Software Defined Radio

Project Report

Prepared by
Hamed Noori
Radiowize Consulting Ltd.

July 2017

Preface

The AURORA (AUtomotive testbed for Reconfigurable and Optimized Radio Access) testbed are currently being developed at University of British Columbia, Department of Electrical and Computer Engineering. Radiowize Consulting, Ltd. is providing specification, configuration and testing services in support of the installation and commissioning of the AURORA Connected Vehicle Testbed.

The work was conducted by Hamed Noori. He holds a master's degree related to intelligent transportation and has experience with connected vehicle technology. The work was overseen by Karl Reardon, PEng, principal of Radiowize Consulting and an experienced telecom engineer. In addition, Prof. David G. Michelson at UBC Radio Science Lab, provided considerable technical inputs and advice.

Executive Summary

Software Defined Radios (SDR) is defined as a radio communication system where most or all components of the communication system that typically are implemented in the hardware such as mixers, modulators, filters, etc. are developed in a computer or embedded system as mean of software. SDR make it very easy for researcher to develop and prototype the system and analysis and evaluate different parameters in the communication system.

In order to take advantage of SDR in AURORA testbed, USRPs are going to be installed at roadside units which makes the AURORA one of the unique connected vehicles testbed that allows researcher to study and perform different technologies using SDR.

Installation of the SDR at the AURORA testbed enables numerous possibility for performing research and development in the novel communication in the connected vehicle fields, such as: Channel modeling study, Study on new MAC layers, Research on the novel networking platform, etc.

Radiowize Consulting is implementing the software development environment for SDR at AURORA and demonstrating USRP and SDR functionality by presenting one or two applications such as emulation of IEEE 802.11 or IEEE 802.11p. These applications can be used as a basis for future research.

Radiowize Consulting configured workstation to be used in development of software for the National Instrument USRP Software Defined Radios. This report provides details information about how to setup the software development environment and also demonstrating sample application of using SDR to emulate IEEE 802.11p using NI and Ettus USRP.

Table of Contents

Table of Figures

1. Introduction

Radiowize Consulting will configure workstation to be used in development of software for the National Instrument and Ettus USRP Software Defined Radios to be installed as part of the five enhanced roadside units and connected test vehicles. Radiowize Consulting will make a best effort to demonstrate the function and operation of the USRPs by installing, configuring and demonstrating the software required to emulate IEEE 802.11 or IEEE 802.11p wireless device.

1.1.Objectives

Software Defined Radios (SDR) is defined as a radio communication system where most or all components of the communication system that typically are implemented in the hardware such as mixers, modulators, filters, etc. are developed in a computer or embedded system as mean of software. SDR make it very easy for researcher to develop and prototype the system and analysis and evaluate different parameters in the communication system.

In order to take advantage of SDR in AURORA testbed, USRPs are going to install at roadside units which makes the AURORA one of the unique connected vehicles testbed that allow researcher to study and perform different technologies using SDR.

Installation of the SDR at the AURORA testbed enables numerous possibility for performing research and development in the novel communication in the connected vehicle fields, such as:

- Channel modeling study.
- Study on new MAC layers
- Research on the novel networking platform.

In the first stage of the testbed and in this work, the goals are to setup the USRP and SDR system to allow future researcher can easily use these facilities. In addition, in order to illustrates the huge possibilities of the SDRs, one or two applications will be developed (e.g. IEEE 802.11p Emulators using USRPs). These applications can be used as a starting point for future research.

The rest of this section will explain the steps and tasks in more details, which includes finished and in progress tasks to achieve the objectives of the project. Also, the deliverable reports will be described.

1.2. Primary Study on Products

Radiowize Consulting study on, and develop software for the using the National Instruments USRP. The primary study will be done to determine the suitable software development platform for USRPs. The study will be done to understand the requirement for software development with the USRP and to determine the configuration procedures. The end goal will be find the best approach to develop software to be able to emulate the IEEE 802.11 or IEEE 802.11p.

1.3.Configuration and Software Development Environment

After understanding the USRP and the requirement for the software development and configuration, Radiowize will implement the suitable software development environment for the USRP. Also, configuration for the equipment will be perform.

1.4.Software Development

After USRP configurations and implementing the software development environment, certain applications will be proposed for implementation. After consulting with AURORA committee, some application will be selected and the require software for running them at the testbed will be developed.

1.5.Deliverable: SDR Report

Detailed report will be provided upon finishing the project for configuration procedures, step-by-step guidelines to use the developed software for USRPs. Also report will cover steps for data generation/collection, software development environment, and procedures and requirement for improving the applications.

2. Methodology

2.1. USRP Hardware Driver:

The main goal is to implement a platform based on Software Defined Radios (SDR) and USRP that capable of performing as a Wi-Fi card and/or a DSRC device. This section provides information related to software development.

USRP family of Software Defined Radios (SDRs) are devices that allow users to transmit and receive many different and custom waveforms at various frequencies. Ettus research provided the USRP hardware driver (UHD) as device driver for USRP products family which support Window, MacOS and Linux systems and several different software and frameworks. In addition to these framework, C++ based UHD API can get access directly to the functionality of the UHD.

UHD is a user-space library that is able to run on a general-purpose processor (GPP) and it provides the capability of controlling the data transmission and transport user waveform samples to and from USRP hardware. In addition, UHD is responsible for controlling the parameters such as gains, center frequency, sampling rate, etc.

There are several frameworks that can be used as a software development tool for USRP using UHD, such as GNU Radio, NI LabVIEW, MATLAB and Simulink. In this work, GNURadio software framework has been used. *Figure 1* present the architecture of the using UHD and USRP.

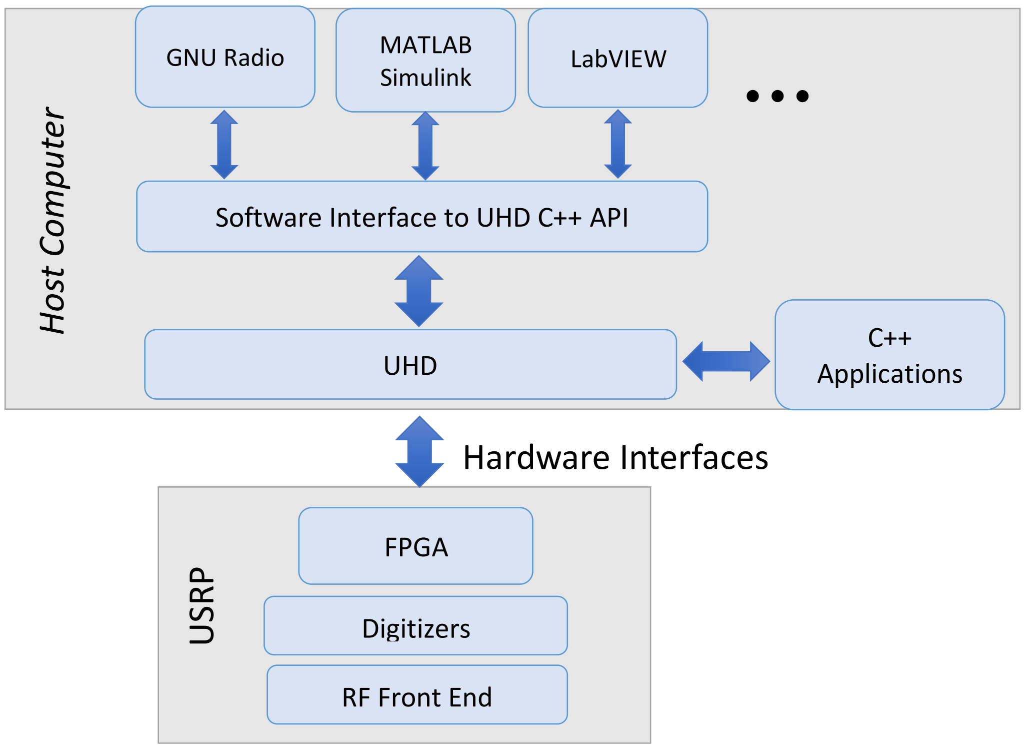


Figure 1. Structure of Implementing the USRP UHD

2.2. Implementation of IEEE 802.11 at GNU Radio

GNU Radio is a free open source toolkit for developing software defined radios and it has an easy to use application programming environment while it optimized DSP performance by combining C++ and Python. It also includes a graphical programming environment called GNURadio Companion. GNU Radio is providing different signal processing blocks to implement software radios and it is capable of communicating with USRPs and low-cost external RF hardware to create software-defined radios. In summary, GNU Radio is responsible for doing all signal processing required for SDRs.

The GNU Radio software platform structured is based on different layers where each layer is responsible for a performing particular function and it could use different programming languages. There are different implementation of standards/protocols/applications on GNU Radio and they can be used as a based framework to build and prototype new SDRs. In this work, the implementation of IEEE 802.11 transceiver by Bastian Bloessl [1] is selected as a based framework in GNURadio. It is an open source software and code repository which support different modulations, coding schemes and it has implemented different channel estimations algorithms.

3. Software Installation

This section provides the procedures of implementation of IEEE 802.11 at GNURadio and running the scenarios with USRP. In order to achieve this goal, first USRP hardware driver should be installed. Then, the GNURadio and it dependencies need to be install in the host computer and finally installation of the implemented IEEE 802.11 can be installed and run using GNURadio and USRP. GNURadio and UHD installation procedures has been done in Ubuntu 16.04 LTS machine and building and installing is performed using the source code.

3.1. UHD Installation

Before installing USRP Hardware Driver, several dependencies need to be built. This section provides information about how to install the UHD.

a) C++ Compiler Installation

The first step is to install the C++ compiler. In this work, the GCC, GNU Compiler Collection has been used. The GCC includes front ends for C, C++, Objective-C, etc. and it includes required libraries for these languages. GCC was originally written as the compiler for the GNU operating system. Following codes is used to install the GCC and process is shown in *Figure 2*.

```
sudo apt-get install gcc
sudo apt-get install gcc-4.8
```

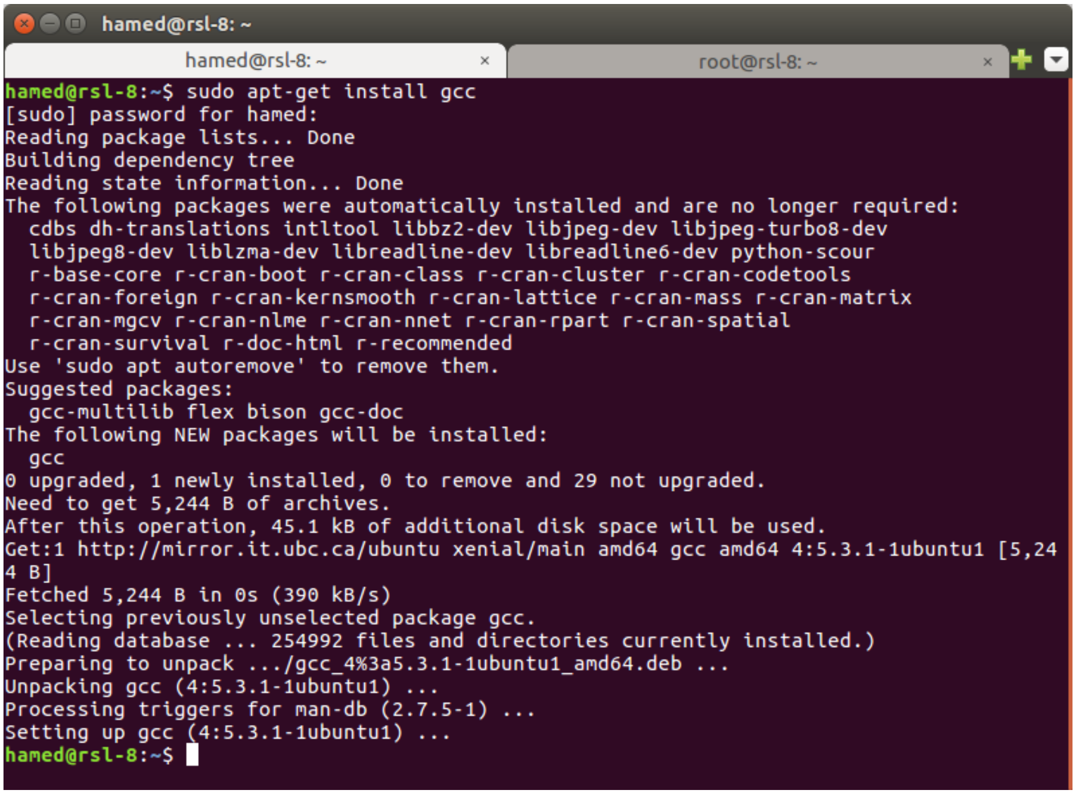


Figure 2. GCC Installation command and screenshot of installation process.

b) CMake Installation

Next step is to install the CMake which is open source cross platform for managing the build process of different software which used a compiler-independent method. Following command shows the process for building CMake from source and process is shown in *Figure 3*.

```
sudo apt-get install build-essential
wget https://cmake.org/files/v3.7/cmake-3.7.1.tar.gz
tar xf cmake-3.7.1.tar.gzcd cmake-3.7.1
./configure
make
sudo apt-get install checkinstall
sudo checkinstall
sudo make install
```

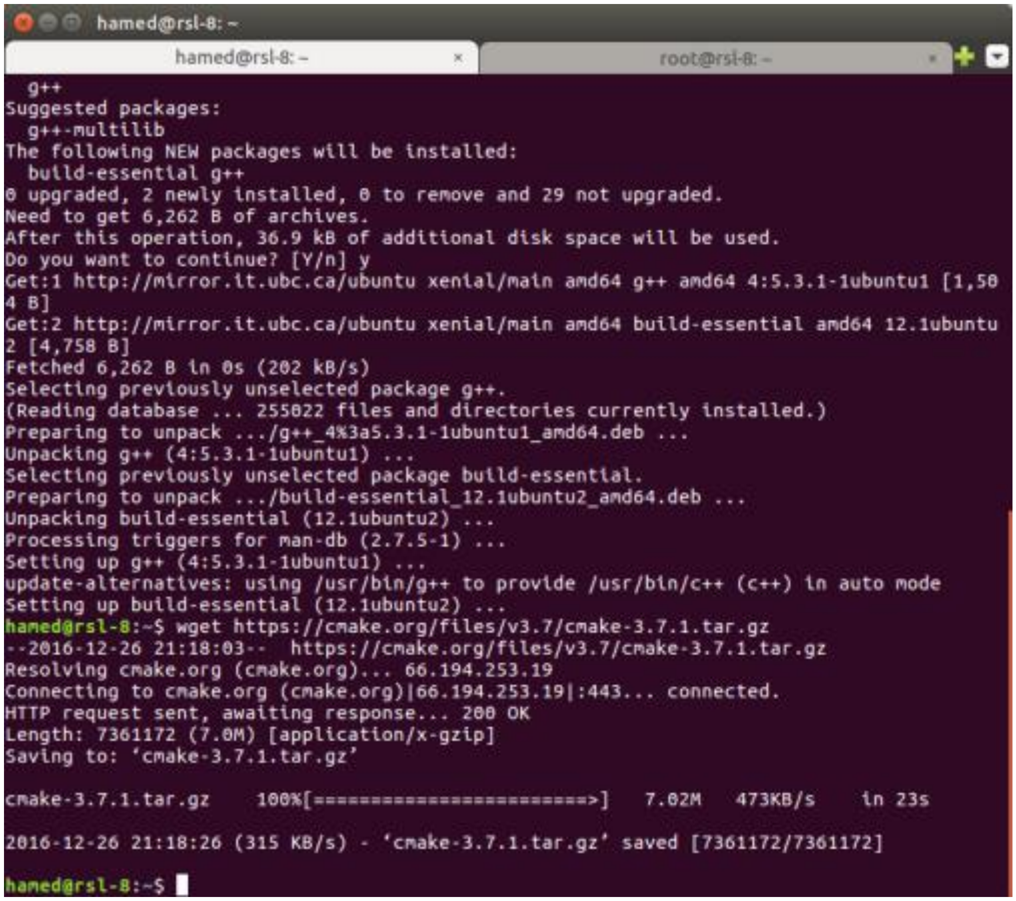


Figure 3. CMake Installation command and screenshot of installation process.

c) Installation of other Dependencies

Next, following software need to be installed:

- **Boost**: provides peer-reviewed C++ sources libraries.
- **Mako**: is used for source code generation.
- **Doxygen**: is standard tool for generating documentation from C++ sources.

