

LED vizualizator zvuka baziran na Arduino platformi

Androić, Ivan

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:110:794862>

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

Download date / Datum preuzimanja: **2024-10-07**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository -
Polytechnic of Međimurje Undergraduate and
Graduate Theses Repository](#)



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVO

IVAN ANDROIĆ

**LED VIZUALIZATOR ZVUKA BAZIRAN NA ARDUINO
PLATFORMI**

ZAVRŠNI RAD

Čakovec, rujan 2022.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI STUDIJ RAČUNARSTVO

IVAN ANDROIĆ

**LED VIZUALIZATOR ZVUKA BAZIRAN NA ARDUINO
PLATFORMI**

**LED SOUND VISUALIZER BASED ON THE ARDUINO
PLATFORM**

ZAVRŠNI RAD

Mentor:
Jurica Trstenjak, dipl.ing; viši predavač

Čakovec, rujan 2022.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
ODBOR ZA ZAVRŠNI RAD

Čakovec, 12. siječnja 2021.

država: **Republika Hrvatska**
Predmet: **Digitalni elektronički sklopovi**
Polje: **2.09 Računarstvo**

ZAVRŠNI ZADATAK br. 2020-RAČ-R-52

Pristupnik: **Ivan Androić (0313022992)**
Studij: **redovni preddiplomski stručni studij Računarstvo**
Smjer: **Inženjerstvo računalnih sustava i mreža**

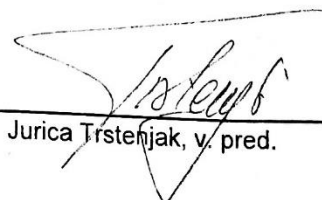
Zadatak: **LED vizualizator zvuka baziran na Arduino platformi**

Opis zadatka:

Potrebno je izraditi platformu sa 2x7 LED traka pri čemu svaka predstavlja jednu frekvenciju ulaznog tona. Samu traku kontrolira Arduino platforma. Potrebno je odabrati odgovarajuću ulaznu jedinicu koja će pretvarati ulazni audio signal i slati ga na Arduino platformu. Intenzitet paljenja trake ovisi o intenzitetu zvuka koji se dovodi na ulaznu jedinicu. Cijeli sustav je potrebno staviti u zajedničko kućište. Provesti testiranje rada pojedinih dijelova modela kao i cijelog sustava.

Rok za predaju rada: **20. rujna 2021.**

Mentor:



Jurica Trstenjak, v. pred.

Predsjednik povjerenstva za
završni ispit:

Zahvala

Ovim putem zahvaljujem se mentoru Jurici Trstenjaku na strpljenju i razumijevanju te njegovoj pomoći kod izrade završnog rada, ali i kod pomoći na svim predmetima koje je održavao.

Želim se zahvaliti svojoj obitelji koja mi je uvijek bila potpora i pružala mi sigurnost u studiranju i životu.

Zahvaljujem se prijateljima uz koje nikad nije bilo dosadno i koji su mi pomogli učiniti studiranje zanimljivim.

Također se zahvaljujem svim profesorima Međimurskog Veleučilišta na trudu i dijeljenju njihovog znanja.

Na kraju se zahvaljujem svim zaposlenicima Međimurskog Veleučilišta koji su se uvijek brinuli kako bi nam studiranje bilo što ljepše i ugodnije.

Ivan Androić

SAŽETAK

Zvuk je mehanički val frekvencija koji se može sastojati od jedne ili više frekvencija. Pojam zvuka shvaćen je kao nešto što čujemo, ali je mnogo kompleksniji i može se prikazati na druge načine. Ovaj završni rad prikazuje zvuk u vizualnom obliku. Glavni izvor informacija je sam zvuk koji se pomoću mikrofona detektira te se šalje na mikrokontroler. Izabrani mikrokontroler za ovaj završni rad je ESP32 koji odgovara potrebnim parametrima kao što su brzina i količina memorije. Na ulaz mikrokontrolera spaja se već spomenuti mikroskop koji zaprima audio informacije te ih šalje mikrokontroleru na obradu. Za obradu podataka koristi FFT (engl. *Fast Fourier Transform*) algoritam koji nakon 1024 dobivena uzorka audio zapisa matematičkom formulom odvaja pomiješane frekvencije unutar audio signala. Nakon matematičke obrade dobije se niz zasebnih frekvencija koje se nakon toga spremaju. Mikrokontroler zasebne frekvencije šalje na izlaz te ih dodjeljuje određenom nizu LED dioda. Kao prikaz izlaznih informacija korištena je LED matrica od 192 zasebne adresabilne LED diode. Važno je da su LED diode adresabilne kako bi se svakom snopu koji se sastoji od 12 LED dioda mogao dodijeliti određeni frekvencijski raspon. Na kućištu je raspoređeno 16 snopova od kojih se svaki sastoji od 12 LED dioda. Raspon frekvencija koji snopovi prikazuju kreće se od 304 Hz do 3154 Hz. Prvih 8 frekvencijskih raspona zrcaljeni je prikaz drugih 8 frekvencijskih raspona što znači da prvi snop LED dioda prikazuje istu frekvenciju kao i zadnji snop LED dioda. Sustav također prepoznaje i glasnoću audio zapisa koja se prikazuje kao visina snopa na LED matrici. LED diode su međusobno serijski povezane pomoću konektora koji ih spajaju u jednu zasebnu cjelinu. Sve komponente ugrađene su u drveno kućište s prozirnim plastičnim poklopcem.

Programski dio kôda pisan je u Arduino IDE sučelju koji je potrebno podesiti kako bi rad s ESP32 mikrokontrolerom bio moguć.

Ključne riječi: *ESP32, LED, FFT, Arduino, audio signali*

SADRŽAJ

1.UVOD	7
2. HARDWARE	8
2.1. ESP 32 mikrokontroler	8
2.2 WS2812B LED traka.....	10
2.3 Veza između LED traka.....	11
2.4 Detektor zvuka (mikrofon).....	12
3.ZVUK	13
4. FREKVENCIJA.....	14
4.1 Amplituda.....	15
5. Izrada Vizualizatora zvuka	17
6. ARDUINO IDE SUČELJE.....	18
6.1 Skidanje i instalacija Arduino IDE sučelja.....	18
6.2 Instalacija ESP32 razvojne pločice.....	20
6.3 Instalacija biblioteka	23
7. OBJAŠNJENJE KÔDA	25
7.1 Definiranje parametara i pinova	25
7.2 Definiranje matrice	26
7.3 Definiranje tipova podataka	27
7.4 Postavljanje FastLED parametara	28
7.5 Prikupljanje uzoraka za obradu.....	29
7.6 FFT (Fast Fourier Transform).....	30
7.7 Dodjela frekvencijskih raspona	33
8.PRIKAZ VIZUALIZATORA ZVUKA.....	34
9. ZAKLJUČAK.....	35
POPIS LITERATURE	36
POPIS PROGRAMSKIH KÔDOVA	37
POPIS SLIKA.....	38
POPIS TABLICA	39

