

**Česká zemědělská univerzita v Praze**

**Technická fakulta**

**Katedra elektrotechniky a automatizace**



**Diplomová práce**

**Návrh a vývoj zařízení pro monitorování pohybu  
partikulární látky**

**Bc. David Voska**

**© 2023 ČZU v Praze**

# ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. David Voska

Informační a řídicí technika v agropotravinářském komplexu

Název práce

**Návrh a vývoj zařízení pro monitorování pohybu partikulární látky**

Název anglicky

**Design and development of equipment for monitoring the movement of particulate matter**

---

## Cíle práce

Cílem diplomové práce je provedení analýzy možností měření pohybu sypké látky. Výstupem práce bude návrh a realizace zařízení pro monitorování pohybu partikulární látky obsahující gyroskop a akcelerometr s bezdrátovým přenosem naměřených dat.

## Metodika

Prostudování hardwarových a softwarových možností řešení. Navržení několika variant provedení úlohy. Výběr nejvhodnější varianty s kritickým hodnocením návrhu. Specifikace funkcí modelu podle cílů práce.

## Doporučený rozsah práce

60stran, bez příloh

## Klíčová slova

akcelerometr, gyroskop, partikulární látka, detekce

---

## Doporučené zdroje informací

BELL, C A. *Beginning sensor networks with Arduino and Raspberry Pi*. [New York, New York]: Apress, 2013. ISBN 1430258241.

GARDNER, J W. – VARADAN, V K. – AWADELKARIM, O O. *Microsensors, MEMS, and smart devices*. Chichester: Wiley, 2001. ISBN 0-471-86109-.

MORRISS, S B. *Automated manufacturing systems : actuators, controls, sensors, and robotics*. New York: Glencoe, 1995. ISBN 0028023315.

ROSE, C W. *An introduction to the environmental physics of soil, water, and watersheds*. New York: Cambridge University Press, 2004. ISBN 0-521-53679-0.

---

## Předběžný termín obhajoby

2022/2023 LS – TF

## Vedoucí práce

doc. Ing. Miloslav Linda, Ph.D.

## Garantující pracoviště

Katedra elektrotechniky a automatizace

Elektronicky schváleno dne 24. 2. 2022

**doc. Ing. Miloslav Linda, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 25. 3. 2022

**doc. Ing. Jiří Mašek, Ph.D.**

Děkan

V Praze dne 15. 10. 2022



### **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci " Návrh a vývoj zařízení pro monitorování pohybu partikulární látky" jsem vypracoval (a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne datum odevzdání

---

## **Poděkování**

Rád bych touto cestou poděkoval vedoucímu práce Ing. Miloslavu Lindovi, Ph.D. za cenné rady a především poskytnutí prostředků k dokončení celého projektu a Ing. Barboře Černilové za vytisknutí pouzdra na 3D tiskárně.

# **Návrh a vývoj zařízení pro monitorování pohybu partikulární látky**

## **Abstrakt**

Cílem práce bylo sestavit zařízení na měření pohybu partikulární látky, které nám bude moci poskytnout odpověď na otázku s jakým zrychlením a úhlovou rychlostí se daná partikulární látka pohybuje. Požadavek na zařízení je, aby bylo co nejmenší a dokázalo bezdrátově komunikovat s výpočetním prostředkem a předávat mu data. Teoretická část popisuje princip akcelerometru a gyroskopu jakožto hlavních součástí zařízení a také se zabývá možnostmi bezdrátové komunikace. Praktická část zahrnuje výběr HW komponent, popis několika variant prototypu spolu s elektrickým zapojením a deskou plošných spojů. Dále zde najdeme postup výroby pouzdra pro zařízení spolu s nákresy pro 3D tisk.



Klíčová slova: akcelerometr, gyroskop, partikulární látka, detekce

# **Design and development of equipment for monitoring the movement of particulate matter**

## **Abstract**

Main goal of this diplomathesis is to develop a device for measuring of movement of particular matter that would be able to give us an answer to what acceleration and angular velocity the matter is moving with. Prerequisite for the device is to be small as much as possible and be able to communicate wirelessly to pass data. Theoretical part is describing accelerometer and gyroscope as main parts of the device as well as means of wireless communication. Practical part is displaying the way how HW components have been chosen, description of device's prototypes with electric circuits and printed circuit boards. Case for the device is also included alongside with 3D design.

**Keywords:** accelerometer, gyroscope, particular matter, detection

# Obsah

<b>1 Úvod.....</b>	<b>15</b>
<b>2 Cíl práce a metodika .....</b>	<b>16</b>
2.1 Cíl práce .....	16
2.2 Metodika.....	16
<b>3 Teoretická východiska .....</b>	<b>17</b>
3.1 Metody detekce pohybu látky .....	17
3.1.1 MEMS.....	17
3.1.2 Akcelerometr.....	17
3.1.3 Piezorezistivní akcelerometr.....	18
3.1.4 Kapacitní akcelerometr .....	19
3.1.5 Gyroskop.....	21
3.1.6 Vibrační gyroskop.....	21
3.1.7 Komunikace se zařízeními MEMS .....	22
3.2 Metody bezdrátové komunikace .....	22
3.2.1 Bluetooth.....	22
3.2.1.1 Hardwarová specifikace BLE.....	23
3.2.1.2 Topologie sítě .....	23
3.2.1.3 Profily .....	25
3.2.1.4 Komunikace se zařízením.....	25
3.3 Datové úložiště.....	25
3.3.1 FLASH EEPROM.....	26
<b>4 Vlastní zařízení pro detekci pohybu partikulární látky .....</b>	<b>27</b>
4.1 Komponenty .....	27
4.1.1 Výběr bezdrátového modulu.....	27
4.1.1.1 HM-11 BLE CC2541 CC2540 .....	28
4.1.2 Výběr mikroprocesoru .....	28
4.1.2.1 Atmega328.....	29
4.1.2.2 ESP8266-12F.....	31
4.1.3 Výběr FLASH paměti .....	32
4.1.3.1 SPI W25Q64.....	33
4.1.4 Výběr baterie a pouzdra .....	34
4.1.4.1 Baterie GP CR2032 .....	34
4.1.4.2 Držák CR2032 BC-2002 .....	34
4.1.4.3 Držák CR2032 pro Arduino LilyPad.....	35

4.1.4.4	Baterie CR2450 Li-ion .....	35
4.1.4.5	Držák CR2450 plastový.....	36
4.1.5	Výběr akcelerometru a gyroskopu .....	36
4.1.5.1	GY-521 MPU6050 .....	36
4.2	Sestavení prototypů .....	37
4.2.1	Voltampérová charakteristika CR2032.....	38
4.2.2	Voltampérová charakteristika CR2450.....	39
4.2.3	Porovnání voltampérových charakteristik CR2032 VS CR2450.....	40
4.2.3.1	Závěr.....	40
4.2.4	Výroba vývojového kitu ESP8266-12F.....	41
4.2.4.1	Zapojení.....	41
4.2.5	ESP8266-12F + baterie CR2032.....	42
4.2.5.1	Zapojení.....	42
4.2.5.2	Oživení.....	42
4.2.6	ESP8266-12F + baterie CR2450.....	43
4.2.6.1	Zapojení.....	43
4.2.6.2	Oživení.....	43
4.2.7	ESP8266-12F+ CR2450+MPU6050.....	43
4.2.7.1	Zapojení.....	43
4.2.7.2	Oživení.....	44
4.2.8	Atmega328 + CR2032 .....	44
4.2.8.1	HW Programátor.....	44
4.2.8.2	Zapojení.....	44
4.2.8.3	Nastavení .....	45
4.2.8.4	Oživení.....	45
4.2.9	Výroba vývojového kitu Bluetooth HM-11 BLE pro Atmega328 .....	46
4.2.9.1	Zapojení.....	46
4.2.9.2	Oživení.....	46
4.2.9.3	Nastavení .....	48
4.2.10	Atmega328+CR2032+Bluetooth+SPI FLASH+MPU6050.....	48
4.2.10.1	Zapojení.....	49
4.2.10.2	Oživení.....	49
4.3	Porovnání prototypů.....	50
4.4	Vývoj software pro vybraný prototyp .....	51
4.4.1	Požadavky.....	51
4.4.2	Úvod.....	52

4.4.3	Vývojový diagram .....	53
4.4.4	Konstantně nastavené parametry senzoru.....	56
4.4.5	Závěr z programového vývoje .....	56
4.4.5.1	Samplovací frekvence.....	57
4.4.6	Otestování programu na vybraném prototypu .....	57
4.4.6.1	Testování .....	57
4.4.6.2	Závěr.....	58
4.5	Sestavení vlastního senzoru .....	58
4.5.1	Schéma.....	58
4.5.2	Návrh PCB .....	60
4.5.3	Seznam použitých komponent .....	60
4.5.4	Výroba PCB .....	60
4.5.5	Osazení PCB .....	61
4.5.5.1	Zhodnocení .....	62
4.6	Oživení senzoru.....	63
4.6.1	Zapojení .....	63
4.7	Pouzdro.....	63
4.7.1	Půdorys .....	64
4.7.2	Vytisknutý 3D model.....	64
4.7.3	Zhodnocení vytisknutého modelu.....	65
4.8	Měření .....	65
4.8.1	Dosah bezdrátového signálu .....	65
4.8.2	Odběr elektrické energie .....	65
4.8.3	Data ze senzoru .....	65
4.8.3.1	Oscilace .....	66
4.8.3.2	Zdvih.....	69
4.8.3.3	Písmeno B.....	70
4.8.3.4	Pohyb po stole .....	71
4.8.3.5	Písmeno X.....	73
4.8.3.6	Závěr z měření .....	75
<b>5</b>	<b>Základní parametry senzoru.....</b>	<b>75</b>
<b>6</b>	<b>Výsledky a diskuse .....</b>	<b>75</b>
<b>7</b>	<b>Seznam použitých zdrojů .....</b>	<b>76</b>
<b>8</b>	<b>Seznam obrázků, tabulek, grafů a zkratk.....</b>	<b>79</b>
8.1	Seznam obrázků .....	79
8.2	Seznam tabulek .....	81

8.3	Seznam použitých zkratk.....	82
<b>9</b>	<b>Přílohy .....</b>	<b>83</b>
9.1	Ukázka sběru dat pomocí mobilního telefonu.....	83
9.2	Popis tříd .....	83
9.2.1	WBT s interfacem IBT.....	83
9.2.2	WFLASH s interfacem IFLASH .....	84
9.2.3	WMPU s interfacem IMPU .....	85
9.2.4	WEEPROM s interfacem IEEPROM .....	86
9.2.5	WMAINFCE s interface, IMAINFCE.....	87
9.2.6	SampleConfig .....	88
9.2.7	Sample.....	88
9.2.8	Constants.....	88

# 1 Úvod

V zemědělství si již neumíme představit život bez počítačů a elektronických zařízení, které nám usnadňují každodenní těžkou práci v terénu. Pokud chceme naše postupy zdokonalit, musíme při tom provádět měření, které nám poskytnou potřebné údaje k tomu, abychom dosáhli vytyčeného cíle. Například sledování chování určité látky při násypu, nebo pohyb půdy v určité hloubce po přejezdu těžké techniky, může být předmětem zájmu našeho zkoumání. Abychom však mohli měřit příslušné chování, potřebujeme přístroj dostatečně malý na to, aby jeho velikost neovlivňovala prováděná měření. S příchodem technologie CMOS a následné integrace čidel do stále menších a menších rozměrů křemíkových čipů, se nám naskýtá možnost vytvořit zařízení, která jsme schopni nejenom vyrobit opravdu malá, ale zároveň díky nízké spotřebě je i napájet například knoflíkovými bateriemi po relativně dlouhou dobu.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Cílem práce bylo navrhnout a sestavit zařízení na měření pohybu partikulární látky. Látkou může být jakýkoliv materiál, který je sypký a jehož pohyb chceme měřit. Měření by nám mělo dát odpověď na otázku s jakým zrychlením se zařízení a potažmo látka pohybuje, případně jakým směrem. Požadavek na zařízení je, aby bylo co nejmenší, jelikož chceme minimalizovat tření mezi látkou a zařízením tak, aby zařízení bylo unášeno a kopírovalo pohyb látky do nejmenších detailů. Zařízení je potřeba umístit do pouzdra kulového tvaru, které bude možné rozšroubovat v půlce a bude tak tvořit dvě polokoule. Dále musíme se zařízením určitým způsobem komunikovat a zabezpečit tak přenos dat ze zařízení do výpočetního prostředku k dalšímu zpracování. Tyto požadavky kladou velké nároky na použité komponenty, a proto musíme použít nejmenší a nejmodernější integrované součástky spolu s adekvátním zdrojem elektrické energie tak, aby bylo dosaženo požadovaného stupně miniaturizace.

### **2.2 Metodika**

Pro zahájení prací na projektu, bylo nutné nejprve načíst příslušnou literaturu a poté vybrat kandidáty na komponenty, z kterých bude zařízení sestaveno dle předem daných kritérií. V další fázi následoval návrh elektrického zapojení jednotlivých komponent a teoretické zhodnocení funkčnosti. Následovalo sestavení jednotlivých modulů komponent ve formě nepájitelného pole, které bylo propojeno do funkčního celku. Byl vytvořen program v jazyce C++ a nahrán do zařízení. Celek byl otestován pod elektrickým napětím a po potvrzení funkčnosti byly vytvořeny návrhy plošných spojů, které byly odeslány do výroby. Plošné spoje byly osazeny vybranými komponenty a sestaveny do jednoho celku. Následně proběhlo otestování funkčnosti prototypu. Na závěr byl celý projekt zhodnocen.



## 3 Teoretická východiska

Je potřeba nejdříve zjistit, jaké jsou možné metody detekce pohybu partikulární látky a způsoby bezdrátové komunikace zařízení s výpočetním prostředkem.

### 3.1 Metody detekce pohybu látky

Pohyb můžeme měřit vzdáleně z pevného stanoviště. Například pomocí optických senzorů či ultrazvuku. V našem případě chceme měřit pohyb látky tím, že vložíme do látky objekt, který se s ní pohybuje. Máme proto dvě možnosti čím měření provádět. Jedná se o akcelerometr nebo gyroskop typu MEMS. V praxi pracují akcelerometr a gyroskop společně a tudíž musíme použít oba najednou.

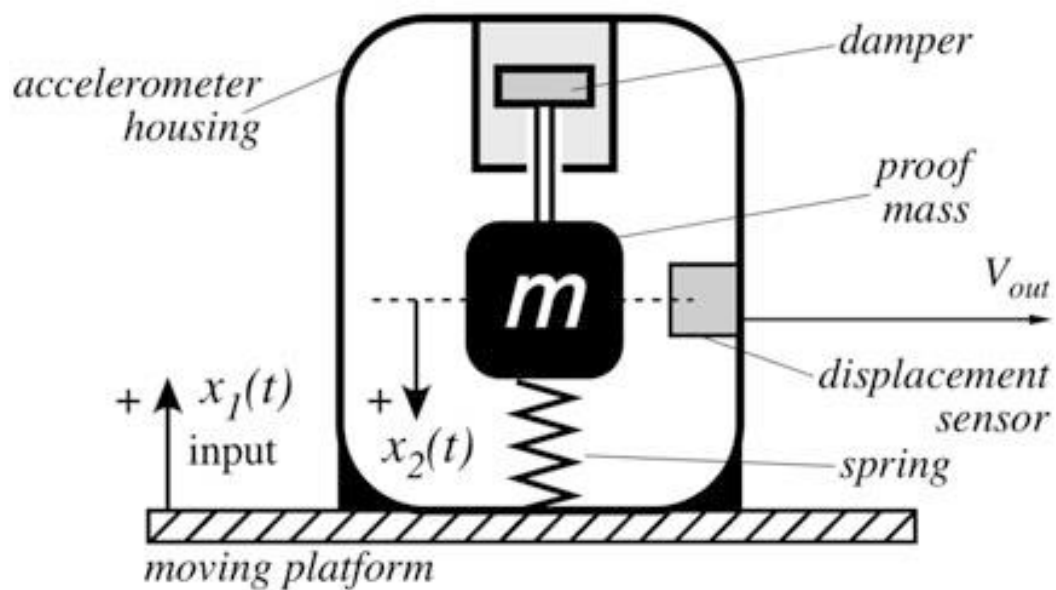
#### 3.1.1 MEMS

MEMS(microelectromechanical system) jsou obecně řečeno elektrická zařízení, která se skládají z mikro mechanických senzorů, akčních členů a mikro elektrických obvodů. Principiálně jsou sensory postaveny na piezoresistivitě nebo kapacitě. [1].

#### 3.1.2 Akcelerometr

Akcelerometr je zařízení, které měří zrychlení v požadovaných osách. Využívá se gravitačního zrychlení země  $9,80665 [m * s^{-2}]$ , pro určení náklonu v příslušných osách. Osvědčeným konceptem v této oblasti jsou zařízení typu MEMS.

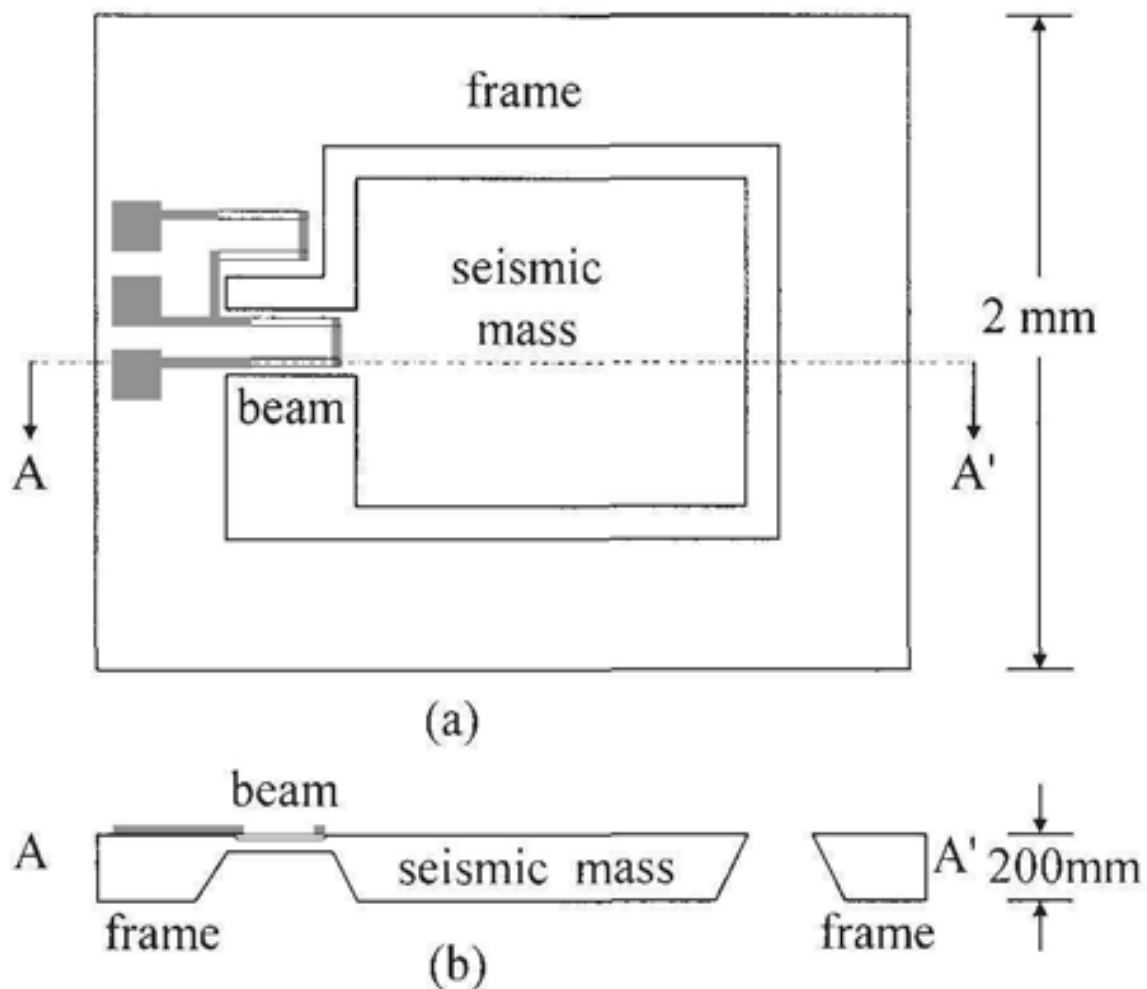
Z druhého Newtonova zákona  $a = \frac{F}{m} [m * s^{-2}]$  vyplývá, že pro měření akcelerace, budeme potřebovat hmotu o známé hmotnosti a senzor měřící sílu, kterou hmota působí na senzor. [2] Akcelerometr se skládá z rámu, seismické hmoty, ramene, pružiny a tlumiče. Seismická hmota se vychýlí ze své pozice vždy, když začne na akcelerometr působit vnější síla. Toto vychýlení je měřeno odporovou změnou (piezorezistivní akcelerometr), nebo změnou v kapacitě (kapacitní akcelerometr) [1]



Obrázek 1: Koncept akcelerometru [2].

### 3.1.3 Piezorezistivní akcelerometr

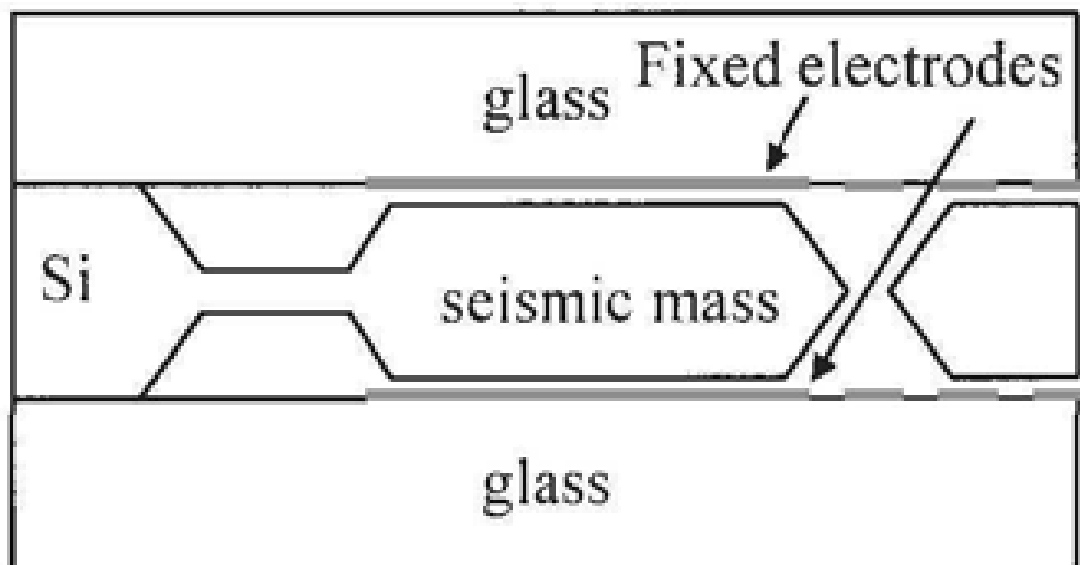
Skládá se ze seismické hmoty, ramene a dvou rezistorů. Jeden je na rameni a druhý na rámu. Pokud působí vnější síla na seismickou hmotu, tak hmota působí na rameno, které se deformuje a nastane změna v odporu rezistoru v rameni, která je proporcionální ke zrychlení.



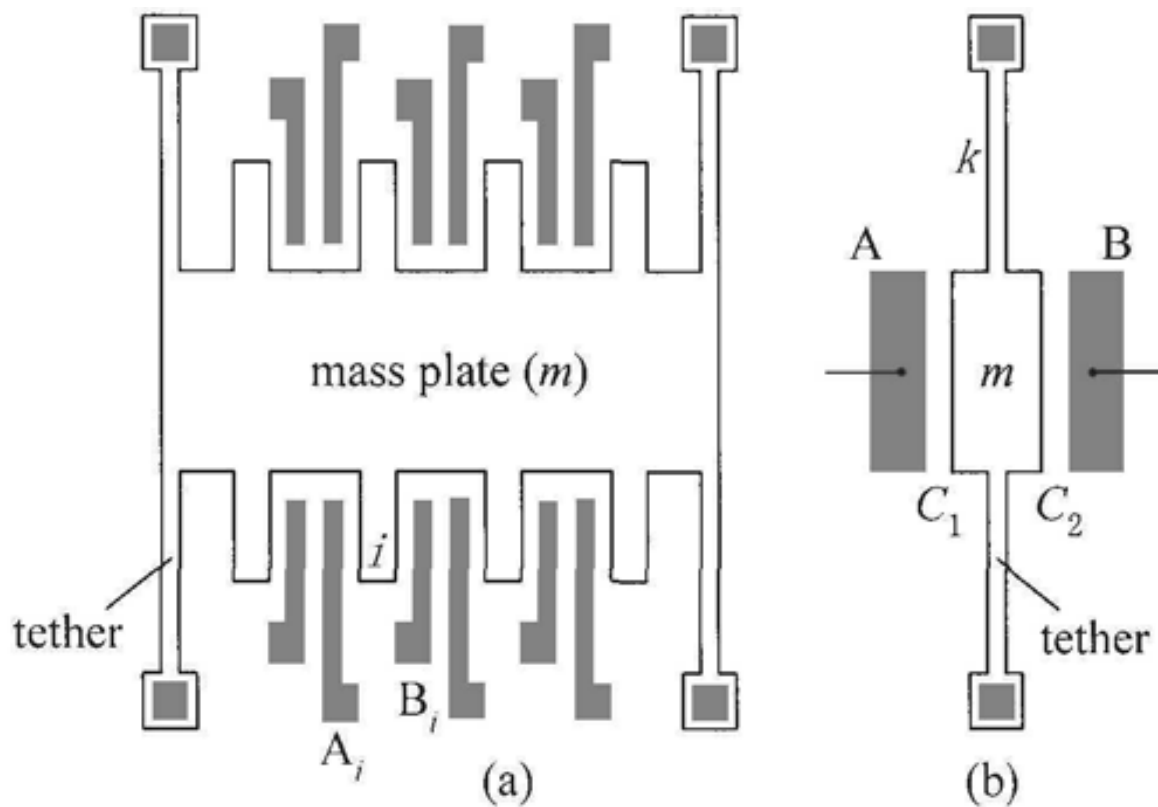
Obrázek 2: Piezorezistivní akcelerometr [1].

### 3.1.4 Kapacitní akcelerometr

Se skládá ze seismické hmoty, která slouží jako pohybující se elektroda. Je ukotvena na pružných listech do rámu a slouží pro vrácení hmoty do výchozí polohy po odeznění vnější síly. Tato hmota je umístěna mezi dvěma skleněnými pláty, na kterých je druhá elektroda. Tato sendvičová struktura slouží jako kapacitor. Princip spočívá v tom, že jakmile se změní poloha seismické hmoty, změní se i kapacita mezi elektrodami, která se dá převést na zrychlení.



