

Software Redefined Communication System

Neethu Prasannan, Jyothish Lal G, Roshni Das, K.P Soman

¹Centre for Excellence in Computational Engineering & Networking Amrita Vishwa Vidyapeetham

Abstract: "Beginning with practical difficulties in teaching communication systems in class room, this paper describes a set of innovative experimental demonstrations developed using SDR". Communication engineering is one of the interesting, at the same time difficult subject to learn if the concept is not clear or well explained. Normal practice is that the faculties will just give a theoretical class on communication systems and for students the various communication processes like filtering, modulation, demodulation etc. are just imaginary things. Giving a clear idea about these things to a graduate and under graduate student is a little difficult task. The main aim of SDR is to create an attractive learning platform for the students where they are freed from the boring routine of theoretical learning. We are challenging current education system to think outside the "board", where students will feel excited to learn something new so that they can physically feel and have a real time experience, which would be quite difficult to forget, for which the expense required would be quite high. So in order to better our possibilities, we suggest this software which would equip them to visualize a solid image of what they are learning about. Another fact is that as this software is so simple and is easy to learn and is feasible for everyone who would like to have basics in electronics and communication. This software is worth a stepping stone for those who would dream about being a "real" engineer. This will boost engineering students having great struggle to understand and learn communication subjects. The soul of education system should transform from "cramming as you learn" to "see as you learn".

Keywords: Attractive learning platform, GNU Radio, SDR, USRP, Experimental demonstration.

I. Introduction And Background

Do we ever realize the importance of communication systems? We are getting connected to others whenever we want and on our own terms. Mobile phones have changed the way we live. Now the cell phones can not only allow us to call, but give us turn-by-turn directions and allow us to perform various actions like surf web, send mail etc. For communication we have various means like cell phones, Internet, satellite, radar systems etc.

We have started studying the basics of communications from our lower classes. But at that stage students are just getting a theoretical knowledge whereas the practical experience is acquired only at engineering levels. In almost all the engineering colleges communication labs are completely hardware based [3]-[5]. Before getting into lab sections, students should have a clear idea about circuit designing, which is quite difficult to attain. Implementing a hardware circuitry is a hectic job. Moreover, a communication engineer has to face a lot of problems in the lab. But if we have GNU radio, with a little knowledge in software, entire communication can be easily studied [1]-[2]. The normal practice in engineering colleges for doing analog/digital communication techniques is the usage of modulation kits, which are highly precise. Hardware implementation of modulation and demodulation circuits in breadboard has several drawbacks such as impedance mismatch, frequency tuning, accurate selection of parameters etc. Another thing is generated output is obtained in CRO, which is again a hard-ware, from which obtaining the idea about actual communication process is difficult, whereas SDR provides both time domain and frequency domain representation simultaneously. Also in conventional circuits debugging is difficult. But in SDR, wherever there is error the connection arrow appears red in color. So error analysis is quite easy. Another advantage of SDR is that we get used with real time signals as we can connect to an external hardware called as USRP (universal software radio peripheral). This device can capture signals of different frequency bands. Instead of a bunch of fixed circuits, everything can be bought into a single and alterable canopy called software.

Using GNU radio, a radio can be built by creating a flow graph where the main blocks are signal processing blocks implemented in c++ and the flow graphs are constructed and run in python. For dealing with SDR, the only thing required is a little knowledge in software. Apart from executing the preassembled blocks we can also create new blocks by C++ and python. Actually communication is nothing but mathematics. It is the hardware part making it complex for students. The basic processes in communication are analog and/or digital modulation schemes, amplification, filtering etc. Amplification of a signal is just scaling and modulation is simply multiplication. So with simple knowledge in mathematics we can handle entire communication system. SDR is one of the leading technology, not only for classroom purposes, but also can act as a key enabling technology for a variety of other reconfigurable radio equipment commonly discussed in the advanced wireless

markets such as adaptive radio, cognitive radio[11] and intelligent radio. Cognitive radio is one of the ongoing research topics, which will reduce cost of spectrum sensing.

