International Journal of Engineering & Technology, 7 (2.31) (2018) 9-12

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Design approach for wideband FM receiver using RTL-SDR and raspberry PI

P. Satya Narayana[1], M.N.V.S. Syam Kumar[2*], A. Keerthi Kishan[3], K.V.R.K. Suraj[4]

[1]*Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.*
[2]*Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.*
[3]*Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.*
[4]*Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India.*
**Corresponding author E-mail:syamkumar846@gmail.com*

Abstract

Software defined radio replaced majority of hardware modules like mixers, filters, modulators and demodulators etc., with Software blocks in the field of radio electronics and communication. In this some or all the functionalities are Configurable using this software implemented on technologies like FPGAs, DSPs etc. Owing to lack of ease in implementing and reconfiguring huge hardware modules, we move on to implement an adaptable communication system with the help of SDR, as it can be easily configured to work with wide range of frequencies. We find various SDR transceiver modules which can be interfaced with digital computer and aided with firmware like GNU radio, SDR shark, etc., allowing us to construct blocks with the help of built in components that decode and process the received data and produce required output. In requirement of implementing a cost-effective, compact sized and portable system, we use a processing unit providing enough computational power to perform signal processing tasks which is Raspberry pi. Here we are going to implement a low cost SDR communication system that capture, process and visualize the Wide Band Frequency signal.

Keywords**:** *GNU radio, software defined radio(SDR), Raspberry PI.*

1. Introduction

A communication system became an eye to visualize and communicate the information all over the world. It became a need and inherent in day to day lives, From the initial stages of communication in analog way including Amplitude Modulation, Frequency modulation, Phase modulation schemes, PWM, Delta modulation, PPM, PAM, pulse code modulation, the desire for accuracy and better way of communication rapidly transformed analog schema into digital.

The digitized communication allows more clear and accurate communication without losses, eliminating the defects of the analog communication systems like distortion, interference, etc. As the theory for digital communication is improved, corresponding practicalities are also realized. Different circuits implementing the digital techniques are realized which may or may not be application specific. At the start of the digital era, the chip technology is not much developed, with a different chip for a specific application. Suppose an FM transceiver used to have a large circuit for processing the signals of desired range only. As the range of the signal increases, the RF front end should be changed and the circuit for processing the signals should also be changed.

The transformation in the digital field paved a path to originate the software dependent radio. Various companies including Realtek, Elonics and Noolec supported the development of the software-defined hardware and software. The main purpose of the software defined radio is to act as very cheap, multi operating system compatible RF front end for the wide range of the signals frequencies.

The Whole system has RTL-SDR collecting the data in analog form and converting into digital form, with raspberry pi used for the computation purpose. When Raspberry pi is joined to RTL-SDR, it acts as a low-cost communication node which can receive wide range of the signals and process for a purpose and capable of transmitting it with the help of external hardware support.

In this present work, we implemented various modulation schemes and an FM receiver in GNU– RADIO using RTL-SDR hardware.

2. Literature review

From [1] we can get reference to create a low cost and power effective system which can satisfy our requirement. Various types of hardware systems that are available in market have been described and finally the hardware components and their integrations to implement signal processing techniques have been dealt in this paper. This paper deals with integration process of RTL-SDR module with Raspberry pi. In situation like places with limited resources, low power and portable systems should be deployed, thus Raspberry pi meet the need of power optimization and less complex, more integrated low cost system. This paper also discusses various components that have to integrate as single circuit to have an RTL module (RF front end). It also explains RTL2832U receiver's specifications like operating bands, sampling frequency and other specific details that help in setting parameters in GRC software. Finally operation of RTL SDR system has been explained.

From [2],this paper gives a small tutorial how to work with SDR. Multiple companies provide various software interfaces to RTL module to do the task of signal transmission and reception. But this paper deals with a open source software GNU Radio Companion (GRC). We have learnt about the software that its backend of GRC works in python, but this software provides GUI

