Telecommun Syst
DOI 10.1007/s11235-013-9850-7

Modern analog and digital communication systems development using GNU Radio with USRP

R. Gandhiraj · K.P. Soman

Abstract In this modern world many communication devices are highly intelligent and interconnected between each other. Any up-gradation of the hardware in the existing communication devices is not easier one. Compatibility of the new hardware with existing hardware is highly essential. But the new protocols may or may not support the older one. The solution for these problems can be provided by using the reconfigurable hardware design. The hardware can be reprogrammed according to the new change in technology upgradation. The cost of commercially available hardware and software requirements for setting up such a module is very high. This can be solved by using Open source hardware and software such as Universal Software Radio Peripheral (USRP) and GNU Radio. This work demonstrates how the modern analog communication system like Community Radio Schemes and Radio Data System (RDS) and digital communication systems such as Simple Digital Video Broadcasting (DVB) and OFDM based data communication can be developed using the Open source hardware USRP1. This work will be helpful even for first year level of engineering students to easily implement any communication and control applications with cheaper cost.

Keywords Software defined radio (SDR) · Community radio · Radio data system (RDS) · Digital video broadcasting (DVB) · GStreamer · OFDM based data communication

R. Gandhiraj (✉)
Communication Engineering Research Group,
Department of Electronics and Communication Engineering,
Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: r_gandhiraj@cb.amrita.edu

K.P. Soman
Center for Excellence in Computational Engineering and
Networking, Amrita Vishwa Vidyapeetham, Coimbatore, India
e-mail: kp_soman@amrita.edu

1 Introduction

A software radio [9] is a radio system which performs the required signal processing in software instead of using dedicated integrated circuits in hardware. The benefit is that since software can be easily replaced in the radio system, the same hardware can be used to create many kinds of radios for many different transmission standards; thus, one software radio can used for a variety of applications.

GNU Radio is an open-source and free development tool which provide signal processing modules to implement radio communication system. It consists of two modules one can do the simulation of system model; other can do the real-time implementation of the system using RF hardware. It can be used in academic, research and commercial application development with real-world solutions. GNU Radio applications are created using Python language and critical signal processing modules were written using C++. Another advantage of this tool is, it can read recorded signal file as data for further signal processing with file write facility, without using real RF hardware. This is tool is licensed under GNU General Public License with copyright from Free Software Foundation (FSS).

1.1 Role of GNU Radio in software radio

GNU Radio performs all the signal processing. It can be used to write applications to receive data out of digital streams or to push data into digital streams, which are then transmitted using hardware. GNU Radio has filters, channel codes, synchronization elements, equalizers, demodu-

Published online: 12 September 2013

lators, vocoders, decoders, and many other which are typically found in radio systems. More importantly, it includes a method of connecting these blocks and then manages how data is passed from one block to another. Extending GNU Radio is also quite easy; if we find a specific block that is missing, we can quickly create and add it.

Since this tool is software, it is capable of handling the digital data. For input from receiver and output to the transmitter data types by default will be in complex baseband form. The analog front end of the hardware module will translate this baseband signals into the corresponding pass band carrier frequency. Other data types such as bits, bytes, vectors, etc. also can be used in intermediate processing stage. High level programming language is Python and highly level signals processing blocks can be created by the C++. These modules can be called from Python using SWIG interface.

1.2 SDR block diagram

Universal Software Radio Peripheral (USRP) is a universal hardware which can be accessed via GNU Radio software on Linux platform as shown in Figs. 1, 2. SDR hardware is having three major parts. Left one is called as Radio Frequency (RF) front end which will Transmit (Tx) or Receive (Rx) real analog signals. The second block will act as bridge between analog block and digital block. The third block is designed for all type of signal processing as software form. Antenna can transmit and receive the RF signal. It can be connected using Sub Miniature Version A (SMA) connectors with RF frontend (daughter board). Both paths can function separately. USRP has many available daughterboard's which vary from each other on the basis of their operating frequency. Daughterboard's acts as the RF frontend of USRP and are connected to the USRP motherboard.

On USRP motherboard, the analog signals are converted to digital samples and mixed down to baseband within the FPGA. Sample rate is also reduced by doing decimation. Looking into the Fig. 2, data sampled by the FPGA are sent to the host by USB. Further signal processing is controlled by the GNU Radio framework connecting to the host computer (right block in Fig. 2). GNU Radio is providing various pre-assembled signal processing blocks for waveform creation and analysis in software radio development. GNU Radio Companion (GRC) is acting as Integrated Development Environment (IDE) for application development. The detailed description of each block is given in the later sections.

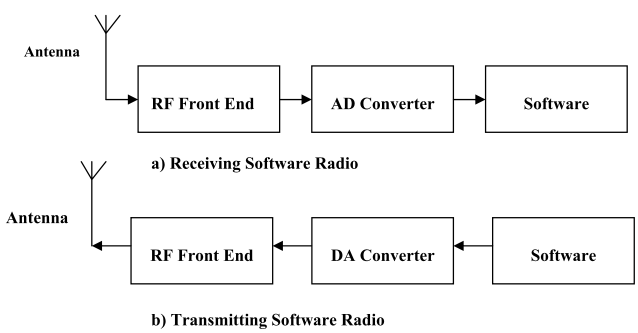
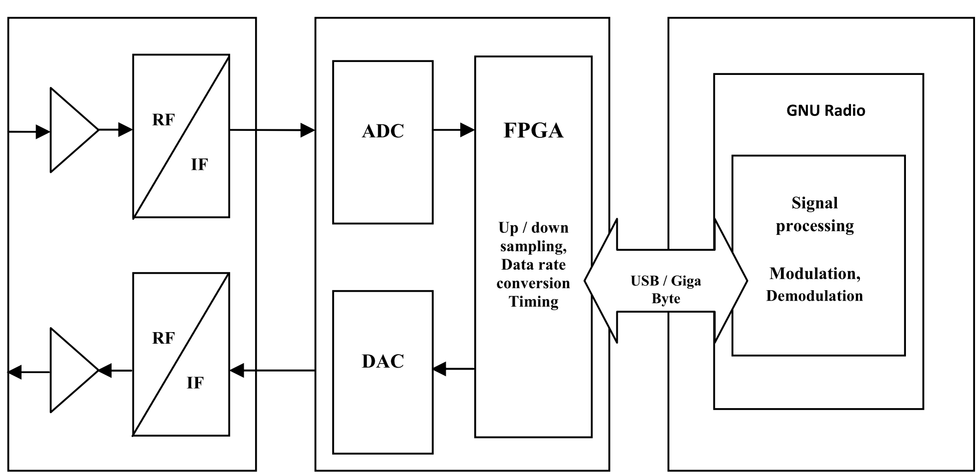


Fig. 1 Generalized structure of SDR (**a**) receiving end of software radio, (**b**) transmitting end of software radio

1.3 GNU Radio

GNU (GNU's not UNIX) radio is a free/open-source software toolkit for building software radios, in which transmitted waveforms and demodulation of the received waveforms is done by the software. It changes radio hardware problems into software problems. GNU Radio provides functions to support for implementing spectrum analyzer, an oscilloscope, concurrent multichannel receiver and a collection of modulators and demodulators. The fundamental characteristic of software radio is that software defines the transmitted waveforms, and software demodulates the received waveforms. While in most of the radios the processing is done

Fig. 2 Block diagram of software defined radio [15]



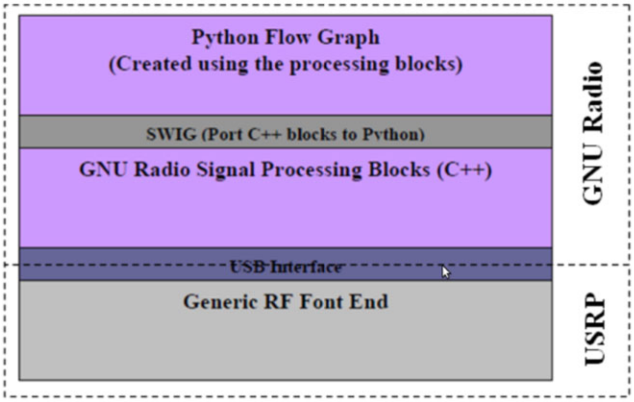


Fig. 3 Hierarchy of Software and hardware interface

with either analog circuit or analog circuit combined with digital part. Software radio is a revolution in radio design due to its ability to create radios that change easily, creating new options for users. Software radios can do pretty much anything a traditional radio can do.

