



PERÚ

Ministerio
del Ambiente

Instituto
Geofísico del Perú - IGP

Dirección
Científica

Radio Observatorio
de Jicamarca

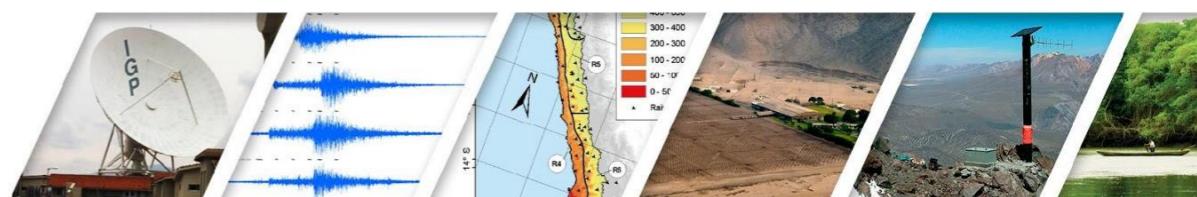


MIT Haystack Observatory -Program Research

USRP implementation of multi static meteor radar transmit and receive applications

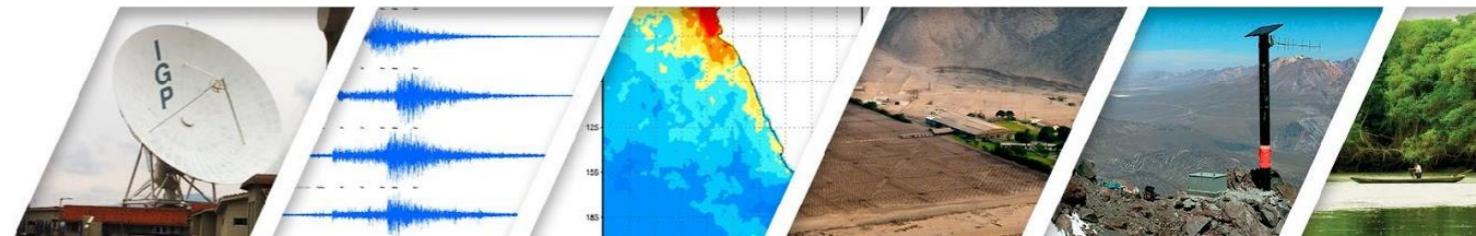
Alexander Valdez Portocarrero

July 22 2019



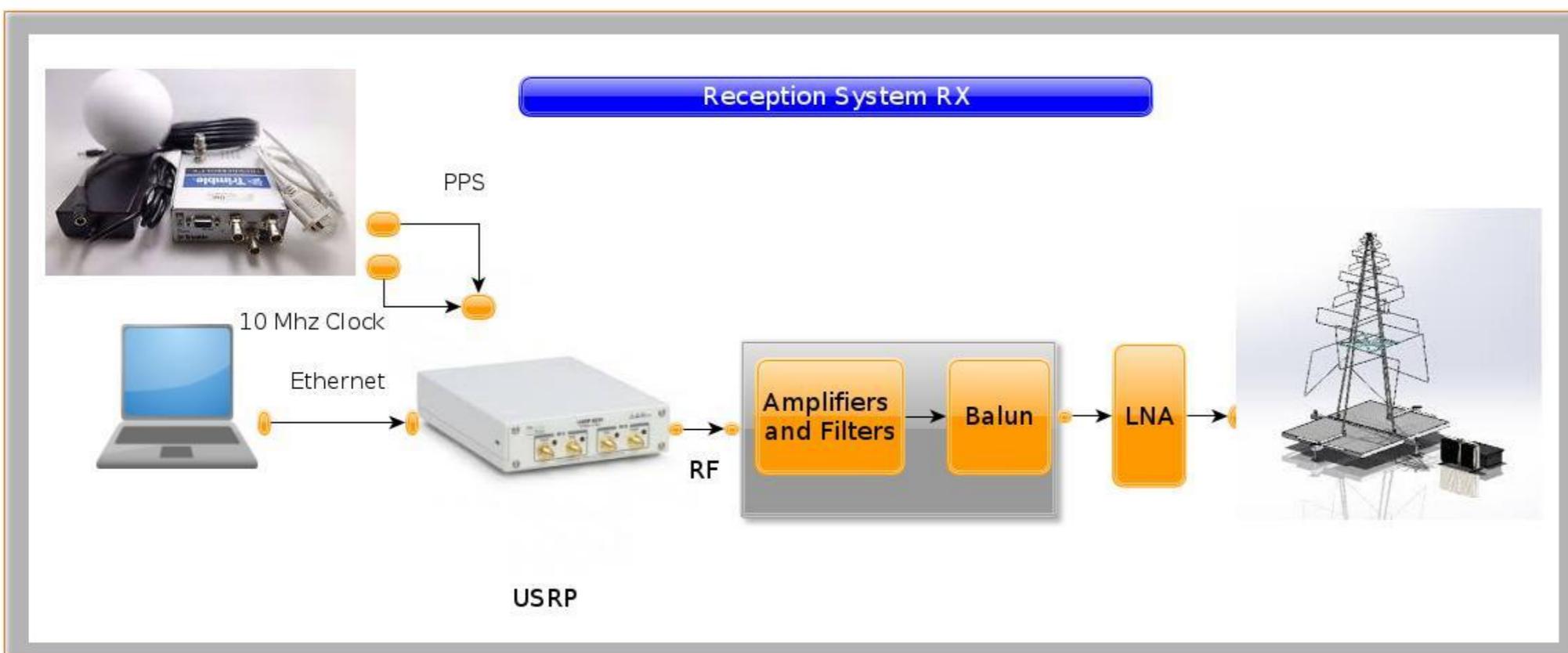
