WIRELESS INTERCEPTION AND SNIFFING VIA SDR

Authors

Capt. Waqar ul haq

Capt. Hassan Zahid

Capt. Muhammad Umair

Capt. Muhammad Umair

Regn Number

241048

241051

241063

241045

Supervisor

DR. SHIBLI NISAR

Presented to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology,
in partial implementation for the obligations of B.E Degree in Electrical Engineering

(June), 2021

CERTIFICATE OF CORRECTIONS & APPROVAL

Verified that effort covered in this paper titled **“Wireless Interception and Sniffing Via SDR**”, conducted by **Capt. Waqar Ul Haq, Capt. Hassan Zahid, Capt. Muhammad Umair, Capt. Muhammad Umair** under the guidance of **Dr. Shibli Nisar** for partial implementation of Degree of Bachelor of Electrical Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad for the duration of academic year 2020-2021 is appropriate and accepted. The information that has been used from other sources has been suitably confessed / referred.

Approved by

Supervisor
Asst Prof Shibli Nisar, PhD

Date: 12-06-2021

DECLARATION

No part of effort submitted in this paper has been presented in endorsement of another award or qualification in either this institution or somewhere else.

Signature of Team Leader

Capt. Waqar Ul Haq
Registration Number

241048

Plagiarism Certificate (Turnitin Report)

This paper has been verified for plagiarism and piracy. Turnitin report, validated by Supervisor, is attached.

Signature of Team Leader

Capt. Waqar Ul Haq

Registration Number

241048

Signature of Supervisor

Asst Prof Shibli Nisar, PhD

.

Acknowledgements

الْحَمْدُ للَّهِ رَبِّ الْعَلَمِينَ

We are thankful to Allahﷻ, as He is the one who directed us in completion of this effort at each phase. He is the one who blessed us with this idea and made possible for us to come up with new ideas and process new thoughts to complete this project.

We are plentifully grateful to our dear parents who raised us and made us capable to walkthrough difficulties and hurdles of life with ease. There is no payback for the efforts they did for us and pain they bared for our cause.

We would also like to express gratitude to our supervisor **Dr Shibli Nisar** for his support throughout our paper.

We would also like to pay exclusive thanks to **Lt Col Hasnat Khurshid** for his enormous assistance and support. Without his help we would not have been able to complete our project.

Devoted to our remarkable parents, much-loved sisters, supportive brothers and encouraging wives whose incredible assistance and support led us to this brilliant accomplishment.

Abstract

Electronic Warfare is one of the major deciding factors in success of conventional and unconventional warfare. As RF technology is advancing with a very high pace, the conventional Electronic Warfare equipment are getting obsolete along with. Pakistan Army has indoctrinated a large quantity of Electronic Warfare equipment and still more is required to compete current requirement, owing to high tension scenario at borders and operational areas. These high-cost equipment are prone to frequent upgradation and maintenance as well as they are highly resource dependent. Considering above mentioned issues related to the EW equipment that Pak Army is using, they are seldom utilized for difficult terrains and border areas and are to be kept as reserve for conventional warfare. To aid heavily strained sector of EW, we utilized Software Defined Radios (SDR) to form a small portable detachment that can analyze RF spectrum, demodulate, decode and Identify Radio Sets being used in the vicinity of operator. These detachments being highly mobile and very less resource demanding, can be moved to any location for operation. We made considerable efforts in demodulation of non-encrypted radio channels and RF finger printing of radio by analyzing their frequency spectral density and frequency-time graphs. As Software Defined Radios are entirely computer dependent, the possibility of obsoletion of equipment is considerably reduced. New protocols can be programmed using opensource software and they can be implemented by SDR connected to computer via USB. All the computation and Digital Signal Processing (DSP) is to be handled by computer attached to SDR, hence performance of SDR is directly dependent on processing power of computer they are attached with, which is very cheap in comparison to traditional EW equipment upgradation. The RF fingerprinting is an important enhancement to EW sector, as this technology was not available to Pak Army, and we proudly have for the first time provided possible methodology and promising results.

Key Words: *Electronic Warfare, Software Defined Radios, RF fingerprinting.*

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

The research work in this dissertation has been presented in three main parts. First part is related to the wireless spectrum monitoring and investigation of unknown signal frequencies. The objective of this part is to formulate possible methods for wireless spectrum monitoring and interception of any unencrypted analogue and digitally modulated signal. The second part includes jamming and other wireless attacks that can be launched to signal of interest. Finally, third part revolves around signal identification and fingerprinting of radio sets by various techniques that can be adopted using SDR.

1.1 Background, Scope and Motivation

Software Defined Radios are highly dynamic radio sets that can perform any digital signal processing via computer attached to them with USB or ethernet interface. Since the signal processing is done using a computer software, this gives the fluidity to software defined radios in comparison to traditional hardware radios in which signal processing is usually done using analogue elements and circuitry. A traditional hardware RF device can fulfill a single purpose for which it was manufactured, for example a Wi-Fi modem cannot demodulate FM radio signals and same applies for an FM radio. But a single circuitry of SDR can perform both functions because the signal processing depends only on software and is independent of hardware. The computational power, being cheaper than complex hardware radios provide very cost-effective alternatives to its counterparts[1].

This feature or dynamicity in Software Defined Radios motivated us to research methods that could aid EW, particularly jamming and interception. As only software is required to change its function, hence a single SDR and a laptop is enough to provide all functions that were required. Moreover, new methods of encoding and modulating data over wireless signal can be adopted easily by few amendments in software.

The SDR which we have chosen for our project is HackRF One which is a complete opensource equipment and is available in market easily for a very low cost. It has ability to transmit as well as receive signals and is half duplex in nature.

1.2 SDR and HackRF One

Concept of SDR is briefly explained above. In this paragraph I would like to explain conceptual working methodology of SDR, which will further improve overall understanding of this equipment. An ideal SDR converts desired signal into bits of information and feeds it to computer for further processing, the computer is mainly responsible for all digital signal processing. Responsibility of SDR lies with catching, pre-filtering, pre-amplifying, and converting an analogue signal into digital signal and then packing digital signals into predefined bits. After this, it sends these signals to computer via USB interface[2].

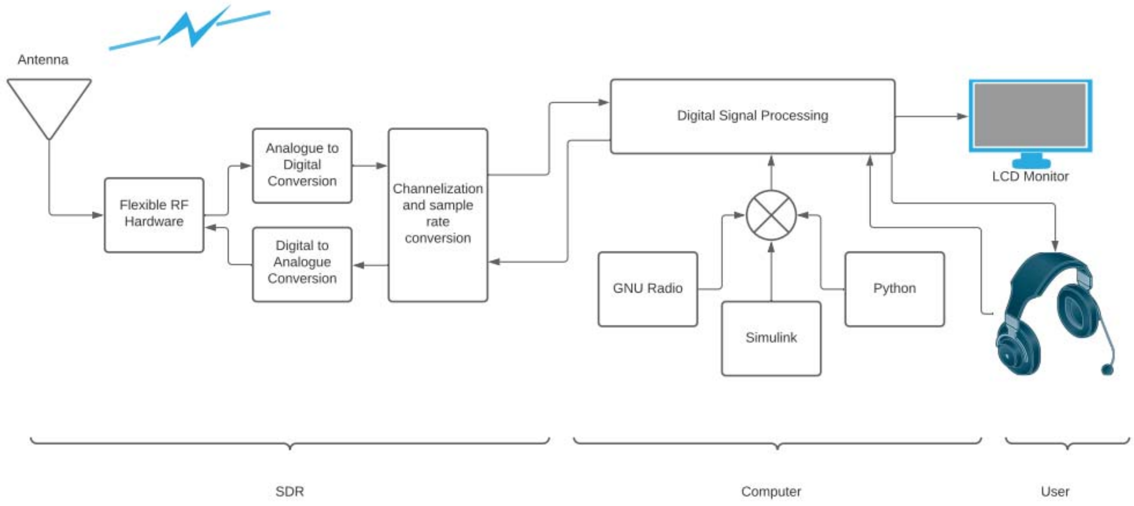

Antenna
Flexible RF Hardware
Analogue to Digital Conversion
Digital to Analogue Conversion
Channelization and sample rate conversion
Digital Signal Processing
LCD Monitor
GNU Radio
Simulink
Python
SDR
Computer
User

Figure 1.2.1.1 SDR workflow

HackRF One is an opensource SDR designed and developed by Great Scott Gadgets which can transmit and receive radio signals from 1 Mega Hertz upto 6 Giga Hertz. It is specifically developed to provide testing and development platform for next generation radio technologies. It is to be connected to computer via USB interface.

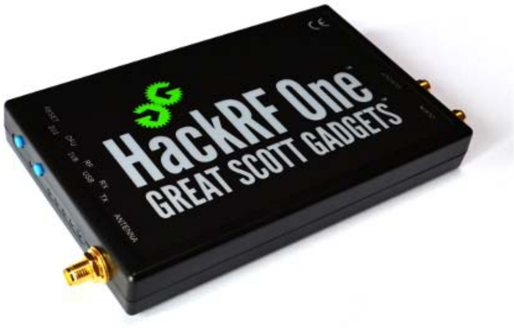


Figure 1.1.2.2 HackRF One

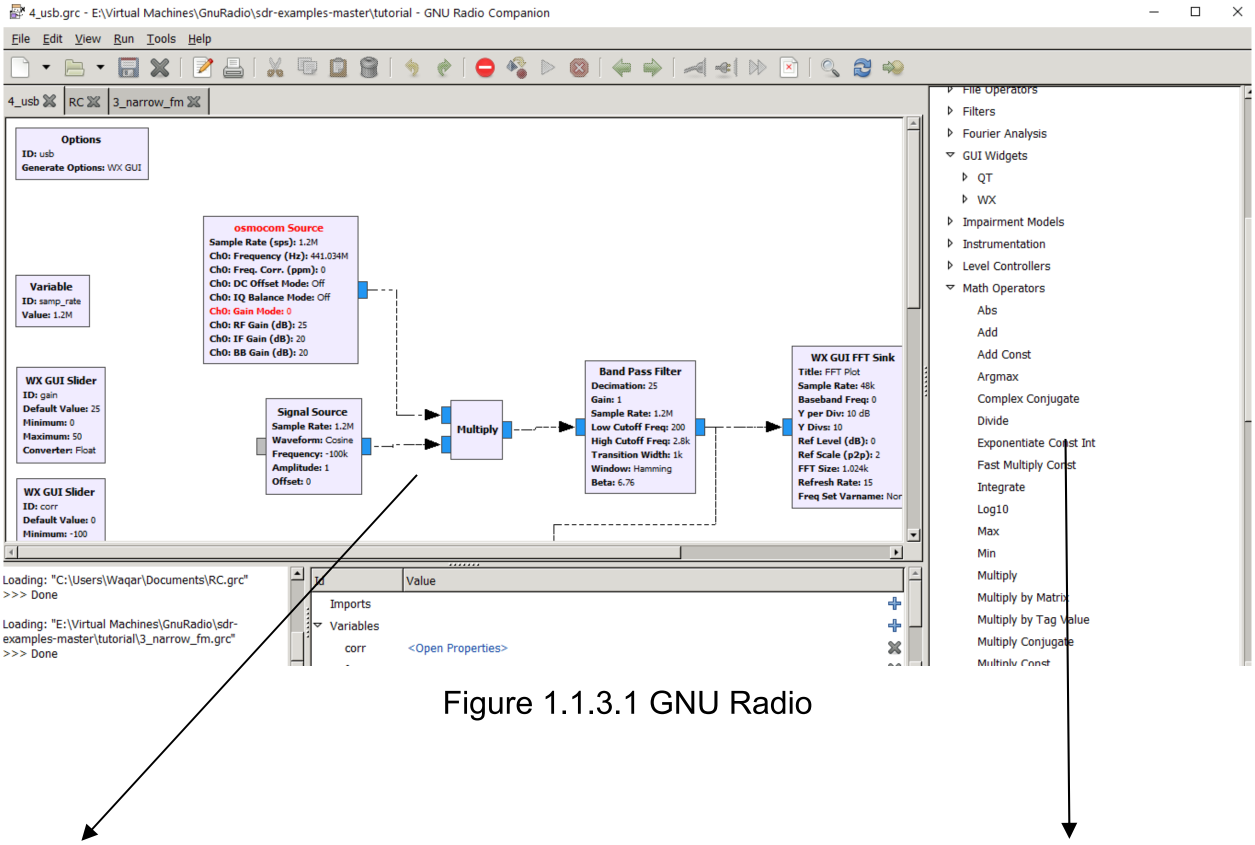
Following table explains important features of HackRF One SDR[4]

Table 1.1 HackRF One features

Operating Frequency	1Mhz – 6GHz
Mode of Operation	Half Duplex
Samples per second	20 million
Quadrature Samples	8-bit
Compatibility	GNU Radio, SDR#, GQRX
Antenna Port Power	50mA at 3.3V (Software Controlled)
Hardware	Open Source

1.3 GNU Radio

GNU Radio is a software system that provides graphical interfaced signal processing blocks to implement software defined radios. It is open-source programme that provides python based codes in the form of blocks specifically for signal processing. It works with python as Simulink works with Matlab. It is important to introduce functions of some commonly used blocks to provide better understanding of its ability and its importance in respect to SDR[3].



