Communication Design Project

Radio-Heads

Introduction

- This presentation will discuss the implementation of a point-to-point digital wireless communication system using GNU Radio, a free and open-source software development toolkit for building software-defined radios (SDRs).
- We will explore the transmission of bit streams, files, images, and audio in real-time, highlighting the capabilities and advantages of GNU Radio for such tasks.
- Its open-source nature allows for customization, while the availability of pre-built signal processing blocks simplifies the development process.

Requirements

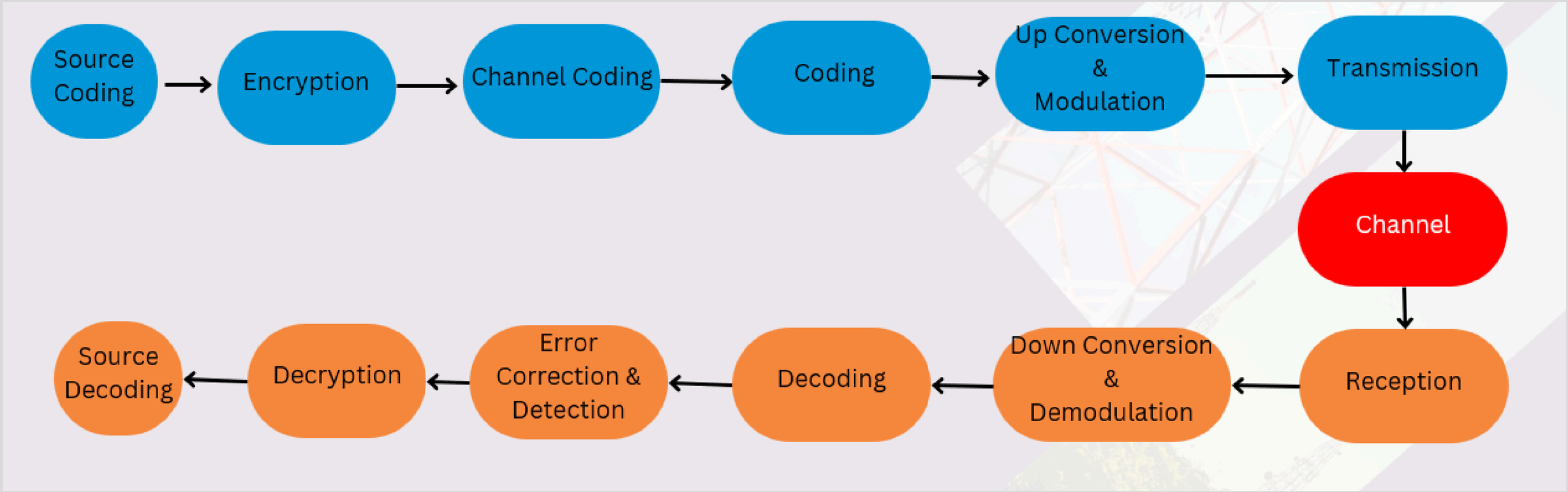
1.Transmitter design-

- Encoding
- Security implementation
- Reliability implementation
- Jamming protection
- Modulation

2.Receiver design-

- Reliability enhancement
- Demodulation
- Decoding

Block diagram



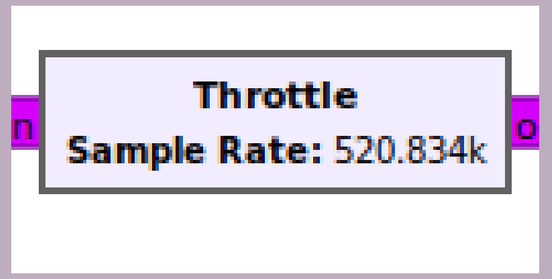
Methodology

1. Source/ sink

File source - Reads data from a file and outputs a stream of bytes.

File sink - Writes a stream of bytes to a file.

Throttle - Regulate the rate at which samples are processed, preventing overflow



2. Encryption/ Decryption

Custom Python blocks are used to handle encrypting and decrypting tasks.

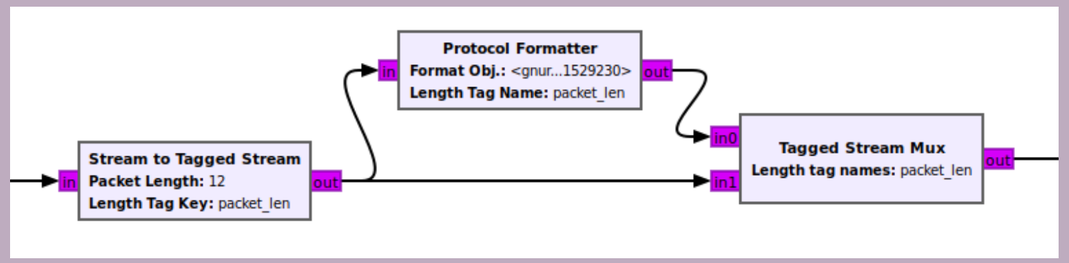
gr.sync block is used

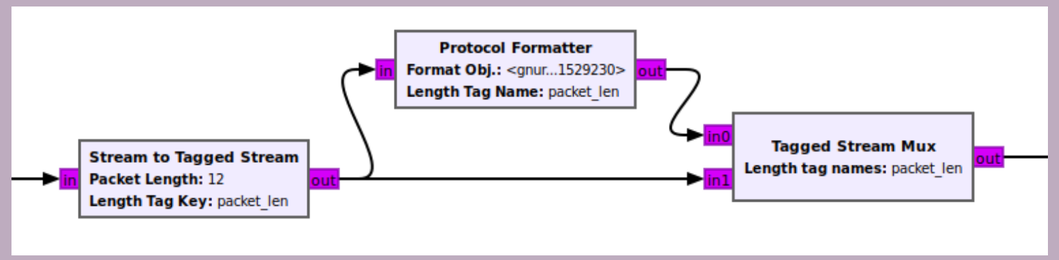
Cesar’s cipher algorithm

add an 8 bit fixed data to the incoming stream

3. Encoder / Decoder

- Stream CRC32 - Calculates and adds a 32-bit cyclic redundancy check (CRC) to the input data stream. This helps detect errors in transmission.
- Protocol formatter - Formats the input data according to a specific protocol. This involve adding headers, trailers, or other formatting information.
- Tagged stream mux - Multiplexes multiple input streams into a single output stream. Each input stream is tagged so that the receiver can identify which stream each data item belongs to.
- Correlate Access Code - Tag Stream - Correlates the input stream with a known access code. This is used at the receiver for synchronization purposes in some protocols.
- Repack bits - Changes the size of each bit in the input stream. This can be used to match the bit rate of the transmission medium.