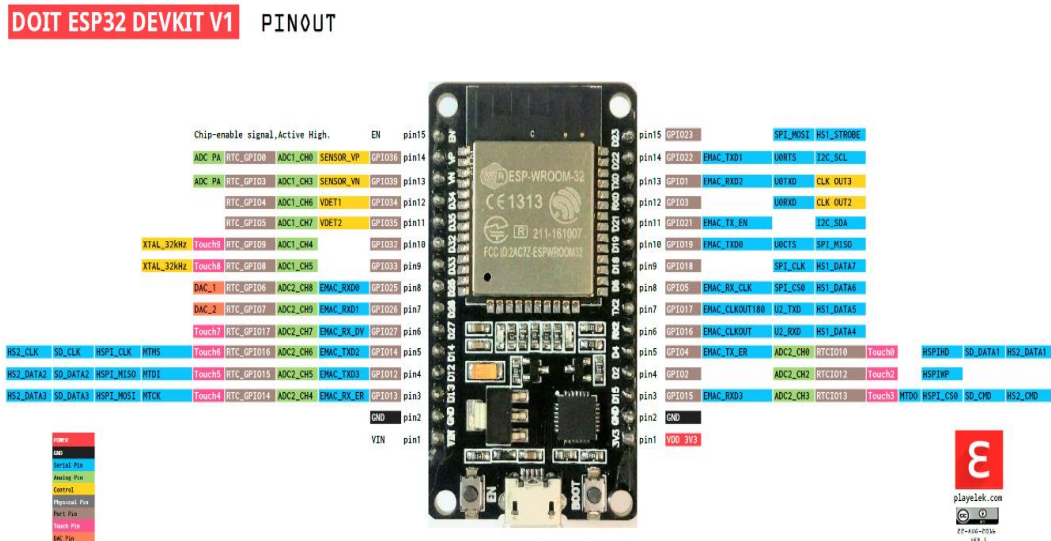
1.UVOD

Cilj ovog završnog rada je napraviti Vizualizator zvuka koji prima različite frekvencije tonova preko senzora zvuka (mikrofona) koji se nakon toga šalju ESP32 mikrokontroleru na obradu te dobivenu frekvenciju šalje na određenu adresu LED trake koja projicira frekvenciju i jačinu tona u svjetlosnu skalu. Ovaj završni zadatak koristi ESP32 mikrokontroler kao mozak koji obrađuje dobivene podatke te ih prosljeđuje na LED trake. Ulazni medij je audio signal koji se snima preko mikrofona te se obrađuje. Audio signal potrebno je obraditi pomoću FFT (engl. *Fast Fourier Transform*) algoritma koji odvaja pomiješane signale te nakon obrade daje jasne signale određenog frekvencijskog raspona. Sučelje u kojem je izrađen programski dio zadatka je Arduino IDE. Rad sadrži 2x8 LED snopova koji prikazuju 8 različitih frekvencijskih pojasa. Način na koji su frekvencije prikazane je taj da prvih 8 snopova prikazuje 8 različitih frekvencijskih pojasa, dok je drugih 8 snopova zrcaljena slika istih frekvencijskih pojasa koje prikazuje prvih 8 snopova. Za izradu ovog rada koristi se ESP32 mikrokontroler, WS2812B adresabilne LED trake, konektor, mikrofoni. Podaci potrebni za obradu i prikazivanje audio signala u vizualni prikupljeni su na stranici *Arduino project hub*.

2. HARDWARE

2.1. ESP 32 mikrokontroler

Prvotni plan za izradu ovog završnog rada bilo je korištenje Arduino Uno mikrokontrolera, ali tu nastaje problem nedovoljne memorije za spremanje programa te brzina obrade samih audio podataka. Istraživanjem ostalih Arduino mikrokontrolera ne pojavljuje se nijedan koji bi imao dovoljno memorije za spremanje programa te kao rješenje na ovaj problem dolazi ESP32 mikrokontroler. ESP32 jeftini je mikrokontroler s ugrađenim Bluetooth i WI-FI modulima. Riječ je o dvojezrenom mikrokontroleru s 512 kB SRAM memorije koju je moguće proširiti do 8 MB pomoću vanjskih modula. Zbog velike količine memorije i brzine ovaj mikrokontroler odličan je za prepoznavanje i obradu zvuka, prepoznavanje slika, IoT. Važno je spomenuti kako podržava (engl. *Arduino framework*) i sve biblioteke pisane za Arduino. ESP32 razvija i proizvodi tvrtka *Espressif Systems* te je nasljednik ESP8266 mikrokontrolera.



2.2 WS2812B LED traka

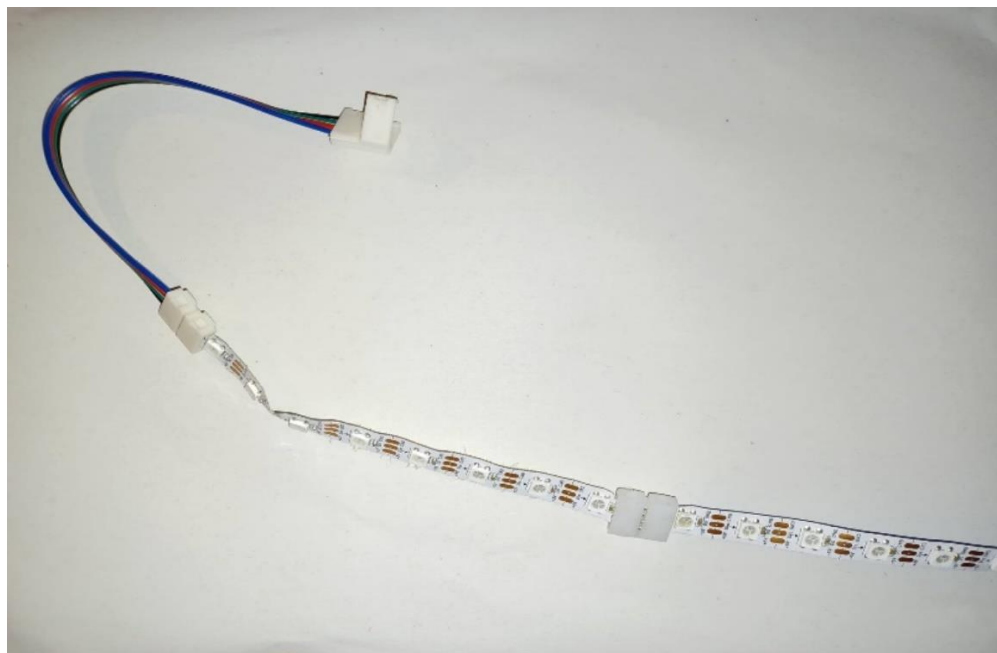
Kao izlaz za prikaz audio zapisa u vizualni za ovaj rad koristi se WS2812B LED traka.

LED traka je fleksibilna pločica na koju su montirane svjetleće diode. Najčešća razliku u dizajnu LED traka je u načinu upravljanja pojedinim LED diodama. Postoje dvije vrste: adresabilne i neadresabilne. Važnost ove trake jest ta da je ona adresabilna, što znači da svaka LED dioda u nizu ima svoju adresu. Potreba adresabilne LED trake u ovom radu je ta da jedan niz dioda predstavlja jednu frekvenciju, bez da time utječe na rad i predstavljanje frekvencije drugog niza dioda

Ovaj završni rad sadržava 16x12 LED dioda i prikazuje raspon od 16 frekvencija.

Adrese povezanih dioda koje prikazuju istu frekvenciju dobiju se na način da prvoj diodi u nizu te frekvencije dodamo broj 16.

