2013 International Conference on Advanced Computing and Communication Systems (*ICACCS* -2013), Dec. 19 – 21, 2013, Coimbatore, INDIA

Real-Time Communication System Design using RTL-SDR and Raspberry Pi

Danymol R, Ajitha T
Centre for Excellence in Computational
Engineering and Networking,
Amrita Vishwa Vidyapeetham,
Coimbatore-641112, India
{danymol.r, ajitha.tharakan}@gmail.com

Gandhiraj R
Department of Electronics and Communication
Engineering,
Amrita Vishwa Vidyapeetham,
Coimbatore-641112, India
r_gandhiraj@cb.amrita.edu

***Abstract*— RTL-2832U USB receiver - Raspberry Pi setup receives and decodes the FM signals and sends to another computer across the Local Area Network. It can be realized by means of SDR-Sharp software in remote PC; hence it can be used for communication. Raspberry Pi is a low cost, credit-card sized computer. Whereas RTL-2832U USB receiver has a mode where it dumps raw IQ samples over USB to the computer. The GNU Radio experiments supports this relatively cheap USB receiver as a software defined radio. Using GNU Radio Companion, all real-time communication experiments can be reproduced.**

Keywords— RTL-SDR; Raspberry Pi; GNU Radio; SDR Sharp

I. Introduction

In this communication system, RTL-SDR and Raspberry Pi combination receives the FM signals and transfers it to a remote PC. SDR Sharp software installed in remote computer helps to observe the received signals. Raspberry Pi is a credit card sized, low cost computer which is capable of performing almost all the tasks as our desktop PC does. RTL-SDR is a low cost Software Defined Radio, in which the components are programmed and implemented in software instead of the conventional way of hardware implementation. GNU Radio is an open source free toolkit, used to create software-defined radios. Raspberry Pi acts as a Remote Server for RTL2832u SDR. The RF front-end followed by an analog-to-digital converter constitutes the RTL-SDR, which provides samples to a host computer. The rest of the processing is done in software.

One of the important feature of the RTL driver for the RTL2832U SDR stick is the TCP server which allows to send data from the RTL2832U stick across home network (LAN connection) to a remote PC running a program to process the data back into SDR information such as SDR Sharp. SDR Sharp is a simple, small visceral and fast PC-based Digital Signal Processing application for Software Defined Radio. Even if the Raspberry Pi is not quite powerful enough to run current programs to decode and process the SDR data from the RTL2832U stick, it is well capable of running the rtl_tcp server. This means that the RTL2832U stick can be directly plugged into the Raspberry Pi and wind up with a very small and portable SDR radio server. Raspberry Pi can either be plugged directly to your router or use Wi-Fi for more flexibility in placement. If decided to go through the Wi-Fi route, it is suggested to use wireless N since bandwidth can be an issue. A Wi-Fi dongle can either be used on the Raspberry Pi's USB port or a Wi-Fi gateway interface. The nice thing about the Wi-Fi gateway is that it needs no drivers to connect to the Raspberry Pi since it converts Wi-Fi data to Ethernet. If decided to go through dongle route, that dongle should be compatible with Raspberry Pi.

II. Raspberry Pi

Raspberry Pi is a Linux-based, single-board computer (costs around 30 dollars) having a 700MHz ARM architecture CPU, 256MBytes of RAM, two USB ports, and a 10/100 Ethernet controller which would improve the thinking capacity of the people. Like any other computer, the Raspberry Pi also uses an operating system and here the operating system is a flavor of Linux called Raspbian. And there are many different flavors of the same just similar to Raspbian. Linux is a highly preferred match for Raspberry Pi because it is open source and free. On one hand, it keeps the price of the platform low and on the other, it makes it more ethically hackable. The operating system runs from an SD flash card, allowing it to instantly be switched by swapping cards. Its potential uses are astonishing, and as yet, not fully researched, but it has already been tested as a multimedia player with pelting capabilities, an internet browser, a games machine and a hardware development board. It is meant for people of all ages and skill levels, especially for educational purposes. The General Purpose Input/Output (GPIO) expansion board for Raspberry Pi allows any external peripherals to connect to this device. This board enables controlling light-emitting diodes (LEDs), servo motors, sensors and other electronic components. The Raspberry Pi has two interfaces which will allow us access to control things.

[978-1-4799-3506-2/13/$31.00 ©2013 IEEE]

The first of these is a kind of standard interface which almost built in all modern computers, called a USB or Universal Serial Bus. The second of these interfaces on the Raspberry Pi is called the GPIO which stands for General Purpose Input and Output. This part of the Raspberry Pi allows the user to easily control all manner of devices, in turn helps to implement so many electronics experiments [1]. In this experiment, USB port is used to interface the RTL-SDR dongle.

