

Android Application: Sound Pressure Level Meter and Frequency Spectrum Analyzer

N. D. N. C. Bandara and W. K. I. L. Wanniarachchi*

*Department of Physics, University of Sri Jayewardenepura,
Nugegoda, Sri Lanka*

**Corresponding author
E mail : iwanni@sjp.ac.lk*

Abstract

In this paper we present the development of Sound Pressure Level Meter and Frequency Spectrum Analyzer application for Android based devices. The developed application can visualize real time waveform and frequency spectrum of an input audio signal in both linear and dB scale. The GPS location, current date and time are also embedded to the application for convenience. A comparison was carried out with currently available smartphone apps of similar type and the results are in good agreement with the app developed in this project. The current version of the app is capable of capturing signal of maximum frequency up to 4000Hz. Therefore, analysis of frequency spectrum of the voice transmission which range between 300Hz to 3400Hz can be carried out successfully.

Introduction

The introduction of smart phones has replaced many individual items in daily use such as, clock, calendars, camera, etc. Currently, the market share is dominated by Android, Apple iOS, and Blackberry based smartphones [1]. Since Google offers Android as open source, it has won hearts and minds of many mobile application developers. Consequently, there is a wide variety of Android apps to select for consumers compared to any other smartphone running platform. The microphone in a smartphone can be used to capture sound waves and with the use of an appropriate mobile app, the smartphone can be used as a sound level meter. This method of sound analysis is much cheaper compared to commercial sound level meters for everyday applications such as measurement of sound pollution due to recreational activities, factory sound level detection or for music composers. Most of the sound level detection apps currently available in the market are focused on detection of sound level only (eg. Sound meter by ABC apps, Decibel meter by Tacotty apps). However, if such a sound level monitoring app to be used in field work such as detecting sound levels in a city, finding the location requires additional apps such as GPS locators. Therefore, this project is focused on developing a mobile app with sound level pressure meter with ability to detect the GPS location, date and time, 2D wave plotter, and 2D real time frequency spectrum of an audio signal.

Audio Sound Capture and Signal Processing

Sound is created when a vibrating object causes ripples of varying air pressure. The pressure change is propagated by the collision of air molecules. These varying pressure cause the eardrum to move back and forth. Similar to the way the human ear works, an audio signal is captured using a microphone in a smartphone. This captures the air pressure variations and transformed them into electrical signals in the range of +1V to -1V. To store a sound wave in a computer, samples of the wave are taken and it is represented by a binary number consisting of a number of bits. The number of samples are taken for each channel per second is defined as the sampling rate. According to the Nyquist theorem, sampling rate should be at least twice as the maximum frequency value of the input signal [2-4]. The number of bits used to represent a single sample determines the resolution of the digitized sound. This process of digitization of the sound is called “pulse code modulation” (PCM). In this work 16-bit analog to digital conversion was used.

Audio hardware in latest smartphones consist of high quality analog to digital converters of 16-bit data width [4]. The audio hardware also support higher sampling rates for quality audio recording even at 44.1 kHz, which allows capturing audible frequency range 20 Hz to 20 kHz.

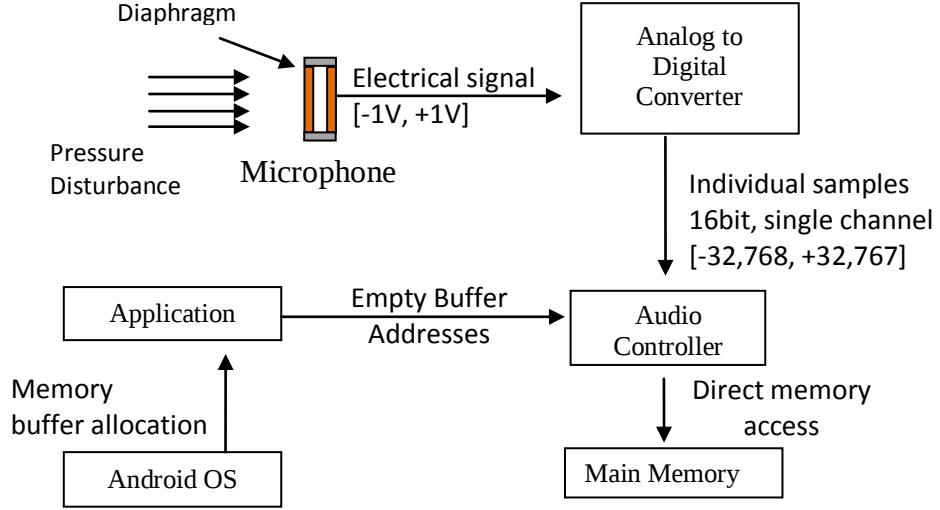


Figure 1: Digital audio recording in a smart phone

As shown in Figure 1, in smart phones, the analog to digital converter is connected to the audio control hardware. The Android application is responsible for sending commands to the audio controller and allocating memory for audio data in to a buffer. The audio controller then transfers buffered audio data directly to RAM where data can be further processed.

