N321 Getting Started Guide

Contents

- 1 Kit Contents
  - 1.1 N300
  - 1.2 N310
  - 1.3 N320
  - 1.4 N321
- 2 Verify the Contents of Your Kit
- 3 You Will Need
- 4 Proper Care and Handling
- 5 Install and Setup the Software Tools on Your Host Computer
- 6 Connecting the Device
  - 6.1 Interfaces Overview
  - 6.2 Setting up a Serial Console Connection
    - 6.2.1 Connecting to the microcontroller
  - 6.3 Connecting to the ARM via SSH
- 7 Updating the Linux File System
  - 7.1 File System Partition Layout
  - 7.2 Updating the file system with Mender
    - 7.2.1 Mender Update Process
  - 7.3 Updating the files system by writing the disk image
- 8 Updating the Network Configurations
- 9 Updating the FPGA Image
  - 9.1 Network Mode FPGA Image Update
  - 9.2 Embedded Mode FPGA Image Update
- 10 Setting Up a Streaming Connection
  - 10.1 1Gb Streaming SFP Port 0
  - 10.2 10Gb Streaming SFP Port 1
  - 10.3 Dual 10Gb Streaming SFP Ports 0/1
- 11 Verifying Device Operation
  - 11.1 Subdevice Specification Mapping
    - 11.1.1 N300
    - 11.1.2 N310
      - 11.1.2.1 UHD 3.11.x.x - 3.12.x.x
      - 11.1.2.2 UHD 3.13.x.x+
    - 11.1.3 N320
    - 11.1.4 N321
  - 11.2 Supported Sample Rates
    - 11.2.1 Example Sample Rates
    - 11.2.2 N320/N321 Example Sample Rates
  - 11.3 Probe the USRP
    - 11.3.1 N300/N310
    - 11.3.2 N320
    - 11.3.3 N321
  - 11.4 ASCII Art Example
  - 11.5 Benchmarking your system
    - 11.5.1 1 Gb Interface
    - 11.5.2 10 Gb Interface SFP 1
    - 11.5.3 Dual 10 Gb Interface
- 12 USRP N3xx Device Specific Operations
  - 12.1 White Rabbit Ethernet-Based Synchronization
  - 12.2 N320/N321
  - 12.3 Turning the Device Off/On
  - 12.4 Enable Auto Booting
  - 12.5 Default Password
- 13 Technical Support and Community Knowledge Base
- 14 Legal Considerations
- 15 Sales and Ordering Support
- 16 Terms and Conditions of Sale

- USRP N300
- DC Power Supply (12V, 7A)
- 1 RJ45 ? SFP+ Adapter
- 1 Gigabit Ethernet Cat-5e Cable (3m)
- USB-A to Micro USB-B Cable (1m)
- Getting Started Guide
- Ettus Research Sticker

- USRP N310
- DC Power Supply (12V, 7A)
- 1 RJ45 ? SFP+ Adapter
- 1 Gigabit Ethernet Cat-5e Cable (3m)
- USB-A to Micro USB-B Cable (1m)
- Getting Started Guide
- Ettus Research Sticker

- USRP N320
- DC Power Supply (12V, 7A)
- 1 RJ45 ? SFP+ Adapter
- 1 Gigabit Ethernet Cat-5e Cable (3m)
- USB-A to Micro USB-B Cable (1m)
- Getting Started Guide
- Ettus Research Sticker

- USRP N321
- DC Power Supply (12V, 7A)
- 1 RJ45 ? SFP+ Adapter
- 1 Gigabit Ethernet Cat-5e Cable (3m)
- USB-A to Micro USB-B Cable (1m)
- Getting Started Guide
- Ettus Research Sticker

Ensure that your kit contains all the items listed above. If any items are missing, please contact sales@ettus.com? immediately.

- microSD Card Writer
- For Network Mode: A host computer with an available 1 or 10 Gigabit Ethernet interface for sample streaming. In addition to the Ethernet interface used for sampling streaming, your host computer will require a separate 1 Gigabit Ethernet interface for command and control streaming.
- For Stand-Alone Embedded Mode: A host computer with an available 1 Gigabit Ethernet port or a USB 2.0 port to remotely access the embedded Linux operating system running on ARM CPU.

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 cause the device to become non-functional. Take the following precautions to 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 device.
- 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

In order to use your Universal Software Radio Peripheral (USRP?), you must have the software tools correctly installed and configured on your host computer. A step-by-step guide for doing this is available at the Building and Installing the USRP Open-Source Toolchain (UHD and GNU Radio) on Linux, OS X and Windows Application Notes.

To find the latest release of UHD, see the UHD repository at https://github.com/EttusResearch/uhd.

The USRP N310 requires UHD version 3.11.0.0 or later.

The USRP N300 requires UHD version 3.12.0.0 or later.

The USRP N320/N321 requires UHD version 3.14.0.0 or later.

White Rabbit Ethernet-Based Synchronization of the N3xx USRP requires UHD version 3.12.0.0 or later. For additional details on White Rabbit Ethernet-Based Synchronization, please see the application note, Using Ethernet-Based Synchronization on the USRP? N3xx Devices.

When you receive a brand-new device, it is strongly recommended that you download the most recent filesystem image from the Ettus Research website and write it to the SD card that comes with the unit. It is not recommended that you use the SD card from the factory as-is. Instructions on downloading the latest filesystem image and writing it to the SD card are listed below.

Note that if you are operating the device in Network Mode, the version of UHD running on the host computer and the USRP N3xx must match.

Listed below are the interfaces to connect to the USRP N3xx. Each interface has specific functionality, limitations and purpose.

Serial Console

The Serial Console provides a low level interface to the device typically used for debugging.

1 Gigabit RJ45 Connection

The 1 Gigabit RJ45 Connection interfaces with the on-board ARM CPU. When operated in "Network mode", this interface can optionally be used for UHD management traffic. Regardless of the operation mode (Network vs Embedded) this interface can be used to connect to the ARM via SSH. By default, the 1Gb RJ45 connection is configured to use a DHCP assigned IP address.

Dual SFP+ Connections

The Dual SFP+ Connections support multiple configurations for streaming high-speed, low-latency data, depending upon the FPGA image which is loaded.

QSFP+ Connection (N320/ N321 Only)

The QSFP+ Connection supports 2 x 10Gb lanes for streaming high-speed, low-latency data, while the onboard SFP0 connection is used for White Rabbit Ethernet-Based Synchronization.

It is possible to gain shell access to the device using a serial terminal emulator via the Serial Console port. Most Linux, OSX, or other Unix based operating systems have a tool called `screen` which can be used for this purpose.

If you do not have `screen` installed, it can be installed via your package manager. For Ubuntu/Debian based operating systems it can be installed with `apt` such as:

```
sudo apt install screen
```

The default Baud Rate for the Serial Console is: `115200`

The exact device node you should attach to depends on your operating system's driver and other USB devices that might already be connected. Modern Linux systems offer alternatives to simply trying device nodes; instead, the OS might have a directory of symlinks under `/dev/serial/by-id`:

```
$ ls /dev/serial/by-id
usb-Digilent_Digilent_USB_Device_25163511FE00-if00-port0
usb-Digilent_Digilent_USB_Device_25163511FE00-if01-port0
usb-Silicon_Labs_CP2105_Dual_USB_to_UART_Bridge_Controller_007F6CB5-if00-port0
usb-Silicon_Labs_CP2105_Dual_USB_to_UART_Bridge_Controller_007F6CB5-if01-port0
```

NOTE: Exact names depend on the host operating system version and may differ.

Every N3XX series device connected to USB will by default show up as four different devices. The devices labeled `"USB_to_UART_Bridge_Controller"` are the devices that offer a serial prompt. The first (with the `if00` suffix) connects to the `ARM CPU`, whereas the second connects to the `STM32 Microcontroller`.

If you have multiple N3xx Serial Consoles connected to a single host, you may have to empirically test nodes.

Connecting to the ARM CPU can be performed with the command:

```
$ sudo screen /dev/serial/by-id/usb-Silicon_Labs_CP2105_Dual_USB_to_UART_Bridge_Controller_007F6CB5-if00-port0 115200
```

Upon starting the USRP N3xx, boot messages will appear and rapidly update. Once the boot process successfully completes, a login prompt like the following should appear:

```
OpenEmbedded test ni-n3xx-313ABDA ttyPS0

ni-n3xx-313ABDA login:
```

Enter the username: ?`root`?

By default, the `root` user's password is left blank. Press the `Enter` key when prompted for a password.

You should now be presented with a shell prompt similar to the following:

```
root@ni-n3xx-<motherboard serial #>:~#
```

Using the default configuration, the serial console will show all kernel log messages (which are not available when using SSH), and give access to the boot loader (U-boot prompt). This can be used to debug kernel or boot-loader issues more efficiently than when logged in via SSH.

Using the Serial Console interface, it is possible to connect to the STM32 microcontroller with the command below. The STM32 controls the power sequencing and several other low level device operations.

```
$ sudo screen /dev/serial/by-id/usb-Silicon_Labs_CP2105_Dual_USB_to_UART_Bridge_Controller_007F6CB5-if01-port0 115200
```

The STM32 interface provides a very simple prompt. The command `help` will list all available commands. A direct connection to the microcontroller can be used to hard-reset the device without physically accessing it (i.e., emulating a power button press) and other low-level diagnostics.

