Verifying the Operation of the USRP Using UHD and GNU Radio

Contents

- 1 Application Note Information
- 2 Revision History
- 3 Abstract
- 4 Overview
- 5 Proper Care and Handling
- 6 Finding the Device
- 7 UHD Programs
- 8 Benchmarking your system
- 9 Receiving Samples
- 10 Transmitting Samples
- 11 Terminal DFT/FFT
- 12 Transmit test signal
- 13 Additional UHD Utilities

AN-936 by Neel Pandeya

This AN explains how to use UHD and GNU Radio, once installed, to verify the correct operation of the USRP. Several test procedures are explained in detail. Several tests make use of an optional spectrum analyzer and signal generator.

This application note will cover testing your USRP to verify it is operational. You will need to have a working installation of UHD to preform the following steps to verify your USRP. If you do not have UHD installed, please reference the Building and Installing the USRP Open-Source Toolchain (UHD and GNU Radio) on Linux, OS X and Windows Application Notes.

Alternatively, you may use the Live SDR Environment to perform the following steps. More information about the Live SDR Environment is available at the Live SDR Environment Getting Started Guides page.

All Ettus Research products are individually tested before shipment. The USRP? is guaranteed to be functional at the time it is received by the customer. Improper use or handling of the USRP? can easily cause the device to become non-functional. Listed below are some examples of actions which can prevent damage to the unit:

- Never allow metal objects to touch the circuit board while powered.
- Always properly terminate the transmit port with an antenna or 50? load.
- Always handle the board with proper anti-static methods.
- Never allow the board to directly or indirectly come into contact with any voltage spikes.
- Never allow any water, or condensing moisture, to come into contact with the boards.
- Always use caution with FPGA, firmware, or software modifications.

Never apply more than -15 dBm of power into any RF input

Always use at least 30dB attenuation if operating in loopback configuration

- Report detailed information on UHD-supported Software Radio Peripherals attached by USB, network, or embedded configuration. Details include unit names, revision numbers, and available sensors on all attached USRP motherboards and daughterboards.

```
uhd_usrp_probe
```

Example output from `uhd_usrp_probe`:

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

-- Opening a USRP2/N-Series device...
-- Current recv frame size: 1472 bytes
-- Current send frame size: 1472 bytes
  _____________________________________________________
 /
|       Device: USRP2 / N-Series Device
|     _______________________________________________
|    /
|   |       Mboard: N210r4
|   |   hardware: 2577
|   |   mac-addr: 00:00:00:00:00:00
|   |   ip-addr: 192.168.10.2
|   |   subnet: 255.255.255.255
|   |   gateway: 255.255.255.255
|   |   gpsdo: none
|   |   serial: xxxxxx
|   |   FW Version: 12.4
|   |   FPGA Version: 11.1
|   |
|   |   Time sources:  none, external, _external_, mimo
|   |   Clock sources: internal, external, mimo
|   |   Sensors: mimo_locked, ref_locked
|   |     _____________________________________________
|   |    /
|   |   |       RX DSP: 0
|   |   |
|   |   |   Freq range: -50.000 to 50.000 MHz
|   |     _____________________________________________
|   |    /
|   |   |       RX DSP: 1
|   |   |
|   |   |   Freq range: -50.000 to 50.000 MHz
|   |     _____________________________________________
|   |    /
```

