N310 Getting Started Guides

Contents

- USRP N300/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

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.

It is strongly? recommended to use the latest stable version of UHD. It is generally recommended to use the maintenance branch of the latest stable version, for example `UHD-3.13`.

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.

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.

It is recommended to use the UHD-3.13 maintenance branch at the time of this writing.

NOTE: 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 E320 USRP must match.

Before operating the device, it is? ?strongly? recommended to re-image the microSD card with the latest version of the Embedded Linux file-system. The image should match the version of UHD installed on the host computer.

To obtain the file system SD card image, run the following command on the host computer with Internet access:

```
$ sudo uhd_images_downloader -t sdimg -v
```

Example Output:

```
$ sudo uhd_images_downloader -t sdimg -v
[INFO] Images destination: /usr/local/share/uhd/images
[INFO] No inventory file found at /usr/local/share/uhd/images/inventory.json. Creating an empty one.
845962 kB / 845962 kB (100%) n3xx_common_sdimg_default-v3.11.1.0.zip
[INFO] Images download complete.
```

NOTE: In the output of the command, the folder destination where the images are saved is printed out.

An alternative method to identify your installation prefix is to run the command:

```
$ uhd_config_info --install-prefix
```

Example Output:

```
$ uhd_config_info --install-prefix
Install prefix: /usr/local
```

The default folder location for FPGA and SD card images is:

```
<UHD_INSTALL_PREFIX>/share/uhd/images/
```

Verify the SD card image is at this location:

```
$ ls /usr/local/share/uhd/images/*.sdimg
```

Example Output:

```
/usr/local/share/uhd/images/usrp_n3xx_fs.sdimg
```

Insert the microSD card into the host computer.

To identify the device where the microSD card is, run the command:

```
dmesg | tail
```

Example Output (partially truncated for readability):

```
[21265.575488] usb-storage 1-2:1.0: USB Mass Storage device detected
[21266.586983] scsi 0:0:0:0: Direct-Access     Generic  Mass-Storage     1.11 PQ: 0 ANSI: 2
[21266.588024] sd 0:0:0:0: Attached scsi generic sg0 type 0
[21267.299812] sd 0:0:0:0: [sdb] 31116288 512-byte logical blocks: (15.9 GB/14.8 GiB)
[21267.302687]  sdb: sdb1 sdb2 sdb3 sdb4
```

NOTE: In this specific example configuration, the SD card has been attached to `sdb`.

Another method to finding the device node the disk is attached at is to use the Linux utility `lsblk`:

Example Output:

```
$ lsblk
NAME           MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sdb              8:0    1  14.9G  0 disk
??sdb4           8:4    1   128M  0 part  /media/user/data
??sdb2           8:2    1   7.3G  0 part  /media/user/primary
??sdb3           8:3    1   7.3G  0 part  /media/user/secondary
??sdb1           8:1    1    16M  0 part  /media/user/boot
```

Some operating systems by default will auto-mount the partitions on a block device when it is attached. Before writing a new disk image to the SD card, you should first unmount any mounted partitions. This can be done with the Linux utility `umount` as shown below:

```
$ sudo umount /media/user/data
$ sudo umount /media/user/primary
$ sudo umount /media/user/secondary
$ sudo umount /media/user/boot
```

Running the command `lsblk` again will show these partitions have been unmounted:

Example Output:

```
$ lsblk
NAME           MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sdb              8:0    1  14.9G  0 disk
??sdb4           8:4    1   128M  0 part
??sdb2           8:2    1   7.3G  0 part
??sdb3           8:3    1   7.3G  0 part
??sdb1           8:1    d1    16M  0 part
```

WARNING: The Linux utility `dd` can cause unrecoverable data loss if the incorrect disk is selected, or if the parameters are input incorrectly. Ensure you have selected the correct input and output parameters for your system configuration.

NOTE: You must use a 16GB or larger SD card.

The ?`<SD_CARD_DEV_NAME>?` device node depends on your operating system and which other devices are plugged in. Typical values are ?`sdb`? or `mmcblk0?`.

In the examples above:

```
<IMAGE>=/usr/local/share/uhd/images/usrp_n3xx_fs.sdimg
```

```
<SD_CARD_DEV_NAME>=/dev/sdb
```

Write the disk image with the command:

```
$ sudo dd if=<IMAGE> of=<SD_CARD_DEV_NAME> bs=1M
```

The step of writing the disk image to the SD card can take a while to complete.

Example Output:

```
$ sudo dd if=/usr/local/share/uhd/images/usrp_n3xx_fs.sdimg of=/dev/sdb bs=1M
15160+0 records in
15160+0 records out
15896412160 bytes (16 GB, 15 GiB) copied, 1160.93 s, 13.7 MB/s
```

To ensure the disk is synchronized, run the `sync` command:

```
$ sync
```

You can now remove the microSD card from your Host computer and insert it into the USRP N310.

For more information on updating the file-system, refer to the UHD Manual at ?http://uhd.ettus.com.

Listed below are the interfaces to connect to the USRP N300/N310. 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 "Host mode", this interface can optionally be used for UHD management traffic. Regardless of the operation mode (Host 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.

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 N300/N310, 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 8000 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 8000 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 #>:~#
```

The USRP N300/N310 systemd network configuration files are located at: `/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=v4

[DHCPv4]
UseHostname=false
```

`sfp0 (static):`

```
[Match]
Name=sfp0

[Network]
Address=192.168.10.2/24

[Link]
MTUBytes=8000
```

`sfp1 (static):`

```
[Match]
Name=sfp1

[Network]
Address=192.168.20.2/24

[Link]
MTUBytes=8000
```

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.

