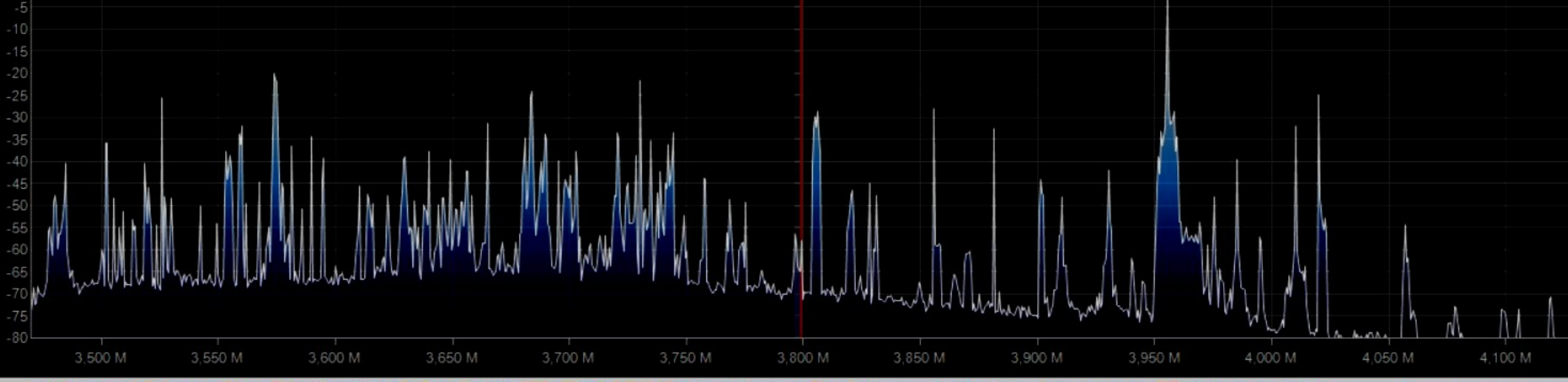


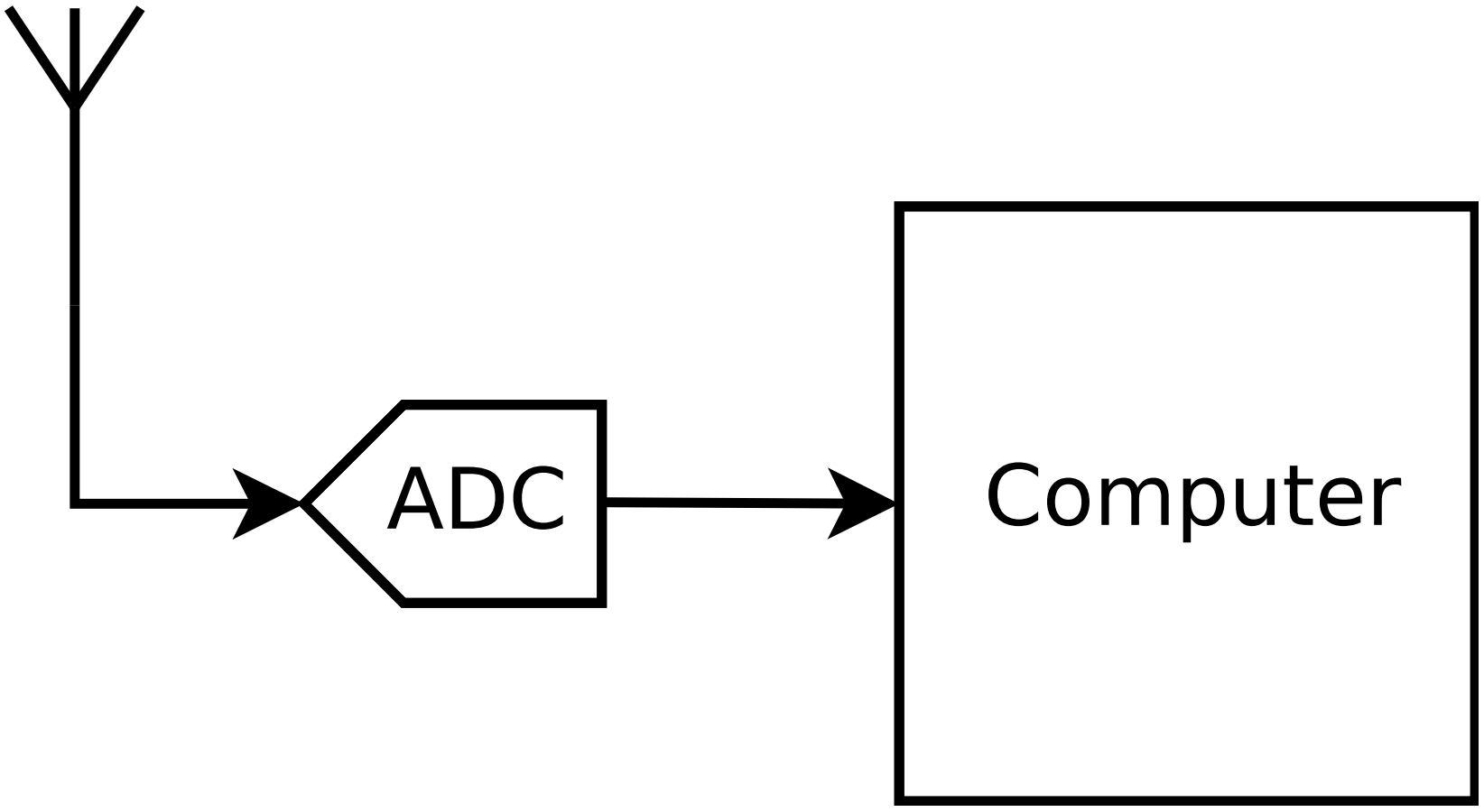

# What's New in Software-Defined Radio?

# Agenda

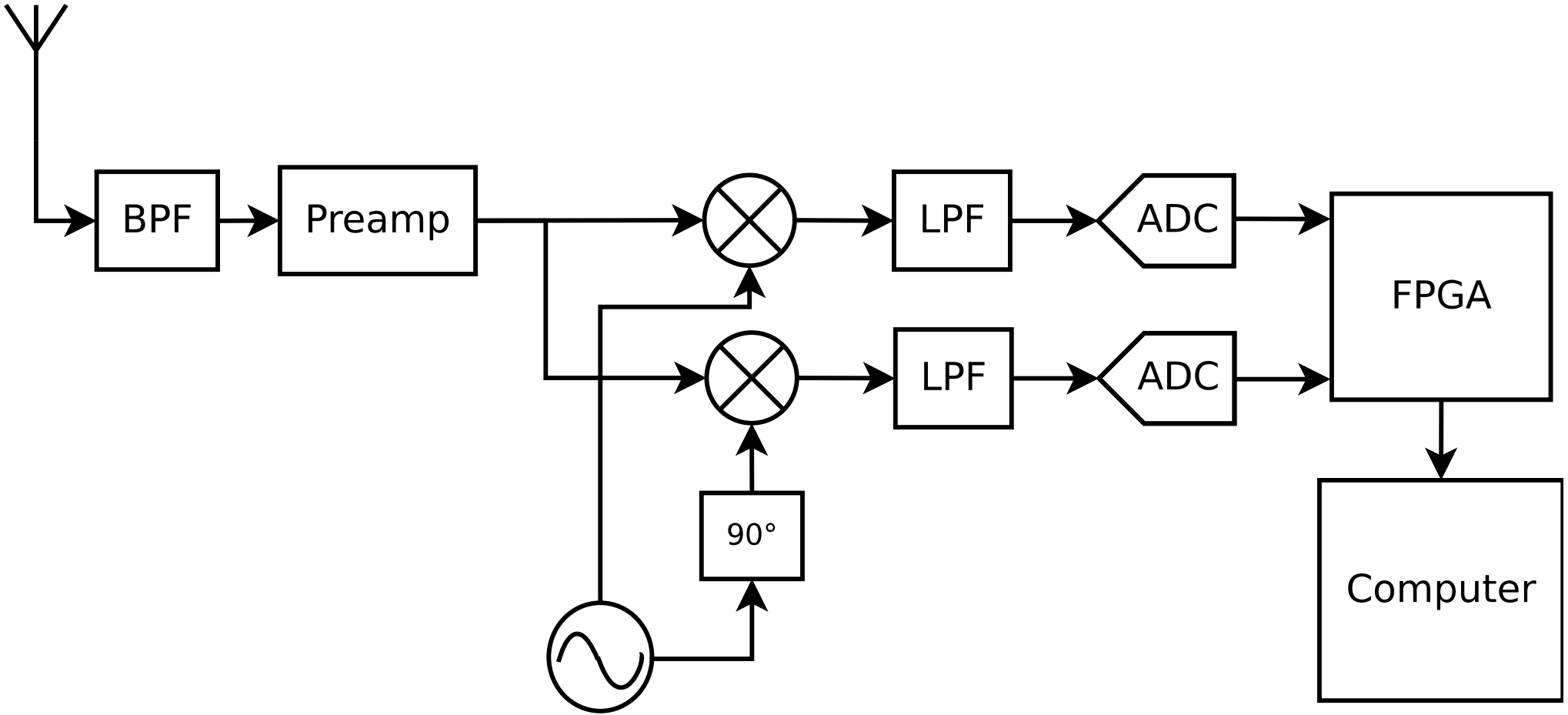
1. What is SDR?
2. Why SDR?
3. New hardware
4. Spectrum painting
5. HD Radio
6. Reverse Engineering
7. Digital Voice

# What is SDR?

# “Ideal” SDR Receiver



# Practical SDR Receiver



# Why SDR?

# The Comprehensive GNU Radio Archive Network

The Comprehensive GNU Radio Archive Network (CGRAN) is a free open source repository for 3rd party GNU Radio applications a.k.a Out Of Tree Modules that are not officially supported by the GNU Radio project.

## Browse~Checkout~Hack

Search

| Name | Tags | Description | Repository |
|---|---|---|---|
| gr-eventstream | scheduler, streams, bursty | The event stream scheduler | Github |
| Receiver for Vaisala Weather Sonde | | Receiver for Vaisala Weather Sonde | Github |
| gr-pyqt | gui, plotting, pyqt, pyqwt | Python QT Plotters and Message Tools Repo | Github |
| gr-pcap | pcap, packet | PCAP recording and playback | Github |
| gr-microtelecom | hardware, source | Microtelecom's Perseus SDR source module | Github |
| gr-lte | LTE, synchronization, estimation, PBCH | LTE downlink receiver blocks | Github |
| gr-nmea | sdr, gps, nmea | interface to NMEA and GPSD sources | Github |
| gr-ieee802-11 | IEEE 802.11, WiFi, OFDM | IEEE 802.11 a/g/p Transceiver | Github |
| An IEEE 802.15.4 (ZigBee) Transceiver | sdr, IEEE 802.15.4, ZigBee | gr-ieee802-15-4 | Github |
| gr-uhdgps | uhd, gps | GR Blocks to assist in GPS Data logging with UHD and a GPSDO | Github |
| gr-radar | radar, UHD | GNU Radio Radar Toolbox | Github |
| gr-nacl | encryption | GNU Radio module for data encryption using NaCl library | Github |

