

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**MONITOROVANIE A RIADENIE SIGNÁLOV Z
GYROSKOPU PROSTREDNÍCTVOM WEBOVEJ
APLIKÁCIE**

TECHNICKÁ DOKUMENTÁCIA

1 Vývojárska príručka

1.1 Popis projektu

Cieľom zadania je monitorovať resp. riadiť signály získané z reálnych senzorov resp. simulačných a virtuálnych prostredí. Monitorovanie resp. riadenie sa má uskutočňovať prostredníctvom webovej aplikácie, aby bola naplnená koncepcia IoT.

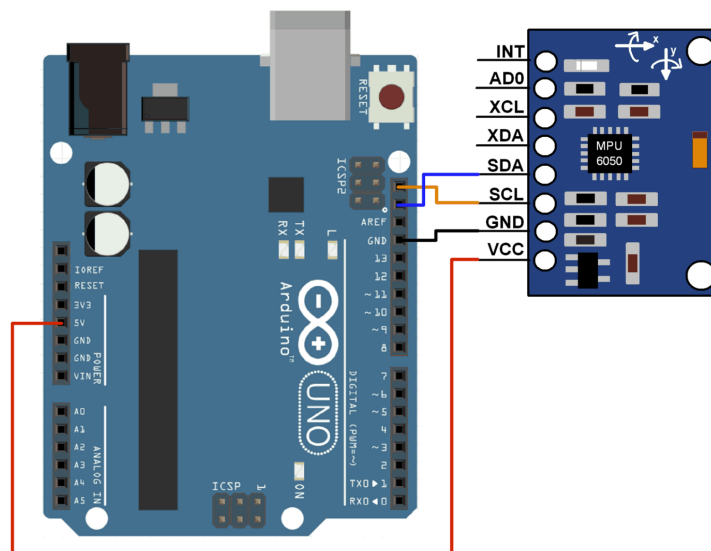
Aby sme koncepciu naplnili, zvolili sme si senzor monitorujúci uhlové zrýchlenie - gyroskop. Na zber dát zo senzora sme použili mikrokontrolér Arduino UNO, resp. jeho čínsku kópiu. Táto hardvérová platforma má USB rozhranie pomocou, ktorého sme zozbierané dáta posielali cez sériovú linku na náš virtuálny server bežiaci v prostredí Raspberry Pi OS (RPIOS). Rovnako na tomto virtuálnom serveri bežala MySQL databáza, do ktorej boli dáta zapisované a neskôr pomocou webového rozhrania načítavané v prehliadači klienta. Na tvorbu webového rozhrania bol použitý webový aplikačný framework pre jazyk Python - Flask. Ide o open-source framework a umožňuje rýchly vývoj webových aplikácií. Samotná webová aplikácia bola naštýlovaná pomocou css a časť jej funkcionality bola implementovaná pomocou javascriptu.

1.2 Hardvérová konfigurácia

Na zostavenie systému sme použili nasledovné HW komponenty:

- Arduino UNO (Čínska kópia)
- Senzor MPU-6050 - gyroskop aj akcelerometer snímajúci v troch osiach
- Prepojovacie vodiče
- USB 2.0 kábel - typ A/B
- Server s raspberry Pi OS

Na schéme môžeme vidieť HW zapojenie meracej časti Arduino Uno a senzoru MPU-6050.



Obr. 1: Schéma zapojenia

1.3 Softvérová konfigurácia

Na realizáciu projektu bol použitý SW s nasledovnou konfiguráciou.

1.3.1 Arduino IDE

Rozhranie slúžilo na programovanie mikrokontroléra Arduino UNO. Na zber údajov zo senzora MPU-6050 sme použili nasledovné hlavičkové súbory:

- Adafruit_MPU6050.h - potrebné doinštalovať cez Library Manager
- Adafruit_Sensor.h
- Wire.h - hlavičkový súbor je súčasťou knižnice rozhrania Arduino IDE

Hlavičkové súbory od Adafruitu je potrebné doinštalovať cez Library Manager. Sú súčasťou *Adafruit MPU6050* knižnice a pri tvorbe projektu bola využívaná verzia 2.2.4.

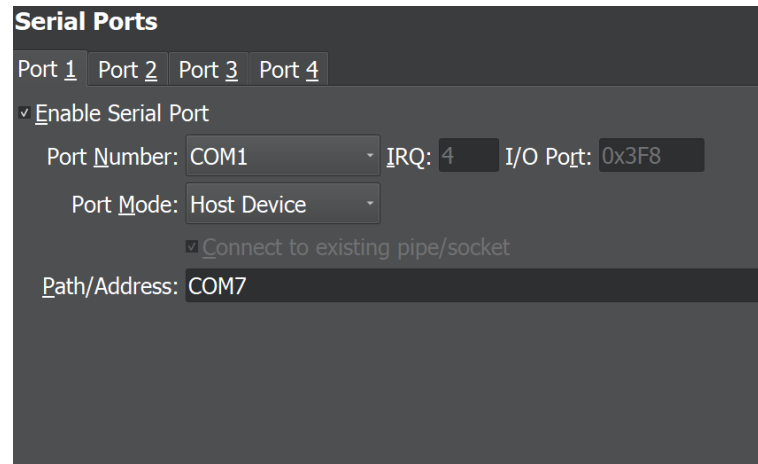
1.3.2 Visual Studio Code

Ako programovacie prostredie sme využívali Visual Studio Code s rozšírením SSH (ssh extension) čím sme si otvorili komunikačný kanál s našim webovým serverom.

