Software Defined Radio Based Laboratories in Undergraduate Computer Networking Courses

Dr. Deng Cao, Central State University

Dr. Deng Cao received his Ph.D in Computer Science from West Virginia University in 2013. He earned two master degrees in Statistics and Physics from West Virginia University, and his bachelor degree in Physics from Hunan Normal University in China. Dr. Cao joined Central State University in 2013 and currently serves as an assistant professor in the department of Mathematics and Computer Science. His research interests include advanced biometrics, computer vision, pattern recognition and machine learning.
.

Dr. Zhiqiang Wu, Wright State University

Dr. Zhiqiang Wu received his BS from Beijing University of Posts and Telecommunications in 1993, MS from Peking University in 1996, and PhD from Colorado State University in 2002, all in electrical engineering. He has worked at West Virginia University Institute of Technology as assistant professor from 2003 to 2005. He joined Wright State University in 2005 and currently serves as full professor. Dr. Wu is the author of national CDMA network management standard of China. He also co-authored one of the first books on multi-carrier transmission for wireless communication. He has published more than 100 papers in journals and conferences. He has served as Chair of Acoustic Communication Interest Group of IEEE Technical Committee on Multimedia Communications. His research has been supported by the National Science Foundation, Air Force Office of Scientific Research, Air Force Research Laboratory, Office of Naval Research, and NASA. His work on software defined radio implementation of cognitive radio won the Best Demo Award at IEEE Globecom 2010.

Prof. Bin Wang, Wright State University

Prof. Bin Wang earned his Ph.D. from the Ohio State University in 2000. He joined the Wright State University in September 2000, where he is currently full professor of computer science and engineering. His research interests include optical networks, real-time computing, mobile and wireless networks, cognitive radio networks, trust and information security, and semantic web. He is a recipient of the US Department of Energy Career Award. His research has been supported by US Department of Energy, National Science Foundation, Air Force Office of Scientific Research, Air Force Research Laboratories, Ohio Supercomputer Center, and the State of Ohio.

Prof. Chi-Hao Cheng, Miami University

Dr. Chi-Hao Cheng received the B.S. degree in control engineering from National Chiao Tung University, Taiwan in 1991, and the M.S. and Ph.D. degrees from The University of Texas at Austin in 1996 and 1998 respectively, both in Electrical and Computer Engineering. He is currently a professor in the Department of Electrical and Computer Engineering at Miami University, Ohio. His primary professional interests lie in signal processing algorithm development and its applications in numerous communications system and component development including wireless and optical communications systems. He is co-inventors of three US patents.

Software Defined Radio Based Laboratories in Undergraduate Computer Networking Courses

Abstract

The explosion of wireless technology has made it a hot topic in undergraduate education. Many undergraduate students are intrigued by the secrets behind wireless communication and networking, but few institutions can provide hands-on laboratories in their networking courses due to expensive hardware equipment. Funded by a collaborative NSF TUES type II project, a series of affordable and evolvable software defined radio (SDR) based laboratories was implemented and institutionalized at three institutions to demonstrate its capability and adaptability. As a participating institution, Central State University worked closely with Wright State University and Miami University and successfully adapted the novel SDR based laboratories. We further initialized our own laboratory modules to improve undergraduate students' understanding and learning. The laboratory modules were integrated into two undergraduate level networking-related courses, and the course assessment showed positive learning outcomes. The exploratory project is a work in progress and we will continue the development in order to lead a national model of SDR laboratory based courses.

1. Introduction

Today, there are more than 355 million wireless subscribers in the US, which is 110% of the US population. There are 208 million smart phones and 35 million tablets, and 44% of US households are wireless only. It is reported that every $1 invested in wireless deployments amounts to $10 in added Gross Domestic Product, including impact upon jobs and wages[1]. The explosion of wireless technology has made it a hot topic in undergraduate education. Many talented students in the Science, Technology, Engineering, and Mathematics (STEM) programs are intrigued by the secrets behind wireless devices and eager for hands-on experience to discover how they work. However, few institutions can offer a laboratory in their networking courses at the undergraduate level, simply because the traditional approach of conducting networking laboratories is too expensive. For example, the Telecommunications Instructional Modeling System (TIMS) communication laboratory equipment[2] used by Auburn University[3] and Georgia Tech[4] in their curricula costs $100,000 for one basic setup, and upgrading the hardware for different laboratory modules would cost even more. Thus, a more affordable solution to conduct networking laboratories is desired, and it should be made evolvable so that it can be easily adapted for various educational programs.