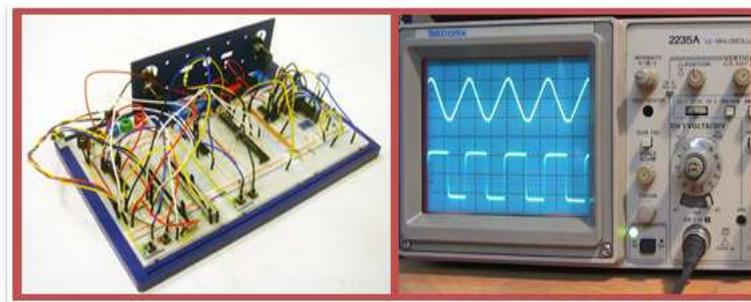


Fig. 1. Conventional method

One of the current issues in the communication system is “allocation of spectrum”. Radio frequency spectrum is a limited natural resource. In India, it is confined to 9 kHz to 3000 GHz. This confined region is used for about 40 different communication systems like fixed, mobile, broadcast, radio, satellite etc. and everything depends on how effectively we are using it. So obviously spectrum allocation is a challenging task. SDR technology promises a new way for spectrum sensing. Even a software programmer can simply access spectrum and this provides a clever use of it. The next section of this paper provides an overview of SDR and the characteristic that make it an ideal platform for classroom demonstration. It also includes its hardware and software details. This is followed by ‘How to work with GNU radio’. The next section presents experimental demonstrations followed by the comparison of conventional method with SDR. The final section presents conclusions and plans for future works.

Software defined radio (SDR) [6], [7] is a radio in which software defines signals, frequencies, modulation, demodulation, filtering etc. SDR design began in 1987, when the United States Air Force’s Rome Laboratory (AFRL) developed a programmable modem. The term software defined radio was introduced by Joseph Mitola III in 1991. It is a new technology for representing the entire communication system by software. SDR reflects the convergence of two dynamic technologies: digital signal processing and real time downloadable software run on very fast microprocessor. SDR has two subsystems, hardware and a software system. Hardware part act as an interface between digital and analog world and the whole signal processing is done by software. A number of hardware Systems like USRP, soft rock [10], flex radio systems, high powered software defined radio and others are available. Among this the commonly used is USRP [9] of which almost all the specifications are re-programmable. The antennas that can transmit and receive are connected to RF front end of USRP. There are two paths: one for transmission and the other for reception. Normally a USRP kit will have motherboard and daughter boards. One mother board can handle up to 4 daughter boards. FPGA act as the motherboard and the RF front end act as daughter boards. For doing the signal processing we require a software platform. Multiple software modules are available but the actual software being used is GNU radio, which is an open software defined radio. GNU radio provides a library of signal processing blocks for waveform generation and processing. PYTHON and C++ are the main programming languages used. Blocks are implemented in C++ and the connections between blocks are made by PYTHON. SWIG (Simplified Wrapper and Interface Generator) is used as the interface between C++ and PYTHON. A block contains various parameters which include the number of input and output port as well as the type of data flowing through each. All these parameters can be set by the user as per the requirement. Usually for working with software, we need a user interface. GUI used here is GNURADIO companion (GRC). It allows the user to interact with GNU radio signal processing blocks.

II. Installing Gnu Radio Companion Software Using CDE

CDE stands for Code, Data, and Environment. It is developed in Stanford university [14] by Philip J. Guo for automatically creating portable linux software. It solves all the problems that we usually encounter while installing GNU Radio companion software. Once we install GNU radio in a machine successfully, CDE allows us to copy the successfully installed software into any other similar (hardware + major version of OS) system and run. Main obstacle with GNU radio in making an entry into academia was that its intended primary users ECE students and faculty- were not comfortable with the OS (Linux) on which GNU radio practitioners work.

Installation was taking hours and most weak hearted students abandon the idea if the first attempt to install fails. Now it is a question of copying from a machine where it is working successfully. Pictorial representation for Automatic Packaging of Code, Data, and Environment is shown in fig.3.