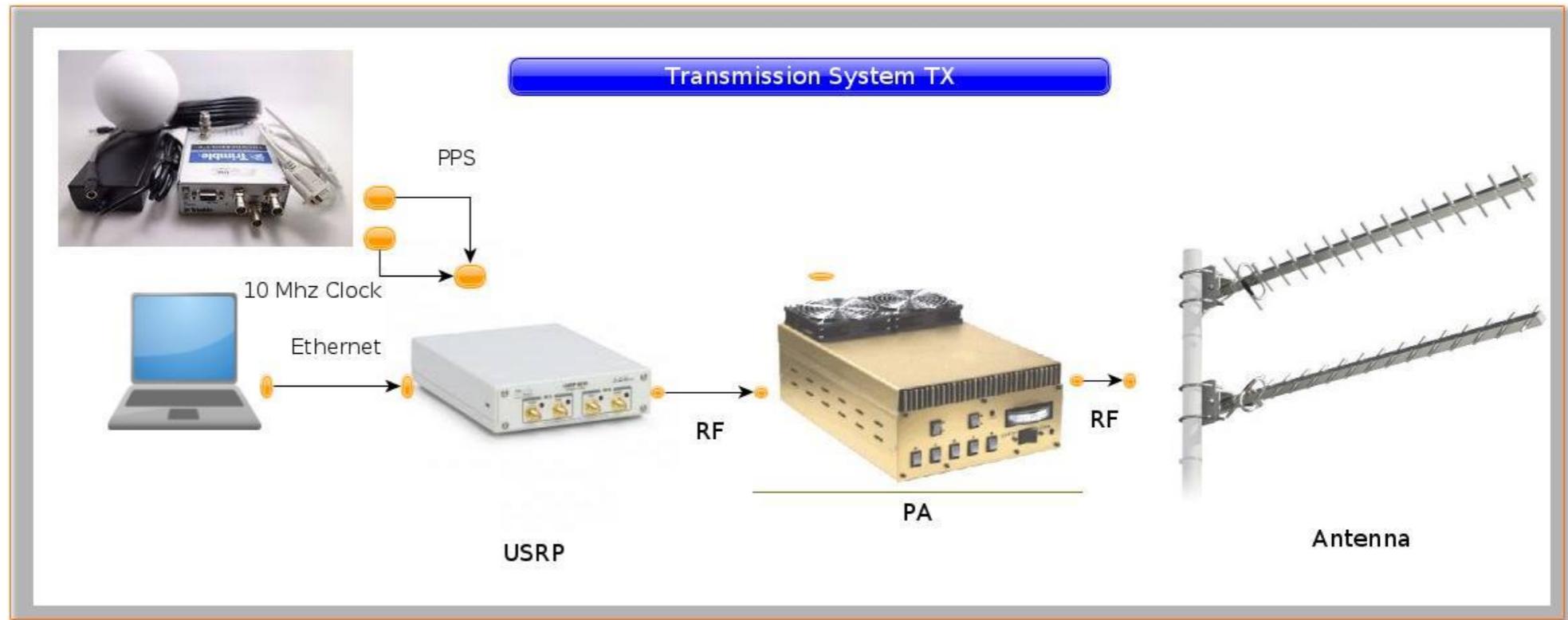
OUTLINE

- Implementation of Multistatic Radar for Meteor Detection
- Documentation and Starting
- Hardware and Software available.
- Acquisition
- Transmission
- Analyze
- Test and Processing
- GNURADIO Companion
- Experiment and Meteor Campaign
- Results
- Conclusion and Recommendation.



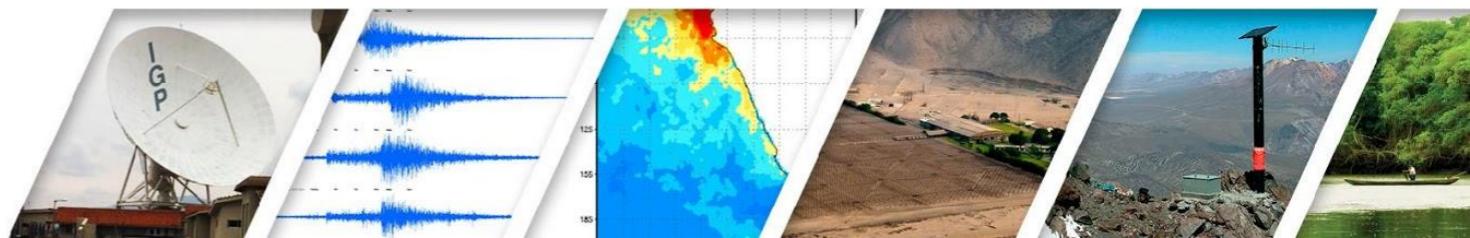
Implementation of Multistatic Radar Meteor Detection