4_usb.grc - E:\Virtual Machines\GnuRadio\sdr-examples-master\tutorial - GNU Radio Companion

Options
ID: usb
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 1.2M

WX GUI Slider
ID: gain
Default Value: 25
Minimum: 0
Maximum: 50
Converter: Float

WX GUI Slider
ID: corr
Default Value: 0
Minimum: -100

osmocom Source
Sample Rate (sps): 1.2M
Ch0: Frequency (Hz): 441.034M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: 0
Ch0: RF Gain (dB): 25
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Signal Source
Sample Rate: 1.2M
Waveform: Cosine
Frequency: -100k
Amplitude: 1
Offset: 0

Multiply

Band Pass Filter
Decimation: 25
Gain: 1
Sample Rate: 1.2M
Low Cutoff Freq: 200
High Cutoff Freq: 2.8k
Transition Width: 1k
Window: Hamming
Beta: 6.76

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 48k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: Nor

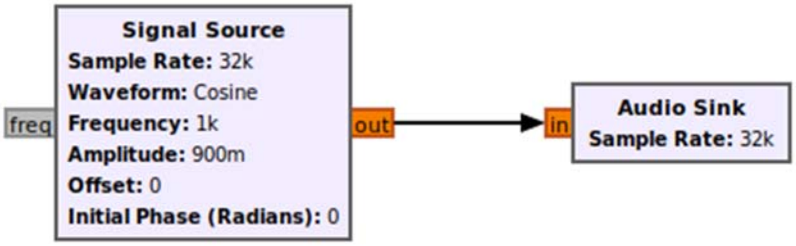
```
Loading: "C:\Users\Waqar\Documents\RC.grc"
>>> Done

Loading: "E:\Virtual Machines\GnuRadio\sdr-
examples-master\tutorial\3_narrow_fm.grc"
>>> Done
```

Id	Value
Imports	
Variables	
corr	<Open Properties>

Figure 1.1.3.1 GNU Radio

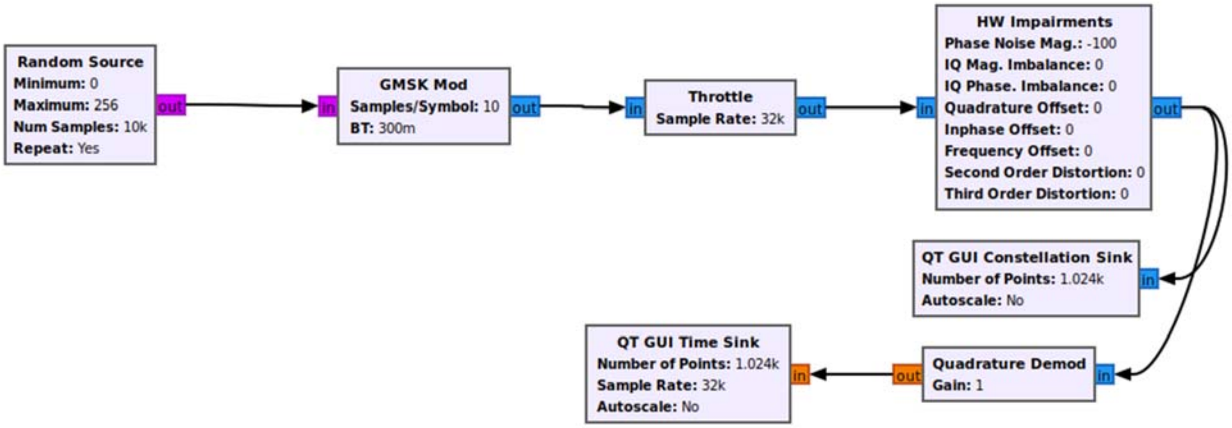
To deal with digital signals, the individual processing stages for example filtering, correction, analysis, detection etcetera as processing blocks, which can be connected using simple flow-indicating arrows:



Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 900m
Offset: 0
Initial Phase (Radians): 0

Audio Sink
Sample Rate: 32k

When building a signal processing application, we have to make a complete graph of blocks and is called as flowgraph in GNU Radio.



Random Source
Minimum: 0
Maximum: 256
Num Samples: 10k
Repeat: Yes

GMSK Mod
Samples/Symbol: 10
BT: 300m

Throttle
Sample Rate: 32k

HW Impairments
Phase Noise Mag.: -100
IQ Mag. Imbalance: 0
IQ Phase. Imbalance: 0
Quadrature Offset: 0
Inphase Offset: 0
Frequency Offset: 0
Second Order Distortion: 0
Third Order Distortion: 0

QT GUI Constellation Sink
Number of Points: 1.024k
Autoscale: No

QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 32k
Autoscale: No

Quadrature Demod
Gain: 1

GNU Radio comes with a large set of existing blocks. An index to all of them can be found in GNU Radio documentation. Few are most commonly used blocks in flowgraphs.

1.3.1 Waveform Generators

1.3.1.1 Constant Source

A constant source of signal (DC)

1.3.1.2 Noise Source

Can produce Active White Gaussian Noise or random noise

1.3.1.3 Signal Source (e.g. Sine, Square, Saw Tooth)

These generate signals of our choice (cosine, square, Saw Tooth etc.) and desired frequency.

1.3.2 Modulators

1.3.2.1 AM Mod/Demod

This block can perform Amplitude Modulation and demodulation with provided sample rate.

1.3.2.2 Continuous Phase Modulation

This block will take data in complex form and modulate this data with a sine wave of provided frequency and mentioned sample rate

1.3.2.3 PSK Mod / Demod

This block will perform phase shift keying modulation and demodulation on provided digital data.

1.3.2.4 GFSK Mod / Demod

Gaussian Frequency Shift Keying is a type of Frequency Shift Keying modulation where signal is passed through a gaussian filter to shape the pulses before modulating which greatly reduces spectral bandwidth and out-of-band spectrum. This is helpful when adjacent channel has high power and there is a chance of interference between channels. This block modulates data into GFSK and demodulates GFSK signal into data.

1.3.2.5 GMSK Mod / Demod

Gaussian Mean Shift Keying Modulation/ Demodulation.

1.3.2.6 QAM Mod / Demod

Quadrature Amplitude Modulation/ Demodulation.

1.3.2.7 WBFM Receive

Wide Band Frequency Modulation / Demodulation.

1.3.2.8 NBFM Receive

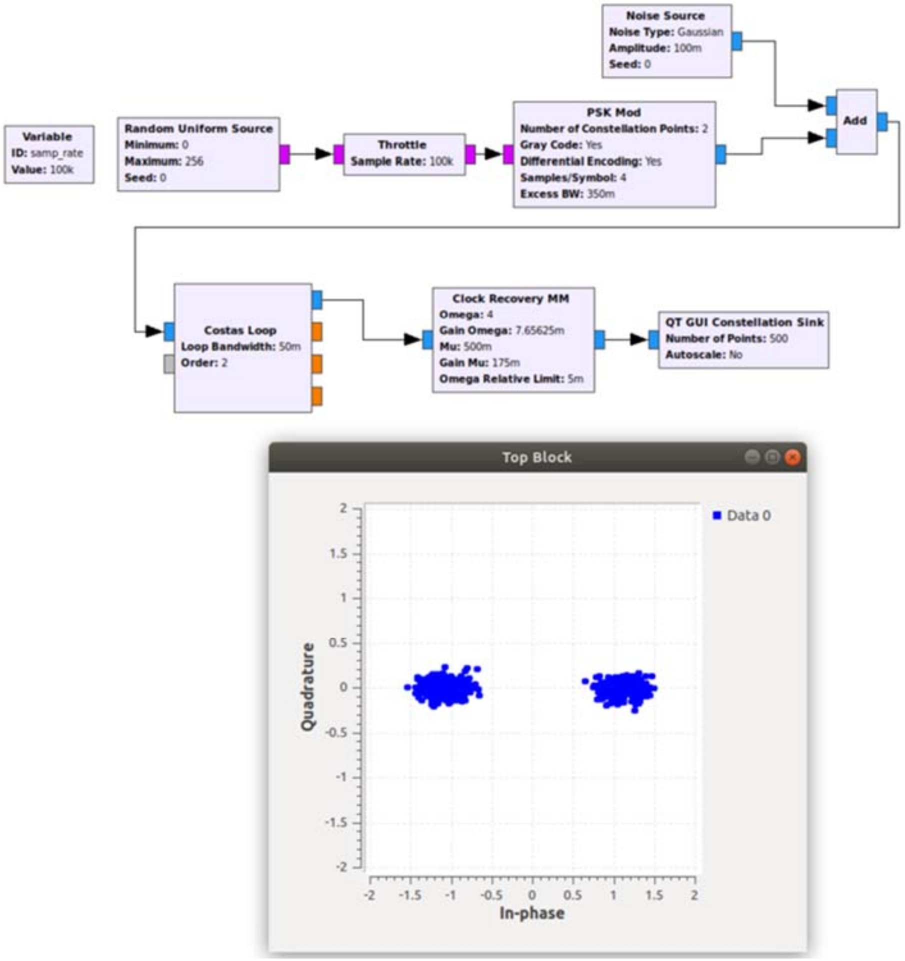
Narrow Band Frequency Modulation / Demodulation.

1.3.3 Instrumentation (i.e., GUIs)

Instrumentation are very useful blocks for analyzing received or transmitted signals visually.

1.3.3.1 Constellation Sink

This visualizes constellation of a modulated signals; it shows amplitude and phase of a digital modulation.



