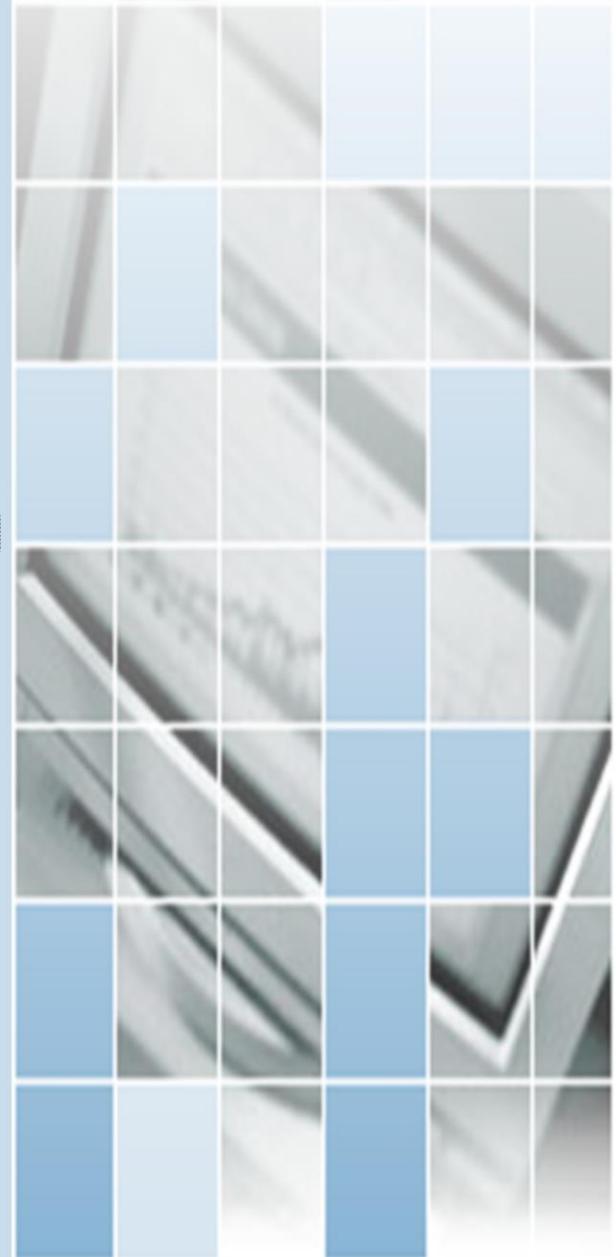


Upravljanje rasvjetom putem visokofrekventnog zvuka

Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za električne sustave i obradu informacija
Sveučilište u Zagrebu



- △ Za sve ukućane, posebno onima s poteškoćama u kretanju
- △ Programiranje Arduina, izrada Android aplikacije, obrada zvučnih signala
- △ Komunikacija na visokim frekvencijama

Sažetak

Pametna rasvjeta je jedan od *assistant living* projekata razvijen s ciljem da svim ukućanima olakša upravljanje osvjetljenjem, posebice onima s poteškoćama u kretanju. U ovom projektu omogućuje se regulacija osvjetljenja u boravišnoj prostoriji korištenjem pametnog telefona. Klasične realizacije pametne rasvjete podrazumijevaju povezivanje uređaja putem Interneta ili Bluetooth-a. Temeljna ideja ovog projekta je ostvariti bežični sustav koji koristi visokofrekventni zvuk za upravljanje rasvjetom, čime bi znatno smanjili potrošnju energije. Nedostaci provedene implementacije su ograničenja pojedinih komponenata sustava.

Sadržaj

1. UVOD	3
2. OPIS SUSTAVA	4
2.1. Arduino Uno	4
2.2. Pametni telefon Samsung Galaxy S7	5
2.3. Senzor zvuka VM309	7
2.4. 5VDC relē VMA406	8
2.5. Žarulja.....	8
2.6. Zvučnik.....	10
3. REALIZACIJA SUSTAVA.....	11
4. PROGRAMSKA PODRŠKA.....	12
4.1. Odašiljač.....	12
4.2. Prijemnik	16
4.2.1. Povećanje frekvencije uzorkovanja	16
4.2.2. Goertzelov algoritam.....	18
5. TESTIRANJE SUSTAVA.....	21
ZAKLJUČAK	23
6. LITERATURA.....	24
7. POJMOVNIK	25
8. RASPODJELA ZADATAKA.....	26

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sisteme i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uвijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

1. Uvod

Često nam je iritantno i zamorno ugrožavati svoju komforntost zbog želje za regulacijom svjetla. Tijekom boravka u određenoj prostoriji, i obavljanja svakodnevnih aktivnosti poput gledanja TV-a, odmaranja ili učenja ponekad nam je jako naporno prekinuti te radnje u svrhu uključivanja ili isključivanja svjetla. Umjesto odlaska do prekidača, ovaj projekt omogućuje nam uključivanje/isključivanje svjetla korištenjem pametnog telefona koji je u današnje doba ionako uvijek uz nas.

Sustav podržava dva načina rada, čiji se odabir vrši putem Android aplikacije „Pametna rasvjeta“ na pametnom telefonu. Prvi način rada omogućuje trenutnu promjenu stanja svjetla. Korisnik bira želi li u prostoriji u kojoj boravi imati uključeno ili isključeno svjetlo. Drugi način rada korisniku omogućuje uključenu rasvetu sve dok se nalazi u toj prostoriji te gašenje rasvjete pri izlasku, bez potrebe za njegovim djelovanjem.

U ovoj implementaciji komunikacija između pametnog telefona i mikrokontrolera koji upravlja rasvjetom odvija se putem visokofrekventnog zvuka. Na taj način postignuta je znatna ušteda energije i produžen životni vijek baterije na mobitelu, što je danas jako važno svim korisnicima. Za ostvarivanje komunikacije potrebni su samo zvučnik i mikrofon, što omogućuje implementaciju i na druge sustave unutar pametne kuće. Međutim, kvaliteta opreme znatno utječe na doseg komunikacije koji je moguće ostvariti te na visinu frekvencije koja se može postići.

