

# Mogućnost izrade pametnog teniskog reketa s Arduino platformom

---

**Karl, Domagoj**

**Undergraduate thesis / Završni rad**

**2018**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:547847>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-16**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni preddiplomski studij računarstva**

**MOGUĆNOST IZRADE PAMETNOG TENISKOG  
REKETA S ARDUINO PLATFORMOM**

**Završni rad**

**Domagoj Karl**

**Osijek, 2018.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 05.09.2018.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada**

Ime i prezime studenta:	Domagoj Karl
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3863, 19.09.2017.
OIB studenta:	02128722979
Mentor:	Doc.dr.sc. Tomislav Matić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Mogućnost izrade pametnog teniskog reketa s Arduino platformom
Znanstvena grana rada:	<b>Arhitektura računalnih sustava (zn. polje računarstvo)</b>
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	05.09.2018.
Datum potvrde ocjene Odbora:	12.09.2018.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**IZJAVA O ORIGINALNOSTI RADA**

Osijek, 17.09.2018.

Ime i prezime studenta:

Domagoj Karl

Studij:

Preddiplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

R3863, 19.09.2017.

Ephorus podudaranje [%]:

3%

Ovom izjavom izjavljujem da je rad pod nazivom: **Mogućnost izrade pametnog teniskog reketa s Arduino platformom**

izrađen pod vodstvom mentora Doc.dr.sc. Tomislav Matić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

## SADRŽAJ

<b>1. UVOD</b> .....	<b>1</b>
<b>1.1. Zadatak</b> .....	<b>1</b>
<b>2. PAMETNI TENISKI REKET</b> .....	<b>2</b>
<b>2.1. Tenis</b> .....	<b>2</b>
<b>2.2. Koncept pametnog reketa</b> .....	<b>2</b>
<b>2.3. Komercijalni primjeri pametnog reketa</b> .....	<b>3</b>
<b>2.3.1. Babolat</b> .....	<b>3</b>
<b>2.3.2. Sony</b> .....	<b>3</b>
<b>3. RAZVOJ PAMETNOG REKETA</b> .....	<b>4</b>
<b>3.1. Osnovni dijelovi pametnog reketa</b> .....	<b>4</b>
<b>3.1.1. Akcelerometar-žiroskop</b> .....	<b>4</b>
<b>3.1.2. Piezoelektrični senzor</b> .....	<b>5</b>
<b>3.1.3. Arduino</b> .....	<b>5</b>
<b>3.1.4. Bluetooth modul</b> .....	<b>5</b>
<b>3.1.5. Baterijski modul</b> .....	<b>6</b>
<b>3.2. Shema sklopovlja i spajanje</b> .....	<b>6</b>
<b>3.3. Programska podrška</b> .....	<b>7</b>
<b>4. TESTIRANJE I EKSPERIMENTALNI REZULTATI</b> .....	<b>13</b>
<b>4.1. Prikupljanje podataka o udarcima</b> .....	<b>13</b>
<b>4.2. Prikaz podataka</b> .....	<b>15</b>
<b>4.3. Analiziranje udaraca i stvaranje intervala</b> .....	<b>16</b>
<b>4.4. Eksperimentalni rezultati</b> .....	<b>18</b>
<b>5. ZAKLJUČAK</b> .....	<b>19</b>
<b>LITERATURA</b> .....	<b>20</b>
<b>SAŽETAK</b> .....	<b>21</b>
<b>ABSTRACT</b> .....	<b>22</b>
<b>ŽIVOTOPIS</b> .....	<b>23</b>
<b>PRILOZI</b> .....	<b>24</b>

# **1. UVOD**

U ovom radu se istražuju mogućnosti izrade pametnog teniskog reketa na Arduino platformi, što bi jedan takav reket trebao sadržavati te kako bi on trebao raditi. U drugom poglavlju opisano je kakav je tenis sport te zašto bi pametni teniski reket bio koristan, također, dani su postojeći komercijalni primjeri pametnog teniskog reketa. U trećem poglavlju opisuju se dijelovi koje jedan pametni reket treba sadržavati, kako oni rade te kako se oni koriste u primjeni za teniski reket. Prikazuje se kako je fizički napravljen pametni teniski reket i kako se programiraju dijelovi koji su korišteni. U četvrtom poglavlju prikazano je na koji način su prikupljeni podaci o udarcima te kako reket, na kraju, prepoznaje udarce te njegovu točnost prepoznavanja u usporedbi sa postojećim komercijalnim primjerima. Naposljetku, u petom poglavlju dana je analiza uspješnosti izrade pametnog teniskog reketa te moguća, buduća, poboljšanja samog rada.

## **1.1. Zadatak**

Opisati u radu što je to i kako radi pametni teniski reket. Istražiti mogućnosti izrade pametnog teniskog reketa s Arduino platformom.

## **2. PAMETNI TENISKI REKET**

Pametni teniski reket je reket koji se koristi kao i obični reket za igranje tenisa. Razlika u odnosu na obični reket je ta da pametni reket obrađuje, analizira i prepoznaje teniske udarce te ima mogućnost povezivanja s vanjskim uređajem na kojemu se prikazuju podaci, npr. mobilni telefon ili računalo. U idućim potpoglavljima objasnit će se osnovni pojmovi i pojave u tenisu te koncept pametnog reketa i postojeći primjeri.

### **2.1. Tenis**

Tenis je sport koji se igra reketom i lopticama na linijama označenoj podlozi te mrežom koja dijeli podlogu na dvije polovice na kojima se nalaze igrači. Tenis je prvi put igran između 1859. i 1865. godine u Velikoj Britaniji [1]. Teniska igra sadrži različite udarce. Za potrebe treninga i analize udaraca, jako je dobro imati uređaj koji može odrediti položaj i kretanju reketa u trenutku udarca. Zbog velike brzine udaraca jako je teško golim ljudskim okom odrediti greške u samim udarcima. Kamerom se može snimati osoba u igri, ali kasnije zahtjeva opsežan posao video-montaže i analize da bi se mogli odrediti tipovi udaraca te pogreške u udarcima. Pametni reket omogućava da se u stvarnom vremenu može odrediti tip udarca te njegove parametre.

U tenisu postoje četiri osnovna tipa udaraca: servis, forhend, bekend i volej (eng. serve, forehand, backhand, volley). Nadalje, svaki od njih se dijeli na podtipove: sječena, spin, ravna (eng. slice, spin, flat) [2]. Svi udarci imaju karakteristična svojstva. Na osnovu tih razlika, pametni reket može odrediti o kojem se udarcu radi te na koji je način on ispravno, odnosno loše izveden, u tom trenutku.

### **2.2. Koncept pametnog reketa**

Pametni reket fizički treba izgledati jednako, ili približno jednako, kao i neki drugi „obični“ teniski reket kako bi igrač koji igra njime zadržao iste karakteristike svoje igre. Unutar pametnog reketa nalaze se senzori te uređaj koji obrađuje podatke, reket još mora imati i mogućnost povezivanja s mobilnim uređajem ili računalom kako bi se prikazali podaci o udarcima korisniku reketa.

Senzori koji se nalaze u pametnom reketu služe da bi se odredio trenutak udarca te položaj i kretanju reketa u trenutku udarca. Uređaj koji obrađuje podatke senzora mora imati dovoljno veliku brzinu i mogućnost obrade podataka. Također, reket mora posjedovati i vlastiti izvor električne energije, bateriju, za napajanje uređaja koji se nalaze u njemu.

