

# FREKVENCIJSKI ANALIZATOR AUDIO SIGNALA TEMELJEN NA ARDUINO PLATFORMI

---

**Greblo, Jazz Richard**

**Undergraduate thesis / Završni rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:219127>

*Rights / Prava:* [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

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



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**FREKVENCIJSKI ANALIZATOR AUDIO SIGNALA TEMELJEN  
NA ARDUINO PLATFORMI**

Rijeka, rujan 2023.

Jazz Richard Greblo

0069090823

SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**FREKVENCIJSKI ANALIZATOR AUDIO SIGNALA TEMELJEN  
NA ARDUINO PLATFORMI**

Mentor: Doc. dr. sc. Ivan Volarić

Rijeka, rujan 2023.

Jazz Richard Greblo

0069090823

Rijeka, 20. ožujka 2023.

Zavod:           Zavod za automatiku i elektroniku  
Predmet:        Elementi automatizacije postrojenja  
Grana:           2.03.03 elektronika

## ZADATAK ZA ZAVRŠNI RAD

Pristupnik:    **Jazz Richard Greblo (0069090823)**  
Studij:        Sveučilišni prijediplomski studij elektrotehnike



Zadatak:       **Frekvencijski analizator audio signala temeljen na Arduino platformi /  
Arduino based audio signal frequency analyzer**

### Opis zadatka:

Potrebno je izraditi sustav za mjerenje spektra audio signala temeljen na Arduino platformi. Ulazni signal je potrebno dobiti preko mikrofona (npr. MAX9814 modul), te korištenjem FFT biblioteke izračunati Fourierovu transformaciju signala. Izlaz iz sklopa potrebno je ostvariti preko 32x8 LED ekrana (npr. MAX7219 modul). Svaki stupac ekrana predstavlja određeni frekvenacijski raspon, a visina predstavlja amplitudu ulaznog signala na tom frekvenacijskom području.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

*Jazz Greblo*

Zadatak uručen pristupniku: 20. ožujka 2023.

Mentor:

Doc. dr. sc. Ivan Volarić

Predsjednik povjerenstva za  
završni ispit:

Prof. dr. sc. Dubravko Franković

## IZJAVA

Sukladno članku 8. Pravilnika o završnom radu završnom ispitu i završetku preddiplomskih sveučilišnih studija/ stručnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 1. veljače 2020. izjavljujem da sam samostalno izradio završni rada prema zadatku preuzetog dana 20. ožujka 2023.

Rijeka, 4. rujna 2023.



---

Jazz Richard Greblo



# SADRŽAJ

1.	UVOD .....	7
2.	ARUDINO .....	8
2.1.	Arduino Uno .....	9
2.2.	Osnovne funkcije u kodu .....	12
2.3.	Arduino biblioteke .....	12
2.4.	Analogno digitalna pretvorba .....	12
3.	MAX9814 MIKROFON MODUL .....	15
4.	FOURIEROVA TRANSFORMACIJA .....	17
4.1.	Brza Fourierova transformacija (FFT) .....	20
4.2.	Energija signala .....	23
5.	MAX7219 MODUL .....	25
5.1.	Matrični 8x8 zaslon .....	25
5.2.	MAX7219 upravljački sklop .....	27
5.3.	SPI protokol .....	27
6.	SHEMA I PRINCIP RADA .....	29
7.	PROGRAMSKI KOD .....	33
8.	ZAKLJUČAK .....	37
9.	LITERATURA .....	38
10.	SAŽETAK I KLJUČNE RIJEČI .....	39
10.1.	Sažetak .....	39
10.2.	Ključne riječi .....	39
11.	SUMMARY AND KEYWORDS .....	40
11.1.	Summary .....	40
11.2.	Keywords .....	40

## 1. UVOD

U sklopu ovog završnog rada izrađen je frekvencijski analizator signala temeljen na Arduino platformi. Ideja je mikrofonskim modulom uzrokovati audio signal te pomoću Fourierove transformacije razložiti signal na njegove osnovne frekvencije. Amplitude pojedinih frekvencija se zatim prikazuju na LED matičnom zaslonu u vidu visine stupaca koje predstavljaju pojedine frekvencije.

Korištena je Arduino razvojna pločica koja će uzrokovati audio signal i zapisivati ga u niz diskretnih vrijednosti nad kojim je izračunata Fourierova transformacija da bismo dobili uvid u frekvencijski sadržaj signala u nekom određenom vremenskom intervalu. Fourierova transformacija realizirana je koristeći Arduino biblioteku „FFT“ koja ima gotove funkcije za računanje diskretne Fourierove transformacije iz niza vrijednosti. To se treba sve odvijati relativno brzo kako bi se nakon računanja spektra signala određenog broja uzoraka - amplitude pojedinog signala odmah prikazivale na LED zaslonu u stvarnom vremenu bez nekog prevelikog kašnjenja.

U projektu je korišten Arduino Uno R3, mikrofonski modul s predpojačalom AGC Mic Amp - MAX9814, te LED Matrix 32x8 display - MAX7219.



## 2. ARUDINO

Arduino je ime za otvorenu računalnu i softversku tvrtku odnosno platformu koja omogućava dizajn elektroničkih sklopova koji između ostalog omogućuju spajanje računala s fizičkim svijetom preko senzora i prekidača. Tvrtka Arduino dizajnira i proizvodi razvojne pločice temeljene na mikrokontrolerima, kao i razne module koji se mogu spajati na njih. Njihovi hardverski proizvodi licencirani su pod licencom CC BY-SA (eng. *Creative Commons Attribution licence – Share alike*), a softver je licenciran pod GPL (eng. *General Public License*), koja dopušta proizvodnju Arduino ploča i distribuciju softvera.

Arduino projekt rođen je 2003. godine, kako bi se olakšao pristup i korištenje elektroničkih i programskih sredstava. Dizajniralo ga je nekoliko studenata s Instituta za interaktivni dizajn u Ivrei, Italija. Učinili su to kako bi studenti elektronike imali jeftiniju alternativu popularnom BASIC Stampu, razvojnim pločicama koje su tada koštale više od sto dolara, pa ih nije svatko mogao priuštiti.

Arduino razvojna pločica koristi razne tipove mikrokontrolera, te je opremljena setovima digitalnih i analognih ulazno/izlaznih (I/O) pinova koji se mogu spojiti na različite druge sklopove. Razvojne pločice imaju serijska komunikacijska sučelja, uključujući univerzalnu serijsku sabirnicu (USB), koja se također koriste za učitavanje programa.

Mikrokontroleri se mogu programirati korištenjem C i C++ programskih jezika (*Embedded C*), korištenjem standardnog API-ja (*Application Programming Interface*) koji je također poznat kao Arduino programski jezik. Uz korištenje tradicionalnih alatnih prevoditelja, projekt Arduino pruža integrirano razvojno okruženje (IDE) i alat naredbenog retka. Većina Arduino ploča sastoji se od Atmel 8-bitnog AVR mikrokontrolera (ATmega8, ATmega168, ATmega328, ATmega1280 ili ATmega2560) s različitim količinama flash memorije, pinova i značajki. Ploče koriste jednoredne ili dvoredne ženske konektore koji olakšavaju spajanje i ugradnju u druge sklopove. Na ploče se mogu povezati dodatni moduli koji se nazivaju štitovi. Većina ploča uključuje linearni regulator napona koji ulazni napon pretvara na potrebnih 5V ili 3.3V te kristalni oscilator od 16 MHz koji daje radni takt mikroprocesoru.

Mikrokontroler na Arduino razvojnoj pločici unaprijed su programirani s boot loaderom koji pojednostavljuje učitavanje programa u flash memoriju na mikrokontroleru. Arduino ploče

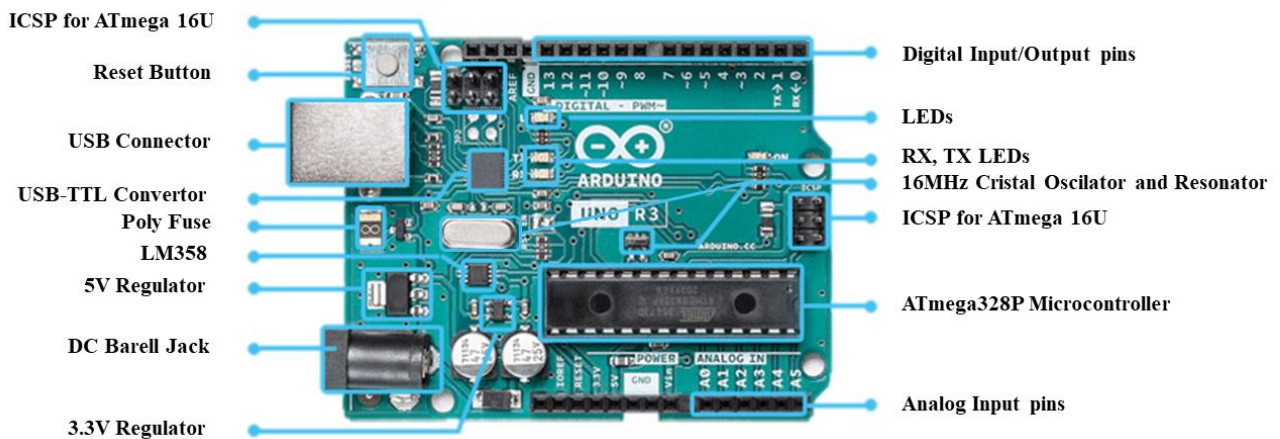
programiraju se putem USB-a, korištenjem USB-na-serijskih adapterskih čipova kao što je FTDI FT232.

Postoje mnoge inačice Arduino pločica dok su neke od korištenijih:

- Arduino Diecimila
- Arduino Uno R2 i R3
- Arduino Leonardo
- Arduino Micro
- Arduino Nano
- Arduino Mega
- Arduino Due

## 2.1. Arduino Uno

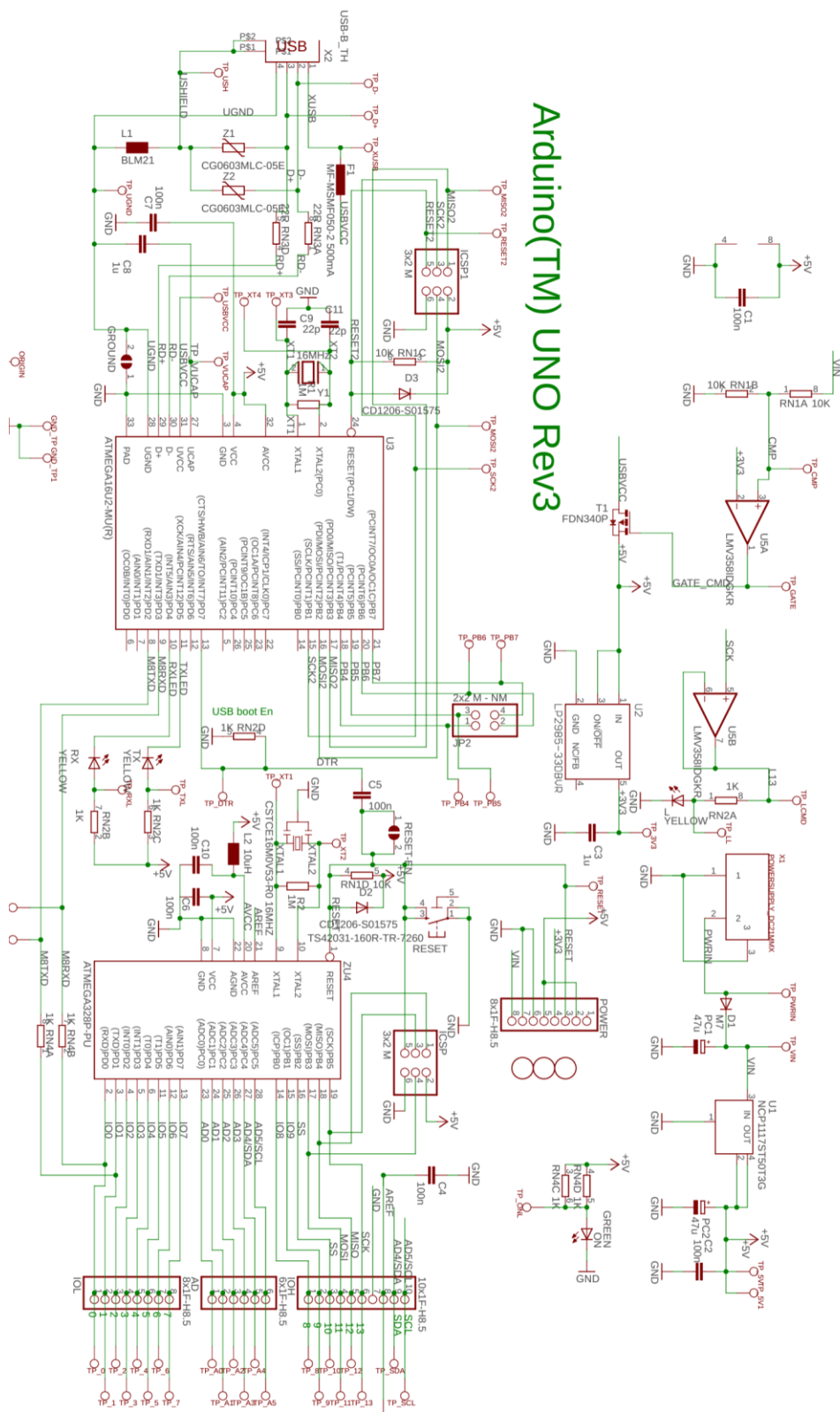
U ovom završnom radu je korištena verzija Arduino Uno R3 koja je jedna od najkorištenijih inačica Arduino razvojnih pločica. S obzirom na svoju fleksibilnost, jednostavnost korištenja i veliku zajednicu korisnika, Arduino Uno je izvrstan izbor za sve koji žele započeti s eksperimentiranjem u svijetu elektronike i programiranja. Na slici 2.1 su prikazani glavni dijelovi Uno-a koji su ukratko objašnjeni u tablici 2.1. Na slici 2.2 je prikazana shema svih dijelova na jednoj Arduino Uno razvojnoj pločici.