SDR# v1.0.0.1631 - Spy Server

000.003.799.000

AIRSPY

Source: Spy Server

Spy Server

sdr://airspy.com:5557 C

Device: Airspy HF+ SN: 39323035

Server: 2.0.1629 47 kB/s

Use full IQ

Bandwidth 660 kHz

Radio

NFM AM LSB USB

WFM DSB CW RAW

Shift 0

Filter Blackman-Harris 4

Bandwidth 2.400

Order 1.000

Squelch 50

CW Shift 1.000

FM Stereo

Step Size

Snap to Grid 100 Hz

Lock Carrier

Correct IQ

Anti-Fading

Swap I & Q

Audio

AGC

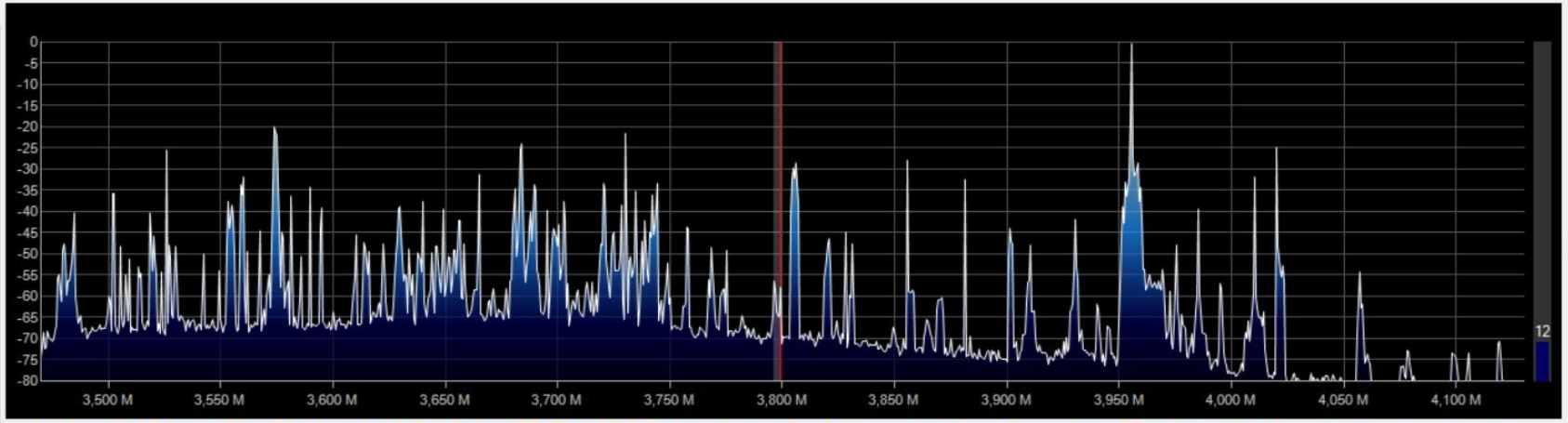
FFT Display

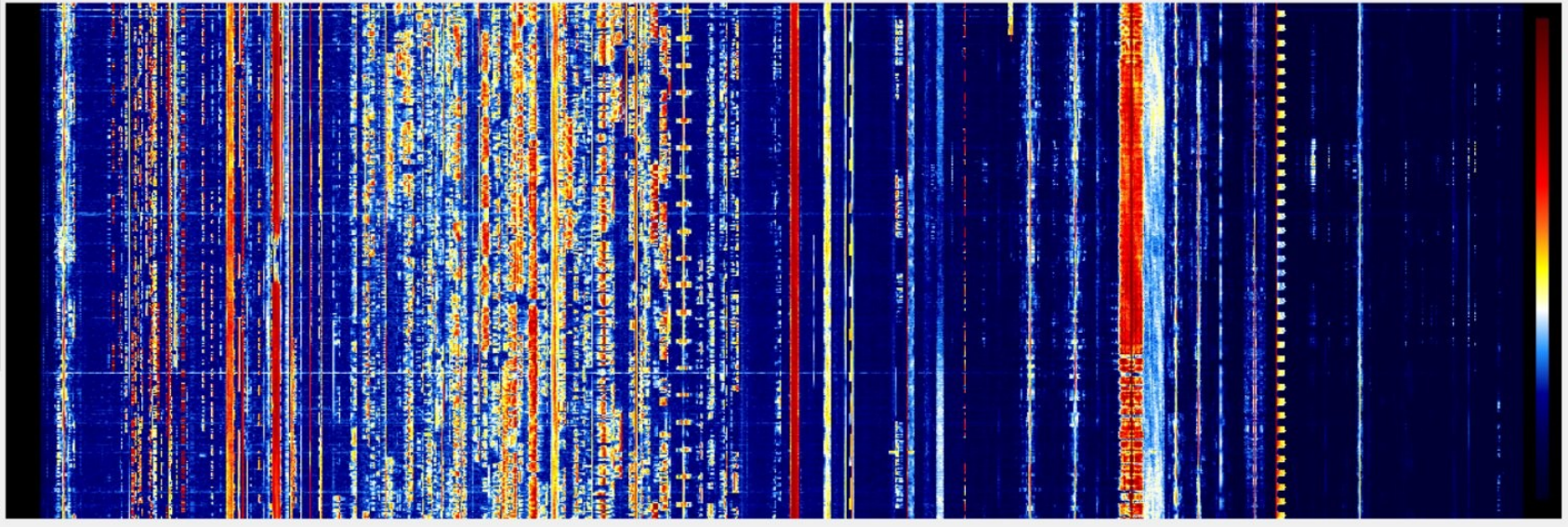
Audio Noise Reduction *

IF Noise Reduction *

Baseband Noise Blanker *

Demodulator Noise Blanker *



Zoom

Contrast

Range

Offset

# New Hardware

# RTL-SDR

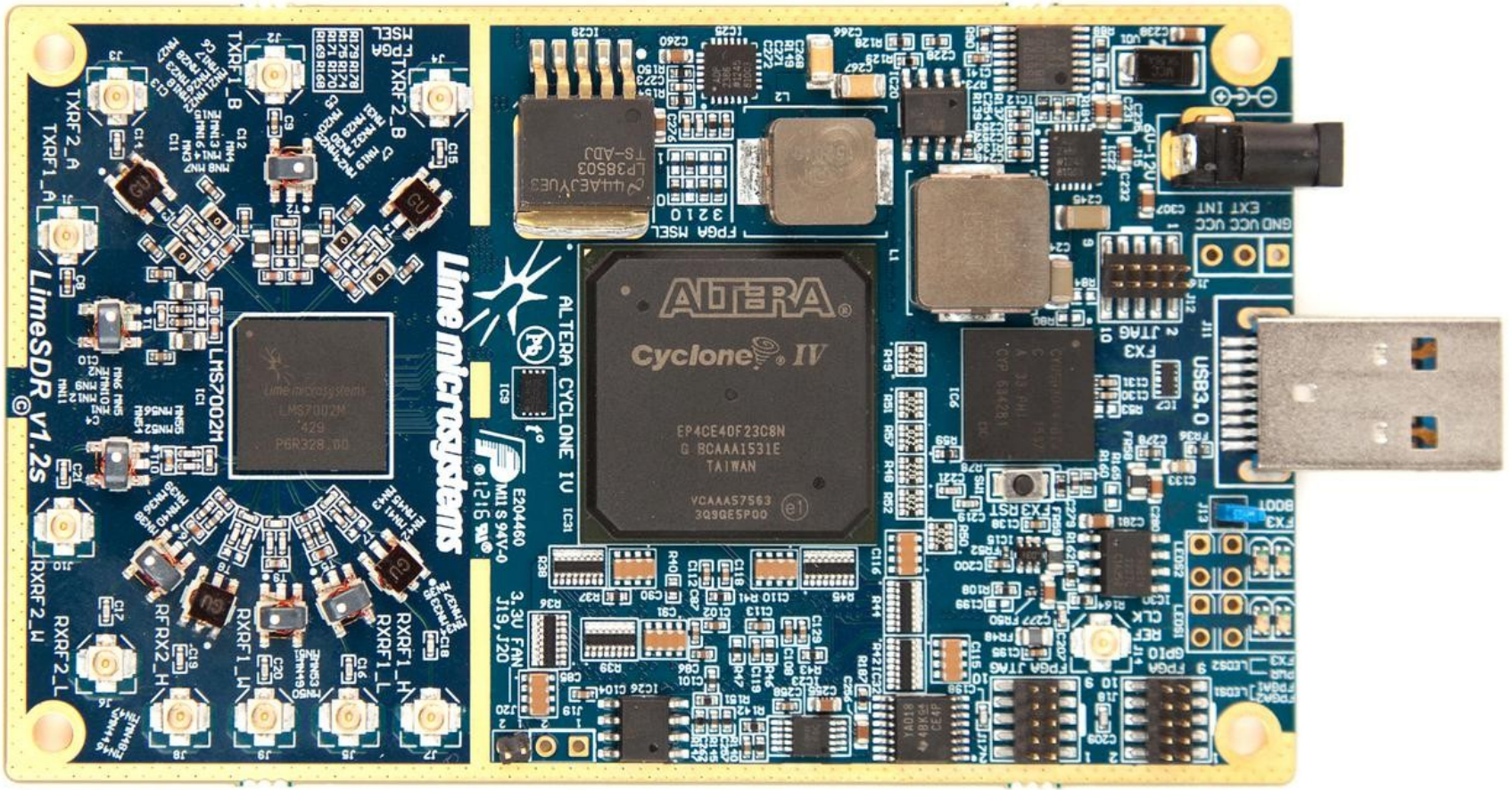

NooElec.com
RTL-SDR.COM
QUICKSTART SETUP GUIDE: RTL-SDR.COM/QSG
DVB-T+DAB+FM+SDR
RTL2832U R820T2 TCXO+BIAS T+HF
V.3

# Airspy

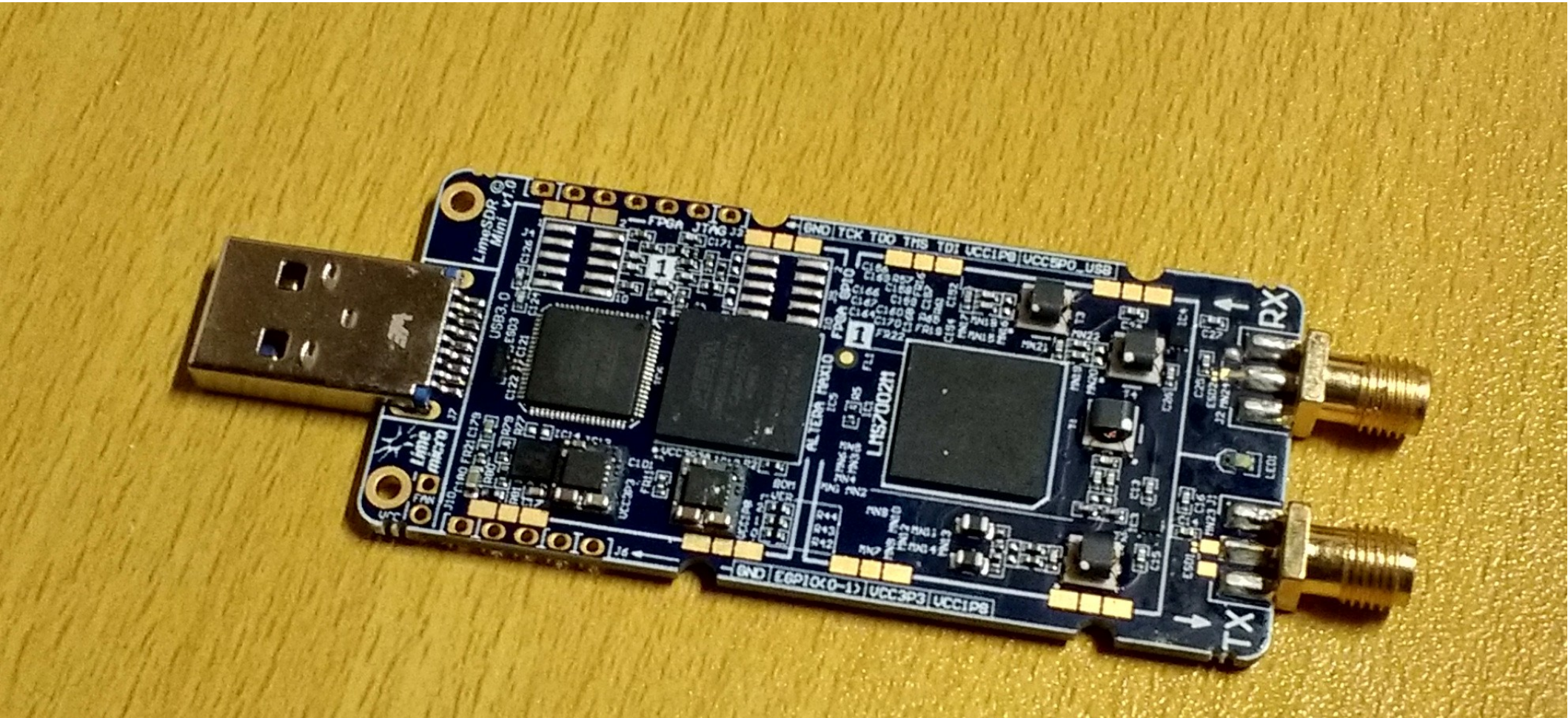
# Airspy Mini

# Airspy HF+

# LimeSDR


LimeSDR v1.2s
Lime Microsystems
LMS7002M
ALTERA
Cyclone IV
EP4CE40F23C8N
ALTERA CYCLONE IV
USB3.0
TXRF1_A
TXRF2_A
TXRF1_B
TXRF2_B
RXRF2_W
RXRF2_L
RXRF1_W
RXRF1_L
RXRF1_H
RFRX2_H
3.3V FAN
J19, J20

# LimeSDR Mini



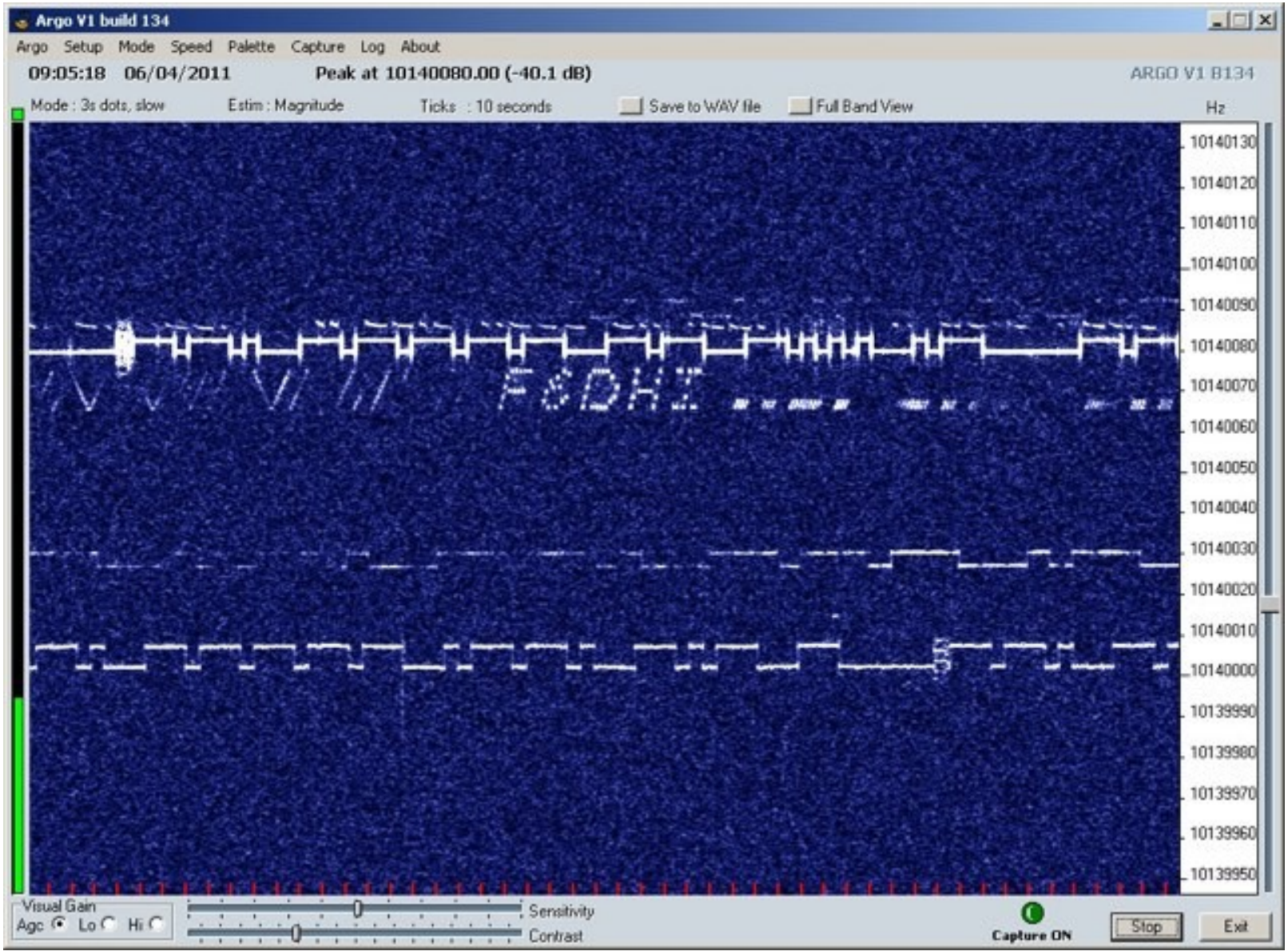
# ICOM IC-7300


POWER
TRANSMIT
TUNER
VOX/BK-IN
PHONES
MIC
TWIN PBT
CLR
P.AMP
ATT
NOTCH
NB
NR
AF
RF/SQL
ICOM
HF/50MHz TRANSCEIVER IC-7300
LSB
FIL2
13:33
TX
VFO A 1
S/Po
7.073.00
7.030
7.200
7.04 7.06 7.08 7.10 7.12 7.14 7.16 7.18
ATT OFF
HOLD
AUDIO SCOPE
0 1.0 2.0 3.0 4.0kHz
LEVEL:-30dB
300ms/Div
ATT
HOLD
LEVEL
TIME
EXPD/SET
MULTI
XFC
TX/RX
AUTO TUNE
RIT
ΔTX
CLEAR
SPLIT
A/B
V/M
MPAD
MENU
FUNCTION
M.SCOPE
QUICK
EXIT
SPEECH