2. Opis sustava

Sustav za upravljanje rasvjetom zasniva se na komunikaciji između pametnog telefona i mikrokontrolera putem zvuka visoke frekvencije. Za ostvarivanje sustava korišteni su:

- Arduino Uno
- Pametni telefon Samsung Galaxy S7
- Senzor zvuka VMA309
- 5VDC relej VMA406
- Žarulja 25W
- Zvučnik

2.1. Arduino Uno



Slika 1 Arduino Uno

Arduino je električka prototipna platforma namijenjena kreiranju električkih projekata. Sastoji se od hardware dijela koji je zapravo fizički električki programabilni strujni krug (poznat kao i mikrokontroler) i software dijela koji se naziva IDE (Integrated Development Environment). IDE se pokreće na računalu i služi za programiranje odnosno upravljanje samom pločicom. U okviru ovog projekta korištena je Arduino Uno razvojna platforma otvorenog tipa koja se temelji na 8 bitnom Atmega328P mikrokontroleru.

Glavne značajke:

- 14 digitalnih ulazno/izlaznih priključaka
- 6 analognih ulaznih priključaka
- Glavni izvor takta : kristalni oscilator od 16 MHz-a
- Flash memorija: 32KB
- SRAM : 2KB
- EEPROM : 1KB

Sadrži i tipku za reset, priključak za USB, priključak za napajanje i ICSP, a preporučuje se napajanje od 7-12 V ili preko USB priključka spojenog na računalo

2.2. Pametni telefon Samsung Galaxy S7



Slika 2 Samsung Galaxy S7

Samsung Galaxy S7 koristi Android 6.0 platformu koja se može nadograditi na 8.0 verziju koju koristimo u ovoj implementaciji. Ekran je veličine 5.1 inča, a interna memorija 32GB. Sadrži Li-Ion bateriju od 3000mAh koja je ugrađena u mobitel i nije dostupna korisniku. Mikrofon prima zvuk i do 192kHz, uz aktivno uklanjanje buke. Za reproduciranje zvuka iz mobitela, poput glazbe ili tonova koji će se koristiti u projektu „Pametna rasvjeta“, baterija može trajati i do 60h. Podržava .mp3 i .wav datoteke, a sadrži zvučnik i 3.5mm jack na koji se mogu priključiti vanjski zvučnik ili slušalice.

Za razliku od većine drugih mobitela, kod Samsung Galaxy-a je uz izlaz nadograđeno aktivno vanjsko pojačalo. Zbog toga nastaje veoma čist zvuk s većom glasnoćom. Iako je stereo preslušavanje povećano u odnosu

Upravljanje rasvjetom putem visokofrekventnog zvuka

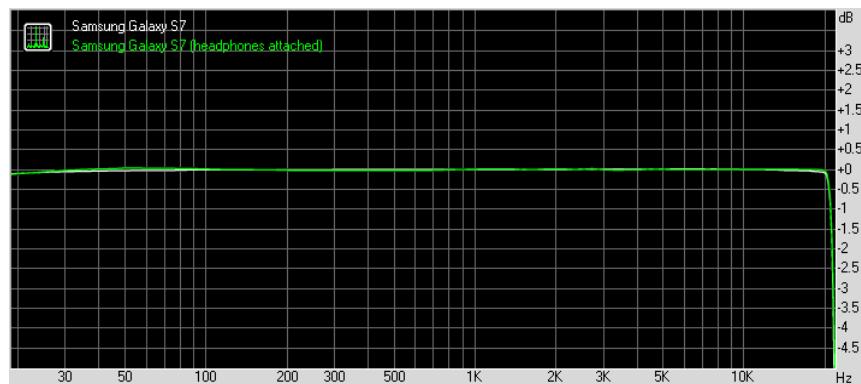


na druge pametne telefone obitelji Samsung, koristeći slušalice zvuk i dalje ostaje veoma dobar. U nastavku je dana tablica s testiranjima zvučnika i slušalica.

Tablica 1 Karakteristike zvuka kod Samsung Galaxy S

Test	Frekvenčijski odziv	Razina šuma	Dinamički opseg	THD	IMD + Šum	Stereo preslušavanje
Samsung Galaxy S7	+0.01, -0.04	-92.5	92.6	0.0027	0.0078	-92.7
Samsung Galaxy S7 (slušalice)	+0.05, -0.05	-91.9	92.1	0.0044	0.063	-82.4

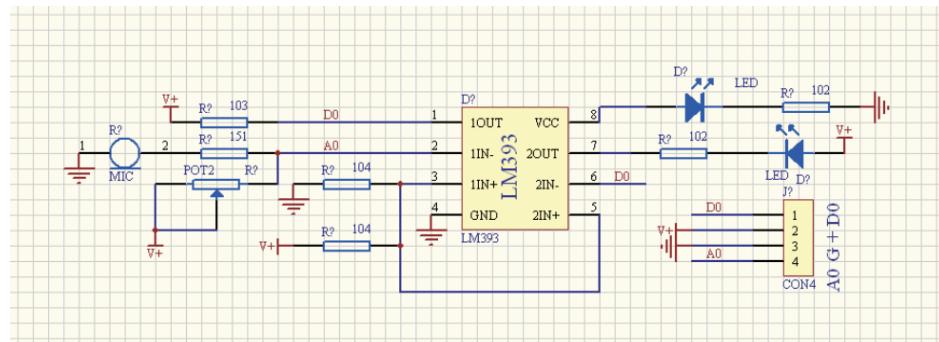
Sljedeći graf prikazuje ovisnost izlaza zvučnika o frekvenciji koja se s mobitela šalje. Može se vidjeti da karakteristika zvuka nakon 15kHz naglo pada, što označava veoma tih izlaz iz zvučnika pri radu s visokim frekvencijama, odnosno onima višim od 15kHz, što će biti jedna od glavnih prepreka u implementaciji sustava „Pametna rasvjeta“.



Slika 3 Samsung Galaxy S7 - frekvenčijski odziv

2.3. Senzor zvuka VM309

Senzor zvuka s mikrofonom VMA309 omogućuje prijem zvučnih signala u frekvencijskom rasponu od 50 Hz do 20 kHz. Na tiskanoj pločici zajedno s mikrofonom nalazi se i dodatno sklopovlje koje obrađuje signal za rad s mikrokontrolerom. Na sljedećoj slici vidljiv je shematski prikaz VMA309 senzora.



