RF Fundamentals

GNU Radio

Outline

- GNU Radio Introduction
- GNU Radio Hands-on Exercises
- Q&A

GNU Radio

- We will use GNU Radio to
  - analyze & simulate signals
  - teach signal processing
- So let’s learn how to use it first!

GNU Radio is…

- A signal processing library
- Designed for real-time
- The software part of an SDR
- Not a radio application
- The tool to **build your own** transceivers
- **FOSS**: Free and Open Source Software

GNURadio
THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM

GNU Radio

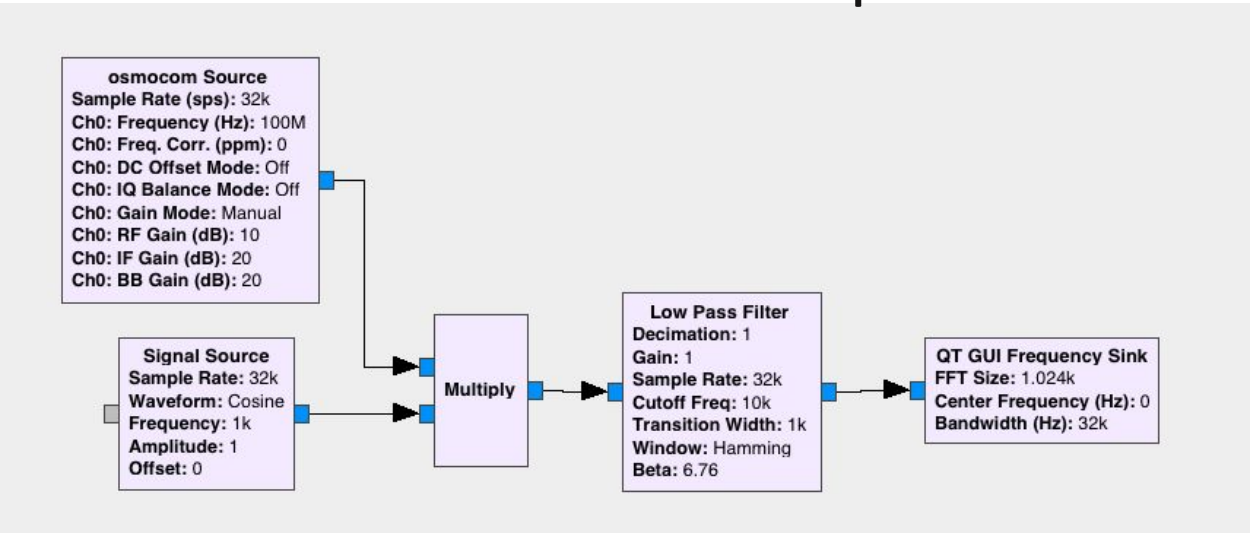
- GNU Radio is an open source framework to do DSP for radio (communications, RADAR, radio astronomy...).
- Also useful for applications that do similar computing (even particle accelerators!).
- It comes with:
  - A GUI application called “GNU Radio companion” where systems can be implemented by dragging and dropping blocks onto a canvas (making a “flowgraph”)
  - A rich library of processing blocks accessible both through GNU Radio companion, and C++ and Python APIs
  - A “runtime”, that moves data between these blocks and runs the code of each block
- In the GNU Radio ecosystem there are out-of-tree modules, which implement new blocks that don’t fit into or exist in the in-tree library
- There are also full applications that use GNU Radio for their DSP (for instance, GQRX, or QRadioLink)

GNU Radio

- Open-source framework for SDR and signal processing
- Founded by Eric Blossom in 2001
- Block-based dataflow architecture
- Each block runs in its own thread
- Data flows through a graph called a Flowgraph
- Blocks are nodes in a Flowgraph, and perform operations and signal processing
- Signals normalized between -1.0 and +1.0
- Similar in concept to MathWorks SimulinkTM
- Running C++ and Python under-the-hood
- Can write code directly, or use the GNU Radio Companion (GRC) graphical tool

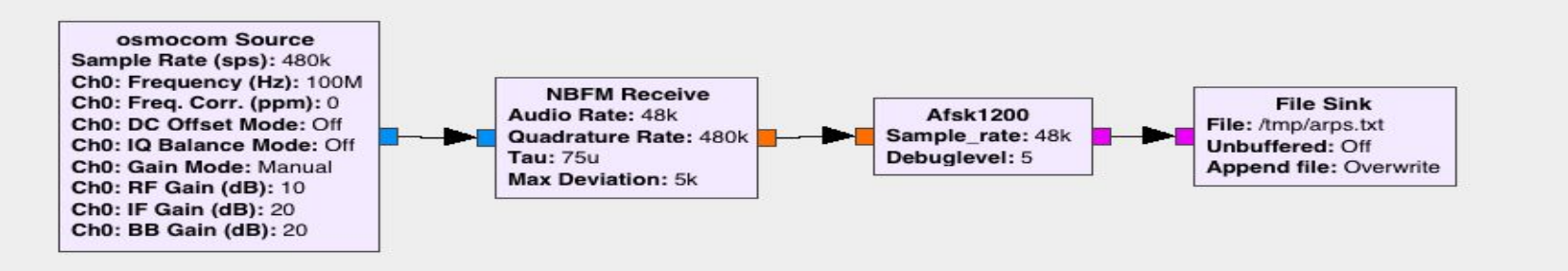
Basic Concept: Flow Graph

- Transceivers are implemented as *flow graphs*
- Similar to Simulink / schematics
- Define structure and parameters of *blocks*



