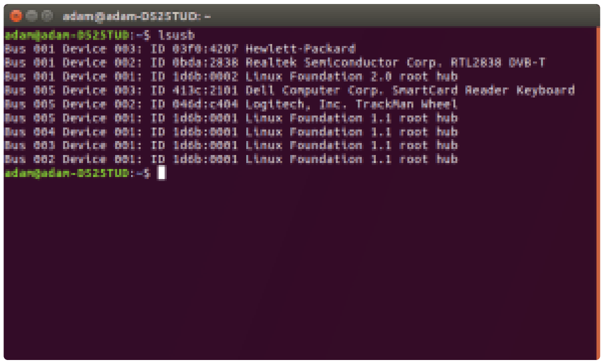
Using NooElec NESDR SMArt on Ubuntu 17.04

NOOELEC NESDR SMART

In this HowTo, I will show you how to get started using a NooElec NESDR SMArt on a computer running Ubuntu 17.04.

The first thing to do after you have your NESDR SMArt plugged into your Ubuntu computer is to run the command `lsusb`.[1)] This will list all of the USB devices attached to the computer. If things are working as they should be, you will see the NESDR SMArt listed as: `Realtek Semiconductor Corp. RTL2838 DVB-T`. The DVB-T is the telling part. It means that the operating system has recognized the device and loaded, what it believes to be, the correct driver, since the default use of the device is to receive television broadcasts.

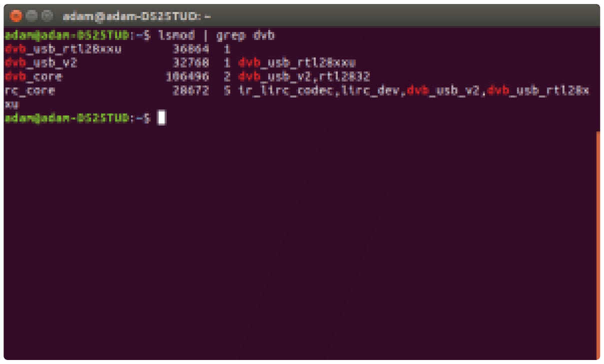
Click on images to see a larger version



```
adam@adam-DS25TUD:~$ lsusb
Bus 001 Device 003: ID 03f0:4207 Hewlett-Packard
Bus 001 Device 002: ID 0bda:2838 Realtek Semiconductor Corp. RTL2838 DVB-T
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 003: ID 413c:2101 Dell Computer Corp. SmartCard Reader Keyboard
Bus 005 Device 002: ID 046d:c404 Logitech, Inc. TrackMan Wheel
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
adam@adam-DS25TUD:~$
```

LSUSB SHOWS THAT THE DONGLE IS SEEN

This can be further seen by using the command: `lsmod | grep dvb`, which lists the loaded modules (drivers) and filters them to just show the ones that have the letters dvb in them. You will see they are loaded.

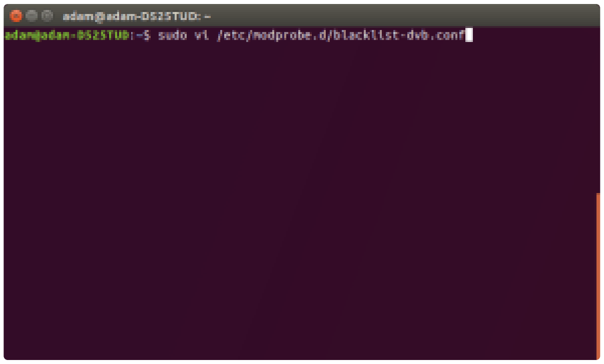


```
adam@adam-DS25TUD:~$ lsmod | grep dvb
dvb_usb_rtl28xxu       36864  1
dvb_usb_v2             32768  1 dvb_usb_rtl28xxu
dvb_core              106496  2 dvb_usb_v2,rtl2832
rc_core                28672  5 ir_lirc_codec,lirc_dev,dvb_usb_v2,dvb_usb_rtl28x
xu
adam@adam-DS25TUD:~$
```

LSMOD SHOWS THAT THE DVB DRIVER IS INSTALLED AND LOADED

What we need to do now is remove those modules (drivers) and load the ones dedicated to using the device as an SDR.