1.3.3 VirtualBox

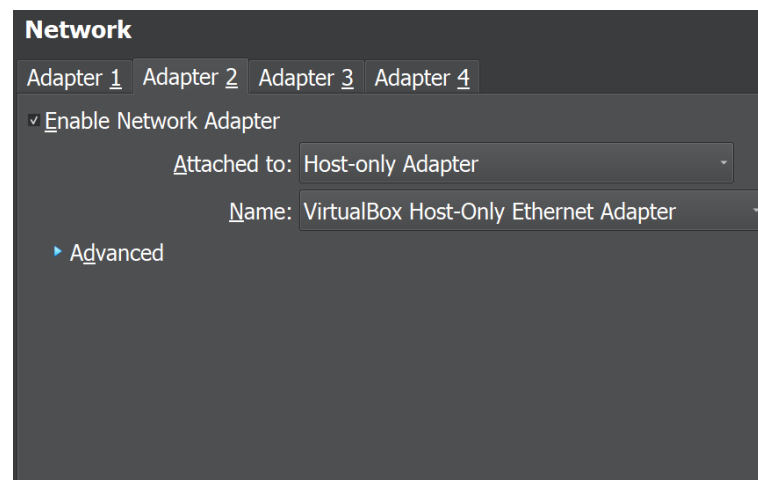
Serverová časť nám beží vo virtuálnom prostredí Raspberry PI OS, ktorý sme mali spustený v open-source virtualizačnom softvéri. Inštaláciou obrazu operačného systému sa nebudeme zaoberať. Je však potrebné spraviť konfiguračné zmeny virtuálneho stroja v nástroji VirtualBox. To nám zabezpečí komunikáciu s mikrokontrolérom Arduino UNO

cez sériový port a komunikáciu na databázu, webový server a ssh službu spustenú vo virtuálnom OS.



Obr. 2: Nastavenie virtuálnej sériovej linky

treba dbať na to aby hodnota pola Path/Adress zodpovedala hodnote portu na ktorom komunikuje Arduino UNO (Window - Device Manager - Ports).



Obr. 3: Nastavenie virtuálneho sieťového adaptéra

1.3.4 Raspberry PI OS

Ako už bolo spomenuté Raspberry PI OS beží vo virtuálnom prostredí nástroja VirtualBox. Slúži ako server a beží na ňom webový server a databáza. Zhrnieme si prerekvizity a konfiguráciu. Pred akoukoľvek inštaláciou nižšie spomínaných balíčkov odporúčame dosiahnuť aktuálnosť inštalácií programových balíčkov v Raspbian-e.

"sudo apt-get update" a "sudo apt-get upgrade".

Webový server

1. Raspbian štandardne obsahuje inštaláciu Pythonu, v prípade, že OS nemá Python nainštalovaný, je potrebné ho doinštalovať.
2. Na beh webového servera je potrebné nainštalovať podporu websocketov pre mikrof-ramework Flask. Použijeme inštaláciu pre Python3.

```
"sudo pip3 install flask-socketio".
```

Webová aplikácia

Na prácu s MySQL DB cez webové rozhranie potrebujeme doinštalovať knižnicu pre mysql a databázový connector.

```
"sudo pip3 install mysql"  
"sudo pip3 install mysql.connector"
```

Na prácu so sériovým portom sme využívali knižnicu PySerial.

```
"sudo pip3 install pyserial"
```

Webová aplikácia používa aj externé zdroje pomocou hypertextovej referencie, bez nutnosti ich inštalácie (v súbore index.html). Je preto žiaduce mať povolený zoznam IP na firewall. Štandardne sú tieto prestupy povolené a netreba vykonávať navyše žiadnu konfiguráciu. Nižšie uvádzame zoznam využitých zdrojov a potrebné prestupy.

