Application Of SDR

Guided By:
Prof. Sapan Mankad

Prepared By:
Mo Karimkhan

Nirma University

April-20,2013

Table of contents

Objective

Our objective is to learn how to transmit Bluetooth signal to various bluetooth nodes located under coverage area. Once node receive signals, it transmit back to main device. Different channels should be allocated to different bluetooth node.

On receiving the signal at source device, identify the location of particular sender and record it. Head device will learn to predict the location of particular node using supervised learning. So When user want to know the probable location of particular device then based on previous experience it can be predicted.

This approach can help bling people to keep track of surrounding devices. Those who forget their things very often can keep track of

SDR

Software defined radio is a radio communication systems wherein which the hardware like filters, amplifiers, modulators etc are implemented in the personal computers or some embedding devices. By doing so, hardware complexity can be widely reduced. In addition to software part there is a RF front end preceding this software section. By using such a design one can transmit and receive variety of signals based on the applications .

In SDR, signal is captured by an antenna which is further converted into digital samples with regular intervals. These digital values are then processed in software, where the required application is written. The resulting output can be then converted back into audio, video or required form.

GNU Radio

GNU Radio is a free and open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in hobbyist, academic and commercial environments to support both wireless communications research and real-world radio systems.[4]

GRC

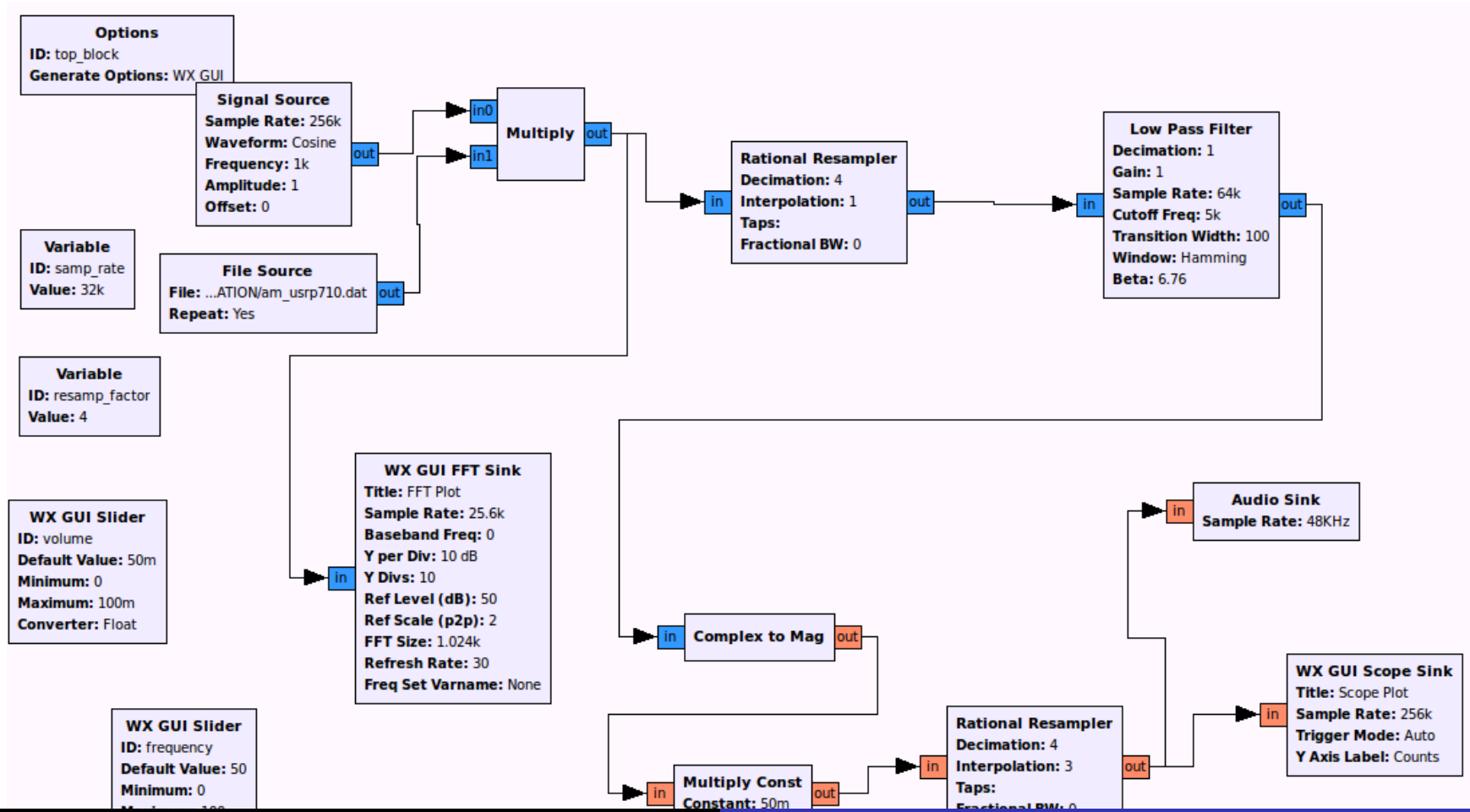
The GUI termed GRC allows user to implement GNU radio signal processing blocks in a manner similar to Simulink and Labview.