Fourier transform plays a major role in this process. It converts time-domain based real functions to frequencies (e.g. sound waves).

Fourier transform of $X(t)$ is $X(w)$. It converts time space to frequency space.

$$X(w) = F\{X(t)\} = \int_{-\infty}^{+\infty} X(t) \cdot e^{-i\omega t} dt$$

The time and frequency domains contain signals made up of several complex points which have a real part and the imaginary part. The FFT decomposes an N point time domain signal into N time domain signals each consists of a single point. After the N frequency spectra corresponding to these N time domain signals is calculated. Lastly, the N spectra are synthesized into a single frequency spectrum.

Methodology

Audio Recording Process and App Features

In this android app development, “AudioRecord”, “AudioFormat” and “MediaRecorder” classes are used to record the audio signal [5,6]. These classes are used to create an “AudioRecord” object to be used as the recorder. This is the place that the sampling rate, resolution, and the channel configuration are defined. Using this recorder, sound is digitized and the relevant calculation is done continuously while the calculated data are plotted using the “GraphView” library [7]. In this library GraphView class is used to plot a series on two axes. The developed app can produce real time waveform and the real time frequency spectrum of the input signal. Both graphs have some special features. Graph can be start or stop at any time. The y axis is automatically adjusted for different y values. The user can obtain the amplitude value by touching the desired location of the graph. Switch between two graphs is possible. The developed app also has the capability to access the GPS of the smartphone as well. That feature is useful when taking sound level at a particular location by the user. The date and time is displayed in the format of “yyyy, mm, dd, hh:mm:ss”. Flow chart of the waveform and the frequency spectrum process is shown in Figure 1 below. In waveform, rms amplitudes are calculated and in frequency spectrum, FFT coefficients are calculated.

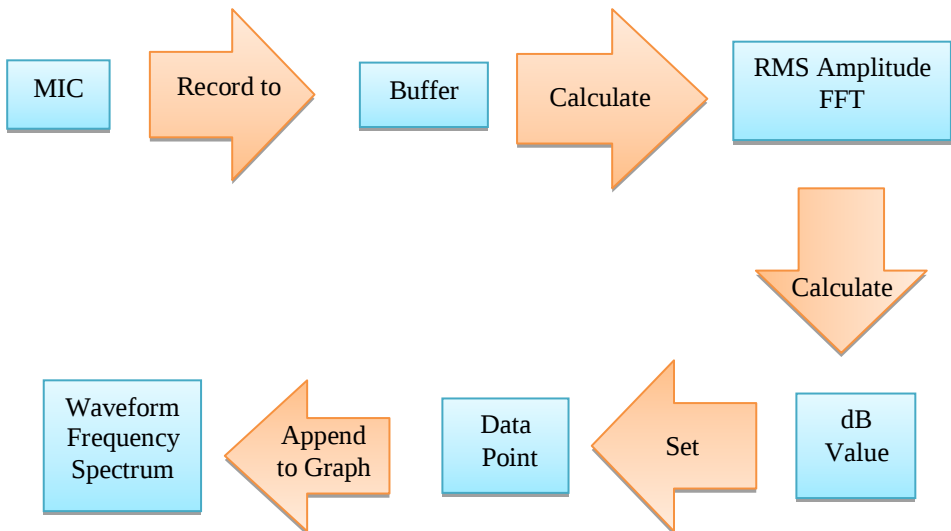


Figure 2: Flow chart of the waveform and frequency spectrum plotting process

The waveform graph plots amplitude (signal level) vs time. The graph is scrollable and scalable for the ease of use. Initially the sound values are reading into a buffer sized 256 with the sampling rate of 8000 Hz and the resolution of 16 bits. The RMS amplitude [8] was calculated as follows in equation (1).

$$RMS \text{ Amplitude} = \sqrt{\frac{1}{N} \sum_{n=1}^N |X[n]|^2} \quad \text{--- (1)}$$

Convert to the decibel using log scale.

$$dB \text{ Amplitude} = 20 \log_{10}(RMS \text{ Amplitude}) \quad \text{--- (2)}$$

Decibel scale can be different depending on the reference value. It is a relative unit which is based on the loudness of the sound wave. Here the reference value used as 1.0 value. This has been done because human ear is sensitive to relative differences (ratio between two values) rather than absolute differences [9, 10]. This app displays the maximum Amplitude value in “dB” units as well.