1. CSS framework TailWind
2. WebGL (Web Graphics Library) - JavaScript API pre vytváranie 3D grafiky vo webovom prehliadači bez použitia pluginov alebo iných doplnkových aplikácií.
3. Chart JS - JavaScript pre vykresľovanie grafov
4. Gauge Chart - JavaScript pre vykresľovanie analógových ciferníkov

Source	Destination Host	Port/Protocol	Poznámka
RPI OS server	code.jquery.com	443/https	Štandardná JQuery knižnica
RPI OS server	cdnjs.cloudflare.com	443/https	Využitie pre WebGL a Socket.IO
RPI OS server	cdn.jsdelivr.net	443/https	Knižnica pre vykresľovanie grafov
RPI OS server	cdn3.devexpress.com	443/https	JavaScript knižnica DevExpress

Tabuľka 1: Tabuľka s používanými knižnicami a ich zdrojovými hostami.

Databázový server

Ako DB server sme používali MySQL databázu, podľa dostupných návodov z prednášky. Inštalácia je nasledovná.

```
sudo pip3 install simple-websocket
// aby pythonovsky server nevyhadzoval warning

sudo apt-get install default-mysql-server default-mysql-client
// instalacia databazoveho servera a klienta
sudo mysql_secure_installation
// zabezpecenie databazy heslom
Enter current password for root (enter for none):
Set root password? [Y/n] y
New password:heslo //moze byt ine, silnejsie
Re-enter new password:heslo //moze byt ine, silnejsie
Remove anonymous users? [Y/n] y
Disallow root login remotely? [Y/n] n
Remove test database and access to it? [Y/n] n
Reload privilege tables now? [Y/n] y

//uprava konfiguraku, aby sme mohli pristupovat zvonku
cd /etc/mysql/mariadb.conf.d/
sudo cp 50-server.cnf 50-server.cnf.zaloha
sudo nano 50-server.cnf
"bind-address = 127.0.0.1" treba zmenit na "bind-address = 0.0.0.0"
sudo service mysql restart //restart mysql servera
```

1.3.5 GitHub

Pre hosting a správu nášho projektu pomocou verzovacieho systému Git sme využívali webovú platformu GitHub. Všetky zdrojové kódy sú dostupné v našom repozitári na GitHub: https://github.com/MefjuG/POIT_2023.

1.3.6 SSH - nepovinné

Na programovanie webovej aplikácie sme využívali Microsoft Visual Studio Code. Na Virtuálnom serveri sme si preto potrebovali nastaviť službu SSH.

```
"sudo apt install openssh-server"
```

SSH služba sa následne zapína/vypína pomocou

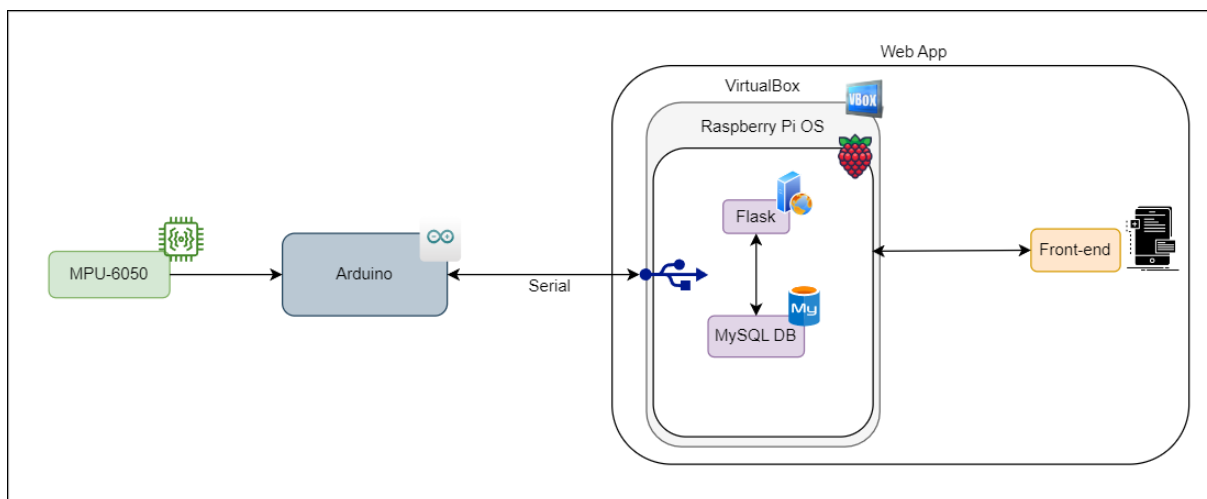
```
"sudo service ssh start" alebo "sudo service ssh stop"
```

Tiež sme potrebovali doinštalovať knižnice potrebné pre VSCode server.

```
"sudo apt-get install g++-multilib"
```

