

John Petrich, W7FU

1629 East Lake Sammamish Pkwy NE, Sammamish, WA 98074;
petrich@u.washington.edu

Tom McDermott, N5EG

3950 Southview Ter, Medford, OR 97504;
n5eg@arrl.net

Digital Signal Processing and GNU Radio Companion

The authors present an easy way to include DSP in your SDR radio projects.

Understanding the fundamental identity of the analog and digital signal flow processes makes it easy to apply practical knowledge and hands on Amateur Radio experience to author digital signal processing (DSP) functions using the GNU Radio DSP library with the user friendly graphical interface, GNU Radio Companion (GNU Radio Companion). Does authoring your own DSP programs for your next SDR radio project, interest you? Would you like to develop DSP applications that permit real-time transmit and receive capability as well as simulation ability? Are the thoughts of learning a new programing language and coping with the complexity of DSP intimidating? Would a free and easy to learn DSP software development environment be something you'd want to learn more about? If so, the open source DSP software library, GNU Radio, with the companion graphical user interface, GNU Radio Companion, may be the application for you.

How Complex is DSP?

DSP can be simplified and demystified on a fundamental level by comparing analog and digital signal processing. The signal flow logic of a DSP system parallels the logic of an analog signal flow. Both use similar processing units. Multipliers (mixers), multiply-accumulate (filters), multiplication (amplifiers), and I/O devices, are used in the same flow order to process signals. In the analog realm, a simple, classic analog super heterodyne receiver architecture might involve inputting a signal to an analog mixer, which in turn converts the signal flow to an intermediate frequency (IF). That IF signal is filtered in an analog filter, and then flows to circuits to be detected/mixed to the audio range with output to a speaker. A DSP

system signal flow is constructed in the same manner using digitally implemented mixers (multipliers), followed by digitally implemented filters (multiply-accumulate), digitally implemented demodulators, and so on. The DSP signals are rendered in the digital domain and processed at baseband (at or near "0" Hz).

GNU Radio and the Companion Graphical User Interface

With GNU Radio Companion, the user's focus is on rendering the DSP via the graphical overlay rather than having to write code and deal with algorithms and data management issues. With GNU Radio Companion, the user has access to the full functional capabilities of the GNU Radio DSP software library and the ability to author DSP systems. GNU Radio is an open source software library made up of DSP signal processing units written in *Python* and *C++* code.¹ GNU Radio Companion is the graphical overlay on top of the foundational GNU Radio code.² With GNU Radio Companion, it is not necessary to learn these foundational codes. The user has access to the full functional capabilities of GNU Radio using only the GNU Radio Companion graphical overlay. A DSP system is comprised of a collection of signal processing units. These units are simply stages in processing a radio signal, rendered in the digital domain. The GNU Radio Companion user connects various signal processing units, rendered as graphical blocks, to author a DSP system. Different blocks and/or arrangements of blocks represent different algorithms and different

DSP signal flows. Underneath the GNU Radio Companion overlay, GNU Radio automates the real-time nature of DSP, the handling of buffers, timing, multi-threading, and other software tasks.

The Building Blocks of a DSP System Using GNU Radio Companion

The same familiar analog mixers, filters, audio, I/O connections, and so on, rendered in the digital domain, are examples of DSP process units. It is the collection of these process units to form a system that enables the DSP to process information. To author a DSP system, the digital domain signal processing units are sequenced. To build a radio, the sequence of DSP units is connected to an SDR front end and forms the back end of a practical Software Defined Radio.

Within GNU Radio Companion, the foundational signal processing units of GNU Radio are graphically rendered as rectangular boxes with I/O ports, and are called "blocks." There are more than 300 DSP blocks available in the GNU Radio Companion DSP library, which can be used to author DSP projects. These blocks are collected into 46 broad library categories.

A selection of blocks that have recognizable applications for typical Amateur Radio DSP projects are presented in Table 1. The list does not exhaust the range of possible blocks of interest to the creative user. An explanation of Table 1 will help the user to understand the GNU Radio library organization. The first column in Table 1 lists some typical DSP processes, such as a receiver input or GUI controls. The middle column lists the GNU Radio Companion names of the DSP blocks and

¹Notes appear on page 46.

