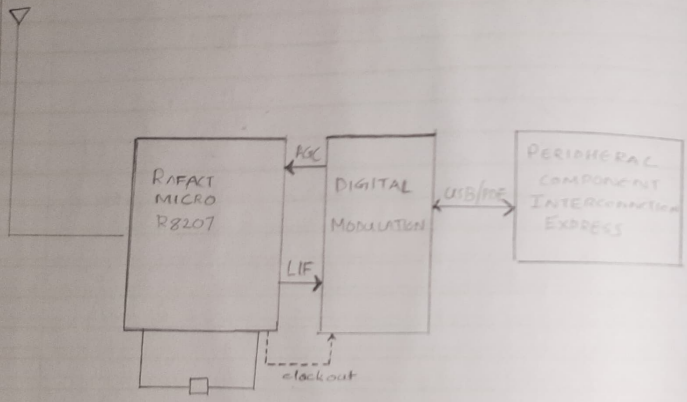
INDEX

Sl. No.	Name of Practical	Date	Page No.	Remarks
	PART C			
1.	EXPERIMENTING WITH SOFTWARE DEFINED RADIO (SDR)	8/1/25	1	
2.	EXPERIMENT WITH GNU RADIO AND FLOWCHART	8/1/25	5	
2A.	AM MODULATION	8/1/25	10	
2B.	FM MODULATION & DEMODULATION	22/1/25	12	

BLOCK DIAGRAM R820TIC



8/01/25

1

PART-C

EXPERIMENT NO:1

EXPERIMENTING WITH SOFTWARE DEFINED RADIO (SDR)

Aim: ~~Software defin~~

To familiarize with software Defined Radio.

Theory:

A software defined radio (SDR) is a radio communication system which helps in the modulation of radio signals. It performs significant amount of signal processing using a general purpose computer. The main components of a hardware radio such as mixers, local oscillators filters etc will be simulated using software. Both hardware and software are required for constructing a S.D.R.

Hardware required:

(1) RTL SDR dongle

(2) Hack RF one if you want to transmit

(3) If the lab or class is far away from the FM stations & one is unable to recieve FM station with clarity then an FM transmitter using

Raspberry Pi can be utilized.

Software Required:

(1) All experiments have to be done in ubuntu 20.04 LTS version.
(2) Gqrx (Graphical Radio reciever)
(3) GNU Radio (software development tool kit)
(4) Cubic SDR (if you want to transmit)
(5) They are plenty of others software available

Installation Procedure:

Step I: Install ubuntu 20.04 LTS

Step II: Take the terminal sudo apt update
This update step help one ubuntu installation to be current with that of ubuntu server.

Step III: Sudo apt install gnu radio.
GNU radio comes with lots of downloads
All the device & necessary libraries will be downloaded & installed.

Step IV: To check whether the installation is accurate type st and then press tab
If the files like stl-fm, stl-sdr
stl-test etc are displayed then, software installation is accurate.

Step V: Install sudo apt install gqrx-sdr.

R8207 - Rafael : it is a tuner IC.
RTL2832U - it is a digital modulator
USB 2.0
28.8 MHz crystal clocks
MCX connector
IR sensor
Power LED
All the parameters of a R820T chip of digital demodulator IC are valid using the software. RTL8232u digital demodulator IC will split the incoming RF signal into I & Q elements.

Step VII : It is very important ubuntu kernel realize the RTL SDR dongle for that type lsusb before the dongle is inserted press enter the list of all possible peripherals attached to the usb will be displayed

Step VIII : Plugin the RTL-SDR dongle.

Step VIII : Again type lsusb & press enter Realtek semiconductor comp RTL2838 DVB-T will be displayed.

Step IX : Type gqrx. Once we know that kernel has sucessfully recognized the dongle type gqrx. Press enter configuration window will open. From configuration window select the device.

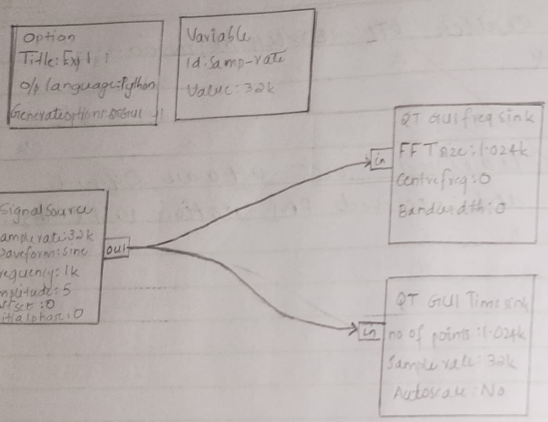
- "Realtek RTL 2838UHIDK SN000...."

Press Ok

Result

Familiarized with software Defined Radio and recieved FM station 104.8 MHz.

