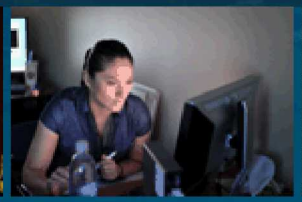
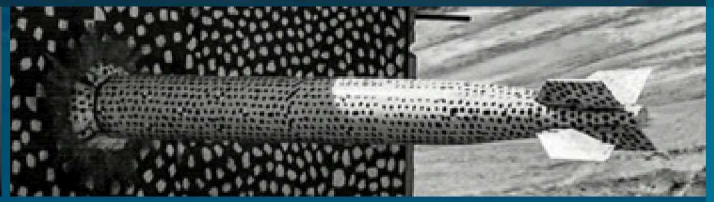
This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2019-3226C

Accessing the RF Spectrum with GNU Radio





PRESENTED BY

Jacob A Gilbert

at GOMACTech – 25 March 2019

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Presentation Summary

Objective:

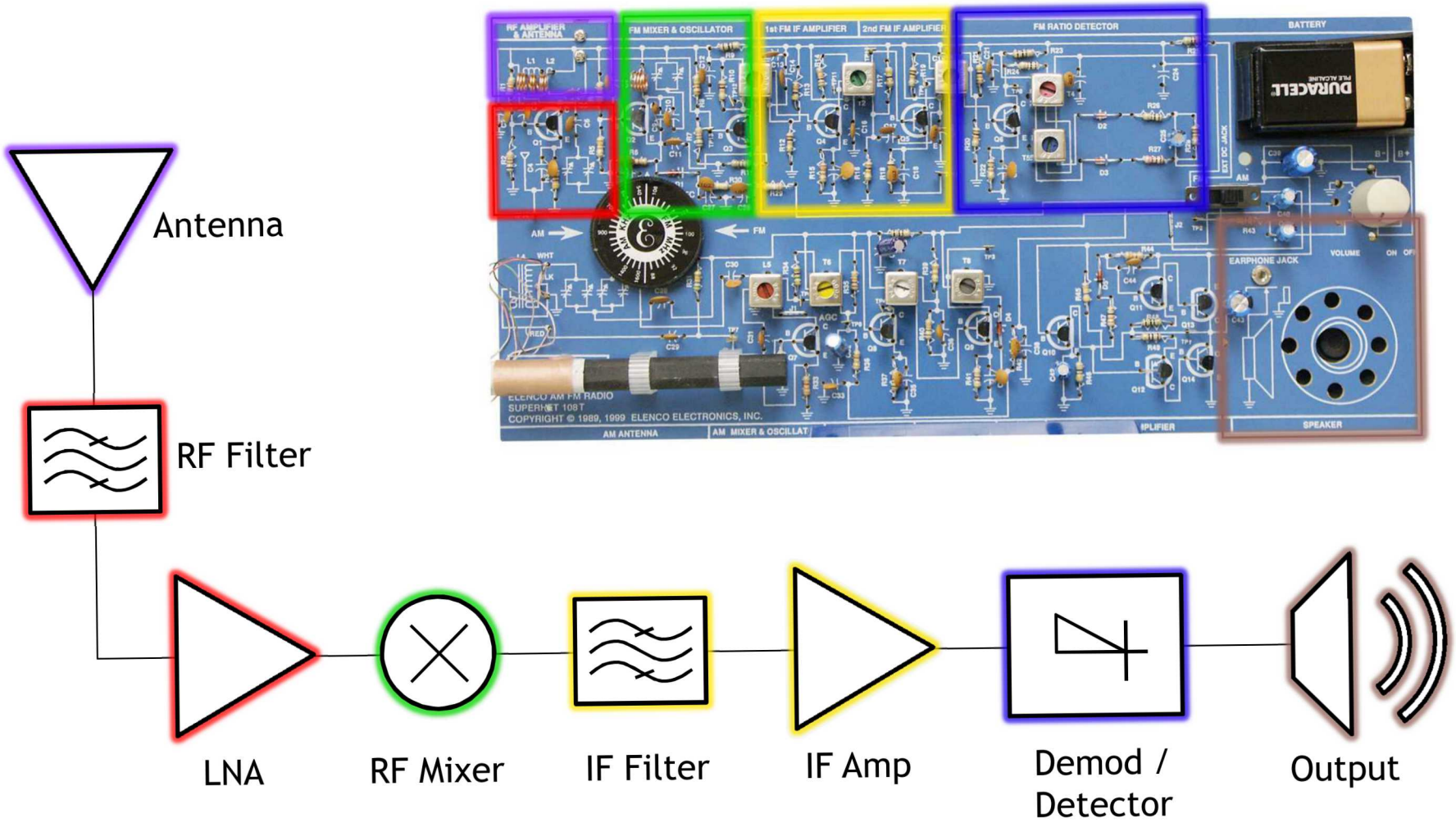
Provide an overview of GNU Radio, and how it enables access to the RF spectrum.

Content:

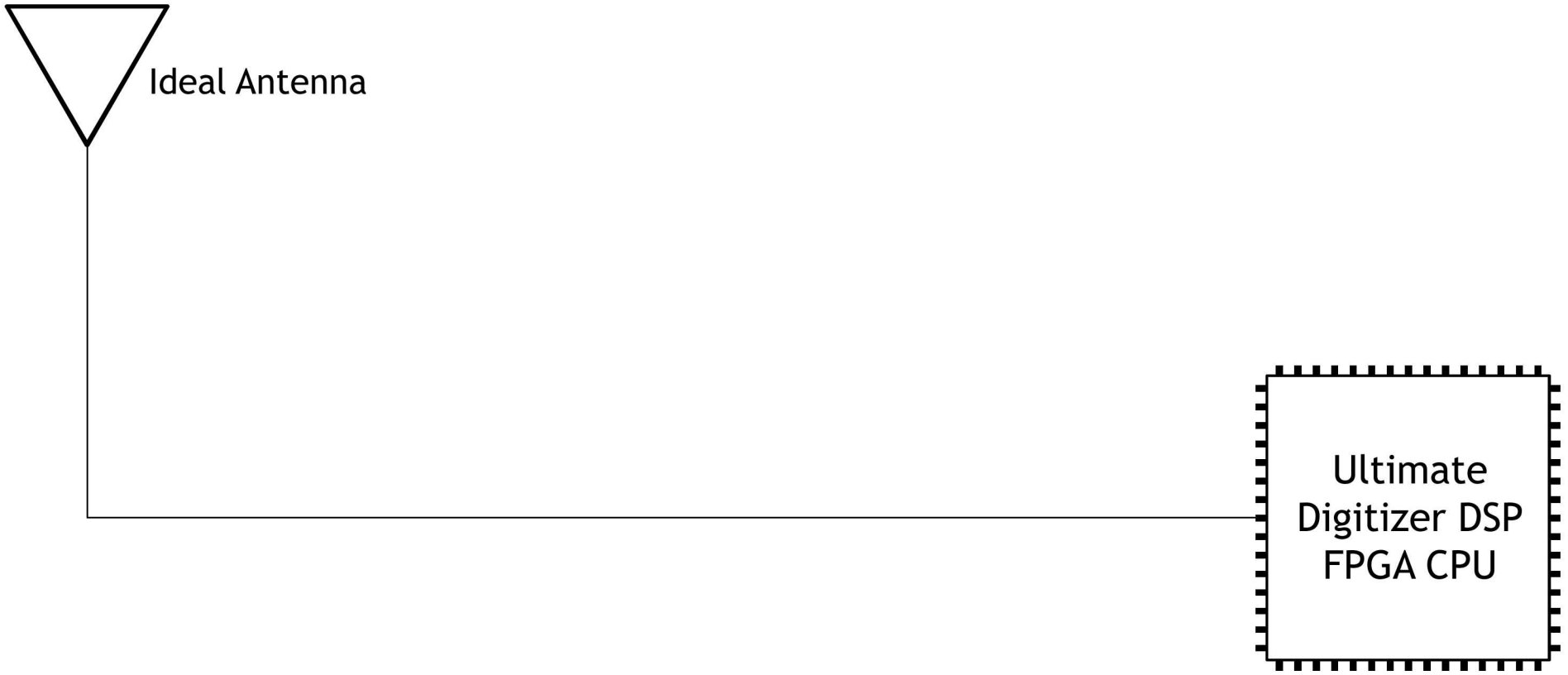
- Brief history of Software Radio
- GNU Radio Overview
- SDR Applications
- Takeaways

What is Software Radio?

What is ***NOT*** Software Radio?

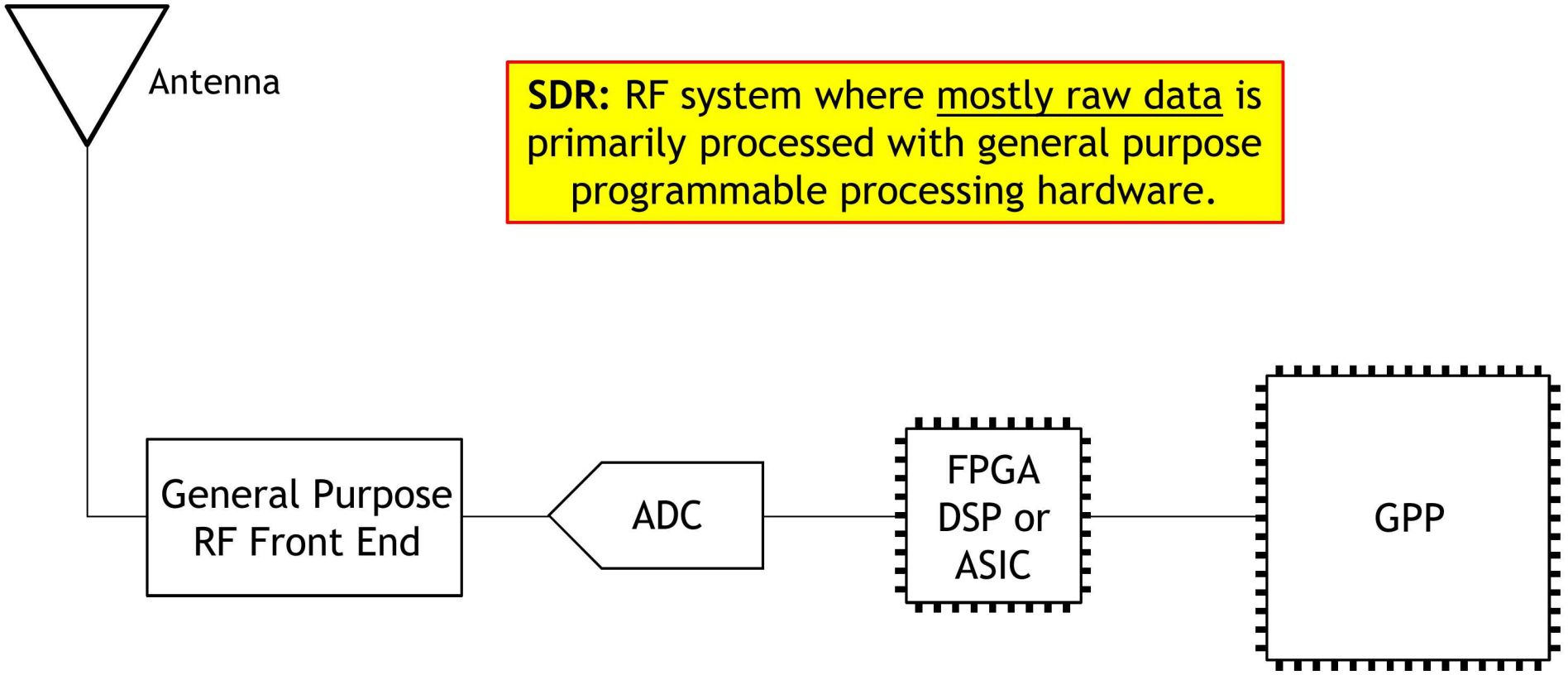


The ‘Ideal SDR’



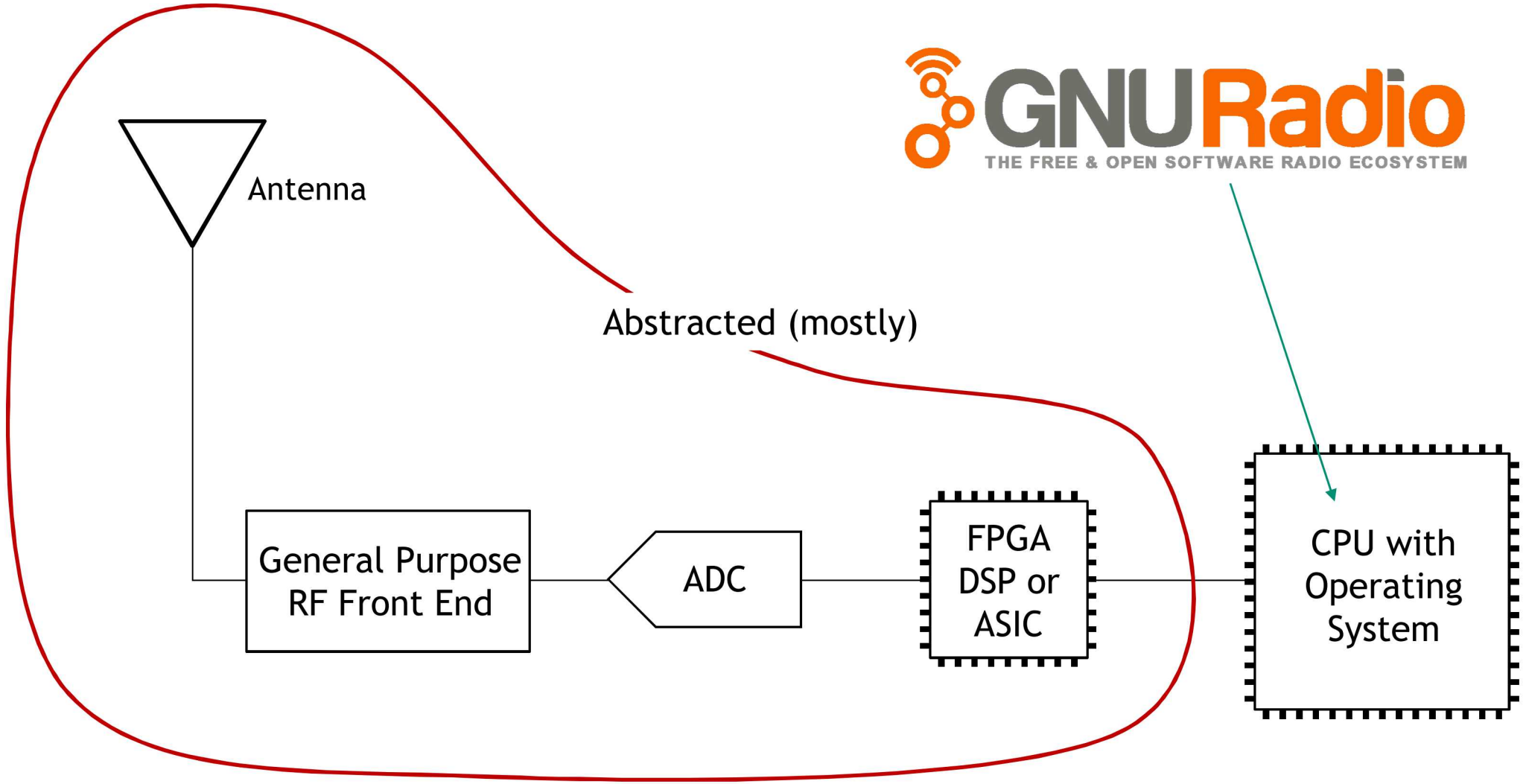
The ‘Ideal SDR’

What is a Software Radio?