- MIMO
- MISO
- SIMO**
- SISO



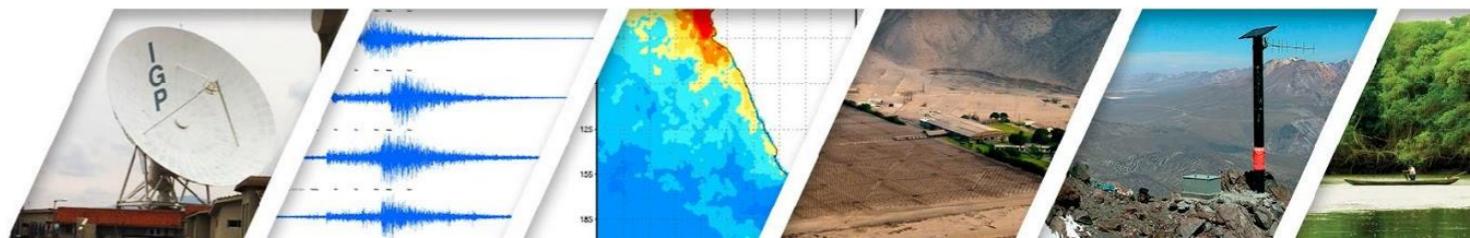
Documentation and Start

- Coded continuous wave meteor radar
 Juha Vierinen 1 , Jorge L. Chau 2 , Nico Pfeffer 2 , Matthias Clahsen 2 , and Gunter Stober 2
 1 MIT Haystack Observatory, Route 40 Westford, 01469 MA, USA
 2 IAP,Kühlungsborn, Germany
- Novel specular meteor radar systems using coherent MIMO techniques to study the mesosphere and lower thermosphere
 Jorge Luis Chau 1 , Juan Miguel Urco 1 , Juha Pekka Vierinen 2 , Ryan Andrew Volz 3 , Matthias Clahsen 1 , Nico Pfeffer 1 , and Jörg Trautner 1
 1 Leibniz Institute of Atmospheric Physics at the University of Rostock,Germany
 2 UiT Arctic University of Norway, Norway
 3 MIT Haystack Observatory, USA
- Radio Array of Portable Interferometric Detectors(RAPID) : Design and Applications
 Frank D. Lind, Colin J. Lonsdale, Ryan Volz, Anthea Coster, Chris Eckert, Russ McWhirter, Jim Marchese, Robert Schaefer, William Rideout, Reggie Wilcox
 Haystack Observatory- Massachusetts Institute of Technology
- Digital RF 2.0
 Juha Vierinen, William Rideout, Frank Lind, Robert Schaefer
- A multistatic HF beacon network for ionospheric specification in the Peruvian sector
 D. L. Hysell M. A. Milla J. Vierinen



Hardware and Software available

- Hardware available:
 - USR N310
 - URSP N200 + BASIC RX
 - USRP N200 + UBXN40
- Software available
 - Gnuradio
 - Uhd
 - [https://kb.ettus.com/Building and Installing the USRP Open-Source Tool chain \(UHD and GNU Radio\) on Linux](https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Tool_chain_(UHD_and_GNU_Radio)_on_Linux)
 - Digital_rf
 - https://github.com/MITHaystack/digital_rf
 - Programs: thor.py , tx.py , prc_analyce.py, drf_plot.py and create_waveform.py



THOR.py

```
gnuradio-config-info -v
3.7.13.4
```

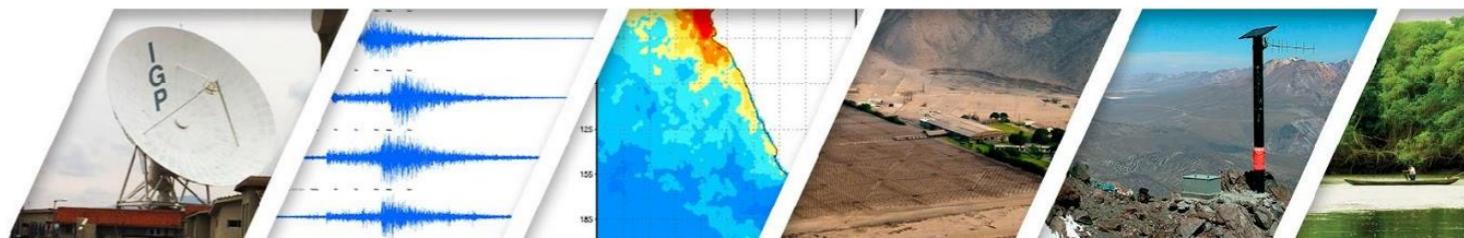
```
uhd_config_info --v
UHD
3.14.0.HEAD-0-g6875d061
```

Sistema Operativo
 LSB Version:
 core-9.20170808ubuntu1-noarch:security-9.20170808ubuntu1-noarch
 Distributor ID: Ubuntu
 Description: Ubuntu 16.04 LTS
 Release: 16.04
 Codename: xenial

Ubicacion:
 /home/alex/HaystackOP/

-m 192.168.10.2 -d "A:0" -c ch1 -y "TX/RX" -f 20e6 -F 10e3 -g 20 -b 0 -r 1e6 /data/test

ARGUMENTOS DE ENTRADA		ESCRITURA EN METADATA
-m	MBOARD	sequence_num
-d	subdev	num_subchannels
-c	CH	sample_rate_numerator
-f	Frequency	samples_rate_denominator
-F	OFFSET	init_utc_timestamp
-g	Gain	computer_time
-b	Bandwidth	
-r	rate	
DIR	Path	