17/8/25



8/01/25

5

EXPERIMENT NO: 2

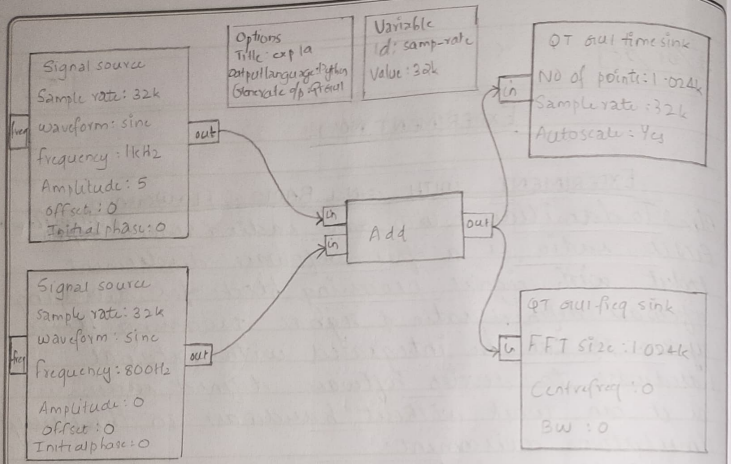
EXPERIMENT WITH GNU RADIO & FLOWGRAPH

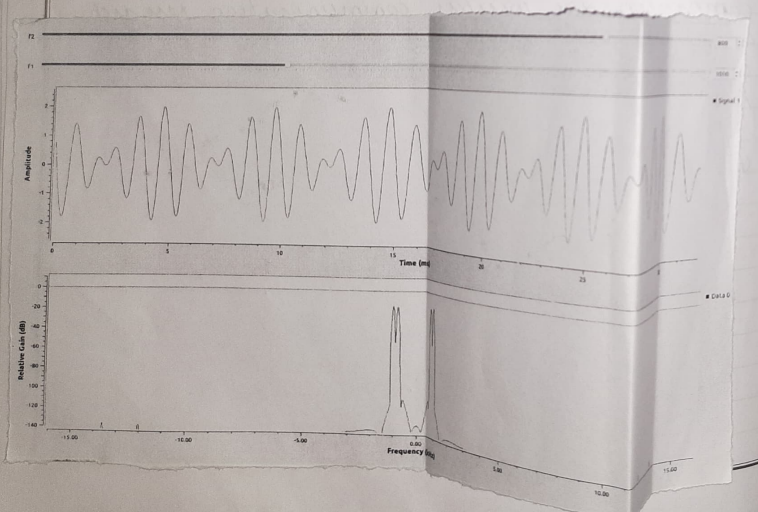
Aim: To familiarise with GNU radio & realise various flowgraphs output

GNU radio is a free software development toolkit with signal processing blocks for developing software defined radio & signal processing blocks system. It can be integrated with external hardware to develop software defined radio or it can work without hardware in the purely simulations environment.

Worldwide habbists, academician & commercial project developers are using GNU radio companion for wireless communication research and for developing software radios. GNU radio companion is a graphical user interphase that help us to built the GNU radio programs.

Step 1: In the terminal, type GNU radio companion press enter. The GNU companion window opens we can see two block options. Option and variable. Option block helps us to set the parameter of programs. Variable block help us to set the samplerate. Double click on the option block. Properties: option window appears. We can proceed the id, title & author.





6

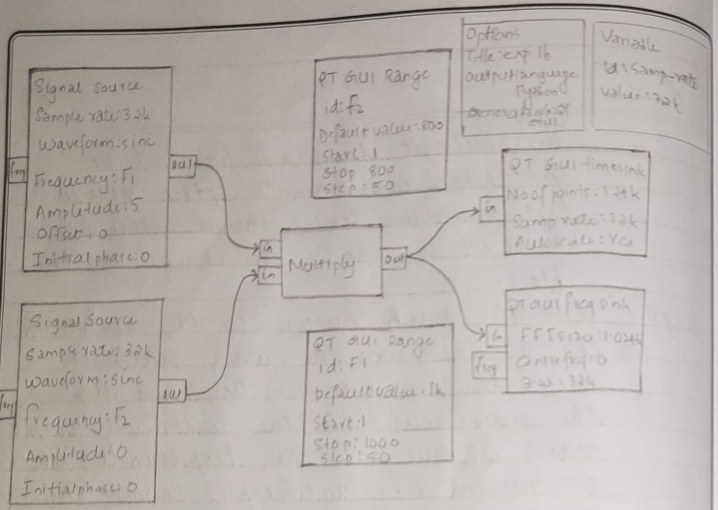
The output language has to be set as python and generate option QT GUI kept the source. Enter the id, title, author. Press OK. We can see the changes in option block.

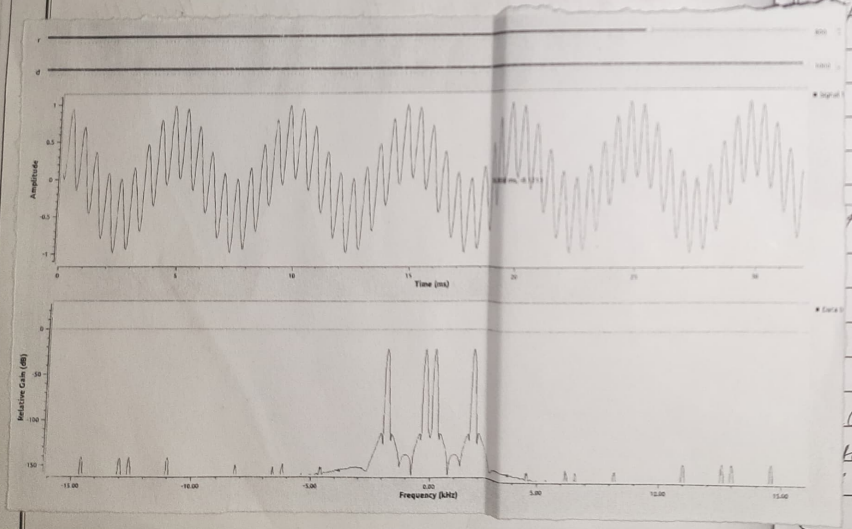
step II: Click the search button in the right of the window from the wave generator select the signal source. Drag and drop the signal source to the workspace. Again search QT GUI from the dropdown list select QT GUI sink. Drag and Drop it. Connect the block. If the end points of the connected blocks are of different colour then it is error. The title of the block will turn red if there is any error in the parameter provided. Then we confirm the flowgraph created.

