First Assignment

In this assignment, the main objective is to use GNU Radio Companion (GRC) software to create analog/digital signals and apply different modulation schemes on them. The GRC software is a graphical user interface that allows you to build GNU Radio flow graphs. To learn the basics of GNU Radio, a brief tutorial is provided below in which you can learn how to create and execute a GRC flow graph using basic blocks such as signal sources and graphical sinks.

I. Obtaining GRC

For this course, we work with GRC version 3.8.1.0, the latest version at the time of writing.

a) Linux: For linux users, installation should be easiest: Install the package *gnuradio* from your package repository.

b) Windows: Windows users can download an installer to get GRC here: http://www.gcndevelopment.com/gnuradio/index.htm

c) Mac OS: According to students of previous years, installation of GRC on Mac OS systems is quite hard. If possible, try to find a Windows/Linux computer/laptop to work on for this assignment. In case there is no such alternative available, follow the Mac OS installation guide here http://www.gcndevelopment.com/gnuradio/index.htm

II. Getting Started with the GRC

To launch the GRC software, open a Terminal in Linux and type *gnuradio-companion*. Then, an untitled GRC window similar to Figure 1 should open.

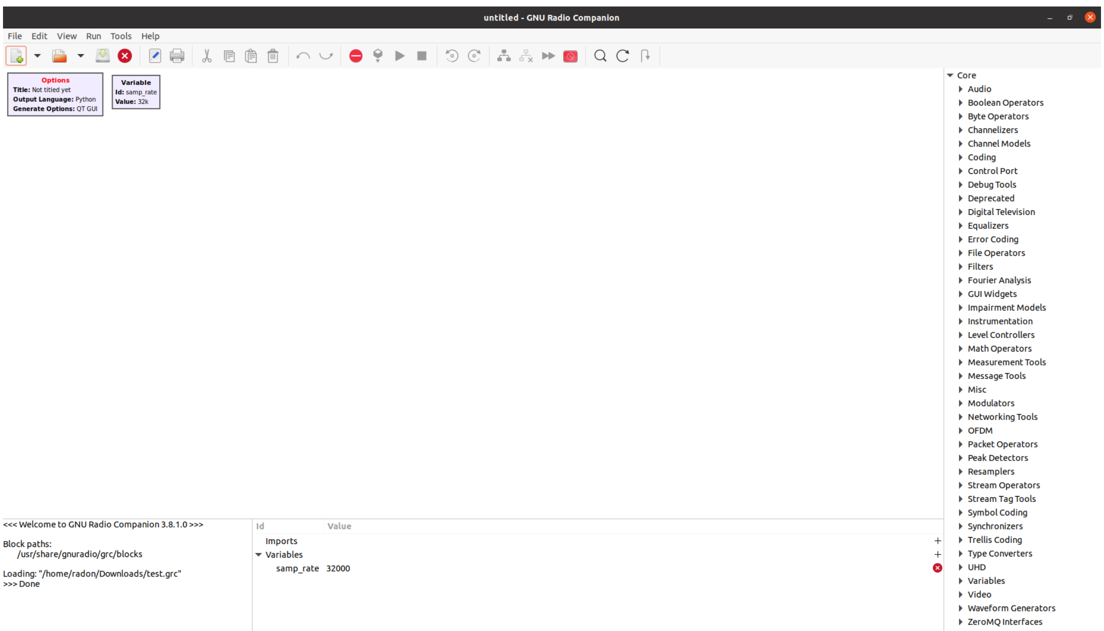

Fig. 1: Blank GRC flow graph

In this window, there are two basic blocks. The Options block sets some general parameters for the flowgraph. Note that the 'Options' block has a red top line. For blocks in general, this means that something is wrong in the block properties. Double-click on the Options block to see its properties. You should see a window like shown in Figure 2. Then, in the opened window

