# **Lab 04**

**Analysis of demodulated AM signal in time and frequency domain and applying filtering operation to listen audio signal**

**Objectives**

**Part I:**

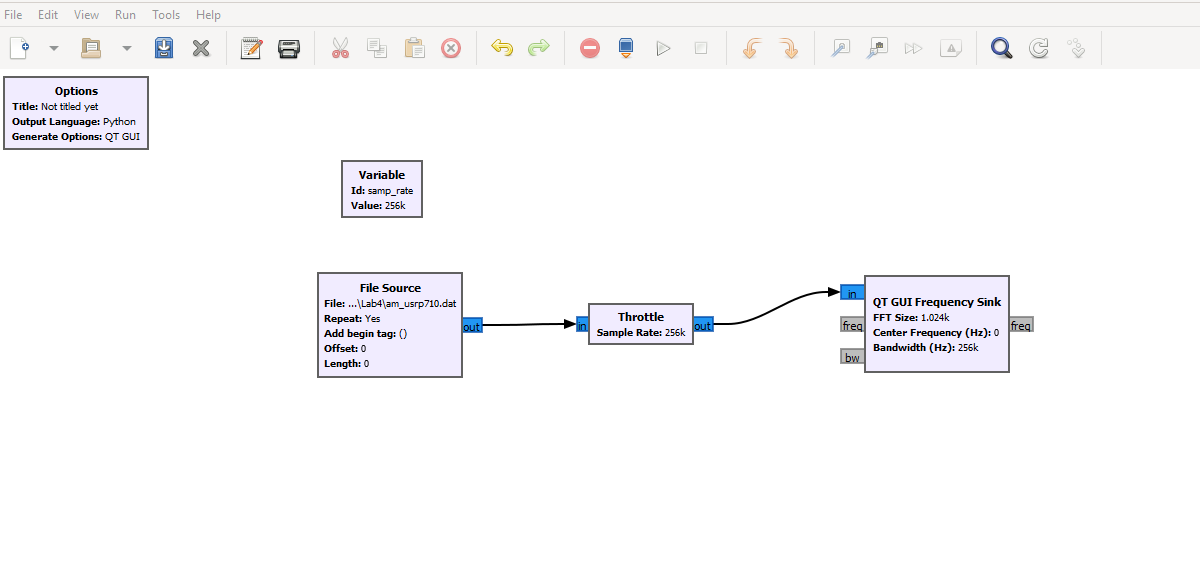
1. To open and analyze Amplitude Modulation (AM) signal recorded from Universal Software Radio Peripheral (USRP).
2. To demodulate AM signal and plot it.
3. To be able to listen demodulated audio signal.

**Part II:**

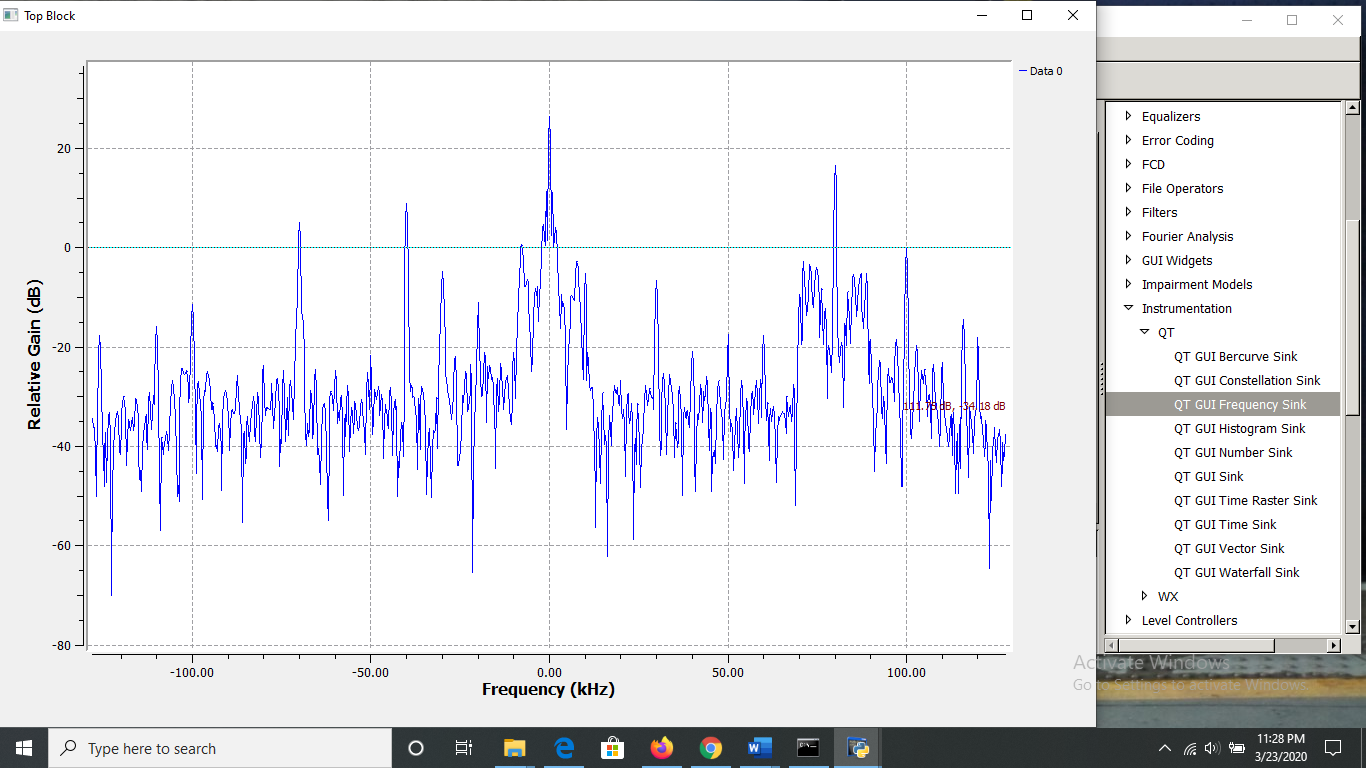
1. Generate and observe noise signal in GNU radio.
2. Model an AWGN channel

