Transmission & Reception of Audio files using OFDM signals with USRP

Simegnew Getiye, Estifanos

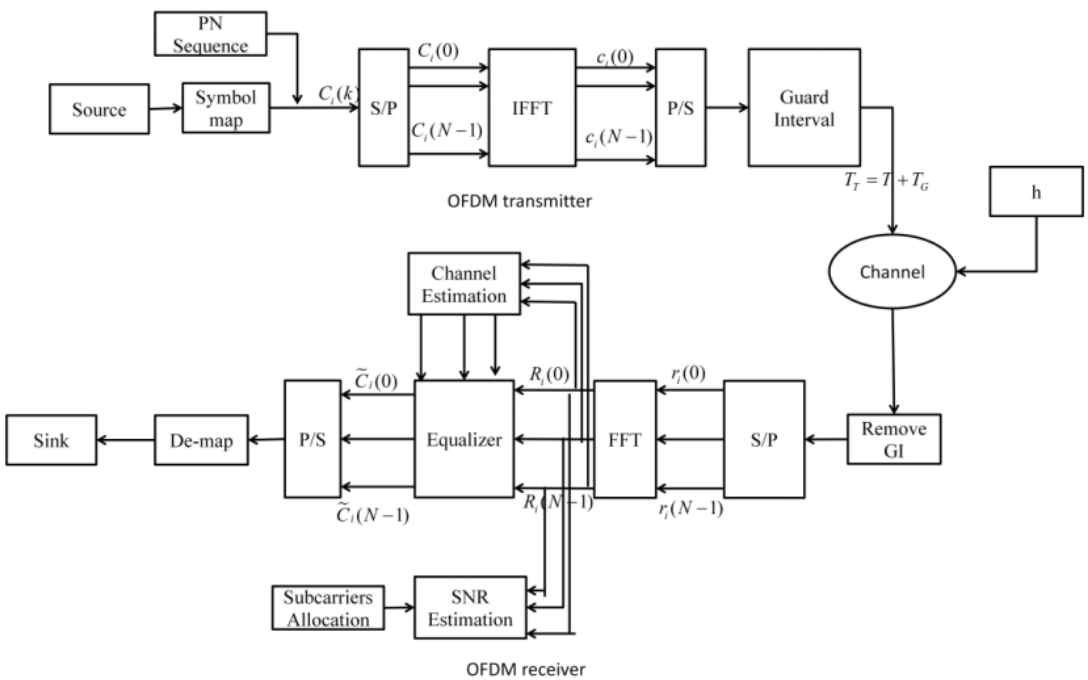
Joshua Tetteh Ocansey

Fangzou Thierry ludovic

February 2016

1. Introduction

OFDM stands for Orthogonal Frequency Division Multiplexing, and it is the modulation used for the data transmission. With OFDM, data is transmitted in parallel on multiple, orthogonal subcarriers. It is a multiplexing method, which means that different data channels share the bandwidth available. The main idea of OFDM system is to separate the single high rate data stream in to N parallel low data sub-streams that are modulated on to N orthogonal sub carriers. This process is simply made in the discrete time through an N point Inverse Discrete Fourier Transform (IDFT) or Inverse Fourier transform (IFFT) element and the resulted signal is transmitted in sequence. The information at the receiver is retrieved by performing a DFT/FFT unit.



Software Defined Radio (SDR) is where all the signal manipulations and processing works in radio communication are done in software instead of hardware. Universal Software Radio Peripheral (USRP) is a flexible low-cost platform for SDRs. GNU Radio is an open source software toolkit which consists of signal processing blocks library and the glue to tie these blocks together for building and deploying SDRs. The USRP will digitize the inflow data from the air and passing it to the GNU Radio. GNU Radio will then further process the signal by demodulating and filtering until the signal is translated to a packet or a stream of data.

Figure 1. The group

In this experimental assignment we tried to transmit an audio file using OFDM modulation with the involvement of the above described tools.

2. Set up

The equipment and software that has been used for this test are:

Hardware :

- A PC used has 8Gb of RAM memory with a core i5 processor at 2.45 GHz
- A PC used has a double core Intel Core 2 Duo processor at 2,33 GHz.
- Two Gigabit Ethernet network interfaces connect two USRP2 devices in the ISM band 2.4 GHz -6 GHz and 0.4 GHz-4.4 GHz.

