Lab 3 - Exploring the RF Spectrum

ECE531 – Software Defined Radio

Spring 2024

Daniel Gallagher, danielgallagher@arizona.edu
Department of Electrical & Computer Engineering
University of Arizona, Tucson AZ 85721

1 Overview

The radio frequency (RF) spectrum is a portion of the electromagnetic spectrum that ranges between 3kHz and 300GHz. We use the RF spectrum extensively for communications, navigation, astronomy, radar, and more. The Federal Communications Commission (FCC) regulates the usage of the RF spectrum in the United States. Figure 1 shows a map of the frequency allocations in the United States. You may zoom into this map for detail, or better yet download your own printable copy [1]. Further details on the FCC frequency allocation is available at [2].

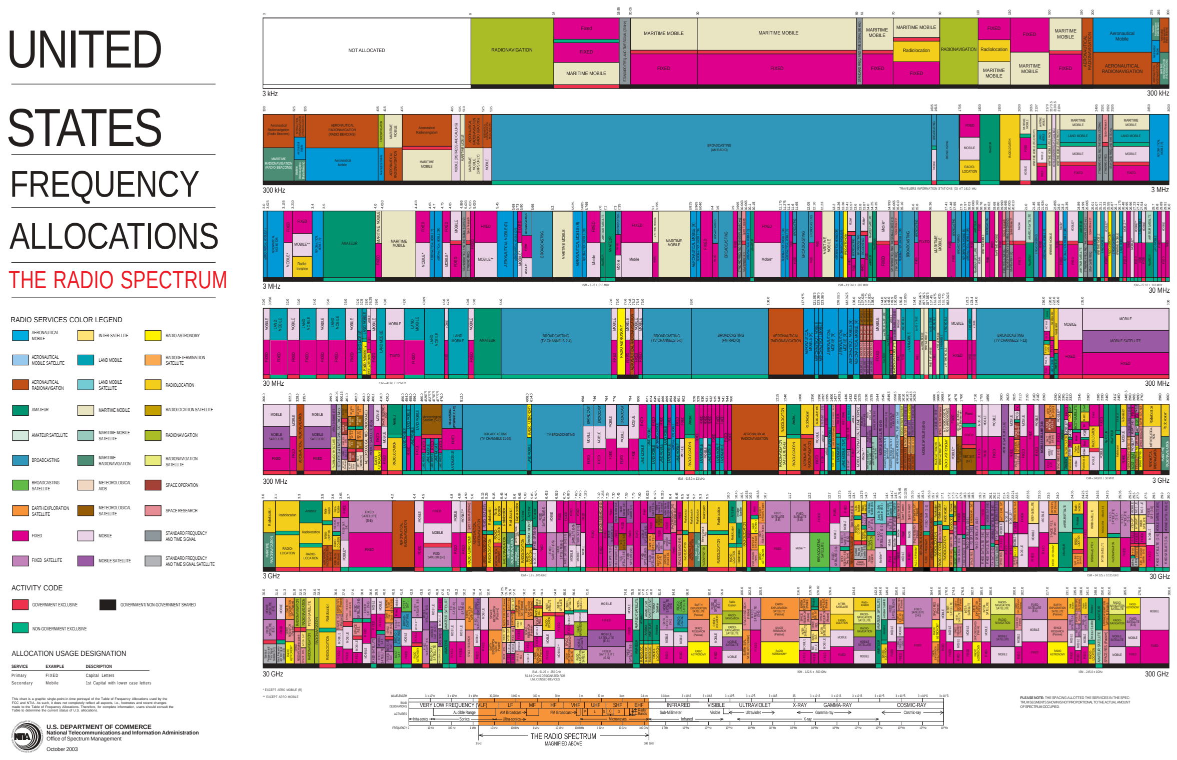


Figure 1: FCC frequency band allocations [1]

2 Software

As an outcome of Lab 2, you should now have a software configuration that is compatible with the PlutoSDR. The ECE531 virtual machine (VM) image also has a complete Linux environment that includes GNU Radio with gr-iio support. The VM and local MATLAB install can be used for Lab 3 and subsequent labs this semester.

