

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных
технологий

Телекоммуникационные технологии

Отчёт по лабораторной работе 12

Работу

выполнил:

Д.А. Ефимов

Группа:

3530901/00201

Преподаватель:

Н. В. Богач

Санкт-Петербург
2023

Содержание

1. FSK	3
1.1. Теоритическая основа	3
1.2. Схема в GNU Radio	3
1.3. Тестирование	6
1.4. Вывод	7

1. FSK

1.1. Теоритическая основа

Frequency Shift Key - вид модуляции, при которой скачкообразно изменяется частота несущего сигнала в зависимости от значений символов информационной последовательности. Частотная модуляция весьма помехоустойчива, так как помехи искажают в основном амплитуду, а не частоту сигнала.

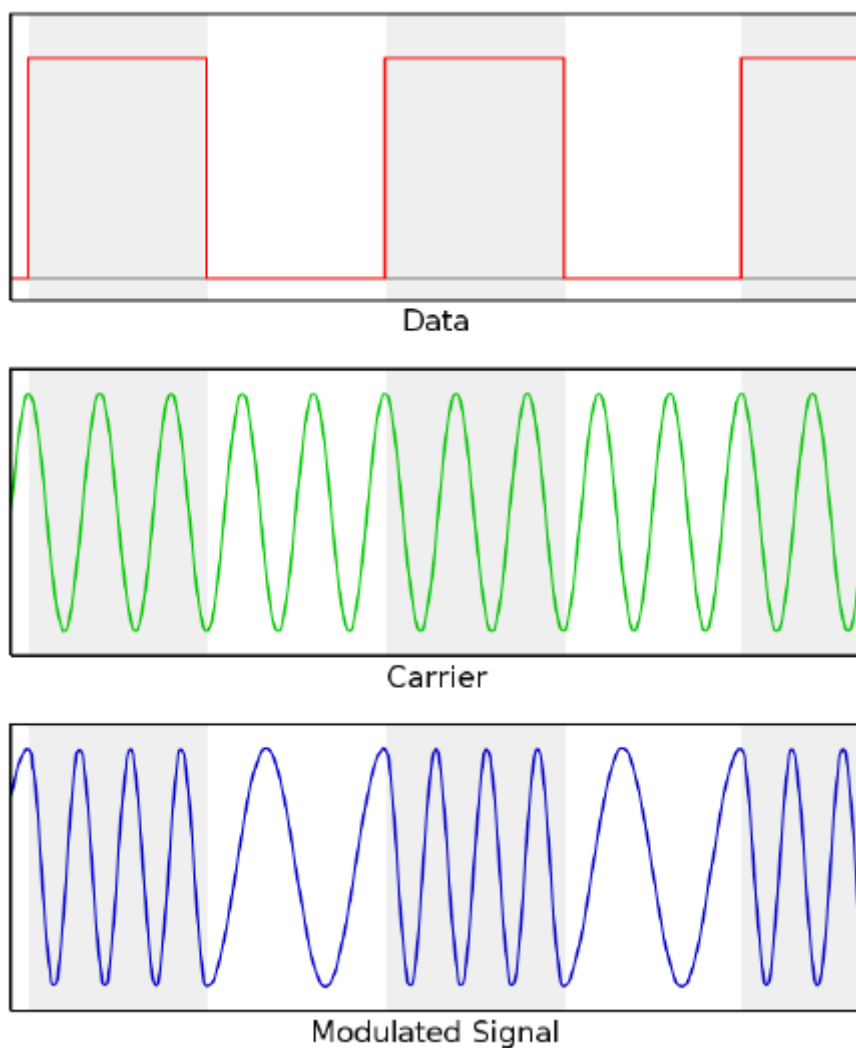


Рисунок 1.1. Пример FSK с двоичными данными

1.2. Схема в GNU Radio

Для модулирования и тестирования данного процесса, необходимо построить следующую схему в графическом интерфейсе GNU Radio:

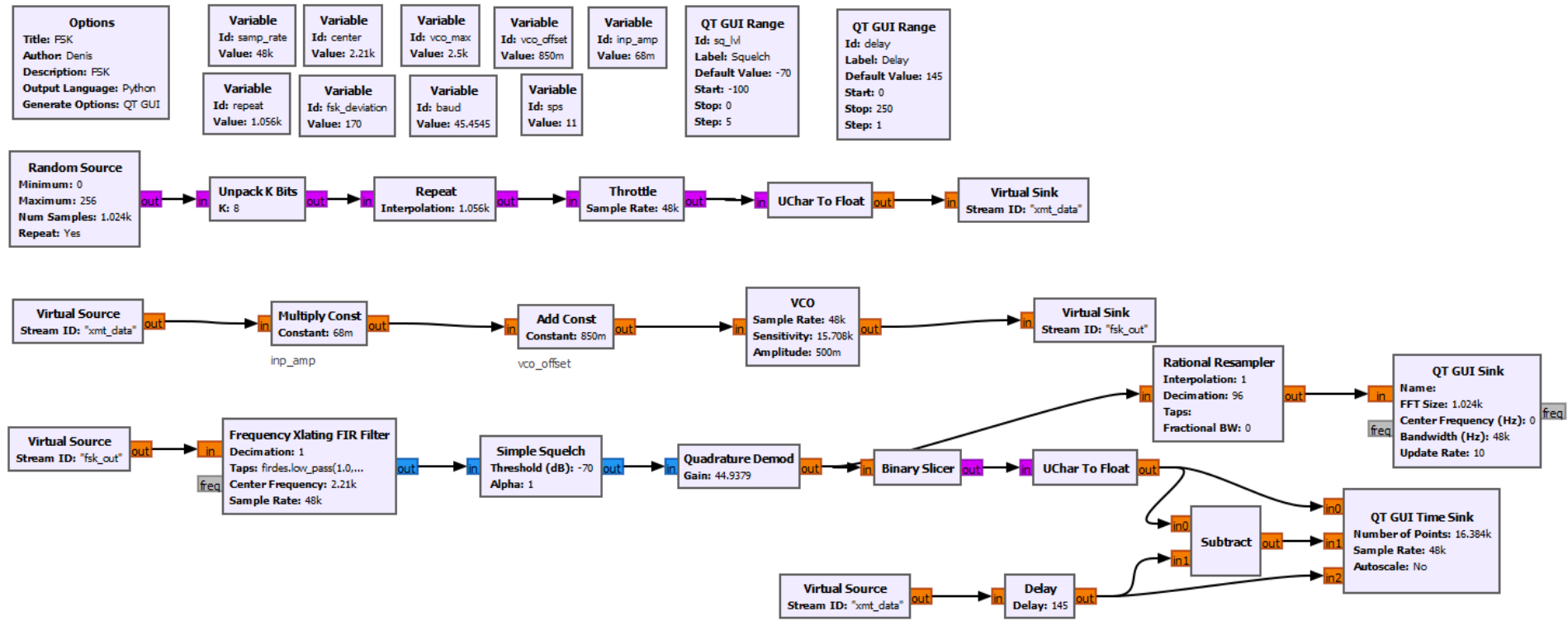


Рисунок 1.2. Схема FSK

В данной схеме используются следующие блоки:

- Frequency Xlating FIR Filter - Этот блок выполняет частотный перевод сигнала, а также понижает разрешение сигнала, запуская на нем децимирующий FIR-фильтр.
- Simple Squelch - Простой блок шумоподавления на основе средней мощности сигнала и порога в дБ.
- Quadrature Demod - Этот блок вычисляет произведение одновыборочного отложенного и сопряженного входного сигнала и нераскрытого сигнала, а затем вычисляет аргумент (также известный как угол, в радианах) результирующего комплексного числа.
- Binary Slicer - Нарезает числа с плавающей запятой, производя 1-битный вывод. Положительный ввод производит двоичную 1, а отрицательный ввод производит двоичный ноль.
- QT GUI Sink - Выводы необходимой информации в графическом интерфейсе.
- Options - Этот блок устанавливает некоторые общие параметры графа потоков. Такие как название проекта, авторство и другие.
- Variable - Этот блок сопоставляет значение с уникальной переменной. Есть возможность использования переменной в другом блоке, благодаря идентификатору (id) блока переменных.
- Multiply Const - Умножает входной поток на скалярную или векторную константу.
- Add Const - Прибавляет к входному потоку скалярную или векторную константу.
- QT GUI Range - Этот блок создает переменную с выбором виджетов. Переменной может быть присвоено значение по умолчанию, и ее значение может быть изменено во время выполнения в указанном диапазоне.
- Random Source - Генератор случайных чисел.
- Unpack K bits - Преобразует байт с k релевантных битов в k выходных байтов с 1 битом каждый.
- Repeat - Количество раз для повторения входных данных, выступающих в качестве коэффициента интерполяции.
- Throttle - Этот блок служит для того, чтобы дросселировать поток так, чтобы средняя скорость не превышала удельную скорость.
- Uchar To Float - Преобразует unsigned chars в поток float.
- Virtual Sink - Служит для сохранения потока в вектор.
- Virtual Source - Работает в паре с Virtual Sink блоком. Источник данных, который передаёт элементы на основе входного вектора.
- VCO - Генератор с регулируемым напряжением. Создает синусоиду частоты на основе амплитуды входного сигнала.