By default, the RJ45 1Gb management interface is configured to be assigned a DHCP IP address.

If you have access to a network which provides a DHCP server (such as a common router's LAN), attach the RJ45 1Gb port to this network. Details vary by vendor, however, most router management interfaces will provide a list of attached devices to the LAN including their IP address.

Without access to a router management interface, you can identify the IP address by connecting to the ARM CPU via Serial Console as detailed in the section above and running the command `ip a`:

Example Output:

```
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.151/24 brd 192.168.1.255 scope global dynamic eth0
       valid_lft 42865sec preferred_lft 42865sec
3: sfp0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc pfifo_fast qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.2/24 brd 192.168.10.255 scope global sfp0
       valid_lft forever preferred_lft forever
4: sfp1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 9000 qdisc pfifo_fast qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
```

If you do not have access to a network with a DHCP server, you can create one using the Linux utility `dnsmasq`:

```
$ sudo dnsmasq -i <ETHERNET_ADAPTER_NAME> --dhcp-range=192.168.1.151,192.168.1.254 --except-interface=lo --bind-dynamic --no-daemon
```

NOTE: Modify the value `<ETHERNET_ADAPTER_NAME>` to match the interface you would like to create a DHCP server on.

After the device has obtained an IP address, you can remotely log into it from a Linux or macOS system with SSH, as shown below:

```
$ ssh root@192.168.1.151
```

NOTE: The IP address may vary depending on your network setup.

NOTE: The `root` password default password is empty/blank.

On Microsoft Windows, the SSH connection can be established using the third-party program ?Putty?.

After logging in, you should be presented with a shell like the following:

```
root@ni-n3xx-<motherboard serial #>:~#
```

Before operating the device, it is? ?strongly? recommended to update to the latest version of the Embedded Linux file system. If you are operating the device in Network Mode, the version of UHD running on the host machine and N3xx USRP must match.

There is two ways to update the file system for the N3xx USRP:

1. Mender

2. Physically remove microSD card from device and write a new file system to the microSD card.

The SD Card is divided into four partitions. There is two root file system partitions, a boot partition and a data partition.

Any data you would like to preserve through Mender updates should be saved to the `data` partition, which is mounted at `/data`.

Mender is third-party software that enables remote updating of the root file system without physically accessing the device (see also the Mender website https://mender.io). Mender can be executed locally on the device, or a Mender server can be set up which can be used to remotely update an arbitrary number of USRP devices. Users can host their own local Mender server, or use servers hosted by Mender as a paid service; contact Mender for more information.

When updating the file system using Mender, the tool will overwrite the root file system partition that is not currently mounted. Any data stored in the root partitions will be permanently lost with a Mender update.

After updating a partition with Mender, it will reboot into the newly updated partition. Only if the update is confirmed by the user, the update will be made permanent. This means that if an update fails, the device will be always able to reboot into the partition from which the update was originally launched, which presumably is in a working state. Another update can be launched now to correct the previous, failed update, until it works.

To obtain the file system Mender image (these are files with a `.mender` suffix), run the following command on the host computer with Internet access:

```
$ sudo uhd_images_downloader -t mender -t n3xx --yes
```

Example Output:

```
[INFO] Using base URL: https://files.ettus.com/binaries/cache/
[INFO] Images destination: /usr/local/share/uhd/images
365684 kB / 365684 kB (100%) n3xx_common_mender_default-v4.6.0.0.zip
[INFO] Images download complete.
```

The downloaded "zip" archive is extracted into the `Images destination` directory with the filename `usrp_n3xx_fs.mender`. Next, you will need to copy this Mender file system image from the `Images destination` directory to the USRP N3xx to have its filesystem changed. This can be done with the Linux utility `scp`, for example as follows:

```
$ scp /usr/local/share/uhd/images/usrp_n3xx_fs.mender root@192.168.1.51:~/.
```

Note: The path and IP may be different for your configuration; the command above assumes you're using the default UHD installation path of `/usr/local` and that the N3xx's IP is `192.168.1.51`.

After copying the Mender file system image to the N3xx, connect to the N3xx to gain shell access via either the Serial Console or SSH.

On the N3xx, you first need to determine the version of UHD currently running on the USRP; an easy way to do this is via the command

```
# uhd_config_info --version
```

Example output:

```
UHD 3.14.1.1-0-g0347a6d8
```

The mender command to execute is different for UHD version 4.0 or newer versus prior to version 4.0. For the former use `mender install` followed by the mender file; for the latter use `mender -f -rootfs` followed by the mender file. Starting with UHD version 4.0 one can use mender to upgrade or downgrade the UHD filesystem version between any UHD v4 versions (e.g., 4.1 to 4.6; 4.6 to 4.1). The following commands assume that the UHD filesystem is version 4; if not then substitute the other mender command.

Run `mender install /path/to/latest.mender` to update the file system, e.g.:

```
# mender install usrp_n3xx_fs.mender
```

The artifact can also be stored on a remote server:

```
# mender install http://server.name/path/to/latest.mender
```

This procedure will take a quite a few minutes to complete. While executing, mender will show progress, e.g.:

```
................................   0% 1024 KiB
...
................................   1% 4096 KiB
...
................................  99% 386048 KiB
............                      100% 386458 KiB
INFO[3865] wrote 7851737088/7851737088 bytes of update to device /dev/mmcblk0p3  module=device
INFO[3871] Enabling partition with new image installed to be a boot candidate: 3  module=device
```

After mender has logged a successful update, reboot the device:

```
# reboot
```

Upon reboot log back in to the USRP N3xx and run the command `uhd_find_devices` to verify that the UHD version is as desired and that the command runs successfully -- it should find at a minimum the USRP it is being executed on and will find more if other USRPs are on the same network.

If upon reboot the device is not working or the UHD version is not as desired, then the easiest way forward is to overwrite the sdcard filesystem manually with the desired UHD version.

If upon reboot everything is as desired and the device seems functional, then commit the changes so that the boot loader knows to permanently boot into this partition:

```
# mender commit
```

To identify the currently installed Mender artifact from the command line, the following file can be queried on the N3xx:

```
# cat /etc/mender/artifact_info
```

If you are using a Mender server, the updates can be initiated from a web dashboard. From there, you can start the updates without having to log into the device, and you can update groups of USRPs with a few clicks in a web GUI. The dashboard can also be used to inspect the state of USRPs. This is a simple way to update groups of rack-mounted USRPs with custom file systems.

For more information on updating the filesystem, refer to the UHD Manual?.

Please see the separate application note, Writing the USRP File System Disk Image to a SD Card, for step-by-step instructions on writing the file system image to the SD card.

The USRP N3xx systemd network configuration files are located at: `/data/network/`

```
# ls /data/network/
eth0.network  int0.network  sfp0.network  sfp1.network
```

or for older versions of the file system: `/etc/systemd/network/`

```
# ls /etc/systemd/network/
eth0.network  sfp0.network  sfp1.network
```

For details on configuration please refer to the systemd-networkd manual pages.

The factory settings are as follows:

```
eth0 (DHCP):

    [Match]
    Name=eth0

    [Network]
    DHCP=ipv4

    [DHCP]
    UseHostname=false

sfp0 (static):

    [Match]
    Name=sfp0

    [Network]
    Address=192.168.10.2/24

    [Link]
    MTUBytes=9000

sfp1 (static):

    [Match]
    Name=sfp1

    [Network]
    Address=192.168.20.2/24

    [Link]
    MTUBytes=9000
```

Additional notes on networking:

- Care needs to be taken when editing these files on the device, since `vi` / `vim` sometimes generates undo files (e.g. `/etc/systemd/network/sfp0.network~`), that `systemd-networkd` might accidentally pick up.
- Temporarily setting the IP addresses or MTU sizes via `ifconfig` or other command line tools will only change the value until the next reboot or reload of the FPGA image.
- If the MTU of the device and host computers differ, streaming issues can occur.
- Streaming via SFP0 at 1 Gb rates requires a MTU of `1500`
- Streaming via SFP0 at 10 Gb rates requires a MTU of `9000`

For addition details on network configuration here: https://files.ettus.com/manual/page_usrp_n3xx.html#n3xx_network_configuration

The FPGA image should match the version of UHD installed on the host computer, when operated in Network mode. Connect the device to the host computer using either the RJ45 or SFP+ port, refer to the section above for detailed instructions.

