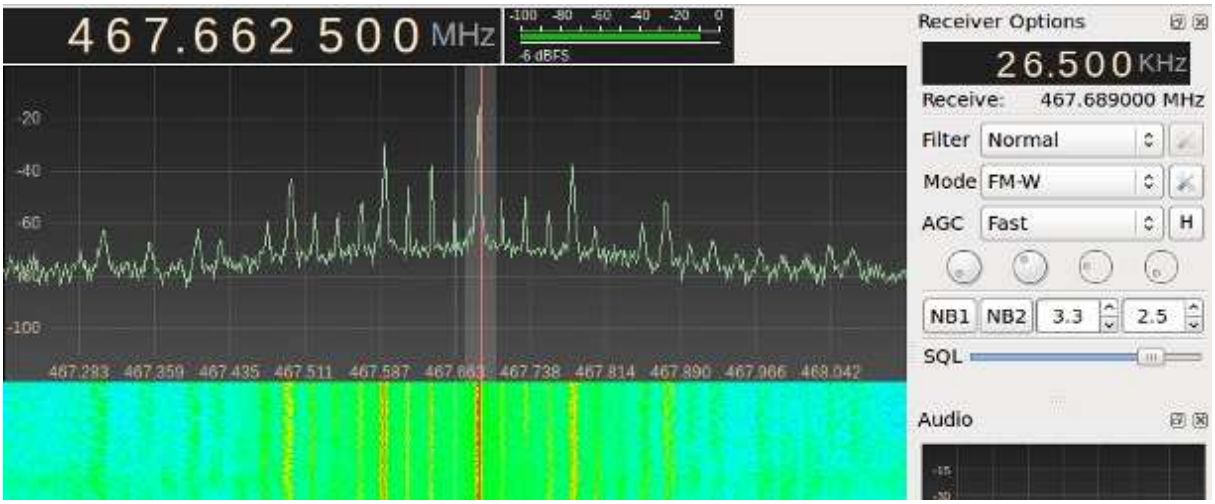
Getting Started With RTL-SDR

Tom Nardi



The last few months have seen an explosion of activity in the field of Software Defined Radio (SDR), after it was discovered that cheap USB TV tuners based on the Realtek RTL2832U chip could be dialed into frequencies well outside their advertised ranges. What was designed and sold as a simple device for watching TV on your computer could be turned into a radio capable of receiving anything between 64 MHz to 1700 MHz with open source software.

Now, anyone with about $20 USD to spare can tune into everything from police and fire transmissions to the International Space Station.

Tuner Hardware

Before you can start exploring the airwaves, you'll need a USB tuner supported by RTL-SDR, the software used to unlock the full potential of the Realtek RTL2832U chip. For best results, you'll also want to get one that uses the Elonics E4000 tuner, as that will give you the broadest frequency response. The RTL-SDR project maintains a short compatibility list which can help narrow things down a bit:

I purchased the Ezcap

I purchased the Ezcap EZTV668 from DealExtreme and can confirm it works perfectly with the latest RTL-SDR build. The only downside when ordering from DealExtreme is that shipping can often take a very long time; it was well over a month before the device arrived. You can try your luck on eBay, though it looks like the prices have gone up a bit since sellers are now realizing the market for these devices has just expanded considerably.