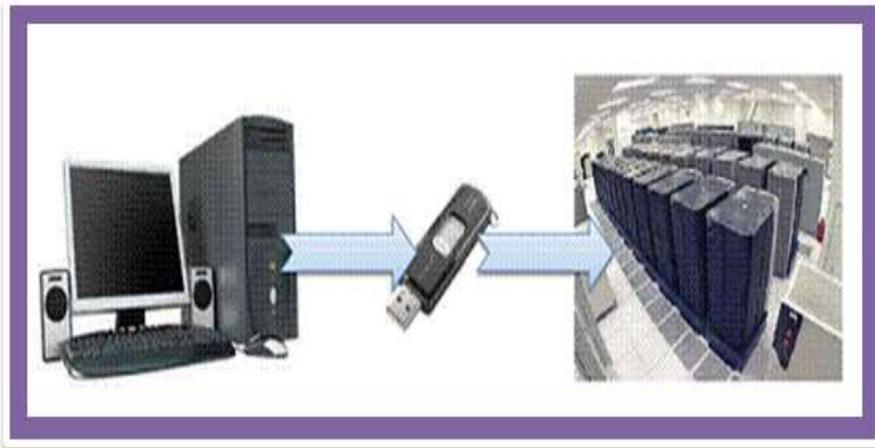


Fig. 3. Automatic Packaging of Code, Data and Environment

III. Working With Gnu Radio

For opening GRC, type “gnuradio-companion” in terminal and press enter. All the preassembled blocks will be available there. For doing any communication process, user has to create a flow graph just like building circuits in conventional method. We can drag or double click to add the blocks to workspace, and then connect the blocks just by clicking the output stream of one block and input stream of the other block. The privacy of changing the parameters like amplitude, frequency, sample rate etc. are up to the user. If there is any mismatch in the connecting link between blocks, it will show red color. Thus we can execute any blocks which are available; moreover we can create new blocks. Screen shot for Common representation of GRC window is shown in fig.4.

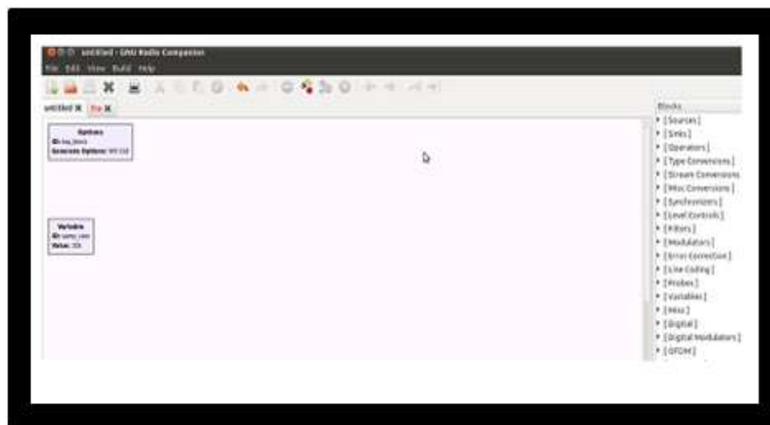


Fig. 4. GRC Window

IV. Experimental Demonstration

This section contains the discussion of six classroom experimental demonstrations in communication theory. The goal is to make the entire communication theory easy and liable from the complex hardware representations. The demonstrations simultaneously illustrate the time domain and frequency domain plots (spectrum). A spectrum represents the relative amounts of different frequency components in any signal.

A. Experiment 1: Basic signal generation

Any waveforms like sine, cosine, step etc. can be generated using SDR. The fig.5 depicts a sine wave generation.



Fig. 5. Sine Wave generation

B. Experiment 2: AM Modulation and demodulation

In amplitude modulation, the amplitude of a carrier signal is varied by the modulating voltage, whose frequency is invariably lower than that of the carrier. Formally AM is defined as a system of modulation in which the amplitude of the carrier signal is made proportional to the instantaneous amplitude of the modulating voltage. Consider a sinusoidal carrier signal as

$$C(t) = A_c \cos(2\pi f_c t) \tag{1}$$

An amplitude-modulated wave

$$S(t) = A_c [1 + k a_m(t)] \cos(2\pi f_c t) \tag{2}$$

$BW = 2f_m$ where maximum modulating frequency used. The procedure for generating an amplitude modulated signal is illustrated using GRC in fig.6. Fig.7 illustrates Time domain representation and fig.8 illustrates frequency domain representation.

