

# Modulation and Coding – laboratory

## Digital Modulation

### Frequency Shift Keying (FSK)

The aim of the exercise is to develop algorithms for modulation and decoding for the two types of digital modulation: *Frequency Shift Keying (FSK)* and *Continuous Frequency Shift Keying (CFSK)*.

#### 1. Frequency Shift Keying (FSK)

The idea of FSK modulation is based on encoding information in the carrier signal frequency changes. In a sense, so it is a reference to an analog frequency modulation (FM), where the information was encoded using a small (compared to frequency the carrier signal) changes the frequency. In the case of frequency modulation, the frequency of the carrier signal changes according to the change in the amplitude of the modulating signal.

The main difference between FM and FSK modulation is hidden in the form of the modulating signal. In the FSK modulation the modulating signal is a binary (signal have only two values - zero and one), the modulation process is shown in Figure 1:

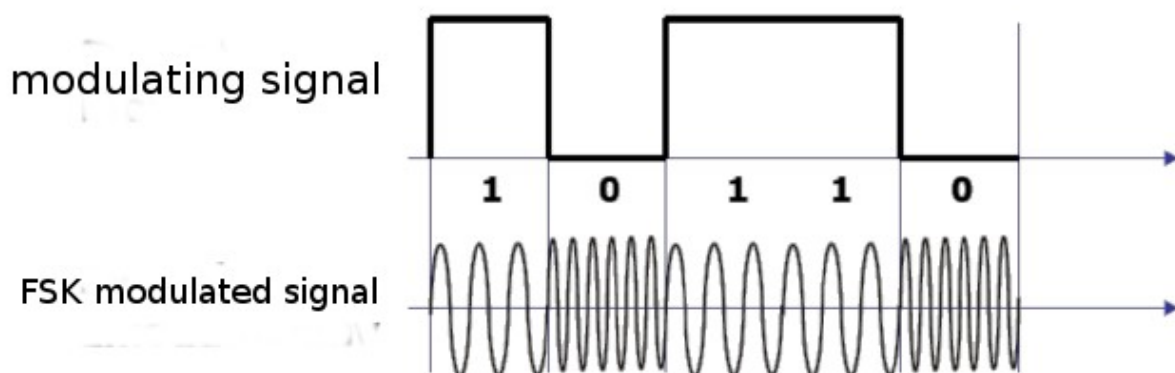


Figure 1: Example of FSK modulation

In the simple case, just enough when you define two frequency  $f_1$  for the symbol "0" and  $f_2$  for the

symbol "1" ( $f_2 > f_1$ ). For the symbol "0" signal is transmitted with carrier frequency equal to  $f_1$  and for symbol "1" signal is transmitted with carrier frequency equal to  $f_2$ .

## 1.1 FSK Modulator

- Build FSK modulator according to Figure 2:

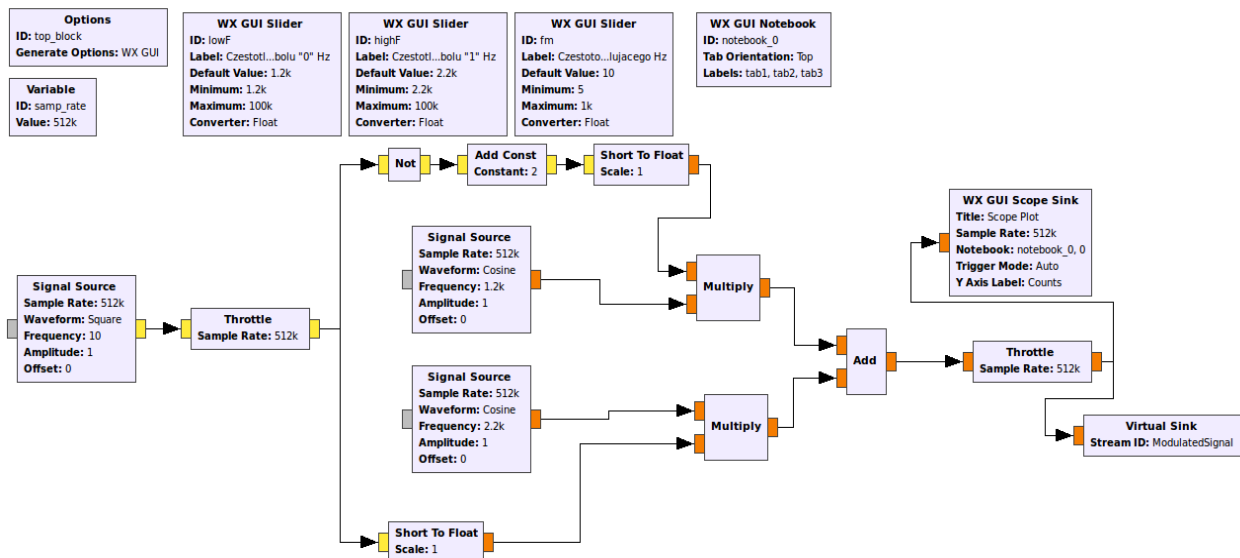


Figure 2: flow chart of FSK modulator (binary-FSK)

- Modulated signal: (type: float, wave: square,  $f = 50$  Hz)
- The frequency for the symbol "0", should be regulated in a range from 1 kHz to 100KHz, default: 1.2 kHz
- The frequency for the symbol "1", should be regulated in a range from 1 kHz to 100KHz, default: 2.2 kHz
- Run script, show results (should be similar to fig. 3).

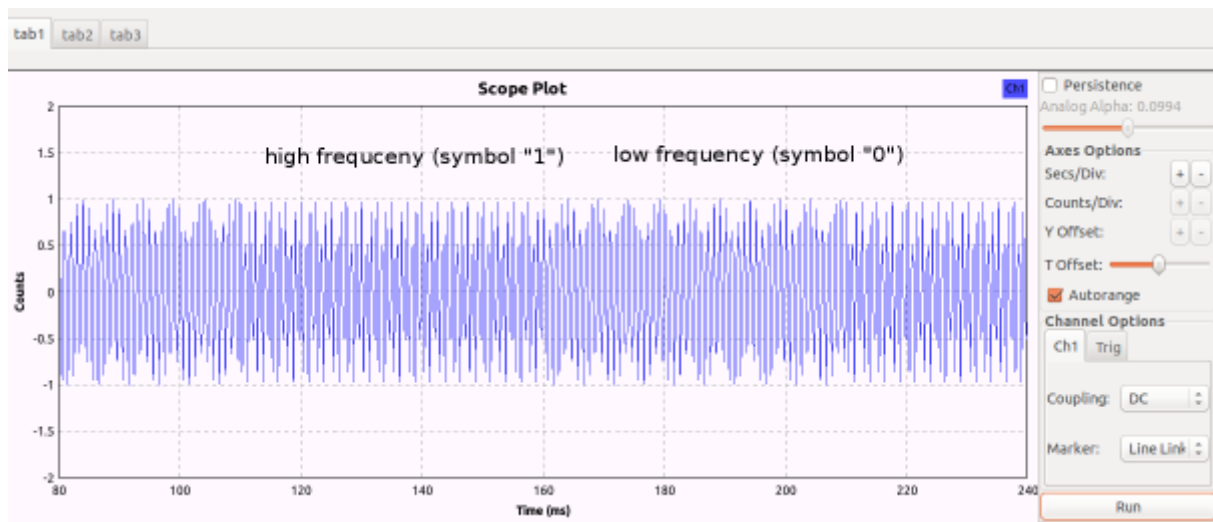


Fig 3: Example of signal modulated by FSK modulator

## 1.2 FSK Decoding

- Build FSK demodulator (for signal decoding) according to Figure 4:

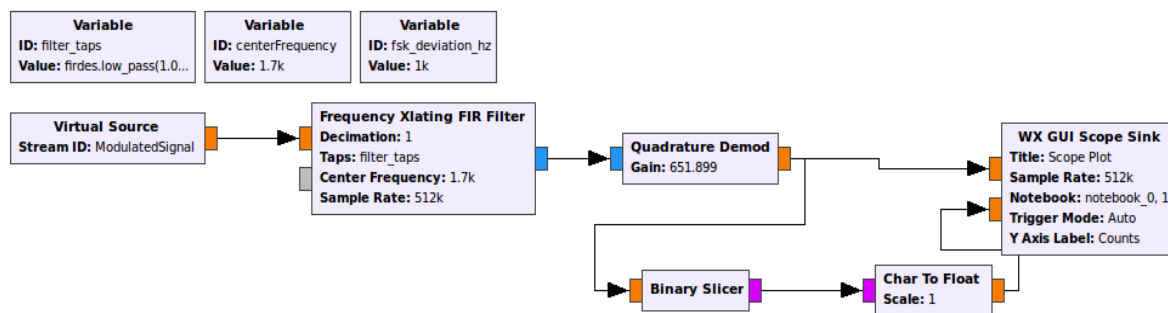


Fig 4: FSK decoding flow char.

- In order to filter out all other signals use the filter *frequency Xlating FIR Filter*. It requires an application filter parameters as a string parameter (variable *filter\_taps*). Create a variable called *filter\_taps*, whose value will be as follows:

$$\text{firde}(\text{low\_pass}(1.0, \text{samp\_rate}, 1000, 400))$$

where **1000** is the difference between the frequencies of the two symbols.

- The center frequency of filter "*X lating FIR filter*" will be automatically calculated according to the following:

$$CenterFrequency = (Frequency\_Symbol\_1 + Frequency\_Symbol\_0) / 2$$

where *Frequency\_Symbol\_0* and *Frequency\_Symbol\_1* are names of variable (e.g. *WX\_Slicer* ID) for both symbol frequencies. Variable *CenterFrequency* should be use to fill “Center Frequency” field on "Frequency *Xlating* FIR filter" component. The bandwidth of the filter should cover a range from *Frequency\_Symbol\_0* to *Frequency\_Symbol\_1*.

- *fsk\_deviation\_hz* is the frequency deviation FSK and is calculated as follows:

$$fsk\_deviation\_hz = Frequency\_Symbol\_1 - Frequency\_Symbol\_0$$

Create variable “*fsk\_deviation\_hz*”. This Variable should be used instead of variable ( “*fsk\_deviation\_hz*”) in equation in "Gain" field in “Quadrature Demod” component.

- Run script, show results (should be similar to fig. 4).