VID	PID	tuner	device name
0x0bda	0x2832	all of them	Generic RTL2832U (e.g. hama nano)
0x0bda	0x2838	E4000	ezcap USB 2.0 DVB-T/DAB/FM dongle
0x0ccd	0x00a9	FC0012	Terratec Cinergy T Stick Black (rev 1)
0x0ccd	0x00b3	FC0013	Terratec NOXON DAB/DAB+ USB dongle (rev 1)
0x0ccd	0x00d3	E4000	Terratec Cinergy T Stick RC (Rev.3)
0x0ccd	0x00e0	E4000	Terratec NOXON DAB/DAB+ USB dongle (rev 2)
0x185b	0x0620	E4000	Compro Videomate U620F
0x185b	0x0650	E4000	Compro Videomate U650F
0x1f4d	0xb803	FC0012	GTek T803
0x1f4d	0xc803	FC0012	Lifeview LV5TDeluxe
0x1b80	0xd3a4	FC0013	Twintech UT-40
0x1d19	0x1101	FC2580	Dexatek DK DVB-T Dongle (Logilink VG0002A)
0x1d19	0x1102	?	Dexatek DK DVB-T Dongle (MSI DigiVox? mini II V3.0)
0x1d19	0x1103	FC2580	Dexatek Technology Ltd. DK 5217 DVB-T Dongle
0x0458	0x707f	?	Genius TVGo DVB-T03 USB dongle (Ver. B)
0x1b80	0xd393	FC0012	GIGABYTE GT-U7300
0x1b80	0xd394	?	DIKOM USB-DVBT HD
0x1b80	0xd395	FC0012	Peak 102569AGPK
0x1b80	0xd39d	FC0012	SVEON STV20 DVB-T USB & FM

Compatibility list from RTL-SDR project

Antenna Upgrade

The proper selection, construction, and tuning of antennas is a very complex subject that is far outside of the scope of this document. That being said, the antenna that comes with your tuner is probably going to suck, **hard.**

In almost all cases, the pack-in antenna that comes with these tuners is just a little metal stick and a thin unshielded wire. With this antenna, the best you could realistically hope to receive is a strong local FM radio station (of course, to be fair, that's more or less all these things were designed to do in the first place). For anything else, you'll need to swap out the stock antenna with something a little less worthless.

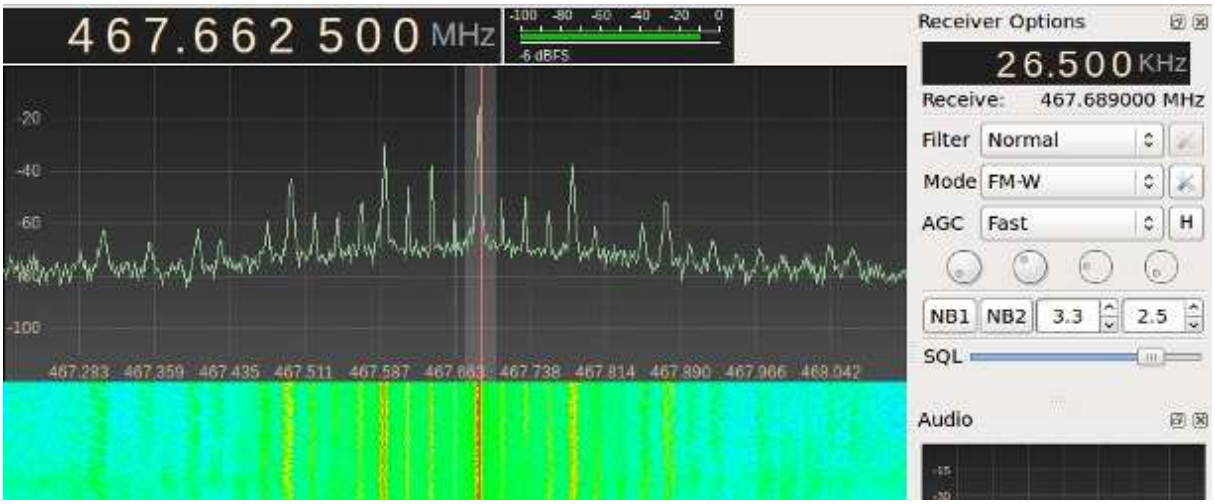
Luckily, it doesn't take a whole lot to improve on the situation. Even a cheap set of "rabbit ears" will work much better than the included antenna, and will give you good enough performance to start exploring the higher frequencies. Though you'll still need a more robust antenna for long range reception, and satellite work will require its own type of antennas if you hope to succeed.