QT GUI sink acts as an oscilloscope & help us to see the signal source. Always keep the autoscale as yes. If multiple waveforms are to be displayed no of i/p can be given as two or more. A python file is also generated while saving the o/p.

Audio sink block helps us to hear the tones of the signal frequency.

Add two sinusoidal signals of frequency 1kHz & 800Hz. Display the result in both time & freq domain.





Step I: On search bar type signal source
Drag the signal source to the main window.

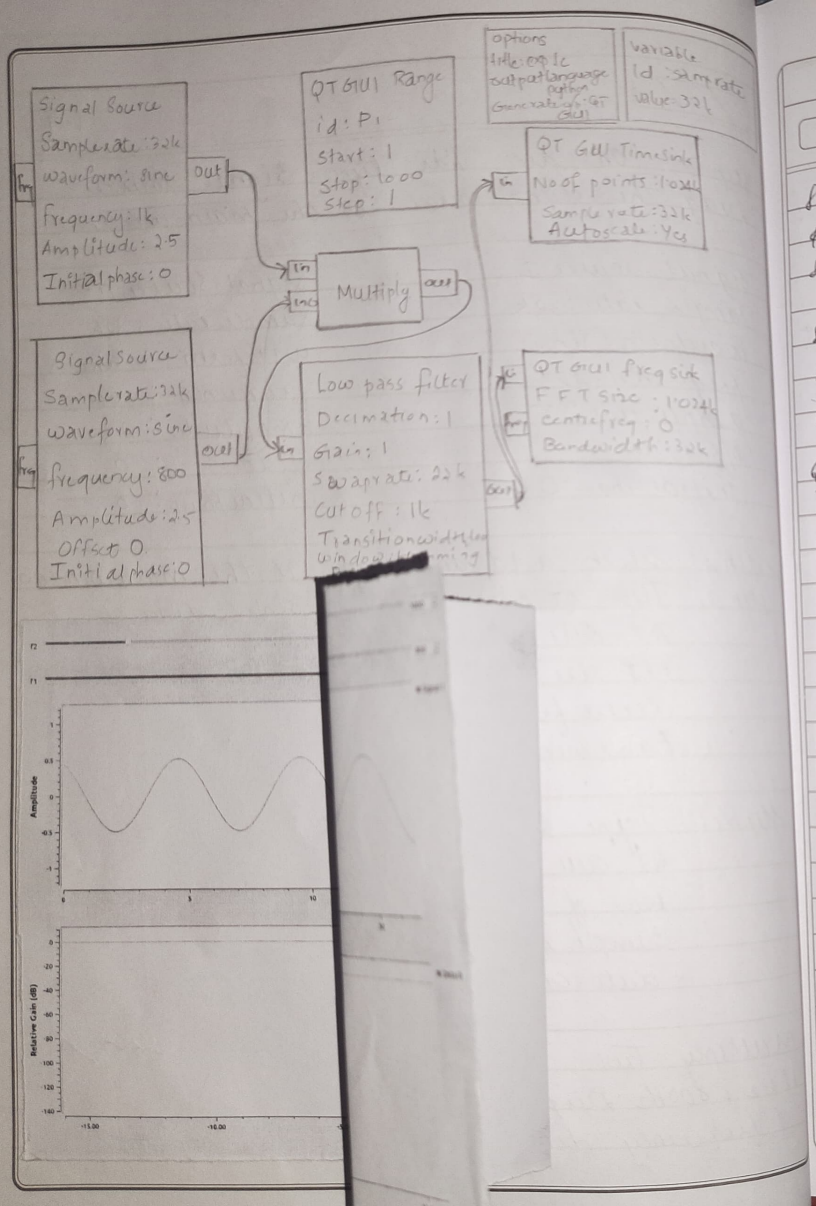
Signal source I	Signal Source II
sample rate: 32k	sample rate: 32k
waveform: sine	waveform: sine
frequency: 1k	frequency: 800Hz
amplitude 2.5	amplitude: 2.5
offset: 0	offset: 0
Initial phase: 0	initial phase: 0

Step 2: Select add and drag to the main window

Step 3: Type QT GUI frequency sink & drag it.
QT GUI frequency sink
FFT size: 1024k
Centre frequency: 0
Band width: 32k

Step 4: Type QT GUI Time Sink & drag it.
QT GUI Time Sink
No of points: 1024k
Sample rate: 32k
autoscale: yes

Multiply two sinusoidal signals of frequency
Hz & 800Hz. Display the result in both the time
& frequency domain.



8

Step 1: Add 2 signal source of frequency 1kHz & 800Hz to the workspace.

Step 2: Add multiply block from the drop down list to the workspace.

Step 3: Add QT GUI time sink & frequency sink

Step 4: Interconnect all the block of adjusting the parameters.