1.4 Architektúra systému

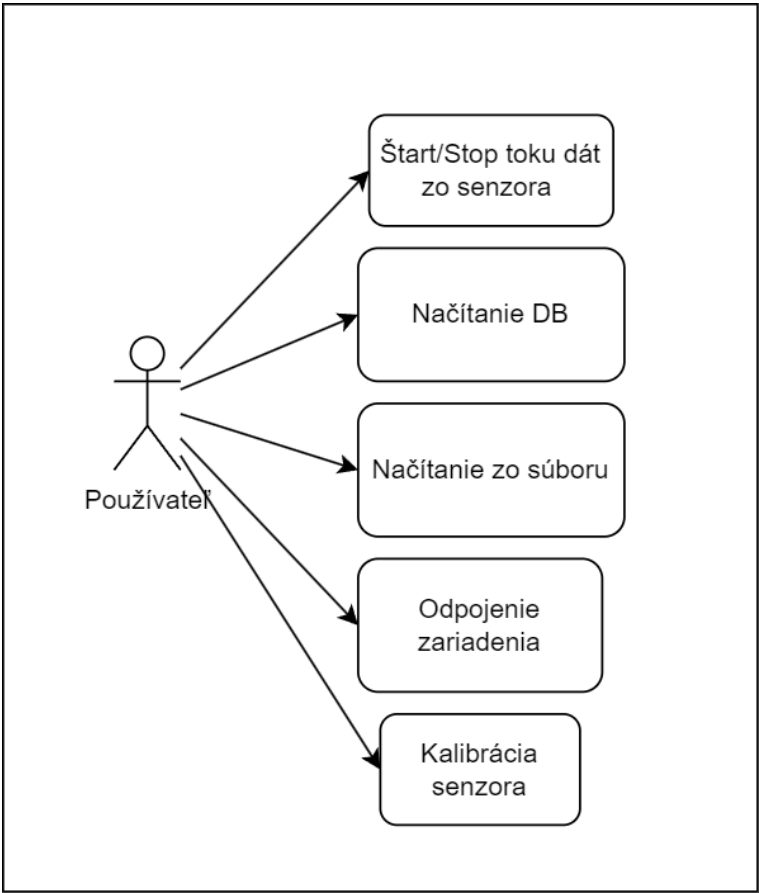
Navrhli sme architektúru systému, ktorá zahŕňa senzor MPU6050, Arduino UNO, Raspberry OS, Flask a MySQL databázu. Nasledovná schéma systému zobrazuje, ako sú jednotlivé komponenty projektu navzájom prepojené.



