Reverse Engineering Radio Signals

Zezadas

Currently working at S21 SEC

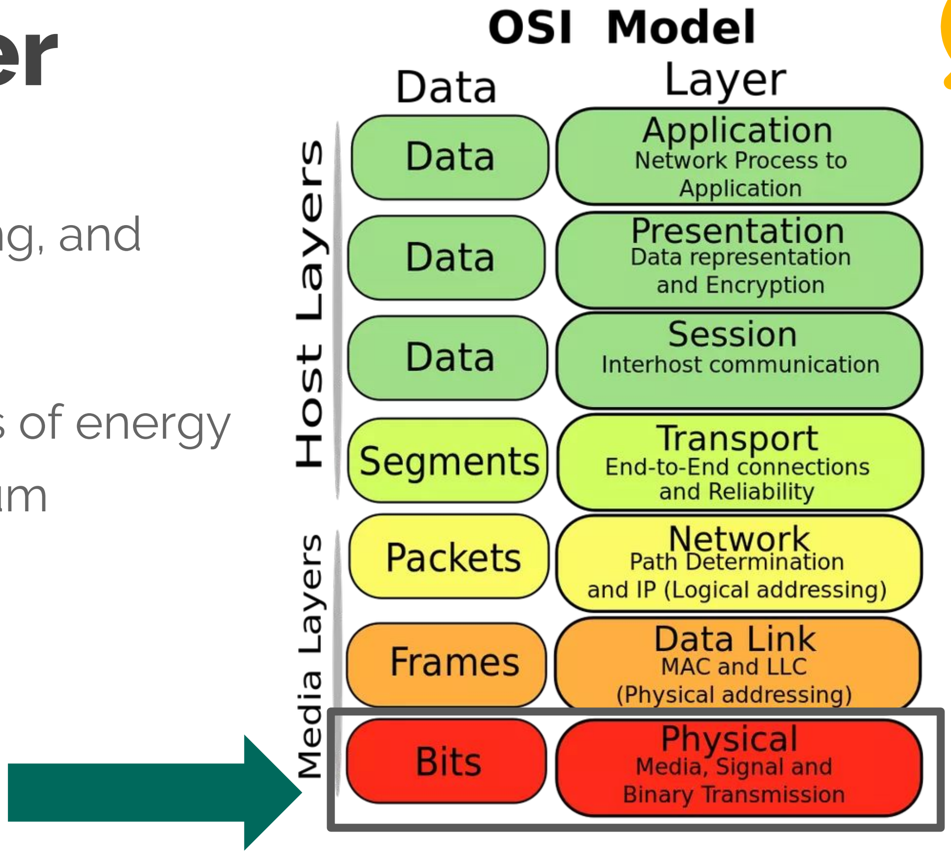
https://peidei.me

https://sefod.eu

@0xz3z4d45

Physical Layer

- **In wire** - Voltage, timing, and wiring defining 1s and 0s
- **In wireless** - Patterns of energy being sent over RF medium



OSI Model
Data
Layer
Host Layers
Data
Application
Network Process to Application
Data
Presentation
Data representation and Encryption
Data
Session
Interhost communication
Segments
Transport
End-to-End connections and Reliability
Media Layers
Packets
Network
Path Determination and IP (Logical addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal and Binary Transmission

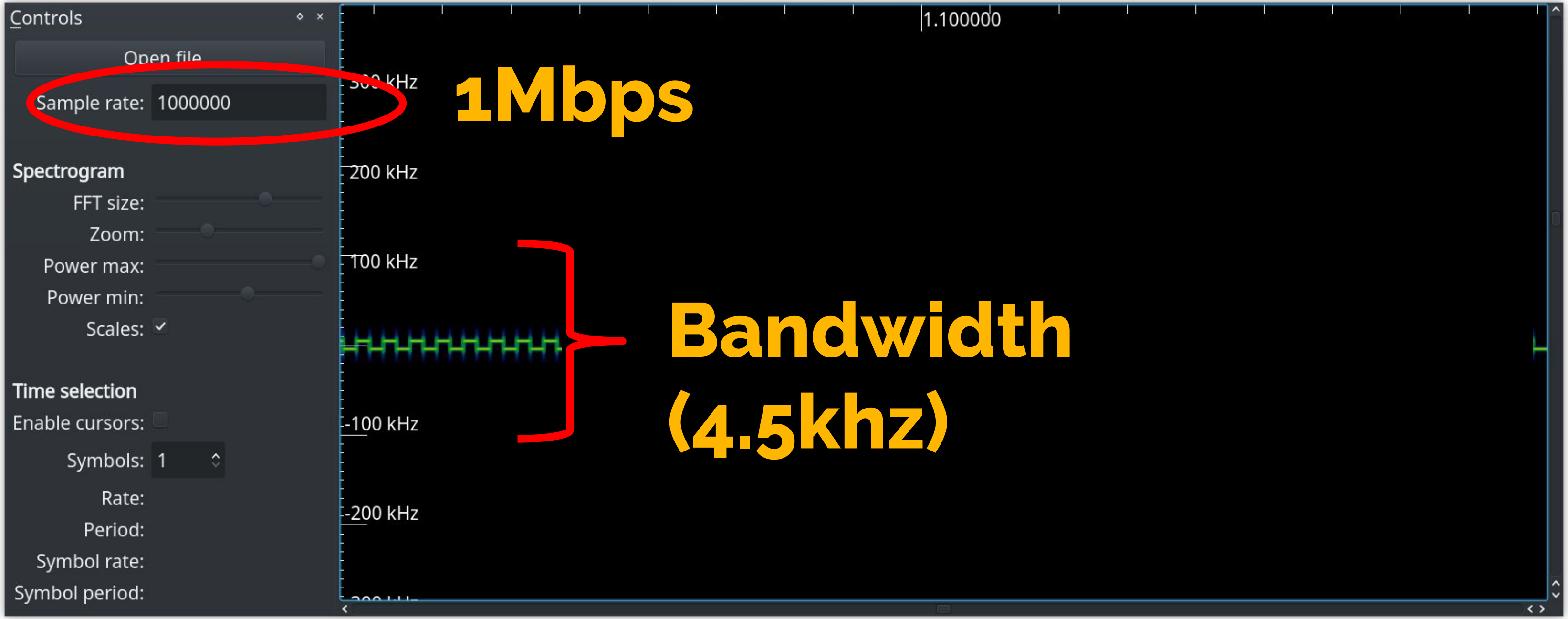
RF Reverse Engineering Methodology

- OSINT
- Characterize the channel
- Identify Modulation
- Determine Symbol Rate
- Synchronize
- Extract Symbols

