



Incorporating PlutoSDR in the Communication Laboratory and Classroom: Potential or Pitfall?

Dr. John "Ed" E. Post P.E., Embry-Riddle Aeronautical University

John. E. Post received the B.S. degree in electrical engineering from Texas Tech University in 1981, the M.S. degree in engineering from the University of Texas at Austin in 1991, and the Ph.D. degree in electrical engineering from Stanford University in 2005. He was commissioned a second lieutenant in the United States Army in December, 1981 and served on active duty until his retirement as a lieutenant colonel in June, 2006. His military service included two tours as an Instructor and later Assistant Professor with the Department of Electrical Engineering and Computer Science at the United States Military Academy, West Point, NY. His final military assignment was as Chief of Engineering with the Defense Threat Reduction Agency's Test Division at Kirtland AFB, NM. After retiring from the military, he joined the Computer, Electrical, and Software Engineering Department at Embry-Riddle Aeronautical University, Prescott, AZ, where he is an Associate Professor and currently serving as Chair. His research interests include design and optimization of planar microwave circuits and devices, optimizing the design of low-noise microwave amplifiers, and engineering education. Dr. Post is a senior member of IEEE. He is currently serving as the faculty advisor for the Embry-Riddle IEEE Student Chapter. He is also a Registered Professional Engineer of the Commonwealth of Virginia.

Dr. Dennis A. Silage, Temple University

Dennis Silage received the PhD in EE from the University of Pennsylvania. He is a Professor of Electrical and Computer Engineering at Temple University, teaches digital data communication, digital signal and image processing and embedded processing systems. Dr. Silage is past chair of the Electrical and Computer Engineering Division of ASEE, recipient of the 2007 ASEE National Outstanding Teaching Award and the 2011 ASEE ECE Division Meritorious Service Award. He is a Life Member of ASEE and a Life Senior Member of the IEEE.

Incorporating PlutoSDR in the Communication Laboratory and Classroom: Potential or Pitfall?

John E. Post

Computer, Electrical, and Software Engineering, Embry-Riddle Aeronautical University

Dennis Silage

Electrical and Computer Engineering, Temple University

Abstract

The reduced expense and growing capability of student owned equipment is fostering the open laboratory paradigm and revolutionizing the curriculum of many undergraduate analog and digital communications courses in electrical engineering¹. Among other possibilities, student owned portable equipment facilitates hands-on experiential learning and provides the opportunity to flip the laboratory to increase student engagement².

Up until now, this trend has had reduced impact in analog and digital communications because the most capable equipment, such as the Universal Software Radio Peripheral (USRP) was too expensive or inexpensive equipment, such as the ubiquitous RTL Software Defined Radio (RTL SDR) dongle, lacked the necessary features for full transceiver implementation¹. Currently retailing for \$99 (academic price), the Analog Devices Active Learning Module-Pluto Software Defined Radio (or PlutoSDR) appears to have the potential to bridge the gap between these two extremes³.

Here we will explore potential opportunities, benefits, and pitfalls to be avoided, of incorporating the PlutoSDR in the classroom and open laboratory environments. We begin by reviewing the hardware capability, limitations and setup requirements of the PlutoSDR. Next, example communication laboratories and demonstrations using PlutoSDR and GNU Radio and MATLAB/Simulink will be described. Finally, two semesters of student observations/comments on incorporating PlutoSDR into the student experience from Temple University are presented.

PlutoSDR

The Analog Devices Active Learning Module-Pluto Software Defined Radio or PlutoSDR is based on the Analog Devices AD9363 RF agile transceiver device^{3,4}. This device provides up to 20 MHz of tunable channel bandwidth between 325 MHz to 3.8 GHz, although it is possible to extend the frequency range and the bandwidth. It can transmit or receive 61.44 MSPS in full duplex using separate receive and transmit channels.

The PlutoSDR has a compact form-factor, is USB powered, and can be controlled by a variety of software packages such as MATLAB/Simulink or GNU Radio through the USB port using libio drivers supporting OS X®, Windows®, and Linux®. This allows users to exploit the capabilities of the PlutoSDR on a variety of host platforms. Additionally, custom Hardware Description Language (HDL) software may be loaded onto the Xilinx Zynq System-on-Chip device, as shown in Figure 1.

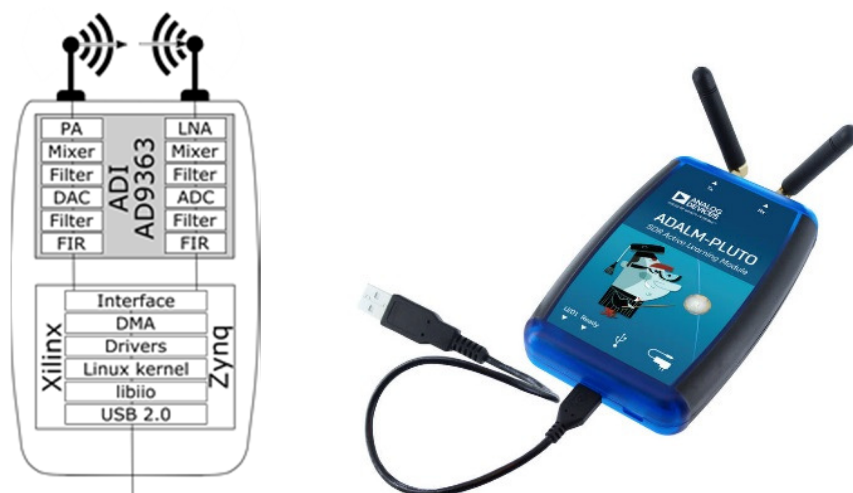


Figure 1. PlutoSDR (Courtesy of Analog Devices).

