



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

НОВИ САД

Департман за рачунарство и аутоматику

Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR) РАД

Кандидат: Радован Бирдић

Број индекса: 11946

Тема рада: Једно решење програмског окружења за графичку анализу аудио сигнала на E2LP платформи

Ментор рада: Др Миодраг Темеринац

Нови Сад, октобар, 2014



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Радован Бирдић		
Ментор, МН:	Др Миодраг Темеринац		
Наслов рада, НР:	Једно решење програмског окружења за графичку анализу аудио сигнала на Е2LP платформи		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2014		
Издавач, ИЗ:	Ауторски репринг		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)			
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:			
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У овом раду је реализовано програмско окружење за графичку анализу аудио сигнала. Окружење је верификовано на Е2LP платформи у склопу лабораторијских вежби.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	Др Небојша Пјевалица, доцент	
	Члан:	Др Иван Каштелан, асистент	Потпис ментора
	Члан, ментор:	Др Миодраг Темеринац, ред. професор	



KEY WORDS DOCUMENTATION

Accession number, ANO:			
Identification number, INO:			
Document type, DT:	Monographic publication		
Type of record, TR:	Textual printed material		
Contents code, CC:	Bachelor Thesis		
Author, AU:	Radovan Birdić		
Mentor, MN:	Miodrag Temerinac, PhD		
Title, TI:	One implementation of the programming environment for graphical analysis of audio signals on E2LP platform.		
Language of text, LT:	Serbian		
Language of abstract, LA:	Serbian		
Country of publication, CP:	Republic of Serbia		
Locality of publication, LP:	Vojvodina		
Publication year, PY:			
Publisher, PB:	Author's reprint		
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, PD: <small>(chapters/pages/ref./tables/pictures/graphs/appendices)</small>			
Scientific field, SF:	Electrical Engineering		
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, S/KW:			
UC			
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, N:			
Abstract, AB:	This paper presents the implementation of the programming environment for graphical analysis of the audio signal. The environment is verified on E2LP platform as a part of laboratory exercises.		
Accepted by the Scientific Board on, ASB:			
Defended on, DE:			
Defended Board, DB:	President:	PhD Nebojša Pjevalica	
	Member:	PhD Ivan Kaštelan	Menthor's sign
	Member, Mentor:	PhD Miodrag Temerinac	

Zahvalnost

Zahvaljujem se mr Željku Lukaču na stručnoj pomoći tokom izrade završnog (bachelor) rada i porodici koja mi je bila velika podrška tokom studiranja.



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



SADRŽAJ

1.	Uvod	1
2.	Teorijske osnove.....	3
2.1	E2LP platforma	3
2.1.1	Marvell Armada 1500.....	4
2.2	Android operativni sistem.....	4
2.3	Android aplikacije.....	4
2.4	Android GUI – Grafička sprega.....	5
2.4.1	View	5
2.4.2	Widgets.....	5
2.4.3	Layouts - Rasporedi.....	6
2.4.4	Events - Događaji	6
2.4.5	Adapteri	6
2.5	DFT – Diskrenta Furijeova transformacija	6
2.6	FFT – Brza Furijeova transformacija.....	7
2.7	Prozorska funkcija Blackman	7
3.	Koncept rešenja	8
4.	Programsko rešenje.....	13
4.1	Klasa FFTAnalyzer.java	13
4.1.1	Klasa DrawAudioAnalyze.....	14
4.1.2	Funkcija makeWindow.....	14
4.1.3	Funkcija reduceTo1024	14
4.1.4	Funkcija returnScale	15
4.1.5	Funkcija resetDraw.....	15

4.1.6	Funkcija startOptions.....	15
4.1.7	Funkcija enableDisableMenu	15
4.1.8	Funkcija writeSettings	15
4.1.9	Paket android.graphics	16
4.2	Klasa WavFile.java	16
4.2.1	Funkcija openWavFile.....	16
4.2.2	Funkcija readFrames	16
4.2.3	Funkcija getNumFrames	17
4.3	Klasa Complex.java	17
4.3.1	Funkcija public Complex	17
4.4	Klasa FFT.java.....	17
4.4.1	Funkcija fft	17
5.	Testiranje programskog okruženja i primer izvođenja vežbe.....	18
6.	Zaključak	20
7.	Literatura	21

SPISAK SLIKA

Slika 2.1 Arhitektura E2LP ploče	3
Slika 2.2 Dodatna ploča sa Marvell Armada 1500 integrisanim kolom	4
Slika 2.3 Izgled ViewGroup stabla	5
Slika 3.1 Izgled aplikacije.....	8
Slika 3.2 Generisanje padajućeg menija za dužinu prozora	9
Slika 3.3 Odabiranje pozicije početka obrade	9
Slika 3.4 Opcija uključivanja prozorske funkcije	10
Slika 3.5 IsCRTavanje u <i>Points</i> režimu	11
Slika 3.6 IsCRTavanje u <i>Lines</i> režimu	11
Slika 3.7 Prikaz uvećanja dela signala.....	11
Slika 3.8 Dijagram blok toka obrade	12
Slika 5.1 Spektar snage signala sa dužinom prozora $N = 64$	18
Slika 5.2 Spektar snage signala sa dužinom prozora $N = 512$ (uvećano).....	19
Slika 5.3 Spektar snage signala sa dužinom prozora $N = 4096$ (uvećano).....	19

SPISAK TABELA

Tabela 4.1 Sadržaj osnovnog paketa programskog rešenja 13

SKRAĆENICE

E2LP	- <i>Embedded Engineering Learning Platform</i>
FFT	- <i>Fast Fourier Transform</i>
DFT	- <i>Discrete Fourier Transform</i>
PNG	- <i>Portable Network Graphics</i>
SDK	- <i>Software Development Kit</i>
WAV	- <i>Waveform Audio File Format</i>
GUI	- <i>Graphical User Interface</i>
XML	- <i>Extensible Markup Language</i>

1. Uvod

Potrebno je realizovati programsko okruženje za grafičku analizu audio signala koje se može koristiti na E2LP - platformi u nastavi iz digitalne obrade signala. Realizovano okruženje je potrebno verifikovati u sklopu laboratorijskih vežbi.

Vežbe su zamišljene tako da se evaluacija testnog audio signala vrši u MATLAB-u i da se ispita u programskom okruženju. Programsко okruženje je potrebno pokrenuti na platformi sa ugrađenim resursima, u ovom slučaju na E2LP platformi.

Da bi zadatak u potpunosti bio uspešno izvršen studentima su potrebna sledeća znanja:

1. Poznavanje gradiva iz predmeta OAiS DSP 1 (odabiranje, kvantizacija, DFT,FFT).
2. Poznavanje Matlab i Java programskih jezika.
3. Poznavanje rada na E2LP platformi.
4. Poznavanje rada pod operativnim sistemom Android.

