

Očitavanje senzorskih podataka korištenjem računala Raspberry Pi 3

3. travnja 2017.

Sadržaj

1 Uvod

2 Raspberry Pi

- 2.1 Sklopovlje
- 2.2 Programska podrška
- 2.3 Primjeri korištenja

3 Senzori

- 3.1 Komunikacijski protokoli
- 3.2 Akcelerometar - Adafruit LIS3DH
- 3.3 Mikrofon - Adafruit MAX4466?

4 Pregled dostupnih programskih okvira

5 Primjeri

- 5.1 Programski kôd
- 5.2 Očitani podaci

6 Zaključak

7 Literatura

1 Uvod

Ovaj će seminarski rad obraditi problematiku korištenja računala Raspberry Pi 3 za prikupljanje podataka sa senzora. Ukratko će se opisati sklopovska arhitektura računala Raspberry Pi 3 te pripadajuća programska podrška, uz nekoliko primjera korištenja.

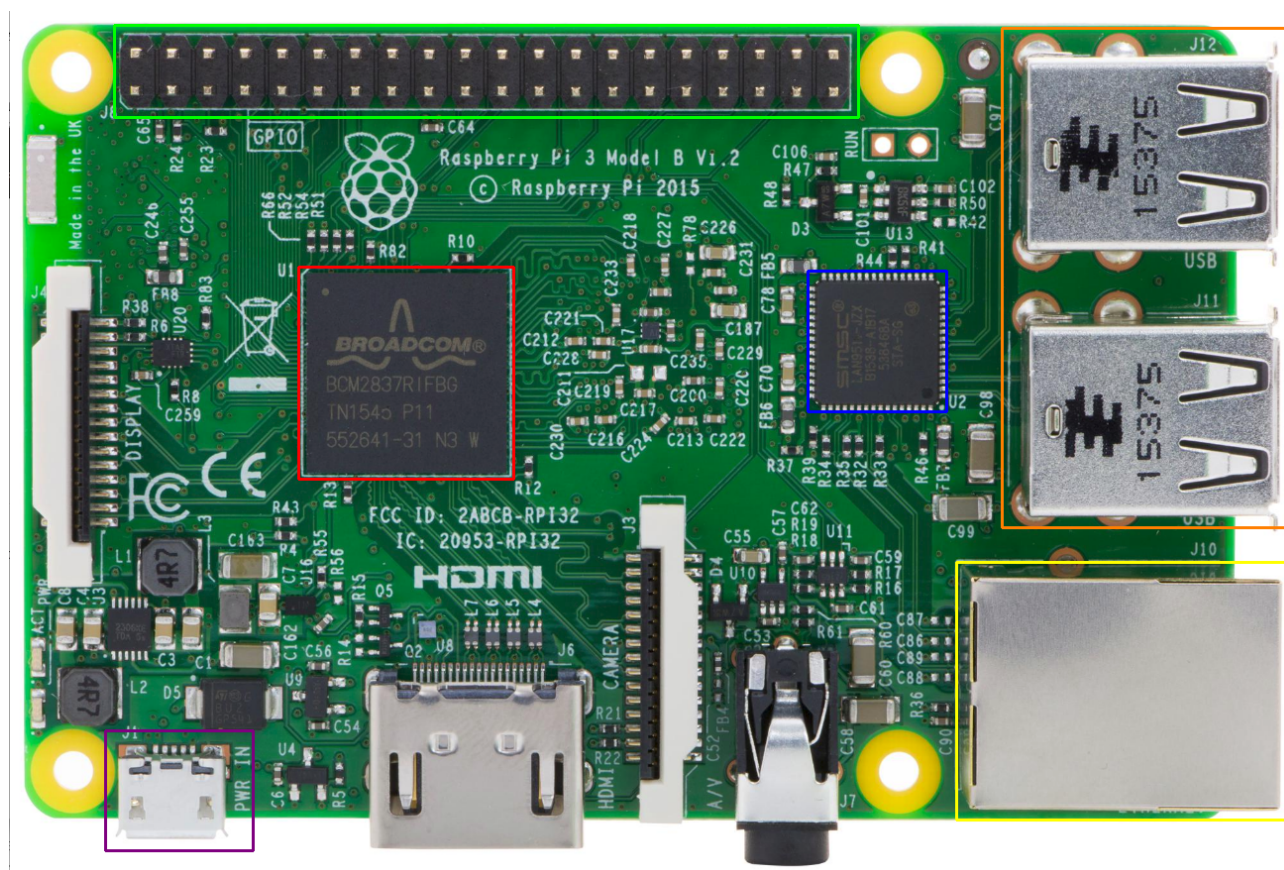
Kao vrlo pristupačno malo računalo, Raspberry Pi je vrlo popularan kao ugradbeno računalo, a budući da ugradbena računala vrlo često za svoj rad koriste raznolike senzore, za ovaj će rad biti ključno razumjeti osnovne principe povezivanja senzora i računala, uzimajući u obzir sklopovski i programski aspekt. Shodno tome, bit će izložen kratak opis jednog akcelerometra, u funkciji senzora za mjerenje vibracija, te jednog mikrofona, poglavito u funkciji senzora glasnoće.

Ukratko će se opisati neke od dostupnih biblioteka i programskih okvira namijenjenih za rad sa sensorima, s naglaskom na već spomenute akcelerometre i mikrofone. Konačno, bit će pokazan i jednostavan primjer programskog koda za očitavanje podataka sa senzora uz prateći primjer vizualnog prikaza senzorskih podataka.