III. RTL-SDR

The RTL-SDR is an inexpensive software-defined radio based on DVB-T TV tuners with RTL2832 chips. The RTLSDR can also act as a wide band radio scanner. It will attract the hardware hackers, tinkerers, Ham Radio enthusiasts, and anyone who is interested in RF. RTL-SDR receivers use a Realtek RTL2832 quadrature sampling detector in combination with a programmable oscillator (most commonly an Elonics E4000, Fitipower FC0013, FC0012, FC2580, and R820T). Experiments indicate that these devices perform fairly well without any hardware modification, and the software (drivers, firmware, and user interfaces) is sophisticated. It was initially possible to tune a desired frequency, sample a chunk of spectrum, and write the data to a file readable by a conventional SDR program. For example, GNU Radio or HDSDR would be used to read the file and demodulate signals in a 2 MHz wide band of recorded spectrum. Recent generations of RTLSDR compatible software can directly access the data stream providing excellent real-time multimode reception [2]. RTLSDR hardware consists of USB, PCI and antennas .The USB dongle used here is based on the Realtek RTL2832U chip, which is an ADC and USB data pump. Figure 1 shows the RTL-SDR USB dongle. The Elonics E4000 is the cheaper and most desirable tuner which is capable of tuning in the range of 60MHz to 1700MHz. RTL2832U chip is used with one among the three tuners chips: FC0012 - Tuner 50MHz to 1000MHz; FC0013 - Tuner 50MHz to 1700MHz; R820T-Tuner 24MHz to 1850MHz. The E4000 Elonics chip is a digitally controlled tuner which is shown in figure 2. The tuner selects a frequency band around a target center frequency and demodulates it to the baseband. In this experiment, the RTL2832U chip serves as an analog to digital converter that samples the baseband signal and outputs the samples to a host computer through a Wired-Local Area Network connection. This dongle was originally made to receive and decode the FM/AM/SSB signals. Antii Palosaari, a V4L/DVB kernel developer, discovered that there is a device mode in which raw samples can be captured and transferred to a host computer. This feature enables this device to be used as an inexpensive Software-defined radio [3].

GNU Radio contains a lot of signal processing blocks which is useful to build simulation like environment, which in turn can be helpful to implement wireless communication and real-time communication experiments [4]-[7]. GNU Radio applications are primarily written using the Python programming language, while the signal processing blocks are implemented in C++ language. By installing and using RTLSDR hardware and software we have now entered the long established world of scanning and receive only Ham Radio. Here instead of using expensive purpose built hardware, a PC (which you no doubt already own) and a 20 dollar USB dongle is used. It is now possible to listen to:

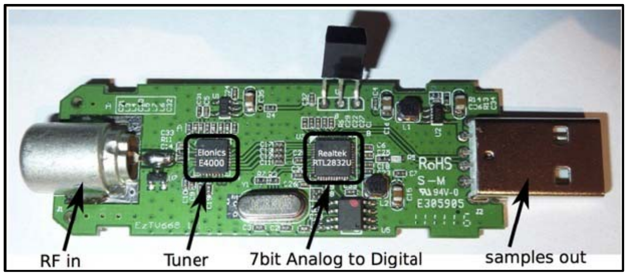

Fig. 1. RTL-SDR USB Dongle

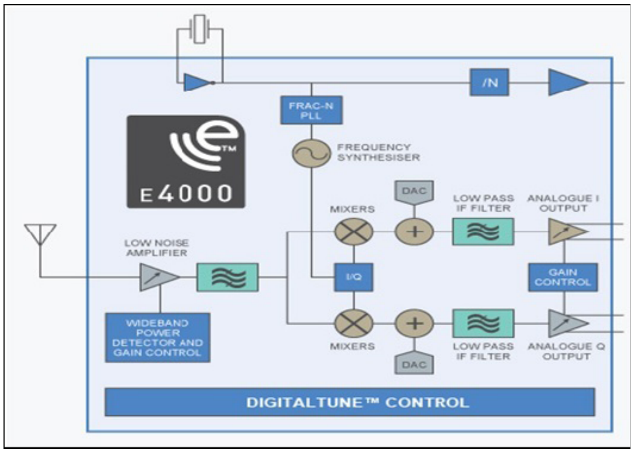

Fig. 2. Elonics E4000 Tuner/Quadrature Sampling Detector Diagram

1) AM: Almost all AM transmissions are below the bottom frequency of RTL-SDR dongles so a translator will be needed to get these frequency bands.
2) Upper/Lower Sideband (USB/LSB): Same as AM frequency band reception.
3) FM: Both narrow band and wideband frequency band can be obtained. The former is used on two way radio systems such as emergency services and private radio networks (like couriers and taxis) and the latter is the usual broadcast FM that can be used in kitchen and car. Aircraft, boats and ships can also utilize RTL-SDR to listen to narrow band FM. The SDR sharp software can receive both narrowband and wideband FM and the latter do stereo FM too!
4) CW: Continuous wave for Morse code enthusiasts [8].
5) With GNU Radio digital modes such as pagers (POCSAG) ADS-B (aircraft positions), AIS (ship

