

Practical Approach to

Software Defined Radios

Practical Approach to

Software Defined Radios

Ram & Amitesh Pandey

Copyrights © 2019 Ram & Amitesh Pandey

All rights reserved. No part of this book may be reproduced, stored, or transmitted by any means-whether auditory, graphical, mechanical, or electronic-without written permission of the author. Any unauthorized reproduction of any part of this book is illegal and is punishable by law.

To the maximum extent permitted by law, the author and publisher disclaim all responsibility and liability to any person, arising directly or indirectly from any person taking or not taking action based on the information available in this publication.

ISBN-13: 978-93-89113-62-4

ISBN-10: 9-38-911362-8

Printed in India and published by BUUKS.

Table of Content

Acknowledgment	9
Preface	11
Chapter1: Introduction to Software Defined Radio	13-35
1. Software Defined Radio	
2. Universal Software Radio Peripheral	
3. Installation of UHD & Gnu radio	
Chapter2: Introduction to Gnu radio	36-74
1. Implement the transmission and reception of signals through the audio source & wav file blocks.	
2. Understand the importance of throttle block in Gnu radio.	
3. Why packet encoder and packet decoder is used in transmission and reception of real time signals in Gnu radio and on SDR?	
4. Analyse the different types of Boolean blocks available in Gnu radio.	
5. Explore the different type converters available in Gnu radio.	
6. Explore the Noise source, Signal source, random source & vector source available in Gnu radio.	
7. Experiment to explore the WX based GUI standards used in Gnu radio.	
8. Experiment to explore the QT based GUI standards used in Gnu radio.	
Chapter3: Getting Familiar with Signal Processing	75-115
1. Generate sine, cosine, square, triangle, sawtooth wave in Gnu radio	
2. Analyse the effects of types of Noises on cosine wave signal in Gnu radio.	
3. Experiment to implement the Quantization of input signals in Gnu radio.	
4. Experiment to implement the Nyquist sampling theorem in Gnu radio.	
5. Experiment to implement an aliasing error in Gnu radio	
6. Experiment to implement the frequency shifting of input signal in Gnu radio.	

7. Analyse the Gaussian Noise signal with change in signal to Noise ratio (SNR) in Gnu radio.
8. Analyse the Bit error rate (BER) in Gnu radio.
9. Analyse the oversampling perfect sampling and under sampling of a signal in Gnu radio.

Chapter4: Filters

116-177

1. Experiment to explore the low pass filter using SDR.
2. Experiment to analyse the effects of low pass filter on the Audio signals using Gnu radio.
3. Experiment to explore the high pass filter using SDR.
4. Experiment to explore the effects of high pass filter on the audio signals using Gnu radio.
5. Experiment to explore the Band pass filter using SDR.
6. Experiment to explore the Band Reject filter using SDR.
7. Experiment to implement the finite impulse response filter (IIR) using SDR.
8. Experiment to implement the Infinite Impulse Response using (IIR) SDR.
9. Experiment to implement the Root cosine filter and analyse its effects on the input signals using Gnu radio.
10. Experiment to analyse different windowing techniques with low pass filter using Gnu radio.

Chapter5: Analog Modulation

178-199

1. Implement Amplitude modulation using cosine wave using USRP B100.
2. Implement the Transmission and reception of audio signal using Amplitude modulation.
3. Implement the Frequency modulation techniques with audio signals using USRP B100.
 - A. Implementation of FM using recorded audio signals.
 - B. Implementation of FM receiver which receives a real time signals from FM Station.

Chapter6: Digital Modulation	200-232
1. Implement Amplitude Shift Keying using Gnu radio and validation using USRP B100.	
2. Implement Phase Shift Keying modulation techniques.	
3. Implement the Frequency Shift Keying using Gnu radio.	
4. Implement the Different modulation techniques (BPSK, QPSK, QAM) using Gnu radio and validate using USRP B100.	
Chapter7: Transmission and Reception of Real world Audio/Video/Text/Image using SDR	233-257
1. Transmission and reception of Text messages using SDR	
2. Transmission and Reception of Audio signals using	
A. BPSK	
B. QPSK	
C. QAM	
D. GMSK	
3. Transmission and Reception of Video signals using Gstreamer and Gnu radio.	
Chapter8: Introduction to Radar System	258-265
Chapter9: OpenBTS “Build your own 2.5 G cell Phone Network	266-279
Chapter10: ADSB Receiver	280-293
1. Introduction to ADSB Receiver	
2. Implementation of ADSB-Receiver using Modes_GUI and Modes_rx GRC	
3. Implementation of ADSB-Receiver using ADSB Blocks in Gnu radio	
Chapter11: Frequency Signal Jamming	294-300
Chapter12: GSM Sniffing using SDR	301-310

Acknowledgement

Writing a book especially involving technical concepts with greater detail is a strenuous effort than what we thought but this arduous journey of authoring a book would have been impossible without effort as a team at Tenet Technetronic. We are eternally grateful to all the team members at Tenet Technetronics that contributed and helped support this wonderful opportunity.

This book is heavily contributed by practical materials created by

Mr. Amitesh Pandey working as an Application Engineer for the Software Defined Radio related products. As always he extends his gratitude towards his parents for their unconditional love, support, prayers, caring and sacrifices for betterment of his future.

Mr. PandiDurai Nallamuthu (Product Manager, Software Radio Platforms) for his support on advising on timely solutions on usage related issues and concerns on the USRP platforms.

Mr. Prabhu Raju and Mr. Ram have supported the logistics and critical technical review of the book with attention to detail and technical accuracies at all stages.

Preface

This book as the title suggests is written with an intention to get familiar with the concepts of Software Defined Radio and hardware/software that help in wireless system design. After extensive survey we found that adopting open source software/hardware could be right fit and hence the use of GNU Radio like open software and USRP open hardware has been used for all experiments throughout the book . Given this reason the book should serve as a good practical guide with detailed steps to follow through to reproduce results as mentioned in the experiments.

GNU Radio is a free & 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 research, industry, academia, government, and hobbyist environments to support both wireless communications research and real-world radio systems. For purposes of easy accessibility of Universal Software Radio Peripheral (USRP) is a range of software-defined radios designed and sold by Ettus Research and its parent company, National Instruments. Developed by a team led by Matt Ettus, the USRP product family is intended to be a comparatively inexpensive hardware platform for software radio, and is commonly used by research labs, universities, and hobbyists.

This book clearly documents both theoretical aspects of a concept and also implementation using software only approach with simulations carried out in GNU radio platform as well as using SDR hardware. In this way readers can clearly see the distinction between simulations versus practical behavior. If you discover any mistakes, missing information or just have any feedback on the book please feel free to contact us at info@tenettech.com.

Tips for reading the book : For effective learning using the book one should install GNU radio software package on a PC and then follow through experiments by downloading the chapter's exercises from the web and proceed along so that there is good understanding. For buying Software Defined Radios one could approach Ettus Research or Tenet Technetronics to get one.

Chapter 1

Introduction to Software Defined Radio

Objectives

With the innovations in the last decade in the programmable hardware as well as their ability to incorporate more volume of software intelligence, it is evident that various technology domains have seen the benefits. The SDR as the name suggests is a combination of highly configurable hardware that can be well configured using software. This chapter mainly covers the fundamentals to introduce the reader towards the concept of SDR, its application areas, different options available as solutions.

Out of the available solutions out there in public the book as mentioned before further elaborates the capabilities of Universal Software Radio Peripheral (USRP) generation of SDRs. However it's fair to say the concepts introduced should be reusable across other available solutions as well.

It is extremely important as well to understand the selection of the right platform when dealing with wireless system design using SDRs. This chapter also introduces the available specification of USRP (while at the time of writing this book) along with its suitability to some representative end applications so that the user is capable of understanding the fundamentals involved in selection of an SDR.

On the software front Gnu Radio Companion (GRC) is used in order to conduct various experiments mainly due to the ease of use since it leverages a signal flow based workflow. The complete information about the installation of GRC and USRP Hardware drive (UHD) is also captured in the best possible detail so that it's easier for a reader to start from scratch.

This chapter is divided in 3 sections:

Section 1: Software Defined Radio.

1. Why Software Defined Radio?
2. What is Software Defined Radio?
3. Architecture of Software Defined Radio.
4. Advantage of Software Defined Radio.
5. Application of Software Defined Radio.

Section 2: Universal Software Radio Peripheral (USRP).

1. What is USRP?
2. Architecture of USRP Series.
3. Types of USRP Series.
4. Types of Radio Frequency Daughterboard.

Section 3: Gnu Radio Companion (GRC)

1. What is GRC?
2. Installation of GRC and USRP Hardware driver (UHD).

Section 1: Software Defined Radio

Why Software Defined Radio?

With the exponential growth in the ways and means by which people need to communicate - data communications, voice communications, video communications, broadcast messaging, command and control communications, emergency response communications, etc. – modifying radio devices easily and cost-effectively has become business critical. Wireless systems demand increasingly high degree of software intelligence and reconfigurability to include field feedback or remote diagnostics/ customization as per user request. Traditional radios in the past have been best in delivering fixed functionality as depicted in the Figure 1.1.

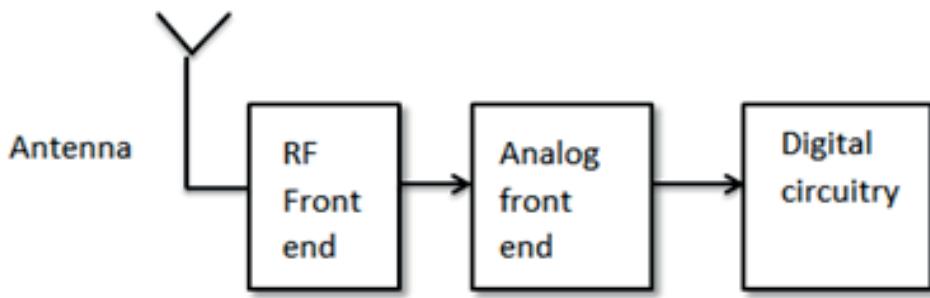


Figure 1.1 Tradition Wireless Communication Systems

Traditional wireless communication system constitute of a Radio frequency signal generation front end followed by an analog to digital converter (ADC) and digital circuitry for analyzing the signals . Given this case for different applications the system was designed with its fixed logic each time exclusively and hence this lead to the modification in the hardware components.

With the advent of programmable digital platforms like the FPGA (Field Programmable Gate Arrays), Digital Signal Processors (DSPs) and Microcontrollers that have increased in processing capacity it has become more a choice to include these as part of the system with much ease. This very feature of programmable hardware now becomes the core of reconfigurable Radio platforms that are popularly known as Software Defined

Radios. The system can be reconfigured at run time with no changes in the hardware hence providing a sense of what is called “One solution fits more”.

Fig 1.2 represents the pictorial view of an SDR where the architecture represents a programmable hardware in the form of a FPGA and interchangeable RF front end so that the platform can be easily customized/reconfigured for different end use cases. The USRP platforms from Ettus research (A National Instruments’ Company) provides various configurations which are introduced during the course of this book that populates different options of FPGA/Hardcore Processors with various software options that makes it completely flexible for different applications. A user could select based on the requirements of Area of FPGAs (Logical Element/Configurable Logical Block count), Number of RF channels, etc.

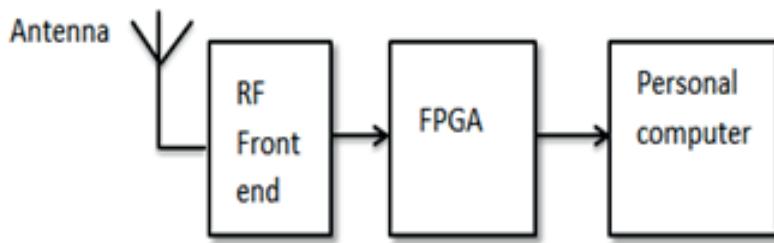


Figure 1.2 Software Defined Radio Communication System

A summary in the form of some of the high level differences between traditional radio vs SDRs is captured in the Table 1. Hence based on these comparisons, it can be observed that Software radio platforms offer a good choice of the day (of course the time at which this book is being written).

What is Software Defined Radio?

Software Defined Radio as the Wikipedia reads out

“Software-defined radio (SDR) is a radio communication system where the digital components that have been traditionally implemented with respect to the hardware like filters, amplifiers, Synchronizer, modulators, demodulators, detectors etc are instead implemented by means of software on a personal computer or on the embedded system. While the concept of SDR is not new, the rapidly evolving capabilities of digital

electronics render practical many processes which were once only theoretically possible.”,

Table 1: Comparison of SDR vs Traditional Wireless Communication System

Difference	Software Defined Radio	Conventional Communication
Design Flexibility	Yes	No
Reliability	Yes	Not completely because of hardware
Design flow time	Faster because of software	Requires time to modify the hardware circuit
Upgradability	Easily done	It is tedious, as complete set of hardware need to be replaced for upgrading.
Reusability	Yes	No
Reconfigurability	Yes	It cannot be reconfigured to other application

SDRs can be used to implement the entire wireless communication system with the transmission and reception of real time signals. SDR platforms that are available off the shelf these days nicely fills the gap between the theoretical concept and the practical aspects.

With the right choice of base platform to process and analyze, RF front end as well as Antennas it should be possible to practically create platforms that cover a wide range of frequencies for study and system design. The book also introduces the choices available to users on the USRP platforms and also some possible applicability to some representative use cases in the forthcoming chapters.

It is also evident clearly that it's just not the reconfigurable hardware but equally important are software frameworks/platforms/development tools that make the

software programming/modelling easier that is vital to have a very effective development setup. As introduced previously this book focuses on using open software like GNU radio to showcase the design flow.

Architecture of Software Defined Radio

Fig 1.3 represents the architecture of Software Defined Radio in a little more in detail with USRP as a reference but more or less other Software Radios more or less follow the same kind of hardware organization. In this case there are three main sections:

1. Base Band Section
2. IF Section
3. RF Section

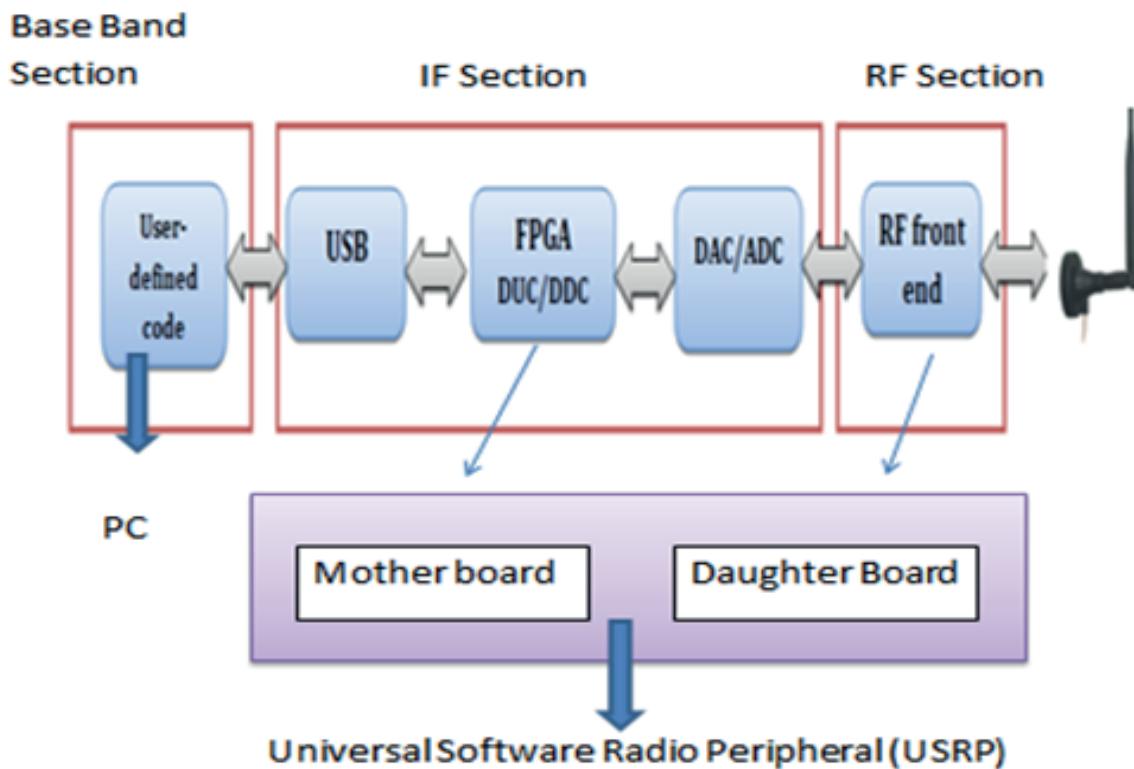


Figure 1.3 Architecture of Software Defined Radio

The base band section includes the host system with Gnu radio companion and UHD installed. UHD is abbreviated as USRP Hardware driver which is used to interface Gnu radio with SDR. With the help of interfacing medium the host system is connected with

IF section. For example USRP B100 is interfaced with the host system using USB 2.0. The combination of IF section and RF section together results in SDR. IF section include FPGA which up convert and down convert the given signals in order to convert the signals into intermediate frequency signals . Once the signal is converted to intermediate frequency signal, it is further converted to digital signals. This digital signal has to be converted to high radio frequency signal before the transmission through Antenna, which can be probably achieved using RF Daughterboard. The RF signal is then further radiated using antenna (Antenna specification changes based on the requirement).

Fig 1.4 represents the architecture of RF Daughterboard (RF Section). RF Daughterboard is where a Transceiver carrier signal is mixed with the local oscillator to form an intermediate frequency.

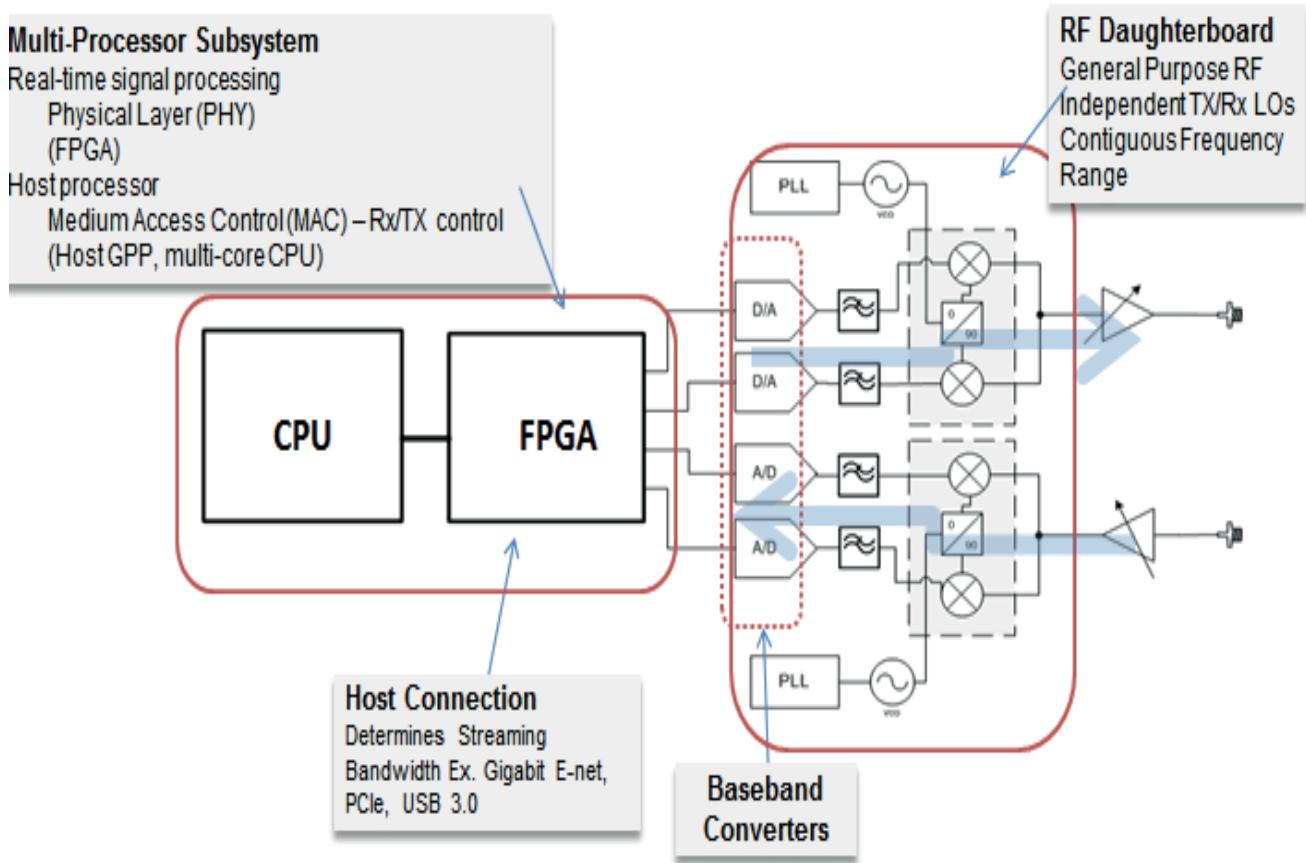


Figure 1.4 Architecture of RF Daughterboard

Phase Lock loop (PLL) connected to voltage control oscillator (VCO) is used to have a linear output phase signals as compared to input signal. VCO will control the frequency

of oscillation with respect to the input voltage. Multiple digital to analog converter are used to convert the signal to and from digital to analog. The multiple combinations of A/D and D/A results in the Baseband converters. There are multiple types of RF Daughterboard based on their range of Radio frequency signal.

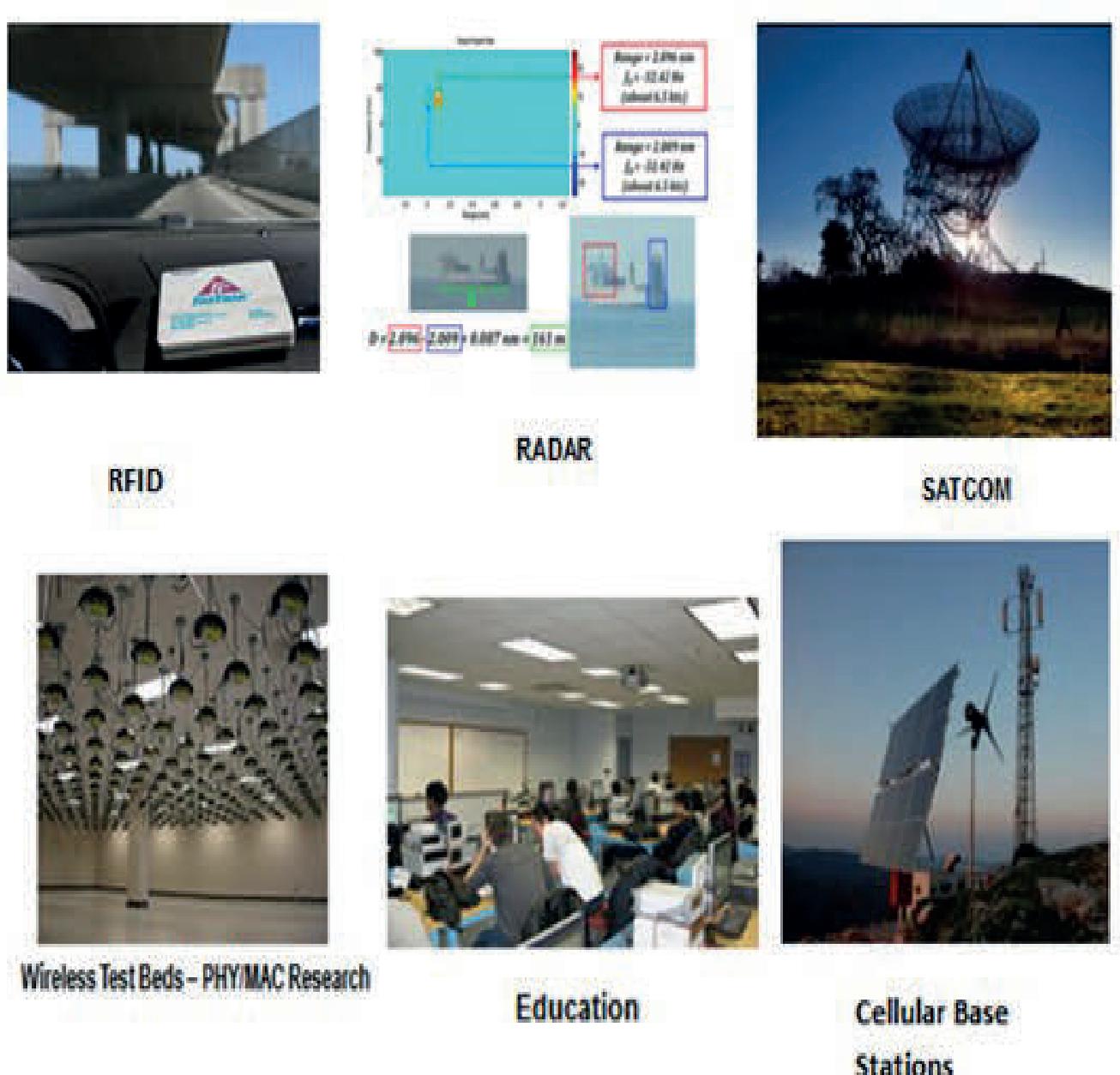


Figure 1.5 Applications of SDR

Applications of SDRs

The list below should cover some important areas where SDR platforms are seeing adoption and production use. There are more than the one listed below and these are listed as a representative case

1. 4G LTE Systems and Beyond.
2. Dynamic spectrum access.
3. Wireless propagation measurements and modeling.
4. Wireless localization (e.g., to serve as a temporary infrastructure in disaster scenarios).
5. IOT enabled Smart grid communications and monitoring of smart meter infrastructure.
6. Cyber security for wireless communications and localization.
7. Device to device and multi-hop communications.
8. TV white space communications.
9. Multiple-Input Multiple-Output Communications including massive MIMO applications.
10. Wireless radar.
11. Intelligent transportation system applications.
12. In VIVO communications.
13. Implementation of GSM and LTE base stations.

Section 2: Universal Software Radio Peripheral

What is Universal Software Radio Peripheral?

Universal Software Radio Peripheral (USRP) is a software defined radio (SDR) device which is used for the various Radio frequency applications. USRP platforms also have some units that host transceivers which can both transmit/receive RF signals in several bands through a single unit. USRPs are seen adopted increasingly in various applications with respect to wireless communications system design and used by Researchers, wireless system design engineers, students and universities worldwide. USRP platforms are open source hardware which implies that the hardware design files and firmware and other details and available openly with great flexibility for users to even design their own or derive and customize these platforms for their own use. USRP platforms have tight integration and support under the GNU radio platforms and this combination makes it a very attractive combination.

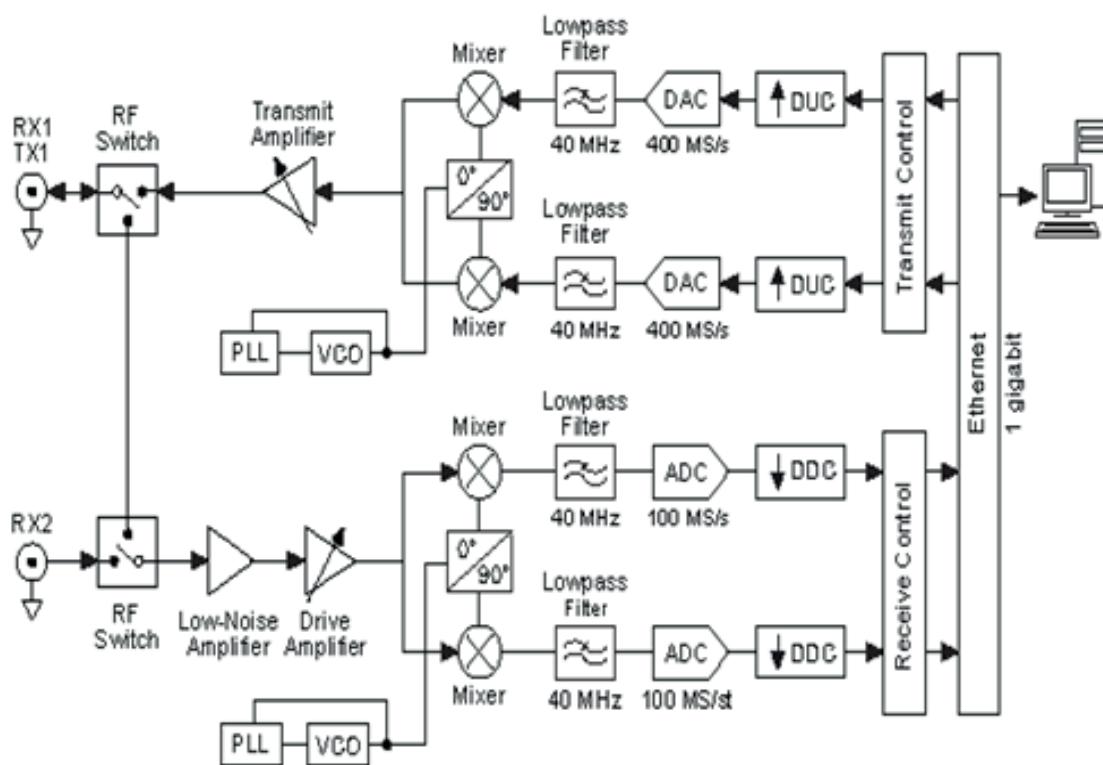


Figure 2.1 Architecture of USRP

Varying from model to model of USRP, only some characteristics and specifications changes but the basic architecture remains the same. The RF frontend, the mixers, filters, oscillators and amplifiers required to translate a signal from the RF domain and the complex baseband or IF signals. The baseband of IF signals are sampled by ADCs and the digital samples are clocked into an FPGA. The stock FPGA image provides digital down-conversion, functionality, which includes fine-frequency tuning and several filters for decimation. After decimation, raw samples or other data are streamed to a host computer through the host interface. The reverse process applies to the transmit chain.

The bandwidth of the USRP device varies at each point in the signal chain. Three general types of bandwidth specifications are the analog bandwidth, the FPGA processing bandwidth, and host bandwidth. The system bandwidth is generally the minimum of the RF daughterboard, FPGA processing, and host bandwidth.

Note:

A special care is needed to avoid an analog bandwidth that is greater than the ADC/DAC sample rate of any device.

Types of USRP

There are different series of USRP which are as follows:

1. BUS series
2. Embedded series
3. Network series
4. X series

Bus series

Ettus Research Bus Series uses a USB 3.0 interface to transfer samples to and from the host computer. These are designed for applications that do not require the higher bandwidth and Dynamic range. The USRP Bus Series provides a fully integrated, single board, Universal Software Radio Peripheral platform with continuous frequency coverage from 70 MHz – 6 GHz.



Designed for low-cost experimentation, it combines a fully integrated direct conversion transceiver providing up to 56MHz of real-time bandwidth, an open and reprogrammable Spartan6 FPGA, and fast and convenient bus-powered Super Speed USB 3.0 connectivity. The Bus series has many versions like B200, B210, B200min, and B205mini. Refer link

<https://kb.ettus.com/B200/B210/B200mini/B205mini>

Embedded series



The Embedded Series combines the same functionality of other USRP devices with an Open Multimedia Applications Platform embedded processor. The devices in this family do not need to be connected to an external PC for operation. The Embedded Series is designed for applications that require stand-alone operation. It offers a portable stand-alone SDR platform designed for field deployment. The flexible transceiver from Analog

Devices provides up to 56 MHz of instantaneous bandwidth and spans frequencies from 70 MHz – 6 GHz to cover multiple bands of interest. Refer link

https://kb.ettus.com/Ettus_USRP_E300_Embedded_Family_Hardware_Resources

Network series

High-performance USRP devices that provide higher dynamic range and higher bandwidth. This series also provides a MIMO expansion port which can be used to synchronize two devices from this series. The USRP Network Series offers high-bandwidth, high-dynamic range processing capability. The Gigabit Ethernet interface of the USRP Network Series allows high-speed streaming capability up to 50 MS/s in both directions (8-bit samples). These features, combined with plug-and-play MIMO capability make the USRP Network an ideal candidate for software defined radio systems with demanding performance requirements. Refer link

<https://kb.ettus.com/N200/N210>

<https://kb.ettus.com/N300/N310>



X series

High-performance, scalable software defined radio (SDR) platform for designing and deploying next generation wireless communications systems. The Ettus Research USRP X310 is a high-performance, scalable software defined radio (SDR) platform for

designing and deploying next generation wireless communications systems. The hardware architecture combines two extended-bandwidth daughterboard slots covering DC – 6 GHz with up to 160 MHz of baseband bandwidth, multiple high-speed interface options (PCIe, dual 10 GigE, dual 1 GigE), and a large user-programmable Kintex-7 FPGA in a convenient desktop or rack-mountable half-wide 1U form factor.



Refer link

<https://kb.ettus.com/X300/X310>

Types of RF Daughterboard

A daughterboard is a circuit board that plugs into and extends the circuitry of another circuit board. The USRP family features a modular architecture with interchangeable daughterboard modules that serve as the RF front end. Several classes of daughterboard modules exist. Transmitter daughterboard modules can modulate an output signal to a higher frequency. Receiver daughterboard modules can acquire an RF signal and convert it to baseband. Transceiver daughterboard modules combine the functionality of a Transmitter and Receiver.

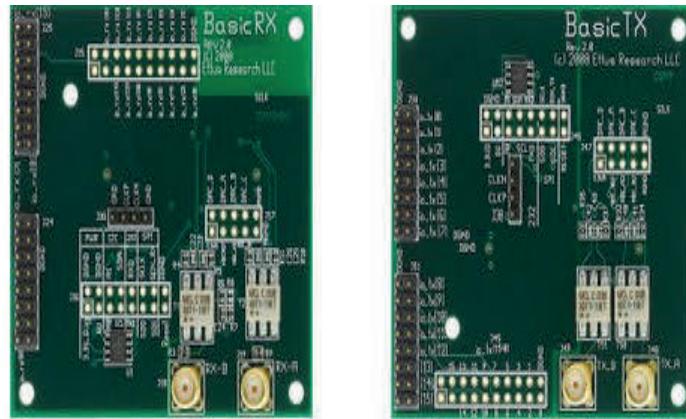
There are different types of RF Daughterboard which are as follows:

1. Basic TX/Basic RX
2. WBX
3. SBX
4. CBX
5. UBX

Table 2: Comparisons of different types of USRP Series

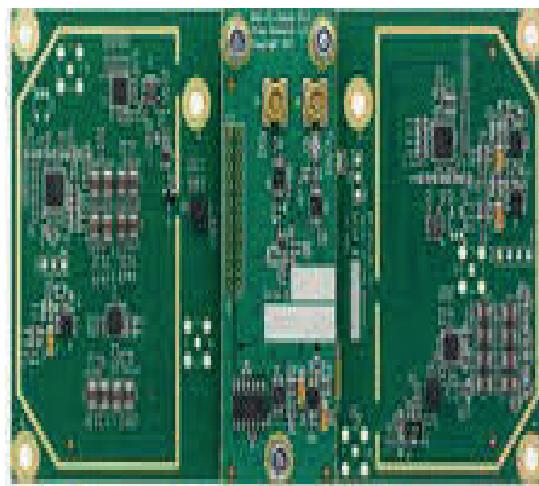
USRP B	USRP N	USRP X	USRP E
B200,B210, B200min,B205min	N200,N210, N300,N310	X300,X310	E310,E312,E313
Interfaced using USB 3.0	Interfaced using Ethernet	Interfaced using USB 3.0, 1 Giga Ethernet, 10 Giga Ethernet	Interfaced using Ethernet
1 Daughterboard	1 Daughterboard	2 Daughterboard	NA
NA	MIMO capable 2X2 -> MIMO cable 4X4 -> Octoclock	MIMO capable 2X2 -> MIMO cable 4X4 -> Octoclock	MIMO capable 2X2 -> MIMO cable 4X4 -> NA
GPSDO kit applicable in B200,B210	GPSDO kit is applicable	GPSDO kit is applicable	NA
FPGA Xilinx SPARTAN 6	FPGA Xilinx Spartan 3 A	FPGA Xilinx Kintex 7	FPGA Xilinx Zyng- 7000
RF – 70Mhz to 6Ghz	RF – DC to 6 Ghz	RF – DC to 6 Ghz	RF – 70Mhz to 6Ghz

Basic Tx/Rx



The BasicRX/BasicTX daughterboards are low-cost daughterboards that provides direct access to the ADC inputs. The boards can accept real-mode signals from 1 to 250 MHz. The BasicRX/BasicTX is ideal for applications using an external front end providing relatively clean signals within operable bandwidth.

WBX



The WBX is a wide bandwidth transceiver that provides up to 100 mW of output power and a noise figure of 5 dB. The LO's for the receive and transmit chains operate independently. The WBX provides phase coherent operation, although with a 180-degree ambiguity, which must be calibrated out in the application.

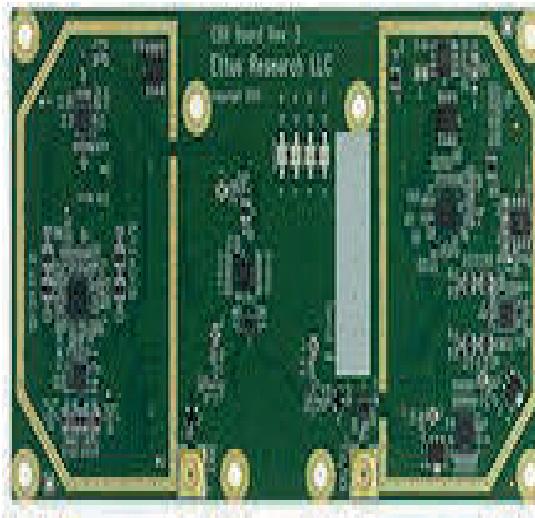
SBX



The SBX is a wide bandwidth transceiver that provides up to 100 mW of output power, and a typical noise figure of 5 dB. The local oscillators for the receive and transmit chains operate independently, which allows dual-band operation. The SBX is MIMO capable, and provides 40 MHz or 120 MHz of bandwidth. The SBX is ideal for applications requiring access to a variety of bands in the 400 MHz-4400 MHz range. Example application areas include Wi-Fi, WiMAXs-band transceivers and 2.4 GHz ISM band transceivers. The SBX is capable of phase coherent operation, and therefore is suitable for MIMO and Phased Array applications.

CBX

The CBX is a full-duplex, wideband transceiver that covers a frequency band from 1.2 GHz to 6 GHz with an instantaneous bandwidth of 40 MHz or 120 MHz. The CBX can serve a wide variety of application areas, including Wi-Fi research, cellular base stations, cognitive radio research, and RADAR. The CBX does not provide phase coherent operation, and therefore is not recommended for MIMO and Phased Array applications.



UBX

The UBX daughterboard is a full-duplex wideband transceiver that covers frequencies from 10 MHz to 6 GHz. Coherent and phase-aligned operation across multiple UBX daughterboards on USRP X Series motherboards enables users to explore MIMO and direction finding applications. The UBX daughterboard works interchangeably with other USRP daughterboards and is supported by the USRP Hardware Driver (UHD) software.

API for seamless integration into existing applications. The UBX is capable of phase coherent operation, and therefore is suitable for MIMO and Phased Array applications, on the X Series. Additionally this capability is only available on the X Series devices.



Table 3: Comparisons of different types of RF Daughterboard

WBX	SBX	UBX	CBX
RF Signal 50mhz to 2.2 Ghz	RF Signal 400mhz to 4.4 Ghz	RF Signal 10Mhz to 6 Ghz	RF Signal 1.2 Ghz to 6 Ghz
MIMO Capable	MIMO Capable	MIMO Capable	MIMO Capable
Noise Figure 5 dB	Noise Figure 5 dB	Noise Figure 5 dB	Noise Figure 5 dB
o/p 100mW	o/p 100mW	o/p 100mW	o/p 100mW
2 types →WBX 40 and WBX 120	2 types →WBX 40 and WBX 120	2 types →WBX 40 and WBX 160	2 types →WBX 40 and WBX 120
Analog Bw 40 MHz or 120MHz.	Analog Bw 40 MHz or 120MHz.	Analog Bw 40MHz or 160MHz.	Analog Bw 40 MHz or 120MHz.
→No phase synchronization	→Phase synchronization	→Phase synchronization →RF shielding	→Phase synchronization

Section 3: Introduction to GNU Radio

What is GNU Radio Companion?

GRC abbreviated as GNU Radio companion is a free & open-source software development toolkit which provides analyzing of real time signals using signal processing blocks in order to implement the radio standards. It can be interface with external Radio frequency hardware for validation, or else can be used without hardware in a simulation-like environment. It is widely used in research, industry, Academia, Government, and Hobbyist environments

Installation of UHD and Gnu radio

Before building UHD and GNU Radio, you need to make sure that all the dependencies are first installed. Hence type the command in terminal as given bellow:

```
sudo apt-get update
```

Add sudo as a prefix if the system is a root user. Once the system has been updated, then install the required dependencies for UHD and GNU Radio. This chapter focuses on Ubuntu 16.04, therefore the dependencies files are

```
sudo apt-get -y install git swig cmake doxygen build-essential libboost-all-dev libtool  
libusb-1.0-0 libusb-1.0-0-dev libudev-dev libncurses5-dev libfftw3-bin libfftw3-dev  
libfftw3-doc libcunit-1.13-0v5 libcunit-dev libcunit-doc ncurses-bin cpufrequtils  
python-numpy python-numpy-doc python-numpy-dbg python-scipy python-docutils  
qt4-bin-dbg qt4-default qt4-doc libqt4-dev libqt4-dev-bin python-qt4 python-qt4-dbg  
python-qt4-dev python-qt4-doc python-qt4-doc libqwt6abi1 libfftw3-bin libfftw3-dev  
libfftw3-doc ncurses-bin libncurses5 libncurses5-dev libncurses5-dbg libfontconfig1-dev  
libxrender-dev libpulse-dev swig g++ automake autoconf libtool python-dev libfftw3-dev  
libcunit-dev libboost-all-dev libusb-dev libusb-1.0-0-dev fort77 libsdl1.2-dev python-  
wxgtk3.0 git-core libqt4-dev python-numpy ccache python-opengl libgsl-dev python-  
cheetah python-mako python-lxml doxygen qt4-default qt4-dev-tools libusb-1.0-0-dev  
libqwt5-qt4-dev libqwtplot3d-qt4-dev PyQt4-dev-tools python-qwt5-qt4 cmake git-core
```

```
wget libxi-dev gtk2-engines-pixbuf r-base-dev python-tk liborc-0.4-0 liborc-0.4-dev  
libasound2-dev python-gtk2 libzmq-dev libzmq1 python-requests python-sphinx  
libcomedi-dev python-zmq python-setuptools
```

If the installation of the dependencies completes without any errors, then you can proceed to build and install UHD and GNU Radio.

If the installation of the dependencies completes without any errors, then you can proceed to build and install UHD and GNU Radio.

Building and Installing UHD from the source code

Make a folder to hold the repository.

```
cd $HOME
```

```
mkdir workarea-uhd
```

```
cd workarea-uhd
```

Next, clone the repository and change into the cloned directory.

```
git clone https://github.com/EttusResearch/uhd
```

```
cd uhd
```

Checkout the desired UHD version. You can get a full listing of tagged releases by running the command:

```
git tag -l
```

Next, create a build folder within the repository.

```
cd host
```

```
mkdir build
```

```
cd build
```

Next, invoke Cmake to create the make files

```
cmake../
```

Next, run Make to build UHD.

```
make
```

Next, you can optionally run some basic tests to verify that the build process completed properly.

```
make test
```

Next, install UHD, using the default install prefix, which will install UHD under the /usr/local/lib folder. You need to run this as root due to the permissions on that folder.

```
sudo make install
```

Next, update the system's shared library cache.

```
sudo ldconfig
```

At this point, UHD should be installed and ready to use. You can quickly test this, with no USRP device attached, by running uhd_find_devices. You should see something similar to the following.

Linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.000.HEAD-0-g6e1ac3fc

No UHD Devices Found

Building and Installing the Gnu Radio from the source code.

First, make a folder to hold the repository.

```
cd $HOME
```

```
mkdir workarea-gnuradio
```

```
cd workarea-gnuradio
```

Next, clone the repository.

```
git clone --recursive https://github.com/gnuradio/gnuradio
```

Next, go into the repository and check out the desired GNU Radio version.

```
cd gnuradio  
git checkout v3.7.10.1  
git submodule update --init --recursive
```

Next, create a build folder within the repository.

```
mkdir build  
cd build
```

Next, invoke Cmake to create the files.

```
cmake ../
```

Next, run Make to build GNU Radio.

```
make
```

Next, you can optionally run some basic tests to verify that the build process completed properly.

```
make test
```

Next, install GNU Radio, using the default install prefix, which will install GNU Radio under the /usr/local/lib folder. You need to run this as root due to the permissions on that folder.

```
sudo make install
```

Finally, update the system's shared library cache.

```
sudo ldconfig
```

At this point, GNU Radio should be installed and ready to use. You can quickly test this, with no USRP device attached, by running the following quick tests.

```
gnuradio-config-info --version
```

`gnuradio-config-info --prefix`

`gnuradio-config-info --enabled-components`

There is a simple flow graph that you can run that does not require any USRP hardware. It's called the dial tone test, and it produces a PSTN dial tone on the computer's speakers. Running it verifies that all the libraries can be found, and that the GNU Radio run-time is working. You can try launching the GNU Radio Companion (GRC) tool, a visual tool for building and running GNU Radio flow graphs.

`gnuradio-companion`

The installation of UHD and GNU Radio should now be complete. At this point, connect the USRP to the host computer. If the interface is Ethernet, then open a terminal window, and try to ping the USRP with "ping 192.168.10.2". The USRP should respond to the ping requests. If the interface is USB, then open a terminal window, and run:

`lsusb`

Also try running

`uhd_find_devices`

`uhd_usrp_probe`

Inference

Hence this chapter provides the complete description about the Software Defined Radio and its application in today's world of wireless communication system. The Universal Software Defined Radio (USRP) is a type of SDR, which is used to validate the simulated results. Hence the gap between the theoretical aspects and practical conditions can be properly filled and analyzed. The further chapters will provide a step by step pictorial representation of the wireless communication system with respect to the real time application.

Chapter 2

Introduction to Gnu Radio blocks

Gnu radio software platform provides blocks that include many of the generic, standard, application specific blocks used for many commonly used signal and communication systems. It helps to create flow graphs/system models with an easy representation of either subsystems/entire wireless radio systems. Gnu radio flow graphs can be simulated on a personal computer. The successful simulation can also be easily converted with specific blocks to interact with SDR hardware like the USRP to test the implementation either in combination with real hardware or even in some cases deploy the whole graph in a standalone executable format.

This chapter in the book is mainly documented with the intention that it introduces the types of blocks available in Gnu radio companion (GRC) with the help of small experiments so that the reader can practically try these by installing GNU radio and reproduce the results.

A List of experiments have been curated as below in such a way that there is good usage of mainly used block sets in GNU radio and would help use the learnings further in constructing different setups in future, it is also carefully structured with a clear objective, steps to create a GNU radio graph and results to be observed mainly with the intention of the book being directly used for instructional purposes for instance by a professor to teach his classroom of students.

Table 4: List of GNU radio Experiments

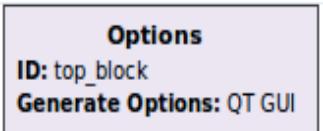
Sr.No	Description
1	Experiment to implement the transmission and reception of signals through the audio source & wav file blocks.
2	Experiment to understand the importance of throttle block in Gnu radio.
3	Why packet encoder and packet decoder is used in transmission and reception of real time signals in Gnu radio and on SDR?
4	Experiment to analyze the different types of Boolean blocks available in Gnu radio.
5	Experiment to explore the different type converters available in Gnu radio.
6	Experiment to explore the Noise source, Signal source, random source & vector source available in Gnu radio.
7	Experiment to explore the WX based GUI standards used in Gnu radio.
8	Experiment to explore the QT based GUI standards used in Gnu radio.

Experiment 1

Aim

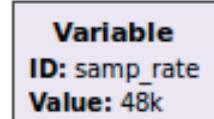
The purpose of this experiment is to explore Audio Sink/Source and WAV file source blocks in Gnu Radio.

Blocks used in this experiment



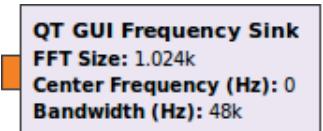
1. Options

Options block is used to select the standard QT or WX for the GUI tool kit on the machine. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as QT GUI.



2. Variable

Variable block is used to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 48K for proper conversion of analog signals to discrete sampled signals.



3. QT GUI frequency Sink

GUI stands for Graphic user interface with standard QT. Frequency sink is used to represent the frequency plot for the desired output signals with FFT size selected as 1.024K by default.

Gnu Radio Flow graphs

Audio signal can be transmitted using two ways in Gnu radio:

1. Transmission through Audio Source (PC line in/out).
2. Transmission through Wav File Source.

1. Transmission through Audio Source

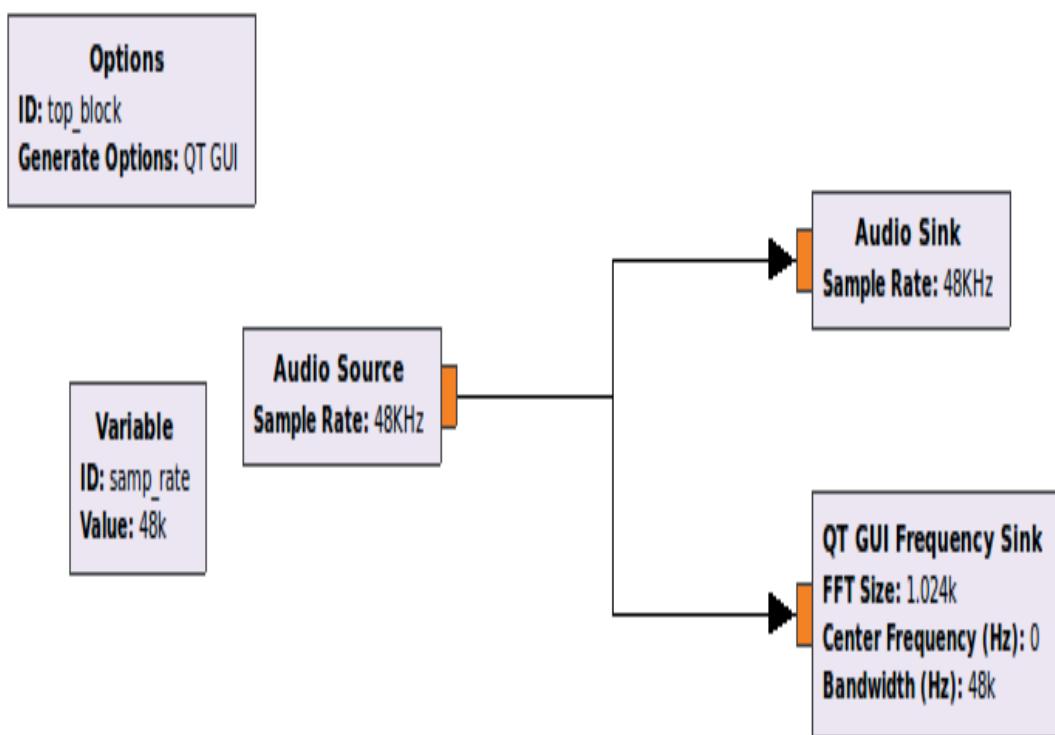


Figure 2.1.1 Loop back of an Audio Signal between Source and Sink

Audio Source block is used to capture inputs from hardware like Microphones ports on a PC. It reads audio input signals and samples it out at the sampling frequency of 48 KHz. Audio Sink block is a block which writes out the output to hardware like a speakers etc. So using these blocks one could create simple flow graphs that operates on audio data and also perform different analysis on the incoming signal and observe the behavior either within GNU radio or even write out the outputs to an audio port to listen into what happens.

Fig 2.1.2 represents a frequency response of a random signal which is captured from the microphones. In this case it seems to be an irregular signal as it does not contain any information containing signals through it. The carrier frequency is centered at 0 KHz which is observed in Fig 2.1.2.

Result

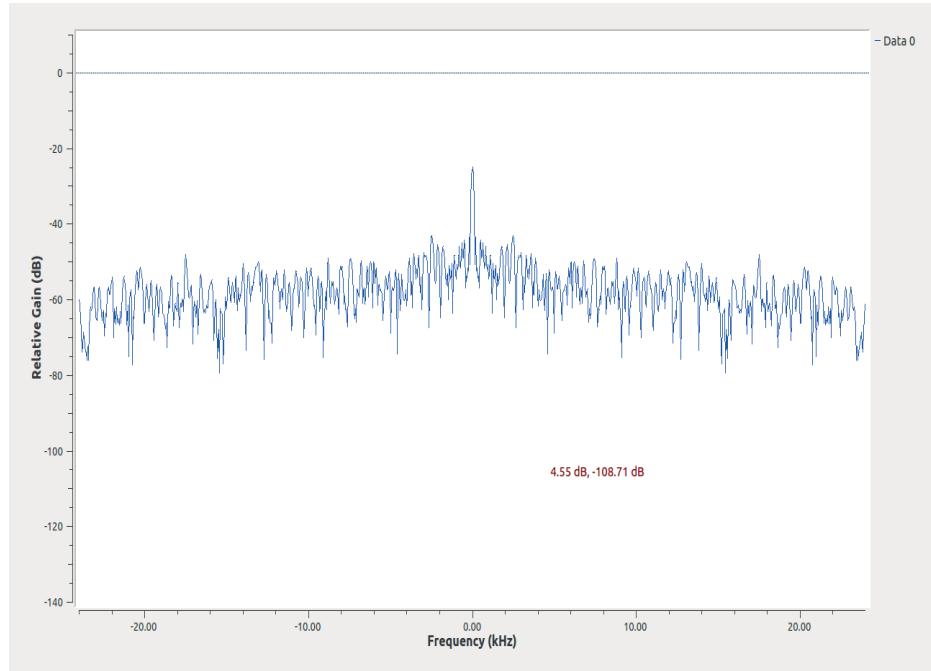


Figure 2.1.2 Random audio signals with frequency centered at 0 KHz

2. Transmission of audio signal through Wave File Source

Wav File Source/Sink are category of GRC blocks which helps in reading from an audio file stored on disk or write out an audio file which is stored in a particular extension (.wav) format which can be played using standard audio players on a host system. The recorded audio sound can be transmitted multiple times by applying repeating mode ON. The recorded audio signals contain some information which is delivered with relatively high Gain; hence the frequency response of a signal represented in Figure 4 does not resemble a flat response as earlier. The carrier signal is centered at 0 KHz frequency.

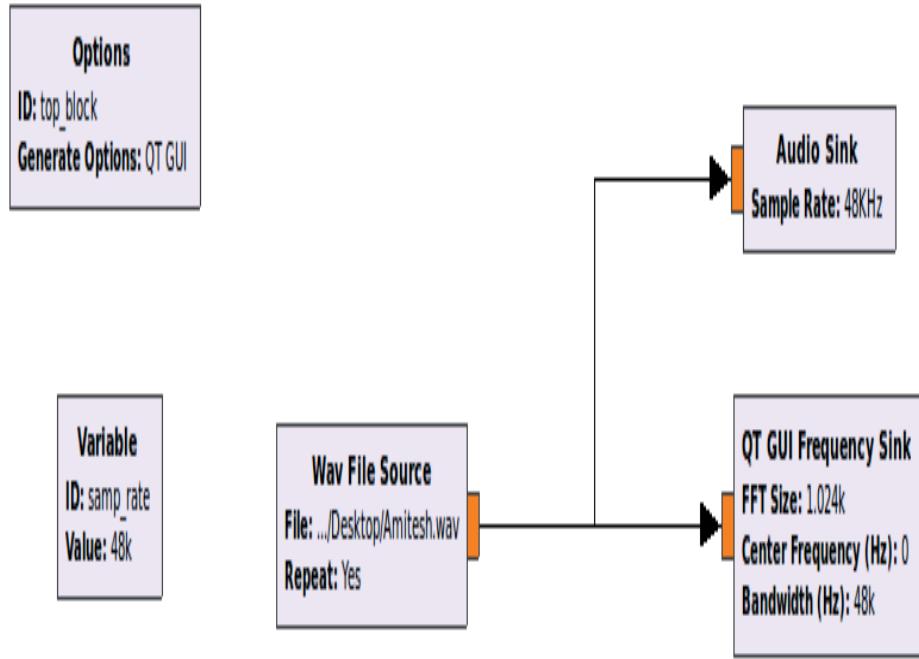


Figure 2.1.3 Transmission of signal through Wav File Source Flow graph.

Result

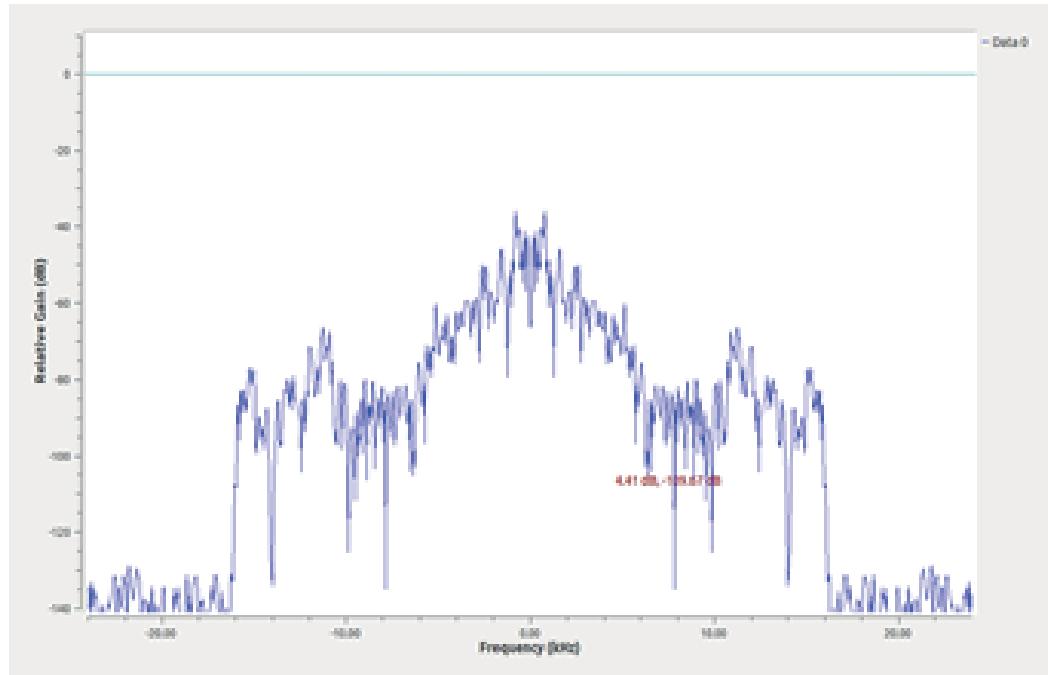


Figure 2.1.4 Audio signals with frequency centered at 0 KHz

Inference

This experiment aims in exploring audio source, Audio Sink and WAV file source blocks and their usage. Depending upon input, whether it is recorded or else by direct usage of microphone appropriate blocks could be used.

EXPERIMENT 2

Aim

The purpose of this experiment is to explore the importance of Throttle block in Gnu Radio.

Introduction

Throttle block is used to limit the amount of data signals passing through the system as per the sample rate. It protects the GNU Radio platform from excess consumption of the CPU resource which occurs when the flow graph is not being properly regulated by the external hardware like audio source, audio sink, USRP source, USRP sink.

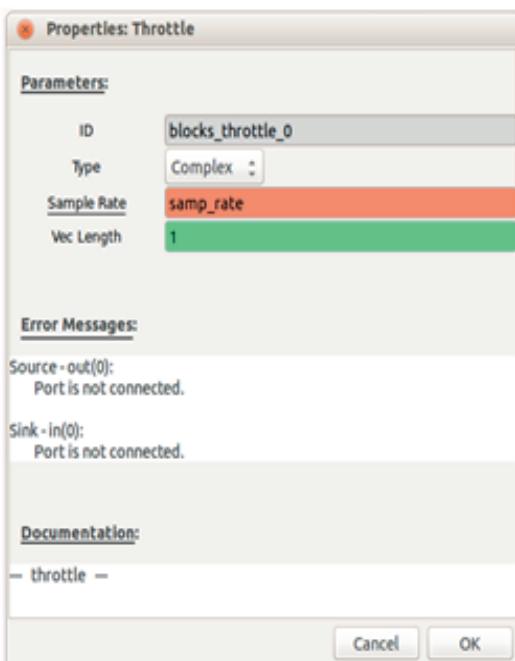
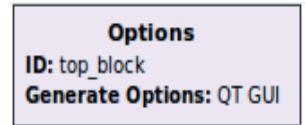


Figure 2.2.1 Throttle Block

The type in the block represents the types of representation of data which are explained as:

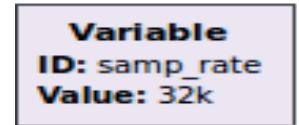
1. Complex: Input and output are represented in complex values (real +imaginary).
2. Float: Input and output are represented in real values.
3. Int: Input and output are represented in integer values.
4. Short: Input and output are represented in short integer values.
5. Byte: Input and output are represented in character form or byte values form.

Block explanation of the flow graph



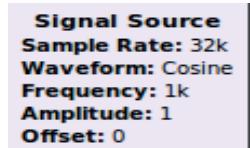
1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as QT GUI.



2. Variable

Variable block is used to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 32K for proper conversion of analog signals to discrete sampled signals.



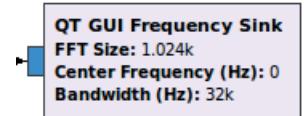
3. Signal source

Signal source is used to generate a cosine wave signal with carrier frequency of 1k, 2k and 3k respectively which is selected to have a proper discrimination between low frequency and high frequency signals. Amplitude of cosine wave never exceeds 1.



4. Throttle

Throttle is used to avoid excess consumption of CPU resources from the flow graph. In this experiment throttle block is used with the sample rate defined as 32 KHz.



5. QT GUI frequency Sink

GUI stands for Graphic user interface with standard QT. Frequency sink is used to represent the frequency plot for the desired output signals with FFT size selected as 1.024K by default. Here the carrier frequency is centered to 1 KHz.

Gnu Radio Flow graphs

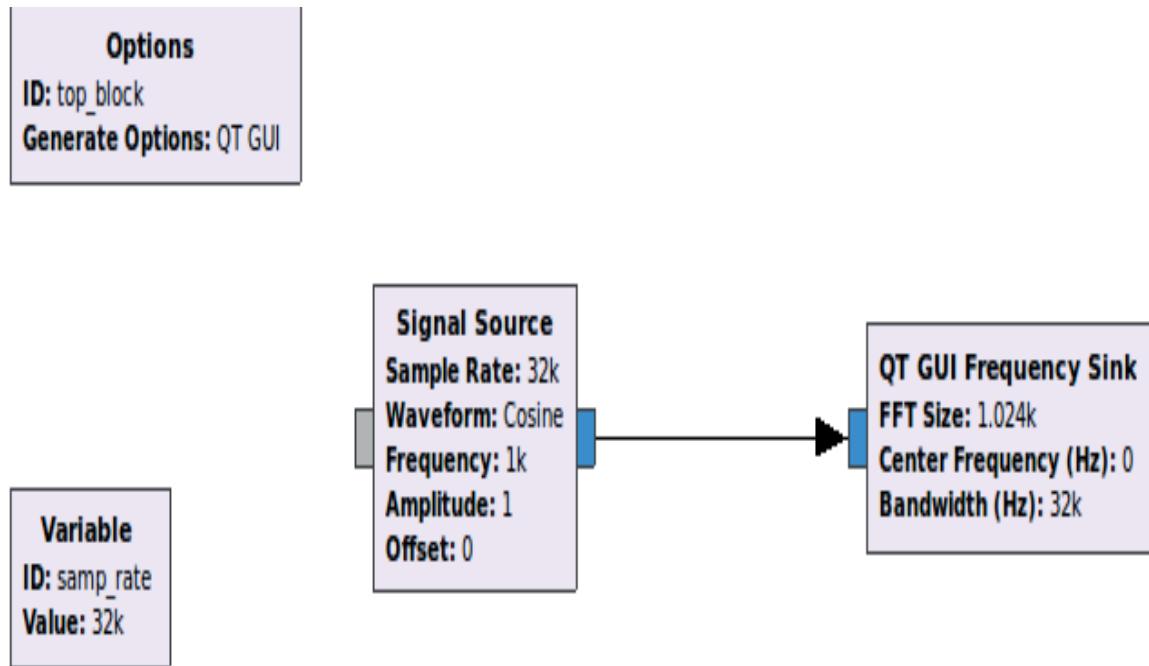


Figure2.2.2 Flow graph without Throttle block

Figure 2.2.2 represents flow graph without having a Throttle block. A signal source block is used as a cosine waves carrier signals with frequency of 1 KHz. A signal source is directly connected to the QT GUI Frequency Sink block to get the frequency response of a given signals.

Warning Message

```
>>>Done  
  
Generating: '/home/sdr-tenet/Desktop/top_block.py'  
>>>Warning: This flow graph may not have flow control: no audio or RF hardware blocks found. Add a Misc->Throttle block to your flow graph to  
avoid CPU congestion.  
  
Executing: /usr/bin/python2 -u /home/sdr-tenet/Desktop/top_block.py
```

Figure 2.2.3 Warning Message about the Throttle block

When the above flow graph is run, we do get the desired result in frequency domain with a warning message displayed as mentioned in Figure 2.2.3. The warning message informs to add a throttle block in the given flow graph which is used in to avoid the CPU congestion, Thus providing a perfect flow control on the given flow graph.

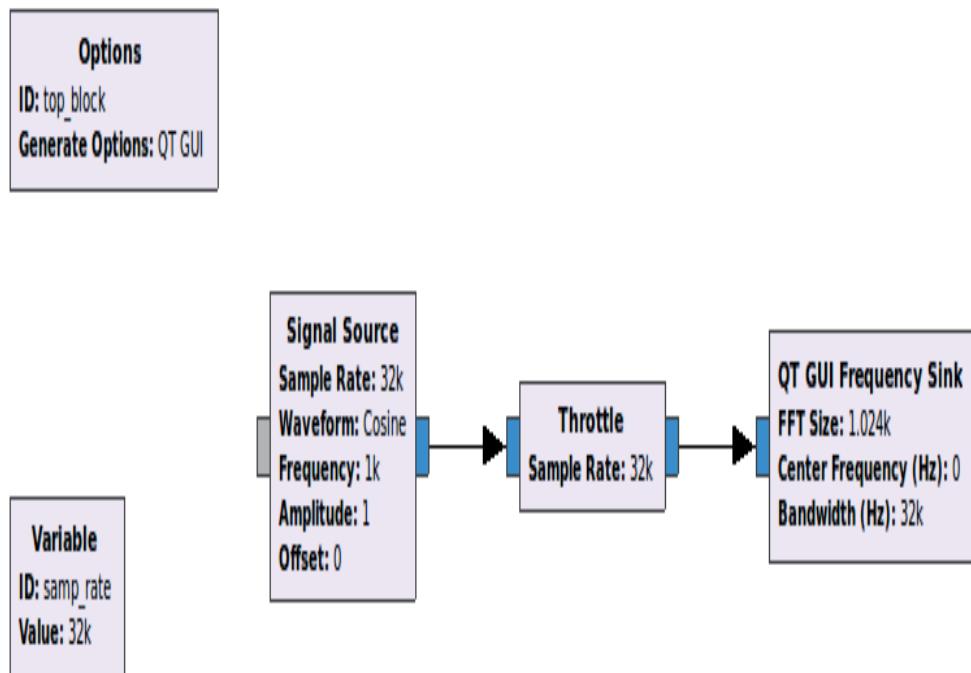


Figure 2.2.4 Flow graph with Throttle block.

Hence to avoid the CPU congestion, throttle block is connected to the given flow graph. The sample rate of 32 KHz is used to sample down the carrier signals of Frequency 1

KHz. Hence the desired result is obtained with no warning message generated. Thus, there is a perfect control in the flow of an experiment.

NO Warning Message

```
>>>Done
```

```
Generating: '/home/sdr-tenet/Desktop/top_block.py'
```

```
Executing: /usr/bin/python2-u /home/sdr-tenet/Desktop/top_block.py
```

Figure2.2.5 Flow graph is generated successfully with no warning messages

Results

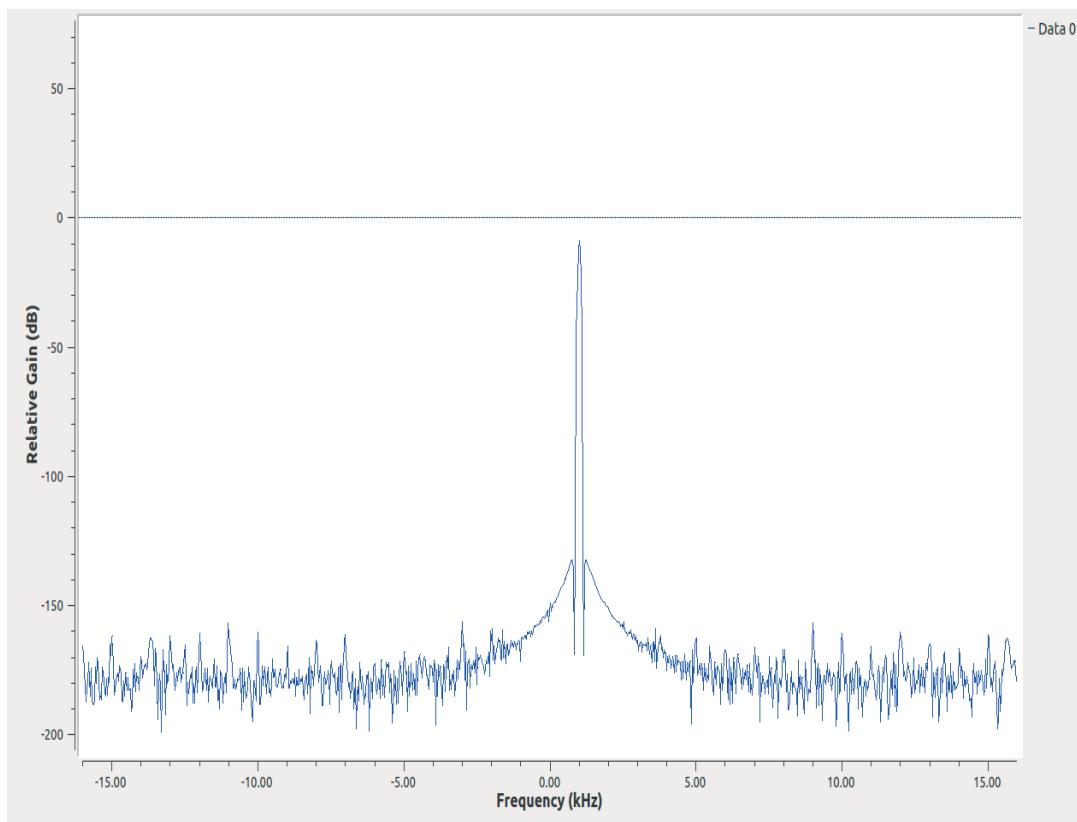


Figure2.2.6 Cosine wave signal centered at 1 KHz

Inference

This experiment depicted the usage of Throttle block and its importance in Gnu radio platform with the help of an experiment as mentioned above. Thus throttle block avoid the CPU congestion thus providing a perfect flow control in the flow graph.

EXPERIMENT 3

Aim

The purpose of the experiment is to identify the reason behind the usage of Packet encoder and packet decoder block for transmission and reception of real time signals in Gnu Radio.

Introduction

All real time signals (like audio, video, text, image signals) are analog in nature. Analog signals are continuous signals which cannot be transmitted directly through the wireless communication system, as it results in the loss of information delivered. Hence these analog signals are discretized using the sampling and quantization techniques. These sequences of digital values are then divided in various groups called as packet. The divisions of byte data packets are done in a sequence order so that there is no loss of data. The packet of data is further attached with the preamble and the unique access code to form a frame. The preamble allows the devices on the network to easily synchronize their receiver clocks; hence synchronization of frames can be achieved. The access code is the unique code which is given to each frame. It can help to receive the proper sequence of frames, so that there is no loss of information. It can also help in identifying the loss of frames due to the presence of conductive medium.

Blocks Description

1. Packet Encoder

Packet Encoder Block packetizes the data. Data are wrapped into a packet of a given payload length size with a header, access code, and preamble to form a frame. The header is just 2 times the repetition of the payload length. Payload length defines the size of the frame. Larger the payload size, greater is the efficiency in the reception of signals with less loss of information. The preamble and access code are left blank for the default case. Bits per symbol is defined as the number of bits transmitted per symbol. Bits/symbol changes as per the modulation scheme as it follows the mathematical calculation as $2^n = m$, where n is bits/symbol and "m" is the constellation points.

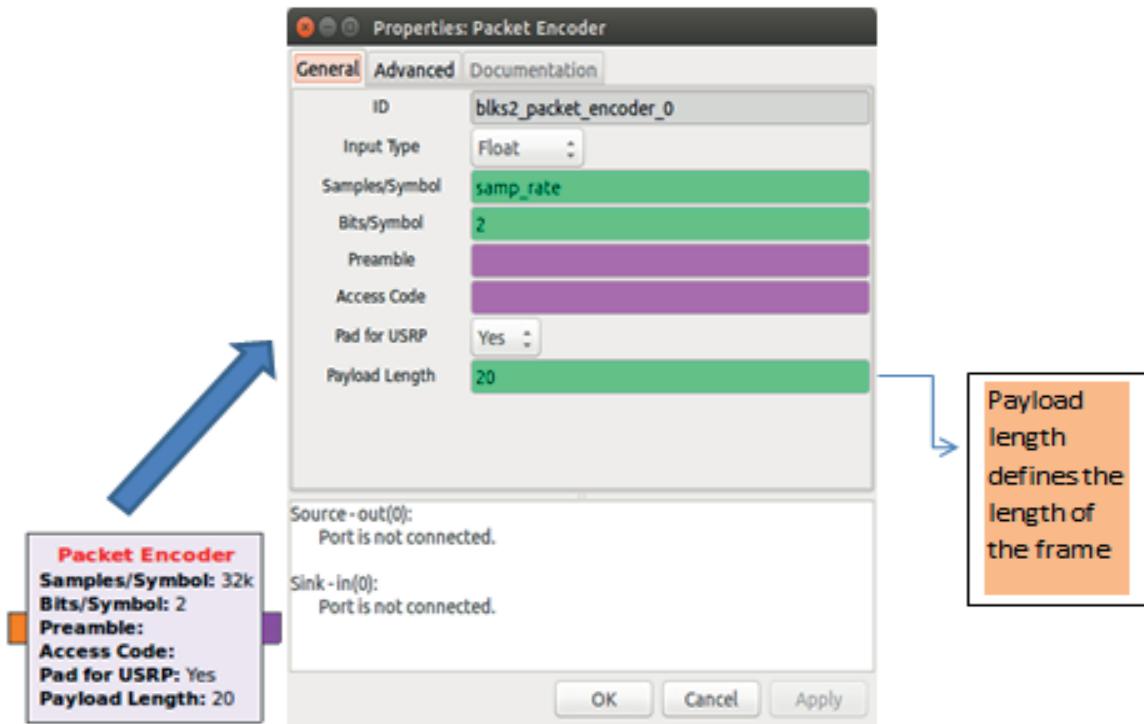


Figure 2.3.1 Specification of Packet Encoder

Bits/symbol for different modulation techniques are:

1. BPSK = 1 Bits/symbol
2. QPSK = 2 Bits/symbol
3. 8PSK = 3 Bits/symbol
4. 16QAM = 4 Bits/symbol
5. 32QAM = 5 Bits/symbol
6. 64QAM = 6 Bits/symbol
7. 128QAM = 8 Bits/symbol

2. Packet decoder

The Packet decoder identifies the access code with the number of bits received. When access code is received, then it reads the header to get the payload length. Hence extracts the payload and display it as an output.

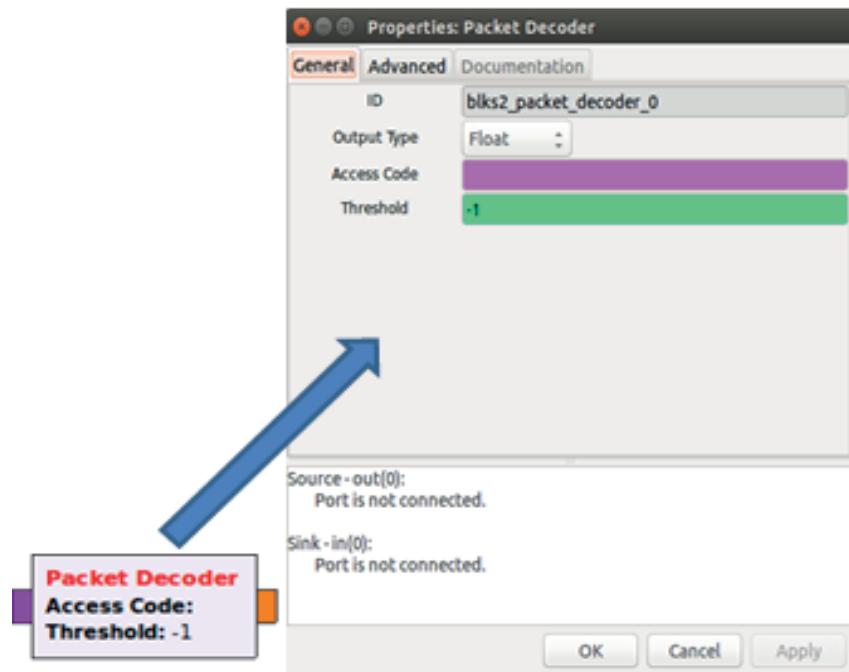


Figure 2.3.2 Specification of Packet Decoder

EXAMPLE

Consider an example of transmission and reception of an image through packet encoder and packet decoder in Gnu radio. The flow graph is as follows:

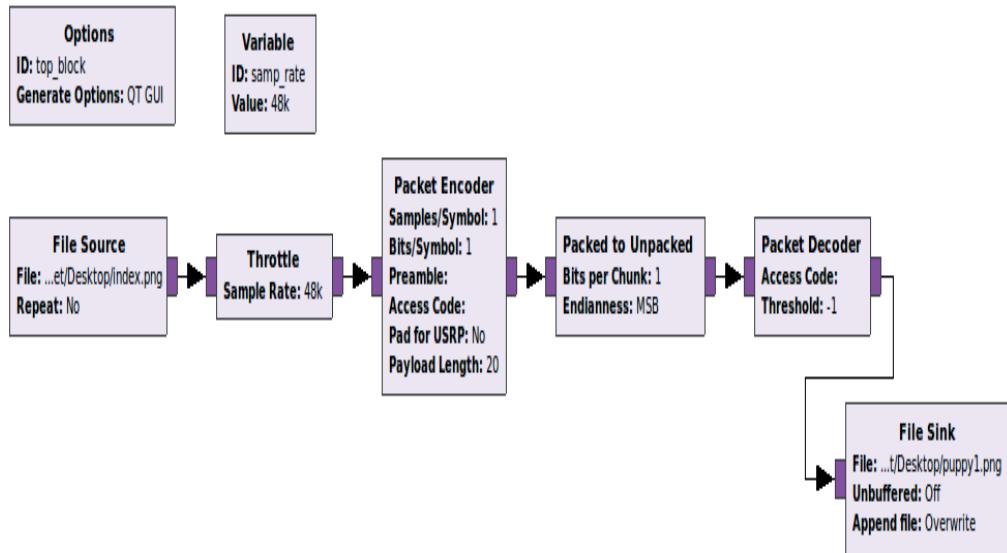


Figure 2.3.3 Transmission and reception of an image.

File source is used to read the (.png) image file which is passed through the packet encoder. Preamble and access code is left blank for the default case. The frames are then processed through Packed to Unpacked block. Packed to Unpacked block is used to convert a stream of packed bytes to stream of unpacked bytes. The streams of unpacked bytes are further being dismantled to set of data removing its preamble and access code. Once the data is processed and arranged properly in sequence then file sink is used to display the resulted received image. An empty filename with (.png) need to be created and its file location need to be specified in file sink. Payload length is considered as 20, based on the experimental setup and the type of an image.



Figure 2.3.4.A) Transmitted image. 2.3.4. B) Image after received.

Fig 2.3.4.A represents a transmitted (. Png) image and Fig 2.3.4.B is the received image. The error in the reception of an image signals occurs mainly when the payload length is considered to some lower value (I.e. the size length of the frame is less). Hence based on the image and try and error basis, the payload length is defined as 20. Hence the resulting error free image is obtained in Fig 2.3.4.B.

Inference

Hence this experiment provides a complete explanation about the importance of packed encoder and packed decoder block. It helps in creating a frame which includes the information bits along with the preamble and access code. Hence in such way, the real time signals can be transmitted and received with respect to wireless communication system.

EXPERIMENT 4

Aim

The purpose of this experiment is to explore the various types of Boolean blocks available in Gnu Radio.

Block Description

1. And gate

The AND gate is a basic logic gate which has a high output (i.e. 1) only, if all its inputs are high. The AND operation is represented by a dot (.) i.e. A.B. The truth table of AND gate is given below:

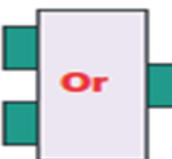
2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1



2. Or gate

The OR gate is a logic gate which has a high output (i.e. 1) only, if one or more inputs are high. The OR operation is represented by a plus (+). The truth table of OR gate is given below:

2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



3. Not gate

The NOT gate also known as an inverter which produces an inverted version of the input at its Output. The truth table of NOT gate is given below:

NOT gate	
A	\bar{A}
0	1
1	0



4. Xor gate

The XOR gate is also termed as an 'Exclusive-OR' gate which has high output only if the two inputs are same. An encircled plus sign (\oplus) is used to show the EXOR operation. The truth table of XOR gate is given below:

2 Input EXOR gate		
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



5. And Const

And Const implements the function output = in AND const. This can be useful to mask certain bits in a byte, hence forcing them to zero.



Inference

Hence this experiment successfully provides a complete description about the Boolean blocks available in Gnu radio.

EXPERIMENT 5

Aim

The purpose of this experiment is to explore the various types of converters available in Gnu Radio.

Introduction

There are different sets of type conversion blocks available in Gnu radio. The table below represents the type of conversion and its color indication in Gnu Radio.

Table 5: Colour Indication for different data types

Type Conversion	Colour Indication
Int	GREEN
Float	ORANGE
Char	PINK
Complex	BLUE
Short	YELLOW

Block Description



1. Int to Float

Int to float block is used to convert the binary integer data values to the decimal float values. It specifies the vector length for processing a vector which is by default set as 1. The scale factor is considered as 1 in order to normalize a sine wave.



2. Float to Int

Float to Int block is opposite of Int to Float block. It is used to convert the decimal float value to the binary integer values.



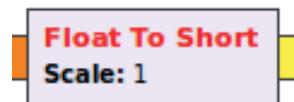
3. Float to Char

Float to char block is used to convert the decimal values to character values or byte values. The char values are represented in ASCII values.



4. Float to Complex

This block is used to convert the decimal float values to the complex values (i.e. it converts the decimal values to the real term and the imaginary term). The complex numbers are always expressed as $x = r_1 + i \cdot r_2$, Where r_1 is the real number and r_2 is considered to be an imaginary values.



5. Float to Short

It converts a floating point (real) value to a 16-bit short integer. Float to Short block sets the scale factor as 1 in order to normalize a sine wave to the full range of short integer. Values outside the range [-32768, 32767] will get saturate.



6. Char to Short

It converts the character values (I.e. byte values) to 16-bit short integer values.



7. Char to Float

It converts the character values to the decimal float values.



8. Short to Char

It converts the 16-bit Short integer values to the character values.



9. Complex to Imag

It converts the complex signals (I.e. real signal + imaginary signals) to the Imaginary signals ignoring the real signals.



10.Complex to Mag

It converts the complex signals to the magnitude of signals. The magnitude of signals is expressed as $M = \sqrt{\text{real}^2 + \text{imaginary}^2}$.



11.Complex to Real

It converts the complex signals (i.e. real signal + imaginary signals) to the real signals ignoring the imaginary signals.



12.Complex to Mag2

It converts the complex signals to the square of magnitude of signals. The square of magnitude of signals is expressed as $M=(\sqrt{\text{real}^2 + \text{imaginary}^2})^2$.



13.Complex to Arg

It implements the function as $\text{out} = \text{Arg}(\text{in})$. The output value is in radians from -pi to pi. The input data type is complex while the output value is floating point.

Inference

Hence this experiment provides a complete description about the type of data conversions available in Gnu radio.

EXPERIMENT 6

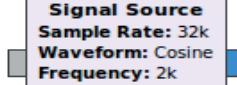
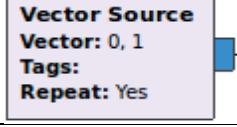
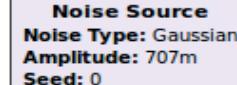
Aim

The purpose of this experiment is to explore the various types of input source blocks available in Gnu Radio.

Introduction

There are 4 different types of type's input sources available in Gnu radio as shown in table:

Table 6: Different types of signal blocks

Input Source	BLOCKS
Signal Source	 Signal Source Sample Rate: 32k Waveform: Cosine Frequency: 2k Amplitude: 1 Offset: 0
Random Source	 Random Source Minimum: 0 Maximum: 25 Num Samples: 1k Repeat: Yes
Vector Source	 Vector Source Vector: 0, 1 Tags: Repeat: Yes
Noise Source	 Noise Source Noise Type: Gaussian Amplitude: 707m Seed: 0

Blocks Description

1. Signal Source

Signal Source Block is a type of input signal processing block which include a set of variables like sample rate, frequency and amplitude as an input, to produce an output. It

is used to generate a Sine wave, Cosine wave, Square, Triangle, Constant and Sawtooth signal. Signal Source is capable to create an input signals with Int, Float, Complex, Short data type.

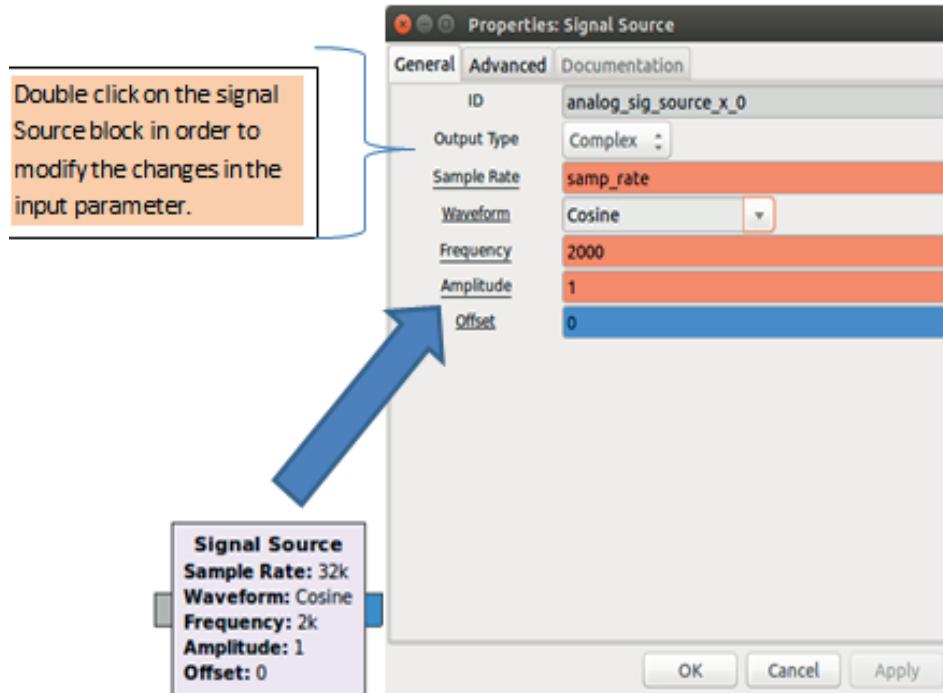


Figure 2.6.1 Input source block in Gnu Radio

1. Int type provides an output with 32 bit integer.
2. Float type provides an output with decimal real value.
3. Complex type provides an output with both real and an imaginary term.
4. Short type provides an output with 16 bit integer.

