RADIO ASTRONOMY

GnuRadio – Software defined radio

- FM receiver system -

Bao Nguyen Gia, Sagi Summer School 2024 (3S24)
Tay Nguyen University
Mail: Nguyengiabao21102014@gmail.com
Advisors: Bang Dinh Nhan

OUTLINE

01

Experiment purpose

02

Installation
GNU Radio

03

FM receiver

1. Experiment purpose

- Describes how to receive broadcast commercial radio stations transmitting frequency modulated (FM) signals using the RTL-SDR receiver.
- Study the waveform of fm in Gnu Radio
- Explore the FM signal reception capability using GNU Radio
- Record signal power and bandwidth parameters
- Study FM modulation and demodulation
- Analyze the waveform after FM modulation

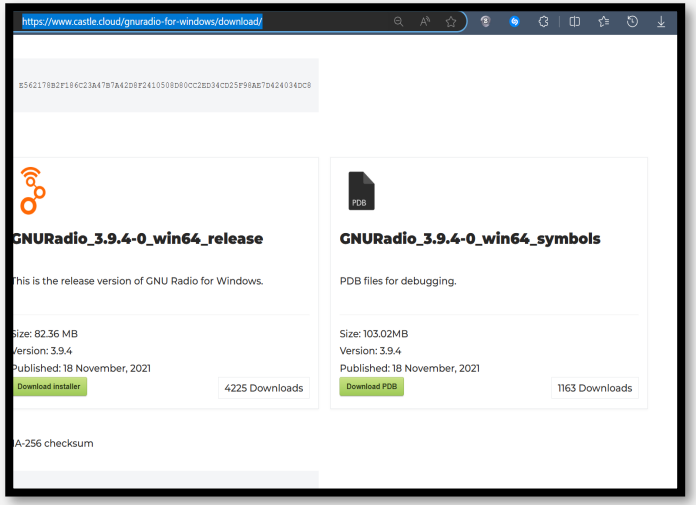
1. Experiment purpose

- The following hardware is needed:

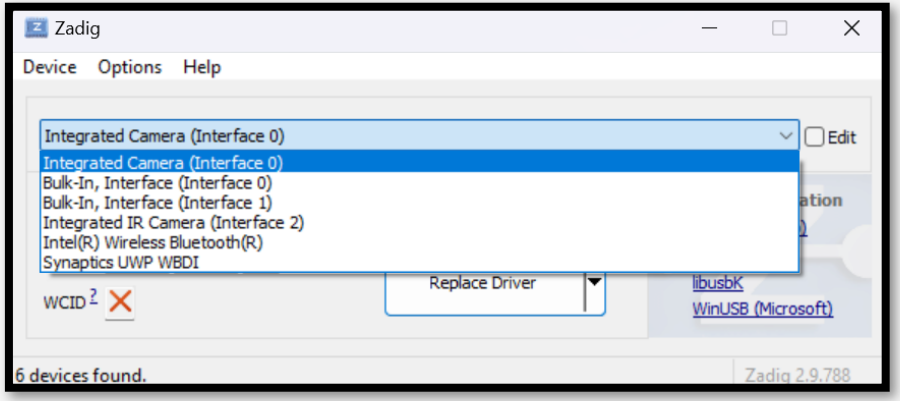
- RTL-SDR Receiver v5
- More info: nesdr_smart_rtl_sdr_v5_datasheet_revision_1.pdf (nooelec.com)
- VHF Antenna

2. Installation GNU Radio

- You can dowload on this link:
Download GNU Radio for Windows – CASTLE
- And then install as usual.



- You may encounter an error where the RTL-SDR is not recognized, so you can dowload Zadig, install driver need for RTL_SDR and try to encounter again.