Challenge Details:

- **Sample rate -** 1Mbps
- **Frequency -** 153.350 Mhz

Inspectrum:



Sample rate: 1000000

Scales: ☑

Enable cursors: □

Symbols: 1

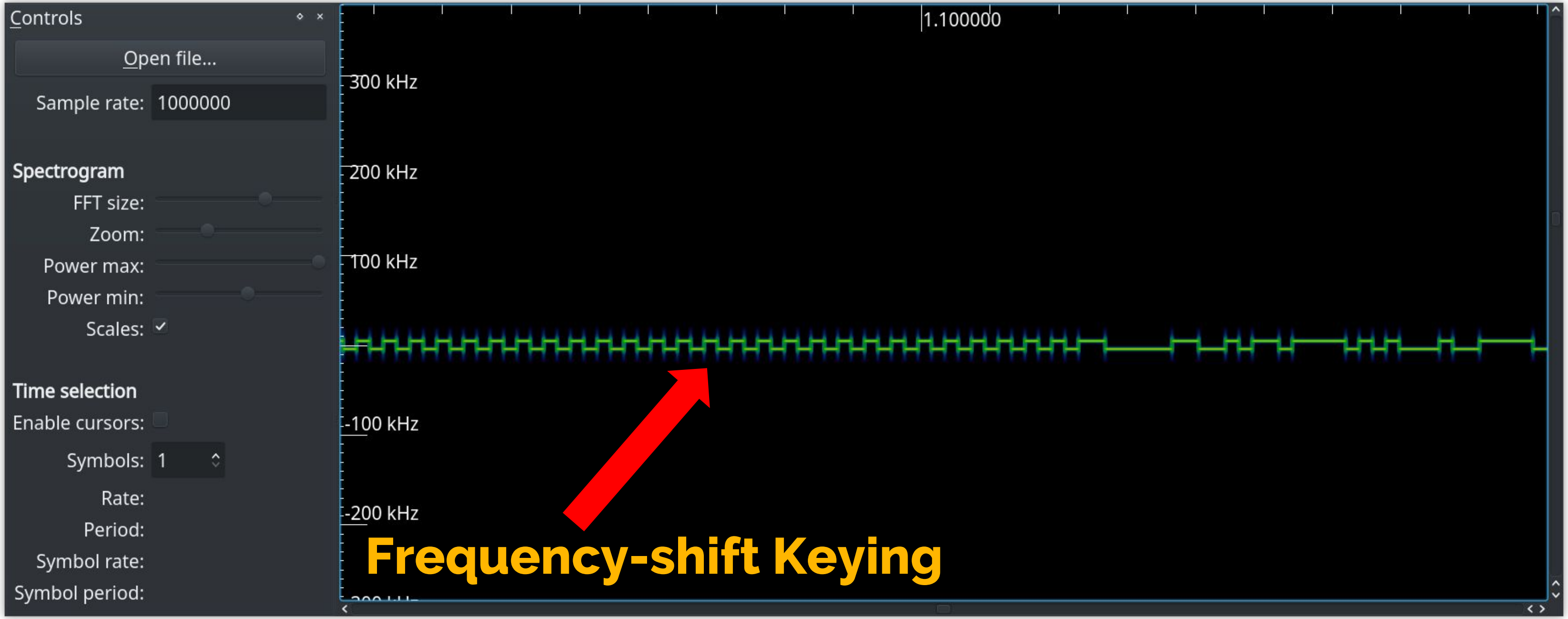
1Mbps

Bandwidth (4.5khz)

RF Reverse Engineering Methodology

- OSINT
- ✓ Characterize the channel
- Identify Modulation
- Determine Symbol Rate
- Synchronize
- Extract Symbols

Inspectrum:



Controls

Open file...

Sample rate: 1000000

Spectrogram

FFT size:

Zoom:

Power max:

Power min:

Scales: ✓

Time selection

Enable cursors: □

Symbols: 1

Rate:

Period:

Symbol rate:

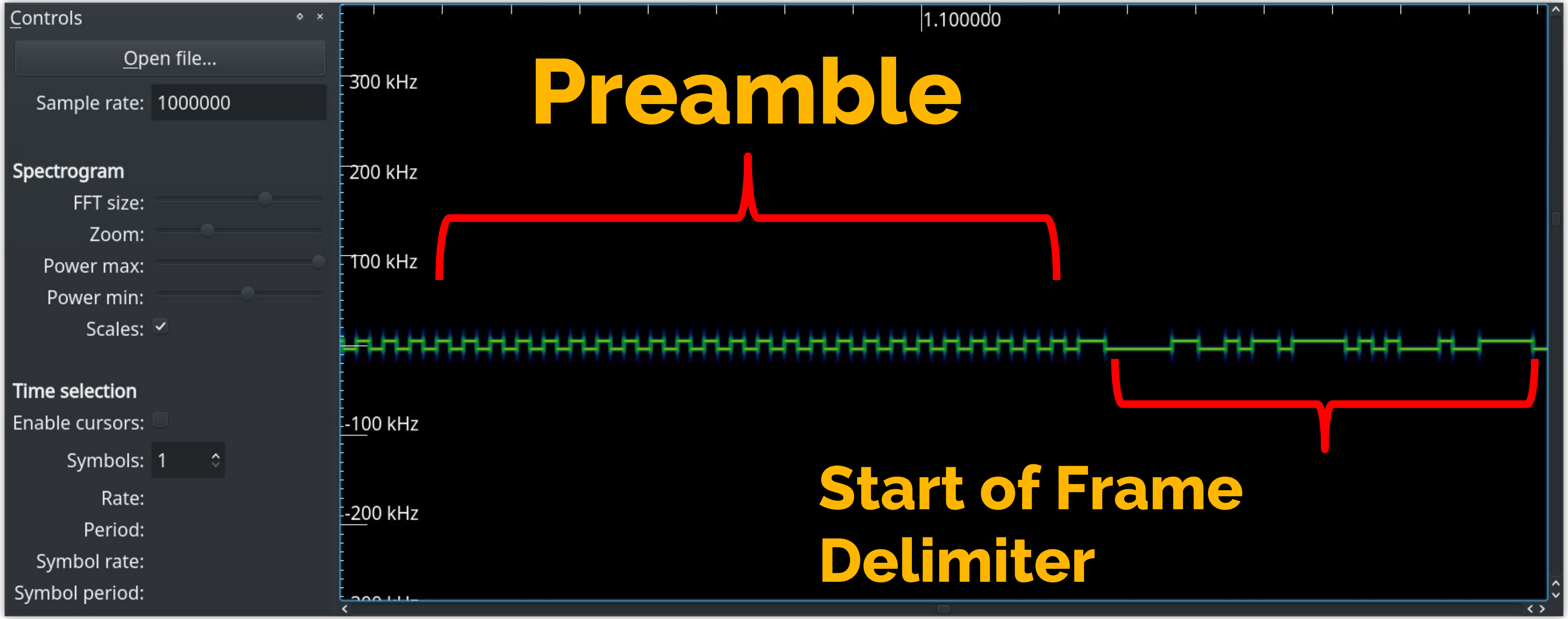
Symbol period:

Frequency-shift Keying

RF Reverse Engineering Methodology

- OSINT
- ✓ Characterize the channel
- ✓ Identify Modulation
- Determine Symbol Rate
- Synchronize
- Extract Symbols

Inspectrum:



Preamble

Start of Frame Delimiter

Sample rate: 1000000

Scales: ✓

Enable cursors: □

Symbols: 1

RF Reverse Engineering Methodology

OSINT

✓ Characterize the channel

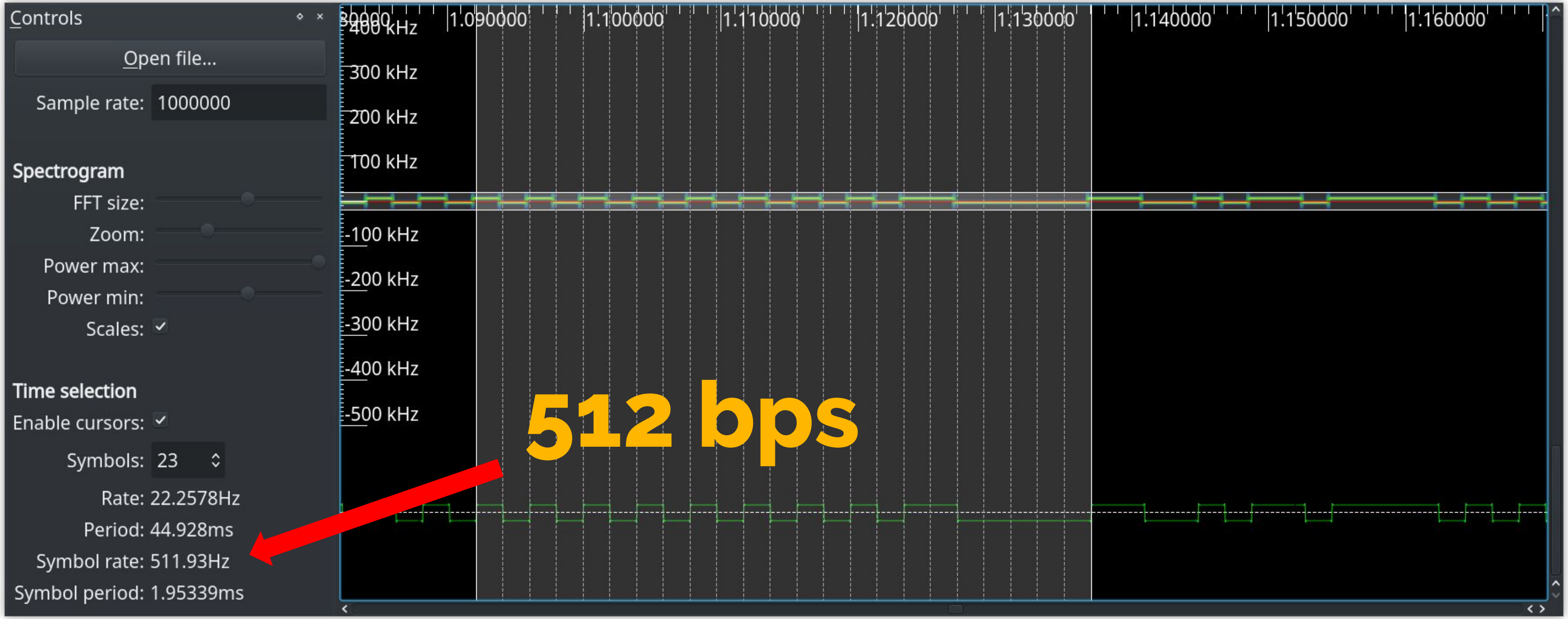
✓ Identify Modulation

Determine Symbol Rate

✓ Synchronize

Extract Symbols

Inspectrum:



Controls

Open file...

Sample rate: 1000000

Spectrogram

FFT size:

Zoom:

Power max:

Power min:

