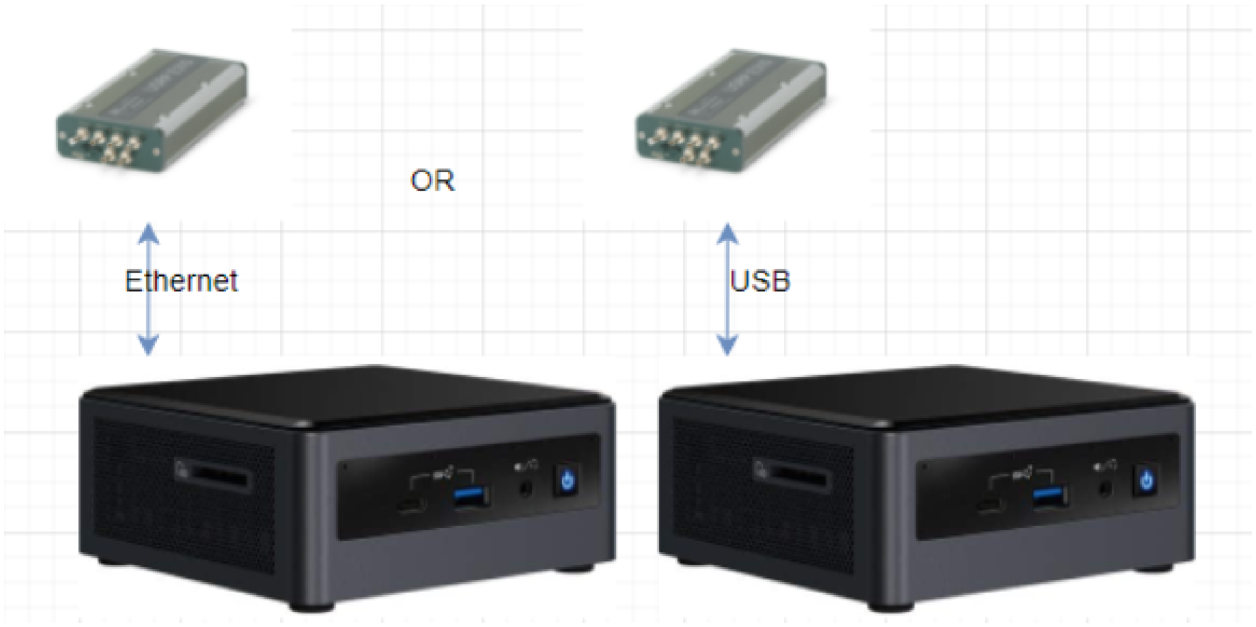
Getting started with the USRP e310 and GNSS-SDR

After going through trials and tribulations over the last 12 weeks, the USRP e310 SDR is finally working with GNSS-SDR. The following steps describe the process to install and get the software running on a fresh installation of ubuntu for an intel Nuc.

Current issues is that the GNSS-sdr software won't keep tracking the GPS satellites with the current configuration file.

Physical connection.

To set up the e310, you must have it connected to the host computer through USB. To communicate with GNSS-sdr you must have a connection over ethernet. In my examples, the host IP will be 10.10.10.3 and the e310 ip will be set to 10.10.10.4.



Example of physical connection

Building and Installing UHD software on the host computer.

Make sure that your ubuntu software is up to date with the following commands. Any software that is out of date will be updated.

```
sudo apt-get update
sudo apt-get upgrade
```

Get dependencies for gnuradio and Gnss-dr

Install the dependencies needed for GNUradio, and GNSS-SDR using the following two commands.