As a starter antenna I've had good luck with the RadioShack "Budget TV Antenna" (Catalog #: 151874), though cheaper versions of the same concept can certainly be found online. With this type of antenna you will be able to pick up broadcasts from the police and fire department, NOAA weather reports, and local ham conversations pretty easily. On the subject of antennas, depending on which tuner you get, your device may need an adapter to use standard coax.

Your tuner may have a Belling-Lee connector, commonly (though incorrectly) called a "PAL connector". RadioShack sells an appropriate adapter as a "European TV Adapter" (Catalog #: 278261) for under $10. With this adapter installed, you'll be able to connect your RTL2832U tuner to essentially any standard antenna either directly (in the case of a TV antenna) or through an adapter (such as F to SO-239).

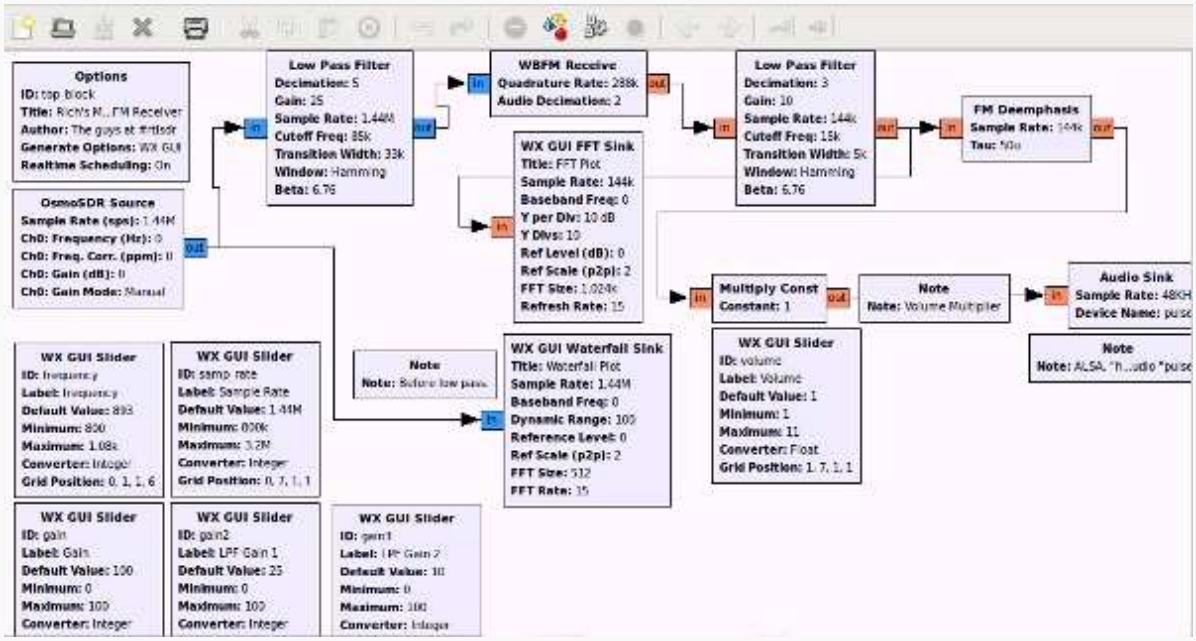
Getting Started With RTL-SDR

Tom Nardi



GNU Radio

GNU Radio is an incredible piece of software, allowing you to design and create software based radios. When these software radios are combined with a general purpose hardware receiver such as RTL-SDR, you can replace what would have once taken a whole room full of expensive radio equipment with a decently powerful desktop computer.



A "simple" software radio in GNU Radio

Unfortunately, this power comes at a steep price. GNU Radio is a very complex program, and anyone without a working knowledge of radio technology will have a lot of trouble getting anything done on their own with it. Luckily for those of us without ham licenses, there is more user-friendly software available for RTL-SDR, but they all require GNU Radio installed to function, so we'll need to go ahead and build

GNU Radio even if you never intend on using it directly.