Following command is used to install the mentioned programs:

```
wget -O boost_1_55_0.tar.gz
http://sourceforge.net/projects/boost/files/boost/1.55.0/boost_1_55_0.tar.gz/download
tar xzvf boost_1_55_0.tar.gz
cd boost_1_55_0/
sudo apt-get update
sudo apt-get install build-essential g++ python-dev autotools-dev libicu-dev build-essential
libbz2-dev libboost-all-dev
./bootstrap.sh --prefix=/usr/local
./b2
sudo ./b2 install
sudo apt-get install mako
gunzip doxygen-$VERSION.src.tar.gz
tar xf doxygen-$VERSION.src.tar
cd doxygen-$VERSION
mkdir build
cd build
cmake ../
make
```

d) UHD Installation from Source Code

The UHD source is stored in a Git repository and could be install using following command.

```
git clone git://github.com/EttusResearch/uhd.git
or
git clone --recursive git://github.com/EttusResearch/uhd.git   // Including FPGA codes which
will populate the fpga-src submodule inside the repository.
```

Next step is building the UHD with downloaded source codes. First, Make file should be generated using following command:

```
cd $Home/host
mkdir build
cd build
cmake ../
```

Next, configuration variables need to be passed into CMake using following commands and build and install process can be start:

```
cmake -DCMAKE_INSTALL_PREFIX=64
make
make test
sudo make install
sudo ldconfig      // adding libuhd.so in LD_LIBRARY_PATH
```

After installation, the following command can be run to test whether the process was successful or not and the results will be a warning message that shown below:

```
> uhd_find_device

linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.000.HEAD-0-g6e1ac3fc

No UHD Devices Found
```

3.2. GNU Radio Installation

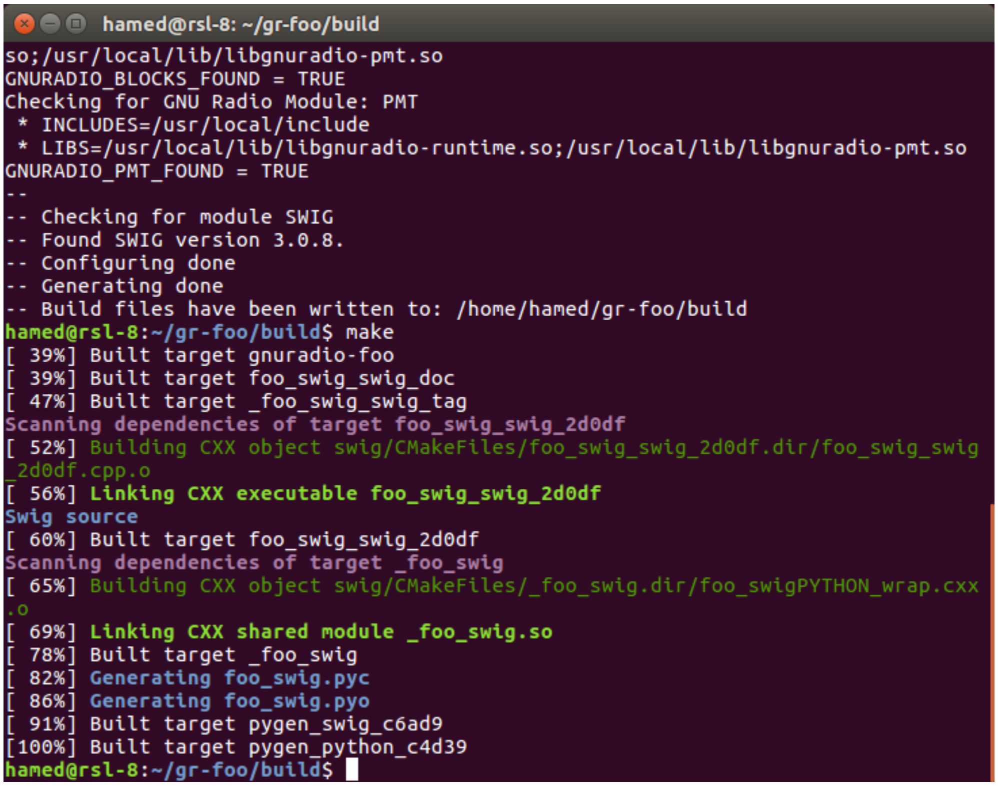
In order to install the GNURadio, building and installing the GNURadio using source code is selected. Similar to the process for UHD, to build GNU Radio from source code, first step is cloning the GitHub repository and next step is selecting a branch or tagged release of the repository, and final step is building and installing it. Following command is used for the procedure:

```
cd $HOME
mkdir workarea-gnuradio
cd workarea-gnuradio
git clone --recursive https://github.com/gnuradio/gnuradio
cd gnuradio
git checkout v3.7.10.1
mkdir build
cd build
cmake ../
make
make test
sudo make install
sudo ldconfig
```

3.3. Installing IEEE 802.11 Modules.

Similar to GNURadio, the IEEE 802.11 modules use master and branches for development, and some of blocks and branches from IEEE 802.11 framework are in septate module called gr-foo and it need to be installed before using IEEE 802.11. Following command is used to installed the gr-foo project and ssss shown the procedure:

```
git clone https://github.com/bastibl/gr-foo.git
cd gr-foo
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig
```



```
hamed@rsl-8: ~/gr-foo/build
so;/usr/local/lib/libgnuradio-pmt.so
GNURADIO_BLOCKS_FOUND = TRUE
Checking for GNU Radio Module: PMT
 * INCLUDES=/usr/local/include
 * LIBS=/usr/local/lib/libgnuradio-runtime.so;/usr/local/lib/libgnuradio-pmt.so
GNURADIO_PMT_FOUND = TRUE
--
-- Checking for module SWIG
-- Found SWIG version 3.0.8.
-- Configuring done
-- Generating done
-- Build files have been written to: /home/hamed/gr-foo/build
hamed@rsl-8:~/gr-foo/build$ make
[ 39%] Built target gnuradio-foo
[ 39%] Built target foo_swig_swig_doc
[ 47%] Built target _foo_swig_swig_tag
Scanning dependencies of target foo_swig_swig_2d0df
[ 52%] Building CXX object swig/CMakeFiles/foo_swig_swig_2d0df.dir/foo_swig_swig
_2d0df.cpp.o
[ 56%] Linking CXX executable foo_swig_swig_2d0df
Swig source
[ 60%] Built target foo_swig_swig_2d0df
Scanning dependencies of target _foo_swig
[ 65%] Building CXX object swig/CMakeFiles/_foo_swig.dir/foo_swigPYTHON_wrap.cxx
.o
[ 69%] Linking CXX shared module _foo_swig.so
[ 78%] Built target _foo_swig
[ 82%] Generating foo_swig.pyc
[ 86%] Generating foo_swig.pyo
[ 91%] Built target pygen_swig_c6ad9
[100%] Built target pygen_python_c4d39
hamed@rsl-8:~/gr-foo/build$
```

Figure 4. Installation of gr-foo and screenshot of installation process.

Similarly, gr-ieee802.11 can be installed using following command:

```
git clone git://github.com/bastibl/gr-ieee802-11.git
cd gr-ieee802-11
mkdir build
cd build
cmake ..
make
sudo make install
sudo ldconfig
```

Before using the IEEE 802.11 framework, the maximum shared memory need to be changed and environmental variable need to be adopted as shown in *Figure 5*.

```
sudo sysctl -w kernel.shmmax=2147483648
export PATH=~/usr/bin:${PATH}
export LD_LIBRARY_PATH=~/usr/lib:${LD_LIBRARY_PATH}
export PYTHONPATH=~/usr/lib/python2.7/site-packages:${PYTHONPATH}
```

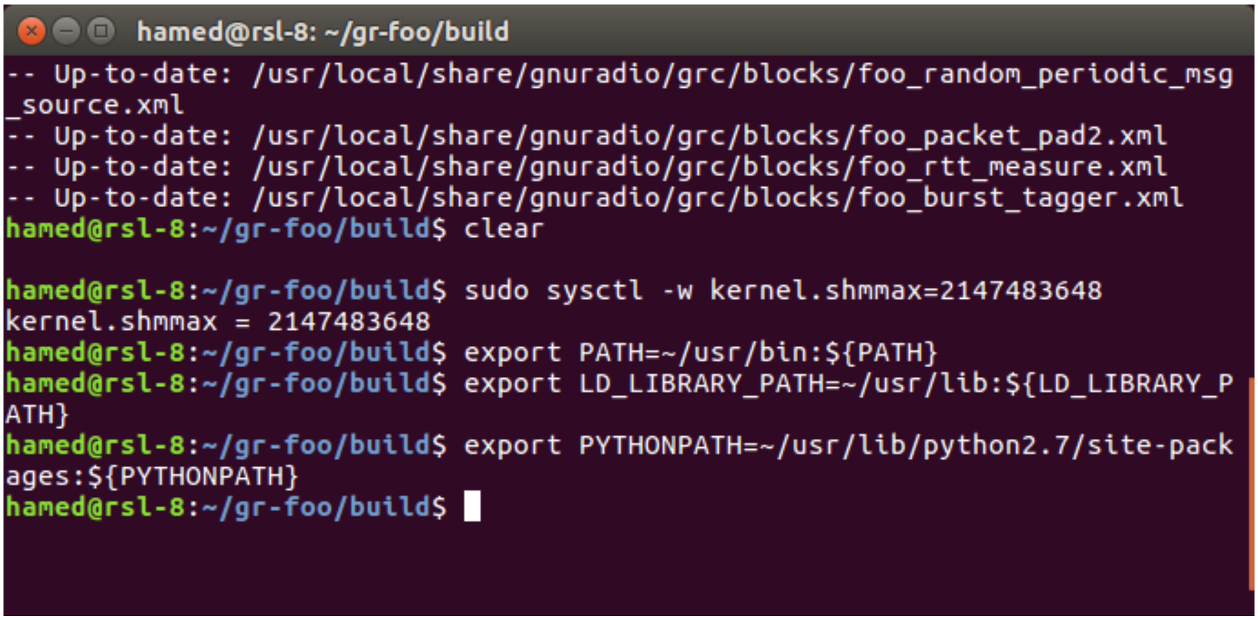


Figure 5. Adopting the environmental variable process.

4. Demonstration of the Functionality

4.1. GNURadio Companion Setup

As mentioned before, in this project GNURadio is used which is a modular "flowgraph"-oriented framework for digital signal processing. GNURadio has a graphical programming environment called GNURadio Companion (GRC) which the output a Python program. GNU Radio is a framework to develop signal processing blocks and create a "flowgraphs", which comprise radio processing applications. GNURadio comes with a huge sets of existing processing blocks that user can combine with their own develop blocks. As an example, Figure *6* presents a flowgraph in the GNURadio companion.

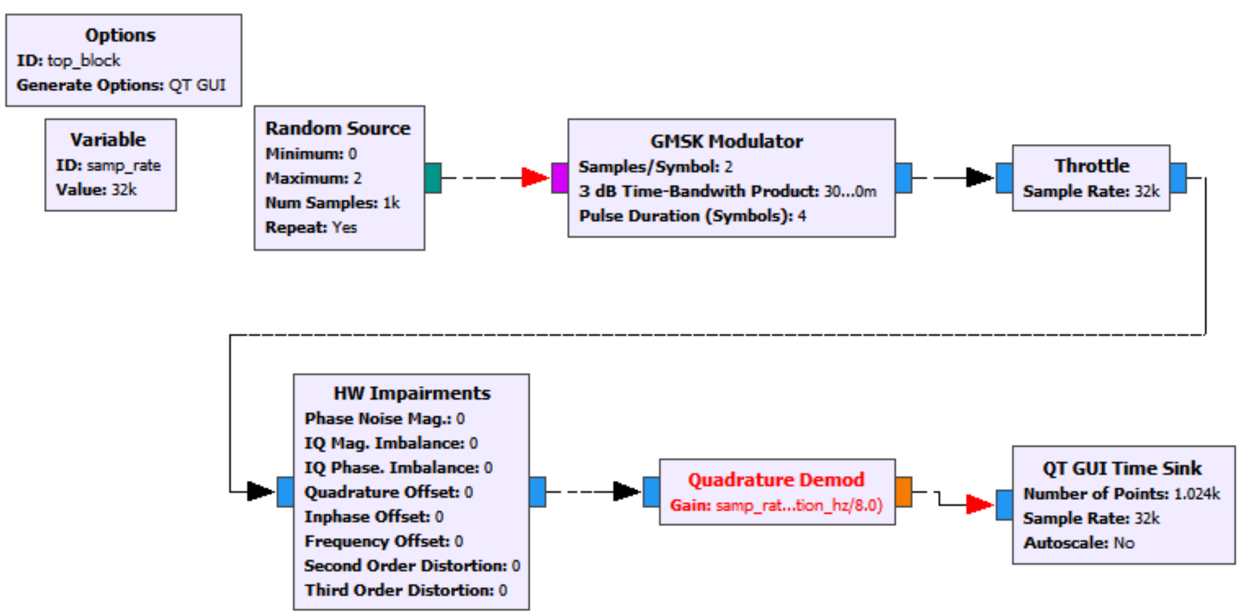


Figure 6.Set of different processing blocks and a flow graph in the GNURadio Companion.

