A High Frequency Transceiver for Amateur Radio Using Software Defined Radio

Jack Sun, Youssef Chedid, Kaveh Hajimohammadreza, Efrain Mendoza, Silvestre Sanchez, Kinal Vachhani, James Flynn, Sharlene Katz
California State University, Northridge

Abstract

This paper describes a Software Defined Radio (SDR) based High Frequency (HF) transceiver for amateur radio that was designed to fulfill the senior design project requirement in Electrical and Computer Engineering at California State University, Northridge. The uniqueness of the SDR approach is that most of the hardware components found in a conventional transceiver are replaced by software. This results in near-ideal filters and a reduction in overall physical size. In addition, any improvements or changes to the SDR transceiver would only require a software upgrade as opposed to a labor-intensive redesign of the hardware version of the transceiver itself.

The paper will describe the specifications of the HF transceiver and the various modes of operation. This will be followed by a detailed explanation of how all these different components come together in one software package.

Although this started out as a senior project designed to teach the technical aspects of engineering, the scope quickly expanded to encompass many facets of a real engineering project: time management, written and oral communications, and working cohesively within a multidisciplinary team. This paper will elucidate the myriad of challenges the team encountered and overcame.

I. Introduction

This paper details the senior design team project by students at California State University, Northridge (CSUN). The purpose of the project was to build a software defined radio (SDR) based high frequency (HF) amateur radio transceiver. The twelve month project included both the technical and project management aspects commonly found in a real-life design project. Electrical and computer engineering undergraduates at CSUN are required to complete a group project as part of their senior design course. This project is one of the activities that are used to demonstrate that our graduates have achieved the following outcomes:

- an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- an ability to function on multidisciplinary teams
- an ability to communicate effectively

While all of the projects assigned as part of senior design attempt to demonstrate these outcomes, our preliminary assessment shows evidence that the unique aspects of this project provided deeper challenges and opportunities to the students. The challenges of the projects motivated students to higher levels of performance in the required outcomes while working with SDR gave them the opportunity to realize their designs within the time frame of the course.

Section II of the paper will provide background information on SDR. Section III will describe the project itself and the various components implemented. Section IV of the paper will provide the results of the project and the different aspects learned through the project. Section V discusses our assessment of the educational outcomes listed above. Finally, Section VI will present the summary and conclusions.

II. Background

The foundation of this project is software defined radio. The main idea behind SDR is to take signal processing schemes traditionally done in hardware and achieve the same results with software.

Although the eventual goal of SDR is to move all software signal processing as close to the antenna as possible, there was a front-end device used immediately after the antenna in this project: the universal software radio peripheral (USRP). For receiving radio signals, the USRP provides analog to digital convertors (ADCs), the required frequency down conversion and the decimation of the digital samples needed prior to sending the signal to the host computer1. The process is reversed on transmission. The USRP digitally up converts the signal to the desired frequency before digital to analog convertors (DACs) convert the signal back to analog and send the signal back out to the antenna1.

Figure 1 details a traditional hardware receiver. Figure 2 depicts an SDR system. In this case, the receiver front-end is merely a filter and gain stage while the software in the host computer replaces the balance of the receiver after analog to digital conversion. As outlined in Figure 3, the USRP provides the analog to digital conversion and frequency down conversion.