The suggested method of installing the latest version of GNU Radio is to use the “build-gnuradio” script written by Marcus Leech. Running this script will download all of the latest sources, configure the system, and build all of the binaries. This may take quite some time depending on your hardware, so don’t expect to do anything with your computer for the next hour or so.

To start the GNU Radio build process, simply download the script and make it executable:

```
bash$ wget http://www.sbrac.org/files/build-gnuradio
Resolving www.sbrac.org... 174.142.32.20
Connecting to www.sbrac.org|174.142.32.20|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 29181 (28K) [text/plain]
Saving to: `build-gnuradio'
100%[======================================>] 29,181      --.-K/s   in 0.1s
2012-06-24 19:38:08 (255 KB/s) - `build-gnuradio' saved [29181/29181]
bash$ chmod +x ./build-gnuradio
bash$ ./build-gnuradio
This script will install Gnu Radio from current GIT sources
You will require Internet access from the computer on which this
script runs. You will also require SUDO access. You will require
approximately 500MB of free disk space to perform the build.

This script will, as a side-effect, remove any existing Gnu Radio
installation that was installed from your Linux distribution packages.
It must do this to prevent problems due to interference between
a linux-distribution-installed Gnu Radio/UHD and one installed from GIT source.

The whole process may take up to two hours to complete, depending on the
capabilities of your system.

Proceed?
```

This a very complete and well implemented installation script, it’s even smart enough to install required dependencies (there are quite a few) via your distribution’s package manager. Not only does the script build and install GNU Radio itself, but it will also download all of the RTL-SDR packages as well.

If you really want to build GNU Radio from source manually, you can do so by following along with the build guide from the GNU Radio Wiki. This is how I originally got GNU Radio installed on my machine before using “build-gnuradio”, though I would certainly use the script for future installations.

Note: Your distribution may already include GNU Radio in its repositories, but that version is likely too old, and certainly won’t include the RTL-SDR software with it. So even if you are on a distribution which offers precompiled GNU Radio packages, build it from source.

RTL-SDR Benchmark

Before going any farther, it would be a good idea to use our newly installed RTL-SDR software to test the hardware and make sure everything is operating as expected. RTL-SDR includes a handy tool to benchmark the capability of a connected RTL2832U tuner, which we can use to verify the hardware and software is communicating properly. To test your tuner and run a quick benchmark on it, simply do the following:

```
bash$ rtl_test -t
Found 1 device(s):
  0:  ezcap USB 2.0 DVB-T/DAB/FM dongle

Using device 0: ezcap USB 2.0 DVB-T/DAB/FM dongle
Found Elonics E4000 tuner
Benchmarking E4000 PLL...
[E4K] PLL not locked for 51000000 Hz!
[E4K] PLL not locked for 2229000000 Hz!
[E4K] PLL not locked for 1115000000 Hz!
[E4K] PLL not locked for 1245000000 Hz!
E4K range: 52 to 2228 MHz
E4K L-band gap: 1115 to 1245 MHz
```