Software Defined Radio (SDR)[5] is a radio communication system where most traditional hardware components are implemented by means of computer software. A basic SDR system may only consist of a personal computer equipped with a sound card or a Radio-frequency (RF) hardware frontend. Instead of using expensive special-purpose hardware such as RF signal generator, spectrum analyzers, or modulators/demodulators, most signal processing tasks are handled using the computer's general purpose processor. Such a system produces a radio which can receive and transmit various radio protocols (waveforms) based on the software used. A RF frontend usually provides better performance than an integrated sound card, and its cost is relatively low. For example, the price of an Universal Software Radio Peripheral (USRP) motherboard[6] is usually under $1,000. As for the software solution, a few number of software products that support SDR can be found in the market. For example, a $5,511 teaching bundle is offered by National Instruments[7], which includes two pieces of SDR boards and one piece of courseware with limited functionality. Another software product, the Academic LabVIEW suite, also offered by National Instruments, costs $1655 to $2999 for a one-year subscription depending on different versions. Compared to these commercial software systems, a preferable solution is GNU Radio[8], a free and open-source software development toolkit that can create various signal processing blocks and can be combined with external RF frontend to create SDR, or work alone in a simulation-like environment. Due to the high flexibility of GNU Radio, it is widely used in academic and commercial environments to support wireless communication research and real world radio systems. Thus, using USRP and GNU Radio, SDR based laboratory modules can be created in a more affordable and evolvable manner.

2. Laboratory Establishment

In 2013, Central State University were funded by a collaborative NSF TUES type II project. The major goal of the project is to develop a suite of evolvable SDR based experiments and laboratories, and demonstrate its capability and adaptability in three institutions: Wright State University, Miami University and Central State University. Compared to the other two institutions, our university is a relatively small institution with a large diverse population of students. SDR based experiments or laboratories had never been introduced to the students before, which made us a suitable experimental group, but also made it more difficult for the students to understand and accept SDR based laboratories.

2.1. Hardware Setup

The fund for hardware purchase was not requested in this particular project. In order to experience and develop SDR based laboratory modules, we borrowed one set of USRP hardware[6] from Wright State University. It has two USRP1 motherboards ($719 per each), each comes with an Altera Cyclone FPGA, 64 MS/s dual ADC, 128 MS/s dual DAC, and USB 2.0 connectivity. The USRP1 platform can support two complete RF daughter boards and can operate from DC to 6 GHz. The daughter board we used on each motherboard is RFX400, which has 2 quadrature frontends for transmitting and receiving, and the bandwidth is 40MHz for both frontends (see Figure 1).

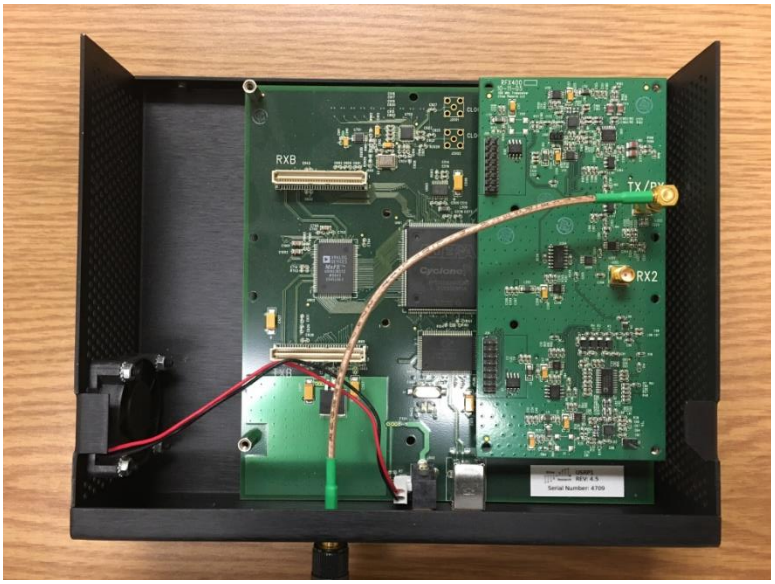

Figure 1: A USRP1 Motherboard with RFX400 Daughter Board

2.2. Software Setup

In order to improve undergraduate students' understanding and learning, the following step-by-step laboratory modules were developed:

I) **Ubuntu/Windows duel operating system installation**. Many SDR development tools, including GNU Radio, only officially support Linux operating system. However, some undergraduate students may not be familiar with Linux at the first place. Developed by Central State University, this module offers a complete guide on how to install Ubuntu[9], a Debian-based Linux operating system, on a Windows PC. The module introduces basic Ubuntu operations and commands as well. As an alternative solution, this module also provides a guide on how to install and run Ubuntu as a guest operating system on Microsoft Windows using Oracle VirtualBox[10].