4. Modulation/ Demodulation

- Modulation schemes used are Frequency Shift Keying (FSK) and Gaussian Frequency Shift Keying (GFSK)
- These are better than QPSK:

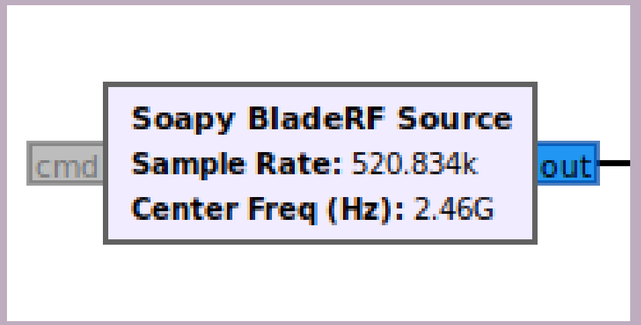
Simplicity: Relatively simple modulation scheme to implement in software.
Robustness to noise: Generally more robust to noise than QPSK.

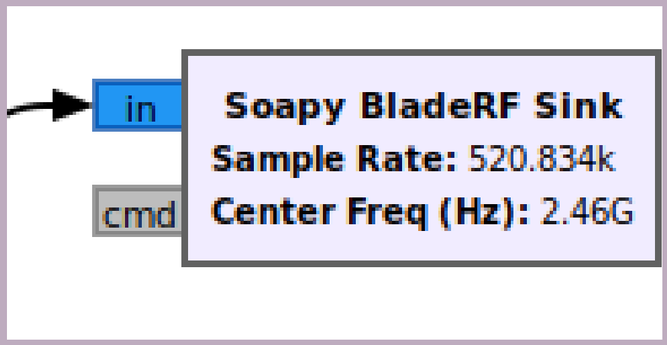
5. Transmission/ Reception

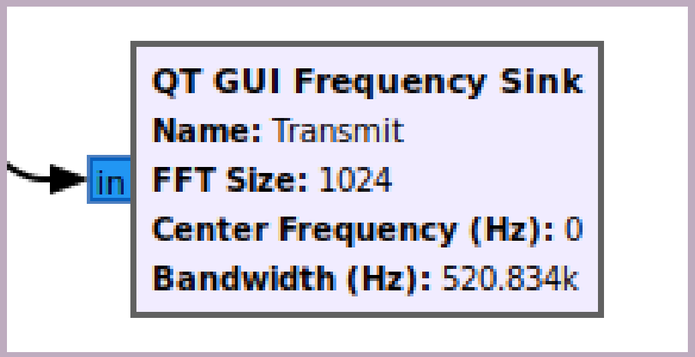
- Soapy bladeRF source - Receives data from a bladeRF SDR device.
- Soapy bladeRF sink - Transmits data to a bladeRF SDR device.

6. Visualization blocks

QT GUI frequency sink - Displays the spectrum of the input signal in a graphical user interface (GUI).