В этом примере используется Baudot Radioteletype, следовательно битовое время = 22 миллисекунды. Получаем скорость передачи 45,4545. Коэффициент повторения равен $\text{samp_rate} * 0,022$.

В VCO генерируются сигналы 2295 Гц (отметка = 1) и 2125 Гц (отметка = 0). При выборе полной шкалы частоты 2500 Гц (vco_max) для входа +1 чувствительность VCO = $(2 * \text{math.pi} * 2500 / 1) = 15708$. Можно использовать любую частоту выше 2295 Гц. 2500 Гц — хорошее круглое число. Глядя на вывод виртуального источника «xmt_data», Mark = +1.0 и Space = 0.0. Частота отметки 2295 Гц создается вектором $\text{inr_amp} = (1,0 * 0,068) + \text{vco_offset} = 0,918$, что равно $(2295/2500)$. Параметр отводов частотного Xlating FIR Filter равен 'firdes.low_pass(1.0,samp_rate,1000,400)'.
 Теперь посмотрим, что у нас с данными: Источник генерирует случайные байты (от 0 до 255). Далее этот байт распоковывается в каждый бит становится байтом со значащим младшим разрядом. Для ограничения потока использует Throttle. Приёмник при помощи фильтра смещает принимаемый сигнал так, чтобы он был сосредоточен вокруг центральной частоты - между частотами Mark и Space. Шумоподавителю добавлен для реального приёма сигналов. Блок Quadrature Demod производит сигнал, который является положительным для входных частот выше нуля и отрицательным для частот ниже нуля. Когда данные доходят до Binary Slicer, то на выходе получает биты, это и есть наша полученная информация.

1.3. Тестирование

Запустим моделирование.

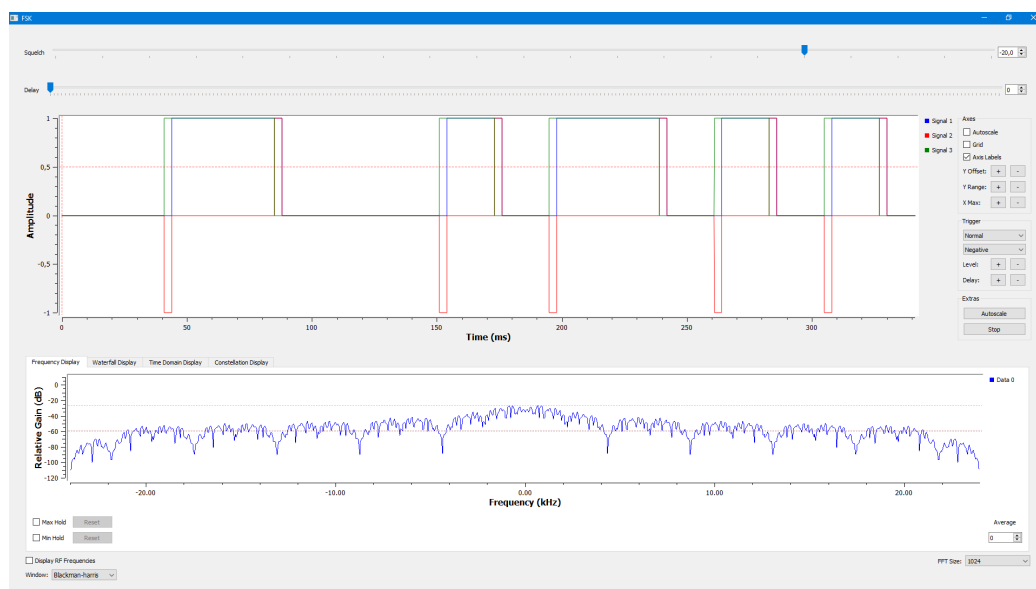


Рисунок 1.3. Тестирование без задержки с наложением шума

На графике можно видеть 3 сигнала разного цвета. Зеленный показывает данные которые были переданы передатчиком. Синий - это данные полученные приемником. А красный отвечает за разницу между зеленым и синим. Красный сигнал должен быть равен нулю, это сигнализирует о том, что данные передаются верно, однако на диаграмме сверху видно, что это не так и выходит так, что полученная информация различается от той, которую передавали. Принятый сигнал находится на некоторое количество бит позади, потому что цепочка передатчика и приемника имеет много блоков и фильтров, которые задерживают сигнал. Для того чтобы это исправить необходимо сделать

задержку между приемом и передачей данных, что и обеспечивает блок Delay. Правильном задержкой является 145.



Рисунок 1.4. Тестирование с установленной задержкой

Теперь как можно заметить все хорошо и данные передаются и получаются корректно.

1.4. Вывод

В данной лабораторной работе был рассмотрен один из способов модуляции. При помощи GNU Radio была создана и протестирована необходимая модель.