Software:

- A GNU/Linux PC with the Ubuntu 14.04 (Trusty Tahr) distribution.
- GNU Radio software

3. Description

The whole system comprises two main blocks of transmitter and receiver. These blocks are more or less similar except some elements in the receiver part.

3.1 The transmitter

This block consists of audio file source, audio encoder, CRC & header adder, symbol mapper, FFT (Reverse) or iFFT , cyclic prefix adder, amplifier and file sink. Our main goal is to transmit audio file so that we use .WAV file as a source which can be implemented in GNURADIO easily. The input audio file is encoded in the audio encoder and CRC with header is added to the payload data that will be used to verify the validity of the received frame. The payload is repacked according the choice of modulation. Our choice of modulation was QPSK so 2 bits are being repacked together. Both the payload and the header are then mapped in to a vector of complex constellation of QPSK and BPSK respectively. The OFDM carrier allocator allocates occupied carriers, pilot carriers, pilot symbols and synchronization words. The FFT (Reverse) or iFFT takes a vector of complex values and computes the iFFT and it represents the output. The cyclic prefix takes OFDM symbols as its input, resulting in an output symbols with cyclic prefix.

The OFDM symbols are then amplified by a multiplication with a constant value. Finally sent to the receiver through air channel using the UHD (USRP Hardware Driver).