SDR: RF system where mostly raw data is primarily processed with general purpose programmable processing hardware.

What is Software Radio?



How Did We Get Here? (the quick version)

1970's: Gold Room TRW - Midas Baseband DSP

1980's: E-Systems: "Software Radio"

1990's: SPEAKEasy, J. Mitola NTC-92, JTRS/CORBA

2001: John Gilmore $320k grant for GR to Eric Blossom

2004: Ettus Universal Software Radio Peripheral (USRP)

2006: GR 3.0 Released

2010: Tom Rondeau takes over as project lead

2013: GR 3.7 released

2016: Ben Hilburn takes over as project lead

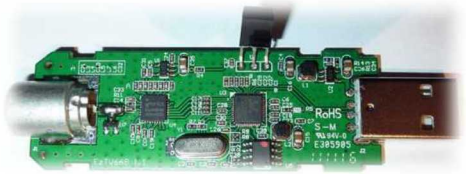
2019: GR 3.8 released (?!)

Democratized Spectrum Access – Commodity SDR Hardware

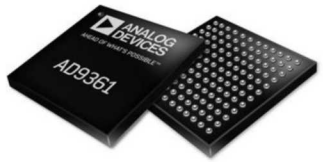
Proliferation of Software Radio is strongly influenced by available SDR hardware

Early SDR Hardware

Ettus Research USRP1



RTL-SDR

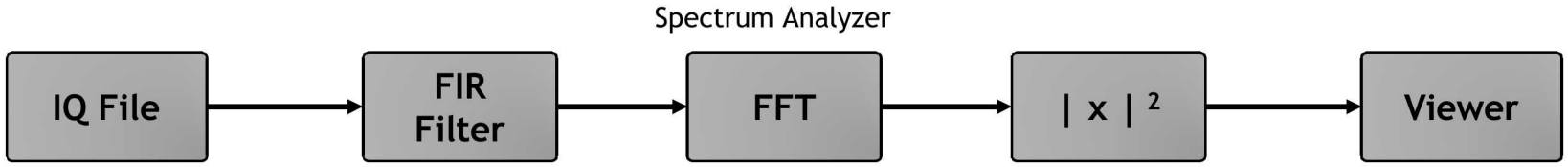


Analog Devices AD9361

What is GNU Radio?

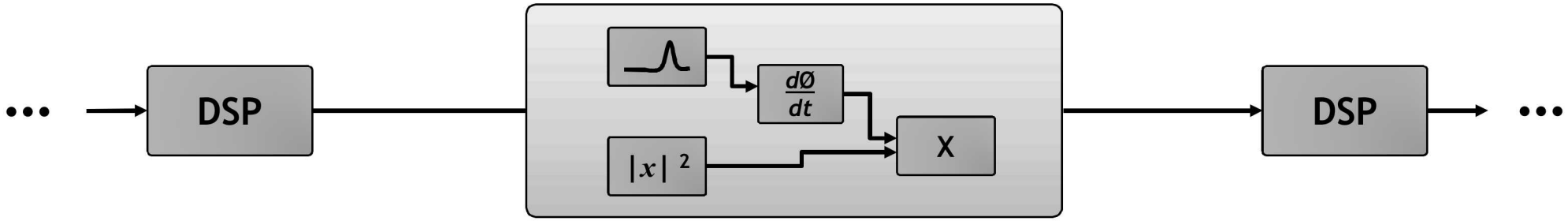
What is GNU Radio?

- **Powerful, open source framework for streaming DSP execution**



What is GNU Radio?

- Powerful, open source framework for streaming DSP execution
- **Substantial collection of common DSP functionality ready-to-use**



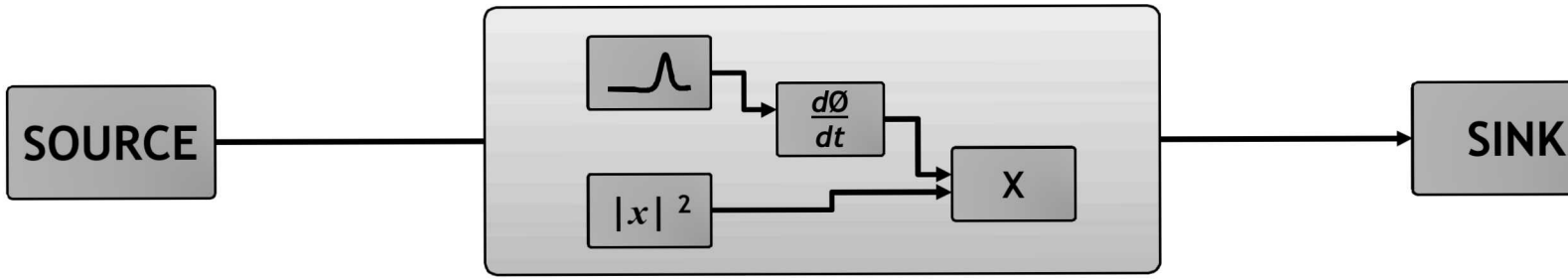
- Arithmetic Operations
- Digital Filtering
- Resampling
- Transforms
- Synchronization

- Boolean Operations
- Modulators / Demodulators
- Forward Error Correction
- Data Simulation Tools
- Data Conversion / Mapping

What is GNU Radio?

- Powerful, open source framework for streaming DSP execution
- Substantial collection of common DSP functionality ready to use
- **Provisions for moving data into and out of GR**

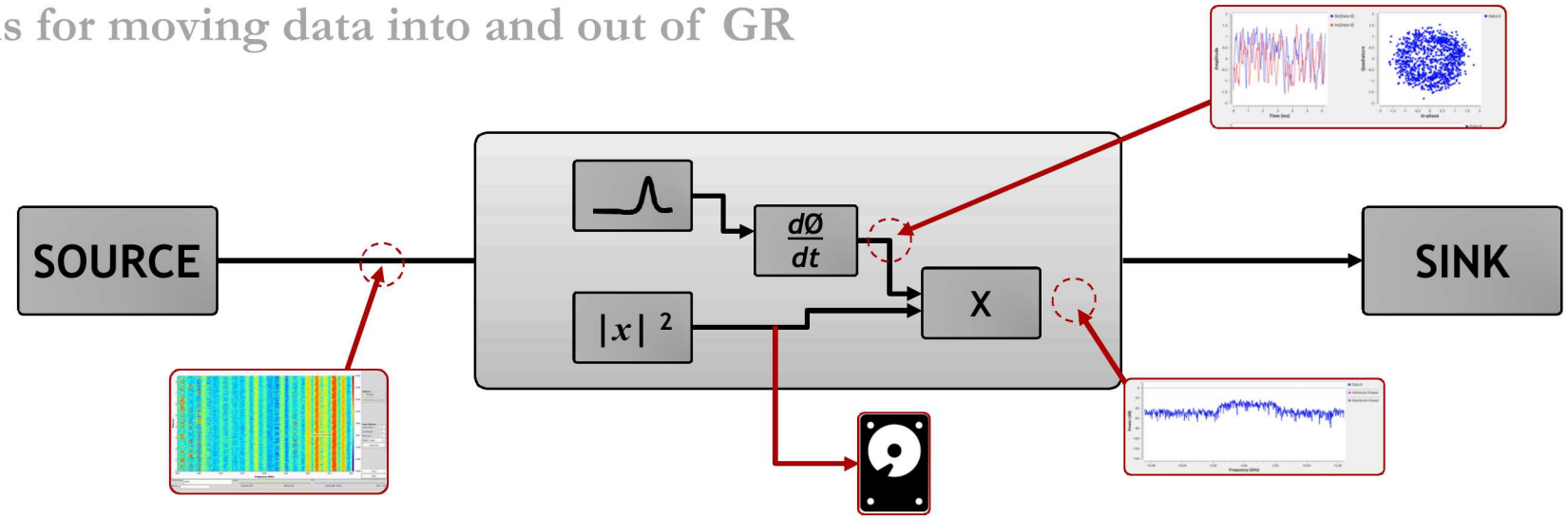
- File
- RF Hardware
- Network
- Sig Gen



- File
- RF Hardware
- Network
- GUI

What is GNU Radio?

- Powerful, open source framework for streaming DSP execution
- Substantial collection of common DSP functionality ready to use
- Provisions for moving data into and out of GR

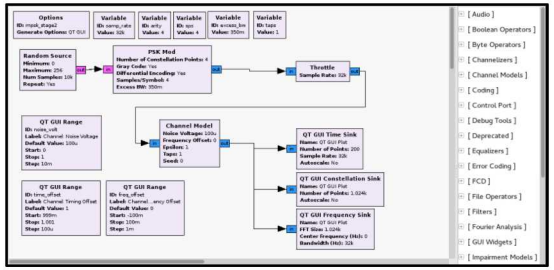


➢ **Flexible analysis and debugging features**

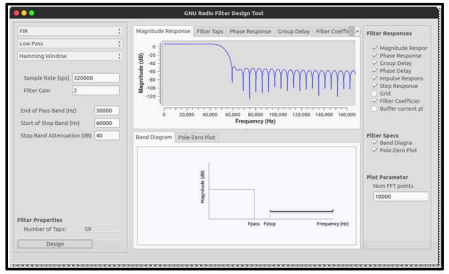
What is GNU Radio?

- Powerful, open source framework for streaming DSP execution
- Substantial collection of common DSP functionality ready to use
- Provisions for moving data into and out of GR

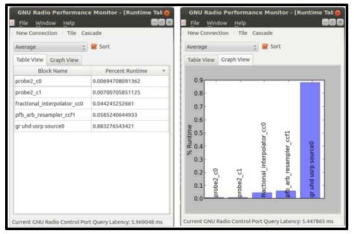
GNU Radio Companion



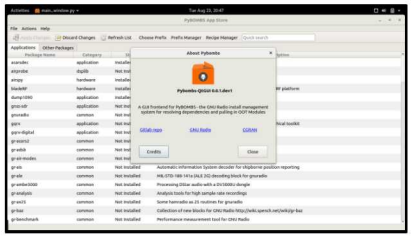
GR Filter Design Tool



GR ControlPort



PyBombs



- Flexible analysis and debugging features
- **Collection of tools and APIs to facilitate streaming DSP applications**

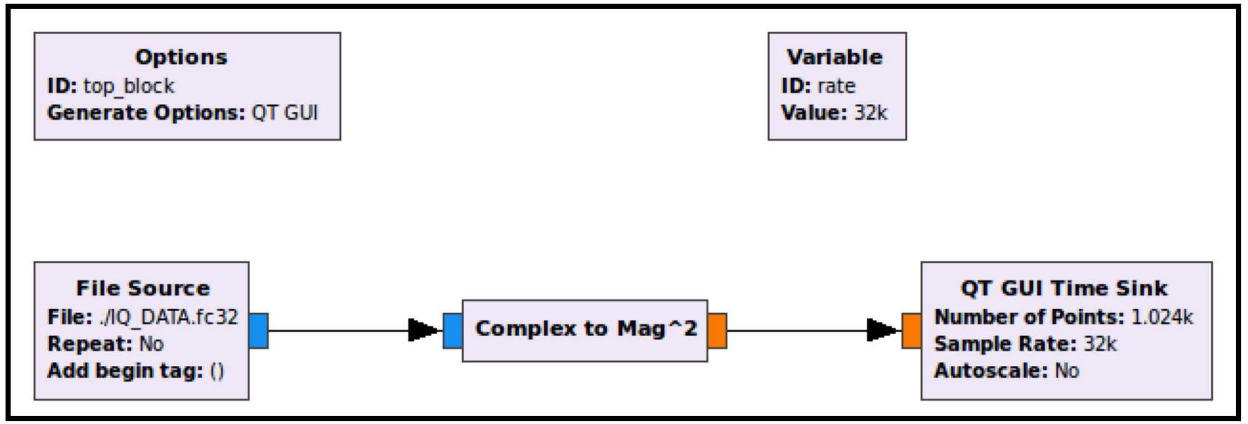
What is GNU Radio?

- Powerful, open source framework for streaming DSP execution
- Substantial collection of common DSP functionality ready to use
- Provisions for moving data into and out of GR

#irc

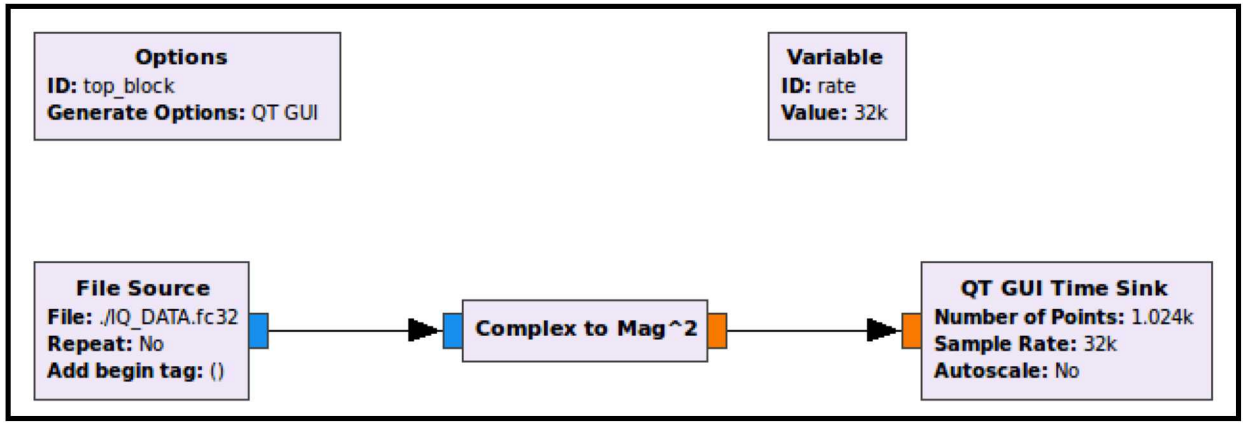
- Flexible analysis and debugging features
- Collection of tools and APIs to facilitate streaming DSP applications
- **Substantial and capable community**