Instead of a bunch of fixed circuits, everything has been bought into a single and easily alterable canopy called the software. GNU Radio is an open source software toolkit which provides a library of signal processing blocks and the glue to tie these blocks together for building and deploying software defined radios. Using GNU Radio, a radio can be built by creating a flow graph where the main components are signal processing blocks and the connection between these blocks represent the data flow between them. The signal processing blocks are implemented in C++ and the flow graphs are constructed and run in Python [5]. A block contains various parameters which include the number of input and output ports it has as well as the type of data that flows through each. All these parameters can be changed according to demand of the applications some blocks have only output ports or input ports. Input and output ports serve as data sources and sinks in the graph. The hierarchy of Software and hardware interface is given in Fig. 3 [1–3].

1.4 USRP (universal software radio peripheral)

The Universal Software Radio Peripheral (USRP) is a device which allows the creation of a software defined radio using any computer with an USB 2.0 port. Variety of daughter boards allows the USRP to be used on different radio frequency ranges. The system design of the USRP is open source. A typical setup of the USRP board [16, 17] consists of one mother board and up to four daughter boards, as shown in Fig. 4.

The USRP consists of a motherboard containing up to four 12-bit, 64 M sample/sec ADCs, four 14-bit, and 128 M sample/sec DACs, a million gates, Field Programmable Gate Array (FPGA) and a programmable USB 2.0 controller as

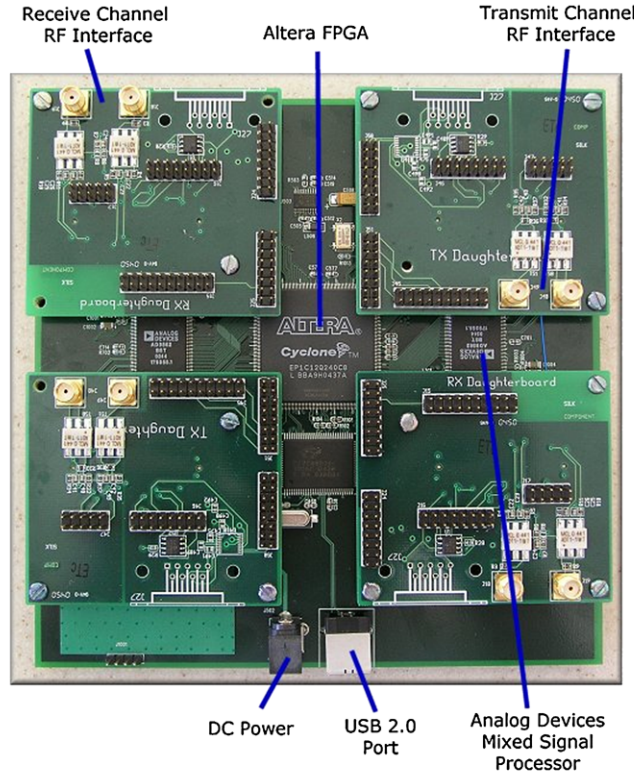


Fig. 4 Top view of Universal Software Radio Peripheral kit with four daughter boards connected [10]

shown in Fig. 4. Each fully populated USRP motherboard supports four daughter boards, two for receiving and two for transmitting. RF front ends are implemented on the daughter boards. One USRP can simultaneously receive and transmit on two antennas in real time.

All sampling clocks and local oscillators are fully coherent, thus allowing the creation of multiple input, multiple output (MIMO) systems [14]. In the USRP, high sampling rate processing takes place in the FPGA, while lower sampling rate processing occurs in the host computer. The two onboard digital down converters (DDCs) mix, filter, and decimate (from 64 M Samples/s) incoming signals in the FPGA. Two digital up converters (DUCs) interpolate baseband signals to 128 MS/s before translating them to the selected output frequency. The DDCs and DUCs combined with the high sampling rates also greatly simplify analog filtering requirements.

Daughter boards mounted on the USRP provide flexible, fully integrated RF front-ends. The USRP accommodates up to two RF transceiver daughter boards (or two transmit and two receive) for RF I/O. The comparison between two versions of USRP can be seen in Table 1.

1.5 GNU Radio companion

GNU Radio programming [10] has a steep learning curve, due to its command line interface. To aid beginners, Josh

Table 1 Comparison between USRP and USRP2 board

	USRP	USRP2
Manufacturer	Ettus Research	
ADCs	64 MS/s 12-bit	100 MS/s 14-bit
DACs	128 MS/s 14-bit	400 MS/s 16-bit
Mixer	Programmable decimation- and interpolation factors	
Max. BW	16 MHz	50 MHz
PC connection	USB 2.0 (32 MB/s half duplex)	Gigabit Ethernet (1000 MBit/s)
RF range	DC - 5.9 GHz, defined trough RF daughterboards	

Blum of Johns Hopkins University, has developed a graphical user interface (GUI) for GNU Radio. This GUI termed GNU Radio Companion (GRC) allows users to interact with GNU Radio signal blocks in a manner similar to LabVIEW or Matlab Simulink. The entire interface is completely designed with GNU Radio in mind, and encompasses over 150 blocks from the GNU radio project. Blocks are manually integrated into GRC via descriptive python definitions. The definitions are very flexible, and allow multiple GNU Radio blocks to be grouped into a single GRC super-block.

1.6 Python programming language

Python is back bone for the GNU Radio programming [6]. GNU Radio offers a framework for building signal processing applications blocks using Python language. Highly critical signal processing blocks were written in C++ codes. Graphical User Interface (GUI) GNU Radio Companion (GRC) created using Python language and it will support designer to create the Python program. GNU Radio provides a data flow abstraction. Signal processing block written in C++ can be accessed from Python through SWIG interface.

2 Analog communication systems implementation

This section covers the implementation of Analog modulation schemes such as Community Radio implementation using Frequency Modulation (FM) and Radio Data Systems implementation using GNU Radio and USRP hardware [19].

2.1 Community Radio scheme

In addition to commercial and public broadcasting, Community radio is a service oriented broadcasting model. It will serve for the certain geographical communities and for their interest. It cannot compete with the commercial broadcast service; in other words it is purposely made for sharing daily information for the farmers, daily prices all cultivated and produced goods within the region. In India, Ministry of Information and Broadcasting, provides seed fund for the educational institutions who can serve the regional people through community radio schemes. The actual cost to establish this system can be reduced by order ten because of the Open source hardware and software.

2.1.1 Frequency modulation (FM)

Angle-modulation schemes, which include frequency modulation (FM) and phase modulation (PM) belong to the class of nonlinear modulation schemes [13]. This family of modulation schemes is characterized by their high-bandwidth requirements and good performance in the presence of noise. These schemes can be visualized as modulation techniques that trade-off bandwidth for power and, therefore are used in situations where bandwidth is not the major concern and as high SNR required. Frequency modulation is widely used in high-fidelity FM broadcasting, TV audio broadcasting, microwave carrier modulation, and point-to-point communication systems. The time-domain representation of angle-modulated signals, when the carrier is $c(t) = A_c \cos(2 f_c t)$ and the message signal is $m(t)$, is given by $u(t) = A_c \cos(2\pi f_c t + 2\pi k_f \int_{-\infty}^{t} m(t)dt)$, where, k_f represent the deviation constants of FM. The frequency-domain representation of angle-modulated signals is, in general is very complex due to the nonlinearity of these modulation schemes. Assume $m(t) = a\cos(2\pi f_m t)$ and the modulated signal is of the form $u(t) = A_c \cos(2\pi f_c t + \beta_f \cos(2\pi f_m t))$, where $\beta_f = k_f a / f_m =$ Modulation index. In the case of sinusoidal message signal, the modulated signal can be represented by $u(t) = \sum_{n=-\infty}^{\infty} A_c J_n(\beta)\cos(2\pi f_c + n f_m)t$, where $J_n(\beta)$ is the Bessel function of the first kind and of order n and β. In the frequency domain,

$$U(f) = \sum_{n=-\infty}^{\infty} \left[\frac{A_c J_n(\beta)}{2} \delta\big(f - (f_c + n f_m)\big) + \frac{A_c J_n(\beta)}{2} \delta\big(f + (f_c + n f_m)\big) \right]$$

Obviously, the bandwidth of the modulated signal is not finite. However, the effective bandwidth of the signal as the