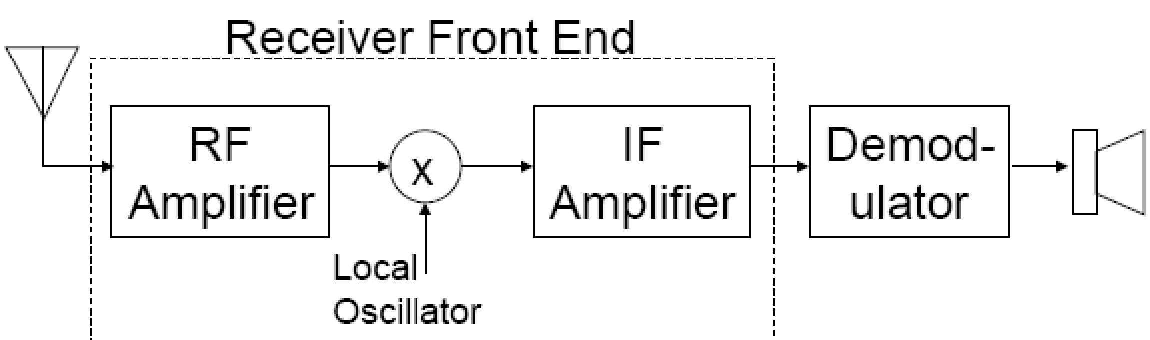

Figure 1. Traditional Hardware Receiver

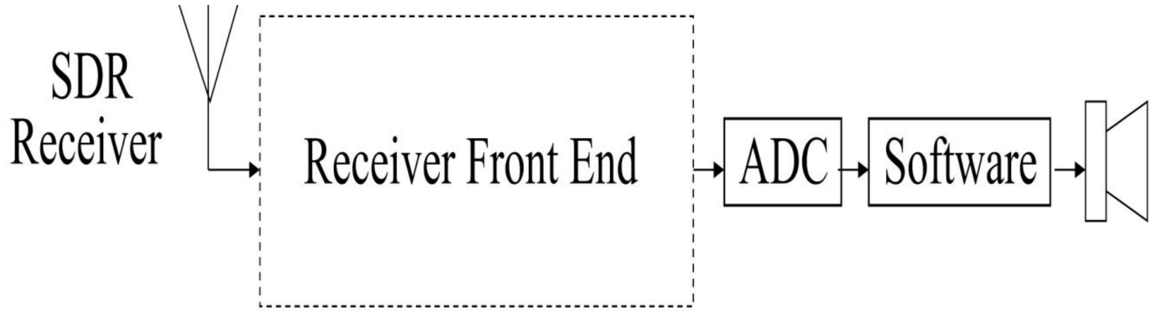

Figure 2. SDR Receiver

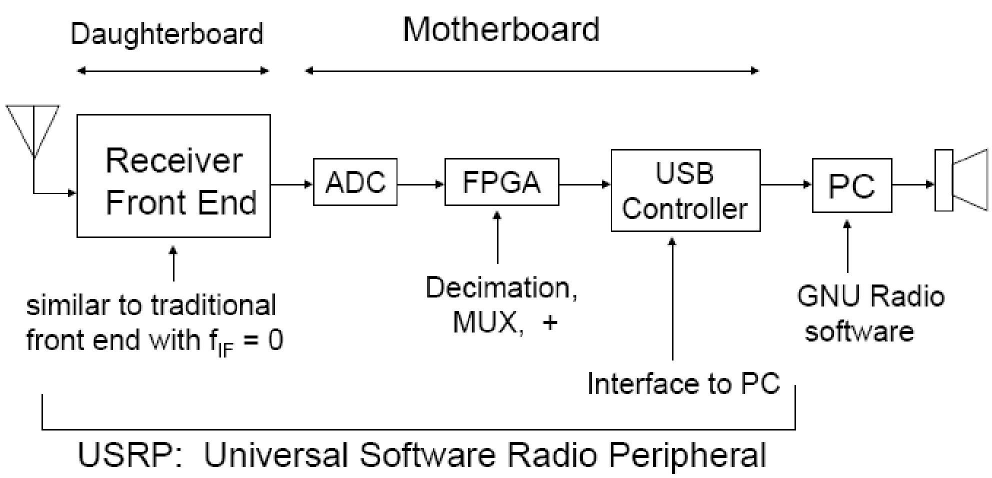

Figure 3. SDR Receiver with USRP

For the SDR HF transceiver, the software is the brain of this entire transceiver operation. The bulk of the software for this project was done with GNU Radio, a "free software development toolkit" which eventually led to the creation of the USRP2. GNU Radio has an extensive library of code blocks which can modulate, demodulate, filter and process a signal in the desired manner.

Besides utilizing GNU Radio, much of the coding – especially for the Graphical User Interface (GUI) – was done entirely from scratch using the Python programming language. Since GNU Radio is written in Python, it is only reasonable to write the additional code in Python to ensure maximum compatibility.