GNU Radio Basic Theory of Operation



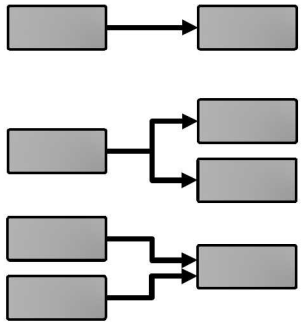
GNU Radio Flowgraph

GNU Radio Basic Theory of Operation



GNU Radio Flowgraph

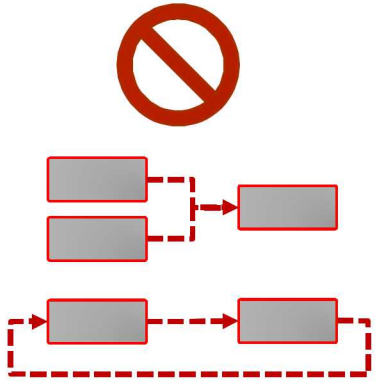
OK



Block to single block

1:N output connections

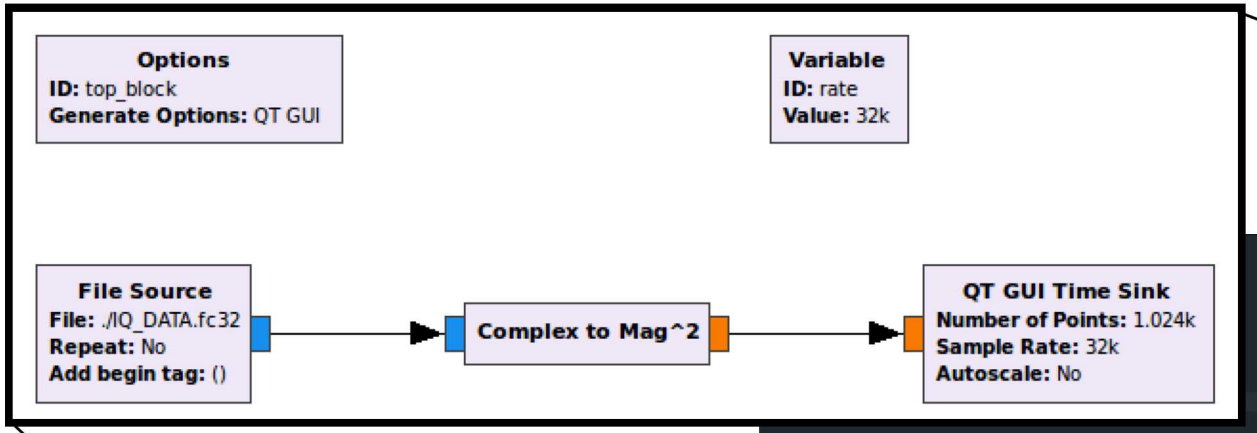
Multiple inputs to block



N:1 input connections

Loops / Feedback

GNU Radio Basic Theory of Operation

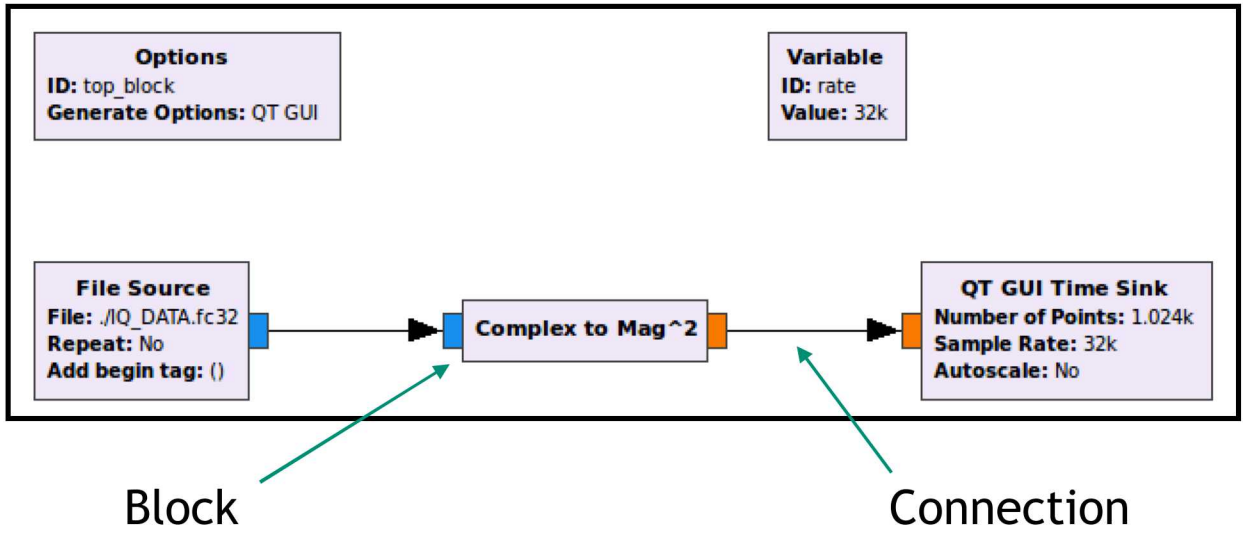



```
class top_block(gr.top_block):

    def __init__(self):
        gr.top_block.__init__(self, "Top Block")

        # Variables
        self.rate = rate = 32000

        # Blocks
        self.blocks_file_source_0_0_0_0 = blocks.file_source(gr.sizeof_gr_complex*1, './IQ_DATA.fc32', False)
        self.blocks_file_source_0_0_0_0.set_begin_tag(pmt.PMT_NIL)
        self.blocks_file_sink_0 = blocks.file_sink(gr.sizeof_float*1, 'rate', False)
        self.blocks_file_sink_0.set_unbuffered(False)
        self.blocks_complex_to_mag_squared_0 = blocks.complex_to_mag_squared(1)

        # Connections
        self.connect((self.blocks_complex_to_mag_squared_0, 0), (self.blocks_file_sink_0, 0))
        self.connect((self.blocks_file_source_0_0_0_0, 0), (self.blocks_complex_to_mag_squared_0, 0))


def main(top_block_cls=top_block, options=None):

    tb = top_block_cls()
    tb.start()
    try:
        raw_input('Press Enter to quit: ')
    except EOFError:
        pass
    tb.stop()
    tb.wait()


if __name__ == '__main__':
    main()
```

GR Flowgraphs:

- Top level GR object (gr::top_block)
- Defined in Python or C++
  - Language not performance-critical
- Basic GR Executable

GNU Radio Basic Theory of Operation



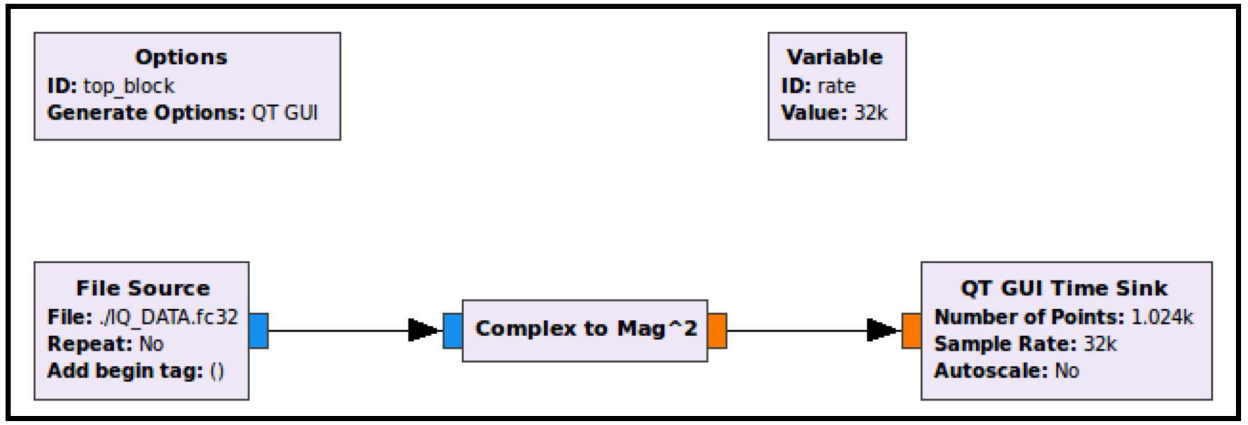
GR Blocks:

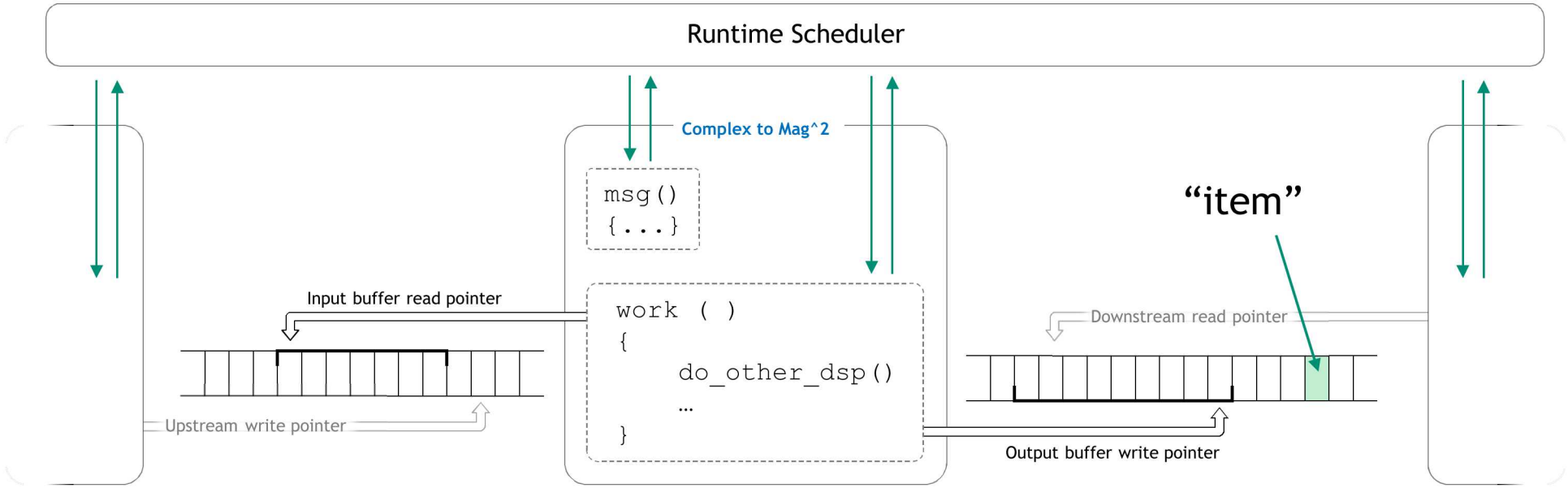
- Fundamental GR signal processing unit
- Written in C++ (preferred, fast) or python
- 1:1, 1:N, N:1, arbitrary I/O relationships
- Reads items, does processing, writes items

GR Connections:

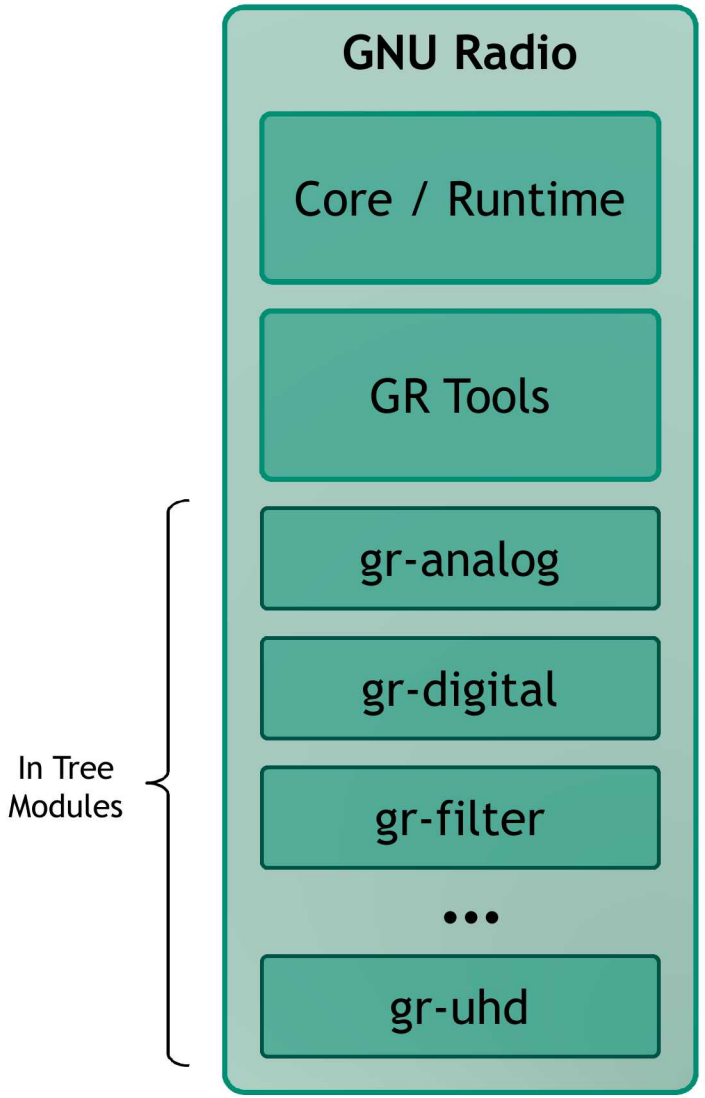
- Buffers which allow for passing items between blocks
- Very efficient shared memory gr::buffer implementation
- One write pointer, one or more read pointers