# Hardware Comparison

| Radio | Max. bandwidth (MHz) | Bits | Tuning range (MHz) | Approx. Cost (CAD) | Notes |
|---|---|---|---|---|---|
| RTL-SDR | 2.4 | 8 | 24-1750 | $25 | RX-only |
| Airspy | 10 | 12 | 24-1800 | $210 | RX-only |
| Airspy Mini | 6 | 12 | 24-1800 | $125 | RX-only |
| Airspy HF+ | 0.66 | 16 | 0.009-31, 60-260 | $250 | RX-only |
| LimeSDR | 56 | 12 | 0.1-3800 | $375 | 2x2 MIMO |
| LimeSDR Mini | 28 | 12 | 10-3500 | $175 | Full duplex |

# Spectrum painting


Argo V1 build 134
Argo Setup Mode Speed Palette Capture Log About
09:05:18 06/04/2011
Peak at 10140080.00 (-40.1 dB)
ARGO V1 B134
Mode : 3s dots, slow
Estim : Magnitude
Ticks : 10 seconds
Save to WAV file
Full Band View
Hz
10140130
10140120
10140110
10140100
10140090
10140080
10140070
10140060
10140050
10140040
10140030
10140020
10140010
10140000
10139990
10139980
10139970
10139960
10139950
Visual Gain
Agc Lo Hi
Sensitivity
Contrast
Capture ON
Stop
Exit

# drmpeg / gr-paint





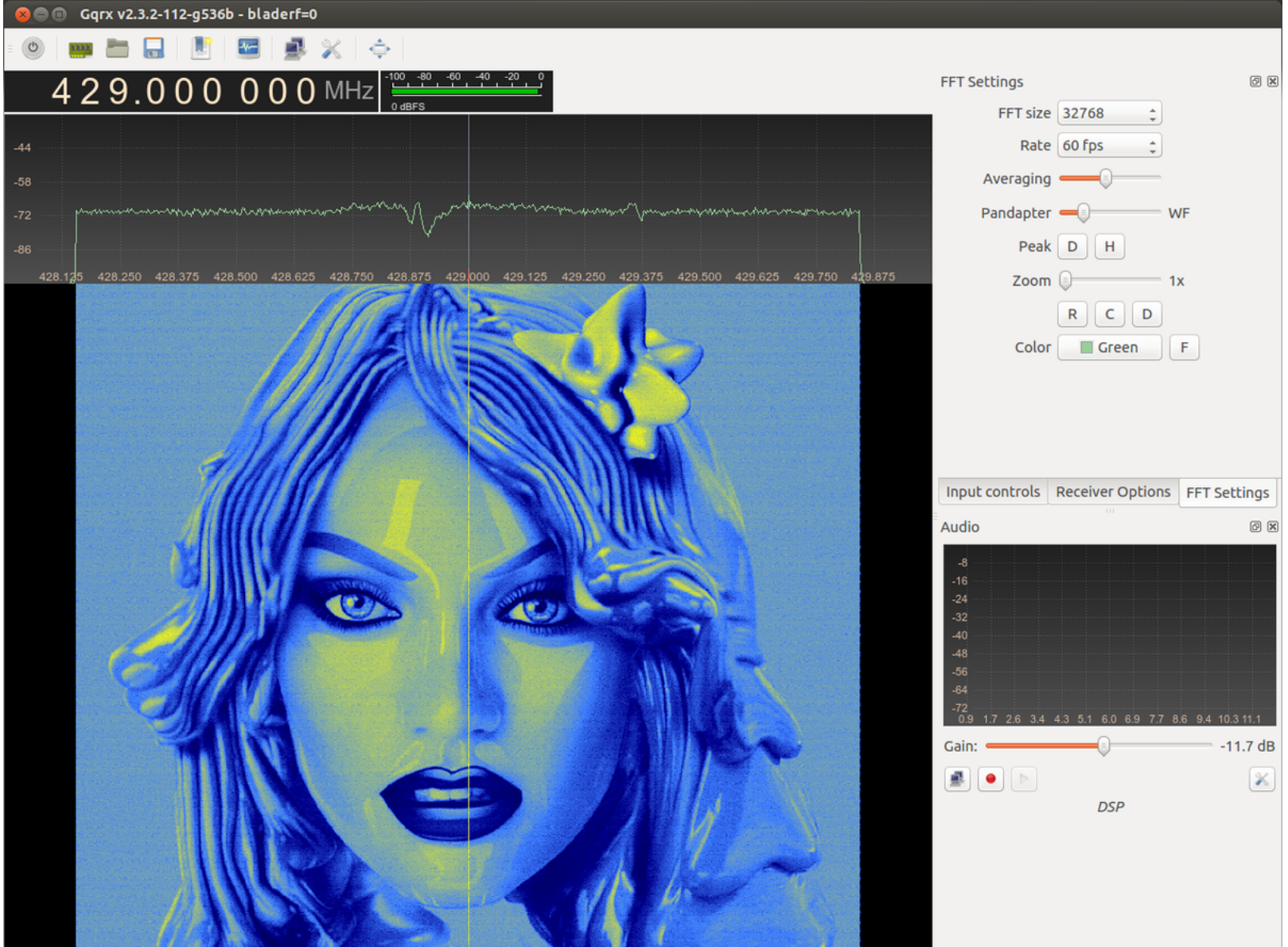
An OFDM Spectrum Painter for GNU Radio

31 commits 1 branch 0 releases 2 contributors GPL-3.0



drmpeg Use a GRC category in XML files. Latest commit 7f2cbf2 19 days ago

| | | |
|---|---|---|
| apps | Add sin(x)/x correction for bladeRF. | 2 years ago |
| cmake | Update build to latest. | 9 months ago |
| docs | Update build to latest. | 9 months ago |
| examples | first commit | 3 years ago |
| grc | Use a GRC category in XML files. | 19 days ago |
| include/paint | Install paint_config.h include file. | a year ago |
| lib | Update build to latest. | 9 months ago |
| python | repeat mode for image source | 2 years ago |

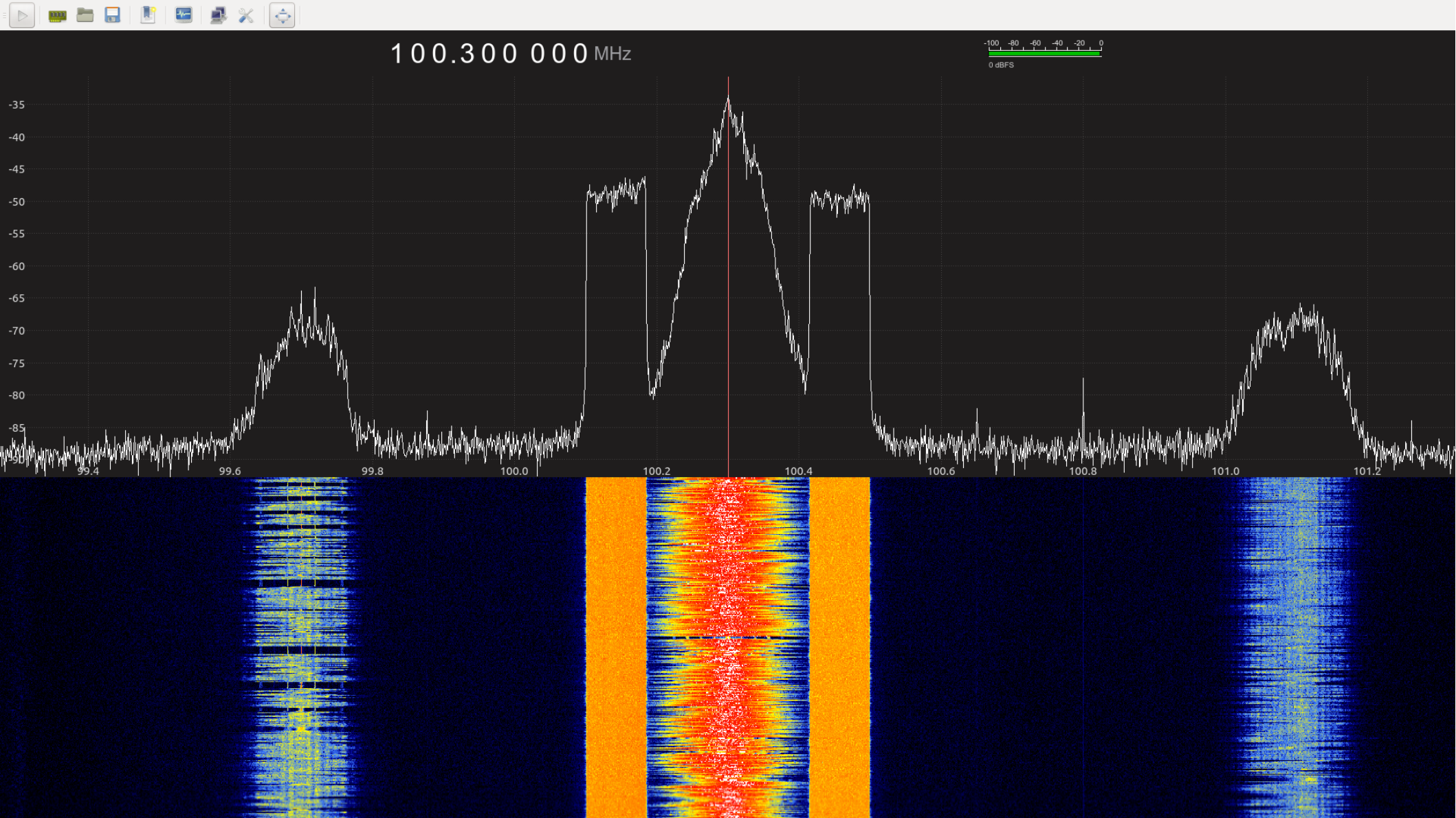

Gqrx v2.3.2-112-g536b - bladerf=0
429.000 000 MHz
-100 -80 -60 -40 -20 0
0 dBFS
-44
-58
-72
-86
428.125 428.250 428.375 428.500 428.625 428.750 428.875 429.000 429.125 429.250 429.375 429.500 429.625 429.750 429.875
FFT Settings
FFT size 32768
Rate 60 fps
Averaging
Pandapter WF
Peak D H
Zoom 1x
R C D
Color Green F
Input controls
Receiver Options
FFT Settings
Audio
-8
-16
-24
-32
-40
-48
-56
-64
-72
0.9 1.7 2.6 3.4 4.3 5.1 6.0 6.9 7.7 8.6 9.4 10.3 11.1
Gain: -11.7 dB
DSP


HD Radio®


HD Radio®


Drm


DAB+
Digital Audio Broadcasting


100.300 000 MHz
-100 -80 -60 -40 -20 0
0 dBFS
-35
-40
-45
-50
-55
-60
-65
-70
-75
-80
-85
99.4
99.6
99.8
100.0
100.2
100.4
100.6
100.8
101.0
101.2