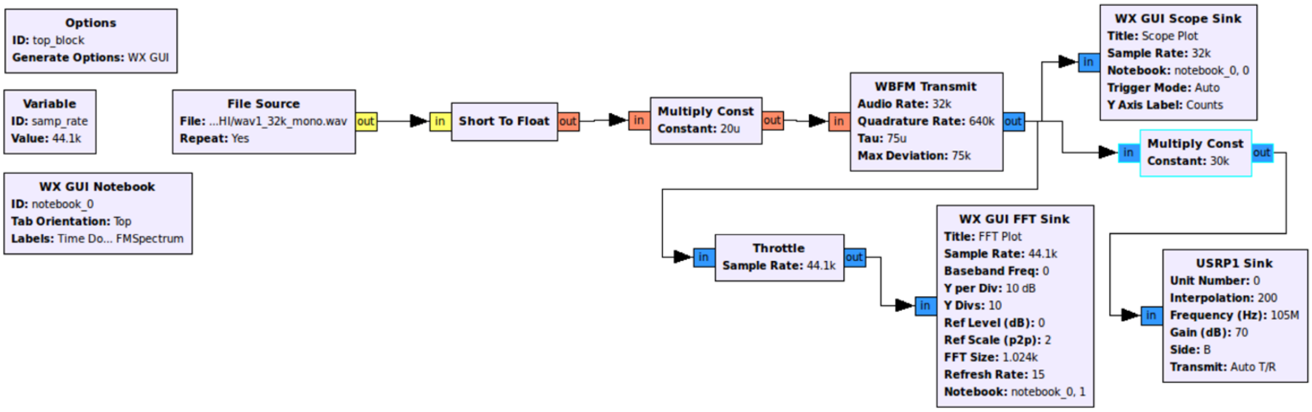


Fig. 5 Community Radio FM broadcast using USRP hardware

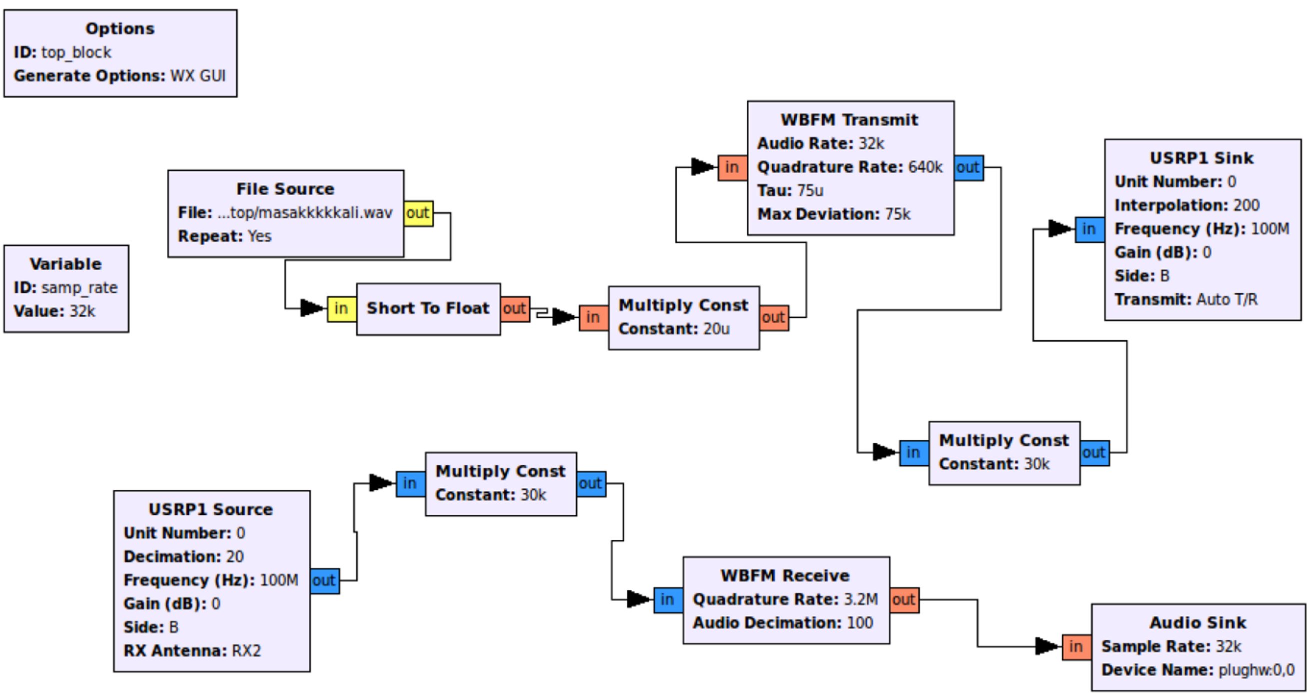


Fig. 6 FM transmission and reception using same USRP1 hardware

bandwidth is containing 98 % to 99 % of the modulated signal power. This bandwidth is given by Carson's rule as $B_T = 2(\beta + 1)W$, where β is the modulation index; W is the bandwidth of the message and B_T is the bandwidth of the modulated signal.

2.1.2 *Implementation of FM using GRC*

The implementation of the FM broadcasting using gnu-radio companion was done. The signal source taken from the file source which contains recorded programme sampled at 32 kHz. Source connected to type converter and was given as input to the Wide Band FM modulator. Digitally amplified FM signal connected to the USRP1 Sink for the broadcasting the programme.

2.1.3 *Specification*

Input Signal: File Source, Sampling Frequency = 32 kHz. Wide Band FM: Sensitivity = 75 µ, Frequency Deviation = 75 kHz, Carrier Frequency: 100 MHz, Gain = 50.

Figure 5 depicts the setup for the FM transmission using USRP1 kit and Fig. 6 is structure for the FM transmission and reception using the same USRP1 hardware kit. This hardware setup can be used for any Community Radio setup within the specific region, like Education institution, rural

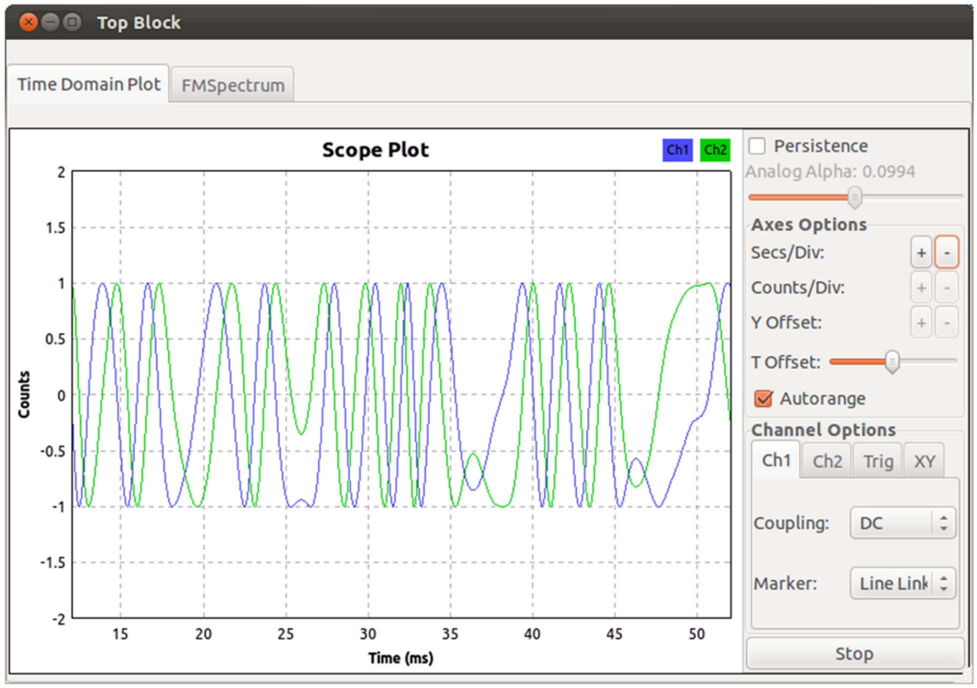


Fig. 7 FM broadcasting time domain plot

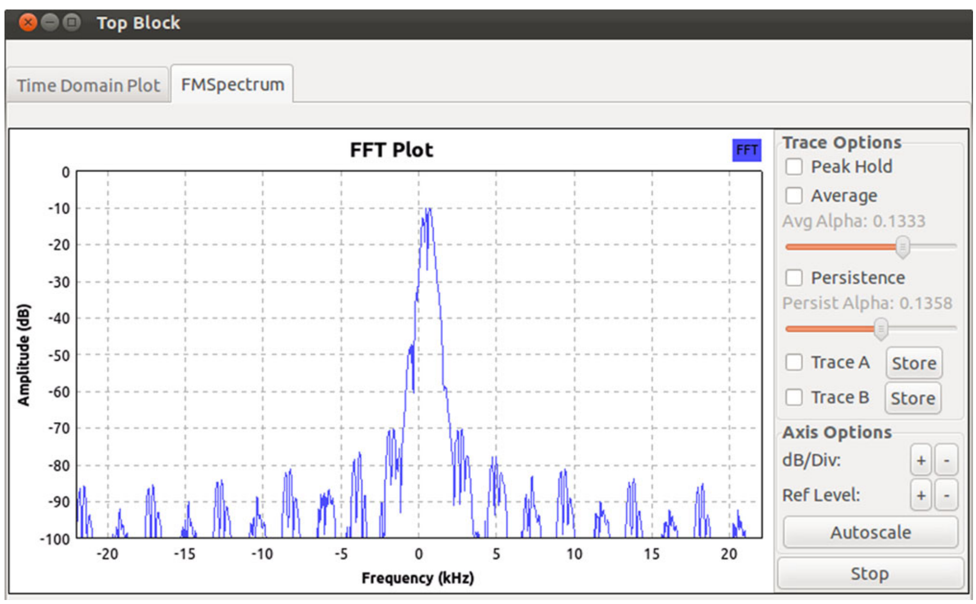


Fig. 8 FM broadcasting spectral plot

area and for service activities for the Non Governmental Organization (NGOs).