## **2.3. Komercijalni primjeri pametnog reketa**

Trenutno na tržištu postoji nekoliko proizvođača pametnih teniskih reketa. Neki od njih su napravljeni na način da je u tvorničkom obliku integrirana elektronika unutar reketa, dok su neki napravljeni kao dodatni uređaj koji se stavlja na dršku ili neki drugi dio reketa kao dodatak koji ima senzore, procesnu jedinicu i bateriju sadržanu u jednom, malom, uređaju.

### **2.3.1. Babolat**

Babolat je francuski proizvođač teniskih reketa i ostale popratne teniske i badmintonske opreme. 2014. godine na tržištu su prvi put prikazali pametni teniski reket [3]. To je bio fizički identičan model koji su prije proizvodili, ali je od te godine sadržavao elektroniku za prepoznavanje i analizu udaraca. Također, uz reket se dobiva i prikladna aplikacija za pametne mobitele s kojom se može spojiti na reket te tako pregledavati podatke o igri i udarcima [4].

Danas su u prodaji dostupni Pure Aero Play model reketa, koji sadrži elektroniku za prepoznavanje udaraca te „običan“ model pod nazivom Pure Aero, koji je po specifikacijama isti kao „pametni“ reket, ali ne sadrži elektroniku za prepoznavanje udaraca. Pametni reket se prodaje po cijeni od \$319 [5], dok se obični prodaje za \$219 [6].

### **2.3.2. Sony**

Iste godine kao i Babolat, Sony je na tržište izbacio „pametni senzor za tenis“, uređaj koji se kupovao odvojeno od reketa te se stavlja na kraj drške teniskog reketa. U sebi je sadržavao sve potrebne senzore, bateriju te ima mogućnost povezivanja s pametnim telefonom. Uz uređaj je dolazila i prikladna aplikacija preko koje se mogla vidjeti detaljna analiza udaraca.

Također, uz podatke o udarcima, reket je pružao i praćenje-uživo (eng. Live Mode Video) način rada, u kojemu se kroz aplikaciju snimao video igrača na kojemu su u stvarnom vremenu prikazivani podaci o udarcima, što je omogućilo detaljnu, interaktivnu, analizu igre [7].

2014. godine je predviđena cijena za ovaj uređaj bila oko \$175, danas ovaj uređaj više nije u prodaji [8].



### **3. RAZVOJ PAMETNOG REKETA**

U ovom poglavlju govorit će se o razvoju pametnog reketa. Ono se sastoji od razvoja sklopovlja i spajanje istog te odgovarajuće programske podrške. U idućim potpoglavljima navodit će se svi dijelovi od kojih se sastoji pametni reket te njihova funkcija u samom reketu. Također, bit će prikazan način spajanja te kako se razvija programska podrška.

#### **3.1. Osnovni dijelovi pametnog reketa**

Osnovni dijelovi od kojih se sastoji pametni reket su: senzori, procesna jedinica, Bluetooth modul i baterijski modul. Senzori se koriste za prikupljanje podataka o udarcima, procesna jedinica služi za obradu podataka koji se dobivaju od senzora, dok Bluetooth modul služi za povezivanje s vanjskom jedinicom, pametnim mobitelom ili računalom. Sve prethodno navedene uređaje napaja baterijski modul. Više informacija o pojedinim dijelovima opisano je u idućim potpoglavljima.

##### **3.1.1. Akcelerometar-žiroskop**

Akcelerometar je uređaj koji mjeri akceleraciju u danom trenutku te ju uspoređuje s konstantnom akceleracijom na Zemljinoj površini koja je uzrokovana gravitacijskom silom. Na taj način, akcelerometar otkriva pokrete, odnosno određuje promjenu brzine i položaja u prostoru [9].

Žiroskop je uređaj koji mjeri kutnu brzinu i određuje položaj u danom trenutku na način da uspoređuje promjenu kuta zakretanja u odnosu na referentnu točku. Žiroskop radi na principu zakona očuvanja kutnog momenta koji kaže da ukupni kutni moment zatvorenog sustava ostaje konstantan prije i nakon izmjene kutnog momenta između objekata koji se nalaze u sustavu [10].

GY-521 je modul koji je korišten u izradi pametnog reketa, a na njemu se nalazi MPU-6050 integrirani sklop. MPU-6050 sadrži tri-osni akcelerometar i tri-osni žiroskop u jednom integriranom sklopu jako male veličine. Povezuje se pomoću I<sup>2</sup>C (eng. Inter-integrated Circuit) komunikacijskog protokola sa procesnom jedinicom. Ovaj senzor se na pametnom reketu koristi kako bi se mogao odrediti položaj i kutna brzina reketa u udarcu i nalazi se u dršci reketa. Tehničke specifikacije navedenog integriranog sklopa nalaze se u tablici 3.1 [11].

Tab. 3.1: Tehničke specifikacije MPU-6050 integriranog sklopa.

Opseg žiroskopa:	Opseg akcelerometra:	Način povezivanja:	Operativni napon:	Operativna struja (intenzivno korištenje):	Veličina integriranog sklopa:	Veličina cjelokupnog modula:
±2000 °/s	±16 g	I <sup>2</sup> C	2.375V– 3.46V	3.8mA	4mm/4mm/ 0.9mm	21.2mm/16.4 mm/3.3mm

### 3.1.2. Piezoelektrični senzor

Piezoelektrični senzor je uređaj koji koristi piezoelektrični efekt. Promjenom svog oblika, stvara električnu struju te se na taj način mogu mjeriti vibracije, promjena pritiska, temperature, sile. Na pametnom reketu se koristi da bi se odredila vibracija na okviru reketa za vrijeme udarca, tj. koristi se za određivanje trenutka udarca reketa o lopticu. Ovaj senzor se nalazi zalijepljen na okvir reketa, te je povezan žicom sa procesnom jedinicom.

### 3.1.3. Arduino

Procesna jedinica korištena u pametnom reketu je Arduino Pro Micro koji sadrži ATmega32U4 integrirani sklop. Odabran je model nižeg operativnog napona (3.3 V) zbog podudarnosti s ostalim korištenim dijelovima. Navedena pločica je odabrana zbog male veličine te jednostavnog spajanja i programiranja s računalom. Arduino se nalazi u dršci reketa te je povezan žicama sa ostalim uređajima u reketu. Tehničke specifikacije odabrane procesne dostupne su u tablici 3.2 [12].

Tab. 3.2: Tehničke specifikacije Arduino Pro Micro pločice.

Operativni napon:	Operativna frekvencija:	Način povezivanja:	Broj A/D priključaka:	Broj digitalnih priključaka:	Veličina pločice:
3.3 V	8 MHz	Micro USB utor	9 (10-bitni)	12 (od kojih 5 podržava PWM)	33mm/18mm/4mm