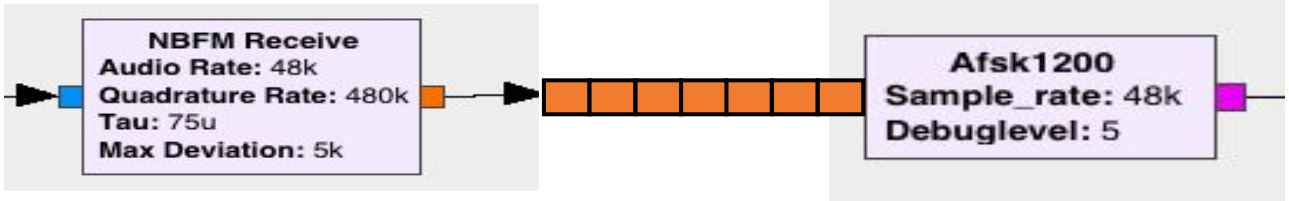
Basic Concept: Block

- Written in C++ or Python
- Implement one logical step
- Each block run in separate thread

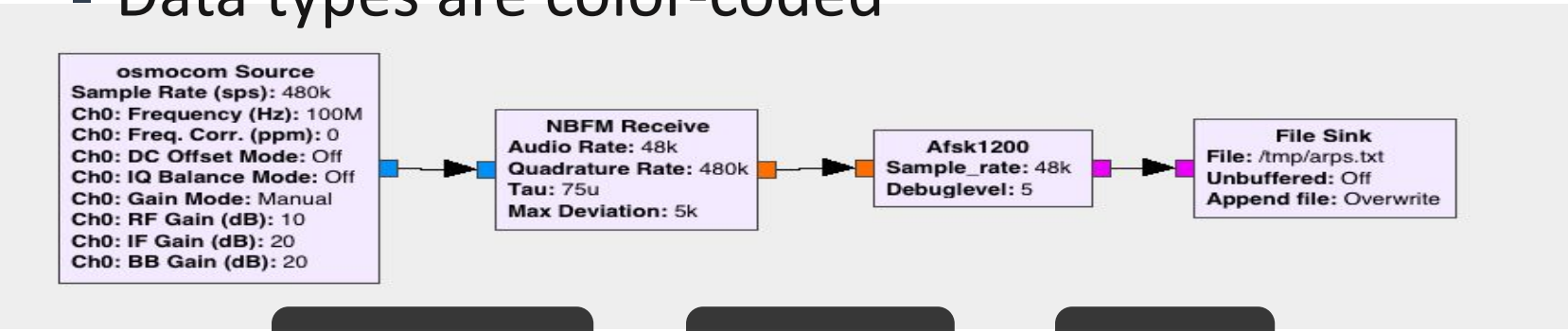


