RTL-SDR: A Brief Introduction

Michael Khachatrian

August 2018

1 Introduction

"RTL-SDR is a very cheap $25 USB dongle that can be used as a computer based radio scanner for receiving live radio signals in your area (no internet required). Depending on the particular model it could receive frequencies from 500 kHz up to 1.75 GHz. Most software for the RTL-SDR is also community developed, and provided free of charge." The RTL-SDR is actually surprisingly accurate for such a cheap device, if interested further on its stability, take a look over here: https://www.rtl-sdr.com/wp-content/uploads/2015/03/rtl_R820T2.pdf

2 What operating system should I use it with?

I highly recommend with going for Windows or Linux, unfortunately there is very little support for SDR software on Mac OS.

3 What softwares/packages should I get to start out with the RTL SDR?

There are several softwares and packages one should have. First, look into a software that lets you visualize and record what the radio is picking up (like waterfalls) and tune into basic everyday AM/FM radio. These softwares are good for getting an overall grasp on the functionalities of the RTL-SDR and getting a feel for what one can do with it. On Windows, SDR# is the best software for this, while on Linux, GNURadio and GQRX are also popular options. If you are looking for something cross platform, go with CubicSDR, but functionality will be a bit decreased. Follow this link for setup instructions for the above mentioned softwares:

https://www.rtl-sdr.com/rtl-sdr-quick-start-guide/

Next, look into interesting plugins you can get for the software you downloaded to add more functions. For example, for SDR#, refer to this page for many neat plugins:

https://www.rtl-sdr.com/sdrsharp-plugins/

Some of my favorites were "Frequency Manager + Scanner and Scanner Metrics and Frequency Entry Package" a.k.a. "fmsuite", which has a feature that lets you scan a very broad range of frequencies and log any signals that exceeded a certain power threshold set by yourself. Other plugins I like are the "Level Meter Plugin" and "Heatmap Generator Plugin". Here is a plugin that changes the interface of SDR# to put it in dark mode, which conserves battery and takes out some of the inefficient space usage of the default SDR# layout:

https://www.rtl-sdr.com/forum/viewtopic.php?f=5&t=2123

Here are some plugins for GNURadio:

http://cgran.org

And some for GQRX:

http://gqrx.dk/doc/external-applications

Overall, SDR# is definitely best for customization through the multitude of plugins it offers, but it being restricted to Windows can be a disadvantage.

After downloading one of the above softwares, look into calibration. Here is a useful YouTube tutorial:

https://www.youtube.com/watch?v=gFXMbr1dgng (for SDR#). I just calibrated based on the signals emitted by NOAA stations since they are stable and consistent. Use this site to find which station is near you and calibrate based on that

http://www.nws.noaa.gov/nwr/coverage/station_listing.html

This is the closest one to Pasadena:

http://www.nws.noaa.gov/nwr/coverage/site2.php?State=CA&Site=KWO37

Here's a tutorial that uses GSM signals emitted by cell phone towers to calibrate using GQRX. GSM signals are also very reliable:

https://www.youtube.com/watch?v=mJIU04PHKTo

These are some really quick ways to calibrate, later I outline a more accurate method that uses a separate software entirely, and is my preferred method.

Tip: Put your RTL-SDR to use for 20 minutes before collecting any data with it/doing something specific with a software. The RTL-SDR has heat sensitive components that affect its stability and it is best to let it heat up through use (such as listening to music on your favorite station).

4 Other software/package recommendations

While the above softwares are neat for visualization and a few other things, more specialized software is needed for collecting useful data. Note: The links I provide in the descriptions are for Linux. For Windows users, I have uploaded Windows compatible versions of these softwares and uploaded them into a Google Drive folder here: https://drive.google.com/drive/folders/1GuBn1i8eA3l9-xqH13B7egS3–AUVDyo?usp=sharing

1. **Rtl_power**: This is the most useful software. Check this site for very useful documentation of the multitude of features it offers: http://kmkeen.com/rtl-power/. It lets you do extremely wide band surveys and logs data into an

