Paper ID #10633

Ultra Low-Cost Software-Defined Radio: A Mobile Studio for Teaching Digital Signal Processing

Dr. Cory J. Prust, Milwaukee School of Engineering

Dr. Cory Prust is Assistant Professor in the Electrical Engineering and Computer Science Department at Milwaukee School of Engineering (MSOE). He earned his BSEE degree from MSOE in 2001 and his Ph.D. from Purdue University in 2006. Prior to joining MSOE in 2009, he was a Technical Staff member at MIT Lincoln Laboratory. He teaches courses in the signal processing and embedded systems areas.

Dr. Steven Holland, Milwaukee School of Engineering

Steven S. Holland (M '13) was born in Chicago, IL, in 1984. He received the B.S. degree in electrical engineering from the Milwaukee School of Engineering (MSOE), Milwaukee, WI, in 2006, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts Amherst, in 2008 and 2011 respectively. From 2006 to 2011, he was a Research Assistant working in the Antennas and Propagation Laboratory (APLab), Department of Electrical and Computer Engineering, University of Massachusetts Amherst. He was then a Senior Sensors Engineer with the MITRE Corporation, Bedford, MA from 2011 to 2013. Since 2013 he has been an Assistant Professor of Electrical and Computer Engineering at the Milwaukee School of Engineering.

His research interests include ultrawideband antenna arrays, electrically small antennas, Radar systems, digital and analog circuits, and engineering education.

Dr. Holland received the Best Student Paper Award at the 2010 Antenna Applications Symposium, Allerton Park, Monticello, IL, and is a member of Tau Beta Pi.

Dr. Richard W. Kelnhofer, Milwaukee School of Engineering

Dr. Kelnhofer is the Program Director of Electrical Engineering and an Assistant Professor at Milwaukee School of Engineering (MSOE). Formerly, he held engineering and managerial positions in the telecommunications industry. He received his Ph.D. in Electrical Engineering from Marquette University in 1997 and is a Professional Engineer registered in the State of Wisconsin. Dr. Kelnhofer teaches courses in circuits, communication systems, signal processing, and information and coding theory.

Ultra Low-Cost Software-Defined Radio: A Mobile Studio for Teaching Digital Signal Processing

Abstract

Software-defined radio (SDR) is being used by many institutions as a teaching tool to illustrate and explore concepts presented in signal processing and communication courses. The inherent flexibility of SDR coupled with the ability to capture, visualize, and process real-world signals provides numerous benefits in classroom and laboratory settings. Furthermore, exposure to SDR is increasingly important for students wishing to pursue careers in the telecommunication, networking, and radar fields. An undergraduate laboratory can be outfitted with relatively high-performance SDRs at a reasonable cost.

It was recently discovered that USB digital television tuners can be used as SDR receivers. Since this discovery, the tuners have been successfully used in a wide variety of applications. At a cost less than $20 (USD), these so-called “RTL-SDR” devices set a new price point for SDR technology that is particularly attractive within an educational context.

This paper presents the use of these low-cost SDRs and supporting software for teaching digital signal processing (DSP) concepts to undergraduate electrical and computer engineering students. The proposed approach creates an interactive learning environment based on mobile studio pedagogy. A series of studio projects have been developed, each of which requires implementation and testing of DSP algorithms on data received by student-owned SDRs. Data sources include signals of opportunity as well as instructor-generated test signals. The result is a mobile learning environment in which students can visualize and apply abstract theoretical concepts, implement real-time algorithms, and rapidly test their designs using real-world data.

Introduction

Software-defined radio (SDR) technology has transformed much of the modern communications and networking fields. Generally speaking, an SDR is a flexible hardware platform in which the majority of the radio functionality is implemented in software. While a traditional hardware radio is designed specifically for one or a small number of applications, the functionality of a single SDR can be modified via firmware updates and changes to the back-end signal processing in order to implement a wide variety of systems. Many modern SDRs are coupled with embedded processors and networking interfaces, allowing stand-alone systems to be updated and reconfigured remotely. Continuing advances in hardware performance are broadening the scope of SDR applications to areas such as radar, radio-frequency identification, radio astronomy, and various other forms of remote sensing.