PlutoSDR Setup

Setting PlutoSDR up for Windows requires downloading and installing the latest drivers⁵, while in Linux the necessary modules are included by default in most modern distributions. Once the Windows drivers are installed *PlutoSDR* should appear as a removable mass storage device. Opening the device accesses the *info.html* page which provides additional quick start information. The PlutoSDR initially is configured to work with Windows applications such as MATLAB⁶, Simulink⁷, and IIO Oscilloscope⁸.

However, it is possible to open a serial connection to the PlutoSDR and issue commands that extends the tuning range from 70 MHz to 6 GHz and the bandwidth to 56 MHz using the AD9364 configuration, rather than the default AD9363 configuration⁹. Installation of the PlutoSDR with MATLAB/Simulink provides this change in the configuration.

IIO Oscilloscope

Analog Device's IIO Oscilloscope is an application that provides menu-based control of the PlutoSDR transmit and receive channels through selection of the RF Comms2/3/4 tab in the application⁸. This provides an opportunity to apply the PlutoSDR as an inexpensive spectrum analyzer to observe the properties of analog and digital modulation during lectures. Figure 2 shows the settings necessary to receive the 88 to 108 MHz broadcast stations in the FM band, while Figure 3 shows a typical power spectral display that results. Other demonstrations, such as reception of WiFi or GSM signals, are also possible.

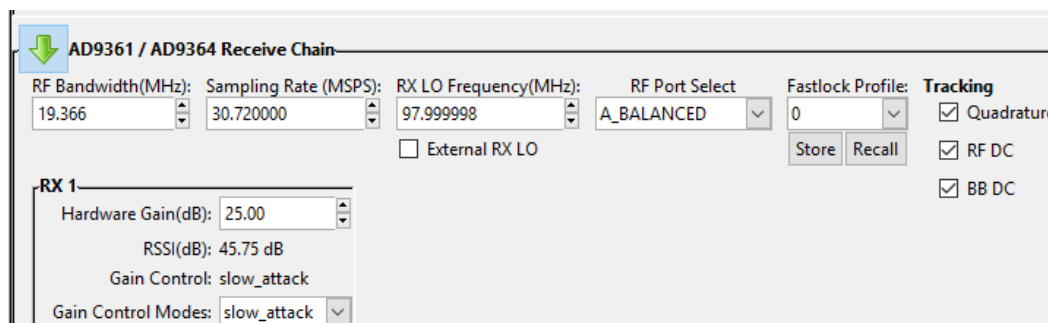


Figure 2. Receive portion of the IIO oscilloscope menu with LO frequency and RF bandwidth appropriate for reception of the FM broadcast band.

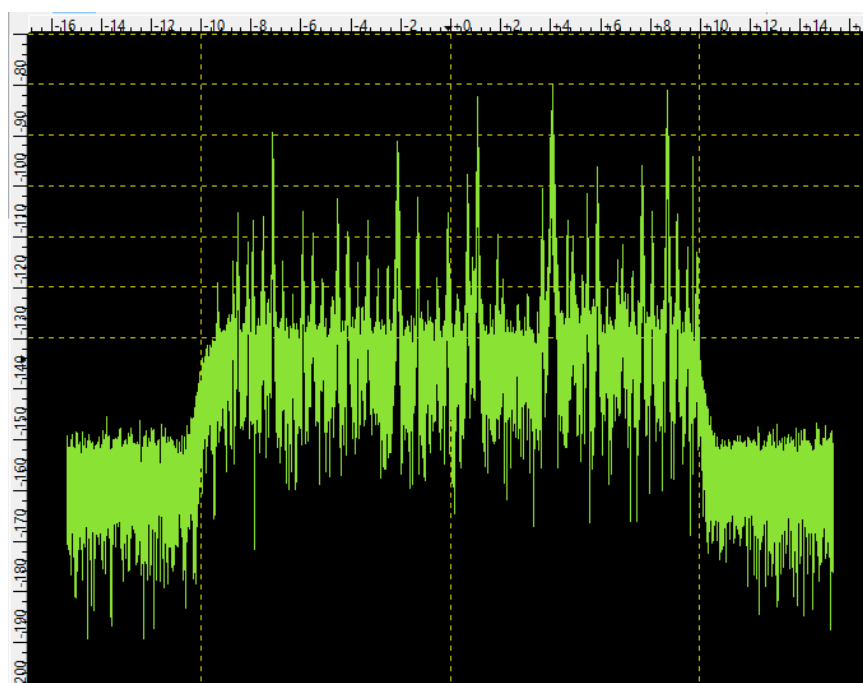


Figure 3. Power spectrum of the FM broadcast band as received by the PlutoSDR.

