2018 3rd International Electrical Engineering Conference (IEEC 2018)
Feb, 2018 at IEP Centre, Karachi, Pakistan

Software Defined Radio Waveforms implementation on GNU Radio

Atif Javed, Abdul Samad, Faisal Saleem, Wajid Gulistan
Research and Development Department
National Radio and Telecommunication Corporation Haripur, Pakistan
{atif.javed,abdul.sammad,faisal.saleem,wajid.gulistan}@nrtc.com.pk

Abstract—A Software Defined Radio is comprised of both software and hardware that can be dynamically reconfigured to enable communication between a wide variety of changing communication standards, protocols and radio links. It is a rapidly evolving technology that has generated a widespread interest all over the telecommunication industry. SDR technology facilitates implementation of some of the functional modules of a radio system such as modem, up/down converter, source and channel encoders/decoders and data link-layer protocols at a software level. Waveform development is one of the most crucial development part of an SDR. By Loading different waveforms on a single SDR one can switch between multi-mode, and multiband wireless devices as well as be able to communicate with legacy radios and compatible with a modern Internet protocol Adhoc router in self-healing and self-forming data networks. Using GNU Radio platform, an analog (AM/FM) and a digital OFDM waveform is implemented with selection of different parameters like secure/non-secure, codecs, and modulation order along with an RF hardware to work in a real time environment. Output of each waveform is tested in a real time environment (GNU radio) in simulation using multiple RF front end hardware and, on a spectrum, analyzer is observed that is in close conformity with theoretical approximation.

I. INTRODUCTION

A. Software Defined Radio

A Software Defined Radio (SDR) is a radio that is substantially defined in software and whose physical layer behavior can be significantly altered through software. In SDR's a large portion of the functionality is implemented through software. This approach increases the flexibility of the device thereby changing its operating parameters and addition of new features without any physical modification to the system. Many analog and digital waveforms have been developed so far for Software Defined Radios that provide different functionalities in different testing environment. The main difference lies within the focus of implementations, some waveforms are designed which achieved higher data rate, while others prioritize range and bandwidth. Analog waveforms are an integral part of SDRs because of their ability to provide a very good performance over a bad channel as well as interoperability between different radio platforms. Digital waveforms on the other hand provide encryption, adaptability, efficient spectrum usage and greater user control [4]. Software Defined Radio can be simulated in real time environments such as Gnu radio, RED-HAWK along with other external hardware like USRP, LimeSDR [10].

B. Motivation

Radio is essential for communications over large bodies of water, territory controlled by enemy forces and terrain where the construction of wire lines is impossible or impractical. Generally, in electronic warfare, radios are used for espionage, reconnaissance, tactical operations, airborne and maritime communications, electronic counter measures (ECM) and electronic counter-counter measures (ECM). Our focus of implementation is providing common analog and digital waveform support in a single GUI for beginners and researchers in the domains of Software Defined Radios. It involves a complete working flow of transmitter and receiver modules along with RF hardware with user customization ability. The objective is to implement Amplitude Modulation(AM), Frequency Modulation(FM) and Orthogonal Frequency Division Multiplexing(OFDM) waveforms on GNU radio allowing users to control functionalities like modems and codecs, RF tuning and channel control with multiple hardware.

II. LITERATURE REVIEW

A. Analog Modulation

Analog Modulation are the most primitive modulation schemes where the message signal is embedded in amplitude (A.M), frequency (F.M) and phase (P.M) of the carrier signal. For Amplitude Modulation, the entire spectrum shifts around a high carrier frequency which are then recovered by multiplying again with the carrier frequency and filtering the resultant through a low pass filter. Amongst A.M, the popular techniques are Double Sideband Suppressed Carrier (DSB-SC), Single Side Band(SSB) and vestigial sideband. On the other hand, frequency modulation schemes constitute Narrow Band FM and wide band FM with channel spacing of around 20KHz and 200KHz respectively.FM signals can be recovered using a frequency discriminator that transforms the continuously changing frequency into amplitude modulation and then applying an envelope detector. Analog modulations can be coherent or non-coherent depending upon whether the same carrier be used on the receiving end or not.