2 Raspberry Pi

Raspberry Pi je serija malih računala razvijanih od strane zaklade Raspberry Pi Foundation. Odlikuju ih niska cijena, dobre performanse s obzirom na cijenu, pristupačnost i lako korištenje te male fizičke dimenzije. Programska je podrška otvorena i vrlo dobro dokumentirana, a zajednica ljudi koji koriste Raspberry Pi je velika, pristupačna i konstruktivna.

U trenutku pisanja su dostupni Raspberry Pi 1, 2 i 3, te minijturni Raspberry Pi Zero. U nastavku će rada riječ biti o Raspberry Pi 3 inačici, koja se vidi na Slici 1, a bit će oslovljena kao "RPi3".



Slika 1: Raspberry Pi 3B.

2.1 Sklopovlje

Glavni čipovi Glavne procesne jedinice nalaze se u čipu Broadcom BCM2837, koji je na slici označen crvenim pravokutnikom. Riječ je o tzv. *System-on-Chip* (SoC) čipu koji sadrži:

- **CPU** - 64-bitni ARMv8 Cortex A53 s četiri jezgre na 1.2 GHz,
- **GPU** - VideoCore IV na 400 MHz.

Taj je SoC spregnut s **radnom memorijom** s druge strane tiskane pločice. Riječ je o LPDDR2 SDRAM memoriji, s kapacitetom od 1 GB.

Za bežičnu je komunikaciju zadužen procesor osnovnog pojasa (engl. *baseband processor*) BCM43438, također od tvrtke Broadcom, koji podržava WiFi i Bluetooth 4.1 protokole. Smješten

je na stražnjoj strani pločice, a spregnut je s antenom na prednjoj strani. Nije zgorega napomenuti da je Raspberry Pi 3 prva inačica ovog računala koja ima ugrađeno sklopovlje za bežičnu komunikaciju.

Konačno, SMSC LAN9514 (na slici uokviren plavom bojom) vrši funkciju USB čvora i Ethernet upravljača. Povezan je s procesorom jednom USB vezom, pa sa svakim od četiri USB priključka, te s Ethernet priključkom procesor komunicira preko te USB veze.

Ulazno-izlazni priključci Kao i svako drugo računalo, i RPi3 bi bio prilično beskoristan bez mogućnosti komunikacije s vanjskim svijetom. Naravno, na njemu postoji mnoštvo ulazno-izlaznih sučelja, a način na koji su ona izvedena je uvelike zaslužan za takvu popularnost ovog računala. Za ovaj seminar najvažniji priključci označeni su na Slici 1, a u nastavku je dan pregled tih sučelja:

- Četiri **USB 2.0** priključka [narančasti pravokutnik]
- Jedan **Ethernet** priključak [žuti pravokutnik]
- Jedan **microUSB** priključak - samo za napajanje, ne i za komunikaciju [ljubičasti pravokutnik]
- Četrdeset ulazno-izlaznih pinova opće namjene (**GPIO** - *General Purpose Input/Output*) [zeleni pravokutnik]
- Ulaz za **microSD** karticu (s druge strane pločice)
- Audio priključak i HDMI izlaz
- Poseban priključak za službenu Raspberry Pi kameru
- Poseban *display* priključak

Od posebne su važnosti za ovaj rad GPIO pinovi. Oni, su, naime, ključni za komunikaciju sa senzorima koji će biti razmotreni kasnije u radu. Omogućuju najčešće korištene *low-level* protokole za sklopovsku komunikaciju: **I²C**, **TTL** i **SPI**. Naravno, sadrže i pinove za napajanje i uzemljenje sklopova, kao i pinove namijenjene za korisničku definiciju ponašanja.

2.2 Programska podrška

Programska je podrška za RPi3, baš kao i sklopovlje, izvedena s ciljem jednostavnosti i fleksibilnosti korištenja. Budući da nije riječ o mikrokontroleru (poput, primjerice, popularnog Arduina), već o potpuno opremljenom računalu, očito je da RPi3 mora imati neki operacijski sustav (u nastavku: OS). On se u glavnini slučajeva nalazi na **microSD** kartici, koja ima ulogu sekundarne memorije.¹ Ovdje kao najpopularniji OS za RPi u priču ulazi **Raspbian**.

Raspbian je besplatan i otvoren operacijski sustav baziran na distribuciji Linuxa zvanoj "Debian". Ono zbog čega je Raspbian toliko popularan jest činjenica da je taj **OS optimiziran upravo za Raspberry Pi sklopovlje**.² Drugim riječima, Raspbian je distribucija Linuxa koja:

- je vrlo jednostavna za instalaciju i dolazi potpuno opremljena za normalan računalni rad - uključujući i **grafičko korisničko sučelje** (*GUI* - *Graphical User Interface*)
- dolazi s već instaliranom programskom podrškom za komunikaciju sa specijaliziranim sklopovljem (poput GPIO pinova) - primjerice, **iz terminala, C-a ili Pythona**.

¹Moguće je pokrenuti OS i s, primjerice, nekog USB tvrdog diska, ali to nadilazi temu ovog seminara.

²Ovo se može zaključiti i iz imena distribucije, koje je *portmanteau* pojmova "Raspberry Pi" i "Debian".

2.3 Primjeri korištenja

Uzevši u obzir ranije spomenute prednosti RPi3 računala - cijena, pristupačnost i dimenzije - RPi računala najčešće nađu svoju primjenu u dva konteksta:

- u edukativnim projektima (pristupačnost i cijena)
- u ugradbenim sustavima (male dimenzije i cijena)

U nastavku su ukratko izložena dva primjera korištenja RPi3 računala. Primjeri su relativno jednostavni, ali dobro ilustriraju moguće koncepte korištenja tog računala.