NRSC STANDARD

# NATIONAL RADIO SYSTEMS COMMITTEE

## NRSC-5-D
## In-band/on-channel Digital Radio Broadcasting Standard
## April, 2017

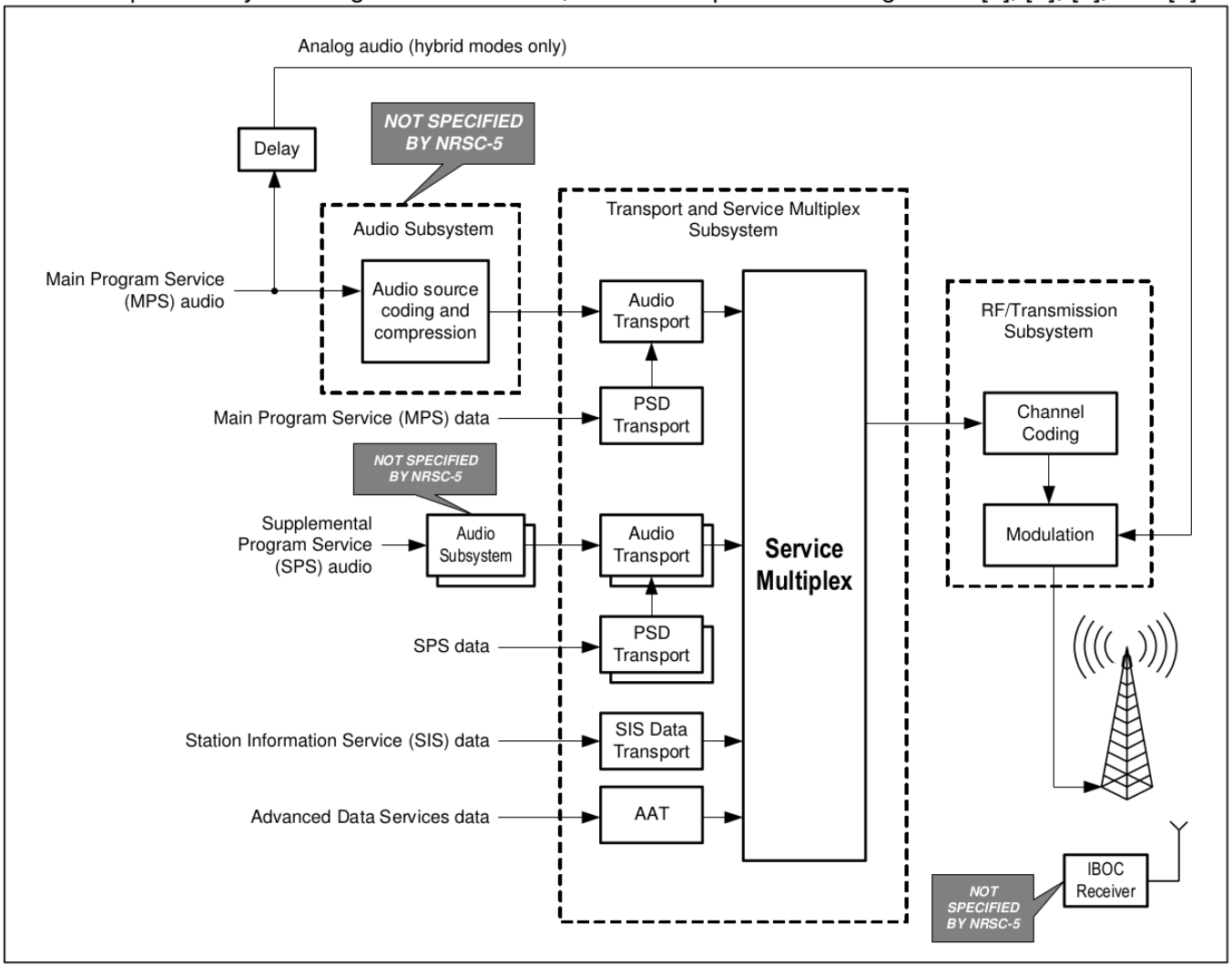



*Figure 1: Overview of IBOC digital radio broadcasting system*

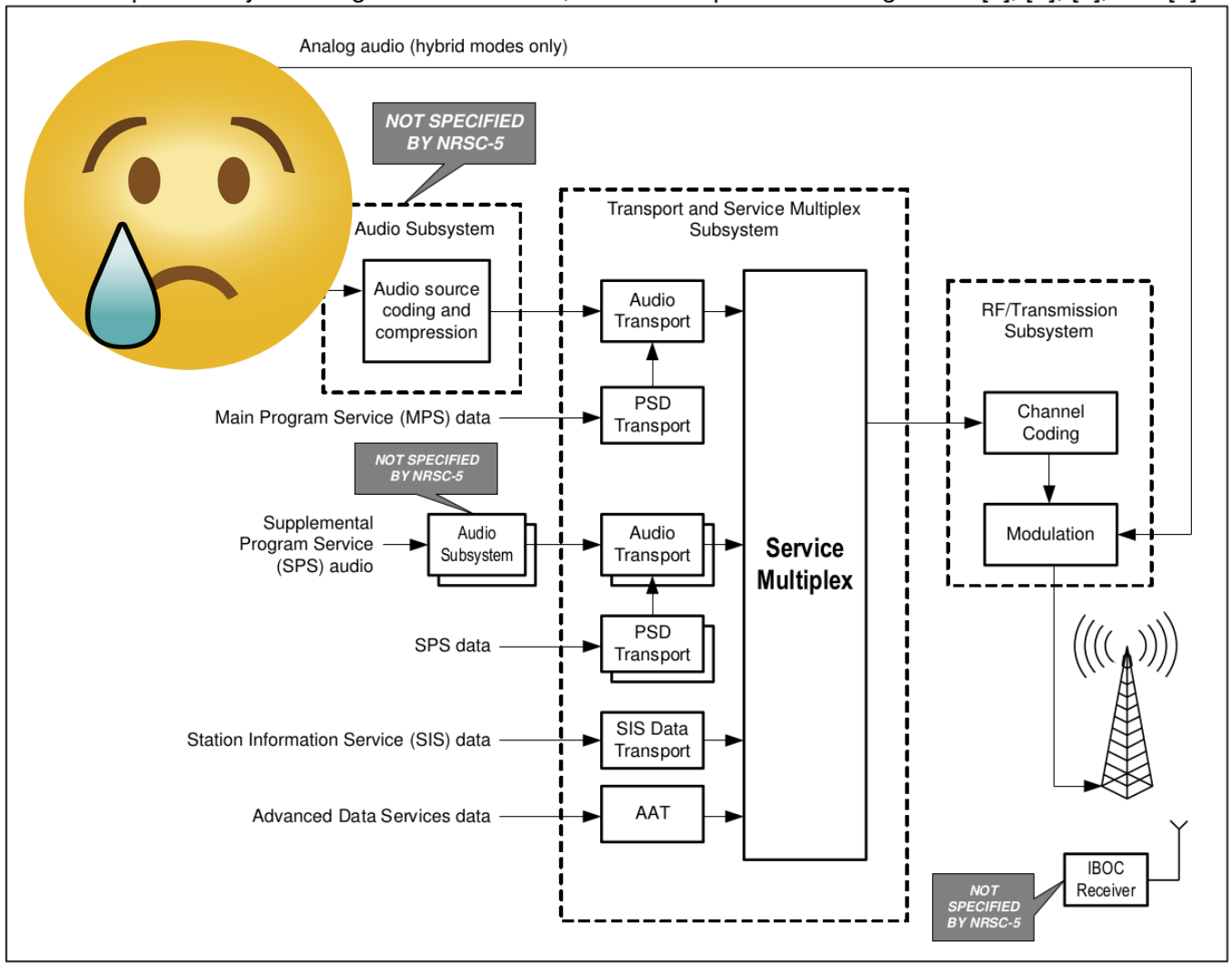



*Figure 1: Overview of IBOC digital radio broadcasting system*

# argilo / nrsc-5

Unwatch 5 Star 16 Fork 3

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

A software implementation of HD Radio (NRSC-5-C) — Edit

**15** commits **1** branch **0** releases **1** contributor GPL-3.0

Branch: **master** New pull request Create new file Upload files Find file Clone or download

**argilo** Generate a signal with 8 channels (HD1 through HD8). Latest commit 62b7829 on Oct 16

| | | |
|---|---|---|
| .gitignore | Add P1 audio transport generator. | a month ago |
| COPYING | Initial commit. | 2 months ago |
| README.md | Generate a signal with 8 channels (HD1 through HD8). | a month ago |
| hd_tx_hackrf.grc | Add flowgraph for the HackRF. | a month ago |
| hd_tx_hackrf.py | Add flowgraph for the HackRF. | a month ago |
| hd_tx_usrp.grc | Add flowgraph for the HackRF. | a month ago |
| hd_tx_usrp.py | Add flowgraph for the HackRF. | a month ago |
| l1_gen.py | Add Layer 1 FM. | 2 months ago |


95.7 HD3+
HACK-FM/HackRF!/
HD Radio®

# RECEIVING NRSC-5

by **Theori** — **09 Jun 2017**

NRSC-5-C is the standard for digital terrestial radio in the United States. The physical layer and protocols are well documented on the NRSC's website. The audio compression details are conspicuously absent considering that the goal of the standard is digital audio.

## Overview

The development goal of NRSC-5-C was digital terrestial radio using the already allocated FM bandwidth. The result was an IBOC (in-band on-channel) system that supports both analog and digital signals within the existing FM allocations. This hybrid scheme is flexible and the station can choose to allocate more or less of the spectrum to analog or digital. Eventually, the entire spectrum allocation could be used for a pure digital signal.

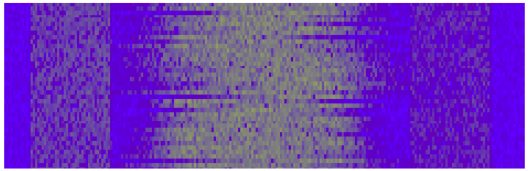


*Figure 1: Spectrogram showing analog FM signal with digital sidebands*

# theori-io / nrsc5





NRSC-5 receiver for rtl-sdr

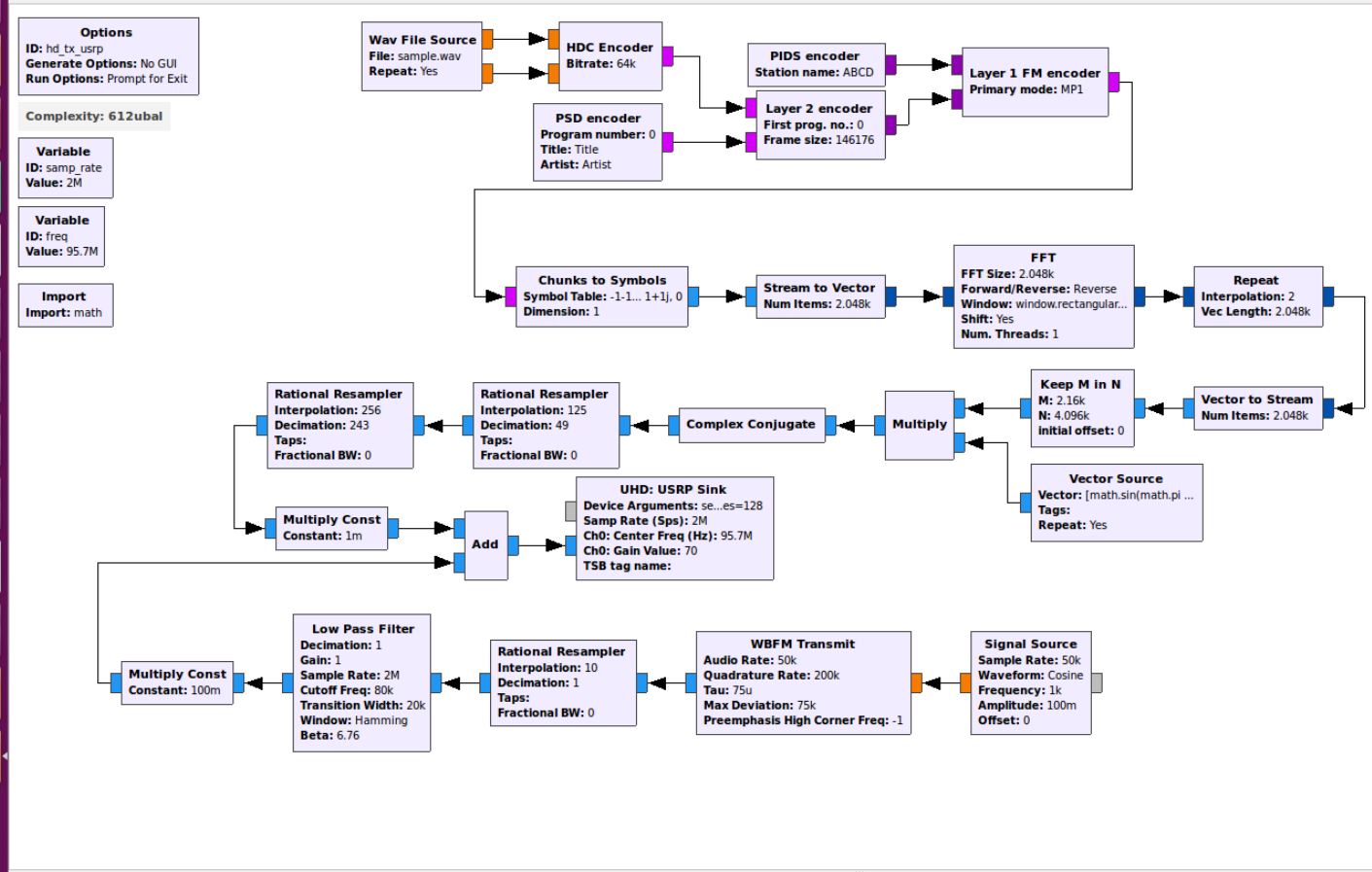
122 commits 6 branches 0 releases 5 contributors



awesie committed on GitHub Merge pull request #56 from theori-io/fix-cfo-idx-overflow ... Latest commit 1bd3f79 8 days ago

| | | |
|---|---|---|
| src | Fix cfo_idx integer overflow. | 8 days ago |
| support | Add support for HDC block type 0. Fixes #8. | 3 months ago |
| .gitignore | Add a gitignore. | a month ago |
| CMakeLists.txt | Fix CMPLXF and strndup on MinGW. | a month ago |
| LICENSE | Public release | 3 months ago |
| README.md | Accept frequency in MHz or Hz. | 8 days ago |
| nrsc5.rb | Also require git for build | 3 months ago |

README.md




| Id | Value |
|---|---|
| Imports | |
| import_0 | import math |
| Variables | |
| freq | 95.7e6 |
| samp_rate | 2000000 |

- Debug Tools
- Deprecated
- Digital Television
- Equalizers
- Error Coding
- FCD
- File Operators
- Filters
- Fourier Analysis
- GUI Widgets
- Impairment Models
- Instrumentation
- Level Controllers
- Math Operators
- Measurement Tools
- Message Tools
- Misc
- Modulators
- Networking Tools
- NOAA
- OFDM
- Packet Operators
- Pager
- Peak Detectors
- Resamplers
- Stream Operators
- Stream Tag Tools
- Symbol Coding
- Synchronizers
- Trellis Coding
- Type Converters
- UHD
- Variables
- Video
- Waveform Generators
- ZeroMQ Interfaces
- (no module specified)
- NRSC-5
  - HDC Encoder
  - Layer 1 FM encoder
  - Layer 2 encoder
  - PIDS encoder
  - PSD encoder
- RDS

# argilo / gr-nrsc5

Unwatch 3 | Unstar 2 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Settings | Insights

A GNU Radio implementation of HD Radio (NRSC-5)

Edit

sdr radio gnuradio Manage topics

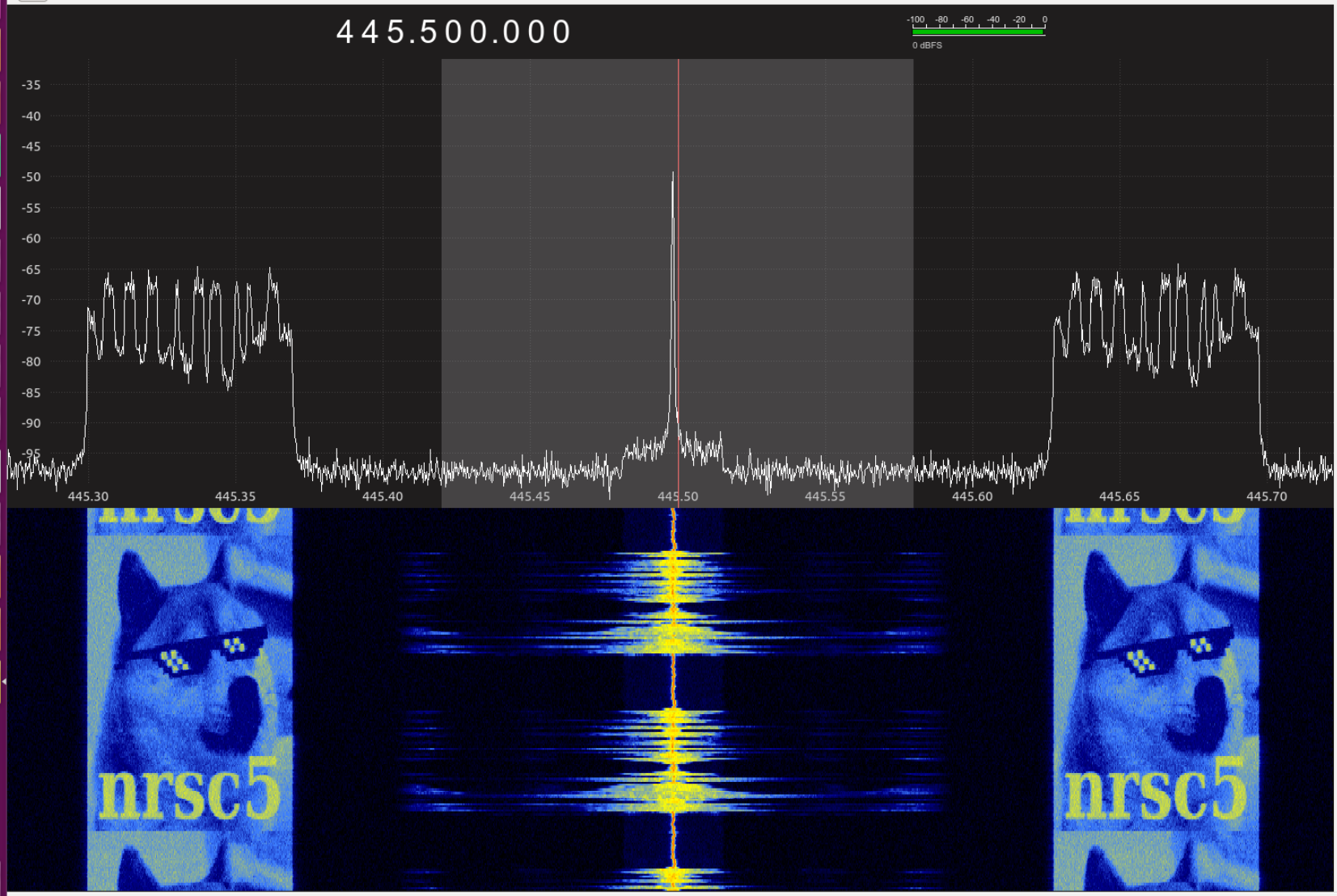
**43** commits | **1** branch | **0** releases | **1** contributor | GPL-3.0

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download**

**argilo** Add sample WAV file. ···

Latest commit `9fc3a30` 4 days ago

| | | |
|---|---|---|
| apps | Add primary service mode 3. | 16 days ago |
| cmake | Empty GNU Radio module. | 28 days ago |
| docs | Get HDC encoder block working. | 22 days ago |
| examples | Empty GNU Radio module. | 28 days ago |
| grc | Implement primary modes 5 and 6. | 12 days ago |
| include/nrsc5 | Updated references to NRSC-5-D. | 6 days ago |
| lib | Updated references to NRSC-5-D. | 6 days ago |




Input controls

LNB LO 0.000000 MHz

Hardware AGC

LNA gain 0.0 dB

Swap I/Q

No limits

DC remove

IQ balance

Freq. correction 0.0 ppm

Antenna RX

Reset frequency controller digits

Input controls | Receiver Options | FFT Settings

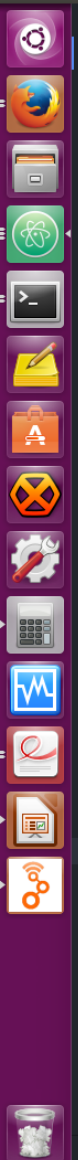
Audio

-20
-40
-60
2 4 6 8 10

Gain: 0.0 dB

UDP Rec Play ...