Obrázek 3: Kapacitní akcelerometr [1].



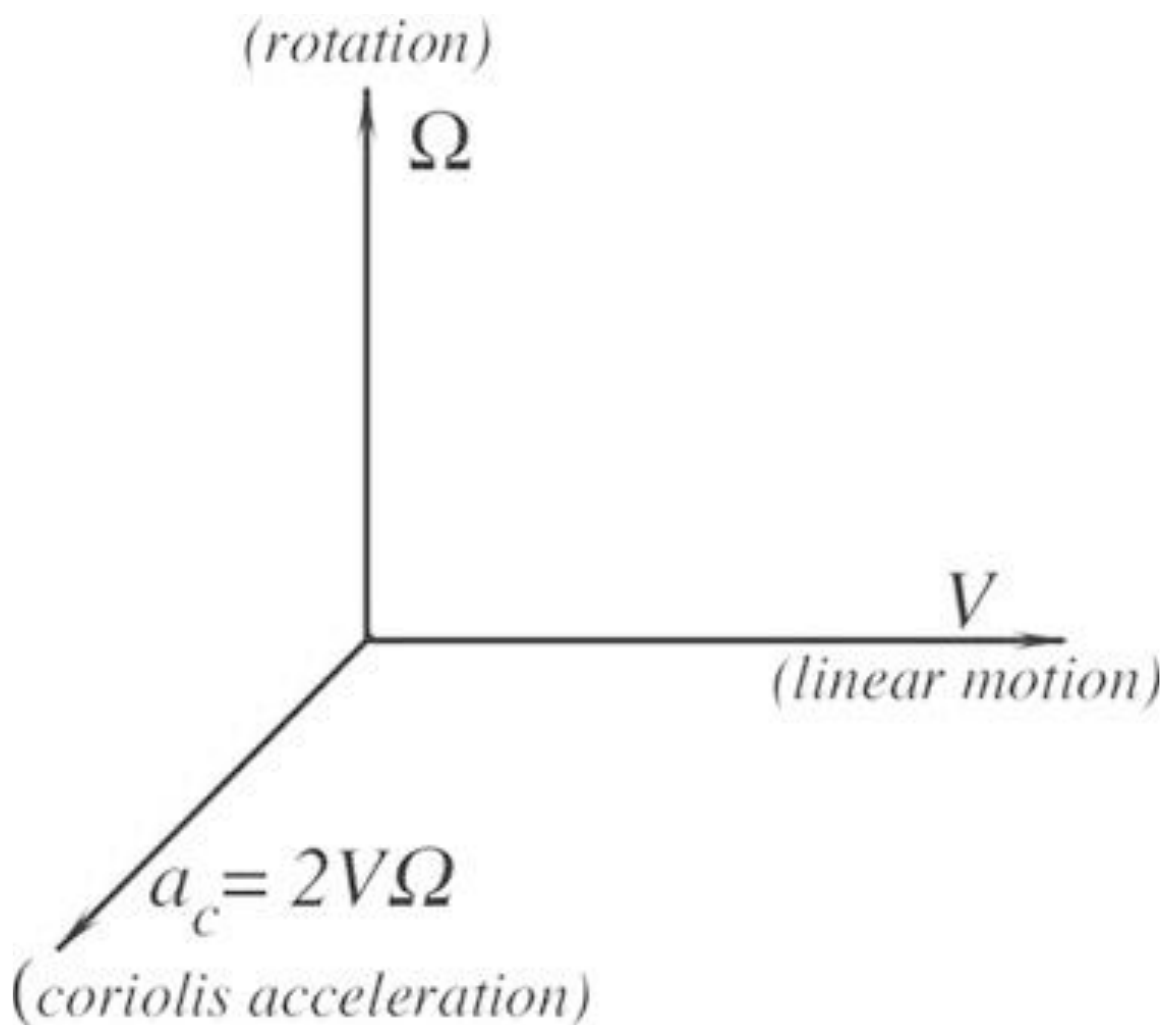
Obrázek 4: Kapacitní akcelerometr, pružné lamely [1].

### 3.1.5 Gyroskop

Gyroskop je zařízení, měřící úhlovou rychlost [ $rad * s^{-1}$ ] okolo určité osy. Úhlová rychlost je fyzikální veličina, která popisuje otáčivý pohyb tělesa. Úhlová rychlost omega je dána vztahem  $\Omega = \frac{d\varphi}{dt}$  [ $rad * s^{-1}$ ]. [2]. Nejvíce používaným gyroskopem v mikroelektronice je Vibrační gyroskop.

### 3.1.6 Vibrační gyroskop

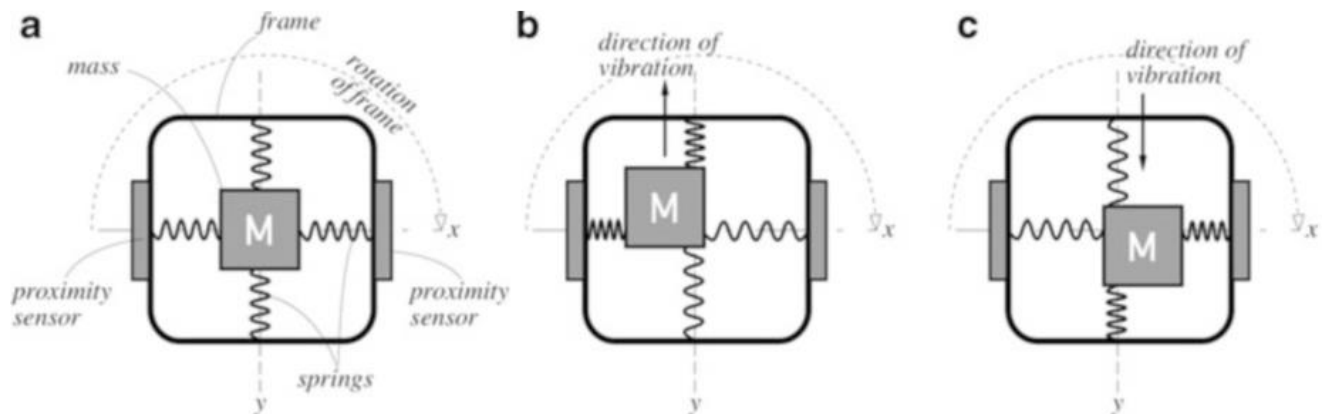
Využívá efektu Coriolisovy síly  $F = 2m\Omega v$  [ $N$ ], kdy na pohybující se těleso v neinerciální soustavě působí právě tato síla, která vyvolá pohyb tělesa ve směru kolmém na osu otáčení neinerciální soustavy a směru pohybu tělesa. [3].



Obrázek 5: Vektor Coriolovy akceleraace [2].

Princip je takový, že hmota o určité hmotnosti osciluje podél osy y s určitou frekvencí, to znamená, že hmota se pohybuje nahoru a dolů velkou rychlostí. Jakmile nakloníme gyroskop,

dojde k aktivaci Coriolisovy síly, která pohne hmotou doprava, nebo doleva na ose x. Tento pohyb je zaznamenán senzorem, který převede pohyb na úhlovou rychlost [ $rad * s^{-1}$ ]. [2]



Obrázek 6: Konceptuální diagram Gyroskopu [2].

### 3.1.7 Komunikace se zařízeními MEMS

Nejběžnější komunikace se děje prostřednictvím sběrnice I2C. Dále se používá protokol UART aj.

## 3.2 Metody bezdrátové komunikace

Možností jak komunikovat bezdrátově máme mnoho. Mezi neznámější patří Wi-Fi a Bluetooth. Pro naše účely se hodí pouze komunikace přes Bluetooth, jelikož je energeticky nejméně náročná.

### 3.2.1 Bluetooth

Bluetooth je komunikační standard znám hlavně díky mobilním telefonům. Rozeznáváme dvě základní verze. Bluetooth classic a Bluetooth Low Energy. Tyto verze nejsou mezi sebou napřímo kompatibilní. Existují však zařízení, které kompatibilitu zajišťují. [4]

Device	BR/EDR (classic Bluetooth) support	BLE (Bluetooth Low Energy) support
Pre-4.0 Bluetooth	Yes	No
4.x Single-Mode (Bluetooth Smart)	No	Yes
4.x Dual-Mode (Bluetooth Smart Ready)	Yes	Yes

Obrázek 7: Verze Bluetooth [4].

Momentálně je k dispozici verze 5.0, nicméně pro naše účely se hodí verze 4.0 BLE, protože má velkou podporu jak ze strany výrobců, tak v komunitě. Velkou výhodou BLE je, že se zařízení nemusí párovat jako verze classic.

### 3.2.1.1 Hardwarová specifikace BLE

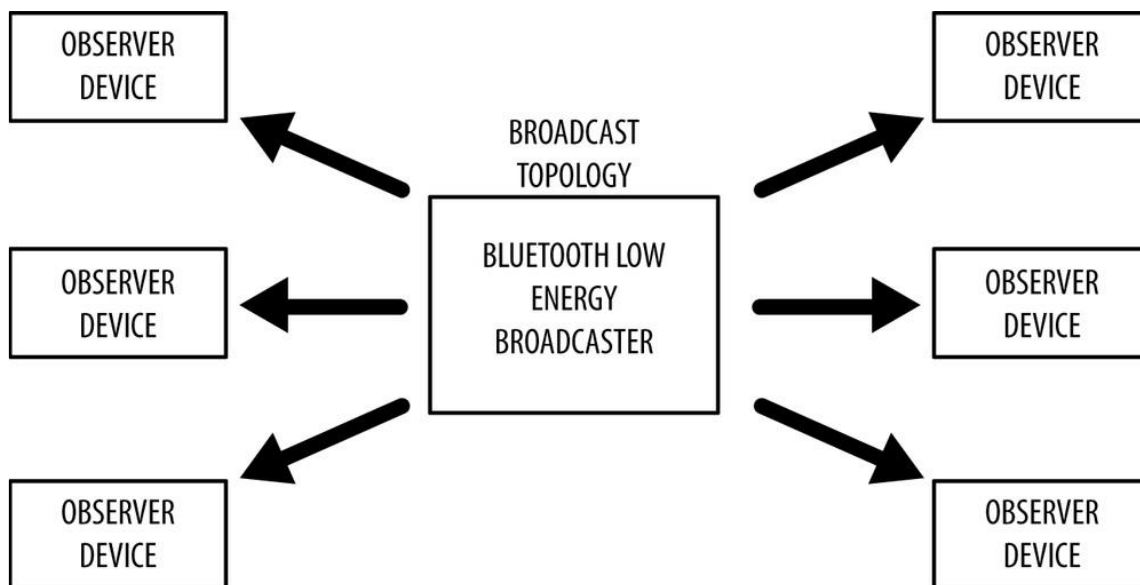
Pracuje na frekvenci 2.4 GHz, která je určena pro industriální, vědecké a medicínské účely. BLE pracuje se 40 kanály, mezi kterými se přeskakuje, tzn. Frequency hopping, který zabraňuje nadměrnému rušení ze strany jiných vysílačů. Jako modulační protokol se používá Gaussian Frequency shift Keying modulace, která má modulační rychlost 1 Mbit/s. Tato rychlost je zároveň teoreticky nejvyšší možná pro přenos dat, ale zpravidla se dosahuje rychlosti okolo 40 Kbit/s. Vysílací výkon je zpravidla někde mezi -30 a 0 dBm, což by mělo umožnit dosáhnout komunikační vzdálenosti až 30m, ale zpravidla s dosahuje okolo 5m. [4]

### 3.2.1.2 Topologie sítě

Každé zařízení má pevnou adresu MAC, která je 48 bitová a je unikátní. Vedle této adresy existuje ještě programovatelná adresa, která brání skenovacím zařízením ve sledování. BLE poskytuje dvě možnosti komunikace a to broadcasting, nebo direct connection. Broadcasting znamená, že zařízení vysílá broadcast pakety (Broadcast topology), které mohou být přijmuty kýmkoliv, tento režim se označuje také jako iBeacon. Zařízen vystupuje ve dvou rolích.

- Observer –Scanuje opakovaně vysílací pásmo, zdali se tam nenachází broadcastovací pakety.
- Broadcaster-Vysílá broadcastovací pakety.

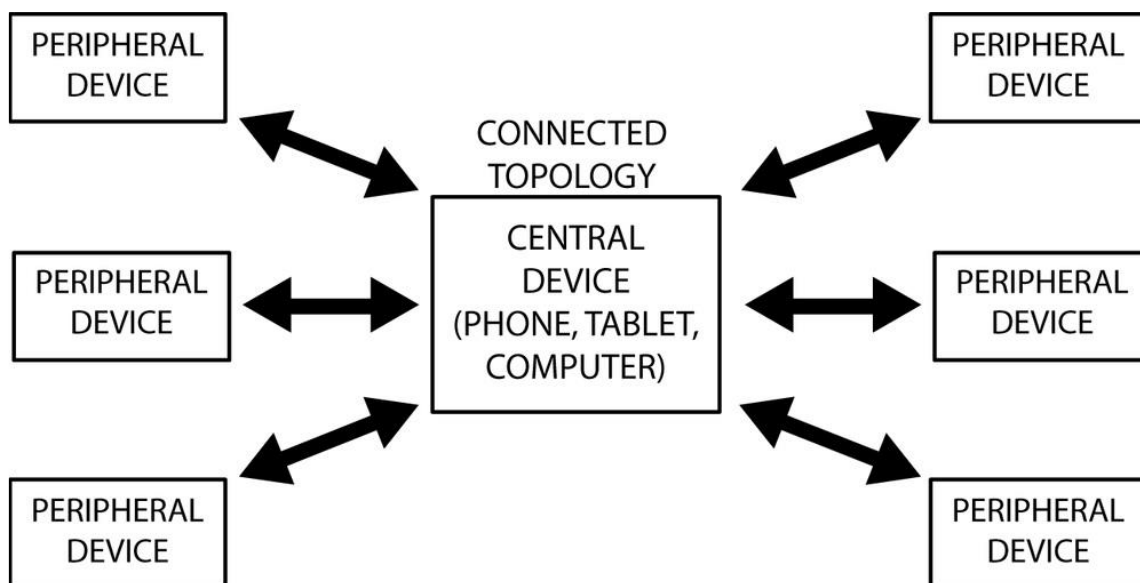
Existuje zde omezení maximální velikosti paketu na 31 bytů. A Nemůžeme využívat bezpečnostní vrstvu, která poskytuje párování a šifrování a nemůžeme rovněž používat Generic Attribute Profile (GATT), o kterých bude psáno později. [4]



Obrázek 8: Bluetooth Observer koncept [4].

Druhou možností je se přímo připojit k určitému zařízení prostřednictvím tzv. Direct connection a s ním komunikovat podobně jako protokol TCP/IP (Connected topology). Zařízení zde vystupují ve dvou rolích.

- Master-Skenuje vysílací frekvence a hledá advertising pakety. Pokud je najde, zahájí připojování.
- Slave-Zařízení vysílá advertising pakety a přijímá připojení od Master zařízení. Neexistuje zde omezení na velikost paketů a je zde možnost využívat GATT. Spotřeba zařízení v tomto režimu je menší než při broadcastování.



Obrázek 9: Connected topologie [4].



### 3.2.1.3 Profily

Pro nastavení parametrů zařízení a získávání dat, se používají profily. Existují dva základní profily.

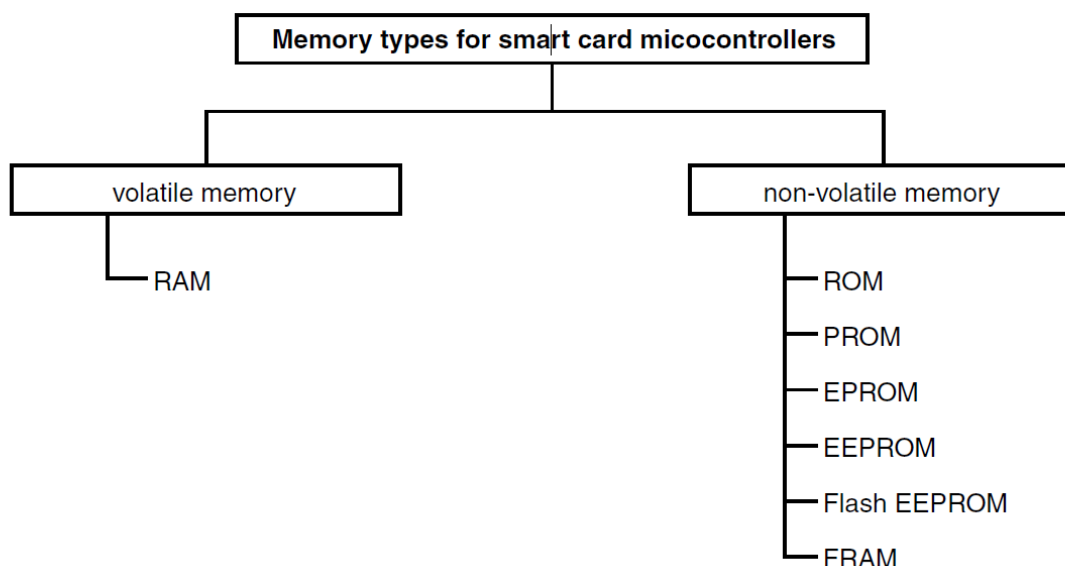
- GAP (Generic Access Profile) – Slouží k nastavení základních atributů zařízení jako například jméno, nebo v jaké topologii má pracovat.
- GATT (Generic Attribute Profile) – Zde najdeme data, která chceme prezentovat skrz zařízení Bluetooth. Například pokud je Bluetooth zařízení napojeno na sensor srdečního tepu, zde můžeme nalézt číselnou hodnotu ze senzoru.

### 3.2.1.4 Komunikace se zařízením

Jak již bylo zmíněno, komunikovat se dá pomocí profilů přístupných pomocí API knihoven např. z Androidu, nebo z Windows. Další možností komunikace je pomocí USB či UART prostřednictvím HCI (Host Controller Interface).

## 3.3 Datové úložiště

Budeme se zde zabývat pouze elektronickými datovými úložišti, která se nejvíce hodí pro tento projekt. Základní rozdělení je uvedeno níže ve schématu.



Obrázek 10: Rozdělení paměti [5].

### 3.3.1 FLASH EEPROM

Volatilní paměť potřebuje k uchování dat neustálý přísun elektrické energie, zatímco nonvolatilní si uchová svůj obsah i po jejím odpojení. Zaměříme se pouze na nonvolatilní paměti typu FLASH EEPROM, protože potřebujeme zajistit zapsaná data i po výpadku napájení. Nevýhoda této paměti je, že má omezený počet zapisovacích cyklů, zpravidla se uvádí 100 000 programovacích cyklů. Paměť je tvořena malými kondenzátory a na základě náboje se rozlišuje logická 1 nebo 0. Pokud je na kondenzátoru náboj, znamená to logickou 1. Z hlediska organizačního je paměť rozdělena do bloků, které se nazývají Page. Obvyklá velikost je od 1-256 bytů. Pokud chceme zapisovat data, musí být splněna podmínka, že celá Page obsahuje pouze logické 1, protože zápis umí pouze kondenzátory vybíjet. Pokud je chceme znovu nabít, musíme celou Page smazat. [5]

## 4 Vlastní zařízení pro detekci pohybu partikulární látky

Již víme, jaké jsou možnosti detekce pohybu partikulární látky, a také víme, jak komunikovat bezdrátově s výpočetním prostředkem. Na základě získaných znalostí jsme již schopni vybrat jednotlivé komponenty a sestavit tak zařízení dle cílů této práce.

### 4.1 Komponenty

Dle zadání by zařízení mělo být schopno fungovat na baterii typu CR2032 3V, alternativně pak CR2450 3V po dobu alespoň 2 hodin a přenášet data na vzdálenost alespoň 3 m, nebo tyto data zapisovat do vnitřní nonvolatilní paměti a poté přenést v jednom sledu. Rozměr zařízení by neměl překročit 35 mm v průměru. Jak již bylo řečeno výše, zařízení se skládá z mikroprocesoru, bezdrátového modulu, nonvolatilní paměti, akcelerometru a gyroskopu, baterie CR2032, nebo CR2450 3V. Tyto komponenty musí být propojeny mezi sebou vodivými cestami. Desky plošných spojů budou rozebrány podrobněji na konci kapitoly. Výběr komponent probíhal ve dvou fázích. Nejprve se vybraly komponenty, které se následně nakoupily a poté se na nepájivém poli vytvořily prototypy. Následně se porovnály jejich vlastnosti a vybral se ten nejvhodnější prototyp.

#### 4.1.1 Výběr bezdrátového modulu

Nejprve bylo třeba vybrat bezdrátové moduly, protože na základě tohoto modulu bylo možno sestavit porovnávací tabulku pro mikroprocesory. Byly vybrány dva nejpoužívanější bezdrátové komunikační standardy Bluetooth a Wi-Fi. Na trhu v době psaní této práce nebyl k dispozici samostatný Wi-Fi modul, proto se pozornost zaměřila pouze na Bluetooth.

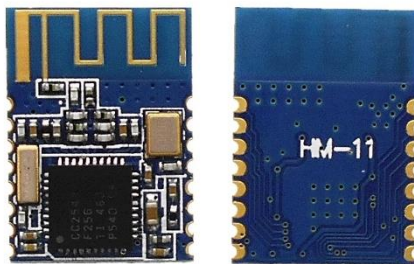
Byly definovány vstupní parametry, které musel splňovat každý modul.

1. Podpora BLE.
2. SMD provedení.
3. Napájení 3 V.
4. Komunikace I2C, nebo UART.
5. Dosah minimálně 5 m.
6. Probuzení z režimu spánku pomocí vektoru přerušení.
7. Rozměry maximálně 24 x 24 mm.

Byl proveden průzkum českého trhu a zjistilo se, že v nabídce je pouze jedna varianta skladem, která splňuje daná kritéria a tím je HM-11 BLE Modul Bluetooth 4.0 CC2541 CC2540.

#### 4.1.1.1 HM-11 BLE CC2541 CC2540

Tento modul má rozměry 18.5x13.5 mm, je standardu BLE 4.0 a dokáže komunikovat dle výrobce do vzdálenosti 100m. Obsahuje integrovanou anténu a je postaven na čipsetu CC2541. Spotřeba v pohotovostním režimu 0.5 mA a aktivní komunikace by měla odebírat maximálně 9 mA. Lze také regulovat vysílací výkon následovně -23dbm, -6dbm, 0dbm, 6dbm. Podporuje režim Master, Slave, nebo iBeacon. Nastavení se provádí pomocí AT příkazů prostřednictvím UART rozhraní. Jakmile se odpojíme od UARTU, zařízení začne pracovat ve zvoleném režimu.



Obrázek 11: HM-11 Bluetooth.

#### 4.1.2 Výběr mikroprocesoru

Páteří celého zařízení je řídicí mikroprocesor, který propojuje jednotlivé komponenty dohromady a určuje chování celého systému. Od mikroprocesoru bylo požadováno, aby měl co nejmenší spotřebu, rozměry a zároveň nebylo obtížné programování aplikací.

Na trhu je nepřehledné množství mikroprocesorů a platforem, bylo proto potřeba provést předvýběr, na základě stanovených obligatorních vstupních parametrů, které musí splňovat každá varianta. Těmito parametry jsou.

1. I2C sběrnice.
2. UART rozhraní pro komunikaci s většinou komponent na trhu.
3. Podpora programovacího jazyka C++, která je zprostředkována programovacím nástrojem Arduino.

4. Napájení 3 V.
5. FLASH paměť alespoň 20 KB.

Dle aktuální nabídky na českém trhu v době psaní této práce byly k dispozici pouze mikroprocesory Atmega328 a ESP8266.

#### 4.1.2.1 Atmega328

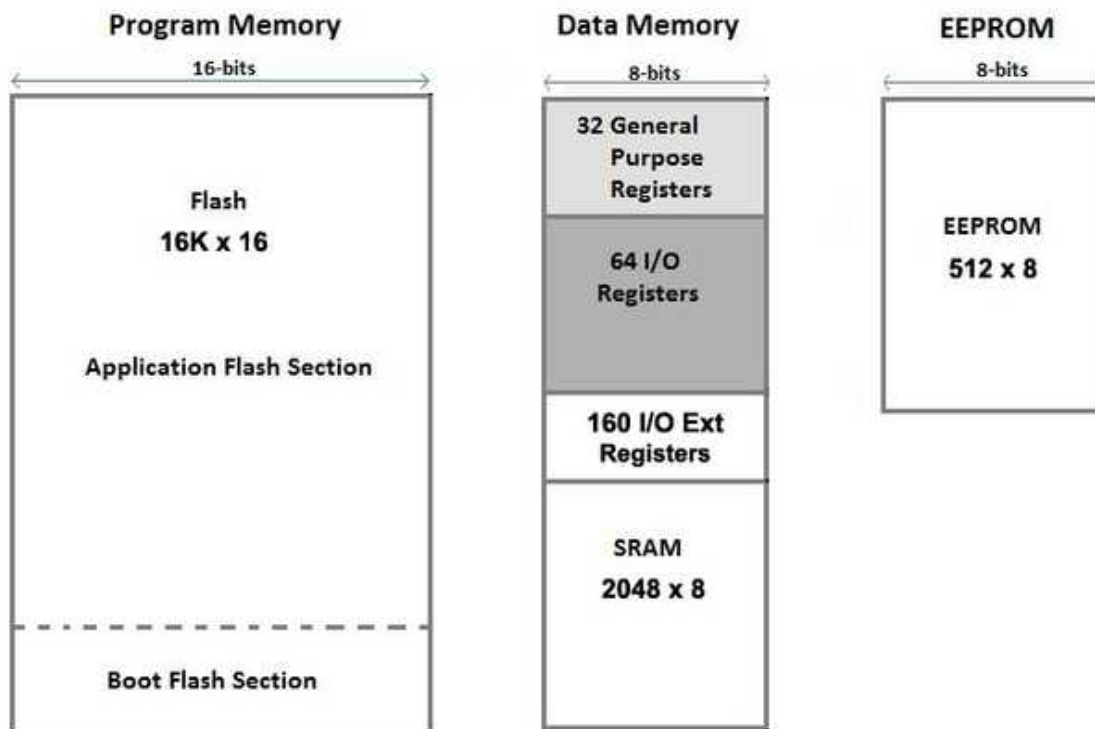
Atmega328 je 8 bitový CMOS mikroprocesorem se 16bitovými instrukcemi, založený na Harvardské architektuře, jenž se vyznačuje především oddělenou fyzickou pamětí pro data a instrukce.