1) Digital Modulation: The main advantage of digital communication over analog is that it tends to be far more resilient and immune to transmission and interpretation errors than information transmitted over an analog medium. we need far less information to transmit digital data and information can be easily regenerated at the receiving thus making the spectrum usage efficient.

a) Orthogonal Frequency Division Multiplexing(OFDM): For digital modulation scheme we used Orthogonal Frequency Division Multiplexing (OFDM) which is inarguably the most popular technique of multi carrier transmission. The idea is to utilize several carriers, spread regularly over a frequency band orthogonally so that the available bandwidth is utilized to maximal efficiency, sub carriers overlapping in frequency is allowed due to orthogonality which makes them independent of each other. High rate data stream is divided into many low rate data streams that are transmitted over several multiplexed orthogonal sub carriers [5]. These low rate data streams allow addition of sufficient guard time between two symbols which would have been very small in high rate data stream. This enables the system to perform well in dispersive channel which in turn causes the symbols to spread in time and interfere with each other because of inter symbol interference(ISI) [3].

III. SYSTEM DESIGN

1) Design Parameters: The main aim of the radio is to transmit audio voice over a wireless channel to a considerate distance.The human audible frequency varies from 20 to 3400 Hz wheras the audio file was sampled at 4800 KHz. The overall sampling rate was chosen to be from 8-48 KHz as per the requirements.For ground to ground communication we selected VHF/UHF band that is supported by RF hardwares at our disposal. Furthermore, for SDR testing platform we needed to provide different modulation scheme , Codecs type, and a secure communication channel (encryption) over a digital modem as well as provide bandwidth efficient sidebands of analog modulation detail of which are shown in Table I.These parameters directly complied with our communication requirements and provided a smooth and reliable communication over a wireless channel.

TABLE I: System Specifications

Operating Frequency	UHF/VHF Band
Analog Modulation	AM(DSB/LSB/USB) FM(NBFM,WBFM)
Digital Waveform	OFDM(BPSK,QPSK,8PSK),SC=512,CP=128
Channel Selection	Manual
Encryption	AES 256
CODEC	GS721, CODEC2, CVSD
Media type	Audio Source, Wave File Source

2) Implementation Flow: Figure 1 shows the implementation flow for the project where the signal processing block can serve for both analog and digital waveform and RF block can be replaced with USRP,LimeSDR blocks.

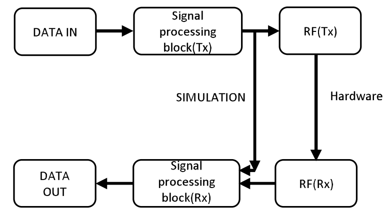

Fig. 1: Generic implementation flow

IV. IMPLEMENTATION

A. Software and Hardware Selection

For implementation we have utilized Gnu radio as it supports multiple hardware platform and facilitates programing via easy to use blocks. It has a much less latency and low power consumption. The sampling rate and the data rate can be varied easily and on runtime. Gnu radio possess much less synchronization issues and provides users with a variety of signal processing block. In addition to that it supports custom blocks built via C++/Python scripts [13].

TABLE II: Comparison of RF hardware

Hardware	**LimeSDR**	**USRP(SBX)**
Frequency range	100KHZ-3.8GHZ	400-4400 MHZ
Supported bandwidth	61.44 MHZ	50 MHZ
TX power	Up to 10dbm	Up to 20dbm
Channels(TX, RX)	2	1
Interface	USB 3.0	Gigabit Ethernet
ADC	12-bits 40 MS/s	14-bits 100 MS/s
DAC	12-bits 40 MS/s	16-bits 400 MS/s

To provide RF front end to our waveform we have used both USRP N200 [12] and Lime SDR [14]. To observe real time simulation results, we need RF hardware that act as a bridge between host PC and wireless medium via antenna through proper RF/IF conversion as per decide parameters. Both USRP and Lime SDR can fulfill our communication requirements and are stated Table II with their characteristic parameters.
Lime Suite is a collection of software supporting drivers with several hardware platforms including the LimeSDR, drivers for the LMS7002M transceiver RFIC, and other tools for developing with LMS7002M-based hardware. Installing the Lime Suite enables many SDR applications such as GQRX to work with supported hardware through the bundled SoapySDR support library.In addition to that, it adds support for all SDRs that are supported by gr-osmosdr (LimeSDR) and Ettus uhd(USRP) via the Soapy Osmo and Soapy UHD Drivers. Figure 2 shows the flow of driver and dependencies installations.