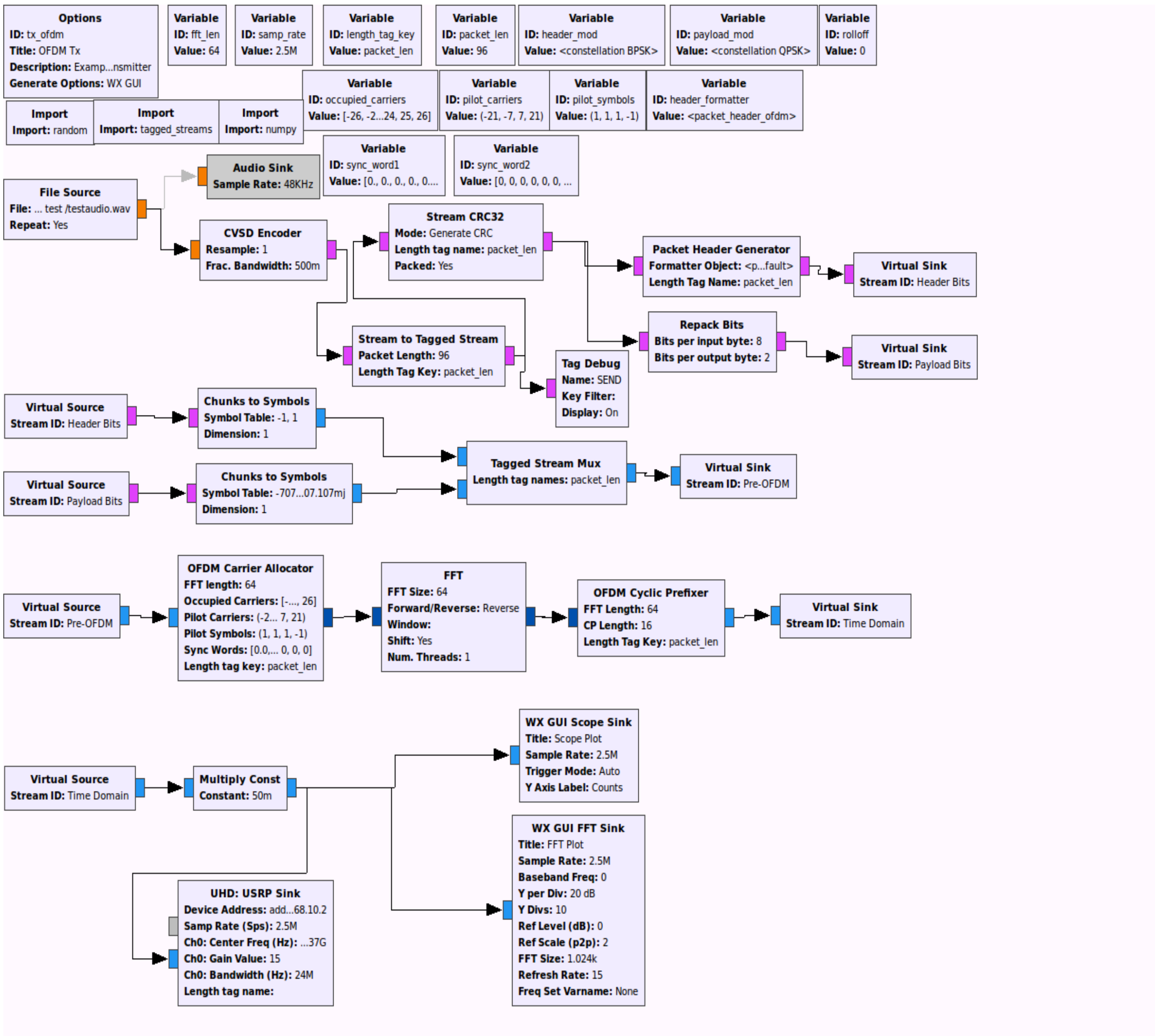


Figure 2. OFDM Transmitter

3.2 The Receiver

The receiver mainly consists of OFDM synchronizer, frequency modulator, signal mixer, FFT and frame acquisition.The input signal coming from the antenna that takes the bandwidth corresponding to the number of carriers that contain actual data is directly transmitted to the synchronizer(Schmidl and Cox). This synchronizer is responsible for making timing synchronization and frequency error correction. The frequency error correction is fed to the frequency modulator to generate a signal proportional to the frequency error of the sync block. Then it is mixed with the received data to correct the error and input to the de-multiplexer.

The de-multiplexer start to receive the data once it gets the beginning of the packet and output the header and the payload for demodulation.

