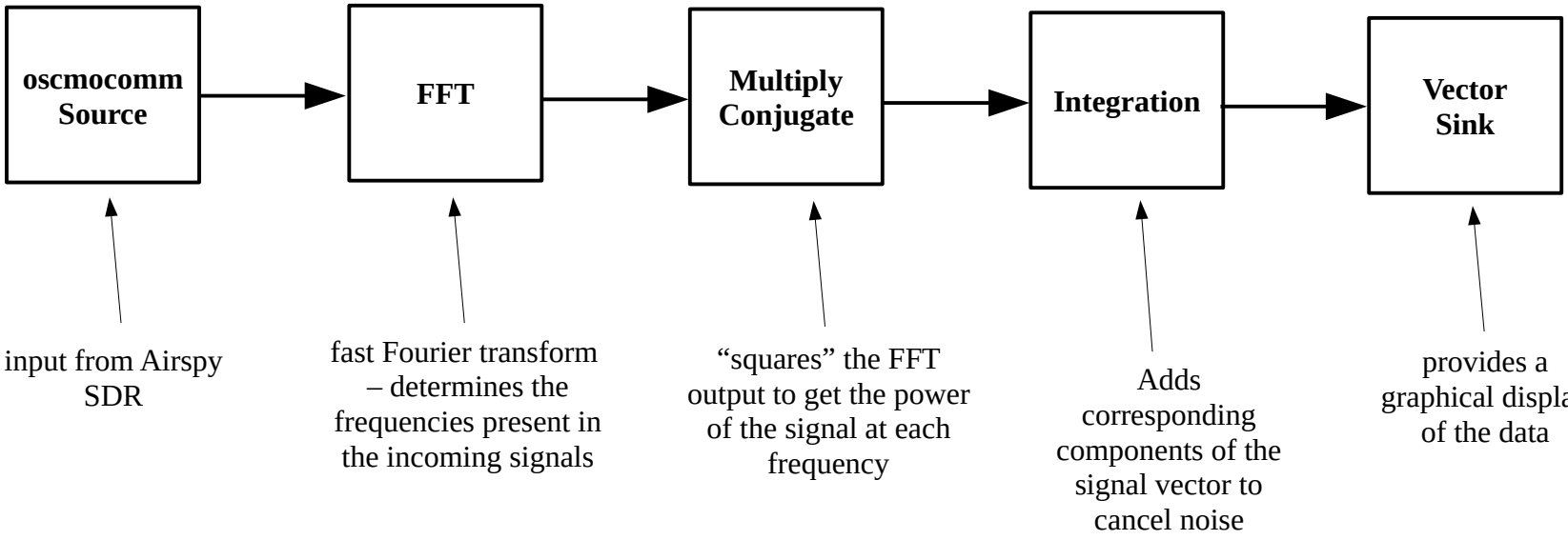
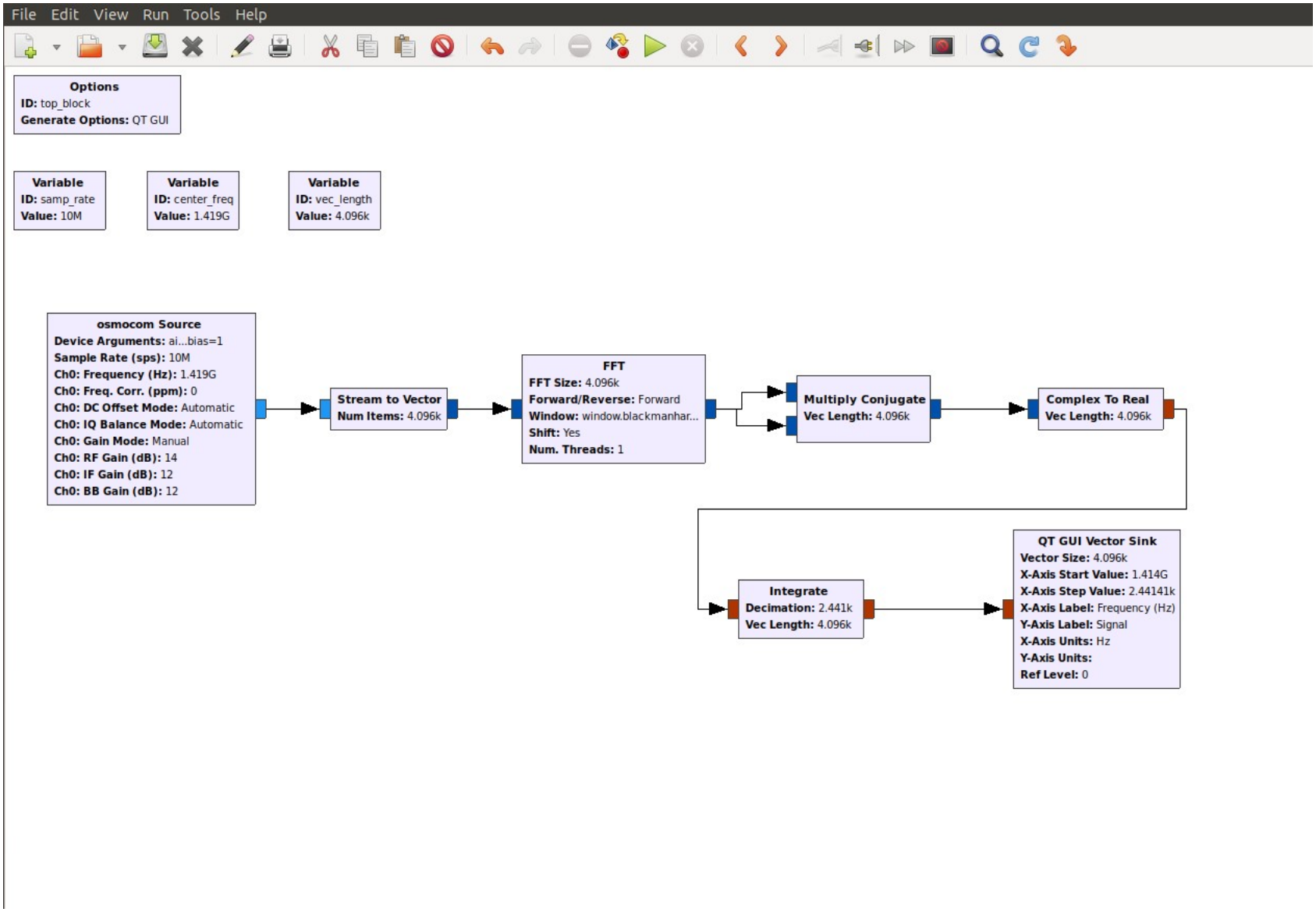
Building a GnuRadio Spectrometer for an HI Radio Telescope Explained

