LAB 1

Introduction to the USRP

1.1 Objective

The purpose of this introductory laboratory exercise is to ensure that students have a working installation of LabVIEW Communications on their computers and know how to connect to the USRP software defined radio.

1.2 Background

The Wireless Innovation Forum defines Software Defined Radio (SDR) as:

"Radio in which some or all of the physical layer functions are software defined." [1]

SDR refers to the technology wherein software modules running on a generic hardware platform are used to implement radio functions. By combining the NI USRP hardware with LabVIEW software you can create a flexible and functional SDR platform for rapid prototyping of wireless signals including physical layer design, record and playback, signal intelligence, algorithm validation, and more.

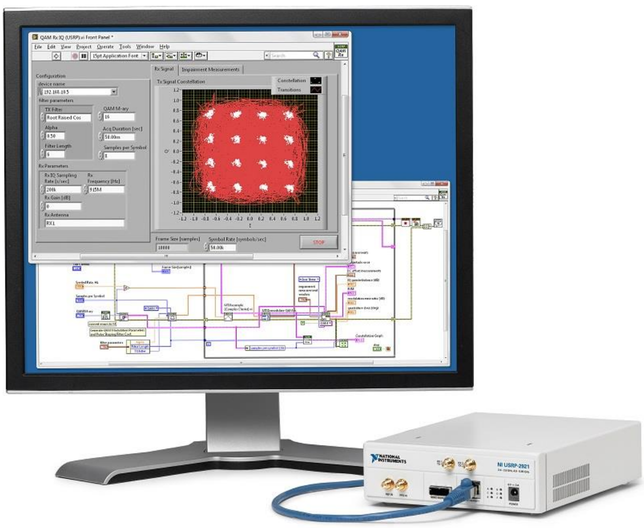


Figure 1. Hardware Setup in a Wireless Communications Lab

NI USRP Hardware

The NI USRP connects to a host PC creating a software defined radio. Incoming signals at the SMA connector inputs are mixed down using a direct-conversion receiver to baseband I/Q components, which are sampled by a analog-to-digital converter (ADC). The digitized I/Q data follows parallel

[1] http://www.wirelessinnovation.org/what_is_sdr

paths through a digital downconversion (DDC) process that mixes, filters, and decimates the input signal to a user-specified rate. The downconverted samples are passed to the host computer.

For transmission, baseband I/Q signal samples are synthesized by the host computer and fed to the USRP at a specified sample rate over Ethernet, USB or PCI express. The USRP hardware interpolates the incoming signal to a higher sampling rate using a digital upconversion (DUC) process and then converts the signal to analog with a digital-to-analog converter (DAC). The resulting analog signal is then mixed up to the specified carrier frequency.

More information about NI SDR hardware can be found in the respective Getting Started Guide[2] available in the start menu.

