Streaming processed data from the E31x with GNU Radio and ZMQ

Contents

AN-335

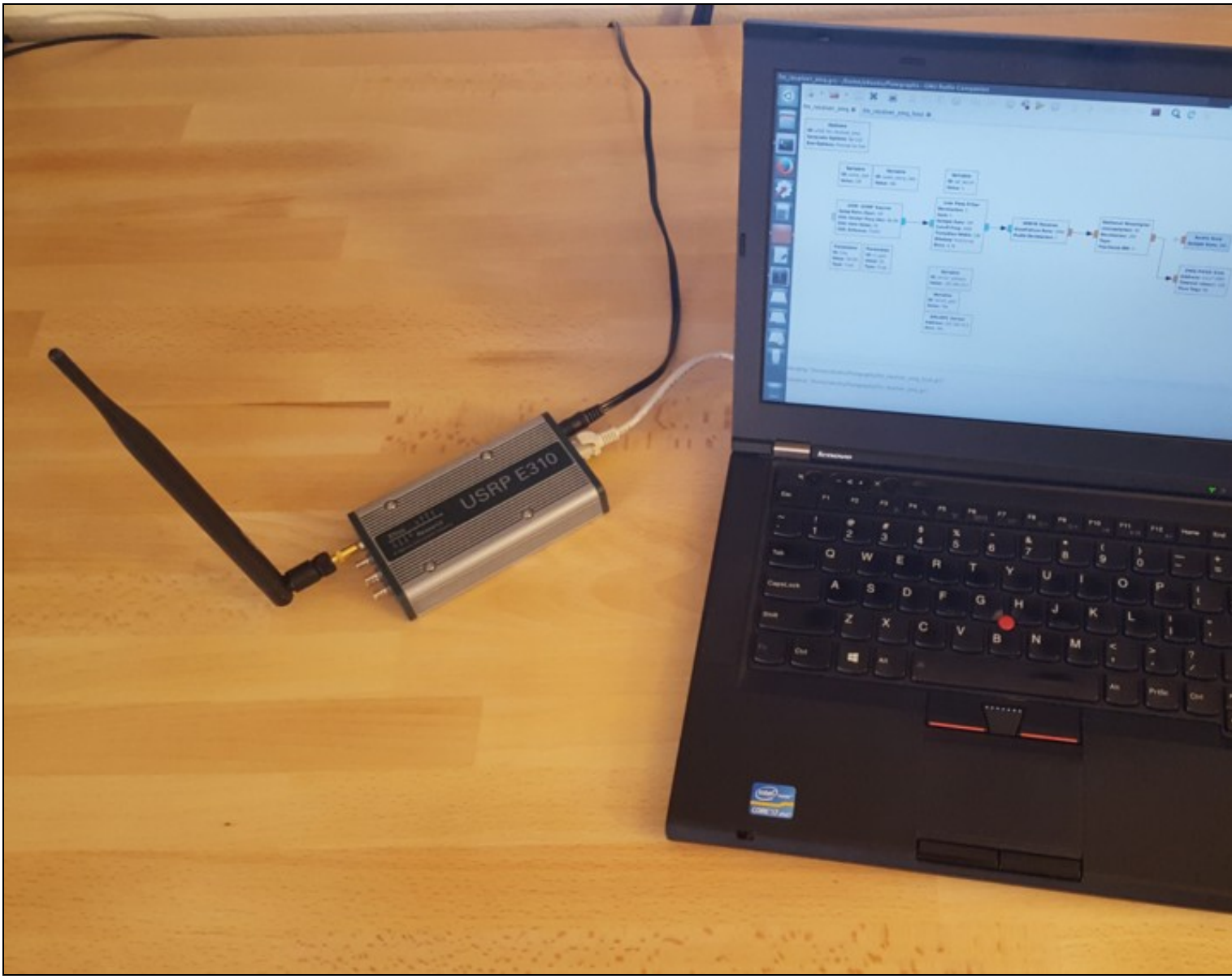
This application note will demonstrate using the USRP E310 to remotely stream processed data to a host machine.

The USRP E310 being an embedded USRP is designed to process data locally, unlike other USRP models which will stream samples to the host machine for processing. The USRP E310 is built around the Analog Devices AD9361 transceiver and Xilinx Zynq 7020 FPGA, which includes an ARM processor. Running an Open Embedded Linux distribution it supports UHD, GNU Radio and RFNoC for processing.

This application note will focus on using GNU Radio to bring in the stream of samples and perform the digital signal processing to demodulate a commercial wide-band frequency modulated radio station (WBFM) locally on the USRP E310. The demodulated audio stream will then be sent via the ZeroMQ (ZMQ) interface to a host machine, where it then will be played on the host's audio system / speakers. This application note will also demonstrate using the XMLRPC blocks of GNU Radio to adjust variables within a running flow graph on the remote E310.

- Ethernet cable
- Antenna
- USRP E310/E312/E313
- Host computer with GNU Radio installation

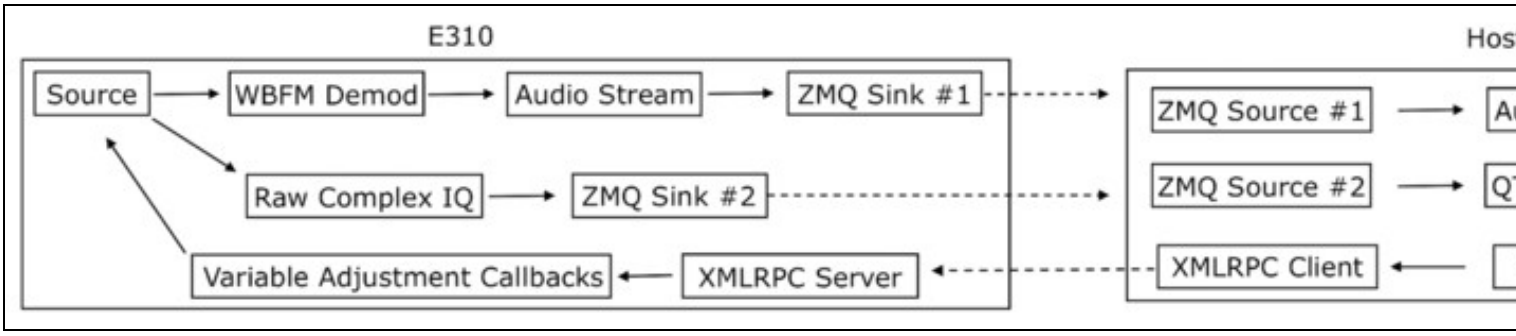
Note: The GNU Radio installation will need to have the `gr-zeromq` component enabled. For details on installing GNU Radio, please see the step-by-step guides at the Building and Installing the USRP Open-Source Toolchain (UHD and GNU Radio) on Linux, OS X and Windows Application Notes.



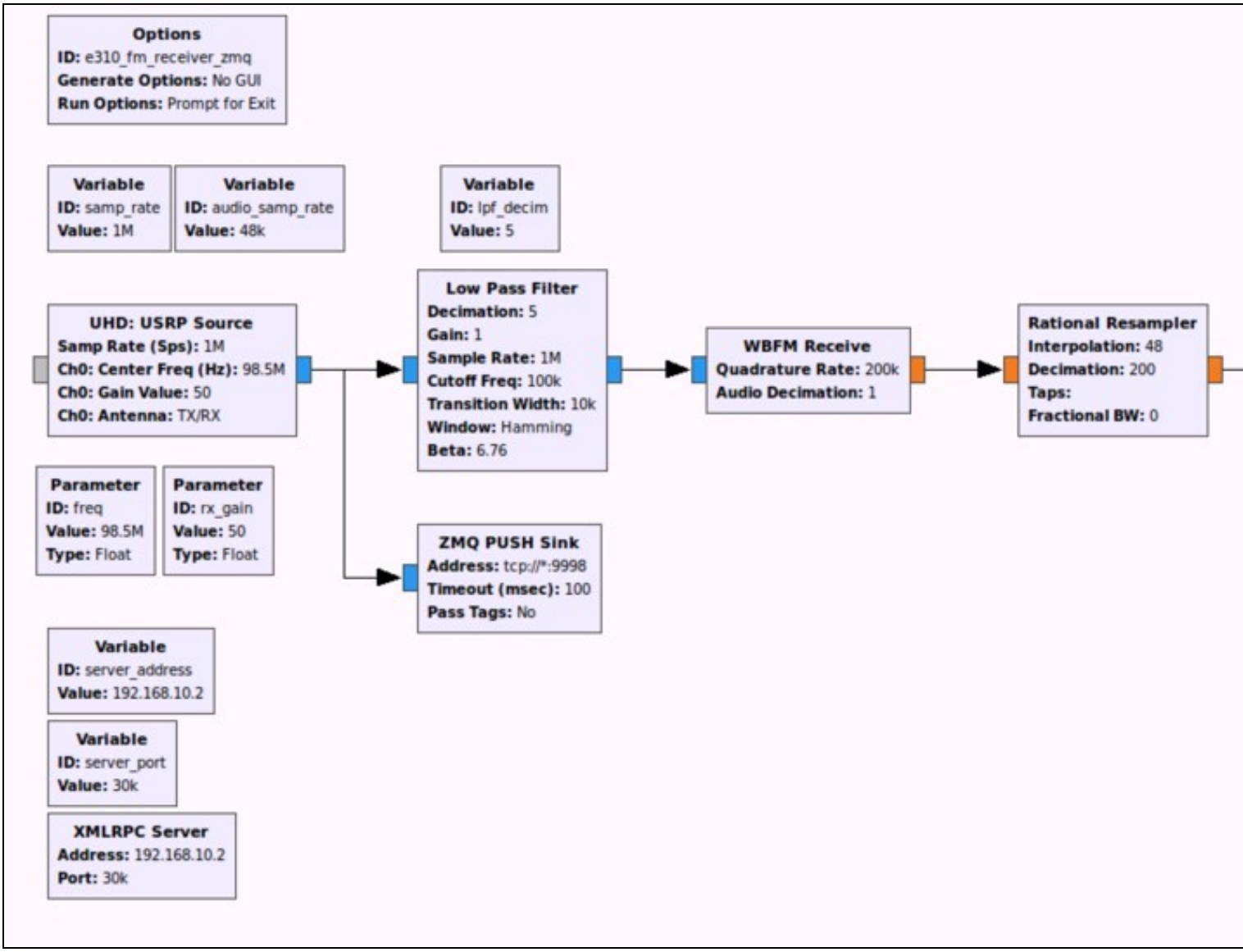
The block diagram below shows the high-level overview of the complete application. The flow graph running on the USRP E310 will demodulate a WBFM radio station, and then stream the audio over a ZMQ Sink to the host computer, where it will then be played over the host's speakers via the Audio Sink.