There are numerous benefits to implementing a transceiver that uses SDR. The first benefit is that filters that would be impractical or impossible to implement in a hardware system can now be easily implemented with SDR at the expense of some processing power. However, the resounding theme and the main benefit of SDR is flexibility. With a single receiver front-end, software can be written to transmit and receive AM, FM, CW (Morse code), or any other mode. This realization can be achieved without ever having to alter the hardware in any way. As soon as a new block of code is written, it can be implemented and tested immediately. The flexibility to fine-tune and the ability to make on-the-fly alterations are astounding and practically unheard of in the hardware industry.

One may be able to imagine the difficulty, not to mention financial restrictions, to make changes or updates to an already existing hardware system. With software defined radio, the system can be changed to adapt to current needs quickly and inexpensively.

The impetus behind this project should be mentioned here. This project was funded and made possible by Edwards Air Force Base. EAFB and CSUN have been partnered since 2003 in an effort to increase and maintain the number of students that work or intern at EAFB as well as increase interests in complex projects such as this. Thanks to that partnership, the students have gained and will continue to gain invaluable knowledge and experience that will help them succeed in the workforce.

III. Project Description

Amateur radio transceivers are devices used by the over 1 million hobbyists around the world to transmit and receive signals. Many different transceiver models are available, ranging in price from several hundred dollars to well over $2000. The purpose of this project is to create a transceiver using SDR with comparable performance to a high-end hardware transceiver.

As an integral part of an improved senior design course, students taking part in this project were exposed to a near real life engineering world. The CSUN faculty acted as management/customers, setting time and resource limitations on the students. The students were required to give weekly oral presentations on their progress and problems encountered. The faculty offered guidance to possible solutions, but emphasized real constraints on solutions, including cost, lead time, reliability, manufacturability and team time investment.

The SDR transceiver was designed with the following requirements:

1. The SDR Transceiver operates in the high frequency (HF) amateur radio band from 1.8 to 29.7 MHz.
2. The transceiver has a GUI that will be understandable to amateur radio operators.
3. The GUI has the following information for the operator:
   - Received signal strength
   - Transmitted signal power
   - Received bandwidth
   - Mode
   - Volume
   - Transmit frequency
   - Receive frequency

- RF gain/attenuation
- Display of frequency in use plus surrounding frequencies

4. The transceiver transmits and receives in the following modes.
   - Amplitude Modulation (AM)
   - Frequency Modulation (FM)
   - Single Sideband (both USB and LSB)
   - Continuous Wave (CW, or Morse Code)
   - Digital Modulation
5. The transceiver conforms to current FCC regulations regarding:
   - Transmitted Bandwidth
   - Harmonic suppression
   - Spurious emissions
6. An instruction manual covering installation (software) and operation is provided.
7. The following transceivers were used as inspirations for design features and specification comparisons:
   - ICOM IC7700, IC7800
   - Yaesu FT-1000D, FTDX-9000
   - Kenwood TS-2000

IV. Project Outcomes and Results

Technical Accomplishments

The project consisted of specific requirements and modulation schemes that were researched, analyzed mathematically, and implemented into a working system that would be distributed freely under the GNU General Public License. The project software and related items are hosted on SourceForge, a website dedicated to source code repository and open source software projects.

Although five modulation schemes were mentioned in Section III, digital modulation was ultimately scrapped due to time constraints. The team decided it was best to have four complete modes rather than several incomplete modes at the end.

AM pioneered the architecture used throughout the rest of the project for implementing the various modulation modes. It was also the first mode implemented into the GUI with the required applications, such as power meter and S-meter. Because the design of the AM section was very thorough, this implementation process was utilized in all subsequent modes.

A figure of the main GUI window is shown in Figure 4 below. This is the first item the user sees when starting up the software. The upper portion allows the user to select a specific mode via a drop-down box. After choosing the desired mode, the user must click on the select button to start that particular mode. Below the mode selection is the parameter display. A bar gauge functions as the output power meter on transmit and the signal strength, or S-meter, on receive. The original intent was to design two discrete gauges, but after reviewing the hardware transceivers available, the team decided to implement one gauge with separate power and S-meter labels – a common feature in the hardware transceivers.