Part 1: A Simple Spectrometer

Block diagram, in its simplest form:



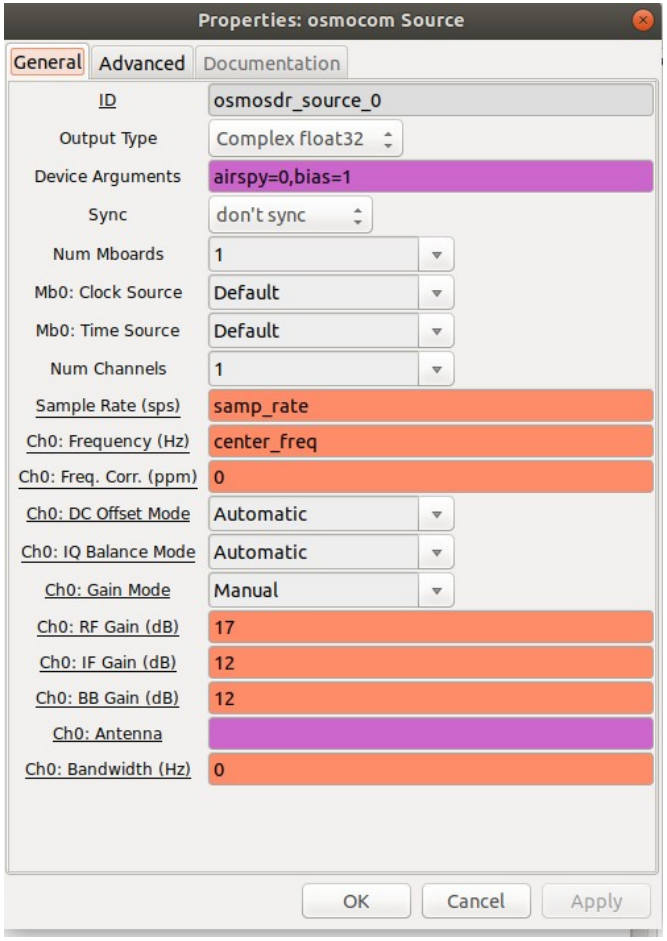
The completed Gnuradio program:



Building the Spectrometer, with Explanations

osmocomm Source block:

- indicates the Airspy as the input block, and its settings.