TRANSMISION

tx.py

```
gnuradio-config-info -v
3.7.11
```

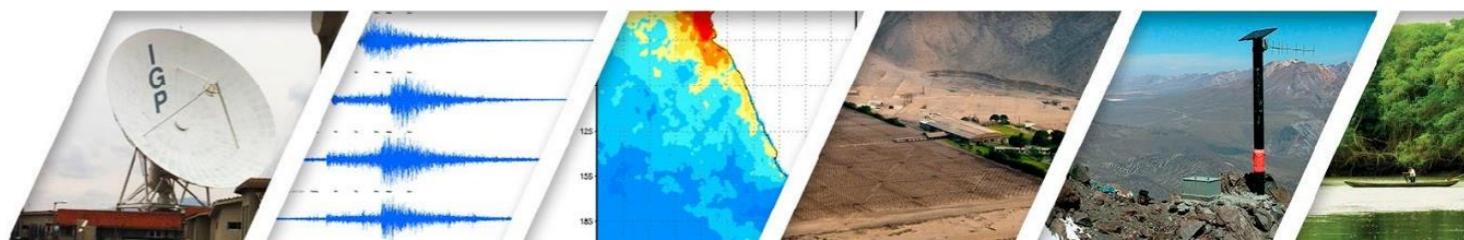
```
uhd_config_info --v
UHD
3.14.0.0-269-g11bbc3c0
```

Sistema Operativo
 LSB Version:
 core-9.20170808ubuntu1-noarch:security-9.20170808ubuntu1-noarch
 Distributor ID: Ubuntu
 Description: Ubuntu 18.04.1 LTS
 Release: 18.04
 Codename: bionic

Ubicacion:
 /home/soporte/GNURADIO_TX/uhd/host
 /examples/python/

```
python tx.py -m 172.16.5.189 -r 1e6 -f 25.0e6 -d "A:0" -y "TX/RX" -G 1 -g 33 -b 40e6 -l 1 --starttime 2019-07-10T13:05:00Z
--duration 240 --sync_source "external" /home/alex//sounder/waveforms/code-l10000-b10-000000.bin
```

ARGUMENTOS DE ENTRADA	
-m	
-f	freq
-r	rate
-d	subdev
-c	channels
-g	gain
--sync	external
-l	duration
-File	File



prc_analyze.py

- `create_pseudo_random_code(clen=10000, seed=0)`
- `periodic_convolution_matrix(envelope, rmin=0, rmax=100)`
- `create_estimation_matrix`

Nranges Number of code(`code, rmin=0, rmax=1000, cache=True`):

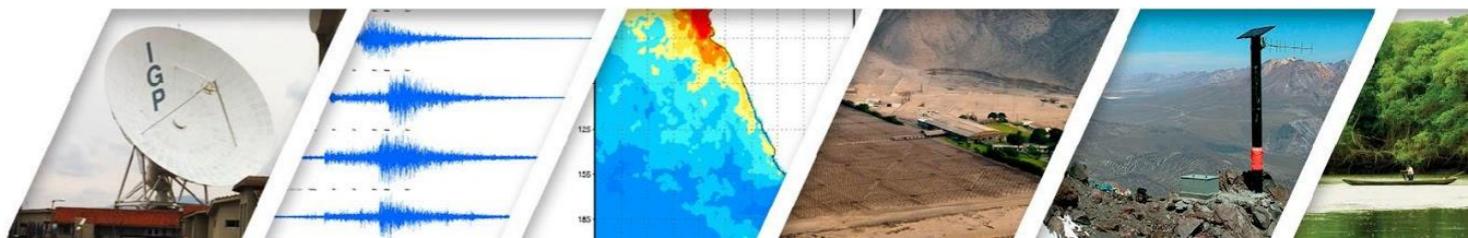
$$\text{res} = (1000, 10000) * z$$

Z= Data of our hdf5 file. (10000)

