

Санкт-Петербургский политехнический университет имени Петра великого  
Институт компьютерных наук и кибербезопасности  
Высшая школа компьютерных технологий и информационных систем

## Лабораторная работа №12

Дисциплина: Телекоммуникационные технологии

Тема: Моделирование простого передатчика и приемника с частотной  
манипуляцией (FSK)

Выполнил студент гр. 5130901\10202 \_\_\_\_\_ Ануфриева В.Д.

Принял преподаватель \_\_\_\_\_ Богач Н.В.

" \_\_\_\_ " \_\_\_\_\_ 2024 г.

Санкт-Петербург  
2024 г.

# Содержание

<b>1</b>	<b>Постановка задачи:</b>	<b>2</b>
<b>2</b>	<b>Ход работы:</b>	<b>2</b>
<b>3</b>	<b>Тестирование:</b>	<b>4</b>
<b>4</b>	<b>Передача файлов с использованием пакетов и AFSK:</b>	<b>5</b>

# 1 Постановка задачи:

Изучить пример моделирования простого передатчика и приемника с частотной манипуляцией (FSK). Для создания потоковой диаграммы использовать графический пользовательский интерфейс gnuradio-companion (GRC).

## 2 Ход работы:

Используя gnuradio-companion, построим заданную блок-схему: Опишем основные блоки,

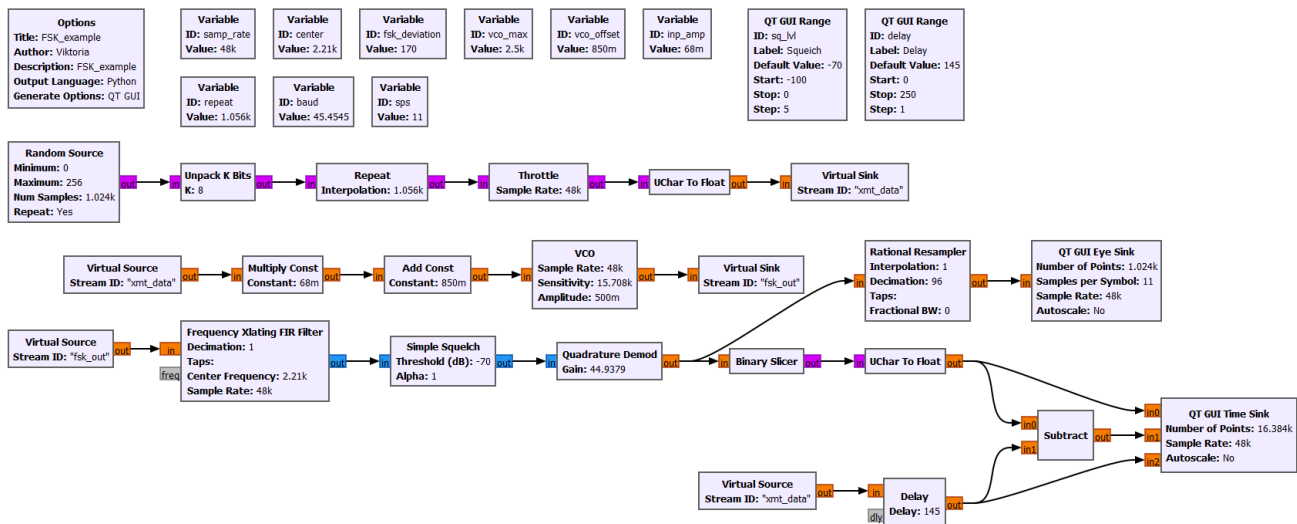


Рис. 1: Исследуемая блок-схема

использованные на данной схеме:

- **Options** — определяет имя файла для потоковой диаграммы, автора и т.д.
- **QT GUI Range** — используется для изменения значения задержки и уровня сигнала в процессе моделирования.
- **Random Source** — случайный источник, генерирующий байтовые значения от 0 до 255.
- **Unpack K Bits** — блок выбирает K младших битов из байта и расширяет их до K байтов со значениями 0 или 1.
- **Repeat** — повторяет каждый ввод времени интерполяции.
- **Throttle** — ограничитель скорости передачи данных.
- **UChar To Float** — преобразовывает поток беззнаковых символов в поток чисел с плавающей запятой.
- **Virtual Sink** — в сочетании с блоком Virtual Source это по сути то же самое, что протягивание провода между двумя блоками.
- **Virtual Source** — в сочетании с блоком Virtual Sink это по сути то же самое, что протягивание провода между двумя блоками + источники сигналов «xmt\_data» и «fsk\_out».

