



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**KONSTRUKCE PERA S UTAJENÝM SNÍMÁNÍM**

**BEHAVIORÁLNÍCH VLASTNOSTÍ PODPISU**

CONSTRUCTION OF A PEN WITH HIDDEN ACQUISITION OF BEHAVIORAL SIGNATURE

SAMPLES

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**DOMINIK MÜLLER**

**VEDOUcí PRÁCE**

SUPERVISOR

**prof. Ing. MARTIN DRAHANSKÝ, Ph.D.**

BRNO 2020

## Zadání bakalářské práce



23080

Student: **Müller Dominik**  
Program: Informační technologie  
Název: **Konstrukce pera s utajeným snímáním behaviorálních vlastností podpisu**  
**Construction of a Pen with Hidden Acquisition of Behavioral Signature Samples**

Kategorie: Vestavěné systémy

Zadání:

1. Prostudujte literaturu týkající se snímání a rozpoznávání charakteristik písma a podpisu, zejména behaviorálních (dynamických). Seznamte se s existujícími senzory pro snímání dynamických vlastností pohybů pera a možnými způsoby bezdrátového přenosu dat (zejména Bluetooth).
2. Navrhněte konstrukci a design pera se senzory pro snímání behaviorálních charakteristik písma a podpisu a přenosu těchto dat do PC.
3. Návrh z předchozího bodu prakticky realizujte a otestujte.
4. Shrňte dosažené výsledky a diskutujte možná rozšíření řešení.

Literatura:

- LECH, Michał; CZYŻEWSKI, Andrzej. Handwritten signature verification system employing wireless biometric pen. In: *Intelligent Methods and Big Data in Industrial Applications*. Springer, Cham, 2019. p. 307-319.
- FANCHER, Lee Ann; PRIESTLEY-HOPKINS, Deborah A.; JEFFRIES, Lynn M. Handwriting acquisition and intervention: A systematic review. *Journal of Occupational Therapy, Schools, & Early Intervention*, 2018, 11.4: 454-473.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Drahanský Martin, prof. Ing., Dipl.-Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 14. května 2020

Datum schválení: 31. října 2019

## Abstrakt

Tato práce se zabývá konstrukcí pera pro skryté snímání behaviorálních vlastností podpisu. Navržený systém umožňuje bezdrátové snímání dynamických vlastností podpisu v reálném čase, včetně provedení záznamu. Ke snímání těchto vlastností jsou využity inerciální senzory a tlakový snímač. V rámci práce byla navržena speciální deska plošných spojů, okolo které byl postaven model pera. Pro interakci s perem a prezentaci dat byla vyvinuta počítačová aplikace. Přijatá data jsou dále zpracovávána pro rekonstrukci pohybu pera pisatelem. Ke komunikaci s perem je využita technologie Bluetooth Low Energy.

## Abstract

This thesis deals with the construction of pen for hidden acquisition of behavioral signature samples. The proposed system provides realtime data acquisition with recording possibility. Signature samples are acquired using inertial sensors and pressure sensor. The custom printed circuit board with pen model have been developed. The special application has been developed for pen control. Received data are further processed for pen movement recognition. Bluetooth Low Energy has been used for communication and data transfer.

## Klíčová slova

Bluetooth, Bluetooth Low Energy, inerciální senzory, inerciální měřící jednotka, biometrické pero, akcelerometr, gyroskop, analýza rukopisu, analýza podpisu

## Keywords

Bluetooth, Bluetooth Low Energy, inertial sensors, inertial measurement unit, biometric pen, accelerometer, gyroscope, handwriting analysis, signature analysis

## Citace

MÜLLER, Dominik. *Konstrukce pera s utajeným snímáním behaviorálních vlastností podpisu*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Martin Drahanský, Ph.D.

# Konstrukce pera s utajeným snímáním behaviorálních vlastností podpisu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Ing., dipl.-Ing. Martina Drahanského, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Dominik Müller

4. června 2020

## Poděkování

Děkuji prof. Ing. Martinu Drahanskému, Ph.D., za vedení, konzultace a cenné poznatky při vypracování této práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Rešeršní část</b>	<b>3</b>
2.1	Bluetooth Low Energy . . . . .	3
2.2	Akcelerometr . . . . .	7
2.3	Gyroskop . . . . .	9
2.4	Inerciální měřicí jednotka . . . . .	10
2.5	Odporové tlakové senzory . . . . .	10
2.6	Biometrie . . . . .	11
<b>3</b>	<b>Návrh</b>	<b>12</b>
3.1	Aktuálně využívané technologie . . . . .	12
3.2	Hardware . . . . .	14
3.3	Model pera . . . . .	24
3.4	Firmware . . . . .	25
3.5	Software . . . . .	26
<b>4</b>	<b>Realizace</b>	<b>30</b>
4.1	Firmware . . . . .	30
4.2	Software . . . . .	33
4.3	Deska plošných spojů . . . . .	37
4.4	Model pera . . . . .	38
4.5	Kompletace . . . . .	38
<b>5</b>	<b>Vyhodnocení</b>	<b>39</b>
5.1	Pero . . . . .	39
5.2	Aplikace . . . . .	40
<b>6</b>	<b>Závěr</b>	<b>42</b>
	<b>Literatura</b>	<b>43</b>

# Kapitola 1

## Úvod

Podpis jakožto forma autorizace, autentizace, či projevu vlastnictví je tu s námi již od pradávna a ani s nástupem moderních technologií neztrácí na významu. Prakticky jakýkoli dokument, který má mít nějakou právní váhu, musí být podpisem opatřen. Z hlediska možného zneužití je podpis jako takový velice zranitelný. Podepsaný dokument může projít rukama mnoha lidí než dorazí na své finální místo. Samotný podpis není však nijak vizuálně chráněn. I když jediná možnost jeho verifikace je většinou právě vizuální analýza. Není proto divu, že se podpis stal terčem falšování za účelem podvodu. Možnostmi obrany proti této činnosti se zabývá obor písmonalectví.

Pokud se však oprostíme pouze od vizuální části podpisu, zjistíme, že se podpis skládá i z úkonů provedených při jeho tvorbě. Tato práce si klade za cíl sestavit pero vhodné právě pro zkoumání těchto úkonů. Takovéto pero může pomoci při autentizaci uživatele představením nových překážek, které musí potencionální útočník překonat. Na druhou stranu může být pero zároveň využito také ve prospěch útočníka, který díky němu bude schopen lépe studovat podpis oběti.

Práce je rozdělena do šesti kapitol. Ve 2. kapitole jsou představeny všechny technologie, které jsem musel nastudovat k vypracování této práce. Jmenovitě se jedná o protokol Bluetooth, pomocí kterého je řešena komunikace s perem, fungování použitých senzorů, které mi pomohlo při jejich výběru a jejich použití. Biometrie mi pak objasnila, na jaké vlastnosti podpisu bych se měl zaměřit.

V kapitole 3 jsou představeny aktuálně využívané technologie, včetně popisu principu jejich funkce. Dále se kapitola zaměřuje na můj postup při návrhu pera. Návrh pera je rozdělen do několika částí: výběr a představení použitých součástek, vytvoření desky plošných spojů pro propojení těchto součástek, model plastového obalu pera, který celému výrobku dává jeho finální vzhled a nakonec i návrh firmware a software. Během popisu vybraných částí jsou zmíněny i alternativy, o kterých jsem během výběru uvažoval a také to, proč jsem je nakonec nezvolil.

V kapitole 4 popisují implementaci firmware pro pero a počítačovou aplikaci. Popis obsahuje vysvětlení běhu programu, zvolené technologie a architekturu programu. Popis aplikace dále obsahuje seznámení s uživatelským rozhraním. Kromě programovací části projektu lze v této kapitole nalézt popis osazení desky plošných spojů, včetně oprav, které jsem na ní musel provést. V neposlední řadě zde lze nalézt i popis výroby modelu pera, včetně postupu jeho kompletace.

V kapitole 5 je sestavené pero společně s aplikací otestováno a jsou shrnuty dosažené výsledky.

# Kapitola 2

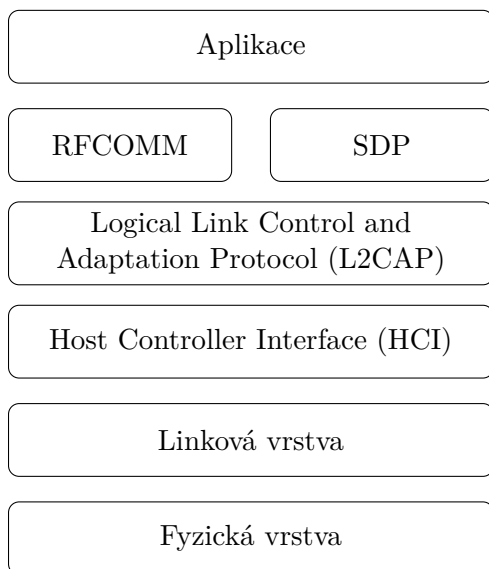
## Rešeršní část

Pro vyhotovení projektu bylo potřeba nastudovat Bluetooth technologii. Ta je použita pro komunikaci mezi perem a počítačem. Bylo také výhodné zjistit princip fungování jednotlivých senzorů, kvůli lepšímu pochopení jejich limitací a použití. Vlastnosti podpisu zkoumá obor Biometrie, bylo tedy příhodné nastudovat i poznatky z tohoto oboru.

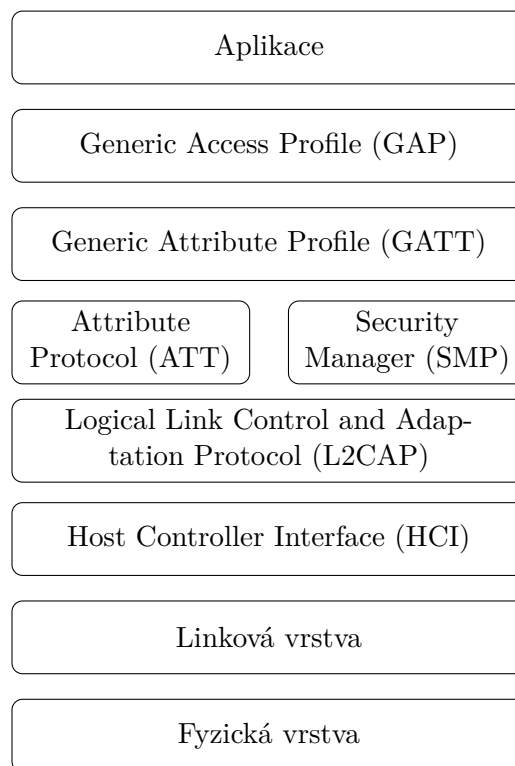
### 2.1 Bluetooth Low Energy

Tato sekce (včetně obrázků) byla převzata z [20], pokud není explicitně uvedeno jinak.

Bezdrátová technologie Bluetooth je krátkovlnný komunikační standard. Jeho klíčové vlastnosti jsou robustnost, nízká spotřeba energie a nízká pořizovací cena. S touto technologií se můžeme setkat ve dvou podobách, a to ve formě Basic Rate (BR) a Low Energy. BLE (Bluetooth Low Energy) vzniklo jako součást Bluetooth 4.0 specifikace. Obě tyto formy mají stejný způsob objevení zařízení, navázání spojení a spojovací mechanismus. [3]



Obrázek 2.1: Architektura Bluetooth BR [3]

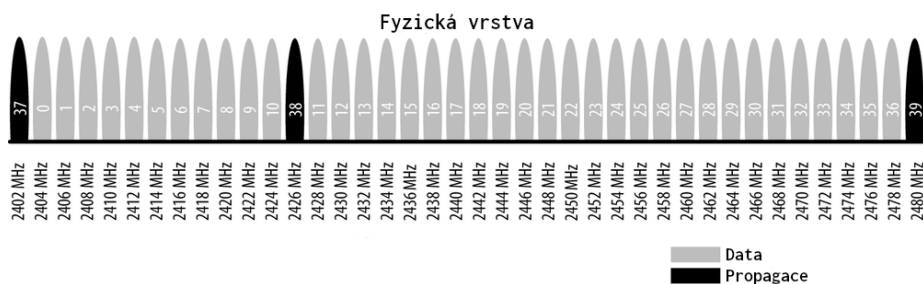


Obrázek 2.2: Architektura BLE [4]

Pro pochopení protokolu BLE je důležité znát jeho jednotlivé vrstvy a jejich funkci. V následujících podsekcích je proto představím.

### 2.1.1 Fyzická vrstva

Fyzická vrstva je část zařízení schopná modulace a demodulace analogového signálu a jeho převodu na digitální podobu. Rádio využívá pro komunikaci 2.4 GHz pásmo rozdělené do 40 kanálů po 2 MHz. Tři z těchto kanálů jsou využity pro navázání spojení a odesílání broadcast zpráv. Zbylé kanály slouží pro komunikaci zařízení.



Obrázek 2.3: Frekvenční pásmo



### 2.1.2 Linková vrstva

Linková vrstva přímo komunikuje s fyzickou vrstvou a obvykle je implementována kombinací speciálního hardware a software. Jedná se o jedinou kritickou část architektury, která je povinná dodržet časování požadované specifikací. Výpočetně náročné a snadno automatizovatelné operace (kontrolní součet, šifrování, generátor náhodných čísel) jsou tak typicky implementovány pomocí aplikačního čipu.

Linková vrstva také obsahuje softwarovou část zajišťující navázání spojení. BLE zařízení může být spojeno v režimu master nebo slave, případně obojí. Zařízení, které zahájí spojení, je automaticky master a zařízení akceptující spojení je slave. Komunikace mezi zařízeními je 1:N. Tedy platí, že master se může připojit k více slave zařízením a slave může být připojen k více master zařízením.

Je důležité, aby linková vrstva byla spolehlivá. Proto jsou všechny zprávy opatřeny 24 bitovým kontrolním součtem. V případě špatného kontrolního součtu je zpráva zahozena a je vyžádána nová.

BLE protokol má pouze jeden formát zprávy, ale rozlišuje dva typy. Jeden typ pro propagaci zařízení a druhý pro data. Propagační zprávy mohou sloužit pro všesměrové vysílání, nebo pro detekci slave zařízení. Každá propagační zpráva se skládá až z 31 bytů dat včetně hlavičky zprávy. Zasílání těchto zpráv pak může probíhat v intervalech od 20 ms až po 10,24 s. Protokol definuje dva základní typy skenování. Pasivní, kdy skenující zařízení zaznamenává propagační zprávy, ale neodpovídá na ně. Aktivní, kdy skenující zařízení zaznamenává propagační zprávy a na každou posílá potvrzení.

Pro rozlišení zařízení je každé z nich identifikováno 48 bitovou adresou, označovanou také jako MAC (media access control) adresa. Tuto adresu lze často vidět ve formátu dvojtečkou oddělených bytů v hexadecimálním formátu (např. `cc:f6:b4:94:ef:36`). [3]

Při navazování spojení odešle master zařízení požadavek o připojení a očekává odpověď od slave zařízení. Požadavek o připojení obsahuje mimo jiné i frekvenční skok. Ten určuje následující sekvenci skoků, kterou budou master i slave dodržovat.

### 2.1.3 Host Controller Interface (HCI)

HCI představuje jednotné rozhraní pro přístup k řadiči Bluetooth. Definuje také formát paketů a pravidla pro přenos dat. BLE specifikace navíc definuje rozšiřující způsoby přenosu pro určité rozhraní. Jedná se například o UART (univerzální asynchronní přijímač-vysílač), USB (univerzální sériová sběrnice), SDIO (Secure Digital Input Output).

### 2.1.4 Logical Link Control and Adaptation Protocol (L2CAP)

L2CAP umožňuje dvě základní funkcionality. První, kde slouží jako multiplexor, který zpracovává protokoly vyšších vrstev a zapouzdří je do standardních BLE paketů a také naopak. Druhou, provádějící skládání a rozkládání paketů z vyšších vrstev s obsahem větším než 27 bytů.

### 2.1.5 Attribute Protocol (ATT)

Jedná se o jednoduchý bezstavový klient-server protokol, založený na attributech, které zařízení prezentuje. BLE zařízení mohou být klient, server, nebo obojí. Nezáleží na tom, zda-li je zařízení master, či slave. Každý server obsahuje data ve formě atributů a každý atribut obsahuje unikátní 16 bitový identifikátor, typ atributu (UUID), množinu pravidel, hodnotu

a délku hodnoty. Typ atributu je určen 128 bitovým číslem. Ovšem pro zkrácení odesílaného paketu jsou typy definované standardem zkráceny na počátečních 16 nebo 32 bitů a zbytek UUID je fixní (Bluetooth Base UUID – *0000-1000-8000-00805F9B34FB*).

Pokud chce klient přečíst nebo zapsat hodnotu atributu, odešle serveru požadavek o přečtení, respektive pro zápis. Server je pak nucen odpovědět hodnotou atributu, nebo potvrzením zápisu. Klient, v případě čtení, musí správně interpretovat přijatá data podle typu atributu. V případě zápisu je klient povinen předat data ve správném formátu, jinak je server může odmítnout.

