



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
Katedra radioelektroniky

Bezdrátový motion capture sensor pro virtuální reality

Wireless motion capture for virtual reality

Bakalářská práce

Studijní program: Elektronika a komunikace

Vedoucí práce: doc. Ing. Stanislav Vítek, Ph.D.

Jakub Jirout
Říčany 2021

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Jirout** Jméno: **Jakub** Osobní číslo: **483217**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Bezdrátový motion capture sensor pro virtuální reality

Název bakalářské práce anglicky:

Wireless motion capture for virtual reality

Pokyny pro vypracování:

Cílem práce je návrh a implementace bezdrátového motion capture sensoru, kterým bude možné ovládat digitální dvojče ve virtuální (VR) nebo rozšířené realitě (AR). Při vypracování se řiďte následujícími pokyny:

- 1) prostudujte možnosti využití zachycení pohybu a orientace ve virtuálním prostoru
- 2) navrhňte a implementujte motion capture sensor s využitím IMU a vhodného mikrokontroléru
- 3) implementujte rozhraní pro připojení sensoru do vhodného VR/AR prostředí
- 4) zařízení otestujte

Seznam doporučené literatury:

- [1] JERALD, Jason. The VR book: Human-centered design for virtual reality. Morgan & Claypool, 2015.
- [2] NORTHROP, Robert B. Introduction to instrumentation and measurements. CRC press, 2018.
- [3] PANGILINAN, Erin; LUKAS, Steve; MOHAN, Vasanth. Creating augmented and virtual realities: theory and practice for next-generation spatial computing. " O'Reilly Media, Inc.", 2019.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Stanislav Vítek, Ph.D., katedra radioelektroniky FEL

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

Datum zadání bakalářské práce: **16.09.2021**

Termín odevzdání bakalářské práce: **04.01.2022**

Platnost zadání bakalářské práce: **19.02.2023**

doc. Ing. Stanislav Vítek, Ph.D.
podpis vedoucí(ho) práce

doc. Ing. Stanislav Vítek, Ph.D.
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

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 Říčanech dne 29.12.2021

podpis

Chtěl bych poděkovat svému vedoucímu doc. Ing. Stanislavu Vítkovi, Ph.D. za cenné rady, konstruktivní připomínky a vstřícnou komunikaci při vedení mé bakalářské práce. Dále bych chtěl poděkovat své rodině za trpělivost a pochopení.

Abstrakt

Motion capture se v posledních dvou desetiletích stal nedílnou součástí animovaných filmů a her. Existuje řada způsobů snímání lidského pohybu, které se liší náročností a cenou. Jeden z těchto způsobů je snímání pomocí IMU rozmístěných po těle, pro které není nutné mít k dispozici veliký prostor. V této práci jsem sestavil a následně naprogramoval levné zařízení, které používá IMU a tenzometry pro snímání pohybu ruky a ohybu prstů. K tomu jsem využil vývojovou desku Thunderboard Sense 2, která přenáší data pomocí technologie BLE do virtuálního prostředí v programu Blender, kde jsou zobrazena pomocí virtuálního dvojčete snímané ruky.

Klíčová slova: Motion capture, BLE, vývojová deska, snímání pohybu, IMU, Blender

Motion capture has become a vital part of animated films and videogames in the last two decades. There are many ways to capture human movement, their main differences are how complex they are and their price. One of them is tracking movement using IMUs placed on different parts of the body, which doesn't require considerable space to be used. In this paper I present how I constructed a cheap device, which uses IMU and flex sensors to track the movement of the hand and its fingers. I used an IoT development kit Thunderboard Sense 2 that uses BLE to transfer data to a virtual environment in Blender, where the data are being represented using a virtual twin of the tracked hand.

Keywords: Motion capture, BLE, IoT development kit, IMU, Blender

Obsah

1 Úvod	10
1.1 Cíl práce	10
1.2 Struktura práce	10
2 Analýza	11
2.1 Motion Capture	11
2.1.1 Historie	12
2.1.2 Optické systémy	13
2.1.3 Mechanické systémy	15
2.1.4 Magnetické systémy	15
2.1.5 Inerciální systémy	15
2.1.6 Využití	16
2.2 Vývojové desky	16
2.2.1 Požadavky	17
2.2.2 Příklady vhodných desek	18
3 Použité technologie	20
3.1 Vývojová deska	20
3.2 Sériová komunikace	20
3.3 BLE	22
3.3.1 GAP	22
3.3.2 Advertising	22
3.3.3 GATT	23
3.4 Blender	23
3.5 Simplicity Studio	24
4 Realizace	25
4.1 Zapojení	25
4.2 Thunderboard Sense 2	27
4.3 Zpracování dat a jejich vizualizace	29
4.3.1 Blender	29
4.3.2 Visual Studio	31
5 Závěr	32
5.1 Možná budoucí rozšíření	32
6 Zdroje	33

7	Seznam příloh	35
8	Příloha A: Návod k použití	36
9	Příloha B: Obsah elektronické přílohy	37

Seznam zkratek

ADC (Analog to Digital Converter)

BLE (Bluetooth Low Energy)

DOF (Degrees Of Freedom)

FPS (Frames Per Second)

GATT (Generic Attribute Profile)

GPS (Global Positioning System)

I2C (Inter Integrated Circuit)

IMU (Inertial Measurement Unit)

IoT (Internet of Things)

MISO (Master In Slave Out)

MOSI (Master Out Slave In)

RAM (Random Access Memory)

SCL (Synchronous Clock)

SCLK (Serial Clock)

SDA (Synchronous Data)

SNR (Signal to Noise Ratio)

SoC (System on Chip)

SPI (Serial Peripheral Interface)

SS (Slave Select)

UART (Universal Asynchronous Receiver-Transmitter)

USB (Universal Serial Bus)

VR (Virtual Reality)

Seznam obrázků

Obrázek 2.1: Motion capture během snímání a jeho výsledné zpracování [2]

Obrázek 2.2: Snímání pohybu herečky pomocí počítače ANIMAC [4]

Obrázek 2.3: Oblek se speciálním vytištěným vzorem [6]

Obrázek 2.4: Arduino Nano 33 BLE Sense [12]

Obrázek 2.5: EFR32BG22 Thunderboard kit [14]

Obrázek 3.1: Thunderboard Sense 2 [16]

Obrázek 4.1: Schéma pinů na vývojové desce [20]

Obrázek 4.2: Sestrojené zařízení

Obrázek 4.3: Hodnoty napětí na napájecích pinech

Obrázek 4.4: Příklad kódu sběru dat z IMU [22]

Obrázek 4.5: Algoritmus pro čtení dat přenesených přes BLE.

1 Úvod

Počítačová animace je proces vytváření iluze pohybu rychlým vykreslováním snímků, mezi kterými je malý rozdíl. Využívá se vlastnosti lidského oka, které je schopné reagovat na změny jen určitou rychlostí, tzv. setrvačnost lidského oka [1]. Pokud tedy oko vidí několik podobných snímků rychle za sebou, není schopné jednotlivé snímky rozeznat, což vytváří iluzi plynulého pohybu.

Jeden ze způsobů animování je vytvoření jednotlivých snímků ručně pomocí speciálního softwaru. Tento způsob se stále používá, je ale velice časově náročný. Existuje lepší možnost, která díky její praktičnosti a realistickým výsledkům nabyla na popularitě s názvem motion capture. Ta nám umožňuje snímat lidi, zvířata nebo různé objekty s velikou přesností a jejich pohyby přenášet do virtuálního světa.