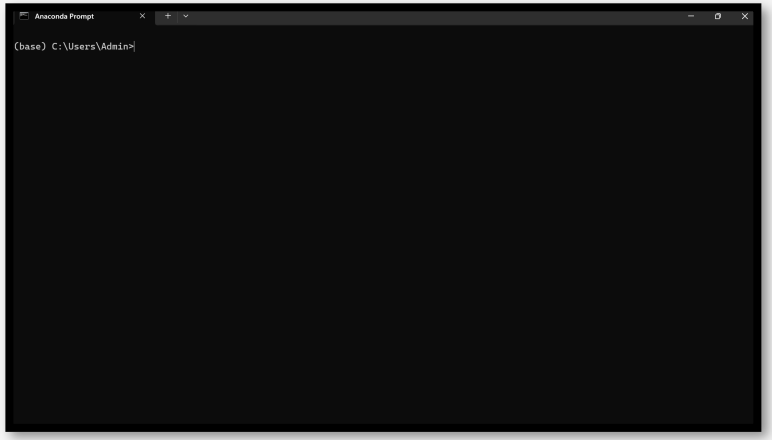


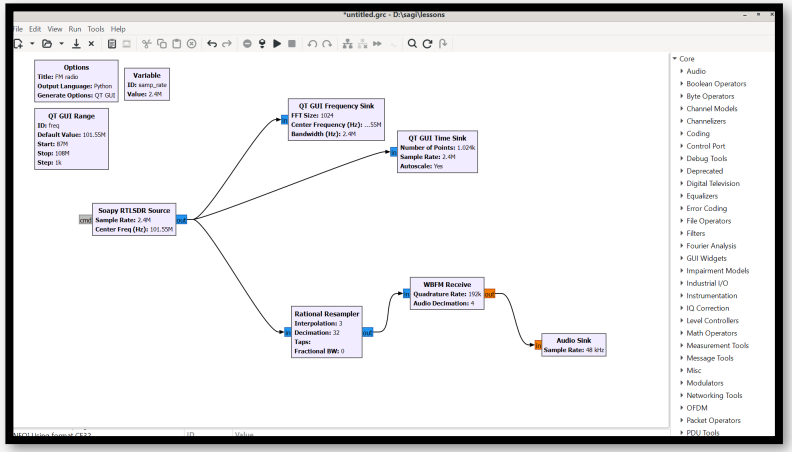
2. Installation GNU Radio

- If you have previously installed the Anaconda software, you can open it.

- Now, run the following commands in sequence to open GNU Radio:

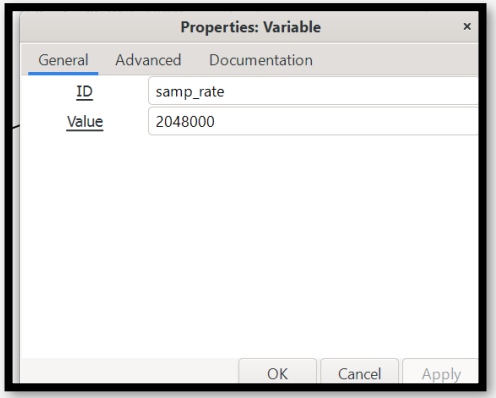
- conda create -n gnuradio
- conda activate gnuradio
- conda install gnuradio
- conda upgrade –all
- conda activate gnuradio
- gnuradio-companion

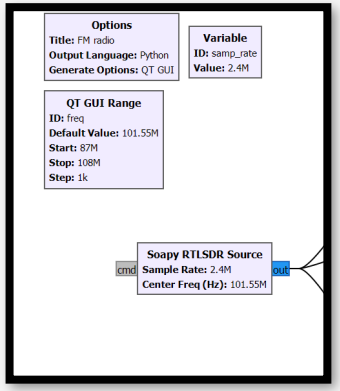


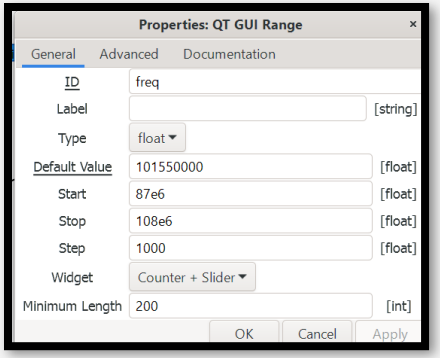


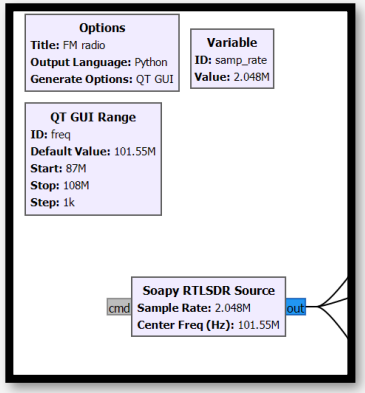
3. FM Receiver

- Add the Soapy RTLSDR Source block to the flowgraph.
- Update the `samp_rate` variable to the value of 2.048M.
- The RTL-SDR source block still has an undefined freq.
- Add the QT GUI Range block into the flowgraph.
- Within the Vietnam, the smallest frequency of a radio station is 87 MHz and the largest frequency is 108 MHz, and each channel is separated by 200 kHz.
- The freq variable within the Soapy RTLSDR Source block is now defined.





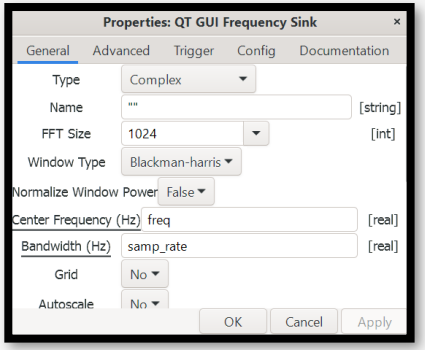


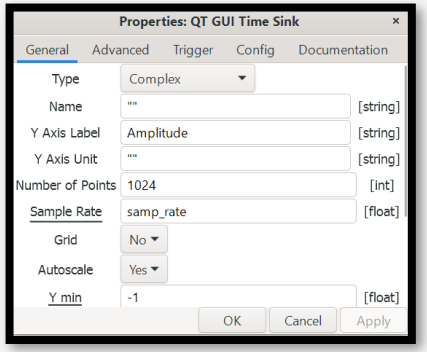


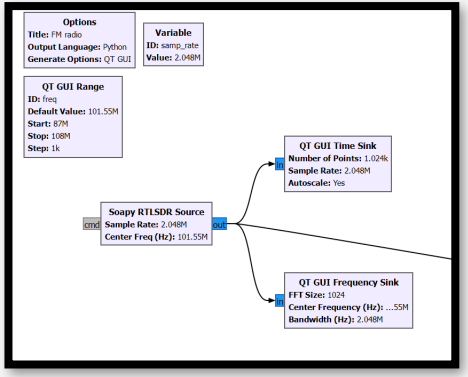
3. FM Receiver

- **Add Time & Frequency Plots**
- Drag in a QT GUI Time Sink and QT GUI Frequency Sink. Enable the auto-scaling in the QT GUI Time Sink block

