Lab 0 - Introduction to SDR

In the laboratory students will learn to configure a ADALM-PLUTO Software-Defined Radio (SDR) and perform a basic experiment and will analyze the signals obtained.

> **NOTE:**
>
> Before starting this laboratory, students are required to access the following two software tools:
>
> Install SDR software on their computers by referring to the "Installing GNU Radio and IIO libraries" document.
>
> Sign up and familiarize themselves with the RHL-RELIA web interface. For registration, please follow the steps outlined in the "RHL-RELIA - Quickstart guide."

GNU Radio, an open-source project, is dedicated to developing software for SDR systems. The GNU Radio Companion (GRC) serves as a graphical design tool, allowing users to create GNU Radio software without coding. Users can add blocks representing signal processing functions to a diagram, configure them, and interconnect them through a graphical interface. This diagram, known as a "flowgraph," is then transformed into an executable script that processes samples by executing the configured blocks in the required order. In this lab, GRC is utilized to configure and run SDR software. Advantages of GNU Radio include their non-dependence on programming and the self-documenting nature of flowgraphs.

1 Building a I/Q system

As the first step, let's create a basic I/Q signal generator and visualize it in a constellation domain as is illustrated in Figure 1. I/Q signals or complex signals, referred to as in-phase (I) and quadrature-phase (Q) signals, offer a versatile and efficient solution in communication systems. This approach brings advantages such as optimized spectrum utilization, streamlined signal processing, and the capability to implement sophisticated modulation and demodulation schemes.

 This project is funded by the National Science Foundation Award #2141798.

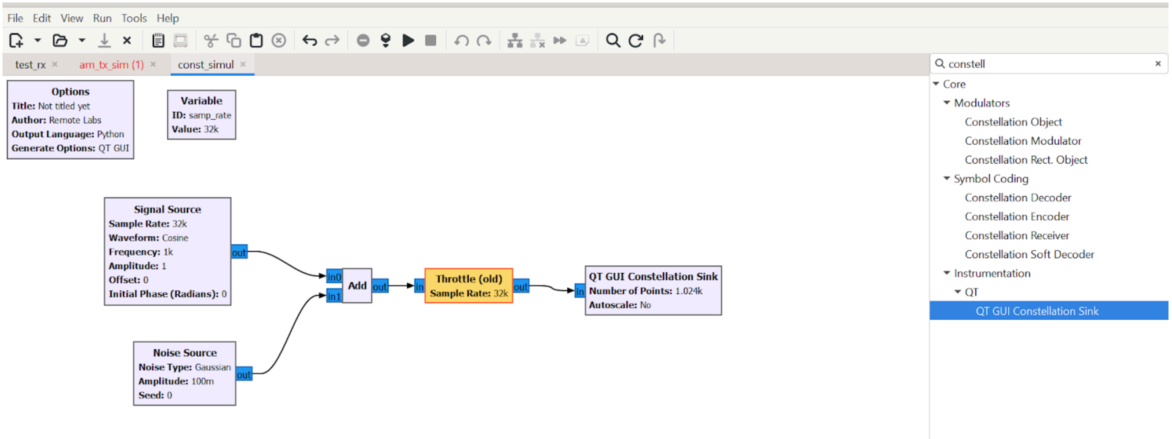

Figure 1

In this flowgraph, the "Signal Source" block generates a 1kHz I/Q cosine tone with a sampling rate of samp_rate (32kS/s), and the resulting signal is visualized in a constellation plot. A constellation plot serves as a visual representation that simplifies the analysis and comprehension of complex signals in digital communication systems.