1.1 Cíl práce

Cílem této práce je sestavit zařízení, které bude možné nasadit na ruku a tím snímat její pohyb a ohyb prstů. Vývojová deska a tenzometry budou umístěné na exoskeletonu na hřbetu ruky, který bude vytvořen na míru pro zvolenou vývojovou desku. Data budou v reálném čase posílána bezdrátovou technologií BLE do programu Blender na PC, kde budou vizualizována v podobě virtuálního dvojčete snímané ruky.

1.2 Struktura práce

Nejdříve provedu analýzu toho, co vlastně motion capture je a jakou má historii. Porovnáám různé druhy systémů snímání a stručně uvedu jaké využití tato technologie má napříč obory. Stanovím požadavky na výběr vývojové desky s jejíž pomocí budu pohyb snímat a tato data posílat do PC. Poté uvedu příklady desek, které splňují stanovené požadavky. V další kapitole rozeberu technologie, se kterými se v rámci mé práce setkám. Nakonec popíšu realizaci praktické části této práce, které je rozdělena na části.

2 Analýza

2.1 Motion Capture

Motion capture je proces snímání pohybu lidí nebo různých subjektů a jeho následné přenesení do virtuálního světa. Díky této technologii se herci mohou převtělít do fiktivních postav, což nejen nesmírně usnadňuje práci animátorů, ale zároveň to posouvá kvalitu a přirozenost animací na vyšší úroveň. Vytváří se tak nová kategorie herectví, protože si herci musí celou scénu představit ve studiu, kde jsou pouze nutné kulisy, které jsou součástí scény. K nahrávání je potřeba mít dostatečně velký prostor pro zvolenou inscenaci, nasazený speciální oblek a v určitých případech výkonný PC se správným softwarem pro zvolený systém senzorů. Snímání obličeje funguje na podobných principech jako snímání celého těla, ale spadá do vlastní kategorie.

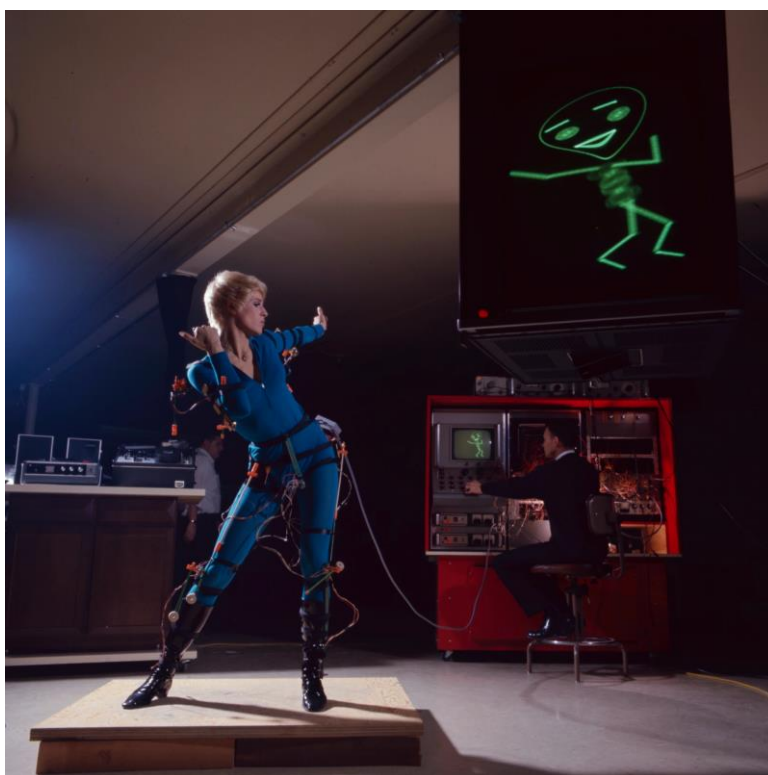
Důležitým pojmem je také tzv. *Performance capture*, který označuje snímání a zpracovávání pohybu, obličeje a hlasu najednou. Jedná se tedy o kompletní přenesení hereckého výkonu do virtuálního prostředí. Při jedné z forem *facial capture* mají herci kromě speciálního obleku také po celém obličeji tečky, které snímají kamery vyvedené z přilby na hlavě. V některých případech na jejich obličeji z velké blízkosti svítí silná světla pro bezchybné snímání zmíněných bodů.



Obrázek 2.1: Motion capture během snímání a jeho výsledné zpracování [2]

2.1.1 Historie

Jako úplný začátek motion capture se považuje animace Koko the Clown z roku 1915, kterou Max Fleischer vytvořil pomocí nové metody nazvané *rotoscoping*, která se používá dodnes. Ta spočívá v obkreslování snímků filmu, na kterém je zachycena inscenace či libovolný pohyb. Díky této nové metodě byly kreslené animace plynulé a přirozené [3]. Využilo jí několik slavných filmů jako například Popelka (1950), nebo Pán Prstenů (1978), seriálů i několik videoher. Poprvé se podařilo vytvořit počítačovou animaci založenou na pohybu člověka v reálném čase v roce 1967. V tomto roce sestrojil Lee Harrison III počítač „ANIMAC“, který dokázal díky speciálnímu obleku pohybovat s postavou na CRT obrazovce. Oblek snímal pohyb pomocí potenciometrů, které byly společně do počítače připojené.



Obrázek 2.2: Snímání pohybu herečky pomocí počítače ANIMAC [4]

Ke konci 80. let byl uveden do prodeje jeden z prvních veřejně dostupných obleků *The Data Suit* od firmy VPL Research, který využíval optických vláken na připojení senzorů k počítači. Ten dokázal zpracovat příchozí data 15 až 30krát za sekundu [5]. V moderní kultuře proslavil technologii motion capture herečkový výkon

Andyho Serkise, který ztvárnil slavnou postavu *Golluma* z trilogie *Pán Prstenů*. Bylo to poprvé, co se na plátně kin objevila zcela počítačem generovaná postava, která kopírovala lidské pohyby a výrazy v obličeji. V poslední době byl jedním z velkých pokroků v oblasti používání motion capture ve hrách titul *Hellblade: Senua's Sacrifice* od studia Ninja Theory, ve kterém bylo poprvé možné pohyby herečky v reálném čase přenášet do plně vykreslovaného herního světa. Je to další pokrok, který ušetří čas při tvorbě herních titulů, který se následně dá využít ke zlepšení jiných oblastí.

V dnešní době je motion capture běžná forma animace v zábavním průmyslu. Cena těchto systémů v posledních letech klesá a díky tomu je mohou využívat i malá studia nebo jednotlivci u sebe doma. Nejčastější formou domácího motion capture systému jsou VR headsety, které mají ve většině případů příslušenství na snímání pohybu rukou a případně i dalších částí těla.

2.1.2 Optické systémy

Optické systémy představují nejkvalitnější metodu motion capture. Jsou založené na zpracovávání dat z několika kamer rozmístěných po studiu nebo venku, kde probíhá natáčení. Snímaný subjekt má na všech částech těla upevněné reflektivní nebo vyzařující optické prvky (tzv. *marker*), které jsou snímány speciálními kamerami. Jejich snímání funguje na stejném principu jako lidský zrak; pokud vidíme jeden bod ze dvou přesně definovaných míst současně, dokážeme s velkou přesností určit jeho polohu.

Ale mohlo by se stát, že herec svým pohybem zastíní jeden nebo více markerů, což by znemožnilo jejich snímání. Aby se tomuto problému předešlo, má každé studio běžně mezi 2-48 kamerami. Existují i případy, kdy se toto číslo blíží ke 100. Pokud se ale i tak stane, že některý *marker* není v určitou chvíli dobře vidět, lze chybu následně nahradit pomocí softwarových algoritmů - tzv. „post-process“ úpravy.