SDR#

The PlutoSDR is relatively new and only a few software defined radio programs support it. SDR# (or *SDR Sharp*) is a freeware software package that is widely used with inexpensive RTL SDR dongles but can also utilize the PlutoSDR¹⁰. Updating the PlutoSDR firmware¹¹ and installing the appropriate driver¹² permits the SDR# software to control the receiver and demodulate AM/FM/DSB/SSB/CW signals.

SDR# was implemented with a modular software architecture, and plug-ins from third-party developers are available which implement several useful functions that extend its capability¹³. Figure 4 shows the settings, spectrum, and waterfall display that result during reception of FM broadcast band stations using SDR#.

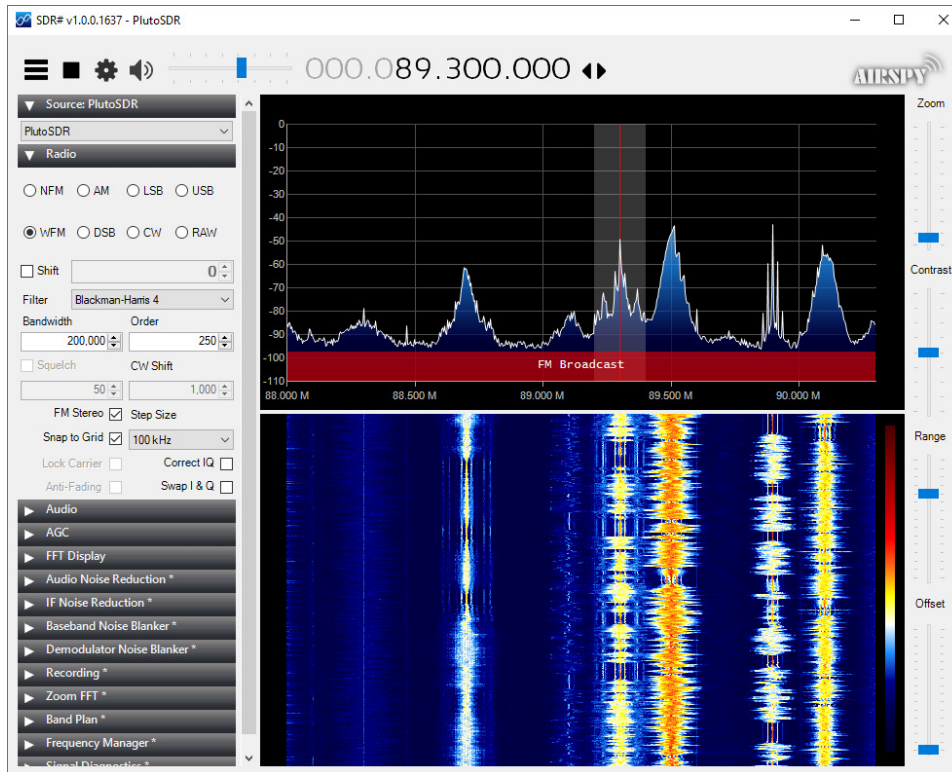


Figure 4. SDR# display during reception of FM broadcast band stations using PlutoSDR.

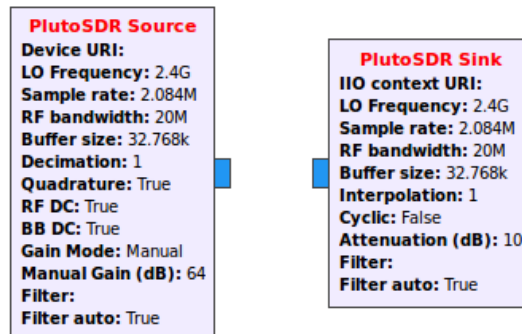


Figure 5. PlutoSDR source and sink modules from GNU radio.

GNU Radio Companion

GNU Radio Companion (GRC) is free, open-source, Linux-based software that contains the digital signal processing blocks necessary to implement software defined radio (SDR) functions¹⁴. A convenient way of using GRC on any PC is to use a bootable USB flash drive imaged with the *GNU Radio Live SDR Environment*¹⁵. After restarting the computer and booting from the USB drive it is possible to download and install the gr-iio blocks that contain the source and sink GNU radio modules that are compatible with the PlutoSDR¹⁶. Figure 5 depicts the PlutoSDR source and sink blocks that become available with the installation of the gr-iio blocks.