**Part I:**

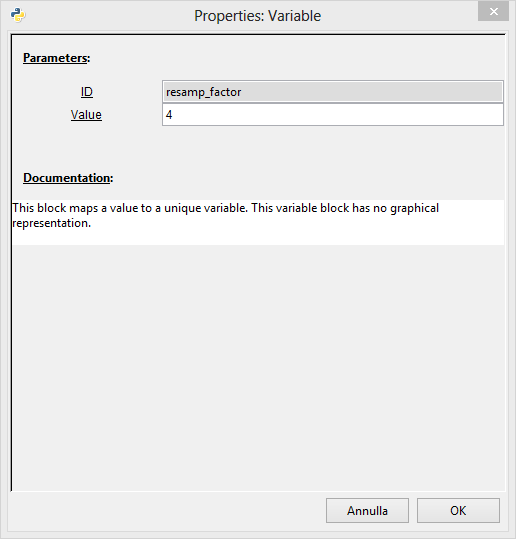
1. **Open and analyze Amplitude Modulation (AM) signal recorded from Universal Software Radio Peripheral (USRP)**
2. Before beginning this lab activity, make sure that you have already downloaded the file am\_usrp710.dat from aulaweb (or teams). The data file contains several seconds of recorded signals from the AM broadcast band. This data file was obtained from the USRP.
3. Construct the flow graph shown below consisting of a File Source, Throttle, and QT GUI Frequency Sink. Set the Sample Rate in the variable block to “**256000**”. This is the rate at which the saved data was sampled.



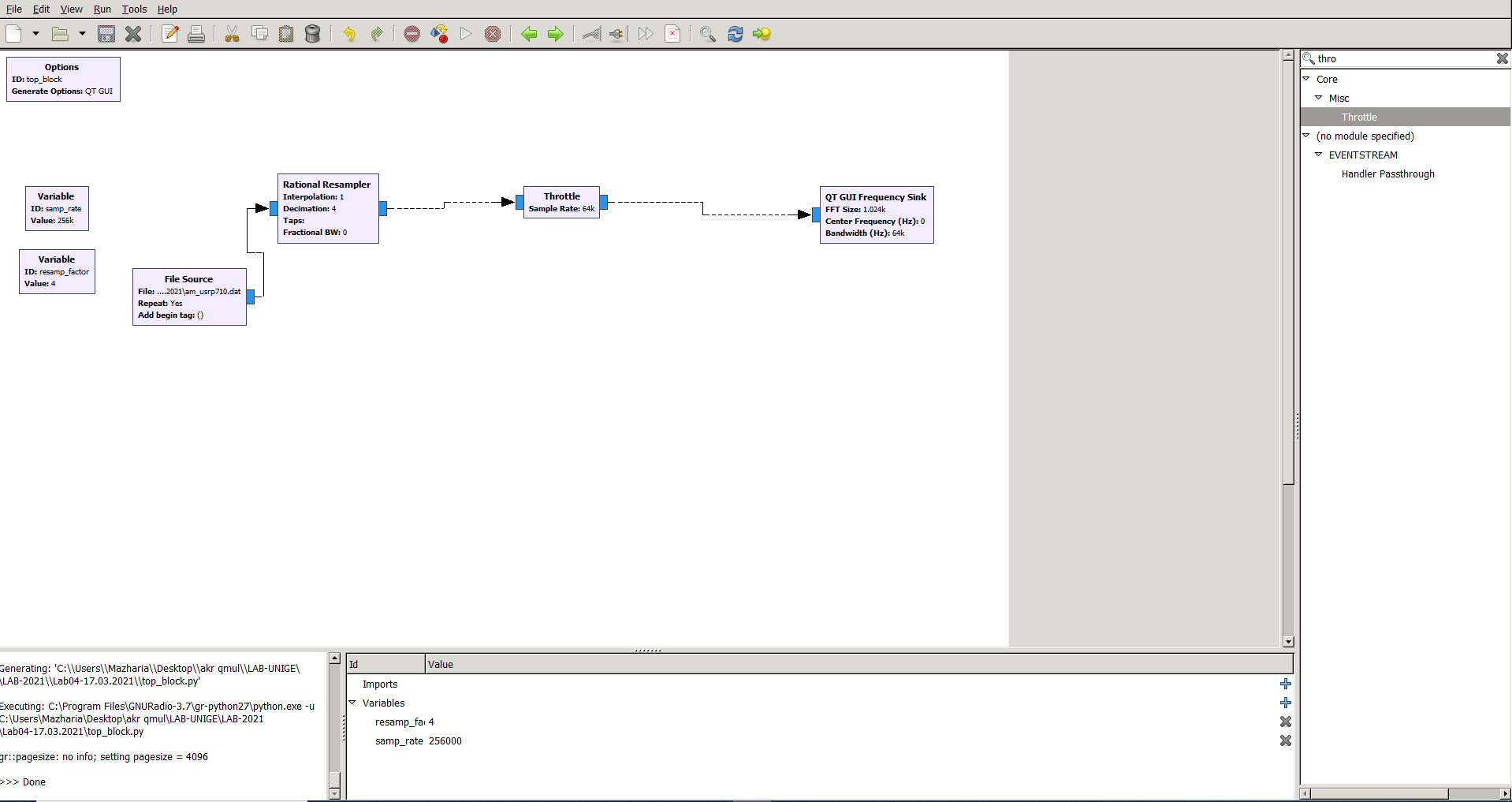
1. Double click on the File Source block. Click on the ellipsis (…) next to the File parameter. Locate the am\_usrp710.dat from the path “Desktop\Physical layer models and techniques for software radio\folder name”. The path to your file will appear in the block. Please notice that (for a bug of GRC) the path should contain **forward slashes** instead of **backward slashes** so you will need to manually substitute “\” with “/” (***use only if needed***). Set the Output Type to Complex. Set Repeat to Yes. This will cause the data to repeat so that you have a continuously playing signal.
2. Save and execute the flow graph. You should observe an FFT display similar to the one shown below. You may need to click on Autoscale button to scale the data as shown. Note the following:
   1. This data was recorded with a USRP whose central carrier frequency was set to 710KHz. Thus, the signal you see at the center (indicated as 0 KHz) is actually at 710 KHz. Similarly, the signal at 80 KHz is actually at 710KHz + 80KHz = 790KHz.
   2. The display spans a frequency range from just below -120KHz to just above 120KHz. This exact span is 256KHz, which corresponds to the Sample Rate that the data was recorded at.
   3. The peaks that you observe on this display correspond to the carriers for AM broadcast signals. You should also be able to observe the sidebands for the stronger waveforms.