```
|   |   |       RX Dboard: A
|   |   |   ID: WBX, WBX + Simple GDB (0x0053)
|   |   |   Serial: xxxxxx
|   |   |     ____________________________________________________
|   |   |    /
|   |   |   |       RX Frontend: 0
|   |   |   |   Name: WBXv2 RX+GDB
|   |   |   |   Antennas: TX/RX, RX2, CAL
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 68.750 to 2200.000 MHz
|   |   |   |   Gain range PGA0: 0.0 to 31.5 step 0.5 dB
|   |   |   |   Bandwidth range: 40000000.0 to 40000000.0 step 0.0 Hz
|   |   |   |   Connection Type: IQ
|   |   |   |   Uses LO offset: No
|   |   |     ____________________________________________________
|   |   |    /
|   |   |   |       RX Codec: A
|   |   |   |   Name: ads62p44
|   |   |   |   Gain range digital: 0.0 to 6.0 step 0.5 dB
|   |   |   |   Gain range fine: 0.0 to 0.5 step 0.1 dB
|   |   |  ___________________________________________________
|   |   |    /
|   |   |   |       TX DSP: 0
|   |   |   |
|   |   |   |   Freq range: -50.000 to 50.000 MHz
|   |   |  ___________________________________________________
|   |   |    /
|   |   |   |       TX Dboard: A
|   |   |   |   ID: WBX (0x0052)
|   |   |   |   Serial: xxxxxx
|   |   |   |     ____________________________________________________
|   |   |   |    /
|   |   |   |   |       TX Frontend: 0
|   |   |   |   |   Name: WBXv2 TX+GDB
|   |   |   |   |   Antennas: TX/RX, CAL
|   |   |   |   |   Sensors: lo_locked
|   |   |   |   |   Freq range: 68.750 to 2200.000 MHz
|   |   |   |   |   Gain range PGA0: 0.0 to 25.0 step 0.1 dB
|   |   |   |   |   Bandwidth range: 40000000.0 to 40000000.0 step 0.0 Hz
|   |   |   |   |   Connection Type: IQ
|   |   |   |   |   Uses LO offset: No
|   |   |   |     ____________________________________________________
|   |   |   |    /
|   |   |   |   |       TX Codec: A
|   |   |   |   |   Name: ad9777
|   |   |   |   |   Gain Elements: None
```

- Find UHD-supporting Software Radio Peripherals attached by USB, network or embedded configuration. Uses broadcast packets for discovery. Often blocked by routers, switches, firewalls.

```
uhd_find_devices
```

Example output from `uhd_find_devices`:

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

--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    type: usrp2
    addr: 192.168.10.2
    name:
    serial: xxxxxx
```

- View firewall settings with:

```
sudo iptables -L
```

The example programs included with the UHD driver will be located at:

```
$install_prefix/lib/uhd/examples
```

On Linux, the default installation prefix location is `/usr/local`.

The example programs are located in the folder:

```
/usr/local/lib/uhd/examples
```

On OSX, the configuration is similar to Linux, however the default installation prefix location is `/opt/local`.

The example programs will be located in the folder:

```
/opt/local/lib/uhd/examples
```

On Windows, the default installation prefix will be `C:\Program Files\UHD\`.

The example programs will be located in the folder:

```
C:\Program Files\UHD\lib\uhd\examples
```

Before executing the following commands, change directory to the appropriate location.

- Benchmarks interface with device.

```
./benchmark_rate --rx_rate 10e6 --tx_rate 10e6
```

Example output from `benmark_rate`:

```
$ ./benchmark_rate --rx_rate 10e6 --tx_rate 10e6
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.000.HEAD-0-g6e1ac3fc


Creating the usrp device with: ...
-- Opening a USRP2/N-Series device...
-- Current recv frame size: 1472 bytes
-- Current send frame size: 1472 bytes
Using Device: Single USRP:
  Device: USRP2 / N-Series Device
  Mboard 0: N210r4
  RX Channel: 0
    RX DSP: 0
    RX Dboard: A
    RX Subdev: WBXv2 RX+GDB
  TX Channel: 0
    TX DSP: 0
    TX Dboard: A
    TX Subdev: WBXv2 TX+GDB

Setting device timestamp to 0...
Testing receive rate 10.000000 Msps on 1 channels
Testing transmit rate 10.000000 Msps on 1 channels

Benchmark rate summary:
  Num received samples:    100104043
  Num dropped samples:     0
  Num overflows detected:  0
  Num transmitted samples: 100229019
  Num sequence errors:     0
  Num underflows detected: 0
  Num late commands:       0
  Num timeouts:            0


Done!
```

- Saves samples to file

```
./rx_samples_to_file --freq 98e6 --rate 5e6 --gain 20 --duration 10 usrp_samples.dat
```

Example output from `rx_samples_to_file`:

```
$ ./rx_samples_to_file --freq 98e6 --rate 5e6 --gain 20 --duration 10 usrp_samples.dat
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.000.HEAD-0-g6e1ac3fc


Creating the usrp device with: ...
-- Opening a USRP2/N-Series device...
-- Current recv frame size: 1472 bytes
-- Current send frame size: 1472 bytes
Using Device: Single USRP:
  Device: USRP2 / N-Series Device
  Mboard 0: N210r4
  RX Channel: 0
    RX DSP: 0
    RX Dboard: A
    RX Subdev: WBXv2 RX+GDB
  TX Channel: 0
    TX DSP: 0
    TX Dboard: A
    TX Subdev: WBXv2 TX+GDB

Setting RX Rate: 5.000000 Msps...
Actual RX Rate: 5.000000 Msps...

Setting RX Freq: 98.000000 MHz...
Actual RX Freq: 98.000000 MHz...

Setting RX Gain: 20.000000 dB...
Actual RX Gain: 20.000000 dB...

Waiting for "lo_locked": ++++++++++ locked.

Press Ctrl + C to stop streaming...

Done!
```