To obtain all the FPGA images for a specific version of UHD, run the following command on the host computer with internet access:

```
$ sudo uhd_images_downloader
```

Example Output:

```
$ sudo uhd_images_downloader
[INFO] Images destination: /usr/local/share/uhd/images
00006 kB / 00006 kB (100%) usrp1_b100_fw_default-g6bea23d.zip
19810 kB / 19810 kB (100%) x3xx_x310_fpga_default-gf1ba32fe.zip
02757 kB / 02757 kB (100%) usrp2_n210_fpga_default-g6bea23d.zip
02123 kB / 02123 kB (100%) n230_n230_fpga_default-ge57dfe0.zip
00522 kB / 00522 kB (100%) usrp1_b100_fpga_default-g6bea23d.zip
00491 kB / 00491 kB (100%) b2xx_b200_fpga_default-ge57dfe0.zip
02415 kB / 02415 kB (100%) usrp2_n200_fpga_default-g6bea23d.zip
08988 kB / 08988 kB (100%) e3xx_e320_fpga_default-g3de8954a.zip
23045 kB / 23045 kB (100%) n3xx_n310_fpga_default-g3de8954a.zip
00523 kB / 00523 kB (100%) b2xx_b205mini_fpga_default-ge57dfe0.zip
18937 kB / 18937 kB (100%) x3xx_x300_fpga_default-gf1ba32fe.zip
00017 kB / 00017 kB (100%) octoclock_octoclock_fw_default-g14000041.zip
00007 kB / 00007 kB (100%) usrp2_usrp2_fw_default-g6bea23d.zip
00009 kB / 00009 kB (100%) usrp2_n200_fw_default-g6bea23d.zip
00450 kB / 00450 kB (100%) usrp2_usrp2_fpga_default-g6bea23d.zip
00144 kB / 00144 kB (100%) b2xx_common_fw_default-ga69ab0c.zip
25107 kB / 25107 kB (100%) n3xx_n320_fpga_default-g3de8954a.zip
00464 kB / 00464 kB (100%) b2xx_b200mini_fpga_default-ge57dfe0.zip
00319 kB / 00319 kB (100%) usrp1_usrp1_fpga_default-g6bea23d.zip
04839 kB / 04839 kB (100%) usb_common_windrv_default-g14000041.zip
00009 kB / 00009 kB (100%) usrp2_n210_fw_default-g6bea23d.zip
16065 kB / 16065 kB (100%) n3xx_n300_fpga_default-g3de8954a.zip
05578 kB / 05578 kB (100%) e3xx_e310_fpga_default-g4bc2c6f.zip
00885 kB / 00885 kB (100%) b2xx_b210_fpga_default-ge57dfe0.zip
[INFO] Images download complete.
```

NOTE: In the above example output, the Images Destination folder is printed:

```
[INFO] Images destination: /usr/local/share/uhd/images
```

To list the N3xx FPGA images with a full path, run the command:

```
$ ls -w 1 /usr/local/share/uhd/images/usrp_n3*.bit

/usr/local/share/uhd/images/usrp_n300_fpga_AA.bit
/usr/local/share/uhd/images/usrp_n300_fpga_HG.bit
/usr/local/share/uhd/images/usrp_n300_fpga_WX.bit
/usr/local/share/uhd/images/usrp_n300_fpga_XG.bit
/usr/local/share/uhd/images/usrp_n310_fpga_AA.bit
/usr/local/share/uhd/images/usrp_n310_fpga_HG.bit
/usr/local/share/uhd/images/usrp_n310_fpga_WX.bit
/usr/local/share/uhd/images/usrp_n310_fpga_XG.bit
```

```
/usr/local/share/uhd/images/usrp_n320_fpga_AQ.bit
/usr/local/share/uhd/images/usrp_n320_fpga_HG.bit
/usr/local/share/uhd/images/usrp_n320_fpga_WX.bit
/usr/local/share/uhd/images/usrp_n320_fpga_XG.bit
/usr/local/share/uhd/images/usrp_n320_fpga_XQ.bit
```

To update the default `HG` variant of FPGA image, run the command:

```
$ uhd_image_loader --args "type=n3xx,addr=<N3xx_IP_ADDR>,fpga=HG"
```

Example Output:

```
uhd_image_loader --args "type=n3xx,addr=192.168.1.151,fpga=HG"
[INFO] [UHD] linux; GNU C++ version 5.4.0 20160609; Boost_105800; UHD_3.11.1.HEAD-0-gad6b0935
[INFO] [MPMD] Initializing 1 device(s) in parallel with args: mgmt_addr=192.168.1.151,type=n3xx,product=n310,serial=313ABDA,claimed=False,
[INFO] [MPM.main] Launching USRP/MPM, version: 3.11.1.0-gunknown
[INFO] [MPM.main] Spawning RPC process...
[INFO] [MPM.PeriphManager] Device serial number: 313ABDA
[INFO] [MPM.PeriphManager] Found 2 daughterboard(s).
[INFO] [MPM.PeriphManager.UDP] No CHDR interfaces found!
[INFO] [MPM.PeriphManager.UDP] No CHDR interfaces found!
[INFO] [MPM.RPCServer] RPC server ready!
[INFO] [MPM.RPCServer] Spawning watchdog task...
[INFO] [MPM.PeriphManager.UDP] No CHDR interfaces found!
[INFO] [MPMD] Claimed device without full initialization.
[INFO] [MPMD IMAGE LOADER] Starting update. This may take a while.
[INFO] [MPM.PeriphManager] Updating component `fpga'
[INFO] [MPM.PeriphManager] Updating component `dts'
[INFO] [MPM.RPCServer] Resetting peripheral manager.
[INFO] [MPM.PeriphManager] Device serial number: 313ABDA
[INFO] [MPM.PeriphManager] Found 2 daughterboard(s).
[INFO] [MPMD IMAGE LOADER] Update component function succeeded.
```

To load a different default FPGA image (i.e. `XG`, `WG`), modify the device argument `fpga=` to a value of `fpga=XG` or `fpga=WG`.

To specify the path to a custom FPGA image, use the ?`--fpga-path`? argument.

```
$ uhd_image_loader --args "type=n3xx,addr=<N3xx_IP_ADDR>" --fpga-path=/path/to/custom/fpga.bit
```

The Verilog code for the FPGA in the USRP N3xx is open-source, and users are free to modify and customize it for their needs. However, certain modifications may result in either bricking the device, or even in physical damage to the unit. Please note that modifications to the FPGA are made at the risk of the user, and may not be covered by the warranty of the device.

It is possible to update the FPGA image when operated in Embedded mode. Connect to the ARM CPU via Serial Console or SSH as detailed in the section above.

Updating the FPGA image from the ARM CPU is the same as detailed above for a Network mode update, except it is not required to provide an `addr` device argument.

```
uhd_image_loader --args "type=n3xx,fpga=HG"
```

```
root@ni-n3xx-313ABDA:~# uhd_image_loader --args "type=n3xx,fpga=HG"
[INFO] [UHD] linux; GNU C++ version 7.2.0; Boost_106400; UHD_3.11.1.0-0-unknown
[INFO] [MPMD] Initializing 1 device(s) in parallel with args: mgmt_addr=127.0.0.1,type=n3xx,product=n310,serial=313ABDA,claimed=False,skip_in
[INFO] [MPMD] Claimed device without full initialization.
[INFO] [MPMD IMAGE LOADER] Starting update. This may take a while.
[INFO] [MPM.PeriphManager] Updating component `fpga'
[INFO] [MPM.PeriphManager] Updating component `dts'
[INFO] [MPM.RPCServer] Resetting peripheral manager.
[INFO] [MPM.PeriphManager] Device serial number: 313ABDA
[INFO] [MPM.PeriphManager] Found 2 daughterboard(s).
[INFO] [MPMD IMAGE LOADER] Update component function succeeded.
```

For more information on updating the FPGA image, refer to the UHD Manual at http://uhd.ettus.com .

The device supports multiple, high-speed, low-latency interfaces on the SFP+ ports for streaming samples to the host computer.

Complete the steps below to set up a streaming connection over the 1 Gigabit Ethernet interface on `SFP Port 0`.

When streaming via SFP Port 0 at 1 Gb speeds, it is important that the connection is direct between the Host and USRP. Placing a switch or other network gear between the Host and USRP can reduce throughput of the transport link. It is also generally recommended to avoid using USB to Ethernet Adapters for the high speed streaming interface, as they may limit performance or cause periodic flow control errors.

NOTE: The `HG` FPGA image must be loaded for `SFP Port 0` to operate at 1Gb speeds. If the `XG` image is loaded, the port will be unresponsive at 1Gb speeds.

1. Configure your Host's Ethernet adapter as shown below. This interface should be separate from the 1Gb NIC/network which is connected to the 1Gb RJ45 management interface.

```
IP Address: 192.168.10.1
Subnet Mask: 255.255.255.0
Gateway: 0.0.0.0
MTU: 1500
```

NOTE: When operating `SFP Port 0` at 1Gb speeds, it is important to set a MTU of `1500` and not a value of `automatic`.

2. Insert the ? RJ45 ? SFP+ adapter ?into? `SFP Port 0`? .

3. Connect the adapter to a host computer using the Ethernet cable to SFP0.

The ? Green LED? above ?`SFP Port 0`? should illuminate.

4. To test the connection,? ?`ping`? the device at address `192.168.10.2?` from the host, as shown below:

```
$ ping 192.168.10.2
```

```
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=1.06 ms
^C
--- 192.168.10.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.065/1.065/1.065/0.000 ms
```

Press CTRL+C to stop the ping program.

Proceed to the next section "Verifying Device Operation".