Primjer: u prvi frekvencijski raspon spadaju diode s adresom – 1,17,33...209



Slika 2. WS2812B LED TRAKA

Izvor: Autor

2.3 Veza između LED traka

Za vezu između svake LED trake korišteni su konektori koji spajaju krajeve LED trake.

LED trake na sebi sadrže tri konektora +5V (konektor za spajanje izvora napajanja), D0 (konektor koji služi za prijenos informacija od mikrokontrolera do svake od LED dioda) i GND (konektor za uzemljenje).

Dužina LED trake je po potrebi jer se traka može rezati običnim škarama. Između svake diode nalazi se spoj koji nakon što se prereže može biti spajan pomoću konektora ili lemljenjem.



Slika 3. Veza između LED traka

Izvor: Autor

2.4 Detektor zvuka (mikrofon)

Glavni izvor informacija ovog završnog rada je zvuk. Zvuk koji se obrađuje nekako mora doći do mikrokontrolera te je zbog toga potreban modul detektora zvuka.

Detektor zvuka je modul koji šalje signal preko digitalnog izlaza nakon što detektira zvuk.

Osnovni dio modula je kondenzatorski mikrofon koji energiju zvuka pretvara u električnu. Titranjem tanke električki vodljive membrane mijenja se električni kapacitet kondenzatora, a time i izmjenična struja punjenja i pražnjenja kondenzatora, to uzrokuje izmjenični pad napona na otporniku koji se kasnije očitava kao audio signal.

Na modulu se nalazi i analogni komparator napona koji radi kao voltmetar te mjeri razliku između dvije točke i uspoređuje ih. Ako je napon u prvoj točki veći od napona u drugoj točki prekidač je uključen (postoji zvuk koji se obrađuje), u obratnom slučaju prekidač je isključen (ne postoji zvuk za obradu).

Na modulu također postoji i potenciometar koji služi za podešavanje osjetljivosti modula na zvuk.



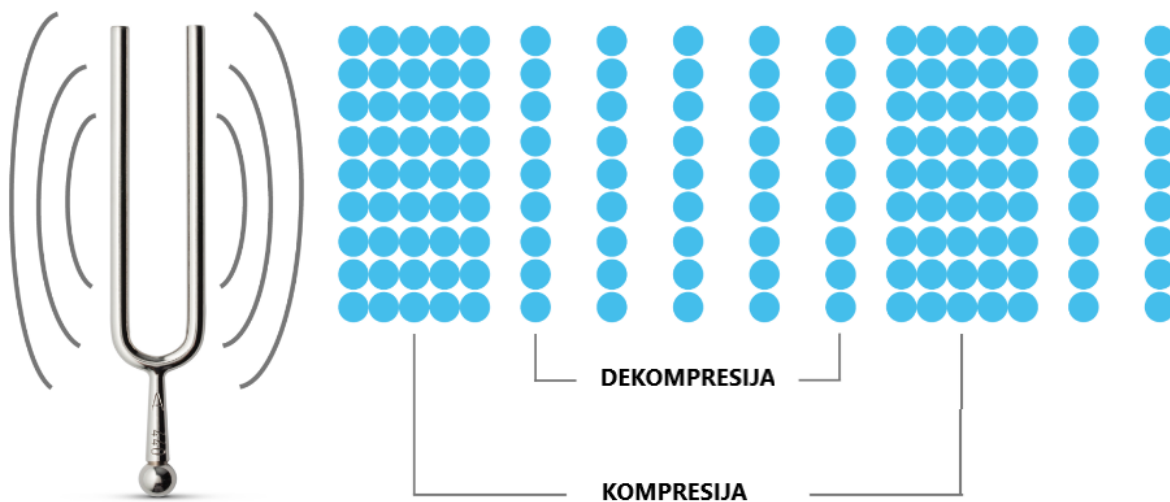
Slika 4. Modul detektora zvuka

Izvor: https://phillipjfray.com/products/voice-sound-detection-sensor-module-for-arduino?pr_prod_strat=copurchase&pr_rec_pid=6941215522996&pr_ref_pid=6923849990324&pr_seq=uniform (20.7.2022)

3.ZVUK

Informacija na ulazu je audio zapis (zvuk) koji je detektiran na senzoru i preko ESP32 mikrokontrolera obrađen te preslikan i prikazan kao vizualni zapis na LED trakama. Zvuk na ulazu sastavljen je od više frekvencija, te ga se zbog toga mora obraditi pomoću FFT (engl. *Fast Fournier Transform*) algoritma.

Zvučni val sastoji se od naizmjeničnih kompresija i dekompresija krećući se određenom brzinom, zbog toga nastaju oscilirajuće varijacije tlaka tj. frekvencija nekog zvuka.



Slika 5. Primjer kompresije i dekompresije

Izvor: <https://www.pasco.com/products/guides/sound-waves> (3.9.2022)

4. FREKVENCIJA

Frekvencija je fizikalna veličina koja iskazuje broj ponavljanja neke periodične veličine u jedinici vremena.

Postoji odnos između frekvencije vala i njegove periode (duljine trajanja) $T=1/f$. Zvučni valovi s visokim frekvencijama imaju kratka razdoblja trajanja, dok oni s niskim frekvencijama imaju duga razdoblja trajanja.

Ljudsko uho može čuti frekvencije između 20 Hz i 20000 Hz, frekvencije iznad 20000 Hz nazivaju se ultrazvukovi, a frekvencije ispod 20 Hz nazivaju se infrazvukovi.

Primjer računanja duljine trajanja frekvencije:

$$=30 \text{ Hz}$$

$$30=1/T \rightarrow T=1/30 \rightarrow T=0,033\text{s}$$

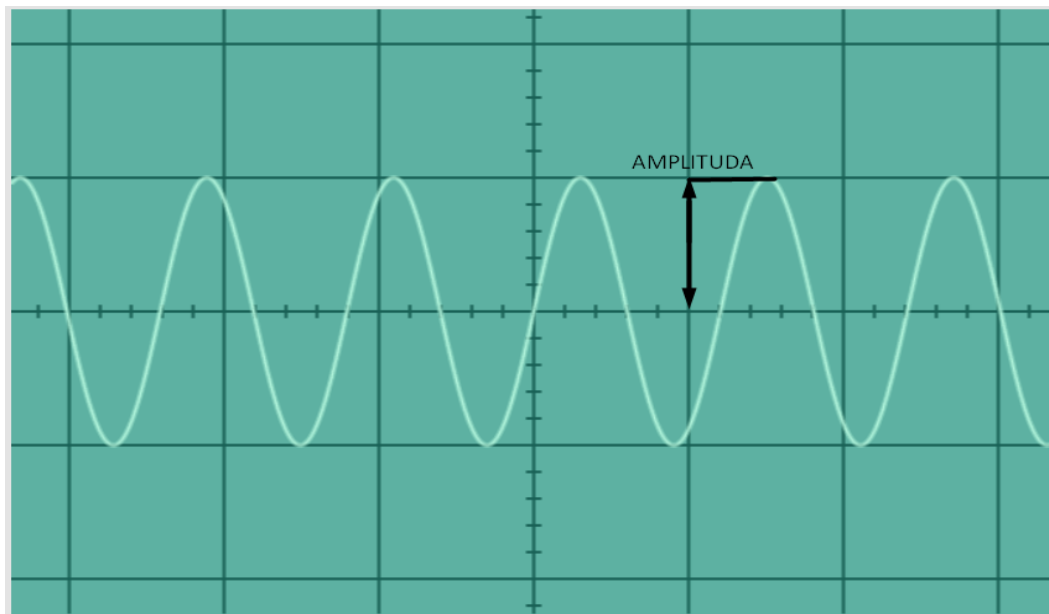
Primjer računanja frekvencije:

$$T=0,033\text{s}$$

$$0,033=1/f \rightarrow f=1/0,033 \rightarrow 30 \text{ Hz}$$

4.1 Amplituda

Amplituda zvučnog vala ovisi o promjeni tlaka zraka. U praksi se obično koristi logaritam omjera amplitude i referentnog tlaka zraka izraženog u decibelima.