- Open the QT GUI Frequency Sink and enter freq as the Center Frequency (Hz)

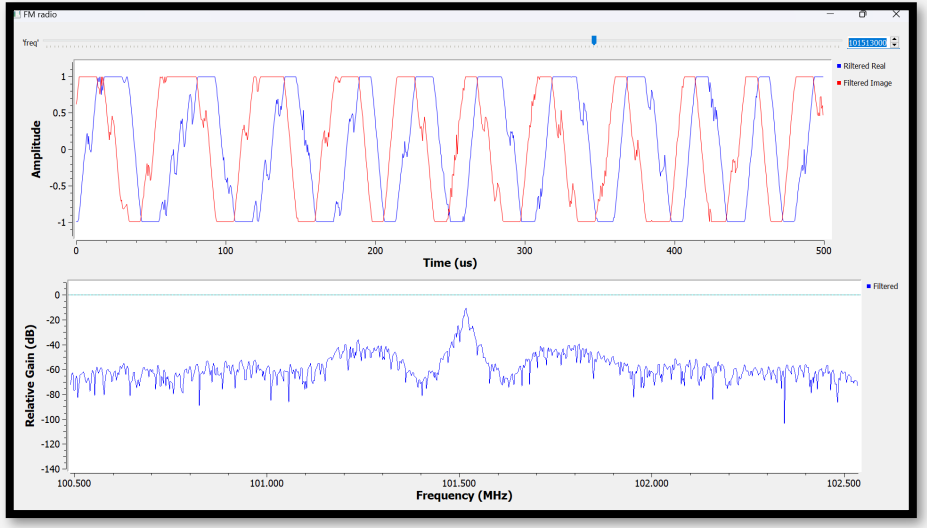


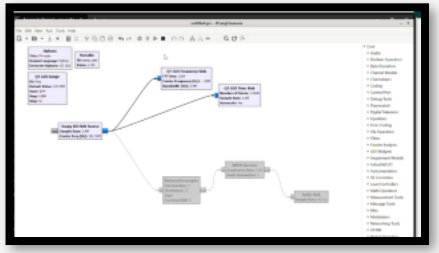




3. FM Receiver

- Run the flowgraph. A time plot and frequency plot will be displayed. This simple flowgraph represents a basic tuner application, such that dragging the slider bar or entering a frequency manually will retune the RTL-SDR and start producing sampled data.

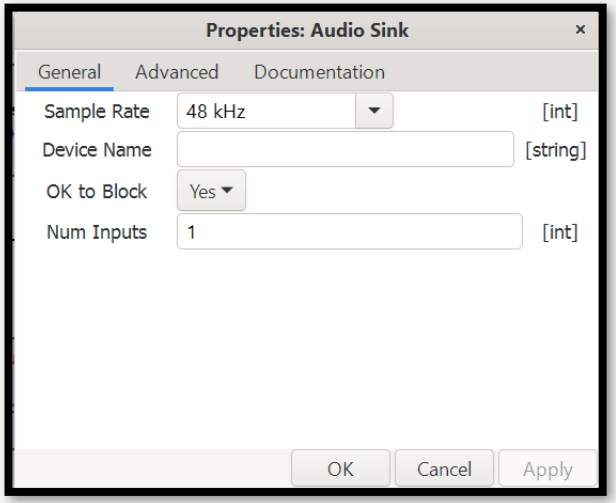


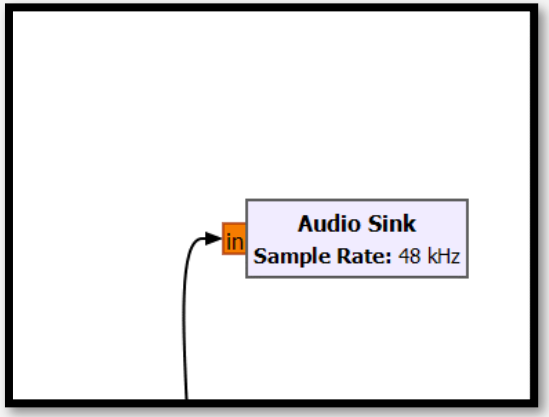


3. FM Receiver

❖ **FM Demodulator**

- An Audio Sink block is needed to play the sound of a demodulated FM radio station. Drag in an Audio Sink block and open the properties. Select 48 kHz
- The flowgraph should now look like the following