2.1 Fosphor

Fosphor is an open-source, GPU-accelerated FFT and Waterfall display tool created by Sylvain Munaut. gr-fosphor provides a GNU Radio block for RTSA-like spectrum visualization using OpenCL and OpenGL acceleration. Fosphor's high frame rate, great for visualizing fast-changing signals.

The instant-gnuradio VM image includes a CPU accelerated OpenCL build of fosphor [3]. GPU acceleration, or even FPGA acceleration [4], can improve performance significantly. The block has been used successfully on Linux, OSX and Windows; Linux is recommended. Proper configuration of OpenCL can prove challenging.

- Fosphor details and configuration instructions: `https://osmocom.org/projects/sdr/wiki/fosphor`
- gr-fosphor source is available at: `http://git.osmocom.org/gr-fosphor`

The VM image includes gr-osmosdr [5], which includes the utility `osmocom_fft`. The instant-gnuradio VM is configured so osmocom should recognize the PlutoSDR without additional arguments required. Call `osmocom_fft` with -F switch to enable the fosphor display; as shown below. You may also call with the `--help` argument to see additional functionality.

```
$ osmocom_fft -F
```

2.2 Other Software Tools

These tools below may be of interest for some, but are not required to complete this lab.

2.2.1 SDR#

SDR#, pronounced SDR Sharp, is a windows based SDR software tool available for download at: `https://airspy.com/download/`. (Note: Currently for PlutoSDR support, you must use the unskinned SDR# build). SDR# has a plugin to available to add PlutoSDR compatibility at: `https://github.com/Manawyrm/sdrsharp-plutosdr/releases`. Instructions on configuring SDR# with the PlutoSDR can be found at `https://www.rtl-sdr.com/plutosdr-quickstart-guide/`.