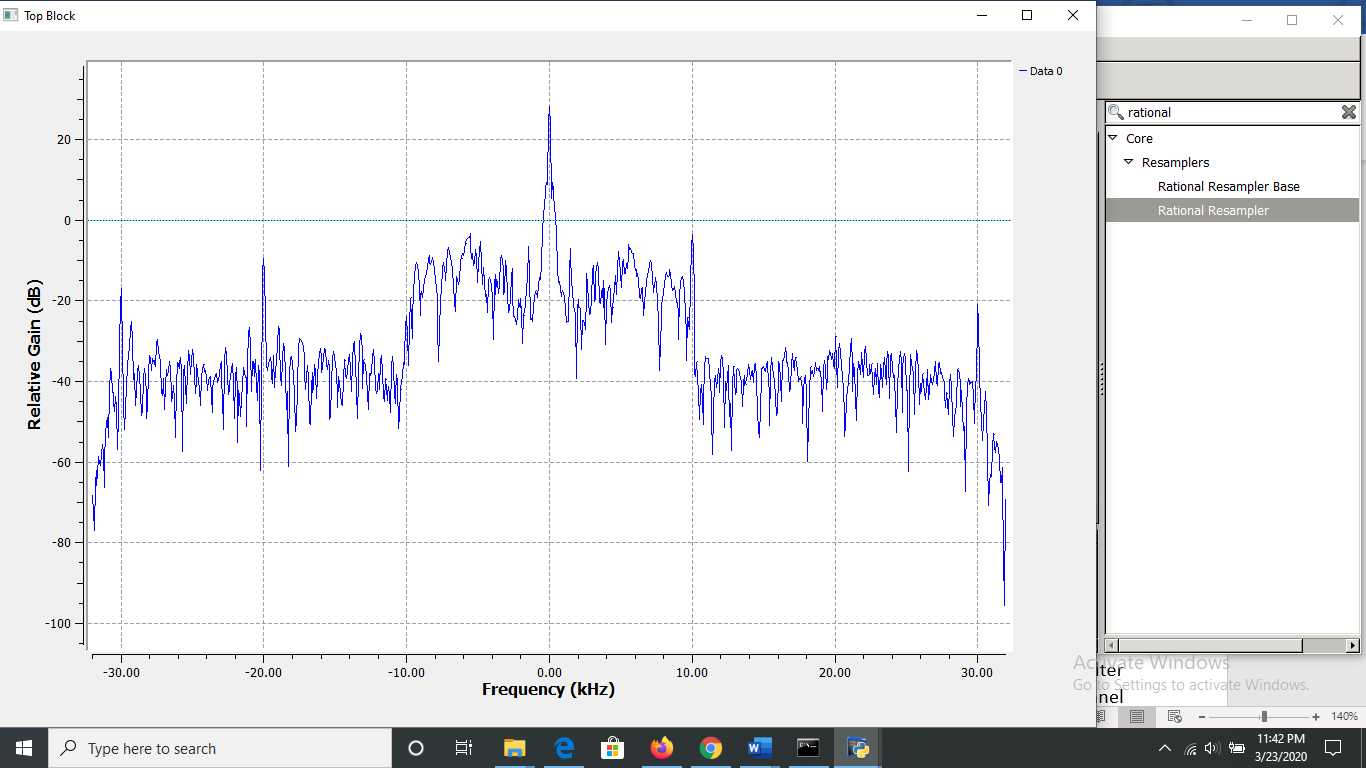
1. In this step we will expand the frequency scale on the FFT display so that you can view the signals with greater resolution. Recall that the span of the frequency axis is determined by the sample rate (256K for this file). While we cannot change the original data, we can resample it to either increase or decrease the sample rate. We will decrease the sample rate by using decimation. Modify the flow graph as follows:
   1. Add a Variable block (from the->”**Variables**” category). Set the ID to “**resamp\_factor**” and the Value to “4” as shown below.