Table 1**Selected DSP processes and corresponding GNU Radio Companion blocks**

DSP process	GNU Radio Companion block	GNU Radio Companion library category
Receiver input	FUNcube Dongle, RTL 2832 TV Dongle, Hermes NB SDRstick™, Ettus UHD	FCD, RTL, HPSDR, source UHD
Audio input	Audio source	Audio
Transmitter output	UHD sink, Hermes NB	UHD, HPSDR
Audio output	Audio sink	Audio
Data recording	File sink	Audio
Filtering	Bandpass, Low pass, High pass, Band Reject	Filter
Mod/De-mod	Rx or TX: WBFM, NBFM, AM, PSK, and others	Modulators
GUI display/testing	FFT, Histogram, scope, waterfall, and others	Instrumentation
Networking	TCP, UDP source and sink	Networking Tools
Resampling	Rational and Fractional Resampler	Filters
GUI controls	Radio button, Slider, Tab, and others	Widgets
DSP flow control	Valve, Selector, Null source	Misc., Source

Parameters:

<u>ID</u>	low_pass_filter_0
<u>FIR Type</u>	Float->Float (Decimating)
<u>Decimation</u>	1
<u>Gain</u>	1
<u>Sample Rate</u>	samp_rate
<u>Cutoff Freq</u>	6e3
<u>Transition Width</u>	1e3
<u>Window</u>	Hamming
<u>Beta</u>	6.76

Error Messages:

Source - out(0):
Port is not connected.

Sink - in(0):
Port is not connected.

Documentation:

This filter is a convenience wrapper for an fir filter and a firdes taps generating function.

Sample rate, cutoff frequency, and transition width are in Hertz.

The beta parameter only applies to the Kaiser window.

Figure 1 — Low pass filter parameter box.

the right hand column lists the GNU Radio Companion library category in which the particular DSP blocks are collected. The top few rows of Table 1 list the signal sources (such as front ends of SDR receivers) and signal outputs or sinks (such as front ends of SDR transmitters). The list of supported sources and sinks include popular SDR front ends such as the FUNcube Dongle, RTL 2832 TV Dongle, SDRstick™, Hermes/Metis, and the family of Ettus transceivers.^{3,4}

^{5,6} Going down the rows:

- The Modulators library category is a collection of standalone, drop in, DSP units for either receiving or transmitting AM, FM, PSK and other modulation modes.
- The Instrumentation library category contains blocks that find dual use as tuning aids in the operational GUI for a radio application and use as trouble shooting tools in the development of DSP projects.
- The Networking Tools category contains

blocks to network real world radios with local and wide area networks, a rapidly evolving aspect of radio communications technology.

- The Resampling process is a type of DSP process that permits changing the sample rate of a signal, increasing (interpolating) or decreasing (decimating) the sample rate. These Resampler blocks permit control of sample rates for individual DSP blocks including filters, real world I/O devices, and instrumentation displays.

- Widgets provide the GUI controls for real world operational interfaces. These controls include radio buttons, sliders, and GUI tabs, among others.

- The Miscellaneous category contains selector, valve and null source blocks, which make for practical software control of transmit and receive functions within the SDR transceiver flow graph.

Table 1 illustrates that each signal processing unit performs a limited amount of signal processing and that the blocks must be linked to perform more complex DSP processes. The function of each block is determined by the underlying mathematical algorithm for that DSP process. The algorithm contained in each block is individually programmable. The user programs the DSP block with the mathematical parameters (math variables) that define the desired processing.

An example of user configurable GNU Radio Companion block parameters is depicted in Figure 1, a parameter box for a DSP low pass filter block. The parameter box is brought on screen by double clicking the block of interest. This low pass filter block has been programmed to process a floating point digital connection between blocks. The Cutoff Frequency — bandwidth in this case — is user defined as 6 kHz. The Transition Width — slope of the filter roll off — was chosen to be 1 kHz. Sample Rate, Decimation, and Gain are also user configurable, depending on the particular DSP process being authored.