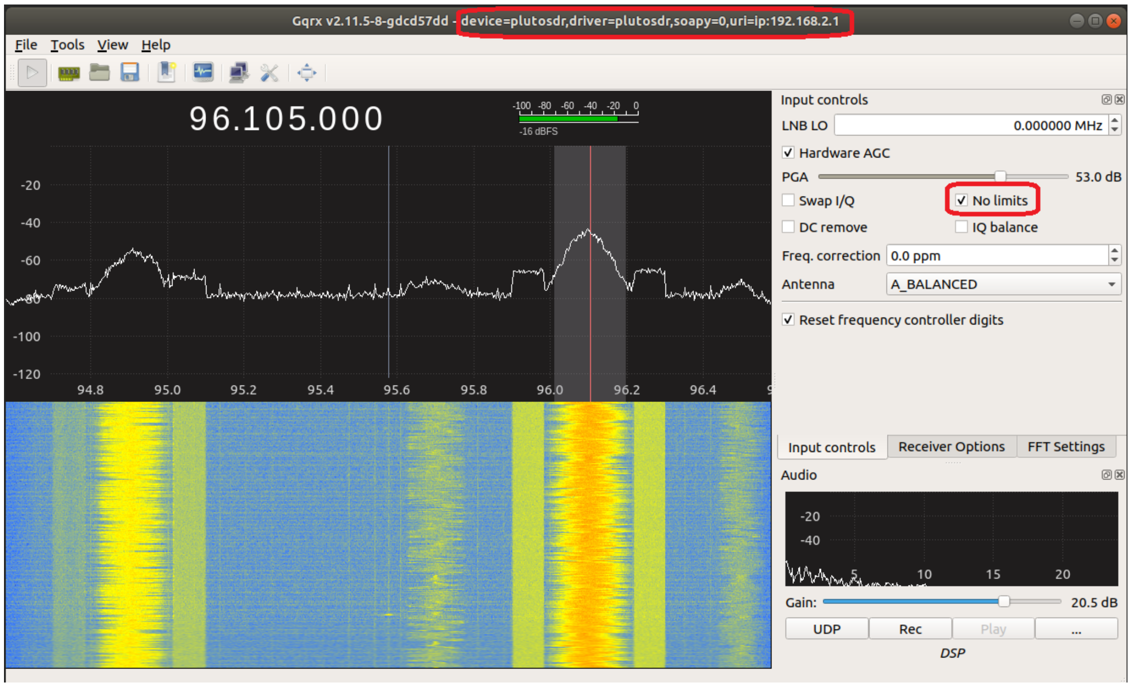
2.2.2 GQRX

GQRX is a SDR software package derived from GNU Radio and available at http://gqrx.dk/. As with SDR#, GQRX allows visualization, tuning, and demodulation to audio of signals within one application. Instructions for configuring GQRX with the PlutoSDR can be found at: https://www.rtl-sdr.com/adalm-pluto-sdr-hack-tune-70-mhz-to-6-ghz-and-gqrx-install/

To use GQRX with the Pluto, choose “Other” under devices and enter “`plutosdr=0`” as the device string. The device string can also be set as listed below and shown in Figure 2. Be sure to check “No Limits” under input controls to allow tuning on the full PlutoSDR frequency range.

Device string:

```
device=plutosdr,driver=plutosdr,soapy=0,uri=ip:192.168.2.1
```



Gqrx v2.11.5-8-gdcd57dd - device=plutosdr,driver=plutosdr,soapy=0,uri=ip:192.168.2.1

96.105.000

Input controls
LNB LO: 0.000000 MHz
✓ Hardware AGC
PGA: 53.0 dB
☐ Swap I/Q ✓ No limits
☐ DC remove ☐ IQ balance
Freq. correction: 0.0 ppm
Antenna: A_BALANCED
✓ Reset frequency controller digits

Audio
Gain: 20.5 dB

Figure 2: GQRX with PlutoSDR hardware on instant-gnuradio Virtual Machine

3 Spectrum Exploration

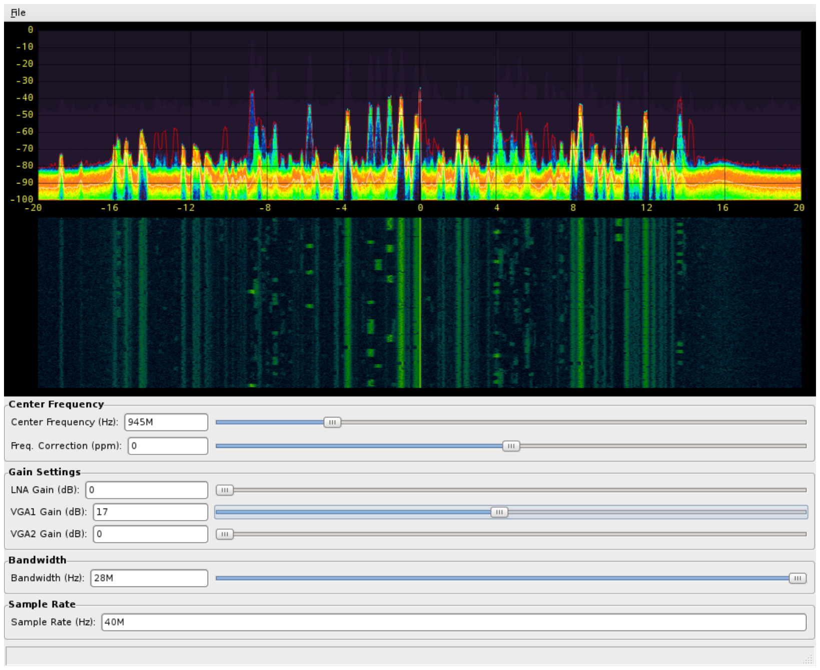
Using software defined radios to receive over-the-air signals and exploring the RF spectrum can be an interesting and fun activity. Section 2 mentions a few of the tools that make this possible on SDR.

3.1 Signal Identification

The sigidwiki.com signal identification guide is intended to help identify radio signals through example sounds and waterfall images. Most cataloged signals at sigidwiki are received and recorded using a software defined radio. This can be a useful resource to identify signals when exploring the RF spectrum.