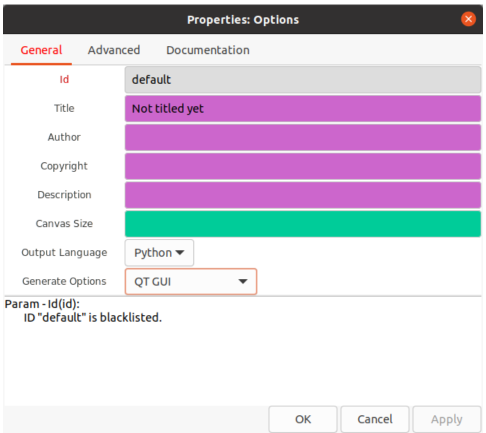

Fig. 2: Properties dialog box for top block

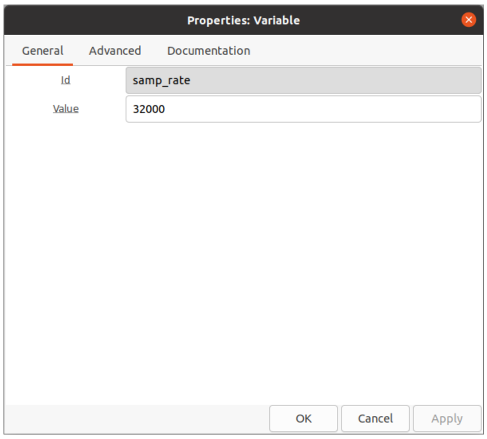

Fig. 3: Properties dialog box for sample_rate

- set some id instead of *default*. This is our problem, as explained by the bottom text box: Creators of GRC blacklisted 'default' as id.
- Optionally pick a title for your flowgraph.
- click OK to close the properties dialog.

The other block in the flow graph is a variable block that defines a variable called sample_rate and sets a value for it. Click on this block to see the variable name and its value, that is shown in the Figure 3. This variable can be used later to initialize other parameters in flow graphs.

A. *Adding Blocks to the Flow graph*

On the right side of the window in Figure 1 is a list of the blocks that are available. By expanding any of the categories (click on triangle to the left) you can see the available blocks. You can also click on the magnifying glass in the toolbar on the top of the window and simply type a search term to search all categories.

- Open the **Waveform Generators** category and double-click/drag-and-drop the **Signal Source**. Note that a Signal Source block will now appear in the main window.
- Double-click on the Signal Source block and the properties dialog will open. In Waveform option, you can specify the output waveform to be Constant, Sine, Cosine, Square, Triangle, and Sawtooth. For now, set Amplitude to 0.5, and change the output type from *complex* to *float*, and close the dialog.

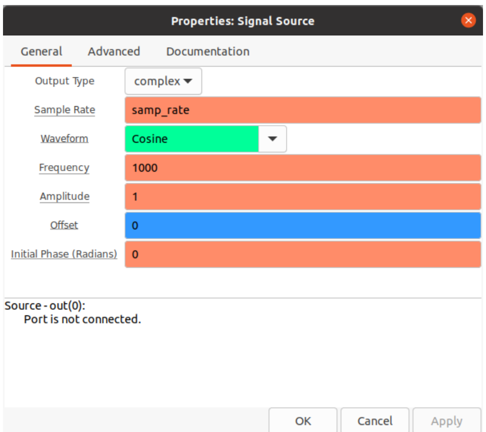

Fig. 4: Properties dialog box for signal source

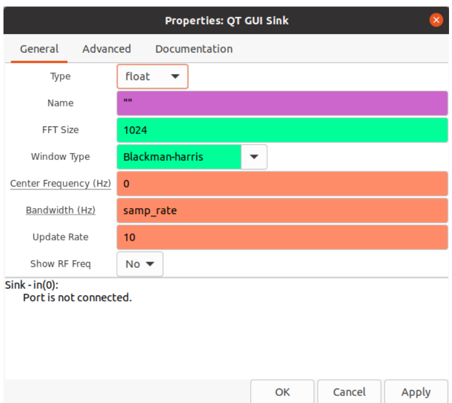

Fig. 5: Properties dialog box for scope sink

This Signal Source is now set to output a 1 kHz sinusoid with a peak amplitude of 0.5. In the flow graph, the Signal Source block will have an orange output tab, representing a float I/O type. If the block output type is chosen as complex instead of float, then the output tab will be blue.

