Communication Design Project

Team Radio 1

→ Next

Content of table

→

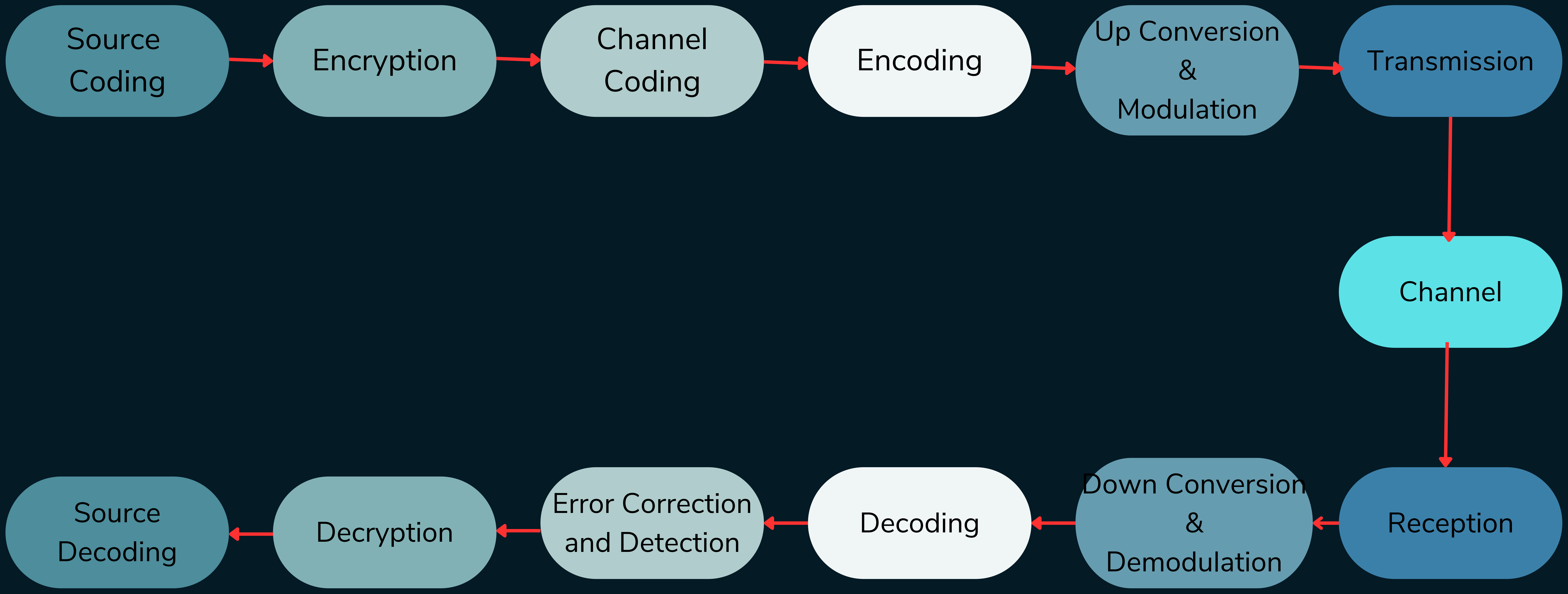
Design Requirements

SUCCESSFUL TRANSMISSION AND RECEPTION OF A BIT STREAM, AN IMAGE, AND REAL TIME VOICE STREAM

- WITHOUT JAMMING
- WITH JAMMING
- PERFORMANCE WITH THE DISTANCE
- WITH FREQUENCY OFFSET BETWEEN TRANSMITTER AND THE RECEIVER
- WITH TIMING OFFSET BETWEEN TRANSMITTER AND THE RECEIVER
- MULTIPATH EFFECT CORECTION

Design Methodology

Our Design


Source Coding
Encryption
Channel Coding
Encoding
Up Conversion & Modulation
Transmission
Channel
Reception
Down Conversion & Demodulation
Decoding
Error Correction and Detection
Decryption
Source Decoding

•DESIGNED A SYSTEM THAT CAN TRANSMIT DATA THROUGH A SIMULATED CHANNEL WITH BARE MINIMUM BLOCKS.

Software Used - GNU Radio

Flowgraphs - Transmission
Reception
Real-time transmission
Encryption

→

DESIGN A SYSTEM THAT CAN TRANSMIT DATA THROUGH A REAL CHANNEL USING SOFTWARE DEFINED TRANSMISSION AND RECEPTION EQUIPMENT.

EQUIPMENT USED - BLADERF

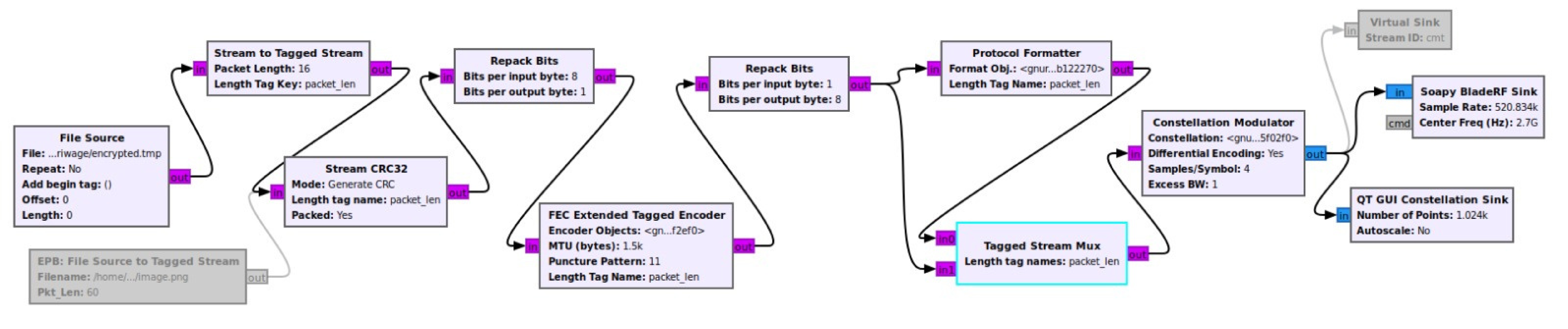
METHOD FOLLOWED -

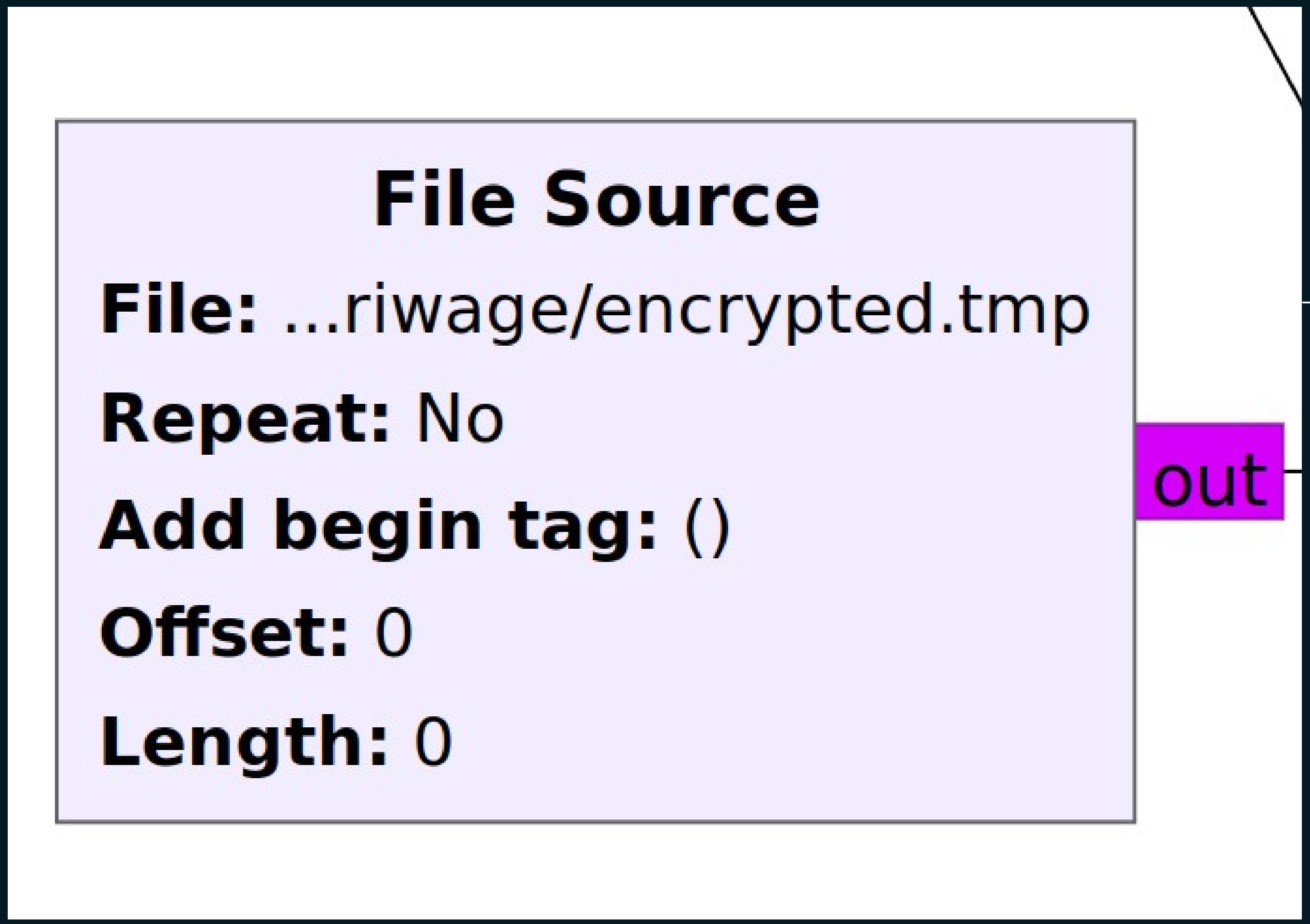
WE REPLACED CHANNEL BLOCK WITH THE TRANSMITTER AND THE RECEIVER BLOCKS FOR BLADERF AND CHANGED THE PARAMETERS AS REQUIRED.