- **Multiply Const** — умножает входной поток на константу.
- **Add const** — блок добавляет постоянное значение к каждому элементу, который проходит через него.
- **VCO** — генерирует стандартные сигналы RTTY (radioteletype) частотой 2295 Гц (Отметка =1) и 2125 Гц (пробел =0).
- **Frequency Xlating FIR Filter** — выполняет преобразование частоты на сигнале и одновременно понижает дискретизацию сигнала через прореживающий КИХ-фильтр. Основное использование этого блока — эффективный канализатор, позволяющий выделить узкополосную часть широкополосного сигнала без необходимости центрирования этой узкополосной части по частоте.
- **Simple Squelch** — простой блок шумоподавления, основанный на средней мощности сигнала и пороге в дБ. Выход равен входу, если средний вход  $\geq$  порога, и нулю в противном случае.
- **Quadrature Demod** — принимает поток сложных выборок и создает поток чисел с плавающей запятой, которые представляют частотную демодуляцию.
- **Binary Slicer** — разделяет значение с плавающей запятой, создавая 1-битный выходной сигнал.
- **Rational Resampler** — многофазный КИХ-фильтр с рациональной повторной выборкой.
- **QT GUI Eye Sink** — графический приемник на основе QT, который принимает набор сложных потоков и используется для определения правильной центральной частоты.
- **QT GUI Time Sink** — графический приемник, основанный на библиотеке QT, предназначенный для отображения нескольких сигналов во временной области.
- **Delay** — создаётся заданная временная задержка.
- **Subtract** — производится вычитание по всем входным потокам.  $\text{выход} = \text{вход}_0 - \text{вход}_1 - \dots$
- Переменная **baud**, описывающая скорость передачи. Так как время передачи 22 миллисекунды, скорость передачи равна  $1/0.022$ .
- Переменная **repeat**, описывающая коэффициент повторяемости —  $(\text{int})(\text{samp\_rate} * 0.022)$ .
- Переменная **vco\_offset** = 0.85
- Переменная **inp\_amp** = 0.068

#### Принцип работы передатчика:

1. Случайный источник генерирует байтовые значения от 0 до 255.
2. При распаковке K битов каждый бит входных данных преобразуется в отдельный байт со значением в младшем значащем бите.
3. Поскольку аппаратное обеспечение не задействовано, для ограничения потока через систему используется блок Trottle.

Блоки виртуального приемника и виртуального источника используются вместо прямого подключения для создания более чистой потоковой диаграммы. При желании можно выполнять и прямые подключения потоков.

### Принцип работы приёмника:

1. Частотный КИХ-фильтр сдвигает принятый сигнал так, чтобы он был центрирован вокруг center частоты – на полпути между частотами mark и space.
2. Далее – шумоподавление, необходимое для реального приема сигналов RTTY.
3. Блок квадратурного демодулирования выдает сигнал, который является положительным для входных частот выше нуля и отрицательным для частот ниже нуля.
4. Когда полученный сигнал подается на двоичный «срезатель», на выходе получается байт, равный 1 или 0.
5. Мы получили двоичные распакованные данные.
6. Дополнительно добавлен приемник QT GUI Eye Sink с графическим интерфейсом QT. Он используется при настройке сигналов в реальном времени для определения правильной центральной частоты.

## 3 Тестирование:

Убедимся в том, что через схему проходит исходный поток битов. Для этого сравним его с входным потоком "xmt\_data".

Диаграмма на симуляции отображает оба сигнала. Сравнив их с небольшой задержкой, (См. Рис.2) видим, что принятый сигнал отстает на некоторое количество битов, потому что цепочка передатчика и приемника имеет множество блоков и фильтров, которые задерживают сигнал.

Для компенсации этого отставания необходимо задержать передаваемые биты на ту же величину — используем блок Delay. Отрегулировав задержку, получаем синхронизированные потоки (См. Рис.3). Это же мы видим при вычитании одного сигнала из другого – красная кривая. В итоге правильная задержка составляет около 145.