Pametna brava Pametne su brave koncept novijeg datuma, a označavaju autorizaciju ulaza osobe u neki prostor, i to ne koristeći (samo) klasični, mehanički ključ, nego korištenje kartica, lozinki, ili čak primjerice, prepoznavanja lica. Za implementaciju jedne takve brave može poslužiti bilo kakav mikrokontroler, i RPi3 bi svakako bio pretjeran za tako jednostavan zadatak, ali ono što bi moglo umanjiti taj *overhead* jest, primjerice, praćenje pristupa prostoru pomoću logova, udaljeno upravljanje bravom, ili agregiranje više neovisnih zadataka slične složenosti na jednom RPi-ju.

Računarska sekcija Kluba studenata elektrotehnike konkretno koristi Raspberry Pi 3 za agregiranje informacija o radu nekoliko servera, a u planu je spajanje NFC čitača (*Near-Field Communication* - tehnologija beskontaktna komunikacije korištena u studentskim iskaznicama i bankovnim karticama) na taj isti RPi3, kako bi se na taj način autenticiralo osobe koje žele ući u sekciju.

Kućni VPN i WiFi pristupna točka U kontekstu komunikacijskih mreža, VPN (*Virtual Private Network*) je protokol spajanja na mrežu koji omogućuje virtualno pristupanje mreži s drugog mjesta i prosljeđivanje svog lokalnog prometa kroz taj "tunel" na način koji osigurava privatnost i sigurnost, kao i (do neke mjere) svojstva mreže (dozvole i zabrane, primjerice, ili simulacija lokalnog pristupa) kakva su na udaljenoj mreži na koju je korisnik VPN-om spojen. Implementacijski detalji VPN-a nipošto nisu trivijalni, i neće se dalje razmatrati u ovom radu, ali takav se sustav relativno bezbolno može upogoniti upravo koristeći Raspberry Pi, umjesto da se (samo?) za taj zadatak koristi neko (red veličine skuplje) računalo opće namjene.

Još jedna zanimljiva mogućnost koju pruža RPi3 (ali ne i ranije inačice - barem ne bez dodatnog sklopovlja) jest postavljanje pristupne točke za WiFi, koja bi bežični promet prosljeđivala žičnoj Ethernet vezi. Tako se može, u nedostatku potpunijeg i trajnijeg rješenja, ostvariti *ad hoc* bežična mreža ili pojačivač signala.

3 Senzori

Sada kada je definirano računalo koje će agregirati i analizirati očitavanja senzora koje ćemo koristiti, vrijeme je da i o njima nešto rečemo. Slučaj koji promatramo jest **mjerenje vibracija** na mirnim površinama poput vrata, sa svrhom utvrđivanja **izvora** vibracija. Shodno tome, slijede kratki pregledi jednog akcelerometra - senzora za mjerenje relativnog ubrzanja - te jednog senzora glasnoće. Ipak, prije toga, radi lakšeg razumijevanja slijedi vrlo kratki opis dva ranije spomenuta *low-level* komunikacijska protokola.

3.1 Komunikacijski protokoli

I²C I²C (engl. *Inter-Integrated Circuit*) komunikacija je protokol za sinkronu serijsku komunikaciju. Odlikuje ga iznimna jednostavnost korištenja - za upogoniti I²C komunikaciju potrebne su samo dva pina:

- **SDA** (*Serial DAta line*) - podatkovna linija
- **SCL** (*Serial CLock*) - linija sa signalom takta, za sinkronizaciju.

Protokol omogućuje spajanje više uređaja s I²C mogućnosti na istu sabirnicu, što znači da postoji i koncept adresiranja, kao i *master-slave* arhitekture. Detalji implementacije toga su preširoki za ova rad, ali ukratko - svaki I²C uređaj mora imati svoju, na sabirnici jedinstvenu adresu, koja se obično sastoji od dvije heksadekadske znamenke. Ono što je jednostavnost uporabe žrtvovala jest mogućnost istovremene dvosmjerne (engl. *full duplex*) komunikacije - budući da postoji samo jedna podatkovna linija, očito je da je to nemoguće, već je moguća samo **naizmjenična dvosmjerna** (engl. *half duplex*) komunikacija.

SPI SPI (*Serial Peripheral Interface*) je također protokol za sinkronu serijsku komunikaciju, ali nešto drugačije koncipiran (i, posljedično, implementiran) od I²C-a. Također je zamišljen kao *master-slave* arhitektura s jednim *masterom*, a za komunikaciju koristi četiri pina:

- **SCLK** (*Serial CLock*) - linija sa signalom takta, za sinkronizaciju,
- **MOSI** (*Master Output Slave Input*) - izlaz iz *master* komponente i ulaz u *slave* komponentu,
- **MISO** (*Master Input Slave Output*) - ulaz iz *slave* komponente i izlaz u *master* komponentu; također jedina linija koja ulazi u *master* komponentu,
- **SS** (*Slave Select*) - pin za adresiranje *slave* komponente.

Budući da su u SPI protokolu prisutne dvije podatkovne linije, sada je već **moguća *full duplex*** komunikacije, i to je ujedno glavna prednost SPI-a nad I²C-om.