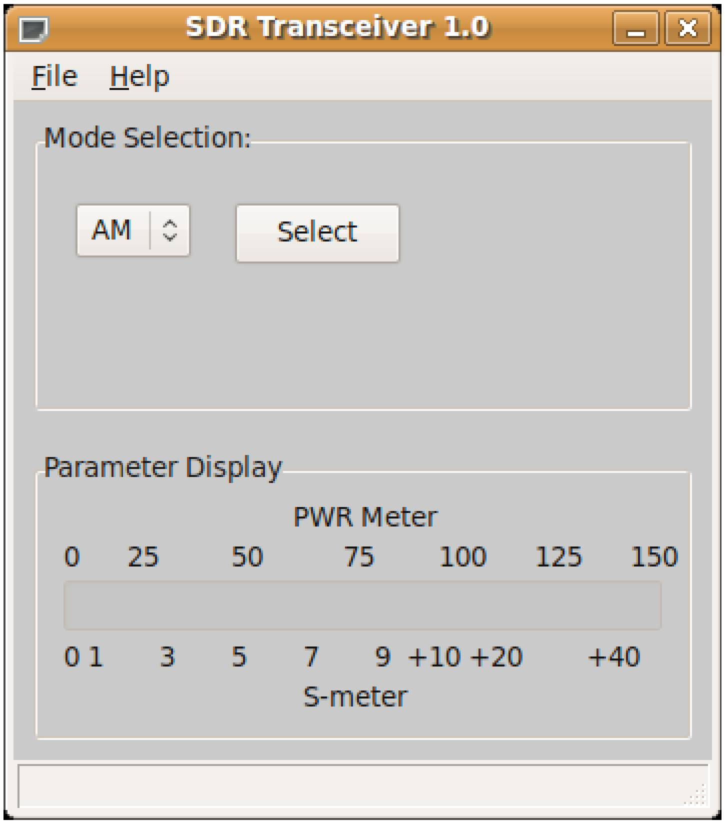

Figure 4. SDR Transceiver Main GUI Window

After the user confirms the selected mode, a new window will appear alongside the main GUI window. Many of the features shown in this AM example appear in other modes. However, some are tailored to have more or fewer features. As examples, FM contains the squelch control and CW only contains one receiver channel.

It is important to note here that only one mode may be active at any time. If a mode is already running, the selection of another mode is restricted because resources such as the audio driver are already being dedicated to the first mode. In order to select another mode, the user must first explicitly close the currently running mode.