We get **res(100,1000)**, and finally ---> After fft----> SPEC

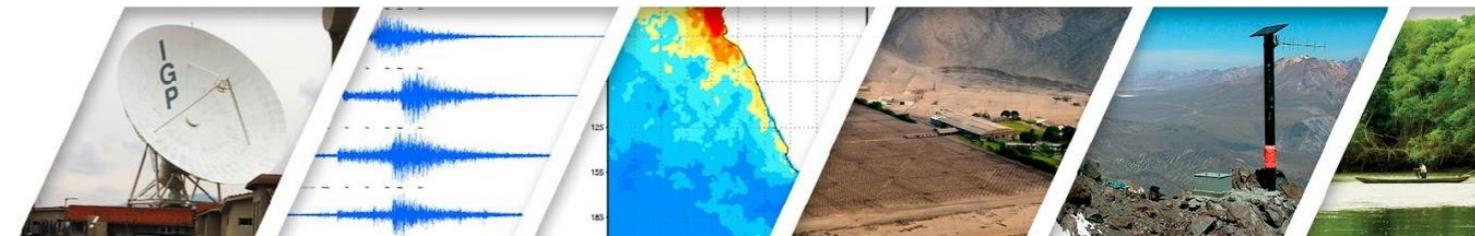
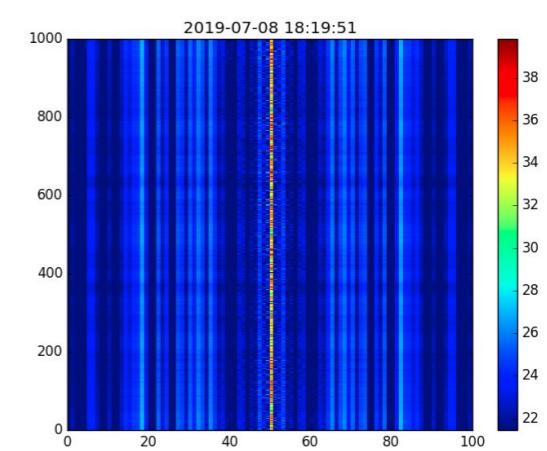
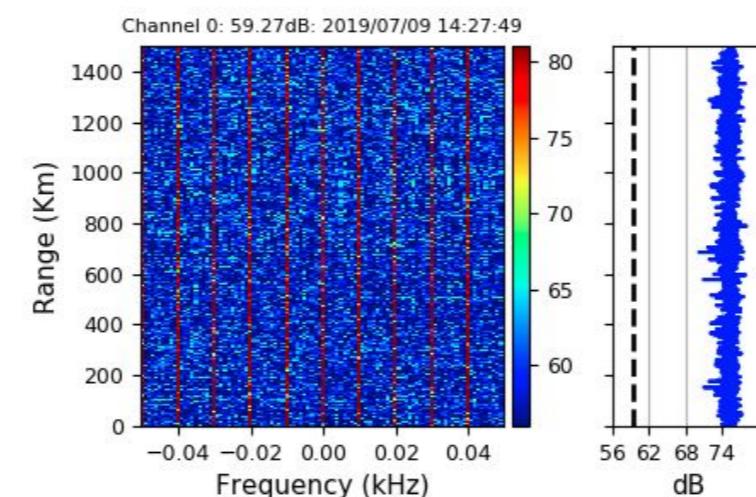
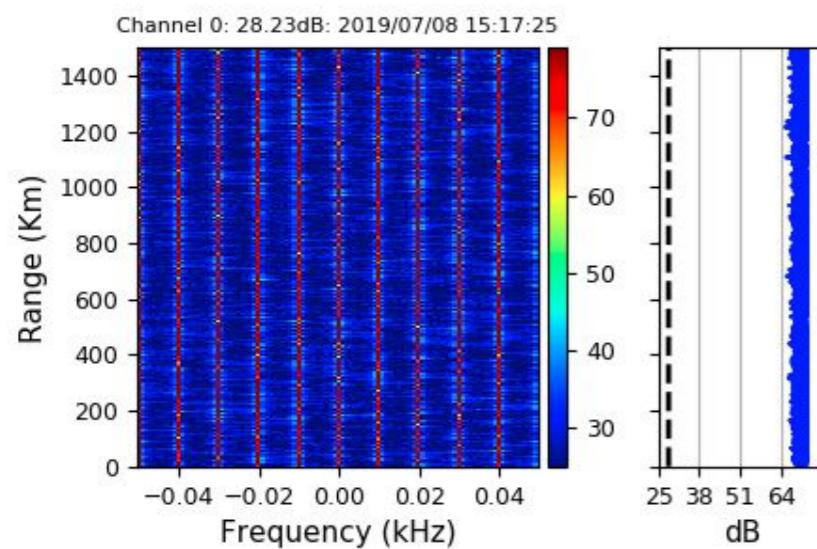
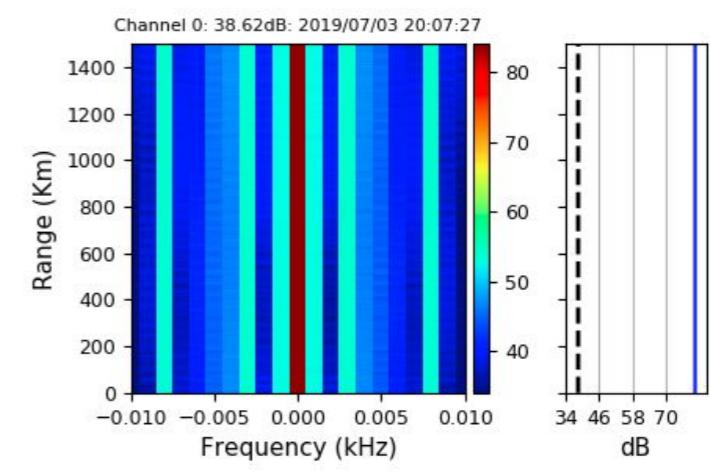
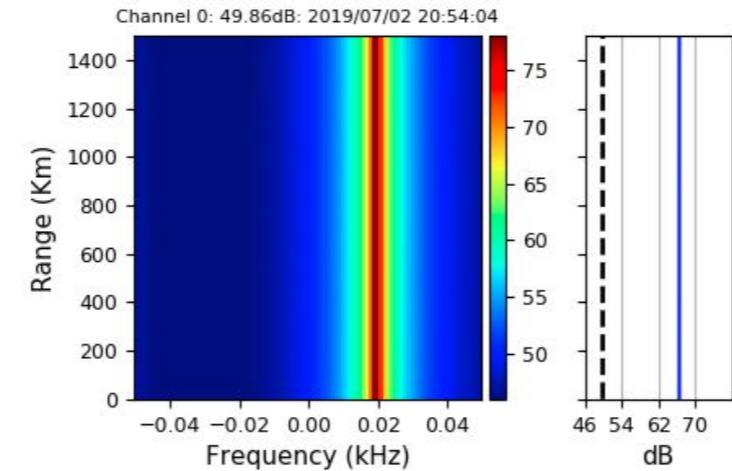
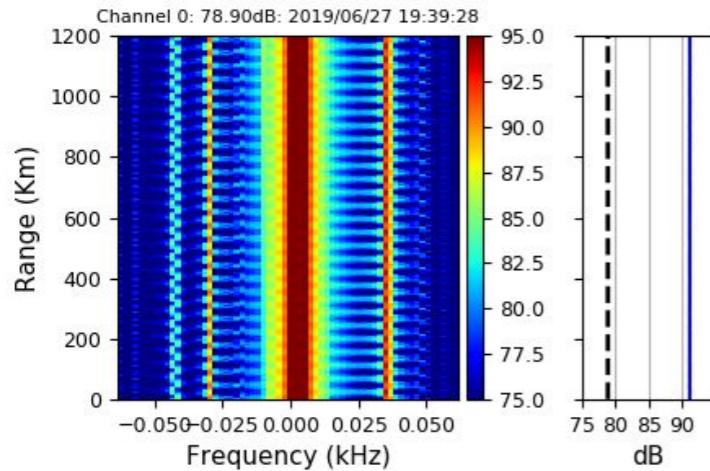
100 profiles