GRC provides a graphical user interface to the user so that any hardware functions like mixers, oscillators, etc., can be implemented as a block and can be executed. These blocks use Python code to define its functioning (at the background) and xml code for creating GUI (in the foreground). [4]

GRC Execution

- Execute the grc file:
- python topblock.py
- After execution it generated the flow graph with GUI display.
- Now we take Signal recorder by USRP hardware device and simulate how the system works. This signal was recorder keeping USRP at 710 KHz and sample rate 256KHz.

Execute recorder .dat file



Block description

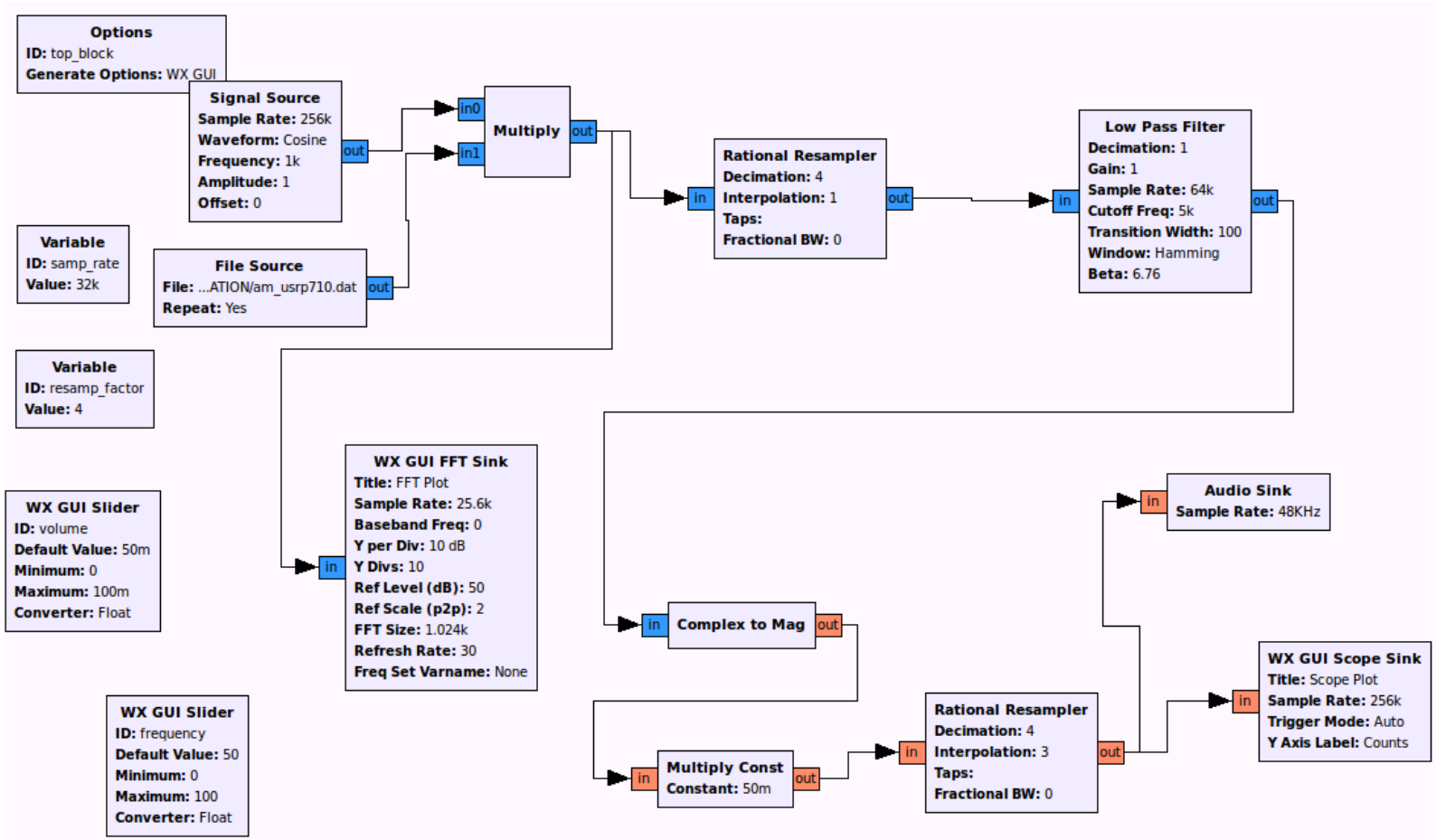
Some basic blocks of the grc flow diagram:

- Rational Resampler that we have added here will control the rate of incoming signal.
- Decimation: Reduce the sample rate by given factor
  Interpolation: Increase the sample rate in multiplication factor.
- Scope sink is used to see output in graphical windows.
- Multiple constant to apply an attenuator to reduce the size of the samples.