All the types of signal generated are limited to 1 as an amplitude i.e. [-1, 1]. The frequency is the carrier frequency which is set to 2 KHz. An error will occur if the carrier frequency is defined higher than half of the sample rate. The offset which represent a complex output specifies the offset which is added to the waveform generated.

2. Random Source

It is used to generate the number of samples with random numbers. The random numbers are generated within the range defined as minimum and maximum. The

number of samples can be made repeating by allowing the repeat mode ON. It can define the output samples in Int, Short, byte data type.

1. Int type provides an output with 32 bit integer.
2. Short type is used to generate output with 16 bit integer.
3. Byte type is used to generate an output of 8 byte character value.

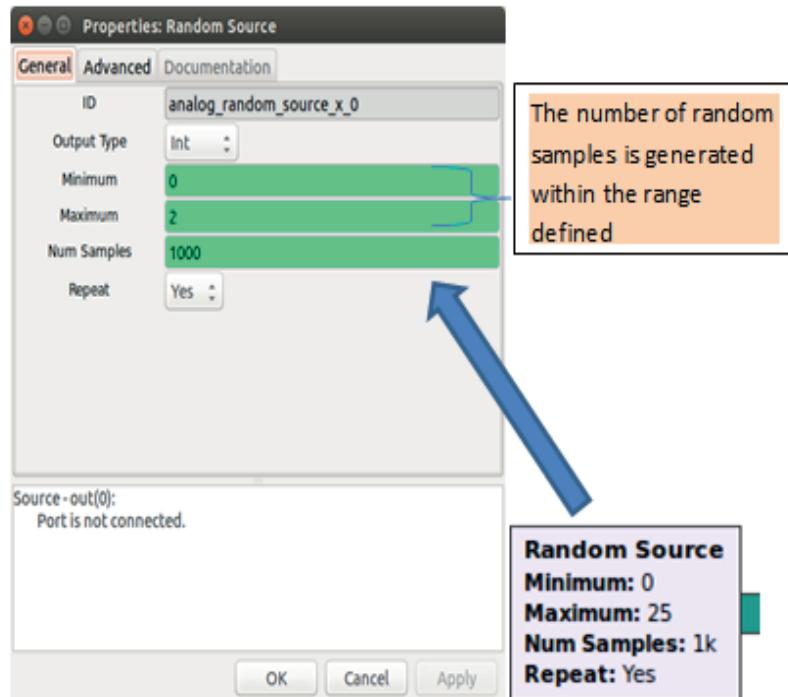
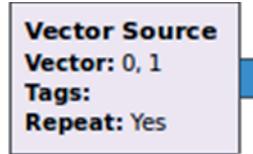


Figure 2.6.2 Random source block in Gnu Radio

3. Vector Source

It is a type of block which is used to represent a vector signals with the amplitude value defined in the vector slot. The number of vector signals can be made repeating by allowing the repeat mode ON. It can define the output samples in Int, float, Complex, byte data type.

1. Int type provides an output with 32 bit integer.
2. Float type provides an output with decimal real value.
3. Complex type provides an output with both real and an imaginary term.
4. Byte type is used to generate an output of 8 byte character value.



4. Noise Source

It is used to implement noise signals. The distribution of the noise can be selected from various standard distributions. There are various types of noise like Gaussian noise, uniform noise, Laplacian noise, Impulse noise.

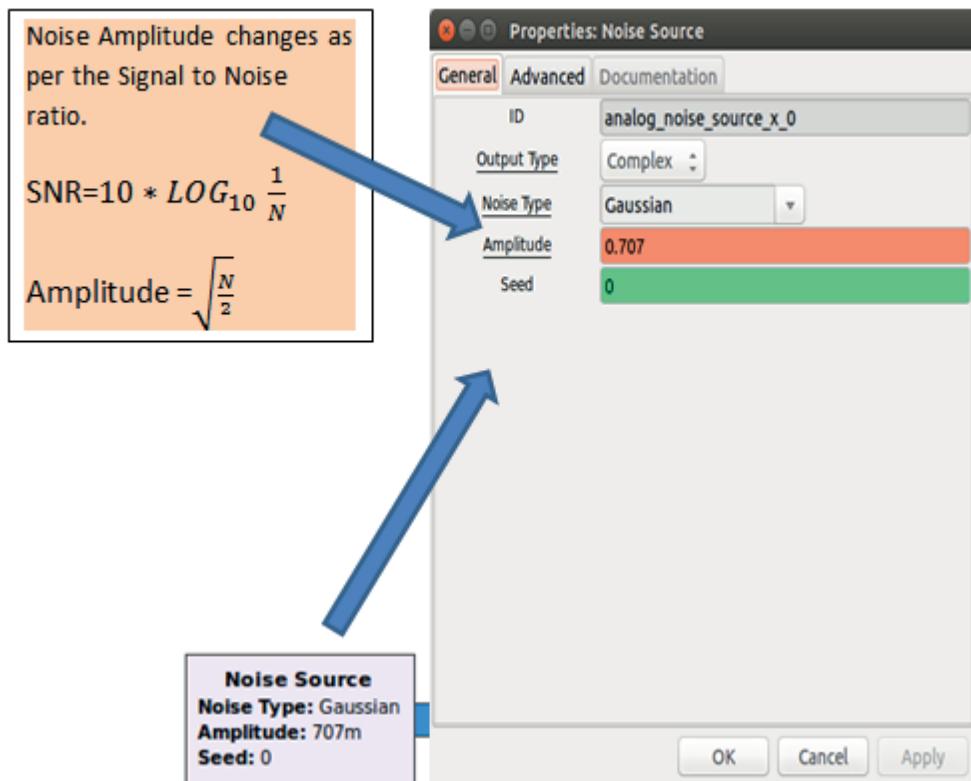


Figure 2.6.3 Noise source block

1. Uniform Noise: It is used to get the uniform distribution.
2. Impulse Noise: It is used to provide an impulse distribution.
3. Laplacian Noise: It is used to select the Laplacian distribution.
4. Gaussian Noise: It is used to select the Gaussian distribution.

Noise Amplitude changes with change in the Signal to Noise ratio (SNR). The Mathematical relation between the amplitude and snr is expressed in the Fig 2.6.3.

Example

If N= 1 then

$$\text{SNR} = 10 * \text{LOG}_{10} \frac{1}{1} = 0 \text{ dB},$$

$$\text{Hence Amplitude} = \sqrt{\frac{0}{2}} = 0.707$$

Hence further 0dB, 1dB15dB will have noise amplitude as 0.707, 0.630, 0.562... 0.125.

Inference

Hence this experiment successfully provides complete information about the type of input source blocks available in Gnu radio.

EXPERIMENT 7

Aim

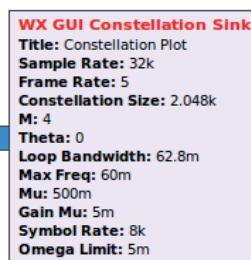
The purpose of this experiment is to explore the various WX GUI blocks available in Gnu Radio.

Blocks Description

GUI stands for Graphic user interface with backend standards as WX. Hence it is also called as WX-based graphical sink the takes a complex stream and display it in frequency domain, waterfall diagram, time domain, constant plots, and constellation diagram. The lists of WX GUI Standards are listed below:

1. WX GUI Constellation sink

It is a graphical sink which consider a set of a complex streams and display them on IQ constellation. It can plot the streams only in complex format. WX GUI constellation sink is used for plotting the constellation diagram for various digital modulation schemes. Constellation diagram represents the number of signals components modulated by a digital modulated scheme. It displays the scheme in 2 dimensional scatter diagrams. Sample Rate, Frame Rate, Window size can be manually adjusted. Notebook section is used to gives a display the graphical element in the Notebook page.



2. WX GUI FFT sink

It is a graphical sink used to display the multiple signals in frequency domain. It can plot the streams of data in Float and complex format. The different Windowing option can also be selected based on the experimental setup required. A Blackman windowing

technique is considered as best Windowing techniques as compared to other techniques due to its minimized side lobes property. Grid position is used to position the graphical element in the window. Notebook section is used to gives a display the graphical element in the Notebook page.



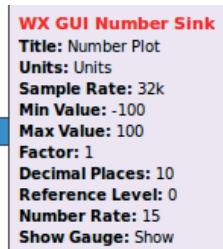
3. WX GUI Histo sink

It is WX based graphical sink which displays the histogram of a data. The purpose of histogram is to graphically summarize the distribution of data. It can plot the data set in float data format represented by orange color in GRC (Gnu radio companion).



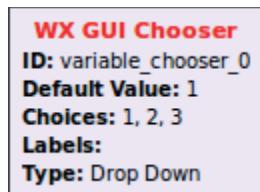
4. WX GUI Number sink

It is a graphical sink which is used to display the scalar value or a numerical value for the set of input data streams. It displays the value in float, complex format. The output is the average of various input data streams. The incoming numbers are multiplied by the factor, and then added to by the reference level.



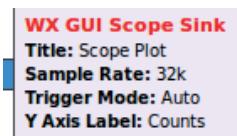
5. WX GUI Chooser

It creates a variable with a radio button, drop down. It is used to provide a multiple choice between the multiple options. Leave the label blank for the default case.



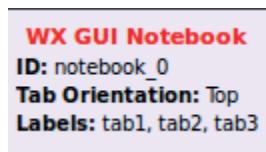
6. WX GUI Scope sink

It is a graphical sink, which is used to display the multiple signals in time domain. It can plot the output in Complex and Float format. The trigger mode is used to analyze the time domain signals in different mode (like Normal, Freerun, and Stripchart).



7. WX GUI Notebook

Notebook section is used to gives a display the graphical element in the Notebook page. Grid position is used to position the graphical element in the window.



8. WX GUI Waterfall sink

It is a graphical sink, which display multiple signals on spectrogram plot. It takes either set of floating point or complex streams and plots them on a waterfall plot or spectrogram plot. It does not support multiple input streams. Waterfall diagram can be analyzed on different Windowing techniques.

WX GUI Waterfall Sink
Title: Waterfall Plot
Sample Rate: 32k
Baseband Freq: 0
Dynamic Range: 100
Reference Level: 0
Ref Scale (p2p): 2
FFT Size: 512
FFT Rate: 15
Freq Set Varname: None

9. WX GUI Slider

It is a block which creates a variable with a slider. It provides a wide range of slides from minimum to maximum values. The values defined must be an integer real value .The number of steps must be between 0 and 1000.The Grid is used to position the graphical element in the Window.

WX GUI Slider
ID: variable_slider_0
Default Value: 50
Minimum: 0
Maximum: 100
Converter: Float

10. WX GUI Text box

It creates a variable with a text box. Default value is zero for default condition It can convert the text in float, integer, string, and Evaluate. Label can be anything or else can be left blank. It is just a labelling to the textbox.

WX GUI Text Box
ID: variable_text_box_0
Default Value: 0
Converter: Float

11. WX GUI Check Box

It creates a variable with check box form. Label can be anything or else can be left blank. A check box switches between the two states. The default being true and false. Override True and False to use the alternative states.

WX GUI Check Box
ID: variable_check_box_0
Default Value: True
True: True
False: False

12. WX GUI Static Text

This block creates a variable with a static text form. It can convert in Float, Integer or else in String format. Format should be a function that converts a value into a string .It can be set to none for the default formatter.

WX GUI Static Text
ID: variable_static_text_0
Default Value: 0
Converter: Float

Inference

Hence this experiment describes a complete information about the different types of WX based Graphics User Interface standards and their roles in Gnu radio companion. WX is the older versions of GUI which has an issues that the entire instrumentation block performs very poor especially on the integrated graphic cards. In various plots like FTT plot , scope plot, the graphs does not provide a wide range from negative to positive but only covers the from 0 to positive range. Hence to overcome this problem, a QT GUI standard is needed which is well elaborated in the next experiment.

EXPERIMENT 8

Aim

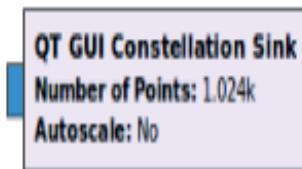
The purpose of this experiment is to explore the various QT GUI blocks available in Gnu Radio.

Blocks Description

GUI stands for Graphic user interface with backend standards as QT. Hence it is also called as QT-based graphical sink the takes a complex stream and display it in frequency domain, waterfall diagram, time domain, constant plots, and constellation diagram. The lists of QT GUI Standards are listed below:

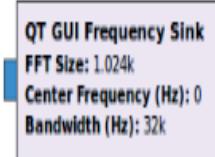
1. QT GUI Constellation sink

It is a graphical sink which consider a set of a complex streams and display them on IQ constellation. It can plot the streams either in complex data or complex message format. QT GUI constellation sink is used for plotting the constellation diagram for various digital modulation schemes. Constellation diagram represents the number of signals components modulated by a digital modulated scheme. It displays the scheme in 2 dimensional scatter diagrams.



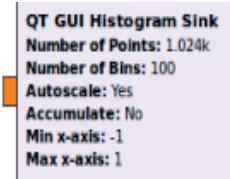
2. QT GUI Frequency sink

It is a graphical sink used to display the multiple signals in frequency domain. It can plot the streams of data as Float data /float messages and Complex data /complex messages. Frequency sink tunes the center frequency in such a way that the carrier signals will lies within the bandwidth (BW) defined. (By default BW is equal to sample rate). Comparisons between the multiple frequency responses can be achieved by increasing the number of inputs.



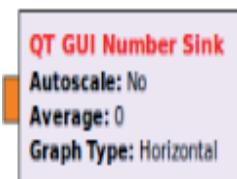
3. QT GUI Histogram sink

It is QT based graphical sink which displays the histogram of a data. It allows setting and changing the number of points at once at runtime. The purpose of histogram is to graphically summarize the distribution of data. It can plot the data set either in float data or float messages represented by orange color in GRC (Gnu radio companion). QT GUI Histogram sink block has a bin which is used to sort the data in to given order mentioned.



4. QT GUI Number sink

It is a graphical sink which is used to display the scalar value or a numerical value for the set of input data streams. It displays the value in Int, float, short, byte data type. The output is the average of various input data streams, hence no need of averaging the output further.



5. QT GUI sink

It is a graphical sink, which can plot the output in time domain, Frequency domain Spectrogram (waterfall diagram). There is an option available to OFF any one of the domain output mentioned above. It tunes the center frequency mentioned so that the signals falls within the desired BW.

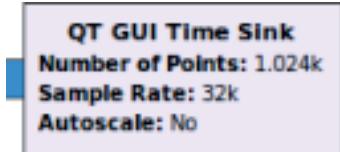


6. QT GUI Time sink

It is a graphical sink, which is used to display the multiple signals in time domain. It can plot the output in

1. Complex data/complex messages
2. Float data / float messages

Multiple input data signals can be compared in time domain by increasing the number of inputs.



7. QT GUI Vector sink

It is a graphical sink, which is used to display the multiple vector signals. It plots vectors of data as it is with each signal plotted with different color.



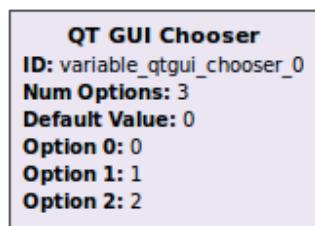
8. QT GUI Waterfall sink

It is a graphical sink, which display multiple signals on spectrogram plot. It takes set of floating point streams and plots them on a waterfall plot or spectrogram plot. It does not support multiple input streams.



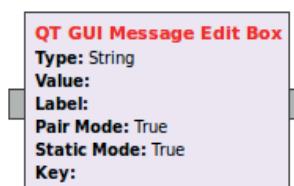
9. QT GUI Chooser

It is a block which chooses the number of options available in chooser. Each option has its own label. It set a number of options to “list” to enter a single list of options and labels. When the label is an empty list, the options will be used as label. It can have Int, Float, and String type of data.



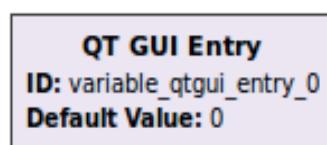
10. QT GUI Message edit box

It creates QT edit box, where the value is posted as a message. When data types are checked and found to be in the wrong format, then WARN log message is produced. Failures are either produces as log messages or the action is simply dropped. It can support String, Int, Float, Double, Complex in both scalar and vector format.



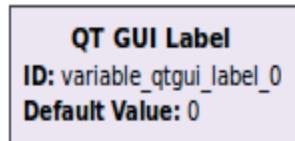
11. QT GUI Entry

It creates a variable with a text entry box. If label is left empty, then variable Id is used as label. It supports Float, Int, String, Boolean data type.



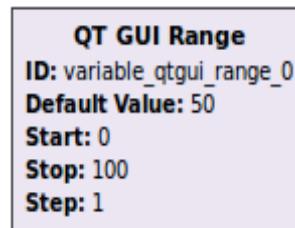
12. QT GUI Label

It is a block designed to create a variable with a label widget for text. If the label is left empty, then variable Id is used as label. It supports Float, Integer, String, Boolean data type.



13. QT GUI Range

It creates a variable with slider. The variable must be a value with real integer number. The value must lie between the start and the stop range mentioned. If the label is left empty, then the variable Id is used as the label.



14.QT GUI Tab widget

It creates a tab widget which organizes the other widgets. “Tab_id “can be used in GUI hint, in order to represent the other widgets.



15. QT GUI Push button

It creates a variable push button. If the label is left empty, then the variable Id is used as the label. This block selects two values of similar type and the selection of value is performed based on whether the button is pressed or released.

QT GUI Push Button
ID: variable_...push_button_0
Default Value: 0
Pressed: 1
Released: 0

Inference

Hence this experiment successfully explains about the different types of QT based Graphics User Interface standards and their roles in Gnu radio companion.

Chapter 3

Getting familiar with signal processing

This chapter deals with fundamentals of signals processing concepts especially signal generation, adding different kinds of noise to signals to analyze various behaviors and measure/analyze parameters like signal to noise ratio (SNR), Bit error rate (BER) with different Noise amplitude. In addition to this the reader could use this chapter to get a quick walkthrough on fundamentals like Nyquist sampling and its effects leveraging a list of experiments that are demonstrated to practically understand the concepts. A list of the same is presented below.

Table 7: List of Signal Processing Experiments

Sr.No	Description
1	Experiment to generate sine, cosine, square, triangle, sawtooth wave in Gnu radio
2	Experiment to analyze the effects of types of Noises on cosine wave signal in Gnu radio.
3	Experiment to implement the Quantization of input signals in Gnu radio.
4	Experiment to implement the Nyquist sampling theorem in Gnu radio.
5	Experiment to implement an aliasing error in Gnu radio
6	Experiment to implement the frequency shifting of input signal in Gnu radio.
7	Experiment to analyze the Gaussian Noise signal with change in signal to Noise ratio (SNR) in Gnu radio.
8	Experiment to analyze the Bit error rate (BER) in Gnu radio.
9	Experiment to analyze the oversampling perfect sampling and under sampling of a signal in Gnu radio

Experiment 1

Aim

The main purpose of this experiment is to analyze different types of signals generated using Signal Source block using Gnu radio.

Introduction

Signal source block in Gnu radio as discussed earlier in previous chapter, describes about the generation of different types of signals. In this experiment we will analyze different signals and evaluate the results obtained from this block.

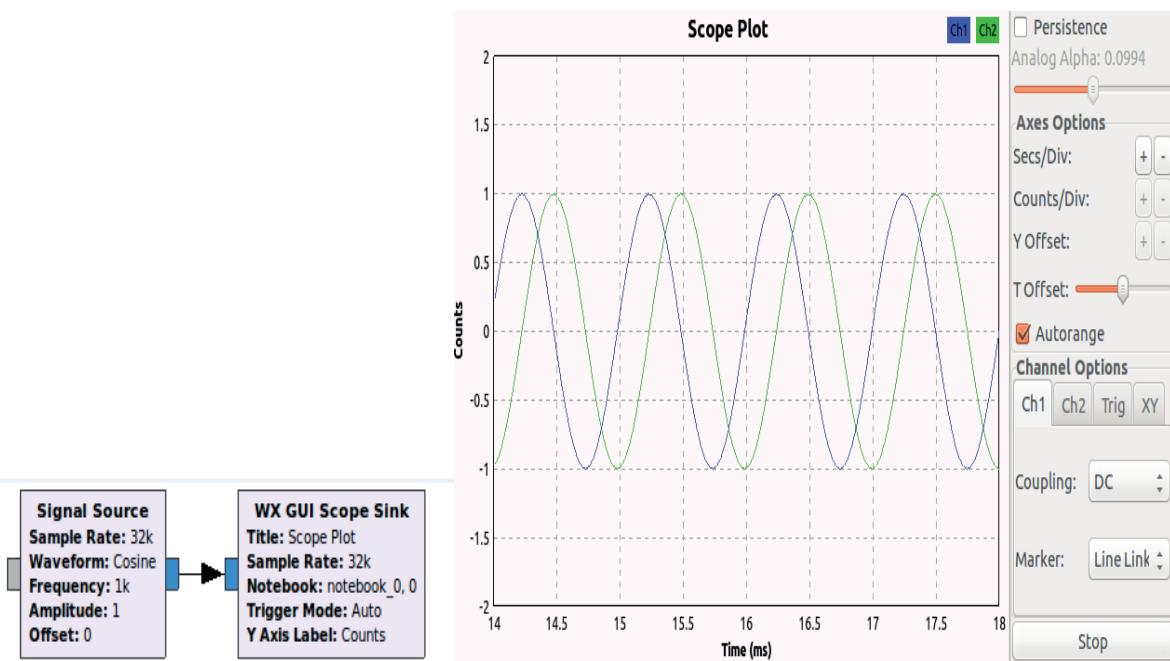


Figure 3.1.1 Generation of cosine wave.

Fig above represent the generation of cosine wave signals with a carrier signal of 1000Hz.WX GUI Scope sink is used to generate a signal in time domain. Hence the result is obtained as shown above. The result above is represented in complex format with blue color representing a real time signal and green color indicating an imaginary signal. The result can also be generated in real time signals only with selection of float data type. Fig 3.1.2 represents a sine wave generation of signals with carrier signals of 1000 Hz.

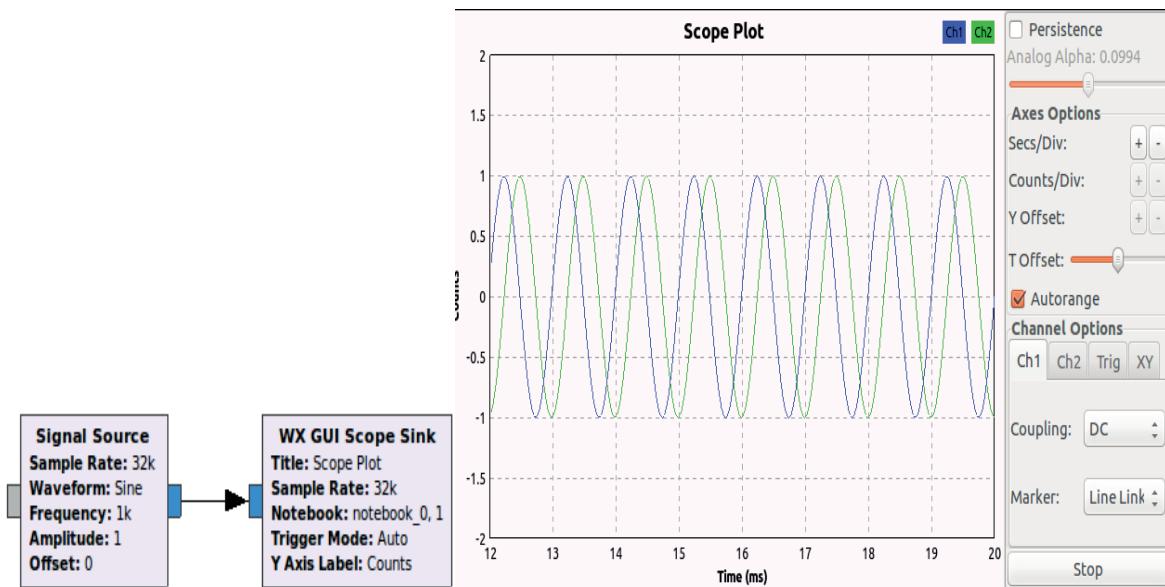


Figure 3.1.2 Generation of sine wave.

It is observed that the sine waves are identical to cosine wave but the only difference is that cosine wave lead sine wave by 90° . Hence sinewaves starts with 0 amplitude and cosine waves starts with high amplitude. Here it is assumed to be 1. Fig 3.1.3 represents the generation of square signals with a carrier signals of 1000 Hz.

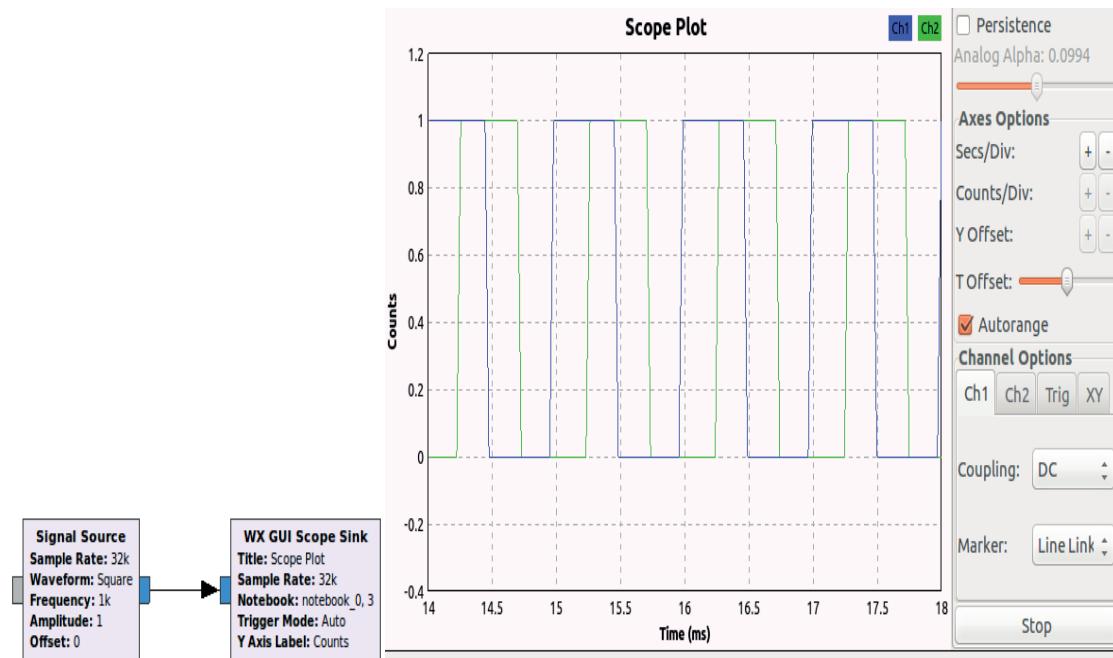


Figure 3.1.3 Generation of Square wave signals.

Fig 3.1.3 represents the generation of square waveform. It is representation of signals in complex domain with real and imaginary term. There is always phase difference between the real and imaginary term. Square wave lies between the amplitude from 0 to 1.

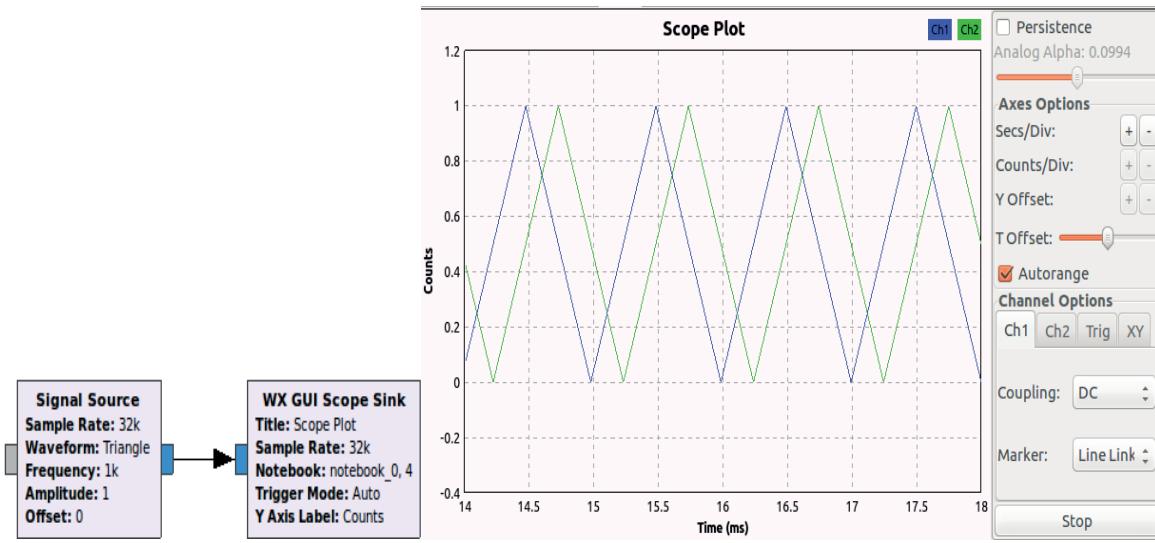


Figure 3.1.4 Generation of Triangular wave signals.

The above fig represent the generation of triangular waves. From the Fig it can be observed that the triangular wave is a type of continuous wave with linear and periodic in nature. It can be obtained by the integral of square wave signals.

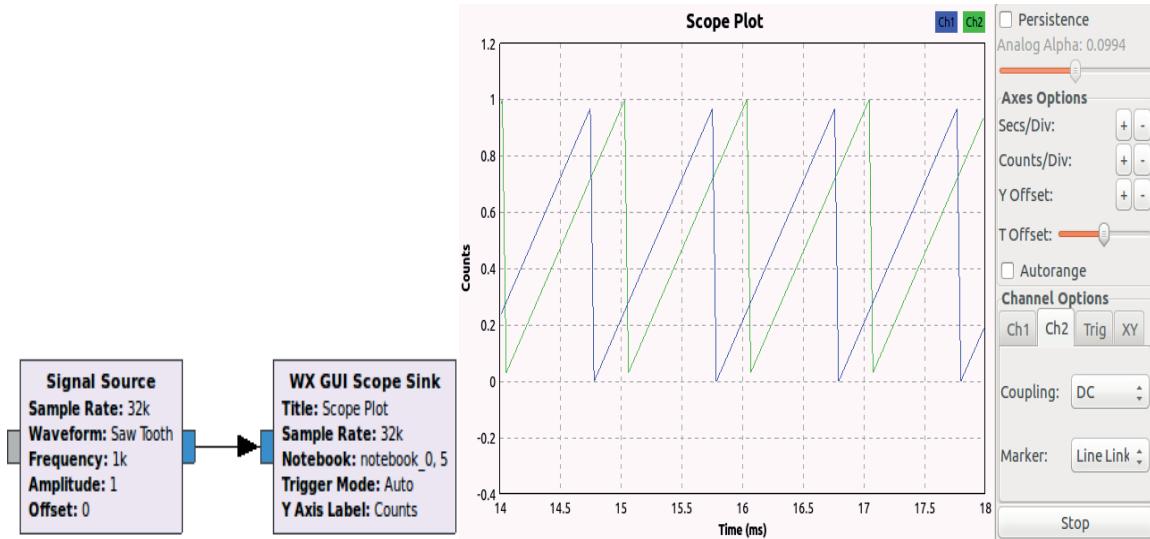


Figure 3.1.5 Generation of Sawtooth wave signals.

The Fig 3.1.5 represents a sawtooth wave signals with a carrier signals of 1000 Hz. The Sawtooth waveform also called as saw waveform is a type of continuous and a non-sinusoidal waveform. Sawtooth waveform is used in music where it helps in generating sound.

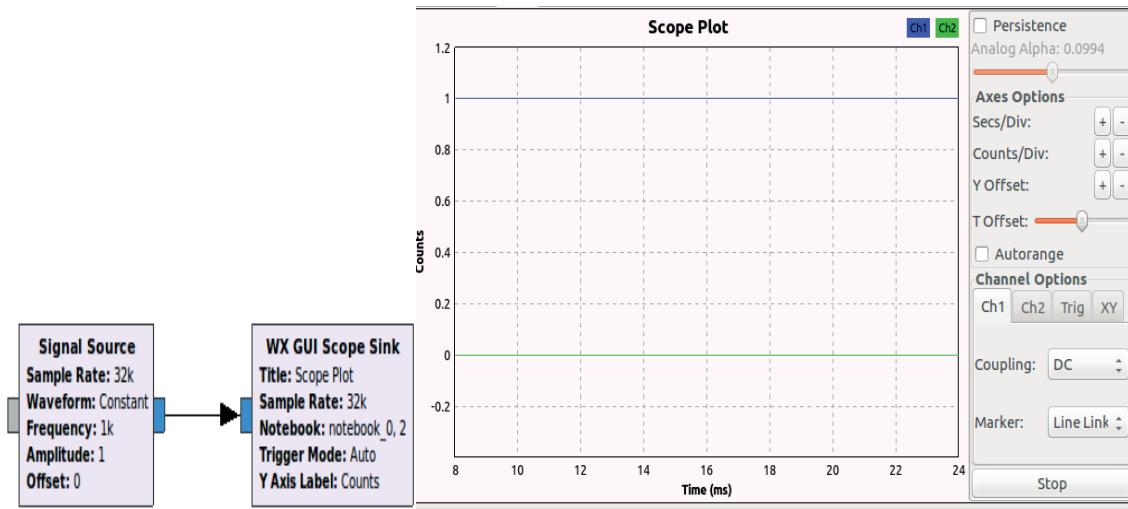


Figure 3.1.6 Generation of Constant signals.

It is a type of signal with constant amplitude 1; hence it is termed as the constant signals.

Inference

Hence this are the types of signals which can be generated by the signal source block in Gnu radio. The visual effects of each type of signals can be analyzed within the amplitude range from 0 to 1.

EXPERIMENT 2

Aim

The main purpose of this experiment is to generate different types of Noise signals in Gnu Radio.

Introduction

This experiment mainly deals with the generation of different types of Noise signal possibly available in the Noise source block. The impact of these types of noise is analyzed using Histogram and representation of signals in time domain. The block includes different types of noises which are as possible.

1. Laplacian Noise
2. Gaussian Noise
3. Impulse Noise
4. Uniform Noise

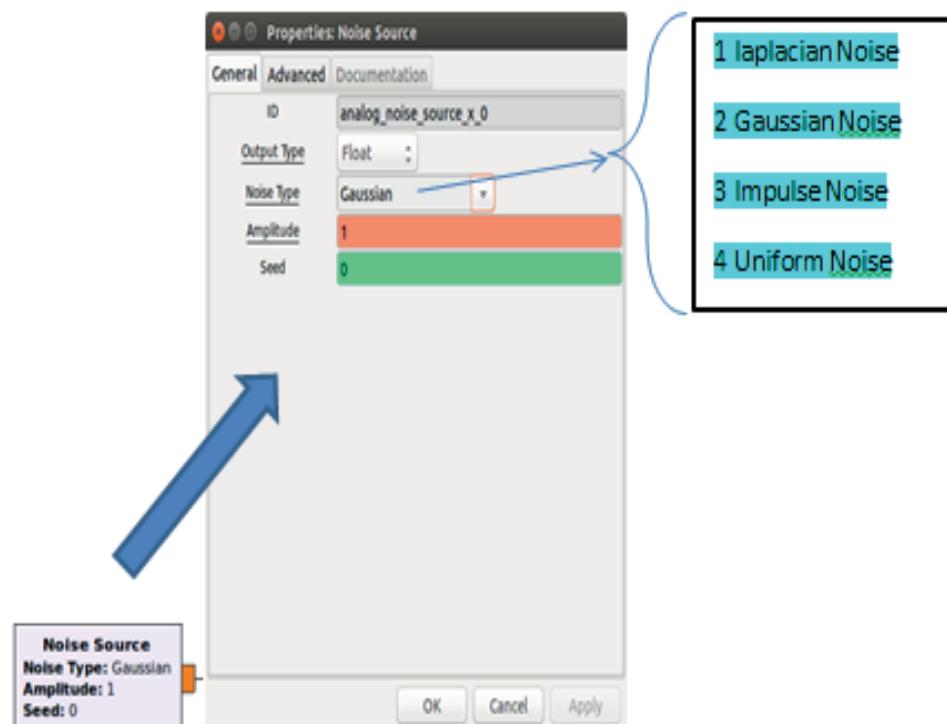


Figure 3.2.1 Noise Source Block

1. Laplacian Noise

Laplace distribution is mathematically defined as a function which provides the probabilities of occurrence of different possible outcomes in an experiment. Hence it is called as continuous probability distribution. The Laplace Noise simply defines the addition of variable Y with input variable X having Laplace distribution.

$$F(x) = X(x) + Y(x)$$

Where Y(x) represent a variable containing Laplace distribution.

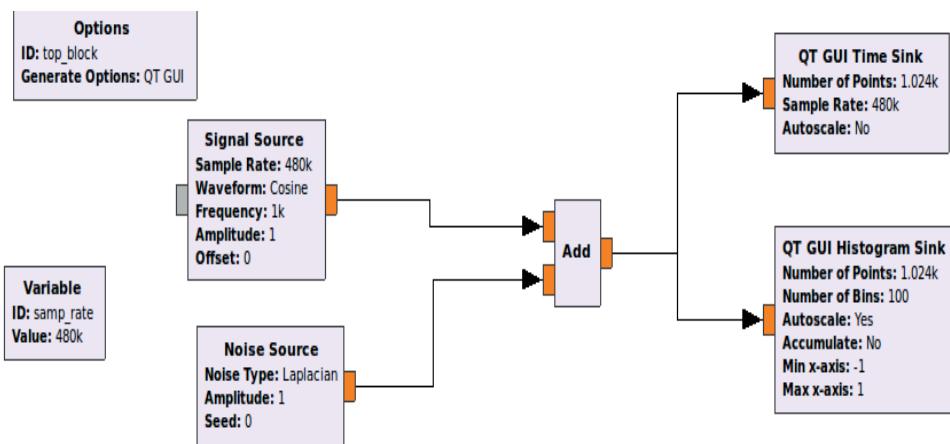


Figure 3.2.2 Flow graphs indicating Laplacian Noise.

Fig 3.2.2 represents a Noise source indicating Laplace distribution with amplitude defined as 1. A 1000 Hz cosine carrier frequency with 480 KHz sample rate is used as an input.

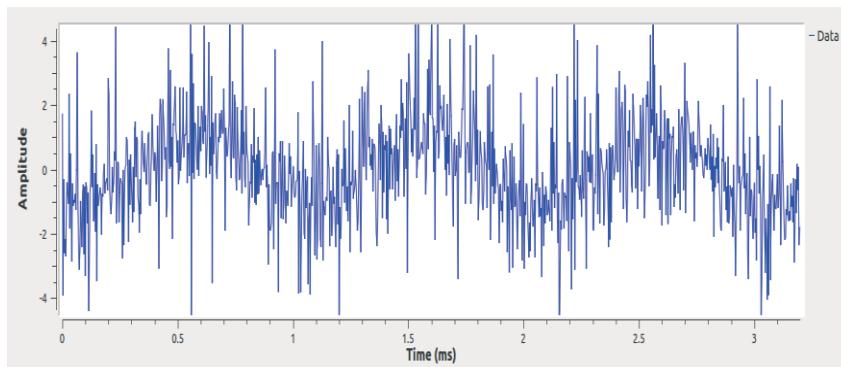


Figure 3.2.3 Time domain representation of Laplace Noise

Fig 3.2.3 represents the time domain representation of Laplacian Noise distribution added with a cosine carrier waves signals. Fig 3.2.4 represents a Normal distribution Histogram plot of Laplacian Noise distribution.

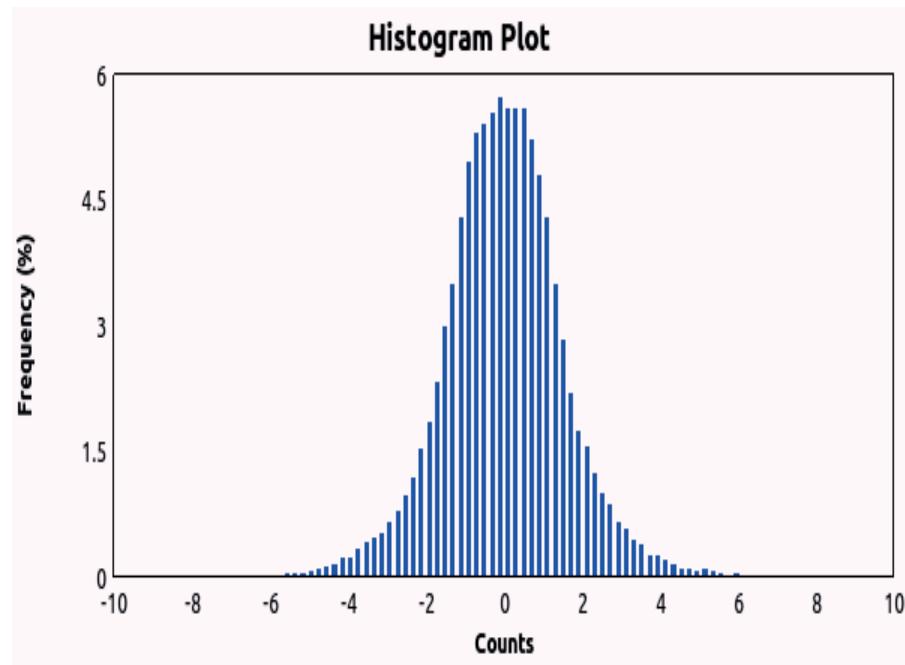


Figure 3.2.4 Histogram plot for Laplace Distribution.

Histogram helps in summarizing the distribution of univariate data set. It is used to plot the continuous set of data. It is said to be normal distribution because the set of data lying on the right sides falls equally with the data sides on left sides. The histogram plot represents a narrow width curve representing a normal distribution.

2. Gaussian Noise

Gaussian Noise is a type of Statistical Noise having probability density function equal to Normal distribution (I.e. Gaussian noise result in the continuous probability distribution). Statistical function is fraction of variance of dependent variable which cannot be predicted by the given independent variable. Gaussian Noise is also called as White Noise because if a white Noise is passed through the linear filter, it still results in the Gaussian Noise. The parameters like variance and means changes.

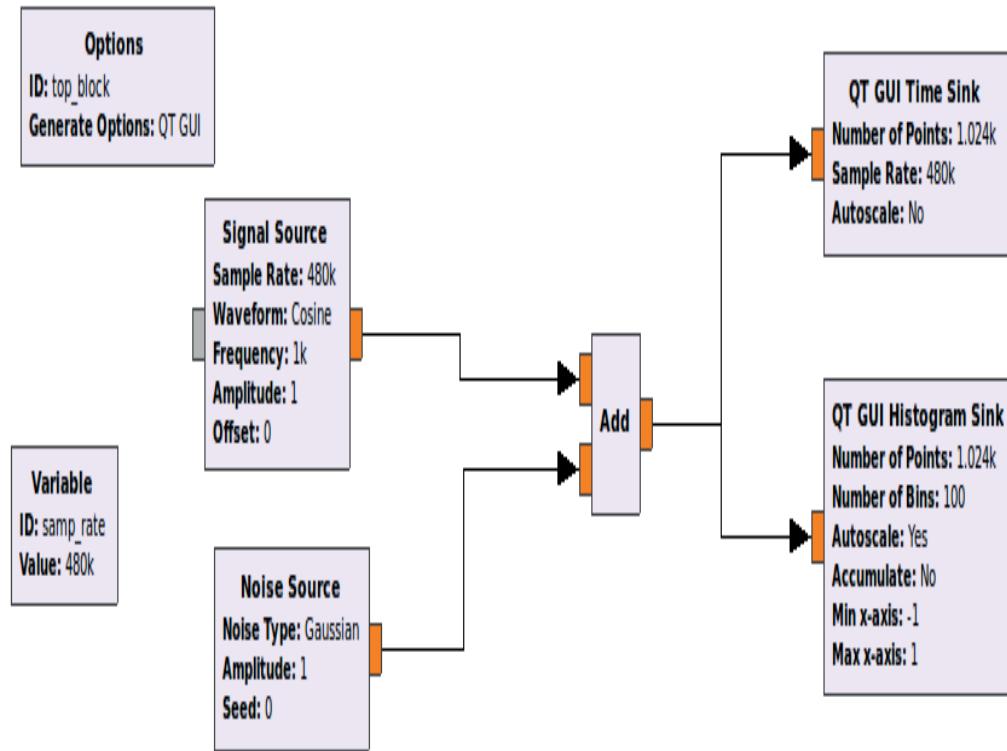


Figure 3.2.5 Flow graph indicating Gaussian Noise.

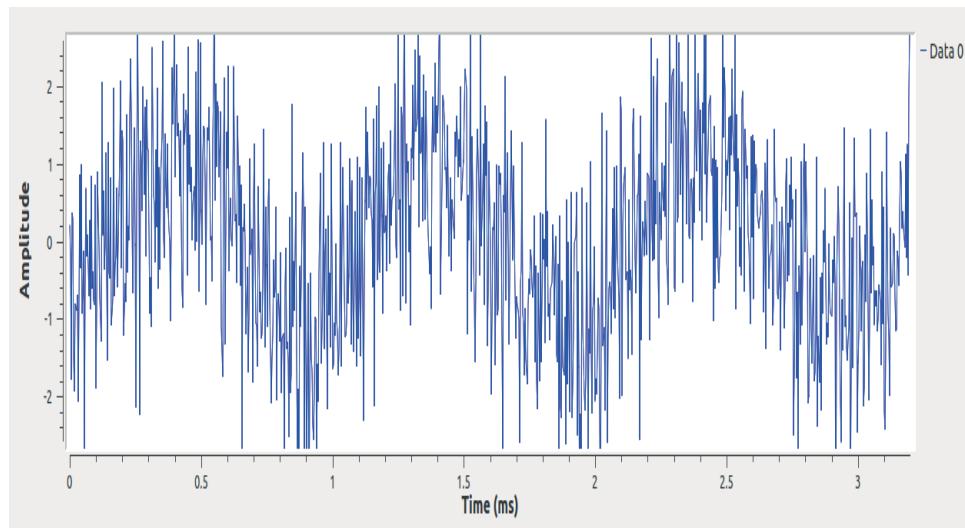


Figure 3.2.6 Time domain representation of Gaussian Noise with cosine wave signal.

Fig 3.2.7 represents the Histogram representation of Gaussian Noise where it provides an estimation of probability distribution for the set of continuous variable.

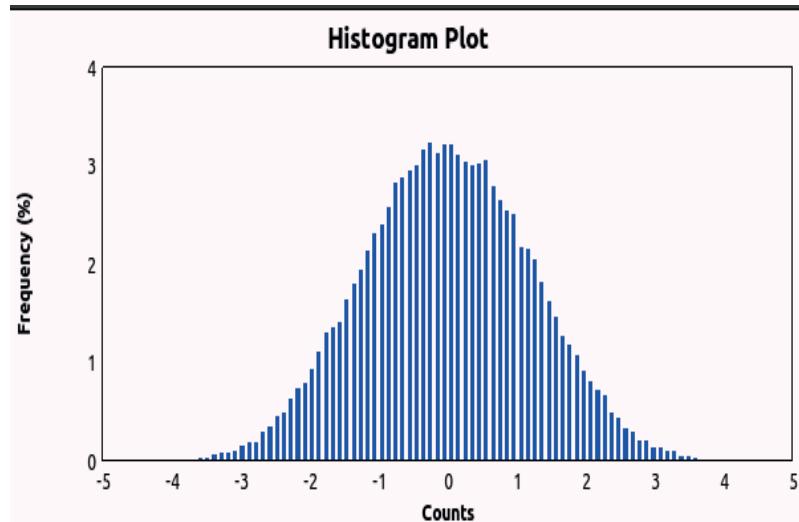


Figure 3.2.7 Histogram plot (Gaussian distribution).

Gaussian Noise is mostly used with respect to the real time signal like signal processing, image processing. Hence in such case the histogram plot for Gaussian Noise represents a wide width smooth curve representing a normal distribution.

3. Impulse Noise

Impulse Noise is a type of acoustic noise generated because of unwanted instantaneous sharp sounds for example noise generated from scratching any object, utensils falling, sounds coming while tapping on the wall etc.

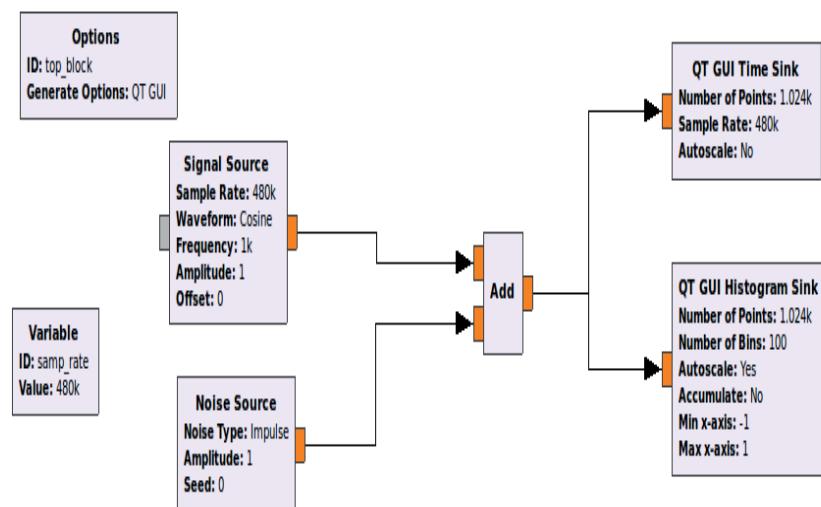


Figure 3.2.8 Flow graph indicating Impulse Noise.

Fig 3.2.8 represents the flow graph of cosine wave signals with 1000 Hz mixed with an impulse noise signal generated with amplitude as 1. From the result obtained in Fig 3.2.9, it can conclude that impulse noise like a flicker noise which occurs due to the instantaneous generation of unwanted signals generated. Hence it is not much effecting the desired output signals but difficult to remove it. A special type of filter is used which is called as a median filter in order to remove the flicker generated in the output signals.

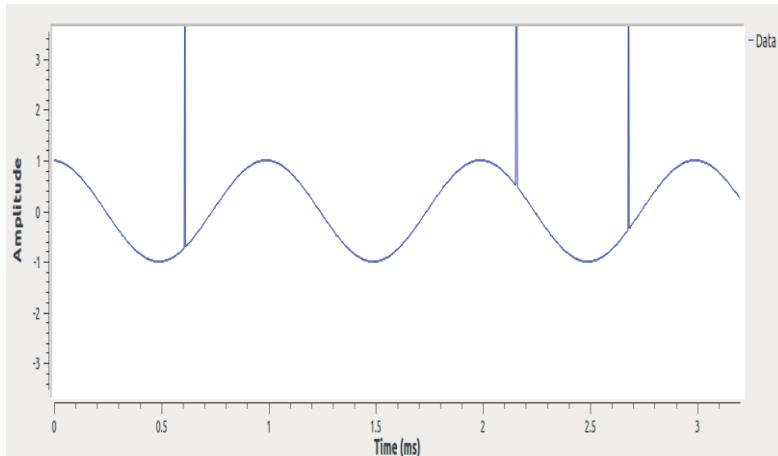


Figure 3.2.9 Time domain representation of Impulse Noise.

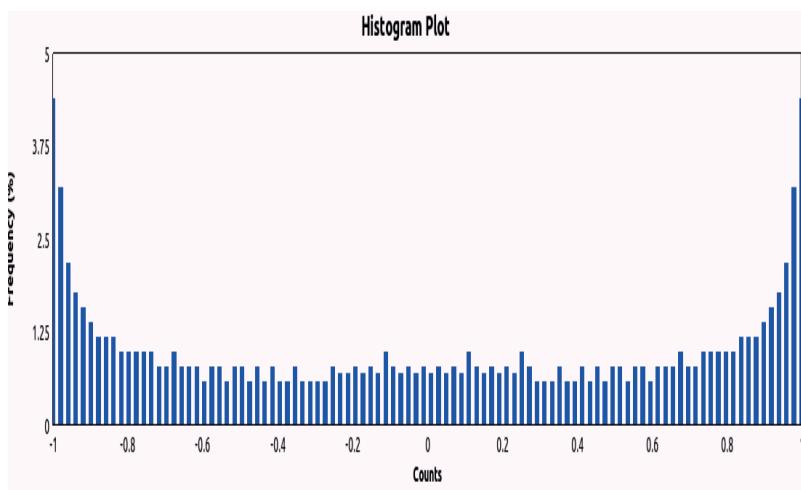


Figure 3.2.10 Histogram plot (Impulse distribution).

Fig 3.2.10 represents a histogram plot of Impulse Noise indicating a wide distribution of variables on the X axis which are not linear. The histogram plot represents a high spike generated at the end with some nonlinear distribution at the center.

4. Uniform Noise

Uniform Noise is a type of noise having constant amplitude which is defined as 1. Fig 3.2.11 defines the flow graph of cosine wave signal with uniform noise signals with it.

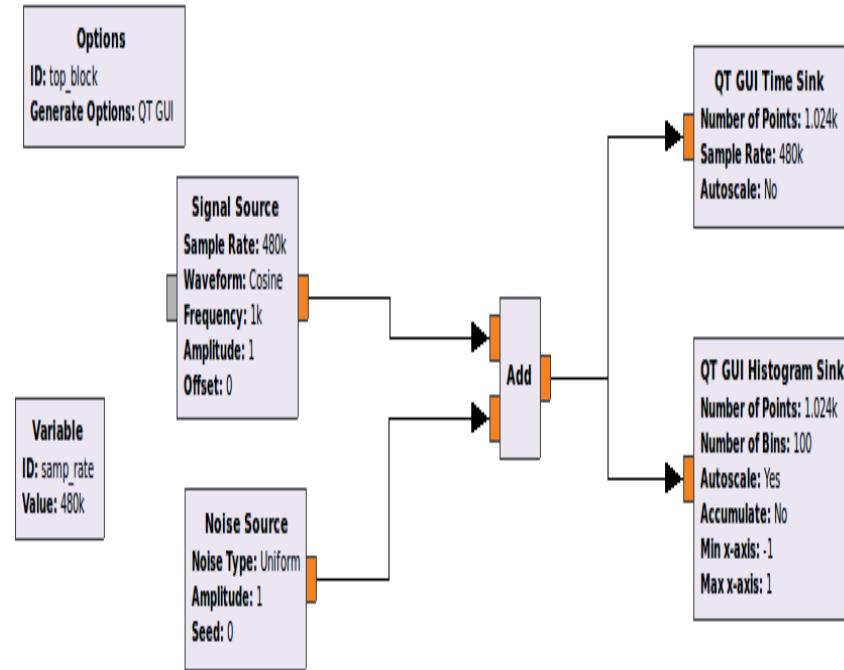


Figure 3.2.11 Flow graph indicating Uniform Noise.

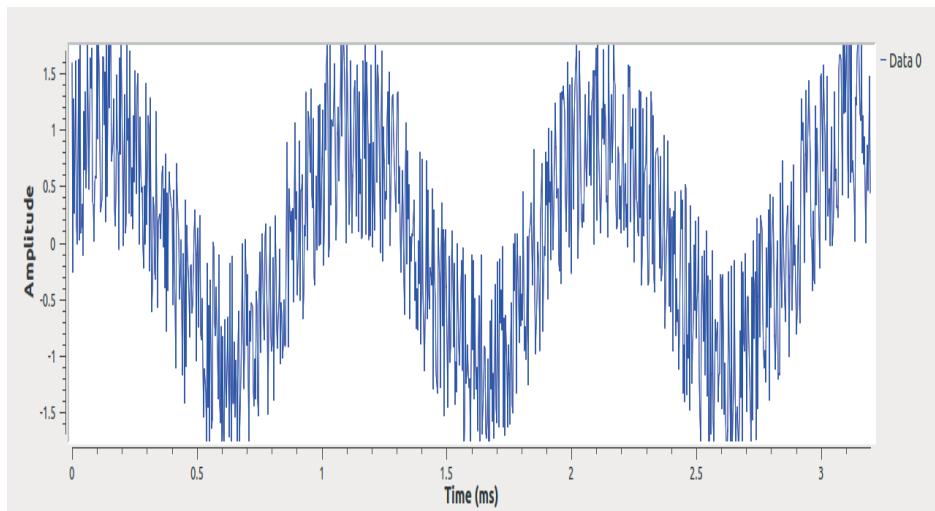


Figure 3.2.12 Time domain representation of Uniform Noise.

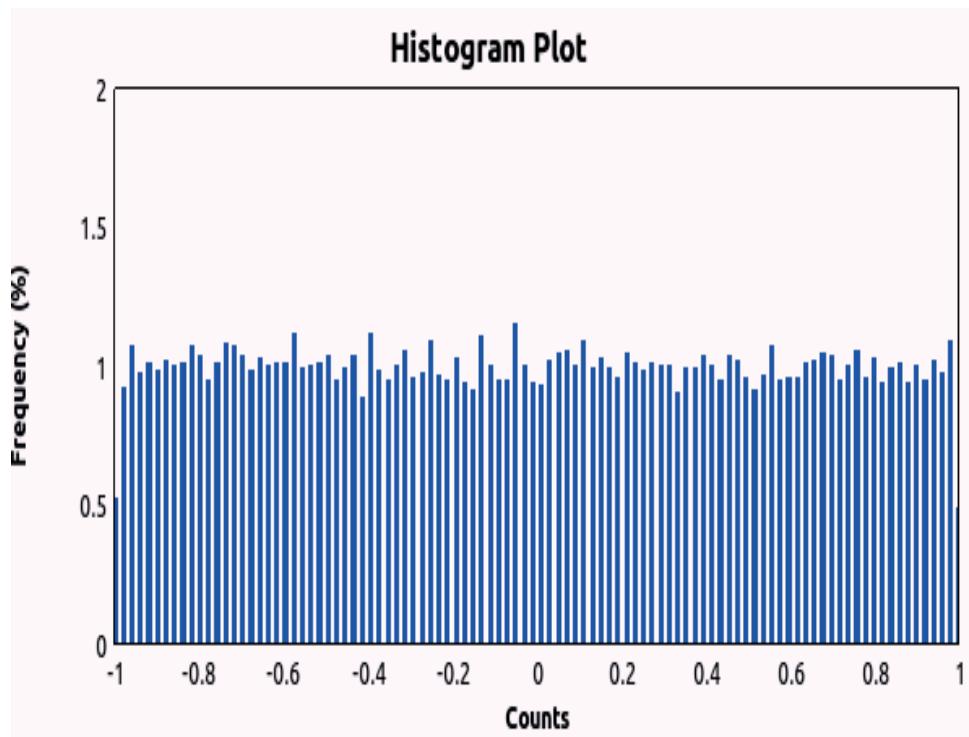


Figure 3.2.13 Histogram plot (Uniform distribution).

Fig 3.2.12 represents a time domain representation of cosine wave signals with uniform in noise amplitude signals. Fig 3.2.13 represents a histogram plot with a very wide width of linear variables having uniform amplitude.

Inference

Hence these are the different types of signal processing Noise signals which can be generated in Gnu radio with the help of small experiments. The results are analyzed with respect to the histogram which can help in estimating the probability distribution for the set of continuous variable.

EXPERIMENT 3

Aim

The main purpose of this experiment is to provide a complete description about the Quantization process and implement it using cosine wave signal in Gnu Radio.

Introduction

Consider a continuous forms of analog audio signals .Before transmitting, this input wave signals need to be sample down to discrete time signals. Further to digitalize these discrete values, the amplitude of the signals needs to be approximated between 0 and 1. Hence Quantization process plays an important role. Quantization is a process of round off or approximation of a continuous analog signal. It is the process of converting a continuous range of values into a finite range of discrete values. The quantizing of an analog signal is done by discretizing the signal with a number of quantization levels or it is also called as sub division levels. For examples in the figure mentioned below there are 2 bit quantized (Consider positive half or either negative half of cosine wave signals) levels. The quality of a Quantizer output depends upon the number of quantization levels. Higher the Quantizer levels, better is the Quantization process.

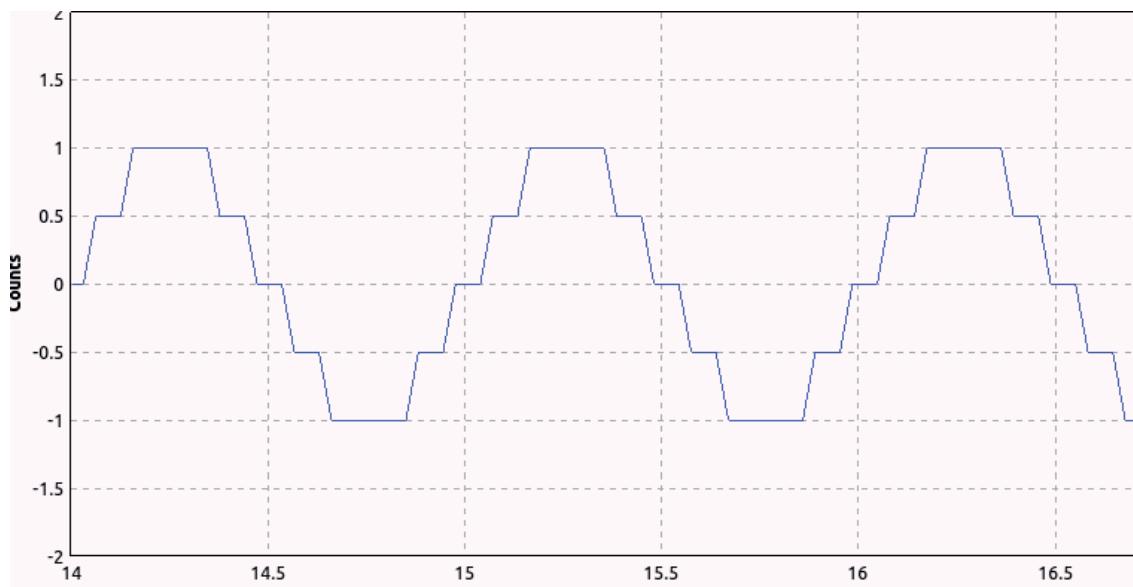


Figure 3.3.1 Quantization process.

Quantization process results in the loss of information. Hence on the demodulation side of wireless communication system there is always an error generated which cannot be reconstructed back to original signals. Step size is defined as the minimum possible differences measured in terms of amplitude in order to generate Quantization levels.

Let Step size be defined as S , having Quantization levels $n = 2$

Hence step size of the quantized output is mathematically expressed as

$$S = \frac{VH - VL}{n}$$

Where input cosine signal has a peak which swings from VL TO VH along the Y axis.

EXAMPLE

Considering an example from the mention in fig 3.3.1,

Given Data:

We have $VH = 1$ and $VL = 0$,

Quantization levels $n = 2$

To Find: Step size =?

Solution:

$$\text{Step size } S = (1-0)/2 = 0.5$$

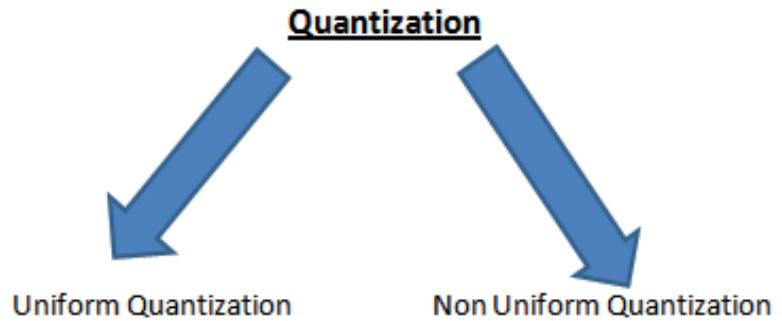
Hence for getting 2 Quantization level for amplitude which swings from 0 to 1, we need a Step size of 0.5

There are two types of Quantization process which are

1. Uniform Quantization process
2. Non uniform Quantization process.

Process in which the quantization levels are uniformly placed is called as the uniform Quantization process, whereas the quantization levels which are not uniformly placed as

termed as Non uniform Quantization. Quantization has a number of applications in digital image and audio production.



Block explanation of the flow graph

Options
ID: top_block
Generate Options: WX GUI

1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as WX GUI.

Variable
ID: samp_rate
Value: 32k

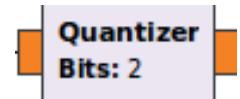
2. Variable

Variable block is used in order to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 32K for proper tuning of sampled signals. (Tuning is done on try and error based, which changes based on the frequency of input signals).

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0

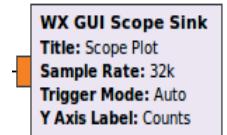
3. Signal source

Signal source is used to generate a cosine wave signal with carrier frequency of 1k which can be varied as per your requirement. Amplitude of cosine wave never exceeds 1.



4. Quantizer

Quantizer is a block which generate a quantized output with 2 bit quantization levels having a step size of 0.5 as per explained above.



5. WX GUI Scope Sink

GUI stands for Graphic user interface with standard WX. Scope sink is used to represent the time plot for the desired output signals. Trigger mode is kept auto for default automatic which will be operated by the software.

Gnu Radio Flow graphs

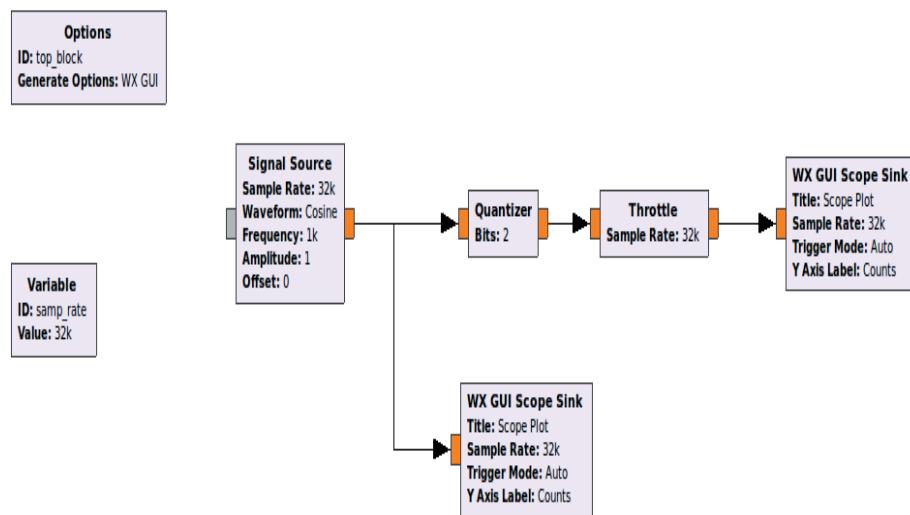


Figure 3.3.2 Quantization of cosine wave signal Flow graph.

Results

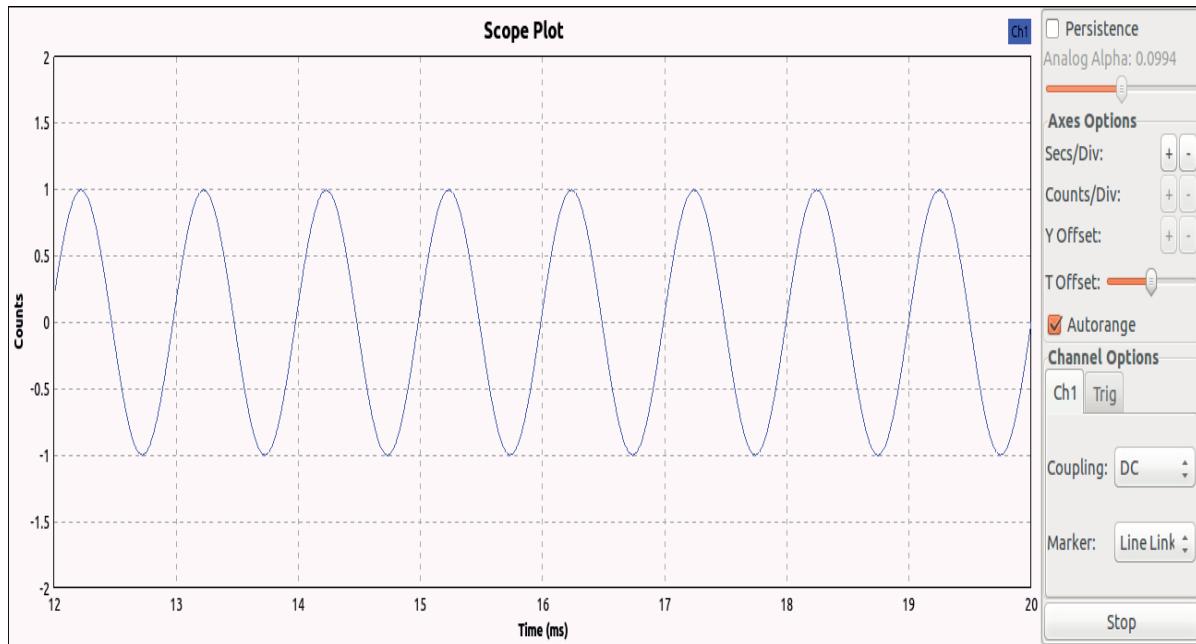


Figure 3.3.3 Input cosine wave signal with carrier frequency of 1 KHz.

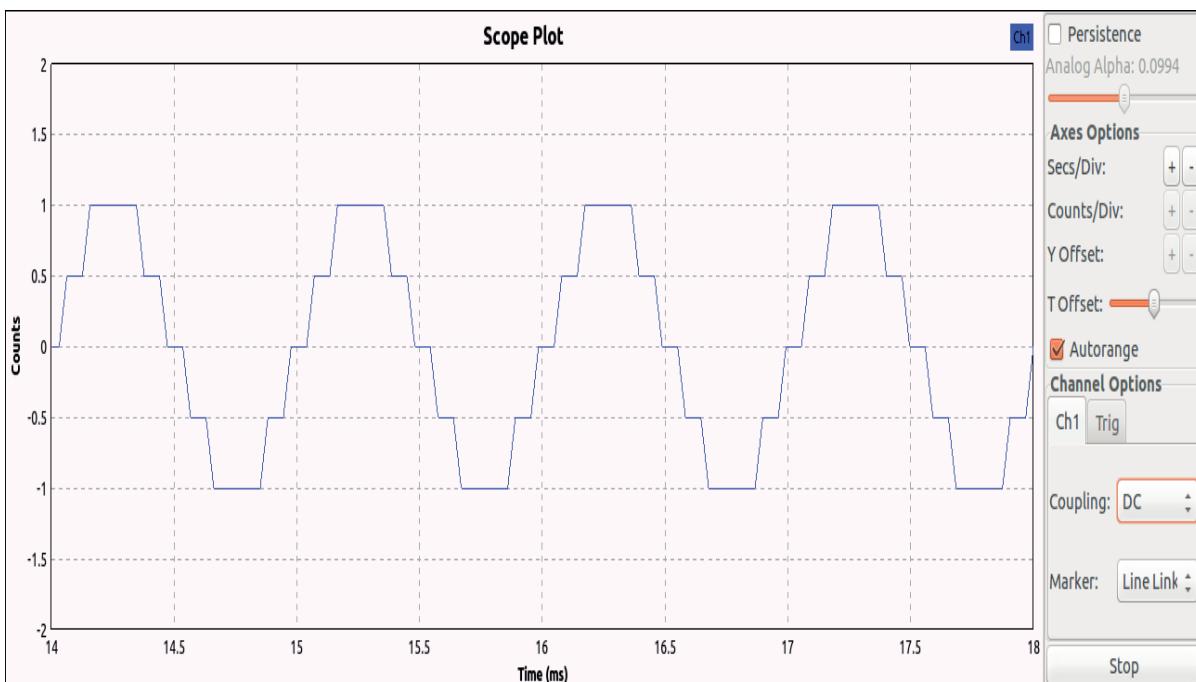


Figure 3.3.4 Quantized output signal.

Inference

This experiment demonstrates the implementation of Quantization process of cosine wave signals through Gnu Radio platform. It is found that with increase in the quantization level, the better will be the digitalization of the discrete sample values. The step size is calculated based on the difference in the highest and lowest amplitude count divided by the quantization levels. Hence it can be noticed that for large quantization level, the step size has to be small as possible (Quantization level and Step size are inversely proportional to each other).

EXPERIMENT 4

Aim

The main purpose of this experiment is to understand the importance of Nyquist theorem in digital wireless communication system. This experiment aims in implementing the concepts of Nyquist theorem using Gnu Radio.

Introduction

Nyquist theorem was developed by Harry Nyquist in the year 1928 and states that “ an analog signal waveform is converted to digital format and can be reconstructed back to original analog signal if and only if the sampling rate is greater than or equal to, twice the highest frequency component in the analog signal ”.

Nyquist theorem is mathematically expressed as

$$f_s \geq 2f_{\max}$$

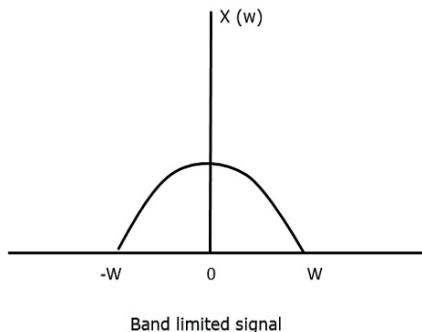
Where f_s = sampling frequency or sample rate.

f_{\max} = Maximum frequency.

The sample rate determines the number of samples taken per second. The sample rate is inversely proportional to sample period which is mathematically expressed as

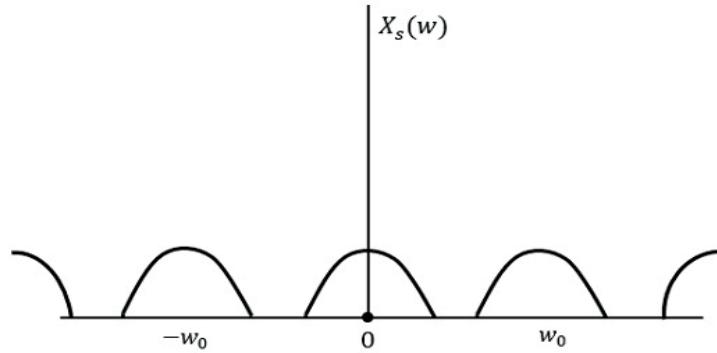
$$\text{Sampling frequency } = f_s = \frac{1}{t_s}$$

Consider a bandlimited Signal with $x(w)$ as an input signal and signal is bandlimited to $[-w, w]$.



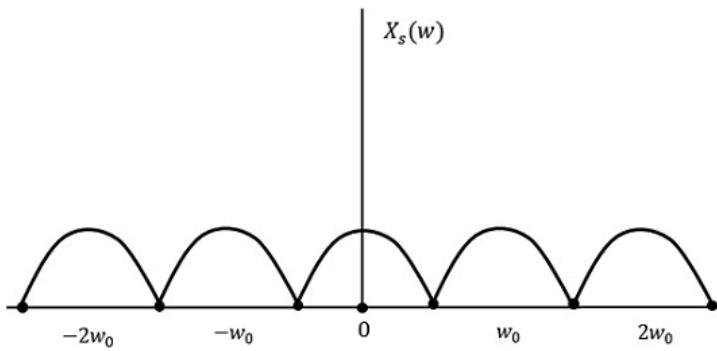
CASE 1:

If the sampling frequency greater than two times the maximum frequency then the signals can be reconstructed back to original signals without any loss is represented as shown in figure below:



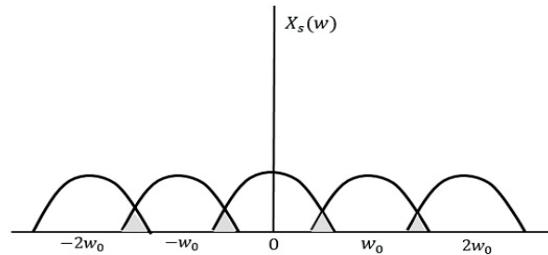
CASE 2:

If the sampling frequency is sampled with equal to 2 times the maximum frequency then still signals can be recovered back to original. It is considered to have a perfect sampling condition.



CASE 3:

If the sampling frequency is sampled less than the 2 times the maximum frequency then the Nyquist criteria fail. Hence original signals cannot be recovered back in such condition. It is consider having poor sampling criteria. Such conditions may result in loss of information and result to Aliasing error (for Aliasing error refer to next experiment).



Hence a Nyquist criterion is an important phenomenon in the transmission and reception of real time signals. For example if there are two frequency channels which are sampled and if one of the channel frequencies is not satisfying the Nyquist criteria, then the channel may get sampled to some unknown frequency channel or there might be overlapping of two frequency channels which will result in the loss of information.

Advantage of Nyquist theorem

1. Helps in maintaining the stability of system.
2. Provides information about the relative stability of system about its stability and instability.
3. Avoid Aliasing error.

Gnu Radio Flow graphs

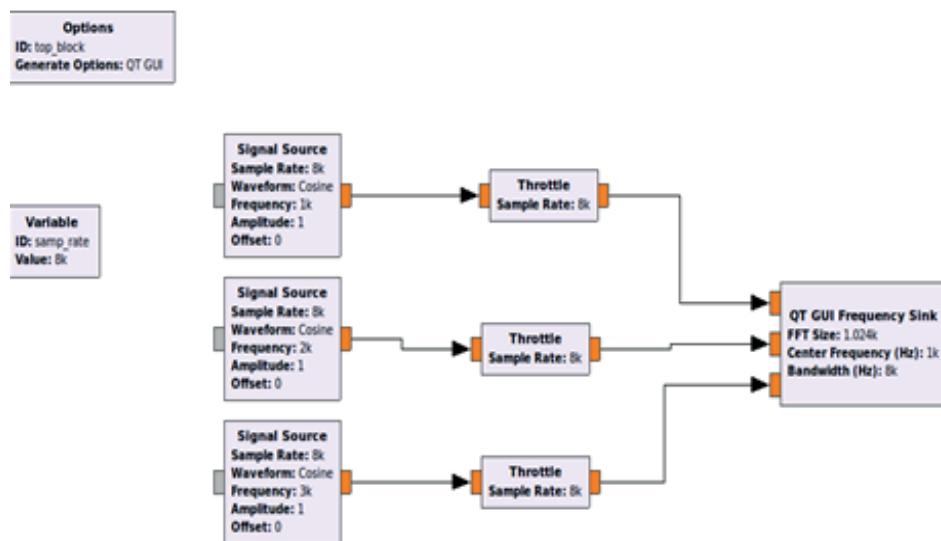


Figure 3.4.1 Nyquist Theorem flow graph

Fig 3.4.1 represents Nyquist theorem flow graph where 3 signal source is considered in order to create a carrier signal with frequency $f_1 = 1$ KHz, $f_2 = 2$ KHz, $f_3 = 3$ KHz. The sample rate is 8000 Hz. Sample rate is chosen in such a way that it is greater than twice of the frequency mentioned above.

Results

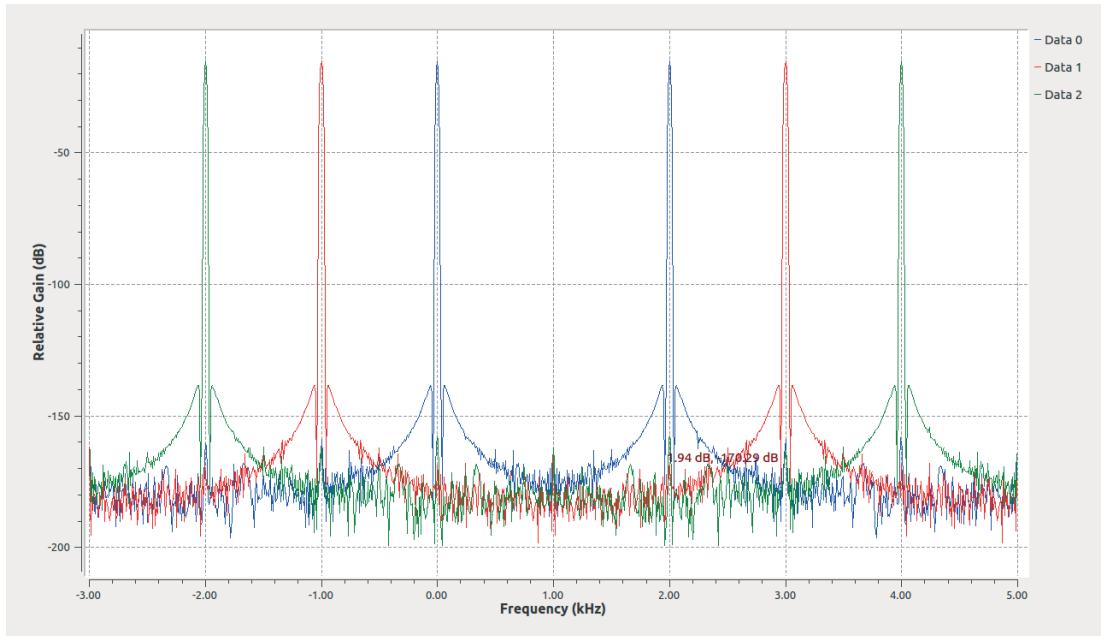


Figure 3.4.2 Frequency samples at 2 KHz, 3 KHz & 4 KHz.

Inference

This experiment aims in providing a complete description about the Nyquist criteria and thus implementing it on Gnu radio. Fig 3.4.2 represents three carrier frequency signals which are satisfying the Nyquist criteria ($f_s \geq 2f_{\max}$) and hence there is no overlapping of frequency carrier signals takes place. This results in perfect sampling theorem as explained in case 2 above. The next experiment will describe about the Aliasing error which occurs when Nyquist criteria is not satisfied.

EXPERIMENT 5

Aim

The main purpose of this experiment is to analyze the aliasing error in digital wireless communication system and implement it on the Gnu radio platform.

Introduction

Nyquist theorem states that “ an analog signal waveform is converted to digital format and can be reconstructed back to original analog signal if and only if the sampling rate is greater than or equal to, twice the highest frequency component in the analog signal ”.

Nyquist theorem is mathematically expressed as

$$f_s \geq 2f_{\max}$$

Where f_s = sampling frequency or sample rate.

f_{\max} = Maximum frequency.

The sample rate determines the number of samples taken per second. The sample rate is inverse of sample period which is mathematically expressed as

$$\text{Sampling frequency} = f_s = \frac{1}{t_s}$$

If the Nyquist criteria are not satisfied then aliasing error is generated. Aliasing error is defined as an effect caused in different signals to become indistinguishable. It also refers to the distortion which result in when the signal reconstructed from samples is different from the original continuous signal. Hence a Nyquist criterion is an important parameter in the wireless communication system, as it may results in the loss of information.

Method to avoid Aliasing effect

1. Increase the sample rate beyond the Nyquist criteria.

2. Make use of low pass filter.
3. Zero padding techniques (Refer some reference book for detail).

Gnu Radio Flow graphs

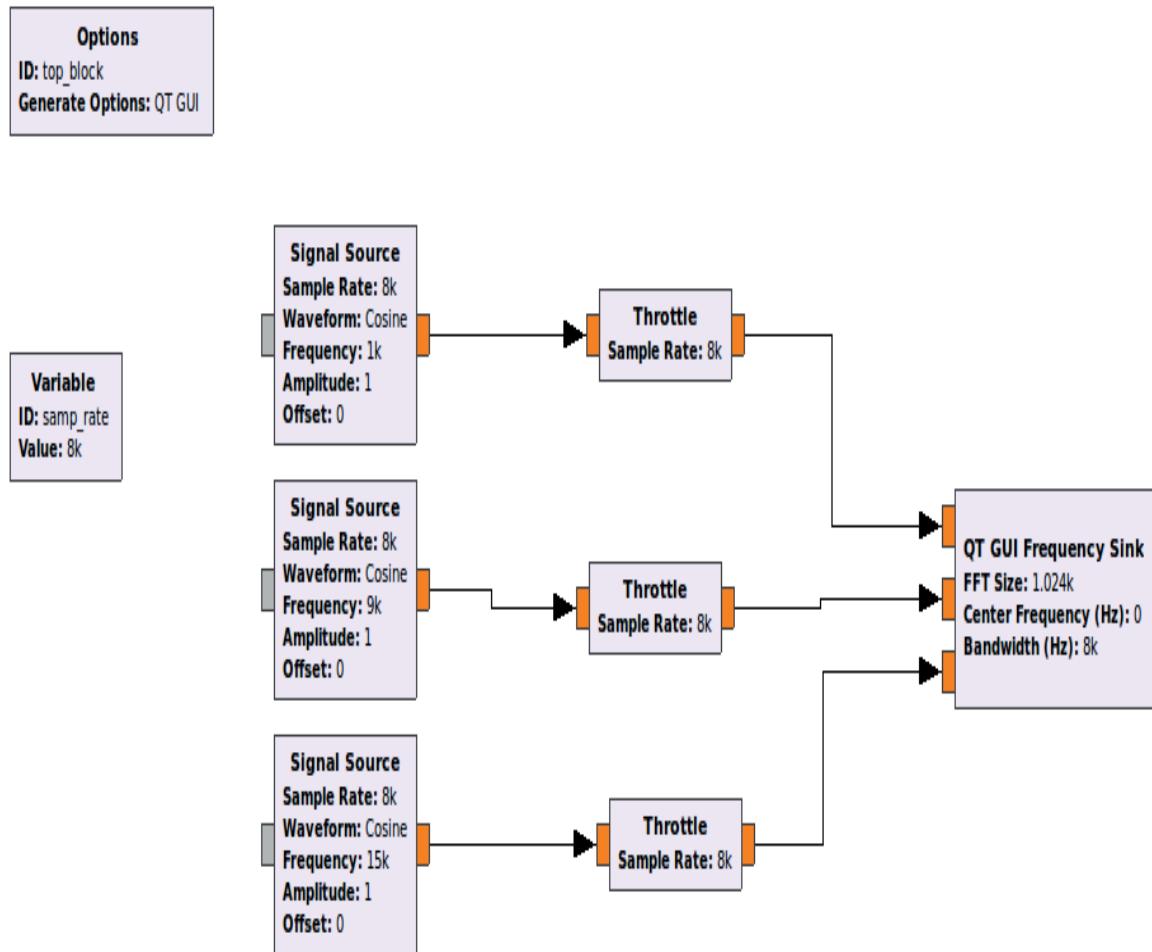


Figure 3.5.1 Aliasing error flow graph

Fig 3.5.1 represents a flow graph where 3 signal source are considered with the carrier signal having frequency $f_1 = 1 \text{ KHz}$, $f_2 = 9 \text{ KHz}$, $f_3 = 15 \text{ KHz}$. The sample rate is 8000 Hz.