Variable
ID: samp_rate
Value: 100k

Random Uniform Source
Minimum: 0
Maximum: 256
Seed: 0

Throttle
Sample Rate: 100k

PSK Mod
Number of Constellation Points: 2
Gray Code: Yes
Differential Encoding: Yes
Samples/Symbol: 4
Excess BW: 350m

Noise Source
Noise Type: Gaussian
Amplitude: 100m
Seed: 0

Add

Costas Loop
Loop Bandwidth: 50m
Order: 2

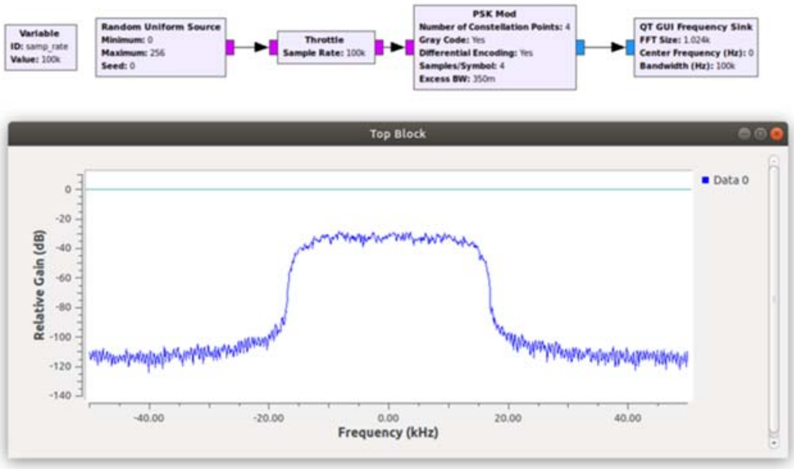
Clock Recovery MM
Omega: 4
Gain Omega: 7.65625m
Mu: 500m
Gain Mu: 175m
Omega Relative Limit: 5m

QT GUI Constellation Sink
Number of Points: 500
Autoscale: No

Figure 1.1.3.2 Example of constellation sink

1.3.3.2 Frequency Sink

This is a graphical sink which displays signals of interest frequency domain. It is a graphical version of FFT.



Variable
ID: samp_rate
Value: 100k

Random Uniform Source
Minimum: 0
Maximum: 256
Seed: 0

Throttle
Sample Rate: 100k

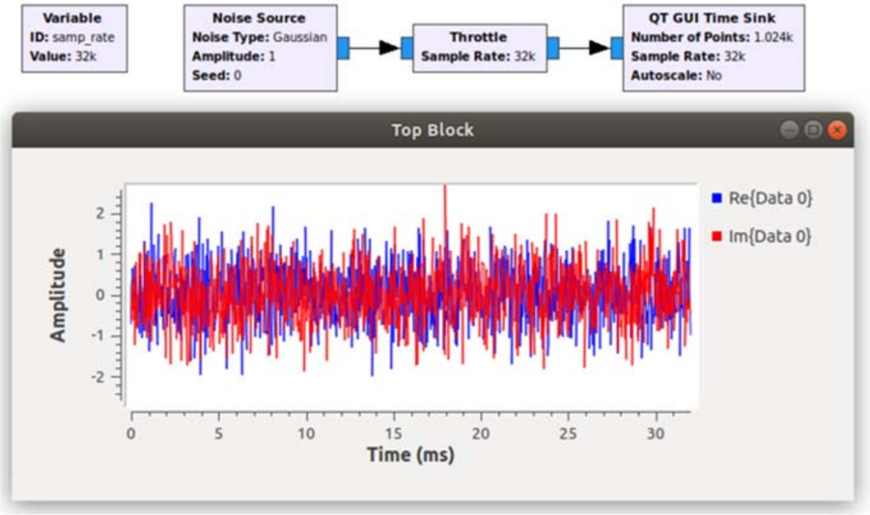
PSK Mod
Number of Constellation Points: 4
Gray Code: Yes
Differential Encoding: Yes
Samples/Symbol: 4
Excess BW: 350m

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 100k

Figure 1.1.3.3 Example of frequency sink

1.3.3.3 Time Sink

It is a graphical interface which displays signal of interest in time domain.



Variable
ID: samp_rate
Value: 32k

Noise Source
Noise Type: Gaussian
Amplitude: 1
Seed: 0

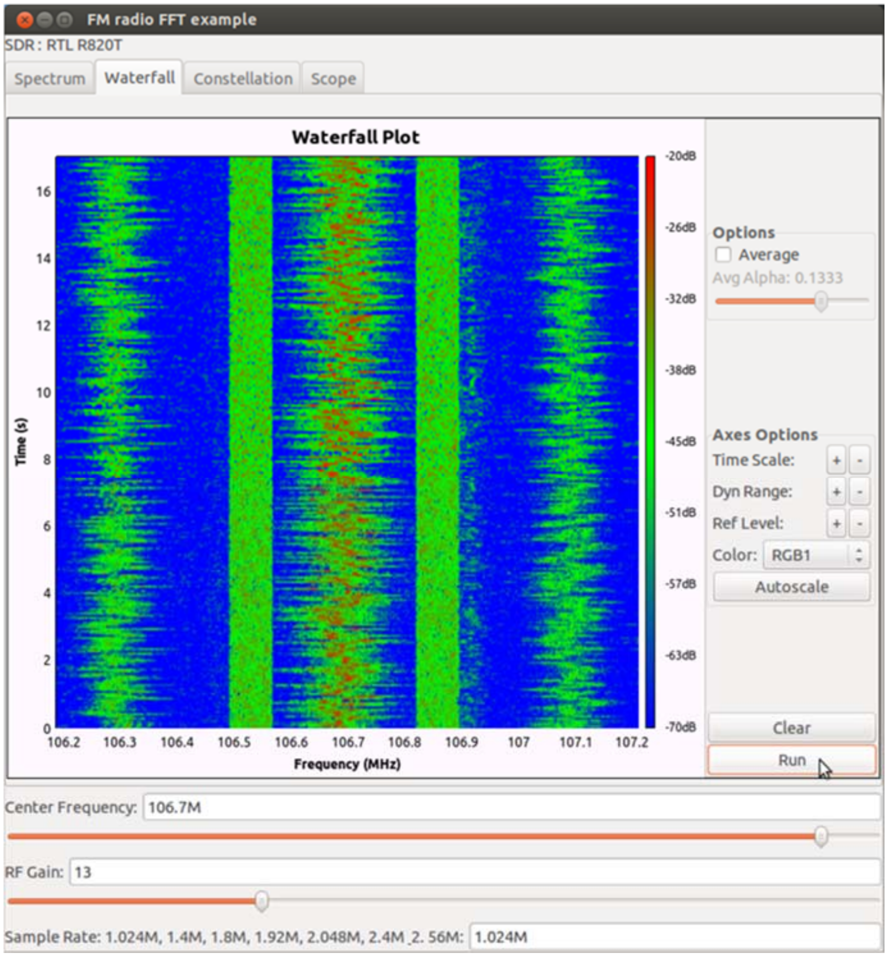
Throttle
Sample Rate: 32k

QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 32k
Autoscale: No

Figure 1.1.3.4 Example of time sink

1.3.3.4 Waterfall Sink

This shows spectral density of signals in the form of color map, we have extensively utilized this feature in our project.



SDR : RTL R820T

Avg Alpha: 0.1333

Color: RGB1

Center Frequency: 106.7M

RF Gain: 13

Sample Rate: 1.024M, 1.4M, 1.8M, 1.92M, 2.048M, 2.4M_2. 56M: 1.024M

Figure 1.1.3.5 Example of waterfall plot

1.3.4 Math Operators

These do basic math operations on the signals. They are self-explanatory.

- **Abs**
- **Add**
- **Complex Conjugate**
- **Divide**
- **Integrate**
- **Log10**
- **Multiply**
- **RMS**

- **Subtract**

1.3.5 Filters

Filters are basic components of DSP and their use is almost compulsory in every signal flowgraph. Few filter blocks provided by GNU Radio are.

- Band Pass / Reject Filter
- Low / High Pass Filter
- IIR Filter
- Generic Filter bank
- Hilbert

Tasks as signal normalization, sync and visualization can be done by making a suitable signal processing flow graph. Just by connecting appropriate block one after other a complete system is made. There is also option of writing own block in python if some logic is not already present is available blocks.

It is important to consider GNU Radio is primarily a structure for the development of signal processing blocks and their interaction. It has inbuilt extensive library of blocks. But it should be borne in mind that GNU Radio itself is not a software that is ready to do something specific. it is the operator's task to make something useful out of it.

1.4 Workflow of project

Having given brief explanation of basic elements composing our project, it is now pertinent to explain working methodology of our project before diving in detailed explanation of each part.

As explained earlier, in aid of already present equipment relate to EW with Pakistan Army, we have made a small detachment that can monitor, intercept, attack and fingerprint wireless spectrum and device. For this we assumed that our detachment will be based on one SDR, Laptop and an operator. For all the tasks mentioned above, we made an appropriate sequence, following that, will make all tasks easily possible to accomplish.

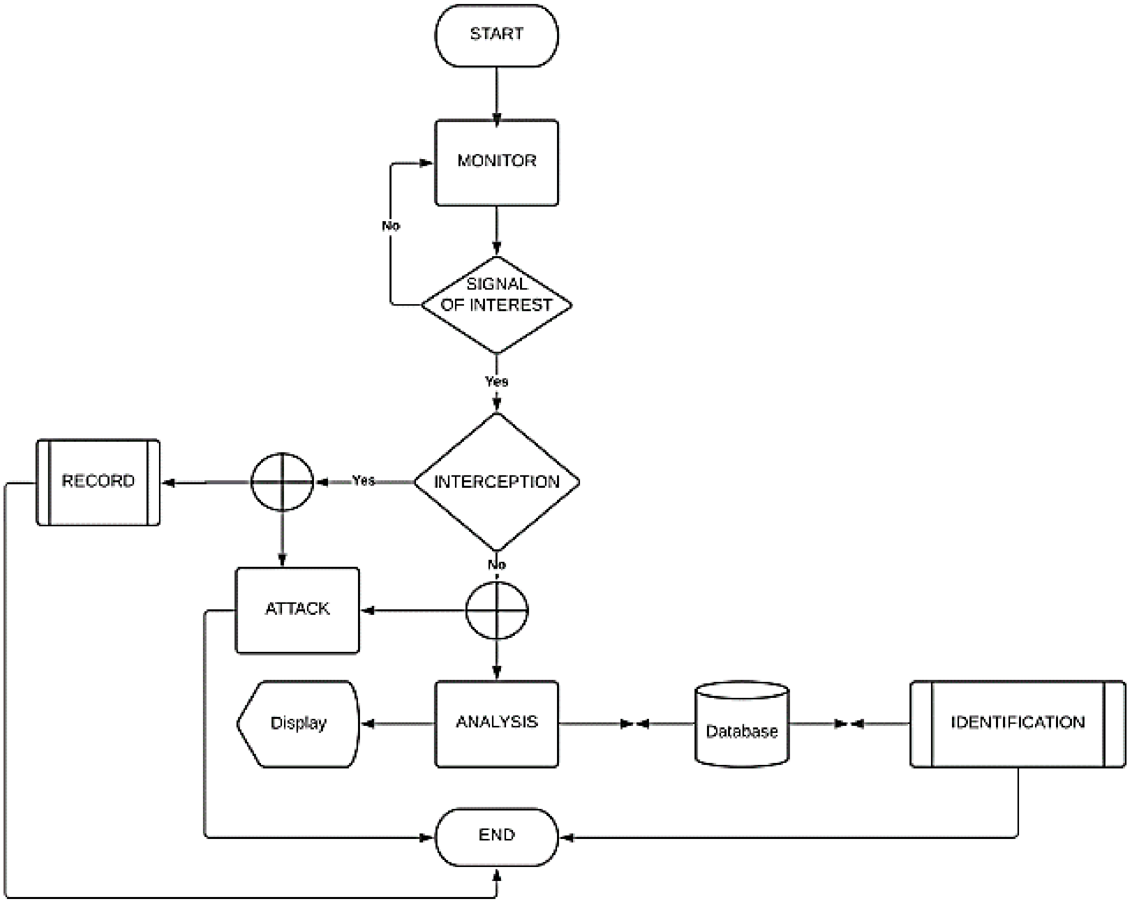


Figure 1.1.4.1 Workflow of project

First the operator will continuously monitor for any new unknown signal being transmitted in area of operation, based on signal characteristics it will be decided by operator that either this is signal of interest or not. If it is signal of interest, operator will try its interception provided the signal is transmitted in plaintext. If signal is encrypted, then it will be saved in database and will be analyzed further in detail. Also, operator will have liberty to attack wireless frequency with jamming or spoofing. We will explain all steps with detail in coming parts.

