

RTL-SDR for Linux Quick Start Guide

Second Edition v2.18

by Kenn Ranous KA0SBL



RTL-SDR for Linux Quick Start Guide

v2.18 by Kenn Ranous - KA0SBL

Page 3	Introduction
Page 4	Antennas & Feedlines
Page 5	Installing Drivers
Page 7	Gqrx - General Purpose Radio & Spectrum Software
Page 8	Frequency Correction - PPM adjustments
Page 10	Cubic SDR - General Purpose Radio & Spectrum Software
Page 11	ADS-B Aircraft Tracking with Dump1090
Page 14	Monitoring Trunked Radio Systems with SDR Trunk
Page 15	Lightning Detection with SDR
Page 17	Additional Resources

Introduction:

If you've installed Linux on your PC hopefully you'll find this guide quick and easy in any Debian based distro. Created in Linux Mint v18.3, I'll cover the software I've found to be complete and working. Most things RTL-SDR have come a long way since inception. Updates still sometimes break things & documentation isn't always complete, but community support is good and the reward is worth it. If you have some amateur radio background but are new to SDR many of the controls will be familiar. If not, you'll learn as you go.

First, a quick primer on what RTL-SDR is. In Europe, TV uses the DVB-T standard. Inexpensive USB receivers are available to tune in on a computer. Based on the RTL2832 demodulator & USB interface chip, tuning is done by the R820T2 chip. By installing specialized drivers and software, these devices can be used as a **Software Defined Radio** capable of receiving 24 – 1800 MHz in blocks of ~2 MHz at a time. Bypassing the tuner chip by switching the receiver to direct sampling mode enables DC – 24 MHz reception.

A few caveats: North American TV is on the ATSC standard so these devices can't be used to watch TV. Also, the RTL hardware was originally designed to receive strong broadband signals. It lacks the sensitivity and selectivity of expensive receivers, so it can be over-loaded by strong signals such as FM Broadcast and pagers. They're also susceptible to RFI. Filters and pre-amplifiers are available to overcome some of these limitations.

The unit used here is version 3 of the RTL-SDR.com dongle. It has a 820T2 tuner, TCXO, anti-static diode, and a metal enclosure with heat-sink. It can be switched into direct sampling mode via software, no hardware mods needed. 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 CPU with 2GB RAM or better. None of the applications here are highly demanding of Video or HDD.

Antennas & Feedlines:

Regardless of cost, any radio is only as good as its antenna. There's an abundance of information on-line and in ARRL publications, so I'll be brief.

- Rule #1, there is no such thing as the perfect antenna for all situations. It does not exist. Like anything, there's compromises and trade-offs. A broadband antenna may be more useful across a wider swath of spectrum, but it won't gather as much signal as an antenna designed for a specific frequency. Also, a broadband antenna collects unwanted signals which may overload the receiver.
- Rabbit ears are a good starter antenna. It's easy to adjust the length and orientation.
- Attic or outdoor antennas for TV may work well. Note they are usually in the horizontal plane, hence they are horizontally polarized. Ham, Public Service, NWR and similar services tend to use vertical antennas and so emit vertically polarized waves. Because of this difference, they may not be stellar performers. Newer TV antennas designed for HDTV may lack the ability to pull in FM radio. TV antennas are broadband by design.
- Discone antennas are broadband and work well for general purpose use.
- 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. (<30MHz)
- Ground planes, J-poles, dipoles, yagis, quads and log periodic antennas can offer superior performance when designed for a specific band.
- RG-6 coax and type F connectors, the same type used for cable and satellite TV is perfectly acceptable to use. It performs well up to several GHz, is inexpensive and commonly available. Type F connectors are probably the easiest type of connector to work with. Compression fittings are preferred, but a good quality side-crimp properly done is just fine. SMA to Type F connectors are available online.

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 (right-click) an administrator.