QT GUI Range

To vary the frequency range of i/p signal source QT GUI Range can be used

Step 1: Search QT GUI Range.

Step 2: Drag & drop QT GUI range into the workspace

Step 3: Double click the box & the change adjust the properties

Step 4: Similarly select another QT GUI range for second source

Step 5: Run the experiment.

Multiply the sinusoidal signal with varying freq.

Step 1: Open terminal & type gnu radio companion.

Step 2: Take search bar & type signal source.

Step 3: Drag two signal source to display window

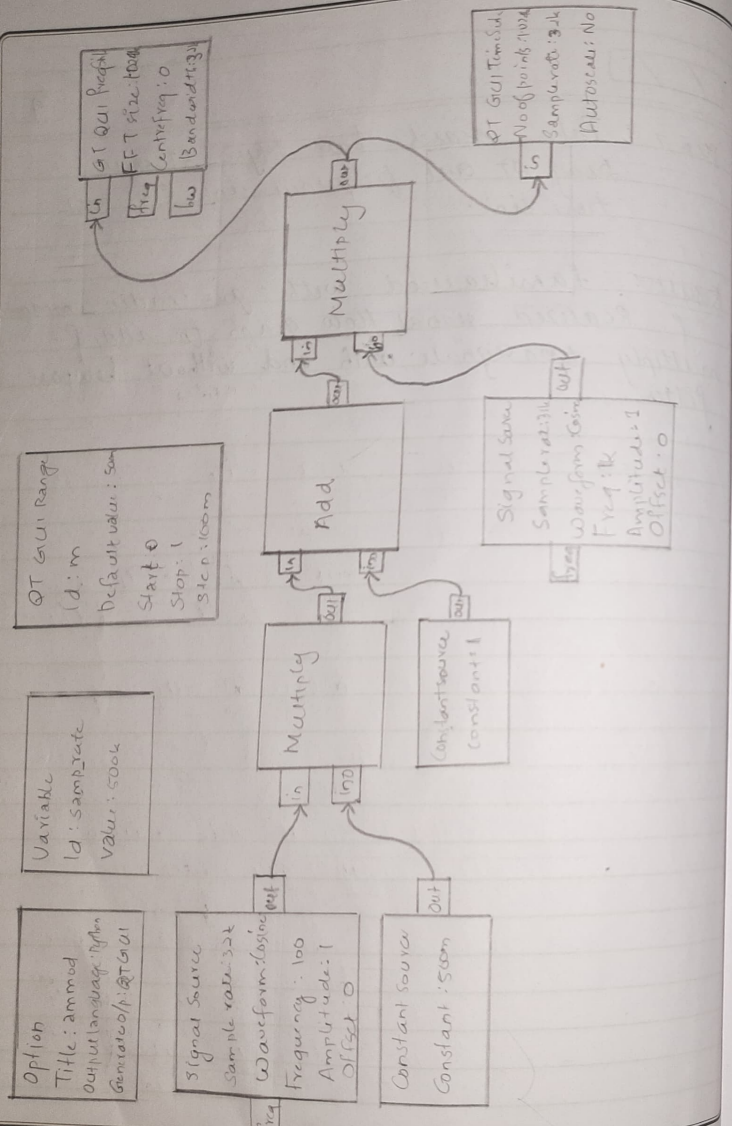
Step 4: Double click on the signal source & change properties.

Step 5: Take search bar & type multiply

Step 6: Double click to multiply & change properties

Step 7 : Take search bar & type QT GUI. Drag QT GUI frequency sink & QT GUI time sink.

Result : Familiarized with gnu radio companion & Realized signal flow graph to add & multiply two signals with and without Lowpass filter



8/1/25

10

EXPERIMENT NO: 2A

AMPLITUDE MODULATION

Aim:

To realize an am wave using gnu radio and observe the output.

Theory:

Modulation is defined as the process by which some characteristics of a carrier signal is varied in accordance with modulating signal. The base band signal is refferd to as modulating signal & the o/p of the modulation process is called modulated signal.

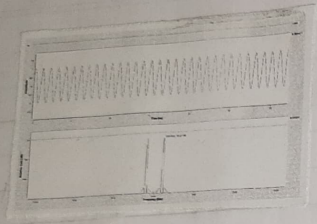
Amplitude modulation is defined as the process in which the amplitude of the carrier wave is varied with the instantenous values of the message signal. This technique is also called Double side band with full carrier scheme (DSBFC).

General Equation:

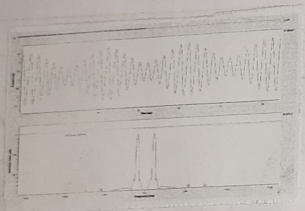
$$S_{AM}(t) = A_c (1 + k_a m(t)) \cos 2\pi f_c t$$

If the type of modulation is single tone modulation then message signal is replaced by single tone $m(t) = A_m \cos 2\pi f_m t$.