Electrical engineering departments at many universities have responded to the advent of SDR by infusing the technology into their curriculum. Because the technology lies at the interface between many engineering fields (communications, networking, embedded systems, digital signal processing, FPGA development, etc.), the implementations and types of courses vary tremendously. The exact manner in which SDRs are being used within a curriculum varies by institution. Some have essentially replaced the traditional hardware-based communications and networking laboratories with SDRs, thus exposing students to modern implementations[1,2]. Because the radio functionality is now primarily implemented in software, many of these courses focus on digital signal processing algorithms and techniques. Other institutions are using SDRs to supplement existing courses and other initiatives with real-world examples and demonstrations[3,4,5,6]. The SDR and its supporting software tools provide a new gateway through which students can learn and better understand the concepts being investigated.

Decreasing equipment costs have made SDRs affordable for most institutions. Examples include the Ettus Research[7] USRP family of radios which, depending on the exact feature set, range in price from $700-$1700 (USD) per unit. Therefore, an undergraduate laboratory could be outfitted with multiple SDR stations at a reasonable cost. The inherent flexibility of the hardware and its broad range of uses makes the investment attractive, especially compared to the much larger costs associated with traditional dedicated hardware.

This paper presents an approach to teaching signal processing concepts to senior undergraduate students that utilizes a very low cost, receive only SDR. The primary intent is to establish a mobile studio, in effect giving each student unlimited access to the hardware and software tool set for project work. The approach utilizes the so-called “RTL-SDR” device for data collection, paired with GNU Radio and MATLAB for signal analysis and algorithm development. The remaining sections describe the benefits of a studio pedagogical model, provide details regarding the intended course structure, and discuss a series of studio projects to be completed by students.

Mobile Studio Design Pedagogy

The studio design model is a central part of the curriculum for architectural programs, where it is used to foster creativity and develop a student’s understanding of the design process. The studio design model also assists students by integrating other issues such as societal impact and legal requirements into the design process. Because the design process is core to engineering, studio design principles are increasingly being used in engineering curriculums[8].

Features of the design studio include project based assignments (studio projects) that span multiple weeks or the whole semester. Students are provided with criteria for each project that may include an estimated budget, minimal specifications, and basic requirements. The criteria provide guidance but ultimately allows students to explore different solutions. Students are encouraged to be creative, experiment, and develop multiple design iterations. Finally, a working prototype of the solution is required at the end of each studio project assignment. The studio design model provides students with crucial “hands-on” experience and promotes a student’s ability to integrate theoretical concepts with real applications.

Depending on the scope of the studio project or the educational discipline, the design studio model may only be found in capstone courses. For example, the instrumentation and development suites found in electrical engineering laboratories can be costly. This fact places constraints on the number of laboratory classrooms that are equipped and hence limits the ability to provide the studio design based model across a broader range of courses in an electrical engineering curriculum. In the past, this may have not been a significant problem as many students entering an electrical engineering program had previous hands-on experience. However, this has changed and the latest generation of students entering electrical programs have a wide variety of experiences and capabilities. This later fact makes the use of studio design courses more important than ever.

In order to promote studio design into a broader range of electrical engineering courses, a mobile based design studio model was developed[9]. The model was based on the use of a portable cost effective electronics instrumentation platform. Because of the low-cost, each student could afford to purchase their own platform and the small size allowed students to explore and experiment outside the physical constraints of a laboratory setting. The use of mobile studio concepts based on low-cost hardware platforms have found favor in other areas of electrical engineering as well. Three years ago our electrical engineering program transitioned its embedded mircoprocessor sequence over to a mobile studio model using the low-cost Arduino platform. Lessons learned from this experience have initiated and informed the development of this work’s proposed DSP course.

Teaching DSP applications – current approach

Like many other electrical and computer engineering programs, the curricula at Milwaukee School of Engineering includes a required course in digital signal processing. The primary audience for the course is junior-level students. A subset of those students then takes one or more elective level courses in related areas. It is one of those elective courses, *Applications of Digital Signal Processing*, which is the subject of this paper.

As its title suggests, the course focuses on real-world applications of digital signal processing techniques. It is a heavily laboratory-oriented course through which students implement and test DSP algorithms on actual hardware. The course structure is project-based, through which each student independently (or sometimes in small groups) must implement and test actual DSP-based systems. For many years, the popular Texas Instruments TMS320C6713 DSP Starter Kit (DSK) has been the primary hardware platform used by students. The hardware is capable of processing stereo audio signals in real-time, thus allowing students to implement algorithms such as FIR and IIR digital filters, adaptive filters, modulators and demodulators, and Fast Fourier Transforms. Historically, student projects have often included examples related to communication systems such as quadrature amplitude modulation, frequency modulation, and single-sideband communication. The focus on communication systems has been driven by instructor interest and expertise, as well as the fact that the majority of students who enroll in the course have also completed a required physical layer analog and digital communications system course.

