Software Defined Radio:

Investigating Hardware and Software Integration

by Joao Paulo Rodrigues

Advisor: Dr. Yu-Dong Yao

April 2013

Abstract

This report analyzes aspects of the Cognitive Radio for the purpose of research by educational institutions. The main focus is to understand the relationship between hardware and software components currently being used in this field of research. The software chosen was the open source GNU Radio, which is the primary software used in several researches in Stevens Institute of Technology. Several hardware components were considered, in regards to their architecture, bandwidth, and compatibility with the GNU Radio. Some of the hardware included the USRP by Ettus Research, the High Performance Software Defined Radio, the Quicksilver, the FlexRadio Systems family, and the Persaund. Theses from different universities and organizations, as well as specifications for each of the hardware were investigated to help in the comparison between the possible front end components. As a result, the USRP family proved to be the most suitable to the GNU Radio amongst the investigated hardware. The USRP seemed to be the choice of many researchers due to its high degree of customization, low cost and flexibility.

Table of Contents

Introduction:

As technology advances, so does our ability of solving problems more efficiently and accurately. The fields of computer and electrical engineering are proofs of such marvelous advances since today new technology is developed at an exponential rate. Educational institutions play an important role on initiating the creation of such technologies because of their ongoing research. The software defined radio is an example of these developments. Research is being performed to help revolutionize the way communication devices work. It is important however, to understand how it is possible to create advanced systems by the integration of hardware and software components. To optimize the results of such studies, the best hardware should be selected. This report will analyze and compare different hardware components and their compatibility to the GNU radio.

Software-Defined Radio:

Software-Defined Radio (SDR) is usually defined as a communication system that can be configured by means of software implementation. Traditional radios rely too much in hardware components such as filters, amplifiers, and modulators. Because each type of radio involves a different combination of hardware components, it is very difficult to have one radio that has several different functions. The SDR, theoretically, would allow all of these functions to be implemented by means of software, and therefore would widely transmit a great variety of radio protocols. SDR has great applications for the military, mobile devices communications and real time transmissions [6]. In order to process all the information, SDR usually goes through a particular frequency band (RF) digitization, in which an analog-to-digital (A/D) converter digitizes radio waves captured by an antenna [8].

Cognitive Radio & Applications:

SDR, not only is helpful with communications, but it may also have "cognitive" capabilities. Such radios, are called *Cognitive Radio*, and have much more applications since they can sense the spectrum, avoid interferences and transmit information without interfering with other signals. This would allow usage of frequency bands in a much more efficient manner. As Anthony Hsu and his team explained in their report for the research at Stevens Institute of Technology frequency hopping, a method of changing transmission frequencies in a certain period of time, would provide several applications for a radio system. Three examples by the team were: making the system "less susceptible to interference", "make signals difficult to intercept" and "allows the radio frequency spectrum to be utilized more frequently"[6].

GNU (GNU Not UNIX) Radio:

GNU is an open source software development kit that allows for implementation of SDR by using programmable blocks that control signal processing abilities. The blocks, which can be programmed with C++ and Python, perform the function of the hardware components previously mentioned, such as equalizers and demodulators. The GNU runs in Linux and Windows platforms. Recently, GNU has started to support other types of software as well such as LabView and Matlab. Technically, no hardware is needed to run GNU radio. However, the usage of hardware allows transmission and reception of signals [2]. Further sections of this report try to analyze specifications from each of the software and compare them for use in research institutions such as Stevens Institute of Technology.

Ettus Family Hardware:

Although the GNU radio has capabilities to work without any hardware, the hardware usage may enhance the results of a study in signal communications. Current research at Stevens Institute of Technology and many other institutions seem to use the Universal Serial Radio Peripheral as their hardware component, because of its capabilities and recommendations by the GNU project website. The hardware has the responsibility of simulating various frequency bands, RF, which is made possible by connecting different daughterboards to the USRP devices. The prices of the USRP family products range in the $1,000.00s. Different daughterboards and the range of frequencies they are designed for are depicted in the graph below, as documented by Matt Ettus himself. Other information about the daughterboards such as frequency ranges, features and bandwidth can be found in the document as well.

