Lab 2 - Getting Started on PlutoSDR



**ECE531 – Software Defined Radio**

**Spring 2022**

Daniel Gallagher, danielgallagher@arizona.edu

Department of Electrical & Computer Engineering

University of Arizona, Tucson AZ 85721

*Due Date: February 21, 2021*

1. **Radio Setup and Environmental Noise Observations 10**
   1. Signal Loopback . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
   2. Measurements and the Radio . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
2. **Lab Report Preparation & Submission Instructions 15**

# Radio Setup and Environmental Noise Observations

In these laboratory experiments an Analog Devices ADALM-PLUTO SDR with AD9363 transceiver will be used. This SDR in the most basic sense is a System-on-Chip (SoC) with an attached RF module providing complex baseband data. The MATLAB interface utilizes the libiio kernel driver to talk with the SDR through a MATLAB class called iio\_sys\_obj\_matlab. The two system objects provided in the hardware support package for PlutoSDR are:

* comm.SDRRxPluto: PlutoSDR Receiver System object
* comm.SDRTxPluto: PlutoSDR Transmitter System object

These objects are typically constructed through the sdrrx or sdrtx function calls as in:

|  |
| --- |
| rx = sdrrx('Pluto') tx = sdrtx('Pluto') |

1

2

However, these objects can also be directly instantiated directly. The resulting object of sdrrx either way will have the following basic properties, which will be directly printed to the terminal when not using the semicolon as:

|  |
| --- |
| rx = sdrrx('Pluto') rx = comm.SDRRxPluto with properties:  DeviceName: 'Pluto'  RadioID: 'usb:0'  CenterFrequency: 2.4000e+09  GainSource: 'AGC Slow Attack'  ChannelMapping: 1  BasebandSampleRate: 1000000  OutputDataType: 'int16'  SamplesPerFrame: 3660  ShowAdvancedProperties: false |

1

2

3

4

5

6

7

8

9

10

11

12

The transmitter System object comm.SDRTxPluto has near identical properties except for

## Signal Loopback

In the signal loop back experiment, we transmit the same signal that we receive by connecting the TX and RX with a coaxial SMA cable.

### MATLAB Loopback

We first perform signal loopback via matlab. A sinusoidal tone is output by the transmitter and is simultaneously received at the receiver. To perform this, the provided loopback.m script was used. This script collects multiple buffers of data, which can be necessary since there is an unknown startup lag for the transmitter and the desired signal can be missed. This script should generate a plot similar to the one shown in Figure 7.

The script generated the below signals. Notice the choppiness. This choppiness arises because the parameter *samples per frame* is not a multiple of , where is the period of the signal x(t) and is the sampling interval. When samples per frame is not a multiple of this ratio, the first sample is sampled at a different location than the ending sample, which gives rise to the choppiness observed in the dark graphs below. The lighter graph shows the output at the receiver when samples per frame is .

Chart, histogram

Description automatically generated

Figure 1. Looped back sinusoid observed from the Pluto receiver

Gain = -30 dB, varied amplitudes.

|  |  |
| --- | --- |
| Figure 2. Rx gain = 30 dB, amplitude = 1, gain mode: manual | Figure 3. Rx gain = 30dB, amp = 0.5 |

ACG Gain Modes

|  |  |
| --- | --- |
| Figure 4. tx gain = -20 dB, amplitude = 1, ACG fast attack | A screenshot of a computer  Description automatically generated with medium confidence  Figure 5. tx gain = -30 dB, amplitude = 1, ACG fast attack |

Starting from an Rx gain of -30 dB, the sinusoid appeared as a square wave from the point of view of the receiver, as shown below. This occurs because when multiplication of -30 db corresponds to amplification of the input signal by a factor of 10. Clearly, this shifts the signal up farther than the scope can represent. amplitude is too high

Chart

Description automatically generated

Figure 6. gain = -30 dB, amplitude = 1, gain mode = manual, rx gain = 35 dB

What gain value does this occur? this begins to occur at an rx gain of 30 dB.

### GNU Radio Loopback

Now that MATLAB has been used for a loopback test, lets attempt the same test with GNU Radio to verify both tools are functioning properly.

To perform this test, we can utilize the provided PlutoTestSine.grc flowgraph in GNU Radio Companion. This flowgraph is configured to simultaneously transmit and receive in the 2.4GHz band with a sample rate of 2.084MHz. The “QT GUI Range” variable source frequency and RX gain at run-time.