1. In first case $fs \geq 2*f_1$, Hence it satisfies the Nyquist criteria.
2. In second case $fs=8 \text{ KHz} < 18 \text{ KHz}$, hence the Nyquist criteria fails.
3. In third case $fs= 8 \text{ KHz} < 30 \text{ KHz}$, hence the Nyquist criteria fails.

Results

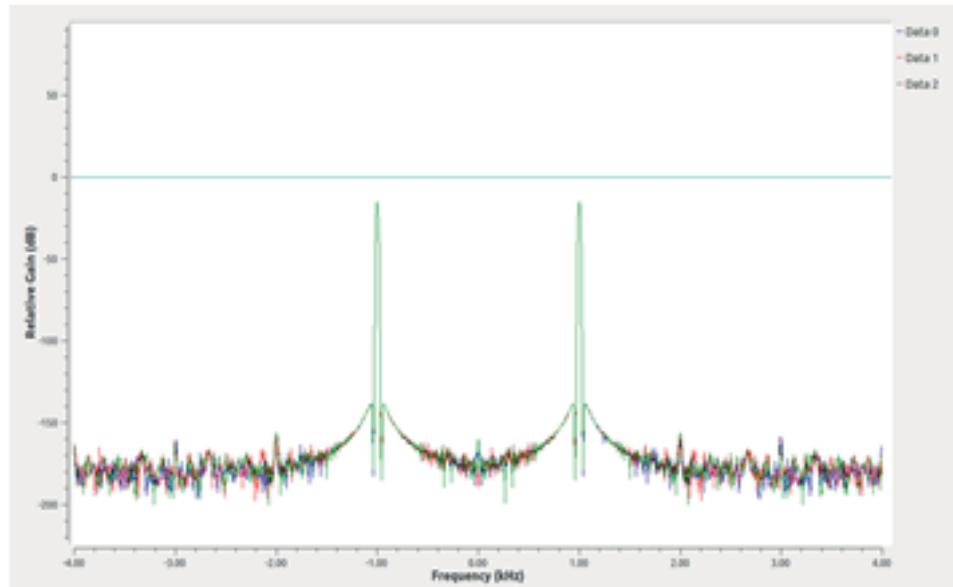


Figure 3.5.2 Aliasing error

Inference

This experiment aims in explaining the aliasing error caused when Nyquist criteria is not satisfied. Fig 3.5.2 represents three carrier frequency signals which are mixed because the signals lies beyond the Bandwidth defined. Hence overlapping of frequency carrier signals takes place and there is no proper distinguish between the signals.

EXPERIMENT 6

Aim

The main purpose of this experiment is to implement the frequency shift of a given signal using Gnu radio.

Introduction

The main concept behind the frequency shift of a signal is to add a constant value to the frequency signal component which results in the signal with different frequency signals. If the sampling frequency is defined as F_s , the repeating sequence of 1, -1

i.e. 1,-1, 1,-1, 1,-1, 1,-1 ... will act as a cosine wave signal with frequency $F_s/2$.

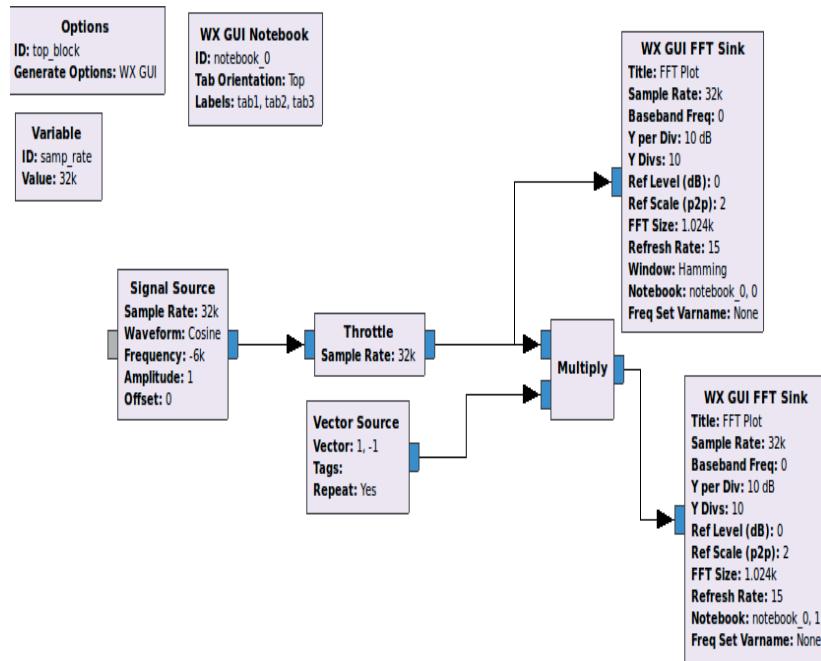


Figure 3.6.1 The Flow graph of frequency shift for a given cosine wave signals

In the figure above the cosine wave signals is generated with 6 KHz frequency signals in an imaginary direction. The negative symbol does not mean that the signals have a negative frequency signals but it indicates that the signal is in opposite polarity. The cosine signal is passed through the throttle block in order to avoid the CPU congestion.

The constant vector source with a vector of 1,-1 in a repeating mode is multiplied with the cosine wave signals. Considering a sampling frequency $F_s = 32$ KHz. Hence as per the concept defined above the cosine wave signal will be sampled at $F_s/2 = 32/2 = 16$ KHz

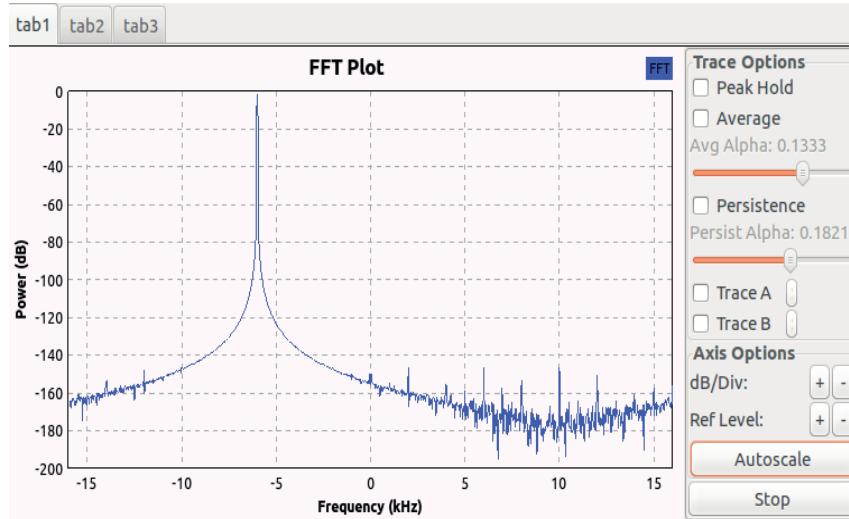


Figure 3.6.2 FFT of cosine wave signals at - 6 KHz

Fig 3.6.2 represent a FFT of a cosine wave signals at – 6 KHz frequency signals .Therefore if the sine wave is at - 6 KHz then frequency after mixing with vector source will be shifted to shifted Frequency = $-6+16 = 10$ KHz . Hence the frequency of signal is shifted to 10 KHz shown in the Fig 3.6.3.

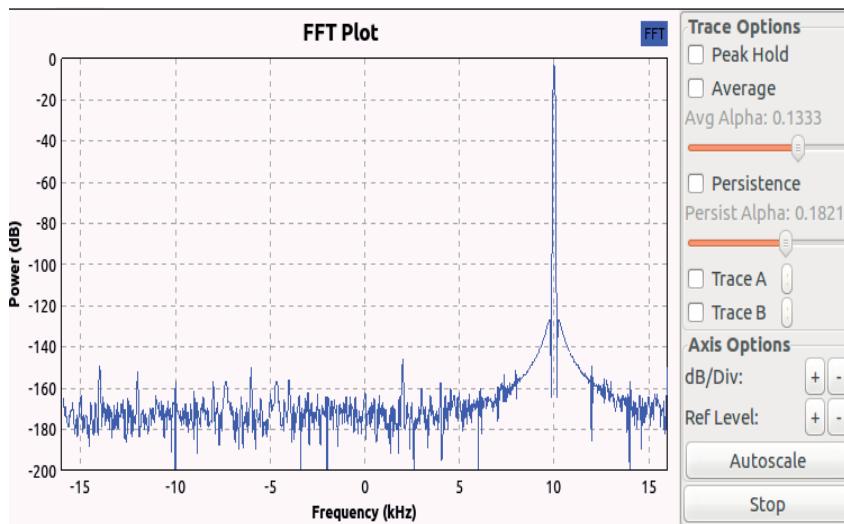


Figure 3.6.3 Frequency shifted to 10 KHz.

Inference

Hence the experiment successfully explains the frequency shift of a given cosine wave signals. It follows the concept of multiplying the repeating a constant vector source with a cosine wave then the cosine wave signals will be sampled at half the sampling frequency. Hence the shifted frequency signals will be the addition of the $F_s/2$ and cosine wave frequency signal.

EXPERIMENT 7

Aim

The main purpose of this experiment is to determine Signal to Noise Ratio (SNR) which is an important measure while evaluating digital wireless communication system.

Introduction

SNR or S/R is abbreviated as Signal to Noise ratio. It is defined as a ratio of input signal strength to background noise. The ratio is usually measured in decibels (dB). Consider the incoming signal strength as V_{in} , and the noise level, V_{noise} , then the signal-to-noise ratio, S/N.

SNR is mathematically defined as

$$SNR = 20 \log_{10} \left(\frac{V_{in}}{V_{noise}} \right) \text{ dB}$$

SNR is also defined as the ratio of mean to standard deviation of a signal which is mathematically expressed as

$$SNR = \frac{\mu}{\sigma}$$

Example

Case 1

$V_{in} = 0$ & $V_{noise} = 0$, Hence $SNR = 20 \log_{10} (0) = 0 \text{ dB}$

In this case the noise level competes with the input signal, hence there a reduction in data speed due to which Bit Error Rate will increase. There is loss in the packet transmission and reception in case of wireless communication system.

Case 2

$V_{in} = 10$ & $V_{noise} = 1$, Hence $SNR = 20 \log_{10} \left(\frac{10}{1} \right) = 20 \text{ dB}$.

In this case, the signal strength is high as compared to Noise signal; hence the SNR is high and results in low Bit Error Rate. The result in this case is easily readable as compared to previous section.

Case 3

In this case, the Signal strength is very weak as compared to Noise signal. Then SNR seems to be poor with high Bit Error Rate. Hence there is loss in the information containing signals. In this case reliable communication is not possible for the wireless communication system.

Gnu Radio Flow graphs

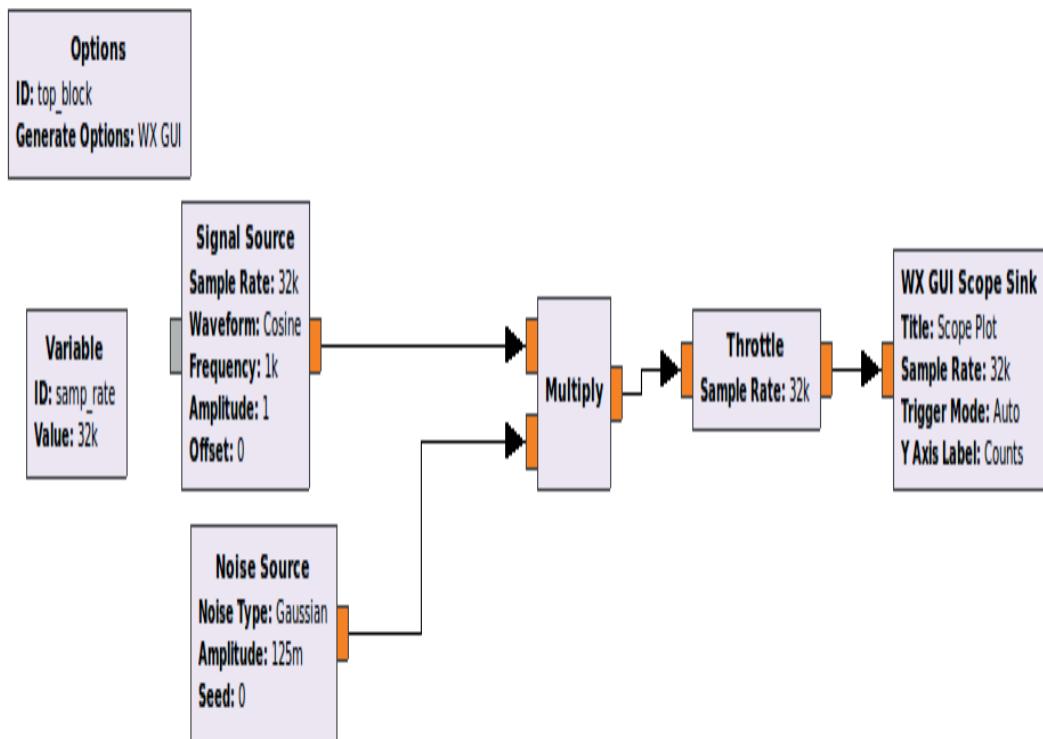


Figure 3.7.1. Flow graph

Fig 3.7.1 represents the flow graph showing the implementation of SNR with small experiment. A cosine carrier wave signals with 1 KHz frequency signals is used as an input. A Gaussian noise is used with change in Noise amplitude indicating change in SNR, which is well explained BER experiment (refer experiment implementing BER). The Noise amplitude is calculated as:

$$\text{SNR} = 10 \log_{10} \frac{1}{N}, \quad \text{therefore the Noise Amplitude} = \sqrt{\frac{N}{2}}$$

Hence the Noise amplitude $N=0.125$ represent 0dB SNR and Noise amplitude $N=0.707$ represent 15 dB SNR. The SNR is kept varying from 0 dB to 15 dB. Based on the variation in the SNR, the variation in the output signal is obtained which is explained in result section.

Results

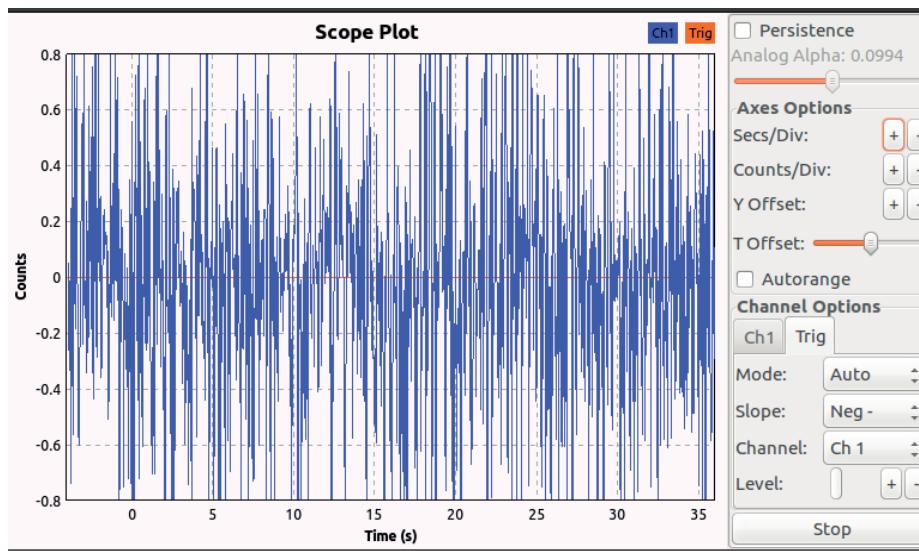


Figure 3.7.2 Time domain representation with 0dB SNR.

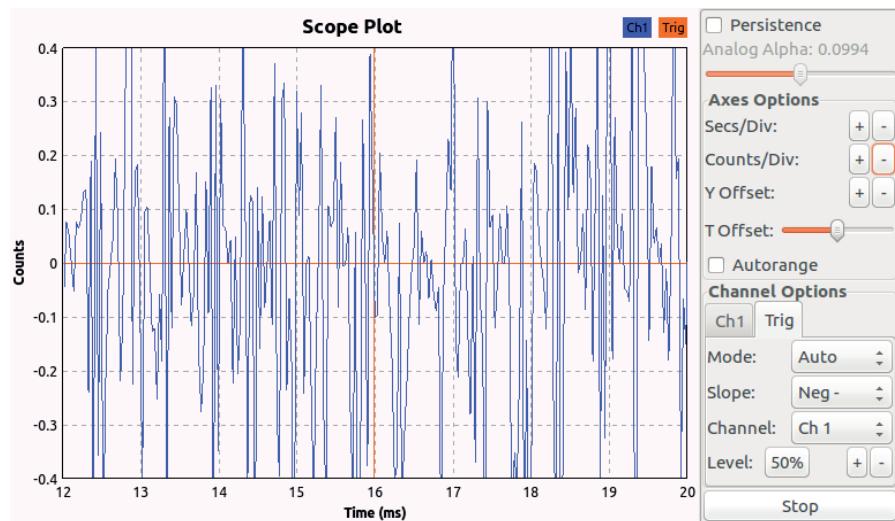


Figure 3.7.3 Time domain representation with 10dB SNR.

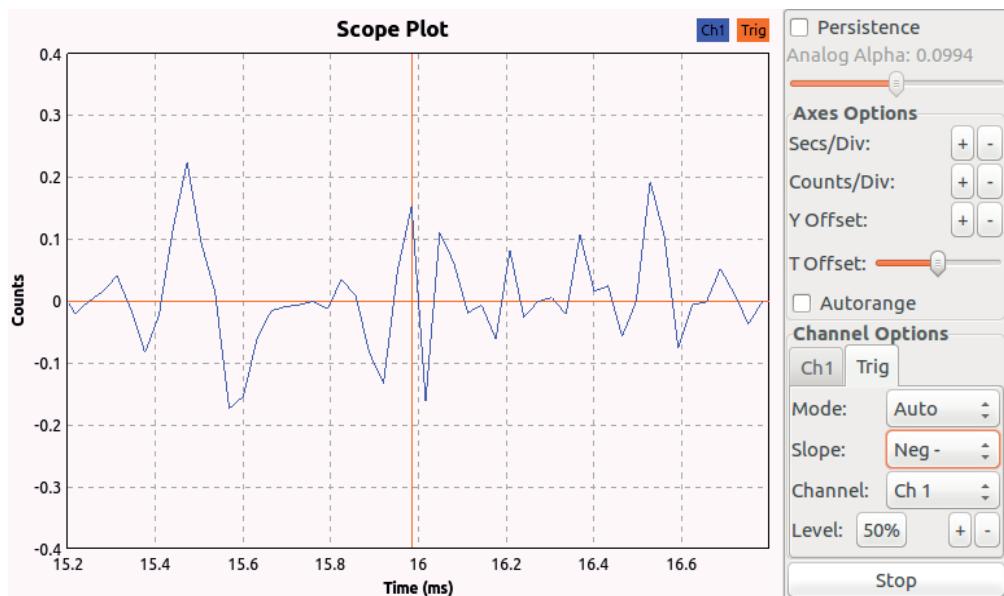


Figure 3.7.4 Time domain representation with 15dB SNR.

Inference

This experiment aims in providing a complete description about the Signal to Noise ratio and thus implementing it with Gaussian noise signals with Gnu radio companion. Hence from the fig 3.7.2, 3.7.3, 3.7.4 it was observed that with increase in the SNR, the output signals is readable and are clearer. Hence BER is reduced because the noise signal is weaker than the input signal strength.

EXPERIMENT 8

Aim

The main purpose of this experiment is to determine Bit Error Rate (Bit Error Rate) while evaluating digital wireless communication system.

Introduction

The bit error rate denoted as (BER) is defined as number of bits received after passing through the lossy channel medium which will not be equal to the number of transmitted bits. Hence it results in the error which is the basis on which BER is determined. The lossy channel medium includes noise, distortion, interference etc. It is one of the primary parameter in analyzing any wireless communication system.

The BER is mathematically expressed as the ratio of number of errors by the number of bits transmitted as shown below:

$$\text{BER} = \frac{\text{NUMBER OF ERRORS}}{\text{NUMBER OF BITS TRANSMITTED}}$$

Consider an example that the number of bits transmitted is 1000000 bits and the BER is found to be in the order of 10⁻⁶ i.e. BER is 0.000001. Hence it can be said that only one bit is found to have an error out of 1000000 bits. Therefore BER should be as low as possible in order to improve the efficiency of system with less loss in signals. Hence data rate is faster to achieve the overall transmission time for overall data transmitted.

Considering that

Transmitted bits TX= 1 0 1 1 1 1 0 0 0 1 and

Received bits are RX=1 1 1 1 0 1 0 0 0 1

Hence the number of bits affected is 2.

$$\text{Therefore } \text{BER} = \frac{2}{10} = 0.2 \text{ i.e. } 20\% \text{ error}$$

Hence 20% Bit error rate is obtained from the set of received bits when analyzed with respect to transmitted bits.

Steps to reduce the Bit Error Rate are as follow:

1. Reduce the Bandwidth, in order to improve the Signal to Noise ratio. Hence BER reduces.
2. Lower modulation scheme reduces the BER, but in the expense of throughput of signals.
3. Increase in the transmitted power decreases the BER.

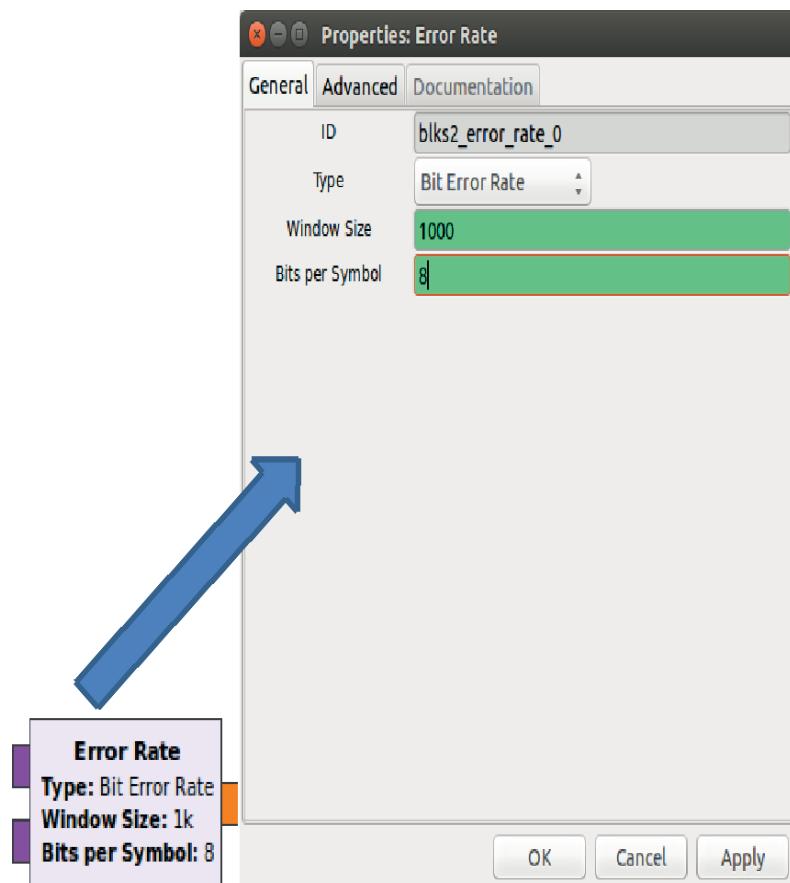


Figure 3.8.1 BER blocks description.

The Fig 3.8.1 represents a block explanation of BER. Bits per symbol are defined as the number of bits transmitted per symbol. Increase in the bits per symbol result in reduced in BER. Window size defines the total length required to displays the number of samples. The window size will start from 1 and will reach to the maximum window size. Once the window has reached the maximum, then the old samples are shifted out of the window frame thus allowing the new samples inside the windows.

Gnu Radio Flow graphs

Fig 3.8.2 represents the flow graph showing the usage of Error Rate plot with a small experiment. Error rate block can be used to analyze both BER and SER (Symbol error rate). A cosine carrier signal of 10 KHz frequency signals is used and is mixed with the Gaussian noise signals with 0db SNR. Error rate is used to analyze BER by comparing the input signals and the signals mixed with the noise. Window size is kept as 1000 Hz by default and assuming a better BER, hence bits per symbol are considered as 8. It can vary as per the experimental requirement.

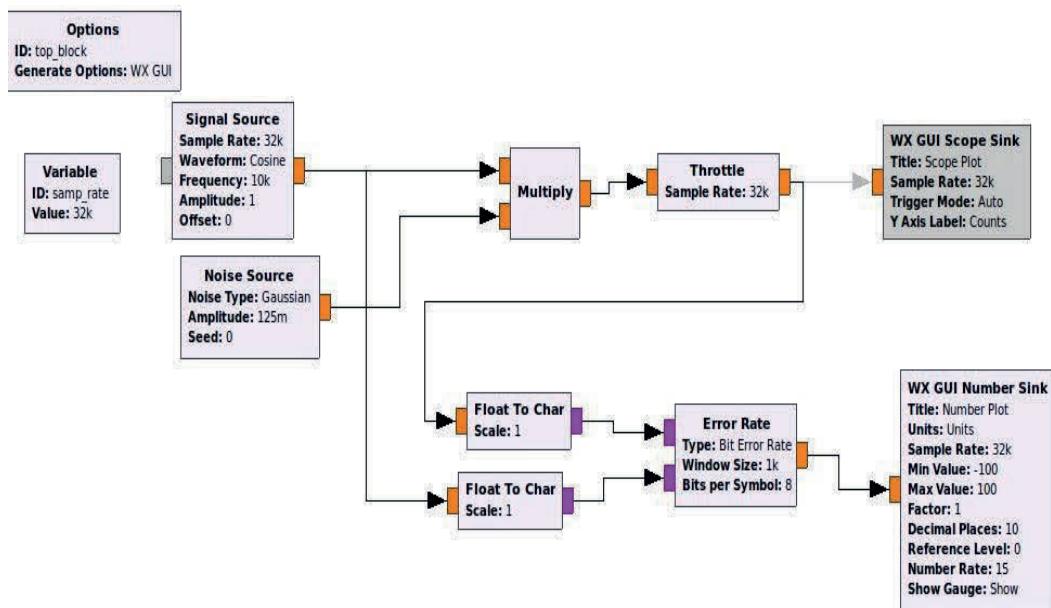


Figure 3.8.2 Flow graph.

Results

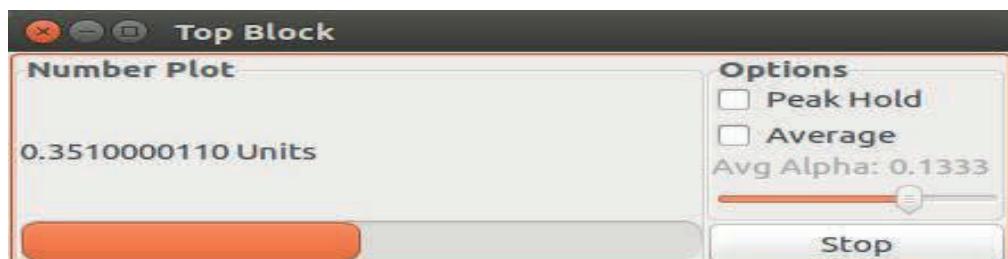


Figure 3.8.3 Number plot defining the BER as 0.3%.

Inference

This experiment demonstrated the Error Rate block and thus implementing the Bit error rate with a small experiment on Gnu radio companion. Hence from the experiment it was observed that the BER decreases with increase in bits per symbol. Thus number of bits transmitted will be correct and error bit will reduce.

EXPERIMENT 9

Aim

The main purpose of this experiment is to generate the Over sampling, under sampling and perfect sampling of cosine wave signals in Gnu Radio.

Introduction

Sampling is a process where the variation in the functions with respect to space, time, and frequency takes place. Hence sampling is defined as a technique which is used to reduce the continuous-time signal by considering the signals at discrete time; hence it results in the generation of discrete-time signal. For example a sound wave is converted to a sequence of samples for the further signal processing analysis. Quantization process which is explained in next chapter contributes in the reduction of samples. The sampling of signals can be obtained in three forms which are listed below:

1. Under sampling
2. Perfect sampling
3. Over sampling

Ideally the signals has to be in Perfect sampling, as it can band limit the signals within the defined frequency range, hence it can eliminate the problems like Aliasing error which results in no loss of signals. In case of perfect sampling, the Nyquist criteria is satisfied hence the signals can be reconstructed back to the original form. The under sampling results in the sampling of bandpass filtered signals at the sample rate less than the Nyquist rate. Therefore the under sampled signal cannot be reconstructed back. It is also called as the bandpass sampling. Over sampling is defined as the process of sampling a signal at a sampling frequency greater than the Nyquist rate. Hence the oversampled signals can be reconstructed back to original time domain signals.

Gnu Radio Flow graphs

The figure mentioned below represent two cosine wave signals with 10 kHz and 100 KHz frequency signals. The Add Const is used to change the samples with respect to under, perfect, over sampling. The value is set based on the condition of Nyquist criteria.

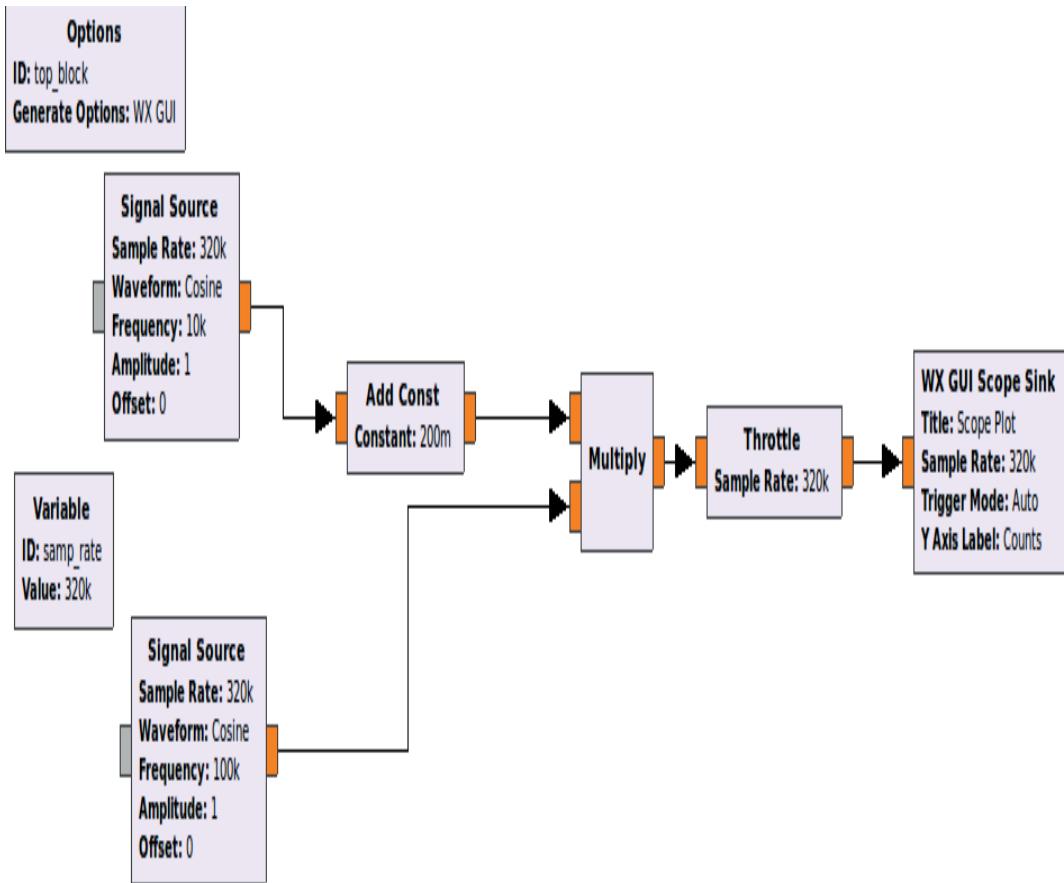


Figure 3.9.1. Sampling of cosine wave signal Flow graph.

Let us evaluate the three different scenarios as shown below:

1. If the Add Const is set as 0.2 then it results in under sampling, as the signal is sampled bellow the sample rate, hence the reconstruction of signal is not possible.
2. If the Add Const is set as 1 then it result in the perfect sampling, hence the reconstruction of signals is possible.
3. If the Add Const is set as 2 which is greater than the sample rate, then it result in the over sampling of signals. The reconstruction of signal is possible.

WX based GUI standard block is used. Scope Sink is used in order to represent the time domain representation of input signals. Throttle block is used to avoid the CPU congestion which takes place because of large sample value.

Results

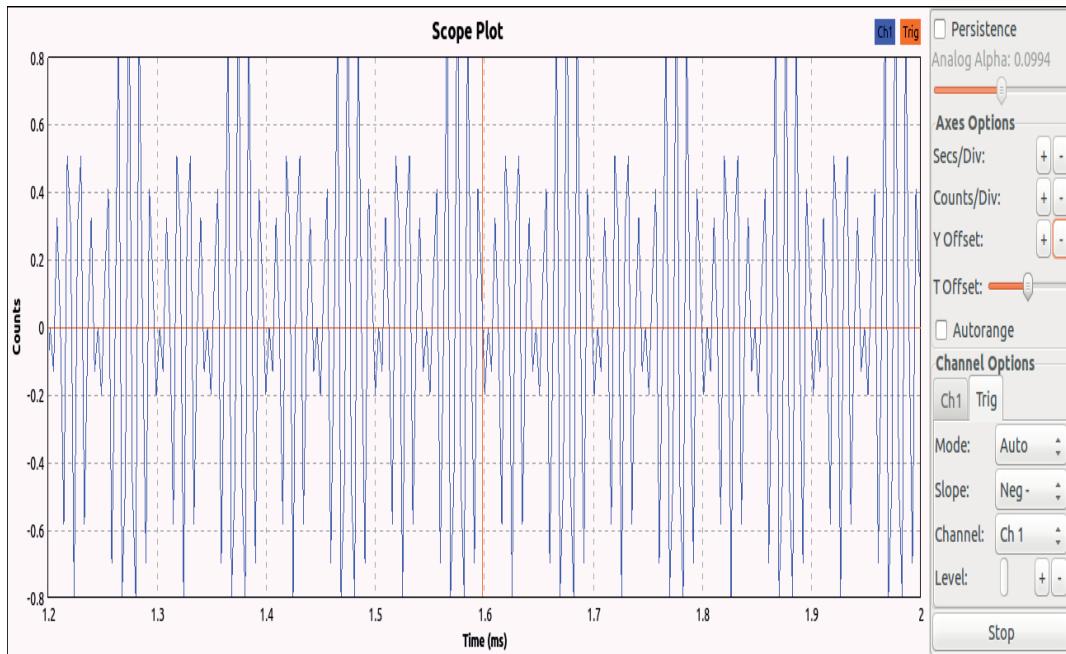


Figure 3.9.2 under sampling of cosine wave signals.

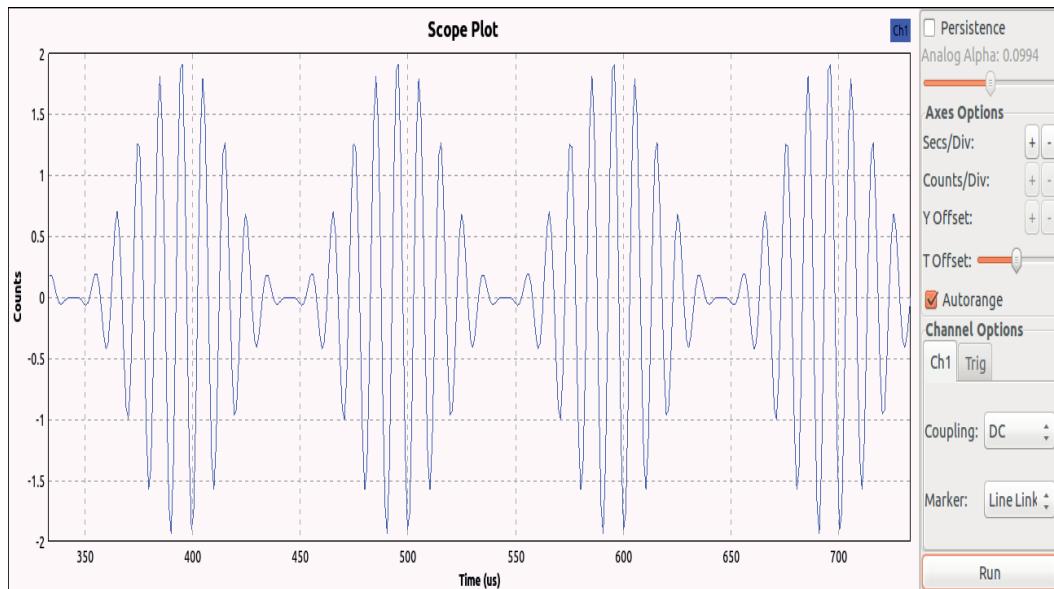


Figure 3.9.3 Perfect sampling of cosine wave signals

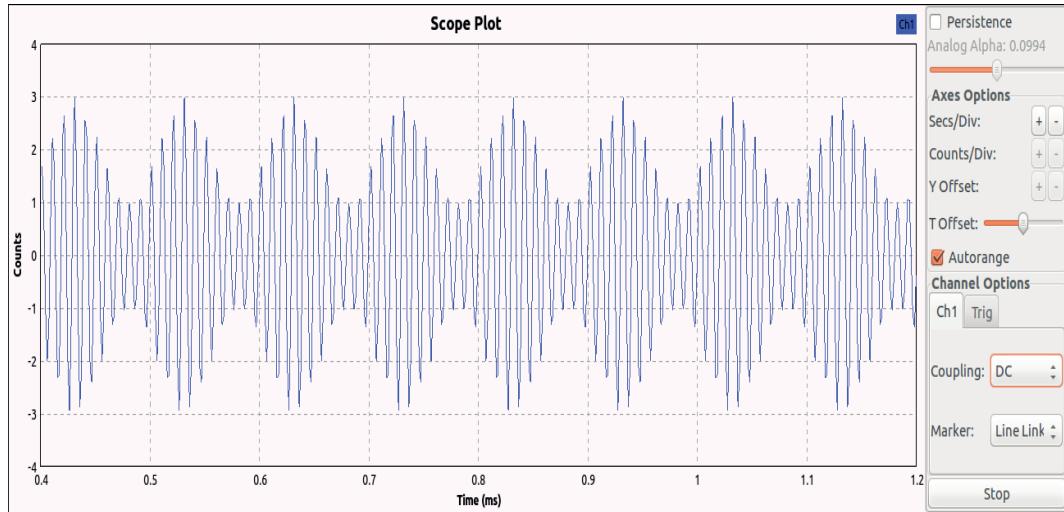


Figure 3.9.4 over sampling of cosine wave signals.

Inference

This experiment demonstrated the implementation of sampling process of cosine wave signals through Gnu Radio platform. Fig 3.9.2, 3.9.3, 3.9.4 represents the under sampling, perfect sampling, over sampling of a signals. It is found that the signals can be reconstructed to its original form only if the signals are sampled with perfect and over sample because these sampling process satisfies the Nyquist criteria (The signals are sampled equal to or greater than the sample rate) . Hence the Aliasing issues can be avoided. The under sampling fails to reconstruct back the original signals which result in the Aliasing error problem.

Chapter 4

Filters

This chapter mainly deals with the implementation of the passive and active type of filters used in the wireless communication systems. In the Radio frequency applications, the electronics systems have to deal with very high frequency signals. Hence some attenuation in the signals is needed to analyze the signals within the defined bandwidth range. A high level list of filter types and also their implementation using both GNU radio and USRP platforms have been captured.

1. Low pass filter,
2. High pass filter,
3. Band pass filter,
4. Band reject filter,
5. FIR filter and
6. IIR filter respectively.



Analog Filters

Digital Filters



Table 8: List of Filters experiment

Sr.No	Description
1	Experiment to explore the low pass filter using SDR.
2	Experiment to analyze the effects of low pass filter on the Audio signals using Gnu radio.
3	Experiment to explore the high pass filter using SDR.
4	Experiment to explore the effects of high pass filter on the audio signals using Gnu radio.
5	Experiment to explore the Band pass filter using SDR.
6	Experiment to explore the Band Reject filter using SDR.
7	Experiment to implement the finite impulse response filter (IIR) using SDR.
8	Experiment to implement the Infinite Impulse Response using (IIR) SDR
9	Experiment to implement the Root cosine filter and analyze its effects on the input signals using Gnu radio.
10	Experiment to analyze different windowing techniques with low pass filter using Gnu radio.

EXPERIMENT 1

Aim

The purpose of this experiment understand the functioning of a Low Pass Filter and also implement and understand the behavior using GNU Radio and USRP hardware platform.

Introduction

A Low pass filter is defined as a passive type of filter which is used in order to reject all the unwanted high frequencies signals generated from the system, thus allowing only low frequency signal through it. Cut off frequency which is denoted as f_c is an important parameter as it defines the boundary condition required in the frequency response of a system. Hence in case of Low pass filter, the energy beyond the boundary level will get attenuated or reduced. Therefore Low pass filter is also defined as a filter allowing only those frequency signals which are lower than the cut off frequency signals. In a simple way, Low pass filter can also be explained as a filter which removes the short term fluctuation and thus provides a smoothening of signals as an output. It is also called as a corner frequency or a break frequency.

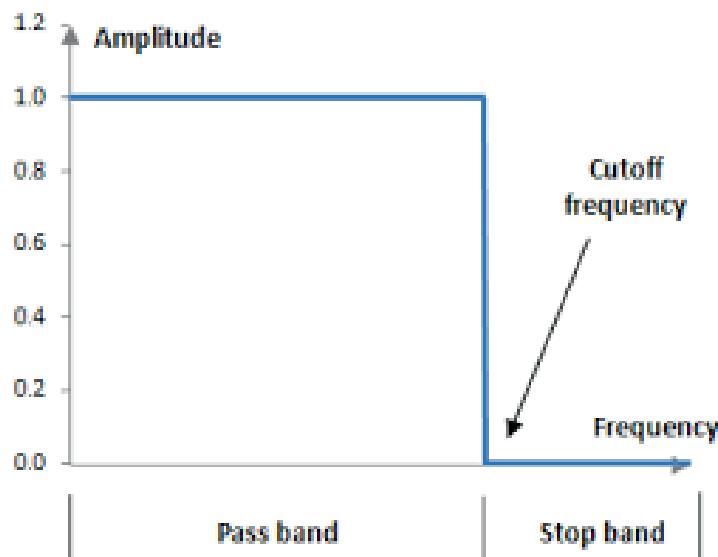


Figure 4.1.1 Frequency response of Low pass filter.

Low pass filter consist of pass band and stop band separated by the cut off frequency as shown in the above fig 4.1.1. Pass band is an area of frequency band in which the attenuation level (reduction in the amplitude of signals) is nearly equal to zero ($\alpha = 0$). Stop band is defined as an area of frequency band in which attenuation factor is extremely high.

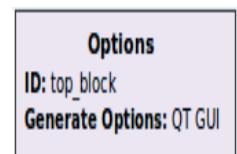
Advantages of Low pass filter

1. Provide smoothening of output signals.
2. Avoid Aliasing error.
3. Eliminate the unwanted high frequency signals and allow the low frequency information content signals through it.

Application of Low pass filter

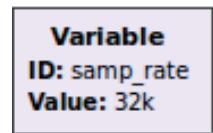
1. Used in Audio application for equalization process.
2. Used in super heterodyne receivers for the efficient reception of base band signals
3. Used as an anti-imaging and anti-aliasing filters in various digital signal processing application.

Block explanation of the flow graph



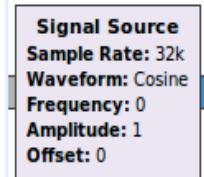
1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as QT GUI.



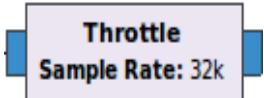
2. Variable

Variable block is used in order to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 32K for proper conversion of analog signals to discrete sampled signals.



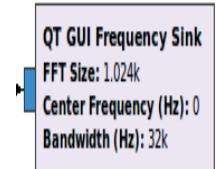
3. Signal source

Signal source is used to generate a cosine wave signal with carrier frequency of 0k, 5k and 10k respectively which is selected in order to have a proper discrimination between low frequency and high frequency signals. Amplitude of cosine wave never exceeds 1.



4. Throttle

Throttle is used to avoid the consumption of CPU resources from the flow graph. In this experiment throttle block is used with the sample rate defined earlier.



5. QT GUI frequency Sink

GUI stands for Graphic user interface with standard QT. Frequency sink is used to represent the frequency plot for the desired output signals with FFT size selected as 1.024K by default.



6. Low pass filter

Low pass filter block is used for the purpose mentioned above with cut off frequency 7 KHz and transition width of 10. Description about low pass filter is further explained below in Gnu radio flow graph section.

Gnu Radio Flow graphs

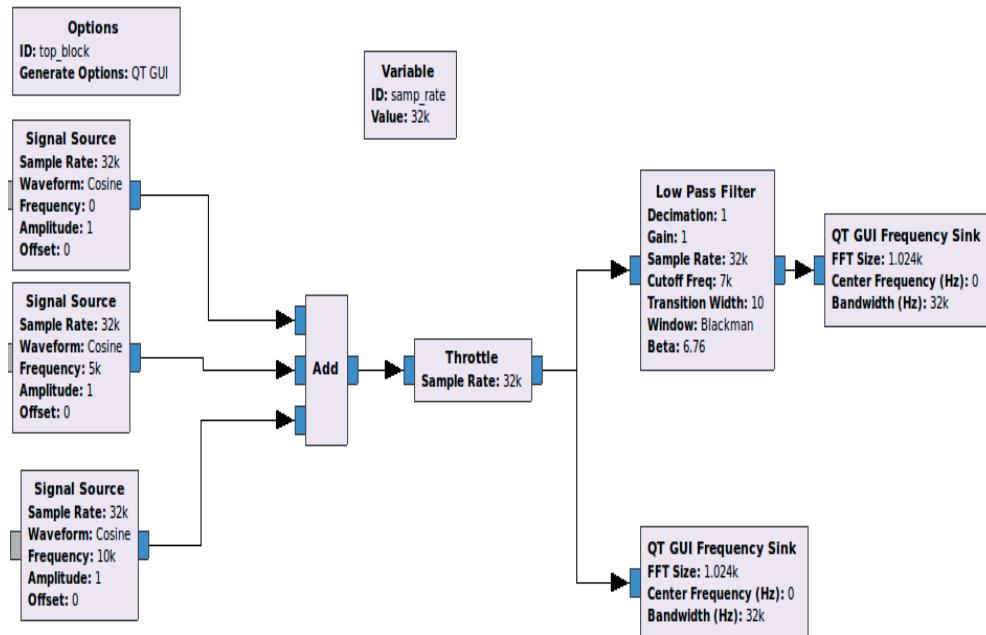


Figure 4.1.2 Low pass filter Flow graph.

The figure above provides information about the three carrier signals with the frequency of 0 KHz, 5 KHz, and 10 KHz respectively. These carrier signals are having amplitude of 1, as cosine waves always lies in the range from [-1, 1]. These carrier signals are added and passed through the throttle in order to avoid the congestion happening in system memory (CPU). The sample rate of 32 K is considered (It can change as per the experimental requirement). The output of throttle signal is then

allowed to pass through the Low pass filter block having cut off frequency 7 KHz and a transition width of 10. Transition width is basically used to Control the steepness in the attenuation of the signal above the cut off frequency. The Blackman window is used as a windowing technique. The Blackman window contains an extra cosine term; hence it provides a minimum leakage output with reduced in side lobes as compared to other windowing techniques.

Results

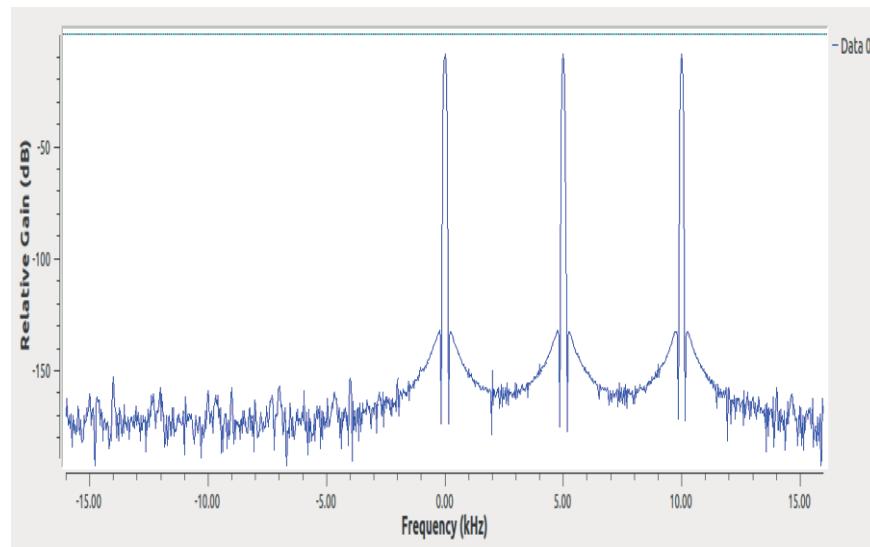


Figure 4.1.3 Input cosine wave signal with 3 carrier frequency of 0 KHz, 5 KHz & 10 KHz.

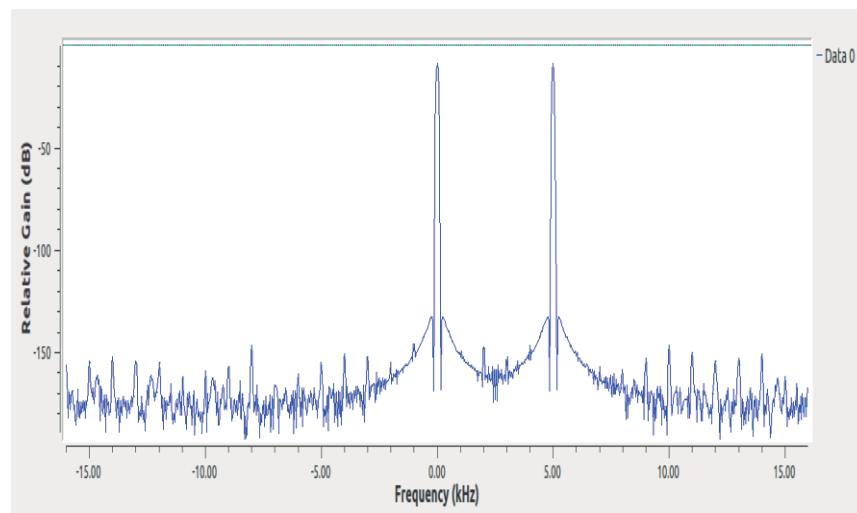


Figure 4.1.4 Low pass filtered output signal.

Low Pass Filter implementation on SDR (USRP B100)

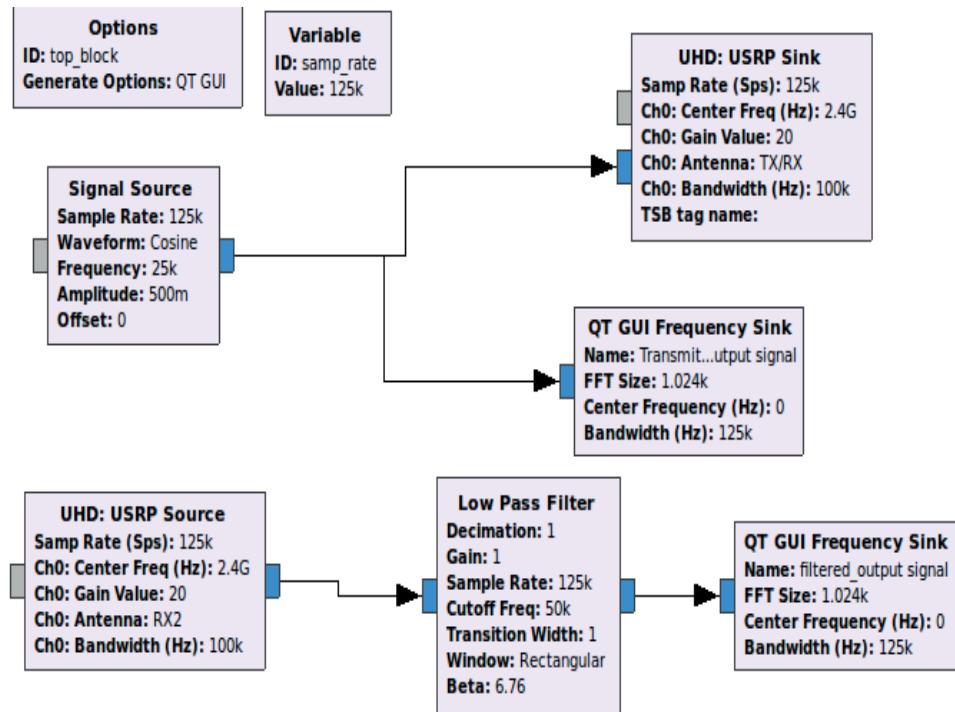


Figure 4.1.5 Low Pass filters validation using USRP B100.

Fig 4.1.5 represents the implementation of Low pass filter using Software Defined Radio (USRP B100). UHD: USRP Sink and UHD: USRP Source is used to interface Gnu radio with USRP. Gain is set as 20 dB which can vary up to 31.5 dB. Center frequency is tuned to 2.4 Ghz. Center frequency can be varied up to 6GHz. beyond 6GHz, USRP B100 cannot support.

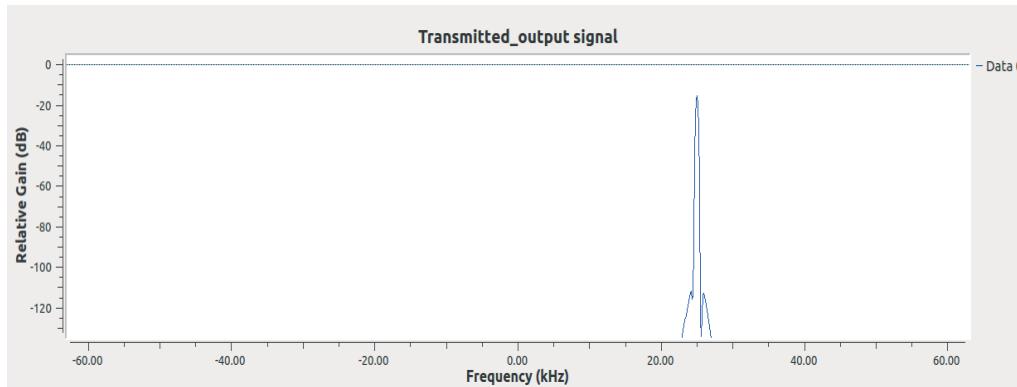


Figure 4.1.6 Transmitted signals using USRP B100.

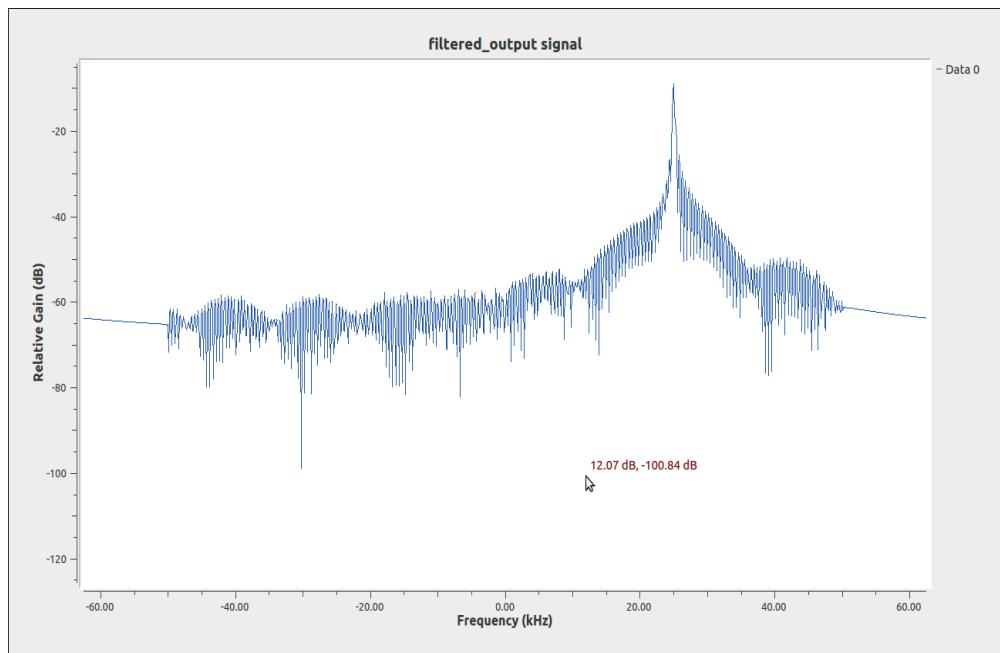


Figure 4.1.7 Filtered output signals within the cut off frequency “50 KHz”.

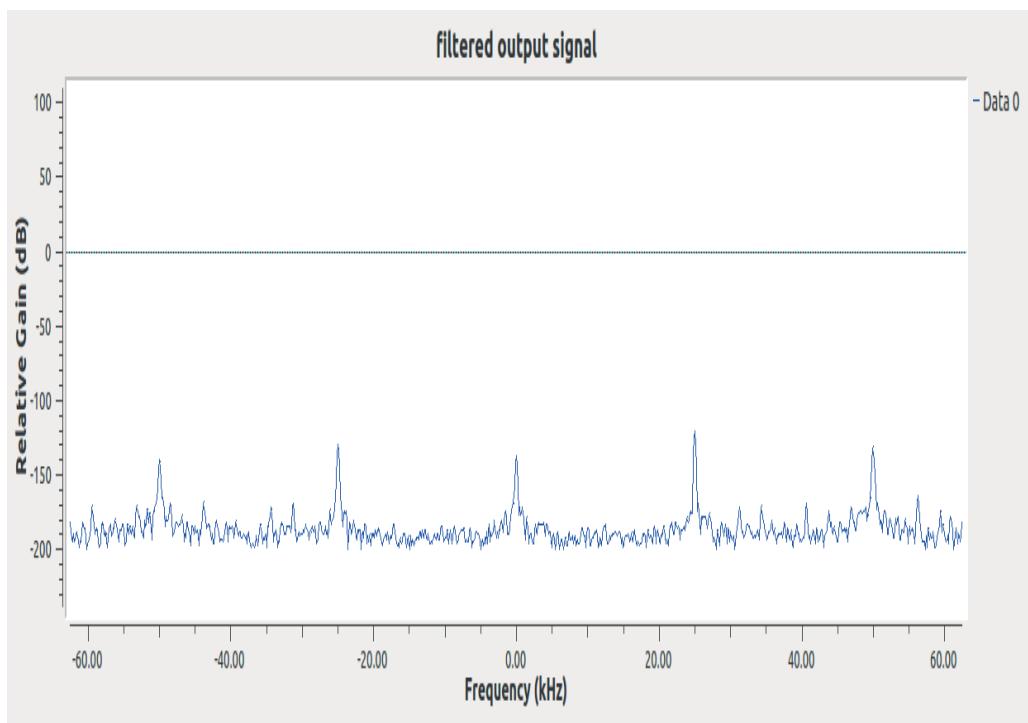


Figure 4.1.8 Filtered output signals beyond the cut off frequency “20 KHz”.

Inference

From fig 4.1.3 and 4.1.4, the results can be brought down to the conclusion that out of the three carrier signals namely 0 KHz, 5KHz, 10KHz , only low frequency signals (I.e.0KHz and 5KHz) is allowed to pass through it and 10 KHz signal is attenuated ,as it lies beyond the cut off frequency defined ($f_c= 7\text{KHz}$). The similar results (fig 4.1.6 and 4.1.7) are obtained; when a carrier signal of 25 KHz with the low pass filter is validated using USRP B100. The cut off frequency defined is 50 KHz. Fig 4.1.8 indicates a noise signal with “No “carrier signals (25 KHz), when the cut off frequency is adjusted to 20 KHz. But unlike a Fig 4.1.4, the result in fig 4.1.7 is a combination of small amount of an unwanted signal along with 25 KHz frequency signals which can be minimized further with the proper selection of windowing techniques and also reduction in the transition width.

EXPERIMENT 2

Aim

The purpose of this experiment is to analyze the effects of Low Pass Filter on the Audio signals using Gnu Radio platform.

Introduction

A Low pass filter rejects all unwanted high frequencies signals and allows only low frequency signals through it.

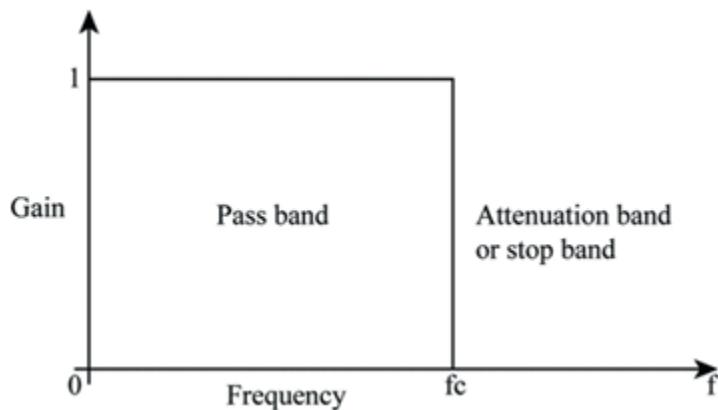


Figure 4.2.1 Frequency response of Low pass filter.

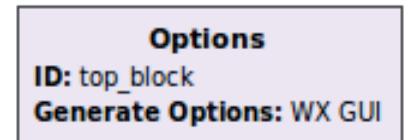
1. Low pass filter consist of pass band and stop band separated by the cut off frequency as shown in the above fig 4.2.1.
2. Pass band is an area of frequency band in which attenuation (reduction in the amplitude of signals) is nearly equal to zero ($\alpha = 0$).
3. Stop band is defined as an area of frequency band in which attenuation factor is extremely high.

Cut off frequency which is denoted as f_c is defined as the boundary required in the frequency response of a system, in which energy beyond the boundary will be attenuated or reduced.

Hence Low pass filter is a filter which will allow only those signals which are lower than the cut off frequency signals, thus attenuating frequency signals greater than cut off

frequency. Fig 4.2.1 represents an ideal low pass filter which is unstable, noncausal and not rational. Practically Low pass filter will have some signals generated in stop band, which will have still attenuation less than the pass band.

Block explanation of the flow graph



1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as WX GUI.



2. Variable

Variable block is used in order to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 48K for proper conversion of analog signals to discrete sampled signals.



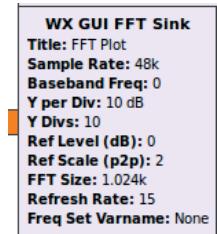
3. Wav File Source

Wav File Source is used to read the audio signals which are in (.Wav) form. The link of the wav audio signal is placed in file with the repeating mode on.



4. Audio Sink

Audio Sink is used as a speaker which will help in proper audible of audio signals with the sample rate of 48 KHz.



5. WX GUI FFT Sink

GUI stands for Graphic user interface with standard WX. It is used to represent the frequency plot for the desired output signals with FFT size selected as 1.024K by default.



6. Low pass filter

Low pass filter block is used for the purpose mentioned above with cut off frequency 5.3 KHz and transition width of 1000. Description about low pass filter is further explained below in Gnu radio flow graph section.



7. WX GUI Waterfall Sink

WX GUI Waterfall Sink represents the variation in the frequency of the Audio signals taking place with increase in high and low pitch. It can help in identifying the highest

frequency which needs to be attenuated. Hence cut off frequency can also be identified by this block.

Gnu Radio Flow graphs

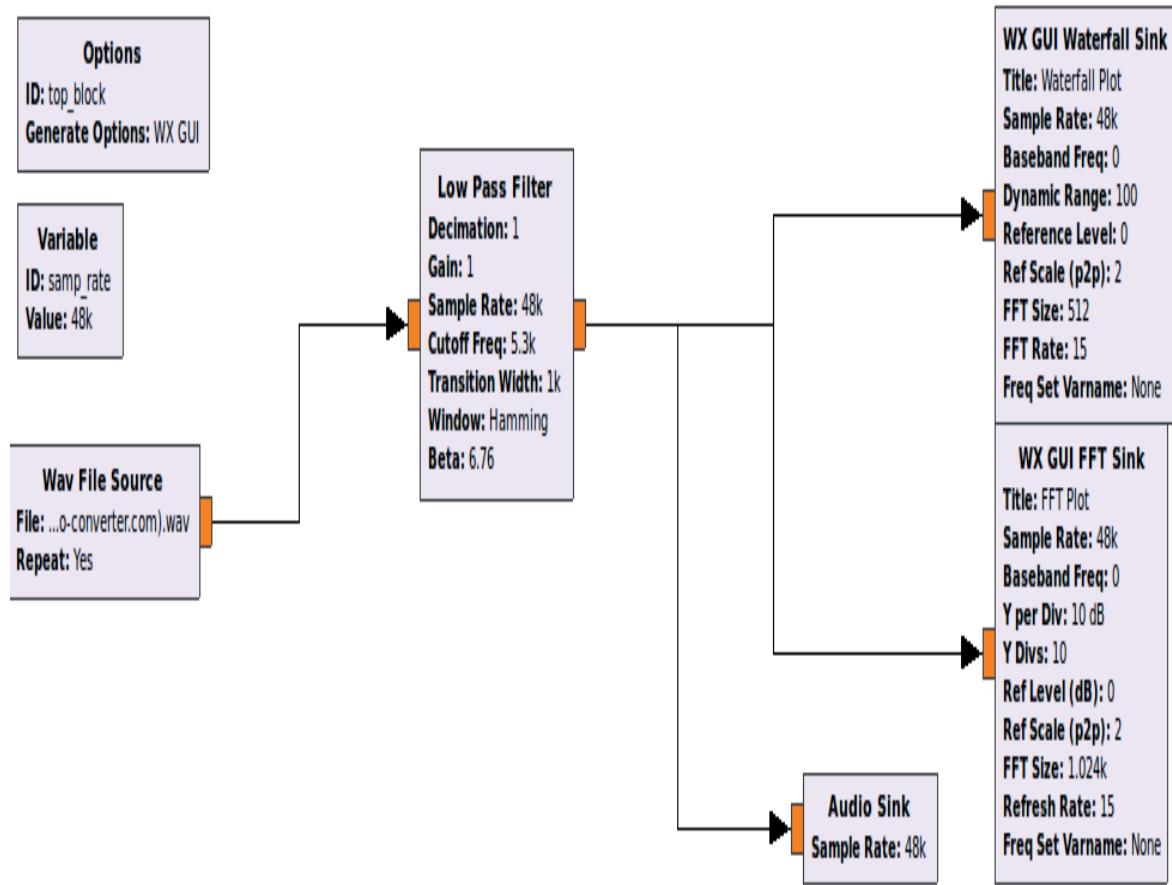


Figure 4.2.2 Low pass filter Flow graph.

The figure above addresses the architecture of low pass filter needed for this experiment. The audio signal is converted to (.Wav) file and its link is placed in the wav file Source. A wav file Source is used to analyze the audio signals .These wav output is further passed through Low pass filter (LPF). The LPF will allow the signal which is less than the cut off frequency of 5.3 KHz. The Transition width is adjusted to 1K which is basically used to ensure that the entire bandwidth of desired signals is allowed to pass through the filter. Audio sink is used to play the audio songs which are sampled as 48 KHz (the audio selected for this experiment is supporting 48 KHz sample rate).

Results

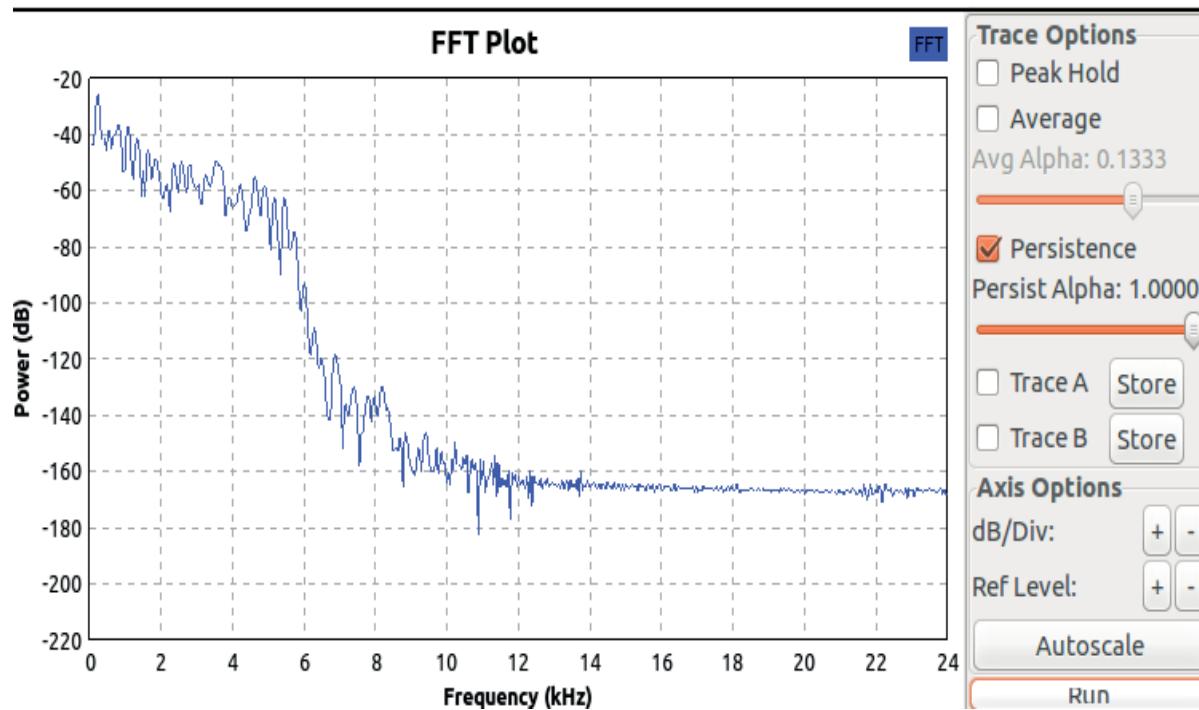


Figure 4.2.3 Frequency plot representing Pass band and Stop band.

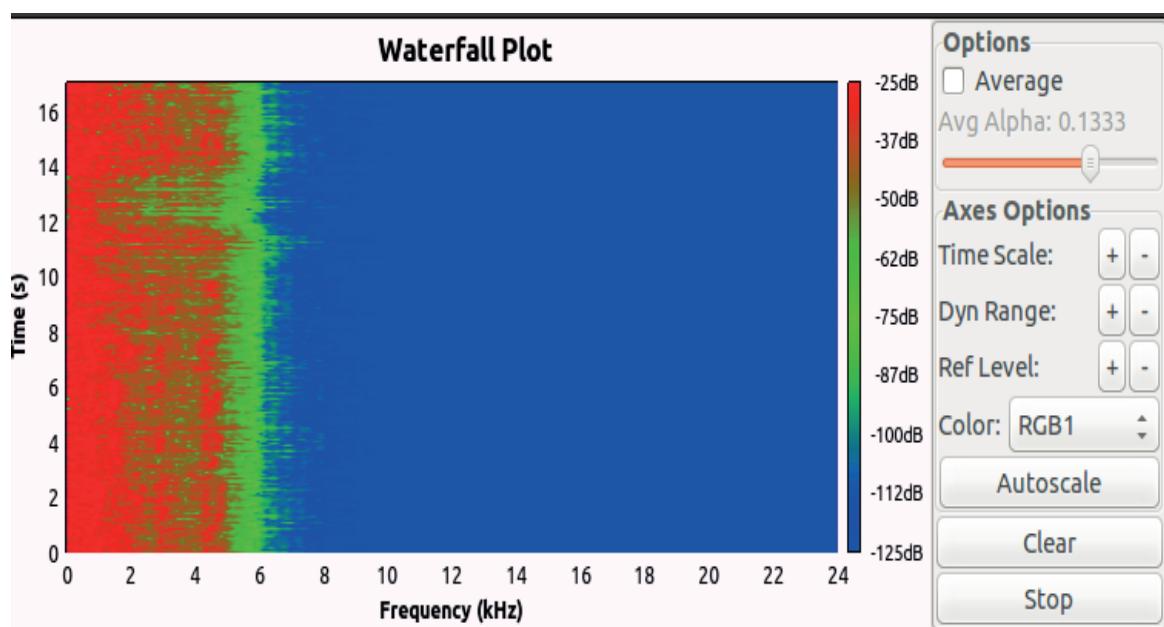


Figure 4.2.4 Waterfall plot for the audio signal.

Inference

From this experiment, it can be concluded that the audio signals need to be filtered with the sample rate supported by the input audio signals (sample rate here defined is 48 KHz, which changes based on the input signals). Fig 4.2.3 represents a frequency plot of an filtered output signals elaborating on the pass band and stop band frequency separated by the cut off frequency $f_c = 5.3$ KHz. Fig 2.4 shows a waterfall plot which indicate that the signals after 5.3 KHz frequency is attenuated, thus allowing the low frequency signals through it. It helps in avoiding the aliasing effect in digital signal processing.

EXPERIMENT 3

Aim

The purpose of this experiment understand the functioning of HIGH Pass Filter and also implement and understand the behavior using GNU Radio and USRP hardware platform.

Introduction

A High pass filter is types of filter which is designed in order to reject all the low frequencies signals, thus allowing only high frequency signals to pass through it. It allow only those signals which are greater than the cut off frequency signals, thus attenuating frequency signals lower than the cut off frequency. The cut off frequency is mathematically calculated as:

$$f_c = \frac{1}{2\pi RC}$$

Where, R= Resistor, C= Capacitor, f_c = Cut off frequency.

High pass filter is also called as low cut filter.

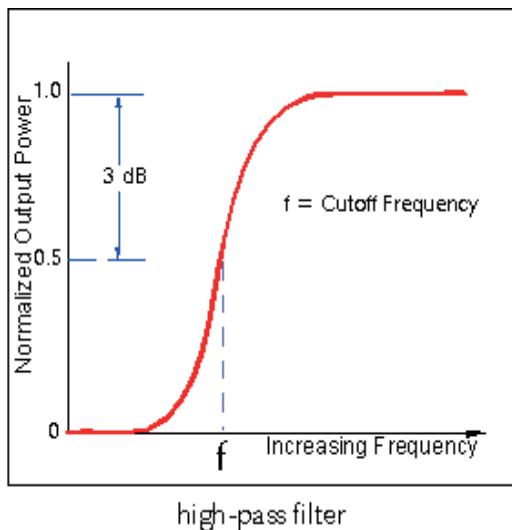


Figure 4.3.1 Frequency response of High pass filter

High pass filter consist of pass band and stop band separated by the cut off frequency as shown in the above fig 4.3.1.

Pass band is an area of frequency band in which attenuation (reduction in the amplitude of signals) is nearly equal to zero ($\alpha = 0$). Stop band is defined as an area of frequency band in which attenuation factor is extremely high. The frequency response of High pass filter is opposite as compared to low pass filter with stop band first followed by pass band.

Advantages of High pass filter

1. Remove a small amount of low frequency noise from an N dimensional signal.

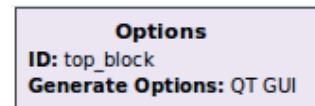
Disadvantages of High pass filter

1. High pass filter filters out the DC offset of a signal.
2. An unwanted ripple in the pass band/stop band is generated if the component is not properly selected.

Application of High pass filter

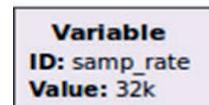
1. Used in Audio amplifier for amplifying the audio signals.
2. Used in imaging processing and in various digital signals processing application.
3. Used in the loud speakers to reduce the low level noise

Block explanation of the flow graph



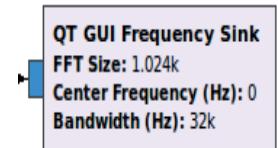
1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as QT GUI.



2. Variable

Variable block is used in order to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 32K for proper conversion of analog signals to discrete sampled signals.



3. QT GUI frequency Sink

GUI stands for Graphic user interface with standard QT. Frequency sink is used to represent the frequency plot for the desired output signals with FFT size selected as 1.024K by default.



4. High pass filter

High pass filter block is used to allow high frequency signals with signals greater than cut off frequency 4 KHz and have a transition width of 5. Description about High pass filter is further explained below in Gnu radio flow graph section.

Gnu Radio Flow graphs

The figure above provides information about three carrier signals with the carrier frequency of 0 KHz, 5 KHz, and 10 KHz respectively. These carrier signals are added and passed through the throttle in order to avoid the congestion happening in system memory. The sample rate of 32 K is considered (It can change as per the experimental requirement). The output of throttle signal is then allowed to pass through the High pass filter block having cut off frequency 4 KHz and a transition width of 5. Transition

width is basically used to ensure that the steepness of desired signals is attenuated by the filter.

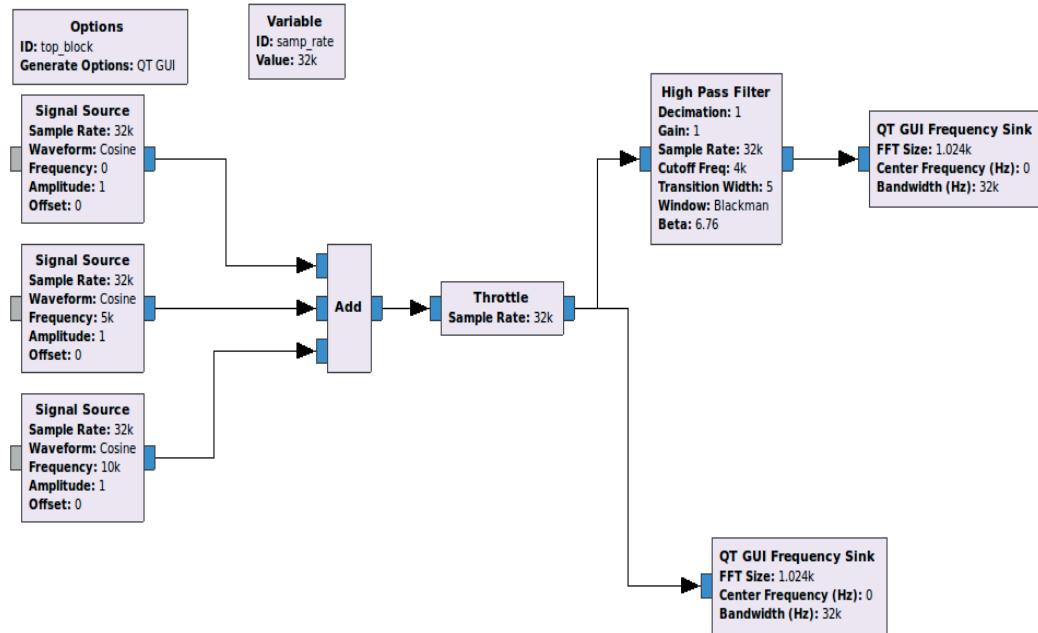


Figure 4.3.2 High pass filter Flow graph.

Results

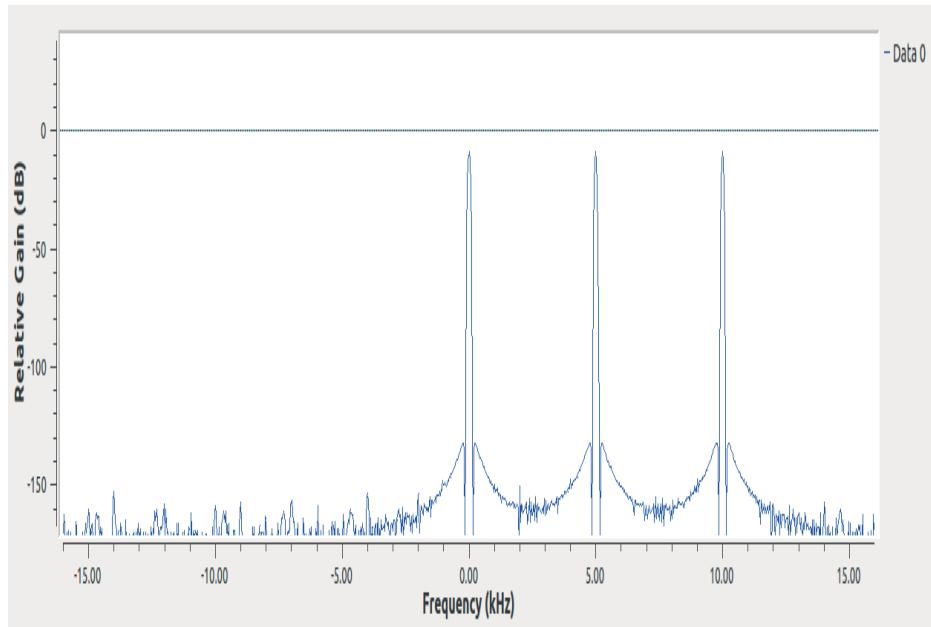


Figure 4.3.3 Input cosine wave signal with 3 carrier frequency of 0 KHz, 5 KHz & 10 KHz.

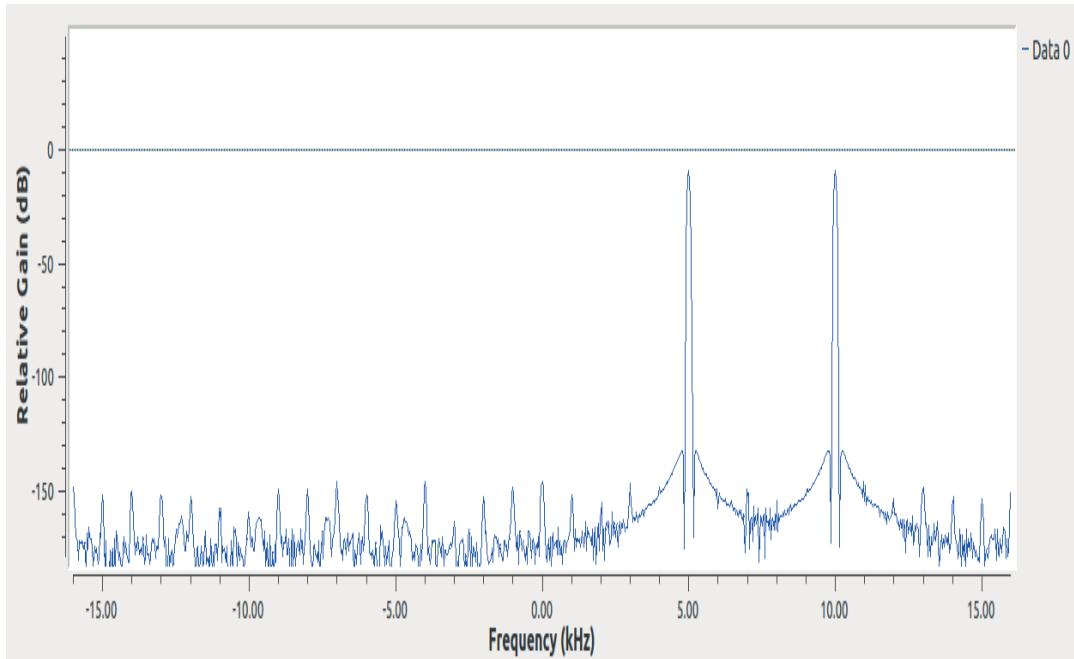


Figure 4.3.4 Filtered output signal.

Implementation of High pass filter using SDR (USRP B100)

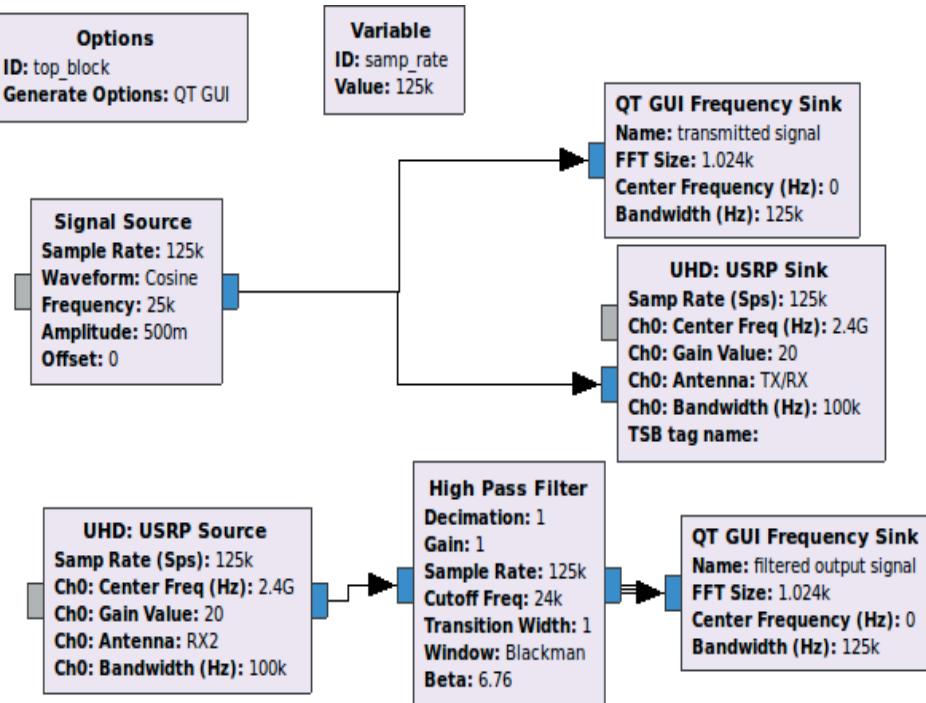


Figure 4.3.5 High pass filter implementation using USRP B100.

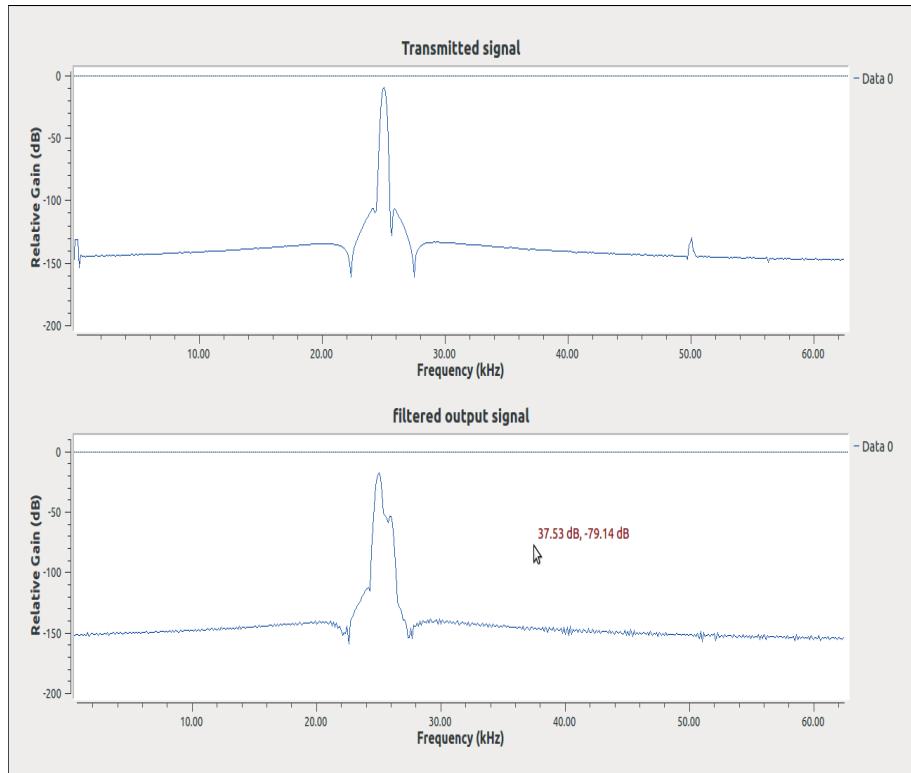


Figure 4.3.6 Validation result using USRP B100.

