Open Source Software Defined Radio Platform for GNSS Recording and Simulation

Alison Brown, Jarrett Redd, and Michael Dix, *NAVSYS Corporation*

BIOGRAPHY

Alison Brown is the President and Chief Executive Officer of NAVSYS Corporation, which she founded in 1986. Dr. Brown has a PhD in Mechanics, Aerospace, and Nuclear Engineering from UCLA, an MS in Aeronautics and Astronautics from MIT, and an MA and BA in Engineering from Cambridge University. She is a fellow of the Institute of Navigation and an Honorary Fellow of Sidney Sussex College, Cambridge.

Jarrett Redd is a Senior Hardware Engineer at NAVSYS Corporation. He holds a BS in Computer Science with a minor in Mechanical Engineering from Texas A&M University. He holds five patents.

Michael Dix is a Product and Marketing Manager at NAVSYS Corporation. He holds an MS Forest Resources Conservation with a concentration in Geomatics and a BA in Business Administration from University of Florida.

ABSTRACT

The Software Defined Radio (SDR) platform continues to shape the Global Navigation Satellite System (GNSS) testing and development environment. Open source SDR platforms are now being developed and offer GNSS development capability to a variety of users that have not previously had sufficient resources to be engaged. An open source platform also offers many advantages in terms of customization and system development.

NAVSYS has integrated GNSS waveforms into SDRs such as the Ettus Research™ USRP™ N210 and the Racelogic LabSat devices for use as inexpensive GNSS signal simulators. In this paper we show how an open source SDR platform can be used with GNU Radio open source architecture for building a GNSS signal simulator. We will also show how this can be integrated with NAVSYS GNSS Signal Architect® Simulator Software for a turnkey system to carry out multi-frequency/multi-code generation with the option for users to update and modify the signal generation software.

INTRODUCTION

Commercially, most GNSS signal generators and record/playback systems are designed specifically for high volume production test applications for devices that use commercial GPS, SBAS, GLONASS, Galileo and other GNSS receivers[1]. The quality of these commercial systems has continued to increase with the addition of new GNSS signals and features.[2] Even with all of the flexibility provided by these commercial systems, there is still a need for full access to GNSS signal simulation environments for some GNSS developers to generate their own customized simulation scenarios.

To meet the need for flexible radio hardware, the open source community continues to emerge and contribute to the advancement of open source SDR technology. Much of the advancement is focused towards SDR platforms that allow users of all skill levels to continually create, test, and develop radio capabilities which can include GNSS receiver technology.

One of the major advancements within this community has been the GNU Radio software architecture. In this paper we show the steps involved for using GNU Radio compatible SDRs and the GNU Radio open source software with NAVSYS' Signal Architect Simulator Software[3] to provide a flexible, inexpensive GNSS signal simulation capability with the capability to generate customized signal simulations.

STEP 1: INSTALL GNU RADIO COMPANION (GRC) OPEN SOURCE SOFTWARE

GNU Radio[4] is a free and open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available, low-cost external RF hardware to create SDRs or without hardware in a simulation-like environment. GNU Radio has been adopted in both academic and industry laboratory environments.

Within the GNU Radio architecture, the GRC is a graphical tool used by developers to easily create signal flow graphs and to generate flow-graph source code. As shown in Figure 1, GRC allows a user to program an SDR to transmit and receive RF information and is capable of being customized for a variety of applications.

To run GRC, a user must have either a Windows or Linux-based computer with sufficient signal processing bandwidth for disk, Ethernet, USB, CPU, and memory components to handle the real time data flow required. The GCR software provides installation instructions[5].

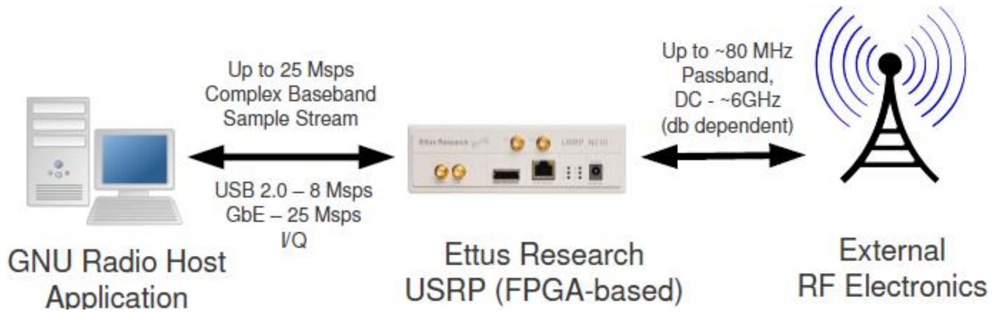


Figure 1 GRC Integration Overview

STEP 2: PURCHASE GRC COMPATIBLE SDR THAT COVERS GNSS FREQUENCY BANDS

In order to generate the GNSS signals, it is necessary to have an SDR that is capable of covering the range of GNSS frequencies required for a particular project. Figure 2 shows the primary GNSS frequency bands and can be used to compare against SDR specifications.