3.2 Akcelerometar - Adafruit LIS3DH

Tvrtka Adafruit se bavi razvojem otvorenog i pristupačnog sklopovlja, poglavito u edukativne svrhe. Tvrtka je od osnutka izdala na tržište mnoštvo vrlo raznolikih komponenti, a među njima i nekoliko akcelerometara. Jedan od njih je **LIS3DH**, vrlo popularan akcelerometar koji mjeri na 3 osi. Glavne karakteristike ovog sklopa su:

- mjerenje na **tri osi**
- **10-bitna** preciznost
- **I²C** i **SPI** mogućnosti
- vrlo mala potrošnja
- vrlo niska cijena



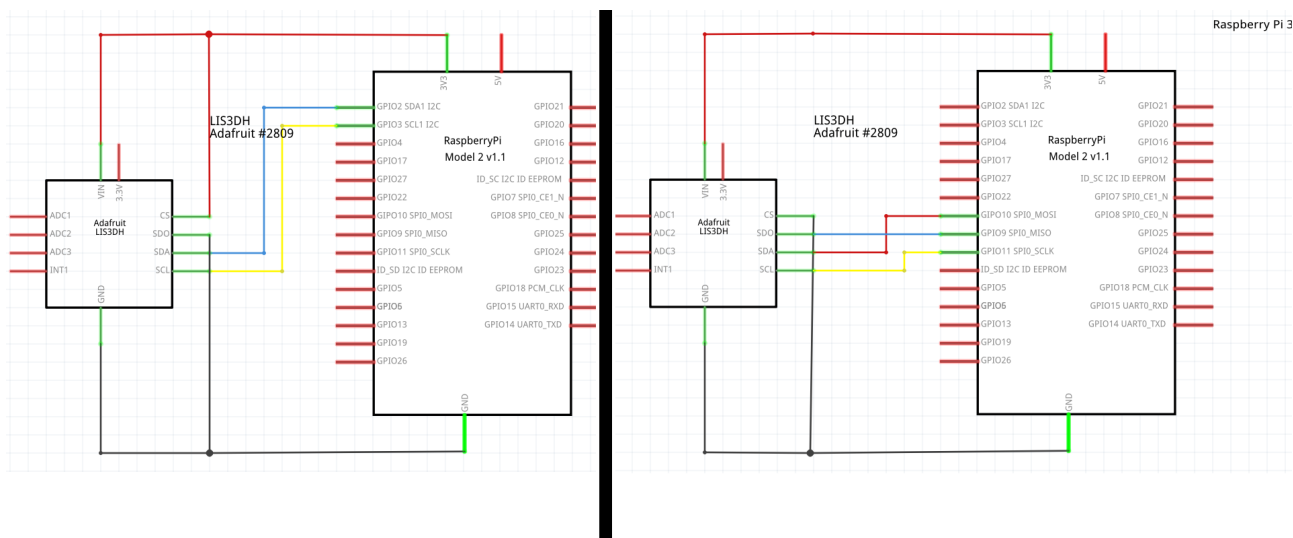
Slika 2: Akcelerometar LIS3DH.

Ako se koristi I²C, onda CS (*Chip Select*) mora biti ili otpojen ili pritegnut na visoku naponsku razinu. U protivnom, ako se koristi SPI, onda CS mora biti pritegnut na masu.

Sklop ima dvije moguće I²C adrese - 0x18 i 0x19, tako da je moguće spojiti dva ovakva senzora na isti *master*. Adresa se određuje tako da se pin SDO ostavi "u zraku" ili pritegne na nisku razinu za 0x18, odnosno na visoku razinu za 0x19.

RPi3 u svojim GPIO pinovima ima dva para I²C pinova - I²C0 i I²C1. Svejedno je koji se koristi, samo je ključno da se parovi ne pomiješaju - SDA0 mora biti uparen s SCL0, a ne s SCL1, recimo. Primjer I²C spajanja prikazan je na Slici 3 lijevo.

Spajanje preko SPI-a izvodi se tako da je SCL i dalje u ulozu signala takta, ali je pin označen kao SDA sada u ulozu MOSI, a pin SDO u ulozu MISO pina. CS, kao što je već rečeno, za SPI mora biti uzemljen. Primjer SPI spajanja prikazan je na Slici 3 desno.



Slika 3: Primjer povezivanja LIS3DH i RPi3 - I²C (lijevo) i SPI (desno).

Ovime su ispunjeni sklopovski preduvjeti za osnovnu komunikaciju između senzora i RPi-ja, i za jednostavne primjere nije potrebno ništa više, ali ovaj senzor ima još nekoliko pinova koji nude interesantne mogućnosti:

- INT - **prekidni pin**
- A1, A2, A3 - analogno-digitalni pretvornici (engl. **ADC** - *Analog-Digital Converter*)

Pinovi za ADC omogućuju da se na njih spoji neki analogni ulaz, koji će čip potom digitalizirati, s 10-bitnom preciznošću. Ovo se može pokazati pogodnim za situaciju u kojoj nemamo drugi ADC uređaj.

Prekidni pin omogućuje slanje prekidnog signala kao rezultat različitih događaja - senzor, naime, ima ugrađenu podršku za prepoznavanje jednostrukog i dvostrukog kuckanja, promjene orijentacije i slobodnog pada. Tu je podršku moguće iskoristiti kao okidač za generiranje prekida. Dakle, kada se dogodi neki događaj koji želimo pratiti (primjerice, kucanje po nekoj površini), sklop može generirati prekidni zahtjev da obavijesti RPi da je došlo do takvog događaja.

3.3 Mikrofon - Adafruit MAX4466?

test

4 Pregled dostupnih programskih okvira

Tijekom pisanja ovog rada bio je dostupan samo akcelerometar LIS3DH, a budući da je taj senzor za kratku i jednostavnu demonstraciju rada sa senzorima bio i više nego dovoljan, LIS3DH je korišten za sva mjerenja prikazana u nastavku. Senzor će biti spojen na RPi koristeći protokol I²C (kao što je prikazano na Slici 3 lijevo), a za komunikaciju na programskoj razini koristit ćemo programski jezik **Python**, koji je vrlo pogodan upravo za brzu izradnju i testiranje prototipa različitih sustava.