Scales: ✓

Time selection

Enable cursors: ✓

Symbols: 23

Rate: 22.2578Hz

Period: 44.928ms

Symbol rate: 511.93Hz

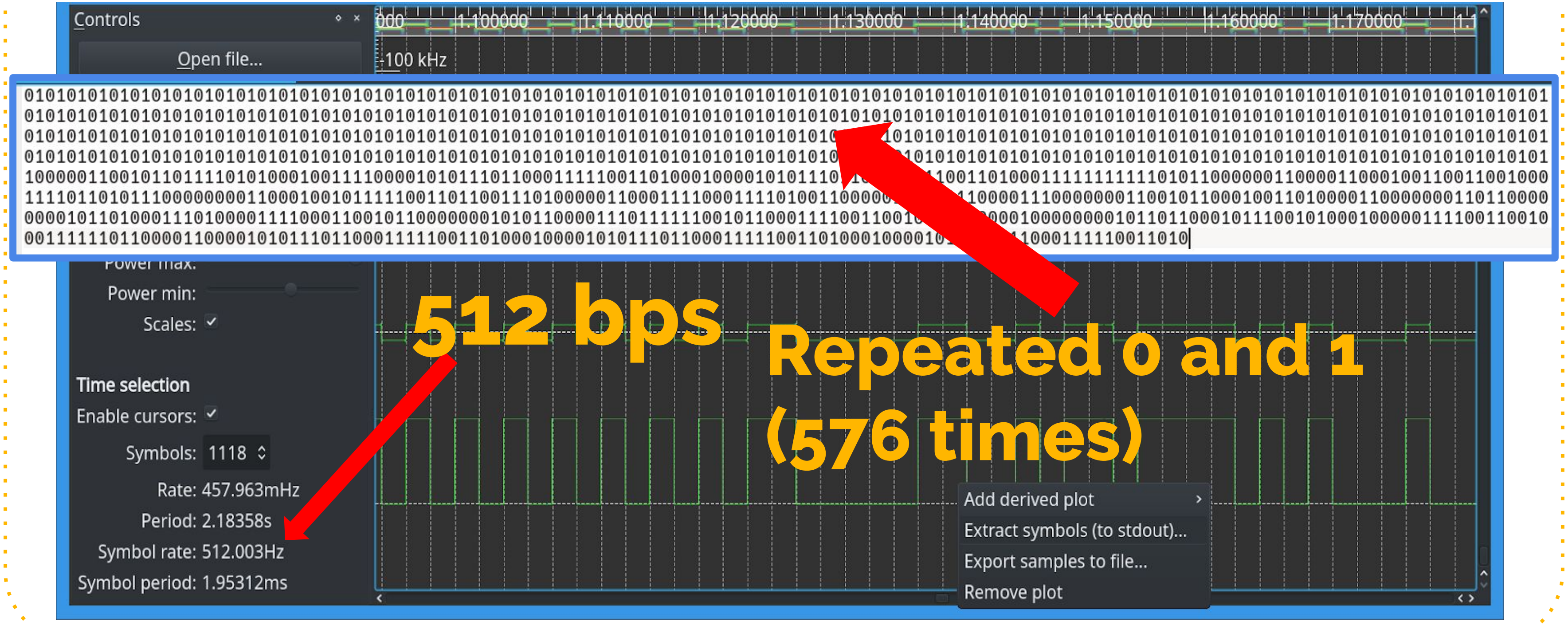
Symbol period: 1.95339ms

512 bps

RF Reverse Engineering Methodology

- OSINT
- ✓ Characterize the channel
- ✓ Identify Modulation
- ✓ Determine Symbol Rate
- ✓ Synchronize
- Extract Symbols

Inspectrum:



```
010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101
01010101010101010101010101010101010101010101010101010101010101010101010101010    101010101010101010101010101010101010101010101010101010101010101
01010101010101010101010101010101010101010101010101010101010101010101010101010     01010101010101010101010101010101010101010101010101010101010101
100000110010110111101010001001111000001010111011000111110011010001000010101110 10    110011010001111111111101011000000110000110001001100110010000
11110110101110000000001100010010111110011011001110100000011000111100011110100110000    01100001110000000011001011000100110100001100000001101 10000
0000101101000111010000111100011001011000000001010110000111011111110010110001111001100 10     0000100000000010110110001011100101000100000011110011001 0
00111111101100001100000101011101100011111001101000100001010111011000111110011010001000010101   11000111110011010
```

Power max:

Power min:

Scales: ✓

Time selection

Enable cursors: ✓

Symbols: 1118

Rate: 457.963mHz

Period: 2.18358s

Symbol rate: 512.003Hz

Symbol period: 1.95312ms

512 bps

Repeated 0 and 1 (576 times)

RF Reverse Engineering Methodology

OSINT

✓ Characterize the channel

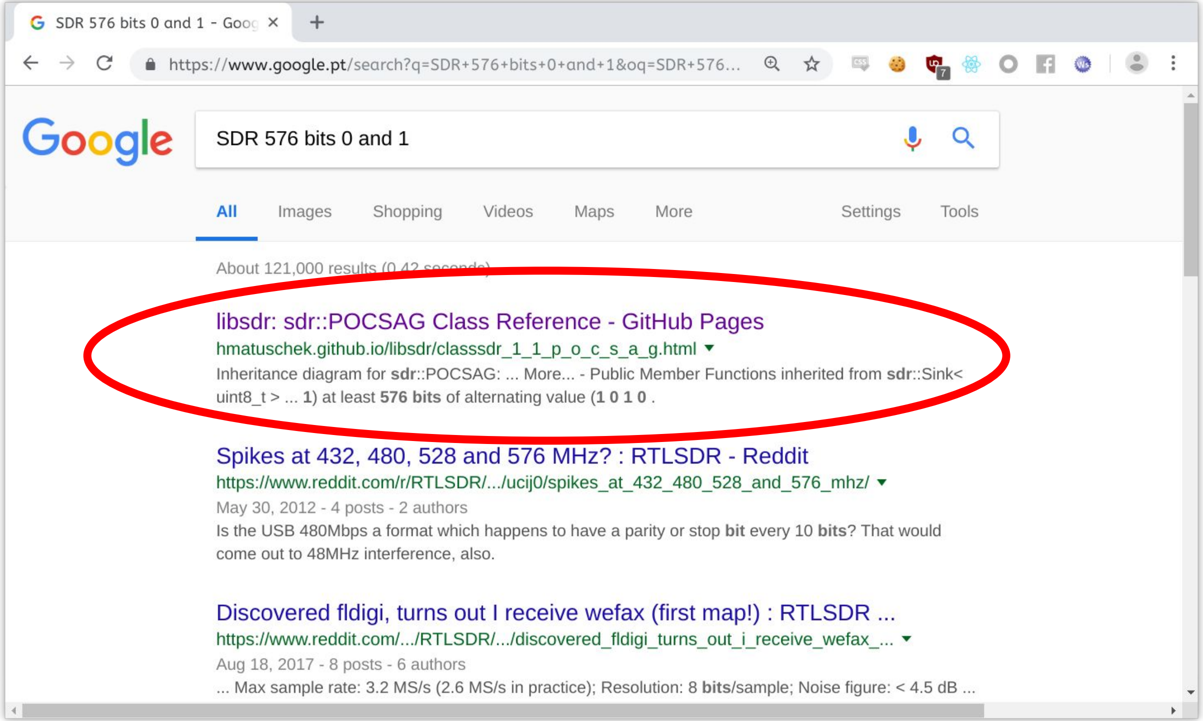
✓ Identify Modulation

✓ Determine Symbol Rate

✓ Synchronize

✓ Extract Symbols

OSINT:


SDR 576 bits 0 and 1 - Goog
https://www.google.pt/search?q=SDR+576+bits+0+and+1&oq=SDR+576...
Google
SDR 576 bits 0 and 1
All
Images
Shopping
Videos
Maps
More
Settings
Tools

About 121,000 results (0.42 seconds)

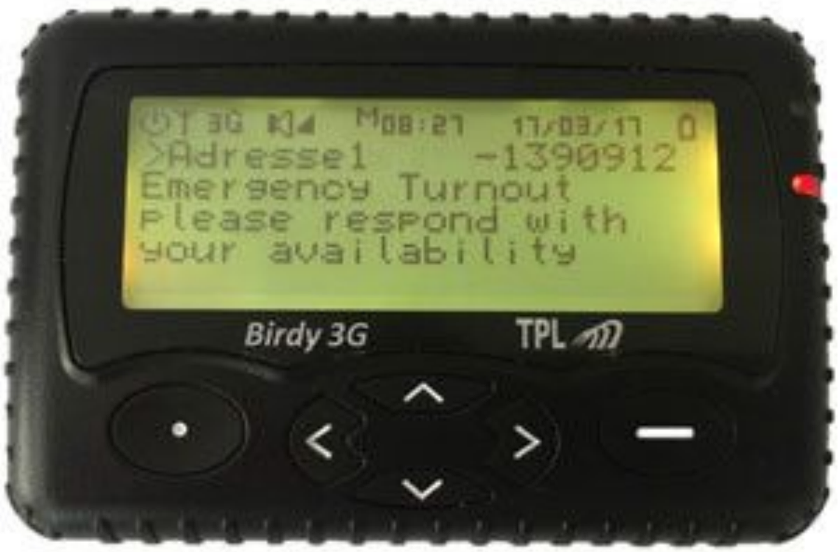
libsdr: sdr::POCSAG Class Reference - GitHub Pages
hmatuschek.github.io/libsdr/classsdr_1_1_p_o_c_s_a_g.html
Inheritance diagram for **sdr**::POCSAG: ... More... - Public Member Functions inherited from **sdr**::Sink< uint8_t > ... **1**) at least **576 bits** of alternating value (**1 0 1 0** .

Spikes at 432, 480, 528 and 576 MHz? : RTLSDR - Reddit
https://www.reddit.com/r/RTLSDR/.../ucij0/spikes_at_432_480_528_and_576_mhz/
May 30, 2012 - 4 posts - 2 authors
Is the USB 480Mbps a format which happens to have a parity or stop **bit** every 10 **bits**? That would come out to 48MHz interference, also.

Discovered fldigi, turns out I receive wefax (first map!) : RTLSDR ...
https://www.reddit.com/.../RTLSDR/.../discovered_fldigi_turns_out_i_receive_wefax_...
Aug 18, 2017 - 8 posts - 6 authors
... Max sample rate: 3.2 MS/s (2.6 MS/s in practice); Resolution: 8 **bits**/sample; Noise figure: < 4.5 dB ...

POCSAG:

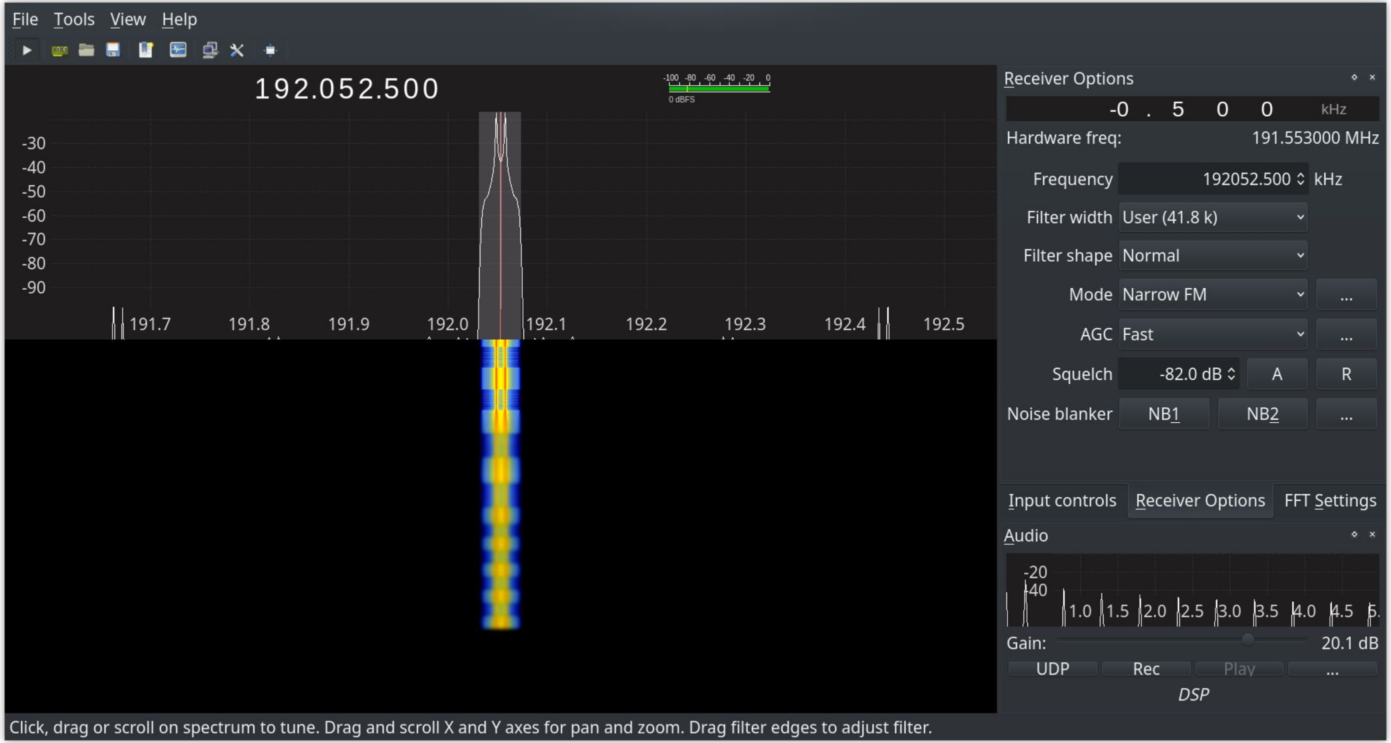
Pager



RF Reverse Engineering Methodology

- ✓ OSINT
- ✓ Characterize the channel
- ✓ Identify Modulation
- ✓ Determine Symbol Rate
- ✓ Synchronize
- ✓ Extract Symbols