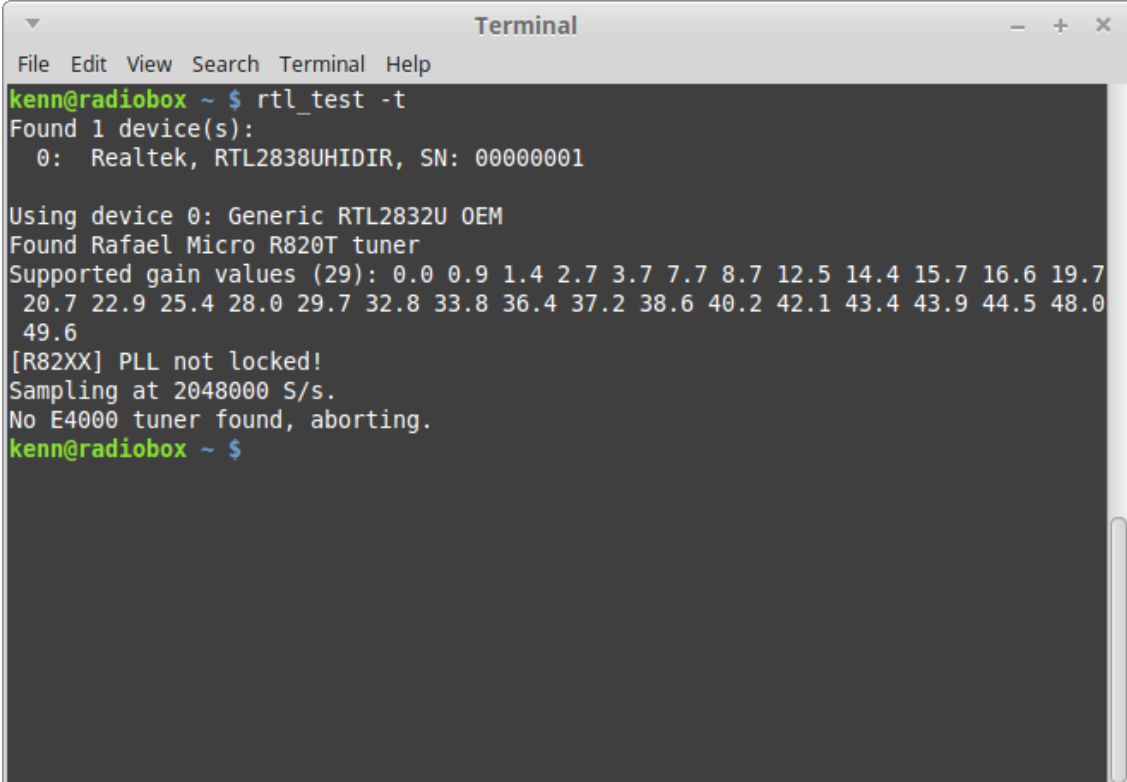
B. Create a new file 'blacklist-rtl.conf' and add this one line:

```
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
```