Slika 4 VMA309 shema

Signal s mikrofona se dovodi na dva izlaza. Jedan od njih je analogni i provodi onaj signal koji je dobiven na mikrofonu. Drugi izlaz je digitalni i prije dovođenja signala na izlaz, signal s mikrofona se prvo dovodi na otporničko dijelilo koje je podesivo s trimerom. Optimalan otpor trimera je dobiven eksperimentalno i iznosi $8.9\text{ k}\Omega$. Potom se signal s dijelila dovodi na komparator i ako pređe fiksnu granicu od 2.5V, komparator postavlja digitalni izlaz u visoku razinu.

Na slici 3 je prikazan korišteni sklop s mikrofonom. Vrijednost signala na digitalnom izlazu se prikazuje na LED diodi na sklopu pod imenom LED2 tako da se jednostavno može odrediti stanje digitalnog izlaza.



Slika 5 VMA309

Značajke modula :

- napajanje: 3.3 do 5 VDC
- izlazni priključci AO i DO
- 2 LED indikatora: indikator napajanja i indikator izlaza komparatora •
osjetljivost: 48 - 66 dB
- temperature rada: -40 to +85°C

2.4. 5VDC relej VMA406

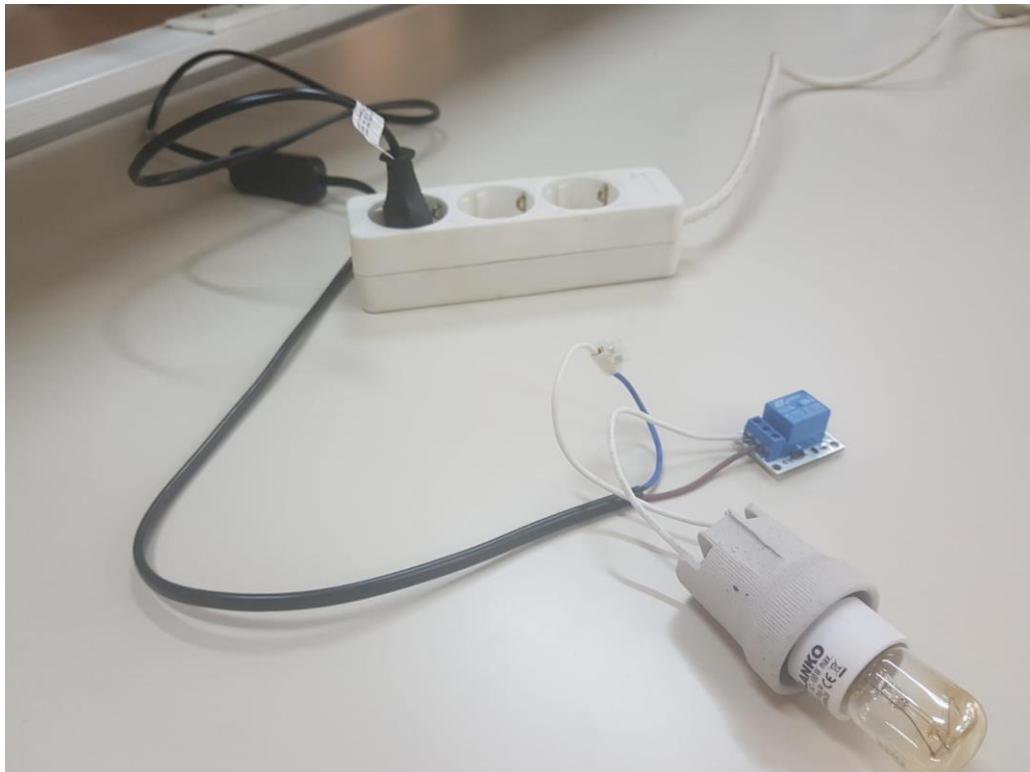


Slika 6 5VDC relej

Modul s relejem omogućava kontrolu trošila koju trebaju veliku struju ili veliki napon, bez obzira rade li na istosmjernom ili izmjeničnom naponu. Relej VMA406 je Arduino kompatibilni relej koji radi pri naponu napajanja od 5V. Dimenzije releja su 40 x 27 x 18 mm. Pri 250VAC mjera struje je 10A, odnosno pri 30VDC 10A (nevodljivo). Postoje tri priključka i tri kontakta releja. Priključci su spojeni na GND, 5V i upravljački ulaz koji prima 5 do 12 VDC. Relej je zapravo elektronski prekidač koji se aktivira signalom kojega pošaljemo s Arduina. Postojanje signala na upravljačkom ulazu utječe na položaj sklopke unutar releja, odnosno na to jesu li kontakti spojeni. Tri kontakta su C-engl. *common*, odnosno zajednički priključak, NC-engl.*normally closed* 120-240V, i NO- engl. *normally open* 120-240V.

2.5. Žarulja

U implementaciji ovog projekta rasvjetu kojom sustav upravlja predstavlja žarulja Philips od 25 W. Modul s relejem omogućava kontrolu žarulje kojoj treba veći napon od onog koji se može dobiti iz mikrokontrolera. Na sljedećoj slici prikazan je način spajanja žarulje s relejem.

**Slika 7 Spoj žarulje s relejem**

2.6. Zvučnik

Zbog ograničenja opreme, odnosno zvučnika na mobitelu i mikrofona na senzoru zvuka, postignut je „proof of concept“ na malom dometu do 20cm. U svrhu demonstracije ovog projekta unutar pametne kuće dodatno će se koristiti zvučnik koji će pojačavati signal s mobitela kako bi se domet povećao. Postoji više vrsta zvučnika koji se mogu koristiti:

Bluetooth zvučnik koji se postavi u blizini mikrofona

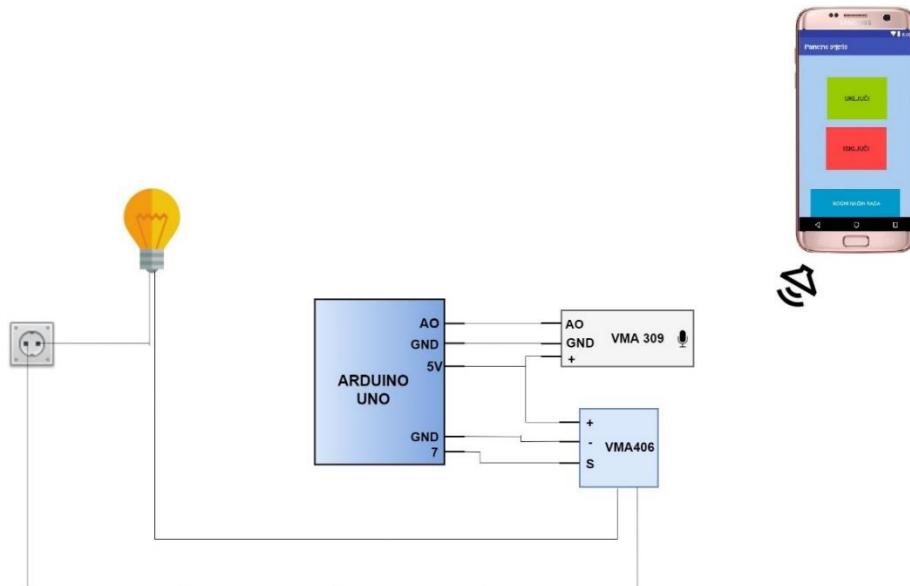
Mini zvučnik koji je preko 3.5mm audio priključka spojen s pametnim telefonom