Decode:



Receiver Options

-0 . 5 0 0 kHz

Hardware freq: 191.553000 MHz

Frequency 192052.500 kHz

Filter width User (41.8 k)

Filter shape Normal

Mode Narrow FM

AGC Fast

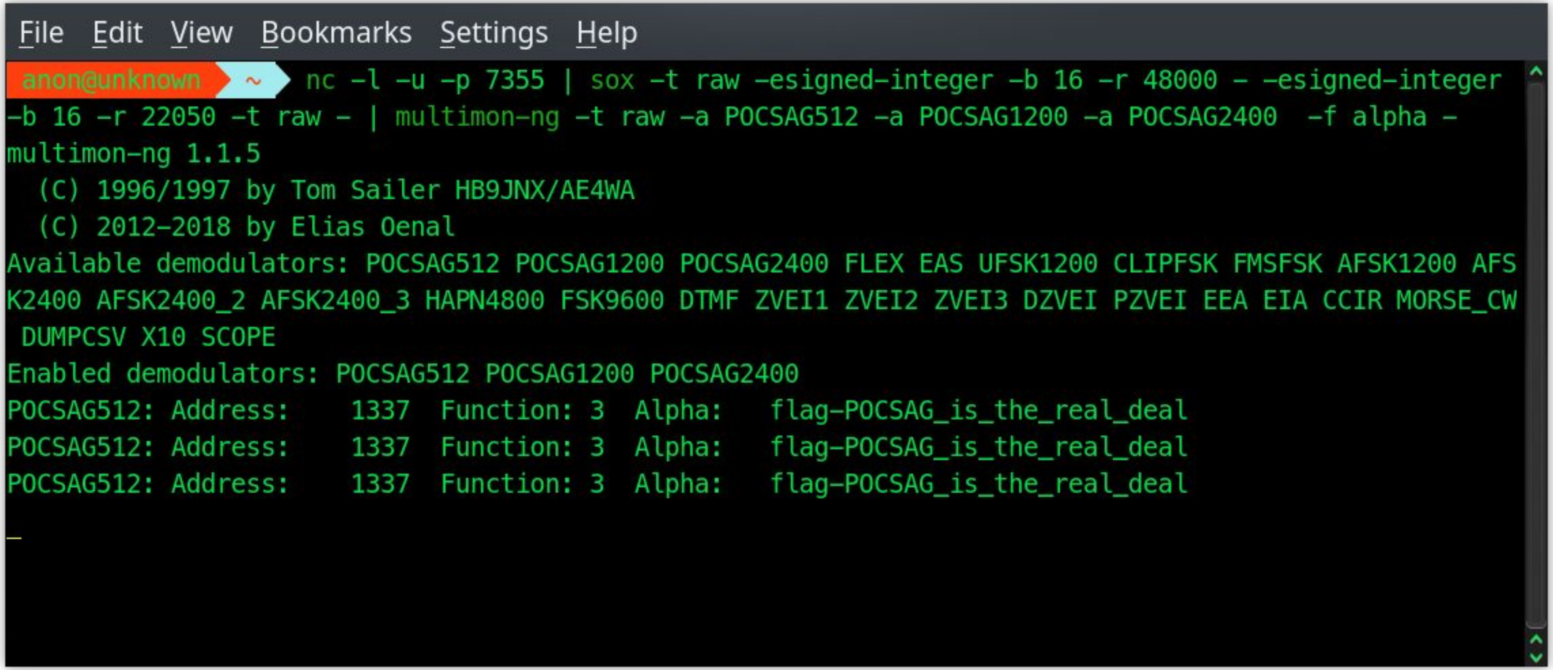
Squelch -82.0 dB

Noise blanker NB1 NB2

Gain: 20.1 dB

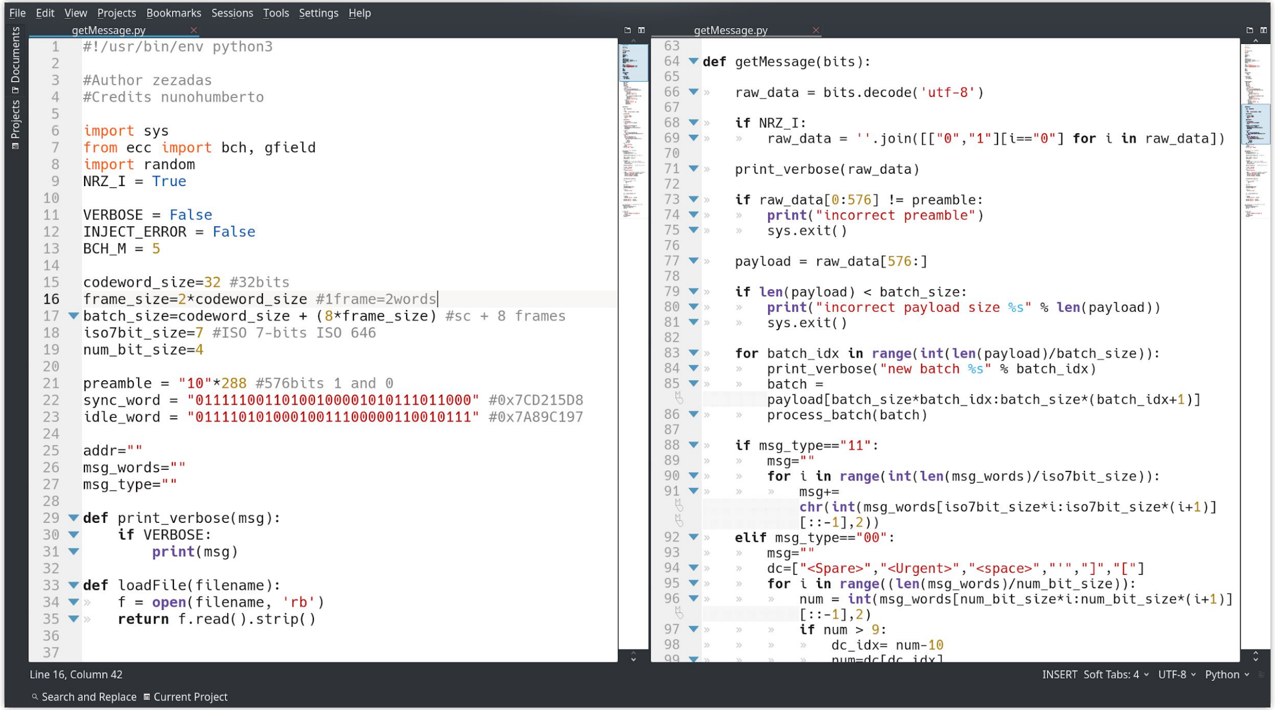
Click, drag or scroll on spectrum to tune. Drag and scroll X and Y axes for pan and zoom. Drag filter edges to adjust filter.