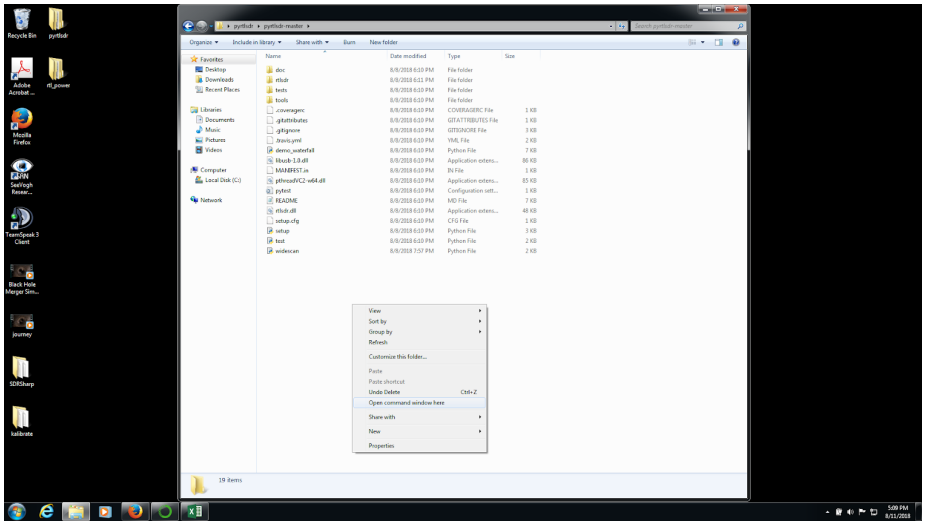
easily analyzable .csv file, check in the "Rendering" section of the documentation on the linked site. The commands should work on Windows too.

2. **pyrtlsdr**: A very neat pythonic interface for the RTL-SDR: https://github.com/roger-/pyrtlsdr . Among its useful features are easily and quickly plotting broad ranges of frequencies and the power levels at said frequencies, and displaying a radio waterfall in matplotlib. Sadly documentation of its features is a bit limited but looking around the GitHub and the help resources I will link later helps. The commands work the same way on Windows.

3. **Kalibrate**: My preferred way of finding the ppm to calibrate the RTL-SDR: https://github.com/steve-m/kalibrate-rtl It uses signals from GSM towers to calibrate. Here is a tutorial for using it on Windows: http://rtl-sdr.sceners.org/?p=193 And here is one for installing it on Linux should difficulties arise: https://github.com/steve-m/kalibrate-rtl/wiki/How-to-install

5 A Tip For Running Programs in Windows

On Windows, launch the command line directly from folders containing the necessary drivers/packages by using Shift+Right Click and selecting the option that lets you launch a command line from there. For example, if I want to use pyrtlsdr, here is how I would do it:



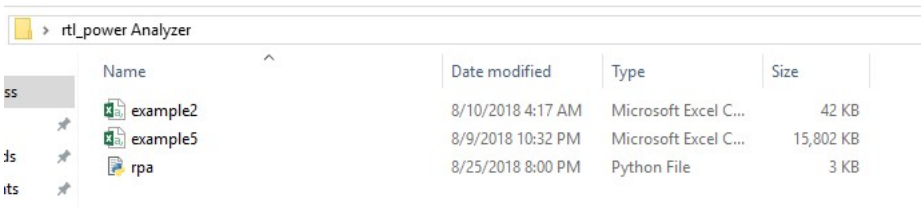
6 rtl_power Analyzer

rtl_power Analyzer is a python script I wrote that lets one extract certain information from the .csv files generated by rtl_power. It uses Python 3.6 and its dependencies are:

