



## UHD Compatible RF frontends

Version: 2024-12-23

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Compatibility .....	1
<b>2</b>	<b>Installation with the Amarisoft software .....</b>	<b>2</b>
2.1	Introduction .....	2
2.2	Automatic Installation .....	2
2.2.1	Amarisoft OTS .....	2
2.2.2	Direct install .....	2
2.3	Manual Installation .....	2
2.3.1	UHD .....	2
2.3.2	TRX driver .....	2
2.3.2.1	Compile your driver .....	2
2.3.3	Configuration file .....	3
<b>3</b>	<b>Configuration .....</b>	<b>4</b>
<b>4</b>	<b>Frontends .....</b>	<b>5</b>
4.1	USRP N200/N210 setup .....	5
4.2	USRP B200/B210 setup .....	5
4.3	USRP X300/X310 setup .....	6

# 1 Introduction

## 1.1 Compatibility

Amarisoft components are compatible with following UHD devices:

- Ettus USRP n2x0 series
- Ettus USRP b2x0 series
- Ettus USRP x3x0 series
- Ettus USRP n3x0 series

Compatibility have been tested up to UHD 3.x versions

## 2 Installation with the Amarisoft software

### 2.1 Introduction

You have two ways to install the TRX driver for the UHD compatible RF frontends:

- automatic
- manual

Notes:

- these methods would install some packages, so make sure you have root privileges when you run the script.
- this install creates a symlink for the TRX driver so please do not remove this directory afterwards.

### 2.2 Automatic Installation

There are two automatic methods available.

Both are only available on Fedora and Ubuntu distribution.

#### 2.2.1 Amarisoft OTS

If you are using Amarisoft OTS install, i.e using the `install.sh` script provided with your Amarisoft software components, you'll be asked during process to select proper RF frontend model.

#### 2.2.2 Direct install

To start your automatic install, use the following command where `<path>` is the path to the directory where you have already installed your LTE component (eNB or UE) and `type` should be set to `enb` or `ue` accordingly.

```
./install <path> <type>
```

Then, inside `<path>`, select your frontend model:

```
./config/rf_select.sh <model>
```

Where `<model>` can be `n2x0`, `b2x0`, `x3x0` or `n3x0`.

### 2.3 Manual Installation

#### 2.3.1 UHD

Follow Ettus documentation to install UHD driver: [https://files.ettus.com/manual/page\\_build\\_guide.html](https://files.ettus.com/manual/page_build_guide.html).

#### 2.3.2 TRX driver

##### 2.3.2.1 Compile your driver

As Amarisoft provides UHD TRX driver source code, you need to compile it locally.

You need development package to be installed as explained on Ettus webpage. Then just type:

```
make
```

This will create a `trx_uhd.so` that you need to copy inside your Amarisoft component (`eNB` or `UE`) directory.

### 2.3.3 Configuration file

Config files are located inside `config.ue` and `config.enb` directories. Depending on the component you want to use copy the proper directory inside your Amarisoft component config subdirectory:

```
cp -r config.xx <path>/config/rf_driver
```

Note that destination name must be `rf_driver`.

### 3 Configuration

The following properties are available:

<b>args</b>	String. Set the UHD specific arguments (they are defined in the UHD documentation).
<b>sync</b>	Optional string. Select the synchronization type. <b>none</b> is the default value and selects the internal clock. <b>external</b> selects an external clock and PPS. <b>mimo</b> selects the N2x0 MIMO cable. <b>external_clock</b> selects an external clock but keeps the internal PPS.
<b>dl_sample_bits</b>	Optional integer (default = 16). Set the number of bits per DL sample. Allowed values are 8, 12 or 16.
<b>ul_sample_bits</b>	Optional integer (default = 16). Set the number of bits per UL sample. Allowed values are 8, 12 or 16 (Use 8 bits carefully).
<b>tx_subdev</b>	Optional string. Sets tx subdev as specified in UHD API ( <a href="http://files.ettus.com/manual/classuhd_1_1usrp_1_1multi__usrp.html#a3b8d9d9fb9a1ec51e81a207cd299e517">http://files.ettus.com/manual/classuhd_1_1usrp_1_1multi__usrp.html#a3b8d9d9fb9a1ec51e81a207cd299e517</a> ).
<b>rx_subdev</b>	Optional string. Sets rx subdev as specified in UHD API ( <a href="http://files.ettus.com/manual/classuhd_1_1usrp_1_1multi__usrp.html#a7f94ed00059cc7dd30567d031b3f9679">http://files.ettus.com/manual/classuhd_1_1usrp_1_1multi__usrp.html#a7f94ed00059cc7dd30567d031b3f9679</a> ).