Decode:



```
anon@unknown  ~  nc -l -u -p 7355 | sox -t raw -esigned-integer -b 16 -r 48000 - -esigned-integer
-b 16 -r 22050 -t raw - | multimon-ng -t raw -a POCSAG512 -a POCSAG1200 -a POCSAG2400  -f alpha -
multimon-ng 1.1.5
  (C) 1996/1997 by Tom Sailer HB9JNX/AE4WA
  (C) 2012-2018 by Elias Oenal
Available demodulators: POCSAG512 POCSAG1200 POCSAG2400 FLEX EAS UFSK1200 CLIPFSK FMSFSK AFSK1200 AFS
K2400 AFSK2400_2 AFSK2400_3 HAPN4800 FSK9600 DTMF ZVEI1 ZVEI2 ZVEI3 DZVEI PZVEI EEA EIA CCIR MORSE_CW
 DUMPCSV X10 SCOPE
Enabled demodulators: POCSAG512 POCSAG1200 POCSAG2400
POCSAG512: Address:    1337  Function: 3  Alpha:   flag-POCSAG_is_the_real_deal
POCSAG512: Address:    1337  Function: 3  Alpha:   flag-POCSAG_is_the_real_deal
POCSAG512: Address:    1337  Function: 3  Alpha:   flag-POCSAG_is_the_real_deal

_
```

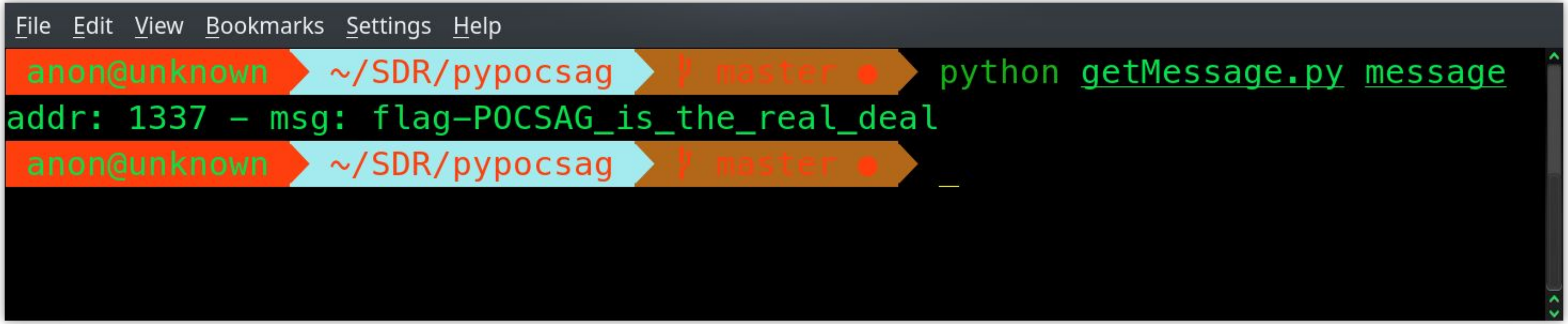
Python POCSAG Decoder:


File Edit View Projects Bookmarks Sessions Tools Settings Help
Documents
Projects
getMessage.py
getMessage.py
Line 16, Column 42
INSERT Soft Tabs: 4 UTF-8 Python
Search and Replace
Current Project

```python
#!/usr/bin/env python3

#Author zezadas
#Credits nunohumberto

import sys
from ecc import bch, gfield
import random
NRZ_I = True

VERBOSE = False
INJECT_ERROR = False
BCH_M = 5

codeword_size=32 #32bits
frame_size=2*codeword_size #1frame=2words
batch_size=codeword_size + (8*frame_size) #sc + 8 frames
iso7bit_size=7 #ISO 7-bits ISO 646
num_bit_size=4

preamble = "10"*288 #576bits 1 and 0
sync_word = "01111100110100100001010111011000" #0x7CD215D8
idle_word = "01111010100010011100000110010111" #0x7A89C197

addr=""
msg_words=""
msg_type=""

def print_verbose(msg):
    if VERBOSE:
        print(msg)

def loadFile(filename):
    f = open(filename, 'rb')
    return f.read().strip()