ADDING NEW BLOCKS AND CHANGING PARAMETERS AS REQUIRED TO IMPLEMENT VARIOUS DESIGN REQUIREMENTS.

- Transmitting a bit stream
- Transmitting an image
- Transmitting a real time voice stream
- Jamming
- Varying distance
- Timing offset
- Frequency offset

Transmission Flowgraph



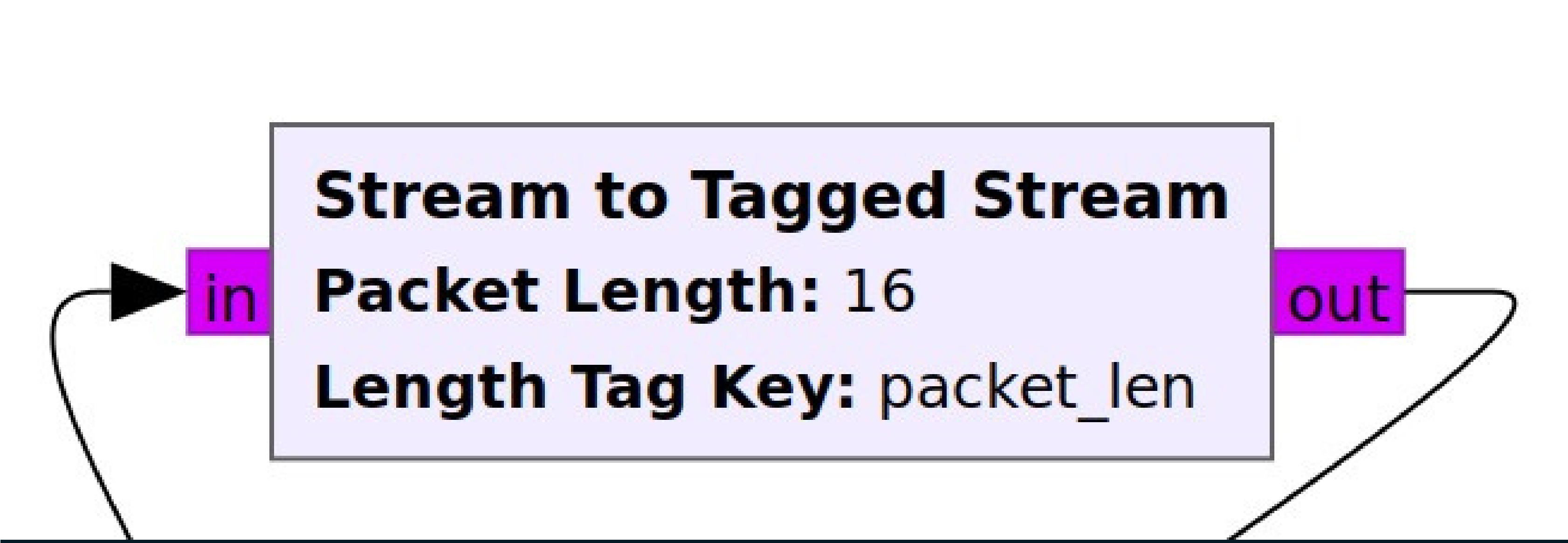


File Source

- In GNU Radio, the File Source block serves to read data from a file for signal processing.
- It can handle binary, ASCII, and complicated binary files, and can be configured with the file format, bits per sample, sample rate, and file path.

Stream to Tagged Stream

- This facilitates the conversion of a continuous stream to a tagged stream, allowing for efficient data organization.
- This is valuable for applications requiring precise sample identification, synchronization, and event marking in signal processing workflows.

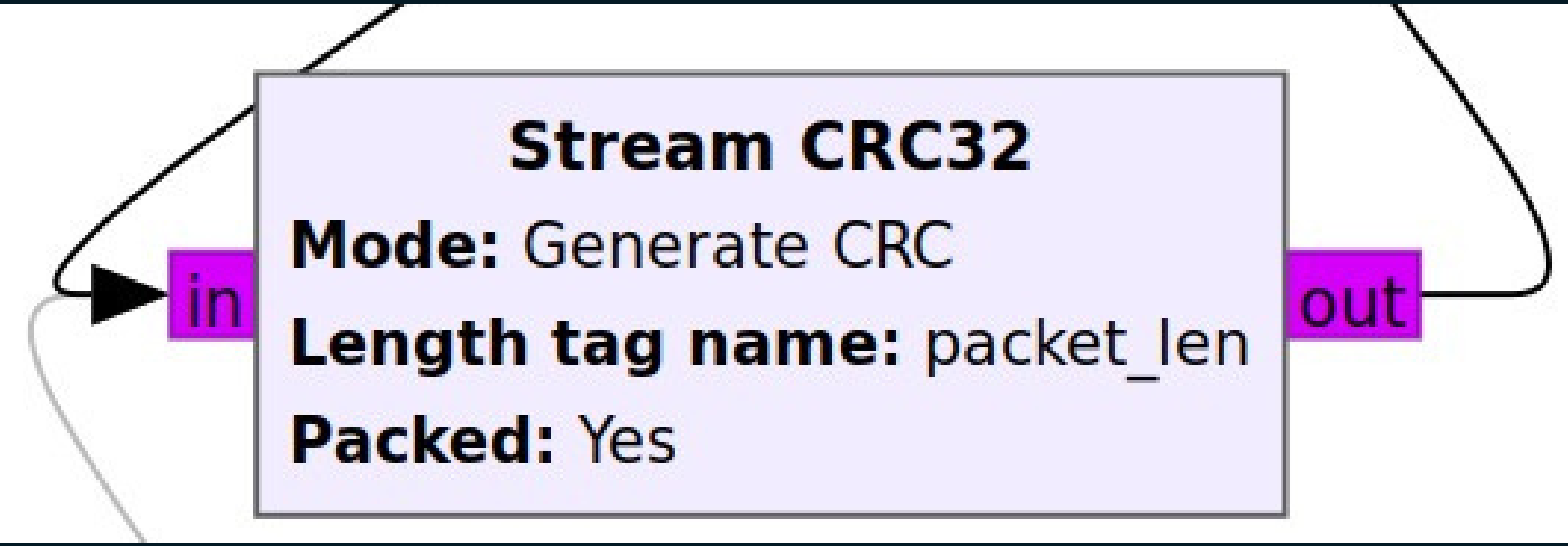


Stream CRC32

- This is used to calculate the Cyclic Redundancy Check (CRC) value for a data stream.

- CRC is a type of error-checking code that detects changes to raw data.

- The block is useful for ensuring data integrity in communication systems, file transfers, or any application where detecting errors in transmitted or stored data is important.