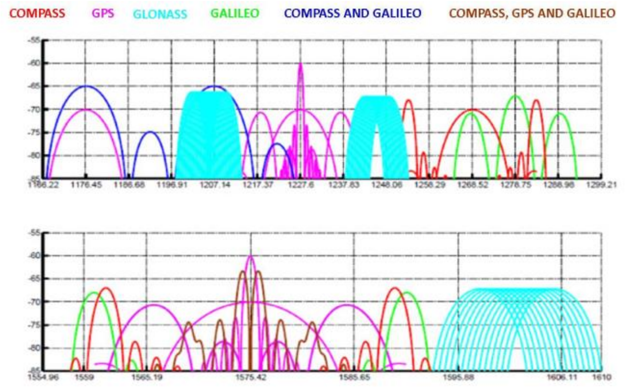


Figure 2 GNSS Frequency Bands

Examples of two SDR platforms that are compatible with GNU Radio and compatible for use for GNSS simulation applications is the Ettus USRP family[6] or the bladeRF[7] SDR built by Nuand.

Ettus provides a family of USRP products with different features (frequency ranges, sample rates, multi-output capability). The most inexpensive product currently offered that could be used for GNSS signal simulation is the USRP Instant SDR Kit[8] which starts at $675 including a USRP B100 SDR with a WBX Daugterboard and a LiveUSB SDR Environment with drivers, software and development tools on a bootable USB drive. USRP models provide a dynamic development platform to record and playback GNSS signals in a static or mobile environment.

The Ettus radio supports a wideband transceiver front-end that can accommodate up to the full 25 MHz of the GPS signal band and can be tuned to operate at any of the GPS signal frequencies (L1:1575.42 MHz, L2:1227.60 MHz or L5: 1176.45 MHz). This allows record and playback of both the civil and military GPS codes and also other GNSS satellite signals using the GRC tools.

The USRP provides an interface between high speed analog to digital converters and an Ethernet or USB interface to record or playback the I/Q binary data from the 16-bit analog-digital converters. Daughterboards available for the USRP provide conversion between the baseband signals present at the data converters and the GNSS frequency bands.

A GNSS signal simulator configuration requires a transceiver daughterboard to be installed in the USRP radio, such as the WBX. The tunable range of the WBX (50 MHz to 2.2 GHz) covers all the current GNSS frequencies. An RF pre-filter is used to band-limit the GNSS signals to the sample rate selected for use by the GNU Radio software to avoid spectral folding from the USRP channel bandwidth. For example, a 2 MHz filter centered at L1 is optimal based on the Nyquist sampling frequency of 2 MHz I/Q. If sampling at 20 MHz, then a 20 MHz pre-filter should be used[1].

Another compatible commercially available SDR platform is the bladeRF. It began as a project on the popular crowdfunding website Kickstarter in January 2013. With an initial funding goal of $100,000, the project received almost twice that amount with 501 backers[9].

The bladeRF is billed as “A low-cost, open source USB 3.0 Software Defined Radio platform with many examples and tutorials to help you experiment with RF,” and the goal was to provide a dynamic and accessible SDR environment for a variety of users. NAVSYS was one of the early backers of this project to support the open source development of this SDR technology for GNSS record and playback applications. The bladeRF provides a full open architecture design for full duplex 28MHz channels which capture the GNSS frequencies in its 300MHz – 3.8GHz range. It contains a modular expansion board design for adding GPIO, Ethernet, and 1PPS synchronization. Nuand also publishes full hardware schematics and configuration for the bladeRF. This increases the ability for users to customize the SDR platform and its capabilities.

STEP 3: CONFIGURE GRC TO RECORD AND PLAYBACK GNSS SIGNALS INTO GPS USER EQUIPMENT (UE)

The objective of this step is to use the GRC tools to configure the SDR for recording a sequence of the GNSS I/Q signal spectrum and then playing these signals back through the SDR transceiver into a GNSS receiver. This step verifies that the SDR connections and settings have been correctly configured and that the GNSS receiver will acquire and track on signals recorded and played back through the SDR.

To configure GRC, you will need to have installed the GRC software (Step 1) and also the associated SDR control software for the chosen hardware (Step 2) in accordance with the SDR vendor's instructions. Once these steps are complete, the SDR control software can be launched to confirm that the radio is connected properly and can be appropriately controlled by the computer. As always, input and output power requirements and restrictions must be properly respected to avoid damage. The manufacturer's manual should be consulted for all hardware involved to ensure the connections and power flow are managed properly.

The record and playback of the GNSS signals are accomplished by executing the following steps for generating and executing a GRC flow graph.

The first step is *Flow Graph Generation* which is started by launching GRC and creating a new and empty flow graph using the GUI tools.

An example flow graph can be seen in Figure 3. This particular flow graph provides for record and playback, as well as multi-band operation. The exact configuration shown is for dual band recording, whereas the example below is for single band recording.

For this example, we will create a "record" or flow graph that receives a signal using the SDR and writes the data to a file for storage.