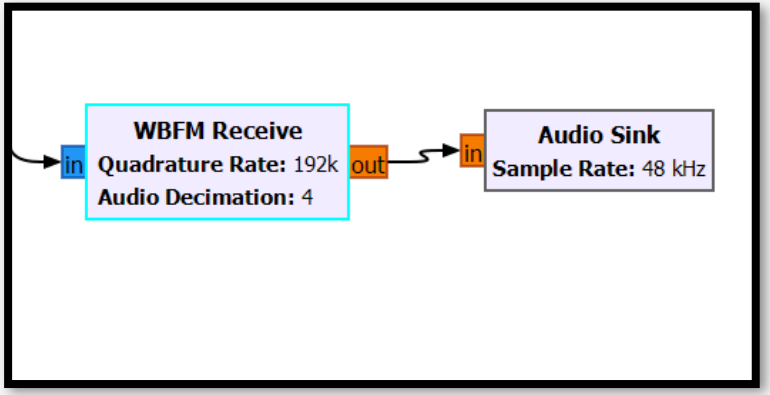




3. FM Receiver

- The rest of this tutorial will work backwards from the Audio Sink and establishing blocks and connections towards the output of the RTL-SDR.

- The next block that is needed is the an FM demodulator. Drag in the WBFM Receive block, which takes complex IQ as an input, demodulates the FM thereby producing real output samples and also performs a decimation.

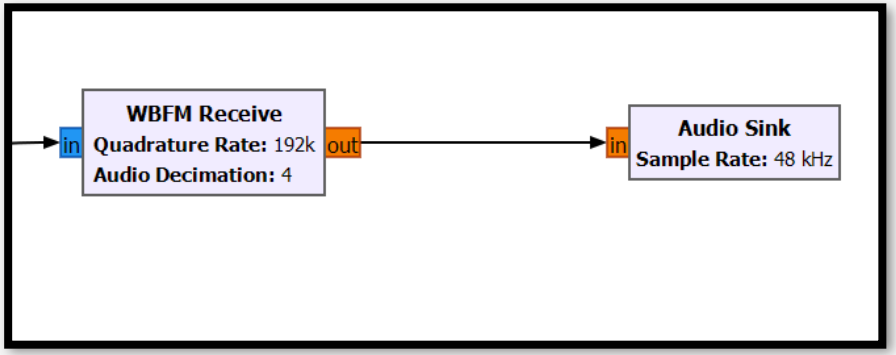


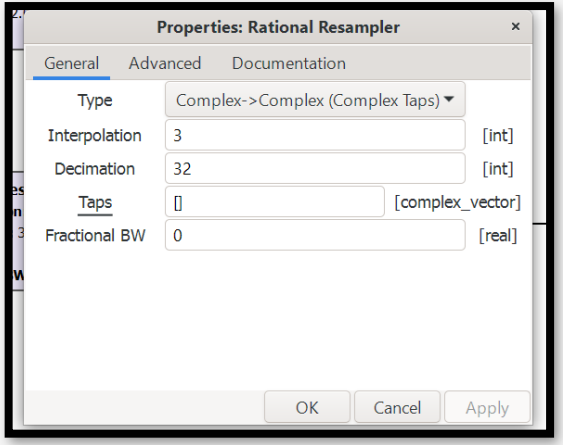
3. FM Receiver

- The flowgraph should now look like the following

- A sampe rate change is needed to convert from the RTL-SDR output of 2.048 MHz to the WBFM input of 192 kHz. The required sampling rate change can be simplified as 192000/2048000 = 192/2048 = 3/32, a rational ratio. Therefore the Rational Resampler block can be used to implement the sample rate change.

- Drag in the Rational Resampler block and open the properties. Enter 3 for the interpolation and 32 for the decimation.