B. Waveform Implementations on GNU Radio

Waveforms are the true essence of SDR radio as they drive a radio to operation. Without a waveform an SDR cannot be made to operate regardless of its hardware specification [**?**]. A waveform constitutes all of the signal processing

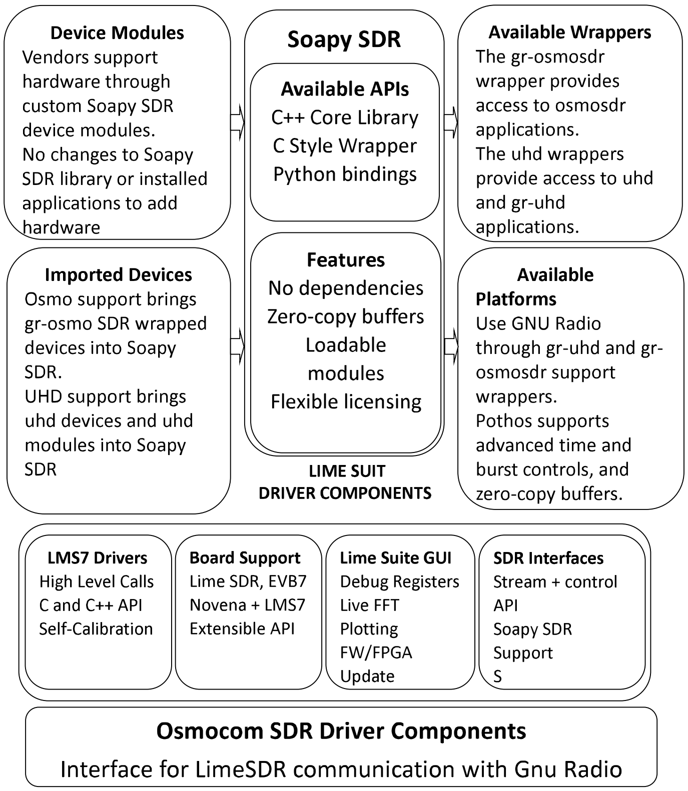

Fig. 2: Lime SDR setup with GNU

function performed in the software part including carrier generation, modulation, encryption, filtration and correction e.t.c. Different waveform can be implemented to achieve high data rate, efficient spectrum utilization and good range as per the user demands [1]. The main idea of the project was to implement all of the waveform modules under a single user interface so that the user can varies features at run time and investigate the effect of multiple vocoders, with encryption(AES),and on different modulation schemes which shown in Figure 3. Moreover the parameters like sampling rate and RF frequency provides a higher degree of optimization at the output for different modulation schemes. The GUI begins from the choice between analog and digital waveform. Analog portion constitutes simple AM and FM waveforms with further customization at USB, LSB, CW,NBFM and WBFM consequently. The digital waveform on the other hand follows a complete sequence from Vocoders to encryption, FEC Correction, Digital Modulations and OFDM modem. The user can decide whether he wishes to transmit live audio or a preset audio file and can switch between these modes at runtime. A virtual graphical knob embedded within the GUI provides user with the option of channel selection.A push to talk enable is implemented using python script and tested between transmitter and receiver. Additional features also include Squelch, AGC and rational re-samplers at different implementation levels of the program. All of these operations are then sink to RF terminals which are provided by blocks like UHD and Osmocom in case of USRP and LimeSDR respectively [11].