so that various blocks can be used to design the system. We have come to know the basic blocks that are required and their utility has been described. Various analog and digital modulation schemes like AM, PM, FM, ASK, BPSK, FSK and multi rate operations like Decimation and Interpolation have been discussed.
From [3], it is yet another paper discussing integration of RTL-SDR with Raspberry pi, it gives explanation of FM reception and sending the received information across LAN. The work described here in this paper is on the integration software SDR sharp in remote pc. This paper also gives reference of procedure and a set of commands that are needed to make raspberry pi act like system .We have to install the GRC or SDR sharp software in raspberry pi unit,and its plugged in with RTL dongle .The reception has been started ,the signal received is digitalized in dongle and fed to raspberry pi and decoding using various blocks implemented in SDR sharp .
From [4], it deals in with 802.11 a/g/p OFDM Receiver design using GNU Radio. It mainly explains how to detect start of 802.11 a/g/p encapsulated frame using method which is based on autocorrelation of short time sequence. How Frequency offset, phase offset correction can be done and necessity to correct, has been discussed. It has discussed important task, Channel estimation which must be done mandatorily when dealing with real time systems. This channel estimation helps to modulate signals at transmitter so that there would be less loss due to channel's noise. Signal field detection which tells how actual input signal present and Frame decoding tasks are implemented. Thus we got to know various types of losses that a channel impends on OFDM packet and methods of detecting packet with noise.
Thus the reference of all these papers has given us strong theoretical basics for implementation of our minor project. Since OFDM is being utilized in the present day in high data rate communication systems we aim at working with OFDM receiver design as a further work to be done.

3. Theoretical analysis

Software defined radio

Software Defined Radio is software based communication system. In a traditional communication system, where the components like mixtures, filters, amplifiers, modulators/demodulators, detectors, etc., are implemented in hardware. But in software defined radio, the components of the communication system are implemented on embedded system by means of software components written in python. The digital electronics which are theoretical are made into reality with the help of SDR.
A SDR framework involves working with computer equipped with components like analog-to-digital converter, external sound card, assisted with RF front end .The signal processing techniques are handed over to GP processor. By following this design we come up with a radio communication system that can widely transmit and receive process, different protocols.
The main concept is to connect an analog-to-digital converter to an antenna. The converter is read by the digital signal processor. This data stream is converted to a specific form required by the software. Finally, the software will manipulate the data based on the application requirement.
The typical software defined receiver is shown in the following block diagram.

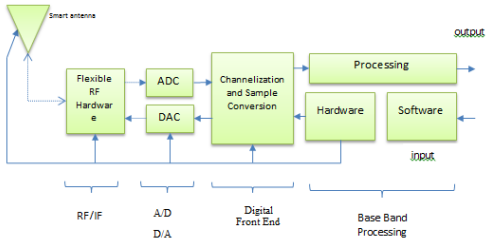

Fig. 1: Architecture of software defined radio

The SDR based transmitter will be of similar in structure to the transmitter. A digital signal processor generates a stream of numbers. A digital-to-analog converter will convert these numbers into analog form. Finally, an antenna is assisted to send the analog data into space.
The above scheme is not completely up to the mark due to the limits in the technology. The major problem is the transformation between analog and frequency domains at a higher rate. The accuracy and higher rate cannot be obtained simultaneously without relying upon physical processes like interference and electromagnetic resonance for assistance.

RTL-SDR

RTL-SDR is a cost effective device that meet our needs of choosing an software defined radio. It uses a DVB-T TV tuner dongle supported by RTL-2832U chipset. RTL-SDR is the created by Osmocom and AnttiPalosaari, Eric Fry, who first identified that the accessing and manipulating of signal data in I/Q form using a computer aided with specific drivers.
In market, many dongles with distinct tuners are available. Some of the tuners available are Elonics E 4000, Rafael Micro R820T/2, Fitipower FC0013, Fitipower FC0012, FCI FC2580. In this project, the experimental study is completely based on the RTL-SDR which is based on the Rafael Micro R820T/2 tuner.
With the maximum sample rate being 3.2MS/s (mega samples per second) the RTL-SDR is unfortunately unstable at this rate as it drops samples at this rate. The optimum sample rate at which we don't lose any samples is 2.4 MS/s. With an input impedance of about 75 ohms for a dongle usage of 50-ohm cable on a 75 ohms input, result in minimal loss of 0.177 db. The actual resolution of the RTL-SDR is 8 bits with an count of effective Number of Bits (ENOB) being estimated at ~7. However the Decimator factor in the software may raise this value.
For the most general GUI based software defined radio software, we need at least a dual core processor. Several applications can also be accommodated on the single board computers like Raspberry Pi 3 and Android mobile devices. The device is as shown below:

Fig. 2: RTL-SDR

GNU Radio

For signal processing applications, software should be required for processing the signals. In this project, we use GNU – RADIO software for signal manipulations. GNU–RADIO is an open source development toolkit that provides various signal processing blocks to implement software radios. It can be used as a simulation tool which can be used without hardware and used with readily available hardware. Its usage is prominent in hobbyist, commercial environments to support both real-word radio systems and wireless communications research.
This software is licensed under the GNU General Public License (GPL) version 3 or later. GNU – RADIO will work in Windows and Linux environment. GNU – RADIO supports many RTL-SDR dongles provided drivers of the required dongle.
The GNU – RADIO platform that provides all the signal processing. It can be used to write the applications to receive data out of digital streams or to push data which can be transmitted using hardware. GNU – RADIO provides filters, synchronization elements, vocoders, equalizers, channel coders, demodulators and many other elements. These elements are called blocks. It manages how the data is passed from one block to another. The

powerful feature of the GNU – RADIO platform is that we can create and add new blocks.
GRC is a tool of GNU Radio software which helps the user to drag and drop the blocks in it. GRC was created to simplify the usage of GNU Radio by allowing one to create python files graphically as oppose to creating them code alone. The interface looks as below:

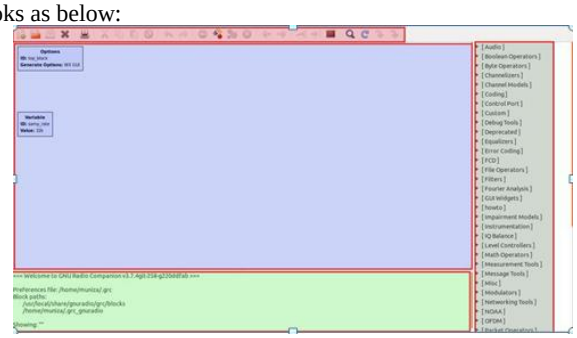

Fig. 3: GRC interface

Raspberry PI

Raspberry Pi is the progression of a little single – board PC. All the model of Raspberry Pi has a Broadcom system on a chip (SoC) with a coordinated ARM perfectly central processing unit (CPU) and an on-chip graphics processing unit (GPU). In this undertaking, it is utilized for figuring reason. It can run Linux working framework on it. GNU – RADIO is perfect to Linux condition. With the assistance of a Raspberry Pi and RTL – SDR associated, a unit is shaped, and it can go about as an ease organize hub in a correspondence framework.

4. Experiment analysis

At the time of modulation, the deviation term in the modulated wave will decide the type of modulation. If the deviation term value is +/- 3 kHz, then it is termed as narrow band FM and if its value is +/- 75 kHz, then it is termed as wideband FM. To support higher quality transmissions, signals with large deviation are used. To receive an FM signal, the receiver should be sensitive to frequency variations of the incoming signal. The system should be made insensitive to amplitude variations, since the information will not be present in the amplitude variations rather in frequency variations.
Practical circuits which are designed for receiving FM signals are Coincidence FM demodulator, Ratio Detector, PLL, Slope FM detector, Foster-Seely FM demodulator, Phase locked loop FM demodulator, Quadrature FM demodulator.
Although, there are many devices to demodulate a signal, the GNU–RADIO uses python programming to do this demodulation process. The WBFM receive block have two properties which include Quadrature Rate and Audio Decimation Rate. As simple as that, the two parameters are given based on the parameters of the other blocks or components in the flow graph. The property panel is as shown in below

Fig. 4: Properties GUI of WBFM receive

In this paper, WBFM receiver is implemented practically using RTL-SDR. The experimental setup is shown below

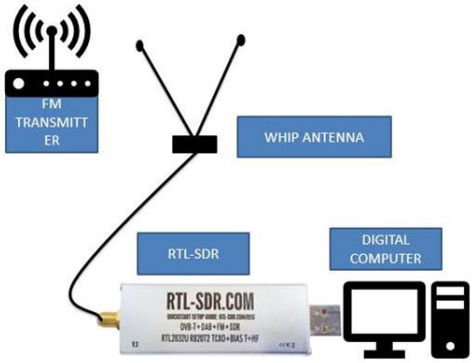

Fig. 5: Experimental setup of wideband FM receiver

The flow graph for wideband FM receiver is illustrated in following figure:

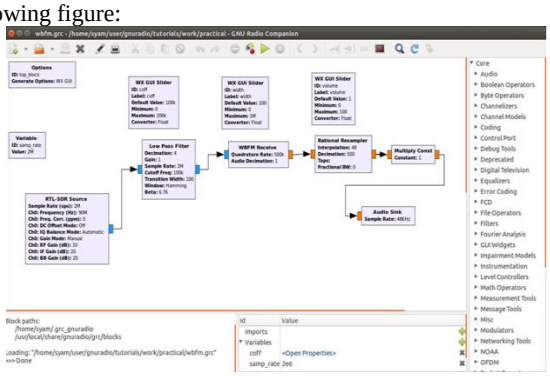

Fig. 6: Flow graph of wideband FM receiver

The blocks used in the design of wideband FM receiver are RTL-SDR source, low pass filter, WBFM Receive block, Rational Resampler, Multiply const, audio sink, WX GUI slider, variable, options as shown in FIG.4. The description of the blocks and their contribution to the project is described below in detailed.

Options

Options block is used to setup the properties regarding the flow graph.