How Does the GNU Radio Companion User Create DSP Systems in Practice?

The GNU Radio Companion graphical interface makes the authoring of DSP systems easy and approachable. Using GNU Radio Companion terminology, the total DSP signal flow system, graphically rendered by arrow linked blocks, is termed the “flow graph.” The flow graph constitutes the totality of the DSP that has been authored. To construct a DSP system using GNU Radio Companion, one moves the desired signal processing block(s) from the on screen library, on the right of the screen, into the work space in the center of the screen. The desired parameters are entered into each block. The ports of all of the signal processing blocks are linked serially with arrow link connections by left clicking the input and output ports of the blocks.

Figure 2 depicts the flow graph of a GNU Radio Companion DSP that generates a 1 kHz audio tone output to the user’s computer speaker. A Waveform source block, on the left side of the work area, is connected with an arrow to the Audio sink block (also known as a computer sound card), on the right side of the work area. The user-entered audio frequency parameter is displayed in the Waveform source block as a 1000 Hz cosine. The user selected sample rate parameter for the sound card is displayed in the audio sink block as 48000 Hz. The “Options” box in the upper left corner of Figure 2 identifies the flow graph file name for display purposes and is not a GNU Radio Companion DSP block. The smaller box, labeled as a “Variable” block, is one way to specify the system sample rate as 48000 Hz.

Flow graphs of this type are visually simple and facilitate an intuitive understanding of the DSP system function.

The math variables that define the process, entered as parameters, are displayed within each block. Trouble shooting in a GNU Radio Companion flow graph is as easy as following the signal flow logic illustrated by the arrows, and by visualizing whether the parameters displayed in each block are correct. GNU Radio Companion provides automatic error messages if ports are not properly connected and if parameters are not entered correctly. GNU Radio Companion also contains a category of instrument tool blocks such as FFT, oscilloscope, constellation and auto correlation displays that can be used to troubleshoot and optimize the DSP flow graph. The combination of the GNU Radio Companion visualization of the signal processing blocks, the automatic error messages and the included instrument blocks facilitate the rapid layout and validation of your DSP flow graph.

Step By Step Guide to Get GNU Radio With GNU Radio Companion Up and Running

GNU Radio is a continuously evolving open source DSP software library with GNU Radio Companion (GRC) as the graphical user interface. The entire DSP library and graphical interface package is designated by the GRC version number. Installation of a GNU Radio Companion version automatically installs the GNU Radio DSP software library that supports the GNU Radio Companion interface.

To get started with GNU Radio Companion, there are four decisions:

- 1) The computer hardware platform,
- 2) The operating system (OS),
- 3) The GNU Radio Companion installation method,
- 4) Deciding which version of GNU Radio Companion to install

Computer Hardware Requirements

Consider your intended use of GNU Radio Companion to decide how much compute capability you need. Basic learning with simple DSP systems is easy with inexpensive computer systems such as a single-core, 2 GB computers, and are used successfully by the authors as starting points. These computers are fine for learning and demonstration projects, though the process of down loading and installation of GNU Radio Companion versions with this level of computer can take several hours.

If your goals are to develop more elaborate DSP systems where real time performance is important, and to easily keep up with GNU Radio Companion version updates, a high performance computer is a wise choice. Dual and quad core CPUs with clock speeds above 2.5 GHz, and 4 GB or more of RAM are preferred for more elaborate DSP development, such as transceiver DSPs. High performance transceivers implemented in GNU Radio Companion typically consume less than 2 GB of memory, and involve no page file swaps. With this level of computer the installation of the GNU Radio Companion package can be completed in less than an hour.

