Hardware/Software Implementation of Software Defined Radio (SDR) Application in Engineering

Vinayak A. Karaval
Student Department of Electronics and telecommunication Engineering
Sinhgad Institute of Technology and Science
Pune, India

Gajanan A. Parab
Student Department of Electronics and telecommunication Engineering
Sinhgad Institute of Technology and Science
Pune, India

Nikeeta G. Nikam
Student Department of Electronics and telecommunication Engineering
Sinhgad Institute of Technology and Science
Pune, India

Vishal U. Rathod
Student Department of Electronics and telecommunication Engineering
Sinhgad Institute of Technology and Science
Pune, India

***Abstract*—Current technology and policy have created an apparent spectrum scarcity, a situation in which it seems there is not enough available RF spectrum to deploy the next generation of wireless services. It has been shown that this appearance is incorrect because the majority of licensed spectrum is unutilized. In order to address this problem a new technology, the spectrum sensing software radio. This idea has sparked a range of new technologies and algorithms for supporting dynamic spectrum access.**

The term software defined radio (SDR) first appeared in Joseph Mitola's 1991 paper Software Radios: Survey, Critical Evaluation and Future Directions". Mitola introduces the concept of applying digital signal processing (DSP) on general purpose hardware and using digital to analog converters (DAC) to build digital communications systems. The API developed for specific interfaces for general purpose processors (GPPs, defined as any processors supporting CORBA), digital signal processors (DSPs), and FPGAs, down to timing diagrams for communicating with FPGA buffers. It is use to define the manner in which hardware and software components interact. SDR technology should be seen as an enabler rather than a system technology, if properly applied, it will facilitate reconfigurability and run-time reconfiguration, and eventually cognitive radio.

***Keywords*— GNU radio, USRP, GSM SIM 900A, GPP, DAC, CORBA, API, FPGA, DSP**

I. Introduction

In this we focus on user friendly Software-Defined Radio developed work for prototype, research and education in wireless communications and networks. Using the USRP as the RF front end, this interface will use GNU radio for software radio development and signal processing libraries of digital baseband component of the communication transceiver design. This combination of software and hardware will enable the rapid, implementation, design, and verification of the digital communications systems in simulation, while allowing user to easily test the system with near real time over-the-air transmission. Starting with installation guides and scripts for Linux, GNU Radio and GRC, first receiver implementations System design.

II. System Design

Fig1. Proposed block diagram of system

The block diagram of the Software Defined Radio.

It is divided into five parts:

- Smart antenna
- RF Hardware
- ADC and DAC
- Channelization
- Baseband Processing.

1) Software-GNU

2) Hardware

a) USRP

b) GSM SIM900A

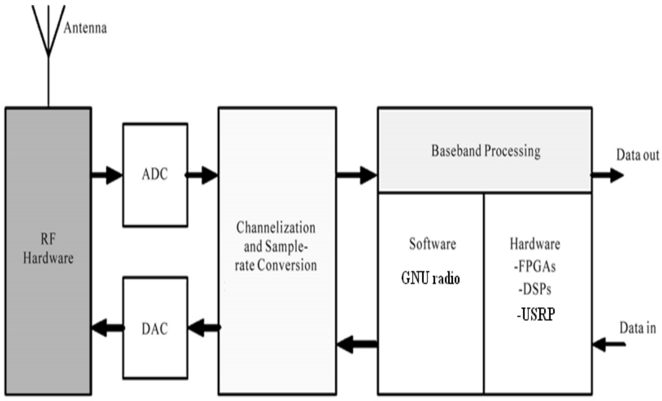

Fig. 1. Block Diagram of Software Defined Radio.

As in the figure 1 the block diagram of the system consists of RF hardware along with smart antenna which is used to collect desire band of frequency and amplify those frequencies and send further blocks that is to the ADC at receiver side and DAC at the transmitter side. After that channelization part is there which is digital filter and sample rate converter. The last part is baseband processing. It comprises software and hardware part. The software used here is GNU Radio and the USRP and GNU SIM900A the hardware part.

III. Module Design

The implementation of ideal software radio would require either the digitization at the antenna, allowing complete flexibility in the digital domain/ the design of a completely flexible radio frequency (RF) front-end for handling wide range of modulation formats and carrier frequencies. The ideal software Defined radio, however, is not yet fully utilized in commercial systems due to technology limitations and cost considerations.

A. Smart Antenna

The receiver begins with smart antenna that provides a gain versus direction characteristic to minimize noise, multipath, and interference. The smart antenna provides same benefits for the transmitter side also in smart antenna system. The array antennas are used which consist of distributed antenna elements whose outputs are combined together to increase the performance of overall system. A smart antenna is an antenna array system helped by some "smart" algorithm designed to adapt to different signal environments. Smart antennas and software defined radio complement each other well. Software radios provide the flexibility needed for effective smart antennas.