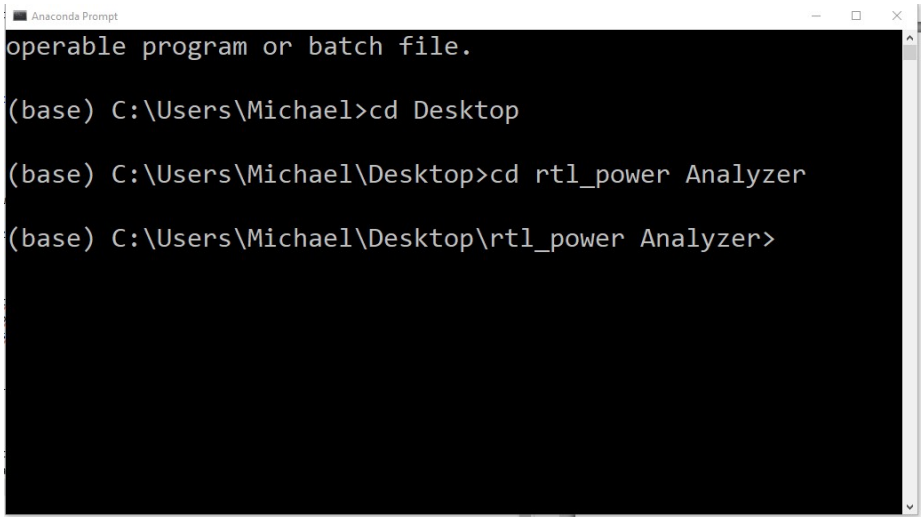
1. numpy
2. matplotlib
3. itertools
4. argparse

These all come with Anaconda, which I highly suggest using to run this script. The way to run the script is:

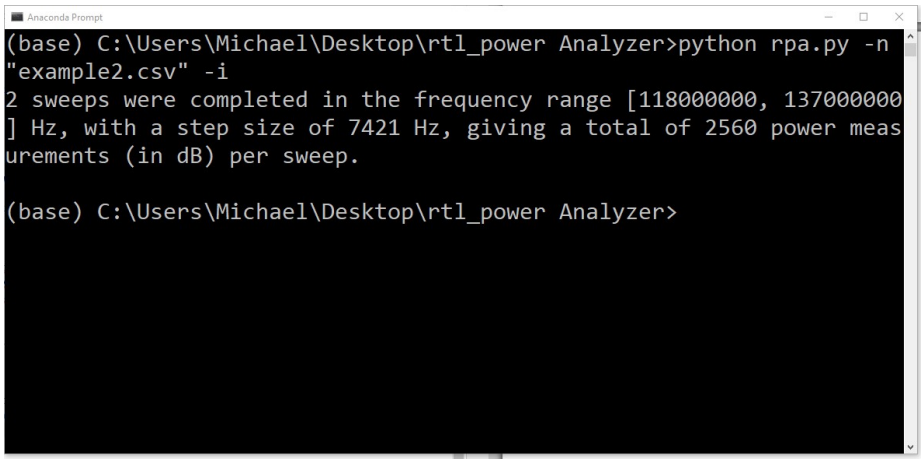
1. Create a new folder with the rpa.py file and whatveer .csv file you want to analyze in it.



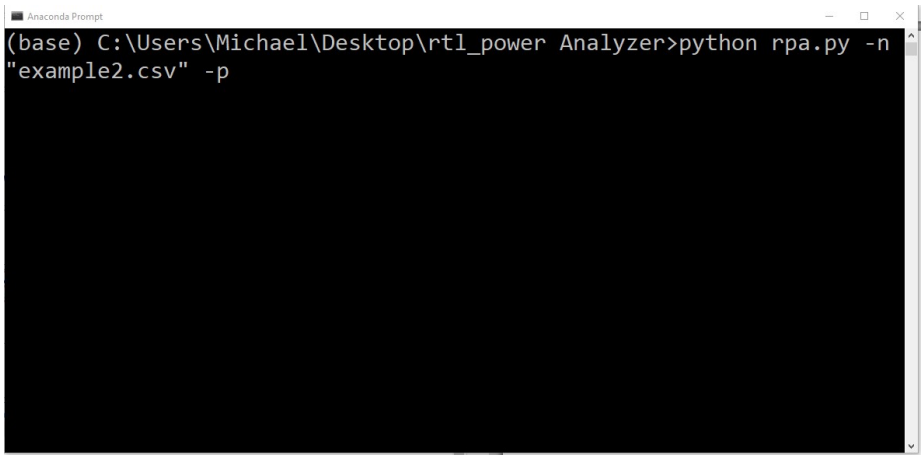
2. Start a command line prompt (preferably an Anaconda prompt), and navigate to your folder.