Slika 6. Amplituda zvučnog vala od 829 HZ

Izvor: Autor

Jačina glasa prilikom normalnog govora iznosi ispod 60 dB, glasnoća nekog radnog prostora iznosi oko 85dB, dok je glasnoća nekog rock koncerta 125 dB.

Decibeli se mjere prema logaritamskoj skali, što znači da se za povećanje od 10 dB zvuk pojačava deseterostruko, pr. ako je izmjereno da je zvuk nekog stroja 70 dB, ekvivalent 80 dB, bilo bi 10 takvih strojeva.

Tablica 1: *Razina glasnoće u decibelima*

Vrsta zvuka	Razina glasnoće(fon)
prag <u>osjeta</u>	0 dB
Šaptanje	20 dB
tiha <u>glazba</u>	40 dB
bučan govor	60 dB
<u>prometna ulica</u>	80 dB
prolazak brzog <u>vlaka</u>	100 dB
<u>motor zrakoplova</u>	120 dB
prag <u>bola</u>	130 dB

Izvor: <https://www.rapidtables.org/hr/electric/decibel.html> (25.7.2022)

5. Izrada Vizualizatora zvuka

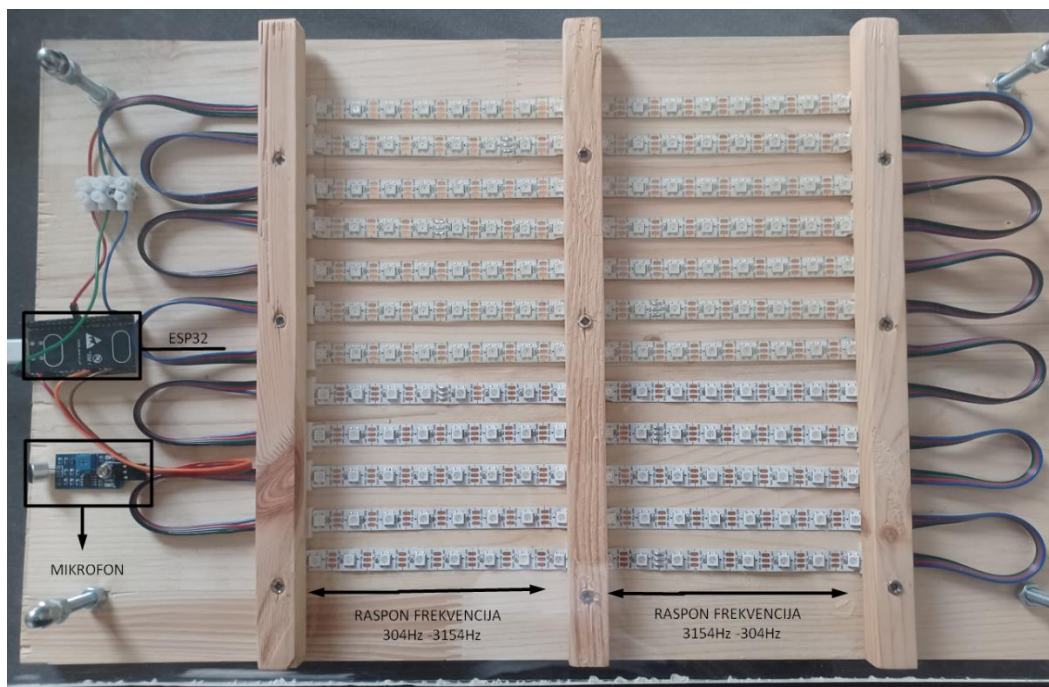
Materijali potrebni za izradu ovog Vizualizatora zvuka su: mikrofonski, ESP32 razvojna pločica, konektori za povezivanje LED traka, LED trake, prozirni pleksiglas i drvo za kućište.

Prvi korak izrade ovog završnog rada je lijepljenje LED traka na prije izmjerene razmaka i poziciju.

Nakon lijepljenja potrebno je međusobno povezati LED trake pomoću konektora (LED trake su međusobno spojene serijski).

Kao zadnji korak potrebno je povezati mikrofonski i LED trake na ESP32 razvojnu pločicu.

Mikrofon je povezan na pinove GND¹, +3V i na PIN35² (audio ulaz). LED trake povezane su na GND, +5V i na PIN 5³.



Slika 7. Vizualizator zvuka

Izvor: Autor

¹ GND- pin na razvojnoj pločici koji služi kao uzemljenje.

² PIN35- pin na koji je spojen mikrofonski i koji služi kao audio ulaz

³ PIN5- izlaz iz mikrokontrolera, pin koji spaja ledice

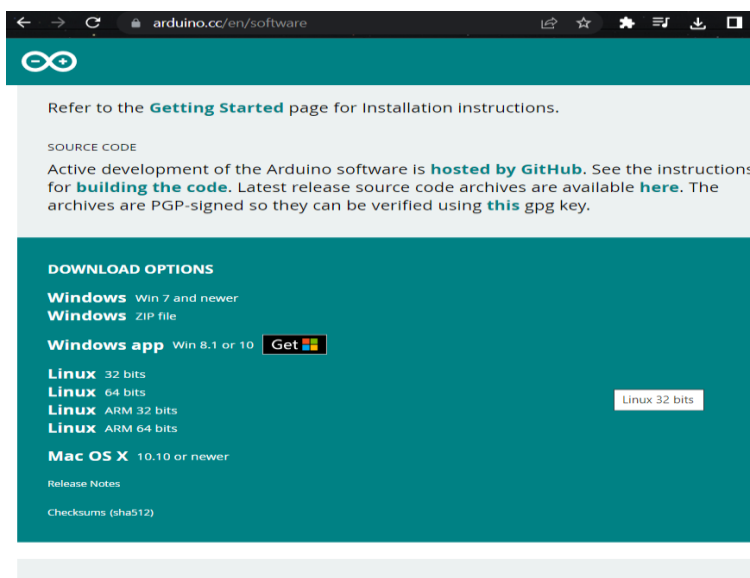
6. ARDUINO IDE SUČELJE

Arduino IDE je (engl. *Open Source Software*) koji služi za programiranje i učitavanje programa na razvojnu pločicu.

Podržan je za Windows, Linux i Mac OS X operativne sisteme.

6.1 Skidanje i instalacija Arduino IDE sučelja

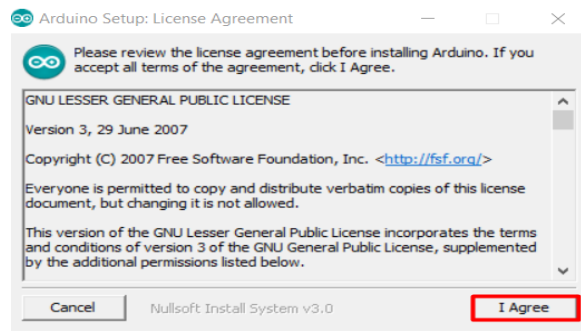
1. korak- na linku: <https://www.arduino.cc/en/software> potrebno je odabrati odgovarajući operacijski sustav te skinuti odgovarajuću verziju.