2.1.4 *Output screen*

See Figs. 7 and 8.

2.2 Radio data system (RDS) [18]

Radio Data system is device which can receive radio signal with data. RDS is built on Frequency Modulation (FM) radio broadcast transmission with many facilities. The main applications of RDS are to update the traffic status and radio station name as text form in display of digital radio. This system is manly in famous in Europe. First deployment was done at Germany onto FM using 57 kHz subcarrier [9].

Data can be added to the baseband signal and modulated using RF carrier. The baseband signal contains mono audio consisting of the left and right $(L + R)$ component and is transmitted audio frequencies up to 15 kHz. The $L - R$ component is then frequency translated as a double side band suppressed carrier (DSB-SC) at 38 kHz. A 19 kHz single tone signal used as pilot is also added with above mixture.

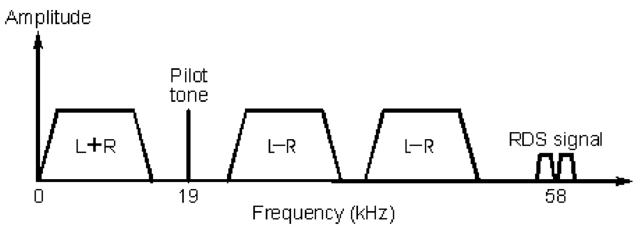


Fig. 9 The structure of the baseband signal spectrum including the RDS subcarrier

The pilot tone will be used as to enable the receiver demodulator to regenerate 38 kHz. The $L - R$ component of stereo signal is placed above the hearing range to avoid the interference with mono signal. The Data component is in 57 kHz subcarrier which is above the stereo difference $L - R$ component. This value is equal to the three times of pilot tone frequency.

The data is modulated using Quadrature Phase Shift Keying (QPSK). Phase modulation gives good immunity to data corruption caused by noise maintaining with actual transmitted data rate. The subcarrier frequency placed at the harmonic of pilot tone, it also minimizes the interference to audio signals.

2.2.1 *RDS baseband coding*

The data rate for text information is 1187.5 bits per second. Data rate is equal to RDS sub carrier divided by 48 to maintain the synchronous decoding with minimum data error. Data format consists four blocks. Each block consist 10 bit check word and 16 bit information. Approximately 11.4 groups will be transmitted with the data rate of 1187.5 bits per second. There are 16 different group structures available for different stations which will transmit different data at arbitrary times. Mixing of different data kept as minimum within groups. The first block of a group always consist the PI code and PTY and TP will be in block 2, as shown in Fig. 10.

To simplify the decoding process, first four bits of second block will be used to identify the type of group. The data was differentially encoded at the subcarrier level. As per the differential coding scheme if the input data level is "0" the output remains same and when a "1" at input the output changes its state. The power density close to 57 kHz is limited by encoding each binary digit as bi-phase signal. The encoded data is passed through the low pass filter.

2.2.2 *Hardware setup*

See Fig. 11.

2.2.3 *Acronyms for RDS Radio data system*

The common RDS acronyms and abbreviations relate to the various operational modes and facilities. These RDS acronyms are widely used in view of the fact that within countries that have adopted the system, most car radios today incorporate RDS as standard. The table provides information about the various possible services along with the FM transmission (Table 2).

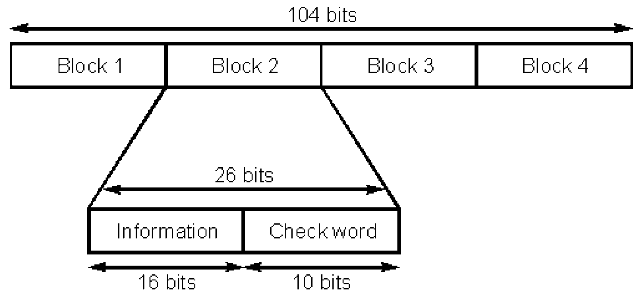


Fig. 10 RDS data format

Fig. 11 RDS reception using FM enabled mobile phone

2.2.4 *Hardware and software specification*

Mother Board: USRP1
Daughter Board: WBX Board
Input data: Wave File, Sampled at 32000 Hz
GNU Radio version: 3.4.2

2.2.5 *GNU Radio block diagram*

See block representation in Fig. 12 and output in Fig. 13.

3 Digital communication systems implementation

In this section implementation of the modern digital communication such as Digital Video Broadcasting (DVB) and OFDM based data communication systems setup using USRP1 will be discussed.

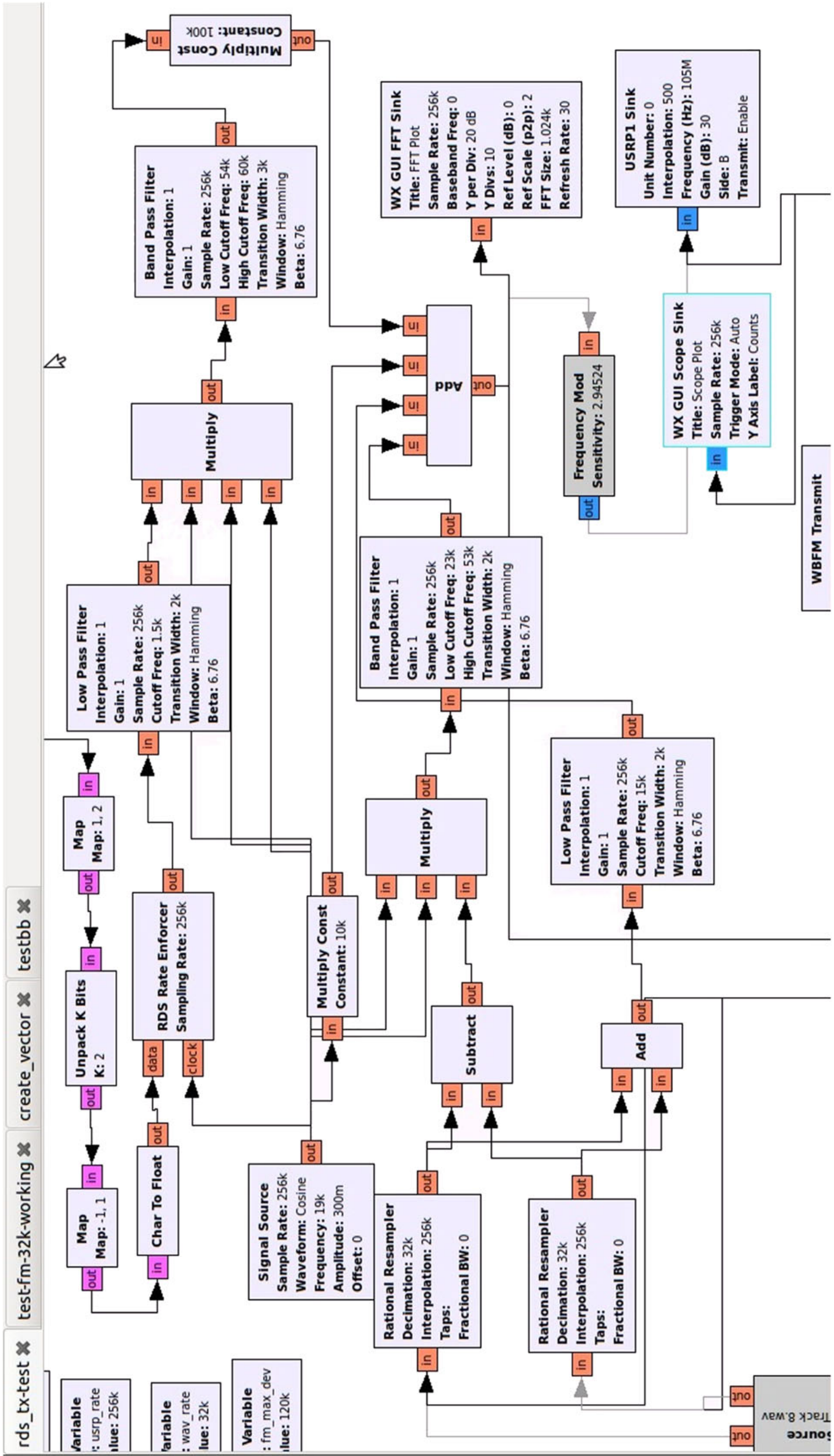


Fig. 12 GRC block for RDS FM transmitter using USRP1. (Source file uploaded at: https://docs.google.com/open?id=0B5aOpaYPZoL8b3A5Rk5HOWpVQTA) or http://nlp.amrita.edu:8080/SDR4U/files/RadioDataSystemTransmitter.grc