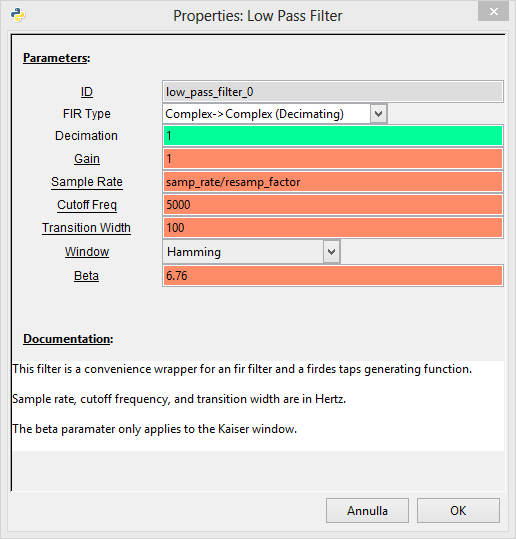
* 1. Add the **Rational Resampler** (NOT Rational Resampler Base!) from the ->”Resamples” category as shown below. Set its decimation factor to “**resamp\_factor**”. Thus, it will use the value of the variable set in the previous step to decimate the incoming data. That means that it will divide the incoming data rate by the decimation factor. In this example, the incoming **256K samp/sec** data will be converted down to **256K/4 = 64K samp/sec**.
  2. Note that the Throttle and FFT Sink now need their Sample Rates changed to correspond to this new rate. Change the Sample Rate in both of these blocks to samp\_rate/resamp\_factor. Now we can change the decimation factor in the Variable block, and it will be reflected in each of the other blocks automatically.
  3. Your flow graph should now appear as shown below.



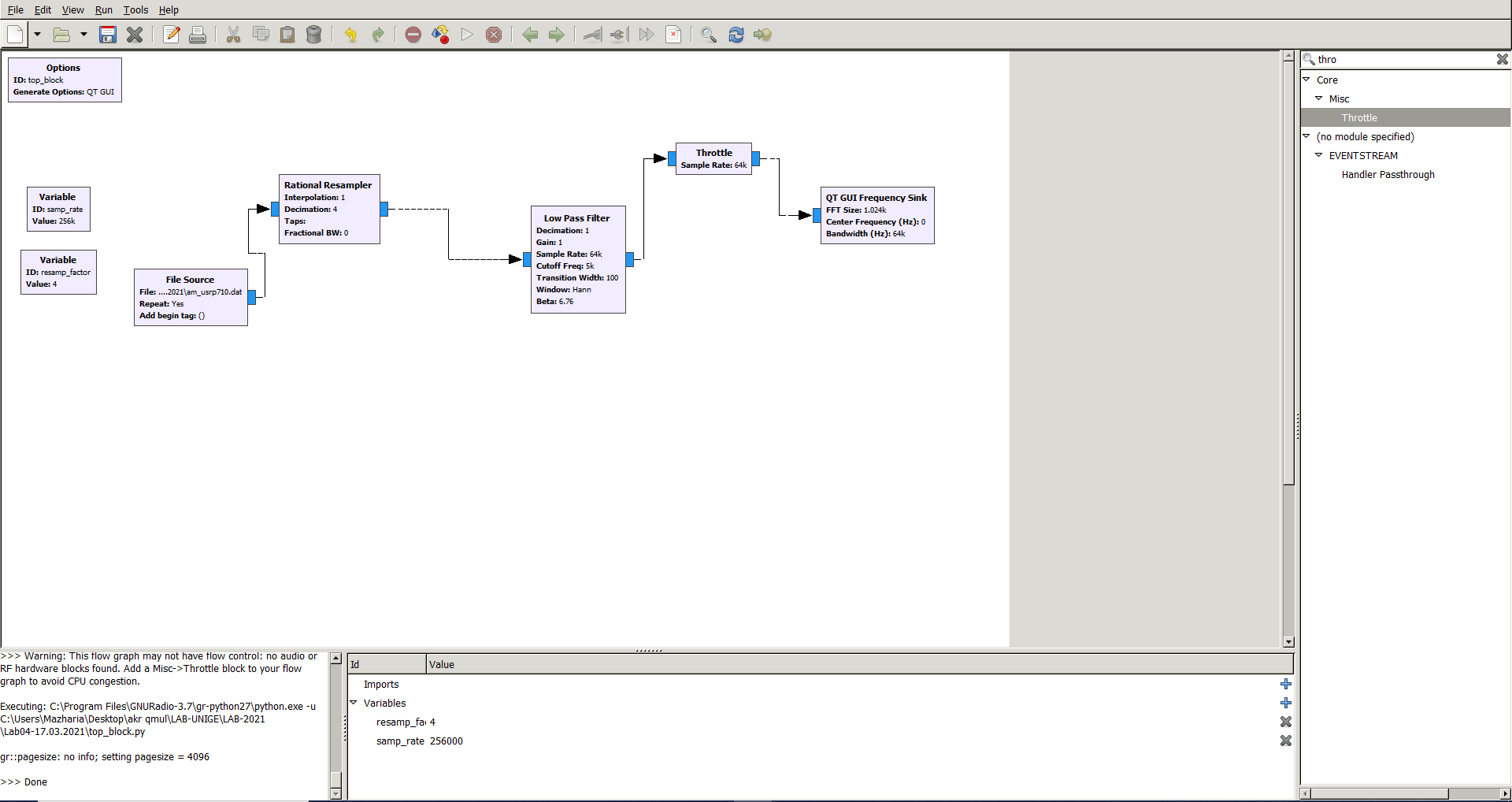
1. Execute the new flow graph. You should now observe a frequency span of only 64KHz (-32KHz to +32KHz).