Budući da nam je Bluetooth zvučnik bio dostupan odlučile smo njega iskoristiti u ovu svrhu. Pametni telefon poveže se preko Bluetooth-a na zvučnik koji šalje zvučne signale prema senzoru zvuka.



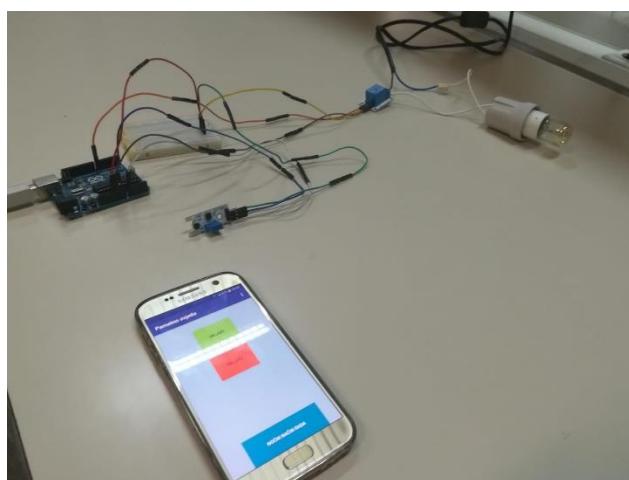
Slika 8 Bluetooth zvučnik

3. Realizacija sustava



Slika 9 Shema spajanja cijelog sustava

Na slici 8 ilustriran je shematski prikaz spajanja prethodno opisanih komponenti. Napon napajanja od 5V potreban je senzoru zvuka VMA 309 te releju VMA 406. Iz tog razloga praktično je upotrijebiti *protoboard* kako bi razveli priključak od 5V s Arduinom. Arduino je potrebno spojiti mini USB kabelom na računalo.



Slika 10 Realizirani sustav

4. Programska podrška

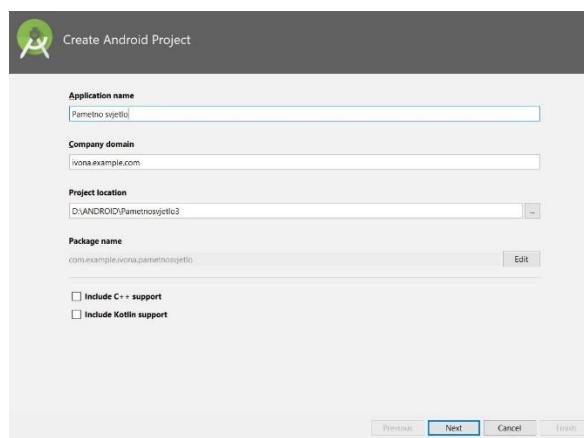
Prethodno opisan sustav vrši naredbe zapisane u programskoj podršci koja se može podijeliti na dva dijela:

1. Odašiljač – pametni telefon koji šalje visokofrekventni zvuk
2. Prijemnik – na osnovu primljenog zvučnog signala regulirala svjetlo

4.1. Odašiljač

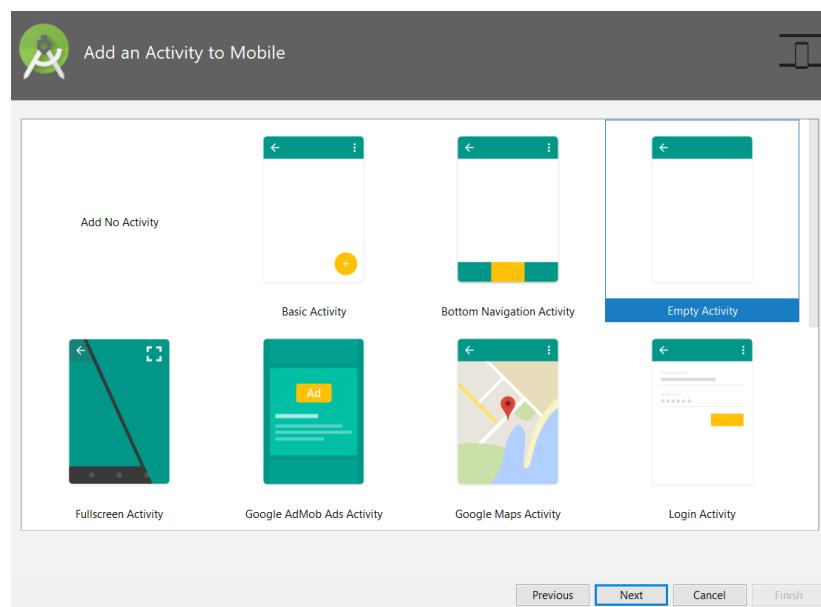
U sklopu ovog projekta razvijena je Android aplikacija koja omogućuje upravljanje rasvjetom putem pametnog telefona. Reproduciranjem zvuka visoke frekvencije mobitel signalizira mikrokontroleru kako treba regulirati osvjetljenje. Aplikacija je razvijena u razvojnem okruženju Android Studio. Android Studio je službeno integrirano razvojno okruženje za Android platformu i dostupan je za besplatnu instalaciju na službenoj web stranici <https://developer.android.com/studio/>. Programska potpora Android aplikacija piše se u Javi, dok se izgled korisničkog sučelja specificira XML-om.

Nakon instalacije svih potrebnih programskih paketa najprije je potrebno stvoriti novi Android projekt. Naziv projekta i same aplikacije je „Pametno svjetlo“. Potrebno je odabrati i platformu na kojoj će se izvršavati razvijena aplikacija. Odabrana verzija je API 25: Android 7.1.1 (Nougat).