### 3.1.4. Bluetooth modul

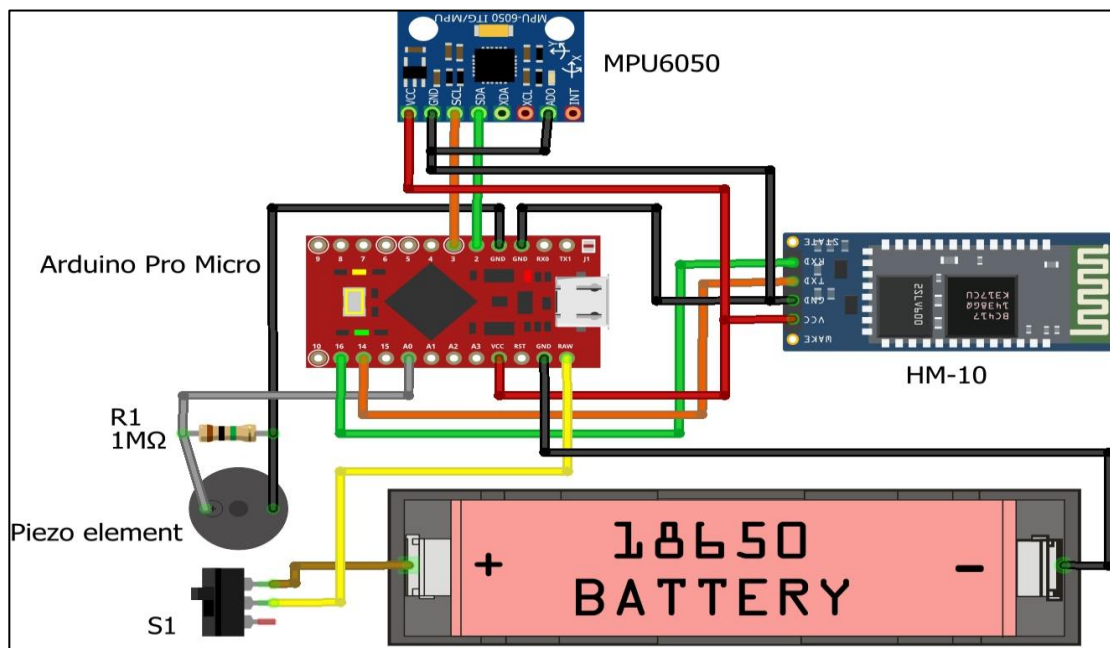
Bluetooth modul služi za povezivanje s vanjskim uređajem, u ovoj primjeni, s mobilnim uređajem te računalom. Koristi se kako bi se poslale informacije s procesne jedinice na vanjski uređaj za potrebe analize udaraca. U razvoju pametnog reketa korišten je HM-10 modul koji je BLE (eng. Bluetooth Low Energy) tipa, zbog niske potrošnje električne energije i jednostavnijeg povezivanja s novijim pametnim mobitelima. Odabrani modul radi na operativnom naponu od 3.3V kao i svi ostali dijelovi. Uređaj se nalazi u dršci reketa.

### 3.1.5. Baterijski modul

Baterijski modul korištena za napajanje svih uređaja u pametnom reketu je Litij-Ionska 18650 ćelija, nazivnog napona 3.7 V-4.2 V te kapaciteta od 2000 mAh. Nalazi se u dršci reketa zajedno sa ostalim dijelovima.

### 3.2. Shema sklopovlja i spajanje

Dijelovi koji se nalaze u pametnom reketu međusobno su spojeni žicama. Spajanje je vršeno prema shemi na slici 3.1 koja se nalazi ispod.



Sl. 3.1: Shema sklopovlja pametnog reketa izrađena Fritzing računalmim programom.

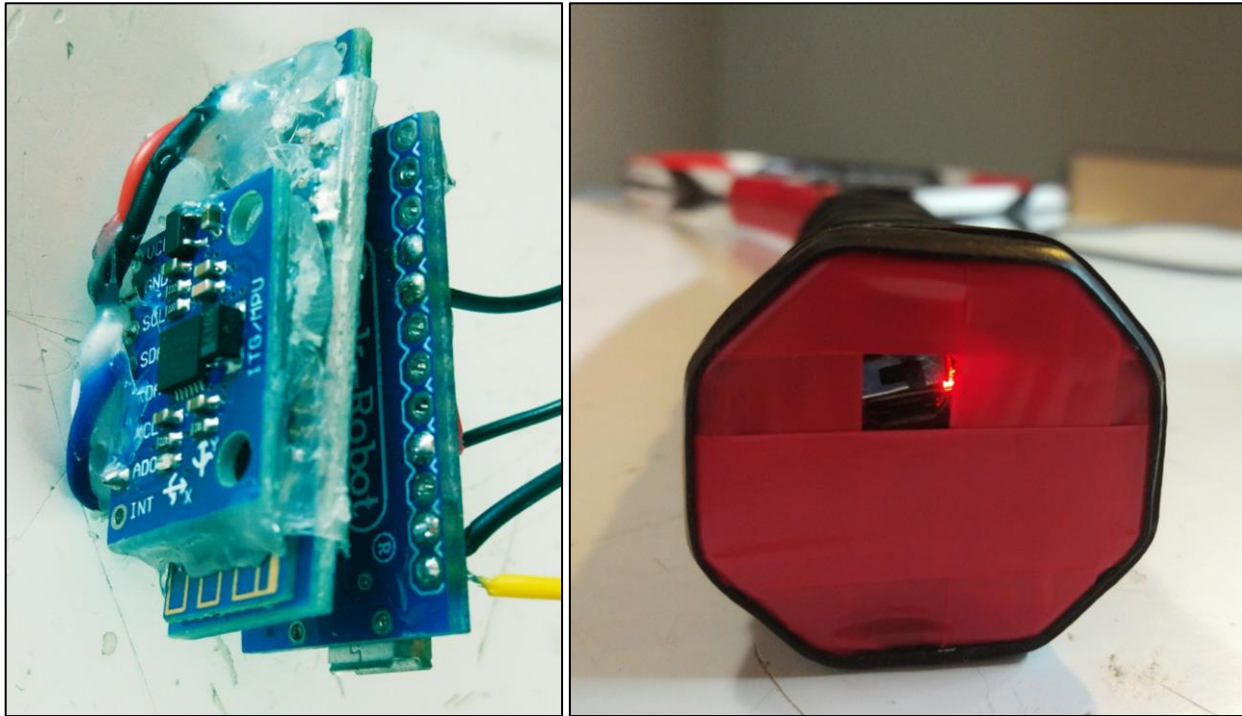
Na shemi sklopovlja, uz sve prethodno navedene dijelove, može se primijetiti otpornik koji služi da se nakon vibracije (udarca loptice u reket) na piezo elementu vrijednost napona na analognom priključku Arduina mogla vratiti na 0V, tj. spustiti na vrijednost prije vibracije. Kada bi se stavila mala vrijednost otpora (npr. 100  $\Omega$ ) taj napon bi se spustio na 0V jako brzo i Arduino ne bi stigao prepoznati napon sa piezo elementa – iz tog razloga odabrana je velika vrijednost otpora, od 1 M $\Omega$ , kako bi se usporila prijelazna pojava spuštavanja napona na analognom priključku Arduina.

Dodatno, uz prethodno navedene dijelove, sklopovlje sadrži i prekidač koji služi za jednostavno uključivanje sklopovlja, tj. mogućnost da se odspoji baterijski modul od sklopovlja.

Sklopovlje je ugrađeno u dršku reketa tako da zauzima najmanje moguće mjesta, posloženo je na način da se tiskane pločice nalaze jedna na drugoj, gdje je na vrhu Arduino (procesna jedinica), u sredini Bluetooth modul te akcelerometar-žiroskop na dnu. Tiskane pločice su međusobno

odvojene tekućom plastikom zbog izolacije. Piezoelektrični senzor se nalazi zalijepljen na okviru reketa te je spojen dvjema žicama, kroz malu rupu na dršci reketa, na Arduino.

Baterijski modul se nalazi iznad (dublje u dršci reketa) te je spojena žicama preko prekidača na Arduino. Fizički izgled spojenih dijelova te njihova pozicija na reketu prikazano je na slici 3.2.



Sl. 3.2: Prikaz sklopovlja pametnog reketa; lijevo-spoj tiskanih pločica; desno-donja strana drške reketa.

Na lijevoj slici se može vidjeti tri pločice zajedno spojene te sloj tekuće plastike između, kako bi se fizički odvojile jedna od druge, dok se na desnoj slici može vidjeti prekidač i USB utor, koji služe za paljenje sklopovlja te jednostavno spajanje sa računalom za potrebe programiranja sklopovlja.