CHAPTER 2: WIRELESS MONITORING AND INTERCEPTION

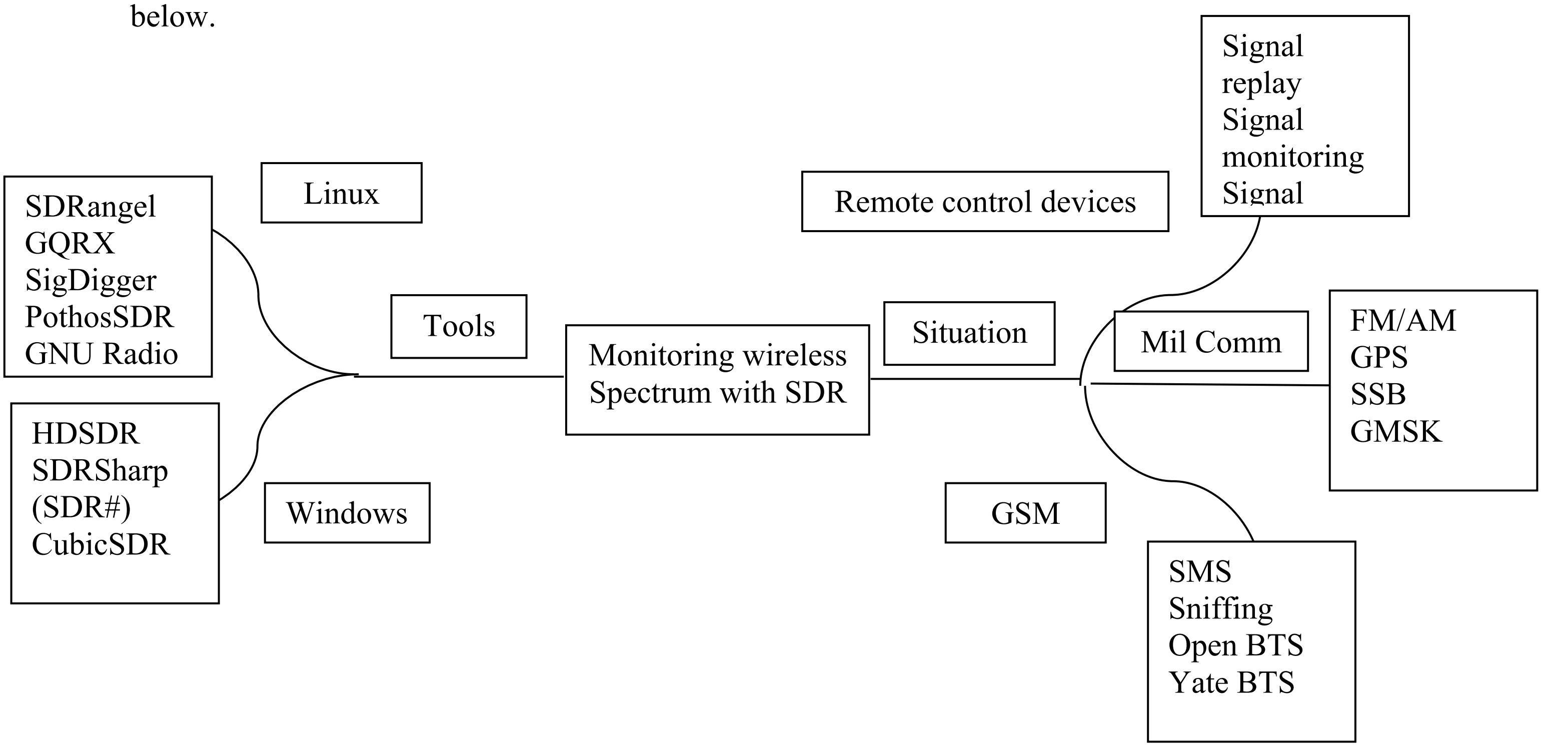
2.1 Monitoring

Most important and time-consuming part is identifying signal of interest. This will be done with constant monitoring wireless spectrum and pointing out any unusual presence of data in spectrum. Monitoring is a continuous procedure, and it requires high degree of alertness and presence of mind while performing this task. Also, the success in monitoring is highly dependent on the experience of operator as he must point out any unusual appearance of signals that might be coming from enemy equipment. There are few techniques which we developed, to aid operator in judging any signal of interest, while doing our research which we will explain in this chapter. Interception comes when operator has successfully identified any signal of interest. If the signal is unencrypted then it can easily be intercepted, provided the modulation is known to the operator. We have resolved this issue of identification of modulation as well, using waterfall graphs.

Wireless spectrum monitoring is done with the help of two techniques. One is Fourier transforming and displaying result in graph of frequency in Hz with respect to magnitude in dBm. This graph confirms at which frequency some form of data is present. Second technique is measuring power spectral density of signals at every frequency. Graphically, this is referred as 'waterfall'. This waterfall confirms presence of signal in frequency as well as gives idea of the type of modulation and encoding being done on the signal, which helps in classification of the signal. Both these techniques can be utilized by making appropriate flowgraph in GNU Radio but taking in consideration the ease of operation for the operator, we have utilized linux based open-source software. For interception, we also have utilized windows and linux based open-source and freeware software. List of software we used is as follow:

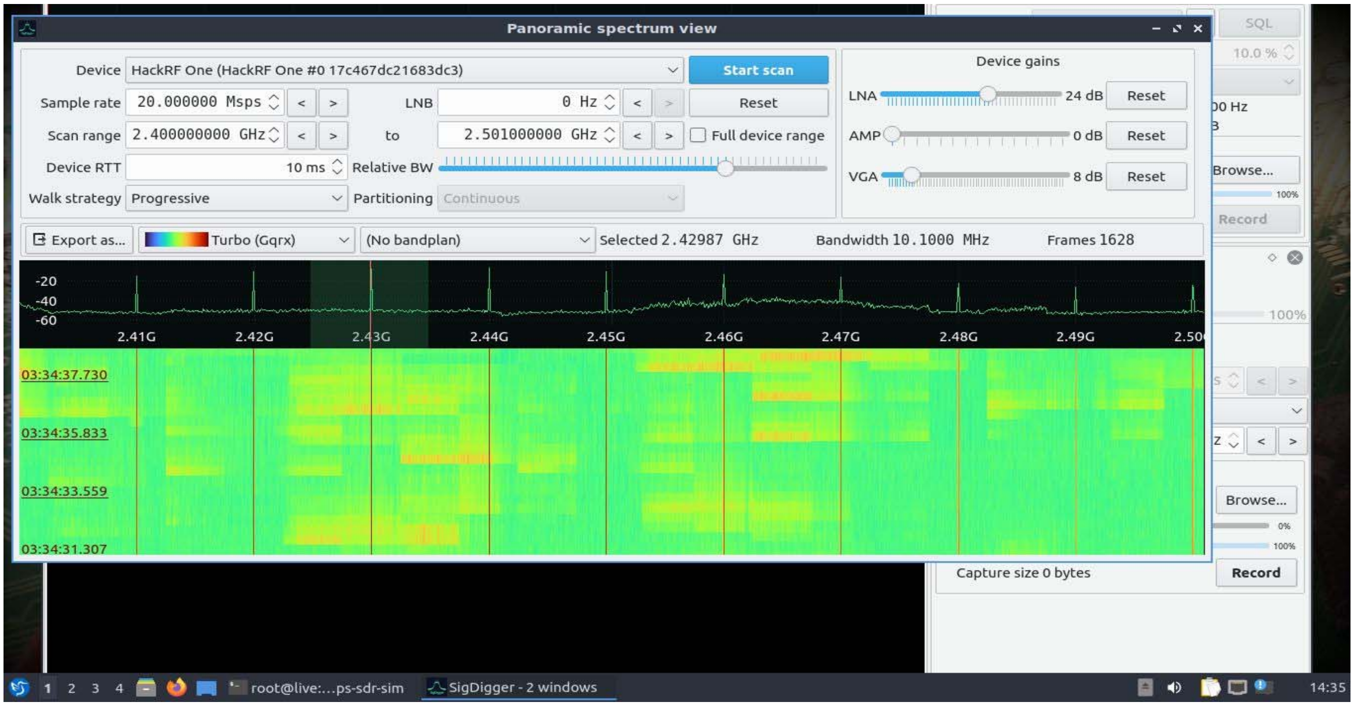
- SDR#
- Sigdigger
- GQRX
- SDR Console V3
- HDSDR

Also, we have made some effort that encapsulates all available opensource software available, their operating system and for what purpose they can fit, in the form of a flowgraph displayed below.



For military based monitoring, the bandwidth of interest is from 3MHz – 30MHz, 110MHz-600MHz and then 2.4MHz to 2.5MHz which covers all HF, UHF, VHF, and microwave data of enemy equipment.

HackRF One has a sampling rate of 20 million this means it can provide a bandwidth for monitoring of 20MHz. thus we have a window of 20MHz for searching signal of interest with a single HackRF One. This gives a great challenge for monitoring a large spectrum because 20MHz window is insufficient to provide efficient monitoring. To counter this problem HackRF One firmware provides facility of sweep, exploiting this feature, we can progressively measure wireless spectrum in discreet windows of 20MHz in desired bandwidth and result is displayed on screen when HackRF One is done scanning. But again, If HackRF One is programmed to measure its complete range, then this significantly reduces its resolution. Instead, through test and trial we have come to conclusion that for effective results, maximum 200MHz bandwidth should be measured at a time. For monitoring, we have utilized the services of “Sigdigger”, which is a Linux based software that provides us facility of panoramic view. In fig 2.1, we have observed wi-fi channels in from 2.4 to 2.5 GHz and concluded that our wi-fi router is using channels of 2.43, 2.46 and 2.47 GHz channels. Also, with careful observation, we further discovered that each channel had a width of 20MHz.


Panoramic spectrum view

Device HackRF One (HackRF One #0 17c467dc21683dc3)

Sample rate 20.000000 Msps

LNB 0 Hz

Scan range 2.400000000 GHz to 2.501000000 GHz

□ Full device range

Device RTT 10 ms

Walk strategy Progressive

Partitioning Continuous

Device gains

LNA 24 dB

AMP 0 dB

VGA 8 dB

Turbo (Gqrx)

(No bandplan)

Selected 2.42987 GHz

Bandwidth 10.1000 MHz

Frames 1628

Capture size 0 bytes

Figure 2.1.1 Monitoring wi-fi channels

We were also able to determine signal power in dBm using SDRConsole V3. This gives an idea of approximate distance of radio set being observed; however, it is inaccurate if we have no clue of its absolute transmission power. We can counter this problem efficiently if two or more SDR are utilized and their distances are kept known, we can observe rate of change of power reducing over time and figure out what at what power it was transmitted using friis transmission equation.