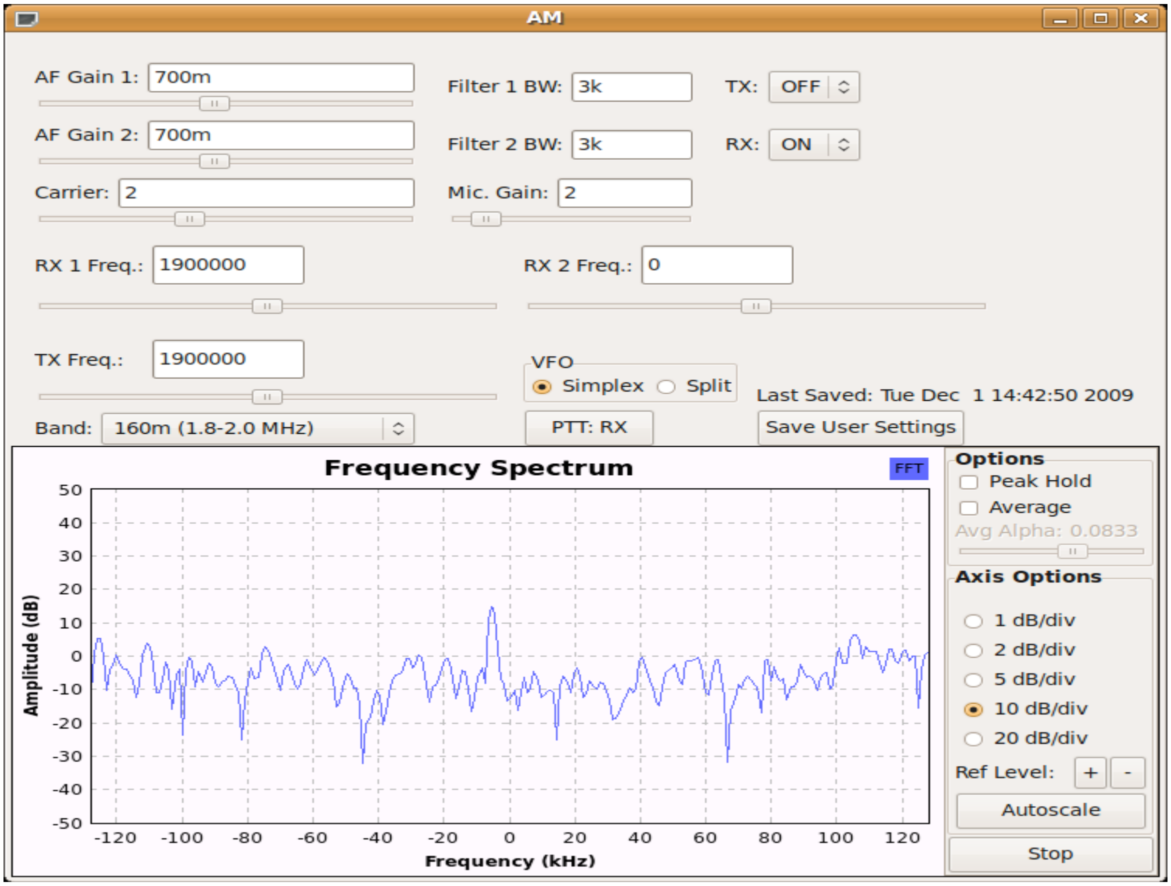

Figure 5. AM Mode Window

As shown in Figure 5 above, the user has control over many features and functions. In AM and SSB, there are two receiver channels; the user is able to listen on two separate frequencies (one channel in one speaker, the second channel on the other). Because of this, there are two audio frequency (AF) gain controls, which are the receiver volume controls. To the right of the AF Gains are the receive filter bandwidth controls. They control the filter cutoff frequency for the two receiver channels. The bandwidth is limited to a maximum of 4 kHz by the sampling rate used in the filter sections.

Below the receiver controls are the carrier and microphone gains. These two controls set the power output of the transmitter. The carrier control sets the carrier level while the microphone gain sets the modulation level.

In the Variable Frequency Oscillator (VFO) section, the two choices are simplex or split. In simplex mode, the receiver and transmitter frequencies are "locked" such that moving one will also move the other. In split mode, the receiver and transmitter frequencies may be moved independently of one another. When the second receiver – designated RX 2 Freq. – is set to zero, it represents an offset of zero such that the speakers will only play the main receiver frequency channel.

The PTT button stands for Push-To-Talk, and by default it is on receive. Clicking on the button will have the text change to "PTT: TX," signaling the user that the mode is now transmitting.

The two drop-down boxes on the top-right of the window allows more functions such as both receiver and transmitter off, or both on, should there be a desire to implement full-duplex operations for future teams. On the lower-left, a drop-down box displays all the bands allowed by FCC regulations in the HF range.

Because a major objective of this project is to emulate a program that would be used commercially by many people around the world, a save feature was implemented. The "Save User Settings" button is able to store every feature the user has control over into an external file. A timestamp of when the user last saved the settings is displayed on both the mode window as well as the external file for reference. This function was created for ease of use and provides both a friendlier and faster way to use the software.

The SDR HF Transceiver was completed and tested to specifications on time in November 2009. Using a receiver pre-amplifier and a 140 watt transmit power amplifier provided by faculty, the transceiver was used to carry out a number of contacts. Stations in Seattle, Washington, Fairbanks, Alaska and Mexico City were contacted and signal reports received. Subjective assessments rated the signal as clear and understandable.