DSP

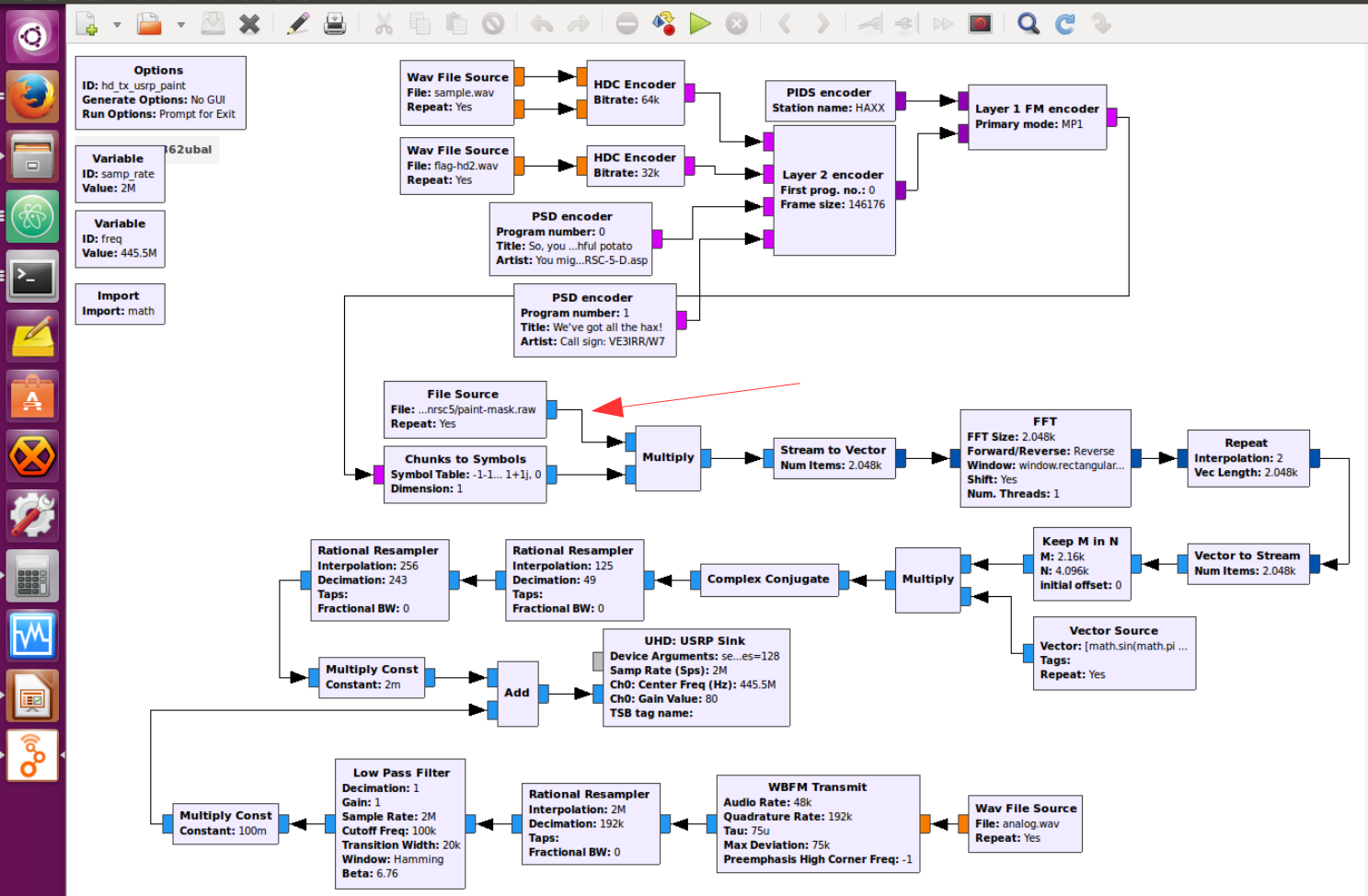






paint.py

```python
from PIL import Image
import math
import struct

out_cols = 191

image = Image.open("doge2.png")
in_cols, in_rows = image.size
out_rows = round(out_cols * in_rows / in_cols)

image = image.rotate(180)
image = image.resize((out_cols, out_rows), Image.ANTIALIAS)
image = image.convert('L')
pixels = list(image.getdata())

with open('paint-mask.raw', 'wb') as f:
    for r in range(out_rows):
        row = pixels[r * out_cols:(r+1) * out_cols]
        row = [math.pow(10, (p / 255) * 1.0 - 0.5) for p in row]
        row = ([0] * 478) + row + ([0] * 711) + row + ([0] * 477)
        for _ in range(20):
            for i in range(2048):
                f.write(struct.pack('ff', row[i], 0))
```




| Id | Value |
|---|---|
| Imports | |
| import_0 | import math |
| Variables | |
| freq | 92.9e6 * 0 + 445.5e6 * 1 + 918.0e6 * 0 |
| samp_rate | 2000000 |

- Debug Tools
- Deprecated
- Digital Television
- Equalizers
- Error Coding
- FCD
- File Operators
- Filters
- Fourier Analysis
- GUI Widgets
- Impairment Models
- Instrumentation
- Level Controllers
- Math Operators
- Measurement Tools
- Message Tools
- Misc
- Modulators
- Networking Tools
- NOAA
- OFDM
- Packet Operators
- Pager
- Peak Detectors
- Resamplers
- Stream Operators
- Stream Tag Tools
- Symbol Coding
- Synchronizers
- Trellis Coding
- Type Converters
- UHD
- Variables
- Video
- Waveform Generators
- ZeroMQ Interfaces
- (no module specified)
- NRSC-5
  - HDC Encoder
  - Layer 1 FM encoder
  - Layer 2 encoder
  - PIDS encoder
  - PSD encoder
- RDS

# Reverse Engineering


CHAMBERLAIN
CHAMBERLAIN
CHAMBERLAIN
CHAMBERLAIN

“Your Security+ opener and hand-held remote control have been factory-set to a matching code which changes with each use, randomly accessing over 100 billion new codes.”

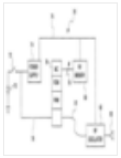
Google

chamberlain rolling code

All Shopping Images Videos More Settings Tools

About 1,970 results (0.18 seconds)

## Rolling code security system



www.google.ca/patents/US6980655

Grant - Filed 17 Oct 2001 - Issued 27 Dec 2005 - Bradford L. Farris - The Chamberlain Group, Inc.

A **rolling code** transmitter is useful in a security system for providing secure encrypted RF transmission comprising an ... **Chamberlain** ...

Overview · Related · Discuss

## Rolling code encryption process for remote keyless entry system

www.google.ca/patents/US5420925

Grant - Filed 3 Mar 1994 - Issued 30 May 1995 - Paul A. Michaels - Lectron Products, Inc.

An encryption process for a **rolling code** of a remote keyless entry system. ... 2000, The **Chamberlain** Group, Inc. **Rolling code** security system.

Overview · Related · Discuss

## Rolling code security system

www.google.ca/patents/US7492905

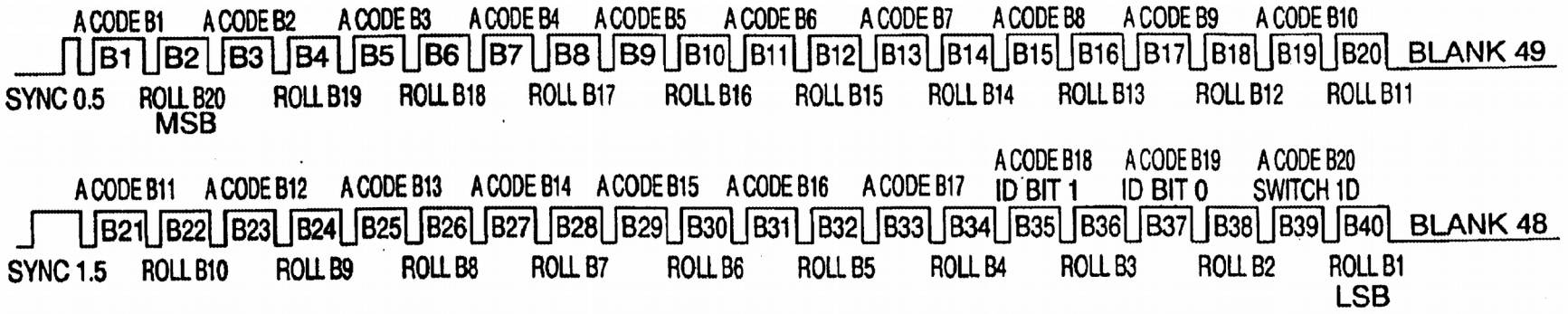
Grant - Filed 14 Aug 2002 - Issued 17 Feb 2009 - James J. Fitzgibbon - The Chamberlain Group, Inc.

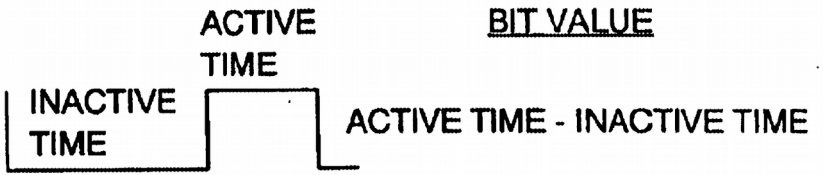
A **rolling code** transmitter is useful in a security system for providing secure encrypted RF

The program code listing for the transmitter is set forth at pages A-1 through A-19 and for the receiver at pages A-20 through A-51 of the attached appendix. Referring now to FIGS. 7A through 7C, the flow chart set forth therein describes the operation of the transmitter. A rolling code is incremented by three in a step **500**, followed by the rolling code being stored for the next transmission from the transmitter when the transmitter button is pushed. The order of the binary digits in the rolling code is inverted or mirrored in a step **504**, following which in a step **506**, the most significant digit is converted to zero effectively truncating the binary rolling code. The rolling code is then changed to a trinary code having values 0, 1 and 2 and the initial trinary rolling code is set to 0. It may be appreciated that it is trinary code which is actually used to modify the radio frequency oscillator signal and the trinary code is best seen in FIG. **6**. It may be noted that the bit timing in FIG. **6** for a 0 is 1.5 milliseconds down time and 0.5 millisecond up time, for a **1**, 1 millisecond down and 1 millisecond up and for a 2, 0.5 millisecond down and 1.5 milliseconds up. The up time is actually the active time when carrier is being generated. The down time is inactive when the carrier is cut off. The codes are assembled in two frames, each of 20 trinary bits, with the first frame being identified by a 0.5 millisecond sync bit and the second frame being identified by a 1.5 millisecond sync bit.

routine is exited in a step **708**. In the event that it is an active time, the active time is stored in memory in a step **710** and the bit counter is tested in a step **712**. If the bit counter zero, control is transferred to a step **714**, as may best be seen in FIG. **8**B and a test is made to determine whether the inactive time is between 20 milliseconds and 55 milliseconds. If it is not, the bit counter is cleared as well as the rolling code register and the fixed code register in step **716** and the routine is exited in step **718**.

In the event that the inactive time is between 20 milliseconds and 55 milliseconds, a test is made in a step **720** to determine whether the active time is greater than 1 millisecond, as shown in FIG. **8**C. If it is not, a test is made in a step **722** to determine whether the inactive time is less than 0.35 millisecond. If it is, a frame **1** flag is set in a step **728** identifying the incoming information as being associated with frame **1** and the interrupt routine is exited in a step **730**. In the event that the active time test in step **722** is not less than 0.35 millisecond, in the step **724**, the bit counter is cleared as well as the rolling code register and the fixed register and the return is exited in the step **726**. If the active time is greater than 1 millisecond as tested in step **720**, a test is made in a step **732** to determine whether the active time is greater than 2.0 milliseconds. If it is not, the frame **2** flag is set in a step **734** and the routine is exited in step **730**. If the active time is greater than 2 milliseconds, the bit counter


A CODE B1
A CODE B2
A CODE B3
A CODE B4
A CODE B5
A CODE B6
A CODE B7
A CODE B8
A CODE B9
A CODE B10
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17 B18 B19 B20
BLANK 49
SYNC 0.5
ROLL B20 MSB
ROLL B19
ROLL B18
ROLL B17
ROLL B16
ROLL B15
ROLL B14
ROLL B13
ROLL B12
ROLL B11
A CODE B11
A CODE B12
A CODE B13
A CODE B14
A CODE B15
A CODE B16
A CODE B17
A CODE B18 ID BIT 1
A CODE B19 ID BIT 0
A CODE B20 SWITCH 1D
B21 B22 B23 B24 B25 B26 B27 B28 B29 B30 B31 B32 B33 B34 B35 B36 B37 B38 B39 B40
BLANK 48
SYNC 1.5
ROLL B10
ROLL B9
ROLL B8
ROLL B7
ROLL B6
ROLL B5
ROLL B4
ROLL B3
ROLL B2
ROLL B1 LSB