To do this, please take the following steps:

1. Open GNU Radio Companion on your computer and create a new QT GUI flowgraph (File -> New -> QT GUI).
    a. To launch GNU Radio Companion, the method depends on your operating system. For Linux users:
        i. Open a terminal (Applications -> Terminal).
        ii. Run "gnuradio-companion".
2. In the right panel, locate and expand the "Waveform Generators" category. Find the "Signal Source" block and drag it to the left pane, adding a new "Signal Source" block to your flowgraph.
3. Add the “Noise Source” block and double click on that to modify the amplitude to from 1 to 0.1
4. Add an “Add” block and connect to sum up the signal and noise.
5. Add a "Throttle (old)" block to the left pane. Note. Depending on the version of GRC you might have only "Throttle". In that case, use simply "Throttle".
6. Connect the output port of the "Add" block to the input port of the "Throttle (old)" block. Click the output port of the "Signal Source" block
7. Add a "QT GUI Constellation Sink" block and connect its input port to the output port of the "Add" block. Go to the "Options" block, double-click on it, locate "Run," and ensure it is set to AutoStart.

Save the program in a grc file. Click on "File" -> "Save" and enter a name for the file.

2 Building a AM transmitter in GRC (simulation)

For the initial task, let's construct an AM Transmitter using a GNU Radio-based simulation, as illustrated in Figure 2.

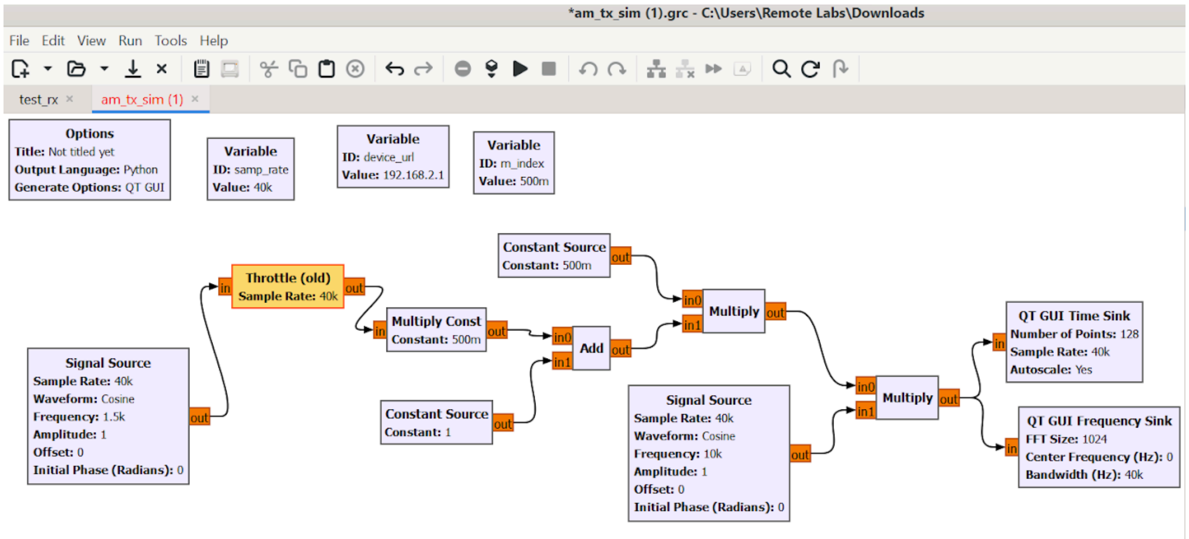

Figure 2

In this flowgraph, the "Signal Source" block generates a 1.5kHz cosine tone with a sampling rate of samp_rate (40kS/s) that will act as the signal to be transmitted. Then another "Signal Source" block will work as a modulator at 10KHz.

To do this, please take the following steps:

8. Open GNU Radio Companion on your computer and create a new QT GUI flowgraph (File -> New -> QT GUI).
    a. To launch GNU Radio Companion, the method depends on your operating system. For Linux users:
        i. Open a terminal (Applications -> Terminal).
        ii. Run "gnuradio-companion".
9. In the right panel, locate and expand the "Waveform Generators" category. Find the "Signal Source" block and drag it to the left pane, adding a new "Signal Source" block to your flowgraph.
10. By default, all signal types are set to complex (blue), but for this assignment, the types should be changed to float (orange). You will need to manually adjust the type of input/output accordingly.
11. Double-click the newly added "Signal Source" block from the left pane (not from the block list in the right pane). A properties window will appear. Change the following:
    a. Frequency: 1000 -> 1500.

b. Click "Okay" to apply.

