Final Project Report
on

USRP AS LINUX INTERFACE

Group Members (Team A1)

Amit Patil
Harshvardhan Aggarwal
Saurav Sarkar
Vishal Agnihotri

ABSTRACT:

As a part of this project, we aim to make use of Universal Software Radio Peripheral USRP devices for facilitating communication between two devices over wireless channels. For this we are using unregulated ISM frequency band i.e. 2.4GHz for sending a sine wave signals between wireless devices. A single containing a sine wave is first modulated and sent through one USRP device. This modulated signal is received by the second device, which then demodulates it to obtain original signal. After this sanity check enabling us to understand how USRP works, next we planned to do an audio streaming application that transmits audio file such as a song over the free wireless channel and received and demodulated on other. We then play back the received audio and provide a facility to store the audio as a file for future playbacks. Next we plan to vary the frequency on the transmitter side and implement frequency tuning at the receiver which allows it to tune into the frequency that the transmitter is transmitting on. With an aim of taking it to the next level, we also plan to transmit video over the USRP devices. This project was chosen because it provides us detailed insights into how the signal processing works and how factors like environment and presence of interference affect the transmission and eventual quality of the received signal. Also, through the project we understand, how varying the factors like modulation schemes, sampling rate etc. affects the reception quality.

INTRODUCTION:

The USRP product family includes a variety of models that use a similar architecture. A motherboard provides the following subsystems: clock generation and synchronization, FPGA, ADCs, DACs, host processor interface, and power regulation. These are the basic components that are required for baseband processing of signals. A modular front-end, called a daughterboard, is used for analog operations such as up/down-conversion, filtering, and other signal conditioning [2]. In this project we are using two USRP- N210 and communication is happening over ISM band 2.4 GHz.

The problem that we are addressing here is the one where we need to add connectivity between two remote devices which can transmit and receive signal in the wireless channel and communicate efficiently. Generally USRP is used for the testing purpose and not for the actual product. This is because of the overall flexibility introduced by default in the wireless transmitter and receiver by means of USRP. This particular provision has opened many research areas for USRP based designs. The GNU Radio using primarily the USRP uses a USB 2.0 interface, an FPGA, and a high-speed set of analog-to-digital and digital-to-analog converters, combined with reconfigurable free software. Its sampling and synthesis bandwidth is a thousand times that of PC sound cards, which enables wideband operation [3].

Following are some more applications where USRP is used in the real world [2]:

- RFID reader
- Testing equipment
- Cellular GSM base station
- GPS receiver
- FM radio receiver
- FM radio transmitter
- A digital television (ATSC) decoder
- Passive radar
- Synthetic aperture radar
- Amateur radio

- Teaching aid
- Digital Audio Broadcasting (DAB/DAB+/DMB) transmitter
- Mobile WiMAX receiver with USRP N2x0

RELATED WORK:

A lot of work has been done in the field of signal processing and GNU Radio has been hugely favored recently as a platform for application development. Due to its versatility and feature support capabilities, we also choose GNU radio as our development platform. There have been projects that deal with FM, AM and SSB receivers and also real time audio transmission over GNU radio. One of the interesting research has been done by Zhi Yan who demonstrated an experimental platform with the USRP board developed by Ettus Research that can facilitate the development of PHY and MAC layers functionality. Some key results from this experience so far are presented, including an over-the-air interoperability analysis of two different SDR (software defined radio) architectures in the same frequency band, the effect of dynamic spectrum access on legacy system and analysis of actual co-existence experiences involving primary users and secondary opportunistic spectrum users in [4]. Our research indicated that there wasn't much work done on transmitting streaming audio over the USRP devices. Also, none of the projects dealt with frequency tuning where in the frequency of the transmitter can be varied and the receiver has to tune into the same frequency.