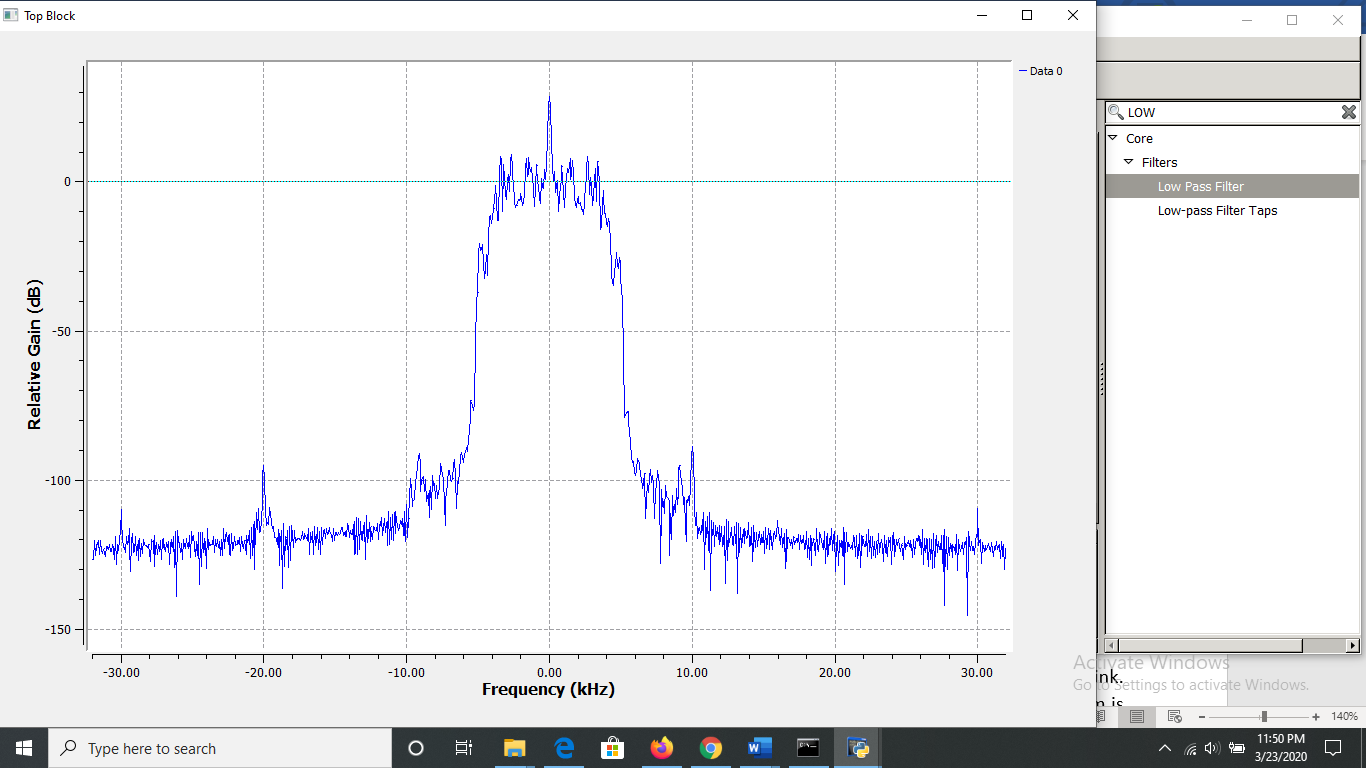
1. The bandwidth of an AM broadcast signal is **10KHz** (+/- 5KHz from the carrier frequency). You may find it useful to click the “Stop” button on the FFT plot to see this more clearly. Also, note that many stations also include additional information outside of the 10KHz bandwidth. This will be discussed later.
2. In order to select the station at **710 KHz** (represented by **0KHz** in our case) we need to insert a filter to eliminate all but the one station that we want to receive. This is often referred to as a channel filter. Since the station at **710 KHz** has been moved to 0 KHz (in the USRP) we will use a low pass filter. The station bandwidth is **10 KHz**, so we will use a low pass filter that cuts off at **5 KHz**. Insert the **Low Pass Filter** (from the “Filters” menu) between the **Rational Resampler** and the **Throttle**. Set the parameters as shown below.