positions), AP25 and TETRA (digital trunk radio) and many others can be received and demodulated.

6) Satellite reception including receiving HAM transmissions from the International Space Station are possible too. RTLSDR and a 2.5m dish can be used to track the carrier signal on deep space robots such as Voyager and the Mars missions.

7) GPS reception is currently being worked on, in future it would be able to do.

A 'C' based software, SDR Sharp, is used to process the data in remote PC. It can demodulate AM/WFM/NFM/DSB/SSB signals. The main purpose of SDR Sharp is to offer a simple proof of concept application to get hands into DSP techniques.

The Auto-tuner in SDR Sharp works just like a wideband monitor. A threshold level can be set and any signal that exceeds will cause the software to automatically jump to that peak and demodulate the audio. It will even log the frequency in a MS Excel spreadsheet. A roll span and dwell time can also be set. The dongle will jump sequentially through the predefined range and autotune to each peak encountered. This in effect becomes an extremely fast scanner/spectrum analyzer.

IV. RASPBERRY PI CONNECTION WITHOUT KEYBOARD AND MONITOR

In cases where no keyboard or screen available for your Raspberry Pi, it can be very useful to remotely connect using a nearby network and a laptop (figure 3). If Raspberry Pi need to be connected without connecting it to a separate keyboard and monitor, perhaps to run a script or try out a little python program. This is quite easy using Putty in windows in which SSH is used to provide access to the Raspberry Pi's terminal. SSH (Secure SHell) is simply the name of the network protocol used to transfer this data, it can work through your home network, or even using a serial link. Regardless of the connection which is used (select the correct IP or COM port), the Putty or a similar program can be used to connect to the Raspberry Pi and run most programs which usually run within the terminal (such as bash scripts or python or nano etc).

1) To switch on (or off) SSH , raspi config (just type *sudo raspi-config* from the terminal) can be accessed and select SSH in the menu .

2) Using Raspi-config enable/Disable SSH. Then Raspberry Pis IP address can be entered (if connecting via the network), which can be find out by typing hostname-I.

3) Putty Configuration for given IP address (eg: Putty Configuration for IP 192.168.10.1) to use SSH through a serial (RS232) link, there is an excellent USB lead which allows to connect to the GPIO header pins directly and use the Raspberry Pis built in serial pins. To get X-11 working, there is only three things you need to do extra over normal SSH access.

4) Ensure that in Raspberry Pi whether X-11 forwarding is enabled.

5) Use nano with the following command:

```
$ sudo nano /etc/ssh/sshd config
```

In the line /etc/ssh/sshd config file which controls X11 forwarding, ensure that it says: X11Forwarding yes. Save if required then reboot

6) Type the command: $ *sudo reboot*

7) Run a X-Windows server after installing in the computer download and run Xming.

8) Ensure your SSH program (such as Putty) has X-11 enabled. In the Putty configuration, find connection, SSH, X11 and tick the check-box for X11 forwarding. If the X display location is leaved blank, it will assume the Server as default server[10].

9) Then a window will appear which asks for login id and password. Enter the Raspberry Pi's login id and password.

10) Type $ *startlxde*

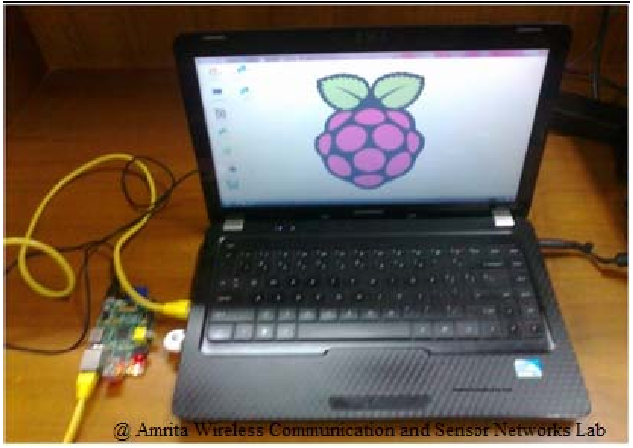

Fig. 3. Raspberry Pi Connection using LAN Cable

V. EXPERIMENTAL SETUP

In this paper, the transmitting data is a FM signal received by means of RTL-SDR connected on a Raspberry Pi. It transmits the received signal via Local area network, and displays on a remote PC, where SDR sharp is installed. First of all, doing this experiment it takes attention on choosing RTL sticks. The RTL2832U / E4000 sticks should found work fine. But RTL2832U / R820T sticks may not be fully supported in the RTL drivers used here. Experimental set up of this objective would take the following steps [10]:

1)Install the latest Debian release on your Pi and update it.