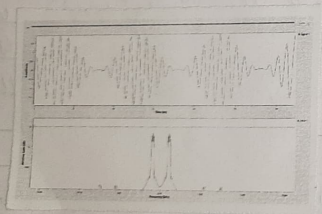
M = 0



M = 0.5



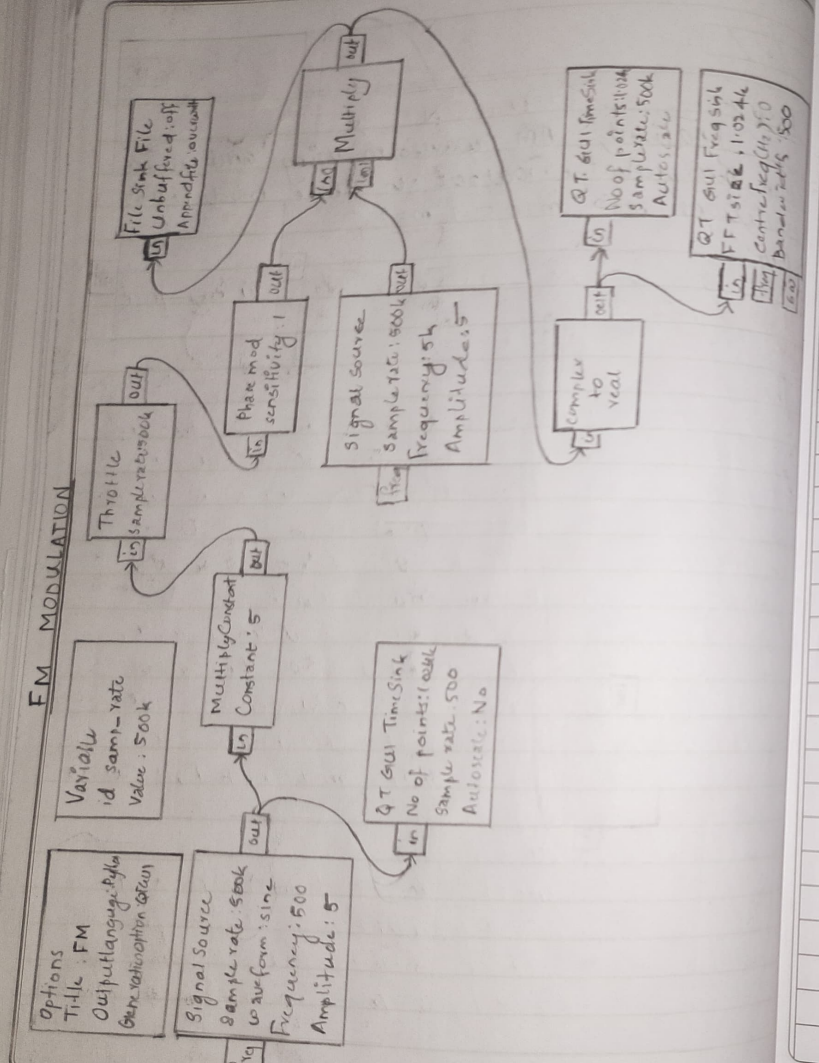
m = 1



Result

Realized AM waveform using GNU radio and observed the output in three modulation conditions m = 0, m = 0.5 & m = 1

17/2/25



22/01/25

12

EXPERIMENT NO: 2B

FM MODULATION AND DEMODULATION

Aim:

(i) Realize fm modulated sig using gnuradio & observe the o/p

(ii) Realize fm demodulated sig using gnuradio & observe the o/p.

Theory:

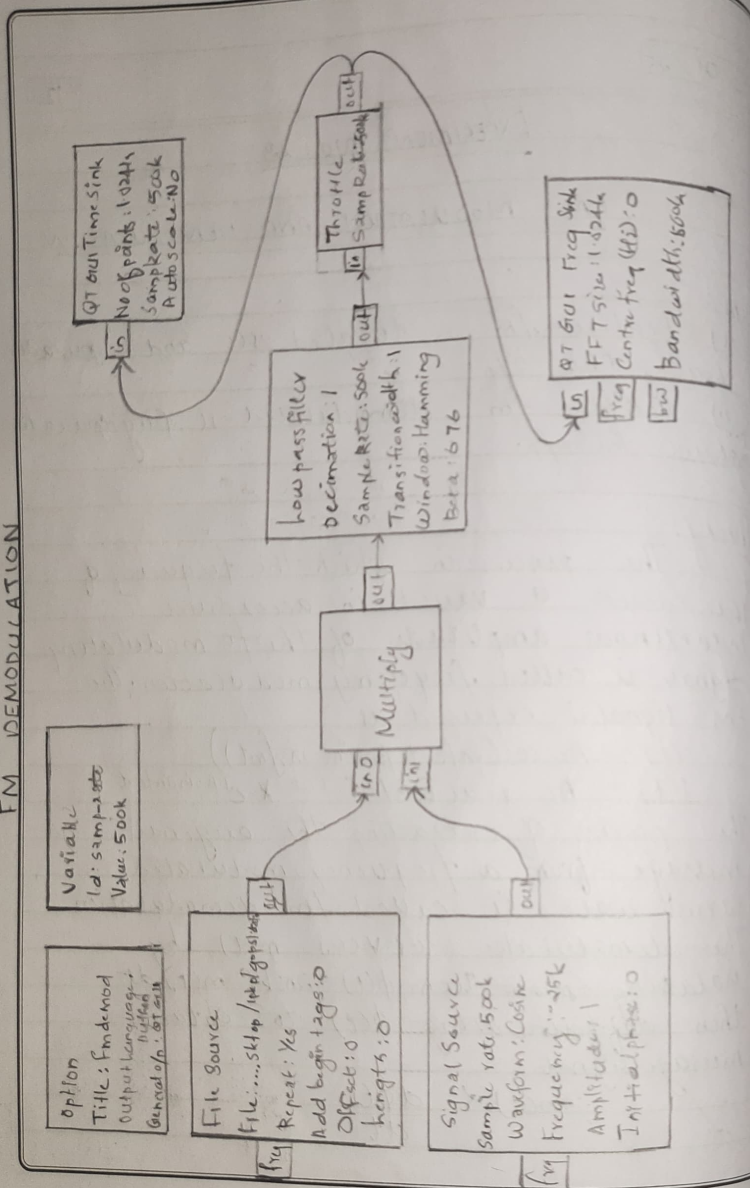
The process in which the frequency of the carrier is varied in accordance instantaneous amplitude of the fo modulating signal is called frequency modulation. The FM signal is expressed as