The following references can also help determine what signals are available in your location:

- `http://reboot.fcc.gov/reform/systems/spectrum-dashboard`
- `http://www.radioreference.com/`



Center Frequency
Center Frequency (Hz): 945M
Freq. Correction (ppm): 0

Gain Settings
LNA Gain (dB): 0
VGA1 Gain (dB): 17
VGA2 Gain (dB): 0

Bandwidth
Bandwidth (Hz): 28M

Sample Rate
Sample Rate (Hz): 40M

Figure 3: Spectrum Exploration using fosphor

1. Using fosphor, or other spectrum waterfall plot, identify at least four different modulation techniques using received over-the-air transmissions. (i.e. AM, FM, LTE (OFDM), GSM, etc).
2. Describe the frequency spectrum and time characteristics observed which reveal the modulation.

3.2 IEEE 802.11 Wireless Local Area Networks

Another interesting experiment is using the PlutoSDR to observe the spectrum of wireless local area networks within the vicinity (WLANs). Most high population density areas employs numerous wireless communication networks for a variety of applications, such as the university wireless network. Consequently, you can use your PlutoSDR experimentation platform to plot their magnitude spectrum. IEEE 802.11 [6] is one type of WLAN standard that possesses a list of carrier frequencies for a collection of Wi-Fi channels. For example, The IEEE 802.11 standard defines Channel 1 of the 2.4 GHz band to be centered at 2.412 GHz as seen in Figure 4.

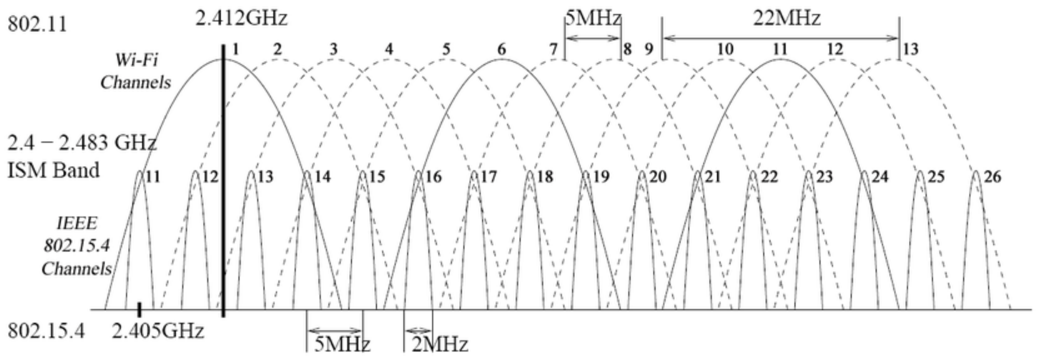


Figure 4: IEEE 802.11 Wi-Fi channels

1. Specify the *CenterFrequency* parameter of `sdrrx` with this carrier frequency, use the `FFTs` or `dsp.SpectrumAnalyzer` objects to plot their magnitude spectrums.
2. Since your AD9364 transceiver supports the 6 GHz band as well, also use the `dsp.SpectrumAnalyzer` to plot spectrum in the 5 GHz band.

You might want to turn on "Spectral Averaging" or "PlotMaxHoldTrace" and adjust the amplitude, since the peaks of these wireless signals could be rather low. It might also be useful to change the *BasebandSampleRate* parameter of `sdrrx` to adjust the frequency resolution for a better graph of the spectrum.

4 Collecting Spectral Data

Utilizing captured data from a file source can make testing much more repeatable when debugging issues during algorithm development. However, keep in mind that recording complex samples at high sample rates can result in large files very quickly.