Slika 8. Skidanje Arduino IDE sučelja

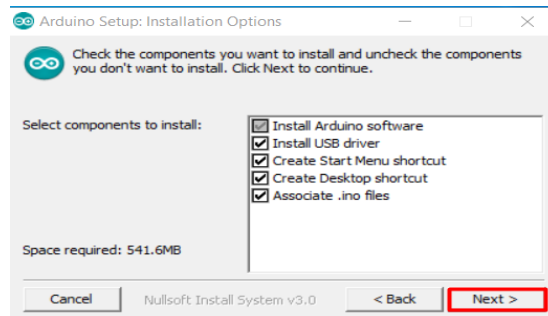
Izvor: Autor

2. Sljedeće je potrebno instalirati Arduino IDE



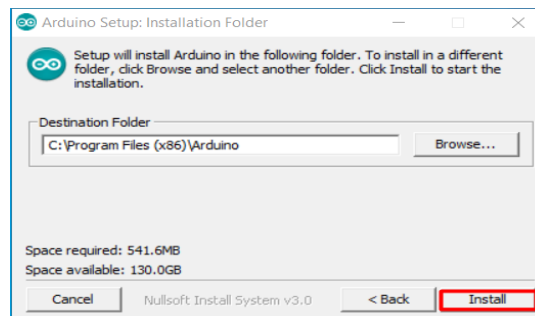
Slika 9. Instalacija Arduino IDE

Izvor: Autor



Slika 10. Instalacija Arduino IDE

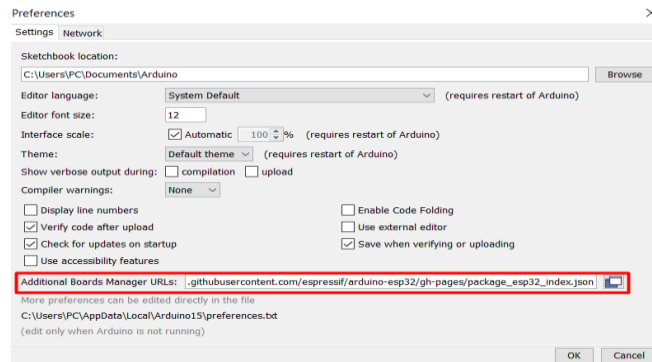
Izvor: Autor



Slika 11. Instalacija Arduino IDE

Izvor: Autor

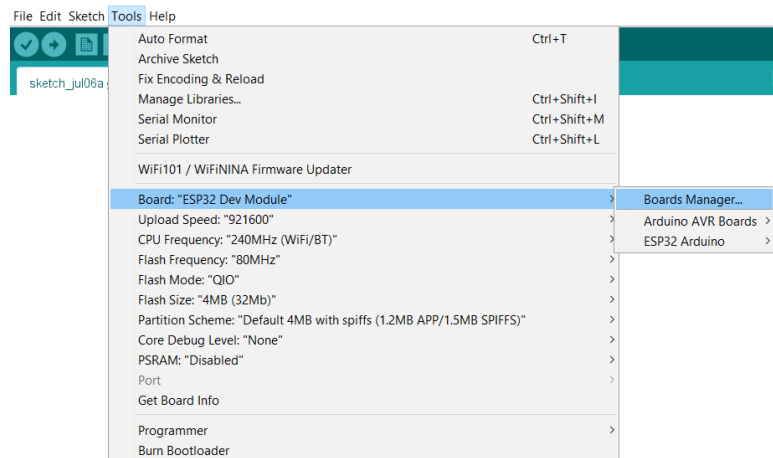
2. korak- Nakon toga potrebno je u novo otvorenom prozoru u polje (engl.*Additional_Boards_Manager_URL*) upisati link: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json te pritisnuti tipku OK



Slika 13. Instalacija ESP32 razvojne pločice

Izvor: Autor

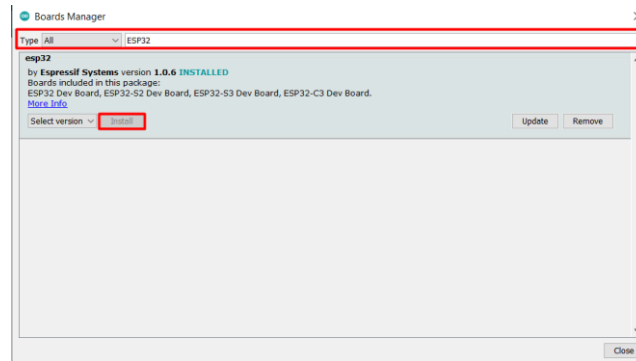
3. korak- U gornjem kutu kliknuti na *tools*, zatim otvoriti *Board* i kliknuti na *Boards manager*



Slika 14. Instalacija ESP32 razvojne pločice

Izvor: Autor

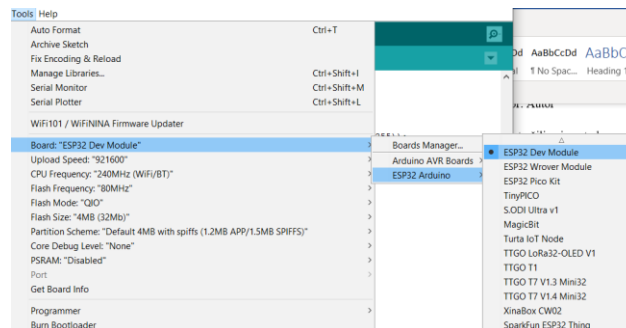
4. korak- Nakon otvaranja novog prozora u tražilicu potrebno je upisati „ESP32“, te nakon toga pritisnuti tipku *install*.



Slika 15. Instalacija ESP32 razvojne pločice

Izvor: Autor

5. korak- Nakon instalacije ESP32 razvojne pločice potrebno ju je izabrati kao primarnu kako bi se program mogao prebaciti na nju.



Slika 16. Instalacija ESP32 razvojne pločice

Izvor: Autor

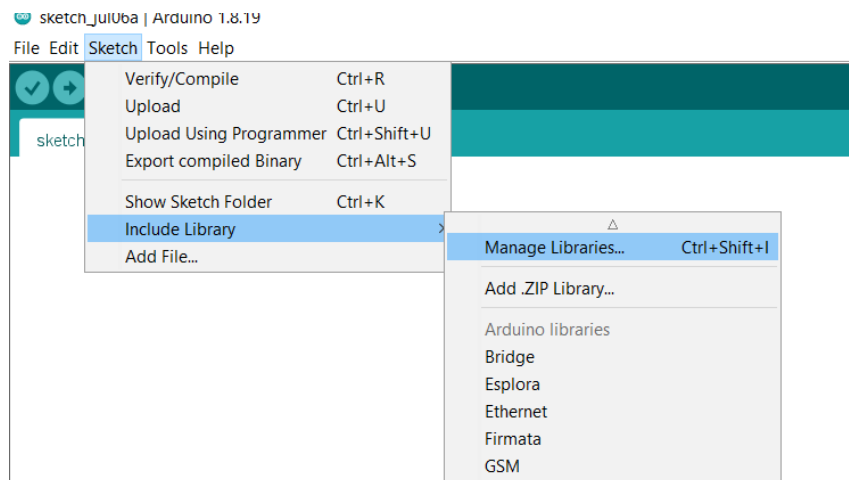
6.3 Instalacija biblioteka

Za rad programa prvo je potrebno instalirati određene biblioteke u Arduino IDE sučelju.