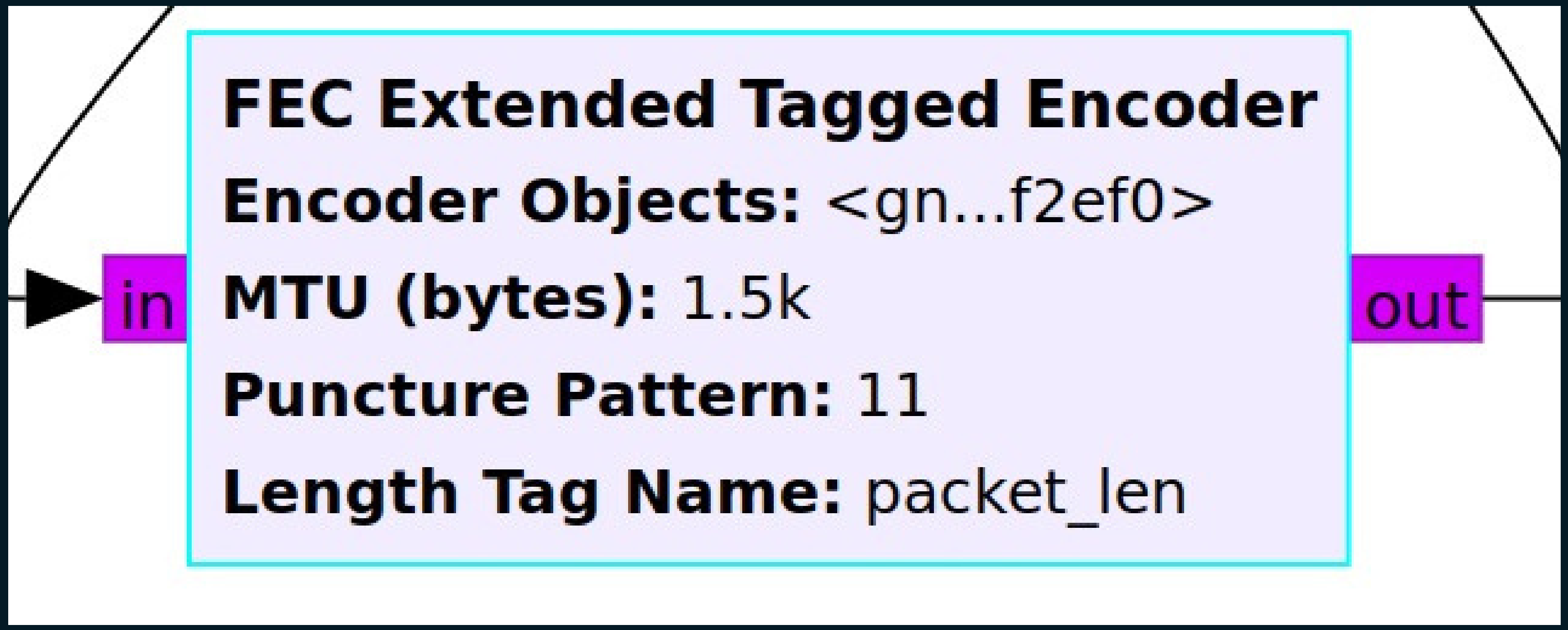


Channel Coding (FEC)

- Since a feedback can’t be obtained, FEC is used as the error correction method
- Transmitter encodes the data with an error correcting code and sends the coded message.
- Channel decoder in transmitter accepts message bits and adds redundancy according to prescribed rule
- Channel decoder in receiver exploits the redundancy to decide which message bits were transmitted.
- There are two FEC methods: Block coding and convolution coding.
- As convolution coding has better error controlling correction capabilities than block coding, we use convolution coding here.

FEC Extended Tagged Encoder

- Used to implement the forward error correction method.
- This block encodes an unpacked stream using a variety of Encoder Definition blocks.



Encryption

- **Encryption Method**: Symmetric key cryptography employed.
- **Algorithm Used**: Static key XOR encryption.
- **Data Security**: Ensures the confidentiality and integrity of the data file.
- **Secure Communication:** Facilitates secure communication through encryption and key exchange.

→

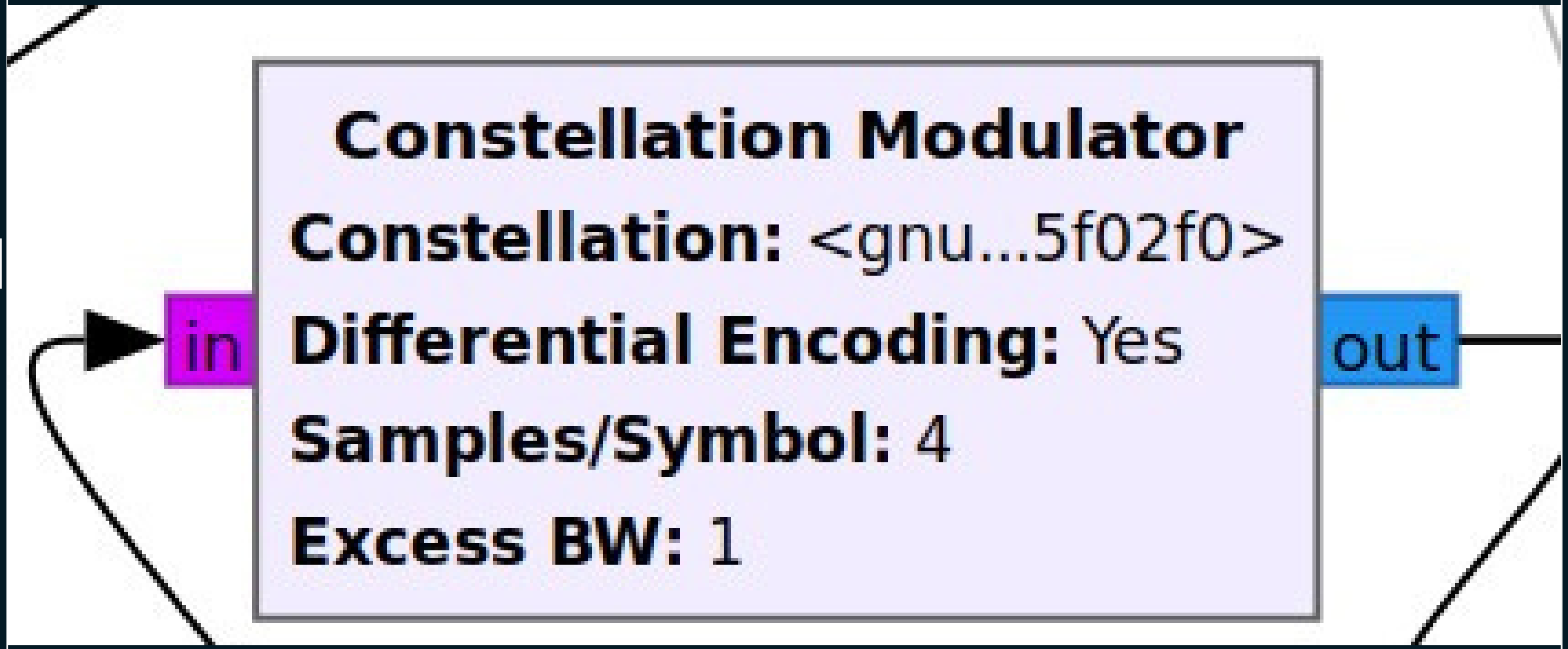
- We use a user-defined key here to encrypt.
- We can share the key using asymmetric key or transferring it through a safe channel.