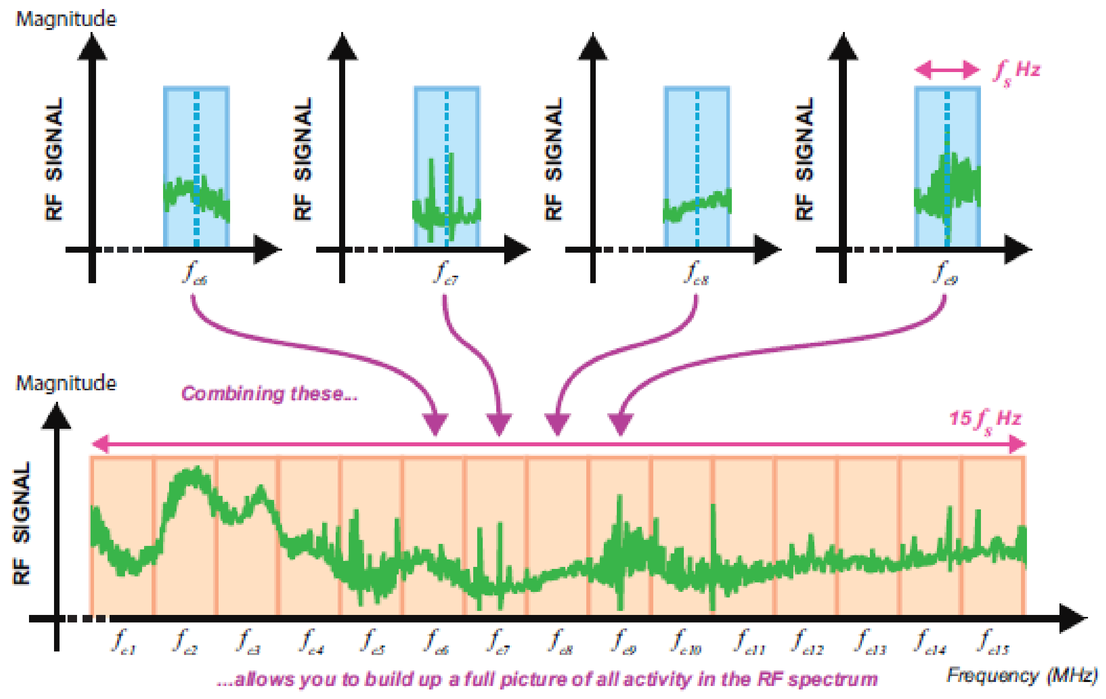
In MATLAB, data samples can be recorded using the `comm.BasebandFileWriter` system object and read using the `comm.BasebandFileReader` system object. For more information on these, see §5.2 in the textbook.

In GNU Radio, the File Sink is available to record complex samples. These samples can then be read using the File Source block, Inspectrum, or Octave and MATLAB. Inspectrum is a tool for analysing captured signals, primarily from software-defined radio receivers [7].

Source files for reading GNU Radio sample files in MATLAB can be found at `https://github.com/gnuradio/gnuradio/tree/master/gr-utils/octave`

4.1 Sweeping the Spectrum: Receiving from 60MHz to 6GHz

The PlutoSDR is limited to a maximum bandwidth of 56MHz, but is capable of tuning from 70MHz to 6GHz (with firmware modification). By performing a frequency sweep with a single SDR, we are able to stitch together multiple sampling intervals to get a picture of the wider spectrum. The local oscillator can step by F_s to get each next adjacent band, as shown in Figure 5.



Magnitude
RF SIGNAL
f_s Hz
f_{c6}
f_{c7}
f_{c8}
f_{c9}
Combining these...
15 f_s Hz
f_{c1} f_{c2} f_{c3} f_{c4} f_{c5} f_{c6} f_{c7} f_{c8} f_{c9} f_{c10} f_{c11} f_{c12} f_{c13} f_{c14} f_{c15}
...allows you to build up a full picture of all activity in the RF spectrum
Frequency (MHz)

Figure 5: Combining individual "tunes" to sweep the spectrum allows full RF spectrum visualization.

1. Write a script to capture the entire tunable bandwidth by stitching together multiple intermediate frequencies at different tuned LO
2. Label some known signals on the resulting spectrum plot

Note: You may wish to use a lower sampling frequency and more LO steps due to the amplitude variation of the Pluto RF front end.

5 Understanding Your Wireless Devices

RF transmitting devices are typically registered with the FCC for compliance, including emissions test results. A devices FCC ID can be used to search the FCC database to determine some details such as manufacturer, frequency and modulation methods. Internet sites, such as http://fcc.io/, make searching the FCC database very easy.

SDRs and RF scanners make it very easy to record and reply fixed code sequences. For security, devices such as garage door openers and keyless entry systems utilize rolling codes to prevent replay attacks, where an eavesdropper records the transmission and replays it at a later time to cause the receiver to 'unlock'.