II) **GNU Radio and USRP Hardware Driver (UHD) installation**. Developed by Central State University, this module offers a complete guide on how to install GNU Radio and UHD on Ubuntu. Depending on their background and interest, students can install GNU Radio and UHD in two different ways: The first way is to use the build-in script provided by Marcus Leech[8], which can automatically download and install GNU Radio and UHD with a few command lines in a terminal window. The second way is to manually install UHD and GNU radio from the sources. This approach requires more work, but the students who explore this approach can acquire the latest version of GNU Radio components and make changes to the GNU Radio core.

III) **GNU Radio and USRP Testing**. Adapted from Wright State University and modified by Central State University, this module offers various ways to test and debug GNU Radio, such as checking UHD connection, running GNU Radio examples, and creating signal processing blocks. The module provides several laboratories of Python programming, which teach students how to edit and run Python source code that calls the blocks. For example, in one laboratory the students need to connect two sine waves into the left and right channels of a sound card, respectively, and create a dial tone generator (see Figure 2). An incomplete source code (without the highlighted statements) is provided and the students need to complete and test the code by themselves. This experiment does not require USRP front, therefore it can be done on any computer with a sound card and an Ubuntu operating system.

```python
#!/usr/bin/env python

#dial_tone_example.py

# import/include blocks from the main gnu radio package
from gnuradio import gr
from gnuradio import audio
# create the top block
class dial_tone(gr.top_block):
    def __init__(self):
        gr.top_block.__init__(self)
        # setup some parameters
        sampling_rate = 44100
        amplitude = 0.1
        # create two signal sources
        src0 = gr.sig_source_f(sampling_rate, gr.GR_SIN_WAVE, 440, amplitude)
        src1 = gr.sig_source_f(sampling_rate, gr.GR_SIN_WAVE, 540, amplitude)

        #create a signal sink
        dst = audio.sink(sampling_rate)

        #connect all the blocks together
        self.connect(src0, (dst, 0))
        self.connect(src1, (dst, 1))

# run the flow graph, the main function
if __name__ == '__main__':
    try:
        dial_tone().run()
    except KeyboardInterrupt:
        pass
```

Figure 2: Python Code for a Dial Tone Generator

IV) **GNU Radio and USRP Implementation**. Adapted from Wright State University and modified by Central State University, this module includes three independent laboratories, which are "Amplitude Modulation Implementation", "Frequency Modulation Implementation", and "Digital Modulation Implementation: Transmitter and receiver". In these laboratories students can use GNU Radio Companion (GRC), a graphical tool included in GNU Radio, to create signal flow graphs with signal processing blocks (see Figure 3). When a student choose some blocks from the right-side menu and connect them together, a flow graph will be created. The corresponding source code will be automatically generated and can be saved in a *.py* file. Students can change the properties of the blocks in GRC or edit the source code directly.