Inference

From fig 4.3.3 and 4.3.4, it has been observed that the carrier frequency 0 KHz is attenuated, as it lies below the cut off frequency defined ($f_c = 4 \text{ KHz}$). The carrier signal with 5 KHz and 10 KHz are passed further through the High pass filter, since these signals are greater than the cut off frequency. The similar results is obtained, when a carrier signal of 25 KHz with High pass filter is validated using Software Defined Radio (USRP B100). The validated results are well explained in figure 4.3.6 mentioned above.

EXPERIMENT 4

Aim

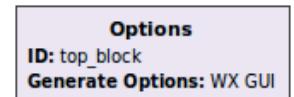
The purpose of this experiment is to analyze the effects of High Pass Filter on the Audio signals using Gnu Radio platform.

Introduction

A High pass filter rejects all low frequencies signals which is less than the cut off frequency and allows only high frequency signals which are greater than the cut off frequency through it. High pass filter is opposite to Low pass filter. It consists of first stop band followed by pass band separated by the cut off frequency as describe bellow:

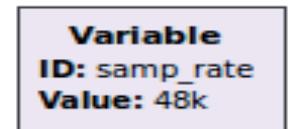
1. Pass band is an area of frequency band in which attenuation (reduction in the amplitude of signals) is nearly equal to zero ($\alpha = 0$).
2. Stop band is defined as an area of frequency band in which attenuation factor is extremely high.
3. Cut off frequency which is denoted as f_c is defined as the boundary required in the frequency response of a system, in which energy beyond the boundary will be attenuated or reduced.

Block explanation of the flow graph



1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as WX GUI.



2. Variable

Variable block is used in order to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 48K for proper conversion of analog signals to discrete sampled signals.



3. Wav File Source

Wav File Source is used to read the audio signals which are in (.Wav) form. The link of the wav audio signal is placed in file with the repeating mode on.



4. WX GUI FFT Sink

GUI stands for Graphic user interface with standard WX. It is used to represent the frequency plot for the desired output signals with FFT size selected as 1.024K by default.



5. High pass filter

High pass filter block is used to attenuate all the signals with frequency greater than cut off frequency 4.19 KHz and transition width of 1000. Description about High pass filter is further explained below in Gnu radio flow graph section.

WX GUI Waterfall Sink
Title: Waterfall Plot
Sample Rate: 48k
Baseband Freq: 0
Dynamic Range: 100
Reference Level: 0
Ref Scale (p2p): 2
FFT Size: 512
FFT Rate: 15
Freq Set Varname: None

6. WX GUI Waterfall Sink

WX GUI Waterfall Sink represents the variation in the frequency of the Audio signals taking place with increase in high and low pitch. It can help in identifying the highest frequency which needs to be attenuated. Hence cut off frequency can also identified by this block.

Gnu Radio Flow graphs

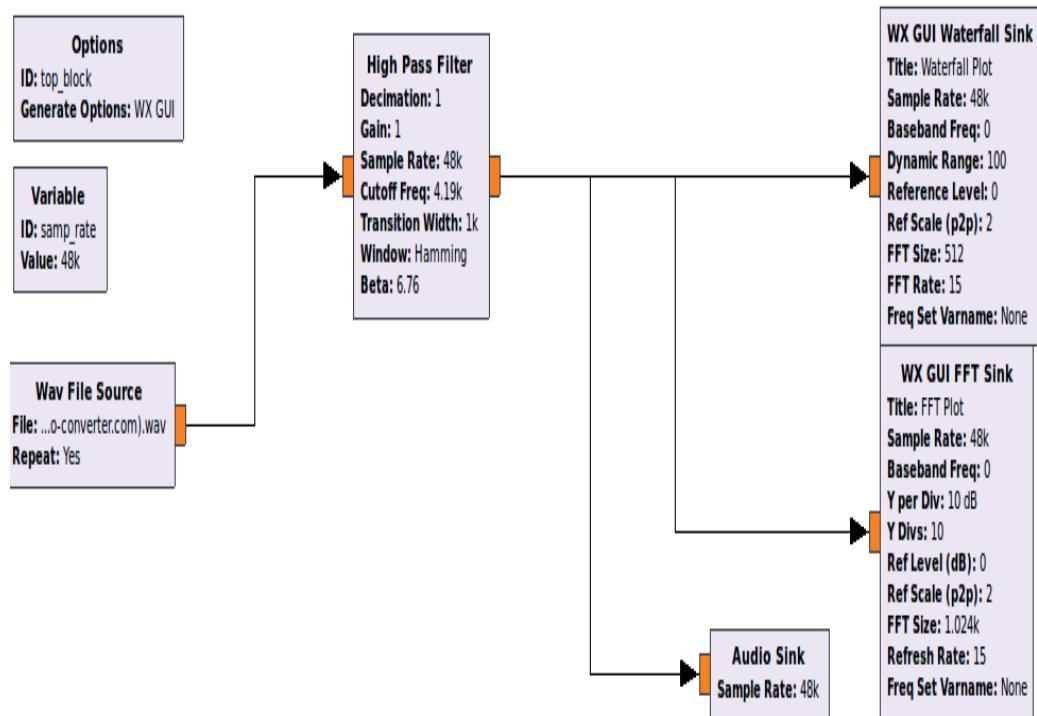


Figure 4.4.1 High pass filter Flow graph.

The figure above provides information about the architecture needed for this experiment. The audio signals are converted to (.Wav) file and its link is placed in the wav file Source, which is used in order to analyze the audio signals. These wav output is

further passed through High Pass Filter (HPF). The HPF will allow the signals which are greater than the cut off frequency of 4.19 KHz. The Transition width is adjusted to 1K which is basically used to ensure that the entire bandwidth of desired signals is allowed to pass through the filter. Audio sink is used to play the audio songs which are sampled as 48 KHz (the audio selected for this experiment is supporting 48 KHz sample rate).

Results

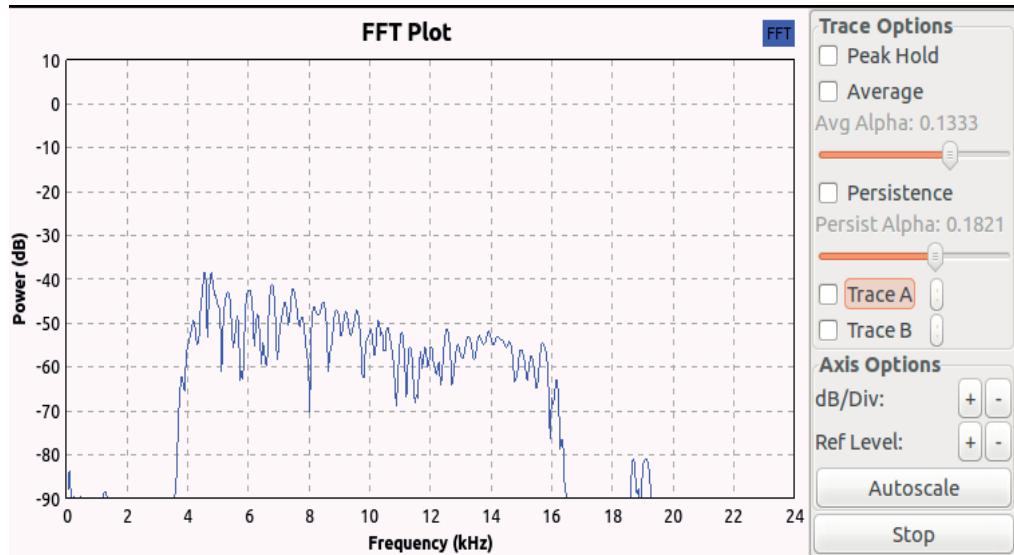


Figure 4.4.2 Frequency plot representing Pass band and Stop band.

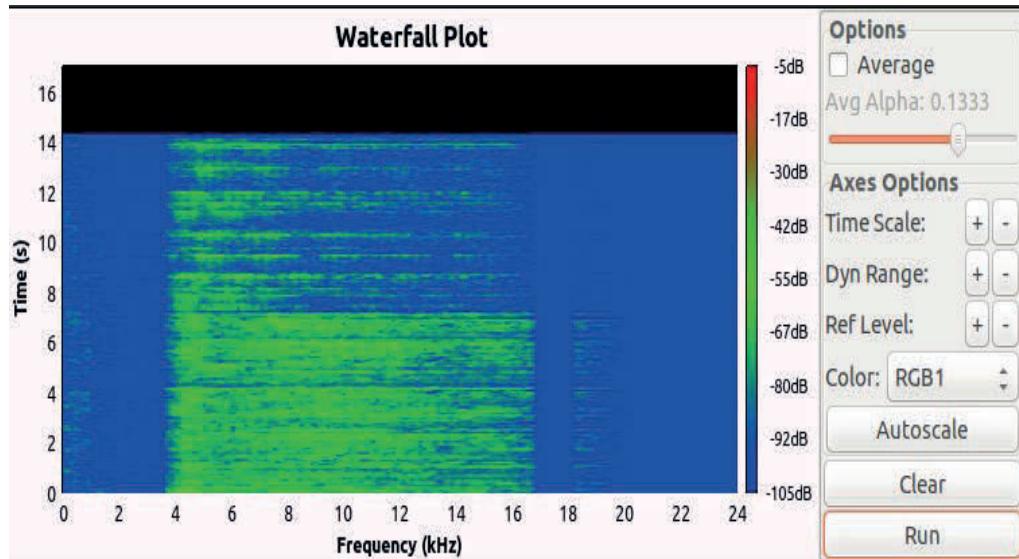


Figure 4.4.3 Waterfall plot for the audio signals

Inference

From this experiment, the effect of High pass filter on the input audio signals can be well analyzed in terms of frequency domain and waterfall diagram. Fig 4.4.2 represents a frequency plot of a filtered signals, elaborating on the stop band and pass band frequency separated by the cut off frequency approximately $f_c = 4.1$ KHz. Fig 4.4.3 represent a waterfall diagram indicating the variation in the frequency signals with change in the time domain .

EXPERIMENT 5

Aim

The purpose of this experiment understand the functioning of Band Pass Filter and also implement and understand the behavior using GNU Radio and USRP hardware platform.

Introduction

A Low pass filter is used to filter out the unwanted high frequency signals thus allowing the low frequency signals. A High pass filter is types of filter which is designed in order to reject all the low frequencies signals, thus allowing only high frequency signals to pass through it. A filter circuits need to be designed in such a way that a filter can combine the properties of the LPF (low pass filter) and HPF (high pass filter) into a single filter, which is known as a bandpass filter. Band Pass Filter shows the ability to pass frequencies relatively unattenuated over a specified frequency band.

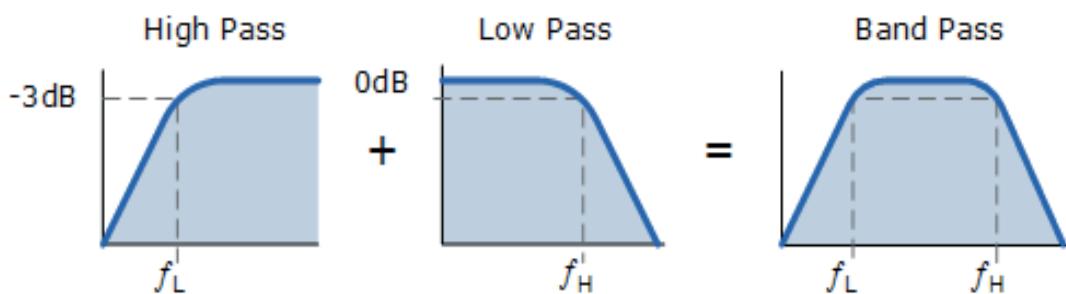


Figure 4.5.1 Frequency response of Band pass filter.

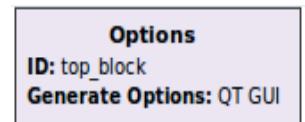
It is a combination of stop band and pass band separated by lower cut off frequency f_L and higher cut off frequency f_H .

Application of Band pass filter

1. Bandpass filters are widely used in wireless communication system.

2. In a receiver, a bandpass filter allows signals within a selected range of frequencies to be heard thus preventing the signals from getting unwanted frequencies.
3. Bandpass filters are used in all kinds of instrumentation, Sonar, even medical applications like Electrocardiograms, EEGs etc.

Block explanation of the flow graph



1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as QT GUI.



2. Variable

Variable block is used in order to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 32K for proper conversion of analog signals to discrete sampled signals.



3. Band pass filter

Band pass filter block is used to allow the frequency unattenuated between the low cut off frequency 1.5 KHz and the high cut off frequency of 7 KHz. Transition widths is

selected as 5 in order to ensure that all the steepness in the desired frequency lies beyond the bandwidth is attenuated.

Gnu Radio Flow graphs

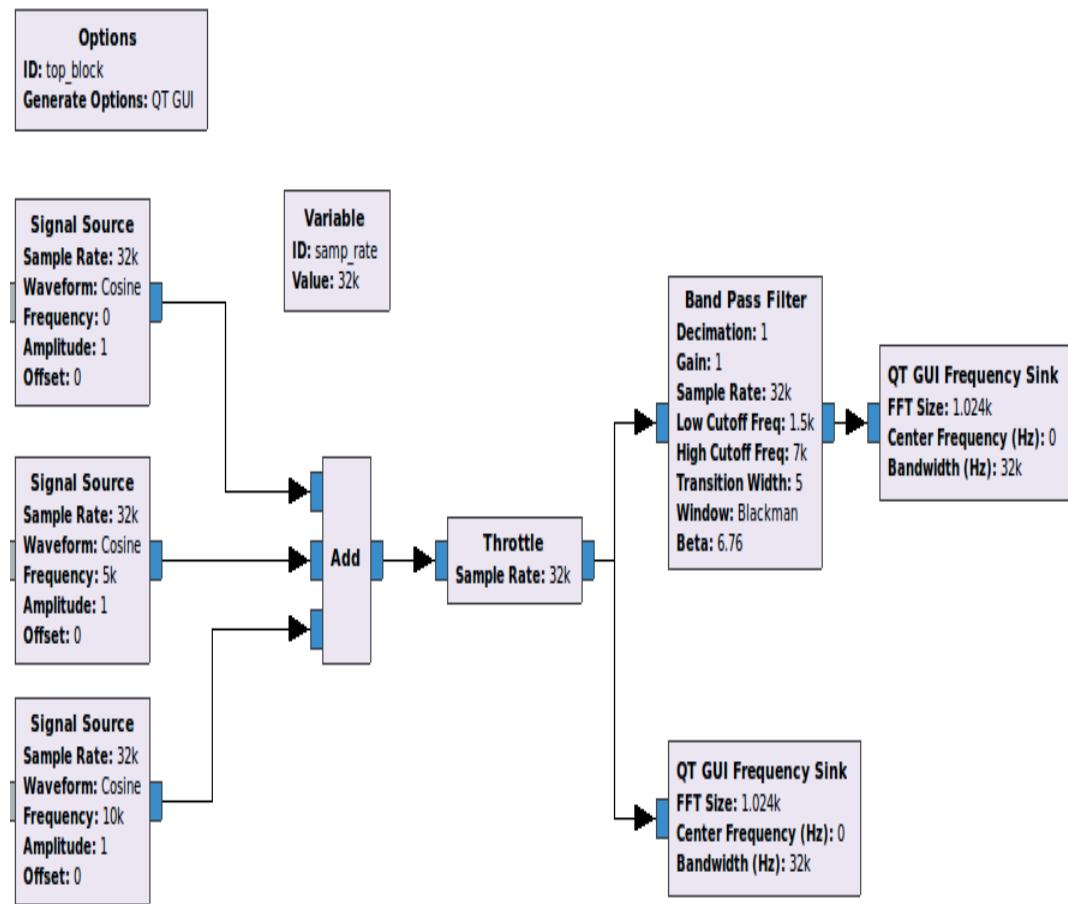


Figure 4.5.2 Band pass filter Flow graph.

The figure above provides information about three carrier signals with the carrier frequency of 0 KHz, 5 KHz, and 10 KHz respectively. These carrier signals are added and passed through the throttle in order to avoid the congestion happening in system memory. The sample rate of 32 K is considered (It can change as per the experimental requirement). The output of throttle signal is then allowed to pass through the Band pass filter block having low cut off frequency 1.5 KHz and high cut off frequency 7 KHz with a transition width of 5. Transition width as mentioned earlier in Low pass filter is used to Control the steepness in the attenuation of the signal.

Results

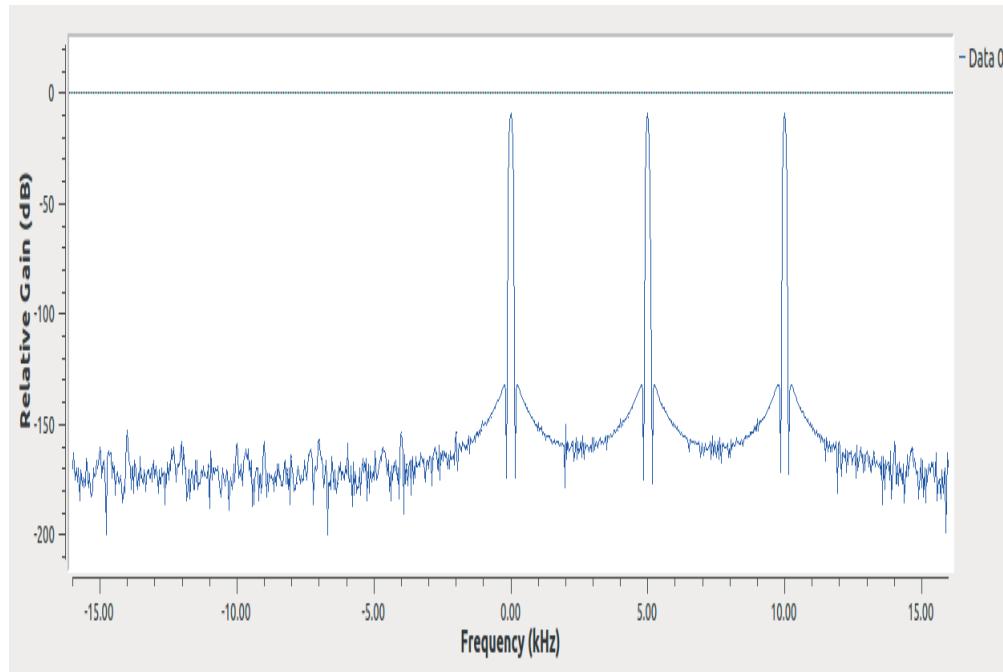


Figure 4.5.3 Input cosine wave signal with 3 carrier frequency of 0 KHz, 5 KHz & 10 KHz.

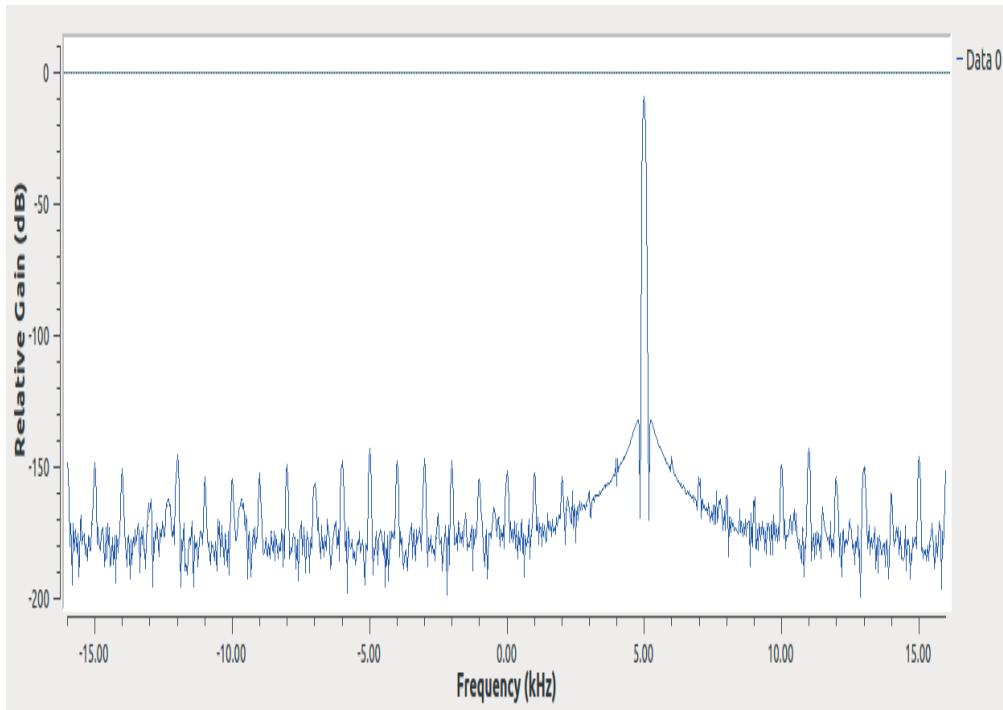


Figure 4.5.4 Filtered output signal.

Implementation of Band Pass Filter using SDR (USRP B100).

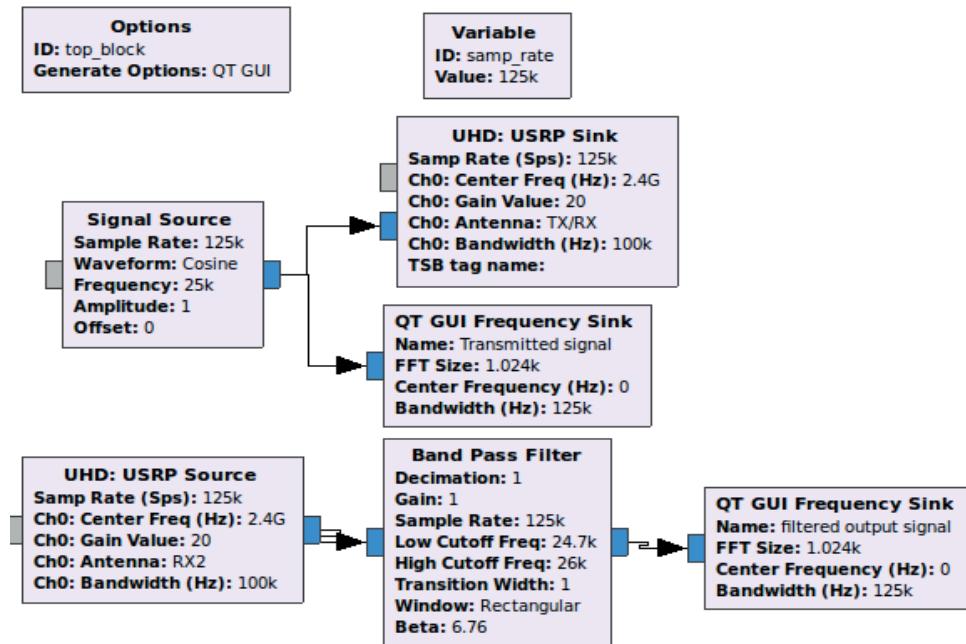


Figure 4.5.5 Band pass filter Flow graph.

Results

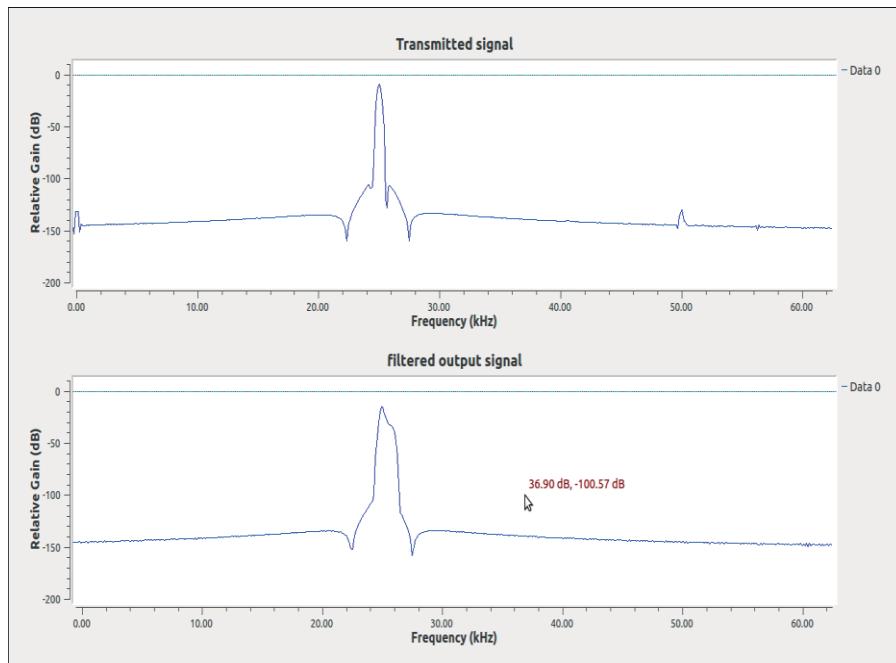


Figure 4.5.6 Validated output signals using USRP B100.

Inference

It can be concluded that , the carrier signal lying within the cut off frequency (1.5KHz and 7KHz) will be accepted by the Band pass filter(I.e. 5KHz) , thus rejecting all the other carrier frequency signals which lies beyond the cut off frequency defined (I.e. 0 KHz and 10 KHz). Similar results are obtained when the experiment with a carrier signal of 25 KHz along with the Band pass filter is validated using USRP B100 (fig 4.5.6). The cut off frequency signals defined 24.7 KHz and 26 KHz.

EXPERIMENT 6

Aim

The purpose of this experiment understand the functioning of Band Reject Filter and also implement and understand the behavior using GNU Radio and USRP hardware platform.

Introduction

The Band Reject filter is a type of frequency selective filter which is exactly defined opposite to the Band pass filter. It is defined as a filter which allows all the frequencies except for the band of frequency which lies between the low cut off frequency and the high cut off frequency. The Band Reject filter is also called as the Band Stop filter. It is a combination of low pass filter and high pass filter as shown in the fig 6.1.a.

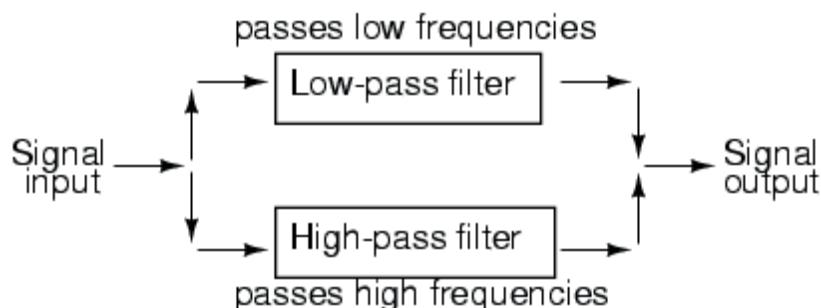


Figure 4.6.1.a the signal flow for Band Reject filters.

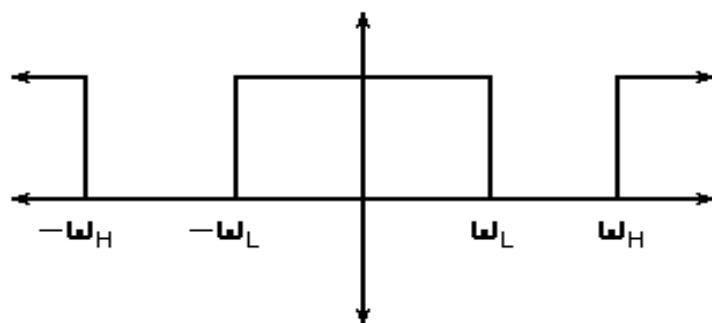


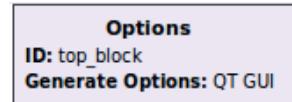
Figure 4.6.1.b Frequency response of Band Reject filters.

It is a combination of 1st as a pass band and 2nd as a stop band separated by lower cut off frequency f_L and higher cut off frequency f_H . The frequency response of band reject filter is represented in fig 6.1.b.

Application of Band Reject filter

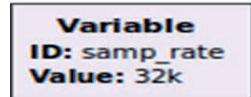
1. In image and signal processing Band Reject filter is used to reject noise.
2. It is also used in medical field applications like EGC for removing line noise.
3. It is used in high quality audio applications like buffers.
4. It is used as the telephone line noise reducers and DSL internet services.

Block explanation of the flow graph



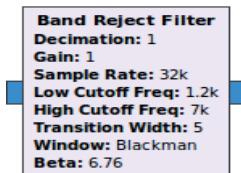
1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as QT GUI.



2. Variable

Variable block is used in order to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 32K for proper conversion of analog signals to discrete sampled signals.



3. Band Reject filter

Band Reject filter block is used to allow all the frequency except for the frequency lies between the low cut off frequency 1.2 KHz and the high cut off frequency of 7 KHz. Transition widths is selected as 5 in order to ensure that steepness of the desired frequency lies beyond the bandwidth need to attenuated.

Gnu Radio Flow graphs

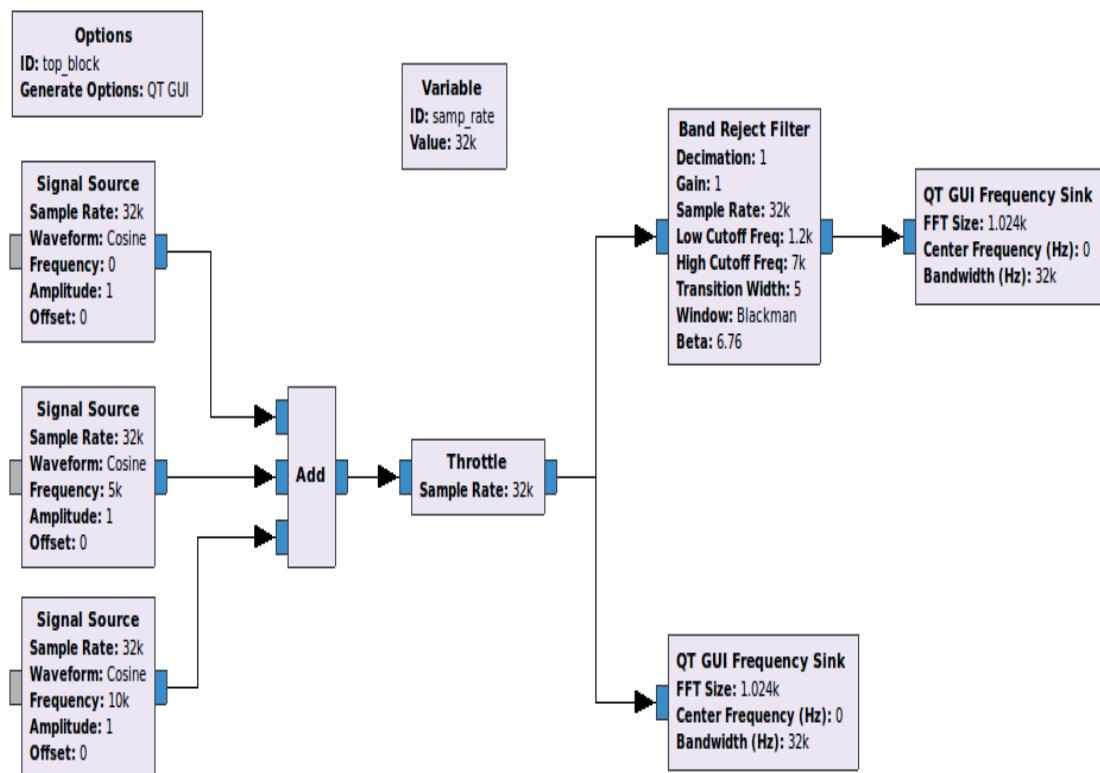


Figure 4.6.2 Band Reject filter Flow graph.

The figure above provides information about three carrier signals with the carrier frequency of 0 KHz, 5 KHz, and 10 KHz respectively. These carrier signals are added and passed through the throttle in order to avoid the congestion happening in system memory. The sample rate of 32 K is considered (It can change as per the experimental requirement). The output of throttle signal is then allowed to pass through the Band Reject filter block having low cut off frequency 1.2 KHz and high cut off frequency 7 KHz with a transition width of 5.

Results

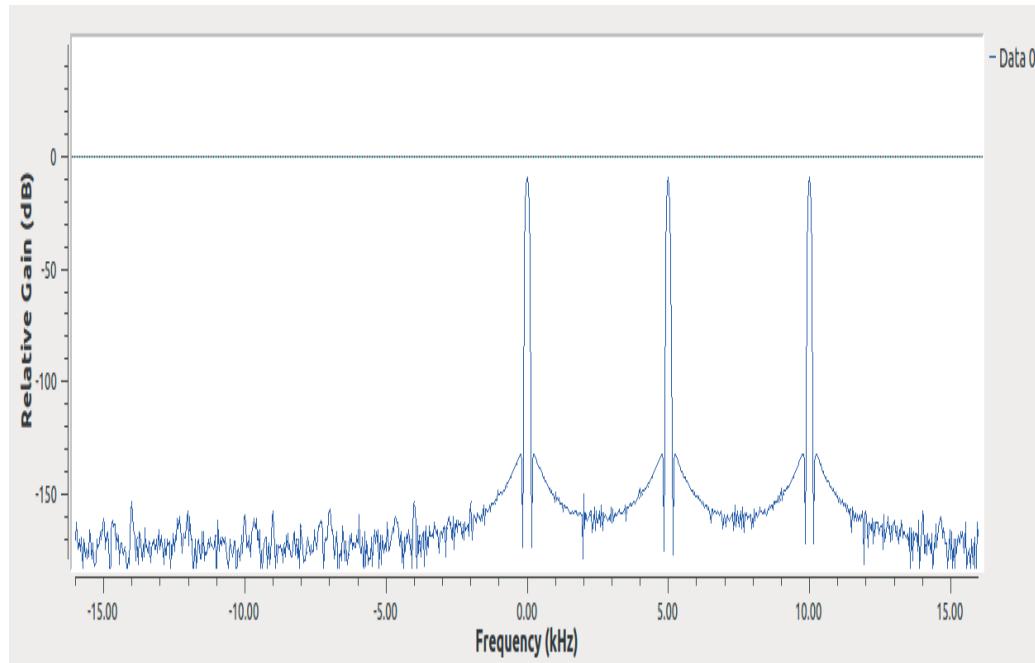


Figure 4.6.3 Input cosine wave signal with 3 carrier frequency of 0 KHz, 5 KHz & 10 KHz.

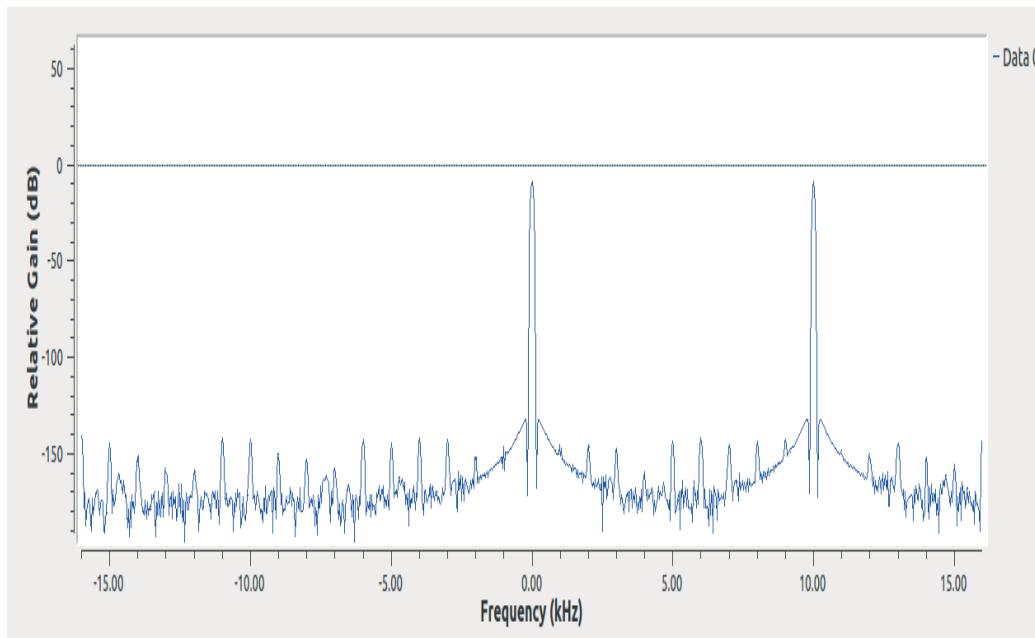


Figure 4.6.4 Filtered output signal.

Implementation of Band Rejection Filter using SDR (USRP B100)

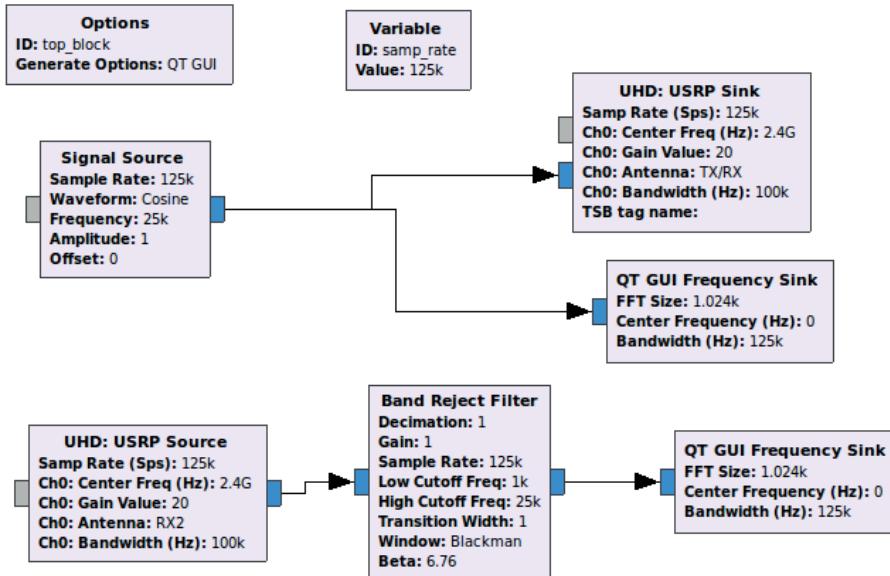


Figure 4.6.5 Band Reject Filter using USRP B100.

Results

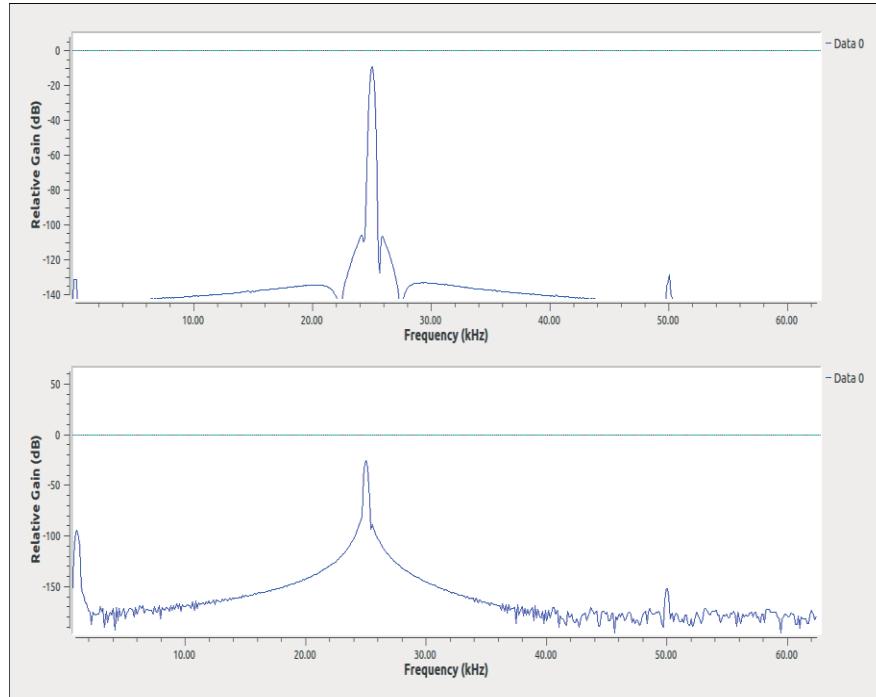


Figure 4.6.6 Validated output signal using USRP B100.

Inference

Fig 4.6.3 represents an addition of three carrier signals which on passing through the Band Reject filter, the desired output signals is obtained which is represented in fig 4.6.4. It was observed that the frequency signal within the low cut off frequency 1.2 KHz and high cut off frequency 7 KHz are attenuated, thus allowing all the other frequencies through it. Hence carrier frequency of 0 KHz and 10 KHz are not attenuated and 5 KHz is attenuated. Fig 4.6.5 represents the validation of Band Reject filter using USRP B100. Low cut off frequency and high cut off frequency is selected as 1 KHz and 25 KHz respectively. Fig 4.6.6 indicates that the frequencies lies between the cut off frequencies will be rejected, Hence frequency beyond the cut off frequency (carrier frequency of 10 KHz and unwanted carrier) are allowed.

EXPERIMENT 7

Aim

The purpose of this experiment is to analyze the effects of Finite Impulse Response Filter on the Audio signals using Gnu Radio platform.

Introduction

FIR filter is a type of digital filters used in Digital Signal Processing applications. FIR is abbreviated as Finite Impulse Response which means that it is a type of filter whose impulse response is of finite period, Hence it settles down to zero in finite time. In simple term if an impulse is applied (i.e. "1" sample), then zeroes will automatically appear after the "1" sample has made its way through the delay line of the filter. It is finite because there is no feedback in the FIR filter.

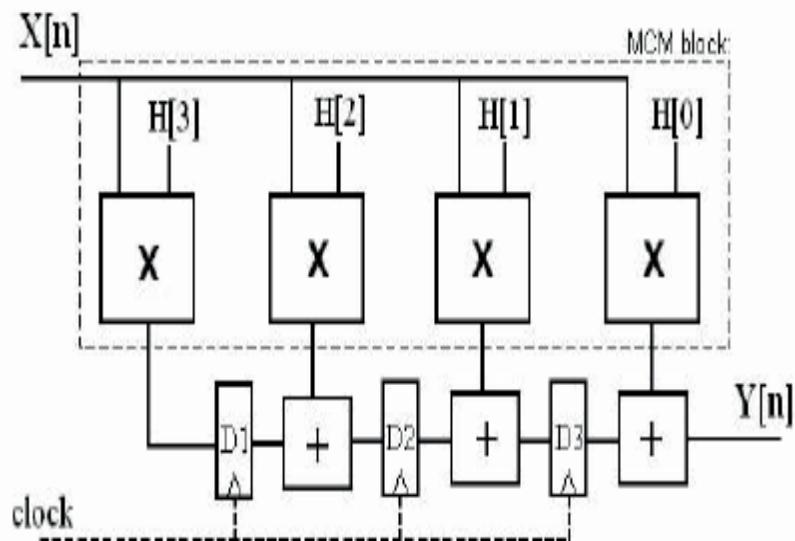


Figure 4.7.1 Finite Impulse Response filter.

The impulse response of an N th order discrete time FIR filter takes precisely $N+1$ samples before it then settles down to zero. FIR filters are designed with a multiplier, adders and a series of delays to create the output of the filter as shown in Fig 4.7.1. The result of delays operates on input samples. The output is obtained by the summation of

all the delayed samples multiplied by the appropriate coefficient. H stands for the coefficient which is used for the multiplication process. It is unique in each stage. Hence it is a process of choosing the length and coefficients of the filter. The larger the delay factor better is the resulted output signal. Delay stage at some point if it is made extremely high, then it results in the distortion in signals.

Advantage of FIR filter

1. FIR filters can have exactly linear phase
2. FIR filters are convenient and easy to implement.
3. FIR filters can have long delay between input and output.

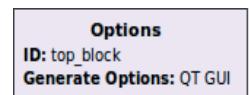
Disadvantages of FIR filter

1. Large and complicated circuit
2. Consumed too much of power.
3. Requires more memory and calculation in order to achieve the desired response.
4. Does not provide accuracy in the frequency response.

Application of FIR filter

1. It is used in Bandlimited devices which can avoid the aliasing effect and ensure that the transmitted bit lies within the bandwidth range.
2. It is used in Noise suppression techniques devices like biomedical devices, imaging devices.

Block explanation of the flow graph



1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as QT GUI.

Variable
ID: samp_rate
Value: 44k

2. Variable

Variable block is used in order to set the sample rate constant through each block. Once Sample rate is set constant in Variable block, then it will remain constant in every block. Here sample rate is considered as 44K for proper conversion of analog signals to discrete sampled signals.

Decimating FIR Filter
Decimation: 1
Taps: 2

3. Decimating FIR filter

The FIR filter provides an impulse response with finite period. The Taps is a coefficient or a delay pair. The number of Taps defines

1. The amount of memory required to implement the filter.
2. The number of calculation required.
3. The amount of filtering a filter can do.

QT GUI Waterfall Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 44k

4. QT GUI Waterfall Sink

QT GUI Waterfall Sink represents the variation in the frequency of the Audio signals taking place with increase in high and low pitch. It can help in identifying the highest frequency which needs to be attenuated. Hence cut off frequency can also identified by this block.

QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 44k
Autoscale: No

5. QT GUI Time sink

QT GUI Time sink is used to represent the output signal in time domain with the sample rate of 44 KHz and number of points as 1.024K.

Gnu Radio Flow graphs

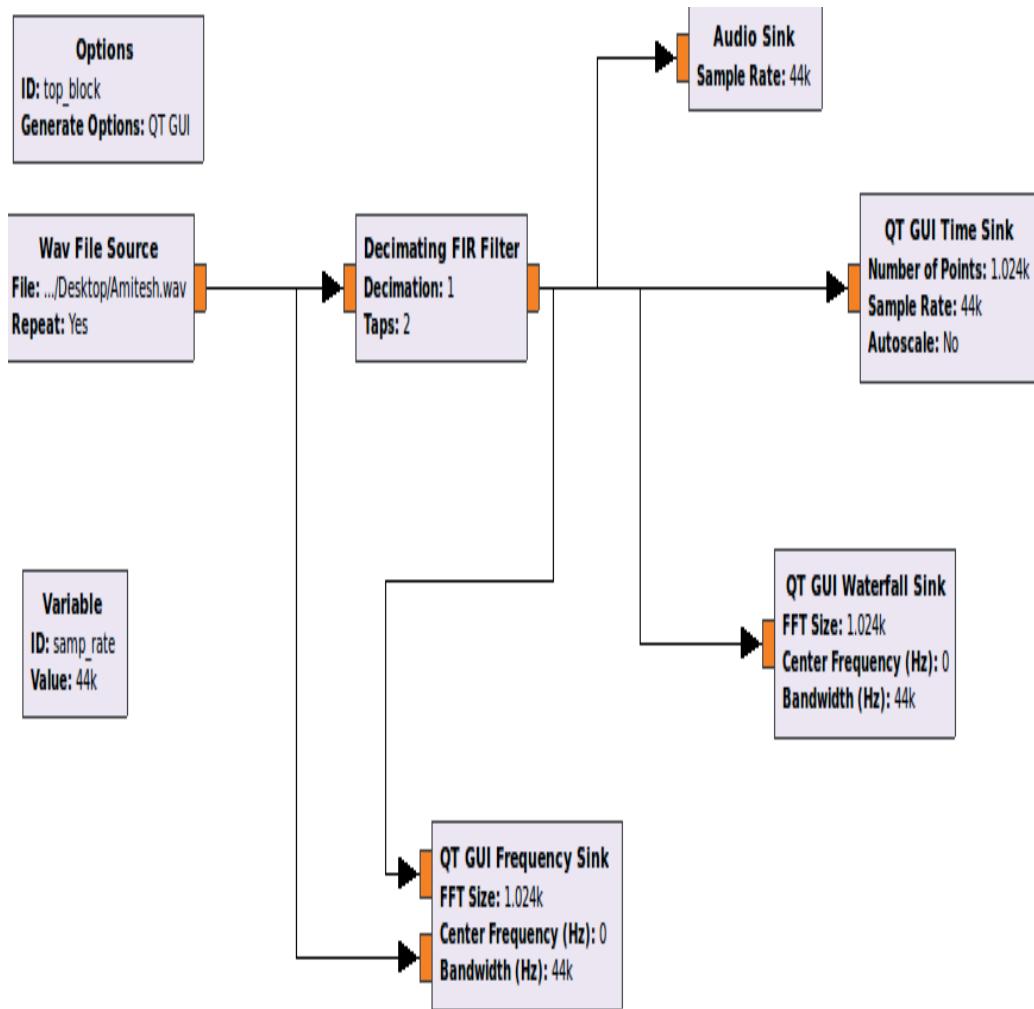


Figure 4.7.2 FIR filter Flow graph.

The figure above provides information about the architecture needed for this experiment. The audio signals are converted to (.Wav) file and its link is placed in the wav file Source, which is used in order to analyze the audio signals .These wav output is further passed through the FIR filter. The FIR filter will allow the signal to decimate as much as possible to band limit the signals, thus eliminating the noise present. The taps

is set as 2, based on the experimental input data, as taps more than 2 was leading to distortion in the information signals. Audio sink is used to play the audio songs which are sampled as 44 KHz (the audio selected for this experiment is supporting 44 KHz sample rate).

Results

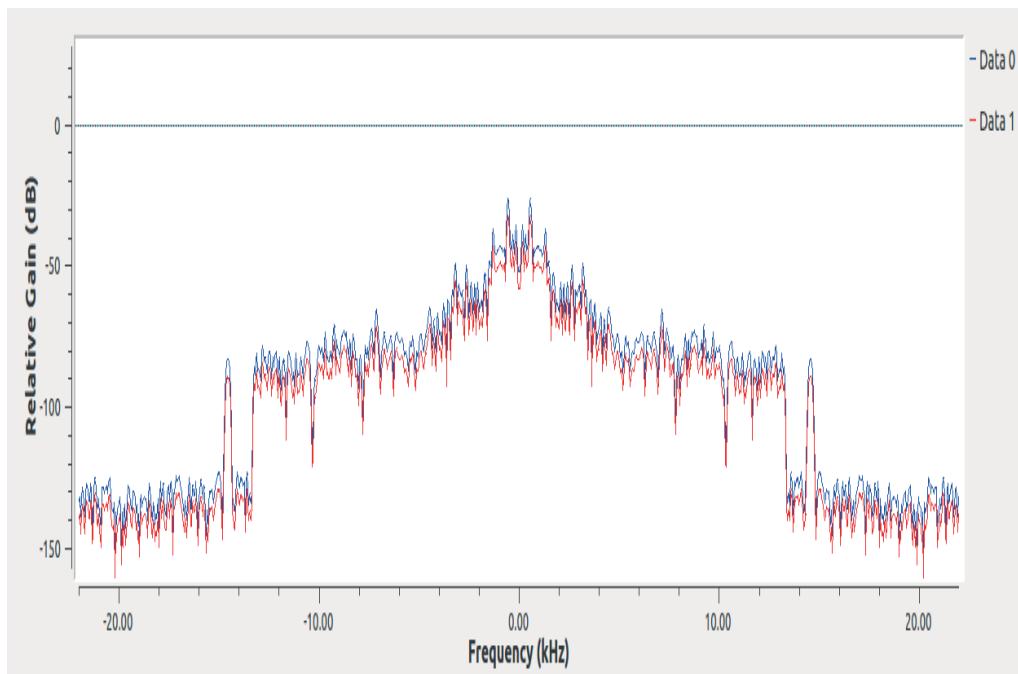


Figure 4.7.3 Frequency plot for the audio signals through FIR filter.

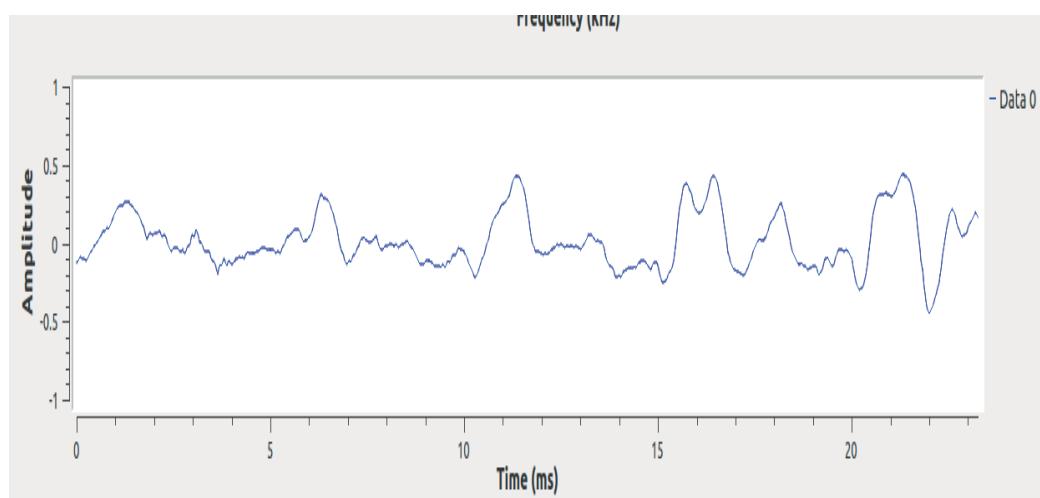


Figure 4.7.4 Time plot for the filtered audio signals

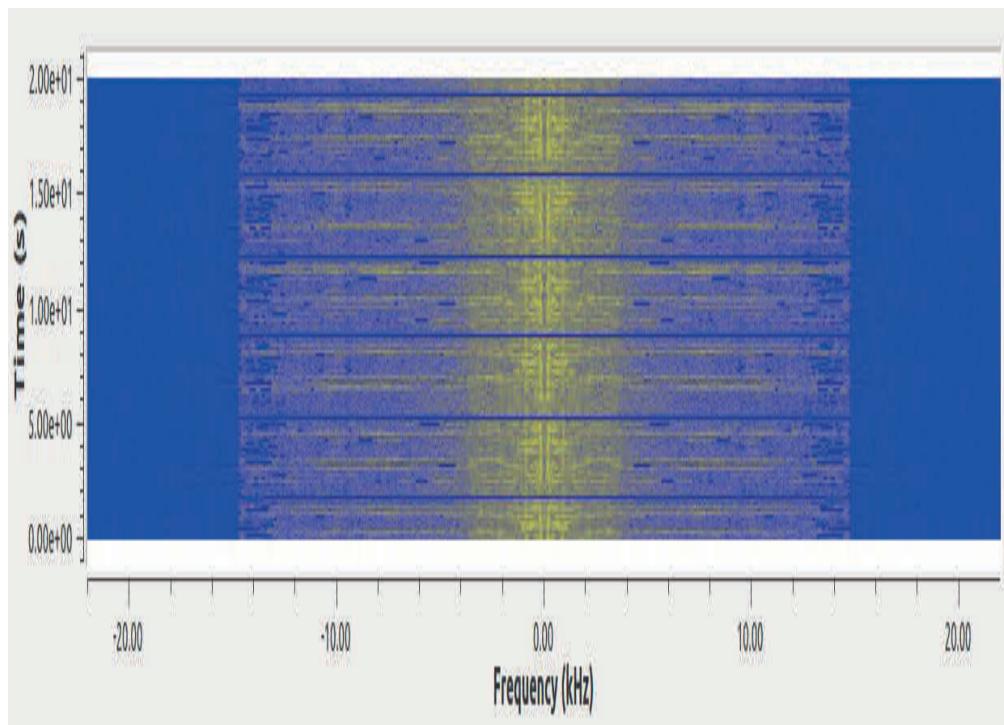


Figure 4.7.5 Waterfall diagram.

Inference

Fig 4.7.3 represents a frequency plot elaborating on the maximum possible number of attenuation done in order to avoid the noise from a given output signals. The number of taps or coefficient is set as 2 based on the try and error basis. Fig 4.7.4 provides a time domain representation of FIR filtered output signal. Fig 4.7.5 represents waterfall diagrams which indicate the variation in the types of signals taking place in terms of frequency. Hence the cut off frequency can also be identified using Waterfall diagram. It helps in avoiding the aliasing effect in digital signal processing. The disadvantage of FIR filter is that it does not provide an accurate frequency response and hence less stability is obtained. It can be solved by IIR filter which is explained in next experiment.

EXPERIMENT 8

Aim

The purpose of this experiment is to analyze the effects of Infinite Impulse Response Filter (IIR filter) on the Audio signals using Gnu Radio platform.

Introduction

IIR filter is abbreviated as Infinite impulse response filter, which is a type of digital filter used in the digital signal processing. The system is said to have an infinite impulse because the system consist of a feedback system. In simple term, if 1 sample is applied as an impulse then number of non-zeroes values will be followed after it.

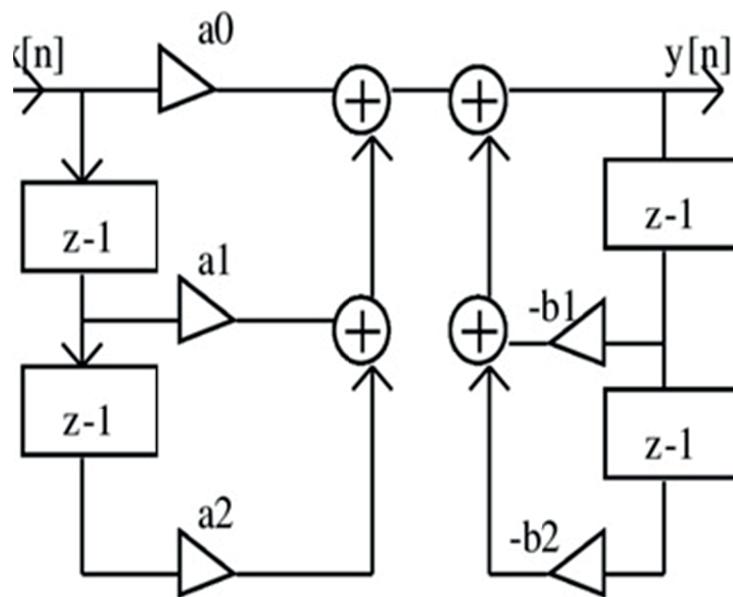


Figure 4.8.1 Infinite Impulse Response filter.

As compared to FIR system, from fig 4.8.1, it is observed that the IIR filter has large number of delays due to its feedback concept. Hence the complexity of a system increases with increase in the stability. IIR filter have no linear phase. The stability of IIR filter is also affected sometime, if feedback received is not perfect. Due to the complexity and large consumption of power in case of IIR filter, FIR filter is found to have more application.

Advantage of IIR filter

1. IIR filters have less memory
2. IIR need less calculation as compared to FIR
3. IIR provides more stability as compared to FIR due to its feedback concept.

Disadvantages of IIR filter

1. Complicated circuit due to its feedback concept.
2. It contains large number of delay, hence provide stability but reduces the efficiency.
3. It is more susceptible to problems like noise like when the output isn't computed perfectly and is fed back, the imperfection can compound.

Application of IIR filter

1. It is used in various digital signals processing application.
2. It is used in Image processing application used in medical field.
3. It is used in loudspeaker to avoid problems like crossover errors.

Block explanation of the flow graph



1. Options

Options block is used to select the standard QT or WX. All the blocks like QT GUI and WX GUI will be operated only by proper selecting the generate option. In this case Generate option is selected as QT GUI.



2. IIR filter

The IIR filter provides an impulse response with infinite period. The Taps is a coefficient or a delay pair. With an increase in the number of Taps, the number of calculation will

be more, hence providing an efficient amount of filtered output. There are two types of Taps available in IIR filter:

1. Feed-forward Taps – It's an open circuit with no feedback.
2. Feedback Taps – It's a close loop circuit providing feedback.

Gnu Radio Flow graphs

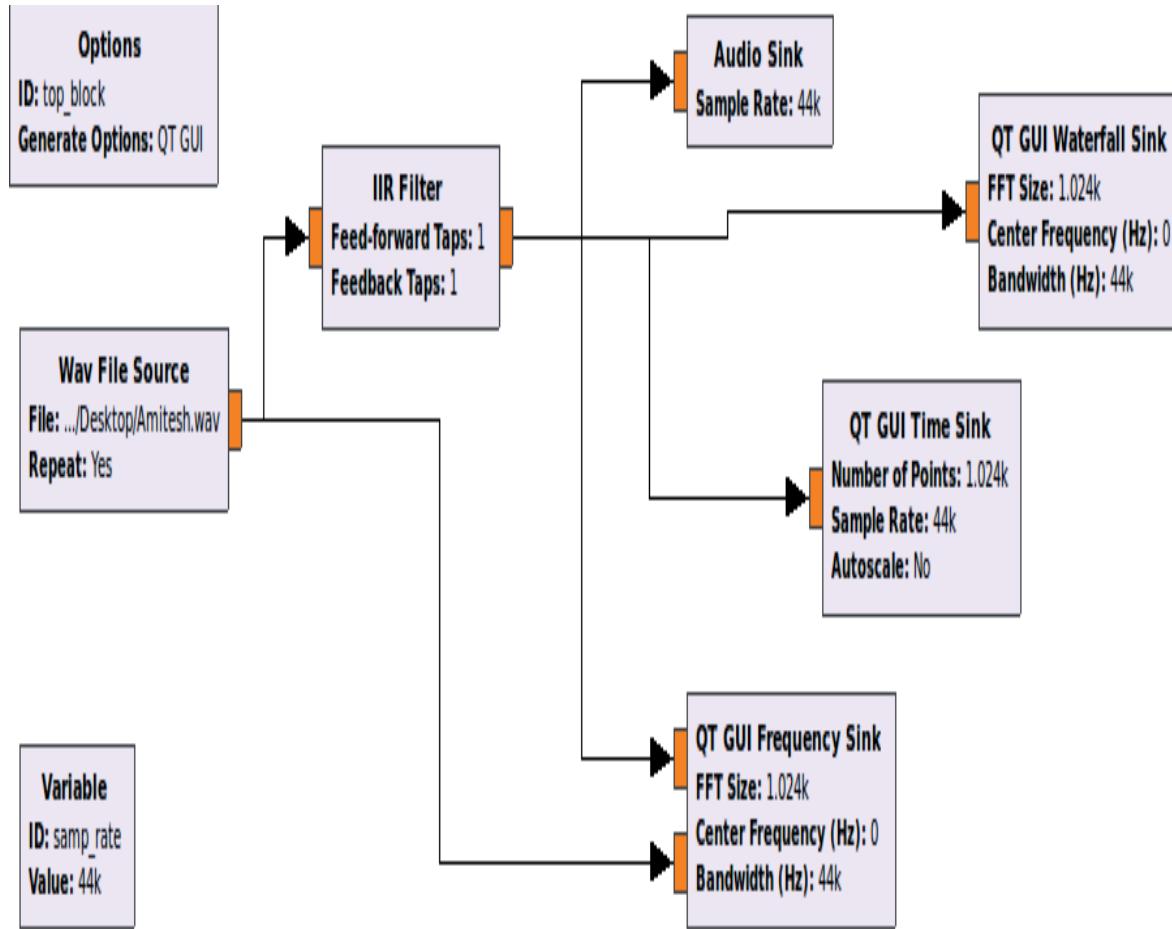


Figure 4.8.2 IIR filter Flow graph.

The audio signals is converted to (.Wav) file and its link is placed in the wav file Source, which is used in order to analyze the audio signals .These wav output is further passed through the IIR filter. The IIR filter will allow the signal to attenuate as per the feedback provided based on the past output data, thus allowing to band limit the signals, and eliminate the noise present in it. The FIR and IIR filter works on the similar fashion, with the only difference is that the IIR filter consists of a closed feedback loop. Audio sink is

used to play the audio songs which are sampled as 44 KHz (the audio selected for this experiment is supporting 44 KHz sample rate, so sample rate can be selected based on the input audio signals used).

Results

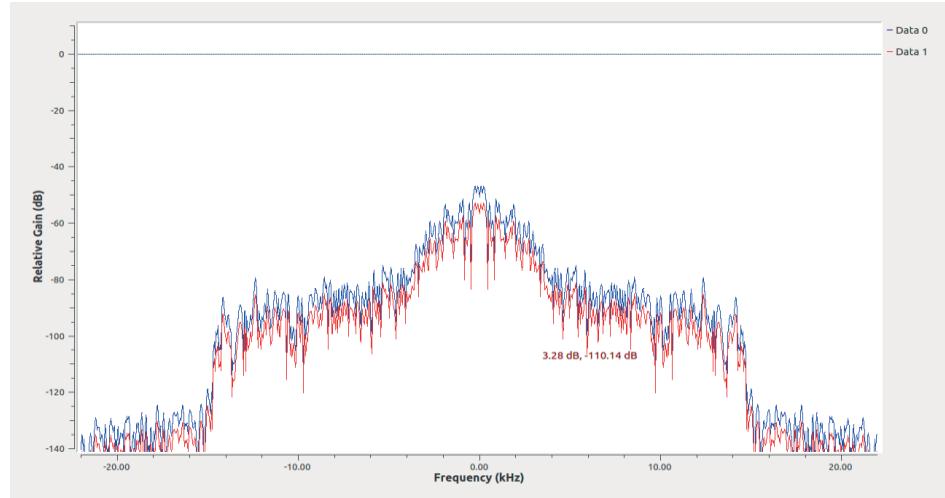


Figure 4.8.3 Frequency plot for the audio signals through IIR filter.

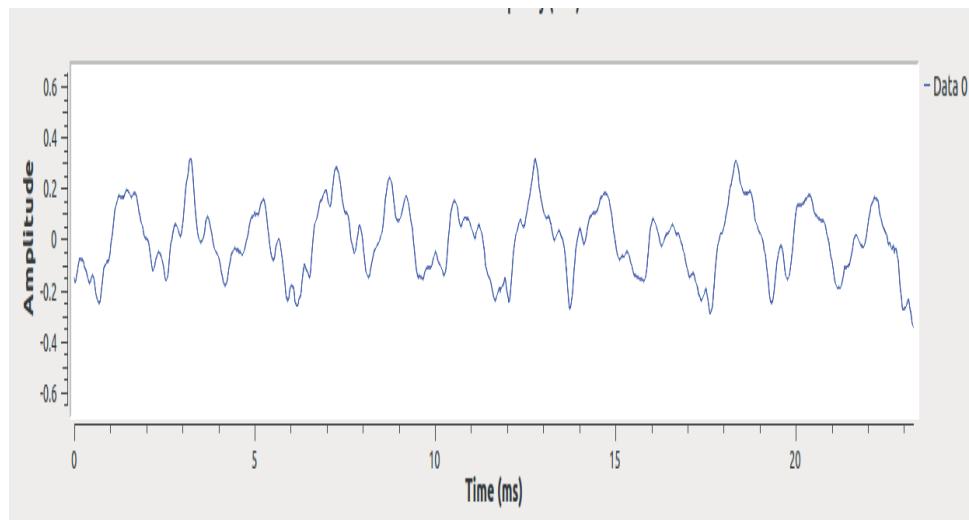


Figure 4.8.4 Time plot for the filtered audio signals

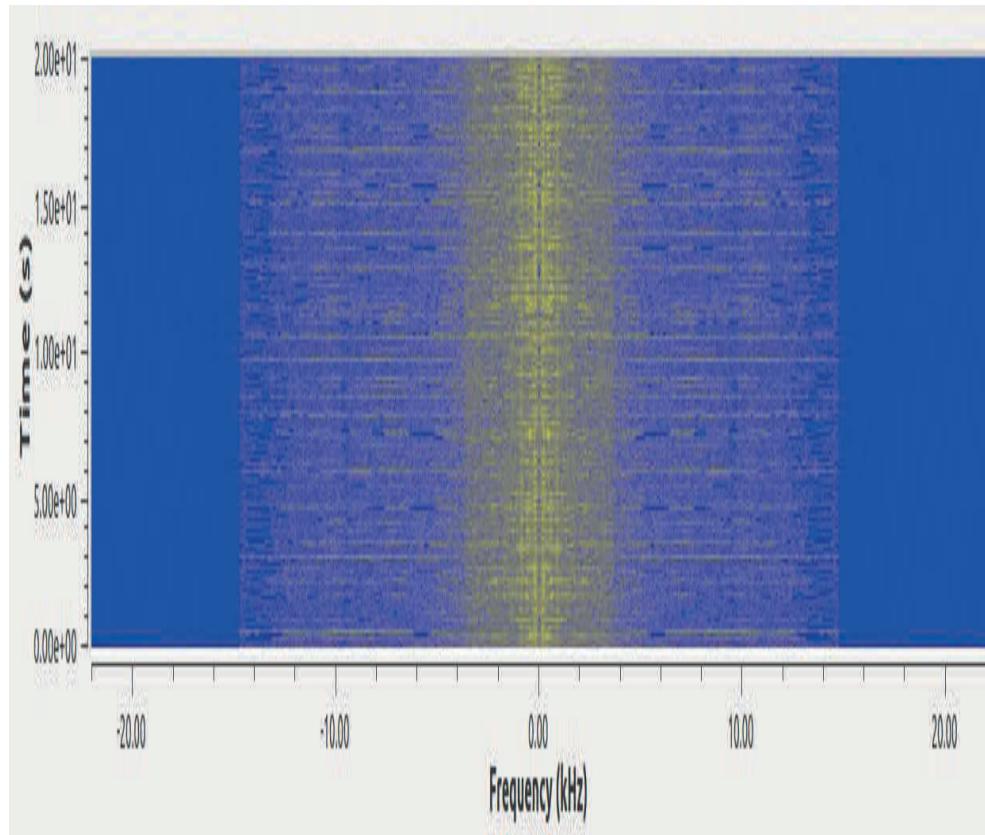


Figure 4.8.5 Waterfall diagram.

Based on the analysis a short comparisons is made between FIR and IIR filter as shown below:

FIR filter	IIR filter
Always make linear phase	Difficult to control and has no particular phase.
Always stable	Might not be stable if feedback received is wrong.
Has no limited cycle	Has limited cycle

Based on the comparison, it can be said that FIR filter has more advantages as compared to IIR filter.

Inference

The motive of this experiment was to have a comparative analysis between the FIR filter and IIR filter. Fig 4.8.3 represents a frequency plot representing the attenuation level done based on the feedback loop, in order to avoid the noise from a given output signals. The feedforward Taps is set as 1default case and feedback Taps is set as 1with an assumption that only 1 delay is applied. Feedforward Tap can never be 0. Fig 4.8.4 provides a time domain representation of IIR filtered output signal. Fig 4.8.5 represents waterfall diagrams which indicate the variation in the types of signals taking place in terms of frequency. It helps in avoiding the aliasing effect in digital signal processing.

EXPERIMENT 9

Aim

The purpose of this experiment is to explore the Root cosine filter and analyze its effects on the input signals using Gnu Radio.

Introduction

A root-raised-cosine filter (RRC) is used in the transmitter and receiver side in a digital wireless communication system to perform matched filtering. Hence it helps in minimizing intersymbol interference (ISI). The combined response of two such filters is equal to the raised-cosine filter. It is a type of filter used for pulse shaping the output signals in a digital communication system. The roost raised cosine filter is defined as a spectrum which has an odd symmetry approximately $\frac{1}{2T}$.

Where T is symbol period for communication system.

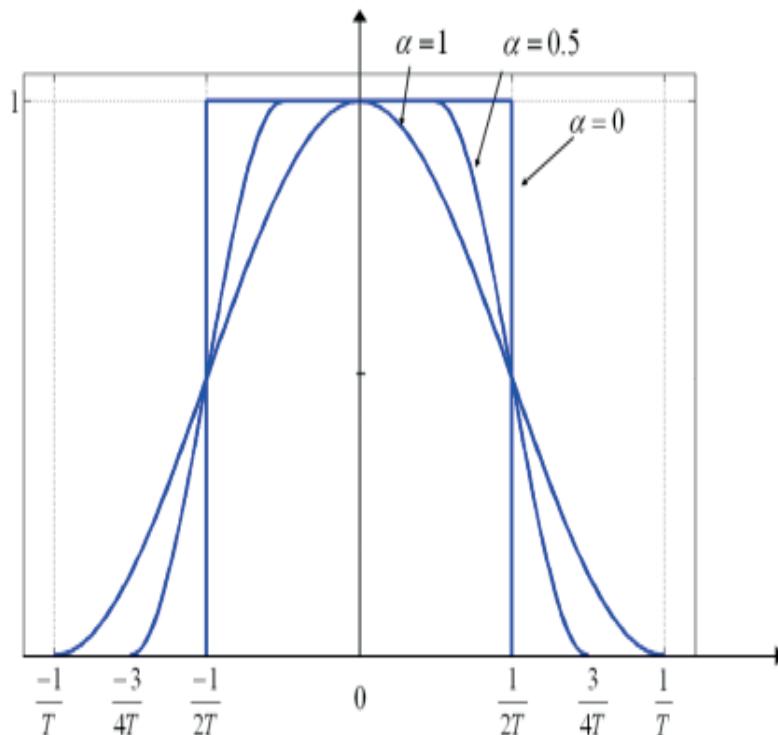


Figure 4.9.1 Frequency response of root raised cosine filter with different roll off factor.

Roll-off factor denoted as α defines the steepness of a transmitted signals with frequency. It is a measure of the excess bandwidth of the filter denoted by F . Hence roll off factor is mathematically expressed as

$$\alpha = 2*T*F$$

Where T defines the symbol rate. The value of α lies within the range $[0, 1]$. From the figure above it can be seen that with $\alpha = 0$, the ideal low pass filtered output signal is obtained and with increase in α value near to 1, the carrier signals start spreading .Hence the Bandwidth also increases which result in intersymbol interference (ISI) error .Therefore roll off factor α should be near to 0.

Gnu Radio Flow graphs

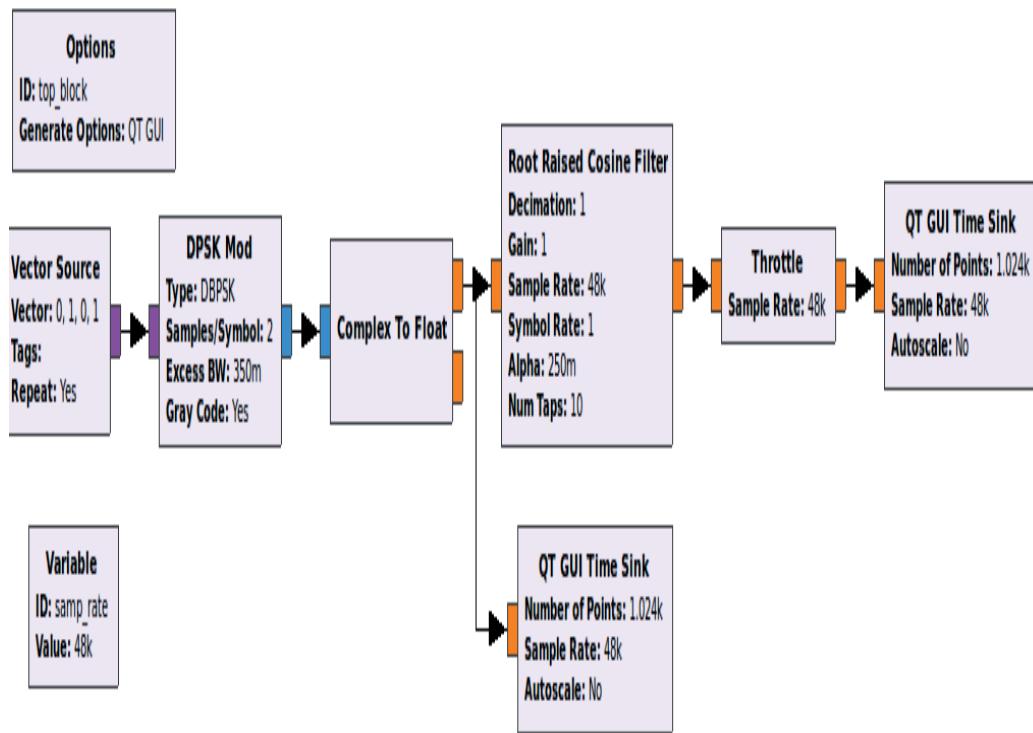


Figure 4.9.2 Root Raised Cosine Filter flow graph.

The vector source is used in order to generate the vector signals with the sequence of $[0, 1]$ in a repeating mode. The Differential BPSK modulation scheme is used with 2 as samples per symbol. Any modulation scheme can be used as per the experimental

requirement. Description about the DPSK modulation scheme is explained in digital modulation chapter. The modulated signal is then passed through the root raised cosine filter. As per the theory, the roll off factor is considered as 0.25 with number of taps as 10, which defines the number of time the signals is filtered in order to receive the desired pulse shaped output.

Results

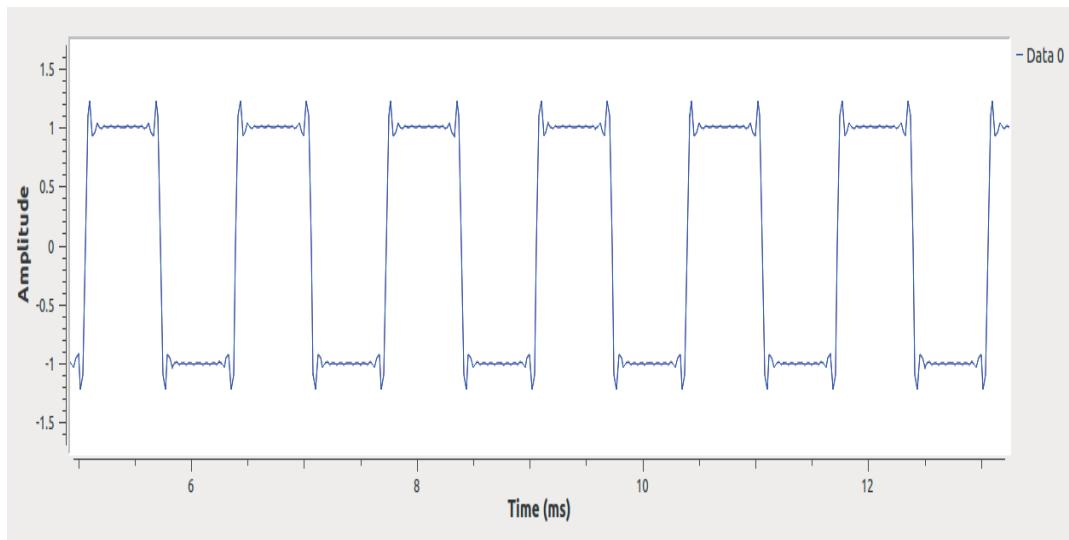


Figure 4.9.3 the input vectored signals.

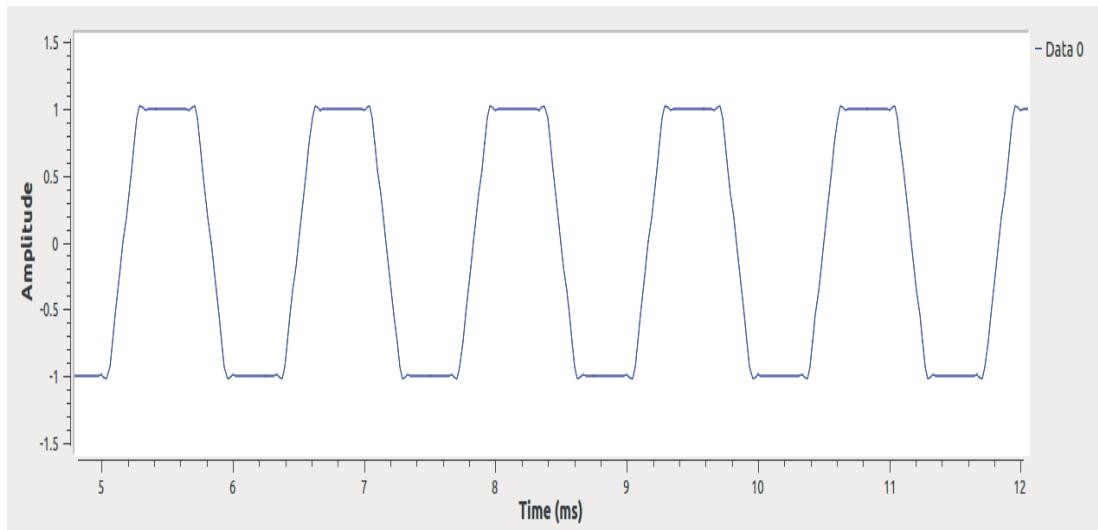


Figure 4.9.4 Pulse shaped output signals.

Inference

The purpose of this experiment was to implement the Root cosine filter by pulse shaping output signals. Fig 4.9.3 represents the input vectored signals with 0 and 1 in a repeating mode. Fig 4.9.4 is the pulse shaped output with roll off factor $\alpha = 0.25$. The desired signal is filtered with 10 numbers of taps in order to minimize the ISI. Hence it can be concluded that by minimizing the roll off factor near to Zero, one can get the accurate pulse shaped output signal.

EXPERIMENT 10

Aim

The purpose of this experiment is to analyze different windowing techniques with Low Pass Filter using Gnu Radio.

Introduction

Information containing digital signals is sufficiently large enough to analyze the data, because statistical calculations require all points to be available for analysis. Hence in order to avoid these problems, small subsets of the whole dataset need to analyze. This process is called as windowing technique.

Windowing technique is a process of taking a small subset of a larger dataset, for processing and analysis. There are different types of Windowing techniques as mentioned below:

1. Rectangular Windowing
2. Hamming Windowing
3. Hanning Windowing
4. Kaiser Windowing
5. Blackman Windowing

Rectangular Window

The rectangular window, involves truncating of the dataset before and after the window, hence it does not modify the contents of the window at all (i.e. all the data points outside the window are truncated and assumed to be zero).

$$W(n) = \begin{cases} 1, & -\frac{M-1}{2} \leq n \leq \frac{M-1}{2} \\ 0, & \text{Otherwise} \end{cases}$$

Where M is the window length of sample

As M increases, the main lobe becomes narrows (better frequency resolution). M has no effect on the height of the side lobes. Fig 10.1 represents a frequency spectrum of low pass filter with rectangular window. It does represent a very wide bandwidth spectrum. Hence it is considered to be a poor method of windowing techniques as it results in power leakage.

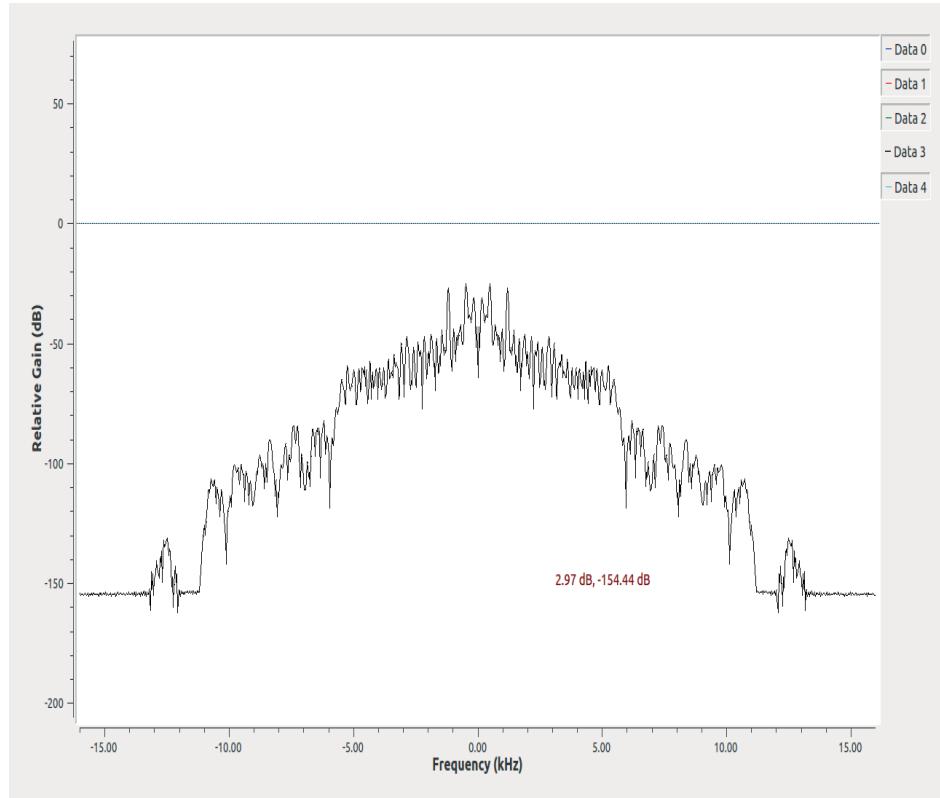


Figure 4.10.1 Rectangular window with low pass filter.

Hamming Window

The disadvantage of rectangular window is that it has wide spectrum which include some side lobes. Hence power leakage takes place. The solution to this problem is the Hamming window. The Hamming window is a taper formed by using a raised cosine with non-zero endpoints, optimized to minimize the nearest side lobe. The Hamming window is mathematically defined as:

$$W(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{M-1}\right) \quad 0 \leq n \leq M-1$$

Where M is defined as the length of window for n number of samples.

It was recommended for smoothing the truncated function in the time domain. It does have relatively narrow frequency spectrum as compared to rectangular window. Still further narrowing can be obtained from Hanning window.

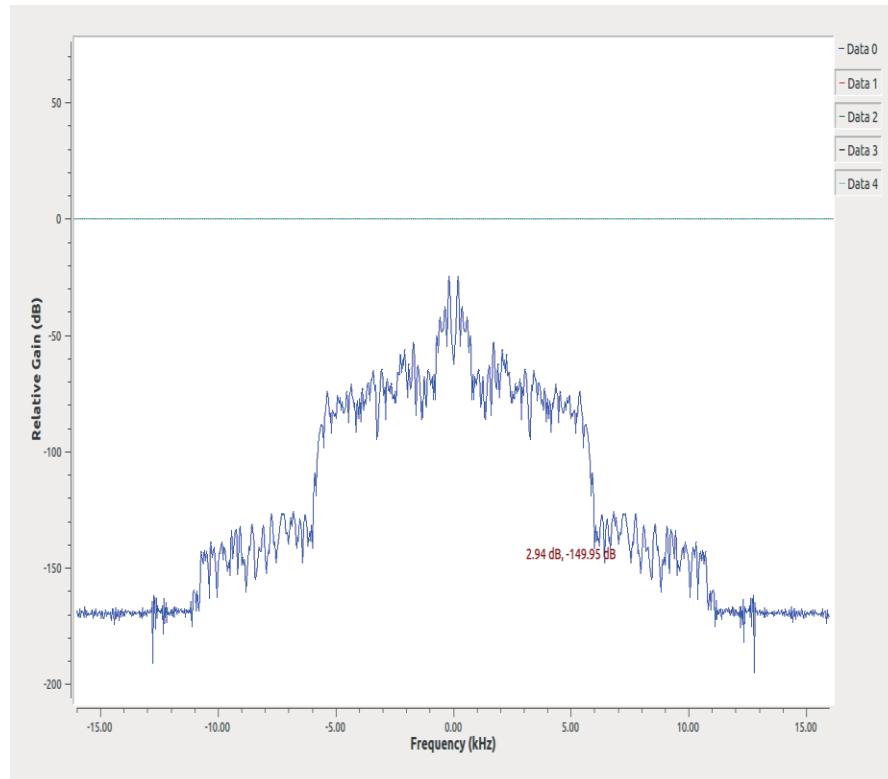


Figure 4.10.2 Hamming Window with low pass filter.

