typu, tj. celkem 6 bajtů je nutno odesílat.

Nakonec senzor typu LPS22HB poskytuje přepočítané hodnoty o teplotě a tlaku, obojí ve formátu reálného čísla (32 bitů), proto je nutno odesílat 8 bajtů.

Nakonec čas (v milisekundách) je posílán jako 32bitové číslo bez znaménka a hodnota napětí baterie je přepočítávána a posílána jako reálné číslo (32 bitů). To znamená 8 bajtů pro obě tyto hodnoty.

Počet bajtů nutných k pravidelnému odesílání je proto

$$N_{B(TX)} = 15 \cdot 12 + 6 + 8 + 8 = 202 \quad (4.1)$$

Data ze senzorů jsou seřazena dle tabulky 4.1. Po přidání ostatních dat má datový paket následující strukturu:

Čas [ms]	Napětí [V]	Data LSM6DSL	Data LPS22HB	Data LIS2MD
4 B	4 B	15 · 6 B	8 B	6 B

Tabulka 4.2: Struktura datového paketu

Počet bajtů se vejde do limitu počtu bajtů, které lze odeslat v jediném fyzickém paketu BLE komunikace, což zaručuje konzistenci dat mezi jednotlivými odeslanými pakety.

■ Konfigurační pakety

Citlivosti jednotlivých senzorů typu LSM6DSL jsou získány po dvojicích pro akcelerometr a gyroskop a ze senzoru typu LIS2MDL pro magnetometr. Každá citlivost je získána jako reálné číslo (32 bitů). Aby klientská aplikace mohla snadno určit, která citlivost přísluší kterému senzoru a které jeho části, je před každou čtveřicí bajtů reprezentující danou citlivost přidán bajt obsahující ASCII znak A, G nebo M (podle příslušnosti k akcelerometru, gyroskopu nebo magnetometru). Zároveň je před každou skupinou těchto bajtů uvedeno ID senzoru dle tabulky 4.1, kterému hodnoty přísluší.

■ 4.4 Nahrávání programu do modulu

Program jsem do modulu nahrával za pomoci SWD rozhraní a programátoru ST-Link (umístěném na STM32F4-Discovery Kitu). Postup použití ST-Link na kitu je popsán v uživatelském manuálu UM1472. Jednotlivé piny umístěné na debugovacím konektoru CN2 na kitu jsou:

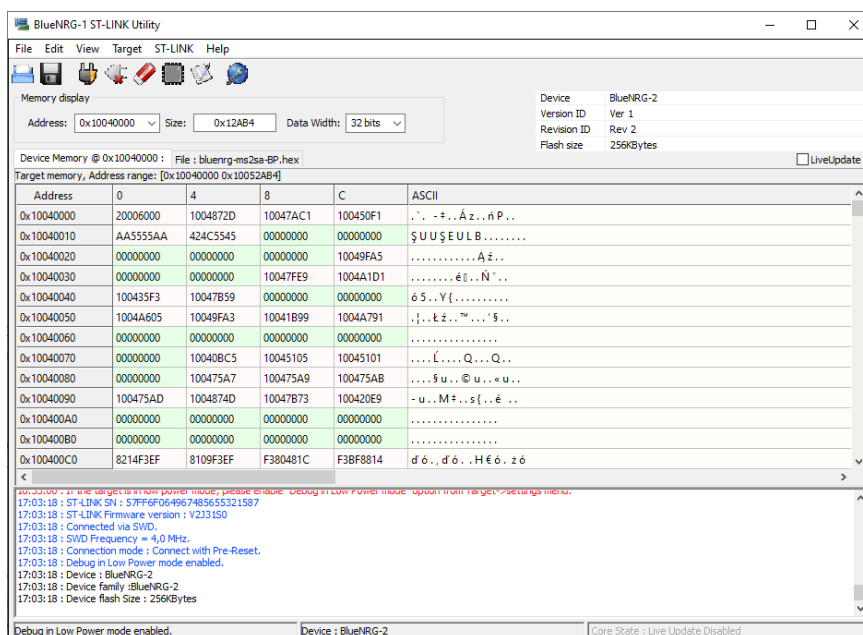
Číslo pinu	Označení	Účel
1	VDD_TARGET	Napájení cílového MCU (3.3 V)
2	SWCLK	Hodiny SWD
3	GND	Zem
4	SWDIO	Datový vstup a výstup SWD
5	NRST	Resetovací signál cílového MCU
6	SWO	(Nepoužívaný)

Tabulka 4.3: Popis debugovacího konektoru CN2 na kitu STM32F4-Discovery

Pro přímočarost jsou tyto piny vyvedeny ve stejném pořadí i na konektoru J3 umístěném na rukavici (vizte podkapitulu 3.8).

Nepodařilo se mi ovšem nastavit prostředí μ Vision tak, abych mohl program nahrát přímo z něj, a pak využívat jeho funkce na debugování programu.

K tomuto účelu výrobce modulu uveřejnil jako součást použitého SDK aplikaci BlueNRG-1 ST-Link Utility, ve šlo bez problémů číst i zapisovat paměť modulu. Vybrané nastavení programátoru ST-Link lze vidět na snímku obrázky 4.1.



Obrázek 4.1: Nahrávání programu v aplikaci BlueNRG-1 ST-Link Utility

Kapitola 5

Zpracování a vizualizace dat

5.1 Orientace a rotace

Velmi důležité je rozlišovat mezi pojmy orientace a rotace objektu. **Orientace** může být definována jako neměnný stav, popisující konečné natočení objektu vůči všem třem osám, ale neříkající nic o tom, jak se do tohoto stavu objekt dostal. **Rotace** je posloupnost všech orientací, kterými se objekt do konečné orientace dostal.

5.2 Použití kvaternionů pro popis orientace

Kvaterniony jsou rozšířením komplexních čísel o další dvě imaginární osy. Jedná se tedy o čtveřici reálných čísel doplněných příslušnými imaginárními jednotkami. Pro pochopení dalšího textu je zde uvedeno několik základních definic a tvrzení týkajících se počítání s kvaterniony.

Prostor \mathbb{H} všech kvaternionů definujeme jako

$$\mathbb{H} = \{q \mid q = q_0 + q_1i + q_2j + q_3k, q_0, q_1, q_2, q_3 \in \mathbb{R}, i^2 = j^2 = k^2 = ijk = -1\} \quad (5.1)$$

kde i, j, k jsou imaginární jednotky. Číslo q_0 (též s) se nazývá **skalární (reálnou) část** kvaternionu a trojice q_1, q_2, q_3 (též \vec{v}) tvoří jeho **vektorovou (imaginární) část**.

Součet kvaternionů q_1 a q_2 definujeme jako

$$q_1 + q_2 = (s_1, \vec{v}_1) + (s_2, \vec{v}_2) = (s_1 + s_2, \vec{v}_1 + \vec{v}_2) \quad (5.2)$$

Součin kvaternionů q_1 a q_2 definujeme jako

$$q_1 \cdot q_2 = (s_1, \vec{v}_1) \cdot (s_2, \vec{v}_2) = (s_1s_2 - \vec{v}_1\vec{v}_2, \vec{v}_1 \times \vec{v}_2 + s_1\vec{v}_2 + s_2\vec{v}_1) \quad (5.3)$$

Součin kvaternionů od prvního pohledu není komutativní (obsahuje vektorový součin), tedy obecně $q_1q_2 \neq q_2q_1$.

Důsledkem je možnost výpočtu vektorového součinu vektorů \vec{v}_1, \vec{v}_2 jako vektorovou část výsledného součinu libovolných dvou kvaternionů q_1, q_2 s nulovými skalárními částmi a vektorovými částmi odpovídajícími vektorům \vec{v}_1, \vec{v}_2 :

$$\vec{v}_1 \times \vec{v}_2 = \text{Vec}(q_1 \cdot q_2) \quad (5.4)$$

Norma kvaternionu q je definována jako

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = \sqrt{qq^*} \quad (5.5)$$

Rozlišení některých typů kvaternionů:

- **Skalární kvaternion** má vektorovou část nulovou, tj. $q = (s, \mathbf{o})$.
- **Ryzí kvaternion** má skalární část nulovou, tj. $q = (0, \vec{v})$.
- **Opačný kvaternion** q_1 ke kvaternionu q_2 je takový, pro který platí $q_1 = -q_2$, z čehož plyne $s_1 = -s_2$ a zároveň $\vec{v}_1 = -\vec{v}_2$.
- **Jednotkový kvaternion** je takový, pro který platí $\|q\| = 1$.
- **Konjugovaný kvaternion** je takový, pro který platí $q^* = (s, \vec{v})^* = (s, -\vec{v})$.
- **Inverzní kvaternion** q^{-1} ke kvaternionu q je definován jako $q^{-1} = \frac{q^*}{\|q\|^2}$.

Kvaternion orientace (rotace) je jednotkový kvaternion, pro který platí

$$q = \left(\cos\left(\frac{\alpha}{2}\right), \vec{n} \sin\left(\frac{\alpha}{2}\right) \right) \quad (5.6)$$

a udává rotaci o úhel α v kladném směru okolo osy rotace dané jednotkovým vektorem \vec{n} , protože rotací danou kvaternionem q ryzího kvaternionu $v = (0, \vec{v}_v)$ je ryzí kvaternion $w = (0, \vec{v}_w)$, pro který platí

$$w = qvq^* \quad (5.7)$$

Protože je kvaternion q jednotkový, normy kvaternionů w i v jsou shodné, došlo proto pouze k rotaci (zachování délky).

Dosazením definice q (5.6) do výše uvedeného vztahu, roznásobením a několika úpravami dostaneme rovnost

$$\vec{v}_w = (\vec{v}_v - \vec{n}(\vec{n} \cdot \vec{v}_v)) \cos \alpha + (\vec{n} \times \vec{v}_v) \sin \alpha + \vec{n}(\vec{n} \cdot \vec{v}_v) \quad (5.8)$$

kde člen v první závorce je vektor kolmý na \vec{n} a poslední člen součtu je vektor rovnoběžný s \vec{n} (důsledky Gram-Schmidtovy ortogonalizace), jedná se o známý Rodriguesův vzorec pro rotaci vektoru \vec{v}_v okolo osy \vec{n} o úhel α .

Matice rotace R v prostoru \mathbb{R}^3 okolo os z, y, x o Eulerovy úhly ψ, θ, ϕ , udané ve stejném pořadí, je dána jako

$$\mathbf{R} = R_{\psi}^Z \cdot R_{\theta}^Y \cdot R_{\phi}^X \quad (5.9)$$

$$= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (5.10)$$

$$= \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (5.11)$$

Tuto matici R lze vyjádřit pomocí prvků kvaternionu rotace q jako

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_0q_2 + 2q_1q_3 \\ 2q_0q_2 + 2q_1q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_2 - 2q_0q_3 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (5.12)$$

$$= \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 - 2q_0q_3 & 2q_1q_2 + 2q_0q_3 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 + q_2^2 - 1 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_2 - 2q_0q_3 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 + q_3^2 - 1 \end{bmatrix} \quad (5.13)$$

Proto z kvaternionu rotace q lze určit Eulerovy úhly ϕ, θ, ψ jako

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan2\left(\frac{R_{32}}{R_{33}}\right) \\ -\arcsin(R_{31}) \\ \arctan2\left(\frac{R_{21}}{R_{11}}\right) \end{bmatrix} = \begin{bmatrix} \arctan2\left(\frac{2q_2q_3 + 2q_0q_1}{2q_0^2 + q_3^2 - 1}\right) \\ -\arcsin(2q_1q_3 - 2q_0q_2) \\ \arctan2\left(\frac{2q_1q_2 + 2q_0q_3}{2q_0^2 + 2q_1^2 - 1}\right) \end{bmatrix} \quad (5.14)$$

Z Eulerových úhlů ϕ, θ, ψ lze určit kvaternion rotace q jako

$$\mathbf{q} = \begin{bmatrix} \cos(\psi/2) \\ 0 \\ 0 \\ \sin(\psi/2) \end{bmatrix} \begin{bmatrix} \cos(\theta/2) \\ 0 \\ \sin(\theta/2) \\ 0 \end{bmatrix} \begin{bmatrix} \cos(\phi/2) \\ \sin(\phi/2) \\ 0 \\ 0 \end{bmatrix} \quad (5.15)$$

$$= \begin{bmatrix} \cos(\phi/2) \cos(\theta/2) \cos(\psi/2) + \sin(\phi/2) \sin(\theta/2) \sin(\psi/2) \\ \sin(\phi/2) \cos(\theta/2) \cos(\psi/2) - \cos(\phi/2) \sin(\theta/2) \sin(\psi/2) \\ \cos(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \sin(\psi/2) \\ \cos(\phi/2) \cos(\theta/2) \sin(\psi/2) - \sin(\phi/2) \sin(\theta/2) \cos(\psi/2) \end{bmatrix} \quad (5.16)$$

Každá matice rotace R je ortonormální a platí pro ni tedy vztah $R^{-1} = R^T$. Nechť se matice rotace mění s časem, tj. $R = R(t)$. Z ortonormality plyne

$$\mathbf{R}(t)\mathbf{R}(t)^T = \mathbf{I} \quad (5.17)$$

a derivací je rovnost

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T + \mathbf{R}(t)\dot{\mathbf{R}}(t)^T = \mathbf{0} \quad (5.18)$$

ve které lze vidět rovnost $\mathbf{S}(t) + \mathbf{S}(t)^T = \mathbf{0}$ platnou pro všechny antisymetrické matice $\mathbf{S}(t)$. Odtud

$$\dot{\mathbf{R}}(t) = \mathbf{S}(t)\mathbf{R}(t) \quad (5.19)$$

Vektorový součin vektoru $\vec{\mathbf{a}} = (a_1, a_2, a_3)$ s libovolným vektorem $\vec{\mathbf{b}} = (b_1, b_2, b_3)$ je lineární operací, a proto lze psát

$$\vec{\mathbf{a}} \times \vec{\mathbf{b}} = \begin{bmatrix} a_2b_3 - b_2a_3 \\ a_3b_1 - b_3a_1 \\ a_1b_2 - b_1a_2 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \vec{\mathbf{b}} = [\mathbf{A} \times] \vec{\mathbf{b}} \quad (5.20)$$

kde antisymetrická matice $[\mathbf{A} \times]$ je maticovým operátorem vektorového součinu (pro vektor $\vec{\mathbf{a}}$).

