

# RTL-SDR for Linux Quick Start Guide



by Kenn Ranous

# **RTL-SDR for Linux Quick Start Guide**

**v10.16 by Kenn Ranous**

Page 3	Introduction
Page 4	Installing Drivers
Page 6	Gqrx - General Purpose Radio & Spectrum Software
Page 8	Cubic SDR - General Purpose Radio & Spectrum Software
Page 9	ADS-B Aircraft Tracking with Dump1090
Page 12	Advanced Aircraft Tracking with Virtual Radar Server
Page 16	Spectrum Analysis with Qspectrum Analyzer
Page 17	Monitoring Trunked Radio Systems with SDR Trunk
Page 19	Other interesting RTL-SDR applications
Page 19	Additional Resources
Page 19	In Closing ...

## **Introduction:**

How to get an RTL-SDR dongle working on a Linux PC made (relatively) quick and easy. These instructions were created for Linux Mint and should work in any Debian or Ubuntu based distribution. I'll cover the software that I've found to be complete and working. Note that most things RTL-SDR are a work in progress, including this guide. Calibrate your expectations to that of beta software. Updates sometimes break things and documentation isn't always accurate. I'll be updating this guide using the convention vMonth.Year in an attempt to keep up with what works.

Preferred dongles have the 820T2 tuner, a TXCO and are in a shielded enclosure with heat-sink. Use a good quality USB extension cable with ferrite cores on both ends. Try to position the dongle away from the computer and other sources of electrical noise to minimize interference.

PC system requirements are modest. Decoding and GUI displays need some processing power so opt for a dual-core or better CPU. None of the applications here are highly demanding of Video or HDD. At least 2GB of RAM is best.

Regardless of cost, any radio is only as good as it's antenna. For now I'll just touch on antennas as there's an abundance of information on-line and in ARRL publications.

- A set of TV rabbit ears is a good way to get started.
- Discone antennas are broadband and work well for general purpose use.
- Outdoor TV antennas work quite well for scanning.
- Turnstiles and Folded Dipoles excel at FM broadcast reception.
- For ADS-B a 12 element Coaxial Collinear is notably superior to a discone.
- Dipoles and long wires work well for the HF bands.
- Ground planes, J-poles, dipoles, yagis, quads and log periodic antennas can offer superior performance when designed for a specific band.

In summary, VHF/UHF frequencies tend to be line-of-sight and can be blocked by some building materials, especially metal. Higher is better and outdoors is best.

**Watch out for those power lines!**

## **Installing the drivers:**

1. Open a terminal and confirm you're in your home directory. This is the preferred convention and will be assumed throughout this text. Text highlighted in gray is exactly what you want to type at the command line. Yes it's case sensitive.

2. Update your distribution.

```
sudo apt-get update
```

3. Install the tools needed to retrieve (git), compile (cmake) and build (build-essential).

```
sudo apt-get install git
```

```
sudo apt-get install cmake
```

```
sudo apt-get install build-essential
```

4. Install libusb-1.0-0-dev which is a C library that provides generic access to USB devices.

```
sudo apt-get install libusb-1.0-0-dev
```

5. Retrieve, build and compile the RTL2832U Osmocom drivers from the source.

```
git clone git://git.osmocom.org/rtl-sdr.git
```

```
cd rtl-sdr/
```

```
mkdir build
```

```
cd build
```

```
cmake ../ -DINSTALL_UDEV_RULES=ON
```

```
make
```

```
sudo make install
```

```
sudo ldconfig
```

```
sudo cp ../rtl-sdr.rules /etc/udev/rules.d/
```

6. Blacklist the default driver that is automatically loaded for using the dongle as a TV device as it doesn't work for SDR purposes and clashes with the new Osmocom drivers we just installed.

A. Open your /etc/modprobe.d folder as an administrator.

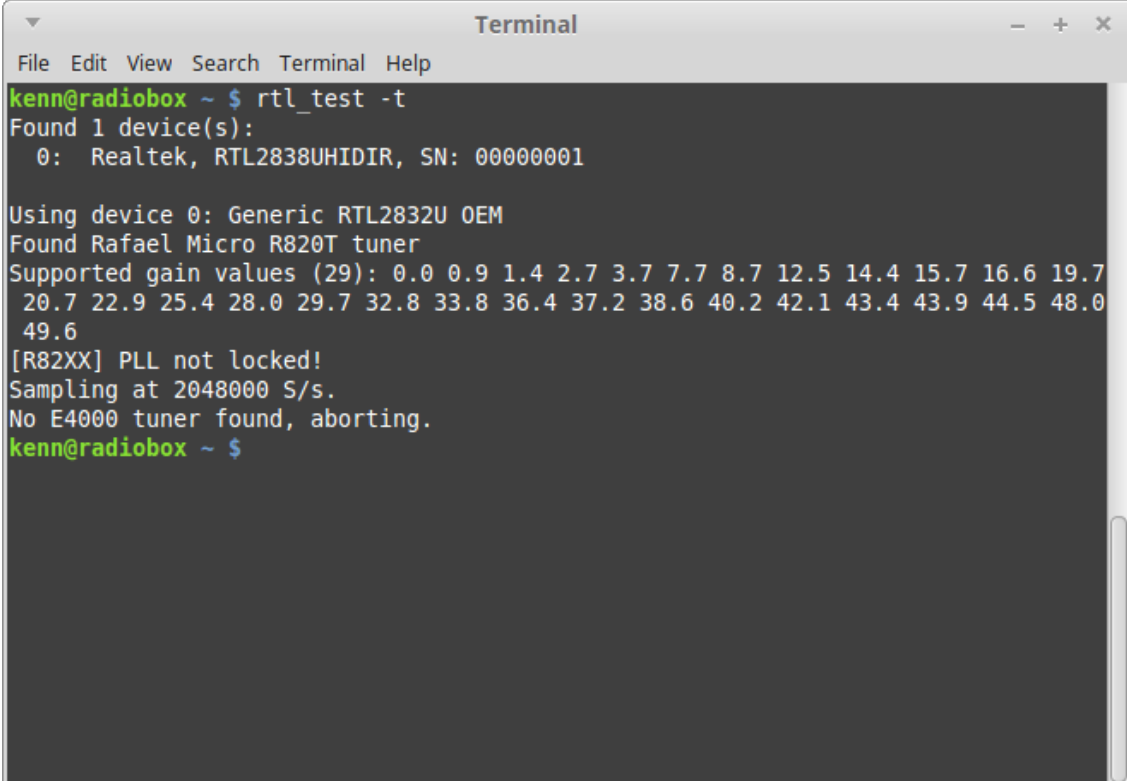
B. Create a new file 'blacklist-rtl.conf' and add this one line:

```
blacklist dvb_usb_rtl28xxu
```

C. Save the file, close the editor and restart the machine.

7. Test that the dongle is working by opening a terminal and typing:

```
rtl_test -t
```



```
Terminal
File Edit View Search Terminal Help
kenn@radiobox ~ $ rtl_test -t
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.
No E4000 tuner found, aborting.
kenn@radiobox ~ $
```

Don't worry about messages like PLL not locked or E4000 tuner not found or that it says R820T and not R820T2. If you're seeing the above then your drivers and dongle are working. You're now ready to install some applications.