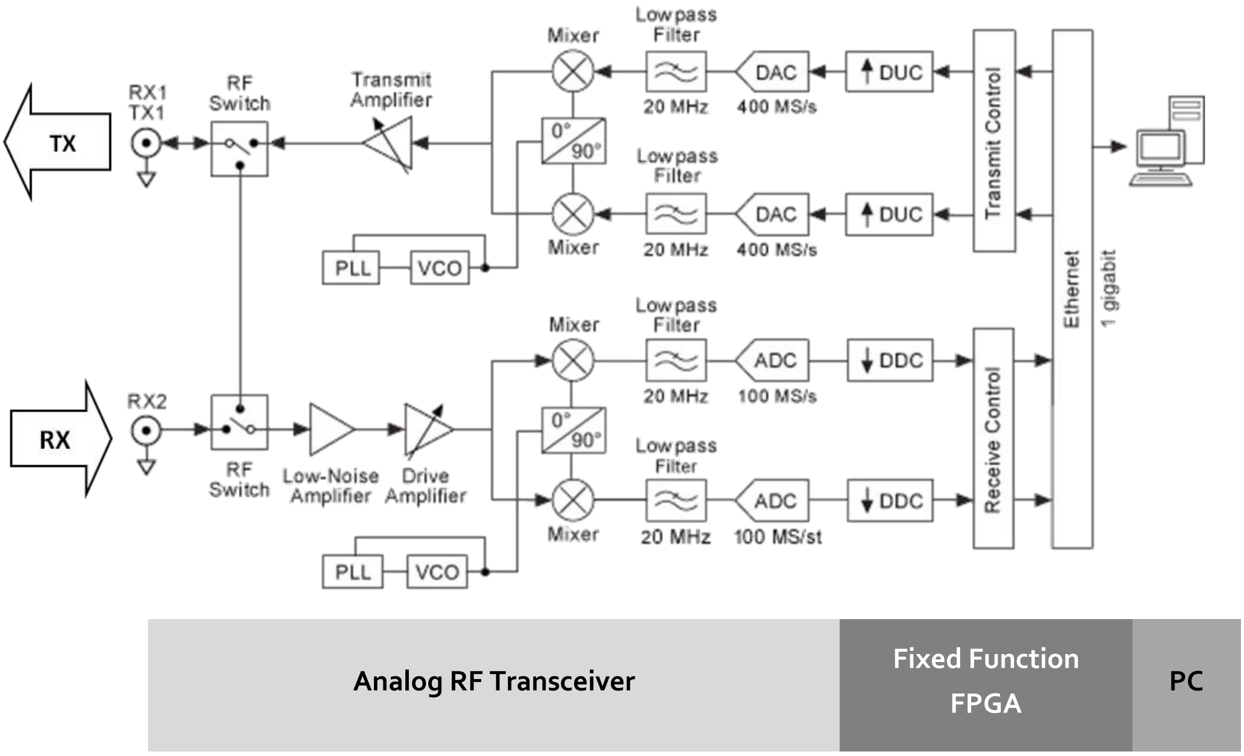


Figure 2. Typical Block Diagram of an NI USRP

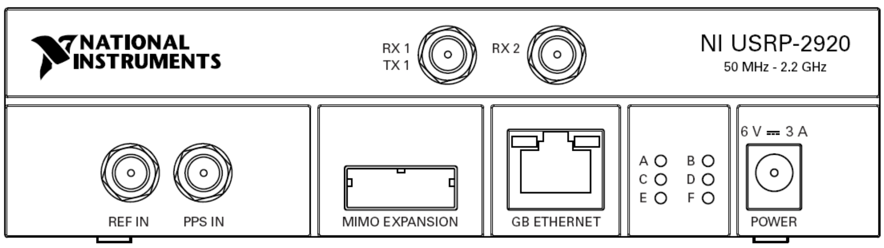


Figure 3. Front View of an NI USRP-2920 Software Defined Radio

---

[2] Available from the Start Menu→All Programs→National Instruments→NI-USRP→Documentation

NI LabVIEW Communications System Design Software

LabVIEW is a graphical programming language developed by National Instruments. The basic building block of LabVIEW is the virtual instrument (VI). Conceptually, a VI is analogous to a procedure or function in conventional programming languages. Each VI consists of a *block diagram* and a *front panel*. The block diagram describes the functionality of the VI, while the front panel is a top level interface to the VI. The construct of the VI provides two important virtues of LabVIEW: code reuse and modularity. The graphical nature of LabVIEW provides another virtue: it allows developers to easily visualize the flow of data in their designs. NI calls this Graphical System Design. Also, since LabVIEW is a mature data flow programming language, it has a wealth of existing documentation, toolkits, and examples which can be leveraged in development.

In this course you will use National Instruments SDR hardware. LabVIEW provides a simple interface for configuring and operating various external I/O, including the NI SDR hardware used in lab. This is the main reason why you will use LabVIEW as the programming language to build an SDR in this course. You should realize that the algorithms considered here could also be programmed in optimized C/C++, assembly, or VHDL and implemented on a DSP, microcontroller, or an FPGA. The choice of hardware and software in this lab is mostly a matter of convenience.

In future labs you will need to be familiar with LabVIEW and the documentation/help available to you. This is the only lab in this course which will give you the opportunity to learn and practice LabVIEW programming; so it is important that you take this opportunity to ask the instructor any questions you might have about LabVIEW programming. The following tutorials and reference material will help guide you through the process of learning LabVIEW:

- LabVIEW Communications System Design Suite 2.0 Online Manual
- LabVIEW Communications Guided Help tutorials
- Context help

The online manual can be found through our website at "www.ni.com/manuals". The context help window displays basic information about LabVIEW objects when you move the cursor over each object. To toggle the display of the context help window, select View » Context Help or press <Ctrl-H>.