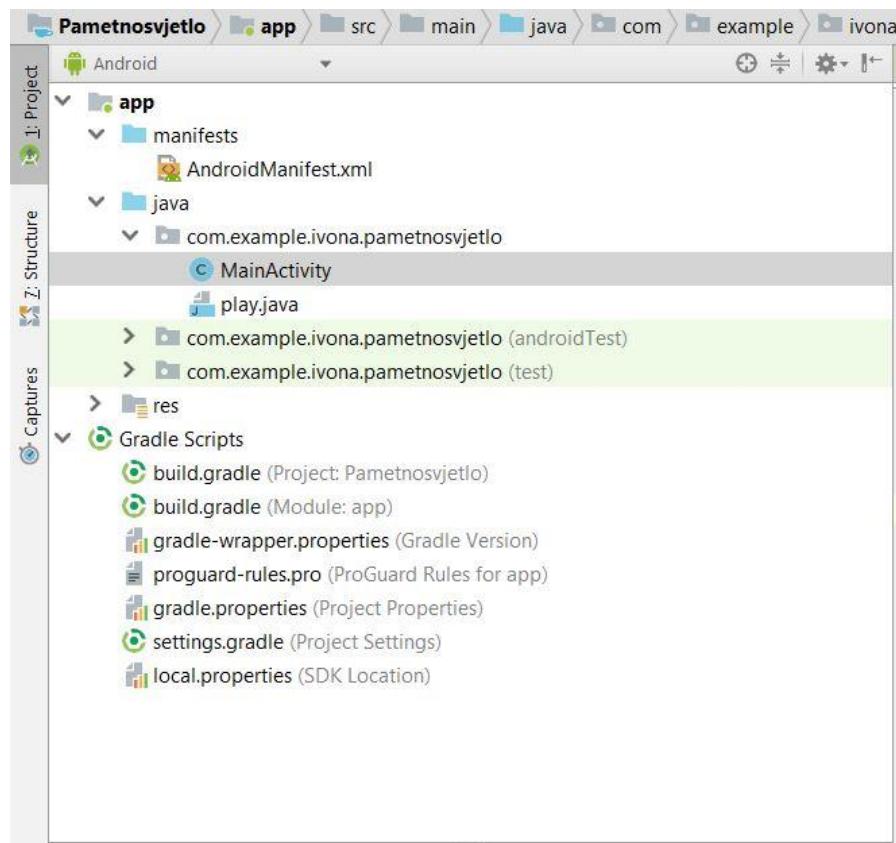
Slika 11 Stvaranje novog projekta

Pri kreiranju novog projekta, dodaje se i početna aktivnost koja predstavlja prvo grafičko sučelje koje vidi korisnik pri pokretanju aplikacije. Potrebno je odabrati praznu aktivnost (*Empty Activity*) i dodijeliti joj naziv *MainActivity* (slika 8).



Slika 12 Empty Activity

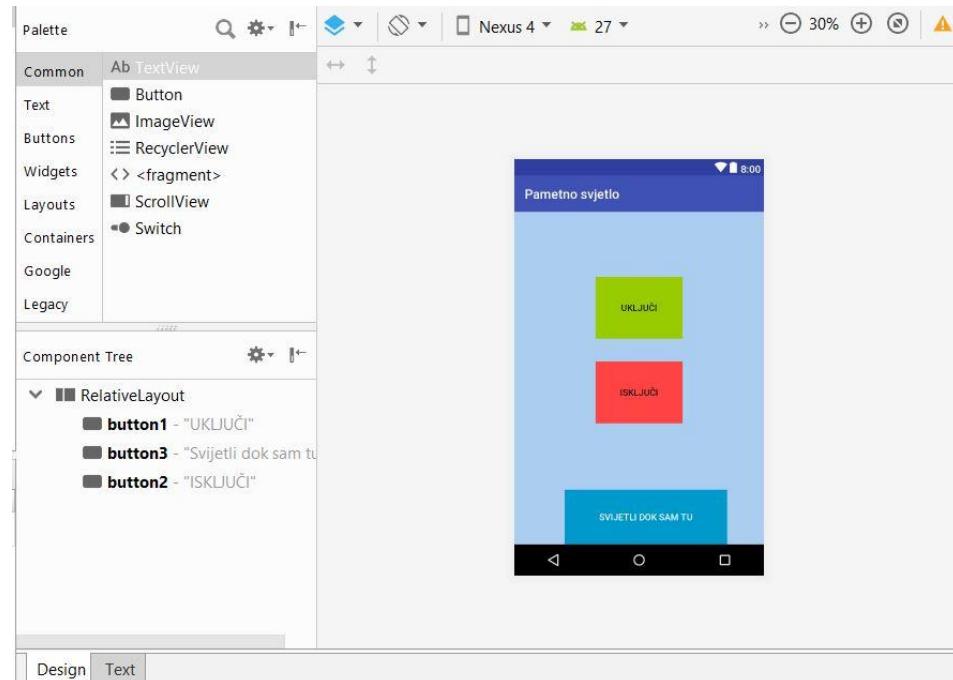
U prikazu izvornih datoteka *MainActivity* nalazi se unutar direktorija *java*. Kako su Android aplikacije pisane u Java programskom jeziku, programski kod se prevodi u izvršni kod. Izvršni kod je datoteka s ekstenzijom *.apk* koja se može instalirati na mobilni uređaj s operacijskim sustavom Android. Zadatak aktivnosti je prikaz korisničkog sučelja programa i omogućavanje interakcije korisnika s programom. Unutar izvornih datoteka nalazi se i manifest. Manifest je strukturirana XML datoteka naziva *AndroidManifest.xml* u kojoj se uređuje aplikacijski pristup sustavu uređaja. U njemu se nalaze informacije o komponentama aplikacije potrebne Android operacijskom sustavu prije pokretanja same aplikacije. To je potrebno kako bi mobitel znao što treba napraviti s aplikacijom te koje je resurse mobitela potrebno dopustiti da bi aplikacija radila. Res direktorij sadrži resurse projekta kao što su slike i XML datoteke koje opisuju izgled sučelja. Poddirektorij *layout* sadrži XML datoteke koje opisuju prikaz komponenti na ekranu. U poddirektoriju *values* sadržane su XML datoteke sa stringovima koji se koriste u projektu, bojama, dimenzijama teksta.