Command 1

```
sudo apt-get -y install autoconf automake build-essential ccache cmake cpufrequtils doxygen
ethtool fort77 g++ gir1.2-gtk-3.0 git gobject-introspection gpsd gpsd-clients inetutils-tools
libasound2-dev libboost-all-dev libcomedi-dev libcppunit-dev libfftw3-bin libfftw3-dev libfftw3-doc
libfontconfig1-dev libgmp-dev libgps-dev libgsl-dev liblog4cpp5-dev libncurses5 libncurses5-dev
libpulse-dev libqt5opengl5-dev libqwt-qt5-dev libsdl1.2-dev libtool libudev-dev libusb-1.0-0
libusb-1.0-0-dev libusb-dev libxi-dev libxrender-dev libzmq3-dev libzmq5 ncurses-bin
python3-cheetah python3-click python3-click-plugins python3-click-threading python3-dev
python3-docutils python3-gi python3-gi-cairo python3-gps python3-lxml python3-mako
python3-numpy python3-numpy-dbg python3-opengl python3-pyqt5 python3-requests
python3-scipy python3-setuptools python3-six python3-sphinx python3-yaml python3-zmq
python3-ruamel.yaml swig wget
```

Command 2

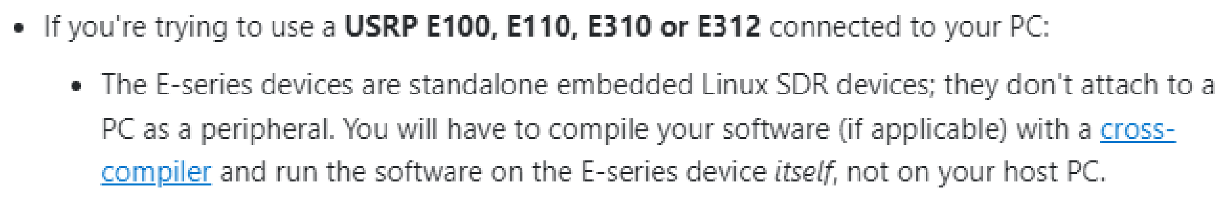
```
sudo apt-get install build-essential cmake git pkg-config libboost-dev \
libboost-date-time-dev libboost-system-dev libboost-filesystem-dev \
libboost-thread-dev libboost-chrono-dev libboost-serialization-dev \
libboost-program-options-dev libboost-test-dev liblog4cpp5-dev \
libblas-dev liblapack-dev \
libarmadillo-dev libgflags-dev libgoogle-glog-dev libhdf5-dev \
libgnutls-openssl-dev libmatio-dev libpugixml-dev libpcap-dev \
libprotobuf-dev protobuf-compiler libgtest-dev googletest \
python3-mako python3-six
```

Acquiring and setting up the UHD repo
We need to use an older version of UHD, so we can use the network streaming mode on the SDR. We will install Version 3.9 to use streaming mode. The steps to download and get to version 3.9 are below.

Commands

```
git clone https://github.com/EttusResearch/uhd
cd uhd/host
git switch UHD-3.9.LTS
mkdir build
cd build
```

To get the software running on a new version you must run the software on the SDR. To get new software on the e310 then you must cross compile from the host pc to load new software. This process is slow and isn't recommended due to computation speed and compile time.



- If you're trying to use a **USRP E100, E110, E310 or E312** connected to your PC:
    - The E-series devices are standalone embedded Linux SDR devices; they don't attach to a PC as a peripheral. You will have to compile your software (if applicable) with a cross-compiler and run the software on the E-series device *itself*, not on your host PC.

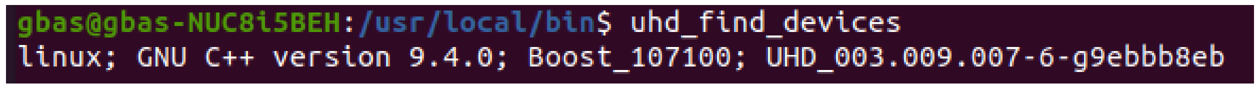
Building and installing UHD on the host PC
The following commands are used to build and compile the UHD software on the host PC. The UHD cmake command has to have the -DENABLE_E300=ON argument or the network mode and e3x0 configurations won't be installed.

```
cmake -DENABLE_E300=ON ../
sake
sake test
sudo make install
sudo ldconfig
```

Note to uninstall: Sudo make uninstall

Verify version
Use the following command to verify the version of the UHD software that you installed on the host computer. This version is 3.9.7.