```

```python

def getMessage(bits):

    raw_data = bits.decode('utf-8')

    if NRZ_I:
        raw_data = ''.join([["0","1"][i=="0"] for i in raw_data])

    print_verbose(raw_data)

    if raw_data[0:576] != preamble:
        print("incorrect preamble")
        sys.exit()

    payload = raw_data[576:]

    if len(payload) < batch_size:
        print("incorrect payload size %s" % len(payload))
        sys.exit()

    for batch_idx in range(int(len(payload)/batch_size)):
        print_verbose("new batch %s" % batch_idx)
        batch =
        payload[batch_size*batch_idx:batch_size*(batch_idx+1)]
        process_batch(batch)

    if msg_type=="11":
        msg=""
        for i in range(int(len(msg_words)/iso7bit_size)):
            msg+=
            chr(int(msg_words[iso7bit_size*i:iso7bit_size*(i+1)]
            [::-1],2))
    elif msg_type=="00":
        msg=""
        dc=["<Spare>","<Urgent>","<space>","'","]","["]
        for i in range((len(msg_words)/num_bit_size)):
            num = int(msg_words[num_bit_size*i:num_bit_size*(i+1)]
            [::-1],2)
            if num > 9:
                dc_idx= num-10
                num=dc[dc_idx]
```

Python POCSAG Decoder:

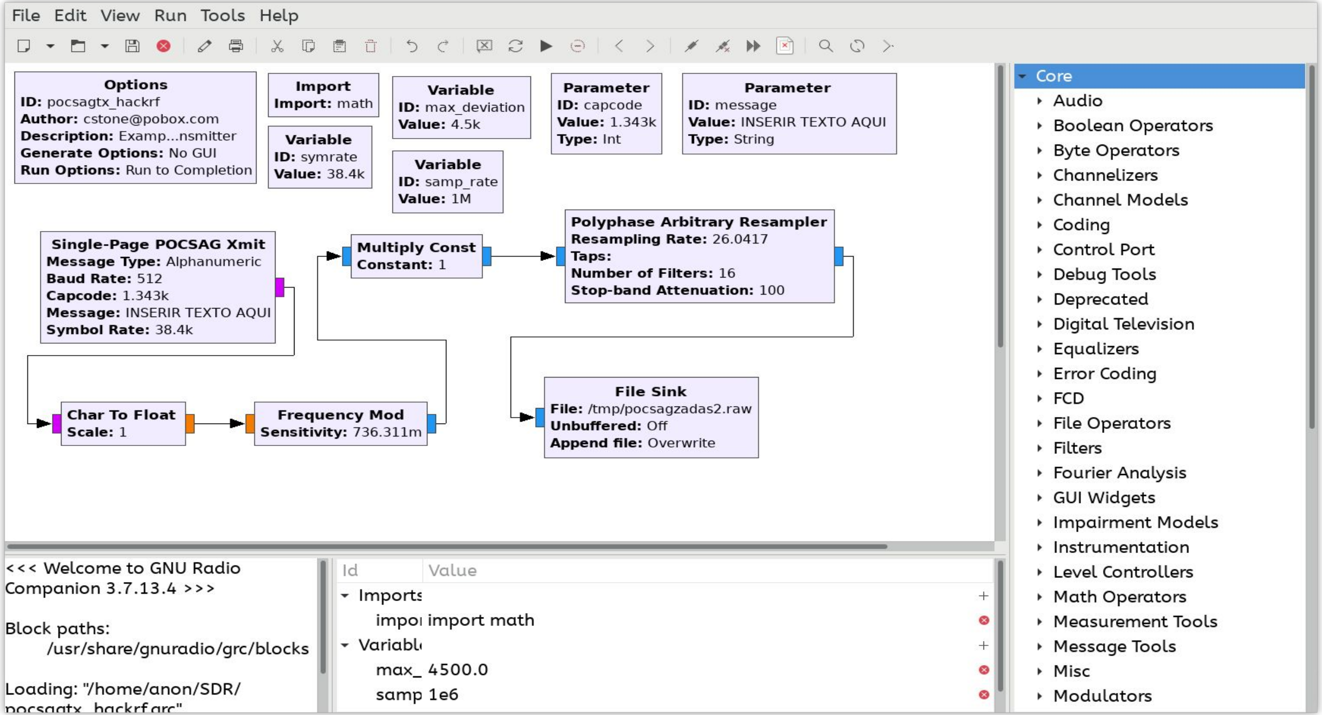


```
anon@unknown  ~/SDR/pypocsag  master ●  python getMessage.py message
addr: 1337 - msg: flag-POCSAG_is_the_real_deal
anon@unknown  ~/SDR/pypocsag  master ●  _
```

https://github.com/zezadas/pypocsag

Demo

Create Sample:



Options
ID: pocsagtx_hackrf
Author: cstone@pobox.com
Description: Examp...nsmitter
Generate Options: No GUI
Run Options: Run to Completion

Import
Import: math

Variable
ID: max_deviation
Value: 4.5k

Parameter
ID: capcode
Value: 1.343k
Type: Int

Parameter
ID: message
Value: INSERIR TEXTO AQUI
Type: String

Variable
ID: symrate
Value: 38.4k

Variable
ID: samp_rate
Value: 1M

Single-Page POCSAG Xmit
Message Type: Alphanumeric
Baud Rate: 512
Capcode: 1.343k
Message: INSERIR TEXTO AQUI
Symbol Rate: 38.4k

Multiply Const
Constant: 1

Polyphase Arbitrary Resampler
Resampling Rate: 26.0417
Taps:
Number of Filters: 16
Stop-band Attenuation: 100

Char To Float
Scale: 1

Frequency Mod
Sensitivity: 736.311m

File Sink
File: /tmp/pocsagzadas2.raw
Unbuffered: Off
Append file: Overwrite

```
<<< Welcome to GNU Radio
Companion 3.7.13.4 >>>

Block paths:
        /usr/share/gnuradio/grc/blocks

Loading: "/home/anon/SDR/
pocsagtx_hackrf.grc"
```

Id	Value
Imports	
impo	import math
Variabl	
max_	4500.0
samp	1e6

Winner(s)

1. Nuno Humberto

Assets

- inspectrum (https://github.com/miek/inspectrum)
- gqrx (https://github.com/csete/gqrx)
- https://www.fontenay-ronan.fr/decoding-pocsag-on-ubuntu-with-a-rtl-sdr-dongle/
- pypocsag (https://github.com/zezadas/pypocsag)
- gr-mixalot (https://github.com/unsynchronized/gr-mixalot)
- gr-mixalot arch package (https://aur.archlinux.org/packages/gr-mixalot-git/)
- Reverse Signals - https://www.youtube.com/watch?v=QeoGQwT0Z1Y

Thanks!

🌐 https://peidei.me

🌐 https://sefod.eu

@0xz3z4d45