Created by Richard Bell 10/29/2014

Introduction

GNU Radio is open source software that provides built in modules for standard tasks of a wireless communications system. Within the GNU Radio framework is gnuradio-companion, which is a GUI that lets you connect blocks together with wires to create a data flow graph. GNU Radio can also be used as a DSP simulation tool.

If you want to use GNU Radio with hardware, such as a universal software radio peripheral (USRP), then you will need to install UHD before you install GNU Radio. The order of install is important. This provides GNU Radio with modules that allow easy interaction with the USRP.

PyBOMBS is a tool that was written to help make the GNU Radio install process easy. It lets you choose between various modules for installation, such as uhd, gnuradio, etc.

Created by Richard Bell 10/29/2014

Install UHD and GNU Radio with PyBOMBS

At the time this guide was written, the objective was to install UHD and GNU Radio v3.7.6 on Ubuntu 14.04 for use with a USRP N210. The install process uses PyBOMBS which you can read about here: http://gnuradio.org/redmine/projects/pybombs/wiki. I am following this install guide specifically and adding my own experiences as I go along http://gnuradio.org/redmine/projects/pybombs/wiki/QuickStart. You will need to have sudo admin rights on the machine you are installing to.

If you are starting with a fresh Ubuntu install, you should not have to worry about any GNU Radio or UHD files existing in the system by default. If you inherited a system, you first need to uninstall any old GNU Radio and UHD installs that might be on the system because they will interfere with this new install. To do this execute:

sudo apt-get purge gnuradio gnuradio* uhd uhd*

Check your system (using find gnuradio or find uhd or searching another way) for any leftover gnuradio files and get rid of them. If you find a build directory, which will usually contain several make files, execute:

sudo make uninstall

in the directory to remove it.

Now you need to make sure you have libboost1.54-dev and Git installed. At the time of this guide being written, you have to use libboost1.54-dev, you can't use libboost1.55-dev. You can check all your installed packages by executing:

dpkg-query -l | less

If you can't find the package installed, then install it by executing the following:

sudo apt-get install -y libboost1.54-dev

sudo apt-get install git

Now you need to clone the pybombs.git repository to your local machine and install. To do this, execute the following:

cd directory_you_want_things_installed

git clone https://github.com/pybombs/pybombs.git && cd pybombs

When prompted with questions, you can hit enter to use default selections unless you know what you're doing and you want to change something. I recommend leaving the defaults as is. Don't worry about the Git username either, it's used for people that want to commit new gnuradio code for development.

At this point you have a choice on how you want to proceed to install uhd and gnuradio. If you want to use the experimental app store, which is a GUI that lets you click modules to install, from the pybombs directory execute:

./app_store.py

Because the app store is experimental, you may notice bugs at times. If the bug so severe that it prohibits you from installing what you need, then use the next method. From the app store GUI, click on the UHD icon to install UHD. After that finishes installing, click on GNU Radio.

If you choose not to use use the app store to install modules, then you will use the command line with the pybomb shell script. To see a list of modules that can be installed, from the pybombs folder execute:

./pybombs list

When you find the name of modules you want to install, execute:

./pybombs install module_name_to_install

For example, to install UHD and GNU Radio, execute:

./pybombs install uhd

./pybombs install gnuradio

These two installs should take ~40 min. Finally you need to setup your environment variables so you can run gnuradio from the command line and find all the dependencies. To do this, cd to the pybombs folder and execute:

./pybombs env

Now add the following line to your .bashrc file (located in your home directory), or .bash_aliases file if you created one, so it executes every time you open a terminal:

source target/setup_env.sh

After closing all terminals and opening a new one, execute:

gnuradio-companion

gnuradio-companion will launch (I add an alias to my .bash_aliases file to rename this command to something shorter, like grc). The first time I did this, I got an error about scipy needing to be installed, but gnuradio-companion still opened. To overcome this error, I simply installed scipy and it resolved:

sudo apt-get install python-scipy

Now you should enable real time scheduling so gnuradio is given priority to meet timing requirements. To do this execute:

sudo gedit /etc/security/limits.conf

Add the following to the file and save:

@usrp – rtprio 50

Now you need to add the usrp group to your machine. To do that execute:

groupadd usrp

Now you need to add yourself to the usrp group. To do that execute:

sudo usermod -a -G usrp <username>

To confirm that you've been added to the usrp group, execute:

groups <username>

Now you need to log out of your account and back in for these changes to take effect. You can confirm they work by opening gnuradio-companion (execute ***gnuradio-companion***), turning on realtime scheduling (double click the options block in the flow graph and set the realtime scheduling to ON) and running the empty companion file (click the green play button at the top middle of the window). You should not see any errors in the info panel at the bottom.

Updating GNU Radio, UHD or PyBOMBS with Git

Since updates for this software are not very frequent, nor do you need to update your system every time a new version comes out, the safest way to update is to delete the pybombs folder all together and start at the beginning of this guide. It might take about an hour, but if you only do this once every few months, it is not that bad. If you want to try and use the git pull method, there is a chance you will encounter an error due to dependencies being violated after a specific modules update, in which case you should delete the pybombs folder and start fresh. To continue with the git pull methods, read on.