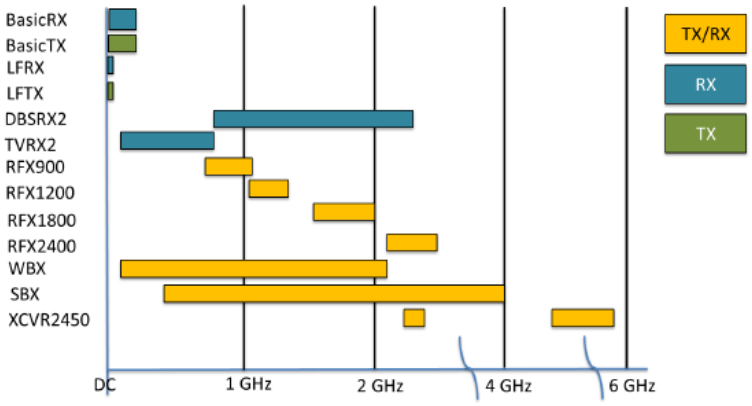

Figure 1 Daughterboards for USRP

The USRP devices can be subdivided into two main categories: USB 2.0 connections (USRP1) or Gigabit Ethernet (USRP2). The greatest difference between the two is the speed of connectivity and transfer of information between the computer and the USRP devices [5][6]. Some researchers believe that the performance of the GNU radio would increase by using

hardware accelerators such as Xilinx Field Programmable Gate Arrays (FPGAs) in addition to USRP devices [7].

High Performance Software Defined Radio (HPSDR):

A new type of research that seems to be starting to become popular is the HPSDR. The project involves breaking down the overall design of the radio into several modules, which are responsible for optimizing specific functions. This method allows users to choose what modules are more appropriate for their interests and furthermore, enables the user to design new components. Specifications for each of the modules can be found on the HPSDR website. The with the GNU radio [3].

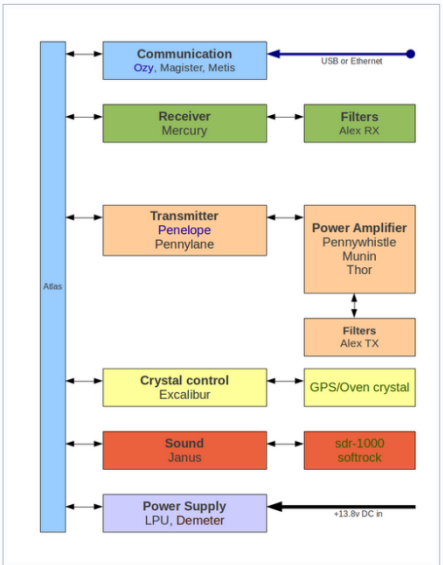

Figure 2 HPSDR Hardware Flowchart

Quicksilver:

Also known as QS1R, the quicksilver is yet another of the hardware transmitter available in the market. The main downside is that the GNU radio does not directly support the usage of the hardware. However, as pointed out by Adeel Anwar in the discussion boards for the gnu radio users may use a UDP source block to stream the samples from QS1R to the GNU [1]. The QS1R costs $899.99 and offers other software such as Winrad. It has a range of 10kHz up to 62.5Mhz with up to 2MHz bandwidth and is supported by Windows, OSX, and Linux software [4].

FlexRadio Sytems:

FlexRadio Sytems hardware seems to be a very competitive choice of hardware. The hardware is very commercial based, and its prices range from $649.00 to $7499.00 depending on how advanced the system is. One down side is that the hardware doesn't really support the GNU radio anymore. Instead, one would have to use the PowerSDR, which is provided for free in FlexRadio Systems' website. The PowerSDR uses c#-based language and is advertised on their website to require no programming experience at all. The most advanced transceiver however, only has frequency coverage from about 30kHz to 77MHz or 135Mhz to 165Mhz [10].

Other Hardware:

Other hardware that were found to be used as software radios were the PERSEUS SDR and the SoftRock. The Perseus is distributed by microtelecom and not much information is available online. On microtelecom's website PERSEUS is advertised as a receiver for short waves and works from 10Khz to 40MHz [9]. The SoftRock was also found often upon research

of SDRs, but it seems they are homemade boards and do not have much support and reliability as compared to the others.