2.2 Identification of signal of interest

Identification, as explained earlier will come from experience and time. When an operator will be sure about occupation of spectrum by friendly forces, any new suspicious signal will be regarded as signal of interest. This means all the subsequent work as explained in workflow diagram (Figure 1.8) will be applied to any suspicious signal that was not already occupied by friendly forces. Type of modulation and encoding can be identified by careful observation of time spectrum and power spectrum density of signal. For example, in amplitude modulation, frequency is same whereas power changes as per data. So, its power spectral density will be same across frequency by will fade completely when data is 0 and appear completely when data is 1.

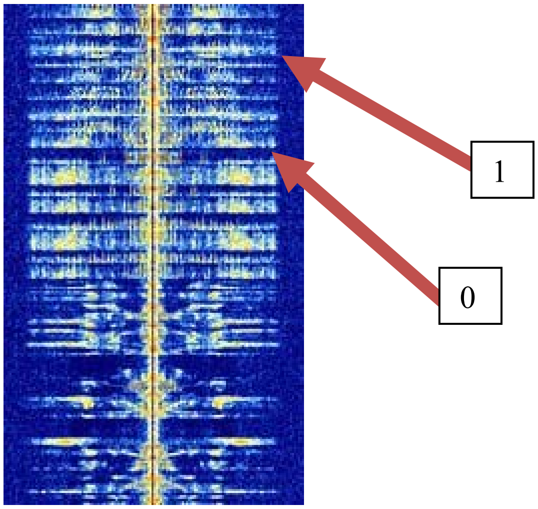
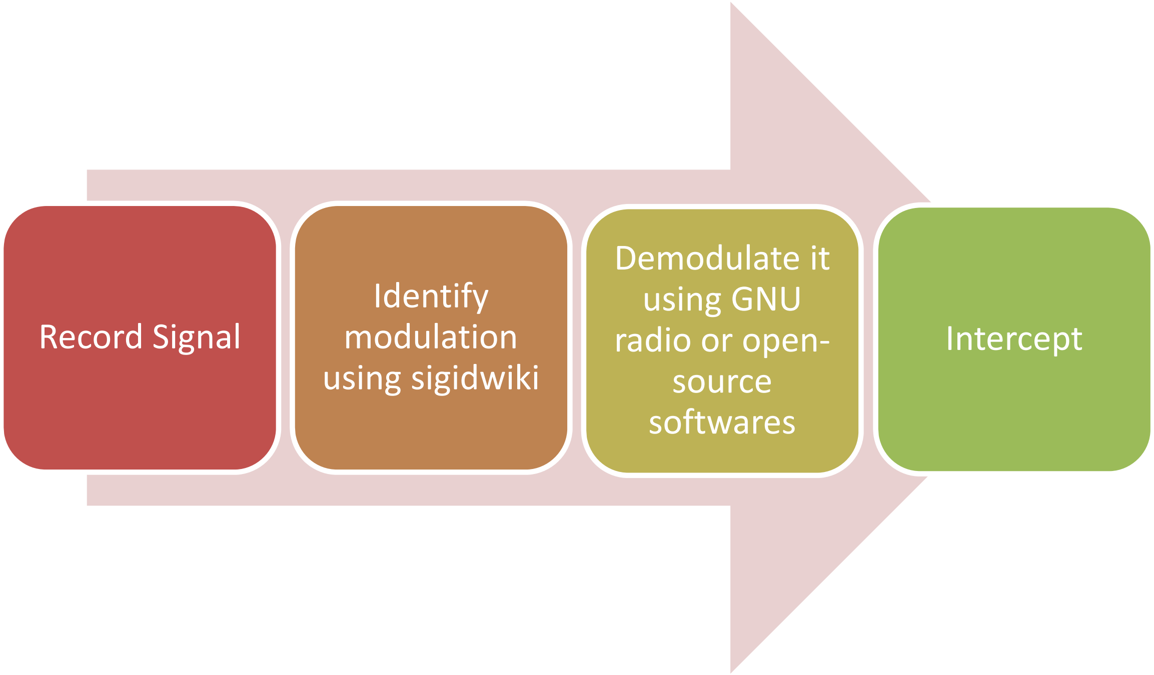


Figure 2.2.1 Amplitude modulation PSD

2.3 Interception

In our project, for interception, we have assumed that data is in plain text when transmitted over the air. However, if encryption can be broken via some known algorithm then interception can also be done to encrypted signals like GSM 2G text messages and calls can be intercepted if they are encrypted by A5/0 and A5/1. Although this is possible, but we have not included such scenario in our project. We have intercepted signals by following technique shown:


Record Signal
Identify modulation using sigidwiki
Demodulate it using GNU radio or open-source softwares
Intercept

Sigidwiki is a huge database with pictures of power spectral density of various signals transmitted by military and non-military appliances. It also explains any possible methods to demodulate and decode signal using different software available over internet. We also have used SDR# for demodulation of analogue and digital radio signals. This software is very easy to use and can be easily operated by a signalman in case of our proposed mobile detachment. So far, we have successfully intercepted HARRIS HF 5800 and commercially available Push to Talk Radio WH 118 and ASELSAN 9611 SDR on plain text.

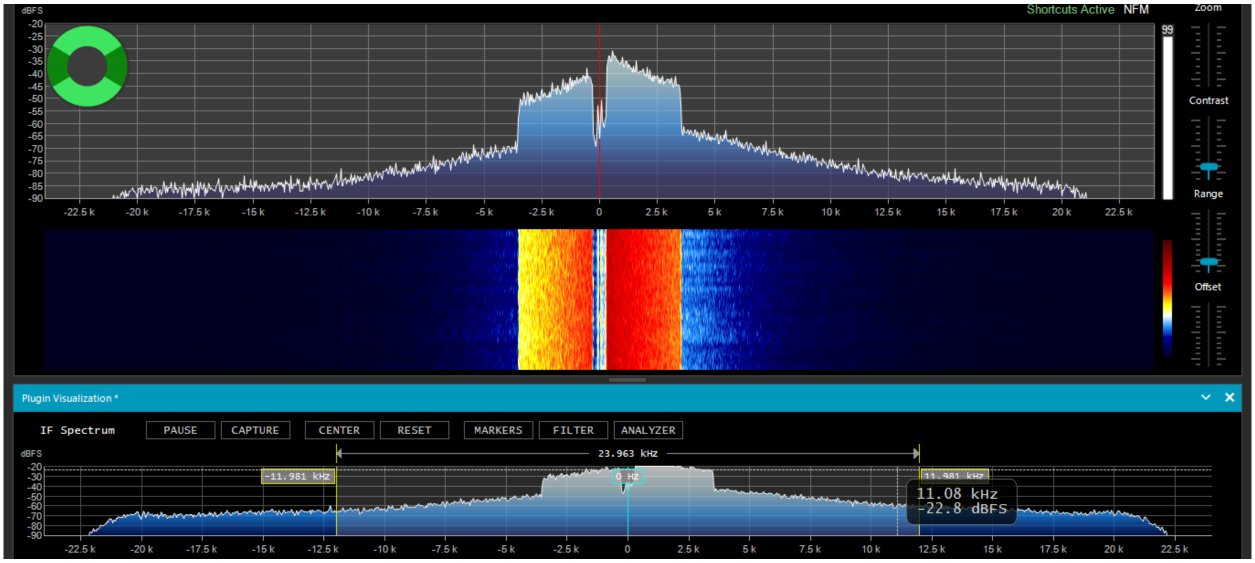


Figure 2.3.1 Interception Using SDR#

CHAPTER 3: ATTACK

3.1 Introduction to wireless attacks

In this chapter, we will explain possible attacks that could be launched to any frequency within the range of HackRF One i.e., from 1MHz to 6GHz. The methods we developed for this project were tested on radio sets and digital appliances as RC car and wireless doorbell. It should be kept in mind that the methods are applicable to any SDR hence the limitation of frequency range will be according to SDR being used and whether that SDR is capable of transmission or not. HackRF One can transmit with power as discussed below:

HackRF One's absolute maximum TX power varies by operating frequency:

a. 10 MHz to 2150 MHz: 5 dBm to 15 dBm, generally increasing as frequency decreases.
b. 2150 MHz to 2750 MHz: 13 dBm to 15 dBm
c. 2750 MHz to 4000 MHz: 0 dBm to 5 dBm, increasing as frequency decreases.
d. 4000 MHz to 6000 MHz: -10 dBm to 0 dBm, generally increasing as frequency decreases.

Through most of the frequency range up to 4 GHz, the maximum TX power is between 0 and 10 dBm. The frequency range with best performance is 2150 MHz to 2750 MHz. But nevertheless this transmission power can be increased using power amplifier which are available in market in price range of Rupees 4000-8000 /-

We have divided our project in three different types of attacks. Foremost is the most used attack i.e., jamming. Second attack is spoofing, in which we do analyze signal and recreate it with data of our own choosing and then spoof recipient appliance, for example GPS spoofing. The third type of attack is signal replay attack in which we capture a signal and replay that signal as it is repeatedly so that receiving device starts behaving abnormally. All three types of attacks will be explained in coming sub-paragraphs.

3.2 Jamming

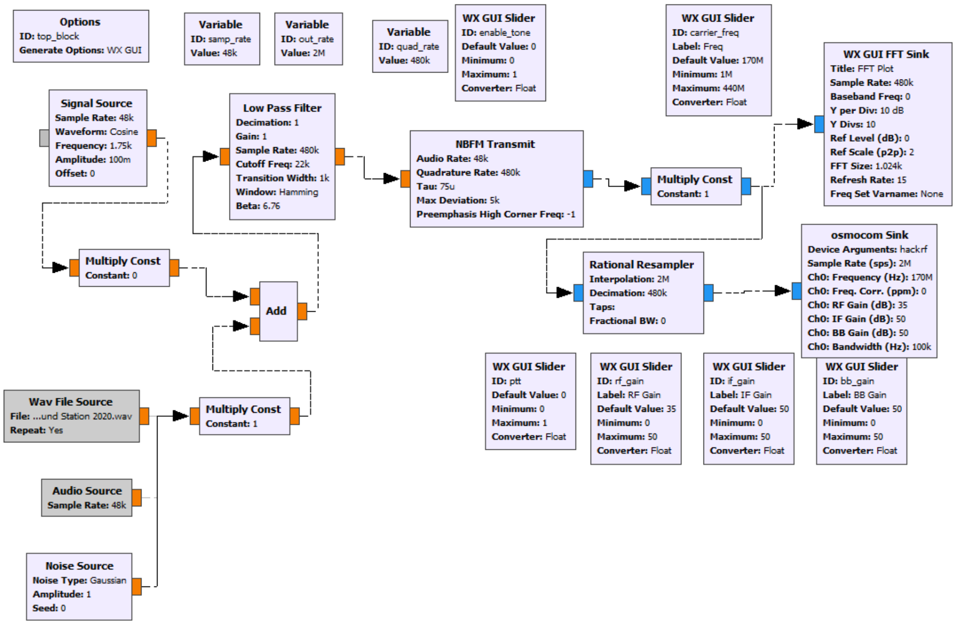
Radio jamming is the deliberate blocking or interference with wireless communications. For a radio jammer to work, it has to decrease signal to noise ratio so that communication is disrupted. The same concept can be applied to digital wireless network for that we have to disrupt information flow.

Jamming is usually distinguished from interference that can occur due to device malfunctions or other accidental circumstances. Devices that simply cause interference are regulated differently[5]. Unintentional "jamming" occurs when an operator transmits on a busy frequency without first checking whether it is in use, or without being able to hear stations using the frequency. Another form of unintentional jamming occurs when equipment accidentally radiates a signal, such as a cable television plant that accidentally emits on an aircraft emergency frequency.