GPIO komunikacija u Pythonu Kao što je rečeno, i Adafruit i Raspberry Pi Foundation su tvrtke otvorene filozofije poslovanja, pa nude gotove biblioteke funkcija za rad s njihovim proizvodima. Iako je Adafruitu matična platforma Arduino, postoji i službena biblioteka za korištenje GPIO sučelja s njihovim proizvodima u Pythonu. Budući da smo za korištenje akcelerometra predvidjeli koristiti I²C, koji koristi određene GPIO pinove, svakako ćemo trebati tu biblioteku, dostupnu na Github repozitoriju: https://github.com/adafruit/Adafruit_Python_GPIO

Biblioteka za LIS3DH Sada kada imamo način komunikacije između Pythona i GPIO sučelja, potrebno je nekako uspostaviti komunikaciju i između GPIO sučelja i akcelerometra. Za taj će zadatak dobro poslužiti (neslužbena) biblioteka *python-lis3dh*, koja implementira glavnu funkcionalnost akcelerometra, a dostupna je na Github repozitoriju: <https://github.com/mattdy/python-lis3dh>

Biblioteka za MAX4466 PLACEHOLDER

5 Primjeri

Kao primjer korištenja akcelerometra provedeno je nekoliko relativno jednostavnih mjerenja:

1. Prvo je mjerenje provedeno tako da je senzor bio pričvršćen za vodoravnu površinu stola ljepljivom trakom - paralelno s njom - a promatrana su dva tipa podražaja:

- (a) **kucanje** po stolu uobičajene jačine
- (b) **agresivno udaranje** šakom po stolu

I kucanje i udaranje došlo je s druge strane stola - emulirajući, primjerice, kucanje na ulaznim vratima u stan. Također, svaki je period sadržao dva udarca, pa pauzu, a mjerenja su 4 perioda. Svrha ovakvog mjerenja je utvrđivanje mogućnosti detekcije udarca na temelju **impulsne vibracije**. Radi lakšeg čitanja resultantnog grafa i utvrđivanja preciznosti detekcije kucanja, ritam kucanja i udaranja pratio je metronom.

2. Drugo je mjerenje provedeno tako da je senzor bio pričvršćen na tijelo bas gitare, te je potom **odsviran ton A** (približno 55 Hz). Ovo je mjerenje odrađeno sa svrhom utvrđivanja koliko dobro senzor "snima" **kontinuiranu vibraciju** (s prirodnim prigušenjem).

Za prvo je mjerenje također korišten i **prekidni pin** (spojen na GPIO 04, što je definirano u datoteci `LIS3DH.py`), koji je generirao prekide kada bi otkrio kucanje. Budući da drugo mjerenje nije imalo impulsne podražaje, taj pin nije korišten.

Za komunikaciju između laptopa i RPi-ja korišten je **SSH (Secure SHell) protokol**, koji omogućuje udaljeno upravljanje nekim računalom u mreži - u bitnome, omogućuje mrežni pristup terminalu OS-a na RPi-ju i pokretanje naredbi na njemu.

Programski kôd i rezultati mjerenja (u grafičkom obliku) dani su u nastavku.

5.1 Programski kôd

Očitavanje podataka Za testiranje senzora korišten je Python program iz navedenog repozitorija, čija je glavna funkcija izložena u nastavku:

```
1 #!/usr/bin/python
2
3 from LIS3DH import LIS3DH
4 from time import sleep
5
6 if __name__ == '__main__':
7     sensor = LIS3DH(debug=True)
8     sensor.setRange(LIS3DH.RANGE_2G)
9     sensor.setClick(LIS3DH.CLK_SINGLE, 80, mycallback=clickcallback)
10
11     print "Starting stream"
12     while True:
13         x = sensor.getX()
14         y = sensor.getY()
15         z = sensor.getZ()
16
17 # raw values
18     print "\rX: %.6f\tY: %.6f\tZ: %.6f" % (x,y,z)
19     sleep(0.1)
```

Izlaz testnog programa se, dakle, većinski sastoji od ispisa **očitanja u X, Y i Z smjerovima**, a vrijednosti su realni brojevi, ugrubo u rasponu od -1 do 1. Radi sažetosti nije prikazan kôd odgovoran za otkrivanje kucanja, koje u ispis dodaje još nekoliko linija koje obavještavaju korisnika da je očitano kucanje ("single click").

Program je, naravno, pokrenut na RPi-ju, Bash naredbom:
`python testLIS3DH.py > out.txt.`

Agregiranje i iscrtavanje rezultata Da bismo na temelju očitanih podataka mogli iscrtati grafove, izlaz mjerenja spremljen je u tekstualnu datoteku `out.txt` na RPi-ju. Koristeći **SCP (Secure Copy Protocol)**, protokol za kopiranje datoteka između računala u mreži, ta je datoteka potom prebačena na laptop, gdje su podaci agregirani i vizualizirani. Za to je poslužila jednostavna Bash skripta:

```
#!/bin/bash

scp pi@$rpiw:/home/pi/Documents/Seminar/python-lis3dh/$1 .
tail -n +19 $1 > $1_nice
python plotter.py < $1_nice
```