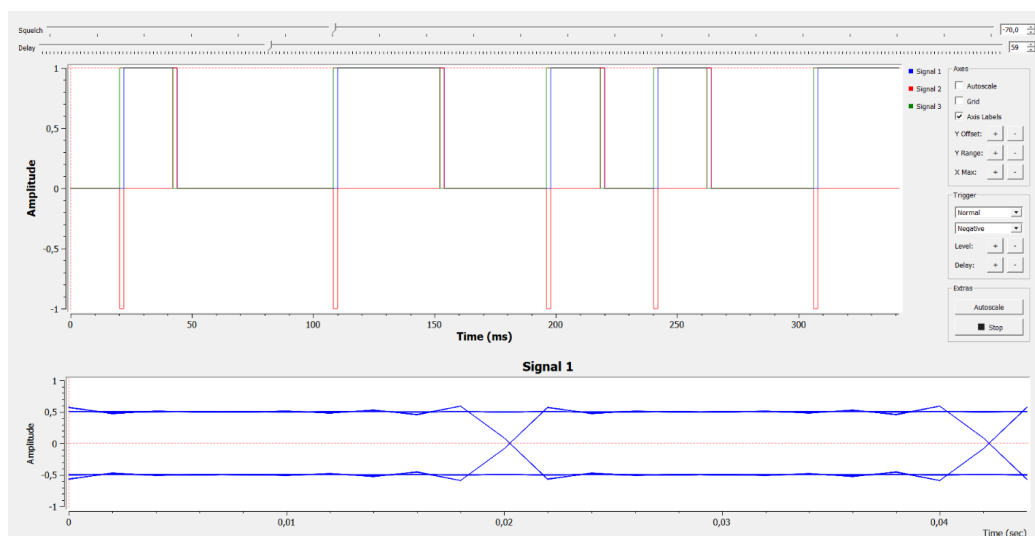


Рис. 2: Диаграмма с задержкой

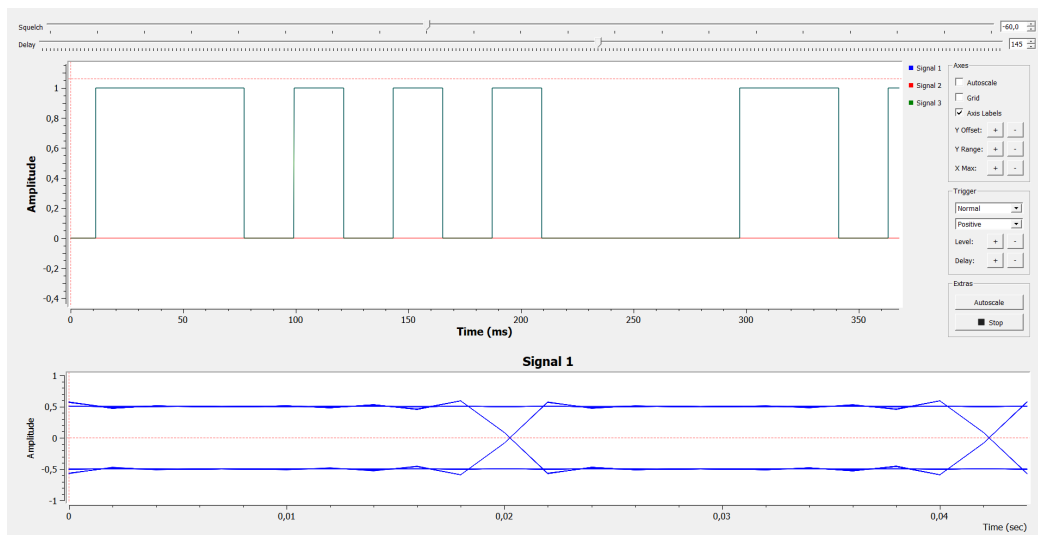


Рис. 3: Диаграмма со скомпенсированной задержкой

## 4 Передача файлов с использованием пакетов и AFSK:

Теперь рассмотрим пример использования FSK для отправки пакетов удаленному приёмнику. Как сказано в статье, FSK можно использовать для отправки любого содержимого данных.