PROBLEM DESCRIPTION:

With wireless communications, the frequency at which the transmitter transmits changes due to several reasons. It can either due the spread spectrum employed e.g. (FHSS) for security and frequency re-use or due to the design of transmitters (e.g. Variable frequency Systems) which use more than one frequency to transmit. Frequency tuning is an essential component of the receiver because if the receiver, for receiver to pick to accurate transmission, it needs to the exact frequency the transmitter is transmitting the data. Therefore, using GNU radio we plan to implement a frequency tuning mechanism, in which when the transmitter jumps to random frequency within a specified range, in the receiver we should be able to search and lock on to the transmitting frequency manually. Since, we were totally unfamiliar with the GNURadio platform, our progress plan was divided as follows:

1. Understand the hardware layout and the components of USRP setup.

2. Install GNURadio software toolkit that serves as a development platform for designing and implementing SDRs. Also, a number of dependencies were installed during this process.

3. Establish connectivity between the USRP- N210 and the PC using UHD (universal Hardware Driver) as an interface.

4. Once, connectivity was established, we ran a number of benchmark program setups provided at GNU radio site [1] and [2] to get familiarized with USRP.

5. For Sanity, we sent a simple sine wave and checked whether the received modulated sine wave is properly getting demodulated in the receiver side.

6. After sanity check, we aimed for audio streaming where song being streamed from one station over wireless channel was being played on the receiver side.

7. Success of this made us ambitious enough to try for video streaming as well. But it involves lot of codec constraints hence we just managed to loopback the video streaming on a single device. This part anyways was just an addition to the specified task.

EXPERIMENTAL SETUP:

The project was divided into the following stages and is briefly explained below.

1. In the first stage, installation of Software defined Radio (SDR) provided by GNU known as GNURadio was installed. This step also involved installation of Universal Hardware Driver (UHD) for enabling communication between the PC and USRP.

2. In the second stage, two USRPs were used to test the communication medium i.e. wireless air interface with the aim to test whether the two USRP could send and receive a signal wave. This step also involved reading of materials related to USRP and its functioning and significant signal processing knowledge was also assimilated.

3. In the third stage, the basic communication between USRPs was extended to include audio transmission. Now, the USRP were able to transmit and receive streaming audio albeit with a little delay and noise.

The detailed explanation of each of the above three steps is as follows.

I. In the first stage, being the longest stage, a large amount of time was spent in understanding the functioning of GNURadio. It took a while to get accompanied with USRP, GNURadio and UHD since any of the information that exists was distributed among various websites and forums. The GNURadio and UHD were finally installed after finding a script build-gnuradio[1]. The installation required a computer to be running a 64-bit version of latest Ubuntu along with presence of 1 Gigabit Ethernet (1 GigE) equipment. In the next step, connectivity between the USRP and computer was tested by running command uhd_find_devices, after connecting them using a Ethernet cable. Blinking of LED lights D, F indicated that the USRP can listen to ping packets sent by host computer. This stage completed successful communication between a USRP and host computer via Ethernet.

II. The second stage was critical to understanding of the USRP boards along with the digital communication theory that followed it. In this stage we ventured to implement transmission of sinusoidal wave from host computer, pass it to USRP via Ethernet to transmit it over air, receive the signal over air by another USRP and then passing it to another computer via Ethernet. For transmitting and receiving a signal on USRP, a lot of digital signal processing skills are required like the sampling rate at transmitter and receiver side and the modulation demodulation schemes to be used. On the transmitter side, the signal source block that generates a sine wave was used. This block was connected to a re-sampler that would resample the received signal to a higher sampling rate that is supported by USRP. From the USRP, a frequency was chosen over which the USRP will transmit the signal wave. The USRP adds 2.4 GHz of frequency while transmitting and offsets it when receiving it. At the receiver side, the same frequency was used to capture the signal and again a re-sampler was applied to bring down the sampling rate to the original rate. In the end, an audio source and an FFT plot was used to measure and analyze the received signal. The block diagram of the process is shown below:

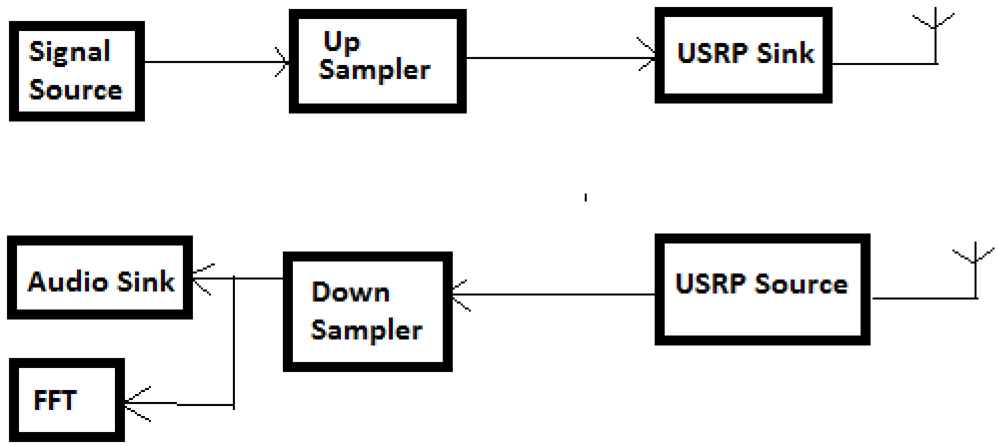

Figure 1: Block Diagram of the Setup

III. In the third, after successfully transmitting a sine wave to another computer wirelessly, the next step was to send an audio wave and play it on the other side. On the transmitter side, an audio source block was used which takes an audio file in .wav format. The next step was to modulate the raw audio signal so that it can be prevented from adding noise to actual signal when received on other side. For modulating, GNURadio provides many modulation blocks and one of them useful is the NBFM (Narrowband FM) transmitter. This block modulates the signal to FM. After this, again a re-sampler was required to bring up the sampling rate of the audio signal to match the sampling rate supported by USRP. The audio was transmitted over a FM band frequency in the range 80 MHz to 108 MHz.

At the receiver side, the same frequency was used to receive the signal at USRP. A frequency translating filter was used which performs two steps. In the first step it will shift the entire spectrum down by value specified by center frequency i.e. it filters the channel to pass the signal of interest, filters out rest of the signal in the band and shift the signal of interest down to zero frequency. In the second step of frequency translating filter, we apply a low pass filter so that other signals are filtered out from the signal of interest. Next a re-sampler was used to down sample the signal to actual sample rate followed by NBFM receiver which would demodulate the signal to produce actual signal. The output of this block was fed to an audio sink and an FFT was plotted to analyze the receiving signal. In the actual run, however, the audio signal received was appended with some noise but the peak of reconstructed signal was the same at both transmitter and receiver side.