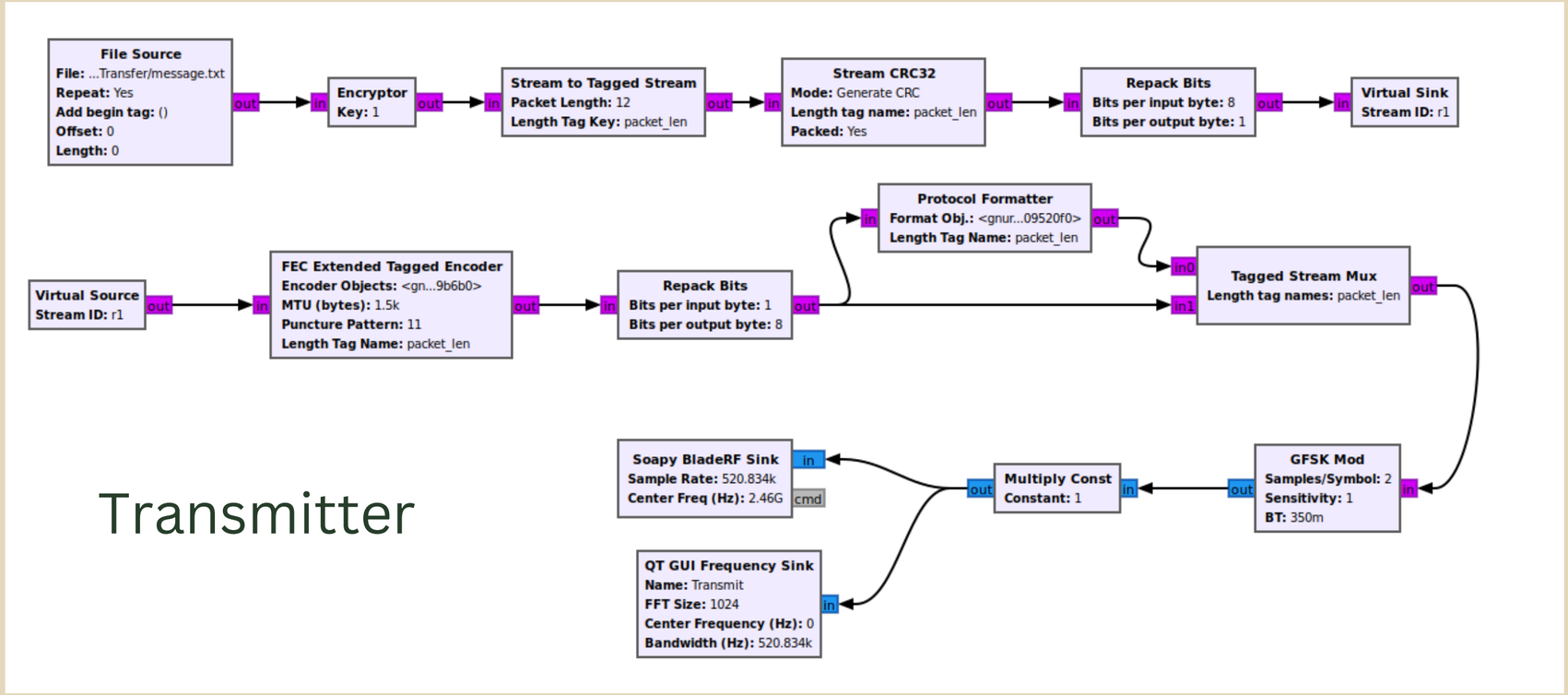


Bit Stream Transfer

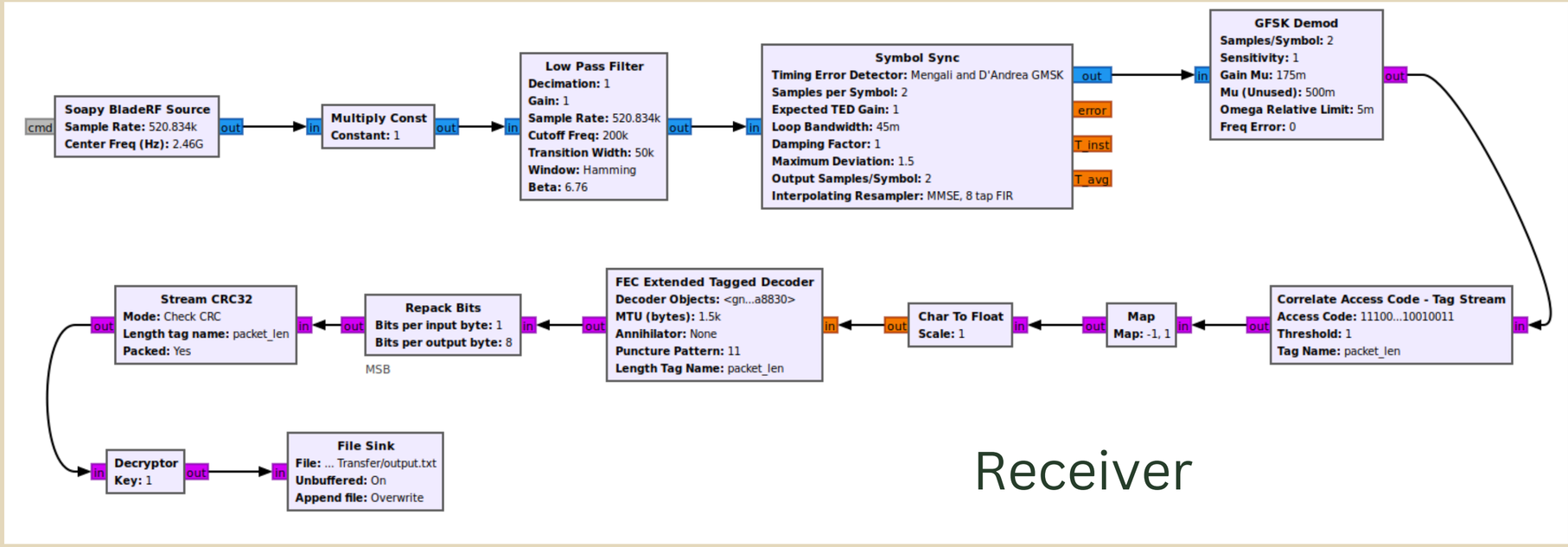
Flow graph overview

- This GNU Radio flowgraph implements a file transfer system using Gaussian Frequency Shift Keying (GFSK) modulation. The process begins with reading a bitstream from the "message.txt" file, encoding it, and modulating the signal with GFSK.
- The modulated signal undergoes filtering, simulating channel effects such as noise, timing offset, and frequency offset. After demodulation and symbol synchronization, the flowgraph performs access code correlation and CRC checking for packet detection and error detection, respectively. The received bitstream is then decoded, and the resulting data is saved to the "output.txt" file.

Flow graph



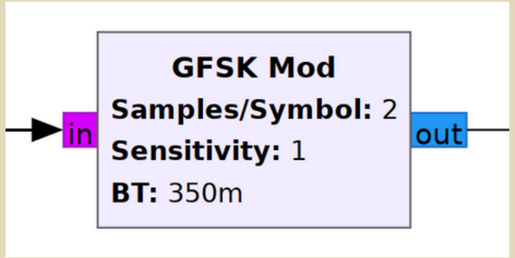
Transmitter



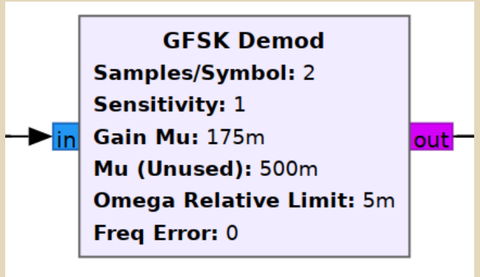
Receiver

Blocks and Parameters

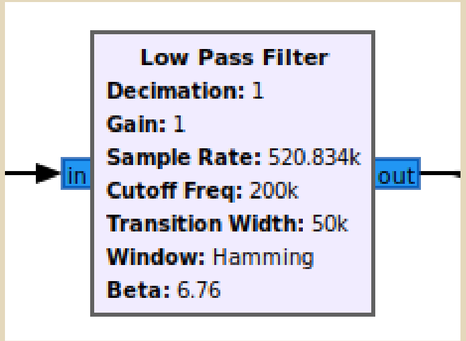
- GFSK mod - Modulates the formatted data using Gaussian Frequency Shift Keying (GFSK) modulation. The output of the GFSK Mod block is a modulated RF signal that carries the encoded audio data.