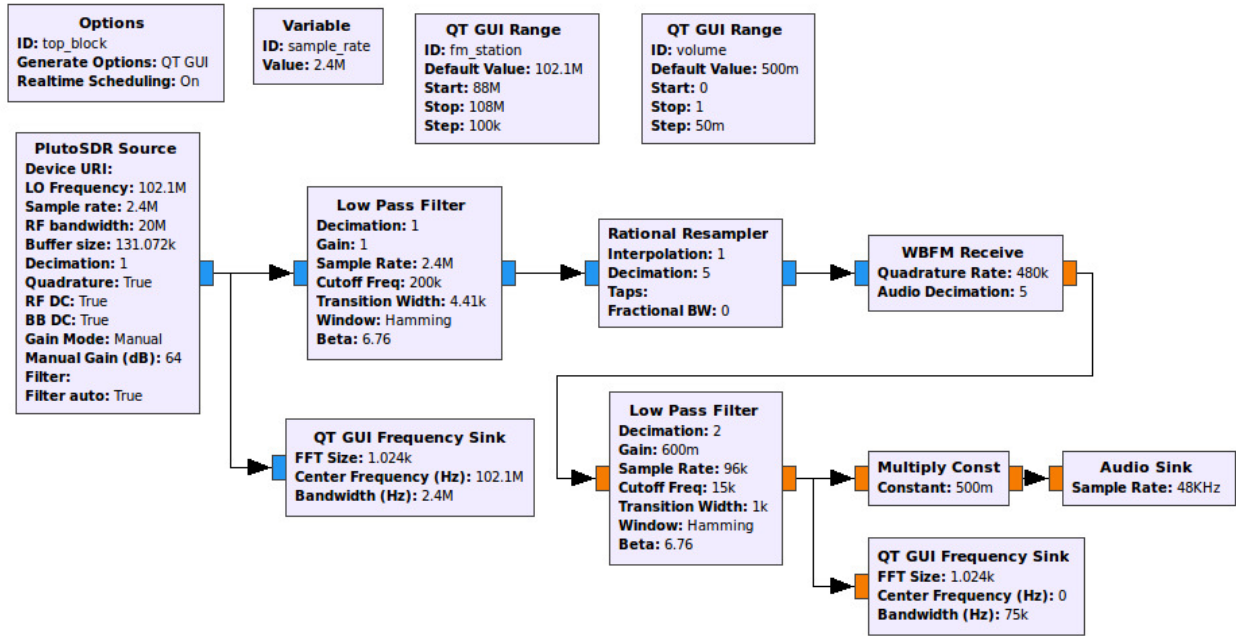


Figure 6. GRC flowgraph for FM broadcast band monophonic reception using PlutoSDR.

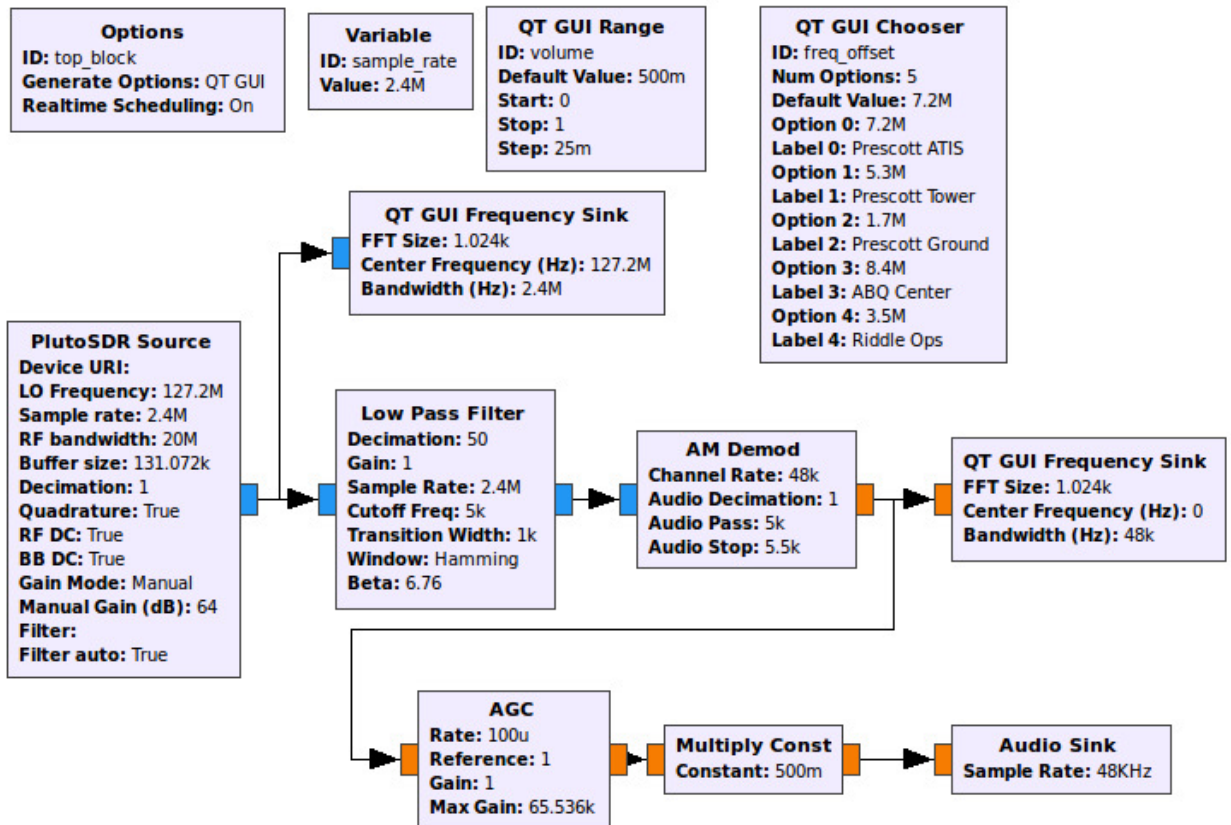


Figure 7. GRC flowgraph for VHF air band reception using PlutoSDR.