We have divided jamming into two categories:

3.2.1 Deliberate Jamming

In this type of jamming, we are deliberately inserting noise in the frequency to be jammed using SDR via GNU Radio. The noise source block in GNU Radio provides AWGN and Random noise values which can be transmitted directly to frequency, or we can modulate that noise. The modulation will be of target radio set so that when target radio set receives this signal and demodulate it, it will receive noise. This does not decrease signal to noise ratio of channel; hence it can deceive very powerful radio equipment's. all we must bear in mind is our transmission power should be greater than target radio set transmission power so that it cannot transmit to or receive from any neighbouring radio equipment.



Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 48k

Variable
ID: out_rate
Value: 2M

Variable
ID: quad_rate
Value: 480k

WX GUI Slider
ID: enable_tone
Default Value: 0
Minimum: 0
Maximum: 1
Converter: Float

WX GUI Slider
ID: carrier_freq
Label: Freq
Default Value: 170M
Minimum: 1M
Maximum: 440M
Converter: Float

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 480k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Signal Source
Sample Rate: 48k
Waveform: Cosine
Frequency: 1.75k
Amplitude: 100m
Offset: 0

Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 480k
Cutoff Freq: 22k
Transition Width: 1k
Window: Hamming
Beta: 6.76

NBFM Transmit
Audio Rate: 48k
Quadrature Rate: 480k
Tau: 75u
Max Deviation: 5k
Preemphasis High Corner Freq: -1

Multiply Const
Constant: 1

Multiply Const
Constant: 0

Add

Rational Resampler
Interpolation: 2M
Decimation: 480k
Taps:
Fractional BW: 0

osmocom Sink
Device Arguments: hackrf
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 170M
Ch0: Freq. Corr. (ppm): 0
Ch0: RF Gain (dB): 35
Ch0: IF Gain (dB): 50
Ch0: BB Gain (dB): 50
Ch0: Bandwidth (Hz): 100k

WX GUI Slider
ID: ptt
Default Value: 0
Minimum: 0
Maximum: 1
Converter: Float

WX GUI Slider
ID: rf_gain
Label: RF Gain
Default Value: 35
Minimum: 0
Maximum: 50
Converter: Float

WX GUI Slider
ID: if_gain
Label: IF Gain
Default Value: 50
Minimum: 0
Maximum: 50
Converter: Float

WX GUI Slider
ID: bb_gain
Label: BB Gain
Default Value: 50
Minimum: 0
Maximum: 50
Converter: Float

Wav File Source
File: ...und Station 2020.wav
Repeat: Yes

Multiply Const
Constant: 1

Audio Source
Sample Rate: 48k

Noise Source
Noise Type: Gaussian
Amplitude: 1
Seed: 0

Figure 3.2.1 Jamming GNU Radio flowgraph

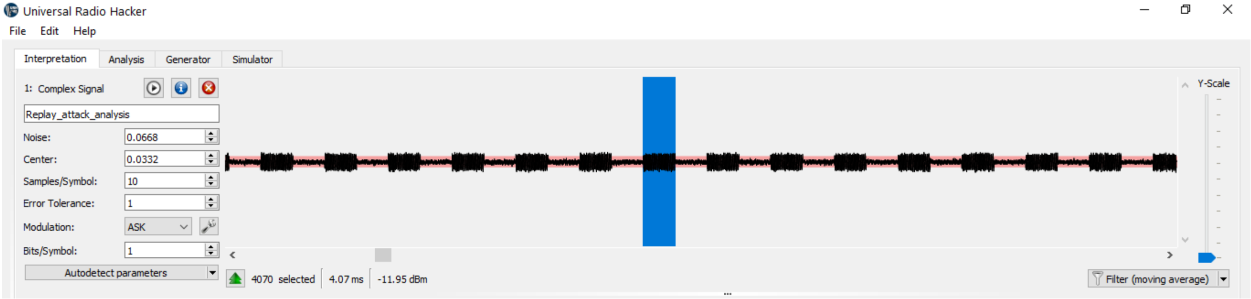
3.2.2 Silent Jamming

GNU Radio provides option to send .wav sound files, so we thought of another method for radio jamming. When we insert noise in the frequency, operator and some modern radios easily detect that they are being jammed. They then switch towards possible counter measures like frequency hopping or switching to another channel. We came up with an idea of silent files, what we did is that we recorded silent radio files and converted them to .wav format which GNU Radio supports and then transmitted these files with high power towards target radio set at same modulation and frequency target radio set is using. Figure 3.2.1 explains how noise or .wav silent files can be modulated to narrow band FM in any desired frequency.

3.3 Spoofing

As explained earlier, in this type of attack we must analyze signal first to transmit data of our own choice to make target appliance act in an abnormal pattern. We utilized a program known as Universal Radio Hacker to record and analyze data transmitted by remote of a RC toy car. Figure 3.3.1 shows that the signal was Amplitude Shift Keyed with data 1 and 0 being transmitted alternatively for forward move of the car. one symbol contains 4290 samples, and one symbol is being transmitted at 4.07 ms symbols hence symbol rate is 245 symbols per second. it takes 7.09 ms to retransmit symbol ‘1’. Also if we zoom transmitted signal we can see that the data is in form of sine wave with frequency 6.66 MHz as per Figure 3.3.3

So now, we have idea of what we can make this duplicate signal using GNU Radio as shown in figure 3.3.4



1: Complex Signal

Replay_attack_analysis

Noise: 0.0668

Center: 0.0332

Samples/Symbol: 10

Error Tolerance: 1

Modulation: ASK

Bits/Symbol: 1

4070 selected | 4.07 ms | -11.95 dBm

Figure 3.3.1 Radio Control RC car signal analysis

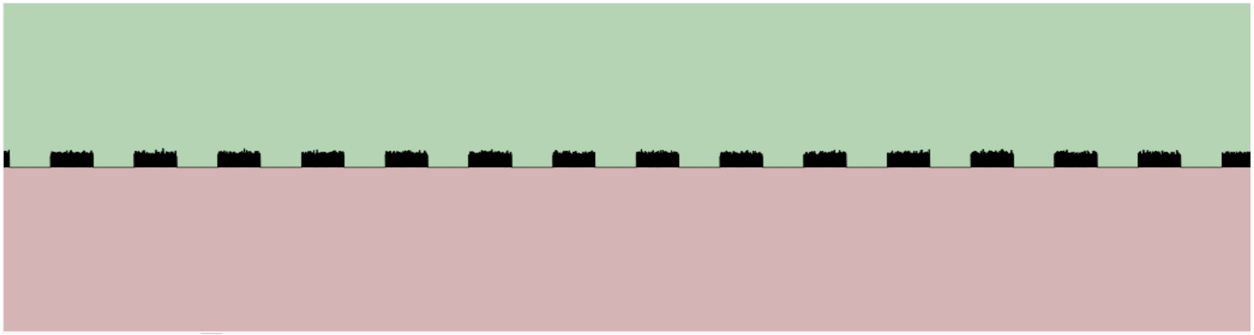
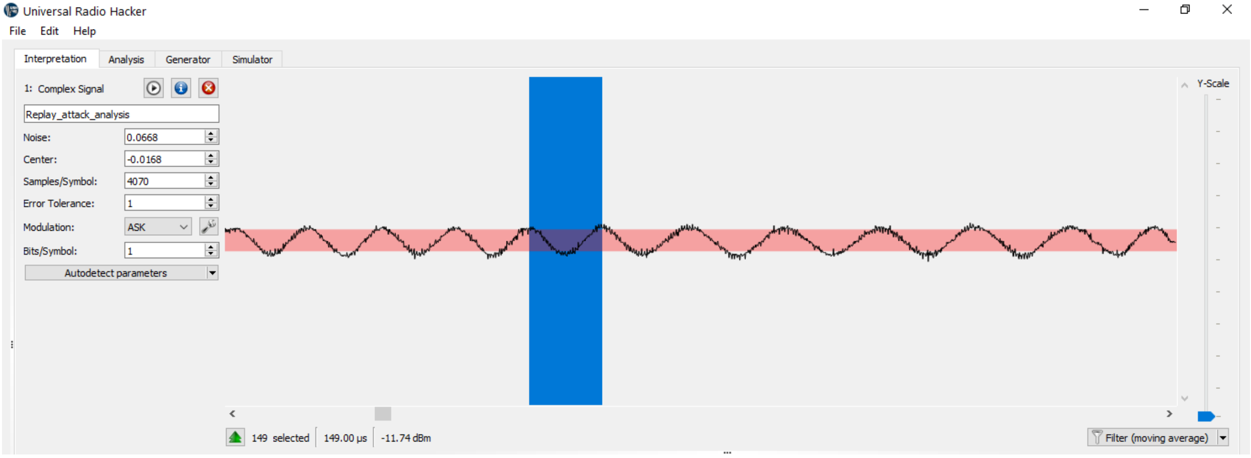


Figure 3.3.2 Demodulated Signal



1: Complex Signal

Replay_attack_analysis

Noise: 0.0668

Center: -0.0168

Samples/Symbol: 4070

Error Tolerance: 1

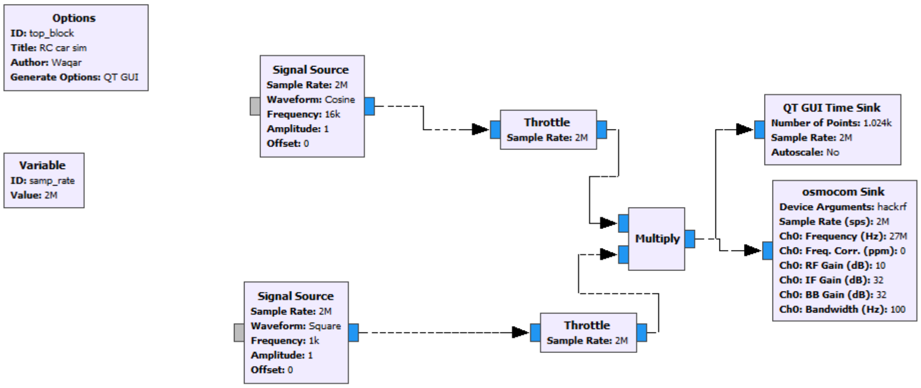
Modulation: ASK

Bits/Symbol: 1

149 selected | 149.00 µs | -11.74 dBm

Figure 3.3.3 Data frequency

As 7.09ms is used to transmit subsequent 1 we can use a square wave of frequency 1/7.09m = 141Hz and multiply this square wave with sine wave of frequency 6.67 kHz. Once transmitted this signal at 27 MHz, we successfully spoofed RC car to move forward without giving actual command from its remote control.