$$s(t) = A_c \cos(2\pi f_c t + \beta \sin(2\pi f_m t))$$

$$s(t) = A_c \text{ real}[e^{j2\pi f_c t} * e^{j\beta \sin 2\pi f_m t}]$$

The process of extracting the original message from a frequency modulated carrid wave is called fm demodulation The demodulator ~~recovers~~ m(t) by isolating phase term $\phi(t) = 2\pi k_f \int m(t)\,dt$ then differentiating $\phi(t)$ to obtain message signal

$$m(t) \propto \frac{d\phi(t)}{dt}$$

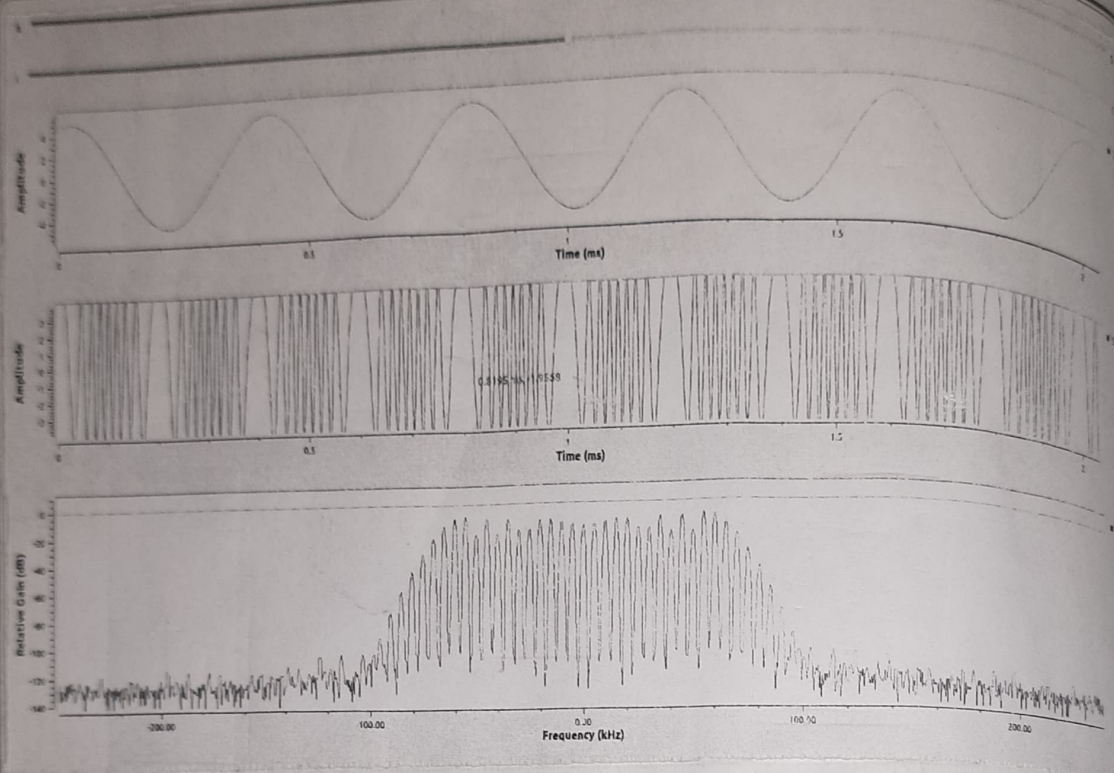


For demodulating we ~~the~~ save the fm modulated signal in a file. The retrieved signal is then multiplied by a cosine wave for down conversion. The resultant signal is then passed through a low pass filter to remove high frequency signal and extract baseband signal.