There will be three pulls you need to make to keep this installation up to date. You will cd to the following directory and then execute the pull request and rebuilds as follows:

cd /home/<user_name>/<path_to_pybombs_folder>/pybombs

git remote update

git pull

./pybombs update

The above is the recommended way of updating your installation, because it maintains all dependencies for sure. No rebuild necessary here because it's all python stuff. More details can be found here:
http://gnuradio.org/redmine/projects/pybombs/wiki/Using
You can however, go to a specific modules install directory and update just the module directly. For example, gnuradio can be updated like this:

cd …/pybombs/src/gnuradio

git remote update

git pull

rm -Rf build --optional. Will require a complete re-compile which takes a while (~30 min)

mkdir build --optional. Do only if you deleted the directory in the optional step above

cd build

cmake ../ --optional. I would only do it if you deleted the build directory in the optional step

make

sudo make install

And UHD like this:

```
cd .../pybombs/src/uhd/

git remote update

git pull

cd host/build

make

sudo make install
```

Now you have all the up to date repository files. You can create a shell script that does all this for you in one command if you like. If you're not sure where the .git files are located for a module, you can do a general search to find all .git files on your computer by executing:

```
sudo find / -name '*.git'
```

Setup Networking for USRP

Now we need to setup the networking controls on the host machine. To connect to the USRP via Ethernet, the host must have a static IP address. By default, this is not the case. The following setup will create two commands. One command disables DHCP and Network Manager and assigns a static IP address to the computer. In this state, you will not be able to use the regular internet because it requires DHCP. The other command restarts DHCP and Network Manager to the normal state so you can use the internet, but can't connect to the USRP. There is no way to use both the internet and the USRP at the same time as of this guide being written.

We are going to change the interfaces file as described in this post:
http://askubuntu.com/questions/27432/configuring-ethernet-network

```
sudo gedit /etc/network/interfaces
```

add the following to the file, replacing <username> with your username:

```
mapping eth0
script /home/<username>/net.sh
map eth0 eth0-net
map eth0 eth0-usrp

iface eth0-net inet dhcp

iface eth0-usrp inet static
address 10.0.8.1
netmask 255.255.255.0
```

Now we need to create a new file in your home directory and add the following to it and save:

```
gedit /home/<username>/net.sh
```

add this to the file:

```
#!/bin/sh
echo eth0
```

Now to connect to the USRP you would execute:

```
sudo stop network-manager

sudo ifdown eth0

sudo ifup eth0=eth0-usrp
```

To switch back to using the internet, you would execute:

sudo stop network-manager

sudo ifdown eth0

sudo ifup eth0=eth0-usrp

Lastly, to make your life easier, you should make an alias for each of these sets of three commands. What I do is this:

sudo gedit /home/<username>/.bash_aliases

Add the following lines to the file:

alias ethnet="sudo stop network-manager; sudo ifdown eth0; sudo ifup eth0=eth0-net; ifconfig -a"

alias ethusrp="sudo stop network-manager; sudo ifdown eth0; sudo ifup eth0=eth0-usrp; ifconfig -a"

Now when you want to connect to the USRP you execute:

ethusrp

and when you want to connect to the internet you execute:

ethnet

We are finally ready to interact with the USRPs themselves. With the ethusrp command executed so you have a static IP address and with the USRP connected to the host machine with an Ethernet cable either directly or through a switch with multiple USRPs, you can execute:

uhd_find_devices

This will return a list of USRPs that the host recognizes being connected to with their device type, IP address info, name and serial number. If you do not get a list of devices that are connected, something in the previous steps was not done correctly. As a first debug step execute:

ifconfig -a

Confirm that you have a static IP address set. In the eth0 section, on the second row you should see inet addr:10.0.8.1 if you set the static IP address to be 10.0.8.1. If you see something different or can't find that line, you probably don't have your network settings correct. Go back and try that section again.

If you want more detailed information on each of the devices, you can use the following command:

uhd_usrp_probe

If you have multiple USRPs connected to a switch and you want to probe one of them specifically, then add the following flag

uhd_usrp_probe --args="addr=10.0.8.3"

or

uhd_find_devices --args="addr=10.0.8.3"

You are now ready to dive into GNU Radio and use the USRPs to build a software defined radio.

Data Analysis

There are two programs that I've found useful when analyzing data collected or created by GNU Radio. The first is xxd. If you have a binary file that you want to analyze, cd to the files directory and execute:

xxd -b -c 8 -l 64 <file_name>

you can review the man pages for more on this command or type xxd -help, but here is a brief summary:

-b sets the binary output to the terminal
-c 8 sets the number of octets per line, i.e. there will be 8 groups of 8 bits per row in the terminal
-l 64 sets the total number of octets to display across all rows (64 here)

The other useful program is Octave. This is the open source equivalent of MATLAB. It shares many of the same commands and functionality and is powerful for analysis or creating test vectors for input data. You can execute ***octave*** at a command line to begin a session, or save a text file (<file_name>.m) with octave code and run it by executing ***octave <file_name>*** from a terminal.