### 2.1.6 Generic Attribute Profile (GATT)

Vrstva GATT rozšiřuje vrstvu ATT a přidává hierarchii a model pro abstrakci dat. Definuje, jak jsou data organizována a vyměňována. Lze ho tedy požadovat za základ pro transport dat. Zachovává stejnou klient-server architekturu jako vrstva ATT. Zapouzdří však data do služeb, které mohou obsahovat jednu, či více charakteristik.

Služba dodržuje strukturu atributu. Lze ji tedy vnímat jako specializovaný atribut. Její účel je sdružovat charakteristiky do skupin. Typ atributu u služby určuje, jedná-li se o službu primární nebo sekundární. Primární služba je základní typ GATT služby a obsahuje standardní funkcionalitu. Sekundární služba je zamýšlena pro použití uvnitř jiné primární služby a měla by sloužit pouze pro její úpravy. Hodnota atributu u služby pak určuje její UUID.

Každá charakteristika obsahuje minimálně dva atributy. Jeden pro deklaraci charakteristiky a další pro hodnotu charakteristiky. Charakteristika může obsahovat i více deklarácí pro rozšíření metadat charakteristiky. Deklarace charakteristiky je obsažena v hodnotě atributu a skládá se z vlastnosti charakteristiky (čtení, zápis, upozornění, atd.), odkazu na atribut obsahující hodnotu a UUID charakteristiky. Atribut pro hodnotu charakteristiky obsahuje uživatelská data a jeho typ by měl odpovídat UUID charakteristiky uvedené v její deklaraci. Deskriptory charakteristiky rozšiřují informace o charakteristice a její hodnotě. Skládají se vždy z jednoho atributu a spadají do kontextu charakteristiky. Deskriptor může být různého typu. Nejčastěji však obsahuje text popisující charakteristiku (např. Teplota v obývacím pokoji).

	Identifikátor	UUID	Oprávnění	Hodnota
Služba	0x0021	Služba	Čtení	HRS
Charakteristika	0x0024	Charak.	Čtení	N 0x27 HRM
	0x0027	HRM	Čtení	bpm
Descriptor	0x0028	CCCD	Čtení/Zápis	0x0001
Charakteristika	0x002A	Charak.	Čtení	RD 0x2C BSL
	0x002C	BSL	Čtení	prst

Obrázek 2.4: Ukázka služby - monitor tepu (HRS). První charakteristika deklaruje atribut 0x0027 jako naměřenou hodnotu (HRM) v režimu notifikace. Deskriptor 0x0028 v tomto případě slouží jako přepínač pro klienta, pro zapnutí či vypnutí notifikací. Druhá charakteristika deklaruje atribut 0x002C jako polohu senzoru na těle.

### 2.1.7 Generic Access Profile (GAP)

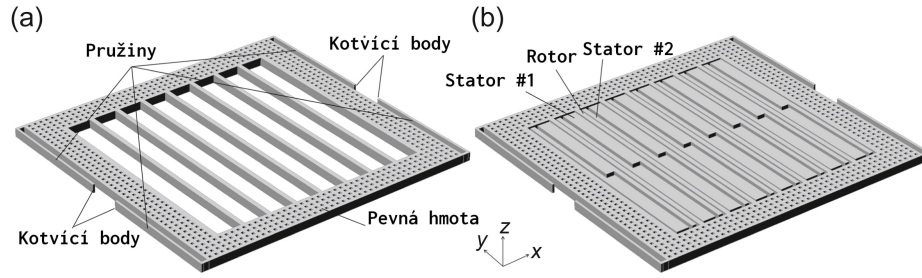
GAP je vrstva umožňující komunikaci BLE zařízení. Představuje rámec, který každé zařízení musí dodržovat, aby mohlo komunikovat s ostatními zařízeními. Zařízení může být provozováno v jedné, či více rolích (Broadcaster, Observer, Central, Peripheral). Každá role představuje určitá omezení a vyžaduje jisté procedury. Ty mají různé požadavky na to, v jaké roli a v jakém stavu musí být zařízení. Pro splnění těchto požadavků může zařízení přepnout svůj stav (Broadcast, Non-discoverable, Any-connectable, atd.). Příklady procedur jsou např. ukončení spojení, pozorování, objevování zařízení atd.

## 2.2 Akcelerometr

Tato sekce (včetně obrázků) byla převzata z [13], pokud není explicitně uvedeno jinak.

Akcelerometry jsou inerciální senzory využívané pro snímání lineární akcelerace.

Zařízení obsahuje pohyblivou část, které je umožněno se pohybovat pouze v jedné ose. To je zajištěno ukotvením pohyblivé části k pouzdru pomocí sady pružin. Pro snímání výchylky pohyblivé části se využívají sady kapacitních paralelních plátů. Ty jsou v jednotlivých buňkách umístěny vždy opačně vůči předchozí buňce.



Obrázek 2.5: Schéma jednoosého akcelerometru. (a) kostra bez elektrod a (b) pohled na plně sestavené zařízení s navzájem opačnými elektrodami

V dnešní době je nejčastěji využívána technologie diferenciálního rozložení buněk. Oproti původním technologiím má hlavní výhodu v poměru citlivosti ku potřebné ploše. Každá buňka tak obsahuje pohyblivou část (rotor) a dvě elektrody připevněné k pouzdru (statory), společně formující dva MEMS kondenzátory  $C_1$  a  $C_2$ .

Kapacita kondenzátorů se řídí rovnicemi

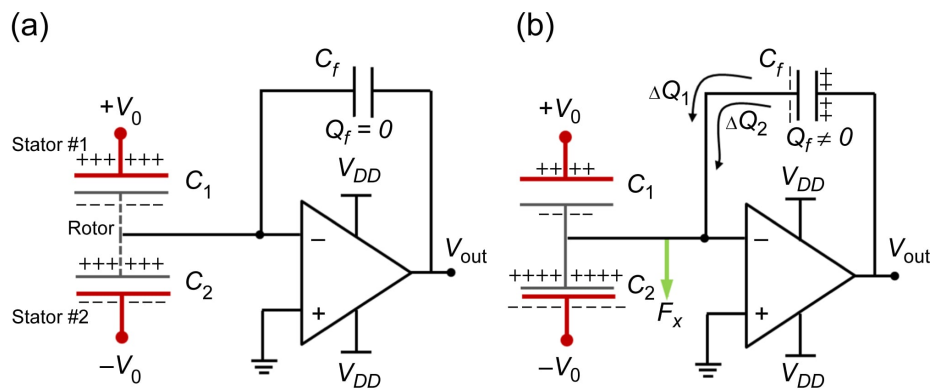
$$C_1 = \frac{\varepsilon_0 A}{(x_0 + x)}; \quad C_2 = \frac{\varepsilon_0 A}{(x_0 - x)}, \quad (2.1)$$

kde  $\varepsilon_0$  je permitivita vakua,  $A$  je čelní plocha elektrod,  $x_0$  je vzdálenost mezi rotorem a statorem v klidovém stavu a  $x$  je výchylka rotoru oproti klidovému stavu. Proměnné  $\varepsilon_0$ ,  $x_0$  a  $A$  lze prohlásit za konstantní. Výslednou kapacitu tak ovlivňuje pouze výchylka pohyblivé části. Tu lze popsat pomocí rovnice

$$m\ddot{x} + b\dot{x} + k_m x = F_{ext}, \quad (2.2)$$

kde  $m$  je hmotnost pohyblivé části,  $b$  je zrychlení,  $k_m$  je pružnost pružin v ose  $x$  a  $F_{ext}$  je suma vnějších sil působící na zařízení.

Změnu kapacity je třeba převést na napěťový signál. Pro představu o principu tohoto měření je model zjednodušen pouze na dvě statické elektrody s jednou pohyblivou elektrodou (viz. obrázek 2.6).



Obrázek 2.6: Schéma měřícího obvodu (a) Stav ve výchozí pozici - nulový náboj (b) Stav po aplikování síly  $F_x$

Ve schématu je znázorněno zapojení nábojového zesilovače. V případě (a) na zařízení nepůsobí žádné vnější síly a zařízení je v klidovém stavu.  $C_1$  a  $C_2$  se tak rovnají a proto,

pokud je dosadíme do rovnic

$$Q_1 = C_1(V_0 - V_m), \quad (2.3)$$

$$Q_2 = C_2(V_0 - V_m), \quad (2.4)$$

výsledný náboj na rotoru je nulový. V případě (b) na zařízení působí vnější síla  $F_x$  způsobující vychýlení rotoru a odpovídající změnu kapacity. Důsledkem je snížení náboje na  $C_1$  a zvýšení náboje na  $C_2$ . Změna náboje vůči výchozí hodnotě je

$$\Delta Q_1 = -V_0 \Delta C_1, \quad (2.5)$$

$$\Delta Q_2 = +V_0 \Delta C_2. \quad (2.6)$$

Výsledný náboj na vstupu operačního zesilovače pak odpovídá

$$\Delta Q_{tot} = \Delta Q_1 + \Delta Q_2 = V_0(\Delta C_2 - \Delta C_1) = V_0 \Delta C = 2V_0 \frac{C_0}{x_0} x \quad (2.7)$$

a způsobuje úměrnou změnu napětí na výstupu operačního zesilovače podle rovnice

$$V_{out} = \frac{\Delta Q_{tot}}{C_f} = 2V_0 \frac{C_0 x}{C_f x_0}. \quad (2.8)$$

Výsledné rozhraní akcelerometru tak může být

1. Analogové, kde může být výstupní signál z nábojového zesilovače dále upravován (pro větší stabilitu či přijatelnější hodnoty). Toto rozhraní je výhodné v případě čistě analogového zapojení, nebo v případě vlastního převodu z analogové na digitální reprezentaci.
2. Digitální, kde je převod z analogové reprezentace na digitální řešen výrobcem. Výrobce často využije různé způsoby filtrování pro zvýšení stability, nebo představí možnosti využití front a s tím souvisejících přerušení. Výstupní rozhraní je pak nejčastěji pomocí SPI/I2C.

## 2.3 Gyroskop

Tato sekce (včetně obrázků) byla převzata z [13], pokud není explicitně uvedeno jinak.

Gyroskop slouží pro měření úhlové rychlosti okolo dané osy. Tak jako u jiných gyroskopů i měření MEMS gyroskopu vychází z fyzikálního principu Coriolisovy síly. To znamená, že objekt pohybující se rychlostí  $v$  a úhlovou rychlostí  $\Omega$  okolo osy kolmé k vektoru  $\vec{v}$ , je objektem Coriolisovy akcelerace:

$$a_{Cor} = 2\Omega \times v. \quad (2.9)$$

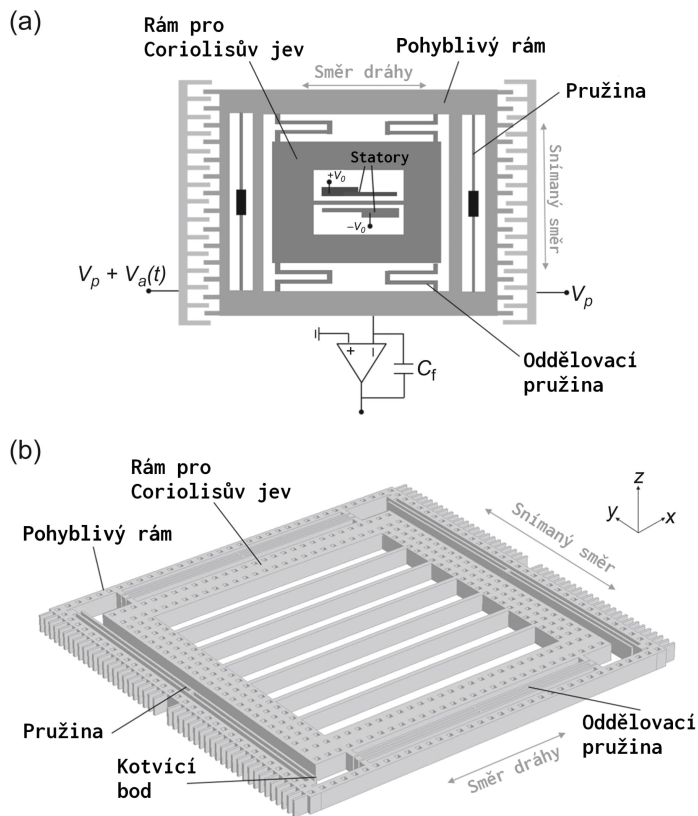
Odpovídající Coriolisova síla působí také na objekt směřující kolmo vůči ose rotace a vůči směru akcelerace a to s modulem

$$F_{Cor} = 2m\Omega v. \quad (2.10)$$

Je proto důležité zajistit, aby se Coriolisova síla projevovala pouze za přítomnosti rychlosti  $v$ . Toho se v gyroskopu docílí zabudováním sekce pro aplikování známé rychlosti na zavěšenou část.

Na obrázku 2.7 je příklad vnitřní struktury MEMS gyroskopu. Gyroskop je sestaven z vnějšího rámu uchyceného tak, aby se mohl volně pohybovat po ose  $x$  a zamezil pohybu

po ose  $y$ . Spolu s dalšími uchycenými částmi tvoří gyroskop i dvě řady hřebenových elektrod k vyvinutí rychlosti  $v$ . Skupina pružin spojuje vnější rám s vnitřním (také pohyblivým rámem) tak, aby vnější rám vedl rám vnitřní po ose  $x$ , ale zároveň umožnil volný chod v ose  $y$ .



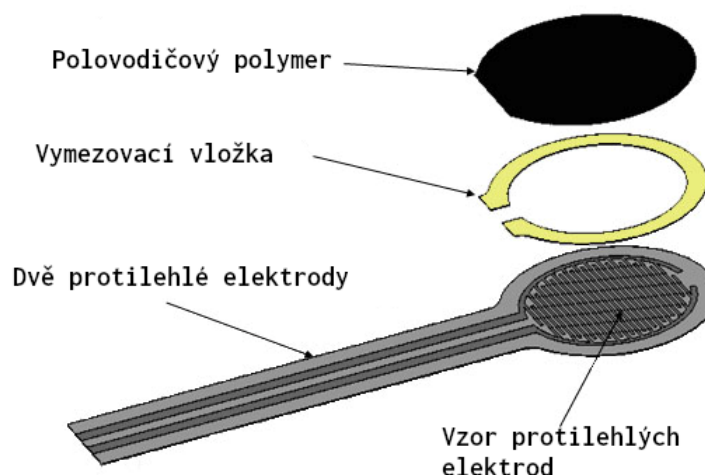
Obrázek 2.7: Průřez MEMS gyroskopu

## 2.4 Inerciální měřicí jednotka

Inerciální měřicí jednotka (Inertial Measuring Unit - IMU) je elektronické zařízení, skládající se z různého množství a typů senzorů určených pro měření pohybu a orientace. Všechny tyto senzory se nacházejí ve společném pouzdře a často využívají i společné rozhraní. Nejčastěji využívaná je kombinace akcelerometru a gyroskopu. Lze však také najít jednotky i s magnetometrem pro detekci orientace, či jednotky se senzorem atmosférického tlaku. Mezi výhody sjednocení senzorů do jednoho pouzdra patří nejen zredukování potřebných součástí, ale hlavně zaručené rozložení senzorů výrobcem.

## 2.5 Odporové tlakové senzory

Odporový tlakový snímač je zařízení reagující snížením svého odporu při zvýšení síly aplikované na jeho povrch.



Obrázek 2.8: Ukázka konstrukce senzoru [1]

Typická konstrukce se skládá ze dvou polymerových plátů. Na první plát jsou naneseny dvě protilehlé elektrody. Na druhý je nanesena vrstva polovodičového polymeru. Pláty jsou poté k sobě přiloženy upravenými stranami. [23]

Pokud na snímač nepůsobí žádná síla, je jeho odpor v řádu mega ohmů. S přibývajícím silou působící na snímač jeho odpor klesá. Pokud síla přesáhne určitý bod, odpor se začne ustalovat. A to až do bodu, kdy dosáhne saturace a odpor zůstává vesměs konstantní. [23]

## 2.6 Biometrie

Biometrie je automatické rozpoznávání lidí na základě jejich charakteristických anatomických rysů (např. obličej, otisk prstu, duhovka, sítnice) a charakteristického chování (např. podpis, chůze). [6]

Tento obor se, mimo jiné, zabývá i rozpoznáváním podpisu. Zkoumá statické a dynamické vlastnosti provedené během podepisování. Statické vlastnosti podpisu jsou vlastnosti určující vzhled podpisu. Dynamické vlastnosti podpisu jsou ty vlastnosti, které popisují, jak byl podpis vytvořen. K popisu těchto vlastností využíváme následující pojmy:

- **Stroky** – místa mezi zvednutím a položením pera
- **Kritické body** – body nesoucí více informace než ostatní body (koncové body, stroky a body změny trajektorie) [5]

Do dynamických vlastností, které lze v průběhu podpisu zkoumat patří:

- X-funkce
- Y-funkce
- Přítlak (Z-funkce)
- Rotace pera během podpisu
- Úchop pera pisatelem

# Kapitola 3

## Návrh

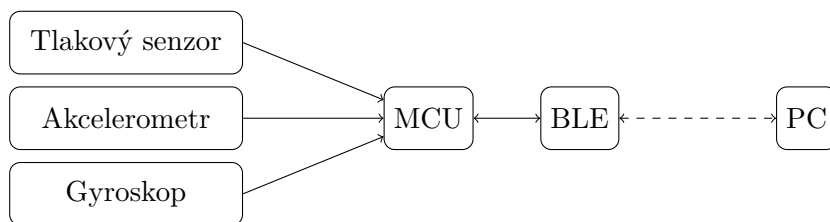
Při návrhu pera jsem se rozhodoval, jaké dynamické vlastnosti podpisu bude pero snímat. Rozhodl jsem se pro X-funkci a Y-funkci, které umožní rekonstrukci podpisu. Dále pro přítlak, který bude, mimo jiné, sloužit i pro detekci začátku podpisu. A nakonec i pro rotaci pera a jeho úhel náklonu během podpisu.