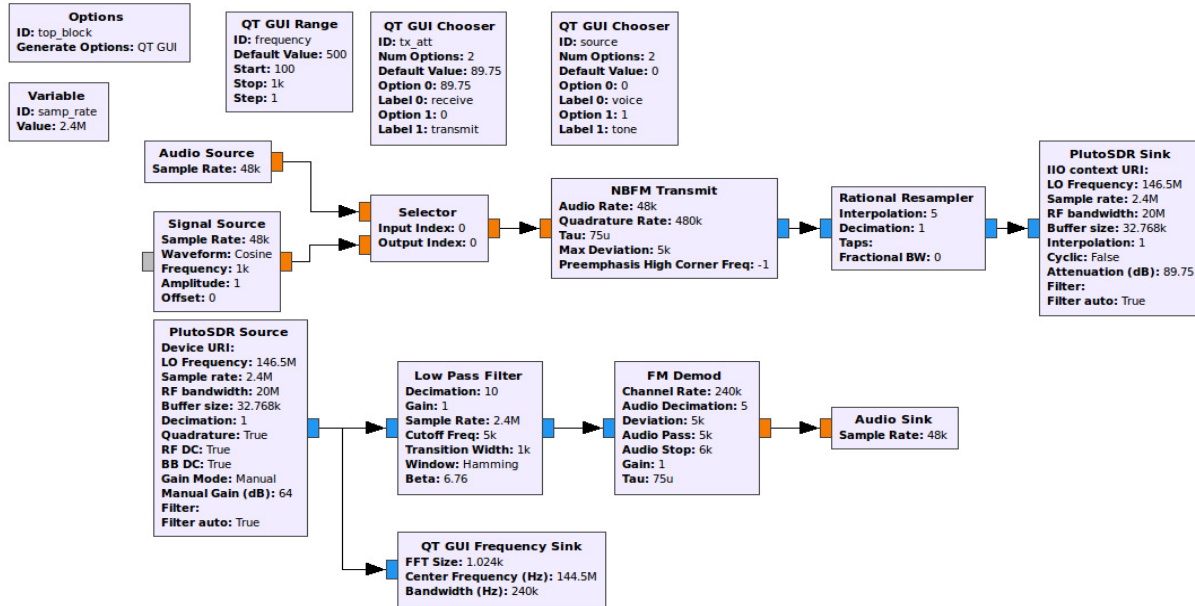


Figure 8. GRC flowgraph for a NBFM transceiver operating in the Amateur Radio 2-meter band.

GRC flowgraphs for two example receivers and an example transceiver are shown in Figures 6, 7 and 8. Figure 6 depicts the GRC flowgraph for monophonic reception of commercial FM broadcast signals. Figure 7 depicts the GRC flowgraph for a VHF air band receiver capable of receiving air traffic transmissions. Figure 8 depicts the GRC flowgraph for a narrowband FM transceiver operating at 146.52 MHz in the Amateur Radio 2-meter band at a frequency allocated for simplex voice communications.

It is important to note that the PlutoSDR sink block does not permit disabling the transmitter, but it is possible to attenuate the transmitted signal in 0.25 dB steps by up to 89.75 dB. This reduces the power of the transmitted signal to near the noise floor of the receiver. Additionally, the GSM antenna shipped with PlutoSDR permits limited reception of VHF signals. Performance is enhanced through use of an appropriate external antenna for each case.

MATLAB and Simulink

The *Support Package for the ADALM-PLUTO Radio*¹⁷ from The MathWorks¹⁷ provides not only MATLAB functions but also *Transmitter* and *Receiver* Simulink blocks for the PlutoSDR in the *Communication System Toolbox*, as shown in Figure 9. The *Transmitter* and *Receiver Simulink* blocks have parameter windows for their configuration, as shown in Figure 10.

Simulink has been used for the simulation of a digital communication systems in undergraduate education for over a decade^{18,19}. For simplicity in a first course in digital communications, these Simulink simulations feature correlated transmitters and receiver for carrier frequency and phase and symbol timing. However, the PlutoSDR provides an opportunity to investigate carrier frequency and phase synchronization and symbol timing recovery both in simulation and implementation.



Figure 9. Communications System Toolbox for the PlutoSDR in Simulink.