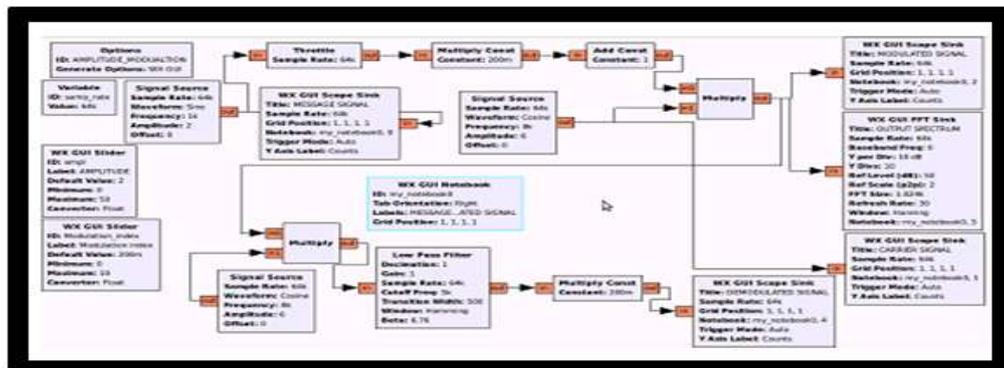


Fig. 6. AM Modulation and Demodulation Implementation

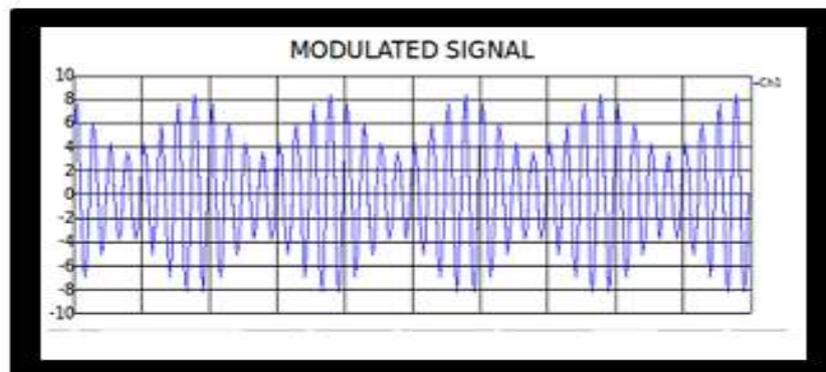


Fig. 7. AM Modulation: Time Domain Representation

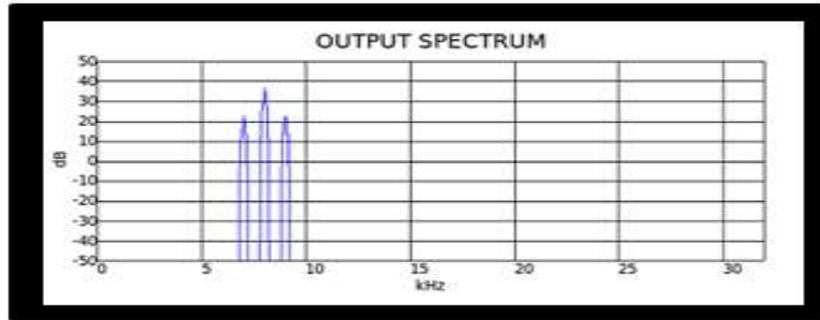


Fig. 8. AM Modulation: Frequency Domain representation

Spectrum realization get into a conclusion that the upper and lower sideband is mirror images of each other i.e. same information is present in both the sidebands. AM modulation is also termed as single or double side-band modulation techniques. The double side-band is same as above, where we use the whole spectrum. In single sideband modulation, either the lower or the upper band is filtered out. Only one half is enough to regain our message signal.

C. Experiment 3: FM Modulation and Demodulation

Frequency modulation is a system in which the amplitude of the modulated carrier is kept constant, while its frequency and rate of change are varied by the modulating signal. The frequency modulated wave is given by:

$$f = f_c(1 + kV_m \cos \omega_m t) \tag{3}$$

where f_c -unmodulated (or average)carrier frequency, k -proportionality constant, $V_m \cos \omega_m t$ -instantaneous modulation voltage. $BW = 2(b + 1) f_m$ where b is the modulation index and f_m is the maximum modulating frequency used. Time domain and frequency domain representation of FM modulation is shown in fig.9 and fig.10 respectively.