Frequency spectrum plots the amplitude (signal level) against the frequency in Hz. The amplitude scale can be dynamically changed in to decibel scale or the linear scale. To convert the signal values in to the frequency components, the Discrete Fourier Transformation (DFT) techniques can be used [11-13]. DFT translate data from the time domain to the frequency domain. Fast Fourier Transformation (FFT) library was used to get the frequency domain data in this work.

Results

Snap shots of developed app interfaces taken within “Emulator” are listed below. Emulator is a virtual device that comes with the Android Studio which can be run in the development operating system as a real device. Basic functions of the app and the uses of buttons which have been used are also explained. Figure 6-8 show the main interface of a wave graph and the frequency spectrum graph. It has buttons for “start”, “frequency spectrum”, “Location”, “Date and time” and a text view of “Touch graph to see value”. After the “Start” button is pressed, button name will change to “Stop” and graph plotting will be started and it shows the maximum amplitude value.

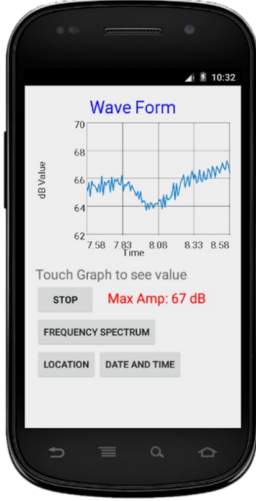


Figure 3: Working Wave graph dB scale graph

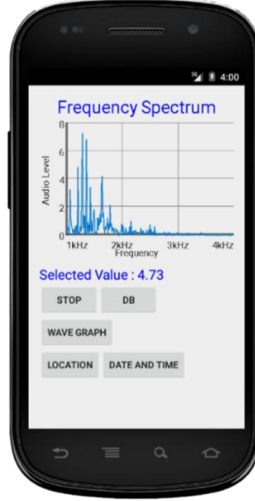


Figure 4: Working Frequency Spectrum linear scale graph

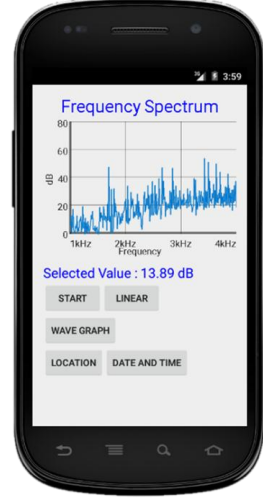


Figure 5: Working "Frequency Spectrum dB scale graph

Comparison with Other Applications

Four sound level meter applications were chosen from the Google Play Store for the purposes of study and compare the results with the developed app. The chosen apps and their associated names are shown in Table 1. Particular sound level meter apps were chosen based on the 3+ rating app on Google Play Store, as of 15-10-2015. Android tablet was held by hand at the same location with their microphone oriented towards the speaker of another android phone and the sound level values were recorded. The pink noise sample was used as the audio clip that was downloading from the web site [14]. Readings were taken in three rounds by changing the volume level of the android phone which use as the source device.

Table 1: Sound Level Meter app comparison

APP Name	dB value		
	Round 1	Round 2	Round 3
Sound Level Meter by EONSOFT	45 dB	58 dB	74 dB
Noise meter by JINASYs	40 dB	55 dB	70 dB
Sound Meter by Smart Tools Co. (ST)	51 dB	65 dB	81 dB
Sound Meter (Basic App)	40 dB	56 dB	70 dB
Audio Spectrum Analyzer*	41 dB	56 dB	71 dB

*app developed in this work

Round 1, Round 2, and Round 3, represent the ascending order of the volume level in the speaker. Sound Meter (ST) gives much higher value compared to others. Table shows average value of the results. It can be vary +/- 1dB range for this chosen noise clip. When comparing the results, this Audio Spectrum Analyzer app which is developed in this work gives the value approximately equal to most of the chosen apps in the Google play store.