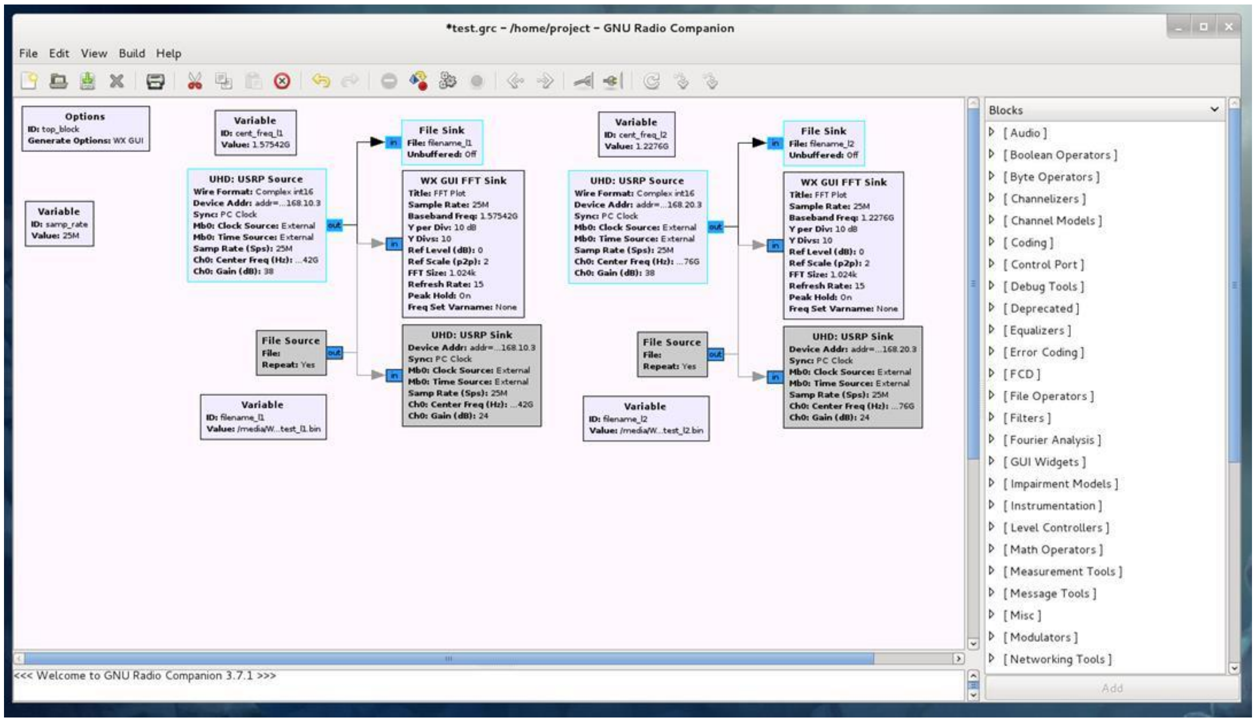
1. Add an "SDR Source" widget or plugin to the flow graph. This widget is called a "source" widget because it serves as a source of data for the system. Note that this widget would be specific to the SDR you are using and would connect to the driver for that SDR. In the case of an Ettus USRP N210 SDR, the widget would be called "USRP Source".
2. Next, add a "File Sink" widget to the flow graph and point it at a file where you wish to store the data. This widget is called a "sink" widget because it serves as a sink (destination or end point) for the data in the system. Refer to the GRC manual if needed.
3. Next, connect an arrow between the output on the source widget and the input on the sink widget. This will allow GRC to flow data between the two widgets, allowing the signal from your SDR to be delivered in real time to the disk for storage in the specified file. If you are having trouble connecting the arrow, then there is probably a mismatch between the data format in the file and the expected data format required by the SDR. If this is the case, there is a library of data conversion widgets available in GRC, or you can write your own. Look through them and place them on the screen in between the source and sink, as required.

It is important to note that you will probably need to configure your source widget (and thereby your SDR) to match the parameters of the signal that you wish to record; e.g., center frequency, sample rate, etc. You may also wish to add a Fast Fourier Transform graph, "FFT Graph", or other similar visualization widget to your flow graph. This should be placed between the source and sink and will allow you to visualize in real time the spectrum being received by the radio. Again, there are numerous configurations on this widget and it must be made to match the parameters of the rest of the system. Similarly, filters or other real time data manipulation widgets can also be added, as desired. Any GRC supported components can be used here. Save the flow graph when everything is complete.

Flow Graph Execution in Record

After verifying that all cable connections and hardware/software settings are correct, you can start the flow graph by clicking on the start button. Continuing the "single band record" example from above, once the process is started, the SDR will begin receiving and data will begin flowing from the SDR to the disk.

In this case, an antenna is connected to the radio to receive the signal of interest, so input power requirements are probably not a concern, but refer to the manufacturer's instructions to be certain. However, the given GNSS antenna is probably active and will need power injection. This can be provided using the power injection block provided with the antenna, if any, but often the antenna relies on power injected by a GNSS receiver, so you will need to provide it manually by use of a bias-T or similar device. Note that it is important to protect the SDR from this injected power by using a DC block or similar on the input.