Spojením výsledku výše se vztahem (5.19) postupně vyplyne, že časovou derivací každé matice rotace je vektorový součin určitého vektoru (konkrétně vektoru odpovídajícího úhlové rychlosti) s maticí rotace:

$$\dot{\mathbf{R}}(t) = [\boldsymbol{\Omega} \times](t)\mathbf{R}(t) \quad (5.21)$$

Podobně, ale složitěji lze odvodit časovou derivaci kvaternionu rotace q :

$$\dot{q} = \frac{1}{2}\boldsymbol{\omega} \cdot q \quad (5.22)$$

Tato rovnost je důsledkem výše uvedených vztahů mezi maticemi, vektorovým součinem a kvaterniony, včetně některých důležitých závěrů a kvaternionových identit plynoucích z teorie grup, konkrétně z izomorfizmu Lieovy grupy $SO(3)$ reprezentující všechny rotace v prostoru \mathbb{R}^3 s podprostorem prostoru kvaternionů \mathbb{H} obsahujícím všechny kvaterniony rotace.

Na některé tyto vztahy se dále odkazují níže při vysvětlení Madgwickova algoritmu.

Přestože nejsou kvaterniony řešením žádných algebraických rovnic (která vždy spadají nejvýše do prostoru komplexních čísel \mathbb{C}), nachází velké uplatnění například v počítačové grafice. Jejich výhodou oproti použití jinak běžného popisu orientace pomocí Eulerových úhlů (tj. úhlů vůči všem třem osám) je možnost správného popisu rotace, bez rizika vzniku singulární matice popisující orientaci (problém se nazývá „gimbal lock“). Lze snadno interpolovat rotaci (tj. kvaterniony rotace, např. lineárně mezi počátečním a cílovým kvaternionem rotace, např. algoritmy LERP a SLERP) pro dosažení plynulé rotace v prostoru (čehož se hojně využívá např. při generování animací na počítači).

5.3 Madgwickův algoritmus

Jedná se o jeden z nejčastěji používaných jednoduchých AHRS (angl. *Attitude and Heading Reference Systems*, neboli také MARG, angl. *Magnetic, Angular Rate and Gravity*) algoritmů. Jeho cílem je výpočet orientace sensorického systému obsahujícího akcelerometr, gyroskop a magnetometr, vůči Zemi, za použití minimálního výpočetního výkonu. Jeho princip používá iterativní metodu výpočtu kvaternionu reprezentujícího natočení systému vůči Zemi s využitím metody gradientního sestupu. V následujícím textu stručně vysvětluji tento princip, dopodrobna vysvětlený v původním článku [24], publikovaným autory Madgwick et al.

5.3.1 Orientace z gyroskopických dat

Výsledkem gyroskopického měření jsou úhlové rychlosti $\omega_x, \omega_y, \omega_z$ otáčení sensorického systému okolo všech jeho 3 os x, y, z , udané v jednotkách rad/s , kde označení os odpovídá na obr. 5.1 (převzato z [17]) alternativnímu Taitovu-Bryanovu značení odpovídajících úhlů náklonů, angl. *pitch, roll, yaw*. Pokud z těchto údajů zformujeme kvaternion (zapsaný vektorově po složkách) jako

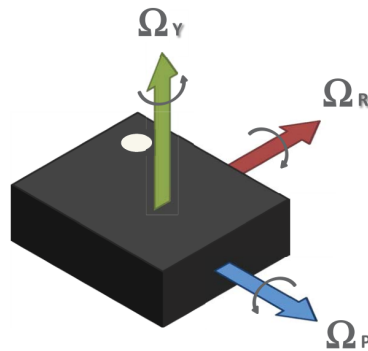
$${}^S\omega = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \end{bmatrix} \quad (5.23)$$

kde horní levý index S označuje vztah údajů k rámu sensorického systému, můžeme kvaternion reprezentující okamžitou úhlovou rychlost rotace rámu vůči Zemi vypočítat s využitím vztahu (5.22) jako

$${}^S\hat{q}_{\omega,t} = \frac{1}{2} {}^S\hat{q}_{\omega,t-1} \cdot {}^S\omega \quad (5.24)$$

kde stříška nad symbolem zde značí, že je kvaternion normalizovaný, a dolní index ω značí, že výpočet byl proveden na základě úhlových rychlostí. Výsledný kvaternion rotace bude z těchto dat počítán později numerickou integrací.

Kvaternion ${}^S\hat{q}_{\omega,t-1}$ odpovídá rotaci rámu plynoucí z předchozího měření.



Obrázek 5.1: Vyznačení os gyroskopu vůči rámu senzoru LSM6DSL

5.3.2 Orientace v homogenních polích

V homogenním poli je nejprve určen referenční směr toku vůči Zemi kvaternionem ${}^E\hat{\mathbf{d}}$. Naměřené hodnoty (v jednotlivých osách rámu senzoru) jsou vyjádřeny kvaternionem ${}^S\hat{\mathbf{s}}$.

Výpočet kvaternionu rotace sensorického rámu vůči Zemi je pak formulován jako optimalizační úloha:

$$\min_{\hat{\mathbf{q}}} f({}^S\hat{\mathbf{q}}, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{s}}) \quad (5.25)$$

kde minimalizovaná funkce odvozená pomocí vztahu (5.7) je

$$f({}^S\hat{\mathbf{q}}, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{s}}) = {}^S\hat{\mathbf{q}}^* \cdot {}^E\hat{\mathbf{d}} \cdot {}^S\hat{\mathbf{q}} - {}^S\hat{\mathbf{s}} \quad (5.26)$$

a značí odchylku rotovaného referenčního směru toku od měření. Je hledán tedy optimální kvaternion rotace rámu, pro který odpovídá rotovaný referenční směr toku pole Země naměřenému směru toku v osách rámu.

5.3.3 Optimalizační algoritmus

Madgwickův algoritmus využívá optimalizačního algoritmu gradientního sestupu (angl. *gradient descent*) pro nalezení optimálního řešení úlohy (5.25). Kvaternion rotace v $(k+1)$ -ním iteračním kroku je počítán jako

$${}^S\hat{\mathbf{q}}_{k+1} = {}^S\hat{\mathbf{q}}_k - \mu \frac{\nabla f({}^S\hat{\mathbf{q}}_k, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{s}})}{\|\nabla f({}^S\hat{\mathbf{q}}_k, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{s}})\|}, k = 0, 1, 2, \dots, n \quad (5.27)$$

kde hodnota gradientu funkce f odpovídá směru největšího růstu počítané odchylky, a lze ji vypočítat za pomoci Jakobiánu této funkce jako

$$\nabla f({}^S\hat{\mathbf{q}}_k, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{s}}) = \mathbf{J}^T({}^S\hat{\mathbf{q}}_k, {}^E\hat{\mathbf{d}}) \cdot f({}^S\hat{\mathbf{q}}_k, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{s}}) \quad (5.28)$$

a μ je velikost iteračního kroku.

5.3.4 Bloková schémata algoritmu

Předpokladem je, že se uvažuje homogenita magnetického i gravitačního pole Země. Zároveň se předpokládá, že magnetometr ovlivňuje pouze magnetické pole Země, a akcelerometr pouze gravitační pole Země.

Výsledná minimalizovaná funkce f je upravena pro využití pouze dat z akcelerometru (pro 6-DOF senzory) nebo z akcelerometru společně s magnetometrem (pro 9-DOF senzory). Referenční směr gravitačního pole Země je uvažován pouze vertikální, tj.

$${}^E\hat{\mathbf{d}} \equiv {}^E\hat{\mathbf{g}} = (0, 0, 0, 1) \quad (5.29)$$

a referenční směr magnetického pole je uvažován pouze dvojrozměrný s prvky vertikálními a horizontálními, s hodnotami odpovídajícím magnetické inklinaci, tj.

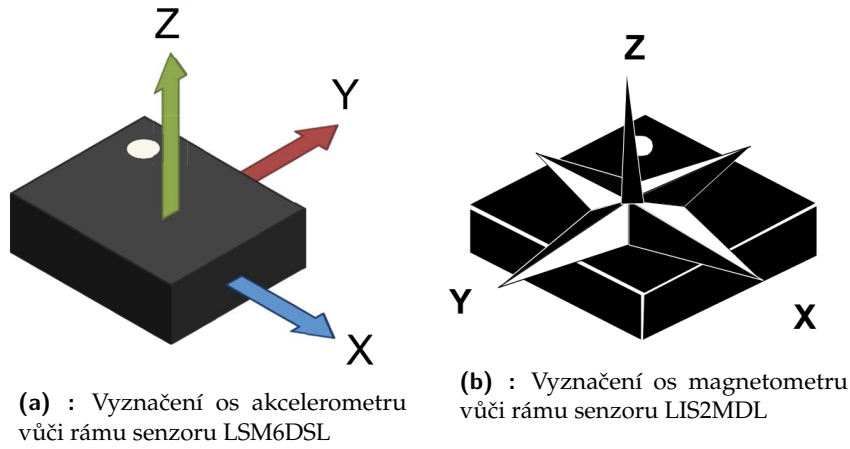
$${}^E\hat{\mathbf{a}} \equiv {}^E\hat{\mathbf{b}} = (0, b_x, 0, b_z) \quad (5.30)$$

Výsledky měření akcelerometru v základních jednotkách m/s^2 a magnetometru v jednotkách μT jsou vyjádřeny pomocí kvaternionů postupně jako

$${}^S\hat{\mathbf{s}} \equiv {}^S\hat{\mathbf{a}} = (0, a_x, a_y, a_z) \quad (5.31)$$

$${}^S\hat{\mathbf{s}} \equiv {}^S\hat{\mathbf{m}} = (0, m_x, m_y, m_z) \quad (5.32)$$

s osami odpovídajícími popisu senzorů 5.2 (převzato z [17] a [18]).



Obrázek 5.2: Vyznačení os vůči rámcům senzorů

Iterační algoritmus by běžně vyžadoval větší počet iterací pro výpočet každé nové orientace rámu senzoru, dle původního článku [24] však stačí pouze jeden iterační krok.

Výsledná derivace kvaternionu rotace je dána jako vážený průměr výpočtů z gyroskopických údajů a z homogenních polí (daný váhovým koeficientem β):

$${}^S\dot{\mathbf{q}}_{est,t} = {}^S\dot{\mathbf{q}}_{\omega,t} - \beta \frac{\nabla f}{\|\nabla f\|} \quad (5.33)$$

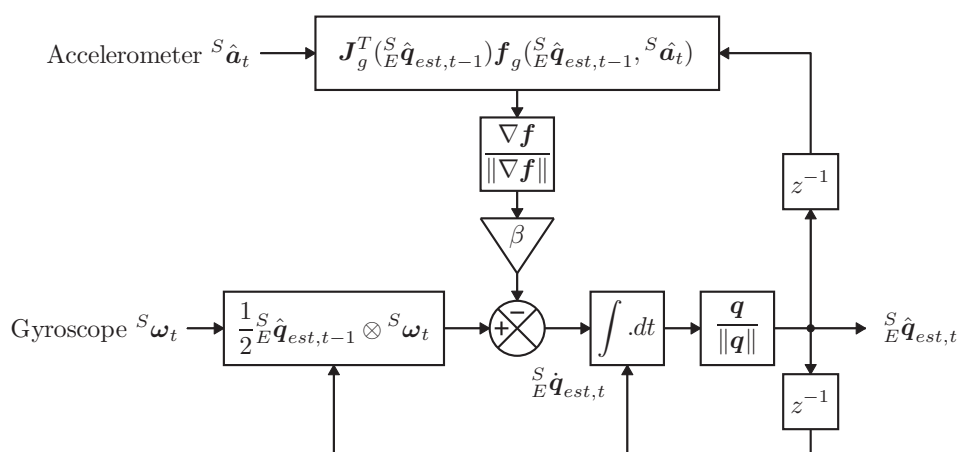
Numerickou integrací vzorce výše je na závěr vypočten výsledný kvaternion rotace jako

$${}^S\hat{\mathbf{q}}_{est,t} = {}^S\hat{\mathbf{q}}_{est,t-1} + {}^S\dot{\mathbf{q}}_{est,t} \delta t \quad (5.34)$$

kde δt je vzorkovací perioda.

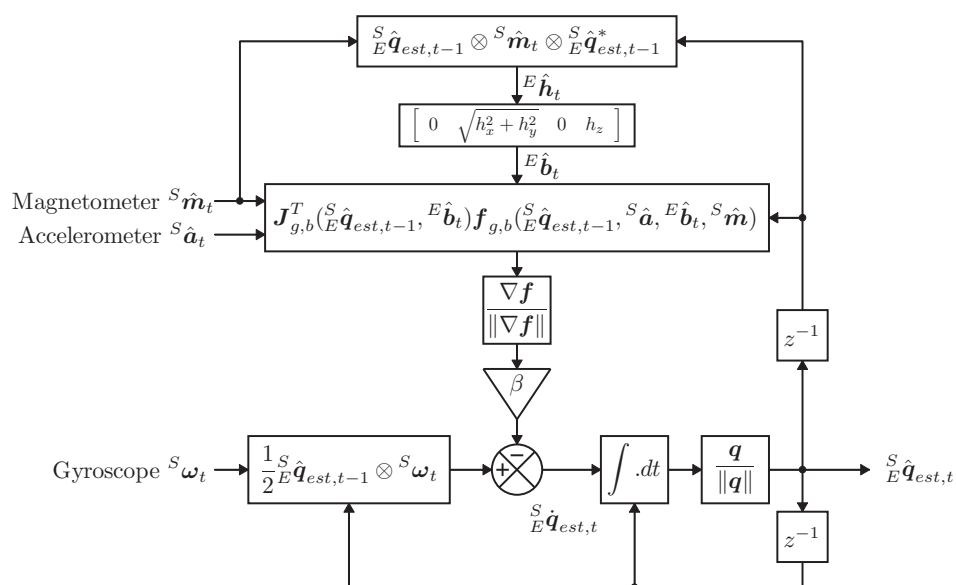
Z výsledného kvaternionu rotace ${}^S\hat{\mathbf{q}}_{est,t}$ jsou jednotlivé úhly orientace senzoru vypočteny dle vzorce 5.14.