Lessons Learned

Many technical lessons were learned through the SDR Transceiver project. The first is being able to go from designing, through implementation, to debugging and troubleshooting. The technical challenges include having to translate the theories learned in classroom settings regarding the various modes into software, which is not always a one-to-one translation. The implementation process taught the team how to link the software code to the hardware, such as the USRP and the antenna, and vice versa. This often resulted in errors and bugs that the team had to troubleshoot before patching the system to resolve the issues.

The non-technical lessons learned are equally, if not more, important than the technical lessons. The first is project management, which proved to be crucial during the twelve months for the project. Good project management includes balancing every member's time and schedule as well as laying out a timeline of what needs to be done in a timely manner. There are many components in the project, and managing every member's time became more and more crucial as time went on. Most of the members from the team are close to graduation, thus time management included other facets of the group's lives, such as school and work. Having good project management keeps everybody on pace on what and when something needs to be accomplished. Furthermore, this provides a list of goals that the team is ultimately moving towards. Being able to see what components still need to be completed as time goes on keeps everyone focused and looking ahead.

Time management was facilitated by the use of Gantt charts to keep a timeline of what needed to be completed within a certain amount of time. To further keep every task on schedule, a weekly meeting was established to brief the professors on what has transpired over the past week and what tasks would be worked on the in following week.

Working in a real-life setting meant dealing with rules and regulations. In the case of the SDR Transceiver project, the team had to constantly deal with FCC rules and regulations – examples being sideband suppression and bandwidths allowed.

Another important lesson learned was documentation. However, the lesson was not just documentation, but *good* documentation. This includes the software code, the schematics, and reasons for approaching an issue in certain ways. Good documentation keeps everyone else in the group, as well people outside the group, aware and knowledgeable of every aspect of the project.

Lastly, the concept of regression testing was monumental throughout this project. The concept of regression testing is to re-test the entire system every time a new component is inserted into the overall system. The reason for doing so is because there could be unintended results when a new part is introduced into the system – often deleterious to the overall progress of the project.

The SDR Transceiver project required students to be proficient in both electrical and computer engineering fields. The reason is that students must have a background in programming and communications to implement all the features of this project. Every member on the team is an Electrical Engineering major. At CSUN, electrical engineering majors only take one programming course, which made the design and implementation of the features and functions difficult. It is because of this, a new programming language – Python – had to be learned in a relatively short amount of time.

Comparing the SDR Transceiver to the hardware transceivers sold in the consumer market, the SDR Transceiver has several advantages. The first is financially, as the SDR Transceiver only requires the purchase of the USRP (both software and hardware require an antenna, so it is not counted in this comparison). At $700, the USRP is very cheap compared to the transceivers sold in the market.

Another distinct advantage of the SDR Transceiver is achieving results not realizable, or at the very least, not practical, in hardware transceivers. For instance, in the SDR Transceiver the filter is set to 380th order – resulting in near-ideal responses. Such a high order is simply not realizable or financially feasible in hardware transceivers.

Other advantages were made apparent as the SDR Transceiver was being tested. It met or exceeded all specifications and compared favorably with high-end amateur transceivers. For example, Part 97 of the FCC regulations stipulates that the unwanted sideband suppression should exceed 50 dB3. The Yaesu FT-1000D has 55 dB of unwanted sideband suppression4.

The SDR Transceiver achieves 70 dB of suppression. Another example is 3rd order intermodulation distortion (IMD), a standard measure of how much distortion in present in a sideband signal. The FT-1000D has 3rd order distortion products 36 dB down from the main signal, while the SDR Transceiver was measured at 35 dB down4.

The last main advantage is the flexibility discussed throughout the entirety of this paper. The user can change settings instantly without having to solder or change any hardware components in the process. Furthermore, troubleshooting and debugging is several times easier because any changes could be tested right away. Furthermore, the concept of the SDR Transceiver is completely built on the concept of modular design. After selecting the mode from the main GUI, the new mode window is started as a new process separate from the main GUI. The only tie between the two processes is the S-meter and power meter data from the mode window to the main GUI window. This makes the design of additional modes, such as digital modulation, an easy addition into the software because each mode has its own Python file. Simply code the new mode, put the file in the same directory as the GUI file, follow the code used to initiate a new process for other modes like AM or SSB, and the new digital modulation has been integrated into the software system. Furthermore, due to the modular design of making the modes entirely separate processes from the main GUI, crashing a specific mode will not crash the entire software program.