Hanning Window

The Hanning window, invented by Von Hann, has been defined as a linear combination of modulated rectangular windows. It is the shape of one cycle of a cosine wave with 1 added to it so Hanning function is always positive. The sampled signal values are multiplied by the Hanning function. In case of Hanning window the ends lobes are forced to zero regardless of what the input signal is doing.

$$W(n) = 0.5(1 - \cos(2\pi n/N)), \quad 0 \leq n \leq N.$$

Where window length is L = N + 1.

Kaiser Window

The Kaiser window, also known as the Kaiser–Bessel window, was developed by James Kaiser is mathematically expressed as:

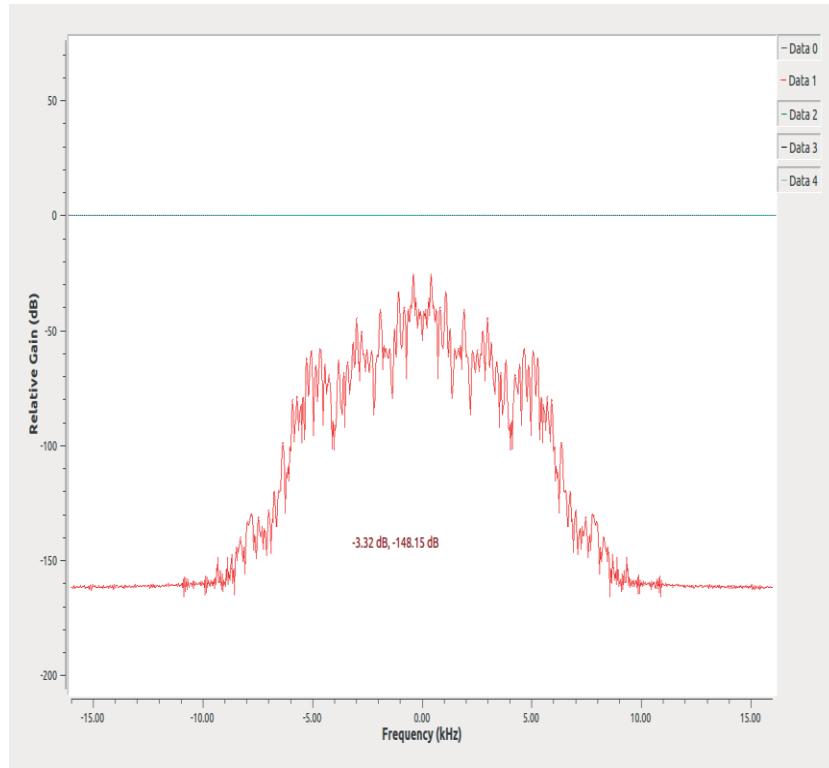


Figure 4.10.3 Hanning Window with low pass filter.

$$W(t) = \begin{cases} \frac{I[\alpha \sqrt{1 - (\frac{t}{r})^2}]}{I[\alpha]}, & |t| \leq r, \\ 0, & |t| > r, \end{cases}$$

Where

1. I is the zeroth-order modified Bessel function of the first kind,
2. r is the window duration,
3. α is a non-negative real number that determines the shape of the window.

It is used for modified discrete transform. The Kaiser window is used in the Advanced Audio Coding digital audio format. Fig. 10.4 represents a frequency spectrum of Kaiser Window techniques which seems to have narrow main lob with reduction in side lob.

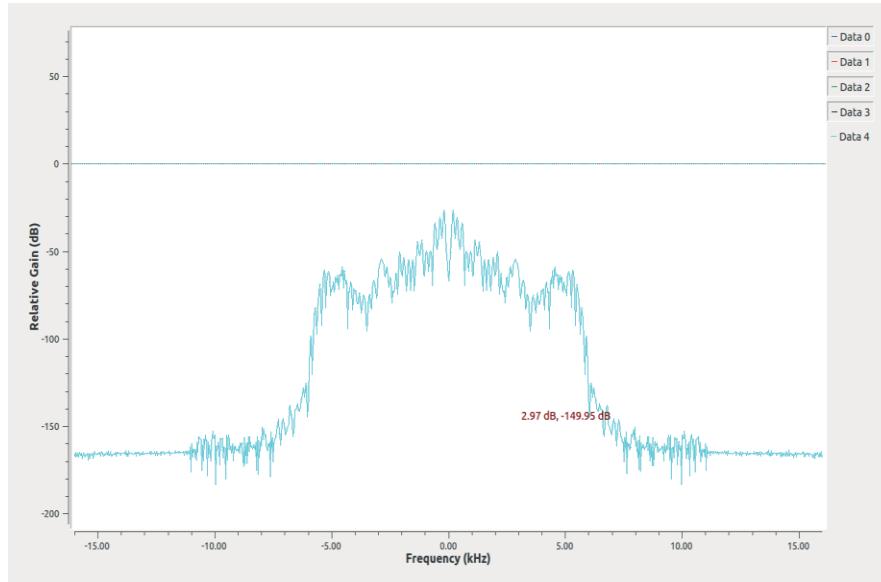


Figure 4.10.4 Kaiser Window with low pass filter.

Blackman Window

The Blackman window is a taper formed by using the first three terms of a summation of cosines. It was designed to have close to the minimal leakage possible. The Blackman Window is mathematically expressed as:

$$w[k+1] = 0.42 - 0.5 \cos\left(2\pi \frac{k}{n-1}\right) + 0.08 \cos\left(4\pi \frac{k}{n-1}\right), \quad k = 0, \dots, n$$

From the equation above we can find that the Blackman window contains an extra cosine term; hence it provides a minimum leakage output with reduced in side lobes as compared to other windowing techniques. But the performances of Blackman window are slightly worst as compared to Kaiser Windowing techniques. In window sequence function, as the Blackman window has more terms which indicates that more accuracy in the calculation of results. For this reason, the Result and the simulation both are more accurate in Blackman window.

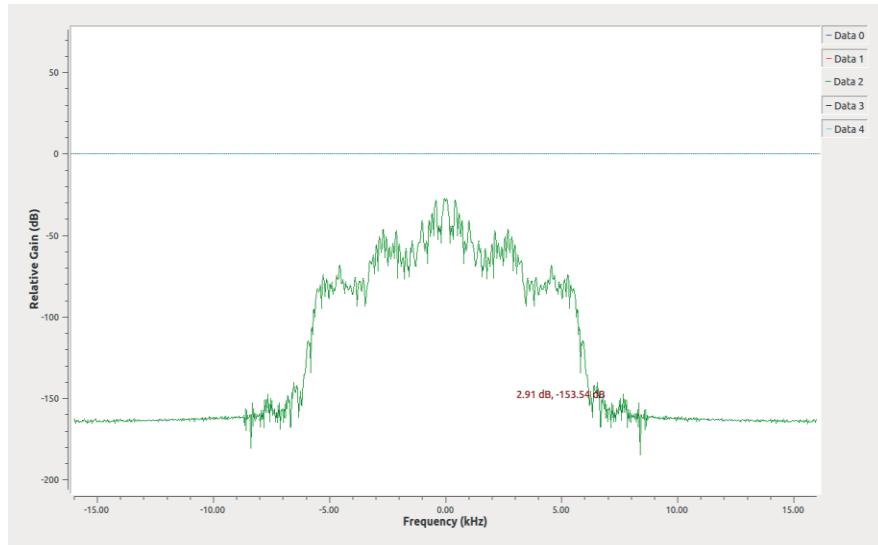


Figure 4.10.5 Blackman Window with low pass filter.

Gnu Radio Flow graphs

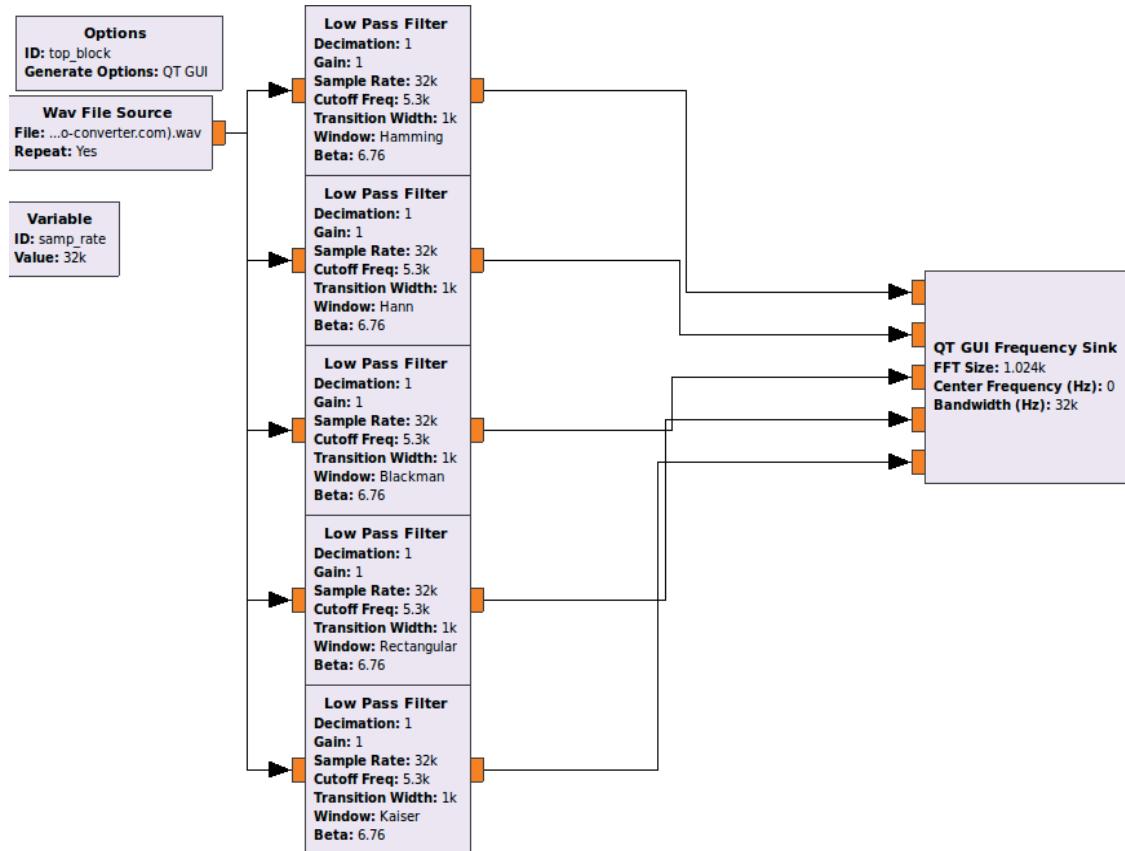


Figure 4.10.6 Low pass filter with different window techniques.

The sample rate of 32 K is considered (It can change as per the experimental requirement). The output signals from Wav File Source is then allowed to pass through the Low pass filter block having cut off frequency 5.3 KHz and a transition width of 1000. Transition width is basically used to Control the steepness in the attenuation of the signal above the cut off frequency. Multiple low pass filters are used with different window techniques. The Blackman window is used as a windowing technique.

Results

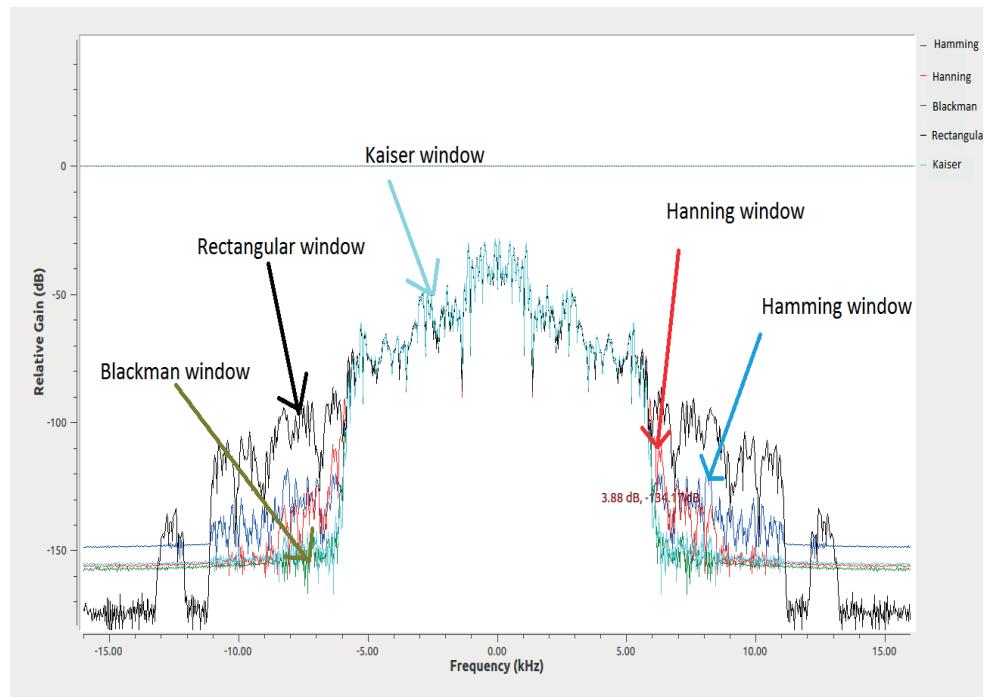


Figure 4.10.7 Comparisons of different windowing techniques.

Inference

The purpose of this experiment was to implement different types of Windowing techniques using Low pass filter of an audio signals through Gnu Radio platform. Fig 4.10.7 represents the comparative analysis of different windowing techniques .Based on the comparisons it can be concluded that the Blackman window is found to be better, as it results in the output signal having a narrow main lob with reduction in the side lobes. Thus it results in minimum leakage as compared to other windowing techniques.

Chapter 5

Analog Modulation

This chapter deals with the introduction and implementation of Analog modulation techniques with a series of experiments. Modulation deals with the process of transferring an analog baseband signal to a high frequency radio signals. There are two ways to modulate analog signals.

1. Amplitude modulation techniques.
2. Frequency modulation techniques.

Table 9: List of Analog Modulation experiment

Sr No	Description
1	Implement Amplitude modulation using cosine wave using USRP B100
2	Implement the Transmission and reception of audio signal using Amplitude modulation
3	Implement the Frequency modulation techniques with audio signals using USRP B100 3. A) Implementation of FM using recorded audio signals. 3. B) Implementation of FM receiver which receives a real time signals from FM Station.

EXPERIMENT 1

Aim

Implement Amplitude modulation using cosine wave in Gnu Radio and validation on USRP B100.

Specification of USRP B100

Table 10: Specification of *USRP B100*

1	Name	USRP B100
2	Interfacing with host	Through USB 2.0
3	Daughterboard	WBX
4	RF data board	50MHz – 2.2 GHz
5	1pps/Ref	Yes
6	Power	6V DC ,3 Amp

Introduction

Amplitude modulation is a type of analog modulation technique which is defined as the change in the amplitude of a carrier signal which does not contain any information, with respect to instantaneous change in the amplitude of information containing modulated signal. Carrier signal is a high frequency signal f_c with A_c as amplitude of a carrier signal. Hence carrier signal is mathematically formulated as:

$$C(t) = A_c \cdot (\cos(2\pi f_c t))$$

Modulated signal is expressed as

$$M(t) = A_m \cdot (\cos(2\pi f_m t))$$

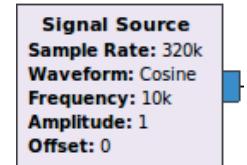
Where A_m = Amplitude modulation of a modulated signal and f_m as a frequency of a modulated signal.

Therefore amplitude modulated signal is defined as

$$S(t) = [A_c + A_m \cos(2\pi f_m t)] \cos(2\pi f_c t)$$

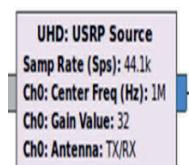
The amount of carrier signal required in the modulation is determined by the modulation index denoted by symbol m . Modulation index is defined as the ratio of amplitude of a modulated signal by the amplitude of a carrier signal. If $m < 1$ then the modulated signal is called as under modulation. If $m = 1$ then the modulated wave is considered to be a 100% perfect modulation. If $m > 1$ then it is termed as an over modulation as it results in the interferences of signals and loss of information takes place. Hence the modulation index should be always ≤ 1 . Therefore in this experiments, modulation is considered to have a perfect modulation ($m=1$).

Block explanation of the flow graph



1. Signal Source

Signal source is used to generate the cosine or sine wave carrier signal with Sample rate of 320 KHz having frequency of 10 KHz. (Carrier signals can be selected based on your requirement).



2. UHD: USRP Source

UHD stands for USRP hardware driver software which specifies various properties as explained. The sample rate defines the rate of baseband samples between the System

and USRP B100. Centre frequency is tuned to 1 MHz in order to have a perfect transmission and reception. The input data to the UHD USRP Source can be a float or a complex data type depending on the data streams. The Gain of the system is taken to be 32 max for the better efficiency of a system.



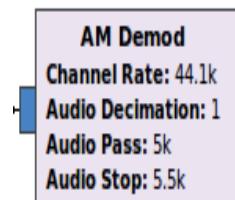
3. UHD: USRP Sink

Input Type controls the data type of the stream in gnu radio. Wire Format controls the form of the data over the bus/network. Num Motherboards selects the number of USRP motherboards in this device configuration. Reference Source is used in order to sync the motherboard with respect to time and clock references.



4. Add Const

Add Const is set to 1 in order to have a 100% perfect amplitude techniques.



5. AM Demod

While demodulating, two steps is needed which are:

1. Creating a baseband signal.
2. Applying the filter in order to remove the unwanted high frequency signal.

Channel Rate is the incoming sample rate of the AM baseband with the audio decimation rate of 1. Audio pass and Audio stop are the low passband frequency filter and the low stop band frequency filter which is by default set to 5 KHz and 5.5 KHz.

```

QT GUI Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 320k
Update Rate: 10

```

6. QT GUI Sink

Graphics user interface with standard QT provides a complete representation of signals in terms of time domain, frequency domain, waterfall display, constellation display. Centre frequency can be tuned in order to have a perfect transmission and Reception of signals with a Bandwidth considered as 320 KHz (It can be changed as per the requirement).

Gnu Radio Flow graphs

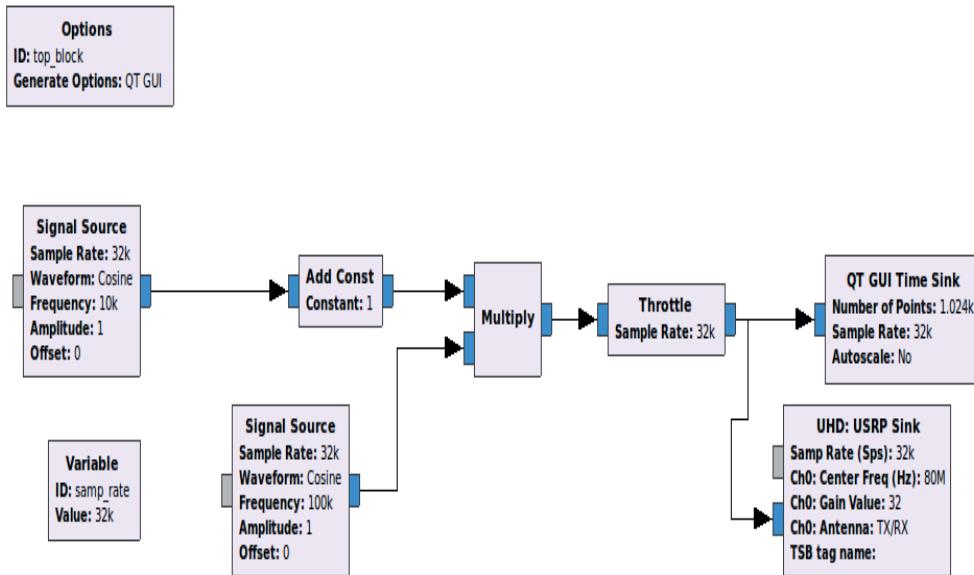


Figure 5.1.1 Transmitter side (AMPLITUDE MODULATION).

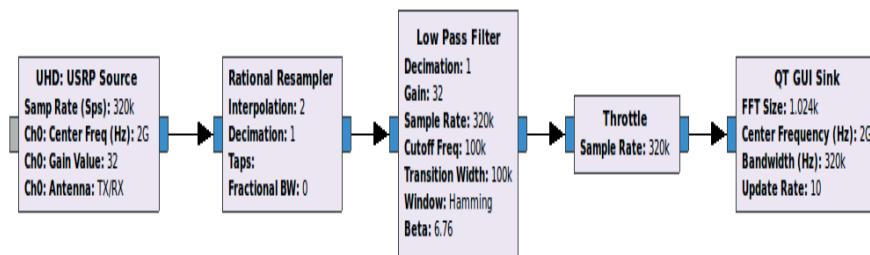


Figure 5.1.2 Receiver side (AMPLITUDE DEMODULATION).

NOTE

In this experiment single USRP B100 is use for both transmission and reception, hence Fig. 5.1.3 represents a loop back system.

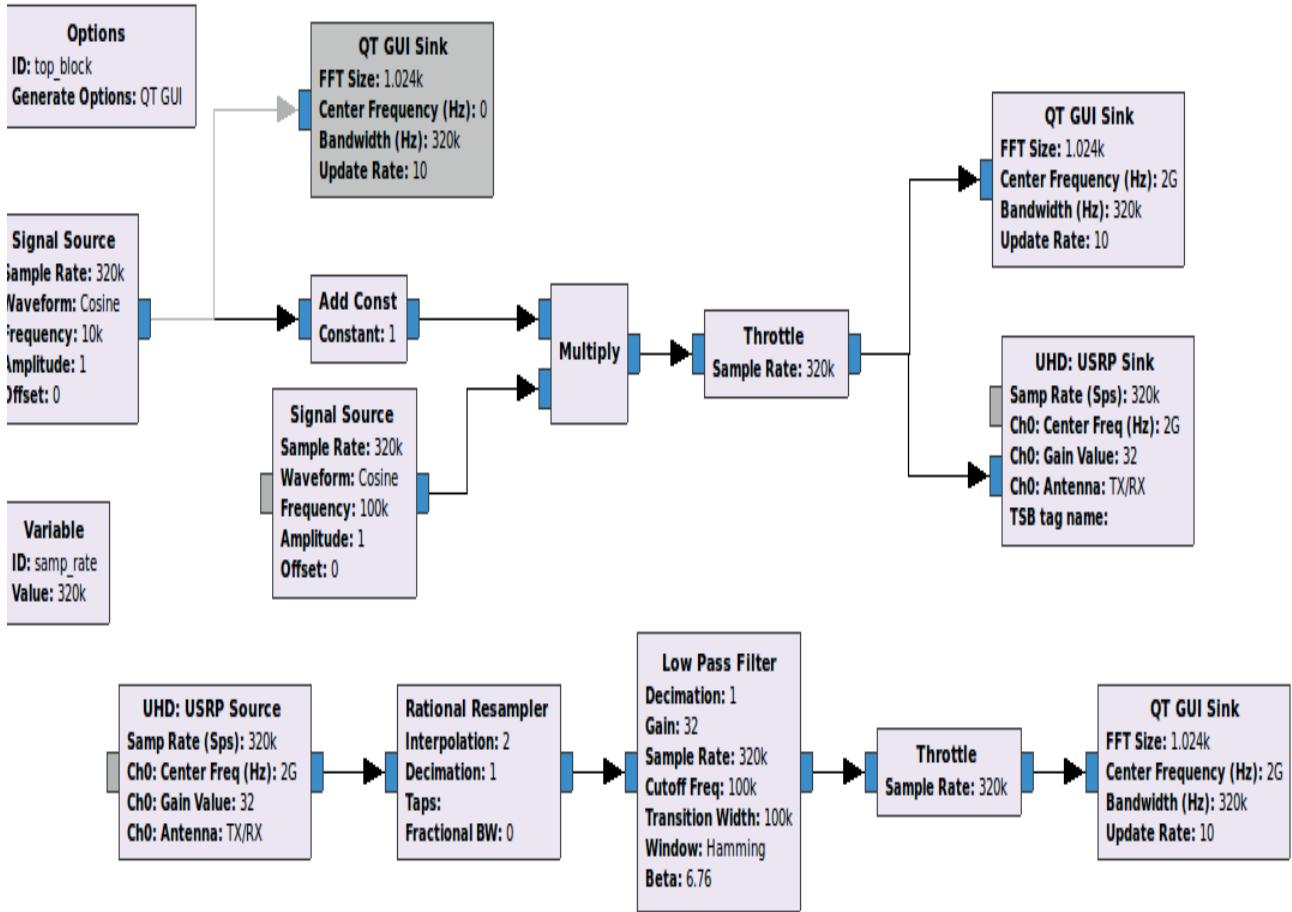


Figure 5.1.3 Amplitude modulation flow graph using USRP B100

The overall amplitude of the carrier signal is modulated with the message signal. Hence the carrier signals changes in line with respect to the modulating signal. The carrier signal is the high frequency signals therefore it has to be greater than the message signal. In the flow graph mentioned above, two signals of frequency $f_1=10\text{kHz}$ and $f_2=100\text{kHz}$ are considered with the sampling rate of 320k, where f_1 is the message signal and f_2 is the carrier signal. The ADD const is made as 1 in order to have a 100 % perfect modulation technique. Hence no loss of information takes place.

The host system is interfaced with USRP B100 using USB port 2.0. USB 2.0 transfers data at 480 Mb/sec speed with 500mA power. Rational resampler provides a combination of interpolation and decimation which is basically used to either increase or decrease the sampling rate.

Results

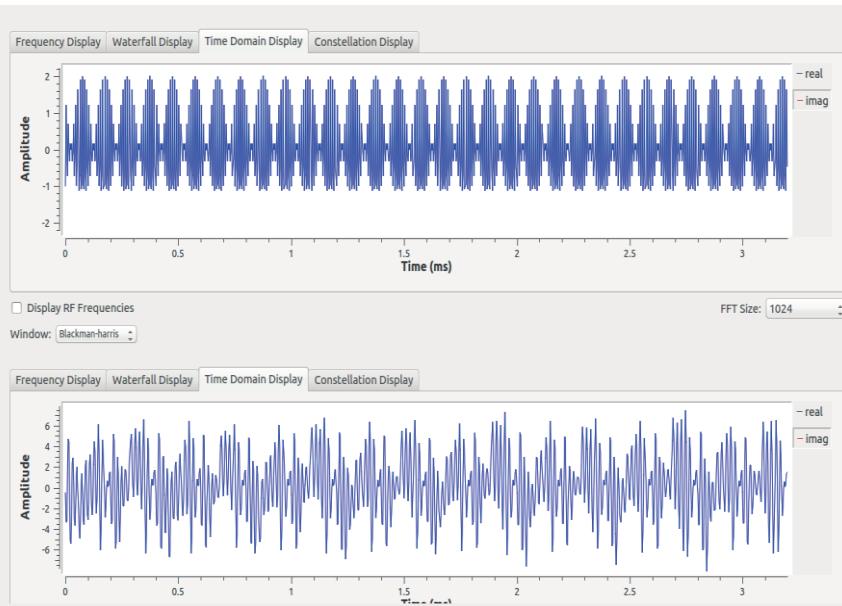


Figure 5.1.4 Transmission and Reception of cosine wave using USRP B100.

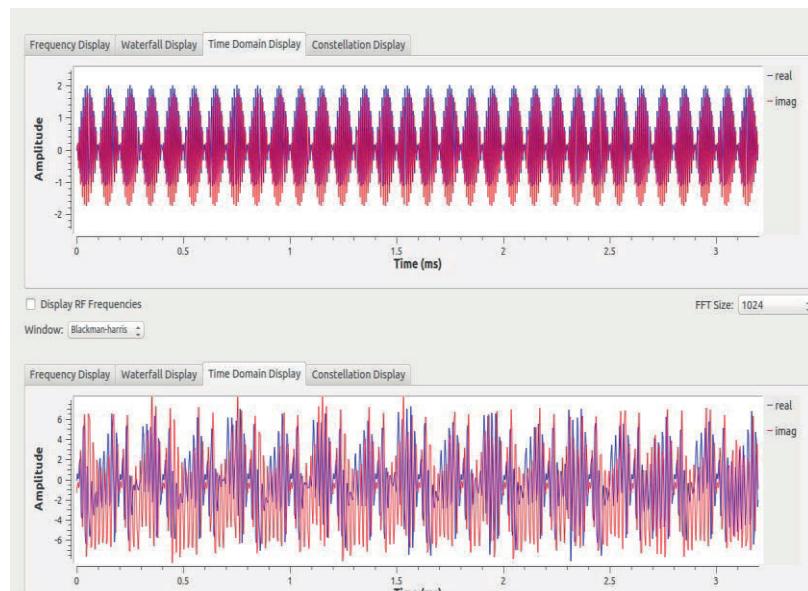


Figure 5.1.5 Complex representation of cosine wave using Amplitude Modulation.

Once the signal is resampled, then the low pass filter is used to filter out the high frequency signals thus allowing only low frequency signals through it. The center frequency f_c is tuned to 2 GHz in order to have a proper transmission and reception of signals. Fig 5.1.4 represents a transmission and reception of cosine wave signal considering only real terms value whereas fig 5.1.5 represent the complex transmission and reception of cosine wave signal. The signals are found to be distorted due to the conductive medium but it can be avoided with proper selection of noise cancellation techniques.

Inference

The purpose of this experiment was to analyze the Amplitude modulation techniques by transmitting and receiving the cosine wave signals through the WBX Trans receiver USRP B100 system having an RF signal ranging from 50 MHz to 2.2 GHz. The single USRP B100 is used for both transmission and reception, hence loop back system are created. The message signal is received which is not exactly same but identical to the original signal. This is due to the loss of signals occurring, which can be avoided with higher modulation techniques and also with the proper selection of noise cancelation techniques.

EXPERIMENT 2

Aim

To implement the Amplitude modulation techniques using voice signal in Gnu Radio and validation on USRP B100.

Experimental Setup

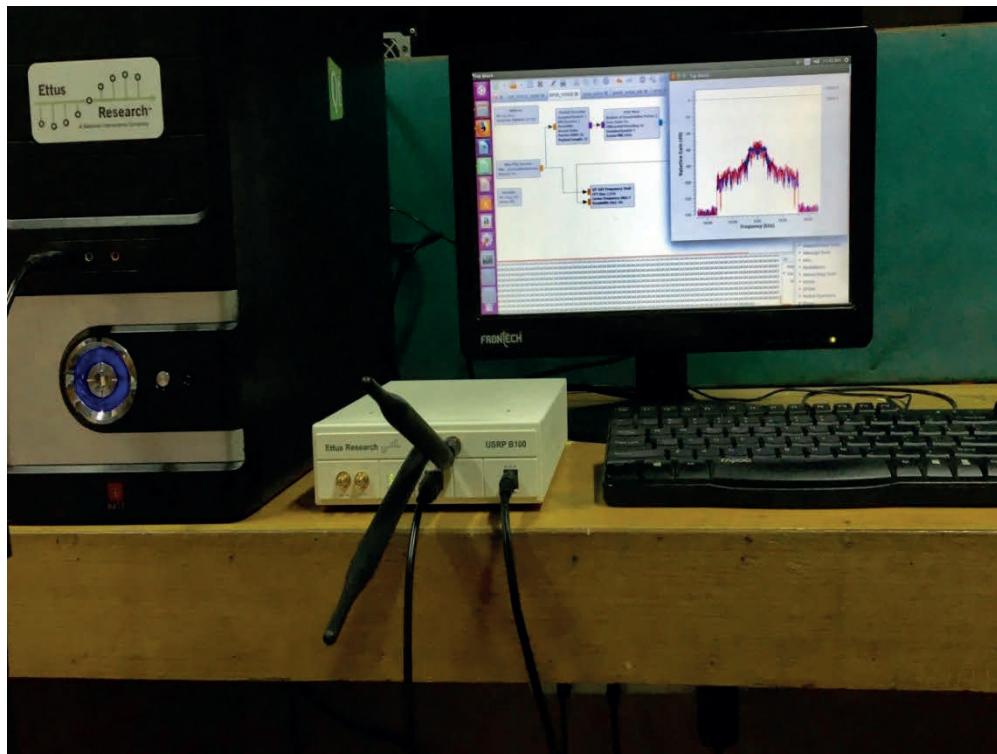


Figure 5.2.1 Experimental Setup

Hardware setup required:

1. System with Ubuntu as an operating system and Gnu radio installed.
2. USRP B100.
3. RF Daughterboard WBX = 50MHz to 2.2 GHz.
4. Interfacing system with USRP B100 using USB 2.0
5. Power adapter supporting 6V DC, 6 Amp.
6. VERT900 omnidirectional antennas.

Gnu Radio Flow graphs

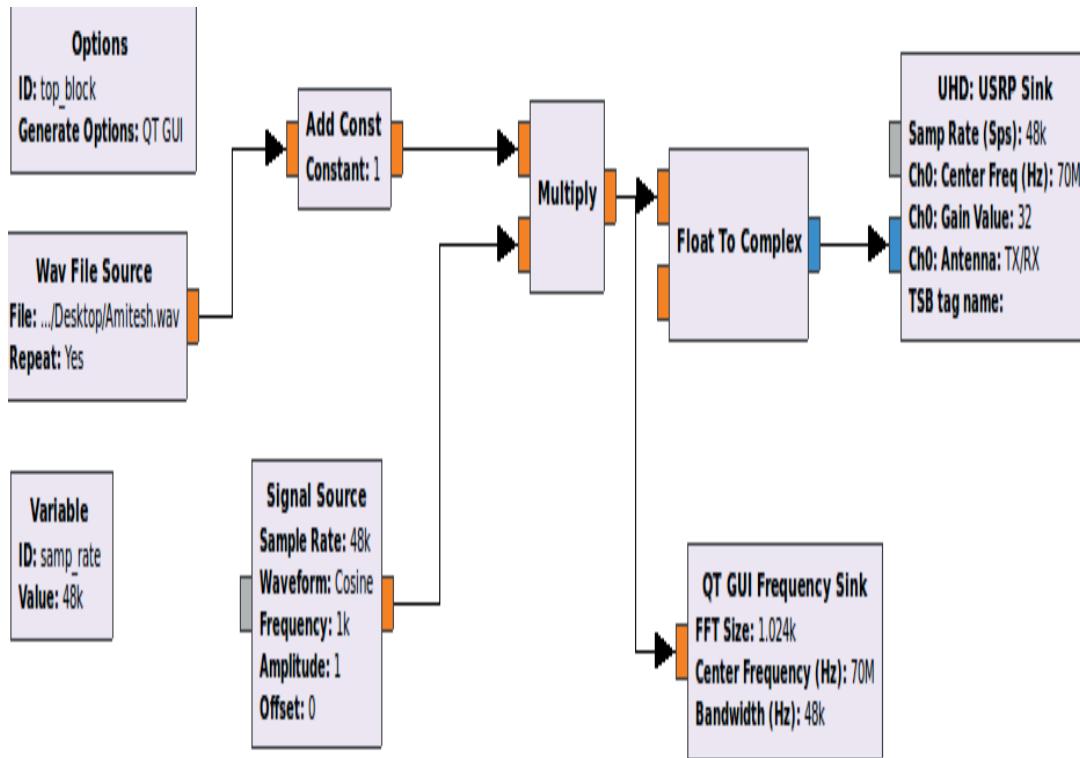


Figure 5.2.2 Transmitter side (AMPLITUDE MODULATION).

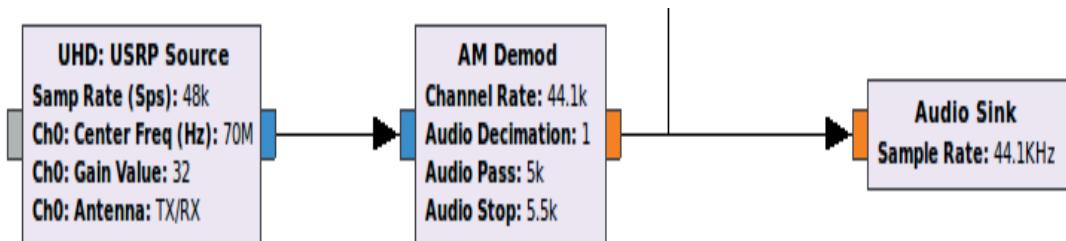


Figure 5.2.3 Receiver side (AMPLITUDE DEMODULATION).

NOTE:

In this experiment single USRP B100 is use for both transmission and reception, hence Fig. 5.2.4 represents a loop back system.

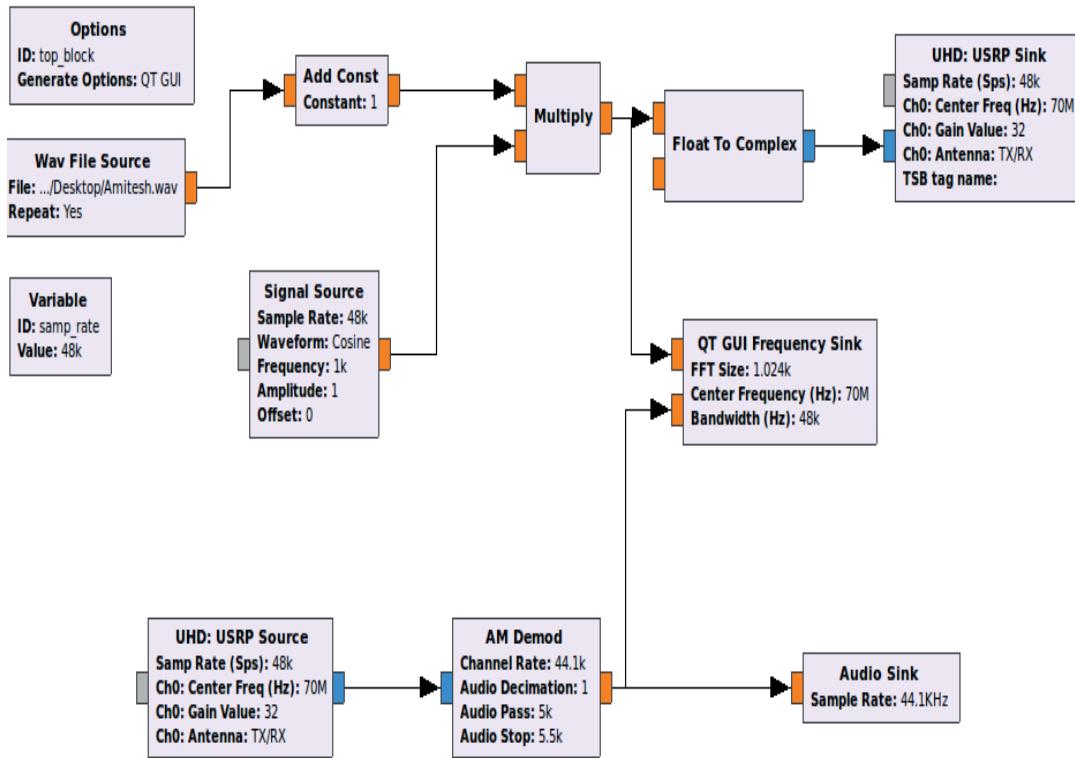


Figure 5.2.4 Amplitude modulation flow graph using USRP B100.

Results

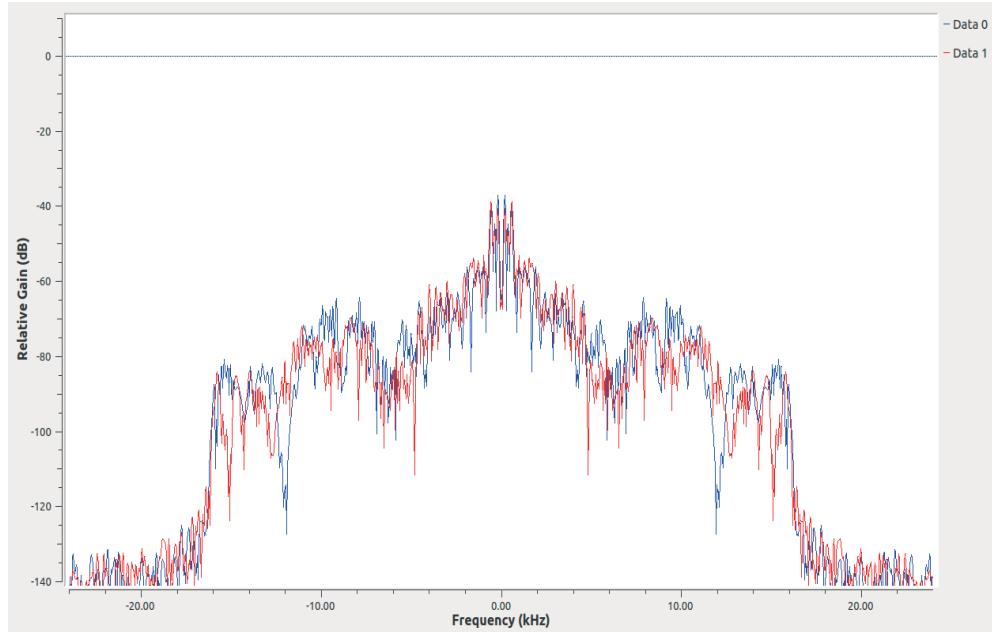


Figure 5.2.5 Transmission and Reception of voice signal using Amplitude modulation

The overall amplitude of the carrier signal is modulated with audio signal. In the flow graph mentioned above, one carrier signals of frequency $f=1k$ is considered with the sampling rate of 48k. The information contain audio signal is multiplied with the cosine carrier signal. The host system is interfaced with USRP B100 using USB port 2.0. The received signal is allowed to pass through the AM Demod block, which will attenuate the signals with respect to pass band and stop band.

Inference

This experiment assist in analyzing the transmission and reception of voice signals using Amplitude modulation techniques through the WBX Trans receiver USRP B100 system having an RF signal ranging from 50 MHz to 2.2 GHz. The voice signal received contains audio signals with some amount of noise which can be avoided with higher modulation techniques and also with the proper selection of noise cancelation techniques. Fig 5.2.5 represents the transmission and reception of audio signals centered at 0 KHz frequency. Red color indicates the received signal and blue color indicate the transmitted signal.

EXPERIMENT 3

Aim

To implement the Frequency modulation techniques using voice signal in Gnu Radio and validation on USRP B100.

Introduction

Frequency modulation is a type of radio modulation techniques where information is added to the carrier signals and changed by varying the frequency of a transmitted signal. Information is delivered in forms of voice or music. It is commonly used for radio signals greater than 30 MHz. In general, frequency modulation (FM) is defined as modulation techniques where the information containing signals are encoded in a carrier wave by varying the instantaneous frequency of the wave. As per UK Broadband based, frequencies are tuned from 87.5MHz to 108 MHz.

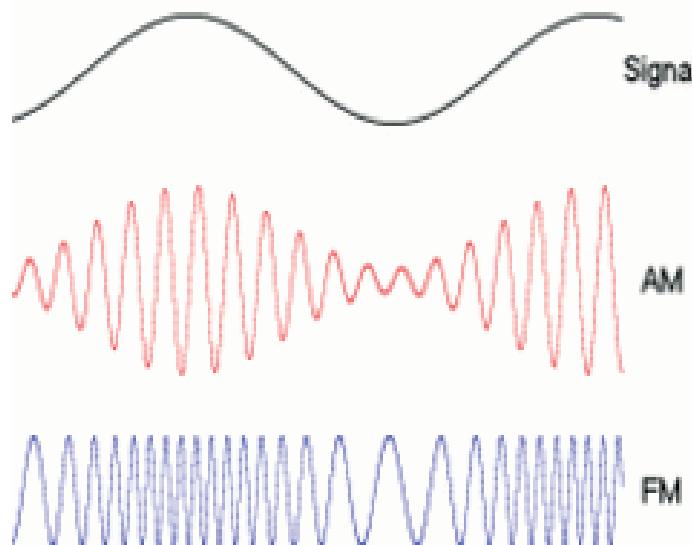


Figure 5.3.1. Waveform of AM & FM techniques.

FM is mathematically expressed as given bellow:

$$Y(t) = A_c \cos(2\pi f_c t + \frac{A_m f_\Delta}{f_m} \sin(2\pi f_m t))$$

Where f_m = baseband modulated signal

A_m = Amplitude of modulated sinusoidal signal

f_Δ = peak frequency deviation

F_c = carrier base frequency

A_C = carrier amplitude

Modulation index defines the amount of modulated variable varies around its unmodulated level. It is mathematically expressed as

$$h = \frac{\Delta f}{f_m}$$

Advantages of FM over AM

1. Voice quality of FM is better than AM because AM signals tend to interfere more with the medium.
2. Noise appears more in AM Transmission because it affects the amplitude of the carrier wave, whereas in FM there is no information contained in amplitude.
3. FM is clearer in transmission than AM.
4. Less radiated power.
5. Small geographical interference between neighboring stations.

Disadvantages of FM over AM

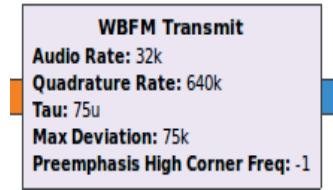
1. AM has short bandwidth, hence more stations can be broadcast in a large area as compared to FM.

NOTE:

This experiment is divided in two sections:

1. FM modulation and demodulation with recorded audio
2. FM Receiver receiving real time signals from FM stations

Block explanation of the flow graph



1. WBPM Transmit

WBPM Transmit performs FM modulation with input audio rate (I.e. input le rate) of 32 KHz and output quadrature rate (I.e. Output sample rate) of 640 KHz. Quadrature rate is a multiple of audio rate. Here quadrature rate is considered to be $20 * 32000 = 640000$.



2. WBPM Receiver

WBPM Receive performs FM demodulation. Quadrature rate here defines the input sample rate that we receive from UHD: USRP Source. Audio Decimation is the attenuation factor needed to decimate the audio signals. Hence the output of WBPM Receive will be $440 \text{ KHz} / 10 = 44 \text{ KHz}$ audio sample rate. The audio decimation rate varies as per the experimental setup.



3. Low Pass Filter

Low pass filter is used to avoid the unwanted high frequency signals and allowing only low frequency signals through it. Decimation is basically used to down sample the incoming signals. Transition Width is used to set the transition width between the pass band and stop band. A small transition width will increase the length of the FIR filter.

Window specifies the window function that will be applied to the FIR filter. Beta parameter is by default set to 6.76 as it is been used for the Kaiser window.

SECTION 3.1:

FM modulation / FM demodulation with recorded audio

Gnu Radio Flow graphs

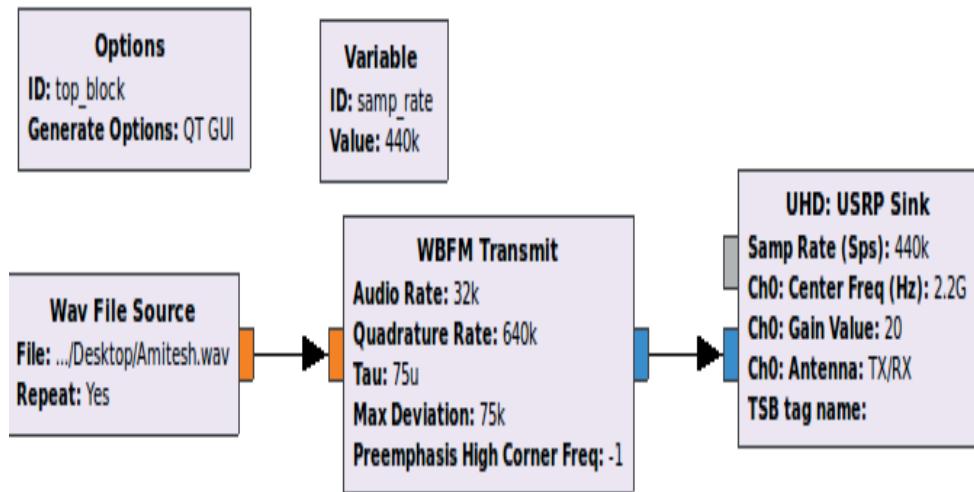


Figure 5.3.1.1 Transmitter side (FREQUENCY MODULATION).

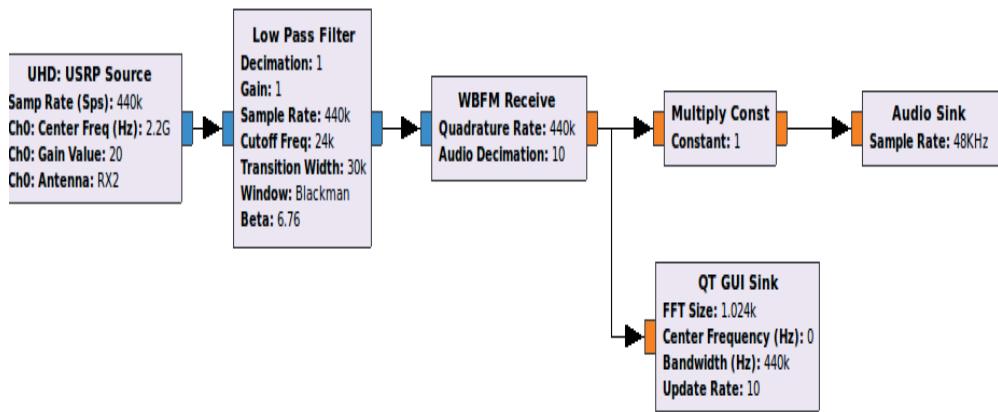


Figure 5.3.1.2 Receiver side (FREQUENCY DEMODULATION).

NOTE:

In this experiment single USRP B100 is use for both transmission and reception, hence Fig. 5.3.1.3 represents a loop back system.

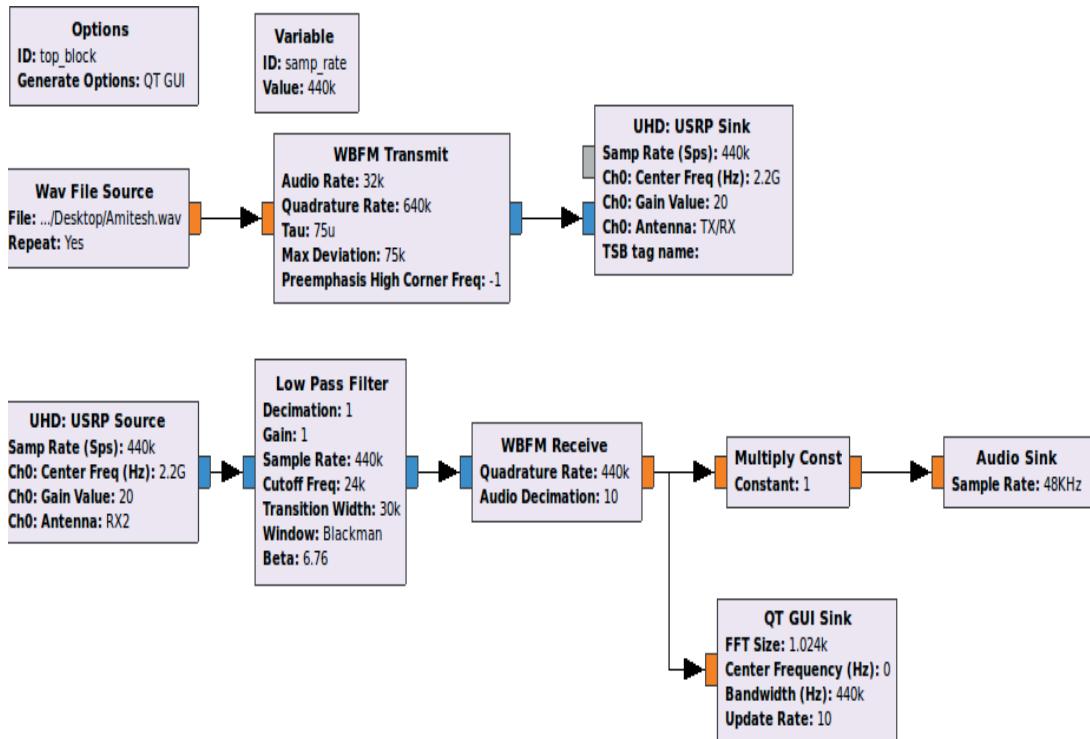


Figure 5.3.1.3 Frequency modulation and demodulation flow graph using USRP B100.

The audio wave signals saved in (. Wav) format is given as an input to WBFM (Wide Bandwidth Frequency Modulation) transmit block. WBFM transmit block takes 32 KHz input audio sample rate and after modulation it delivers 640 KHz output Quadrature sample rate. USRP B100 is used for validation which is set with 20 dB gain value and tuned to Center frequency of 2.2 GHz radio frequency. The gain value changes depending on the output value. For example if the input power is 10mW and the output power needed from the experimental setup is 1W then gain is adjusted in such a way that the system gives out an output power of 1 W. It is mathematically expressed as:

$$\text{GAIN} = \frac{\text{OUTPUT POWER}}{\text{INPUT POWER}} = \frac{1}{10\text{mW}} = \frac{1000}{10} = 100 \text{ Watt}$$

$$= 10 * \log_{10}(102) = 20 \text{ dB.}$$

Results

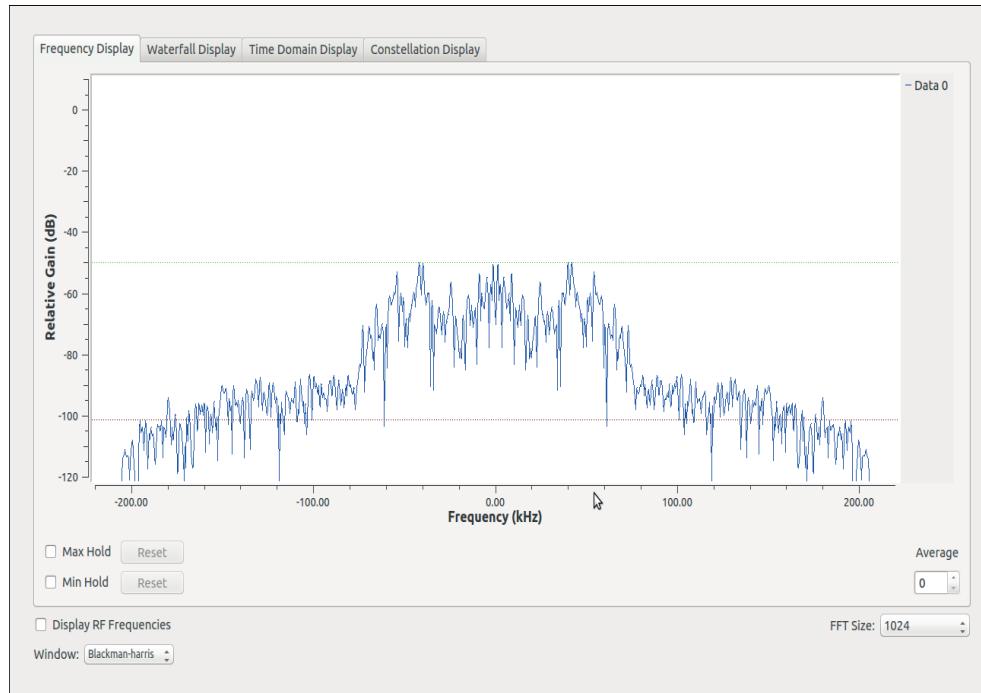


Figure 5.3.1.4. Frequency response of Voice signals using USRP B100.

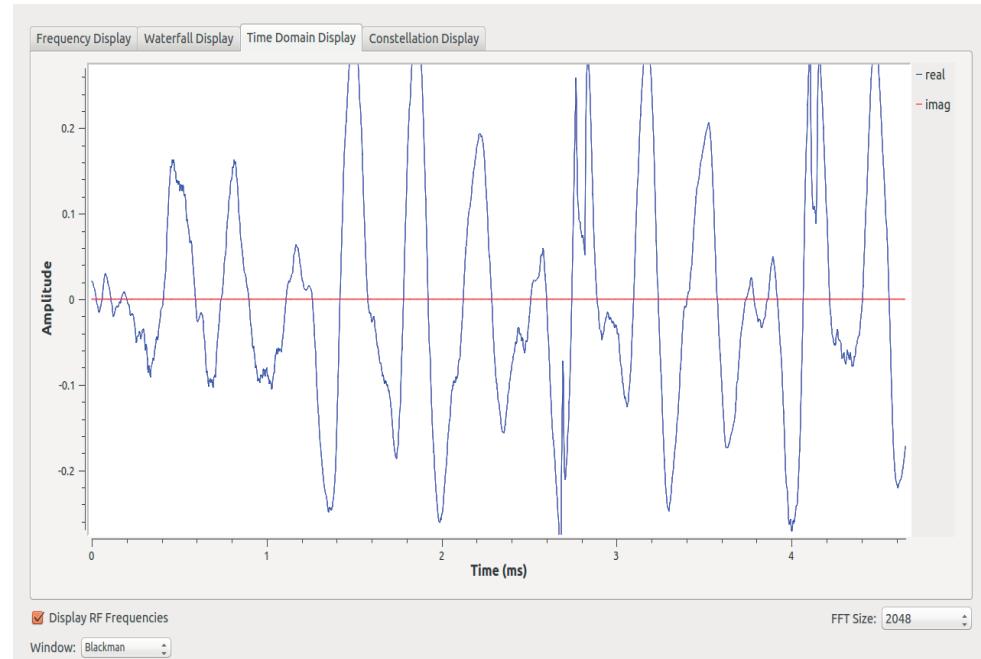


Figure 5.3.1.5. Representation of signals in time domain.

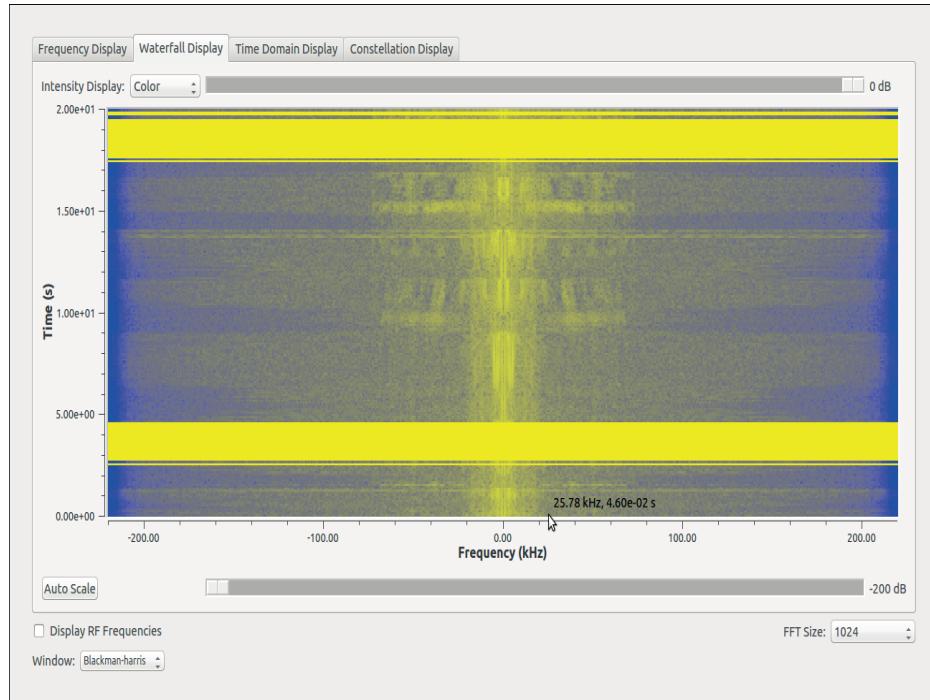


Figure 5.3.1.6. Waterfall diagram.

Inference

The motive behind the experiment was to analyze the frequency modulation and demodulation with voice signals and validate using the WBX Trans receiver USRP B100 system having an RF signals ranging from 50 MHz to 2.2 GHz. Fig 5.3.1.4 represent the output audio signals in a frequency domain. The clarity of audio signal in case of FM is greater than the AM. From fig 5.3.1.5 the variation in the amplitude of signals can be noticed depending on the pitch of voice modulation. Fig 5.3.1.6 represent the waterfall diagram for a given output signal. The dark yellowish colour indicates the variation in the frequency for high pitch and low pitch sound. From waterfall diagram we can understand the minimum and the maximum frequency required for the audio signals. The audio signal lies within [20 KHz to 75 KHz] frequency range. It can vary depending upon the input signals.

SECTION 3.2:

FM Receiver receiving real time signals from FM station

Gnu Radio Flow graphs

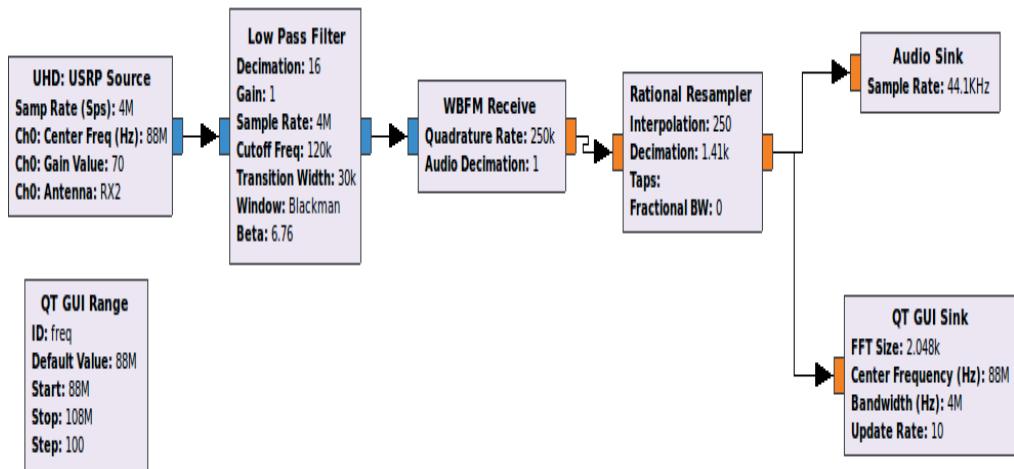


Figure 5.3.2.1 FM Receiver in USRP B100.

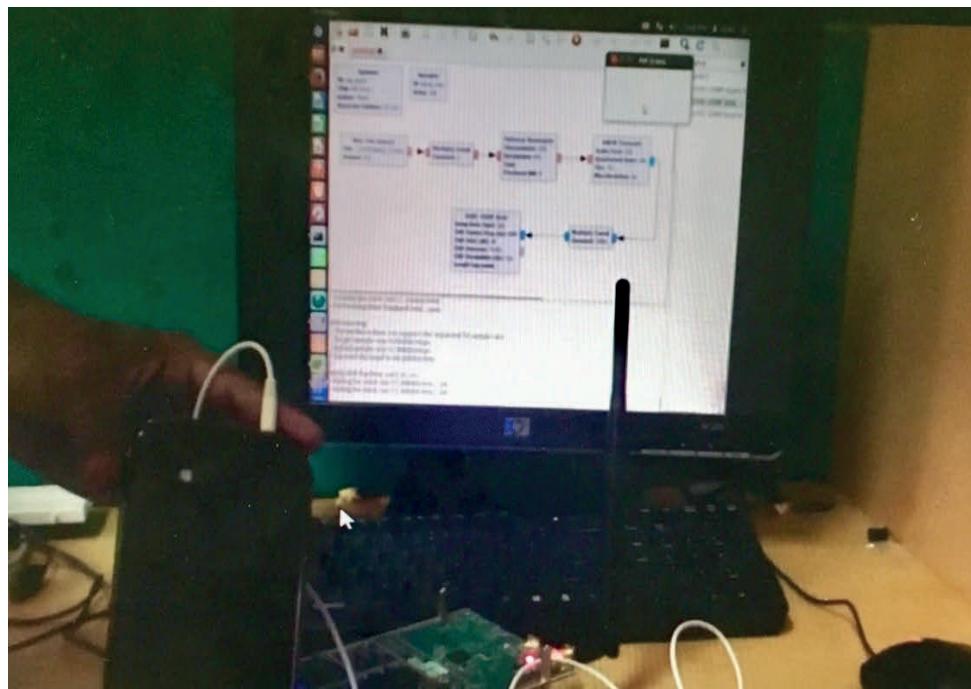


Figure 5.3.2.2 Experimental Hardware Setup.

Results



Figure 5.3.2.3 Frequency spectrum for the received signal.

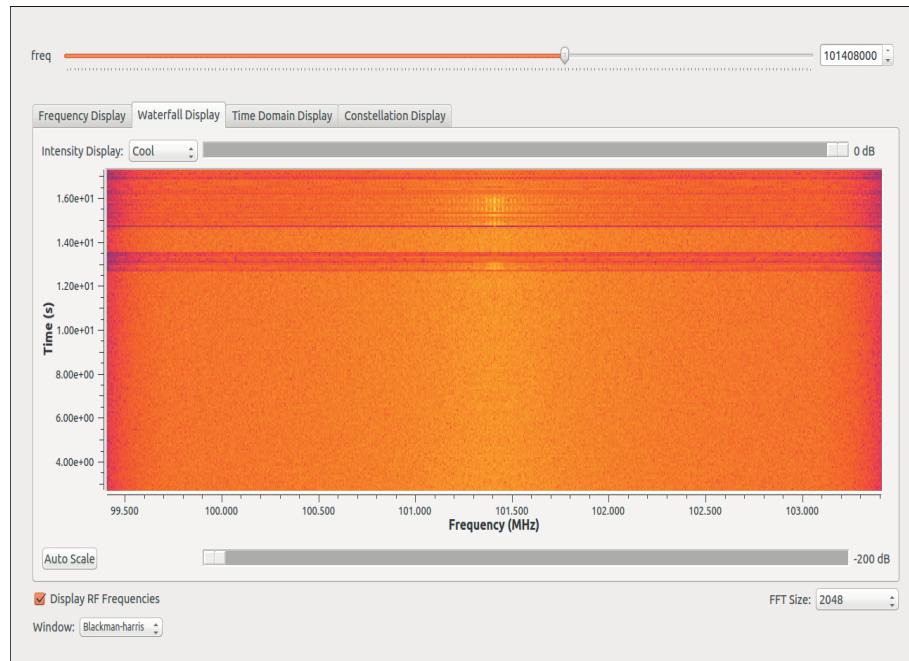


Figure 5.3.2.4 Variation in the waterfall diagram when any FM Station signal is received.

NOTE

Things to be noticed while performing this experiment:

1. The proper positioning of an antenna is needed when receiving signals from FM station.
2. Make sure that the experiment is performed in a network zone area.
3. Tune the FM Station which is currently working or active.

Inference

This experiment aims in receiving the radio FM signals using FM receiver where it catches the real time signals from FM Station. From fig 5.3.2.3, it was observed that the FM station of 96.5 MHz is received with some information delivered. The variation in the frequency range is well observed in fig 5.3.2.4.

Chapter 6

Digital Modulation

Digital Modulation is defined as a process of modulating a carrier signals using discrete values. A digital signal consists of large rectangular pulse signals with 0s and 1s having relatively large infinite bandwidth. Hence in order to transmit such signal, a channel medium is needed with infinite bandwidth, which practically does not exist. Therefore digital modulation is needed as it will transfer a bit of digital streams over the analog channel at high frequency, thus reducing the usage of large bandwidth. In this chapter, the different types of digital modulation techniques are explored with the help of experiments using Gnu radio and validating using Software Define Radio.

Table 11: List of Digital Modulation Experiments

Sr No	Description
1	Implement Amplitude Shift Keying using Gnu radio and validation using USRP B100
2	Implement Phase Shift Keying modulation techniques
3	Implement the Frequency Shift Keying using Gnu radio
4	Implement the Different modulation techniques (BPSK, QPSK, QAM) using Gnu radio and validate using USRP B100.

EXPERIMENT 1

Aim

To implement Amplitude shift keying (ASK) using Gnu radio and validation using USRP B100.

Introduction

ASK (Amplitude Shift Keying) is a type of amplitude modulation wherein the amplitude of a carrier signals changes with change in the digital bits. Let us consider bit symbol transmitted is 1 then the carrier signals of fixed amplitude can be obtained. If bit symbol transmitted is 0, then it represents that there is no informative carrier signals.

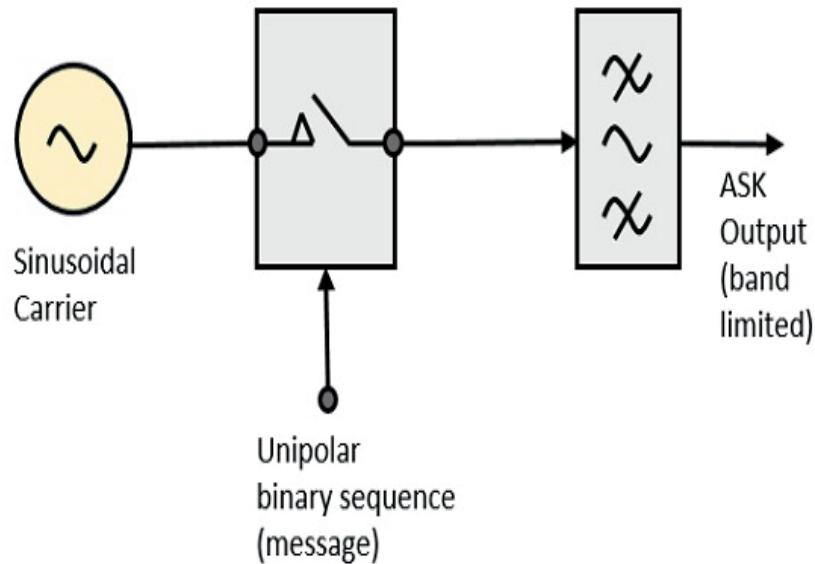


Figure 6.1.1 ASK modulation techniques

ASK modulation comprises of a sinusoidal carrier signal which is converted to unipolar binary sequence signal with amplitude range from 0 to 1. The carrier signals represents a continuous high frequency signals with respect to time. Further the signals is allowed to pass through the bandlimited filters in order to shape the output signals based on the

amplitude and phase characteristics of the filter. The fig 6.1.2 represents the asynchronous way of ASK demodulation techniques. The received signals is allowed to pass through the Rectifier in order to reconvert back the signal to Unipolar binary sequence with amplitude ranging from 0 to 1.

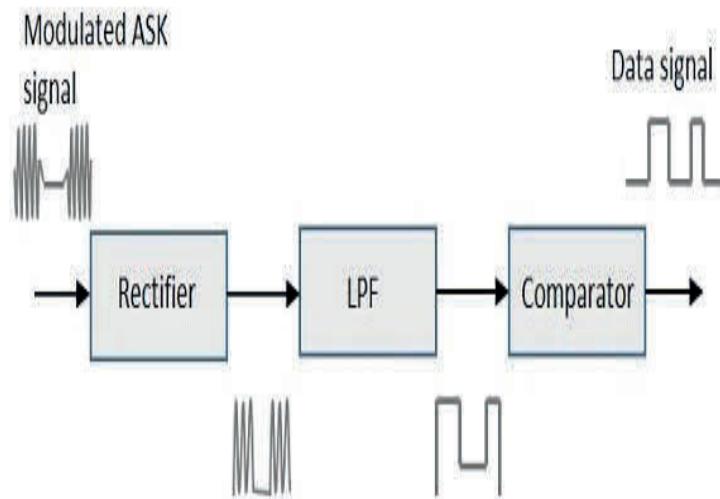


Figure 6.1.2. ASK Demodulation techniques

The signals are further passed through the bandlimited filter in order to limit the frequency signals within the Bandwidth defined. The low pass filter is used which will compress the high frequency signals thus allowing only low frequency signals. The demodulated signal is finally received from the comparator.

Advantages of ASK

1. High Bandwidth efficiency
2. ASK modulation and demodulation techniques are inexpensive with simple receiver architecture.

Disadvantages of ASK

1. Low power efficiency
2. Easily effected by the Noise signals

Application of ASK

1. It is used in the telegraph worldwide
2. It is used in the fiber optical data communication system

Gnu radio Flow graph

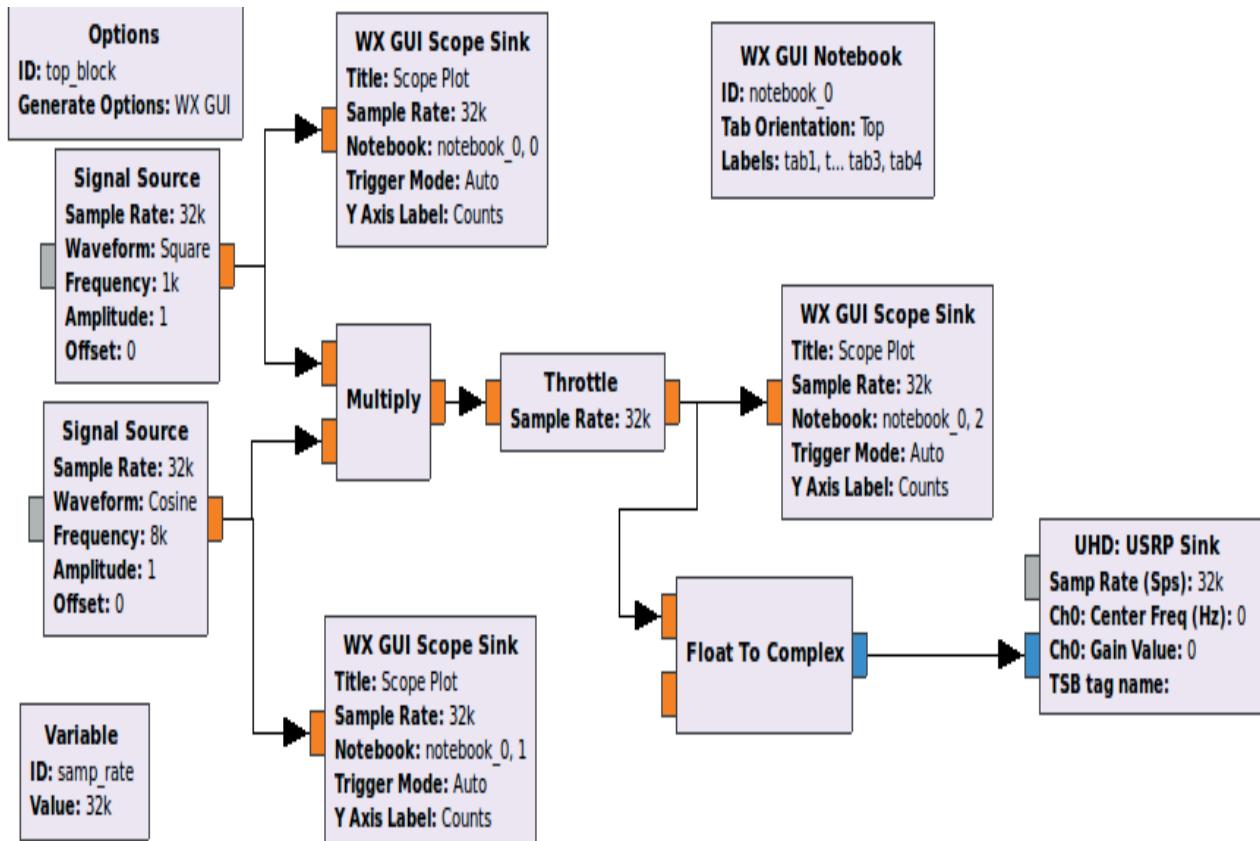


Figure 6.1.3. ASK Modulation flow graph

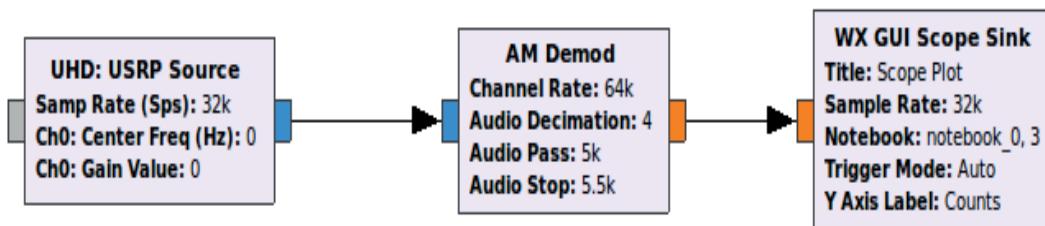


Figure 6.1.4. ASK Demodulation flow graph

NOTE:

In this experiment single USRP B100 is used for both transmission and reception, hence Fig. 6.1.5 represents a loop back system.

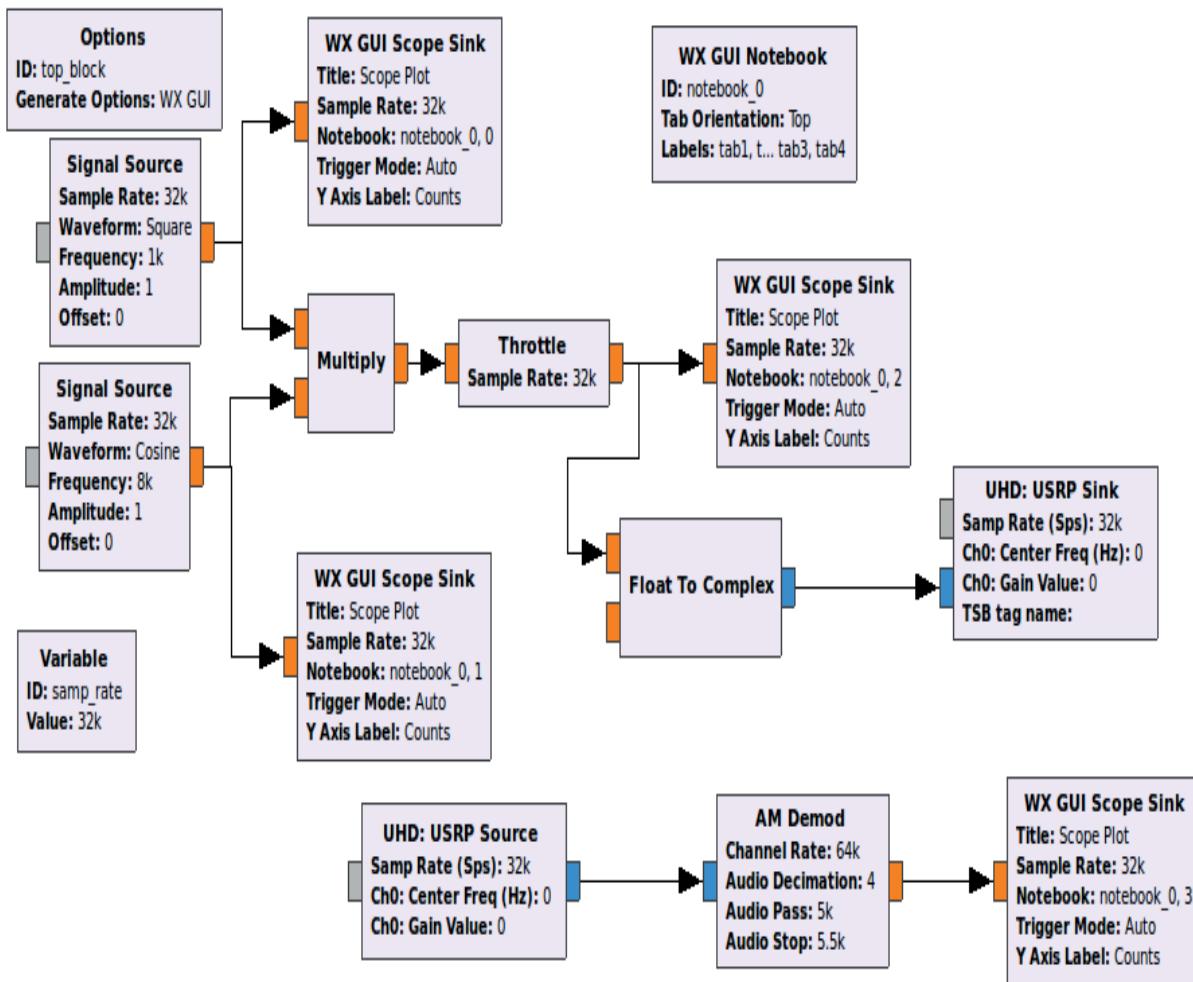


Figure 6.1.5 Gnu radio flow graph for ASK modulation and demodulation.

The Fig 6.1.5 represents a gnu radio flow graph for the ASK modulation and demodulation techniques. The square wave signal is given as input to the signal source with 1 KHz as a carrier frequency. The input signal is multiplied with cosine wave signal with carrier frequency of 8 KHz. The data are sampled at 32 KHz. The signal is received and demodulated with channel rate of 64 KHz with audio passband of 5 KHz and stopband of 5.5 KHz.

Results

Fig 6.1.6 represents an input square wave signals with a carrier frequency signals of 1 KHz. The input signals are multiplied by the cosine wave carrier signals with the carrier.

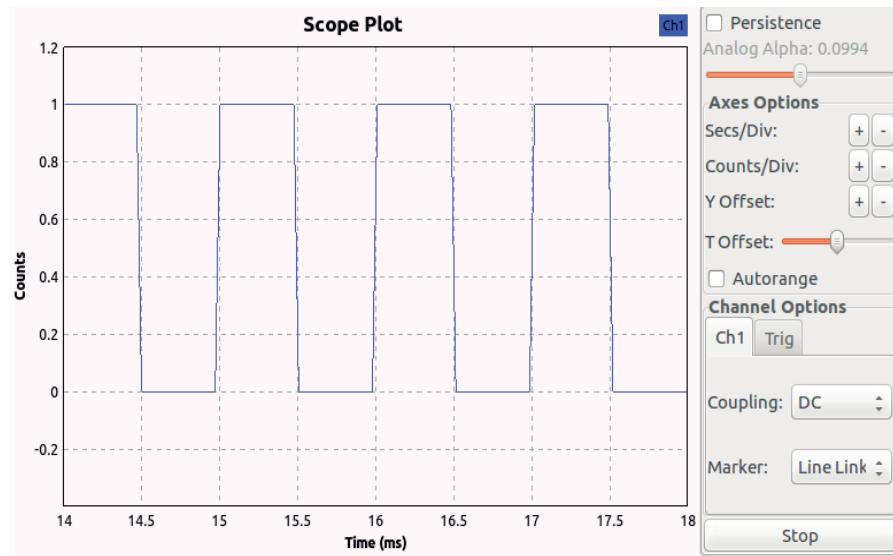


Figure 6.1.6.The input square wave signal.

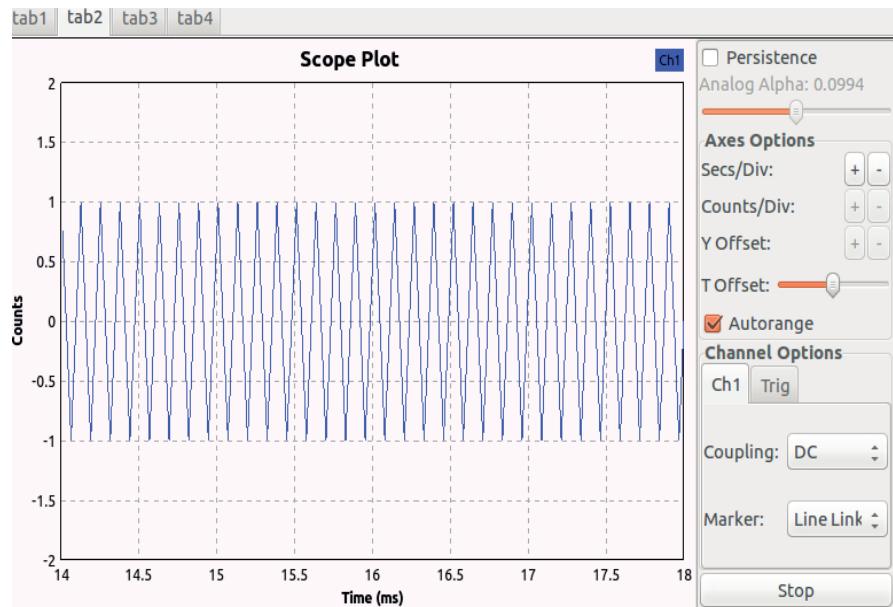


Figure 6.1.7 the carrier wave signals with 8 KHz carrier frequency.

Frequency signals with 8 KHz as shown in Fig 6.1.7. The fig 6.1.8 is the ASK Modulated received signals with high amplitude carrier signals with 1 binary bit symbol and no signals transmission for 0 binary bit symbol. Fig 6.1.9 represent the ASK demodulated output signal with some irregularity which can be properly clipped off with the help of a clipper.

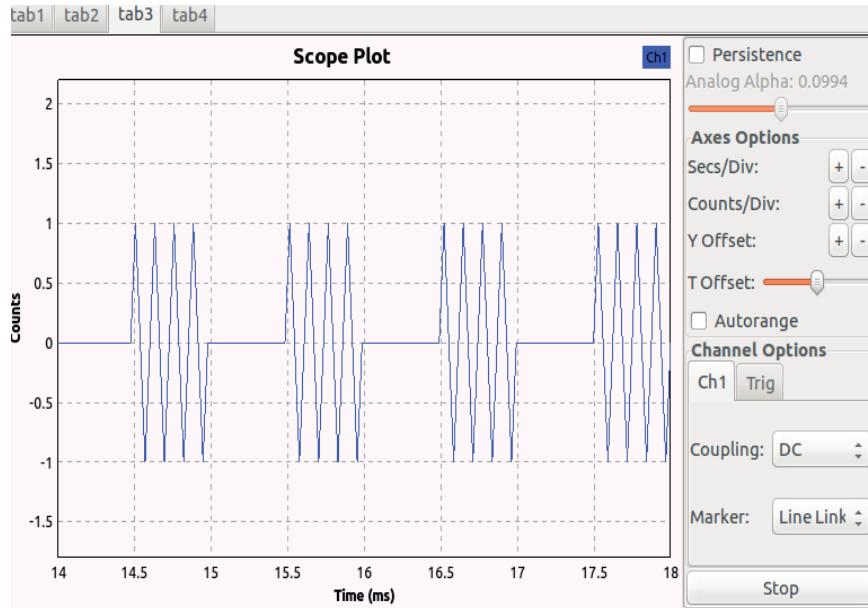


Figure 6.1.8 The ASK Modulated received signals.

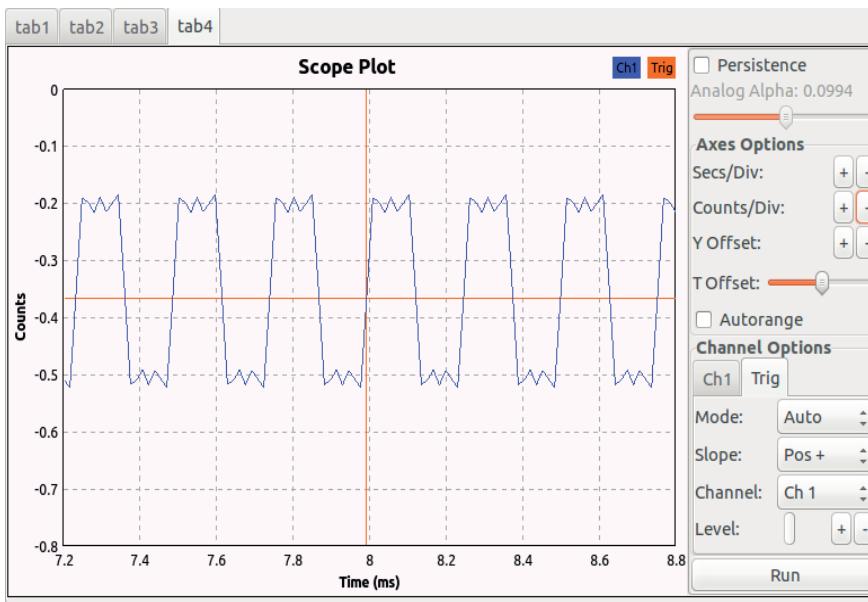


Figure 6.1.9 The ASK Demodulated received signals

Inference

Hence this experiment successfully results in the implementation of Amplitude Shift Keying modulation and demodulation using Gnu radio and finally validating using USRP B100. From the experiment it was observed that the received output signal can be properly achieved with some irregularity. Hence it can be solved with proper usage of clipper which can clip off the signals from the top in order to have a smooth representation of square wave signal.