Z původního článku [24] jsem uvedl bloková schémata 5.3 a 5.4 algoritmu určeného pro 6-DOF senzory a pro 9-DOF senzory.



Obrázek 5.3: Blokové schéma Madgwickova algoritmu pro 6-DOF senzory

Blokové schéma 5.4 je doplněno o kompenzaci zkreslení magnetometrického měření (angl. *magnetic distortion compensation*), jež vyžaduje předchozí kalibraci magnetometru (výpočet kompenzačních koeficientů).



Obrázek 5.4: Blokové schéma Madgwickova algoritmu pro 9-DOF senzory

5.3.5 Implementace

Při implementaci jsem vyšel z hotové implementace naprogramované v jazyce C dostupné z [25]. Tuto implementaci jsem přepsal do platné syntaxe jazyku Python a zachoval celou funkční část implementace.

■ 5.4 Jiné algoritmy

■ 5.4.1 Mahonyho algoritmus

Algoritmus publikovaný autory Mahony et al. v článku [26] obdobně využívá k výpočtu orientace rámu senzoru vůči rámu Země kvaternionů. Jeho odvození je ovšem komplikovanější a vychází opět z teorie grup, dále z teorie signálů a řízení. Vhodným proporcionálním a integrálním koeficientem je řízena zpětnovazební smyčka stabilizující výstupní kvaternion rotace.

Při implementaci tohoto algoritmu v Pythonu jsem obdobným způsobem jako u Madgwickova algoritmu vyšel z dostupné implementace v [25] a upravil ji pro Python.

■ 5.4.2 Kalmanův filtr

Nejčastěji používaným řešením této problematiky je tzv. Kalmanův filtr, díky jeho přesnosti a efektivitě. Při popisu rotační kinematiky ve třech dimenzích se ovšem zvyšují výpočetní nároky nutné na jeho provoz.

Jeho odvození a způsob možné implementace pro výpočet orientace 9-DOF senzorů je publikován např. v článku [27]. Tento přístup jsem se rozhodl neimplementovat kvůli jeho složitosti a nedostatku snadno dostupných hotových a funkčních implementací.

■ 5.5 Vizualizace

■ 5.5.1 Výběr programovacího jazyka

Při programování aplikace určené pro prostorovou vizualizaci rukavic schopnou BLE komunikace jsem zvážil několik možností. Přitom jsem se snažil dodržet jednoduchost celé implementace a přenositelnost mezi operačními systémy (angl. *cross-platform*).

- **C/C++** není nejvhodnější volbou z hlediska jednoduchosti ani přenositelnosti. Pokud by byla aplikace psána v prostředí operačního systému Windows, pro komunikaci s rukavicemi by bylo nejvhodnější využít knihoven Windows SDK, konkrétně rozhraní WinRT. Vizualizaci by bylo nezbytné naprogramovat za využití rozhraní DirectX nebo OpenGL, s nutností vytvoření shaderů v HLSL a výpočetních algebraických algoritmů pro umístění pozorovatele i objektů v 3D prostoru.
- **Java** by byla vhodnou volbou díky přenositelnosti a dispozici snadno implementovatelné vizualizace, např. za pomoci knihovny Java3D. Bohužel v současné době neexistuje přenositelná knihovna pro BLE komunikaci. Pod Unixovými systémy lze využít např. knihovny TinyB, nicméně pod Windows by bylo nezbytné napsat a zkompileovat vlastní knihovnu

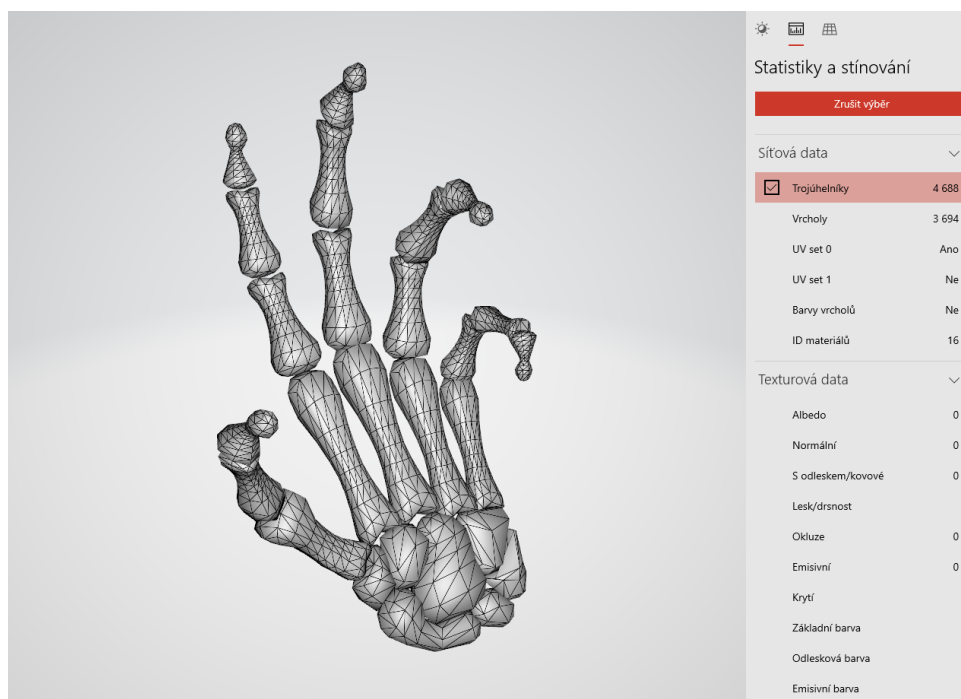
využívající výše zmíněné rozhraní WinRT, kterou využít v jazyce Java např. rozhraním JNI (Java Native Interface).

- **Python** nabízí snadnou přenositelnost mezi operačními systémy, přičemž nabízí velké množství knihoven umožňující snadnou prostorovou vizualizaci objektů, i knihovny pro BLE komunikaci.

Z výše zmíněných důvodů jsem vybral jazyk Python, s open-source knihovnami **ratcave** [28] pro vykreslování a **Bleak** [29] pro zajištění komunikace s rukavicemi.

■ 5.5.2 Třírozměrný model ruky

Aby vykreslený model ruky měl správné proporce (odpovídající fyziologii lidské ruky), vybral jsem volně dostupný (licencovaný pod CC Attribution 4.0) model obsahující vytvarované kosti všech článků prstů. [30]



Obrázek 5.5: Vykreslený model lidské ruky v programu 3D prohlížeč (součást Windows 10)

Tento model jsem musel následně upravit v programu Blender - narovnat články prstů, určit přesné souřadnice všech kloubů a převést model do formátu Wavefront OBJ, jelikož jiné formáty nejsou defaultně podporovány knihovnou raticave.

■ 5.5.3 Způsob implementace

Dalšími důležitými knihovnami použitými při implementaci byly:

- **asyncio** - knihovna vyžadovaná knihovnou Bleak, umožňující asynchronní přístup k Bluetooth adaptéru v počítači
- **numpy** - velmi efektivní knihovna určená pro nejrůznější matematické výpočty
- **pyglet** - knihovna vyžadovaná knihovnou ratcave, umožňující tvorbu okénkových aplikací
- **multiprocessing** - knihovna umožňující programování vícevláknových aplikací v Pythonu

Knihovnu multiprocessing bylo nezbytné použít kvůli blokaci celého vlákna jak klientem pro BLE komunikaci, tak i vykreslovacím skriptem. Knihovna umožňuje společné sdílení předem definovaných částí paměti oběma vlákny.

Po přijetí dat BLE klientem dojde k jejich okamžitému zpracování zvoleným algoritmem pro vypočtení výsledných úhlů, a následné uložení do sdílené paměti. Vykreslovací skript má vlastní časování pro zajištění konstantního počtu snímků za sekundu (FPS). V těchto intervalech přistupuje ke sdílené paměti, ze které čte vypočtené úhly a podle toho rotuje jednotlivé části modelu lidské ruky.

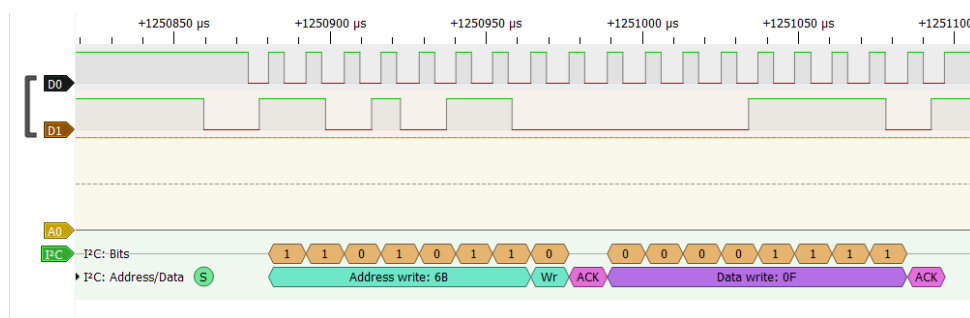
Každý třírozměrný útvar (angl. *mesh*) odpovídající každému článku prstu a hřbetu ruky má úchytný bod pro rotaci (angl. *pivot point*) umístěn v předcházejícím kloubu.

Kapitola 6

Zkouška zařízení

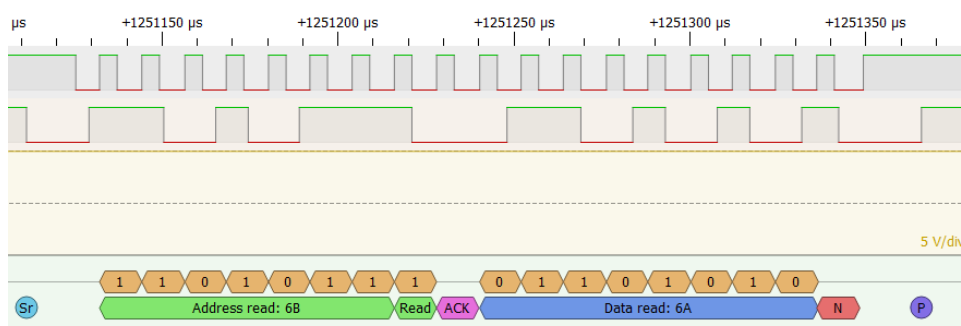
6.1 Ověření funkce multiplexoru

Velmi důležité bylo ověření správné komunikace pomocí multiplexované sběrnice I2C s jednotlivými senzory. Pro ověření jsem za pomoci logického analyzátoru ověřil časování hodinového signálu linky SCL (přednastaveného na 100 kHz) a obousměrné komunikace na multiplexované lince SDA s jednotlivými senzory typu LSM6DSL, zasláním 7bitové adresy senzoru (0x6B) s přidaným bitem 0 označujícím směr komunikace z mikroprocesoru do senzoru, a následně dotazu na identifikační registr (WHO_AM_I) dostupného standardně na adrese 0x0F.



Obrázek 6.1: Zaslání identifikačního požadavku po lince I2C

Každý ze sensorů poskytl bezbychnou odpověď na tento dotaz, začínající opět 7bitovou adresou senzoru (0x6B), následující bitem 1 označujícím směr komunikace ze senzoru do mikroprocesoru, a končící správnou identifikační hodnotou 0x6A, kterou lze srovnat s dokumentací senzoru [17].

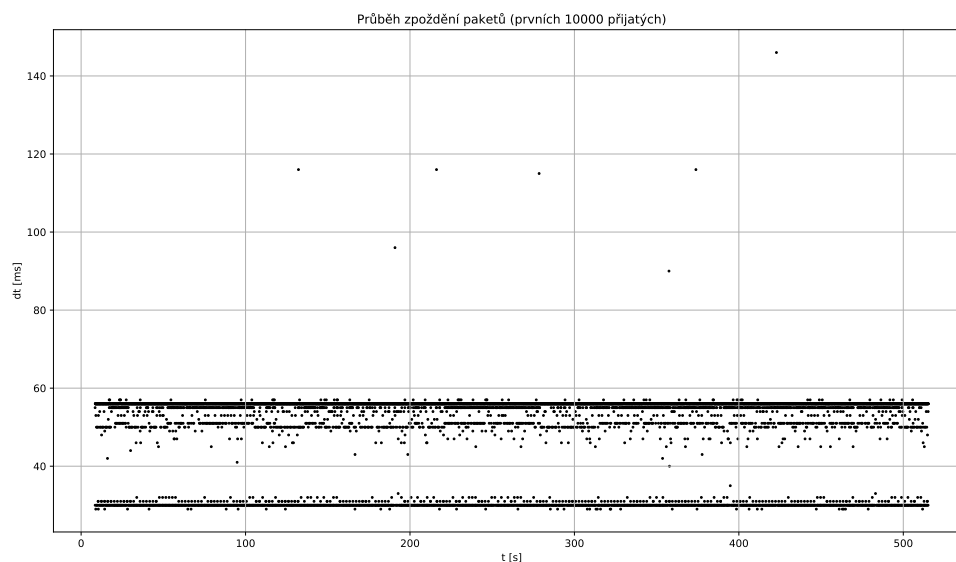


Obrázek 6.2: Odpověď na identifikační požadavek po lince I2C

6.2 Vizualizace nezpracovaných dat

Pro měření vizualizovaná za pomoci grafů a histogramů jsem zaznamenal prvních 10 tisíc přijatých paketů, přičemž celá rukavice a tedy i všechny senzory byly umístěny v klidu.