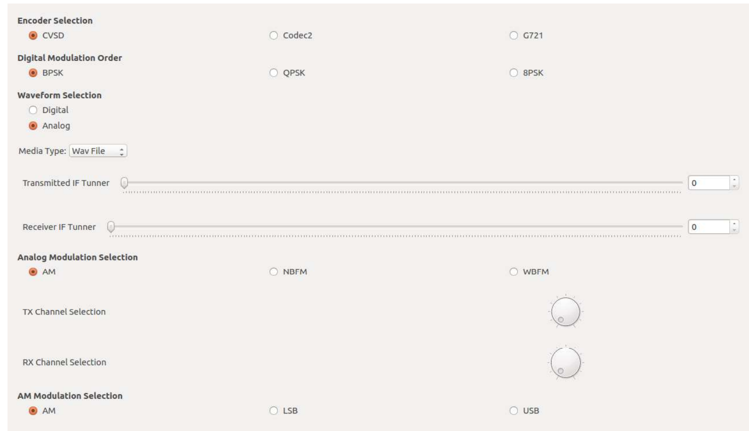

Fig. 3: Waveform Platform GUI

1) Analog Waveform Implementation: Analog Waveforms need to be implemented as they provide high quality, backward compatibility and user friendly interfaces in tactical operations.In theses waveform a voice or an audio signal is transmitted via continuous signal varying in amplitude(A.M), Phase (Phase Modulation) and frequency (Frequency Modulation).

Figure 4 shows a message signal whose bandwidth is approximately 24kHz because we sample the voice data at 48kHz so according to Nyquist theorem B=Rs/2 where B is bandwidth and Rs=sample rate. This signal serves as the input of the AM/FM modulation.

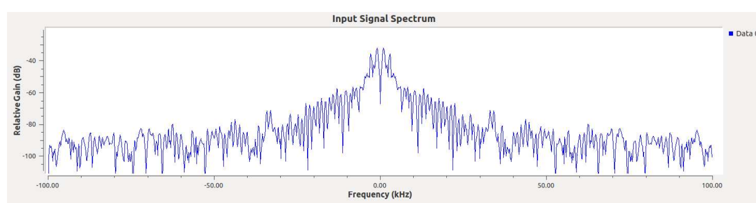

Fig. 4: Input Signal Spectrum

As message signal is shifted to a higher frequency carrier (IF=100KHZ) Figure 6a spectrum represent the AM Double side band modulation in which information is embedded into carrier signal and bandwidth becomes twice of the message signal which is B=2fm. Figure 6b-c shows the lower side band and upper side band of the signal that contain half of the bandwidth of single side band of modulation.

Amplitude Modulated Signal real time spectrum shown in Figure 5 that is shifted to passband frequency of 800MHZ and its bandwidth is roughly greater than 200KHZ.

Figure 7 indicate FM modulated signal output of Wide band and Narrow band.WB harmonics is at infinity, so we cannot depict the exact bandwidth of WBFM but in general we assume 200kHz Bandwidth in for NBFM and 25KHz for NBFM modulation respectively.

Demodulated signal spectrum shown in Figure 8 for double,lower,and upper side band respectively.