ACTIVE TIME
INACTIVE TIME
BIT VALUE
ACTIVE TIME - INACTIVE TIME

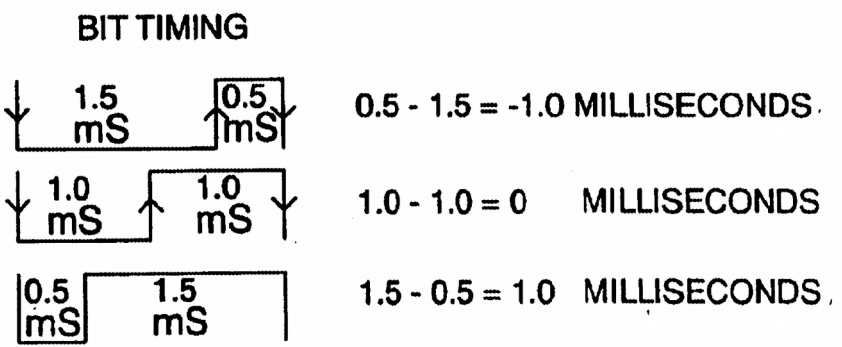

BIT TIMING
1.5 mS
0.5 mS
0.5 - 1.5 = -1.0 MILLISECONDS
1.0 mS
1.0 mS
1.0 - 1.0 = 0 MILLISECONDS
0.5 mS
1.5 mS
1.5 - 0.5 = 1.0 MILLISECONDS

*Fig. 6*

# FCC ID Search

**Equipment Authorization Approval Guide**

- Knowledge Database (KDB)
- FCC ID Search
- Equipment Authorization Measurement Procedures
- Equipment Authorization Procedures
- Equipment Authorization Presentations
- Grantee Code
- Importation
- Equipment Authorization

## FCC ID Search Form

Help | Advanced Search

**Grantee Code: (First three or five characters of FCCID)***

HBW

**Product Code: (Remaining characters of FCCID)**

1573

search

## Advanced Search

OET Home Page

Filing Options

- Grantee Registration
- Modify Grantee Information
- Submit Correspondence
- Renew Test Firm/Add Exhibits
- Test Firm Accrediting Body Login
- Return to 159 Form
- Add/Modify Grant Deferral Date
- Change Short-Term Confidential Date

Reports

- Pending Application Status
- Authorization Search
- Grantee Search
- Pending Grantee Search
- TCB Search
- Test Firms
- Test Firm Accrediting Bodies

FCC > FCC E-filing > EAS > Search

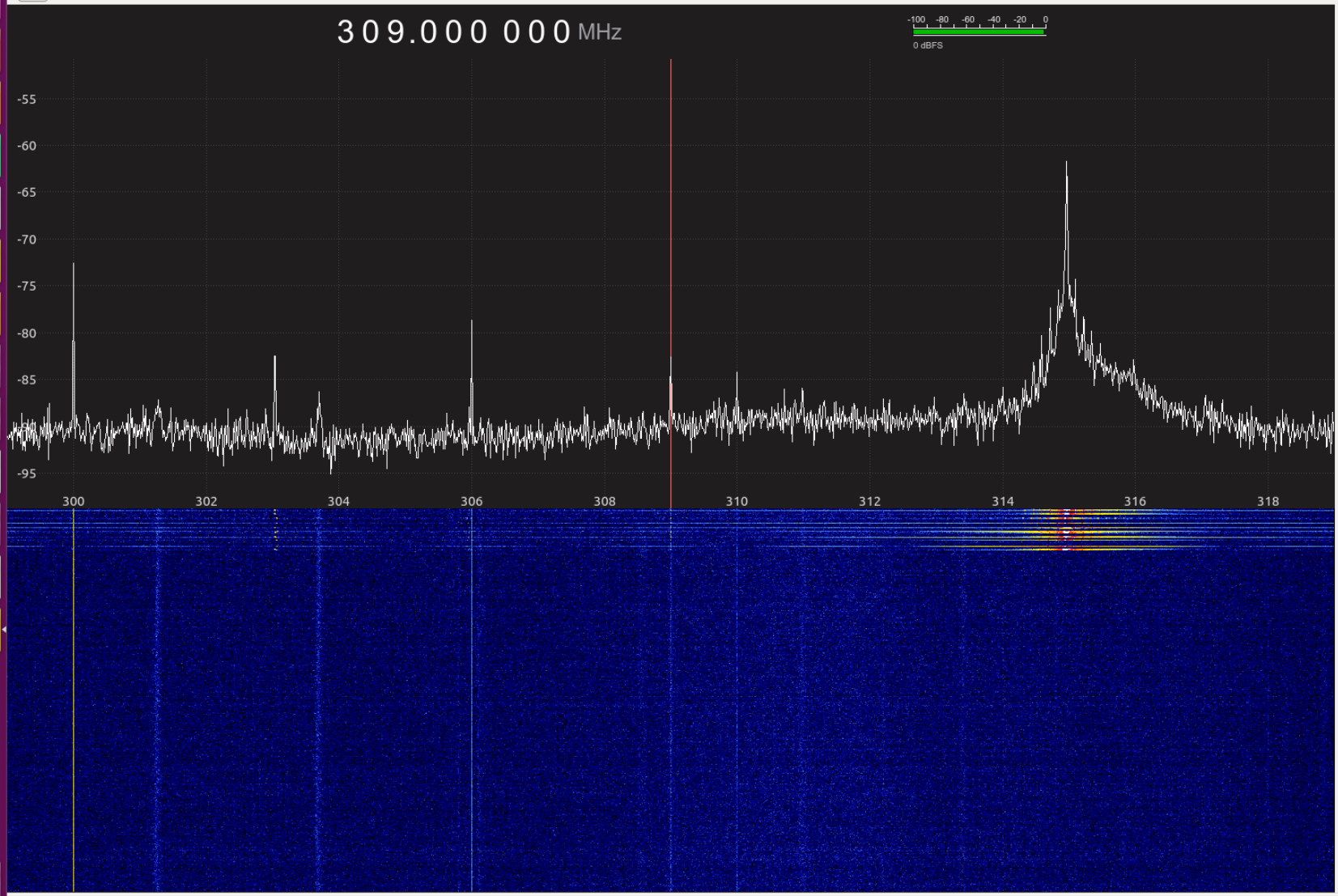
FCC Site Map

**2 results were found that match the search criteria:**
**Grantee Code: HBW Product Code: 1573**

**Displaying records 1 through 2 of 2.**

| View Form | Display Exhibits | Display Grant | Display Correspondence | Applicant Name | Address | City | State | Country | Zip Code | FCC ID | Application Purpose | Final Action Date | Lower Frequency In MHz | Upper Frequency In MHz |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Detail Summary | | | Chamberlain Group Inc, The | 845 Larch Avenue | Elmhurst | IL | United States | 60126 | HBW1573 | Original Equipment | 09/20/2001 | 315.0 | 315.0 |
| | Detail Summary | | | Chamberlain Group Inc, The | 845 Larch Avenue | Elmhurst | IL | United States | 60126 | HBW1573BC | Original Equipment | 02/26/2002 | 315.0 | 315.0 |

Perform Search Again




Receiver Options

0.000 kHz

Hardware freq: 309.000000 MHz

Filter width: Normal

Filter shape: Normal

Mode: Demod Off ...

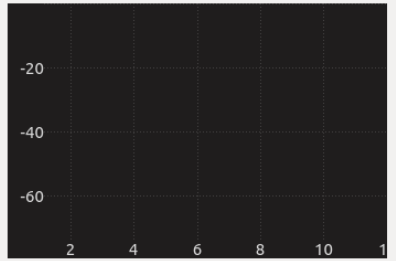
AGC: Medium ...

Squelch: -53.0 dBFS A

Noise blanker: NB1 NB2 ...

Input controls | Receiver Options | FFT Settings

Audio




Gain: 10 dB

UDP Rec Play ...

*DSP*

Click, drag or scroll on spectrum to tune. Drag and scroll X and Y axes for pan and zoom. Drag filter edges to adjust filter.

step1.grc - /home/argilo/Documents/chamberlain - GNU Radio Companion

step1 step2

**Options**
**ID:** step1
**Generate Options:** QT GUI

**Variable**
**ID:** samp_rate
**Value:** 2M

**osmocom Source**
**Sample Rate (sps):** 2M
**Ch0: Frequency (Hz):** 315M
**Ch0: Freq. Corr. (ppm):** 0
**Ch0: DC Offset Mode:** Off
**Ch0: IQ Balance Mode:** Off
**Ch0: Gain Mode:** Manual
**Ch0: RF Gain (dB):** 1
**Ch0: IF Gain (dB):** 20
**Ch0: BB Gain (dB):** 20
**Ch0: Bandwidth (Hz):** 1M

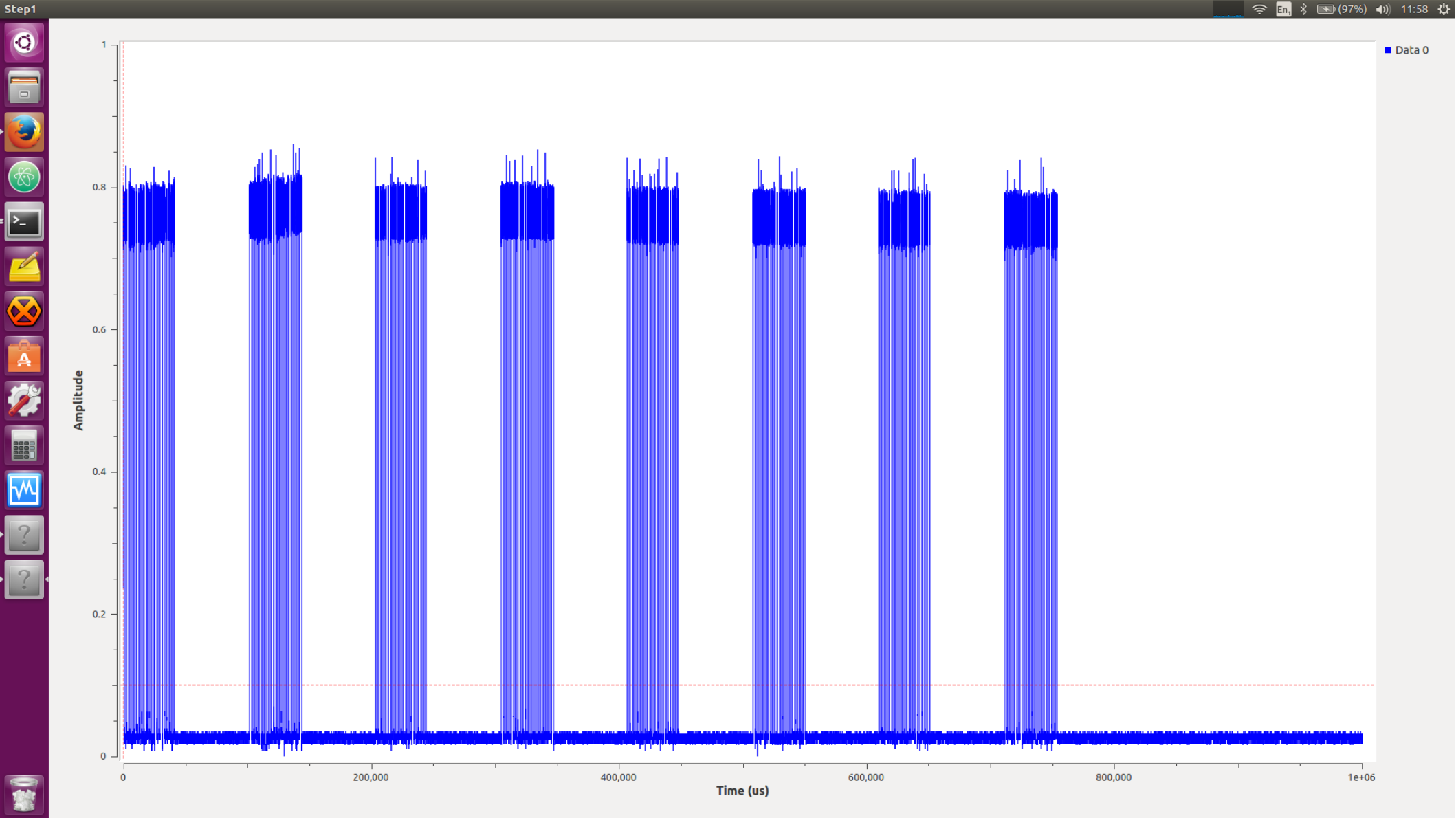
**Complex to Mag**

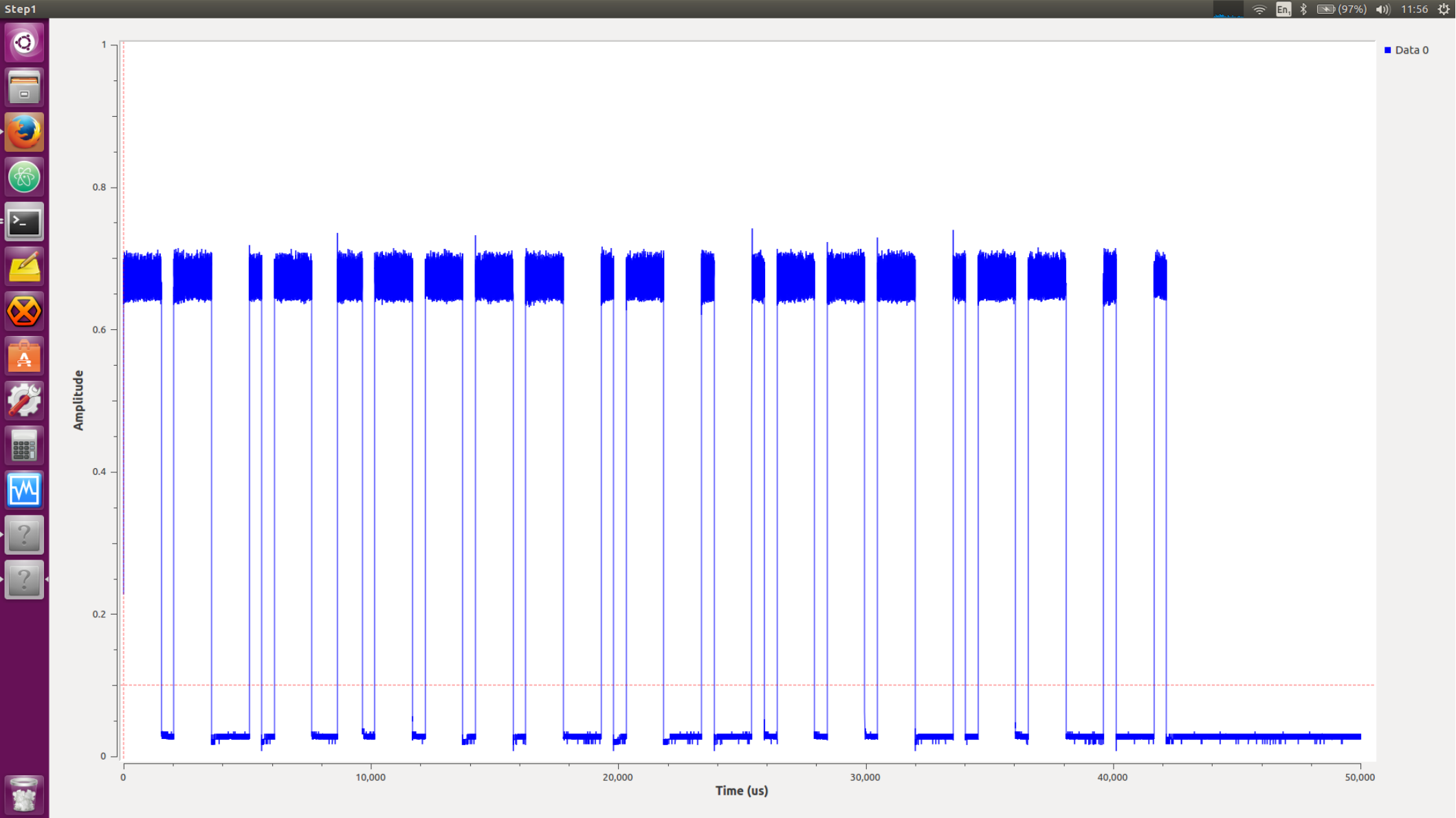
**QT GUI Time Sink**
**Number of Points:** 100k
**Sample Rate:** 2M
**Autoscale:** No