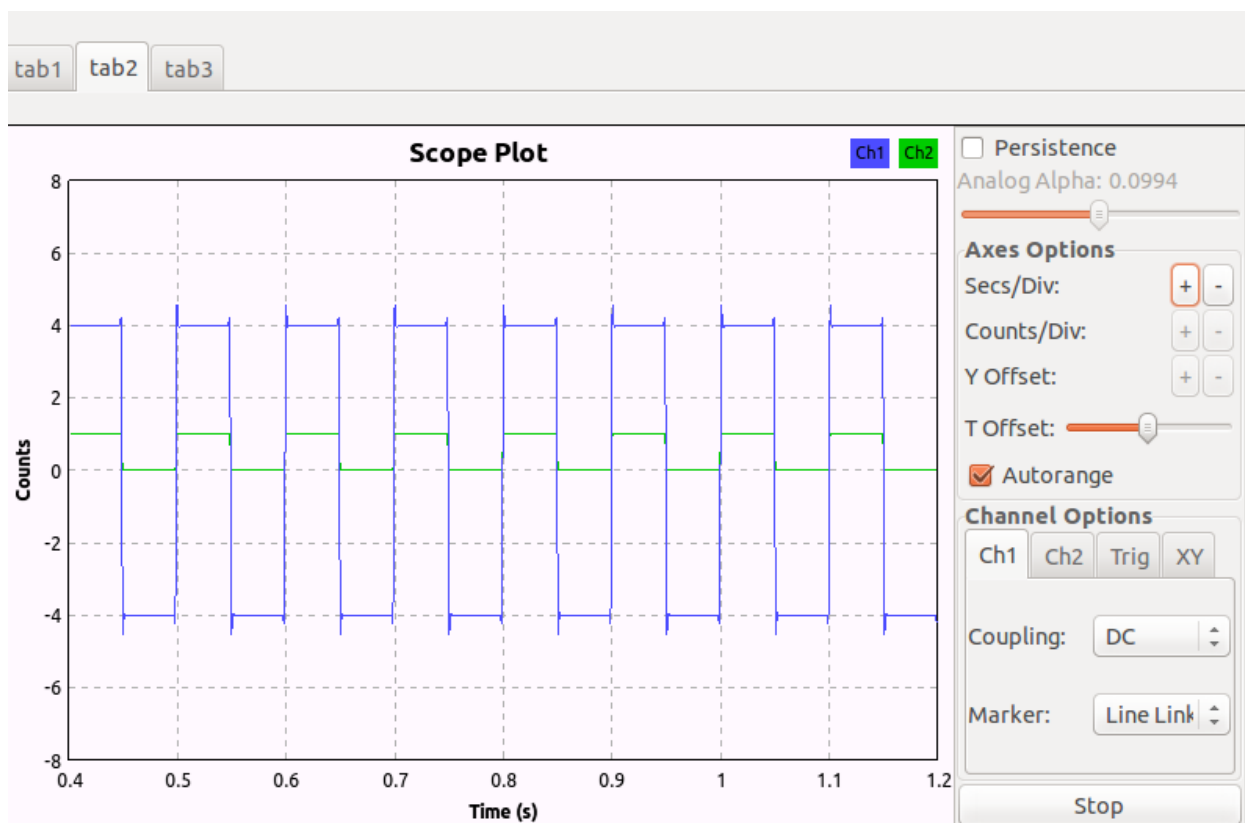


Fig 4: Recovered FSK signal

- Try with different frequencies for both symbols.

## 2. Continuous FSK Modulation

Classical FSK modulation does not ensure the continuity of the phase - when the switching frequency from first to second signal both signals can be in various phases. Lack of continuity in phase can cause a variety of problems during signal transmission.

However, there are versions of the FSK modulation, which ensures the continuity phase. The most important of these are:

- MSK (*Minimal Shift Keying*).
- CPFSK (*Continuous Phase FSK*).

Because the MSK modulation directly leads to the GMSK modulation, (and GMSK will be the subject of the next lab) so we use modulation CPFSK:

- Build CPFSK demodulator (for signal decoding) according to Figure 5:

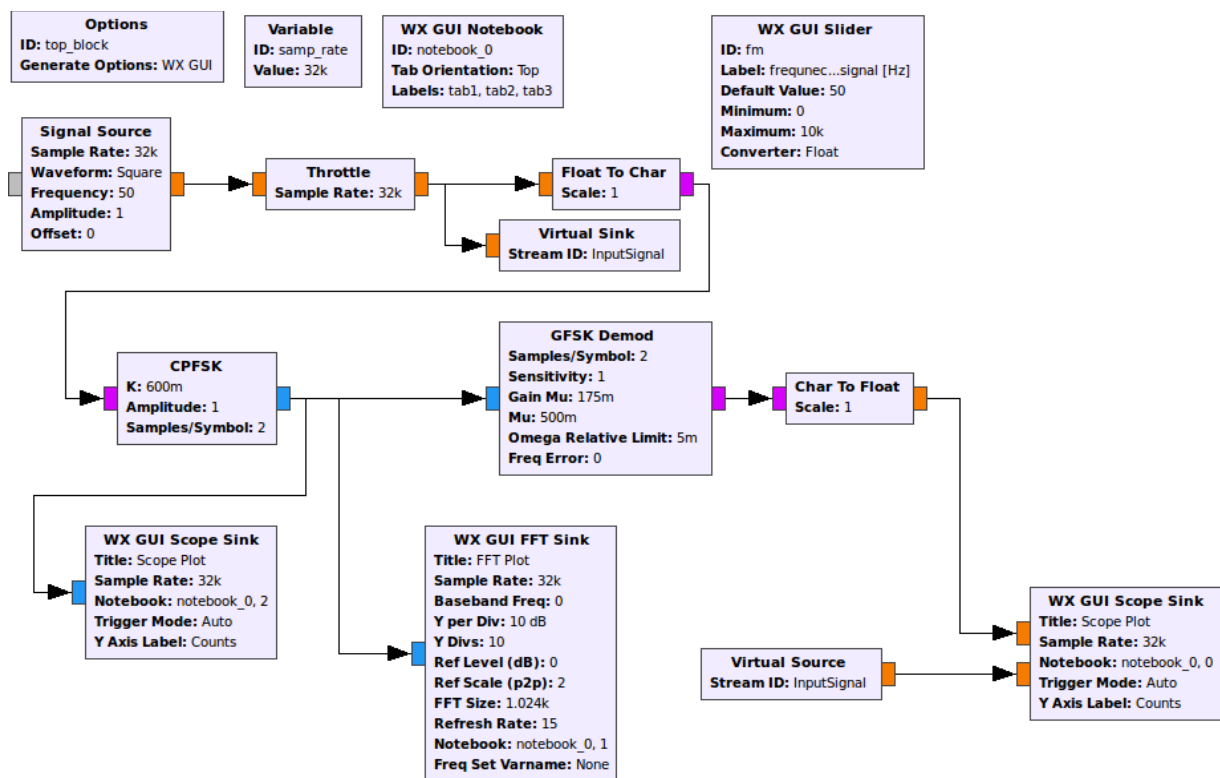


Fig. 5: flow chart of CPFSK modulator and demodulator.

- Run script. Show results.
- See how changing part of the frequency spectrum of the signal CPFSK while changing the frequency change of the modulating signal

### 3. Continuous FSK Modulation – real data transmission using SDR devices.

Exercises require two posts (two computers and two SDR devices), so you can work at team. First team will be make a transmitter (section 3.1) and second one will make a receiver (section 3.2).