12. Add a "Throttle (old)" block to the left pane. You can expand the "Misc" category in the right pane or use "CTRL + F" to search for "Throttle (old)". Note. Depending on the version of GRC you might have only "Throttle". In that case, use simply "Throttle".
13. Connect the output port of the "Signal Source" block to the input port of the "Throttle (old)" block. Click the output port of the "Signal Source" block and then click the input port of the "Throttle (old)" block.
14. Similarly, add a "Multiply Const" block and connect its input port to the output port of the "Throttle (old)" block. Double-click the block and change the multiplication constant to 0.5.
15. Alternatively, use the variable block. This block has a name and a value, and you can enter the variable name in any parameter input.
16. Add the rest of "Add," "Multiply," and "Signal Source" blocks as shown in Figure 2.
17. Add a "QT GUI Time Sink" block and connect its input port to the output port of the "Multiply" block. Double-click the "QT GUI Time Sink" block, modify parameters for data visualization. Repeat this for the "QT GUI Frequency Sink" block.
18. Go to the "Options" block, double-click on it, locate "Run," and ensure it is set to AutoStart.
19. Save the program in a grc file. Click on "File" -> "Save" and enter a name for the file.

Running in a RHL-RELIA system

If you did everything correctly, the flowgraphs is now ready to be executed. From the menu bar, click 'Run' -> ‘Generate’ to compile your program.

- If your flowgraph has syntax errors, you won't be able to execute the program and the "Execute" button will get disabled. Properties or connections containing errors are marked red on the graphical representation of the flowgraph. To see the list of errors, click 'View' -> 'Flowgraph Errors' from the menu bar. Common errors are:
    - Incorrect output port -> input port connection.
    - Data type mismatch (paired ports must share a same data type - complex number, floating point number, etc.). In many cases, you can change the data type by double-clicking the blocks.
    - Incorrect block properties.

If you have successfully implemented and executed the program described above, proceed to RHL-RELIA, log in as a student, and access the lab. Reserve one module, and upon successful reservation, enter the RELIA User Interface. Upload the two previously designed grc files and select one in the Rx column and the other in the Tx column, as illustrated in Figure 3.