Example of file size output from `rx_samples_to_file`:

```
$ ls -al usrp_samples.dat
-rw-rw-r-- 1 user user 200040000 Sep  1 14:43 usrp_samples.dat
```

- Transmits samples from file

```
./tx_samples_from_file --freq 915e6 --rate 5e6 --gain 10 usrp_samples.dat
```

Example of output from `tx_samples_from_file`:

```
$ ./tx_samples_from_file --freq 915e6 --rate 5e6 --gain 10 usrp_samples.dat
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.000.HEAD-0-g6e1ac3fc


Creating the usrp device with: ...
-- Opening a USRP2/N-Series device...
-- Current recv frame size: 1472 bytes
-- Current send frame size: 1472 bytes
Using Device: Single USRP:
  Device: USRP2 / N-Series Device
  Mboard 0: N210r4
  RX Channel: 0
```

```
    RX DSP: 0
    RX Dboard: A
    RX Subdev: WBXv2 RX+GDB
  TX Channel: 0
    TX DSP: 0
    TX Dboard: A
    TX Subdev: WBXv2 TX+GDB

Setting TX Rate: 5.000000 Msps...
Actual TX Rate: 5.000000 Msps...

Setting TX Freq: 915.000000 MHz...
Actual TX Freq: 915.000000 MHz...

Setting TX Gain: 10.000000 dB...
Actual TX Gain: 10.000000 dB...

Checking TX: LO: locked ...

Done!
```

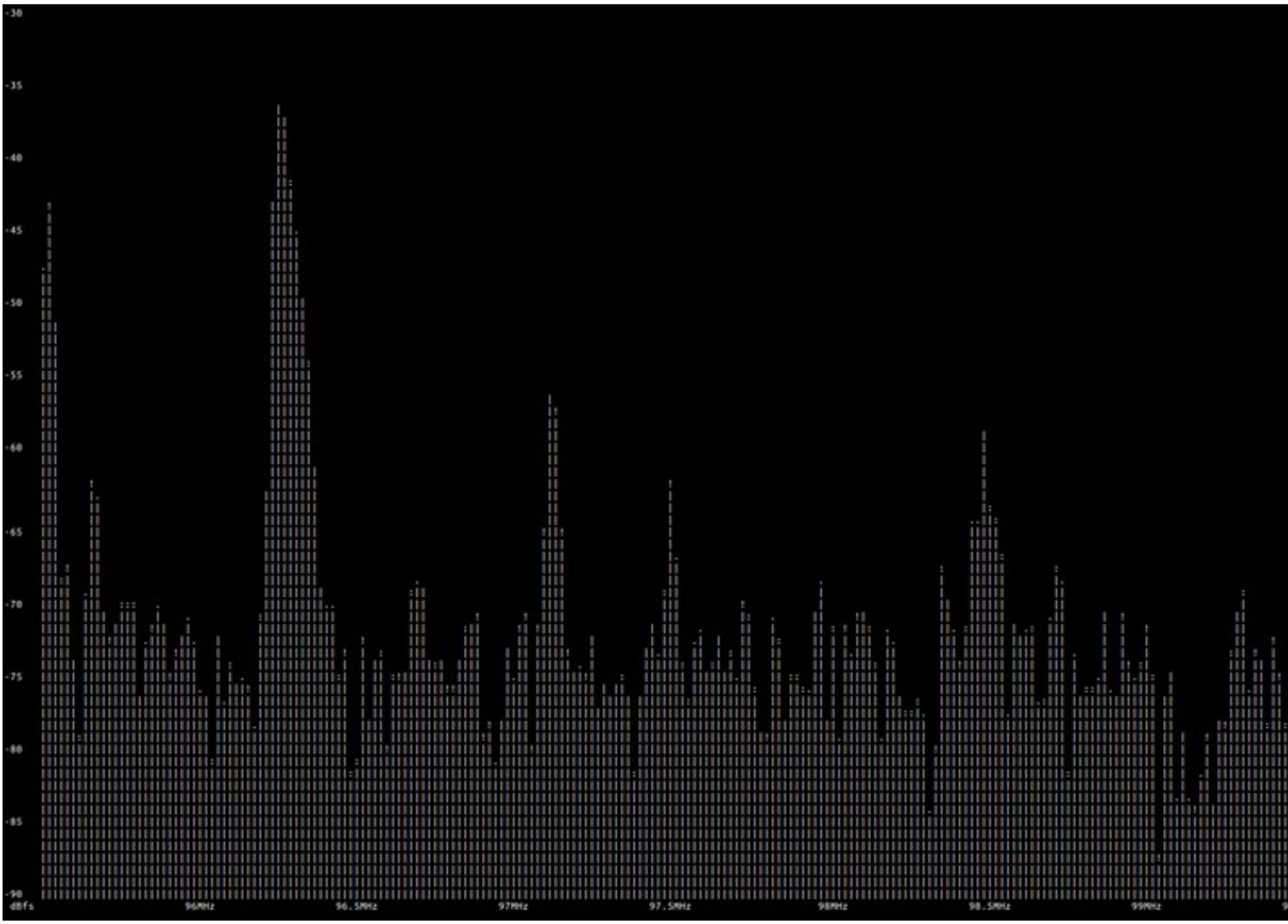
- Creates ASCII/Ncurses FFT

```
./rx_ascii_art_dft --freq 98e6 --rate 5e6 --gain 20 --bw 5e6 --ref-lvl -30
```