Biblioteke u Arduino IDE sučelju su već gotove kolekcije kôdova koje nam omogućuju lakše povezivanje i upravljanje sensorima.

Za ovaj završni zadatak potrebne su dvije biblioteke: „FastLED_NeoMatrix.h“ i „arduinoFFT.h“ instalacija istih prikazana je u sljedećim koracima.

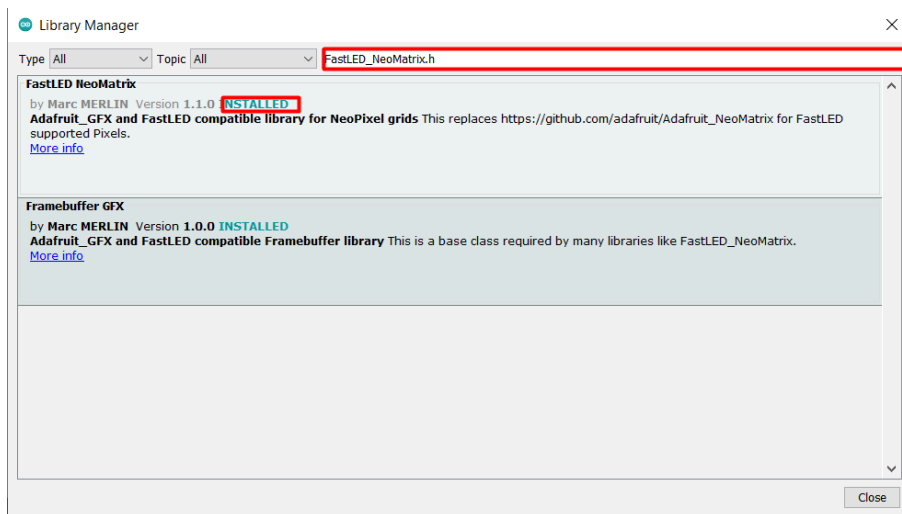
1. korak- U Arduino IDE sučelju potrebno je kliknuti na „Sketch“, nakon toga kliknuti na „Include Library“ te na kraju kliknuti na „Manage Libraries“



Slika 17. Instalacija biblioteka

Izvor: Autor

2. korak- potrebno u novootvorenom prozoru u tražilicu upisati ime biblioteke, te ju instalirati.



Slika 18. Instalacija biblioteka

Izvor: Autor

7. OBJAŠNJENJE KÔDA

7.1 Definiranje parametara i pinova

U prvom dijelu programa potrebno je definirati biblioteke (*libraries*) te sve potrebne parametre i pinove koji se koriste na ESP32 razvojnoj pločici.

```
#include <FastLED_NeoMatrix.h>

#include <arduinoFFT.h>

#define UZORCI 1024

#define FREKVENCIJA_RAZDOBLJA_PROMATRANJA 40000

#define AMPLITUDA 100000

#define ULAZNI_AUDIO_PIN 35

#define LED_PIN 5

#define RASPORED_BOJA GRB

#define CHIPSET WS2812B

#define MAX_MILIAMPERA 2000

Const int POSTAVKE_SVJETLINE = 100;

#define ELEKTRICNI_NAPON_LED 5

#define BROJ_POJASEVA 16

#define VANJSKI_ŠUM 500
```

Kôd 1. Program za definiranje parametara i pinova

Izvor: Autor

7.2 Definiranje matrice

Dio programa kojim se definira veličina matrice.

```
const uint8_t SirinaMatrice = 16;
const uint8_t VisinaMatrice = 12 ;
#define BROJ_LEDICA (SirinaMatrice * VisinaMatrice)
#define SIRINA_POJASA (SirinaMatrice / (BROJ_POJASEVA - 1))
#define VRH_MATRICE (VisinaMatrice - 0)
```

Kôd 2. Program za definiranje veličine matrice

Izvor: Autor

7.3 Definiranje tipova podataka

Ovaj dio programa definira *Peak*⁴ i konstruktor koji sadrži četiri argumenta te dva *double array*.⁵

```
unsigned int perioda_uzorka_us;
byte peak[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
int PrijasnjaVisinaSnopa[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
int VrijednostFrekvPojasa []={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
double StvrnoPromatranje[UZORCI];
double Spremnik[UZORCI];
unsigned long NovoVrijemePromatranja;
arduinoFFT FFT = arduinoFFT(StvarnoPromatranje, Spremnik, UZORCI,
FREKVENCIJA_RAZDOBLJA_PROMATRANJA);
```

Kôd 3. Program za definiranje tipova podataka i konstruktora

Izvor: Autor

⁴*Peak*- najveća vrijednost koju postiže zvučni tlak.

⁵*Double array*- skup podataka zapisanih u linearnom obliku koji sadrži dvodimenzionalnu listu podataka.

7.4 Postavljanje FastLED parametara

Kako bi se moglo upravljati parametrima kao što je svjetlina, maksimalna snaga i samim LED diodama potrebno je kreirati funkciju koja se poziva nakon pokretanja programa i povezati funkcije *FastLED* biblioteke s već postavljenim parametrima.

Funkcija također sadrži *perioda_uzorka_us* koja služi kako bi postavili vrijeme između svake detekcije zvuka koja se šalje na obradu u mikrokontroler.

```
void setup() {  
  Serial.begin(115200);  
  
  FastLED.addLeds<CHIPSET, LED_PIN, RASPORED_BOJA>(LED_diode, BROJ_LEDICA)  
  .setCorrection(TypicalMD5050);  
  
  FastLED.setMaxPowerInVoltsAndMilliamps(ELEKTRICNI_NAPON_LED,  
  MAX_MILIAMERA);  
  
  FastLED.setBrightness(POSTAVKE_SVJETLINE);  
  
  FastLED.clear();  
  
  perioda_uzorka_us=round(1000000*(1.0/FREKVENCIJA_RAZDOBLJA_PROMATRANJA)  
  );  
}
```

Kôd 4. Program za postavljanje FastLED parametri

Izvor: Autor

7.5 Prikupljanje uzoraka za obradu

Za prikupljanje uzoraka za obradu prvo je potrebno spremi analogni uzorak preko ULAZNI_AUDIO_PIN, nakon toga potrebno je proći 25 μ s kako bi se snimio sljedeći uzorak te se taj proces ponavlja sve dok se ne spremi 1024 uzorka.