Obr. 4: Architektúra systému

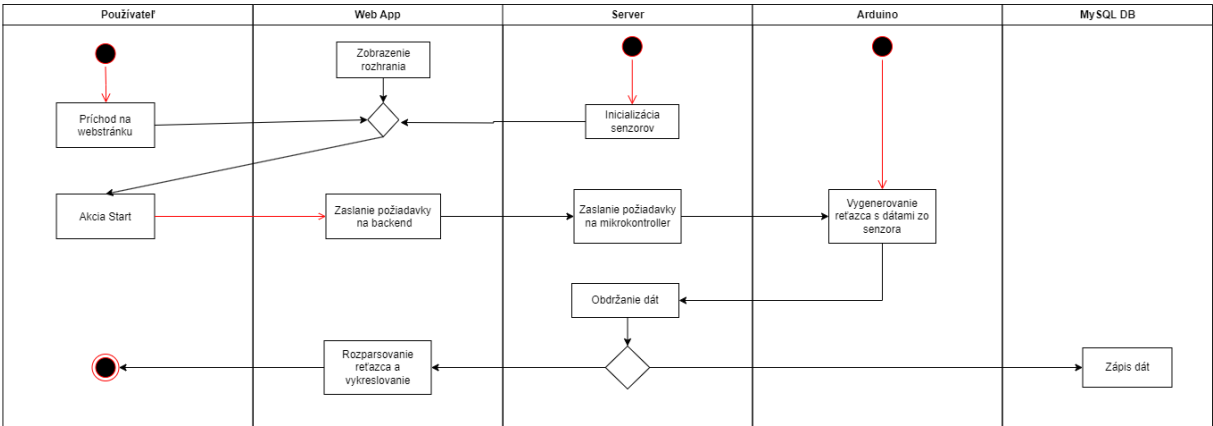
1.5 UML diagramy

1.5.1 Diagram prípadov použitia

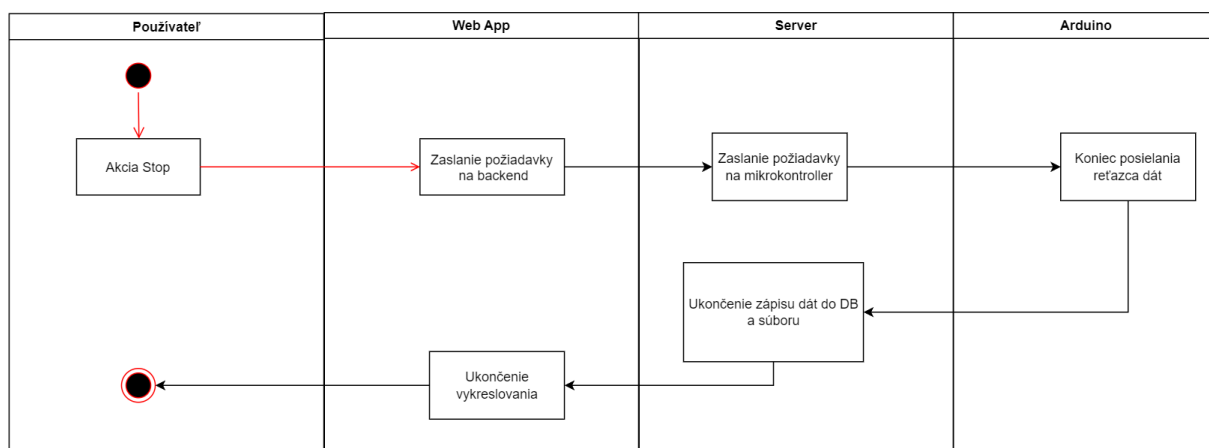


Obr. 5: Prípady použitia

1.5.2 Diagramy aktivít

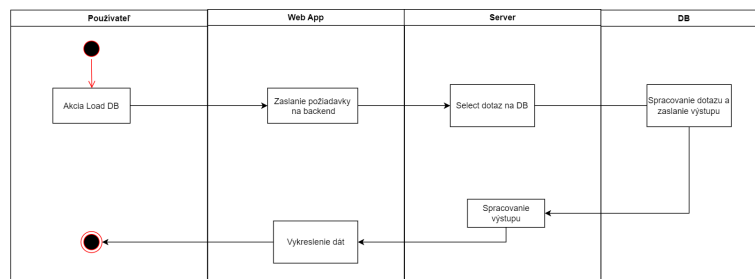


Obr. 6: Activity diagram - Start



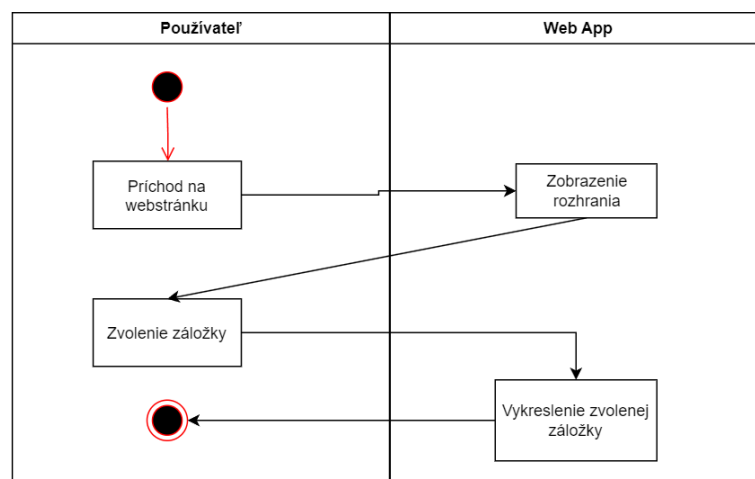
Obr. 7: Activity diagram - Stop

Activity diagram pre Disconnect nebudeme popisovať nakoľko je podobný ako pri STOP. Jediný rozdiel je, že server po zaslaní žiadosti na Arduino UNO odpojí mikrokontrolér a nie je možné vykonať akúkoľvek nadchádzajúcu akciu zo strany používateľa na mikrokontrolér.



Obr. 8: Activity diagram - Načítanie DB

Activity diagram pre načítanie zo súboru nebudeme popisovať nakoľko je podobný ako pri načítaní DB. Jediný rozdiel je, že server neposiela dotaz na DB ale rovno načíta súbor z filesystemu.



Obr. 9: Activity diagram - voľba záložky

1.6 Komunikačné protokoly

Komunikácia medzi Arduino UNO a Raspberry OS prebieha sériovo cez USB. Arduino UNO posiela dáta vo formáte JSON, kde každá položka reprezentuje senzorové údaje o akcelerácii (aX, aY, aZ), uhlových rýchlostiach (gX, gY, gZ), teplote (temp) a uhlových pozíciách (angX, angY). Ktoré sú predvypočítavané na strane Arduino Uno. Tieto dáta sa následne prijímajú v Raspberry OS a spracúvajú sa v aplikácii Flask, ktorá používa WSGI komunikáciu s klientom a s MySQL databázou.

Flask zabezpečuje správne rozparovanie dát z JSON formátu a spracovanie údajov do MySQL databázy a textového súboru. Aplikácia používa MySQL Connector pre komunikáciu s databázou.. Použitie ORM (Object-Relational Mapping) umožňuje aplikácii pracovať s databázou pomocou Python objektov, čo zjednodušuje prístup k dátam v databáze.

Klientská strana aplikácie je implementovaná pomocou html, CSS a JavaScriptu. Využíva AJAX požiadavky na obnovovanie dát a načítavanie nových údajov z Flask API.