Ovaj rad je sačinjen od 7 poglavlja:

- U prvom poglavlju je dat kratak opis rada.
- Drugo poglavlje sadrži teorijske osnove potrebne za uspešnu realizaciju svih elemenata ovog rada.
- U trećem poglavlju dat je koncept rešenja.
- Četvrto poglavlje sadrži detaljan opis svih modula i funkcija koje se nalaze u njima.

- Poglavlje broj pet sadrži opis testiranja programskog okruženja i primer izvođenja vežbe.
- Šesto poglavlje sadrži zaključak.
- U sedmom poglavlju je dat spisak korišćene literature.

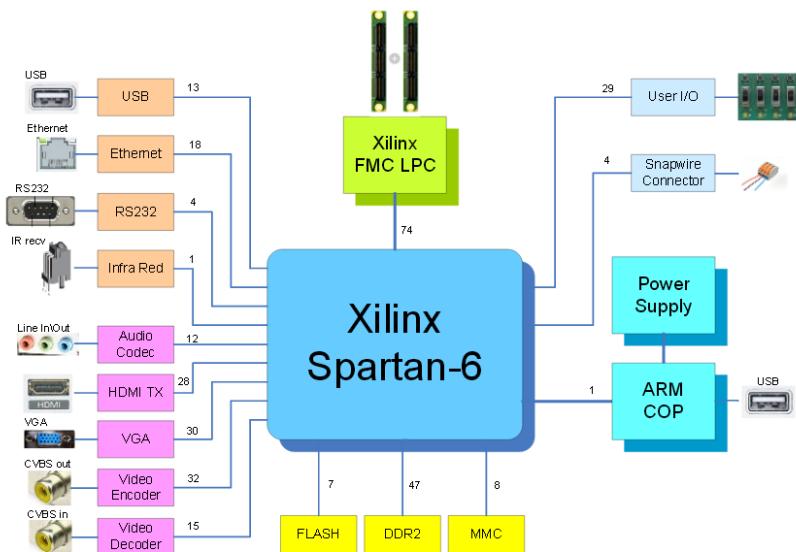
2. Teorijske osnove

2.1 E2LP platforma

Osnovna ideja E2LP projekta je da se obezbedi jedinstvena platforma koja će pokrivati celokupan proces za učenje o ugrađenim sistemima. Kako ugrađeni sistemi postaju sve složeniji, inžinjeri moraju da izađu na kraj sa sve kompleksnijim problemima, da predlažu nova rešenja i sisteme u cilju postizanja specifičnih zadataka. Zbog tog rasta kompleksnosti ugrađenih sistema, E2LP treba da korisnicima omogući ovakav pristup rada u nastavnom planu [1].

E2LP platforma (Slika 2.1) se sastoji od glavne ploče i od dodatnih ploča. Dodatne ploče zasnovane na sledećim integrisanim kolima:

- Mikrokontroler zasnovan na ARM arhitekturi
- Marvell ARMADA 1500, namenjen za digitalnu obradu signala

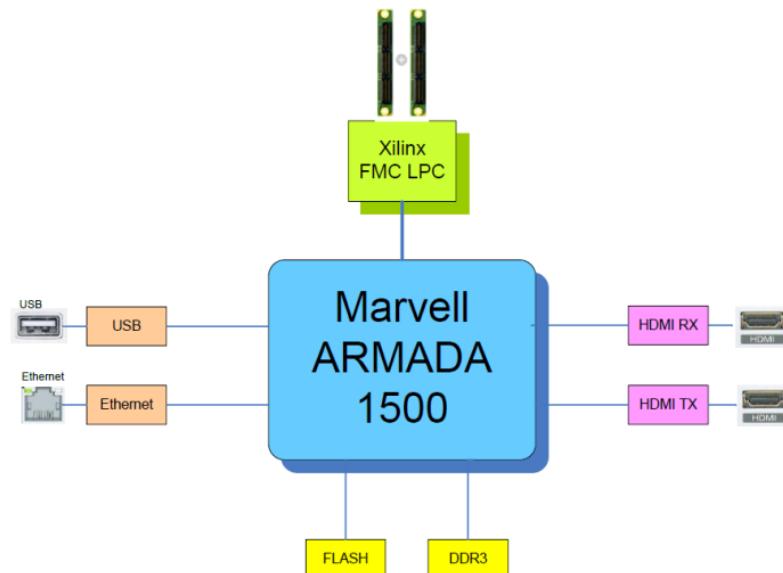


Slika 2.1 Arhitektura E2LP ploče

2.1.1 Marvell Armada 1500

Dodatna ploča sa Marvell Armada 1500 [2] integriranim kolom može da koristi sve resurse E2LP ploče i dodatno proširuje mogućnosti jer sadrži USB port, HDMI izlaz, LAN ulaz, DDR3 memoriju i Flash memoriju. Namenjena je za izradu laboratorijskih vežbi i pokriva oblasti:

- Obrada jednodimenzionalnih signala
- Obrada dvodimenzionalnih signala
- Računarske mreže i komunikacije
- Sistemsku programsku podršku
- Razvoj programske podrške za Android



Slika 2.2 Dodatna ploča sa Marvell Armada 1500 integriranim kolom

2.2 Android operativni sistem

Android™ operativni sistem je trenutno najrasprostranjeniji operativni sistem za mobilne telefone, zasnovan je na Linux kernelu i prilagođen je tako da se može koristiti na većini mobilnih uređaja, uključujući tablet, laptop, netbook, smartbook računare, čitače elektronskih knjiga, pa čak i ručne satove.

2.3 Android aplikacije

Android aplikacije se pišu korišćenjem Java programskog jezika. Android SDK alati prevode kod, zajedno sa svim podacima i potrebnim datotekama u Android paket tj. u

arhivsku datoteku sa ekstenzijom *.apk*. Ova datoteka se šalje na Android uređaj, gde se postavlja (engl. *Install*) i omogućava pristup od strane krajnjih korisnika.

Aplikacija može da zatraži dozvolu za pristup podacima[4] kao što su kontakti korisnika, SMS poruke, kamera, i drugo. Sve dozvole programer nabraja u *AndroidManifest.xml* datoteci, a moraju biti odobrene od strane korisnika u trenutku postavljanja aplikacije inače aplikacija neće biti postavljena.

2.4 Android GUI – Grafička sprege

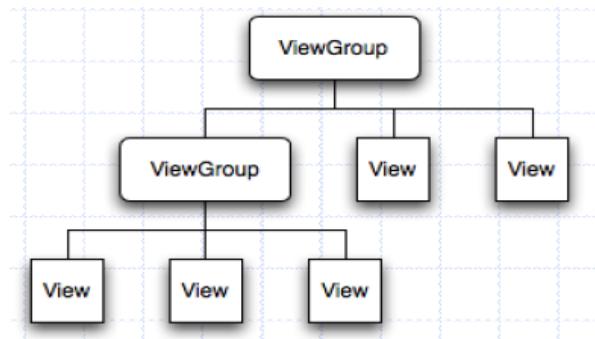
Za realizovanje grafičke sprege pomoću koje korisnik ostvaruje komunikaciju sa aplikacijom, ponuđen je niz elemenata [5]:

- Komponente (view, widgets).
- Raspored (layouts) po kom će date komponente biti iscrtane.
- Događaji (events)
- Adapteri

2.4.1 View

ViewGroup je osnovna klasa za layout kontejnere koji sadrže *View* elemente kao svoje potomke i odgovorna je za njihov raspored na ekranu.