V. Assessment of Educational Outcomes

Senior design at CSUN is used to demonstrate that our graduates have achieved the following outcomes:

- an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- an ability to function on multidisciplinary teams
- an ability to communicate effectively

The transceiver project described in this paper was part of a Design Clinic sponsored by Edwards Air Force Base (EAFB). This gave students the opportunity to work on a real world problem with realistic constraints. The requirements presented to the students included the need to create a system that was comparable to existing commercial amateur radio transceivers in terms of performance and operation along with the need to operate within FCC regulations.

The team consisted of six electrical and computer engineering students. The aspects of the project covered several of the disciplines within electrical and computer engineering including communications, signal processing, electronics, antenna design, digital systems, and software. This provided an excellent experience for students to experience the challenges of functioning on a multidisciplinary team.

All senior design students are required to submit both oral and written reports on their projects throughout the course. The funding provided by the EAFB design clinic gave the faculty advisors on this project the opportunity to enrich the communications portion of this course. Students were required to give oral progress reports on a weekly basis. The reports were critiqued by three faculty. Two of the faculty advisors have extensive industry experience. This exposed students to many of the presentation styles currently used in industry. In addition to the standard report on their project, the student participants were also required to write a paper for submission to a conference.

The assessment of this project was informal. However, even without a formal assessment, evidence was collected to indicate that the projects and the format in which they were offered did improve the ability of our graduates to demonstrate the outcomes listed above. The following indications were noted and will be measured formally in the coming year:

- Faculty advisors observed a noted improvement in presentation skills of the students
- Representatives from industry attending student presentations indicated that students made better presentations than generally expected from new graduates and were impressed with the technical level of their work
- Industry representatives also commented on the excellent teamwork
- As compared to other senior design students, the presentations were notably better
- Student participants indicated their satisfaction with working on the project even though they realized it was more work than other senior projects
- Student enthusiasm for the project was evident from their dedication and hard work
- Faculty advisors noted that the use of SDR enabled the students to complete more complex and interesting projects in the communications area.

One negative aspect of this project was the amount of faculty support that it required was well above the level that is normally available to senior design students. In this case the support was possible due to the funding available from the EAFB Design Clinic and the faculty interest in SDR. The department is investigating ways to implement the model used in a more cost effective manner.

VI. Summary and Conclusions

This senior design team project by the students at the California State University, Northridge, focused on the concept of Software Defined Radio. Funded by Edwards Air Force Base, Software Defined Radio aims to lessen the number of hardware components, replacing them with software instead. Being able to upgrade or patch via software, rather than redesigning the hardware system from scratch, has enormous benefits in today's world of digital technology.

This project amalgamated aspects of a real-life design project not normally found in classroom settings. The interdisciplinary nature of the project meant the students have had exposure to both the analog and digital sides of electrical and computer engineering. The main lesson learned through this project is how to take everything learned in classrooms and apply them in a development setting with project and time management, from budgeting every member's time to ensuring the project is completed on schedule.

Bibliography

1. "USRP." GNU Radio: The GNU Software Radio. 22 Nov. 2009. <http://gnuradio.org/trac/wiki/UsrpFAQ/Intro>.
2. "Welcome to GNU Radio." GNU Radio: The GNU Software Radio. 22 Nov. 2009. <http://gnuradio.org/trac>.
3. "Part 97: Amateur Radio Service." ARRL: The National Association for Amateur Radio. 3 Dec. 2009. <http://www.arrl.org/FandES/field/regulations/news/part97/>.
4. "FT-1000 Operating Manual." Yaesu: Amateur Radio Division of Vertex Standard. 3 Dec. 2009. <http://www.yaesu.com/downloadFile.cfm?FileID=722&FileCatID=158&FileName=FT%2D1000%5FOpManual%40.pdf&FileContentType=application%2Fpdf>.