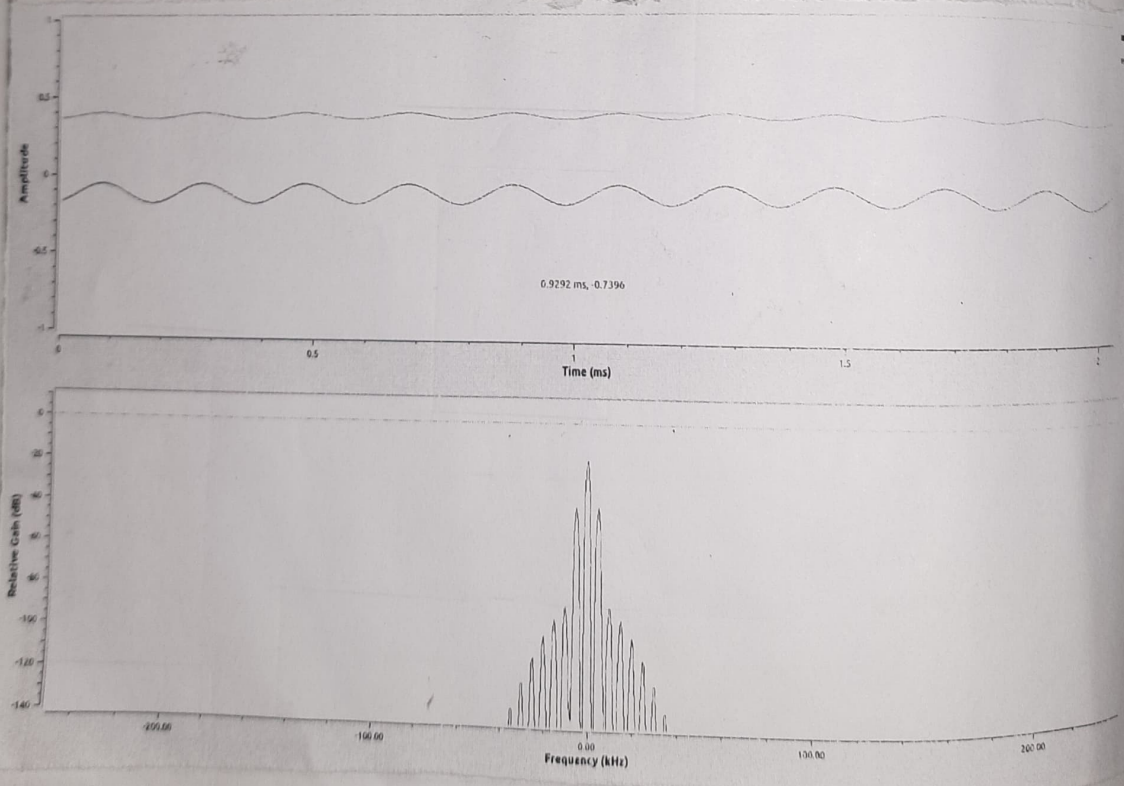
Another method is, the demodulator recovers m(t) by isolating phase term $\phi(t) = 2\pi k_f \int m(t)\, dt$ then differentiating $\phi(t)$ to obtain original signal