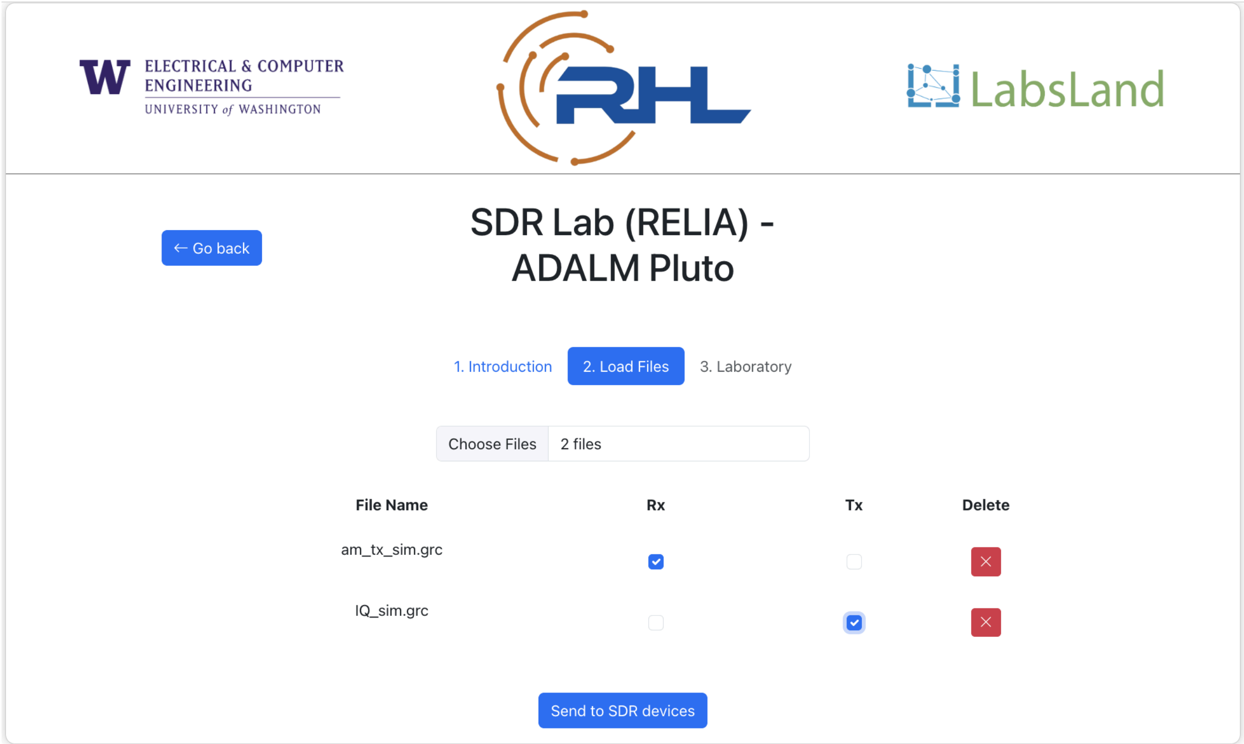

Figure 3

Q1. <u>I/Q system (15 points)</u>

1. Following the execution of the I/Q system in the Constellation diagram, what shape do you observe in the plot? Provide your reasoning for the observed shape and explain why it takes that specific form **(Providing screenshots of the plots from RHL-RELIA can help support your analysis.)**

   Hint: Think about the "Unit circle".

2. Observe the impact on the diagram if you were to increase the amplitude of the "Noise" block.
3. Explain the changes that may occur in the Constellation diagram due to the increase in amplitude of the "Noise" block, as conducted in part 2.

Q2. <u>AM transmitter (15 points)</u>

1. Upon completing the simulation of the AM transmitter, what diagrams do you observe? Provide an explanation of your interpretation of the plots presented in both the Time and Frequency domains **(Providing screenshots of the plots from RHL-RELIA can help support your analysis.).**
2. Determine the power of each peak (point of maximum power) in the frequency plot.
3. What parameter influences the separation of the peaks? Justify your response.

3. PlutoSDR Integration to GNU Radio

Now, you are ready to run real-world wireless experimentations with GNU Radio. This is made possible by the **"PlutoSDR Source"** and **"PlutoSDR Sink"** GNU Radio blocks. "PlutoSDR Source" block will receive and output wireless signals from your PlutoSDR, and the "PlutoSDR Sink" block will transmit wireless signals which you input to this block (again, using your PlutoSDR).

Properties of the "PlutoSDR Sink" block: (TRANSMITTER)