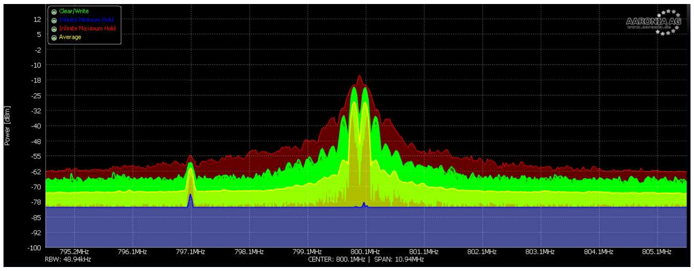

Fig. 5: Amplitude Modulation Real Time Spectrum

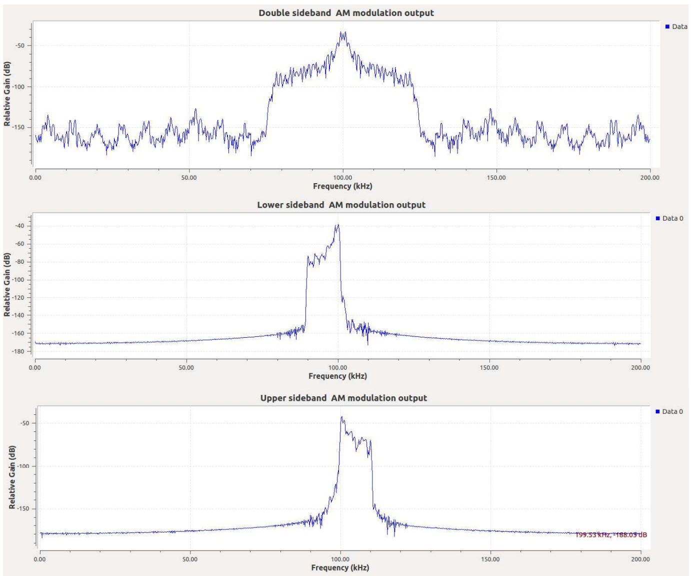

Fig. 6: Amplitude Modulated Signal

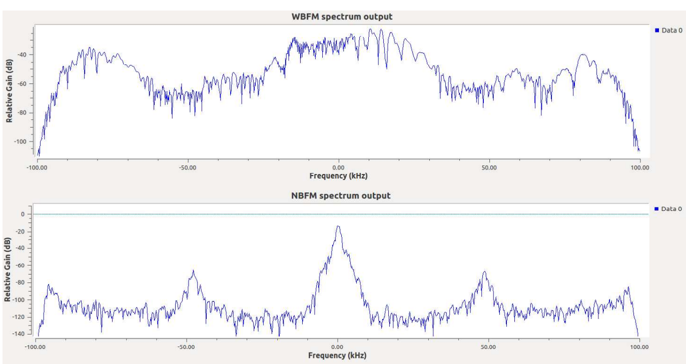

Fig. 7: Frequency Modulated Signal

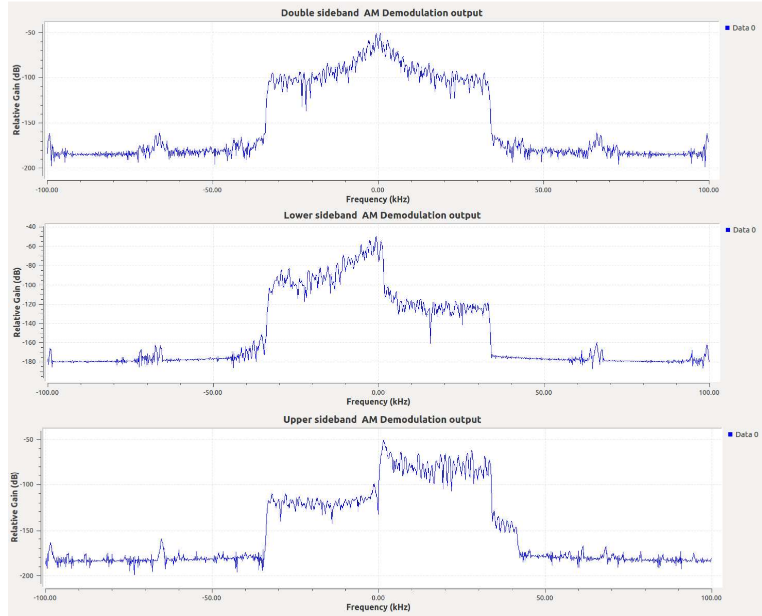

Fig. 8: AM Demodulated Signal

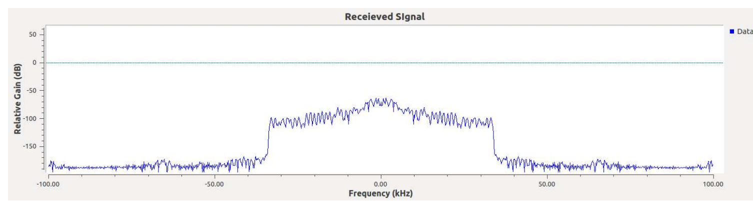

Fig. 9: Demodulated Received Signal

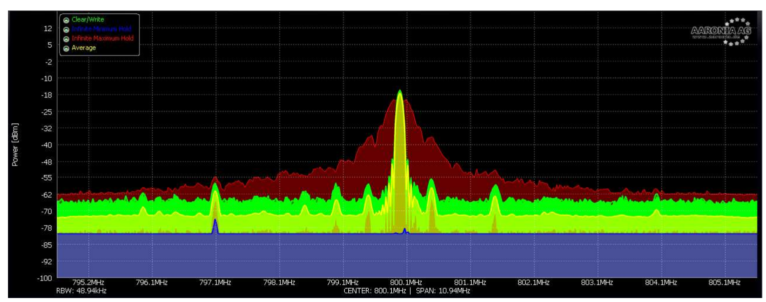

Fig. 10: Frequency Modulation Real Time Spectrum

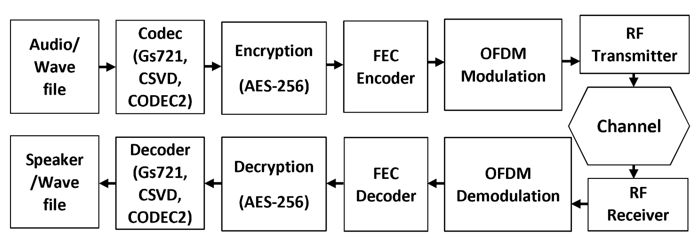

Fig. 11: Digital Modulation implementation Flow

Figure 9 shown received signal which is recovered from wireless medium after demodulation either for digital or analog mode.

Frequency Modulated real time signal spectrum shown in Figure 10 which has center frequency is at 800MHZ and its bandwidth is around 200KHZ.

2) Digital Waveform Implementation: In digital implementation firstly the input samples taken from the wav file or audio source are digitized to bits and then sent to compression through one of the codecs. We used multiple codecs, each codec takes input as audio bit stream and compress signal using sinusoidal coding which then recreated by modeling speech as a sum of harmonics related to sin waves. Codec2 is an open source speech codec design for HF/VHF digital radios.It is tested with multiple data rate like 3200,1600,700 each in bits/s .G721 is ITU standard codec that utilizes Adaptive differential pulse-code modulation (ADPCM) to produce a digital signal in case of speech codec, output is 32kbits/s.Another lossy compression codec (Continuously variable