A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is "kenn@radiobox ~ \$". The command "rtl_test -t" has been executed. The output shows: "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.", and the prompt "kenn@radiobox ~ \$".

```
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.

Ggrx General Purpose Radio/Spectrum Analyzer:

This can be found in the Synaptic Package Manager. To get the latest version, visit <http://ggrx.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. If it doesn't appear, click the circuit board icon on the toolbar and check the following:

Device: Realtek RTL2838xxx

Device String: rtl=0 [Normal use >24MHz]*

Device String: rtl=0,direct_samp=2 [Direct Sample mode <24MHz]*

Input rate: 1800000 [aka Sample Rate]*

Decimation, Bandwidth, LNB LO: 0

** Note you're choosing one of the above Device Strings on startup, the first for VHF/UHF OR the latter for HF/MF.*

*** This is the amount of radio spectrum you'll see. When tuning in the HF/MF bands it's sometimes helpful to select a lower value. A higher value means you can view a larger slice of spectrum, but go too high and it gets choppy.*

Click OK then the > Power Button in the upper left corner to start receiving. Click or mouse wheel on the digits to change frequency. On the right under Receiver Options select the appropriate demodulator. (ie Narrow FM for NOAA Weather Radio or WFM for broadcast FM).

Hover over each item to see what it does. Under Input controls, toggle hardware AGC and adjust LNA gain to maximize the signal but not so high that static or distortion takes over. Adjust for a noise floor of -60db to -70db. Note that you'll need to alter gain settings as you tune around the spectrum.

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

Frequency Correction with GQRX:

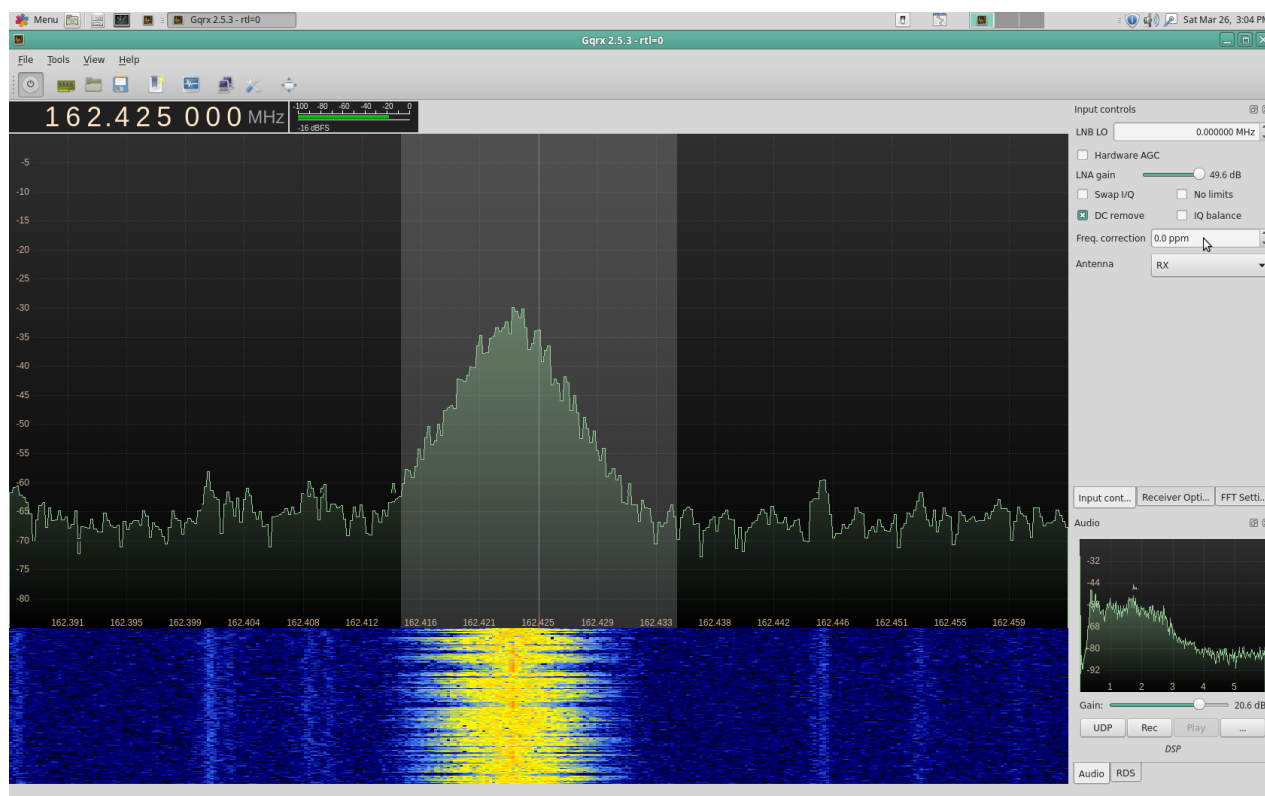
Later model dongles such as RTL-SDR's v3 have improved TXCO's for accurate frequency readouts regardless of temperature. For most uses in normal mode no calibration is needed.

However, older dongles and ones in direct sampling mode may need PPM adjustments. Once you've determined your PPM error correction factor you can use that value in most other RTL software. As long as the dongle doesn't overheat, this value should remain reasonably constant.

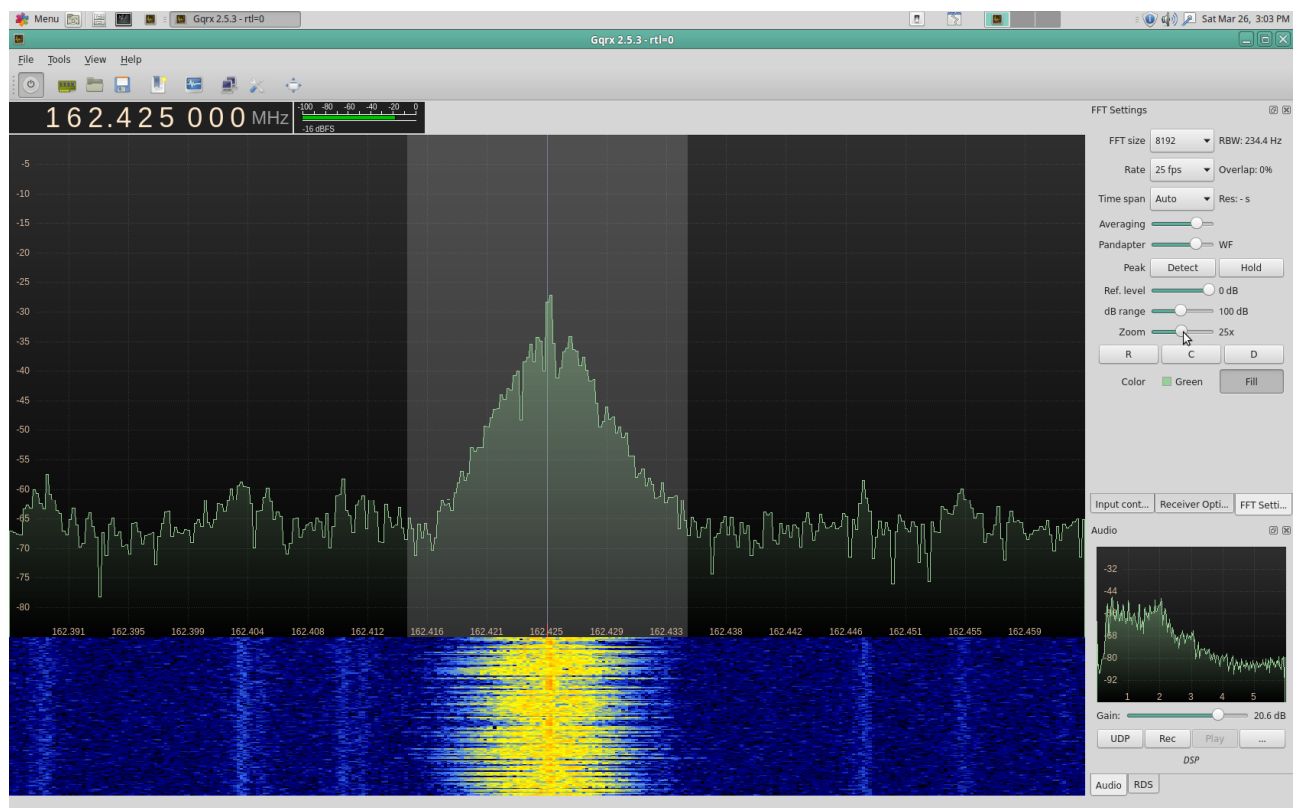
To determine your dongles PPM, launch GQRX and tune your SDR to one of the National Weather Radio frequencies. Give the unit at least 15 minutes to warm up.

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.



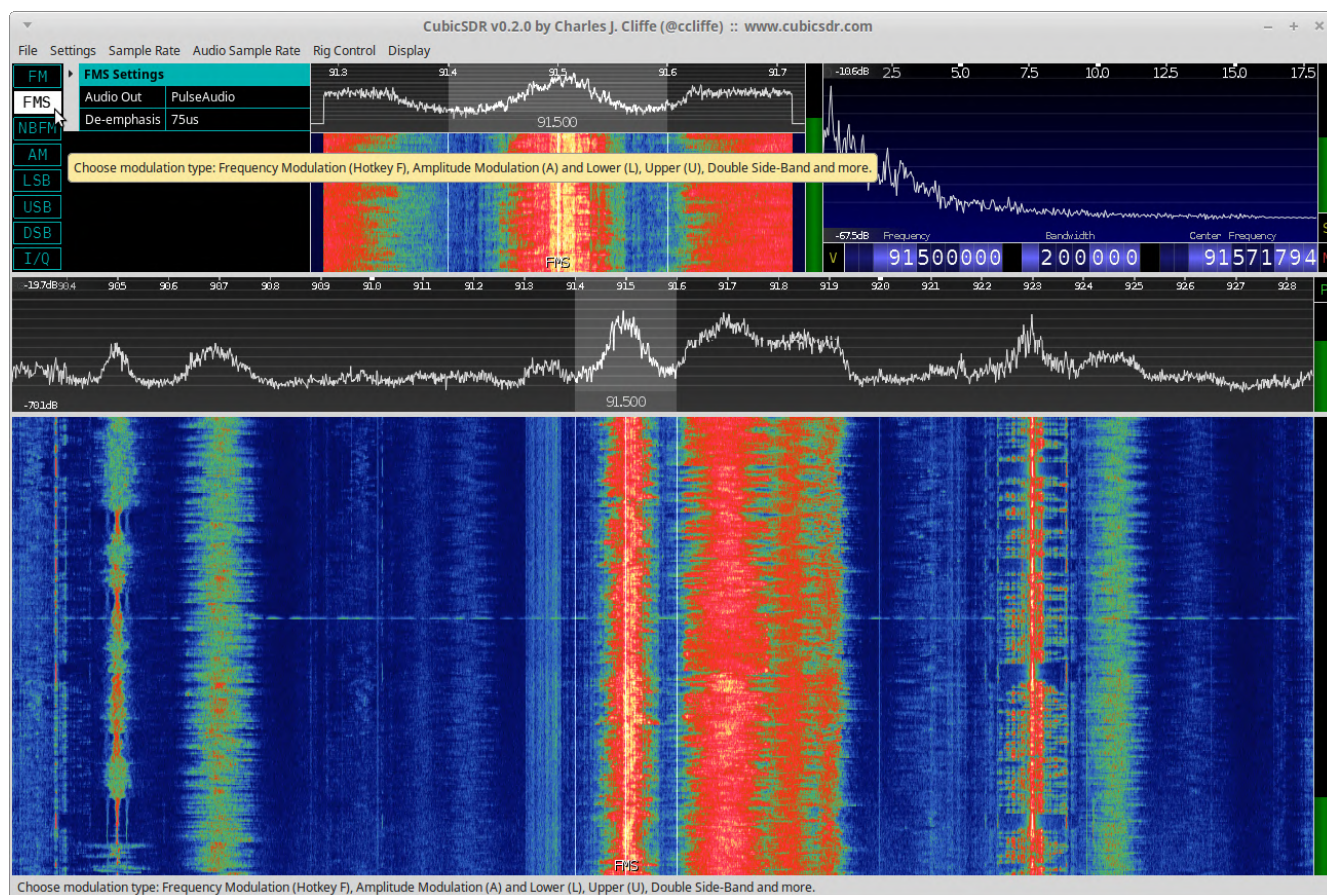
Adjust your PPM + or - until tuning line is centered on signal.

To check and adjust your PPM in Direct Sampling mode in the HF bands, tune to one of the NIST time stations on 2.5, 5.0, 10.0, 15.0 or 20.0 MHz. Choose whichever station comes in best. These transmitters are very frequency accurate. The NIST WWV is quite the resource and worth the time for any radio enthusiast to know about.

<https://www.nist.gov/pml/time-and-frequency-division/radio-stations/wwv>

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 or get it via flatpak. My preferred method is to just download the .ApplImage file and place it in it's own CubicSDR folder under my home folder. 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' for VHF/UHF. Or, select Q-ADC under Direct Sampling to access HF & MW bands.



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 .ApplImage 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 (malcolmrobb, 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
```