This application will also stream the complex IQ samples to the host via a second ZMQ Sink to produce a FFT on the host computer for visual reference only.

On the GNU Radio flow graph that is running on the host computer, there are QT GUI widgets to adjust the Frequency and RF Gain, that will be sent via XMLRPC commands to the E310, to adjust the variables within the running flow graph. The Audio Gain GUI widget will locally adjust the audio volume on the host computer.

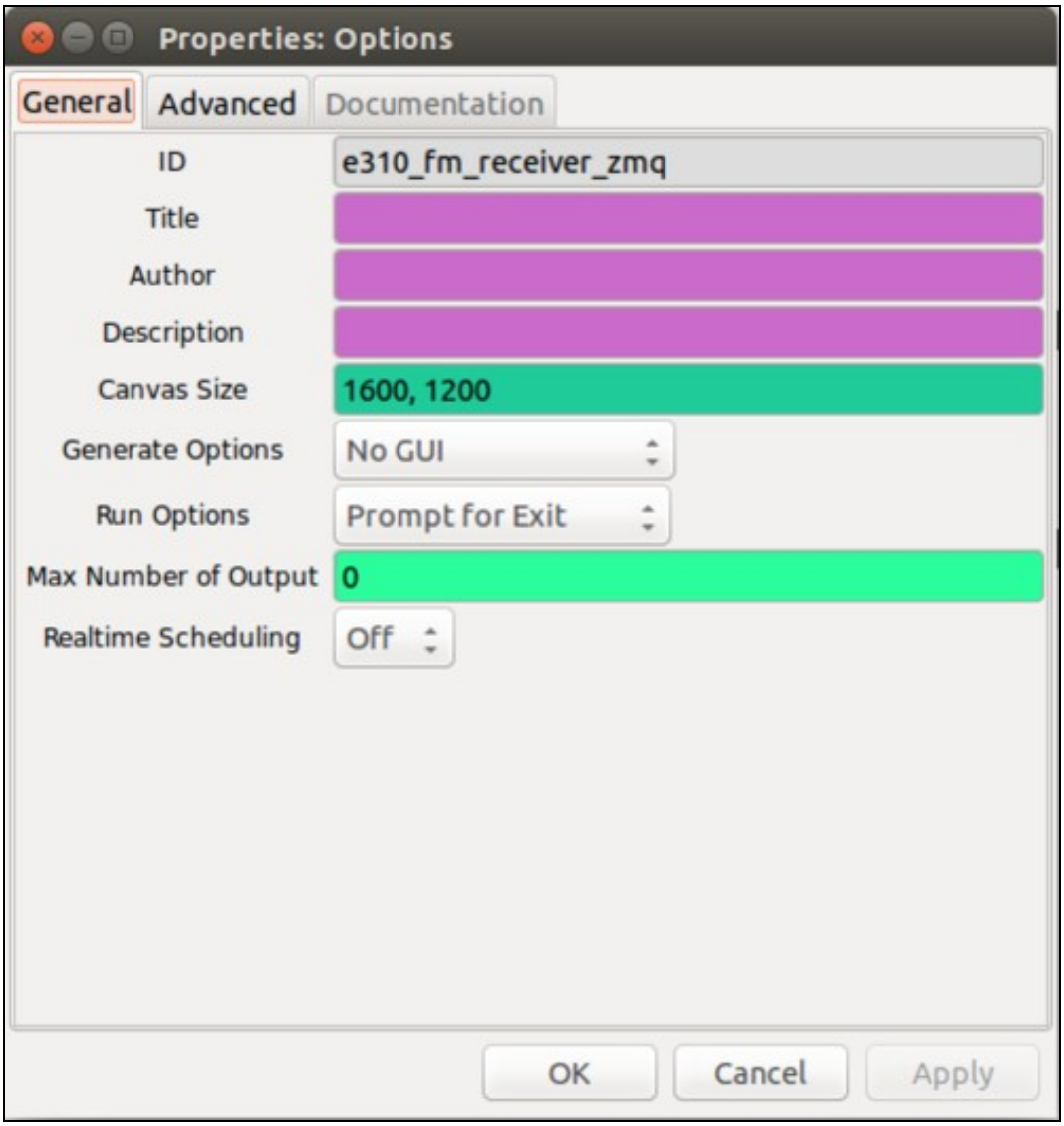


The figure below shows the flow graph that will run on the E310. There are several important settings to note.

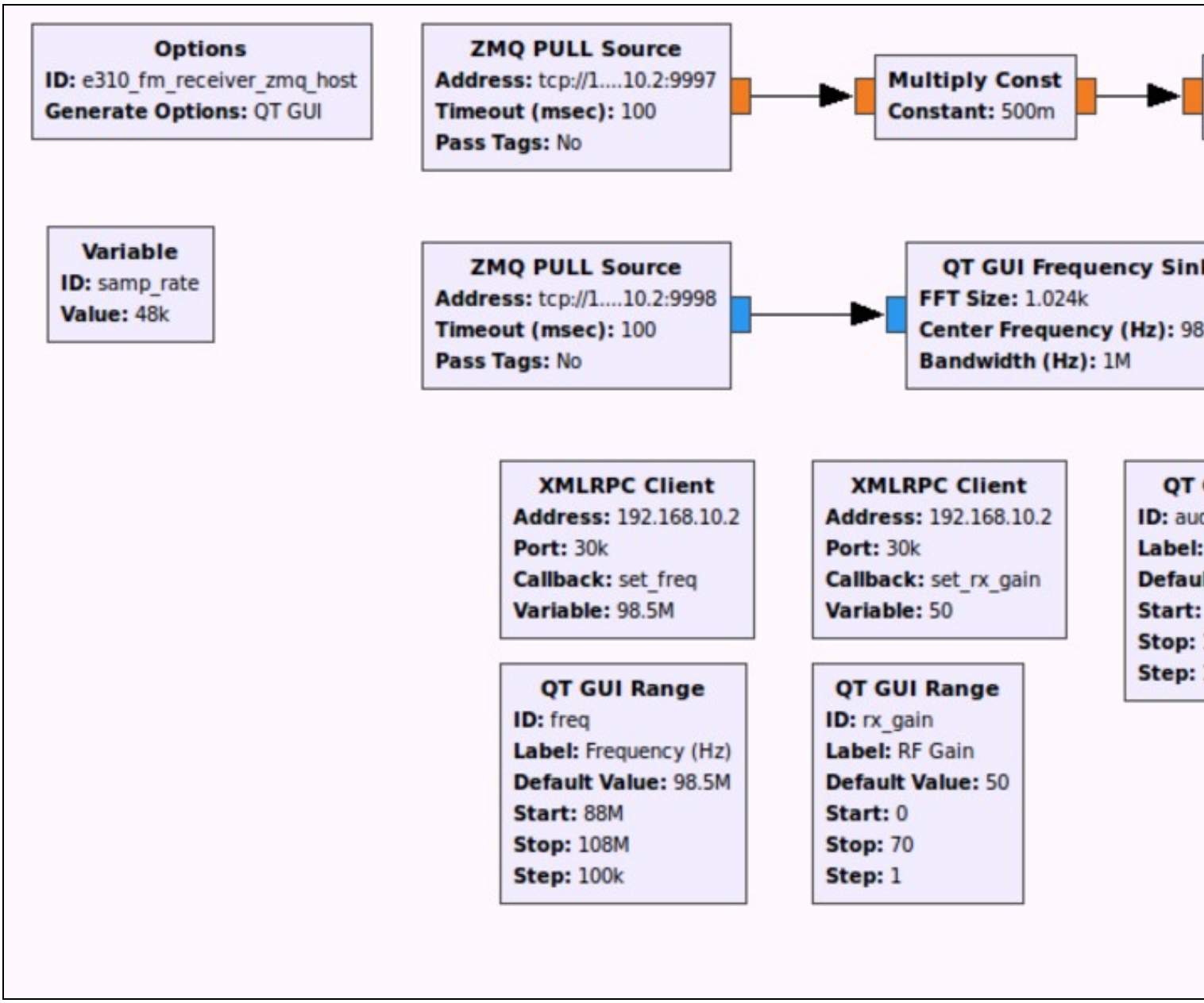


Within the `Options` block:

- Generate Options: No GUI
- Run Options: Prompt for Exit

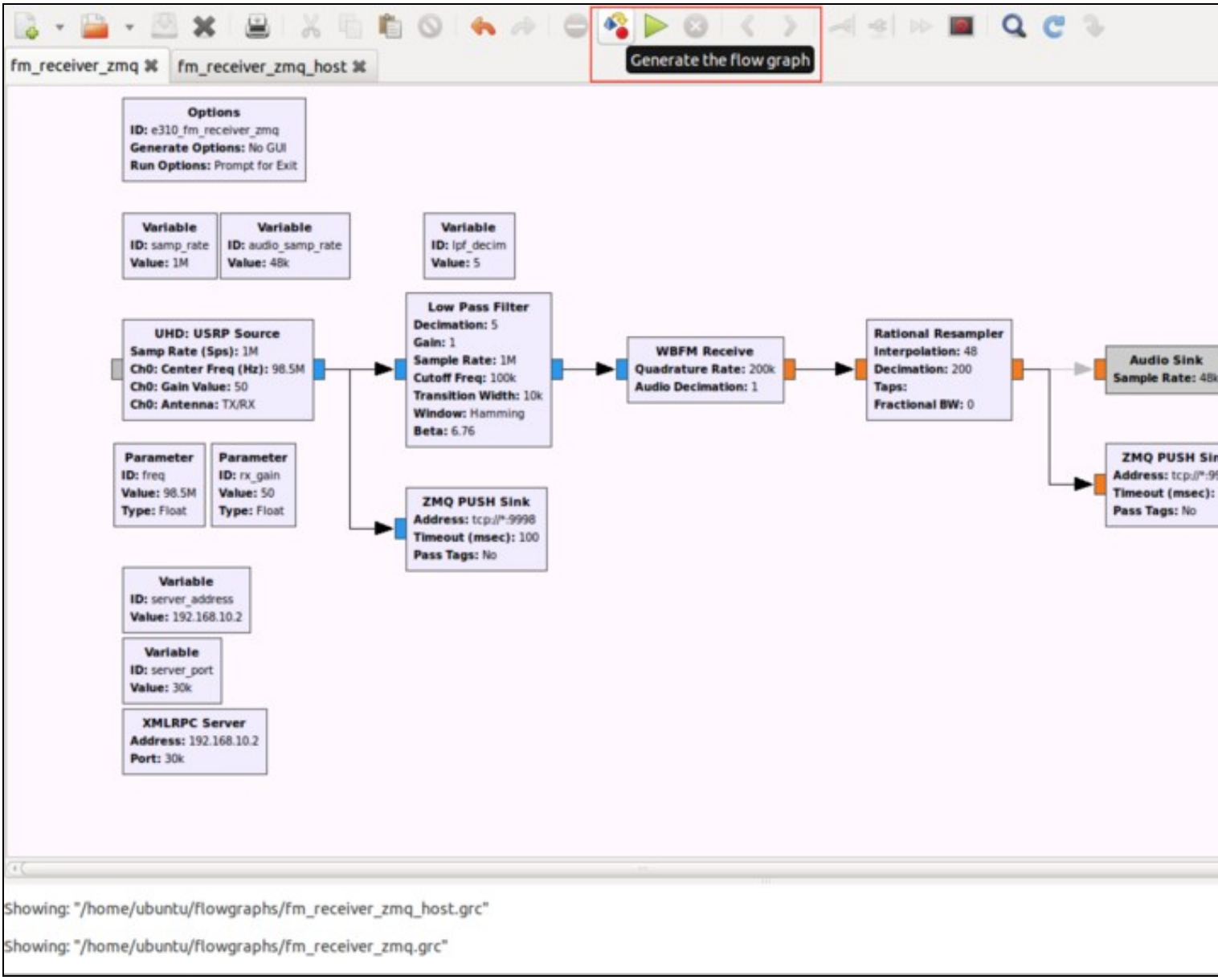


The figure below shows the flow graph [fm_receiver_zmq_host.grc] that will run on the host computer.

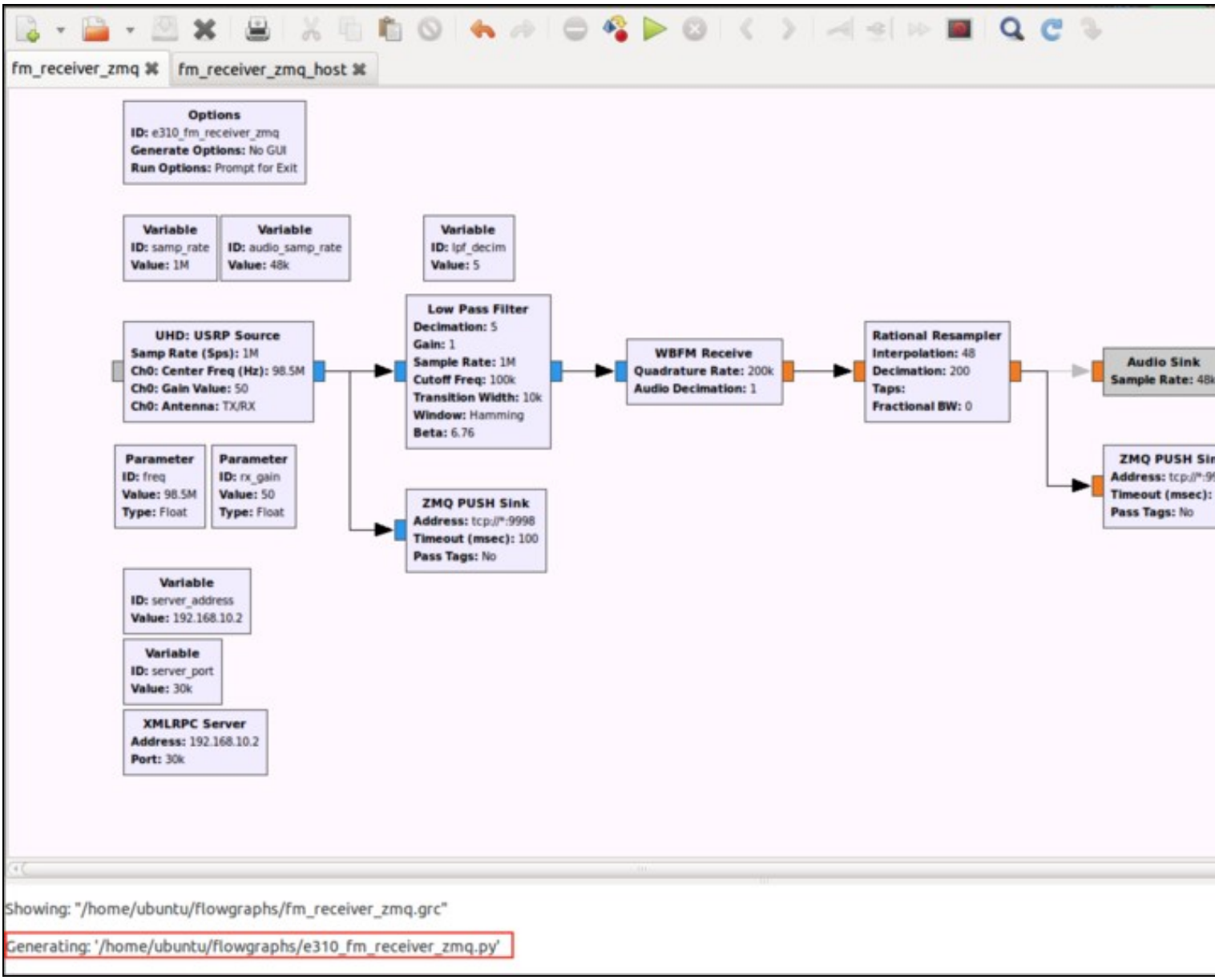


Before you're able to run the flow graph on the USRP E310, you will first need to generate the Python file and transfer it to the USRP E310.