Aby mohl software správně zpracovávat data z kamer, musí vědět jakou mají v daném prostoru orientaci a polohu. Tradiční řešení je kalibrace pomocí pravoúhlého trojúhelníku, na kterém jsou nalepené markery a tzv. Wand. Trojúhelník se položí na zem tak, aby ho měla každá kamera zhruba uprostřed zorného pole. Následně technik začne s Wand různě mávat ve snímaném prostoru

po dobu několika sekund. Jakmile software kalibraci dokončí, uvědomí technika a systém je připraven na snímání. Nicméně existuje alternativa v podobě obleků se speciálním vytištěným vzorem s názvem „Sierpiński triangle“. Díky tomuto vzoru není třeba kalibrovat kamery, protože software zpracovávající data z kamer jejich polohu zjistí podle tvaru zmíněných vytištěných vzorů, které jsou pod různými úhly zkreslené.



Obrázek 2.3: Oblek se speciálním vytištěným vzorem [6]

Markery jsou kuličky o velikosti dvou centimetrů až tří milimetrů [7], které jsou připevněné buď přímo na kůži, nebo suchým zipem na speciálním obleku. Chceme-li snímat statický objekt, jehož tvar známe, stačí aby na něm byly 3 markery reprezentující osy 3D prostoru. Markery se dále dělí na aktivní a pasivní.

Pasivní markery jsou potaženy retroreflektivním materiálem, který odráží světlo svítící okolo čoček kamer. Ty jsou nastaveny tak, aby ignorovaly všechny ostatní odrazy, které mají menší intenzitu než markery. Používají se pouze v podmínkách, kde lze kontrolovat osvětlení.

Aktivní markery fungují na principu svítících LED. Díky tomu, že vyzařují vlastní světlo je při snímání dobrý poměr SNR, což umožňuje snímat z větší vzdálenosti než v případě těch pasivních a tím i větší prostor. To také znamená, že tento způsob je vhodný pro venkovní podmínky.

Výjimkou jsou takzvané „markerless“ systémy, které využívají pokroky v oblasti počítačového vidění. V případě těchto systémů nejsou třeba žádné senzory na těle, protože je vše vyhodnocováno na základě vidění jedné nebo více kamer.

Složité algoritmy rozeznají tvary lidského těla a rozdělí ho do částí, jejichž pohyb následně sleduje.

2.1.3 Mechanické systémy

Mechanické systémy měří ohyb kloubů pomocí potenciometrů upevněných na exoskeletu, který si subjekt oblékne. Většinou se jedná o konstrukci z umělé hmoty, která může být vytisknutá pomocí 3D tiskárny a pomocí suchých zipů se připevní na tělo. Takový systém s sebou nese určité omezení pohybu pocházející z podstaty exoskeletů. Pokud bychom chtěli znát i polohu v prostoru, potřebovali bychom buď speciální algoritmy na odhad pozice na základě pohybu, nebo externí senzory.

2.1.4 Magnetické systémy

Magnetické systémy se skládají z vysílače a několika přijímačů. V každém přijímači jsou 3 cívky měřící změnu toku magnetického pole. Podle velikosti indukovaného napětí a proudu se určí orientace a pozice přijímače. Vzhledem k principu měření je tento systém velice náchylný na zkreslení údajů vlivem kovových předmětů v blízkosti senzorů, nebo pohybem na kraji snímaného prostoru.

2.1.5 Inerciální systémy

Inerciální systémy jsou založené na malých čípech zvaných IMU a doprovodných *fusion* algoritmech. V IMU je akcelerometr, gyroskop a někdy i magnetometr. Zmíněné senzory jsou ale velmi náchylné na *drift* a tím i na akumulovanou chybu. Kombinací dat z těchto senzorů lze s určitou přesností odhadnout orientaci. Nicméně levná IMU budou vždy mít drift, se kterým je potřeba počítat. Určování relativní polohy v reálném čase s pomocí samotného akcelerometru není možné [8]. Je nutné kombinovat data s jiným senzorem; nejlépe GPS nebo některým optickým systémem. Akcelerometr má veliký drift a jeho dvojitou integrací pro zjištění polohy se v rámci sekund akumuluje obrovská chyba, která se zvětšuje exponenciálně. Existují ale studie [9] na použití algoritmů

s využitím pravděpodobnostních modelů pro zvýšení přesnosti, které mají dobré výsledky, ale všechny využívají *post-process* úprav naměřených dat. Výhodou těchto senzorů je především jejich cena, ale i velikost a možnost snímání v jakémkoliv prostředí.

2.1.6 Využití

První praktické využití systémů motion capture bylo při výzkumu v oboru biomechaniky, kde vědci využili potenciometry pro studii ohýbání kolene [10]. Avšak dnes se tyto technologie používají v řadě dalších oborů:

- Zábavní průmysl – Kromě využití zmíněných ve 2. kapitole je ještě jedna kategorie, která je od nedávna na vzestupu s názvem Vtubing. Jedná se o tvoření internetových videí nebo streamů, ve kterých vystupuje virtuální postava ovládaná pomocí motion capture.
- Výzkum – Motion capture přináší nové způsoby studia pohybového ústrojí u lidí a u zvířat.
- Sport – S pomocí detailního sledování pohybu vrcholového sportovce dokáže trenér podrobně vysvětlit jaké chyby sportovec dělá a jak být efektivnější.
- Vojenský průmysl – Pomalu ale jistě proniká forma tréninku ve VR i do vojenské sféry.
- Zdravotnictví – Sledováním pohybového ústrojí mohou lékaři zjistit zdravotní problémy nebo naopak kontrolovat, zda léčba pacientů probíhá, jak má. VR systémy se také používají pro studijní účely. Studenti se například mohou učit, jak probíhá operace, aniž by bylo třeba opravdového pacienta.

2.2 Vývojové desky

Vývojové desky jsou desky plošných spojů obsahující mikroprocesor, paměť, vstupní a výstupní porty ve formě pinů nebo konektorů a často také LED diody, různé senzory a případně tlačítka. Komunikace mezi moduly na desce a ostatními moduly nebo periferiemi je zprostředkována pomocí rozhraní jako I2C,

UART nebo SPI. Nejdůležitější součástí je debugger, který je integrovaný na desce. Ten umožňuje komunikaci s procesorem a dává mu instrukce. Na stránkách výrobce lze stáhnout programovací prostředí, které v sobě má API nutné k naprogramování vývojové desky.

2.2.1 Požadavky

K dosažení cílů této práce existují určité požadavky na zvolený hardware. Vlastnosti vývojových desek se dělí do určitých kategorií. Je třeba zvážit jaké máme požadavky a podle toho vybrat takovou, která bude specifikacemi vyhovovat. Je několik oblastí, ve kterých se vývojové desky liší.