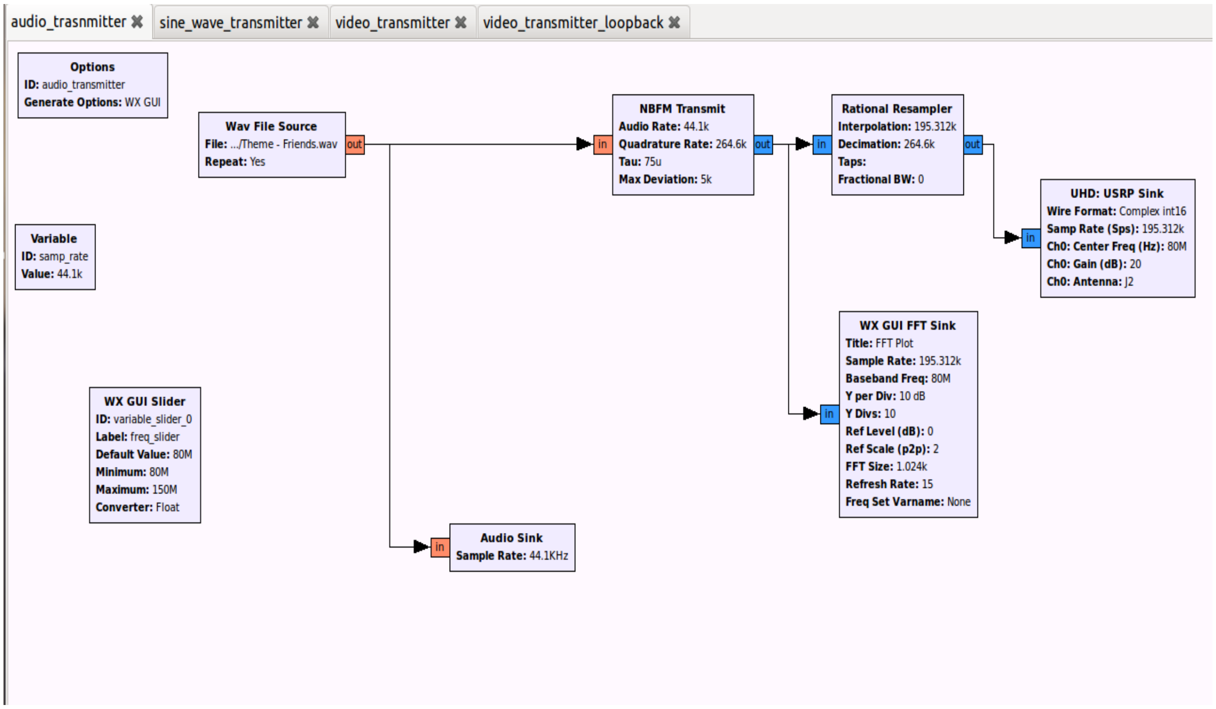

Figure 2: GRC code for Audio Transmitter

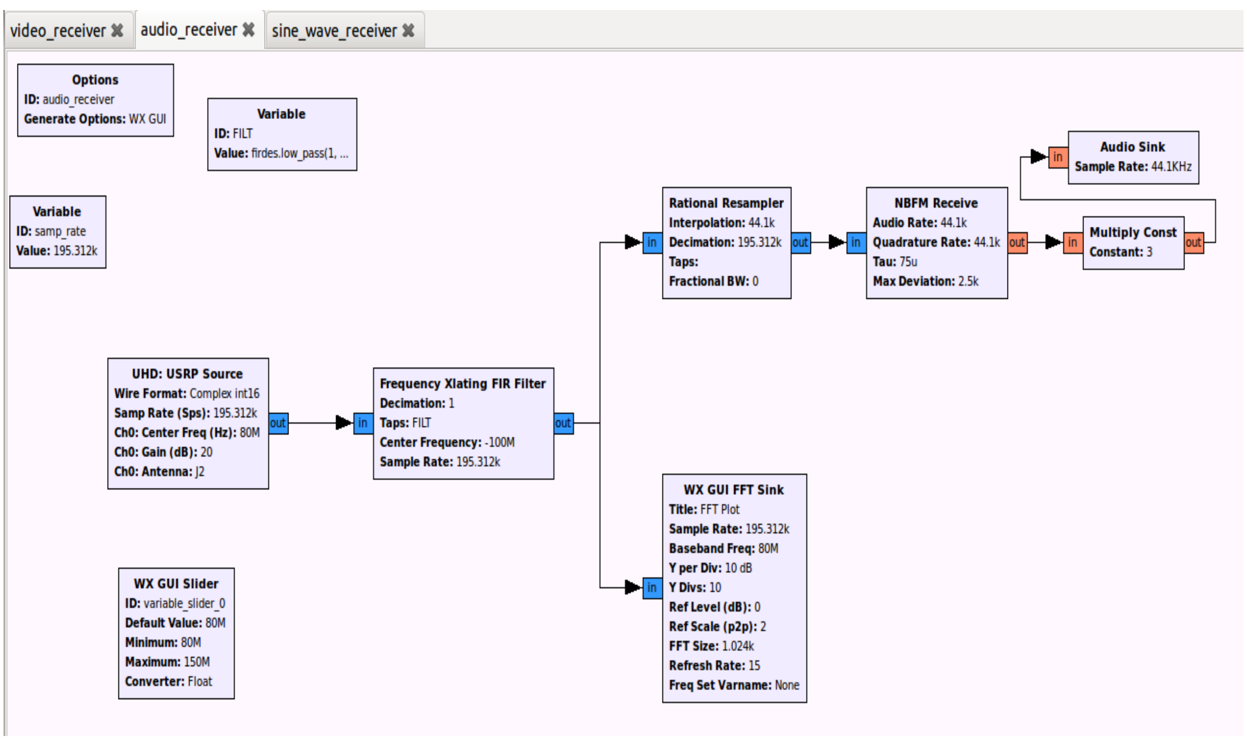

Figure 3: GRC code for Audio Receiver

EXPERIMENTAL RESULTS AND ANALYSIS:

Through above experiments aimed to make the USRP210 work as wireless interface so that it can be interfaced with any desktop device and provide wireless connectivity. With initial experiment we tried to send out a simple sine wave over one USRP to another. The aim of this experiment was to establish the basic connectivity over the USRP.