Options
ID: top_block
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 25M

Variable
ID: cent_freq_l1
Value: 1.57542G

UHD: USRP Source
Wire Format: Complex int16
Device Addr: addr=...168.10.3
Sync: PC Clock
Mb0: Clock Source: External
Mb0: Time Source: External
Samp Rate (Sps): 25M
Ch0: Center Freq (Hz): ...42G
Ch0: Gain (dB): 38

File Sink
File: filename_l1
Unbuffered: Off

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 25M
Baseband Freq: 1.57542G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Peak Hold: On
Freq Set Varname: None

File Source
File:
Repeat: Yes

UHD: USRP Sink
Device Addr: addr=...168.10.3
Sync: PC Clock
Mb0: Clock Source: External
Mb0: Time Source: External
Samp Rate (Sps): 25M
Ch0: Center Freq (Hz): ...42G
Ch0: Gain (dB): 24

Variable
ID: filename_l1
Value: /media/W...test_l1.bin

Variable
ID: cent_freq_l2
Value: 1.2276G

UHD: USRP Source
Wire Format: Complex int16
Device Addr: addr=...168.20.3
Sync: PC Clock
Mb0: Clock Source: External
Mb0: Time Source: External
Samp Rate (Sps): 25M
Ch0: Center Freq (Hz): ...76G
Ch0: Gain (dB): 38

File Sink
File: filename_l2
Unbuffered: Off

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 25M
Baseband Freq: 1.2276G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Peak Hold: On
Freq Set Varname: None

File Source
File:
Repeat: Yes

UHD: USRP Sink
Device Addr: addr=...168.20.3
Sync: PC Clock
Mb0: Clock Source: External
Mb0: Time Source: External
Samp Rate (Sps): 25M
Ch0: Center Freq (Hz): ...76G
Ch0: Gain (dB): 24

Variable
ID: filename_l2
Value: /media/W...test_l2.bin

```
<<< Welcome to GNU Radio Companion 3.7.1 >>>
```

Figure 3 Example Record Flow Graph

When the desired elapsed recording time has completed, the execution can be halted by clicking on the stop button. This will stop the disk writing, and for most SDR plugins, GRC will stop the radio-receive operation as well, but be aware that it is possible for the radio to be occasionally left in a receiving state. In this case, the radio will need to be manually halted.

While some SDRs have a maximum transmit duty cycle and continuous transmission can result in overheating or damage to the hardware, this is rarely the case with receive operations and the SDR can usually be run continuously in this mode for hours or even days.

In the steps below we describe the opposite example: a "single band playback" or transmission flow graph that takes a data file that sends the data to an SDR for transmission.

1. Add a "File Source" widget to the flow graph and point it at a file you have created. Refer to the GRC manual if needed. Point it to the file containing the signal data that you wish to playback.
2. Next, add an "SDR Sink" widget or plugin to the flow graph. Note that this widget would be specific to the SDR you are using and would connect to the driver for that SDR. In the case of an Ettus USRP N210 SDR, the widget would be called "USRP Sink".
3. Next, connect an arrow between the output on the source widget and the input on the sink widget to allow the GRC to flow data between the two widgets allowing the data in the BIN file to be delivered in real time to your SDR. If you are having trouble connecting the arrow, it is probably caused by a mismatch between the data format in the file and the expected data format that the SDR requires. If this is the case, check the GRC's library of available data conversation widgets (or you can write your own), and place the correct widget on the screen in between the source and sink, as required.

You will probably also need to configure your sink widget (and thereby your SDR) to match the parameters of the scenario that you generated. Again, all data visualization and manipulation tools provided by GRC can be used here. Save the flow graph when everything is complete.

After verifying that all cable connections and hardware/software settings are correct, you can start ***Flow Graph Execution*** by clicking on the start button. Continuing the "playback" example from above, once the process is started, data will begin flowing to the SDR and the SDR will transmit the data. Transmission typically involves much higher power levels than reception, so make sure to review the power requirements per the manufacturer's instructions to avoid damaging the equipment. For example, most GPS UE are expecting to receive a signal from an antenna, and hence, of a very low power. SDRs transmit at various powers and can easily overpower and damage GPS UE, so the signal should be properly attenuated.

In some cases, the SDR sink widget may automatically repeat the file when it reaches the end. This generally will not work well for GNSS receivers, so you would want to disable this feature. Some GNSS receivers may have trouble computing a navigation solution for your GNSS scenario file if the time of the scenario differs greatly from the receiver time. This is particularly troublesome if the scenario is in the past (as far as the receiver is concerned). To prevent this, either reset the receiver to factory defaults (usually receiver time will default to the start of the epoch or 1980 or similar) or manually set the receiver time near the scenario time.

The simulation can be halted by clicking on the stop button or by simply allowing the file to play to the end. For most SDR plugins, GRC will stop the radio transmission as well, but be aware that it is possible for the radio to be occasionally left in a transmitting state. In this case, the radio will need to be manually halted.

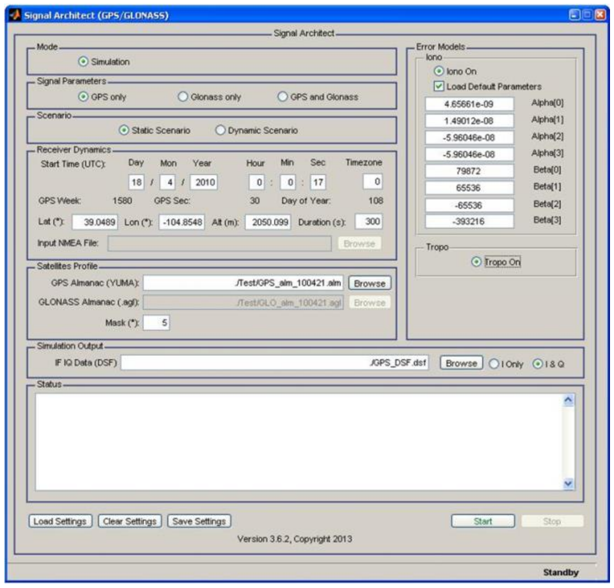
It is important to note that some SDRs have a maximum transmit duty cycle and continuous transmission can result in overheating or damage to the hardware. If this is the case with your SDR, be careful when running for extended lengths or when leaving the SDR transmitting after your simulation has completed.