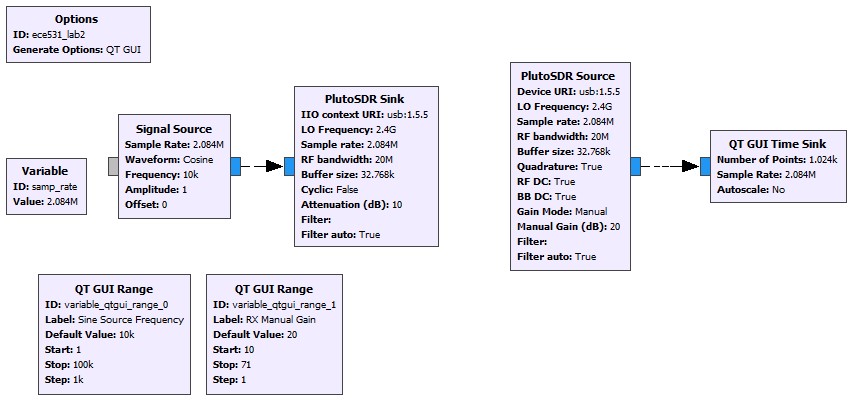


Figure 7: GRC flowgraph for Lab 2, Section 1.1.2

*Note: Pluto SDR source is sourcing the data from the Pluto receiver; Pluto SDR sink is feeding the signal in to the Pluto transmitter*

*sink in the GRC = source in the Pluto*

Double-click on each PlutoSDR block (source and sink) to look at the block properties.

* 1. The RF bandwidth refers to the bandwidth of the configurable bandpass filter in Pluto’s front end. This is not the same bandwidth as the sink’s block. The RF bandwidth was 20 MHz.
  2. The “Cyclic” boolean selection for in the PlutoSDR Sink block? If set to true, the first buffer of samples will be repeated until the program is stopped. The PlutoSDR IIO block will report its processing as complete: the blocks connected to the PlutoSDR IIO block won't execute anymore, but the rest of the flow graph will. This is useful for

1. What does the Manual Gain control in the PlutoSDR source? What other strategies are available? the Pluto can be configured to have a fixed receive gain or a dynamic receive gain, called Automatic Gain Control (AGC). AGC is a closed-loop feedback circuit that controls the amplifier’s gain in response to the received signal. Its goal is to maintain a constant output power level despite a varying input power level.[[1]](#footnote-1) In contrast, when in manual gain control mode, the ACG circuit is turned off and the user specifies what receive gain to use.[[2]](#footnote-2)
2. Run the flowgraph. Note: you may have to enter change the Device URI. On some operating systems libIIO finds the first available device when this field is left blank.

A picture containing graphical user interface

Description automatically generated

Figure 8. Rx gain of 20 dB

1. Adjust the RX gain. At what value does the received signal begin to distort or clip? The signal begins to become more square starting at a gain of 15 dB.
2. Replace the “QT Time Sink” block with a “QT GUI Sink”. Explore the options provided by the new sink block. What happens if you increase the number of averages? When the number of averages is zero, some spurs’ heights are very large, and on average the heights of the spurs are random. As the number of averages increases, the height of the spurs decreases as frequency increases. Why does the frequency response change when you change the window function? The purpose of window functions is fixme The input signal changes because it’s being convolved with different window functions.
3. What is the transmitted RF frequency of the sinusoid? Why? The transmitted RF frequency of the sinusoid is 2.4 GHz + 10k.

### GNU Radio as a libIIO client

The PlutoSDR source and sink blocks contained in the Industrial IO module provide a convenient way to interface with PlutoSDR hardware. Generic blocks are also available and can be useful when interfacing with generic SDR hardware compatible with IIO. The specific values necessary for the IIO Device Source/Sink and IIO Atribute blocks were found previously using IIO command line tools, such as iio\_attr. The default values for these blocks are shown in Figure 9. A quick reference on using these tools will be posted to D2L.

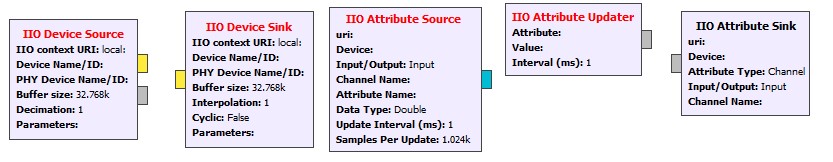


Figure 9: Default values for IIO Device blocks to be used for Section 4.1.3.

Figure 9Repeat the loopback experiment from Section 4.1.2 without using the PlutoSDR source and sink blocks; use only the “IIO Device Source / Sink” and “IIO Attribute Source / Sink / Updater”

1. Change the TX and RX center frequency from 2.4GHz to 915MHz.
2. Increase the sample rate from 2.084 MSPS to a greater value of your choosing.
3. Verify each of the above have been set in the hardware using IIO Scope plugins or iio\_attr.

## Measurements and the Radio

In this experiment, we show that the receiver has its inherent noise. This could be from the ADC circuit, quantization noise, or thermal noise. It is therefore best to transmit a signal with an an amplitude as close as possible to full scale, which for the Pluto is . Transmitting a signal whose amplitude is too low may become merged in with the receiver noise. Such a signal will be inseparable from the noise. The takeaway for this experiment was that it is always desirable if the signal’s amplitude is close full scale.