Table 2 Acronyms for the RDS

AF	Alternative frequencies	A list of a station's frequencies in adjacent transmitter areas
CT	Clock time and date	Data containing time and date information so that the receiver can display. Using this signal has the advantage over a self-contained clock in the radio that it adjusts itself to changes between BST and GMT (UTC) and it does not loose or gain time.
DI	Decoder information	This signal allows for miscellaneous function in the radio to be controlled.
EON	Enhanced other networks	Information transmitted giving the radio a cross-reference to other stations for Travel service and other features.
MS	Music/speech	This allows for the relative levels of speech and music to be altered.
PI	Programme identification	This is a station code used in conjunction with AF data to provide automatic tuning to the best signal for a chosen service.
PIN	Programme identification number	This signal identifies a given programme and allows the radio to turn itself (and possibly a recorder) on for that programme.
PTY	Programme type selection	A signal which allows for the selection of listening by one of 15 types of programme rather than by the station.
PS	Programme service name	A signal which enables the name of the station to be displayed.
RT	Radio text	This allows information about the programme to be displayed by the radio.
TDC	Transparent data channel	This allows for data to downloaded via RDS.
TP/TA	Travel service	These signals enable the travel information to be heard, regardless of the choice of listening.

3.1 Simple DVB with GNU Radio

It is a simple video broadcasting setup based on Gstreamer, GNU Radio and the Universal Software Radio Peripheral (USRP) [20]. It is not the standard of DVB-T/DVB-S and is simple web camera stream on the air. An UVC compliant webcam is used as a cheap but efficient video source. Current webcams can provide video at HD resolutions, though the image quality is not as good as with a good High Definition camcorder.

3.1.1 Gstreamer

Gstreamer is software framework written in C for the video processing. This tool will capture the video frames, adding text overlay, source coding (compressing) and channel coding and multiplexing with transport stream. It is pipeline-based multimedia frame work by C language based on GObject. It will help programmer to do audio playback, combined audio and video playback, streaming, editing and recording of video captured by any digital camera. The pipelining concept is the backbone for the video editing, media players and stream media broadcasters. It is cross platform tool, can work on Linux, Macintosh, Solaris, Microsoft Windows and OpenBSD. Many programming languages like Python, Perl, C++ and Ruby can also interact with framework.

3.1.2 Conceptual prototype

This setup uses the video to MPEG-TS/H.264 and uses GMSK modulation schemes. No error correction schemes used. GNU Radio and Gstreamer are interlinked using the named pipes. This setup is robust and functional, even it suffers by MPEG-TS muxer don't support the CBR, which actually needed by USRP with GNU Radio, Figs. 14, 15. This tool freezes every half the second and it will be buffered in memory temporarily.

3.1.3 Procedure to execute the setup

Prerequisites

1. To install the Gstreamer librarier in the computer, type following command
   $ sudo apt-get install gstreamer-tools
2. To install the video player "mplayer", type in terminal
   $ sudo apt-get install mplayer
3. Procedure to Run Application
   3.1 Open the 'gnuradio-companion" in on terminal and open "gmsk_sim.grc"
   3.2 Open Terminal in Linux
       % Removing Existing txfifo.ts and rxfifo.ts
       $ rm txfifo.ts
       $ rm rxfifo.ts
   3.3 Manually create txfifo.ts and rxfifo.ts
       $ mkfifo txfifo.ts
       $ mkfifo rxfifo.ts