A typical project involves prototyping of the DSP algorithm, usually in MATLAB, to help students gain an understanding of the underlying concepts. Because the eventual goal of the project is a real-time implementation, this prototyping is often carried-out in a manner which mimics a sample-by-sample simulation. These simulations often begin with simple test signals, then proceed to more complex signals including, whenever possible, real data. Next, students begin experimenting with the real-time hardware implementations. We currently use the TI Code Composer Studio software for programming the DSK units in C. In cases where the project is a system involving many sub-components (e.g., multiple digital filters), we encourage students to implement, test, and characterize each sub-component before assembling the complete system. With proper guidance and suggestions, we encourage students to generate their own test signals for exercising the algorithms. Our overall approach has been strongly influenced by the large community of educators using the DSK hardware in undergraduate laboratories.

Student access to the expensive DSK hardware ($395 USD) is limited by a number of factors. Our current inventory does not allow long-term loans (i.e., for the duration of the course) of the hardware to each student. Further, the DSK units are in high demand, particularly when the required junior-level course which uses the same hardware for several lab experiments runs in the same term. For that reason, even short term loans (i.e., for an evening or weekend) are often not possible. These limitations have become increasingly problematic especially since many of the student's embedded programming courses have migrated to a student-owned mobile hardware platform. This allows unlimited, on-demand access for their project work at an extremely low cost. So in essence, our student's expectations regarding cost and access to hardware resources have changed significantly. While it is certainly possible for some students to purchase their own equivalent hardware, for example the similarly featured TMS320C6748 DSP Development Kit ($195 USD), very few students have opted to purchase their own. Therefore, an easy to use, very low-cost platform was required in order to implement studio projects in this course.

Teaching DSP applications – proposed approach

It was recently discovered that USB DVB-T dongles meant for digital television reception can be used as software-defined radio receivers. These devices, based on the Realtek RTL2832U demodulator and hence the name RTL-SDR, are capable of streaming 8-bit I/Q samples through a USB 2.0 interface at a maximum rate of 3.2MS/s. A variety of RTL-SDR variants are currently available, the primary difference being the specific RF tuner paired with the RTL2832U. Figure 1 below shows one such variant based on the Rafael Micro R820T tuner which is capable of tuning between 24-1766MHz. This particular dongle is widely available at a cost of $18 USD.

Figure 1: The RTL-SDR based on the R820T tuner and the mobile studio

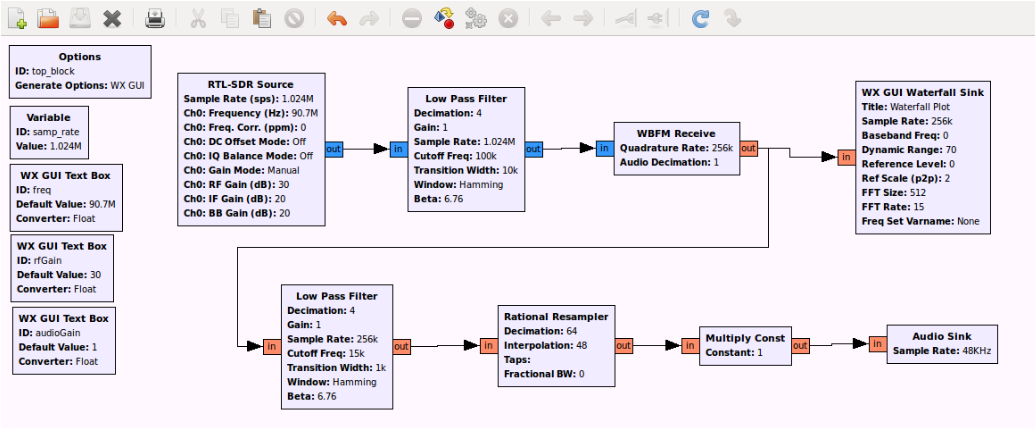
Together with a host PC running the appropriate software, each student can have their own mobile platform cable of recording, analyzing, and processing a wide-range of signals. Hence, these ultra-low cost RTL-SDR devices provide a mobile studio for students studying digital signal processing. We expect that the ability to capture and process actual real-world signals using their own hardware will be particularly impactful for many of our students. Due to the relatively large tunable bandwidth, students can explore a large range of the RF spectrum and “see” signals such as FM radio, digital television, amateur radio, GPS, and commercial aircraft ADS-B transmission.