Následně jsem prozkoumal hotová řešení zabývající se podobným tématem, abych zjistil, jak k tomuto problému přistupovali ostatní. Z průzkumu hotových řešení jsem došel k závěru, že jediná používaná technologie umožňující požadované vlastnosti a skryté použití, je technologie inerciálních senzorů.

Pro přenos dat mezi perem a počítačem jsem zvolil technologii BLE. Podobně jako Wi-Fi nabízí univerzálnost a velké rozšíření napříč zařízeními. Vyžaduje však méně energie pro provoz. BLE oproti Bluetooth BR jsem zvolil zejména díky nižší spotřebě energie. Ostatní bezdrátové technologie jsem zavrhl kvůli jejich řídké integraci v zařízeních a tedy nutnosti využívat externí přijímač.

O zpracování a prezentaci dat se stará aplikace běžící na počítači. Aplikace zařizuje připojení k peru, umožňuje jeho nastavení a přijímá data, která zařízení propaguje. Přijatá data následně prezentuje uživateli.

Výsledný systém se tedy skládá ze dvou částí a to z pera a počítače. Jelikož je pero bateriové zařízení, stará se pouze o záznam dat. Jakékoliv náročnější operace nad daty jsou nežádoucí z důvodu úspory energie. S obdrženými daty pracuje až počítač, který je i prezentuje uživateli.



Obrázek 3.1: Blokové schéma systému

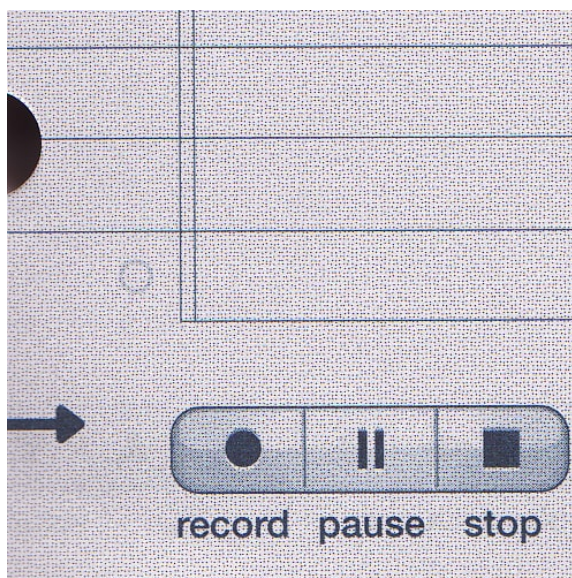
### 3.1 Aktuálně využívané technologie

Během průzkumu technologií aktuálně využívaných ke snímání psaného textu jsem objevil tyto technologie:



- **Infračervená kamera**

Zaznamenávání pohybu pera je realizováno pomocí vysokorychlostní infračervené kamery a speciálně potištěného papíru. Papír je potištěn takovým vzorem, který pomocí kamery umožní rozhodnout, v jakém směru se pero pohybuje a tím tak zpětně rekonstruovat pohyb pera. Speciální potisk papíru také umožňuje vytvoření specifických znaků, kterým lze přiřadit speciální význam. (např. tlačítko pro nahrávání viz. obrázek 3.2)



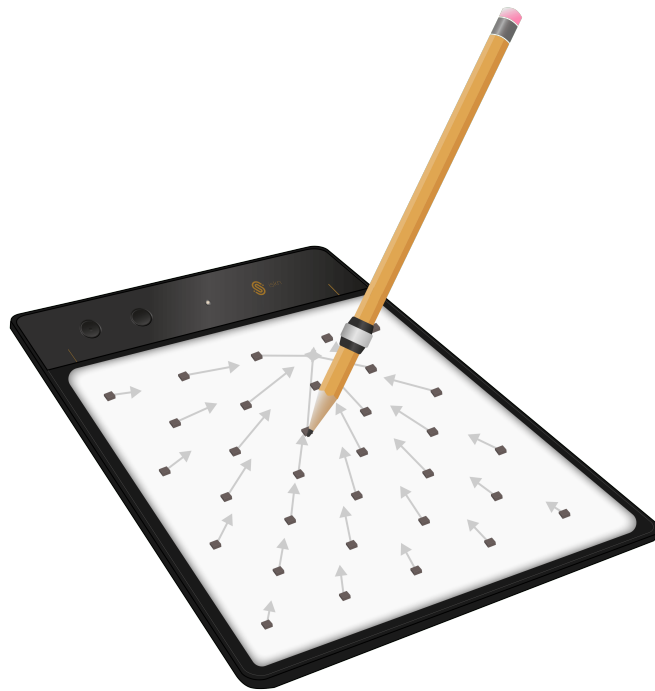
Obrázek 3.2: Ukázka speciálního papíru [18]

- **Inerciální senzory**

Záznam realizovaný pomocí inerciálních senzorů je využíván často ve vědeckých pracích [7] [22] [12] [9] [14], zabývajících se podobnou tematikou jako tato práce. Tyto práce často využívají kombinaci akcelerometru a gyroskopu umístěného v místě úchopu pera. Prezентují také dobré výsledky při rozpoznávání napsaných znaků (obrazců). V komerčních produktech jsou tyto senzory využívány minimálně. Jako jednu z výjimek lze uvést novou verzi S Pen pro telefon Galaxy Note 10 od korejského výrobce Samsung. Ovšem využití senzorů je zde zaměřené primárně na detekci gest pro systémové akce.

- **Hallova sonda**

Zařízení založená na hallových sondách využívají pro rekonstrukci pohybu pera pole těchto sond. Pole sond se nejčastěji implementuje jako podložka pro psaní a magnet uchycený k peru. Magnet je uchycen k peru takovým způsobem, aby byly sondy schopny detekovat jeho magnetické pole. Poloha magnetu je kritická, jelikož sondy detekují jeho sílu a směr. Pomocí těchto získaných informací lze pak zpětně rekonstruovat pohyb pera, včetně jeho natočení vůči podložce.



Obrázek 3.3: Ukázka podložky s hallovými sondami [8]

- **Tablet / Podložka**

Pro snímání pohybu a přítlaku pera je využita speciální podložka a také (ne nutně) pomocná elektronika v peru. Snímání může být provedeno různými technologiemi např. Wacom AES, Wacom EMR, N-Trig a dalšími. Tyto technologie mohou pro měření využívat různé principy. Ovšem cíl mají stejný. A sice obdržet pozici hrotu pera na podložce a některé i zjistit míru přítlaku pera na podložku.

Z průzkumu hotových řešení se mi jako nejlepší jeví inerciální senzory. Jejich výhodou je snadné skrytí senzorů uvnitř pera, široký výběr senzorů na trhu a jejich malá velikost. Ostatní technologie vyžadují externí části pro správnou funkčnost, nebo nutnost viditelně odhalit senzor.

## 3.2 Hardware

### 3.2.1 Inerciální senzory

Senzory, které budu v systému používat, jsou tříosý akcelerometr a tříosý gyroskop. Akcelerometr bude snímat X-funkci a Y-funkci. Zároveň bude kompenzovat měření gyroskopu a tím zvyšovat přesnost odhadu natočení pera.

Během podpisu se míra akcelerace pohybuje od -1 G do 1 G a úhlová rychlost od -300 °/s do 300 °/s. Pro akcelerometr tedy lze očekávat hodnoty maximálně v mezích od -2 G do 2 G, pokud uvažujeme gravitaci. [19]

Při výběru jsem vyhledával senzory s velkou přesností, rychlým vzorkováním a malými rozměry.

Jako tříosý akcelerometr mi nejlépe vyšel čip FXLS8471Q<sup>1</sup> od společnosti NXP. Čip měří hodnoty pomocí 14 bit ADC a nabízí volitelný rozsah 2 G, 4 G nebo 8 G, se vzorkováním od 1.56 Hz do 800 Hz. Ve své kategorii má i dobrou odolnost proti nízkofrekvenčnímu zkreslení. Obsahuje také pomocné periférie pro detekci akcí (např. detekce volného pádu, detekce poklepání), ale ty nebudou využity.

Jako tříosý gyroskop jsem zvolil čip BMG250<sup>2</sup>. Čip nabízí 16 bitové hodnoty a volitelný rozsah od 125 °/s do 2000 °/s se vzorkováním od 25 Hz do 3200 Hz. Výhodou je již hotová knihovna přímo od výrobce.

Pro čip FXLS8471Q existuje alternativa FXOS8700CQ<sup>3</sup>, která obsahuje stejný tříosý akcelerometr a navíc tříosý magnetometr. Využití magnetometru pro inerciální senzory slouží pro kompenzaci dlouhotrvající odchylky. V případě krátkodobé události, jako je podpis, tak nemá význam a neplánuji tuto možnost využít.

Pro čip BMG250 existuje alternativa BMI270<sup>4</sup>. BMI270 obsahuje 16 bitový akcelerometr a 16 bitový gyroskop. Tímto čipem tedy lze nahradit akcelerometr FXLS8471Q, nebo doplnit pero o druhý akcelerometr. Tato alternativa také nabízí možnost porovnání výsledků při použití akcelerometru a gyroskopu odděleně, akcelerometru a gyroskopu v jednom pouzdře a použití dvou akcelerometrů s jedním gyroskopem.

### 3.2.2 Tlakový senzor

Pro měření přítlaku jsem se rozhodoval mezi piezoelektrickým a odporovým měřením. Z důvodu jednoduchosti obvodu a nedostupnosti piezoelektrického senzoru v potřebné velikosti, jsem se rozhodl pro odporový tlakový snímač.

Trh s odporovými tlakovými senzory v rozměrech pod 1 cm je velice malý a prakticky se dají pořídit pouze senzory od výrobce Ohmite. Vybral jsem tedy nejmenší variantu jejich FSR řady o průměru 7.62 mm, FSR04. Sensor funguje v rozsahu od 20 g do 5 kg a jeho charakteristika by měla odpovídat grafu 3.4.

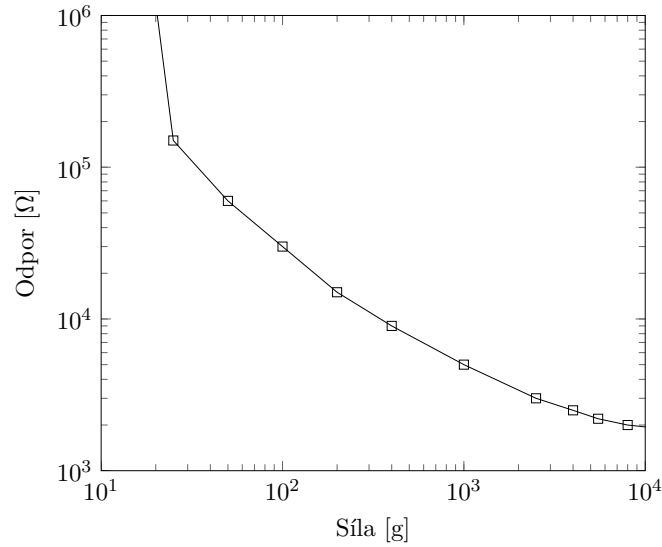
---

<sup>1</sup>FXLS8471Q - <https://www.nxp.com/docs/en/data-sheet/FXLS8471Q.pdf>

<sup>2</sup>BMG250 - <https://www.mouser.com/datasheet/2/783/BST-BMG250-DS000-02-967971.pdf>

<sup>3</sup>FXOS8700CQ - <http://www.farnell.com/datasheets/2295099.pdf>

<sup>4</sup>BMI270 - <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi270-ds000-2.pdf>



Obrázek 3.4: Typická FSR charakteristika [23]

### 3.2.3 Umístění senzorů

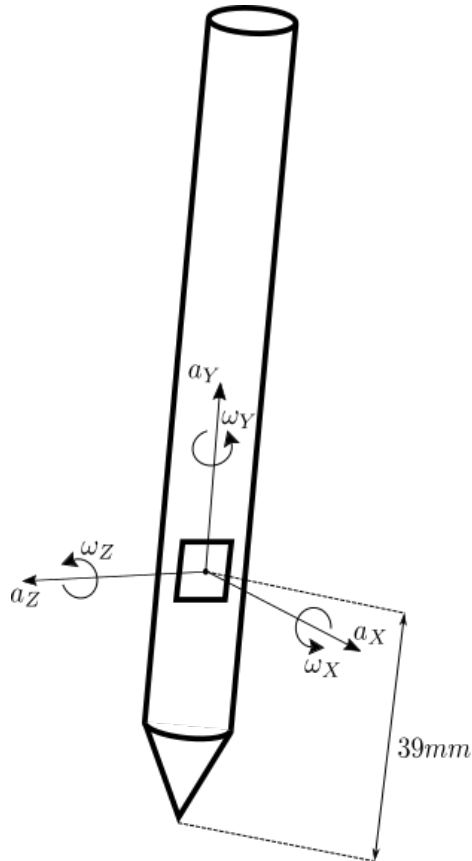
Při práci s inerciálními senzory je důležité promyslet jejich umístění v zařízení. V případě, kdy zařízení nerotuje, je umístění senzoru nepodstatné. Ovšem, pokud zařízení může rotovat, tato rotace ovlivní měření akcelerometru.

Míru ovlivnění měření uvádí rovnice

$$A_r = \omega * (\omega * r), \quad (3.1)$$

kde  $\omega$  je úhlová rychlost zařízení a  $r$  je vzdálenost akcelerometru od těžiště. Proto je potřeba umístit senzor do těžiště zařízení (kde se také nachází centrum otáčení), nebo matematicky kompenzovat pomocí úhlové rychlosti. [17]

V mém případě, kdy uživatel mění těžiště pera podle toho, jak pero uchopí, je jeho vhodné umístění nereálné. Pokusil jsem se však ovlivnění akcelerometru minimalizovat umístěním senzoru do místa vyhrazeného pro úchop pera. Pro co nejvyšší přesnost akcelerometru je tedy potřeba kompenzovat vychýlení pomocí software.



Obrázek 3.5: Ukázka umístění senzoru

### 3.2.4 Bluetooth modul

Existují dva způsoby, jak lze BLE implementovat. První způsob je samotný čip a vlastní návrh radiové i všech ostatních potřebných částí. Druhý způsob je sestavený modul. Sestavený modul má již všechny potřebné části osazené a odladěné výrobcem. Pro jednoduchost tak budu vybírat pouze ze sestavených modulů. Bluetooth moduly lze najít ve dvou provedeních.

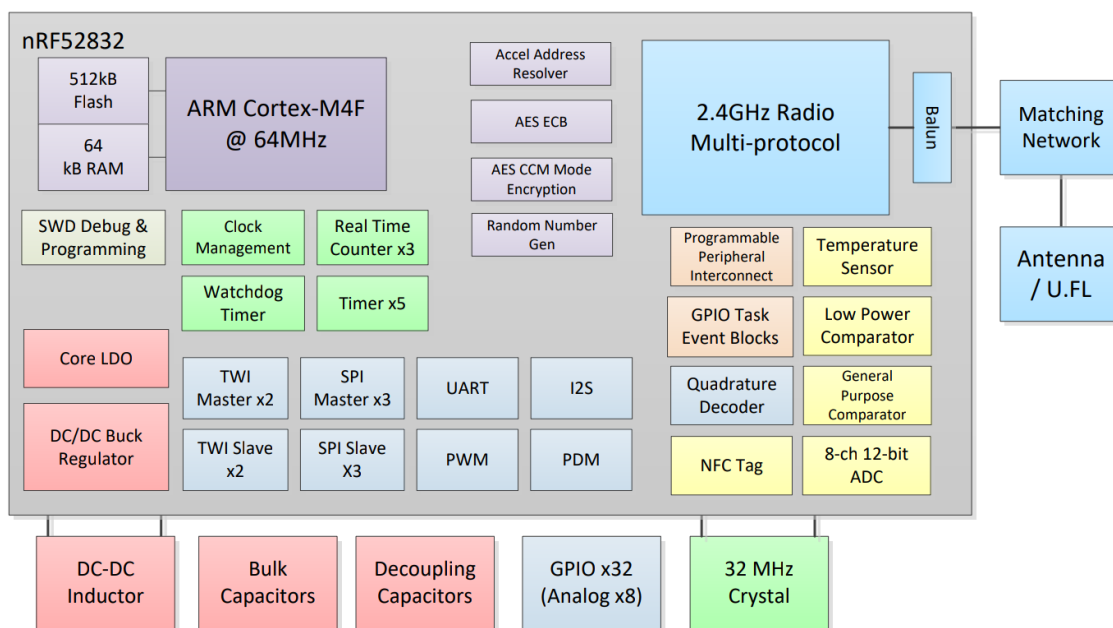
Prvním provedením je mikrokontrolér fyzicky oddělený od BLE modulu a komunikující mezi sebou přes určité rozhraní (I2C, UART, SPI, aj.). Komunikace je většinou postavena na určitém protokolu na bázi příkaz/odpověď (ACI, AT, aj.). Použité rozhraní a způsob komunikace je ale daný výrobcem. Pomocí protokolu lze nastavit modul samotný nebo odesílat/přijímat data. Veškerá komunikace a BLE operace tak nezatěžují hlavní mikrokontrolér a ten může v mezičase vykonávat jiné instrukce.