### 3.3. Programska podrška

Za programiranje Arduina korišten je Arduino IDE (eng. Integrated Development Environment) program koji se koristi za programiranje svih Arduino pločica koje trenutno postoje. U daljnjem tekstu objasnit će se dijelovi kôda koji služe za spajanje s pojedinim senzorima koji se nalaze u reketu te dijelovi kôda za slanje informacija na vanjske uređaje korištenjem Bluetooth modula.

U prethodnim potpoglavljima navedeno je da modul sa akcelerometrom i žiroskopom koristi I<sup>2</sup>C komunikacijski protokol za povezivanje s procesnom jedinicom (Arduinom), za takvu vrstu komunikacije potrebna nam je „Wire.h“ biblioteka koja je uključena u standardne biblioteke Arduino IDE razvojne programske podrške [13].

Na slici 3.3 prikazan je kôd koji nam služi za početno postavljanje, tj. inicijalizaciju GY-521 modula, koji se zadano nalazi u „sleep“ načinu rada te ga je prije korištenja potrebno staviti u „power“ način rada. Također, potrebno je postaviti i opseg akcelerometra i žiroskopa koji će se koristiti.

**Linija Kôd**

```
1: void setupMPU()
2: {
3:     Wire.beginTransmission(0x68);
4:     Wire.write(0x6B);
5:     Wire.write(0b00000000); //power mode
6:     Wire.endTransmission();
7:
8:     Wire.beginTransmission(0x68);
9:     Wire.write(0x1B);
10:    Wire.write(0b00011000); //gyro scale: +/- 2000deg/s
11:    Wire.endTransmission();
12:
13:    Wire.beginTransmission(0x68);
14:    Wire.write(0x1C);
15:    Wire.write(0b00011000); //acceleration scale: +/- 16g
16:    Wire.endTransmission();
17: }
```

*Sl. 3.3: Kôd za uključivanje i postavljanje akcelerometar-žiroskop modula.*

Za komunikaciju s modulom koristeći I<sup>2</sup>C komunikacijski protokol prvo se mora koristiti „beginTransmission“ funkcija, u koju se pod argument navodi adresa I<sup>2</sup>C uređaja kojemu se šalje naredba. Nakon toga, koristi se „write“ funkcija u koju se pod argument navodi kojem se registru pristupa<sup>1</sup>. Nakon odabira registra u koji se zapisuje, opet se naredbom „write“ šalju vrijednosti koje se zapisuju u odabrani registar. Posljednje, koristi se „endTransmission“ funkcija koja obilježava kraj slanja naredbi I<sup>2</sup>C uređaju [14].

Na slici 3.4 prikazan je kôd koji služi za čitanje vrijednosti akcelerometra sa MPU-6050 integriranog sklopa. Slično kao i na prethodnom primjeru, prvo se navodi adresa uređaja koji se koristi te se nakon toga odabire registar koji je potreban. Nastavak je nešto drugačiji, jer se radi o čitanju vrijednosti iz registara, za razliku od prethodnog primjera gdje su se zapisivale vrijednosti u registre, koristi se „requestFrom“ funkcija koja se sastoji od dva argumenta, adrese I<sup>2</sup>C uređaja te veličina podatka koji se očekuje (u bajtovima). Prva dva bajta podatka koji se primaju je

---

<sup>1</sup> detaljan popis svih registara dostupan je u dokumentu tehničkih specifikacija proizvođača MPU-6050 integriranog sklopa [11]

vrijednost akceleracije X-osi, druga dva su vrijednost akceleracije Y-osi te na kraju, vrijednost akceleracije Z-osi.

**Linija Kôd**

```
1: inline void recordAccelRegisters(int j)
2: {
3:     Wire.beginTransaction(0x68);
4:     Wire.write(0x3B);
5:     Wire.endTransmission();
6:     Wire.requestFrom(0x68, 6);
7:     while(Wire.available() < 6);
8:     aX[j] = Wire.read() << 8 | Wire.read();
9:     aY[j] = Wire.read() << 8 | Wire.read();
10:    aZ[j] = Wire.read() << 8 | Wire.read();
11: }
```

*Sl. 3.4: Kôd za čitanje vrijednosti akcelerometra sa akcelerometar-žiroskop modula.*

Za čitanje vrijednosti sa žiroskopa je sličan kôd kao i za čitanje vrijednosti sa akcelerometra, razlika je jedino adresa registra sa kojeg se čita vrijednost te naziv varijable u koju se spremaju vrijednosti.

Na slici 3.5 prikazan je način čitanja vrijednosti vibracije sa piezo elementa koji je spojen na analogni priključak Arduina.

**Linija Kôd**

```
1: pinMode(piezoPin, INPUT); //inicijalizacija
2: piezo[i] = analogRead(piezoPin); //čitanje vrijednosti
```

*Sl. 3.5: Kôd za čitanje vrijednosti vibracije na piezo elementu.*

Za čitanje vrijednosti vibracije s piezo elementa korištena je funkcija „analogRead“ koja je ugrađena kao osnovna funkcija u Arduino biblioteku, ali zbog premale brzine čitanja vrijednosti sa A/D pretvornika koji se nalazi na pločici bilo je potrebno izmijeniti koeficijent frekvencijskog djelitelja za A/D pretvornik [14]. Koeficijent frekvencijskog djelitelja za A/D pretvornik je broj koji označava s koliko se dijeli osnovna operativna frekvencija (8 MHz) ATmega32U4 integriranog sklopa, signal podijeljene frekvencije nakon djelitelja ide u A/D pretvornik te se vrši očitavanje s analognog priključka.

Početno, tj. tvornički zadana, vrijednost koeficijenta iznosi 128, što uz potrebnih 13 ciklusa takta za pretvaranje, daje trajanje očitavanja vrijednosti na analognom priključku od 208  $\mu$ s (uzorkovanje od 4.8 kHz), dok trajanje očitavanja sa akcelerometar-žiroskop modula traje oko 40  $\mu$ s (uzorkovanje od oko 25 kHz) za sve osi. Vrijednosti vibracije u pametnom reketu potrebne su samo zbog prepoznavanja trenutka početne vibracije (kako bi se prepoznao trenutak udarca reketa o lopticu),

dok sam raspon te vrijednosti ne treba biti 10-bitna vrijednost (na Arduino pločici se nalazi 10-bitni pretvornik, a za trenutne potrebe dovoljna je 8-bitna vrijednost vibracije-raspon vrijednosti: 0-255). Povećanjem brzine očitavanja vibracije (smanjenjem koeficijenta frekvencijskog dijelitelja za A/D pretvornik na 8), smanjio se raspon na 8-bitnu vrijednost što je uvelike smanjilo vrijeme očitavanja, koje je spušteno na 13  $\mu$ s (uzorkovanje od 77 kHz), dok su dobivene vrijednosti vibracije ostale zadovoljavajuće točnosti za potrebe očitavanja vibracije s piezo elementa [15].

Kôd za mijenjanje koeficijenta frekvencijskog djelitelja za A/D pretvornik nalazi se ispod na slici 3.6.

#### **Linija Kôd**

```
1: #ifndef cbi
2: #define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
3: #endif
4: #ifndef sbi
5: #define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit))
6: #endif
7:
8: //u setup() funkciji:
9:     cbi(ADCSRA, ADPS2); //0
10:     sbi(ADCSRA, ADPS1); //1
11:     sbi(ADCSRA, ADPS0); //1
```

*Sl. 3.6: Kôd za mijenjanje koeficijenta frekvencijskog djelitelja.*

Kôd za mijenjanje koeficijenta je napisan kao makro funkcije za jednostavnije mijenjanje vrijednosti koeficijenta. Prema tehničkim specifikacijama korištenog, ATmega32U4, mikrokontrolera koji se nalazi na Arduino pločici, određene su vrijednosti koje trebaju biti upisane u „ADCSRA“ registar da bi se taj koeficijent frekvencijskog djelitelja za A/D pretvornik postavio na željenu vrijednost [16].