Slika 13 Izvorene datoteke

Pokretanjem aplikacije pokreće se glavna aktivnost *MainActivity*. Na zaslonu se pojavljuju tri različita gumba pomoću kojih korisnik izravno upravlja radom sustava. Tri gumba označavaju tri različita načina rada sustava.

Postavljanje i uređivanje gumbova vrši se pomoću XML datoteka koje se nalaze u poddirektoriju Layout unutar Res direktorija. Iz ponuđene *Palette* dodali smo tri gumba : „Uključi”, „Isključi” i „Svijetli dok sam tu” koji pokrivaju sve načine rada sustava.



Slika 14 Postavljanje gumbova u Android Studio-u

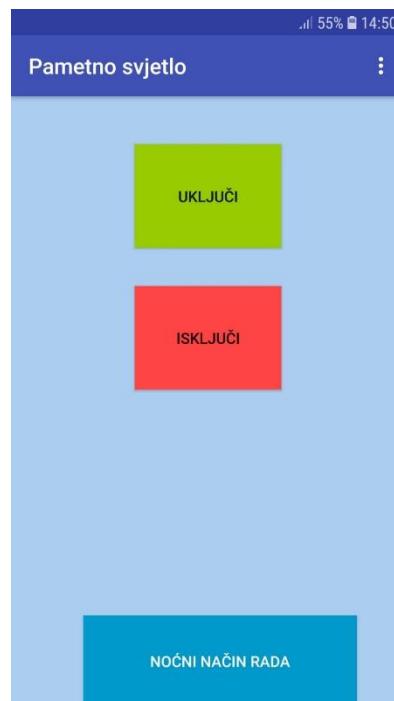
Nakon što smo definirali i postavili gumbove potrebno je svakom od njih dodijeliti odgovarajuću funkciju. Prvo moramo pozvati metodu `setOncliklistener` i zatim odgovarajućem objektu `Button` dodati novi `Listener` koji će reagirati na klik gumba. Pomoću funkcije `onClick` definiramo što će se dogoditi kad korisnik pritisne gumb.

```
public void onClick(View v) {
    isButtonClicked = !isButtonClicked;
    if (isButtonClicked) {
        freqSound3.start();
    } else {
        freqSound3.stop();
        try {
            freqSound3.prepare();
        }
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
}
```

Pritiskom na pojedini gumb zvučnik pametnog telefona reproducira signal od 15 kHz (`freqSound3`) kojeg prijemnik detektira putem mikrofona.

Nakon prijema signala ovisno o duljini njegovog trajanja i prethodnog stanja žarulje mikrokontroler regulira osvjetljenje.

Reproduciranje zvuka na frekvenciji od 15 kHz omogućuje nam klasu *MediaPlayer*. U našoj Android aplikaciji koristimo gotovi paket *media* iz kojeg preuzimamo klasu *MediaPlayer*.



Slika 15 Prikaz aplikacije "Pametno svjetlo"

4.2. Prijemnik

Sustav prijemnika čini senzor zvuka s mikrokontrolerom. Mikrofon koji se nalazi na senzoru zvuka prima zvučne signale iz pametnog telefona. Iz analognog izlaza senzora zvuka na ulazni pin Arduina A0 pristižu signali predstavljeni naponima 0-5V.

4.2.1. Povećanje frekvencije uzorkovanja

Budući da mikrokontroler mora izvršiti frekvencijsku analizu signala, potrebno ga je obraditi, odnosno uzorkovati. Cilj je ostvariti detekciju na vrlo visokim frekvencijama. Prema Nyquistovom pravilu o uzorkovanju, frekvencija uzorkovanja mora biti barem dva puta veća od najviše frekvencije koja se detektira. Budući da želimo detektirati frekvencije do 20 kHz frekvencija uzorkovanja mora biti minimalno oko 40kHz.

Arduino pruža jednostavan način čitanja analognog ulaza, a to je pomoću funkcije AnalogRead(). Zadane postavke određuju da je ADC clock rezultat dijeljenje 16MHz, frekvencije na kojoj radi procesor, sa „prescale“ faktorom. Inicijalno zadani „prescale“ faktor iznosi 128, te je $16\text{MHz}/128=125\text{kHz}$. Budući da konverzija analognog signala u digitalni traje 13 ADC clockova, frekvencija otipkavanja je onda $125\text{kHz}/13$, a to je 9600Hz.

Kako bi omogućili mnogo veće frekvencije otipkavanja potrebno je podesiti ADC, odnosno smanjiti „prescale“ faktor. Takve se promjene odvijaju u setup funkciji podešavanjem ADPS2, ADPS1 i ADPS0. Na sljedećoj slici vidljivo je koje kombinacije odgovaraju kojoj frekvenciji uzorkovanja. Naravno, smanjenjem *prescale* faktora utječemo i na razlučivost signala koje uzorkujemo, pa ni njega nije poželjno previše smanjiti.

Prescale	ADPS2	ADPS1	ADPS0	Clock freq (MHz)	Sampling rate (KHz)
2	0	0	1	8	615
4	0	1	0	4	307
8	0	1	1	2	153
16	1	0	0	1	76.8
32	1	0	1	0.5	38.4
64	1	1	0	0.25	19.2
128	1	1	1	0.125	9.6

Slika 16 Kombinacije za odabir frekvencije uzorkovanja

Budući da radimo s frekvencijama koje su sigurno manje od 20 khz, dovoljan sampling rate koji želimo postići je 38,4kHz. To je realizirano ovim postavkama:

```
ADCSRA |= (1 << ADPS2) | (1 << ADPS0);
```

Još jedan način za ubrzanjem programa je izbjegći korištenje funkcije AnalogRead(), koja sama traje otprilike 100 us, nego korsititi "ADC Free Running mode". U ovakovom načinu rada ADC kontinuirano prima analogni ulaz i pretvara ga u digitalni, te šalje interrupt na kraju svake konverzije. Ovakav način rada ima dvije glavne prednosti:

1. Ne gubi vrijeme čekajući da sljedeće uzorkovanje omogući čitanje s analognog ulaza, nego se to kontinuirano odvija
2. Povećava učinkovitost uzorkovanja smanjujući "jitter"

Tako smo podesili Arduino da radi na frekvenciji uzorkovanja 38,4 khz što je dovoljno za frekvencije koje želimo detektirati.

