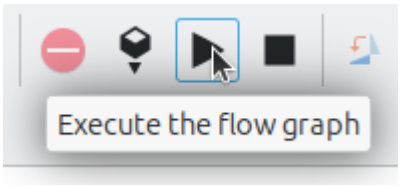
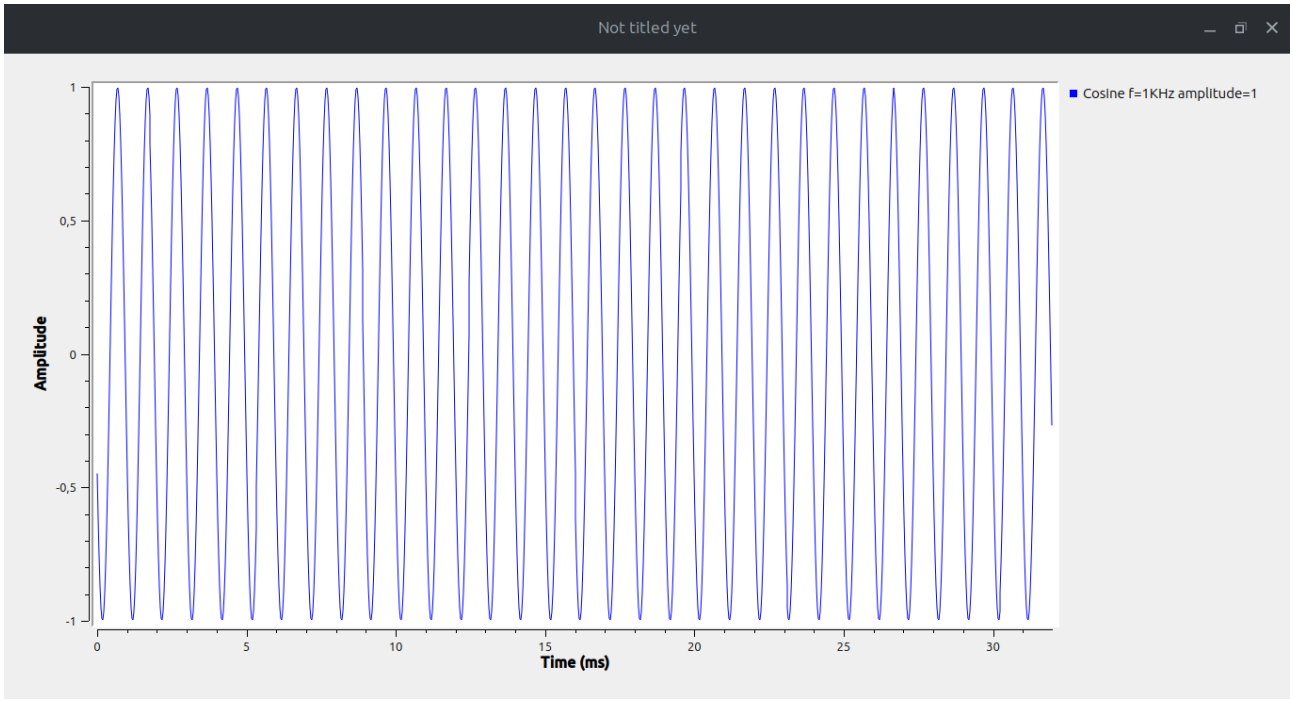
HY330 – Telecommunication Systems
Chris Papastamos | csd4569
Assignment 1

Exercise 1



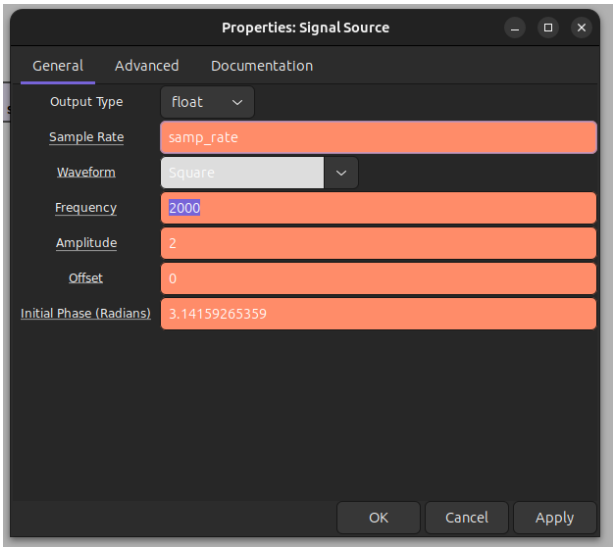
First up, after loading the given .grc file I executed the flowgraph and got the following output



