Lab 3 - Exploring the RF Spectrum

ECE531 – Software Defined Radio

Spring 2022

Rahel Mizrahi, rmizrahi@email.arizona.edu

Due Date: March 7, 2022

1 Introduction: Spectrum Exploration

This lab was all about exploring the RF spectrum using software defined radio. To that end, we viewed the bands in which the signals we use every day reside and analyzed several various modulation techniques. In the first section, we explore different signals and identify distinct modulation techniques. In the second section, we view the entire spectrum the Pluto is able to tune to. In the third section, we analyze and (almost) decode the codes that RF transmitting devices use to unlock cars.

1.1 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/

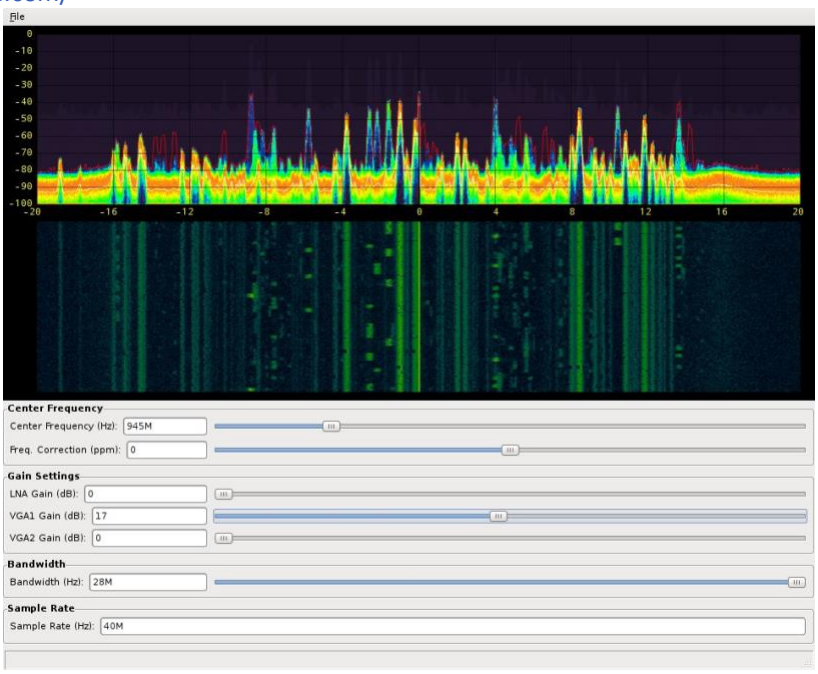


Figure 1. Spectrum Exploration using fosphor

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).

1. Describe the frequency spectrum and time characteristics observed which reveal the modulation.

1.2 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.

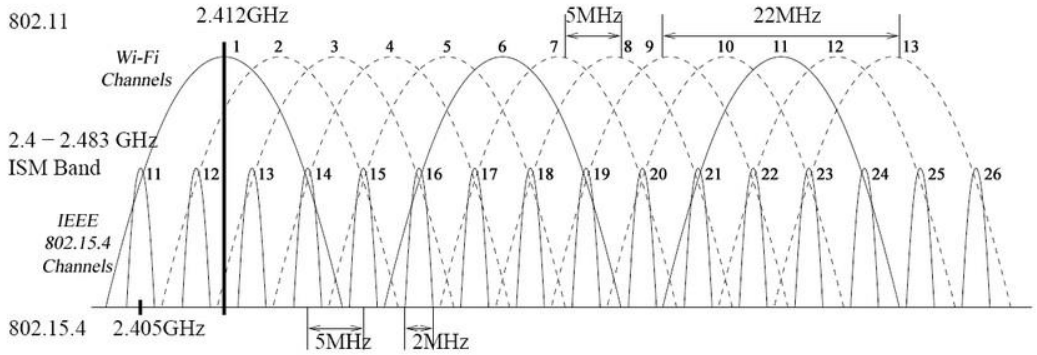
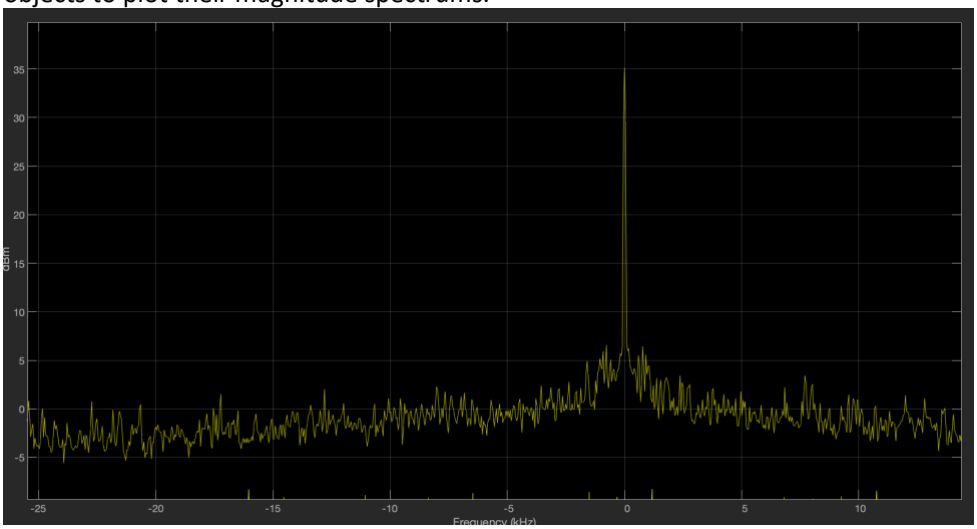