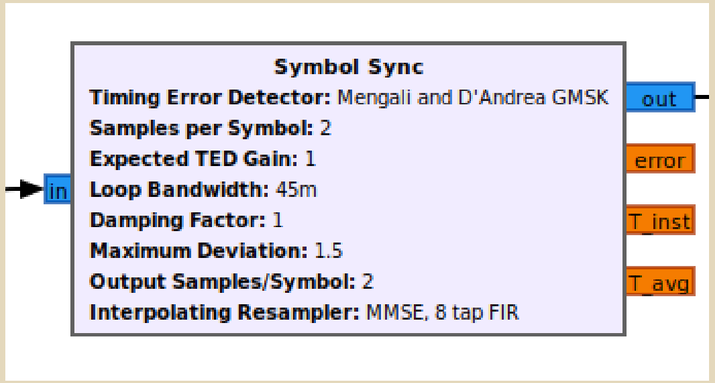
- GFSK demod - This block on the receiving end demodulates the GFSK signal back into a stream of bits.



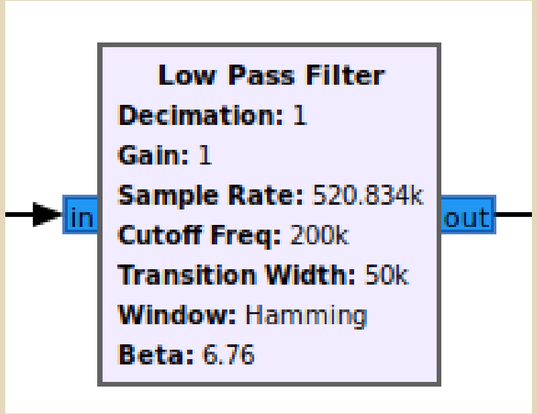
- Low pass filter - Filters the modulated RF signal output from the GFSK Mod block to remove unwanted high-frequency components.



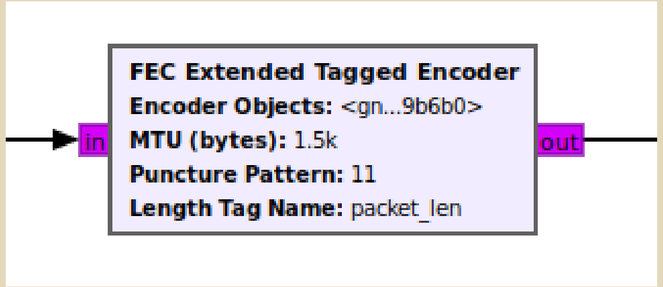
- Symbol sinc - Performs symbol timing synchronization, also known as clock recovery. Its purpose is to extract the timing information from a received signal and align the samples to the center of the individual symbols (data bits).



- Low pass filter - Filters the modulated RF signal output from the GFSK Mod block to remove unwanted high-frequency components.



- FEC extended tagged encoder - Adds forward error correction (FEC) capabilities. In this case, the FEC Extended Tagged Encoder block is configured to use a repetition code with a repetition factor of 3.



Why use GFSK?

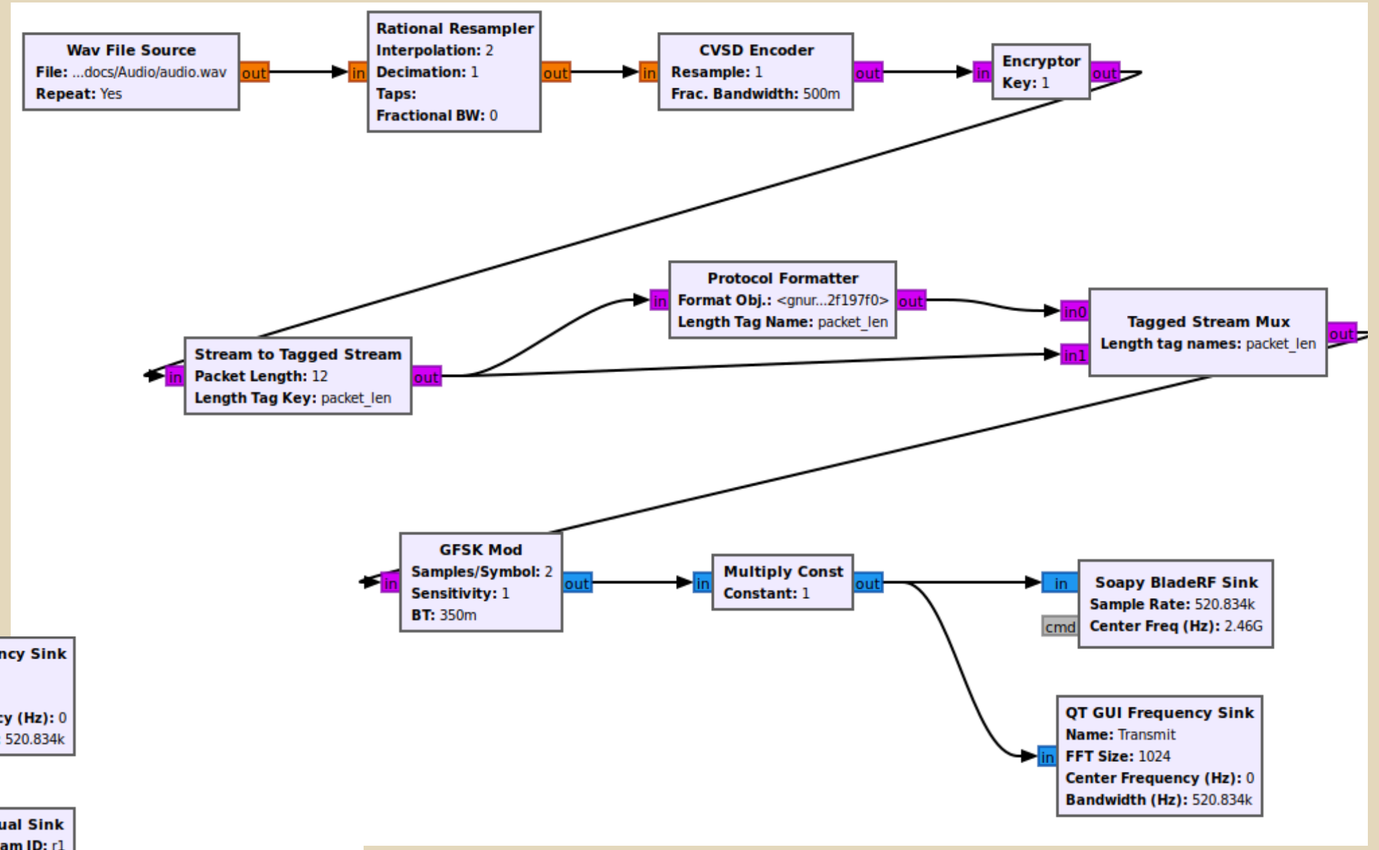
- Simplicity: FSK is a relatively simple modulation scheme to implement in software.
- Robustness to noise: FSK is generally more robust to noise than QPSK or GFSK.
- Faster Data Rate: GFSK is generally faster than FSK, allowing it to transmit more data per unit time.
- Reduced Bandwidth Expansion: GFSK exhibits reduced bandwidth expansion compared to FSK

