TEACHING SOFTWARE DEFINED RADIO USING THE USRP AND LABVIEW

Thad B. Welch

Boise State University
Boise, Idaho, USA
t.b.welch@ieee.org

Sam Shearman

National Instruments
Austin, Texas, USA
sam.shearman@ni.com

ABSTRACT

The universal software radio peripheral (USRP) family of products has become a popular platform for hardware-based research and test bed validations conducted by universities in the software defined radio (SDR) and cognitive radio (CR) fields. With the recently released version of National Instruments (NI) LabVIEW, the USRP now offers a scalable, simpler, and easier to use combined platform. The new software support broadens the accessibility of the USRP platform for teaching applications and will spur further adoption within university communication systems classrooms, teaching laboratories, and their natural follow-on coursework. This paper will discuss the utilization of LabVIEW-based virtual instrumentation with the USRP to rapidly create real-time communication systems demonstrations for the classroom and/or laboratory settings. The combination of the USRP, LabVIEW, and Windows support enables implementation and exploration of both foundational and more advanced concepts related to signal processing and communications.

Index Terms— Communication Engineering Education

1. INTRODUCTION

Universities have been teaching software defined radio (SDR) courses and related communication system/signal processing topics for a number of years [1-13]. Unfortunately, these courses typically experience one or more of the following problems:

a. The required equipment is expensive.
b. The signal is not actually processed in real-time (recorded and processed at a later time).
c. The required equipment or its software, have a very steep learning curve.
d. System development takes a significant amount of time, requiring several hours, if not weeks, to achieve a functioning prototype.

The universal software radio peripheral (USRP) based system we are describing addresses all of the previously mentioned problems. Specifically,

a. The complete USRP2 system that we used cost about $3000 – an order of magnitude less expensive than some systems.
b. Real-time processing is possible when data is streamed to and from the USRP2 via a gigabit per second Ethernet connection, to a reasonably powerful computer running LabVIEW software.
c. The required equipment and software is easy to install, configure, connect, and learn.
d. System development proceeds quickly.

2. THE UNIVERSAL SOFTWARE RADIO PERIPHERAL (USRP)

The USRP is a family of computer-hosted radios developed by Ettus Research [14] that allows users to create a software defined radio using any computer with a USB 2.0 or Gigabit Ethernet port. A wide variety of plug-on daughterboards allow the USRPs to be used at different radio frequency bands. Daughterboards are currently available from DC to 5.9 GHz.

Since the FPGA's primary purpose on the mother board is to provide signal filtering and rate conversion from the system's analog-to-digital converter (ADC) and to the system's digital-to-analog converter (DAC), its firmware can be modified to meet your particular needs or just downloaded without modification, as a preconfigured file. The decision to be personally involved in, or knowledgeable of, the algorithm development and/or programming of the FPGA depends on the skill set that you are trying to impart to your students.

Some of the USRP family support multi-radio cooperation using multiple-input, multiple output (MIMO) techniques. This is enabled by installing a MIMO interconnecting cable between two USRP devices. For example, four USRP devices with two MIMO interconnecting cables would

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

allow for two transmit and two receive radios. This could form a “2 x 2” MIMO link.

In this development effort, we exclusively utilized several USRP2 systems configured with a WBX daughterboards. The WBX daughterboard [15] is capable of Full-duplex transceiver over a range of 50 MHz to 2.2 GHz. We selected the WBX daughterboard to provide a wide range of both transmitter and receiver frequencies, as well as offering access to a large number of real-world radio signals for students to explore. For most students, the utilization of real-world signals during projects, homework assignments, and/or laboratory exercises, has proven very motivational. Since no single daughterboard will cover all of the possible usages and frequency ranges, a wide variety of standard and custom configuration boards are available from Ettus Research [16].

3. HOST-BASED PROCESSING WITH LABVIEW

The USRP2 relies on host-based software for processing and synthesis of communications signals for transmit or receive. Programmable control of the USRP hardware and software transfer of the data for transmit or receive is provided through LabVIEW driver, newly developed by Ettus Research. This driver is the “enabling function” or “glue” that allows for this seamless, bi-directional communication between the USRP and the host computer. It offers an object-oriented application programming interface (API) that can be called directly or indirectly from a variety of programming languages, operating systems, and development environments. For example, the driver can be used with GNURadio [19] a widely used open-source library and programming framework that is popular for Linux-based development of host-based SDR /USRP applications for the USRP with C++ and Python-based programming.

A straight forward comparison of traditional communication system design and a USRP/LabVIEW-based design follows. The USRP performs the traditional analog signal conditioning functions required of almost any radio, the required signal conversion (ADC and DAC), and then performs the rate conversion necessary to provide in-phase and quadrature (I & Q) signal samples, both to and from, the processor. In a USRP/LabVIEW-based system the powerful, but cost effective processor is replaced by, for example, a multi-core processor, with a phenomenal amount of processing resources, memory, and flexibility. In some ways, this approach is wasteful, but from an educator’s prospective, it adds flexibility to the variety of systems that can be quickly designed merely by reconfiguring the existing system.