After installation of GNURadio, in order to run GRC, following command as shown in . :

```
> sudo -H gnome-terminal
-------
gnuradio-companion  // in the new terminal
```

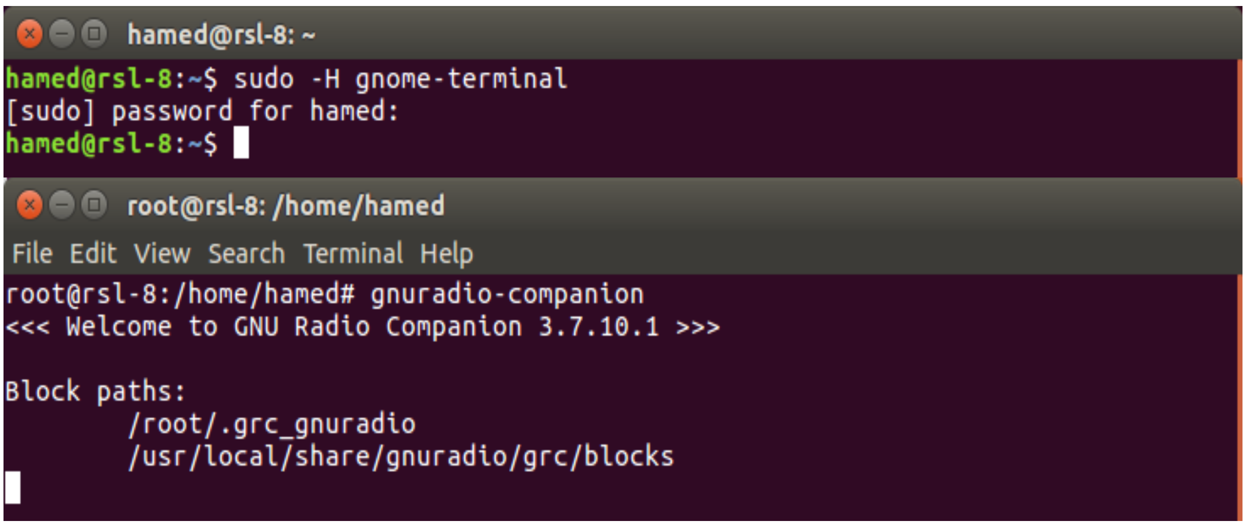


Figure 7. Running the GNURadio Companion (GRC).

The GRC interface includes four parts: Library, Toolbar, Terminal, and Workspace as shown in Figure 8.

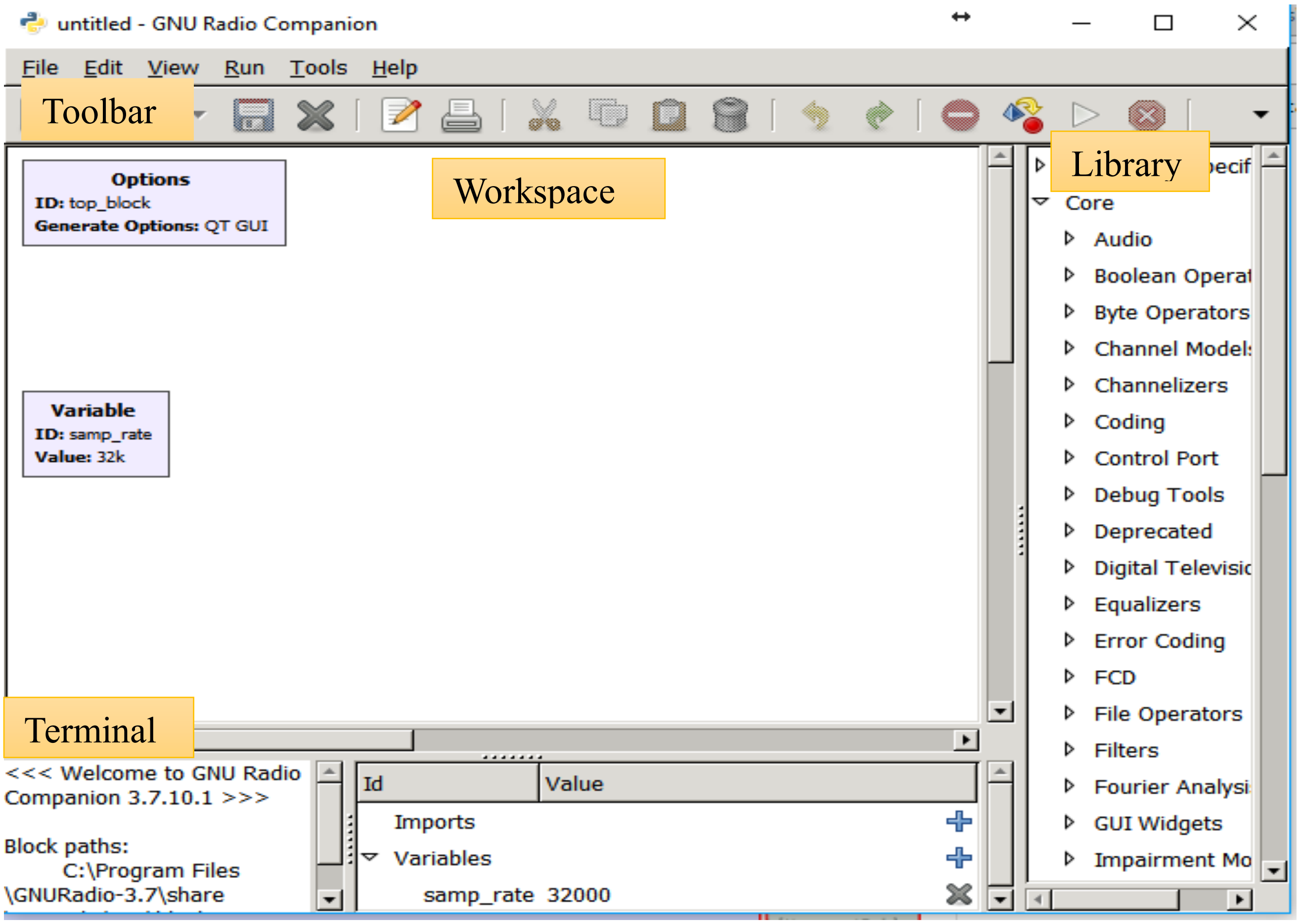


Figure 8.GNURadio Companion (GRC) Interface.

Using library section on the GRC interface, different blocks can be found by browsing categorized, or by click on search icon. Each block has it owns properties that can be open by double click on the block in the workspace area. Figure 9 shows an example of the modifying the properties of a block in GRC.

The Library contains the different blocks installed in the GRC block paths and it contains a large set of pre-developed blocks, such as:

- **Waveform Generators**
  - Constant Source
  - Noise Source
  - Signal Source
- **Instrumentation**
  - Constellation Sink
  - Frequency Sink
  - Histogram Sink
  - Number Sink
  - Time Raster Sink
  - Time Sink
  - Waterfall Sink
  - Channel Models
- **Channel Model**
  - Fading Model
  - Dynamic Channel Model
  - Frequency Selective Fading
- **Filters**
  - Band Pass / Reject Filter
  - Low / High Pass Filter
  - IIR Filter
  - Generic Filterbank
  - Hilbert
  - Decimating FIR Filter
  - Root Raised Cosine Filter
  - FFT Filter
- **Modulators**
  - AM Demodulation
  - Continuous Phase Modulation
  - PSK Mod / Demodulation
  - DPSK Mod / Demodulation
  - GMSK Mod / Demodulation
  - QAM Mod / Demodulation
  - WBFM Receive
  - NBFM Receive
- **Math Operators**
  - Abs
  - Add
  - Complex Conjugate
  - Integrate
  - Log10
  - Multiply
- **Fourier Analysis**
  - FFT
  - Log Power FFT
  - Goertzel
  - Fractional Resampler
  - Polyphase Arbitrary Resampler
  - Rational Resampler
  - Clock Recovery MM
  - Correlate and Sync
  - Costas Loop
  - FLL Band-Edge
  - PLL Freq Det
  - PN Correlator
  - Polyphase Clock Sync

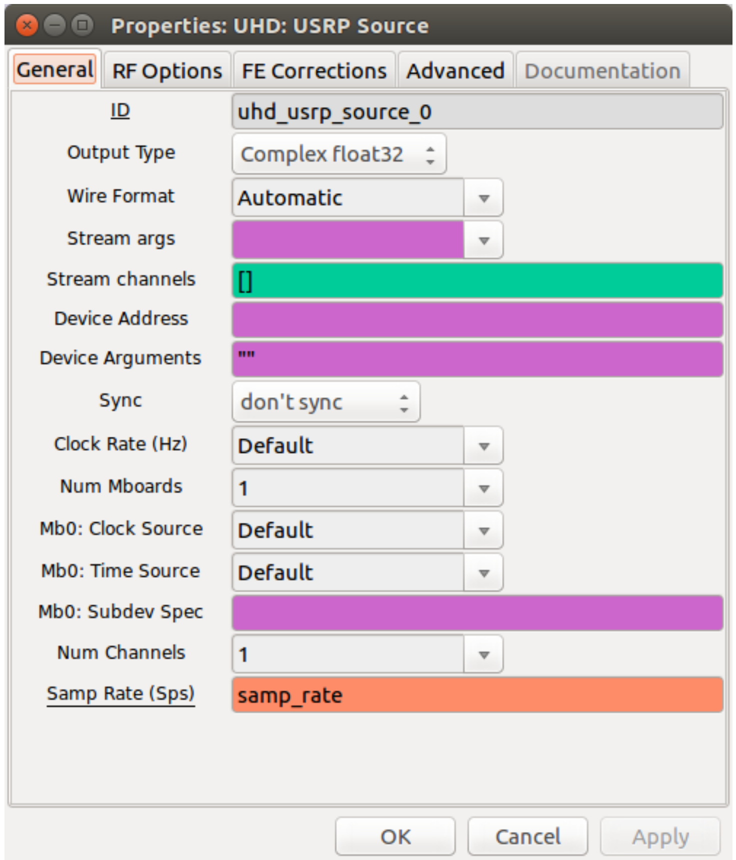


Figure 9.Example of modifying the properties of each block in GRC.

4.2. Hardware Setup

In order to test the functionality of IEEE 802.11 implementation in GNURadio, two National Instrument, NI USRP 2920 and one Ettus Research USRP N210 have been used, which each one can be configured in transmitter or receiver operation mode. *Figure 10* shows the internal structure of NI USRPs which includes transmitter and receiver section. Following section describes the transmitter and receiver operation mode:

- **Transmitter**: The USRP receives a waveform from the host PC and the signal is up-converted to a radio frequency (RF) before being sent to an amplifier and then transmitted over the air with the antenna.

- **Receiver**: In the receiving operation mode, the received RF signal in USRP will be mixed with a desired carrier frequency in order to down-convert it to a complex I/Q baseband. Next, the digital received signal will be down-sampled to a rate specified by the user and passed to the host PC for processing. Using the NI USRP with full digitizer rate(100MSa/s), will provide a bandwidth of 20 MHz.

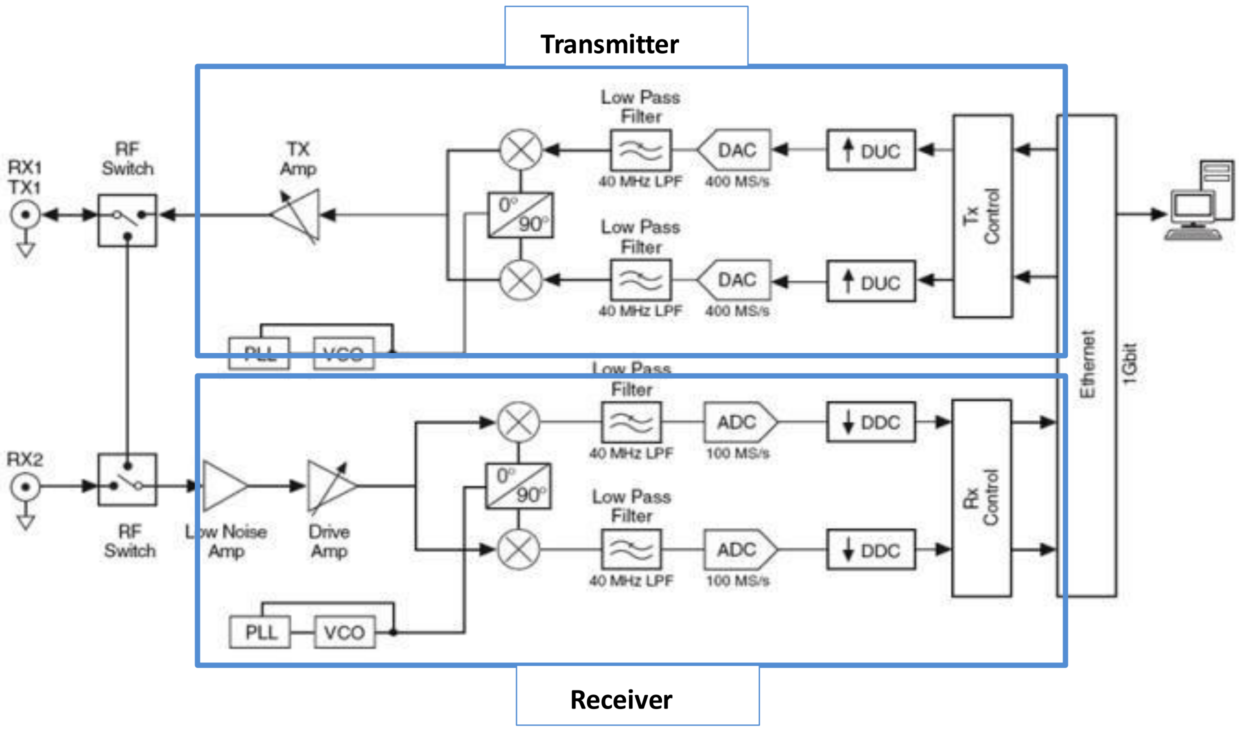


Figure 10.USRP Block Diagram.

The NI USRPs communicate with the host computer over gigabit Ethernet connection and before using them network configuration need to be done. The default IP address for the NI 2920 USRPs is 192.168.10.4. First, a static IP address in the host computer need to be configured on the same subnet as the connected device, as shown below.

- Host PC static IP address: 192.168.10.1
- Host PC subnet mask: 255.255.255.0
- Default USRP device IP address: 192.168.10.4

To perform a communication test between two USRPs, each USRP was connected to the separate Lenovo mini PC, one acting as a transmitter and other as a receiver. A sample wireless and wired setup at RSL@UBC is shown in *Figure 11*.