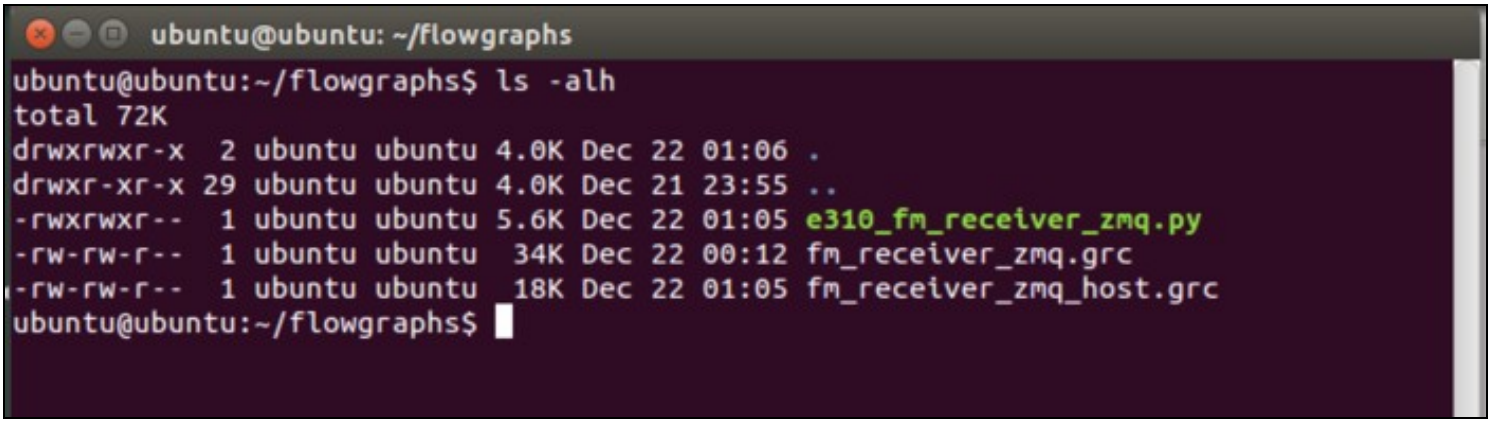
To generate the Python application code from the flow graph is done by using the Generate button with GNU Radio as shown in the figure below.



Note: After running the Generate function, within the console of GNU Radio Companion, it will print the location of the generate Python file. The filename will match the value of the `ID` setting within the `Options` block.

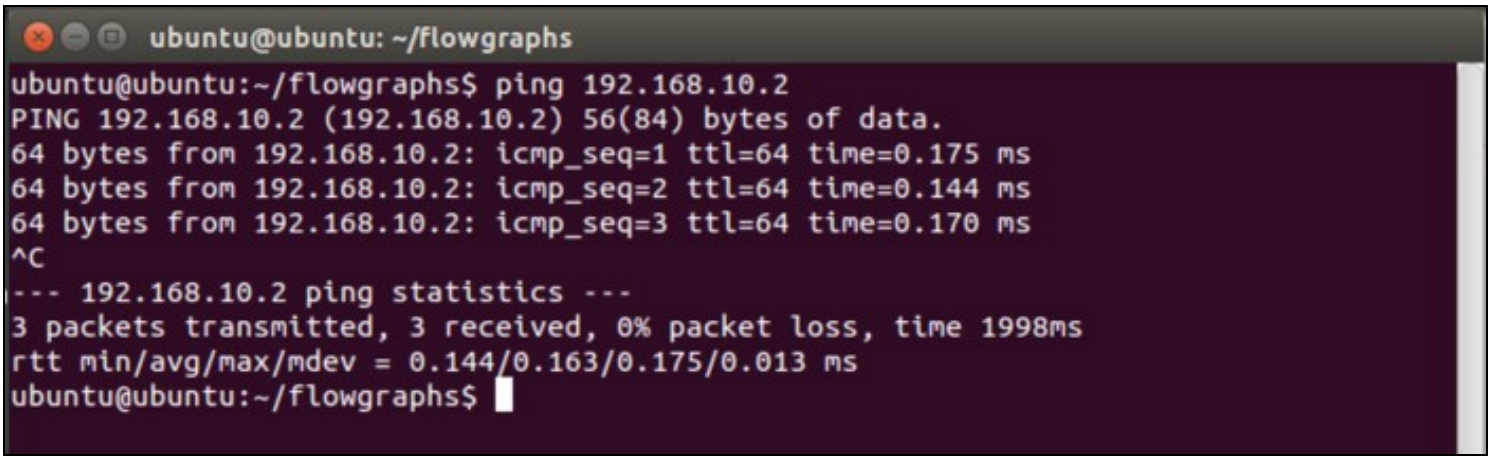


Open a new Terminal window, and navigate to the directory where the Python file was generated.



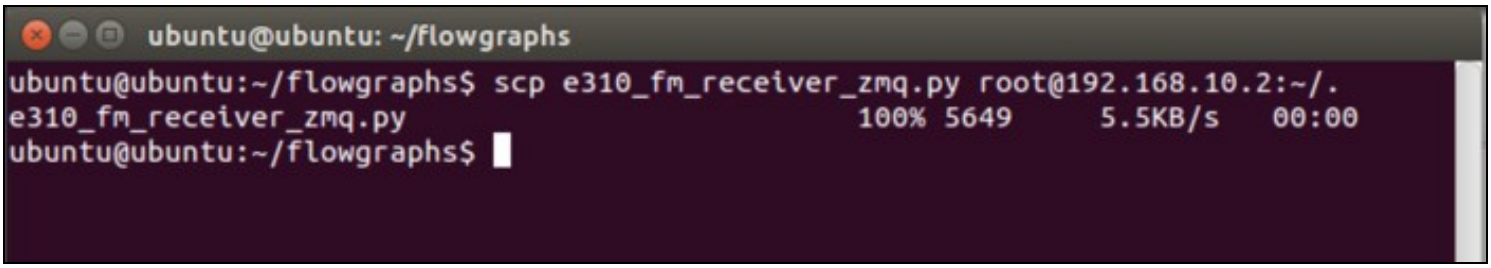
Before attempting to copy the file the USRP E310, ping it to verify you're able to communicate with it. **Note:** This application note assumes the IP address of the USRP E310 is `192.168.10.2`.

```
$ ping 192.168.10.2
```



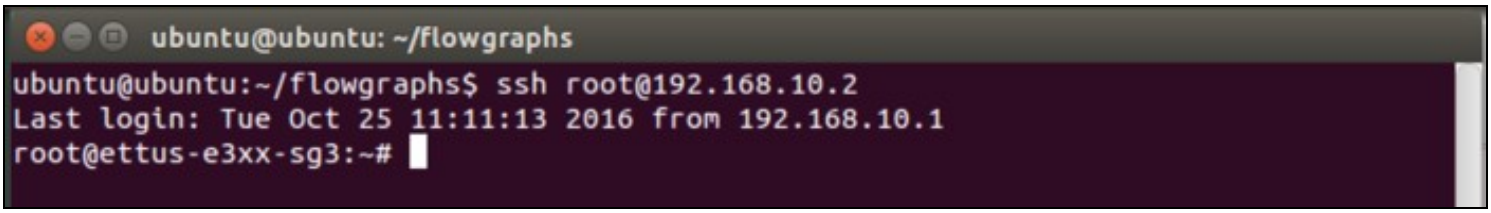
Next, we will use the command line utility `scp` (Secure Copy) to copy the generated Python file to the E310.

```
$ scp e310_fm_receiver_zmq.py root@192.168.10.2:~/.
```

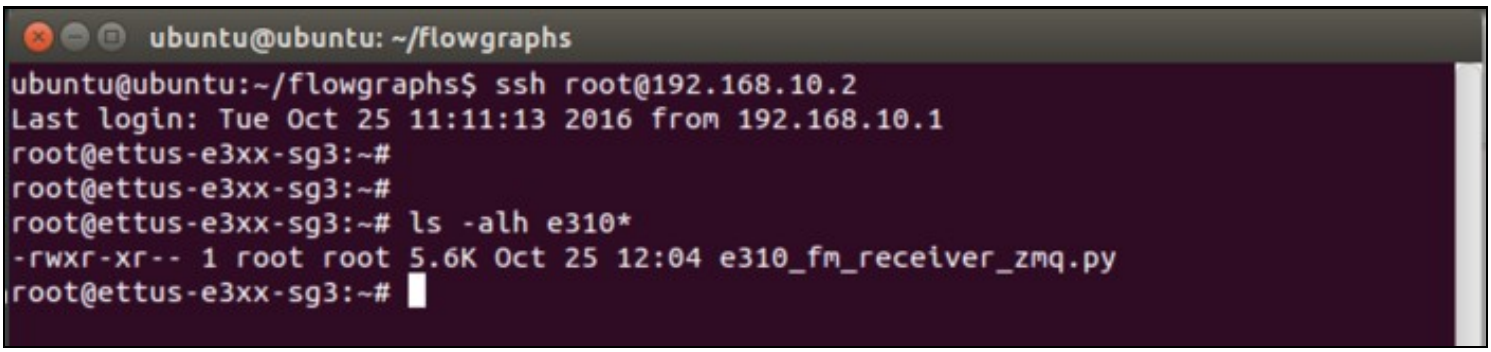


Next, we will SSH into the E310 and verify the file was transferred successfully.

```
$ ssh root@192.168.10.2
```