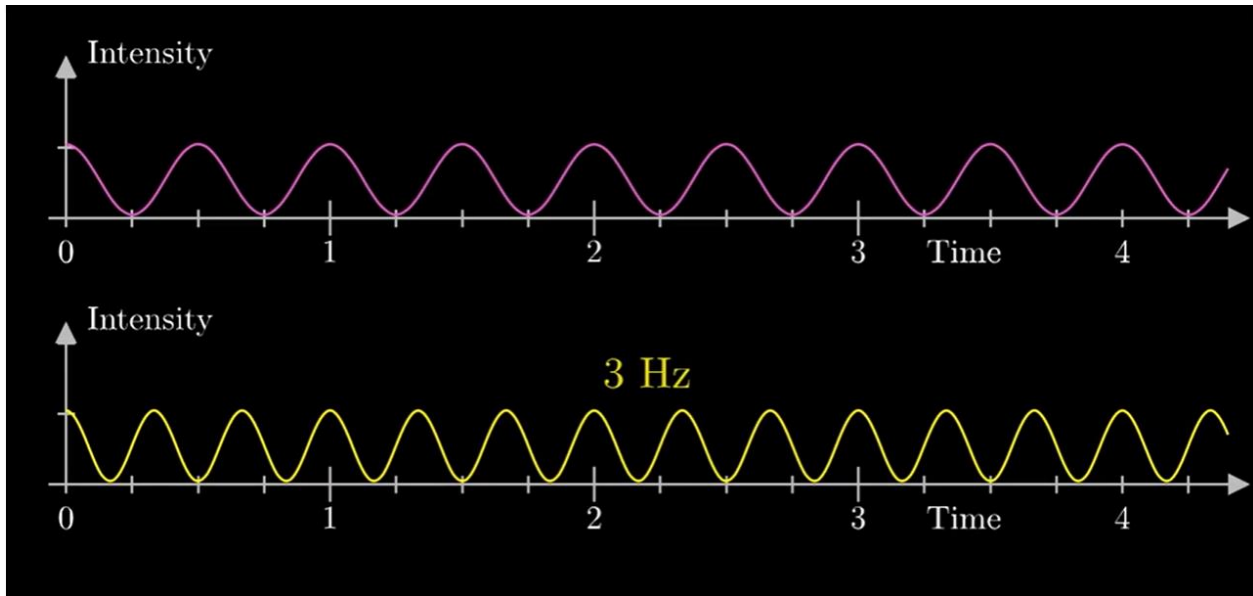
```
void loop() {  
  for (int i = 0; i<BROJ_POJASEVA; i++){  
    VrijednostFrekvPojasa[i] = 0;  
  }  
  for (int i = 0; i < UZORCI; i++) {  
    NovoVrijemePromatranja = micros();  
    StvrnoPromatranje[i] = analogRead(ULAZNI_AUDIO_PIN);  
    Spremnik[i] = 0;  
    while ((micros() - NovoVrijemePromatranja) < perioda_uzorka_us) {}  
  }  
}
```

Kôd 5. Program za prikupljanje uzoraka za obradu

Izvor: Autor

7.6 FFT (Fast Fourier Transform)

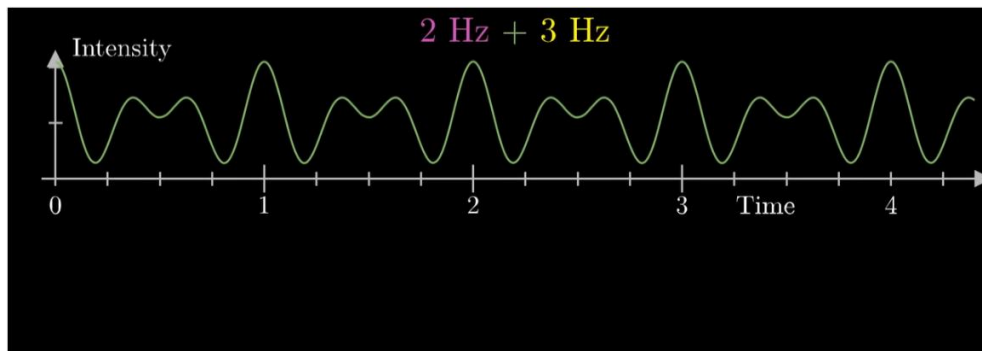
Fast Fourier Transform matematička je operacija koja povezuje određeni proizvedeni zvuk s njegovom frekvencijom. Posebna važnost ovog algoritma je njegova sposobnost da odvaja pomiješane signale (više tonova koji se izvode odjednom) te svakome od njih dodjeljuje njegovu frekvenciju.



Slika 19. Frekvencije od 2Hz i 3Hz prije spajanja

Izvor: <https://towardsdatascience.com/fast-fourier-transform-937926e591cb> (10.8.2022)

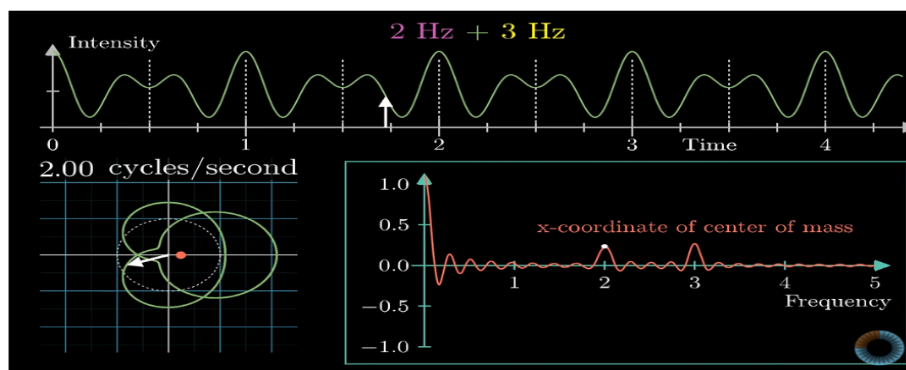
Ovo su signali od 2Hz i 3Hz nakon što su spojeni. Kako bi se svaki signal mogao prikazivati odvojeno potrebno ih je odvojiti pomoću *Fast Fourier Transform* algoritma.



Slika 20. Princip rada FFT

Izvor: <https://towardsdatascience.com/fast-fourier-transform-937926e591cb> (10.8.2022)

Način na koji se dva ili više signala odvajaju pomoću FFT-a je da se sinusoida sa slike iznad prebaci u kružni oblik, nakon toga je potrebno postaviti sredinu kružnog oblika (sredinu mase). Sredina mase je crvena točka i crvena sinusoida na slici ispod, vidljivo je da se nakon transformacije kružnog uzorka u sinusoidu dobiva rezultat na 2Hz i 3Hz te na taj način dobivamo dvije frekvencije koje su bile pomiješane. Kružni oblik je ista sinusoida dvaju pomiješanih signala koja nam pomaže kako bi se dobila odskakanja na određenoj frekvenciji i time daje mogućnost prepoznavanja signala [4].



Slika 21. Odvajanje i prepoznavanje dvaju signala

Izvor: <https://towardsdatascience.com/fast-fourier-transform-937926e591cb> (11.8.2022)

U dijelu progama ispod prikazane su funkcije kojima FFT(engl. *Fast Fourier Transform*) algoritam obrađuje audio signale.

```
FFT.DCRemoval();  
FFT.Windowing(FFT_WIN_TYP_HAMMING, FFT_FORWARD);  
FFT.Compute(FFT_FORWARD);  
FFT.ComplexToMagnitude();
```