Dakle, nakon prebacivanja datoteke s očitanjem na laptop (prva linija skripte) i njenog uređivanja (uklanjanje početnih linija koje ne sadrže mjerene podatke; druga linija skripte), u trećoj je liniji pokrenut vlastiti Python program koji agregira podatke i, koristeći ugrađenu biblioteku `matplotlib.pyplot`, iscrtava ih na grafove. Ukupno su generirane 4 za mjerenje impulsa, odnosno 3 krivulje za mjerenje kontinuirane vibracije:

```
1 import matplotlib.pyplot as plt
2
3 X = []
4 Y = []
5 Z = []
6 C = []
7
8 while(1):
9     try:
10         line = raw_input().split()
11         if "detected" in line:
12             continue
13         elif "click" in line:
14             C.append(1)
15         else:
16             X.append(float(line[1]))
17             Y.append(float(line[3]))
18             Z.append(float(line[5]))
19             C.append(0)
20     except EOFError as e:
21         break
22
23 plt.plot(X, color="r", label="X axis")
24 plt.plot(Y, color="g", label="Y axis")
25 plt.plot(Z, color="b", label="Z axis")
26 plt.plot(C, color="k", label="Click detection")
27 plt.ylabel("Relative acceleration")
28 plt.xlabel("Time")
29 plt.show()
```

Nije z gorega napomenuti da je bilo potrebno malo modificirati postojeći programski kôd kako bi proradio, jer je Adafruit repozitorij promijenjen od zadnje verzije `python-lis3dh` biblioteke. Promjene se odnose na datoteku `LIS3DH.py` iz biblioteke `python-lis3dh`:

- Linija koja se odnosi na učitavanje I²C podrške više nije `from Adafruit_I2C import Adafruit_I2C` jer je `Adafruit_I2C` verzija te biblioteke koja se više ne koristi. Sada je uputno koristiti biblioteku `Adafruit_GPIO.I2C`, pa taj *import* izgleda ovako: `import Adafruit_GPIO.I2C as I2C`. Nakon što su načinjene te promjene, u tijelu biblioteke još je potrebno izmijeniti svako pojavljivanje zastarjelog objekta `Adafruit_I2C` u aktualni `I2C.Device`.
- Konstruktor klase `LIS3DH` za pretpostavljenu oznaku I²C sabirnice koristi vrijednost `"-1"`, dok RPi ima samo sabirnice s oznakama `"0"` i `"1"`. Prema tome, ovisno o tome koja se od njih koristi (u nastavku rada je to sabirnica 1), potrebno je ili "hardkodirati" vrijednost sabirnice ili naći način za detekciju korištene sabirnice. Zbog vremenskih ograničenja i obujma ovog rada, prihvatljivo je rješenje bilo "hardkodirati" vrijednost `"1"`, pa tako zaglavlje konstruktora klase `LIS3DH` izgleda ovako:
`def __init__(self, address=0x18, bus=1, debug=False).`

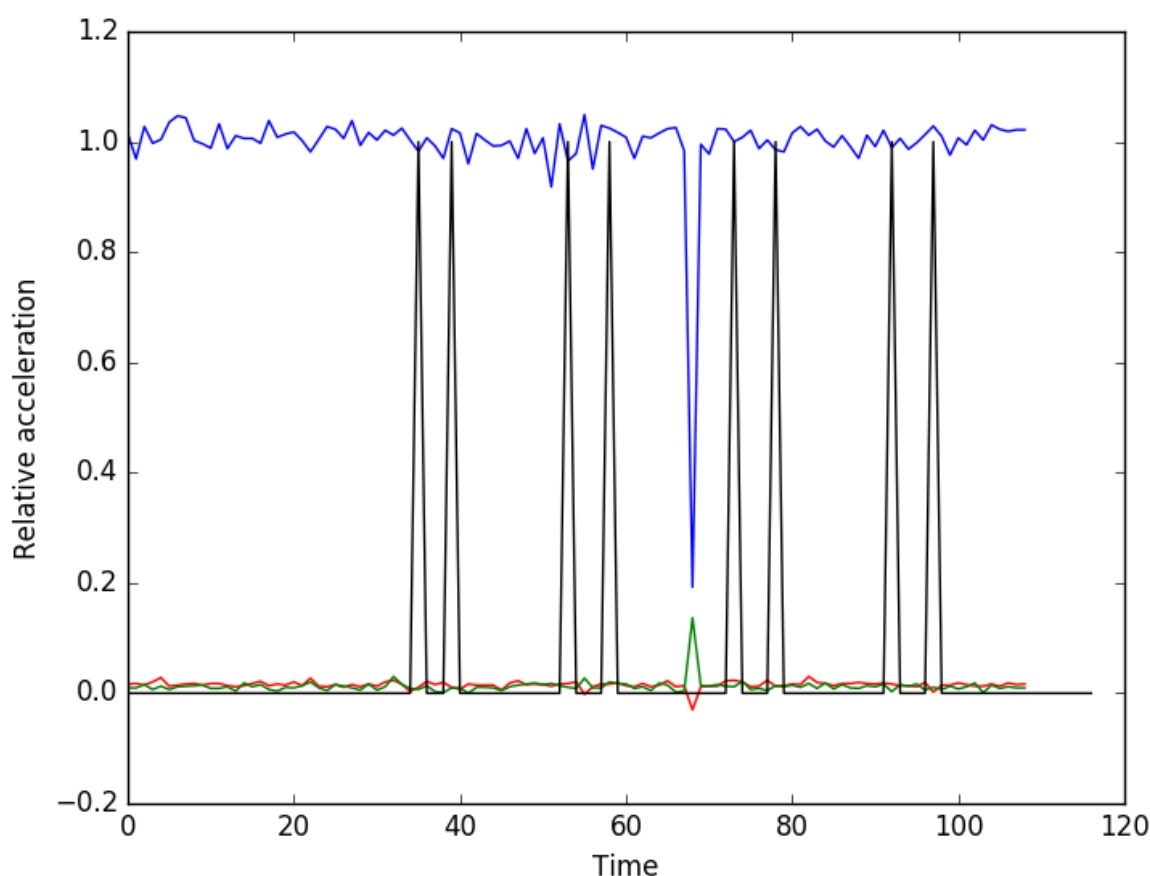
Nakon ovih preinaka, biblioteka je postala funkcionalna.