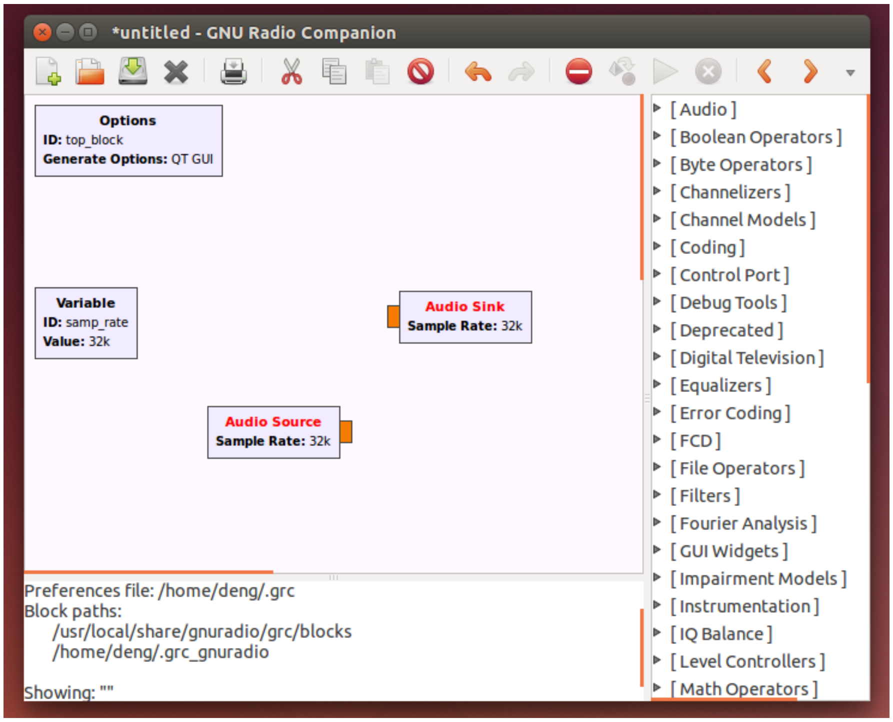

Figure 2: GNU Radio Companion Interface

Figure 3 shows a sample GRC diagram flow graph that contains a signal source, a throttle to limit the sample rate, a scope sink for time domain display, and a FFT sink for frequency domain display. Figure 4 shows the corresponding FFT plot and scope plot.