Complete the steps below to set up a streaming connection over the 10 Gigabit Ethernet interface on SFP Port 1.

NOTE: Both the HG and XG FPGA images support 10Gb speeds over SFP Port 1.

1. Configure your Host's 10Gb Ethernet adapter as shown below.

```
IP Address: 192.168.20.1
Subnet Mask: 255.255.255.0
Gateway: 0.0.0.0
MTU: 9000
```

NOTE: When operating at 10Gb speeds, it is important to set a MTU of 9000 and not a value of automatic.

2. Connect the USRP to a host computer using either a 10Gb SFP or Fiber cable to SFP Port 1.

The ? Green LED? above ?SFP Port 1? should illuminate.

3. To test the connection,? ?ping? the device at address 192.168.20.2? from the host, as shown below:

```
$ ping 192.168.20.2
```

Press CTRL+C to stop the ping program.

Proceed to the next section "Verifying Device Operation".

Complete the steps below to set up a streaming connections over the Dual 10 Gigabit Ethernet interface on SFP Ports 0/1.

NOTE: The XG FPGA image must be loaded for SFP Port 0 to operate at 10 Gb speeds. If the HG image is loaded, the port will be unresponsive at 10 Gb speeds.

1. Configure your Host's #1 10Gb Ethernet adapter as shown below.

```
IP Address: 192.168.10.1
Subnet Mask: 255.255.255.0
Gateway: 0.0.0.0
MTU: 9000
```

2. Configure your Host's #2 10Gb Ethernet adapter as shown below.

```
IP Address: 192.168.20.1
Subnet Mask: 255.255.255.0
Gateway: 0.0.0.0
MTU: 9000
```

NOTE: When operating at 10Gb speeds, it is important to set a MTU of 9000 and not a value of automatic.

3. Connect the USRP to a host computer using either a 10Gb SFP or Fiber cables to SFP Ports 0/1.

The ?Green LEDs? above ?SFP Ports 0/1? should illuminate.

4. To test the SFP Port 0 connection,? ?ping? the device at address 192.168.10.2? from the host, as shown below:

```
$ ping 192.168.10.2
```

Press CTRL+C to stop the ping program.

5. To test the SFP Port 1 connection,? ?ping? the device at address 192.168.20.2? from the host, as shown below:

```
$ ping 192.168.20.2
```

Press CTRL+C to stop the ping program.

Proceed to the next section "Verifying Device Operation".

For more details on Network Setup and Configuration, please see the ?Interfaces and Connectivity? section on the N300/N310 or N320/N321 hardware resources pages.

Once you have successfully setup a management interface and streaming interface, you can now verify the devices operation using the included UHD utilities.

The USRP N300 contains 2 channels, each represented on the front panel as RF0-1. Below is the subdev mapping of RF Ports.

- RF0 = A:0
- RF1 = A:1

The USRP N310 contains 4 channels, each represented on the front panel as `RF0-3`. Below is the `subdev` mapping of RF Ports.

- RF0 = A:0
- RF1 = B:0
- RF2 = C:0
- RF3 = D:0

- RF0 = A:0
- RF1 = A:1
- RF2 = B:0
- RF3 = B:1

The USRP N320 contains 2 channels, each represented on the front panel as `RF0-1`. Below is the `subdev` mapping of RF Ports.

- RF0 = A:0
- RF1 = B:0

The USRP N321 contains 2 channels, each represented on the front panel as `RF0-1`. Below is the `subdev` mapping of RF Ports.

- RF0 = A:0
- RF1 = B:0

Additional details of UHD Subdevice Specifications can be found here in the UHD Manual:
http://files.ettus.com/manual/page_configuration.html#config_subdev

The USRP N300/N310 supports the three fixed Master Clock Rates listed below.

- 122.88 MHz
- 125.00 MHz
- 153.60 MHz

The USRP N320/N321 supports the three fixed Master Clock Rates listed below.

- 200.00 MHz
- 245.76 MHz
- 250.00 MHz

Sample rates as delivered to/from the host computer for USRP devices are constrained to follow several important rules.

It is important to understand that strictly-integer decimation and interpolation are used within USRP hardware to meet the requested sample rate requirements of the application at hand. That means that the desired sample rate must meet the requirement that master-clock-rate/desired-sample-rate be an integer ratio. Further, it is strongly desirable for that ratio to be even. This ratio is the decimation (down-conversion) or interpolation (up-conversion) factor. The decimation or interpolation factor may be between 1 and 1024. There are further constraints on the decimation or interpolation factor. If the decimation or interpolation factor exceeds 128, then it must be evenly divisible by 2. If the decimation or interpolation factor exceeds 256, then it must be evenly divisible by 4.

Listed below are common sample rates for the given master clock rates. This is not a complete listing of the supported sample rates.

Master Clock Rate	**Decimation / Interpolation Rate Host Sample Rate [Msps]**														
1	2	4	6	8	10	12	14	16	18	20	30	32	64	10
122.88e6	61.44e6	30.72e6	20.48e6	15.36e6	12.288e6	10.24e6	8.7771e6	7.68e6	6.8267e6	6.144e6	4.096e6	3.84e6	1.92e6	1.2
125e6	62.5e6	31.25e6	20.833e6	15.625e6	12.5e6	10.417e6	8.9286e6	7.8125e6	6.9444e6	6.25e6	4.1667e6	3.90625e6	1.953125e6	1.2
153.6e6	76.8e6	38.4e6	25.6e6	19.2e6	15.36e6	12.8e6	10.971e6	9.6e6	8.5333e6	7.68e6	5.12e6	4.8e6	2.4e6	1.5

Listed below are common sample rates for the given master clock rates. This is not a complete listing of the supported sample rates.

Master Clock Rate	**Decimation / Interpolation Rate Host Sample Rate [Msps]**														
1	2	4	6	8	10	12	14	16	18	20	30	32	64	100
200e6	100e6	50e6	33.33e6	25e6	20e6	16.66e6	14.2857e6	12.5e6	11.11e6	10e6	6.667e6	6.25e6	3.125e6	2e6
245.76e6	122.88e6	61.44e6	30.72e6	20.48e6	15.36e6	12.288e6	10.24e6	8.7771e6	7.68e6	6.8267e6	6.144e6	4.096e6	3.84e6	1.92
250e6	125e6	62.5e6	31.25e6	20.833e6	15.625e6	12.5e6	10.417e6	8.9286e6	7.8125e6	6.9444e6	6.25e6	4.1667e6	3.90625e6	1.95

Additional information on Sample Rates can be found here in the UHD Manual:
http://files.ettus.com/manual/page_general.html#general_sampleratenotes

The UHD utility `uhd_usrp_probe` provides detailed information of the USRP device.