View je osnovni element za pravljenje korisničke sprege. To je pravougaoni objekat na ekranu odgovoran za iscrtavanje sadržaja i obradu događaja. *View* komponente mogu biti organizovane u strukturu poput stabla (Slika 2.3). Definišu se u *xml* deskriptoru ili direktno u kodu.



Slika 2.3 Izgled ViewGroup stabla

2.4.2 Widgets

Predstavlja komponente korisničke sprege:

- Komponente za forme, za interakciju s korisnikom:
 - *TextView* - ispisuje tekst

- *Button* - dugme
 - *ToggleButton* – dugme sa unutrašnjim stanjem (uključen/isključen)
 - *Spinner* – padajući meni. Sastoji se od zaglavlja koje prikazuje trenutno odabranu opciju i prozora koji prikazuje sve opcije.
 - *CheckBox, RadioButton* – dugmad za izbor
- Polje za unos teksta(slova, bojevi) *EditText*.

2.4.3 Layouts - Rasporedi

Klase za prostorni razmeštaj komponenti na ekranu. Jedna aktivnost može da ima jedan ili više *layout-a*. Postoji mogućnost kombinovanja (jedan *layout* unutar drugog).

- *LinearLayout* : slaže komponente jednu do druge, horizontalno ili vertikalno.
- *RelativeLayout* : komponente se pozicioniraju relativno jedna u odnosu na drugu.
- *TableLayout* : omogućuje tabelaran prikaz komponenti.
- *TabLayout* : Klasa *TabActivity* nasleđuje *Activity* i služi kao osnovna klasa u kojoj će biti *Tab*.

2.4.4 Events - Događaji

Android aplikacije reaguju na događaje korisničkog interfejsa. Dodir po kontroli se pretvara u događaj koji registruje odgovarajući “*Listener*”.

2.4.5 Adapteri

Adapteri povezuju komponente sa kolekcijama podataka, obično povezuju komponente koje listaju ili prikazuju više od jednog podatka (liste, spineri, itd.).

2.5 DFT – Diskretna Furijeova transformacija

Diskretna Furijeova transformacija (discrete Fourier transform, DFT) je jedna od najčešće korišćenih transformacija u digitalnoj obradi signala. Ona omogućava da se diskretni periodični signal predstavi sumom sinusnih komponenti pridružujući svakoj komponenti odgovarajuću amplitudu i fazni pomeraj. U opštem slučaju, DFT transformiše niz od N kompleksnih odbiraka u vremenskom domenu u niz od N kompleksnih koeficijenata u frekvencijskom domenu.

$$X(n) = \sum_{k=0}^{N-1} x(n) * e^{-2j\pi \frac{nk}{N}}$$

2.6 FFT – Brza Furijeova transformacija

Zbog brzine izračunavanja, za računanje spektra signala koristićemo brzu Furijeovu transformaciju. Koristi se algoritam sa decimacijom u vremenu koji se dobija tako što se pri računanju Diskretnе kosinusne transformacije grupišu parni i neparni odbirci signala $x(k)$.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi nk}{N}} = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{(2n+1)k}$$

Korišćenjem ove procedure, N -člana DFT je svedena na dve $N/2$ -člane DFT. Sledeći istu proceduru decimacije, svaka $N/2$ -člana DFT se može svesti na dve $N/4$ -člane DFT, i tako sve dok se izračunavanje ne svede na 2-člane DFT [3].

2.7 Prozorska funkcija Blackman

Prilikom primene diskretne Furijeove transformacije na signale čiji sadržaj se ne ponavlja sa periodom N signal se podeli na blokove dužine N i na svaki blok se primeni DFT. Da bi se izbegao uticaj blokovske obrade signala koristi se tehnika pod nazivom prozoriranje. Prozoriranjem se blok odbiraka koji se obrazuje uobličava u cilju smanjenja diskontinuiteta na početku i kraju bloka [3].

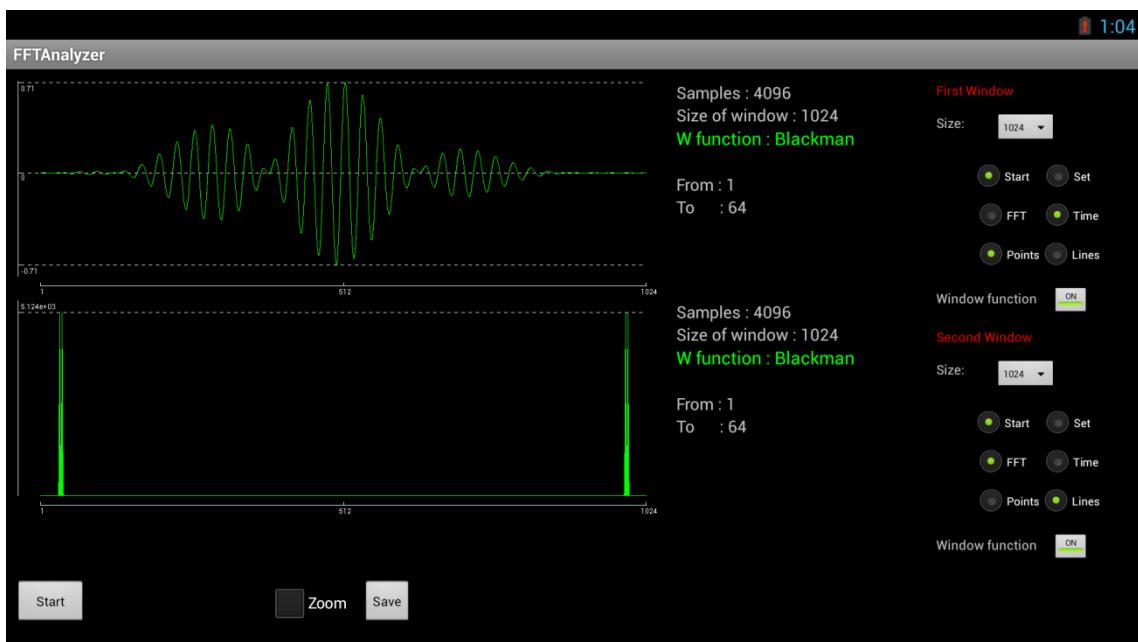
Za svrhu prozoriranja u radu se koristila Blackman prozorska funkcija.