## **Gqrx General Purpose Radio/Spectrum Analyzer:**

This can be found in the Synaptic Package Manager. To get the latest version, visit <http://gqrx.dk> for instructions on how to add their PPA's (Personal Package Archive) to your system and install. They've made it pretty easy.

On first startup you'll see the Configure I/O devices box. Default values are usually correct, just make sure your RTL dongle is selected then hit OK.

Click the Power Button in the upper left corner to start receiving. Input the desired frequency and under Mode select the appropriate demodulator. (ie Narrow FM for NOAA Weather Radio).

Hover over each item to see what it does. Toggle hardware AGC and adjust LNA gain to maximize the signal but not so high that static or distortion takes over. Note that you'll need to alter gain settings as you tune around the spectrum. I find setting gain and AGC so the noise floor is around -60db to -70db works well.

You can also bookmark favorite frequencies and organize them by category.

If the audio sounds choppy, try reducing the sample rate. Also check under the FFT Settings tab and reduce the FFT size and Rate fps. You may also need to adjust your PPM as described next.

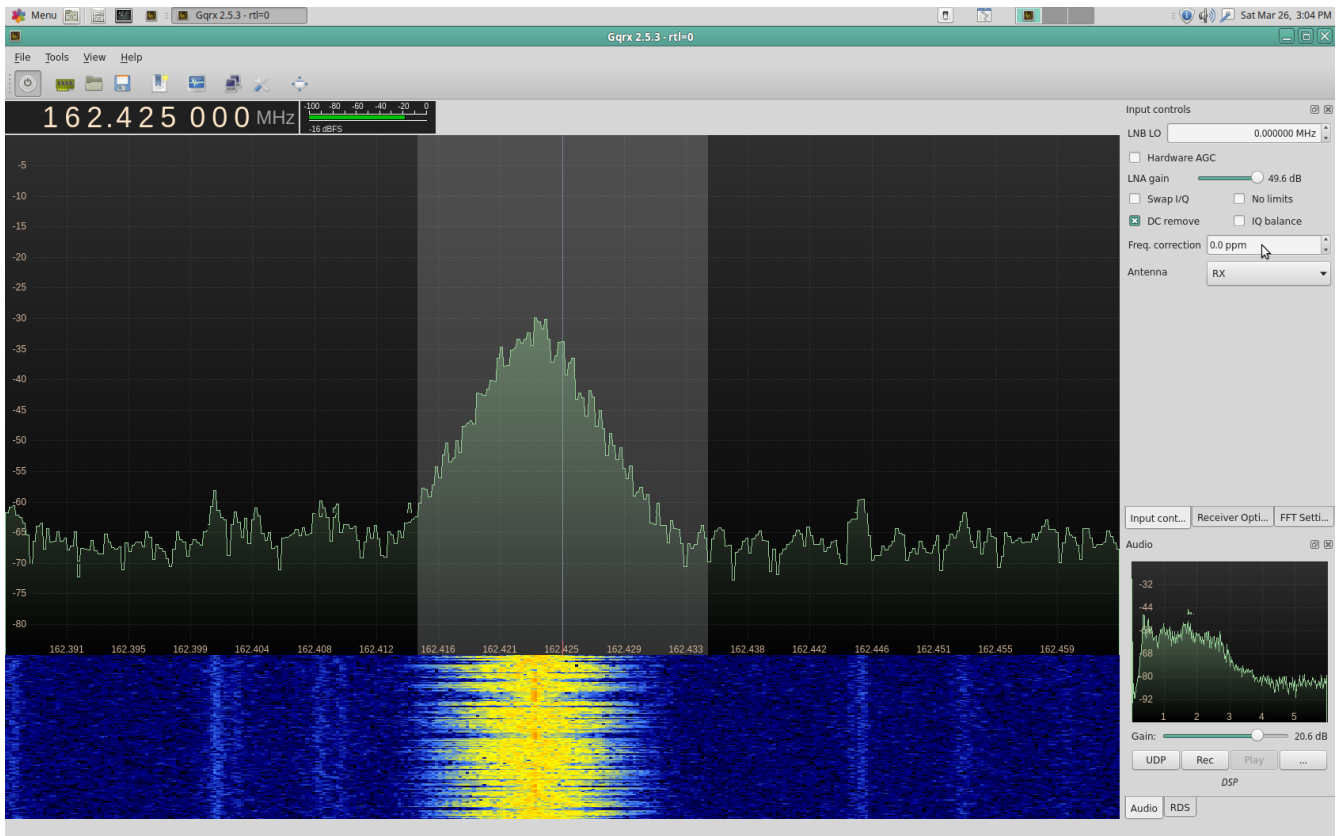
### **Frequency Correction with GQRX:**

The RTL-SDR dongles are mass-produced devices designed for TV where precise frequency accuracy is not important. For use with narrow-band signals we need a greater degree of accuracy. Later model dongles have a more accurate TXCO but older ones may be off by enough to compromise reception.