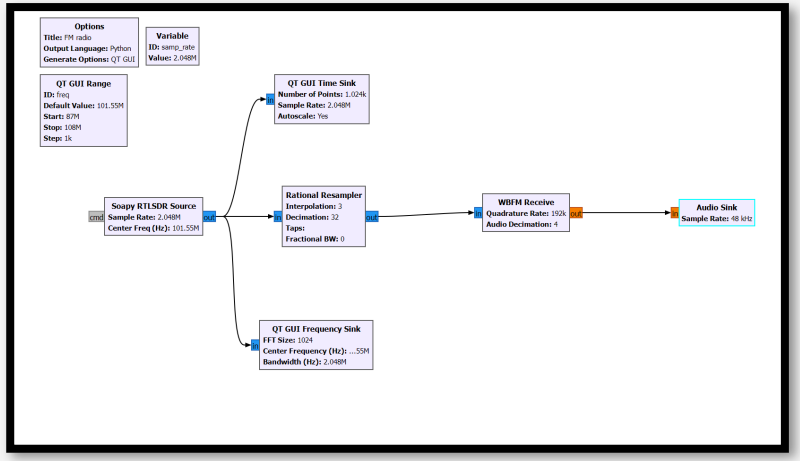


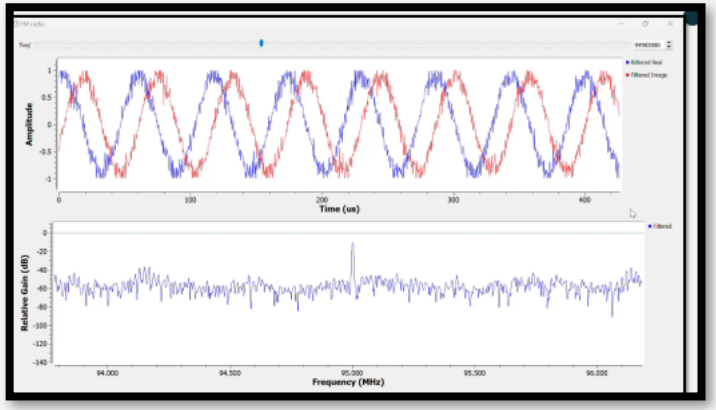
3. FM Receiver

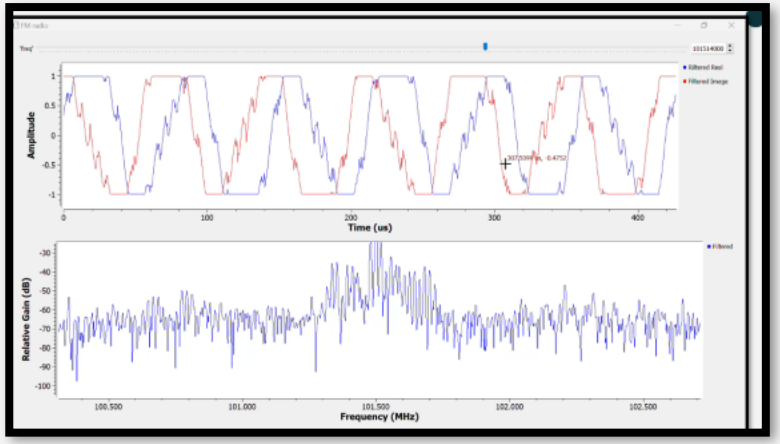
- The flowgraph is now complete and should look like the following

- Run the flowgraph. The same GUI window that was displayed previously will appear but now audio should be playing through your computer. You can drag the bar at the top of the screen to tune to different channels.