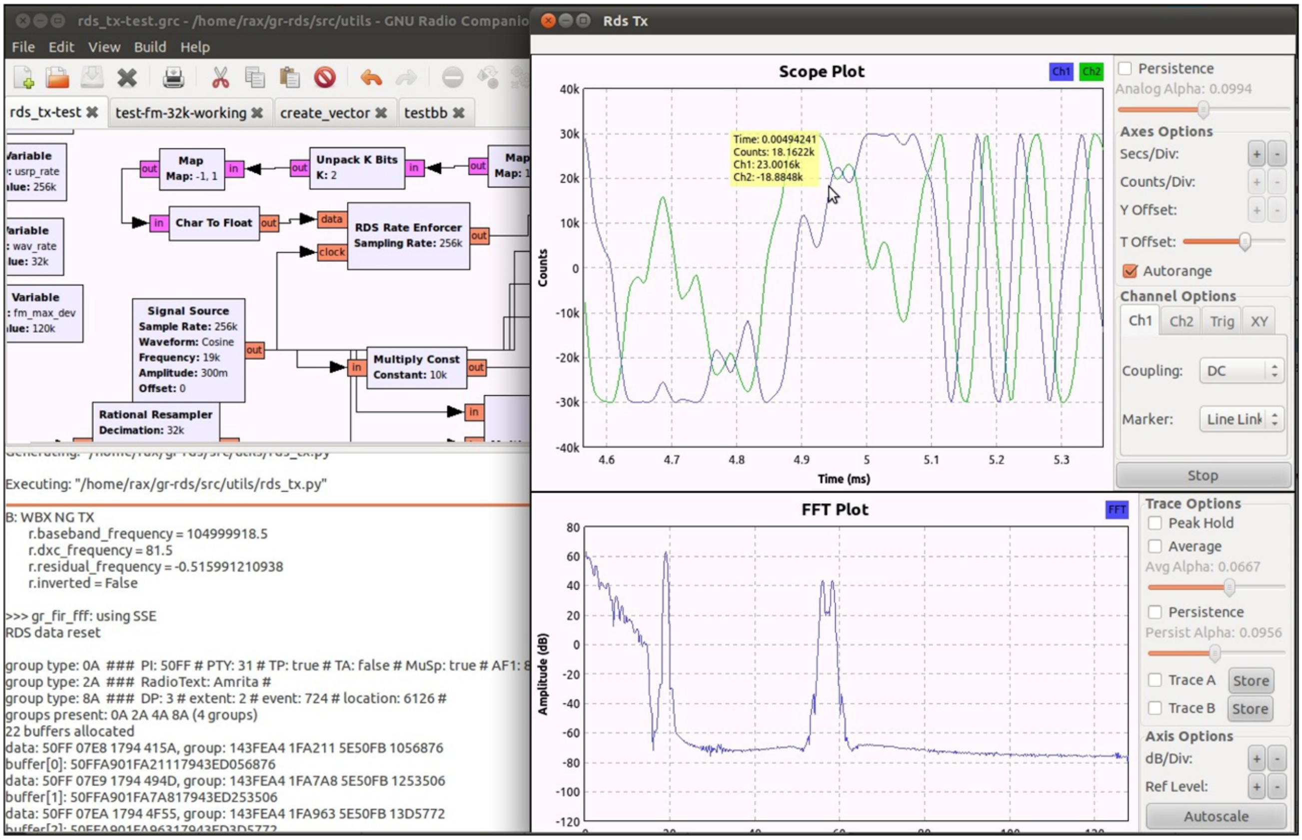


Fig. 13 Time domain and spectrum plot of RDS

Fig. 14 Experimental setup model

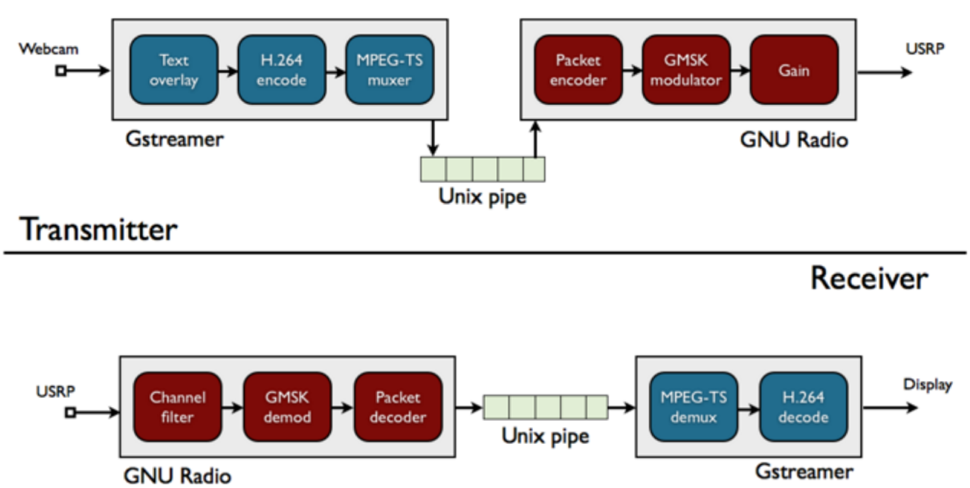


Fig. 15 Conceptual block diagram of DVB setup

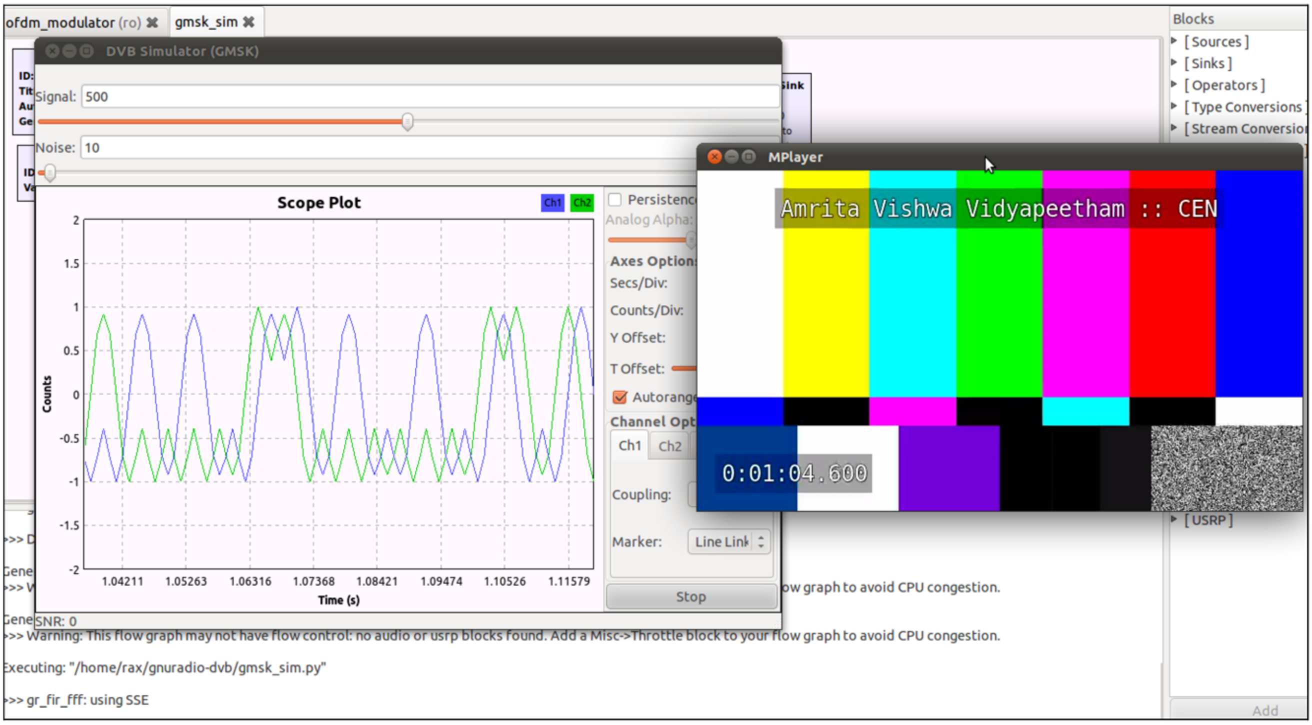


Fig. 16 Internally generated video and captured signal

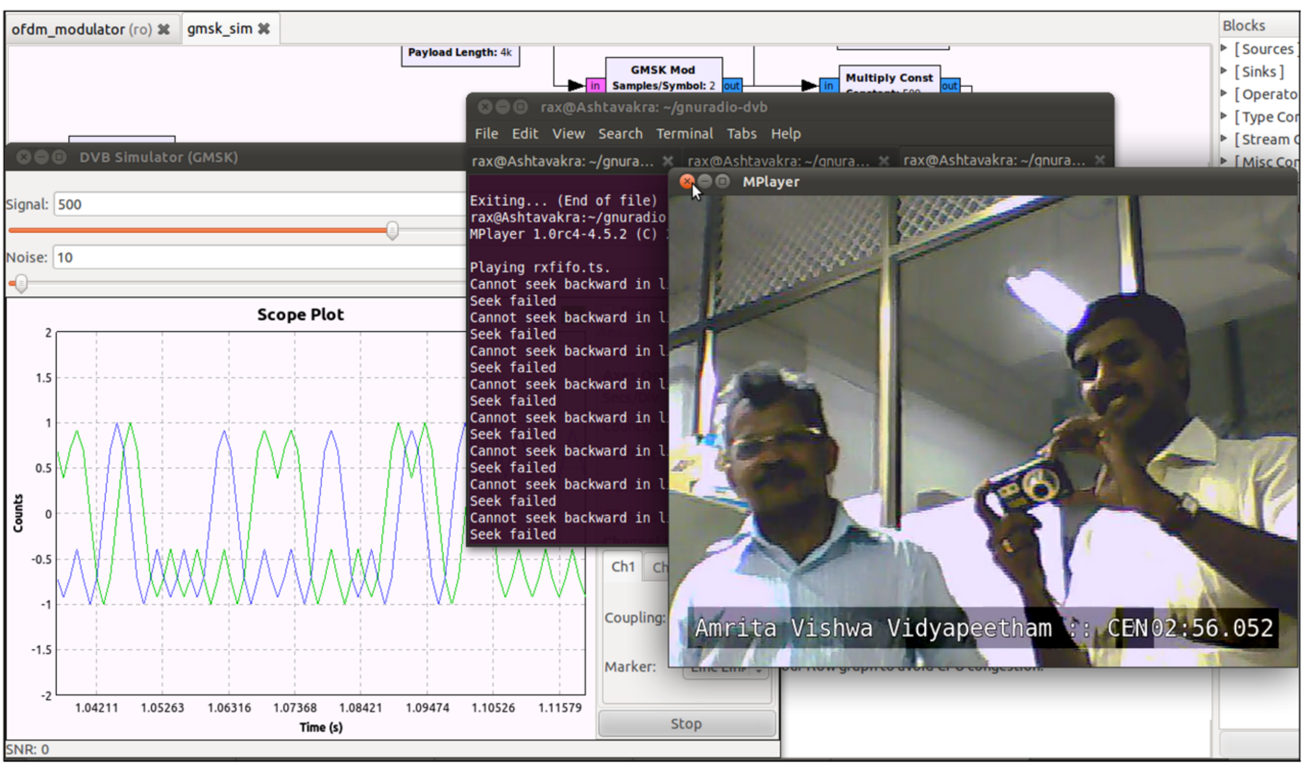


Fig. 17 Video capturing by web camera reception using PC

4. If default video stream to be generated means, in terminal type following,

   $ gst-launch -e videotestsrc ! video/x-raw-yuv, framerate = 25/1, *width* =640, *height* = 360 *! timeoverlay halign=left valign* = *bottom shaded-background* = *true ! textoverlay text* = *"Amrita Vishwa Vidyapeetham:: CEN" halign* = *center valign* = *top shaded-background* = *true ! x264enc bitrate* = 498 *! mpegtsmux ! filesink location* = *txfifo*.ts

   4.1 To capture the video stream from WEB CAM, in terminal type following,

   $ gst-launch -e videotestsrc ! video/x-raw-yuv, framerate = 25/1, *width* = 640, *height* = 360 *! timeoverlay halign* = *left valign* = *bottom shaded-background* = *true ! textoverlay text* = *"Amrita Vishwa Vidyapeetham:: CEN" halign* = *center valign* = *top shaded-background* = *true ! x264enc bitrate* = 498 *! mpegtsmux ! filesink location* = *tx-fifo*.ts
5. Open another terminal and type following

   $ mplayer rxfifo.ts
6. Now run the "gmsk_sim.grc". In the screen txed video stream will be captured and will be played using "mplayer".

Fig. 18 Experimental lab setup

3.1.4 Simulator GRC block

See result in Fig. 16 and experimental setup in Figs. 17, 18.

3.2 OFDM based data communication using USRP1

For the high data rate communication one of the efficient multi carrier communication techniques OFDM can be used. In this work, the data communication done using OFDM based communication systems using USRP1.