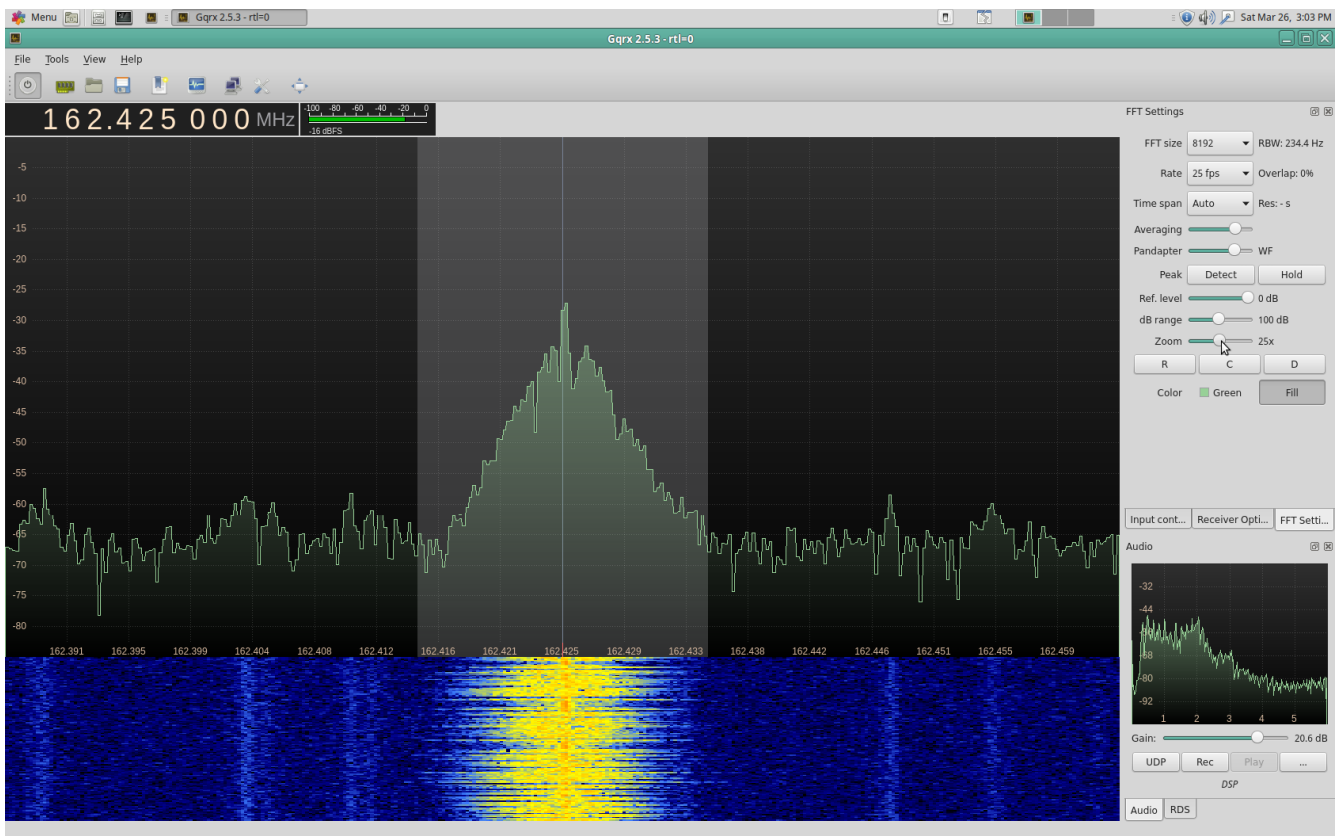
Fortunately, there's an easy fix. Most applications have provisions to adjust the PPM. Once you've determined your PPM error correction factor you can use that value in most software. As long as the dongle doesn't overheat, this value should remain reasonably constant. To determine your dongles PPM, launch GQRX and tune your SDR to one of the National Weather Service Radio frequencies as listed.

162.400	162.425	162.450	162.475	162.500	162.525	162.550
---------	---------	---------	---------	---------	---------	---------

Zoom in on the signal and adjust your PPM until the tuning line is centered on the signal. See the next two images for details.



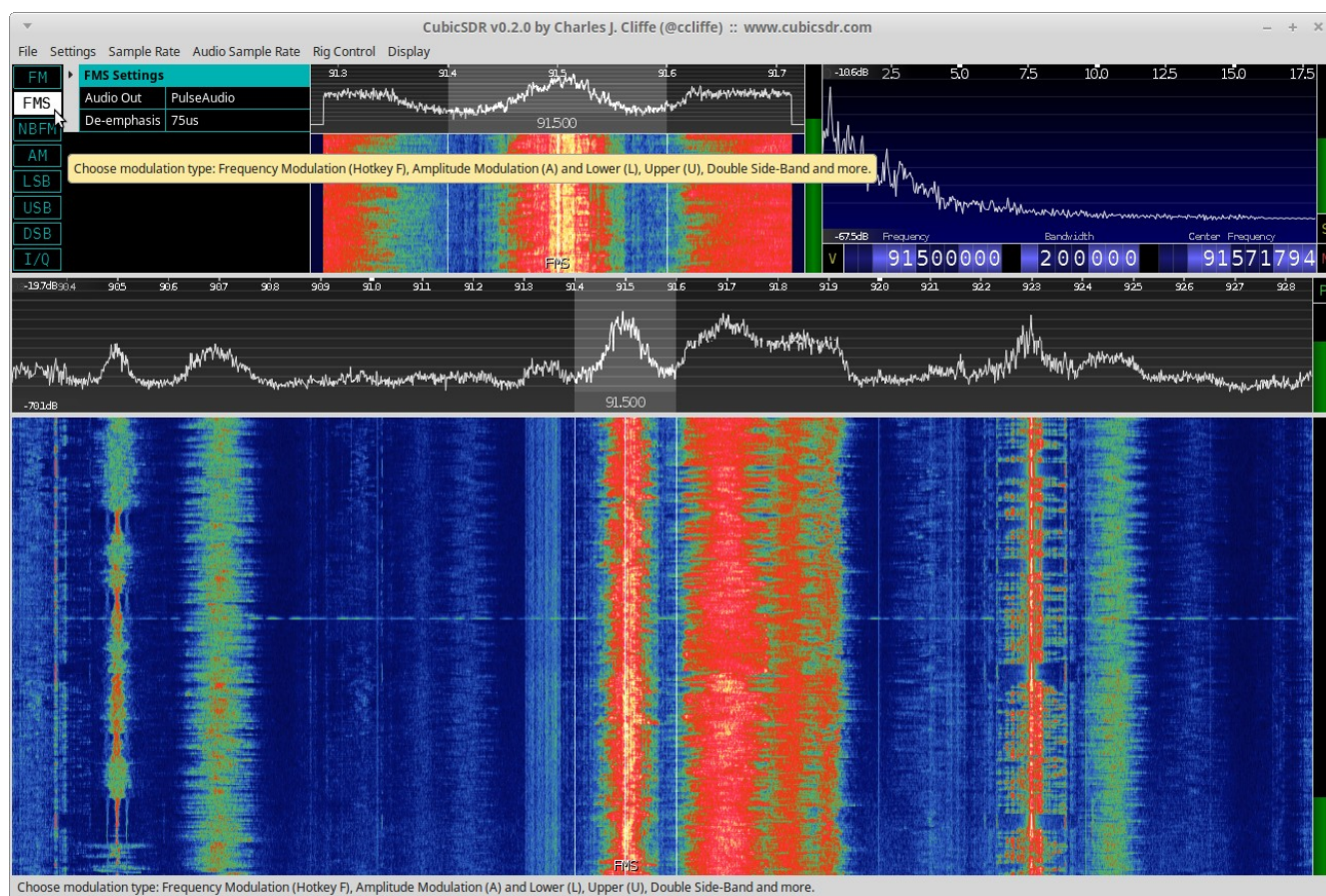
*Uncorrected PPM error.*



*Adjust your PPM + or - until tuning line is centered on signal.*

## **CubicSDR General Purpose Radio/Spectrum Analyzer:**

A bit more visually oriented than Gqrx, I find that CubicSDR delivers smoother audio. The latest v0.2.0 is available for download from <http://cubicsdr.com> where there's instructions on how to build it, get it via flatpak or you can just download the .Applmage file. Locate it in your Downloads folder and Right-click > Properties > Permissions and tick 'Allow executing file as a program' then double-click to run. (yes, it's that simple, just one 13 meg file) On startup, select 'Generic RTL2832U' and click 'Start'.