V druhém provedení mikrokontrolér obsluhuje jak klientskou aplikaci, tak i rádio a obsluhu pro BLE. Je tak potřeba počítat s pamětí flash a RAM, kterou zabere kód pro obsluhu BLE a řízení běhu aplikace. Jeho velikost není zanedbatelná, jelikož se může pohybovat až v řádu stovek kilobytů. Architektura této implementace vyžaduje kontrolér, který řídí běh aplikace a přepíná kontext mezi klientskou aplikací a obsluhou BLE. Obvykle tak výrobce (pro usnadnění vývoje) implementuje kontrolér, obsluhu BLE a aplikační rozhraní pro řízení BLE z uživatelské aplikace. Při vývoji uživatelské aplikace je nutné brát zřetel na možné přepnutí kontextu aplikace a také obtížné ladění aplikace.

První provedení je ideální pro aplikace, kde je mikroprocesor hodně využíván klientskou aplikací a nestíhal by tak obsluhovat BLE. V mém případě se jedná pouze o pravidelné čtení hodnot ze senzorů. Zátěž mikroprocesoru je tak minimální. Ačkoli je první provedení jednodušší na implementaci, vyžaduje fyzicky větší prostor a také více energie. Z těchto důvodů jsem upřednostnil druhé provedení modulu.

Hlavními kritérii při výběru modulu byly velká paměť pro záznam dat, malá velikost modulu, existence evaluační desky a dobrá knihovna pro práci s modulem. Všechny tyto body splňuje modul od firmy Rigado (nyní u-blox) BMD-350.

## BMD-300 Series Modules



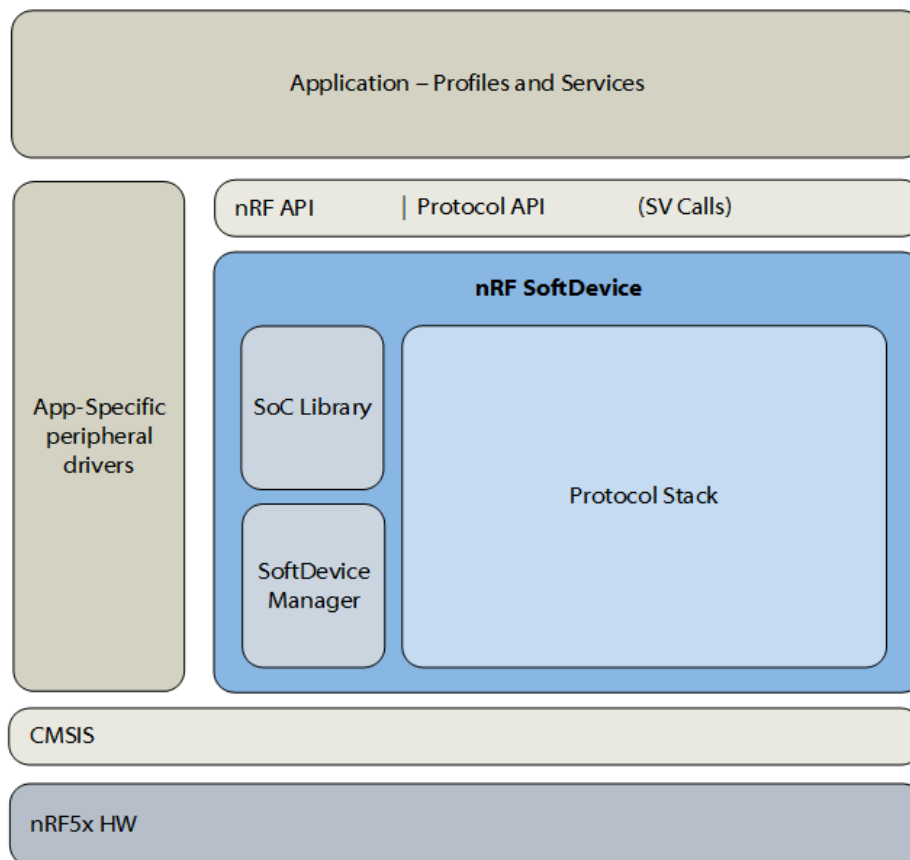
Obrázek 3.6: Blokové schéma modulu [21]

Se svými velmi malými rozměry 6.4 mm x 8.65 mm umožní snížení průměru pera na minimum. Modul je postavený okolo čipu NRF523832 od firmy Nordic Semiconductors, jedná se o rozšířenou verzi mikroprocesoru ARM Cortex-M4. Tento čip nabízí úctyhodných 512 kB paměti Flash a 64 kB paměti RAM. Podporuje nejnovější specifikaci Bluetooth 5.2 a BLE. Za vyzdvihnutí stojí také obsažené programovatelné rozhraní periférií, které umožňuje jakoukoliv periférii (SPI, I2C, aj.) přiřadit na jakýkoliv pin. Umožňuje také přiřazení více periférií na jednom pinu, ovšem neručí za konzistentnost. Tato skutečnost tak velice usnadní rozložení desky plošných spojů a přispěje k celkové minimalizaci rozměrů.

Pro práci s modulem má firma Nordic Semiconductors připravené SDK (Software development kit – vývojářský balíček) s rozsáhlou online dokumentací a příklady různých implementací, ze kterých lze vycházet. SDK obsahuje také binární soubory s různými variantami SoftDevice.

SoftDevice je kód obsluhující bluetooth. Stará se o přepínání kontextu a poskytuje aplikační rozhraní pro uživatelskou aplikaci. Varianty SoftDevice jsou značené  $Sxyz$ , kde  $x$  značí implementovaný protokol (BLE, ANT, BLE a ANT),  $y$  roli zařízení (Periférie,

Centrála, atd.) a z skupinu podporovaných čipů. Pro můj účel jsem tak vybral S112, které využívá protokol BLE v roli periférie.



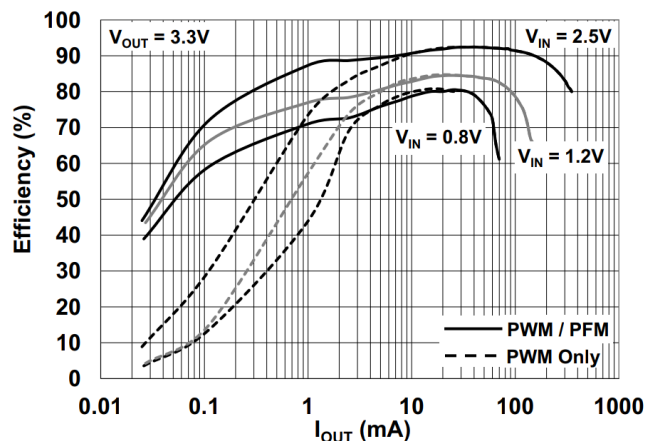
Obrázek 3.7: Ukázka architektury SoftDevice [16]

### 3.2.5 Napájecí část

O napájení pera se bude starat baterie s regulátorem pro úpravu napětí. Vzhledem k požadavku na nenápadnost zařízení neobsahuje nabíjecí konektor. Proto jsem zařadil do výběru i nenabíjecí baterie.

Z důvodu omezeného prostoru pouzdra pera jsem výběr baterie zúžil na baterie alkalické a baterie na bázi lithia. Lithiové baterie jsou v takto malých rozměrech těžko dostupné a mají téměř dvakrát menší kapacitu než baterie alkalické. Alkalické baterie existují ve standardizovaném balení AAAA, které se často využívá v grafických perech či stylusu a mají velice dobrou kapacitu. Jejich nominální napětí je 1.5 V, což je pro napájení pera, které potřebuje napětí 3.3 V, nedostačující. Proto je přidán měnič step-up, který napětí zvýší na potřebnou úroveň.

Jako step-up měnič byl vybrán čip MCP1640. Je to čip určený přímo pro zvyšování nízkého napětí baterií. Jeho výhodou je schopnost zvyšovat napětí už od 0.35 V. Umožní tak využít veškerou energii baterie. Jeho efektivita je přitom obstojná, špatnou efektivitu má pouze při velmi malém odběru proudu.

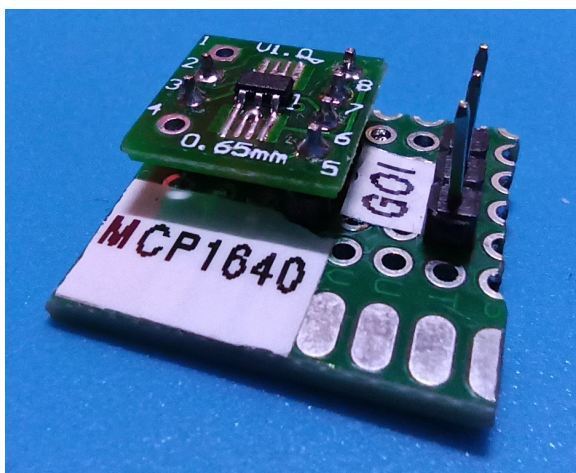


Obrázek 3.8: Efektivita měniče za různých podmínek při výstupním napětí 3.3 V [2]

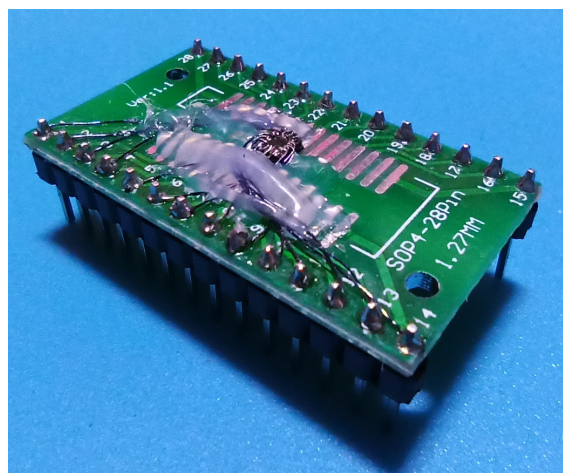
### 3.2.6 Ověření schéma

Po výběru všech potřebných součástek jsem vytvořil elektronické schéma (viz. obrázek 3.12). Pro kontrolu jsem podle něho zapojil obvod na nepájivé pole. Jelikož jsou všechny součástky typu SMD (surface mount device), bylo nutné využít adaptéry pro připojení k nepájivému poli. Pro bluetooth modul jsem využil evaluační desku, ostatní součástky byly napájeny na pomocné desky.

Pro test napájecí části jsem zakoupil i druhou verzi čipu MCP1640 (vhodnější pro ruční pájení). Po napájení jsem zdroj otestoval pomocí laboratorního zdroje a elektronické zátěže. Během testování jsem ověřil funkčnost měniče, jeho efektivitu a také jeho chování během napětí pod 1 V.



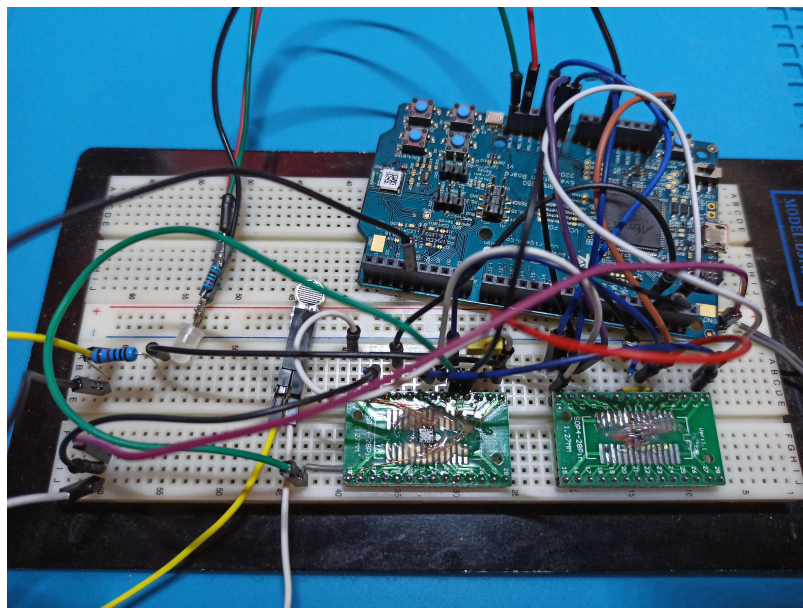
Obrázek 3.9: Testovací přípravek pro napájecí část



Obrázek 3.10: Adaptér pro gyroskop a akcelerometr

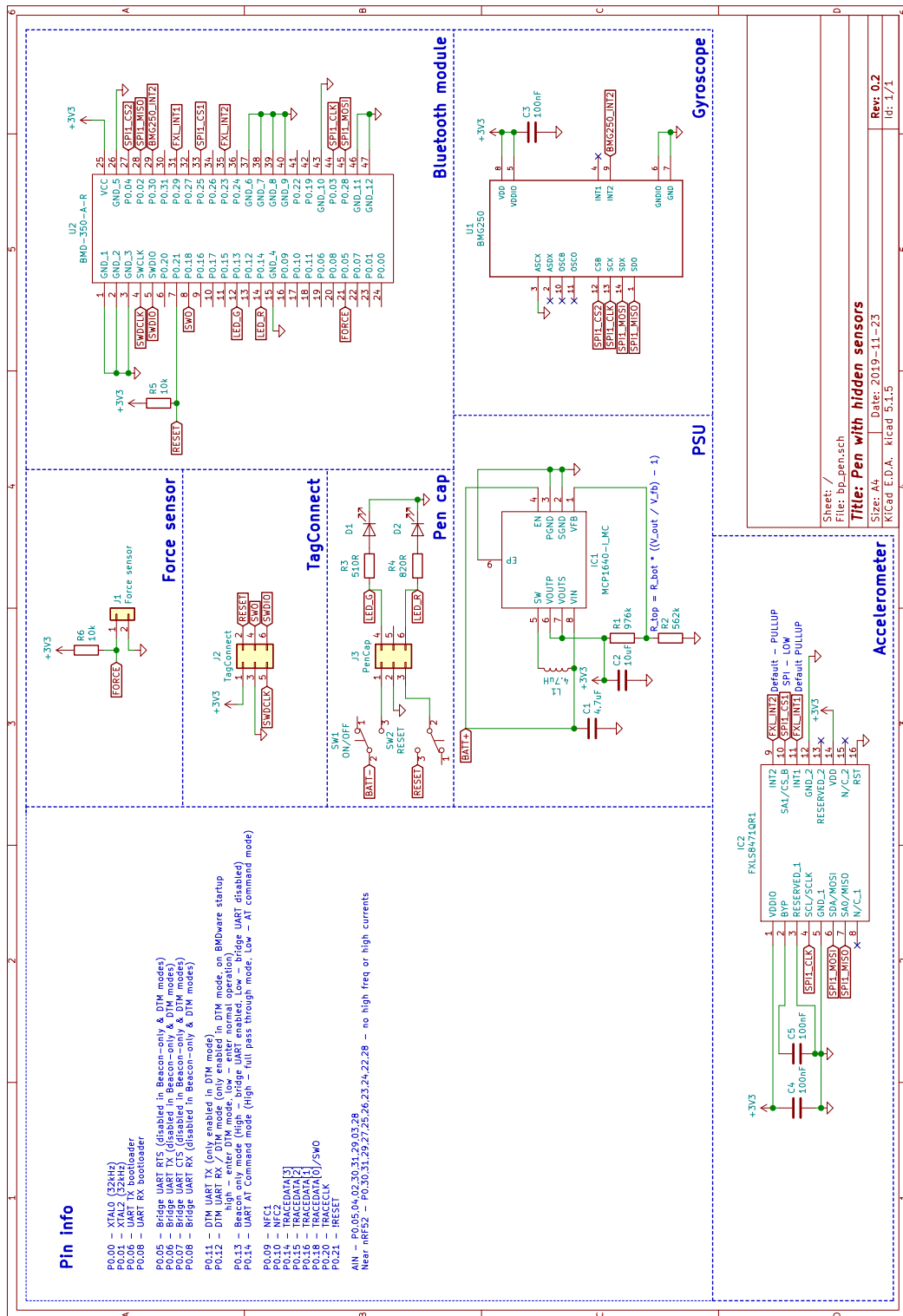
Pro akcelerometr a gyroskop jsem neměl potřebný adaptér. Otočil jsem tedy čip vzhůru nohama, zespodu přilepil vteřinovým lepidlem a pod mikroskopem, s pomocí lakovaných drátků, čip napojil. Pro ochranu spoje jsem ho izoloval pomocí tavné pistole.