Uhd_find_devices



```
gbas@gbas-NUC8i5BEH:/usr/local/bin$ uhd_find_devices
linux; GNU C++ version 9.4.0; Boost_107100; UHD_003.009.007-6-g9ebbb8eb
```

There is a slight mismatch between the two uhd versions HOST 3.9.7 e310 = 3.9.2

Flashing and Installing UHD software on the host computer.

Downloading 3.9.2 image

The image that you install on the e310 is going to depend on the serial number of the device that you are loading.

For my example, the e310 that we have uses SG3 instead of SG1. To find the image for the SG3 version can be found on the ETTUS website. The correct image in my example is "sdimage-gnuradio-demo.direct.xz"
https://files.ettus.com/e3xx_images/e3xx-release-4/ettus-e3xx-sg3/

Flashing the SD card

Bmap tools will be used to flash the sd card with the image. Bmap can be installed using the following command.

```
sudo apt-get install -y bmap-tools
```

The image may be installed using the following command once the image is downloaded from the ettus website.

```
Sudo bmaptool copy sdimage-gnuradio-demo.direct.xz /dev/<SD DEVICE NAME> –nobmap
```

This command will take 15-30 minutes to flash the sd card. The sd card will be ready to put back into the e310 once the flashing is over. Once the card is inserted into the e310, you may power on the device.

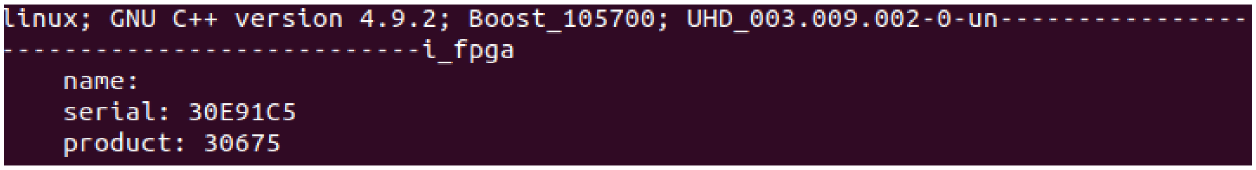
Verify version

You must connect to the e310 through usb to verify the version and configure the network settings. The following command is used to connect to the e310 over USB.

```
sudo screen /dev/ttyUSB0 115200
```

The username for the e310 is **root** by default. The find devices command can be used to verify the version of the UHD image.

```
uhd_find_devices
```



```
linux; GNU C++ version 4.9.2; Boost_105700; UHD_003.009.002-0-un----------------
---------------------------i_fpga
    name:
    serial: 30E91C5
    product: 30675
```

You should see something similar to the above image. You should see that your version is 3.9.2. After verifying that you have the same version, you can proceed with configuring the network settings.

Configuring network settings
On the e310 device, you have to configure your address and netmask. Once the network is configured then you may initiate network mode.

```
ifconfig eth0 10.10.10.4  netmask 255.255.255.0 up
usrp_e3x0_network_mode
```

SSH into sdr
Assuming that you set up the network correctly, you should be able to SSH into the e310 using the IP that you set in the previous step.

```
ssh root@10.10.10.4
```

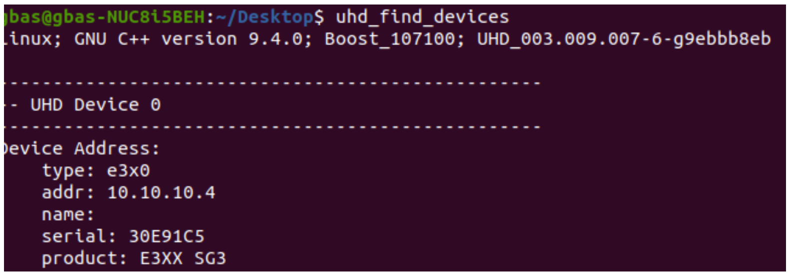
If that doesn’t work then type

```
Ssh-keygen -f “/home/gbas/.ssh/known_hosts” -R “10.10.10.4”
```

Now that you verified that the IP is correct, attempt to find the devices on the host computer using the UHD_FIND_DEVICES and probe commands.

```
uhd_find_devices
```

You should see something like the image below. You will see the device that you configured and setup printed out.