The flow graph should be look like,



1. Execute the flow graph. You should see that only the station between **+/- 5KHz** remains.

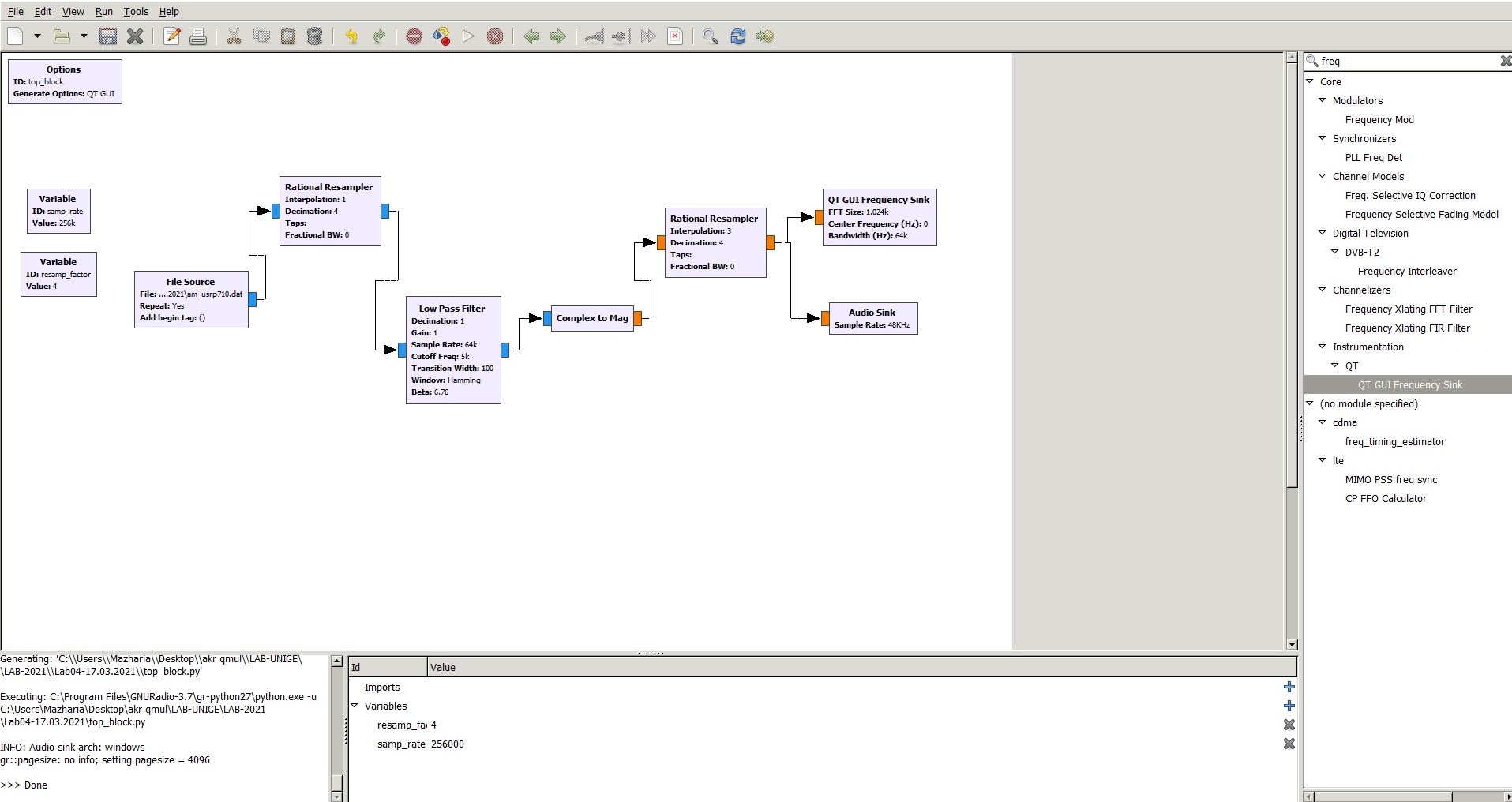


1. **Demodulate AM signal and plot the signal**
2. The next step is to demodulate the signal. In the case of AM, the baseband signal is the envelope or the magnitude of the modulated waveform. GNU Radio contains a **Complex to Mag block** (in the “Type Converters” category) that can be used for this purpose. Insert this block between the **Low Pass Filter** and the **Throttle.**
3. Note that the titles of some of the blocks are now red and the Execute Flow Graph icon is dimmed. This indicates an error. Prior to adding this block, all the block inputs and outputs were complex values. However, the output of the **Complex to Mag** block is Float (a real number). Thus, any blocks following this block should be Type: Float. Modify the Throttle and frequency Sink accordingly.

Execute and observe.

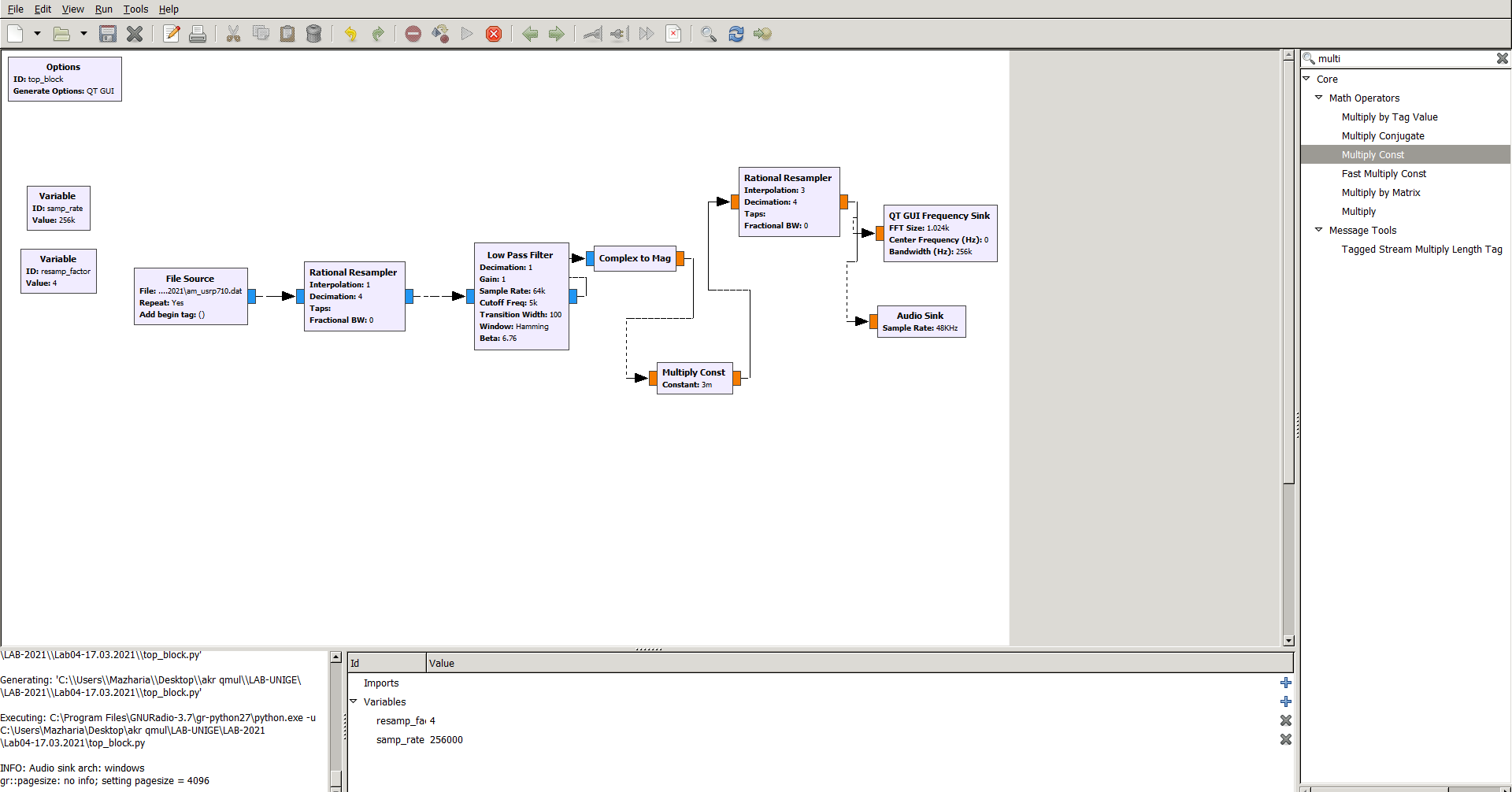
1. **Listen Audio signal**
2. The next step is to listen to this **demodulated** waveform to confirm that it is in fact receiving the baseband signal. Remove the Throttle and the FFT Blocks. Add an **Audio Sink** block to the output of the Complex to Mag block. Note that the default sample rate for the Audio Sink is 32KHz. Most current audio cards require a sample rate of at least 44.1KHz. Change the “sample rate” of the Audio Sink block to “48 KHz”.
3. Note that the sample rate out of the Complex to Mag block is 64K and the input to the Audio Sink is 48K. In order to convert 64K to 48K we need to divide (decimate) by 4 and multiply (interpolate) by 3. Insert a Rational Resampler between the Complex to Mag and Audio Sink blocks and set the decimation and interpolation as noted above. Also set its Type to Float->Float
4. Place a QT GUI frequency Sink at the output of the Rational Resampler (in addition to the Audio Sink). Change its Type to Float. The flow graph should be like the one shown below