Audio File Transfer

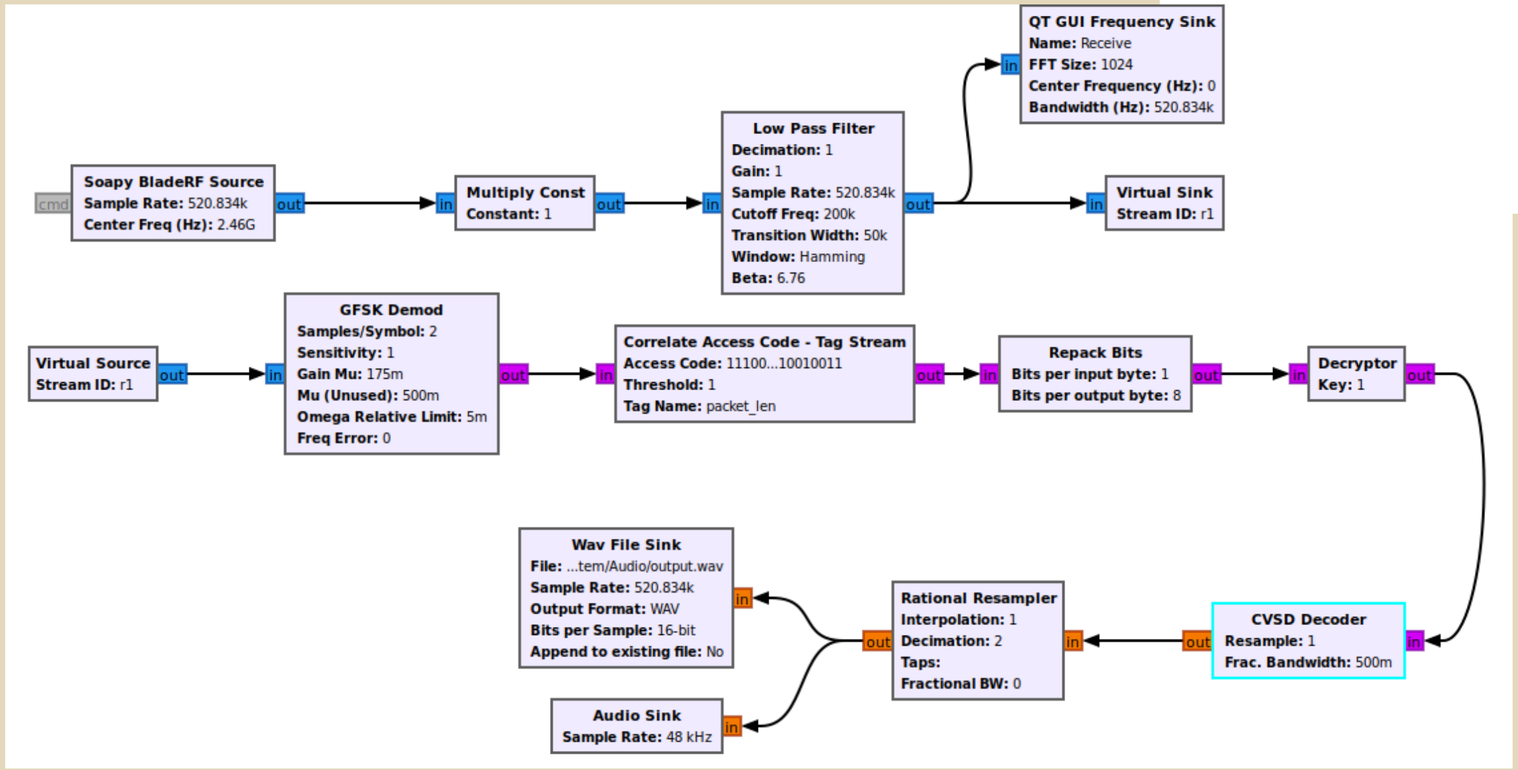
- This communication system demonstrates the transmission of audio data using Gaussian Frequency Shift Keying (GFSK) modulation and the bladeRF software-defined radio platform. The system effectively converts audio signals into a digital bit stream, modulates it using GFSK, and transmits it through bladeRF. At the receiver end, the GFSK signal is demodulated, noise is removed, and error checking is performed to reconstruct the original audio data.