Блок-схема приёмника представлена на Рис.4:

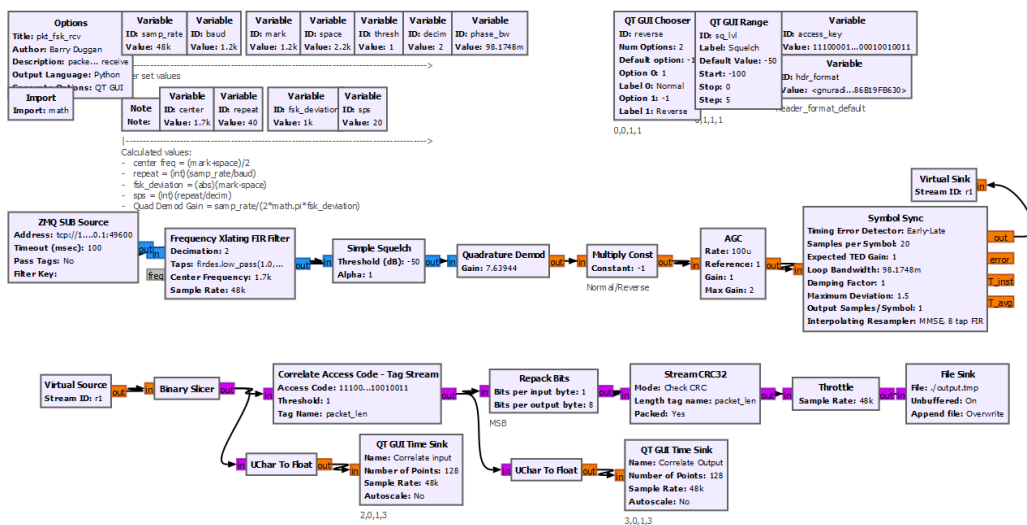


Рис. 4: Исследуемая блок-схема приёмника

Данная схема принимает сигнал AFSK, используя тот же частотный КИХ-фильтр и квадратурные демодулирующие блоки, что и в предыдущей схеме. Если частота Mark ниже, чем частота Space, значения 1 и 0 поменяются местами. Исправим это с использованием блока умножения константы с константой, установленной в QT GUI Chooser.

Для нормирования входного сигнала до 1, т.к. этого требует блок синхронизации символов, используем блок AGC. Как только поток битов подается в двоичный срезатель, на выходе получается распакованный байт, равный 1 или 0.

Чтобы определить начало пакета и добиться выравнивания байтов, блок корреляции потока кода доступа с тем же кодом доступа обнаруживает код доступа и передает полезную

нагрузку в `Stream_CRC32` для проверки допустимого CRC. Если CRC правильный, данные отправляются в `File Sink`.

Передача файла:

Блок 'ЕРВ: источник файла в помеченный поток' представляет собой встроенный блок Python, который заменяет блок File\_Source, блок Stream\_to\_Tagged\_Stream и части блока Burst\_Shaper. Блок Python выполняет следующие функции:

- Отправка заголовка, чтобы разрешить приемнику синхронизацию.
- Чтение файла в виде фрагментов "Pkt\_Len".
- Преобразование данных в Base64, который выдает 4 байта выходных данных на каждые 3 байта входных данных.
- Отправка каждого фрагмента Base64 с исправленными тегами "packet\_len".
- Отправка заполнителя после файла, чтобы убедиться, что все буферы были сброшены.

Преамбула выглядит так: "% 50 заглавных букв "U" ". Это повторяется четыре раза, чтобы приемник мог выполнить синхронизацию. Заполнитель post-файла отправляется 10 раз.