Celá komunikácia medzi klientom, serverom a databázou prebieha pomocou HTTP protokolu. Toto je vhodné do budúcnosti zmeniť na protokol HTTPS, ktorý zabezpečuje šifrovanie dát a zabezpečenú komunikáciu.

2 Používateľská príručka

Táto sekcia obsahuje krátku používateľskú príručku pre náš projekt. Obsahuje základné informácie o funkcionalite aplikácie a návod na jej používanie.

2.1 Inštalačná príručka

Detaily konfigurácie jednotlivých komponentov sú popísané vo vývojárskej príručke. V tejto sekcii si zhrnieme postup inštalácie a spustenia.

1. Zostaviť HW časť podľa schémy zapojenia Arduino Uno a senzora MPU-6050 z obrázku 1.
2. Nahrať kód s potrebnou funkcionalitou na Arduino Uno cez USB rozhranie, pomocou nástroja Arduino IDE. Jedná sa o zdrojový súbor s názvom *arduino_final.ino*
3. Inštalácia a konfigurácia Virtualizačnej platformy, v našom prípade VirtualBox.
4. Inštalácia virtuálneho stroja s Raspbian OS.
5. Konfigurácia VM portov a sieťového adaptéra podľa pokynov v kapitole 1.3.3
6. Spustenie Raspbianu a konfigurácia potrebných knižníc a MySQL DB podľa pokynov v kapitole 1.3.4
7. Prenesenie priečinka *semestralka* obsahujúci zdrojové súbory k webovému frameworku Flask a webovej aplikácii na monitorovanie a riadenie signálov zo senzora MPU-6050
8. Zmena konfiguračnej hlavičky pre MySQL DB konektor v súbore *semestralka.py*, tak aby zodpovedala konfigurácii našej DB.

```
mysql_config = {  
    'user': 'user',  
    'password': 'heslo',  
    'host': 'IP',  
    'db': 'mysql',  
    'raise_on_warnings': True  
}
```

9. Otvorenie priečinka v Terminále na Raspbiane a spustenie webovej aplikácie pomocou príkazu

```
"sudo python semestralka.py"
```

10. Zistenie adresy, na ktorom je aplikácia spustená. Pomocou tejto adresy pristupujeme k rozhraniu aplikácie z externého prostredia. Po zadaní príkazu je to väčšinou druhý záznam z terminálu.