For the purposes of our course, the mobile studio consists of the following:

- An RTL-SDR and stock antenna.
- A modern laptop computer running Windows.
- Supporting software packages, including
    - SDR# - a PC-based application for SDR providing real-time radio functionality, data recording, and fully supports the RTL-SDR devices.
    - MATLAB
    - GNU Radio, accessible from a bootable Ubuntu Linux Live USB flash drive

The basic course structure remains identical in that the primary intent is teaching applications of digital signal processing. We maintain the strong emphasis on using MATLAB to prototype and investigate the DSP algorithms. The mobile SDR platform can be leveraged to enhance those simulations by the inclusion of real data that is recorded by the student. Data can be easily recorded using software such as SDR#[10] and processed off-line.

From the perspective of a real-time DSP system, the laptop computer is now the real-time processor. There are a number of software packages that can be used to access the real-time data stream from the RTL-SDR. We propose to use the GNU Radio[11] package. GNU Radio is a freely available, open-source tool kit for developing software defined radio applications. Of particular interest for our course is the GNU Radio Companion (GRC) which provides a flow-diagram based graphical user interface to GNU Radio. Through GRC, students can rapidly prototype and test signal processing algorithms, as well as view results. GRC provides a variety of visualization tools such as oscilloscope displays, FFT and waterfall displays, as well as audio sinks that interface to the PC soundcard. Figure 3 below shows a screen capture from GRC that implements an FM radio receiver using a built-in WBFM Receive block and includes an audio sink to play back demodulated sound.



Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 1.024M

WX GUI Text Box
ID: freq
Default Value: 90.7M
Converter: Float

WX GUI Text Box
ID: rfGain
Default Value: 30
Converter: Float

WX GUI Text Box
ID: audioGain
Default Value: 1
Converter: Float

RTL-SDR Source
Sample Rate (sps): 1.024M
Ch0: Frequency (Hz): 90.7M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 30
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Low Pass Filter
Decimation: 4
Gain: 1
Sample Rate: 1.024M
Cutoff Freq: 100k
Transition Width: 10k
Window: Hamming
Beta: 6.76

WBFM Receive
Quadrature Rate: 256k
Audio Decimation: 1

WX GUI Waterfall Sink
Title: Waterfall Plot
Sample Rate: 256k
Baseband Freq: 0
Dynamic Range: 70
Reference Level: 0
Ref Scale (p2p): 2
FFT Size: 512
FFT Rate: 15
Freq Set Varname: None

Low Pass Filter
Decimation: 4
Gain: 1
Sample Rate: 256k
Cutoff Freq: 15k
Transition Width: 1k
Window: Hamming
Beta: 6.76

Rational Resampler
Decimation: 64
Interpolation: 48
Taps:
Fractional BW: 0

Multiply Const
Constant: 1

Audio Sink
Sample Rate: 48KHz

Figure 3: Screen capture of a broadcast FM radio receiver in GNU Radio Companion

Along with the student-owned RTL-SDR, we also plan to leverage a more capable SDR for in-class experiments. By pairing the low-cost RTL-SDR with an SDR transmitter, such as the Ettus Research B200 shown in Figure 4, the instructor can generate a wide variety of signals that can be used for controlled experiments within the course. We plan to include such experiments in order to demonstrate lecture concepts, as active learning exercises, and to investigate anticipated “stumbling blocks” that students may encounter during their assigned projects.

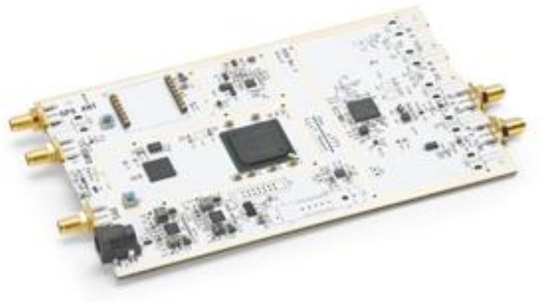


Figure 4: Photo[7] of the Ettus B200 SDR. Test signals generated by this or a similar software defined radio can be can be received by each student's mobile studio platform.