Hover over each section for info, it's pretty self explanatory. Enter a frequency directly or click on one of the streams in the lower waterfall panel to start listening. If the audio sounds choppy, click 'Input Bandwidth' on the top menu and select a lower number. The .Applmage file can be placed anywhere you like and sessions can be saved as .xml files.



## **ADS-B Aircraft Tracking with Dump1090**

Dump1090 is a good ADS-B decoder. Originally created by antirez, several developers (malcolrobb, mutability and tedsluis) have modified the original code to make improvements. Each one is called fork. Install as many as you like, just make sure each is in it's own unique folder.

I'll focus on the Ted Sluis fork as it seems to detect more aircraft and provides more information. It also has interesting features such as range & altitude color codes and ten map & chart styles.

<http://github.com/tedsluis>

Open a terminal and type:

```
git clone git://github.com/tedsluis/dump1090.git dump1090
cd dump1090
make
./dump1090 --interactive --net --enable-agc
```

Within a few moments you should see a list of aircraft in the terminal window. Open a web browser and in the address bar enter localhost:8080 and you should see a map with aircraft. You might need to hit the browser refresh button once or twice to get it going.

To launch dump1090 without needing to remember paths and optimal flags, create a text file named d1090 in your /home/username/bin folder. (Note there's no extension) Enter these lines, substituting your own lat, lon and PPM values.

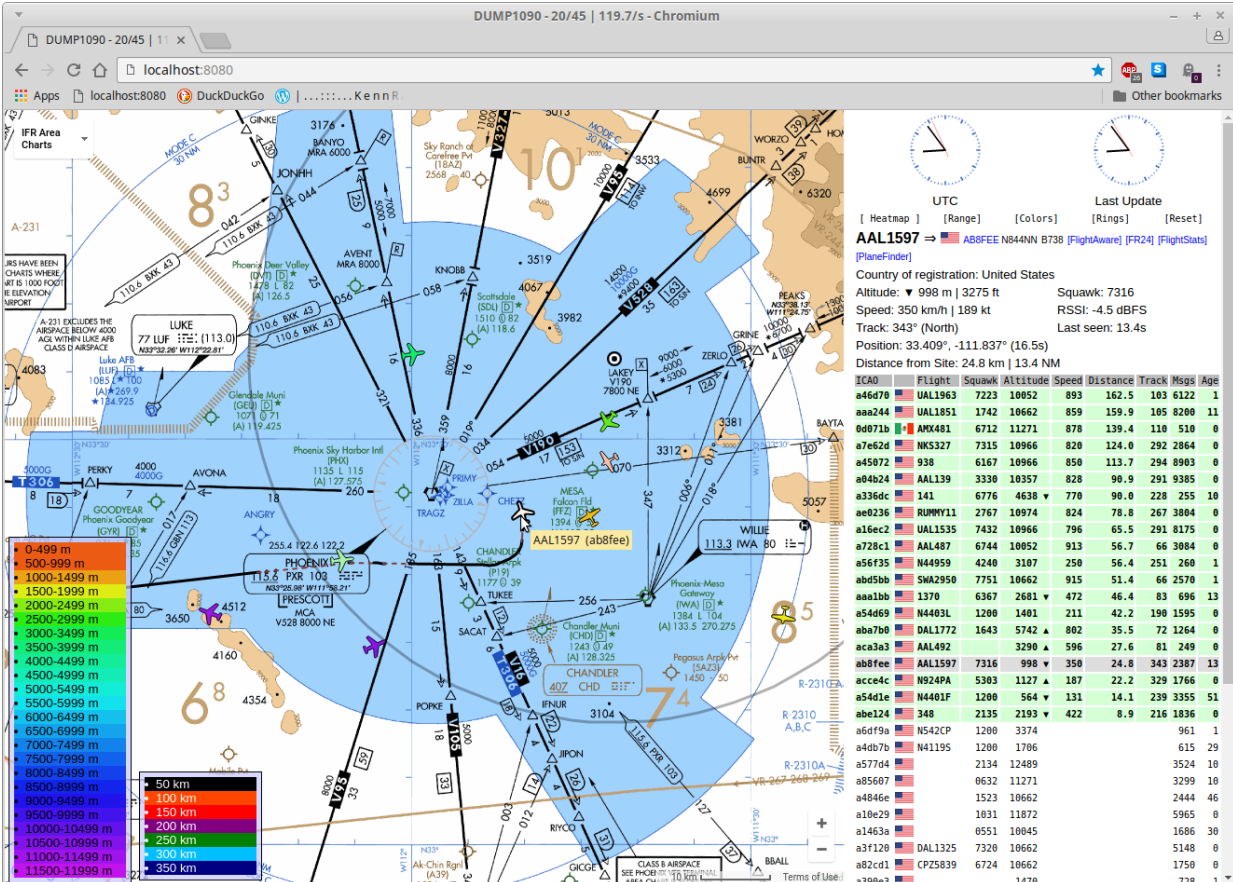
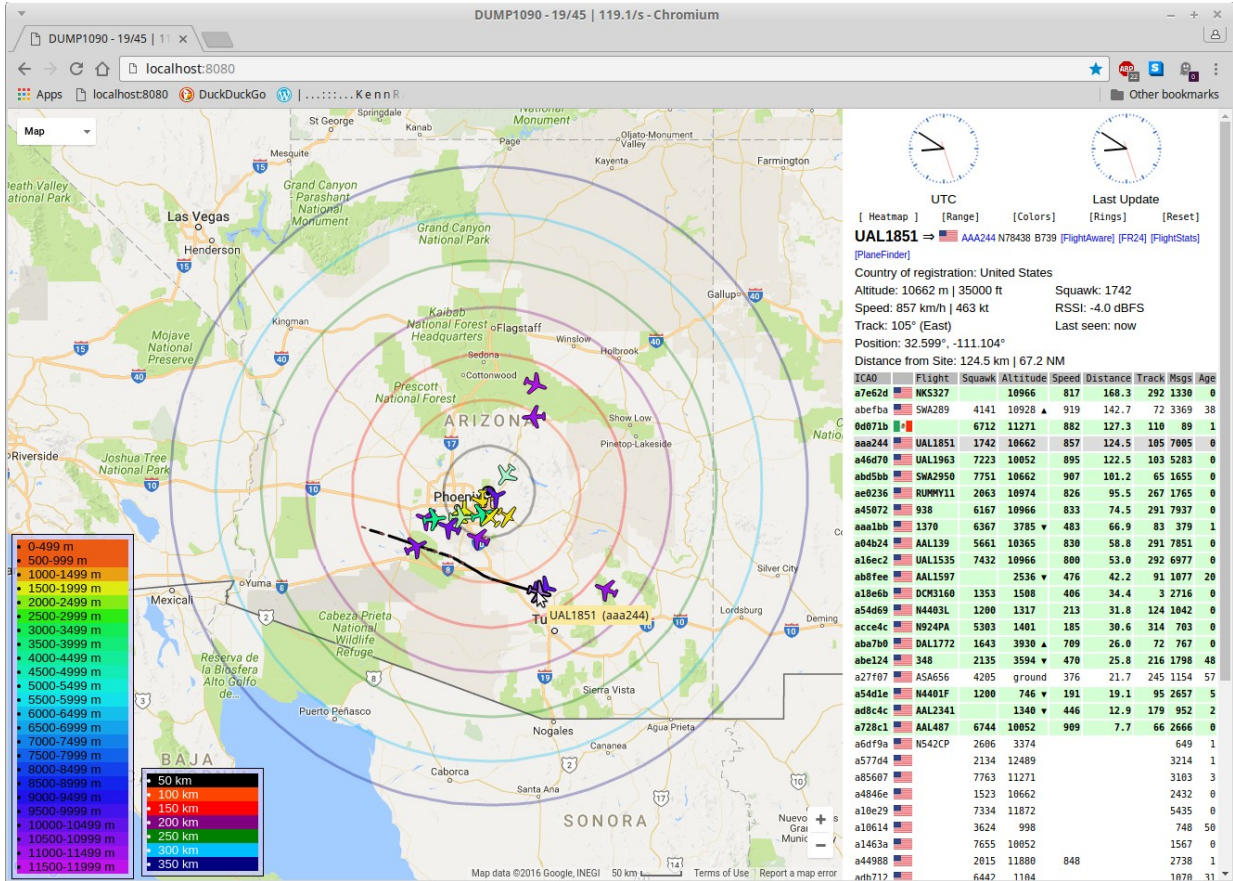
```
#!/bin/bash
#A bash script to start Ted Sluis' dump1090 with optimal flags
cd /home/yourname/dump1090
./dump1090 --interactive --net --enable-agc --gain -10 --ppm 12
--phase-enhance --lat 33.0 --lon -111.0 --oversample --dcfilter
```