Obrázek 3.11: Zapojení na nepájivém poli

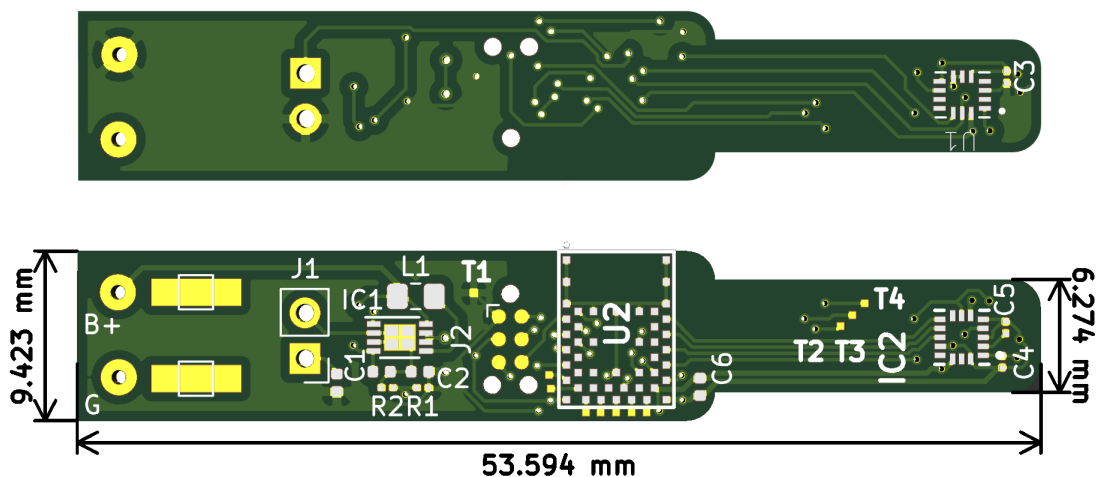
Po zapojení všech částí jsem ověřil, zda jsem s nimi schopen navázat komunikaci a zkontroloval jsem správnost schéma. Během ověřování vykazoval akcelerometr FXLS8471Q nekonzistentní chování. Akcelerometr má automatickou detekci rozhraní podle stavu jednoho pinu. Pokud je pin ve stavu vysoké impedance, vybere rozhraní SPI, jinak I2C. Navzdory tomu, že pin měl být ve stavu vysoké impedance, vybral čip rozhraní I2C a komunikace se senzorem selhala. Toto chování bylo náhodné a ne až tak časté. Přisoudil jsem ho tedy parazitním jevům zapříčiněným použitím nepájivého pole. Tento problém jsem více neřešil a považoval schéma za korektní.



Obrázek 3.12: Finální schéma pera

### 3.2.7 Deska plošných spojů

Desku plošných spojů (PCB) jsem pro minimalizaci rozměrů navrhl na míru v programu KiCad.



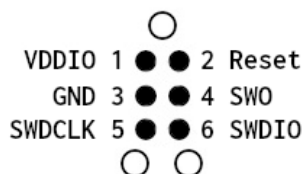
Obrázek 3.13: Ukázka spodní a vrchní částí desky odeslané do výroby

Při návrhu jsem umístil bluetooth modul na výšku (U2), což zapříčinilo celkové rozšíření desky, ale umožnilo snadnější vedení cest. K umístění na výšku mě vedl i požadavek z referenčního manuálu na absenci měděné vrstvy v oblasti antény.

Inerciální senzory (IC2) jsou umístěny na pravém konci desky. Senzory komunikují pomocí společné sběrnice SPI. To zmenšilo počet potřebných cest a umožnilo více zúžit část za bluetooth modulem a také umožnilo umístění tří testovacích bodů (T2-T4).

Další testovací body jsou umístěny pod a vlevo od bluetooth modulu. Testovací body se využívají k pomoci při ladění aplikace, slouží jako pojistka v případě chyby při návrhu desky, nebo pro rozšíření funkcionality, s kterou se při návrhu nepočítalo. Výjimkou je testovací bod T1, který slouží pro měření výstupního napětí z měniče.

Vlevo od bluetooth modulu se nachází konektor Tag-Connect pro programování a ladění desky pomocí protokolu SWD. Volba tohoto konektoru byla nezbytná, protože ostatní konektory jsou příliš vysoké. Nepodařilo se mi však nalézt jeho referenční zapojení. Je tak možné, že pro připojení konektoru bude nutná jeho úprava.



Obrázek 3.14: Zapojení Tag-Connect na desce pera

Zapojení měniče step-up (IC1) je podle referenčního manuálu.

Vlevo od měniče je umístěn tlakový senzor. Více vlevo jsou dvě měděné plošky pro napájení opory senzoru.

Přívod energie pro desku značí body B+ (pro kladný pól baterie) a G (pro záporný pól). Napětí přivedené na desku může být v rozmezí od 0.8 V do 3.6 V (při startu desky) a napětí může klesnout až na 0.35 V. Deska nemá ochranu proti přepólování, ani proti přepětí.

### 3.3 Model pera

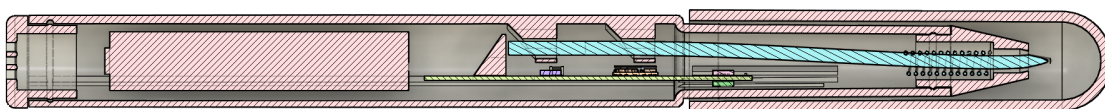
Pro konstrukci pera se standardně využívá více možností. Kvůli jednoduchosti jsem zvolil konstrukci pera s pevně zabudovanou tuhou a víčkem. Obvyklá pera mají výšku okolo 14 cm a průměr okolo 1 cm. Při samotné konstrukci jsem tak cílil na tyto rozměry. Především průměr pera jsem chtěl zachovat co nejmenší, kvůli zajištění komfortu psaní. Toho se mi podařilo dosáhnout zúžením držadla oproti zbytku pera.

Model pera jsem navrhoval v programu Fusion 360. Modeloval jsem ho takovým způsobem, aby se později dalo pero vytisknout na 3D tiskárně a jeho kompletace byla co možná nejsnazší. Tvar pera se vesměs řídil tvarem desky plošných spojů. Ta se sice musela kvůli tuze posunout lehce mimo střed pera, ale to nijak neovlivnilo finální rozměry. Konečné rozměry pera jsou 149 mm na výšku, 10 mm v průměru v užší části a 13 mm v širší části.



Obrázek 3.15: Model pera

Na obrázku 3.16 je ukázáno rozložení všech částí uvnitř pera. Deska plošných spojů a baterie se do pera zasouvají z horní části pera, na kterou doléhá uzávěr. Uzávěr obsahuje LED diodu pro indikaci stavu pera, přepínač pro odpojení baterie od zařízení, přepínač pro resetování desky a konektor pro připojení k desce. Uzávěr obsahuje tyto části pouze pro snadnější interakci s perem během testování a v případě potřeby ho lze odpojit a nahradit verzí bez těchto prvků. Tuha se zasouvá do pera spodem, projde dvěma oporami a zarazí se o tlakový senzor. Pružina pak zajišťuje nutný přítlak pera na senzor, aby tuha nevyjžděla.



Obrázek 3.16: Průřez perem i se všemi částmi

### 3.4 Firmware

Před samotnou implementací firmware jsem si specifikoval přibližnou podobu programu a funkcionality, které by měl firmware podporovat.

Program po startu a inicializaci čeká na příchozí spojení. Po navázání spojení čeká na činnost od klienta. Po obdržení akce ji vyhledá a provede. V případě čtení atributu jeho hodnotu předá uživateli, v případě zápisu změni hodnotu parametru a informuje uživatele o úspěšné změně. Po odpojení klienta program přeruší vykonávanou akci a čeká na další připojení.

Akce slouží pro vykonání nějaké procedury v peru. Atributy pak slouží pro úpravu parametrů programu, nebo jejich vyčtení. Celý koncept atributů a akcí byl navržen s cílem snadné nastavitelnosti zařízení, bez nutnosti přeprogramování desky a s možností jednoduchého přidávání dalších atributů. V následující tabulce 3.1 je seznam těchto akcí a atributů včetně jejich krátkého popisu.

Akce				
Id	Název	Popis		
0x00	Idle	Výchozí akce systému. Přeruší vykonávanou akci.		
0x10	Logging	Pero zahájí měření. Naměřené hodnoty ihned odesílá přes bluetooth.		
0x20	Record	Pero zahájí měření. Data odesílá až po překonání hranice přitlaku a to po určitou dobu. Poté se přepne do akce Idle.		
0x30	Soft restart	Pero se restartuje.		
Atributy				
Id	Název	Přístup	Výchozí hodnota	Popis
0x02	Sample frequency	Čtení/Zápis	50	Vzorkovací frekvence senzorů v Hz.
0x03	Buffer size	Čtení	40 096	Velikost kruhového bufferu v bytech.
0x04	Buffer fill (%)	Čtení	0	Procento naplnění bufferu.
0x05	Buffer fill	Čtení	0	Naplnění bufferu v bytech
0x06	Buffer overflow	Čtení/Zápis	0	Chování pro naplnění bufferu. 0 - zastavení, 1 - ztráta hodnot
0x07	Led state	Čtení	0	Aktuální stav LED diody. 0 - vypnuto, 1 - zelená, 2 - oranžová, 3 - červená
0x08	Pressure threshold	Čtení/Zápis	8 000	Hodnota přitlaku pro zahájení měření.
0x21	Record duration	Čtení/Zápis	3 000	Doba nahrávání v milisekundách.

Tabulka 3.1: Seznam podporovaných akcí a atributů

## 3.5 Software

Před tím, než jsem začal software implementovat, identifikoval jsem případy užití, které od aplikace budu vyžadovat. Dále jsem si vytvořil přibližný návrh uživatelského rozhraní aplikace. Cílovou skupinou pro tuto aplikaci není typický uživatel, ale je to někdo, kdo si aplikaci dále přizpůsobí a je seznámen s řešenou problematikou. Během návrhu uživatelského rozhraní jsem tak necítil na perfektní intuitivnost aplikace. Dále jsem vybral programovací jazyk a prozkoumal nabídku balíčků, které by mi mohly usnadnit vývoj aplikace.

Aplikace slouží jako prostředek k interakci s perem. Umožňuje připojení k peru a zasílání příkazů. Zaznamenává data, která pero propaguje. Získaná data pak aplikace prezentuje uživateli pomocí grafů a umožňuje jejich export do souboru. Zároveň umožňuje úpravu získaných dat a provedení operací nad nimi, včetně jejich následné vizualizace.

### 3.5.1 Výběr technologií

Aplikace bude navržena s využitím architektury MVC (Model-View-Controller) v jazyce Python 3. Pro implementaci uživatelského rozhraní jsem zvolil knihovnu TkInter. TkInter je lehká, objektově orientovaná GUI (graphic user interface – grafické uživatelské rozhraní)

knihovna, postavená na základě Tcl/Tk. Výhodou knihovny TkInter je, že je dodávána společně s jazykem Python. Není tak potřeba doinstalovávat další balíčky.

Pro komunikaci s BLE zařízením jsem vybral balíček Bluepy. Je to jednoduchý wrapper nad programem BlueZ. Program BlueZ je oficiální aplikace pro protokol Bluetooth určená pro systémy Linux. Tato závislost tedy limituje použití aplikace pouze pro počítače se systémem Linux. K dispozici je však více balíčků, budu tedy cílit na jednoduchou nahraditelnost tohoto balíčku za jiný.

Při výběru nástroje pro vizualizaci dat je nutné brát v potaz kompatibilitu balíčku s knihovnou TkInter. Mezi hojně používané balíčky patří balíček matplotlib. Jelikož podporuje integraci do knihovny TkInter a umožňuje vizualizaci tak, jak bych si ji představoval, proto jsem ho použil.

### 3.5.2 Rekonstrukce pohybu

Aplikace z pera obdrží neupravené naměřené hodnoty. Je tedy na ní, aby tato data správně rekonstruovala v pohyb.

Prvním krokem pro rekonstrukci pohybu je převedení naměřených hodnot na správné jednotky. Hodnoty z akcelerometru je potřeba převést na  $m/s^2$  toho lze docílit pomocí rovnice

$$\vec{a} = \vec{Acc}/Acc_{LSB}; \vec{Acc} = (Acc_x, Acc_y, Acc_z)^T, \quad (3.2)$$

kde  $\vec{Acc}$  je vektor hodnot z akcelerometru a  $Acc_{LSB}$  je citlivost senzoru v  $G$ . Pro gyroskop je postup podobný, jen jeho citlivost je udávána v  $^\circ/s$ , je tedy nutné převést výsledek na  $rad/s$ .

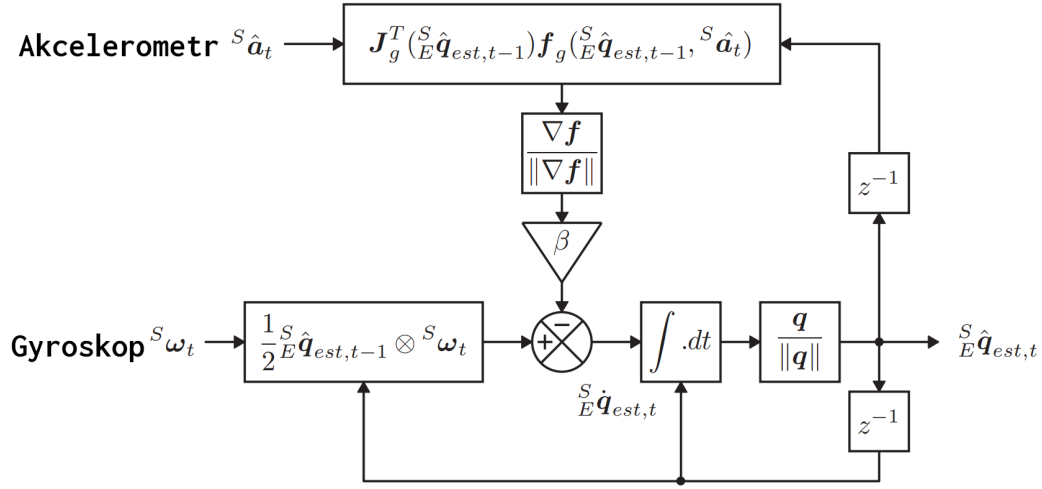
$$\vec{\omega} = \frac{\vec{Gyr}}{Gyr_{LSB}} * \frac{\pi}{180}; \vec{Gyr} = (Gyr_x, Gyr_y, Gyr_z)^T, \quad (3.3)$$

$\vec{Gyr}$  značí vektor hodnot gyroskopu a  $Gyr_{LSB}$  citlivost gyroskopu.

Pro aproximaci natočení pera v prostoru jsem vyhledal algoritmus Madgwick IMU. Článek o jeho tvorbě uvádí dobré výsledky i při nízkých obnovovacích frekvencích. Statickou chybu  $0.8^\circ$  RMS (efektivní hodnota) a dynamickou chybu  $1.7^\circ$  RMS [10]. Jeho výhodou je využití kvaternionů, které vyžaduje méně matematických operací při provádění transformací.

$$\beta = \left\| \frac{1}{2} \hat{q} \otimes [0 \quad \tilde{\omega}_{max} \quad \tilde{\omega}_{max} \quad \tilde{\omega}_{max}] \right\| = \sqrt{\frac{3}{4} \tilde{\omega}_{max}} \quad (3.4)$$

Algoritmus má jeden nastavitelný parametr  $\beta$ , který představuje chybu měření gyroskopu, vyjádřený jako velikost derivátu kvaternionu. Tento parametr je vhodné nastavit jako maximální možnou chybu měření každé osy  $\tilde{\omega}_{max}$ . [10]



Obrázek 3.17: Blokové schéma pro aproximaci orientace [10]

Z blokového schéma lze vidět, že aproximace orientace  ${}^S \hat{q}_{est,t}$  je vypočítána integrací odhadované změny orientace  ${}^S \dot{q}_{est,t}$ , která je zbavena chyby měření gyroskopu,  $\beta$ , ve směru daném akcelerometrem. [10]

Aproximace orientace,  ${}^S \hat{q}_{est,t}$ , je dále použita pro transformaci vektoru akcelerace,  $\vec{a}$ , z lokálního souřadnicového systému do globálního souřadnicového systému. Pro tuto akci je potřeba převést vektor akcelerace do formy kvaternionu  $q_{\vec{a}}$ :

$$\vec{a} = (a_x, a_y, a_z)^T \leftrightarrow 0 + a_x i + a_y j + a_z k = q_{\vec{a}}. \quad (3.5)$$

Transformace se pak provede pomocí rovnice:

$$q_{\vec{a}_{est}} = {}^S \hat{q}_{est,t}^{-1} * q_{\vec{a}} * {}^S \hat{q}_{est,t}. \quad (3.6)$$

Vektorová část výsledného kvaternionu  $q_{\vec{a}_{est}}$  je transformovaný vektor akcelerace  $\vec{a}_{est}$ . [15]

Jelikož neplánuji pro rekonstrukci pohybu využít akceleraci v ose  $z$ , není potřeba pro  $\vec{a}_{est}$  kompenzovat podíl gravitace. Dále je tedy integrací zrychlení  $\vec{a}_{est}$  získána rychlost  $\vec{v}_{est}$ . Integrací rychlosti je pak získán odhad pozice pera  $\vec{p}_{est}$  v globálním souřadnicovém systému.

$$\vec{v}_{est} = \int \vec{a}_{est} dt \quad (3.7)$$

$$\vec{p}_{est} = \int \vec{v}_{est} dt \quad (3.8)$$

Pro získání odhadu pozice hrotu pera,  $\vec{p}_{h_{est}}$ , je potřeba vektor vzdálenosti senzoru od hrotu pera,  $\vec{h}$ , rotovat odhadovaným natočením pera. To lze provést převedením  $\vec{h}$  do formy kvaternionu:

$$\vec{h} = (0, 0, -0.039)^T \leftrightarrow 0 + 0i + 0j - 0.039k = q_{\vec{h}}. \quad (3.9)$$

A následnou rotací vektoru:

$$\vec{h}_{est} = {}^S \hat{q}_{est,t} * q_{\vec{h}} * {}^S \hat{q}_{est,t}^{-1}. \quad (3.10)$$



Odhad pozice hrotu pera v globálním souřadnicovém systému je pak součet pozice pera a pozice hrotu pera vůči peru:

$$\vec{p}h_{est} = \vec{p}_{est} + \vec{h}_{est}. \quad (3.11)$$

# Kapitola 4

## Realizace

V této kapitole popíši implementaci firmware a software, výrobu a osazení desky plošných spojů a nakonec i konstrukci samotného pera.