Slika 2.1 Prikaz Arduino Uno pločice [1]

Tablica 2.1. Dijelovi Arduino Uno-a

USB – B socket	USB connector ima dvije funkcije: jedna je komunikacija sa računalom te kako bi se napisani program učitavao na Arduino uz pomoć bootloader-a, a druga je napajanje same pločice
ICSP pins	Ovi pinovi se uobičajeno koriste za programiranje mikrokontrolera kada je bootloader nedostupan. Također omogućuju međusobno komunikaciju dviju ili više Arduino razvojnih pločica te vanjskih modula preko SPI protokola za komunikaciju.
Reset button	Ovaj tipkalo koristi se za resetiranje ATmega328 mikrokontrolera.
USB-TTL Interface Chip	Ovaj čip služi za komunikaciju sa računalom.
Crystal Oscillator/ Resonator	Kvarcni kristalni oscilator ili keramički rezonator služe za generiranje radnog takta mikrokontroleru, odnosno određuje koliko će se operacija izvršiti u sekundi.
LM358	Koristi se kao komparator za kontrolu ulazne snage. Kada se ulazna snaga osigurava kroz DC utor ili Vin pin, krug za kontrolu putanje napajanja isključit će USB pin za napajanje od strujnog kruga što će zapravo zaštititi USB priključak.
Voltage Regulator	Smanjuju ulazni napon koji može biti između 7 – 12 V koji se dovodi na Vin pin ili na DC utor, Jedan je regulator od 5 V za mikrokontrolere, a drugi je regulator od 3,3 V.
Digital and Analog I/O	Arduino Uno ima 14 digitalnih I/O pinova i 6 analognih ulaza. Digitalni I/O pinovi su logičke razine od pri čemu jedinica predstavlja 5V, a nula predstavlja 0V. Arduino Uno podržava 6-kanalne 10-bitne ADC ulaze kroz pinove A0-A5.
LED	Arduino Uno ima 4 LED diode. Jedna se koristi kao indikator napajanja, druge dvije se koriste za prikaz aktivnosti Rx i Tx pinova. Četvrta LED je vezana na digitalni pin 13, koji se može koristiti za testiranje Arduino ploče ili jednostavno kao indikator.
ATmega328P microcontroller	Arduino Uno koristi mikrokontroler sa 28 pinova, koji je unaprijed programiran s bootloaderom koji omogućuje izravno učitavanje programa na Arduino putem USB-a bez potrebe za vanjskim programatorom.



Slika 2.2. Shematski prikaz Arduino Uno R3 pločice [2]

## 2.2. Osnovne funkcije u programskom kodu

Funkcija `setup()` – ova funkcija se izvršava samo jednom pri pokretanju programa. U njoj se postavljaju inicijalne postavke i konfiguracije za Arduino razvojnu pločicu. Primjerice, konfiguracija digitalnih i analognih pinova, postavljanje brzine serijske komunikacije i slično.

Funkcija `loop()` – ova funkcija se izvršava neprekidno nakon izvršavanja `setup()` funkcije. Svi dijelovi koda koji se nalaze unutar `loop()` bloka izvršavaju se u beskonačnoj petlji. Ovdje se piše glavni dio programa koji se ponavlja sve dok Arduino razvojna pločica nije isključena.

## 2.3. Arduino biblioteke

Biblioteke su datoteke napisane u C ili C++ (.c, .cpp) koje u sebi imaju već gotove funkcije koje služe za olakšavanje pisanja programskog koda. Neke od funkcija biblioteka uključuju osnovne komunikacijske protokole i podršku za neke od najčešćih tipova hardvera poput servo motora i znakovnih LCD zaslona. U ovom projektu je korištena biblioteka „SPI“ za komunikaciju putem serijskog komunikacijskog protokola (SPI) sa LED zaslonom, zatim „MD\_MAXxx“ biblioteka koja omogućava ispisivanje riječi i crtanje po LED matičnom zaslonu te „FFT“ biblioteka koja ima definirane funkcije za izračunavanje brze Fourierove transformacije iz diskretnog niza.

## 2.4. Analogno digitalna pretvorba

Arduino Uno ima 6 analogno-digitalno (AD) konverzijskih kanala koji se mogu koristiti za čitanje analognog signala u rasponu od 0 do vrijednosti na koje je spojen pin Aref (unaprijed je postavljen na 5V ukoliko Aref nije spojen). U Arduino IDE-u, koristimo funkciju `analogRead()` kako bismo čitali ulazni analogni signal povezan s određenim pinom. Ova funkcija uzorkuje signal i vraća digitalnu vrijednost koja predstavlja izmjerenu vrijednost analognog ulaza. Arduino Uno ima 10-bitni AD pretvarač (ADC) što znači da će dati digitalnu vrijednost u rasponu od 0 - 1023 ( $2^{10} - 1$ ).

Rezolucija označava najmanju promjenu vrijednosti koja se može prepoznati i tako uzrokuje promjenu digitalnog izlaza. Ona se računa po slijedećem izrazu

$$A_{ref} / ((2^n) - 1) \quad (2.1)$$

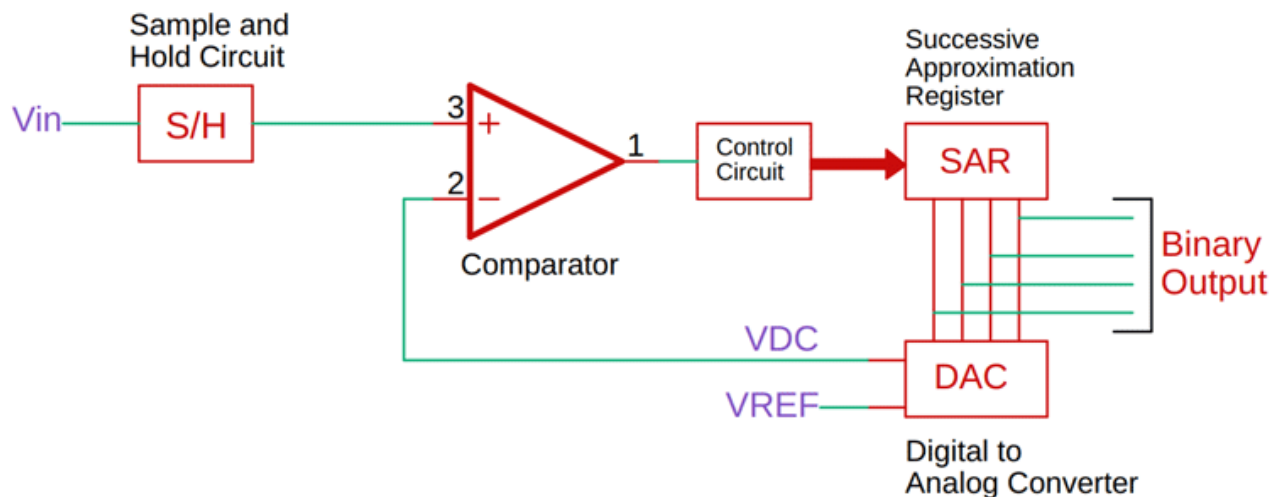
gdje je:

$A_{ref}$  je vrijednost na koju je spojen pin za referentnu vrijednost napona AD pretvorbe

$n$  - broj bitova pretvarača.

Za maksimalan napon od 5V najmanja promjena vrijednosti koju Arduino Uno može očitati je 4.89mV. Promjenom referentnog napona na manje vrijednosti moguće je očitavati i puno manje promjene napona čime se povećava točnost mjerenja signala malih vrijednosti.

AD pretvorba se na Arduinou odvija metodom sukcesivne aproksimacije. Shema sklopa koji taj proces provodi prikazan je na slici 2.2. Sastoji se od sklopa uzorkovanja i zadržavanja, komparatora, digitalno-analognog pretvarača (DAC), registra za sukcesivnu aproksimaciju te kontrolni krug.



Slika 2.2 Shema ADC-a za metodu sukcesivne aproksimacije [3]

Kad god započne nova pretvorba, sklop uzorkovanja i zadržavanja uzorkuje ulazni signal te se taj signal uspoređuje sa određenim izlaznim signalom DAC-a. Primjer – ulazni napon je 1.76V, referentan napon 5V i rezolucija je 4-bitna. U prvom taktu se *most significant bit* (MSB) postavlja u 1 te ostali bitovi u 0 i binarna kombinacija 1000 se digitalno analognim pretvaračem pretvara 2.5V odnosno pola od maksimalnog napona. Pošto je ulazni napon manji od 2.5 V MSB se postavlja u 0 i

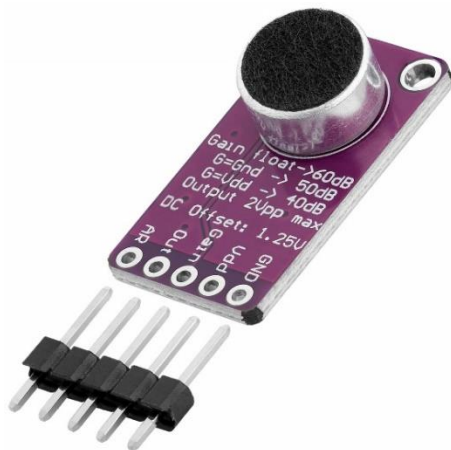
slijedeći bit se postavlja u 1. Binarna kombinacija 0100 nakon pretvorbe u DAC-u ( $2.5/2 = 1.25V$ ) u usporedbi s ulaznim naponom je manja i onda se za taj bit ostavlja 1. Dalje kombinacija 0110 pretvorbom ( $1.25 + 1.25/2 = 1.875V$ ) predstavlja veći napon od ulaznog i taj bit se onda vraća na 0. Zadnja kombinacija – binarno 0101 – analogno ( $1.25 + 1.25/2/2 = 1.5625V$ ) predstavlja manji napon od ulaznog i taj bit ostaje 1. To nam daje rezultantu binarnu kombinaciju 0101 odnosno aproksimirani napon od 1.5625 sa 4-bitnim pretvaračem.

Broj radnih taktova koji su potrebni za jednu AD pretvorbu jednak je broju bitova pretvarača odnosno broj radnih taktova raste s povećanjem rezolucije. Korištenom mikrokontroleru je potrebno 13 radnih taktova za jednu konverziju, a pošto je frekvencija radnog takta AD pretvornika 125kHz, za jednu pretvorbu je potrebno 104us. To znači da je maksimalna frekvencija uzorkovanja 9615Hz. Prema podatkovnoj tablici ATmega328P, preporučena maksimalna frekvencija radnog takta ADC-a ograničena je zbog internih dijelova u krugu pretvorbe.