Analysis and Conclusion:

To compare the different hardware components, a Quality Functional Deployment table was created. The characteristics analyzed were: cost, warranty, frequency range, compatibility with GNU, compatibility with other software, documentation and availability. The weighs of each characteristic are 1- being the minimal importance, 3- strong importance and 7-critical importance. When analyzing the HPSDR, a cost, frequency and warranty was not applicable since the HPSDR may be composed of different hardware, due to user flexibility to choose what they want use. Below is the table with the results based on the information analyzed previously.

	Weight	Ettus Family	HPSDR	QS1R	FlexRadio	PERSEUS	SoftRock
Cost	3	3	N/A	7	1	7	7
Warranty	1	3	N/A	1	7	1	1
Freq. Range	7	7	N/A	3	3	3	1
Compability w/ GNU	7	7	1	3	1	1	3
Comp. w/ other software	3	3	7	3	7	3	3
Documentation	3	3	1	1	7	1	1
Availability	3	7	3	3	7	1	1
	Total	137	40	85	101	65	65
	Rank	1st	5th	3rd	2nd	4th	4th

Based on the results, the Ettus Family hardware is the most suitable for research with the usage of the GNU radio software. Ettus Family has the best variety of transceivers as the user may choose the most appropriate type of transceiver based on their needs. The great variety of frequency ranges allows users to test details such as pseudorandom hopping and other

components of a cognitive radio. The FlexRadio Systems' hardware also offered good quality components but with a limited frequency range, which might not be the most appropriate for research about cognitive radios.

More comparisons could be made within the Ettus Family's hardware components, however, without deep knowledge on the most recent research performed at Stevens it is difficult to decide what requirements the hardware needs to meet. There are many types of Ettus products and each of them has its own advantage depending on what they will be used for. The website provides documentation and comparisons of all of its products, so one may choose what best meets the requirement quite easily. As research on GNU radio and SDR become more popular, one may expect an increase in hardware front end components, but for now, Ettus Research proves to be doing a good job with their products.

Works Cited

[1]Anwar, Adeel. QS1R SDR support? (http://lists.gnu.org/archive/html/discuss-gnuradio/2012-10/msg00033.html)

[2]Blossom, Eric, et al. GNU radio. (http://gnuradio.org)

[3]Covington, Phil, Phil Hafrman, Bill Tracey, et al. Open Source High Performance Software Defined Radio. (http://openhpsdr.org/index.php)

[4]Covington, Phil, et al. Software Radio Laboratory LLC wiki. (http://qs1r.wikispaces.com/)

[5]Ettus, Matt, et al. Ettus Research LLC. (http://www.ettus.com/)

[6]Hsu, Anthony, Thomas Bell, Kevin Gajewski (2009). Pseudorandom Multiband Frequency Hopping for Interference Avoidance Using GNU Radio and USRP. *NSF Research Experience for Undergraduates Paper. Stevens Institute of Technology*. (https://sites.google.com/site/ahsustevens/)

[7]Irick, Charles Robert (2010). Enhancing GNU Radio for Hardware Accelerated Radio Design. *MS Thesis. Virginia Polytechnic Institute and State University.* (http://scholar.lib.vt.edu/theses/available/etd-06072010-203946/unrestricted/Irick_CR_T_2010.pdf)

[8]Manicka, Naveen (2007). GNU Radio Testbed. *MS thesis. University of Delaware*. (http://www.eecis.udel.edu/~manicka/Research/NaveenManicka_Thesis.pdf)

[9]Microtelectom s.r.l. (http://microtelecom.it/perseus/)

[10]Youngblood, Gerald. FlexRadio Systems – Software Defined Radios. (http://www.flexradio.com/)

Images List

Figure 1. Matt Ettus. Application Note Selecting an RF Daughetboard. *Ettus Research LLC*. (http://www.ettus.com/content/files/kb/Selecting_an_RF_Daughterboard.pdf)

Figure 2. Phil Coving, et al. Hardware Block Diagram. *Open HPSDR* (http://openhpsdr.org/index.php).