Options
ID: top_block
Title: RC car sim
Author: Waqar
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 2M

Signal Source
Sample Rate: 2M
Waveform: Cosine
Frequency: 16k
Amplitude: 1
Offset: 0

Signal Source
Sample Rate: 2M
Waveform: Square
Frequency: 1k
Amplitude: 1
Offset: 0

Throttle
Sample Rate: 2M

Throttle
Sample Rate: 2M

Multiply

QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 2M
Autoscale: No

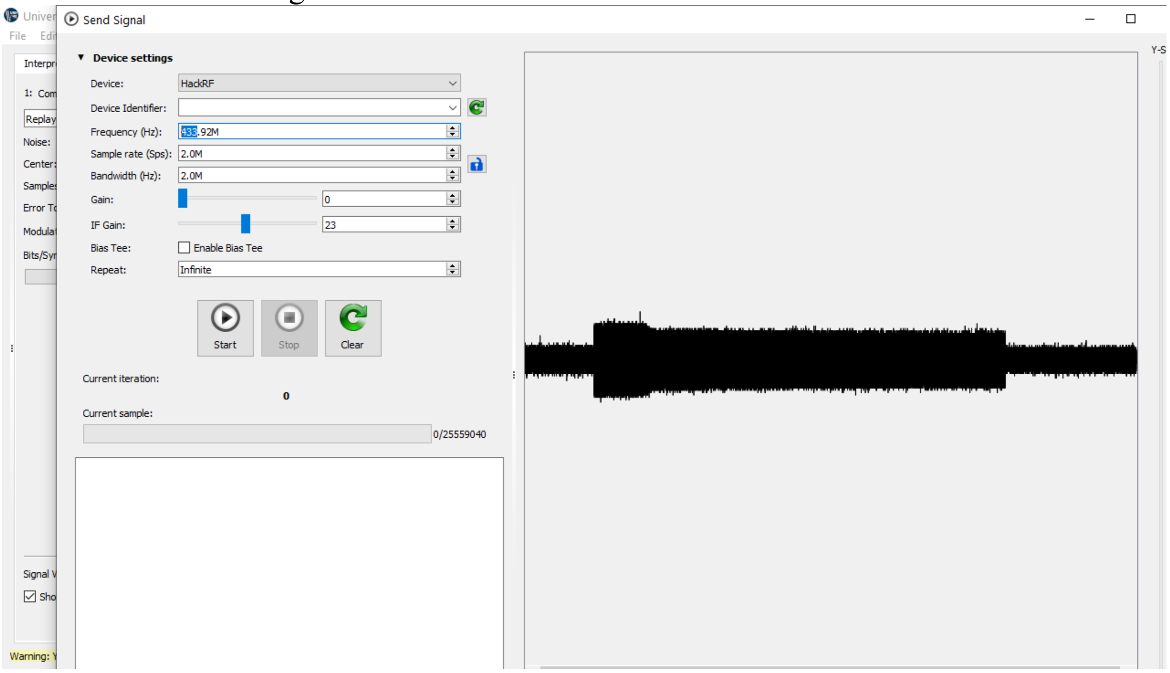
osmocom Sink
Device Arguments: hackrf
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 27M
Ch0: Freq. Corr. (ppm): 0
Ch0: RF Gain (dB): 10
Ch0: IF Gain (dB): 32
Ch0: BB Gain (dB): 32
Ch0: Bandwidth (Hz): 100

Figure 3.3.4 RC car Spoofing GNU Radio flowgraph

3.4 Replay Attack

In replay attack, we transmit signal in the same form as recorded. This can be useful for the situation when a signal is in very complex form and one is unable to analyze it completely. The signal can be recorded using Universal Radio Hacker as well as GNU Radio and without performing any operation on signal, we can transmit is using same software.

We have used RC car and RC drone for the said attack and used Universal Radio Hacker to record and retransmit the signal.



Send Signal

▾ Device settings

Device: HackRF
Device Identifier:
Frequency (Hz): 433.92M
Sample rate (Sps): 2.0M
Bandwidth (Hz): 2.0M
Gain: 0
IF Gain: 23
Bias Tee: □ Enable Bias Tee
Repeat: Infinite

Current iteration: 0

Current sample: 0/25559040

Figure 3.4.1 Retransmission of recorded signal

Chapter 4: SIGNAL IDENTIFICATION AND FINGERPRINTING

The last part of the project comprises of most important part of our project. Fingerprinting radio signals is a process to distinguish two or more radios or any wireless device, of same model; based on irregularities in frequency spectrum caused by radio transmission. These irregularities are known as parameters, and they are present due to slight differences between components of same models during manufacturing as well as during its operational life due to mishandling or accidental fall etc.

Currently, Pakistan Army lacks this technology and we proudly state that our efforts has contributed a huge step towards this technology.

4.1 Parameters

There are two main parameters for signal identification we considered during our research. One is its intended transmission and second is spurious transmission. The intended transmission is the kind of transmission that a radio is asked to transmit by its operator. This transmission is being aired on the frequency defined by the operator, the differences between these transmission of same model radios will be known as 'Indented transmission irregularities[6]. Meanwhile, we have observed that due to differences in local oscillator, radio also transmit signals other than defined frequencies and these transmissions differ from each other, these types of irregularities are covered in 'Spurious transmission irregularities.

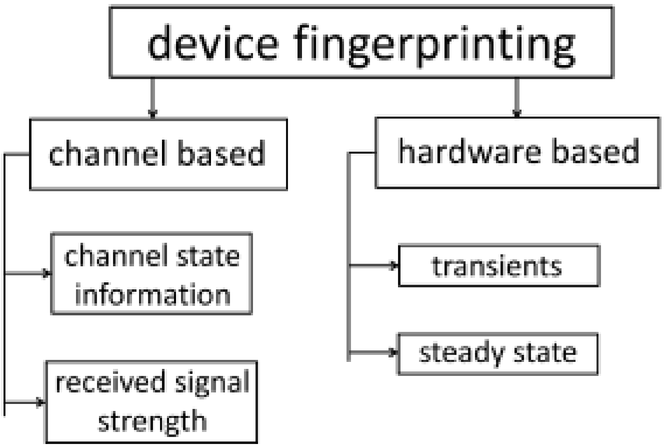
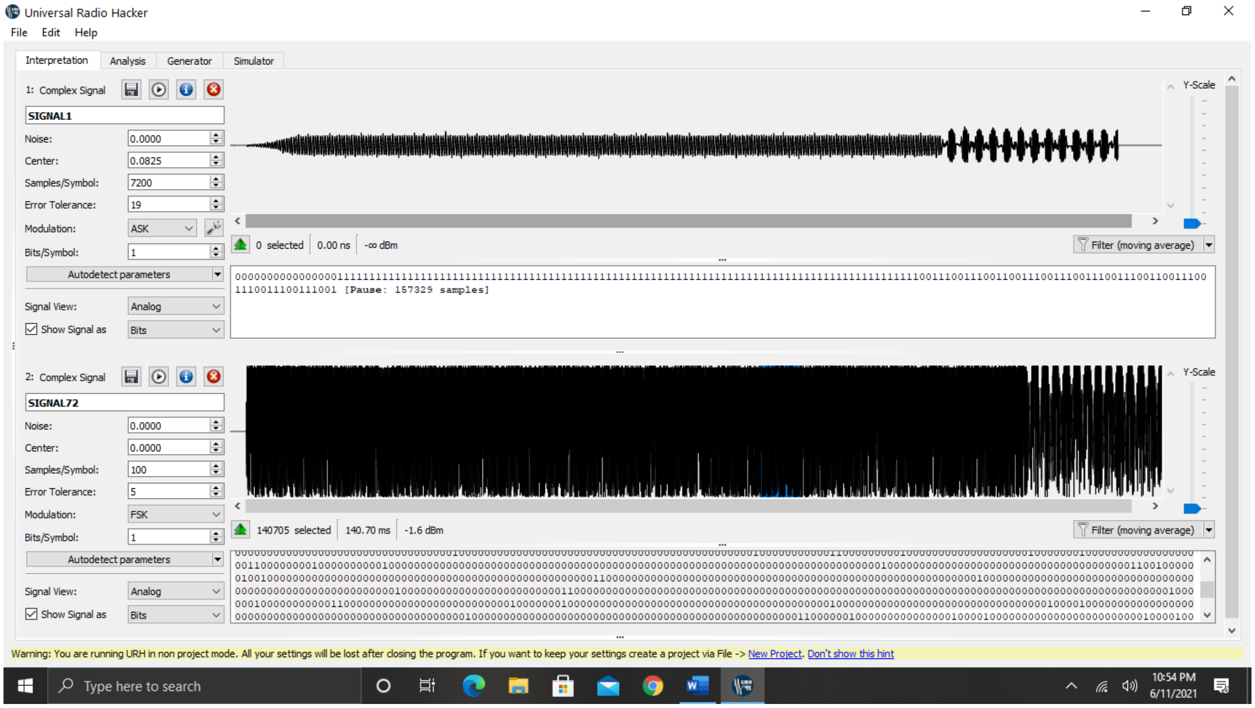


Figure 4.1.1 Simplified classification of wireless devices fingerprinting methods

4.1.1 Intended transmission irregularities.



1: Complex Signal

SIGNAL1

Noise: 0.0000
Center: 0.0825
Samples/Symbol: 7200
Error Tolerance: 19
Modulation: ASK
Bits/Symbol: 1
Autodetect parameters
Signal View: Analog
☑ Show Signal as Bits

0 selected | 0.00 ns | -∞ dBm | Filter (moving average)

```
0000000000000000111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111001110011100110011100111001110011100110011100
1110011100111001 [Pause: 157329 samples]
```

2: Complex Signal

SIGNAL72

Noise: 0.0000
Center: 0.0000
Samples/Symbol: 100
Error Tolerance: 5
Modulation: FSK
Bits/Symbol: 1
Autodetect parameters
Signal View: Analog
☑ Show Signal as Bits

140705 selected | 140.70 ms | -1.6 dBm | Filter (moving average)

```
000000000000000000000000000000000010000000000000000000000000000000000000000000000100000000000110000000001000000000000000000010000000100000000000000000
001100000000100000000001000000000000000000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000001100100000
010010000000000000000000000000000000000000000000000000001100000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000
000000000000000000000000010000000000000000000000000110000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001000
000100000000000110000000000000000000000000100000001000000000000000000000000000000000000000001000000000000000000000000000000000100001000000000000000000
000000000000000000000000000000000000100000000000000000000000000000000000000000000000000110000001000000000000000100001000000000000000000000001000010000
```

Warning: You are running URH in non project mode. All your settings will be lost after closing the program. If you want to keep your settings create a project via File -> New Project. Don't show this hint

10:54 PM
6/11/2021

Figure 4.1.2 Intended transmission irregularities.

4.1.2 Test conditions

We recorded time-based graph of two radios that were manufactured by same company and had same model (W-118). The signals were recorded at same distance and they were set to transmit with same power.

4.1.3 Results

Once their signal was recorded, we were able to easily distinguish between their transient response. As you can see in Figure 4.1.1 the signal of radio 1 has ramping transient response on the other hand, Radio 2 has sudden sharp response. It is very hard to distinguish between two radios in steady state, but after several experiments we concluded that Radio 2 was transmitting data with higher amplitude than Radio 1. Here I would like to mention that the amplitude should not be confused with transmitting power. Since the radios were modulating signal on NBFM, there amplitude will be constant when sending data, this constant can be of any value and this value, by our experiment we found that, differs with each radio being used.