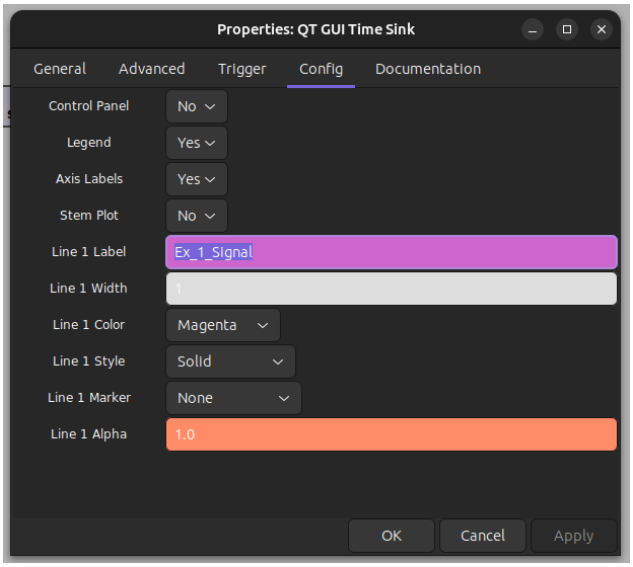
This is the output of the flowgraph which is displaying the output of the flow sink since it is the only outputting block in the flowgraph. In the graph we can see a cosine wave, shifted by a phase of 90 radians, a frequency of 1000MHz and an amplitude of 1.

Lets now change the Sample Source and the Time Sink and observe the changes to the signal.

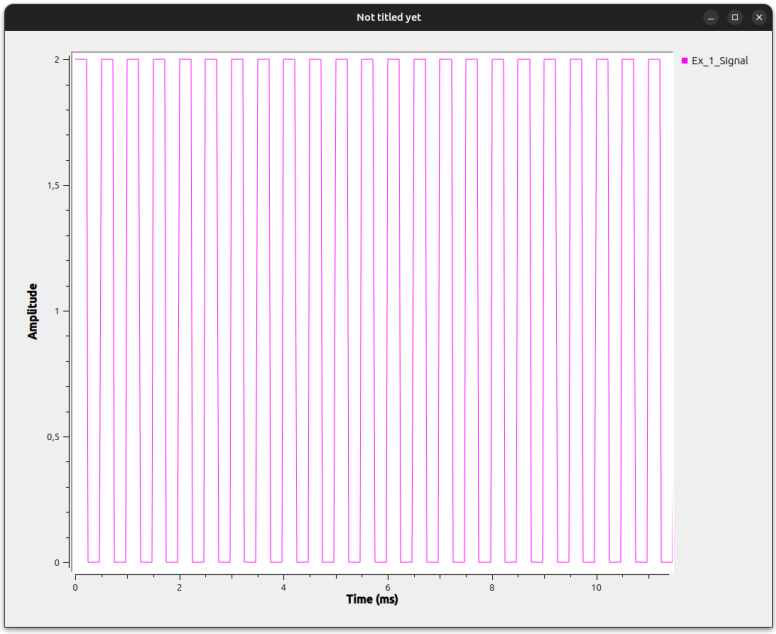
Starting off with the Signal source, I changed the Waveform to a square pulse, the frequency to 2000MHz, the amplitude to 2 and the initial phase to the value of pi (That way the signal is going to be offsetted by half a period)



Next up, I am going to change the config tab of the Time Sink block. The changes I made are the Line 1 Label and the Line 1 Color to Mnagenta.



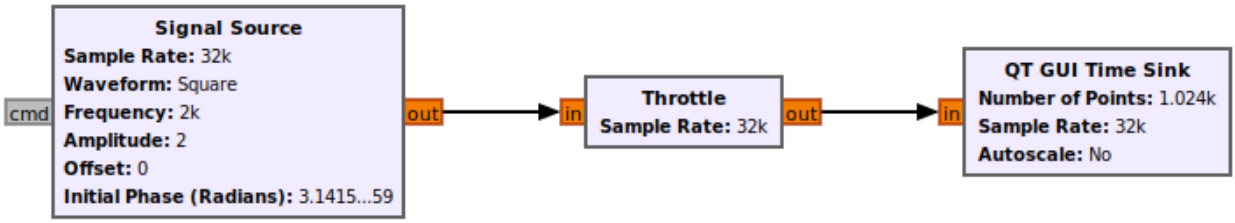
Lets now examine the output of the flowgraph



In the output we can actually observe the square pulse requested by the Signal Source. The signal is also shifted by half a period (originally the square pulse starts from 0) and the amplitude is 2 (0-2 because square pulse does not include negative amplitudes). We can also observe that in 2ms 4 periods of the signal occur, that means that every period is 0,5ms. From the frequency formula we have 1/0,5ms = 2GHz or 2000MHz.