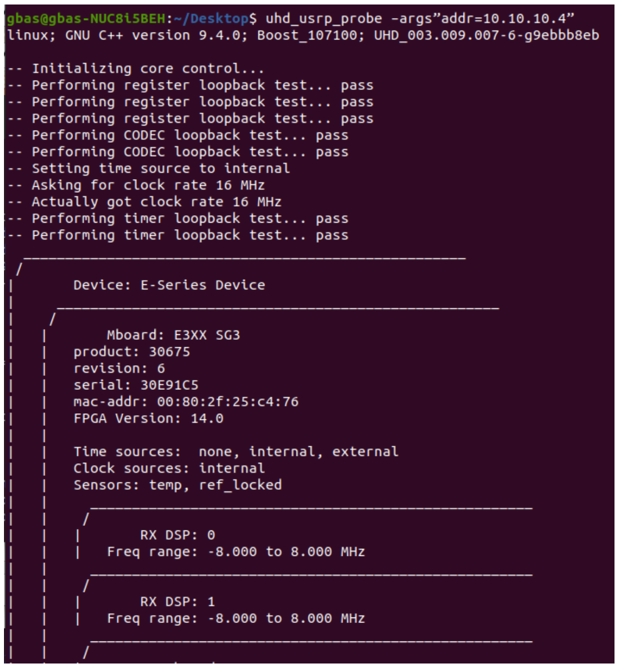


```
bas@gbas-NUC8i5BEH:~/Desktop$ uhd_find_devices
inux; GNU C++ version 9.4.0; Boost_107100; UHD_003.009.007-6-g9ebbb8eb

--------------------------------------------------
- UHD Device 0
--------------------------------------------------
evice Address:
    type: e3x0
    addr: 10.10.10.4
    name:
    serial: 30E91C5
    product: E3XX SG3
```

To verify that everything is going correctly then you type in the following command and it should find the configured e310. An example is shown below.

```
uhd_usrp_probe –args”addr=10.10.10.4”
```



```
gbas@gbas-NUC8i5BEH:~/Desktop$ uhd_usrp_probe -args”addr=10.10.10.4”
linux; GNU C++ version 9.4.0; Boost_107100; UHD_003.009.007-6-g9ebbb8eb

-- Initializing core control...
-- Performing register loopback test... pass
-- Performing register loopback test... pass
-- Performing register loopback test... pass
-- Performing CODEC loopback test... pass
-- Performing CODEC loopback test... pass
-- Setting time source to internal
-- Asking for clock rate 16 MHz
-- Actually got clock rate 16 MHz
-- Performing timer loopback test... pass
-- Performing timer loopback test... pass
  _____________________________________________________
 /
|       Device: E-Series Device
|     _____________________________________________________
|    /
|   |       Mboard: E3XX SG3
|   |   product: 30675
|   |   revision: 6
|   |   serial: 30E91C5
|   |   mac-addr: 00:80:2f:25:c4:76
|   |   FPGA Version: 14.0
|   |
|   |   Time sources:  none, internal, external
|   |   Clock sources: internal
|   |   Sensors: temp, ref_locked
|   |     _____________________________________________________
|   |    /
|   |   |       RX DSP: 0
|   |   |   Freq range: -8.000 to 8.000 MHz
|   |     _____________________________________________________
|   |    /
|   |   |       RX DSP: 1
|   |   |   Freq range: -8.000 to 8.000 MHz
|   |     _____________________________________________________
|   |    /
```