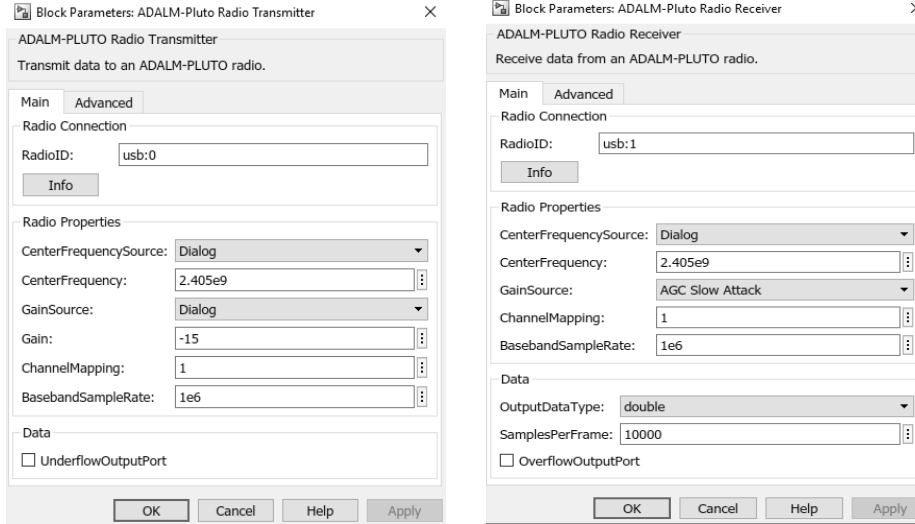


Figure 10. *Transmitter* and *Receiver* PlutoSDR *Simulink* block parameter windows.

For binary phase shift keying (BPSK) the data source in Figure 9 provides random data in the range $[-1, 1]$ at 1 kb/sec which generates a modulated BPSK baseband signal. The modulated baseband signal is then *up converted* by the PlutoSDR *Transmitter* block to the carrier frequency f_c of 2.405 GHz for transmission. The received carrier modulated BPSK signal is *down converted* by the PlutoSDR *Receiver* block. A *Spectrum Analyzer* block from the *Sinks, DSP System Toolbox* displays the resulting power density spectrum (PSD) of the output²⁰.

The observed BPSK PSD is a double-sided spectrum since the PlutoSDR *Receiver* block outputs complex numbers. The double-sided PSD of a BPSK signal (PSD_{BPSK}) with bit time T_b or a data rate $r_b = 1/T_b$, peak amplitude A volts, carrier frequency f_c Hz and a load $R_L \Omega$ is given by Equation 1¹⁹. The *sinc* ($\sin x/x$) term predicts that there are periodic *nulls* in the PSD_{BPSK} at multiples of the binary data rate r_b .

$$PSD_{BPSK}(f) = \frac{1}{4R_L} \left[A^2 T_b \operatorname{sinc}^2 \left(2\pi(T_b/2)(f - f_c) \right) + A^2 T_b \operatorname{sinc}^2 \left(2\pi(T_b/2)(f + f_c) \right) \right] \quad (1)$$

A *Simulink* simulation of PSD_{BPSK} results in the single sided PSD shown in Figure 11 (top)¹⁹. The carrier frequency $f_c = 20$ kHz and the data rate $r_b = 1$ kb/sec here. The single sided PSD_{BPSK} for the *Simulink* simulation is centered at 20 kHz and has periodic nulls, as predicted by Equation 1, every ± 1 kHz centered at 20 kHz.