**The flow graph looks like,**



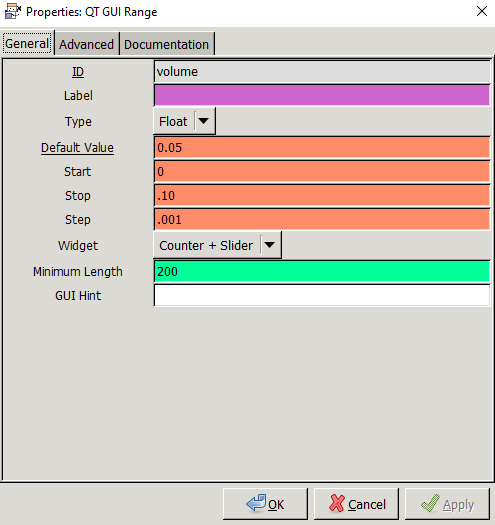
**And, observe the output.**

1. Execute the flow graph. The Frequency Sink should open and display the output waveform. However, you may not yet hear the audio from your speaker. This is due to the fact that the values of the samples going into the Audio Sink block are too large for the speaker. We need to apply an attenuator to reduce the size of the samples.
2. Insert a **Multiply Const** block from the -> ”Math Operators” category between the Complex to Mag block and the **Rational Resampler**. Set the IO Type of the block to Float. Set the constant value to **3m (0.003).**



You may be able to listen the audio now.

1. Add a variable slider block. Set the parameters as shown below.

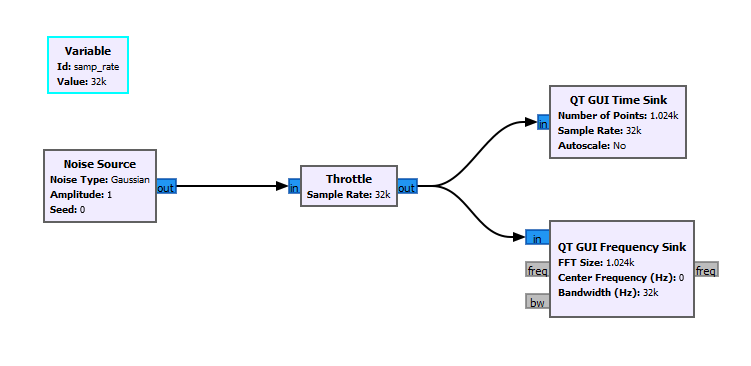


Set the constant in the Multiple const block to “volume” so that the slider controls it.