$$x(k) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right)$$

$$a_0 = \frac{1-\alpha}{2}; \quad a_1 = \frac{1}{2}; \quad a_2 = \frac{\alpha}{2}$$

3. Koncept rešenja

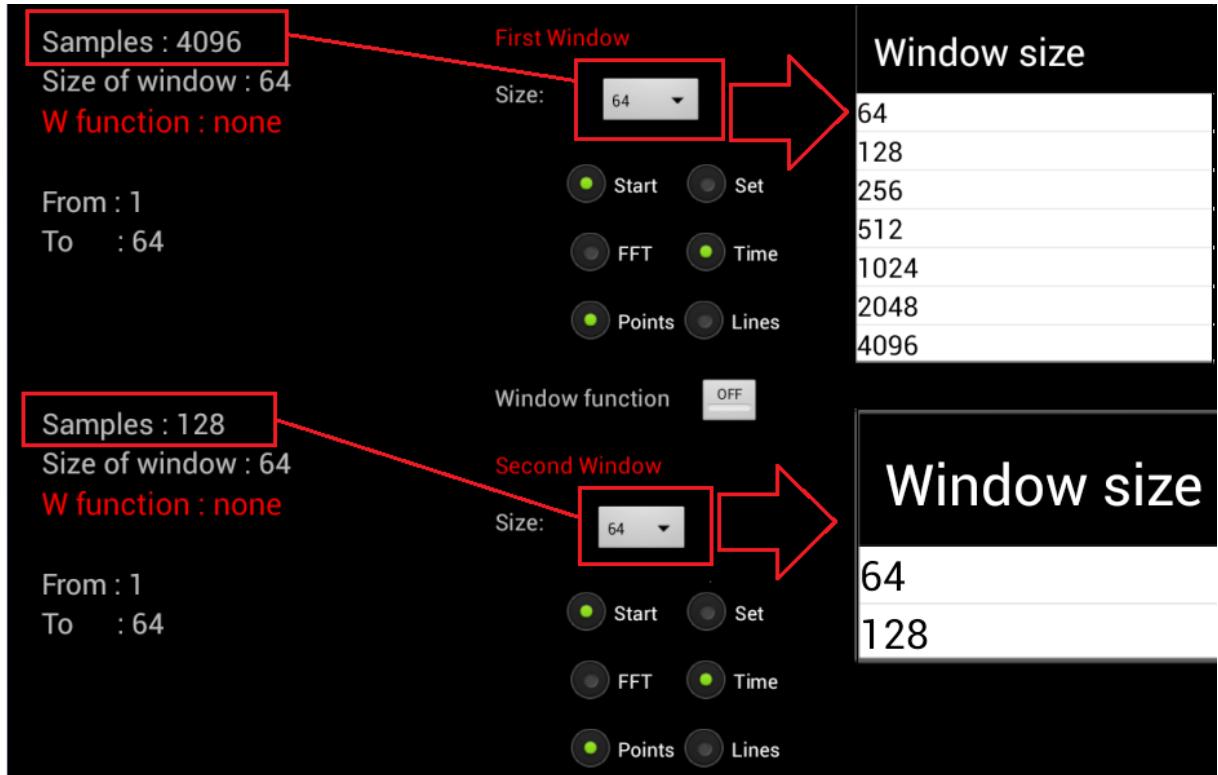
Ideja je da se realizuje programsko okruženje koje bi omogućilo korisnicima korišćenje aplikacije sa minimalnim izmenama u kodu (programsко zadavanje putanje do signala za obradu). Ono bi im omogućilo da lakše savladaju pomenutu oblast. Zbog toga je potrebno realizovati aplikaciju koja je što jednostavnija za korišćenje. Izgled aplikacije je prikazan na slici 3.1.



Slika 3.1 Izgled aplikacije

Na samom startu se vrši učitavanje audio signala za obradu. Signal koji se učitava mora biti u *.wav* formatu. Nakon učitavanja signala, bira se veličina prozora za obradu istog. Biranje veličine prozora je realizovano tako da je korisniku ponuđen padajući meni sa vrednostima

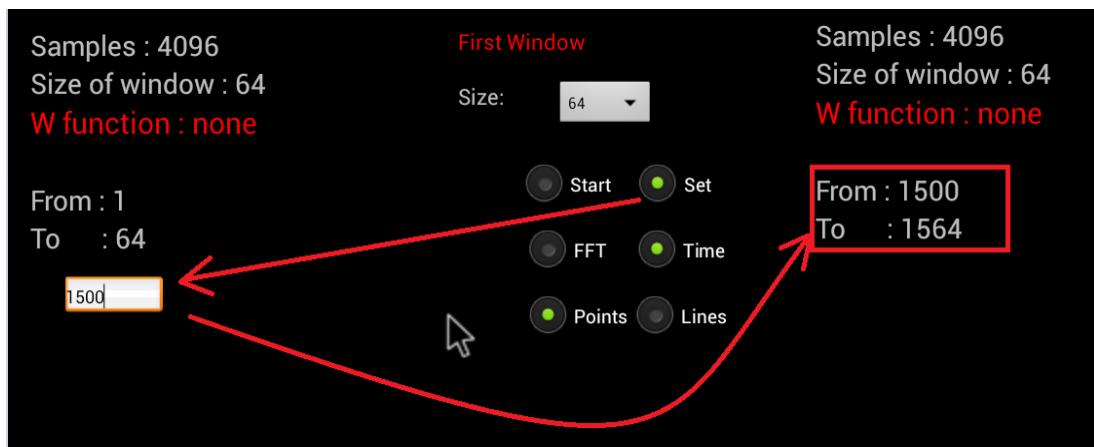
stepena broja 2, čiji se sadržaj generiše nakon učitavanja audio signala. Ako učitani signal sadrži 128 odbirka, u padajućem meniju će maksimalna dužina prozora biti 128. Kako se dužina učitanog signala povećava ili smanjuje, tako se i padajući meni menja (Slika 3.2).