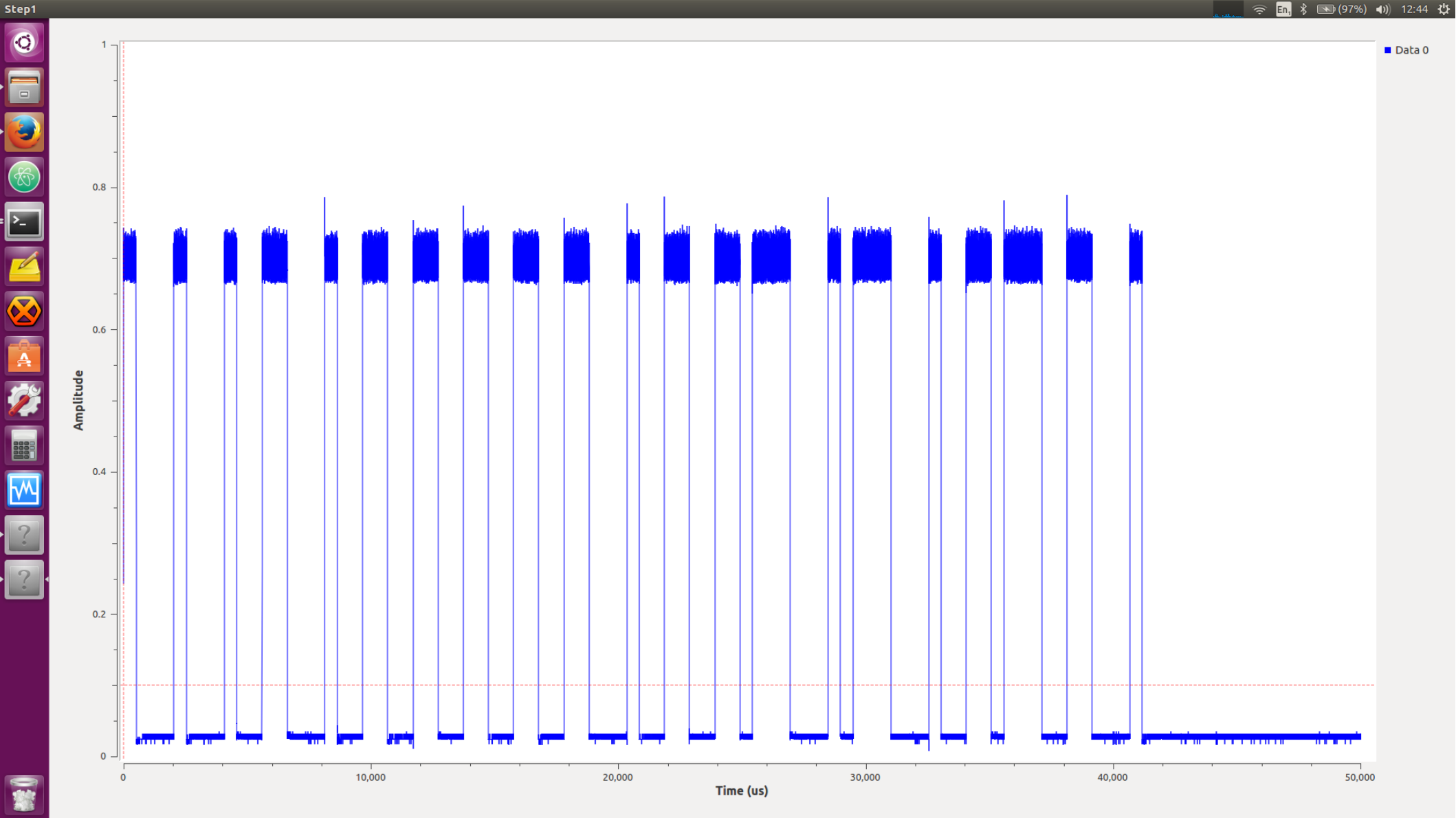
```
>>> Done

Loading: "/home/argilo/Documents/chamberlain/step1.grc"
>>> Done

Loading: "/home/argilo/Documents/chamberlain/step2.grc"
>>> Done
```

| Id | Value |
|---|---|
| Imports | |
| Variables | |
| samp_rate | 2000000 |

- Core
  - Audio
  - Boolean Operators
  - Byte Operators
  - Channelizers
  - Channel Models
  - Coding
  - Control Port
  - Debug Tools
  - Deprecated
  - Digital Television
  - Equalizers
  - Error Coding
  - FCD
  - File Operators
  - Filters
  - Fourier Analysis
  - GUI Widgets
  - Impairment Models
  - Instrumentation
  - Level Controllers
  - Math Operators
  - Measurement Tools
  - Message Tools
  - Misc
  - Modulators
  - Networking Tools
  - NOAA

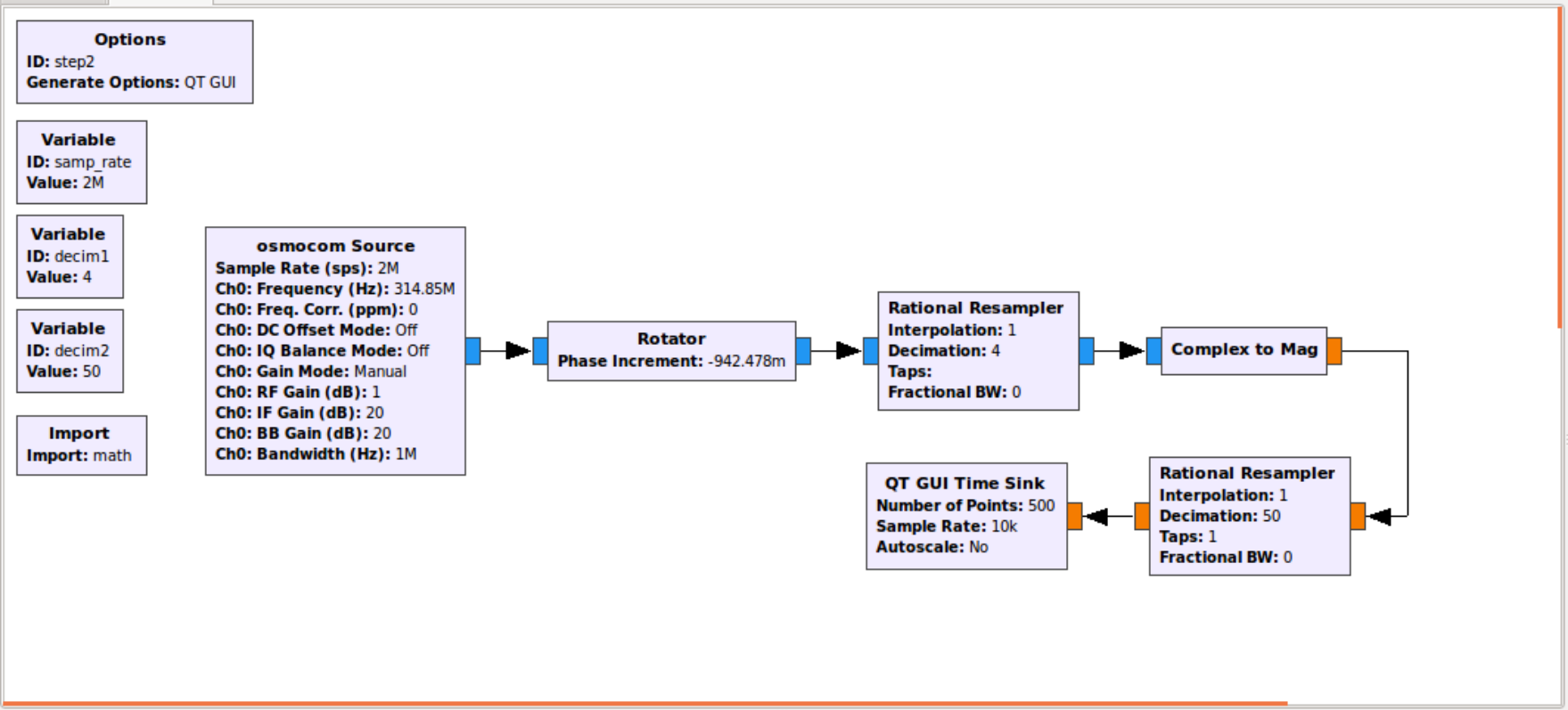

Step1
Data 0
Amplitude
Time (us)
0
0.2
0.4
0.6
0.8
1
0
200,000
400,000
600,000
800,000
1e+06


Step1
Data 0
Amplitude
Time (us)
0
0.2
0.4
0.6
0.8
1
0
10,000
20,000
30,000
40,000
50,000


Step1
Data 0
Amplitude
Time (us)
0
0.2
0.4
0.6
0.8
1
0
10,000
20,000
30,000
40,000
50,000

step2.grc - /home/argilo/Documents/chamberlain - GNU Radio Companion

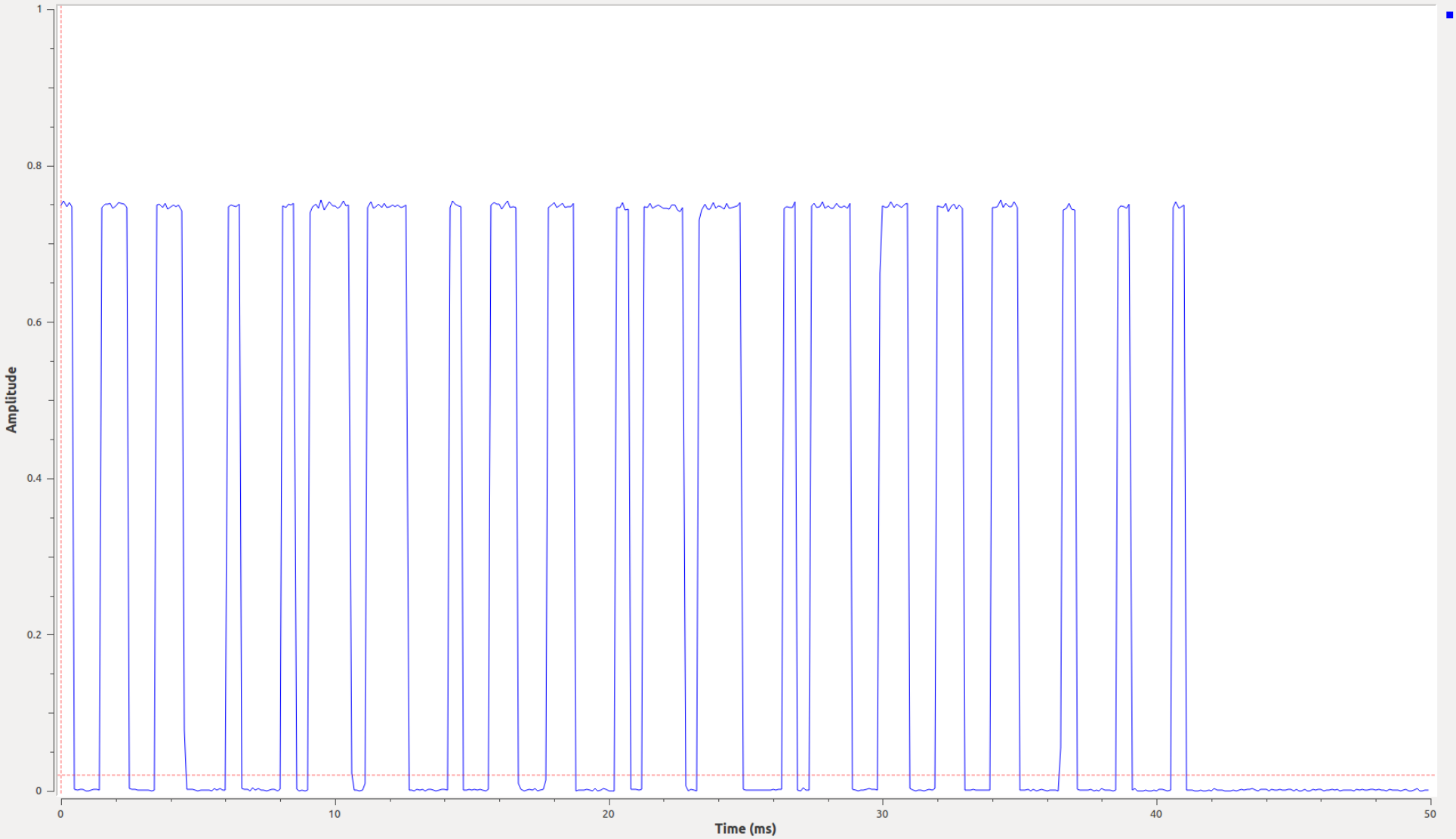
step1 step2




```
>>> Done

Loading: "/home/argilo/Documents/chamberlain/step1.grc"
>>> Done

Loading: "/home/argilo/Documents/chamberlain/step2.grc"
>>> Done
```