B. RF Hardware

In this part the desired RF signals are filtered and then amplified to adequate level. The amplified RF signals are transferred to digitization system.

C. ADC and DAC

Most practical software radios digitize the signal as early as possible in the receiver side while keeping the signal in digital domain and converting to the analog domain as late as possible for the transmitter using digital to analog converter (DAC). Often received signal is digitized in the intermediate frequency (IF) band. Conventional radio architectures employ super heterodyne receiver, in which the RF signal is picked up by the antenna along with other unwanted/spurious signals, filtered, amplified with low noise amplifier (LNA), and mixed with local oscillator (LO) to an IF. Depending upon the application, the number of stages of this operation may vary. Finally, the IF is then mixed exactly to baseband frequency. Digitizing the signal with an analog to digital converter (ADC) in the IF range eliminates the last stage in conventional model in which problems like carrier offset and imaging are encountered. When sampled the digital IF signals give spectral replicas that can be placed accurately near the baseband frequency, allowing digitization to be carried out simultaneously.

D. Channelization and Sample Rate Conversion

Channelization and sample rate conversion are essential to interface output of the ADC to the processing hardware to implement the receiver. Likewise, digital filtering (channelization) and sample rate conversion are often necessary to interface the digital hardware that creates the modulated waveforms to digital to analog converter at the transmitter side.

E. Baseband Processing

Processing is performed in software using field programmable gate arrays (FPGAs) or DSPs. The software used to modulate and demodulate the signal is GNU radio software. This forms a typical model of software defined radio. The hardware used is USRP B210 at the transmitter side and GSM SIM900A at the receiver side. The software radio provides flexible radio architecture that allows changing the radio personality, possibly in real-time and in process somewhat guarantees a desired QoS. This flexibility in hardware architecture combined with software architecture, through the implementation of techniques such as object-oriented programming to provide software defined radio with ability to seamlessly integrate itself into multiple networks with wildly different air interfaces. In addition, software radio architecture gives system new capabilities that are easily implemented with the software.

IV. Software Requirements

A. Gnu Radio

At the time of selection, GNU Radio version 3.6 was the latest stable release of GNU Radio available and supports Ubuntu 10.04. GNU Radio mainly consist of signal processing block library, system flow chart and signal visualizer. Applications are written in Python and C++ programming languages C++ is used to create signal processing blocks, while python is used to create flow graphs, script and combine signal processing block. Although initially created to run on Linux OS for which there is currently more installation support, installation package have now been developed for Windows and Mac OS.

V. Hardware Requirements

A. Transmitter Section

1) Universal Software Radio Peripheral (USRP): The hardware used here is USRP B210 board and GSM SIM900A.At transmitter side USRP B210 is used and at the receiver side GSM SIM900A is used. The features of both USRP B 210 and GSM SIM900 are as follows.

a) USRP B210: The USRP motherboard consists of a FPGA with components to provide the ADC, DAC, etc. functionality whereas RF daughterboard provides RF-Front-End functionality.

Fig. 2. USRP B210 board.

B. Receiver Section

a) GSM SIM900A: GSM(Global System For Mobile) works on frequencies 900MHz and 1800MHz.It is very compact in size and easy to use as plug in GSM modem. The modem is designed with 3V3 and 5VOlt DC TTL interfacing circuitary.The baud rate can be configurable from 9600-115200 bps through AT (Attention) Commands. This GSM Modem has internal TCP/IP stack to enable user to connect with internet through GPRS features.

Conclusion

To perform the FM receiver with the help of software defined radio GNU radio software is used whish simulate the flow graph and automatically generates python code which is similar to C++ programming language. The requirements set by need for multi-band, multi-mode operation and reconfigurability have implications on implementations of various parts of a software defined radio set, ranging from the selection of processing hardware to the RF front end. There are few critical points: considering the physical implementation, the analog-to-digital conversion and the power consumption of many of the components are among the most important issues, which limit the choice of physical layer architecture and eventually the achievable performance.

References

[1] Software Defined Radio: Architectures, Systems and Functions (Markus Dillinger, Kambiz Madani, Nancy Alonistioti) Page xxxiii (Wiley & Sons, 2003, ISBN 0-470-85164-3)

[2] Mitola III, J. (1992). Software radios-survey, critical evaluation and future directions. National Telesystems Conference. pp. 13/15 to 13/23. doi:10.1109/NTC.1992.267870. ISBN 0-7803-0554-X.

[3] http://www.ettus.com/downloads/ettus_ds_usrp2_v5.pdf[dead link]

[4] EU Seeks To End Mandatory GSM for 900Mhz - Source

[5] How To Build an FM Receiver with the USRP https://www.youtube.com/watch?v=KWeY2yqwVA0.