

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1

**Bežični prijenos audio signala
putem BLE sučelja razvojnog
sustava STM32WB5MM-DK**

Jelena Gavran

Zagreb, lipanj 2022.

SADRŽAJ

Indeks slika	v
Indeks isječaka koda	v
1. Uvod	1
2. STM32WB5MM-DK razvojni sustav	2
2.1. BLE protokol	3
2.1.1. Upravljač	4
2.1.2. Domaćin	5
2.2. MEMS mikrofoni	7
3. Povezivanje razvojnog sustava i računala	8
3.1. FP-AUD-BVLINKWB1 Function Pack	8
3.2. BlueST SDK	8
4. Programska potpora za korisničko sučelje	9
4.1. Razvojni alat PyQt	9
4.2. Implementacija korisničkog sučelja	10
5. Obrada audio signala	12
6. Zaključak	14
Literatura	15

INDEKS SLIKA

1.1. Blok shema sustava	1
2.1. Konfiguracija STM32WB5MM-DK razvojnog sustava	2
2.2. BLE arhitektura stoga	3
2.3. Automat sloja veze	5
4.1. Uvodni izbornik	10
4.2. Sučelje za podešavanje parametara	11
4.3. Prije pokretanja snimanja	11
4.4. Skeniranje uređaja	11
4.5. Snimanje zvuka	11
4.6. Završetak snimanja	11
5.1. Primjer prikaza analize audiozapisa	13