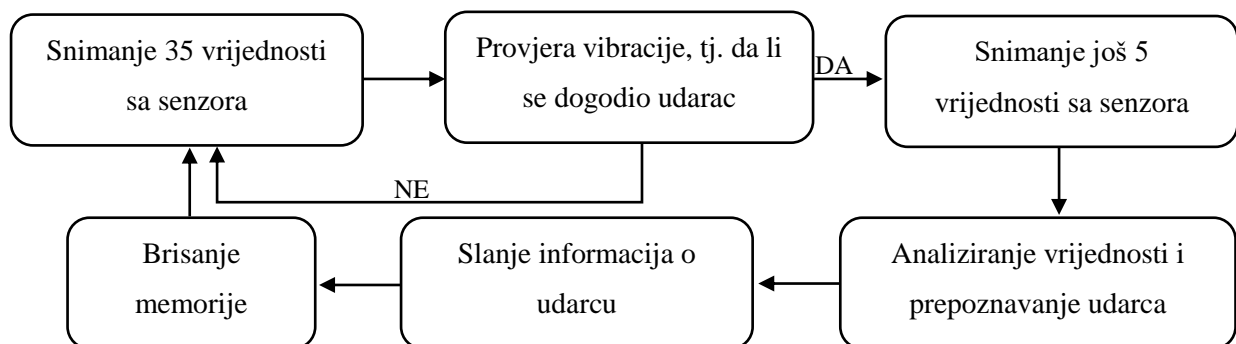
Za slanje podataka Bluetooth modulom korištena je „SoftwareSerial.h“ biblioteka, koja je kao i, prethodno korištena, „Wire.h“ uključena u standardne biblioteke Arduino IDE programskog sučelja. Odabrana biblioteka je korištena iz razloga da bi se moglo u isto vrijeme programirati Arduino preko računala i slati podaci na vanjski uređaj Bluetooth vezom. Na slici 3.7 prikazan je kôd koji nam služi za slanje poruke na uređaj spojen Bluetooth vezom.

## Linija Kôd

```
1: #include <SoftwareSerial.h> //BLE
2:   SoftwareSerial bleSerial(13,16); //TX,RX
3:
4:   //u setup() funkciji:
5:     bleSerial.begin(9600);
6:     bleSerial.println("Phone connected");
```

Sl. 3.7: Kôd za slanje poruke na uređaj spojen Bluetooth vezom.

Na slici 3.8 prikazan je dijagram toka glavnog dijela programa, „loop()“ funkcije, koja se neprestano ponavlja.



Sl. 3.8: Dijagram toka glavnog programa.

Na slici 3.8 može se vidjeti da se snimaju, ukupno, 40 vrijednosti sa senzora te se onda nakon toga vrši analiza udarca. Analiza udarca se zasniva na dvije vrijednosti, po tri osi, na akcelerometru i žiroskopu te četiri vrijednosti vibracije sa piezo elementa. Odabrana je vrijednost od 40 vrijednosti zbog povećane brzine te veće vjerojatnosti da će se udarac pronaći u zadanom rasponu.

Funkcija koja odrađuje analizu udarca je jako dugačka zbog puno uvjeta koje sadrži, ali je jednostavno napisana. Svaki udarac ima određene intervale vrijednosti senzora u kojima se nalazi, tako da funkcija analize udarca provjerava u kojim se intervalima nalaze vrijednosti senzora te na kraju vraća informaciju koji udarci su mogući s obzirom na dobivene vrijednosti sa senzora. Na slici 3.9, koja se nalazi na idućoj stranici, prikazana je skraćena funkcija analize udarca. Više o samom dobivanju intervala vrijednosti koje zadovoljavaju za određene udarce u idućem poglavlju.



**Linija**   **Kôd**

```
1: String analyzeHit()
2: {
3:     String output = "udr: ";
4:     if(//provjera intervala vrijednosti senzora)
5:         output += "fh-sp, "; //forehand spin
6:     //identično za sve ostale udarce
7:     if (output.length() < 7)
8:         output += "unk"; //nepoznat udarac (eng. unknown)
9:     else
10:        output.remove(output.length()-2); //brisanje zareza
11:
12:     return output;
13: }
```

*Sl. 3.9: Kôd za analizu udarca.*

Na slici iznad, mogu se vidjeti komentari kod provjere intervala vrijednosti senzora gdje nisu uvršteni stvarni uvjeti iz razloga što ih ima jako puno, cijeli kod u kojemu se nalaze svi intervali dostupan je kao prilog uz rad.

## **4. TESTIRANJE I EKSPERIMENTALNI REZULTATI**

Nakon fizičkog spajanja i pozicioniranja uređaja u reketu razvijena je programska podrška u kojoj se vršilo prikupljanje podataka sa senzora. Reket nakon udaraca šalje parametre senzora u trenutku udarca na mobilni telefon te su se tako prikupljali osnovni podaci o udarcima. Kasnije, promatrane su razlike između različitih udaraca te stvoreni intervali u kojima se nalaze parametri senzora za pojedine udarce. Ti intervali su činili uvjete za udarce te je napravljena analiza udaraca. Više o pojedinim procesima opisano je u idućim potpoglavljima.

Za primanje podataka s reketa na mobilnom uređaju korištena je „Serial Bluetooth Terminal“ aplikacija koja je dostupna preko Google Play usluge na uređajima s Android operativnim sustavom.

### **4.1. Prikupljanje podataka o udarcima**

Prikupljanje podataka o udarcima vršeno je na način da su se na mobilni uređaj slale vrijednosti sa senzora, dok su se na teniskom igralištu izvodili isti udarci više puta, prosječno 50 puta za svaki udarac. Na slici 4.1 prikazan je kôd za slanje vrijednosti sa senzora preko Bluetooth modula.

U kôdu su napisani komentari uz varijable kako bi se prepoznalo koje vrijednosti senzora se koriste. Varijable sa žiroskopa i akcelerometra su zapisane kao polje od tri vrijednosti, za koje svaka vrijednost u polju predstavlja jednu od osi, X,Y ili Z. Varijable za vibraciju su poredane po jakosti vibracije (vbrH) i po vremenskom slijedu (vbrF), početno od vibracije koja se prvi put pojavljuje, a da je iznad određenog praga (15) koji uzet kao vibracija koja se smatra da se radi o vibraciji udarca reketa o lopticu.

Nakon svakog ispucavanja istih udaraca (za svaki različiti udarac posebno), iz prethodno navedene aplikacije na mobilnom uređaju, izvezena je tekstualna datoteka iz koje su se kasnije zapisivali intervali svih varijabli koje su slične za određeni udarac, ali više o samom prepoznavanju udarca bit će objašnjeno u idućem potpoglavljju.