- In order to view this wave, we need one of the graphical sinks. Expand the **Instrumentation** category and then the **QT** subcategory. Double-click on the **QT GUI Sink**. It should appear in the main window. Double-click on the block and change the Type to Float. Leave the other Parameters at their default values as shown in Figure 5. Click OK to close the properties dialog.
- In order to connect these two blocks, click once on the out port of the Signal Source, and then once on the in port of the Scope Sink. The resulting flow graph is shown in Figure 6.

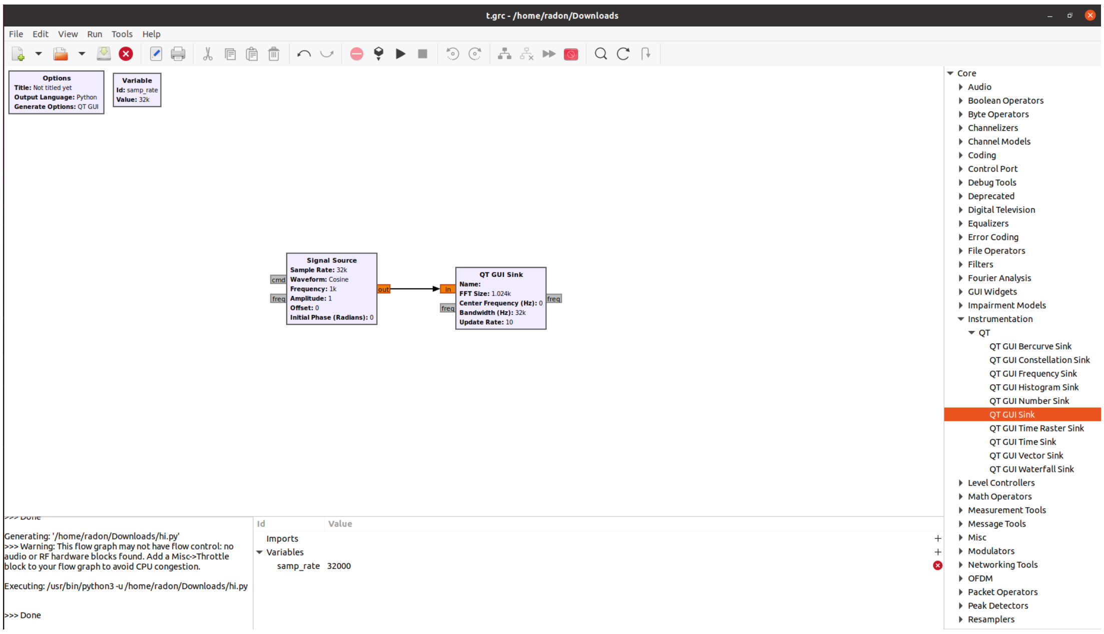

Fig. 6: Flow graph with signal source and scope sink

- If you run the flowgraph now, you will get output, *but* The problem with this flowgraph is that although the sample rate is set to 32000, there is no block which enforces this sample rate. Therefore, the flowgraph will consume as much of the computer's resources as it possibly can. This is wasteful and makes your device heat up a lot. To fix this problem, disconnect the Signal Source from the GUI Scope Sink by clicking on the arrow and pressing the Delete key. Expand the **Misc** category and place a **Throttle** in your flowgraph. Connect this block between the Signal Source and the GUI Scope Sink. If all went well, you will get a flowgraph resembling Figure 7.

B. Executing the Flow graph

In order to observe the operation of this simple system, we must generate the flow graph and then execute it. To do so, click the  icon in the GRC toolbar to execute the flow graph. Note: If the flowgraph has not yet been saved, a file dialog will appear when you click this button. Name this file as you want and save it. You must save it to not get errors. The generation stage converts your saved GRC flowgraph into executable Python code. The execution stage then runs the Python code that was generated in the previous step. A window should open, with several tabs. In the **Time Domain** tab, we can see several cycles of the sinusoid. Confirm that the frequency and amplitude match the values you expect.

C. Working with Sinks

For most of the tasks, you will only be interested in the **Time Domain** tab of the QT GUI sink. Instead of using the QT GUI sink, which generates 4 plots in 4 tabs, we can just use the **QT GUI Time Sink**, which only displays the time domain graph.