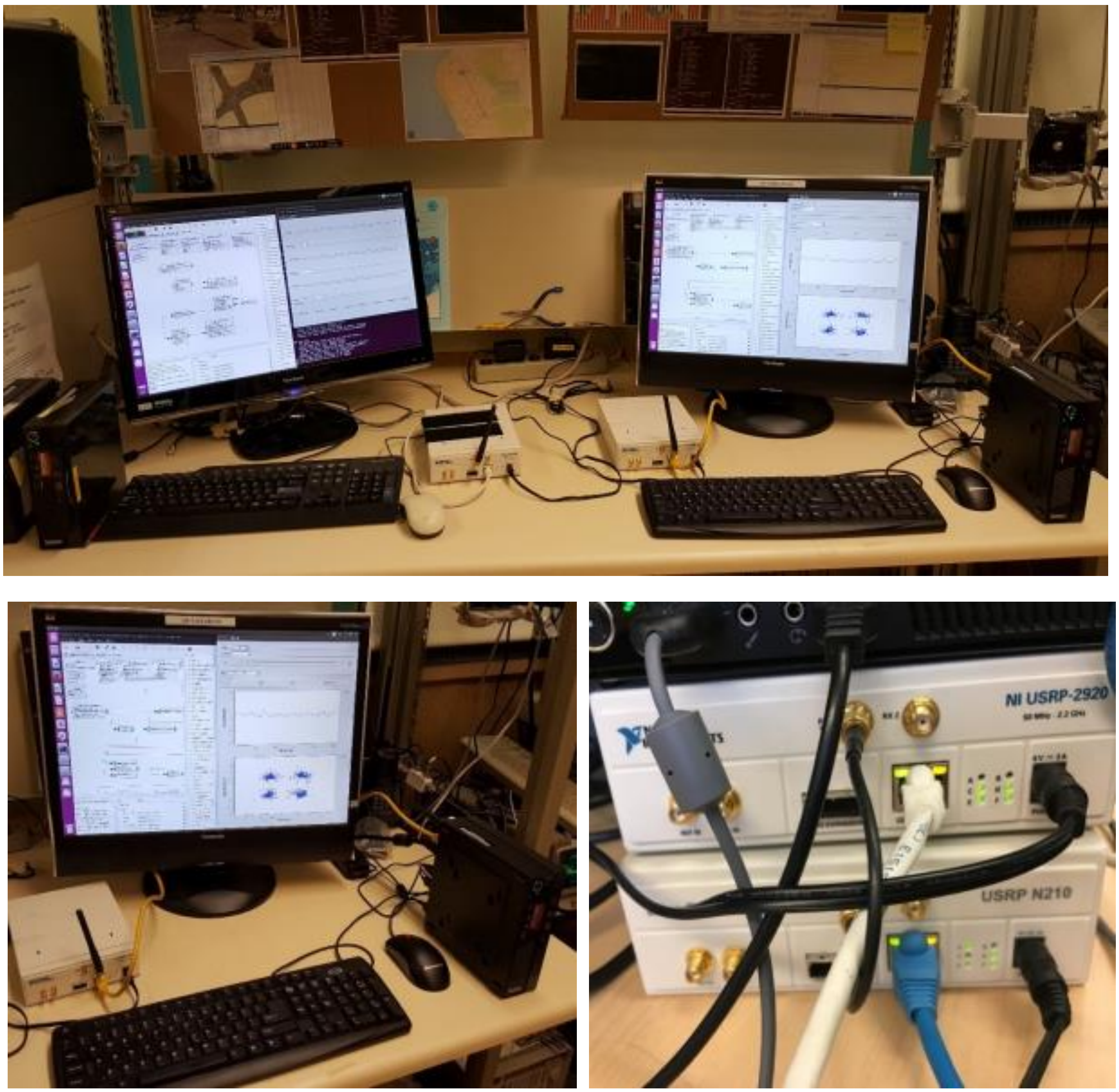


Figure 11.USRPs Hardware Setup at RSL@UBC.

4.3. Application Software Setup

Previous sections explained the required hardware and software setup in order to be able to run an application and develop prototype using USRP and GNURadio. This section provides information about three different IEEE 802.11 standard application software those are developed in GNURadio. First, a Wi-Fi simulation will be explained. Next, information about a transmitter and receiver over Wi-Fi using the NI USRP will be provided.

a) WiFi Simulation

The IEEE 802.11 modules developed by Bastian Bloessl [1], includes a Wi-Fi simulations flowpraph that can be used to performed basic study on the Wi-Fi before using USRPs. Figure 12 shows the flowgraph in the GRC workspace environment. This software includes several radio blocks, the XML and python code for each block, and different types of chooser blocks, such as:

- Channel Estimation Block
- Encoding block
- Wi-Fi MAC layer block
- Wi-Fi Physical Layer block
- Constellation Visualization block

By clicking on the execute flow graph button on the GNURadio companion toolbar, the simulation will be run and the output and controller shown in Figure 13 will be the output results.

Using this controller, following properties can be selected:

- Signal to Noise Ratio.
- PDU Length.
- Messages Interval
- Encoding method
- Channel estimation method

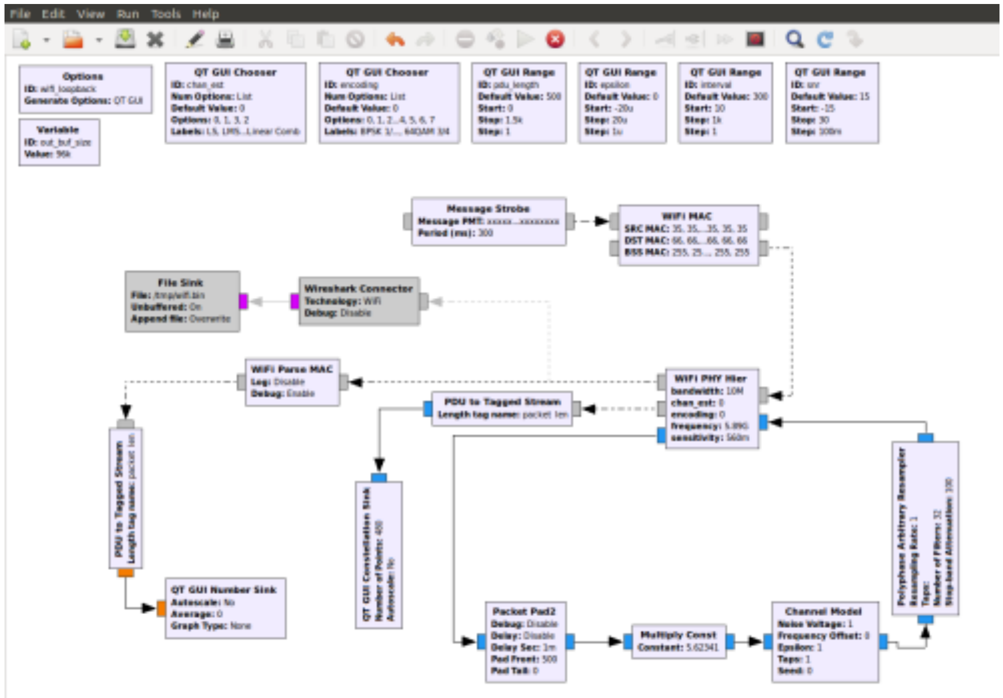


Figure 12.Wi-Fi Simulation software developed in GNURadio.

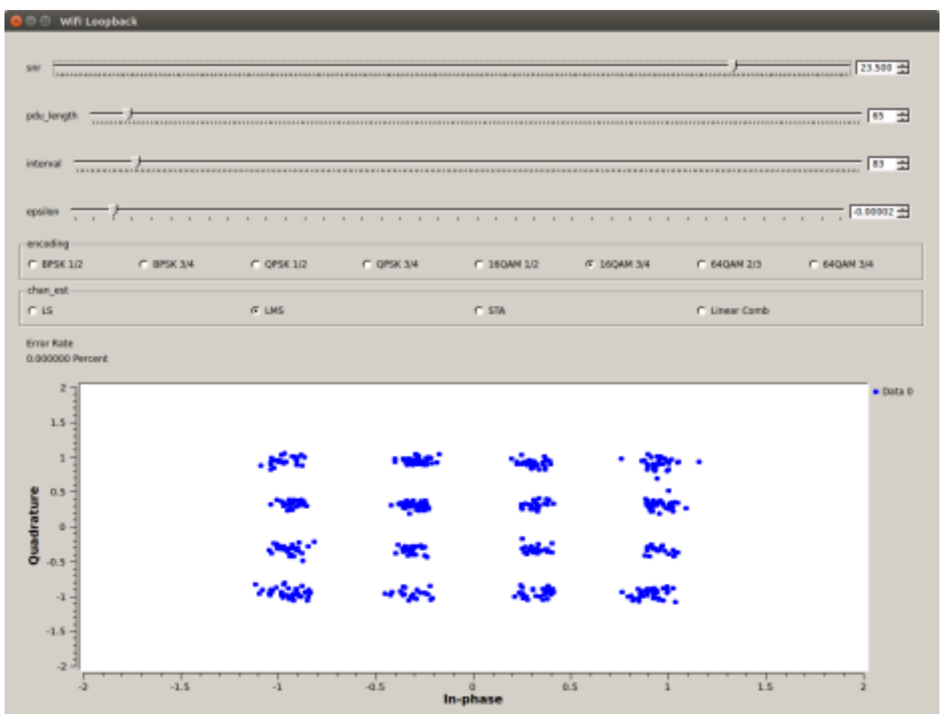


Figure 13.Wi-Fi simulation software controller and output.

b) Wi-Fi Transmitter

To perform a full communication chain using USRPs, one of them is configured as a transmitter over Wi-Fi. *Figure 14* illustrates the blocks and flowgraph which are used in the transmitter side. The Lenovo mini PC was connected to the NI USRP 2920. This flowgraph includes several radio blocks, the XML and python code for each block, and different types of chooser blocks, such as:

- Encoding block
- Wi-Fi MAC layer block
- Wi-Fi Physical Layer block
- USRP Sink block

By clicking on the execute flow graph button on the GNURadio companion toolbar, the flowgraph will be run and the USRP will start to transmit the packets and also the parameters controller which shown *Figure 15*in will be the output results.

Using this controller, following properties can be selected:

- Transmitter Gain (range from 0 to 1).
- Sample Rate (10 MHz and 20 MHz).
- PDU Length.
- LO Offset of USRP.
- Packets Interval.
- Encoding method.
- Frequency Selection (from 2.2 to 5.9).

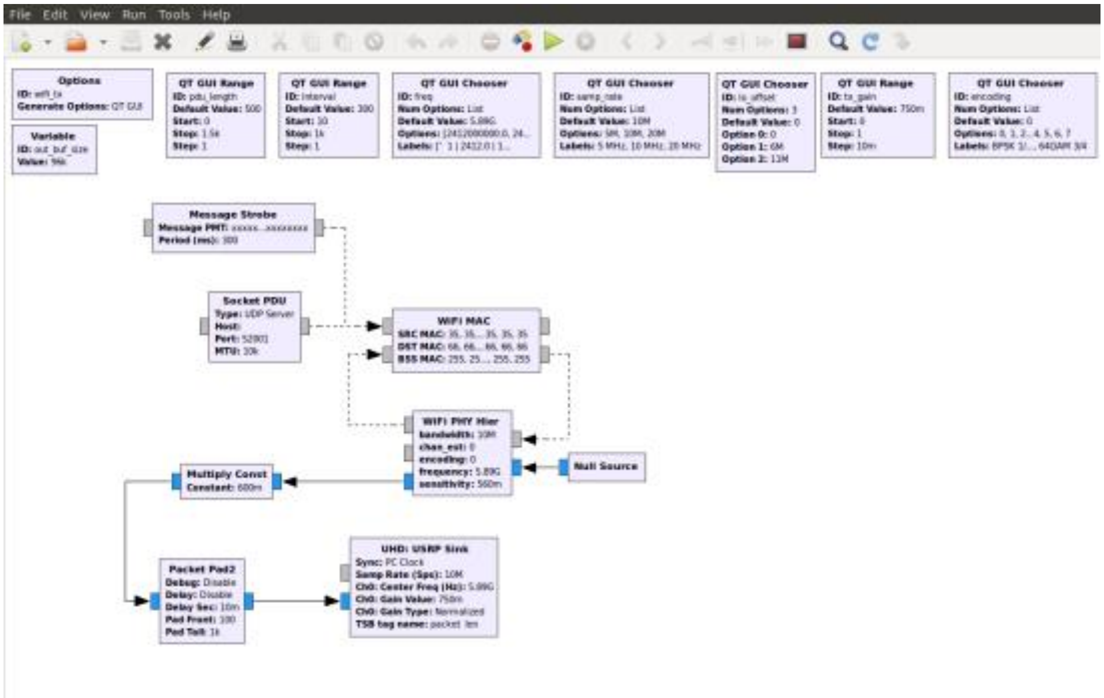


Figure 14. Wi-Fi Transmitter flowgraph and blocks developed in GNURadio Companion.

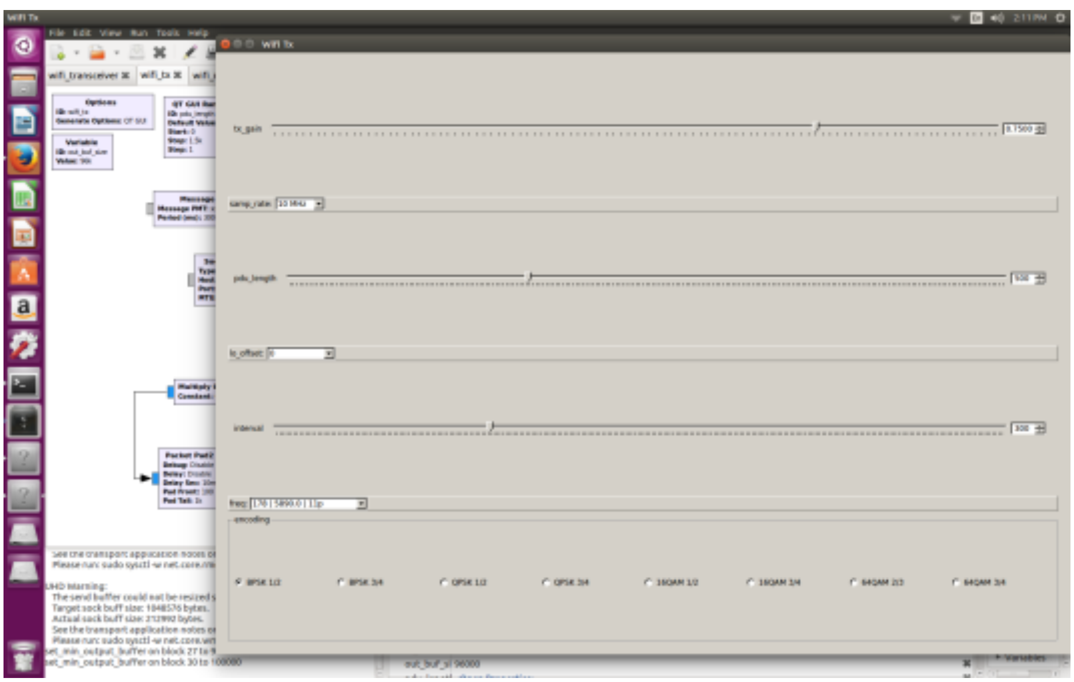


Figure 15. Wi-Fi Transmitter parameters controller in GNURadio Companion for NI USRP.

c) WiFi Receiver and Wi-Fi Detection

To perform a full communication chain using USRPs, one of them is configured as a receiver over Wi-Fi. *Figure 16* illustrates the blocks and flowgraph which are used in the transmitter side. The Lenovo mini PC was connected to the NI USRP 2920. This flowgraph includes several radio blocks, the XML and python code for each block, and different types of chooser blocks, such as:

- Encoding block
- Wi-Fi MAC layer block
- Wi-Fi Physical Layer block
- USRP Sink block
- Channel Estimation block
- Frame Detection method
- Phase and Frequency Offset Correction
- Signal Decoding
- Frame Decoding

By clicking on the execute flow graph button on the GNURadio companion toolbar, the flowgraph will be run and the USRP will start to transmit the packets and also the parameters controller which shown in *Figure 17 Figure 15* will be the output results.

Using this controller, following properties can be selected:

- Receiver Gain (range from 0 to 1).
- Sample Rate (10 MHz and 20 MHz).
- LO Offset of USRP.
- Channel Estimation method.
- Graphical illustration of Amplitude.
- Constellation diagram

In addition to communications between two USRP, a Wi-Fi signal detections test also is performed using the same receiver software in GRC and Ettus Research USRP N210. The hardware setup is shown in Figure 18 and the output and results is illustrated in Figure 19 which is include the detection of the frames from "*ubcprivate*" Wi-Fi network.