Building and installing GNUradio

The first step will be getting the GNUradio repo. The following commands can be used to set up the repo.

```
cd
git clone https://github.com/gnuradio/gnuradio.git
cd gnuradio
git checkout maint-3.8
git submodule update --init --recursive
mkdir build
cd build
```

INSTALLING

The following steps are used to install the GNUradio software after the initial repository is set up.

```
cmake -DCMAKE_BUILD_TYPE=Release -DPYTHON_EXECUTABLE=/usr/bin/python3 ../
make -j4
sudo make install
sudo ldconfig
```

GNUradio is now installed and you may move on to installing the GNSS-sdr software.

Building and installing GNSS-SDR

Install the following prerequisite library.

```
sudo apt-get install -y libfmt-dev
```

Set up and install the gnss-sdr library using the following commands.

```
git clone https://github.com/gnss-sdr/gnss-sdr
cd gnss-sdr/build
git checkout next
cmake ..
make
sudo make install
Sudo ldconfig
```

make sure to run volk_profile and volk_gnsssdr_profile commands after installation to optimize the gnss-sdr software for your machine.

Running the GNSS-SDR software

The gnss-sdr software now may be ran through the following command.

```
gnss-sdr --config_file=/home/gbas/work/usrp_GPS_e310.conf
```

Where the config file is the file that is configured for your device. An example config file is liste i the appendix.

GUI - GNSS monitor

A graphical interface may be installed to view the GNSS data. You may use the following commands to install and run the monitor.

```
git clone https://github.com/acebrianjuan/gnss-sdr-monitor
cd gnss-sdr-monitor/build
cmake ..
```

/home/gbas/work/gnss-sdr-monitor/build/src/gnss-sdr-monitor

If you have issues with it not working or not showing on the map, consider using ctrl+C to kill the program from the terminal that started it. Then open the software *after* the gnss-sdr has started showing solutions (green messages). If these don't work, double check the port numbers to ensure it is going to the right place.

Conclusion

Everything is mostly working, but the monitor and gnss-sdr software isn't locking onto the GPS satellites. I'm not exactly sure if the issue is with the antenna, a configuration setting, or the hardware itself.

Links

https://files.ettus.com/e3xx_images/README
https://wiki.gnuradio.org/index.php/Copy_an_image_file_to_the_SD_card
https://kb.ettus.com/Writing_the_USRP_File_System_Disk_Image_to_a_SD_Card
https://kb.ettus.com/Software_Development_on_the_E310_and_E312
https://gitlab.com/librespacefoundation/sdrmakerspace/sdr-pr/-/wikis/Partial-reconfiguration-on-the-E310-Board/Testing-a-simple-design-with-UHD?version_id=b0e8d2156949f8669ff55b31311c28475c863db5
https://gitlab.com/librespacefoundation/sdrmakerspace/sdr-pr/-/wikis/Partial-reconfiguration-on-the-E310-Board/Tools-installation-RFNoC,-UHD,-GNU-Radio
`https://files.ettus.com/manual/page_usrp_e3x0.html#e3x0_upgrade_sd_card`
https://github.com/acebrianjuan/gnss-sdr-monitor
Setup remote work.
Connected to e310, not working anymore
Following this guide.
https://kb.ettus.com/E310/E312_Getting_Started_Guides

verified that the software is installed on the nuc.
https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux
https://wiki.gnuradio.org/index.php/InstallingGR
https://gnss-sdr.org/build-and-install/

Extra installation steps for future use.