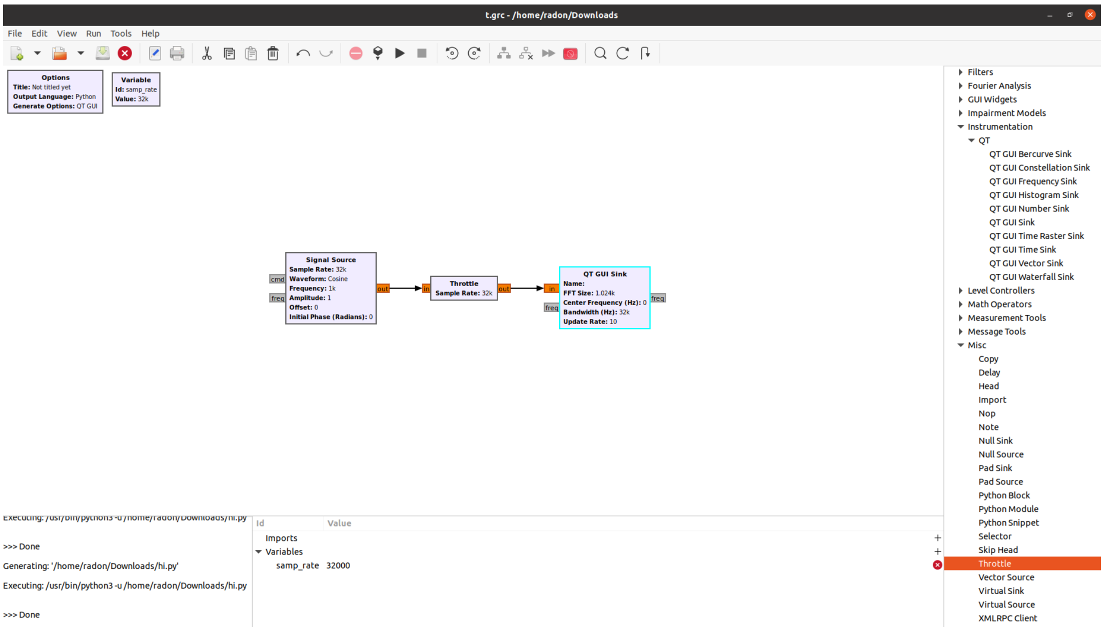

Fig. 7: Flow graph with throttle added

- (T1) **When setting the source signal to a 1kHz frequency, how low can you drop the sample rate?**
  (T2) **Experiment with this and see how the output changes as you drop below the Nyquist rate.**

D. Working with Audio I/O

Create the flowgraph shown in Figure 8.