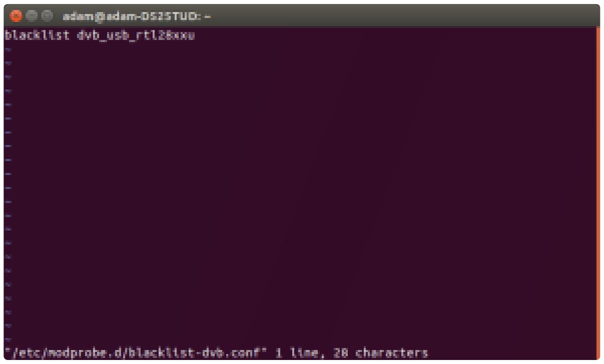
We start by "blacklisting" the default drivers. This is done by editing the file
2) `/etc/modprobe.d/blacklist-dvb.conf`.



```
adam@adam-DS25TUD:~$ sudo vi /etc/modprobe.d/blacklist-dvb.conf
```

EDIT THE FILE

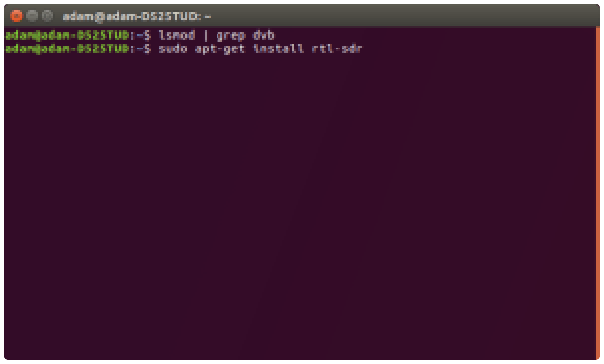
Add the following to the file: `blacklist dvb_usb_rtl28xxu`, save, and close it.



```
blacklist dvb_usb_rtl28xxu
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/modprobe.d/blacklist-dvb.conf" 1 line, 28 characters
```

CONTENTS OF FILE

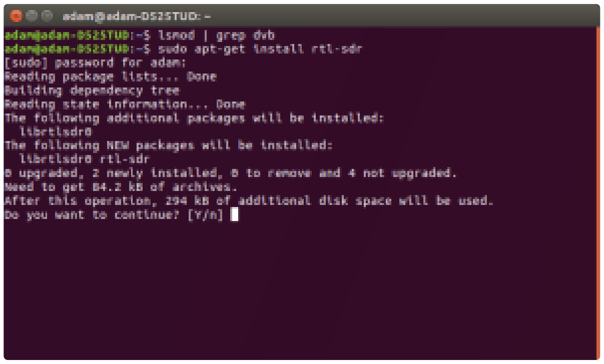
What that does is disallow the default module (driver) to load. Now we need to tell it to load the driver we want. But first, we need to download it onto the computer.



```
adam@adam-DS25TUD:~$ lsmod | grep dvb
adam@adam-DS25TUD:~$ sudo apt-get install rtl-sdr
```

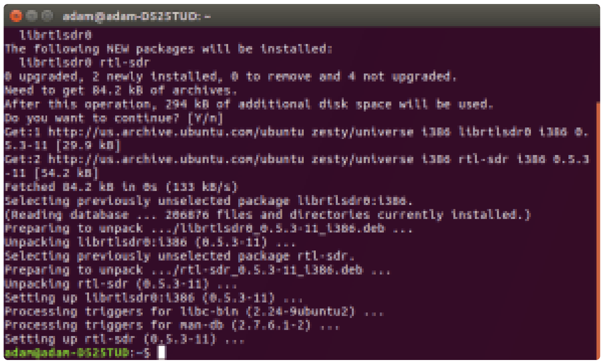
NOW INSTALL THE RTL-SDR DRIVER AND RELATED SOFTWARE

`sudo apt-get install rtl-sdr` will install the package we need. This will have the drivers and utilities related to using a SDR.



```
adam@adam-DS25TUD:~$ lsmod | grep dvb
adam@adam-DS25TUD:~$ sudo apt-get install rtl-sdr
[sudo] password for adam:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  librtlsdr0
The following NEW packages will be installed:
  librtlsdr0 rtl-sdr
0 upgraded, 2 newly installed, 0 to remove and 4 not upgraded.
Need to get 84.2 kB of archives.
After this operation, 294 kB of additional disk space will be used.
Do you want to continue? [Y/n] █
```

IT LOOKS LIKE WE HAVE WHAT WE WANT.