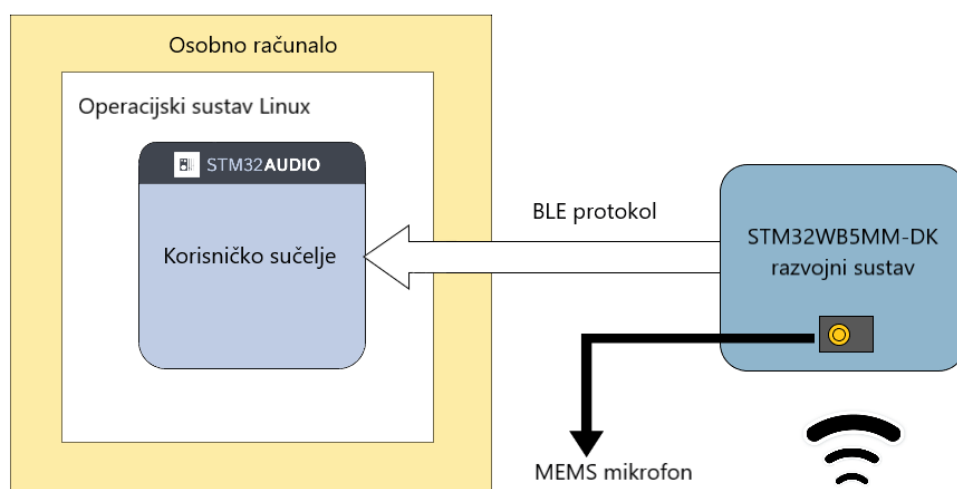
INDEKS ISJEČAKA KODA

5.1. Fourierova transformacija	12
--	----

1. Uvod

Ovaj završni rad dio je doktorskog rada ... koji se bavi

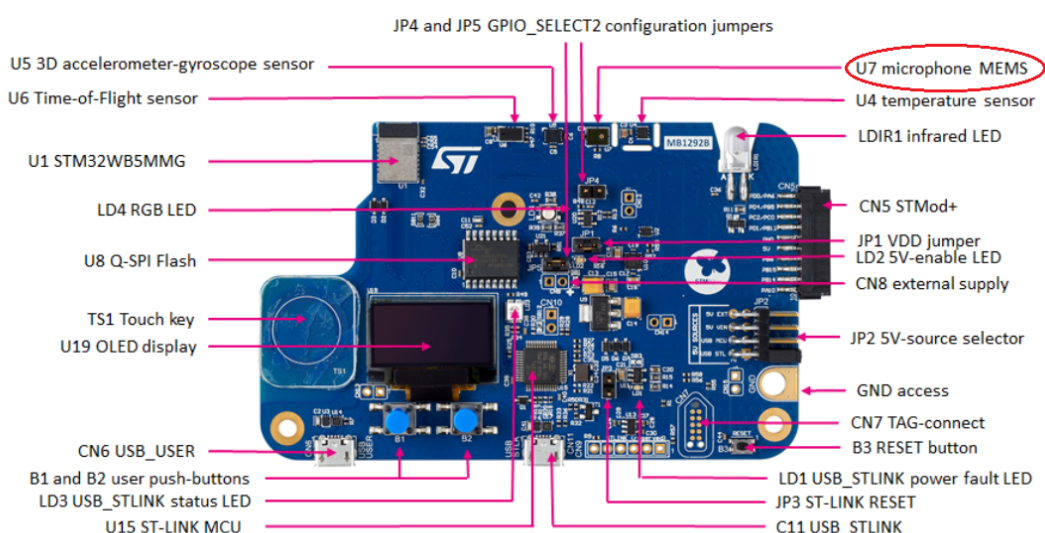
U okviru ovog završnog rada razvijena je programska potpora za mikrokontroler STM32WB5M te je uspostavljeno BLE komunikacijsko sučelje između razvojnog sustava i osobnog računala s operacijskim sustavom Linux. Sučeljem se prenosi zvučni signal sniman MEMS mikrofonom s mikrokontrolera na računalo. Također je i razvijeno korisničko sučelje za pokretanje komunikacije, prijem i pohranu signala. Blok shema sustava prikazana je na Slici 1.1.



Slika 1.1: Blok shema sustava

2. STM32WB5MM-DK razvojni sustav

Razvojni sustav temelji se na STM32WB5MMG modulu tvrtke *STMicroelectronics*, koji je dio linije STM32WBx5 razvojnih sustava. Kao i svi mikrokontroleri iz te skupine, modul sadrži 32-bitni Arm Cortex-M4, aplikacijski procesor koji radi na frekvenciji do 64 MHz, te Cortex-M0+, mrežni procesor s frekvencijom rada do 32 MHz. Modul sadrži 1 MB *Flash* memorije i 256 KB SRAM-a. Budući da je modul RF primopredajnik, podržava protokole Bluetooth, Zigbee, Thread i konkurentne bežične standarde. Sustav također ima 0.96-inčni 128x64 zaslon, RGB LED lampice te senzore za temperaturu, dodir, I2C, *Time-of-Flight* senzor i žiroskop. Od ostale periferije najznačajniji je digitalni MEMS mikrofون. STM32 modul je višeprotokolni, bežični uređaj niske potrošnje energije (engl. *ultra-low-power*) primarno namijenjen razvoju aplikacija koje koriste audio, USB ili *Bluetooth Low Energy* (BLE) protokol.



Slika 2.1: Konfiguracija STM32WB5MM-DK razvojnog sustava

2.1. BLE protokol

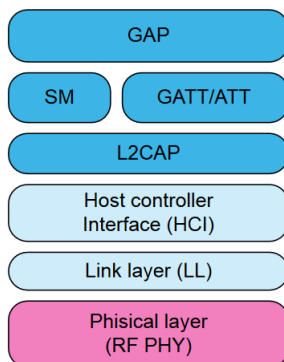
Bluetooth protokol korišten je za povezivanje razvojnog sustava s matičnim računalom i za prijenos audio signala s mikrokontrolera. BLE je vrsta bežične komunikacije namijenjena komunikaciji kratkog dometa s niskom potrošnjom energije. Razvijen je kako bi se postigao standard vrlo male snage koji radi s baterijom veličine kovanice (engl. *coin-cell batteries*) nekoliko godina. Klasična Bluetooth tehnologija razvijena je kao bežični standard, što je omogućilo razvoj bežičnih i prenosivih uređaja, no ne podržava dug život baterije zbog brze i nepredvidive komunikacije te složenih postupaka povezivanja. BLE uređaji troše samo dio energije koju troše standardni Bluetooth proizvodi te omogućavaju malenim uređajima s malim baterijama bežično povezivanje s uređajima koji koriste klasični Bluetooth.