Lastly, some SDRs have sensitive electronics that can be damaged by connecting or disconnecting cables while the radio is on. This is especially true while the radio is transmitting. Make sure to shut down the radio before reconfiguring setups. Also be aware that reflected power (VSWR) can be an issue in some setups, resulting in hardware damage over time, if excessive.

STEP 4: BUILD SIMULATED GRC FILES USING SIGNAL ARCHITECT AND PLAYBACK TO GNSS UE

The Signal Architect Simulator Software developed by NAVSYS was designed for creating customized GNSS scenarios for playback with an open source SDR platform. The software interface, pictured in Figure 4, allows the user to specify static or dynamic scenarios, load appropriate almanac files, and apply atmospheric correction models among other settings. The simulated scenario is defined using either an input NMEA (recorded during real world testing or generated by simulation) or KML file (generated by Google Earth or similar). The satellite signals generated are defined through the satellite almanac data loaded as a Yuma Almanac file and the specified user mask angle.

The Signal Architect software operates by generating recorded I/Q data files in the same format as generated by the GRC record signal flow graph. The software includes multi-threading capabilities to take advantage of high speed multi-core processors and reduce the time for I/Q data file generation. These binary I/Q data files can then be replayed back into the SDR through GRC as described in Step 3 to generate simulated GNSS signal scenarios as shown in Figure 5. The base software includes a compiled version of the full NAVSYS' GNSS Signal Architect Toolbox that includes the capability for L1 GPS signal generation. As described in the following section, the user can upgrade to purchasing the full GNSS Signal Architect Toolbox which allows for editing and control of the low level I/Q generation for the GRC files. Other optional features also allow for generation of different GNSS signal codes and frequencies.


Signal Architect (GPS/GLONASS)

Signal Architect

Mode: [✓ Simulation]

Signal Parameters: [✓ GPS only] Glonass only GPS and Glonass

Scenario: [✓ Static Scenario] Dynamic Scenario

Receiver Dynamics
Start Time (UTC): Day 18 / Mon 4 / Year 2010 Hour 0 : Min 0 : Sec 17 Timezone 0
GPS Week: 1580 GPS Sec: 30 Day of Year: 108
Lat (°): 39.0489 Lon (°): -104.8548 Alt (m): 2050.099 Duration (s): 300
Input NMEA File:

Satellites Profile
GPS Almanac (YUMA): ./Test/GPS_alm_100421.alm
GLONASS Almanac (.agl): ./Test/GLO_alm_100421.agl
Mask (°): 5

Simulation Output
IF IQ Data (DSF) ./GPS_DSF.dsf I Only [✓ I & Q]

Error Models
Iono
[✓ Iono On]
[✓ Load Default Parameters]

4.65661e-09	Alpha[0]
1.49012e-08	Alpha[1]
-5.96046e-08	Alpha[2]
-5.96046e-08	Alpha[3]
79872	Beta[0]
65536	Beta[1]
-65536	Beta[2]
-393216	Beta[3]

Tropo
[✓ Tropo On]

Status

Version 3.6.2, Copyright 2013

Standby

Figure 4 Signal Architect Simulator Software

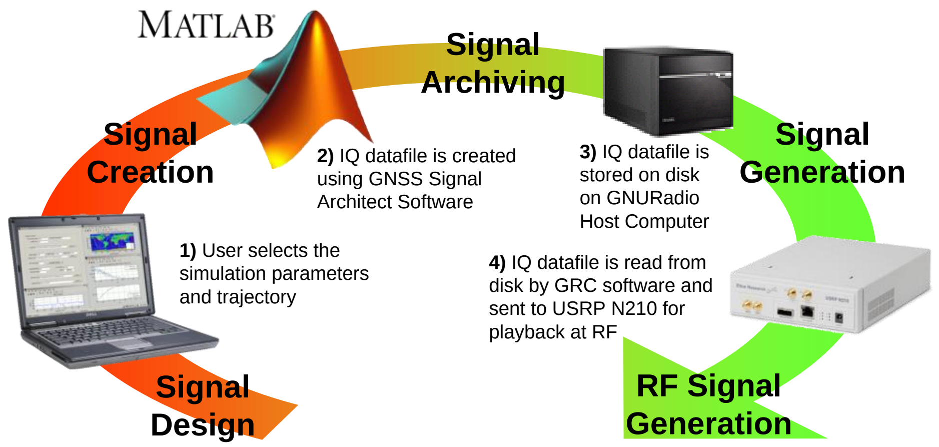

MATLAB
Signal Archiving
Signal Creation
Signal Generation
2) IQ datafile is created using GNSS Signal Architect Software
3) IQ datafile is stored on disk on GNURadio Host Computer
1) User selects the simulation parameters and trajectory
4) IQ datafile is read from disk by GRC software and sent to USRP N210 for playback at RF
Signal Design
RF Signal Generation

Figure 5 GNSS Signal Architect Operation

SIGNAL ARCHITECT OPTIONAL UPGRADES
The base version of the GNSS Signal Architect Simulation Software allows the user to interface with any GNU Radio capable SDR to create static or dynamic L1 scenarios. The GNU Radio software also allows for real-time RF signal visualization during record and playback.