Slika 3.2 Generisanje padajućeg menija za dužinu prozora

Kada je signal učitan, može da se odabere tačno pozicija od kog će početi obrada(isCRTavanje) signala. Postoji mogućnost da se počne od samog početka signala ili od neke pozicije u signalu (Slika 3.3). U zavisnosti od pozicije od koje počinje analiza signala, pozicija na kojoj se analiza završava se dobija formulom:

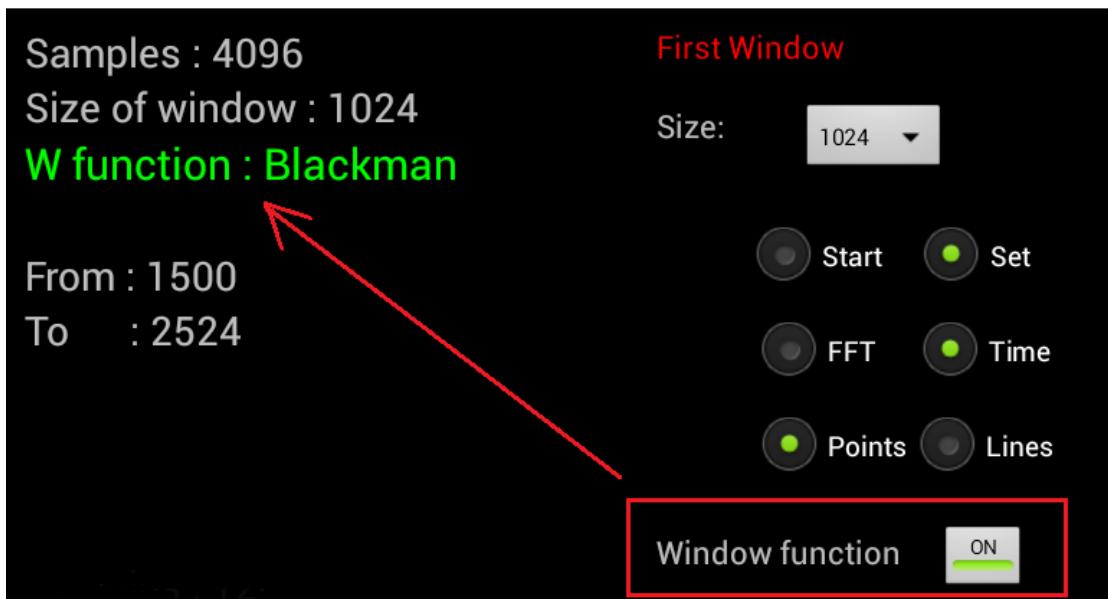
$$\text{krajnji odbirak} = \text{početni odbirak}(x) + \text{dužina prozora}$$



Slika 3.3 Odabiranje pozicije početka obrade

Nakon odabране dužine prozora i pozicije u signalu od koje će obrada početi, na korisniku je da odabere da li će nad signalom biti primenjena ili ne prozorska funkcija. U kodu su realizovane Blackman, Hamming i Hann prozorska funkcija, ali se za trenutne potrebe koristi samo prva. U slučaju da se odabere opcija da signal bude prozoriran, pomnoži se sa prozorskog funkcijom i prosleđuje na dalju obradu (Slika 3.4).

$$\text{novi signal} = \text{signal} * \text{prozorska funkcija}$$

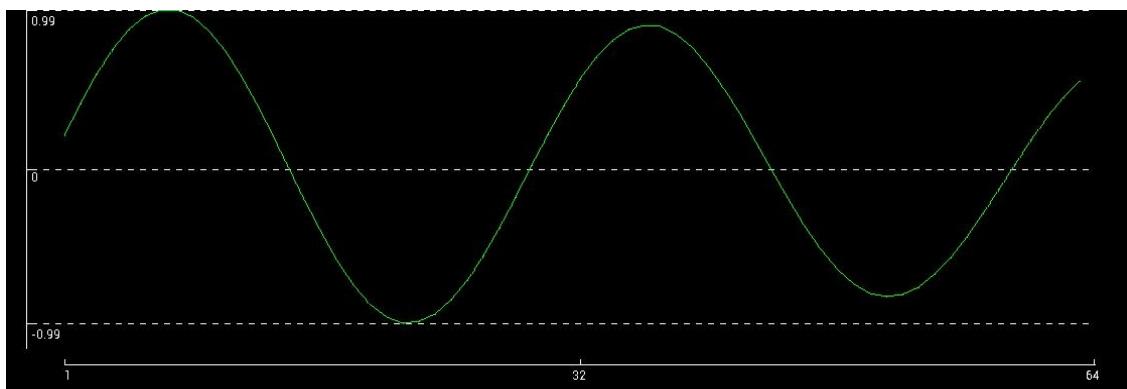
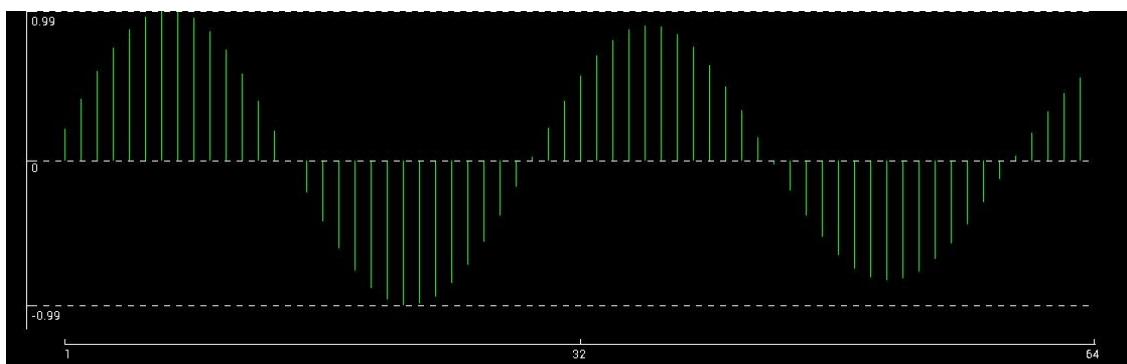


Slika 3.4 Opcija uključivanja prozorske funkcije

Poslednji korak u obradi je računanje FFT-a nad signalom. U ovom radu je ralizovana FFT sa decimacijom u vremenu. Deo signala nad kojim se vrši računanje, mora biti dužine stepena dvojke, a to je obezbeđeno kod biranja dužine prozora.