BLE radi u istom opsegu od 2,4 GHz kao i standardni Bluetooth, no koristi različite kanale od standardnog Bluetootha. Koristi 40 kanala od 2 MHz za prijenos podataka korištenjem modulacije Gaussova pomaka frekvencije (metoda koja se koristi za glatke prijelaze između podatkovnih impulsa), zbog čega skokovi frekvencije proizvode manje smetnji u usporedbi sa standardnom Bluetooth komunikacijom.

Arhitektura BLE tehnologije naziva se još i BLE stog zbog slojevite strukture. Stog se sastoji od dvije glavne komponente:

- Upravljač (engl. *Controller*)
- Domaćin (engl. *Host*)

Upravljač se sastoji od fizičkog sloja i sloja veze. Host uključuje protokol kontrole i prilagodbe logičke veze (L2CAP), upravitelja sigurnosti (SM), protokol atributa (ATT), generički profil atributa (GATT) i generički profil pristupa (GAP). Sučelje između komponenti naziva se sučelje host kontrolera (HCI).



Slika 2.2: BLE arhitektura stoga

2.1.1. Upravljač

Fizički sloj

Fizički sloj je radio brzine od 1 Mbps koji prenosi informacije GFSK (*Gaussian Frequency Shift Keying*) frekvencijskom modulacijom. Radi u 2,4 GHz ISM pojasu bez licence na 2400-2483,5 MHz. BLE sustav koristi 40 RF kanala (0-39), s razmakom od 2 MHz. Postoje dvije vrste kanala:

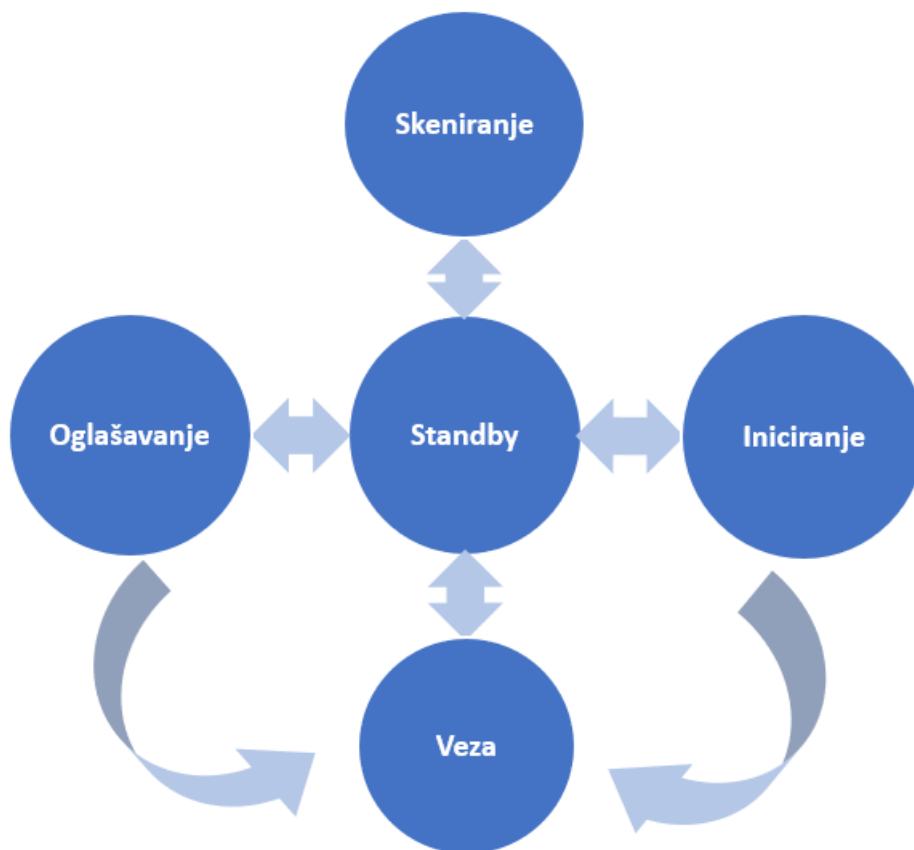
1. Kanali za oglašavanje koji koriste tri fiksna RF kanala (37, 38 i 39) za:
 - (a) Pakete kanala za oglašavanje
 - (b) Pakete korištene za otkrivanje ili povezivanje
 - (c) Pakete korištene za odašiljanje ili skeniranje
2. Podatkovni fizički kanal, koristi ostalih 37 RF kanala za dvosmjernu komunikaciju između povezanih uređaja.