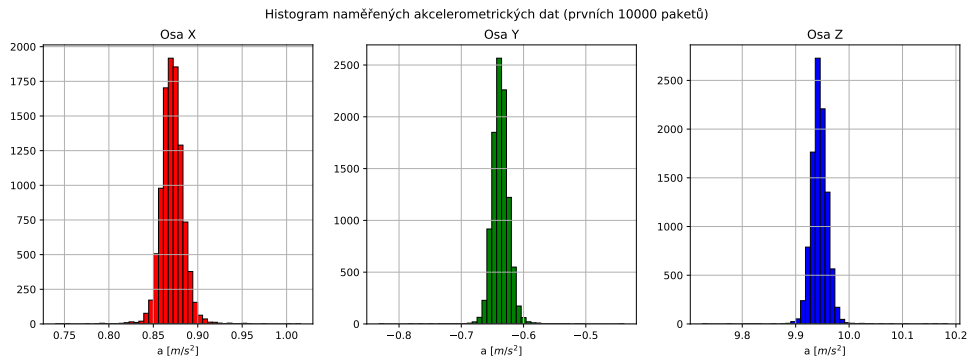
Pro jednotlivé přijaté pakety jsem vykreslil průběh jejich časové rozestupy na grafu č. 6.3. Rozestupy by měly být 30 ms, nicméně často došlo ke zpožděním způsobenými zřejmě nízkým výkonem použitého modulu s integrovaným mikroprocesorem, a tak se vyskytují pakety až s dvojnásobným časovým odstupem od nominální vzorkovací periody.



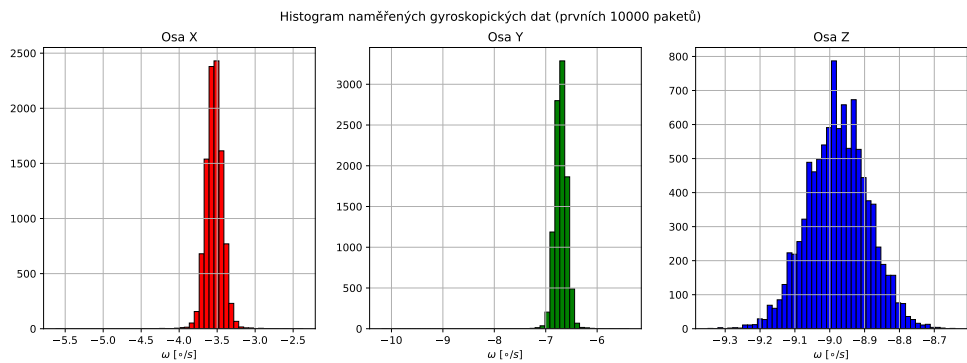
Obrázek 6.3: Měření intervalů mezi pakety

V následujících histogramech 6.4, 6.5 a 6.6 je vykresleno rozdělení naměřených hodnot (kterých je vždy v součtu 10000) ve všech třech osách a to

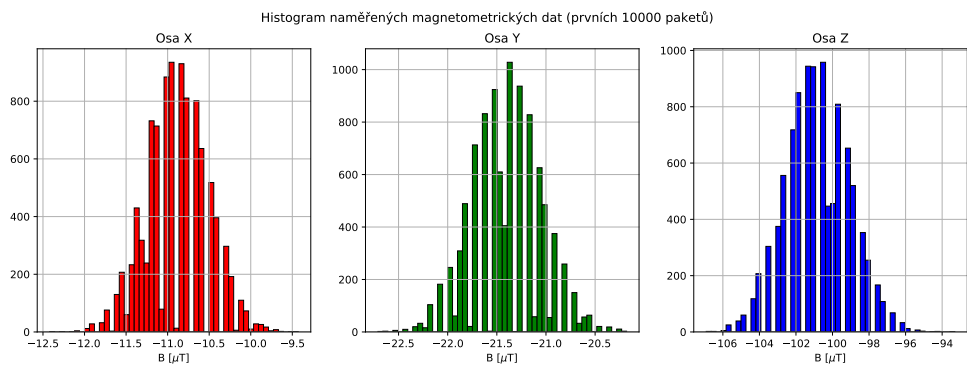
pro akcelerometr, gyroskop i magnetometr umístěných na hřbetu ruky. Toto rozdělení přibližně odpovídá normálnímu (gaussovskému) rozdělení.



Obrázek 6.4: Histogram pro měření akcelerometrem



Obrázek 6.5: Histogram pro měření gyroskopem

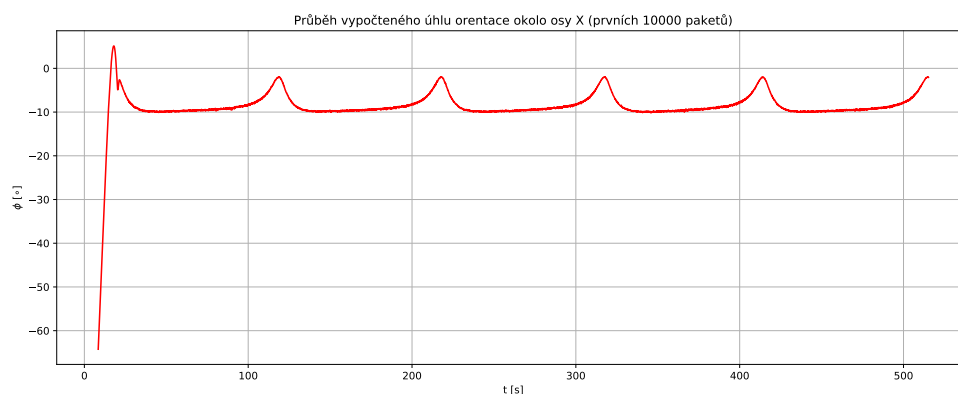


Obrázek 6.6: Histogram pro měření magnetometrem

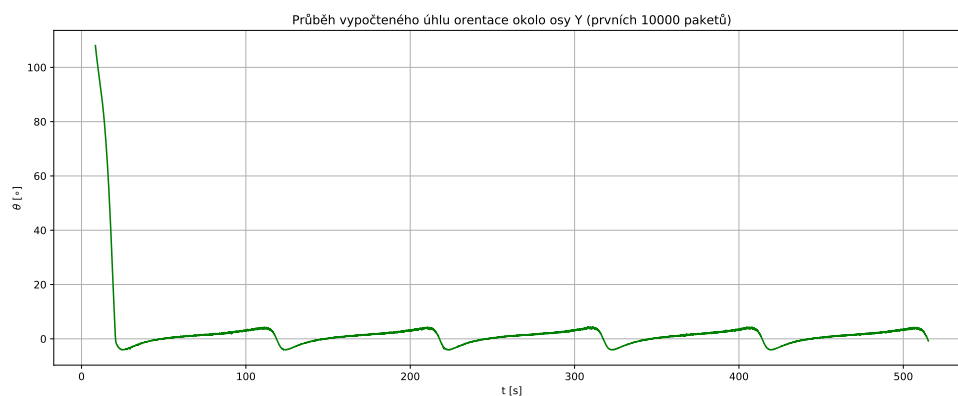
6.3 Vizualizace vypočtených úhlů

S využitím Madgwickova algoritmu byly pro každý přijatý paket vypočítány hodnoty úhlů výsledné orientace pohybových senzorů. K výše uvedeným

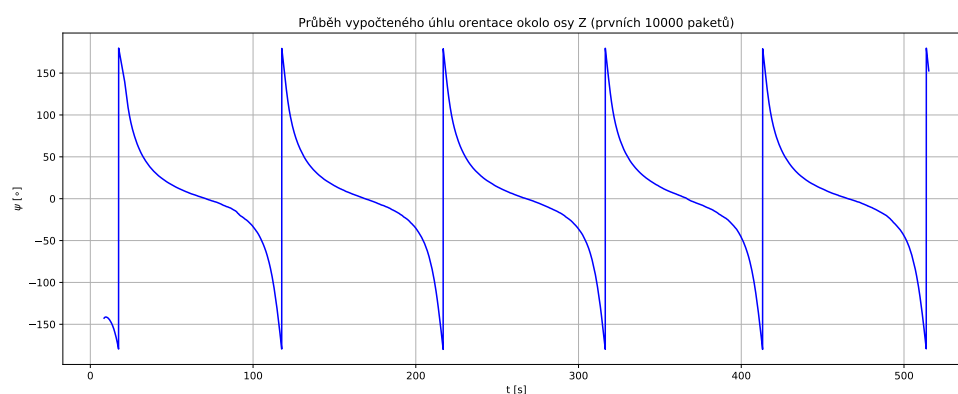
histogramům pro nepohybující se rukavici je dále uveden časový průběh vypočtených úhlů orientace hřbetu ruky ve všech třech osách.



Obrázek 6.7: Časový průběh vypočteného úhlu natočení rukavice okolo osy X



Obrázek 6.8: Časový průběh vypočteného úhlu natočení rukavice okolo osy Y



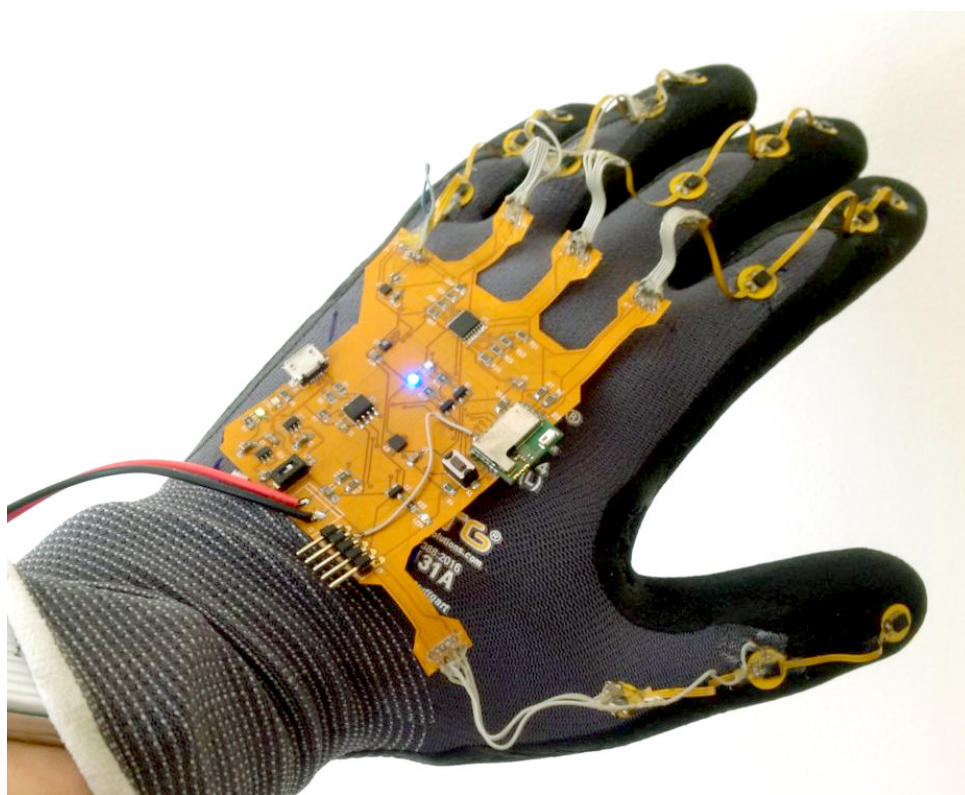
Obrázek 6.9: Časový průběh vypočteného úhlu natočení rukavice okolo osy Z

Při vizualizaci prostorového otáčení okolo osy Z referenčního rámu Země běžně docházelo k nechtěnému periodickému průběhu (od +180 do -180

stupňů) zřetelnému z grafu 6.9, a tento jev se mi nepodařilo dále odstranit.

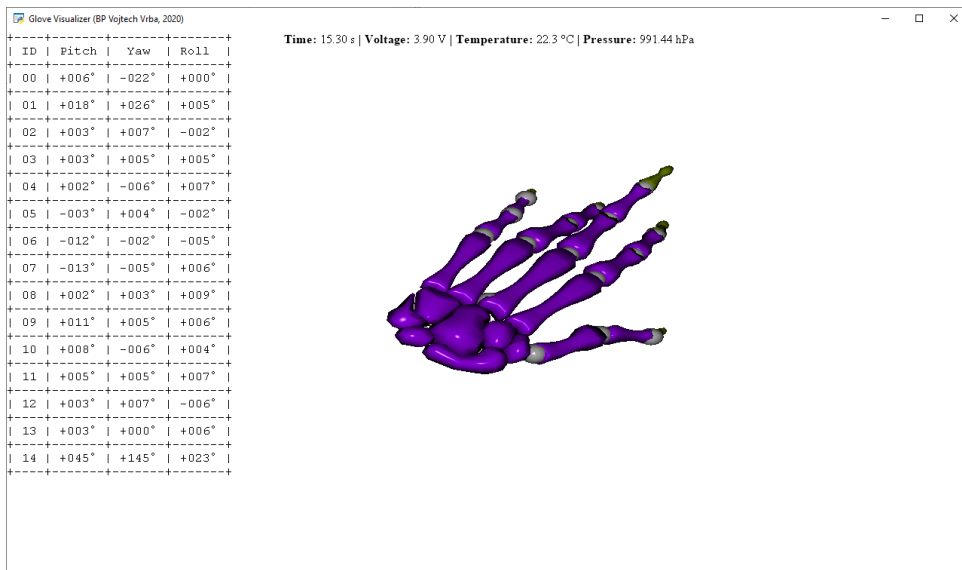
6.4 Prostorová vizualizace

Na fotografii 6.10 je vyobrazena navlečená funkční rukavice napájená z akumulátoru.



Obrázek 6.10: Fotografie navlečené rukavice

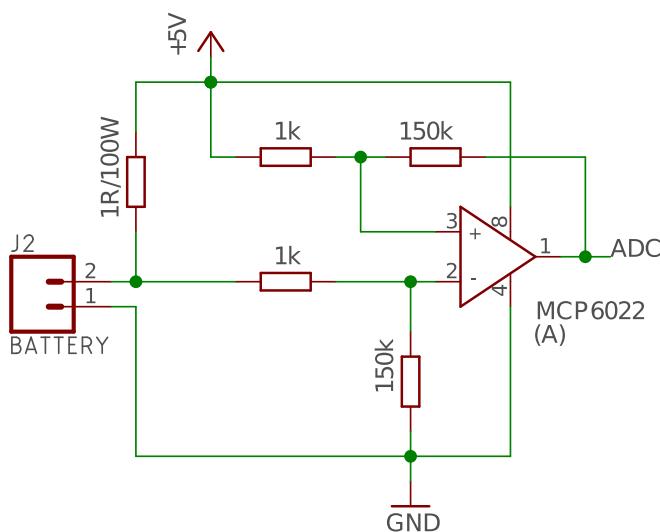
Na snímku obrazovky 6.11 lze vidět prostorovou vizualizaci ruky vykreslenou z naměřených dat.