EXPERIMENT 2

Aim

Implement Phase shift keying (PSK) modulation using Gnu Radio.

Introduction

PSK is abbreviated as Phase Shift Keying which is defined as the digital modulation technique in which the phase of the carrier signal changes with change in input signal at a particular time instance. Depending on the shift of the phase of signals, PSK is broadly divided in two types (I.e. BPSK, QPSK). Here in this experiment, we shall implement the QPSK modulation (I.e. there is change in the phase of signals with 90 degree out of phase). The change in the phase indicates that the amount of data can be transmitted in each cycle. Hence it conveys the data by changing the phase of carrier waves.

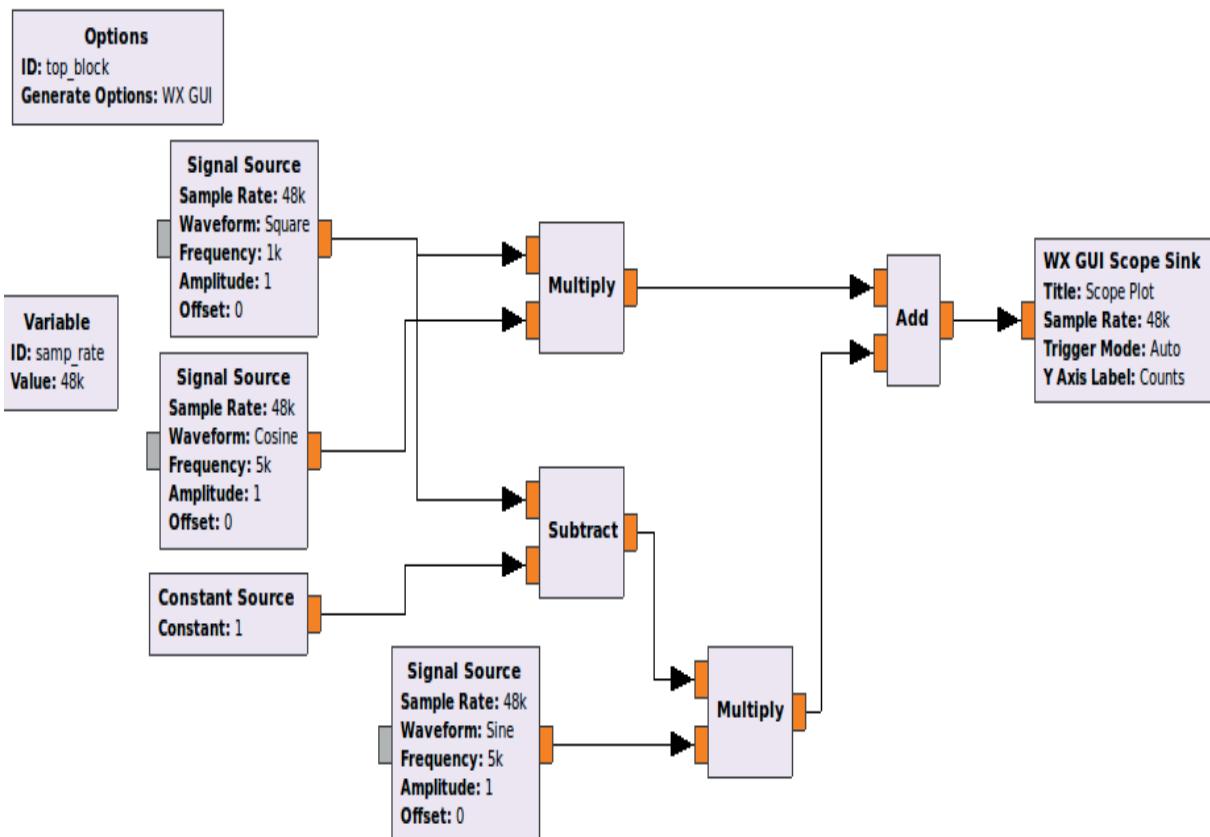


Figure 6.2.1. Phase shift keying Gnu radio Flowgraph.

The figure above indicates the architecture of PSK using Gnu radio. There are two carrier signals having same frequency of 5 KHz with different amplitude structure (i.e. sine wave and cosine wave). A square wave is given as an input which provides alternate 0 and 1 bit in the form of pulse. The signals are alternately multiplied, subtracted and finally adding the combination of two different signals in order to analyze the shift in the phase of signals with change in the binary bits.

The most popular wireless LAN standard, IEEE 802.11b (Wi-Fi) makes use of different PSKs series depending on the data-rate required. Some standard application based on the data rate is as follows:

1. For the basic data-rate of 1 Mbit/s, the wireless LAN uses DBPSK.
2. For the extended data-rate of 2 Mbit/s, DQPSK is used.
3. For data-rate ranging from 5.5 Mbit/s to the full data-rate of 11 Mbit/s, QPSK is employed, but has to be coupled with complementary code keying.

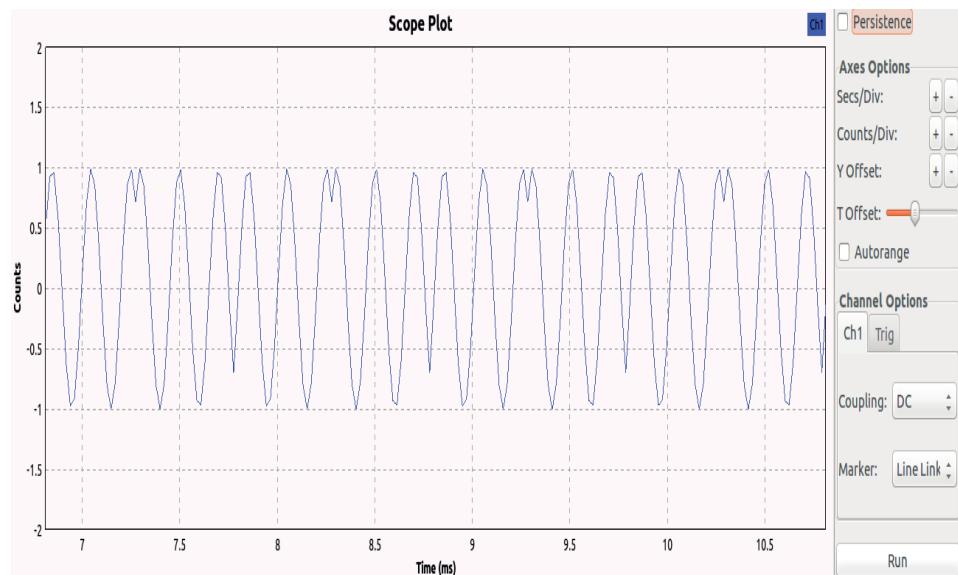


Figure 6.2.2 PSK Output signal

The higher-speed wireless LAN standard, IEEE 802.11g (high speed Wi-Fi) has eight data rates: 6, 9, 12, 18, 24, 36, 48 and 54 Mbit/s where 6 and 9 Mbit/s modes use BPSK. The 12 and 18 Mbit/s modes use QPSK. The figure 6.2.2 indicates the shift in the phase of signals by 90 degree with each time the change in the bits of signals from 0 to 1.

Application

1. It is used in Optical communication.
2. It is used in the local Oscillator.
3. It is used in the Delay and adds modulator.
4. It is used in the Multi-channel WDM.

Inference

This experiment successfully aims in the implementation of Phase shift keying (PSK) digital modulation techniques where there is shift in the phase of signals with the change in the bits of signals.

EXPERIMENT 3

Aim

To implement the Frequency shift keying (FSK) using Gnu radio

Introduction

Frequency Shift Keying (FSK) is defined as a type of digital modulation technique in which the frequency of the carrier signal changes with change in the amplitude of digital signal. For high input i.e. for 1 bit the output signal will have high frequency carrier signals and for low input i.e. 0 bit the output signal will have low frequency carrier signals.

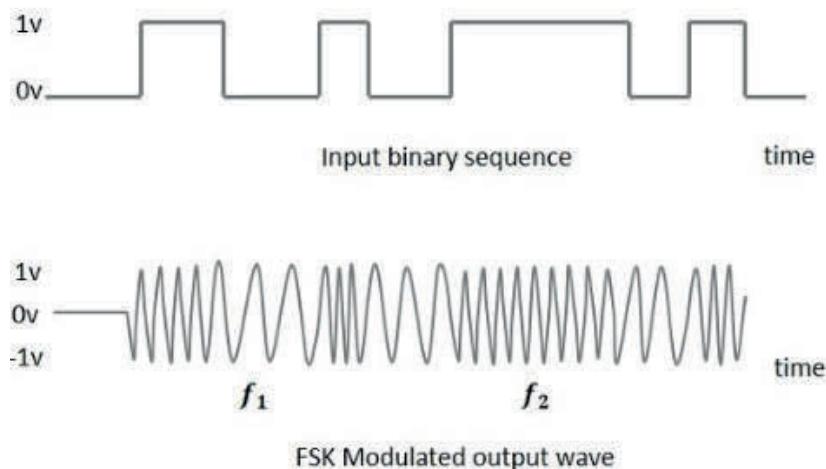


Figure 6.3.1 Representation of FSK modulated signals.

The figure above represent the FSK modulated output signals with two different carrier frequency signals. The compressed carrier represents a high frequency signals, hence for 1 bit is transmitted, then a compressed carrier signal is obtained as shown in fig 6.3.1

Advantage of FSK modulation

1. It has lower probability of error.
2. It provides high SNR (Signal to Noise Ratio).
3. It has higher immunity to noise due to constant envelope.

Disadvantages of FSK modulation techniques

1. It uses larger bandwidth compare to other modulation techniques such as ASK and PSK. Hence it is not bandwidth efficient.
2. The BER (Bit Error Rate) performance in AWGN channel is worse compare to PSK modulation.

Application of FSK modulation techniques

1. It is used on voice grade lines for data rates up to 1200 bps.
2. It is used for high frequency radio transmission from 3 to 30 MHz.
3. It is also used in coaxial cable based LAN (Local Area Network) at higher frequencies.

Gnu Radio Flow graphs

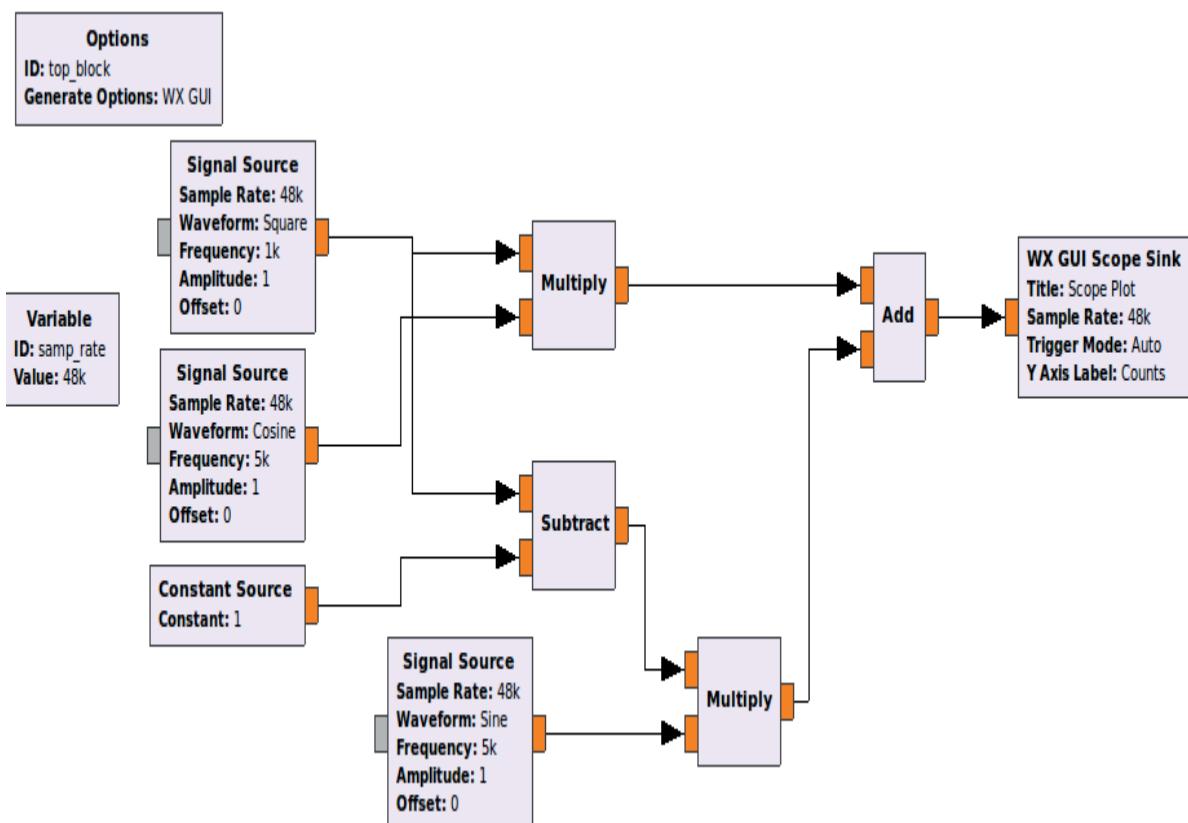


Figure 6.3.2 Frequency shift keying Gnu radio Flowgraph.

Fig 6.3.2 represents the flow graph of frequency shift keying. The two signal source with sine and cosine carrier signal is assumed as an oscillator which generates a same frequency signal of 5 KHz. A square wave signal is generated in order to generate binary bits of 1 and 0 with amplitude of 1. Fig 6.3.3 indicates a two carrier signals. An alternate high and low carrier frequency signal appears depending on the input binary sequence.

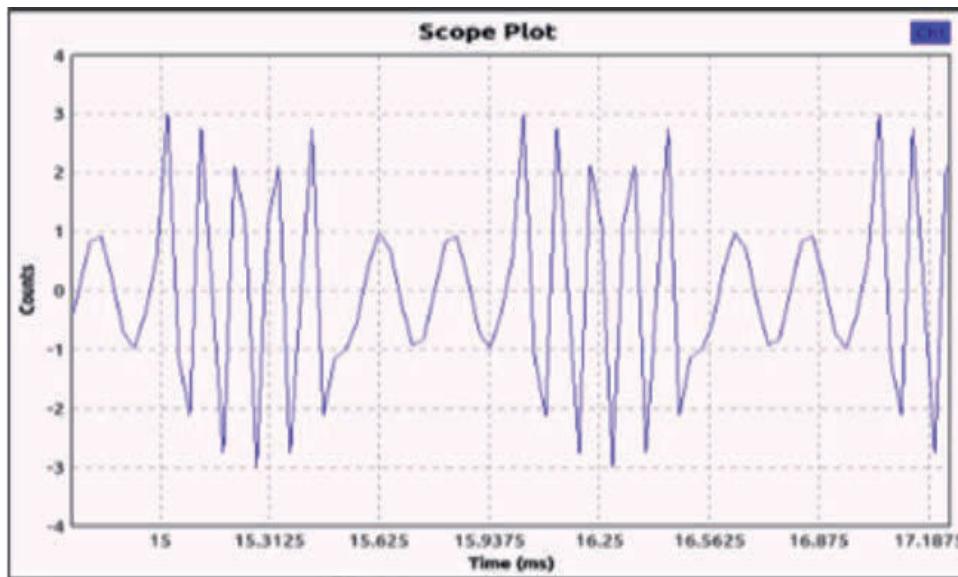


Figure 6.3.3 FSK Output signal.

Inference

The FSK modulation techniques are normally used in communication systems for telemetry and as low frequency radio transmission in the VLF bands. From this experiment, it can be observed that the spectral efficiency decreases to large extent due to sharp jump of cut off frequency. Hence it can be further solved by Gaussian frequency shift keying (GFSK) where the baseband pulses are first passed through the Gaussian filter in order to make pulse smooth hence limit the spectrum within the bandwidth. The filtered outputs are further modulated using FSK modulation techniques.

EXPERIMENT 4

Aim

To implement the Different modulation techniques using Gnu radio and validate using USRP B100.

Introduction

1. Bpsk Modulation techniques

Phase Shift Keying (PSK) is the type of digital modulation technique in which the phase of the carrier signal changes with change in the sine and cosine inputs at a particular time. Binary phase shift keying (BPSK) is one such type of PSK modulation techniques where the sine wave carrier takes two phase reversals such as 0° and 180° .

Consider a sinusoidal carrier signal. The modulation of BPSK is done using a balance modulator, which multiplies the two signals applied at the input. For a zero binary input, the phase will be 0° and for a high input, the phase reversal is of 180° .

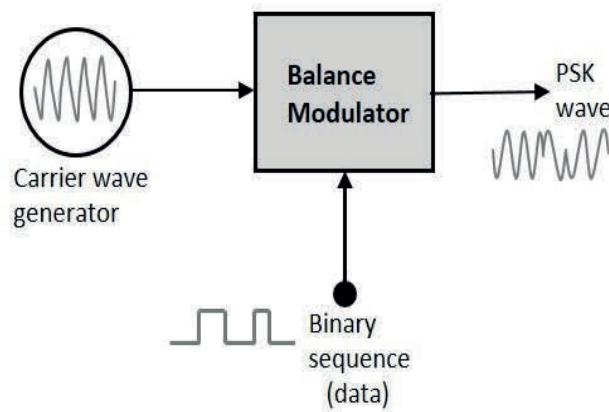


Figure 6.4.1 BPSK Modulator.

The output sine wave of the modulator will be the direct input carrier signal with 180° phase shifted carrier signals. The wave shape is symmetrical' at each phase transition. This is because the bit rate is a sub-multiple of the carrier frequency ω / (2π).

The appearance of a BPSK signal in the time domain is shown below.

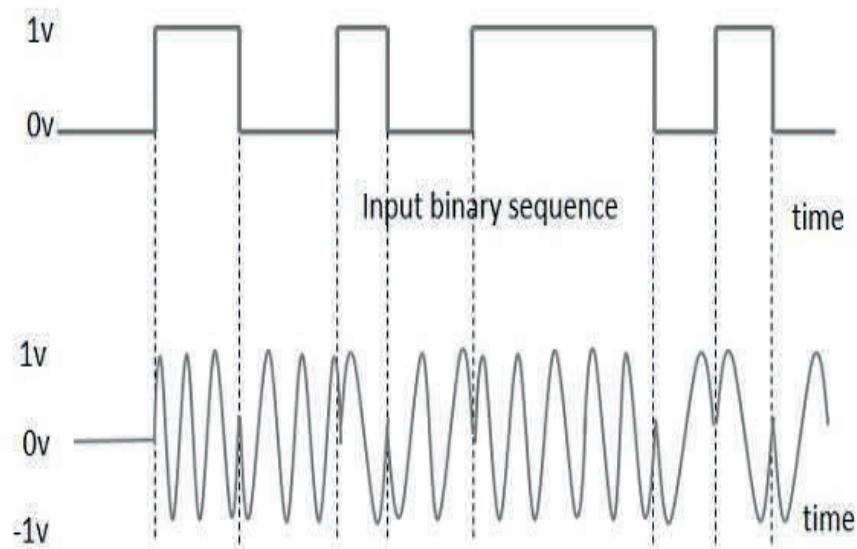


Figure 6.4.2 BPSK Modulated output signal.

Demodulation of a BPSK signal can be considered a two-stage process.

1. Translation back to baseband, with recovery of the bandlimited message waveform
2. Regeneration from the bandlimited waveform back to the binary message bit stream.

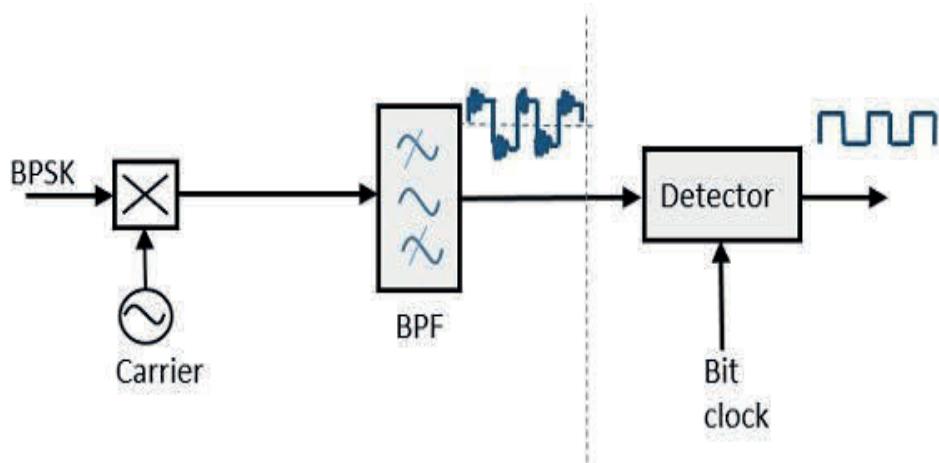


Figure 6.4.3 BPSK Demodulated output signal.

Gnu radio Flowgraph

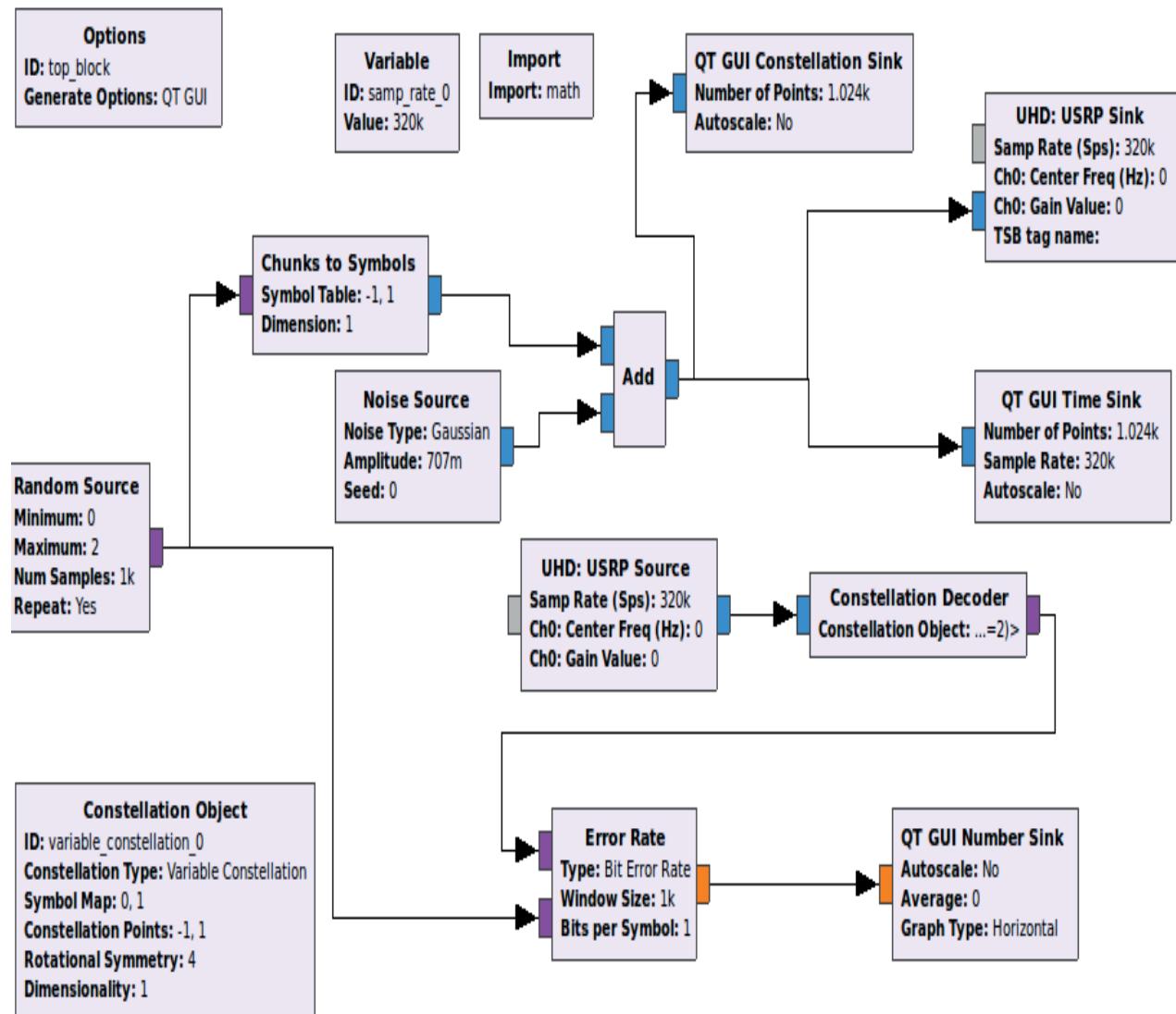


Figure 6.4.4 Gnu radio Flowgraph.

The figure above describes about the implementation of BPSK with the baseband types of signals i.e. low frequency signals with no carrier. A random source block is used to generate random signals with the sample rate of 320 KHz. The random generated signals are converted to (1,-1) as per BPSK modulation techniques using chunks to symbols. A Gaussian Noise is added to the baseband signals with variation in the Signal to Noise ratio from 0 dB to 15Db. The signals are further transmitted using 320 KHz sample rate. The received signal is further compared with change in the bits in order to

analyze the error using Bit error rate block. The bits per symbol are 1 for BPSK. Qt GUI number Sink is used to dispense the scale decimal value.

Results

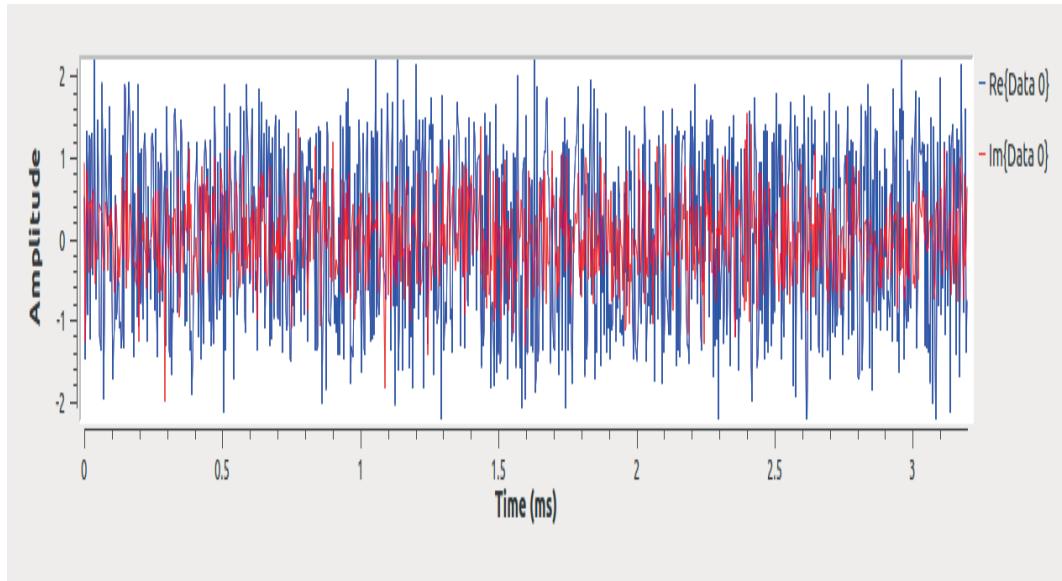


Figure 6.4.5 Time domain representation with SNR 0 dB.

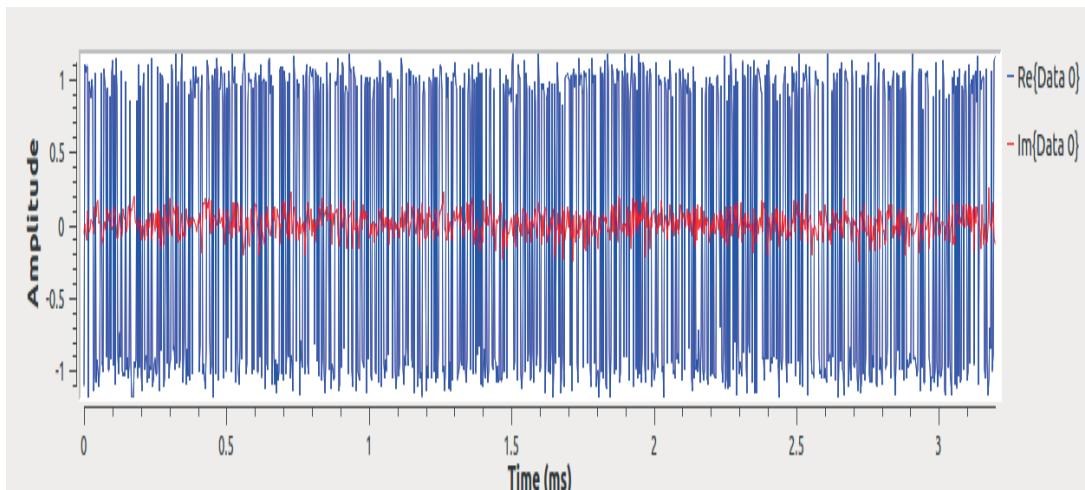


Figure 6.4.6 Time domain representation with SNR 15 dB.

The figure above indicates the variation in the time response of random signal and Gaussian noise with respect to increase in the SNR. For analysis purposes, SNR is set to 0

dB and 15 dB. For 0 dB SNR, the effects of Gaussian noise are very high as compared to 15 dB SNR. The Red colour indicates the Gaussian noise added to the signals. The figure bellow represents a constellation mapping of BPSK constellation mapping. It represents the number of signals modulated by the digital modulation.

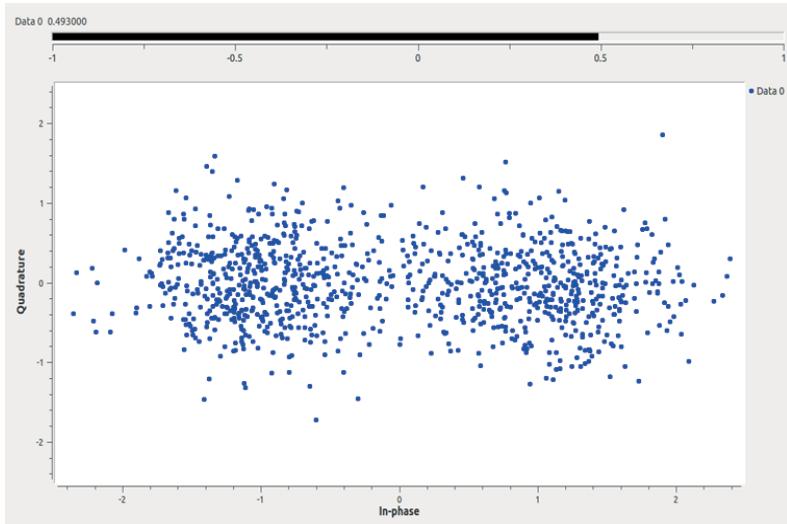


Figure 6.4.7 the effect of Noise on Constellation mapping with 0 dB SNR.

The figure above represents a constellation mapping with 15 dB SNR. Thus the effect of Gaussian noise on the constellation mapping is low as compared to the constellation mapping with 0 dB SNR as shown in the figure bellow.

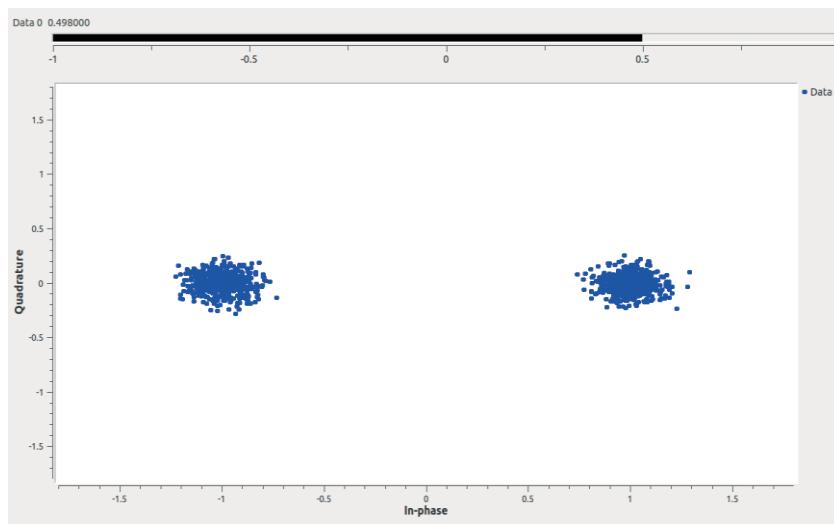


Figure 6.4.8 the effect of Noise on Constellation mapping with 15 dB SNR.

2. QPSK Modulation techniques

Quadrature phase shift keying (BPSK) is one such type of PSK modulation techniques where the sine wave carrier takes four phase reversals with a difference of 90° among them. In QPSK, the carrier varies in terms of phase but not in terms of frequency, and there are four possible phase shifts. Four QPSK phase shifts are 45° , 135° , 225° , and 315° . QPSK transmits 2 bits per symbol, hence QPSK can be used to double the data rate with same Bandwidth of BPSK or even Bandwidth can be half the data rate.

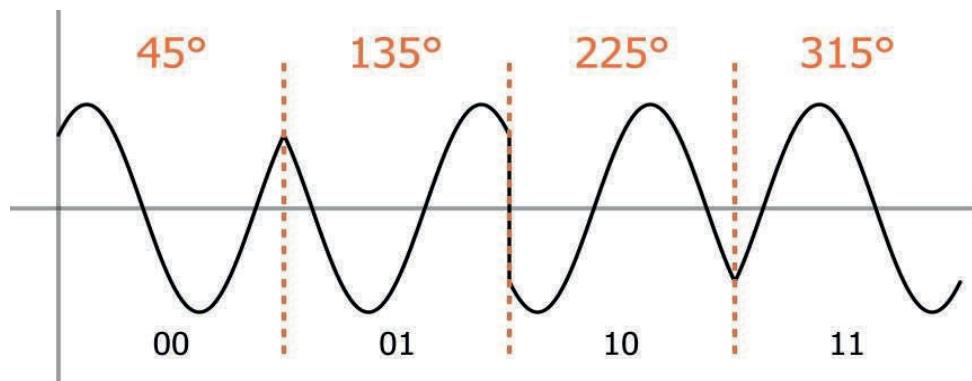


Figure 6.4.9 The Phase shift of QPSK

QPSK symbol doesn't represent 0 or 1 but it represents in 00, 01, 10, or 11 form as shown in the above figure. It is also called as Quaternary phase shift keying.

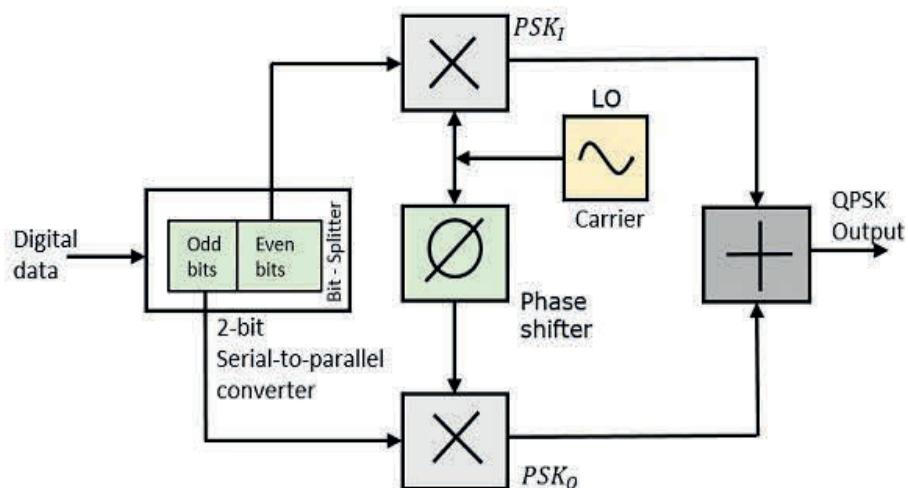


Figure 6.4.10 QPSK Modulator

The message signal's even bits and odd bits are separated by the bits splitter and are multiplied with the same carrier to generate QPSK modulated signals. Demodulation of a QPSK signal can be considered a two-stage process.

1. Translation back to baseband, with recovery of the bandlimited message waveform
2. Regeneration from the bandlimited waveform back to the binary message bit stream.

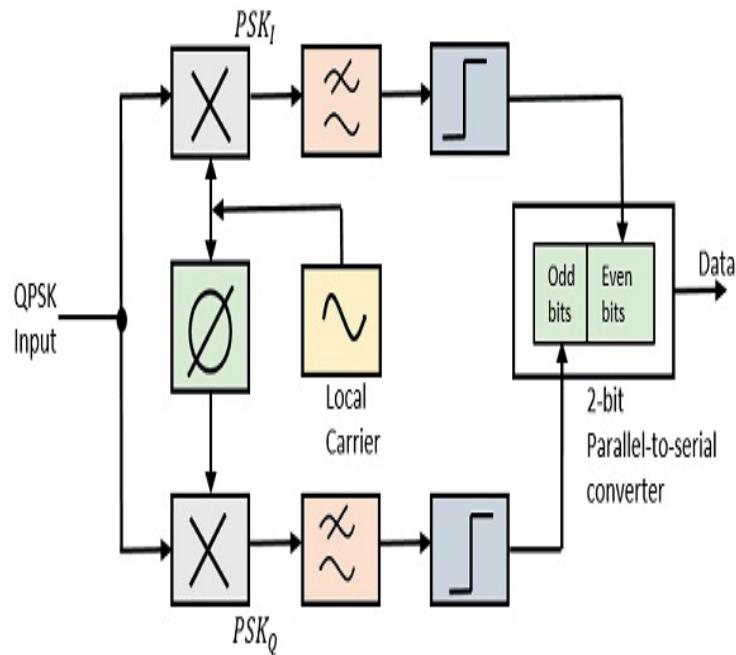


Figure 6.4.11 QPSK Demodulator

The two product detectors at the input of demodulator simultaneously demodulate the two BPSK signals. The pair of bits is recovered from the original data. These signals after processing are passed to the parallel to serial converter.

Advantage of QPSK

1. Good noise immunity.
2. For the same bit error rate, the bandwidth required by QPSK is reduced to half as compared to BPSK.

3. Low error probability.
4. The information transmission rate of QPSK is high as compared to BPSK.

Gnu radio Flowgraph

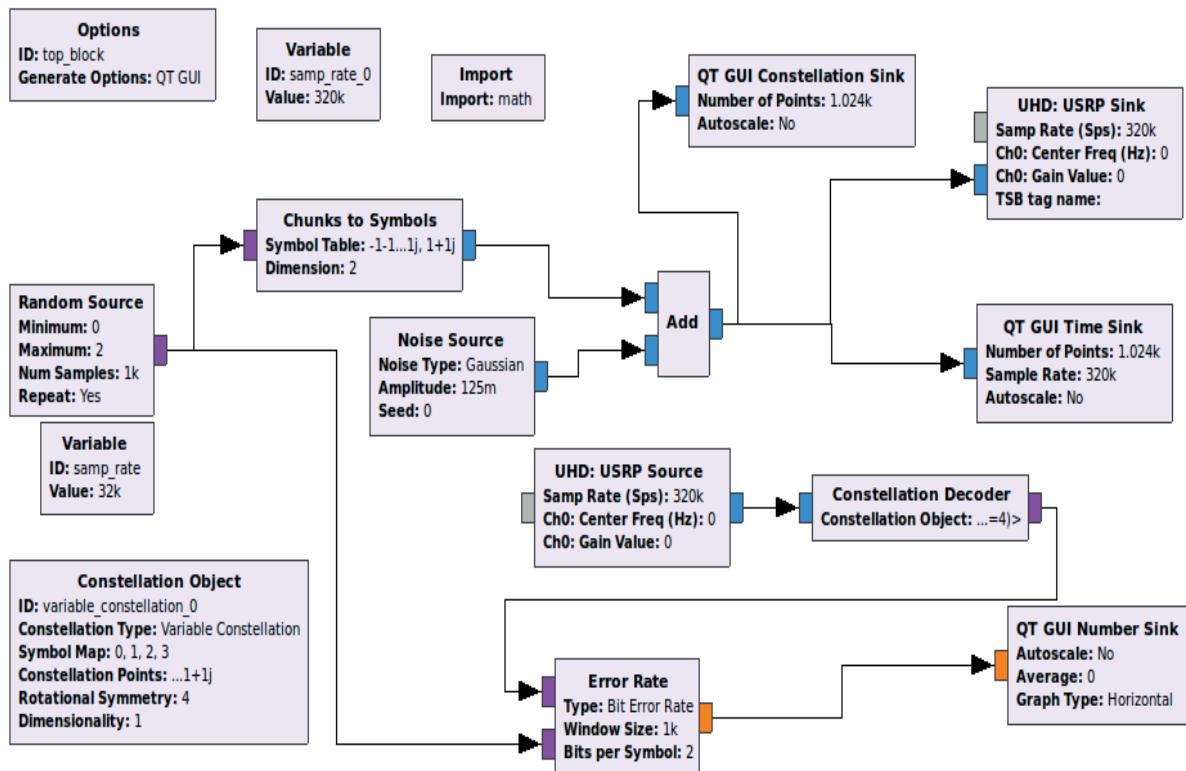


Figure 6.4.12 QPSK Gnu radio Flowgraph

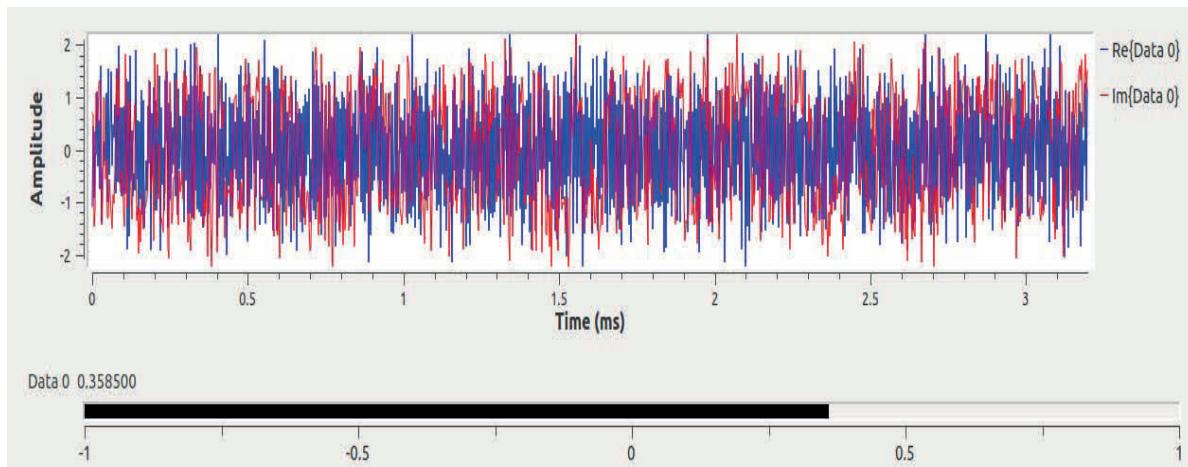


Figure 6.4.13 Time domain representation of signal with 0 dB SNR.

The Figure above represent an implementation of QPSK (4 PSK) modulation techniques using SDR. The fig 6.4.12 and 6.4.13 represent the time domain representation of QPSK modulated signal at 0 dB and 15 dB SNR.

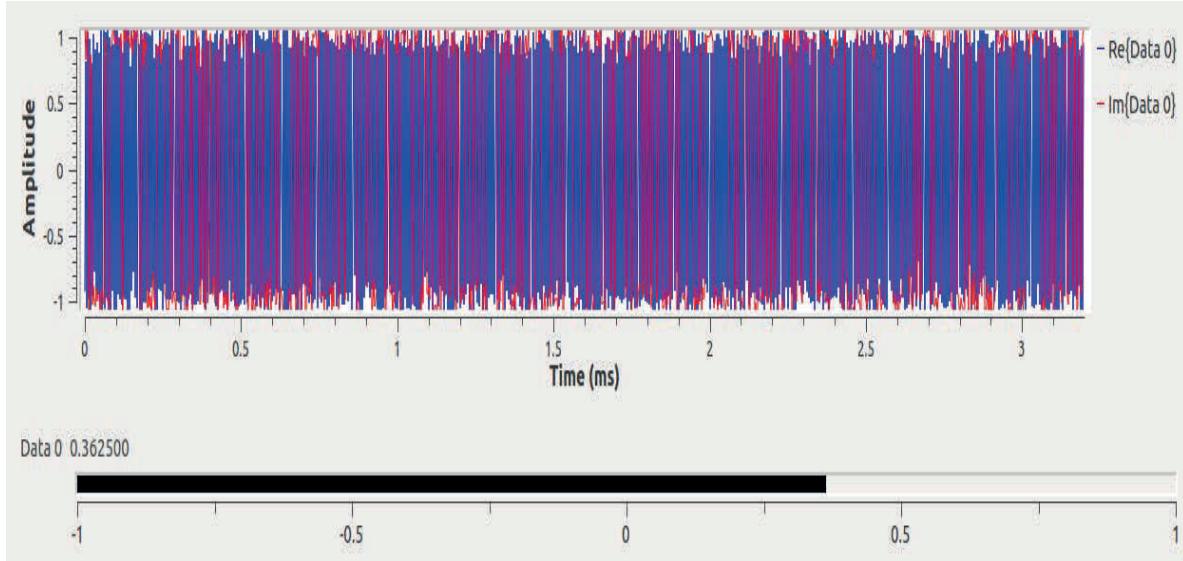


Figure 6.4.14 Time domain representation of signal with 15 dB SNR.

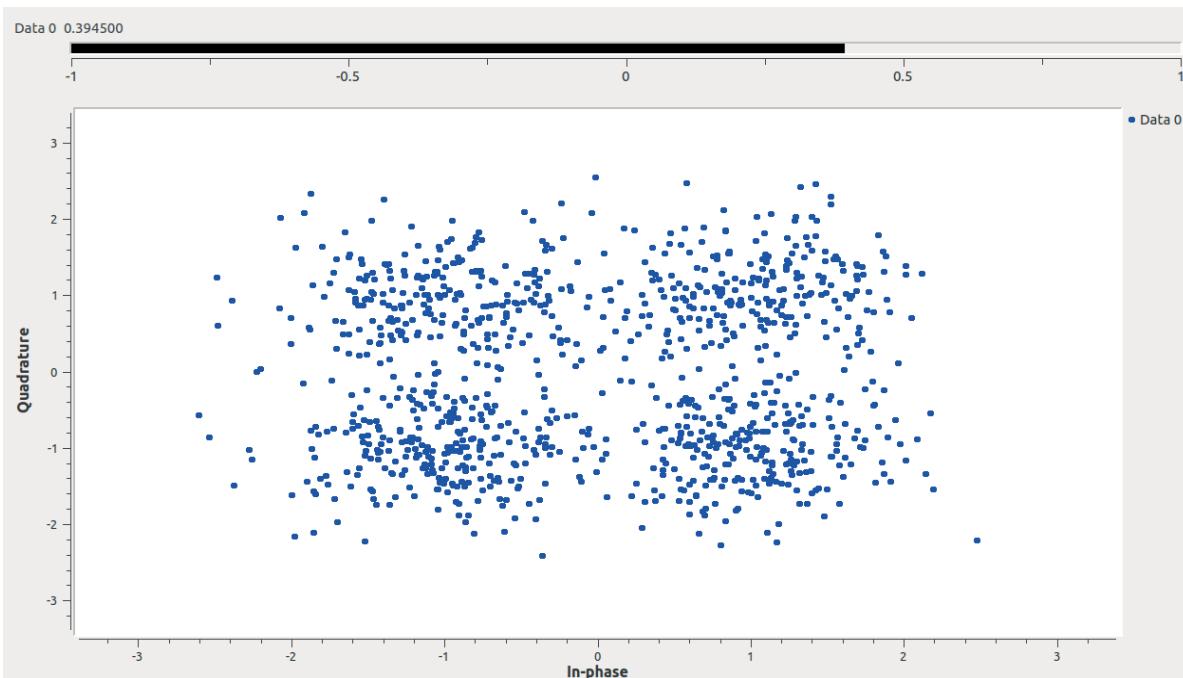


Figure 6.4.15 Constellation mapping with the effect of Noise having 0 dB SNR.

The figure above represents the constellation mapping of QPSK with the effect of 15dB SNR. This indicates that the 4 symbols are digital modulated. The figure bellow indicates the effects of Gaussian noise on constellation mapping with 0 dB SNR.

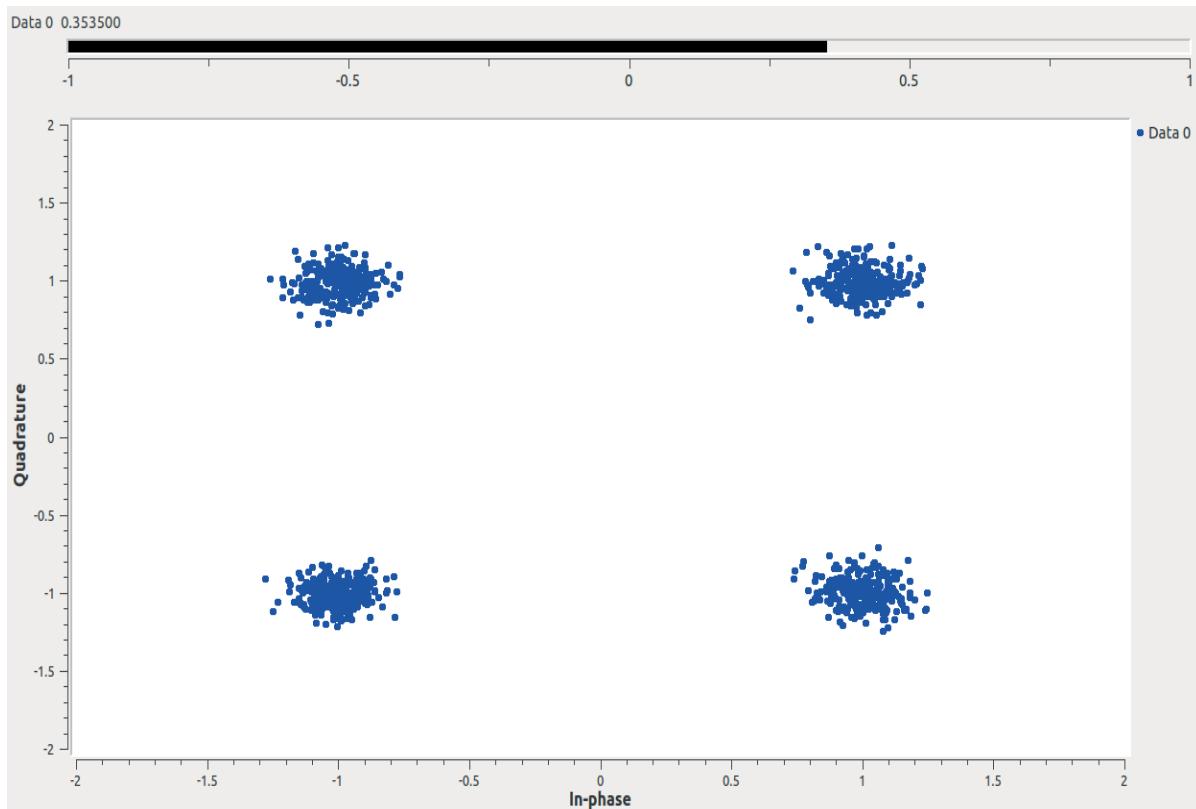
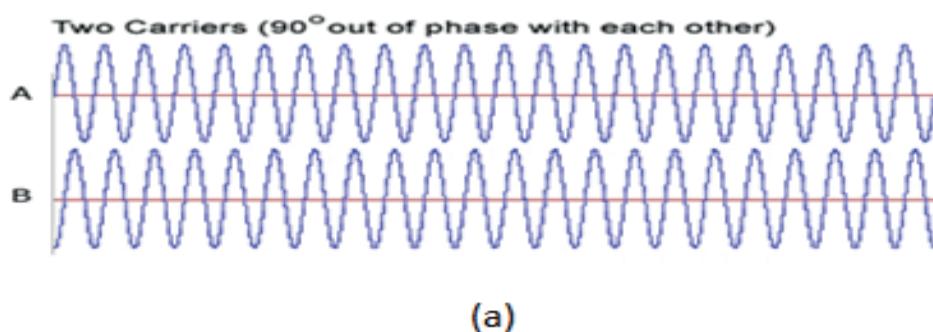
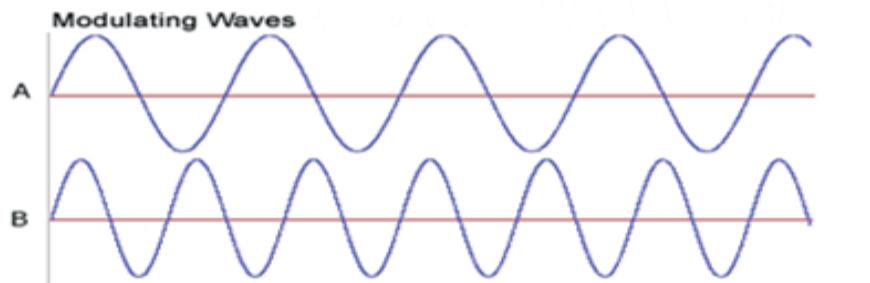


Figure 6.4.16 Constellation mapping with the effect of Noise having 15 dB SNR.

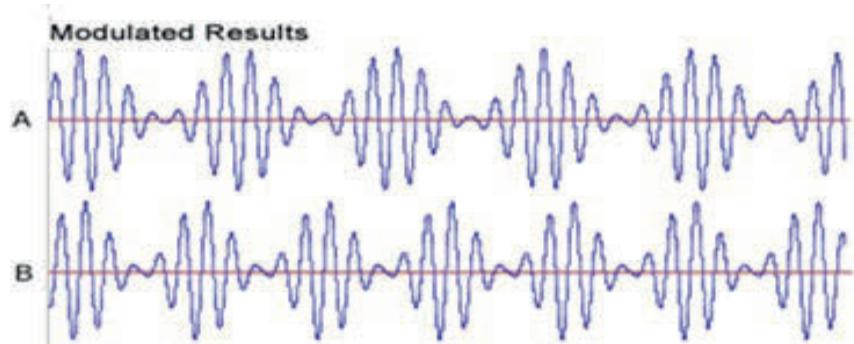
3. QAM techniques:

QAM is abbreviated as Quadrature Amplitude Modulation scheme. It is defined as a data signals which are modulated on the two carriers shifted in phase by 90 degrees.

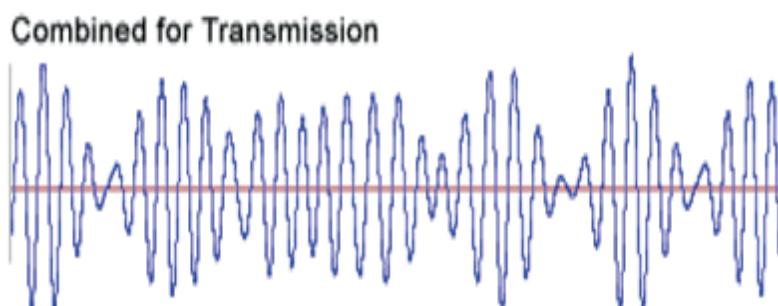




(b)



(c)



(d)

Figure 6.4.17 (a, b, c, d) Flow of QAM signals.

One signal is called the I signal, and the other is called the Q signal. Mathematically, these signals are represented by a sine wave, and the cosine wave. The resultant QAM output is a combination of both amplitude and phase variations. QAM combines two amplitude modulated (AM) signals into a single channel; hence it doubles the effective

bandwidth. This result in multiple analog signals placed on a single carrier. Considering the example of Television signals, which have both colour signals and sound.

Modulation Level (QAM)	Bits/Symbol Bits/s/Hz	Incremental Capacity Gain
4 (QPSK)	2	-
8	3	50%
16	4	33%
32	5	25%
64	6	20%
128	7	17%
256	8	14%
512	9	13%
1024	10	11%
2048	11	10%

Figure 6.4.18 Description about the Bits/symbol based on the types of modulation techniques.

The above figure represents the modulation levels of QAM which depends based on the constellation mapping techniques. QAM use a constellation with the number of points equal to a power of 2. For 8QAM, $2^3 = 8$ where 3 is bits per symbol and 8 represents the constellation mapping. A constellation mapping defines the number of data signals modulated by the digital modulation scheme. Bits per symbol represent the number of bits transmitted per symbol. For example, 8QAM uses 8 point constellation. The advantage of moving to the higher order QAM is that, it is possible to transmit more bits per symbol. But the disadvantage is that the constellation points are closer together and therefore the link is less immune to noise.

The two modulated carriers are combined at the source for transmission. At the destination, the carriers are separated, the data is extracted from each, and then the data is combined into the original modulating information.

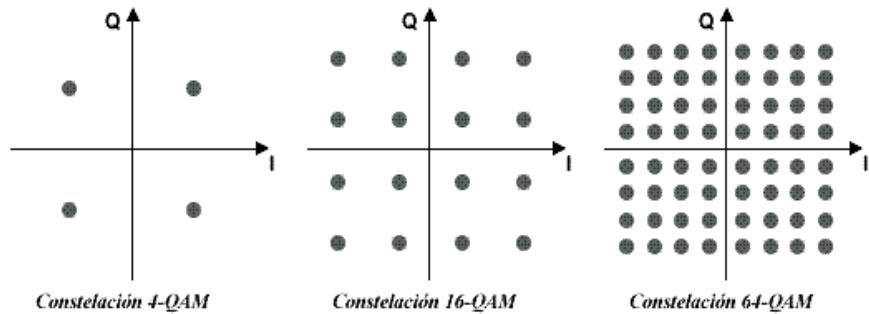


Figure 6.4.19 Constellation point for different types of QAM.

QAM modulation scheme encodes the data signals by varying both amplitude and phase of a carrier signal. Hence it is called as a combination of both ASK and PSK modulation scheme.

There are two carrier signals which are 90 degree phase shifted. These carrier signals are then amplitude modulated with the two data streams known as the In-phase and the quadrature data streams. The two resultant signals are added and processed as required in the RF signal chain.

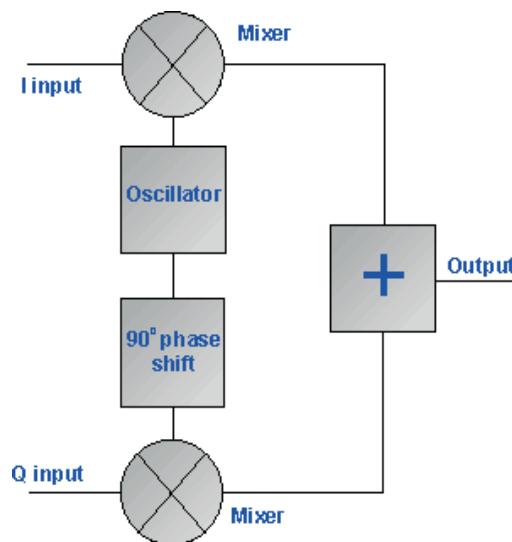


Figure 6.4.20 QAM modulation

The signals enter the system, they are split and each side is applied to a mixer. One half has the In phase local oscillator applied and the other half has the quadrature oscillator signal applied. A further requirement is to derive a local oscillator signal for the demodulation that is exactly on the required frequency for the signal. Any frequency offset will be a change in the phase of the local oscillator signal with respect to the two double sideband suppressed carrier constituents of the overall signal.

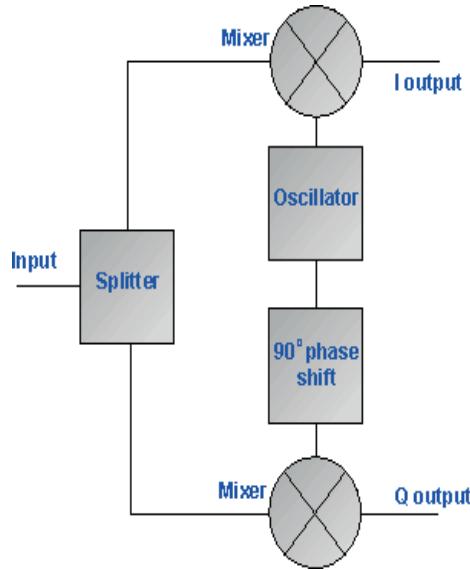


Fig.6.4.21 QAM demodulation

Advantage of QAM

1. Efficient usage of Bandwidth.
2. Provides high data rate.
3. Increases the quality of signals transmitted and received.

Disadvantages of QAM

1. As constellation states are closer, QAM modulation is more susceptible to the noise.
2. QAM receiver is more complex compare to receivers of other modulation types.

3. QAM uses amplitude component of signal to represent binary data, hence linearity is needed which consumes more power.

Gnu radio Flow graph

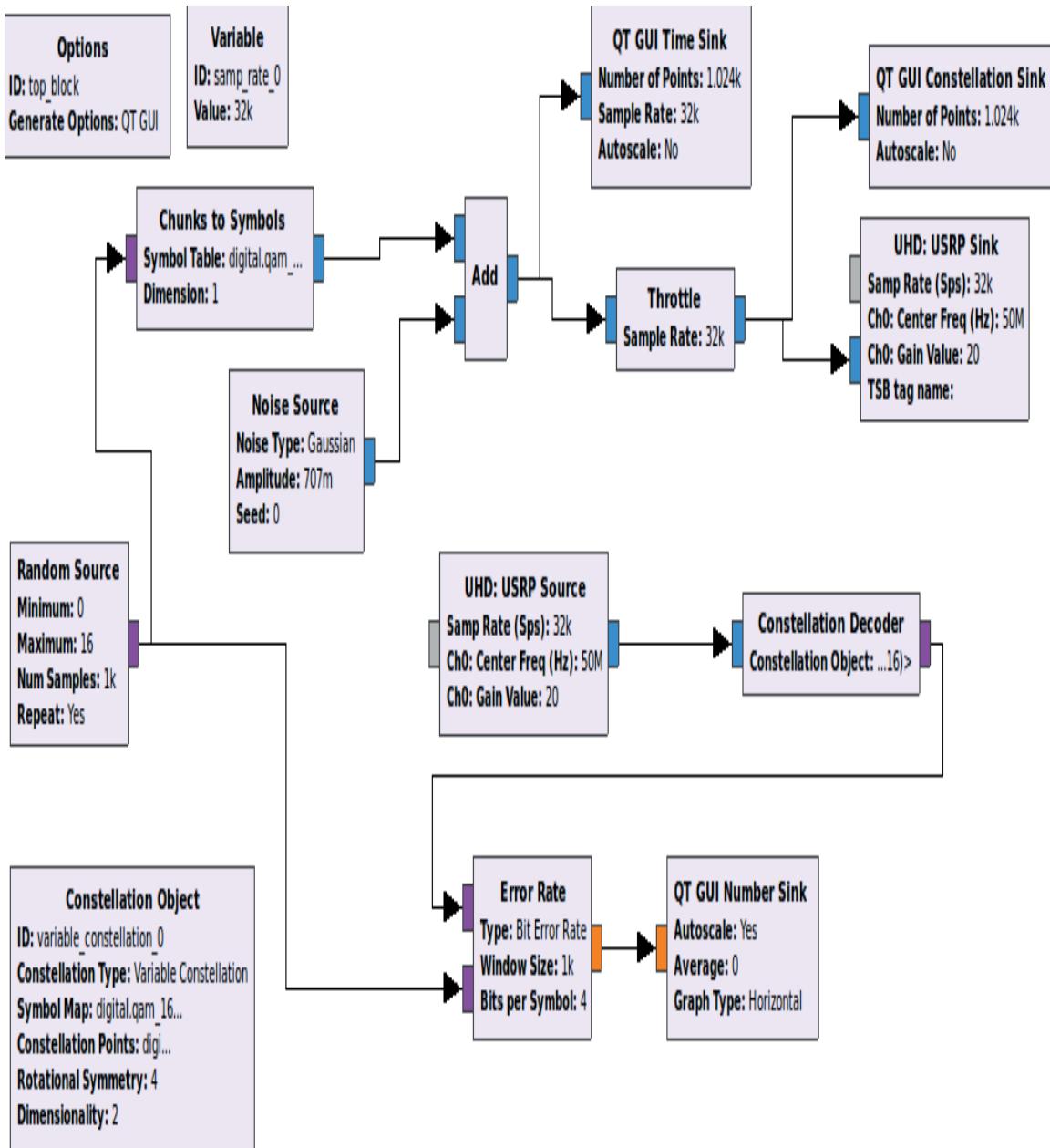


Figure 6.4.22 QAM Gnu radio Flowgraph

The figure represents 16 QAM where random generated signals are passed through the chunks to symbol with symbol table indicating the constellation symbol of 16 QAM. The

fig 6.4.23 and 6.4.24 represents the effect of Gaussian noise on time domain of a signal with variation in SNR (considering 0 dB and 15 dB SNR).

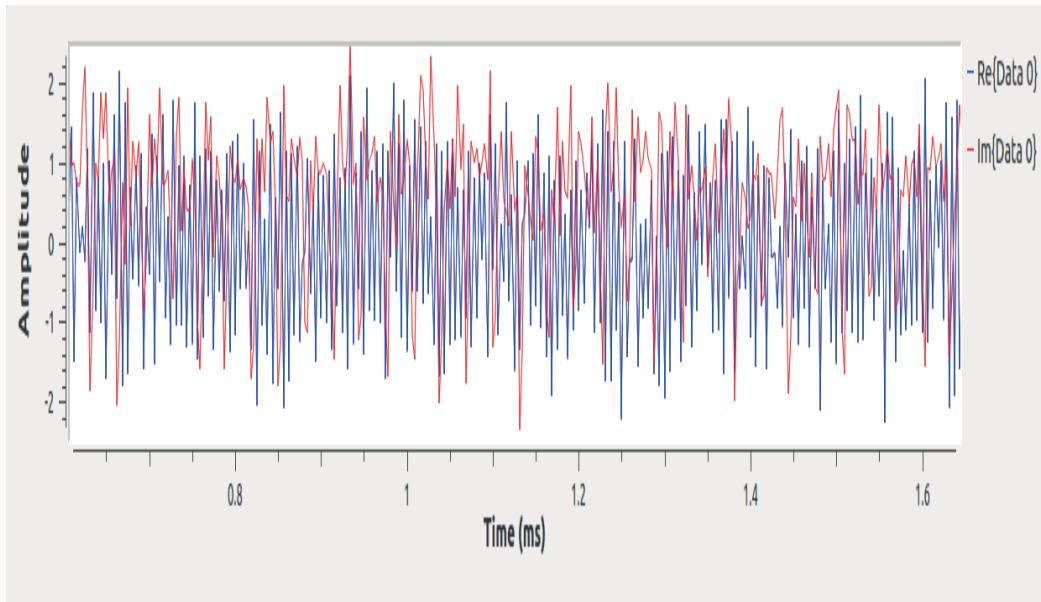


Figure 6.4.23 Time domain representation of QAM signal with 0 dB SNR.

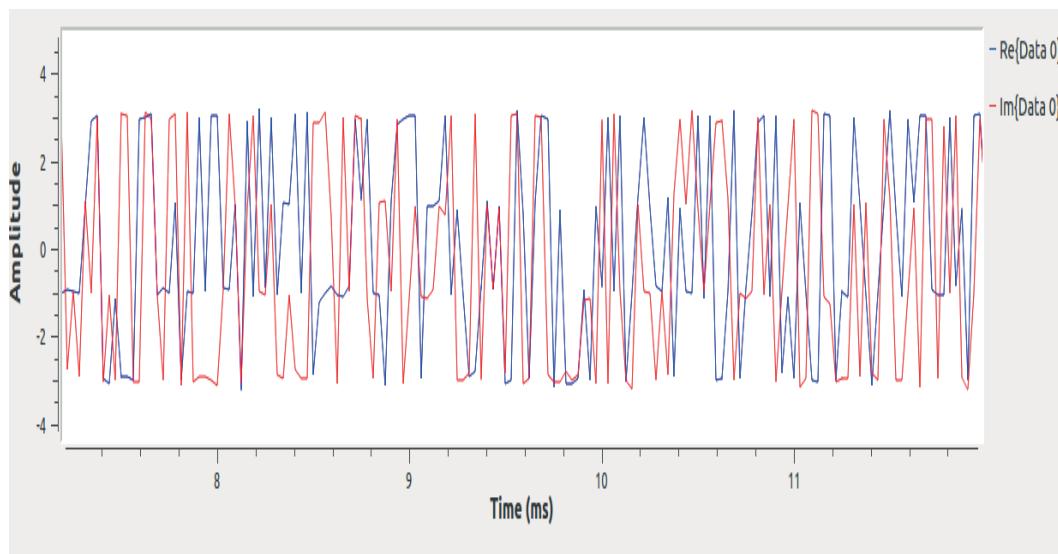


Figure 6.4.24 Time domain representation of QAM signal with 15 dB SNR.

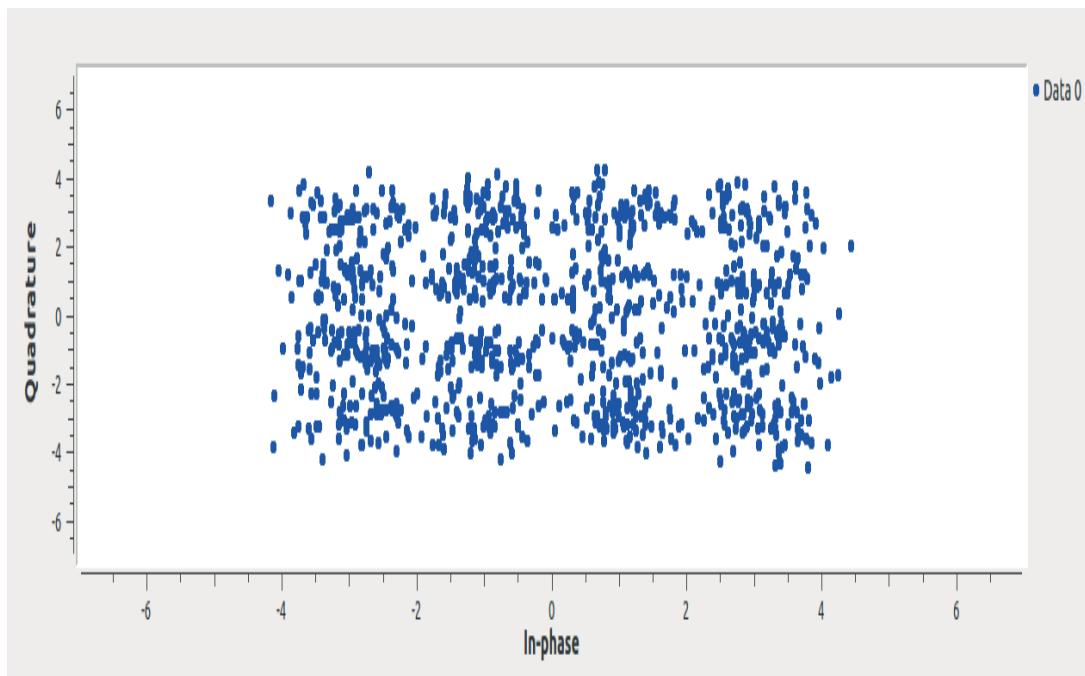


Figure 6.4.25 Constellation points with 0 dB SNR.

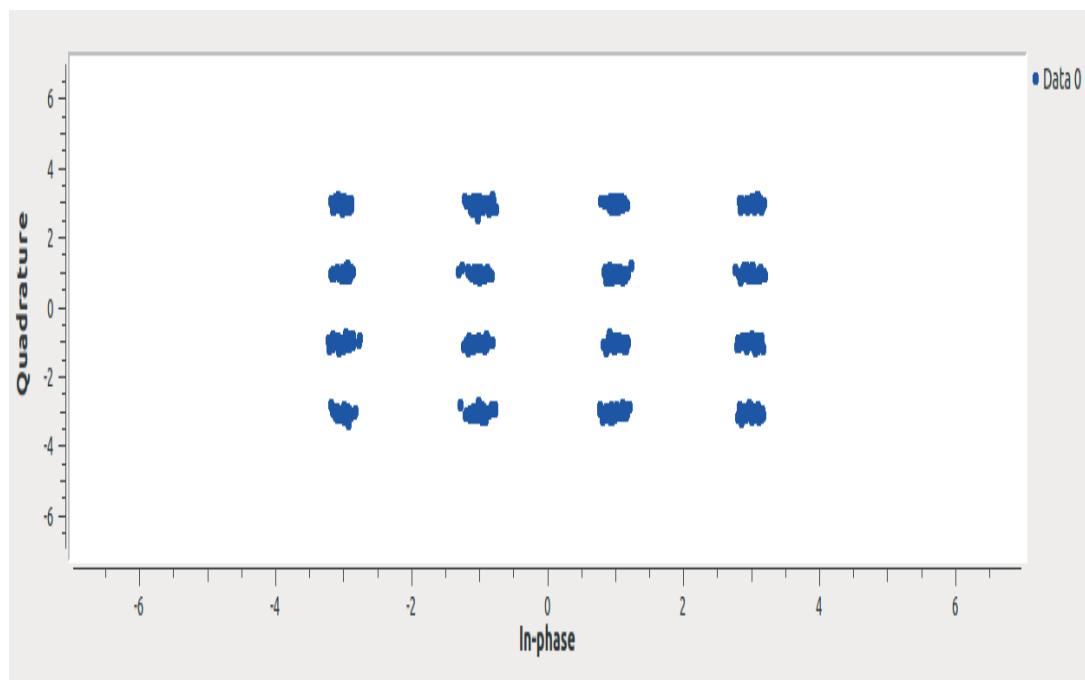


Figure 6.4.26 Constellation points with 15 dB SNR.

The figure 6.4.25 and 6.4.26 represent a constellation mapping of 16 QAM with the effect of Gaussian noise. For the comparison purposes the SNR is varied from 0 db to 15 db.

Table 12: Comparisons of different type of Digital Modulation techniques

Parameter	BPSK	QPSK	16QAM
Data rate	Transmits one bit per symbol with 2 constellation points. Hence BPSK has low data rate.	Transmits 2 bits per symbol with 4 constellation points. Hence QPSK doubles the data rate with same Bandwidth.	Transmits 4 bits per symbol with 16 constellation points. Hence it increases the data rate by 4 times with same Bandwidth.
Complexity	Simpler to design	Complex than BPSK	Complex than PSK Series as it transmits more bits/Bw Hence cost increases.
Bit Error Rate	BER decreases with poor Quality of received signals From the exp $BER = 0.45 \times 10^{-1}$ Is recorded using USRP B100.	BER increases than BPSK with increase in the Quality of signals. From the exp, $BER = 0.6 \times 10^{-1}$ Is recorded using USRP B100.	BER increases with increase in the quality of received signal. From the exp, $BER = 0.77 \times 10^{-1}$ Is recorded using USRP B100.
Relation between Bit rate and Baud rate.	Bit rate = Baud rate.	Bit rate = 2X Baud rate.	Bit rate = 3X Baud rate.

Inference

Hence this experiment showcases the comparisons between the different types of PSK and QAM modulation techniques. From the experiments, it is observed that the higher order digital modulation techniques like QAM, results in increase in the Bit error rate with increase in the quality of transmission and reception of real time signals. Low modulation techniques has low bit error rate with compromising the Quality of signals. Bit error rate can be further reduced using proper selection channel estimation techniques and noise cancellation process at the receiver end.

Chapter 7

Transmission and Reception of Real World Audio/Video/Text/Image using SDR

For the transmission and reception of real world data like audio/video Etc. in Gnu radio, Packet encoder and decoder block are used. USRP B100 series with WBX RF Daughterboard is used to demonstrate the same in this chapter. WBX RF Daughter board support radio frequency in the range from 50MHz to 2.2 GHZ with 100 mW output power.

Table 13: List of real time transmission and reception experiments

Sr No	Description
1	Transmission and reception of Text messages using SDR
2	Transmission and Reception of Audio signals using <ul style="list-style-type: none">1. BPSK2. QPSK3. QAM4. GMSK
3	Transmission and Reception of Video signals using Gstreamer and Gnu radio

EXPERIMENT 1

Aim

To implement the transmission and reception of text messages using Gnu Radio and Validation using Software Defined Radio (USRP B100).

Gnu Radio Flow graphs

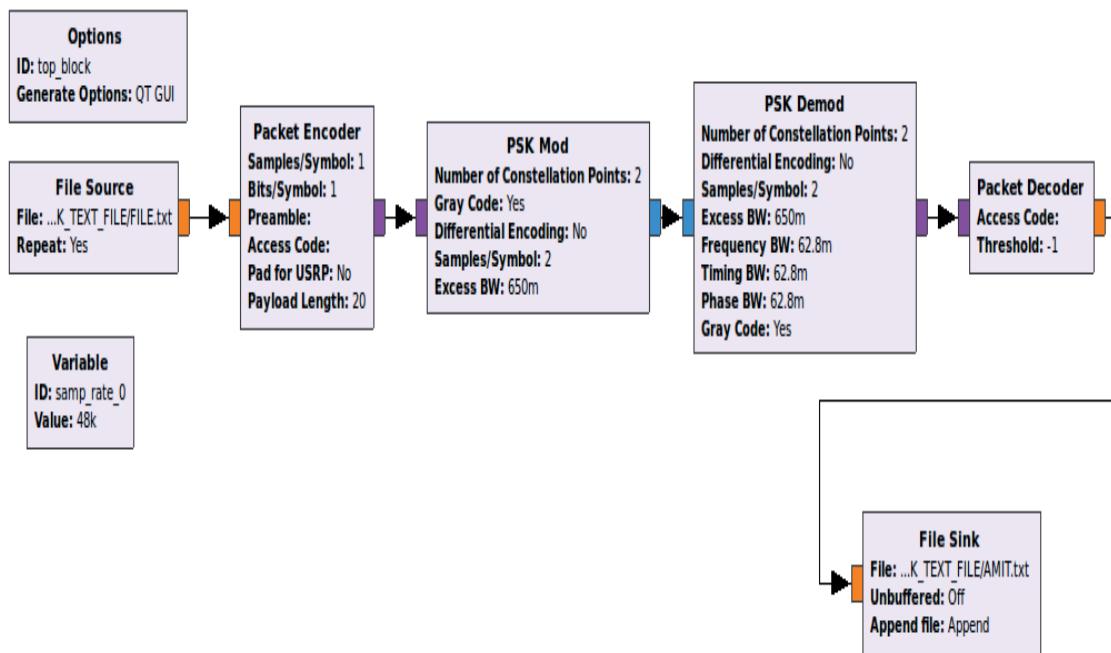


Figure 7.1.1 Flow graph for text message transmission and reception.

The fig above represents a flow graph for the transmission and reception of real time text messages using BPSK modulation schemes. The text file is created with the content mentioned in it and is saved with (.txt) extension as shown in the fig 7.1.2. The text files in transmitted in a repeating mode. The link of this text file is provided in the file source block. The data is then transferred to packet encoder which can store up to 20 bits payload length with preamble and access code attached to it. Each encoded packed has unique access code. Here in this experiment BPSK modulation technique is used with 2

samples/symbol. Once the data is demodulated, the packet is decoded back to data values which are stored in the text file name with (.txt) extension.

Results



Figure 7.1.2. Transmission of Audio signal using Gnu Radio.

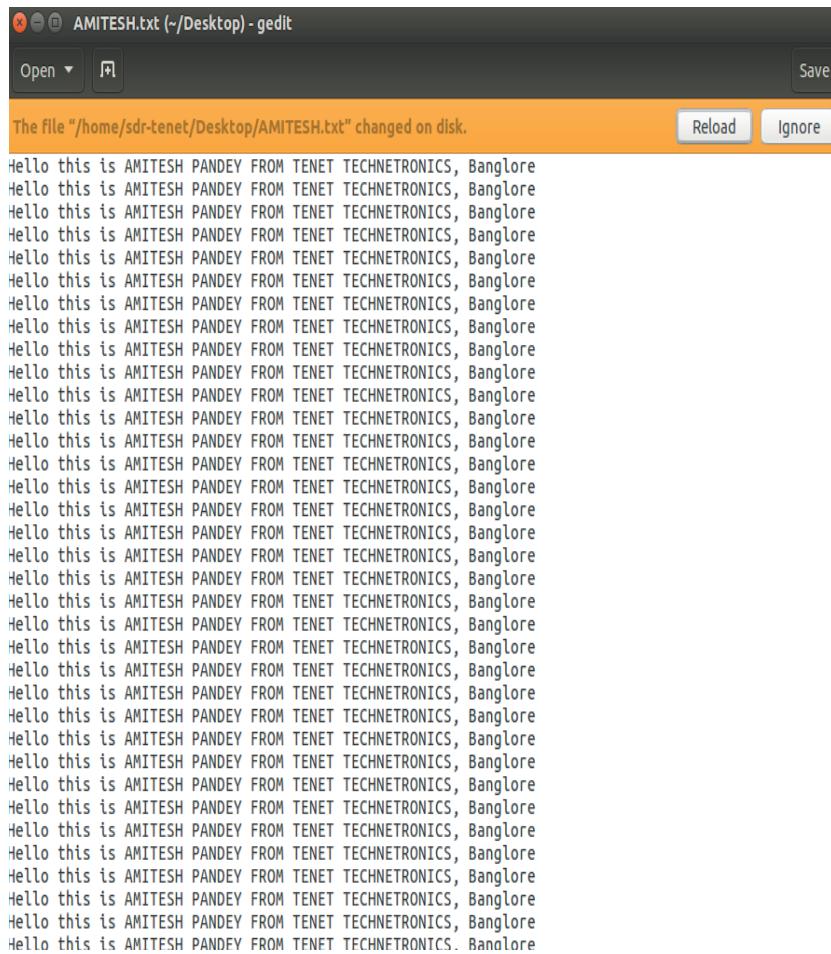


Figure 7.1.3. Received signal

Fig 7.1.2 & 7.1.3 shows a complete transmission and reception of text message signal with no loss of data. The information is transferred in a repeating mode.

Validation using Software Defined Radio (USRP B100)

Gnu Radio flow graph using SDR

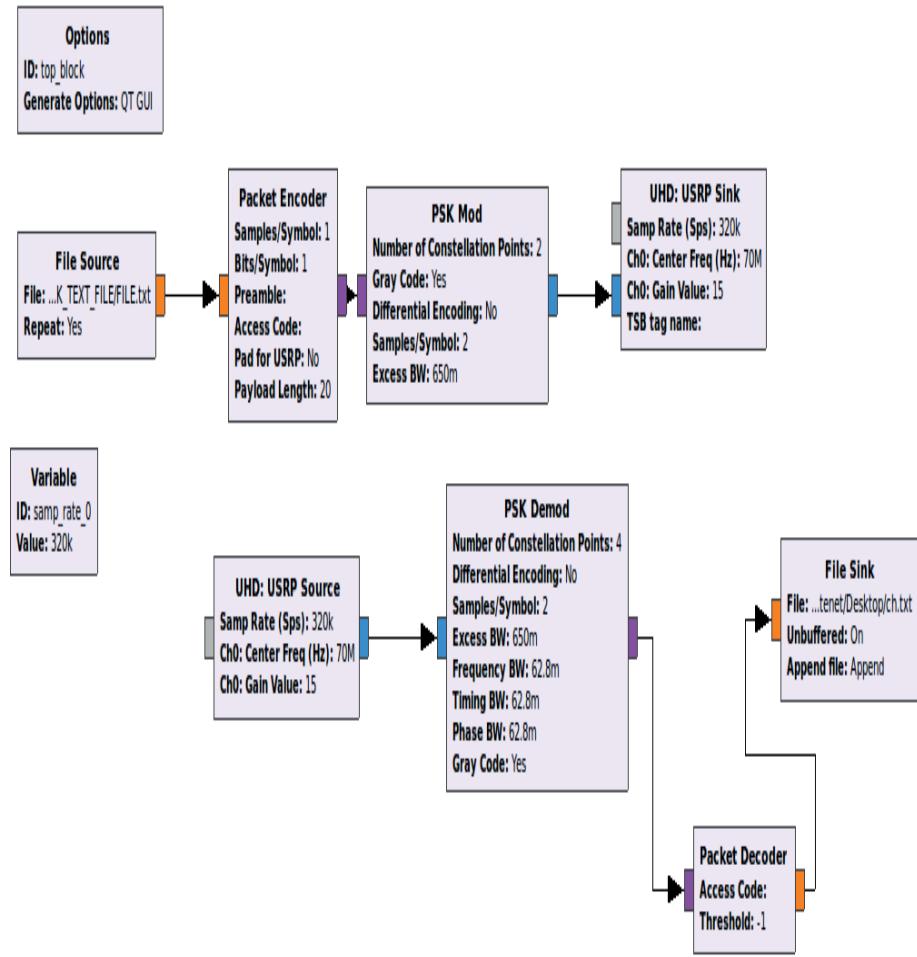
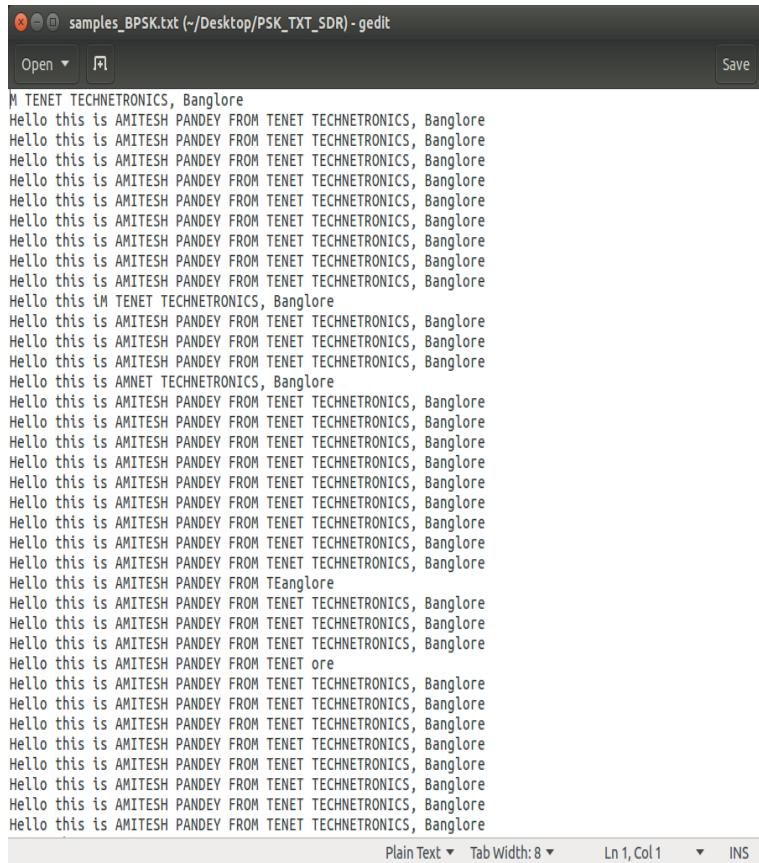


Figure 7.1.4. Flow graph of transmission and reception of text message using SDR.