BPSK-Binary Phase Shift Keying

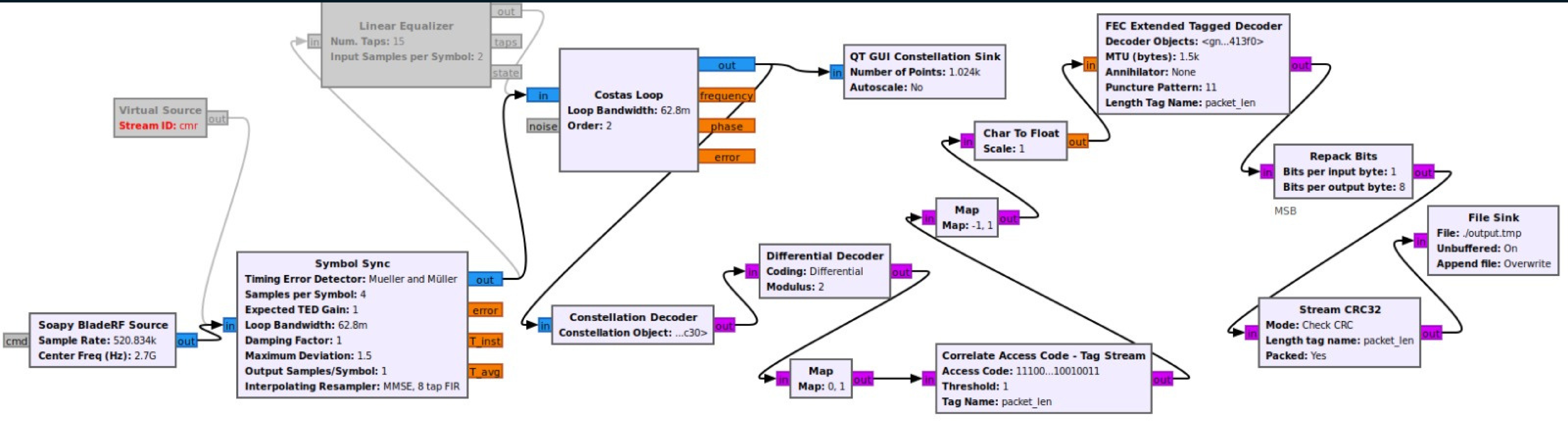
- Information transmission rate is lower comparing other modulations.
- But, BPSK has low error probability and can travel long distances.
- This is more power efficient.Therefore, BPSK was used as the modulation technique.

Constellation Modulator

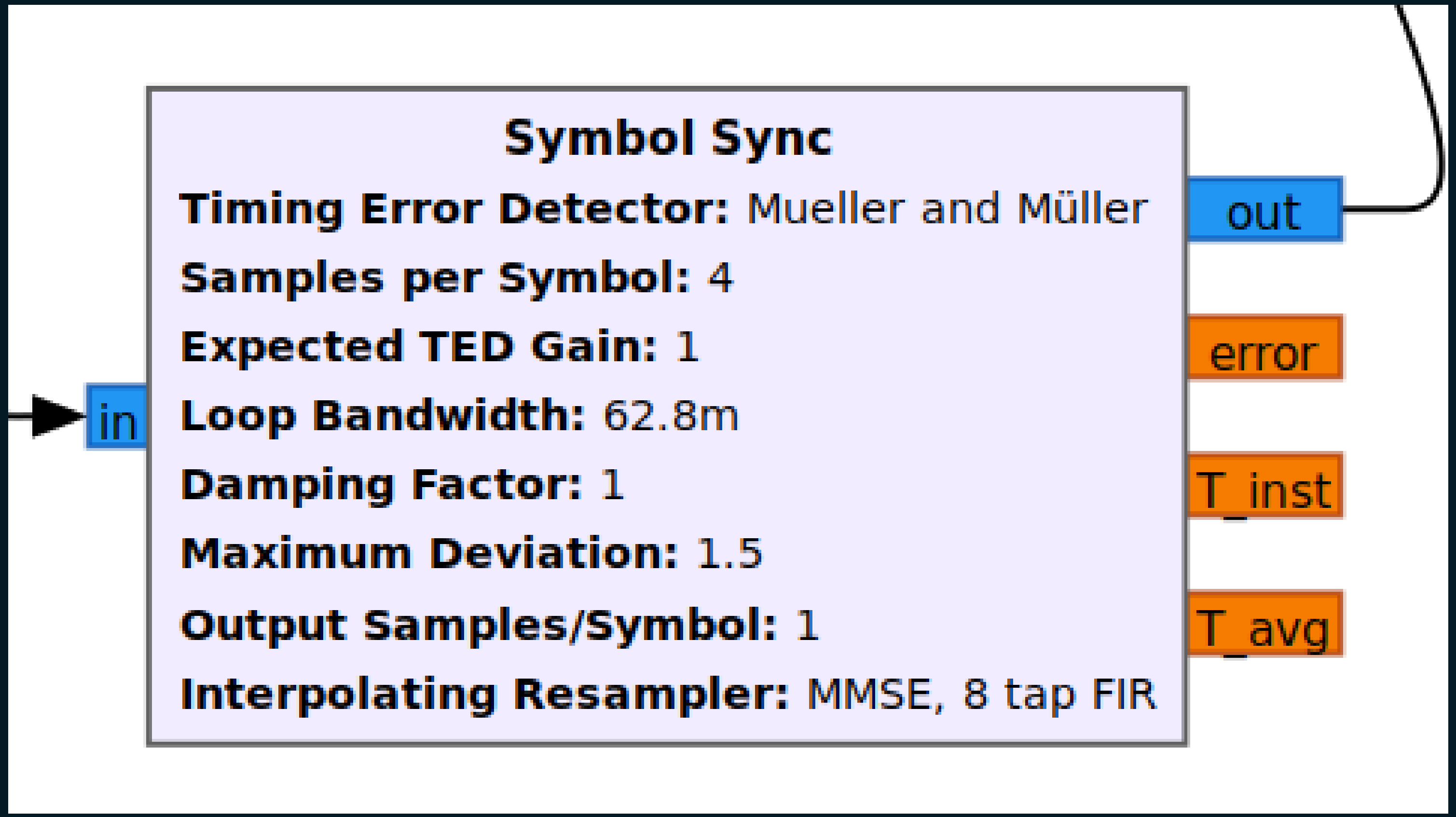
- This signal processing block does modulation by assigning digital symbols to points on a complex plane.
- It uses Differential Encoding, capturing the difference between neighboring symbols instead of encoding each one separately.
- This approach lessens the impact of phase shifts caused by noise or fading, making the receiver less sensitive.
- The result is more reliable data transmission.



Reception Flowgraph



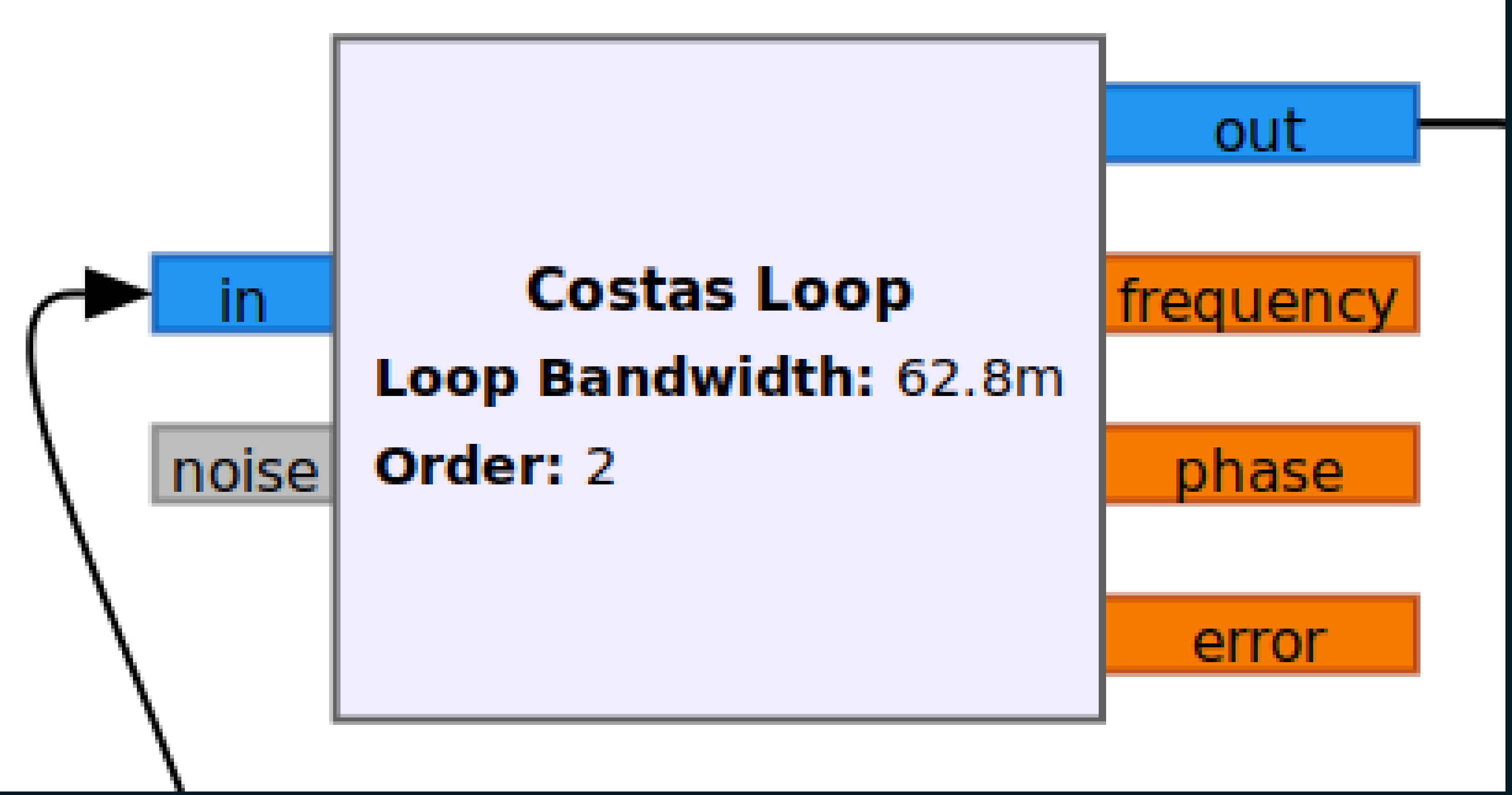
Symbol Sync



- Used for aligning symbols in digital communication.
- It ensures accurate timing, helps recover the symbol clock, and synchronizes the transmitter and receiver.
- By minimizing timing errors, it improves demodulation, contributing to reliable data recovery in digital communication systems.