Variable

Variable block is meant for setting the values for a variable name which can be used anywhere in the entire flow graph. It has two properties, namely ID and Value. In this flow graph, a variable named samp_rate is used as its ID and its value is 2M.

RTL–SDR source

This block is used to configure the receiver. In this project, RTL-SDR is used, so that an RTL-SDR SOURCE block is used. The sampling rate field will be pre-populated with the samp_rate. The ch0 frequency is set to 98.3MHz. Leave the remaining fields as it is.

Low pass filter

To get the information in baseband signal, it should be low pass filtered to some value.

- The cut-off frequency is set to 100 KHz since it is standard cut-off frequency. In this flow graph, a slider named coff is used to vary the cut-off frequency.
- It is not required to process all the frequencies, so we use decimate the samples. Decimation factor is set to 4. So, the sample rate is reduced to 500 KHz.
- Transition bandwidth may vary for different incoming signals. We use a WX GUI slider named width for sliding purpose.

WBFM receive

The demodulation used is wide band frequency demodulation. So, we use a WBFM block.
It has two parameters, namely quadrature rate and Audio decimation rate. Quadrature rate is the output rate of the WBFM RECEIVE block. Audio decimation rate further reduces the output sampling rate by its value. In this flow graph, quadrature rate is set to 500K and Audio decimation rate is set to 1. So, the output sampling rate is 500 KHz.

Rational resampler

Rational resampling is a process of manipulating the sampling rate of a signal at fractional value. It is also called I/D sampling. Where 'I' represents interpolation and D represents Decimation.
Decimation value is set to 500. So, the input of rational re-sampler is reduced by 500. i.e., 500K is divided by 500 and it turns out to 1KHz. Interpolation factor is set to 48. So,
48 is multiplied by 1KHz results in 48KHz which is the required sampling rate of the soundcard of the PC.

Multiply const

This block is used to multiply a constant value to its input. To increase the amplitude or the volume of the audio signal, this block is used. This block has single parameter named constant. A GUI slider with ID volume is used, so that it is referenced from constant parameter in multiply const block. In this way, the volume can be changed.

Audio sink

The final output is given to Audio sink block. Audio sink resembles the soundcard of the PC. It has a single parameter named sample rate which is set to the sound card sample rate which is 48 KHz

5. Discussion of results

Wideband FM receiver is designed as per the flow graph described in the experimental analysis section. The output is analysed by connecting a speaker to the raspberry pi board. The output GUI is described as follows.

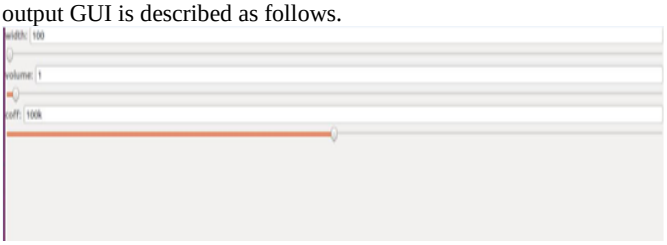

Fig. 7: Output GUI of WBFM receiver

It consists of three controls. They are width, volume and coff. Width determines the bandwidth of the filter used. Volume refers to the output strength of the signal. And coff represents the cut-off frequency of the filter.
The exact output can be achieved or reproduced by adjusting the control values on the GUI.

6. Conclusion

In this paper, we have implemented an FM receiver and gave a brief look of SDR based communication system with the help of RTL-SDR as an RF front end and implemented some modulation schemes in GNU-RADIO with Raspberry pi as computational device.

References

[1] Tomar VS & Bhatia V, "Low Cost and Power Software Defined Radio Using Raspberry Pi for Disaster Effected Regions", *Procedia Computer Science*, Vol.58, (2015), pp.401-407.
[2] Gandhiraj R, Ram R & Soman KP, "Analog and digital modulation toolkit for software defined radio", *Procedia Engineering*, Vol.30, (2012), pp.1155-1162.
[3] Bloessl B, Segata M, Sommer C & Dressler F, "An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio", *Proceedings of the second workshop on Software radio implementation forum*, (2013), pp. 9-16.
[4] Danymol R, Ajitha T & Gandhiraj R, "Real-Time Communication System Design using RTL-SDR and Raspberry Pi, *ICACCS*, (2013).
[5] Vachhani K & Mallari RA, "Experimental study on wide band fm receiver using gnuradio and rtl-sdr", *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (2015), pp.1810-1814.
[6] Sierra EG & Ramirez Arroyave GA, "Low cost SDR spectrum analyser and analog radio receiver using GNU radio, raspberry Pi2 and SDR-RTL dongle", *7th IEEE Latin-Americal Conference Communications(LATINCOM)*, (2015).