- Velikost – Ideálně by měla být co nejmenší, aby zabírala jen část hřbetu ruky. V takovém případě by výsledné zařízení mohlo mít poměrně malé rozměry.
- Napájení – Vzhledem k tomu, že cílem práce je bezdrátové zařízení, je nutné mít možnost napájení z baterie. Bylo by vhodné mít k dispozici slot na baterii, aby bylo zařízení kompaktní.
- Konektivita – Nutností je USB port, aby se vývojová deska dala programovat z PC. Stejně tak je potřeba, aby měla I/O piny na analogový vstup, zem a napájení. Co se týče bezdrátových technologií, jsou na výběr WiFi, Bluetooth nebo BLE. Je třeba vzít v úvahu, že deska bude napájena z baterie, tudíž musíme dbát na úsporu energie. WiFi není vhodná, protože narozdíl od Bluetooth není navržena pro peer-to-peer komunikaci, což má také za následek vysokou spotřebu energie. Nejlepší volbou bude BLE, které funguje na podobném principu jako Bluetooth, ale je navrženo s ohledem na nízkou spotřebu. Teoretický maximální dosah signálu má méně než 100 m, což je dostačující pro domácí studio. Aby se výsledná animace zdála plynulá, je nutné mít nové údaje alespoň 30krát za sekundu. Odeslaná data se budou skládat z 8 čísel o velikosti 16 bitů, z toho 3 pro orientaci, 3 pro akcelerometr a 2 pro snímání prstů. To znamená, že objem dat v jednom přenosu bude 128 bitů 30krát za sekundu, což činí 3,84 kbit/s. Propustnost BLE je 125-500 kbit/s, tedy dostatečná.

- Výkon – Na jednu stranu se budou na desce velice rychle zpracovávat data, ale zároveň by výkonný procesor měl velkou spotřebu.
- Sensory – Pro účely této práce je potřeba mít gyroskop a akcelerometr, které jsou v IMU, a případně magnetometr. Také je nutné, aby v sobě deska měla ADC.
- Cena – Neměla by být příliš drahá. Ideální cena by se měla pohybovat pod 1 000 Kč.

2.2.2 Příklady vhodných desek

Arduino Nano 33 BLE Sense [11]



Obrázek 2.4: Arduino Nano 33 BLE Sense [12]

- Velikost - 45 x 18 mm
- Napájení – z externího zdroje
- Konektivita - 14 I/o pinů, USB, BLE
- Výkon – ARM Cortex-M4 s frekvencí 64MHz
- Sensory – Světelný senzor, mikrofon, 9DOF IMU a senzor teploty, vlhkosti a tlaku vzduchu
- Cena - 934 Kč

EFR32BG22 Thunderboard Kit [13]



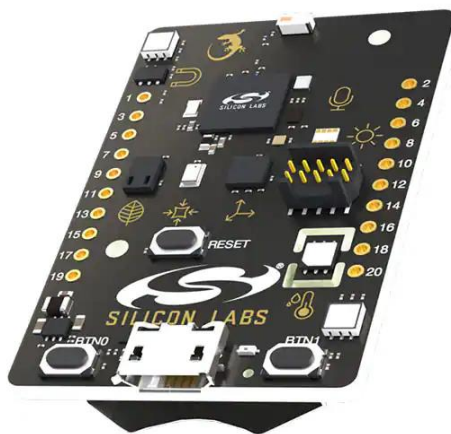
Obrázek 2.5: EFR32BG22 Thunderboard kit [14]

- Velikost – 45 x 30 mm
- Napájení – slot na CR2032 baterii, nebo externí napájení
- Konektivita - 20 I/o pinů, USB, Bluetooth 5.2 s podporou BLE a určování směru ke zdroji rádiových vln.
- Výkon – ARM Cortex-M33 s frekvencí 76.8MHz
- Senzory – senzor vlhkosti a teploty, UV záření a světla, magnetometr a 6DOF IMU
- Cena – okolo 600 Kč

3 Použité technologie

3.1 Vývojová deska

Pro tuto práci jsem použil IoT vývojový kit Thunderboard Sense 2 of firmy Silicon Labs. Jedná se o kompaktní, výkonnou desku, která svými specifikacemi [15] vyhovuje stanoveným požadavkům. Díky slotu na baterii a malým rozměrům je možné ji umístit přímo na hřbet ruky, což umožní malé rozměry výsledného zařízení. Procesor s nízkou frekvencí zajistí delší životnost baterie. Deska byla k dispozici ve škole, což je jeden z důvodů, proč mi ji vedoucí práce doporučil.



Obrázek 3.1: Thunderboard Sense 2 [16]

- Velikost - 45 x 30 mm
- Napájení – slot na CR2032 baterii nebo externí napájení
- Konektivita - 20 I/o pinů, USB, Zigbee, Thread a BLE
- Výkon – ARM Cortex-M4 s frekvencí 38.4MHz
- Senzory – senzor vlhkosti a teploty, plynu, UV záření a světla, tlaku, kvality vzduchu, magnetometr, mikrofon a 6DOF IMU

3.2 Sériová komunikace

Sériová forma komunikace je v dnešní době jedna z nejpoužívanějších. Pro přenos dat využívá pouze jeden vodič v jednom směru komunikace, po kterém

postupně posílá jednotlivě každý bit. Přenos dat může probíhat synchronně nebo asynchronně. Synchronní využívá synchronizovaných hodin na straně vysílače (TX) a přijímače (RX), což umožňuje data posílat v konkrétních časových intervalech. Pro asynchronní přenos je třeba zajistit jiný způsob, jak data na straně přijímače spolehlivě číst. V této práci se využívá asynchronní komunikace.

Často používané asynchronní sběrnice:

- UART – Pro zajištění přenosu dat je třeba aby vysílač i přijímač měly stejnou konfiguraci. Je třeba nastavit rychlost přenosu (*baud rate*), velikost odesílaných dat a konfiguraci start a stop bitů. Tyto bity, jak už vyplývá z jejich názvu, jsou bity, které určují, kdy má přijímač začít číst přijímaná data a kdy přenos skončil. *Baud rate* říká, jak dlouho trvá poslání jednoho bitu. Nejčastější rychlosti jsou 115200 a 9600 baudů [17], což znamená 9600 bitů za sekundu, tzn. přenos jednoho bitu trvá zhruba 104 mikrosekund.

- I2C – Využívá dvou drátů, jeden pro přenos dat (SDA) a druhý pro hodinový signál (SCL). Na přijímači a vysílači je napřed třeba nakonfigurovat délku posílaných dat a frekvenci hodinového signálu. Tato sběrnice funguje na principu *multimaster*. To znamená, že každé připojené zařízení může být master i slave. Aby se data dostala do správného zařízení, *master* nejdříve pošle unikátní adresu koncového zařízení a až poté samotná data. Jedná se o spojení Half-duplex, tedy komunikace může probíhat v obou směrech, ale ne současně.

- SPI – Komunikuje pomocí 4 drátů:
 - SCLK
 - MOSI
 - MISO
 - SS

Jedná o sběrnici typu *single master*. To znamená, že jedno zařízení je *master* a ostatní jsou *slave*. *Master* používá piny SS aby předal konkrétnímu zařízení informaci, že mu bude posílat data. Komunikace je realizována pomocí

dvou drátů pro data (MOSI, MISO) a hodinový signál (SCLK). Spojení je Full-duplex, tedy komunikace může probíhat v obou směrech současně.

3.3 BLE

Bluetooth Low Energy je komunikační technologie zaměřená na IoT. Tedy zejména zařízení napájena z baterie, která vysílají malé objemy dat. Bluetooth používané u běžných aplikací, celým názvem Bluetooth Classic, má oproti BLE vysokou spotřebu, tudíž není vhodné pro malé zdroje napájení. Sice mezi sebou nejsou kompatibilní, ale oba protokoly mohou využívat stejnou anténu, a tudíž většina zařízení s Bluetooth Classic podporuje i BLE. Stejně jako tradiční Bluetooth, BLE komunikuje pomocí 2.4 GHz antény v pásmu 2402 MHz až 2480 MHz, které je rozděleno na 40 kanálů s mezerami 1 MHz [18]. Avšak ve stejném pásmu fungují i jiné technologie jako například WiFi, a proto BLE při komunikaci využívá tzv. “*frequency-hopping*”, což znamená, že se během komunikace rychle mění kanály (frekvence nosné vlny) přes které se data posílají. Díky tomu může existovat několik systémů využívajících stejné frekvenční pásmo na jednom místě.