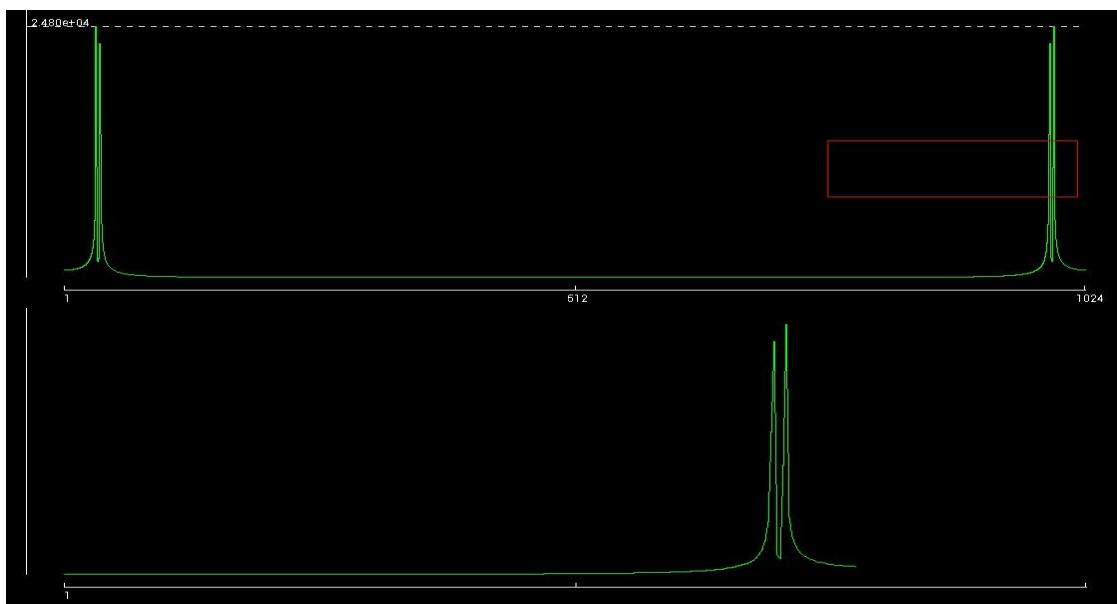
Kada signal prođe ceo tok obrade, na kraju samo ostaje da se iscrtava dano rešenje. Za iscrtavanje rezultata su data dva prozora, svaki je širine 1024. U slučaju da je izabrana dužina prozora manja od 1024, vršiće se interpolacija signala koji se iscrtava da bi se "razvukao" po celom prozoru. Ukoliko je dužina prozora veća od 1024, rezultujući signal se decimira. U prozorima se pored rezultujućeg signala iscrtavaju i dve skale, horizontalna i vertikalna. Obe skale se automatski menjaju tokom rada. Horizontalna skala prikazuje dužinu signala koji se iscrtava a vertikalna njegovu minimalnu i maksimalnu vrednost. Ponuđene su dve opcije za iscrtavanje:

- *Points*: iscrtavanje ekvivalentno Matlabovoj funkciji za iscrtavanje *plot* (Slika 3.5).
- *Lines*: iscrtavanje ekvivalentno Matlabovoj funkciji za iscrtavanje *stem* (Slika 3.6).

Slika 3.5 IsCRTavanje u *Points* režimuSlika 3.6 IsCRTavanje u *Lines* režimu

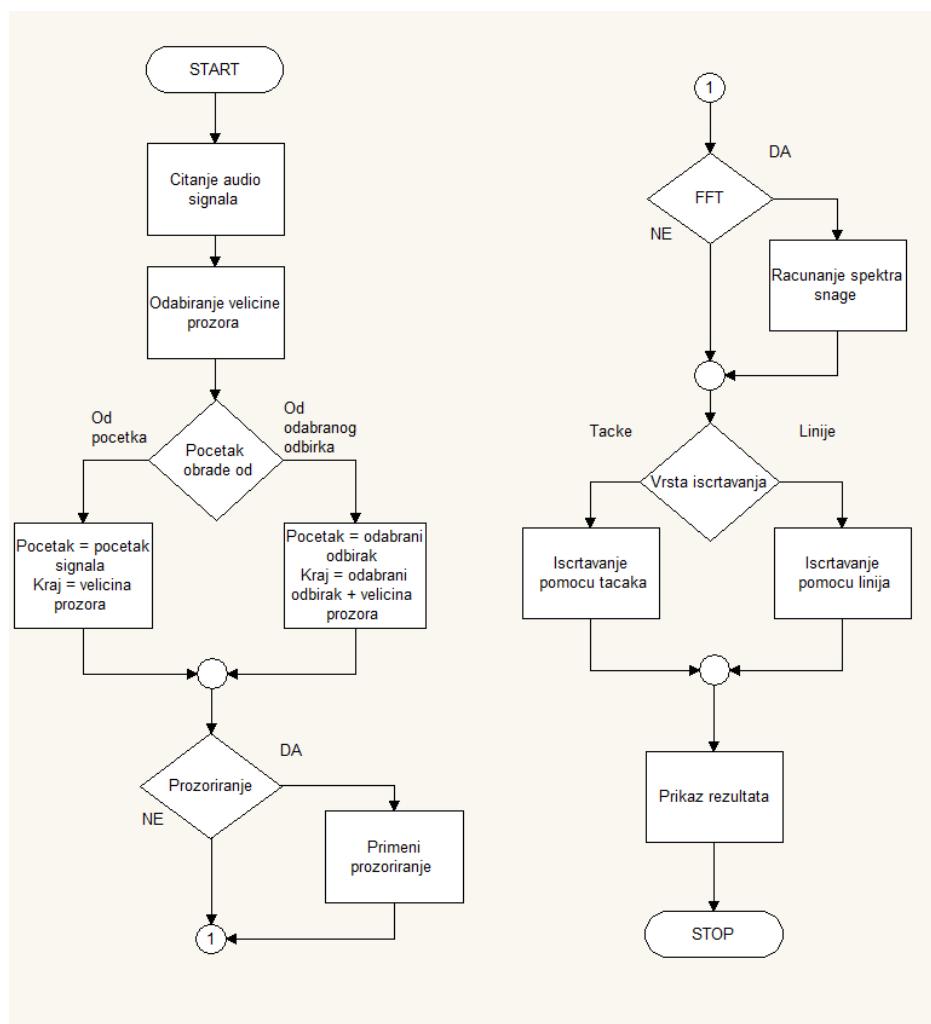
Osim navedenih opcija, u aplikaciji su još ponuđene i opcije:

- *Zoom*: korisnik može da rezultate obrade uveća za neku detaljniju analizu slike (Slika 3.7)
- *Save*: korisnik može da snimi rezultate obrade u formatu .PNG



Slika 3.7 Prikaz uvećanja dela signala

Algoritam po kom fukcionije program je prikazan na Slici 3.8.



Slika 3.8 Blok dijagram tokova obrade

4. Programsко rešenje

Programska realizacija projektnog zadatka napisana je u Java programskom jeziku. Realizovano rešenje zasnovano je na korišćenju Android sistemskih klasa, nasleđivanjem, menjanjem postojećih i dodavanjem novih. Verzija Androida koja se koristila je 4.2.2 *Jelly Bean*. Eclipse ADT se koristio kao programsko okruženje u kome je pisan kod. Osnovni paket koji čini programsko rešenje zadatka prikazan je u tabeli 4.1.

Naziv paketa	Opis paketa	Sadržaj paketa - Klase
audioanalyze.android.audio	Sadrži skup klasa koje služe za obradu signala i za prikaz grafike na ekranu	FFTAnalyzer.java WavFile.java Complex.java FFT.java

Tabela 4.1 Sadržaj osnovnog paketa programskog rešenja

4.1 Klasa FFTAnalyzer.java

Ovo je klasa koja se prva startuje nakon pokretanja aplikacije. Ona nasledjuje Android sistemsku klasu *Activity*.

Klasa sadrži sledeće operacije:

- Instancira celokupan grafički deo aplikacije, dva prozora za isrtavanje, glavni meni aplikacije, dugmad za pokretanje i zaustavljanje rada aplikacije i za

snimanje rezultata iscrtavanja i opcionalno dugme za uvećavanje rezultata iscrtavanja.