Obrázek 6.11: Snímek vizualizace

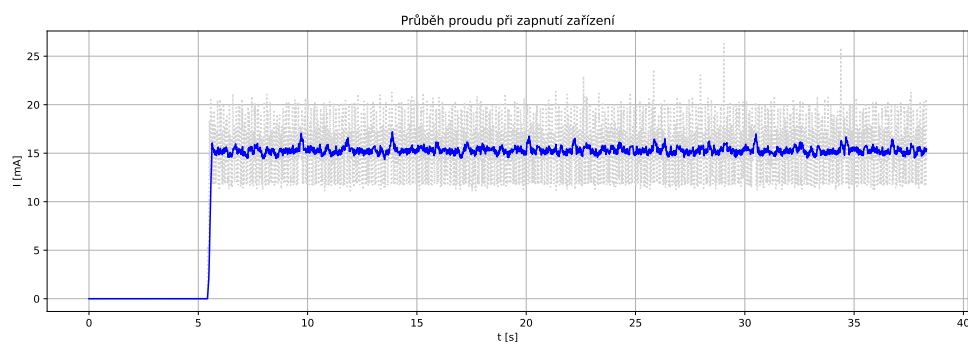
6.5 Měření spotřeby

Spotřebu proudu jsem měřil s využitím operačního zesilovače MCP6022 zapojeného jako rozdílový zesilovač dle schématu na obr. 6.12. Úbytek napětí na výkonovém rezistoru s nominální hodnotou 1 Ohm byl zesílen rozdílovým zesílením $G_D = \frac{150k}{1k} = 150$. Toto zapojení bylo napojeno na konektor J2 umístěný na rukavicích, určený primárně pro připojení Li-Po akumulátoru. Výstupní napětí bylo přivedeno na 10bitový AD převodník s vzorkovací frekvencí 100 Hz, fungující v celém rozsahu napětí od 0 do 5 V.



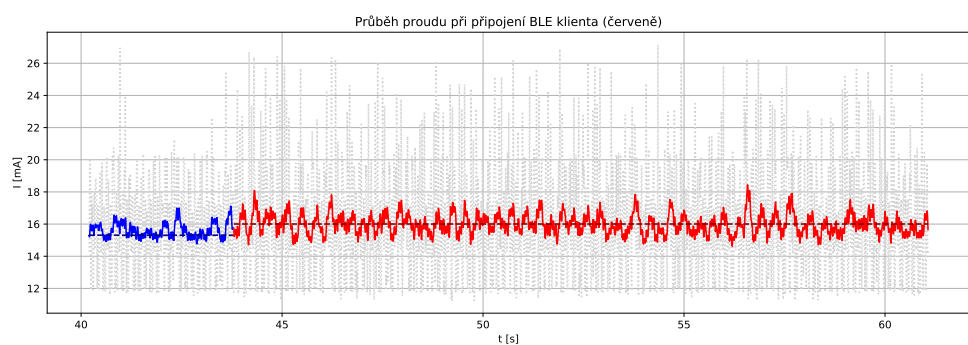
Obrázek 6.12: Zapojení obvodu pro měření proudu

Vzorkovaný signál je zakreslen na průbězích světle šedou barvou, a vyfiltrováný průběh je zobrazen barevně. Ve stavu, kdy nebyli připojeni k rukavici žádní klienti přes BLE, byla průměrná hodnota proudu rovna přibližně 15.3 mA.



Obrázek 6.13: Časový průběh napájecího proudu po přivedení napájení

V okamžiku připojení BLE klienta se průměrná spotřeba rukavice zvýšila na 16 mA, tj. téměř o 1 mA.



Obrázek 6.14: Časový průběh napájecího proudu při připojení BLE klienta

Kapitola 7

Diskuze

7.1 Objevené chyby a nedostatky

V samotném návrhu plošného spoje jsem po pájení objevil několik chyb, kterými jsou:

1. Chybné zapojení diody LED5 do pinu DIO12 Bluetooth modulu. Tento pin není možné dle dokumentace modulu použít jako výstupní, nýbrž pouze jako vstupní. Tedy diodu LED5 není možné ovládat, a kvůli plovoucí úrovni napětí na Gate tranzistoru BSS138 dochází k jejímu náhodnému rozsvěcení a zhášení diody. Tato chyba ovšem nebyla fatální, funkce indikační LED neměla vliv na funkčnost zbytku zařízení.
2. Chybějící spoj na plošném spoji určeném pro prsteníček, konkrétně mezi druhým a posledním článkem prstu chybí zemní spoj. Chyba vznikla špatným ověřením návrhu před odesláním do výroby. Tento problém jsem vyřešil nahrazením chybějícího spoje připájením náhradního flexibilního drátku.

Volba ohebného plošného spoje byla nanejvýš nešťastnou. V místě tuhých a větších součástech (např. konektoru J3 pro SWD rozhraní) snadno docházelo k jejich ulomení a také k lámání samotných vodivých cest na plošném spoji. Musel jsem proto ulomené cesty nahradit připájením ohebných drátků. Návrh ohebných DPS by měl zároveň mít co nejvíce zaoblené veškeré hrany. Jejich tloušťka, přirovnatelná k tloušťce papíru, způsobovala trhání DPS v místech ostřejších hran. V případě těchto problémů jsem musel nahradit celé větší části ohebných plošných spojů.

Volba kolmých konektorů pájených do prokovených otvorů DPS pro připojení spojů určených pro jednotlivé prsty k plošnému spoji na hřbetu ruky nebyla vhodná, jelikož opět snadno docházelo k lámání DPS. Tento problém jsem řešil přímým připájením drátků do otvorů určených pro konektory a následným zalepením epoxidovým lepidlem, aby tyto často namáhané části desek byly co nejtvrdší.

7.2 Další vývoj

Rukavice byla navržena pro pevně zvolené rozměry lidské ruky, s jistou volností umožněnou elasticitou samotných pracovních rukavic. Jelikož se proporce lidských rukou mohou velmi lišit, nebylo toto řešení dostatečně univerzální, pevně přilepené části spojů s pohybovými senzory se v některých případech nacházely přímo nad mezičládkovými klouby, a měření se tak stalo bezcenným.

Z tohoto důvodu a z důvodu příliš krátkých desek plošných spojů určených pro články prstů by bylo lepším řešením použití elastických vodičů (angl. *stretchable wires*) pro propojení neelastických částí nesoucích pohybové senzory.

Pro co nejdůvěhodnější fungování magnetometru by měl být vypracován přístup k jeho kalibraci spojené s inklinací magnetického pole Země v místě měření, nicméně kalibrační algoritmus jsem do svého řešení neimplementoval. Nutné je i nalezení nejvhodnějších parametrů pro nastavení gyroskopů a akcelerometrů (ODR, citlivost aj.).

Další úpravou hodnou dalšího vývoje je návrh i implementace upravených algoritmů pro přepočítání naměřených dat na výsledné úhly. Úhly natočení jednotlivých článků prstů v prostoru jsou na sobě závislé, správnějším přístupem by byl algoritmus produkující všechny orientace článků prstů naráz, čímž by se mohla výrazně snížit chybovost celého přepočtu.

V neposlední řadě stojí za zmínění i možné zlepšení programu pro prostorovou vizualizaci s příjemnějším uživatelským rozhraním, v němž by bylo možné ovládnutí kamery za pomoci myši a klávesnice, případně přímá interakce s vizualizovaným modelem ruky doplněná o zobrazení dalších informací, což by mohlo být také předmětem dalšího vývoje.

Kapitola 8

Závěr

V rámci své bakalářské práce jsem se nejprve věnoval teorii komunikačního standardu Bluetooth Low Energy v5.0, kterou jsem stručně vysvětlil v kapitole 2.

Poté jsem se věnoval návrhu a konstrukci vlastního sensorického systému tvořeného rukavicemi nesoucími akumulátor, potřebné napájecí obvody, programovatelný BLE modul a 15 pohybových sensorů doplněných o magnetometr, teploměr a snímač atmosferického tlaku.

Podarilo se mi zkonstruovat a naprogramovat funkční prototyp rukavic, vyfotografovaných na 6.10. Jedním z primárních požadavků na návrh byla nízká spotřeba celého zařízení, které se mi podařilo dosáhnout průměrným odběrem okolo 16 mA. Mezi objevené chyby a nedostatky patří především nesprávně zapojená LED dioda a problémy způsobené samotnou volbou materiálu desek plošných spojů, i nesprávným rozměrem některých navržených spojů, detailněji popsané v kapitole 7.

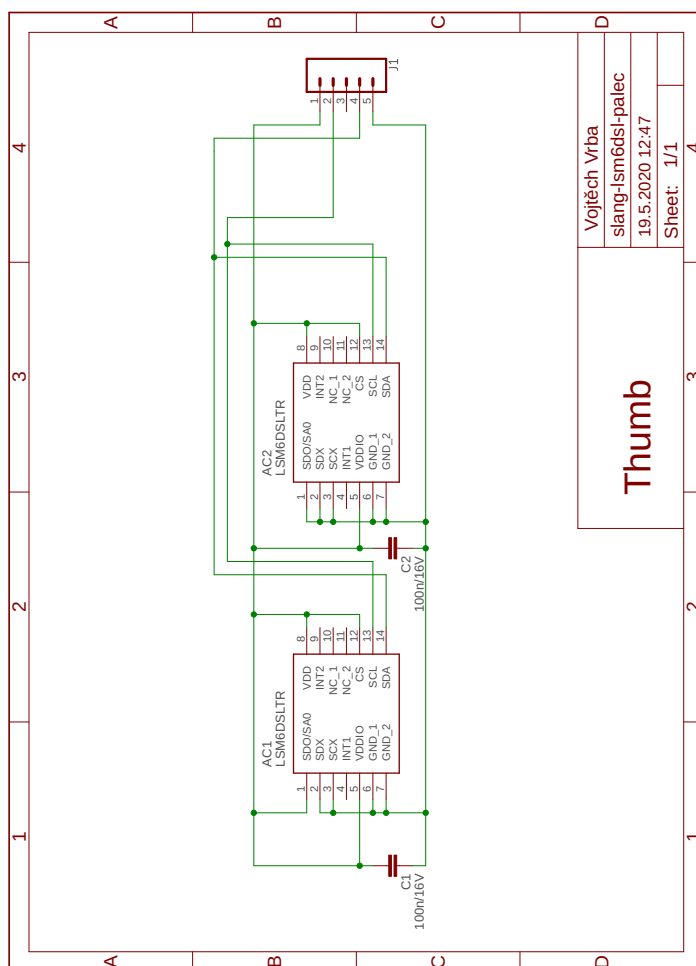
Současně jsem se věnoval algoritmům určeným na přepočítání naměřených údajů na prostorovou orientaci rámu senzoru vůči rámu Země. Kvůli jednoduchosti na implementaci jsem vybral Madgwickův algoritmus, a v kapitole 5 jsem stručně vysvětlil teorii spojenou s jeho odvozením.

Poslední praktickou částí mé práce byla tvorba programu pro prostorovou vizualizaci zjištěných orientací jednotlivých pohybových sensorů, posléze spárovaných s částmi (kostmi) třírozměrného modelu lidské ruky. Vzhled dokončené aplikace v provozu je na snímku obrazovky 6.11.

Přestože je výsledkem mé práce principiálně funkční sensorický systém doplněný o vizualizaci, je zde stále velký prostor pro zlepšení jak u samotné konstrukce, tak přepočítání algoritmů i vizualizace. Poznámky k dalšímu vývoji jsou uvedeny v kapitole 7.