3.3.1 GAP

GAP je protokol, který řídí připojení dvou zařízení a *advertising*. Stará se o vzájemnou viditelnost zařízení a o to, jakým způsobem spolu mohou komunikovat. Tento protokol definuje dvě kategorie zařízení; *Peripheral* a *Central*.

- *Peripheral* – Ve většině případů jeden nebo více senzorů. Jedná se o malé, často baterií napájené zařízení, které není schopné větších výpočetních operací ať už kvůli úspoře baterie, nebo z důvodů nízkého výkonu.
- *Central* – Větší a výkonnější zařízení, které zpracovává data a zároveň řídí navázanou komunikaci.

3.3.2 Advertising

Advertisements jsou až 31 bajtové pakety vysílané zařízením, které chce být viditelné pro své okolí. Tím dává najevo, že je možné s ním navázat spojení, tedy se spárovat. Tyto pakety obsahují název zařízení, sílu signálu a seznam UUID

identifikátorů, které jsou v zařízení k dispozici. Pro tyto účely se využívají kanály 37, 38 a 39, kterými se střídavě posílají *advertising* pakety. Neustálé posílání těchto paketů by ale způsobilo velkou spotřebu energie. Proto se zavedl pojem „Advertising interval“, který určuje, jak často se tyto pakety vysílají. Zvolí se takový časový úsek, aby byla vhodná rovnováha mezi spotřebou energie a spolehlivostí konektivity zařízení.

Jakmile *Central* zařízení přijme *advertising* paket, může si říct o dodatečná data, kterým se říká “Scan Response” pakety. Ty mají stejnou velikost, ale umožňují poslat dodatečné informace. Tato funkce se dá využít například v případě, že chceme posílat menší objemy dat více zařízením najednou. K tomu se právě využívají *advertising* pakety, protože u BLE může *peripheral* zařízení navázat komunikaci pouze s jedním *central* zařízením.

3.3.3 GATT

Jakmile se naváže spojení a zařízení jsou spárována, přichází na řadu přenos dat. O to se stará protokol GATT, který jim mimo jiné přiřazuje identifikátory s označením UUID. Tato data se jmenují *Services*, *Characteristics* a *Descriptors* a společně se označují jako *attributes*.

- *UUID* – Jedinečné 128bitové číslo, které má každý atribut. Při práci s daty v rámci BLE komunikace se používají právě tyto identifikátory.
- *Services* – Používá se pro seskupení/organizaci *characteristics*, které dohromady popisují určitý jev.
- *Characteristics* – Obsahují data, která jsou přístupná ze strany klienta. Zároveň obsahují *Descriptors* a nastavení toho, co se bude s daty dít. Nejdůležitější z možností jsou *Read*, *Write* a *Notify*.
- *Descriptors* – Dodatečné informace o *Characteristic* jako například název, *UUID* nebo velikost dat.

3.4 Blender

Blender je open source 3D software pro grafickou tvorbu od modelování, přes animace až po tvorbu her. Lze v něm vytvářet tzv *armatures* a *bones*, tedy

kostry, které lze přiřadit k modelu a tím s ním pohybovat. Tyto objekty lze následně ovládat pomocí skriptů, které se vytváří v jazyce Python. Blender je velice populární mezi individuálními tvůrci a malými studiemi. To má za následek, že je na internetu mnoho vzdělávacích materiálů a diskusí na fórech, které řeší specifické problémy. V tomto programu jsem modeloval exoskelet pro uchycení vývojové desky a dalších senzorů, který se vytiskl pomocí 3D tiskárny. Následně jsem ho použil pro zpracování a vizualizaci dat naměřených pomocí senzorů na ruce.

3.5 Simplicity Studio

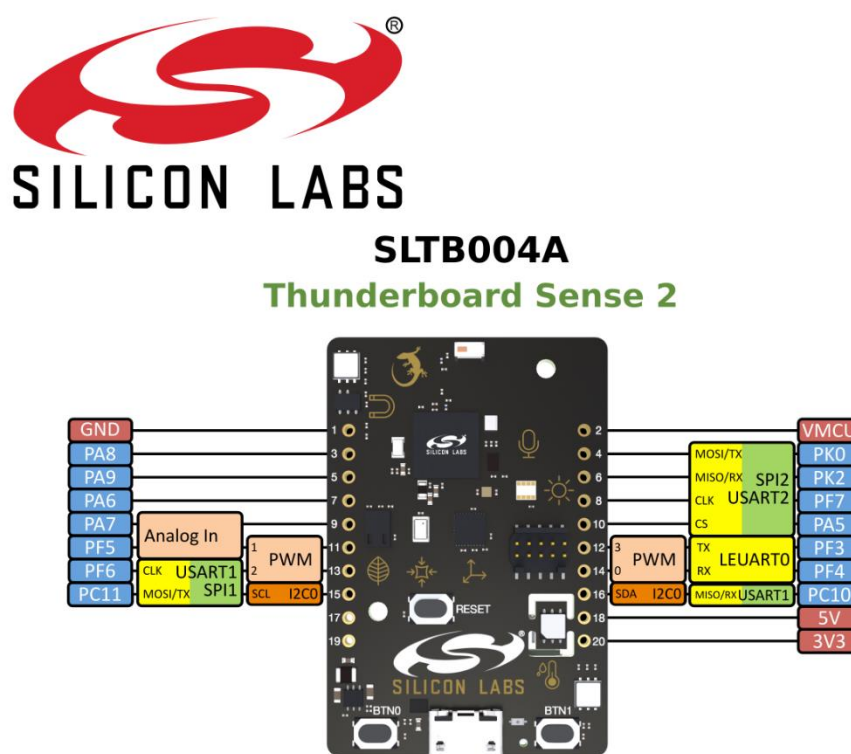
Pro naprogramování zvolené vývojové desky jsem mohl použít buď webové vývojové prostředí mbed, nebo právě Simplicity Studio, pro které jsem se nakonec rozhodl, protože je to software přímo navržený pro zvolený hardware. Prostředí mbed podporuje velké množství desek od různých výrobců, tudíž není zaručena úplná kompatibilita. Simplicity Studio je vývojové prostředí od firmy Silicon Labs určené pro jejich produkty. Nabízí několik příkladů využití funkcí jejich vývojových desek spolu s kompletní dokumentací ostatních produktů. Instalace obsahuje všechny potřebné knihovny a funkce pro ovládání celé desky a jejich možných periférií. Na stránkách firmy je k nalezení popis celého API.

4 Realizace

4.1 Zapojení

Základem je 3D vytištěný exoskelet, který je na hřbetu ruky. Zhruba uprostřed je kolébka navržená na velikost použité vývojové desky, která je jištěná pruhem textilu připevněným suchým zipem. Na stranách exoskeletu jsou otvory na uchycení popruhu, aby zařízení drželo na ruce a na přední straně jsou sedla pro tenzometry. Na prstech jsou umístěné kroužky, ve kterých jsou zasunuty senzory ohybu [19].

