The entire installation guide for getting RTL SDR Drivers, GNU Radio, and Ettus custom blocks working on Ubuntu.

Commands with the command prefix "sudo apt-get …" installs a widely used library on the Ubuntu filesystem. These commands can be done on an Ubuntu OS installed to any machine (PC or Raspberry Pi). The machine used will determine the file system space (a memory space greater than 16 Megabytes is adequate for a correct installation of all the necessary libraries/modules in this tutorial, and a 16/32 Megabyte SD card is often used among Raspberry Pi users).

The purpose of this installation is to get the operating system to recognize RTL SDRs, acquire GNU Radio for DSP, and get Ettus custom blocks working in GNU Radio. The end goal of these steps will give the user the complete software framework for getting a functional AoA estimator running on a Raspberry Pi or PC (to the level we have achieved).

Important Note: Steps 2 and onwards use the terminology "generic cmake instructions" to refer to this set of instructions to install github libraries into Ubuntu:

$ git clone "*github library here"*
$ cd "*installed library name"*
$ mkdir build
$ cd build
$ cmake ..
$ make
$ make test
$ sudo make install
$ sudo ldconfig

It is critical to know that **Ubuntu versions 14.04 and 16.04** are the only compatible distributions with the Ettus Doa library at this time.

0. Make Sure to update the operating system before the following installations:
   pi@raspberrypi ~ $ sudo apt-get update
   pi@raspberrypi ~ $ sudo apt-get upgrade

1. Install GNU Radio by using the command `sudo apt-get install gnuradio-dev`

   Next, install the RTL SDR drivers, the following commands have been included in this report in case this page (https://gist.github.com/floehopper/99a0c8931f9d779b0998) gets lost in the future:

```
pi@raspberrypi ~ $ cat <<EOF >no-rtl.conf
blacklist dvb_usb_rtl28xxu
blacklist rtl2832
blacklist rtl2830
EOF
pi@raspberrypi ~ $ sudo mv no-rtl.conf /etc/modprobe.d/

pi@raspberrypi ~ $ sudo apt-get install git-core
pi@raspberrypi ~ $ sudo apt-get install git
pi@raspberrypi ~ $ sudo apt-get install cmake
pi@raspberrypi ~ $ sudo apt-get install libusb-1.0-0-dev
pi@raspberrypi ~ $ sudo apt-get install build-essential

pi@raspberrypi ~ $ git clone git://git.osmocom.org/rtl-sdr.git
pi@raspberrypi ~ $ cd rtl-sdr/
pi@raspberrypi ~/rtl-sdr $ mkdir build
pi@raspberrypi ~/rtl-sdr $ cd build
pi@raspberrypi ~/rtl-sdr/build $ cmake ../ -DINSTALL_UDEV_RULES=ON
pi@raspberrypi ~/rtl-sdr/build $ make
pi@raspberrypi ~/rtl-sdr/build $ sudo make install
pi@raspberrypi ~/rtl-sdr/build $ sudo ldconfig
pi@raspberrypi ~/rtl-sdr/build $ cd ~
pi@raspberrypi ~ $ sudo cp ./rtl-sdr/rtl-sdr.rules /etc/udev/rules.d/
pi@raspberrypi ~ $ sudo reboot
```

After sending the above commands, you should be able to perform a simple test which generates a list of RTL SDRs recognized by the operating system. The test also checks the possible tuning range of each RTL SDR and the maximum sampling rate possible on the device (computer or Raspberry Pi) being used.

```
pi@raspberrypi ~ $ rtl_test
Found 1 device(s):
  0:  Generic, RTL2832U, SN: 77771111153705700
```

2. Install the bias_tee software from https://github.com/rtlsdrblog/rtl_biast (this will be used to turn the bias_tee on/off when performing RF Switching (not yet completed)) using the generic cmake instructions.

3. Install armadillo-code from https://github.com/conradsnicta/armadillo-code/ using the generic cmake instructions.

4. Install EttusResearch gr-doa custom blocks from https://github.com/EttusResearch/gr-doa using the generic cmake instructions. On some Ubuntu distributions, a list of missing

dependencies is generated during the execution of the cmake instruction. If a 16.04 Ubuntu version was installed, and steps 0-2 were completed correctly, no additional dependencies need to be installed besides armadillo-code (step 3).

5. Relevant GRC files can be found from https://github.com/jakapoor/AMRUPT. The only file that needs to be installed from the Github at this time is the master flowchart for obtaining rudimentary AoA measurements called musicRTL.grc. Download this file and open it with GNU Radio Companion. GNU Radio Companion should show a complete flowchart resembling Figure 8 from the Design Implementation section. If steps 0-4 were followed correctly, no messages exclaiming “missing block: ‘block name’ not found” should be displayed.