

Options

Title: DCF77 Re... Modulation

Author: henningM1R

Output Language: Python

Generate Options: QT GUI

Complexity: 3.843mbal

Variable

ID: samp\_rate

Value: 192k

Import

Import: zmq

Variable

ID: freq\_DCF77

Value: 77.5k

Variable

ID: decimation

Value: 4

Import

Import: math

QT GUI Range

ID: hi\_thres

Label: upper Threshold

Default Value: 100

Start: 0

Stop: 500

Step: 1

QT GUI Range

ID: lo\_thres

Label: lower Threshold

Default Value: 2.5

Start: 500m

Stop: 50

Step: 100m

QT GUI Range

ID: gain1

Label: Gain1

Default Value: 200

Start: 1

Stop: 400

Step: 1

QT GUI Range

ID: gain2

Label: Gain2

Default Value: 200

Start: 1

Stop: 450

Step: 500m

QT GUI Range

ID: chip\_size

Label: Chip size

Default Value: 75

Start: 70

Stop: 80

Step: 1

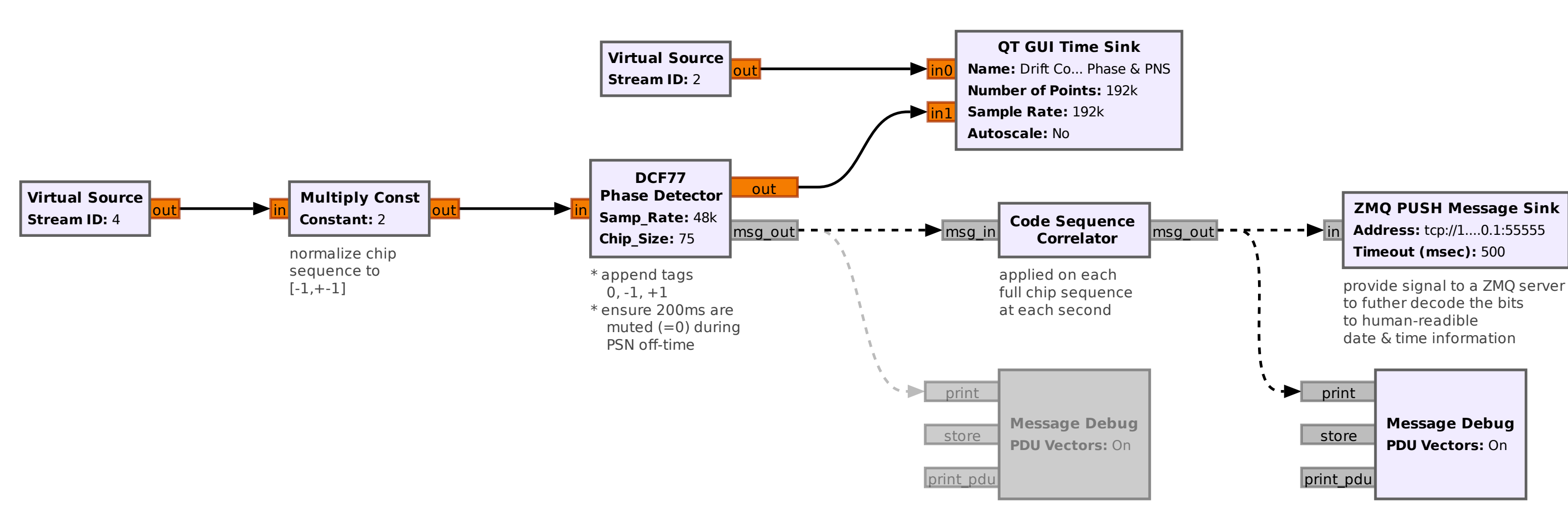
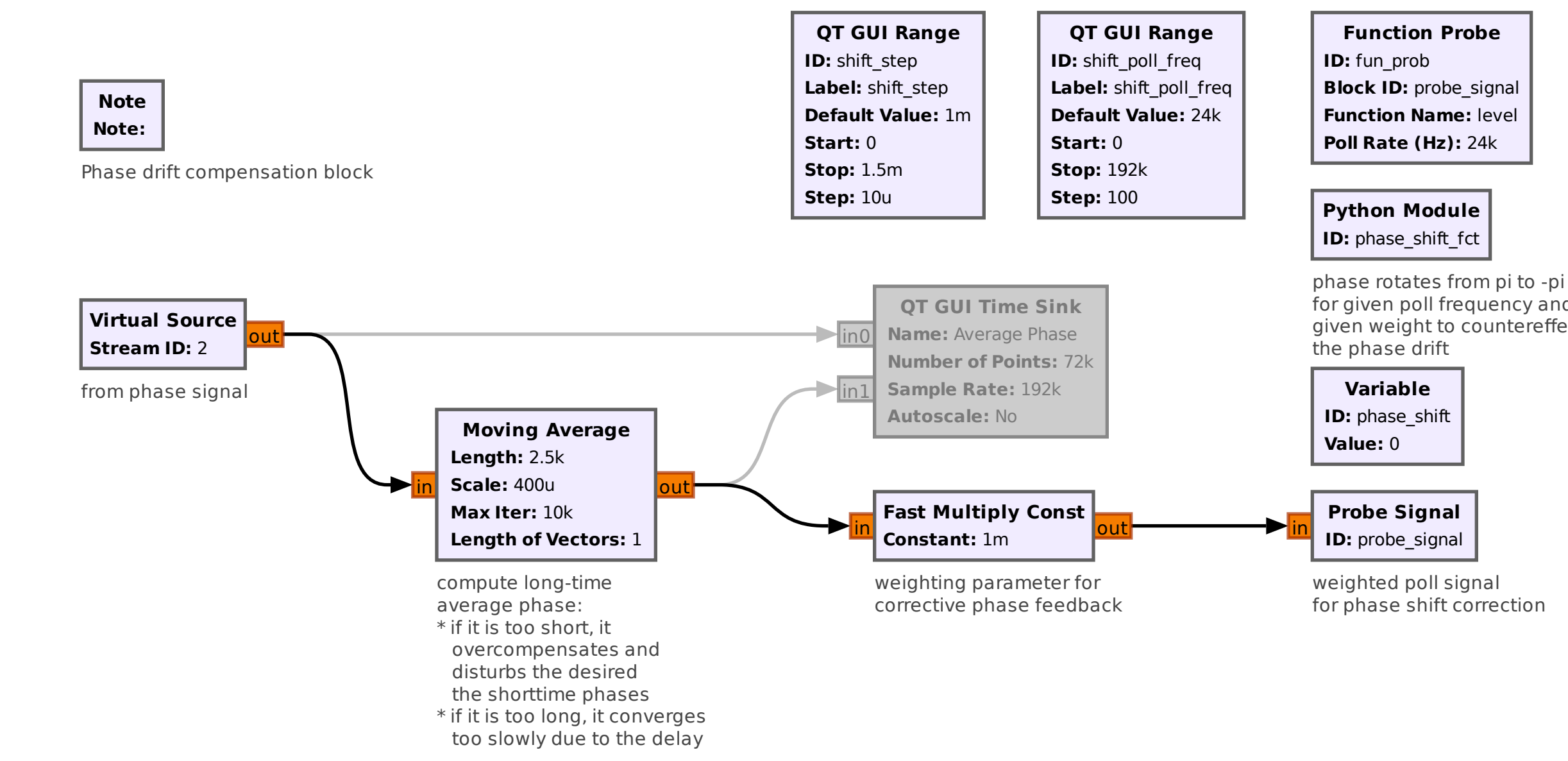
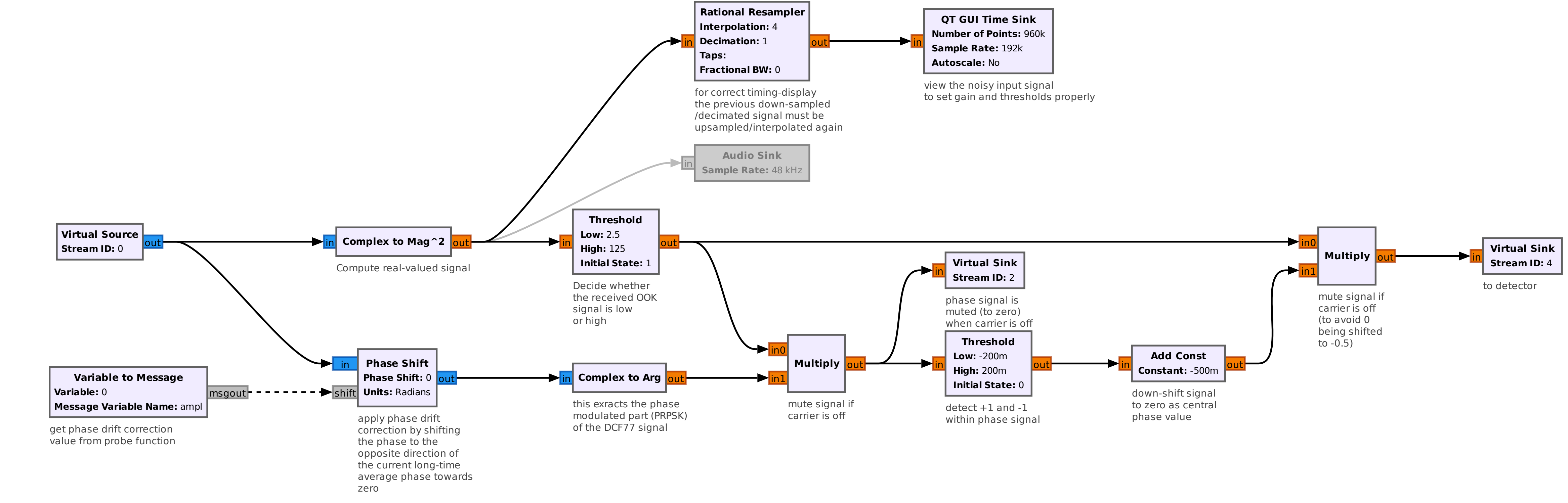
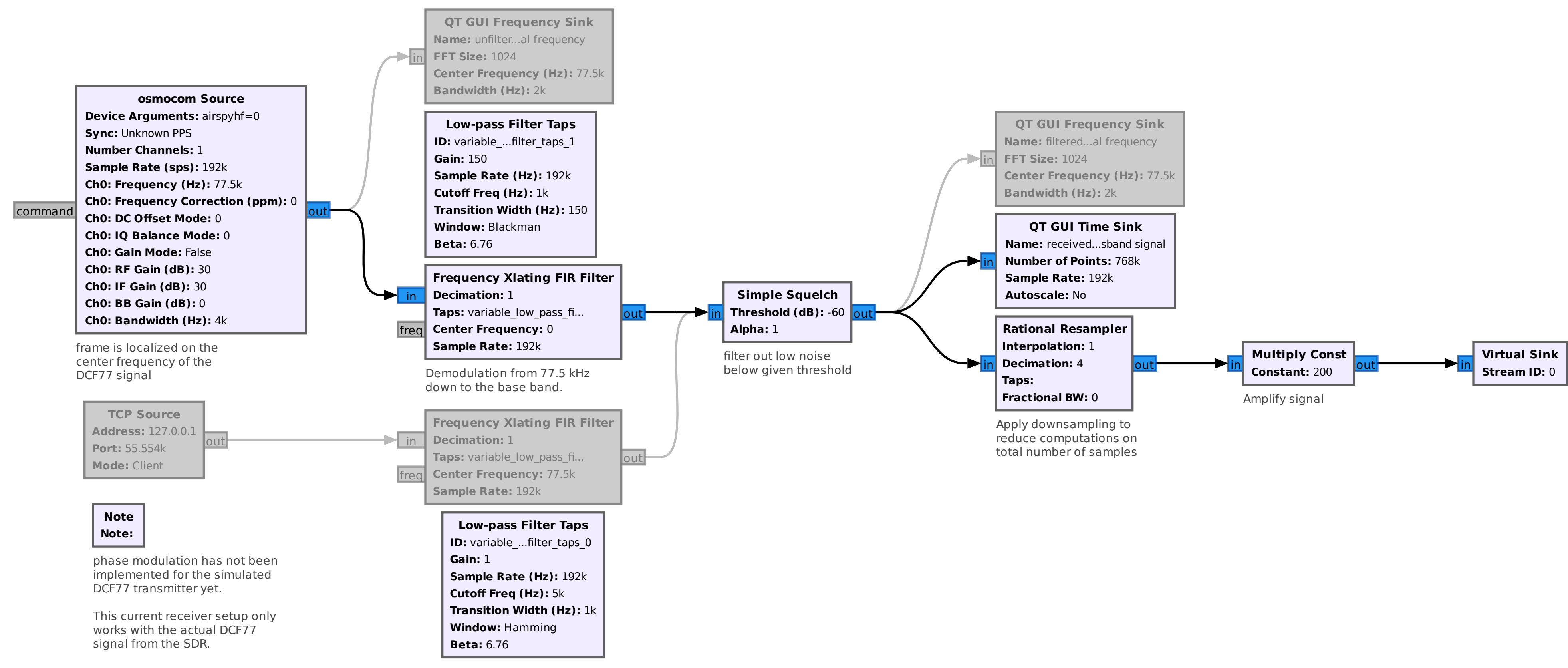
Adapt the upper threshold so that the one value is provided each time the signal is high