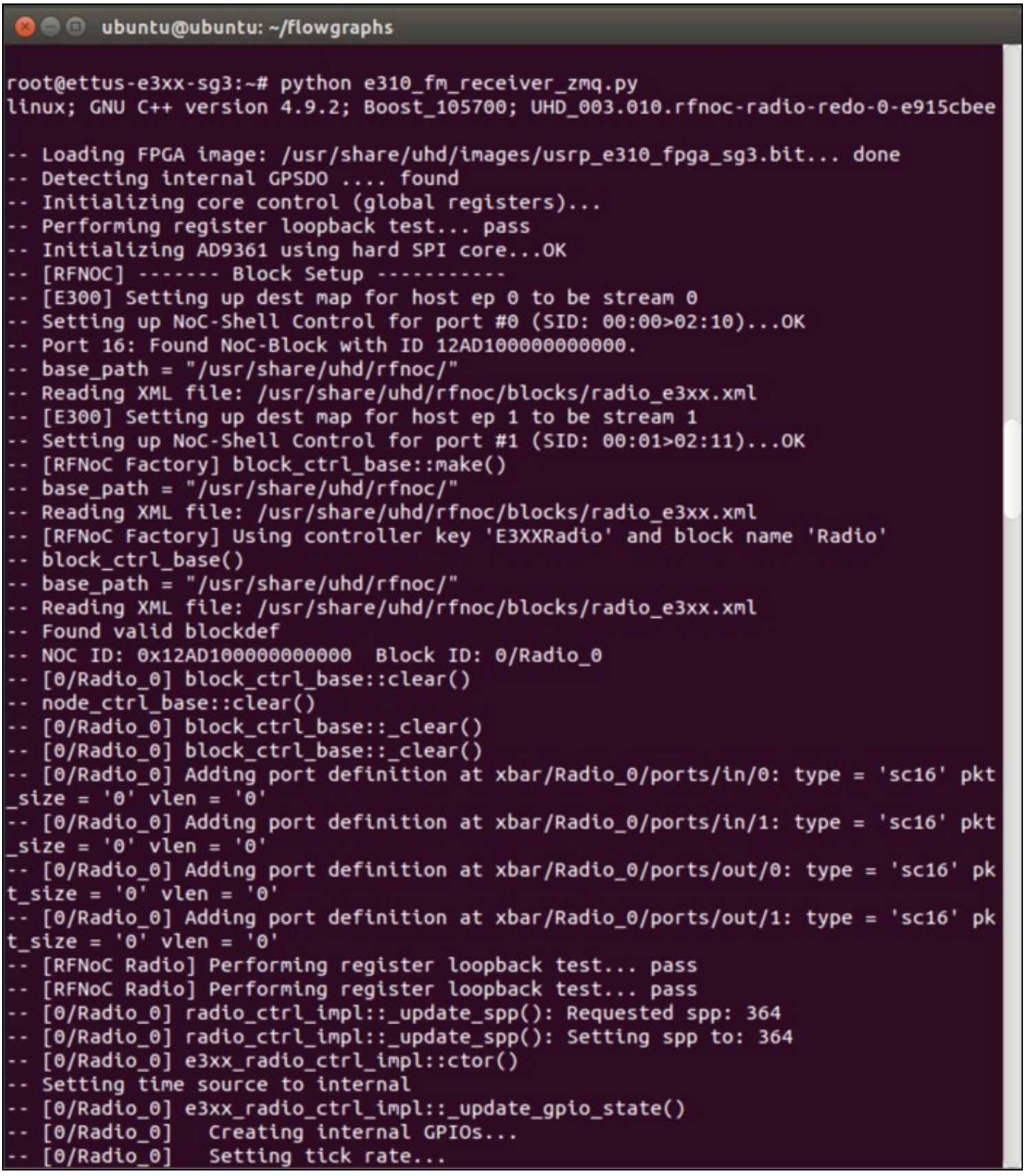



```
root@ettus-e3xx-sg3:~# ls -al e310*
```

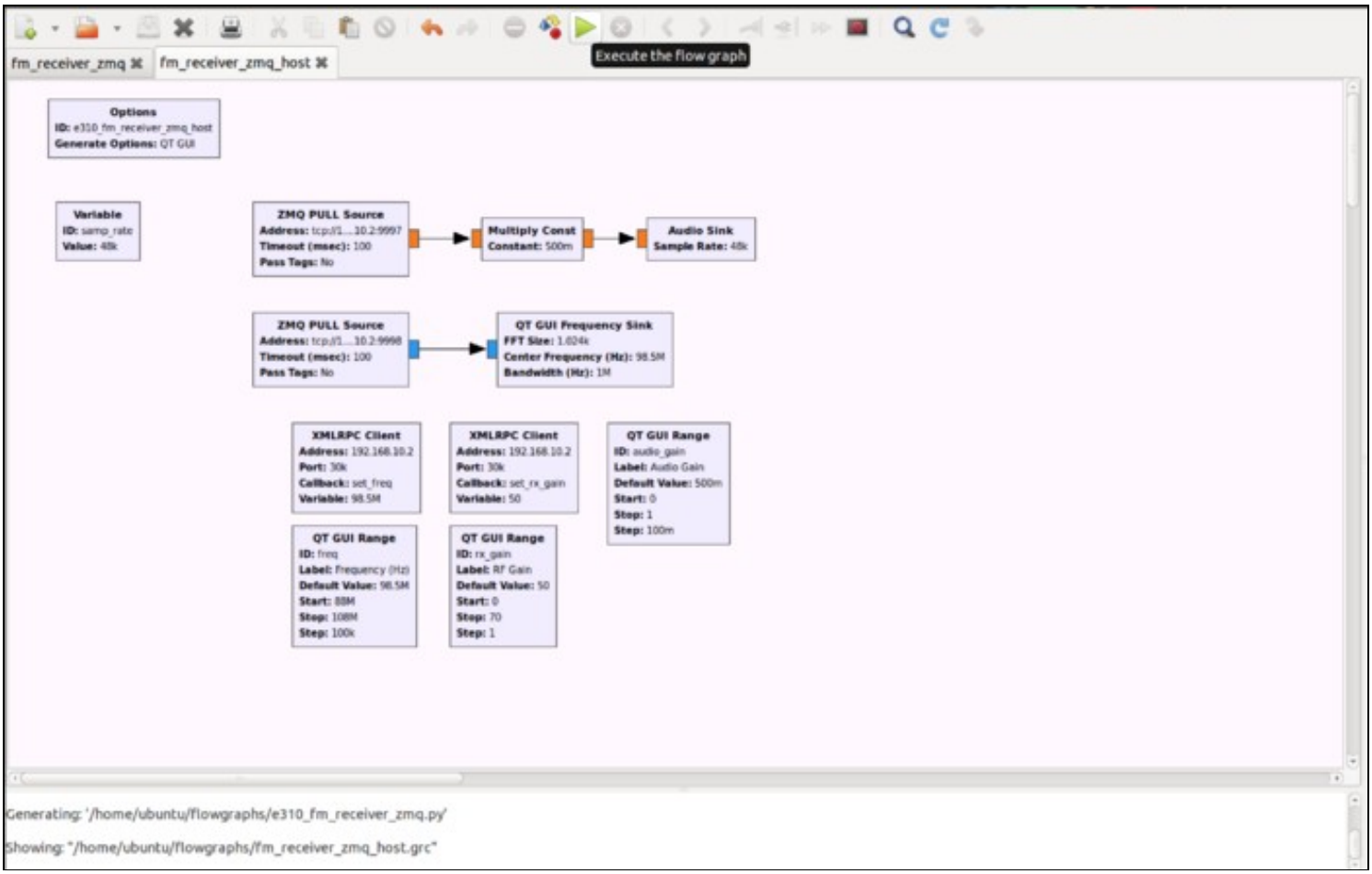


The command below will start the flow graph on the E310, there will be a verbose output.