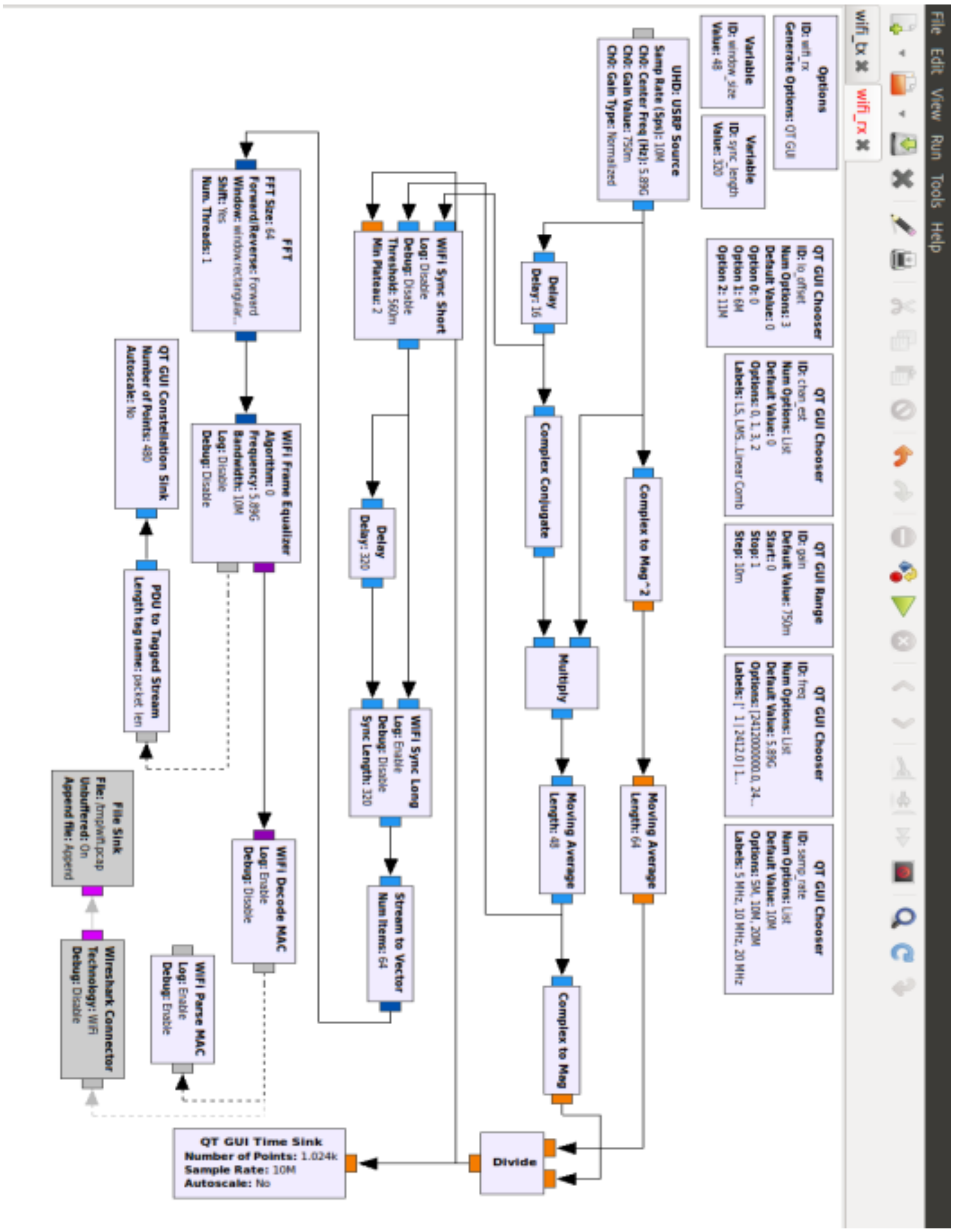


Figure 16. Wi-Fi Receiver flowgraph and blocks developed in GNURadio Companion.

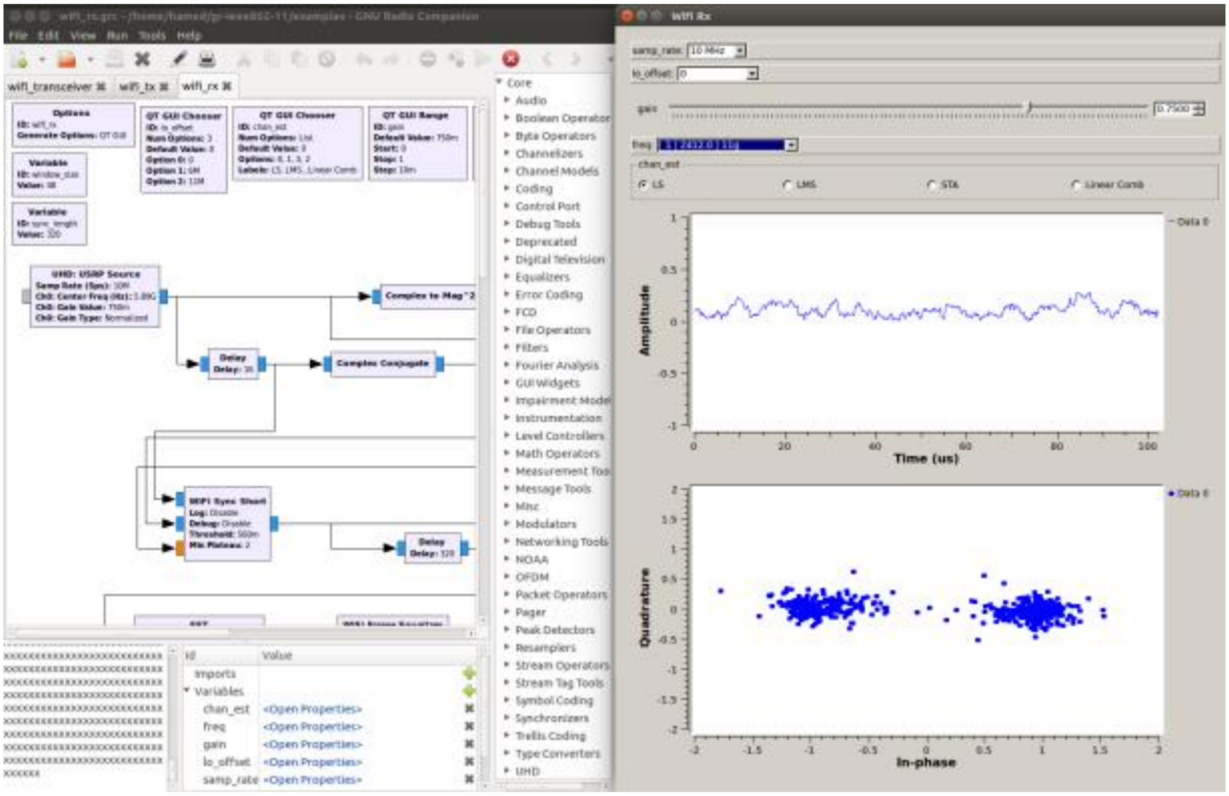


Figure 17. Wi-Fi Receiver output results and signals detections using NI USRP.

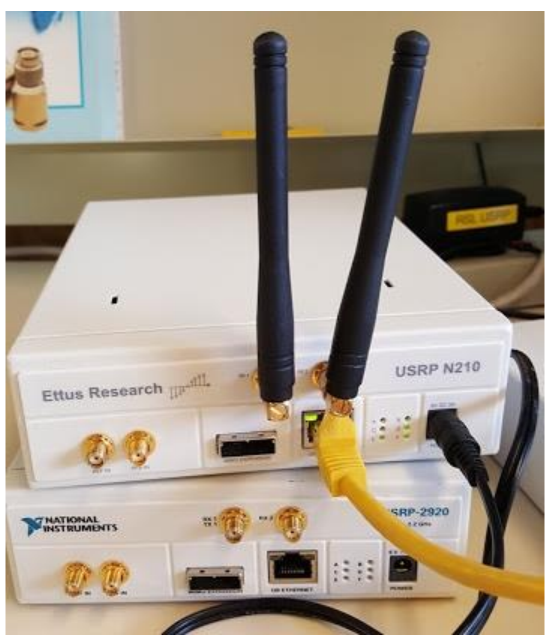


Figure 18.Ettus Research USRP N210 Setup for Wi-Fi Detection.

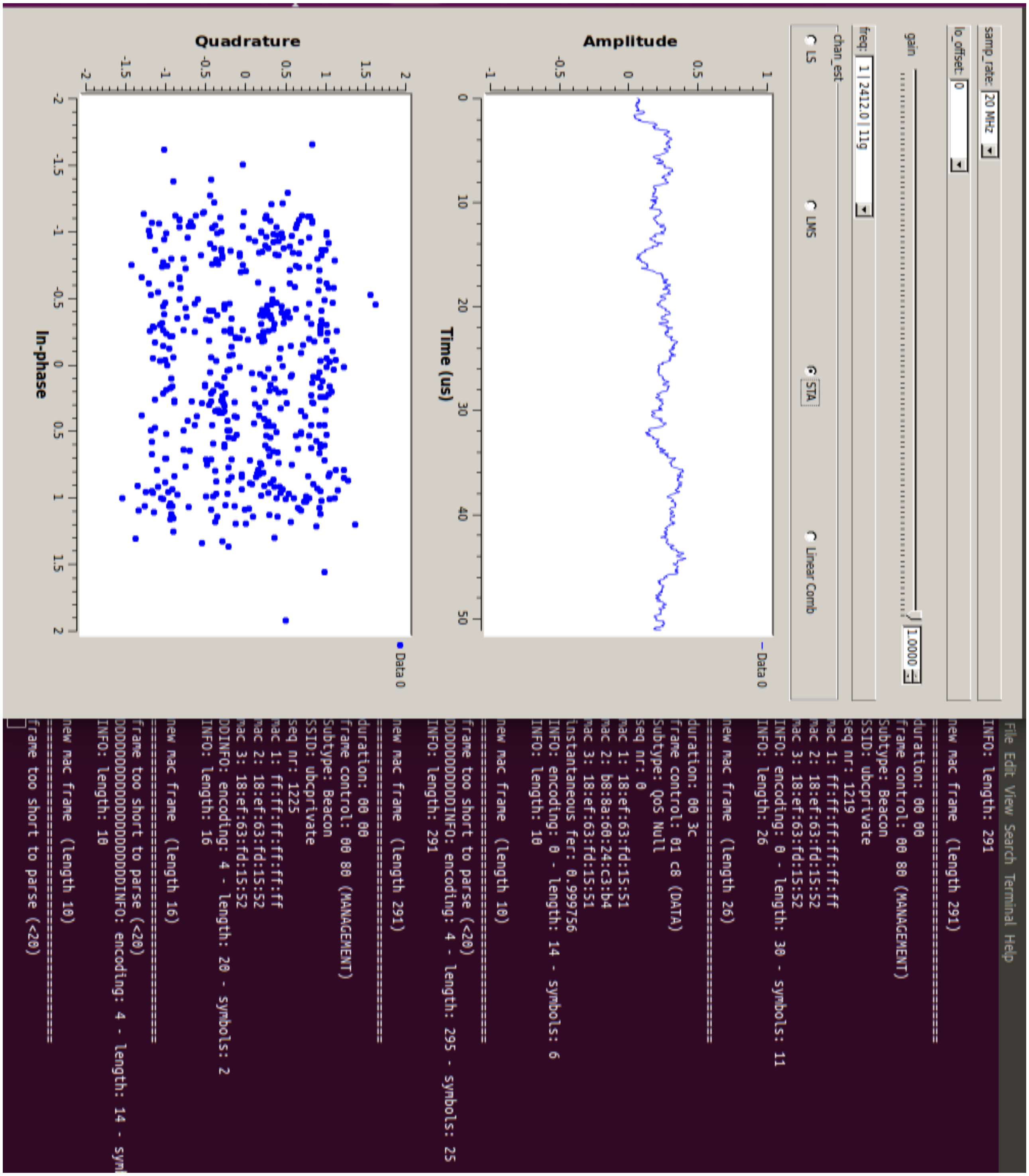


Figure 19. Wi-Fi Receiver results and signals detections.

5. Appendix

Datasheets

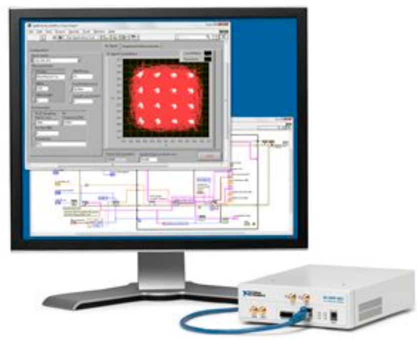
Requirements and Compatibility | Ordering Information | Detailed Specifications

For user manuals and dimensional drawings, visit the product page resources tab on ni.com.

Last Revised: 2015-08-17 15:09:46.0

NI USRP-292x/293x Datasheet

Universal Software Radio Peripherals



- Affordable, rapid prototyping solution
- Tunable RF transceivers options covering 50MHz – 6GHz
- Stream up to 25MS/s baseband IQ for live host-based processing with LabVIEW
- Optional GPS disciplined OCXO improves clock precision and enables global device synchronization position (USRP-293x)
- Plug-and-play MIMO synchronization
- Reference design examples for record & playback, physical layer prototyping, and spectral monitoring
- NI technical support and 1-year extendable warranty

Overview

NI USRP-292x and 293x software defined radio transceivers are designed for wireless communications teaching and research. Programmable with LabVIEW software, the USRP hardware is an affordable and easy-to-use RF platform for rapid prototyping applications such as record & playback, physical layer communication, spectrum monitoring, and more. With the ability to transmit and receive RF signals across a wide range of frequencies with up to 20 MHz of real-time bandwidth and plug-and-play MIMO support, the NI USRP enables a broad range of RF/communications applications covering common standards such as broadcast radio, digital TV, GSM Cellular, GPS, 802.11 (WiFi) and ZigBee®. LabVIEW brings increased productivity with an intuitive graphical programming approach, and m-file script compatibility enabling development of algorithms for physical layer communications. The NI USRP-293x has an integrated GPS-disciplined clock that provides improved frequency accuracy, synchronization capabilities, and GPS position information.

Back to Top

Requirements and Compatibility

OS Information

- Windows 7 32-bit
- Windows 7 64-bit
- Windows 8.1 32-bit
- Windows 8.1 64-bit
- Windows XP

Driver Information

- NI-USRP

Software Compatibility

- LabVIEW

Back to Top

Comparison Tables

Model	Frequency	GPS-Disciplined	Freq. Accuracy (No GPS Antenna)
NI USRP-2920	50 MHz to 2.2 GHz	No	2.5 ppm
NI USRP-2921	2.4 to 2.5 GHz and 4.9 to 5.9 GHz	No	2.5 ppm
NI USRP-2922	400 MHz – 4.4 GHz	No	2.5 ppm
NI USRP-2930	50 MHz – 2.2 GHz	Yes	25 ppb
NI USRP-2932	400 MHz – 4.4 GHz	Yes	25 ppb

Application and Technology

An Affordable Lab-Ready Solution

The USRP software defined radio offers new opportunities for RF and communications education, which traditionally has been focused on mathematical theory. Students can use the platform with LabVIEW to link theory and practical implementation through hands-on execution and exploration of a working communications system using live signals.

