2015

TENET TECHNETRONICS

Getting started with USRP X310

Author: Rakesh S K

What is Gnu radio?

GNU Radio is a software library, which can be used to develop complete applications for radio engineering and signal processing.

Introduction

GNU Radio 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 software-defined radios, or without hardware in a simulation-like environment.

GNU Radio is licensed under the GNU General Public License (GPL) version 3. All of the code is copyright of the Free Software Foundation. While all the applications are implemented using python language while critical signal processing path is done using C++ language.

Idea behind GNURADIO

The goal is to give ordinary software people the ability to 'hack' the electromagnetic spectrum, i.e. to understand the radio spectrum and think of clever ways to use it.

Why GNURADIO

Instead of purchasing multiple expensive radios, a single generic radio can be implemented using gnu radio software and with support of minimal hardware to receive and transmit processed signal at required frequencies and any data type can be passed from one block to another i.e.it can be in bits, bytes, vectors, bursts or more complex data types

Since the performance critical blocks are implemented in C++ using processor floating point extensions the developers are able to implement real-time, high-throughput radio systems in a simple-to-use, rapid-application-development environment.

One can use it to write applications to receive data out of digital streams or to send data into digital streams, which is then transmitted using hardware. GNU Radio has filters, channel codes, synchronisation elements, equalizers, demodulators, vocoders, decoders, and many other elements which are called as blocks that are typically found in radio systems.

Features

The main features of gnu radio are **flexibility** and **configurability.**

Extending GNU Radio is also quite easy, if you find a specific block that is missing you can quickly create and add it.

Getting started with USRP X310

Assembling the X300/X310 kit

Before you can start using your USRP, you might have to assemble the hardware, if this has not yet happened. Make sure you are grounded (e.g. by touching a radiator) in order not to damage sensitive electronics through static discharge!

- Unscrew the top of your X300/X310 (there are 2 screws which can be easily loosened using a small screwdriver).
- Insert the daughterboards by inserting them into the slots and optionally screwing them onto the motherboard.
- Connect the RF connectors on the daughterboards to the front panel. In order to avoid confusion, make sure the internal connections match the labels on the front panel (i.e. TX/RX is connected to TX/RX).
- If you have purchased an internal GPSDO, follow the instructions on Internal GPSDO Application Notes (USRP-X3x0 Models) to insert the GPSDO. Note that you will need an external GPS antenna connected to the rear GPS ANT connector in order to make use of GPS, although your USRP will still be usable without.
- Connect the 1 GigE SFP+ transceiver into the Ethernet port 0 and connect the X300/X310 with your computer.
- Connect the power supply and switch on the USRP.

Network Connectivity

USRP X310 demands minimum **1Gbps** Ethernet card interface.
The USRP-X Series only supports Gigabit and Ten Gigabit Ethernet and will not work with a 10/100 Mbps interface.
To know whether your system Ethernet card is of 1Gbps you can use a tool called ‘ethtool’

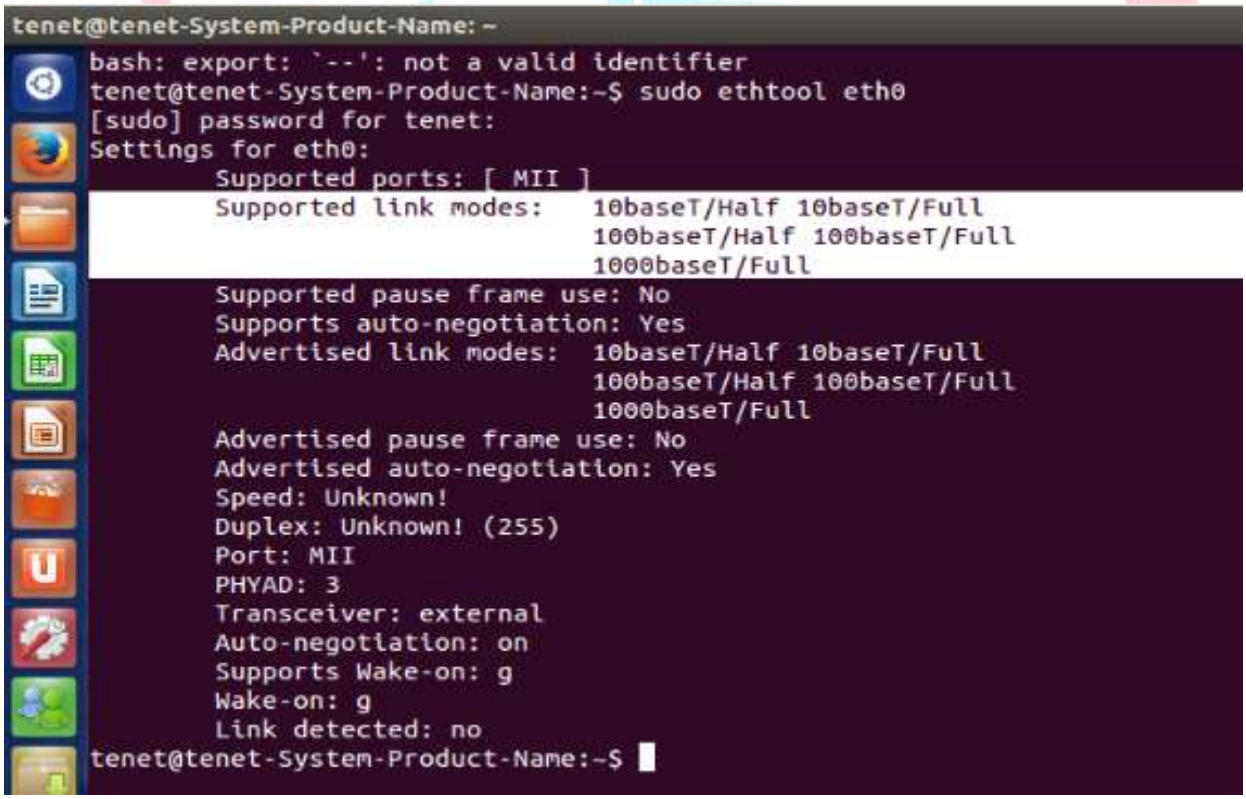
Ethtool installation procedure on Ubuntu:

Run the following commands:

```
$ sudo apt-get install ethtool
```


```
$ sudo ethtool ethx
```

(**ethx** is the name of Ethernet port it may be eth0,eth1,eth3…..)



As highlighted in above image, Ethernet card supports 10MBPS/100MBPS/1GBPS Speed.

Ethernet Interface	USRP Ethernet Port	Default USRP IP Address	Host Static IP Address	Host Static Subnet Mask
Gigabit	Port 0 (HGS Image)	192.168.10.2	192.168.10.1	255.255.255.0
Ten Gigabit	Port 0 (XGS Image)	192.168.30.2	192.168.30.1	255.255.255.0
Ten Gigabit	Port 1 (HGS/XGS Image)	192.168.40.2	192.168.40.1	255.255.255.0

As shown in the aboue table for 1Gigabit ethernet, device adress is 192.168.10.2
Now, to make communication betweeen PC and USRP PC IP address should be 192.168.10.1 and subnet mask is 255.255.255.0

To change IP address of system go to system setting in that select Network as shown below.

Check IP of system by running ‘ifconfig’ command as below

```
$ ifconfig
```

To find USRP device
Run the following command

The USRP device will reply to ICMP echo requests ("ping"). A successful ping response means that the device has booted properly and that it is using the expected IP address.

```
$ ping 192.168.10.2
```

When there is network traffic arriving at the Ethernet port, LEDs will light up. You can use this to make sure the network connection is correctly set up, e.g. by pinging the USRP and making sure the LEDs start to blink.

Run the below command to find the devices connected to your system.

```
$ uhd_find_devices
```

```
tenet@tenet-System-Product-Name:~$ uhd_find_devices
linux; GNU C++ version 4.6.3; Boost_104800; UHD_003.008.002-80-ge28d7844

--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    type: x300
    addr: 192.168.10.2
    fpga: HGS
    name:
    serial: F4C004
    product: X310
```

Updating the FPGA

If the output from uhd_find_devices and uhd_usrp_probe didn't show any warnings, you can skip this step. However, if there were errors regarding the FPGA version compatibility number as shown in below, you will have to update the FPGA image before you can start using your USRP.

```
tenet@tenet-System-Product-Name:~$ uhd_usrp_probe
linux; GNU C++ version 4.6.3; Boost_104800; UHD_003.008.002-80-ge28d7844

-- X300 initialization sequence...
-- Determining maximum frame size... 1472 bytes.
-- Setup basic communication...
Error: RuntimeError: Expected FPGA compatibility number 9, but got 7:
The FPGA image on your device is not compatible with this host code build.
Download the appropriate FPGA images for this version of UHD.
Please run:

 "/usr/local/lib/uhd/utils/uhd_images_downloader.py"

Then burn a new image to the on-board flash storage of your
USRP X3xx device using the burner utility. Please run:

 "/usr/local/lib/uhd/utils/usrp_x3xx_fpga_burner"

For more information, refer to the UHD manual:

 http://files.ettus.com/manual/page_usrp_x3x0.html#x3x0_flash
tenet@tenet-System-Product-Name:~$
```

1. Download the current UHD images. You can use the uhd_images_downloader script provided with UHD.
2. Use the `usrp_x3xx_fpga_burner` utility to update the FPGA image. On the command line, run:

```
3.  usrp_x3xx_fpga_burner --addr=192.168.10.2 --type=HGS
```

If you have installed the images to a non-standard location, you might need to run (change the filename according to your device):

```
usrp_x3xx_fpga_burner --addr=192.168.10.2 --fpga-path
<path_to_images>/usrp_x310_fpga_HGS.bit
```

The process of updating the FPGA image will take several minutes. Make sure the process of flashing the image does not get interrupted.

When your FPGA is up to date, power-cycle the device and re-run `uhd_usrp_probe`. There should be no more warnings at this point, and all components should be correctly detected. Your USRP is now ready for development!

Changing the USRP's IP address

You may need to change the USRP's IP address for several reasons:

- to satisfy your particular network configuration
- to use multiple USRP-X Series devices on the same host computer
- to set a known IP address into USRP (in case you forgot)

To change the USRP's IP address, you must know the current address of the USRP, and the network must be setup properly as described above. You must also know which IP address of the X300 you want to change, as identified by their address EEPROM key (e.g. ip-addr0, see the table above). Run the following commands:

UNIX:

```
cd <install-path>/lib/uhd/utils
./usrp_burn_mb_eeprom --args=<optional device args> --values="ip-addr0=192.168.10.3"
```

Windows:

```
cd <install-path>\lib\uhd\utils
usrp_burn_mb_eeprom.exe --args=<optional device args> --values="ip-addr0=192.168.10.3"
```

You must power-cycle the device before you can use this new address.