Flow graph

Transmitter

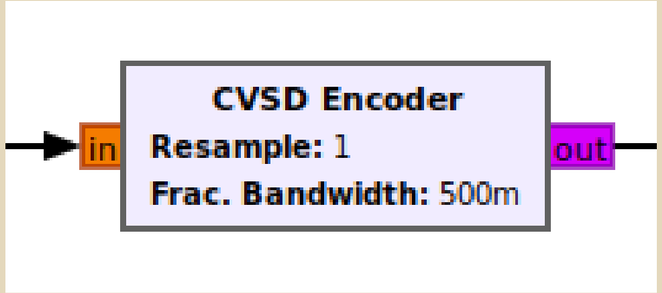


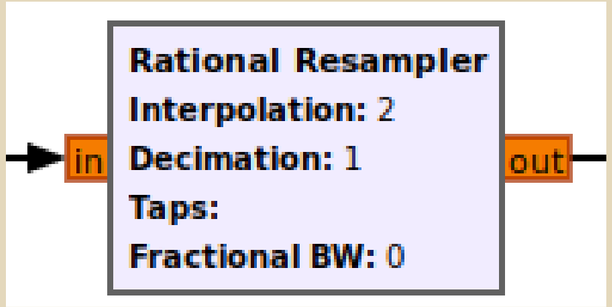
Receiver

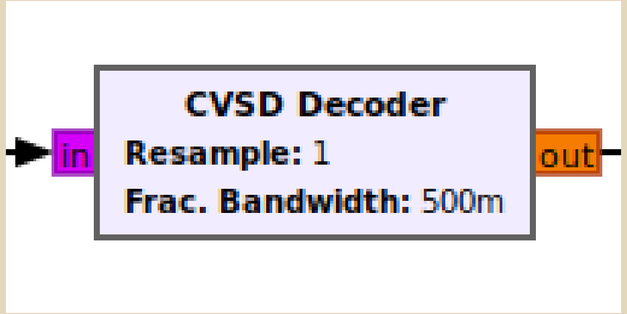


Blocks and parameters

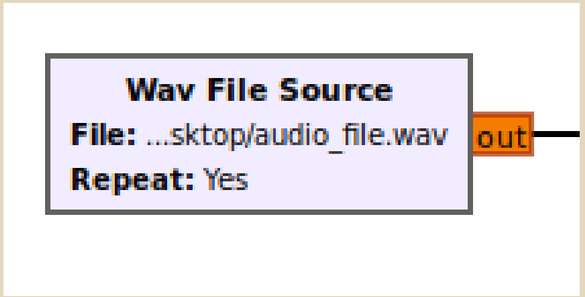
- CVSD Encoder - Responsible for encoding an incoming audio signal into a bitstream using the CVSD (Continuously Variable Slope Delta) modulation scheme. This represents the audio signal by encoding the changes in amplitude between consecutive samples.
- Rational resampler - Changes the sampling rate of a signal using a polyphase filterbank. It can both increase (interpolating) and decrease (decimating) the sampling rate by any rational factor, making it highly versatile for various applications like upsampling for filtering, downsampling for data reduction, and adjusting sample rates for specific tasks.
- CVSD Decoder - Decode the filtered RF signal output from the Low Pass Filter block using the CVSD modulation scheme.



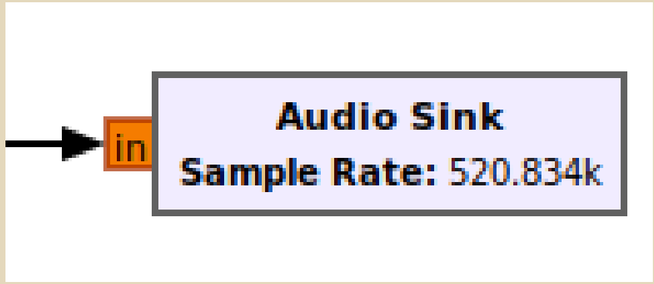




- Wav file source - Source of audio data, reading and providing samples from a WAV file.



- Audio sink - Converts the received decoded audio samples from the CVSD Decoder block into a format that can be played back by an audio output device, such as speakers or headphones.



- Low pass filter - Filters the modulated RF signal output from the GFSK Mod block to remove unwanted high-frequency components.