Upgrades to the Signal Architect software include the following features:

- Multiband Expansion Software (recommended for multi-frequency generation)
- GLONASS (civil codes)
- GPS Military P+M' (L1 and L2).
- Other GNSS upgrades including BeiDou, Galileo and QZSS are in development.
- GNSS Signal Architect Toolbox include M-file source code for GNSS signal generation

Figure 6 and Figure 7 show example system diagrams for complete framework integration for single-band or multi-band operation.

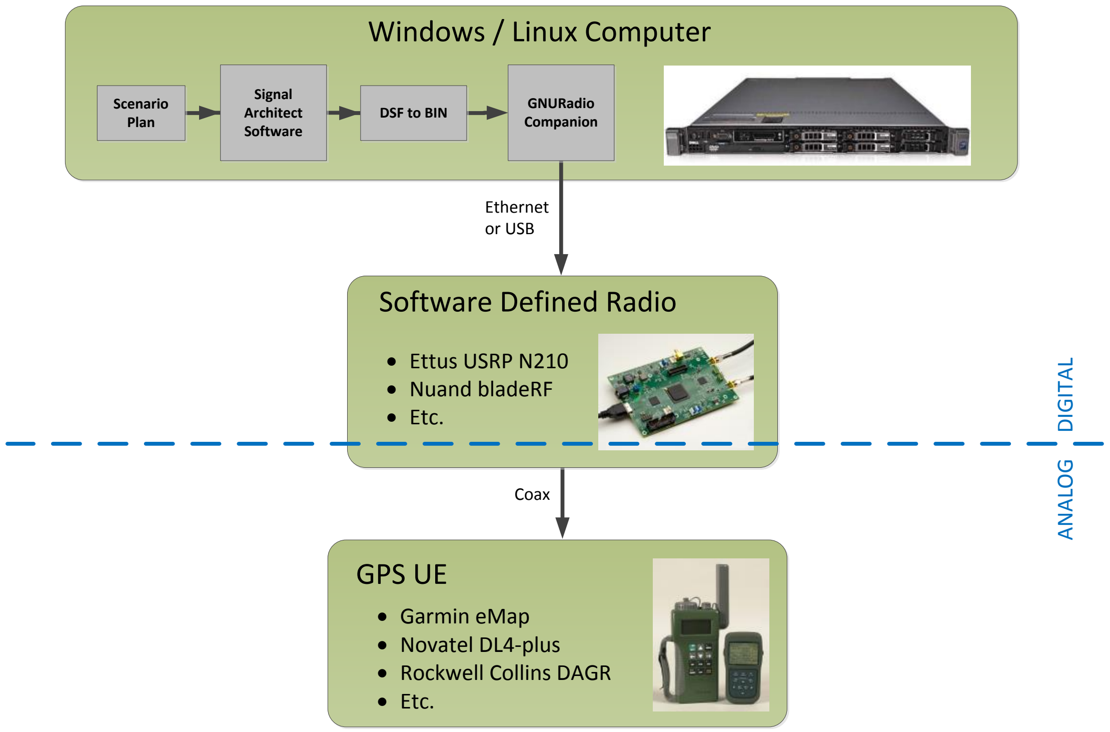


Figure 6 Single Band with GNU Radio Companion

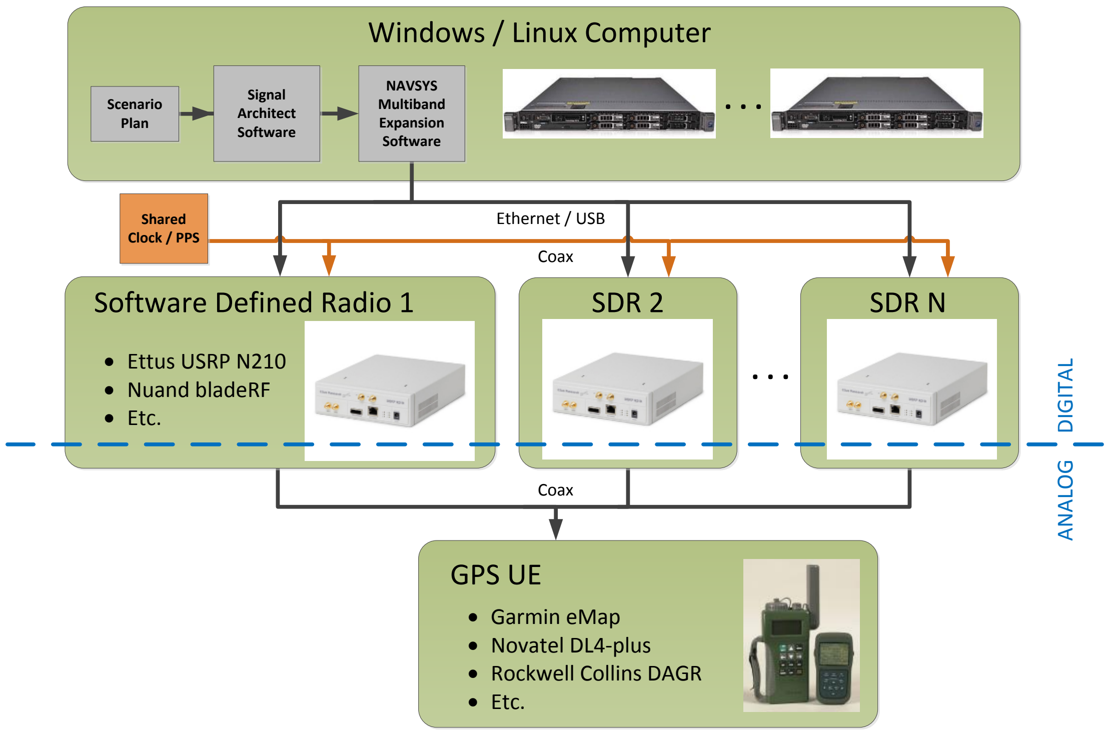


Figure 7 Multi Band with NAVSYS Multiband Expansion Software

Multiband Signal Generation Approach
Described below are two approaches for recording or replaying multiple frequency bands simultaneously. The first approach is using GNU Radio and MIMO (Multiple Input Multiple Output) architecture. This simple set-up is fully within the GRC functionality, requires all data to flow through a single computer (either Windows or Linux), requires a special cable to split the clock and 1 PPS signals for synchronization purposes, and is generally limited to two SDR's with gigabit Ethernet limiting data to approximately 25 Msps, 16 bit I+Q.

The second approach requires an upgrade to include the NAVSYS Multiband Expansion Software. This software has the advantages of compressing the multi-stream I/Q data allowing for playback of simultaneous GNSS signals into multiple SDRs from multiple host computers to speed data generation and playback (see Figure 7). The Multiband Expansion Software option operates using a command line interface to synchronize playback from an unlimited number