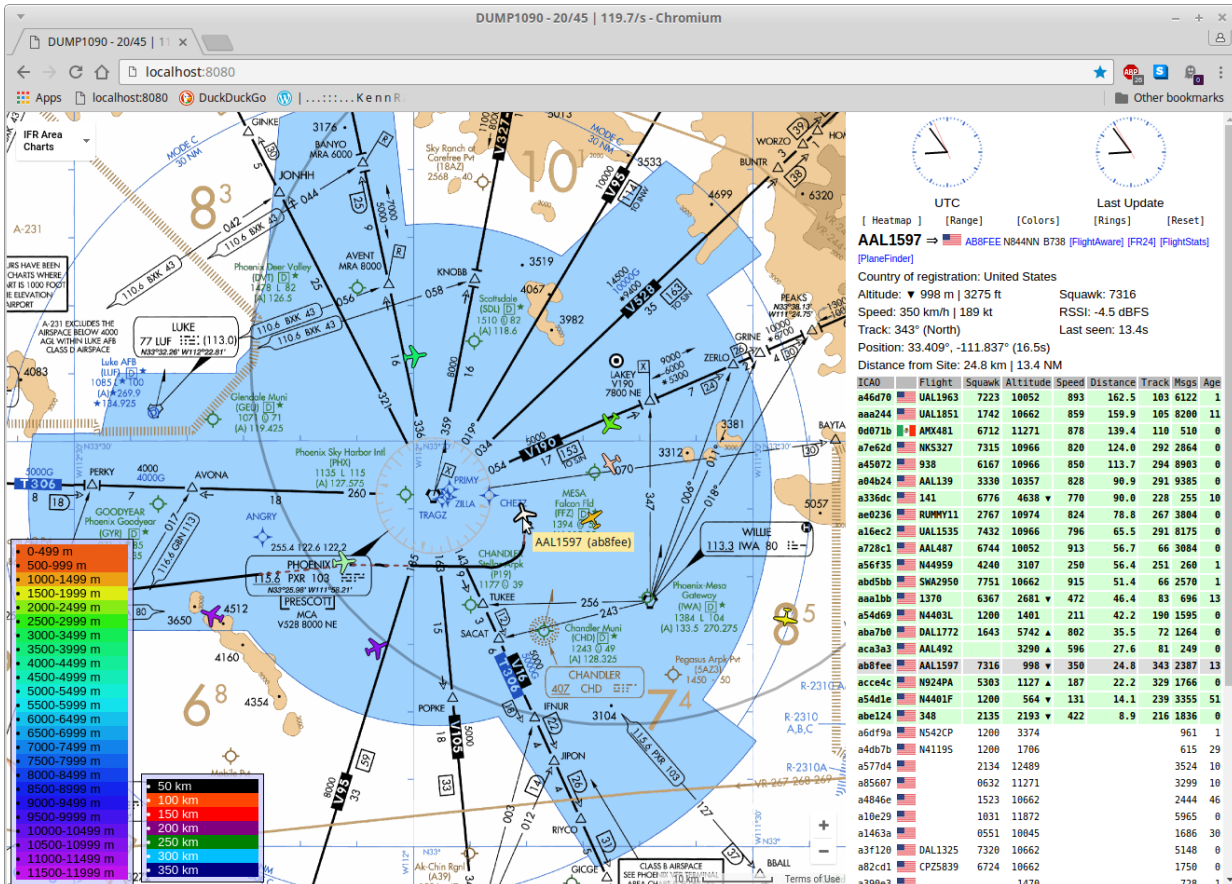
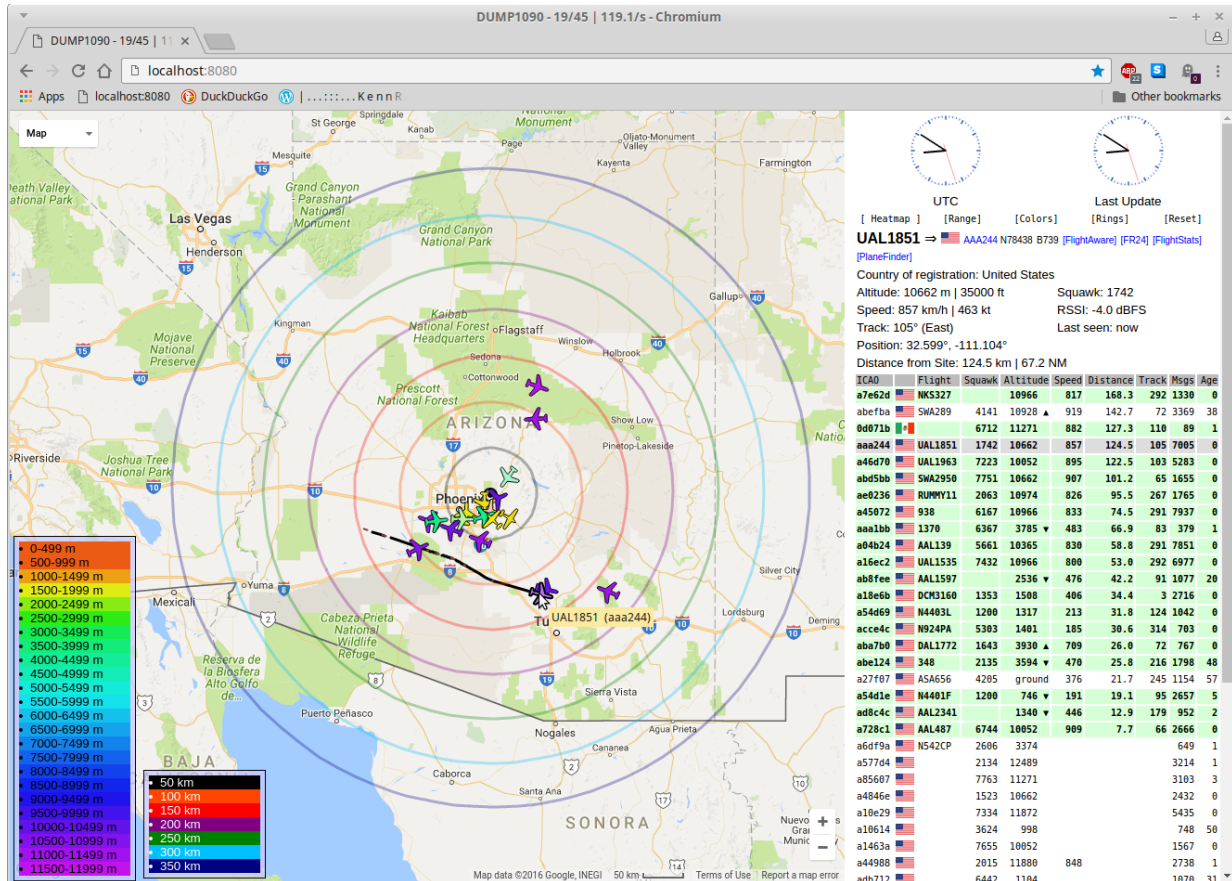
Within a few moments you should see a list of aircraft in the terminal window. Open a web browser and in the address bar enter localhost:8080 and you should see a map with aircraft. You might need to hit the browser refresh button once or twice to get it going.