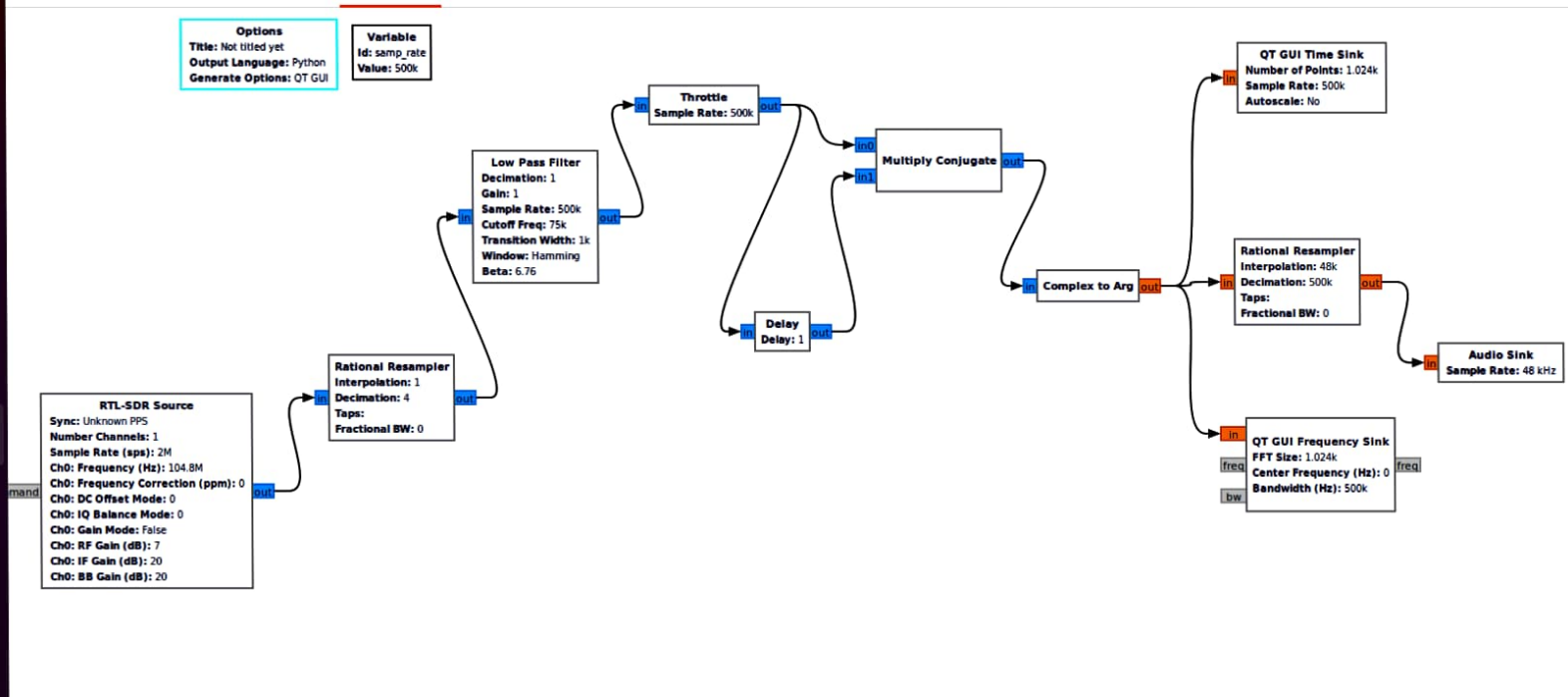
$$m(t) \propto \frac{d}{dt}\phi(t)$$

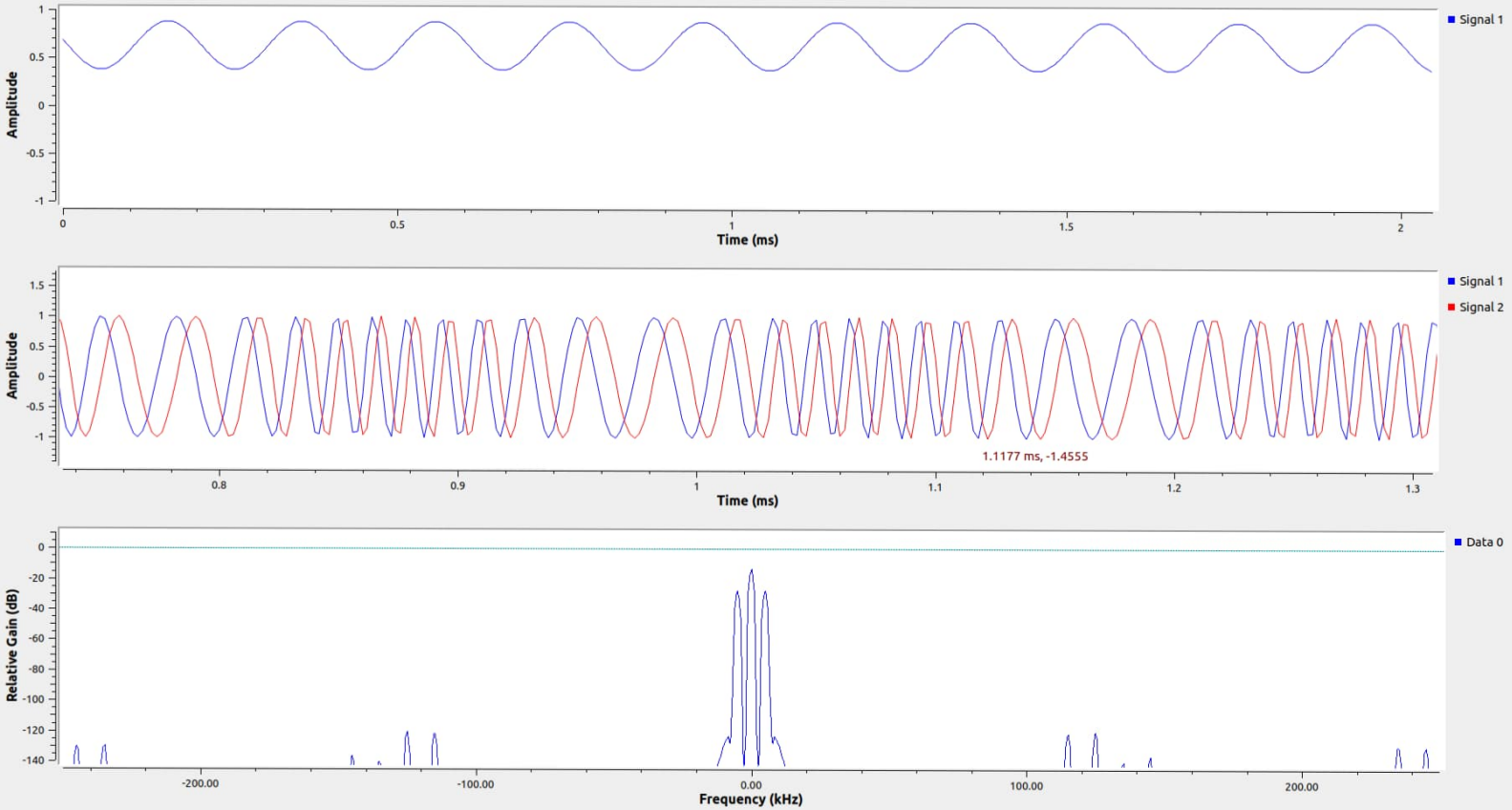
FM DEMODULATION



FM DEMODULATION




Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI
Variable
Id: samp_rate
Value: 500k
RTL-SDR Source
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 104.8M
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 7
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20
Rational Resampler
Interpolation: 1
Decimation: 4
Taps:
Fractional BW: 0
Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 500k
Cutoff Freq: 75k
Transition Width: 1k
Window: Hamming
Beta: 6.76
Throttle
Sample Rate: 500k
Delay
Delay: 1
Multiply Conjugate
Complex to Arg
QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 500k
Autoscale: No
Rational Resampler
Interpolation: 48k
Decimation: 500k
Taps:
Fractional BW: 0
Audio Sink
Sample Rate: 48 kHz
QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 500k


Amplitude
Time (ms)
Signal 1
Amplitude
Time (ms)
1.1177 ms, -1.4555
Signal 1
Signal 2
Relative Gain (dB)
Frequency (kHz)
Data 0