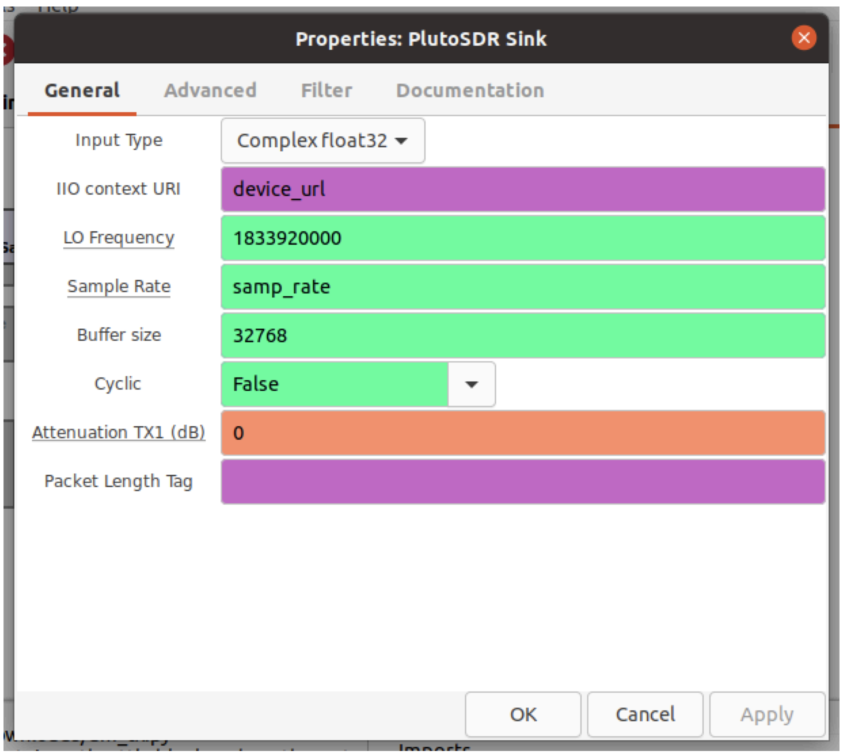

Figure 4

Descriptions of parameters provided by [Analog Devices]:

- **IIO context URI**: If using GNU Radio remote on a PC, set the target IP address using ip:XXX.XXX.XXX.XXX (some boards like the PlutoSDR run a Zeroconf/Avahi deamon so the the URI may look like this: ip:pluto.local) or via USB using the URI usb:XX.XX.XX
- NOTE: If you are utilizing RHL-RELIA, use the IP address 192.168.3.1. If you are employing ADALM-PLUTO in a traditional manner, use the IP address specified in the Config.txt file located on the Pluto drive.
- **LO Frequency(MHz):** Selects the TX local oscillator frequency. Range 70MHz to 6GHz with 1Hz tuning granularity. Read More

- **Sample Rate(MSPS):** Frequency at which the hardware will input/output samples Read More

- **RF Bandwidth(MHz):** Configures TX analog filters Read More

- **Buffer size**: Size of the internal buffer in samples. The IIO blocks will only input/output one buffer of samples at a time.
- **Cyclic**: Set to “true” if the “cyclic” mode is desired. In this case, the first buffer of samples will be repeated on the enabled channels of the PlutoSDR until the program is stopped. The blocks connected to the PlutoSDR IIO block won't execute anymore, but the rest of the flow graph will.
- **Attenuation (dB):** Individually controlls attenuation for TX1. The range is from 0 to -89.75 dB in 0.25dB steps. Read More
- **Filter:** Allows a FIR filter configuration to be loaded from a file. Read More
- **Filter Auto:** When enabled loads a default filter and thereby enables lower sampling / baseband rates.

Properties of the "PlutoSDR Source" block: (RECEIVER)