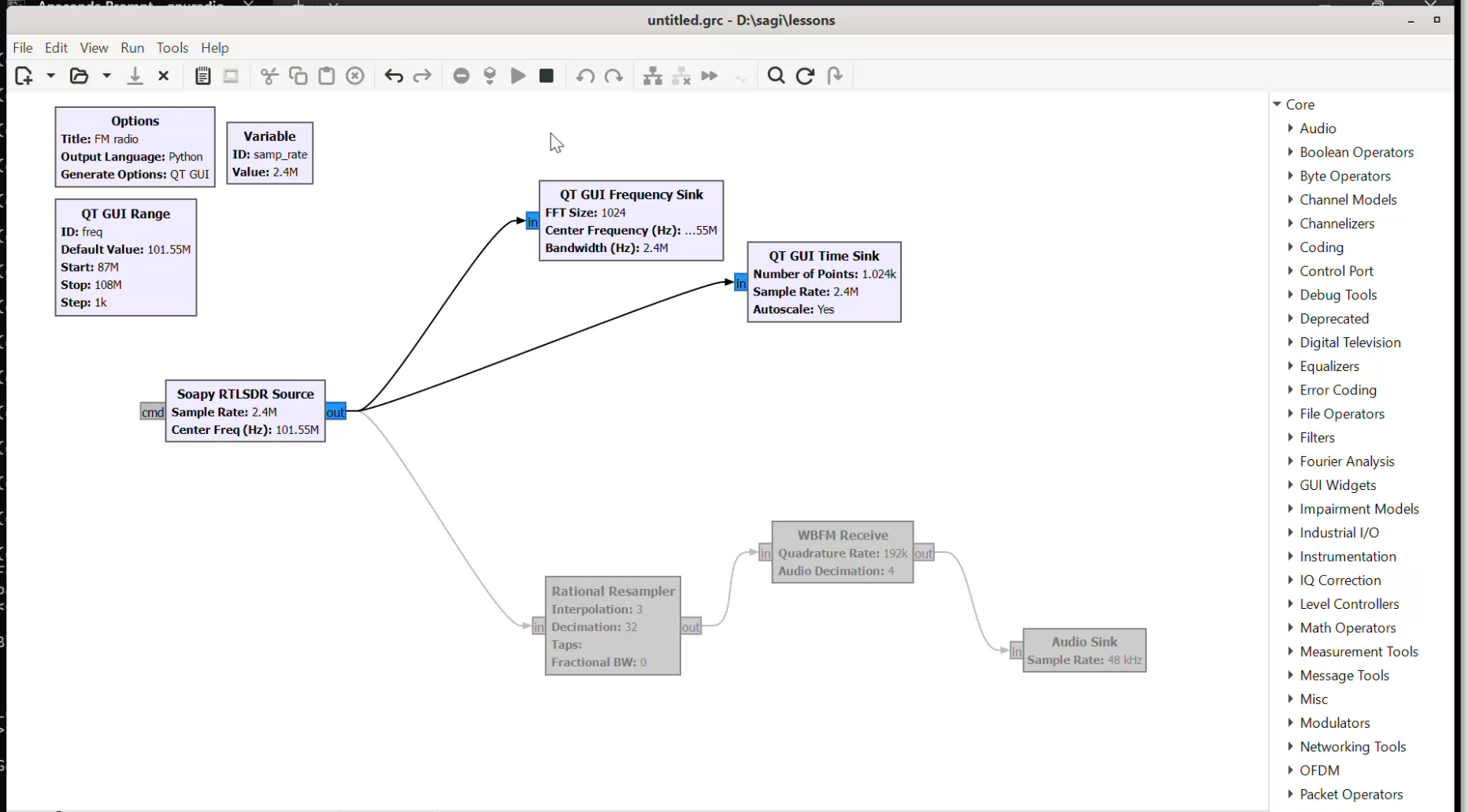
-

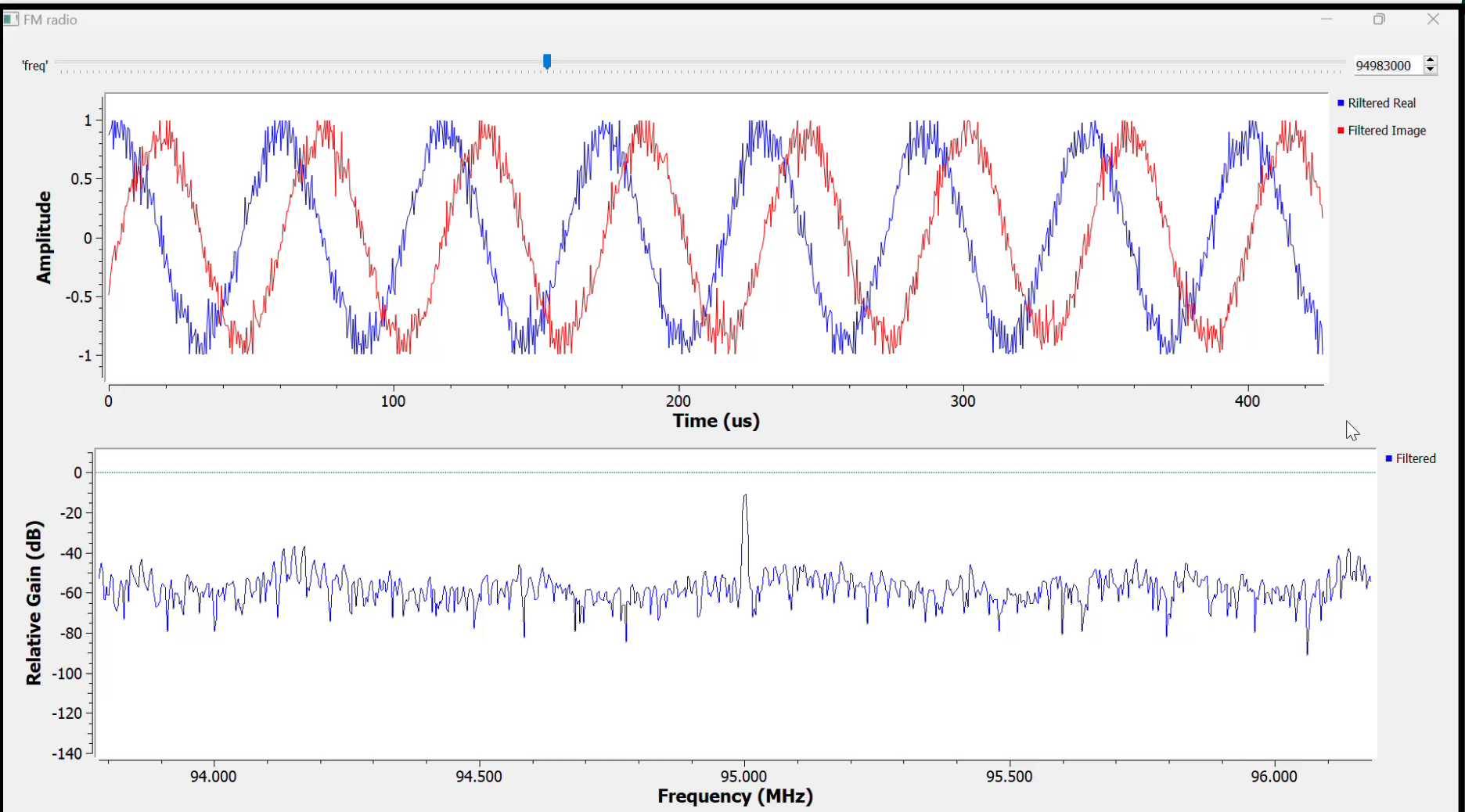