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 Host 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 -v
```

Example Output:

```
$ sudo uhd_images_downloader -v
[INFO] Images destination: /usr/local/share/uhd/images
00006 kB / 00006 kB (100%) usrp1_b100_fw_default-g6bea23d.zip
11698 kB / 11698 kB (100%) n3xx_n310_fpga_default-g61cdf981.zip
02757 kB / 02757 kB (100%) usrp2_n210_fpga_default-g6bea23d.zip
02076 kB / 02076 kB (100%) n230_n230_fpga_default-g61cdf981.zip
00522 kB / 00522 kB (100%) usrp1_b100_fpga_default-g6bea23d.zip
00465 kB / 00465 kB (100%) b2xx_b200_fpga_default-g1c568e6.zip
02415 kB / 02415 kB (100%) usrp2_n200_fpga_default-g6bea23d.zip
29462 kB / 29462 kB (100%) x3xx_x310_fpga_default-g61cdf981.zip
00517 kB / 00517 kB (100%) b2xx_b205mini_fpga_default-g1c568e6.zip
28343 kB / 28343 kB (100%) x3xx_x300_fpga_default-g61cdf981.zip
00017 kB / 00017 kB (100%) octoclock_octoclock_fw_default-g14000041.zip
04839 kB / 04839 kB (100%) usb_common_windrv_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-g14000041.zip
00473 kB / 00473 kB (100%) b2xx_b200mini_fpga_default-g1c568e6.zip
00319 kB / 00319 kB (100%) usrp1_usrp1_fpga_default-g6bea23d.zip
00009 kB / 00009 kB (100%) usrp2_n210_fw_default-g6bea23d.zip
```

```
08058 kB / 08058 kB (100%) n3xx_n300_fpga_default-g61cdf981.zip
04442 kB / 04442 kB (100%) e3xx_e310_fpga_default-g61cdf981.zip
00859 kB / 00859 kB (100%) b2xx_b210_fpga_default-g1c568e6.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 N310 FPGA images with a full path, run the command:

```
$ ls -w 1 /usr/local/share/uhd/images/usrp_n31*.bit

/usr/local/share/uhd/images/usrp_n310_fpga_HG.bit
/usr/local/share/uhd/images/usrp_n310_fpga_XG.bit
```

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

```
$ uhd_image_loader --args "type=n3xx,addr=<N310_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=<N310_IP_ADDR>" --fpga-path=/path/to/custom/fpga.bit
```

The Verilog code for the FPGA in the USRP N300/N310 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.

Run the command `uhd_images_downloader` to download the FPGA images to the device's file system:

NOTE: The 1 GB RJ45 management interface will require Internet access for the next step.

```
# uhd_images_downloader
[INFO] Images destination: /usr/share/uhd/images
[INFO] No inventory file found at /usr/share/uhd/images/inventory.json. Creating an empty one.
00006 kB / 00006 kB (100%) usrp1_b100_fw_default-g6bea23d.zip
11698 kB / 11698 kB (100%) n3xx_n310_fpga_default-g61cdf981.zip
02757 kB / 02757 kB (100%) usrp2_n210_fpga_default-g6bea23d.zip
02076 kB / 02076 kB (100%) n230_n230_fpga_default-g61cdf981.zip
00522 kB / 00522 kB (100%) usrp1_b100_fpga_default-g6bea23d.zip
00465 kB / 00465 kB (100%) b2xx_b200_fpga_default-g1c568e6.zip
02415 kB / 02415 kB (100%) usrp2_n200_fpga_default-g6bea23d.zip
29462 kB / 29462 kB (100%) x3xx_x310_fpga_default-g61cdf981.zip
00517 kB / 00517 kB (100%) b2xx_b205mini_fpga_default-g1c568e6.zip
28343 kB / 28343 kB (100%) x3xx_x300_fpga_default-g61cdf981.zip
00017 kB / 00017 kB (100%) octoclock_octoclock_fw_default-g14000041.zip
04839 kB / 04839 kB (100%) usb_common_windrv_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-g14000041.zip
00473 kB / 00473 kB (100%) b2xx_b200mini_fpga_default-g1c568e6.zip
00319 kB / 00319 kB (100%) usrp1_usrp1_fpga_default-g6bea23d.zip
00009 kB / 00009 kB (100%) usrp2_n210_fw_default-g6bea23d.zip
08058 kB / 08058 kB (100%) n3xx_n300_fpga_default-g61cdf981.zip
04442 kB / 04442 kB (100%) e3xx_e310_fpga_default-g61cdf981.zip
00859 kB / 00859 kB (100%) b2xx_b210_fpga_default-g1c568e6.zip
[INFO] Images download complete.
```

NOTE: The default UHD FPGA Images destination within the N300/N310's file-system is `/usr/share/uhd/images`. The default UHD FPGA Images destination on a typical host installation is `/usr/local/share/uhd/images`.

Updating the FPGA image from the ARM CPU is the same as detailed above for a Host mode update:

```
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: 8000
```

NOTE: When operating `SFP Port 0` at 1Gb speeds, it is important to set a MTU of `8000` 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: 8000
```

NOTE: When operating at 10Gb speeds, it is important to set a MTU of `8000` 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: 8000
```

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: 8000
```

NOTE: When operating at 10Gb speeds, it is important to set a MTU of `8000` 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 hardware resources page located within the Ettus Research Knowledge Base at ?https://kb.ettus.com/N300/N310? .

Once you have successfully setup a management interface and streaming interface, you can now verify the devices operation using the include 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

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

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

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: 0xF1F0D00000000004)
[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
```

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 N300/N310 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 N300/N310 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 N310.

2. From your host computer, run the command:

```
$ /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
```

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: 0xF1F0D00000000004)
[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.

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"
```

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"
```

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.

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"
```

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"
```

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

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"
```

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

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 N310 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