Figure 1. The NI Digital Communications Bundle

The NI Digital Communications Bundle is an affordable turnkey solution that addresses both time and budget concerns. It includes a pair of NI USRP-2920 transceivers, which enables implementation of a live communication link at frequencies between 50 MHz and 2.2 GHz with up to a 20 MHz bandwidth. The bundle also features a laboratory manual, *Digital Wireless Communication: Physical Layer Exploration Lab Using the NI USRP*, by Dr. Robert Heath, a professor at The University of Texas at Austin Department of Electrical and Computer Engineering. This material covers each element of a modern digital communications system and culminates with the design of an OFDM radio. The course material includes a printed lab manual with background content, pre-lab exercises, step-by-step lab instructions, and LabVIEW VIs associated with each exercise.

Lab 1.1	AWGN Simulator
Lab 2.1	Modulation/Demodulation
Lab 2.2	Pulse Shaping
Lab 3	Energy Detection
Lab 4	Equalization
Lab 5	Frame Detection
Lab 6	Intro to OFDM
Lab 7	Frequency Correction and Synchronization
Lab 8	OFDM Channel Coding

Table 1. Topics Covered by Digital Wireless Communication: Physical Layer Exploration Lab Using the NI USRP

Communications Research

LabVIEW provides a scalable platform for communications research bridging design and test, allowing a designer to share code between USRP prototyping and PXI RF test hardware. Get started quickly by leveraging proven reference designs provided by NI for applications such as physical layer prototyping, record and playback, and GPS simulation or integrate your own written in LabVIEW, m-file script or C.

Hardware

The USRP software-defined radios are RF software-programmable radio transceivers designed for wireless communications research. When the USRP is connected to a host PC, it acts as a software-defined radio with host-based digital signal processing capabilities. Each USRP device provides an independent transmit and receive channel capable of full duplex operation in some hardware configurations.

The NI USRP-2930 and NI USRP-2932 include an integrated GPS-disciplined reference clock. The reference clock provides improved frequency accuracy, synchronization capabilities, and GPS position information. Figure 1 shows the USRP system block diagram. There are two separate transmit and receive signal chains which are common among all USRP models. The highlighted section represents the GPS disciplined clock, which is common to NI USRP-293x models. The reference clock is shared among both the transmit and receive paths, and from which the local oscillator (LO) is derived.

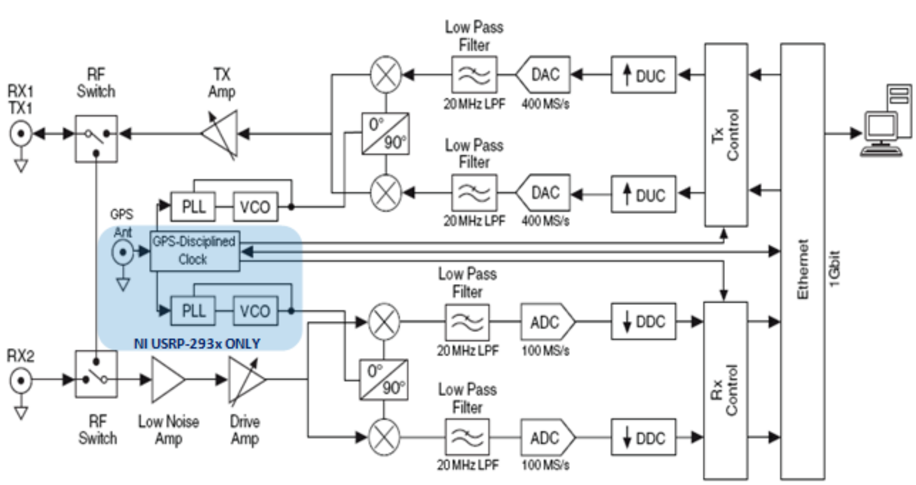


Figure 2. NI USRP-2920 System Block Diagram

LabVIEW Software and the NI-USRP Driver

The LabVIEW development system provides an ideal way to interface with NI USRP hardware for the development and exploration of communications algorithms that process received signals and synthesize signals for transmission. The NI-USRP software driver provides functions (LabVIEW VIs) for the hardware / software configuration with tools for opening / closing sessions and performing read/ write operations.

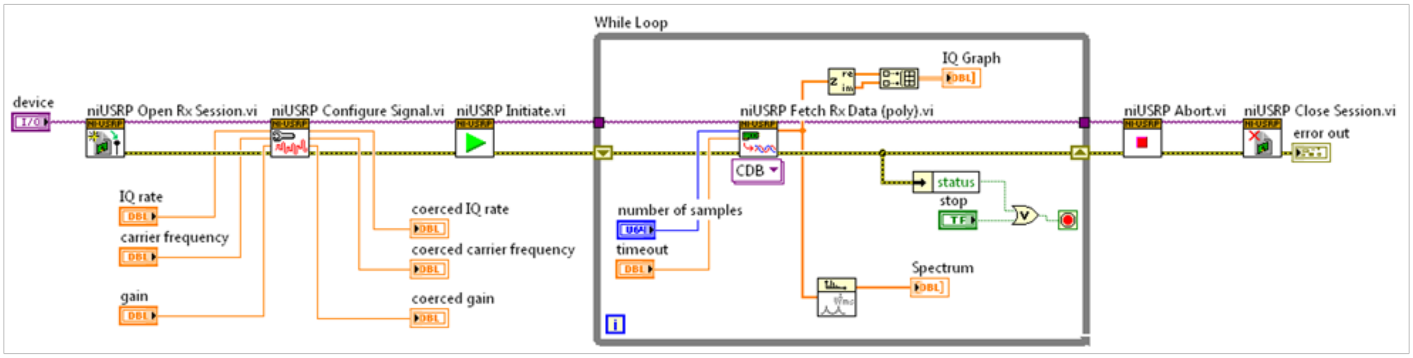


Figure 3. Continuous Receive Using NI-USRP Driver in LabVIEW Software

LabVIEW Modulation Toolkit

The LabVIEW Modulation Toolkit extends LabVIEW with VIs for rapid development of communications systems for simulation or to operate on live signals associated with NI-USRP hardware. On the transmit side, included VIs provide functionality for PN-sequence generation, channel coding, and baseband modulation. Receiver-side functionality includes demodulation, equalization, channel decoding and more. Included utilities enable the addition of baseband impairments, BER measurement, modulation domain measurements, and communications-oriented visualization. An entire communication system can be simulated and analyzed with channel impairments and then updated to transmit / receive live signals.

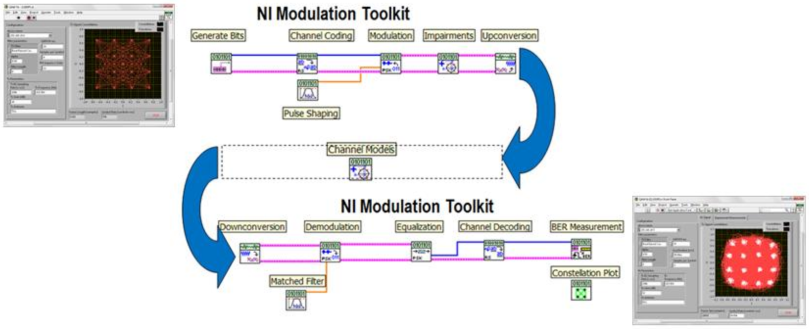


Figure 4. LabVIEW Modulation Toolkit

LabVIEW MathScript RT Module

With the LabVIEW MathScript RT Module, you can efficiently execute .m file syntax in a LabVIEW diagram to combine both graphical and textual models of computation. In addition, you can design and implement communications algorithms using the popular .m file script syntax.

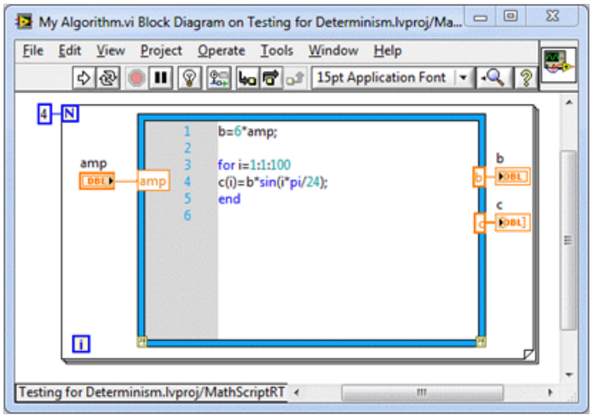


Figure 5. A LabVIEW MathScript RT Node Inside a LabVIEW For Loop

Back to Top

Ordering Information

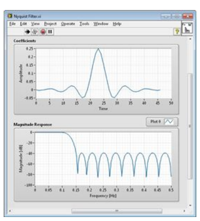
For a complete list of accessories, visit the product page on ni.com.

Products	Part Number	Recommended Accessories	Part Number
Digital Communication Bundle			
NI Digital Communications Bundle: 2 NI USRP-2920 Kits With LabVIEW Add-Ons, MIMO Cable, and Ready-to-Use Courseware	781908-01	No accessories required.	
NI USRP-29xx Hardware Kits			
NI USRP-2921, 2.4 AND 5 GHz Software Radio Kit: LabVIEW Driver, Modulation Toolkit, MathScript RT, Digital Filter Design Toolkit	781907-01	No accessories required.	
NI USRP-2920, 50 MHz to 2.2 GHz Software Radio Kit: LabVIEW Driver, Modulation Toolkit, MathScript RT, Digital Filter Design Toolkit	781906-01	No accessories required.	
NI USRP-2930, 50MHz to 2.2 + GPS Clock GHz Software Radio Kit: LabVIEW Driver, Modulation Toolkit, MathScript RT, Digital Filter Design Toolkit	781910-01	No accessories required.	
NI USRP-2932, 400MHz to 4.4 GHz + GPS Clock Software Radio Kit: LabVIEW Driver, Modulation Toolkit, MathScript RT, Digital Filter Design Toolkit	781911-01	No accessories required.	
NI USRP-2922, 400MHz to 4.4 GHz Software Radio Kit: LabVIEW Driver, Modulation Toolkit, MathScript RT, Digital Filter Design Toolkit	781909-01	No accessories required.	
Optional Accessories			
MIMO Sync and Data Transfer Cable for NI USRP-292x, 0.5M	781916-01	No accessories required.	

Back to Top

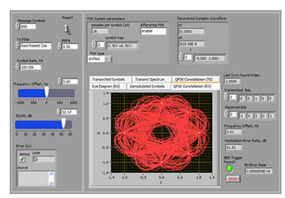
Software Recommendations

LabVIEW Digital Filter Design Toolkit



- Floating- and fixed-point design with LabVIEW or ANSI C autocode generation
- Digital filter design, analysis, and implementation within LabVIEW
- Comprehensive tools that meet basic or advanced needs
- Includes Digital Filter Design MathScript RT Module functions

Modulation Toolkit



- Simulate and measure impairments including DC offset, I/Q gain imbalance, and quadrature skew
- Handles standard and custom modulation formats(AM, FM, PM, ASK, FSK, MSK, GMSK, PSK, QPSK, PAM, QAM)
- Measurements including bit error rate (BER), phase error, burst timing, and frequency deviation
- Quality measurements including EVM, modulation error ratio (MER), and ρ (rho)
- Powerful 3D eye diagrams enhance the suite of traditional 2D eye, trellis, and constellation plots
- This toolkit is included on the RF Device Drivers DVD.

Back to Top

Support and Services

Technical Support

Get answers to your technical questions using the following National Instruments resources.

- **Support** - Visit ni.com/support to access the NI KnowledgeBase, example programs, and tutorials or to contact our applications engineers who are located in NI sales offices around the world and speak the local language.
- **Discussion Forums** - Visit forums.ni.com for a diverse set of discussion boards on topics you care about.
- **Online Community** - Visit community.ni.com to find, contribute, or collaborate on customer-contributed technical content with users like you.

Repair

While you may never need your hardware repaired, NI understands that unexpected events may lead to necessary repairs. NI offers repair services performed by highly trained technicians who quickly return your device with the guarantee that it will perform to factory specifications. For more information, visit ni.com/repair.

Training and Certifications

The NI training and certification program delivers the fastest, most certain route to increased proficiency and productivity using NI software and hardware. Training builds the skills to more efficiently develop robust, maintainable applications, while certification validates your knowledge and ability.

- **Classroom training in cities worldwide** - the most comprehensive hands-on training taught by engineers.
- **On-site training at your facility** - an excellent option to train multiple employees at the same time.
- **Online instructor-led training** - lower-cost, remote training if classroom or on-site courses are not possible.
- **Course kits** - lowest-cost, self-paced training that you can use as reference guides.
-

Training memberships and training credits - to buy now and schedule training later.

Visit ni.com/training for more information.

Extended Warranty

NI offers options for extending the standard product warranty to meet the life-cycle requirements of your project. In addition, because NI understands that your requirements may change, the extended warranty is flexible in length and easily renewed. For more information, visit ni.com/warranty.

OEM

NI offers design-in consulting and product integration assistance if you need NI products for OEM applications. For information about special pricing and services for OEM customers, visit ni.com/oem.

Alliance

Our Professional Services Team is comprised of NI applications engineers, NI Consulting Services, and a worldwide National Instruments Alliance Partner program of more than 700 independent consultants and integrators. Services range from start-up assistance to turnkey system integration. Visit ni.com/alliance.

Back to Top

Detailed Specifications

Characteristics	NI USRP-2920	NI USRP-2921	NI USRP-2922	NI USRP-2930	NI USRP-2932
Transmitter					
Frequency Range	50MHz-2.2GHz	2.4GHz-2.5GHz 4.9GHz-5.9GHz	400MHz-4.4GHz	50MHz- 2.2GHz	400MHz-4.4GHz
SW Adjustable TX Frequency Step			< 1KHz		
MAX Output Power			15 dBm - 20 dBm		
TX Output Power Gain Range			0 dB - 31 dB		
SW Adjustable Output Power Step Size			1dB		
Instantaneous Real-Time Bandwidth			20MHz (16bit samples) 40MHz (8bit-samples)		
DAC (Digital to Analog Conversion)			2 channels, 400MS/s, 16 bit		
DAC SFDR (Spurious Free Dynamic Range)			80 dB		
Receiver					
Software Adjustable RX Frequency Step			< 1KHz		
Max Input Power (Pin)			0 dBm		
Noise Figure			5 to 7 dB		
			20MHz (16bit samples)		

Instantaneous Real-Time Bandwidth	40MHz (8bit-samples)
ADC (Analog to Digital Conversion)	2 channels, 100MS/s, 14 bit
ADC SFDR (Spurious Free Dynamic Range)	88 dB

Reference Clock					
Clock Type	TCXO	TCXO	TCXO	OCXO	OCXO
GPS Disciplined	NO	NO	NO	YES	YES
Freq.Accuracy of 10MHz Ref (No GPS Antenna)	2.5ppm	2.5ppm	2.5ppm	25 ppb	25 ppb

Shared Characteristics (Apply all NI-29xx devices)

Connections		**Physical Specifications**	
TX1 RX1, TX2 RX2, RX2 Ports	SMA	Enclosure Dimensions	6.25” Wide x 1.9” Tall x 8.35” Deep
Ethernet Connection	1 Gigabit Ethernet	Weight	2.63 lbs
Power Adapter	6VDC, 3A	Operating Temperature	23°C ±5°, Room Temperature
Ref Clock (10-Mhz external reference input)	SMA, 10 MHz		
PPS Input (Pulse Per Second reference input)	SMA, 3-5V TTL Compatible		
MIMO Expansion Port	High-Speed SerDes protocol		

Back to Top

GETTING STARTED GUIDE

NI USRP-2920/2921/2922

Universal Software Radio Peripheral

This document explains how to install, configure, and test the National Instruments universal software radio peripheral (USRP) 2920, 2921, or 2922 (NI 292x) device. The NI 292x can send and receive signals for use in various communications applications. This device ships with the NI-USRP instrument driver, which you can use to program the device.