Adapt the upper threshold so that the one value is provided each time the signal is high

default real: 150  
default simulation: 1

default real: 290  
default simulation: 1

Windows needs the import of zmq



The official codeword for symbol 0 from the code-polynomial  $x^9+x^8+1$  is:

```
000001000110000100111001010101100001101111010011011100100010100001010110100
1111101100100100101101111100100110101001100110000000110001100101000110100
10111110100001011000111010110010110011100001111011010000011010110110110
11000001011010111101010101000000101001010111001011011000000111001101000
100111101011101010001001000#0110011100001011110110110011010000111011100001
111111100000111101111000101110011001000001001010011101101000111100111100
110110001010100100011100011011010101110001001100010001000000001
```

The opposite codeword for symbol 1 is its bit-wise inverted version

It has a number of:  
\* 256 ones  
\* 256 zeros

Except for the first two bits, it can be reproduced with the following python script.  
(maybe because the initial state=[0,0,0,0,0,0,0] is not permitted)

```
===
import numpy as np
from pylfsr import LFSR

state = [1,0,0,0,0,0,0,0]

fpoly = [9,5]
L = LFSR(fpoly=fpoly, conf='galois', initstate=state)
L.info()
L.runKCycle(511) # length of period
L.info()
seq = L.seq

===
```