Any host computer capable of running the latest version of LabVIEW [20], that also supports Gigabit Ethernet, should be able to support a USRP-based development effort. High speed, multi-core processors, with extra system memory, will permit higher performance projects to be implemented.

4. POSSIBLE USES FOR THIS SYSTEM

In an academic setting, USRP’s can be used in courses, laboratories, and research involving;

a. Software defined radio (SDR).
b. Cognitive radio (CR).
c. Digital signal processing (DSP).
d. Multi-rate DSP.
e. Communication systems.
f. FPGA system design and implementation.
g. A wide variety of other areas (e.g., homeland security applications, satellite systems, radio astronomy, wildlife tracking, radio frequency identification (RFID), medical imaging, sonar systems, and customizable test equipment development) [21].

5. EXAMPLE OF A STUDENT'S PROGRESS

5.1. Week One - a basic transmitter

In the first week, an independent study undergraduate student with no prior experience with LabVIEW was introduced to the USRP2/LabVIEW setup and tasked with creating a binary phase shift keying (BPSK) transmitter. Within days the student was comfortable enough with LabVIEW and the provided examples to create a working system. Possible uses for such a system include:

a. With rectangular data pulses (bits), the Sinc squared spectrum of a random bit stream becomes apparent.
b. The relationship between bit rate and first null bandwidth becomes apparent.
c. The fact that nulls occur at the spacing of the bit rate becomes apparent.
d. Discussions of the numerous definitions of bandwidth becomes possible.
e. Effect of varying signal and carrier parameters (e.g., amplitude, data rate, and carrier frequency can be explored).

5.2. Week Two - a more sophisticated transmitter

By the end of the second week, this student had implemented:

a. Pulse shaping with variable roll-off factor (raised cosine).
b. Pulse shaping with variable roll-off factor (root raised cosine).

c. Pulse shaping with variable roll-off factor (Gaussian).
d. Random data sources from a number of sequence generators.
e. Quaternary phase shift keying (QPSK).
f. 8-ary PSK (8-PSK).
g. 16-ary PSK (16-PSK).
h. Explored the relationship between data rate, symbol rate, and transmitted bandwidth.

Possible uses for such a system include:

a. Exploring the relationship between roll-off factor and bandwidth.
b. Exploring the roll-off factor's effect on the transmitter's constellation diagram and trajectory diagram.
c. Exploring the differences between raised cosine, root raised cosine, and Gaussian pulses have on the transmitter's constellation diagram and trajectory diagram.
d. Exploring the relationships between data rate, symbol rate, and bandwidth.
e. Calculation of spectral efficiency.

5.3. Week Three - a basic receiver

By the end of the third week, this student had learned about matched filters, phase-locked loops, eye-patterns, symbol timing recovery, and had implemented a BPSK and QPSK receiver. The transmitter and receiver were both controlled from the same host computer after the student added a Gigabit Ethernet switch to the system. Possible uses for such a system include:

a. Importance of a properly matched filter.
b. Effect of transmitter or channel impairments on the receiver's performance. This could include I&Q gain imbalance, quad skew, error vector magnitude, SNR effects, inband interference, weak signals, very strong signals, or the effects of a multipath environment.
c. Bit error rate (BER plots) or bit error probability plots (BEP).

5.4. Week Four - a more complicated receiver and deeper development

By the end of the fourth week, this student had extended his work to 8-PSK and 16-PSK receivers and was exploring 16-QAM and 32-QAM. All of the signals were received over a wireless indoor channel. A channel equalizer was not used. Possible uses for such a system include:

a. Discussion of the relationship between the need for an increased signal SNR while increasing spectral efficiency.
b. The need for a channel equalizer to overcome channel impairments.
c. Effect of phase noise on system performance.
d. The possible introduction of error correcting codes and interleavers to help mitigate the effects of fading.

At this point in the course we decided to remove the reliance on the modulation toolkit and asked the student to create his own data source using a state machine (feedback shift register, maximal length sequence, or m-sequence generator) [22]. This student had significant course work in the digital systems/embedded systems area. It was believed that this previous course work would make the design and implementation of a finite state machine a straight forward effort. This task was accomplished in very little time. The student without prompting created an arbitrary order (up to order 16) linear feedback shift register system with reprogrammable feedback connections.

We believe that given the proper introduction this, and another motivated, student would be able to create the functionality of almost any modulation toolkit block. This powerful process of allowing very rapid student progress is not hindered by a lack of ability to peel back the programmatic layering to add significant depth to the laboratory exercise or the course in general.