1000 Nranges



TEST AND PROCESSING

PRC ANALYZE.py



GNURADIO-COMPANION

tx_usrp3_ch0 ✎ tx_FRIDAY ✎ rx_THURSDAY ✎ test_avp ✎ RX ✎ tx_FRIDAYDV ✎

Options
ID: tx_FRIDAYDV
Generate Options: WX GUI

Vector Source
Vector: (numpy.fromfile('...'))
Tags:
Repeat: Yes

Variable
ID: samp_rate
Value: 625k

Import
import: numpy

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 625k
Trigger Mode: Auto
Y Axis Label: Counts

WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 625k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Multiply Const
Constant: 20

Throttle
Sample Rate: 625k

Low Pass Filter
Decimation: 1
Gain: 5
Sample Rate: 625k
Cutoff Freq: 75k
Transition Width: 50k
Window: Kaiser
Beta: 6.76

Rotator
Phase Increment: 2.81487

UHD: USRP Sink
Device Address: add...6.5.189
Clock Rate (Hz): 125M
Mb0: Clock Source: O/B GPSDO
Mb0: Time Source: O/B GPSDO
Mb0: Subdev Spec: A:0
Samp Rate (Sps): 625k
Ch0: Center Freq (Hz): 25.28M
Ch0: Gain Value: 0
Ch0: Antenna: TX/RX
TSB tag name:

Frequency Xlating FIR Filter
Decimation: 1
Taps: lpftaps
Center Frequency: 280k
Sample Rate: 625k

Band Pass Filter Taps
ID: lpftaps
Gain: 1
Sample Rate (Hz): 625k
Low Cutoff Freq (Hz): 245k
High Cutoff Freq (Hz): 405k
Transition Width (Hz): 10k
Window: Blackman
Beta: 6.76

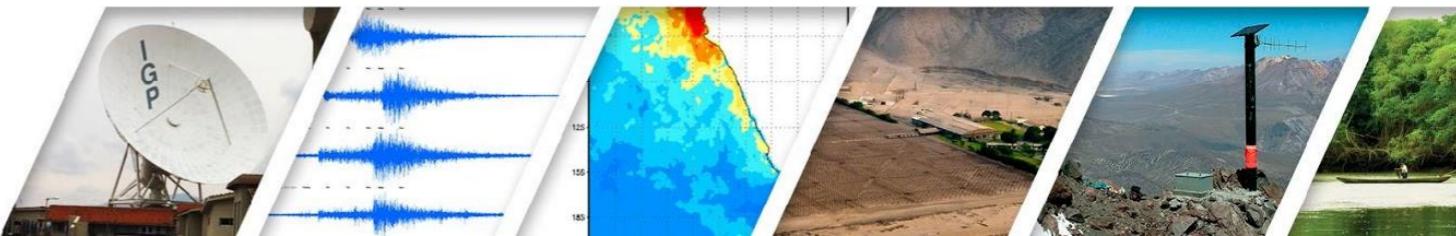
WX GUI FFT Sink
Title: FFT Plot
Sample Rate: 625k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Id	Value
Imports	import_0 import numpy
Variables	lpftaps <Open Properties> samp_rate 1250000/2

```

>>> Done
Loading: "/home/alex/HaystackOP/rx_THURSDAY.grc"
>>> Done
Loading: "/home/alex/test_avp.grc"
>>> Done
Loading: "/home/alex/HaystackOP/RX.grc"
>>> Done