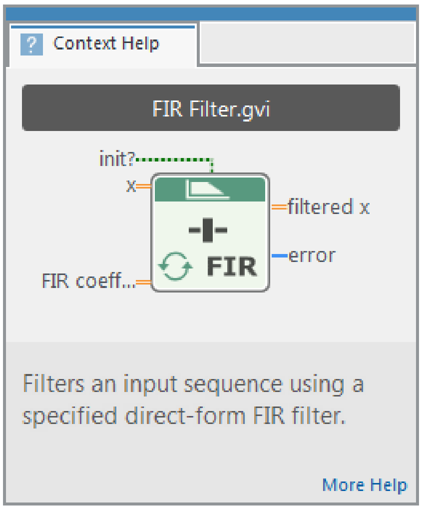


Figure 3. Context Help Window

The LabVIEW online help is the best source of detailed information about specific features and functions in LabVIEW. Online help entries break down topics into a concepts section with detailed descriptions and a how-to section with step-by-step instructions for using LabVIEW functions.

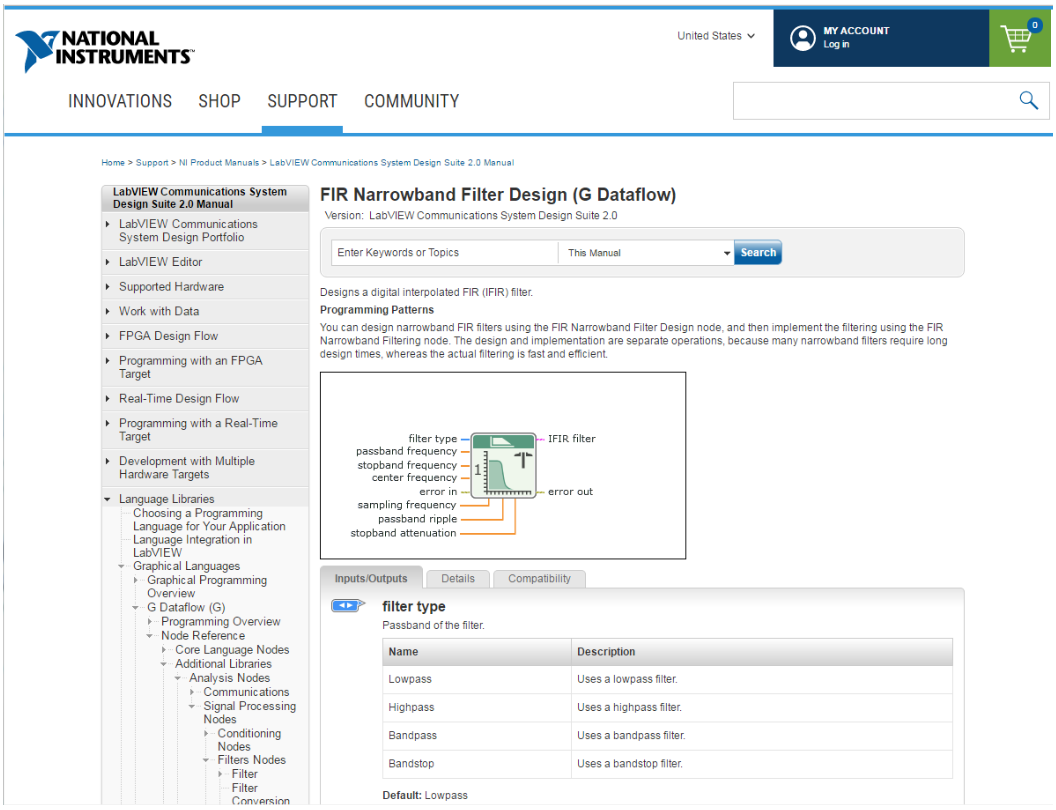


Figure 4. Screenshot of LabVIEW Online Help

1.3 Pre-Lab

In order to complete labs 2 - 12, you will need to download the corresponding LabVIEW program files. Download these files from http://www.ni.com/white-paper/52344/en/ and unzip the files to a convenient location. You will access them as instructed by each lab.