Mezi základní parametry patří:

1. Maximální rychlost procesoru 20 Mhz.
2. 32 KB FLASH nonvolatilní paměti pro instrukce.
3. 1 KB EEPROM nonvolatilní paměti pro uživatelské proměnné.
4. 2 KB SRAM volatilní pro uživatelské proměnné.
5. 2x 8 bitový časovač.
6. 1x 16 bitový časovač.
7. 6x PWM kanál.
8. Osmý kanálový 10 bitový ADC převodník.
9. 2x SPI rozhraní.
10. UART rozhraní.
11. I2C sběrnice.
12. Analogový komparátor.
13. Interní oscilátor.
14. 2x pin přerušení pro arduino.
15. 23x programovatelné vstupně výstupní piny.
16. Rozsah napájecího napětí 1.8 V- 5.5 V.
17. Rozměry 7 x 7 mm.

Spotřeba je řízena pomocí šesti režimů spánku. Velkou výhodou tohoto mikroprocesoru je možnost nastavit si například přesnou rychlost procesoru apod. Přesná rychlost procesoru je důležitá, pokud chceme minimalizovat spotřebu. Aplikační instrukce můžeme nahrávat prostřednictvím Arduino bootloaderu, nebo napřímo pomocí HW programátoru. Zajímavá možnost je nahrávat aplikační instrukce pomocí Arduino bootloaderu, který dovoluje nahrávat C++ kód bez použití programátoru pomocí UART rozhraní. Procesor se nastavuje pomocí HW

programátoru, kde se nastavují tzv. fúze a také se takto nahrávají aplikační instrukce v jazyce C. Nevýhodou je poměrně složitá aplikace procesoru do elektrického obvodu, zejména co se týče pájení na PCB a nutnost použít HW programátor pro nahrání bootloaderu a nastavení fúzí. [6], [7]



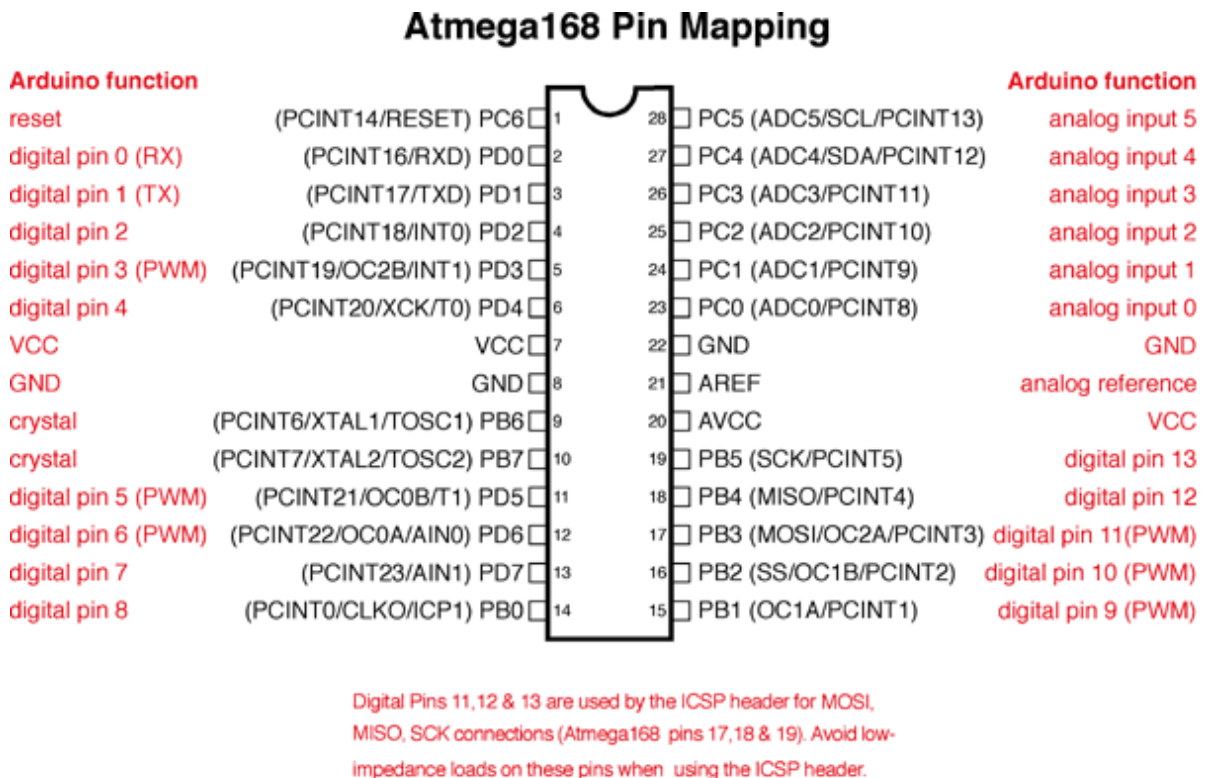
Obrázek 12: Rozložení adresního prostoru Atmega [8].



Obrázek 13: ATmega328.

#### 4.1.2.1.1 Schéma-Arduino Pin Mapping

Pokud chceme na procesoru Atmega168/328 provozovat aplikace programované ve vývojovém prostředí Arduino, musíme pamatovat na to, že rozložení pinů je jiné v porovnání s programováním pro nativní platformu v jazyce C.



Obrázek 14: Schéma-Arduino Pin Mapping [9].

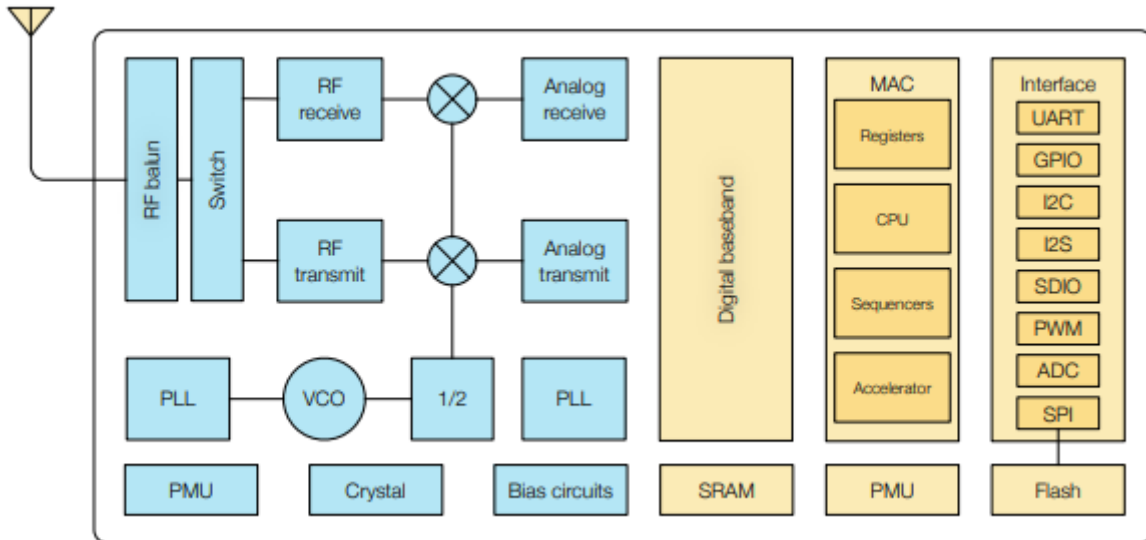
#### 4.1.2.2 ESP8266-12F

Tento modul, který se jmenuje ESP-12F obsahuje 32 bitový mikroprocesorem ESP8266 zpracovávající 16 bitové instrukce založený na Harvardské architektuře s Wi-Fi modulem a integrovanou anténou.

Základní parametry

1. Rychlost cpu do 160 Mhz.
2. 50 KB SRAM.
3. 10 bit ADC převodník.
4. Integrovaná SPI FLASH 32 Mbit.
5. 9 programovatelných vstupně výstupních pinů.
6. Napájecí napětí od 2.7-3.6 V.
7. Rozměry 24 x 16 mm.

Mezi základní přednosti modulu patří to, že obsahuje již integrovanou Wi-Fi a SPI FLASH. Jeho mikroprocesorem umí pracovat na frekvencích 80 Mhz nebo 160 Mhz. Podporuje programování přes vývojový software Arduino po nainstalování pluginu. Aplikační instrukce lze nahrávat pouze přes toto rozhraní. Nevýhodou je poměrně omezený rozsah nastavení parametrů procesoru. [10]



Obrázek 15: Schéma ESP8266 [10].



Obrázek 16: ESP8266.

### 4.1.3 Výběr FLASH paměti

Pro případ, že zařízení je mimo dosah, nebo chceme snížit jeho odběr, je lepší data po dobu měření ukládat a pak až poslat v jednom sledu prostředkem bezdrátové komunikace. Vstupní parametry byly stanoveny následovně.

1. Napájení 3 V.
2. Minimální velikost paměti alespoň 16 Mbit.
3. SMD provedení.

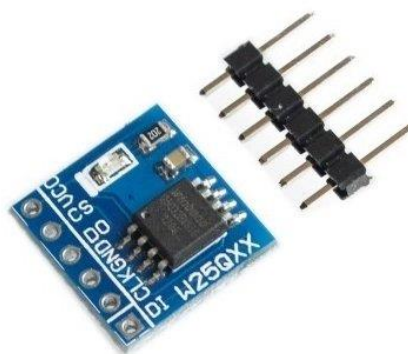


#### 4. Řízení pomocí sběrnice I2C, nebo SPI.

Na českém trhu je omezený výběr FLASH pamětí, a v době psaní této práce byla k dispozici paměť od společnosti Winbond velikosti 64 Mbit.

##### 4.1.3.1 SPI W25Q64

Tato paměť byla vybrána ve dvou provedeních. Prvním je paměť ve formě modulu, který obsahuje rozhraní pro připojení k nepájivému poli.



Obrázek 17: Paměť SPI W25Q64.

Druhé provedení je pouze SMD čip, který se použije ve finálním výrobku.

Parametry této paměti jsou uvedeny níže.

1. Napájení 2.7 V – 3.6 V.
2. Čtecí proud při 50 Mhz je 15 mA.
3. Mazací proud 25 mA.
4. Standby proud 0,050 mA.
5. Velikost paměti 64 Mbit.
6. Velikost Page je 256 bytů.

## 7. Komunikace pomocí SPI.

Mazání dat může být provedeno pouze po blocích, které jsou pevně dané výrobcem a to následovně. Nejmenší blok, který můžeme smazat má 4 KB, následuje blok 32 KB, 64 KB, nebo může být smazána celá paměť. Je také důležité poznamenat, že rychlost mazání a čtení je závislá na taktu procesoru, maximální frekvence je dle datasheetu je 104 Mhz. [11]

### 4.1.4 Výběr baterie a pouzdra

Projekt počítá s napájením pomocí baterie typu CR2032, nebo CR2450. Baterie CR2032 je menší a lehčí, na druhou stranu nedosahuje takové kapacity, jako CR2450, která je ale o poznání robustnější. CR2032 má v průměru 20 mm, výšku 3.2 mm a kapacitu 220 mAh. CR2045 má průměr 24.5 mm, výšku 5 mm a kapacitu kolem 500mAh. Co se uložení baterie týče, průzkum trhu odhalil, že existují dva druhy uložení, které připadají v úvahu pro tento projekt a těmi jsou uložení vyrobené z plastu anebo vyrobené z tenkého plechu. [12]

Byly vybrány následující komponenty.

#### 4.1.4.1 Baterie GP CR2032

Má kapacitou 220mAh a teoreticky by měla poskytnout max. 15mA odběrový proud.

#### 4.1.4.2 Držák CR2032 BC-2002

Výhodou tohoto držáku je, že je minimalistický. Nevýhodou je nutnost pájení na PCB. Má šířku 21 mm, délku 20 mm a výšku 4mm.



Obrázek 18: Držák pro CR2032 –plech [13].

#### 4.1.4.3 Držák CR2032 pro Arduino LilyPad

Má tu výhodu, že má vypínač a je již napájen na PCB. Nevýhodou je robustnost. Má rozměry 29 mm na 29 mm a výšku 5 mm.



Obrázek 19: Držák pro CR2032 -arduino Lily Pad [14].

#### 4.1.4.4 Baterie CR2450 Li-ion

Tato baterie byla vybrána, protože by měla být teoreticky schopna poskytovat mnohonásobně větší proud, než nenabíjecí verze a než její menší verze CR2032 a to kolem 120 mA. Velikost průměru je 24 mm a výška 5 mm. Rozdíl oproti nenabíjecí verzi je ten, že napětí není 3 V ale 3.6 V. [15]



Obrázek 20: Baterie CR2450 Li-ion [16].

#### 4.1.4.5 Držák CR2450 plastový

Průměr je 28 mm a výška 7mm.



Obrázek 21: Držák baterie CR2450-plastový [17].

#### 4.1.5 Výběr akcelerometru a gyroskopu

Vstupní parametry byly stanoveny takto

1. Napájení 3V.
2. Rozměry maximálně 24 x 24 mm.
3. Komunikace po sběrnici I2C.

##### 4.1.5.1 GY-521 MPU6050

V době psaní této práce byl k dispozici na trhu pouze výrobek s čipem MPU6050, který obsahuje jak tříosí akcelerometr tak tříosí gyroskop a navíc ještě měří teplotu.

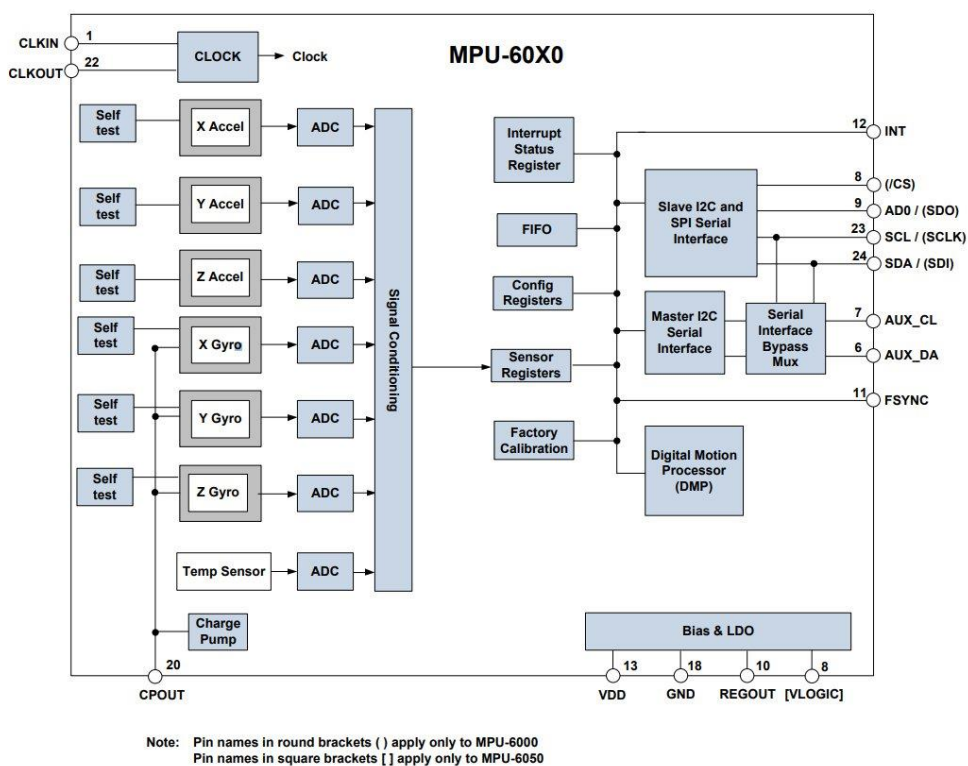
Parametry zařízení

1. 16 bitový AD převodník.
2. Rozsah gyroskopu: 250, 500, 1000, 2000 °/s.
3. Rozsah zrychlení:  $\pm 2 \pm 4 \pm 8 \pm 16$  g.
4. Napájení 3-5 V.
5. Rozměry šířka 15 mm, délka 21 mm a výška 1.2 mm.
6. 1x Vektor přerušení

MPU6050 obsahuje 16 bitový AD převodník, který převádí naměřený potenciál na digitální číslo. Výstup z akcelerometru, gyroskopu a teploměru je ukládán do jediného registru, který se jmenuje FIFO. MPU6050 obsahuje také procesor DMP, který zpracovává data ze všech senzorů a provádí nad nimi výpočet. Mezi tyto výpočty patří například rozeznávání pohybových gest, pohyb ve 3D prostoru, detekce zatřesení apod. Rovněž je zde možné nastavit low-pass filtr, na odstranění šumu. Zařízení je schopné detekovat volný pád, nebo zatřesení a aktivovat vektor přerušení, což je výhodné, pokud chceme šetřit energii. Výhodou je, že existuje velké množství knihoven pro Arduino.

#### 4.1.5.1.1 Provedené úpravy

GY-521 modul obsahuje LED diodu, jejíž odběr činí 1.5 mA. Tato dioda byla odstraněna, aby se snížil odběr zařízení.



Obrázek 22: Schéma MPU6050 [18].



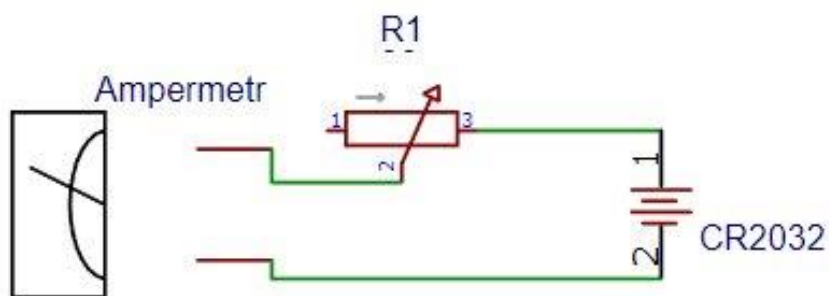
Obrázek 23: MPU6050 [19].

## 4.2 Sestavení prototypů

Nyní již máme hlavní díly, z kterých můžeme postavit jednotlivé prototypy. Bylo použito nepájivých polí, do kterých byly umístěny součástky a ty pospojovány odnímatelnými vodiči. Následně byly změřeny elektrické charakteristiky určitých komponent a prototypů, které pak byly porovnány mezi sebou.

### 4.2.1 Voltampérová charakteristika CR2032

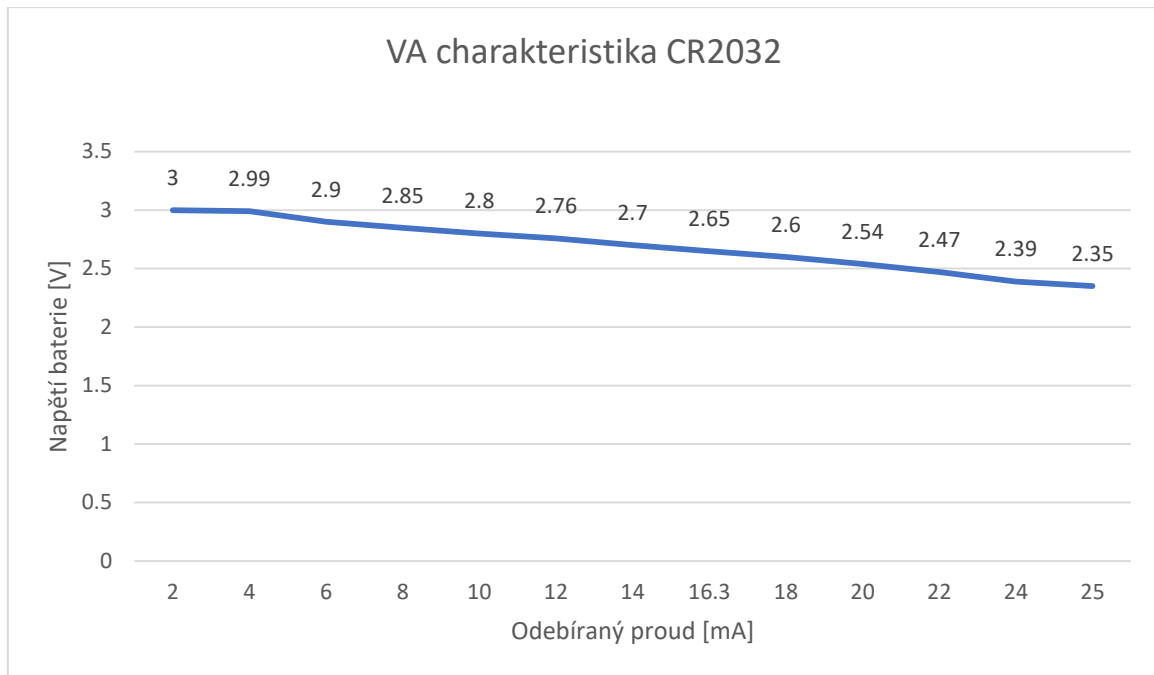
Panovaly obavy, že takto malá baterie nebude schopna dodávat velké množství proudu pro naše zařízení. Bylo proto provedeno měření, mající za úkol zjistit voltampérovou zatěžovací charakteristiku této baterie. Zapojení se skládá z držáku baterie LilyPad, ampérmetru a variabilního rezistoru R1 s rozsahem od 0 do 1.7 k $\Omega$ .



Obrázek 24: Elektrické schéma zapojení-VA CR2032.

Napětí baterie [V]	Odebíraný proud [mA]
3	2
3.03	3
2.99	4
2.95	5
2.9	6
2.87	7
2.85	8
2.83	9
2.8	10
2.78	11
2.76	12
2.74	13
2.7	14
2.67	15.2
2.65	16.3
2.63	17
2.6	18
2.57	19.2
2.54	20
2.50	21
2.47	22
2.42	23
2.39	24
2.35	25

Tabulka 1: VA charakteristika CR2032-Naměřené hodnoty.



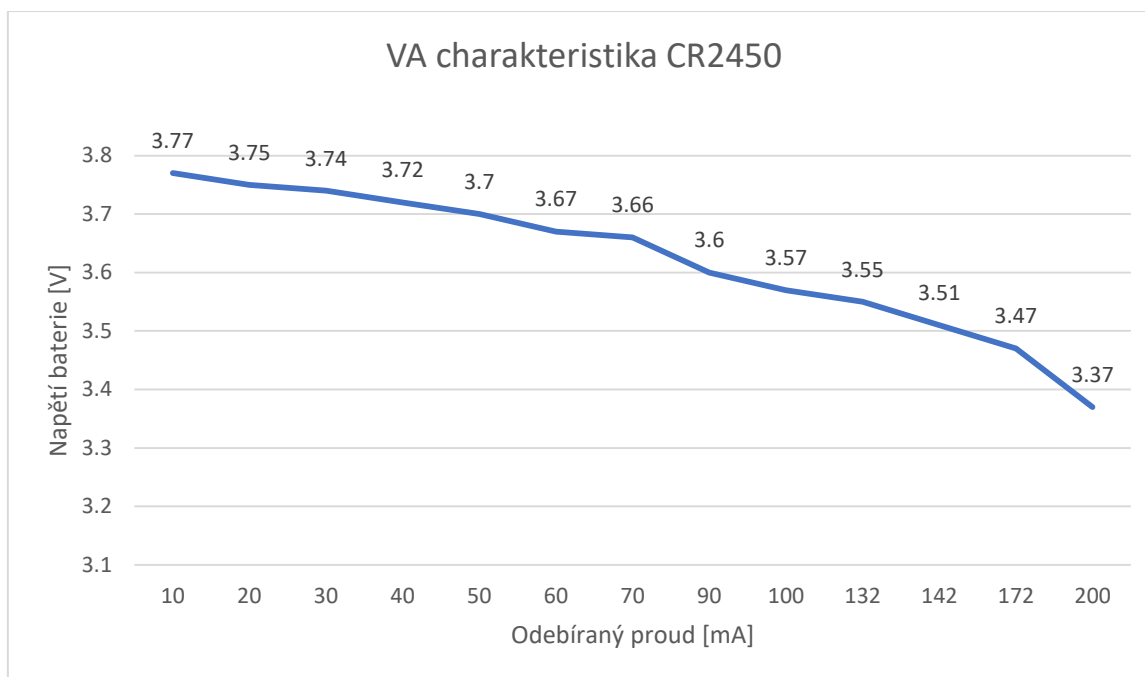
Graf 1: VA charakteristika CR2032.

#### 4.2.2 Voltampérová charakteristika CR2450

Baterie je typu Li-ion a dá se předpokládat lepší charakteristika než nenabíjecí baterie.

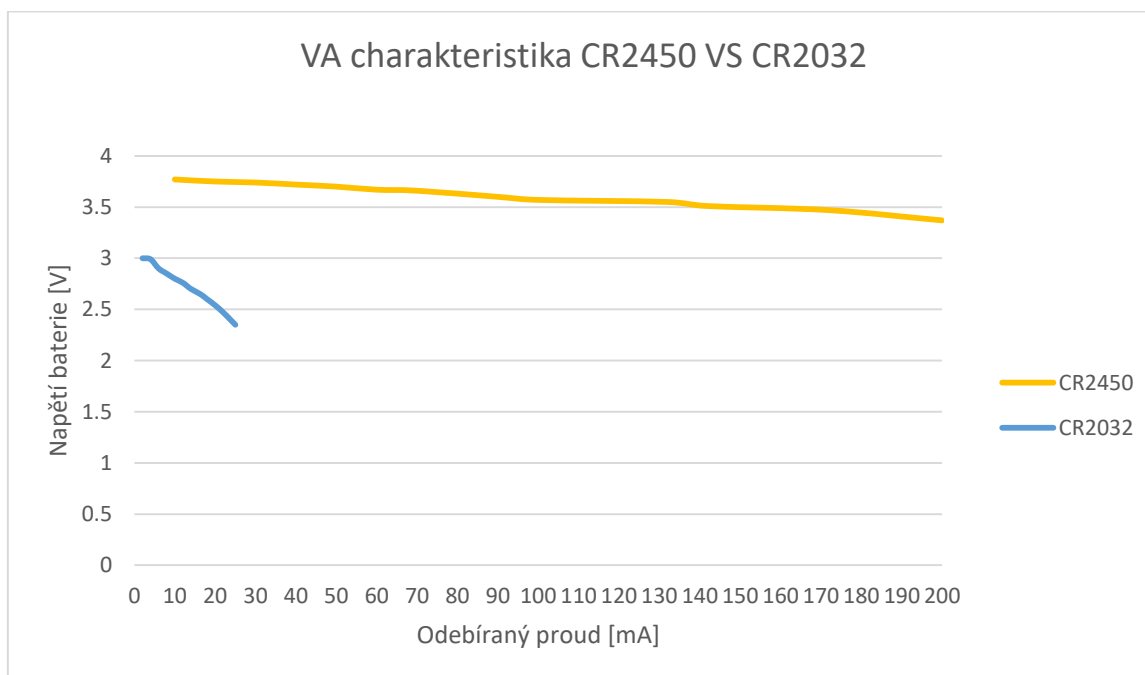
Napětí baterie [V]	Odebíraný proud [mA]
3.77	10
3.75	20
3.74	30
3.72	40
3.7	50
3.67	60
3.66	70
3.6	90
3.6	100
3.55	132
3.51	142
3.47	172
3.37	200

Tabulka 2: VA charakteristika CR2450-Naměřené hodnoty.