The magenta color can also be seen in the signal line as well as the label on the top right legend.

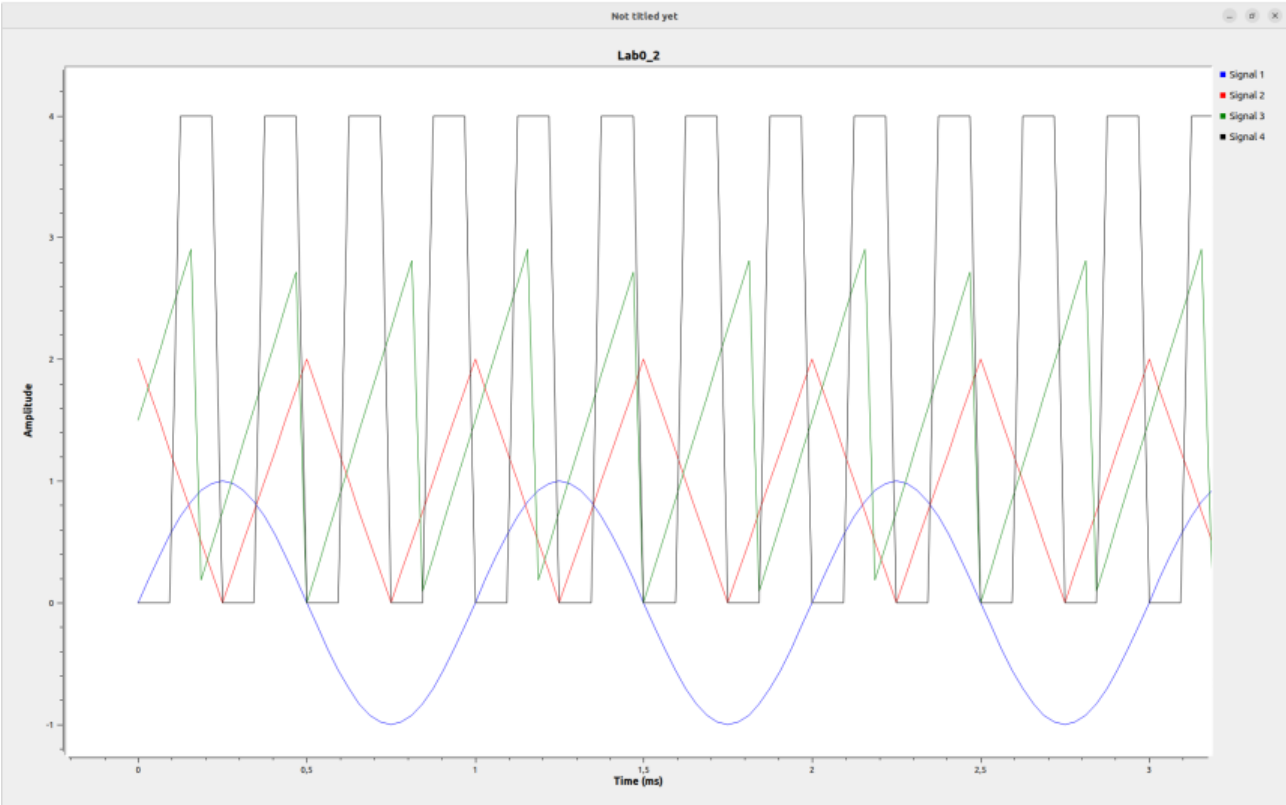
There is one remaining block that is not yet explained and that is the middle Throttle block:



What that block actually does is limits the rate at which the source (in our case the Signal Source) creates data. This is critical to the execution because if there is no throttle block present, there will be no specific sampling rate which will consume the whole CPU core creating noise that can be observed in the output

Exercise 2

Starting of with this exercise I am going to analyze the given output:

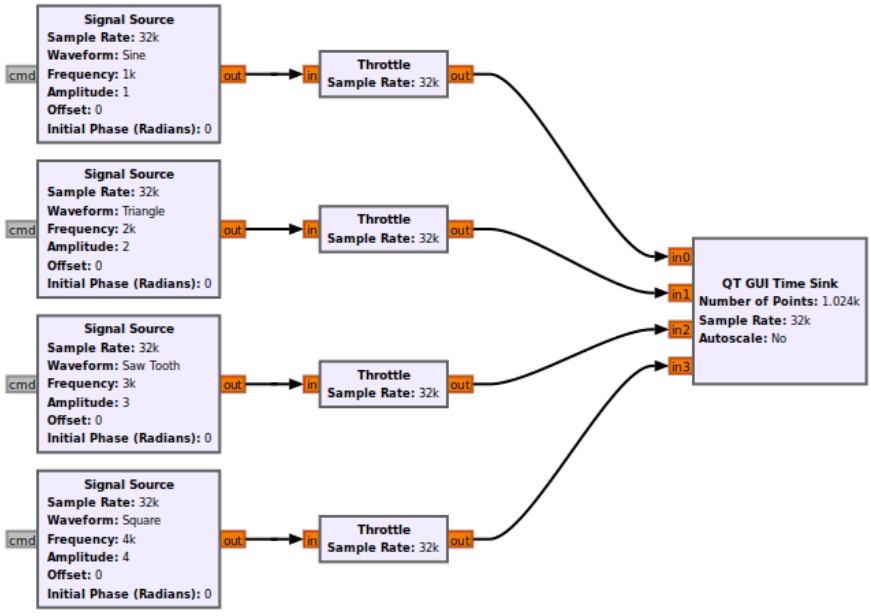


It is pretty clear what every signal is so lets analyze separately:

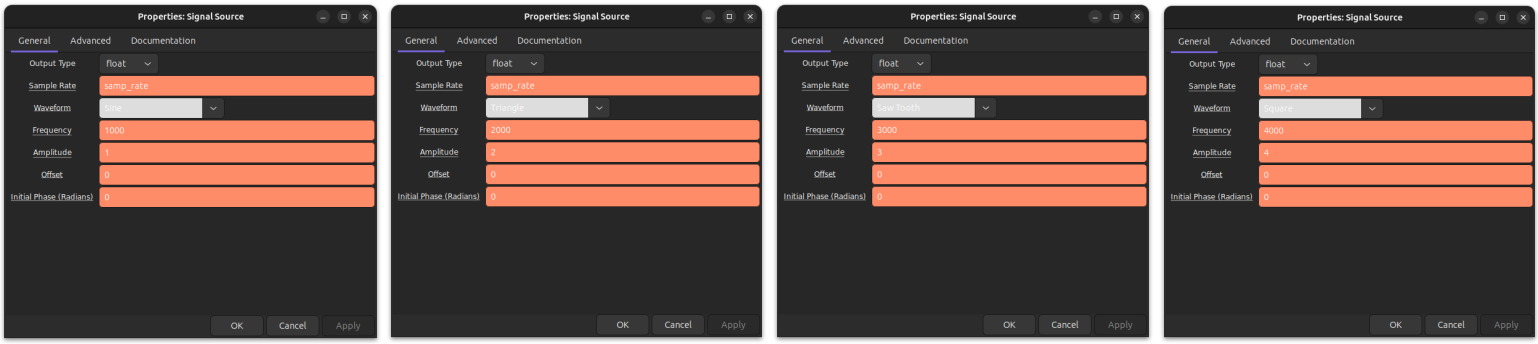
Signal 1: This signal is a Sine with an amplitude of 1 and a frequency of 1000MHz (=1/1ms)
Signal 2: This is a Triangle Pulse with an amplitude of 2 and a frequency of 2000MHz (=2/1ms)
Signal 3: This is a Saw Tooth signal with an amplitude of 3 and a frequency of 3000MHz (3/1ms)
Signal 4: This is a Square Pulse with an amplitude of 4 and a frequency of 4000MHz (=4/1ms)

(There seems to be a pattern on the amplitude and the frequency of the signals)