Sljedeći je korak implementirati algoritam za detekciju frekvencije.

4.2.2. Goertzelov algoritam

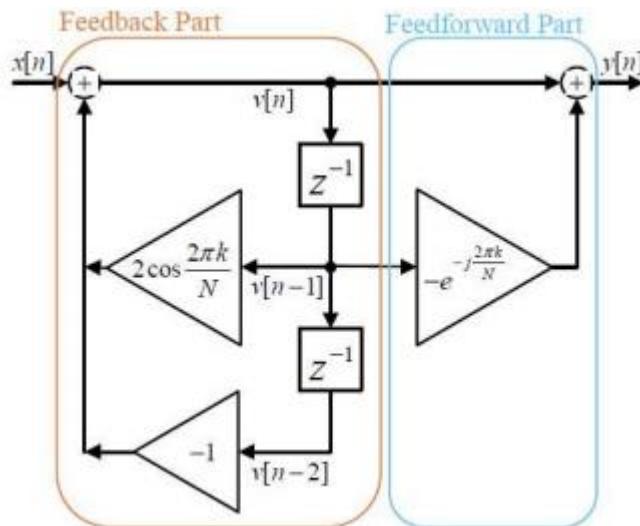
U svrhu detektiranja signala točno određene frekvencije na Arduinu se implementirao Goertzelov algoritam.

Najčešće se frekvencijska analiza signala temelji na brzoj Fourierovoj transformaciji (FFT). FFT je numerički efikasan algoritam za računanje diskretne Fourierove transformacije i daje zadovoljavajuće rezultate u većini slučajeva. Međutim, FFT ima i lošu osobinu, zahtjeva sinkronizaciju frekvencije uzorkovanja s frekvencijom signala. Pored nedostataka koji se odnose na potrebu sinkronizacije, FFT ima nedostatke koji su prouzrokovani blokovskom implementacijom. Naime, FFT obrađuje kompletan blok podataka i ne može dati rezultate za trenutke vremena unutar blokovskog prozora. Ako se to želi omogućiti, potrebna su puno komplikirane implementacije i opseg računanja postaje veći.

Mnogi embedded uređaji nemaju mogućnost provoditi FFT u stvarnom vremenu. Goertzelov algoritam može provoditi detekciju tona koristeći mnogo manje CPU snage nego FFT. Iako postoje hardverska rješenja za detekciju tona, poput implementacije s detektorom tona ili raznim filtrima, implementirati ove funkcije softverski mnogo je jeftinije. Jednostavna struktura Goertzelovog algoritma čini ga pogodnim za male procesore, embedded aplikacije i ostalo.

Osnovni Goertzelov algoritam nastao je 1958. godine i izведен je iz diskretne Fourierove transformacije. Izuzetno je efikasan algoritam kojim se određuju frekvencijske vrijednosti željenog signala iz bloka ulaznih podataka diskretnog vremenskog signala. Na osnovu relacija između frekvencije uzorkovanja, dužine vremenskog bloka ulaznog signala i dodatno izračunatih koeficijenata na osnovu prikupljenih uzoraka, moguće je detektirati postojanje određene frekvencije komponente.

Na sljedećoj slici je prikazan tijek obrade osnovnog Goertzelovog algoritma za svaki diskretan trenutak uzorkovanog ulaznog signala. Goertzelov algoritam je ustvari veoma visoko Q pojasno propusni filter kojeg karakterizira podesiv selektivan opseg i realiziran je kao IIR filter drugog reda.



Slika 17 Prikaz Goertzelovog algoritma

Kod Goerzelovog algoritma, niz N uzoraka signala je transformiran u niz N frekvencijskih koeficijenata koristeći diskretnu Fourierovu transformaciju(DFT).

To znači da se detekcija tona događa svakih N uzoraka. Kao i kod FFT-a, radi se s blokom uzoraka. Međutim, to ne znači da se moraju procesirati podaci u blokovima. Numeričko procesiranje je skraćeno tako da se u prekidnoj petlji prikupljaju uzorci koji se kasnije obrađuju u lopp petlji.

Prilikom implementacije u Arduinu, prije nego što se primijeni Goertzelov algoritam, najprije se trebaju obaviti određeni proračuni.

Treba se odrediti frekvencija uzorkovanja, odabrati veličinu bloka, proračunati kosinus i sinus određene frekvencije, te izračunati koeficijente.

Frekvenciju uzorkovanja već smo odredili podešavanjem ADC clock-a. Veličina bloka N je kao broj točaka u ekvivalentnom FFT-u. Kontrolira rezoluciju frekvencije, odnosno širinu frekvencijskog područja. Primjerice, u našem slučaju uzorkujemo sa 38kHz i broj N je 200, tada je širina frekvencijskog područja 190Hz.

To bi značilo da je za veću rezoluciju potreban što veći N , međutim povećanjem N se produljuje i vrijeme u kojem se ton detektira, budući da se treba čekati duže da se prikupi više uzoraka. Na primjer, za 8kHz frekvencije uzorkovanja, potrebno je 100ms da se spremi 800 uzoraka, dok je za 38kHz potrebna 21ms. Tu može nastati problem ako se detektiraju kratki tonovi, pa se ne uspiju uzeti svi uzorci.

Treći faktor koji utječe na odabir varijable N je veza između frekvencije uzorkovanja i tražene frekvencije. Idealno bi bilo da je frekvencija uzorkovanja umnožak tražene frekvencije i N , kako bi se za svako detektiranje uzeo točno N uzoraka.

Kada smo odredili frekvenciju uzorkovanja, frekvenciju koju želimo detektirati i blok N, potrebno je izračunati koeficijente koji će se koristiti u svrhu detekcije frekvencije.

```
k= (int) (0.5+(N*targ_freq)/sample_rate);  
w=((2.0f*PI)/N)*k;  
cosine=cos (w);  
sine=sin (w);  
coeff=2*cosine;
```

Za svako procesiranje po uzorku bit će potrebne tri varijable koje su označene kao Q0, Q1 i Q2. Q1 i Q2 predstavljaju reverzne varijable koje su se prikupile uzorak prije, odnosno dva uzorka prije i one su na početku bloka inicijalizirane na nulu.

```
for (j=0;j<N;j++){  
Q0=coeff*Q1-Q2+x[j];  
Q2=Q1;  
Q1=Q0;  
}
```

Nakon pokretanja ovih formula N puta, odnosno za svaki uzorak iz bloka, završava se s rekurzivnim operacijama. Tada se pokreću nerekurzivne operacije izračuna realne i imaginarne komponente frekvencije. Pomoću njih računa se magnituda.

```
real = (Q1 - Q2 * cosine);  
imag = (Q2 * sine);  
magnitude = sqrt(real*real + imag*imag);
```

Magnituda je zbog proračuna veoma mala za sve frekvecije koje nisu tražene, dok je za traženu frekvenciju veća. Tako se eksperimentalno, proučavajući magnitudo željenih i neželjenih frekvencija, izabere određeni prag magnitude za koji možemo reći da je frekvencija prisutna. Taj prag se naziva threshold i uspoređujući njega sa magnitudom detektiramo prisutnost frekvencije.