Graf 2: VA charakteristika CR2450.

### 4.2.3 Porovnání voltampérových charakteristik CR2032 VS CR2450



Obrázek 25: Porovnání VA charakteristik CR2032 VS CR2450.

#### 4.2.3.1 Závěr

Ukázalo se, že baterie CR2450 je schopna dodávat proudy mnohonásobně větší, než testovaná baterie CR2032. Proudová maxima pro CR2450 jsou 200 mA a pro CR2032 jen 25 mA.



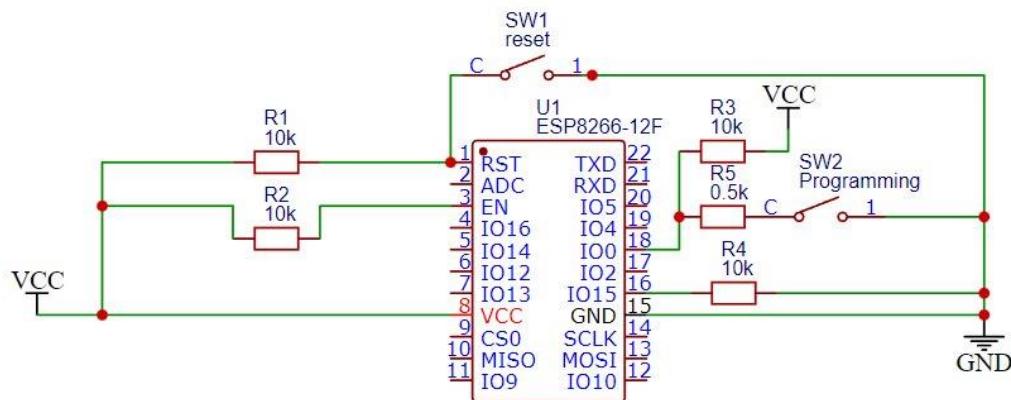
Rovněž linearita napětí u CR2450 je výrazně lepší. Jediná nevýhoda baterie CR2450 je ta, že je rozměrově větší než CR2032.

#### 4.2.4 Výroba vývojového kitu ESP8266-12F

Abychom mohli postavit prototypy s čipsetem ESP8266, musel být nejprve sestaven vývojový kit, který se poté propojil do funkčního celku.

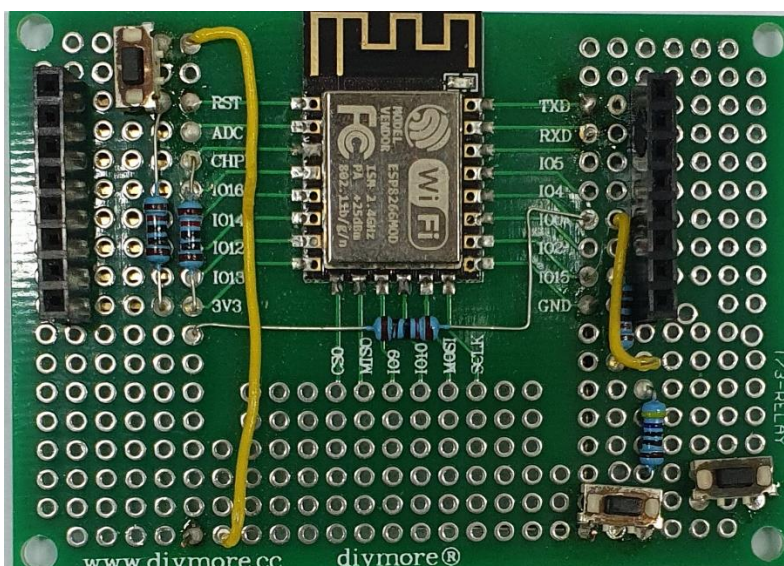
##### 4.2.4.1 Zapojení

Zapojení bylo navrženo na základě datasheetu pro ESP8266-12F. Pro běh programu je potřeba, aby vývod RST byl High a IO0 High. O to se stará rezistor R1 a R3. Pokud chceme zařízení programovat, RST musí být High a IO0 musí být Low. IO0 přepneme do režimu Low stlačením spínače SW2, který připojí vývod na zem. [20]



Obrázek 26: Elektrické schéma vývojové kitu ESP8266-12F.

Pro výrobu bylo použito pájivé pole speciálně vytvořené pro čipset ESP8266, kde jsou již připravené plošky pro tento čipset.

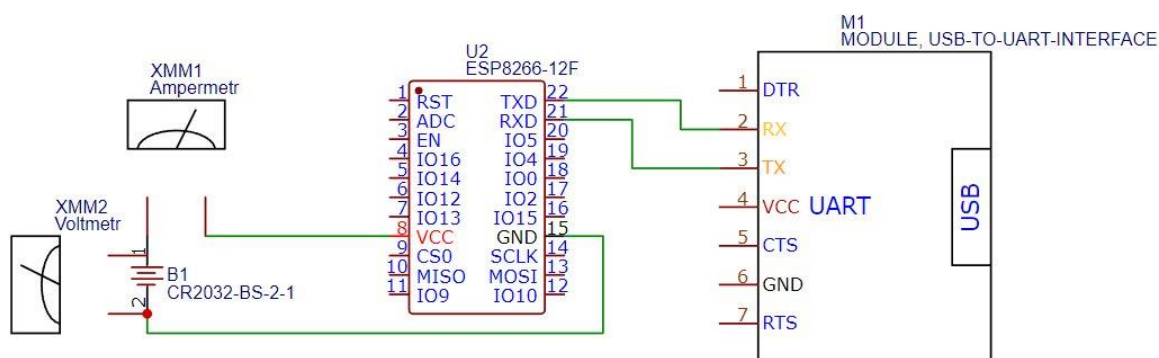


Obrázek 27: Vyrobený vývojový kit ESP8266.

## 4.2.5 ESP8266-12F + baterie CR2032

### 4.2.5.1 Zapojení

Konstrukce tohoto prototypu využívá vývojového kitu ESP8266-12F, který je připojen ke zdroji napětí. Tento zdroj může být reprezentován baterií CR2032 umístěné v držáku Arduino LilyPad, nebo konstantním zdrojem napájení. Dále je propojen s převodníkem USB-UART.



Obrázek 28: Laboratorní zapojení prototypu ESP8266+CR2032.

### 4.2.5.2 Oživení

Nastavení procesoru zůstalo ve výchozím nastavení. Prototyp byl nejdříve připojen k pevnému zdroji napájení a následně byl do něj nahrán jednoduchý program prostřednictvím převodníku USB-UART, který má rozblíkat diodu na modulu ESP8266-12F. Po resetování zařízení, se dioda prototypu rozblíkala a mohlo se tak přejít k testování na baterii. Dioda prototypu se po

připojení na baterii nerozsvítila. Ampérmetr ukazoval hodnotu 45 mA, ale voltmetr jen 1.4 V. Test ukázal, že baterie je měkký zdroj, který pod vyšší zátěží nedokáže udržet konstantní napětí pro tak velký odběr. Tento prototyp se ukázal jako nefunkční.

#### **4.2.6 ESP8266-12F + baterie CR2450**

##### 4.2.6.1 Zapojení

Konstrukce tohoto prototypu využívá dříve sestavený vývojový kit ESP8266-12F, který může být napájen baterií CR2450, nebo konstantním zdrojem napájení a je připojen na převodník USB-UART. Laboratorní zapojení je stejné jako u prototypu pro baterii CR2032.

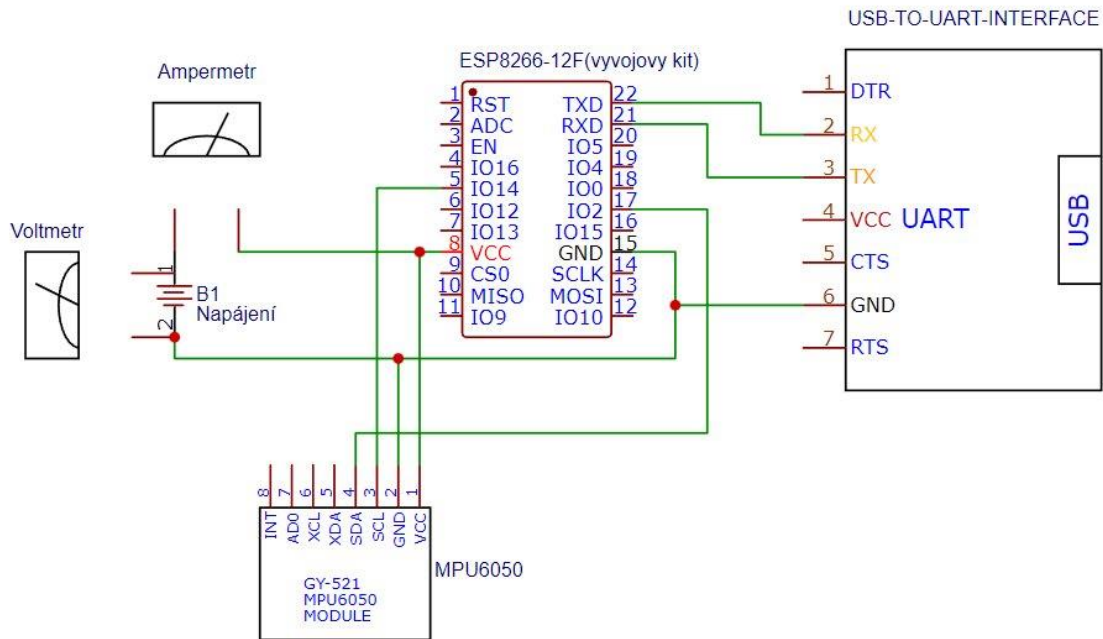
##### 4.2.6.2 Oživení

Po připojení na baterii se dioda rozblíkala a zařízení tak bylo shledáno plně funkční. Ampérmetr ukazoval hodnotu 17 mA a voltmetr 3.8 V. Byl nahrán program prostřednictvím převodníku USB-UART, který využívá Wi-Fi a simuluje tak komunikaci v reálném provozu. Zařízení bylo po nahrání programu funkční a odběr byl v tomto režimu 73 mA.

#### **4.2.7 ESP8266-12F+ CR2450+MPU6050**

##### 4.2.7.1 Zapojení

Již víme, že ESP8266-12F je schopno fungovat s CR2450. Nyní je třeba připojit MPU6050 a dostat tak zapojení, které odpovídá reálnému zařízení v provozu.



Obrázek 29: Zapojení ESP8266-12F+ CR2032+MPU6050.

#### 4.2.7.2 Oživení

Do modulu ESP8266-12F byl nahrán program prostřednictvím převodníku USB-UART, který komunikuje s modulem MPU6050 po sběrnici I2C, čte jeho data a zároveň tyto data bezdrátově vysílá. Odběr prototypu činil 76.4 mA.

### 4.2.8 Atmega328 + CR2032

Na sestavení tohoto prototypu byl použit mikroprocesorem Atmega328 v provedení pro patici DIP28, který má stejné parametry jako ten v provedení SMD. Pomocí programátoru bylo provedeno základní nastavení procesoru.

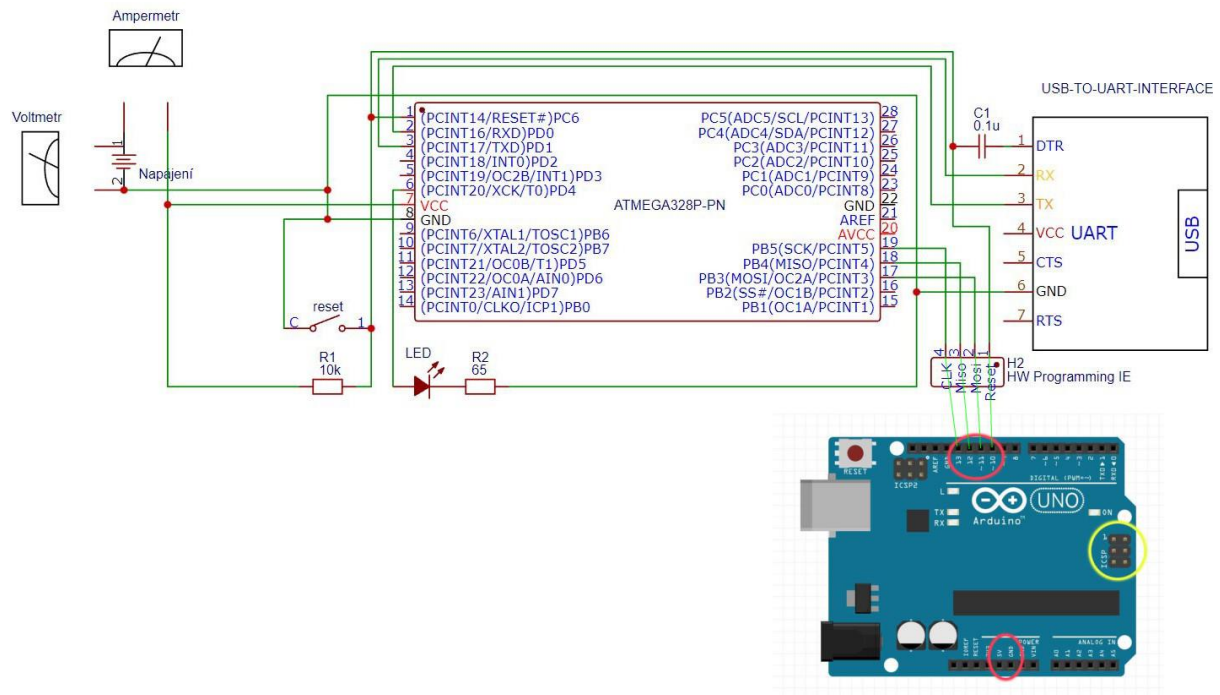
#### 4.2.8.1 HW Programátor

Jako HW programátor bylo použito Arduino UNO, které po nahrání příslušné aplikace, funguje jako běžný HW programátor. [6]

#### 4.2.8.2 Zapojení

Procesor byl vsunut do nepájivého pole a pospojován spojovacími kabely dle schématu zapojení níže. Zdroj napájení může být jak stacionární, tak baterie typu CR2032. V zapojení se

vyskytuje Pull-Up odpor R1, který drží logickou jedničku a zabraňuje tak plavání logického signálu. Dále je zde LED pro signalizační účely, která má proud limitovaný odporem R2. Kondenzátor C1 je zde proto, aby nedocházelo k auto resetu, při nahrávání programu. Tlačítko reset má za úkol restartovat běh programu a vrátit tak obsah registrů na výchozí hodnoty. [6]



Obrázek 30: Zapojení Atmega328 + CR2032 [21].

#### 4.2.8.3 Nastavení

Do mikroprocesoru byl pomocí HW programátoru nahrán Arduino bootloader. Nastaveny byly následující parametry:

- Frekvence procesoru 4 Mhz.
- BOD deaktivován.

Frekvence 4 Mhz byla zvolena s ohledem na kompromis mezi úsporným provozem a frekvencí vzorkování. S touto frekvencí 4 Mhz může mikroprocesor komunikovat prostřednictvím UART rozhraní standardní rychlostí 19200 baudů při tolerované ztrátě paketů. 19200 baudů je 1920 znaků/s, protože dle nastaveného kódování přenosu vychází na přenos jednoho znaku právě 10 bitů a pro naše účely by to mělo být dostačující. [22], [7]

#### 4.2.8.4 Oživení

Prvním krokem bylo připojení zařízení na stacionární zdroj, kde bylo nastaveno napětí 3.3 V dle dokumentace.

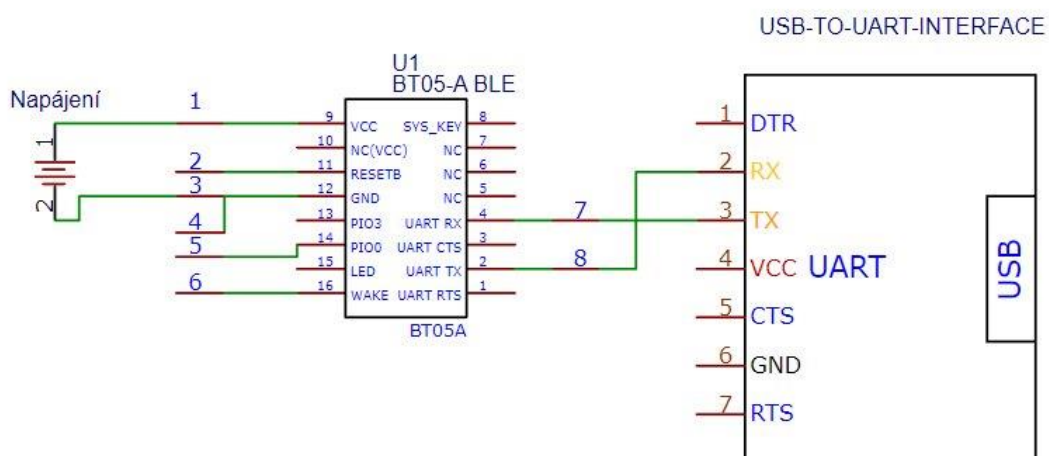
Po úspěšném nastavení mikroprocesoru se HW programátor odpojil a pomocí převodníku USB-UART byl nahrán testovací program, který měl rozblikat diodu R2. Následovalo resetování celého zařízení. Po restartu se dioda R2 rozblikala a byla tak ověřena funkčnost prototypu na stacionárním zdroji napětí. Následovalo připojení baterie CR2032, po jejímž připojení se dioda R2 opět rozblikala. Na ampérmetru byla odečtena hodnota 4.5 mA a na voltmetru 3 V. Mohlo se tedy přejít k testování prototypu se všemi připojenými komponentami, které bude vyvíjený senzor potřebovat pro svou činnost.

#### 4.2.9 Výroba vývojového kitu Bluetooth HM-11 BLE pro Atmega328

Jelikož mikroprocesor Atmega328 neobsahuje Bluetooth, musí se k němu připojit externí modul, aby se zajistila bezdrátové komunikace s okolím. Bylo proto potřeba sestavit vývojový kit, který by se dal propojit pomocí propojovacích kabelů k nepájivému poli, a následně k mikroprocesoru Atmega328.

##### 4.2.9.1 Zapojení

Zařízení bylo napájeno zdrojem 3.3 V.



Obrázek 31: Schéma-Bluetooth vývojový kit.

##### 4.2.9.2 Oživení

Dle dokumentace se tento modul nastavuje pomocí AT příkazů skrze UART rozhraní. Proto bylo potřeba připojit USB-UART převodník přímo k modulu a skrze software RealTerm v PC posílat příkazy. Abychom mohli s modulem komunikovat, musíme nastavit, jakou rychlostí se



budou přijímat a odesílat znaky. Dokumentace uvádí 115200bps, tato rychlost byla nastavena a následně byl připojen pevný zdroj napětí 3.3V. Dále dokumentace uvádí, že k vyvolání nabídky možných nastavení, je potřeba zadat příkaz AT+HELP. Každý příkaz musí končit těmito znaky: \r\n. Po zadání příkazu se ukázala nabídka níže a dá se tedy konstatovat, že modul je z toho hlediska funkční. Aby mohl modul komunikovat s Atmegou328, bylo ho potřeba náležitě nastavit.

```

*****
Command      Description
-----
AT           Check if the command terminal work normally
AT+RESET     Software reboot
AT+VERSION   Get firmware, bluetooth, HCI and LMP version
AT+HELP      List all the commands
AT+NAME      Get/Set local device name
AT+PIN       Get/Set pin code for pairing
AT+PASS      Get/Set pin code for pairing
AT+BAUD      Get/Set baud rate
AT+LADDR     Get local bluetooth address
AT+ADDR      Get local bluetooth address
AT+DEFAULT   Restore factory default
AT+RENEW     Restore factory default
AT+STATE     Get current state
AT+PWRM      Get/Set power on mode(low power)
AT+POWE      Get/Set RF transmit power
AT+SLEEP     Sleep mode
AT+ROLE      Get/Set current role.
AT+PARI      Get/Set UART parity bit.
AT+STOP      Get/Set UART stop bit.
AT+START     System start working.
AT+IMME      System wait for command when power on.
AT+IBEA      Switch iBeacon mode.
AT+IBE0      Set iBeacon UUID 0.
AT+IBE1      Set iBeacon UUID 1.
AT+IBE2      Set iBeacon UUID 2.
AT+IBE3      Set iBeacon UUID 3.
AT+MARJ      Set iBeacon MARJ .
AT+MINO      Set iBeacon MINO .
AT+MEA       Set iBeacon MEA .
AT+NOTI      Notify connection event .
AT+UUID      Get/Set system SERVER_UUID .
AT+CHAR      Get/Set system CHAR_UUID .
-----
Note: (M) = The command support slave mode only.
*****

```

Obrázek 32: Bluetooth modul - seznam příkazů.

Povel (na TxD)	Odezva (na RxD)	Význam
AT	OK	kontrola funkce
AT+BAUD1	OK1200	nastavení přenosové rychlosti 1200 Bd
AT+BAUD2	OK2400	nastavení přenosové rychlosti 2400 Bd
AT+BAUD3	OK4800	nastavení přenosové rychlosti 4800 Bd
AT+BAUD4	OK9600	nastavení přenosové rychlosti 9600 Bd
AT+BAUD5	OK19200	nastavení přenosové rychlosti 19200 Bd
AT+BAUD6	OK38400	nastavení přenosové rychlosti 38400 Bd
AT+BAUD7	OK57600	nastavení přenosové rychlosti 57600 Bd
AT+BAUD8	OK115200	nastavení přenosové rychlosti 115200 Bd

*Tabulka 3: Nastavení UART rychlosti Bluetooth modulu.*

#### 4.2.9.3 Nastavení

- 1) Rychlost přenosu-Odvíjí se od rychlosti mikroprocesoru, s kterým modul komunikuje. Atmega328 byla nastavena na 4 Mhz. Dle datasheetu pro Atmegu328 je maximální rychlost, při které ještě nedochází ke ztrátě paketů 19200 baudů. Rychlost přenosu musela být proto nastavena také na 19200 baudů příkazem AT+BAUD5. [7], [23]
- 2) Vyzařovací výkon-Byl nastaven nejvyšší výkon 6 dBm příkazem AT+POWER3. Experimentování však ukázalo, že nastavení nemá vliv na vysílací výkon. [23]
- 3) Role-Pro naší aplikaci bylo potřeba nastavit, aby Bluetooth pracoval v roli Slave. Příkazem AT+ROLE0 byl tento mód nastaven. [23]
- 4) Jméno zařízení-Bylo potřeba zařízení vhodně pojmenovat proto, abychom ho snáze našli v seznamu Bluetooth zařízení při snaze se k němu připojit. Bylo zvoleno jméno SMARTSPHERE. Příkazem AT+NAMESMARTSPHERE bylo zařízení pojmenováno. [23]

#### 4.2.10 Atmega328+CR2032+Bluetooth+SPI FLASH+MPU6050

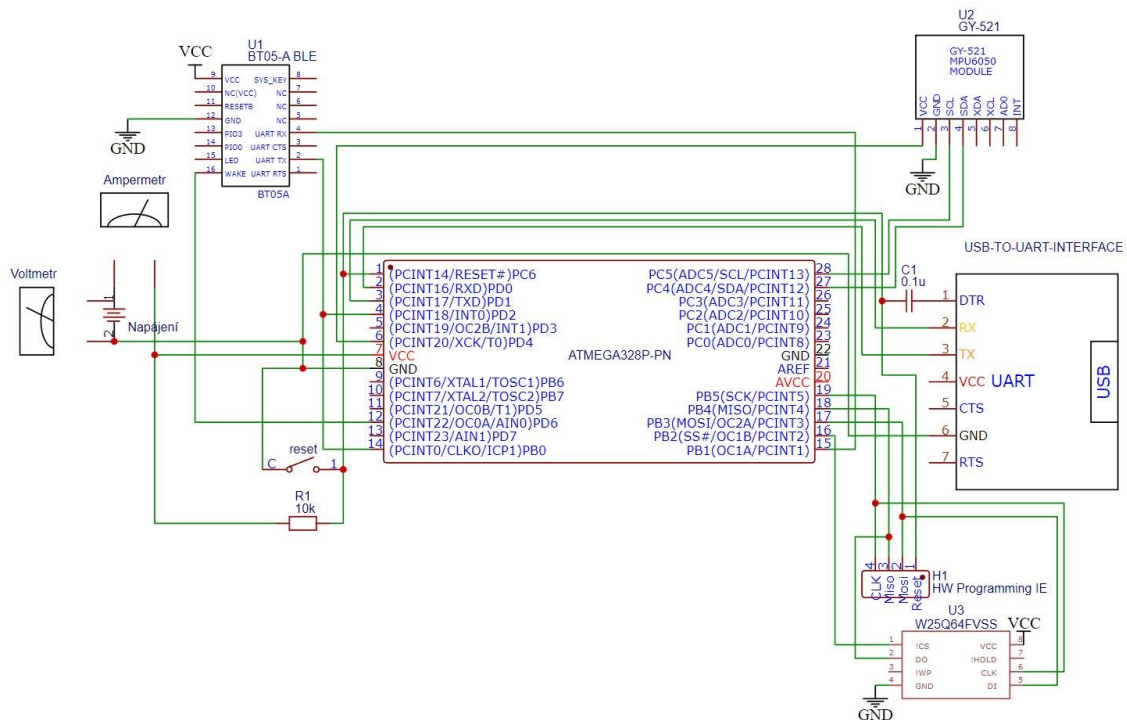
Zapojení Atmega328+CR2032 se ukázalo jako životaschopné a proto nyní můžeme na tomto základě sestavit prototyp, který bude navíc obsahovat Bluetooth modul, SPI FLASH paměť a MPU6050. Tento prototyp by měl být z hlediska parametrů nejslibnější a to z hlediska rozměrů, flexibility, spotřeby elektrického proudu a nastavení parametrů. Panovaly zde ale obavy z toho, že baterie nebude schopna dodávat dostatečný proud při požadovaném napětí. Limitujícím faktorem je zde minimální napětí pro MPU6050, které činí 2.375 V.



#### 4.2.10.1 Zapojení

Toto zapojení kopíruje zapojení Atmega328 + CR2032 a přidává MPU6050 označené ve schématu jako U2 , SPI FLASH paměť označenou jako U3 a Bluetooth modul označený jako U1. Jednotlivé komponenty byly umístěny do nepájivého pole a propojeny pomocí propojovacích kabelů. Moduly byly připojeny k nepájivému poli taktéž pomocí kabelů.