5.2 Očitani podaci

Preostaje nam još proučiti dobivene grafove. Na svim grafovima vrijedi sljedeća legenda:

- Crvena boja - X os akcelerometra,
- Zelena boja - Y os akcelerometra,
- Plava boja - Z os akcelerometra,
- Crna boja - detekcija kucanja.

Kucanje Prvo je na redu kucanje. Prikazano je grafom na Slici 4.

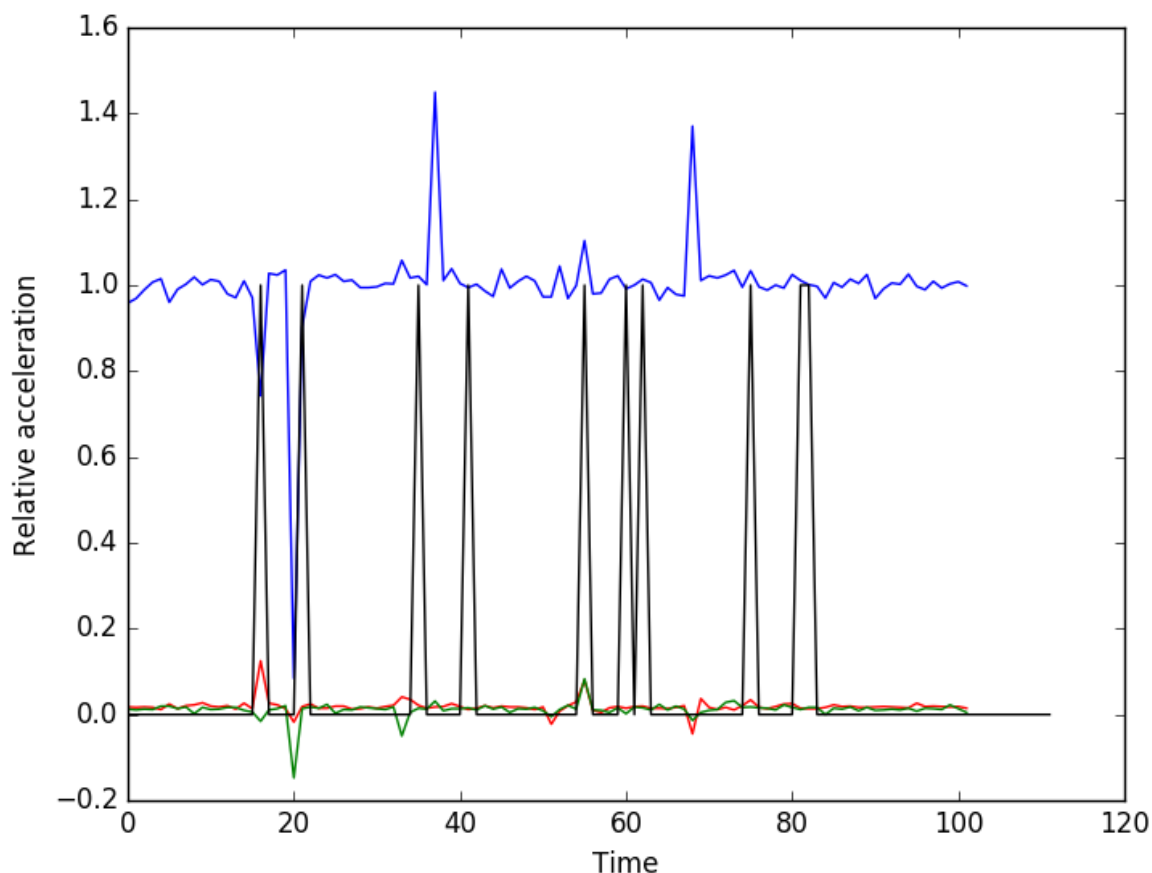


Slika 4: Kucanje

Odmah je očito da specijalizirana detekcija kucanja radi ispravno - vide se 4 para kucanja. Međutim, interesantno je da se, barem "na prvu", kucanje ne može intuitivno iščitati iz valnog oblika nijedne od krivulja. Z-os ima najveće oscilacije, što i ima smisla, jer je akcelerometar bio učvršćen vodoravno i prema gore (zato vrijednosti za Z os osciliraju oko 1, a ne oko 0 kao što je to slučaj za preostale dvije osi), pa je svaki udarac najviše utjecao upravo na nagib senzora.

Dobro je primijetiti i da su već u takvom, relativno slabo učvršćenom okruženju, šumovi i slabe okolne vibracije, dovoljno prigušeni za X i Y osi. Izvor artefakta na cca. 65. vremenskoj jedinici nije poznat, ali interesantno je primijetiti da ni takva velika anomalija nije uspjela "zbuniti" sustav za detekciju kucanja.

Udaranje Sljedeći graf koji promatramo jest udaranje. Prikazan je na Slici 5.