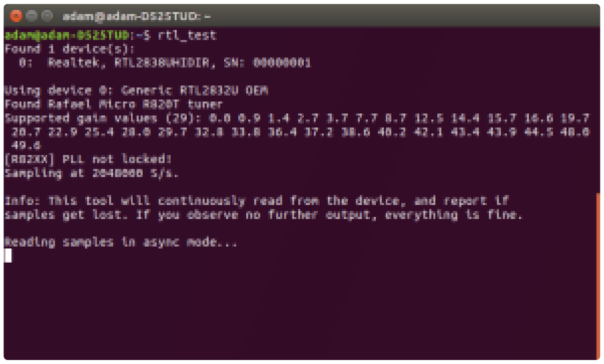


```
adam@adam-DS2STUD: ~
  librtlsdr0
The following NEW packages will be installed:
  librtlsdr0 rtl-sdr
0 upgraded, 2 newly installed, 0 to remove and 4 not upgraded.
Need to get 84.2 kB of archives.
After this operation, 294 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://us.archive.ubuntu.com/ubuntu zesty/universe i386 librtlsdr0 i386 0.
5.3-11 [29.9 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu zesty/universe i386 rtl-sdr i386 0.5.3
-11 [54.2 kB]
Fetched 84.2 kB in 0s (133 kB/s)
Selecting previously unselected package librtlsdr0:i386.
(Reading database ... 206876 files and directories currently installed.)
Preparing to unpack .../librtlsdr0_0.5.3-11_i386.deb ...
Unpacking librtlsdr0:i386 (0.5.3-11) ...
Selecting previously unselected package rtl-sdr.
Preparing to unpack .../rtl-sdr_0.5.3-11_i386.deb ...
Unpacking rtl-sdr (0.5.3-11) ...
Setting up librtlsdr0:i386 (0.5.3-11) ...
Processing triggers for libc-bin (2.24-9ubuntu2) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up rtl-sdr (0.5.3-11) ...
adam@adam-DS2STUD:~$ 
```

IT IS NOW INSTALLED.

If you don't have an antenna hooked up yet, now is the time to do so.

Now come the real test. Let's see if the SDR works at the lowest level we care about; which is tuning on a particular frequency and seeing if we hear anything. We do this by executing `rtl_test`.



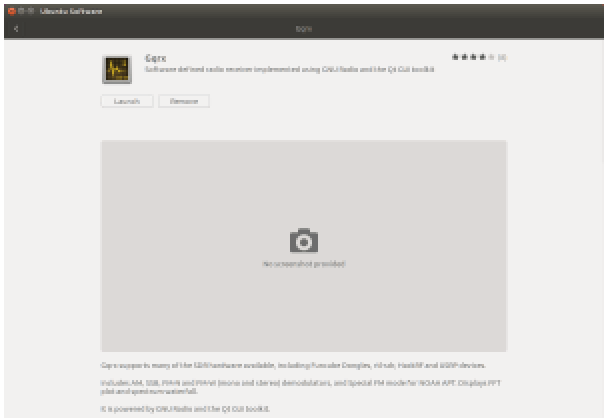
```
adam@adam-DS2STUD: ~
adam@adam-DS2STUD:~$ rtl_test
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7
 20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0
 49.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
```

RESULTS OF RTL-TEST. YOU WANT IT TO STOP HERE AND NOT PRINT ANY MORE LINES.

The first part of the test will test the parameters of the SDR and then it will attempt to tune in to a frequency. It is at this point you want the test to stop printing to the screen; because anything more is an error that occurred.

At this point you have "the guts" installed. You have the modules and libraries loaded that are needed to run the dongle as a software defined radio. The next step is to put a pretty frontend on this to make it easier to use, as well as add functionality — like a waterfall display.