#### 3.1 Transmitter

1. Prepare a file text (any text, not less than 10-15 kb).
2. Build the diagram as shown below:

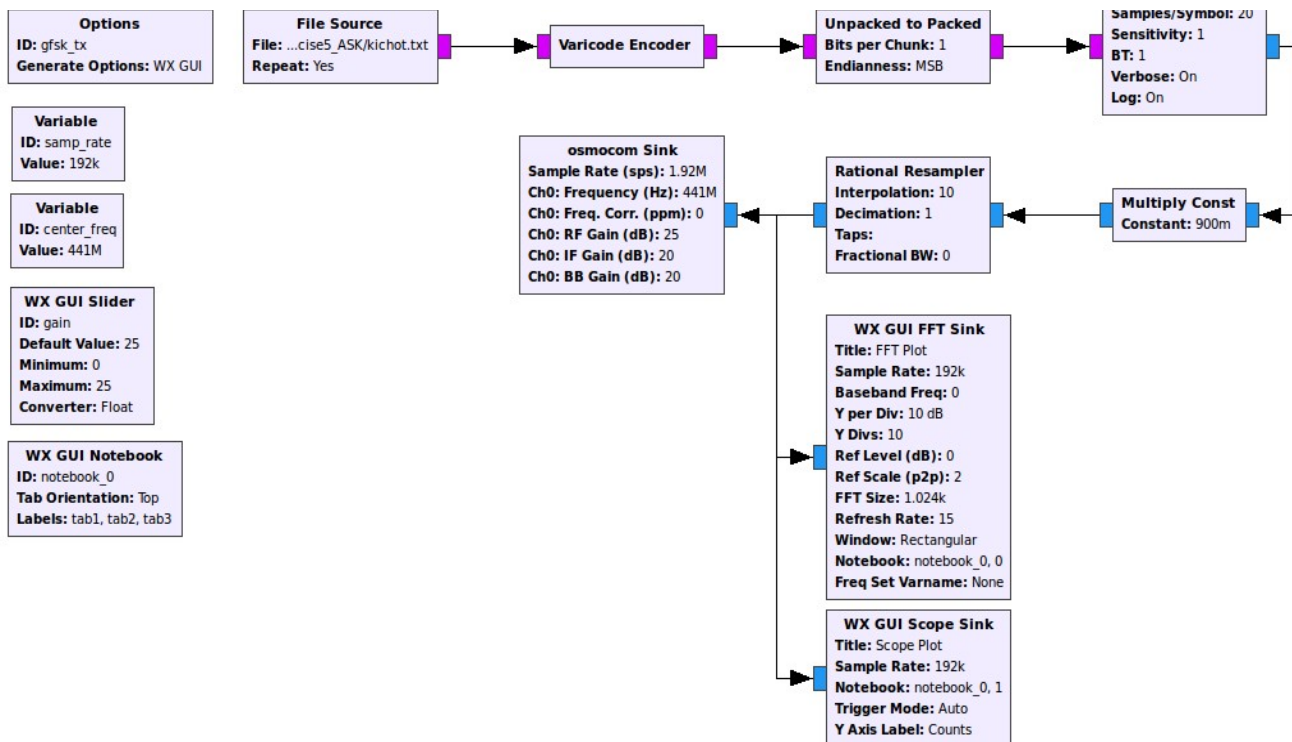


fig 6: Diagram for FSK trasmitter

- *Varicode encoder* - it's used for text compression, also covert byte stream to binary stream [0 and 1].
- *Unpacker to packet* - Convert a stream of unpacked bytes into a stream of packed bytes.
- *Rational Resampler* - changing a sample rate frequency.
- *GFSK Mod* – Continuous FSK modulator implemented on GnuRadio.

#### 3.2 Receiver

1. Build the diagram as shown below:

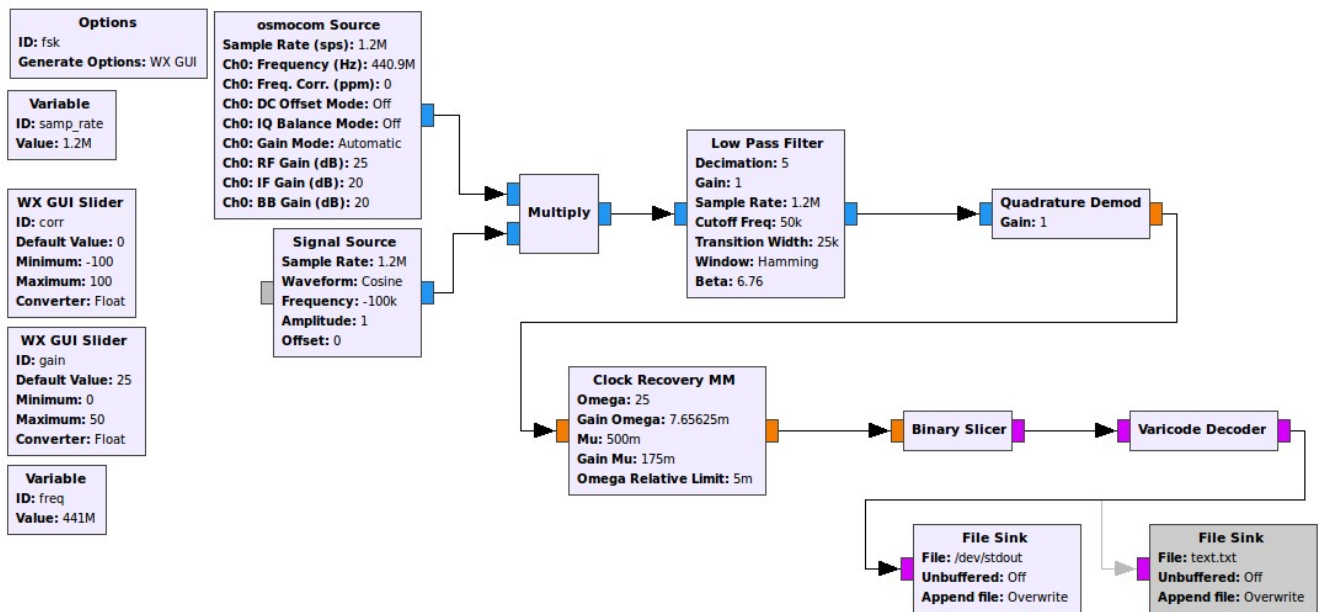


fig 7 Diagram for FSK receiver.

- *Quadrature demod* - quadrature demodulator. Could be used for demodulation / decoding signals modulated by each frequencies changing technique (FM, FSK, MSK etc).
- *Clock Recovery* - this block is need to recover a symbol form input signal (a must know what is symbol duration):
  - in field “*Omega*” type ( $samp\_rate/5/9600$ )
  - Leave the remaining fields unchanged
- *Binary Slice* – Change float signal to binary stream
- *Varicode decoder* – decoded and return text into byte stream (ASCII code)
- Set the output File (component with “stdout” is use for displaying recoverer text into console), type (instead a file path)

*/dev/stdout*

### 3.3 Check trasmission

- Can you receive a correct text?