From your host computer, run the command `uhd_usrp_probe`:

```
$  uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 5.4.0 20160609; Boost_105800; UHD_3.13.1.HEAD-0-ga0a71d10
[INFO] [MPMD] Initializing 1 device(s) in parallel with args: mgmt_addr=192.168.10.2,type=n3xx,product=n310,serial=313ABDA,claimed=False,addr
[INFO] [MPM.main] Launching USRP/MPM, version: 3.13.1.0-gd3b7e90a
[INFO] [MPM.main] Spawning RPC process...
[INFO] [MPM.PeriphManager] Device serial number: 313ABDA
[INFO] [MPM.PeriphManager] Initialized 2 daughterboard(s).
```

```
[INFO] [MPM.PeriphManager] init() called with device args `time_source=internal,clock_source=internal'.
[INFO] [MPM.RPCServer] RPC server ready!
[INFO] [MPM.RPCServer] Spawning watchdog task...
[INFO] [0/DmaFIFO_0] Initializing block control (NOC ID: 0xF1F0D000000000004)
[INFO] [0/DmaFIFO_0] BIST passed (Throughput: 1355 MB/s)
[INFO] [MPM.PeriphManager] init() called with device args `mgmt_addr=192.168.10.2,clock_source=internal,time_source=internal,product=n310'.
[INFO] [0/DmaFIFO_0] BIST passed (Throughput: 1358 MB/s)
[INFO] [0/DmaFIFO_0] BIST passed (Throughput: 1355 MB/s)
[INFO] [0/DmaFIFO_0] BIST passed (Throughput: 1345 MB/s)
[INFO] [0/Radio_0] Initializing block control (NOC ID: 0x12AD100000011312)
[INFO] [0/Radio_1] Initializing block control (NOC ID: 0x12AD100000011312)
[INFO] [0/DDC_0] Initializing block control (NOC ID: 0xDDC0000000000000)
[INFO] [0/DDC_1] Initializing block control (NOC ID: 0xDDC0000000000000)
[INFO] [0/DUC_0] Initializing block control (NOC ID: 0xD0C0000000000002)
[INFO] [0/DUC_1] Initializing block control (NOC ID: 0xD0C0000000000002)
 _____________________________________________________
/
|       Device: N300-Series Device
|     _____________________________________________________
|    /
|    |       Mboard: ni-n3xx-313ABDA
|    |   eeprom_version: 1
|    |   mpm_version: 3.13.1.0-gd3b7e90a
|    |   pid: 16962
|    |   product: n310
|    |   rev: 3
|    |   rpc_connection: remote
|    |   serial: 313ABDA
|    |   type: n3xx
|    |   MPM Version: 1.2
|    |   FPGA Version: 5.2
|    |   RFNoC capable: Yes
|    |
|    |   Time sources:  internal, external, gpsdo, sfp0
|    |   Clock sources: external, internal, gpsdo
|    |   Sensors: gps_tpv, ref_locked, gps_time, gps_locked, temp, gps_sky, fan
|    |     _____________________________________________________
|    |    /
|    |    |       RX Dboard: A
|    |    |     _____________________________________________________
|    |    |    /
|    |    |    |       RX Frontend: 0
|    |    |    |   Name: Magnesium
|    |    |    |   Antennas: TX/RX, RX2, CAL, LOCAL
|    |    |    |   Sensors: lo_locked, ad9371_lo_locked, lowband_lo_locked
|    |    |    |   Freq range: 1.000 to 6000.000 MHz
|    |    |    |   Gain range all: 0.0 to 75.0 step 0.5 dB
|    |    |    |   Gain range rfic: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Gain range dsa: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Gain range amp: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Bandwidth range: 20000000.0 to 100000000.0 step 0.0 Hz
|    |    |    |   Connection Type: IQ
|    |    |    |   Uses LO offset: No
|    |    |     _____________________________________________________
|    |    |    /
|    |    |    |       RX Frontend: 1
|    |    |    |   Name: Magnesium
|    |    |    |   Antennas: TX/RX, RX2, CAL, LOCAL
|    |    |    |   Sensors: lo_locked, ad9371_lo_locked, lowband_lo_locked
|    |    |    |   Freq range: 1.000 to 6000.000 MHz
|    |    |    |   Gain range all: 0.0 to 75.0 step 0.5 dB
|    |    |    |   Gain range rfic: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Gain range dsa: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Gain range amp: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Bandwidth range: 20000000.0 to 100000000.0 step 0.0 Hz
|    |    |    |   Connection Type: IQ
|    |    |    |   Uses LO offset: No
|    |    |     _____________________________________________________
|    |    |    /
|    |    |    |       RX Codec: A
|    |    |    |   Name: AD9371 Dual ADC
|    |    |    |   Gain Elements: None
|    |    |  _____________________________________________________
|    |    | /
|    |    |       RX Dboard: B
|    |    |     _____________________________________________________
|    |    |    /
|    |    |    |       RX Frontend: 0
|    |    |    |   Name: Magnesium
|    |    |    |   Antennas: TX/RX, RX2, CAL, LOCAL
|    |    |    |   Sensors: lo_locked, ad9371_lo_locked, lowband_lo_locked
|    |    |    |   Freq range: 1.000 to 6000.000 MHz
|    |    |    |   Gain range all: 0.0 to 75.0 step 0.5 dB
|    |    |    |   Gain range rfic: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Gain range dsa: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Gain range amp: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Bandwidth range: 20000000.0 to 100000000.0 step 0.0 Hz
|    |    |    |   Connection Type: IQ
|    |    |    |   Uses LO offset: No
|    |    |     _____________________________________________________
|    |    |    /
|    |    |    |       RX Frontend: 1
|    |    |    |   Name: Magnesium
|    |    |    |   Antennas: TX/RX, RX2, CAL, LOCAL
|    |    |    |   Sensors: lo_locked, ad9371_lo_locked, lowband_lo_locked
|    |    |    |   Freq range: 1.000 to 6000.000 MHz
|    |    |    |   Gain range all: 0.0 to 75.0 step 0.5 dB
|    |    |    |   Gain range rfic: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Gain range dsa: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Gain range amp: 0.0 to 0.0 step 0.0 dB
|    |    |    |   Bandwidth range: 20000000.0 to 100000000.0 step 0.0 Hz
|    |    |    |   Connection Type: IQ
|    |    |    |   Uses LO offset: No
|    |    |     _____________________________________________________
|    |    |    /
|    |    |    |       RX Codec: B
|    |    |    |   Name: AD9371 Dual ADC
|    |    |    |   Gain Elements: None
|    |    |  _____________________________________________________
|    |    | /
```

```
|    |    |       TX Dboard: A
|    |    |    ____________________________________________________
|    |    |   /
|    |    |   |       TX Frontend: 0
|    |    |   |   Name: Magnesium
|    |    |   |   Antennas: TX/RX
|    |    |   |   Sensors: lo_locked, ad9371_lo_locked, lowband_lo_locked
|    |    |   |   Freq range: 1.000 to 6000.000 MHz
|    |    |   |   Gain range all: 0.0 to 65.0 step 0.5 dB
|    |    |   |   Gain range rfic: 0.0 to 0.0 step 0.0 dB
|    |    |   |   Gain range dsa: 0.0 to 0.0 step 0.0 dB
|    |    |   |   Gain range amp: 0.0 to 0.0 step 0.0 dB
|    |    |   |   Bandwidth range: 20000000.0 to 100000000.0 step 0.0 Hz
|    |    |   |   Connection Type: IQ
|    |    |   |   Uses LO offset: No
|    |    |    ____________________________________________________
|    |    |   /
|    |    |   |       TX Frontend: 1
|    |    |   |   Name: Magnesium
|    |    |   |   Antennas: TX/RX
|    |    |   |   Sensors: lo_locked, ad9371_lo_locked, lowband_lo_locked
|    |    |   |   Freq range: 1.000 to 6000.000 MHz
|    |    |   |   Gain range all: 0.0 to 65.0 step 0.5 dB
|    |    |   |   Gain range rfic: 0.0 to 0.0 step 0.0 dB
|    |    |   |   Gain range dsa: 0.0 to 0.0 step 0.0 dB
|    |    |   |   Gain range amp: 0.0 to 0.0 step 0.0 dB
|    |    |   |   Bandwidth range: 20000000.0 to 100000000.0 step 0.0 Hz
|    |    |   |   Connection Type: IQ
|    |    |   |   Uses LO offset: No
|    |    |    ____________________________________________________
|    |    |   /
|    |    |   |       TX Codec: A
|    |    |   |   Name: AD9371 Dual DAC
|    |    |   |   Gain Elements: None
|    |    |    ____________________________________________________
|    |    |  /
|    |    |  |       TX Dboard: B
|    |    |  |    ____________________________________________________
|    |    |  |   /
|    |    |  |   |       TX Frontend: 0
|    |    |  |   |   Name: Magnesium
|    |    |  |   |   Antennas: TX/RX
|    |    |  |   |   Sensors: lo_locked, ad9371_lo_locked, lowband_lo_locked
|    |    |  |   |   Freq range: 1.000 to 6000.000 MHz
|    |    |  |   |   Gain range all: 0.0 to 65.0 step 0.5 dB
|    |    |  |   |   Gain range rfic: 0.0 to 0.0 step 0.0 dB
|    |    |  |   |   Gain range dsa: 0.0 to 0.0 step 0.0 dB
|    |    |  |   |   Gain range amp: 0.0 to 0.0 step 0.0 dB
|    |    |  |   |   Bandwidth range: 20000000.0 to 100000000.0 step 0.0 Hz
|    |    |  |   |   Connection Type: IQ
|    |    |  |   |   Uses LO offset: No
|    |    |  |    ____________________________________________________
|    |    |  |   /
|    |    |  |   |       TX Frontend: 1
|    |    |  |   |   Name: Magnesium
|    |    |  |   |   Antennas: TX/RX
|    |    |  |   |   Sensors: lo_locked, ad9371_lo_locked, lowband_lo_locked
|    |    |  |   |   Freq range: 1.000 to 6000.000 MHz
|    |    |  |   |   Gain range all: 0.0 to 65.0 step 0.5 dB
|    |    |  |   |   Gain range rfic: 0.0 to 0.0 step 0.0 dB
|    |    |  |   |   Gain range dsa: 0.0 to 0.0 step 0.0 dB
|    |    |  |   |   Gain range amp: 0.0 to 0.0 step 0.0 dB
|    |    |  |   |   Bandwidth range: 20000000.0 to 100000000.0 step 0.0 Hz
|    |    |  |   |   Connection Type: IQ
|    |    |  |   |   Uses LO offset: No
|    |    |  |    ____________________________________________________
|    |    |  |   /
|    |    |  |   |       TX Codec: B
|    |    |  |   |   Name: AD9371 Dual DAC
|    |    |  |   |   Gain Elements: None
|    |    |  |    ____________________________________________________
|    |    |  |  /
|    |    |  |  |       RFNoC blocks on this device:
|    |    |  |  |
|    |    |  |  |   * DmaFIFO_0
|    |    |  |  |   * Radio_0
|    |    |  |  |   * Radio_1
|    |    |  |  |   * DDC_0
|    |    |  |  |   * DDC_1
|    |    |  |  |   * DUC_0
|    |    |  |  |   * DUC_1
```

```
$ uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 7.3.0; Boost_106600; UHD_3.14.0.0-0-g6875d061
[INFO] [MPMD] Initializing 1 device(s) in parallel with args: mgmt_addr=127.0.0.1,type=n3xx,product=n320,serial=3181FFA,claimed=False
[INFO] [MPM.main] Launching USRP/MPM, version: 3.14.0.0-g6875d061
[INFO] [MPM.main] Spawning RPC process...
[INFO] [MPM.PeriphManager] Device serial number: 3181FFA
[INFO] [MPM.Rhodium-0] Successfully loaded all peripherals!
[INFO] [MPM.Rhodium-1] Successfully loaded all peripherals!
[INFO] [MPM.PeriphManager] Initialized 2 daughterboard(s).
[INFO] [MPM.PeriphManager] No QSFP board detected: Assuming it is disabled in the device tree overlay (e.g., HG, XG images).
[INFO] [MPM.PeriphManager] init() called with device args `time_source=internal,clock_source=internal'.
[INFO] [MPM.Rhodium-0] init() called with args `time_source=internal,clock_source=internal'
[INFO] [MPM.Rhodium-1] init() called with args `time_source=internal,clock_source=internal'
[INFO] [MPM.Rhodium-0.init.LMK04828] LMK initialized and locked!
[INFO] [MPM.Rhodium-1.init.LMK04828] LMK initialized and locked!
[INFO] [MPM.Rhodium-1.DAC37J82] DAC PLL Locked!
[INFO] [MPM.Rhodium-1.AD9695] ADC PLL Locked!
[INFO] [MPM.Rhodium-1.init] JESD204B Link Initialization & Training Complete
[INFO] [MPM.Rhodium-0.DAC37J82] DAC PLL Locked!