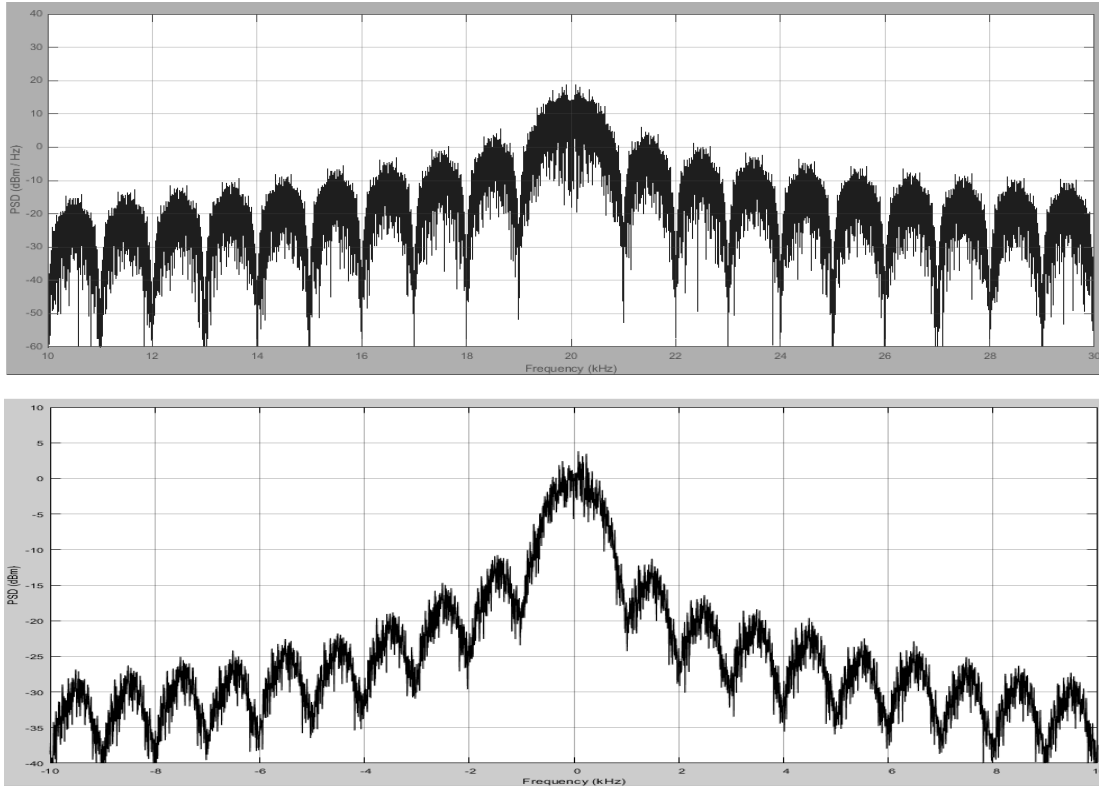


Figure 11. PSD_{BPSK} of the *Simulink* simulation (top) and the PlutoSDR transmitter (bottom) both with $r_b = 1$ kb/sec.

The PSD_{BPSK} of the PlutoSDR transmitter is a double-sided power spectral density centered at 0 Hz or baseband because of the frequency translation from the carrier frequency f_c by the PlutoSDR receiver, as shown in Figure 11 (bottom). Since the data rate r_b is also 1 kb/sec, the PSD_{BPSK} of the PlutoSDR transmitter shows periodic nulls every ± 1 kHz center at 0 Hz.

It's All in How You Do It

Digital communication systems have been taught in the undergraduate curriculum with the approach of integrating an analytical solution with Simulink simulation for over a decade. Course materials have been developed to support the requisite standard text of the curriculum. However, the digital communication laboratory using simulation and the PlutoSDR allows the exploration of topics whose results are more experiential¹⁸. The incalculable value for the undergraduate student seems to be the experience provided by the *what-if* of the results.

However, the pitfalls to be avoided include not to use only a simulation or only an SDR implementation but to couple them together. Another pitfall is that the SDR incorporates additional concerns not seen in a correlated transmitter and receiver simulation. Carrier frequency and phase synchronization and symbol timing recovery are topics usually not covered in the first course in digital communication systems. Finally, the SDR does not provide investigations of baseband signaling.

Recently the students have had access to the PlutoSDR in a flipped classroom and an open laboratory. The NI 2920 USRP with Simulink was used previously in a standard laboratory but required access to a limited number of a second Ethernet interface equipped PC station. The PlutoSDR with its USB interface is amenable to the open laboratory concept and more challenging investigations. These include utilizing a second PlutoSDR to evaluate carrier frequency drift and compensation, as shown in Figure 12.

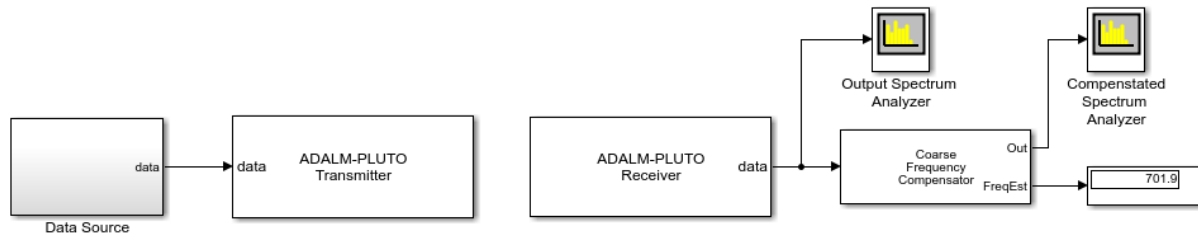


Figure 12. PlutoSDR carrier frequency compensation in Simulink.

The undergraduate students relate that they are appreciative of the opportunity to now have an open laboratory in digital communication systems using the PlutoSDR. As a gauge of the excitement generated, a group of students, sponsored by the IEEE Student Chapter, have even organized a license-free *wireless technology club* as an adjunct to the Temple University Amateur Radio Club www.temple.edu/k3tu. They are using both the PlutoSDR and the RTL-SDR dongle for the reception of Amateur Radio satellite (AMSAT)²¹ telemetry and decoding of the Automatic Packet Reporting System (APRS)²² and the Aircraft Communications and Reporting System (ACARS)²³.

But Does It Work?

The assessment of the coupling of the traditional analytical approach with an SDR equipped laboratory, using either the USRP or the PlutoSDR, has been obtained by interviews of alumni who remain in the area and are engaged in digital communication design and application. These professionals have found their experience with a Simulink simulation and the implementation with an SDR of a digital communication system has facilitated their understanding and transition to design. The feedback obtained from the alumni has been used to improve the presentation of the concepts in the first digital communication course and to further improve a second undergraduate course in telecommunication engineering.