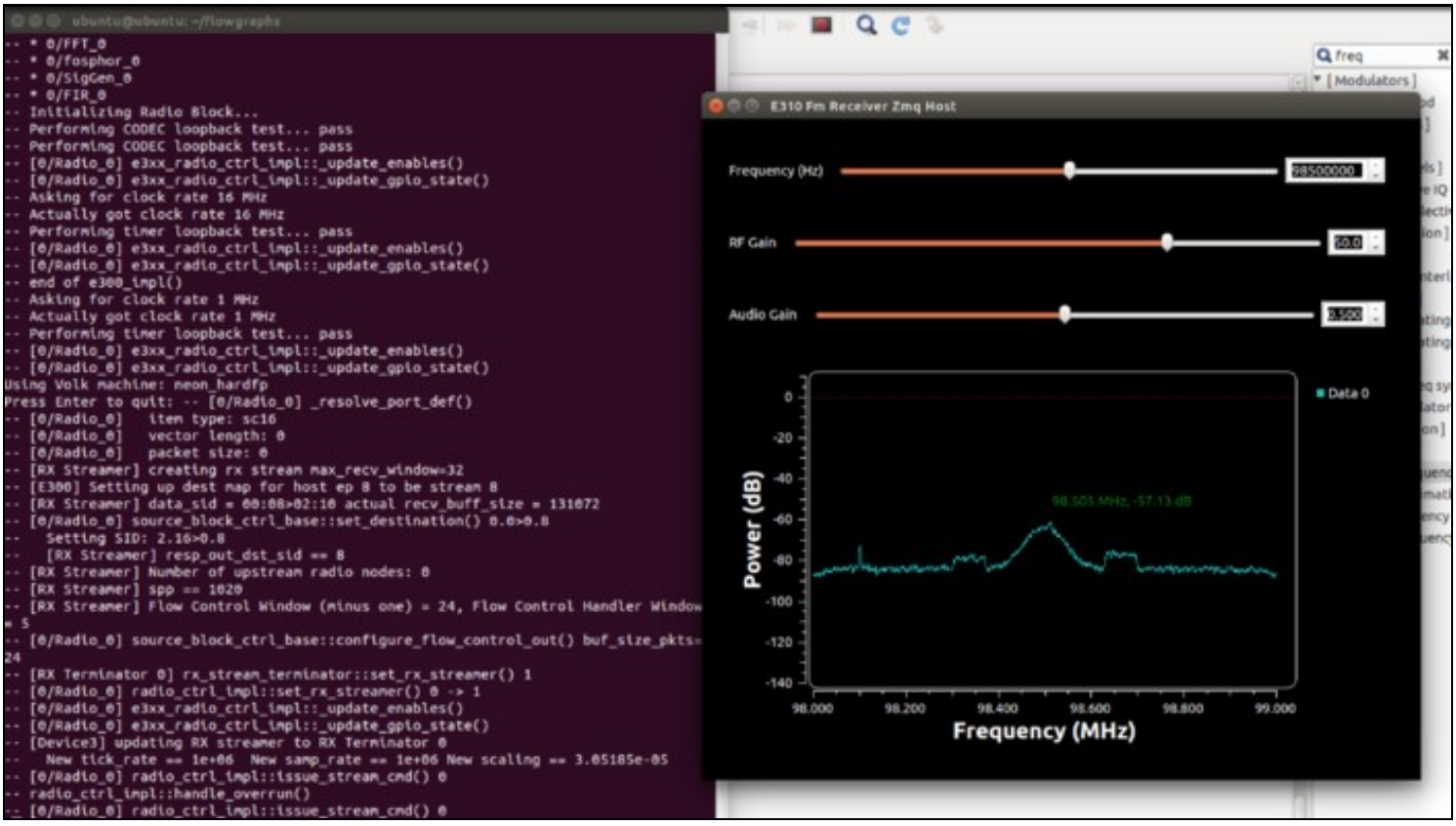
```
root@ettus-e3xx-sg3:~# python e310_fm_receiver_zmq.py
```

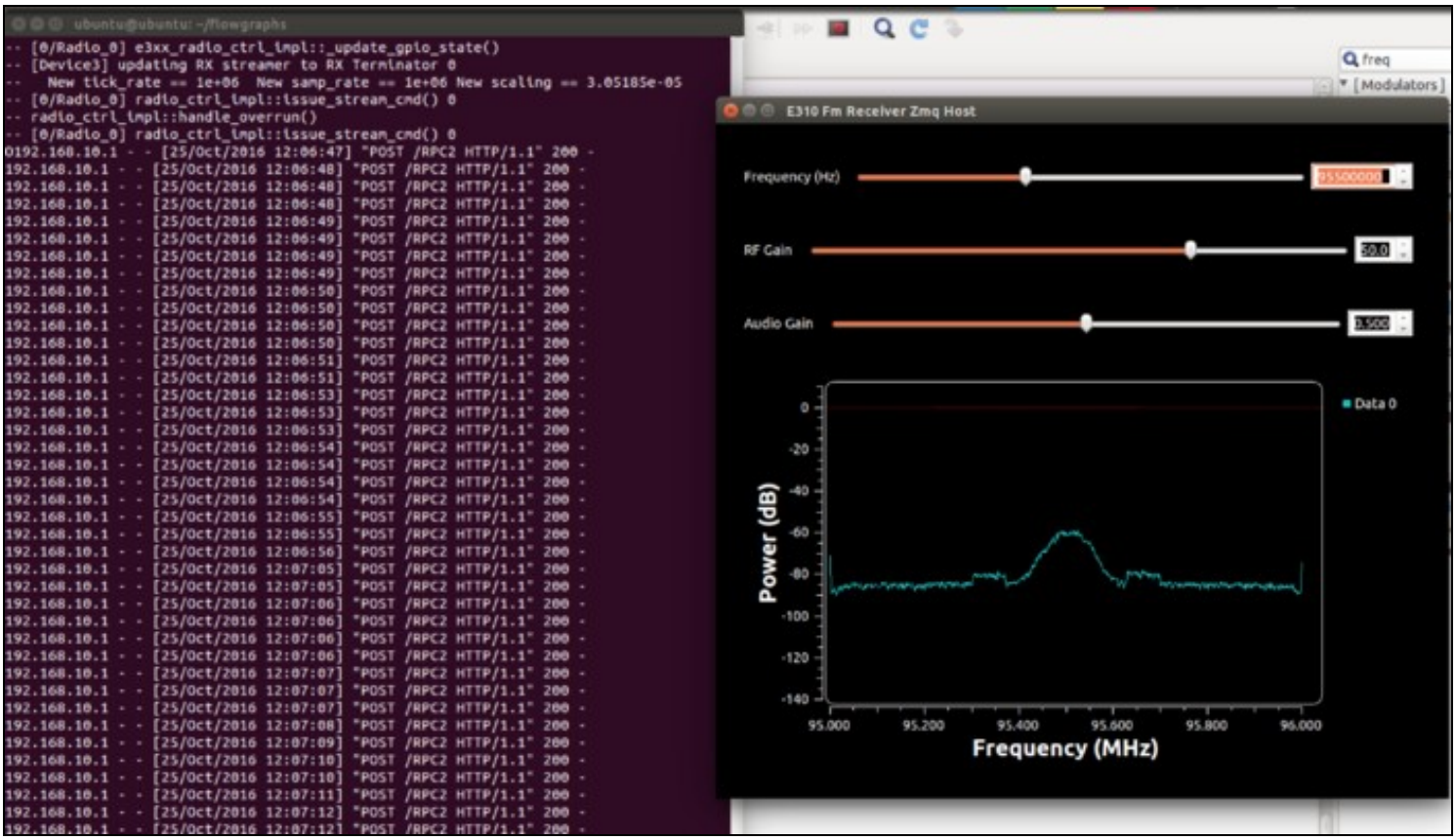


Returning to GNU Radio Companion, open the "fm_receiver_zmq_host.grc" flow graph, and Execute the flow graph.