UHD: USRP Sink is used to transmit the information signal with radio frequency 70 MHz. The gain is set as 15 dB, which can be changed as per the requirement. USRP Sink and USRP Source are tuned to same frequency having gain as 15 dB, in order to have synchronization between the USRP sink and USRP Source. Fig 1.5 represents the received information signals delivered while validating using USRP B100 in a repeating mode

mode. It was found that the some information data is lost or misaligned, thus this occur due to the channel medium which can vary as per the scenario.



The screenshot shows a terminal window titled "samples_BPSK.txt (~/Desktop/PSK_TXT_SDR) - gedit". The window contains a large amount of repeated text: "Hello this is AMITESH PANDEY FROM TENET TECHNETRONICS, Banglore". This text is repeated approximately 50 times. The window has standard Linux-style controls at the top and bottom, including "Save", "Plain Text", "Tab Width: 8", "Ln 1, Col 1", and "INS".

Figure 7.1.5. Received signal

Inference

This experiment successfully demonstrated the transmission and reception of text message signal through Gnu Radio platform and a USRP B100. The BPSK modulation scheme is used, which can be changed to other PSK schemes by changing the constellation points.

EXPERIMENT 2

Aim

To implement the transmission and reception of Audio signal using different modulation techniques in Gnu Radio and USRP B100.

2.1 Implementation using BPSK

Gnu Radio Flow graphs

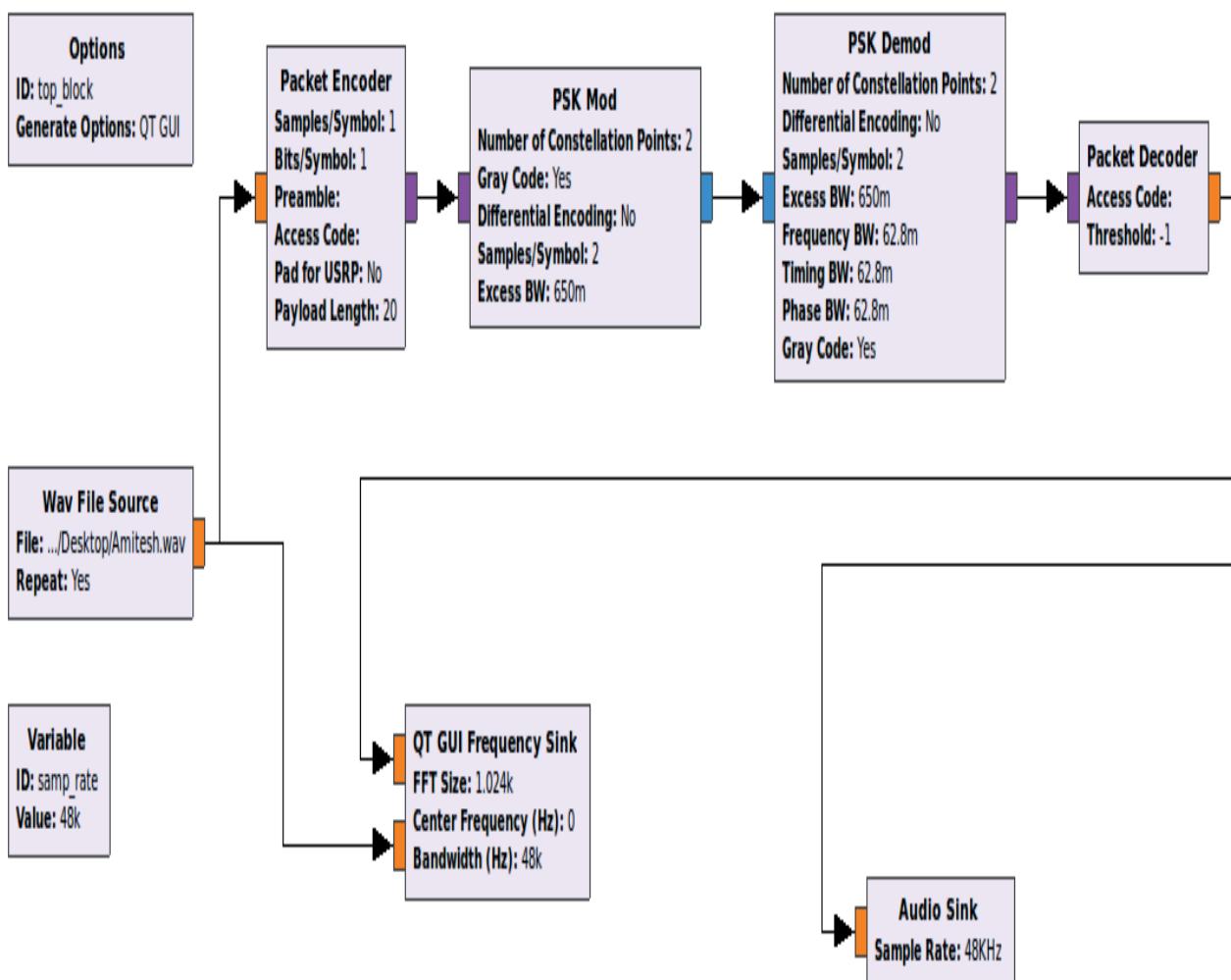


Figure 7.2.1.1. BPSK modulation flow graph using Gnu Radio.

Results

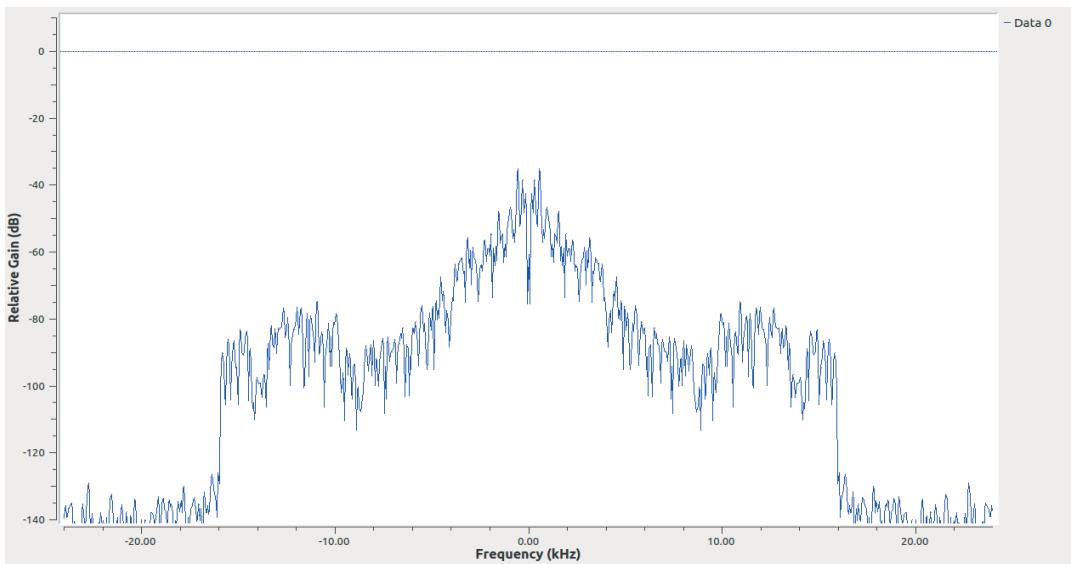


Figure 7.2.1.2 Transmission of Audio signal using Gnu Radio.

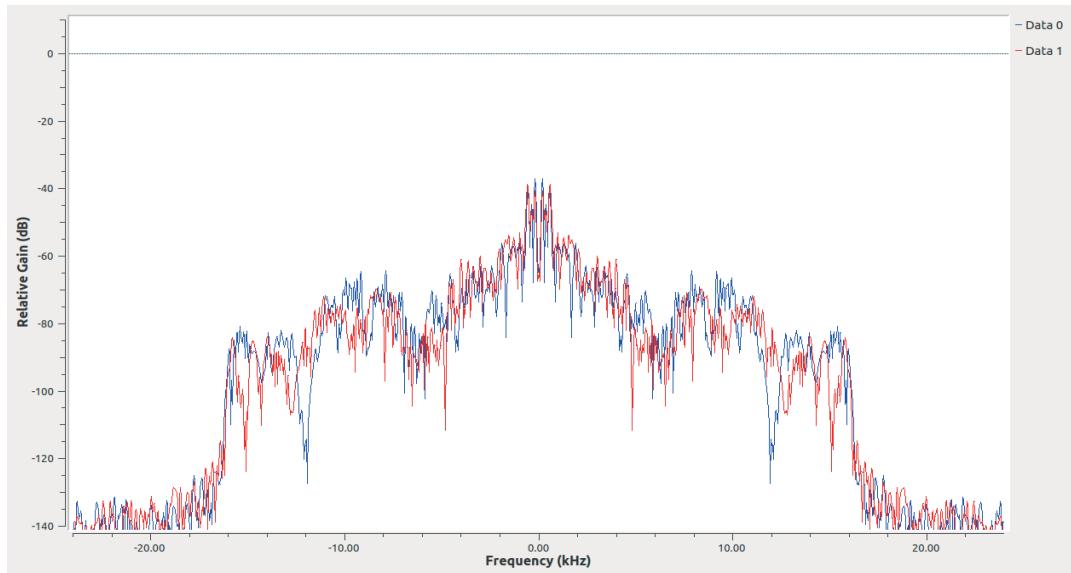


Figure 7.2.1.3. Transmission & Reception of Audio signal using Gnu Radio.

From the above figure it can be observed that the audio signal is perfectly transmitted and received with tuned frequency to 0 KHz. The lower frequency has relatively high gain as compared to high frequency signal. The low frequency signals are information content signal.

Inference

This experiment successfully demonstrates the transmission and reception of Audio signals through Gnu Radio platform .The message signal is received which is not exactly same but identical to the original signal. This is due to the loss of signals occurring, which can be avoided with higher modulation techniques. BPSK modulation techniques has low quality of signals with low bit error rate.

2.2 Implementation using QPSK

Gnu Radio Flow graphs

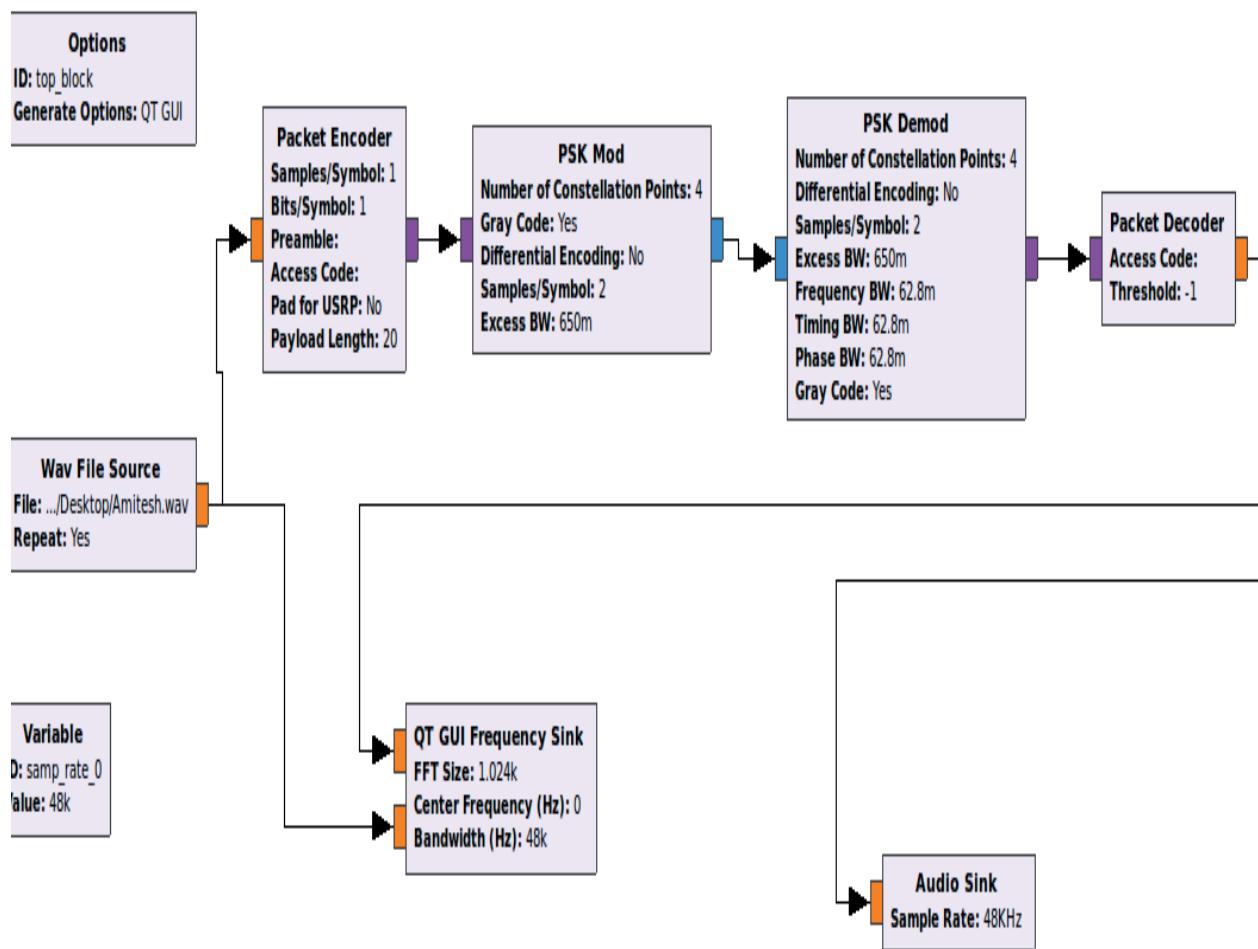


Figure 7.2.2.1. QPSK modulation flow graph using Gnu Radio.

Results

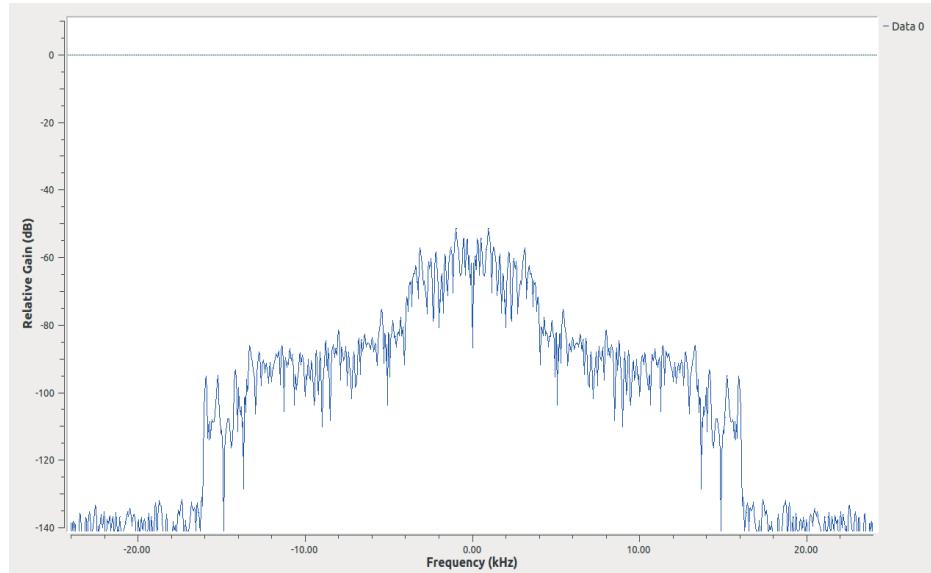


Figure 7.2.2.2 Transmission of Audio signal using Gnu Radio.

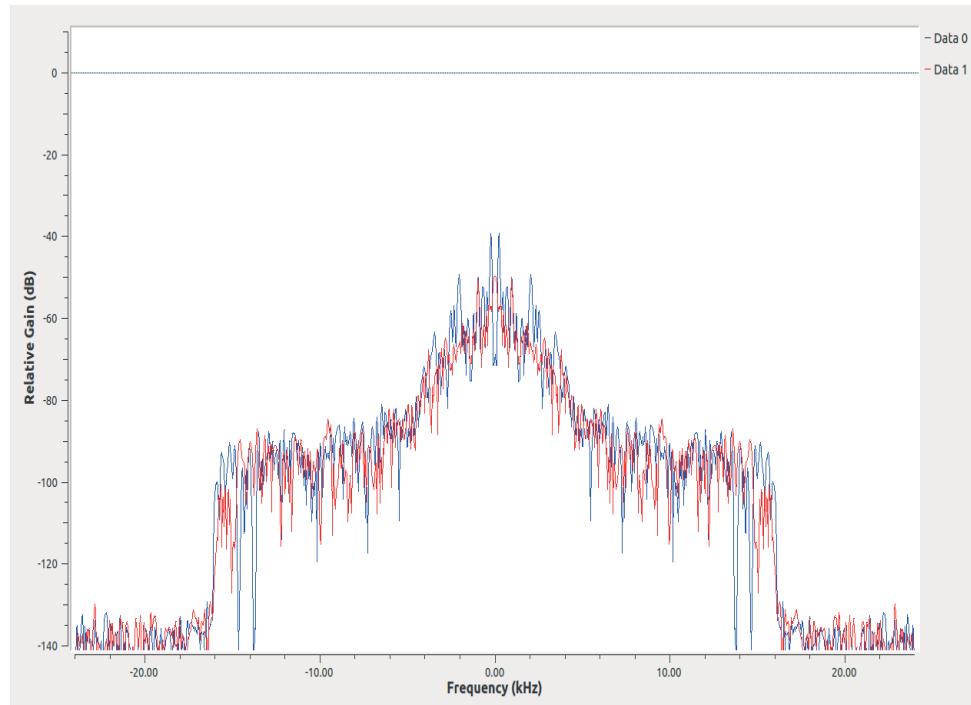


Figure 7.2.2.3. Transmission & Reception of Audio signal using Gnu Radio.

Inference

In this experiment, the implementation and validation of the transmission and reception of Audio signals through QPSK modulation techniques was performed using Gnu Radio and SDR. The message signal received is identical to the original signal. Some loss in signals takes place. QPSK modulation techniques has high quality of signals with high bit error rate as compared to BPSK. QPSK seems to have high data rate with half the Bandwidth as compared to BPSK. It is mainly used in data transmission in radio system.

2.3 Implementation using QAM

Gnu Radio Flow graphs

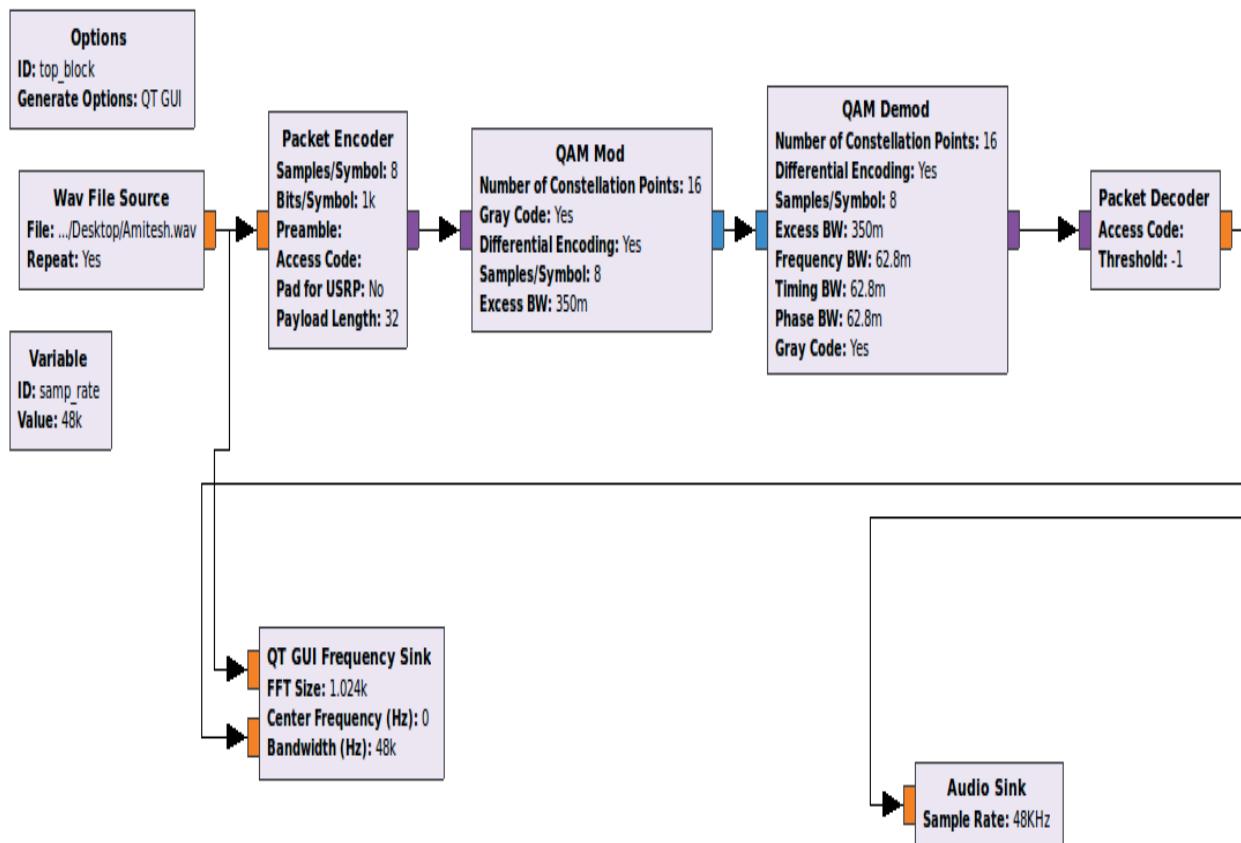


Figure 7.2.3.1. QAM modulation flow graph using Gnu Radio.

Results

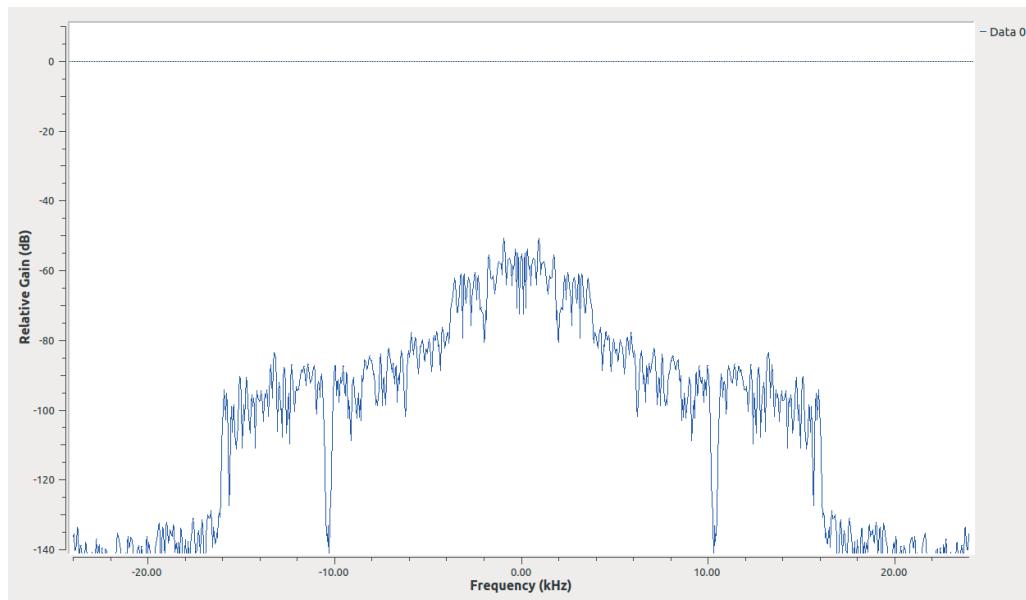


Figure 7.2.3.2. Transmission of Audio signal using Gnu Radio.

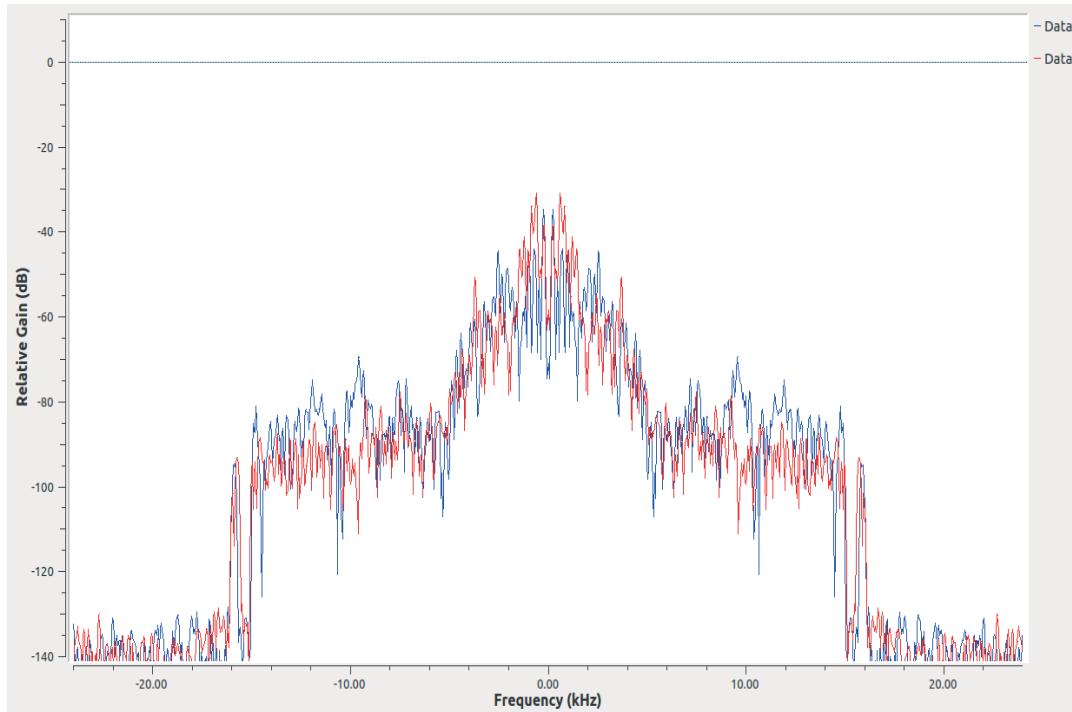


Figure 7.2.3.3. Transmission & Reception of Audio signal in frequency domain using Gnu Radio.

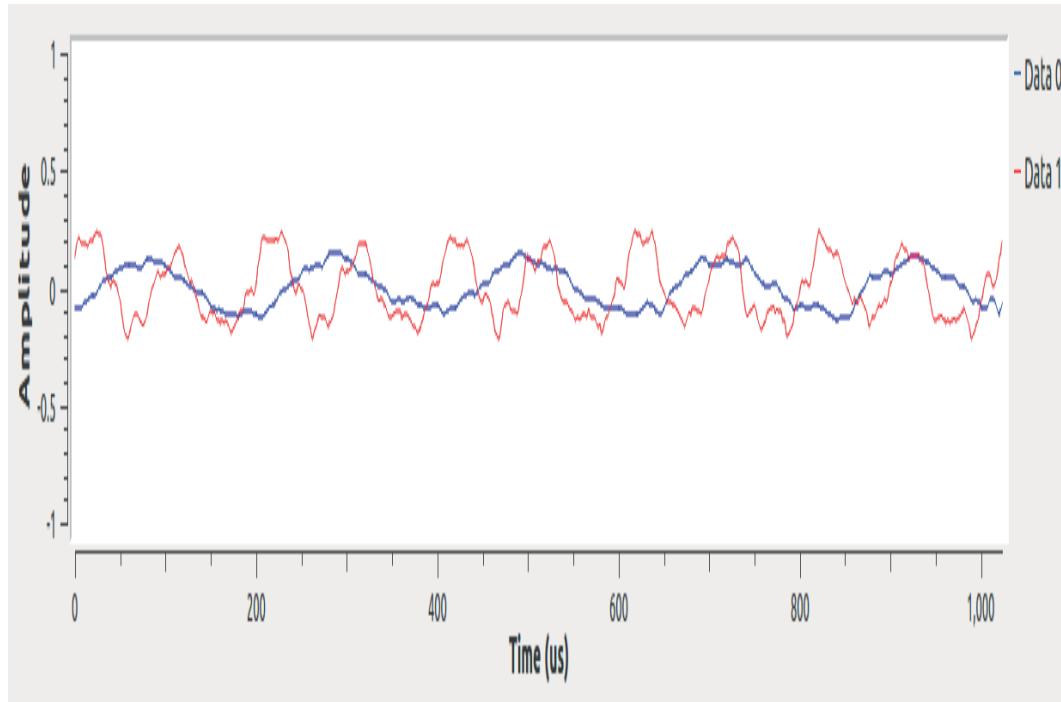


Figure 7.2.3.4. Representation of Transmission and Reception of signals in time domain

Inference

This experiment successfully aims in the transmission and reception of Audio signals through QAM modulation techniques using Gnu Radio platform. QAM modulation techniques has high quality of signals with high bit error rate as compared to PSK. QAM is considered to have the efficient usage of bandwidth. Hence QAM modulation techniques are considered in most of the wireless standards like Wi-Fi, Digital broadcast video, WiMAX etc.

2.4 Implementation using GMSK and validation using SDR

The disadvantage of other forms of phase shift keying is that the sidebands extend outwards from the main carrier which causes interference to other radio communications systems using nearby channels. Hence GMSK modulation has been used in a number of radio communications applications to avoid such problems.

GMSK is abbreviated as Gaussian Minimum Shift Keying. It is derived from MSK modulation scheme which uses the same basic concepts, but applies a Gaussian filter to a signal before its frequency is modulated. A Gaussian filter is a filter which uses a square wave to shape a signal to a more desirable output. GMSK is a type of higher order digital modulation scheme in which the phase of the carrier instantaneously changes with the modulating signal. Gaussian Filter of an appropriate bandwidth is used before the modulation stage. Hence minimum shift keying (MSK) turns to Gaussian Minimum Shift Keying (GMSK).

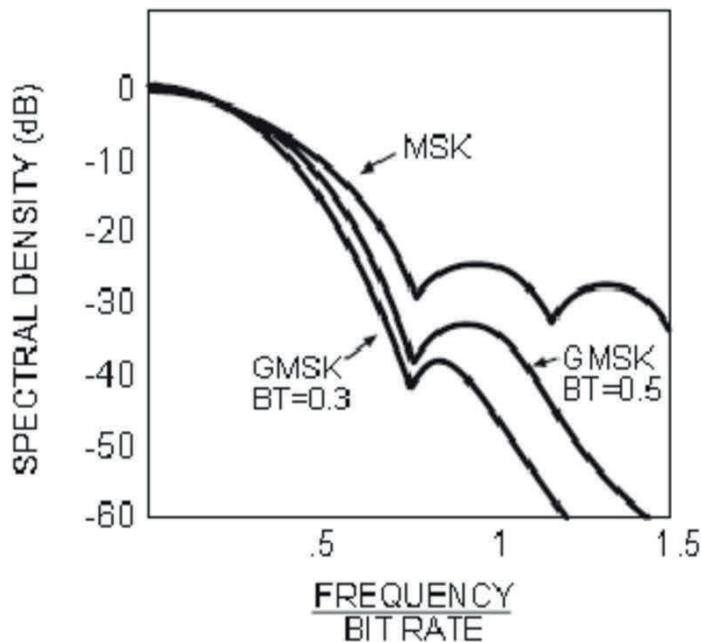


Figure 7.2.4.1. Spectral density of GMSK modulation techniques

While designing a Gaussian filter, it was found that the Gaussian filter is a product of time and bandwidth WT_b . GMSKs power spectrum drops much faster than the MSK's. Furthermore, as WT_b is decreased, the roll-off is much quicker.

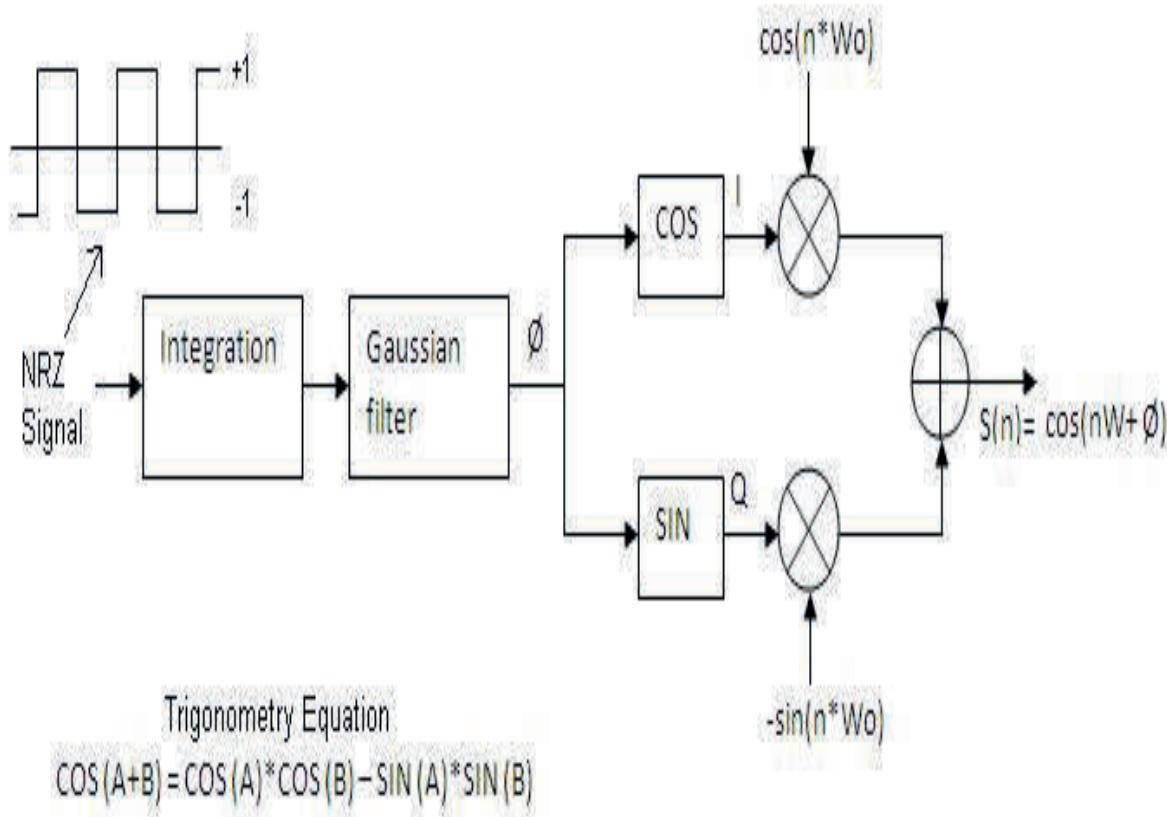


Figure 7.2.4.2 GMSK modulation techniques

NRZ signal is considered as input to the integrator block. The output of Gaussian filter is multiplied by the cos and sin carrier function using mixing block to provide I and Q components. Both the chains are further summed up to give out $S(n)$ which is nothing but a GMSK modulated signal.

The modulated signal is further demodulated and is mixed with cosine and sine carrier signals with the help of a mixer. This output signal is then made to pass through the low pass filter to avoid the unwanted high frequency signals thus allowing only low frequency I and Q components signals through it. The I and Q component is further sent to Derivator to obtain NRZ signals back.

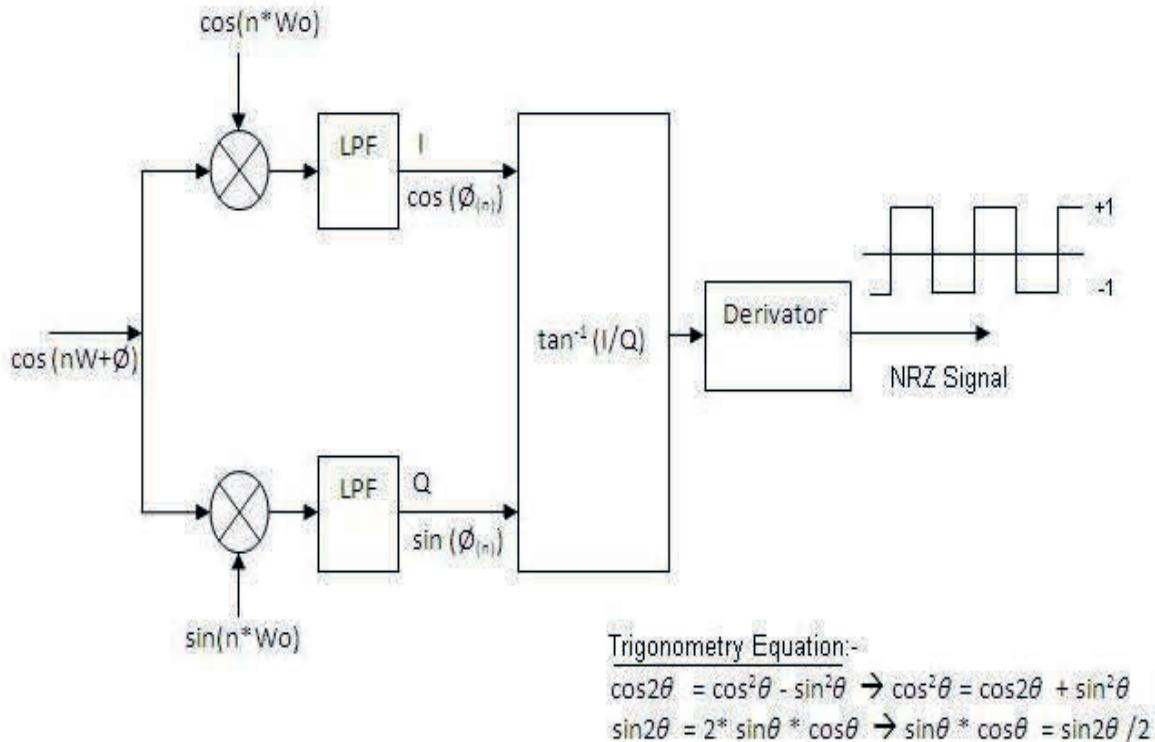


Figure 7.2.4.3. GMSK Demodulation techniques

Advantage of GMSK

1. GMSK spectral efficiency is better than MSK
2. Increases the quality of signals transmitted and received.
3. High spectral efficiency.
4. Perfect transmission and reception of voice modulated signals.
5. Inter symbol interferences is tolerable.
6. Self-synchronizing capability.
7. Good Bit error rate performances.
8. Reduces sideband power.

Disadvantages of GMSK

1. High power consumption.
2. Probability of error is higher than MSK.
3. More complex in designing a system.

Application

1. GMSK is used in wireless communication system for the transmission and reception of audio signals through it.
2. Used in GSM for mobile communication system.
3. Used for GPRS and EDGE system.
4. Used for CDPD packaging network.
5. Used in Radio Broadcasting tower, Bluetooth headset.

Gnu Radio Flow graphs

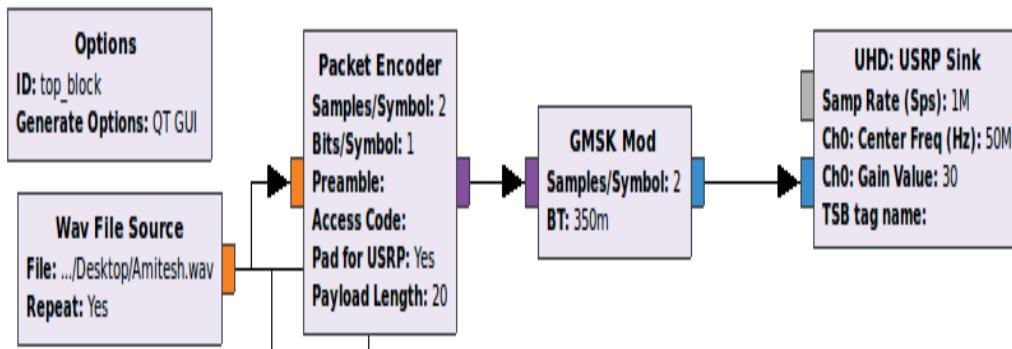


Figure 7.2.4.4 Transmitter side (GMSK modulation) flow graph.

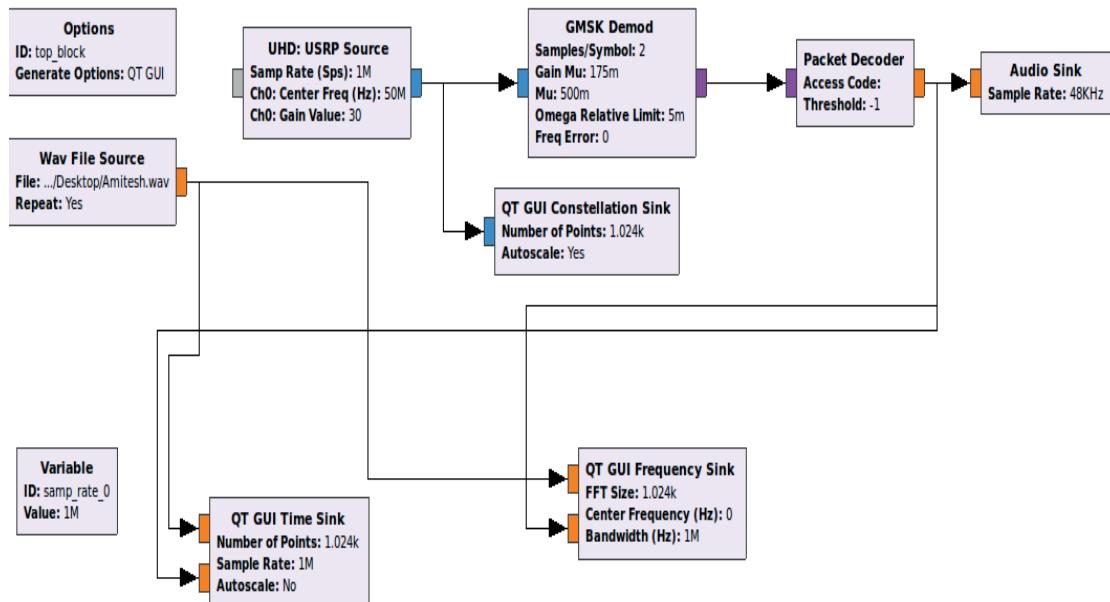


Figure 7.2.4.5. Receiver side (GMSK Demodulation) flow graph.

NOTE:

Since single Transceiver system is used for both transmission and reception of signals, hence in Fig 7.2.4.6 represents loop back system.

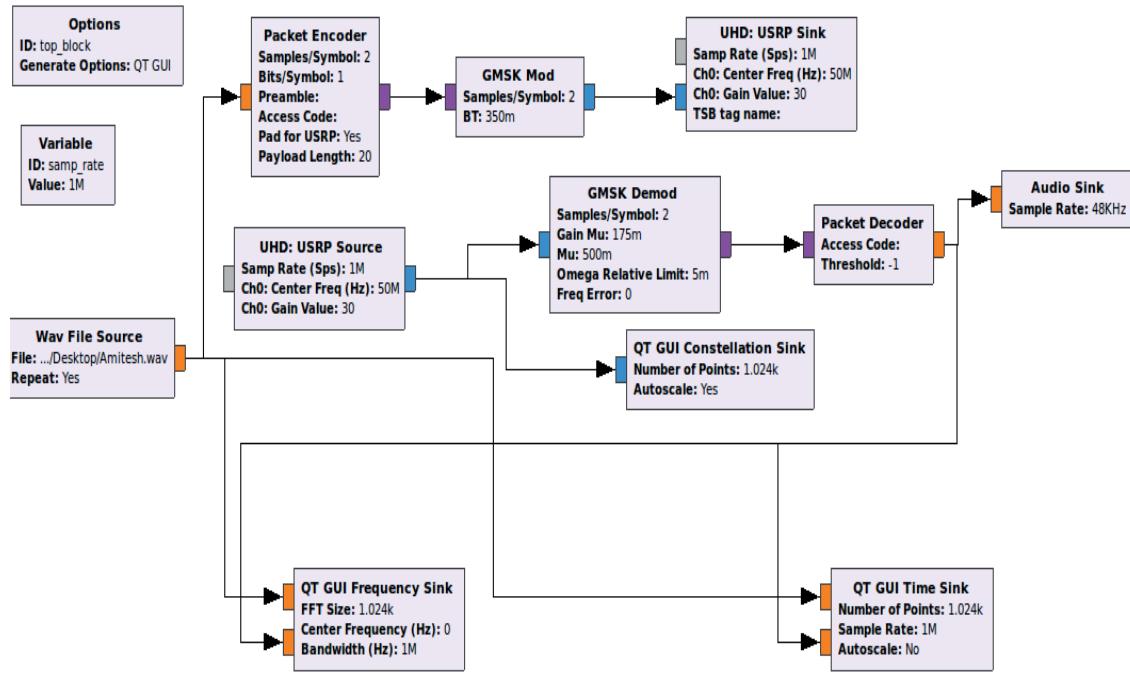


Figure 7.2.4.6. GMSK modulation flow graph using USRP B100.

Results

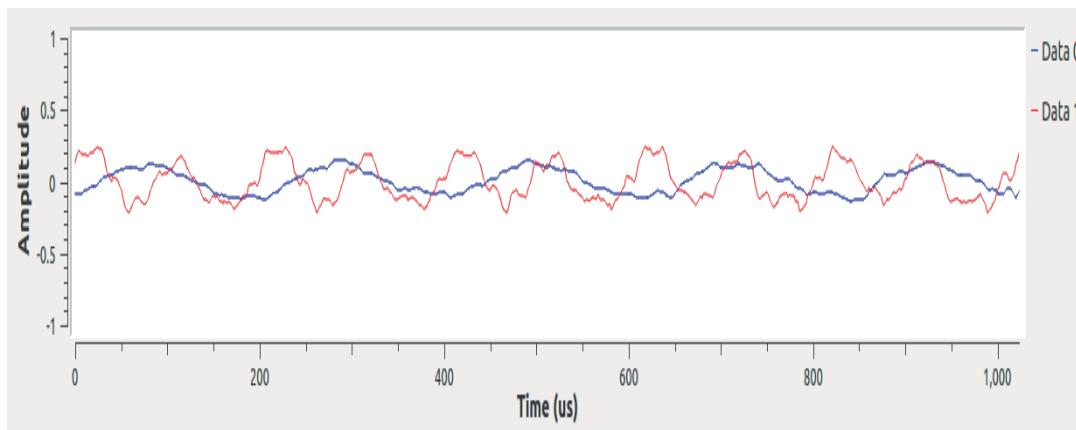


Figure 7.2.4.7. Transmission and Reception of Audio signal in time domain.

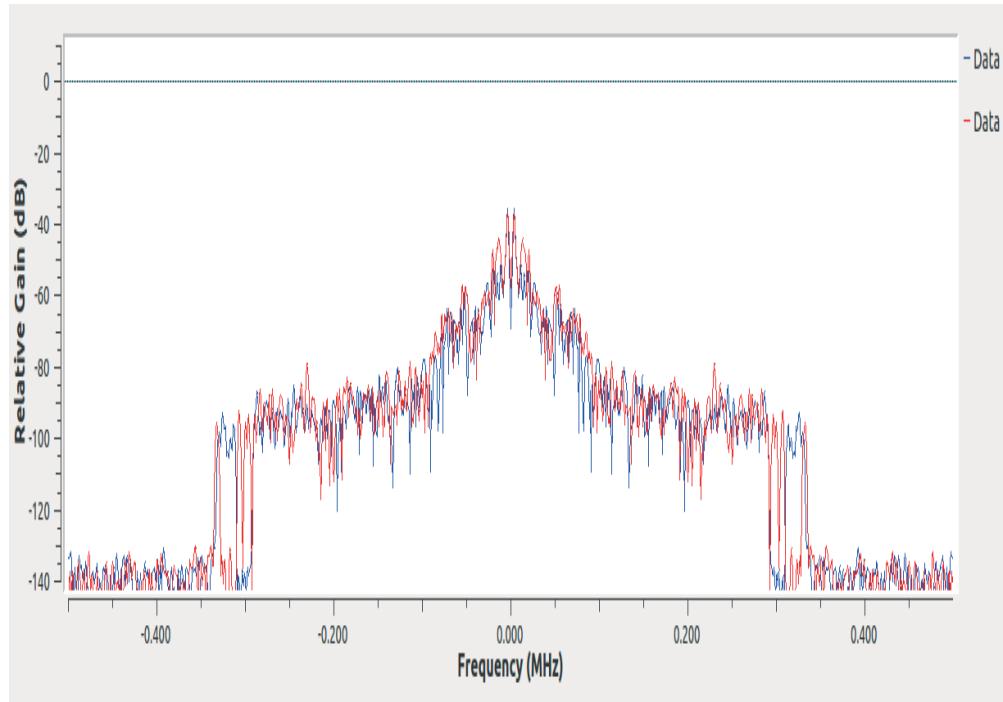


Figure 7.2.4.8. Transmission & Reception of Audio signal in frequency domain

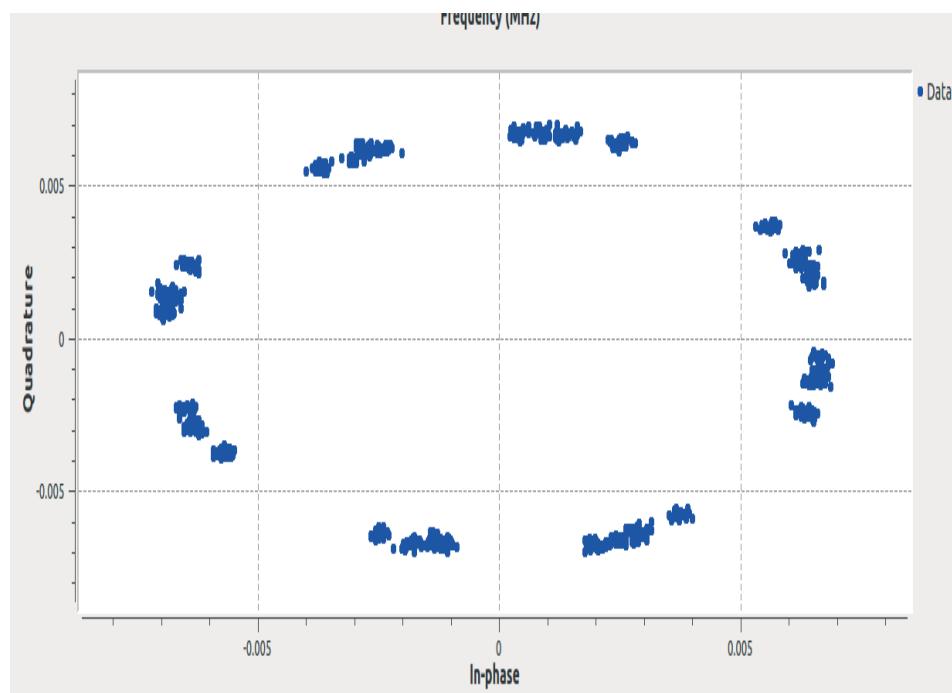


Figure 7.2.4.9. Constellation diagram for GMSK modulation system.

Inference

This experiment successfully demonstrated the transmission and reception of Audio signals through GMSK modulation techniques using Gnu Radio platform and finally validation on USRP B100. A GMSK modulation technique is the higher order modulation scheme which is a derivative of MSK. It provides a high quality transmission and reception of audio signal. Hence GMSK is used in wireless communication application like GSM, GPRS, and Bluetooth etc. Here in the experiment, since Transceiver WBX is used, thus loop back system is considered.

EXPERIMENT 3

Aim

To implement the transmission and reception of VIDEO signal using Gnu Radio and Gstreamer.

Gnu Radio Flow graphs

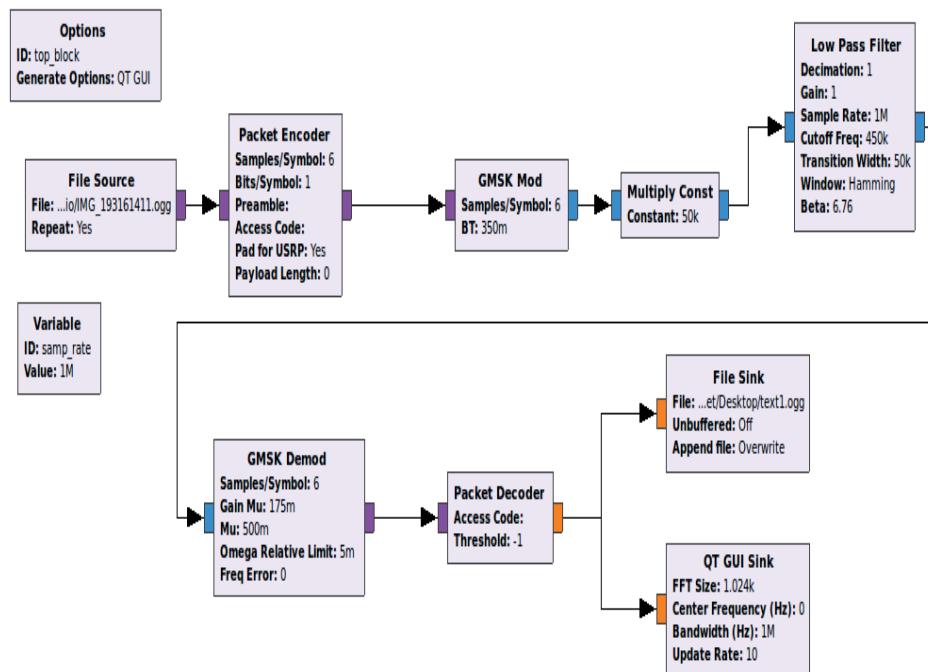


Figure 7.3.1 Flow graph for transmission and Reception of Video signal.

The figure above represents a flow graph for the transmission and reception of real time text messages using GMSK modulation schemes. A video file is created and is saved with (.ogg) extension as shown in the figure 7.3.1. The link of this video file is provided in the file source block which is transmitted in a repeating mode. The data is then transferred to packet encoder with preamble and access code attached to it. The payload length is set as 0 which means by default the block will create the frame length size. Each encoded packet has unique access code. Here in this experiment GMSK modulation technique is used with 6 samples/symbol. Once the data is been demodulated, the packet is decoded back to data values which are stored in the test1 file name with (.ogg) extension. The QT

GUI Sink is used to analyze the parameter in terms of time, frequency, waterfall diagram with also representation in constellation plot.

Results

Fig 7.3.2 represents a frequency domain representation of received video signal. It indicates the response of a system based on the input signals across the range of frequency. The fig 7.3.3 provides a time domain representation of output received signal. In this case the input signal is subjected to the function of time and the received output signal is further analyzed with respect to the function of time.

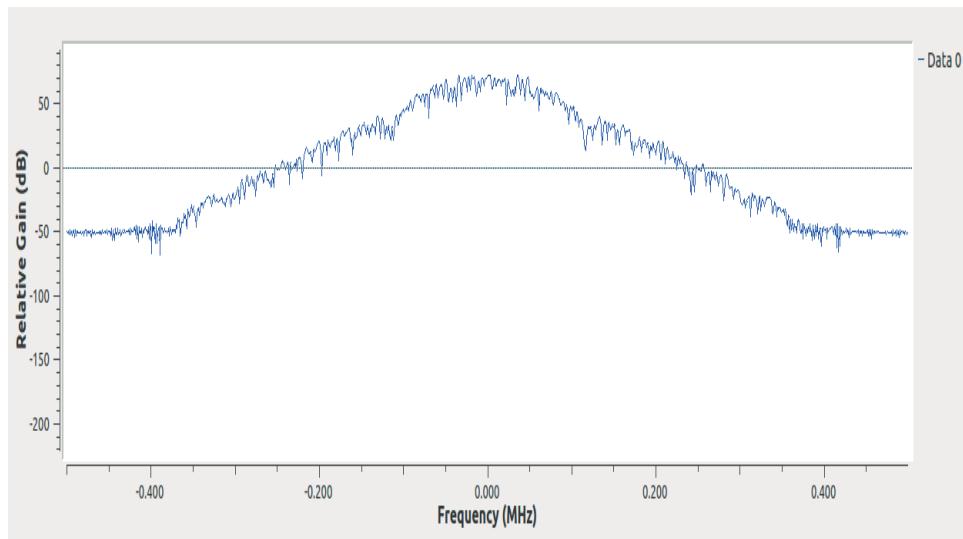


Figure 7.3.2.Output signal in Frequency plot.

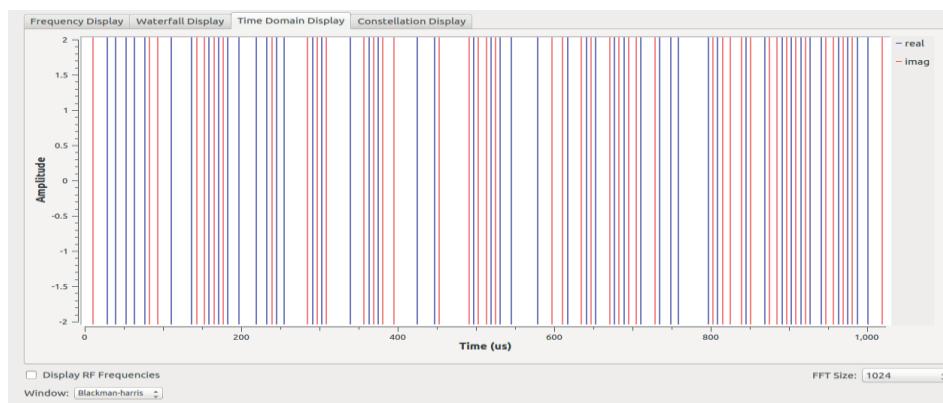


Figure 7.3.3 Output signal in time domain.

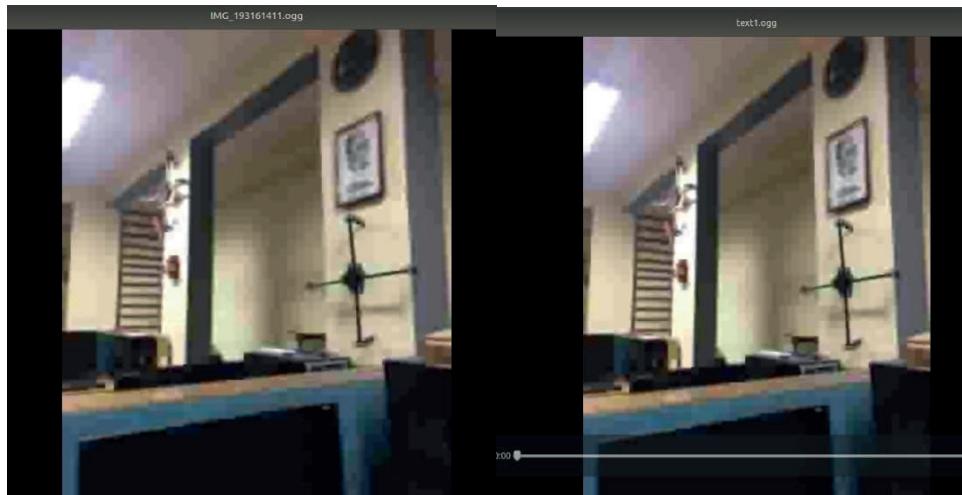


Figure 7.3.4.a) INPUT Video signal 7.3.4.b) OUTPUT Video signal

Fig 7.3.4 shows the transmitted video and the received output video signals in ogg format. Fig 7.3.5 represents the waterfall diagram for the output video signal. The waterfall diagram is a time frequency diagram with frequency on the x axis and time domain on the y axis. It explains the variation in the frequency and time domain signals with change in the input signals (parameters like change in amplitude, audio pitch signals etc changes with reflects in time and frequency domain).

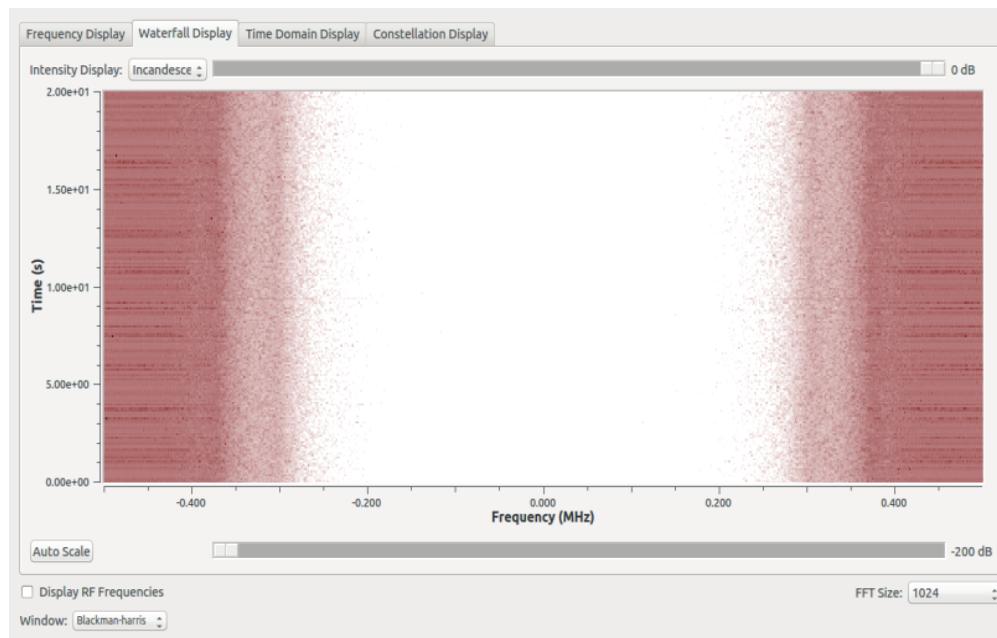


Figure 7.3.5. Waterfall diagram.

The transmission and reception of video is done using the Gstreamer. Gstreamer is a multimedia based pipeline framework which helps in linking all the packet frames created. It builds a system which read files in one format and after processing exports them in other format. Hence making it suitable and compatible for displaying the video.

Installation of Gstreamer

Step 1:

Check whether Gstreamer is available in your system .Hence type commands in terminal as

Gst-launch-1.0

You will get something similar to /usr/local/bin/gst-launch-1.0 which defines the path where Gstreamer is available.

Step 2:

Update the packages

sudo apt-get update

sudo apt-get upgrade

It will take some time.....

Step 3:

Install some packages which are listed below:

sudo apt-get install libgstreamer1.0-0

sudo apt-get install gstreamer1.0-plugins-base gstreamer1.0-plugins-good

sudo apt-get install gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly

sudo apt-get install gstreamer1.0-libav gstreamer1.0-doc gstreamer1.0-tools

Install package to use Gstreamer in python

```
Sudo apt-get install python-gst-1.0 python3-gst-1.0
```

Install the dev packages

```
sudo apt-get install libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
```

```
sudo apt-get install libfontconfig1-dev libfreetype6-dev libpng-dev
```

```
sudo apt-get install libcairo2-dev libjpeg-dev libgif-dev
```

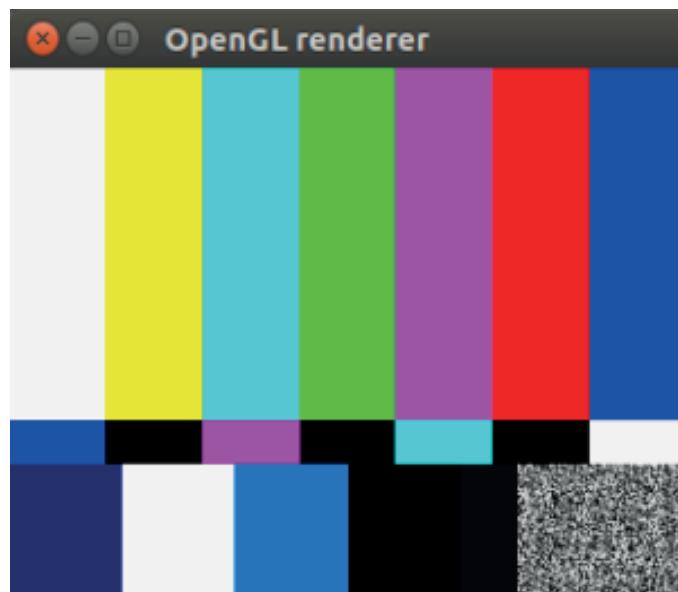
```
sudo apt-get install libgstreamer-plugins-base1.0-dev
```

Step 4:

Now Gstreamer is installed. For verification repeat step 1. Type the command in the terminal to display the video page

```
gst-launch-1.0 videotestsrc ! autovideosink
```

We will get image somewhat like mention below



Hence Now the Gstreamer is ready to stream the videos.

Step 5: (optional)

There are few commands which provide some additional information regarding the parameters.

`gst-inspect-1.0 pulsesink`

It is used to display the bunch of properties for the plugin. The instruction given below will play the sine tone.

`Gst-launch-1.0 audiotestsrc! audioconvert ! audioreample! pulsesink`

`gst -inspect-1.0`

The instruction above will display all the installed plugins and will also mention the count of plugins. For example there are 228 plugins with 1314 feature.

`gst-inspect-1.0 volume`

The instruction above is used to provide the information about the volume

`gst-launch-1.0 autovideosrc device=/dev/ videoX ! autovideosink`

The instruction above is used to live stream of videos from USB webcam

Inference

This experiment provided complete information about the transmission and reception of video signal through Gnu Radio platform and Gstreamer. Gstreamer is used for pipelining the packets frame in synchronization, so that there is no loss in the information signal. The step by step instruction to install the Gstreamer was also presented.

Chapter 8

Introduction to Radar system

This chapter mainly deals with the implementation of frequency modulated continuous wave (FMCW) Radar system using Gnu radio and finally validation on USRP B100. This chapter also provides information about the calculation about the parameters focusing on the range of the target, time required for the target to reach the location etc. In the year 1865, the Scottish physicist James Clerk Maxwell introduced the concept of Electromagnetic field, where he demonstrated that the electric and magnetic fields travels along the space in the form of waves with constant speed of light. This theory of electromagnetic wave signals were further invented by an Italian scientist Guglielmo Marconi. He had demonstrated the first long distance transmission of electromagnetic waves. In the year 1900, Nicola Tesla finally suggested that the reflection of electromagnetic waves can result in identification of location and distance of a moving object. Hence there after many scientists and researchers all over the world worked on the concept of electromagnetic field, which results in the introduction to Radar system.

Radar is abbreviated as

Ra = Radio

D= Detection / Direction

A= And

R= Ranging

Hence by the name itself Radar can be well defined as a system which makes use of electromagnetic wave signal to get the information about the object which is either stationary or moving. The information about the object / target includes range, velocity, time required, Direction etc. The Radar system has been widely explored in various fields like tracking aircrafts, used in defense system and in Air force department etc. Industries are also trying to get the concept of radar in the field of Automobile industries in order to identify the obstruction in front of the vehicle.

How radar works?

A radar system has a transmitter section which emits radio waves called electromagnetic waves signals in all the direction. When these electromagnetic waves come into contact with any target, they are converted into echo signals which are usually reflected or scattered in many directions and a portion of which is received by a receiver to further analyze the signals and process them further to create inference about the direction/speed of the obstacle .

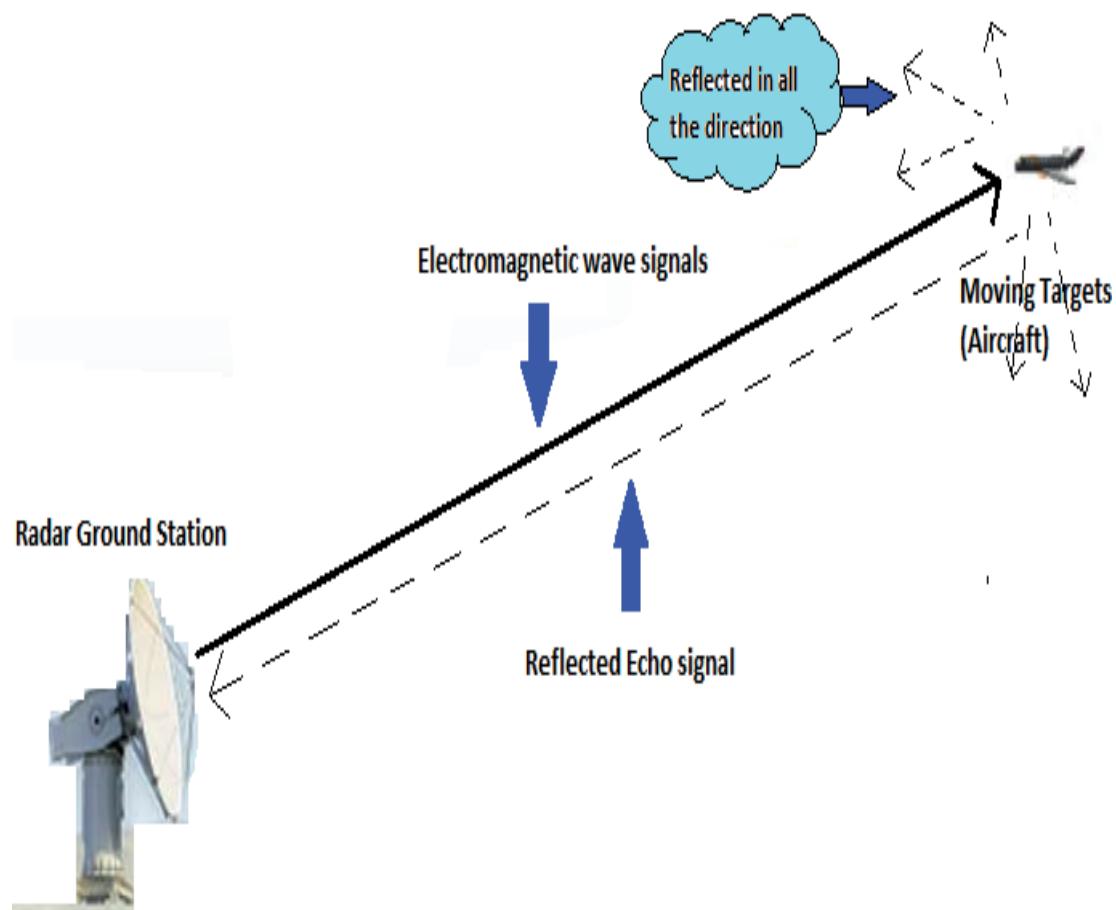


Figure 8.1 Working of Radar.

Fig 8.1 represents the working model of Doppler pulse radar. The continuous wave oscillated signal is given as an input to the power amplifier, where with the help of pulse modulator, the continuous wave input signals are allowed to pass in terms of discrete pulse signals with an alternate ON- OFF pattern. There is a duplexer, which is a type of Transreceiver, through which signals can be transmitted and received alternately. If the

transmission section of radar is ON, then the receiver part of radar is disconnected. Once the electromagnetic signals hits the target, the echo is generated which is reflected back to the receiver end of radar via duplexer. In such case transmitter end of radar is made disconnected. Hence the system is saved from the damage.

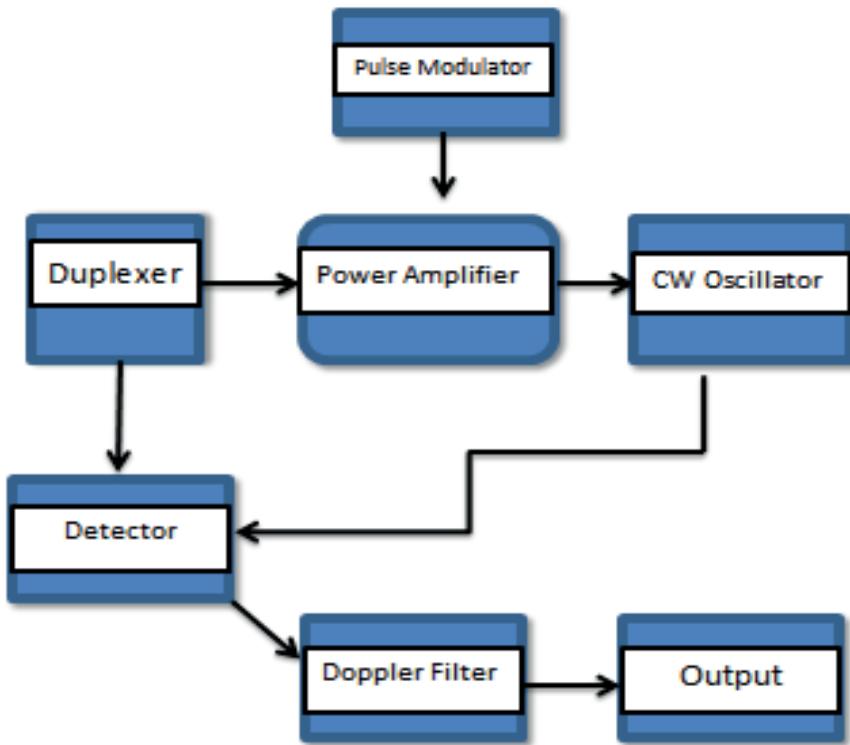


Figure 8.2 Working Model of Radar

There is a detector at receiver end which will compare the changes in the received echo signal with respect to the reference signal and hence the carrier signals is obtained with a Doppler frequency shift. When the target is not stationary like aircraft, in such case using single pulse of echo signal won't be sufficient to track the exact location of the Aircraft. At every instance there will be change in the frequency that is expected to take place. Hence it results in the Doppler Effect which is defined as the instantaneous change in the frequency, when a sound source moves either toward or away from the listener. Doppler frequency shift is mathematically formulated as

$$f_d = \frac{2*v}{\lambda}$$

Where f_d = Doppler frequency, V = speed of Aircraft, λ = wavelength

The Doppler filter will filter out the unwanted carrier signals and will allow only Doppler Signals to pass through it. These signals are further analyzed based on the signal processing techniques in order to identify the information about the target. In this chapter, calculation of range of a target from the radar ground station is considered. In order to calculate the moving targets, multiple pulses are transmitted in a repeating pattern, so that the exact information about the moving target can be identified at every instance. This is also termed as pulse repetition frequency (PRF). The range of the target can be calculated using a mathematical formula mention below:

$$\text{Range} = \frac{v_0 * t}{2}$$

Where v_0 = speed of light = $3 * 10^8$ m/sec, t = measured running time

Application of Radar

1. In air defense it is used for target detection, target recognition, weapon control and Identifying enemy locations in map.
2. The Air Surveillance RADAR is used to detect and display the aircraft's position in the airport terminals.
3. It is used in Air traffic control department.
4. In missile system to guide the weapon.
5. It is used to guide the aircraft to land in bad weather using Precision Approach RADAR.
6. It is used as a Remote sensor to observe weather or planetary positions and monitoring sea ice to ensure smooth route for ships.
7. It is used in Ground traffic control in order to identify the speed of a vehicle.
8. It is used to guide the space vehicle for safe landing on moon.
9. It is used to detect and track satellite.
10. It is used in the missile system to guide the weapon.

Gnu radio Flowgraph (for single delay)

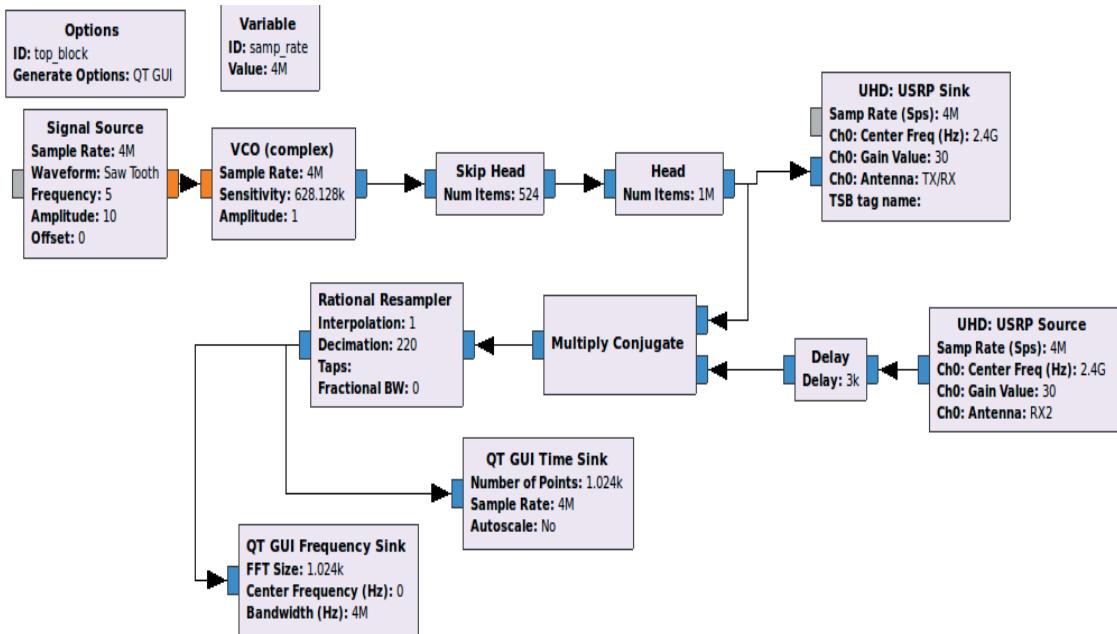


Figure 8.3 Gnu radio Flowgraph for single delay (single Target)

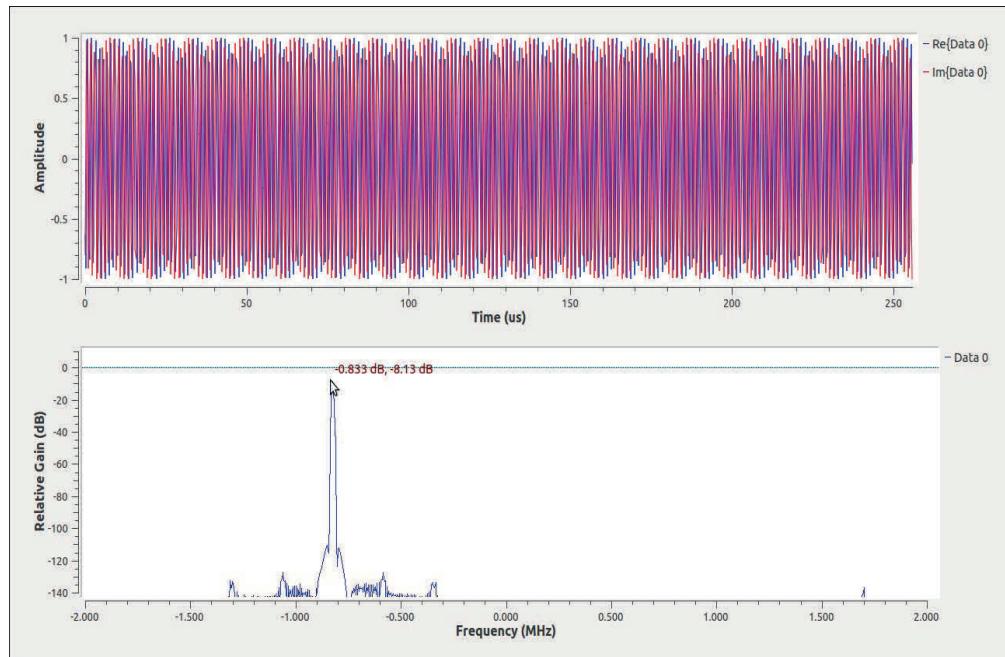


Figure 8.4 Single delay (single Target)

From the fig 8.4 it can be identified that, the single beat indicating a single stationary target is obtained with frequency $f=0.85\text{MHz}$ (Ignoring the negative sign) from the reference. Hence the calculation is as follows

Example

Given: $f = 0.85\text{MHz}$ (from the figure 8.4)

$$\text{Delay in time} = \frac{1}{\text{freq}} = \frac{1}{0.85 \text{ MHz}} = 0.0117 * 10^{-6} = 11.7 \text{ Nsec.}$$

$$\text{Range is calculated as } R = \frac{3 * 10^8 * 11.7 * 10^{-9}}{2} = 1.7 \text{ Km.}$$

Hence the target is at 1.7Km from the radar station on the ground level.

Gnu radio Flowgraph (for multiple delays)

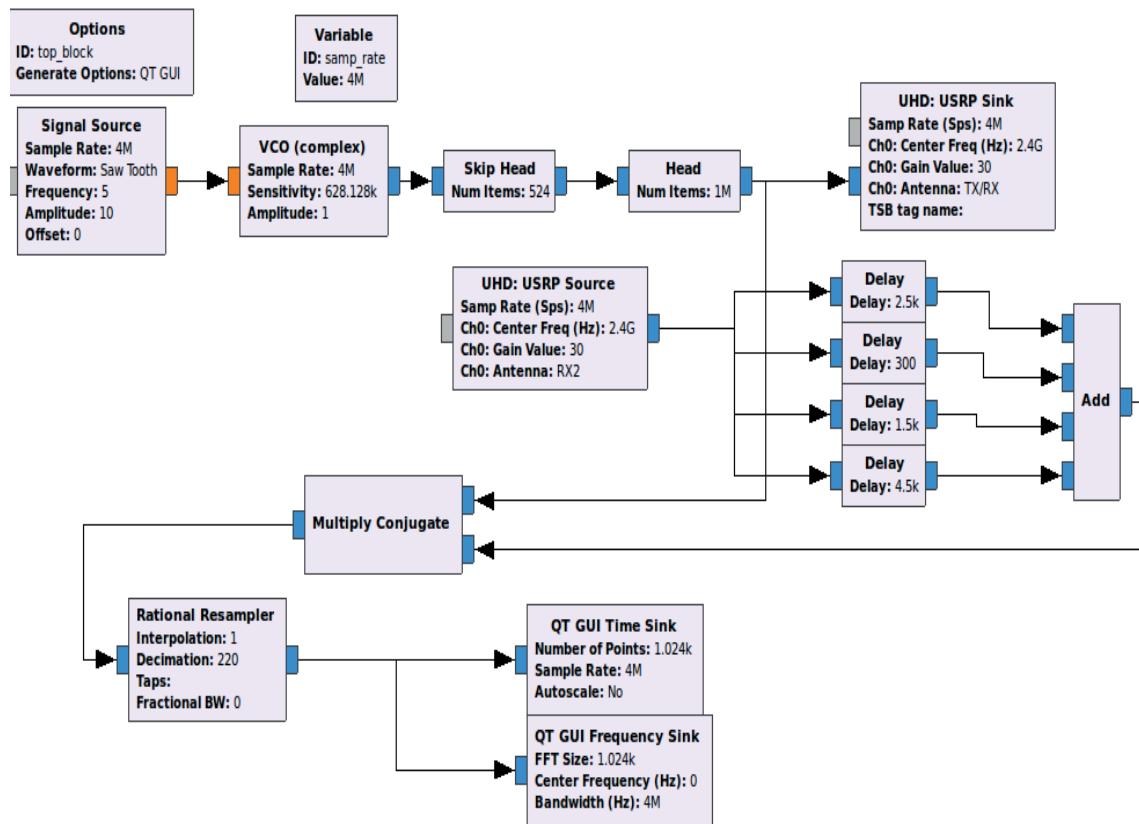


Figure 8.5 Gnu radio Flowgraph for Four delays (Four Targets)

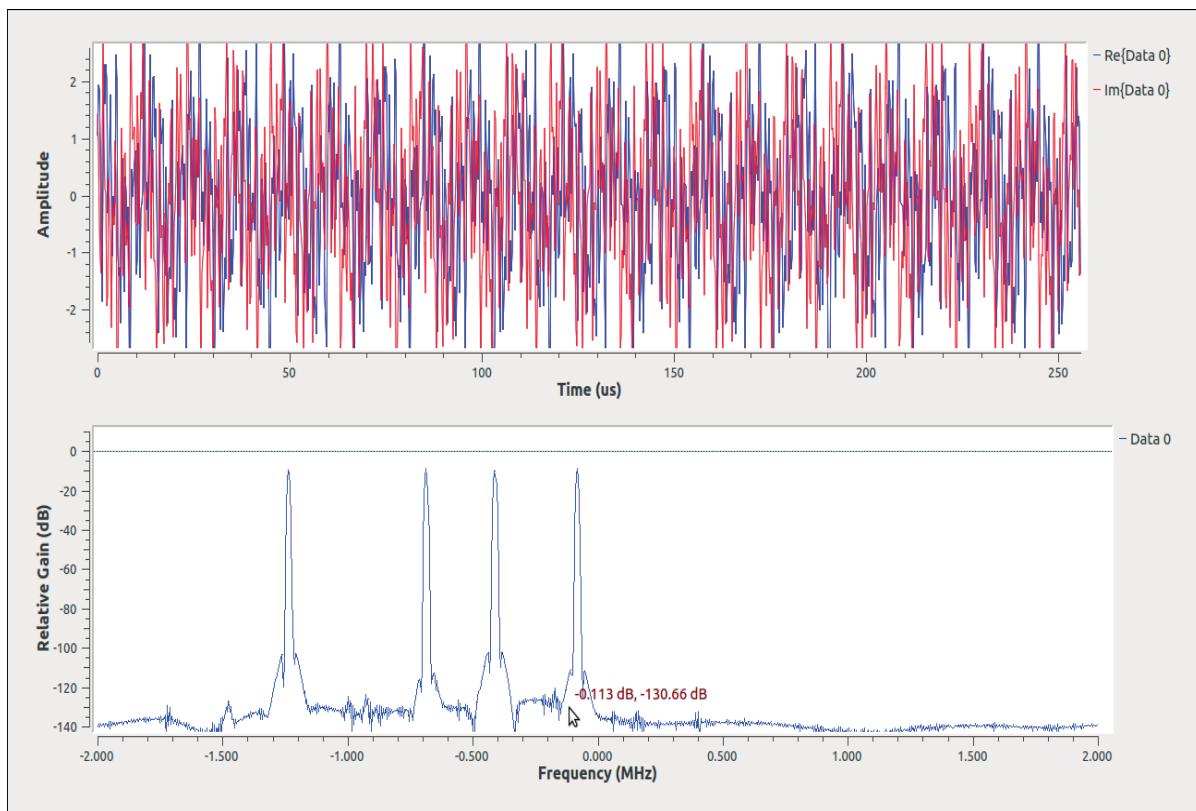


Figure 8.6 Four delay (Four Targets).

The figure 8.6 represent the four beats with frequency $f_1=0.2$ MHz, $f_2= 0.4$ MHz, $f_3= 0.7$ MHz, $f_4= 1.2$ MHz. Thus representing four targets at some distance .From the range formula mentioned above, the range of a target is calculated and is drafted in the table

Table 14: Analyzing the range of targets

Index	Targets frequency	Time taken	Range
1	0.2MHz	50Nsec	7.5Km
2	0.4MHz	25Nsec	3.7Km
3	0.7MHz	14.3Nsec	2.14 Km
4	1.2MHz	8Nsec	1.2Km

Inference

From the table mentioned above, it can be observed that if the frequency of echo signals from the target increases, then it result in decrease in the time taken by the target to reach the radar station because the speed of the target is high. Hence the range of target is less in that case (I.e. the target is somewhere near to radar station).

Chapter 9

OpenBTS –“Build your own 2.5 G Cell Phone Network”

Introduction

There are many isolated geographical areas like the dense forest, hilly areas, on the highest peak, rural areas etc on earth, where people do not have any telephone landlines or any sort of mobile network to communicate with others. But they do have an internet connection via satellite or through Wi-Fi. Hence in such scenario, OpenBTS comes in to the picture where using OpenBTS one can build their own mobile network for their basic communication like voice call and text messages.