- Vrši učitavanje signala i obradu nad njim
- Implementira *OnClickListener* Androidovu sistemsku spregu, na osnovu čega je moguće kliknuti na neko od dugmadi iz menija

Pored navednih operacija, klasa FFTAnalyzer sadrži i druge Android pakete potrebne za realizaciju aplikacije.

4.1.1 Klasa DrawAudioAnalyze

Nasleđuje klasu AsynTask, koja omogućava pravilno i lako korišćenje ulazno-izlazne niti. Pomoću nje je moguće obavljanje pozadinske operacije tokom rada same aplikacije. U telu funkcije *doInBackground()* je smeštena celokupna obrada učitanog signala. Pozivom funkcije *onProgressUpdate()* iscrtava se rezultat obrade. Ovoj funkciji se prosleđuje niz tipa *double* koji sadrži vrednosti koje se iscrtavaju.

```
public class DrawAudioAnalyze extends AsyncTask<Void, double[], Void> {

    @Override
    protected Void doInBackground(Void... params) {

        ...
    }

    protected void onProgressUpdate(double[]... y_byte) {

    }
}
```

4.1.2 Funkcija makeWindow

```
public double[] makeWindow(int n)
```

U ovoj funkciji je realizovana prozorska funkcija Blackman. Kao ulazni parameter šalje se dužina prozora koja je odabrana iz padajućeg menija. Povratna vrednost je niz popunjeno koeficijentima za prozoriranje.

4.1.3 Funkcija reduceTo1024

```
private double[] reduceTo1024(int choice)
```

Funkcija se poziva u slučaju da je izabrana dužina prozora veća od 1024. Pošto je površina za iscrtavanje širine 1024, svi signali koji su duži, moraju da se svedu na datu dužinu. U njoj se radi decimacija signala koji se iscrtava. Ulazni parametar predstavlja izbor na osnovu koga se zna nad kojim signalom treba da se izvrši decimacija. Izlazni parametar je niz sa decimiranim signalom.

4.1.4 Funkcija returnScale

```
public void returnScale()
```

Pozivom ove funkcije se horizontalna i vertikalna skala iscrtavaju sa početnim vrednostima. Koristi se kod izlaska iz *Zoom* moda, kada je potrebno da se ponovo iscrtava originalan signal pa je zbog toga potrebno da i obe skale budu prilagođene.

4.1.5 Funkcija resetDraw

```
public void resetDraw()
```

Ova funkcija služi za ponovno iscrtavanje prvobitnog signala. Koristi se kod izlaska iz *Zoom* moda, kada je potrebno da se ponovo iscrtava originalan signal.

4.1.6 Funkcija startOptions

```
public void startOptions()
```

Funkcija koja inicijalizuje meni aplikacije. Uključuje neke od opcija koje korisnik može da promeni pre samog pokretanja obrade (iscrtavanja).

4.1.7 Funkcija enableDisableMenu

```
public void enableDisableMenu(int menuChoice)
```

Pozivom ove funkcije, omogućuje se i onemogućuje korišćenje menija aplikacije. Poziva se pri startovanju obrade, kada se onemogući korišćenje menija i njegovo omogućavanje kada se obrada završi. Ulazni parametar omogućuje da funkcija zna kad koju operaciju treba da uradi.

4.1.8 Funkcija writeSettings

```
public void writeSettings()
```

Pozivom ove funkcije, pored oba prozora za iscrtavanje se ispisuju informacije o signalu koji se iscrtava:

- Dužina prozora
- Da li se primenjuje prozorska funkcija ili ne
- Početna i krajnja pozicija između kojih se vrši obrada signala

4.1.9 Paket android.graphics

Ovaj paket je uključen u klasu *FFTAnalyzer* i on sadrži klase koje korisniku omogućuju 2D iscrtavanje.

Da bi crtanje bilo moguće, potrebne su 4 komponente:

- *Bitmap*: bit mapa koja sadrži piksele.
- *Canvas*: klasa koja sadrži *draw* pozive za crtanje u datu bit mapu.
- *Vrsta crtanja*: iscrtavanje teksta *drawText()*, linije *drawLine()*, pravougaonika *drawRect()*, tačke *drawPoint()*, putanje spajanjem tačaka *drawPath()*, bit mape *drawBitmap()* i slično.
- *Paint*: klasa koja obezbeđuje bojenje i stil crtanja.

Canvas klasa se koristi iz razloga što pomoću nje može direktno da se utiče na iscrtavanje u aplikaciji.

4.2 Klasa WavFile.java

U ovoj klasi su realizovane funkcije za pravilno čitanje i pisanje .wav datoteka [5].

4.2.1 Funkcija openWavFile

```
public static WavFile openWavFile(InputStream file)
```

Pozivom ove funkcije, otvara se .wav datoteka koja se čija putanja je data ulaznim parametrom *file*.

Primer putanje:

```
Environment.getExternalStorageDirectory().getAbsolutePath() + "/test1.wav"
```

4.2.2 Funkcija readFrames

```
public int readFrames(double[] sampleBuffer, int numFramesToRead)
```

Ova funkcija omogućava čitanje .wav datoteke. Ulazni parametari su niz *sampleBuffer* tipa *double* u koji se smešta broj odbiraka *numFramesToRead*. U niz *sampleBuffer* se smeštaju pročitani odbirci signala. Broj odbiraka se odabira iz padajućeg menija koji sadrži ponuđene dužine prozora. Povratna vrednost je broj odbiraka signala koji su isčitani.

4.2.3 Funkcija getNumFrames

```
public long getNumFrames()
```

Pozivom funkcije dobija se povratna vrednost koja sadrži informaciju o ukupnoj dužini signala koji se obrađuje.

4.3 Klasa Complex.java

U ovoj klasi su realizovane operacije nad kompleksnim brojevima, kao što su osnovne aritmetičke operacije, trigonometrijske operacije i slično.

4.3.1 Funkcija public Complex

```
public Complex(double real, double imag)
```

Stvara se objekat klase sa realnim i imaginarnim delom, koji se u zadatku koristi za dalje računanje.

4.4 Klasa FFT.java

Klasa sadrži funkcije za računanje brze Furijeove transformacije , inverzne brze Furijeove transformacije i konvolucije.