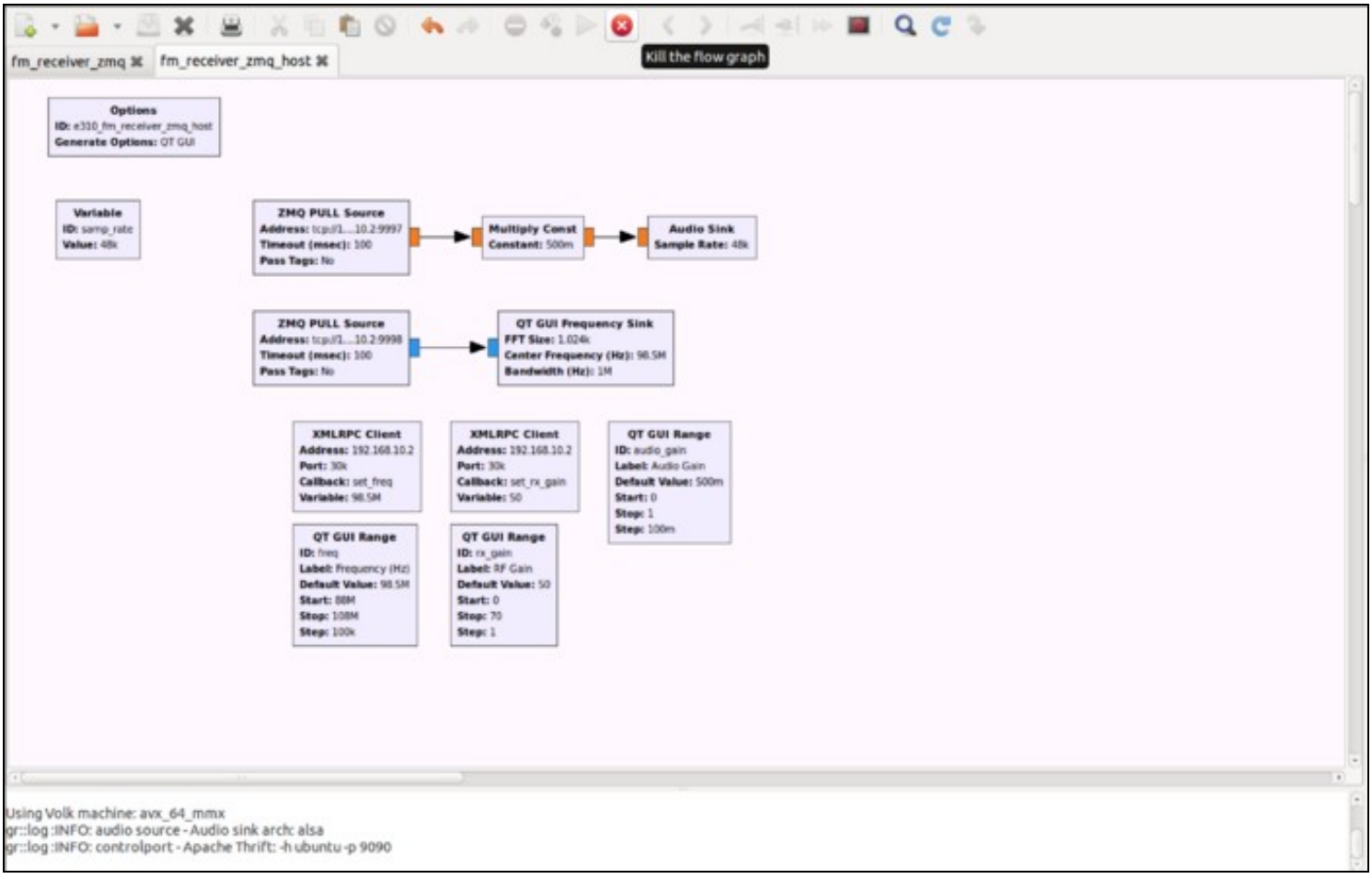


A QT based widget window will open as shown in the figure below. Tune to a strong local radio station by adjusting the QT Range widget.