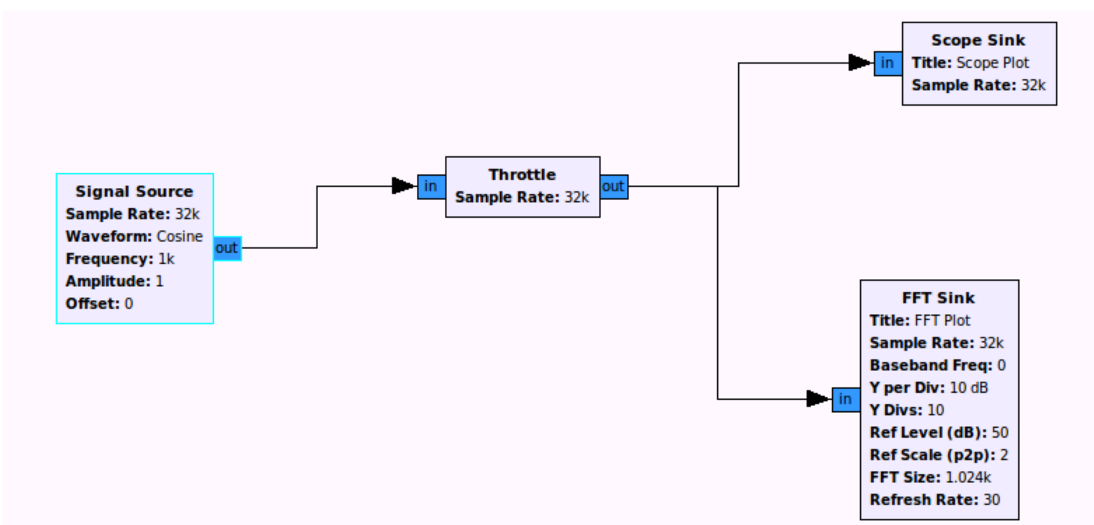

Figure 3: A sample GRC Block Diagram

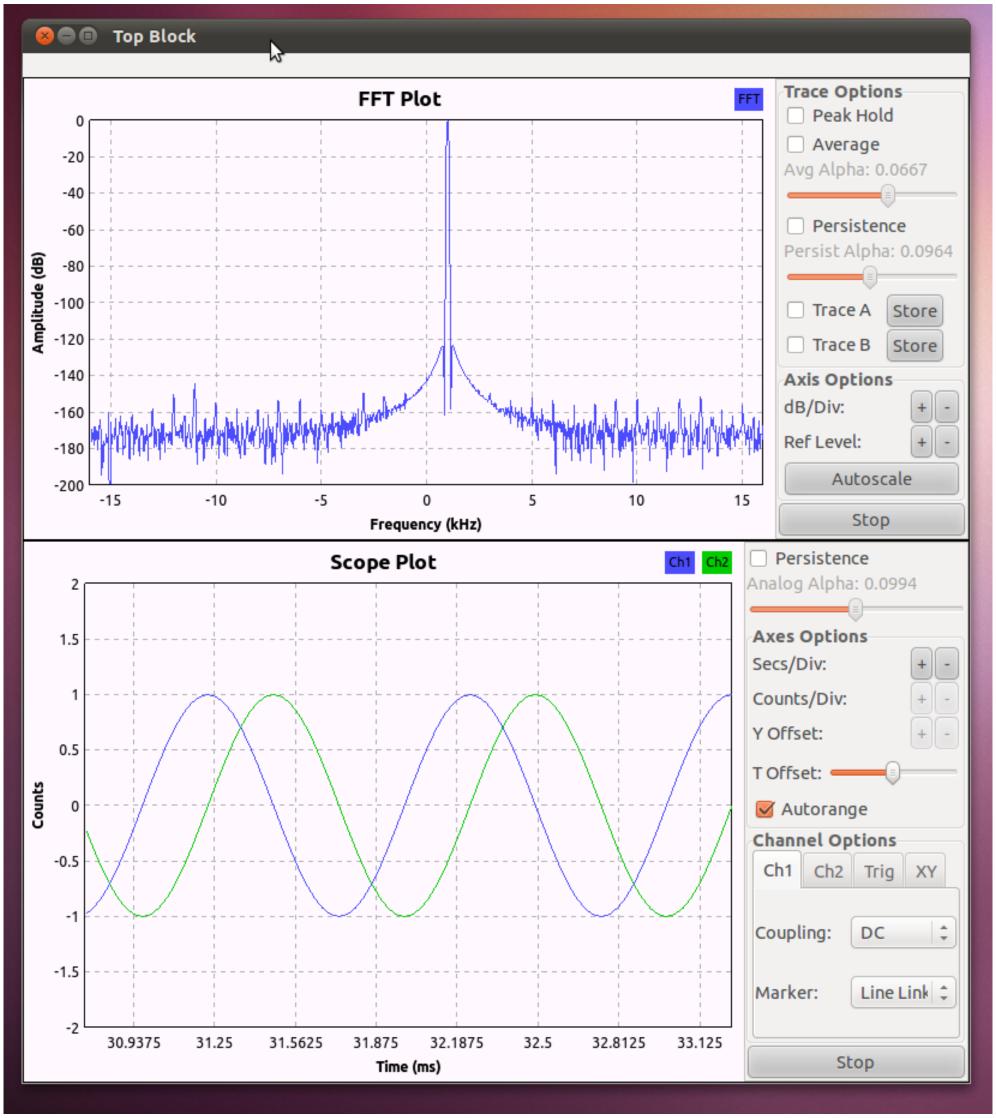

Figure 4: FFT Plot and Scope Plot

Students are asked to complete different signal processing tasks in each laboratory. For example, in "Frequency Modulation Implementation", students need to transmit an audio file between two computers using USRP. In order to do that, students need to connect the right blocks together, and use the right frequency band to implement a FM transmitter. At the receiver side, students need to complete another Python code in order to observe the signal in frequency domain. Figure 5 shows the outcome of the completed code. "Data from USRP" displays the spectrum of passband signal, whose center frequency is located at 430MHz. "Post Demod" display the spectrum of demodulated baseband signal, whose center frequency is at 0 Hz.

