

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
1. Uvod	1
2. STM32WB5MM-DK razvojni sustav	2
3. Programska potpora za korisničko sučelje	3
3.1. Razvojni alat PyQt	3
3.2. Implementacija korisničkog sučelja	4
4. Zaključak	7
Literatura	8

INDEKS SLIKA

1.1. Blok shema sustava	1
3.1. Uvodni izbornik	4
3.2. Primjer prikaza analize audiozapisa	5
3.3. Sučelje za podešavanje parametara	6
3.4. Prije pokretanja snimanja	6
3.5. Skeniranje uređaja	6
3.6. Snimanje zvuka	6
3.7. Završetak snimanja	6

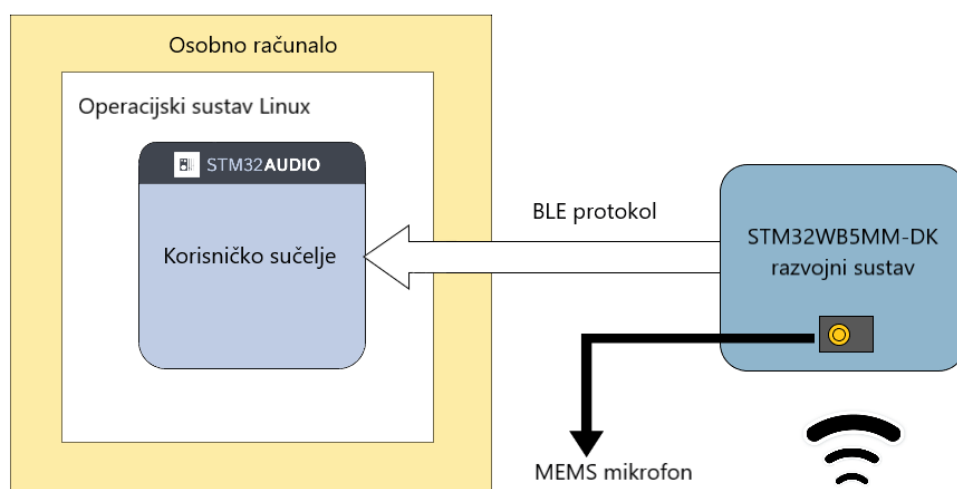
INDEKS ISJEČAKA KODA

3.1. Fourierova transformacija	5
--	---

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 slikom.



Slika 1.1: Blok shema sustava

2. STM32WB5MM-DK razvojni sustav

3. 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.

3.1. 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.

3.2. 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 za analizu.



Slika 3.1: Uvodni izbornik

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

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

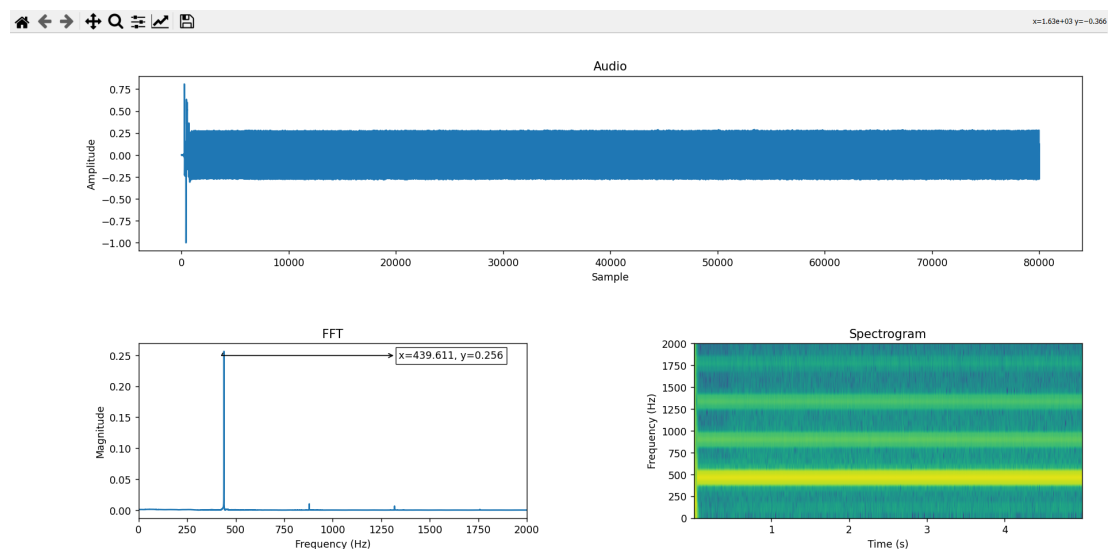
dobivaju se matrica amplituda i frekvencija uzorkovanja. Dobivena matrica reprezentacija je audio signala u vremenskoj domeni, odnosno prikazuje glasnoću (amplitudu) zvuka dok se mijenja u vremenu. Amplituda jednaka nuli označava tišinu.