With Ubuntu, the easiest way to do that is with Gqrx. It is available in the Ubuntu Software store, and therefore, easy to install.



Gqrx

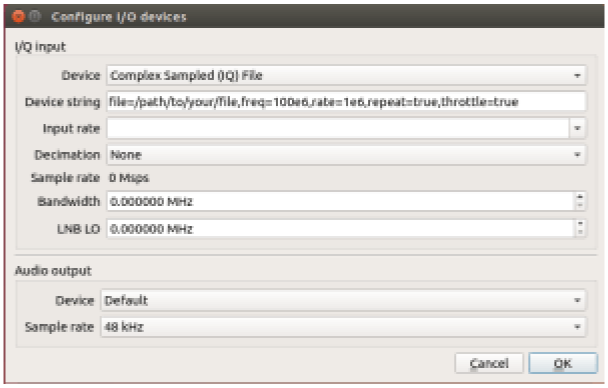
Software defined radio receiver implemented using GNU Radio and the Qt GUI toolkit

Gqrx supports many of the SDR hardware available, including Funcube Dongles, rtl-sdr, HackRF and USRP devices.

Includes AM, SSB, FM-N and FM-W (mono and stereo) demodulators, and Special FM mode for NOAA APT. Displays FFT plot and spectrum waterfall.

It is powered by GNU Radio and the Qt GUI toolkit.

With Gqrx installed, start it up and it is going to ask you what you want to use as the source.



I/Q input

Device: Complex Sampled (IQ) File

Device string: file=/path/to/your/file,freq=100e6,rate=1e6,repeat=true,throttle=true

Input rate:

Decimation: None

Sample rate: 0 Msps

Bandwidth: 0.000000 MHz

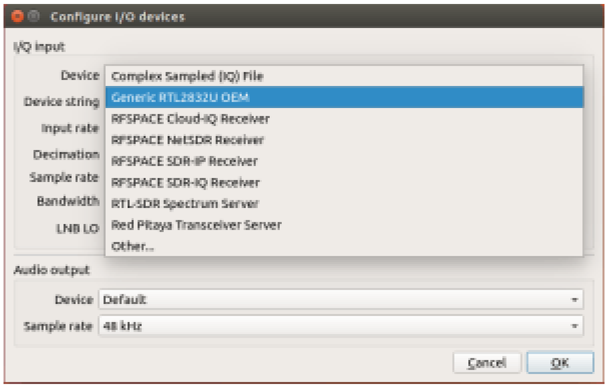
LNB LO: 0.000000 MHz

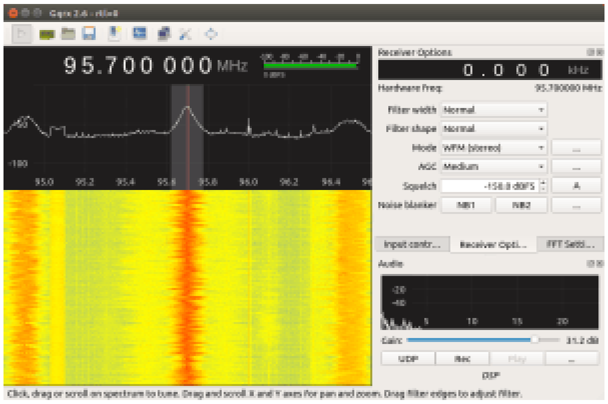
Audio output

Device: Default

Sample rate: 48 kHz

You want to change the Device to your dongle.





Gqrx 2.6 - rtl=0

95.700 000 MHz

Receiver Options

0 . 0 0 0 kHz

Hardware Freq: 95.700000 MHz

Filter width: Normal

Filter shape: Normal

Mode: WFM (stereo)

AGC: Medium

Squelch: -150.0 dBFS

Gain: 31.2 dB

Click, drag or scroll on spectrum to tune. Drag and scroll X and Y axes for pan and zoom. Drag filter edges to adjust filter.

For more information on Gqrx SDR, I suggest starting with this article on the Gqrx SDR website.

References

1. ↑ 5 Ways To Open A Terminal Console Window Using Ubuntu.
2. ↑ A Beginners Guide To The Nano Editor

Version 1.0.0