BLE je tehnologija adaptivnog skakanja frekvencije (AFH) koja može koristiti samo podskup svih dostupnih frekvencija kako bi se izbjegle sve frekvencije koje koriste druge neprilagodljive tehnologije. To omogućuje prelazak s lošeg kanala na poznati dobar kanal korištenjem specifičnog algoritma za skakanje frekvencije, koji određuje sljedeći dobar kanal za korištenje.

Sloj veze

Sloj veze određuje kako dva uređaja mogu koristiti radio za međusoban prijenos informacija. Također definira automat s pet stanja:

- Stanje pripravnosti (engl. *Standby*): uređaj ne šalje niti prima pakete
- Oglašavanje: uređaj šalje oglase putem kanala za oglašavanje
- Skeniranje: uređaj traži uređaje oglašivača
- Pokretanje (iniciranje): uređaj pokreće vezu s uređajem oglašivača
- Veza: uređaj inicijatora u glavnoj je ulozi (engl. *master*) - komunicira s uređajem u podređenoj (engl. *slave*) ulozi i definira vrijeme prijenosa
- Uređaj oglašivača je u ulozi podređenog - komunicira s jednim uređajem u glavnoj ulozi



Slika 2.3: Automat sloja veze

HCI

Sloj sučelja glavnog kontrolera (HCI) pruža sredstvo komunikacije između hosta i upravljača putem softverskog API-ja ili hardverskog sučelja kao što su: SPI, UART ili USB. Dolazi iz standardnih Bluetooth specifikacija, s novim dodatnim naredbama za funkcije specifične uz nisku potrošnju energije.

2.1.2. Domaćin

L2CAP

Protokol logičke veze i sloja prilagodbe (L2CAP) podržava multipleksiranje protokola više razine, operacije fragmentacije paketa i ponovnog sastavljanja, te prijenos informacija o kvaliteti usluga.

ATT

Protokol atributa (ATT) omogućuje uređaju da prikazuje podatke, koji se nazivaju atributima, drugom uređaju. Uređaj koji prikazuje attribute naziva se poslužiteljem, a uređaj koji ih koristi naziva se klijentom. ATT definira skup metoda za otkrivanje, čitanje i pisanje atributa na drugi uređaj. Implementira *peer-to-peer* protokol između poslužitelja i klijenta tipičnom zahtjev-odgovor strukturom. Poslužitelj i klijent imaju sljedeće uloge:

- Poslužitelj
 - Sadrži sve attribute (baza atributa)
 - Prima zahtjeve, šalje odgovore, izvršava naredbe
 - Obavijesti o promjeni vrijednosti atributa
- Klijent
 - Komunicira s poslužiteljem
 - Šalje zahtjeve, čeka na odgovor
 - Može čitati i mijenjati podatke uz dopuštenje poslužitelja

SM

BLE sloj veze podržava enkripciju i autentifikaciju korištenjem načina brojača s CBC-MAC algoritmom (kod za provjeru autentičnosti lančanih poruka) i 128-bitnu AES blok šifru (AES-CCM). Kada se enkripcija i autentifikacija koriste u vezi, 4-bajtna provjera integriteta poruke (MIC) dodaje se korisnom učitavanju podatkovnog kanala PDU. Enkripcija se primjenjuje i na polja od PDU i MIC. Kada dva uređaja žele šifrirati komunikacije tijekom veze, upravitelj sigurnosti (SM) koristi postupak uparivanja. Ovaj postupak omogućuje provjeru autentičnosti dvaju uređaja razmjennom informacija o njihovu identitetu kako bi se stvorili sigurnosni ključevi koji se mogu koristiti kao osnova za pouzdani odnos ili (jednu) sigurnu vezu.