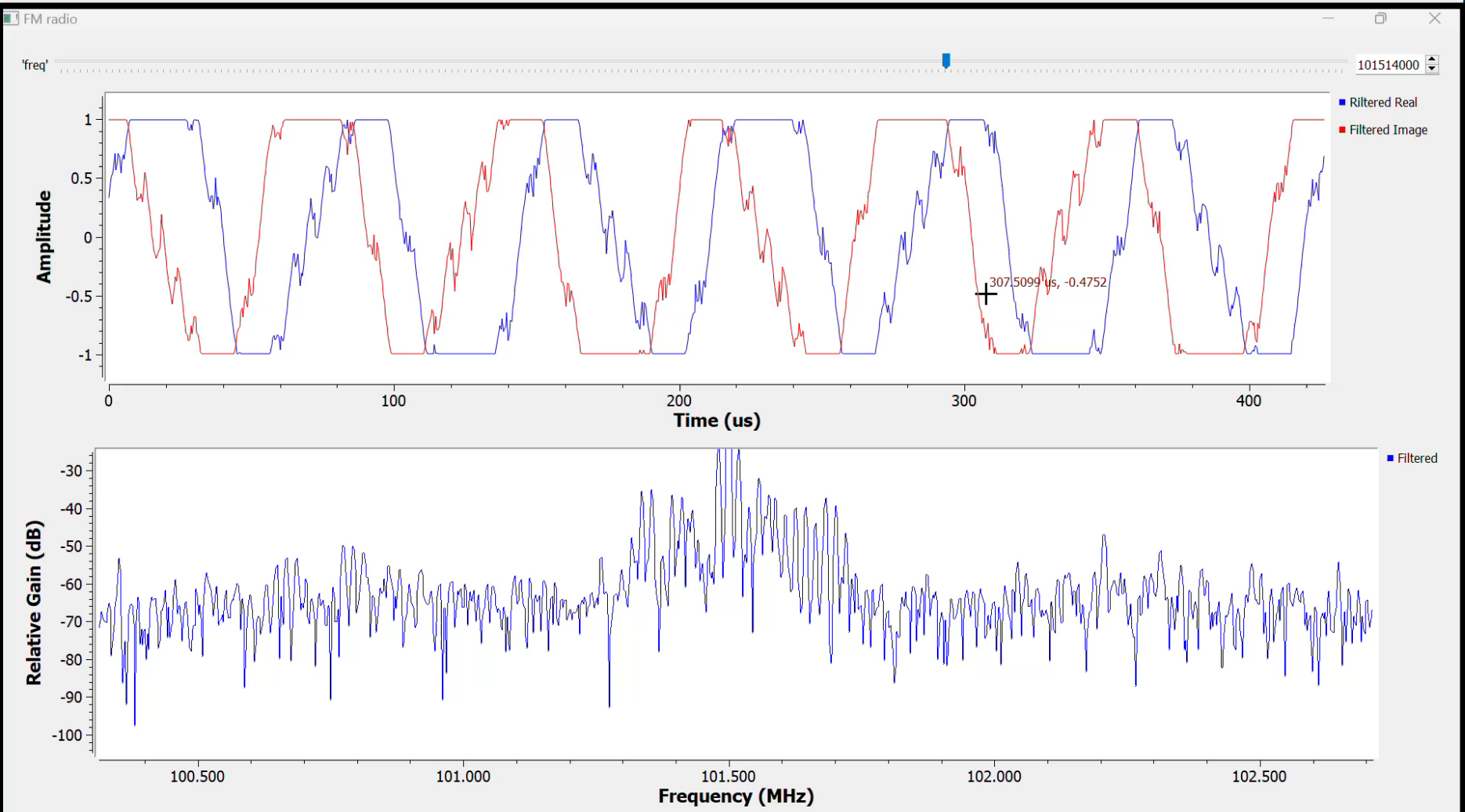




Thanks for listening!