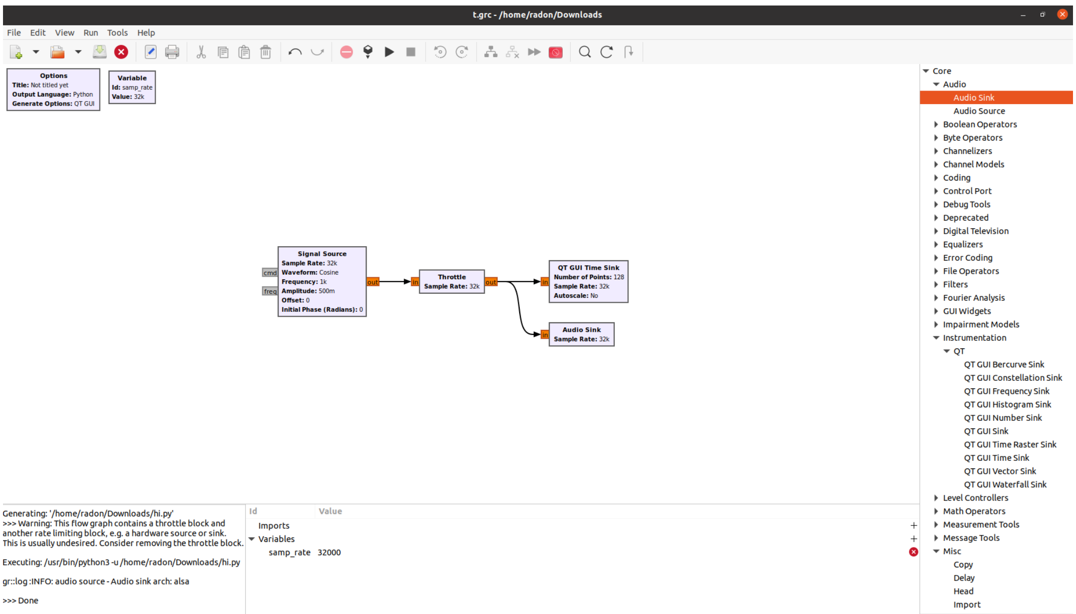

Fig. 8: Flow graph for audio I/O demonstration

- The **Audio Sink** is found in the **Audio** category. The Audio Sink block directs the signal to the audio card of your computer. Note that the sample rate is set to 32000, a sample rate that is usually, but not always supported by computer audio hardware.

- *Check your volume settings first*, and then generate and execute this flowgraph. The graphical display of the scope opens as before. However, now you should also hear the 1 kHz tone. **If you do not hear the tone, ensure that the output from the computer is connected to the speakers (or headset) and that the volume is turned up**. Experiment with changing both the overall sample rate in the flow graph as well as the sample rate in the audio sink to see how the tone is affected.

E. Multiple plots in one graph

For this assignment, you need to create graphs for your final report, and use them to reason why your implementations are correct.
In order to do this, it is useful to have the input and output signals in one plot. Double click on a placed GUI Sink, go to field '*Number of inputs*', and set it to 2. Your scope sink can now handle 2 inputs, as seen in Figure 9. The output is displayed in Figure 10.

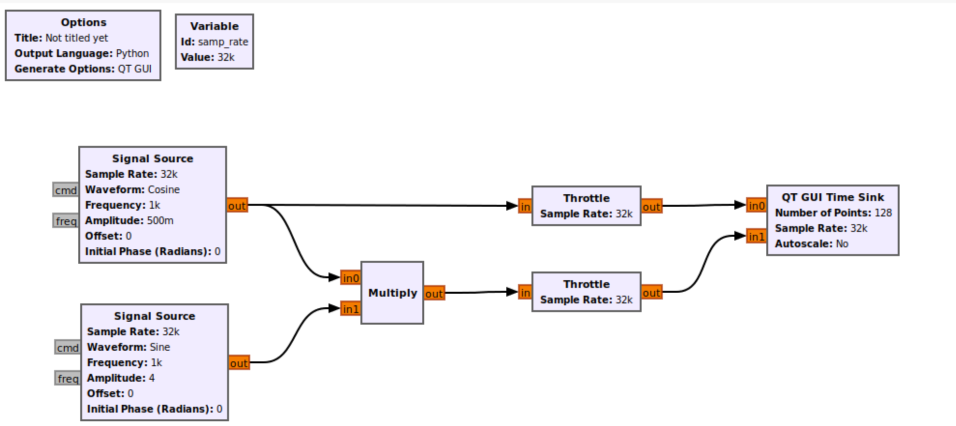

Fig. 9: Example graph leading input and output signals to sink

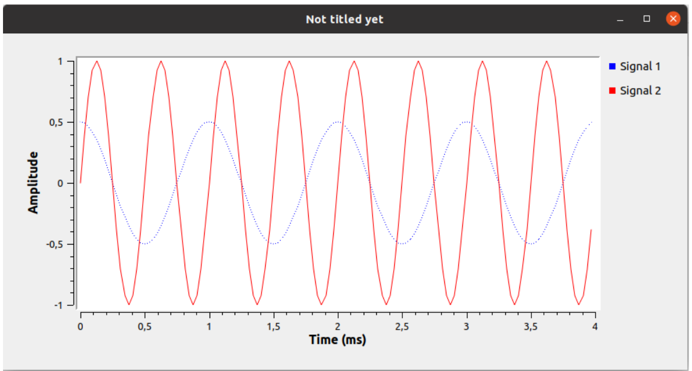

Fig. 10: Output plot with multiple input signals

F. Math Operations

- Construct the flow graph in Figure 11. Set the sample rate to 32000. The two Signal Sources should have frequencies of 1000 and 800, respectively. The Add block is found in the Math Operators category.