GATT

Generički atributni profil (GATT) definira okvir za korištenje ATT protokola, a koristi se za usluge, otkrivanje deskriptora, čitanje, pisanje i obavijesti. U GATT kontekstu, kada su dva uređaja povezana, postoje dvije uloge uređaja:

- GATT klijent: uređaj pristupa podacima na udaljenom GATT poslužitelju putem čitanja, pisanja, obavještavanja

- GATT poslužitelj: uređaj pohranjuje podatke lokalno i pruža metode pristupa podacima udaljenom GATT klijentu

Uređaj istovremeno može biti i GATT poslužitelj i GATT klijent.

GAP

Bluetooth sustav definira osnovni profil koji implementiraju svi Bluetooth uređaji koji se naziva generički profil pristupa (GAP), koji definira osnovne zahtjeve Bluetooth uređaja. Postoje četiri uloge GAP profila:

- Emiter - šalje oglase
- Promatrač - prima oglase
- Periferija - uvijek u načinu oglašavanja i u *slave* ulozi
- Centar - nikada ne šalje oglase, uvijek u *master* ulozi

U kontekstu GAP-a definirana su dva temeljna koncepta:

- GAP načini rada (engl. *modes*): konfigurira uređaj da djeluje na određeni način duži vremenski period. Postoje četiri tipa GAP načina rada: emitiranje, otkrivanje, povezivanje i vezanje
- GAP procedure: konfigurira uređaj da izvrši jednu radnju u ograničenom vremenskom periodu. Postoje četiri tipa GAP postupaka: promatrač, otkrivanje, povezivanje, postupci povezivanja

Istovremeno se mogu koristiti različiti načini otkrivanja i povezivanja.

2.2. MEMS mikrofoni

3. Povezivanje razvojnog sustava i računala

3.1. FP-AUD-BVLINKWB1 Function Pack

3.2. BlueST SDK

4. Programska potpora za korisničko sučelje

Za bolje korisničko iskustvo kreirano je grafičko korisničko sučelje (engl. *Graphic User Interface* - GUI) koje se izvodi na korisničkom računalu. U GUI aplikaciji moguće je pokrenuti snimanje novog audiozapisa te grafički prikazati obradu signala postojećeg zvuka na računalu.

4.1. Razvojni alat PyQt

Aplikacija je izrađena korištenjem razvojnog alata PyQt temeljenog na programskom jeziku Python i pripadnih biblioteka za razvoj grafičkih korisničkih sučelja. PyQt je priključak (engl. *plug-in*) za Python - mostna biblioteka između Pythona i razvojnog alata Qt, koji podržava programski jezik C++. Korištena je inačica *PyQt5*, koja je kompatibilna s Python 3 verzijom.

Osnova *Qt* aplikacija je objektni model koji, koristeći sustav *Meta Object* i klasu *QObject*, proširuje funkcionalnost standardnog programskog jezika C++ i time omogućuje razvoj grafičkih korisničkih sučelja. *PyQt* omata funkcionalnosti *Qt* radnog okvira te ih prilagođava programskom jeziku Python, odnosno kombinira kompleksnost alata za razvoj grafičkog sučelja i jednostavnost programskog jezika.

Osnovna klasa je *QObject* koja pruža sljedeće funkcionalnosti:

- definiranje objekata jedinstvenim imenom,
- hijerarhijska organizacija objekata,
- komunikacija između objekata,
- upravljanje događajima.

Komunikacija između *Qt* objekata odvija se mehanizmom signala i utora (engl. *signals and slots*). Signal se emitira pri promjeni stanja objekta, primjerice pritiskom

na gumb unutar korisničkog sučelja. Pri emisiji signala poziva se funkcija utora s kojom je taj signal povezan te se obrađuje događaj koji je izazvao emisiju.

Stvaranje i uređivanje grafičkih elemenata (engl. *widgets*) omogućeno je klasom *QWidget*. Grafički elementi organizirani su hijerarhijski, pri čemu je glavni prozor "roditelj" ostalih elemenata.