2) Before installing the RTL drivers, install the following dependencies if they are not already installed by typing the following commands in terminal window at the prompt.

```
$ sudo apt-get install git
$ sudo apt-get install cmake
$ sudo apt-get install libusb-1.0-0.dev
$ sudo apt-get install build-essential
```

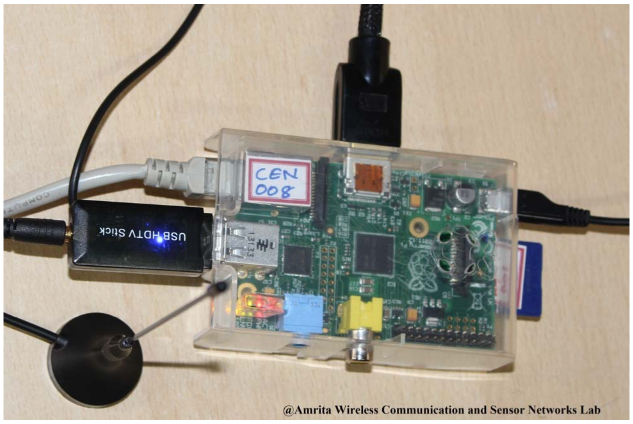

Fig. 4. RTL-SDR connected with Raspberry Pi

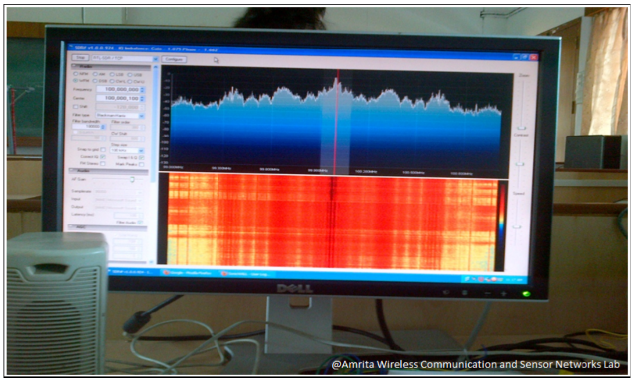

Fig. 5. FM Signal Reception using SDR Sharp

3) Now it is ready to install the RTL drivers using the following commands:

```
$ git clone git://git.osmocom.org/rtl-sdr.git
$ cd rtl-sdr/
$ mkdir build
$ cd build
$ cmake ../
$ make
$ sudo make install
$ sudo ldconfig
```

4) Before this will work it need to locate the RTL directory using the file manager where the drivers were downloaded and copy the rules file into the etc/udev/rules.d directory.

5) Plug in the RTL-2832U stick and issue the *rtl test –t* command to make sure the Raspberry Pi sees the stick.

6) Be sure that port 1234 is open on the router.

7) To start the rtl server type *rtl tcp -a plus the IP address* of Raspberry Pi .(For example *rtl tcp -a 10.0.1.50*)

8) On PC download the latest Dev version and configure it according to these instructions. Go to the interface section and select RTL-TCP and enter the IP address of the Raspberry Pi. Start SDR Sharp processing and it should be getting the data from the Pi and the RTL2832U.

9) Finally remove unnecessary peripherals like keyboard, monitor, and mouse[11].

Figure 4 shows the RTL-SDR dongle connected with Raspberry Pi and FM signal reception on SDR Sharp platform is shown in figure 5.

VI. WHY RASPBERRY PI, RTL-SDR AND REMOTE PC ?

There are several reasons for why we want to do this. Here are a few [11]:

1) One of the most important reason is to reduce the amount of antenna cable used, so that the signal loss will be reduced. The Raspberry Pi and RTL-TCP combination will allow to mount the RTL2832U nearer to the antenna connection point. (Example: Let the antenna is mounted in the upstairs, but monitoring station is downstairs. Rather than using 150 feet of cable for connection, mount the RTL2832U and the Raspberry Pi close to the antenna and use Wi-Fi to send the SDR data downstairs).