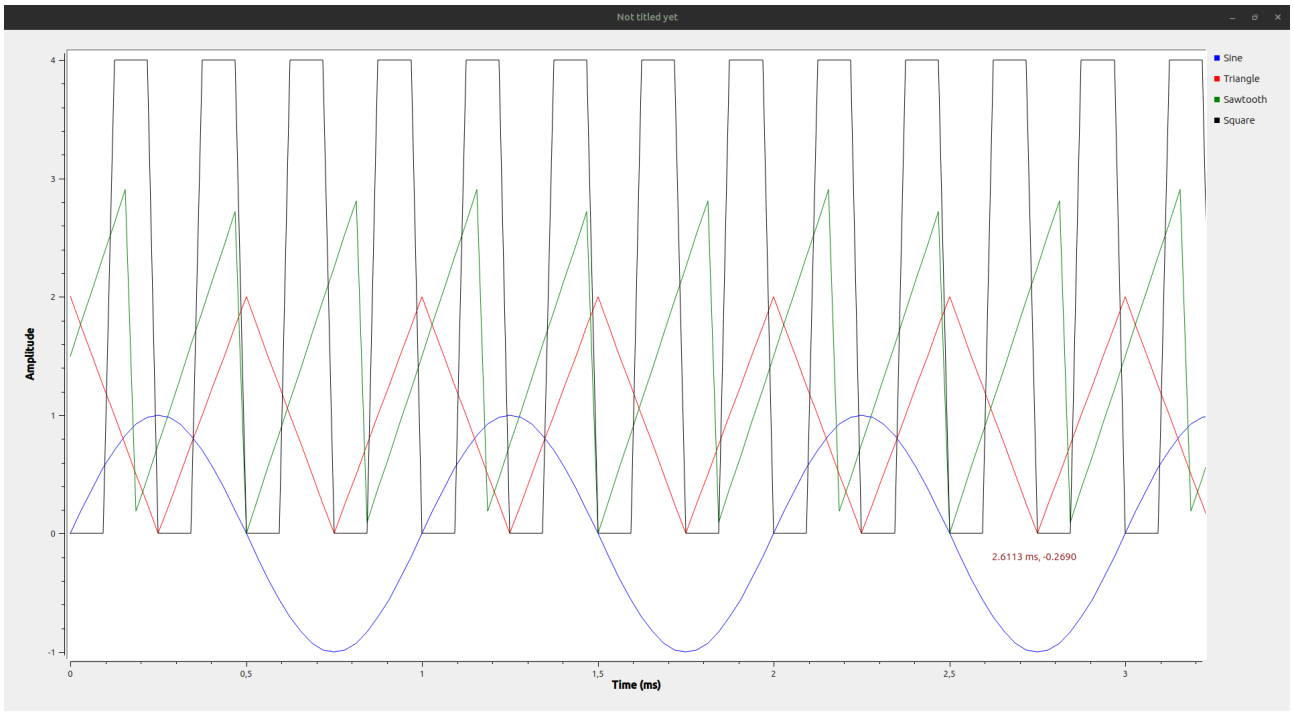
So for the creation of the same output, I created a flowgraph that can be seen on the right. It consists of four Signal Sources, all throttled (it isn't needed on all 4 of them), and a Time sink with named signal outputs.



The config of each Signal Source can be seen in the flowgraph but I will attach a picture of each configuration next just in case.



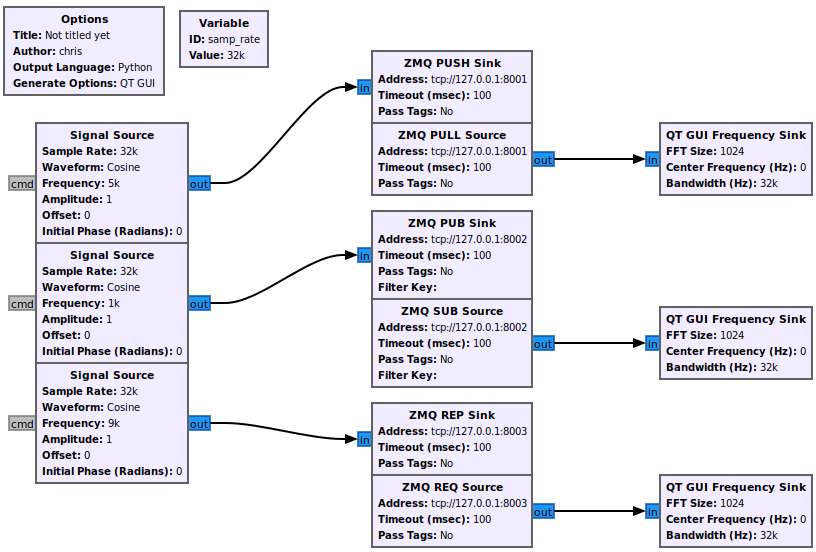
Lets now see the output of the flowgraph I created:



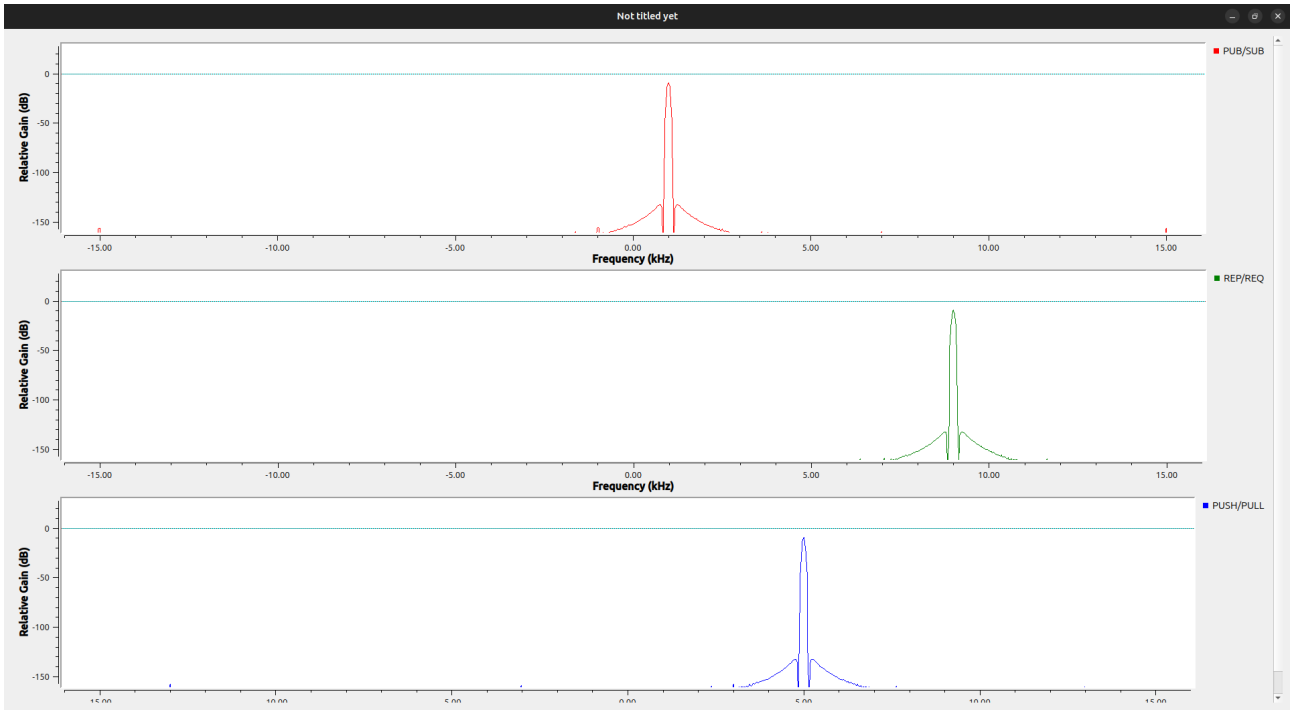
The output that can be seen above matches perfectly the given output that we had to recreate.

Exercise 3

Following the instructions, I created the following flowgraph



The three cosines (of different frequencies) are fed in to the respective system sink and the source of each system is fed into a Frequency Sink. For the address of each system I chose the loopback IP address and a port counting from 8001 and up (8001,8002,8003). The output of the flowgraph can be seen below:



Lets now analyze how each system works:

PUB/SUB: In order to explain the PUB(lish)/SUB(scribe) systems we have to introduce the meaning of a broker. A broker is an entity that exists between the two sides of the communication and each side is only referring to it. First up, the client is informing the broker that it wants to subscribe to an event. Later when the server publishes their data, the broker informs the client (often using an event handler) about the data and hands it to them. This way the client doesn't need to be blocked while waiting for the server to send data.

REQ/REPL: The REQ(uest)/REPL(y) system is probably the most common client-server model. It is pretty self explanatory, the client sends a request for data from the server and the server replies with the data that the client requested (assuming everything went well). This model tho has its weaknesses as the client has to block and wait for the server's response and given a asynchronous system the server may not be ready to reply with the requested data at the time of the request.

PUSH/PULL: The PUSH/PULL system is also a pretty self explanatory one. The server pushes its data to a source and when the client is ready, it pulls the data from that same source. This system is also pretty useful because it allows for asynchronous communication.