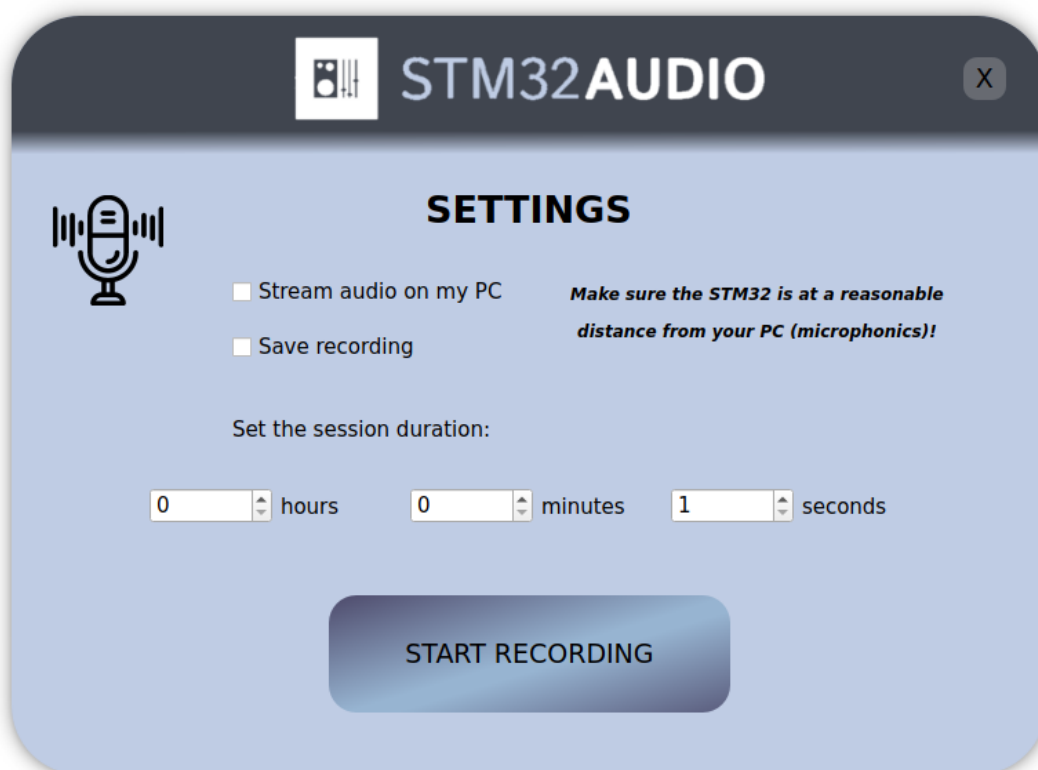
4.2. Implementacija korisničkog sučelja

Pri pokretanju aplikacije, u glavnom prozoru prikazuje se izbornik s gumbima *Record* i *Analyse Audio*. *Record* gumb vodi na sučelje za podešavanje parametara snimanja, dok *Analyse Audio* otvara meni za odabir audio datoteke nad kojom će se provesti obrada signala.



Slika 4.1: Uvodni izbornik

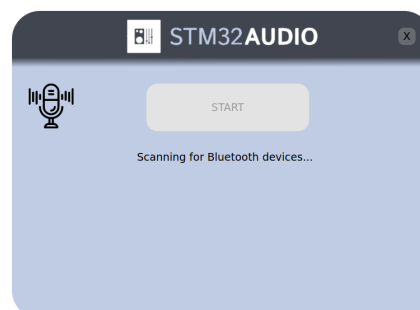
Ovdje će biti tekst koji govori o odabiru parametara.



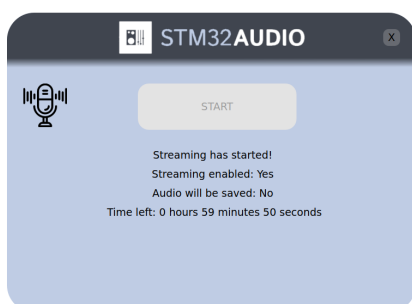
Slika 4.2: Sučelje za podešavanje parametara