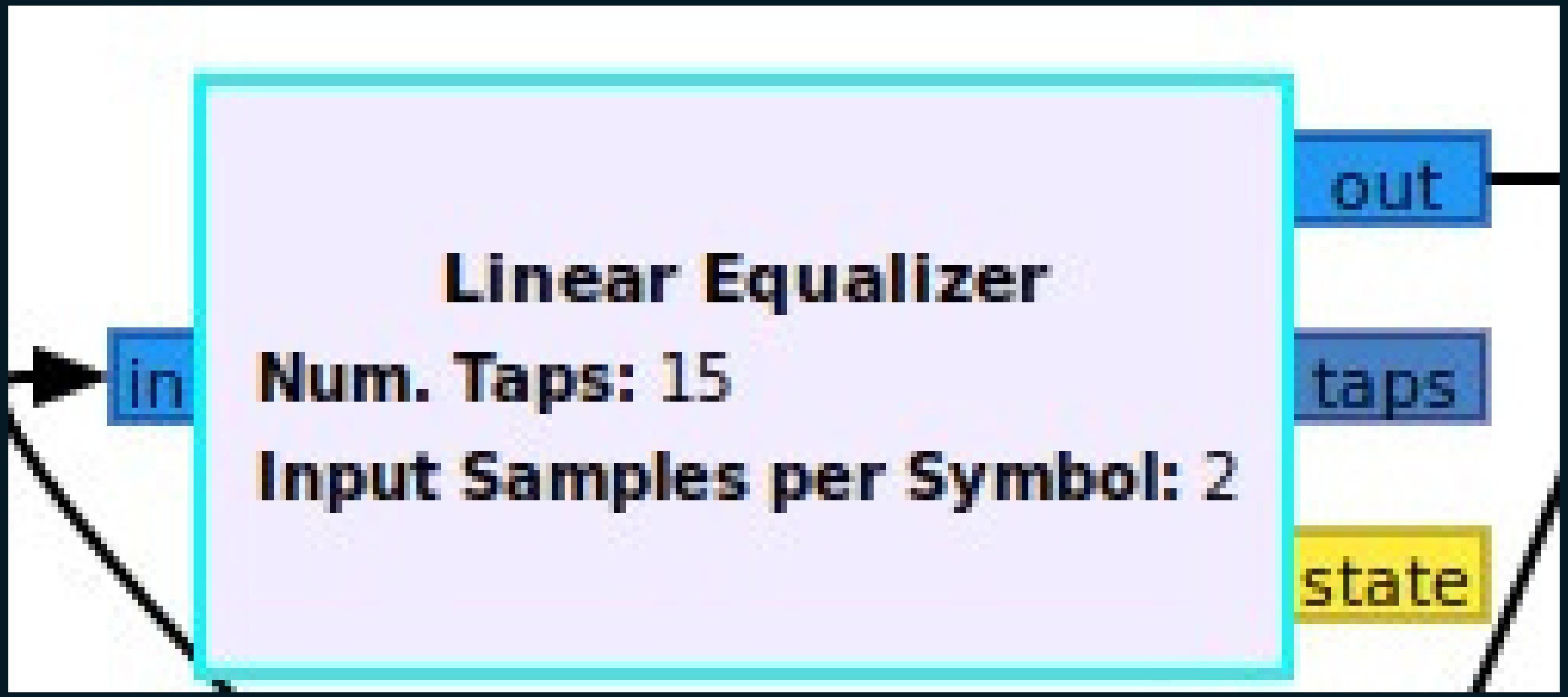
Costas Loop

- Enables carrier phase synchronization in a receiver.
- Utilizes a phase error detector to measure the phase difference between the received signal and the local oscillator.
- Adjusts the local oscillator's phase based on the detected phase error.
- Ensures the local oscillator tracks the phase of the received signal.
- Simple yet effective method for achieving synchronization in carrier phase.



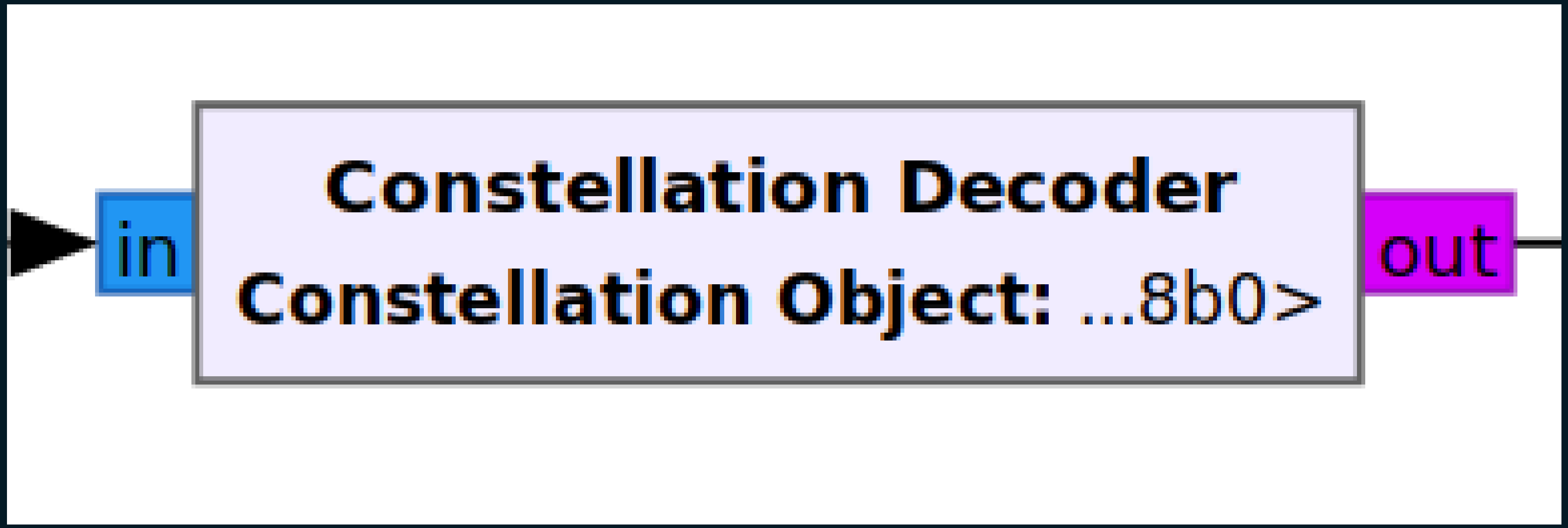
Linear Equalizer

- The Linear Equalizer block equalizes the incoming signal using an FIR filter
- Used in a receiver chain to compensate for the effects of multipath fading, which can cause time- and frequency selective distortion of the received signal.
- Increases the gain of the needed signal and decreases the gain of the distortions