3.2.1 Orthogonal frequency division multiplexing (OFDM)

OFDM is a multi-carrier modulation technique where data symbols modulate a parallel collection of regularly spaced

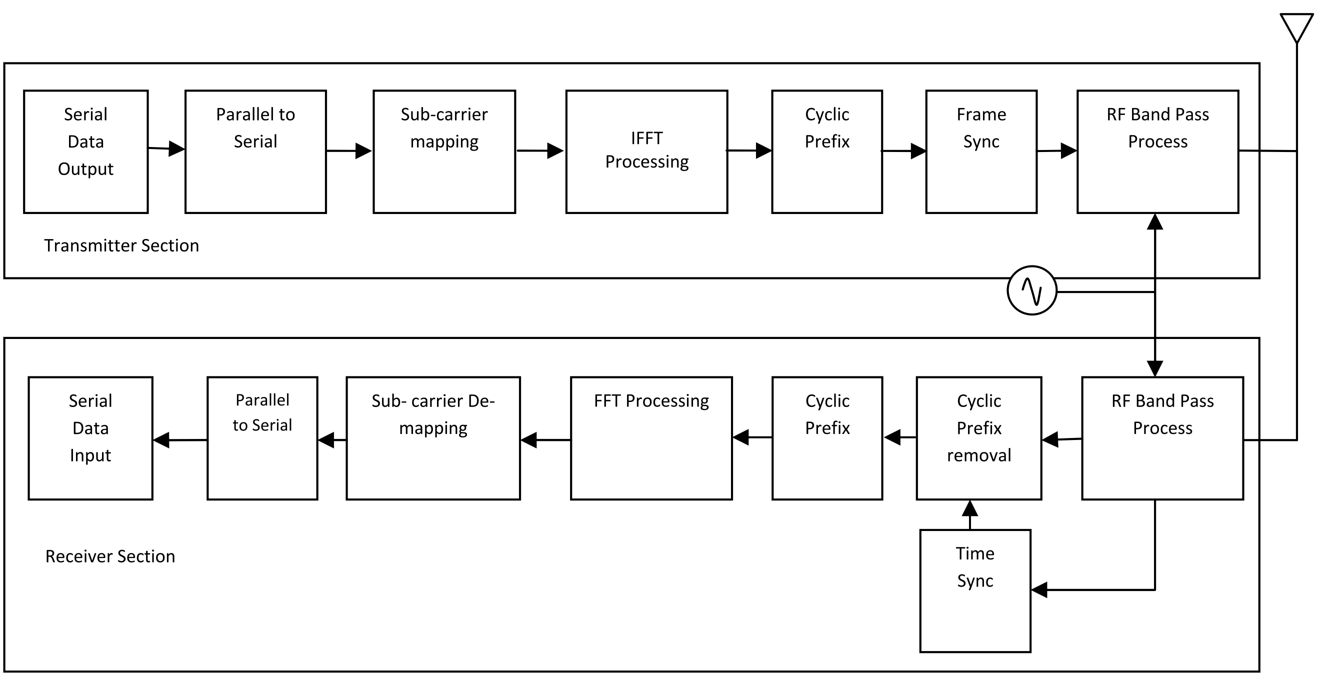


Fig. 19 Fundamental block diagram of OFDM communication systems [12]

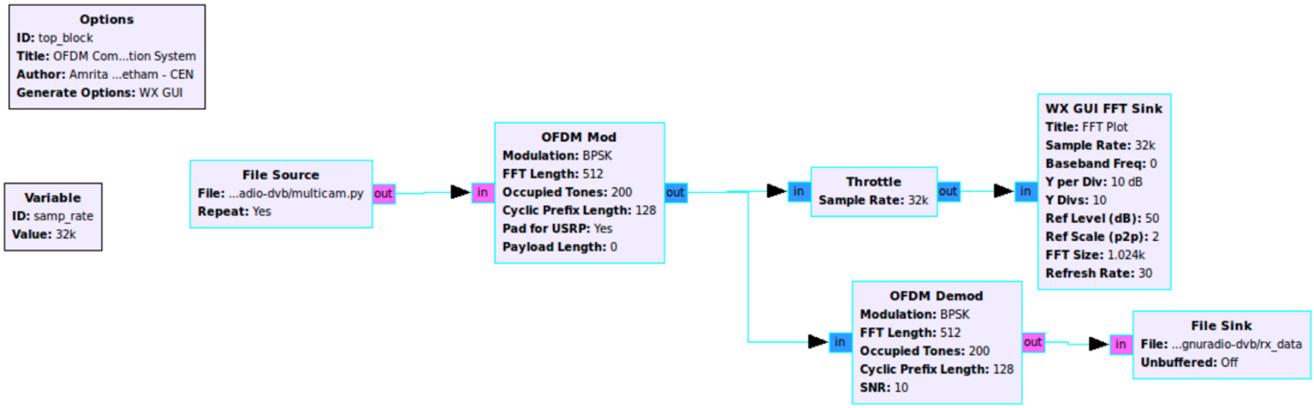
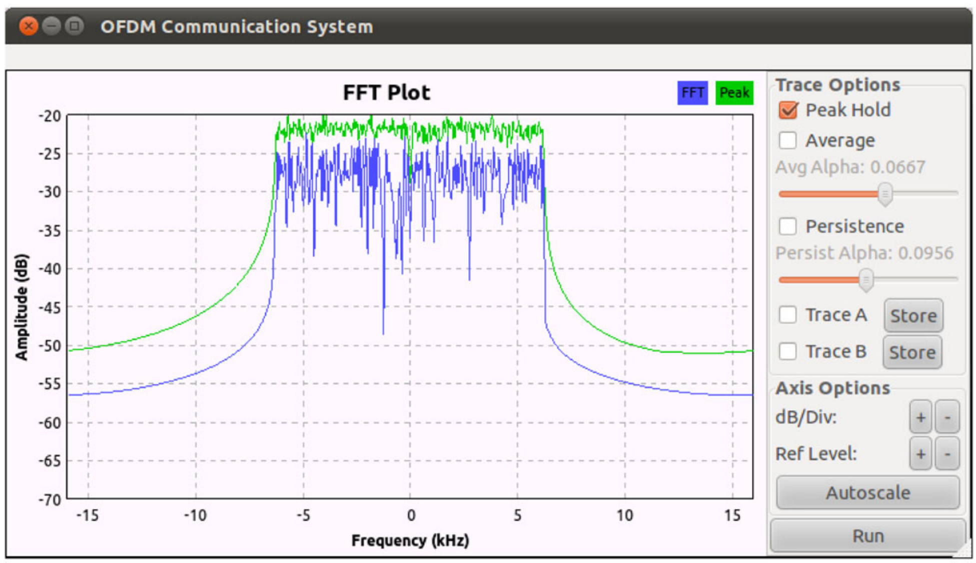


Fig. 20 GRC implementation of basic OFDM based communication

Fig. 21 OFDM spectrum plot



sub-carriers [24]. The sub-carriers have the minimum frequency separation required to maintain orthogonality of their corresponding time domain waveforms, yet the signal spectra corresponding to the different sub-carriers overlap in frequency, Fig. 19. The spectral overlap results in a waveform that uses the available bandwidth with very high bandwidth efficiency. OFDM is simple to use on channels that exhibit time delay spread or, equivalently, frequency selectivity. Frequency selective channels are characterized by either their delay spread or their channel coherence bandwidth which measures the channel de-correlation in frequency. The coherence bandwidth is inversely proportional to the root-mean-square (rms) delay spread. By choosing the sub-carrier spacing properly in relation to the channel coherence bandwidth, OFDM can be used to convert a frequency selective channel into a parallel collection of frequency flat sub channels. Techniques that are appropriate for flat fading channels can then be applied in a straight forward fashion [7, 8, 11].

3.2.2 Specification

Mother Board: USRP1
Daughter Board: WBX Board
Input data: Text file (∗.txt format)
GNU Radio version: 3.4.2

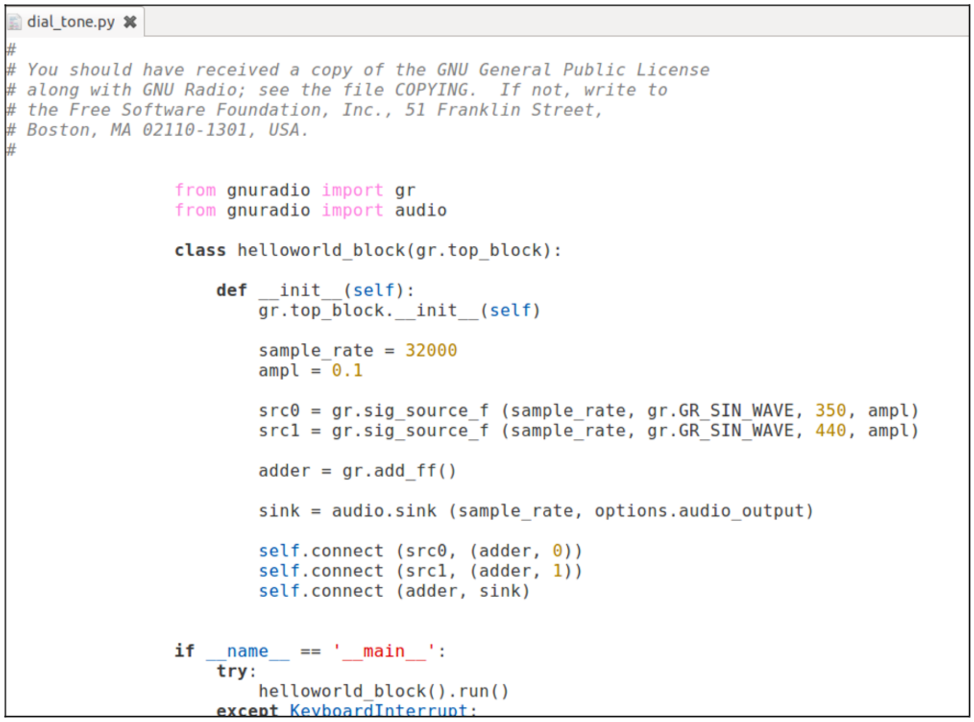
Since this software is open source, applications development is highly depend on versions of GNU Radio tool.