Příloha A

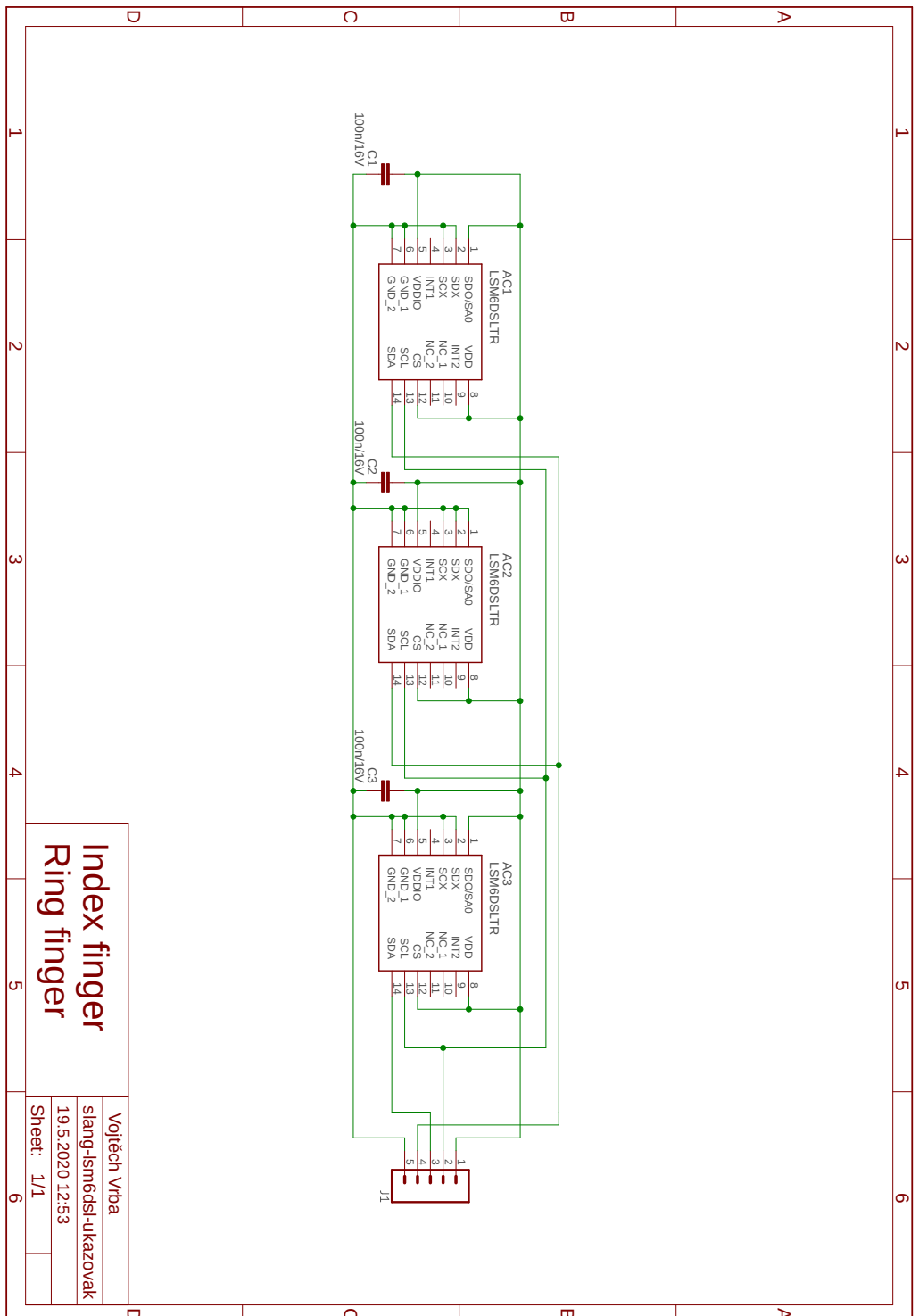
Schémata zapojení



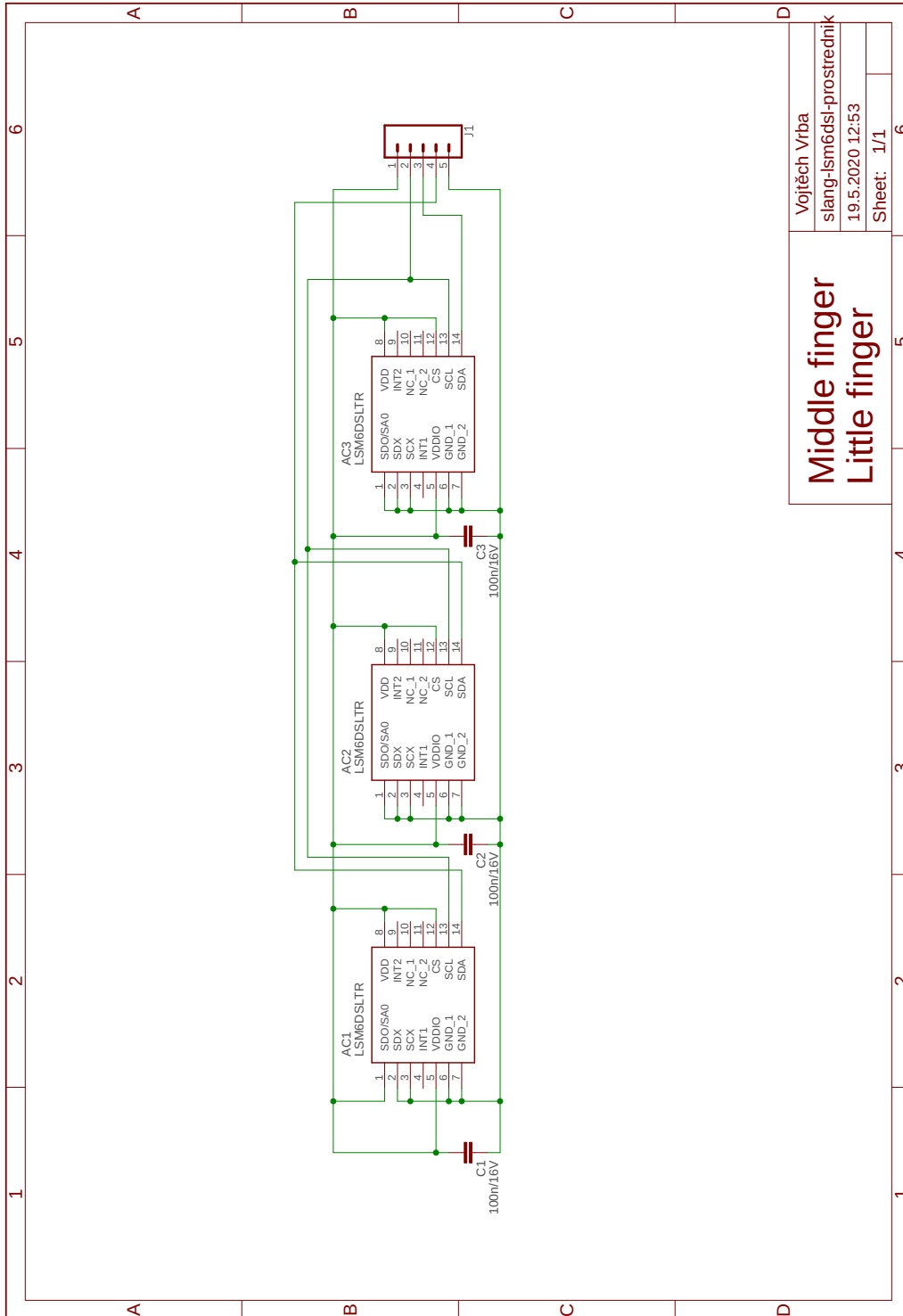
Thumb

Vojtěch Vrba
slang-lsm6dsi-palec
19.5.2020 12:47
Sheet: 1/1

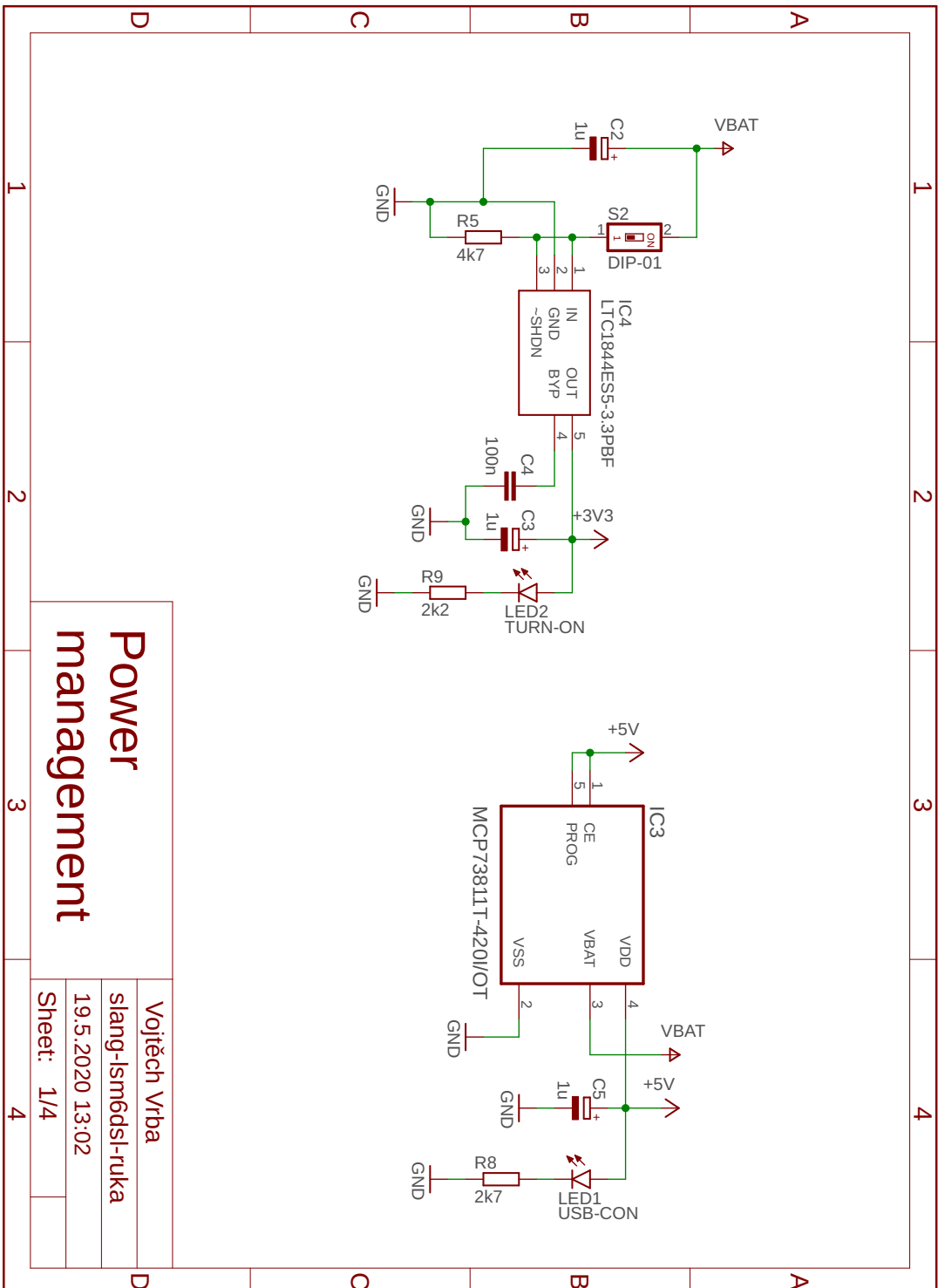
Obrázek A.1: Schéma zapojení pro palec



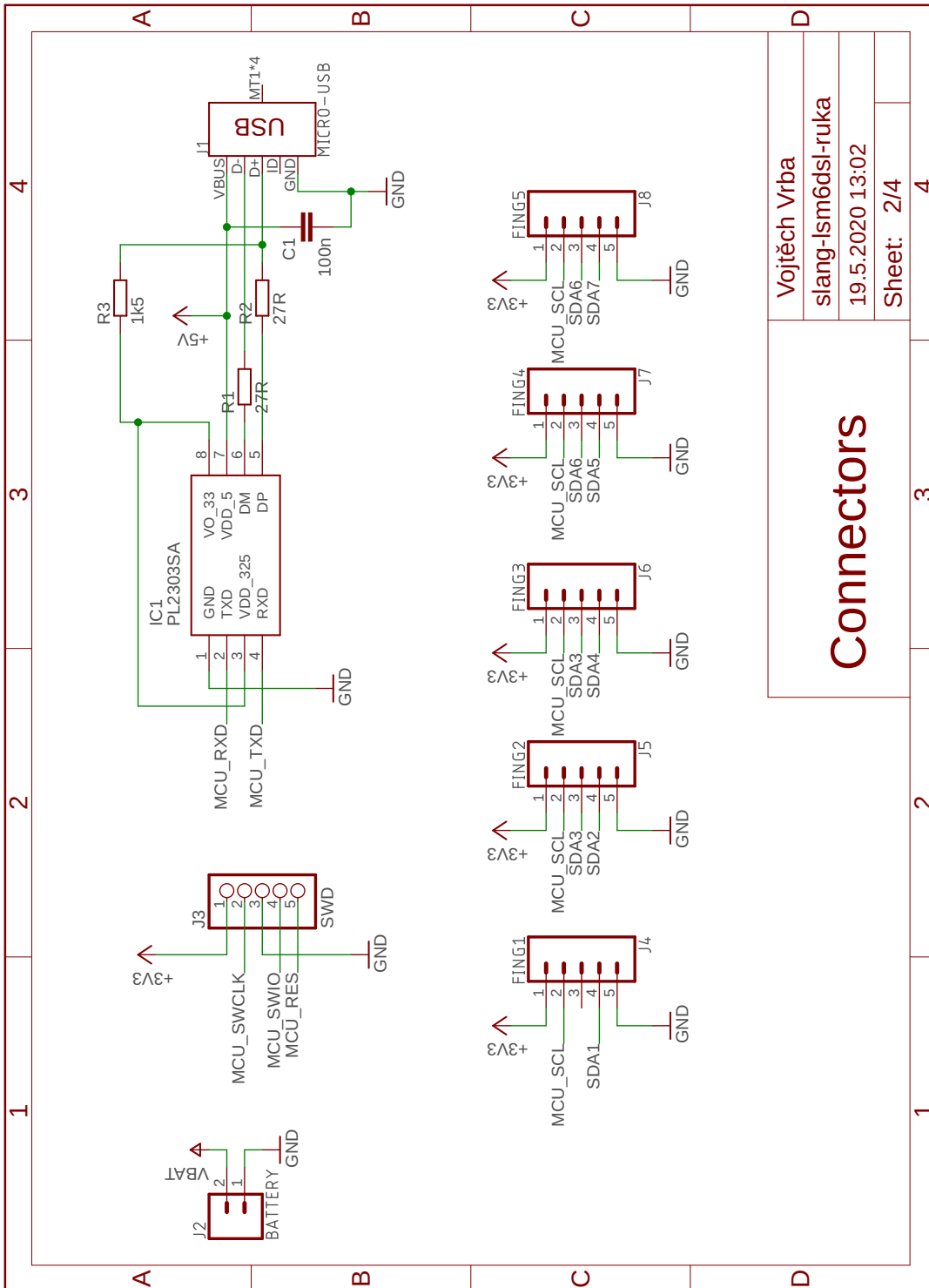
Obrázek A.2: Schéma zapojení pro ukazováček a prsteníček