Discussion

Android operating system has been chosen because it has a high flexibility and supports many types of smartphone categories, therefore the application can be explored widely. Analog signal data are captured by the microphone in smartphone which is controlled by the Android operating system. Performance of the operating system of Android is depending on both embedded hardware and embedded software [15]. Due to hardware limitations of the phone, 8 kHz sampling rate was used in this tests rather than 44.1 kHz. An 8 kHz of sampling frequency is sufficient for effective digitization of the voice signal with maximum frequency of 4 kHz, because speech transmission ranges from approximately 300 Hz to 3400 Hz [16]. 16 bits are used for each sample in high quality sound, therefore there are 2^{16} different levels of values and used a single channel. The developed app can be used either as a Sound level meter or a Frequency Spectrum Analyzer. This is much important due to noise induced hearing loss of exposing a loud sound. The smartphone based sound level meter and the audio frequency spectrum

analyzer can be a good substitute for high end expensive equipment's for sound analysis. For a better performance of a smartphone base sound analyzer, external USB microphone can be used. The wave form of the developed app displays real time noise exposure data of the environment and we can do a survey of a selected area as the GPS, date and time data embedded to the system. These kinds of apps are growing among audio enthusiasts, educators, acoustic and environmental researchers, and the general public [17]. As the future works we hope to upgrade the app for sampling rate 44.1 kHz and intend to include a Sonogram (Energy vs Frequency) graph as well.

References

1. M. Butler, *Android: Changing the Mobile Landscape*, Pervasive Computing IEEE, 10, (2011), ISSN 1536-1268, DOI 10.1109/MPRV.2011.1
2. W. H. Press, B. P. Flannery, S. A. Teukolsky, W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Edition, Cambridge University Press, 978-0521431088
3. M. Nahvi, *Signals and Systems*, Indian Edition, McGraw Hill Education, 978-93-392-2225-3
4. Digitalversus.com (2013) Sound Quality on Smartphones: Beyond the Advertising, [online] Available at: <http://www.digitalversus.com/audio/sound-quality-smartphones-beyond-advertising-a1746.html> [Accessed 2 May 2015]
5. Developer.android.com, Download Android Studio and SDK Tools | Android Developers. [online] Available at: <http://developer.android.com/sdk/index.html> [Accessed 2 May 2015]
6. Developer.android.com, Package Index | Android Developers. [online] Available at: <http://developer.android.com/reference/packages.html> [Accessed 7 May 2015]
7. S. Thannikkunnath, (2014). graph view android | Tutorials on mobile app development.. [online] Appdevtuts.wordpress.com. Available at: <https://appdevtuts.wordpress.com/tag/graph-view-android/> [Accessed 13 Oct. 2015]
8. Wikipedia, (2015). Root mean square. [online] Available at: https://en.wikipedia.org/wiki/Root_mean_square [Accessed 15 Oct. 2015].

9. H. Quené, and C. Welker, (2006). *amplitude_and_intensity_level*. [online] Hum.uu.nl. Available at: http://www.hum.uu.nl/uilots/lab/courseware/phonetics/basics_of_acoustics_1/amplitude_and_intensity_level.html [Accessed 10 Aug. 2015]
10. Puckette, M. (2006). Units of Amplitude. [online] Msp.ucsd.edu. Available at: <http://msp.ucsd.edu/techniques/v0.11/book-html/node9.html> [Accessed 5 Aug. 2015]
11. Grepcode.com, RealDoubleFFT - ca.uol.aig.fftpack.RealDoubleFFT (.java) - GrepCode Class Source. [online] Available at: <http://grepcode.com/file/repo1.maven.org/maven2/me.franfernandez/fftpack/0.1/ca/uol/aig/fftpack/RealDoubleFFT.java> [Accessed 23 Jun. 2015]
12. Androidcookbook.info, (2015). Visualizing Frequencies - Android Media - Android Developer Blog. [online] Available at: <http://www.androidcookbook.info/android-media/visualizing-frequencies.html> [Accessed 11 Oct. 2015]
13. Smyth, N. (2015). [online] Available at: http://www.ebookfrenzy.com/pdf_previews/AndroidStudioEssentialsPreview.pdf [Accessed 18 Jul. 2015]
14. JetCityOrange@gmail.com, (2015). Download Free Pink Noise MP3 File. [online] Jetcityorange.com. Available at: <http://jetcityorange.com/pink-noise/> [Accessed 15 Oct. 2015]
15. P. Purwacandra and F. Wibowo, *An android-based uncertainty digital audio representation for frequency analyzer*, IEEE, 2nd International Conference on Uncertainty Reasoning and Knowledge Engineering, 95-98, (2012), DOI: 10.1109/URKE.2012.6319594
16. Wikipedia, (2015). Voice frequency. [online] Available at: https://en.wikipedia.org/wiki/Voice_frequency [Accessed 26 Oct. 2015].
17. C. Kardous and P. Shaw, *Evaluation of smartphone sound measurement applications*. The Journal of the Acoustical Society of America, 135(4), EL186-EL192, (2014), DOI: [10.1121/1.4865269](https://doi.org/10.1121/1.4865269).