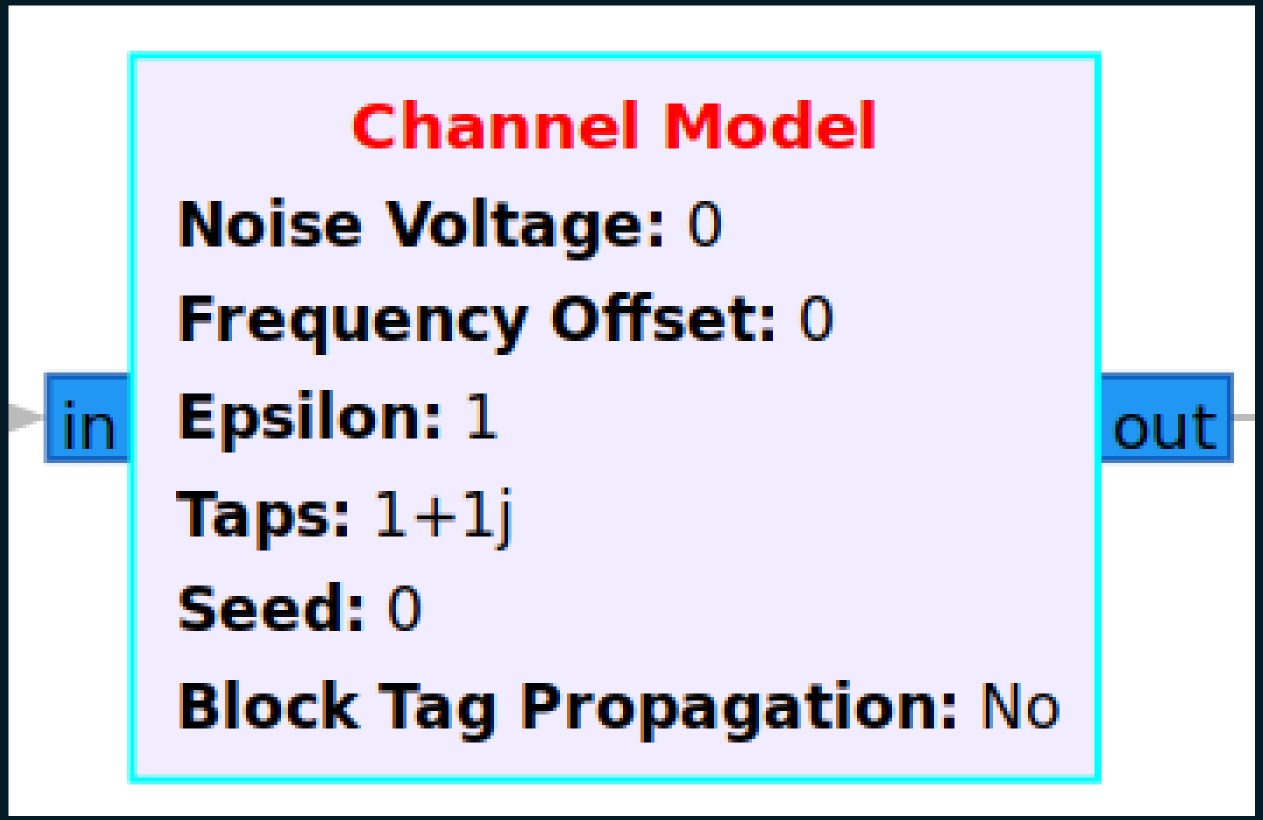


Constellation Decoder

- Recovers transmitted symbols in digital communication.
- Decodes symbols based on their positions in the signal constellation diagram.
- Converts modulated symbols back to their original digital format.
- Facilitates error detection and correction in received symbols.
- Enhances accuracy and reliability of data transmission.



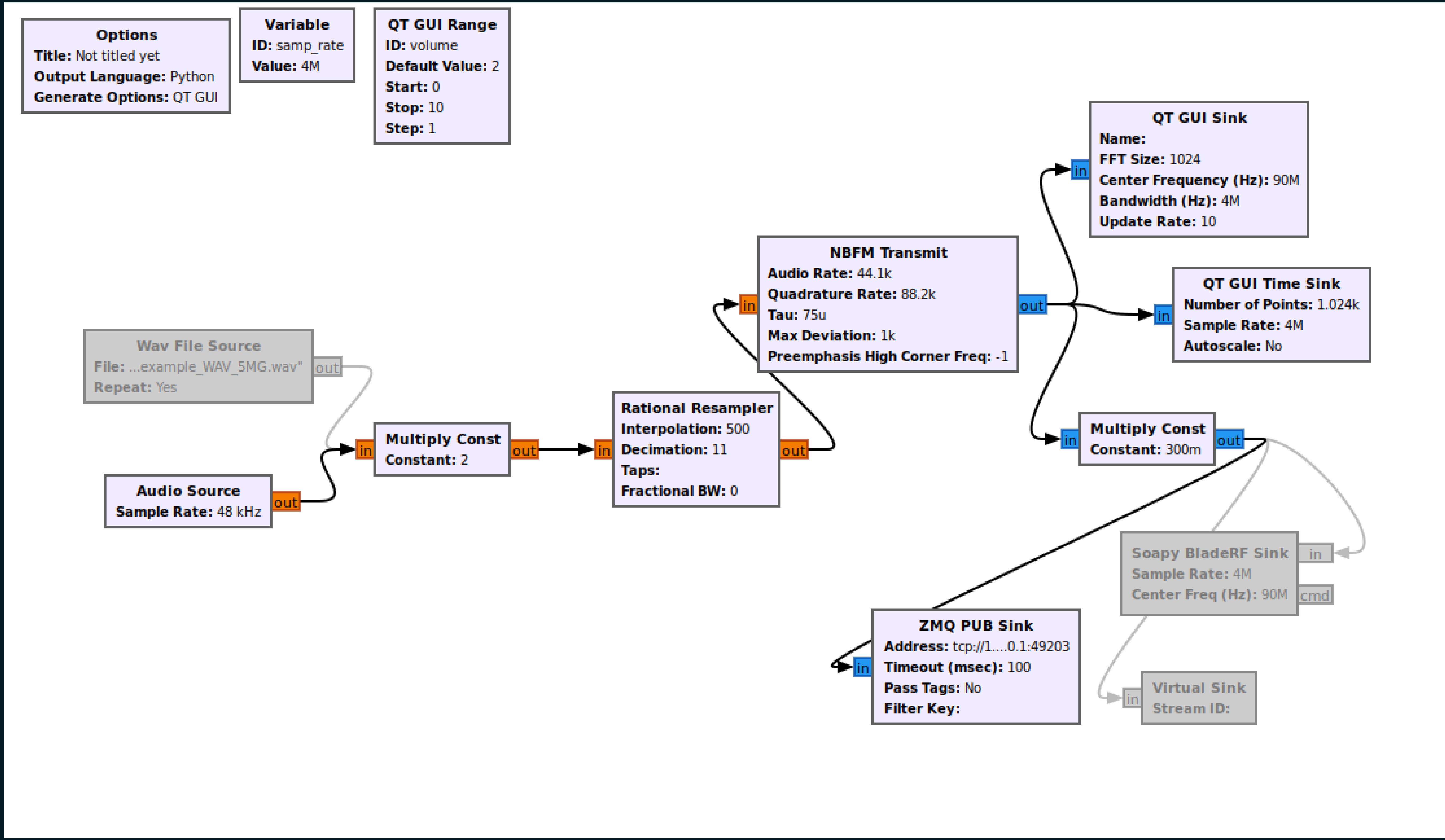
Channel Block



- Simulates signal transmission through diverse communication channels.
- Replicates real-world effects such as fading, noise, and distortion.
- Evaluates system performance under simulated channel conditions.
- Models signal propagation characteristics in different environments.
- Introduces errors to mimic challenges encountered in actual transmission.