Data Streams

- Samples are buffered



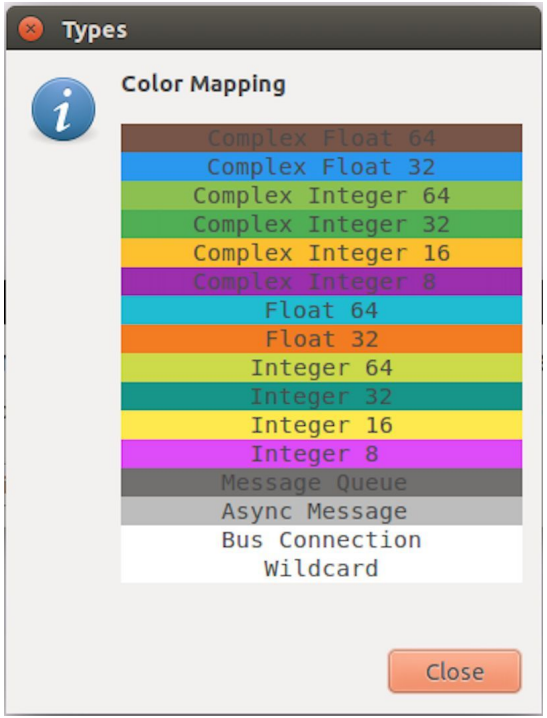
- Data types are color-coded



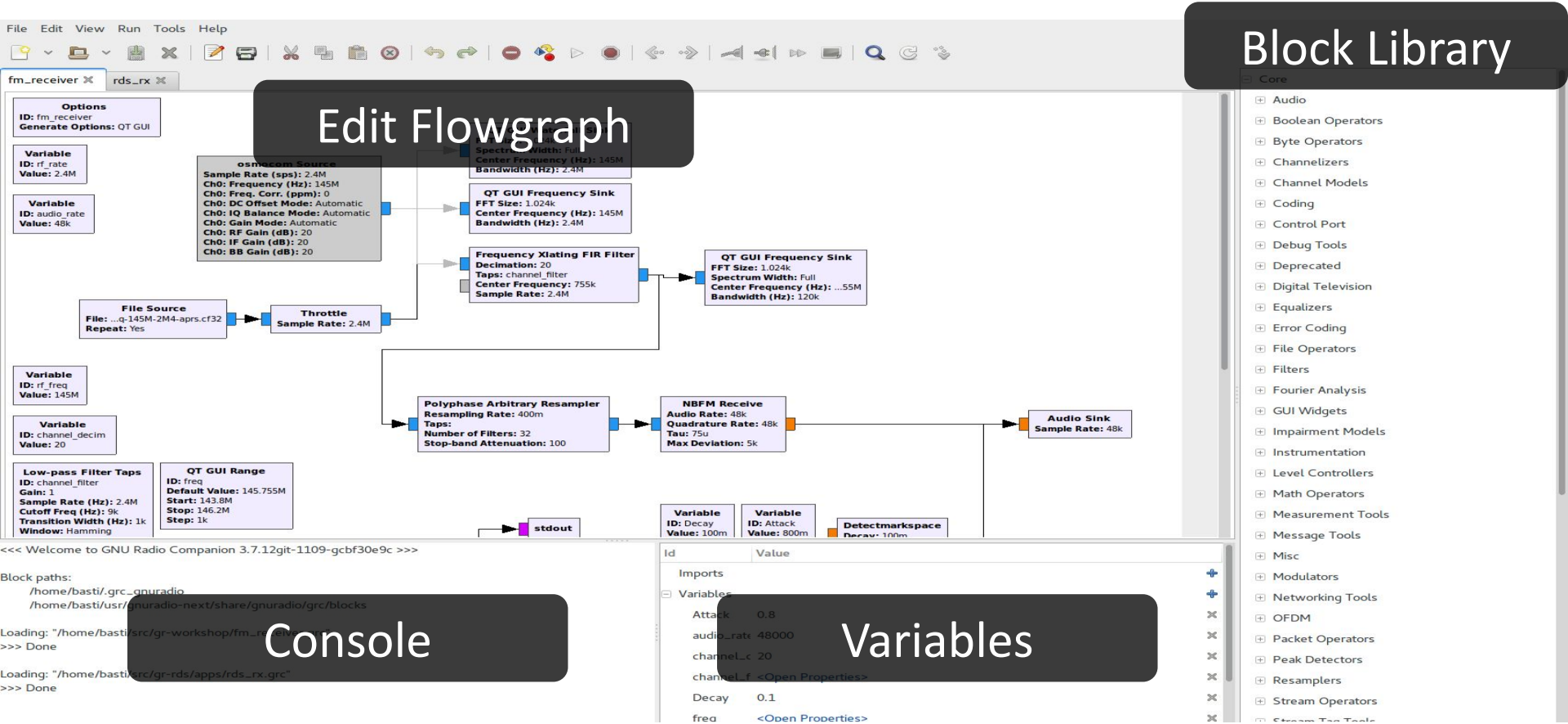
Complex Float Byte