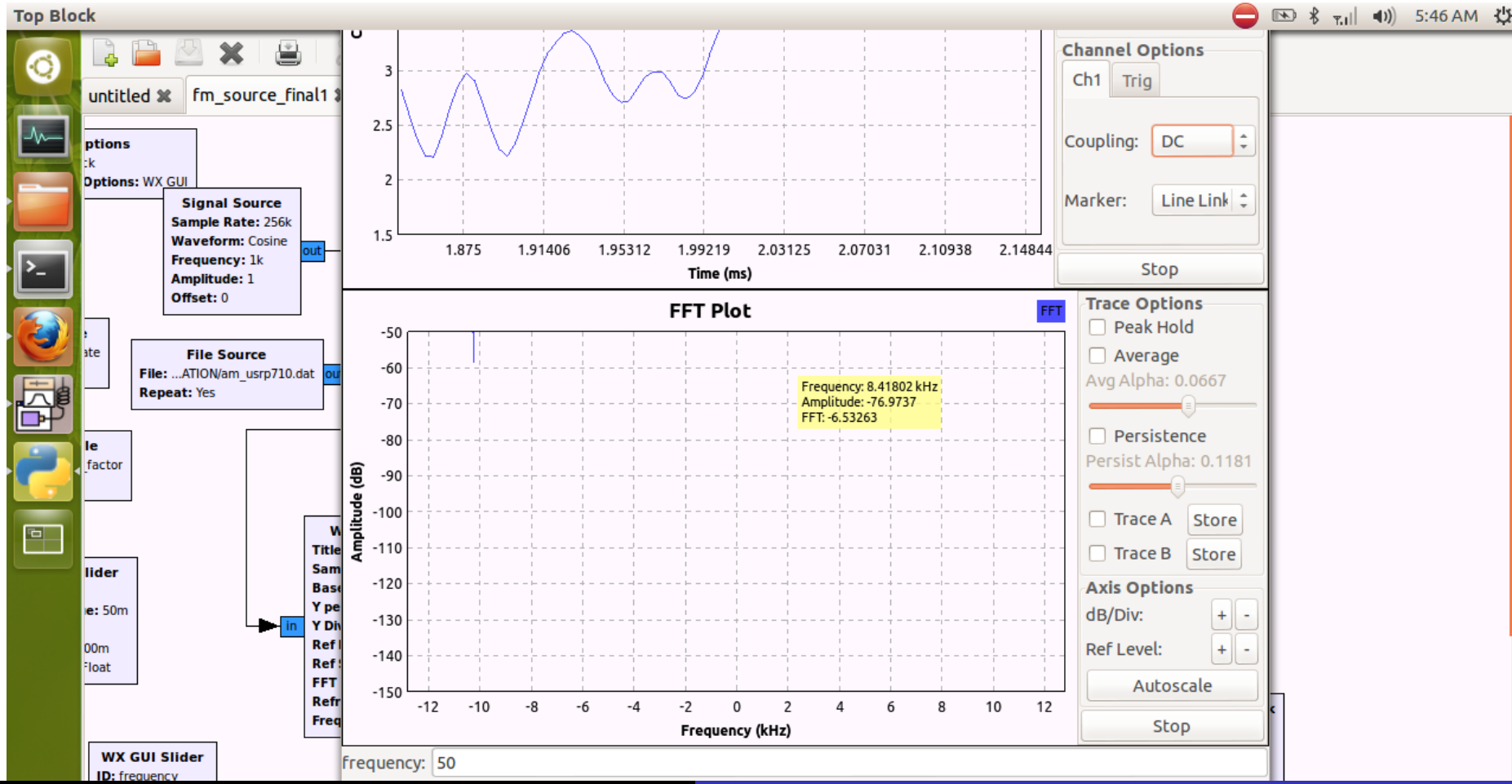
The next step is to demodulate the signal. In the case of AM, the baseband signal is the envelope or the magnitude of the modulated waveform.

GNU Radio contains a Complex to Mag block (in the Type Conversions menu) that can be used for this purpose. However