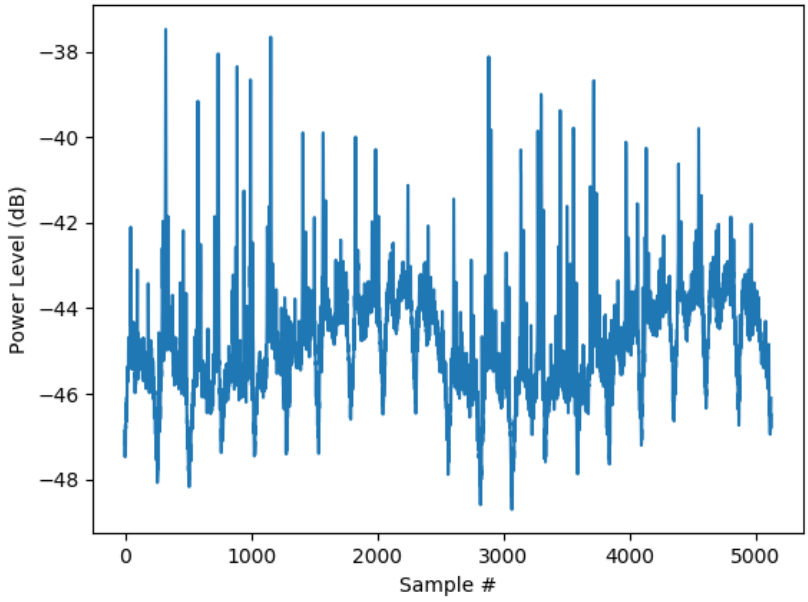
3. Run the rpa.py file through python, specify a file name, and use one of the three commands of the script. -n is for specifying a file name (make sure to put the file name in quotes), -i gives information about the scan completed, -p generates a plot of the power levels, and -v followed by -s i and -f j, where i is the ith sweep and j is the jth frequency, provides the power level (in dB) during that specific sweep and that specific frequency. For example -v -s 34 -f 27 will give the power level during the 34th sweep and 27th frequency. Screen shots are included below. For example, this is how to run the -i command:



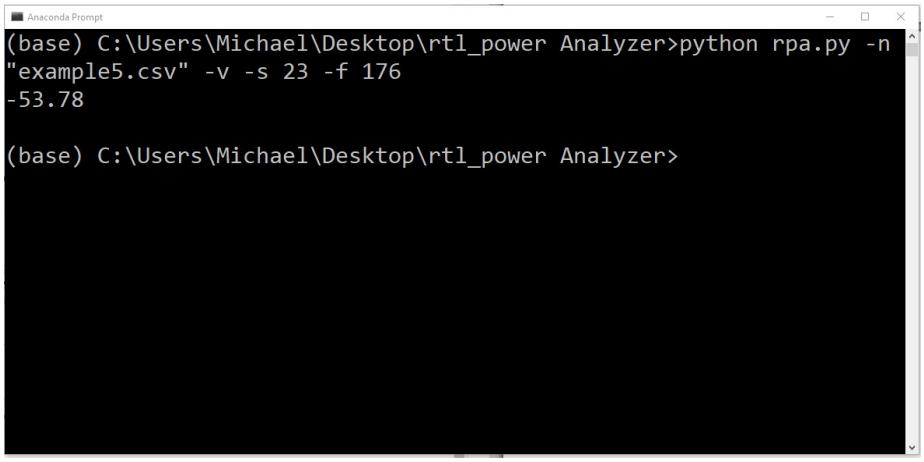
Here is an example of the -p command:



Which generates:



And finally, the -v command:



As of now, I'm still working on adding more features to the program and will upload it to GitHub in the future with more documentation. For now, here is a link to a Google Drive folder that contains the script and two example .csv files to test it on:
https://drive.google.com/drive/folders/1_WBTP5AKSSRHI0wKo2GhTGEKwKT0fbJz?usp=sharing

7 Additional Resources for Questions

If questions come up on anything related to the RTL-SDR, post a thread in these two forums:

https://www.rtl-sdr.com/forum/

https://www.reddit.com/r/RTLSDR/

Both have an active user base. Other places to ask for help are posting on GitHubs of softwares or emailing the software creators directly. Contact me at michaelkhachatrian@gmail.com and I can try my best to help too.