## 4 Frontends

### 4.1 USRP N200/N210 setup

The radio front end must be connected to one gigabit Ethernet port (don't use a switch to connect them to avoid potential packet losses).

If you have 2 radio frontends for MIMO setup you may connect second device to first one using MIMO cable or connect it to another gigabit Ethernet port (MIMO cable remains necessary).

- Configure the Ethernet interface for the USRP. The parameters are:
  - IPv4 address: 192.168.10.1
  - Subnet mask: 255.255.255.0
  - MTU size: 4000

Verify that you can ping the USRP device. Its default IP address is 192.168.10.2:

```
ping 192.168.10.2
```

The large MTU size is needed to use *Jumbo frames* which optimizes the Ethernet bandwidth.

For following steps, you should change network stack buffer sizes (Run as **root**):

```
sysctl -w net.core.rmem_max=50000000
sysctl -w net.core.wmem_max=1048576
```

- Check that the USRP is visible:

```
uhd_usrp_probe --args addr=192.168.10.2
```

It should print the various capabilities of the USRP device. You can send and receive some baseband samples to test the speed of the Ethernet connection:

```
/usr/share/uhd/examples/benchmark_rate --args \
    addr=192.168.10.2,send_frame_size=3972,recv_frame_size=3972 \
    --tx_rate 12500000 --rx_rate 12500000 --duration 30
```

NB: you may find **benchmark\_rate** there:

- Fedora 19/20: /usr/lib64/uhd/examples/benchmark\_rate
- Ubuntu 14.04: /usr/lib/uhd/examples/benchmark\_rate

The test runs during 30 seconds. There should be no error, overflow nor underflow.

- If you did not already do it before, calibrate your device. Run as **root** in this order:

```
uhd_cal_rx_iq_balance --verbose --args addr=192.168.10.2
```

```
uhd_cal_tx_iq_balance --verbose --args addr=192.168.10.2
```

```
uhd_cal_tx_dc_offset --verbose --args addr=192.168.10.2
```

It takes a few minutes for each command.

- MIMO. In order to simplify the setup, you can use the MIMO cable which synchronizes the USRP clocks and allows one USRP to act as an Ethernet switch for the other. Else you need to connect each USRP on a different network interface.

### 4.2 USRP B200/B210 setup

The USRP B200/B210 are supported, but on some PCs the USB3 interface does not work reliably. Moreover, there are still evolutions in the corresponding UHD driver.

We recommend the following steps to try it:

1. Check that the B2x0 is available and upload the firmware:

```
uhd_usrp_probe
```

2. Check that USB3 interface is working:

```
/usr/share/uhd/examples/benchmark_rate --args master_clock_rate=11.52e6 \  
--tx_rate 11520000 --rx_rate 11520000 --duration 30
```

The test runs during 30 seconds. There should be no error, overflow nor underflow.

NB: You should take a look at configuration file in `config/b2x0` as it may require some changes depending on desired bandwidth.

### 4.3 USRP X300/X310 setup

The USRP X300/X310 are supported. The setup is very similar to the N200/N210. The UHD driver version  $\geq 3.7.0$  must be installed. The X3x0 has a specific support for LTE sample rates.

NB: You should take a look at configuration file in `config/x3x0` as it may require some changes depending on desired bandwidth.