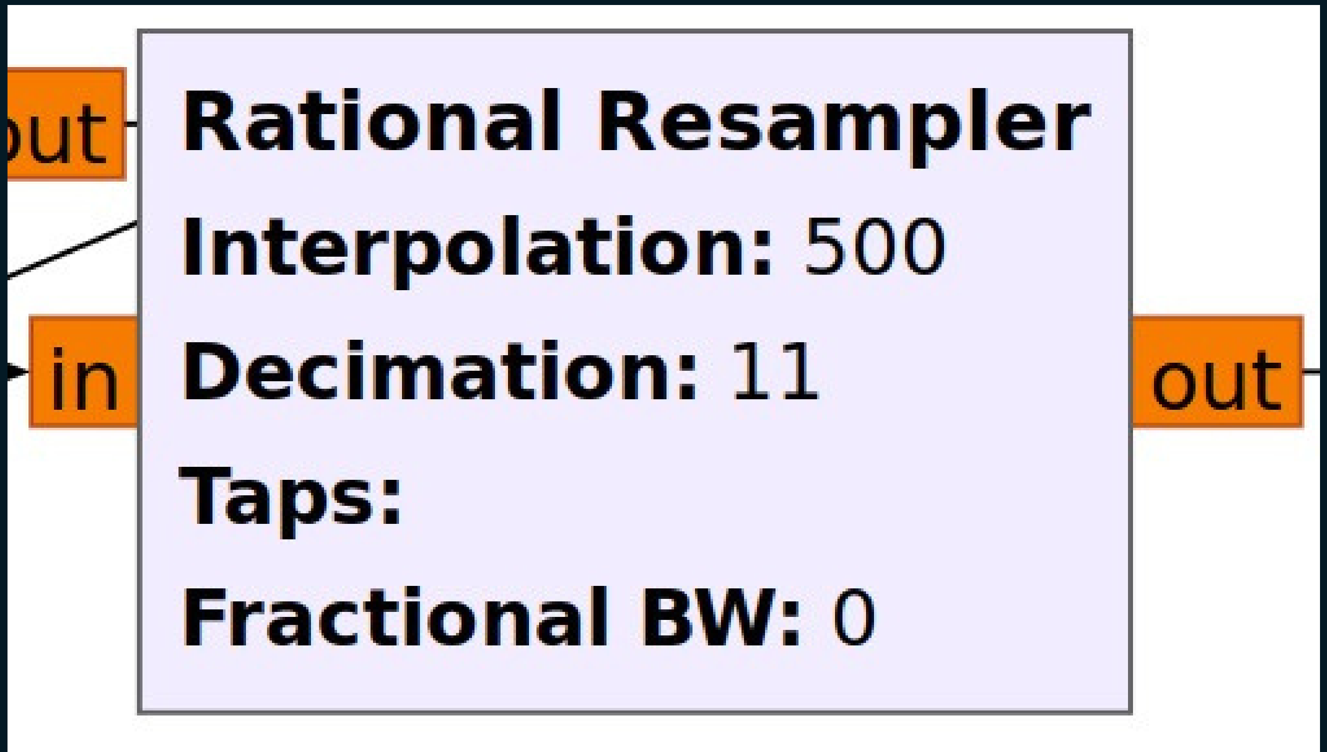
Real Time Audio Transmission

Transmitter Side



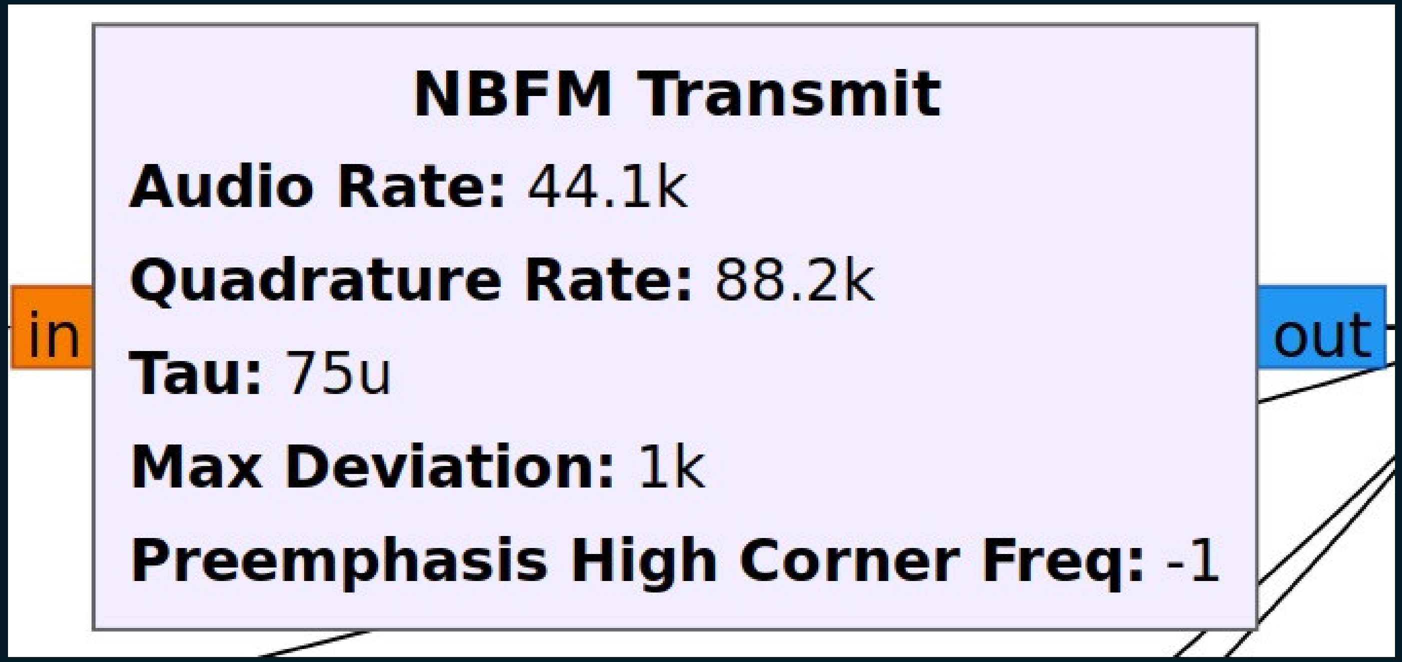
→

Rational Resampler



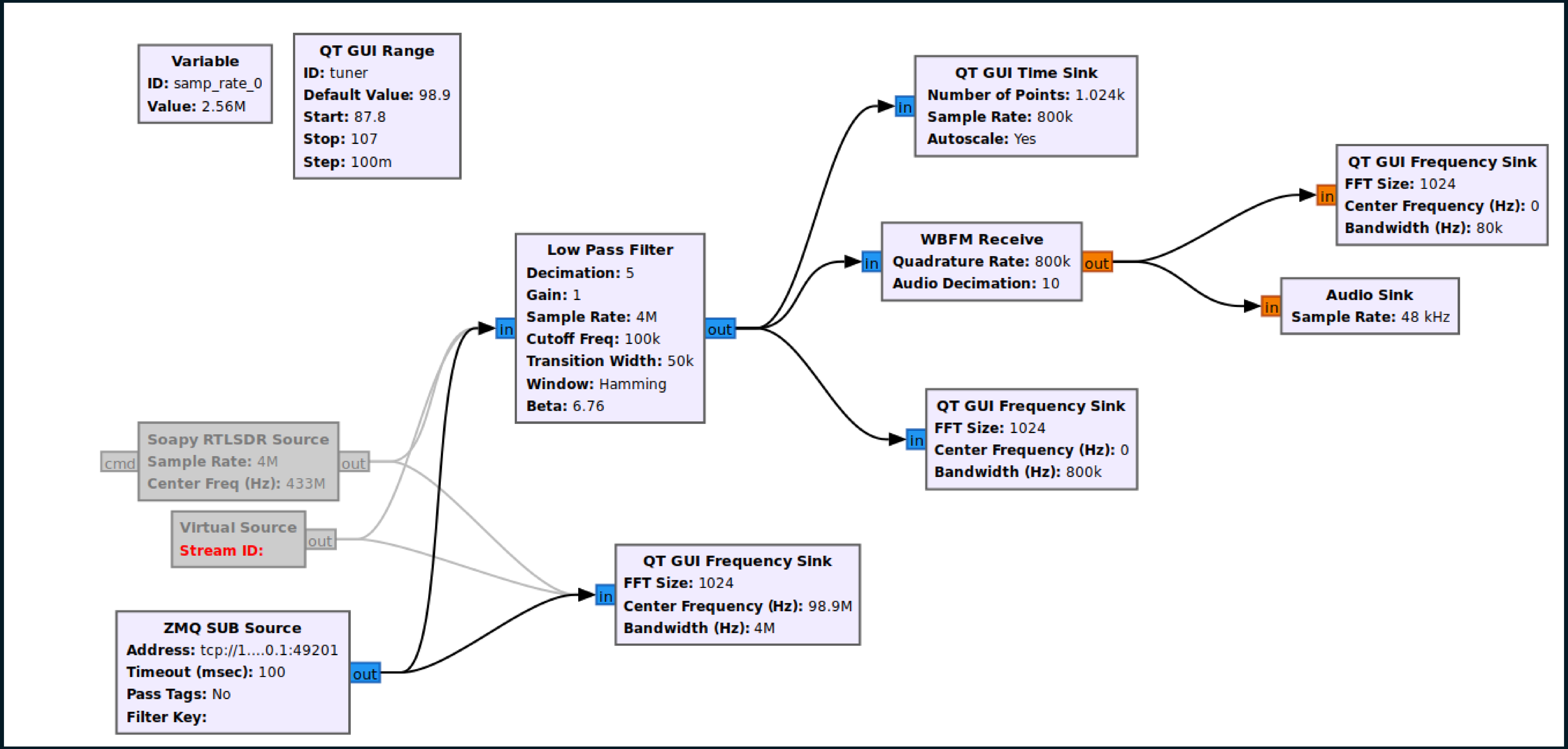
- Converts sampling frequency by a rational factor (L/M, where L and M are integers).
- Upsamples by inserting zero samples between existing samples.
- Downsamples by averaging or discarding samples.
- Removes unwanted high-frequency components from a signal.
- Estimates missing samples between existing samples.

