DSPIRA GnuRadio Lessons: Introduction and Simple Spectrometer Lesson 1: Building a Waveform Generator

GNURadio is very popular and robust software defined radio package. It is open source and is relatively easy to use. All "coding" is done using flowgraphs comprised of interconnected Digital Signal Processing (DSP) blocks. Most commonly used blocks come predefined as part of the software package. One can program one's own blocks as well.

Let's get Familiar – Building a Waveform Generator

For our first project we will build a program that generates a real-valued cosine signal with a frequency of 400 Hz, and outputs it to a speaker. We will also display the signal in a time domain graph and in a frequency domain graph. Then we will modify it so that we can adjust the frequency in real time.

Step 1. Open GNURadio

• To open GNURadio, open a terminal window. Type "gnuradio-companion" and hit Enter.



This opens GNU Radio Companion (GRC):

🖡 T 🖿 T 🛛 🗙 📝 🚍 😹 🖩 🗉 📩 🦘 🖉 🗙 🚱 🕲 🕲 🕲 🕲 🖉 🕒 🖉 😓 🕲 🕲	
Options ID: top block	▼ Core ▶ Audio
In: top_atok	► Boolean Operators
	► Byte Operators
	► byte operators
	Channel Models
	► Coding
Variable	Control Port
ID: samp rate	► Debug Tools
Value: 32k	► Deprecated
	► Digital Television
	► Equalizers
	► Error Coding
	► FCD
	File Operators
	► Filters
	► Fourier Analysis
	► GUI Widgets
	 Of Widgles Impairment Models
	► Instrumentation
	► Level Controllers
	Math Operators
	 Math Operators Measurement Tools
	Message Tools
	 Misc
	► Modulators
	 Networking Tools
	► NOAA
	► NOAA ► OFDM
	 Packet Operators Pager
	Peak Detectors Resamplers
Value	Resampters Stream Operators
	+ Stream Operators
Imports Variables	+ Symbol Coding
	Symbol Cooling Synchronizers
samp_rat 32000	Synchronizers Trellis Coding
	Type Converters
	UHD
	► Variables
	 Variables Waveform Generators
	Waverorm Generators ZeroMQ Interfaces
	 ZeroMQ Interfaces (no module specified)
	(no module specified)

• The canvas window is mostly empty. On the right side of the canvas is a list of the block categories that are available. Click on a triangle next to a category to see what blocks are available in that category. Look under the *waveform generator* category and look for the **signal source** block.

Alternatively, we can click on the magnifying/looking glass to the top right and search for the block we need.

• Add the **signal source** block to the canvas by dragging it into the canvas window (or double-click):

			*untitled	- GNU Radio Co	mpanion		
e Edit View Run Tools Help							
🛛 🕶 🕶 🙆 🖉	<u>ل</u>	F Ê Î	$\land \lor \ominus$	♀ ▶ ■	004**	▶ 🙆 Q C (Ĵ -
Uptions Uptions Uptions Uptions Instant Anguage: Python enerate Options: OTGUI Unive: 324 Uptions: OTGUI Signal Source Sample Rate: 324 Uptions: 0 Signal Source Sample Rate: 324 Uptions: 0 Instant Phase (Radians): 0							 > Fourier Analysis > GUI Widgets > Impairment Models > Instrumentation > IQ Balance > IQ Correction > Level Controllers > Math Operators > Message Tools > Misc > Modulators > Networking Tools > OrDM > Packet Operators > Resamplers > Stream Operators > Stream Tag Tools > Symbol Coding > Symbol Coding > Trellis Coding > Trellis Coding > Wideo > Wariables > Video > Waveform Generators Constant Source Fast Noise Source

Some Helpful Information About GNURadio

Block Port Colors

Notice that the **Signal Source** block has one or two gray ports on the left and a blue one on the right. The color of a port indicates the type of data generated for an output port or the type of data accepted for an input. The most common data types that we will use are:

- Orange is for "Float" data types, which are real-valued 32-bit floating point data samples.
- Blue is for "Complex" data types, which are complex-valued 32-bit floating point data samples. [Electrical engineering and digital signal processing often use complex functions and data.]

Connecting Blocks

To connect two blocks, click on the output port of the first block and then click on the input port of the second block. An arrow will appear connecting the two ports. For the flowgraph to work both ports must use the same data type (i.e., both ports must be of the same color). If they are of different types, then the arrow of the connection will be red instead of black, indicating a problem.