[INFO] [MPM.Rhodium-0.AD9695] ADC PLL Locked!
[INFO] [MPM.Rhodium-0.init] JESD204B Link Initialization & Training Complete
[INFO] [MPM.RPCServer] RPC server ready!
```

```
[INFO] [MPM.RPCServer] Spawning watchdog task...
[INFO] [MPM.PeriphManager] init() called with device args `mgmt_addr=127.0.0.1,clock_source=internal,time_source=internal,product=n320'.
[INFO] [MPM.Rhodium-0] init() called with args `mgmt_addr=127.0.0.1,clock_source=internal,time_source=internal,product=n320'
[INFO] [MPM.Rhodium-1] init() called with args `mgmt_addr=127.0.0.1,clock_source=internal,time_source=internal,product=n320'
[INFO] [0/Replay_0] Initializing block control (NOC ID: 0x4E91A00000000004)
[INFO] [0/Radio_0] Initializing block control (NOC ID: 0x12AD100000000320)
[INFO] [0/Radio_1] Initializing block control (NOC ID: 0x12AD100000000320)
[INFO] [0/DDC_0] Initializing block control (NOC ID: 0xDDC0000000000001)
[INFO] [0/DDC_1] Initializing block control (NOC ID: 0xDDC0000000000001)
[INFO] [0/DUC_0] Initializing block control (NOC ID: 0xD0C0000000000000)
[INFO] [0/DUC_1] Initializing block control (NOC ID: 0xD0C0000000000000)
[INFO] [0/FIFO_0] Initializing block control (NOC ID: 0xF1F0000000000000)
[INFO] [0/FIFO_1] Initializing block control (NOC ID: 0xF1F0000000000000)
 _____________________________________________________
/
|       Device: N300-Series Device
|     _____________________________________________________
|    /
|   |       Mboard: ni-n3xx-3181FFA
|   |   eeprom_version: 2
|   |   mpm_version: 3.14.0.0-g6875d061
|   |   pid: 16962
|   |   product: n320
|   |   rev: 6
|   |   rpc_connection: local
|   |   serial: 3181FFA
|   |   type: n3xx
|   |   MPM Version: 1.2
|   |   FPGA Version: 5.3
|   |   FPGA git hash: 3de8954.clean
|   |   RFNoC capable: Yes
|   |
|   |   Time sources:  internal, external, gpsdo, sfp0
|   |   Clock sources: external, internal, gpsdo
|   |   Sensors: gps_tpv, temp, gps_sky, fan, gps_time, gps_locked, ref_locked, gps_gpgga
|   |    _____________________________________________________
|   |   /
|   |   |       RX Dboard: A
|   |   |   ID: Unknown (0x0152)
|   |   |   Serial: 3175A79
|   |   |    _____________________________________________________
|   |   |   /
|   |   |   |       RX Frontend: 0
|   |   |   |   Name: Rhodium
|   |   |   |   Antennas: TX/RX, RX2, CAL, TERM
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 1.000 to 6000.000 MHz
|   |   |   |   Gain range all: 0.0 to 60.0 step 1.0 dB
|   |   |   |   Bandwidth range: 250000000.0 to 250000000.0 step 0.0 Hz
|   |   |   |   Connection Type:
|   |   |   |   Uses LO offset: No
|   |   |    _____________________________________________________
|   |   |   /
|   |   |   |       RX Codec: A
|   |   |   |   Name: ad9695-625
|   |   |   |   Gain Elements: None
|   |   |    _____________________________________________________
|   |   |   /
|   |   |       RX Dboard: B
|   |   |   ID: Unknown (0x0152)
|   |   |   Serial: 3175A67
|   |   |    _____________________________________________________
|   |   |   /
|   |   |   |       RX Frontend: 0
|   |   |   |   Name: Rhodium
|   |   |   |   Antennas: TX/RX, RX2, CAL, TERM
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 1.000 to 6000.000 MHz
|   |   |   |   Gain range all: 0.0 to 60.0 step 1.0 dB
|   |   |   |   Bandwidth range: 250000000.0 to 250000000.0 step 0.0 Hz
|   |   |   |   Connection Type:
|   |   |   |   Uses LO offset: No
|   |   |    _____________________________________________________
|   |   |   /
|   |   |   |       RX Codec: B
|   |   |   |   Name: ad9695-625
|   |   |   |   Gain Elements: None
|   |   |    _____________________________________________________
|   |   |   /
|   |   |       TX Dboard: A
|   |   |   ID: Unknown (0x0152)
|   |   |   Serial: 3175A79
|   |   |    _____________________________________________________
|   |   |   /
|   |   |   |       TX Frontend: 0
|   |   |   |   Name: Rhodium
|   |   |   |   Antennas: TX/RX, CAL, TERM
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 1.000 to 6000.000 MHz
|   |   |   |   Gain range all: 0.0 to 60.0 step 1.0 dB
|   |   |   |   Bandwidth range: 250000000.0 to 250000000.0 step 0.0 Hz
|   |   |   |   Connection Type:
|   |   |   |   Uses LO offset: No
|   |   |    _____________________________________________________
|   |   |   /
|   |   |   |       TX Codec: A
|   |   |   |   Name: dac37j82
|   |   |   |   Gain Elements: None
|   |   |    _____________________________________________________
|   |   |   /
|   |   |       TX Dboard: B
|   |   |   ID: Unknown (0x0152)
|   |   |   Serial: 3175A67
|   |   |    _____________________________________________________
|   |   |   /
|   |   |   |       TX Frontend: 0
|   |   |   |   Name: Rhodium
|   |   |   |   Antennas: TX/RX, CAL, TERM
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 1.000 to 6000.000 MHz
```

```
|   |   |   |   Gain range all: 0.0 to 60.0 step 1.0 dB
|   |   |   |   Bandwidth range: 250000000.0 to 250000000.0 step 0.0 Hz
|   |   |   |   Connection Type:
|   |   |   |   Uses LO offset: No
|   |   |    ----------------------------------------------------
|   |   |   /
|   |   |   |       TX Codec: B
|   |   |   |   Name: dac37j82
|   |   |   |   Gain Elements: None
|   |    ----------------------------------------------------
|   |   /
|   |   |       RFNoC blocks on this device:
|   |   |
|   |   |   * Replay_0
|   |   |   * Radio_0
|   |   |   * Radio_1
|   |   |   * DDC_0
|   |   |   * DDC_1
|   |   |   * DUC_0
|   |   |   * DUC_1
|   |   |   * FIFO_0
|   |   |   * FIFO_1
```

```
$ uhd_usrp_probe
[INFO] [UHD] linux; GNU C++ version 7.3.1 20180712 (Red Hat 7.3.1-6); Boost_106400; UHD_3.14.0.0-0-g6875d061
[INFO] [MPMD] Initializing 1 device(s) in parallel with args: mgmt_addr=192.168.20.2,type=n3xx,product=n320,serial=3166646,claimed=False,addr
[INFO] [MPM.PeriphManager] init() called with device args `time_source=internal,clock_source=internal,product=n320,mgmt_addr=192.168.20.2'.
[INFO] [MPM.Rhodium-0] init() called with args `time_source=internal,clock_source=internal,product=n320,mgmt_addr=192.168.20.2'
[INFO] [MPM.Rhodium-1] init() called with args `time_source=internal,clock_source=internal,product=n320,mgmt_addr=192.168.20.2'
[INFO] [0/Replay_0] Initializing block control (NOC ID: 0x4E91A00000000004)
[INFO] [0/Radio_0] Initializing block control (NOC ID: 0x12AD100000000320)
[INFO] [0/Radio_1] Initializing block control (NOC ID: 0x12AD100000000320)
[INFO] [0/DDC_0] Initializing block control (NOC ID: 0xDDC0000000000001)
[INFO] [0/DDC_1] Initializing block control (NOC ID: 0xDDC0000000000001)
[INFO] [0/DUC_0] Initializing block control (NOC ID: 0xD0C0000000000000)
[INFO] [0/DUC_1] Initializing block control (NOC ID: 0xD0C0000000000000)
[INFO] [0/FIFO_0] Initializing block control (NOC ID: 0xF1F0000000000000)
[INFO] [0/FIFO_1] Initializing block control (NOC ID: 0xF1F0000000000000)
 ----------------------------------------------------
/
|       Device: N300-Series Device
|    ------------------------------------------------------
|   /
|   |       Mboard: ni-n3xx-3166646
|   |   eeprom_version: 2
|   |   mpm_version: 3.14.0.0-g6875d061
|   |   pid: 16962
|   |   product: n320
|   |   rev: 6
|   |   rpc_connection: remote
|   |   serial: 3166646
|   |   type: n3xx
|   |   MPM Version: 1.2
|   |   FPGA Version: 5.3
|   |   FPGA git hash: 3de8954.clean
|   |   RFNoC capable: Yes
|   |
|   |   Time sources:  internal, external, gpsdo, sfp0
|   |   Clock sources: external, internal, gpsdo
|   |   Sensors: gps_sky, gps_time, gps_gpgga, gps_locked, fan, gps_tpv, ref_locked, temp
|   |    ----------------------------------------------------
|   |   /
|   |   |       RX Dboard: B
|   |   |   ID: Unknown (0x0152)
|   |   |   Serial: 316D814
|   |   |    ----------------------------------------------------
|   |   |   /
|   |   |   |       RX Frontend: 0
|   |   |   |   Name: Rhodium
|   |   |   |   Antennas: TX/RX, RX2, CAL, TERM
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 1.000 to 6000.000 MHz
|   |   |   |   Gain range all: 0.0 to 60.0 step 1.0 dB
|   |   |   |   Bandwidth range: 250000000.0 to 250000000.0 step 0.0 Hz
|   |   |   |   Connection Type:
|   |   |   |   Uses LO offset: No
|   |   |    ----------------------------------------------------
|   |   |   /
|   |   |   |       RX Codec: B
|   |   |   |   Name: ad9695-625
|   |   |   |   Gain Elements: None
|   |   |    ----------------------------------------------------
|   |   |   /
|   |   |       RX Dboard: A
|   |   |   ID: Unknown (0x0152)
|   |   |   Serial: 316D810
|   |   |    ----------------------------------------------------
|   |   |   /
|   |   |   |       RX Frontend: 0
|   |   |   |   Name: Rhodium
|   |   |   |   Antennas: TX/RX, RX2, CAL, TERM
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 1.000 to 6000.000 MHz
|   |   |   |   Gain range all: 0.0 to 60.0 step 1.0 dB
|   |   |   |   Bandwidth range: 250000000.0 to 250000000.0 step 0.0 Hz
|   |   |   |   Connection Type:
|   |   |   |   Uses LO offset: No
|   |   |    ----------------------------------------------------
|   |   |   /
|   |   |   |       RX Codec: A
|   |   |   |   Name: ad9695-625
|   |   |   |   Gain Elements: None
|   |    ----------------------------------------------------
|   |   /
```

```
|   |   |       TX Dboard: B
|   |   |   ID: Unknown (0x0152)
|   |   |   Serial: 316D814
|   |   |    ____________________________________________________
|   |   |   /
|   |   |   |       TX Frontend: 0
|   |   |   |   Name: Rhodium
|   |   |   |   Antennas: TX/RX, CAL, TERM
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 1.000 to 6000.000 MHz
|   |   |   |   Gain range all: 0.0 to 60.0 step 1.0 dB
|   |   |   |   Bandwidth range: 250000000.0 to 250000000.0 step 0.0 Hz
|   |   |   |   Connection Type:
|   |   |   |   Uses LO offset: No
|   |   |    ____________________________________________________
|   |   |   /
|   |   |   |       TX Codec: B
|   |   |   |   Name: dac37j82
|   |   |   |   Gain Elements: None
|   |    ___________________________________________________
|   |   /
|   |   |       TX Dboard: A
|   |   |   ID: Unknown (0x0152)
|   |   |   Serial: 316D810
|   |   |    ____________________________________________________
|   |   |   /
|   |   |   |       TX Frontend: 0
|   |   |   |   Name: Rhodium
|   |   |   |   Antennas: TX/RX, CAL, TERM
|   |   |   |   Sensors: lo_locked
|   |   |   |   Freq range: 1.000 to 6000.000 MHz
|   |   |   |   Gain range all: 0.0 to 60.0 step 1.0 dB
|   |   |   |   Bandwidth range: 250000000.0 to 250000000.0 step 0.0 Hz
|   |   |   |   Connection Type:
|   |   |   |   Uses LO offset: No
|   |   |    ____________________________________________________
|   |   |   /
|   |   |   |       TX Codec: A
|   |   |   |   Name: dac37j82
|   |   |   |   Gain Elements: None
|   |    ___________________________________________________
|   |   /
|   |   |       RFNoC blocks on this device:
|   |   |
|   |   |   * Replay_0
|   |   |   * Radio_0
|   |   |   * Radio_1
|   |   |   * DDC_0
|   |   |   * DDC_1
|   |   |   * DUC_0
|   |   |   * DUC_1
|   |   |   * FIFO_0
|   |   |   * FIFO_1
```