1. Bluetooth modul komunikuje s mikroprocesorem prostřednictvím UART rozhraní, které je na straně procesoru vytvořeno softwarově následovně: na pinu 15 je TX a na pinu 14 je RX. Ze spánku je Bluetooth buzen prostřednictvím pinu 16 na Bluetooth, který je napojen na pin 12 procesoru. UART RX je propojen z Bluetooth pinu 4 na pin 15 procesoru. UART TX je na straně Bluetooth z pinu 2 propojen do procesoru na pinu 14 a zároveň na pin přerušování 4, který probouzí mikroprocesorem ze spánku.
2. MPU6050 komunikuje s mikroprocesorem pomocí sběrnice I2C na pinech SDA a SCL. Napájení Vcc je připojeno na pin 6 procesoru a slouží tak k řízenému napájení modulu.
3. SPI FLASH komunikuje standardně přes SPI rozhraní.



Obrázek 33: Schéma-Atmega+CR2032+Bluetooth+SPI FLASH.

#### 4.2.10.2 Oživení

Procesor Atmega byl již nastaven z předchozí prototypu a nebylo potřeba provádět jakékoliv modifikace. Bluetooth byl rovněž nastaven a ostatní komponenty nastavení nepotřebují. Bylo

tedy možné připojit prototyp na napětí zdroje 3.3 V a ověřit funkčnost. K prototypu byl připojen převodník UART-USB a nahrán program, který ověřil funkčnost všech komponent. Jednalo se zejména o ověření funkce Bluetooth adaptéru, MPU6050 a SPI FLASH paměti. Vše se ukázalo jako funkční. Byl měřen odebíraný elektrický proud, při plné zátěži tzn. všechny komponenty byly v provozu. Ampérmetr ukazoval hodnotu 16.42 mA. Při tomto odběru by teoreticky měl prototyp fungovat i na baterii CR2032. Po připojení prototypu na baterii CR2032 a přivedení všech komponent k činnosti zařízení fungovalo stabilně po testovací dobu 5 minut. Byl také měřen pokles napětí na baterii, který činil 2.84 V. Odebíraný proud se držel na hodnotě 16 mA. Můžeme tedy konstatovat, že tento prototyp je zcela funkční.

### 4.3 Porovnání prototypů

S pomocí vícekritériální analýzy metody Váženého součtu, byla vytvořena tabulka s parametry prototypů, pro usnadnění výběru vhodného prototypu.

- a) Odběr při práci všech komponent byl měřen za podmínek, kdy zařízení vysílalo signál, měřilo a zároveň zpracovávalo přijímaná data. Byla zde zvolena vysoká váha kritéria, protože dle zadání práce musí mít naše zařízení co nejmenší odběr.
- b) Odběr v režimu Modem sleep byl měřen tehdy, když MPU6050 bylo vyplé a mikroprocesor v režimu spánku. Bezdrátové zařízení však bylo schopno přijímat signál a probudit mikroprocesor ze spánku. Byla zde zvolena vysoká váha kritéria, protože dle zadání práce musí mít naše zařízení co nejmenší odběr.
- c) Povrch největší součástky znamená největší rozměr součástky, která byla potřeba k sestavení prototypu a tím určovala minimální velikost celého zařízení. Byla zde zvolena vysoká váha kritéria, protože dle zadání práce musí být naše zařízení co nejmenší.
- d) Náročnost sestavení vyjadřuje míru, jak moc náročné je sestavit prototyp vzhledem k množství komponent a jejich propojení do funkčního celku.
- e) Podpora komunity reflektuje subjektivní názor autora této práce, při hledání informací o konkrétních komponentech prototypu jako celku.
- f) SRAM značí paměť pro práci s proměnnými
- g) FLASH zobrazuje paměť pro kód naší aplikace.
- h) integrované Bluetooth nebo Wi-Fi indikuje, zdali má mikroprocesor prototypu integrován vysílač/přijímač signálu.

	Odběr při práci všech komponent mA	Odběr v režimu modem sleep mA	Povrch největší součástky v mm <sup>2</sup>	Náročnost sestavení(0;10)	Podpora kumunity(0;10)	SRAM v KB	FLASH v KB	integrované Bluetooth nebo wifi (0;1)	
Atmega328+CR2032+Bluetooth+SPI FLASH+MPU6050	16	1	462.25	3	10	2	32	0	
ESP8266-12F+CR2450+MPU6050	76.4	27	784	5	8	50	4000	1	
Povaha (max/min)	min	min	min	min	max	max	max	max	
Váhy	0.2	0.2	0.2	0.06	0.06	0.06	0.16	0.06	
-									
nejlepší (ideální)	16	1	462.25	3	10	50	4000	1	
nejhorší (bazální)	76.4	27	784	5	8	2	32	0	
-									
									Výsledná hodnota
Atmega328+CR2032+Bluetooth+SPI FLASH+MPU6050	1.000	1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.720
ESP8266-12F+CR2450+MPU6050	0.000	0.000	0.000	0.000	0.000	1.000	1.000	1.000	0.280

Tabulka 4: Porovnání prototypů.

Prototyp s mikroprocesorem Atmega328 jasně dominuje oproti řešení s ESP8266-12F a proto byl vybrán pro tuto práci jako nejvhodnější.

## 4.4 Vývoj software pro vybraný prototyp

Byl vybrán prototyp s mikroprocesorem Atmega328. Nyní může začít práce na vývoji software, který bude umět pracovat efektivně a hospodárně s připojenými komponenty a poskytnou uživatelský interface pro ovládání vyvíjeného senzoru.

### 4.4.1 Požadavky

Od softwaru se očekává, že bude umět zobrazovat uživatelské rozhraní na Bluetooth terminálu, které bude v textové podobě a bude zde možné

- a) Spouštět měření a posílat data na Bluetooth terminál.
- b) Zastavit měření.
- c) Spouštět měření, zapisovat data do FLASH paměti a přepnout Bluetooth do režimu spánku.
- d) Číst data z FLASH paměti a poslat je do Bluetooth terminálu.
- e) Smazat data z FLASH paměti.

- f) Nastavit parametry Bluetooth modulu.
- g) Uvést zařízení do režimu Modem sleep.
- h) Zobrazit nápovědu.

Software musí rovněž šetřit baterii zařízení, a proto zde musí být rutina, která se stará o hospodárny provoz. V době nečinnosti by mělo být zařízení v módu Modem sleep.

#### 4.4.2 Úvod

Pro vývoj softwaru bylo použito Visual Studio Code s pluginem Arduino IDE. Samotný kód je členěn do několika navzájem spolupracujících knihoven, které využívá základní knihovna WMAINFCE s rozhraním IMAINFCE. Metody této knihovny jsou pak využívány v základním Arduino sketchy. Kromě vlastních knihoven bylo použito knihoven třetích stran.

Název knihovny	Datový typ Popis
AltSoftSerial.h	Knihovna emulující rozhraní UART na Arduino pinech 8 a 9.
MPU6050_6Axis_MotionApps20.h	Knihovna zprostředkávající komunikaci s MPU6050. Využívá DMP procesor. Aby knihovna fungovala správně, musí být nastavena sensitivity akcelerometru na defaultní hodnotu která je pro naši verzi MPU6050 8192 LSB/g. Tato hodnota koresponduje s definovanou hodnotou MPU6050_ACCEL_FS_2 a nastavuje akcelerometr na nejvyšší sensitivity, kde je maximální možné přetížení 2 g. Gyroskop můžeme nastavit dle uvážení.
SPIMemory.h	Knihovna řídící přístup do paměti FLASH. Nabízí základní funkce jako zápis, čtení a mazání. V této knihovně bylo nutné odkomentova <code>//#define FASTWRITE</code> pro rychlý zápis.
Arduino.h	Základní knihovna pro práci s frameworkem Arduino, obsahuje základní funkce pro Atmega328.
I2Cdev.h	Obsahuje metody, potřebné pro komunikaci prostřednictvím I2C. V této knihovně bylo nutné odkomentovat <code>//#define I2CDEV_BUILTIN_FASTWIRE</code> pro zapnutí zrychlené komunikace.

Tabulka 5: Knihovny třetích stran.

```
#include <avr/sleep.h>
#include "Sample.h"
#include "SampleConfig.h"

SampleConfig config;
IMAINFCE& mainFce = config.getMain();

void setup() {
  mainFce.init();
}

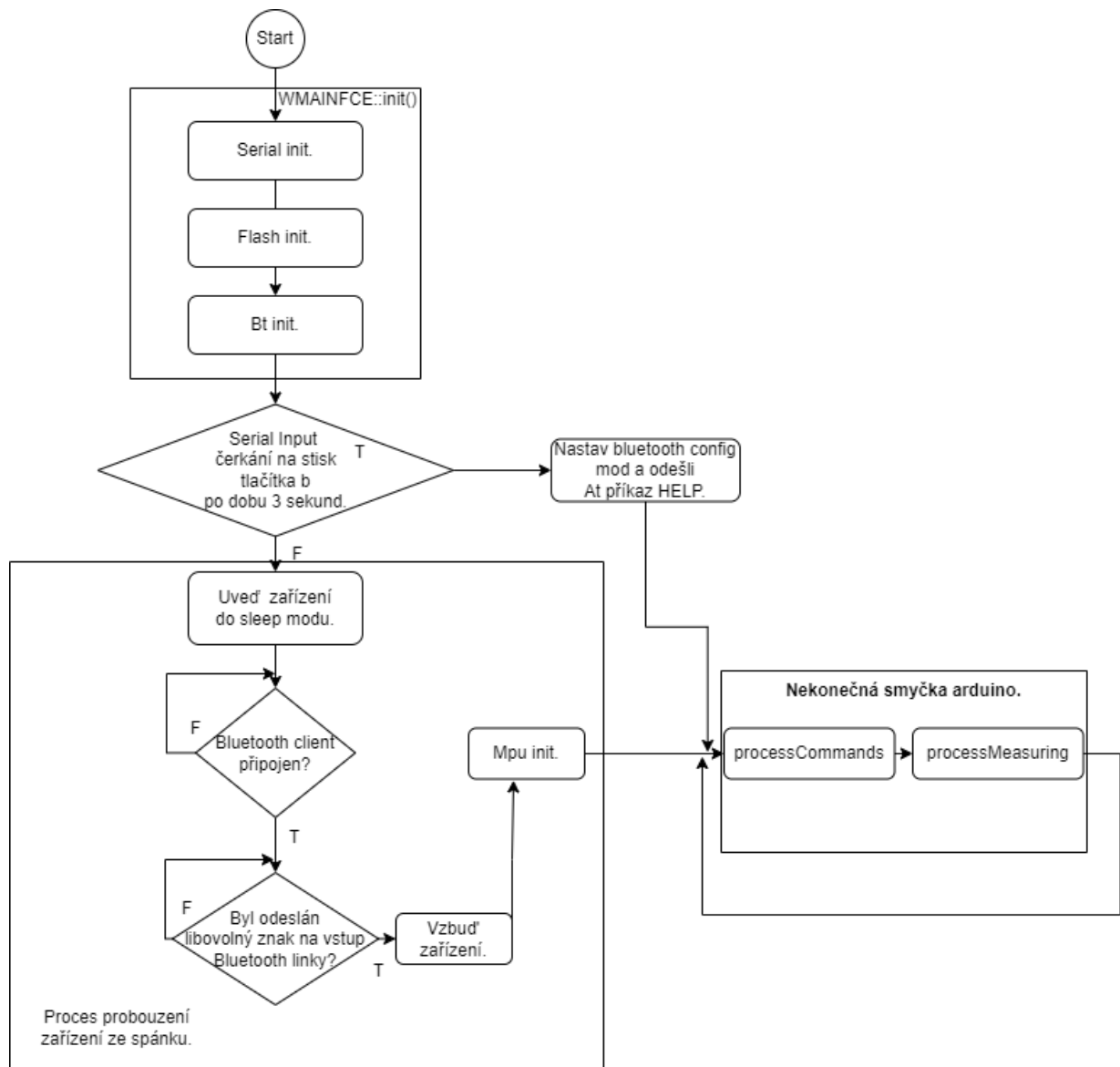
void loop() {

  mainFce.runFunctionsInLoop();
}
```

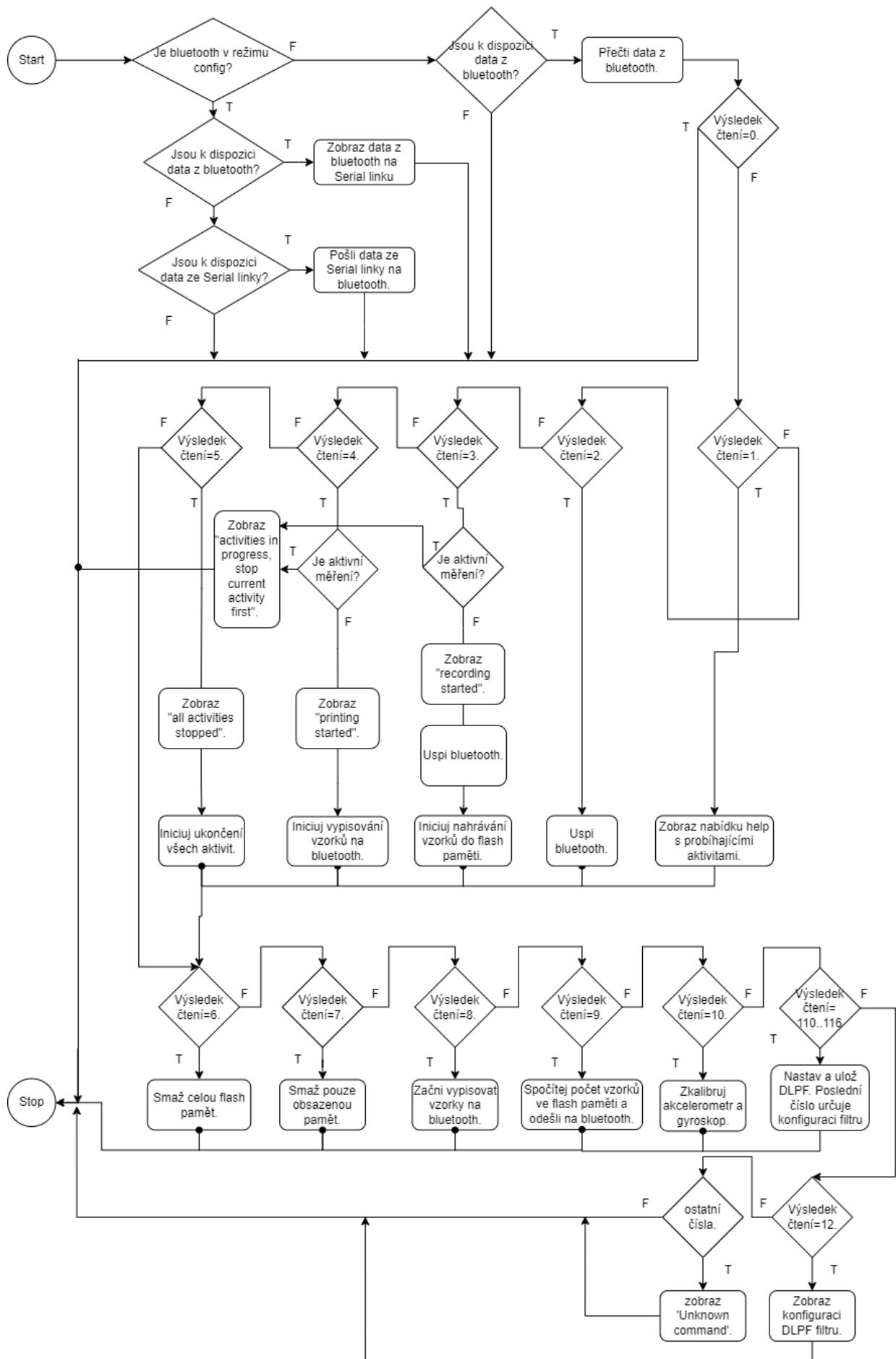
Obrázek 34: Arduino sketch-ukázka kódu.

### 4.4.3 Vývojový diagram

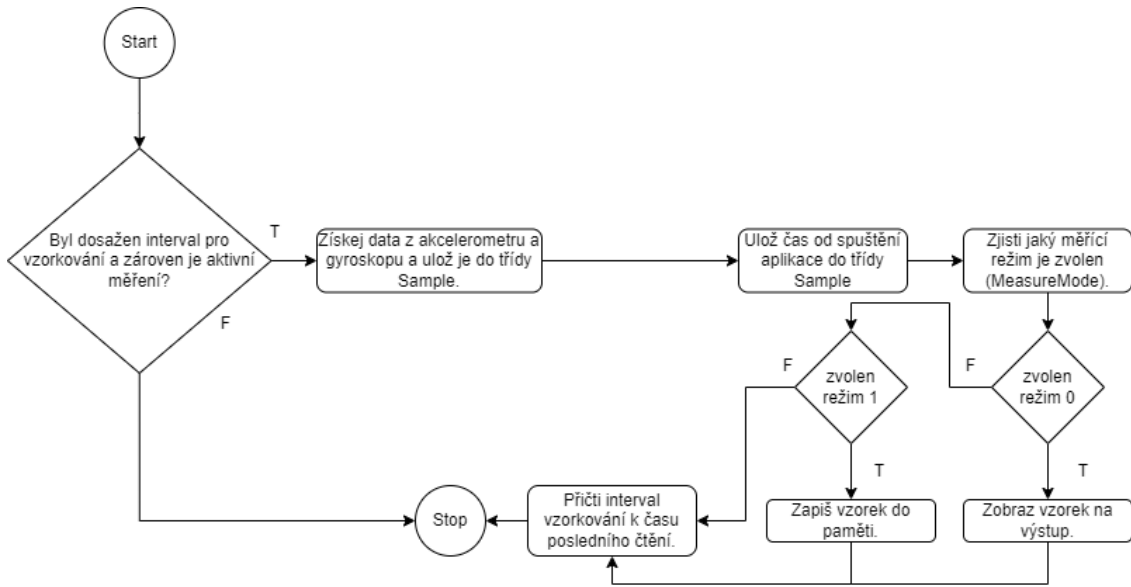
V rozhodovacích blocích T znamená true a F znamená false.



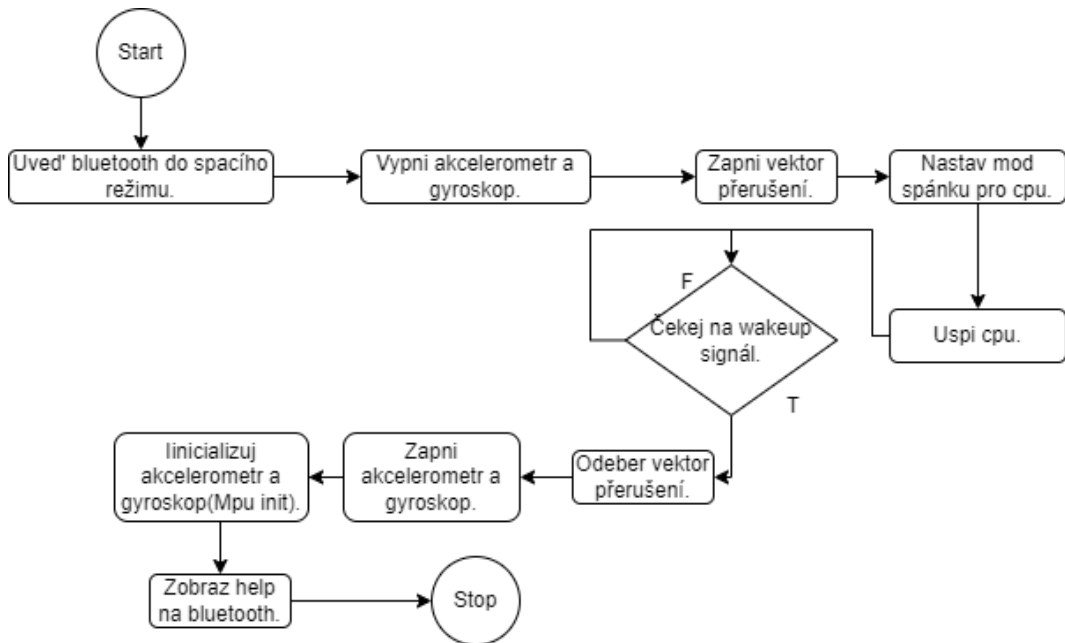
Obrázek 35: Diagram MainLoop.



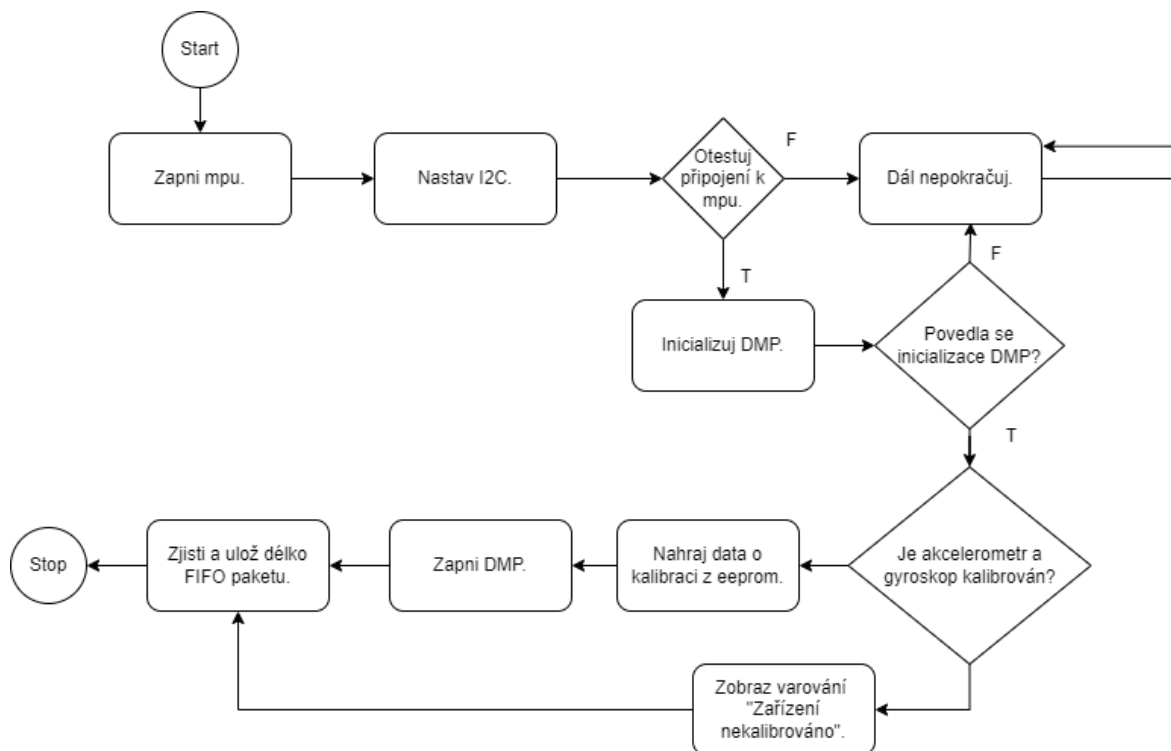
Obrázek 36: Diagram ProcessCommands.



Obrázek 37: Diagram ProcessMeasuring.



Obrázek 38: Diagram SleepMode.



Obrázek 39: Diagram MpuInit.

#### 4.4.4 Konstantně nastavené parametry senzoru.

Název parametru	Datový typ	Popis
MPU6050::setFullScaleAccelRange(0)	int	Nastavuje sensitivitu akcelerometru na 2 g.
MPU6050::setFullScaleGyroRange(0)	int	Nastavuje sensitivitu akcelerometru na 250°/sec.
MPU6050:: setDHPFMode(0)		Vypíná high-pass filtr.
MPU6050:: setRate(4)	int	Nastavuje, jak rychle bude MPU6050 samplovat vzorky do FIFO bufferu. Sample Rate = Gyroscope Output Rate / (1 + SMPLRT_DIV) Sample Rate = 1 KHz / (1 + 4) Sample Rate = 200 Mhz.
WMAINFCE::sensorReadInterval=20	unsigned long	Určuje vzorkovací frekvenci, s jakou jsou data z MPU6050 zobrazována na Bluetooth a zapisována do paměti prostřednictvím metody processMeasuring. Jednotky (ms).
PRINTSAMPLE_VSE=0	definovaná konstanta int	Definuje mód vypisování metody printSample.

Tabulka 6: Nastavené hodnoty některých konstant a proměnných [24].

#### 4.4.5 Závěr z programového vývoje

Podářilo se napsat programový kód, který má následující funkcionality.

1. Úsporný režim.
2. Připojení k senzoru pomocí Bluetooth.
3. Konfigurace Bluetooth modulu prostřednictvím UART Serial.



4. Měření pomocí gyroskopu a akcelerometru za pomoci DMP procesoru.
5. Zobrazení hodnot: čas, akcelerace bez gravitace (X, Y, Z) na UART Bluetooth.
6. Zápis hodnot: čas, akcelerace bez gravitace (X, Y, Z) do paměti FLASH.
7. Mazání a čtení z paměti FLASH.
8. Kalibrace MPU6050.

#### 4.4.5.1 Samplovací frekvence

Frekvence je omezena:

1. Taktem mikroprocesoru.
2. Rychlostí UART přenosu, která v našem případě činí 19200 baudů/s. Při přenosu 24 znaků na řádek, kdy každý znak představuje 10 bitů, je frekvence omezena dle vzorce *frekvence přenosu dat*  $= \frac{1920}{24} = 80 \text{ Hz}$ .
3. Rychlostí čtení dat z FFIFO bufferu MPU6050. Tato rychlost je omezena taktem sběrnice I2C. Při 200 Mhz bylo změřeno 14 ms.
4. V případě zápisu do paměti FLASH je omezení dáno rychlostí sběrnice SPI a samotné FLASH. UART přenos zde rychlost neomezuje. Byla změřena rychlost zápisu jednoho řádku, která činí 4 ms.

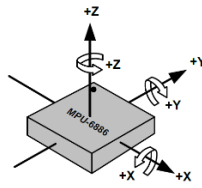
#### 4.4.6 Otestování programu na vybraném prototypu

Program byl úspěšně nahrán do vybraného prototypu. Jeho velikost tzn. velikost sketche činí 29900 bytů, což je 92% FLASH paměti Atmega328, vyhrazené pro programová data. Co se týče dynamické paměti SRAM, která slouží pro práci s proměnnými, tak tam program zabírá 1111 bytů, což je 54% Atmega328 paměti SRAM. Je zde tedy stále prostor, pro úpravu kódu a přidání nových funkcionalit. Lze zde uspořít paměťové místo a výpočetní výkon tak, že se jako výstupní hodnoty použijí přímo hodnoty, které jsou čteny z FifoBufferu a nebudou se dělit hodnotami sensitivity. Toto dělení lze uskutečnit až při zpracování dat na jiném výpočetním prostředku.

##### 4.4.6.1 Testování

Testování mělo za úkol prověřit jak softwarovou, tak hardwarovou část. Byly otestovány funkcionality a to zejména měření hodnot akcelerace ve všech osách. Senzor byl položen na

vodorovnou plochu tak, že osa Z směřuje kolmo na vodorovnou plochu dle následujícího obrázku.