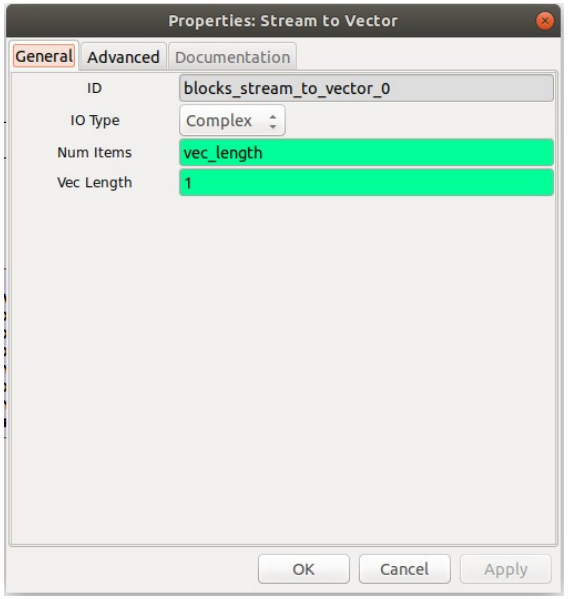


Settings:

- Device Arguments
  - airspy=0 ⟶ identify Airspy as the input device
  - bias=1 ⟶ indicate that the computer is to supply power to Airspy
- sample Rate: samp_rate ⟶ value set as 10 MHz in **Variable** block
- Ch0: Frequency: **center_freq**
  - indicates middle frequency of the 10 MHz Airspy bandwidth, which we will set at 1419 MHz in the center_freq **Variable** block.
- RF Gain: 17
- IF Gain: 12
- Leave thd other settings at their default values.

Stream to Vector block:

- bundles the continuous data stream from the **osmocomm** block into vector chunks to be processed by the FFT block.

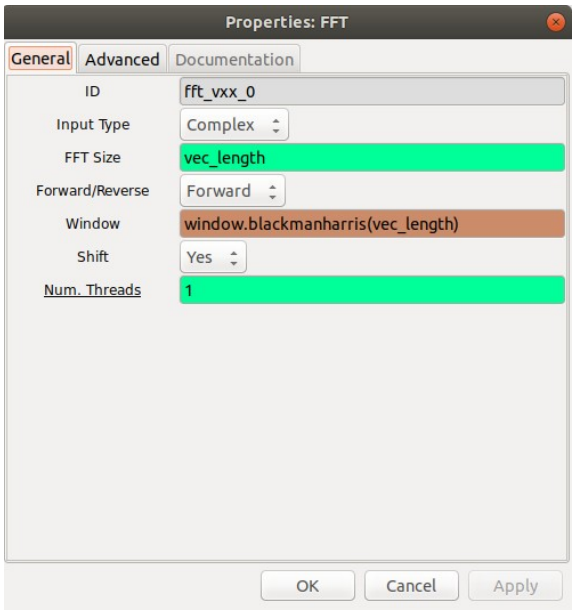


Settings:

- **Num items**: vec_length ⟶ variable set by a **Variable** block (value = 4096)
- **Vec Length**: 1
- NOTE: This nomenclature is confusing; the **Vec Length** is not really a vector length; it indicates that 1 bunch of 4096 **Num Items** is sent into the FFT at a time. This is not to be confused with the variable ***vec_length***, which does describe a vector length.

FFT block:

- performs a Fast Fourier Transform on the data, one vector stream at a time.



Settings:

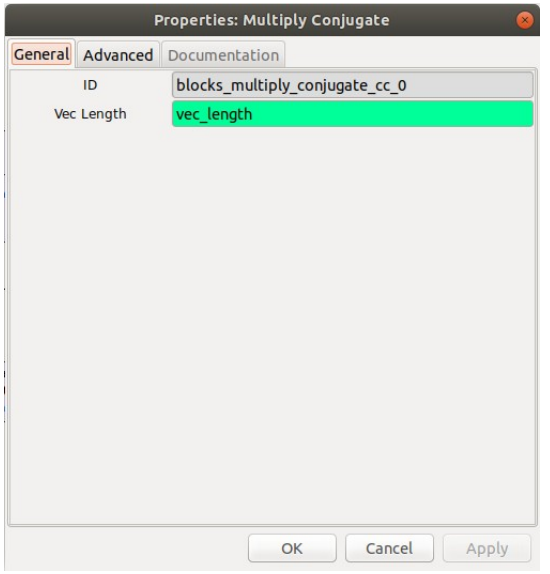
- Input Type: Complex
- FFT Size: vec_length (value = 4096)
- Window: window.blackmanharris(vec_length)
- NOTE: There is a vector size needed in the **Window** function that must match the FFT size. (It's easy to overlook this.)