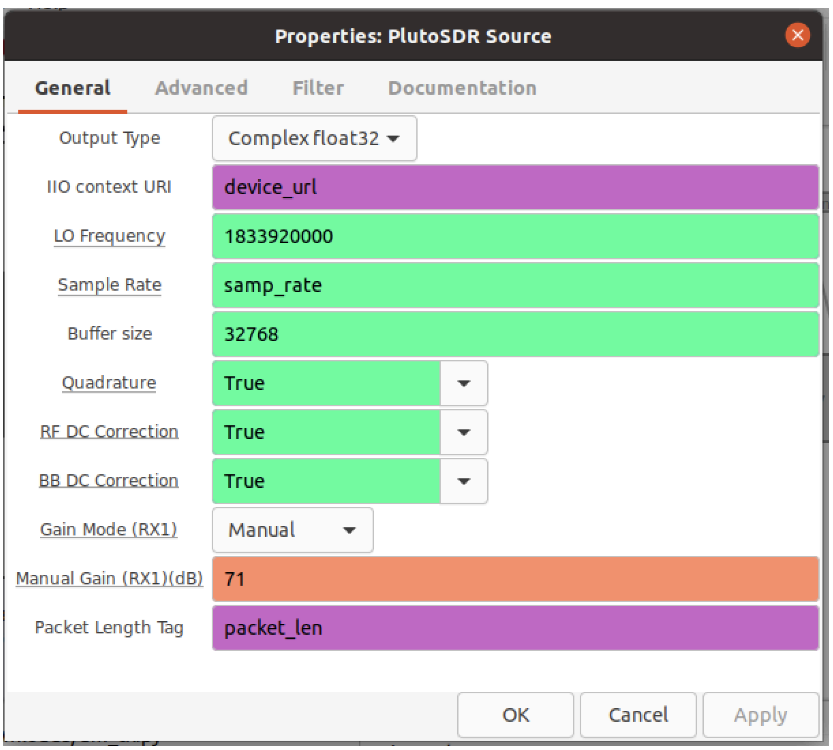

Figure 5

Descriptions of parameters provided by Analog Devices:

- **IIO context URI**: If using GNU Radio remote on a PC, set the target IP address using ip:XXX.XXX.XXX.XXX (some boards like the PlutoSDR run a Zeroconf/Avahi deamon so the the URI may look like this: ip:pluto.local) or via USB using the URI usb:XX.XX.XX
- NOTE: If you are utilizing RHL-RELIA, use the IP address 192.168.2.1. If you are employing ADALM-PLUTO in a traditional manner, use the IP address specified in the Config.txt file located on the Pluto drive.
- **LO Frequency(MHz):** Selects the RX local oscillator frequency. Range 70MHz to 6GHz with 1Hz tuning granularity. Read More
- **Sample Rate(MSPS):** Frequency at which the hardware will input/output samples. Read More
- **RF Bandwidth(MHz):** Configures RX analog filters Read More

- **Buffer size**: Size of the internal buffer in samples. The IIO blocks will only input/output one buffer of samples at a time.
- **Tracking** Read More
    - **Quadrature**
    - **RF DC**
    - **BB DC**

- **Gain Mode:** Selects one of the available modes: manual, slow_attack, hybrid and fast_attack. Read More

- **Manual Gain(dB):** Controls the RX gain only in Manual Gain Control Mode (MGC). Read More

- **Filter:** Allows a FIR filter configuration to be loaded from a file. Read More

- **Filter Auto:** When enabled loads a default filter and thereby enables lower sampling / baseband rates.

Sample flowgraphs:

Create your own GNU Radio + PlutoSDR flowgraphs as illustrated in Figure 6 and Figure 7 to become familiar with the system. Use the parameters listed in Table 1 as a reference.