Color Types

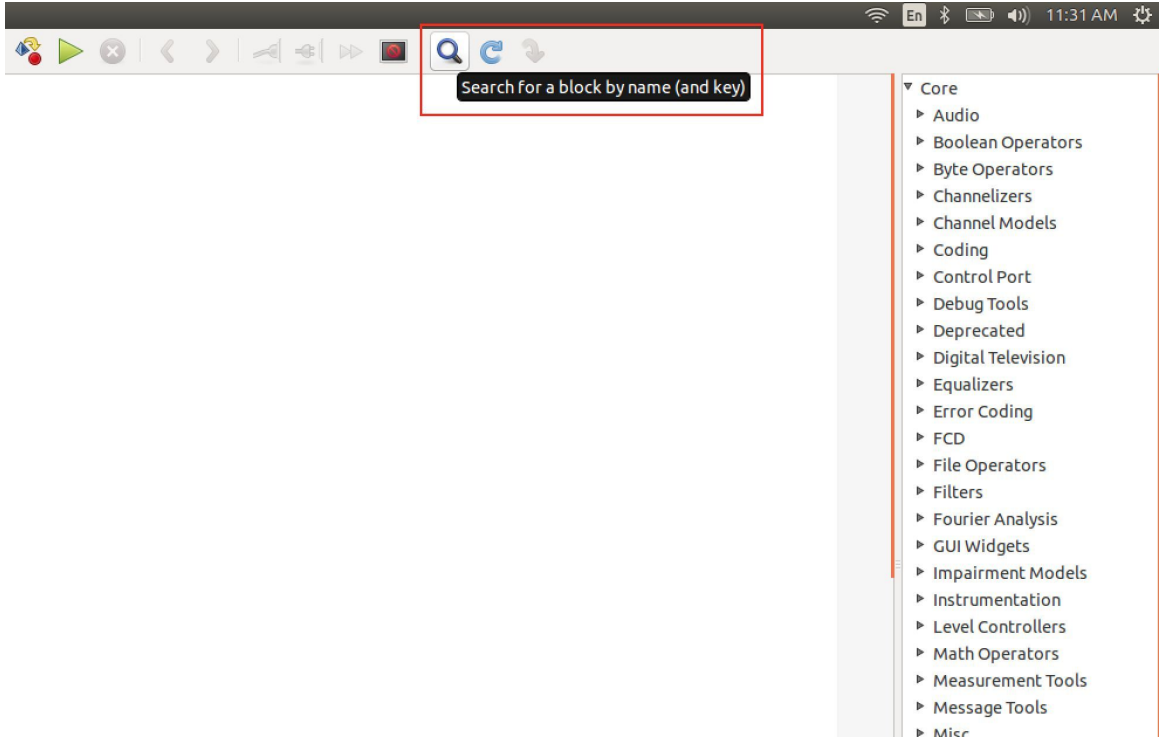
Click on menu item Help->Types



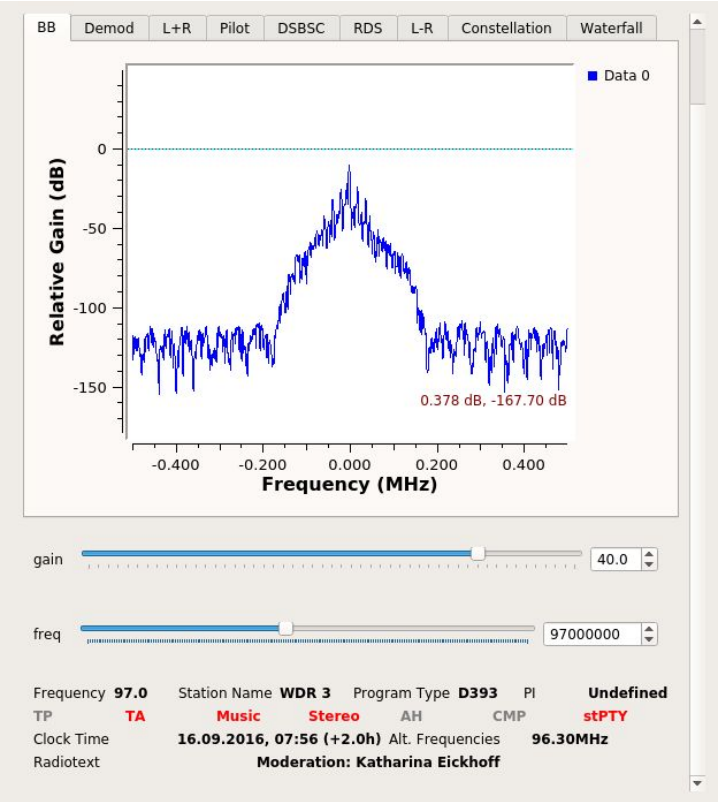
GNU Radio Companion

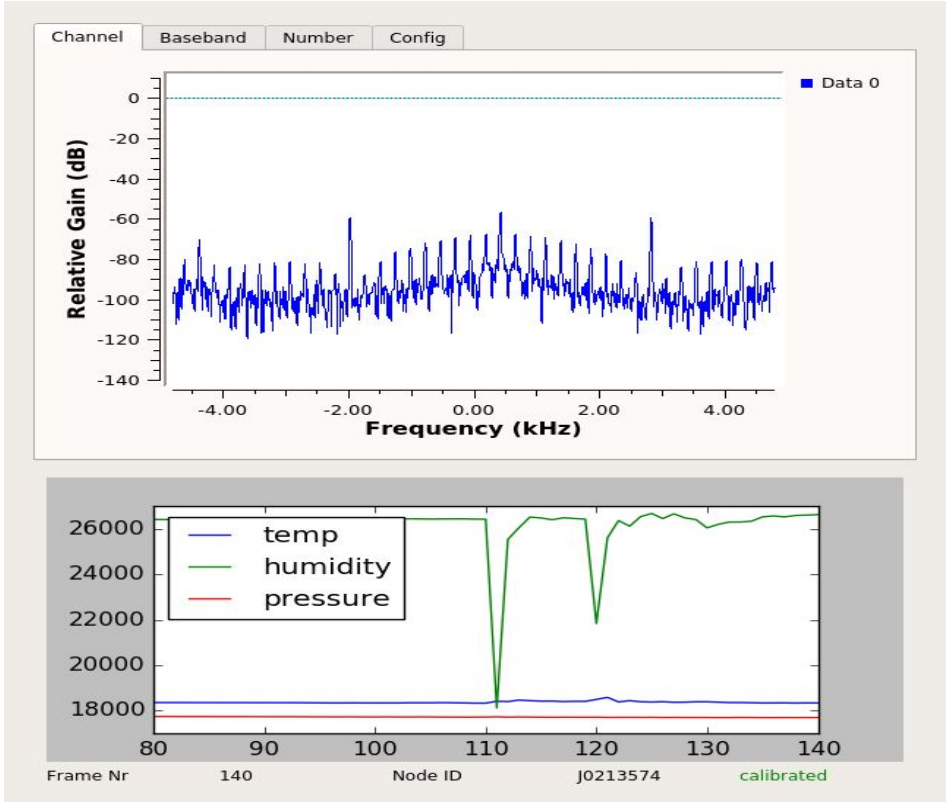