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
yf = fft(samples)
xf = np.linspace(start=0.0, stop=1.0/(2.0*T), num=n//2)
```

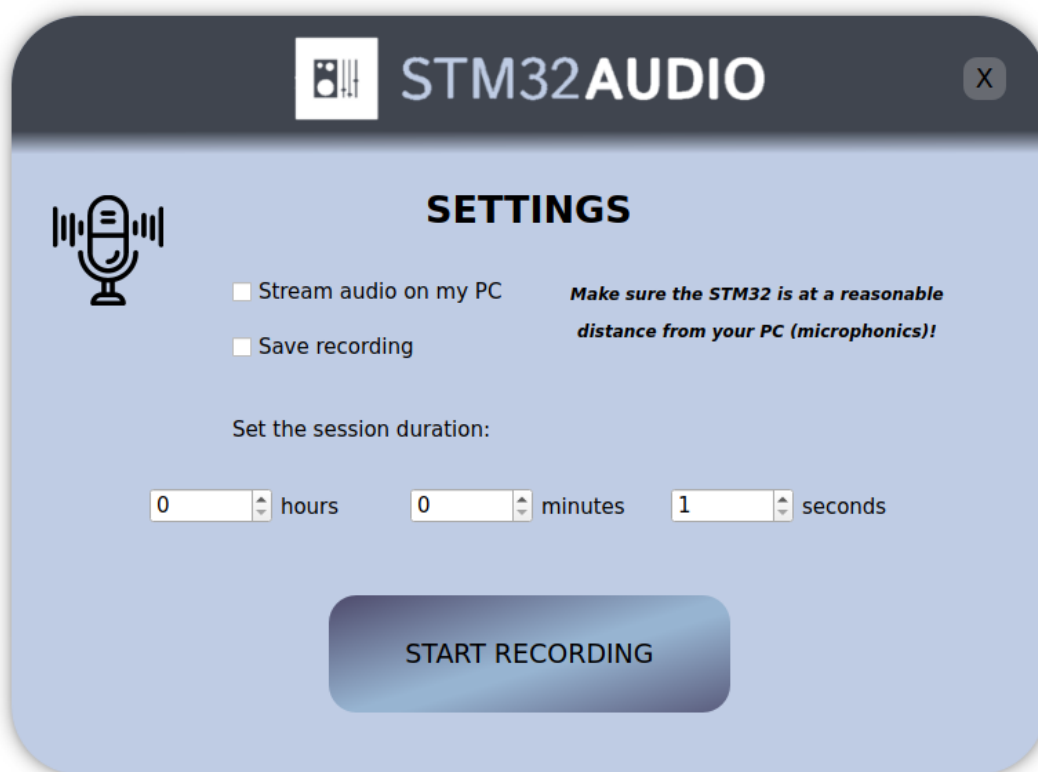
Isječak koda 3.1: Fourierova transformacija

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



Slika 3.2: Primjer prikaza analize audiozapisa

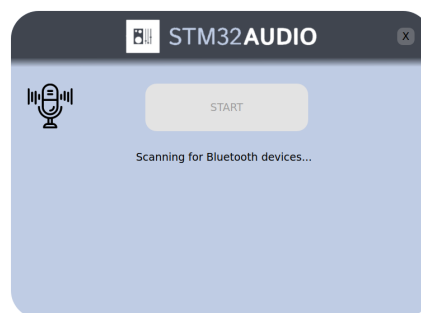
Ovdje će biti tekst koji govori o odabiru parametara.



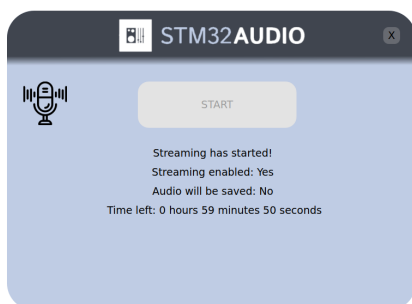
Slika 3.3: Sučelje za podešavanje parametara



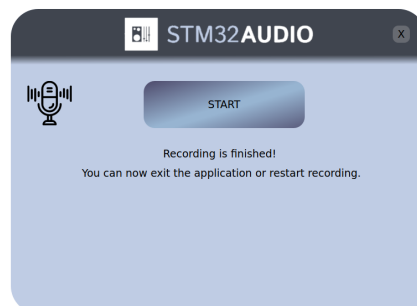
Slika 3.4: Prije pokretanja snimanja



Slika 3.5: Skeniranje uređaja



Slika 3.6: Snimanje zvuka



Slika 3.7: Završetak snimanja

4. 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 mikrofon, 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