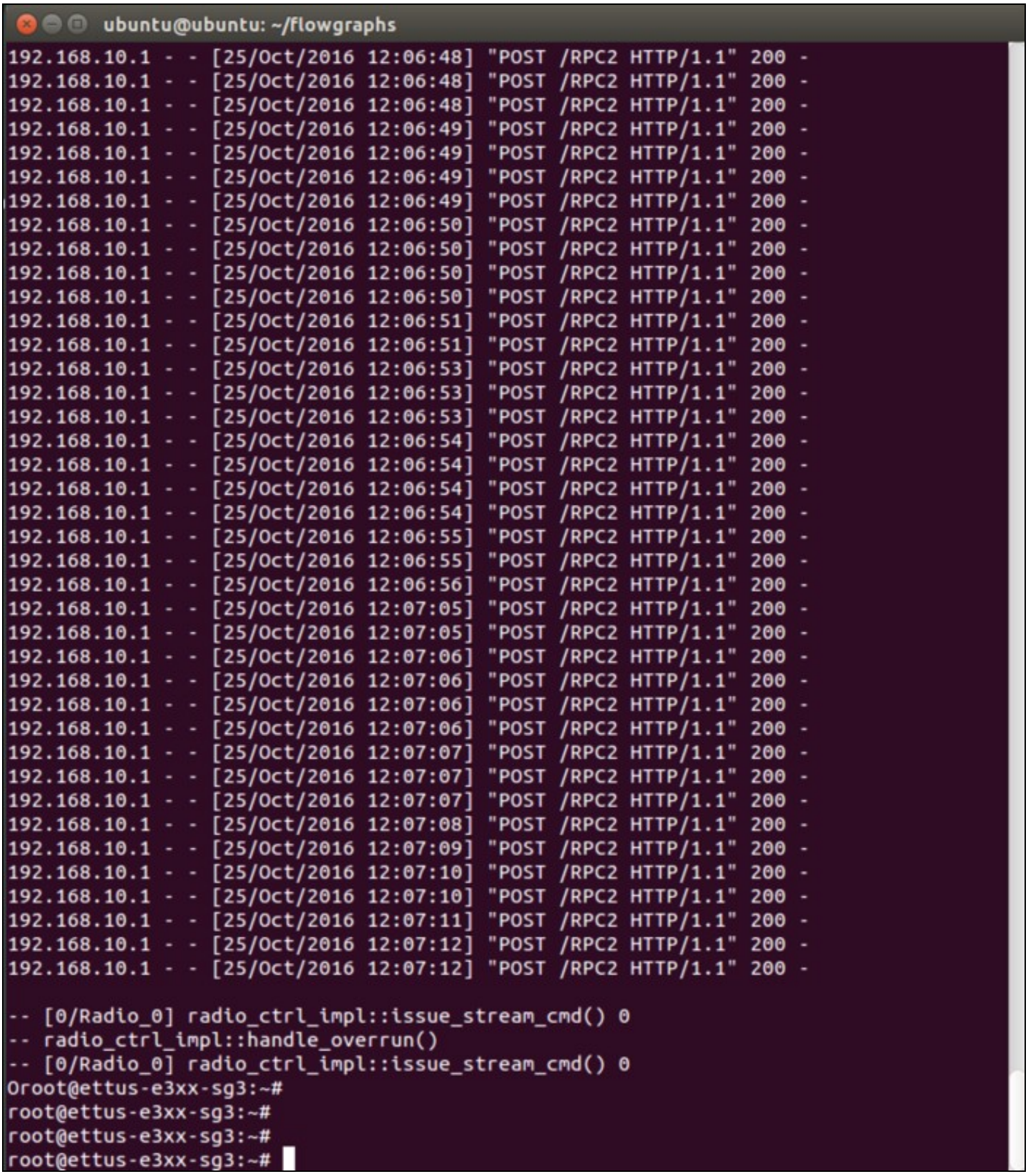




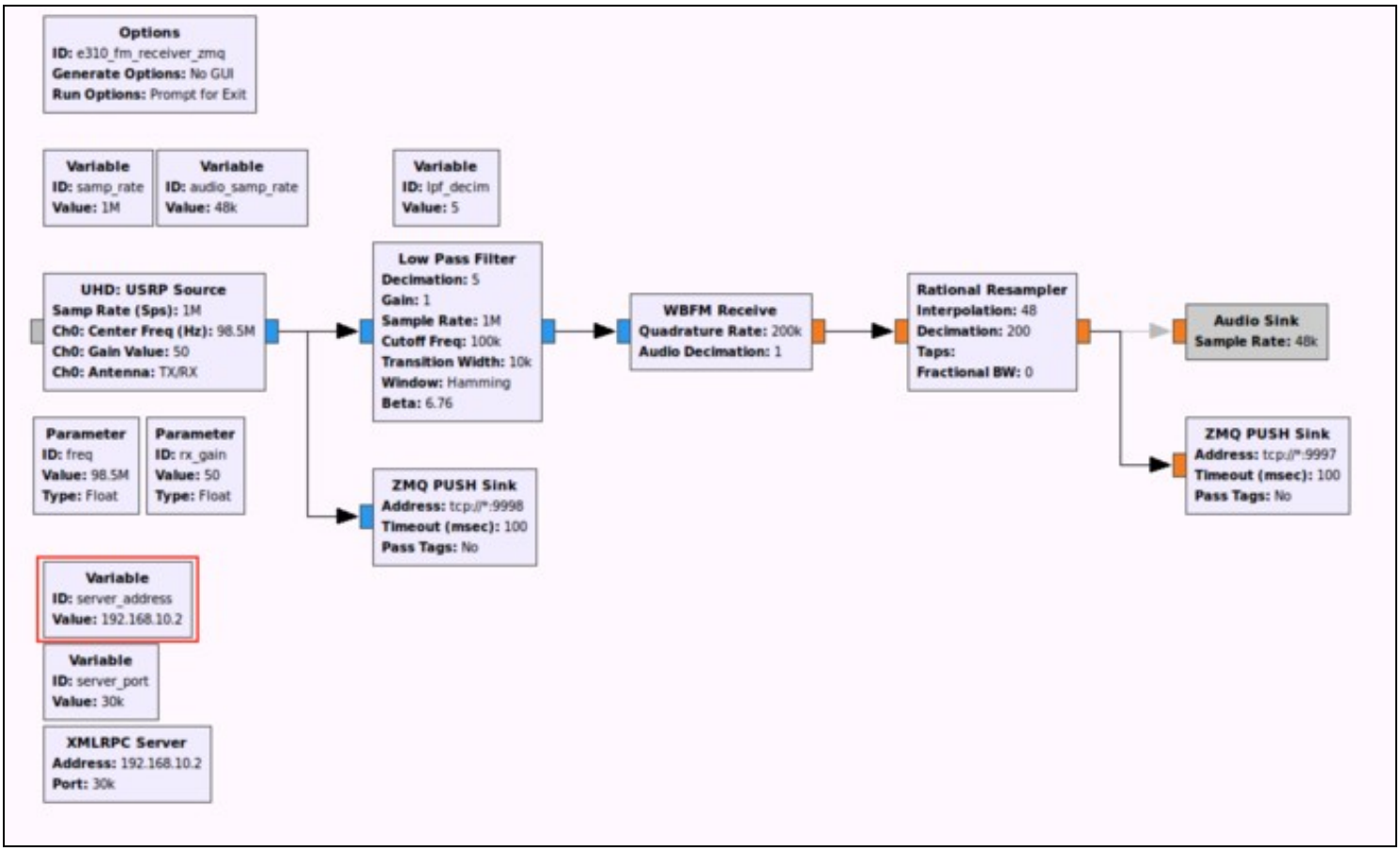
Stopping the application is a multiple step process. First you will need to Kill the flow graph running on the host computer. This can be done by either hitting the "X" in the upper corner of the QT widget window, or by using the "Kill the flow graph" button within GNU Radio Companion.

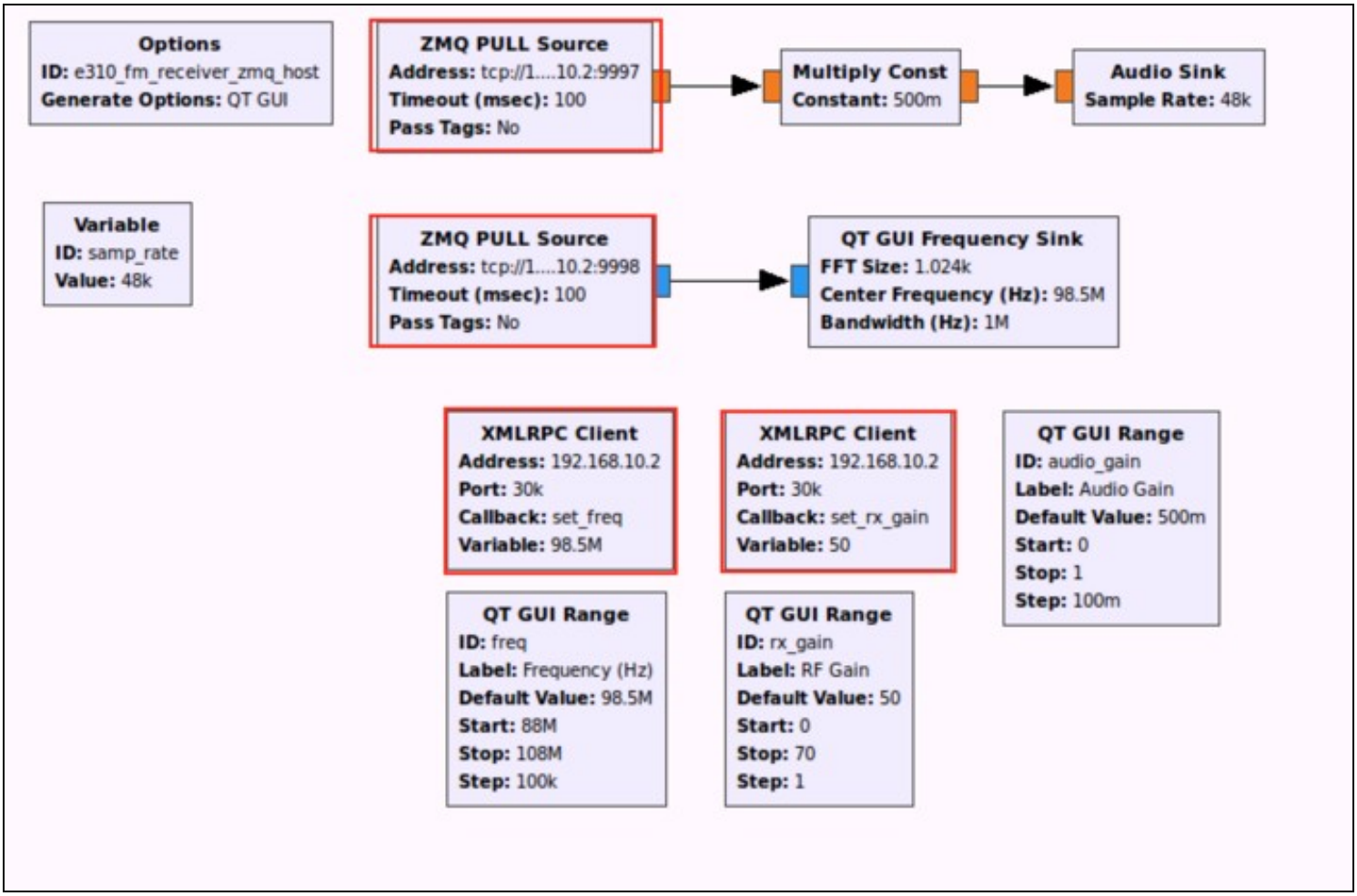


Next, within the Terminal window that is running the flow graph on the E310, simply hit `Enter`.



If your E310 is running with an IP address other than `192.168.10.2`, the flow graphs will need to updated in several locations. The figures below high-light the locations that will need to be changed.


Options
ID: e310_fm_receiver_zmq
Generate Options: No GUI
Run Options: Prompt for Exit
Variable
ID: samp_rate
Value: 1M
Variable
ID: audio_samp_rate
Value: 48k
Variable
ID: lpf_decim
Value: 5
UHD: USRP Source
Samp Rate (Sps): 1M
Ch0: Center Freq (Hz): 98.5M
Ch0: Gain Value: 50
Ch0: Antenna: TX/RX
Low Pass Filter
Decimation: 5
Gain: 1
Sample Rate: 1M
Cutoff Freq: 100k
Transition Width: 10k
Window: Hamming
Beta: 6.76
WBFM Receive
Quadrature Rate: 200k
Audio Decimation: 1
Rational Resampler
Interpolation: 48
Decimation: 200
Taps:
Fractional BW: 0
Audio Sink
Sample Rate: 48k
Parameter
ID: freq
Value: 98.5M
Type: Float
Parameter
ID: rx_gain
Value: 50
Type: Float
ZMQ PUSH Sink
Address: tcp://*:9998
Timeout (msec): 100
Pass Tags: No
ZMQ PUSH Sink
Address: tcp://*:9997
Timeout (msec): 100
Pass Tags: No
Variable
ID: server_address
Value: 192.168.10.2
Variable
ID: server_port
Value: 30k
XMLRPC Server
Address: 192.168.10.2
Port: 30k


Options
ID: e310_fm_receiver_zmq_host
Generate Options: QT GUI
ZMQ PULL Source
Address: tcp://1....10.2:9997
Timeout (msec): 100
Pass Tags: No
Multiply Const
Constant: 500m
Audio Sink
Sample Rate: 48k
Variable
ID: samp_rate
Value: 48k
ZMQ PULL Source
Address: tcp://1....10.2:9998
Timeout (msec): 100
Pass Tags: No
QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 98.5M
Bandwidth (Hz): 1M
XMLRPC Client
Address: 192.168.10.2
Port: 30k
Callback: set_freq
Variable: 98.5M
XMLRPC Client
Address: 192.168.10.2
Port: 30k
Callback: set_rx_gain
Variable: 50
QT GUI Range
ID: audio_gain
Label: Audio Gain
Default Value: 500m
Start: 0
Stop: 1
Step: 100m
QT GUI Range
ID: freq
Label: Frequency (Hz)
Default Value: 98.5M
Start: 88M
Stop: 108M
Step: 100k
QT GUI Range
ID: rx_gain
Label: RF Gain
Default Value: 50
Start: 0
Stop: 70
Step: 1

- E310 Flowgraph Media: fm_receiver_zmq.grc
- Host Flowgraph Media: fm_receiver_zmq_host.grc