of computers (currently Linux only) to share the data load, and playback I/Q data files into an unlimited number of SDR's with full per-radio bandwidth and DAC bit compression to reduce Ethernet bandwidth usage. With the Multiband Expansion Software, the user must manually split the clock and 1 PPS signals to synchronize the SDR RF playback signals.

Figure 8 shows two Ettus USRP SDR's integrated with a GPS receiver for dual record and playback. In this example using the MIMO (25Msps) set-up, the left radio is configured at L1, 4Msps (C/A) and the right radio at L2, 20Msps (P) with dual band GRC output directed over Ethernet to each radio. If the NAVSYS Multiband Expansion Software set-up is used, the left radiois configured for playback at L1, 25 Msps (C/A+P+M') and the right radio: L2, 25 Msps (P+M') with two single band I/Q outputs directed over Ethernet to each radio from multiple host computers. These host computers could even be located in different physical locations, if desired. Figure 9 shows single-band (L1-C/A pictured left) and dual-band (L1-C/A and GLONASS pictured right) signals, respectively, during playback.

Simulated Jammer Option
Another capability for the GNSS signal simulator product allows the user to insert real time simulated or pre-generated/pre-recorded interference signals into the GNSS signals during playback. This is helpful for testing GNSS receivers in a simulated interference environment. The flexibility of the design allows the user to select the interference signal type (e.g. CW, narrow-band, broadband or swept) and also provide high fidelity adjustment of the jammer/signal levels. A partial simplified flow graph for this is shown in Figure 10 below.