Note: make sure those last two lines are all on one line in the editor. Save as d1090 then Right-click the script file, click 'Permission' and tick 'Allow executing file as a program'. At the terminal just `d1090` to start. Easy!

## Terminal

File Edit View Search Terminal Help

Hex	Mode	Sqwk	Flight	Alt	Spd	Hdg	Lat	Long	RSSI	Msgs	Ti\
A44988	S			38975	459				-6.5	28	1
A27F07	S		ASA656	16800	376	169	34.263	-112.177	-5.7	18	15
ADB712	S	0351		5575					-6.4	51	5
A6B554	S			2625					-3.1	70	0
A54B0A	S			1800					-12.0	2	59
A1E6B9	S			14600					-13.3	2	17
A46D70	S			33000	481	102			-8.0	54	0
A1463A	S	4761		33000					-5.1	78	1
ABE124	S	4135		33250	423	246			-8.3	12	6
A23A02	S			13250					-5.8	127	1
AAA244	S	1742	UAL1851	35000	457	108	33.215	-113.070	-6.8	646	0
A10614	S	0460		17925					-6.7	52	1
A1239C	S	0733		12850					-1.4	144	26
A0B473	S			2925					-5.1	41	51
A2E6DD	S	1200		5475	116	086	33.059	-111.749	-6.9	42	13
A54D0A	S	1200	N4400W	1975	74	239	33.478	-111.701	-1.7	839	0
000000	S	4022							-11.5	4	10
A54D4D	S	1200	N4402T	2425	101	034	33.469	-111.677	-4.0	826	5
A2D4D6	S	1200	N28154	3650	116	100	33.394	-111.606	-5.0	552	0
ABEFBA	S	4141	SWA289	17025	394	069	33.379	-111.700	-2.7	1105	1
A54D50	S	1200	N4402W	1900	91	232	33.470	-111.713	-5.8	622	46
A54D87	S	1200	N4404F	3825	120	098	33.393	-111.588	-5.2	795	3
AD7754	S	2473		35000					-5.1	747	0
A16EC2	S	7432	UAL1535	36000	435	293			-7.2	527	2
A4F179	S	0362	N4177L	3400	93	332	33.413	-111.743	-5.0	962	0
AA9E04	S	0766		28725					-2.9	850	0
A10E29	S	7334		38975					-2.1	1801	0
A04B24	S		AAL139	34000	447	293	32.612	-110.254	-5.2	827	0
A4846E	S	2642		21300					-3.8	821	0
A17343	S			17875					-2.5	425	0
A533FB	S	7622		6950					-3.5	933	0
A85607	S	7401		37000					-4.0	995	1
AD20C5	S	1303		3250					-5.9	1682	4
A2EDB4	S	0635		9075					-1.5	501	29
A18E6B	S	1353	DCM3160	13300	374	298	32.878	-111.124	-3.1	1404	0
A45072	S		938	36000	452	292	32.693	-110.505	-4.2	1683	0
A577D4	S	2025		41000					-5.5	1133	3
AB509C	S	1200	N828WW	3675	117	127	33.435	-111.693	-1.5	1172	3
A54D1E	S	1200	N4401F	1975	75	230	33.477	-111.708	-4.7	1390	11
AC7F35	S	7711		37000					-2.2	1342	0
780A29	S	1532	CPA097	34000	511	345	33.313	-111.454	-3.6	3069	1
A0B199	S	7326	AAL2473	30975	490	105	32.784	-111.422	-3.4	4056	0
A445EF	S	0455	UAL1590	35000	502	066	33.506	-111.772	-1.9	4378	0
A08EFD	S	7347	AAL1024	32975	490	105	32.748	-111.225	-3.5	3891	0
A516E0	S	7275		34000					-2.8	1720	0
A4DB7B	S	1200	N4119S	5550					-3.4	429	44
A2856E	S	2643	UAL1672	38000	442	266			-7.8	5044	49
AD7471	S	2040		41000					-4.0	995	0
AD3556	S	0751	N950H	32275	488	098			-9.5	6591	37
ADBEF0	S	3175	AAL1278	37000	463	103	32.401	-109.893	-7.5	5075	1
A4F8E0	S	4723		6575					-6.3	671	0
ADECD7	S	7704	N997CA	4425					-2.8	515	1



## **Advanced Aircraft Tracking with Virtual Radar Server**

While Dump1090 is a good ADS-B decoder, there's another choice when it comes to web serving & mapping functions. In this section, I'll document how to use Dump1090 as the receiver/decoder and VRS (**V**irtual **R**adar **S**erver) to show their location.