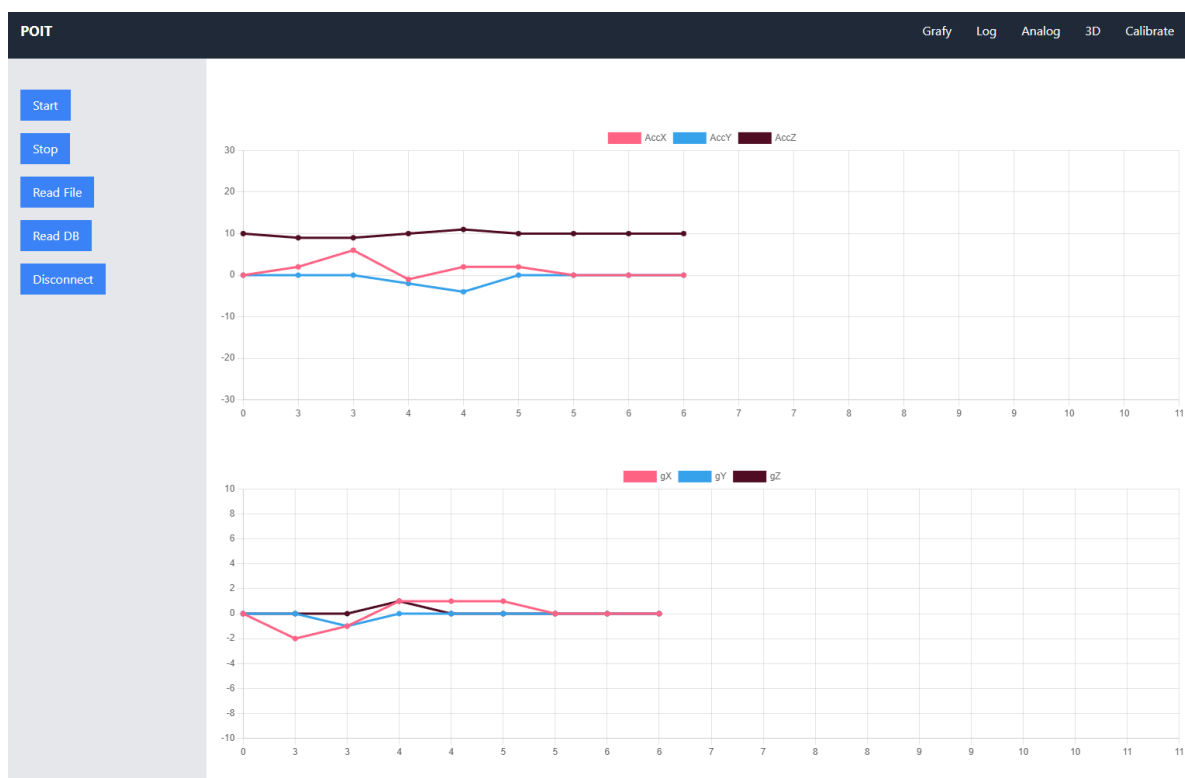
```
"ifconfig"
```

11. V prípade správnej konfigurácie by sa po načítaní používateľského rozhrania vo webovom prehliadači mala v termináli zobrazíť hláška *"MPU6050Found!"*, čo signalizuje úspešnú inicializáciu senzora.

2.2 Popis rozhrania a funkcionalít

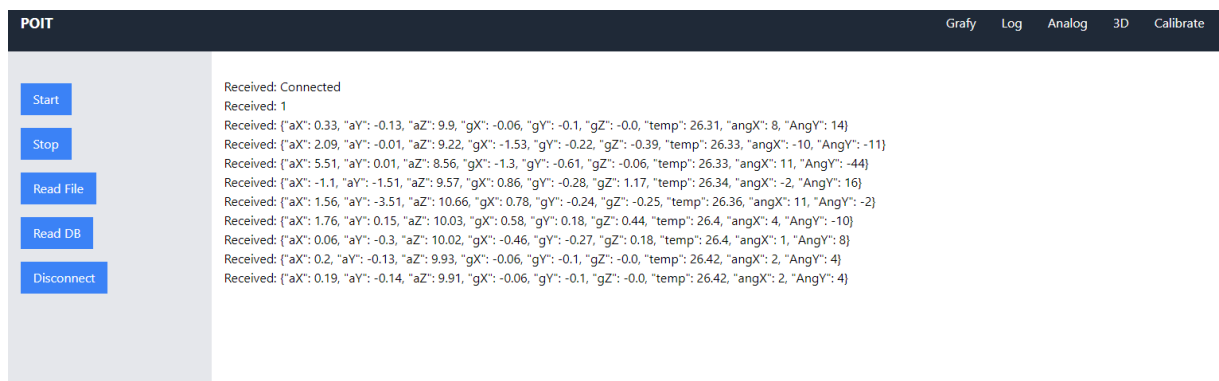
Rozhranie aplikácie si vieme zobrazíť v prehliadači na adrese servera. V rozhraní sa nachádza 5 záložiek.

- **Grafy** - vykresľuje prichádzajúce dáta o akcelerometri a gyroskopu zo senzora v podobe grafov.



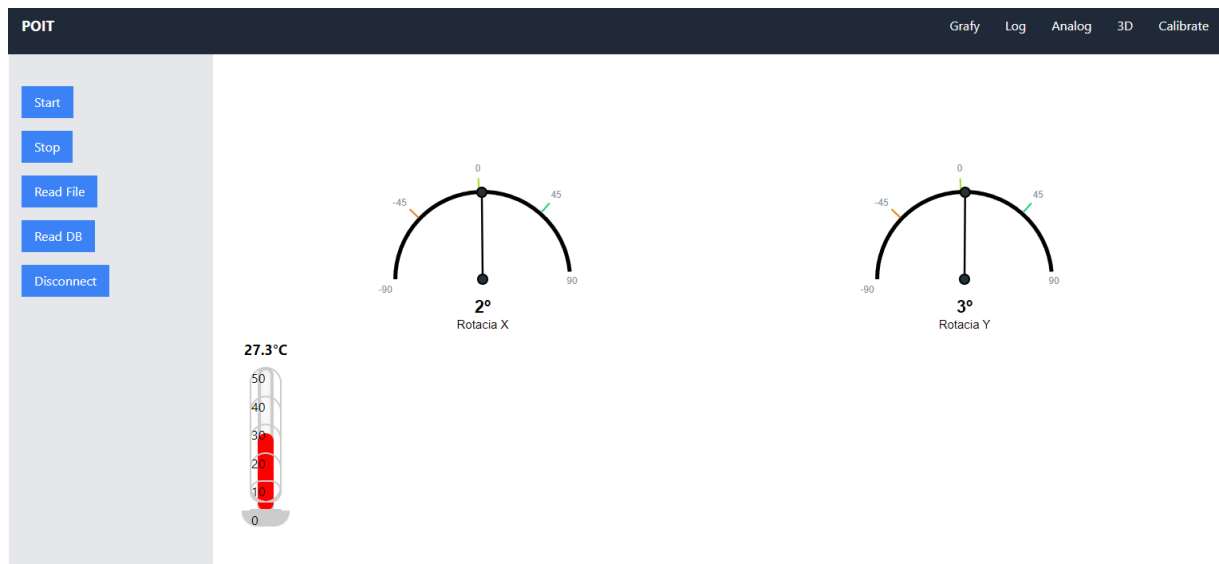
Obr. 10: Rozhranie aplikácie - Grafy

- **Log** - zobrazuje raw formát prichádzajúcich dát v podobe JSON objektu.