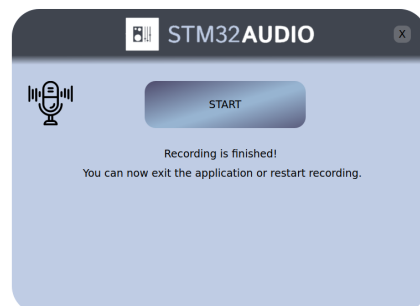
Slika 4.3: Prije pokretanja snimanja



Slika 4.4: Skeniranje uređaja



Slika 4.5: Snimanje zvuka



Slika 4.6: Završetak snimanja

5. Obrada audio signala

Nakon otvaranja audio datoteke, vrši se obrada zvučnog signala. Audio biblioteka *SoundFile* koristi se za učitavanje audio datoteke te, pozivom

```
sf.read(file_path)
```

dobivaju se matrica amplituda i frekvencija uzorkovanja. Dobivena matrica reprezentacija je audio signala u vremenskoj domeni, odnosno prikazuje glasnoću (amplitudu) zvuka dok se mijenja u vremenu. Amplituda jednaka nuli označava tišinu.

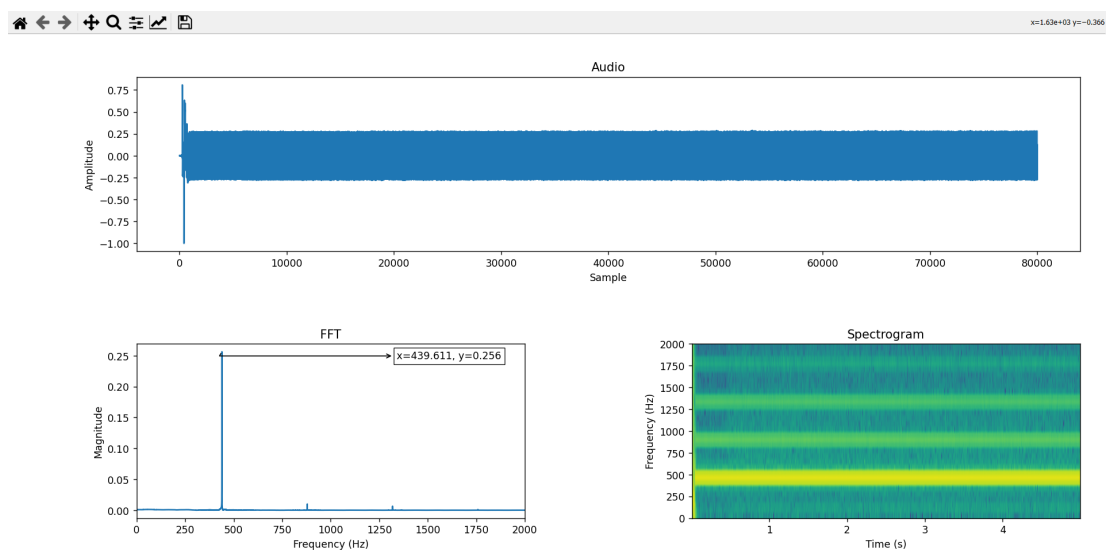
Za analizu odnosa amplitude i frekvencije signala potrebno transformirati signal u frekvencijsku domenu kako bi se prikazalo koje frekvencije se nalaze u signalu. Fourierovom transformacijom signal se dekomponira u odgovarajuće frekvencije. *Scipy* biblioteka sadrži ugrađenu funkciju za brzu Fourierovu transformaciju.

```
n = len(samples)
T = 1/sampling_rate

b, a = iirnotch(50, 50/len(samples), sampling_rate)
```

Isječak koda 5.1: Fourierova transformacija

Dobivene matrice iscrtavaju se grafički pomoću biblioteke *matplotlib*. Ograničen je prikaz frekvencija na 2000 Hz...



Slika 5.1: Primjer prikaza analize audiozapisa

6. Zaključak

LITERATURA

- [1] Kartik Chaudhary. Understanding audio data, fourier transform, fft and spectrogram features for a speech recognition system. URL <https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-features-for-a-speech-recognition-system-1e1e1e1e1e1e>

**Bežični prijenos audio signala putem BLE sučelja razvojnog sustava
STM32WB5MM-DK**

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: STM32WB5MM-DK, BLE, MEMS mikrofoni, korisničko sučelje, obrada signala

**Audio Signal Transmission Using BLE Interface of STM32WB5MM-DK
Development Kit**

Abstract

English abstract.

Keywords: STM32WB5MM-DK, BLE, MEMS microphone, user interface, signal processing