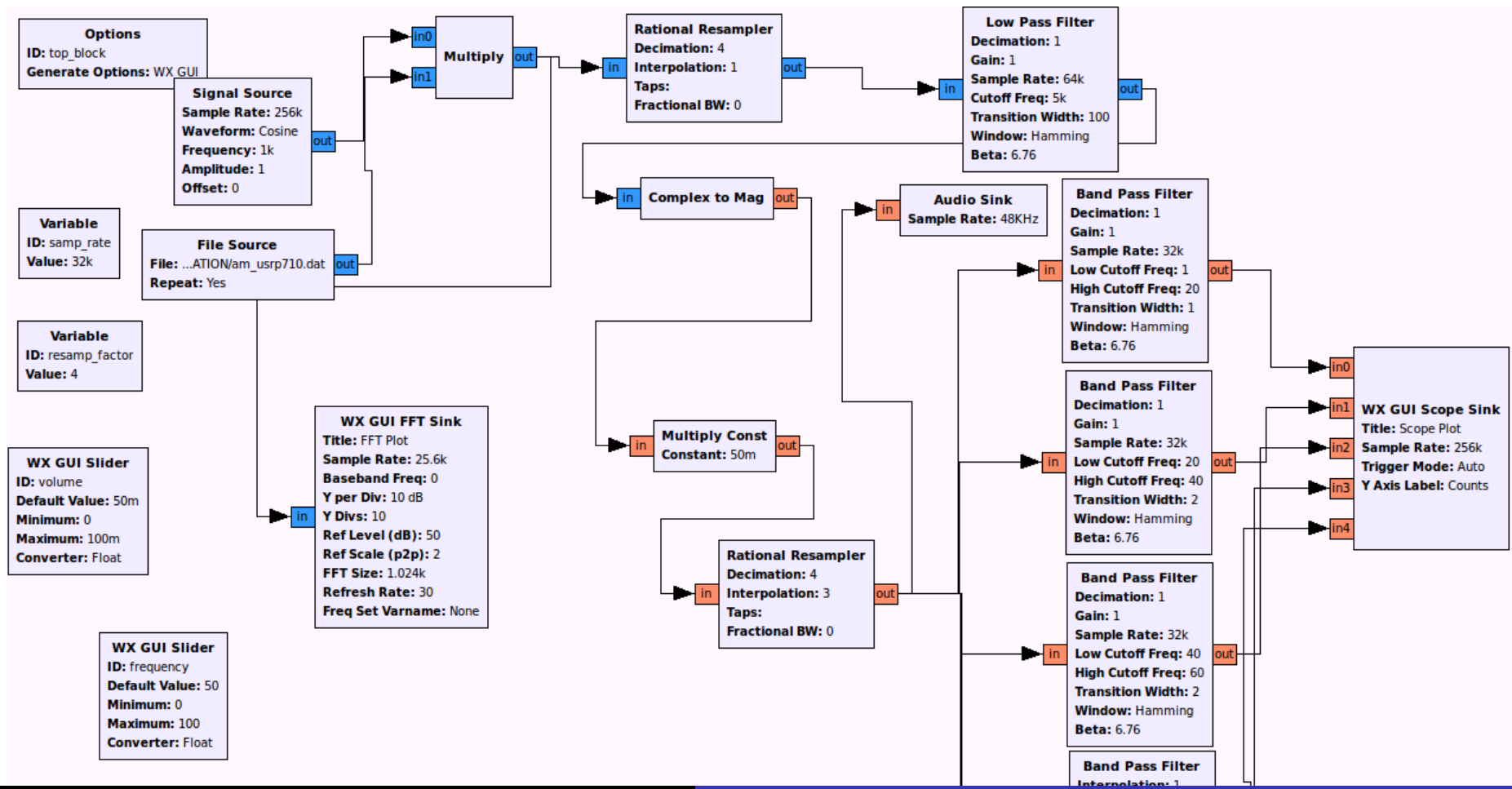
Execute recorded .dat file



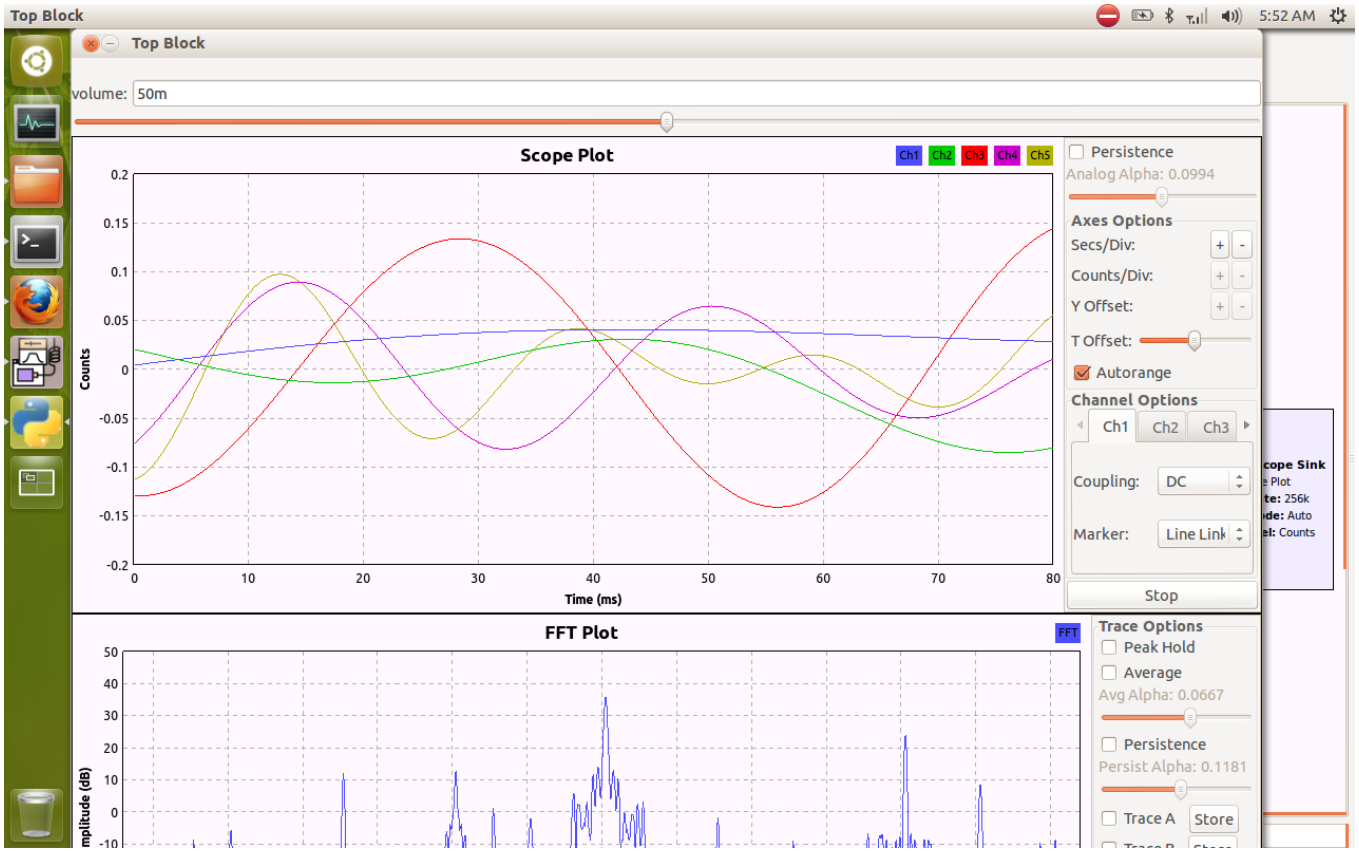
Execute recorded .dat file



Execute recorded .dat file



Execute recorded .dat file



Execute recorded .dat file

Bluetooth Device location tracking

If we embedd this SDR application in mobile device which deal with bluetooth frequency range then we can track the location of different bluetooth nodes.

Devide the range of bluetooth frequency as per the numbers of bluetooth nodes.