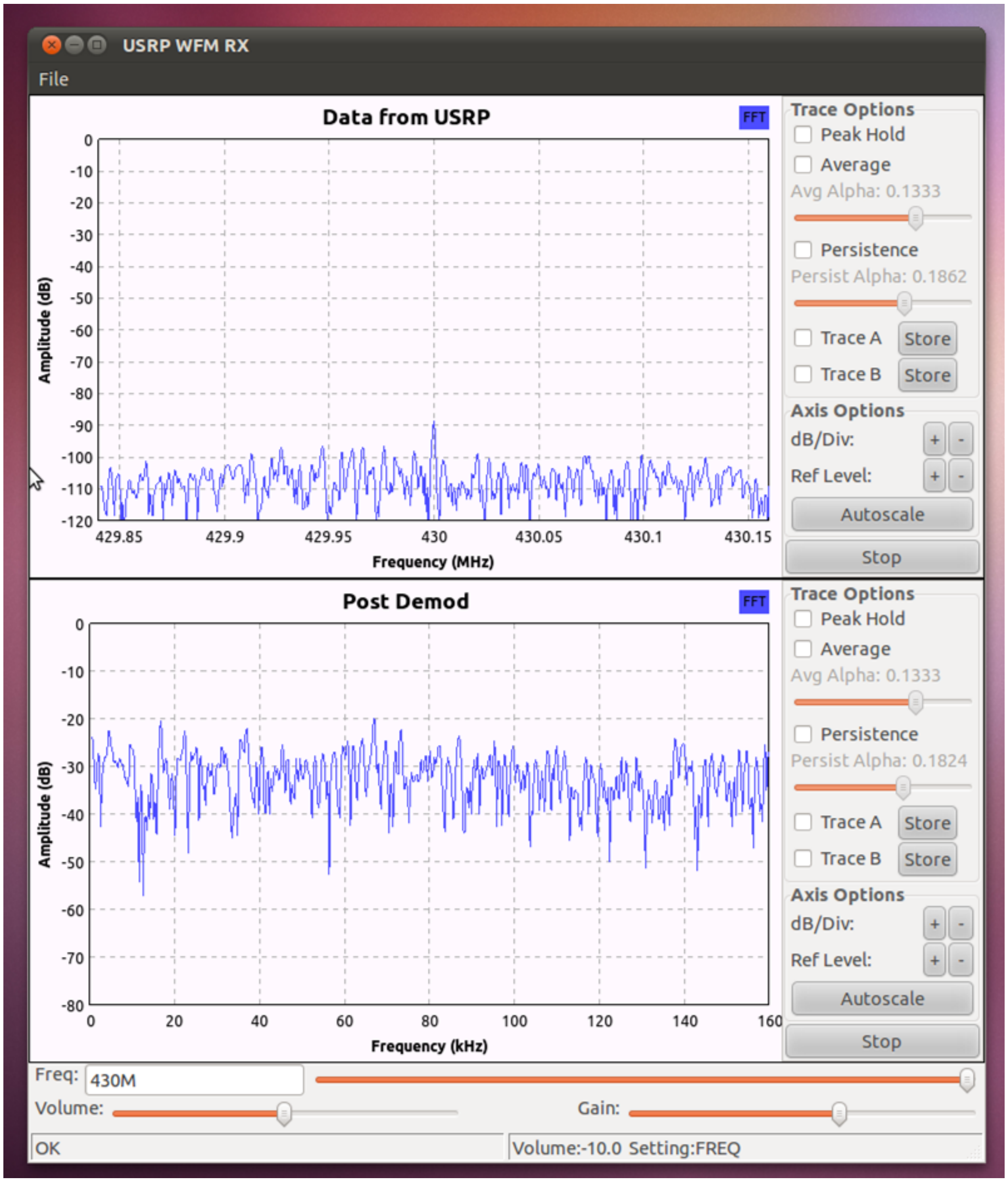

Figure 5: Received Audio Signal in Frequency Domain

In our belief, module I-IV shall cover the basics of GNU radio and USRP that are necessary for undergraduate students, and provide adequate guidelines and examples that allow an institution to build SDR based laboratories from scratch. The complete Python code for each laboratory is provided, so that instructors will be able to edit the code under various circumstances, and students can benefit from different ways of exploring GNU SDR library.

2.3. Course and Research Setup

The four modules were integrated into two of the Computer Science courses: "Computer Networks" and "Senior Project". Both courses are offered once a year. Computer Networks is a major required course for freshmen and sophomores, and also available as an elective for other majors. Senior Project is a major required course only available for seniors in Computer Science major. The laboratory sessions were also available to students who did not register Computer Networks or Senior Project courses. These students were encouraged to attend the laboratory sessions and complete the laboratory reports as well. Advanced students who showed sufficient understanding and strong interest in developing SDR based laboratories were awarded as research assistants. Research assistants were trained to carry out research work according to the project schedule, such as system environment maintenance, module debugging and developing new modules. They were expected to improve their understanding in wireless communication and networking area and gain valuable research experiences that could help them to be graduate school ready.

3. Evaluation & Assessment

3.1. Assessment Methods

The SDR based laboratory sessions were offered in Spring 2015 and Spring 2016. The students' learning outcomes were evaluated and assessed with quantitative and qualitative metrics. The students' academic performance was analyzed based on their laboratory reports and project reports. We also collected anonymous questionnaires and gave comprehensive face-to-face interviews. The laboratory sessions were designed as follows: For the students in Computer Networks course, all laboratory sessions were optional and offered after contact hours. If a student demonstrated adequate knowledge and skills in his/her laboratory report, a small extra credit (no more than 5% of the final grade) will be granted and therefore the student's final grade will be improved. An anonymous questionnaire was collected from each participant after the completion of a laboratory. A comprehensive interview was given to each participant by the end of semester. For students in Senior Project course, all laboratory sessions were elective. By the end of the first month of the semester, students were asked to choose their senior project topics from a number of options, including traditional topics (e.g., app development or website management) and SDR-related topics (e.g., revising laboratory modules towards new application). There was no bonus attached to SDR-related topics. Questionnaires and interviews were given to the participants to collect feedback.