slope data)CVSD is employed whose quality is measured by sampling frequency.It is form of delta modulation and is changed continuously to minimize slope-overload distortion. Furthermore codec output bits stream will be given as input of AES encryption block and then bits will be shuffled in organized manner with a specific Encryption key [7]. These encrypted bits hand over to Forward Error Correction block that adds capability in bits to detect and recover error at receiver end and converted to parallel bit streams depending upon the number of sub-carriers (N). These bit streams become the input of digital modulation block (BPSK, QPSK, 8PSK) where they are subjected to IFFT module with the addition of cyclic prefixes. The OFDM symbols are sent to transceiver where they are transmitted over a wireless channel. The cyclic prefix of length L is a circular extension of the IFFT-modulated symbol, obtained by copying the last L samples of the symbol in front of it. The Transceiver finally performs the steps of digital up conversion (DUC), analog to digital conversion and RF/IF conversion. At receiver the above stated operations are performed in reverse for example the received signal is firstly down converted to lower frequency signal and cyclic prefix is removed. The coherent FFT demodulator will ideally retrieve the exact form of transmitted symbols. Finally, digital demodulation, Decryption, decoder extract our desired data that would serves as the input to the speaker. Figure 11 shows the complete transmitter and receiver implementation flow. The Figure 12 shows the OFDM baseband spectrum of the