GNU Radio Basic Theory of Operation

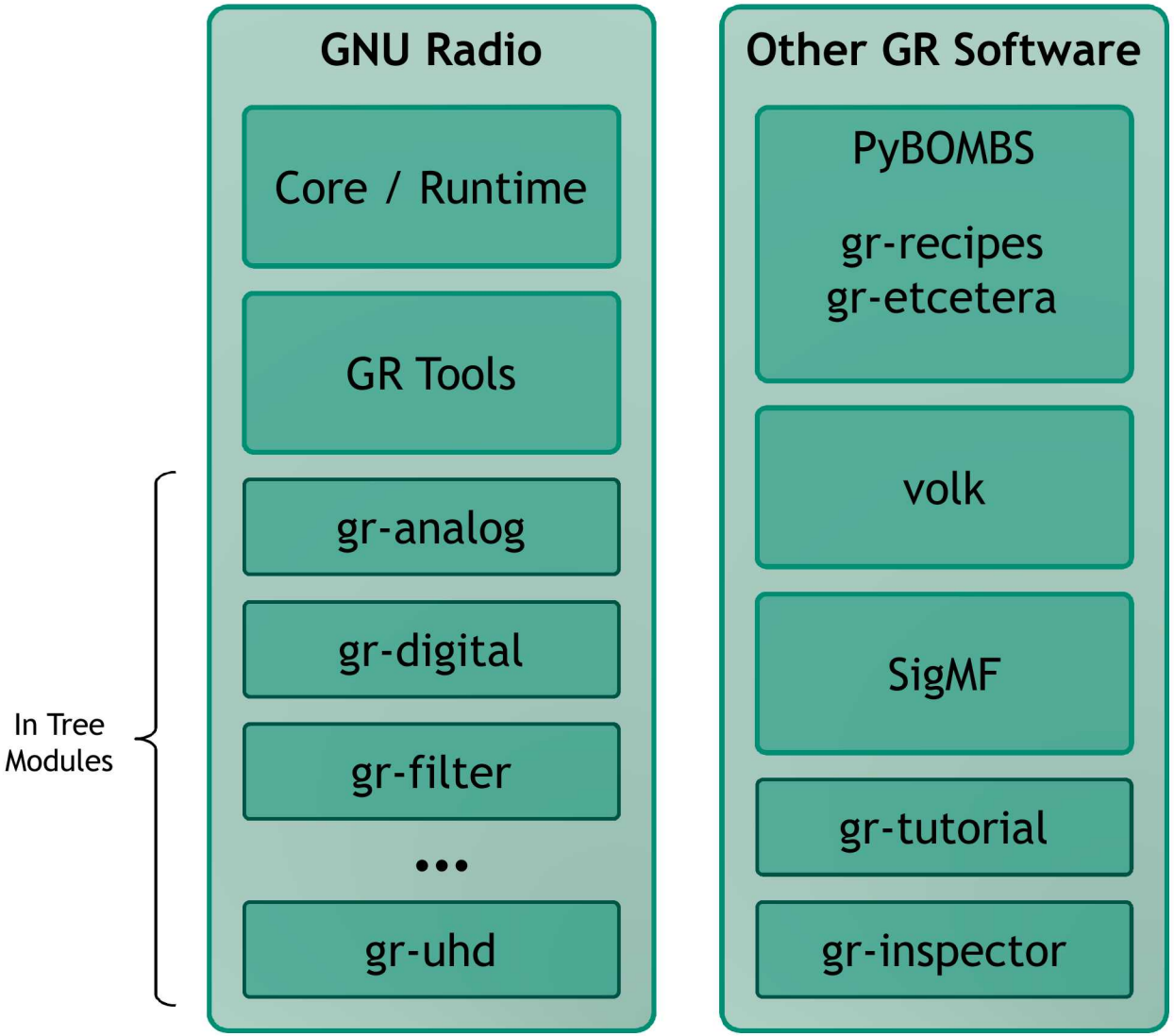




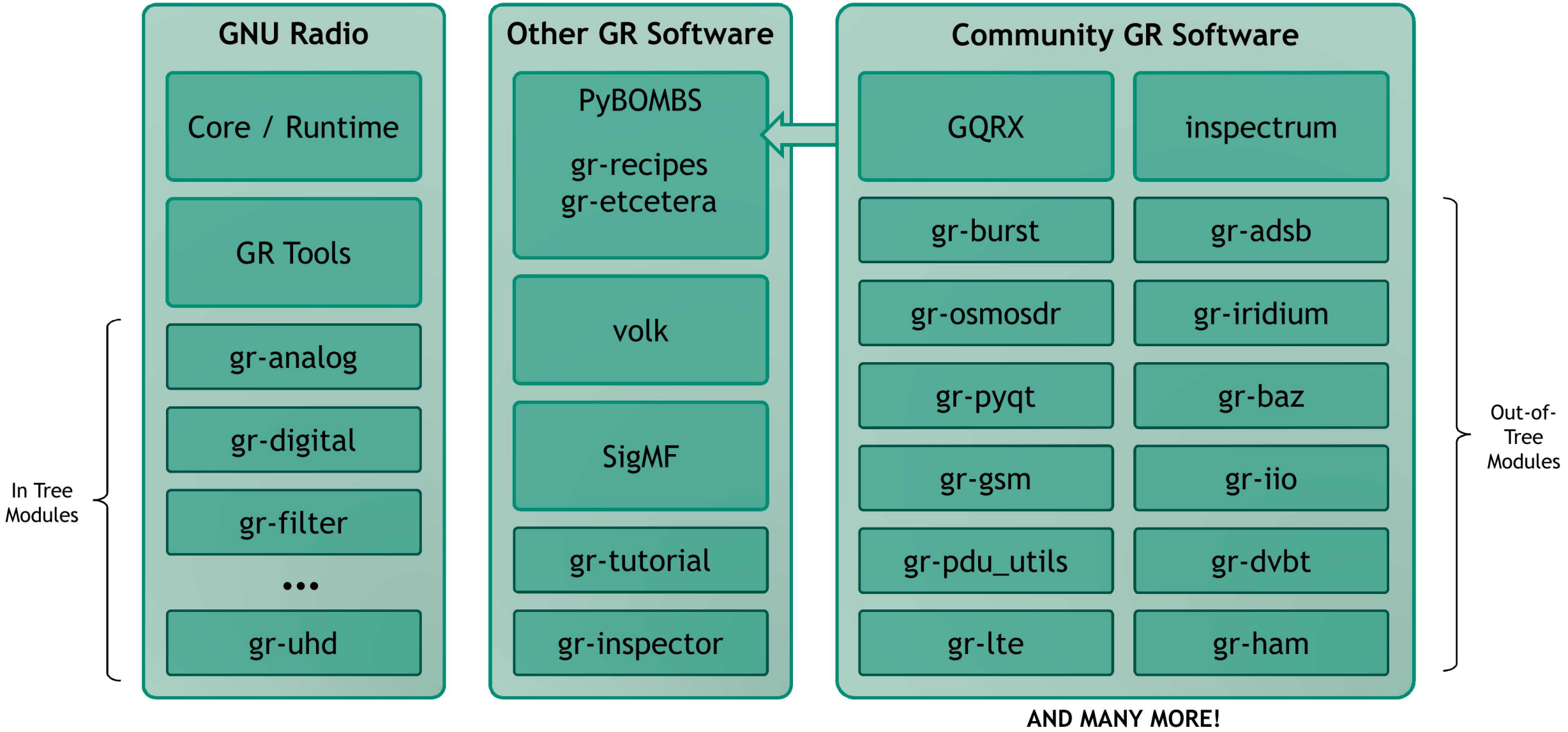
GNU Radio Project Software Organization



GNU Radio Project Software Organization



GNU Radio Project Software Organization



GNU Radio Extensibility – *gr_modtool* and *PyBOMBS*

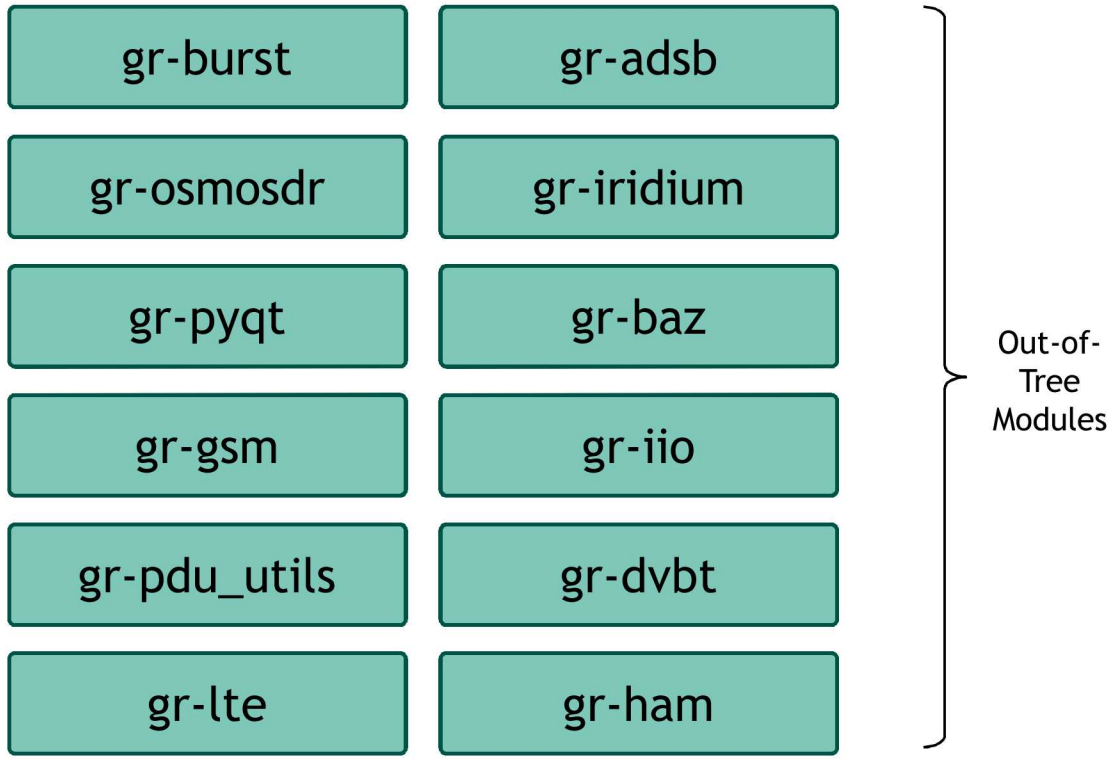
gr_modtool: Built in tools for creating and sharing additional GNU Radio capabilities

- Creates new modules including necessary build structure
- Tools for adding/removing signal processing blocks

PyBOMBS: build system for GR

- Simple management of GNU Radio related software
- Share *recipes* for OOT modules
- Private or public recipe repositories

https://github.com/gnuradio/gr-recipes
https://github.com/gnuradio/gr-etcetera



Using GNU Radio

1. As an Application: Using GNU Radio
   - No programming required (but it helps)
   - Lots of tutorials and resources available
   - Limited to tools that currently exist
   - Windows, Linux, OSX software available
2. As a Framework: Extending GR capability
   - Requires C++ and/or Python
     - Familiarity with Linux, git, and cmake will help
   - Many existing examples to work from
   - Allows users to build anything GR can do
3. As an Obsession: Developing GNU Radio
   - Extending internal features, APIs, scheduler, etc
   - Expertise required (talk to the GR community!)

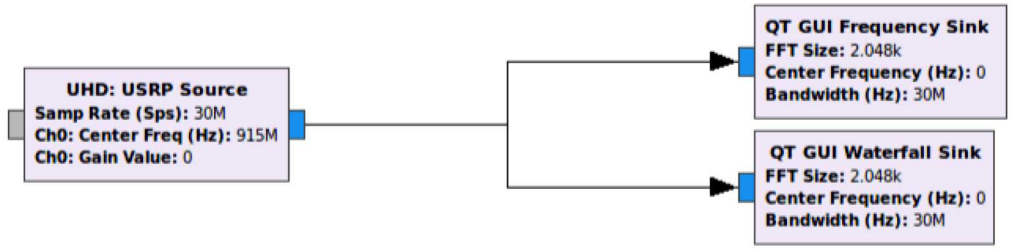
Majority of users start here…

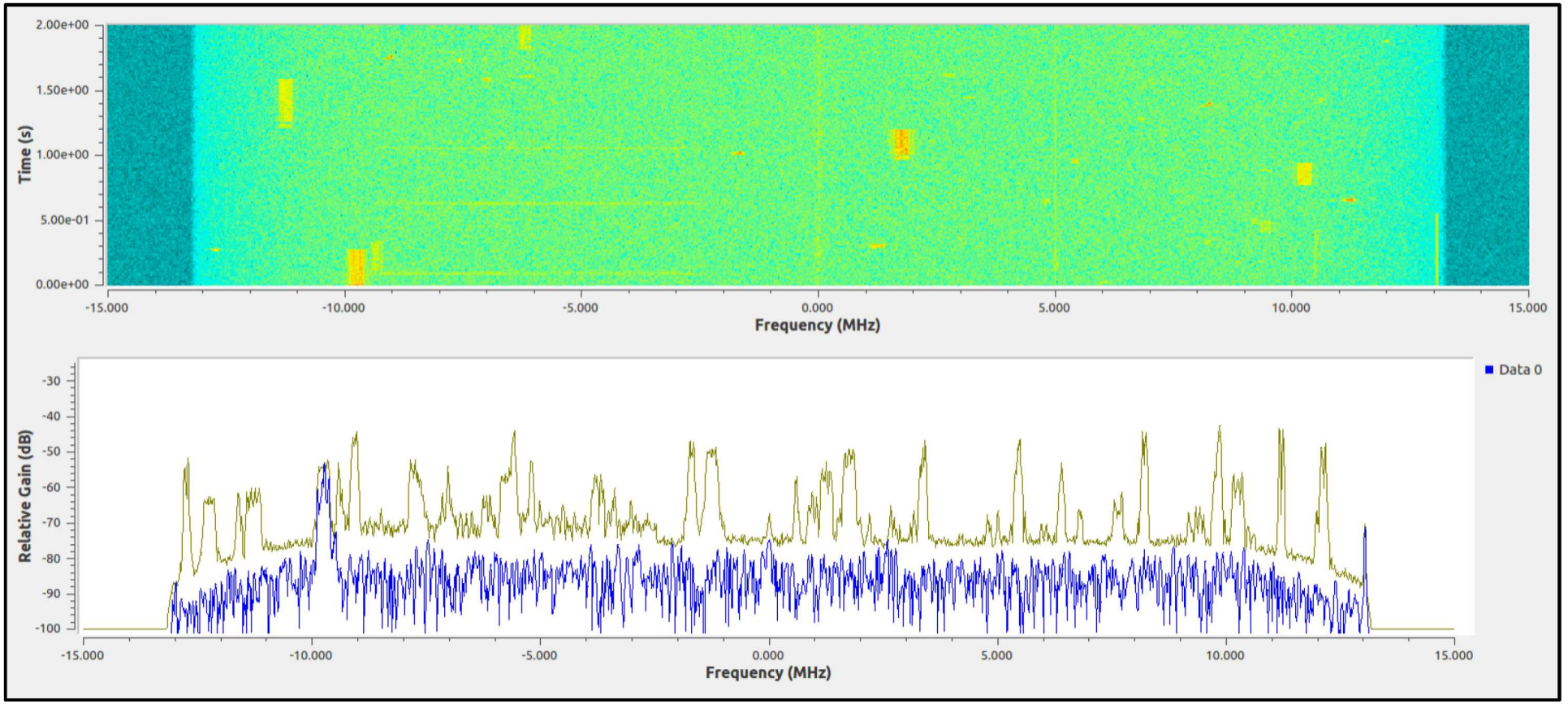
and move here as necessary

Not many people work down here…

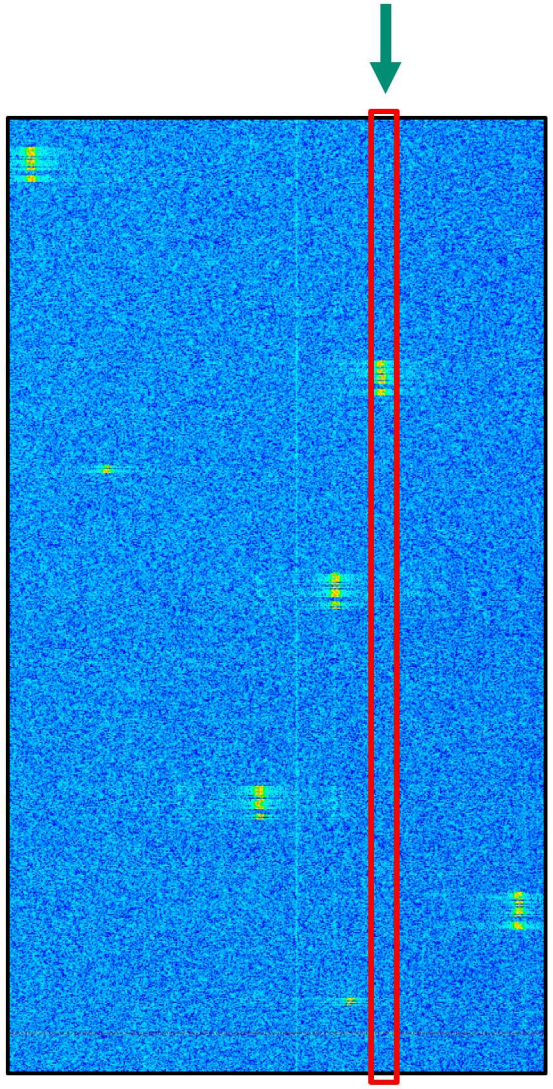
GNU Radio Applications

Basic Spectral Analysis



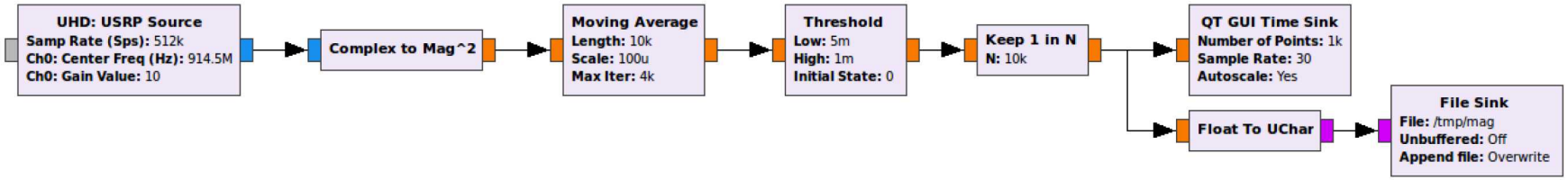


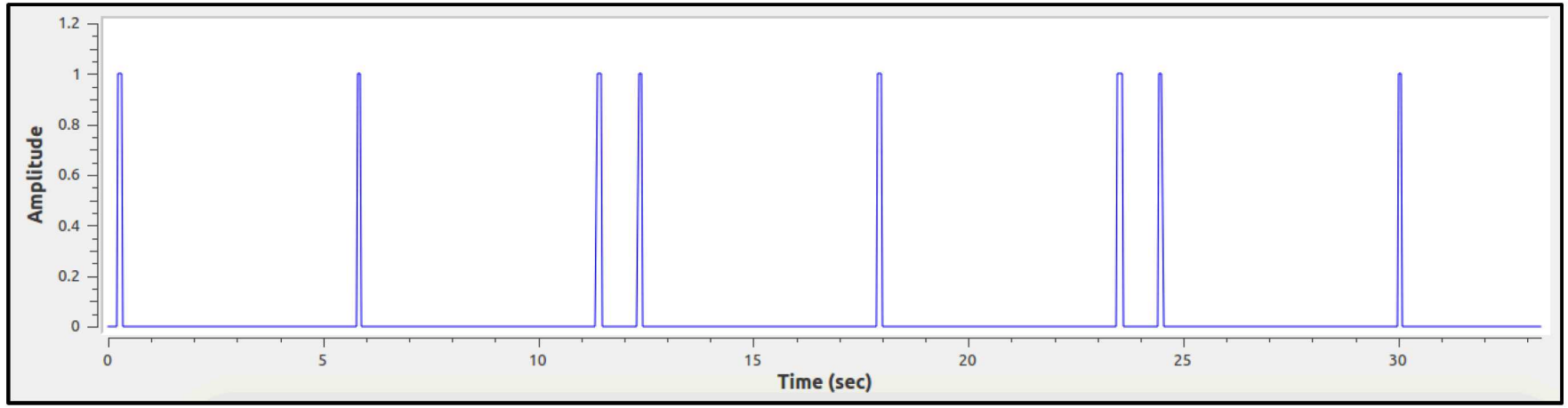
Spectral Analysis – Putting GR to Work



Simple example of how GR can enhance Spectral Analysis:

- Given a FHSS Radio, log when bursts occur on a fixed channel
  - With bonus real-time viewer



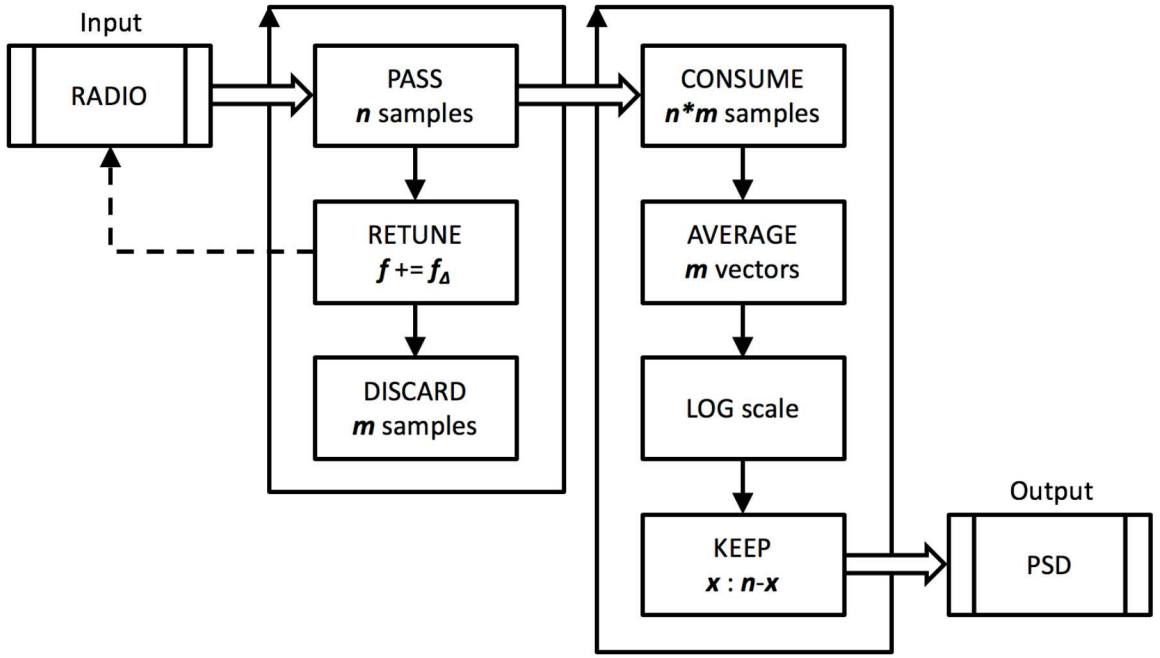


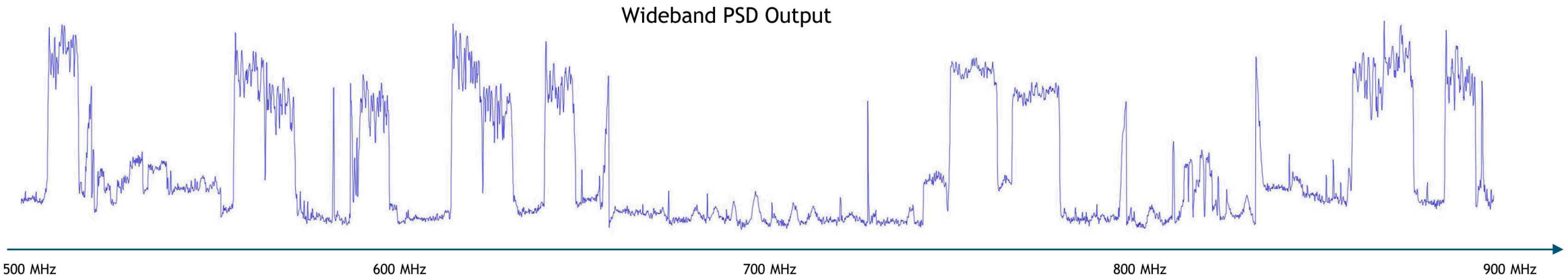
Wideband Spectral Analysis

SDR only has 30 MHz instantaneous BW?

PSD tune / average / stitch flowgraph for ultra wideband spectral monitoring

Doesn't have to be spectrally continuous

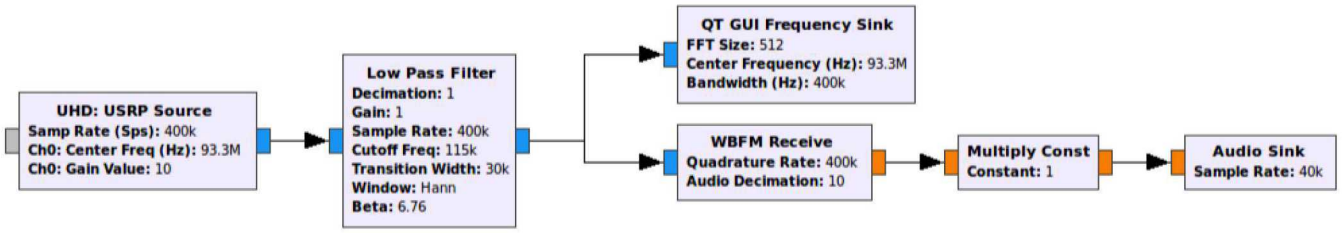


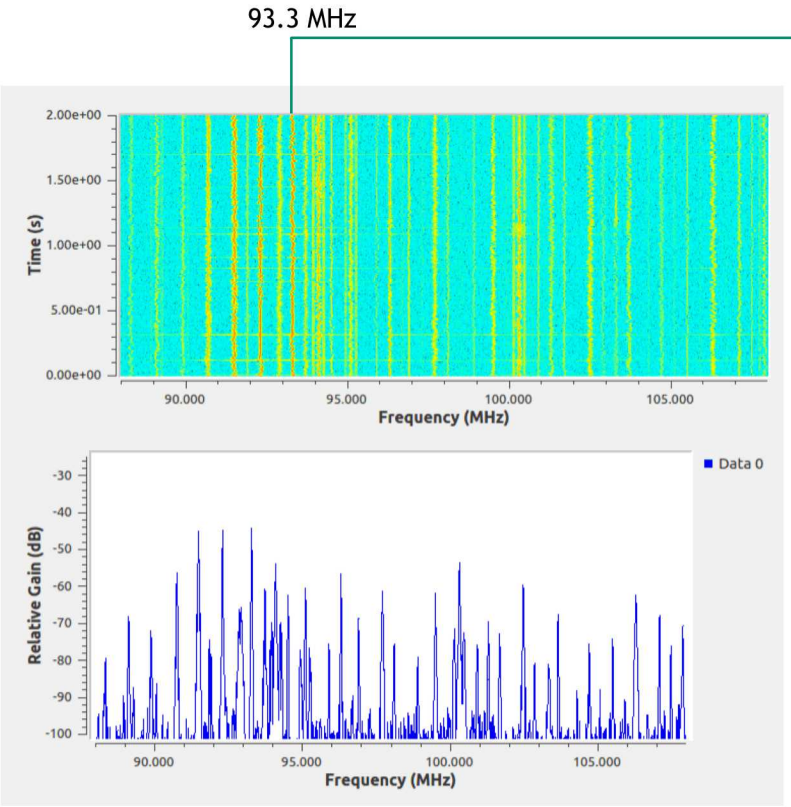


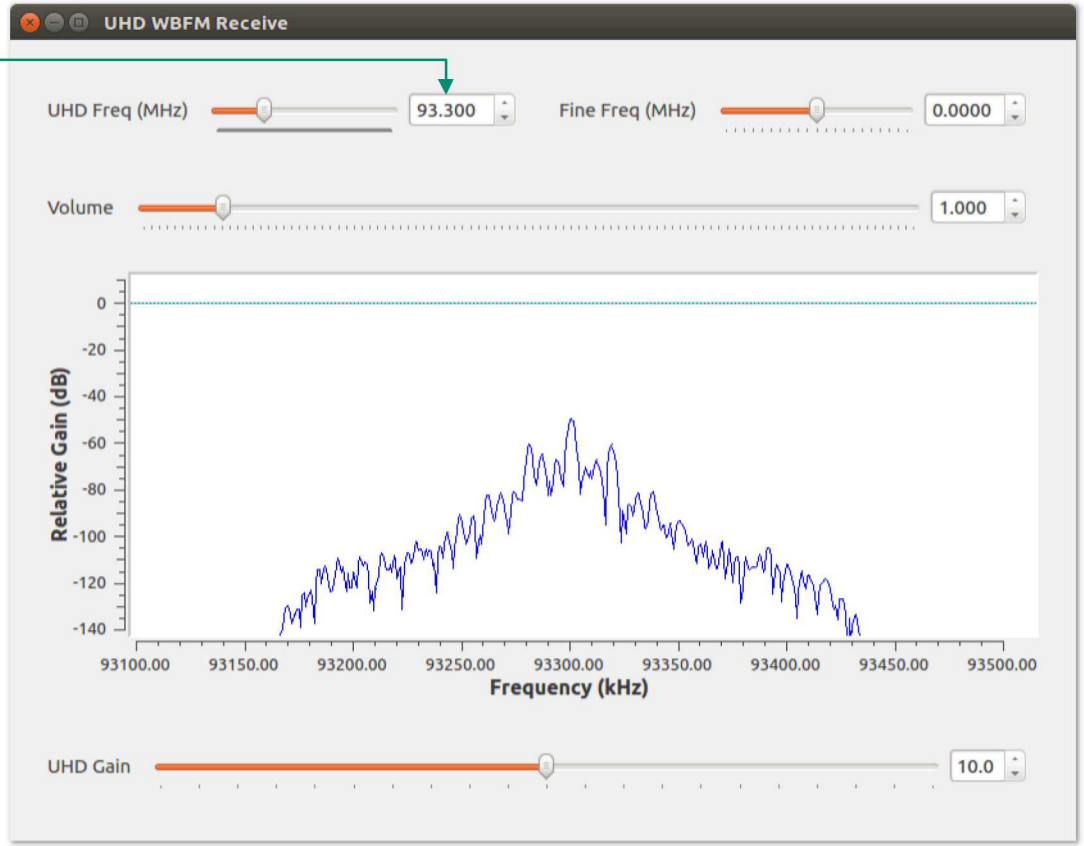
Analog FM Radio Reception

Example app included with GR

uhd_wbfm_rx.grc

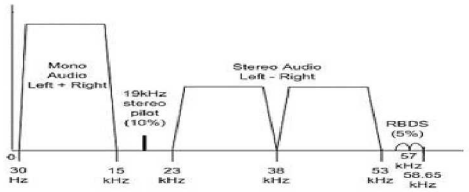


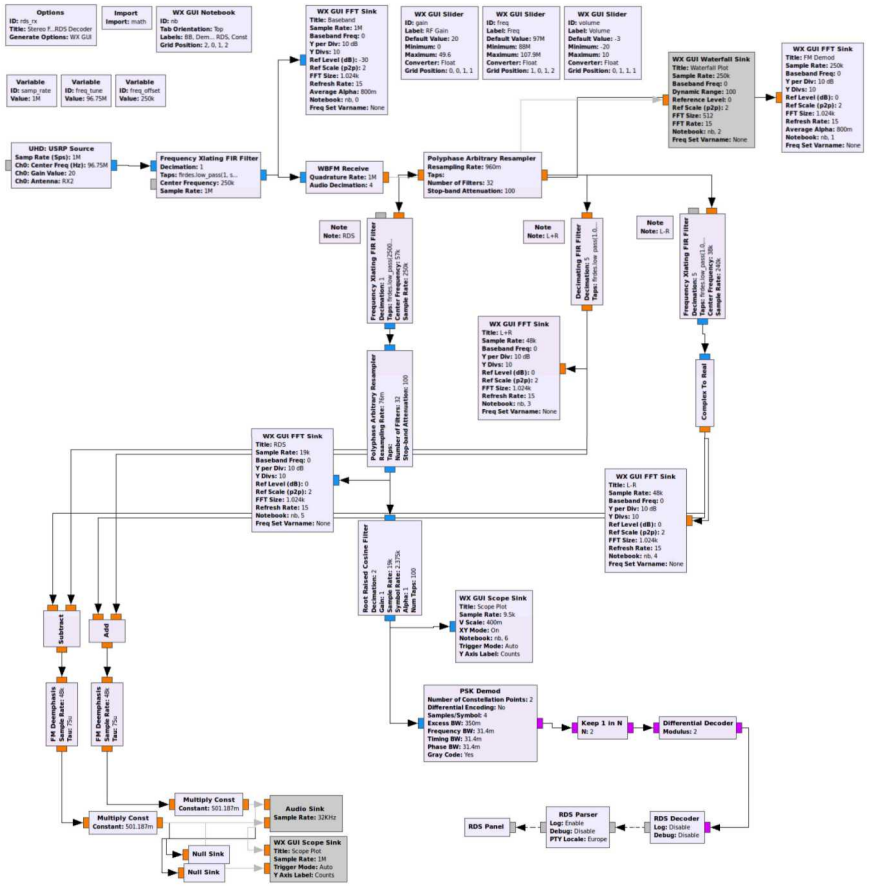




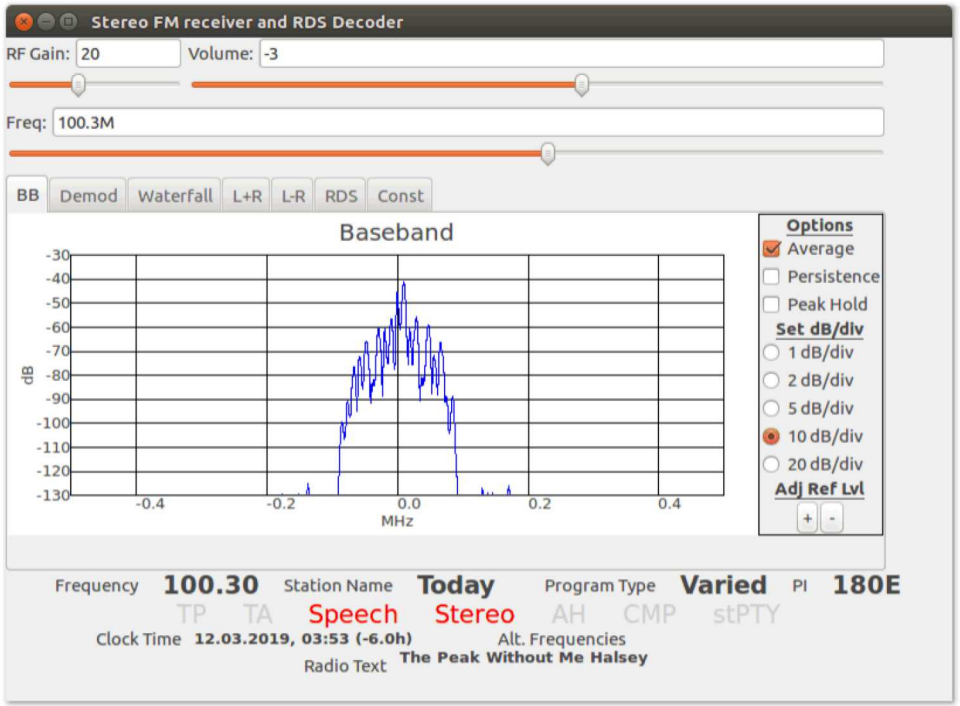
FM Radio Reception – Radio Data Service (RDS)