Figure 2. IEEE 802.11 Wi-Fi channels

Specify the *CenterFrequency* parameter of sdrrx with this carrier frequency, use the FFTs or dsp.SpectrumAnalyzer objects to plot their magnitude spectrums.



1. Since your AD9364 transceiver supports the 6GHz 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.

2 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

2.1 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.

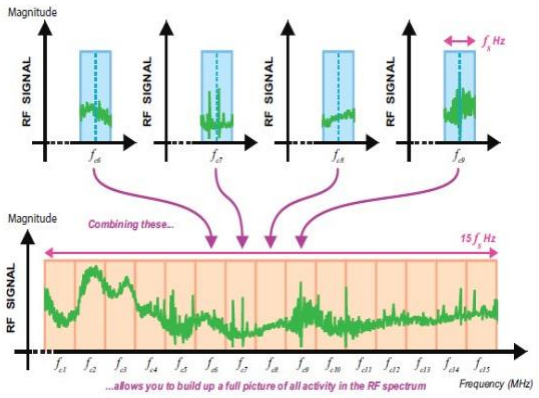


Figure 3. Combining individual "tunes" to sweep the spectrum allows full RF spectrum visualization

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.

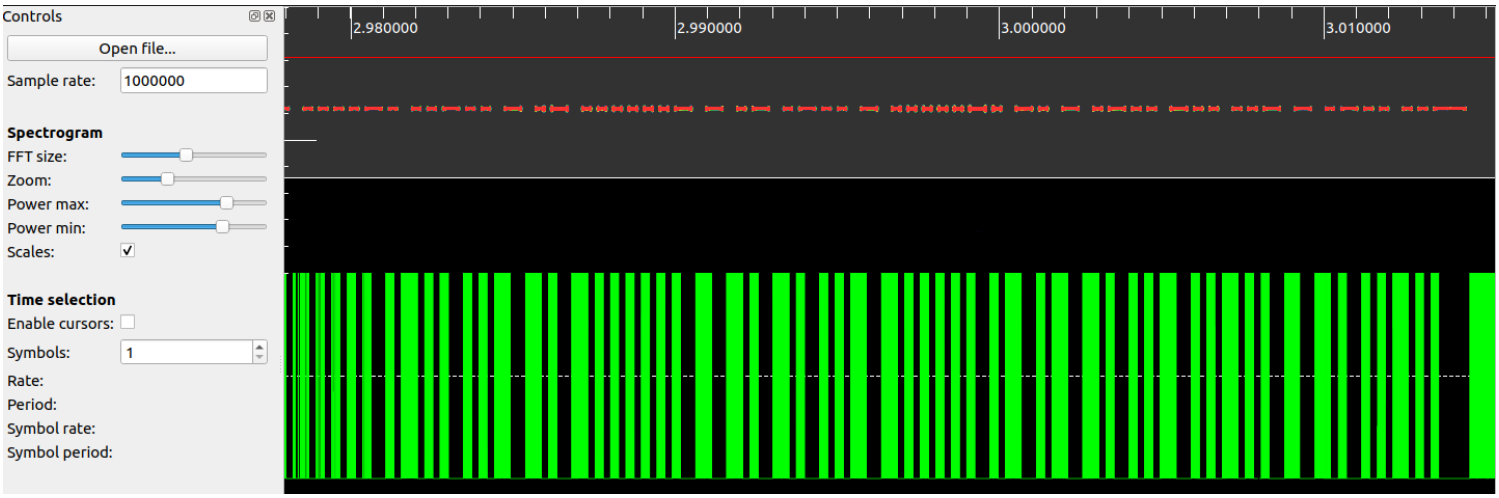
3 Understanding Your Wireless Devices

RF transmitting devices are typically registered with the FCC for compliance, including emissions test results. A device's 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 4. 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.
   My ford fusion hybrid's FCC ID is M3N-A2C931423
2. Look up the device using its FCC ID. What is the device's operating frequency? The device's operating frequency is 314.95 MHz.
3. Use the PlutoSDR to find the signal produced by the device. At what frequency did you find the device? The device was found at 314.95 MHz, and there was another peak nearby this frequency,
4. Look at the FFT plot of the signal. What modulation is the device using? In Inspectrum, I played with the he waterfall plot, at different intervals, shows bursts of different frequencies. This tells me the device is using FSK as its modulation technique. I confirmed this with the device's user's manual.



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?