As explained above, we worked on the project one step at a time starting with establishing and testing basic connectivity over USRPs by sending sine wave and observing the FFT and signal source plot on the receiver. This task was completed successfully. We transmitted sine wave sampled at 32Kbps. We had to up sample this to 192.5 Kbps before it was transmitted over to the USRP. We didn't use any modulation for this setup because it was a simple setup and the reproduction of the signal was faithful. We also established the use frequency slider wherein we varied the frequency of the transmitter saw the changes in FFT and signal plot at the receiver.

As the second part of our project we aimed at transmitting audio (song) over the RF interface which was to be received at the receiver and played back. We choose to use Frequency Modulation over Amplitude Modulation (AM) due to its better Signal to Noise Ratios when higher BW were used. Also, restricting the noise is much easier in FM as compared to AM. Once, we decided the modulation scheme we choose NBFM (Narrowband Frequency modulation) on the transmitter side. Although wider bandwidth is desired for better audio quality but we wished to model our application as close as possible to real FM transmission. Also, since this was an experimental setup with local transmission and reception, choosing a narrow bandwidth didn't have much degrading effect on signal reception but we achieve a better spectrum efficiency. We had some issues with transmitting the FM over USRP and its faithful reproduction on the receiver. Especially with the sampling rate. We initially used the same 44.1 Kbps sample rate while transmitting and that resulted in abysmal reception. This we assume was because that the sample rate for the audio signal that we using on USRP was extremely low as compared to the USRP center frequency i.e. 80-100 MHz (for FM). We fixed the center frequency to 195.315 KHz and were able to achieve excellent reception on the receiver. We used frequency slider to vary the frequency on the transmitter and used a manual slider on the receiver so that it tuned perfectly to the new frequency of the transmitter.

The last experiment that we tried out was to send the video transmission over the USRP, we used a OFDM modulator with QPSK scheme. We used it because the mode is very efficient and digital signals can be made to take up very little bandwidth with QPSK scheme. We were to play the video successfully as a loopback on the same device but the receiver quality of signal was very poor. Hence, we were not able to successfully recreate the signal. We suspect that this was due to improper encoding scheme chosen which can be explored further.

The FFT for the transmitted and receiver signal is shown below. As seen, it’s centered around 80 MHz at which the FM audio is being transmitted.