**Linija Kôd**

```
1: void printRawBLE(){
2:     bleSerial.print("br: "); //broj_udarca
3:     bleSerial.println(countStrokes);
4:     bleSerial.print("vbrH: "); //vibracija, po jačini
5:     for(int i = 0 ; i < 3 ; i++) {
6:         bleSerial.print(piezoH[i]);
7:         bleSerial.print("[");
8:         bleSerial.print(piezoHindex[i]);
9:         if(i == 2) bleSerial.println("]");
10:        else bleSerial.print("],");
11:    }
12:    bleSerial.print("vbrF: "); //vibracija, vremenski
13:    for(int i = 0 ; i < 3 ; i++) {
14:        bleSerial.print(piezo[indexOfFirst+i]);
15:        bleSerial.print("[");
16:        bleSerial.print(indexOfFirst+i);
17:        if(i == 2) bleSerial.println("]");
18:        else bleSerial.print("],");
19:    }
20:    bleSerial.print("rotF["); //rotacija: X,Y,Z
21:    bleSerial.print(indexOfFirst);
22:    bleSerial.print("]: ");
23:    for(int i = 0 ; i < 3 ; i++) {
24:        if(i == 2) bleSerial.println(r0[i]);
25:        else {
26:            bleSerial.print(r0[i]);
27:            bleSerial.print(",");
28:        }
29:    }
30:    bleSerial.print("aclF["); //akceleracija: X,Y,Z
31:    bleSerial.print(indexOfFirst);
32:    bleSerial.print("]: ");
33:    for(int i = 0 ; i < 3 ; i++) {
34:        if(i == 2) bleSerial.println(g0[i]);
35:        else {
36:            bleSerial.print(g0[i]);
37:            bleSerial.print(",");
38:        }
39:    }
40: }
```

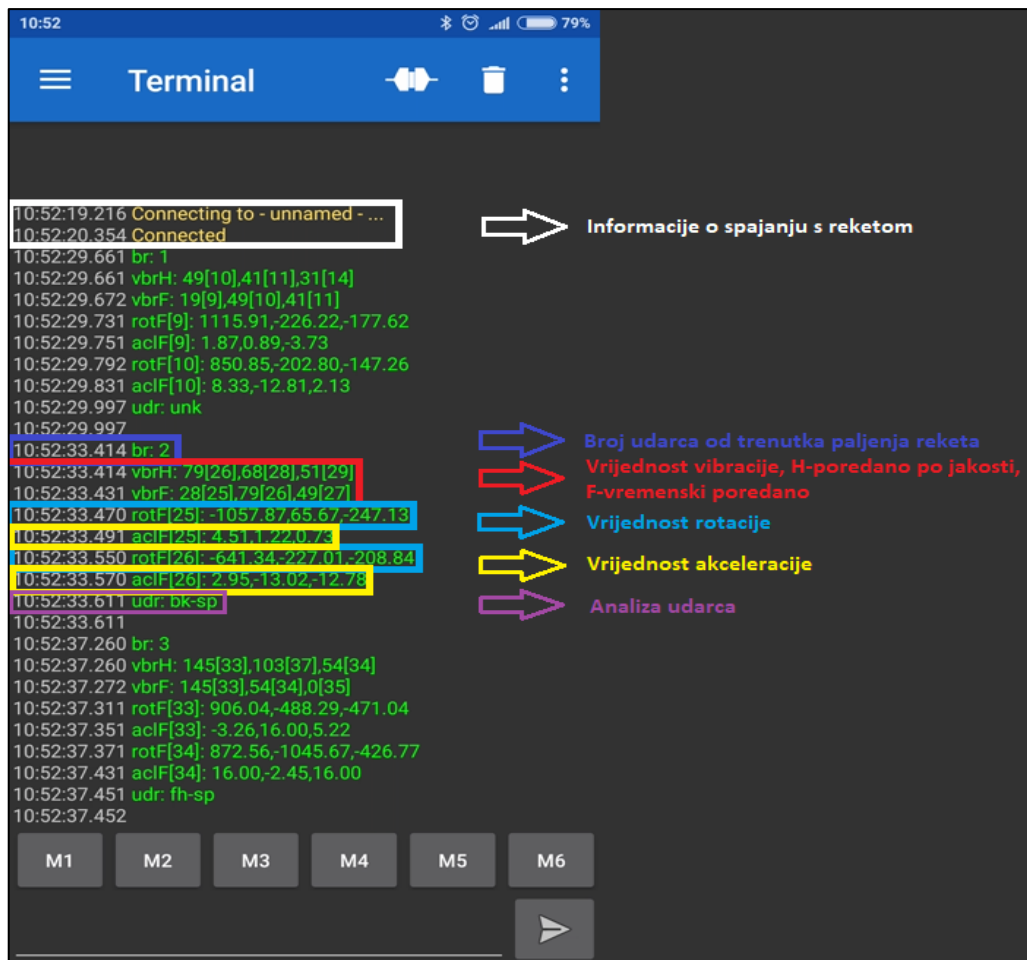
*Sl. 4.1: Kôd za slanje vrijednosti sa senzora.*

## 4.2. Prikaz podataka

Kao što je prethodno navedeno, s reketom se može spojiti preko Bluetooth Terminal aplikacije na mobilnom uređaju te onda reket nakon svakog udarca reket šalje informacije o udarcu. Poruka nakon svakog udarca sadrži ove podatke:

- broj udarca (trenutni broj udarca od trenutka paljenja pametnog reketa),
- parametre vibracije sa piezoelektričnog senzora,
- parametre sa akcelerometra i žiroskopa,
- ispis analize udarca (naziv udarca).

Na slici 4.5 nalazi se prikaz ekrana na kojemu su vidljivi prethodno navedeni podaci.

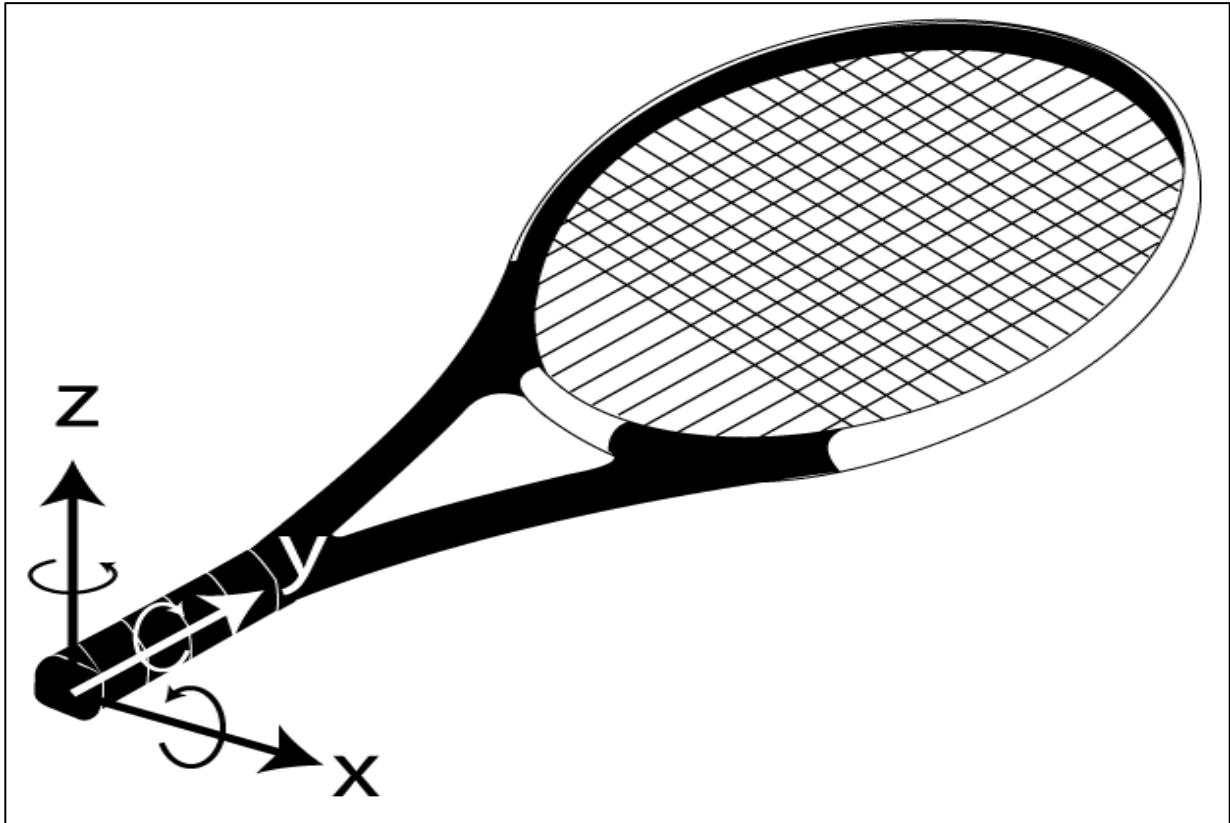


Sl. 4.5: Prikaz ekrana sa detaljima o podacima.

