

САНКТ-ПЕТЕРБУРГСКИЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ПЕТРА ВЕЛИКОГО

**ИКНиК**

Лабораторная работа 12  
«Телекоммуникационные технологии»

**Выполнил:**


студент гр. 5130901/10101

Тучков Д.А.  
(подпись)

Преподаватель:  
Богач Н. В.  
(подпись)

Санкт-Петербург

GNU Radio - это бесплатный набор инструментов для разработки программного обеспечения с открытым исходным кодом, который предоставляет блоки обработки сигналов для реализации программных радиомодулей. Его можно использовать с легкодоступным недорогим внешним радиочастотным оборудованием для создания программно-определяемых радиостанций или без оборудования в среде, подобной моделированию. Он широко используется в исследованиях, промышленности, научных кругах, правительстве и среди любителей для поддержки как исследований в области беспроводной связи, так и реальных радиосистем.



[Log in](#) [Request account](#)

[Page](#) [Discussion](#) [Read](#) [View source](#) [View history](#)

## Low Pass Filter Example

This tutorial describes how to use a low-pass filter in GNU Radio.

The previous tutorial, [Python Block Tags](#), describes how to read and write tags in a Python block. The next tutorial, [Designing Filter Taps](#), describes how to design a set of low-pass filter taps and apply them against a signal.

**Contents** [hide]

- 1 Creating the Flowgraph
- 2 Run the Flowgraph
- 3 Noise Instead of Signal
- 4 Next Steps

**Beginner Tutorials**

Introducing GNU Radio

1. What is GNU Radio?
2. Installing GNU Radio
3. Your First Flowgraph

Flowgraph Fundamentals

1. Python Variables in GRC
2. Variables in Flowgraphs
3. Runtime Updating Variables
4. Signal Data Types
5. Converting Data Types
6. Packing Bits
7. Streams and Vectors
8. Hier Blocks and Parameters

Creating and Modifying Python Blocks

1. Creating Your First Block
2. Python Block With Vectors
3. Python Block Message Passing
4. Python Block Tags

### Creating the Flowgraph

Begin by adding the following blocks to the GRC work space:

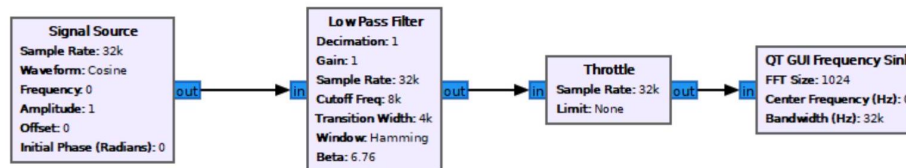
1. Signal Source
2. Low Pass Filter
3. Throttle
4. QT GUI Frequency Sink
5. QT GUI Range

Connect the blocks in the following manner:

## Фильтр нижних частот

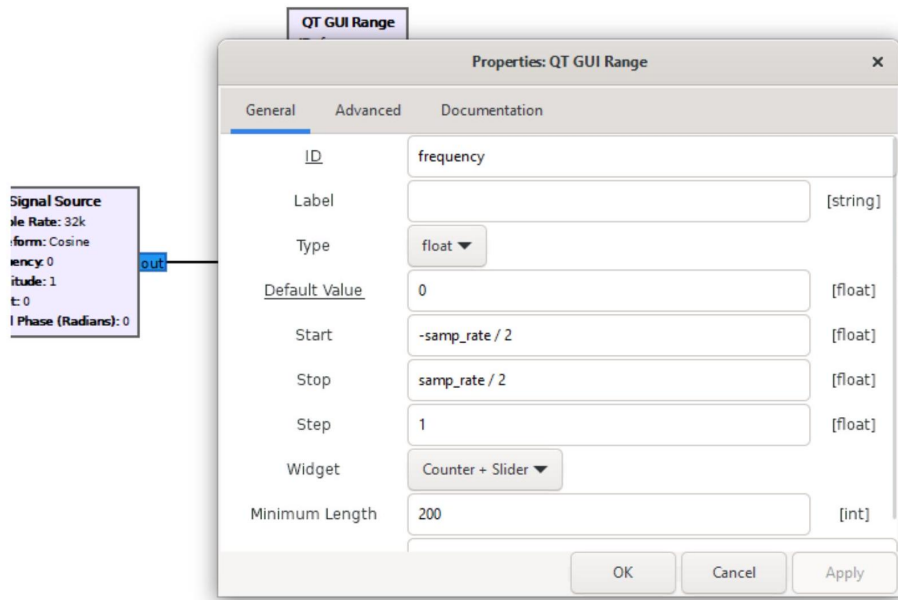
Создадим блок - графа

1. Источник сигнала
2. Фильтр нижних частот
3. Дроссель
4. Приемник частоты QT GUI
5. Диапазон графического интерфейса QT

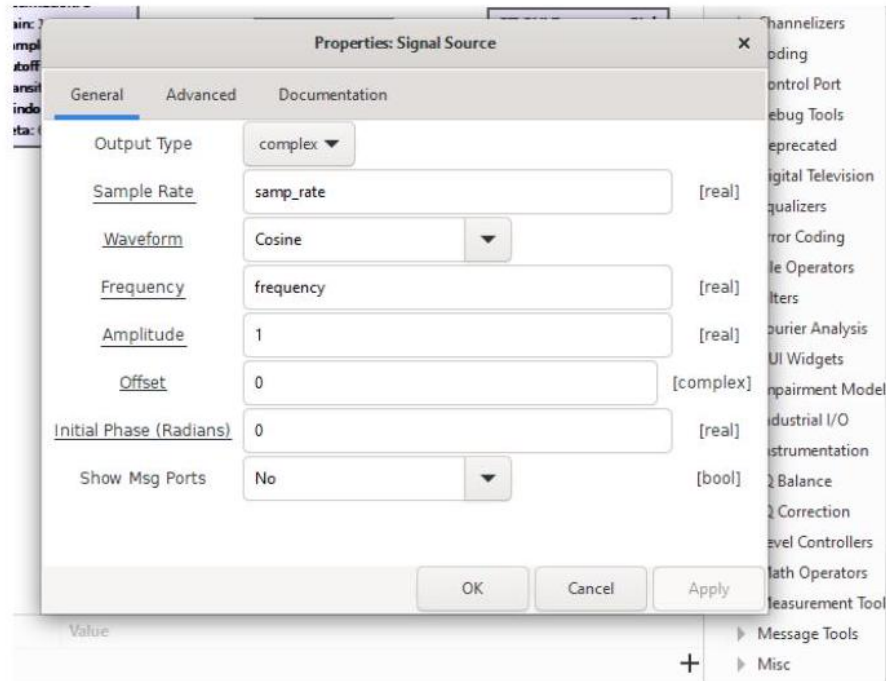


Блок QT GUI Range используется для управления частотой блока источника сигнала. Изменим свойства блока

- Идентификатор: частота
- Значение по умолчанию: 0
- Начало:  $-\text{samp\_rate}/2$
- Стоп:  $\text{samp\_rate}/2$