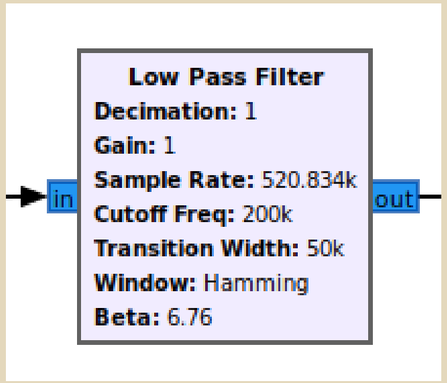


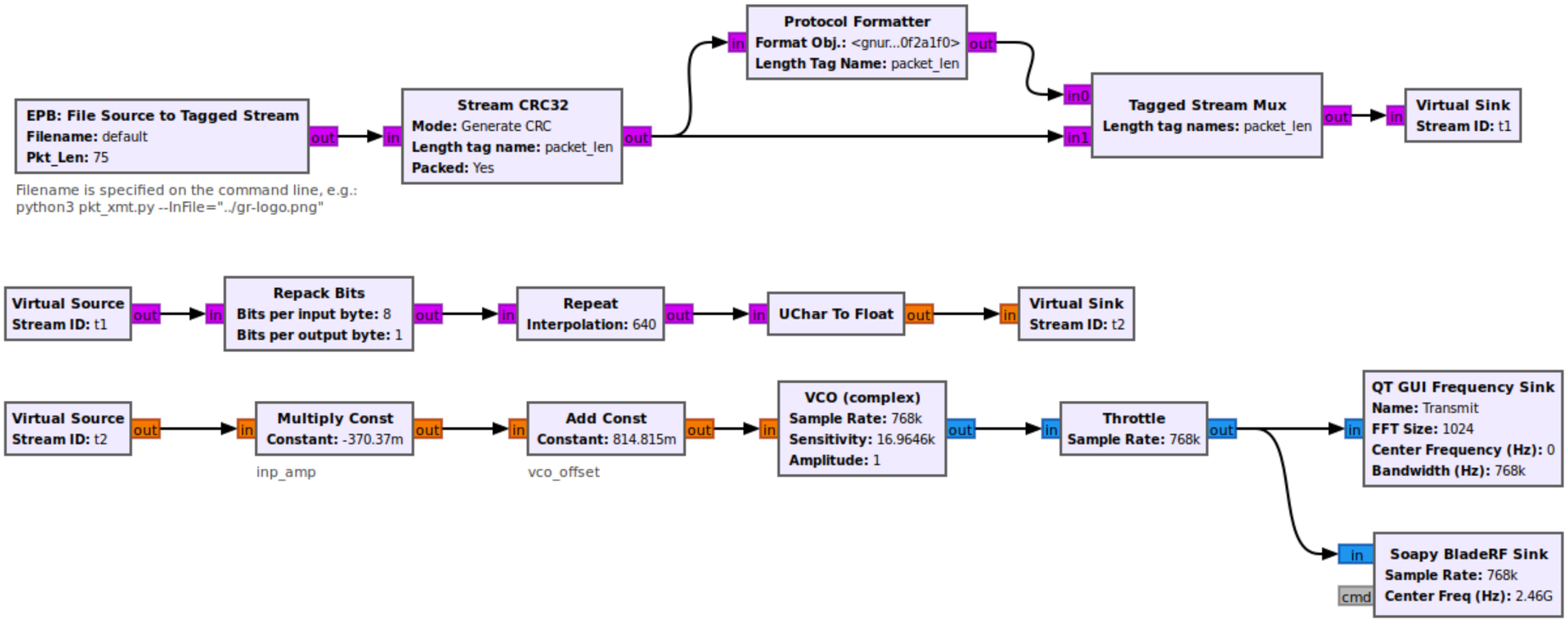
Image File Transfer

- This communication system implements an image transmission scheme utilizing Frequency Shift Keying (FSK) modulation through the bladeRF.

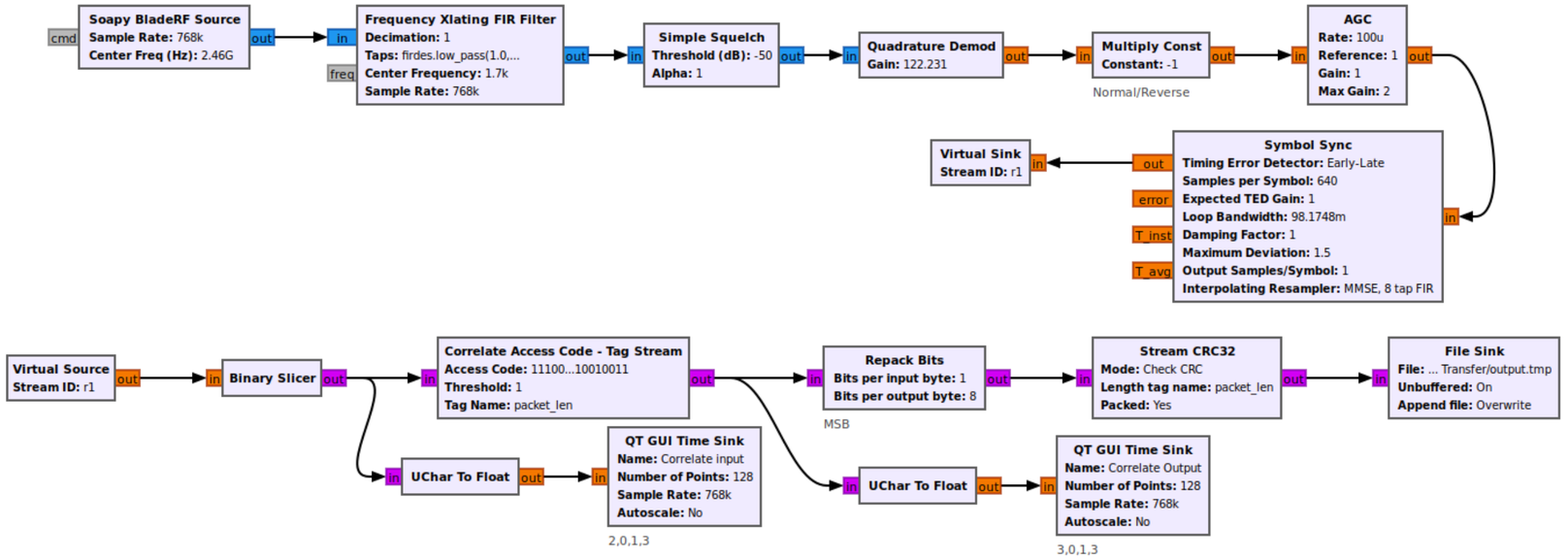
- The process involves bit stream generation, symbol conversion, repetition, and format adjustments for FSK modulation.

- The modulated signal is transmitted via bladeRF, and on the receiver end, FSK demodulation, noise removal, and error checking lead to image reconstruction.

Transmitter



Receiver

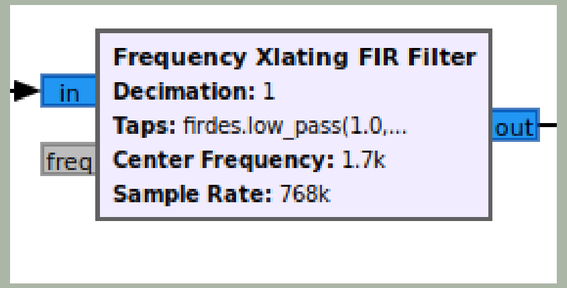


Why use FSK?