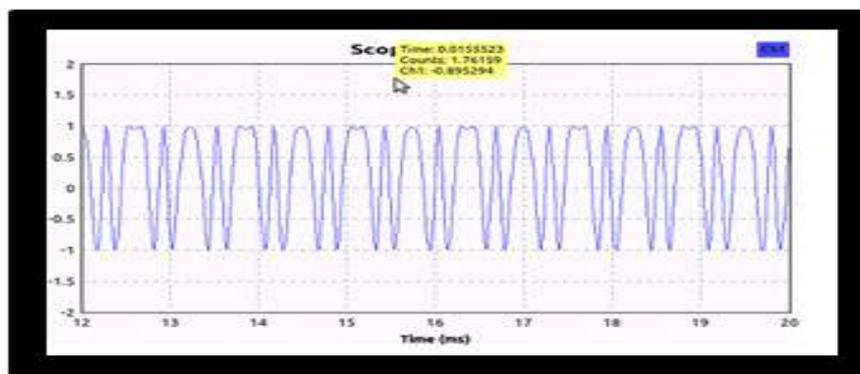


Fig.9.Modulation:Frequency Domain Representation

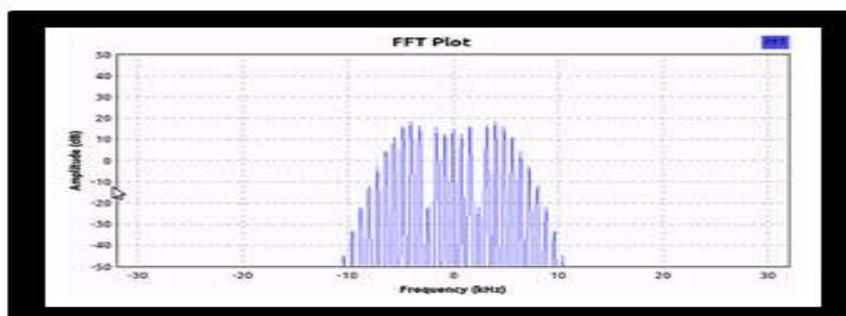


Fig. 10. FM Modulation:Time Domain Representation

D. Experiment 4: ASK Modulation and Demodulation

A Binary Amplitude Shift Keying(BASK)can be defined by

$$S(t) = Am(t) \cos(2\pi fct), 0 \leq t \leq T \tag{4}$$

where A is constant, m(t)=1 or 0, fc carrier frequency, T is the bit duration and has a power $P = \frac{A^2}{2}$.

Therefore S(t) can be written as

$$\begin{aligned} S(t) &= \sqrt{2p} \cos(2\pi fct), 0 \leq t \leq T \\ &= \sqrt{(PT)} \sqrt{(2/T)}\cos(2\pi fct), 0 \leq t \leq T \\ &= \sqrt{E} \sqrt{(2/T)}\cos(2\pi fct), 0 \leq t \leq T \end{aligned} \tag{5}$$

Fig.11 shows time domain representation of ASK modulation.