### 3. MAX9814 MIKROFON MODUL

Mikrofoni su elektroakustički pretvarači koji pretvaraju akustičku energiju u električnu. Svi mikrofoni posjeduju neku vrstu membrane ili pokretnu površinu koju pobuđuje zvučni val. Odgovarajući izlaz je napon koji odgovara amplitudi zvučnog vala u vremenu. U ovom radu korišten je MAX9814 modul koji ima takozvani *electret* mikروفon odnosno kondenzatorski mikروفon koji je često korišten u ovakvim modulima zbog njegove male cijene i dimenzija.

MAX9814 je jeftino, visokokvalitetno mikrofonsko pojačalo s automatskom kontrolom pojačanja (AGC) i niskošumnim prednaponom mikrofona. Uređaj ima pretpojačalo niske razine šuma, pojačalo s promjenjivim pojačanjem (VGA), izlazno pojačalo, generator prednapona mikrofona i AGC upravljački krug. Niskošumno pretpojačalo ima fiksno pojačanje od 12 dB, dok se VGA pojačanje automatski prilagođava od 20 dB do 0 dB, ovisno o izlaznom naponu i AGC pragu. Pojačanje izlaznog pojačala može se podesiti prespajanjem pina Gain na pin VCC ili GND čime se postižu 3 vrijednosti izlaznog pojačanja. Izlazno pojačalo nudi izborna pojačanja od 8dB, 18dB i 28dB, a kaskadom pojačala dobiva se ukupno pojačanje od 40 dB, 50 dB ili 60 dB. Vanjski otpornički djelitelj napona kontrolira prag AGC-a, a jedan kondenzator programira vrijeme napada/otpuštanja. Vrijeme zadržavanja AGC-a je fiksno na 30 ms. MAX9814 je dostupan u 14-pinskom TDFN kućištu koji štedi prostor. Ovaj uređaj je specificiran za prošireni temperaturni raspon od  $-40^{\circ}\text{C}$  do  $+85^{\circ}\text{C}$ . Neki od važnijih podataka vezanih uz električne karakteristike mikrofonskog pojačala prikazane su u tablici 3.1 dok je modul je prikazan na slici 3.1.



Slika 3.1. Mikrofonsko pojačalo MAX9814 [4]



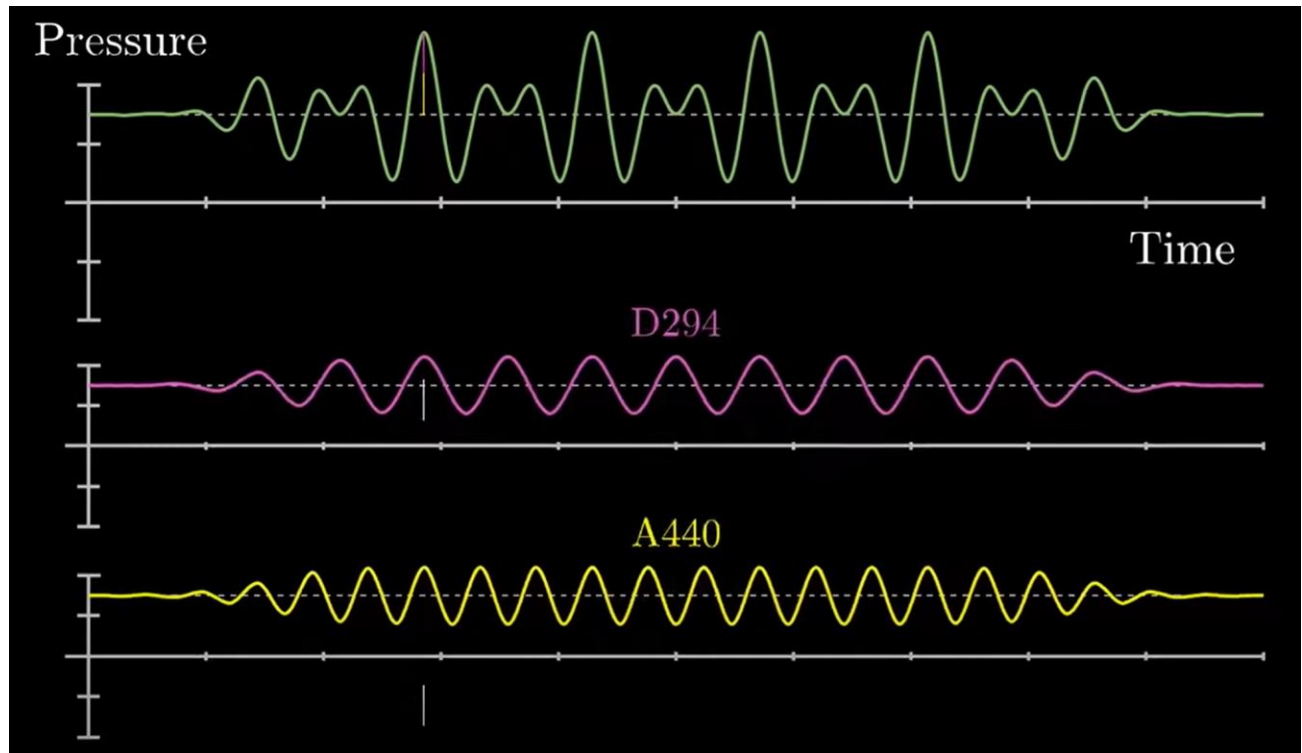
Tablica 3.1. Električne karakteristike mikrofonskog modula [5]

PARAMETAR	SIMBOL	UVJET	MIN	TIPIČNO	MAX	MJERNA JEDINICA
Napon napajanja	$V_{DD}$		2.7		5.5	V
Stuja napajanja	$I_{DD}$			3.1	6	mA
Izlazni šum	N	BW = 20kHz		430		$\mu V_{RMS}$
Aktivni raspon pojačanja	DR			60		dB
THD (Total Harmonic Distorsion) + šum	THD + N	$f_{IN} = 1\text{kHz}$ , BW = 20Hz do 20kHz, $R_L = 10\text{k}\Omega$		0.04		%
DC prednapon pojačala	$V_{IN}$		1.14	1.23	1.32	V
Ulazna impedancija	$Z_{IN}$			100		$\text{k}\Omega$
Attack/Release Ratio	A/R	A/R = GND	1:500			ms/ms
		A/R = VDD	1:2000			
		A/R odspojeno	1:4000			
Pojačanje	A	GAIN = VDD	39.5	40	40.5	dB
		GAIN = GND	49.5	50	50.6	
		GAIN odspojeno	59.5	60	60.5	

## 4. FOURIEROVA TRANSFORMACIJA

Fourierova transformacija (FT) je matematički model kojim se kontinuirani signal može pretvoriti iz vremenske domene u frekvencijsku domenu. Fourierova transformacija rastavlja signal na zbroj sinusoidnih baznih funkcija.

Audio signal je složeni signal sastavljen od više jedno-frekventnih zvučnih valova koji se očituje kao promjena tlaka u mediju. Kad se zvuk snima, snimamo samo rezultantne amplitude tih višestrukih valova, a ne pojedinačne amplitude svakog zvučnog vala zasebno. Na slici 4.1 prikazana su dva zvučna vala jedan frekvencije 440Hz te jedan frekvencije 294Hz te je zelenom bojom prikazan zvučni signal njihove sume.



Slika 4.1 Signal rastavljen na osnovne frekvencije [6]

Formula za direktnu Fourierovu transformaciju računa se prema slijedećem izrazu

$$F(\omega) = \int_{-\infty}^{\infty} x(t) * e^{-j\omega t} dt \quad (4.1)$$

gdje je:

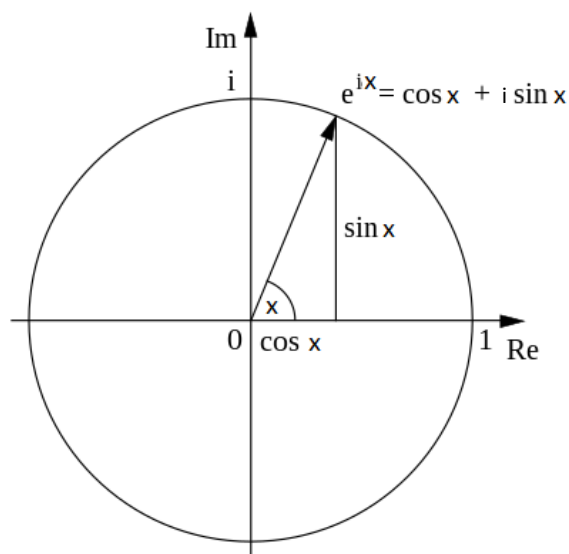
$F(\omega)$  Fourierova transformacija od funkcije  $x(t)$

$e$  je Eulerov broj

$\omega$  je kružna frekvencija odnosno  $2\pi * f$  (*f* je frekvencija).

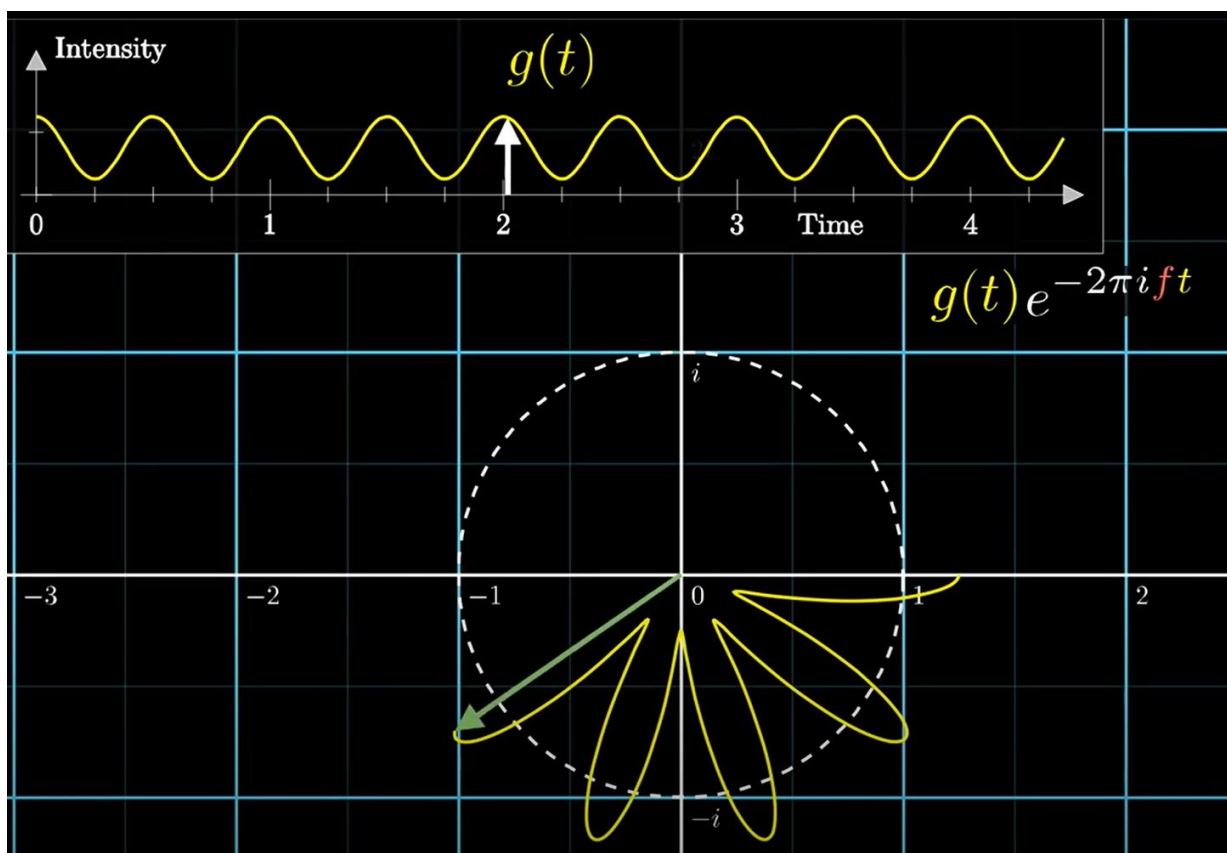
Za bolje razumijevanje izraza potrebno je razumjeti Eulerove formule vezane uz kompleksnu ravninu brojeva. Izraz 4.2 odnosno slika 4.2 se može predočiti na način da je funkcija  $e^{ix}$  rotirajući vektor, koji rotira oko ishodišta kompleksne ravnine tijekom čega  $x$  poprima vrijednosti iz domene realnih brojeva. U tom smislu  $x$  je kut koji stvara kružni isječak, koji spaja ishodište koordinatnog sustava u kompleksnoj ravnini s odgovarajućom točkom na jediničnoj kružnici, s pozitivnom djelom realnom osi. Pri tome taj vektor u kompleksnoj ravnini, rotira smjerom suprotno od smjera kazaljki na satu, a veličina kuta iskazuje se u radijanima.