Example output from `rx_ascii_art_dft`:

```
$ ./rx_ascii_art_dft --freq 98e6 --rate 5e6 --gain 20 --bw 5e6 --ref-lvl -30
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.000.HEAD-0-g6e1ac3fc


Creating the usrp device with: ...
-- Opening a USRP2/N-Series device...
-- Current recv frame size: 1472 bytes
-- Current send frame size: 1472 bytes
Using Device: Single USRP:
  Device: USRP2 / N-Series Device
  Mboard 0: N210r4
  RX Channel: 0
    RX DSP: 0
    RX Dboard: A
    RX Subdev: WBXv2 RX+GDB
  TX Channel: 0
    TX DSP: 0
    TX Dboard: A
    TX Subdev: WBXv2 TX+GDB

Setting RX Rate: 5.000000 Msps...
Actual RX Rate: 5.000000 Msps...

Setting RX Freq: 98.000000 MHz...
Actual RX Freq: 98.000000 MHz...

Setting RX Gain: 20.000000 dB...
Actual RX Gain: 20.000000 dB...

Setting RX Bandwidth: 5.000000 MHz...
Actual RX Bandwidth: 5.000000 MHz...

Checking RX: LO: locked ...
```

Example screenshot of `rx_ascii_art_dft` running:



- Transmits specific waveform

```
./tx_waveforms --freq 915e6 --rate 5e6 --gain 0
```

Example output from `tx_waveforms`:

```
$ ./tx_waveforms --freq 915e6 --rate 5e6 --gain 0
linux; GNU C++ version 4.8.4; Boost_105400; UHD_003.010.000.HEAD-0-g6e1ac3fc


Creating the usrp device with: ...
-- Opening a USRP2/N-Series device...
-- Current recv frame size: 1472 bytes
-- Current send frame size: 1472 bytes
Using Device: Single USRP:
  Device: USRP2 / N-Series Device
  Mboard 0: N210r4
  RX Channel: 0
    RX DSP: 0
    RX Dboard: A
    RX Subdev: WBXv2 RX+GDB
  TX Channel: 0
    TX DSP: 0
    TX Dboard: A
    TX Subdev: WBXv2 TX+GDB

Setting TX Rate: 5.000000 Msps...
Actual TX Rate: 5.000000 Msps...

Setting TX Freq: 915.000000 MHz...
Actual TX Freq: 915.000000 MHz...

Setting TX Gain: 0.000000 dB...
Actual TX Gain: 0.000000 dB...

Setting device timestamp to 0...
Checking TX: LO: locked ...
Press Ctrl + C to stop streaming...
^C
Done!
```

- Downloads FPGA images for current UHD version

```
sudo uhd_images_downloader
```

Example output from running `uhd_images_downloader`:

```
$ [sudo] uhd_images_downloader # Use sudo if installing into /usr or /usr/local on Linux systems
Images destination:      /usr/local/share/uhd/images
Downloading images from: http://files.ettus.com/binaries/images/uhd-images_003.010.000.000-release.zip
Downloading images to:   /tmp/tmpPYd_5J/uhd-images_003.010.000.000-release.zip
58959 kB / 58959 kB (100%)

Images successfully installed to: /usr/local/share/uhd/images
```

- Flashes an image onto the X300/X310 FPGA

`usrp_x3xx_fpga_burner`

- Reading and writing motherboard EEPROM

`usrp_burn_mb_eeprom`

- Reading and writing daughterboard EEPROM

`usrp_burn_db_eeprom`