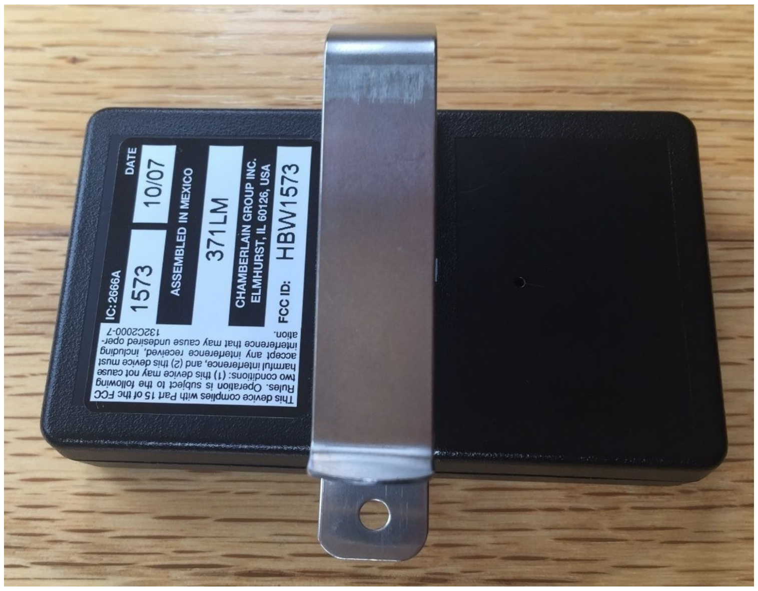


Figure 6: This garage door opener FCC ID label can be used with http://fcc.io/ to find transmit frequency.

1. Find a garage door opener remote control, remote keyfob, or similar RF transmitting device.
2. Look up the device using its FCC ID. What is the device's operating frequency?
3. Use the PlutoSDR to find the signal produced by the device. At what frequency did you find the device?
4. Look at the FFT plot of the signal. What modulation is the device using?
5. What technique could you use to demodulate and convert to binary data?
6. Visually decode the transmitted bits, does the same sequence of bits repeat while you hold down the button? Does the sequence change if you release the button and press it again?

6 Lab Report Preparation & Submission Instructions

Include all your answers, results, and source code in a laboratory report formatted as follows:

- Cover page: includes course number, laboratory title, name, submission date.
- Suggested: Table of contents, list of tables, list of figures.
- Commentary on designed implementations, responses to laboratory questions, captured outputs, and explanation of observations.
- Conclusions to the overall lab that discuss meaningful lessons-learned and other take-aways from the assignment.
- Upload source files with report submission. You may also list select code source in your report appendix. Note: Python files autogenerated from GNURadio do not need to be uploaded in additon to .grc files.

Remember to write your laboratory report in a descriptive approach, explaining your experience and observations in such a way that it provides the reader with some insight as to what you have accomplished. Furthermore, please include images and outputs wherever possible in your laboratory report document.

References

[1] US Dept. of Commerce, "United states frequency allocations: The radio spectrum." [Online]. Available: https://transition.fcc.gov/oet/spectrum/table/fcctable.pdf

[2] Federal Communications Commission, "FCC online table of frequency allocations." [Online]. Available: https://www.ntia.doc.gov/files/ntia/publications/2003-allochrt.pdf

[3] B. Bloessl, "Github: Instant gnu radio." [Online]. Available: https://github.com/bastibl/instant-gnuradio

[4] S. Munaut, "RFNoC: fosphor – How to apply RFNoC to RTSA display acceleration." [Online]. Available: https://archive.fosdem.org/2015/schedule/event/rfnocfosphor/

[5] gr osmocom, "osmocom gnu radio blocks." [Online]. Available: https://osmocom.org/projects/gr-osmosdr/wiki

[6] IEEE 802.11 Working Group, "IEEE 802.11: The working group setting the standards for wireless LANs," [Online]: http://www.ieee802.org/11/.

[7] M. Walters, "Inspectrum: Offline radio signal analyser." November. [Online]. Available: https://github.com/miek/inspectrum