Choosing A Computer Operating System (OS)

Linux Ubuntu is the preferred OS, versions 12.04 and above, with the most recent Ubuntu release 14.04 LTS preferred for new installations. It is a small step to learn *Linux* Ubuntu and implement GNU Radio Companion in a *Linux* environment. GNU Radio Companion can be installed on *Linux* Ubuntu, *Windows* OS and *Mac* OS, but non-*Linux* installs seem plagued by problems, judging from Forum comments, though some *Windows* and *Mac* users do claim success.⁷ Experienced *Windows* and *Mac* OS users

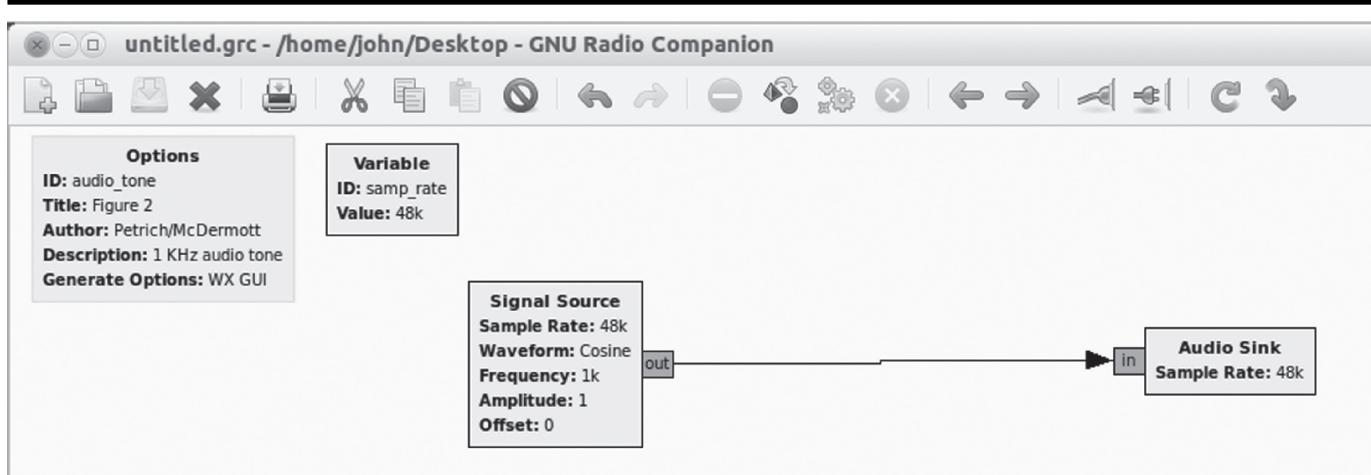


Figure 2 — GNU Radio Companion DSP Flow Graph: The signal source is a 1 kHz audio tone, which is connected to the audio sink (output) to a computer speaker.

may have good results at managing the GNU Radio Companion installation process, while less skilled users are probably wise to take the path of least resistance and choose the *Linux* Ubuntu OS.

Some users are hesitant to learn a new operating system. These concerns are greater than they need be if the user allows for an initial period of adjustment and occasional reference to Ubuntu help resources.⁸ The application installation in *Linux* is quite simple and easily mastered. The installation of GNU Radio Companion requires some basic familiarity with a handful of simple commands in the *Linux* Ubuntu terminal application (see Note 8). In the end, *Linux* Ubuntu has the look and feel of a late model *Windows* operating system. Experienced *Windows* users find *Linux* Ubuntu as logical, business like, and as easy to understand as *Windows*. *Linux* offers the same right click functionality as *Windows*, which greatly simplifies file manipulation. “Right click is your friend” in *Linux*.

Choosing a GNU Radio Installation Method

The third decision involves how to install the *Linux* Ubuntu based GNU Radio Companion application. Fortunately, several possibilities exist. Table 1 shows some common options. “Boot and go” is a great approach for those with limited computer skills, or for those who want to quickly install GNU Radio Companion and experiment. Bootable USB 2.0 memory sticks and DVD discs are available and provide both the Ubuntu operating system and the GNU Radio Companion application in one simple package. The “boot and go” approach, allows the user to run GNU Radio Companion in the *Linux* Ubuntu operating system and avoid making any changes to the host computer hard drive. The “live” GNU Radio Companion / *Linux* bootable media limit you to certain versions of Ubuntu and GNU Radio Companion.