4.1.4 Spurious transmission irregularities.

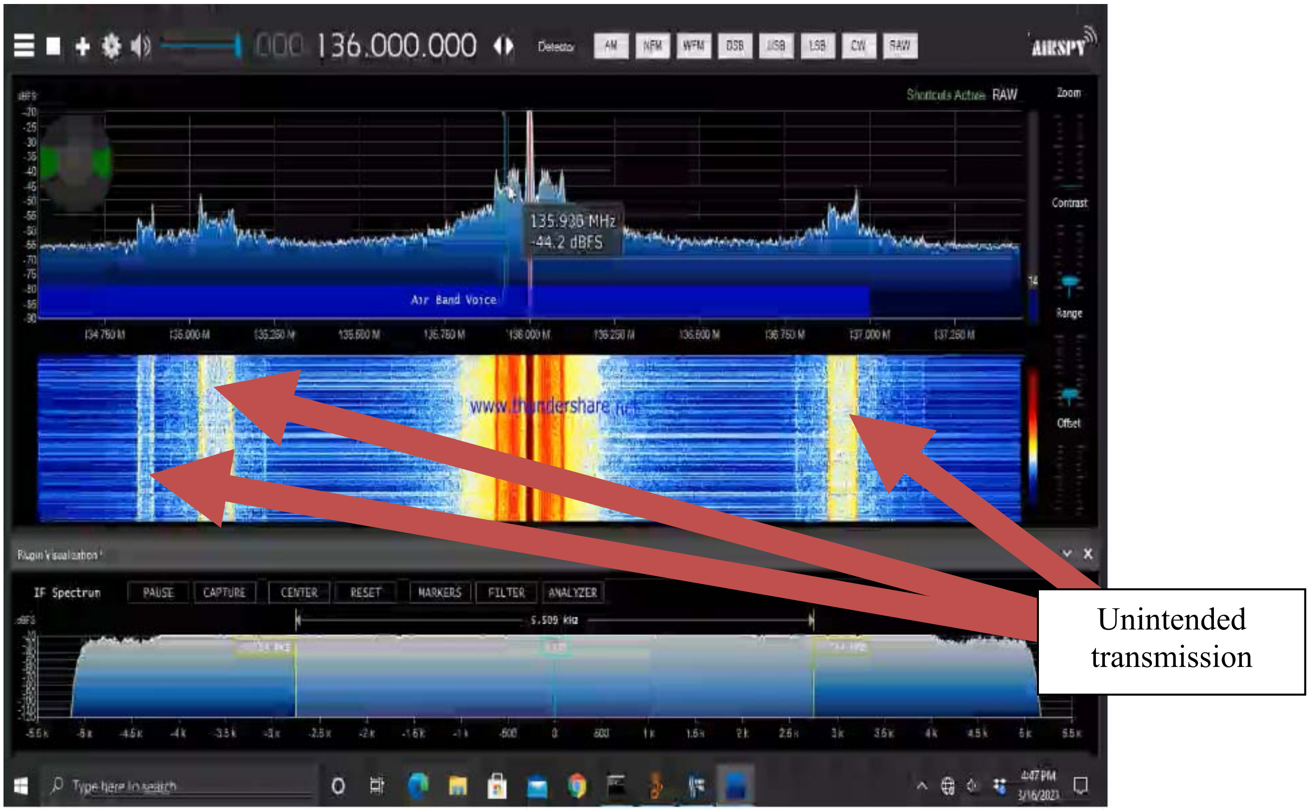


Figure 4.1.2 Unintended transmission irregularities in LMR

Unintended transmission by radios can be due to various reasons including transistor switching, current flow and integrated circuit activity, in addition of other electromagnetic effects. Although shielding and design are used to reduce unintended emissions, the underlying physics of electronic devices precludes their elimination. These unintended radio transmissions are different for different radio sets as it is nearly impossible for a manufacturer to create conditions exactly same during manufacturing of each radio set[7]. Also, there is slight variation within values of individual components that are used in radio sets which can never be absolute. Such conditions made possible for us to conduct successful experiments for fingerprinting various radio sets.

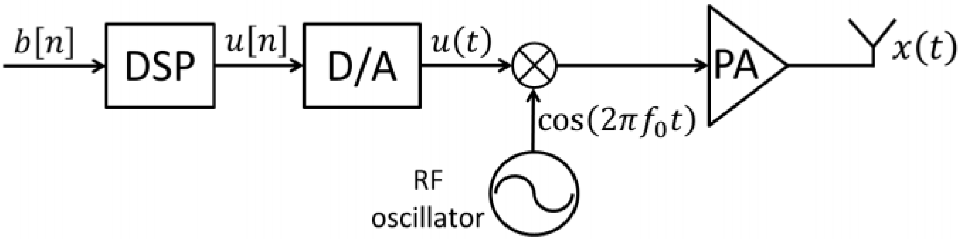


Figure 4.1.3 Basic components of a wireless transmitter, the imperfections of which can be exploited for user identification. b[n] is the sequence of bits to be transmitted, u[n] and u(t) are the digital and analog baseband waveforms, respectively, and x(t) is the transmitted signal up-converted to a carrier frequency fc.

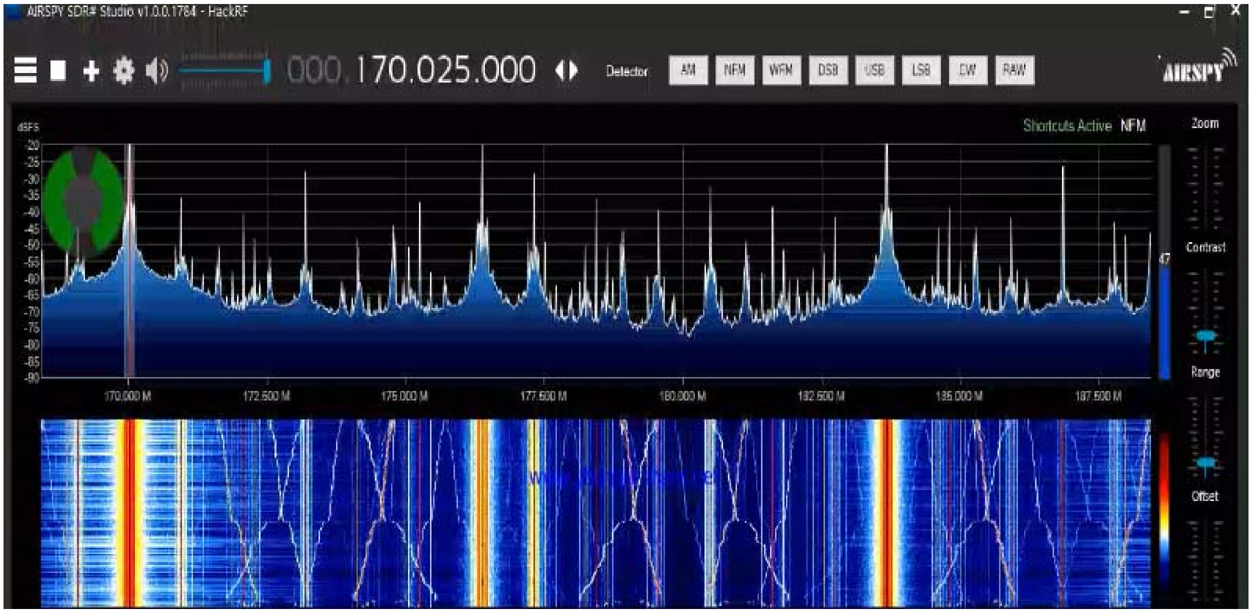


Figure 4.1.2 Unintended transmission irregularities Radio 1

In Figure 4.1.2 unintended radio transmissions of two radio sets are displayed which when compared, shows distinctive patterns of spurious transmission of each radio set. We recorded these signals repeatedly and they always showed same distinctive pattern. Hence, we concluded that these patterns can be associated to each radio and is unique for each radio set.

This can be possible that some radios do not emit unintended radio transmissions at all, and some radios do not show any intended transmission variations. Fingerprinting is thus unique for each radio and can only be successful with huge database containing multiple emission data of each radio in different conditions. This database can be fed into deep learning algorithm and precision results can be obtained with it.

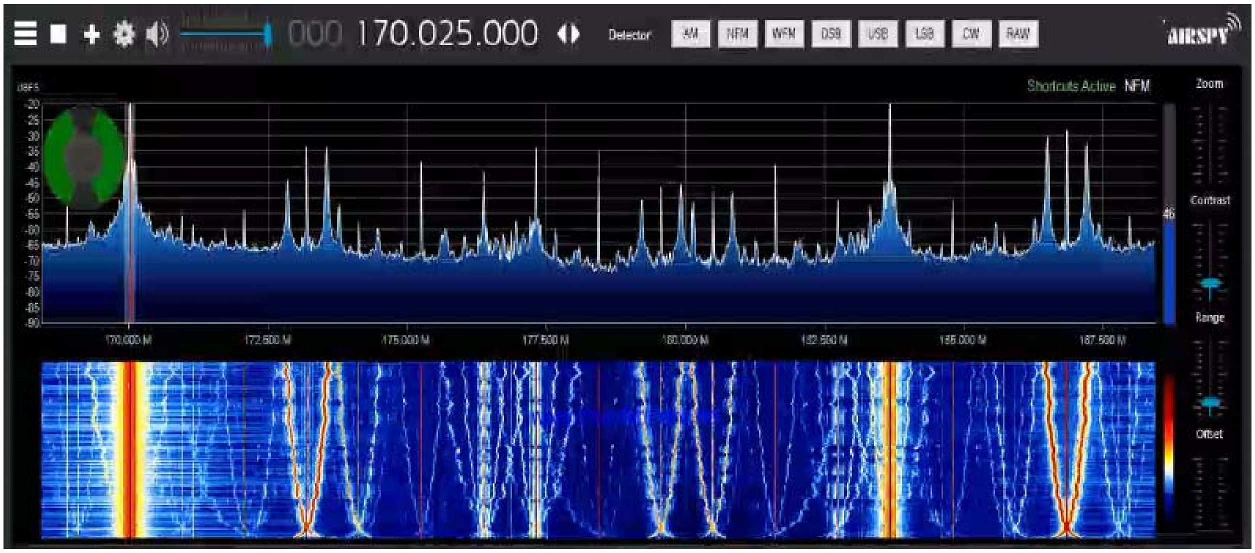


Figure 4.1.4 Unintended transmission irregularities Radio 2

REFERENCES

[1] Paul Clark, David Clark, ed. (2015) *Field Expedient SDR, Vol 1* .2nd edition. Meadow Registry, LLC

[2] Qing Yang, Lin Huang. *Inside Radio: An Attack and Defense Guide.* 1st edition. Singapore: Springer

[3] Derek Kozel, *GNU Radio Wikipedia*, 1 June 2021, https://wiki.gnuradio.org/index.php/Main_Page

[4] Michael Ossman, *Great Scott Gadgets*, 2021, https://greatscottgadgets.com/hackrf/one/

[5] Andrea De Martino. ed. (2012) *Introduction to Modern EW System.* USA: Artech House

[6] Bihl, Trevor J., "Feature Selection and Classifier Development for Radio Frequency Device Identification" (2015). Theses and Dissertations. 231. https://scholar.afit.edu/etd/231

[7] Donald R. Reising, DR-II, Exploitation of RF-DNA for Device Classification and Verification Using GRLVQI Processing USAF AFIT-ENG-DS-12-04

WIRELESS INTERCEPTION AND SNIFFING VIA SDR

ORIGINALITY REPORT

11%	7%	3%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	3%
2	wiki.gnuradio.org Internet Source	2%
3	Submitted to University of New South Wales Student Paper	2%
4	en.wikipedia.org Internet Source	2%
5	Submitted to Wright State University Student Paper	1%
6	scholarworks.umass.edu Internet Source	1%
7	Submitted to Cranfield University Student Paper	<1%
8	Michael Holloway, Chikezie Nwaoha. "Dictionary of Industrial Terms", Wiley, 2013 Publication	<1%

9	Polak, Adam C., Sepideh Dolatshahi, and Dennis L. Goeckel. "Identifying Wireless Users via Transmitter Imperfections", IEEE Journal on Selected Areas in Communications, 2011. Publication	<1%
10	dblp.dagstuhl.de Internet Source	<1%
11	www.science.gov Internet Source	<1%
12	Menon, Vrinda N., and Anagha Suresh. "Analyzing digital overlay technique over marine voice channel for disaster dissemination", 2014 Eleventh International Conference on Wireless and Optical Communications Networks (WOCN), 2014. Publication	<1%
13	Qing Yang, Lin Huang. "Inside Radio: An Attack and Defense Guide", Springer Science and Business Media LLC, 2018 Publication	<1%
14	gitlab.eps.surrey.ac.uk Internet Source	<1%
15	scholar.afit.edu Internet Source	<1%

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off

Signature of Supervisor

Asst Prof Shibli Nisar, PhD