$$e^{ix} = \cos(x) + j\sin(x) \quad (4.2)$$



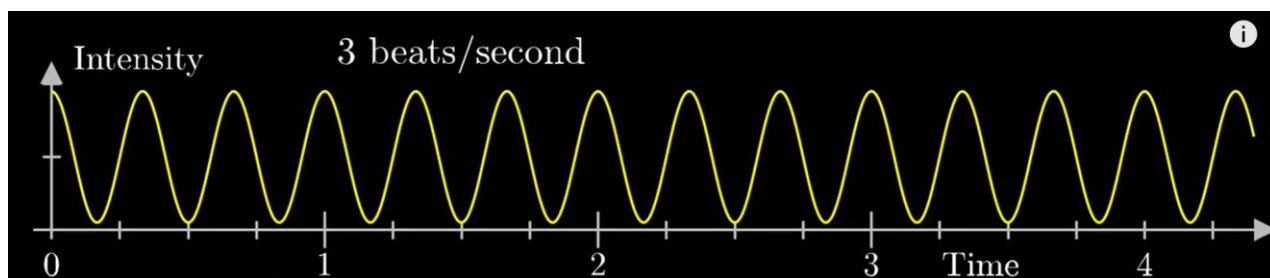
Slika 4.2 Eulerova formula [7]

Fizikalno FT može se predočiti kao računanje središta mase oblika koji nastaje kada se funkcija koja se transformira određenom frekvencijom namata oko ishodišta jedinične kružnice. Namatanje funkcije oko ishodišta odnosno množenje funkcije u vremenu sa izrazom  $e^{-j\omega t}$  prikazano je na slici 4.3.

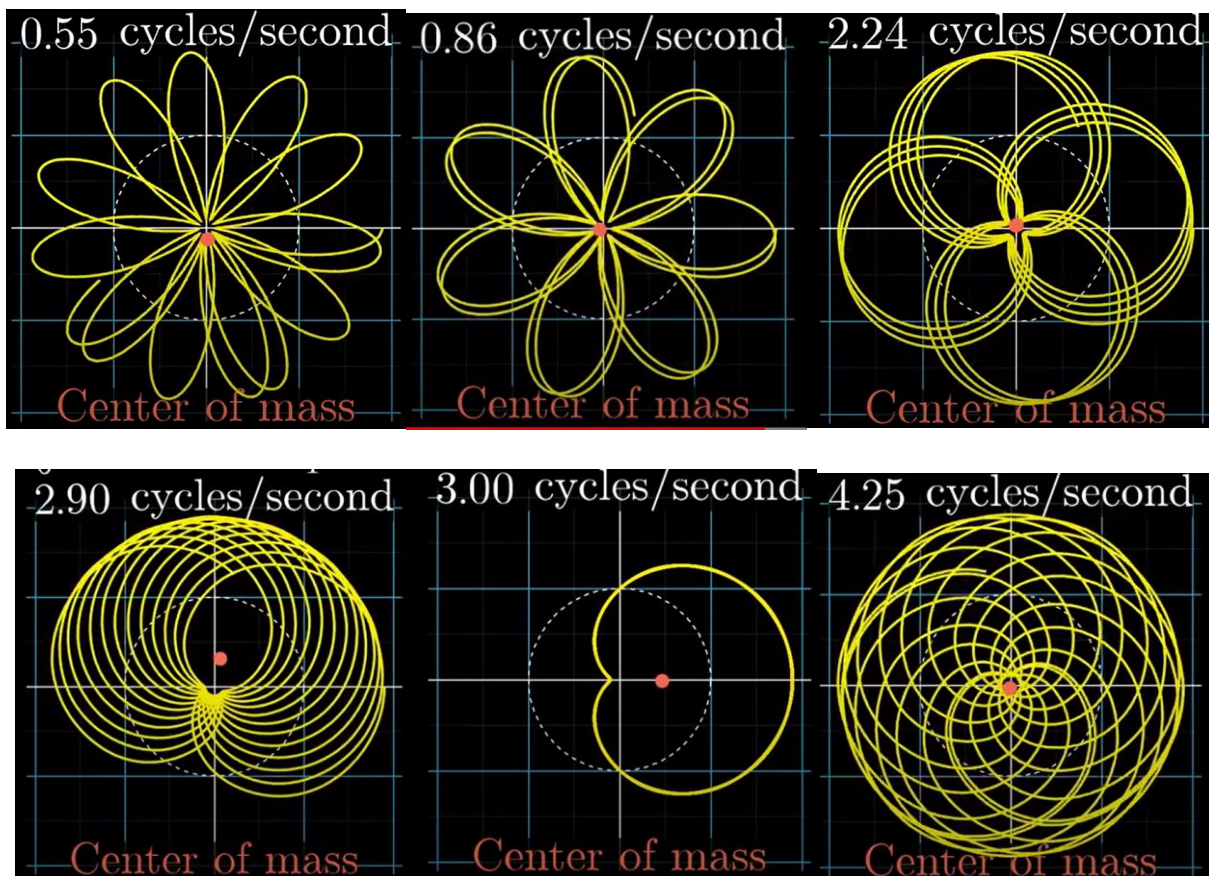


Slika 4.3 Namatanje funkcije  $g(t)$  oko ishodišta jedinične kružnice [6]

Ovisno o frekvenciji kojom namatamo ulazni signal dobivamo različite grafove, a time i različita središta mase. Središte mase oblika dobiva se tako da se integrira ta namotana funkcija i time upravo dobivamo izraz 4.1 za Fourierovu transformaciju. Prikaz središta ili centra mase na slici 4.4 pokazuje kako se središte mase kada se graf iscrtava bilo kojom frekvencijom osim frekvencije ulaznog signala ne mijenja previše i iznosi 0, a kada se graf namotava frekvencijom blizu ili točno 3Hz koja je ujedno i ulazna frekvencija signala iz primjera onda imamo promjenu središta mase. I upravo tim matematičkim modelom možemo vidjeti koje su frekvencije najzastupljenije u nekom signalu te kolika im je amplituda.



Slika 4.4 Ulazni signal [6]



Slika 4.5 Prikaz središta mase za različite frekvencije namatanja ulaznog signala [6]

#### 4.1. Brza Fourierova transformacija (FFT)

Brza Fourierova transformacija (FFT) matematički je algoritam koji izračunava diskretnu Fourierovu transformaciju (DFT) danog niza. Jedina razlika između FT i FFT je ta što FT uzima kontinuirani signal dok FFT uzima diskretni signal kao ulaz. DFT pretvara niz amplituda (diskretni signal) u njegove frekvencijske sastavnice baš kao što FT čini za kontinuirani signal. U našem slučaju,

imamo niz amplituda koje su uzorkovane iz kontinuiranog audio signala. DFT definirana je slijedećim izrazom

$$X[k] = \sum_{n=0}^{N_0-1} x[n] e^{-j2\pi kn/N}$$

gdje je:

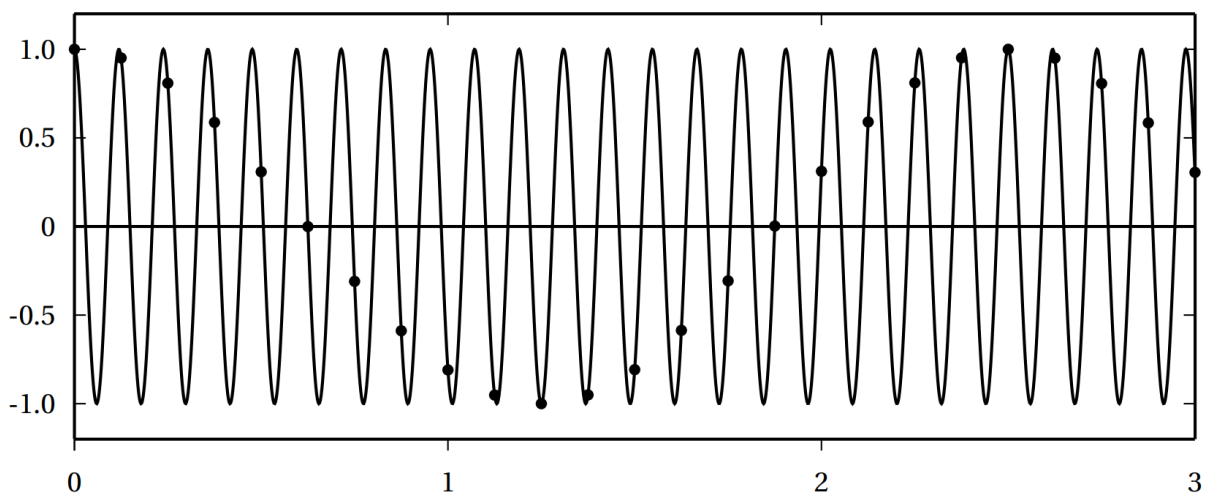
(4.3)

$X[k]$  diskretna Fourierova transformacija od niza  $x[n]$ ,

a  $N$  je broj uzoraka signala.

Fizikalno objašnjenje formule je skoro isto kao i za diskretnu transformaciju samo što se tu uzimaju diskretne točke nekog signala i množe se sa funkcijom  $e^{-jk\omega_0 Tn}$  te računa njihova suma što bi fizikalno opet predstavljalo izračunavanje centra mase diskretnog grafa točaka.

Kod uzorkovanja signala važno je znati maksimalnu frekvenciju vremenski kontinuiranog signala kojeg uzorkujemo kako ne bi došlo do pojave zvane „*Aliasing*“. Aliasing je pojava preklapanja frekvencijskih komponenti signala koja se događa zbog nedovoljno velike brzine uzorkovanja. Drugim riječima, uzrokuje pojavu frekvencija u spektru, kojih nema u izvornom signalu. Pojavu je najlakše shvatiti preko grafičkog prikaza.



Slika 4.6 Prikaz primjera aliasinga [8]

Na slici 4.6 prikazan je primjer aliasinga. Sinusoida sa frekvencijom 8400Hz ( $x(t) = \cos(2\pi \cdot 8400 \cdot t)$ , puna linija) se uzrokuje frekvencijom od 8000Hz, a uzrokovane vrijednosti (označene točkama) mogu se predočiti kao sinusoida frekvencije 400Hz. Iz toga je važno uzrokovati signal barem dva puta u jednoj periodi kako bi se točno moglo predočiti taj signal. To nam govori Nyquist-Shannonov teorem koji kaže da frekvencija uzorkovanja pri analogno-digitalnoj pretvorbi treba biti najmanje dvostruko veća od najviše frekvencije u izvornome analognom signalu.