1. Install your choice of Dump1090 as explained previously.
2. VRS is a .net application. To run it in Linux we need to install mono. Open a terminal and type:

```
sudo apt-get install mono-complete
```

3. Visit <http://www.virtualradarserver.co.uk> and download the mono tarball.
4. Extract/decompress the tarball. I put the VirtualRadar folder in my home folder and will assume that convention in this guide.
5. We need to change the port number that VRS serves on to something besides 8080 so it doesn't conflict with the dump1090 web server. This must be done manually. Open a terminal and type:

```
cd VirtualServer
```

```
mono VirtualRadar.exe
```

Click 'Help' > 'About...' and at the bottom should be a link to the configuration folder. Click it and it will open that folder. (just in case it doesn't, we're looking for the /home/username/.local/share/VirtualRadar folder. Note the period in front of .local means it's a hidden folder.)

In this folder, create a text file called InstallerConfiguration.xml then open it and paste this into it:

```
<?xml version="1.0" encoding="utf-8" ?>
<InstallerSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<WebServerPort>8888</WebServerPort>
</InstallerSettings>
```

I chose 8888 because it's easy to remember and as far as I know nothing else uses it. Use any port you like so long as it isn't being used by something else on your system. Save this file and close your text editor. Shut down VRS.

6. Launch dump1090 at the command line:

```
cd /dump1090  
./dump1090 --net --interactive
```

(Or via the script I showed earlier if you created one.)

7. Launch VRS at the command line:

```
cd /VirtualRadar  
mono VirtualRadar.exe
```

If you like, you can create a script file in your home/username/bin folder called vrs with these lines:

```
#!/bin/bash  
# A script to start Virtual Radar Server  
cd /home/username/VirtualRadar  
mono VirtualRadar.exe
```

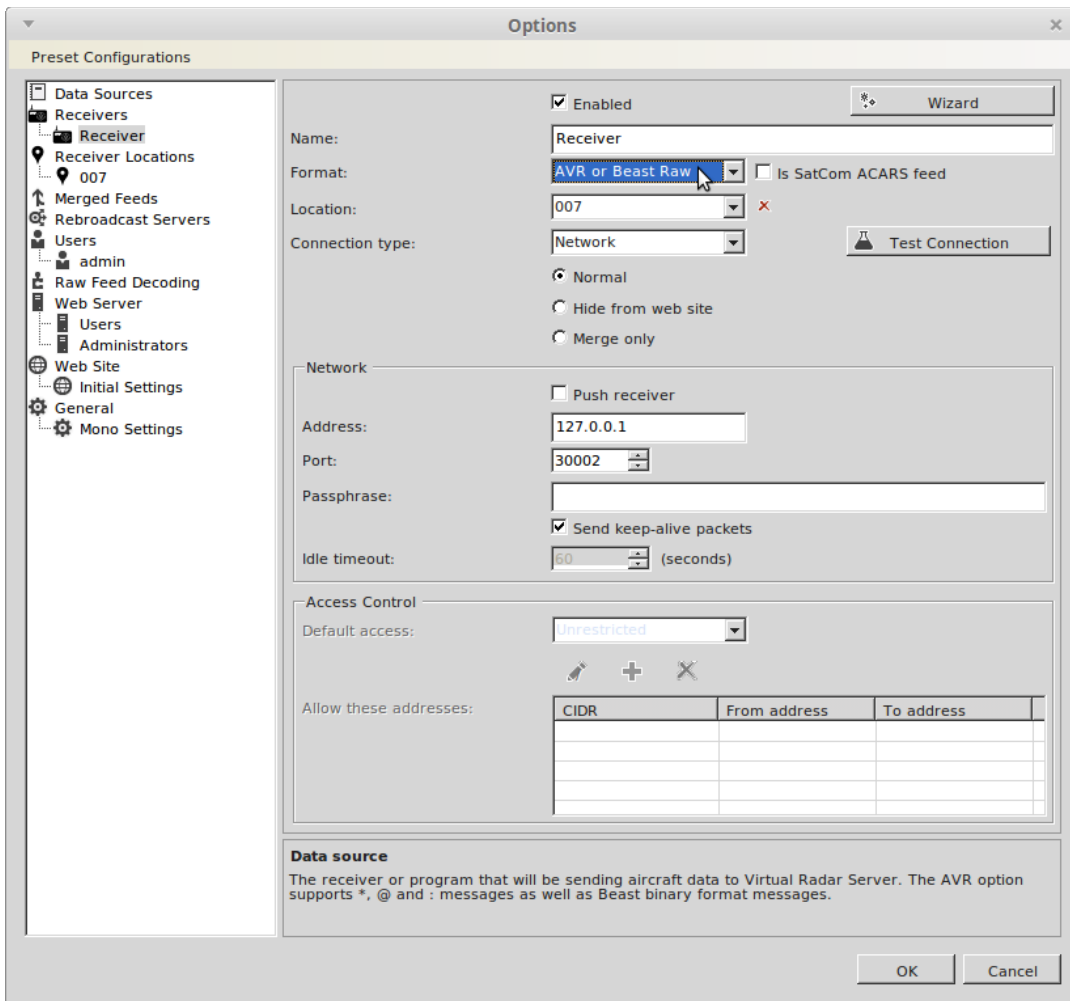
Right click on vrs and click > permissions > 'Allow executing file as a program'.

8. Click 'Tools' > 'Options' > 'Receivers' the green + sign to add a receiver. Run the wizard and tell it you have 'A software defined radio' > 'Dump1090' > and tick 'Yes' to 'Is the SDR decoder running on this computer'. All other settings should be left at default values. Hit OK.