To “boot and go”, you insert the media

into the appropriate USB port or DVD drive, and boot into *Linux* Ubuntu from that media. Once booted into *Linux*, the user can execute GNU Radio Companion or use various available Ubuntu applications such as an e-mail client or Internet browser. Shutting down the computer closes the *Linux* operating system. On restart, the computer will automatically default to the native operating system, unless commanded by the user to again boot from the media into *Linux*. A downside is the reduction in performance due to the limited speed of the USB or DVD data bus. As a consequence, the Ubuntu boot process and the execution of any applications are somewhat slower than what one expects of the same process(s) using a standard hard drive (HD). With “boot and go,” the user cannot save DSP files. Despite these limits, the “boot and go” method makes it very easy to quickly get started with GNU Radio Companion and delay decisions regarding alternative GNU Radio Companion installation approaches.

Alternatively, Ubuntu *Linux* and GNU Radio Companion can be installed directly on the user’s computer hard drive. A hard drive installation is very efficient, and uses the computer’s native processing performance. A dual boot installation offers the flexibility of changing the operating system as needed. There are numerous resources, on-line and text books that will help with this process. The text by Helmke and Graner (see Note 8) is a very helpful resource and is recommended for those new to *Linux* Ubuntu. The book is reasonably priced (\$23 paperback, \$15 Kindle), explains the *Linux* Ubuntu operating system, and the command line terminal application. The paperback book also includes an installation CD of the latest Ubuntu 14.04 LTS operating system software, along with a helpful hard drive partition utility. With that book, and others like it, a trouble free dual boot installation of *Linux* Ubuntu and *Windows* operating systems is easily accomplished. A free on-line source of Ubuntu 14.04 LTS installation software is

available and works well.⁹

Choosing a GNU Radio Companion Version

GNU Radio is automatically embedded with GNU Radio Companion when one installs any GNU Radio Companion version. A GNU Radio Companion version choice is a decision for those whose goals are to fully exploit the potential of GNU Radio Companion and prefer the GNU Radio Companion installation to occur to their hard drive —a choice not applicable to the “boot and go” approach.

Typical of open source software applications, GNU Radio and GNU Radio Companion are in a state of continual evolution. The current version 3.7.3 at the time of this writing has become quite stable. Blocks written for older versions of GNU Radio (3.6 and earlier) may not function in 3.7, but most of the older blocks have now been updated for 3.7 compatibility.

For a beginning user installing the latest version GNU Radio Companion, 3.7.3 is the best choice. Installing the earlier stable version, GNU Radio Companion 3.6.5.1, allows the user to easily exploit the reservoir of previously published GNU Radio Companion DSP projects that can be found on the Internet. All GNU Radio Companion versions are available via free downloads. Table 2 shows how to find and install the latest and legacy versions of GNU Radio Companion.

Adding DSP Blocks to the GRC DSP Library

GNU Radio Companion versions, even the latest version GRC 3.7.3 doesn’t include all of the blocks available for the GNU Radio DSP software library. The DSP library of GNU Radio can be updated without changing GNU Radio Companion versions. Of particular interest to Amateur Radio operators, all GNU Radio Companion versions natively include many common radio interface blocks in the GNU Radio DSP library. The radio interface blocks

Table2:
Methods to install Linux Ubuntu and GNU Radio Companion

Install Method	Boot from:	Where to find:	Comments:
“Boot and Go”	DVD and USB versions. Can boot on large range of computer hardware. Allows user to use a Windows or Mac computer to experiment in <i>Linux</i> without altering the hard drive.	DVD: http://gnuradio.org/redmine/news/36 USB: https://www.ettus.com/product/details/LIVEUSB	Slow operation due to reduced media data rates. Can only save designs to the boot media.
Native <i>Linux</i> and GNU Radio (GRC)	Install <i>Linux</i> and GNU Radio (GRC) to hard drive	Linux Ubuntu: www.ubuntu.com/download/desktop/install-ubuntu-desktop GNU Radio Companion: http://sbrac.org/files/build-gnuradio	Requires learning at least a little <i>Linux</i> . (See text)