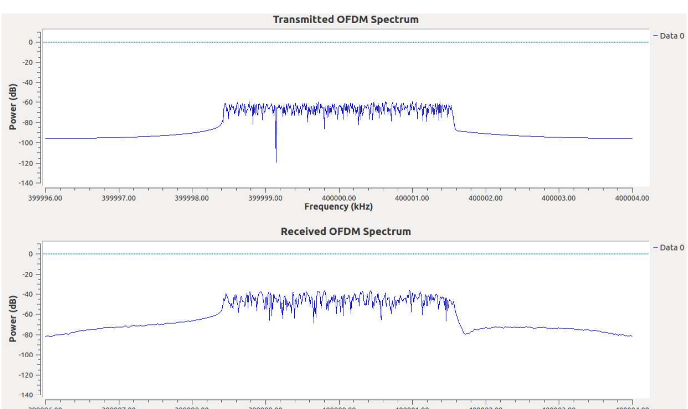

Fig. 12: OFDM Spectrum at Transmitter and Receiver

transmitted signal after codec, encryption, FEC Correction and QPSK modulation and then received signal before the demodulated process. The Figure 13 shows the OFDM spectrum of received signal before demodulation, FEC correction, decryption and codec decompression.

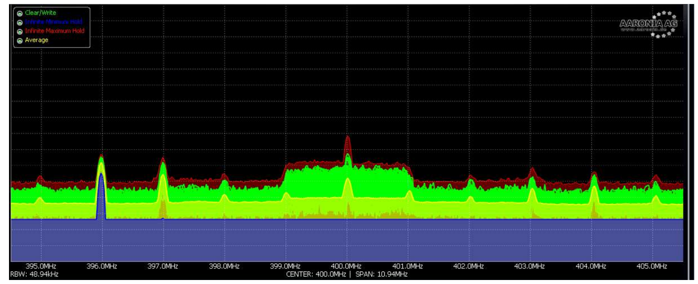

Fig. 13: Real Time OFDM Spectrum

V. Conclusion

A waveform testing platform for Software Defined Radio was successfully implemented on GNU radio using USRP and lime SDR. The waveform was tested for voice input and audio wave file and voice quality was ensured in different schemes. Moreover, hardware results are observed on a spectrum analyzer to confirm bandwidth and other physical parameters to theoretical approximations. Many control features like channel selection, codec selection and frequency tuning were successfully implemented and tested in a real time environment.For SDR designers and developers this implementation play an integral role as they can visualize real time effect of changing different parameters via Software GUI.

Acknowledgment

Sponsored by Research and Development Department of National Radio Telecommunication Corporation

References

[1] Guan. Jianxin,Ye. Xiaohui,. Gao Jun. Liu Quan *The General Waveform Development Flow of Software Defined Radio*. Proceedings of International Conference on Consumer Electronics, Communications and Networks (CECNet),XianNing, April 16-18,2011.

[2] Duc. Toan. Nguyen,*Implementation of OFDM system using GNU Radio and USRP*,2013

[3] M. Litwin,*Orthogonal Frequency Division Multiplexing (OFDM Transceiver Design*,2010

[4] Jeffrey H.Reed,*A Modern Apporach to Radio Engineering*,2002,pp.1-53.

[5] M. Nathan,*OFDM Implementation Details* [online].Available: https://www.gaussianwaves.com/2011/06/introduction-toofdm-orthogonal-frequency-division-multiplexing-part-3

[6] *Codec2 implementation* [online].Available:http://www.rowetel.com/?page_id=452

[7] *AES block Installation* [online].Available:https://github.com/capitanx/gr-aes

[8] *Digital Modulation* [online].Available:http://ee.eng.usm.my/index.php/ms/

[9] *LimeSDR driver flow* [online].Available:https://github.com/pothosware/SoapySDR/wiki

[10] *SDR and hardware detail* [online].Available:http://www.sdr-radio.com

[11] *GNU Radio Tutorials* [online].Available:https://github.com/gnuradio/gr-tutorial

[12] *USRP Specifications* [online].Available:https://www.ettus.com/product/details/UN200-KIT

[13] *GNU radio c++ implementation*[online].Available:https://gnuradio.org/doc/sphinx/

[14] *LimeSDR specifications* [online].Available:https://www.crowdsupply.com/lime-micro/limesdr