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 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 --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 & exit then Right-click the script file > 'Permission' > tick 'Allow executing file as a program'. At the terminal just type `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



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:

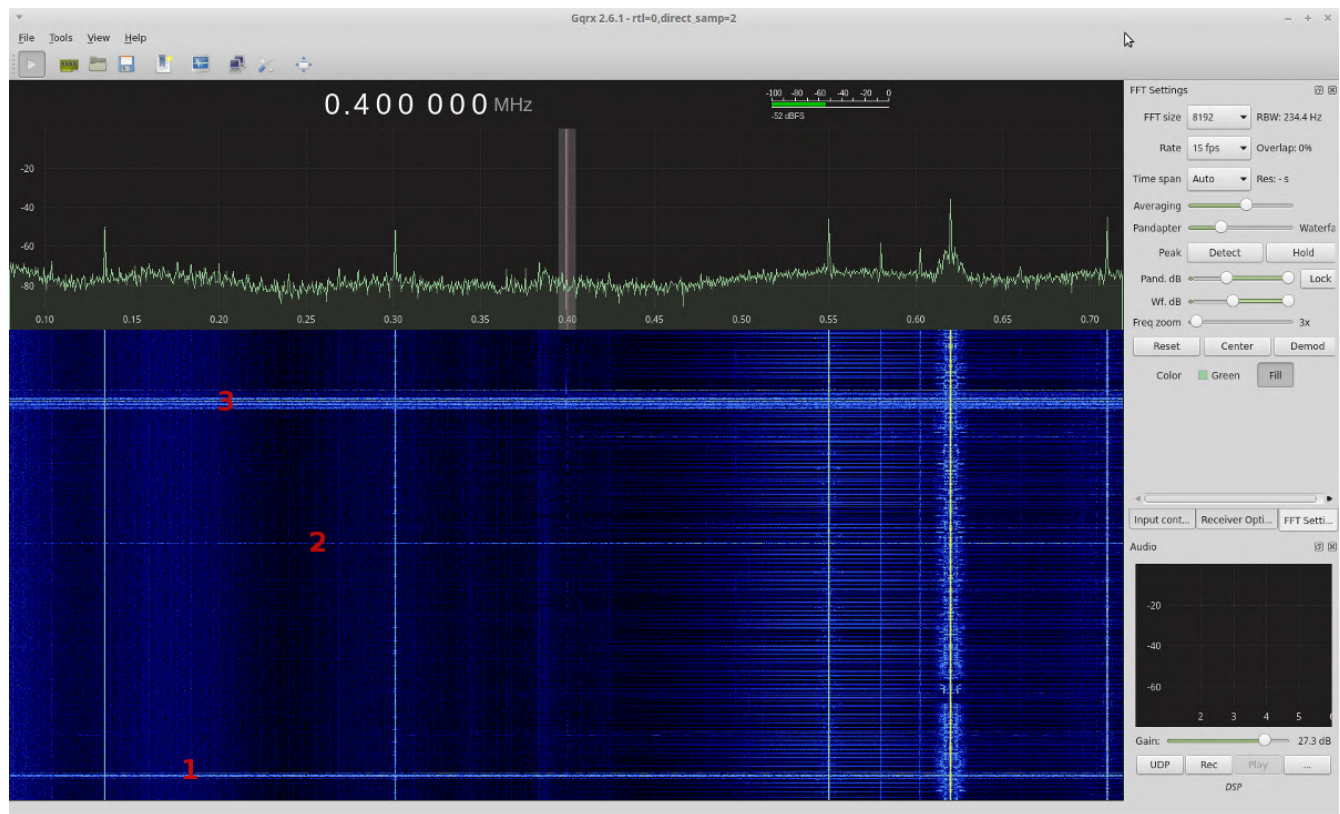
Many municipalities are using the APCO25 digital standard. To listen in you'll need to download and compile the JMBE libraries, created by the same author. The installation guide can be found in the wiki on the sidebar under APCO25. At this time, only phase I (aka type I) is supported.