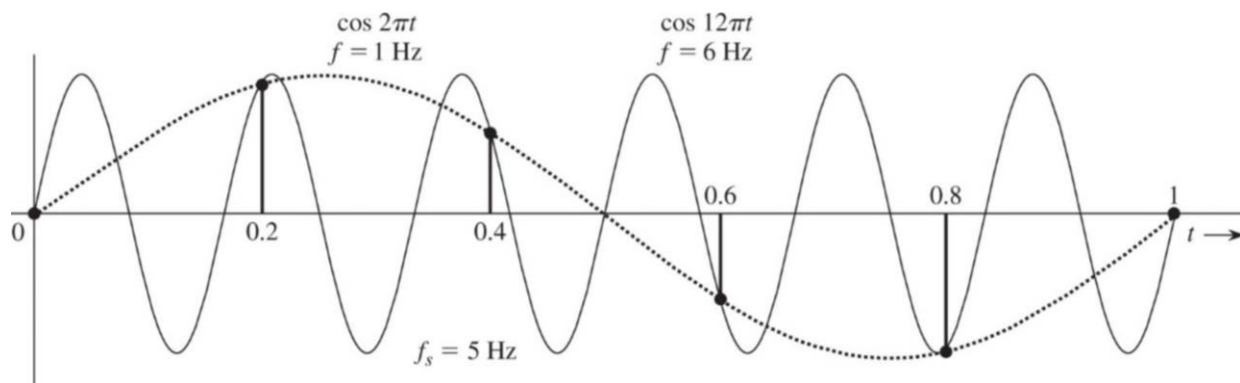
Dokaz tomu pokazan je u slijedećem izvodu. Uzorci kosinusa  $x(t) = \cos(\omega t)$  za frekvencije  $\omega_1 = \omega_s/2 + \omega_x$  i  $\omega_2 = \omega_s/2 - \omega_x$  identični su za frekvenciju uzorkovanja  $\omega_s$ .

$$X(nT) = \cos(\omega nT), \text{ za } \omega = \omega_s/2 \pm \omega_x.$$

$$\begin{aligned} x(nT) &= \cos((\omega_s/2 \pm \omega_x)nT) \\ &= \cos((\omega_s/2)nT)\cos(\omega_x nT) \mp \sin((\omega_s/2)nT)\sin(\omega_x nT) \\ &= \cos((\omega_s/2)nT)\cos(\omega_x nT) \mp \sin(\pi n)\sin(\omega_x nT), \omega_s T = 2\pi \\ &= \cos((\omega_s/2)nT)\cos(\omega_x nT) \mp 0 \cdot \sin(\omega_x nT) \\ &= \cos((\omega_s/2)nT)\cos(\omega_x nT) \end{aligned}$$

Kosinusi čije su frekvencije  $\omega_1 = \omega_s/2 + \omega_x$  i  $\omega_2 = \omega_s/2 - \omega_x$ , a koje smo uzorkovali s frekvencijom uzorkovanja  $\omega_s$ , imaju identične uzorke.

Još jedan grafički primjer dan je na slici 4.7 gdje se kosinus frekvencije  $f_c = 6\text{Hz}$ , uzorkovan s frekvencijom  $f_s = 5\text{Hz}$ , rekonstruira se kao kosinus frekvencije  $f_r = 1\text{Hz}$ :



Slika 4.7 Alliasing signala frekvencije 6Hz uzrokovanog frekvencijom od 5Hz [9]

Matematički izračun frekvencije rekonstruiranog signala za ovaj primjer je:

$$6 = fs/2 + f_x = 2.5 + f_x \therefore$$

$$f_x = 3.5 \therefore fs/2 - f_x = 2.5 - 3.5 = -1$$

Iz čega se vidi da se ukoliko  $\omega_s$  nije barem dvostruko veći od frekvencije signala dolazi do aliasinga. Kao rješenje tom problemu na izlaz mikrofonskog modula se postavlja anti-aliasing filter. U mom završnom radu nema anti-aliasing filtra odnosno nisko propusnog filtra te dolazi do preklapanja frekvencija prilikom puštanja audio signala frekvencije većeg od 4300Hz.

Pošto je maksimalna frekvencija kojom se može odvijati AD pretvorba 9615Hz, po Nyquistovom teoremu najveća frekvencija koja se može vjerno opisati tom frekvencijom uzrokovanja je otprilike 4800Hz. To je bio ujedno i ograničavajući faktor za ovaj završni rad tako da s Arduino Uno-m nije moguće postići frekvencijski raspon prepoznavanja signala frekvencija od 20 do 20kHz što je bio početni cilj.

## 4.2. Energija signala

Energija signala je mjera snage ili intenziteta signala. Odnosi se na ukupnu količinu energije koju nosi signal. Postoje dva tipa signala: signali s konačnom energijom i signali s konačnom snagom. Signali s konačnom energijom imaju beskonačno mali snagu, dok signali s konačnom snagom imaju beskonačno mali ukupni iznos energije

Za kontinuirane signale, energija se računa kao integral od minus beskonačnosti do plus beskonačnosti od kvadrata apsolutne vrijednosti amplitude signala.

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2 dt \quad (4.4)$$



Za diskretne signale, energija se računa kao suma kvadrata apsolutne vrijednosti svakog uzorka signala.

$$E_x = \sum_{n=0}^{N-1} |x[n]|^2 \quad (4.5)$$

Budući da su vremenska i frekvencijska domena ekvivalentni prikazi istog signala, moraju imati istu energiju. To se zove Parsevalova relacija. Ona vrijedi za kontinuirane i za diskontinuirane signale.

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega \quad (4.6)$$

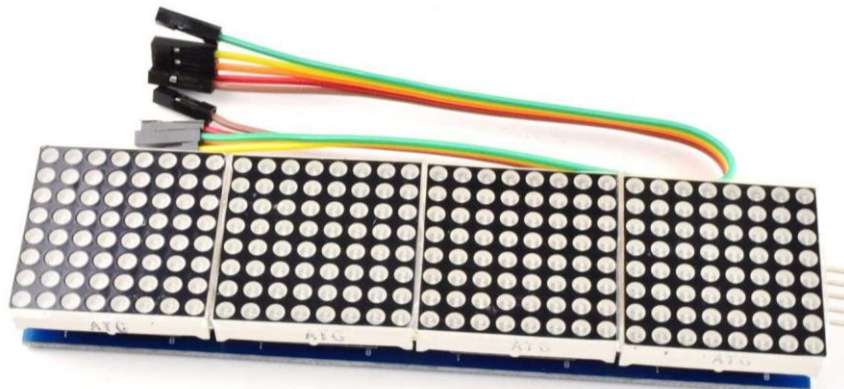
$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2 \quad (4.7)$$

Za određeni frekvencijski pojas energiju možemo izračunati po izrazu:

$$\Delta E_x = \frac{1}{2\pi} |X(\omega)|^2 \Delta\omega \quad (4.8)$$

## 5. MAX7219 MODUL

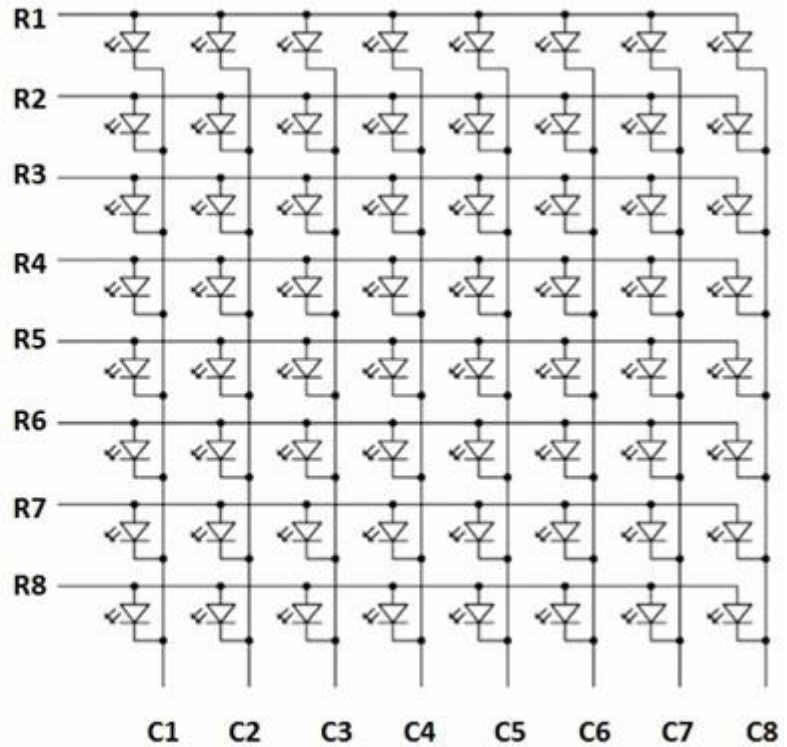
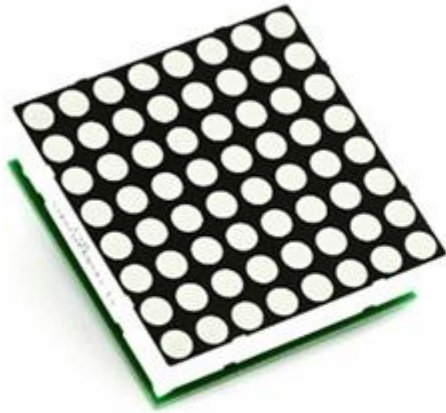
Postoje razne vrste MAX7219 modul pločica, ali njihov osnovni element je 8x8 matrični zaslon i MAX7219 LED upravljač zaslona. U ovom završnom radu korišten je modul FC-16 koji se sastoji od 4 ulančanih 8x8 LED matričnih zaslona koji je prikazan na slici 5.1.



*Slika 5.1. 8x32 LED modul [10]*

### 5.1. Matrični 8x8 zaslon

8x8 točkasti matrični zaslon obično ima 16 pinova, 8 za svaki red i 8 za svaki stupac. Svi redovi i stupci povezani su zajedno kako bi se smanjio broj pinova. Da to nije slučaj, 8x8 matrični zaslon zahtijevao bi 64 pinova, jedan za svaku LED diodu i jedan za konektor zajedničke anode ili zajedničke katode. Povezivanjem redaka i stupaca potrebno je samo 16 pinova za kontrolu cijele matrice što je prikazano na slici 5.2. Ova tehnika upravljanja velikim brojem LED dioda s manje pinova naziva se multipleksiranje.



Slika 5.2. Spoj 64 LED dioda u 8 stupaca i 8 redaka [11]

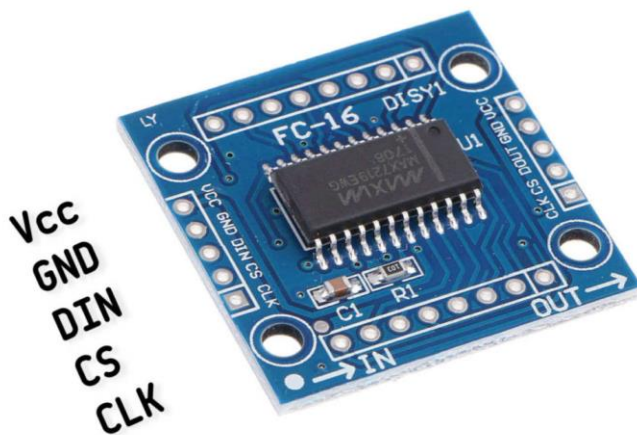
U ovoj tehnici, svaki stupac se aktivira na vrlo kratko vrijeme. Odnosno moramo primijeniti pozitivan napon na odgovarajuće redove te negativan ili uzemljenje na odgovarajući stupac. Kao rezultat toga, u isto vrijeme ne svijetli više od 8 LED odnosno jednog stupca. „Aktivirani“ stupci se izmjenjuju toliko brzo (stotine ili tisuće puta u sekundi) da ljudsko oko percipira zaslon kao potpuno osvijetljen.

Ako redak dobije pozitivan napon, a stupac negativan, tada će samo određena LED svijetliti. Na primjer, ako se svi redovi spoje na pozitivan napon, a red C1 spoji na negativan napon svijetliti će prvi redak. Ako na zaslonu želimo prikazati dijagonalnu crtu od gornjeg lijevog kuta do donjeg desnog kuta moramo prvo spojiti pozitivan napon spojiti na R1 i negativan na C1 pa onda u drugom radnom taktu pozitivan na R2 i negativan na C2 i tako dalje pozitivan na R3 i negativan na C3 u trećem taktu - u svakom taktu mijenjajući upaljeni stupac i njegovu kombinaciju upaljenih svjetlećih dioda.

## 5.2. MAX7219 upravljački sklop

Na pločici pojedinačnog 8x8 MAX7219 modula prikazanog na slici 5.3 nalaze se ženski utori za 16 pinova matričnog zaslona sa 64 svijetleće diode, otpornik koji služi za ograničenje struje, kondenzator koji služi kao filtar te glavni integrirani krug odnosno MAX7219. Također tu se nalaze i pinovi za ulazni napon VCC i uzemljenje GND te pinovi za komunikaciju preko SPI protokola DIN, CLK, CS te DOUT za lančano spajanje više modula zajedno.