Considerations in course design

The proposed mobile studio approach based on the RTL-SDR devices presented a number of pedagogical opportunities, as well as challenges, compared to our previous approach. In this section, we describe several of the key items considered in developing our proposed approach.

The opportunity for each student to capture and process real signals that they themselves collect was a strong motivator for our proposed approach. Observing phenomena in real data vividly demonstrates the theoretical concepts and creates an obvious link to practical implementations. Seeing or hearing their own system operate on real data should boost student interest and excitement in signal processing topics.

Our previous approach placed a strong focus on real-time implementations using the DSK units. The DSK software toolset, Code Composer Studio, provides excellent tools for debugging and profiling the student projects. While not a primary emphasis of the course, these tools helped facilitate discussions of numerical precision and computation time, both of which are essential in a real-time system. Our new approach to real-time processing, specifically GNU Radio running on a laptop, does not offer the same visibility into these issues. At least for some studio projects, we anticipate placing a stronger emphasis on off-line processing algorithms developed in MATLAB.

While GNU Radio can be run on Windows, the primary user and developer platform is Linux. The majority of our students have little to no experience on Linux platforms. Our approach is to use bootable USB flash drives containing a complete Ubuntu Linux distribution and pre-installed version of GNU Radio. This approach significantly reduces the barrier faced by new users. Our institution's technology package provides students with laptop computers, which minimizes hardware compatibility issues.

GNU Radio, and particularly GRC, provides tremendous capability in an easy to use framework. As such, the tool must be used judiciously, especially by students who are still forming their knowledge base and have limited experience. Because the focus of the course is on digital signal processing algorithms and their implementation, asking students to simply connect together pre-defined GNU Radio blocks is somewhat in contradiction with those fundamental goals. Therefore, we plan to use the DSK units for a small number of projects early in the course where the learning goals of the projects will translate directly to the higher level functionality provided by GNU Radio blocks. A simple example of such a project is an FIR filter implementation. Implementation using C on the DSK requires students to design the filter (e,g., using MATLAB toolboxes), consider the feasibility of the design (e.g., assess the filter length), implement the difference equation, and then test the filter. We believe that knowledge gained through these exercises is essential for successful use of the same functionality in GRC.

Because the RTL-SDR devices provide I/Q samples at a high rate (up to 3.2Ms/s), real-time processing in GNU Radio often requires decimation at various stages in the signal flow. The topic of multi-rate signal processing has not previously been discussed in our course. The topic will now need to be introduced early-on. At a minimum, students must understand the key pitfalls associated with decimation and interpolation.

The topic of complex baseband signaling is often a confusing one for students. Because the SDR hardware produces I/Q samples, the mobile studio now provides a practical context in which the concept can be studied. That is, complex baseband is no longer just a lecture topic, but rather the output of a real hardware system. Student projects can emphasize and reinforce the underlying concepts.

As earlier described, we will utilize an SDR transmitter for generating test signals as part of the studio projects. These exercises offer students unique exposure to not only the full communication link, but also provides a meaningful context to discuss legal responsibilities associated with such a system. Students should be aware that there are legal implications any time a transmitter is operated, and that FCC regulation and licensing requirements must be followed. Experimentation in this course could be carried out using one of the popular unlicensed ISM bands, for example 902-928MHz. Though unlicensed, part 15 of the FCC regulations[12] limits transmitter output power at ISM bands to a maximum of 30dBm (1W), and an Effective Isotropic Radiated Power (EIRP) of 36dBm. Students could evaluate the Ettus B200 SDR transmitter to see if it abides by the FCC power limits. For example, the B200 is rated for a maximum power level of 20dBm (100mW), and a typical whip antenna mated with the radio provides an antenna gain of roughly 3dB, thus the EIRP is 23dBm and would fall well under the limits. A logical extension would ask students to determine how EIRP can be maximized while remaining below FCC limits when an external off the shelf amplifier is connected to the B200 output. Students can discuss why the power limits are in place, such as limiting overall transmission range to avoid interference with other users, and why EIRP is the appropriate figure of merit to regulate instead of simply transmitter power.

Example studio projects

In this section we present two studio projects that have been developed for use in our course. Both projects utilize the mobile studio for data collection and for development, implementation, and testing of the digital signal processing algorithms.