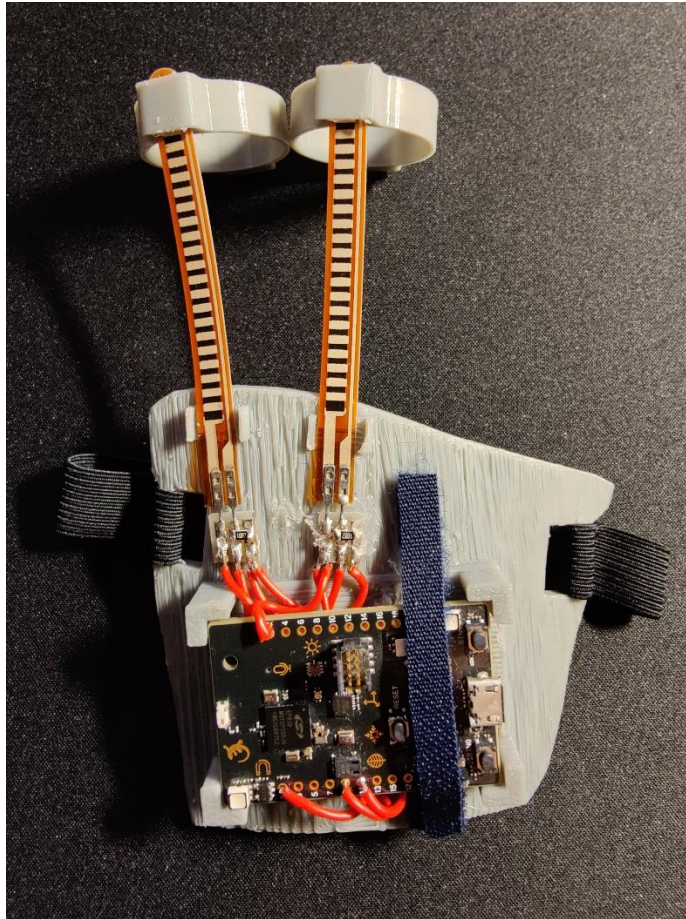
Z příslušných pinů na desce jsou vyvedeny vodiče pro napájení a snímání dat z tenzometrů. Ty jsou připojené pomocí děličů napětí realizovaných na částech desky univerzálních plošných spojů. Tenzometry fungují na principu rezistoru, tudíž jsem v kombinaci s jedním SMD rezistorem sestrojil napěťový dělič, ze kterého lze snímat signál. Napětí a zem jsou mezi děliči propojeny, aby nebylo třeba mít na desce dva dráty na jednom pinu.



Notes:

- UART0/SPI0 in use by stdout on PA0/PA1
- The EFR32MG12 pinout is very flexible - this is just a recommendation.

Obrázek 4.1: Schéma pinů na vývojové desce [20]



Obrázek 4.2: Sestrojené zařízení

Pro dělič jsem zvolil rezistor o hodnotě 10 k Ω . Tenzometr má odpor od 30 k Ω do zhruba 180 k Ω . Záleží na jeho ohybu. Napájení je přivedené z pinu *VMCU*, na kterém jsou různé hladiny napětí podle toho, jestli je deska napájena z USB portu, nebo z CR2032 baterie.

Table 3.1. Thunderboard Sense 2 Power Options

Supply Mode	VIN	VMCU	3V3	5V0
USB power	4.5 - 5.5 V	On-board regulator	On-board regulator	USB VBUS
CR2032 battery	2.0 - 3.3 V	Battery voltage	Turned off and isolated	No voltage present
External battery	2.0 - 3.3 V	Battery voltage	Turned off and isolated	No voltage present
Mini Simplicity	2.0 - 3.3 V	Debugger dependent	Turned off and isolated	No voltage present

Obrázek 4.3: Hodnoty napětí na napájecích pinech

Kde „On-board regulator“ je nastavený na 3.3 V.

4.2 Thunderboard Sense 2

Pro programování vývojové desky jsem zvolil již zmíněné Simplicity Studio, které pracuje s jazykem C. Pomocí API na stránkách výrobce a vzorovým kódům, které jsou k dispozici v IDE a na internetu jsem naprogramoval všechny potřebné funkce.

Při startu se nejprve všechny relevantní funkce, konfigurace a proměnné inicializují. Následně začne kontrolovaná nekonečná smyčka, ve které probíhá veškeré sbírání, kalkulace a následné odesílání dat. Kód v této práci se dá rozdělit na čtyři části: IMU, ADC, Bluetooth a sériová komunikace.

Pro sběr dat z IMU jsem převzal kód z dokumentace výrobce [21]. Cílem je získat data určující orientaci a zrychlení. Pro tyto účely jsem inicializoval globální 16bitová pole s názvy *ovec* pro orientaci a *avec* pro akcelerometr. Jako první je třeba modul inicializovat a nakonfigurovat frekvenci, která určuje, jak rychle se mají zpracovávat data z příslušných senzorů. Je vhodné během této fáze zkalibrovat gyroskop. V nekonečném cyklu jen použiji funkci, která do určeného pole dá naměřené hodnoty.

Nicméně data určující orientaci jsou výsledkem tzv. *fusion* algoritmu, který kombinací dat z gyroskopu a akcelerometru sníží vliv šumu v naměřených datech a tím zvětší přesnost výsledné veličiny. V tomto případě dojde i k převedení naměřených dat na Eulerovy úhly určující orientaci vztahenou k souřadnicovému systému Země. Bohužel jsem nenašel žádnou dokumentaci popisující, jaký algoritmus Silicon Labs používá. Často používaný *fusion* algoritmus je *Kalman filtr*, který kromě kombinace dat využívá i pravděpodobnostních modelů.

```

#include "sl_imu.h"

int main( void )
{
    ...

    int16_t o_vec[3];

    sl_imu_init();

    // Configure sample rate
    sl_imu_configure(10);

    // Recalibrate gyro
    sl_imu_calibrate_gyro();

    while (true) {

        sl_imu_update();

        while (!sl_imu_is_data_ready()) {
            // wait
        }

        sl_imu_get_orientation(o_vec);

        ...
    }
}

```

Obrázek 4.4: Příklad kódu sběru dat z IMU [22]

Snímání ohybu prstů je realizované pomocí tenzometrů, které jsou do desky připojené jako analogový vstupní signál. Tento signál je následně zpracován pomocí ADC, které převede analogový signál (v tomto případě napětí) na digitální. Použitý převodník je 12bitový, je tedy schopný rozeznat $2^{12}-1$ úrovní. Při inicializaci převodníku jsou pro tuto práci důležité 3 údaje:

- Vstup – zda bude stejnosměrný, nebo střídavý.
- Vnitřní napěťová reference pro měřené veličiny – možnosti jsou: 1.25 V, 2.5 V, napájecí napětí, nebo externí reference. Převodník dokáže rozpoznat pouze signály menší, než je stanovená reference.
- Rozlišení převodníku – k dispozici je 6, 8 nebo 12 bitů.

Signál z tenzometrů je stejnosměrný. Rozlišení je nastavené na 12 bitů pro plynulost pohybu, ačkoliv by bylo menší rozlišení dostačující. Vzhledem k tomu, že nejvyšší teoretická hodnota na děliči může být 825 mV, stačí nastavit vnitřní napěťovou referenci na 1.25 V. V nekonečném cyklu volám funkci, která udělá jednu konverzi signálu z obou tenzometrů a tu zapíše do pole *inputs*. Kód je převzatý z Git repozitáře firmy Silicon Labs [23].

Pro funkčnost bluetooth je nutné vytvořit „example project“ s názvem „Bluetooth – SoC Empty“. Ten obsahuje základní strukturu kódu pro fungování bezdrátové komunikace, tedy advertising, strukturu pro odesílání dat a další konfigurace, které nejsou na první pohled vidět. Zařízení se ukazuje pod názvem „Motion device“. Celý princip Bluetooth komunikace v tomto prostředí funguje na principu tzv. *interrupt*. To znamená, že určité funkce mohou aktivovat svůj definovaný *interrupt*, který spustí konkrétní blok kódu. To má na starost konstrukce switch, jejíž *cases* odpovídají naprogramovaným *interrupts*.