3.2.3 GRC block diagram of OFDM communication

See Figs. 20, 21, 22.

4 Conclusions

This work will provide the insight about the modern analog and digital communication system such as Community Radio, RDS, Simple DVB and OFDM based data communication with the real time implementation using USRP1. The



```
dial_tone.py
#
# You should have received a copy of the GNU General Public License
# along with GNU Radio; see the file COPYING.  If not, write to
# the Free Software Foundation, Inc., 51 Franklin Street,
# Boston, MA 02110-1301, USA.
#

        from gnuradio import gr
        from gnuradio import audio

        class helloworld_block(gr.top_block):

            def __init__(self):
                gr.top_block.__init__(self)

                sample_rate = 32000
                ampl = 0.1

                src0 = gr.sig_source_f (sample_rate, gr.GR_SIN_WAVE, 350, ampl)
                src1 = gr.sig_source_f (sample_rate, gr.GR_SIN_WAVE, 440, ampl)

                adder = gr.add_ff()

                sink = audio.sink (sample_rate, options.audio_output)

                self.connect (src0, (adder, 0))
                self.connect (src1, (adder, 1))
                self.connect (adder, sink)


        if __name__ == '__main__':
            try:
                helloworld_block().run()
            except KeyboardInterrupt:
```

Fig. 22 Received data using OFDM

procedure to implement these experiments was discussed and demonstrated in details with results. The GNU Radio based USRP hardware implementation can be used as the test bed for many advanced protocol implementation in real time before going for any hardware mass production. The GNU Radio platform can also be used in Virtual Lab implementation, Deep space imaging, Ionosphere study, Aviation mapping application and Weather Satellite data reception. This work will assist to convert any theoretical study into physically realizable one [21–23].

Acknowledgements This work is a part of Ministry of Human Resources and Development (MHRD), Government of India, funded research project. We wish to thank Mr. Rakesh Peter and Mr. Senthil Murugan for their kind support.

References

1. Aravind, H., Gandhiraj, R., Soman, K. P., Sabarimalai Manikandan, M., & Rakesh, P. (2011). Spectrum sensing implementations for software defined radio in simulink. In *International conference on communication technology and system design 2011, Elsevier-Procedia engineering*, Dec. 7–9.
2. Abinav, A. k., Naveena, K., Pratibha, R., Gandhiraj, R., & Soman, K. P. (2010). SVM based classification of digitally modulated signals for software defined radio. In *International conference on embedded systems 2010*, July 13–15. Coimbatore: Coimbatore Institute of Technology.
3. Abinav, A. k., Naveena, K., Pratibha, R., Gandhiraj, R., & Soman, K. P. (2010). Detection and classification of signals to configure software defined radio. In *Second national conference on recent trends in communication, computation and signal processing*, March 26–27.
4. Al-kamali, F. S., Dessouky, M. I., Sallam, B. M., Shawki, F., & Abd El-Samie, F. E. (2011). Impact of the power amplifier on the performance of the single carrier frequency division multiple access system. *Journal of Telecommunication Systems*. doi:10.1007/s11235-011-9439-y.
5. Blossom, E. How to write a signal processing block. http://www.gnu.org/software/gnuradio/doc/howto-write-a-block.html.
6. Shen, D. (2005). http://www.snowymtn.ca/GNURadio/GNURAdioDoc-5.pdf.
7. Fuxjäger, P., Costantini, A., Valerio, D., Castiglione, P., Zacheo, G., Zemen, T., & Ricciato, F. IEEE 802.11p Transmission using GNURadio. http://userver.ftw.at/~valerio/files/wsr10.pdf.
8. Leipold, F., Tassetto, D., & Bovelli, S. (2011). Wireless in-cabin communication for aircraft infrastructure. *Journal of Telecommunication Systems*. doi:10.1007/s11235-011-9636-8.
9. GNU radio introduction. http://gnuradio.org/redmine/projects/gnuradio/wiki/WhatIsGR.
10. GNU radio companion. http://www.gnuradio.org/trac/wiki/GNURadioCompanion.
11. Garrido-Cantos, R., De Cock, J., Martínez, J. L., Van Leuven, S., & Garrido, A. (2011). Video transcoding for mobile digital television. *Journal of Telecommunication Systems*. doi:10.1007/s11235-011-9594-1.
12. Image reference. http://www.microwavejournal.com/legacy_assets/FigureImg/4324_Fig3_L.jpg.
13. Proakis, J. G., & Salehi, M. (2007). *Fundamentals of communication systems*. Pearson Prentice Hall, India.
14. Chung, J., Yun, Y., & Choi, S. (2011). Experiments on MIMO-OFDM system combined with adaptive beam forming based on IEEE 802.16e WMAN standard. *Journal of Telecommunication Systems*. doi:10.1007/s11235-011-9475-7.
15. Dillinger, M., Madani, K., & Alonistioti, N. (2005). Software-defined radio basics. *Review of Software Defined Radio: Architectures*, *6*(10).
16. Ettus, M. *USRP user's and developer's guide*. Ettus Research LLC.

17. Dickens, M., Laneman, N.J., & Dunn, B. P. (2012). Seamless dynamic runtime reconfiguration in a software-defined radio. *Journal of Signal Processing Systems (Springer)*, *69*(1), 87–94.
18. RDS fundamentals. http://www.radio-electronics.com/info/broadcast/rds/radio-data-system-technology-operation.php.
19. Ranjini, R., Gandhiraj, R., & Soman, K. P. (2011). Analog and digital modulation toolkit for software defined radio. In *International conference on communication technology and system design 2011, Elsevier-Procedia engineering*, Dec 7–9.
20. Simple DVB Experimental Setup. http://wiki.oz9aec.net/index.php/Simple_DVB_with_Gstreamer_and_GNU_Radio.
21. Prasad, S. S., Gandhiraj, R., & Soman, K. P. (2011). Multi-user spectrum sensing based on multitaper method for cognitive environments, In *International Journal of Computer Applications (IJCA)*. New York: Foundation of Computer Science. doi:10.5120/2613-1093.
22. Prasad, S. S., Gandhiraj, R., & Soman, K. P. (2010). Efficient spectral estimation with Slepian tapers in cognitive environment: a review. In *Second national conference on recent trends in communication, computation and signal processing*, March 26–27.
23. Prasad, S. S., Gandhiraj, R., & Soman, K. P. (2010). Cognitive radio as a background for spectrum sensing—a review. In *National conference on recent innovation in technology 2010 (NCRIT 2010)*, Kottayam, Kerala, March 04–06. Organized by Rajiv Gandhi Institute of Technology (Govt. Engineering College).
24. Li, Y., & Stuber, G. L. (2006). *Orthogonal frequency division multiplexing for wireless communication*. Berlin: Springer.

R. Gandhiraj is Assistant Professor in Department of Electronics 8 and Communication Engineering, Member of Communication Engineering and Research Group, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India. He did his B.E. (1995–1999) in ECE in Bharathiar University, Coimbatore. He completed his M.Tech. (2004–2006) in Digital Systems and Communication at National Institute of Technology Calicut (NITC), Kerala. He is having totally 12 years of teaching and research experience with 15 International and National Journals, Conferences publications. Currently, he is pursuing Ph.D. on Cognitive Radio under the guidance of Dr. K.P. Soman, Head, Center for Excellence in Computational Engineering and Networking, Amrita Vishwa Vidyapeetham, Coimbatore. Personal Web Link: https://sites.google.com/site/gsrgandhiraj/.

K.P. Soman is Professor and Head, Center for Excellence in Computational Engineering and Networking, Amrita Vishwa Vidyapeetham, Coimbatore. He has published over 125 papers in International journals and conferences. His areas of interest include high performance computing, Machine Learning, wavelet transform, Computational Linear Algebra, Theory of Convex optimization and Software Defined Radio (SDR). He has authored many famous books, *Insight into Wavelets*, *Insight into Data Mining: Theory and Practice*, *Machine Learning with SVM and other kernel methods* and *Digital Signal Processing—The Sparse way*. Web Link: http://www.amrita.edu/cen/soman.php.