```

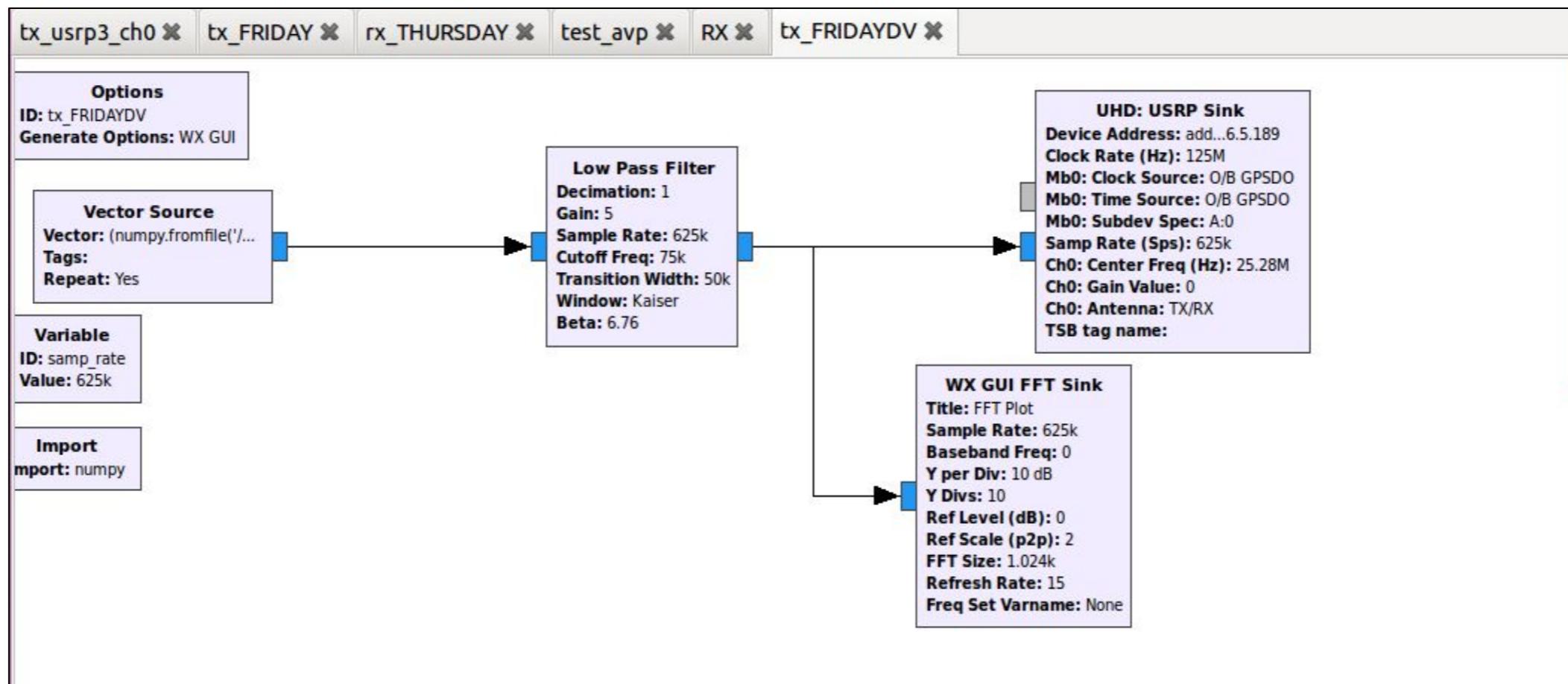


EXPERIMENT-METEOR CAMPAIGN

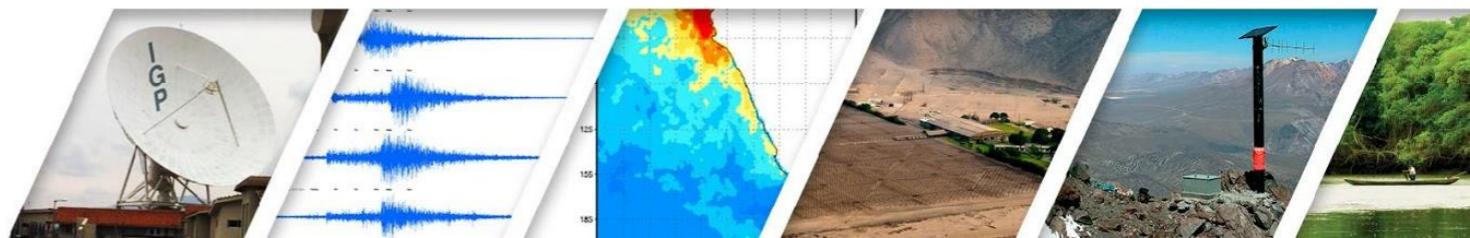
Rx: Thor.py

Tx: Gnuradio companion

Processing: prc_analyze.py