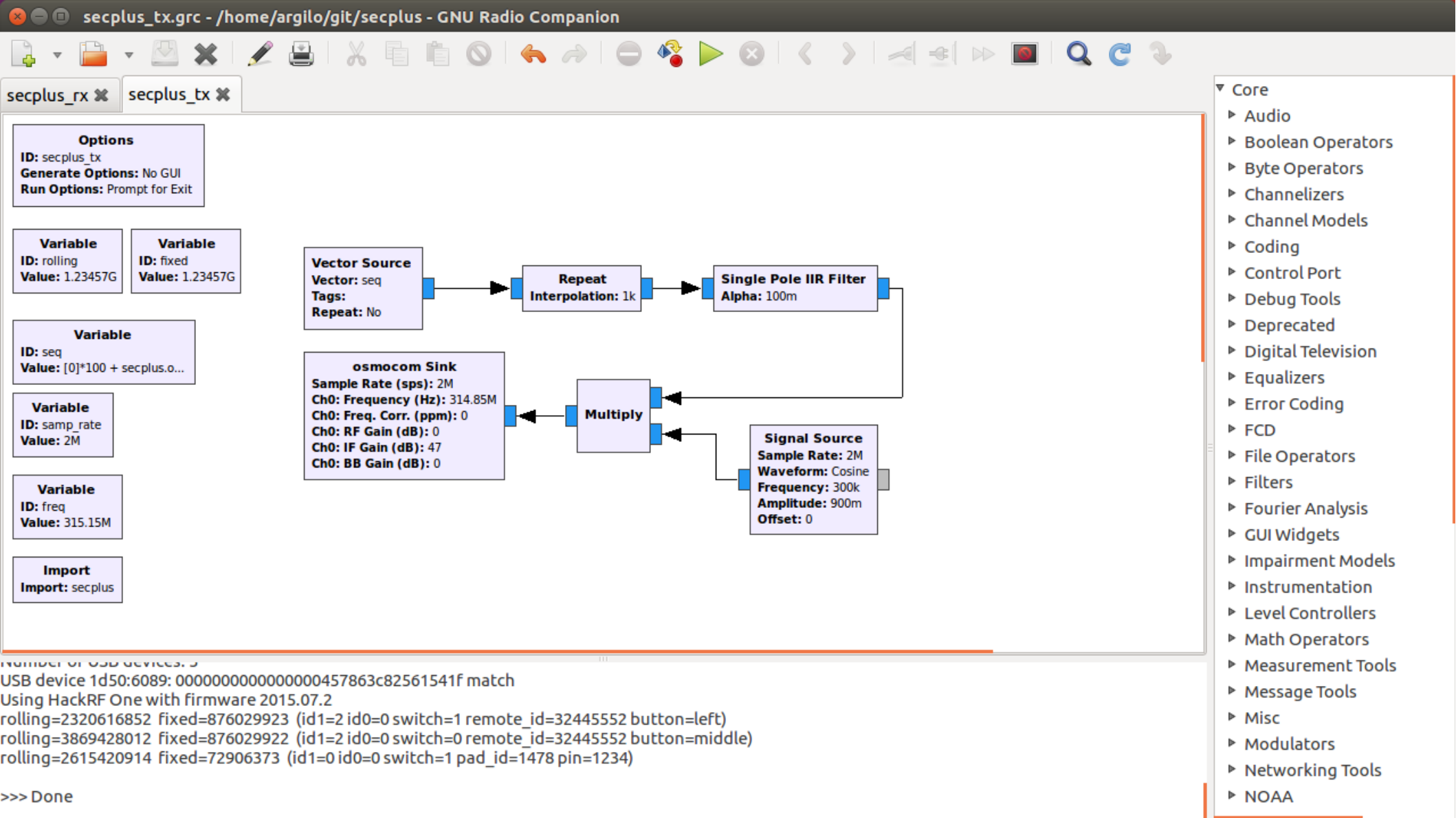
| Id | Value |
|---|---|
| Imports | |
| import_0 | import math |
| Variables | |
| decim1 | 4 |
| decim2 | 50 |

- Core
  - Audio
  - Boolean Operators
  - Byte Operators
  - Channelizers
  - Channel Models
  - Coding
  - Control Port
  - Debug Tools
  - Deprecated
  - Digital Television
  - Equalizers
  - Error Coding
  - FCD
  - File Operators
  - Filters
  - Fourier Analysis
  - GUI Widgets
  - Impairment Models
  - Instrumentation
  - Level Controllers
  - Math Operators
  - Measurement Tools
  - Message Tools
  - Misc
  - Modulators
  - Networking Tools
  - NOAA


Data 0
Amplitude
Time (ms)
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50

```
class blk(gr.sync_block):
    """Decoder for Chamberlain garage door openers"""

    def __init__(self, samp_rate=10000, threshold=0.02):
        gr.sync_block.__init__(
            self,
            name='Chamberlain Decoder',
            in_sig=[np.float32],
            out_sig=[]
        )
        self.samp_rate = samp_rate
        self.threshold = threshold
        self.last_sample = 0.0
        self.last_rise = 0
        self.buffer = []
        self.last_pair = []
        self.pair = []

    def work(self, input_items, output_items):
        for n, sample in enumerate(input_items[0]):
            current_sample = self.nitems_read(0) + n
            if self.last_sample < self.threshold and sample >= self.threshold:
                # rising edge
                self.last_rise = current_sample
            elif self.last_sample >= self.threshold and sample < self.threshold:
                # falling edge
                on_samples = current_sample - self.last_rise
                self.process_symbol(on_samples)
            if current_sample - self.last_rise > 3.25e-3 * self.samp_rate:
                self.buffer = []
            self.last_sample = sample
        return len(input_items[0])

    def process_symbol(self, on_samples):
```

```
USB device 1d50:6089: 0000000000000000457863c82561541f match
Using HackRF One with firmware 2015.07.2
rolling=2320616852 fixed=876029923 (id1=2 id0=0 switch=1 remote_id=32445552 button=left)
rolling=3869428012 fixed=876029922 (id1=2 id0=0 switch=0 remote_id=32445552 button=middle)
rolling=2615420914 fixed=72906373 (id1=0 id0=0 switch=1 pad_id=1478 pin=1234)

>>> Done
```


secplus_tx.grc - /home/argilo/git/secplus - GNU Radio Companion
secplus_rx
secplus_tx
Options
ID: secplus_tx
Generate Options: No GUI
Run Options: Prompt for Exit
Variable
ID: rolling
Value: 1.23457G
Variable
ID: fixed
Value: 1.23457G
Variable
ID: seq
Value: [0]*100 + secplus.o...
Variable
ID: samp_rate
Value: 2M
Variable
ID: freq
Value: 315.15M
Import
Import: secplus
Vector Source
Vector: seq
Tags:
Repeat: No
Repeat
Interpolation: 1k
Single Pole IIR Filter
Alpha: 100m
osmocom Sink
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 314.85M
Ch0: Freq. Corr. (ppm): 0
Ch0: RF Gain (dB): 0
Ch0: IF Gain (dB): 47
Ch0: BB Gain (dB): 0
Multiply
Signal Source
Sample Rate: 2M
Waveform: Cosine
Frequency: 300k
Amplitude: 900m
Offset: 0
USB device 1d50:6089: 0000000000000000457863c82561541f match
Using HackRF One with firmware 2015.07.2
rolling=2320616852 fixed=876029923 (id1=2 id0=0 switch=1 remote_id=32445552 button=left)
rolling=3869428012 fixed=876029922 (id1=2 id0=0 switch=0 remote_id=32445552 button=middle)
rolling=2615420914 fixed=72906373 (id1=0 id0=0 switch=1 pad_id=1478 pin=1234)
>>> Done
Core
Audio
Boolean Operators
Byte Operators
Channelizers
Channel Models
Coding
Control Port
Debug Tools
Deprecated
Digital Television
Equalizers
Error Coding
FCD
File Operators
Filters
Fourier Analysis
GUI Widgets
Impairment Models
Instrumentation
Level Controllers
Math Operators
Measurement Tools
Message Tools
Misc
Modulators
Networking Tools
NOAA

# argilo / secplus

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

A software implementation of the Security+ system used by garage door openers — Edit

6 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

**argilo** Use the slow transmitting mode (100ms per frame) just in case. Latest commit `21002b4` 2 hours ago

| | | |
|---|---|---|
| .gitignore | Initial commit. | 13 hours ago |
| README.md | Initial commit. | 13 hours ago |
| secplus.py | PEP8 formatting. | 4 hours ago |
| secplus_decode.py | Simplify slicing comparisons. | 3 hours ago |
| secplus_rx.grc | Simplify slicing comparisons. | 3 hours ago |
| secplus_rx.py | Simplify slicing comparisons. | 3 hours ago |
| secplus_tx.grc | Use the slow transmitting mode (100ms per frame) just in case. | 2 hours ago |
| secplus_tx.py | Use the slow transmitting mode (100ms per frame) just in case. | 2 hours ago |

## README.md

# Digital voice

README.md

build passing

# Digital Speech Decoder 1.7.0-dev

DSD is able to decode several digital voice formats from discriminator tap audio and synthesize the decoded speech. Speech synthesis requires mbelib, which is a separate package.

## Information

### The DSD Wiki

The DSD Wiki has lots of additional information about DSD including build instructions and answers to the most frequently asked questions. Please browse the Wiki after finishing this README.

https://github.com/szechyjs/dsd/wiki

### Bug reports

If you discover a problem with DSD, we would like to know about it. However, we ask that you please review these guidelines before submitting a bug report:

https://github.com/szechyjs/dsd/wiki/Bug-reports

Please don't contact developers directly through email or other channels. We ask that you follow the guidelines above and use the Issue Tracker.

### Contributing

We hope that you will consider contributing to DSD. Please read this short overview for some information about how to get started.

# argilo / gr-dsd





GNU Radio block for Digital Speech Decoder

Edit

Add topics

61 commits 3 branches 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

argilo Update to latest dsd. Latest commit ab4a739 on Oct 28, 2016

| | | |
|---|---|---|
| apps | Rebuild the module using gr_modtool. | a year ago |
| cmake | Rebuild the module using gr_modtool. | a year ago |
| docs | Rebuild the module using gr_modtool. | a year ago |
| dsd | Update to latest dsd. | a year ago |
| examples | Rebuild the module using gr_modtool. | a year ago |
| grc | Rebuild the module using gr_modtool. | a year ago |
| include/dsd | Rebuild the module using gr_modtool. | a year ago |
| lib | Rebuild the module using gr_modtool. | a year ago |
| mbelib | Update to latest mbelib. | a year ago |
| python | Rebuild the module using gr_modtool. | a year ago |
| swig | Rebuild the module using gr_modtool. | a year ago |
| .gitignore | Add some missing dependencies from the recommended list for Ubuntu 14.04 | 3 years ago |
| CMakeLists.txt | Update to latest dsd. | a year ago |

# Welcome to OP25

History

## NewsFlash

### World's Cheapest P25 Receiver

Balint has done some excellent work to get the $20USD Realtek RTL2832 DVB-T stick working with GNURadio. Take a look at the video to see him use the Realtek receiver together with OP25 to get the cheapest APCO P25 receiver (with DES-OFB support) you're ever likely to find. You should also check out the rtl-sdr page for more info.

### Check out the new presentations page

We have added a new presentation page to collect some of OP25-related presentations, talks and stuff we've done at RUXCON and elsewhere.

## OP25

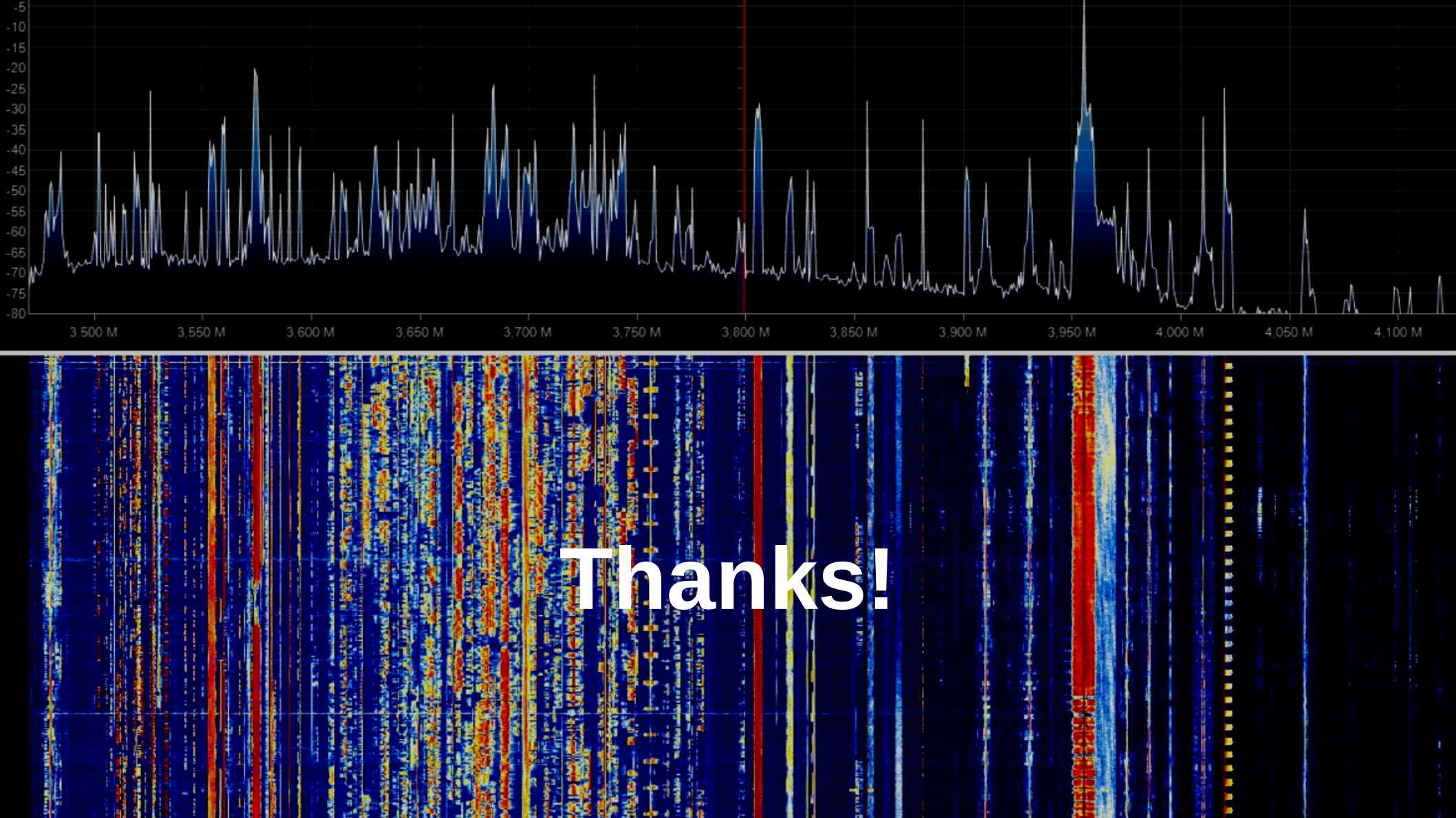
OP25 is a not-for-profit project to bring together folks that are interested in implementing APCO P25 using a software-defined radio. Our goal is to build a software-defined analyzer for APCO P25 signals that is available under the GNU Public License (GPL).

APCO Project 25 is the digital communications standard used by many police and emergency services throughout the world. Most notably the US, Canada and Australia deploy systems based on P25. Compared to existing analogue systems P25 offers improved spectrum use, coverage and flexibility. Provision is made to ensure the confidentiality of traffic, to allow the use of trunking and the provision of data in addition to voice services.

Hardware scanners such as the Uniden BCD996T offer APCO P25 functionality but software-defined radio (SDR) offers significantly improved flexibility. For example, software radio approaches can receive many channels at once, handle both voice and data (including the trunking control channel), decrypt encrypted traffic when the key is known and log

Wiki

Start page
Index by title
Index by date


-5
-10
-15
-20
-25
-30
-35
-40
-45
-50
-55
-60
-65
-70
-75
-80
3,500 M
3,550 M
3,600 M
3,650 M
3,700 M
3,750 M
3,800 M
3,850 M
3,900 M
3,950 M
4,000 M
4,050 M
4,100 M
Thanks!