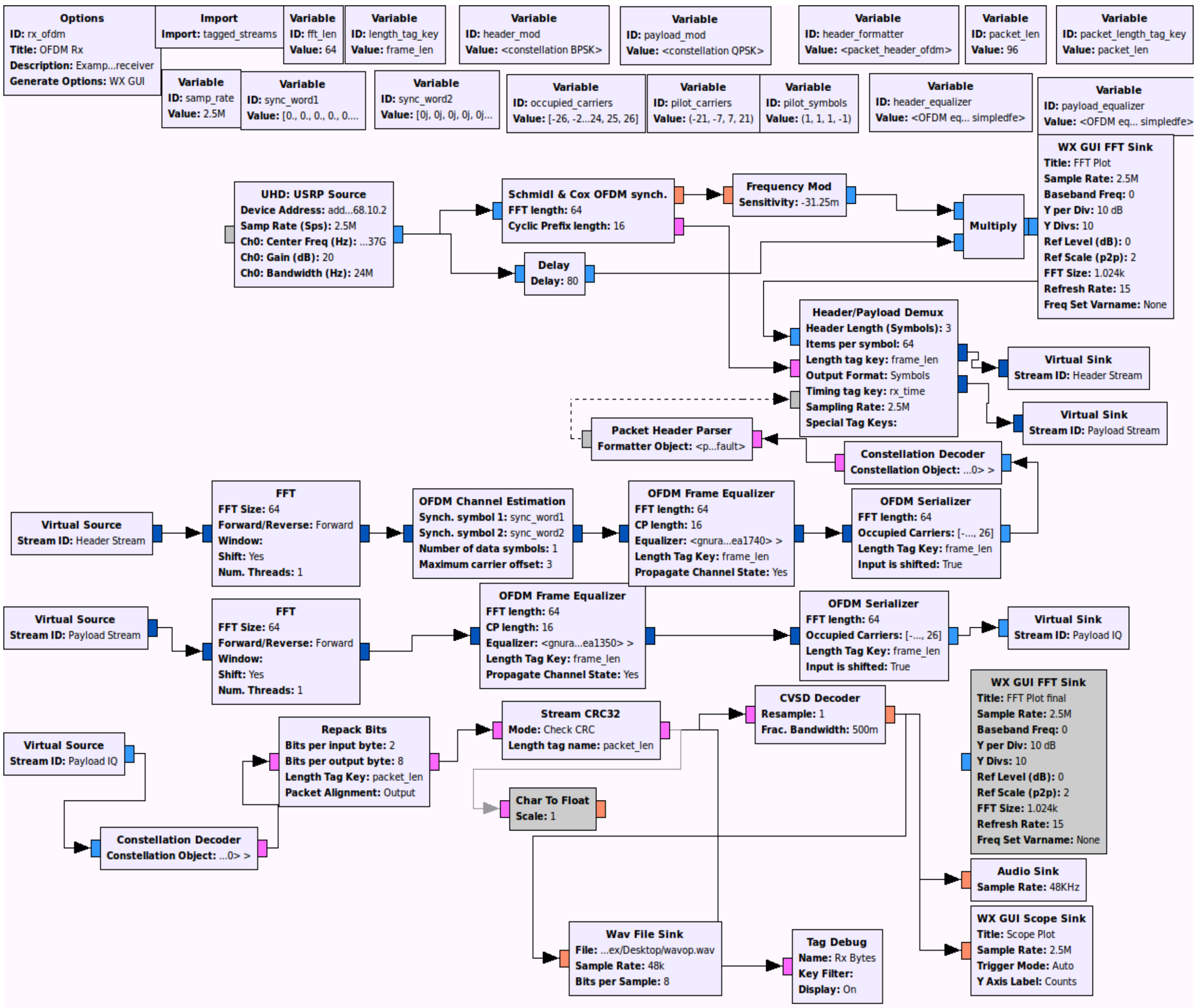


Figure 3.The OFDM Receiver

The FFT then takes a vector of complex values outputted from the Demux with FFT size of 64 and computes the FFT. Channel estimation and equalization blocks follow the FFT blocks, which are responsible for correcting the phase & amplitude distortion caused by the channel and search for beginning of the frame & will let the preamble through respectively. The symbols will be De-mapped, repacked and check the bytes corresponding to the header of the data and sent to the file sink or audio player.

4. Challenges

The first challenge we faced was to make the gnuradio software and the N210 device communication. After that we were unable to send and receive the correct data. The data we were receiving was corrupted data. It was due to the gain of the transmitter which was zero. The processing capacity of our laptop was also one constraint.

Audio files with different sampling rate than our audio sink did not played well unless is compatible with it. The other challenge that we faced was receiving a discontinuous audio file. We found out that transmitting with the audio file source ON, result in discontinuous received file in the receiver part due to sampling and processing mismatches.

5. Results

We used the benchmark OFDM example from the gnuradio software as a starting point and change parameters according to our need. We set up our center frequency to 2.45G for device communication. With this frequency we are able to detect a signal transmission from our GUI. The FFT size was set to 64 and cyclic prefix of 16. The modulation used was QPSK for the payload and BPSK for the header as it is light data. The channel bandwidth were set based on the information found on the device to 24M. The sampling rate for the UHD was set to 2M.