VCC se spoja na 5V. Budući da zaslون može povući puno struje (do 1 A pri maksimalnoj svjetlini), najbolje je koristiti vanjsko napajanje umjesto Arduinovog 5V napajanja. Ukoliko se koristi Arduino napajanje od 5 V, treba postaviti svjetlinu na najviše 25% maksimalne vrijednosti kako bi se izbjeglo pregrijavanje regulatora napona. GND je uzemljenje, DIN je pin na koji se dovode ulazni podatci za komunikaciju preko SPI protokola, na pin CS odnosno Chip select se dovodi digitalni ulaz za omogućavanje korištenja modula, na CLK se dovodi radni takt odnosno „clock“ za SPI komunikaciju



Slika 5.3. Upravljački sklop MAX7219 [11]

## 5.3. SPI protokol

SPI protokol (Serial Peripheral Interface) je serijski komunikacijski protokol koji se koristi za prijenos podataka između mikrokontrolera i perifernih uređaja. Ovaj protokol je često korišten u elektronici i mikrokontrolerskim aplikacijama.

Svaki komunikacijski protokol u kojem uređaji dijele signal clock-a poznat je kao sinkroni. SPI je sinkroni komunikacijski protokol. Postoje i asinkrone metode koje ne koriste signal clock-a. Na primjer, u UART komunikaciji, obje strane su postavljene na unaprijed konfiguriranu brzinu prijenosa podataka koja diktira brzinu i vrijeme prijenosa podataka.

SPI protokol koristi 3 ili 4 žice za komunikaciju ovisno o tome dali uređaju imaju izlaz određenih podataka prema mikrokontroleru. To su SCLK (Clock), MOSI (Master Output Slave Input), MISO (Master Input Slave Output) i CS (Slave Select ili Chip Select). Komunikacija se odvija na način da master (glavni) uređaj kontrolira prijenos podataka. On generira serijski sat (SCLK) koji sinkronizira prijenos podataka između uređaja. Master također šalje podatke koristeći MOSI liniju, dok primatelj koristi MISO liniju za slanje podataka natrag masteru.

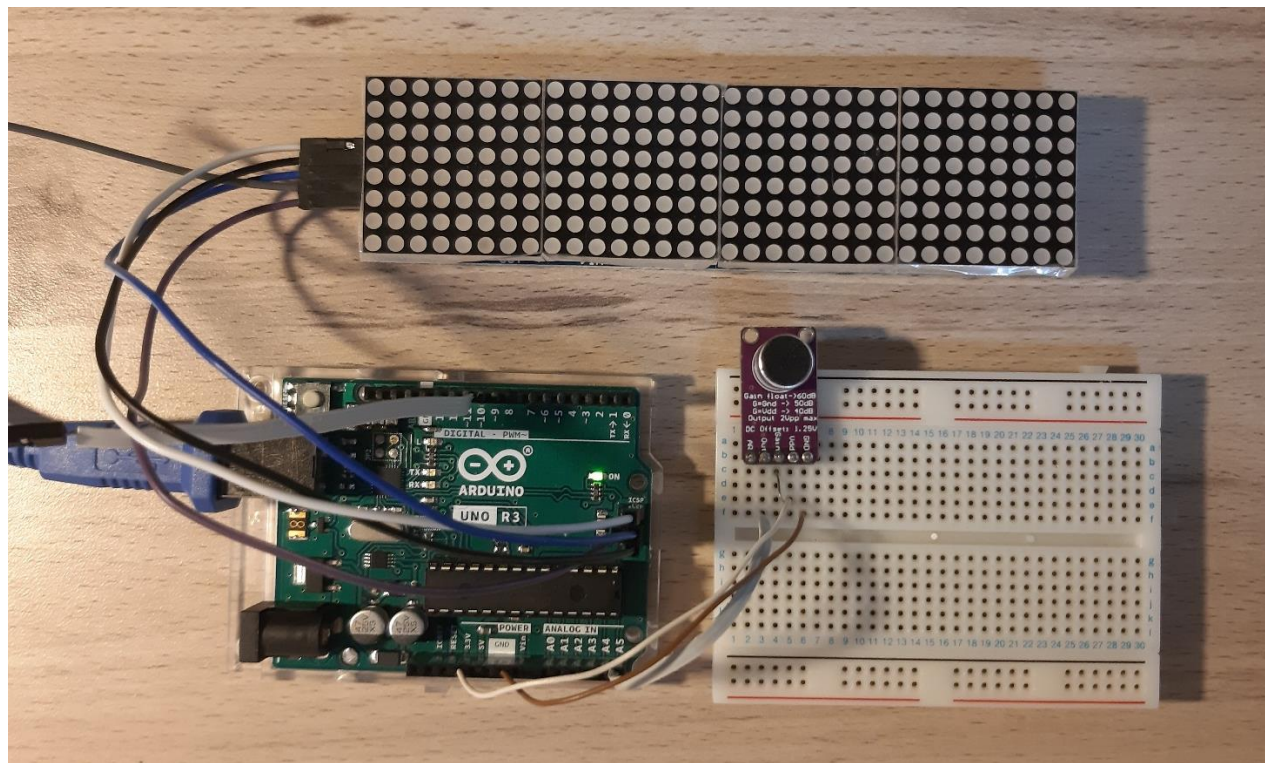
Uz to, svaki periferni uređaj ima svoju CS liniju koja omogućuje masteru da odabere s kojim uređajem želi komunicirati. Ovo omogućuje više uređaja da budu povezani na istu sabirnicu.

SPI podržava puni duplex način rada, gdje glavni uređaj i periferni uređaj istovremeno šalju i primaju podatke. Komunikacija je dvosmjerna i omogućuje brzi prijenos podataka. Također podržava različite načine prenosa podataka kao što su MSB (Most Significant Bit) prvi ili LSB (Least Significant Bit) prvi.

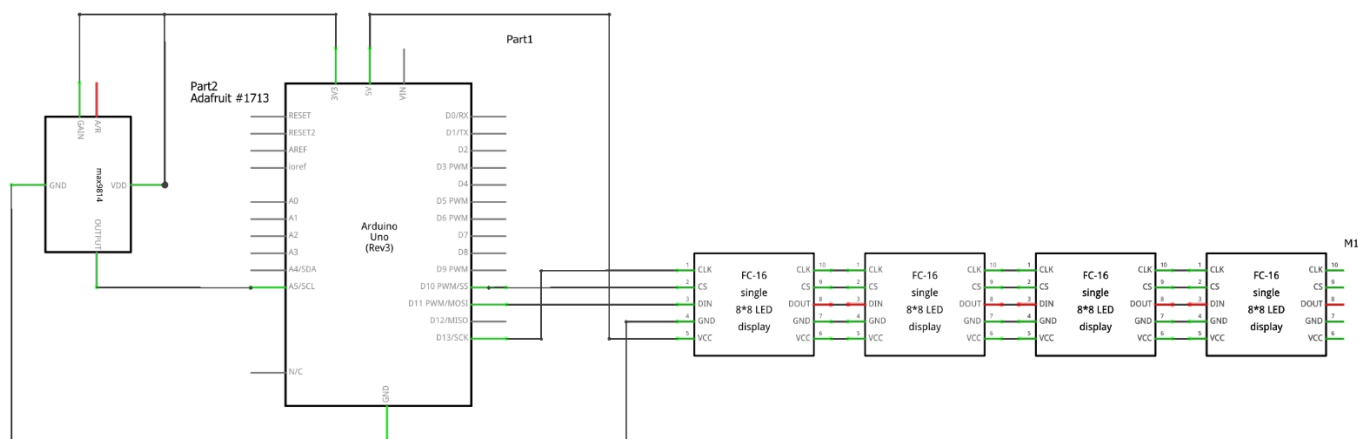
Jedna od prednosti SPI protokola je brzina prijenosa podataka. Uz pravilno konfigurirane uređaje, moguće je postići visoke brzine prijenosa podataka. Međutim, SPI protokol zahtijeva više žica u usporedbi s drugim serijskim protokolima kao što je I2C. Također SPI ne podržava automatsko otkrivanje grešaka ili kontrolu protoka te nema potvrde da su podaci uspješno primljeni (ovo ima I2C protokol).

## 6. SHEMA I PRINCIP RADA

Na slici 6.1 prikazani su dijelovi završnog rada – Arduino Uno, LED matični zaslon MAX7219 te mikrofonsko pojačalo MAX9814. Arduino Uno je napajan i programiran preko USB porta. Na mikrofonski modul je doveden napon od 3.3V pošto mikrofoni na modulu imaju DC prednapon od 1.14 do 1.32V tako da cijeli negativni poluval i cijeli pozitivni poluval ne izgube svoje vrijednosti. Gain na modulu prespojen je na V<sub>dd</sub> tako da se smanji pojačanje na 40dB radi dobivanja što stabilnijeg izlaza iz modula. Izlazni pin modula je spojen na analogni pin A5 na Arduino Uno-u te GND je spojen na pin za uzemljenje na Arduino razvojnoj pločici. LED zaslon je spojen preko I<sup>2</sup>C pinova (MOSI, CLK, VDD, GND) i CS pin je spojen na digitalni I/O pin 10 čime je omogućena komunikacija preko SPI protokola između zaslona i Arduino Uno-a. Intenzitet LED-ica definira se u programskom kodu te ukoliko se želi koristiti napajanje sa Arduino razvojne pločice potrebno je ograničiti intenzitet na najnižu razinu - ali hoće li regulator napona na izdržati ovisi o tome koliko LED-ica odjednom radi i koliko struje vuče zaslon. Zato je bolja opcija spojiti zaslon na svoje zasebno napajanje od 5V kako bi se moglo koristiti zaslon bez pregrijavanja regulatora i *resetabilnog* osigurača. Na slikama 6.2 i 6.3 shematski je prikazan spoj svih navedenih dijelova.