Блок-схема передатчика представлена на Рис.5:

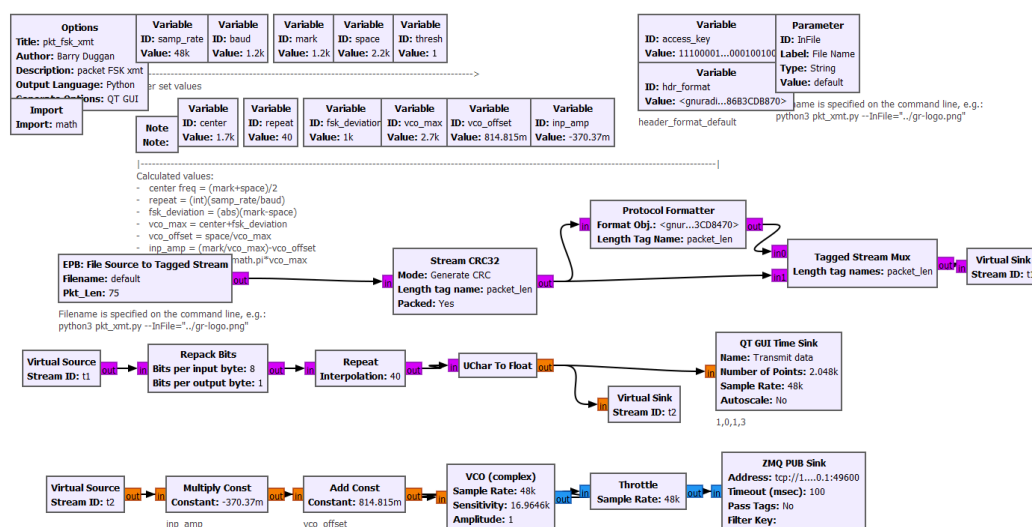


Рис. 5: Исследуемая блок-схема передатчика

Теперь запустим передачу пакетов с желаемым именем файла:

```
python3 pkt_fsk_xmt.py -InFile="../gr-logo.png"
```

1. В окне "pkt\_fsk\_xmt" началась передача файлов. Как сказано в статье, средняя пропускная способность составляет 150 байт в секунду. (См. Рис.6)