FM Radio

Broadcast FM Radio is a natural project topic for our mobile studio setup. Real FM broadcast signals are available essentially everywhere, at all times of day or night. Therefore, the mobile studio can be employed anywhere, at any time. The studio project has been structured to guide students through a series of activities, ultimately leading to a complete, real-time implementation of an FM receiver. Below we outline the initial phases of the studio project.

> *Part 1: FM Signal Exploration using the RTL-SDR.* Most of our students will have had some exposure to frequency modulation concepts through a previous communications systems course. While they have been exposed to concepts such as Carson's Rule and modulation indices, many of them have never seen an actual FM broadcast. Therefore, we begin the studio project by guiding students through the fundamentals of frequency modulation in lock-step with observations and analysis using the mobile studio. With a GRC flowgraph consisting of just a few blocks, students can create a real-time FFT (or waterfall) display of a local FM broadcast and listen to the broadcast signal. Observing the differences in the FM signal during music, speech, and radio-silence will be enlightening for many students.
>
> *Part 2: Demodulation using a Frequency Discriminator.* The next natural step, and first algorithm implementation, would be that of a discriminator-based demodulator. After prototyping and testing in MATLAB, students would implement the discriminator using differentiation filters. If the order of the differentiation filters is kept small, then the entire demodulator can be implemented using familiar low-level blocks (e.g., sample delays, adders) within GRC. The student's demodulators could be tested using instructor generated test signals and live FM broadcasts.
>
> *Part 3: Exploration of a broadcast FM signal.* With the FM demodulator in place, students can now examine the baseband signal. For nearby transmitters, each subcomponent (e.g., mono audio, pilot tone, stereo audio, RBDS) of the baseband signal will be clearly visible.
>
> *Part 4: Extraction of Mono Audio.* At this point, the L+R mono audio channel can be easily extracted by use of a low-pass filter. The exact mode of implementation would depend on student background and/or earlier coverage of the digital filtering within the course. This task presents an opportunity to review and/or explore digital filtering techniques (depending on student background or earlier coverage in the course).

At this point, the project could be taken in any number of directions based on instructor or student interest. One option would be extraction of the 19kHz pilot tone and recovery of the stereo audio signal. If undertaken, a worthwhile intermediate step would be an instructor generated test signal where the left and right audio channels are simple and distinct (e.g., each channel has a single tone, but of different frequencies). Another possibility would be to implement a PLL based demodulator. Through the project, students could investigate the performance differences between their discriminator and PLL implementations. Again, a carefully designed test signal transmitted to their own mobile studio would reinforce the underlying concepts. Yet another project would be recovery of the embedded Radio Broadcast Data System (RBDS) information. Such a project would likely focus on an off-line processing algorithm in MATLAB rather than a real-time implementation.

NOAA APT Weather Satellite Imagery

The U.S. National Oceanic and Atmospheric Administration (NOAA) employs polar orbiting satellites for a variety of environmental monitoring tasks. Included on these satellites is an Automatic Picture Transmission (APT) system providing image data of the earth's surface as the satellite orbits. The continuously broadcasting signal is a 256-level amplitude modulated 2400Hz subcarrier, which is then frequency modulated onto a 137MHz carrier.

Because the stock antenna included with most RTL-SDR devices is insufficient for proper reception of the satellite signal transmitted by these satellites and because the satellites pass overhead only a few times per day, having students collect their own data may not be feasible. As an alternative, we supply students with a raw data file collected using the same RTL-SDR device paired with an appropriate antenna. When possible, this data collection will take place during the term in which the course is offered so that students can participate in the collection. Capturing live data as a satellite passes overhead is a particularly engaging aspect of this project. We could also make the antenna available to students if they wish to capture their own APT signal.

Because the RF signal is frequency modulated, the initial step involves frequency demodulation. Students who have completed the *FM Radio* studio project would be able to directly apply their findings (and, with some modification, their algorithms) to this aspect of the project. Then, the image pixels must be extracted from the remaining 2400Hz subcarrier using various AM demodulation schemes. Development and testing of these algorithms are best suited for off-line processing in MATLAB.

Outgrowth of the SDR Mobile Studio

The authors are also investigating other uses for this mobile studio within the electrical engineering curriculum. A natural use is in the communication systems course, which typically introduces students to concepts such as analog and digital modulation schemes and noise analysis, among other topics. Currently the course laboratory work makes use of expensive equipment that is anchored to a specific lab space and students have limited access. The SDR hardware could replace or at least complement the existing hardware experiments to broaden concrete student interaction with the theoretical course concepts.