To access NI 292x documentation, navigate to **Start»All Programs»National Instruments»NI-USRP»Documentation**.

Contents

Electromagnetic Compatibility Guidelines

This product was tested and complies with the regulatory requirements and limits for electromagnetic compatibility (EMC) stated in the product specifications. These requirements and limits provide reasonable protection against harmful interference when the product is operated in the intended operational electromagnetic environment.

This product is intended for use in industrial locations. However, harmful interference may occur in some installations, when the product is connected to a peripheral device or test object, or if the product is used in residential or commercial areas. To minimize interference with radio and television reception and prevent unacceptable performance degradation, install and use this product in strict accordance with the instructions in the product documentation.

Furthermore, any changes or modifications to the product not expressly approved by National Instruments could void your authority to operate it under your local regulatory rules.

Caution To ensure the specified EMC performance, operate this product only with shielded cables and accessories.

Caution To ensure the specified EMC performance, the length of all I/O cables except those connected to the Ethernet antenna ports must be no longer than 3 m (10 ft).

Caution This product is not approved or licensed for transmission over the air using an antenna. As a result, operating this product with an antenna may violate local laws. Ensure that you are in compliance with all local laws before operating this product with an antenna.

Verifying the System Requirements

To use the NI-USRP instrument driver, your system must meet certain requirements.

Refer to the product readme, which is available on the driver software media or online at *ni.com/manuals*, for more information about minimum system requirements, recommended system, and supported application development environments (ADEs).

Unpacking the Kit

Caution To prevent electrostatic discharge (ESD) from damaging the device, ground yourself using a grounding strap or by holding a grounded object, such as your computer chassis.

1. Touch the antistatic package to a metal part of the computer chassis.
2. Remove the device from the package and inspect the device for loose components or any other sign of damage.

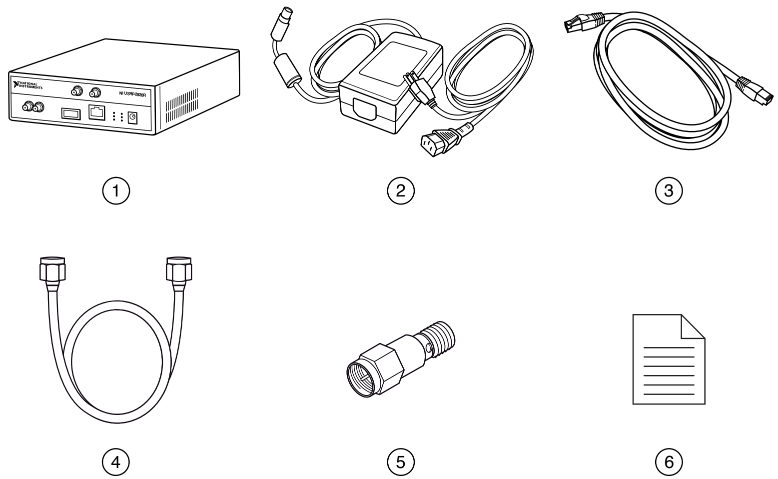
Caution Never touch the exposed pins of connectors.

Note Do not install a device if it appears damaged in any way.

3. Unpack any other items and documentation from the kit.

Store the device in the antistatic package when the device is not in use.

Verifying the Kit Contents



1. NI 29xx Device
2. AC/DC Power Supply and Power Cable
3. Shielded Ethernet Cable
4. SMA (m)-to-SMA (m) Cable
5. 30 dB SMA Attenuator
6. Getting Started Guide (This Document)

Caution If you directly connect or cable a signal generator to your device, or if you connect multiple USRP devices together, you must connect a 30 dB attenuator to the RF input (RX1 or RX2) of each receiving USRP device.

Other Required Items

In addition to the kit contents, you must provide the following additional items:

- Computer with available gigabit Ethernet interface.

Caution This product is not approved or licensed for transmission over the air using an antenna. As a result, operating this product with an antenna may violate local laws. Ensure that you are in compliance with all local laws before operating this product with an antenna.

Optional Items

- LabVIEW Modulation Toolkit (MT), included on the driver software media, which includes MT VIs and functions, examples, and documentation

  Note You must install the LabVIEW Modulation Toolkit for proper operation of the NI-USRP Modulation Toolkit example VIs.
- LabVIEW Digital Filter Design Toolkit, included on the driver software media
- LabVIEW MathScript RT Module, included on the driver software media
- USRP MIMO sync and data cable, available at `ni.com`, to synchronize clock sources
- Additional SMA (m)-to-SMA (m) cables to connect both channels with external devices or to use the REF IN and PPS IN signals
- GPS antenna for devices with GPS disciplined oscillator (GPSDO) support

Preparing the Environment

Ensure that the environment you are using the USRP device in meets the following specifications.

Operating temperature	23 ±5 °C
Operating humidity	10% to 90% relative humidity, noncondensing
Pollution Degree	2
Maximum altitude	2,000 m

Indoor use only.

Note Refer to the USRP device specifications at *ni.com/manuals* for complete specifications.

Caution Do not operate the NI 292x in a manner not specified in this document. Product misuse can result in a hazard. You can compromise the safety protection built into the product if the product is damaged in any way. If the product is damaged, return it to NI for repair.

Installing the Software

You must be an Administrator to install NI software on your computer.

1. Install an ADE, such as LabVIEW.
2. Visit *ni.com/info* and enter the Info Code `usrpdriver` to access the driver download page for the latest NI-USRP software.
3. Download the NI-USRP driver software.

4. Follow the instructions in the installation prompts.

Note Windows users may see access and security messages during installation. Accept the prompts to complete the installation.

5. When the installer completes, select **Restart** in the dialog box that prompts you to restart, shut down, or restart later.

Installing NI 292x Devices

Install all the software you plan to use before you install the hardware.

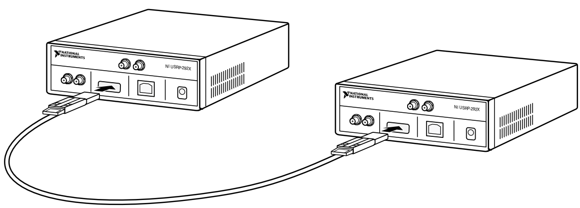
Note The NI 292x device connects to a host computer using a standard gigabit Ethernet interface. Refer to the documentation for your gigabit Ethernet interface for installation and configuration instructions.

1. Power on the computer.
2. Attach the antenna or cable to the front panel terminals of the NI 292x device as desired.
3. Use the Ethernet cable to connect the NI 292x device to the computer. For maximum throughput over Ethernet, NI recommends that you connect each NI 292x device to its own dedicated gigabit Ethernet interface on the host computer.
4. Connect the AC/DC power supply to the NI 292x device.
5. Plug the power supply into a wall outlet. Windows automatically recognizes the NI 292x device.

Synchronizing NI 292x Devices (Optional)

You can connect two USRP devices so that they share clocks and the Ethernet connection to the host.

1. Connect the MIMO cable included with the kit to the MIMO EXPANSION port of each device.



2. If you have not already done so, attach antennas to the USRP devices.

   If you want to use one USRP device as a receiver and the other as a transmitter, attach one antenna to the RX 1 TX 1 port of the transmitter, and attach another antenna to the RX 2 port of the receiver.

The NI-USRP driver ships with some examples that you can use to explore the MIMO connection, including USRP EX Rx Multiple Synchronized Inputs (MIMO Expansion).vi and USRP EX Tx Multiple Synchronized Outputs (MIMO Expansion).vi.

Configuring NI 292x Devices

Setting Up the Network (Ethernet Only)

The device communicates with a host computer over gigabit Ethernet. Set up the network to enable communication with the device.

Note The IP addresses for the host computer and each connected USRP device must be unique.

Configuring the Host Ethernet Interface with a Static IP Address

The default IP address for the NI 292x is `192.168.10.2`.

1. Ensure the host computer uses a static IP address.

   You may need to modify the network settings for the local area connection using the Control Panel on the host computer. Specify the static IP address in the **Properties** page for Internet Protocol Version 4 (TCP/IPv4).
2. Configure the host Ethernet interface with a static IP address on the same subnet as the connected device to enable communication, as shown in the following table.

Table 1. Static IP Addresses

Component	Address
Host Ethernet interface static IP address	`192.168.10.1`
Host Ethernet interface subnet mask	`255.255.255.0`
Default USRP device IP address	`192.168.10.2`

Note NI-USRP uses user datagram protocol (UDP) broadcast packets to locate the device. On some systems, the firewall blocks UDP broadcast packets. NI recommends that you change or disable the firewall settings to allow communication with the device.

Changing the IP Address

To change the USRP device IP address, you must know the current address of the device, and you must configure the network.

1. Verify that your device is powered on and connected to your computer using the gigabit Ethernet interface.
2. Select **Start»All Programs»National Instruments»NI-USRP»NI-USRP Configuration Utility** to open the NI-USRP Configuration Utility.
3. Select the **Devices** tab of the utility.

   Your device should appear in the list on the left side of the tab.
4. In the list, select the device for which you want to change the IP address.

   If you have multiple devices, verify that you selected the correct device.

The IP address of the selected device displays in the **Selected IP Address** textbox.

5. Enter the new IP address for the device in the **New IP Address** textbox.
6. Click the **Change IP Address** button or press <Enter> to change the IP address.

   The IP address of the selected device displays in the **Selected IP Address** textbox.
7. The utility prompts you to confirm your selection. Click **OK** if your selection is correct; otherwise, click **Cancel**.
8. The utility displays a confirmation to indicate the process is complete. Click **OK**.
9. Power cycle the device to apply the changes.
10. After you change the IP address, you must power cycle the device and click **Refresh Devices List** in the utility to update the list of devices.

Confirming Network Connection

1. Select **Start»All Programs»National Instruments»NI-USRP»NI-USRP Configuration Utility** to open the NI-USRP Configuration Utility.
2. Select the **Devices** tab of the utility.

   Your device should appear in the list on the left side of the tab.

   Note If your device is not listed, verify that your device is powered on and correctly connected, then click the **Refresh Devices List** button to scan for USRP devices.

Configuring Multiple Devices with Ethernet

You can connect multiple devices in the following ways:

- Multiple Ethernet interfaces—One device for each interface
- Single Ethernet interface—One device connected to the interface, with additional devices connected using an optional MIMO cable
- Single Ethernet interface—Multiple devices connected to an unmanaged switch

Tip Sharing a single gigabit Ethernet interface among devices may reduce overall signal throughput. For maximum signal throughput, NI recommends that you connect no more than one device per Ethernet interface.

Multiple Ethernet Interfaces

To configure multiple devices connected to separate gigabit Ethernet interfaces, assign each Ethernet interface a separate subnet, and assign the corresponding device an address in that subnet, as shown in the following table.

Table 2. Multiple Host Ethernet Interface Configuration

Device	Host IP Address	Host Subnet Mask	Device IP Address
USRP Device 0	`192.168.10.1`	`255.255.255.0`	`192.168.10.2`
USRP Device 1	`192.168.11.1`	`255.255.255.0`	`192.168.11.2`

Single Ethernet Interface—One Device

You can configure multiple devices using a single host Ethernet interface when the devices are connected to each other using a MIMO cable.

1. Assign each device a separate IP address in the subnet of the host Ethernet interface, as shown in the following table.

Table 3. Single Host Ethernet Interface—MIMO Configuration

Device	Host IP Address	Host Subnet Mask	Device IP Address
USRP Device 0	`192.168.10.1`	`255.255.255.0`	`192.168.10.2`
USRP Device 1	`192.168.10.1`	`255.255.255.0`	`192.168.10.3`

2. Connect Device 0 to the Ethernet interface and connect Device 1 to Device 0 using a MIMO cable.

Single Ethernet Interface—Multiple Devices Connected to an Unmanaged Switch

You can connect multiple USRP devices to a host computer through an unmanaged gigabit Ethernet switch that allows a single gigabit Ethernet adapter on the computer to interface with multiple USRP devices connected to the switch.

Assign the host Ethernet interface a subnet, and assign each device an address in that subnet, as shown in the following table.

Table 4. Single Host Ethernet Interface—Unmanaged Switch Configuration

Device	Host IP Address	Host Subnet Mask	Device IP Address
USRP Device 0	`192.168.10.1`	`255.255.255.0`	`192.168.10.2`
USRP Device 1	`192.168.10.1`	`255.255.255.0`	`192.168.10.3`

Programming the NI 292x

You can use the NI-USRP instrument driver to create communications applications for the NI 292x.

NI-USRP Instrument Driver

NI-USRP features a set of VIs and properties that exercise the functionality of the NI 292x, including configuration, control, and other device-specific functions. Refer to the *NI-USRP Help* for information about using the instrument driver in your applications.

NI-USRP Examples

The instrument driver examples are instructional tools that demonstrate some of the functionality of the NI 292x. You can use these examples separately or integrate them into your systems. NI-USRP includes examples for getting started and other SDR functionality. You can access the NI-USRP examples from the following locations:

- From the Start menu at **Start»All Programs»National Instruments»NI-USRP» Examples**.
- In LabVIEW from **Functions»Instrument I/O»Instrument Drivers»NI-USRP» Examples** palette.

You can access additional examples from the code sharing community at `ni.com/usrp`.

Note The NI Example Finder does not include NI-USRP examples.

Testing the Device (Optional)

Perform a loopback test to confirm that the device transmits and receives signals and is connected correctly to the host computer.

1. Attach the included 30 dB attenuator to one end of the SMA (m)-to-SMA (m) cable.
2. Connect the 30 dB attenuator to the RX 2 TX 2 connector on the front panel of the USRP device, and connect the other end of the SMA (m)-to-SMA (m) cable to the RX 1 TX 1 port.
3. On the host computer, open the niUSRP EX Tx Continuous Async example VI and run it.

   If the device is transmitting signals, the IQ graph displays I and Q waveforms.
4. Open the niUSRP EX Rx Continuous Async example VI and run it.

   If the device is receiving signals, the IQ graph displays I and Q waveforms.

Troubleshooting

If an issue persists after you complete a troubleshooting procedure, contact NI technical support or visit *ni.com/support*.

Device Troubleshooting

Should I Update Device Firmware and FPGA Images?

NI 292x devices ship with firmware and FPGA images compatible with NI-USRP driver software. You may need to update the device for compatibility with the latest version of the software.

When you use the NI-USRP API, a default FPGA loads from persistent storage on the device.

The driver software media also includes the NI-USRP Configuration Utility, which you can use to update the devices.

Updating NI 292x Firmware and FPGA Images (Optional)

The firmware and FPGA images for NI 292x devices are stored in the device internal memory. You can reload the FPGA image or firmware image using the NI-USRP Configuration Utility and an Ethernet connection, but you cannot create custom FPGA images using the Ethernet connection.