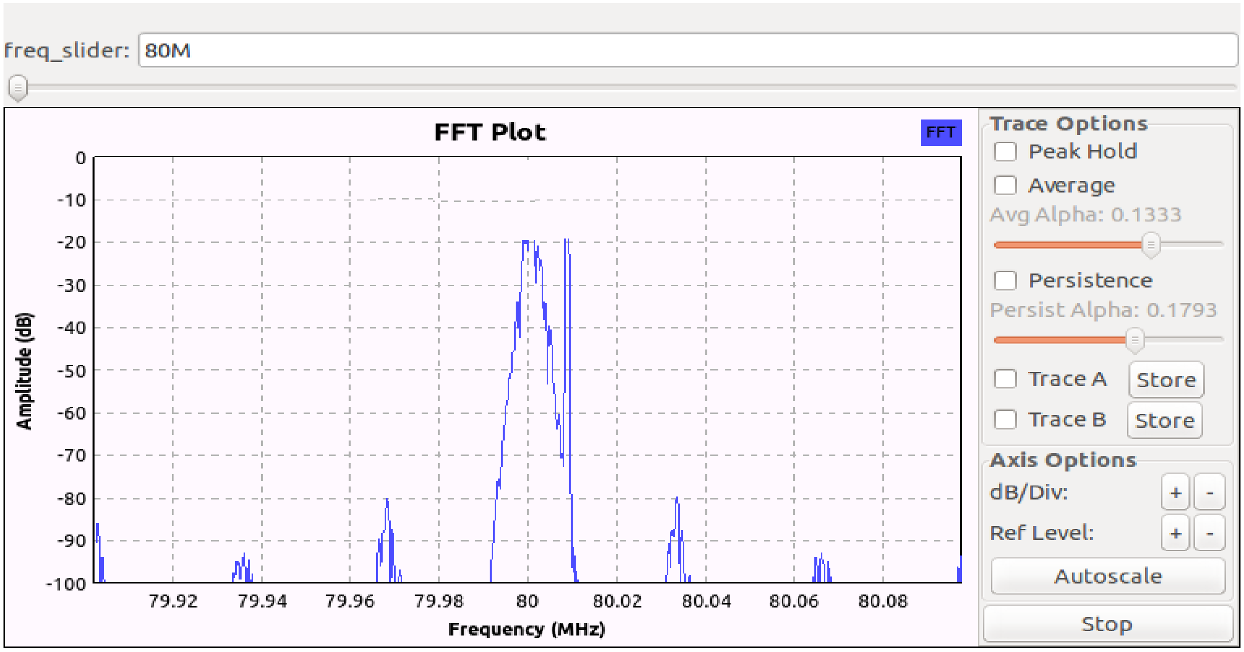

Figure 4: FFT of transmitted FM Audio Signal

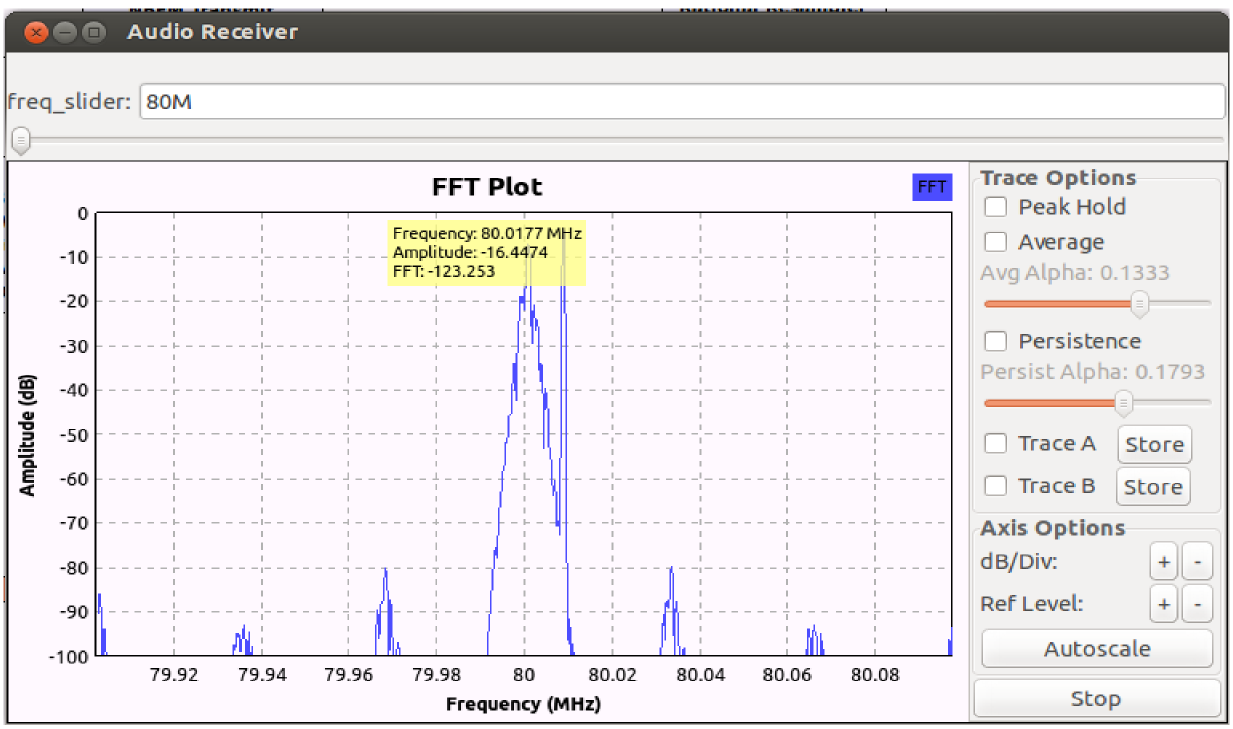

Figure 4: FFT of received FM Audio Signal

ADDITIONAL OBSERVATIONS:

I. We observed that the location of experiment played an important role in the quality of signal received. First it was carried out very close to a Wi-Fi access-point and we saw a lot of side-band disturbance in the reception. We tried different filtering techniques but couldn't alleviate the issue. When the same experiment was later on carried out in the open ground, we received excellent reception. We presume this was because of interference in the 2.4GHz frequency band at which both USRP as well as Wi-Fi operate. Hence, we conclude that inter-carrier interference plays a significant role in signal reception and should be carefully studied.

II. Also, when this test was carried out in the library lobby, with lot of noise a movement, there was disturbance in the reception. This is in coherence with the theory that the environment has an effect on signal reception quality.

FUTURE WORK:

I. Analyze and compare other modulation schemes suitable for video transmission. This would aid towards completion of the last leg of this experiment that is video transmission over the RF.

II. Also the maximum distance within which USRPs can communicate can be explored. We are of the view that distance between the two USRPs plays an important role. Increasing distance between the transmitter and receiver will result in degraded quality. But it would be of interest to quantify the exact effects.

III. The frequency tuning functionality is carried out manually. This work could be extended to make it automatic. We planned to it by using a python script to obtain the peak amplitude of the signals over the known frequency range and using the peak value as the center frequency.