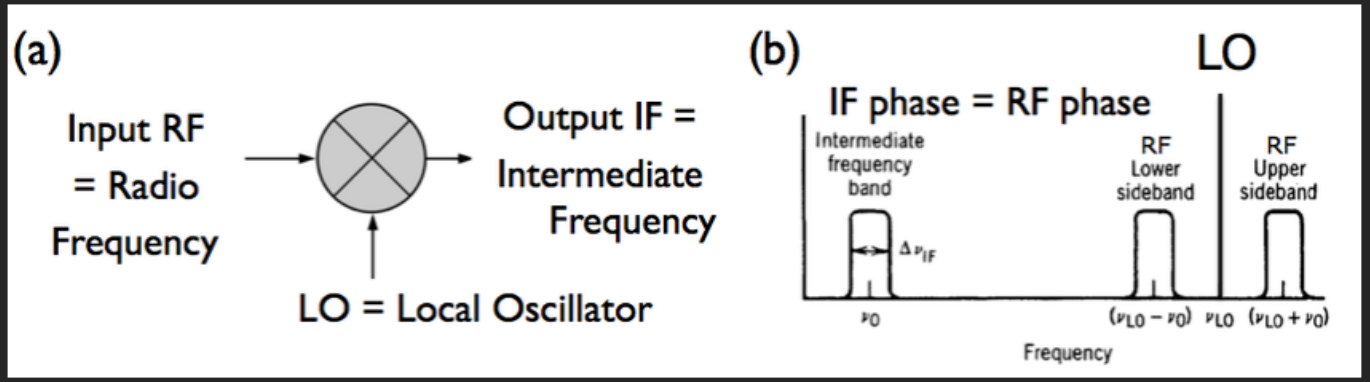

untitled.grc - D:\sagi\lessons
File Edit View Run Tools Help
Options
Title: FM radio
Output Language: Python
Generate Options: QT GUI
Variable
ID: samp_rate
Value: 2.4M
QT GUI Range
ID: freq
Default Value: 101.55M
Start: 87M
Stop: 108M
Step: 1k
QT GUI Frequency Sink
FFT Size: 1024
Center Frequency (Hz): ...55M
Bandwidth (Hz): 2.4M
QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 2.4M
Autoscale: Yes
Soapy RTLSDR Source
Sample Rate: 2.4M
Center Freq (Hz): 101.55M
cmd
out
in
Rational Resampler
Interpolation: 3
Decimation: 32
Taps:
Fractional BW: 0
WBFM Receive
Quadrature Rate: 192k
Audio Decimation: 4
Audio Sink
Sample Rate: 48 kHz
Core
Audio
Boolean Operators
Byte Operators
Channel Models
Channelizers
Coding
Control Port
Debug Tools
Deprecated
Digital Television
Equalizers
Error Coding
File Operators
Filters
Fourier Analysis
GUI Widgets
Impairment Models
Industrial I/O
Instrumentation
IQ Correction
Level Controllers
Math Operators
Measurement Tools
Message Tools
Misc
Modulators
Networking Tools
OFDM
Packet Operators


FM radio
'freq'
94983000
Riltered Real
Filtered Image
Amplitude
1
0.5
0
-0.5
-1
0
100
200
300
400
Time (us)
Filtered
Relative Gain (dB)
0
-20
-40
-60
-80
-100
-120
-140
94.000
94.500
95.000
95.500
96.000
Frequency (MHz)


FM radio
'freq'
101514000
Riltered Real
Filtered Image
Amplitude
1
0.5
0
-0.5
-1
307.5099 us, -0.4752
0
100
200
300
400
Time (us)
Filtered
Relative Gain (dB)
-30
-40
-50
-60
-70
-80
-90
-100
100.500
101.000
101.500
102.000
102.500
Frequency (MHz)

Overview
Radio Astronomy
Radio Telescopes
Early radio tel.
Modern tel.
Sciences
21-cm HI
Redshift
Cosmology
Astrophysics
Planetary Sci.
Remote Sensing
Radio Quiet
Spec. Allocation
Instrumentation
Antennas
Receiver

Heterodyne receiver: Mixer



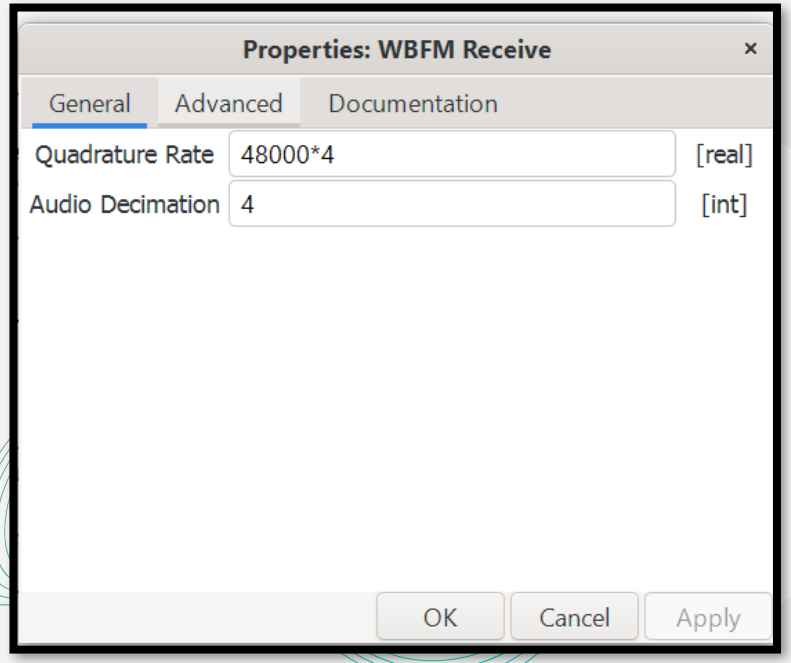
Trigonometry (lượng giác)

$$\cos(\alpha+\beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta$$
$$\cos(\alpha-\beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta$$

Mixer = multiply
Input signal = frequency 1
Local oscillator (LO) = freq 2

Working principle:

$$\cos(2\pi f_1 t)\cos(2\pi f_2 t) = \frac{1}{2}\cos[2\pi(f_1-f_2)t] + \frac{1}{2}\cos[2\pi(f_1+f_2)t]$$

3. FM Receiver

- Add a WBFM Receive block, open the properties and enter in the quadrature rate of 192 kHz and an audio decimation of 4. Note that the quadrature rate must be evenly divisible by the audio decimation factor, and that the audio decimation must be an integer.