3. 2. Assessment Results

The assessment results were summarized in Table 1-2.

Course Title: Computer Networks	Spring 2015	Spring 2016
Student enrollment	4	7
Students who attended at least one laboratory session	1	6
Students who demonstrated adequate knowledge and skills (received more than 70% credits) in their laboratory reports	1	5
Students who agreed that they gained unique knowledge and enhanced their scientific and/or technological understanding in individual laboratory sessions	1	6
Students who agreed that the course with SDR-based laboratory sessions progressively deepen and broaden their skills	1	6
Students who agreed that adding SDR-based laboratory sessions to the course made it more interesting	1	6
Students who agreed that they were better motivated and engaged to stay in the Computer Science program after SDR based laboratories being offered	1	5

Table 1: Course Assessment: Computer Networks

Course Title: Senior Project	Spring 2015	Spring 2016
Students enrollment	2	8
Students who attended at least one laboratory session	1	5
Students who agreed that they gained unique knowledge and enhanced their scientific and/or technological understanding in individual laboratory sessions	1	5
Students who agreed that the course with SDR-based laboratory sessions progressively deepen and broaden their skills	1	5
Students who agreed that the course was enriched by offering SDR based laboratories	1	5
Students who agreed that they were better motivated and engaged to stay in the Computer Science program after SDR based laboratories being offered	1	5
Students who chose SDR-related project topics over other topics	1	4
Students who demonstrated adequate knowledge and skills to complete a satisfactory project report with SDR related topic (received more than 70% credits)	1	4

Table 2: Course Assessment: Senior Project

In addition, two students who did not enroll in Computer Networks or Senior Project showed strong interest in SDR based laboratories and attended several laboratory sessions in Spring 2016. Adding them to the overall laboratory attendance, we have a total of 15 undergraduate student participants.

4. Conclusion

Although not statistically conclusive, the assessment results suggested that the laboratory modules could benefit a diverse population of students by motivating, engaging, and enhancing their learning and understanding. The results demonstrated a positive example of integrating modern technology and research into undergraduate STEM education.

From research perspective, the collaborative project exposed participating undergraduate students to the possibilities of graduate study and encouraged them to choose a career path involving research. We observed that the students developed more intellectual confidence as they were awarded as research assistants. They benefited greatly when collaborating with other students who share their commitment to science, mathematics, and engineering. The project also advanced the research skills of the undergraduate students and enhanced the employment opportunities for these students. For instance, a research assistant recently attended an on-campus job fair before his graduation, and was immediately recruited by an IT company.

With three institutions participating, the collaborative project had an immediate local impact. The hardware solution for the project is inexpensive, and the software we employed to develop the laboratories is open source and free, with a large supporting community. The proposed laboratory suite is therefore affordable and evolvable, and can be easily adapted by other institutions. The exploratory project is a work in progress and we will continue to develop the suite in order to lead a national model of SDR laboratory based wireless communication and networking courses.

References:

1. Ten Amazing Facts About Wireless Tech, http://www.cioinsight.com

2. Telecommunications Instructional Modeling System, http://www.tims.com.au

3. Auburn University Wireless Engineering Undergraduate Program, http://www.eng.auburn.edu/wireless/

4. Georgia Institute of Technology, ECE 4602, http://www.ece.gatech.edu/academics/undergrad/course_schedule.pdf

5. Markus Dillinger, Kambiz Madani and Nancy Alonistioti, *Software Defined Radio: Architectures, Systems and Functions*, Wiley & Sons, 2003

6. Ettus Research, http://www.ettus.com/product/category/USRP-Bus-Series

7. National Instruments Digital Communication Teaching Bundle, http://sine.ni.com/nips/cds/view/p/lang/en/nid/213007

8. GNU Radio, http://gnuradio.org

9. Ubuntu: The leading OS for PC, tablet, phone and cloud, www.ubuntu.com

10. Oracle VM VirtualBox, https://www.virtualbox.org