are used to link actual SDR hardware into GNU Radio DSP, to build a complete SDR. Updates can include radio interface DSP blocks not natively installed in older GNU Radio Companion versions or new radio interface blocks as new SDR front ends and interfaces become available. In GNU Radio language, these uninstalled blocks are referred to as “Out of Tree” (OOT) blocks. The term “OOT blocks” refers to blocks not automatically included in the GNU Radio DSP software library. Installation details are typically published with the blocks. Examples of new radio interface blocks are the TV Dongle, Hermes NB, the SDRstick blocks, and Whitebox, which interface GNU Radio with the TV Dongle, HPSDR Hermes/Metis, SDRstick™, and Whitebox SDR front ends, respectively.^{10, 11, 12} Other DSP blocks and package add-ons to the GNU Radio library can be updated as well.

To install new blocks in GNU Radio Companion, first locate and download the block. Installation is completed with a short series of commands in the *Linux* terminal window to install the block into the GNU Radio library. Locating a new block is as simple as locating the block repository on the Internet. Table 3 lists some repositories for additional blocks.

Step-By-Step Procedure for Installing Ubuntu *Linux*

The following procedure is one method to install Ubuntu 14.04 LTS on a PC. Note that this may irretrievably replace the existing operating system.

1) Download and un-compress Ubuntu 14.04 LTS .iso to the desktop from this site: www.ubuntu.com/download/desktop/install-ubuntu-desktop.

2) Create a boot media by following the instructions on the website.

3) Boot your computer from the boot media. Access the boot menu on most PCs by selecting F-12 at the splash screen.

4) Select “Install Ubuntu” and follow the on-screen directions to complete the installation. The boot medium contains a

user friendly partition utility. Selecting 30 to 50 GB of hard drive space for Ubuntu is sufficient.

5) After the installation is complete, remove the boot media and configure the operating system following the on-screen directions.

Step-By-Step Procedure for Installing GNU Radio

The following procedure is recommended as one reliable method to install GRC3.7.3 and above into Ubuntu 14.04. The procedure consists of five steps:

1) This step (step 1) is not necessary if you have not previously attempted to install GNU Radio Companion onto your hard drive. Otherwise you will need to remove prior GNU Radio Companion installations and residuals, including “distribution” versions downloaded via the Ubuntu Software Center. Even if you have “removed” GNU Radio Companion via the Software Center, follow this procedure. Copy and paste the “Gnuradio_remove” script file into your Ubuntu Home directory. http://svn.tapr.org/filedetails.php?repname=OpenHPSDR+Main&path=%2Ftrunk%2FN5EG%2FGRC3.7%2FGnuradio_remove

2) Copy and paste the latest available “build-gnuradio” script file provided by Marcus Leech (<http://sbrac.org/files/build-gnuradio>) into the Ubuntu Home directory.

3) Make sure that GNU Compiler Collection is installed.

4) Execute the “build-gnuradio” script file.

5) Open the GNU Radio Companion application and check that the application is functional.

Step 1:

(Note: \$ is the prompt character displayed by the terminal, don’t type it yourself.) Open the Terminal and at the “~\$” prompt enter:

\$ sudo apt-get purge gnuradio (and Enter)

Step 2:

1) Copy and paste the file “Gnuradio_remove” to your Home directory as a “shell script.”

2) Open your Home directory in the graphical browser.

3) Right click the file “Gnuradio_remove” and select “Properties”. A popup will open.

4) Select the “Permissions” tab.

5) Select the check-mark “Allow executing file as a program”

6) Close the popup menu then close the Home directory.

7) Open the Terminal. At the “~\$” prompt enter

~\$ sudo ./Gnuradio_remove (and Enter)
(This command instructs the operating system to execute the remove script located in the Home directory.)

Step 3:

Determine if GCC (GNU Compiler Collection) is installed on your system. Using the Terminal, at the “~\$” prompt enter:

~\$ which gcc (and Enter)

If the result is a directory listing such as: /usr/bin/gcc then GCC is installed and no further action is necessary.