5.5. Week Five - the continuing effort

With the vast majority of the topics associated with a first course in "Digital Communications" now completed, the student began to explore additional areas of personal interest. He is currently working on a system that will display a histogram of the received signals power levels. Possible uses for such a system include:

a. An opportunity to explore receiver architecture.
b. An opportunity to discuss and explore actual fading channel parameters compared to existing channel models.
c. Effect of the laboratory environment on the channel's fading parameters (e.g., movement, reflectors, directional antennae, etc …)

6. A FUTURE FULL OF POSSIBILITIES!

The LabVIEW/USRP combination presents an opportunity to enhance communications education by enabling a low cost, hands-on approach with live signals for realistic, real-world demonstrations, laboratory exercises, capstone design projects, and cutting-edge research. You are invited to participate in what we hope will become an ever increasing community of LabVIEW/USRP users.

7. CONCLUSIONS

The universal software radio peripheral (USRP) family of products has become a popular platform for hardware-based research and test bed validations conducted by universities in the SDR and CR fields. The recently version of LabVIEW, along with the USRP now offers a simpler, scalable and easier to use combined platform that will both broaden the accessibility of the technology and platform for hands-on applications and spur further adoption and use within university communication systems classrooms, teaching laboratories, and their follow-on coursework.

Given the rapid systems implementation that can be achieved using LabVIEW and the USRP2, coupled with the dramatic cost reductions that are possible when compared to more traditional SDR platforms, this is a timely and appropriate solution to a wide variety of education and research challenges.

8. REFERENCES

[1] T.B. Welch, C.H.G. Wright, and M.G. Morrow, "Software defined radio: Inexpensive hardware and software tools," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2934 – 2937, March 2010.

[2] C.R.A. Gonzalez, C.B. Dietrich, S. Sayed, H.I. Volos, J.D. Gaeddert, P.M. Robert, J.H. Reed, and F.E. Kragh, "Open-source SCA-based core framework and rapid development tools enable software-defined radio education and research," in *IEEE Communications Magazine*, Vol. 47, Issue 10, 2009, pp. 48 – 55.

[3] T.B. Welch, T. Kent, C.H.G. Wright, and M.G. Morrow, "An Affordable Software Defined Radio," in *Proceedings of IEEE DSP/SPE Workshop*, pp. 791 – 796, January 2009.

[4] S. Katz and J. Flynn, "Using software defined radio (SDR) to demonstrate concepts in communications and signal processing courses," in *Proceedings of IEEE Frontiers in Education Conference*, pp. 1 – 6, 2009.

[5] S. Guzelgoz and H. Arslan, "Modeling, simulation, testing, and measurements of wireless communication systems: A laboratory based approach," in *Proceedings of IEEE Wireless and Microwave Technology Conference*, pp. 1 – 5, 2009.

[6] L.S. Nagurney, "Software defined radio in the electrical and computer engineering curriculum," in *Proceedings of IEEE Frontiers in Education Conference*, pp. 1 – 6, 2009.

[7] C. H. G. Wright, T. B. Welch, M. G. Morrow, and G. Vinyard, "CommFSK: A hardware approach to teaching FSK," *ASEE Computer Education Journal*, vol. XVIII, pp. 38 – 45, April–June 2008.

[8] T. B. Welch, C. H. G. Wright, and M. G. Morrow, "Teaching rate conversion using hardware-based DSP," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. III, pp. 717 – 720, April 2007.

[9] T. B. Welch, C. H. G. Wright, and M. G. Morrow, "Caller ID: A project to reinforce an understanding of DSP based demodulation," *ASEE Computer Education Journal*, vol. XVI, pp. 2 – 7, October 2006.

[10] T. B. Welch, C. H. G. Wright, and M. G. Morrow, *Real-Time Digital Signal Processing: From MATLAB to C with the TMS320C6x DSK*, CRC Press, 2006.

[11] f. j. harris, *Multirate Signal Processing for Communication Systems*, Prentice Hall, 2004.

[12] J. K. Hwang, "Innovative communication design lab based on PC sound card and Matlab: a software-defined-radio OFDM modem example," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. III, pp. 761 – 764, 2003.

[13] J. H. Reed, *Software Radio: A Modern Approach to Radio Engineering*, Prentice Hall, 2002.

[14] http://www.ettus.com/products.

[15] http://www.ettus.com/WBX.

[16] http://www.ettus.com/order.

[17] L.W. Couch, II, *Digital and Analog Communication Systems*, Prentice Hall, 2007.

[18] http://www.ntia.doc.gov/osmhome/allochrt.pdf.

[19] http://gnuradio.org/trac.

[20] http://www.ni.com/labview/.

[21] http://www.ettus.com/downloads/ettus_broch_trifold_v7b.pdf.

[22] R.E. Ziemer, R.L. Peterson, and D.E. Borth, *Introduction to Spread Spectrum Communications*, Prentice Hall, 1995.