Osim detektiranja određene frekvencije, prijemnik je programiran da automatski razlikuje dva načina rada. Potrebno je razlikovati kada korisnik želi upaliti/ugasiti svjetlo, a kada želi imati upaljeno svjetlo samo tijekom boravka u prostoriji. To je ostvareno različitim trajanjima slanja, odnosno primanja frekvencije. Kada prijemnik primi kratki signal tražene frekvencije, u odnosu na trenutno stanje rasvjete pali/gasi svjetlo. Kada signal te frekvencije traje duže od kratkih signala za paljenje/gašenje svjetla, tada se prebacuje na drugi način rada i drži upaljeno svjetlo sve dok prima taj signal, nakon čega ugasi svjetlo.

5. Testiranje sustava

Kao što je već opisano, implementiran je sustav za komunikaciju putem visoke frekvencije. Zvučnik pametnog telefona Samsung Galaxy S7 reproducira zvuk visoke frekvencije, a senzor zvuka VMA309 prima signale koje onda mikrokontroler obrađuje i na osnovu rezultata obrade upravlja rasvjetom.



Slika 18 Testiranje sustava

Kako bi se ispitale karakteristike i maksimalne mogućnosti takvog sustava potrebno ga je testirati na različitim frekvencijama i različitim udaljenostima.

Najprije je obavljeno testiranje mikrofona na senzoru zvuka. Prema specifikacijama mikrofon prima frekvencije u rasponu od 20Hz-20KHz, međutim za frekvencije veće od 17kHz odziv nije dovoljno dobar da bi se takve frekvencije mogle očitati. Zato je zbog ograničenosti mikrofona maksimalna frekvencija signala kojeg mikrokontroler može obraditi 17kHz.

Zbog ograničenosti resursa na ovom projektu nije bilo moguće raditi s boljim mikrofonima. Kako bi se poboljšali rezultati, u nastavku je popis

predloženih komponenata s boljim karakteristikama koje bi se mogle ugraditi u sustav i zamijeniti postojeće:

<https://earthworksaudio.com/microphones/qtc-series-2/qtc50/>

<https://www.freetronics.com.au/products/microphone-sound-input-module#.WyDvy6czZPa>

Budući da se glasnoća zvuka kod zvučnika pametnog telefona naglo smanji za sve frekvencije iznad 15 kHz, za demonstraciju se koristi bluetooth zvučnik u svrhu većeg dometa. Međutim, zvučnik koji je korišten ne prenosi frekvencije veće od 15kHz tako da smo zbog odašiljača ograničeni na komunikaciju od 15kHz.

Zbog navedenih razloga odlučeno je koristiti komunikaciju na 15kHz. Domet bez upotrebe zvučnika iznosi 30cm, što se može uvelike poboljšati upotrebom boljih komponenata, odnosno mobitela s vanjskim zvučnikom i mikrofona veće osjetljivosti.

Zaključak

U okviru ovog projekta realiziran je sustav „Pametna rasvjeta“ koji omogućuje reguliranje svjetlosti unutar prostorije u kojoj korisnik boravi. Sustav se sastoji od tri dijela : odašiljača, prijemnika i podsustava za izvršavanje. Odašiljač je predstavljen pametnim telefonom Samsung Galaxy S7 odnosno njegovim zvučnikom, prijemnik senzorom zvuka VMA 309 i mikrokontrolerom Arduino Uno, a podsustav za izvršavanje žaruljom koju kontrolira relej VMA406. U radu je opisana hardverska implementacija i odgovarajuća programska podrška. Testiranjem sustava ostvarena je uspješna komunikacija na 15 kHz što je visoka frekvencija na koju ne utječu zvukovi iz okoline, ali ne spada u područje koje ljudsko uho ne čuje. Daljnjim razvojem i upotrebom predloženih komponenata mogla bi se ostvariti komunikacija na nečujnim frekvencijama.

6. Literatura

- [1] Arduino Uno Overview,
URL: <https://store.arduino.cc/usa/arduino-uno-rev3>
- [2] VMA309 User Manual, 2017
URL: https://www.velleman.eu/downloads/29/vma309_a4v02.pdf
- [3] Arduino Sound Detection Sensor: Tutorial and User Manual, 2016
URL: <http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-sound-detection-sensor-tutorial-and-user-manual/> (2016-01-22)
- [4] VMA406 User Manual, 2017.
URL: https://www.velleman.eu/downloads/29/vma406_a4v02.pdf
- [5] The Goertzel Algorithm, 2002
URL: <https://www.embedded.com/design/configurable-systems/4024443/The-Goertzel-Algorithm>, (2002-02-28)
- [6] Meet Android Studio, 2017
URL: <https://developer.android.com/studio/intro/> (2018-06-05)
- [7] It's time for a sound revolution, 2017
URL: <https://www.itproportal.com/features/its-time-for-a-sound-revolution/> (2017-09-21)
- [8] Fast sampling from analog input, 2015
URL: <http://yaab-arduino.blogspot.com/2015/02/fast-sampling-from-analog-input.html>, (2015-02-01)
- [9] Arduino Audio Input, 2012
URL: <http://www.instructables.com/id/Arduino-Audio-Input> (2012-02-10)



7. Pojmovnik

8. Raspodjela zadataka

Članovi tima	Individualni zadaci
Mihaela Ivanković	implementacija Goertzelovog algoritma na Arduino, odabir i implementacija hardvera sustava, testiranje sustava
Ivona Lončar	dizajn i izrada Android aplikacije, podešavanje Arduina za rad na višim frekvencijama uzorkovanja, testiranje sustava