Všechny programovací části projektu byly vyvíjeny v prostředí programu Visual Studio Code. Visual Studio Code je lehký open-source editor pro úpravu zdrojového kódu. Tento editor nevyžaduje žádné speciální vytváření projektu, projekt tak lze snadno přenést do jiného vývojového prostředí. Během vývoje bylo použito i rozšíření Cortex-Debug do tohoto editoru. Pomocí tohoto rozšíření lze snadno ladit firmware a nahrávat nový kód na desku.

Popis programovací části seznamuje s výslednou architekturou a principem fungování programu. Mimo jiné zmiňuje i využití technologie, včetně jejich popisu a použití. Určitou část jsem věnoval i popisu uživatelského rozhraní. Ve vybraných sekcích lze také nalézt zavedené formáty dat pro komunikaci mezi perem a počítačem.

Následuje popis osazení desky plošných spojů, včetně jejího otestování. Během testování byly objeveny chyby v návrhu, je zde tedy zmíněna i oprava těchto chyb.

V závěru kapitoly jsou dokončeny potřebné části, mimo desku plošných spojů a jejich složení.

### 4.1 Firmware

Kód pro mikroprocesor využívá vývojářský balíček (SDK) od Nordic Semiconductors ve verzi 16. Tento balíček obsahuje řídicí moduly pro všechny periférie, které jejich rodina mikroprocesorů nabízí. Implementace nových funkcionalit je tak velmi zjednodušena. Obsahuje i binární soubory pro Softdevice a hlavičkové soubory pro komunikaci s ním. V balíčku jsou obsaženy i příklady pro různé případy užití. Lze z nich získat představu o práci s tímto balíčkem, nebo inspiraci pro vlastní implementaci. Sám jsem se inspiroval příkladem pro UART službu přes BLE. Základní kostru projektu jsem postavil na tomto příkladu. Příklad jsem ale značně modifikoval a rozšířil o další funkcionality.

Většina úprav, které bylo potřeba provést, se nachází ve složce `app` uvnitř projektu. Konkrétně se jedná o implementaci ADC (analogově digitální převodník), komunikace SPI, Bluetooth služby a systému akcí a atributů.

Kompilace programu je popsána v souboru `Makefile` a lze ji spustit příkazem `$ make`.

#### 4.1.1 Bluetooth služba

Bluetooth standard neobsahuje služby a charakteristiky, které bych chtěl využít. Bylo proto nutné vytvořit si vlastní službu a charakteristiky. Vytvořil jsem tedy službu s náhodně

vybraným UUID 3f400003-b5a3-f393-e0a9-e50e22dcca9e. Výslednou strukturu služby lze vidět v obrázku 4.1.

	Identifikátor	UUID	Oprávnění	Hodnota
Služba pera	0x0011	Služba	Čtení	PEN
Charakteristika pro stream dat	0x0012	Charak.	Čtení	N   0x0002
	0x0013	0x0002	Čtení	data
Deskriptor	0x0014	CCCD	Čtení/Zápis	0x0001
Charakteristika pro status systému	0x0015	Charak.	Čtení	RD   0x0003
	0x0016	0x0003	Čtení	status
Charakteristika pro zvolení akce	0x0017	Charak.	Čtení	RD   0x0004
	0x0018	0x0004	Zápis	id akce
Charakteristika pro práci s atributy	0x0019	Charak.	Čtení	RD   0x0005
	0x001A	0x0005	Čtení/Zápis	hod. atributu

Obrázek 4.1: Služba pro komunikaci s perem.

Služba obsahuje charakteristiku vyhrazenou pro přenos dat generovaných akcemi. Všechna přenesená data se týkají dat ze senzorů. Formát zprávy proto odpovídá obrázku 4.2 a obsahuje pouze nutné části k minimalizaci času potřebného pro odeslání dat.

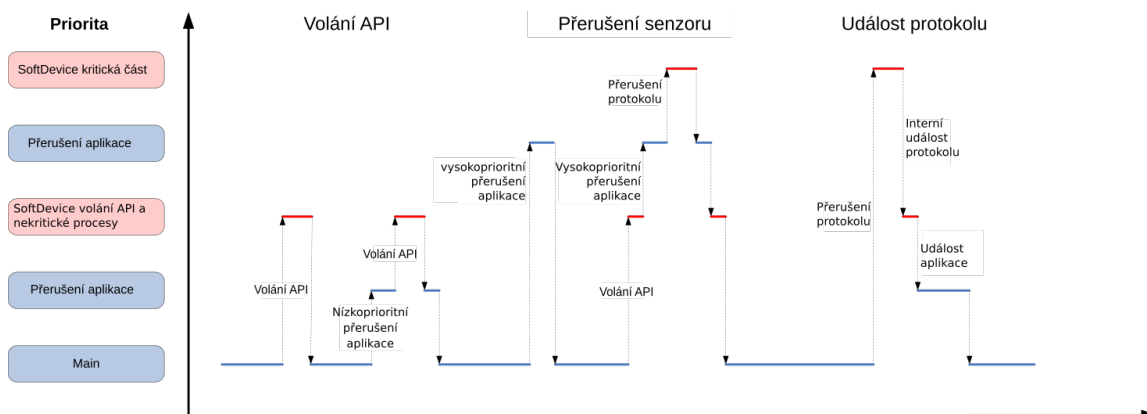
0	7	15	23	31
počítadlo		přítlak		
akcelerometr - x		akcelerometr - y		
akcelerometr - z		gyroskop - x		
gyroskop - y		gyroskop - z		

Obrázek 4.2: Formát dat senzorů

Charakteristika pro status systému obsahuje hodnotu, v jakém stavu se aktuálně nachází systém. Možné hodnoty jsou: nečinný, zaneprázdněný a chyba. Charakteristika pro zvolení akce slouží ke spuštění akce. Charakteristika pro práci s atributy slouží ke čtení, či k zápisu různých parametrů systému.

### 4.1.2 Běh programu

Běh programu je řízen pomocí přerušení a jejich priority. Provádění měření, indikace LED diody, či příjem dat jsou tak provedeny až po příslušném přerušení. Čip nabízí 8 úrovní priorit od 0 do 7. Menší úroveň znamená vyšší prioritu. Softdevice si vyhraduje priority 0, 1 a 4 a kód aplikace může využít ostatní. Obrázek 4.3 na příkladu ukazuje běh programu a jeho přepínání podle priority přerušení.



Obrázek 4.3: Přepínání kontextu podle úrovně priority přerušení [16]

Aplikace, po startu a inicializaci, začne propagovat BLE pakety a čeká na příchozí spojení. Propagaci indikuje svítící zelená LED dioda. Po navázání spojení přepne barvu LED diody na oranžovou a čeká na příkazy od klienta. Pokud by během startu, či inicializace, mělo dojít k chybě, bude tato chyba indikována rozsvícením červené LED diody. Během inicializace jsou vypisovány ladící hlášky na rozhraní UART. Při připojeném programátoru lze tyto hlášky vyčíst připojením se na virtuální com port programátoru.

Samotné měření je spuštěno pouze odpovídající akcí. Měření ze senzorů je řízeno čítačem o volitelné frekvenci. Při přetečení čítače získá obslužná rutina hodnoty ze senzorů a ty uloží do kruhového bufferu. Po uložení dat do bufferu využije čas před dalším přetečením čítače, pro odeslání co možná největšího množství dat a přerušení ukončí. Jelikož se jedná o hlavní funkcionalitu pera, má komunikace se senzory vysokou prioritu přerušení 2. Přerušení čítače má prioritu 5. Chování po naplnění kruhového bufferu je nastavitelné atributem.

### 4.1.3 Senzory

Komunikace s inerciálními senzory probíhá po jednotném SPI rozhraní. Senzory se rozlišují aktivací CS (chip select) pinu čipu. Řídící modul pro SPI v SDK nemá podporu pro více zařízení na jednom rozhraní. Bylo tedy nutné tuto funkcionalitu doplnit.

Pro komunikaci se senzory bylo nutné implementovat pomocné moduly. Pro senzory BMI270 a BMG250 jsem využil knihovny poskytované výrobcem. Pro senzor FXLS8471Q jsem pomocný modul implementoval sám. Pomocné moduly jsem následně napojil na moji aplikaci.

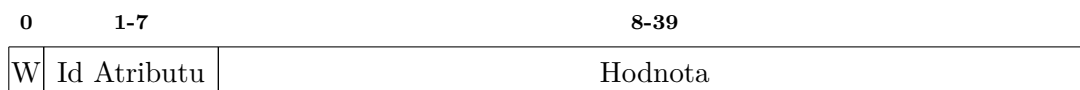
Jelikož je desku možné osadit různými senzory, bylo nutné mít možnost zvolit použitou implementaci. Zvolení implementace je provedeno pomocí podmíněné kompilace, která se rozhoduje podle přepínače `USE_IMU`. V případě, že je tento přepínač definovaný jako hodnota 1, se kód zkompile pro senzor BMI270, jinak pro senzory BMG250 a FXLS8471Q. Definici tohoto přepínače lze nalézt v souboru `sdk_config.h`.

Výstup tlakového snímače nelze měřit na přímo. Je proto potřeba zapojit senzor jako napěťový dělič a měřit napětí na výstupu tohoto děliče. Měření je provedeno pomocí vestavěného ADC v Bluetooth modulu. Bylo tedy potřeba inicializovat tuto periférii a navázat ji na běh programu.

#### 4.1.4 Atributy a akce

Atributy a akce jsou implementovány podle tabulky 3.1. Další akce lze snadno přidávat v souboru `action_table.c`. Stačí definovat metodu, která reprezentuje akci tak, aby odpovídala typu `ActionTable_Func_t`. A ukazatel na tuto metodu, společně s identifikátorem akce, umístit do proměnné `pCommandTable_Commands`. Pro přidání nového atributu je pak proces podobný jako u akce. Do souboru `attribute_table.c` přidat metodu dle typu `AttributeTable_Func_t`, která se zavolá při žádosti o zápis, či o čtení hodnoty atributu. A ukazatel na tuto metodu přidat do proměnné `lpAttributeTable_AttributeTable`, včetně identifikátoru atributu a oprávnění.

Pro vyčtení parametru je potřeba prvně odeslat jeho sedmibitový identifikátor a poté z charakteristiky vyčíst jeho hodnotu. Pro zápis je nutné nastavit bit pro indikaci zápisu (W), identifikátor atributu a hodnotu o 1-4 bytech. Po zápisu lze z charakteristiky vyčíst hodnotu upravovaného atributu. Lze tedy takto ověřit provedení zápisu. V případě neplatné hodnoty se hodnota atributu nezmění. Rámec pro práci s atributy tedy odpovídá obrázku 4.4



Obrázek 4.4: Formát dat pro zápis atributu

K zahájení akce je potřeba odeslat její identifikátor. Pro zastavení prováděné akce je určena výchozí akce Idle.

Nastavení atributů je volatilní, nelze tak změnit výchozí nastavení bez přeprogramování desky.

Pro implementaci atributů se nabízelo využít atributy vrstvy GATT. Ačkoli by byl tento způsob z pohledu klienta čistší, zavrhl jsem ho. Zejména kvůli malému maximálnímu množství atributů, prodloužení části objeovávání charakteristik a složitějšímu přidávání dalších atributů.

## 4.2 Software

Aplikace je napsána v jazyce Python 3 a využívá knihovnu TkInter k tvorbě uživatelského rozhraní. K výpočtu odhadu pozice pera a k rekonstrukci podpisu jsou využity balíčky `pyquaternion` a `numpy`. Pro zobrazení všech grafů v aplikaci je využit balíček `matplotlib`. Pro komunikaci s perem přes BLE je využit balíček `bluepy`. Při běhu aplikace jsou vypisované různé informační hlášky do konzole. Pro barevné zpřehlednění tohoto výpisu je využíván balíček `colorama`. Všechny tyto závislosti lze nainstalovat příkazem `$ pip3 install -r requirements.txt`.

### 4.2.1 Běh programu

Běh aplikace začíná v metodě `main`, která vytvoří hlavní okno aplikace a řadič `DataController`. Řadič poté vytvoří datový model a dále pohledy. Tyto pohledy umístí a přizpůsobí vizuální prvky aplikace.

Datový model při inicializaci vytvoří třídu pro komunikaci přes BLE. Při návrhu jsem cílil na snadnou zaměnitelnost této třídy. Vytvářená třída tedy musí splňovat rozhraní definované abstraktní třídou `BleInterface`. Jelikož třída pro komunikaci běží v jiném vlákne, je potřeba pravidelně volat aktualizací metodu modelu. Ta zkontroluje, zda-li nejsou dostupná nová data a ověří stav připojení. Pravidelná aktualizace je řešena vestavěným plánovačem knihovny `TkInter`. V případě změny stavu připojení, nebo nových dat, upozorní pohled o změně. Pro upozornění pohledu o změně atributů v modelu jsem využil návrhový vzor `Observer`. Tato třída také převádí odesílaná data na správný formát (viz. obrázek 4.2 a 4.4). Datový model, mimo jiné, obsahuje i seznam proveditelných akcí a atributů (viz. tabulka 3.1).

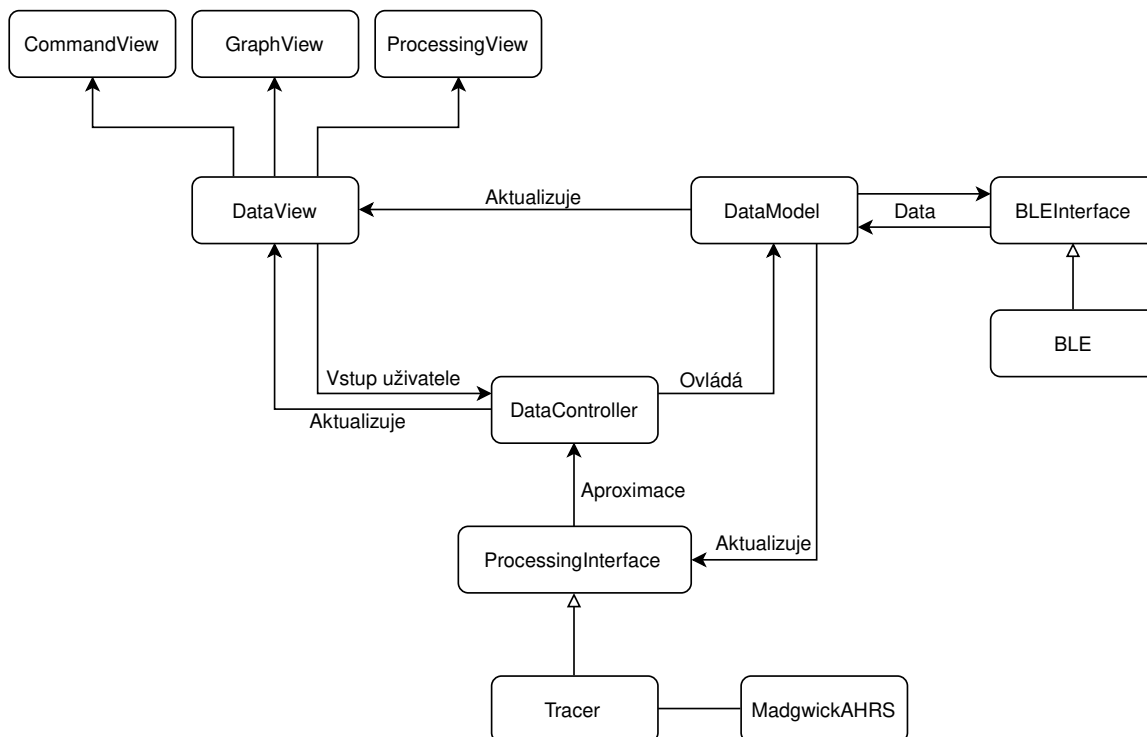
Komunikace přes BLE běží v separátním vlákne. Komunikace mezi vlákny je realizována pomocí událostí a `front`. Hlavní smyčka vlákna pro komunikaci kontroluje výskyt událostí ze strany modelu a podle nich pak zahájí/ukončí spojení, či skenování. Kontroluje také upozornění od pera na nová data. V případě nových dat tato data přečte a zařadí do příchozí fronty. Pokud by se v odchozí frontě vyskytla nějaká data, odešle tato data určené charakteristice.