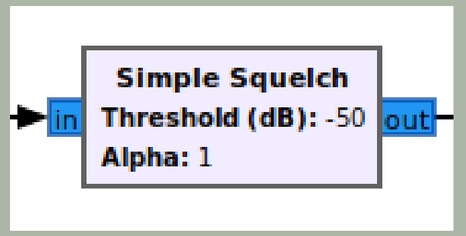
- Simplicity: FSK is a relatively simple modulation scheme to implement in software.
- Robustness to noise: FSK is generally more robust to noise than QPSK or GFSK.
- Accuracy: FSK has a much lower BER than GFSK.

Blocks and Parameters

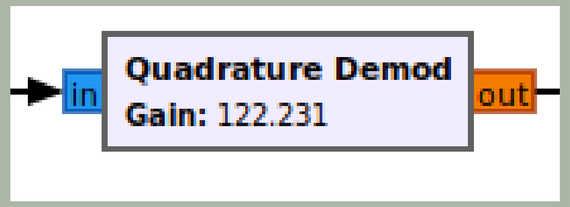
- Frequency Xlating FIR filter - Used for shifting the frequency spectrum of a signal without altering its fundamental characteristics. It achieves this by convolving the input signal with a carefully designed finite impulse response (FIR) filter, resulting in a modified signal with a shifted frequency range.



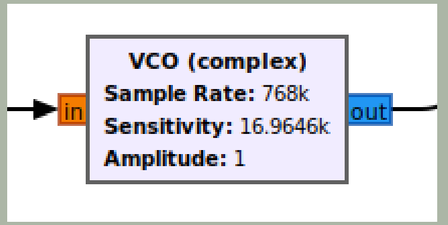
- Simple Squelch - Suppresses weak signals and allows strong signals based on average signal power and threshold.



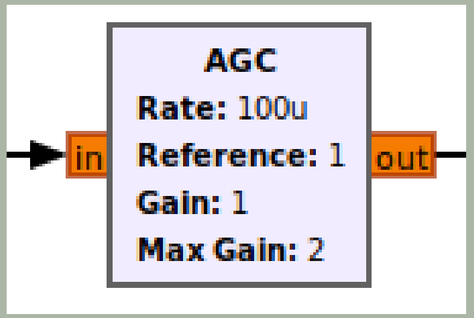
- Quadrature Demod - Demodulates complex signals. It is useful for demodulating FM, FSK, GMSK, and other modulations that use frequency changes to carry information. FSK is used as the modulation scheme for this image transfer.



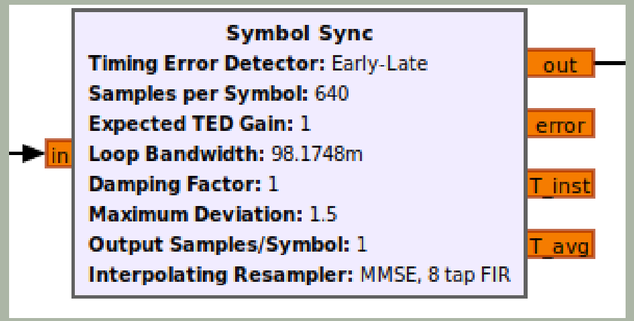
- VCO - Produces a sinusoid frequency based on input amplitude.



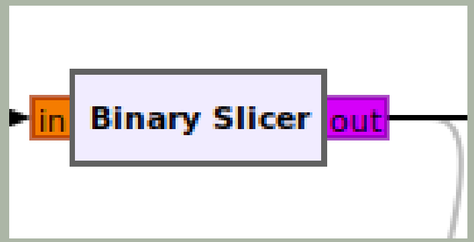
- AGC (Automatic Gain Control) - Dynamically adjusts signal path gain for optimal reception.



- Symbol Sinc - Accurately estimates the symbol timing of a received signal, applies a sinc function to the transmitted symbols, and acts as a matched filter for the sinc-shaped pulses.



- Binary Slicer - Compares input to a threshold for binary output. Slices float values, producing 1-bit output. Positive input produces a binary 1 and negative input produces a binary zero.



PERFORMANCE EVALUATION

File Loss Calculation

- We implemented a python script to calculate the loss between transmitted and received text file.
- We used Bit Error Rate as the loss criterion
- will run this during the demonstration

Methodology

- convert the text files to binary data
- If size does not match, adds zero padding to the smallest file
- Takes pair of bits from transmitted and received bits
- Get XOR of those pairs
- Get the sum

```
calculate_ber.py  message.txt  output.txt
thisara@thisara-PC:~/Downloads/CDP-communication-system/File Transfer/Bit Error
Loss _ Text$ python3 calculate_ber.py message.txt output.txt
Bit Error Rate (BER): 0.435689368839218
```

Image Loss Calculation (for “Monochrome” and “RGBA”)

- We implemented a python script to calculate the loss between transmitted and received image.
- We used Mean Squared Error as the loss criterion
- will run this during the demonstration

Methodology

- Iterate over both transmitted and received images
- Compare each pixel value of transmitted and received image
- Take the difference and Square it
- Get the sum of all those values
- Divide by transmitted image dimensions
- returns the value

- MSE FOR MONO-CHROME IMAGE OF SIZE M×N IS DEFINED AS

$$\text{MSE} = \sum_{i=1}^{M} \sum_{j=1}^{N} [T(i,j) - R(i,j)]^2$$

```
sasika@sasika-Inspiron-5502:~/Downloads/Mean Squared Error Loss$ python3 calcula
teImageLossARG.py thisara_in.png thisara_out.png
Total image loss: 0.0
```

TEAM

- W. M. T. V. GUNAWARDANA - 210198A
- Y. W. S. P. AMARASINGHE - 210035A
- M. J. AHAMETH - 210024N
- W. A. D. N. M. WANNIARACHCHI - 210679B

THANK YOU!