Similarly using USRP source block we can receive the signal coming from device and store and manipulate them.

Classification using Python web interface

Time Stamp	Description	Channel	Channel Status	DTV Detect Counts	Pilot Peak Power	Channel Power	NTSC Detect Counts	FMW Detect Counts/Peak Position	UnkS Peal Pow
20080129131849	scan	21	Occupied	0	0	-102.22783	0	258	-109.3
20080129131849	scan	22	Available	0	0	-102.33383	0	0	0
20080129131849	scan	23	Available	0	0	-101.7682	1	0	0
20080129131849	scan	24	Available	0	0	-101.46953	0	0	0
20080129131849	scan	25	Available	0	0	-101.37267	0	0	0
20080129131849	scan	26	Available	0	0	-101.41559	0	0	0
20080129131849	scan	27	Occupied	0	0	-101.22413	0	1044	-110.7
20080129131849	scan	28	Available	0	0	-101.33753	0	0	0
20080129131849	scan	29	Available	0	0	-101.14767	0	0	0
20080129131849	scan	30	Available	0	0	-100.88655	0	0	0
20080129131849	scan	31	Occupied	0	0	-100.47311	0	1282	-109.7
20080129131849	scan	32	Available	0	0	-100.59804	0	0	0
20080129131849	scan	33	Available	0	0	-100.77374	0	0	0
20080129131849	scan	34	Available	0	0	-101.1159	0	0	0
20080129131849	scan	35	Occupied	0	0	-101.6297	0	1581	-114.3