Multiply Conjugate block:

- calculates the product of the FFT output with its complex conjugate.
- This results in a computation of the spectrum power, which is what we want displayed.

Settings:

- Vec Length: vec_length

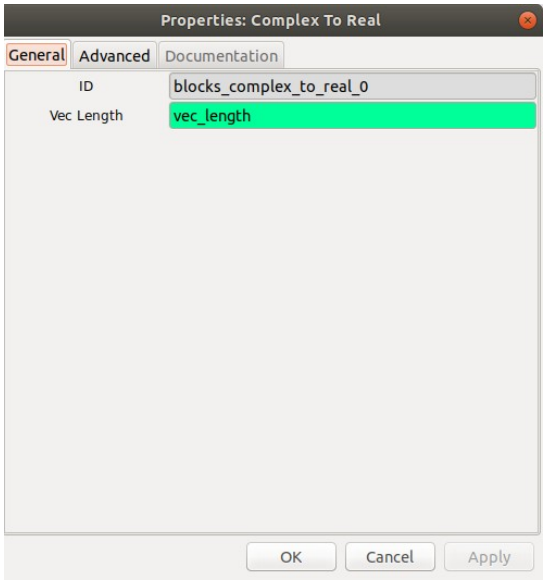


Complex to Real:

- Converts the data stream to a real value.

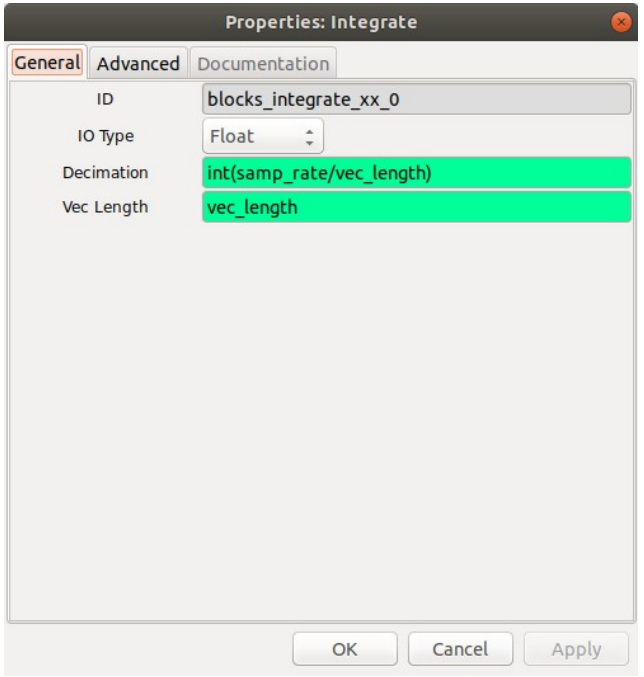
Settings:

- Vec Length: vec_length



Integrate:

- Sums up data by adding corresponding components of the data vector. This provides a method for reducing the noise. Ideally with each summation the noise cancels itself out, due to its randomness, while the signal constructively adds to a greater value with each sum.



Settings:

- IO Type: Float
- Decimation: int(samp_rate/vec_length)
  - This indicates the size or number of data values to add up.
  - Example:

    For samp_rate = 10 MHz = 1.0×10^7 and vec_length = 4096:

    samp_rate/vec_length = 1.0×10^7 Hz/ 4096 = 2441 per sec.

    → 2441 summations are done each second, which means the summation of the vectors are done so that the matching components of the vectors are added, reinforcing peaks at each frequency and canceling corresponding noise at each frequency.
- Vec Length: vec_length

QT GUI Vector Sink:

- This displays the processed signal in a signal vs. frequency graph.

Settings:

- Vec Size: vec_length
- X-Axis Start Value: 1414 MHz = 1414e6
  - We want the 1420.4 MHz of the HI signal to be displayed approximately in the middle of the graph. We will set the 10 MHz sample rate of the Airspy to start at 1414 MHz and stop at 1424 MHz.
- X-Axis Step value: samp_rate/vec_length
  - samp_rate/vec_length = 1.0×10^7 Hz/ 4096 = 2441 channels from 1414 MHz to 1424 MHz
- Axis labels are as indicated.
- NOTE: The axis units are not automatically displayed on the graph, which is why they are included in the **X-Axis Label**. The **X-Axis Units** setting will enable the units of the x-axis (Hz) to be displayed when a the cursor is moved to an (x, y) coordinate on the screen.
- Autoscale: Yes
- Y min: 0
- Ymax: whatever (since Autoscale is on)
- Leave other settings at their default values.