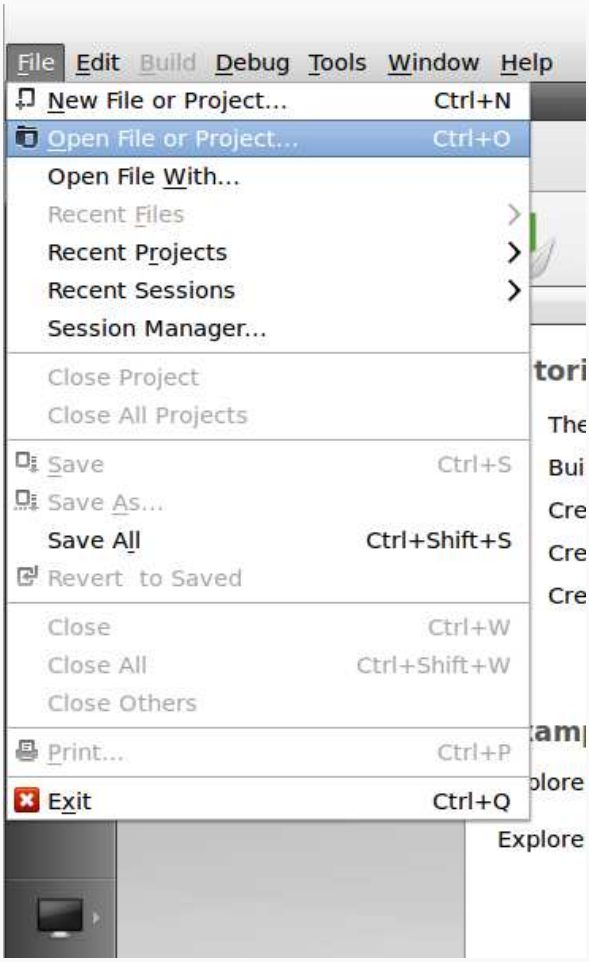
This output shows us that the Ezcap device is being detected properly, and that it contains the desired E4000 tuner hardware. In addition, it has run a quick test and verified its capability to tune between 52 and 2228 MHz (with the expected gap in the 1100-1250 MHz area).

If you **don't** see your device under rtl_test, something's wrong. Check you permissions on the device, and that your distributions kernel is up to date. If everything seems to be in order, you may want to head to ##rtlsdr on Freenode or the RTL-SDR subpage on reddit to ask for help.

Gqrx

Developed by the very talented Alexandru Csete (OZ9AEC), Gqrx has only recently gained RTL-SDR support, but is already one of the best options for new users. It features a very straightforward Qt GUI, and has enough functionality and features to keep a beginner interested while they get the hang of things. I've tried a few of the GUI tools for RTL-SDR, and I've had the best luck (by far) with Gqrx.

Gqrx is a QT4 program, and as such its installation is a bit different from what you might expect. To begin with, make sure you have QT4 installed via your distribution's package manager, and that have the "qtcreator" tool installed.



Opening Gqrx project in Qt Creator

Then download the source for Gqrx via its page on github (click "Download as tar.gz"), and extract it into a directory on your system. After the Gqrx source has been extracted, open Qt Creator, click "File", then "Open File or Project". In the following dialog, navigate to where you extracted the source and select the file named "gqrx.pro".

A dialog should pop up titled "Target Setup" with "Desktop" selected. You can simply click "Finished" on this dialog, after which Qt Creator will scan this file for a minute or so, which will be indicated by two loading bars on the right hand side.