Obrázek A.3: Schéma zapojení pro prostředníček a malíček



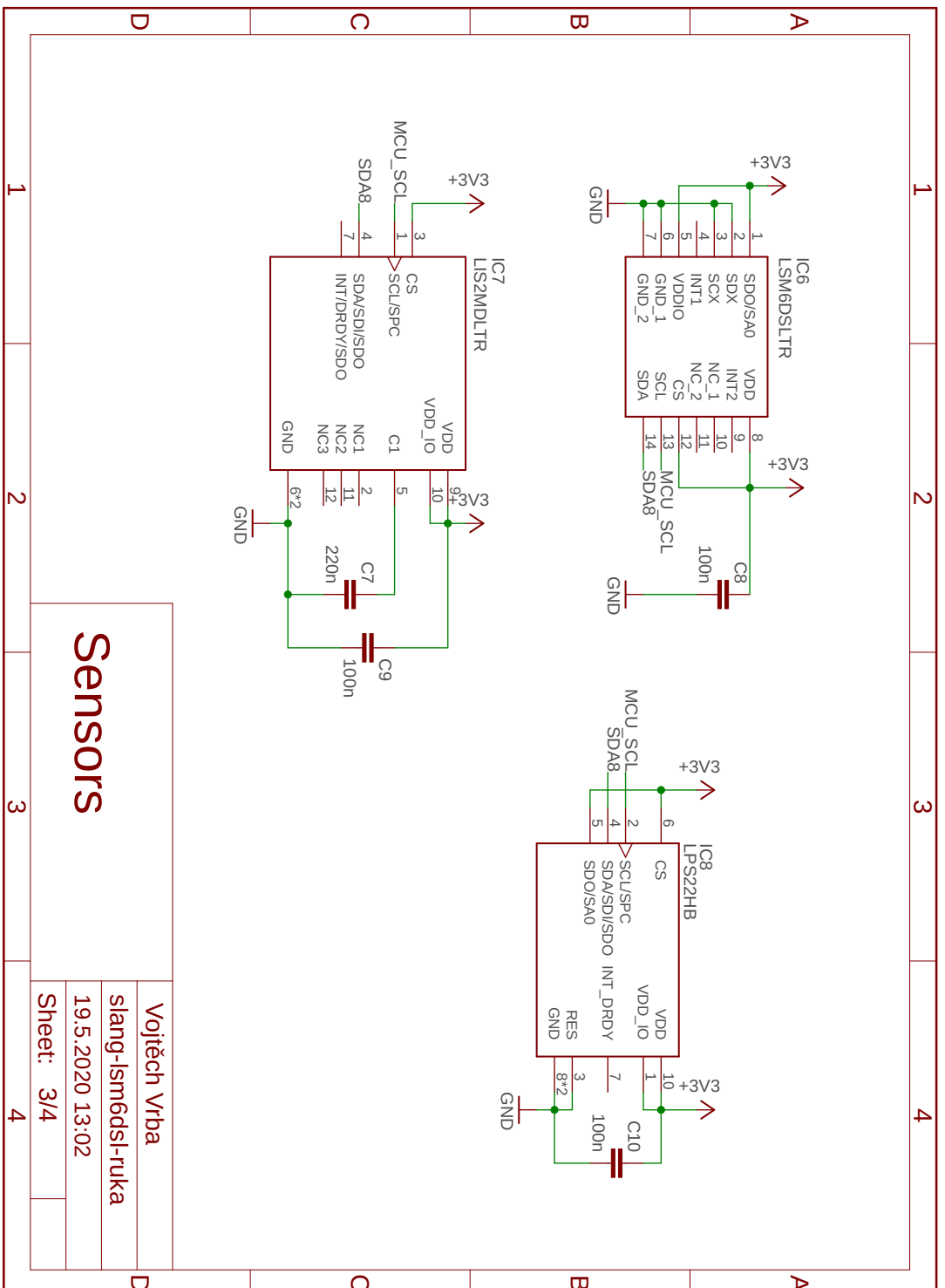
Obrázek A.4: Schéma zapojení pro ruku - napájecí obvody



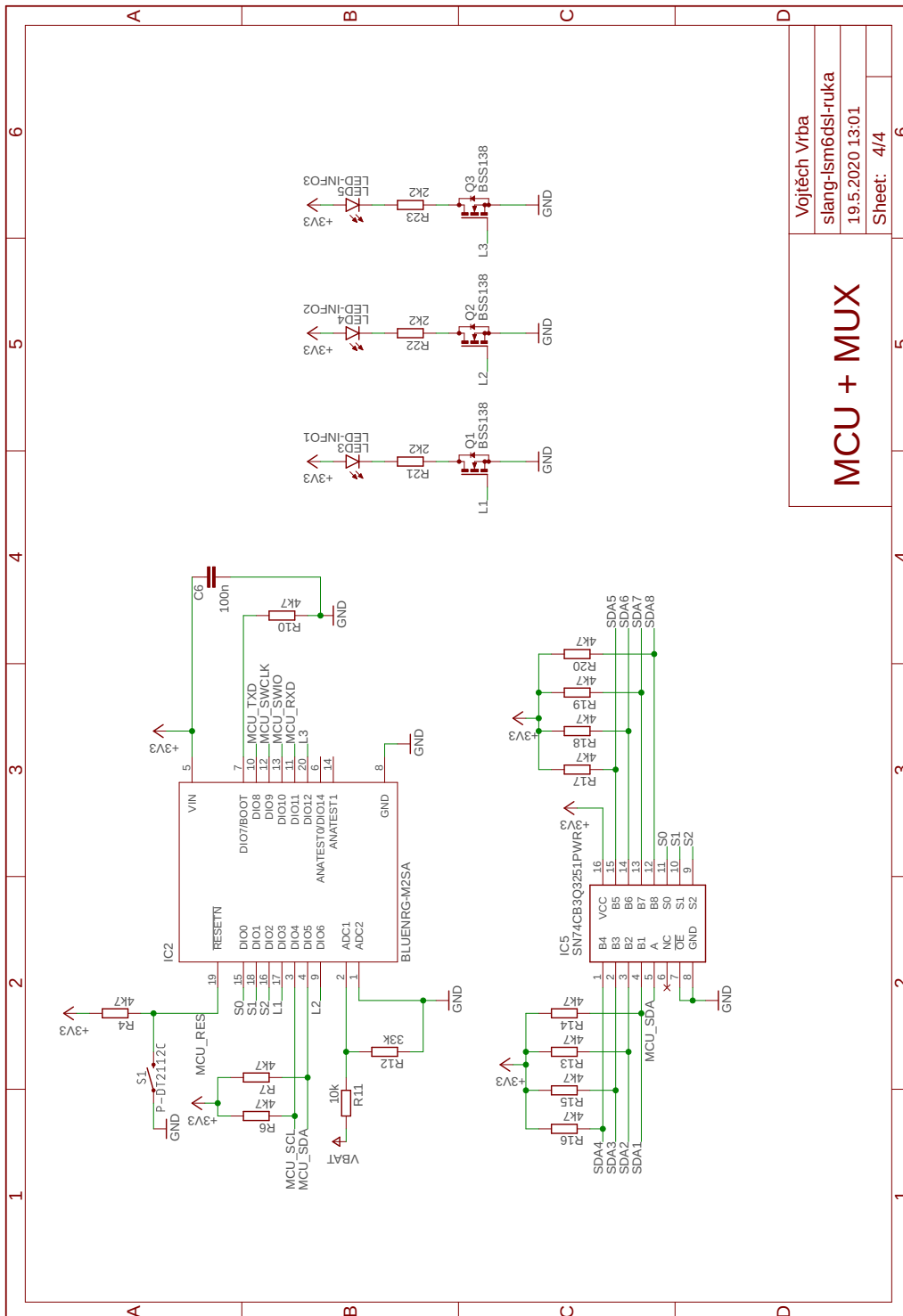
Connectors

Vojtěch Vrba
slang-lsm6dsl-ruka
19.5.2020 13:02
Sheet: 2/4

Obrázek A.5: Schéma zapojení pro ruku - konektory



Obrázek A.6: Schéma zapojení pro ruku - senzory



MCU + MUX		Vojtěch Vrba			
		slang-ism6dsi-ruka			
		19.5.2020 13:01			
		Sheet: 4/4			
1	2	3	4	5	6

Obrázek A.7: Schéma zapojení pro ruku - BLE modul a multiplexor

Příloha B

Seznam součástek

Uvedené počty součástek v každé tabulce odpovídají osazení jednoho kusu příslušné DPS. Tedy pro osazení jednoho kompletního páru rukavic je zapotřebí dvakrát tab. B.1, osmkrát tab. B.2, dvakrát tab. B.3 a dvakrát tab. B.4. Pozn.: V tabulkách není uveden Li-Po akumulátor 3.7 V 700 mAh.

Počet	Označení	Hodnota	Pouzdro	Popis
2	C1, C2	100n	0805	Keramický kondenzátor, 50V X7R 10%
2	AC1, AC2	LSM6DSLTR	LGA-14	6-DOF IMU senzor
1	J1	CONN	1x05 1.27mm	Kolíková lišta

Tabulka B.1: Součástky pro DPS na palec

Počet	Označení	Hodnota	Pouzdro	Popis
3	C1, C2, C3	100n	0805	Keramický kondenzátor, 50V X7R 10%
3	AC1, AC2, AC3	LSM6DSLTR	LGA-14	6-DOF IMU senzor
1	J1	CONN	1x05 1.27mm	Kolíková lišta

Tabulka B.2: Součástky pro DPS na ostatních prstech

Počet	Označení	Hodnota	Pouzdro	Popis
3	C2, C3, C5	1u	Panasonic A	Keramický kondenzátor, 25V X7R 10%
1	C4	100n	0805	Keramický kondenzátor, 50V X7R 10%
1	IC3	MCP73811T-4201/OT	SOT-23	Kontrolér nabíjení akumulátorů
1	IC4	LTC1844ES5-3.3PBF	SOT-23	LDO regulátor 3.3 V
1	J1	MICRO-USB	micro USB2F SMD	USB B micro zásuvka
1	J2	BATTERY	JST/SYP-2P	Konektor Li-Po akumulátoru
1	LED1	USB-CON	0805	Oranžová
1	LED2	TURN-ON	0805	Zelená
1	R5	4k7	0805	Rezistor SMD
1	R8	2k7	0805	Rezistor SMD
1	R9	2k2	0805	Rezistor SMD
1	S2	A6S-1101-H	DIP-01 SMD	Posuvný spínač

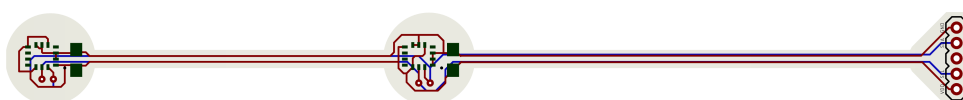
Tabulka B.3: Součástky pro napájecí obvod na hřebtu ruky

Počet	Označení	Hodnota	Pouzdro	Popis
5	C1, C6, C8, C9, C10	100n	0805	Kondenzátor keramický, 50V X7R 10%
1	C7	220n	0805	Kondenzátor keramický, 50V Y5V 20%
1	IC1	PL2303SA	SOIC-8	USB-UART převodník
1	IC2	BLUENRG-M2SA		Bluetooth modul
1	IC5	SN74CB3Q3251PWR	TSSOP-16	Mux / demux
1	IC6	LSM6DSLTR	LGA-14	6-DOF IMU senzor
1	IC7	LIS2MDLTR	LGA-12	Magnetometr
1	IC8	LPS22HB	HLGA-10L	Barometrický senzor
1	J3	SWD	1x05/90 2.54 mm	Kolíková lišta
5	J4, J5, J6, J7, J8	FING1..5	1x05 1.27 mm	Dutinková lišta
3	LED3, LED4, LED5	LED-INFO1..3	0805	Žlutá, červená, modrá
3	Q1, Q2, Q3	BSS138	SOT-23	MOSFET N-kanál
2	R1, R2	27R	0805	Rezistor SMD
1	R11	10k	0805	Rezistor SMD
1	R12	33k	0805	Rezistor SMD
1	R3	1k5	0805	Rezistor SMD
12	R4, R6, R7, R10, R13, R14, R15, R16, R17, R18, R19, R20	4k7	0805	Rezistor SMD
3	R21, R22, R23	2k2	0805	Rezistor SMD
1	S1	P-DT2112C	6.0x3.5 mm	SMD mikropínač

Tabulka B.4: Součástky pro zbytek obvodu na hřbetu ruky

Příloha C

Návrh DPS



Obrázek C.1: Návrh DPS pro palec



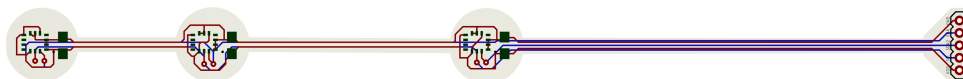
Obrázek C.2: Návrh DPS pro ukazováček



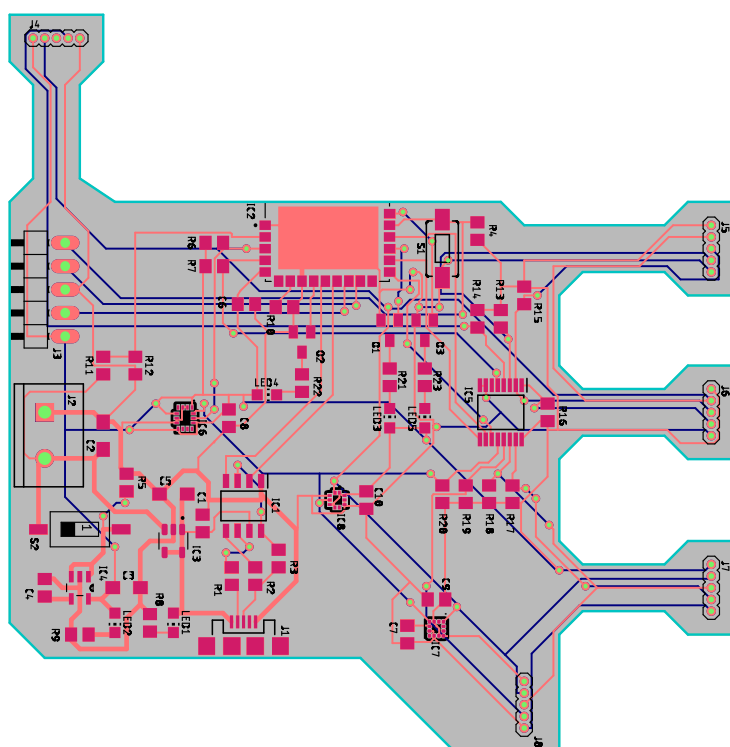
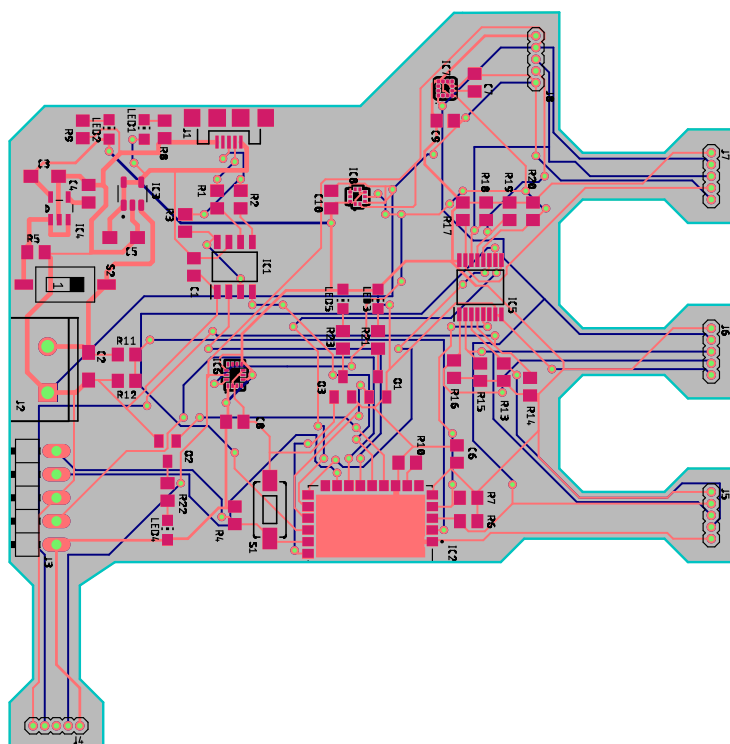
Obrázek C.3: Návrh DPS pro prostředníček