4.4.1 Funkcija fft

```
public static Complex[] fft(Complex[] x)
```

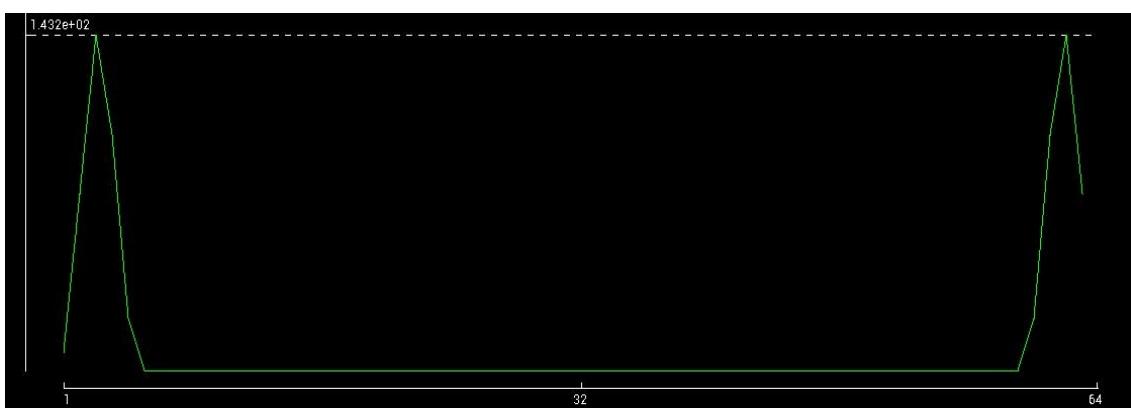
Funkcija za računanje brze Furijeove transformacije. Kao ulazni parametar, prosleđuje joj se niz kompleksnih brojeva. Povratna vrednost je niz sa izračunatim vrednostima brze Furijeove transformacije u kompleksnom obliku.

5. Testiranje programskog okruženja i primer izvođenja vežbe

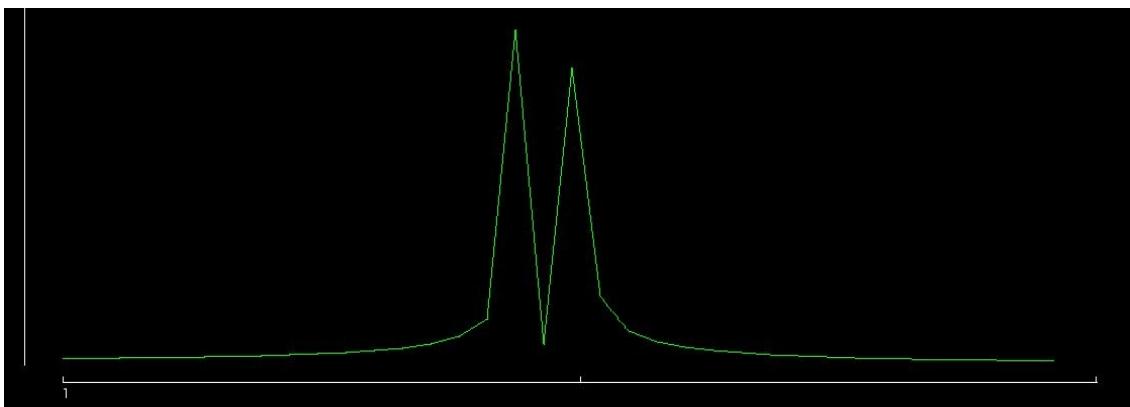
Primer primene okruženja u nastavi biće prikazan kroz jednu od laboratorijskih vežbi iz oblasti digitalne obrade signala. Zadatak vežbe je da se prikažu spektri snage učitanog signala za različite dužine prozora ($N = 64, 512$ i 4096). Cilj vežbe je bio da se pokaže da se sa povećanjem veličine prozora, povećava i spektralna rezolucija. Učita se zadati signal i podešavaju su sledeće opcije:

- Iz padajućeg menija se za prvi prozor iscrtavanja odabere vrednost za dužinu prozora ($N = 64$, $N = 512$, $N = 4096$).
- Početak analize za signal se postavlja da bude od prvog odbirka signala (odabira se opcija „Start“).
- Odabiraju se opcije za računanje spektra snage (odabira se opcija „FFT“).
- Za vrstu iscrtavanja se odabira spajanje tačaka u sva tri slučaja (odabira se opcija „Points“)

Na slikama 5.1, 5.2 i 5.3 su prikazani rezultati obrade.



Slika 5.1 Spektar snage signala sa dužinom prozora $N = 64$



Slika 5.2 Spektar snage signala sa dužinom prozora $N = 512$ (uvećano)



Slika 5.3 Spektar snage signala sa dužinom prozora $N = 4096$ (uvećano)

Treba napomenuti da je korisnik u obavezi da pre početka rada učita *.wav* datoteku, programski u kodu, koju želi da analizira. Datoteka se generiše u Matlab programu sledećom funkcijom:

```
wavwrite(signal, 'ime_datoteke.wav');
```

U datoteku *ime_datoteke.wav* se upiše vrednost niza *signal*

Ukoliko su vrednosti u nizu *signal* veće od 1, potrebno je pre upisa u datoteku sve vrednosti iz niza podeliti sa maksimalnom vrednošću iz niza da bi se izbeglo odsecanje signala.

6. Zaključak

U radu je prikazana realizacija programskog okruženja, opisani su njegova funkcionalnost i namena. Ciljni operativni sistem je Android operativni sistem. Prikazano je ponašanje na primeru laboratorijske vežbe i dati su rezultati analize. Poređenjem rezultata dobijenih pomoću Matlab programa i rezultata dobijenih pomoću realizovanog okruženja, može se zaključiti da aplikacija ispravno radi. Rešenje je ispitivano na E2LP platformi.

Kao problemi programskog okruženja, izdvajaju se:

- Rezolucija koju zahteva programsko okruženje. Mora biti 1920 x 1080 (HD), zbog pravilnog rasporeda komponenti.
- Analiziranje samo .wav datoteka.

U narednim verzijama programskog okruženja, moglo bi da se realizuju sledeće operacije:

- Učitavanje i analiza različitih formata audio datoteka.
- Učitavanje i analiza audio datoteka direktno iz aplikacije a ne programski u kodu.
- IsCRTavanje više signala u jednom prozoru.

Pošto aplikacija ima edukativnu svrhu, njen dalji razvoj će odrediti studenti i nastavnici svojim konstruktivnim predlozima.

7. Literatura

- [1] Ivan Kaštelan, "Produced and assembled E2LP base board", oktobar 2014.
- [2] Ivan Kaštelan, "Produced and assembled extension boards", oktobar 2014.
- [3] Prozoriranje, DFT, FFT, <http://www.rt-rk.uns.ac.rs/studijski-program-2009/vi-2009/oaisdsp1>, nastavni materijal, oktobar 2014.
- [4] J. Six, " Application Security for the Android Platform", 1st Edition, United States of America, O'Reilly Media, oktobar 2014.
- [5] Projektovanje namenskih računarskih struktura 1, <http://www.rt-rk.uns.ac.rs/studijski-program-2009/vi-2009/pnrs1>, nastavni materijal, septembar 2014.
- [6] Java WAV Files IO, <http://www.labbookpages.co.uk/audio/javaWavFiles.html>, oktobar 2014.