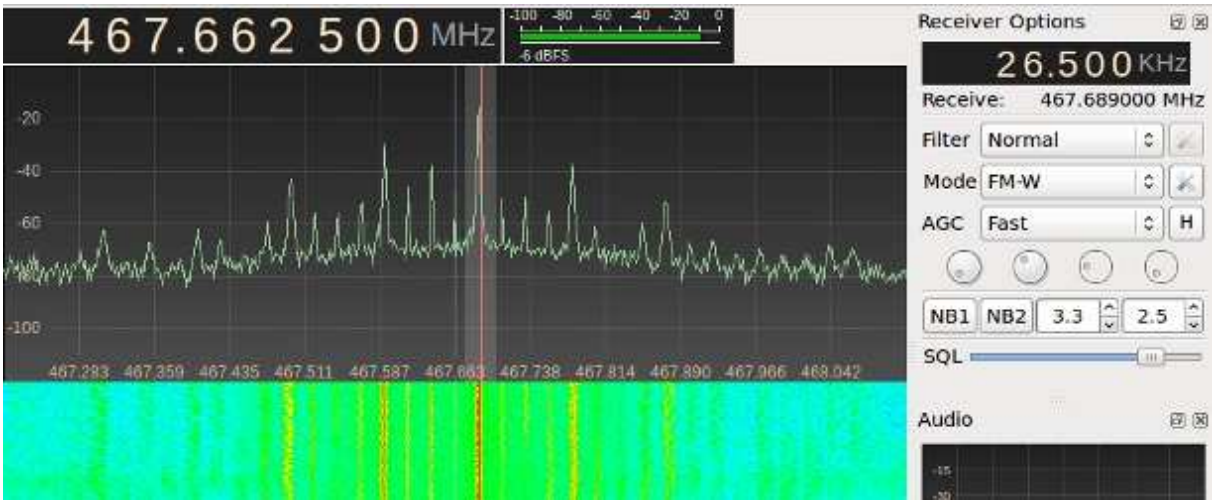
Once the source code has been scanned and Qt Creator is ready to proceed, click "Build" followed by "Build All". This process can take several minutes depending on processor speed, and you will likely see many warnings pop up in the bottom display, this doesn't seem to effect the build and it should complete successfully (I received over 500 warnings before Gqrx finished building).

Once it is complete a directory should have been

created in the same directory the Gqrx source was in called “gqrx-build-desktop”. In this directory you will find the “gqrx” binary which you can run to start the program.