Z toho důvodu jsem použil časovač nastavený na 10 ms, který se aktivuje, když se k zařízení připojí klient. Jakmile uběhne nastavený čas, spustí se blok kódu, ve kterém se všechna naměřená data v konkrétním pořadí zapíšou do společného pole a následně se pomocí funkce *sl_bt_gatt_server_write_attribute_value()* přiřadí ke konkrétní charakteristice v GATT databázi. Data se následně posílají ve formě bitového pole, které je třeba u klienta správně přečíst.

Sériová komunikace přes USB funguje na principu virtuálního COM portu. Simplicity Studio umí nakonfigurovat posílání výstupů z konzole přes sériový port. Jako metodu přenosu jsem zvolil posílání dat ve formě znaků, které na straně klienta zvládnou spolehlivě přečíst.

4.3 Zpracování dat a jejich vizualizace

4.3.1 Blender

V Blenderu jsem pomocí „bones“ vytvořil napodobeninu ruky, kterou ovládám přes skript. Data přijímám přes sériový port pomocí knihovny *pyserial*. Ve skriptu používám knihovnu *threading*, protože je na pozadí spuštěné hlavní vlákno programu, které bych jinak blokoval, tudíž by se nic nevykreslovalo. Dalším důvodem použití této knihovny je to, že data čtu pomocí nekonečné smyčky, kterou bych bez druhého vlákna nemohl zastavit. Přenášená data jsou ve tvaru „120X_“, kde čísla jsou hodnoty ze senzorů, písmeno definuje, jakou hodnotu číslo reprezentuje a podtržítka naznačuje konec, tedy že se mají znaky předtím přeložit a zapsat do příslušné proměnné.

Nejjednodušší způsob, jak reprezentovat rotaci jsou tzv. Eulerovy úhly. V případě 3D prostoru to jsou 3 úhly v radiánech značící rotaci kolem příslušné osy. Tento způsob má však své nedostatky. Nejen že závisí na pořadí aplikace rotace, ale také existuje situace, která se nazývá „Gimbal lock“. Může se totiž stát, že se jedna osa zarovná s některou ze zbylých os. Kvůli tomu začnou obě reprezentovat stejnou rotaci a tím se ztratí jeden stupeň volnosti, což není žádoucí. Jeden ze způsobů, jak tomu předejít je reprezentace rotace pomocí kvaternionů. Kvaterniony jsou čtyřdimenzionální komplexní čísla. Mají 4 složky nazývané w, x, y, z, které mají hodnoty od 0 do 1. Každá z těchto složek reprezentuje konkrétní rotaci. Kombinací těchto rotací a následnou normalizací výsledného vektoru získáme libovolnou rotaci ve 3D prostoru.

Hodnoty orientace z IMU jsou ve stupních. Tudíž je nejdříve převádím na radiány a následně podle převzatých goniometrických vzorců na kvaterniony. Ty potom předávám vykreslovacímu *engine*, který danou rotaci aplikuje na vytvořenou *armature*.

V této práci data z akcelerometru nepoužívám, ale v rámci potenciálního budoucího rozšíření jsem a naimplementoval algoritmus na odstranění vlivu gravitace z naměřených dat. Ten využívá již naměřené hodnoty orientace ve formě kvaternionů. Data naměřená akcelerometrem jsou vztažená k souřadnicovému systému daného senzoru. To znamená, že v danou chvíli se vliv gravitace projeví na osách, které směřují k zemi. To se dá vyřešit zrotováním vektoru gravitačního zrychlení do souřadnicového systému senzoru a jeho následným odečtením od naměřených hodnot.

Na výstupu z ADC na vývojové desce jsou hodnoty v mV. Při spuštění skriptu v Blenderu nejprve provádím kalibraci dat z tenzometrů. Tou zjistím maximální a minimální hodnoty měřeného napětí, když jsou prsty ohnuté nebo natažené. Tyto údaje použiji k odhadnutí ohybu prstů následující rovnicí

$$\left(\frac{\text{naměřená hodnota} - \text{minimální hodnota}}{\text{maximální hodnota} - \text{minimální hodnota}} \cdot 90 \right) - 90 \quad (4.1)$$

Jako maximální úhel ohybu jsem zvolil 90 stupňů. K rotaci používám Eulerovy úhly, protože otáčení probíhá pouze kolem jedné osy, takže případný „Gimbal lock“ mě

nijak neomezí. Nakonec odečítám 90 stupňů z důvodu korekce natočení při vykreslování.

4.3.2 Visual Studio

Visual Studio slouží pouze jako vizualizace dat přenesených přes BLE. Komunikace přes bluetooth je asynchronní operace, proto je třeba použít knihovnu *asyncio*, aby program věděl, že má na odpověď od serveru čekat. Komunikace mezi zařízeními je zprostředkována pomocí knihovny *bleak*. V případě této práce dopředu znám *UUID* dat, která chci číst a také znám MAC adresu vývojové desky, se kterou pracuji. Takže program dopředu ví, k jakému zařízení se připojit a jaká data číst.

Přijímaná data jsou ve formě bajtového pole, ve kterém je každé číslo rozdělené na jednotlivé bajty. Posílaná čísla mají 16 bitů, tedy 2 bajty. Abychom z nich mohli zase mít čitelná data, je třeba vybrat všechny bajty jednoho čísla a ty převést na datový typ *int*. Jak již bylo zmíněno, čísla na straně vývojové desky jsou posílána v konkrétním pořadí, aby bylo možné je přečíst na druhé straně. Následující algoritmus prochází pole přijatých dat, překládá je a následně je ukládá do proměnných.

```
while True:
    data = await client.read_gatt_char(UUID_sensor_data)
    values = [x for x in data]
    for x in range(3):
        temp = [values[0 + (x * 2)], values[1 + (x * 2)]]
        ovec[x] = int.from_bytes(temp, byteorder='little', signed=True)

    for x in range(3):
        temp = [values[6 + (x * 2)], values[7 + (x * 2)]]
        avec[x] = int.from_bytes(temp, byteorder='little', signed=True)

    for x in range(2):
        temp = [values[12 + (x * 2)], values[13 + (x * 2)]]
        fingers[x] = int.from_bytes(temp, byteorder='little', signed=True)
```

Obrázek 4.5: Algoritmus pro čtení dat přenesených přes BLE.

5 Závěr

Cílem této práce bylo navrhnout a naprogramovat bezdrátové motion capture zařízení pro snímání pohybu ruky a jeho následná vizualizace ve virtuálním prostředí. Po analýze vhodných zařízení pro tuto práci a konzultaci s vedoucím práce byla vybrána vývojová deska Thunderboard Sense 2 od firmy Silicon Labs. Praktická část se skládala ze čtyř segmentů: 1. vytvoření podpůrné desky, 2. připojení senzorů ohybu prstů, 3. naprogramování funkce pro čtení dat ze všech relevantních senzorů a jejich odesílání pomocí bezdrátové technologie Bluetooth, 4. zpracování a vizualizace snímaných dat ve virtuálním prostředí.

Ve 2. kapitole byly prozkoumány možnosti snímání pohybu spolu se stručnou historií této technologie. Také byly stanoveny požadavky pro vývojovou desku a následně byla nalezena potenciálně vhodná zařízení. Kapitola 3 popisuje technologie použité v této práci. Obsahem 4. kapitoly je praktická část této práce. Je v ní popsáno, jakým způsobem jednotlivé části fungují a jak vypadá výsledné zařízení.