If the result is a blank line, then GCC is *not* installed, and it is necessary to install it for GNU Radio Companion to build. The package “build-essential” includes the compilers, linkers, make, and cmake. In the Terminal, at the “~\$” prompt enter:

~\$ sudo apt-get install build-essential or the alternative command:

\$ sudo apt-get --reinstall install build-essential (and Enter)

Step 4:

Execute the “build-gnuradio” script from your Home directory.

At the Terminal “~\$” prompt enter:

~\$./build-gnuradio –m (and Enter)
(This command instructs the operating system to execute the build script located in the Home directory and get the latest version of GNU Radio Companion.)

(Note: The build process can take a long time depending on the capabilities of your computer. Computers at the i7 level require about 20 minutes to complete the build process. Computers at the i3 level can take several hours to complete the build and single core computers longer yet. A number of “y”

Table3
Installing GNU Radio Companion Versions From Source

Recommended Installation Steps

Download the Build GNU Radio (GRC) script, and execute it. Can take 1-8 hours depending on computer.

Install GNU Radio (GRC) 3.7.3 (latest version) (recommended)

Command to install GNU Radio (GRC)3.6.5.1 (Older version that is compatible with legacy published DSP flow graphs.)

Source

<http://sbrac.org/files/build-gnuradio>

Command(s)

Download it using web browser.

Use *Linux* Terminal command:
./build-gnuradio –m

Use *Linux* Terminal command:
./build-gnuradio –o

Table 4
Sources for Adding Selected “Out of Tree Modules” (Blocks) to the DSP Library

Out of Tree DSP Block	Location
Hermes / Metis HPSDR interface.	http://svn.tapr.org/repos_sdr_hpsdr/trunk/N5EG
SDRstick™ interface	http://svn.sdrstick.com/listing.php?repname=sdrstick-release

commands, for “yes”, are required in the Terminal during the initial stages of the build process, so pay attention.)

Step 5:

To open GNU Radio Companion after the build is compete, open the Terminal, and at “~\$” enter:

~\$ gnuradio-companion (Enter)

Step-by-Step Procedure for Installing Out-Of-Tree Modules into GNU Radio

To install additional DSP block types, not included in the official released version of GNU Radio (called ‘out-of-tree’) there are three steps:

1) Download and uncompress the source code file for the desired block into a directory (below the Home directory). In this example we will refer to the gr-hpsdr Hermes NB block. Your home directory is usually identified as ~ and we want to uncompress the complete structure into a subdirectory of ~. That complete new block will include files and even deeper directories (for example, build). Normally the uncompress program takes care of setting it all up for us. For example, you might create a new directory ~/gr-hpsdr and unzip the code into it. In Linux, directory names can include the dot character.

2) Enter a series of commands in the Terminal that will compile and install the source code into the GNU Radio DSP library. The commands are those that follow the “\$” sign in this text, for example:

~/gr-hpsdr/build/\$ make

Open GNU Radio Companion and confirm the presence of the block(s) in the library.

Step 1:

Create a new directory and check to make sure that the uncompresssed block files are present in the Home directory.

In the Terminal at “~\$” enter:

~\$ mkdir gr-hpsdr (Enter)
~\$ cd gr-hpsdr (Enter)
~/gr-hpsdr\$

Use the graphical archive manager to extract the source code into the newly created directory.

~/gr-hpsdr\$ ls (Enter)

A list of the files in your new directory will be listed, including the source directory for the block you are intending to install, in this example the directory: gr-hpsdr. It should

include some subdirectories, such as apps, build, cmake, python, and others, and the file CmakeLists.txt. Check to see if you have the build subdirectory, if not create it.

~/gr-hpsdr\$ mkdir build (Enter)

Step 2:

Open a Terminal, and at the “~\$” prompt enter the following series of commands:

~\$ cd ~/gr-hpsdr/build (Enter)

~/gr-hpsdr/build/\$ cmake .. (Enter)

[Note the space after “cmake” and before “..”]