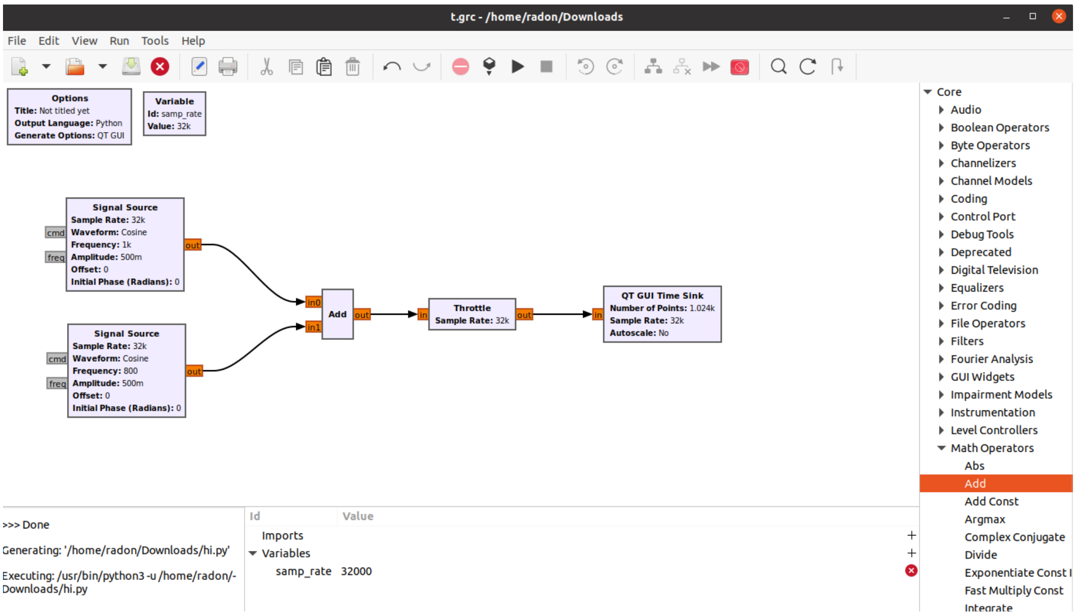

Fig. 11: Flow graph with Add block

- Generate and execute the flow graph. On the Scope plot you should observe a waveform corresponding to the sum of two sinusoids.
- (T3) **Replace the Add block with a Multiply block. What output do you expect from the product of two sinusoids? Confirm your result on the Scope display.**
- Note that the other math operations exist under the Math Operators category and experiment with a few to see if the result is as expected.

III. Tasks

Using the above tutorial, you should implement the following signal modulation schemes. For every one of these tasks, you have to add the following in the report:

1) Explain how you created the flowgraph and the formulas and transformations you used (can be found in the slides)
2) Add a screenshot of the flowgraph
3) Add a screenshot of the flowgraph output
4) Using the flowgraph output, explain why your implementation is correct

For more information about the report, grading, and what to hand in, see the related section below.

- Analogue modulation schemes
  - Frequency Modulation (FM) (see reference implementation below)
  - (T4) **Amplitude Modulation (AM)**
  - (T5) **Phase Modulation (PM)**
- Digital modulation schemes
  - (T6) **Frequency-shift keying (FSK)**

- – (T7) **Amplitude-shift keying (ASK)**
- – (T8) **Phase-shift keying (PSK)**

  Please note that for generating digital signals you should use either **Vector Source** block under **Misc** category or use a **Square** wave in a **Signal Source** block.
- Finally, show the resemblance of
  - – (T9) **FM and PM modulations.**
  - – (T10) **FSK on square waveforms and PSK modulations on triangle waveforms, respectively.**