OpenBTS which is abbreviated as Open Base Transceiver System is a type of software based GSM access point, allowing GSM-compatible mobile phones to establish the session call in Voice over IP networks. It basically converts the internet services to a mobile network (probably 2.5G) and distributes the generated network across a large geographical area. Any GSM phone can connect and use voice services or SMS for their communication purpose. Hence OpenBTS is defined as a transparent source gateway between the mobile and the VoIP network.

In this chapter, we will discuss in detail about the architecture of OpenBTS and how is it different from the Traditional GSM structure. This chapter also deals with the installation of a base operating system, and development environment setup, as well as actually compiling and installing the components that compose the OpenBTS software.

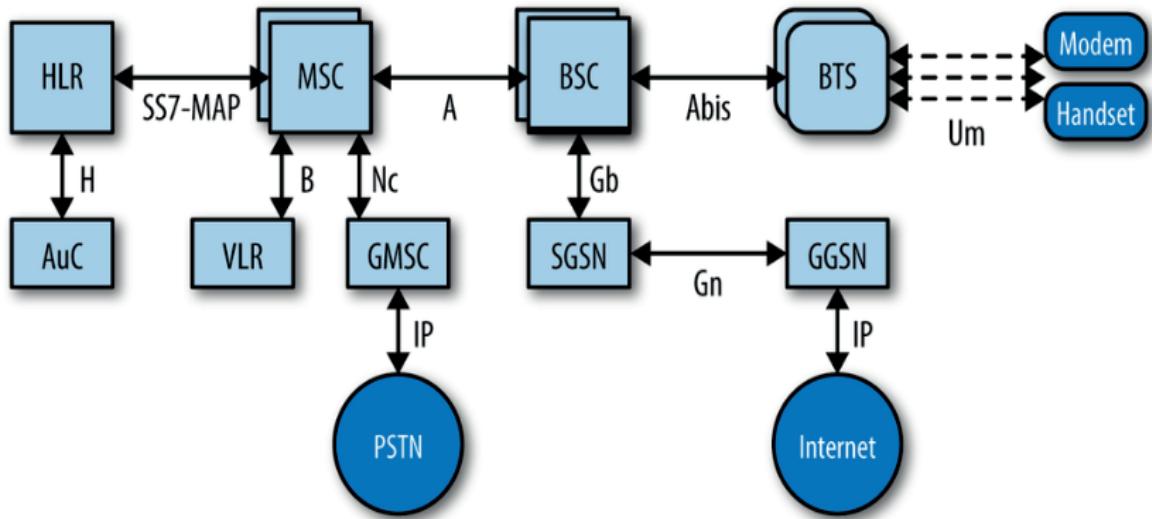
For more detail about the OpenBTS refer to the reference book

“Getting started with OpenBTS “by Michael ledema.

Or else refer the link mention bellow:

<http://openbts.org/site/wp-content/uploads/ebook/Getting Started with OpenBTS Range Networks.pdf>

Architecture of GSM standards



A = A Interface

Abis = Abis Interface

AuC = Authentication Center

B = B Interface

BSC = Base Station Controller

BTS = Base Transceiver Station

Gb = Gb Interface

GGSN = Gateway GPRS Support Node

GMSC = Gateway Mobile Switching Center

Gn = Gn Interface

H = H Interface

IP = Internet Protocol

HLR = Home Location Register

MSC = Mobile Switching Center

Nc = Nc Interface

SGSN = Serving GPRS Support Node

SS7-MAP = Signaling System 7 Mobile Application Part

Um = GSM Mobile Air Interface

VLR = Visitor Location Register

Figure 9. 1. Architecture of GSM Standard.

The figure above describes about the architecture of GSM standard. When the mobile users (i.e. modem, handset) request for the call session, the user will be connected to BTS (Base Transceiver system) of the nearby cell for the unique frequency spectrum. The request is further transferred to BSC (Base station controller) which controls the multiple BTS. The request is further transferred to MSC, Which is the heart of GSM network. The MSC performs the switching function of the system by controlling calls to and from other telephone and data systems and thus the request signal finally reaches PSTN (Public switch telephone network) . Hence it takes a long process to reach the PSTN. Therefore the latency period in GSM (2G network was very high). For more information about the GSM standards one can refer the book “Wireless communication, principles and practice “by T.S Rappaport, Prentice Hall.

The OpenBTS architecture is simplified as compared to the GSM Standards and is represented as below:

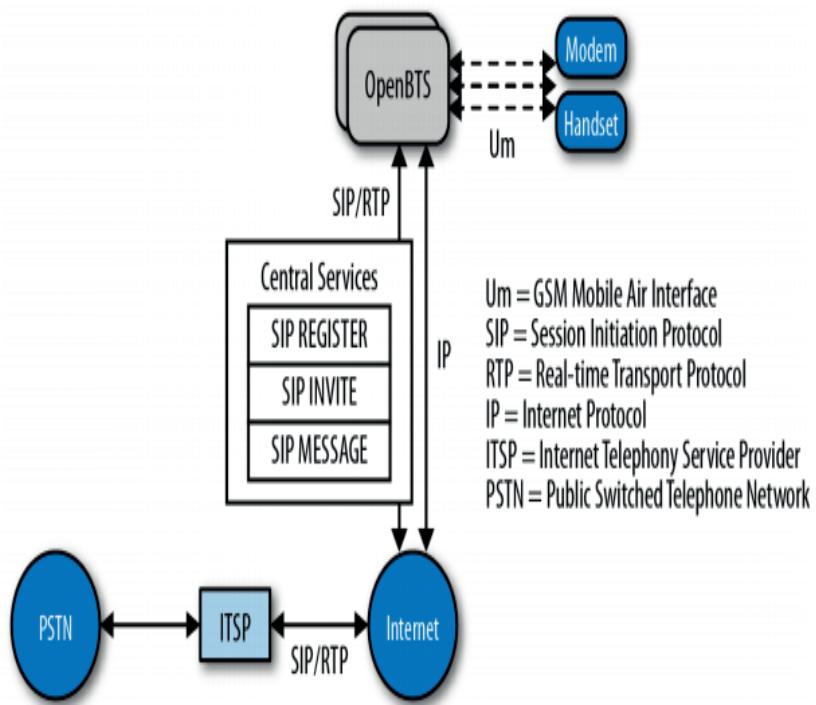


Figure 9.2. Architecture of OpenBTS.

In this architecture, the mobile user (i.e. modem, handset) is connected with the OpenBTS which is installed in the host system. Using OpenBTS system the mobile user system can be directly connected to PSTN (Public switch telephone network) with the help of internet services. OpenBTS is installed in the system .The OpenBTS is further connected to internet using SIP (Session initiation protocol) and RTP (Real time transport protocol). With the help of Internet telephony service provider, The PSTN can be accessed. Once the 2.5G network is created, the signals are transmitted through software Defined Radio (USRP B100). Hence the complicated architecture of GSM structure is simplified by the OpenBTS.

SIP/RTP

It is a type of protocol which helps in sending voice over Internet Protocol (VoIP) by connecting dedicated call session with the help of OpenBTS.

SIP

1. It is abbreviated as Session Initiation Protocol.
2. It is used for establishing a session (call).
3. IP address and port information is exchanged.

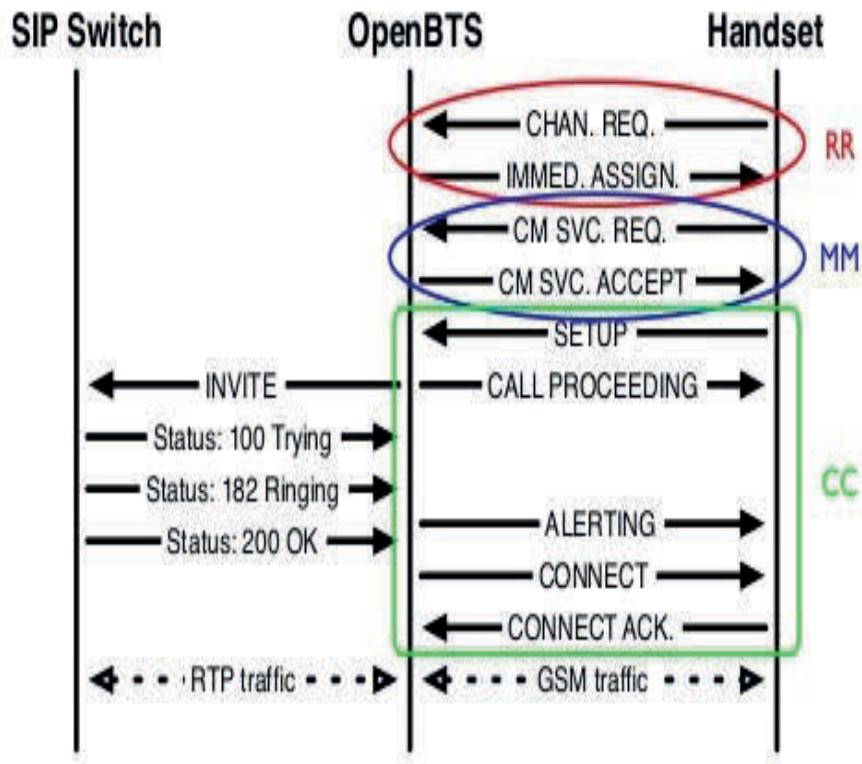


Figure 9.3. Procedure of SIP/RTP

RTP

1. It is abbreviated as Real time transport protocol.
2. After SIP established a session, the protocol is used for exchanging voice packet

RTCP

1. It is abbreviated as Real time control protocol.
2. It monitors the transmission of Statistic and Quality of Services (QoS).

Fig 9.3 represents the flow of SIP/RTP in the transmission and reception of call session. Once the mobile user or handset request for the call session, the channel is immediately assigned to the handset by the OpenBTS. Once the secured channel link is established between the OpenBTS and handset, RTCP will ensure the Quality of Services (QoS). The OpenBTS gives an invitation to SIP Switch to initialize the call session between the two mobile users. Fig 9.3 represents a standard indication of status which is as follows:

1. Status: 100 → indicating the attempt to connect the call session.
2. Status: 182 → ringing the call.
3. Status: 200 → indicate that the call is picked up and a secured channel link is established. Hence the two mobile users are now ready to transfer their information.

Sometime due to the heavy traffic congestion, the users have to hold for some time in order to get the secured call session, which results in RTP traffic condition. Hence OpenBTS further transfers the same message to the mobile user, resulting to GSM traffic.

Installation

Building and Installing OPENBTS on UBUNTU 16.04 operating system. It is applicable for use with USRP B200, B210, and B200MINI.

The steps to get it set up and running are documented further:

1. Install Ubuntu 16.04.2 (64-bit).

Build and install UHD 3.9.6.

Follow the App Note below.

[https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_\(UHD_and_GNU_Radio\)_on_Linux](https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux)

2. Create working folder.

Run the following command on following path directory where memory space is empty:

```
mkdir /home/amitesh/workarea
```

```
cd /home/amitesh/workarea
```

3. Install the UDEV rules.

Only do this if you have a B200, B210, B200mini, and B205mini.

Reference:

https://files.ettus.com/manual/page_transport.html#transport_usb_udev

Run:

```
cd <path_to_UHD_repository>/host/utils
```

```
sudo cp uhd-usrp.rules /etc/udev/rules.d/
```

```
sudo udevadm control --reload-rules
```

```
sudo udevadm trigger
```

4. Create "usrp" group, and add your username to the group.

Run:

```
sudo addgroup usrp
```

```
sudo usermod -a -G usrp $USER
```

5. Add rtsprio line to limits.conf

Reference:

https://files.ettus.com/manual/page_general.html#general_threading_prio

Run:

```
sudo sh -c "echo '@usrp\t-\ttrtprio\t99' >> /etc/security/limits.conf"
```

6. Download FPGA images for UHD 3.9.6

Run:

```
sudo uhd_images_downloader
```

7. Run "uhd_find_devices" and "uhd_usrp_probe" to verify that the system can talk to the radio, and that the radio is alive.

Run:

```
uhd_find_devices
```

```
uhd_usrp_probe
```

8. Clone the OpenBTS build scripts

Run:

```
cd /home/amitesh/workarea/
```

```
git clone https://github.com/RangeNetworks/dev.git
```

```
cd dev
```

9. Update the build scripts to skip installing the repo UHD packages

Run:

```
sed -i 's/installIfMissing\ libuhd-dev/#installIfMissing\ libuhd-dev/g' build.sh
```

```
sed -i 's/installIfMissing\ libuhd003/#installIfMissing\ libuhd003/g' build.sh
```

```
sed -i 's/installIfMissing\ uhd-host/#installIfMissing\ uhd-host/g' build.sh
```

```
sed -i '/#installIfMissing\ uhd-host/a \\techo "Skipping UHD Package Install"' build.sh
```

10. Clone OpenBTS, and all other repositories using the build script utility. Run:

```
./clone.sh
```

Switch to master branch. Run:

```
./switchto.sh master
```

Build OpenBTS and associated tools/programs:

```
./build.sh B210
```

NOTE:

// Even if we are using USRP B100, still the Build instruction for USRP B210 is applicable as it will automatically get adapted to the USRP B100. Hence any USRP mentioned above can be interfaced with OpenBTS using build instruction for USRP B210. //

11. Install built packages. Note that this command will fail. Select "Y" to overwrite config files when prompted. You must change the path of this command to match your existing folder, so replace the "__TIMESTAMP_OF_BUILD__" with the correct timestamp. Run:

```
sudo dpkg -i BUILDS/__TIMESTAMP_OF_BUILD__/*.deb
```

12. Resolve deps from previous command. Run:

```
sudo apt-get -f install
```

13. Re-run the install again for packages, and this time it will pass. Run:

```
sudo dpkg -i BUILDS/__TIMESTAMP_OF_BUILD__/*.deb
```

14. Change permissions on Asterisk database:

```
sudo chown -R asterisk: asterisk /var/lib/asterisk/sqlite3dir
```

15. Start OpenBTS and associated programs.

In four new separate terminals,

Run:

Terminal 1:

```
sudo /usr/local/sbin/smqueue
```

Terminal 2:

```
sudo /usr/local/sbin/sipauthserve
```

Terminal 3:

```
sudo /usr/sbin/asterisk -vvvv
```

Terminal 4:

```
cd /OpenBTS
```

```
sudo ./OpenBTS
```

Now the System is ready with 2.5 G network

```
- vco_rate: 1000.000000MHz
- chan_rate: 320.000000MHz
- out_rate: 64.000000MHz
-
547724051.061346 139894919997248:
system ready

547724051.061377 139894919997248:
use the OpenBTSCLI utility to access CLI

547724051.061555 139894919997248: OpenBTSCLI network socket support for tcp:49300
```

16. Configure OpenBTS

OpenBTS should now be running with a prompt, like this:

OpenBTS >

Adjust the RX Gain. Run:

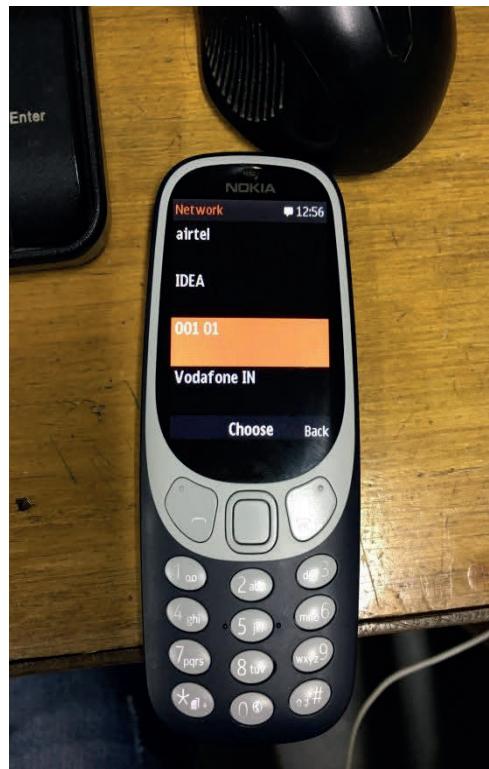
```
OpenBTS > rxgain 30
```

```
1548052531.624044 139701406619456: OpenBTSCLI network socket support for tcp:493  
00  
  
OpenBTS> rxgain 30  
current RX gain is 30 dB  
new RX gain is 30 dB
```

Configure and enable Open Registration. Run:

```
OpenBTS> config Control.LUR.OpenRegistration ".*"
```

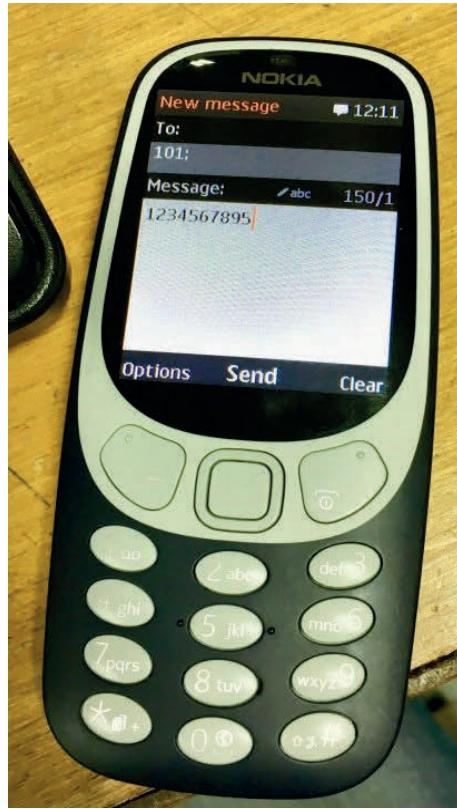
You should now be able to search for the network and connect to it with your handsets.



In the figure above the tested network with (001 01) is created. Hence now mobile user can connect to the network.

Instruction to connect the Mobile user with network:

Register with SIP by sending a 7 to 10 digit phone number to the number 101 via SMS from the handsets or manually add each IMSI with the commands below:



In a new terminal window, run:

```
cd /home/amitesh/workarea/dev/NodeManager
```

Read the list of current subscribers.

Run:

```
./nmcli.py sipauthserve subscribers read
```

Add a new subscriber. Note that the format must be "IMSI<IMSI_VALUE>", with the "IMSI" characters, followed by no space, and the IMSI number, followed by a space and then the desired phone number.

Run:

```
./nmcli.py sipauthserve subscribers create "deepak" IMSI1235678901234 9916123456
```

After you have registered both handsets, test them by sending an SMS message between the phones.

You can call the number "2600", which is the echo-back number, to test Asterisk. Dialing between handsets using the given phone numbers should now work.

Setting the Power for the receiver as

OpenBTS> POWER 20

```
OpenBTS> power 20
current downlink power +20 dB wrt full scale
```

Once the Mobile user is connected to the network, it can be identified using the instruction given bellow:

OpenBTS> tmsis

```
OpenBTS> tmsis
IMSI          TMSI IMEI          AUTH CREATED ACCESSED TMSI_ASSIGNED
405861063734669 - 352094098898730 2    175s    175s    0

OpenBTS>
```

It will list down all the mobile users connected with the network with their unique IMSI and IMEI number

IMSI is abbreviated as International Mobile Subscriber Identity which is an internationally standardized unique number used in order to identify the mobile subscriber.

IMEI is abbreviated as International Mobile Equipment Identity .It is a globally 15 digit unique number assigned to all cellular devices, which helps in identifying the mobile with the mobile network. We can use this number to block a mobile phone from being used by another person if it is stolen.

```
OpenBTS> noise
noise RSSI is -54 dB wrt full scale
MS RSSI target is -50 dB wrt full scale
WARNING: the current noise level is approaching the MS RSSI target, uplink connectivity will be extremely limited.
```

The figure above provide the information about the Noise signal by checking the signal strength for given device using RSSI (Received signal strength indicator). It is almost preferred to have a large difference between the Mobile station RSSI and Noise signal RSSI. If the difference between them is less, then it results in the warning as shown in the figure above. Hence the RSSI difference has to be reduced, which can be obtained by having a large elevation angle between the transmission and reception antenna as shown in the figure bellow:



Hence now on typing the commands again we get,

```
OpenBTS> noise
noise RSSI is -65 dB wrt full scale
MS RSSI target is -50 dB wrt full scale
INFO: the current noise level is acceptable.
```

Hence the difference between the Mobile RSSI and Noise RSSI is found to be large enough as compared to previous scenario; hence the current noise level is acceptable as shown in the figure above. Refer the reference given below to explore more regarding OpenBTS.

REFERENCES

<http://openbts.org/site/wp-content/uploads/ebook/Getting Started with OpenBTS Range Networks.pdf>

Chapter 10

ADSB-Receiver

This chapter mainly deals with the implementation of the ADSB-Receiver using Software Defined Radio. ADSB-Receiver is abbreviated as Automatic Dependent Surveillance Broadcast Receiver. This Chapter mainly deals with

Introduction to ADSB-Receiver

1. Definition of ADSB-Receiver.
2. Working model of ADSB-Receiver.
3. Application of ADSB-Receiver.
4. Advantages of ADSB-Receiver.

Implementation of ADSB-Receiver using Modes_GUI and Modes_rx GRC.

1. Installation of gr-air-modes OOT (Out Of Tree) Module.
2. Implementation of Grc flow graph of ADSB-Receiver.
3. Results and inference.

Implementation of ADSB-Receiver using ADSB Blocks in Gnu radio.

1. Installation of gr-adsb block
2. Installation of Web server dependencies packages.
3. Terminal commands for linking Webserver with gr-adsb.
4. Implementation of Grc flow graph of ADSB-Receiver.
5. Results and inference.

EXPERIMENT 1

Introduction to ADSB-Receiver

Definition of ADSB-Receiver

Automatic – It is always in ON mode does not require any pilot command or any sort of external instructions for activation.

Dependent – Depends on the Position and velocity of Aircrafts obtained from GPS or GNSS Network.

Surveillance - Provides information like Aircraft position, altitude, longitude, velocity and other surveillance data.

Broadcast - The information is continuously transmitted to the Radar station for the monitoring purpose.

Hence ADSB-Receiver is defined as a digital electronics system which is connected with an Aircraft that continuously broadcast the information regarding its exact location to the Ground station or Radar station via data link. It is a type of surveillance techniques which depends on the Aircraft information signals. Hence this signal is used for the surveillance purpose on the ground (ADSB-OUT) or onboard system (ADSB-IN).

Working of ADSB-Receiver

Fig 10.1 represents the working diagram for the ADSB-Receiver technology. ADSB Technology uses GPS (Global Positioning System) satellite to pin point an aircraft location. The ADSB Transponder on the aircrafts will transmit the information data signal like the Aircraft exact position, speed, altitude, longitude, range, height etc to the nearest ADSB ground station. The signal is transmitted twice in every one second on 1090 MHz digital data link. Hence the exact details about the moving airlines can be easily identified with accuracy, as for every instance the signals are analyzed and variation in the signals are measured.

Once the ground station receives the signals, it retransmits the information containing data signal to air control centers via data link or by the satellite connection. The data can also be used by the other airlines in order to locate each other's position. Fig 10.1.1.F represents the analyzing of the signals done in Radar station for the exact location of aircrafts. The accuracy of ADS-B does not degrade with range, atmospheric conditions, or due to any target altitude.



Figure 10.1.1 Working Flow of ADSB-Receiver.

The data Transmitted from Aircraft includes

1. Flight Identification (flight number)
2. ICAO 24-bit Aircraft Address (globally unique airframe code)
3. Position (latitude/longitude)
4. Position integrity/accuracy (GPS horizontal protection limit)
5. Track Angle and Ground Speed (velocity)
6. Emergency indication (when emergency code selected)
7. Special position identification (when IDENT selected)

Application of ADSB

1. It is used for Air – Ground surveillance for example Airport and aircrafts.
2. It is used for Air- Air surveillance for example between two aircrafts.
3. It can also be used to provide the climatic information to aircrafts.
4. It is basically used to provide the surveillances in non-radar areas.
5. It helps in airborne collisions avoidance.
6. It helps in providing air traffic display in the cockpit.

Advantages

1. ADS-B provides a complete reliable, accuracy, real-time information about air traffic pilots to the pilot in the cockpit and controllers on the ground.
2. ADS-B provides a wide range i.e. more than 100 miles; hence it can help in conflict detection and resolution.
3. ADS-B can be implemented at relatively low cost.
4. ADS-B systems are used to enhance the aviation safety through automatic traffic call-outs or warnings of imminent runway incursion.

EXPERIMENT 2

Aim

To implement ADSB-Receiver using Modes_GUI and Modes_rx GRC using Software Defined Radio.

Introduction

Universal Software Radio Peripheral (USRP) which is a type of Software Defined Radio (SDR) is used in various applications based on the military and aerospace department. This experiment mainly focuses on the ADSB-Receiver implementation using SDR. ADSB is a surveillance technology in which an aircraft determines its position via satellite navigation and periodically broadcasts it, thus enabling the aircraft to be tracked. The information is further received to the air traffic control ground stations. It can also be received by other aircraft to provide situational awareness. ADSB is also termed as "automatic" surveillance technology as it does not require any pilot for processing it and delivering the information to the ground station.

In this experiment, the USRP B100 and WBX daughterboard with radio frequency signal ranging from 50 MHz to 2.2 GHz is used to receive the signal from the aircraft and thus decode those ADS-B/Mode-S beacons signals on to the Modes_GUI Out of tree module. The information obtained from the beacons signals are displayed on to the Google Earth map. For the implementation gr-air-modes out of tree module is used.

Installation of GR-air-modes OOT module

The instruction given below will install the dependencies packages required for OOT module

```
$ sudo apt-get install sqlite3 libsqlite3-dev python-zmq python-numpy python-scipy
```

```
$ git clone https://github.com/bistromath/gr-air-modes.git
```

Once the dependencies packages are created, and then select the directory as mentioned below:

```
$ cd gr-air-modes/
```

Build the module and then install the Out-Of-Tree module gr-air-modes as mention bellow:

```
$ mkdir build
```

```
$ cd build
```

Cmake is used to compile the Out of tree module block and check for the error available. Once the cmake is compiled properly install the module as mention bellow:

```
$ cmake../
```

```
$ make
```

```
$ sudo make install
```

```
$ sudo ldconfig
```

Results

After successful building and installing of gr-air-modes OOT module, type the next command to set the sample rate for processing the signals.

```
$ modes_rx -r 10e6
```

Hence modes_rx module is active now with the sample rate of 10 MHz.

The Gain of the receiver is set to 25 dB. Fig 10.2.1 represents a details about the signals received from the aircraft where

1. Type 0 indicates that the aircraft is in the air flying with some speed which is called as air to air Surveillance.
2. Type 1 indicate that the aircraft is on the ground surface (probably on the runway). It is also called as air to ground surveillance.
3. Type 2 provides information about the longitude of direction in which the aircraft is flying.

- Type 11 is displayed when there is no information displayed regarding the aircraft.

```
sdr-tenet@sdrtenet-desktop:~$ modes_rx -r 2e6
Linux; GNU C++ version 5.4.0 20160609; Boost_105800; UHD_003.010.000.HEAD-0-g6e1ac3fc

-- Loading firmware image: /usr/local/share/uhd/images/usrp_b100_fw.thx... done
-- USRP-B100 clock control: 10
-- r_counter: 2
-- a_counter: 0
-- b_counter: 20
-- prescaler: 8
-- vco_divider: 5
-- chan_divider: 5
-- vco_rate: 1600.000000MHz
-- chan_rate: 320.000000MHz
-- out_rate: 64.000000MHz
-- Loading FPGA image: /usr/local/share/uhd/images/usrp_b100_fpga.bin... done
Setting gain to 25
Gain is 25
Rate is 2000000
(-70 0.30511967) Type 0 (short A-A surveillance) from c3e273 at 114600ft (speed 300-600kt)
(-73 0.31551892) Type 5 (short surveillance ident reply) from 13c502 with ident 558 (SPI)
(-74 0.87700917) Type 4 (short surveillance altitude reply) from db23a4 at 84900ft (SPI ALERT)
(-74 1.05788467) Type 4 (short surveillance altitude reply) from de044 at 22125ft (SPI)
(-71 1.25650292) No handler for message type 24 from 9a2c1f
(-70 1.53613567) Type 0 (short A-A surveillance) from ef61da at 21525ft (speed 600-1200kt) (aircraft is on the ground)
(-71 1.73177067) No handler for message type 24 from ed1304
(-70 1.77312317) Type 4 (short surveillance altitude reply) from 986fff7 at 17450ft (GROUND ALERT)
(-73 2.53949092) No handler for message type 24 from ed3d77
(-73 2.82646317) Type 0 (short A-A surveillance) from 8c6a38 at 46575ft (speed 600-1200kt)
(-71 3.22034867) No handler for message type 24 from bd0a98
(-75 3.28638717) No handler for message type 24 from Sac1a4
(-74 3.29336517) No handler for message type 24 from 65d820
(-69 3.83921567) Type 5 (short surveillance ident reply) from a56a9d with ident 2252 (SPI ALERT)
(-74 3.89049917) Type 5 (short surveillance ident reply) from f99f90 with ident 1464 (aircraft is on the ground)
(-72 4.21151317) Type 5 (short surveillance ident reply) from 588eeaa with ident 4204 (SPI ALERT)
(-70 4.24981067) Type 0 (short A-A surveillance) from b210df at 30975ft (Full TCAS resolution) (aircraft is on the ground)
(-72 4.63282267) Type 4 (short surveillance altitude reply) from 557f46 at 9100ft (SPI)
(-72 4.87975192) No handler for message type 24 from 59c569
(-73 4.89320417) Type 5 (short surveillance ident reply) from a98c92 with ident 7468 (SPI)
(-74 4.90510567) Type 5 (short surveillance ident reply) from e70242 with ident 6574 (AIRBORNE ALERT)
(-68 4.90964667) No handler for message type 24 from f39f71
(-69 5.37253617) No handler for message type 24 from 7fe344
```

Figure 10.2.1 tracking the Details about the aircrafts.

- Type 4 provides with the altitude of the aircraft flying in the air. It is used for short surveillances because the aircraft is tracked at every instance and altitude changes time to time.
- Type 5 provides the information about the identification of an aircraft within the short range.
- Type 17 provides with the position details (the aircraft is xyz feet height from the ground surface).

In order to run mode GUI, type the instruction mentioned bellow:

\$ modes_gui

This instruction provides with the graphic user interfaced with the Google map integration

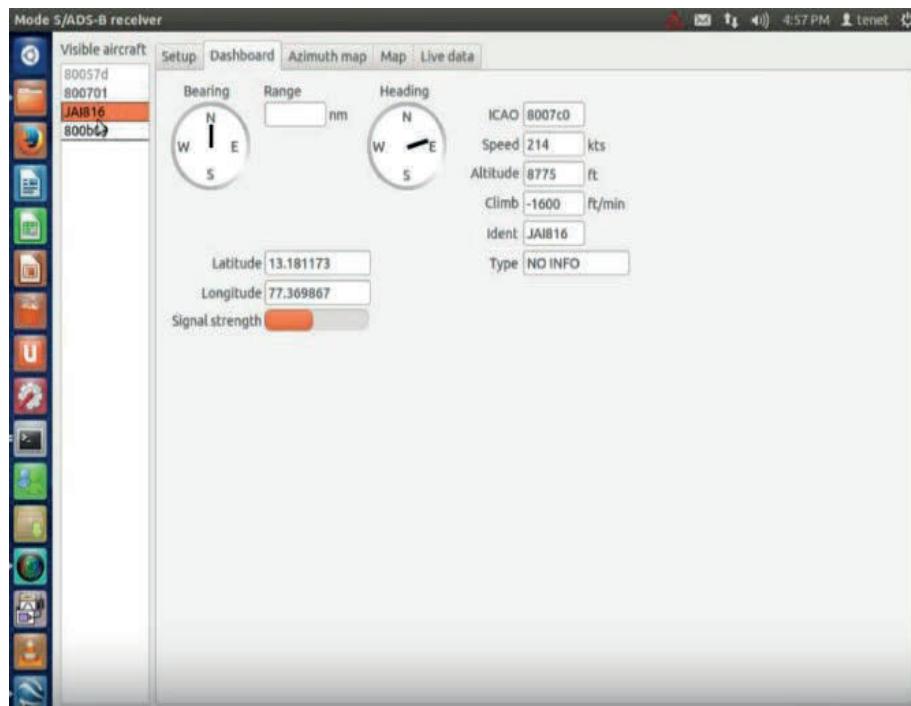


Figure 10.2.2 Dashboards.

The fig 10.2.2 represent the Mode S/ADS-B receiver module which has a dashboards providing a complete details like latitude, longitude, speed altitude etc about the visible aircraft mentioned on the left.

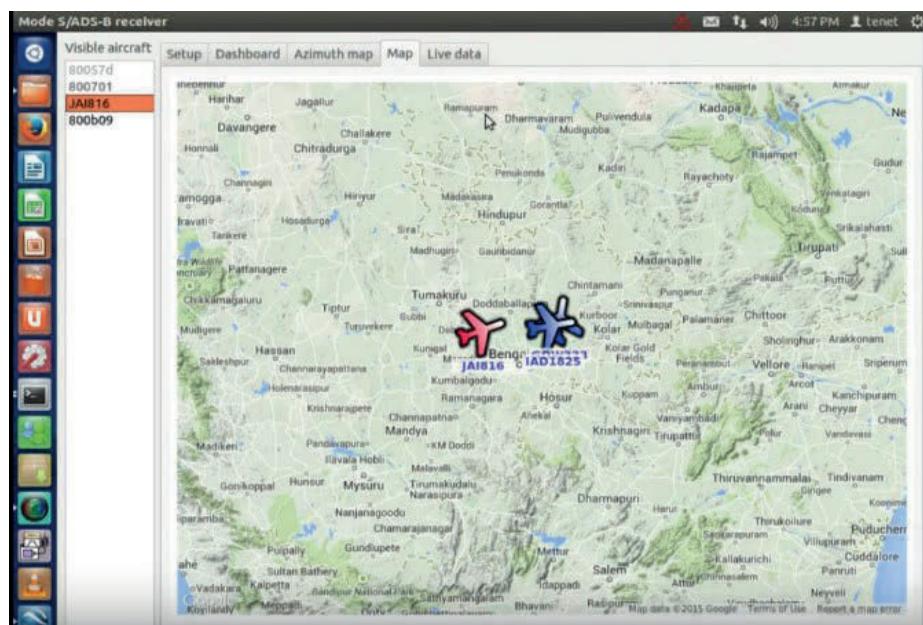


Figure 10.2.3 Tracking an Aircrafts.

The fig 10.2.3 gives a visual representation of the aircrafts flying over Bangalore with its identification number. By clicking on any location in map, we can track the aircraft details flying over the selected radar station. Once have to get a license for the API key in order to access the google map with the ADSB application.

Inference

Hence this experiment successfully demonstrates the Automatic Dependent Surveillance Broadcast-Receiver (ADSB-R) using SDR and gr-air-modes OOT module.

EXPERIMENT 3

Aim

To Implement the ADSB-Receiver using ADSB Blocks in Gnu radio and thus validating using USRP B100.

Installation

Installation of gr-adsb block

Refer the link mentioned bellow in order to download or clone the Grc file for ADSB block.

<https://github.com/mhostetter/gr-adsb#source-build>

Once the GRC file is downloaded, extract the data from the GRC file which is saved with (.Zip) extension. Follow the instruction mentioned bellow:

Remove the Build from the ADS-Receiver GRC file.

Open the terminal and follow the instruction mention bellow

```
$ cd gr-adsb  
$ mkdir build  
$ cd build  
$ cmake ../ Or $ cmake -DCMAKE_INSTALL_PREFIX=<path_to_install>../  
$ make  
$ sudo make install  
$ sudo ldconfig
```

Installation of Web server dependencies

```
$ sudo pip install flask
```

```
$ sudo pip install flask-socketio
```

```
$ sudo pip install gevent
```

```
$ sudo pip install gevent-websocket
```

Instruction to access the Web server

Open the terminal and type the commands mention bellow:

```
$ cd gr-adsb
```

```
$ cd web
```

```
$./webserver.py
```

Open the Web browser and type localhost: 5000. User must have a valid license for Google API Key. The web browser link is obtained with the Google map.

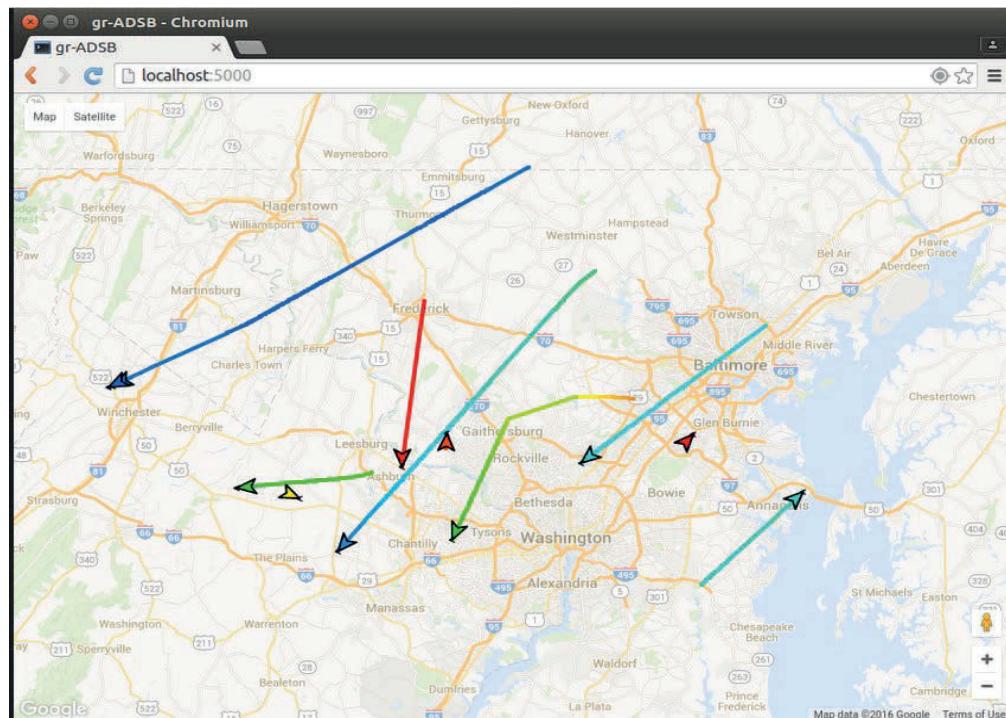


Figure 10.3.1. Localhost: 5000.

Gnu radio Flow graph

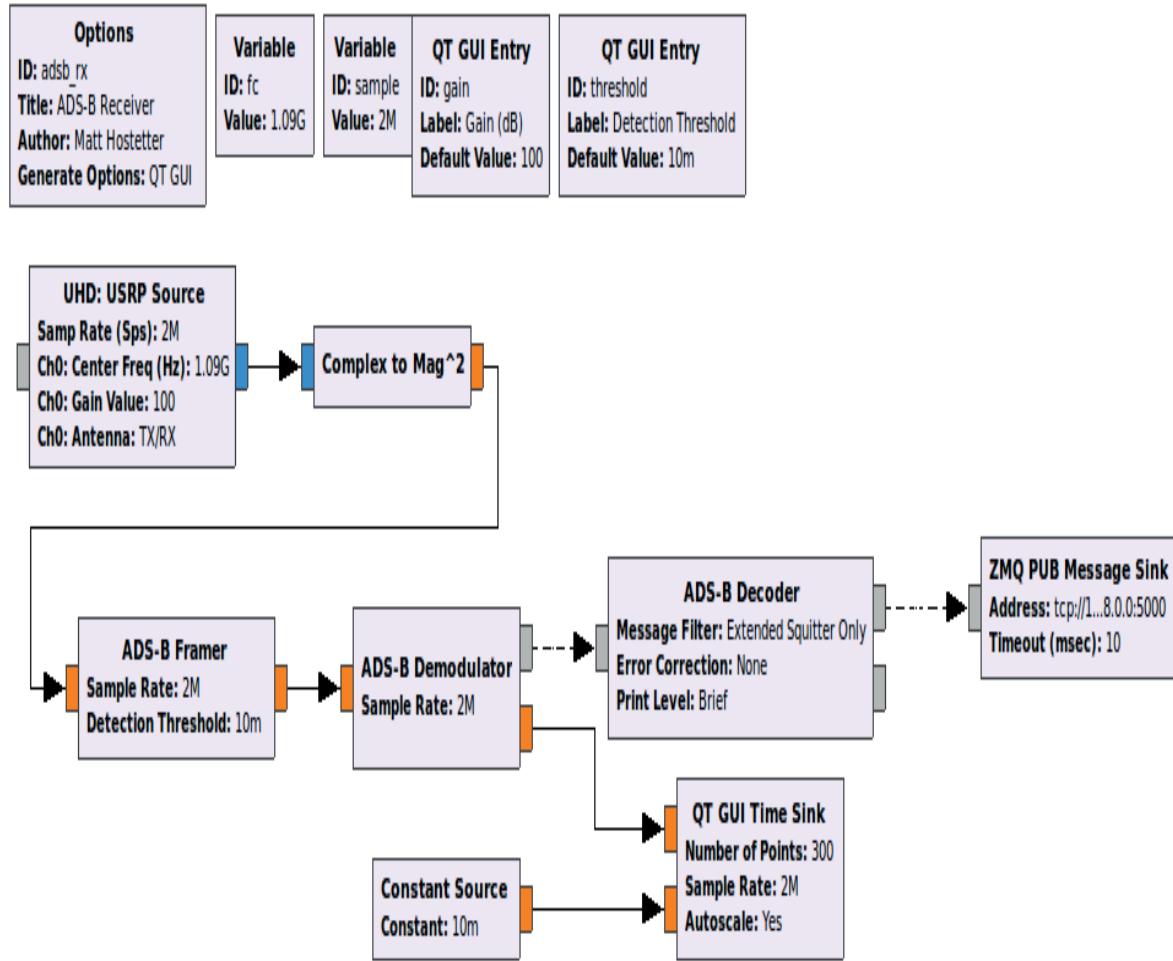


Figure 10.3.2. Map details

Fig 10.3.2 represents the Gnu radio flow graph for ADSB-Receiver. The receiver is tuned to 1090 MHz center frequency with 2 MHz sample rate. The system is made to operate at high gain (i.e. 100 dB). Receiver on the ground station receives the transmitted signal from the aircraft ADSB with 1090 MHz frequency signal hence the receiver system is tuned to 1090 MHz. Once the received signal is decoded, the signal strength greater than the threshold value is considered which indicate that some object is detected in some micro second. The threshold value is set using ADS-B Framer with sample rate 4MHz. ZMQ PUB Message Sink block act as a message port receiver and writes individual message to a ZMQ PUB Socket. A PUB socket has multiple subscribers and it can pass all incoming messages to each subscribers. By defining the ZMQ Socket

address, the subscriber is linked with the receiver and can get all the details obtained from ADSB transponder.

Result

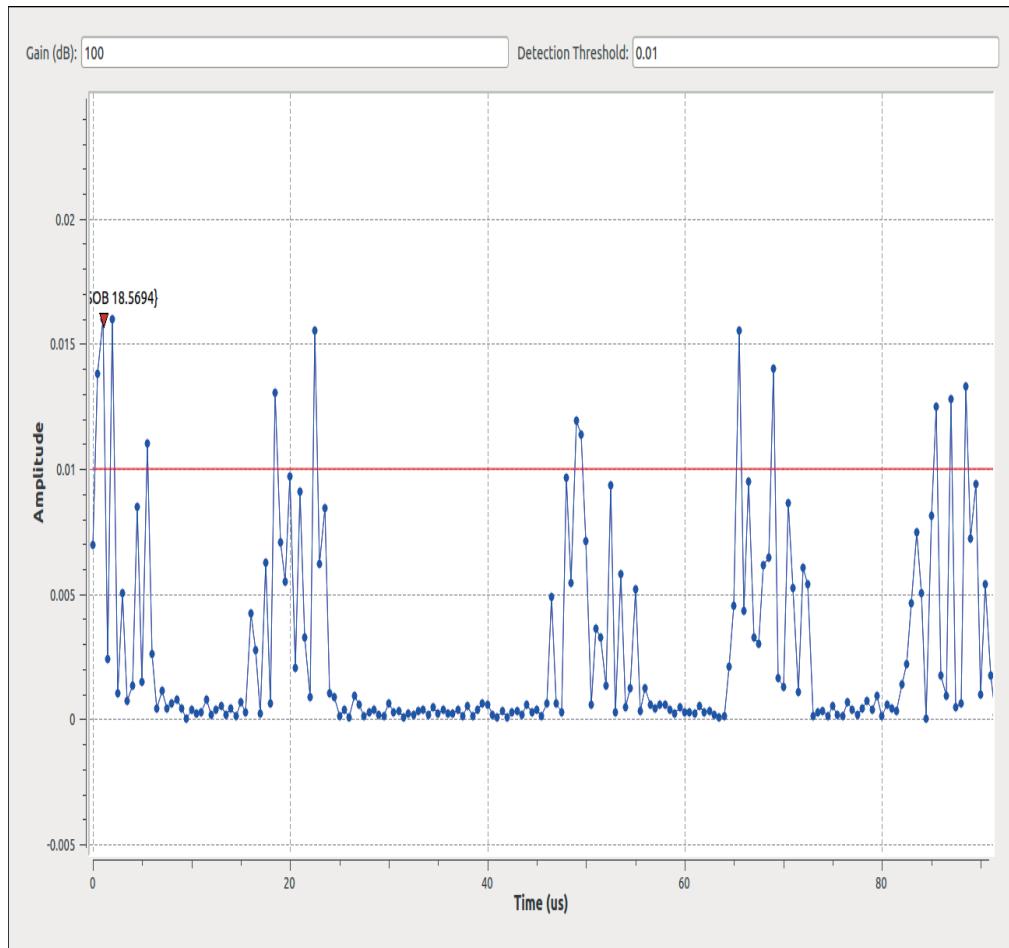


Figure 10.3.3. Time domain representation.

Fig 10.3.3 represent a time domain representation of signals received from the aircraft ADSB transponder. The change in the amplitude of a signal as shown in above figure takes place due to reception of signals from multiple ADSB transponders. The signal having amplitude greater than the threshold value is detected. These beats signal indicates that there are some obstructions appearing in some micro second time towards the receiver. The higher is the amplitude of a signal, greater is the signal strength obtained.

Inference

Hence this experiment successfully demonstrated implementation of the ADSB-Receiver using ADSB block using Gnu radio and validation on USRP B100. The ADSB receiver on the ground station is tuned to 1090 MHz because it is standardized as 1090 ADSB Transponder, according to which the aircraft ADSB transponder will transmit the signals in 1090 MHz frequency range. User will have to get a licensed Google API Key, which will help to visualize the tracking of moving aircraft status as explained in the experiment.

Chapter 11

Frequency Signal Jamming

This chapter mainly deals with the implementation of Frequency Signal Jamming using Software Defined Radio. Frequency Jamming is basically a process of transmitting high signals in the same frequency range in order to decrease the signal to noise ratio. Hence this process results in the deliberately interference or blockage of desired wireless communication frequency signals. The device working on this concept is called as Signal Jammer. Hence any wireless communication system which makes use of radio frequency signals can be used for jamming by allowing a strong radio frequency signal in the same frequency range. Consider an example of Wi-Fi, which can be attacked with a network jammer, thus reducing the signal quality strength until the signal is disconnected. The Fig 1 provides a complete image about the process of signal jamming.

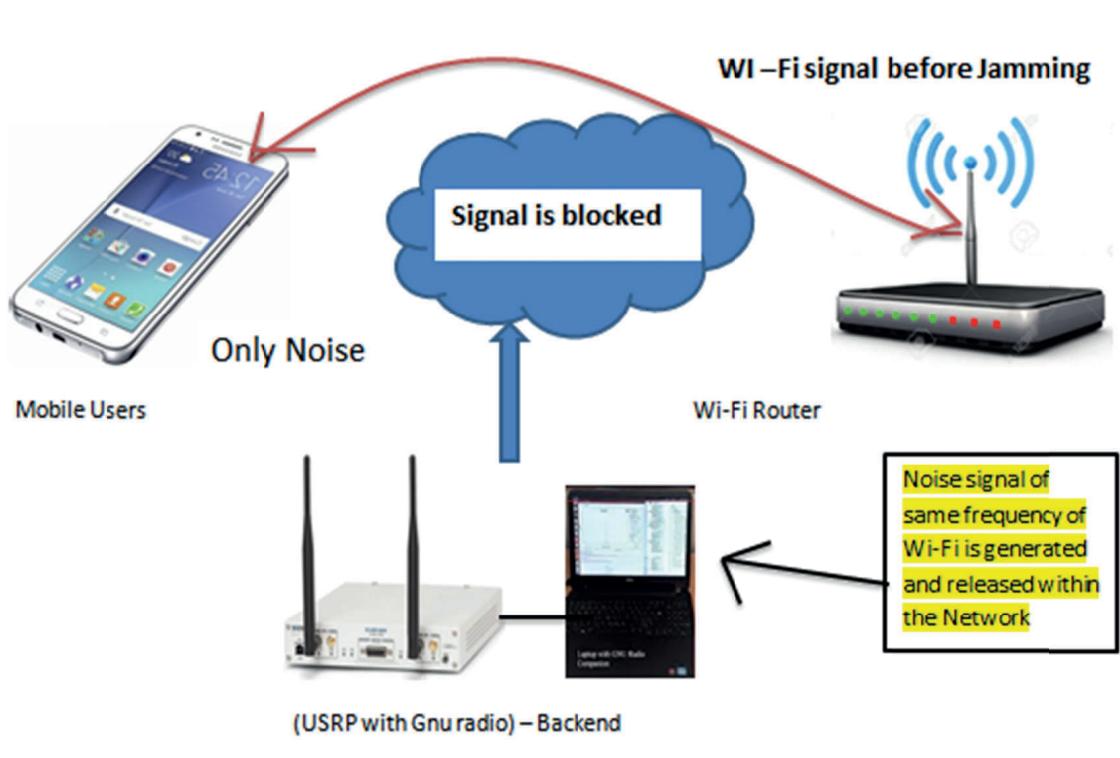


Figure 11.1 Signal Jamming.

The Sender I.e. the Mobile user is connected with the Receiver (Wi- Fi) with some frequency range. A Jammer is created by generating an unwanted noise signal of same frequency range and is further added to the channel medium within the defined Wi-Fi frequency range. Hence none of the mobile users in that range will receive a Wi- Fi signals as it is blocked by the Noise signals generated. Wireless communication system are used for transferring information between two or multiple networks and hence these systems are unaware of the kind of attacks performed using Jamming. Hence numerous security issues are generated. Attackers are able to hinder the wireless channel medium and insert the unwanted messages. Therefore Jamming is considered to be fundamental way of degrading network performance. It adversely corrupts the original message so that it cannot reach to the intended receiver. Therefore it is considered to be as a criminal offence as per the USA government. These attackers cannot be easily identified.

Jamming can be done in two forms.

1. External threat model - Jammer will not be the part of the network.
2. Internal threat model - Jammer will be part of the network.

Attackers try different ways so that their attack effects are significantly different. For example, one with a constant jammer consumes all the resources available and continuously tries to jam the network. The constant jammer can be easily detected, as the user network will be effected continuously even after the correction of issues. On the other hand, a reactive jammer senses the medium and attacks based on the situation. The reactive jammers are hard to detect, as they never attach continuously.

Application of Signal Jamming

1. Stop a particular user group activities by disturbing their communications
2. Eliminate the public nuisance which occurs in places like movie theaters, temples restaurants etc.

Implementation of Wi-Fi signal Jamming using SDR

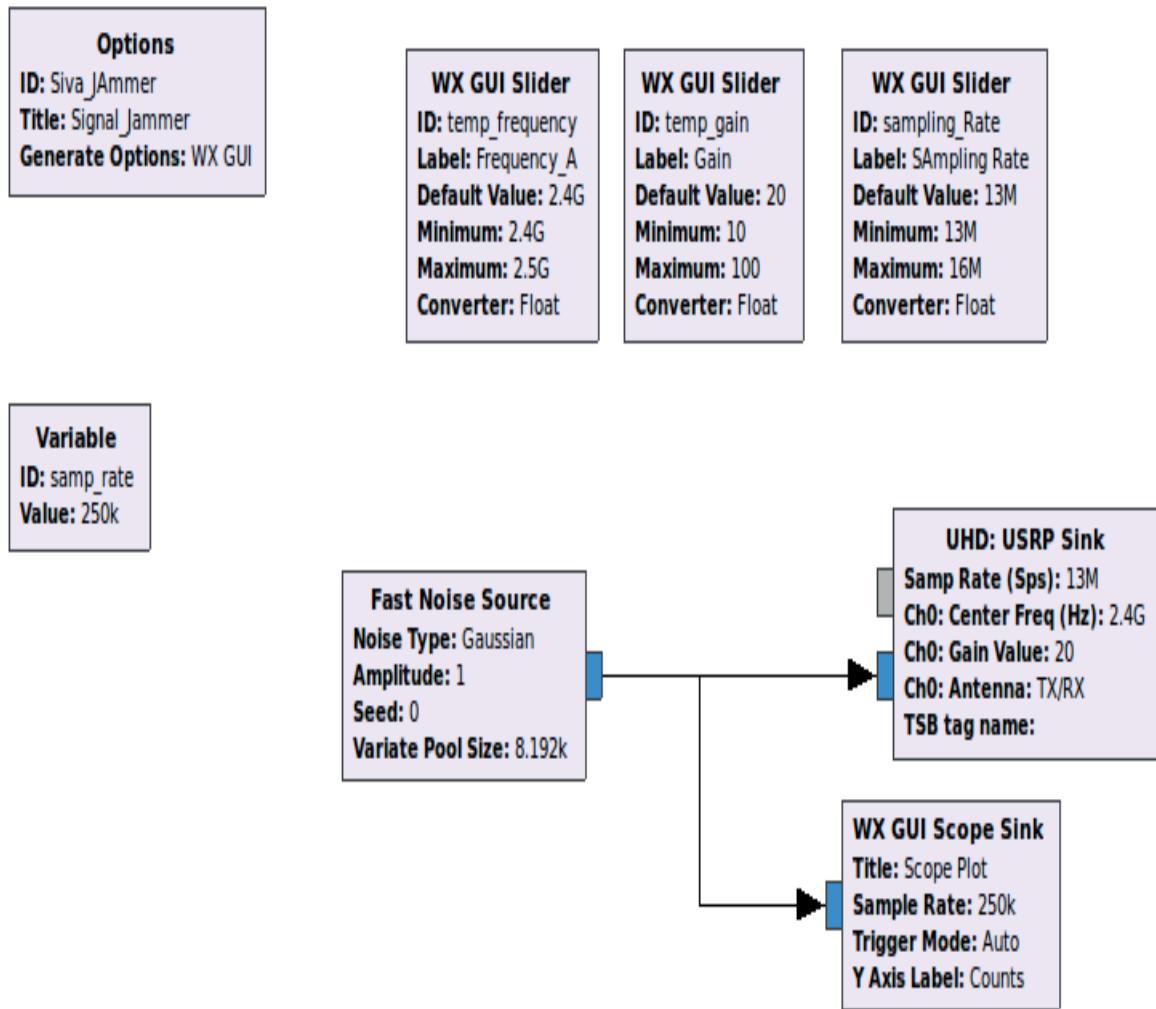


Figure 11.2 Gnu radio flow graph Signal Jamming.

The Gaussian noise is used as an input signals sampled at the rate of 250KHz. Fast noise source block is used which pre-generate a pool of random variates taken from the specified distribution. In this experiment the range of distribution is defined from 2.4GHz to 2.5GHz radio frequency signals. Hence at run time, samples are randomly chosen from the pool of 8.190 Kb/s samples (this sample value is by default). Thus resulting in the fast operation. The Fast noise source block can represent the output in forms of integer, float, and complex and short integer format. The center frequency is tuned top 2.4 GHz frequency signals with 20 dB gain in order to maintain the Wi-Fi standards.

Experimental Setup

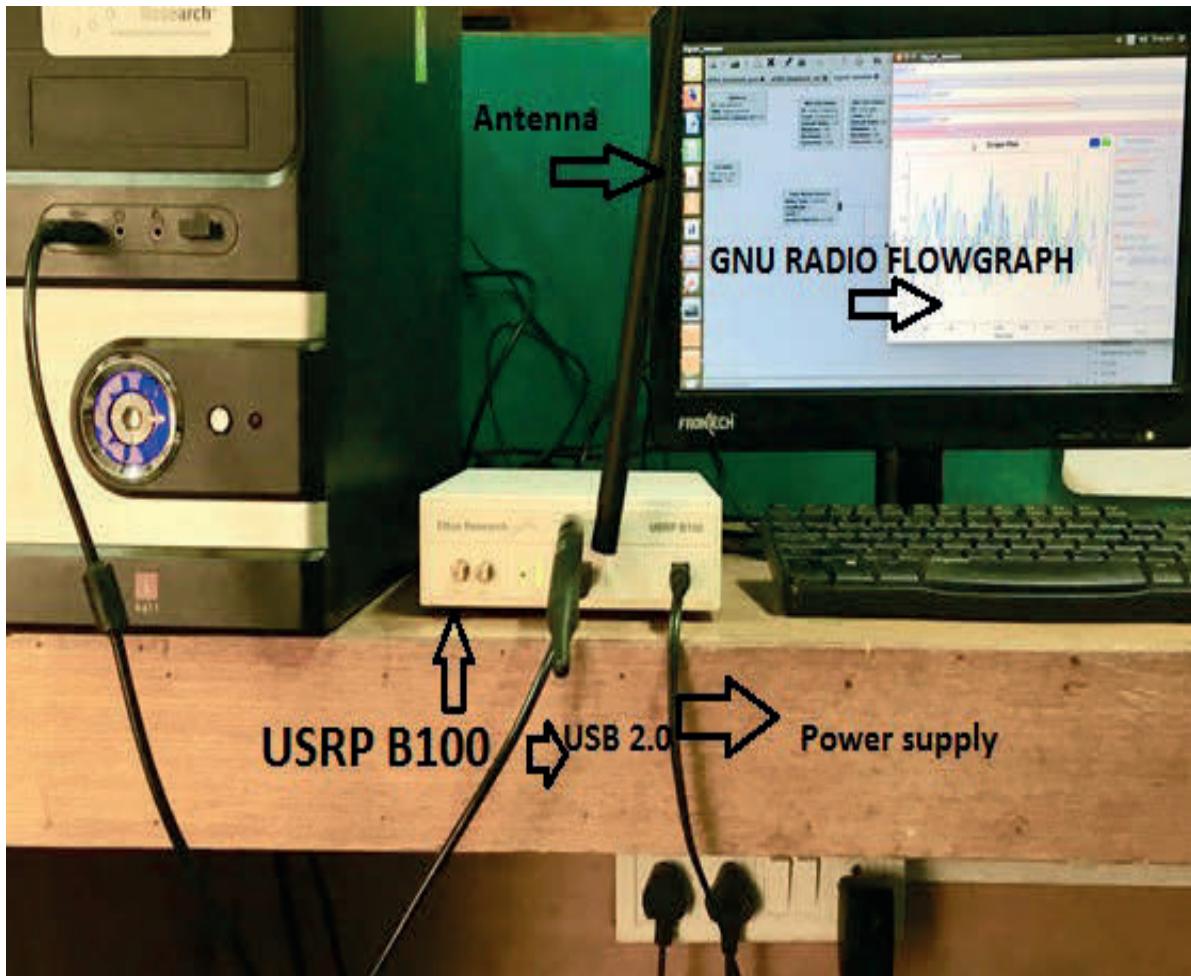


Figure 11.3 Experimental Setup for Signal Jamming.

Fig 11.3 represents an experimental Setup for Wi-Fi Signal Jamming. USRP B100 is used to generate and transmit the unwanted noise signals with same radio frequency range as that of Wi-Fi signals. USRP B100 is interfaced with the system using USB 2.0 with the power supply of 6V DC / 3 Amp current. The omnidirectional antenna is connected with the USRP as an input to RF1 and RF2 Port. The Gnu radio is used for analyzing the signal processing parameters and once the signal is processed, it is been radiated to the channel medium using SDR. Hence the Gnu radio is considered as Backend and SDR is considered as a Frontend system. USRP B100 is interfaced with SBX RF daughterboard supporting radio frequency from 400MHz to 4.4GHz. USRP B100 is able to support up to

6GHz radio frequency signals. Hence any RF daughterboard within this range is compatible with this SDR.

Results

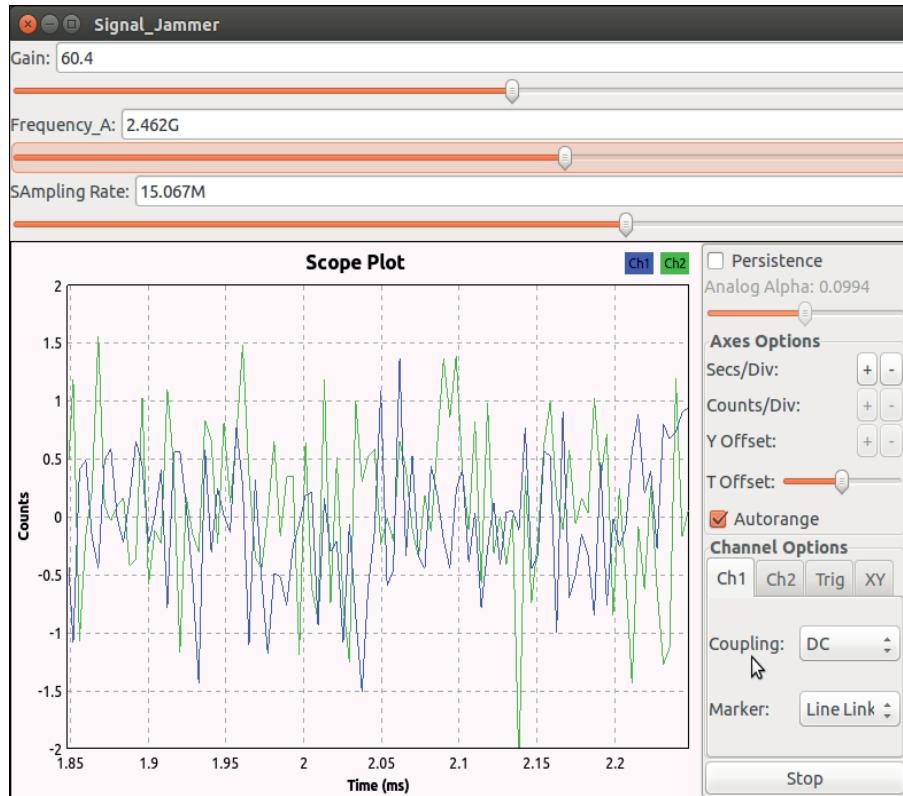


Figure 11.3 Time domain representation of Signal Jamming.

The Fig 11.3 represents a time domain representation of Gaussian Noise signals with a sample rate of 15.067 MHz. The center frequency is tuned to 2.462 GHz based on the Wi-Fi Standards. The radio frequency signals around the system can be identified by the command mention bellow:

```
$ Sudo iwlist wlx84e064b0399 scan
```

Once the command mentioned above is typed in the terminal, the entire frequency signal available around the system is visible which can be observed in Fig 11.4. The fig bellow informs about the Wi-Fi service set identifier device which is registered with name “Tenet”, having center frequency of 2.427 GHz operated at channel 4 (I.e. Tenet Wi-Fi router is operated at 4 channel). The router channels are set at the time of

installation. Hence further when the same range Noise frequency is generated and mixed within the range defined then the Wi-Fi signal will get blocked.

Figure 11.4 Radio Frequency identification around the system.

The signal strength of Wi-Fi is -61.dBm. Ideally the signal strength of Wi-Fi router should be -65.5dBm which is not reliable in real time scenario.

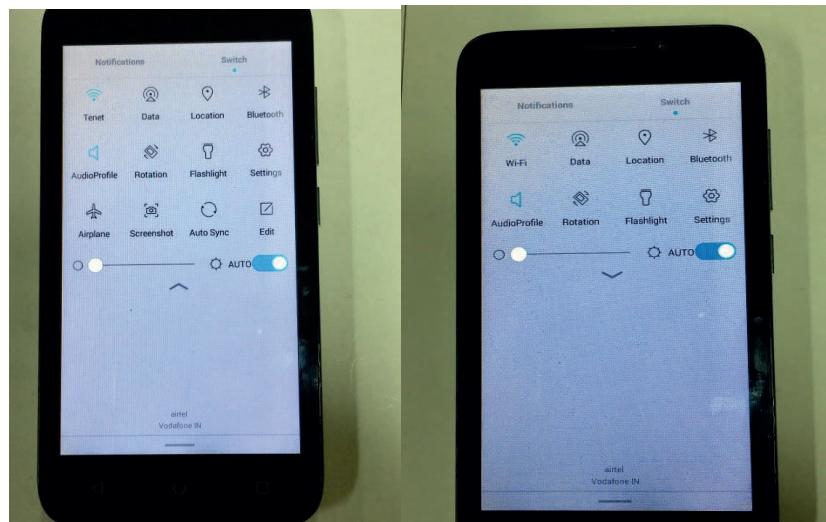


Figure 11.5 (A) the Mobile phone is connected with Tenet Wi-Fi 11.4(B) the Mobile phone got disconnected due to Wi-Fi Jamming.

From the Wi-Fi standards, out of 11 channels only channel 1, 6, and 11 are the non-overlapping channels due to avoidance of Adjacent and co – channel interference. Hence Wi-Fi speed runs faster with high data rate. Fig 11. 5 provide a complete idea about the connection of internet in the mobile system with Tenet Wi-Fi system before and after Jamming of Wi-Fi signals. Once the Gaussian noise of 2.427 GHz radio frequency is radiated the mobile phone got disconnected from the Wi-Fi services.

Inference

Hence the motive of this experiment to describe the concept of signal jamming by blocking the Wi-Fi signals, which has a center frequency of 2.427GHz, is demonstrated successfully. The radio frequency signal radiated by various devices like Wi-Fi router, mobile devices etc around the system can be easily identified by the instruction mentioned above. Hence by introducing the same range of Gaussian noise frequency signals to the channel medium, the signal can be jammed or blocked. The Gaussian Noise is considered with purpose that it is a type of known noise which can be controlled at any instant from the developer side.

NOTE:

Disclaimer

The purpose of this experiment and demonstration is mainly to understand the concept and is not recommended to be tried and tested in a public place causing discomfort to others. This experiment is documented for academic reasons and needs to use their discretion and also abide by the laws locally before trying this one at their end.

Chapter 12

GSM Sniffing using SDR

This chapter mainly deals with the implementation of GSM Sniffing using gr-GSM module with implementation on a USRP X310. The GSM which is abbreviated as Global System for Mobile communication system is a 2nd generation standards which was developed for the purpose of digital voice communication system with TDMA /CDMA as a multiplexing techniques used. It is the standard which evolved in 1980 and lasted till 1990, but still the architecture of GSM structure is considered as base reference for the generation like 3G, 4G and for the upcoming 5G technology. In this chapter, we are going to sniff out the network for 2nd generation (I.e. performing the process of capturing and monitoring the data which passes through the network using Sniffing tool). The information like the location of mobile station and the identification of BTS to which the mobile station is currently connected with Source port address and destination port address can be extracted from this experiment.

Table 15: Specification of USRP X310

Index	USRP X310
1	Interfacing using 1Gb Ethernet cable
2	2 RF Daughterboard slot
3	Support RF signal from DC to 6 GHZ
4	Analog BW 160MHz /channel
5	Output power 9-16 V with 7.5 Amp current
6	Support FPGA Xilinx Kintex 7

This chapter deals with the USRP X310 series which is a type of Software Defined Radio (SDR) having 2 defined RF daughterboard slots with radio frequency signals ranging from DC range to 6 GHz. The specification of USRP X 310 is mentioned above:

What is GSM Sniffing?

The GSM Sniffing as defined above is a process of extracting the complete information regarding the Mobile station dealing from the source port address to destination port address, from the country location to the mobile connected network location, from the amount the packet transmission to the amount of packet received etc. It is a process of continuously tracking or monitoring the information for the dedicated mobile users. One can also extract the information like which mobile user is currently hopping or which user is currently requesting for the secured end to end channel. It also defines the communication protocols supported by the particular mobile system.

The Terminologies used in GSM standards

Some of the important terminologies are defined below

1. MCC (Mobile Country Codes)

It is abbreviated as Mobile Country codes which is defined as a unique three digit decimal number dedicated for the Countries worldwide. The First digit of country code indicate the geographical area for example

- 2 is for europeum countries,
- 4 is for the Asia and middle eastern areas,
- 3 indicate North American and
- 7 are for central and southern part of America.

The Second digit defines the country. For example for India it is represented with a number zero (0). Hence the India is represented 40 country code. The Third digit indicates the State allocated based on the communication operators in the Country. For example, in India Reliance Jio supports

- 1. 404-MCC for states like Assam, Himachal Pradesh, Bihar, Orrisa etc

2. 405-MCC for states like Maharashtra, Karnataka, Gujarat, Madhya Pradesh, Punjab etc

2. MNC (Mobile Network Codes)

MNC is abbreviated as Mobile Network Codes which is used in the combination with the MCC as mentioned above in order to uniquely identify the subscribers in a given particular areas . For example

1. 404-44 indicates Karnataka by the Spice communication PVT Ltd operator.
2. 404-45 indicates Karnataka by the Airtel operator.
3. 404-71 indicates Karnataka by the BSNL Service operator.
4. 404-86 indicate Karnataka by the Vodafone operator.
5. 405-10 indicates Karnataka by the Reliance Jio operator.
6. 405-803 indicates Karnataka by the AIRCEL operator.

3. ARFCN

ARFCN is abbreviated as Absolute radio frequency channel Number which is defined as an allocation of a unique number to each radio channel in GSM standard. With the help of ARFCN, one can identify the uplink and the downlink frequency used by particular mobile users.

4. BCCH

BCCH is abbreviated as Broadcast control channel is a type of logical channel used by the base station in a GSM network in order to send information about the identity of the network. The information obtained by the base station is used by the mobile user in order to access the network.

5. CCCH

CCCH is abbreviated as the Common control channel is a type of channel which is used for transferring the control information from all the mobile stations to the BTS. It

normally occurs when the mobile users is trying to initiate a call or else is responding to a page.

The Installation of Gr-GSM

Install all needed prerequisites with following command mention bellow:

```
sudo apt-get update  
sudo apt-get install  
sudo apt-get cmake  
sudo apt-get autoconf  
sudo apt-get libtool  
sudo apt-get pkg-config  
sudo apt-get build-essential  
sudo apt-get python-docutils  
sudo apt-get libcppunit-dev  
sudo apt-get swig  
sudo apt-get doxygen  
sudo apt-get liblog4cpp5-dev  
sudo apt-get python-scipy  
sudo apt-get python-gtk2  
sudo apt-get gnuradio-dev  
sudo apt-get gr-osmosdr  
sudo apt-get libosmocore-dev
```

Then download the gr-gsm source and build it with following commands:

```
git clone https://git.osmocom.org/gr-gsm  
cd gr-gsm  
mkdir build  
cd build  
cmake ..  
mkdir $HOME/.grc_gnuradio/ $HOME/.gnuradio/  
make  
sudo make install  
sudo ldconfig
```

Now your gr-gsm module is properly installed.

The Installation of Wireshark Sniffing tool

In order to add the PPA follow the instruction. PPA is used to install the Wireshark sniffing tool.

```
sudo add-apt-repository ppa:wireshark-dev/stable
```

Update the system and install the Wireshark tool using the instruction mentioned bellow

```
sudo apt-get update  
sudo apt-get install wireshark
```

Hence Now the Wireshark sniffing tool is installed. Run the command in order to open the wireshark. Run:

```
wireshark or else sudo wireshark.
```

A page will open like mentioned bellow:

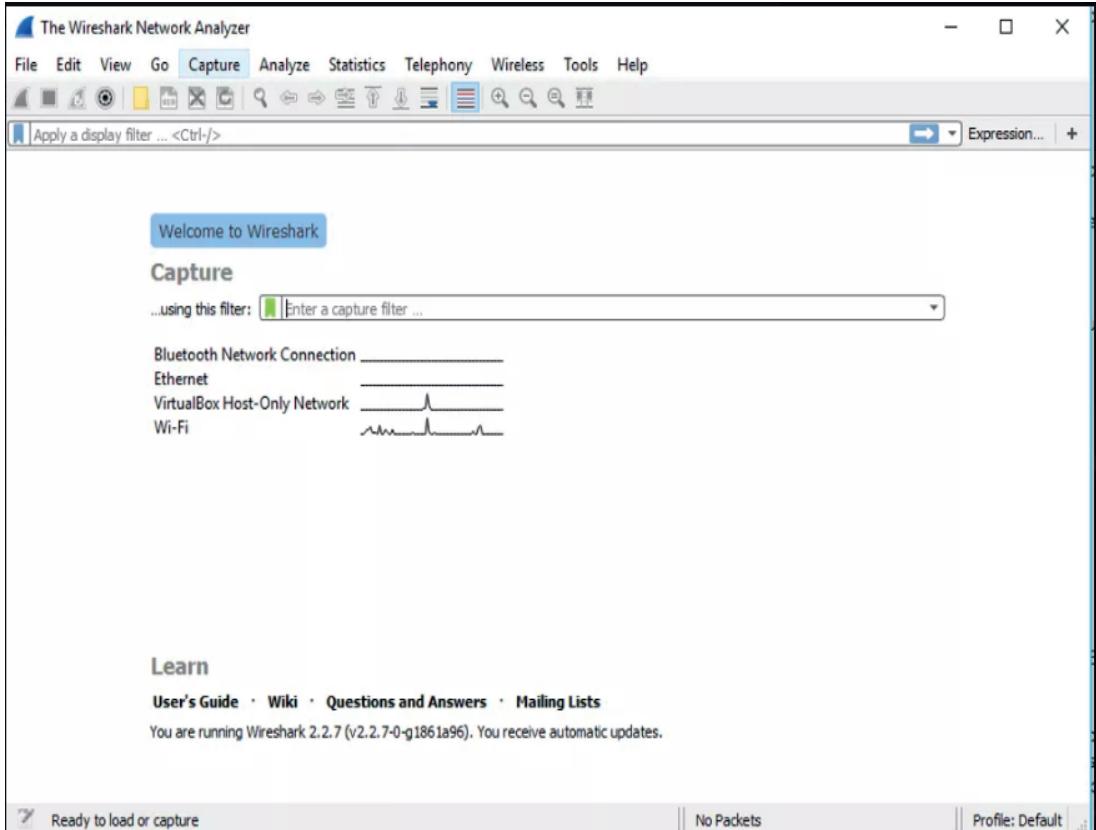


Figure 12.1. Wireshark Network Analyzer

Fig 12.1 represents an image of Wireshark Network Analyzer, indicating the connection of devices to the network via LAN, Ethernet, Wi-Fi, SDR. SDR is represented by loopback network interfaced using wired medium i.e. Ethernet cable wire.

Gnu radio flow graph

Fig 12.2 represents a flow graph of GSM sniffing. The received signals from the Base station, the signal are further processed to GSM input adaptor. The GSM input adaptor is an adaptor of input stream for the GSM receiver. It contains frequency offset corrector and resampler to correct the carrier frequency and sampling frequency offsets. The GSM clock offset control, provides a limit to the frequency offset signals with the cut off frequency of 941.4 MHz as a threshold. The signals are further allowed to pass through the BCCH+CCCH demapper, which demap the control channels; hence it

corresponds to the channel combination specified in GSM. SDCCH is a type of standalone dedicated control channel which is used in the

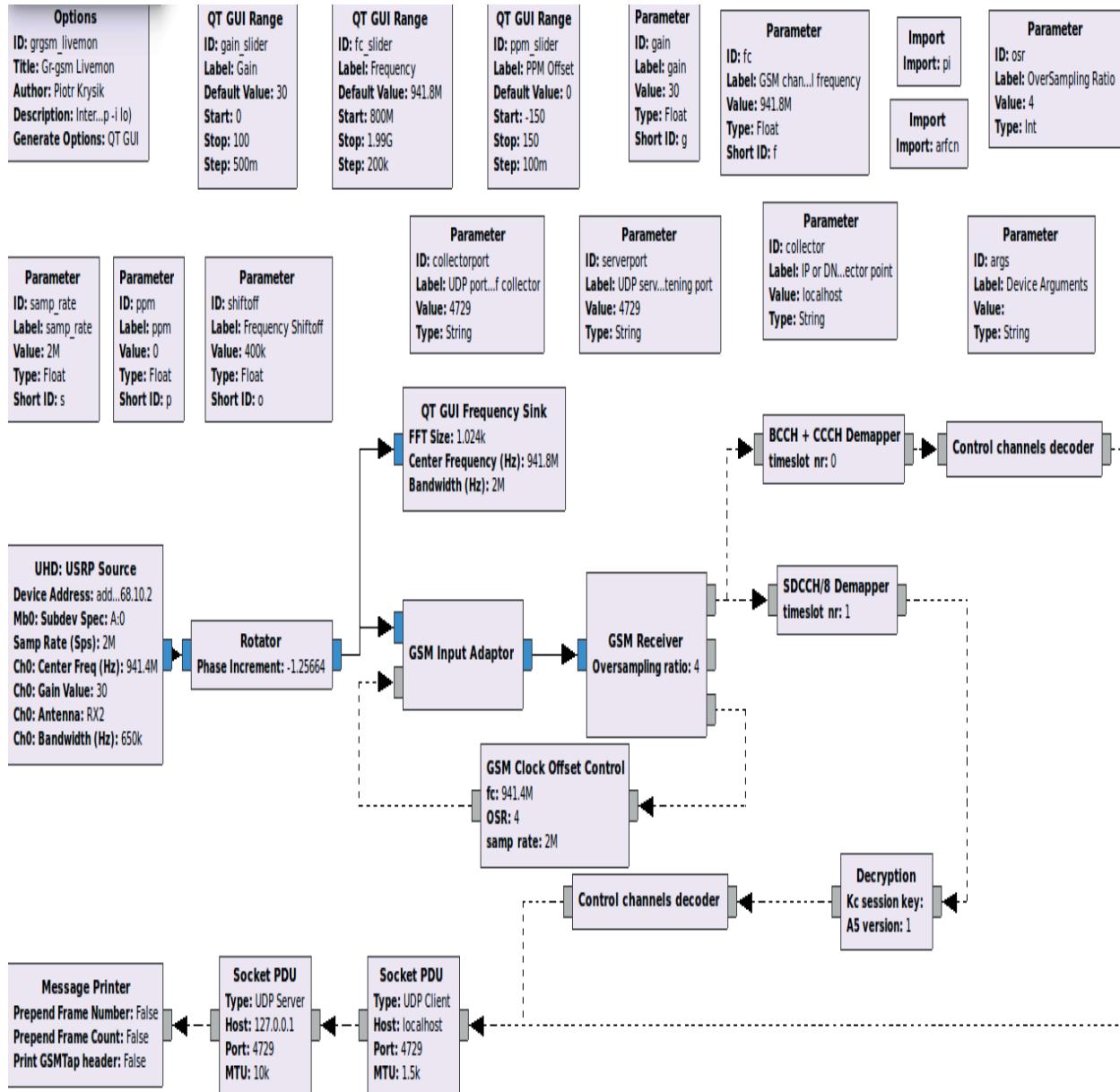


Figure 12.2. GSM Sniffing Gnu radio Flow graph.

GSM standards in order to provide a reliable connection to signaling and SMS messages. Hence these signals are demapped using SDCCH/B receiver. In fig bellow, two Socket PDU is used; one for the UDP client and other is for UDP Server. The GSM message

printer will prepend the frame count as mentioned in the figure. The receiver is tuned to 941.4 MHz center frequency in order to receive the 2G signals from the BTS with a channel Bandwidth 650 KHz.

Results

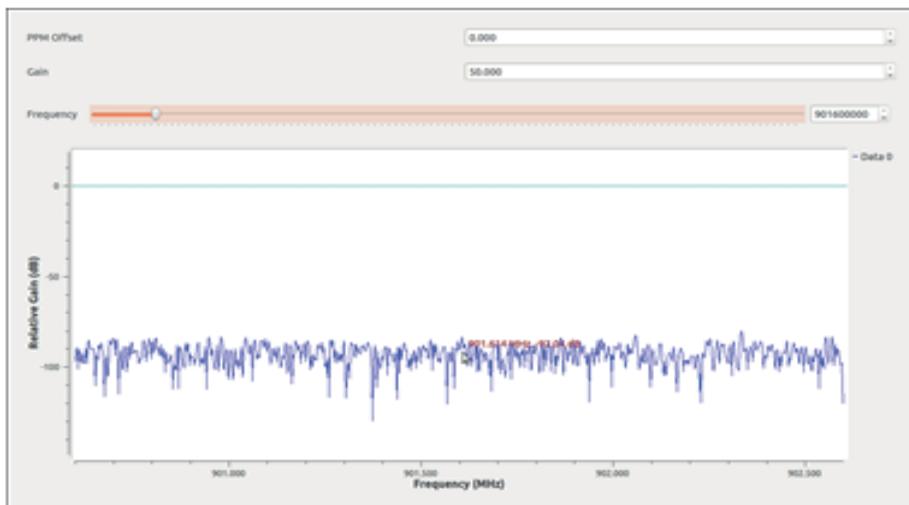


Figure 12.3. Result indicating a flat response (NO reception of packages).

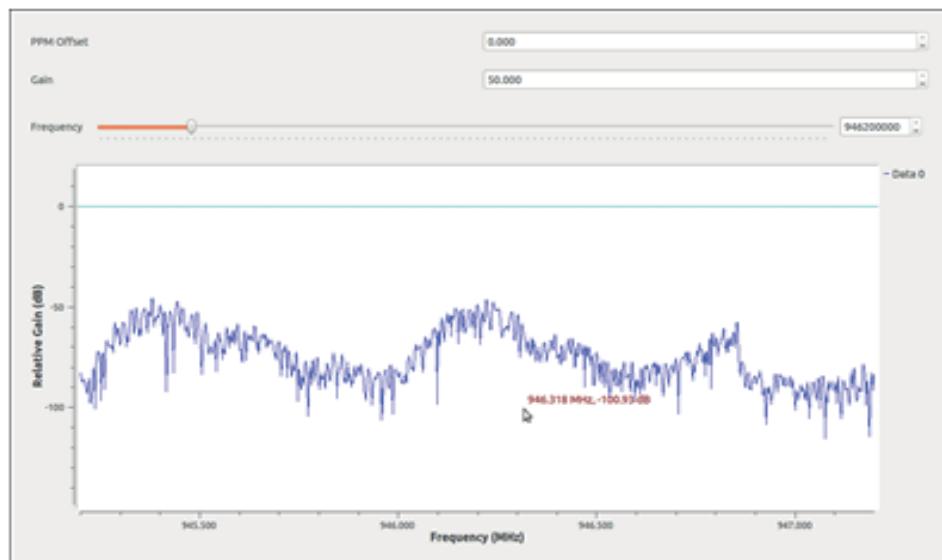


Figure 12.4. Result indicating a reception of packages from BTS

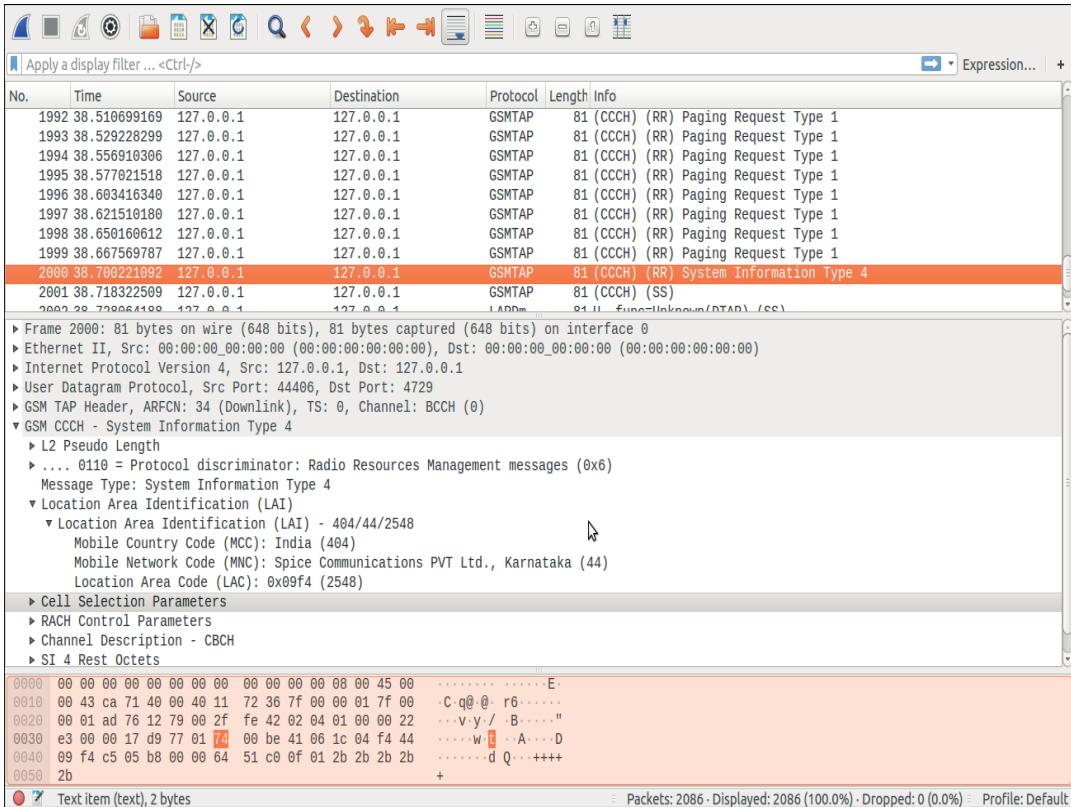


Figure 12.5.Received signals decoded and displayed using Wireshark.

Run the Gnu radio flow graph or else type command as

grgsm_livemon

Open the terminal and then type the command to open the Wireshark sniffing tool to decode the message signal received from the BTS. Click on the loopback network.

Wireshark or else type sudo wireshark

Fig 12.3 represents the frequency response of the signals, which represent a flat frequency response thus indicating that there is no reception of packages. Hence tuning the GSM receiver to proper center frequency is very important. Identify the ARFCN code for any mobile user and then calculate the uplink and downlink frequency based on the ARFCN. Further substitute the center frequency with the downlink frequency. Fig 12.4 thus indicates a frequency response of a GSM receiver which is tuned with the downlink frequency of 946.2 MHz. Hence some curve frequency response of signals is obtained.

Further analyze the signal in Wireshark. Fig 12.5 represents the extracted information for the selected network. The mobile system is in the attempt of connecting to the channel, as it receives the signals from base station via BCCCH for the identity of network. The ARFCN is noted as 34 with transmission of packages as zero. The Fig 12.5 also provides information about the source port address and the destination port address with the local area identification of a mobile user. The mobile user is subscribed to spice communication PVT Ltd, Karnataka with MNC as 44 having MCC as 404 indicating the number is from India.

Inference

While tuning the GSM receiver, proper care is needed in the selection of center frequency which can help in identifying the reception of packages from BTS. Hence identify the ARFCN of mobile user and calculate the uplink and downlink frequency and replace it in the center frequency. Hence some variation in the frequency plot will be seen. The loopback network has to be selected as it receives the information from the USRP X310 having a gain of 50 db.