Kôd 6. Program FFT funkcija koje odvajaju signale

Izvor: Autor

7.7 Dodjela frekvencijskih raspona

Za prikaz frekvencija potrebno je svakoj LED traci dodati frekvencijski pojas koji prikazuje. Prikaz frekvencijskih pojasa je zrcaljen što znači da je prvi frekvencijski pojas isti zadnjemu, drugi je isti predzadnjem...

```
if(i>7 && i<=9) VrijednostFrekvPojasa[0] += StvrnoPromatranje[i];
if(i>9 && i<=13) VrijednostFrekvPojasa[1] += StvrnoPromatranje[i];
if(i>13 && i<=18) VrijednostFrekvPojasa[2] += StvrnoPromatranje[i];
if(i>18 && i<=25) VrijednostFrekvPojasa[3] += StvrnoPromatranje[i];
if(i>25 && i<=36) VrijednostFrekvPojasa[4] += StvrnoPromatranje[i];
if(i>36 && i<=50) VrijednostFrekvPojasa[5] += StvrnoPromatranje[i];
if(i>50 && i<=69) VrijednostFrekvPojasa[6] += StvrnoPromatranje[i];
if(i>69 && i<=97) VrijednostFrekvPojasa[7] += StvrnoPromatranje[i];
if(i>69 && i<=97) VrijednostFrekvPojasa[8] += StvrnoPromatranje[i];
if(i>50 && i<=69) VrijednostFrekvPojasa[9] += StvrnoPromatranje[i];
if(i>36 && i<=50) VrijednostFrekvPojasa[10] += StvrnoPromatranje[i];
if(i>25 && i<=36) VrijednostFrekvPojasa[11] += StvrnoPromatranje[i];
if(i>18 && i<=25) VrijednostFrekvPojasa[12] += StvrnoPromatranje[i];
if(i>13 && i<=18) VrijednostFrekvPojasa[13] += StvrnoPromatranje[i];
if(i>9 && i<=13) VrijednostFrekvPojasa[14] += StvrnoPromatranje[i];
if(i>7 && i<=9) VrijednostFrekvPojasa[15] += StvrnoPromatranje[i];
```

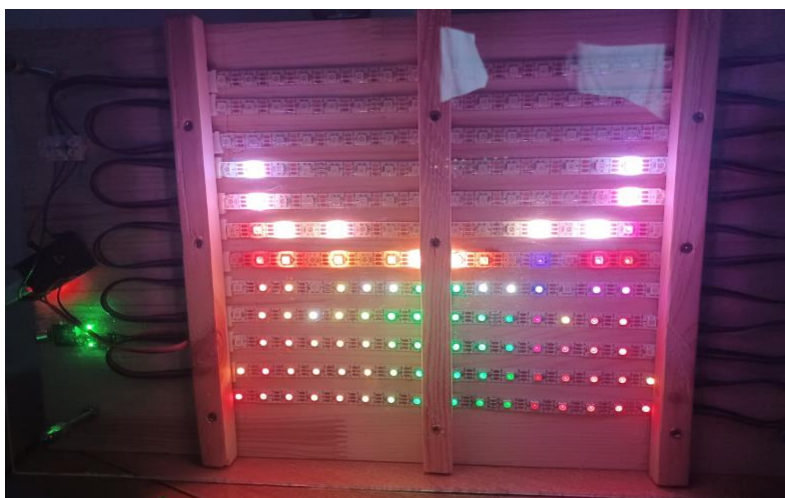
Kôd 7. Program za dodjelu frekvencijskih pojasa LED trakama

Izvor: Autor

8.PRIKAZ VIZUALIZATORA ZVUKA

Slike ispod prikazuju raspon frekvencija koje je vizualizator prepoznao u trenutku slikanja.

Visine snopova prikazuju izraženost i glasnoću svake frekvencije u određenom trenutku promatranja.



Slika 22. Prikaz rada vizualizatora

Izvor: Autor



Slika 23. Prikaz rada vizualizatora

Izvor: Autor

9. ZAKLJUČAK

Korištenjem mikrokontrolera za izradu projekata s vanjskim modulima i dodacima može se naučiti puno o primjeni, upotrebi i programiranju istih. Korištenje mikrokontrolera također pokazuje koliko mali i jeftini uređaj može donijeti razvoju projekata raznih veličina. Jedan od najboljih primjera upotrebe istih je IoT (engl. *Internet of thing*), koji omogućuje povezivanje raznih uređaja u jednu cjelinu kako bi se olakšalo i sinkroniziralo njihovo upravljanje. Ovakvi projekti idealni su za uvod u svijet mikrokontrolera te za napredovanje kod shvaćanja njihovog rada i načina programiranja.

Ovaj vizualizator zvuka moguće je poboljšati izradom jednostavne mobilne aplikacije, kojom bi se mogli mijenjati neki parametri poput jačine svjetlosti, načina na koji se prikazuju frekvencije audio signala, promjena boja LED snopova.

Izrada ovog rada pružila je zanimljive izazove kako kod sklapanja samog *hardware-a* tako i kod pisanja kôda te je potrebno uložiti vremena, upornosti i puno istraživanja za izgradnju ovakvog projekta. Kod pisanja kôda potrebno se upoznati s određenim matematičkim operacijama, kao i sa zvukom i frekvencijama koje su jako zanimljiva tema koja skriva mnoge tajne i koje vrhunski stručnjaci još uvijek otkrivaju.

„Ako želite shvatiti tajne svemira, razmišljajte u terminima energije, frekvencije i vibracije.“

-Nikola Tesla

POPIS LITERATURE

[1.] ESPRESSIF, ESP32 MCU (20.6.2022.)

<https://www.espressif.com/en/products/socs/esp32>

[2.] Branch Education, What is sound? (10.7.2022)

<https://www.youtube.com/watch?v=24yESm63tSY>

[3.] Britannica, Frequency (17.7.2022.)

<https://www.britannica.com/science/frequency-physics>

[4.] Cory Maklin, Fast Fourier Transform (10.8.2022.)

<https://towardsdatascience.com/fast-fourier-transform-937926e591cb>

POPIS PROGRAMSKIH KÔDOVA

Kôd 1. Program za definiranje parametara i pinova.....	25
Kôd 2. Program za definiranje veličine matrice.....	26
Kôd 3. Program za definiranje tipova podataka i konstruktora.....	27
Kôd 4. Program za postavljanje FastLED parametri.....	28
Kôd 5. Program za prikupljanje uzoraka za obradu	29
Kôd 7. Program za dodjelu frekvencijskih pojasa LED trakama	33

POPIS SLIKA

Slika 1. Raspored pinova ESP32 razvojne pločice.....	9
Slika 2. WS2812B LED TRAKA	10
Slika 3. Veza između LED traka	11
Slika 4. Modul detektora zvuka.....	12
Slika 5. Primjer kompresije i dekompresije	13
Slika 6. Amplituda zvučnog vala od 829 HZ	15
Slika 7. Vizualizator zvuka	17
Slika 8. Skidanje Arduino IDE sučelja.....	18
Slika 9. Instalacija Arduino IDE	19
Slika 10. Instalacija Arduino IDE	19
Slika 11. Instalacija Arduino IDE	19
Slika 12. Instalacija ESP32 razvojne pločice	20
Slika 13. Instalacija ESP32 razvojne pločice	21
Slika 14. Instalacija ESP32 razvojne pločice	21
Slika 15. Instalacija ESP32 razvojne pločice	22
Slika 16. Instalacija ESP32 razvojne pločice	22
Slika 17. Instalacija biblioteka	23
Slika 18. Instalacija biblioteka	24
Slika 19. Frekvencije od 2Hz i 3Hz prije spajanja.....	30
Slika 20. Pricip rada FFT	31
Slika 21. Odvajanje i prepoznavanje dvaju signala.....	31
Slika 22. Prikaz rada vizualizatora	34
Slika 23. Prikaz rada vizualizatora	34

POPIS TABLICA

Tablica 1. Razina glasnoće u decibelima.....17