For newer versions of UHD Building and Installing UHD image on the e310.

sudo uhd_images_downloader

If that doesn’t work then navigate to usr/local/bin and run

    Sudo python3 uhd_images_downloader

CONFIG file

```
[GNSS-SDR]

;######### GLOBAL OPTIONS ##################
GNSS-SDR.internal_fs_sps=1000000

;######### SIGNAL_SOURCE CONFIG ############
SignalSource.implementation=UHD_Signal_Source
SignalSource.device_address=10.10.10.4
SignalSource.item_type=cshort
SignalSource.sampling_frequency=1000000
SignalSource.freq=1575420000
SignalSource.gain=40
SignalSource.subdevice=A:A  ; <- Can be A:0 or B:0
SignalSource.samples=0
SignalSource.clock_source=internal

;######### SIGNAL_CONDITIONER CONFIG ############
SignalConditioner.implementation=Signal_Conditioner

;######### DATA_TYPE_ADAPTER CONFIG ############
DataTypeAdapter.implementation=Pass_Through
DataTypeAdapter.item_type=cshort

;######### INPUT_FILTER CONFIG ############
InputFilter.implementation=Fir_Filter
InputFilter.input_item_type=cshort
InputFilter.output_item_type=gr_complex
InputFilter.taps_item_type=float
InputFilter.number_of_taps=11
InputFilter.number_of_bands=2

InputFilter.band1_begin=0.0
InputFilter.band1_end=0.48
InputFilter.band2_begin=0.52
InputFilter.band2_end=1.0

InputFilter.ampl1_begin=1.0
InputFilter.ampl1_end=1.0
InputFilter.ampl2_begin=0.0
InputFilter.ampl2_end=0.0

InputFilter.band1_error=1.0
InputFilter.band2_error=1.0
InputFilter.filter_type=bandpass
InputFilter.grid_density=16
InputFilter.sampling_frequency=1000000
InputFilter.IF=0

;######### RESAMPLER CONFIG ############
Resampler.implementation=Pass_Through

;######### CHANNELS GLOBAL CONFIG ############
Channels_1C.count=8
Channels.in_acquisition=1

;######### ACQUISITION GLOBAL CONFIG ############
Acquisition_1C.implementation=GPS_L1_CA_PCPS_Acquisition
Acquisition_1C.item_type=gr_complex
Acquisition_1C.coherent_integration_time_ms=1
Acquisition_1C.pfa=0.01
Acquisition_1C.doppler_max=5000
Acquisition_1C.doppler_step=250
Acquisition_1C.max_dwells=1
Acquisition_1C.dump=false
Acquisition_1C.dump_filename=./acq_dump.dat

;######### TRACKING GLOBAL CONFIG ############
Tracking_1C.implementation=GPS_L1_CA_DLL_PLL_Tracking
Tracking_1C.item_type=gr_complex
Tracking_1C.extend_correlation_symbols=10
Tracking_1C.early_late_space_chips=0.5
Tracking_1C.early_late_space_narrow_chips=0.15
Tracking_1C.pll_bw_hz=40
Tracking_1C.dll_bw_hz=2.0
Tracking_1C.pll_bw_narrow_hz=5.0
Tracking_1C.dll_bw_narrow_hz=1.50
Tracking_1C.fll_bw_hz=10
Tracking_1C.enable_fll_pull_in=true
Tracking_1C.enable_fll_steady_state=false
Tracking_1C.dump=false
Tracking_1C.dump_filename=tracking_ch_

;######### TELEMETRY DECODER GPS CONFIG ############
TelemetryDecoder_1C.implementation=GPS_L1_CA_Telemetry_Decoder

;######### OBSERVABLES CONFIG ############
```

```
Observables.implementation=Hybrid_Observables

;######### PVT CONFIG ############
PVT.implementation=RTKLIB_PVT
PVT.positioning_mode=Single
PVT.output_rate_ms=100
PVT.display_rate_ms=500
PVT.iono_model=Broadcast
PVT.trop_model=Saastamoinen
PVT.flag_rtcm_server=true
PVT.flag_rtcm_tty_port=false
PVT.rtcm_dump_devname=/dev/pts/1
PVT.rtcm_tcp_port=2101
PVT.rtcm_MT1019_rate_ms=5000
PVT.rtcm_MT1077_rate_ms=1000
PVT.rinex_version=2
```