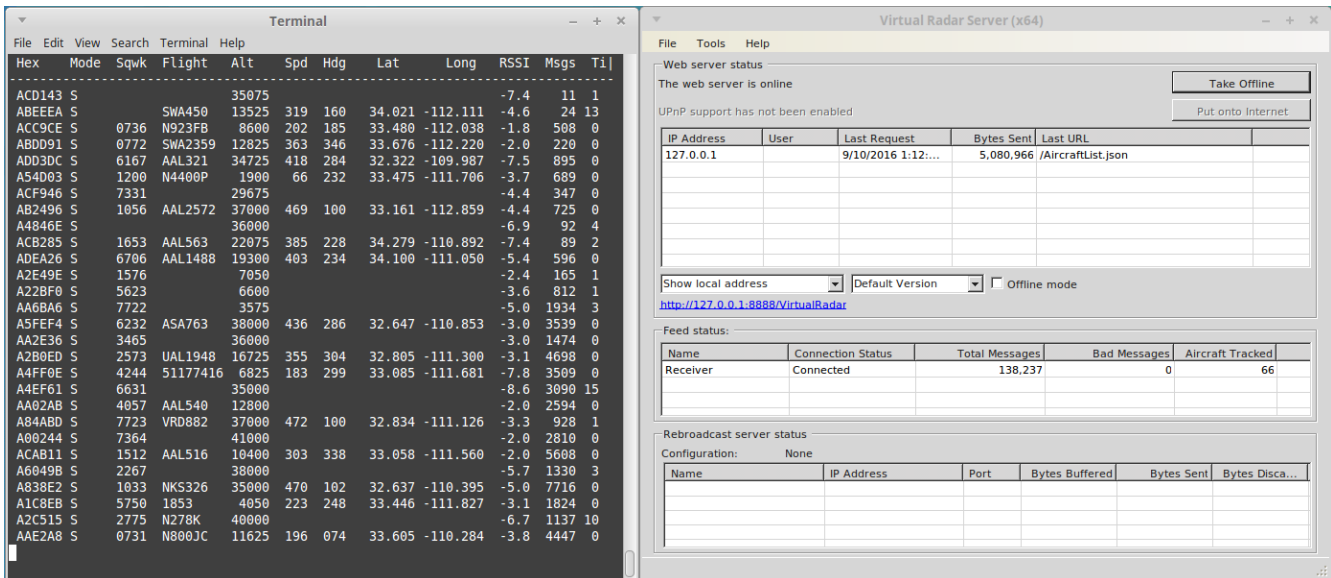
9. Back at the main VRS screen under 'Feed status' you should be seeing Receiver - Connected - a number of Total Messages rising - and some Aircraft Tracked.

Right above that should be a blue hyperlink to <http://127.0.0.1:8888/VirtualRadar>. Click and a browser should open with aircraft on the map.

10. The map might start centered over London. If so move to your location. Now you would think that by looking under 'Tools' > 'Receiver' > 'Receiver Locations' and entering your lat & lon it would start the map centered around your location, but for me it hasn't. If I find a fix I'll post it on the next update.



Under Tools > Options > Receivers.



Dump1090 receiving & decoding while VRS serves & plots.

**N486AS**  
Alaska Airlines  
United States  
Boeing 737NG 990ER/W

Altitude: 37975 ft | Vertical Speed: 0 ft/m amsl | Speed: 503.4 mph | Heading: 276.0° | Distance: 55.36 nmi | Squawk: 6232 | Engines: Twin jet

Species: Wake Turbulence: Landplane | Medium

Route: KMCO Orlando, United States | KSAN San Diego, United States

www.airport-data.com : www.airliners.net : www.airframes.org  
Show on map : Enable auto-select : Submit route correction

Silhouette	Flag	Reg.	ICAO	Callsign	Route	Altitude	Speed
		N531ND	A6B554			2150 ft	
		N591NW	A7A25F			34000 ft	
		N7828A	AA9B65			40000 ft	
		N652AW	AB943D			27125 ft	
		N921US	ACC3B4			25850 ft   473.0 mph	
		N308NV	A33E4A	172		9150 ft   291.4 mph	
		N906JF	AC85FC			32000 ft	
		N67815	ABFC09	UAL1547	KFLL-*KLAX	36000 ft   493.1 mph	
		N533NK	A6BCC8	NKS108	KSAN-KDFW	35000 ft   551.3 mph	
		N717SA	A9979F			15275 ft	
		N907SW	AC8AD2			19225 ft	
		N8620H	ABD8A5	SWA1301	KMSP-*KCLE	38000 ft   503.2 mph	
		N213WN	A1C6A8			23225 ft	
		N985NN	ADBEF0	AAL2502	KSFO-KDFW	37975 ft   494.6 mph	
		N919WN	ACBA1B			29825 ft	
		N38257	A465DF	UAL469	KIAD-KEWR	38000 ft   496.8 mph	
		N201LV	A196C8			38000 ft	
		N361SW	A411A8			12650 ft	

Click on a plane to see its path and info.

Explore the options under 'Menu'. There's also an optional Web Admin Plugin available from <http://virtualradarserver.co.uk> that adds a web-based user interface to monitor and configure the server. VRS can also be made public so others can view your radar system remotely.

If you really want to get the most out of your ADS-B system, look into purchasing an ADS-B pre-selector filter, an LNA (Low Noise Amplifier) aka a pre-amplifier and either purchasing or building a coaxial collinear antenna.

<http://balarad.net> - The best documentation I've found on DIY ADS-B antennas

## **Spectrum Analysis with Qspectrum Analyzer**

A limitation of RTL-SDR dongles is a maximum bandwidth window of about 2.4 MHz and general purpose radio applications store a limited history. If we want to, for example, monitor the entire 2 meter ham band from 144 to 148 MHz for a full day and log the results we'll need another tool. The command line utility `rtl_power` included with the Osmocomm drivers we installed earlier can do this but there's a better way. Using an enhanced tool called `rtl_power_fftw` we can rapidly scan 2.4 MHz chunks in rapid succession and visually display it with Qspectrum analyzer.

1. Install `rtl-power-fftw` and required libraries:

```
sudo apt-get install libfftw3-dev libtclap-dev
```

2. Retrieve, build and compile:

```
git clone git://github.com/AD-Vega/rtl-power-fftw
cd rtl-power-fftw
mkdir build
cd build
cmake ..
sudo make install
cd -L
rtl_power_fftw --version
```

If all has gone well you should see:

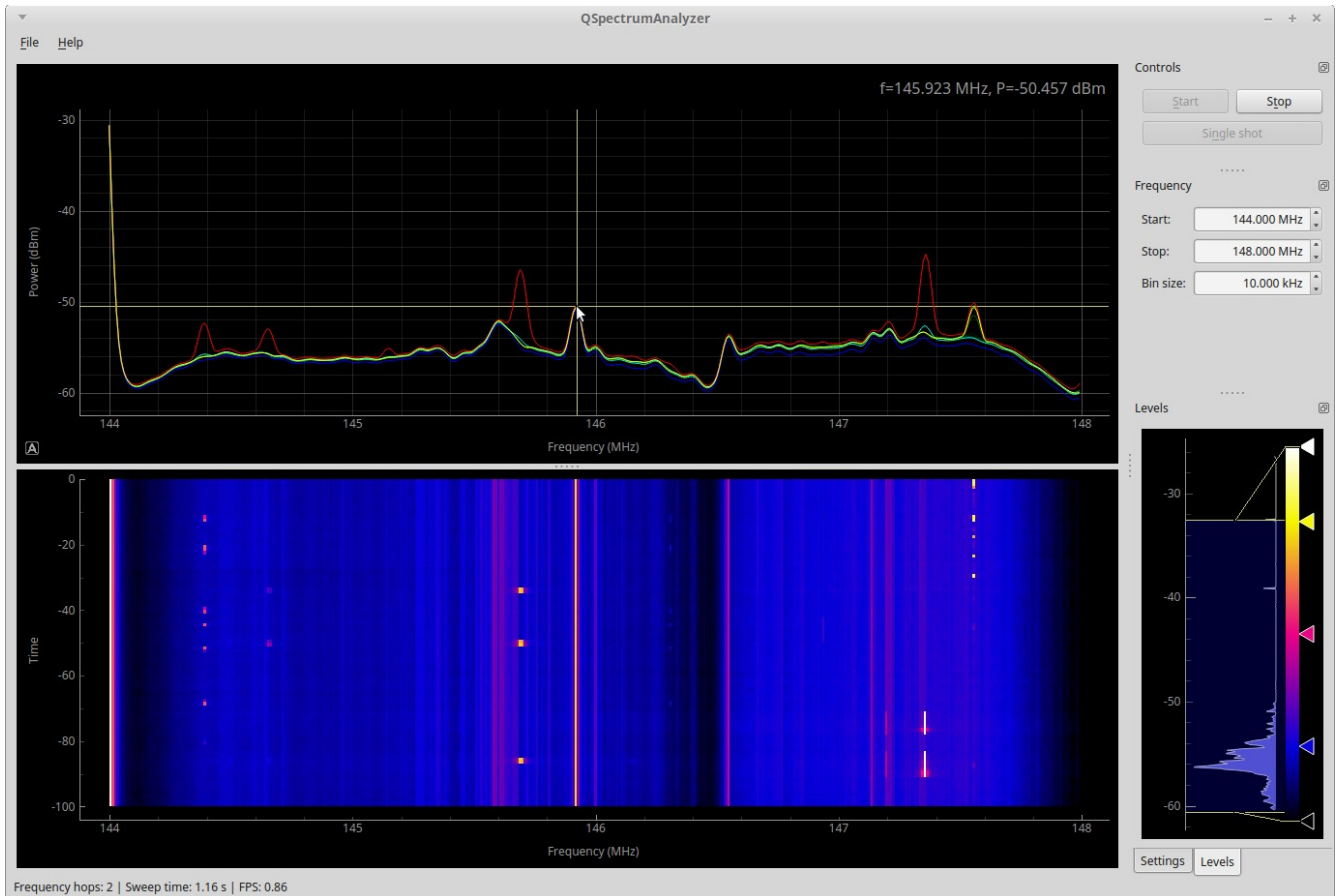
```
rtl_power_fftw version: 1.0-beta2
```

3. Install Qspectrumanalyzer and dependencies then launch:

```
sudo apt-get install python3-pip python3-pyqt4 python3-numpy
sudo pip3 install qspectrumanalyzer
qspectrumanalyzer
```



The GUI should now be running. Click 'File' then 'Settings' and change the backend to 'rtl\_power\_fftw'. Enter the desired Start and Stop Frequencies and click Start. Hover the mouse over any peaks in the upper window to find out what it's frequency is. Note this is a power over time monitor and not a radio application. You won't hear anything, what you'll see is where transmissions are occurring across the given frequency range.



In the above image, I'm scanning the entire 2 meter amateur radio band from 144-148MHz. Each spike represents a signal. The waterfall below represents time, so the dots are short transmissions, likely repeater identifications.

For more information, visit the developer's github pages at:

<https://github.com/AD-Vega/rtl-power-fftw>

<https://github.com/xmikos/qspectrumanalyzer>

## **Monitoring Trunked Radio Systems with SDR Trunk:**

Trunking radio systems are more complex than traditional radio systems as are the applications used to track them. In brief, a typical standard analog radio system will have one or more mountaintop transmitters as needed to cover an area. There might be a few channels or a dozen or more. Mobile units transmit up to the tower at one frequency and the tower retransmits out on another frequency. All the scanner enthusiast needs to know is the output frequencies.

Now imagine a city has 20 different departments each with their own set of mountaintop transmitters and channels. That's a LOT of hardware to maintain. It's also hundreds of dedicated frequencies and spectrum is getting crowded. Most departments probably aren't using most of their channels most of the time anyway.

So imagine all 20 departments share one radio network and a pool of channels. A computer picks a clear frequency and the right call goes to the right department. That, in a nutshell is trunked radio. Some are analog, others are digital. A few are encrypted.

**SDR Trunk** is the only trunking software I've found so far that works in Linux. Installation and usage is a bit more complex than a standard radio application.

<https://github.com/DSheirer/sdrtrunk>

The install instructions are generally complete, a few things to note:

Many municipalities are using the APCO25 digital standard. To listen in you'll need to download and compile the JMBE libraries, created by the same author. The installation guide can be found in the wiki on the sidebar under APCO25. At this time, only phase I (aka type I) is supported.

When creating a new system in SDR Trunk, you're prompted to enter a frequency. Usually that will be the control channel frequency. This can be found for the system you want to monitor on the radioreference.com database. You don't need to enter all the frequencies for a site, just the control channel and SDR Trunk will tune in to and follow across the other channels.

At this point, the best way to accurately follow apco25 type II trunked system is with a digital scanner or via an online feed such as <http://www.broadcastify.com>

## **Other interesting RTL-SDR apps:**

**rtl\_433** receives and decodes sensor data from remote temperature, rain gauges, thermostats, etc. in the ISM bands. It's not limited to 433MHz. A good antenna can pick up several from your immediate neighborhood.

## **Resources:**

<http://sdr.osmocom.org/trac/wiki/rtl-sdr> - Source of the RTL-SDR drivers.

<http://rtl-sdr.com> - Great place to buy and learn about RTL-SDR

<http://distrowatch.org> - A one-stop index of the top 200 Linux distributions

<http://linuxmint.com> - Currently the most popular Linux distribution

<http://balarad.net> - The best documentation I've found on DIY ADS-B antennas

[qsl.net/na4it/dbgp.html](http://qsl.net/na4it/dbgp.html) - A cheap, easy & good dual-band ground-plane antenna

[wikipedia.org/wiki/Electromagnetic\\_spectrum](http://wikipedia.org/wiki/Electromagnetic_spectrum) - A primer on radio waves

[radioreference.com/](http://radioreference.com/) - Excellent database of frequencies

[qsl.net/kb5wck/antenna.html](http://qsl.net/kb5wck/antenna.html) - A good source of antenna design calculations

<https://linuxconfig.org/bash-scripting-tutorial> - Bash Scripting

<http://www.howtogeek.com/67469/the-beginners-guide-to-shell-scripting-the-basics/> - Introduction to Shell Scripting

<http://www.programmingsimplified.com/c-program-examples> - C tutorial

<http://www.cplusplus.com/doc/tutorial/> - C++ tutorial

<http://www.github.com> - Mass quantities of source code!

## **In Closing:**

I'll be updating this guide over time as I discover new software, hardware, tricks and mods. I welcome constructive feedback. You can contact me through my blog at <http://ranous.wordpress.com> where you'll find additional RTL-SDR articles. If you send me a coherent message I'll most likely respond.

*Thanks for tuning in!*