A. *Reference implementation of FM*

To get a hint how to implement the modulation schemes, in this section, a step-by-step implementation of Frequency Modulation (FM) is provided. Recall from FM theory, for a sinusoidal modulating wave $x_m(t) = A_m \sin(2\pi f_m t)$, and a carrier wave $m(t) = A_c \cos(2\pi f_c t)$, the FM wave is $s(t) = A_c \cos(2\pi f_c t + A_m \cos(2\pi f_m t))$.
Sadly, there is no convenient way to perform $\cos(x+y)$ in GNU Radio Companion. However, we know $\cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y)$. With this information, we can expand our formula to

$$s(t) = A_c \cos(2\pi f_c t)\cos(A_m \cos(2\pi f_m t)) - A_c \sin(2\pi f_c t)\sin(A_m \cos(2\pi f_m t)). \quad (1)$$

Figure 12 shows the flow graph of FM based on Equation 1 with f_m = 2 KHz , A_m = 8, f_c = 25 KHz, A_c = 1, and using a sample rate f_s = 2 MHz.

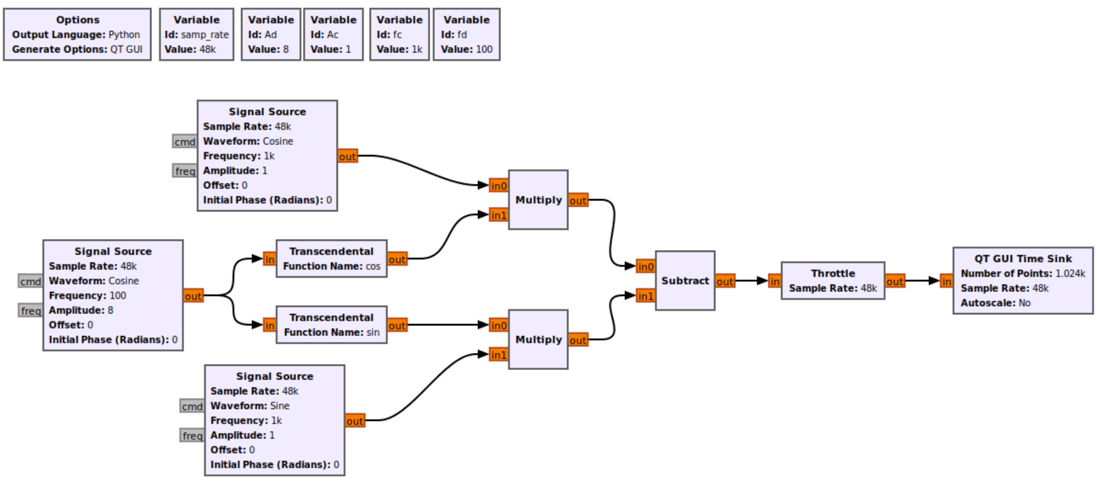

Fig. 12: Flow graph of Frequency Modulation

For the blocks used in Figure 12, a brief explanation is provided in the following:

- In Figure 12, the first block is used for creating the signal $x_m(t) = A_m \cos(2\pi f_m t)$.
- Transcendental Blocks are used to obtain the sine and cosine of $x_m(t)$ and create $I(t) = \sin(A_m \cos(2\pi f_m t))$ and $Q(t) = \cos(A_m \cos(2\pi f_m t))$ in equation (1).
- Then, I(t) and Q(t) are multiplied by $A_c \sin(2\pi f_c t)$ and $A_c \cos(2\pi f_c t)$ which are created using two signal source blocks.
- We use a subtract block to subtract $A_c \cos(2\pi f_c t) * Q(t)$ from $A_c \sin(2\pi f_c t) * I(t)$.
- Finally, the Throttle sample rate is set to samp_rate and GUI Sink is used to see properties of FM signal with Sinusoidal message. You can change the carrier frequency and message frequency and amplitude and see the changes. Figure 13 shows the result for the default parameters.

B. *Teams, Submission and Grading*

You can work on this assignment in a team of **at most two persons**.