Obrázek 40: MPU6050 a jeho osy.

Software byl zkompileován s definovanou konstantou PRINTSAMPLE\_VSE=1 tak, aby bylo možné odečítat hodnoty akcelerometru, při zachování gravitační složky.

Bylo spuštěno vypisování vzorků volbou 4 v menu senzoru. Senzor byl 1 minutu ve výchozí pozici a poté rotován postupně tak, aby všechny osy směřovaly kolmo na vodorovnou plochu. Data byla zaznamenána a zkoumána. Bylo zjištěno, že osa Z zobrazuje hodnotu z proměnné  $az = -8192$ , která indikuje dle datasheetu hodnotu  $1g$ . Tato hodnota odpovídá zemské gravitaci. Každá osa, která byla rotována, ukazovala hodnotu  $-8192$ . Dále bylo senzorem pohybováno v jednotlivých osách a zkoumána lineární akcelerace. Senzor zaznamenal změnu hodnot v každé akcelerované ose.

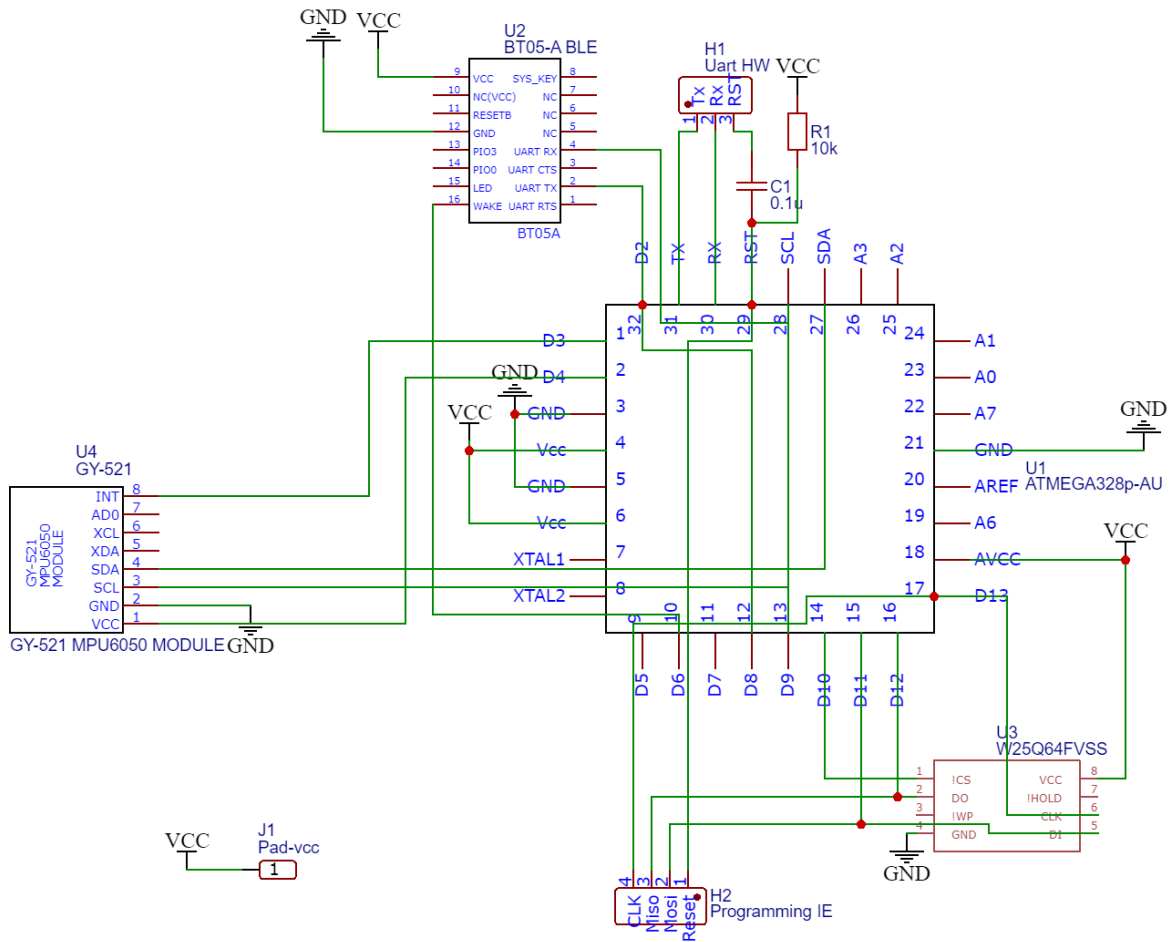
#### 4.4.6.2 Závěr

Ze zjištěných hodnot vyplívá, že senzor zobrazuje data korektně. Následně byly otestovány zbývající funkcionality s konstatováním, že jsou zcela funkční.

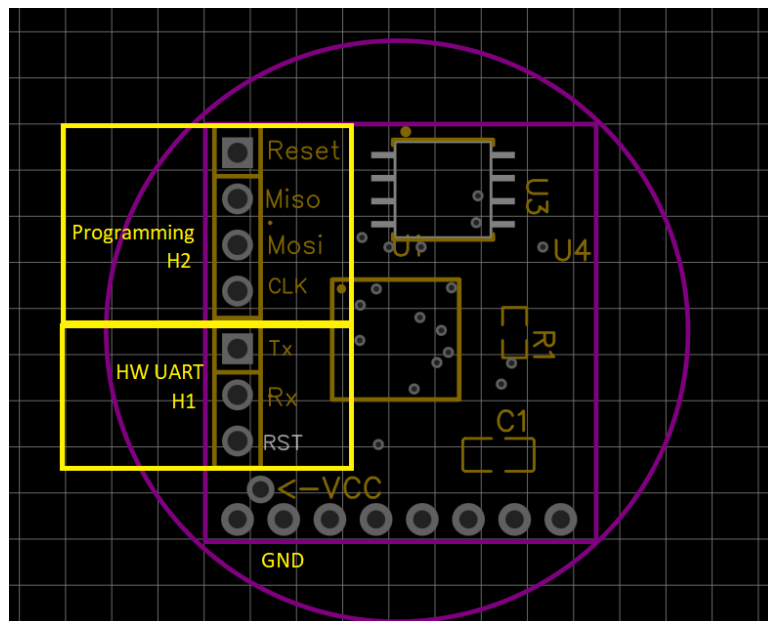
## 4.5 Sestavení vlastního senzoru

### 4.5.1 Schéma

Byl vytvořen návrh plošného spoje v programu EasyEda dle vybraného prototypového schématu s dílčími změnami.



Obrázek 41: Schéma vlastního senzoru.



Obrázek 42: PCB-zapojení pinů.

## 4.5.2 Návrh PCB

Plošné uspořádání elektronické části senzoru, bylo navrženo jako vícevrstvé, kdy jedno PCB slouží jako držák baterie CR2032 a druhé PCB nese veškeré elektronické komponenty. Tyto dvě PCB se následně propojí pomocí spojovacích pinů.

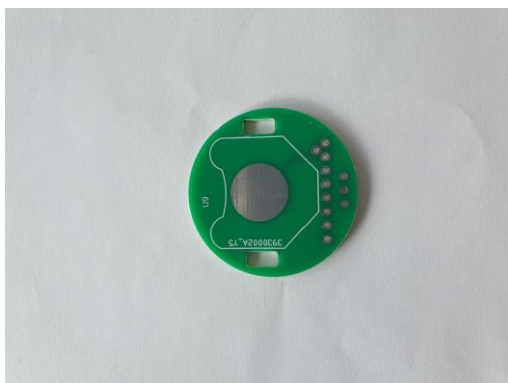
## 4.5.3 Seznam použitých komponent

	Název	Popis	Množství	Cena [kč]
1	HM-11 BLE Modul Bluetooth 4.0 CC2541 CC2540	Bluetooth modul.	1	155
2	Atmega328PU-TH	CPU	1	250
3	W25Q64	Spí FLASH memory.	1	47
4	BC-2002	Držák CR2032.	1	6.50
5	GY-521 MPU6050	Akcelerometr a gyroskop.	1	90
6	YAGEO CKS0603 100n/50V X7R 10%	keramický kondenzátor SMD	1	2.50
7	YAGEO R1206 10k 0,25W 1%	SMD rezistor	1	1.6
8	LA215009 0,3A 50VDC	Posuvný přepínač	1	2
9	Vysoká kolíková lišta - Rovná, 40 Pin samec		1	16
10	DuPont rovná kolíková liště - 40Pin, samec černá		1	3
11	PCB	Deska plošných spojů	2	279
<b>Dohromady</b>				<b>852</b>

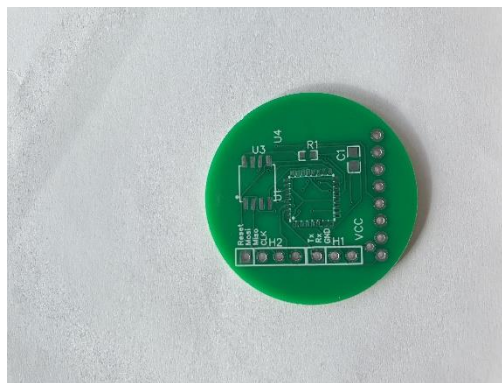
Tabulka 7: Seznam použitých komponent.

## 4.5.4 Výroba PCB

Výroba samotných PCB byla přenechána společnosti LCSC electronics [25], specializující se na výrobu PCB. Bylo objednáno celkem 15 kusů PCB desek. Z toho 5 kusů představovalo držák baterie kruhového tvaru, 5 kusů držáku baterie čtvercového tvaru a 5 kusů základní desky, na které byl umístěn mikroprocesor Atmega328. Držák baterie čtvercového tvaru se nejevil jako vhodný a použil se místo toho držák kruhový. Celkem i s objednááním bylo zaplaceno 279 Kč. Na následujících dvou obrázcích je již hotové PCB před osazením.



Obrázek 43: PCB-držák baterie.



Obrázek 44: PCB-mikroprocesorová část.

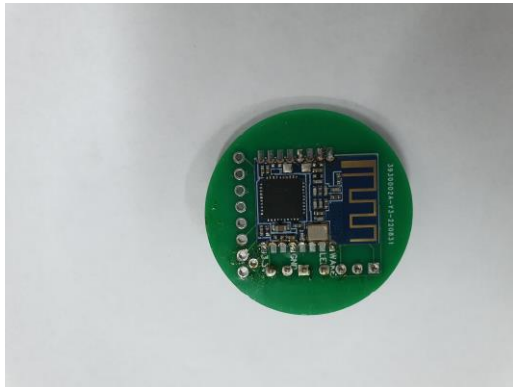
#### 4.5.5 Osazení PCB

Osazení bylo částečně proveden za pomoci pájedla a horkovzdušné stanice. Horkovzdušná stanice byla použita na pájení mikroprocesoru Atmega328, Spi FLASH paměti, odporu a kondenzátoru. Jako pájka byla použita Pájecí pasta Easy Print (Sn62Pb36Ag2) 8g.

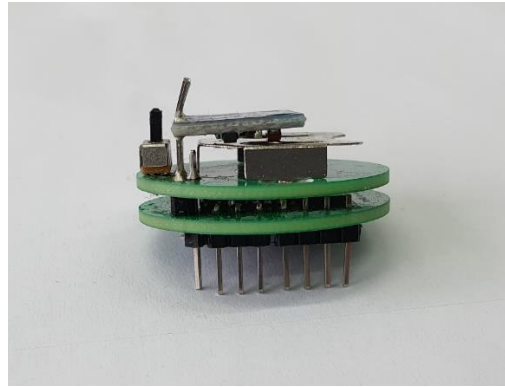


Obrázek 45: Pájecí pasta.

Postup pájen byl takový, že se nejprve PCB očistila od nečistot pomocí technického lihu. Následně se nanese malé množství pasty na místo PCB, kde byla kontaktní plocha pro konkrétní součástku. Součástka se lehce zamáčkla do nanesené pasty a následně byl na kontaktní místo aplikován horký vzduch na několik sekund. Tím došlo k vytvoření vodivého spojení mezi součástkou a PCB. Horkovzdušná stanice byla nastavena na nízký proud vzduchu cca 25% maximálního výkonu, aby pájené součástky neuletěly z PCB. Teplota byla nastavena na 380°C.



Obrázek 46: PCB-osazení- Bluetooth modul.



Obrázek 47: PCB-osazení-pohled z boku.



Obrázek 48: PCB-osazení-pohled shora.



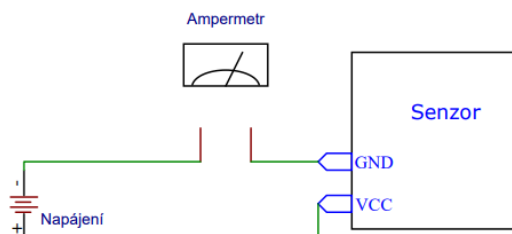
Obrázek 49: PCB-osazení-pohled-zespodu.

#### 4.5.5.1 Zhodnocení

Bylo zjištěno, že pájecí plošky pro mikroprocesor jsou příliš malé a bylo tak obtížné mikroprocesor umístit a připájet. Rovněž ploška pro odpor R1 by mohla být větší, Naopak ploška pro C1 mohla být menší. Dále bylo nutno spojit pomocí pájení přepínač U1 sloužící k zapnutí či vypnutí senzoru na PCB, kde je umístěna baterie. Bylo také zjištěno, že paměť FLASH nepracuje korektně, pokud se nespojí kontakty 8 a7 a proto byly tyto kontakty později spájeny dohromady.

## 4.6 Oživení senzoru

### 4.6.1 Zapojení



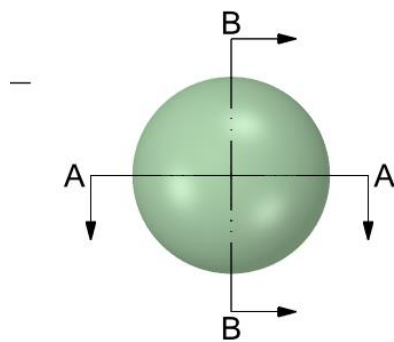
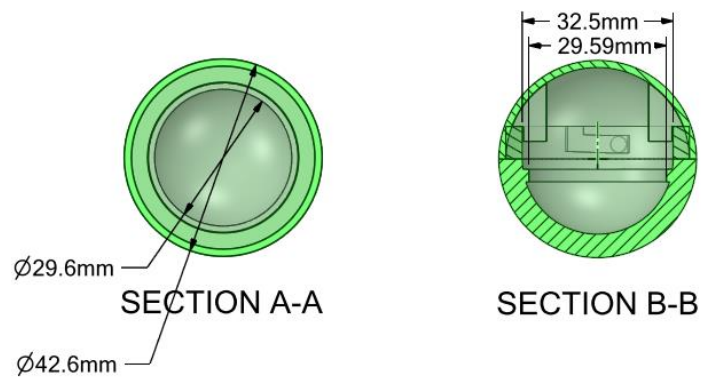
Obrázek 50: Oživení senzoru-elektrické zapojení měřícího obvodu.

Senzor byl připojen na zdroj napětí 3.3 V vodiči tak, že kladný pól byl připojen na kontakt VCC, záporný pól byl připojen na kontakt Gnd dle Obrázek 42: PCB-zapojení pinů. Pomocí HW programátoru byl nahrán Arduino bootloader a nastaveny parametry procesoru dle vybraného prototypu. Byl rovněž nahrán vyvinutý software a ověřena funkčnost celého řešení. Testováním se zjistilo, že hotový senzor se chová stejně jako prototyp a dá se tedy konstatovat, že senzor je funkční a připraven pro následná měření. Nejprve se však muselo vyrobit pouzdro, které by sloužilo jako schránka pro elektronickou část senzoru.

## 4.7 Pouzdro

Pouzdro bylo vymodelováno v 3D programu tak, aby spodní polokoule sloužila pro uložení elektronické části senzoru a vrchní polokoule se zaklesla do té spodní za pomocí západkového systému viditelného v sekci B.

### 4.7.1 Půdorys



Obrázek 51: 3D model-půdorys.

### 4.7.2 Vytisknutý 3D model



Obrázek 52: Vytisknutý 3D model.



### 4.7.3 Zhodnocení vytisknutého modelu

Vytisknutý model musel být upraven tak, aby na sebe jednotlivé plochy dosedly a obě hemisféry bylo možné spojit. Materiál byl odebrán pomocí brusného papíru.

## 4.8 Měření

Senzor bylo potřeba otestovat, zdali poskytuje korektní data v závislosti na trajektorii při jeho pohybu způsobené vnější silou. Dále byl zjištěn dosah bezdrátového signálu Bluetooth. Součástí měření bylo také zjištění elektrických parametrů, uvedených v tabulce níže.

### 4.8.1 Dosah bezdrátového signálu

Měření bylo prováděno v rovné chodbě budovy, cca 15 m dlouhé. Nejprve bylo navázáno Bluetooth spojení se senzorem prostřednictvím mobilního telefonu Samsung S10e a následně byl senzor uveden do režimu `measureMod::printSamples`. Senzor byl položen na podlahu a osoba držící mobilní telefon, se od něj začala postupně vzdalovat. Místo kde došlo k přerušení toku dat bylo zaznamenáno. Zjistilo se, že dosah ve volném prostoru je cca 6 m.

### 4.8.2 Odběr elektrické energie

Bylo provedeno měření odběru elektrické energie v režimu spánku a v měřících režimech `MeasureMode::printSamples` a `MeasureMode::writeSamples`. Zapojení dle Obrázek 50: Oživení senzoru-elektrické zapojení

Operativní mód	Odběr [mA]
Normální režim(Bluetooth připojen)	16
Sleep Mode.	1
<code>MeasureMode::printSamples.</code>	16.7
<code>MeasureMode::writeSamples.</code>	8.7

Tabulka 8: Naměřené hodnoty elektrického odběru senzoru.

Dále bylo změřeno, že při elektrickém napětí baterie nižší jak 2.75 V již senzor není stabilní a zasekává se a je tudíž nepoužitelný.

### 4.8.3 Data ze senzoru

Data ze senzoru byla čtena pomocí programu Serial Bluetooth Terminal 1.41 [26]. Tento program byl nainstalován na mobilním telefonu Samsung S10e. Pomocí programu, který představuje terminál se mobilní telefon připojil k senzoru a odtud se textově ovládal. Měření

se spustilo pomocí terminálu odesláním čísla 4. Na obrazovku se začaly vypisovat jednotlivé vzorky, které se po ukončení měření číslicí 5 odeslaly v textové podobě k dalšímu zpracování. Měření byla prováděna 3x tak, aby se vyloučila nahodilost experimentu. Pro zpracování dat byl použit skript v jazyce Python [27]. Tento skript byl upraven pro potřeby tohoto projektu. Skript nejdříve očistí vstupní data od neúplných či nesprávných záznamů. V dalším kroku vykreslí graf závislosti zrychlení v  $m/s^2$  na čase v ms pro všechny tři osy. Poté je odstraněn šum pomocí frekvenční filtrace signálu. Nakonec pomocí dvojité integrace dat, kdy nejprve integrujeme podle  $dx=1/50$  (Samplovací frekvence senzoru) akceleraci tak, abychom dostali rychlost, a následně integrujeme rychlost, z které získáme dráhu a tím i souřadnice bodu v prostoru. Tyto body pak použijeme jako vstupní argument funkce pro sestavení 3D grafu tak, aby byla názorně vidět trajektorie senzoru. Senzor byl nastaven následovně:

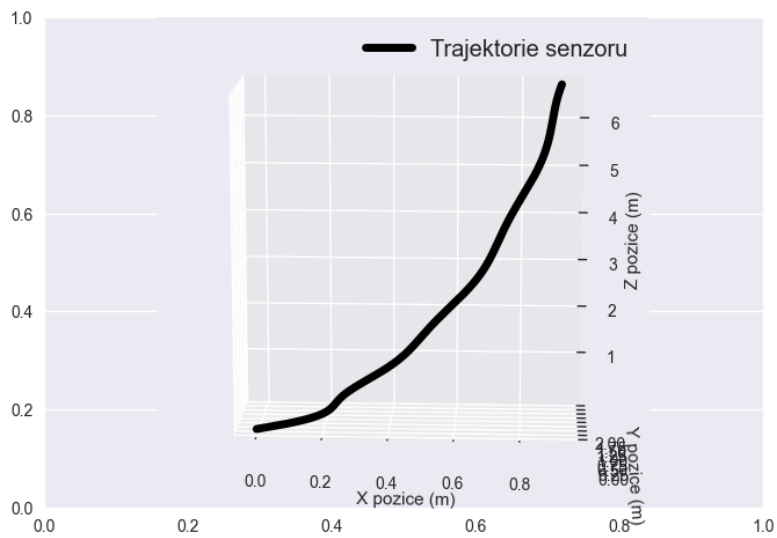
1. Frekvence vzorkování 50 Hz.
2. DLFP 6.
3. Mód měření MeasureMod:: printSamples.

#### 4.8.3.1 Oscilace

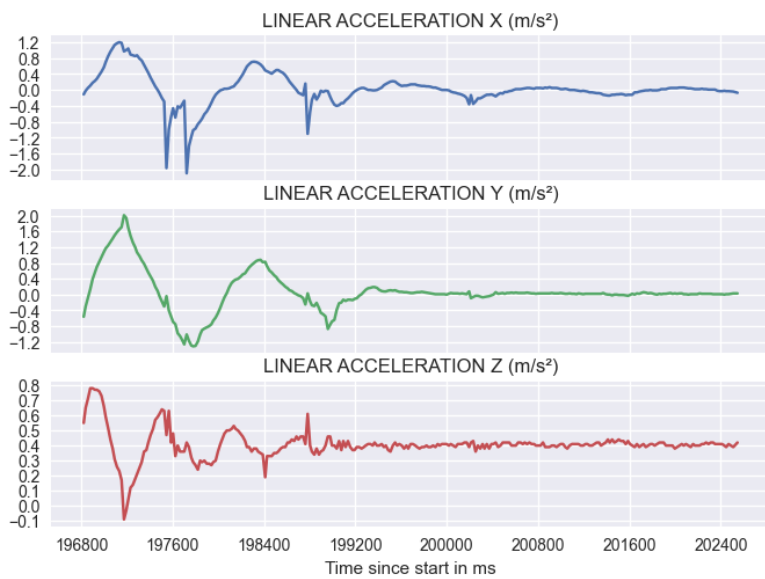
Senzor umožňuje nastavit stupeň filtrace signálu pomocí DLFP, který je popsán na Obrázek 67: DLPF-konfigurace. Předmětem zájmu bylo zjistit, zdali je rozdíl mezi nejmenší a nejvyšší filtrací signálu. První měření tak proběhlo s nastavením DLFP=1. Senzor byl umístěn v pouzdře, které bylo zavěšené na provázku cca 500 mm dlouhém. Bylo spuštěno měření a senzor byl umístěn do pozice tak, aby provázek tvořil rovnoběžku s vodorovnou plochou. Senzor byl spuštěn ve směru gravitace a začal oscilovat zleva doprava. Měření bylo ukončeno v momentě, kdy se senzor ustálil ve stabilní poloze.

### 4.8.3.1.1 DLFP1

### 3D Trajektorie



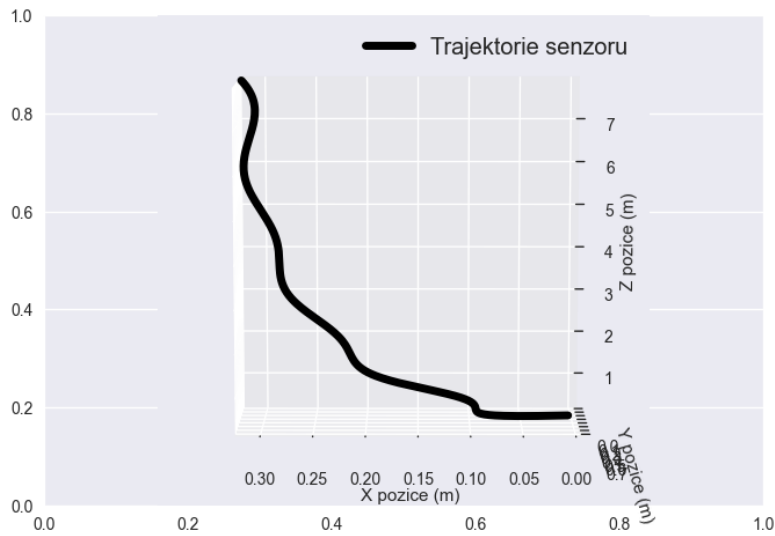
Obrázek 53: Měření-Vizualizace-Oscilace-DLFP=1.



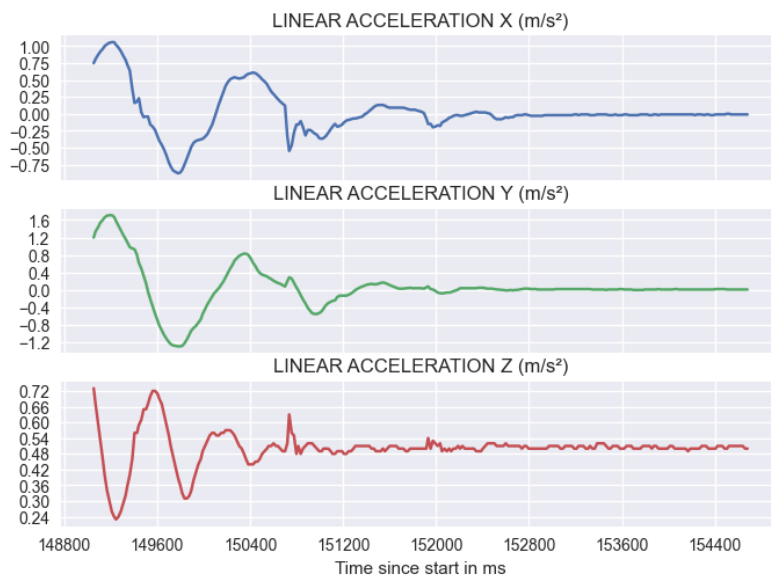
Obrázek 54: Měření-Graf-Oscilace-DLFP=1.

#### 4.8.3.1.2 DLFP6

### 3D Trajektorie



Obrázek 55: Měření-Vizualizace-Oscilace –DLFP=6.



Obrázek 56: Měření-Graf-Oscilace-DLFP=6.