Similarly, radio frequency (RF) and microwave engineering courses could benefit from the mobile studio. Traditional RF instrumentation has limited mobility and is extremely expensive and delicate, thus student exposure is often limited due to time-sharing of finite hardware resources. The low-cost and inherent portability of the RTL-SDR removes these significant barriers and allows students to experimentally investigate important practical RF communication link properties such as transmitter power, antenna gain patterns, wireless propagation effects, and receiver sensitivity. One such experiment is to fix the transmitter power and have students observe changes in the received signal level as the line-of-sight path separation is increased, and then compare the results to predictions from the standard Friis free-space equation and various indoor wireless signal attenuation models[13]. Multipath propagation and fading in an indoor environment could be explored by moving the receiver and observing signal strength levels to find locations of signal drop-outs. In addition, having control of the transmitter power levels allows students to characterize the noise performance of the RTL-SDR receiver hardware, observing first-hand the change in noise levels due to changing the filter bandwidths and low-noise-amplifier (LNA) gains on the SDR receiver front end, and in tandem observe how the receiver noise level dictates the signal sensitivity of the receiver. Finally, students could observe the impact of antenna polarization and pattern effects. Students could observe the change in received signal level when the transmitter and receiver's antennas are rotated with respect to one another and compare the results with theoretical polarization loss metrics[14]. Additionally, the pattern of the transmit antenna could be approximately measured by moving the receiver around the transmitter at a fixed radius and observing the change in signal level; a simple whip transmit antenna produces a null on-axis that provides insight into antenna orientation practices.

Conclusion

A new approach for teaching digital signal processing is proposed that utilizes a mobile studio model to provide students with an immersive design experience using low-cost RTL-SDR devices and supporting software. Emphasis is placed on long-term, multi-milestone projects that cement the connections between theory and real-world applications, where students use real data they collect in signal processing systems they develop. This course will be offered in Spring 2014 at Milwaukee School of Engineering, and the execution and outcomes of this course will be detailed in a future paper.

References

1. Reis, A. L G; Selva, A.F.B.; Lenzi, K.G.; Barbin, S.E.; Meloni, L. G P, "Software defined radio on digital communications: A new teaching tool," *Wireless and Microwave Technology Conference (WAMICON), 2012 IEEE 13th Annual* , vol., no., pp.1,8, 15-17 April 2012
2. Welch, T.B.; Kent, T.; Wright, C. H G; Morrow, M.G., "An Affordable Software Defined Radio," *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th* , vol., no., pp.791,796, 4-7 Jan. 2009
3. Katz, S.; Flynn, J., "Using software defined radio (SDR) to demonstrate concepts in communications and signal processing courses," *Frontiers in Education Conference, 2009. FIE '09. 39th IEEE* , vol., no., pp.1,6, 18-21 Oct. 2009
4. Bonior, J.D.; Corum, S.; McNew, B.; Piras, A.; Witherspoon, B.; Qiu, R.C.; Nan Guo; Zhen Hu, "Using software-defined radio network testbed to enhance undergraduate education and encourage graduate level studies,"*Southeastcon, 2012 Proceedings of IEEE* , vol., no., pp.1,4, 15-18 March 2012
5. Nagurney, L.S., "Software defined radio in the electrical and computer engineering curriculum," *Frontiers in Education Conference, 2009. FIE '09. 39th IEEE* , vol., no., pp.1,6, 18-21 Oct. 2009
6. Hoffbeck, J.P., “Teaching Communication Systems with Simulink and the USRP,” *American Society of Engineering Education Annual Conference, 2012*
7. Ettus Research, LLC, http://www.ettus.com
8. Kuhn, S., “Learning from the Architecture Studio: Implications for Project-Based Pedagogy,” International *Journal of Engineering Education*, Vol 17 Nos 4 and 5
9. Millard, D.L., Chouika, M., and Berry, F., “Improving Student Intuition via Rensselaer’s New Mobile Studio Pedagogy,” *American Society of Engineering Education*, 2007
10. SDR#, http://sdrsharp.com
11. GNU Radio, http://gnuradio.org
12. *Electronic Code of Federal Regulations.* http://www.ecfr.gov. 2013.
13. Siwiak, K., *Radiowave Propagation and Antennas for Personal Communications*, Noorwood, MA, Artech House, 1995.
14. Balanis, C., *Antenna Theory and Design*, Hoboken, NJ, Wiley, 2005.