Search Blocks

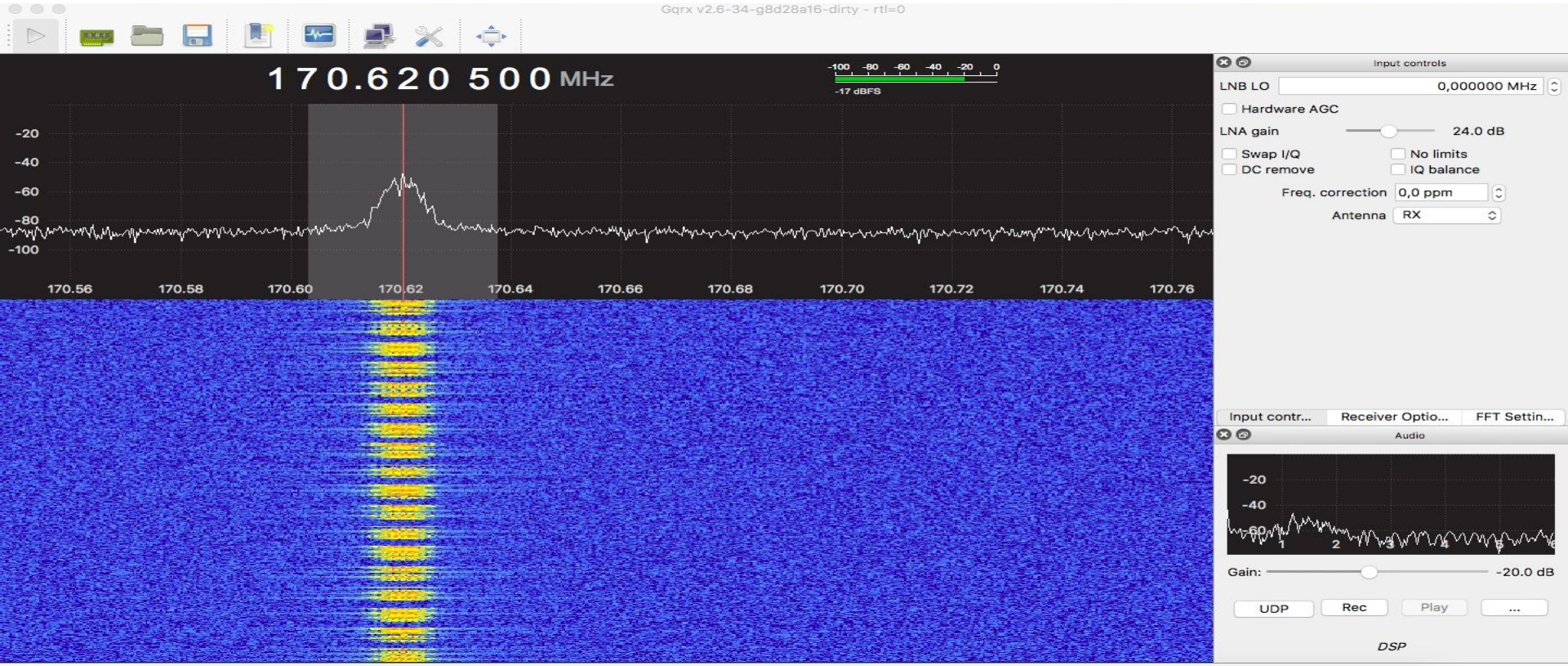


GUI Output and Instrumentation



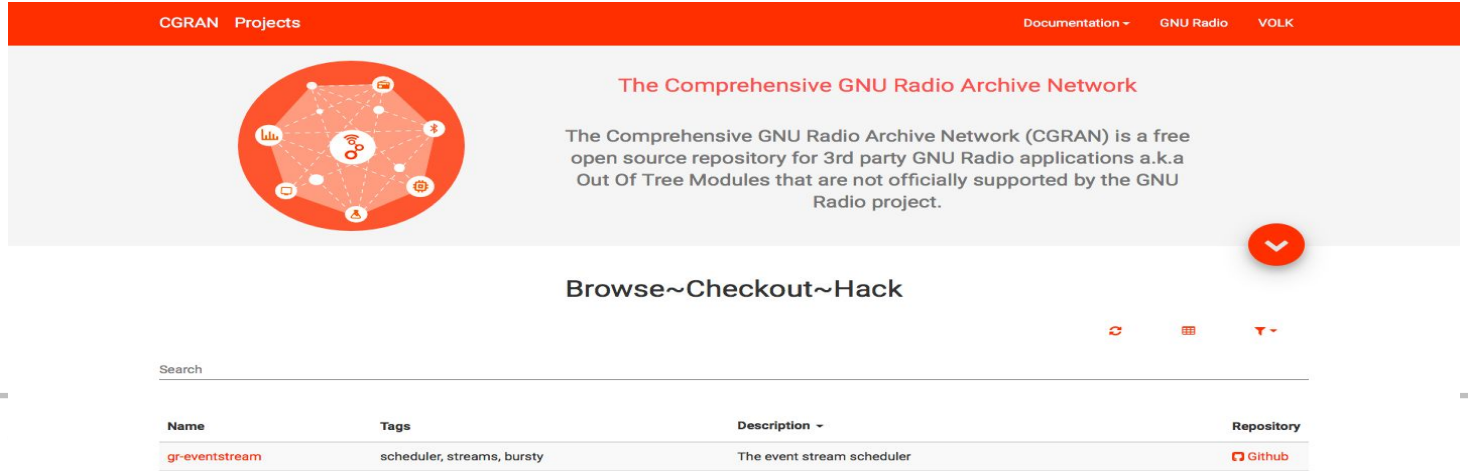


GQRX - a GNU Radio Application

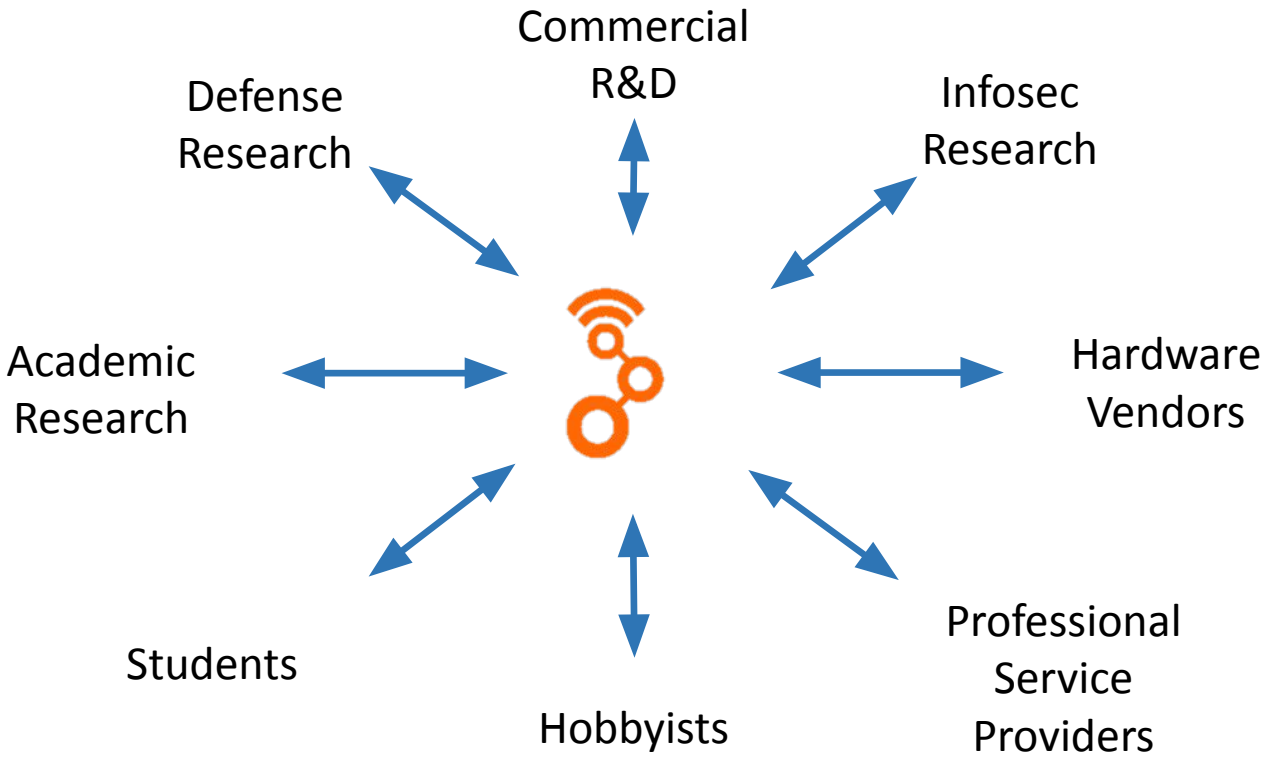


Out Of Tree Modules

- GNU Radio can be extended with OOTs
- OOTs cover more specific functionality
- There is a large number available
- CGRAN is our central database