Low rate digital data passed in-band with broadcast FM signals





```
event137:traffic lighter than normal (with average speeds Q), location:15360
event701:(Q sets of) roadworks, location:37740
event736:(Q sets of) roadworks. Right lane closed, location:37504
event735:closed due to (Q sets of) roadworks, location:37964
event115:slow traffic (with average speeds Q), location:24144
event803:(Q sets of) construction work, location:19300
event701:(Q sets of) roadworks, location:37740
```



OOT Module: https://github.com/bastibl/gr-rds

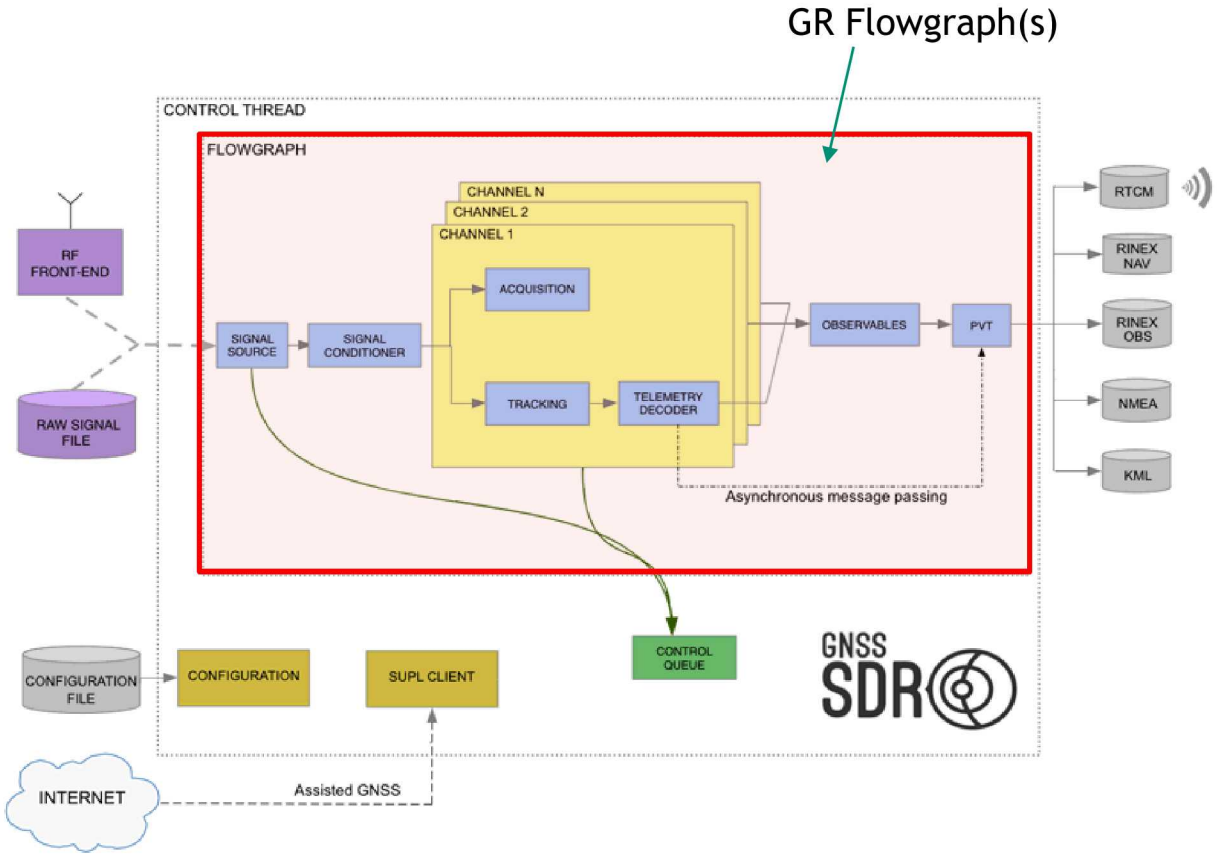
Global Navigation Satellite System

Open source multi-constellation GNSS receiver with DSP implemented using GNU Radio

Signal		Frequency	ID
GPS	L1 C/A	1575.42 MHz	1C
GPS	L2CM	1227.60 MHz	2S
GPS	L5	1176.45 MHz	L5
Galileo	E1 B/C	1575.42 MHz	1B
Galileo	E5a	1176.45 MHz	5X
Glonass	L1 C/A	1602.00 MHz	1G
Glonass	L2 C/A	1246.00 MHz	2G
Beidou	B1I	1561.098 MHz	B1

GNSS-SDR Supported Signals

https://github.com/gnss-sdr/gnss-sdr

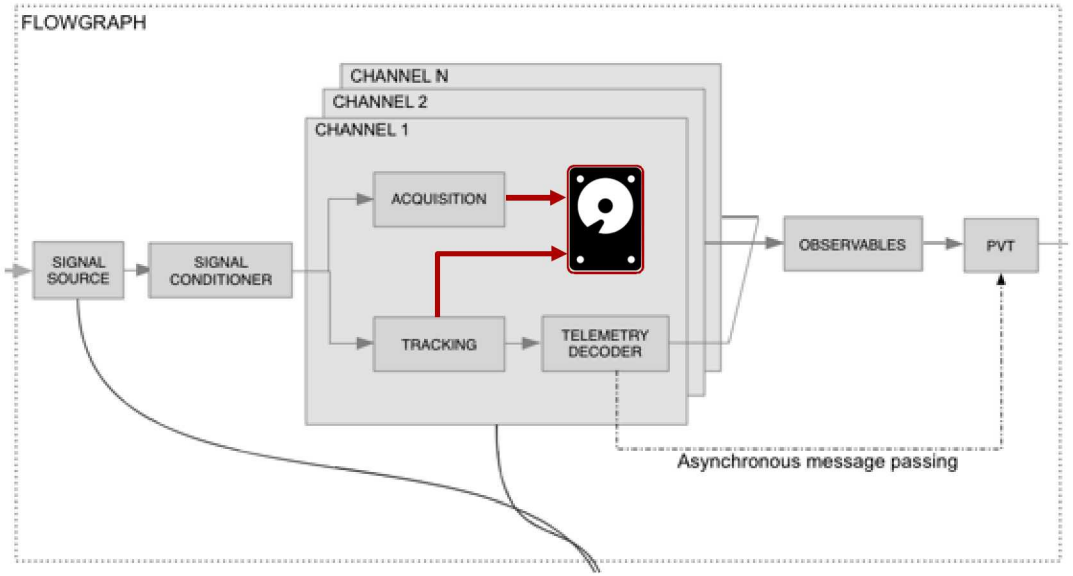


GNSS-SDR Architecture

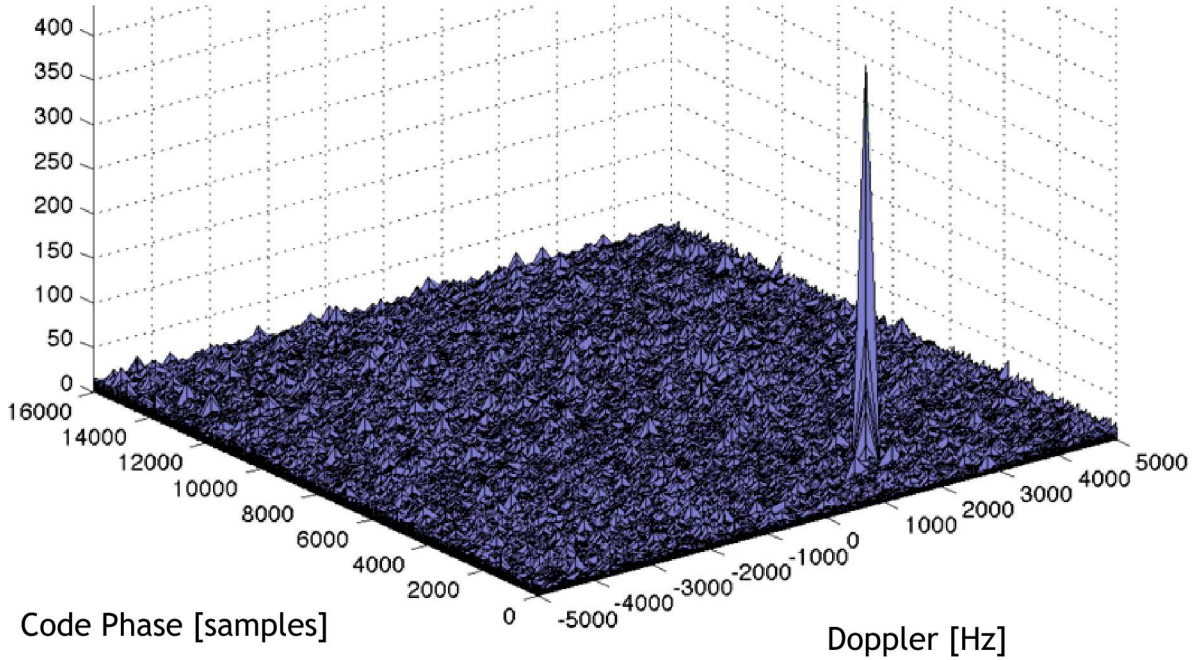
Advantages SDR Provides to GNSS Receivers

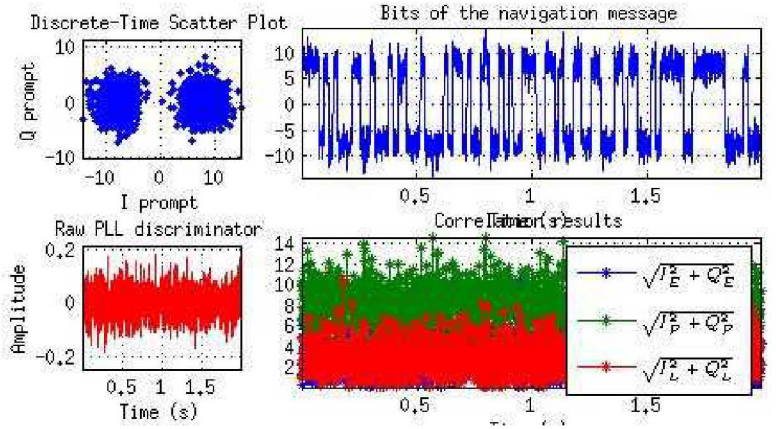
Directly observe/log internal system states