If you see warnings such as:

```
[WARNING] [UDP] The recv buffer could not be resized sufficiently.
```

You need to resize the socket buffers for your network interface card:

```
sudo sysctl -w net.core.rmem_max=288000
sudo sysctl -w net.core.wmem_max=288000
sudo sysctl -w net.core.rmem_max=33554432
```

The UHD driver includes several example programs, which may serve as test programs or the basis for your application program. The source code can be obtained from the UHD repository on github at: https://github.com/EttusResearch/uhd/tree/master/host/examples

You can quickly verify the operation of your USRP N3xx by running the `rx_ascii_art_dft` UHD example program.

The `rx_ascii_art_dft` utility is a simple console –based, real-time FFT display tool. It is not graphical in nature, so it can be easily run over an SSH connection within a terminal window, and does not need any graphical capability, such as X Windows, to be installed. It can also be run over a serial console connection, although this is not recommended, as the formatting may not render correctly.

You can run a simple test of the N3xx USRP by connecting an antenna and observing the spectrum of a commercial FM radio station in real-time, following the steps below:

1. Attach an antenna to the `Ch0/RX2`– antenna port of the N3xx.

2. From your host computer, run the command:

N300/N310

```
$ /usr/local/lib/uhd/examples/rx_ascii_art_dft --args "master_clock_rate=125e6,mgmt_addr=192.168.1.151,addr=192.168.10.2" --freq 98.5e6 --rat
```

N320/N321

```
$ /usr/local/lib/uhd/examples/rx_ascii_art_dft --args "master_clock_rate=250e6,mgmt_addr=192.168.1.151,addr=192.168.10.2" --freq 98.5e6 --rat
```

NOTE: Modify the command– line argument `freq` ?above to specify a tuning frequency for a strong local FM radio station. You will also need to update the IP Address to match your device IP.