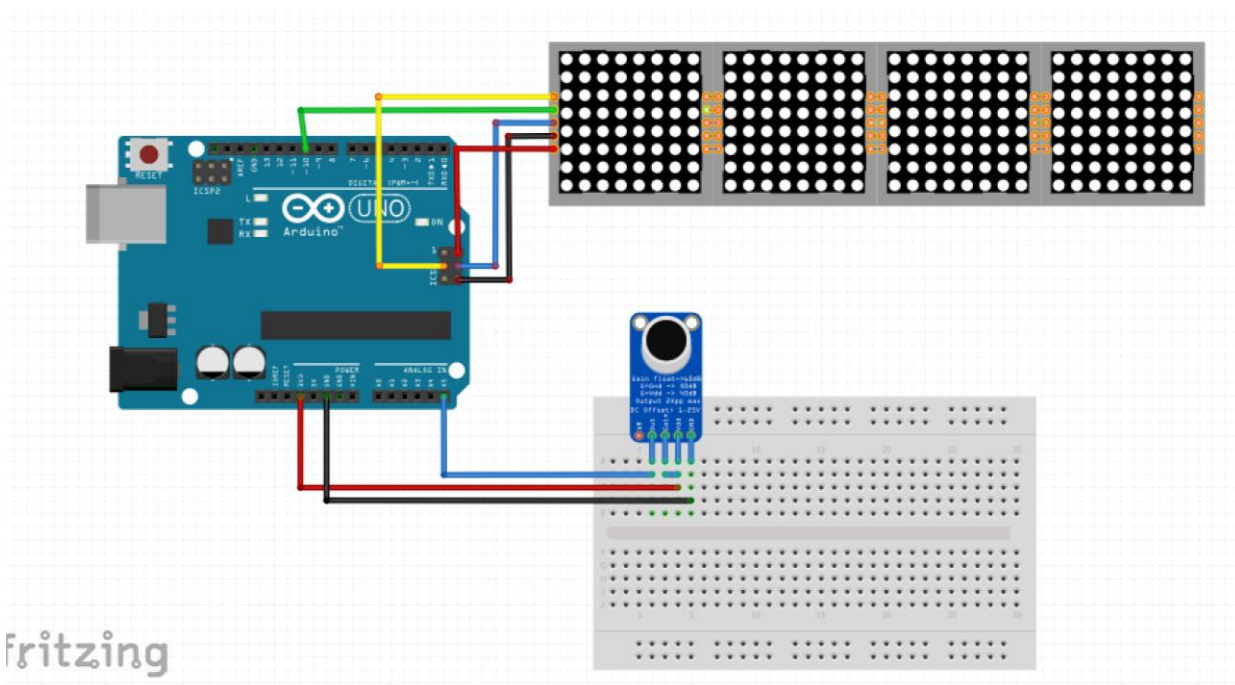


Slika 6.1 Arduino, LED zaslon i mikrofonski modul spojeni



fritzing

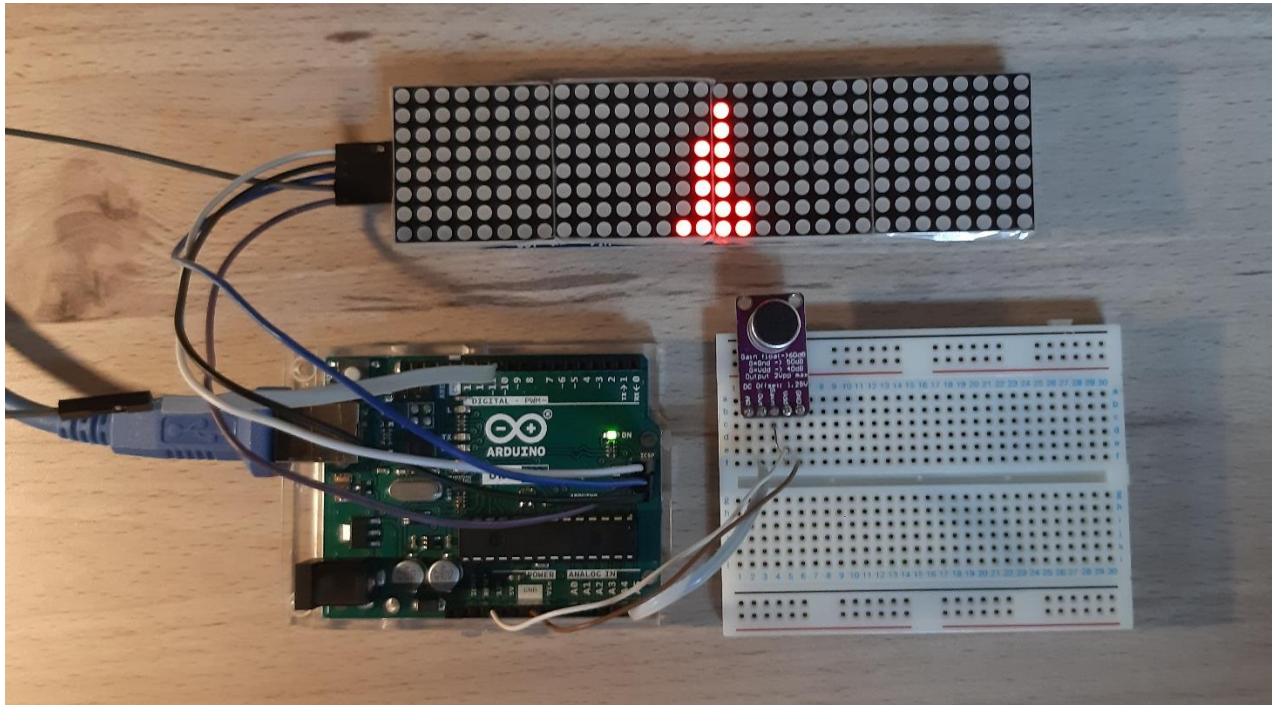
Slika 6.2 Shematski prikaz spoja (Fritzing - Schematic)



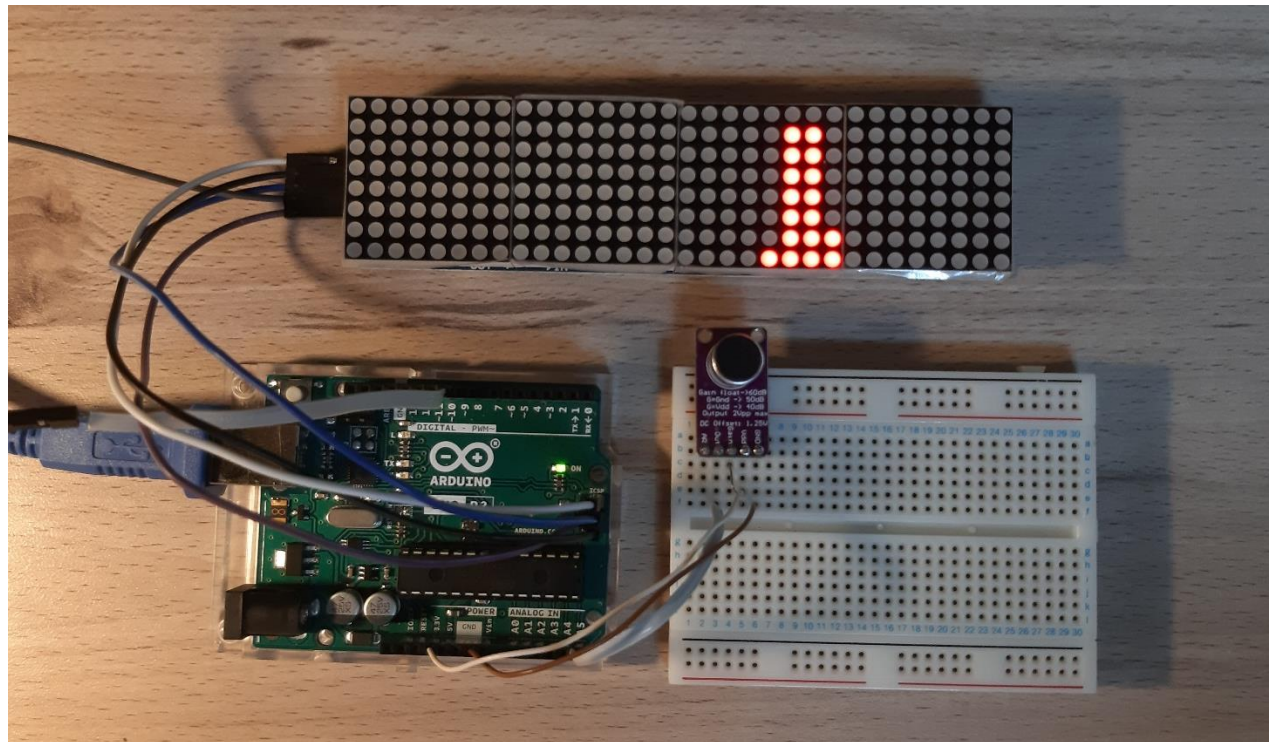
fritzing

Slika 6.3 Grafički prikaz spoja (Fritzing – Breadbord view)

Na slikama 6.4, 6.5 i 6.6 prikazan je zaslon prilikom puštanja zvuka tona frekvencije 2200Hz, 3000Hz te 4300Hz pomoću mobitela i aplikacije za generiranje frekvencija. Pošto je Arduino Uno ograničen sa maksimalnom frekvencijom uzrokovanja na 9615Hz na slikama je prikazano kako je maksimalna frekvencija koja se može detektirati oko 4300Hz što potvrđuje prethodna razmatranja u poglavlju o Fourierovoj transformaciji.

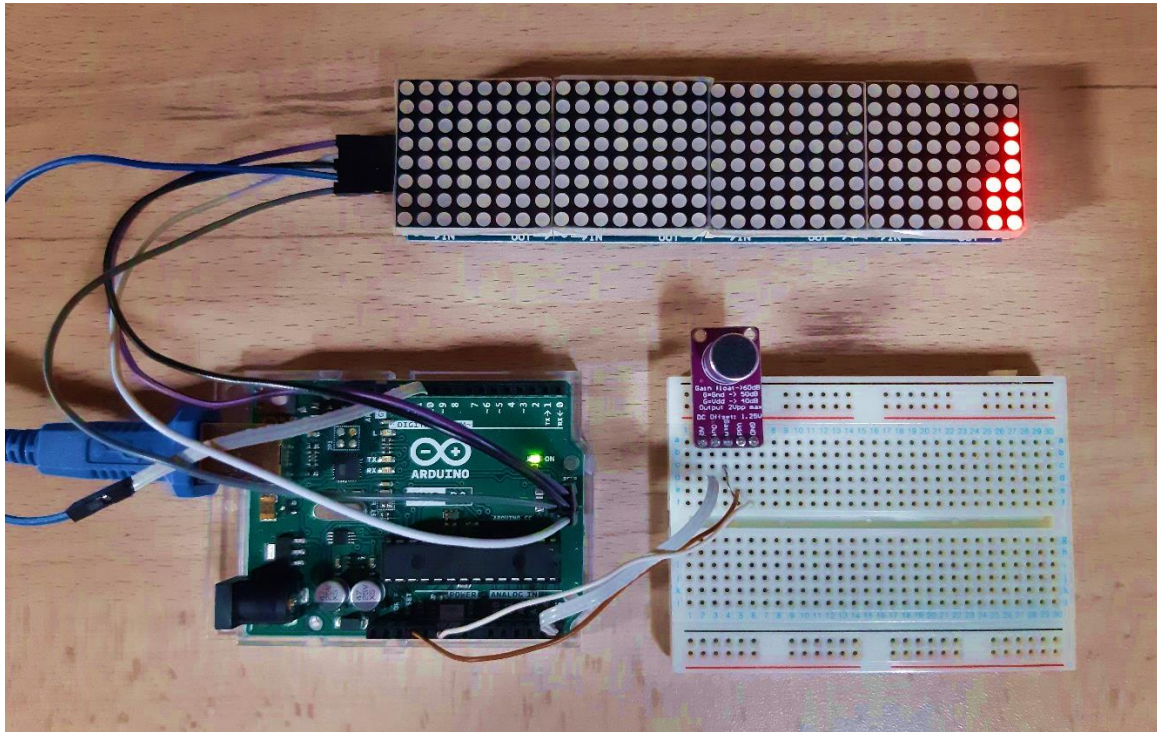


*Slika 6.4 Prikaz rada pri frekvenciji od 2200Hz*



*Slika 6.5 Prikaz rada pri frekvenciji od 3000Hz*





*Slika 6.6 Prikaz rada pri frekvenciji od 4300Hz*

Spoj funkcioniра tako da se audio signal s mikrofonskog modula uzrokuje i sprema u niz diskretnih vrijednosti koje se onda uz pomoć Fourierove transformacije pretvaraju u niz amplituda pojedinih frekvencija iz signala. Nakon toga se te amplitude skaliraju i prilagođavaju prikazu na LED zaslonu gdje svaki stupac predstavlja jedan frekvencijski pojas, dok je amplituda predstavljena visinom stupca.

## 7. PROGRAMSKI KOD

Prvo su definirane knjižnice koje se koriste u projektu: `arduinoFFT` – za izračun DFT-a ulaznog signala, `MD_MAXxx` – za korištenje zaslona, te `SPI` koji služi za korištenje SPI protokola za komunikaciju s MAX7219 integriranim krugom.

```
#include <arduinoFFT.h>
#include <MD_MAX72xx.h>
#include <SPI.h>
```

Potom su definirane varijable: *offset* – označava napon od 1.17V jer Arduino Uno ima 10 bitnu AD pretvorbu tako da je maksimalan digitalni izlaz pretvorbe 1024 odnosno 5V, *samples* – broj uzorka za izračun DFT-a ulaznog signala; *MAX\_VALUE* - varijabla određena koristeći serijsku komunikaciju ispisivanjem vrijednosti koje Fourierova transformacija vraća koja se koristi za skaliranje vrijednosti energija signala za prikaz na zaslonu; *NOISE\_REMOVING\_COEF* - varijabla koja se koristi za uklanjanje malih vrijednosti amplitude određenih frekvencija. Vrsta zaslona je definirana kao `FC16_HW`. Definirani su pinovi na koje su spojeni SPI pinovi MAX7219 modula; *xres* i *yres* – dimenzije LED matrice; *vReal[]* i *vImag[]* – polja u kojima se sprema ulazni signal; konstanta *microphonePin* se postavlja na A5 koja označava na koji je analogni ulaz spojen izlaz iz mikrofonskog modula. *pattern[]* je pomoćna varijabla koja označava razine upaljenih LED dioda po stupcu (*yres*). Ona se definira zato što se nakraju koda kad se poziva funkcija *setColumn()* za prikaz energije određenih frekvencija na LED zaslonu - ona uzima dekadsku vrijednost koju pretvara u binarnu kombinaciju i prikazuje ju u jednom stupcu kao kombinacija upaljenih LED-ica. Odnosno 1 u dekadskom je 1 u binarnom, 3 je u binarnom 11, 7 je u binarnom 111, 15 je u binarnom 1111 i tako dalje svaki slijedeći broj označava jedan broj više upaljenih LED-ica. *FFT* je definiran kao novi objekt u klasi `arduinoFFT()`, te *mx* je definiran kao novi objekt klase `MD_MAX72xx` s konstruktorom koji prima vrstu modula, pin za chip select, te broj 8x8 LED matrica.

```
#define offset 241 // za VDD na mikrofonskom modulu od 3.23V -> (izmjereni DC
prednapon 1.176V)
#define SAMPLES 64
#define MAX_VALUE 2000
#define NOISE_REMOVING_COEF 0.03
#define HARDWARE_TYPE MD_MAX72XX::FC16_HW // vrsta LED zaslona
#define MAX_DEVICES 4 // broj serijskih spojenih modula zaslona
#define CS_PIN 10 // chip select pin
```

```

#define CLK_PIN 13 // clock pin
#define DATA_PIN 11 // podatkovni pin
#define xres 32 // broj stupaca na zaslonu
#define yres 8 // broj redaka na zaslonu

const int microphonePin = A5;

double vReal[SAMPLES];
double vImag[SAMPLES];
double vMagnitude[SAMPLES];

int pattern[]={0, 1, 3, 7, 15, 31, 63, 127, 255}; // niz dekadске vrijednosti broja
upaljenih LED-ica
int newArray[xres]; //pomoćni niz

arduinoFFT FFT = arduinoFFT();
MD_MAX72XX mx = MD_MAX72XX(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);