GNU Radio is used by



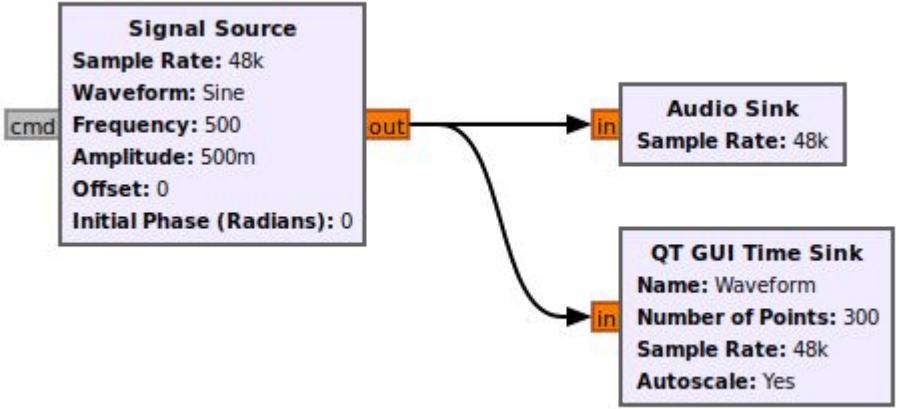
GNU Radio is an Ecosystem

- Active Open Source community since 2001
- PyBombs, OOTs
- GRCon since 2011
- GNU Radio Foundation
- FOSDEM SDR DevRoom
- GSoC, SoCIS, R&S Competition, SDR Academy
- GNU Radio Europe

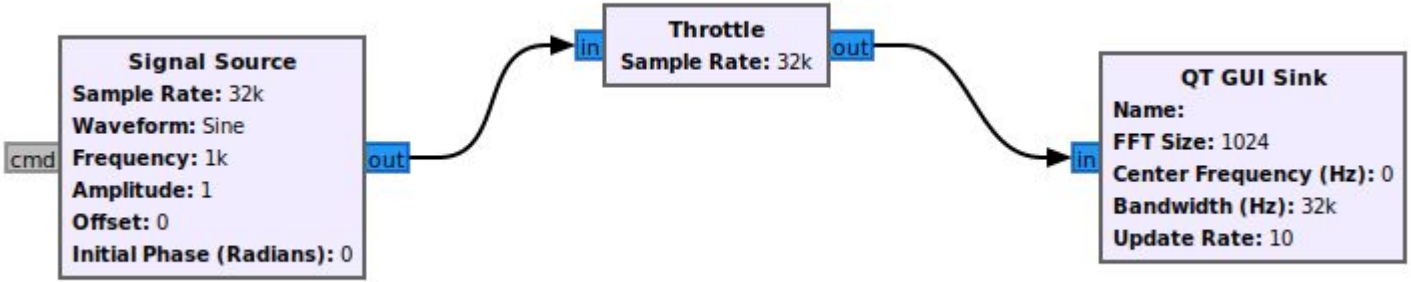
Learn // Discuss // Connect

- Website: www.gnuradio.org
- Development: github.com/gnuradio
- **Mailing List: discuss-gnuradio@gnu.org**
- Wiki: wiki.gnuradio.org
- Slack: slack.gnuradio.org
- Facebook: gnuradioproject
- Twitter: @gnuradio