[1]

Channelization

For each bluetooth device we need different channel so that we can identify uniqely. [1]

I have use bandpass filter(BPS) here to do channelization. There may be more efficient block to do the same but in absent of USRP I prefer to work with BPS.

If USRP is available to use then we can apply similar kind of channelization and transmit the signal to targeted device.

Similarly using USRP source block we can receive the signal coming from device and store and manipulate them.

Classification using Python web interface

Time Stamp	Description	Channel	Channel Status	DTV Detect Counts	Pilot Peak Power	Channel Power	NTSC Detect Counts	FMW Detect Counts/Peak Position	UnkS Peal Pow
20080129131849	scan	21	Occupied	0	0	-102.22783	0	258	-109.3
20080129131849	scan	22	Available	0	0	-102.33383	0	0	0
20080129131849	scan	23	Available	0	0	-101.7682	1	0	0
20080129131849	scan	24	Available	0	0	-101.46953	0	0	0
20080129131849	scan	25	Available	0	0	-101.37267	0	0	0
20080129131849	scan	26	Available	0	0	-101.41559	0	0	0
20080129131849	scan	27	Occupied	0	0	-101.22413	0	1044	-110.7
20080129131849	scan	28	Available	0	0	-101.33753	0	0	0
20080129131849	scan	29	Available	0	0	-101.14767	0	0	0
20080129131849	scan	30	Available	0	0	-100.88655	0	0	0
20080129131849	scan	31	Occupied	0	0	-100.47311	0	1282	-109.7
20080129131849	scan	32	Available	0	0	-100.59804	0	0	0
20080129131849	scan	33	Available	0	0	-100.77374	0	0	0
20080129131849	scan	34	Available	0	0	-101.1159	0	0	0
20080129131849	scan	35	Occupied	0	0	-101.6297	0	1581	-114.3

[1]

Future work

Preparing a Software Defined Radio for android based system using Bluetooth frequency range. By placing small bluetooth chips in house or office on various other device, we can track the location of all those device.

Recording the location of bluetooth devices on each search, applying machine learning techniques like decision tree or other we can predict the location of device.

Conclusion

So we can conclude that using SDR we can transmit and receive any range of signal in surrounding area using USRP.

If we record the receipt of signal for each channel, we can apply superwised learning techniques and predict the location of any particular device. We have to place transceiver on that device.

W. M. Brown, J. E. Clark, A GNU Radio Based Cognitive Signal Identification and Classification Platform III, Metric Systems Corporation 2320 Cousteau Court, Vista, CA 92081 USA

Tatsunori Kimpara and Susumu Ishihara,Using GNU Radio for Experiments on Data Distribution in Wireless Ad-hoc Networks Graduate School of Engineering, Shizuoka University, para@ishilab.net, Graduate School of Science and Technology, Shizuoka University, ishihara@ishilab.net 3-5-1, Johoku, Hamamatsu-Shi, Shizuoka, 432-8011, Japan

Ahmed AZZOUNA, Amel GUEZMILa, Anis SAKLY, Abdellatif MTIBAAa,3rd International Conference on Ambient Systems,

Networks and Technologies Resource Allocation for Multi-user Cognitive Radio Systems using Multi-agent Q-Learning, CSR GROUP of Electronic and Microelectronic Laboratory, University of Monastir, Tunisia National Engineering School of Monastir, University of Monastir, Tunisia.

http://gnuradio.org/redmine/projects/gnuradio/wiki, GNU Radio