Substantial insight into internal GNSS operation



Acquisition Stage Code / Doppler Plot



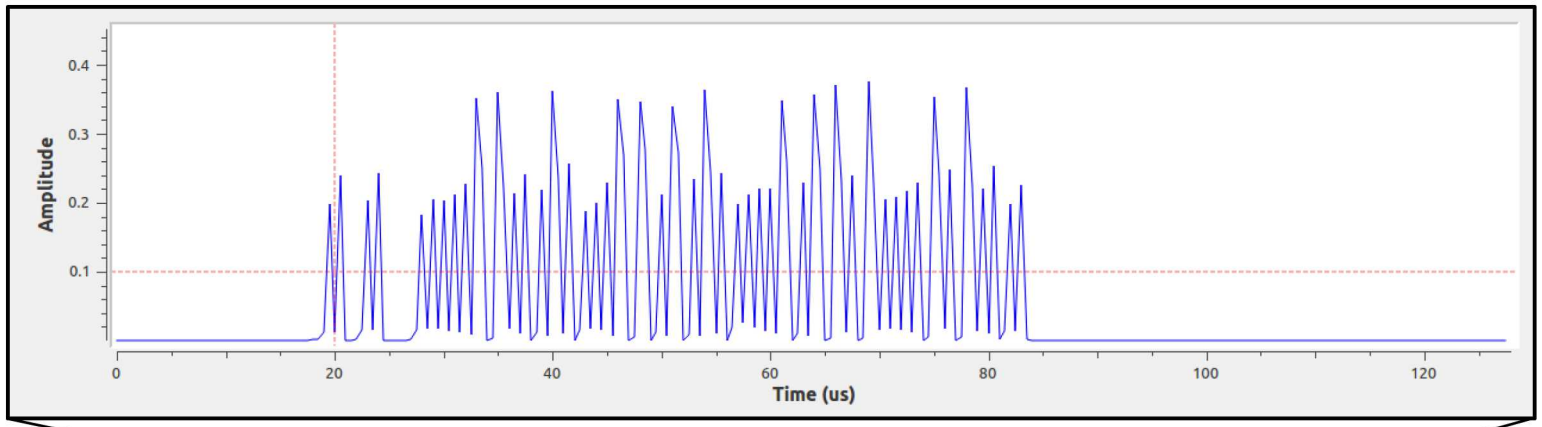


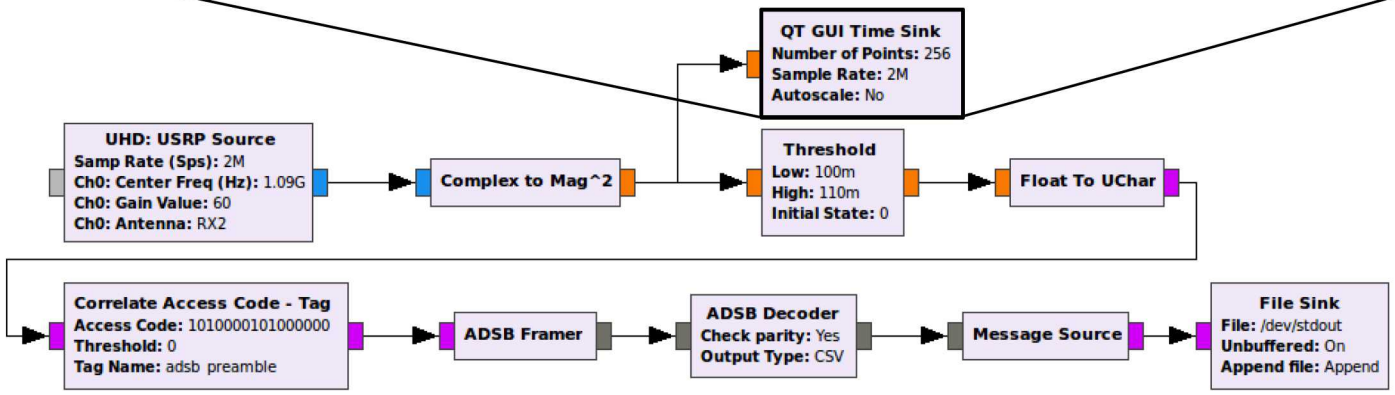
Internal GNSS Tracking Data

ADS-B Aircraft Telemetry

Automatic Dependent Surveillance **Broadcast**

gr-adsb: Simple PPM demodulator and ADS-B deframer



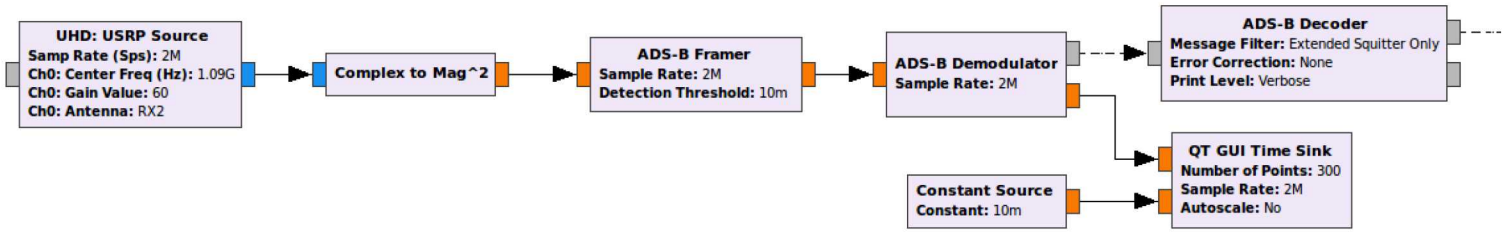


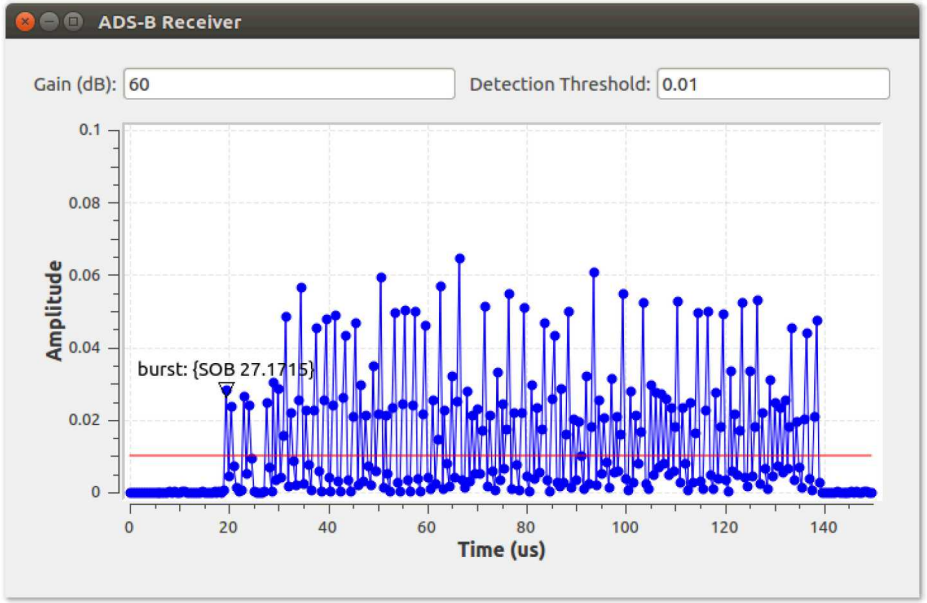

```
A21839,,409,285,,,17,5,19
A21839,,,,,,11,5,23
A4455E,,368,246,,,17,5,19
ABC3C6,,,,,,17,5,29
A4455E,,,,35.79,-106.91,1,17,5,11
A4455E,,368,246,,,17,5,19
A21839,,,,35.01,-99.16,1,17,5,11
A4455E,,,,,,11,5,22
A4455E,,,,35.79,-106.92,0,17,5,11
A21839,,409,285,,,17,5,19
A4455E,,367,246,,,17,5,19
A4455E,,,,,,11,5,22
ABC3C6,,548,72,,,17,5,19
ABC3C6,,,,36.07,-106.60,0,17,5,11
ABC3C6,,,,,,17,5,29
A4455E,,,,,,11,5,22
ABC3C6,,548,72,,,17,5,19
A4455E,,,,,,17,5,29
A4455E,,,,35.79,-106.92,0,17,5,11
A4455E,,367,246,,,17,5,19
ABC3C6,,548,72,,,17,5,19
A4455E,,,,35.79,-106.92,1,17,5,11
A4455E,,367,246,,,17,5,19
A4455E,,367,246,,,17,5,19
ABC3C6,,,,,,17,5,29
A21839,,409,285,,,17,5,19
ABC3C6,,,,36.08,-106.58,0,17,5,11
A21839,,,,,,17,5,31
ABC3C6,,,,36.08,-106.58,0,17,5,11
ABC3C6,,548,72,,,17,5,19
A4455E,,,,35.78,-106.93,0,17,5,11
```

OOT Module: https://github.com/wnagele/gr-adsb

Another look at ADS-B Aircraft Telemetry

A different way of approaching ADS-B

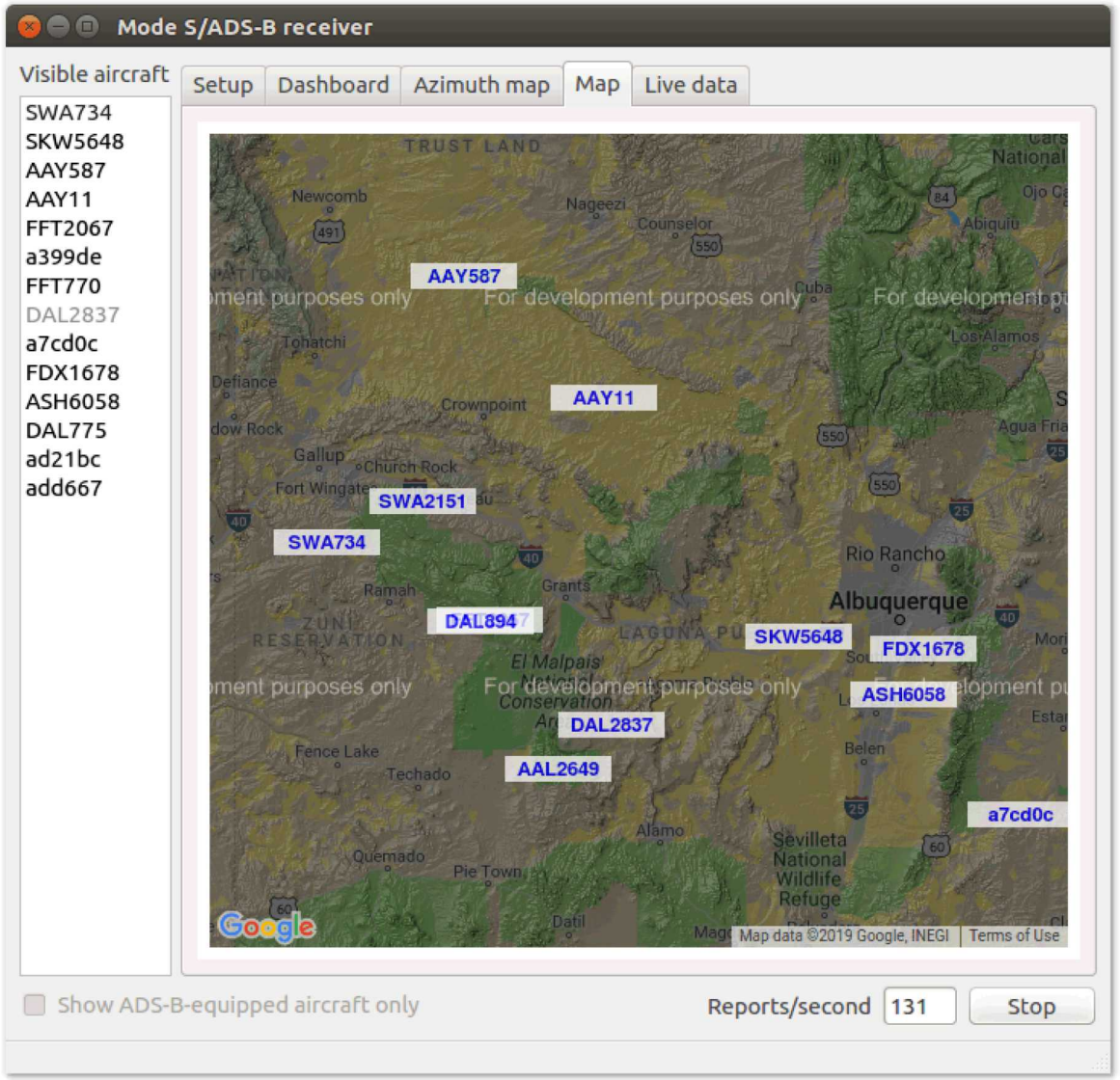




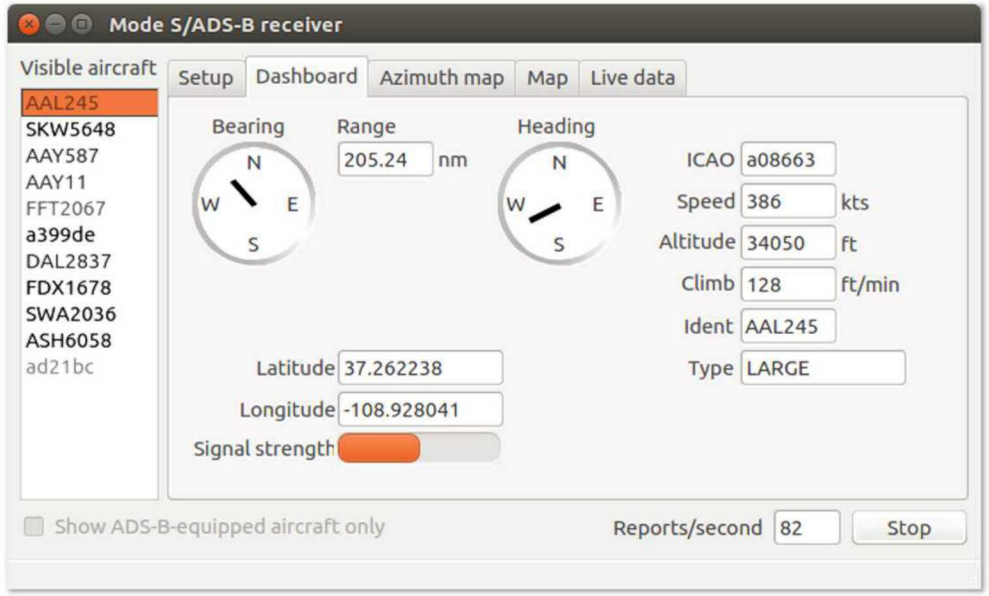

```
-------------------------------------------
SNR:            28.85 dB
DF:             17 Extended Squitter
CRC:            Passed
CA:             5 Level 2 or Above Transponder
AA:             ade543
TC:             19 Airborne Velocity
ST:             1 Ground Velocity
IC:             0 No Change in Intent
Speed:          404 kt
Heading:        158 deg (W)
Climb:          64 ft/min
Climb Source:   0 Geometric Source (GNSS or INS)
-------------------------------------------
SNR:            25.70 dB
DF:             17 Extended Squitter
CRC:            Passed
CA:             5 Level 2 or Above Transponder
AA:             a8913b
TC:             29 Reserved
-------------------------------------------
SNR:            24.20 dB
DF:             17 Extended Squitter
CRC:            Passed
CA:             5 Level 2 or Above Transponder
AA:             ade543
TC:             19 Airborne Velocity
ST:             1 Ground Velocity
IC:             0 No Change in Intent
Speed:          404 kt
Heading:        158 deg (W)
Climb:          64 ft/min
Climb Source:   0 Geometric Source (GNSS or INS)
```

OOT Module: https://github.com/mhostetter/gr-adsb

ADS-B Tool w/ User Interface




```
Type 17 from a7cd0c with velocity 442kt heading 306 VS 64
Type 17 from aa7b33 with velocity 342kt heading 263 VS -64
Type 11 from a39d95 in reply to interrogator 0 with capability level 6
Type 11 from ad07a5 in reply to interrogator 0 with capability level 6
Type 17 from ad21bc at (35.817012, -107.597351) (82.33 @ 311) at 41000ft
Type 0  from aa7b33 at 40000ft (speed 300-600kt)
Type 11 from ad21bc in reply to interrogator 0 with capability level 6
Type 17 from aa7b33 at (35.525711, -106.639824) (35.08 @ 347) at 40000ft
Type 17 from ad07a5 at (36.835785, -107.250710) (131.43 @ 341) at 38000ft
Type 17 from a7cd0c at (34.988434, -106.720911) (12.85 @ 257) at 32000ft
```

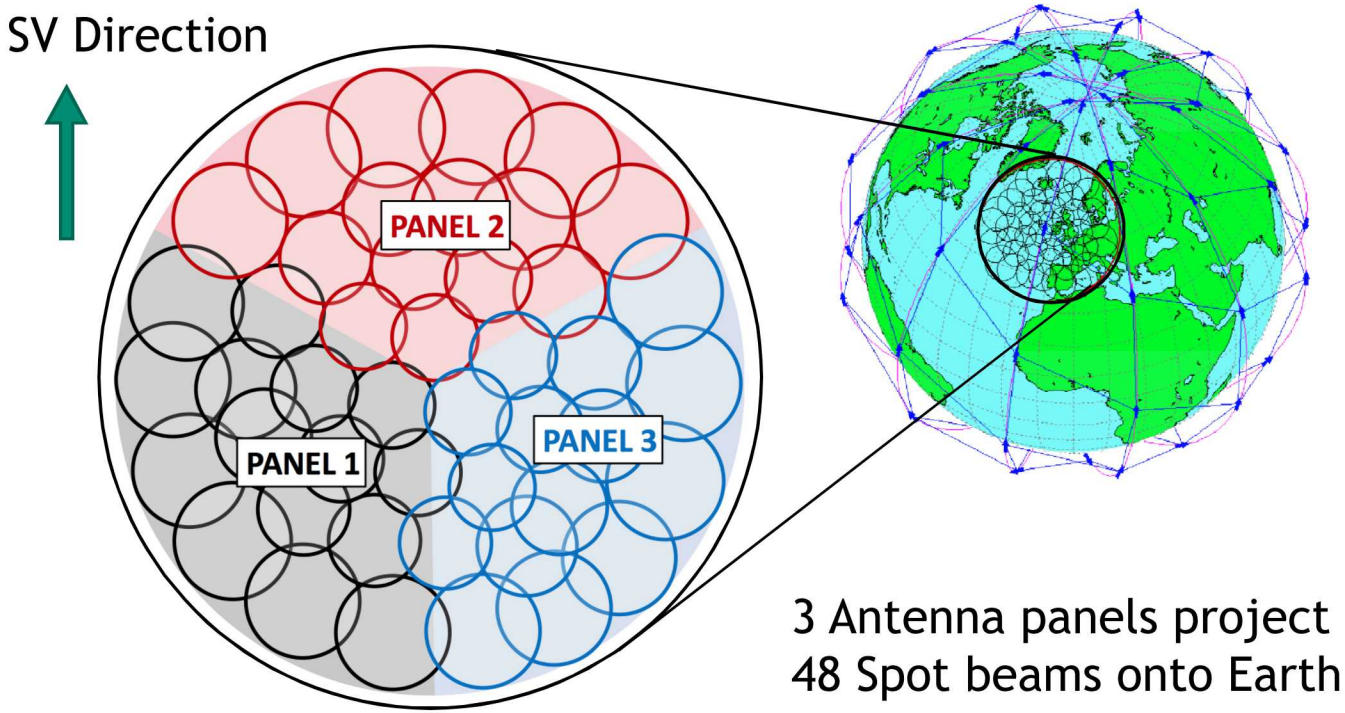


OOT Module: https://github.com/bistromath/gr-air-modes

Observing the Iridium Constellation

Two open source tools for receiving iridium data published by the Chaos Computer Club