3. You should see a real-time FFT display of 2.5 MHz of spectrum, centered at the specified tuning frequency.

4. Type "`Q`" or `Ctrl--C` to stop the program and to return to the Linux command line.

5. You can run with the `?----help` ?argument to see a description of all available command-line options.

Example Output:

```
$ /usr/local/lib/uhd/examples/rx_ascii_art_dft --args "master_clock_rate=125e6,mgmt_addr=192.168.1.151,addr=192.168.10.2" --freq 98.5e6 --rat
```

```
Creating the usrp device with: master_clock_rate=125e6,mgmt_addr=192.168.1.151,addr=192.168.10.2...
[INFO] [UHD] linux; GNU C++ version 5.4.0 20160609; Boost_105800; UHD_3.11.1.HEAD-0-gad6b0935
[INFO] [MPMD] Initializing 1 device(s) in parallel with args: mgmt_addr=192.168.1.151,type=n3xx,product=n310,serial=313ABDA,claimed=False,mas
[INFO] [MPM.main] Launching USRP/MPM, version: 3.11.1.0-gunknown
[INFO] [MPM.main] Spawning RPC process...
[INFO] [MPM.PeriphManager] Device serial number: 313ABDA
[INFO] [MPM.PeriphManager] Found 2 daughterboard(s).
[INFO] [MPM.RPCServer] RPC server ready!
[INFO] [MPM.RPCServer] Spawning watchdog task...
[INFO] [MPM.PeriphManager] init() called with device args `mgmt_addr=192.168.1.151,product=n310,master_clock_rate=125e6'.
[INFO] [0/DmaFIFO_0] Initializing block control (NOC ID: 0xF1F0D000000000004)
[INFO] [0/DmaFIFO_0] BIST passed (Throughput: 1336 MB/s)
[INFO] [0/DmaFIFO_0] BIST passed (Throughput: 1338 MB/s)
[INFO] [0/DmaFIFO_0] BIST passed (Throughput: 1346 MB/s)
[INFO] [0/DmaFIFO_0] BIST passed (Throughput: 1350 MB/s)
[INFO] [0/Radio_0] Initializing block control (NOC ID: 0x12AD100000000310)
[INFO] [0/Radio_1] Initializing block control (NOC ID: 0x12AD100000000310)
[INFO] [0/Radio_2] Initializing block control (NOC ID: 0x12AD100000000310)
[INFO] [0/Radio_3] Initializing block control (NOC ID: 0x12AD100000000310)
[INFO] [0/DDC_0] Initializing block control (NOC ID: 0xDDC0000000000001)
[INFO] [0/DDC_1] Initializing block control (NOC ID: 0xDDC0000000000001)
[INFO] [0/DDC_2] Initializing block control (NOC ID: 0xDDC0000000000001)
[INFO] [0/DDC_3] Initializing block control (NOC ID: 0xDDC0000000000001)
[INFO] [0/DUC_0] Initializing block control (NOC ID: 0xD0C0000000000000)
[INFO] [0/DUC_1] Initializing block control (NOC ID: 0xD0C0000000000000)
[INFO] [0/DUC_2] Initializing block control (NOC ID: 0xD0C0000000000000)
[INFO] [0/DUC_3] Initializing block control (NOC ID: 0xD0C0000000000000)
Using Device: Single USRP:
  Device: N300-Series Device
  Mboard 0: ni-n3xx-313ABDA
  RX Channel: 0
    RX DSP: 0
    RX Dboard: A
    RX Subdev: Magnesium
  TX Channel: 0
    TX DSP: 0
    TX Dboard: A
    TX Subdev: Magnesium
  TX Channel: 1
    TX DSP: 0
    TX Dboard: B
    TX Subdev: Magnesium
  TX Channel: 2
    TX DSP: 0
    TX Dboard: C
    TX Subdev: Magnesium
  TX Channel: 3
    TX DSP: 0
    TX Dboard: D
    TX Subdev: Magnesium

Setting RX Rate: 2.500000 Msps...
Actual RX Rate: 2.500000 Msps...

Setting RX Freq: 98.500000 MHz...
Actual RX Freq: 98.500000 MHz...

Setting RX Gain: 50.000000 dB...
Actual RX Gain: 50.000000 dB...

Checking RX: all_los: locked ...

Done!
```

Included with the UHD driver example programs is a utility, `benchmark_rate` to benchmark the transport link of the system.

A system's maximum performance is dependent upon many factors. `benchmark_rate` will exercise the transport link and CPU of the system.

NOTE: This example requires the `HG` FPGA image to be loaded.

N300/N310

This example will test one full-duplex stream using "RF0/A:0", at a rate of 3.84 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.10.2,master_clock_rate=122.88e6" \
--duration 60 \
--channels "0" \
--rx_rate 3.84e6 \
--rx_subdev "A:0" \
--tx_rate 3.84e6 \
--tx_subdev "A:0"
```

N310

This example will test four full-duplex streams at 1.25 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.10.2,master_clock_rate=125e6" \
--duration 60 \
--channels "0,1,2,3" \
--rx_rate 1.25e6 \
--rx_subdev "A:0 A:1 B:0 B:1" \
--tx_rate 1.25e6 \
--tx_subdev "A:0 A:1 B:0 B:1"
```

N320/N321

This example will test one full-duplex stream using "RF0/A:0", at a rate of 3.84 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.10.2,master_clock_rate=245.76e6" \
```

```
--duration 60 \
--channels "0" \
--rx_rate 3.84e6 \
--rx_subdev "A:0" \
--tx_rate 3.84e6 \
--tx_subdev "A:0"
```

When streaming samples over a 1 Gb transport link, the maximum accumulative rate for all channels is 25 MS/s with a `sc16` OTW format. To achieve higher streaming rates, it is recommended to use the 10 Gb interfaces.

NOTE: This example will work with either the `HG` or `XG` FPGA image.

N300/N310

This example will test one full-duplex stream using "RF0/A:0", at a rate of 31.25 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.20.2,master_clock_rate=125e6" \
--duration 60 \
--channels "0" \
--rx_rate 31.25e6 \
--rx_subdev "A:0" \
--tx_rate 31.25e6 \
--tx_subdev "A:0"
```

N320/N321

This example will test one full-duplex stream using "RF0/A:0", at a rate of 31.25 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.20.2,master_clock_rate=250e6" \
--duration 60 \
--channels "0" \
--rx_rate 31.25e6 \
--rx_subdev "A:0" \
--tx_rate 31.25e6 \
--tx_subdev "A:0"
```

N310

This example will test four full-duplex streams at 30.72 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.20.2,master_clock_rate=122.88e6" \
--duration 60 \
--channels "0,1,2,3" \
--rx_rate 30.72e6 \
--rx_subdev "A:0 A:1 B:0 B:1" \
--tx_rate 30.72e6 \
--tx_subdev "A:0 A:1 B:0 B:1"
```

N320/N321

This example will test two full-duplex streams at 30.72 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.20.2,master_clock_rate=245.76e6" \
--duration 60 \
--channels "0,1,2,3" \
--rx_rate 30.72e6 \
--rx_subdev "A:0 B:0" \
--tx_rate 30.72e6 \
--tx_subdev "A:0 B:0"
```

NOTE: This example requires the `XG` FPGA image to be loaded.

N310

This example will test four full-duplex streams at 62.5 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.10.2,second_addr=192.168.20.2,master_clock_rate=125e6" \
--duration 60 \
--channels "0,1,2,3" \
--rx_rate 62.5e6 \
--rx_subdev "A:0 A:1 B:0 B:1" \
--tx_rate 62.5e6 \
--tx_subdev "A:0 A:1 B:0 B:1"
```

N320/N321

This example will test two full-duplex streams at 62.5 MS/s, for 60 seconds:

```
/usr/local/lib/uhd/examples/benchmark_rate  \
--args "type=n3xx,mgmt_addr=192.168.1.151,addr=192.168.10.2,second_addr=192.168.20.2,master_clock_rate=250e6" \
--duration 60 \
--channels "0,1,2,3" \
--rx_rate 62.5e6 \
--rx_subdev "A:0 B:0" \
--tx_rate 62.5e6 \
--tx_subdev "A:0 B:0"
```

- Using Ethernet-Based Synchronization on the USRP? N3xx Devices

- USRP N320/N321 LO Distribution
- 5G NR EVM Measurements with the USRP N320/N321

To avoid damaging the file system and causing any corruption, do not turn the device off with the power button without first shutting down the system. Use this command to cleanly and properly shut the system down:

```
shutdown --h now
```

Auto booting of the N3xx when power is applied can be configured by enabling the flag on the device's EEPROM with the following command:

```
eeprom-set-flags 0x1
```

The default user is `root` and the password is empty (no password).

It is recommended to update the `root` password, which can be done with the command `passwd`:

Example Output:

```
root@ni-n3xx-serial:~# passwd
Changing password for root
New password:
Re-enter new password:
passwd: password changed.
```

Technical support for USRP hardware is available through email only. If the product arrived in a non−functional state or you require technical assistance, please contact support@ettus.com. Please allow 24 to 48 hours for response by email, depending on holidays and weekends, although we are often able to reply more quickly than that.

We also recommend that you subscribe to the community mailing lists. The mailing lists have a responsive and knowledgeable community of hundreds of developers and technical users who are located around the world. When you join the community, you will be connected to this group of people who can help you learn about SDR and respond to your technical and specific questions. Often your question can be answered quickly on the mailing lists. Each mailing list also provides an archive of all past conversations and discussions going back many years. Your question or problem may have already been addressed before, and a relevant or helpful solution may already exist in the archive.

Discussions involving the USRP hardware and the UHD software itself are best addressed through the **u?srp−-users** ?mailing list at http://usrp-users.ettus.com.

Discussions involving the use of GNU Radio with USRP hardware and UHD software are best addressed through the **d?iscuss−-gnuradio**? mailing list at https://lists.gnu.org/mailman/listinfo/discuss−gnuradio?.

Discussions involving the use of OpenBTS® with USRP hardware and UHD software are best addressed through the **o?penbts−-discuss?** mailing list at https://lists.sourceforge.net/lists/listinfo/openbts−discuss?.?

The support page on our website is located at https://www.ettus.com/support?. The Knowledge Base is located at ?https://kb.ettus.com?.

Every country has laws governing the transmission and reception of radio signals. Users are solely responsible for insuring they use their USRP system in compliance with all applicable laws and regulations. Before attempting to transmit and/or receive on any frequency, we recommend that you determine what licenses may be required and what restrictions may apply.

- NOTE: This USRP product is a piece of test equipment.

If you have any non−-technical questions related to your order, then please contact us by email at orders@ettus.com?, or by phone at +1−408−610−6399 (Monday-Friday, 8 AM - 5 PM, Pacific Time). Please be sure to include your order number and the serial number of your USRP.

Terms and conditions of sale can be accessed online at the following link: http://www.ettus.com/legal/terms-and-conditions-of-sale