~/gr-hpsdr/build/\$ make clean (Enter)

~/gr-hpsdr/build/\$ make (Enter)

~/gr-hpsdr/build/\$ sudo make install

(Enter)

~/gr-hpsdr/build/\$ sudo ldconfig (Enter)

~/gr-hpsdr/build/\$ exit (Enter) and the

terminal will close.

Step 3:

Open the GNU Radio Companion application and verify that the desired block is installed in the DSP library list. If you installed gr-hpsdr, for example, there would now be an hpsdr selection on the panel with all the gnuradio blocks, and inside that hpsdr selection will be the hermesNB block. Other out-of-tree modules will have other names of course.

Congratulations. You now have a working installation of GNU Radio Companion on your computer. In Part 2 of this article we will present several examples of what you can do with GNU Radio.

ARRL member, Official Observer, and Amateur Extra class licensee John Petrich, W7FU, was first licensed as K6OJV in 1955 and then as W7HQJ after moving to Seattle. He is a practicing physician, and Clinical Associate Professor of Psychiatry, School of Medicine, University of Washington. John is active in community affairs, enjoys family life, sea kayaking and cycling. John’s radio experience parallels the evolution of radio communications technology over the past century. He started with a homebrew crystal receiver followed by a much loved and modified single tube regenerative receiver in a cardboard box. Upon earning his license, he graduated to operating QRP using a crystal controlled 6V6 tube transmitter constructed on a wooden chassis. Subsequently his rigs evolved from modified surplus gear to home constructed to full featured analog receivers and transmitters. John’s first love is CW,

the prototypical digital QRP mode. He credits radio with endless opportunities for engaging learning opportunities and long lasting friendships. At present John’s rig is an experimenter’s station built around the HPSDR Atlas bus system. The station is supplemented with back-up rigs using Ettus and SDRstick™ SDR “front ends” and DSP “back ends” built using GNU Radio Companion. John is interested in communicating with others who have similar interests.

ARRL Life Member, and Amateur Extra class licensee Tom McDermott, N5EG, has been licensed 45 years. He is a member of TAPR, IEEE, and Internet2, and has been involved in the development of fiber optic transmission and switching systems since the initial deployment of single-mode fiber in positions ranging from ASIC designer to CTO. He currently is a participant in the IEEE 802.3 Ethernet 100GE and 400GE standards projects. Tom has a BSEE degree from the University of California, Berkeley, and has written one textbook on wireless digital communications. He’s been involved in many computer-related ham projects, from the TEXNET layer 3 packet radio system, to a VNA project, network simulation, and other efforts. His current interest is using a HPSDR Hermes SDR transceiver and GNU Radio to experiment with DSP algorithms.

Notes

¹The GNU Radio Wikipedia page provides more detailed information about this software package: <http://gnuradio.org/redmine/projects/gnuradio/wiki>

²The home page for GNU Radio Companion is found at: <http://gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion>

³For details about the Fun Cube Dongle, go to: www.funcubedongle.com/

⁴There is more information about the SDRstick at: <http://sdrstick.com/>

⁵The TAPR website has detailed information about the High Performance Software Defined Radio project, including the Atlas backplane and the various cards that plug into the backplane to create the radio: www.tapr.org/

⁶To learn more about Ettus Research and their universal software research peripheral (USRP) hardware visit the Ettus website: www.ettus.com/

⁷You can find a lot of information and answers to common questions on the GNU Radio forum at: <https://www.ruby-forum.com/forum/gnuradio>

⁸Matthew Helmke and Amber Graner, *The Official Ubuntu Book*, Seventh Edition, Prentice Hall, ISBN -13: 978-0-13-301760-1.

⁹You can find on-line instructions and files to install Ubuntu at: www.ubuntu.com/download/desktop/install-ubuntu-desktop

¹⁰There is more information about the Hermes and Metis hardware on the TAPR website: www.tapr.org/kits_hermes.html and www.tapr.org/kits_metis.html

¹¹For information about the SDRstick hardware, see: <http://sdrstick.com/>

¹²More information about the novel Whitebox SDR project is available at: radio.testa.co/index.html#document-faq