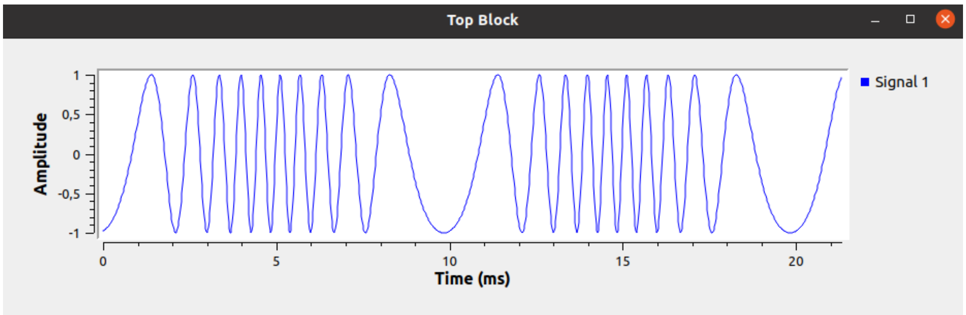

Fig. 13: Output for FM flowgraph, using default parameters

a) Submission: Teams need to submit the following items (only 1 member per team has to do this):

- .grc files answering questions T4, T5, $\cdots$, T10.
- A well-organized report in PDF-format, answering the three questions of tutorial T1, T2, T3. For T4, T5..., T10, you have to:
  1) Explain how you created the flowgraph and the formulas and transformations you used (can be found in the slides)
  2) Add a screenshot of the flowgraph
  3) Add a screenshot of the flowgraph output
  4) Using the flowgraph output, explain why your implementation is correct

Please create a "gzipped tar" archive of your submitting items and name it as **assignment1-sXXXXXXX-sYYYYYYY.tar.gz** in which XXXXXXX and YYYYYYY are your student IDs. Submit the tar-archive by e-mail to "s.f.alvarez.rodriguez@umail.leidenuniv.nl". Deadline: **Friday 12 March, 2021**, at **23:59**.

b) Grading Scheme: For your benefit, we publish the grading scheme.

```
 Grade = points/10

T1 (3)
    [+3] Report explains where the deformation starts.

T2 (3)
    [+1.5] Report explains what happens below nyquist rate
    [+1.5] Report explains what happens on nyquist rate

T3 (3)
    [+1.5] Student gives their intuition (whatever it is)
    [+1.5] Student shows mathematically what should happen

T4 (20)
    [+14] Correct implementation
    [+01] Variable blocks are used
    [+01] Screenshot of output in report
    [+01] Screenshot shows both input and output signals
    [+01] Students show that their implementation works
    [+02] Formula of their implementation

T5 (10)
    [+04] Correct implementation
```

```
    [+01] Variable blocks are used
    [+01] Screenshot of output in report
    [+01] Screenshot shows both input and output signals
    [+01] Students show that their implementation works
    [+02] Formula of their implementation

T6 (8)
    [+03] Correct implementation
    [+01] Variable blocks are used
    [+01] Screenshot of output in report
    [+01] Screenshot shows both input and output signals
    [+01] Students show that their implementation works
    [+01] Formula of their implementation

T7 (6)
    [+01] Correct implementation
    [+01] Variable blocks are used
    [+01] Screenshot of output in report
    [+01] Screenshot shows both input and output signals
    [+01] Students show that their implementation works
    [+01] Formula of their implementation

T8 (7)
    [+02] Correct implementation
    [+01] Variable blocks are used
    [+01] Screenshot of output in report
    [+01] Screenshot shows both input and output signals
    [+01] Students show that their implementation works
    [+01] Formula of their implementation

T9 (20)
    [+10] Students compare their implementations,
    and show the resemblance,
    and the single thing that makes the outputs different.
    [+06] Students show why this single difference exists
    [+02] Screenshot of output in report
    [+02] Screenshot shows input(s) and both output signals


T10 (20)
    [+04] Working PSK on triangular wave (with slides-specified amplitude)
    [+06] Students show that their implementations are equivalent
    [+06] Students show why this is the case
    [+02] Screenshot of output in report
    [+02] Screenshot shows input(s) and both output signals
```

c) Final Words: Please be aware that only two lab sessions are assigned for this assignment: This might not be sufficient to finish the above tasks. So, take some additional time during week to be able to finish the tasks and hand everything in on time.
For late submissions, we subtract 1 point per day past the deadline.