Největším přínosem pro mne v této práci bylo seznámení se se světem motion capture z technického pohledu. Zjistil jsem jakým způsobem fungují různé způsoby snímání pohybu objektů a s jakými překážkami se lze setkat při vývoji takových systémů.

Bohužel během praktické části jsem zjistil, že knihovna *bleak* v programu Blender nefunguje, tudíž vizualizace v tomto programu je možná pouze při připojení desky přes USB kabel. Kód na vývojové desce není přizpůsobený pro komunikaci přes BLE a USB zároveň, proto pro sběr dat přes USB je třeba se k desce připojit přes Bluetooth, aby skončila *advertising* fáze, která jinak blokuje vše ostatní.

5.1 Možná budoucí rozšíření

Dalším logickým krokem ve vývoji by bylo vytvořit systém pro sledování pozice. Toho by se dalo docílit kombinací dat z IMU a z GPS modulu, nebo z kamer podobně jako u optických systémů. Nebo by se mohla sestavit další zařízení na snímání ostatních částí těla, což by mohlo nahradit dodatečné technologie pro sledování pozice, protože bychom znali rotaci více částí těla a vztahy mezi nimi, což by umožnilo odhadnout pohyb v prostoru.

6 Zdroje

1. Setrvačnost oka, Eduportál Techmania [online]. Dostupné z: <https://edu.techmania.cz/cs/encyklopedie/fyzika/svetlo/lidske-oko/setrvacnost-oka> [cit. 29.12.2021].
2. Převzato z: <https://variety.com/wp-content/uploads/2017/07/weta-digital-war-for-the-planet-of-the-apes.jpg?w=681&h=383&crop=1>
3. Motion capture for animation: the fascinating history behind the movies we know today [online]. Dostupné z: <https://motionanalysis.com/blog/an-evolution-of-motion-capture-the-fascinating-history-behind-the-movies-we-know-today/> [cit. 29.12.2021].
4. Převzato z: https://video-images.vice.com/articles/58cfed1b6b613327a77b43fc/lede/1490021676883-GettyImages-589142904.jpeg?crop=1xw:0.556930693069307xh:center:center&resize=900:*
5. Mo-cap suit - Wikipedia. [online]. Dostupné z: https://en.wikipedia.org/wiki/Mo-cap_suit [cit. 29.12.2021].
6. Why do MoCap suits use Sierpinski triangles? Reddit [online] Dostupné z: https://www.reddit.com/r/vfx/comments/f4axl2/why_do_mocap_suits_use_sierpinski_triangles/ [cit. 29.12.2021].
7. Motion capture – Wikipedie. [online]. Dostupné z: https://cs.wikipedia.org/wiki/Motion_capture [cit. 29.12.2021].
8. imu position without GPS or camera. Robotics Stack Exchange [online]. Dostupné z: <https://robotics.stackexchange.com/questions/8857/imu-position-without-gps-or-camera> [cit. 29.12.2021].
9. RIDI: Robust IMU Double Integration [online]. Dostupné z: <https://yanhangpublic.github.io/ridi/index.html> [cit. 29.12.2021].
10. A Brief History of Motion Tracking Technology and How it is Used Today by Rachel Lum, Medium [online]. Dostupné z: <https://medium.com/@lumrachele/a-brief-history-of-motion-tracking-technology-and-how-it-is-used-today-44923087ef4c> [cit. 29.12.2021].

11. Arduino Nano 33 BLE Sense, HWKitchen.cz [online]. Dostupné z: <https://www.hwkitchen.cz/arduino-nano-33-ble-sense/#parameters> [cit. 29.12.2021].
12. Převezato z: https://cdn.myshoptet.com/usr/www.hwkitchen.cz/user/shop/detail_alt_1/3384-1_arduino-nano-33-ble-sense.jpg?60dc2ad0 [cit. 29.12.2021].
13. SLTB010A, cz.farnell.com [online]. Dostupné z: <https://cz.farnell.com/silicon-labs/sltb010a/design-kit-accessories/dp/3370623> [cit. 29.12.2021].
14. Převezato z: https://cz.farnell.com/productimages/standard/en_GB/3370623-40.jpg
15. IoT Development Kit - Thunderboard Sense 2 - SLTB004A - Silicon Labs, Silicon Labs [online]. Dostupné z: <https://www.silabs.com/development-tools/thunderboard/thunderboard-sense-two-kit> [cit. 29.12.2021].
16. Převezato z: <https://www.e-komponent.com/sltb004a-thunderboard-sense-2-iot-kit-rf-haberlesme-kitleri-silicon-labs-1943-56-O.jpg>
17. Most common baud rates table, Lulu's blog [online]. Dostupné z: <https://lucidar.me/en/serialib/most-used-baud-rates-table/> [cit. 29.12.2021].
18. BLE frequency channels, ResearchGate [online]. Dostupné z: https://www.researchgate.net/figure/BLE-frequency-channels-It-can-be-noticed-that-channels-from-0-36-are-assigned-to-data_fig3_321800210 [cit. 29.12.2021].
19. Převezato z: <https://hackaday.io/project/86512-graspex> [cit. 29.12.2021].
20. Převezato z: <https://os.mbed.com/media/uploads/stevew817/tb004.png>
21. Převezato z: <https://docs.silabs.com/gecko-platform/latest/hardware-driver/api/group-imu> [cit. 29.12.2021].
22. <https://docs.silabs.com/gecko-platform/latest/hardware-driver/api/group-imu> [cit. 29.12.2021].
23. https://github.com/SiliconLabs/peripheral_examples/blob/28c0122c1198621bda6f06ff9c802ab03ea22dd3/series1/adsc/adc_scan_interrupt/src/main_s1.c [cit. 29.12.2021].

7 Seznam příloh

Příloha A: Návod k použití

Příloha B: Obsah elektronické přílohy

8 Příloha A: Návod k použití

Blender

Obsah složky *Python_modules* je třeba zkopírovat do umístění:

c:\Program Files\Blender Foundation\Blender 2.93\2.93\scripts\modules

Pro spuštění programu slouží soubor *..\Blender_vizualizace.blend*. V horní liště otevřít kartu „Scripting“, v liště vlevo nahoře kliknout na záložku „Window“ a vybrat „Toggle system console“. V této konzoli se budou vypisovat užitečné informace pro kalibraci. Na řádku 10 je nutné specifikovat COM port, do kterého je deska zapojena. Teď už jen stačí v horní liště kliknout na znak trojúhelníku reprezentující příkaz na spuštění skriptu a postupovat podle textu v konzoli.

Prostředí Visual Studio

Projekt má název „PythonBLE“. Je nutné nainstalovat všechny potřebné knihovny. V levé kartě s názvem „Solution explorer“ po kliknutí pravým tlačítkem myši se zvolí možnost „View All Python Environments“. Níže zvolíme možnost „Packages (PyPI)“ a do pole „Search PyPI and installed packages“ napíšeme název knihovny, kterou chceme stáhnout a stiskneme klávesu *Enter*.

Potřebné knihovny:

- *asyncio*
- *bleak*

Jakmile se všechny knihovny nainstalují, tak kliknutím na tlačítko *Start* v horní liště se program spustí.

9 Příloha B: Obsah elektronické přílohy

- Instalační balíček programu Visual Studio 2019
- Instalační balíček programu Blender
- Potřebné knihovny pro spuštění kódu v aplikaci Blender ve složce *Python_modules*
- Zdrojové soubory pro naprogramování desky ve složce *bt_and_serial_comm*
- Zdrojové soubory pro sběr dat přes BLE v aplikaci Visual Studio 2019