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.

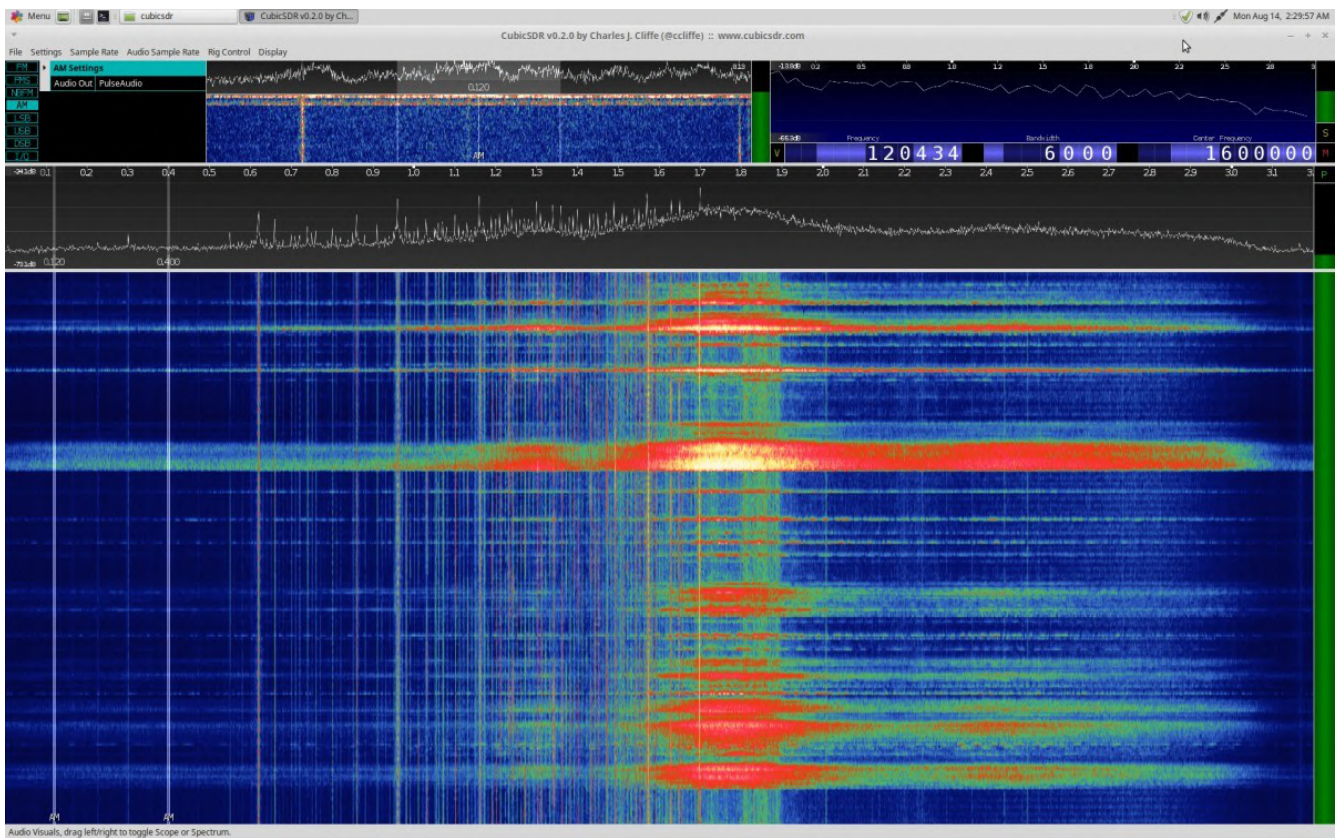
At this point, the best way to accurately follow apco25 type II trunked system is with a digital scanner or via an online feed such as <http://www.broadcastify.com>

Lightning Detection with SDR:

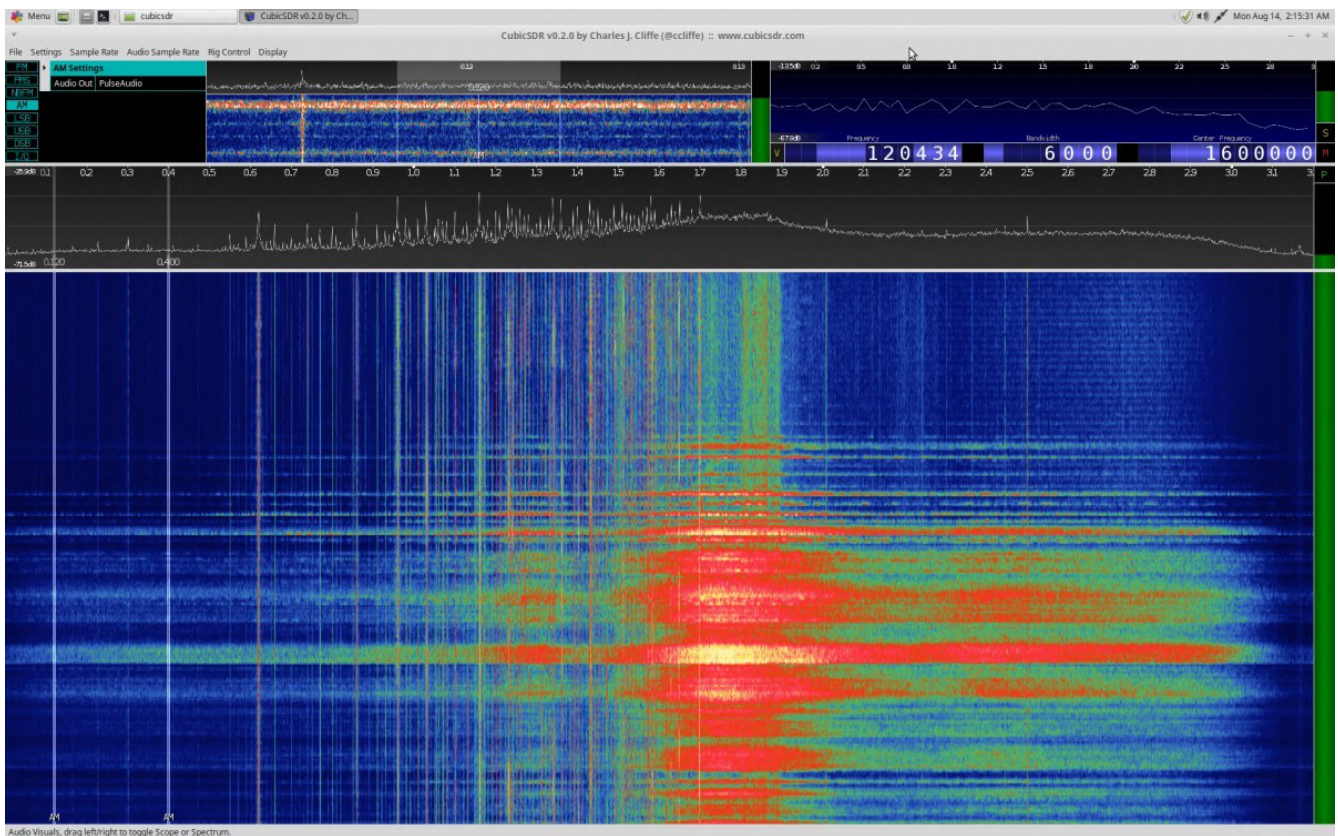
What follows is an informal experiment to detect approaching lightning. Here I'm using GQRX and CubicSDR in direct sample mode but any spectrum analyzer software and dongle that can tune VLF should do it. Many dedicated lightning detectors work by listening to frequencies below the AM broadcast band, hence I chose to monitor 0 to 600kHz. The antenna is a 1/2 wave trapped dipole for HF but any bit of wire at least a few meters long will work.



The waterfall display speed is set to represent ~30 seconds. Event #3 in this image was several rapid-fire and rather close by. I found that by tuning to a quiet part of the spectrum and setting squelch, lightning discharges produce an audible crackle, hence an audible alert without needing to listen to constant static or needing to watch the screen.



This was a particularly close & intense storm captured in CubicSDR.



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 - A cheap, easy & good dual-band ground-plane antenna

wikipedia.org/wiki/Electromagnetic_spectrum - A primer on radio waves

radioreference.com/ - Excellent database of frequencies

www.sigidwiki.com/wiki/Signal_Identification_Guide - ID that waveform

github.com/xmikos/qspectrumanalyzer - Scan large swaths of bandwidth

sdr.hu - A network of shared SDR's around the world.

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://sdr.hu> - A network of shared SDR's around the world.

In Closing:

Feel free to copy or distribute this guide as you see fit. My goal here is to consolidate and share my notes so others can bypass hours of annoying research and just get to the fun stuff. :) I'll be updating this guide using the convention vMonth.Year.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!

- Kenn