NBFM Transmit



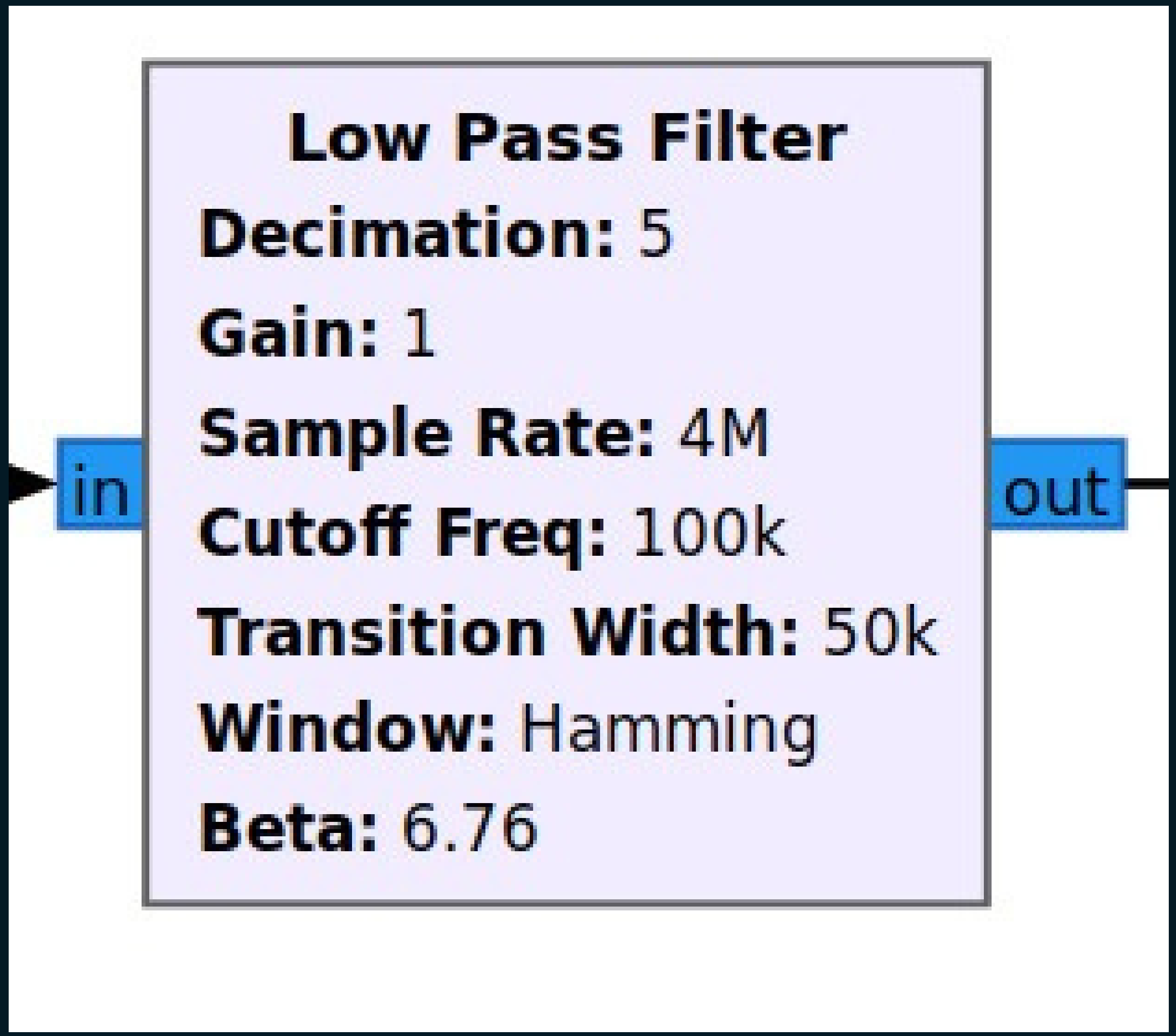
- Efficiently uses bandwidth compared to AM, allowing multiple channels.
- Less susceptible to noise than AM, thanks to frequency modulation.
- Preserves audio quality, resulting in high fidelity.
- Simple to implement and readily available for diverse applications.

Receiver Side



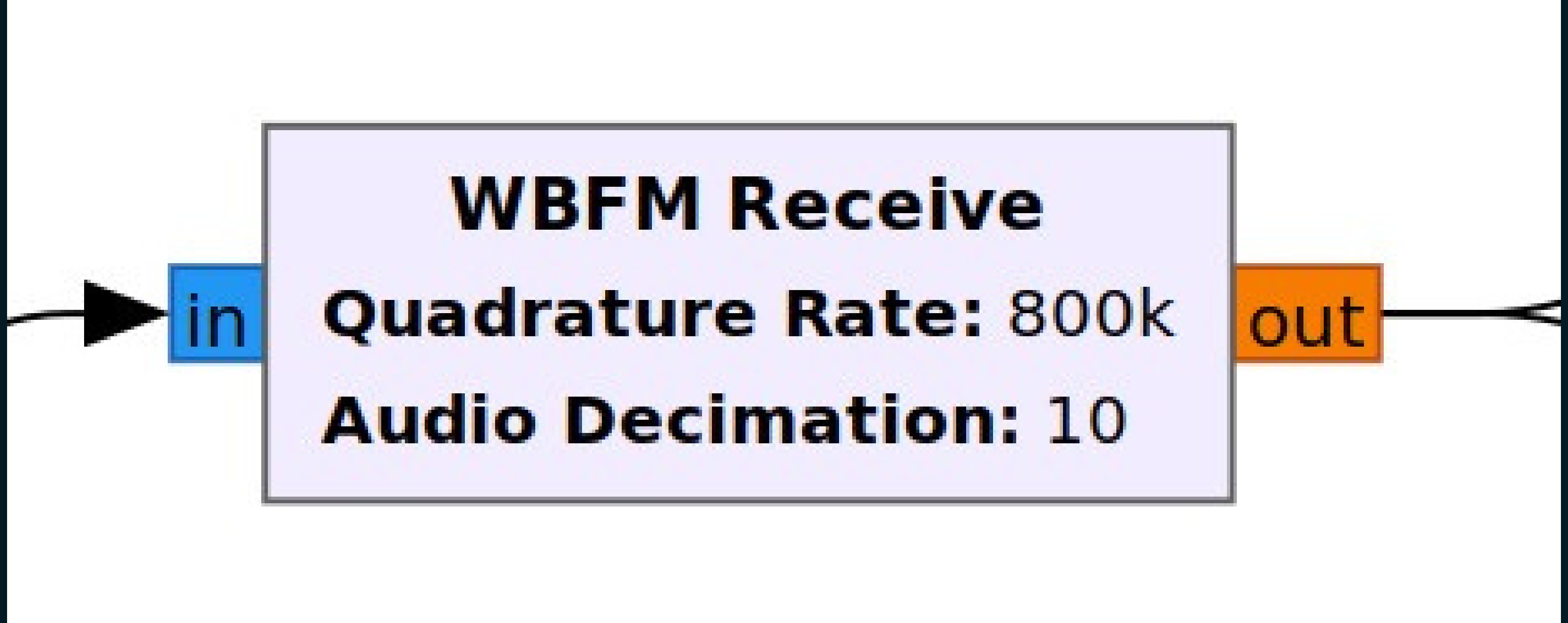
Low Pass Filter

- Removes noise and interference: Improves clarity and prevents aliasing.
- Limits bandwidth: Matches receiver capabilities and optimizes resource utilization.
- Crucial for real-time reception: Improves audio quality, allows reliable data transmission, and ensures accurate sensor measurements.



WBFM Receive

- Demodulates WBFM signals to extract audio information in real-time.
- Crucial for various applications like FM radio, wireless microphones, and audio recording.
- Offers high fidelity, noise immunity, wideband capabilities, and simple implementation.
- Enables reliable and clear real-time audio transmission and reception.




Thank you