Signals in Time Domain



Exploration of Signals in Frequency Domain

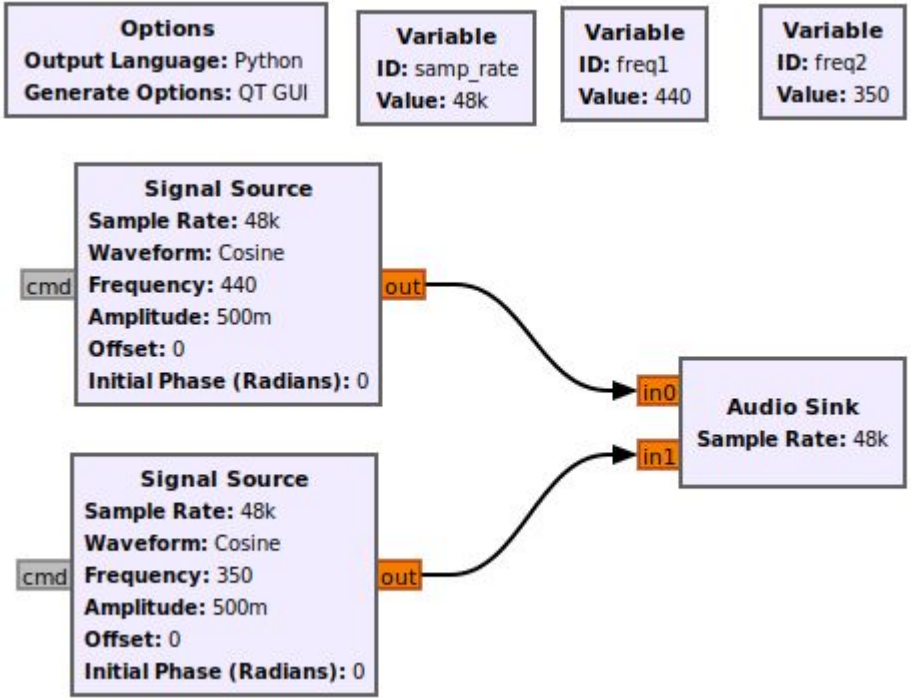


Producing Stereo Sound

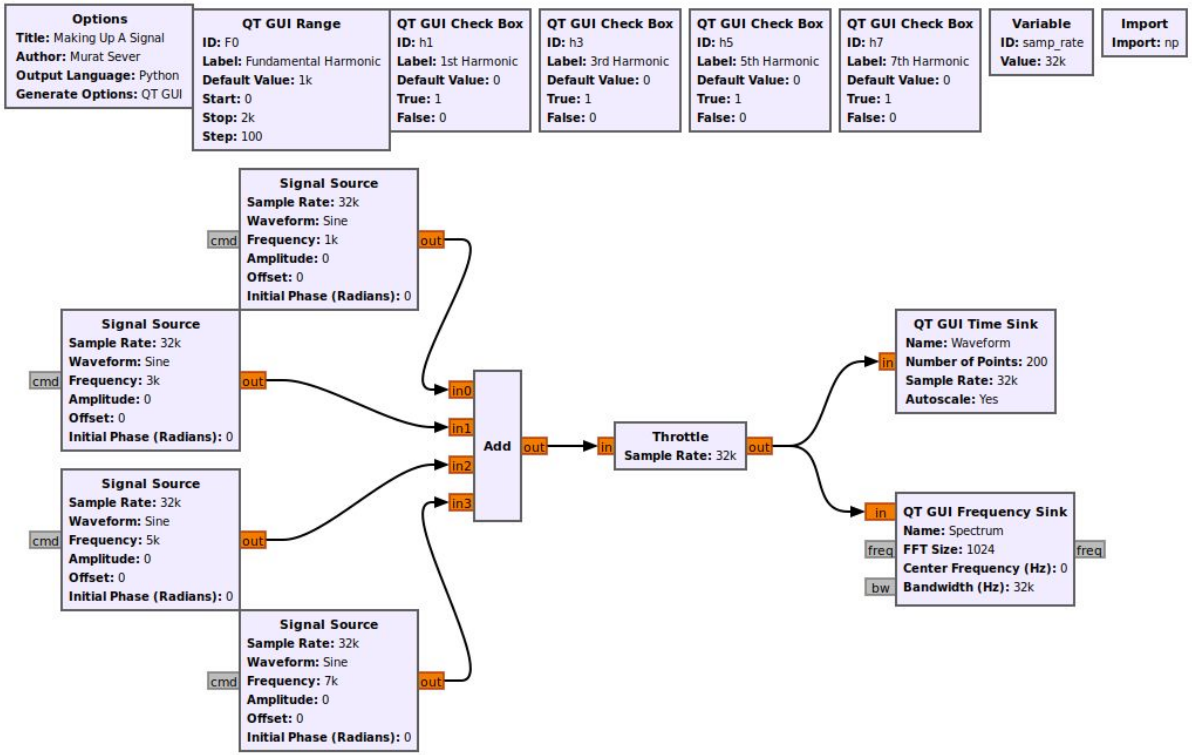
- GNU Radio provides access to sound card via “audio source” and “audio sink” blocks.
- Create the following flowgraph to produce a stereo sound
- Then modify it to produce # dial tone (DTMF tone)

Hint: Search for frequencies to make up # dial tone!

Producing Stereo Sound



Making Up a Signal



Using GNU Radio from Python

- Generate Python from GRC Flow graph
- Invoke directly from the Linux command line:
  - $ python3 makingupasignal.py

Q&A

1. What does throttle block do?
2. Can I add my own functionality to GNU Radio?

Answers

1. A Throttle Block will simply apply host-based timing (against the 'wall clock') to control the rate of the samples it produces (i.e. samples that it makes available on its outputs to downstream blocks)
2. Yes, you can write an "out of tree" module. It won't live inside the GNU Radio source code