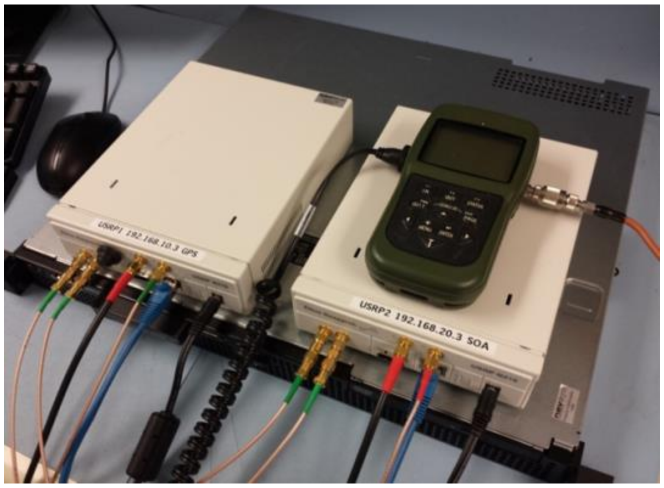


Figure 8 Dual Radio Integration

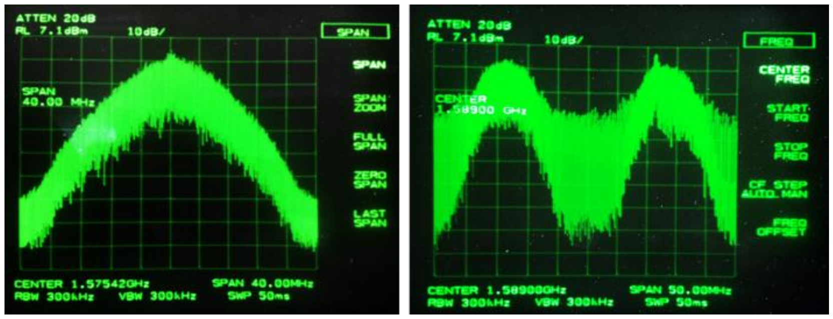
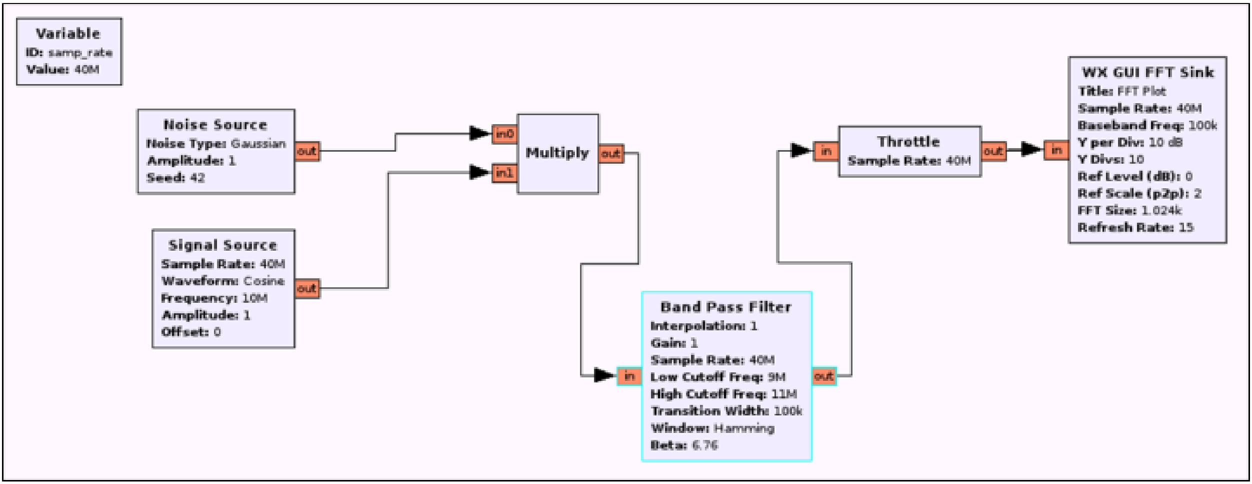


Figure 9 Single (left) and Dual (right) Band Signals



Variable
ID: samp_rate
Value: 40M

Noise Source
Noise Type: Gaussian
Amplitude: 1
Seed: 42

Signal Source
Sample Rate: 40M
Waveform: Cosine
Frequency: 10M
Amplitude: 1
Offset: 0

Multiply

Band Pass Filter
Interpolation: 1
Gain: 1
Sample Rate: 40M
Low Cutoff Freq: 9M
High Cutoff Freq: 11M
Transition Width: 100k
Window: Hamming
Beta: 6.76

Throttle
Sample Rate: 40M

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 40M
Baseband Freq: 100k
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15

Figure 10 Flow Graph for Simulated Jammer

SUMMARY

In this paper we have shown how to configure inexpensive open source SDR platforms for effective GNSS signal generation. Using the GNU Radio architecture, a user can create intuitive signal flow generation for record and playback of GNSS signals using the GNU Radio Companion interface with compatible Software Defined Radios. The open source GRC tools allow for user customization of the signal capture and playback capability. When combined with the NAVSYS Signal Architect Simulator Software, a complete multi-frequency/multi-code generation capability is provided with flexible upgrade options for advanced simulations. With the optional GNSS Signal Architect Toolbox upgrade, the user can fully customize the GNSS signal generation using the M-file MATLAB source code included in the Toolbox.

REFERENCES

1 Alison Brown, Reece Tredway, and Robert Taylor, “GPS Signal Simulation using Open Source GPS Receiver Platform, Proceedings of the 21st Virginia Tech Symposium on Wireless Personal Communications, Blacksburg, Virginia, June 2011, http://www.navsys.com/Papers/11-06-02%20GPS%20Signal%20Simulation%20using%20Open%20Source%20GPS%20Receiver%20Platform.pdf

2 2013 Simulator Buyers Guide, GPS World, May 1, 2013. http://gpsworld.com/2013-simulator-buyers-guide/

3 GNSS Signal Architect Product Suite http://www.navsys.com/Products/signal_architect_product_suite.htm

4 GNU Radio http://gnuradio.org/redmine/projects/gnuradio/wiki#Welcome-to-GNU-Radio

5 http://gnuradio.org/redmine/projects/gnuradio/wiki/InstallingGR

6 Ettus Research Products https://www.ettus.com/product

7 bladeRF - the USB 3.0 Superspeed Software Defined Radio (http://nuand.com/)

8 USRP Instant SDR Kit https://www.ettus.com/product/details/UB100D-BDL

9 http://www.kickstarter.com/projects/1085541682/bladerf-usb-30-software-defined-radio