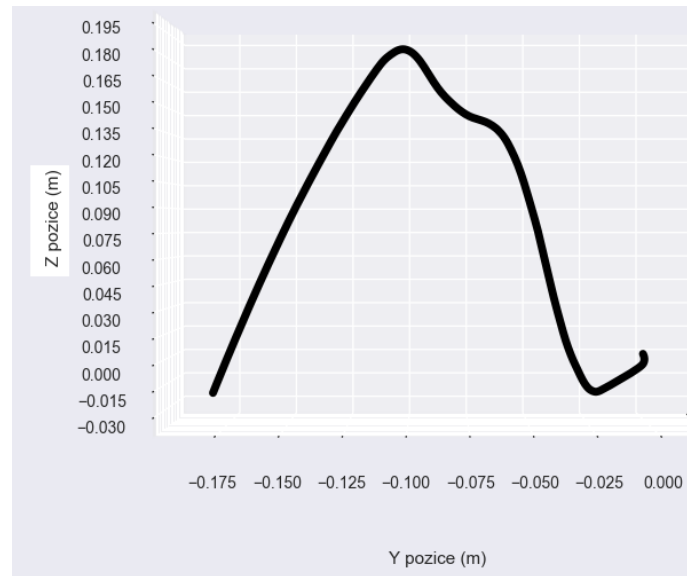
#### 4.8.3.1.3 Výsledky měření

Data z grafů ukázala, že senzor je schopen zachytit oscilaci způsobenou gravitací a volným pádem. Vizualizace toto potvrzuje vykreslením přibližné dráhy senzoru. Ukázalo se, že senzor při DLFP=1 zachycuje více šumu a signál tak není vyhlazený na rozdíl od nastavení s DLFP=6. Všechny další měření proto budou prováděna s tímto nastavením.

#### 4.8.3.2 Zdvih

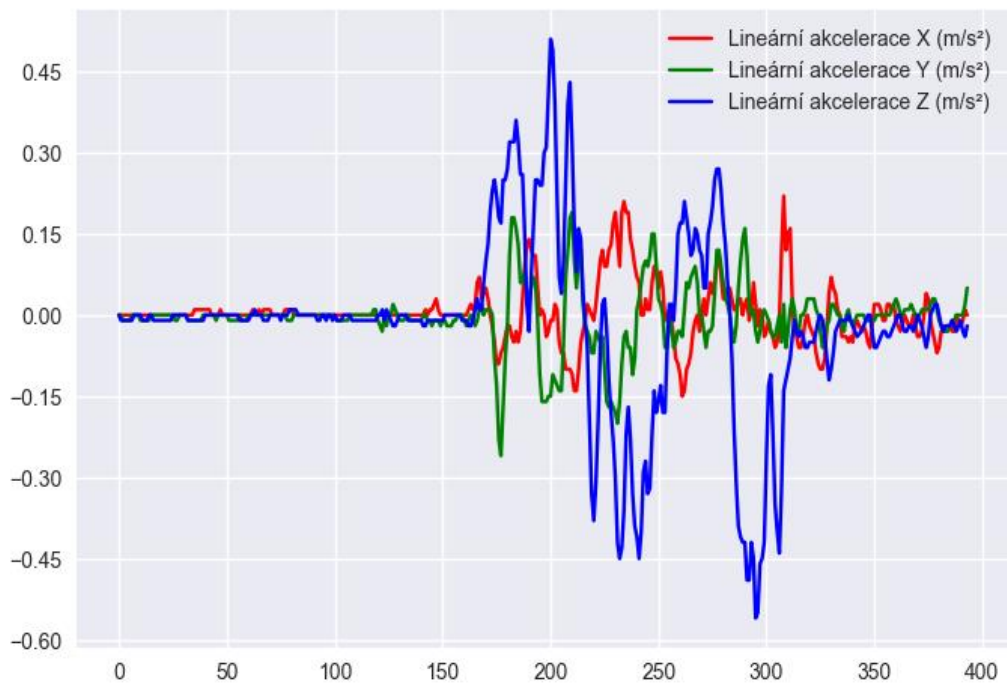
Senzor byl zdvižen z vodorovné podložky ve směru osy senzoru Z, do výšky cca 0.30 m a poté umístěn zpět.

##### 4.8.3.2.1 Výsledky měření



Obrázek 57: Měření-Vizualizace-Zdvih.

Po odečtení hodnot z grafu výše je vidět, že vzdálenost na ose Z je cca téměř 0.21 m, což neodpovídá senzorem vykonané dráze.



Obrázek 58: Měření-Graf-Zdvih.

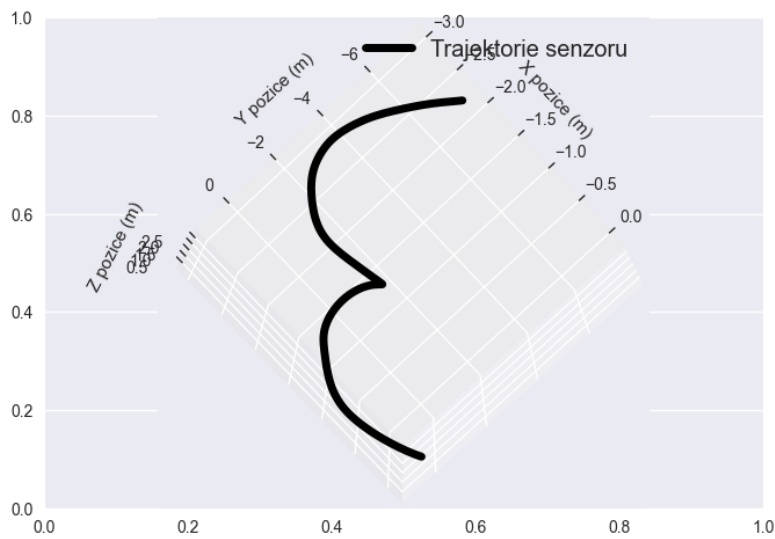
Z grafu je patrné, že senzor při pohybu vykazuje značný drift způsobený šumem v signálu. Převážně v osách X a Y by signál téměř neměl být. Toto se pak projeví i ve vizualizaci trajektorie.

#### 4.8.3.3 Písmeno B

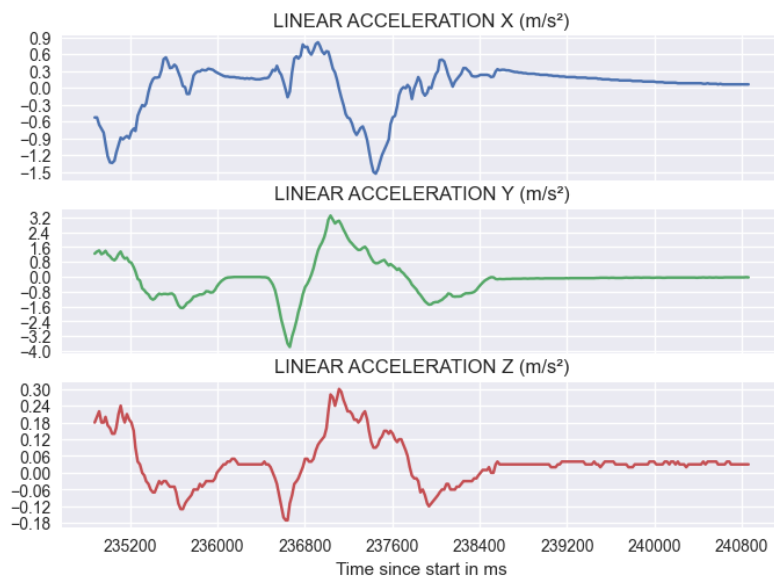
Senzor byl umístěn na vodorovné podložce a byla s ním opsána trajektorie, podobající se písmenu B.

##### 4.8.3.3.1 Výsledky měření

### 3D Trajektorie



Obrázek 59: Měření-Vizualizace-B.



Obrázek 60: Měření-Graf-B.

Graf opět vykazuje šum ve všech osách. Z vizualizace je patrné, že tvar písmene B je zobrazen.

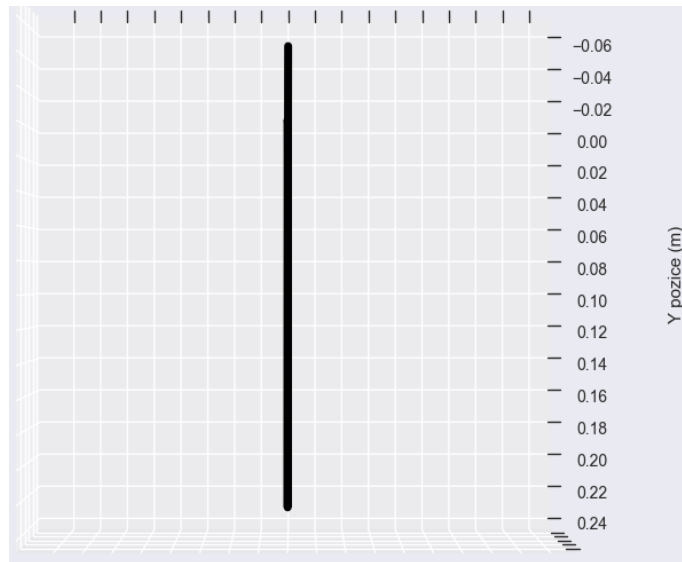
#### 4.8.3.4 Pohyb po stole

Senzor byl položen na vodorovnou rovinu, kde byla odměřena dráha 30 cm, po které se senzor pohyboval pomocí vnější síly v ose senzoru Y. Senzor byl posunován rukou ve směru osy Y, až bylo dosaženo konce dráhy. Měření bylo odesláno ke zpracování v python skriptu.



Obrázek 61: Pohyb po stole.

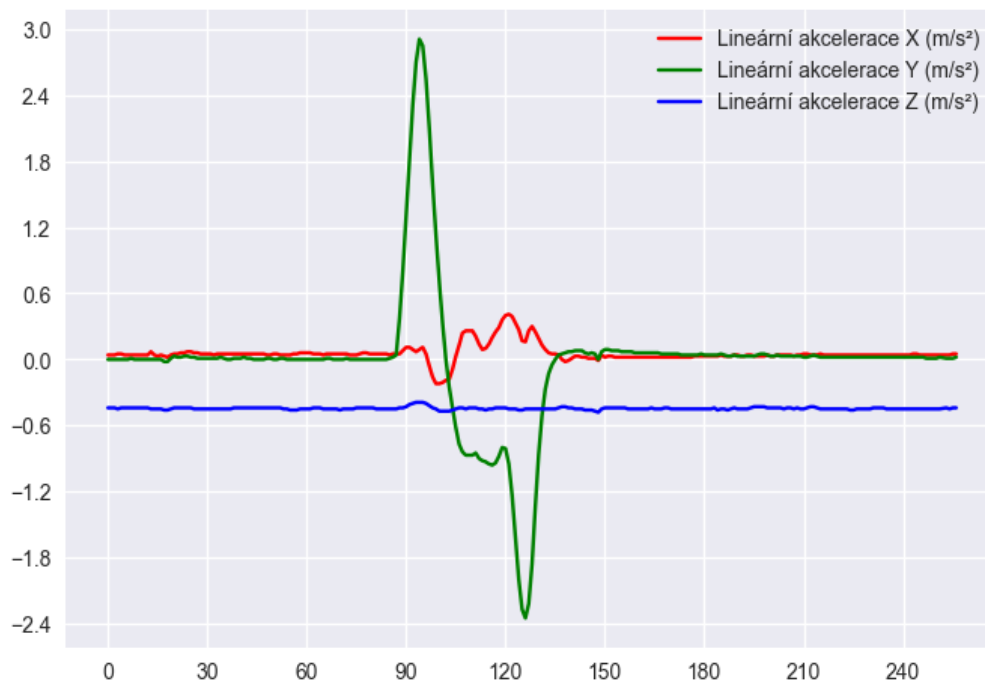
#### 4.8.3.4.1 Výsledky měření



Obrázek 62: Měření-Vizualizace-Pohyb po stole-trajektorie.

Po odečtení hodnot z grafu výše je vidět, že vzdálenost na ose Y je téměř 0.30 m, což je hodnota dráhy, po které byl senzor veden.





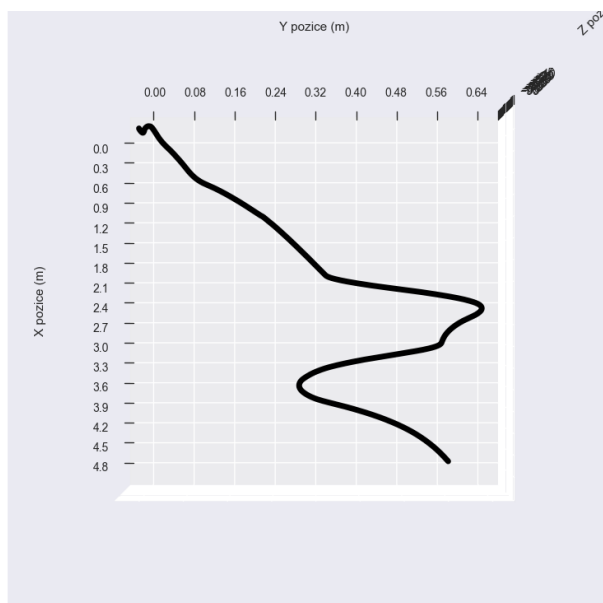
Obrázek 63: Měření-Graf-Pohyb po stole.

Na grafu výše je názorně vidět zachycení pohybu senzoru ve směru osy Y se zrychlením  $3 \text{ m/s}^2$  a následné brždění.

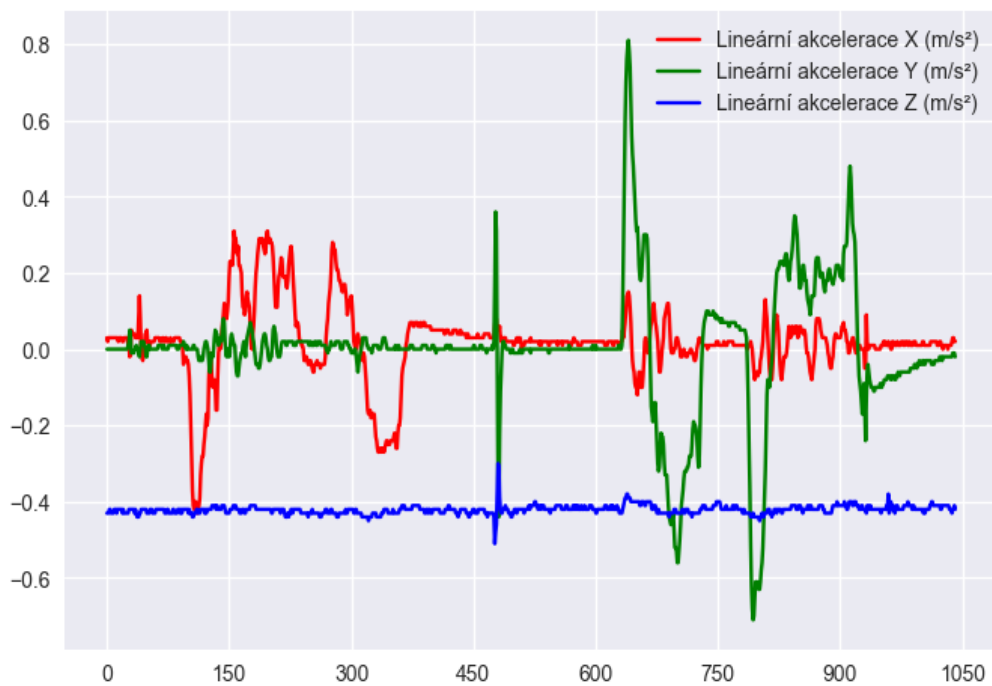
#### 4.8.3.5 Písmeno X

Senzor byl umístěn na vodorovnou podložku a bylo jím posunováno nejprve v ose X ve vzdálenosti 0.30 m a poté zpět do vzdálenosti 0.15 m a následně pak v ose Y od 0.15 v jednom směru a poté zpět o 0.30 m.

##### 4.8.3.5.1 Výsledky měření



Obrázek 64: Měření-Vizualizace-X.



Obrázek 65: Měření-Graf-X.

Z vizualizace je patrné, že obrázek jen stěží připomíná písmeno X. Na grafu je ale patrné, že pohyb byl zaznamenán správně.

#### 4.8.3.6 Závěr z měření

Z měření je patrné, že senzor zachycuje signály akcelerace ve všech třech osách, ale se značným šumem. Některé obrazce nekorespondují se skutečnou dráhou pohybu senzoru, ale to může být způsobeno nepřesným zpracováním dat na straně python skriptu, protože při pohledu na grafovou část se zachyceným zrychlením senzoru ve všech třech osách se tyto zrychlení jeví jako skutečně kopírující pohyb senzoru. Problém by nejspíše vyřešilo zavedení některých známých filtrů jako například Kalmanův filtr.

## 5 Základní parametry senzoru

Parametr	Popis
Takt mikroprocesoru.	4 Mhz
Maximální samplovací frekvence.	50 Hz
Maximální odběr	16.7 mA
Dosah Bluetooth	6 m
Rozměry elektronické části (V*Š*D)	21*32*32 mm
Průměr pouzdra.	42.6 mm
Baterie	CR2032
Výdrž na jednu baterii při zapnutém měření.	80 minut

Tabulka 9: Základní parametry senzoru.

## 6 Výsledky a diskuse

Cílem této práce bylo navrhnout a sestavit senzor schopný měřit akceleraci ve všech třech osách za pomoci gyroskopu a akcelerometru. Práce na projektu započaly výběrem vhodných komponent a sestavením prototypů. Tyto prototypy sloužily zejména k seznámení se s konkrétním typem mikroprocesoru a zjištěním jeho vlastností. Z těchto prototypů byla vybrána jedna varianta na základě vícekritériální analýzy, kde byl kladen důraz na rozměry a elektrickou spotřebu. Na vybraném prototypu probíhal vývoj software, kde byl neustálý boj o místo v paměti mikroprocesoru a muselo se tak programovat efektivně aby místo nedošlo. Také se musel vyvážit poměr mezi výpočetním výkonem a elektrickou spotřebou v momentě, kdy se nastavoval takt mikroprocesoru. Nakonec se podařilo vytvořit programovou část efektivně, za pomoci systému tříd a referencí. Následoval návrh plošného spoje tak, aby zařízení bylo co nejmenší. Byla navržena struktura složená ze tří nad sebou umístěných PCB eliptického tvaru tak, aby se vešly do kulového pouzdra. Osazení PCB probíhalo za pomoci horkovzdušné

stanice, kde byla použita pájecí pasta, jinak by nebylo možné napájet miniaturní CMOS na PCB. Tato práce vyžadovala i tak značnou trpělivost a pevnou ruku. Vytvořený senzor bylo potřeba umístit do kulovitého pouzdra. Byly vymodelovány v 3D programu dvě polokoule, které se do sebe zaklesly a uzamkly západkovým mechanismem. Vyrobený senzor byl umístěn do vytvořeného pouzdra a provedeno měření. Měření ukázalo, že senzor poskytuje korektní data, ale se značným šumem.

Senzor by se dal vylepšit například přidáním barometru, aby se eliminoval šum způsobený kmitáním seismické hmoty. Tento senzor má také značnou spotřebu elektrické energie vzhledem k velikosti baterie CR2032 a bylo dobré zvážit použití například nabíjecí baterie, která by možná nebyla tak náchylná na odebíraný elektrický proud. Tím pádem by se mohl zvýšit i takt mikroprocesoru a tím samplovací frekvence. Co se týče kulového pouzdra, tak zde by bylo potřeba přidat vůle v některých částech zejména pak tam, kde se jednotlivé polokoule dotýkají a redesignovat vnitřní část pouzdra, která drží senzor na místě. Koncept složený z mikroprocesoru Atmega328 a Bluetooth HM-11 se ukázal jako správná cesta, nicméně je potřeba aplikovat určitá vylepšení tak, aby byla na výstupu poskytována optimální data.

## **7 Seznam použitých zdrojů**

- [1] M. Bao, Analysis and Design Principles of MEMS Devices, Elsevier Science, 2005, p. 334.
- [2] J. Fraden, Handbook of Modern Sensors, 2015, p. 758.
- [3] S. Nihtianov, Smart Sensors and MEMS: Intelligent Sensing Devices and Microsystems for Industrial Applications, Woodhead Publishing; 2nd edition, 2018, p. 604.
- [4] K. Townsend, Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking, O'Reilly Media, 2014, p. 282.
- [5] R. Wolfgang, Smart Card Handbook, Paperbackshop Uk Import, 2010, p. 1088.
- [6] docs.arduino.cc, „docs.arduino.cc,“ 11 10 2022. [Online]. Available: <https://docs.arduino.cc/>.

- [7] laskakit.cz, „Atmega323/P datasheet,“ 11 10 2022. [Online]. Available: [https://www.laskakit.cz/user/related\\_files/atmel-42735-8-bit-avr-microcontroller-atmega328-328p\\_datasheet.pdf](https://www.laskakit.cz/user/related_files/atmel-42735-8-bit-avr-microcontroller-atmega328-328p_datasheet.pdf).
- [8] docs.arduino.cc, „Memory Map,“ 11 10 2022. [Online]. Available: [https://docs.arduino.cc/static/31227bd79d08441e6b11a63aa61444c8/80e3c/Memory\\_Map.jpg](https://docs.arduino.cc/static/31227bd79d08441e6b11a63aa61444c8/80e3c/Memory_Map.jpg).
- [9] arduino.cc, „PinMapping168,“ 8 12 2022. [Online]. Available: <https://docs.arduino.cc/hacking/hardware/PinMapping168>.
- [10] espressif, „ESP8266,“ 10 11 2022. [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf).
- [11] winbond.com, „spi flash w25Q64,“ 10 11 2022. [Online]. Available: [https://www.winbond.com/hq/product/code-storage-flash-memory/serial-nor-flash/?\\_\\_locale=en&partNo=W25Q64FV](https://www.winbond.com/hq/product/code-storage-flash-memory/serial-nor-flash/?__locale=en&partNo=W25Q64FV).
- [12] wikipedia, „list of battery sizes,“ 10 11 2022. [Online]. Available: [https://en.wikipedia.org/wiki/List\\_of\\_battery\\_sizes](https://en.wikipedia.org/wiki/List_of_battery_sizes).
- [13] GME, „GME-Držák CR2032,“ 1 1 2023. [Online]. Available: [https://www.gme.cz/data/product/480\\_480/819-125.1.jpg?ts=1503678829](https://www.gme.cz/data/product/480_480/819-125.1.jpg?ts=1503678829).
- [14] GME, „Držák Lily Pad CR2032,“ 1 1 2023. [Online]. Available: [https://www.gme.cz/data/product/480\\_480/pctdetail.772-302.1.jpg?ts=1517394030](https://www.gme.cz/data/product/480_480/pctdetail.772-302.1.jpg?ts=1517394030).
- [15] RS-PRO, „Datasheet RS Pro Lithium-ion button battery,“ 10 11 2022. [Online]. Available: <https://docs.rs-online.com/2a3b/0900766b8170b06f.pdf>.
- [16] www.laskakit.cz, „Li-ion CR2450,“ 1 1 2023. [Online]. Available: [https://www.laskakit.cz/user/shop/big/4256\\_3-6v-li-ion-watch-batteries-lir2450.jpg?6137b46c](https://www.laskakit.cz/user/shop/big/4256_3-6v-li-ion-watch-batteries-lir2450.jpg?6137b46c).
- [17] GME, „Držák CR2450,“ 1 1 2023. [Online]. Available: [https://www.gme.cz/data/product/480\\_480/pctdetail.819-044.1.jpg?ts=1503704969](https://www.gme.cz/data/product/480_480/pctdetail.819-044.1.jpg?ts=1503704969).
- [18] invensense, „mpu6050,“ 10 11 2022. [Online]. Available: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [19] dratek.cz, „MPU6050,“ 1 1 2023. [Online]. Available: <https://dratek.cz/photos/produkty/d/0/830.jpg?m=1502871179>.

- [20] ai-thinker, „docs.ai-thinker.com,“ 7 12 2022. [Online]. Available: [https://docs.ai-thinker.com/\\_media/esp8266/docs/esp-12f\\_product\\_specification\\_en.pdf](https://docs.ai-thinker.com/_media/esp8266/docs/esp-12f_product_specification_en.pdf).
- [21] docs.arduino.cc, „ArduinoISP,“ 1 1 2023. [Online]. Available: <https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoISP>.
- [22] pccompci.com/Baud\_Rate, „Baud\_Rate,“ 1 1 2023. [Online]. Available: [https://www.pccompci.com/Baud\\_Rate.html](https://www.pccompci.com/Baud_Rate.html).
- [23] Banggood, „HM-11 Bluetooth,“ 1 1 2023. [Online]. Available: [https://img.banggood.com/file/products/20150104013200BLE-CC41-A\\_AT%20Command.pdf](https://img.banggood.com/file/products/20150104013200BLE-CC41-A_AT%20Command.pdf).
- [24] i2cdevlib, „i2cdevlib,“ 1 1 2023. [Online]. Available: [https://www.i2cdevlib.com/docs/html/class\\_m\\_p\\_u6050.html](https://www.i2cdevlib.com/docs/html/class_m_p_u6050.html).
- [25] Jlcpcb, „Jlcpcb,“ Jlcpcb, 1 1 2023. [Online]. Available: <https://jlcpcb.com/>. [Přístup získán 1].
- [26] kai-morich, „SimpleBluetoothTerminal,“ SimpleBluetoothTerminal, 1 1 2023. [Online]. Available: <https://github.com/kai-morich/SimpleBluetoothTerminal>.
- [27] M. Raymond, „Exploring-Data-Acquisition-with-Android-Devices-and-Python,“ 1 1 2023. [Online]. Available: <https://github.com/mack-raymond/Exploring-Data-Acquisition-with-Android-Devices-and-Python>.
- [28] V. K. V. O. O. A. Julian W. Gardner, Microsensors, MEMS, and Smart Devices, WILEY INDIA, 2001, p. 524.