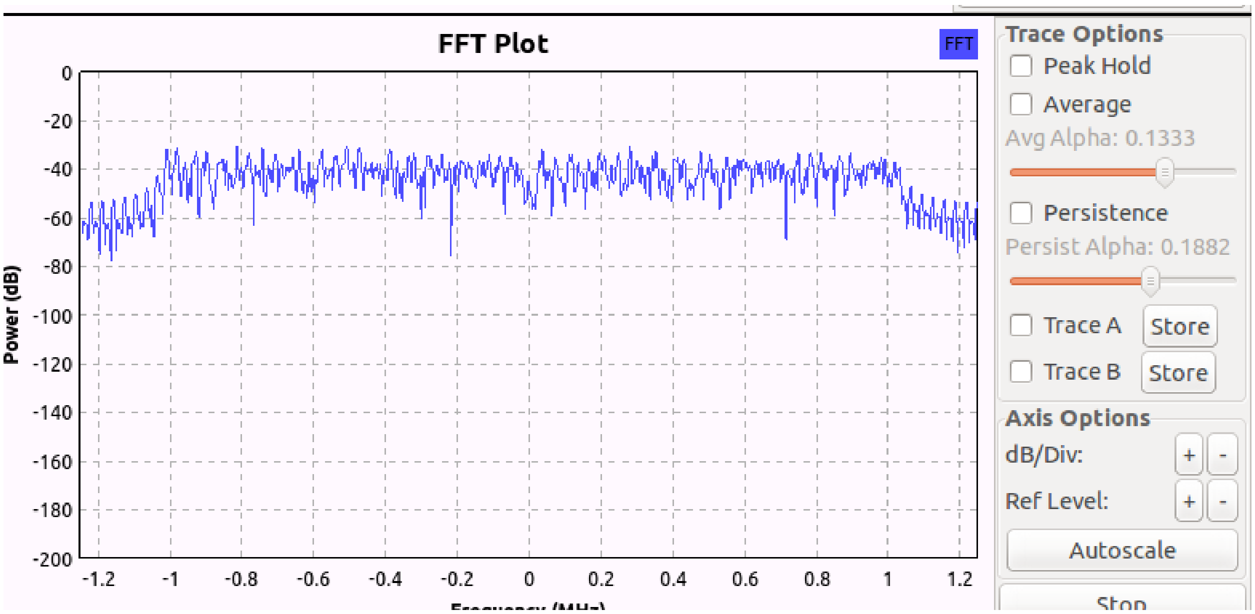


Figure 4. Transmitted signal

Without using audio encoder in the first step we tried to transmit a text file. And it was fine. But it was not as simple as we expected it in the audio transmission. First of all the gnuradio supports .wav file. Thus we need to prepare this type of file with a known sample rate by recording or converting using audacity media tool.

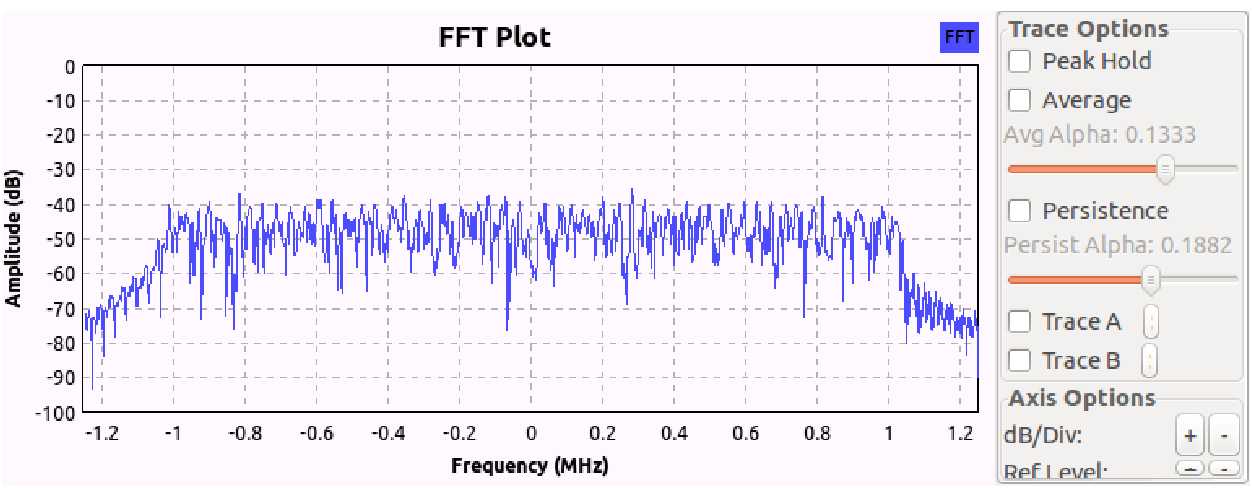


Figure 5. Received signal

We use CVSD (continuously variable slope delta modulation) voice coding method as an encoder. We first tried to send and store the received audio file and were successful. Then we tried both receiving/playing live and storing the file at the same time which was fine as long as the received file sampling rate and the audio sink is the same.

6. Conclusion

We are able to transmit an audio file using OFDM modulation. It was a great moment of learning to see what has been taught theoretically to be implemented in practice. We have seen that OFDM is very error sensitive modulation that needs a very careful celebration for transmission with regard to frequency. And also understand that this tool we have used is very nice for further wireless transmission practical investigations.