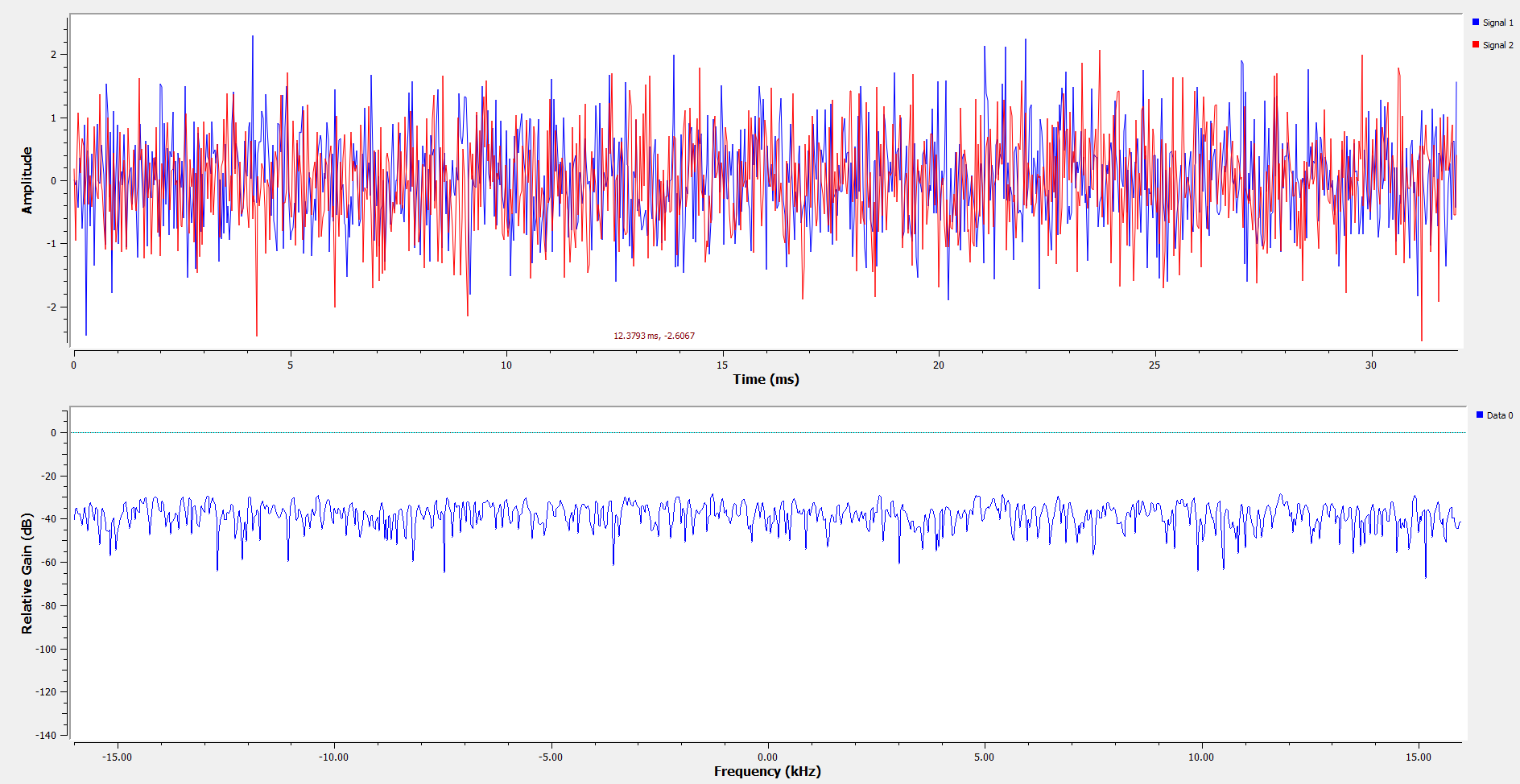
Execute the flow graph and change the slider by increasing or decreasing until the audio is easy to understand.

**Part II:**

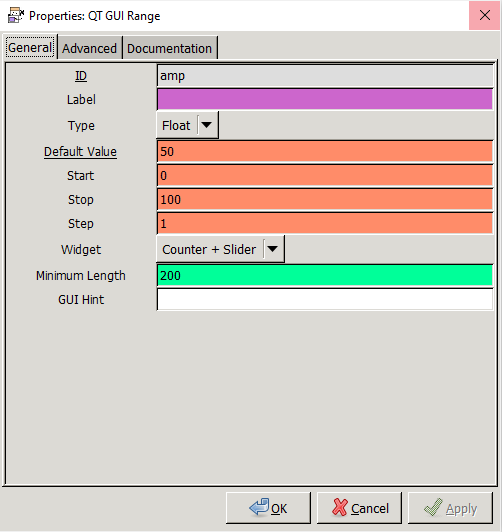
1. **Generate and observe noise signal in GNU radio.**
2. Draw a flow graph as shown in the figure below.



1. The output should be look like

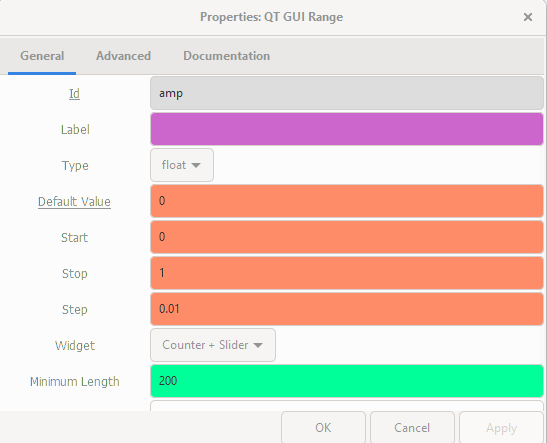


1. Use QT GUI Range and choose the parameters as shown below:



Change the amplitude and observe the output.

1. **Model an AWGN channel**
2. Generate a periodic sinusoidal signal *x(t)* with amplitude 1 and frequency 1KHz using the Signal Source Block.
3. Generate a gaussian noise *n(t)* using Noise Source Block with a varying amplitude. To vary the amplitude, you can use the QT GUI Range with the following parameters:

****

1. Create the proper flow graph to generate the following signal:

*y(t) = x(t) + n(t)*

1. Run and observer the output by changing the amplitude of the noise.

**You can upload your reports on our channel on Teams:**

**LAB - 2024/2025 Physical Layer for software radio Files Lab04 student reports**

**Please try to compress your gnu files (flow graphs) into one file called: Lab04\_yourName**

**Create a flow graph for Part I and another flow graph for Part II**