Indirect course feedback surveys have also been used to gauge the response of the undergraduate student to the utilization of either the USRP or the PlutoSDR as a supplement to Simulink simulation in the digital communication course for two semesters. The inexpensive PlutoSDR, as an alternative to the USRP, has now also facilitated an *open laboratory* and the partial survey results, as shown in Table 1, confirm the utility of this approach.

The suggested responses were numbered from 1 - Strongly Disagree, 2 - Disagree, 3 – Neutral, 4 – Agree and 5 – Strongly Agree.

I found that the use of the USRP and PlutoSDR for laboratory projects improved my understanding of digital communication	4.8
The USRP in the regular laboratory provided a better environment for projects in digital communications	2.1
Having used both the USRP and the PlutoSDR in an open laboratory, I prefer the PlutoSDR	4.4
An open laboratory in digital communications is in accordance with the other open laboratories in ECE and should be continued	4.7

Table 1. Partial course survey on the utilization of the USRP and the PlutoSDR in the digital communications course.

Conclusion and Future Work

The availability of the inexpensive PlutoSDR and the various software environments has now facilitated the flipped classroom in digital communication. It has put this course on par with courses in digital logic, processor systems, circuits and electronics where such inexpensive hardware is currently in use in the flipped classroom and the open laboratory. A new text will be available to incorporate the PlutoSDR with Simulink simulations in bandpass digital communications²⁰.

References

1. Kurt VonEhr, et al., *Software Defined Radio: Choosing the Right System for Your Communications Course*, Proceedings of the 123rd ASEE Annual Conference and Exposition, New Orleans, LA. June 2016.
2. G. J. Kim, et al., *Lessons Learned from Two Years of Flipping Circuits I*, Proceedings of the 122nd Annual ASEE Annual Conference and Exposition, Seattle, WA. June 2015.
3. ADALM-PLUTO – Product Overview
<http://www.analog.com/media/en/news-marketing-collateral/product-highlight/ADALM-PLUTO-Product-Highlight.pdf>
 Accessed: 2018-01-19
4. ADALM-PLUTO – Analog Devices
<http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/ADALM-PLUTO.html#eb-overview>
 Accessed: 2018-01-19
5. ADALM-PLUTO – Windows Drivers
<https://wiki.analog.com/university/tools/pluto/drivers/windows>

Accessed: 2018-01-19

6. MATLAB – Mathematical software package
<https://www.mathworks.com/products/matlab.html>

Accessed: 2018-01-19

7. Simulink – Simulation and model-based design
<https://www.mathworks.com/products/simulink.html>

Accessed: 2018-01-19

8. IIO Oscilloscope – Visualization software
https://wiki.analog.com/resources/tools-software/linux-software/iio_oscilloscope

Accessed: 2018-01-19

9. AD 9364 update
<https://wiki.analog.com/university/tools/pluto/users/customizing>

Accessed: 2018-01-23

10. SDR# software
<https://www.rtl-sdr.com/rtl-sdr-quick-start-guide/>

Accessed: 2018-01-24

11. PlutoSDR Firmware Update
<https://wiki.analog.com/university/tools/pluto/users/firmware>

Accessed: 2018-01-24

12. PlutoSDR driver for SDR#
<https://github.com/Manawyrm/sdrsharp-plutosdr/blob/master/README.md>

Accessed: 2018-01-23

13. SDR# Plug-ins
<https://www.rtl-sdr.com/sdrsharp-plugins/>

Accessed: 2018-01-24

14. GNU Radio
https://wiki.gnuradio.org/index.php/Main_Page

Accessed: 2018-01-24

15. GNU Radio Image
http://www.aaronscher.com/wireless_com_SDR/Create_Live_USB_GNU_Radio.html

Accessed: 2018-01-24

16. PlutoSDR Source/Sink \Blocks
<https://wiki.analog.com/resources/tools-software/linux-software/gnuradio>

Accessed: 2018-01-24

17. PlutoSDR Communications Systems Toolbox Support
<https://www.mathworks.com/hardware-support/adalm-pluto-radio.html>

Accessed 2018-01-26

18. Dennis Silage, *Teaching Digital Communications in a Wireless World – Who Needs Equations?* Proceedings of the 113th Annual ASEE Annual Conference and Exposition, Chicago, IL. June 2006.

19. Dennis Silage, *Digital Communication Systems Using MATLAB and Simulink* (Second Edition), Bookstand Publishing, 2016.

20. Dennis Silage, *Digital Communication Systems Using MATLAB and Simulink and the ADALM-Pluto SDR*, Bookstand Publishing, 2018.

21. Amateur Radio Satellite (AMSAT)

<https://www.amsat.org/>

Accessed 2018-01-26

22. Automatic Packet Reporting System

<http://www.aprs.org/>

Accessed 2018-01-26

23. Aircraft Communications, Addressing and Reporting System

[https://www.skybrary.aero/index.php/Aircraft Communications, Addressing and Reporting System](https://www.skybrary.aero/index.php/Aircraft_Communications,_Addressing_and_Reporting_System)

Accessed 2018-01-26