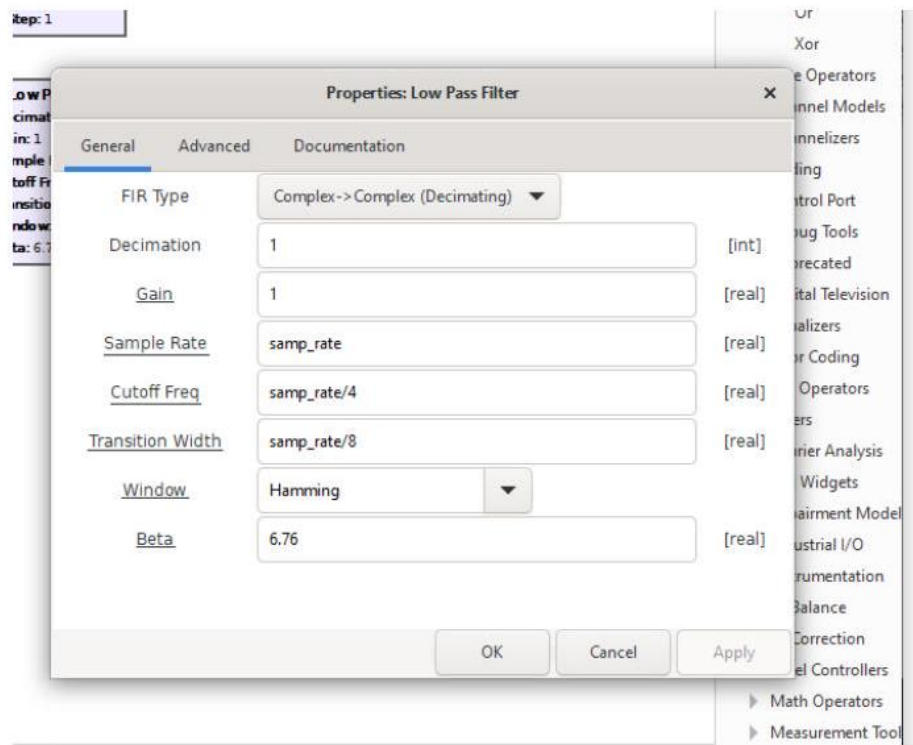


## Изменим свойства Signal Source

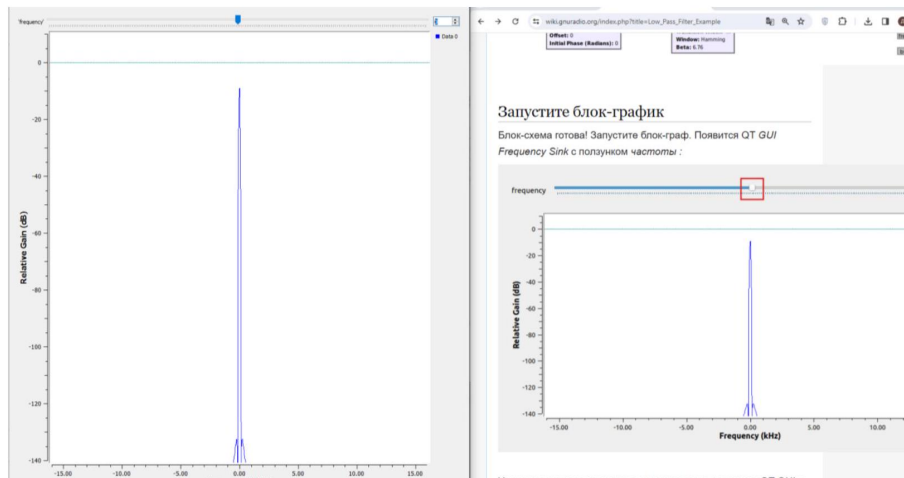


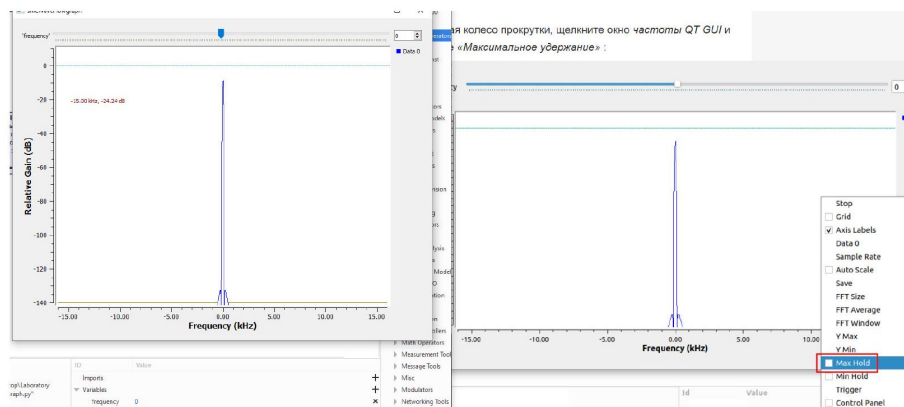
### Изменим свойства блока Low Pass Filter