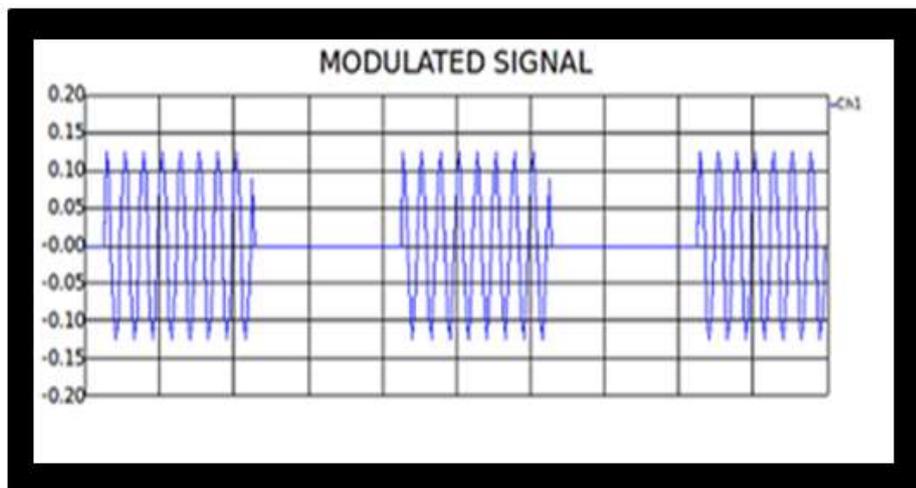


Fig 11.Time domain representation of ASK modulation

E. Experiment 5: FSK Modulation and Demodulation

The binary version of FM is called the Frequency Shift Keying or FSK. Here the frequency does not keep changing gradually over symbol time but changes in discrete amounts in response to a message similar to binary PSK where phase change is discrete. The modulated signal can be written very simply as consisting of Two different carriers.

$$S1 (t) = Ac \cos (2\pi f1t) \tag{6}$$

$$S2 (t) = Ac \cos (2\pi f2t) \tag{7}$$

S1 (t) in response to a 1 and S2 (t) in response to a 0. This can be written as a deviation from the carrier frequency

$$S1 (t) = Ac \cos (2\pi fc - \Delta f) t) \tag{8}$$

$$S2 (t) = Ac \cos (2\pi fc + s\Delta f) t) \tag{9}$$

Here Δf is called the frequency deviation. This is excursion of the signal above and below the carrier frequency and indicates the quality of the signal of the signal such as in stereo FM reception. Each of these frequencies f1 and f2. Are an offset from the carrier frequency, fc. let’s call the higher of these fh and lower fl.The combined signal can be written as, keeping mind that two terms are orthogonal in time and never occur at the same time.

$$BFSK (t) = Ah \cos(2\pi fht + \phi h) + Al \cos(2\pi fl t + \phi l) \tag{10}$$

Fig.12 represents time domain representation of FSK modulation.

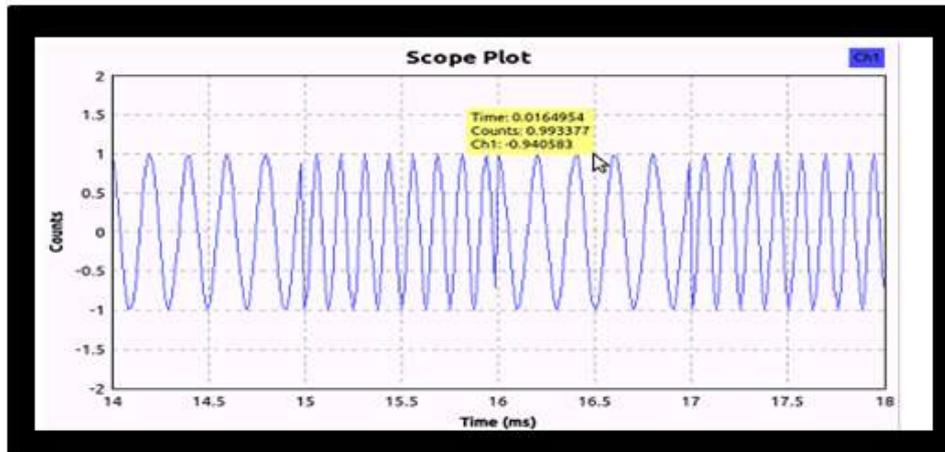


Fig. 12. FSK Modulation:Time Domain Representation

F. Experiment 6:MAC(Media access control)Protocols[12]

This experiment explains how Simplex data communication can be implemented. Transmitter sends 5 packets, then waits one second and sends the next 5 packets. The equipments required for this experiment is PC, USRP transmitter, receiver and RF cable. Experimental setup is shown in Fig.13.

Steps:

1. Open a terminal and start receiver./benchmark rx.py 900M
2. Open a new terminal and start transmitter ./benchmark tx.py 900M

MAC Protocols like voice transmission, Bluetooth, WiFi-both data and video streaming can be implemented by using SDR.