1. If you have not already done so, connect the host computer to the device using the Ethernet port.
2. Select **Start»All Programs»National Instruments»NI-USRP»NI-USRP Configuration Utility** to open the NI-USRP Configuration Utility.
3. Select the **N2xx/NI-29xx Image Updater** tab. The utility automatically populates the **Firmware Image** and **FPGA Image** fields with the paths to the default firmware and FPGA image files. If you want to use different files, click the **Browse** button next to the file you want to change, and navigate to the file you want to use.
4. Verify that the firmware and FPGA image paths are entered correctly.
5. Click the **Refresh Device List** button to scan for USRP devices and update the device list.

   If your device does not appear in the list, verify that the device is on and is correctly connected to the computer.

   If your device still does not appear in the list, you can manually add the device to the list. Click the **Manually Add Device** button, enter the IP address of the device in the dialog box that displays, and click **OK**.
6. Select the device to update from the device list, and verify that you selected the correct device.
7. Verify that the version of the FPGA image file matches the board revision for the device you are updating.
8. To update the device, click the **WRITE IMAGES** button.
9. A confirmation dialog box displays. Confirm your selections and click **OK** to continue.

   A progress bar indicates the status of the update.
10. When the update completes, a dialog box prompts you to reset the device. A device reset applies the new images to the device. Click **OK** to reset the device.

    Note The utility is unresponsive while it verifies that the device reset correctly.
11. Close the utility.

Related Information

Refer to the Load the images onto the on-board flash (USRP-N Series only) section of the UHD - USRP2 and N Series Application Notes

Why Doesn't the USRP Device Appear in MAX?

MAX does not support NI 292x devices. Use the NI-USRP Configuration Utility instead. Open the NI-USRP Configuration Utility from the Start menu at **Start»All Programs» National Instruments»NI-USRP»NI-USRP Configuration Utility**.

Why Doesn't the Device Power On?

Check the power supply by substituting a different adapter.

Why Doesn't the USRP Device Appear in the NI-USRP Configuration Utility?

1. Check the connection between the USRP device and the computer.
2. Ensure that the USRP device is connected to a computer with a gigabit-compatible Ethernet adapter.
3. Ensure that a static IP address of `192.168.10.1` is assigned to the adapter in your computer.
4. Allow up to 15 seconds for the device to completely start up.

In the NI-USRP Configuration Utility, Why Does USRP2 Appear Instead of NI 292x?

An incorrect IP address on the computer may cause this error. Check the IP address and run the NI-USRP Configuration Utility again.

An old FPGA or firmware image on the device may also cause this error. Upgrade the FPGA and firmware using the NI-USRP Configuration Utility.

Why Don't NI-USRP Examples Appear in the NI Example Finder?

NI-USRP does not install examples into the NI Example Finder.

You can access the NI-USRP examples from the following locations:

- From the Start menu at **Start»All Programs»National Instruments»NI-USRP» Examples**.
- In LabVIEW from **Functions»Instrument I/O»Instrument Drivers»NI-USRP» Examples** palette.

Network Troubleshooting

Why Doesn't the Device Respond to a Ping (ICMP Echo Request)?

The device should reply to an internet control message protocol (ICMP) echo request. To ping the device, open a Windows command prompt and enter `ping` *`192.168.10.2`*, where *`192.168.10.2`* is the IP address for your USRP device. If you do not receive a response, verify that the host network interface card is set to a static IP address corresponding to the same subnet as the IP address of the corresponding device. Also verify that the device IP address is set properly.

Related Information

Changing the IP Address on page 6

Why Doesn't the NI-USRP Configuration Utility Return a Listing for My Device?

If the NI-USRP Configuration Utility does not return a listing for your device, search for a specific IP address.

1. Navigate to `<Program Files>\National Instruments\NI-USRP\`.
2. <Shift>-right-click the `utilities` folder, and select **Open command window here** from the shortcut menu to open a Windows command prompt.
3. Enter `uhd_find_devices --args=addr=`*`192.168.10.2`* in the command prompt, where *`192.168.10.2`* is the IP address for your USRP device.
4. Press <Enter>.

   If the `uhd_find_devices` command does not return the listing for your device, the firewall may be blocking replies to UDP broadcast packets. Windows installs and enables a firewall by default. To allow UDP communication with a device, disable any firewall software associated with the network interface for the device.

Why Doesn't the Device IP Address Reset to the Default?

If you cannot reset the default device IP address, your device may be on a different subnet than the host network adapter. You can power cycle the device in a safe (read-only) image, which sets the device to the default IP address of `192.168.10.2`.

1. Open the device enclosure, making sure to take appropriate static precautions.
2. Locate the safe-mode button, a push-button switch (S2), inside the enclosure.
3. Press and hold the safe-mode button while you power cycle the device.
4. Continue to press the safe-mode button until the front panel LEDs blink and remain solid.
5. While in safe-mode, run the NI-USRP Configuration Utility to change the IP address from the default, `192.168.10.2`, to a new value.
6. Power cycle the device without holding the safe-mode button to return the normal mode.

Note NI recommends that you use a dedicated network with no other USRP devices connected to the host computer to avoid the possibility of an IP address conflict. Also, verify that the static IP address of the host network adapter on the computer that runs the NI-USRP Configuration Utility is different from the device default IP address of `192.168.10.2` and different from the new IP address to which you want to set the device.

Note If the device IP address is on a different subnet from the host network adapter, the host system and configuration utility cannot communicate with and configure the device. For example, the utility recognizes, but cannot configure a device with an IP address of `192.168.11.2` connected to a host network adapter with a static IP address of `192.168.10.1` and a subnet mask of `255.255.255.0`. To communicate with and configure the device, change the host network adapter to a static IP address on the same subnet as the device, such as 192.168.11.1, or change the subnet mask of the host network adapter to recognize a wider range of IP addresses, such as 255.255.0.0.

Related Information

Changing the IP Address on page 6

Why Does the Device Not Connect to the Host Interface?

The host Ethernet interface must be a gigabit Ethernet interface to connect to the USRP device.

Ensure the connection between the host network interface card and the device cable connection is valid and both the device and computer are powered on.

A lit green LED in the upper left corner of the gigabit Ethernet connection port on the device front panel indicates a gigabit Ethernet connection.

Front Panels and Connectors

Direct Connections to the NI 292x

The NI 292x is a precision RF instrument that is sensitive to ESD and transients. Ensure you take the following precautions when making direct connections to the NI 292x to avoid damaging the device.

Caution Apply external signals only while the NI 292x is powered on. Applying external signals while the device is powered off may cause damage.

- Ensure you are properly grounded when manipulating cables or antennas connected to the NI 292x TX 1 RX 1 or RX 2 connector.
- If you are using noninsulated devices, such as a noninsulated RF antenna, ensure the devices are maintained in a static-free environment.
- If you are using an active device, such as a preamplifier or switch routed to the NI 292x TX 1 RX 1 or RX 2 connector, ensure that there are no signal transients greater than the RF and DC specifications for the device that are being generated and sourced to the NI 292x TX 1 RX 1 or RX 2 connector.

NI USRP-2920

Figure 1. NI USRP-2920 Front Panel

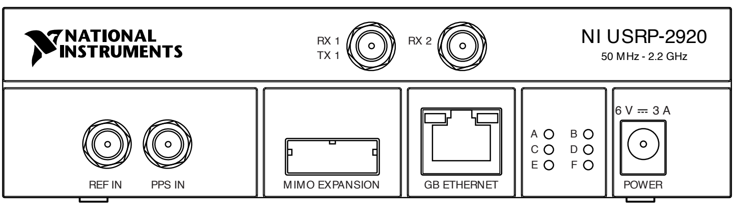


Table 5. NI USRP-2920 Module Front Panel Connectors

Connector	Use
RX 1 TX 1	Input and output terminal for the RF signal. RX 1 TX 1 is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input or output channel.
RX 2	Input terminal for the RF signal. RX 2 is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input channel.
REF IN	Input terminal for an external reference signal for the local oscillator (LO) on the device. REF IN is an SMA (f) connector with an impedance of 50 Ω and is a single-ended reference input. REF IN accepts a 10 MHz signal with a minimum input power of 0 dBm (.632 Vpk-pk) and a maximum input power of 15 dBm (3.56 Vpk-pk) for a square wave or sine wave.
PPS IN	Input terminal for the pulse per second (PPS) timing reference. PPS IN is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input. PPS IN accepts 0 V to 3.3 V TTL and 0 V to 5 V TTL signals.
MIMO EXPANSION	The MIMO EXPANSION interface port connects two USRP devices using a compatible MIMO cable.
GB ETHERNET	The gigabit Ethernet port accepts an RJ-45 connector and gigabit Ethernet compatible cable (Category 5, Category 5e, or Category 6).
POWER	The power input accepts a 6 V, 3 A external DC power connector.

Table 6. NI USRP-2920 Module LEDs

LED	Indication
A	Indicates the transmit status of the module: OFF—The module is not transmitting data. GREEN—The module is transmitting data.
B	Indicates the status of the physical MIMO cable link: OFF—The modules are not connected using the MIMO cable. GREEN—The modules are connected using the MIMO cable.

Table 6. NI USRP-2920 Module LEDs (Continued)

LED	Indication
C	Indicates the receive status of the module: OFF—The module is not receiving data. GREEN—The module is receiving data.
D	Indicates the firmware status of the module: OFF—The firmware is not loaded. GREEN—The firmware is loaded.
E	Indicates the reference lock status of the LO on the module: OFF—There is no reference signal, or the LO is not locked to a reference signal. BLINKING—The LO is not locked to a reference signal. GREEN—The LO is locked to a reference signal.
F	Indicates the power status of the module: OFF—The module is powered off. GREEN—The module is powered on.

NI USRP-2921

Figure 2. NI USRP-2921 Front Panel

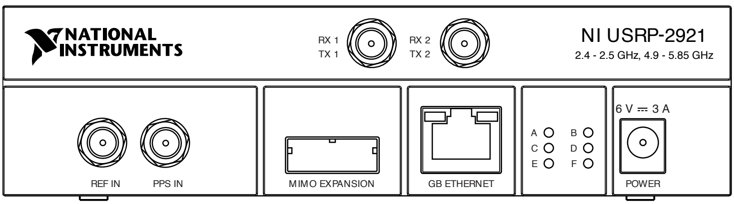


Table 7. NI USRP-2921 Module Front Panel Connectors

Connector	Use
RX 1 TX 1	Input and output terminal for the RF signal. RX 1 TX 1 is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input or output channel.
RX 2	Input terminal for the RF signal. RX 2 is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input channel.

Table 7. NI USRP-2921 Module Front Panel Connectors (Continued)

Connector	Use
REF IN	Input terminal for an external reference signal for the local oscillator (LO) on the device. REF IN is an SMA (f) connector with an impedance of 50 Ω and is a single-ended reference input. REF IN accepts a 10 MHz signal with a minimum input power of 0 dBm (.632 Vpk-pk) and a maximum input power of 15 dBm (3.56 Vpk-pk) for a square wave or sine wave.
PPS IN	Input terminal for the pulse per second (PPS) timing reference. PPS IN is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input. PPS IN accepts 0 V to 3.3 V TTL and 0 V to 5 V TTL signals.
MIMO EXPANSION	The MIMO EXPANSION interface port connects two USRP devices using a compatible MIMO cable.
GB ETHERNET	The gigabit Ethernet port accepts an RJ-45 connector and gigabit Ethernet compatible cable (Category 5, Category 5e, or Category 6).
POWER	The power input accepts a 6 V, 3 A external DC power connector.

Table 8. NI USRP-2921 Module LEDs

LED	Indication
A	Indicates the transmit status of the module: OFF—The module is not transmitting data. GREEN—The module is transmitting data.
B	Indicates the status of the physical MIMO cable link: OFF—The modules are not connected using the MIMO cable. GREEN—The modules are connected using the MIMO cable.
C	Indicates the receive status of the module: OFF—The module is not receiving data. GREEN—The module is receiving data.
D	Indicates the firmware status of the module: OFF—The firmware is not loaded. GREEN—The firmware is loaded.

Table 8. NI USRP-2921 Module LEDs (Continued)

LED	Indication
E	Indicates the reference lock status of the LO on the module: OFF—There is no reference signal, or the LO is not locked to a reference signal. BLINKING—The LO is not locked to a reference signal. GREEN—The LO is locked to a reference signal.
F	Indicates the power status of the module: OFF—The module is powered off. GREEN—The module is powered on.

NI USRP-2922

Figure 3. NI USRP-2922 Front Panel

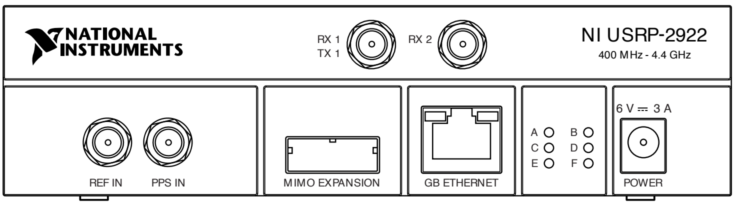


Table 9. NI USRP-2922 Module Front Panel Connectors

Connector	Use
RX 1 TX 1	Input and output terminal for the RF signal. RX 1 TX 1 is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input or output channel.
RX 2	Input terminal for the RF signal. RX 2 is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input channel.
REF IN	Input terminal for an external reference signal for the local oscillator (LO) on the device. REF IN is an SMA (f) connector with an impedance of 50 Ω and is a single-ended reference input. REF IN accepts a 10 MHz signal with a minimum input power of 0 dBm (.632 Vpk-pk) and a maximum input power of 15 dBm (3.56 Vpk-pk) for a square wave or sine wave.

Table 9. NI USRP-2922 Module Front Panel Connectors (Continued)

Connector	Use
PPS IN	Input terminal for the pulse per second (PPS) timing reference. PPS IN is an SMA (f) connector with an impedance of 50 Ω and is a single-ended input. PPS IN accepts 0 V to 3.3 V TTL and 0 V to 5 V TTL signals.
MIMO EXPANSION	The MIMO EXPANSION interface port connects two USRP devices using a compatible MIMO cable.
GB ETHERNET	The gigabit Ethernet port accepts an RJ-45 connector and gigabit Ethernet compatible cable (Category 5, Category 5e, or Category 6).
POWER	The power input accepts a 6 V, 3 A external DC power connector.

Table 10. NI USRP-2922 Module LEDs

LED	Indication
A	Indicates the transmit status of the module: OFF—The module is not transmitting data. GREEN—The module is transmitting data.
B	Indicates the status of the physical MIMO cable link: OFF—The modules are not connected using the MIMO cable. GREEN—The modules are connected using the MIMO cable.
C	Indicates the receive status of the module: OFF—The module is not receiving data. GREEN—The module is receiving data.
D	Indicates the firmware status of the module: OFF—The firmware is not loaded. GREEN—The firmware is loaded.

Table 10. NI USRP-2922 Module LEDs (Continued)

LED	Indication
E	Indicates the reference lock status of the LO on the module: OFF—There is no reference signal, or the LO is not locked to a reference signal. BLINKING—The LO is not locked to a reference signal. GREEN—The LO is locked to a reference signal.
F	Indicates the power status of the module: OFF—The module is powered off. GREEN—The module is powered on.