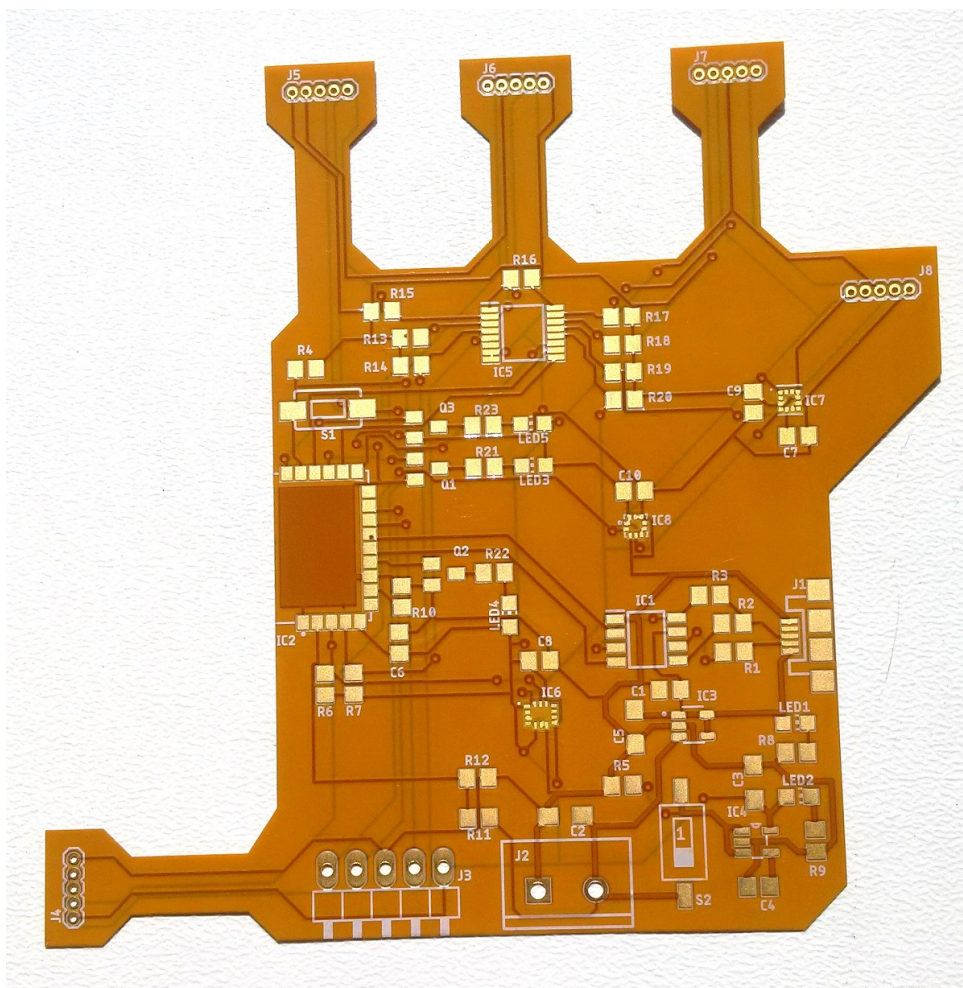
Obrázek C.4: Návrh DPS pro prsteníček



Obrázek C.5: Návrh DPS pro malíček



Obrázek C.6: Návrh DPS pro levou a pravou ruku



Obrázek C.7: Fotografie vyrobené DPS pro pravou ruku

Příloha D

Obsah přiloženého CD

Složka	Popis
EAGLE	Obsahuje kompletní projekt pro program Autodesk EAGLE v9.6.1 se všemi schématy i návrhy DPS.
Gerber	Obsahuje v jednotlivých ZIP souborech data ve formátu Gerber RS-274X pro výrobu všech DPS.
SW_BLE	Obsahuje kompletní projekt pro program Keil μ Vision v5.26.2.0 s kódy pro kompilaci softwaru pro BLE modul BlueNRG-M2SA.
SW_PC	Obsahuje všechny skripty napsané v jazyce Python 3, pro příjem, zpracování a vizualizaci dat na PC (testováno pouze pod Windows 10). Celý program se spouští skriptem <code>start.py</code> .

Tabulka D.1: Obsah složek na přiloženém CD

Příloha E

Bibliografie

1. TEMOCHE, Pablo; RAMIREZ, Esmitt; RODRÍGUEZ, Omaira. A Low-cost Data Glove for Virtual Reality. In: 2012, TCG 31–36.
2. OZ, Cemil; LEU, Ming C. American Sign Language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence*. 2011, roč. 24, č. 7, s. 1204–1213. Dostupné z DOI: [10.1016/j.engappai.2011.06.015](https://doi.org/10.1016/j.engappai.2011.06.015).
3. ZABULIS, Xenophon; BALTZAKIS, Haris; ARGYROS, Antonis. Vision-Based Hand Gesture Recognition for Human-Computer Interaction. 2009. Dostupné z DOI: [10.1201/9781420064995-c34](https://doi.org/10.1201/9781420064995-c34).
4. NIAZMAND, Khalil; TONN, Karin; KALARAS, Anastasios; FIETZEK, Urban; MEHRKENS, Jan-Hinnerk; LUETH, Tim. Quantitative evaluation of Parkinson's disease using sensor based smart glove. In: 2011, s. 1–8. Dostupné z DOI: [10.1109/CBMS.2011.5999113](https://doi.org/10.1109/CBMS.2011.5999113).
5. CONNOLLY, J.; CONDELL, J.; O'FLYNN, B.; SANCHEZ, J. T.; GARDINER, P. IMU Sensor-Based Electronic Goniometric Glove for Clinical Finger Movement Analysis. *IEEE Sensors Journal*. 2018, roč. 18, č. 3, s. 1273–1281. ISSN 2379-9153. Dostupné z DOI: [10.1109/JSEN.2017.2776262](https://doi.org/10.1109/JSEN.2017.2776262).
6. AHMED, Mohamed Aktham; ZAIDAN, Bilal Bahaa; ZAIDAN, Aws Alaa; SALIH, Mahmood Maher; LAKULU, Muhammad Modi bin. A Review on Systems-Based Sensory Gloves for Sign Language Recognition State of the Art between 2007 and 2017. *Sensors*. 2018, roč. 18, č. 7, s. 2208. Dostupné z DOI: [10.3390/s18072208](https://doi.org/10.3390/s18072208).
7. FRANCÉS, Leire; MORER, Paz; RODRIGUEZ, Maria; CAZÓN, Aitor. Design and Development of a Low-Cost Wearable Glove to Track Forces Exerted by Workers in Car Assembly Lines. *Sensors*. 2019, roč. 19, č. 2, s. 296. Dostupné z DOI: [10.3390/s19020296](https://doi.org/10.3390/s19020296).
8. ATALAY, Asli; SANCHEZ, Vanessa; ATALAY, Ozgur; VOGT, Daniel; HAUFE, Florian; WOOD, Robert; WALSH, Conor. Batch Fabrication of Customizable Silicone-Textile Composite Capacitive Strain Sensors for Human Motion Tracking. *Advanced Materials Technologies*. 2017, roč. 2. Dostupné z DOI: [10.1002/admt.201700136](https://doi.org/10.1002/admt.201700136).

9. O'CONNOR, Timothy F.; FACH, Matthew E.; MILLER, Rachel; ROOT, Samuel E.; MERCIER, Patrick P.; LIPOMI, Darren J. The Language of Glove: Wireless gesture decoder with low-power and stretchable hybrid electronics. *PLOS ONE*. 2017, roč. 12, č. 7, e0179766. Dostupné z DOI: [10.1371/journal.pone.0179766](https://doi.org/10.1371/journal.pone.0179766).
10. CHEN, Weiya; YU, Chenchen; TU, Chenyu; LYU, Zehua; TANG, Jing; OU, Shiqi; FU, Yan; XUE, Zhidong. A Survey on Hand Pose Estimation with Wearable Sensors and Computer-Vision-Based Methods. *Sensors*. 2020, roč. 20, č. 4, s. 1074. Dostupné z DOI: [10.3390/s20041074](https://doi.org/10.3390/s20041074).
11. TOWNSEND, Kevin. *Getting started with Bluetooth low energy*. Sebastopol, CA: O'Reilly, 2014. ISBN 978-1491949511.
12. GAITATZIS, Tony. *Bluetooth Low Energy: A Technical Primer*. BackupBrain Publishing, 2017. ISBN 978-1775128083.
13. HEYDON, Robin. *Bluetooth low energy : the developer's handbook*. Upper Saddle River, N.J: Prentice Hall, 2013. ISBN 978-0132888363.
14. MICROCHIP TECHNOLOGY. *MCP73811/2 Datasheet*. Simple, Miniature Single-Cell, Fully Integrated Li-Ion / Li-Polymer Charge Management Controllers. Dostupné také z: <http://ww1.microchip.com/downloads/en/devicedoc/22036b.pdf>.
15. ANALOG DEVICES. *LTC1844 Datasheet*. 150mA, Micropower, Low Noise, VLDO Linear Regulator. Dostupné také z: <https://www.analog.com/media/en/technical-documentation/data-sheets/1844fa.pdf>.
16. BOSCH SENSORTEC. *BMX160 Datasheet*. Small, low power 9-axis sensor. Dostupné také z: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmx160-ds0001.pdf>.
17. STMICROELECTRONICS. *LSM6DSL Datasheet*. iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope. Dostupné také z: <https://www.st.com/resource/en/datasheet/lsm6dsl.pdf>.
18. STMICROELECTRONICS. *LIS2MDL Datasheet*. Digital output magnetic sensor: ultra-low-power, high-performance 3-axis magnetometer. Dostupné také z: <https://www.st.com/resource/en/datasheet/lis2mdl.pdf>.
19. STMICROELECTRONICS. *LPS22HB Datasheet*. MEMS nano pressure sensor: 260-1260 hPa absolute digital output barometer. Dostupné také z: <https://www.st.com/resource/en/datasheet/lps22hb.pdf>.
20. STMICROELECTRONICS. *BlueNRG-M2 Datasheet*. Very low power application processor module for Bluetooth® low energy v5.0. Dostupné také z: <https://www.st.com/resource/en/datasheet/bluenrg-m2.pdf>.
21. U-BLOX. *ANNA-B112 Datasheet*. Stand-alone Bluetooth 5 low energy module. Dostupné také z: https://www.u-blox.com/sites/default/files/ANNA-B112_DataSheet_%28UBX-18011707%29.pdf.

22. PROLIFIC. *PL2303SA Datasheet*. USB to Serial Bridge Controller. Dostupné také z: https://prolificusa.com/app/uploads/2018/02/DS_PL2303SA_d20180327_v1.2.0.pdf.
23. TEXAS INSTRUMENTS. *SN74CB3Q3251 Datasheet*. 1-of-8 FET multiplexer / demultiplexer. Dostupné také z: <http://www.ti.com/lit/ds/symlink/sn74cb3q3251.pdf>.
24. MADGWICK, S. O. H.; HARRISON, A. J. L.; VAIDYANATHAN, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. In: *2011 IEEE International Conference on Rehabilitation Robotics*. 2011, s. 1–7. ISSN 1945-7898. Dostupné z DOI: [10.1109/ICORR.2011.5975346](https://doi.org/10.1109/ICORR.2011.5975346).
25. X-IO TECHNOLOGIES LIMITED. *Arduino library for the BMX160 breakout board with raw sensor data, magnetometer calibration and AHRS examples*. GitHub Repository, 2020. Dostupné také z: <https://github.com/drcpattison/BMX160>.
26. MAHONY, R.; HAMEL, T.; PFLIMLIN, J. Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control*. 2008, roč. 53, č. 5, s. 1203–1218. ISSN 2334-3303. Dostupné z DOI: [10.1109/TAC.2008.923738](https://doi.org/10.1109/TAC.2008.923738).
27. FENG, Kaiqiang; LI, Jie; ZHANG, Xiaoming; SHEN, Chong; BI, Yu; ZHENG, Tao; LIU, Jun. A New Quaternion-Based Kalman Filter for Real-Time Attitude Estimation Using the Two-Step Geometrically-Intuitive Correction Algorithm. *Sensors*. 2017, roč. 17, č. 9, s. 2146. Dostupné z DOI: [10.3390/s17092146](https://doi.org/10.3390/s17092146).
28. GROSSO, Nicholas A. Del; SIROTA, Anton. Ratcave: A 3D graphics python package for cognitive psychology experiments. *Behavior Research Methods*. 2019, roč. 51, č. 5, s. 2085–2093. Dostupné z DOI: [10.3758/s13428-019-01245-x](https://doi.org/10.3758/s13428-019-01245-x).
29. BLIDH, Henrik. *Bluetooth Low Energy platform Agnostic Klient for Python*. GitHub Repository, 2020. Dostupné také z: <https://github.com/hblidh/bleak>.
30. HALE, Zachariah. *Skeleton Hand Model*. 2017. Dostupné také z: <https://sketchfab.com/3d-models/skeleton-hand-a4bf7eabaca24985b3aa2dbfc2d369ca>.