1. Introduction

This paper provides documentation for an Android application that is used in a project managed by FIU scholars. The project aims to detect and localize jamming signals by collecting signal strengths and some additional features such as time and location in a crowd-source based manner. Android users who have a HackRF SDR provide these values from different places along with different frequency spectrums by using the app RF Spectrum Analyzer.

2. Android Application Development Process

2.1 Research on HackRF SDR

HackRF One, which is designed by Michael Ossman is an open source SDR (software defined radio) platform that is able to transmit or receive any radio signal from 1 MHz to 6GHz. HackRF is a relatively inexpensive SDR transceiver, which can span the frequency range 10 MHz - 6 GHz, and has a sampling rate between 8-20 Msps. This hardware platform is USB-powered so that it could be easily carried along an Android phone (e.g., in a car or bag) as seen in Fig. 1.

Figure 1 HackRF SDR attached to an Android device

2.2 About RFAnalyzer Application

RFAnalyzer is an open source android application, which is available on Google Play, created by Dennis Mantz that interfaces the HackRF and some other SDR devices and plots the FFT by providing a good user interface components. In order to collect radio signals, we modify this open source project by putting some extra features into it such as optional band scanning, database synchronization and GPS tracker.

2.3 Design of the Application

Radio signals received by HackRF are sampled in HackRFSource class. Scheduler class continuously reads samples from the source. Then it sends these samples in packets to SignalProcessing class by inserting them in a queue. If the queue is full, the samples are thrown away not to block the input device.

All the signal processing regarding the FFT is done in SignalProcessing class. This runs as a separate thread. It grabs some new samples form the a pool, then process them by finding the magnitude of the signals with the following calculation; $\log_{10}$ (sqrt(re^2 + im^2)).

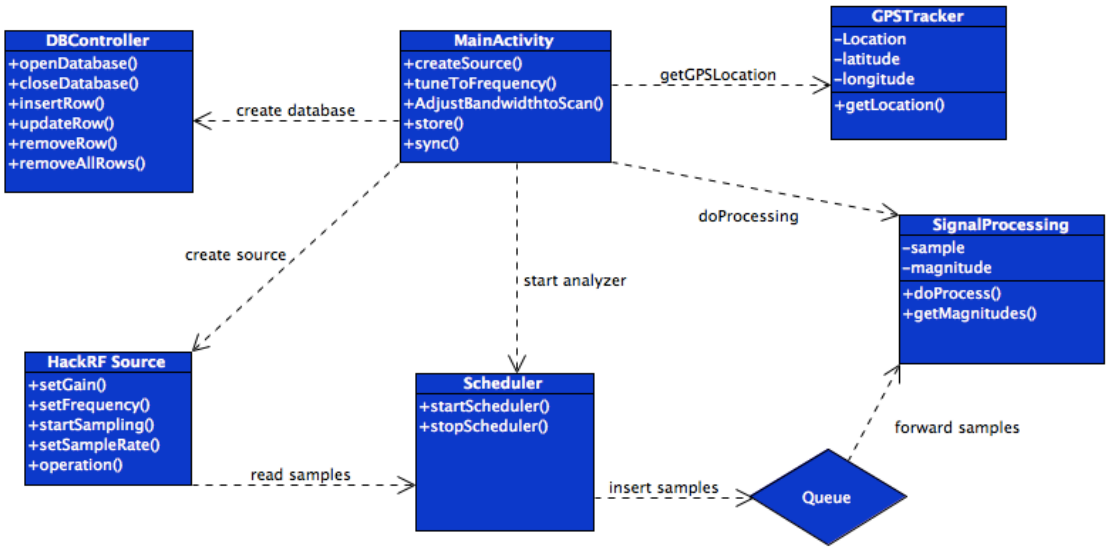


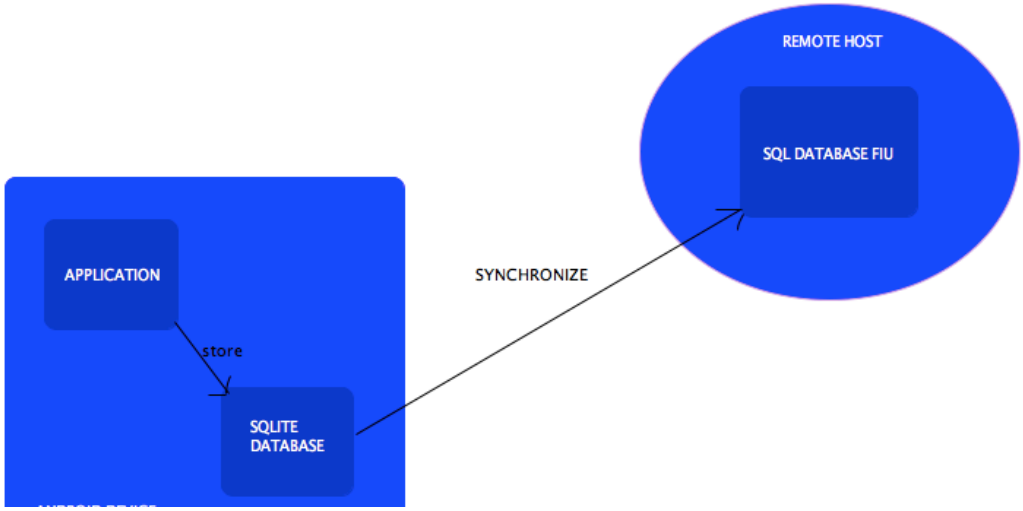
Fig. 3: Class diagram showing the internal structure of the Android application

Regarding the FFT size, the array is initialized and calculated signal strength values are kept in this array. For instance, if the FFT size is set as 16 and the bandwidth is 20MHz, this means that we have 16 points that divide the band equally and the distance between each point will be 1.25MHz (20MHz/16).

In this case, when we want to get the magnitudes of each frequency, we should make some tradeoffs between time and resolution. If we want more accuracy or more smoothness, we should increase the FFT size so that we will have a chance to look at more signal strength values associated with each frequency in a specific bandwidth.

2.4 Database Management

In order to keep values in remote server located in FIU, we use the architecture shown in figure 4. In this architecture, data first gets stored in SQLite database, which is the local database provided by Android devices. It then gets synchronized with remote MySQL server in FIU.



When the user presses the record button, the application starts storing records into local Android SQLite database. We have a single table here called “RFTable” having 6 attributes, which are frequency, signal strength, time, date latitude and longitude. DBController class as had seen in figure 3 handles this process.

In the synchronization part, after data gets stored in SQLite database, the program converts it into JSON (Java Script Object Notation) format. These JSON data then is posted to PHP class using AsyncHttpClient library. PHP class decodes JSON into a PHP array and it inserts data into MySQL database. PHP class then returns back to Android app showing the insertion status of the data.

Data can only be synchronized when the user is connected to FIU network and once it is synchronized up, it is removed from the local Android database.

2.5 Testing and Debugging

In one of the test scenarios, the program scans the 150MHz bandwidth and stores the data with 1.25 MHz intervals in 1.6 seconds. In other words, the duration that the application records 120 tuples with 6 different attributes is about one and half seconds.

Additionally, We realized that a location service provided by android app does not work efficient as we expected. Some performance optimizations should be made in Android GPS system to get more accurate results.

2.6 Release the app

The current repository of the application and the .apk file can be found at the following website: https://github.com/omercelik-cs/FIUProject. You can directly install the program to your Android device by downloading RFAnalyzer.apk file. Before using the app please read the User Manual that can be found in the same website.