To illustrate this, we transmitted an vector of zeros concatenated with a sine wave. When observing the signal from the perspective of the receiver, the zero samples were not zero. Rather, they had some amplitude to them which comes from the hardware itself.

To minimize the impact of hardware imperfections, we increased the amplitude of the signal. The results are shown below:

Table 1. SNR for different signal amplitudes

|  |  |  |  |
| --- | --- | --- | --- |
| signal amplitude | signal power (dBm) | noise power (dBm) | SNR |
| 0.10 | 0.03 | 5.364670E-04 | 17.53 |
| 0.40 | 0.13 | 9.700000E-04 | 21.15 |
| 0.70 | 0.23 | 8.760000E-04 | 24.23 |
| 1.00 | 0.31 | 4.048829E-04 | 28.79 |

Chart, histogram

Description automatically generated

Figure 10. Received loopback signal with signal half nulled, amplitude = 1.

|  |  |
| --- | --- |
| Chart, histogram  Description automatically generated  Figure 11. SFDR when signal amplitude = 0.01  Chart  Description automatically generated  Figure 12. SFDR when amplitude = 0.8 |  |
|  |  |

Chart, line chart

Description automatically generated

Figure 13. SFDR when amplitude = 1

Figures 14 – 16: Spectrum at different receiver gains with signal amplitude = 1

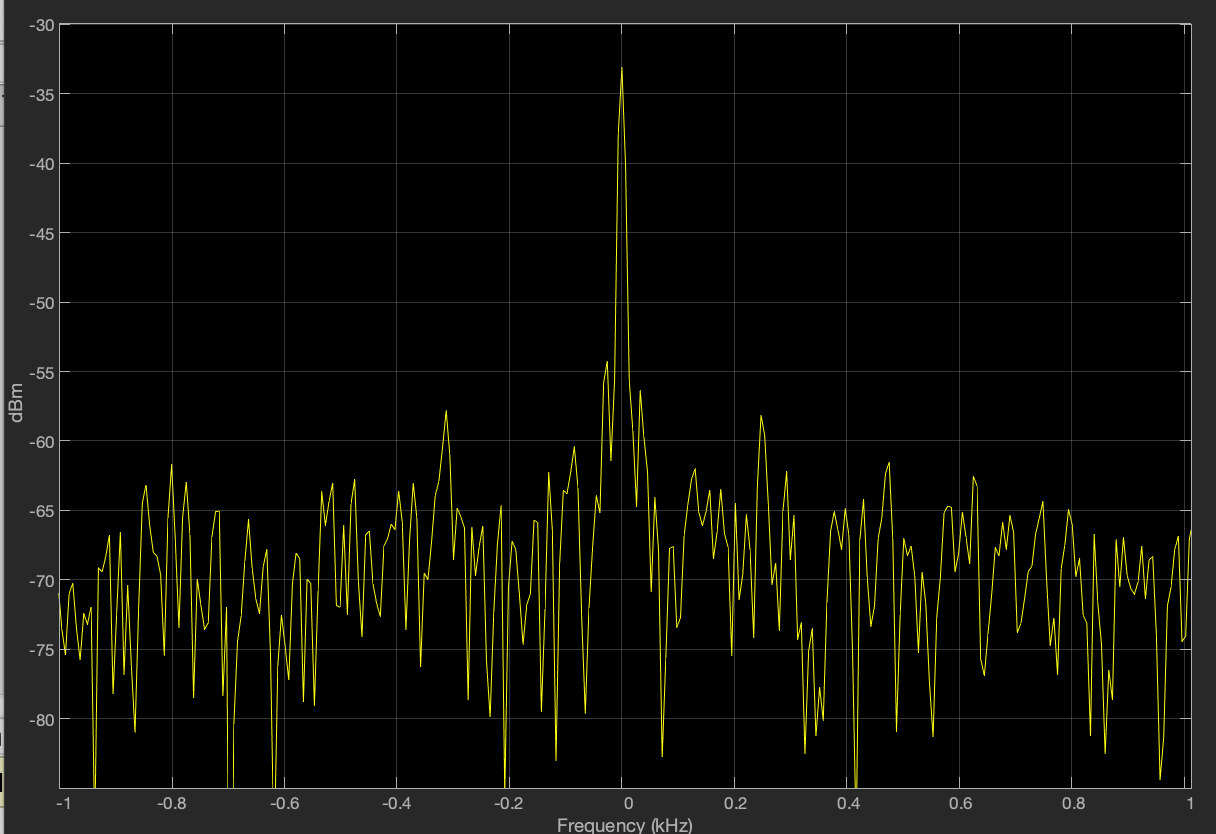


Figure 14. Rx gain = 10

Chart

Description automatically generated

Figure 15. Rx gain = 20

Chart

Description automatically generated

Figure 16. Rx gain = 30

1. https://pysdr.org/content/pluto\_advanced.html [↑](#footnote-ref-1)
2. https://pysdr.org/content/pluto\_advanced.html [↑](#footnote-ref-2)