```

U setup() funkciji uspostavljena je serijska komunikacija s računalom brzine 9600bps, komunikacija s LED zaslonom, te je postavljen intenzitet LED dioda na najnižu razinu.

```

void setup() {
  Serial.begin(9600);
  mx.begin();
  mx.control(MD_MAX72XX::INTENSITY, 1);
}

```

U funkciji loop se unutar prve petlje for petlje izvrši uzrokovanje ulaznog signala s prozorom od 64 uzoraka koji se zapisuje u polje. Od vrijednosti dobivene analogno digitalnom pretvorbom oduzet je *offset* zbog DC prednapona mikrofonskog modula. Korištena funkcija za FFT zahtjeva polje s realnim i imaginarnim vrijednostima signala, a pošto je u našem slučaju riječ o realnom signalu, sve imaginarne komponente u polju vImag su postavljene u 0. Nakon što se signal uzrokuje pozivaju se funkcije koje izračunavaju Fourierova transformaciju i taj se transformirani niz ponovo sprema u polje vReal i vImag. Zatim se izračunava amplituda tih vrijednosti za svaku frekvenciju tako da se računa korijen sume kvadrata realne i imaginarne vrijednosti i upisuje ga se u niz vMagnitude.

```

void loop() {

// Uzrokovanje
  for(int i=0; i<SAMPLES; i++) {
    vReal[i] = analogRead(microphonePin) - offset;
    vImag[i] = 0;
  }
// Uzrokovanje

/*FFT*/
  FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
  FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
/*FFT*/

  for(int i= 0;i<SAMPLES;i++){
    vMagnitude[i] = sqrt(vReal[i]*vReal[i]+vImag[i]*vImag[i]);
  }
}

```

Pošto je spektar parna funkcija koja sadrži i pozitivne i negativne frekvencije, negativne frekvencije prilikom izračuna su odbačene tako što se niz od podijeli sa 2 odnosno gledaju se samo prva polovica niza. U ovom dijelu koda se niz sažima na 32 pojasa (koliko stupaca na LED zaslonu imamo) tako da se računa srednja vrijednost energije više frekvencija zajedno. Taj dio se odvija kroz dvije petlje gdje vanjska for petlja prolazi niz tako u svakoj iteraciji petlje preskače *xres\_ratio* broj polja i onda se u unutarnjoj petlji izračunava srednja vrijednost preskočenih polja i sprema ju se „k“ polje novog *newArray* niza. Ovaj dio koda omogućava da se mijenja broj uzoraka na primjer na 128 pri čemu bi se gledala prva polovica niza od 64 polja te kako bi se dobio željeni niz od 32 polja - za svaka bi se dva polja računala srednja vrijednost i sažimala u jedno polje novog niza.

```

//xres scaling
  int xres_ratio = SAMPLES/xres/2;
  int k=0;
  long int temp=0;
  for(int i=0; i<SAMPLES/2; i=i+xres_ratio, k++){
    for(int j=0; j<xres_ratio; j++){
      temp += vMagnitude[i+j]; // tu se sprema suma preskočenog dijela niza
    }
    newArray[k] = round(temp/xres_ratio);
    temp=0;
  }
//xres scaling

```

Zatim je potrebno skalirati amplitudu pojedinih frekvencijskih pojaseva za prikaz u zaslonu gdje se maksimalna amplituda prikazuje sa 8 upaljenih LED-ica u jednom retku. Glavna for petlja vrti se kroz svih 32 polja novog skaliranog niza *newArray*. S prvim if uvjetom se postiže da ukoliko je amplituda nekog frekvencijskog pojasa dovoljno mala da se može smatrati smetnjom tu amplitudu uopće ne skalira i u else uvjetu tom se pojasu opredjeljuje amplituda 0. *yres\_ratio* varijabla odnosno maksimalna vrijednost amplitude podjeljenja sa brojem redaka LED zaslona - označava vrijednost razine amplitude koja predstavlja uključivanje jedne LED-ice jednog retka na zaslonu. U unutarnjoj for petlji se uspoređuje amplituda određenog frekvencijskog pojasa odnosno polja niza *newArray* sa određenom vrijednosti amplitude. Na način da ako je vrijednost polja veća od prve razine amplitude i manja od druge razine amplitude energije frekvencijskog pojasa u nju se zapisuje broj iz niza *pattern* koji u binarnom sustavu predstavlja uključivanja određenog broja LED-ica. Ukoliko je vrijednost polja veća od definirane varijable *MAX\_VALUE* njoj se opredjeljuje vrijednost 255 odnosno 11111111 u binarnom što predstavlja maksimalnu vrijednost amplitude.

```
//yres scaling
double yres_ratio = MAX_VALUE/yres; //
for(int i=0; i<SAMPLES/2;i++){
    if(newArray[i] > MAX_VALUE*NOISE_REMOVING_COEF){
        for(int j=1; j<yres+1; j++){
            if((newArray[i] > yres_ratio*(j-1)) && (newArray[i] <= yres_ratio*(j))){
                newArray[i] = pattern[j];
            }
            else if(newArray[i]>MAX_VALUE){
                newArray[i] = pattern[8];
                //ako je vrijednost veća od MAX_VALUE upalite sve LED-ice tog reda
            }
        }
    }else {newArray[i] = 0;}
}
//yres scaling
```

Na posljetku se u for petlji pozivom na objekt *mx* funkcijom *setColomn* zadaje broj LED-ica koje se uključe u pojedinom retku. Funkcija *setColomn* prima dva argumenta od kojih je prvi redak, a drugi je vrijednost koja u binarnom predstavlja uključene LED-ice u tom određenom retku. Na primjer naredba *setColumn(2, 170)*; bi značila da se u drugom retku uključi kombinacija 10101010 (to je 170 u binarnom) odnosno da svaka druga LED-ica svijetli u tom retku.

```
for(int i=0;i<xres; i++){
    mx.setColumn(i, newArray[i]);
}
```

## 8. ZAKLJUČAK

U sklopu ovog završnog rada izrađen je frekvencijski analizator signala temeljen na Arduino platformi. U prvom koraku se audio signal koji dolazi od mikrofonskog modula AD pretvorbom pretvara u digitalni te se uzrokuje i zapisuje kao niz brojeva. Zatim se taj niz koji predstavlja vrijednosti amplitude signala pretvara preko Fourierove transformacije u frekvencijski spektar te se naposljetku na LED zaslonu prikazuje prisutnost i amplituda određenih frekvencija u signalu.

Budući da Arduino Uno ima svoje granice ovaj završni rad ima nedostatak što s ovakvom opremom maksimalna frekvencija uzorkovanja je 9615Hz pa je maksimalna frekvencija koja se može točno uzrokovati u praksi otprilike 4300Hz. Ovaj problem se može riješiti korištenjem mikrokontrolera s bržim AD pretvornikom ili korištenje vanjskog bržeg AD pretvornika (npr. ADS1115 modul).

Drugo unaprijeđenje sklopa se može ostvariti u vidu anti-aliasing filtera između mikrofona i AD pretvornika, tj. nisko-propusnog filtera s graničnom frekvencijom duplo manjom od frekvencije uzorkovanja. Filter bi se trebao ugraditi zbog toga što za frekvencije iznad 4300Hz dolazi do aliasinga, tj. preslikavanja frekvencija viših od 4300Hz na frekvencije manje od 4300Hz, npr. frekvencija od 4400Hz se preslikava na 4200Hz. Takvim filterom bi se spriječilo prethodno opisano preslikavanje frekvencija, tako da se frekvencije iznad 4300Hz uopće ne bi uključile u AD pretvorbu.

## 9. LITERATURA

- [1] Jobit, Joseph, „Everything you need to know about the Arduino Hardware“, s interneta, <https://circuitdigest.com/article/everything-you-need-to-know-about-arduino-uno-board-hardware>, 21.3.2022
- [2] Autor nepoznat, „Arduino UNO Reference design“, s interneta, <https://docs.arduino.cc/static/c1593a4c4960ff7b51d1083cb8e45812/schematics.pdf>, 06.03.2019.
- [3] Debashis, Das, „How does Successive Approximation (SAR) ADC Work and Where is it best used?“, s interneta, <https://circuitdigest.com/article/how-does-successive-approximation-sar-adc-work-and-where-is-it-best-used>, 30.10.2020.
- [4] Autor nepoznat, „AZDelivery Max9814 Microphone AGC Amplifier“, s interneta, <https://www.amazon.co.uk/AZDelivery-Max9814-Microphone-Parent/dp/B086W7S86W?th=1>, 08.04.2020.
- [5] Analog devices, „Microphone Amplifier with AGC and Low-Noise Microphone Bias“, s interneta, <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX9814.pdf>, 02.2020.
- [6] Sanderson, Grant (Youtube kanal: 3Blue1Brown), „But what is the Fourier Transform? A visual introduction.“, s interneta, <https://www.youtube.com/watch?v=spUNpyF58BY>, 26.01.2018.
- [7] Autor nepoznat, „Eulerova formula“, s interneta, [https://hr.wikipedia.org/wiki/Eulerova\\_formula](https://hr.wikipedia.org/wiki/Eulerova_formula), 21.6.2023.
- [8] P. Prandoni, M. Vetterli, „Signal Processing for Communications“, EPFL Press, Italija
- [9] Viktor Sučić, „PRIMJENE FOURIEROVIH METODA U ANALIZI SIGNALA I SUSTAVA“, predavanje iz predmeta „Signali i Sustavi“
- [10] Autor nepoznat, „8x32 LED Display For Pi Arduino Various“, s interneta <https://www.ebay.com/itm/194944475382>, 9.10.2023
- [11] Autor nepoznat, „MAX7219 LED Dot Matrix Display Interfacing with Arduino“, s interneta, <https://microcontrollerslab.com/max7219-dot-matrix-display-arduino-tutorial/>, 1.9.2023

## **10. SAŽETAK I KLJUČNE RIJEČI**

### **10.1. Sažetak**

U ovom završnom radu pomoću Arduino platforme, LED zaslona i mikrofonskog pojačala izrađen je frekvencijski analizator signala. Ulazni analogni audio signal koji dolazi sa mikrofonskog pojačala Arduino uzrokuje i pretvara u digitalni te sprema kao diskretan niz vrijednosti tog signala. Zatim se taj signal pretvara korištenjem FFT biblioteke u frekvencijski spektar. Takav transformirani signal se skalira i prilagođuje kako bi se na LED zaslonu prikazala energija pojedinih frekvencijskih pojaseva u signalu.

### **10.2. Ključne riječi**

Arudino, Mikrofonsko pojačalo, Fourierova transformacija, FFT, LED matrični zaslon



## **11. SUMMARY AND KEYWORDS**

### **11.1. Summary**

In this bachelor thesis, a frequency signal analyzer has been designed using the Arduino platform, an LED display and a microphone amplifier. The input analog audio signal coming from the microphone amplifier is sampled by the Arduino and converted to a digital value and stored as a discrete array. Then the signal is converted using the FFT library into its frequency spectrum. This transformed signal is then scaled and adjusted in order to be displayed on the LED screen as energies of the individual signal frequency bands.

### **11.2. Keywords**

Arduino, Microphone amplifier, Fourier transform, FFT, LED matrix display