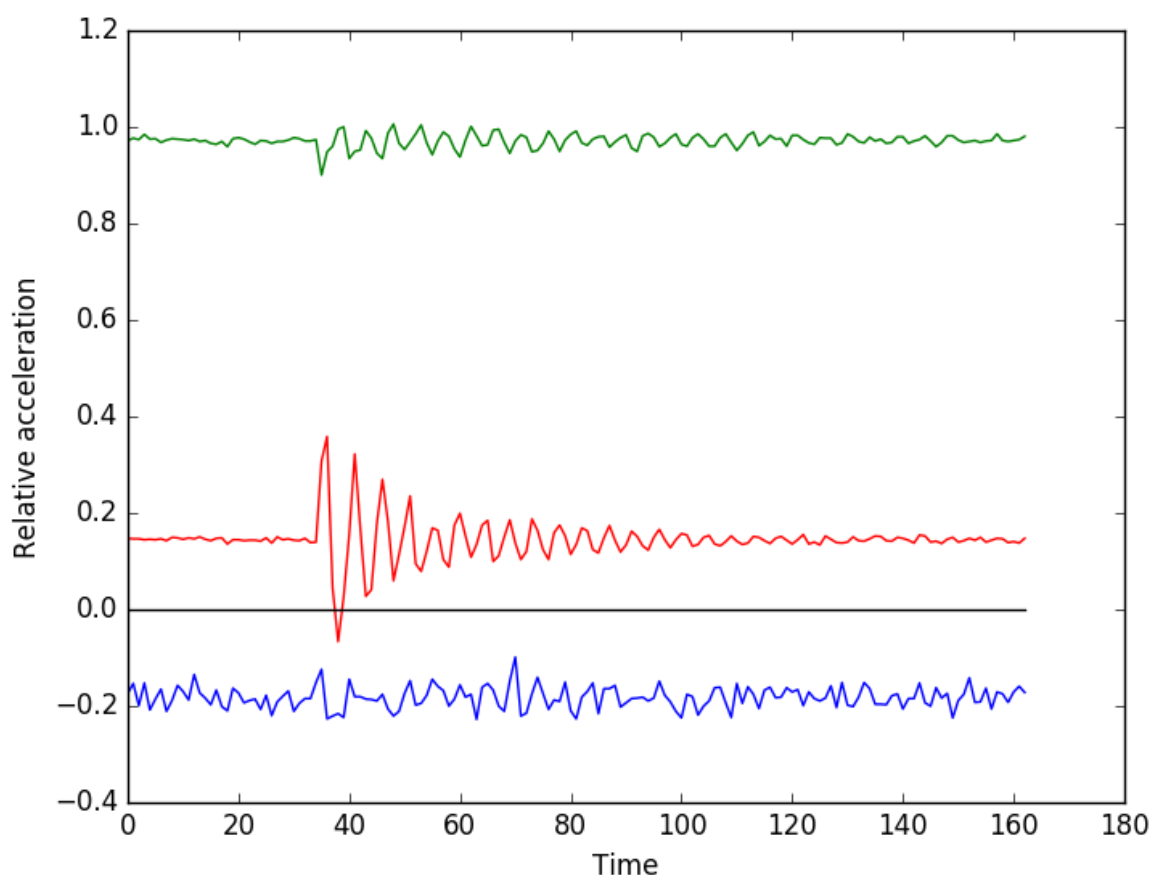


Slika 5: Udaranje

U ovom slučaju detekcija kucanja također uglavnom radi - prvi i drugi par udaraca izgledaju kako bi trebali izgledati. Međutim, na trećem je paru senzor očitao i jedan lažni udarac, dok je zadnji udarac izazvao za senzor toliko snažne vibracije da je nekoliko vremenskih jedinica bio postavljen prekidni signal, što rezultira valnim oblikom koji nema "špicu", nego traje neko vrijeme.

Također, na ovom se grafu već *nešto* može zaključiti i na temelju valnih oblika pojedinih osi. Ipak, očito i dalje takav pristup nije sasvim pouzdan. Možemo zaključiti da sklop snažne udarce ne podnosi najbolje što se tiče preciznosti mjerenja, ali to je možda i adekvatno za sustav gdje je poželjna visoka osjetljivost - u mirnim periodima sklop ne oscilira jako, pa bi se, uz propisno učvršćivanje i eliminaciju šuma, mogao jako dobro iskoristiti.

Prigušeno titranje Posljednji graf predstavlja "snimku" tona A odsviranom na električnoj bas gitari. Graf je prikazan na Slici 6.



Slika 6: Ton A

Čim pogledamo valni oblik na X-osi, očito da je senzor očitao nekakvo prigušeno titranje ugrubo stalne frekvencije. Na Z-osi i dalje vidimo najveći šum, iz istog razloga kao i ranije. Y-os, pak, opet ima prilično malen šum, pogotovo prije trenutka 40. Oko trenutka 40 je trznuta žica, te instrument počinje vibrirati - na X-osi vidimo da je riječ o stabilnoj frekvenciji i urednom prigušenju. Titranje je daleko najizraženije na X-osi zato što je pravac titranja žice skoro potpuno poklopljen s tom osi, pa se svaki titraj daleko najviše osjeti na dijelu sklopa zaduženom za očitavanje na toj osi.

Iz ovog eksperimenta možemo zaključiti da LIS3DH, iako je akcelerometar, a ne mikrofoni, pa vrlo vjerojatno nije optimiziran za stabilno očitavanje frekvencije, pokazuje sasvim pristojne i predvidive rezultate u okruženju s kontinuiranom vibracijom. Bilo bi interesantno ponoviti eksperiment i s mikrofonom MAX4466, pa usporediti rezultate, ali, nažalost, taj mikrofoni trenutno nije na raspolaganju.

6 Zaključak

7 Literatura

- Raspberry Pi službene stranice: <https://www.raspberrypi.org/>
- ADXL345: <https://learn.adafruit.com/adxl345-digital-accelerometer>
- MMA8451: <https://learn.adafruit.com/adafruit-mma8451-accelerometer-breakout/wiring-and-test?view=all>
- LIS3DH: <https://learn.adafruit.com/adafruit-lis3dh-triple-axis-accelerometer-breakout/downloads?view=all>
- Zvučni senzori: <https://www.sunfounder.com/learn/sensor-kit-v2-0-for-raspberry-pi-b-plus/lesson-19-sound-sensor-sensor-kit-v2-0-for-b-plus.html>