CONCLUSION:

Through our experiments and observations, we arrived at the following results:

I. Frequency tuning is critical to RF communications

II. Quality of communication is affected by the distance between the USRPs and the environment.

III. Modulation schemes should be carefully chosen based on the application used.

REFERENCES:

[1] http://www.sbrac.org/files/build-gnuradio

[2] http://en.wikipedia.org/wiki/Universal_Software_Radio_Peripheral#Applications

[3] http://en.wikipedia.org/wiki/Software-defined_radio

[4] Yan, Zhi, et al. "Spectrum sensing, access and coexistence testbed for cognitive radio using USRP." *Circuits and Systems for Communications, 2008. ICCSC 2008. 4th IEEE International Conference on*. IEEE, 2008.

[5] http://www.commsys.isy.liu.se/SDR/labs/gr/lab1.pdf

[6] http://www.commsys.isy.liu.se/SDR/labs/gr/lab2.pdf

[7] http://www.commsys.isy.liu.se/SDR/labs/gr/lab3.pdf

[8] http://radioware.nd.edu/documentation/hardware/the-usrp-board

[9] http://asim.csail.mit.edu/redmine/projects/airblue/wiki/USRP2

[10] http://code.ettus.com/redmine/ettus/projects/uhd/wiki/UHD_build

[11] http://people.bu.edu/mrahaim/NEWSDR/Presentations/NEWSDR_Kelly.pdf

[12] http://lists.gnu.org/archive/html/discuss-gnuradio/2012-01/msg00069.html

[13] https://sites.google.com/site/byungchulweb/setup-usrp2

[14] http://www.wu.ece.ufl.edu/projects/wirelessVideo/project/GNU_Radio_USRP/how_to_test_USRP.htm

[15] http://www.wu.ece.ufl.edu/projects/wirelessVideo/project/GNU_Radio_USRP/

[16] http://rfpoweramp.wordpress.com/2013/02/08/setting-up-the-usrp-n210/

[17] http://files.ettus.com/uhd_docs/manual/html/usrp2.html

[18] http://code.ettus.com/redmine/ettus/projects/uhd/wiki/UHD_Linux

[19] http://code.ettus.com/redmine/ettus/projects/uhd/wiki/GNU_Radio_UHD