Na prikazu se može vidjeti da se ispred svake poruke nalazi vremenska oznaka kada je ona primljena. Može se primijetiti da je za cijelu poruku o udarcu, koja se nalazi u 8 manjih poruka, potrebno oko 200 milisekundi za primitak. Dok je, primjerice, vremenski razmak između udaraca oko 4 sekunde.

### 4.3. Analiziranje udaraca i stvaranje intervala

Na slici 4.2 prikazana je skica reketa sa koordinatnim sustavom, taj koordinatni sustav je okrenut isto kao što je i okrenut akcelerometar-žiroskop modul u reketu te na osnovu njega može se vidjeti smjerovi sila i rotacije koje bi se trebale očekivati za pojedine udarce.



Sl. 4.2: Prikaz koordinatnog sustava akcelerometar-žiroskop modula u odnosu na reket.

Prema prikazu sa slike iznad, određeni su smjerovi rotacije i sile koje se očekuju za pojedine udarce. Na slici 4.3 možemo vidjeti kako se razlikuju dva osnovna udarca, forhend i bekend.

10:52:56.276 br: 8	10:52:59.740 br: 9
10:52:56.276 vbrH: 108[11],78[13],73[12]	10:52:59.740 vbrH: 102[5],83[7],71[6]
10:52:56.292 vbrF: 108[11],73[12],78[13]	10:52:59.752 vbrF: 102[5],71[6],83[7]
10:52:56.398 rotF[11]: 1201.83,-132.44,-926.34	10:52:59.791 rotF[5]: -1375.18,167.26,-247.32
10:52:56.431 aclF[11]: -0.10,-2.90,-6.24	10:52:59.832 aclF[5]: 3.41,4.41,-6.08
10:52:56.432 rotF[12]: 587.44,13.48,-639.94	10:52:59.851 rotF[6]: -661.52,-338.90,-245.67
10:52:56.450 aclF[12]: 2.50,2.61,16.00	10:52:59.891 aclF[6]: -9.55,-11.48,-16.00
10:52:56.491 udr: fh-sp	10:52:59.953 udr: bk-sp
10:52:56.494	10:52:59.953

Sl. 4.3: Prikaz vrijednosti senzora za dva osnovna udarca.

Na lijevoj slici nalaze se vrijednosti senzora za forhend, a na desnoj vrijednosti za bekend. Može se vidjeti da vrijednost za rotaciju na X-osi su jako slične po vrijednosti, ali suprotnog predznaka.

Također, može se vidjeti da se za rotaciju na Y-osi suprotno izmjenjuju na vremenski različitim trenucima, za forhend ta vrijednost prelazi iz negativne u pozitivnu (prije i nakon udarca), dok je za bekind to suprotno. Za rotaciju na Z-osi vrijednosti istih predznaka, ali je vidljivo da je sama vrijednost rotacije veća kod forhend udarca, nego kod bekind, ta pojava se događa zbog samog igrača koji je igrao u tom trenutku, jer puno veću vrijednost rotacije postiže sa forhend udarcem.

Za vrijednost sile nešto je teže procijeniti za određene udarce, zbog velike osjetljivosti i velikih promjena, čak i u istom udarcu, ali se može vidjeti da je po iznosu svaka os u forhend udarcu veća nakon udarca od vrijednosti prije udarca, također, za bekind se može vidjeti suprotno, vrijednosti akceleracija na svim osima je manja nakon udarca od onih prije udarca.

Na slici 4.4 prikazane su dva različita ispisa vrijednosti senzora za isti udarac.

10:52:56.276 br: 8	10:53:10.109 br: 11
10:52:56.276 vbrH: 108[11],78[13],73[12]	10:53:10.109 vbrH: 190[32],172[33],135[36]
10:52:56.292 vbrF: 108[11],73[12],78[13]	10:53:10.114 vbrF: 71[31],190[32],172[33]
10:52:56.398 rotF[11]: 1201.83,-132.44,-926.34	10:53:10.171 rotF[31]: 1177.74,-377.20,-878.17
10:52:56.431 acIF[11]: -0.10,-2.90,-6.24	10:53:10.232 acIF[31]: -3.43,-5.32,-7.71
10:52:56.432 rotF[12]: 587.44,13.48,-639.94	10:53:10.233 rotF[32]: 1134.76,-393.60,-886.28
10:52:56.450 acIF[12]: 2.50,2.61,16.00	10:53:10.271 acIF[32]: -0.52,16.00,2.59
10:52:56.491 udr: fh-sp	10:53:10.292 udr: fh-sp
10:52:56.494	10:53:10.311

Sl. 4.4: Prikaz vrijednosti senzora za dva ista udarca.

Na prikazu se može vidjeti sličnost vrijednosti za isti udarac, na osnovu ispisa vrijednosti za iste udarce stvoreni su intervali. Ti intervali uključuju vrijednosti svih ispravnih udaraca, da bi se kasnije ti intervali koristili kao uvjeti za dobro prepoznavanje udaraca. Primjer, ako se vrijednost rotacije za X-os za forhend-spin udarac kreće između 900 °/s i 1400 °/s, za analizu udarca bit će postavljeni uvjeti za forhend-spin udarac da vrijednost rotacije za X-os mora biti veća od 900 °/s, a manja od 1400 °/s, na isti način uzimaju se uvjeti za sve vrijednosti senzora za svaki udarac posebno.

#### 4.4. Eksperimentalni rezultati

Za kraj, snimljeni su pokazni videozapisi na kojima se vidi snimka udaraca i snimka ekrana u stvarnom vremenu. Videozapisi se mogu pronaći kao prilog uz ovaj rad. U tablici 4.1 prikazane su vrijednosti točnosti prepoznavanja pojedinih udaraca u priloženim videozapisima.

Tab. 4.1: Vrijednosti točnosti prepoznavanja udaraca.

Forhend (spin/flat):	Forhend (slice):	Bekend (spin/flat):	Bekend (slice):	Volej forhend:	Volej bekend:	Servis flat	Servis slice
54/61 (89%)	10/13 (77%)	31/36 (86%)	14/14 (100%)	24/36 (67%)	46/52 (88%)	11/12 (92%)	30/32 (94%)

U tablici su navedeni tipovi udaraca koji su uzeti za testiranje te omjer ispravno prepoznatih i ukupnih udaraca navedenog tipa udarca te njihova točnost prepoznavanja. Ukupna točnost prepoznavanja svih tipova udaraca iznosi 86% na uzorku od 256 udaraca. Ovakva točnost prepoznavanja udaraca je približno jednaka točnosti prepoznavanja komercijalnih sustava<sup>2</sup>. Tek nešto veću točnost komercijalni primjeri postižu na osnovnim udarcima (forhend i bekind spin/flat/slice), dok na volejima postižu dosta manju točnost prepoznavanja ili nemaju mogućnost prepoznavanja voleja. Točnost prepoznavanja servisa je visoka i slična kao i kod komercijalnih primjere te su ukupni rezultati i točnost prepoznavanja udaraca na zadovoljavajućoj razini.