Getting Started With RTL-SDR

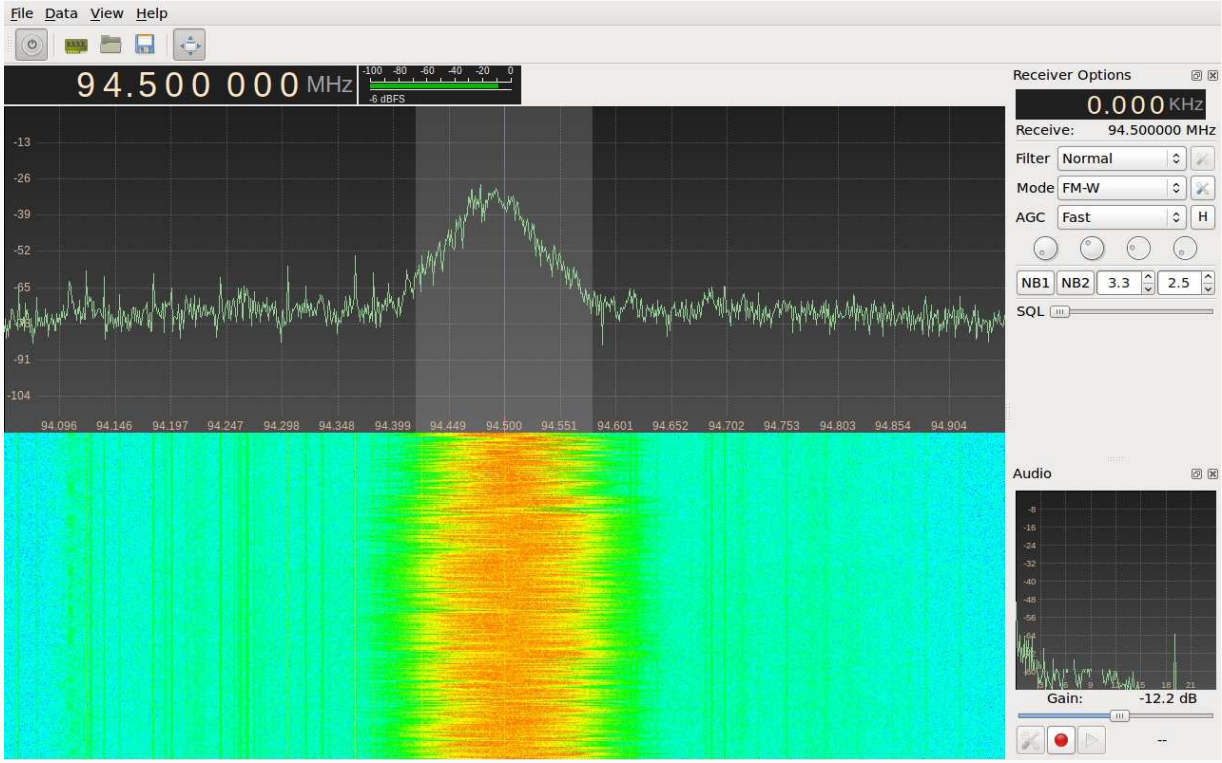
Tom Nardi



Tuning In

So we have our RTL2832U device, a makeshift antenna, and all the software required to do some real work with it. Now what? As a quick demonstration of what RTL-SDR is capable of, let's tune into a local FM radio station (click the image for full size):

Let's walk



Local FM station in Gqrx

through this image to get an idea of what's going on. The button on the top left of the toolbar is the "Power" button which starts and stops the receiver. Once you click the button, you should immediately

hear static and see the displays come to life.

The large white number at the top is the frequency we are tuned to, here, 94.5 MHz. The two large displays below it show there is clear activity at this frequency, shown by the peak in the top display and red lines in the lower “waterfall” display. Generally speaking, this is what we are looking for when exploring the radio spectrum. Walking through channels and looking for obvious signals via these two displays lets you find audio patterns within the sea of static.

On the upper right side of Gqrx, you can see the “Receiver Options” panel. This lets you adjust which mode the radio is in (such as AM, FM, CW, etc) and make changes to the automatic gain control (AGC). Most of the time you won’t need to change these settings very often, as most of the interesting stuff is in FM anyway. The most common change you will make here is switching between the two FM modes, “FM-W” and “FM-N” (broadcast radio stations need FM-W, most everything else will be FM-N).

Also take note of the squelch (SQL) slider, this allows you to cut out the audio unless it contains a sufficiently strong signal. The farther to the right you slide it, the stronger a signal will need to be to trigger the audio feed. This can be useful for things such as listening to ham radio broadcasts; by preventing loud bursts of static in between transmissions. Use caution with the squelch however, as turning it too high can mute weak signals completely.

The lower right side contains the “Audio” controls. The most important thing here is the “Gain” slider, which is essentially the volume. Turn this up to hear more of the signal, which may be required for weaker transmissions.

The Hunt

Knowing the basic controls of Gqrx, you should be able to tune the radio to different frequencies and begin exploring the local RF spectrum.

The previous image showed a very close and powerful transmitter, so accordingly the displays show a massive signal. For other broadcasts, it will not be nearly as obvious. Some signals may only appear as a line only a few pixels wide in the waterfall. When hunting for interesting signals, keep an eye out for *any* activity on the lower waterfall display.

To give you an idea of how this all comes together, the following video shows the basic principles of operation for Gqrx; first setting the frequency in the FM range and listening to some local broadcasts, then switching over to 930 MHz to listen in on some pager transmissions:

What Now?

From this point on, things are pretty much up to you. You now possess a software defined radio setup that would have cost hundreds of dollars just a year ago. Once you’ve become comfortable with tuning the radio and recognizing signals, the only limit to your explorations is your curiosity and patience.

Try looking at the frequencies your devices transmit on, see if you can find the signals for things like wireless weather stations or the key fob for your car. What kind of interesting things can you find?

If you’re looking for a challenge, get an antenna (or build one) that’s strong enough to receive signals from the International Space Station or even a geostationary satellite. The timing and alignment for communications at that range takes practice and dedication, and can offer a challenge to even accomplished radio operators.

With powerful software like GNU Radio and Gqrx backed up by affordable hardware such as the Realtek RTL2832U, the sky *literally* is the only limit to what you can do.