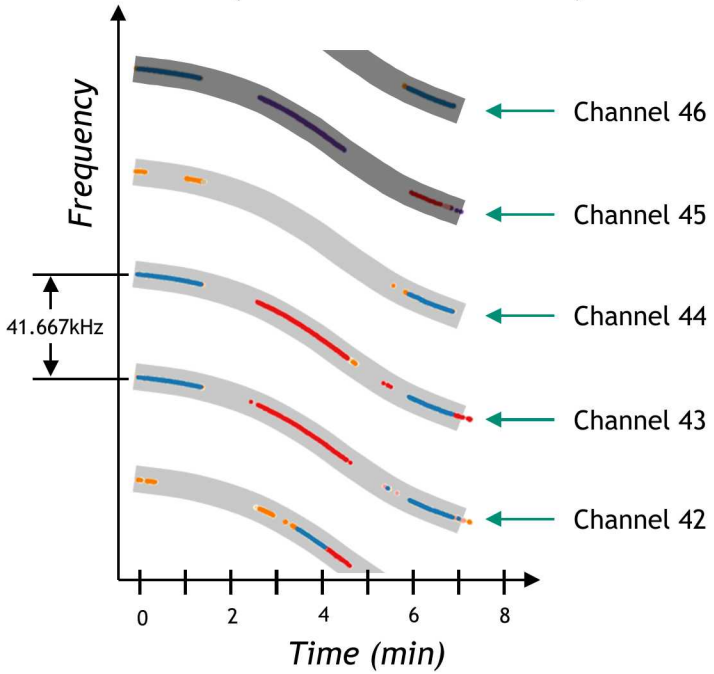
- Constellation of ~66 active LEO Satellites in 6 polar orbits
- Provides voice and data support with true global coverage



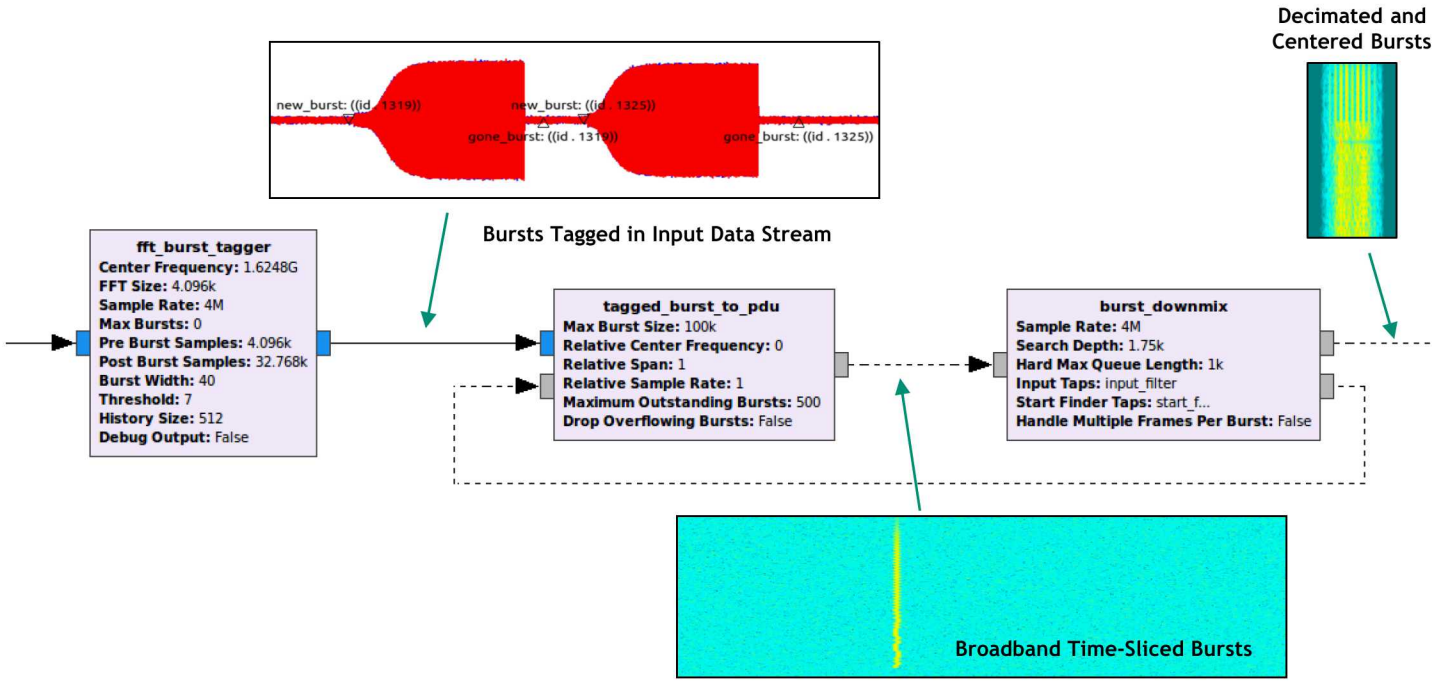
3 Antenna panels project
48 Spot beams onto Earth

Challenges:

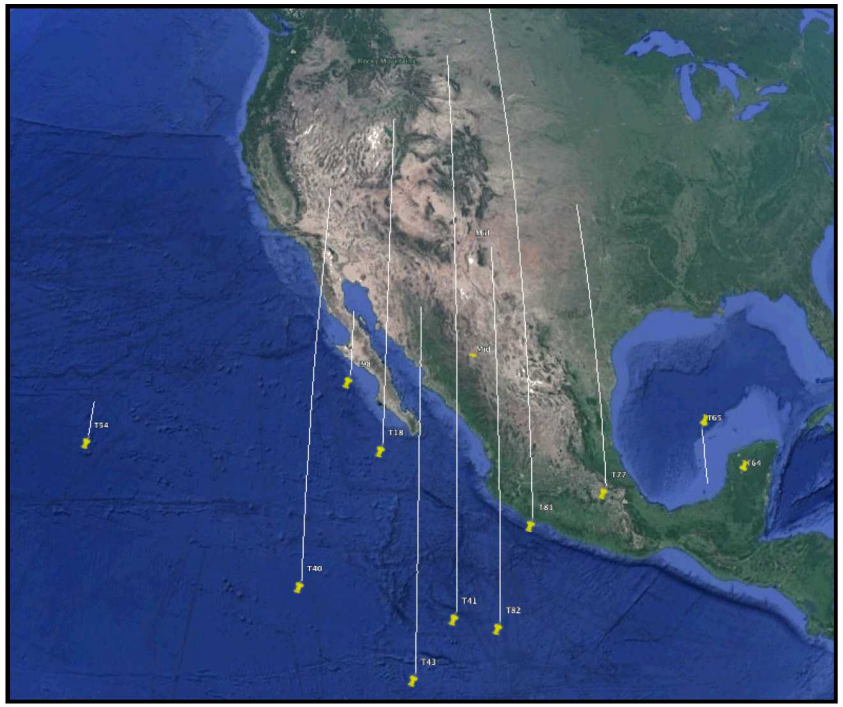
- Protocol not public knowledge
- Burst frequency arbitrary
  - (1616-1626.5 MHz)



Iridium Data Receiver



Iridium Satellite Ground Tracks



FFT based burst detection/frequency localization

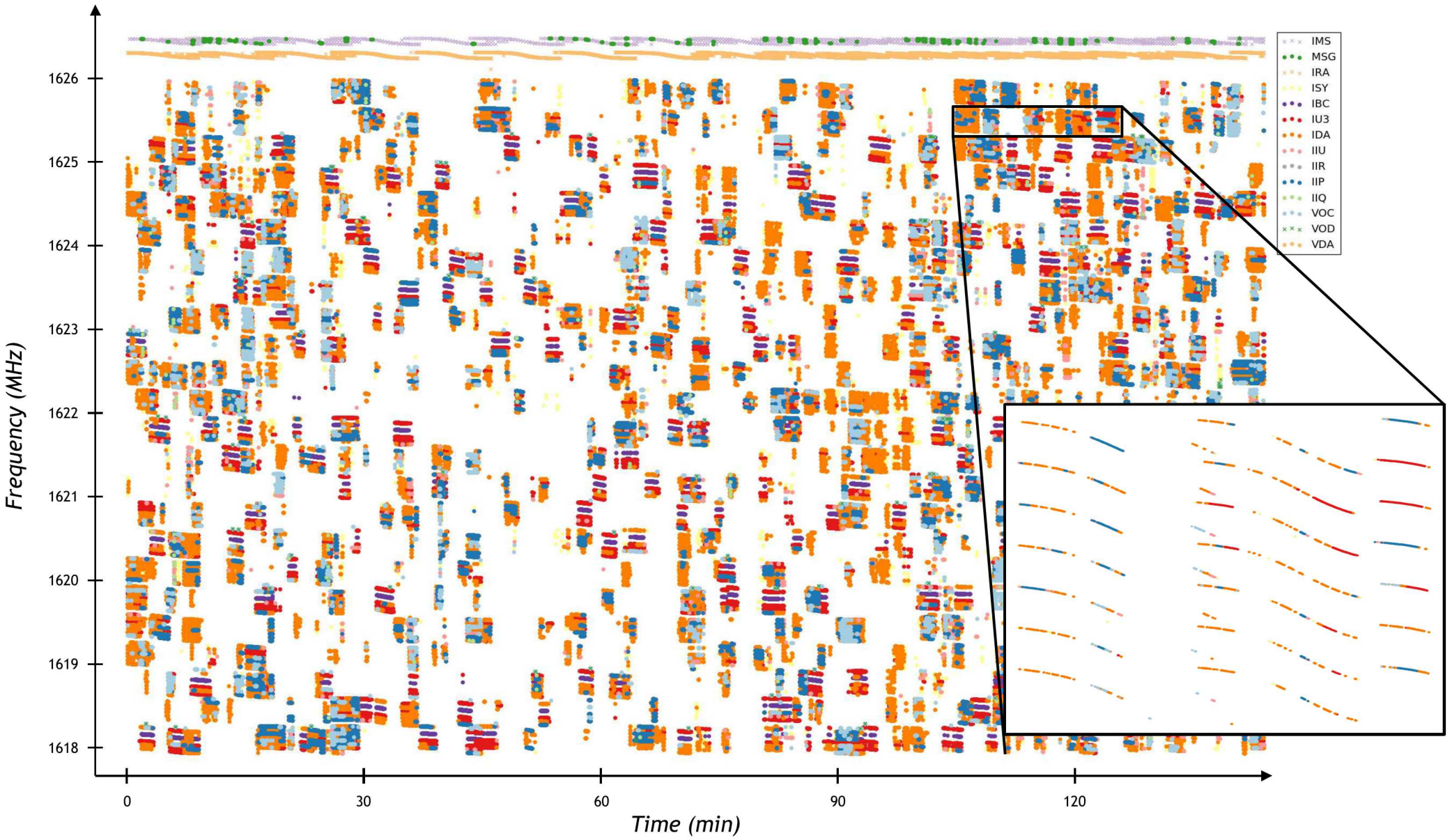
Internal structure of bursts reverse engineered to provide

- Satellite tracks and paging messages (simplex channels)
- Message types and some metadata (data protected?)
- Much more – look at the toolkit for more information!

OOT Module: https://github.com/muccc/gr-iridium

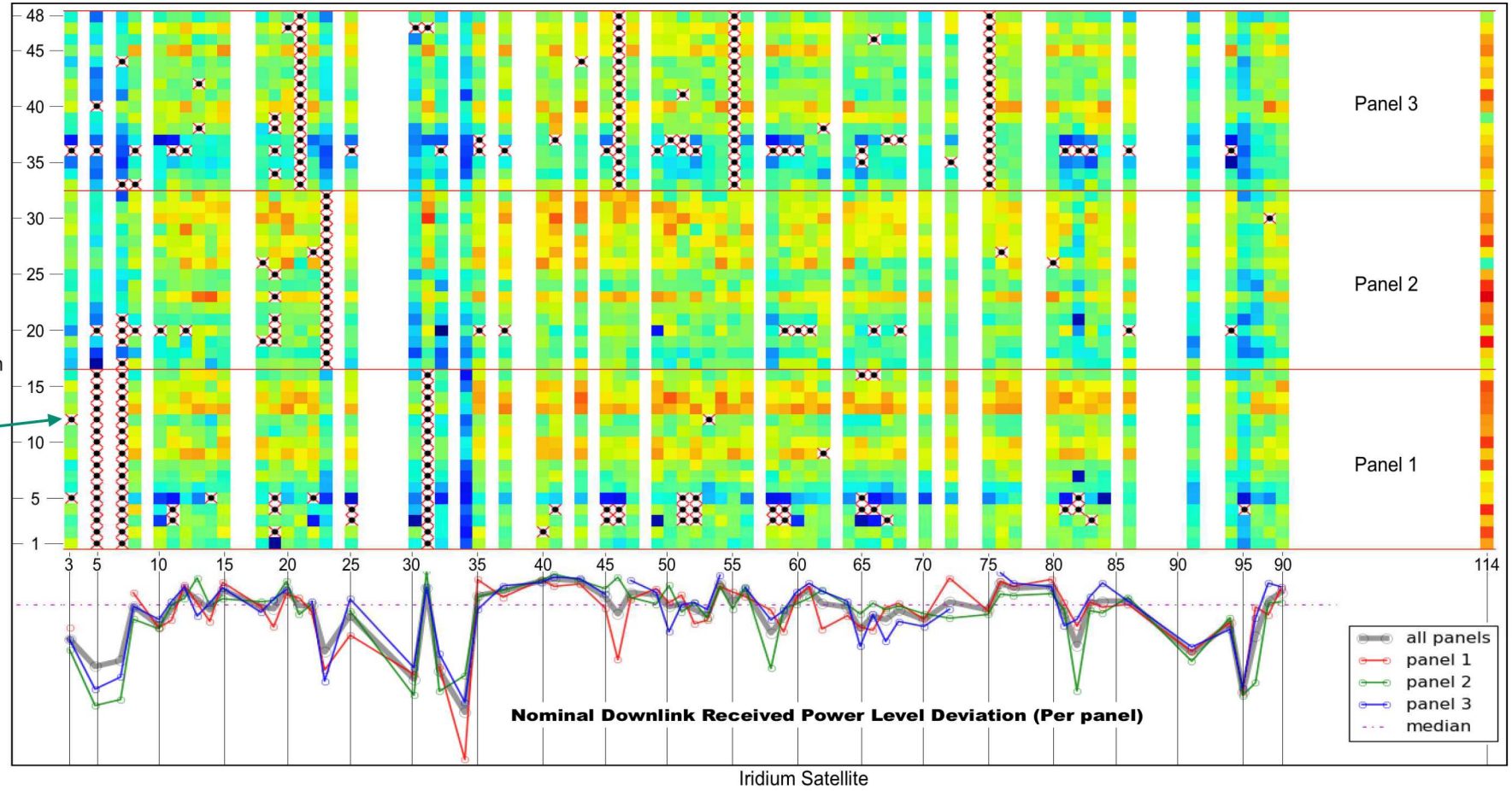
OOT Module: https://github.com/muccc/iridium-toolkit

Two Hours of Iridium Bursts



Iridium SV RF Performance Characterization

No bursts received from SV / Beam



2 weeks of collected data – peak received level per spot beam shown (2016)

Summary

Takeaways

GNU Radio is a highly capable tool for providing access to the RF Spectrum

GNU Radio has a low barrier to entry for developing bespoke SDR tools and systems

There is a massive amount of of high quality software freely available for the GR ecosystem

Go try it out!

(and when you do, share what you build when possible)