A simple sine wave (CW) transmitter

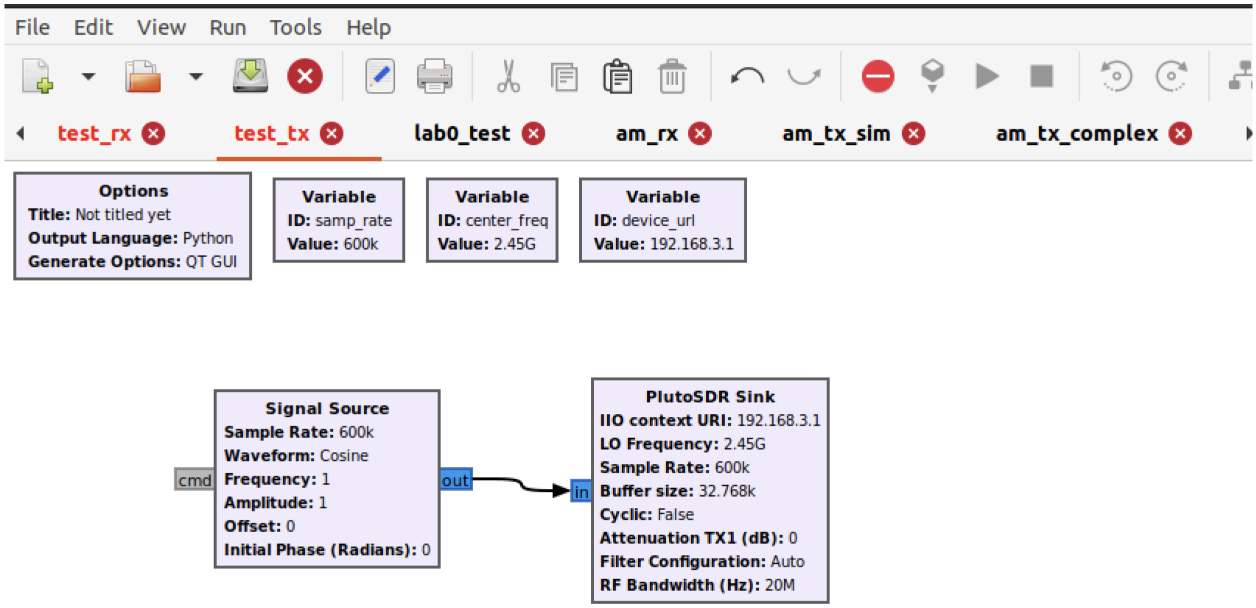

Figure 6

A simple spectrum analyzer (power spectral density plotter)

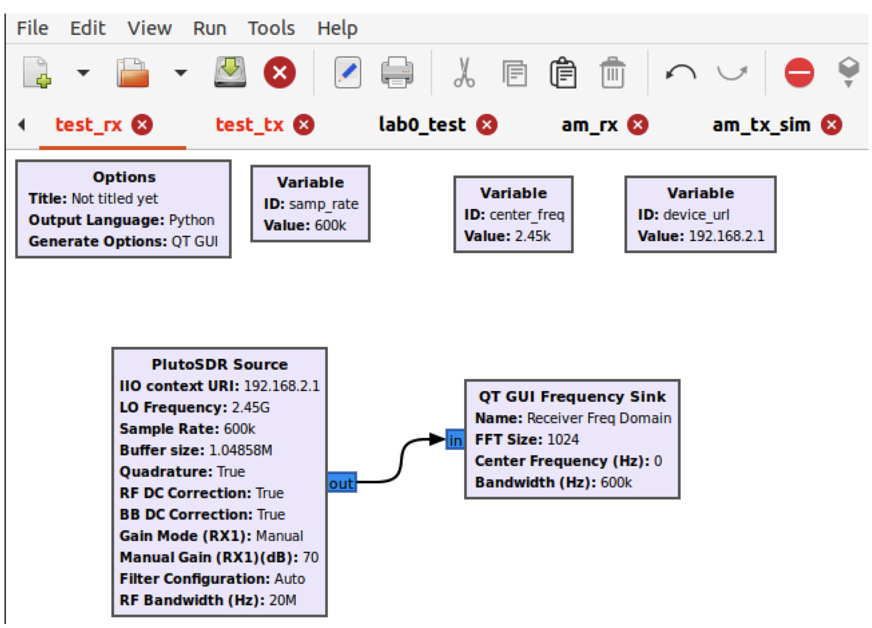

Figure 7

Table 1 - Adalm Pluto Parameters

Sample Rate	600 KHz
Lo Frequency	2.45GHz
FFT Size	1024
Attenuation - TX	0 dB
RF Gain (db) - RX	Manual - 70 dB
IIO context URI - TX	192.168.3.1
IIO context URI - RX	192.168.2.1

Q3. ADALM-PLUTO Integration (20 points)

After utilizing ADALM-PLUTO, please address the following questions **(Providing screenshots of the plots from RHL-RELIA can help support your analysis)**:

1. Share your observations on the generated Frequency plot.
2. Provide comments on the level of noise and identify any interference. Estimate their respective power levels.
3. In this example transmitted frequency in the ADALM-Plutos is set to 2.45GHz. Explore the effects of trying different frequency values, such as 1.4GHz. Capture screenshots of the results you obtain for reference.

Note: You might observe a shift in frequency near 10 kHz in the Frequency Plot. However, this effect is common in wireless devices and will be explored further in Lab 1.

Submission Instructions

Please compile your observations, analyses, and responses to the lab questions into a PDF report. Ensure that your report is well-organized, and include screenshots of relevant plots and visualizations where necessary. Clearly label and number your responses according to the corresponding question numbers.