## 8 Seznam obrázků, tabulek, grafů a zkratk

### 8.1 Seznam obrázků

Obrázek 1: Koncept akcelerometru [2].	18
Obrázek 2: Piezorezistivní akcelerometr [1].	19
Obrázek 3: Kapacitní akcelerometr [1].	20
Obrázek 4: Kapacitní akcelerometr, pružné lamely [1].	20
Obrázek 5: Vektor Coriolovy akcelerace [2].	21
Obrázek 6: Konceptuální diagram Gyroskopu [2].	22
Obrázek 7: Verze Bluetooth [4].	22
Obrázek 8: Bluetooth Observer koncept [4].	24
Obrázek 9: Connected topologie [4].	24
Obrázek 10: Rozdělení paměti [5].	25
Obrázek 11: HM-11 Bluetooth.	28
Obrázek 12: Rozložení adresního prostoru Atmega [8].	30
Obrázek 13: ATmega328.	30
Obrázek 14: Schéma-Arduino Pin Mapping [9].	31
Obrázek 15: Schéma ESP8266 [10].	32
Obrázek 16: ESP8266.	32
Obrázek 17: Paměť SPI W25Q64.	33
Obrázek 18: Držák pro CR2032 –plech [13].	34
Obrázek 19: Držák pro CR2032 -arduino Lily Pad [14].	35
Obrázek 20: Baterie CR2450 Li-ion [16].	35
Obrázek 21: Držák baterie CR2450-plastový [17].	36
Obrázek 22: Schéma MPU6050 [18].	37
Obrázek 23: MPU6050 [19].	37
Obrázek 24: Elektrické schéma zapojení-VA CR2032.	38
Obrázek 25: Porovnání VA charakteristik CR2032 VS CR2450.	40
Obrázek 26: Elektrické schéma vývojové kitu ESP8266-12F.	41
Obrázek 27: Vyrobený vývojový kit ESP8266.	42
Obrázek 28: Laboratorní zapojení prototypu ES8266+CR2032.	42

Obrázek 29: Zapojení ESP8266-12F+ CR2032+MPU6050.....	44
Obrázek 30: Zapojení Atmega328 + CR2032 [21].....	45
Obrázek 31: Schéma-Bluetooth vývojový kit.....	46
Obrázek 32: Bluetooth modul - seznam příkazů.....	47
Obrázek 33: Schéma-Atmega+CR2032+Bluetooth+SPI FLASH.....	49
Obrázek 34: Arduino sketch-ukázka kódu.....	52
Obrázek 35: Diagram MainLoop.....	53
Obrázek 36: Diagram ProcessCommands.....	55
Obrázek 37: Diagram ProcessMeasuring.....	55
Obrázek 38: Diagram SleepMode.....	55
Obrázek 39: Diagram MpuInit.....	56
Obrázek 40: MPU6050 a jeho osy.....	58
Obrázek 41: Schéma vlastního senzoru.....	59
Obrázek 42: PCB-zapojení pinů.....	59
Obrázek 43: PCB-držák baterie.....	61
Obrázek 44: PCB-mikroprocesorová část.....	61
Obrázek 45: Pájecí pasta.....	61
Obrázek 46: PCB-osazení- Bluetooth modul.....	62
Obrázek 47: PCB-osazení-pohled z boku.....	62
Obrázek 48: PCB-osazení-pohled shora.....	62
Obrázek 49: PCB-osazení-pohled-zespodu.....	62
Obrázek 50: Oživení senzoru-elektrické zapojení měřícího obvodu.....	63
Obrázek 51: 3D model-půdorys.....	64
Obrázek 52: Vytisknutý 3D model.....	64
Obrázek 53: Měření-Vizualizace-Oscilace-DLFP=1.....	67
Obrázek 54: Měření-Graf-Oscilace-DLFP=1.....	67
Obrázek 55: Měření-Vizualizace-Oscilace –DLFP=6.....	68
Obrázek 56: Měření-Graf-Oscilace-DLFP=6.....	68
Obrázek 57: Měření-Vizualizace-Zdvih.....	69
Obrázek 58: Měření-Graf-Zdvih.....	70
Obrázek 59: Měření-Vizualizace-B.....	71
Obrázek 60: Měření-Graf-B.....	71
Obrázek 61: Pohyb po stole.....	72



Obrázek 62: Měření-Vizualizace-Pohyb po stole-trajektorie. ....	72
Obrázek 63: Měření-Graf-Pohyb po stole.....	73
Obrázek 64: Měření-Vizualizace-X.....	74
Obrázek 65: Měření-Graf-X. ....	74
Obrázek 66: Sběr dat pomocí mobilního telefonu. ....	83
Obrázek 67: DLPF-konfigurace.....	86

## 8.2 Seznam tabulek

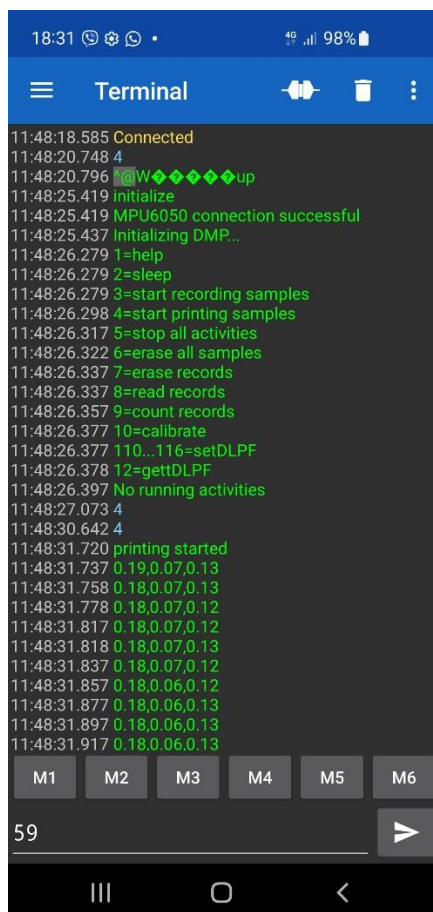
Tabulka 1: VA charakteristika CR2032-Naměřené hodnoty.....	39
Tabulka 2: VA charakteristika CR2450-Naměřené hodnoty.....	39
Tabulka 3: Nastavení UART rychlosti Bluetooth modulu. ....	48
Tabulka 4: Porovnání prototypů. ....	51
Tabulka 5: Knihovny třetích stran. ....	52
Tabulka 6: Nastavené hodnoty některých konstant a proměnných [24].....	56
Tabulka 7: Seznam použitých komponent. ....	60
Tabulka 8: Naměřené hodnoty elektrického odběru senzoru. ....	65
Tabulka 9: Základní parametry senzoru. ....	75
Tabulka 10: WBT-proměnné. ....	83
Tabulka 11: WBT-metody. ....	84
Tabulka 12: WFLASH-proměnné.....	84
Tabulka 13: WFLASH-metody.....	84
Tabulka 14: WMPU-Proměnné. ....	85
Tabulka 15: WMPU-metody.....	86
Tabulka 16: WEEPROM-proměnné. ....	86
Tabulka 17: WEEPROM-metody. ....	87
Tabulka 18: WMAINFCE-proměnné. ....	87
Tabulka 19: WMAINFCE-metody. ....	88
Tabulka 20: SampleConfig-proměnné.....	88
Tabulka 21: SampleConfig-metody.....	88
Tabulka 22: Sample-proměnné. ....	88
Tabulka 23: Constants-konstanty.....	88

### 8.3 Seznam použitých zkratek

I2C	-Inter-Integrated Circuit
MEMS	-Microelectromechanical system
UART	-Universal asynchronous receiver / transmitter
Wi-Fi	-Wireless Fidelity
BLE	-Bluetooth Low Energy
MAC	-Media Access Contro
TCP/IP	-Transmission Control Protocol/Internet Protocol
GATT	-Generic Attribute Profile
GAP	-Generic Access Profile
API	-Application programming interface
USB	-Universal Serial Bus
HCI	-Host Controller Interface
EEPROM	-Electrically erasable programmable read-only memory
SMD	-Surface-mount technology
HW	-Hardware
SPI	-Serial Peripheral Interface
PCB	-Printed Circuit Board
FIFO	-First In, First Out
DMP	-Digital Motion Processor
Li-ion	-Lithium-ion
VA	-Voltampérová charakteristika
LED	-Light-emitting diode
BOD	-Brown-out detection
DLPF	-Digital low-pass filter
CMOS	-Complementary Metal–Oxide–Semiconductor

## 9 Přílohy

### 9.1 Ukázka sběru dat pomocí mobilního telefonu.



```
18:31 98%
Terminal
11:48:18.585 Connected
11:48:20.748 4
11:48:20.796 #@W up
11:48:25.419 initialize
11:48:25.419 MPU6050 connection successful
11:48:25.437 Initializing DMP...
11:48:26.279 1=help
11:48:26.279 2=sleep
11:48:26.279 3=start recording samples
11:48:26.298 4=start printing samples
11:48:26.317 5=stop all activities
11:48:26.322 6=erase all samples
11:48:26.337 7=erase records
11:48:26.337 8=read records
11:48:26.357 9=count records
11:48:26.377 10=calibrate
11:48:26.377 110...116=setDLPF
11:48:26.378 12=gettDLPF
11:48:26.397 No running activities
11:48:27.073 4
11:48:30.642 4
11:48:31.720 printing started
11:48:31.737 0.19,0.07,0.13
11:48:31.758 0.18,0.07,0.13
11:48:31.778 0.18,0.07,0.12
11:48:31.817 0.18,0.07,0.12
11:48:31.818 0.18,0.07,0.13
11:48:31.837 0.18,0.07,0.12
11:48:31.857 0.18,0.06,0.12
11:48:31.877 0.18,0.06,0.13
11:48:31.897 0.18,0.06,0.13
11:48:31.917 0.18,0.06,0.13
M1 M2 M3 M4 M5 M6
59
```

Obrázek 66: Sběr dat pomocí mobilního telefonu.

### 9.2 Popis tříd

#### 9.2.1 WBT s interfacem IBT

Slouží jako rozhraní mezi knihovnou AltSoftSerial.h, která zajišťuje komunikaci mezi Atmega328 a Bluetooth modulem.

Název proměnné	Datový typ	Popis
Bluetooth	AltSoftSerial	Knihovna, která simuluje UART a ten používá ke komunikaci s Bluetooth.
wakeUpPin	byte	Arduino pin, kterým se probouzí Bluetooth z režimu spánku.
btConfigMode	uint8_t	Indikuje, zdali se Bluetooth nachází v konfiguračním modu.

Tabulka 10: WBT-proměnné.

Název metody	Návratová hodnota	Popis
configRead()		Čte příchozí data z Bluetooth.
configWrite()		Zapíše data ze Seriál na UART Bluetooth.
configInit()		Inicializuje konfigurační mód. Volá nápovědu Bluetooth příkazem AT+HELP na UART a nastavuje proměnou btConfigMode=1.
sleep()		Uspává Bluetooth zasláním příkazu AT+SLEEP na UART Bluetooth.
init()		Konfiguruje pro použití s Bluetooth a inicializuje Bluetooth.
println(const _FLASHStringHelper *)		Převodní metoda pro AltsoftSerial.h.
print(const _FLASHStringHelper *)		Převodní metoda pro AltsoftSerial.h.
print(float)		Převodní metoda pro AltsoftSerial.h.
println(float)		Převodní metoda pro AltsoftSerial.h.
getConfigMode()	uint8_t(1 pokud se program nachází v ConfigModu 0 pokud ne.)	Vrací hodnotu proměnné btConfigMode.
available()	int	Vrací počet bytů v UART registru Bluetooth.
readBytes(char *, size_t)	int	Čte data určené délky z UART registru do místa v paměti.

Tabulka 11: WBT-metody.

## 9.2.2 WFLASH s interfacem IFLASH

Komunikuje s FLASH pamětí a zajišťuje základní funkce jako zápis a čtení.

Název proměnné	Datový typ	Popis
flash	SPIFlash	Instance knihovny SPIMemory.h.
bt	IBT	
mainFce	IMAINFCE	

Tabulka 12: WFLASH-proměnné.

Název metody	Návratová hodnota	Popis
init()		Inicializace FLASH paměti pomocí SPIMemory.h.
writeSample(Sample &)		Získá volnou adresu v paměťovém prostoru FLASH a zapíše vzorek Sample. Pokud je FLASH zcela zaplněna, přestane se zápis provádět.
readSamples()		Čte vzorky Sample z FLASH paměti do té doby, než v proměnné timeStamp patří do třídy Sample najde hodnotou NULL. Vzorky vypisuje pomocí fce mainFce.printSample.
countSamples()	Int (počet vzorků)	Načítá ze vzorku Sample pouze hodnotu timeStamp a končí, když je hodnota NULL.
eraseSamples()		Spočítá vzorky a vymaže z FLASH paměti pouze sekce, které obsahují data vzorků Sample.
eraseEntireChip()		Vymaže celý obsah paměti FLASH(velký proudový odběr).
WFLASH(IBT &, IMAINFCE &)		Konstruktor akceptující instanci WBT a WMAINFCE.

Tabulka 13: WFLASH-metody.

## 9.2.3 WMPU s interfacem IMPU

Zprostředkovává komunikaci mezi Atmega328 a MPU6050.

Název proměnné	Datový typ	Popis
eeeprom	IEEPROM	
bt	IBT	
mpuPowerPin	byte	Arduino pin, který napájí modul MPU6050.
packetSize	uint16_t	Velikost DMP paketu v buffer MPU6050.
q	Quaternion	Quaternion.
fifoBuffer[64]	uint8_t	Buffer, kam se ukládají snímaná data ze senzorů MPU6050.
gravity	VectorFloat	Gravitační vektor. Jednotky: (g).
aa	VectorInt16	Hodnota z akcelerometru. Jednotky: (bezrozměrné), 1g=8192 jednotek.
aaReal	VectorInt16	Hodnota z akcelerometru, ale bez gravitace. Jednotky: ( $m/s^2$ ).
aaWorld	VectorInt16	Hodnota z akcelerometru, bez gravitace vynásobené rotační maticí tak, aby hodnoty odpovídali Zemskému rámci tzn. osa z vždy směřuje dolů. Jednotky: ( $m/s^2$ ).

Tabulka 14: WMPU-Proměnné.

Název metody	Návratová hodnota	Popis
WMPU(IEEPROM &, IBT &)		Konstruktor akceptuje instanci WIEEPROM a WBT.
begin()	Int(vrací hodnotu z mpu.dmpInitialize,pokud 0, tak se inicializace nepovedla,1 proběhla úspěšně ).	Zapíná MPU6050, nastavuje I2C a testuje připojení MPU6050 a nastavuje MPU6050. Inicializuje DMP procesor.
init()		Volá begin a nahrává z EEPROM kalibrační data pro MPU6050.
readFifoBuffer_(uint8_t *)		Čte FIFO buffer MPU6050, kde jsou uložena data z akcelerometru a gyroskopu a ukládá je na určené místo v paměti.
readSensorData()		Slouží pro nahrání dat z MPU6050, jejich zpracování a uložení do lokálních proměnných třídy. Volá metodu readFifoBuffer_ a metody knihovny MPU6050_6Axis_MotionApps20.h která uloží do proměnných třídy q ,aa, gravity, aaReal, aaWorld data ze senzoru MPU6050.
loadCalibration()		Nahráje kalibrační data z EEPROM.
getYawPitchRoll(float &, float &, float &)		Volá getQuaterAccelGravity a vypočte hodnoty Yaw, Pitch, Roll, které uloží na určenou adresu v paměti. Rotace Eulerovských úhlů v pořadí ZYX. Jednotky: (°).
getEarthLinearAcc(float &, float &, float &)		Volá getQuaterAccelGravity a použije hodnoty z proměnné aaReal, které dělí hodnotou 835.92 tak, aby výsledkem byl výstup v jednotkách ( $m/s^2$ ). Tyto hodnoty se uloží na určenou adresu v paměti.
calibrate()		Kalibruje MPU6050 pomocí metod z knihovny MPU6050_6Axis_MotionApps20.h a uloží offsety do EEPROM.
isCalibrated()	Bool(true je kalibrován, false není)	Vrací, zdali je MPU6050 zkalibrován.
resetCalibFlag()		Resetuje kalibrovací proměnnou.
turnOn()		Zapíná MPU6050 prostřednictvím digitálního pinu Atmega328.
turnOFF()		Vypíná MPU6050 prostřednictvím digitálního pinu Atmega328.

<b>setDLPF(int)</b>		<p>Aktivuje DLPF konfigurační filtr dle čísla a uloží ho do paměti.</p> <table border="1"> <thead> <tr> <th rowspan="2">DLPF_CFG</th> <th colspan="2">ACCELEROMETER</th> <th colspan="3">GYROSCOPE</th> </tr> <tr> <th>Bandwidth</th> <th>Delay</th> <th>Bandwidth</th> <th>Delay</th> <th>Sample Rate</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>260Hz</td> <td>0ms</td> <td>256Hz</td> <td>0.98ms</td> <td>8kHz</td> </tr> <tr> <td>1</td> <td>184Hz</td> <td>2.0ms</td> <td>188Hz</td> <td>1.9ms</td> <td>1kHz</td> </tr> <tr> <td>2</td> <td>94Hz</td> <td>3.0ms</td> <td>98Hz</td> <td>2.8ms</td> <td>1kHz</td> </tr> <tr> <td>3</td> <td>44Hz</td> <td>4.9ms</td> <td>42Hz</td> <td>4.8ms</td> <td>1kHz</td> </tr> <tr> <td>4</td> <td>21Hz</td> <td>8.5ms</td> <td>20Hz</td> <td>8.3ms</td> <td>1kHz</td> </tr> <tr> <td>5</td> <td>10Hz</td> <td>13.8ms</td> <td>10Hz</td> <td>13.4ms</td> <td>1kHz</td> </tr> <tr> <td>6</td> <td>5Hz</td> <td>19.0ms</td> <td>5Hz</td> <td>18.6ms</td> <td>1kHz</td> </tr> <tr> <td>7</td> <td>-- Reserved --</td> <td></td> <td>-- Reserved --</td> <td></td> <td>Reserved</td> </tr> </tbody> </table> <p><i>Obrázek 67: DLPF-konfigurace.</i></p>	DLPF_CFG	ACCELEROMETER		GYROSCOPE			Bandwidth	Delay	Bandwidth	Delay	Sample Rate	0	260Hz	0ms	256Hz	0.98ms	8kHz	1	184Hz	2.0ms	188Hz	1.9ms	1kHz	2	94Hz	3.0ms	98Hz	2.8ms	1kHz	3	44Hz	4.9ms	42Hz	4.8ms	1kHz	4	21Hz	8.5ms	20Hz	8.3ms	1kHz	5	10Hz	13.8ms	10Hz	13.4ms	1kHz	6	5Hz	19.0ms	5Hz	18.6ms	1kHz	7	-- Reserved --		-- Reserved --		Reserved
DLPF_CFG	ACCELEROMETER			GYROSCOPE																																																									
	Bandwidth	Delay	Bandwidth	Delay	Sample Rate																																																								
0	260Hz	0ms	256Hz	0.98ms	8kHz																																																								
1	184Hz	2.0ms	188Hz	1.9ms	1kHz																																																								
2	94Hz	3.0ms	98Hz	2.8ms	1kHz																																																								
3	44Hz	4.9ms	42Hz	4.8ms	1kHz																																																								
4	21Hz	8.5ms	20Hz	8.3ms	1kHz																																																								
5	10Hz	13.8ms	10Hz	13.4ms	1kHz																																																								
6	5Hz	19.0ms	5Hz	18.6ms	1kHz																																																								
7	-- Reserved --		-- Reserved --		Reserved																																																								
<b>getDLPF()</b>		Vrátí číslo aktuálního nastavení DLPF filtru z registru MPU6050.																																																											

Tabulka 15: WMPU-metody.

## 9.2.4 WEEPROM s interfacem IEEPROM

Zprostředkovává komunikaci s interní EEPROM pamětí Atmega328.

Název proměnné	Datový typ	Popis
<b>OffsetAddr</b>	enum	Určuje místo v paměti, kam se zapisují jednotlivé hodnoty. GyroX = 2, GyroY = 4, GyroZ = 6, AccelX = 8, AccelY = 10, AccelZ = 12
<b>CalibFlagAddr=0</b>		Určuje adresu kalibrační značky v paměti.
<b>CalibFlagTrue</b>		Hodnota, znamenající TRUE.
<b>CalibFlagFalse</b>		Hodnota znamenající FALSE.
<b>DLPFAddr=14</b>		Určuje adresu uložené konfigurace DLPF filtru.

Tabulka 16: WEEPROM-proměnné.

Název metody	Návratová hodnota	Popis
<b>readIntEEPROM(int)</b>	Int(hodnota uložená v paměti)	Čte int z určeného místa v paměti.
<b>writeIntEEPROM(int, int)</b>		Zapisuje int do určeného místa v paměti.
<b>getGyroOffsetX()</b>	int	Vrací offset gyroskopu X.
<b>getGyroOffsetY()</b>	int	Vrací offset gyroskopu Y.
<b>getGyroOffsetZ()</b>	int	Vrací offset gyroskopu Z.
<b>getAccelOffsetX()</b>	int	Vrací offset akcelerometru X.
<b>getAccelOffsetY()</b>	int	Vrací offset akcelerometru Y.
<b>getAccelOffsetZ()</b>	int	Vrací offset akcelerometru Z.
<b>setGyroOffsetX(int)</b>		Nastavuje offset gyroskopu X.
<b>setGyroOffsetY(int)</b>		Nastavuje offset gyroskopu Y.
<b>setGyroOffsetZ(int)</b>		Nastavuje offset gyroskopu Z.
<b>setAccelOffsetX(int)</b>		Nastavuje offset akcelerometru X.
<b>setAccelOffsetY(int)</b>		Nastavuje offset akcelerometru Y.
<b>setAccelOffsetZ(int)</b>		Nastavuje offset akcelerometru Z.
<b>getCalibFlag()</b>		Vrací kalibrační hodnotu.
<b>setCalibFlag()</b>		Nastavuje kalibrační hodnotu, pokud je nastaveno, tak metoda IsCalibrated vrací true.
<b>resetCalibFlag()</b>		Resetuje kalibrační hodnotu, pokud je nastaveno, tak metoda IsCalibrated vrací false.

setDLPFmode(int)		Uloží do paměti konfigurační číslo filtru DLPP.
getDLPFmode()	int	Vrátí konfigurační číslo filtru DLPP z paměti.

Tabulka 17: WEEPROM-metody.

## 9.2.5 WMAINFCE s interface, IMAINFCE

Tato třída využívá ostatní knihovnu a tvoří tak nejvyšší rozhraní, s kterým komunikuje Arduino sketch.

Název proměnné	Datový typ	Popis
MeasureMode	Enum(printSamples, writeSamples)	printSamples=V tomto modu se vypisují Samples na Bluetooth writeSamples=V tomto modu se Samples zapisují na FLASH
PRINTSAMPLE_VSE	definovaná konstanta int	Definuje mod vypisování metody printSample.
bt	IBT	
FLASH	IFLASH	
mpu	IMPU	
instance	IMAINFCE	
lastSensorRead	Unsigned long	
measuring	byte	Proměnná, indikující zdali je prováděno měření, neboli Aktivita. 0= žádná Aktivita není aktivní 1= některá z Aktivit je prováděna.
measureMode	MeasureMode	Proměnná, kde se ukládá aktuální prováděná Aktivita měření.
sensorReadInterval	Unsigned long	Interval v ms, kdy je přečten vzorek z MPU6050. Určuje vzorkovací frekvenci.
interruptPin	byte	Arduino pin, který slouží pro probuzení Atmega328 z režimu spánku.

Tabulka 18: WMAINFCE-proměnné.

Název metody	Návratová hodnota	Popis
printSample(Sample &)		Vypisuje na výstup hodnoty třídy Sample ve tvaru když hodnota <b>PRINTSAMPLE_VSE==0</b> "lax/lay/laz" když hodnota <b>PRINTSAMPLE_VSE==1</b> "timestamp/q.w/q.x/q.y/q.z/lax/lay/laz/ax/ay/az"
commandsRead()		Čte hodnoty bufferu UART rozhraní z Bluetooth a Seriál. Poté převádí přečtené hodnoty na int.
WMAINFCE(IBT &, IFLASH &, IMPU &)		Konstruktor přijímá instanci WBT, WFLASH a WMPU.
stop_all_activities()		Iniciuje zastavení všech aktivit uložením příznaku do proměnné measuring .
start_printing_samples()		Iniciuje vypisování vzorků Samples na Bluetooth, uložením příznaku do proměnné measuring a measureMode.
start_recording_samples()		Iniciuje ukládání vzorků Samples do FLASH paměti, uložením příznaku do proměnné measuring a measureMode. Uspává Bluetooth.
displayActivities()	__FLASHStringHelper*	Vrací název právě probíhající aktivity.
init()		Inicializuje Serial, FLASH, Bluetooth. Na základě stisku tlačítka b v konzoli Serial volá bt.configInit jinak volá sleepMode
ISRFunc()		Mapuje nestatickou funkci na statickou pro metodu arduino.h knihovny attachInterrupt.
processCommands()		Reaguje na data z UART bufferu Bluetooth a Serial tak, že volá příslušnou metodu v závislosti na odeslaném znaku.
processMeasuring()		Řídí vzorkovací frekvenci sběru dat z MPU6050 a v závislosti na MeasureMode proměnné vykonává příslušné metody.
wakeUp()		Spouští rutiny po probuzení z režimu spánku.
runFunctionsInLoop()		Metoda, která se volá v Arduino sketchy v metodě Loop.

sleepMode()		Přepnutí do úsporného režimu zařízení. Uspává Bluetooth, vypíná MPU6050 a uspává Atmega328.
help()		Zobrazí na Bluetooth seznam příkazů a probíhající aktivity.

Tabulka 19: WMAINFCE-metody.

## 9.2.6 SampleConfig

Třída pro získávání instancí jednotlivých tříd.

Název proměnné	Datový typ	Popis
bt	IBT	
eeeprom	IEEPROM	
mpu	IMPU	
mainFce	IMAINFCE	
FLASH	IFLASH	

Tabulka 20: SampleConfig-proměnné.

Název metody	Návratová hodnota	Popis
getMPU()	IMPU	Vrací instanci třídy.
getBT()	IBT	Vrací instanci třídy.
getFLASH()	IFLASH	Vrací instanci třídy.
getMain()	IMAINFCE	Vrací instanci třídy.

Tabulka 21: SampleConfig-metody.

## 9.2.7 Sample

Třída, která definuje proměnné, pro data pocházející z MPU6050.

Název proměnné	Datový typ	Popis
timeStamp	float	Čas od zapnutí Atmega328 v ms. Aktivní pouze když hodnota PRINTSAMPLE_VSE==1.
ax	float	Akcelerace v ose x Jednotky ( $m/s^2$ ). Aktivní pouze když hodnota PRINTSAMPLE_VSE==1.
ay	float	Akcelerace v ose y Jednotky ( $m/s^2$ ). Aktivní pouze když hodnota PRINTSAMPLE_VSE==1.
az	float	Akcelerace v ose z Jednotky ( $m/s^2$ ). Aktivní pouze když hodnota PRINTSAMPLE_VSE==1.
q	Quaternion	Quaternion w,x,y,z.). Aktivní pouze když hodnota PRINTSAMPLE_VSE==1.
lax	float	EarthLinearAcc Jednotky ( $m/s^2$ ),(akcelerace bez vlivu gravitace).
lay	float	EarthLinearAcc Jednotky ( $m/s^2$ ),(akcelerace bez vlivu gravitace).
laz	float	EarthLinearAcc Jednotky ( $m/s^2$ ),(akcelerace bez vlivu gravitace).

Tabulka 22: Sample-proměnné.

## 9.2.8 Constants

Obsahuje seznam definovaných konstant, pro testovací účely.

Název konstanty	Datový typ	Popis
PRINTSAMPLE_VSE	int	Možné hodnoty 1= zapnuto a 0=vypnuto. Na výstupu se objeví všechny definované hodnoty, uvedené v třídách výše.

Tabulka 23: Constants-konstanty.