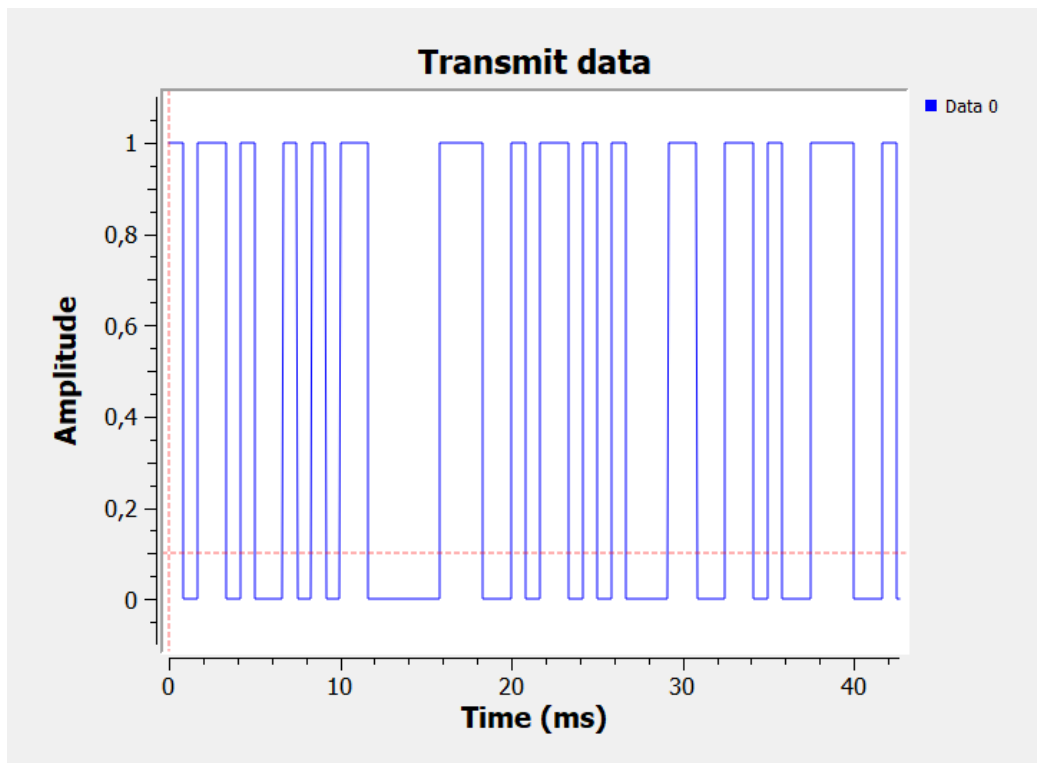


Рис. 6: Диаграмма процесса передачи данных

2. Уже через несколько секунд передача завершается, и в терминал выводятся сообщения об окончании операции: (См. Рис.7) "Конец файла"и "Окончание передачи":

```
(gnuradio_env) C:\Users\Пользователь\Desktop\gr-control\Transmitters> python pkt_fsk_xmt.py --InFile="../gr-logo.png"
pkt_fsk_xmt.py:317: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if StrictVersion("4.5.0") <= StrictVersion(Qt.qVersion()) < StrictVersion("5.0.0"):
pkt_fsk_xmt.py:73: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if StrictVersion(Qt.qVersion()) < StrictVersion("5.0.0"):
End of file
End of transmission
```

Рис. 7: Вывод в терминал

3. Когда на дисплее "pkt\_fsk\_xmt"прекратилась отправка пакетов-заполнителей, дисплей "pkt\_fsk\_rcv"стал статичным, как и ожидалось: (См. Рис.8)



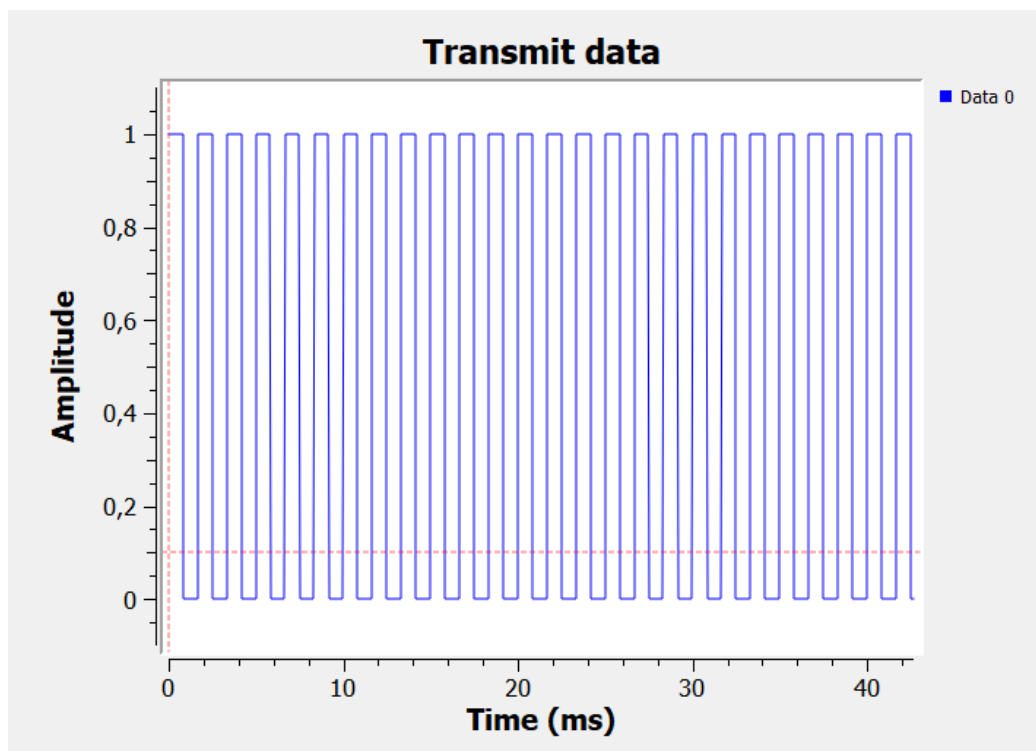


Рис. 8: Вывод в терминал

Работа завершена.

#### Удаление преамбулы и заполнителя:

Принятый файл сохранен в директорию `"/ gr-control /Receivers /output.tmp"`. Этот файл содержит до четырех пакетов преамбулы, за которыми следует переданный файл с кодировкой Base64, а далее следуют пакеты заполнения.

Для удаления этих пакетов преамбулы и заполнения и декодирования текста Base64 написана программа на Python `strip_preamble.py` :

```
python3 strip_preamble.py output.tmp output.png
```

После применения команды выше, все данные из `output.tmp` удалены.