Při obdržení nových dat z pera je datový model předává také do modulu pro zpracování dat (`tracer.py`). Tento modul využívá algoritmus `Madgwick IMU`. Kód pro tento algoritmus byl převzat ze stránek správce projektu [11]. Algoritmus je napsán v jazyce `C`. Jelikož jazyk `Python` umožňuje snadné propojení s jazykem `C`, zkompileval jsem algoritmus jako sdílenou knihovnu a tu poté importoval v modulu `MadgwickAHRS_Wrapper.py`. Modul `tracer.py` tento algoritmus využívá pro aproximaci orientace pera.

Pohled `GraphView` zobrazuje grafy s hodnotami jednotlivých senzorů. Umožňuje také exportovat grafy jako obrázek, případně jako čistá data. Exportovaná data jsou ve formátu  $A_x, A_y, A_z, G_x, G_y, G_z, P$ , kde  $A$  značí akcelerometr,  $G$  gyroskop a  $P$  přítlak. Jednotlivé vzorky dat jsou od sebe odděleny znakem nového řádku.

Pohled `ProcessingView` pak obsahuje kreslicí plochu, na které je vykreslován objekt reprezentující pero. Tento objekt slouží k ukázce odhadované orientace pera. Kromě kreslicí plochy obsahuje pohled i graf, který je určený pro reprodukci podpisu. Graf vykresluje čáru reprezentující odhadovaný pohyb hrotu pera. Barva čáry pak určuje míru přítlaku v průběhu podpisu. Data z tohoto grafu lze exportovat jako obrázek, nebo do souboru. Do souboru se data ukládají ve formátu  $x, y, P$ , kde  $x$  a  $y$  jsou souřadnice bodu a  $P$  míra přítlaku. Jednotlivé vzorky jsou od sebe oddělené znakem nového řádku.

Výsledná architektura aplikace odpovídá diagramu 4.5.



Obrázek 4.5: Diagram objektů aplikace

## 4.2.2 Rozložení GUI

Uživatelské rozhraní je rozděleno na dvě části. V levé části se nachází panel pro ovládání pera. V pravé jsou záložky s grafy prezentujícími data.

Část pro ovládání pera se dále dělí na další sekce. V první části od vrchu se nachází sekce pro připojení k peru. Tlačítko **Scan** vyhledá BLE zařízení v dosahu a zařadí jejich název a mac adresu do rozbalovacího seznamu. Z tohoto seznamu uživatel vybere zařízení, ke kterému se chce připojit. Pokus o připojení může provést pomocí tlačítka **Connect**. Připojování je indikováno zašednutím a změnou textu tlačítka na **Connecting...** Po úspěšném připojení, se toto tlačítko promění na tlačítko pro odpojení. Při neúspěšném připojení je tlačítko vráceno do původního stavu.

Pod první sekci se nachází další sekce, která je určena pro zasílání akcí peru. Z rozbalovacího seznamu lze vybrat požadovanou akci a odeslat ji pomocí tlačítka **Send Action**.

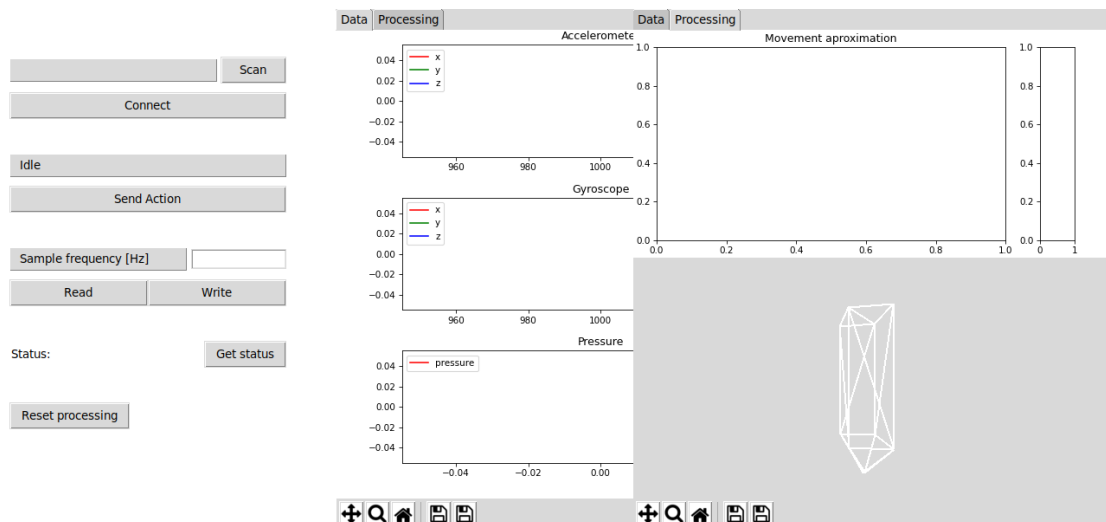
Dále následuje sekce pro čtení a úpravu atributů. Z rozbalovacího seznamu lze vybrat požadovaný atribut. Pokud je atribut pouze pro čtení, či zápis, umožní aplikace použít pouze odpovídající tlačítko. Vedle rozbalovacího seznamu je textové pole obsahující hodnotu atributu. Textové pole je vyplněno automaticky při vyčtení atributu. Při zápisu atributu je zapisovaná hodnota brána z tohoto pole. V případě chyby zápisu je hodnota vrácena na původní.

Předposlední sekci je informace o stavu systému. Dotaz na stav systému lze provést pomocí tlačítka **Get status**.

V poslední sekci je tlačítko pro vynulování odhadu orientace pera.

Část se záložkami obsahuje dvě záložky. První záložka **Data** prezentuje přijaté hodnoty pro každý sensor. Obsahuje panel nástrojů, který umožňuje interagovat s grafem. Zároveň umožňuje export grafů jako obrázku, nebo export naměřených dat sensorů do souboru.

Druhá záložka obsahuje graf pro odhadovaný pohyb hrotu. Graf opět obsahuje panel nástrojů se stejnou funkcionalitou jako panel nástrojů v záložce **Data**. Ale je rozšířený o funkci obnovy výchozího stavu. Pod grafem se nachází kreslicí plátno, které obsahuje objekt prezentující odhad orientace pera.



Obrázek 4.6: Uživatelské rozhraní aplikace

### 4.2.3 Rekonstrukce pohybu

Rekonstrukce pohybu je implementována podle připraveného návrhu (subsekcce 3.5.2). Pro práci s kvaterniony je použit balíček `pyquaternion`. Pro práci s vektory pak balíček `numpy`. Oba tyto balíčky umožnily značně zjednodušit a zpřehlednit výpočty.

Během implementace je také důležité si uvědomit, že senzory nejsou ani zdaleka ideální. Oba senzory produkují jistou míru rušení a zároveň jejich hodnoty nejsou referencovány přesně vůči nule. Jejich hodnoty mají také tendenci měnit se v závislosti na teplotě. Gyroskop má navíc tendenci driftu v průběhu času. Tedy jeho hodnoty se časem mění nezávisle na jeho reálném pohybu. Je důležité snažit se míru chyby co nejvíce minimalizovat, jelikož výpočet obsahuje dvojitou integraci. Chyba je tak po první integraci lineární a po druhé exponenciální.

Pro snížení míry rušení jsem aktivoval dolní propust přímo v akcelerometru. K vyrovnání nulové hodnoty jsem z prvních  $n$  vzorků vypočítal střední hodnotu. Ta je následně vždy odečítána od dalších naměřených vzorků. Ovlivnění měření zapříčiněné změnou teploty jsem zanedbal.

Integrace zrychlení a rychlosti je aproximována pomocí lichoběžníkové formule při vzorkovací frekvenci  $h$  (viz. rovnice 4.1).

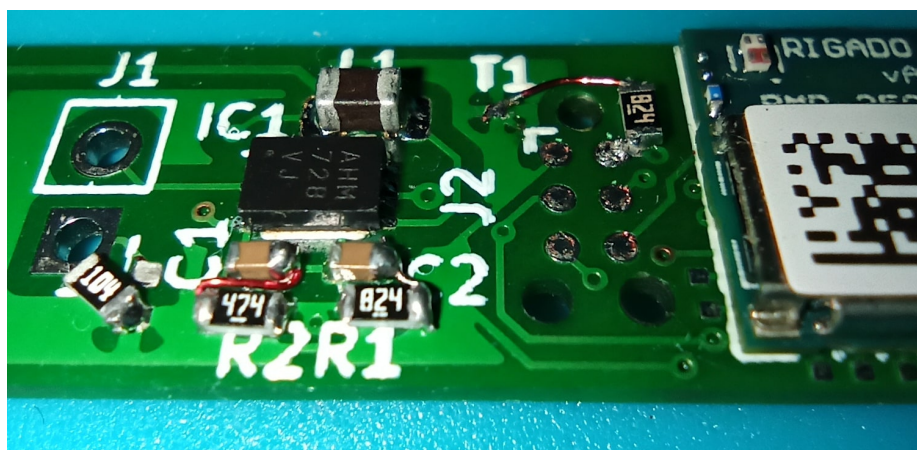
$$\int x dt \approx \sum_{n=1}^k \frac{x_{k-1} + x_k}{2} h \quad (4.1)$$



### 4.3 Deska plošných spojů

Navrženou desku jsem, se všemi potřebnými podklady, odeslal zahraničnímu výrobcí. Odeslané soubory lze najít na příloženém CD. Společně s deskou jsem nechal vyrobit i plechovou šablonu pro osazování SMD součástek.

Po obdržení vyrobených desek jsem na jednu z nich pomocí šablony nanasl pájecí pastu. Na desku jsem osadil všechny SMD součástky z její horní strany a vložil ji do infračervené pece pro zapájení. Po ukončení programu pece jsem desku vyjmul. Jelikož z dolní strany desky bylo ještě potřeba napájet gyroskop, zajistil jsem akcelerometr vteřinovým lepidlem. Na dolní stranu jsem (opět pomocí šablony) nanasl pájecí pastu a umístil gyroskop společně s kondenzátorem. Pomocí horkovzdušné stanice jsem následně obě součástky zapájel. Nakonec zbývalo napájet tlakový senzor a nalepit jeho oporu. Bohužel jsem objednal senzor s nesprávným typem konektoru, byla tedy nutná úprava konektoru pro připájení. Zároveň nebylo možné sehnat rezistory pro zpětnou vazbu měniče a napěťový dělič tlakového senzoru v požadovaném pouzdru. Proto jsem i zde zaimprovizoval a použil rezistor většího pouzdra s dodatečnou modifikací.



Obrázek 4.7: Osazená deska i s provedenými modifikacemi

Po osazení desky jsem otestoval funkčnost ladícího rozhraní včetně toho, zda-li jde modul naprogramovat. Pro programování desky jsem využil program nrfjprog poskytnutý výrobcem. Pro úspěšné naprogramování desky je potřeba připojit mezi počítač a desku programátor. Já jsem využil programátor JLink, lze ale využít i STLink, či jiný programátor podporující rozhraní SWD (serial wire debug). Lze také využít programátor JLink na evaluační desce. Desku je při prvním připojení od výroby potřeba zformátovat (např. pomocí programu nrfjprog příkazem `$ nrfjprog -f nrf52 --recover`).

Během testování se ukázalo, že deska funguje správně, pouze pokud je běh programu spuštěn přes ladící rozhraní, jinak nevykazuje známky života. Při diagnostice problému jsem chybu objevil. Chybu zapříčinil chybějící rezistor stahující reset pin Bluetooth modulu k napájecí úrovni. Reset pin byl tak bez připojeného programátoru, který resetovací pin nastavil na správnou úroveň, ve stavu vysoké impedance. Modul se tak neustále resetoval. Resetovací pin modulu je vyveden pouze na plošku pro programátor. Na tu jsem nechtěl pájet, jelikož by to mohlo znesnadnit programování. Narušil jsem proto ochrannou masku desky v místě, kde vede cesta k resetovací plošce. Na tuto cestu jsem napájel odpor a drátkem propojil s napájecí větví. Tuto chybu jsem opravil i ve finální verzi schématu.

## 4.4 Model pera

Model pera se skládá ze čtyř částí. Tyto části jsem exportoval ve formátu STL z modelovacího programu. Následně jsem je importoval do programu Prusa Slicer, který slouží pro vytvoření souboru s instrukcemi pro 3D tiskárnu (GCode). V tomto programu jsem nastavil parametry tisku, jmenovitě výšku vrstvy na 0.1mm, umístění podpůrného materiálu a orientaci části při tisku. Orientace součástky je důležitá pro výslednou kvalitu a pevnost. Všechny části jsem umístil na výšku. Tento způsob sice snížil výslednou pevnost, ale konečný produkt je vizuálně lepší. Po nastavení tisku jsem vygeneroval soubory GCode a ty přes SD kartu přenesl do 3D tiskárny.

Tisk částí probíhal na 3D tiskárně Prusa i3 MK2S. Jednotlivé části jsem tiskl každou zvlášť a až na hlavní část pera nebyl s tiskem žádný problém. Hlavní část pera měla tendenci odlepovat se od podložky, bylo tedy nutné zvýšit průměr podpůrného materiálu a model lehce modifikovat.

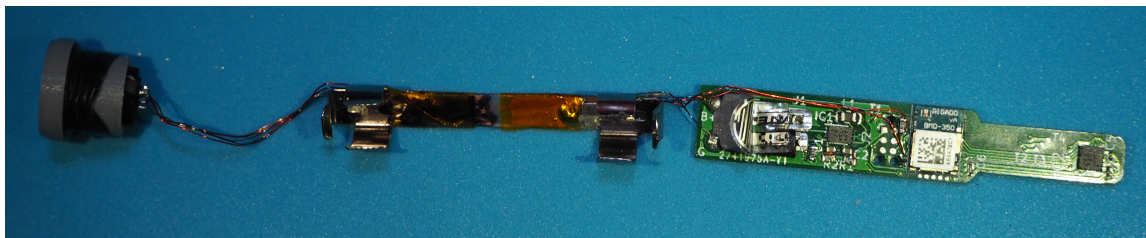
## 4.5 Kompletace

Po dokončení desky plošných spojů a vytištění všech částí pera jsem ještě dokončil zátku pera. Do zátky bylo zapotřebí vložit dva přepínače a jednu dvoubarevnou LED diodu (červená a zelená). Tyto součástky jsem v zátce zajistil vteřinovým lepidlem. Dále jsem propojil drátky tak, aby odpovídaly zapojení ve schématu (viz. obrázek 3.12). Nakonec jsem přidal konektor, kterému bylo nutné lehce zbrousit hrany tak, aby kopíroval obvod zátky. Pod konektor jsem, před jeho vložením do zátky, vtačil lepidlo z tavné pistole. To by mělo zajistit všechny části v zátce. Pokračoval jsem nalepením opory pro tlakový senzor. Následně jsem vytvořil držák pro baterii z proužku cuprexitu a dvou opor pro AAAA baterii.

Zhotovením všech těchto částí jsem mohl začít s jejich propojením. Kladný pól baterie je spojen s deskou. Záporný pól baterie je spojen s konektorem zátky pera. Do tohoto konektoru vedou z desky čtyři další drátky (dva pro led, jeden pro zem a jeden pro reset). Všechna spojení jsem provedl pomocí lakovaných drátků.

Spojené části jsem zasunul do pera s hora. Během vkládání desky do pera je potřeba kontrolovat, aby byla zasunuta do připravené drážky. Tím je deska v peru zajištěna. Po vložení desky lze zasunout zátku. Tu jsem navíc omotal vymežovací páskou pro vymezení vůle a zamezení snadnému otáčení. Zbývalo tak už jen zasunout tuhu s pružinou, která musí projít oporami a tuhu zajistit. Část pro zajištění tuhy jsem opět omotal vymežovací páskou.

Pro případné vyjmutí desky je nejlepší zatlačit dlouhým úzkým předmětem z měkkého materiálu na desku ze spodní strany pera.



Obrázek 4.8: Ukázka kompletní elektroniky

## Kapitola 5

# Vyhodnocení

V této kapitole ověřím funkčnost pera a také shrnu dosažené výsledky.

### 5.1 Pero

Testování jsem zahájil ověřením správné funkce hardware. Toto testování proběhlo v pořádku. Jediným nedostatkem pera je potřeba resetovat ho po každém zapnutí. Tento problém není způsoben kódem, jelikož stejný kód na evaluační desce tento problém nemá. Problém jsem přisuzoval kolísání napětí při startu desky, to ale měření osciloskopem nepotvrdilo. Zkoušel jsem také zaměnit stabilizační kondenzátor u Bluetooth modulu za jinou hodnotu, ale to nezpůsobilo žádný rozdíl.