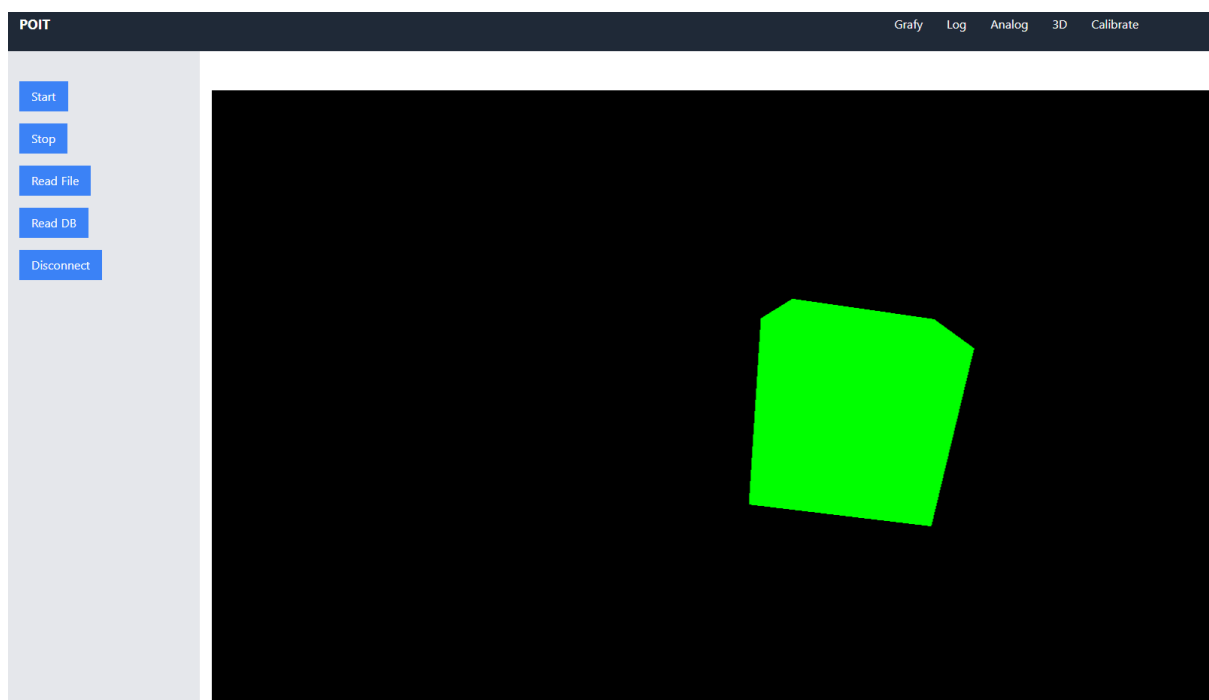
Obr. 11: Rozhranie aplikácie - Log

- **Analog** - vykresľuje prichádzajúce dáta o uhle naklonenia X,Y osi a teplotu z termometra.



Obr. 12: Rozhranie aplikácie - Analog

- **3D** - vykresľuje prichádzajúce dáta o uhle naklonenia v podobe 3D objektu kocky pomocou WebGL knižnice.



Obr. 13: Rozhranie aplikácie - 3D

- **Calibrate** - umožňuje používateľovi ručne nakalibrovať hodnoty senzorov.

Obr. 14: Rozhranie aplikácie - Calibrate

V ľavom menu aplikácie sa nachádza 5 tlačidiel, s ktorými môže používateľ interagovať. Tento panel poskytuje používateľovi začať/skončiť zber dát zo senzora. Načítať súbor na disk servera alebo z DB. Alebo celé spojenie s mikrokontrolérom vypnúť. V tomto prípade nie je možné vykonať žiadnu ďalšiu akciu pomocou tlačidiel. Na opätovné pripojenie sa k senzoru je potrebné vykonať obnovenie stránky.

! Je potrebné podotknúť, že po stlačení tlačidla štart sa dáta neustále snímajú a zapisujú do DB aj súboru. Používateľ nemá kontrolu zvoliť si kam sa má súbor uložiť alebo do akej tabuľky. Zápis pokračuje vždy na konci súboru alebo tabuľky. Po stlačení tlačidla Load File/Load DB sa vždy načíta celý súbor/celá tabuľka DB. Súbor s dátami sa nachádza v priečinku s aplikáciou *semestralka*.

2.3 Návod na používanie

Ak sme postupovali pri inštalácii správne zobrazíme si rozhranie v prehliadači na adrese servera. Ovládanie je popísané v predošlej sekcii a je plne intuitívne.