

Options

Title: MSF60 Receiver

Author: henningM1r

Output Language: Python

Generate Options: QT GUI

Complexity: 590

ubal

Variable

ID: samp\_rate

Value: 192k

Variable

ID: freq\_MSF60

Value: 60k

Import

Import: zmq

Variable

ID: decimation

Value: 15

Windows needs the import of zmq

QT GUI Range

ID: gain

Label: Recv Gain

Default Value: 5k

Start: 1

Stop: 10k

Step: 1

Amplify receiving signal power

QT GUI Range

ID: thres

Label: Threshold

Default Value: 2.5k

Start: 0

Stop: 5k

Step: 10

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

QT GUI Range

ID: cutoff

Label: LP cut-off

Default Value: 10

Start: 5

Stop: 50

Step: 1

QT GUI Range

ID: width

Label: LP width

Default Value: 35

Start: 25

Stop: 150

Step: 1

Note

Note:

\* <https://drtonyr.github.io/MSF60decoder/>

\* [https://www.npl.co.uk/products-services/time-frequency/msf-radio-time-signal/msf\\_time\\_date\\_code](https://www.npl.co.uk/products-services/time-frequency/msf-radio-time-signal/msf_time_date_code)

\* [https://en.wikipedia.org/wiki/Time\\_from\\_NPL\\_\(MSF\)](https://en.wikipedia.org/wiki/Time_from_NPL_(MSF))

```
graph LR
    Command[command] --> Osmocom[osmocom Source]
    Osmocom -- out --> FIR[Frequency Xlating FIR Filter]
    Osmocom -- "freq" --> SinkFreq[QT GUI Frequency Sink]
    FIR -- out --> LPF[Low Pass Filter]
    LPF -- out --> Complex[Complex to Mag^2]
    Complex -- out --> SinkTime[QT GUI Time Sink]
    Complex -- out --> Const[Multiply Const]
    Const -- out --> SinkTime
    Const -- out --> Threshold[Threshold]
    Threshold -- out --> Bit[MSF60 Bit Detector]
    Bit -- out --> SinkTime
    Bit -- msg_out --> Correlator[MSF60 Correlator]
    Correlator -- msg_out --> SinkTime
    Correlator -- msg_out --> SinkZMQ[ZMQ PUSH Message Sink]
    Correlator -- msg_out --> Debug1[Message Debug]
    Signal[Signal Source] -- cmd --> Bit
    Signal -- out --> SinkTime
    SinkTime --> SinkZMQ
    SinkZMQ --> Debug2[Message Debug]
```

The diagram illustrates the architecture of an MSF60 Receiver system. It starts with an **osmocom Source** block, which receives a **command** and outputs to a **Frequency Xlating FIR Filter** and a **QT GUI Frequency Sink**. The FIR filter's output goes to a **Low Pass Filter**, which then feeds into a **Complex to Mag<sup>2</sup>** block. This block's output is split: one path goes to a **QT GUI Time Sink**, and the other goes to a **Multiply Const** block. The **Multiply Const** block's output is split: one path goes to the **QT GUI Time Sink**, and the other goes to a **Threshold** block. The **Threshold** block's output goes to an **MSF60 Bit Detector**. The **MSF60 Bit Detector** has two outputs: one goes to the **QT GUI Time Sink**, and the other goes to an **MSF60 Correlator**. The **MSF60 Correlator** has three outputs: one goes to the **QT GUI Time Sink**, one goes to a **ZMQ PUSH Message Sink**, and one goes to a **Message Debug** block. A **Signal Source** block, which receives a **cmd**, also feeds into the **MSF60 Bit Detector** and the **QT GUI Time Sink**. The **ZMQ PUSH Message Sink** and another **Message Debug** block are also shown.

**osmocom Source**  
Device Arguments: airspyhf=0  
Sync: Don't Sync  
Number Channels: 1  
Sample Rate (sps): 192k  
Ch0: Frequency (Hz): 60k  
Ch0: Frequency Correction (ppm): 0  
Ch0: DC Offset Mode: 0  
Ch0: IQ Balance Mode: 0  
Ch0: Gain Mode: True  
Ch0: RF Gain (dB): 35  
Ch0: IF Gain (dB): 35  
Ch0: BB Gain (dB): 35  
Ch0: Bandwidth (Hz): 4k

frame is localized on the center frequency of the MSF60 signal

**QT GUI Frequency Sink**  
Name: received...nal spectrum  
FFT Size: 1024  
Center Frequency (Hz): 0  
Bandwidth (Hz): 192k

**Frequency Xlating FIR Filter**  
Decimation: 15  
Taps: variable\_low\_pass\_fi...  
Center Frequency: 0  
Sample Rate: 192k

Demodulation from 60 kHz down to the base band.

**Low-pass Filter Taps**  
ID: variable\_...filter\_taps\_1  
Gain: 500  
Sample Rate (Hz): 192k  
Cutoff Freq (Hz): 600  
Transition Width (Hz): 150  
Window: Rectangular  
Beta: 6.76

**Low Pass Filter**  
Decimation: 1  
Gain: 5k  
Sample Rate: 12.8k  
Cutoff Freq: 10  
Transition Width: 35  
Window: Blackman  
Beta: 6.76

dense low-pass filter

**Virtual Source**  
Stream ID: 1

**Threshold**  
Low: 2.5k  
High: 2.5k  
Initial State: 1

Decide if received value is zero or one

**Complex to Mag<sup>2</sup>**

**Multiply Const**  
Constant: 3.5k

is only scaled to be visible in output

**MSF60 Bit Detector**  
Sample\_Rate: 12.8k  
Tolerance: 30m

Detects MSF60 OOK Symbols:  
\* second marker  
\* minute marker  
\* A bit (0 or 1)  
\* B bit (0 or 1)

**Signal Source**  
Sample Rate: 192k  
Waveform: Constant  
Frequency: 1  
Amplitude: 2.5k  
Offset: 0  
Initial Phase (Radians): 0

only to display threshold

**QT GUI Time Sink**  
Name: Signals  
Number of Points: 51.2k  
Sample Rate: 192k  
Autoscale: No

view the noisy input signal to set the thresholds properly

**MSF60 Correlator**

**ZMQ PUSH Message Sink**  
Address: tcp://1....0.1:55555  
Timeout (msec): 500

publish to be read by the python tool via ZMQ for instance

**Message Debug**  
PDU Vectors: On

**Message Debug**  
PDU Vectors: On