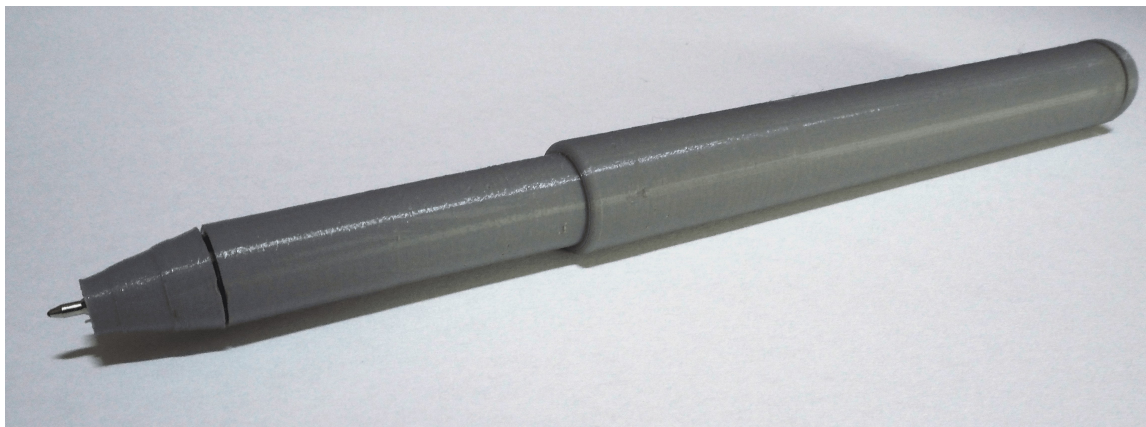
Dále jsem změřil spotřebu pera a zkontroloval jeho výdrž. Pero se zapnout LED diodou, ke kterému není nikdo připojen, spotřebovává z baterie okolo 20 mA při napětí 1,5 V. V případě připojení se k peru se odběr zvýší na 30 mA. Tento proud s klesajícím napětím stoupá a při 1 V dosáhne hodnoty 40 mA (nikdo není připojen) respektive 50mA (aktivní BLE spojení). Pokud předpokládám kapacitu baterie 500 mAh a pomínu veškeré ztráty, mělo by pero teoreticky vydržet 20 hodinový kontinuální provoz.

Po otestování hardware části jsem ověřil funkčnost i firmware. Finální verze pera je osazena senzory BMI270 a FXLS8471Q. Kód desky ale využívá pouze akcelerometr v senzoru BMI270. Testoval jsem připojení k peru, příjem dat, schopnost vyčíst všechny atributy a schopnost vykonat všechny akce. Peru lze nastavit různou vzorkovací frekvenci. Tuto frekvenci je vhodné nastavit v rozmezí od 1 Hz do 166 Hz. Při vyšších frekvencích již Bluetooth modul nestíhá hodnoty ze sensorů získat včas a SoftDevice vyvolá výjimku, což vede pero do chybového stavu. Při frekvencích převyšujících 100 Hz pak dochází k prodlevě odesílání některých zpráv přes BLE a začne se plnit kruhový buffer. Neshledal jsem zde žádný problém, zapnul jsem tedy akci Logging a ověřil správnou funkčnost sensorů.

Akcelerometr jsem testoval náklonem pera v různých osách a sledoval jsem jeho reakci na změnu gravitace. Zkontroloval jsem také, že v případě, že je zkoumaná osa akcelerometru ve směru gravitace, je jeho hodnota blízká 1 G. Gyroskop jsem testoval podobným způsobem, pouze jsem sledoval jeho reakci na rychlost otáčení. Tyto testy sensorů proběhly úspěšně a sloužily pro ověření správné interpretace os sensorů v aplikaci.

Pokračoval jsem testováním snímače přítlaku. U toho jsem testoval v jakém rozmezí síly lze použít jeho hodnoty. Snímač přítlaku je schopen rozeznat již pouhé přiložení pera na papír, bez nutnosti vyvinout jakoukoliv sílu. Meze aplikovatelné síly je okolo 700 g.

Během testování jsem také zjistil, že při podpisu běžně aplikuji sílu v rozmezí od 100 g do 300 g. Použitelné rozmezí snímače přítlaku je tedy dostatečné.



Obrázek 5.1: Konečná podoba pera

## 5.2 Aplikace

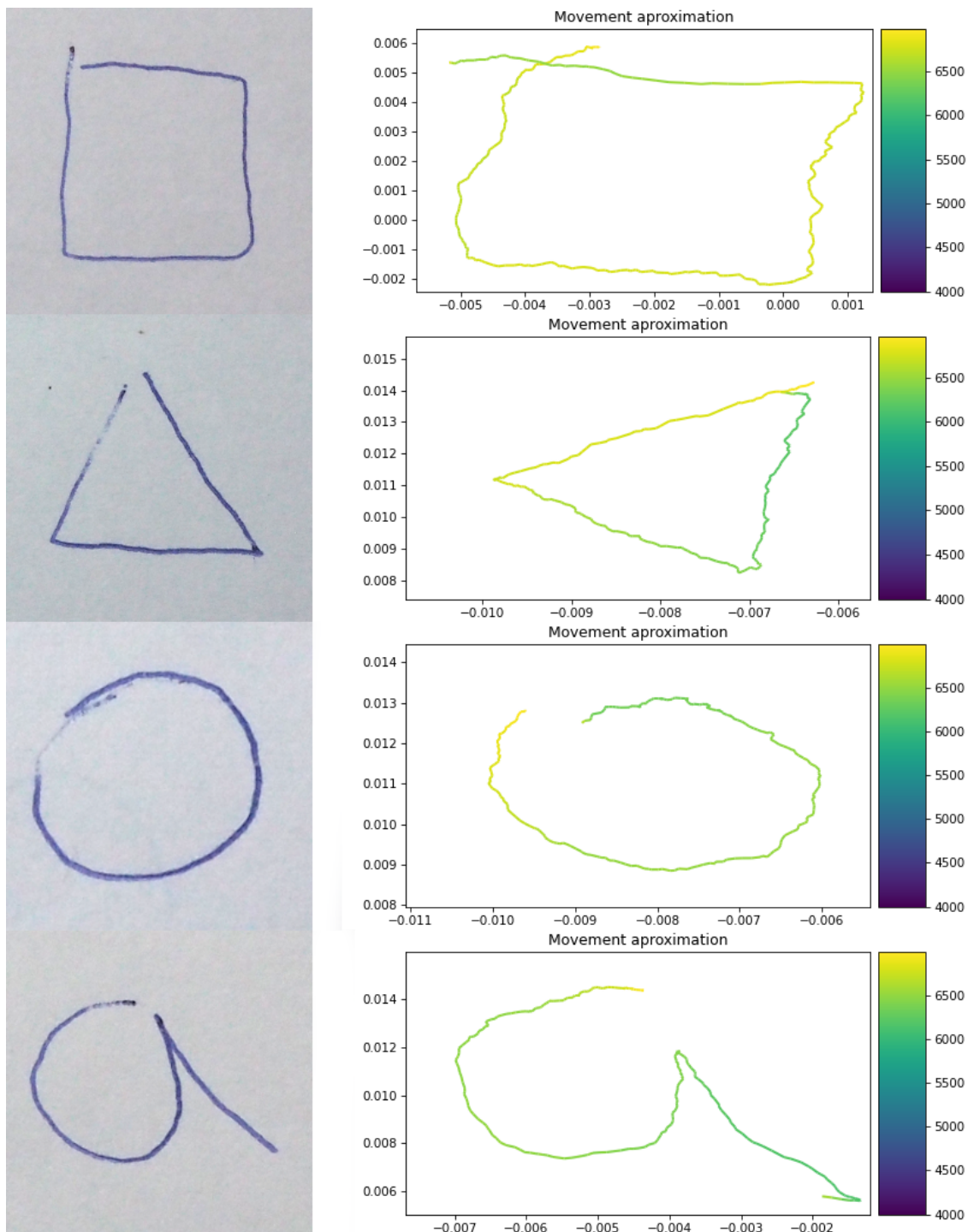
Aplikaci lze spustit příkazem `$ python3 app.py`. Aplikace po stránce uživatelského rozhraní a komunikace s perem funguje dle mých představ.

S částí aplikace pro rekonstrukci podpisu byly během jejího testování problémy. Algoritmus pro orientaci pera do jisté míry funguje. Bohužel ale, i přes snahu lepšího filtrování či úpravy parametru  $\beta$ , trpí driftem. Hlavní problém je tak již při integraci akcelerace převedené do globálního souřadnicového systému. Výsledná rychlost je posunuta oproti nule a zároveň se tento posuv mění s driftem odhadu orientace. S tím přímo souvisí i problém, že rychlost má pouze stoupavou nebo klesavou tendenci.

Tento problém jsem se pokoušel řešit pravidelným hledáním ustálené hodnoty (moment kdy se pero nehýbe). V tomto momentu pak brát naměřené hodnoty jako novou střední hodnotu a pokračovat v měření. Zároveň jsem určil okno pro hodnoty akcelerometru v globálním souřadnicovém systému. Hodnoty v tomto okně jsou brány jako šum a nulují se, mimo toto okno jsou hodnoty neovlivněny. I když tento přístup přinesl jisté zlepšení, nemožňoval i nadále použitelnou rekonstrukci pohybu.

Rozhodl jsem se tak, z důvodu omezeného času, z rekonstrukce odstranit ovlivnění akcelerometrem. Pozici pera tedy neovlivňuje jeho posuv, ale pouze jeho natočení. Po této úpravě sice nelze rekonstruovat slova, ale je stále možné rekonstruovat psaní znaků.

Pro ukázkou rekonstrukce jsem zvolil základní geometrické tvary a písmeno „a“. Pero je nejdříve potřeba zkalibrovat v přímé poloze pomocí funkce Logging. Kalibrace trvá prvních 250 vzorků a uživatel pozná konec kalibrace reakcí objektu v kreslicí ploše na natočení pera. Po kalibraci je možné začít psát. Rekonstrukce pohybu vykreslí křivku pouze pokud detekuje přítlak. Nejlepší vybrané výsledky této rekonstrukce lze vidět v obrázku [5.2](#).



Obrázek 5.2: Vybrané ukázky rekonstrukce

Ačkoliv toto řešení nenaplnuje můj původní záměr, přesto umožňuje demonstrovat schopnosti pera a aplikace. Problém s kompletní realizací rekonstrukce pohybu ovšem nevidím v provedení pera, ale ve zpracování hodnot.

# Kapitola 6

## Závěr

Cílem této práce bylo navrhnout a realizovat pero vhodné pro utajené snímání behaviorálních vlastností podpisu, které by umožňovalo bezdrátový přenos dat. Tento cíl byl úspěšně splněn. Zároveň byla úspěšně vyvinuta aplikace s uživatelským rozhraním pro interakci s perem.

Pero je sestaveno z Bluetooth modulu, který se stará o rádiovou komunikaci a také o řízení pera. Pro snímání behaviorálních vlastností byly zvoleny inerciální senzory (gyroskop a akcelerometr) a odporový tlakový snímač. Společně tak umožňují snímat všechny důležité vlastnosti podpisu. Kvůli minimalizaci rozměrů byla navržena speciální deska plošných spojů a model pera. Tím se podařilo dostat rozměry pera na 149 mm na výšku, 10 mm v průměru držadla a 13 mm v širší části pera.

Aplikace, kromě komunikace s perem, prezentuje přijatá data uživateli pomocí grafů. Po přijetí dat umožňuje data dále zpracovávat a ukáže uživateli odhadovanou orientaci pera a rekonstrukci pohybu pera. V práci je také představen koncept akcí a atributů pro omezení nutnosti přeprogramovávat pero. Aplikace tedy také umožňuje zahajovat jednotlivé akce v peru a nastavovat jeho atributy.

Práce na tomto projektu mi umožnila rozšířit si obzory v jiných odvětvích, než jen v programování a také mi umožnila vyzkoušet si návrh a realizaci kompletního řešení. Při samotné realizaci mě napadla i další možná rozšíření pera, kterými by se dalo zabývat do budoucna.

Prvním z nich je detekce úchopu pera. Detekovat by bylo možné jak pozici prstů, tak i sílu úchopu. Tím by bylo umožněno zkoumat další vlastnost během podpisu. Snímání této vlastnosti by mohlo probíhat pomocí kapacitních snímačů okolo držadla pera a realizováno by mohlo být pomocí nalepené ohebné desky plošných spojů se vzorem měděné vrstvy. Bluetooth modul podporuje kapacitní snímání, ale bylo by potřeba udělat novou desku plošných spojů a adekvátně upravit kód. Toto rozšíření by šlo realizovat i pomocí odporových snímačů přitlaku.

Výhodné může být také programování pera přes BLE. Toto je možné bez dalších úprav hardware. Ale při větší velikosti kódu by mohl být problém s velikostí paměti. Toto rozšíření by vyřešilo nutnost rozebírat pero pro jeho přeprogramování.

Jelikož má pero skryté senzory, nabízí se i možnost dlouhodobého zaznamenávání dat. Pro tento účel by nemusela postačovat vlastní paměť modulu a mohla by se dodělat paměť externí. Bylo by také možné přidat funkcionalitu pro uvedení Bluetooth modulu do režimu spánku.

# Literatura

- [1] *FSR Integration Guide*. Interlink Electronics, Inc. 94-00004 Rev. C.
- [2] *MCP1640/B/C/D Datasheet*. Microchip Technology Inc., srpen 2015. ISBN 978-1-63277-829-1.
- [3] *Bluetooth Core Specification*. Bluetooth SIG, Inc., prosinec 2019. Rev. 5.2.
- [4] AFTAB, M. *Building Bluetooth low energy systems*. Birmingham, UK: Packt Publishing, duben 2017. ISBN 9781786461087.
- [5] DRAHANSKÝ, M. *Dynamické biometrické vlastnosti*. 2005. Vysoké učení technické v Brně, Fakulta informačních technologií.
- [6] DRAHANSKÝ, M. *Úvod do biometrických systémů*. 2005. Vysoké učení technické v Brně, Fakulta informačních technologií.
- [7] HSU, Y., CHU, C., TSAI, Y. a WANG, J. An Inertial Pen With Dynamic Time Warping Recognizer for Handwriting and Gesture Recognition. *IEEE Sensors Journal*. 2015, roč. 15, č. 1, s. 154–163.
- [8] ISKN. [online]. [cit. 25. dubna 2020]. Dostupné z: <https://ya-webdesign.com/imgdownload.html>.
- [9] LECH, M. a CZYŻEWSKI, A. Handwritten Signature Verification System Employing Wireless Biometric Pen. In: Květen 2019, s. 307–319. ISBN 978-3-319-77603-3.
- [10] MADGWICK, S. O. H., HARRISON, A. J. L. a 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.
- [11] MADGWICK, S. *Open source IMU and AHRS algorithms* [online]. [cit. 30. května 2020]. Dostupné z: <https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/>.
- [12] MIYAGAWA, T., YONEZAWA, Y. a ITOH, K. *Handwritten Pattern Reproduction Using Pen Acceleration and Angular Velocity*. Citeseer, 2017.
- [13] NIHTIANOV, S. *Smart sensors and MEMS : intelligent devices and microsystems for industrial applications*. Duxford: Woodhead Publishing, únor 2018. ISBN 978-0081020562.
- [14] PAWLUS, J. *Zařízení pro napodobení statických a dynamických vlastností písma*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně Fakulta informačních technologií. Vedoucí práce prof. Ing. Martin Dražanský, Ph.D.

- [15] SCHWARZ, R. *Quaternions and Spatial Rotation* [online]. [cit. 30. května 2020]. Dostupné z: [https://downloads.rene-schwarz.com/download/M006-Quaternions\\_and\\_Spatial\\_Rotation.pdf](https://downloads.rene-schwarz.com/download/M006-Quaternions_and_Spatial_Rotation.pdf).
- [16] SEMICONDUCTOR, N. *Nordic Semiconductor Infocenter* [online]. [cit. 7. května 2020]. Dostupné z: <https://infocenter.nordicsemi.com/>.
- [17] STANLEY, M. Accelerometer placement - where and why. *NXP blog* [online]. 2012 [cit. 3. května 2020]. Dostupné z: <https://blog.nxp.com/tech-insights/accelerometer-placement-where-and-why>.
- [18] STRIETELMEIER, J. *LiveScribe Pulse Smartpen Review* [online]. [cit. 21. května 2020]. Dostupné z: [https://the-gadgeteer.com/2008/08/01/livescribe\\_pulse\\_smartpen\\_review/](https://the-gadgeteer.com/2008/08/01/livescribe_pulse_smartpen_review/).
- [19] TOHRU, M., YOSHIMICHI, Y., KAZUNORI, I. a MASAMI, H. Handwritten Pattern Reproduction Using Pen Acceleration and Angular Velocity. In: *IEICE Trans. Inform. Syst.* 2000, s. 1137–1140.
- [20] TOWNSEND, K., CUFÍ, C., AKIBA a DAVIDSON, R. *Getting started with Bluetooth low energy*. 1. vyd. Sebastopol, CA: O'Reilly, květen 2014. ISBN 978-1491949511.
- [21] UBLOX. *BMD-350* [online]. [cit. 6. května 2020]. Dostupné z: [https://cz.mouser.com/datasheet/2/1025/BMD-350\\_DataSheet\\_\(UBX-19033354\)-1729516.pdf](https://cz.mouser.com/datasheet/2/1025/BMD-350_DataSheet_(UBX-19033354)-1729516.pdf).
- [22] WANG, J. a CHUANG, F. An Accelerometer-Based Digital Pen With a Trajectory Recognition Algorithm for Handwritten Digit and Gesture Recognition. *IEEE Transactions on Industrial Electronics*. 2012, roč. 59, č. 7, s. 2998–3007.
- [23] YANIGER, S. I. Force Sensing Resistors: A Review Of The Technology. In: *Electro International, 1991*. 1991, s. 666–668.