NOTE: If you see any red arrows or red writing in a flowgraph you will not be able to run the flowgraph until the offending condition has been fixed.

Editing a Block

To edit a block, double-click on it. A window will open for editing the block.

Back to building a waveform generator:

	Properties: Signal Source 🛛 😵
General Advar	ced Documentation
Output Type	float 👻
Sample Rate	samp_rate
Waveform	Cosine 👻
Frequency	1000
Amplitude	1
Offset	0
Initial Phase (Radians)	0
	OK Cancel Apply

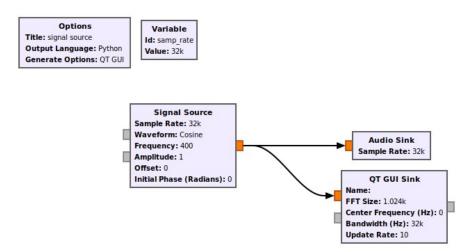
Step 2. Change the "Output Type" in the **signal source** block to "float" and hit Ok.

The output terminal on this block should now be orange.

Step 3. Let's add a speaker and graphical display to the signal output:

- **Change the Signal Source frequency:** Double-click the **Signal Source** block. Then type in "400" for the frequency value.
- Add a speaker to the output: Choose "Audio" under "Instrumentation" (or just simply search for "Audio Sink") and click and drag (or double-click) an **Audo Sink**. Connect this to the output of the **Signal Source** block.
- Add a graphic output: Choose "QT" under "Instrumentation" (or just simply search for "QT GUI Sink") and add a **QT GUI Sink** to the canvas. This block will allow you to see the waveform at the input in the frequency as well as in the time domain. Connect this to the output of the **Signal Source** block. Be sure that all of the data types are properly set.
- The program won't run until you rename it. Double-click the **Options** block in the upper left corner, and give it a title.

The program now should look like the following:



- Save the file: Save the file to an appropriate folder.
- **Run the program:** Now you can run the flow graph by clicking on the green triangle above the canvas or by clicking "Run" on the menu bar. To view the graphs, you can choose between the "Frequency Display" and the "Time Domain Display" tabs. Study some of the other graphical displays too.

Let's Make Some Modifications

Let's add a few features to the program. We want to be able to change the frequency and amplitude of the source signal while the program is running. To do this, we will change their values in the **signal source** block to variable names, such as "freq" and "amplitude". The **QT GUI Range** is a block that can be used to define a variable that can be changed while the program is running.

• Add a frequency variable: Search for "QT GUI Range" and double-click on **QT GUI Range** to add it to the canvas. This block will allow you to define and control a variable.

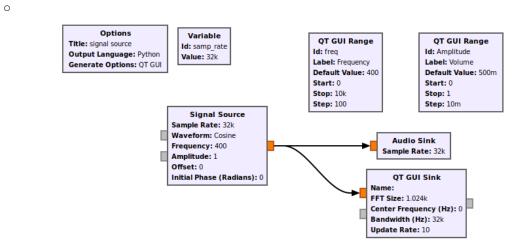
Set the *Id*, *Default Value*, *Start*, *Stop*, and *Step* as shown in the diagram below:

Properties: QT GUI Range 🛛 😣				
General	Advanced	Documentation		
Id	freq			
Label	Frequ	iency		
Туре	float	•		
Default Value	400			
Start	о			
Stop	1000	0		
Step	100			
Widget	Cour	nter + Slider 🔻		
Minimum Leng	th 200			
GUI Hint				
		OK Cancel Apply		

- In the **Signal Source** block, change the Frequency value to "freq" to match the variable we defined in the **QT GUI Range** block.
- Speaker volume Control:
 - Add a **QT GUI Range** variable called "amplitude", with a default of 1 and a range from 0 to 1, step 0.01.
 - In the **Signal Source** block, change the Amplitude value to "amplitude" to match the variable we defined in the **QT GUI Range** block.

Properties: Signal Source 🛛 😣				
General Advar	ced Documentation			
Output Type	float 👻			
Sample Rate	samp_rate			
Waveform	Cosine 👻			
Frequency	freq			
Amplitude	amplitude			
Offset	0			
Initial Phase (Radians)	0			

• Your canvas should look like the following:



• **Re-run the program:** Change the frequency and amplitude sliders, and observe changes in the Time domain graph and the Frequency domain graph.