2) Place the Raspberry Pi/RTL2832U SDR server in one location and use laptop running SDR Sharp to monitor the RTL-2832U SDR radio anywhere in the house.

3) Make the Raspberry Pi / RTL2832U server accessible from outside of the home network and listen to SDR radio while travel.

4) Set up a remote monitoring location in another part of the country [6].

5) Mount the whole setup in a weather proof enclosure powered by solar cells and put it at the top of the antenna tower.

6) Tie all the setup to a helium balloon and having a 500ft antenna, make it fly.

The amateur radio community has worked fast on developing better software to operate the DVB-T radios and manipulate the resulting data. Reception of typical voice or digital modes plus mysterious things including encrypted voice can be done by using software such as GNU Radio or HDSDR. There are GNU Radio modules for P25, Mode-S transponders, ADS-B, GNSS, and INMARSAT downlinks. Performance wise, the devices are pretty good. The specific frequency band or tuning range available goes from about 60 MHz to above 1700 MHz using the best combination: an RTL2832 with the E4000 tuner. The other tuners have narrower ranges and less compatibility with currently software. Dynamic range is limited mostly by an 8 bit A/D converter and a noisy oscillator. Most users should find performance good enough for local stations and perhaps a bit of farther / weaker stations.

VII. RESULTS

Real-time communication system using Raspberry Pi and RTL-SDR was accomplished. The FM signal at 98.3 MHz received by RTL-SDR was transferred by means of TCP server across the LAN connection to a remote computer. The spectrum of the FM signal was realized using SDR Sharp software installed in remote PC.

VIII. CONCLUSION AND FUTURE WORKS

RTL-SDR along with Raspberry Pi can be used for reception of all signals within the bandwidth of spectrum of RTL-SDR. A Satellite Receiver Station can be constructed with RTL-SDR, Raspberry Pi and few other cheap parts. So that the satellite signals can be received and decoded in an inexpensive way. Also it can be used for cheap Automatic Identification System (AIS) ship tracking, and to hack the live DVD broadcast. In AIS ship tracking, a radio scanner or the cheap RTL-SDR can be used to receive these signals, and with the help of decoding software, ship positions can be plotted on a map. Other than that by using the combination of Raspberry pi and RTL SDR, the airplane data and weather data can be received. The ADS-B, an acronym for Automatic Dependent Surveillance-Broadcast technology allows tracking aircrafts using high speed radio transmissions.

IX. ACKNOWLEDGMENT

The authors would like to thank Dr.K.P Soman, H.O.D, Centre for Excellence in Computational Engineering and Networking (CEN) for his motivation. The authors would also like to acknowledge Mr.C.K Vinod (lab instructor, Wireless Communication and Sensor Networks Lab), all staffs and research scholars for their support.

REFERENCES

[1] Matt Richardson and Shawn Wallace, *Getting Started with Raspberry Pi*, United States of America:O'Reilly Media, 2012

[2] "AB9IL.net", http://www.ab9il.net/software-defined-radio/rtl2832-sdr.html, Accessed on May 2013

[3] RTL-SDR Installation Instructions: inst.eecs.berkeley.edu/ ee123/fa12/rtlsdr.html, Accessed on May 2013

[4] Sruthi M B, Gandhiraj R, Soman K P, "Realization of Wireless Communication System in SDR using GNU Radio", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) , vol. 3 Issue 6, June 2013.

[5] Gandhiraj.R, Soman.K.P., "Modern Analog and Digital Communication Systems development using GNU Radio with USRP", International Journal on Telecommunication Systems, Springer, Sep. 2013

[6] Shravan Sriram, Gunturi Srivasta, Gandhiraj R and Soman K P. , "Plugins for GNU Radio Companion", International Journal of Computer Applications, vol. 52, issue 16, pp. 11-16 Aug. 2012.

[7] Sruthi M B,Abirami M, Akhil Manikkoth, Gandhiraj R, Soman K P, "Low cost digital transceiver design for Software Defined Radio using RTL-SDR", *IEEE Int. Multi Conf. on Automation, Computing, Control, Communication and Compressed Sensing,* pp. 852-855, Mar. 2013.

[8] "rtlsdr.wiki", http://www.rtlsdr.org/, Accessed on May 2013.

[9] "Meltwater's Raspberry Pi hardware", http://pihw.wordpress.com/guides/guide-to-remote-connections/ Accessed on June 2013.

[10] "Ham Radio Science, Raspberry Pi as Remote Server for RTL2832u SDR", http://www.hamradioscience.com/raspberry-pi-as-remote-server-for-rtl2832u-sdr/ , Accessed on June 2013.