---

<sup>2</sup> Za usporedbu je korišten test mrežnog časopisa [17]

## 5. ZAKLJUČAK

Cilj ovog rada bio je istražiti mogućnosti izrade pametnog teniskog reketa na Arduino platformi, taj cilj je ispunjen kroz samu izradu pametnog teniskog reketa, pri čemu se prikazalo što svaki pametni teniski reket treba sadržavati te kako zapravo pametni reket treba raditi.

Prikazani su načini spajanja dijelova sa Arduinoom te koja se sva programska ograničenja moraju izmijeniti da bi pametni reket radio kako je zamišljeno. Kroz izradu reketa prikupljeni su podaci o različitim teniskim udarcima koji su kasnije služili za analizu udaraca.

Na kraju, dana je usporedba sa postojećim komercijalnim primjerima u kojoj je vidljivo da je moguće postići zadovoljavajuće rezultate uz korištenje Arduino platforme u pametnom teniskom reketu.

Problem kod izrade ovakvog pametnog reketa je korišten baterijski modul, koji je odabran zbog svoje cijene i karakteristika koje pruža, ali je mana njegova masa, koja uvelike mijenja fizičku karakteristiku reketa, jer reket postaje masivniji i tromiji za igru.

Također, postoje još mogućnosti za napredak cijelog sustava pametnog reketa, a one obuhvaćaju izradu mobilne aplikacije, koja bi detaljnije radila analizu teniske igre (kroz praćenje vremenskih intervala između udaraca i samog načina igre pojedinog igrača) te na taj način pružila poboljšanje u teniskoj igri korisnika pametnog teniskog reketa.



## LITERATURA

- [1] Tennis, Wikipedia, dostupno na: <https://en.wikipedia.org/wiki/Tennis> [21.7.2018.]
- [2] T. Hoskins–Burney, L. Carrington, Tennis Drill Book–2nd Edition, Human Kinetics, 2014.
- [3] History, Babolat, dostupno na: <https://www.babolat.us/history> [21.7.2018.]
- [4] Dusan Johnson, Review: Babolat Play Connected – the first smart racket, Gadgets & Wearables, 3.3.2015., dostupno na: <http://gadgetsandwearables.com/2015/03/03/babolat-play-connected-racquet/> [21.7.2018.]
- [5] Babolat Pure Aero Play Tennis Racquet, Amazon, dostupno na: <http://a.co/i113B4K> [21.7.2018.]
- [6] Babolat Pure Aero Racquets, Amazon, dostupno na: <http://a.co/5HWIDw3> [21.7.2018.]
- [7] Smart Tennis Sensor for Tennis Rackets, Sony, dostupno na: <https://www.sony.co.uk/electronics/smart-devices/sse-tn1w> [21.7.2018.]
- [8] Sam Byford, Sony's \$175 Smart Tennis Sensor will start tracking swings in May, The Verge, 20.1.2014., dostupno na: <https://www.theverge.com/2014/1/20/5326558/sony-smart-tennis-sensor-price-and-availability> [21.7.2018.]
- [9] Accelerometer, Wikipedia, dostupno na: <https://en.wikipedia.org/wiki/Accelerometer> [26.7.2018.]
- [10] Gyroscope, Wikipedia, dostupno na: <https://en.wikipedia.org/wiki/Gyroscope> [26.7.2018.]
- [11] MPU-6050, InvenSense, dostupno na: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/> [24.7.2018.]
- [12] Pro Micro – 3.3V/8MHz, sparkfun, dostupno na: <https://www.sparkfun.com/products/12587> [24.7.2018.]
- [13] M. McRoberts, Beginning Arduino: Second edition (Technology in Action), Apress, 2013.
- [14] M. Margolis, Arduino Cookbook, 2nd Edition, O'Reilly Media, 2011.
- [15] Fast sampling from analog input, Yet Another Arduino Blog, 1.2.2015., dostupno na: <http://yaab-arduino.blogspot.com/2015/02/fast-sampling-from-analog-input.html?m=1> [3.8.2018.]
- [16] ATmega32U4, Microchip, dostupno na: <https://www.microchip.com/wwwproducts/en/ATmega32U4> [3.8.2018.]
- [17] Sensor Accuracy Challenge, Gadget Tennis, 9.6.2016., dostupno na: <http://gadgettennis.com/sensor-accuracy-challenge/> [5.8.2018.]

## SAŽETAK

U ovom radu istraživale su se mogućnosti izrade pametnog teniskog reketa baziranog na Arduino platformi te je napravljen jedan takav pametni teniski reket modificiranjem običnog teniskog reketa. Za izradu reketa korišten je Arduino Pro Micro mikro upravljač, koji je programiran pomoću Arduino IDE razvojnog programskog okruženja. Uz Arduino u reketu se nalaze senzori: akcelerometar, žiroskop i piezo element. Također, reket sadrži i Bluetooth modul koji se koristi za spajanje s vanjskim uređajem (računalom, mobilnim uređajem) za prikazivanje podataka. Programski kôd Arduina se sastoji od tri glavna dijela: 1. postavljanje osnovnih parametara senzora i mikro upravljača; 2. (prije udarca) snimanje vrijednosti senzora; 3. (nakon udarca) analiziranje vrijednosti udarca te slanje podataka na vanjski uređaj. Drugi i treći dio se stalno ponavljaju. Pametni reket je testiran i radi kako je i zamišljeno, prepoznaje udarce i šalje podatke na vanjski uređaj.

Ključne riječi: Arduino, tenis, reket, akcelerometar, žiroskop, piezoelektrični efekt, Bluetooth

## **ABSTRACT**

### Smart Tennis Racquet Development Based on Arduino Platform

In this bachelor paper the possibilities of making a smart tennis racquet based on Arduino platform were explored. A prototype of a smart racquet described in this paper was made and an Arduino Pro Micro microcontroller was used. The microcontroller was programmed using Arduino IDE. With Arduino, three sensors were used: accelerometer, gyroscope and piezoelectric element; also, the racquet contains a Bluetooth module to communicate with external devices (a computer or a mobile phone) for displaying data. Program code of Arduino can be divided in three parts: 1. setting the parameters of sensors and microcontroller; 2. (before hit) recording sensor values; 3. (after hit) analysing the values and sending data to external device. Second and third part are constantly repeated. The smart racquet developed for the purposes of this paper was tested and the results show that the racquet works as it was intended - it recognizes the momentums of tennis ball hits and sends data to external device.

Keywords: Arduino, tennis, racquet, accelerometer, gyroscope, piezoelectricity, Bluetooth

## ŽIVOTOPIS

Domagoj Karl, rođen 16. kolovoza 1996. godine u Virovitici, Hrvatska. Svoj život provodi u Slatini gdje pohađa osnovno obrazovanje u Osnovnoj školi Eugena Kumičića Slatina do 2011. godine. Iste godine upisuje Opću Gimnaziju Slatina koju završava 2015. godine, te godine upisuje smjer računarstvo na tadašnjem Elektrotehničkom Fakultetu Osijek, danas Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek.

Tijekom cijelog trajanja obrazovanja postiže odlične rezultate na školskim natjecanjima u znanju i sportu te je od 2017. godine dobitnik stipendije Zaklade Novo Sutra za darovite studente u deficitarnim zanimanjima. Svoje radno iskustvo stekao je poslom u Avantcar d.o.o. rent-a-caru u Zagrebu kroz pripremu vozila za najam te predaji vozila klijentima.

Potpis: \_\_\_\_\_

## **PRILOZI**

CD s:

- Radom u „docx“ i „pdf“ formatu,
- Programskim kôdom Arduina u „txt“ formatu,
- Pokaznim videozapisima korištenja pametnog reketa.