Instructors, please visit www.ntspress.com/publications/usrp-labs/ for more information.

1. Ensure LabVIEW Communications is installed
2. Open LabVIEW Communications
3. From the main window (also called the lobby), navigate to the LEARNING tab at the upper right hand corner. Open and complete the guided help tutorial from Learn→Getting Started→Introduction to the LabVIEW Editor
4. To return to the lobby, select File→Close All
5. Complete 5 other guided help tutorials to learn more about LabVIEW Communications. From the lobby click Learn→Programming Basics to find the following:
   - Designing a User Interface
   - Debugging your VI
   - Basic Data Types
   - Arrays
   - While Loops

Bring any questions or concerns regarding LabVIEW or these tutorials to your instructor's attention. For the remainder of this lab you should be familiar with the basics of LabVIEW programming and where to look for help.

1.4 Lab Procedure

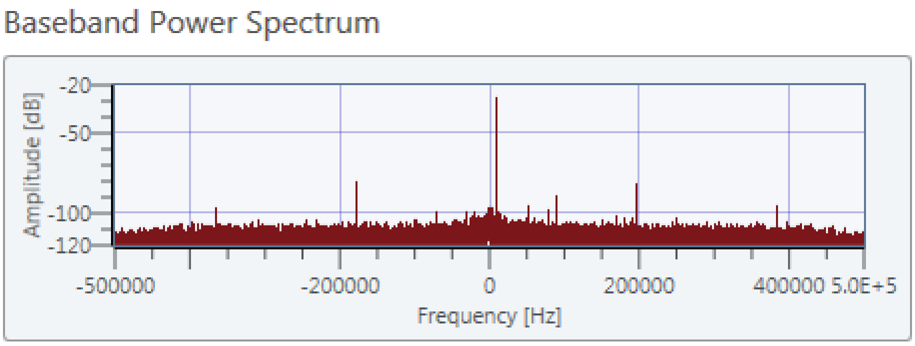
1. Connect the TX1 output to the RX2 SMA connector using a loopback cable and 30 dB attenuator provided.
2. Connect the USRP software defined radio to the computer as described in the Getting Started Guide[3] for your NI USRP transceiver.
3. Launch the NI USRP Configuration Utility[4] to find the *Device Name* for your NI USRP device.
4. From the lobby in LabVIEW Communications, open the following NI USRP example:

   Examples→Hardware Input and Output→NI USRP Host→Single Device→Single Channel→Continuous→RX Continuous Async

5. Give the example a project name and click Create.
6. From the example, open another NI USRP example:

   Help→Examples→Hardware Input and Output→NI USRP Host→Single Device→Single Channel→Continuous→TX Continuous Async

7. Give the example a project name and click Create.
8. On the Tx Continuous Aysnc example (referred to as the transmitter program), enter the device name you found using the USRP configuration utility and note the value of the *tone frequency* control. This program generates a single frequency tone at baseband and sends it to the USRP.
9. Run the transmitter program.
10. On the Rx Continuous Async example (referred to as the receiver program) example window, enter the same device name as the transmitter and change the *Active Antenna* to RX2**.**
11. Run the receiver program and analyze the Baseband Power Spectrum Graph. You should see a spike near the center of the graph. This is the single tone that was generated by the transmitter.



12. Without changing the value of the *Carrier Frequency* control on the receiver or transmitter program, "move" the location of the single tone on the Baseband Power Spectrum graph to 150 kHz.

---

[3] Available from the Start Menu→All Programs→National Instruments→NI-USRP→Documentation
[4] Available from the Start Menu→All Programs→National Instruments→NI-USRP

Note: Changes made to the program do not apply while it is running. You should stop the program and start it for change to take effect.

Questions

1. On the *Diagram* of the receiver program, write the name of each node (sometimes called function or block) and its purpose on the block diagram. (Such as niUSRP Open Rx Session creates a session handle to the device for other functions).
2. On the *Diagram* of the transmitter program, write the name of each node and its function on the block diagram. (Such as niUSRP Open Tx Session creates a session handle to the device for other functions).
3. Describe how you changed the "spike" on the Baseband Power Spectrum graph to 150 KHz.

1.5 Report

Submit all of the answers to the *Questions* section above.