- Частота среза:  $\text{samp\_rate}/4$
- Ширина перехода:  $\text{samp\_rate}/8$

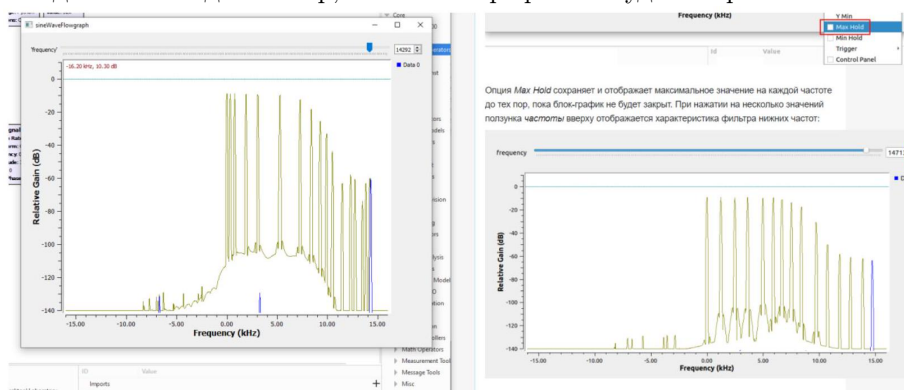


Запустим Блок-график

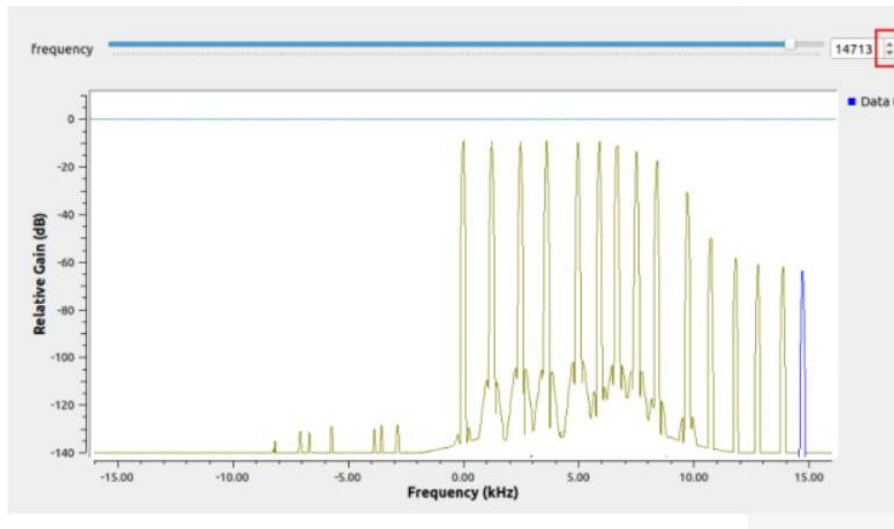




Опция Max Hold сохраняет и отображает максимальное значение на каждой частоте до тех пор, пока блок-график не будет закрыт.



Опция Max Hold сохраняет и отображает максимальное значение на каждой частоте до тех пор, пока блок-график не будет закрыт. При нажатии на несколько значений ползунок частоты вверху отображается характеристика фильтра нижних частот:



Попробуем и другие настройки  
MinHold