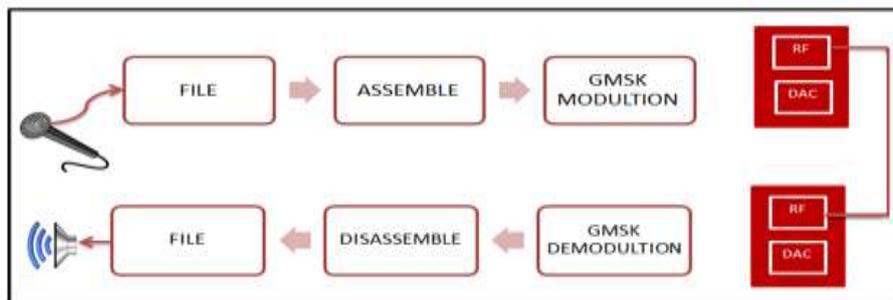


Fig. 13. Block Schematic of Simplex data transmission

V. Comparison Of Conventional Method With Sdr

Comparison of conventional method with SDR [8] is shown in table.1

Table.1

No:	Properties	Conventional Method	SDR
1	Reconfigurability	Configuration is done	On the fly configuration
2	Cost	High	Drastically low
3	Portability	Mostly suitable for static application	Can be placed anywhere
4	Local Oscillator	The changes in frequency stability and phase leads to spurious output signals in the output of the mixer	This problem is avoided here completely. Signal generation is done in PC
5	Waveform Application	The hardware design can be implemented for only one type of waveform in particular	The type of waveform can be changed when necessary

VI. Conclusion

Communication across the globe is now enjoying the advantages of SDR. Its beauty lies in its ability to receive and transmit various types of waveforms and its processing by using a common hardware. Even though it is difficult to write software program for various systems, SDR offers an excellent opportunity to examine new and better ways for defining communication system. Reduced content of expensive custom silicon and complex circuitry are main advantage of SDR. SDR offers the possibility of adaptively choosing operating frequency and mode best suited for prevailed conditions. Opportunity to recognize errors and avoiding the interface with communication channels should be highlighted. Also there is numerous possibilities for new experiments. 'Lack of awareness' in academics is the main issue. Since it works in open source platform, it can be easily downloaded and can be used as a demonstration tool for communication systems. SDR can be included in the syllabus of graduate students. Not only the graduate students even an under graduate student [13] can take the advantage of this technology. Moreover researches on SDR are still in process to develop cognitive radio [11] which is capable of sensing the surrounding wireless environment and user communications and computing needs and acting to meet those needs. SDR will surely change the complexities in entire communication system. This section contains the discussion of six classroom experimental demonstrations in communication theory.

References

- [1] Sharlene Katz "Using Software Defined Radio (SDR) To Demonstrate Concepts In Communications and Signal Processing Courses", Proceedings of the 39th ASEE/IEEE Frontiers in Education Conference, 2009.
- [2] Goodman, Peter "A Software-Defined Radio Project for First Year ECET Students", Proceedings of the ASEE Annual conference, 2007.
- [3] Kubichek, Robert, et al, "A Comprehensive Suite of Tools for Teaching Communications Courses", Proceedings of the ASEE Annual Conference, 2006.
- [4] Dunne, Bruce, and Cooke, Melvin, "Design of a Hardware Platform for analog Communications Laboratory", Proceedings of the ASEE Annual Conference, 2008.
- [5] Frolik, Jeff, "Laboratory Enhancement of Digital and Wireless Communications Courses", Proceedings of the ASEE Annual Conference, 2005.
- [6] Reed, Jeffrey, "Software Radio: A Modern Approach to Radio Engineering", Prentice Hall, 2005.
- [7] Bard, John and Kovarik, Vincent, "Software Defined Radio: The Software Communications Architecture, "Wiley series in software radio, 2007.
- [8] Ramanathan R, "Implementation of a Low Cost Synthetic Aperture Radar using software defined radio", Proceedings of second international Conference on computing. Communication and networking technologies, 2010.
- [9] Ettus Research LLC: <http://www.ettus.com>
- [10] Soft rock and other SDR Radios: <http://www.softrockradio.org/>
- [11] Fette, Bruce, "Cognitive Radio Technology, "Newnws, 2006.
- [12] "Media access control Protocols", by Alaleddin Mohammed
- [13] "Software Defined Radio Laboratory" University of Tehran
- [14] <http://www.stanford.edu/pgbovine/cde.html>