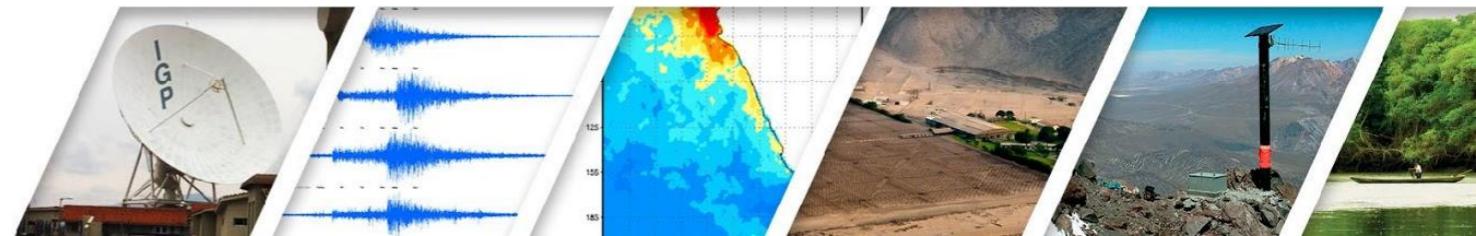


Tx_Program-GNURADIO- COMPANION

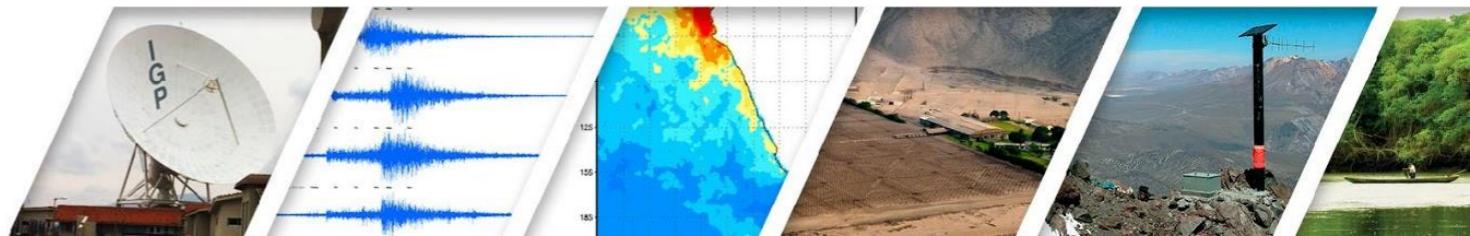
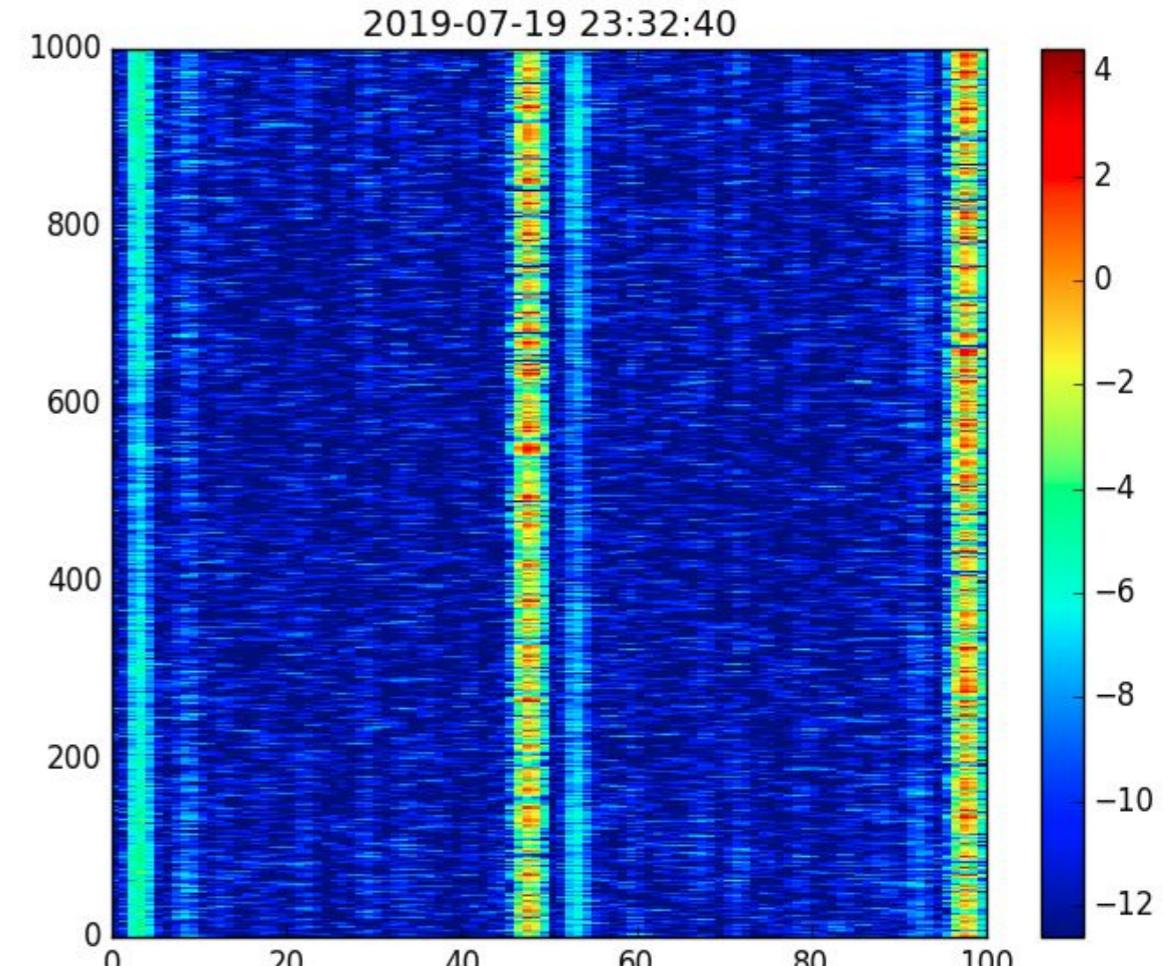
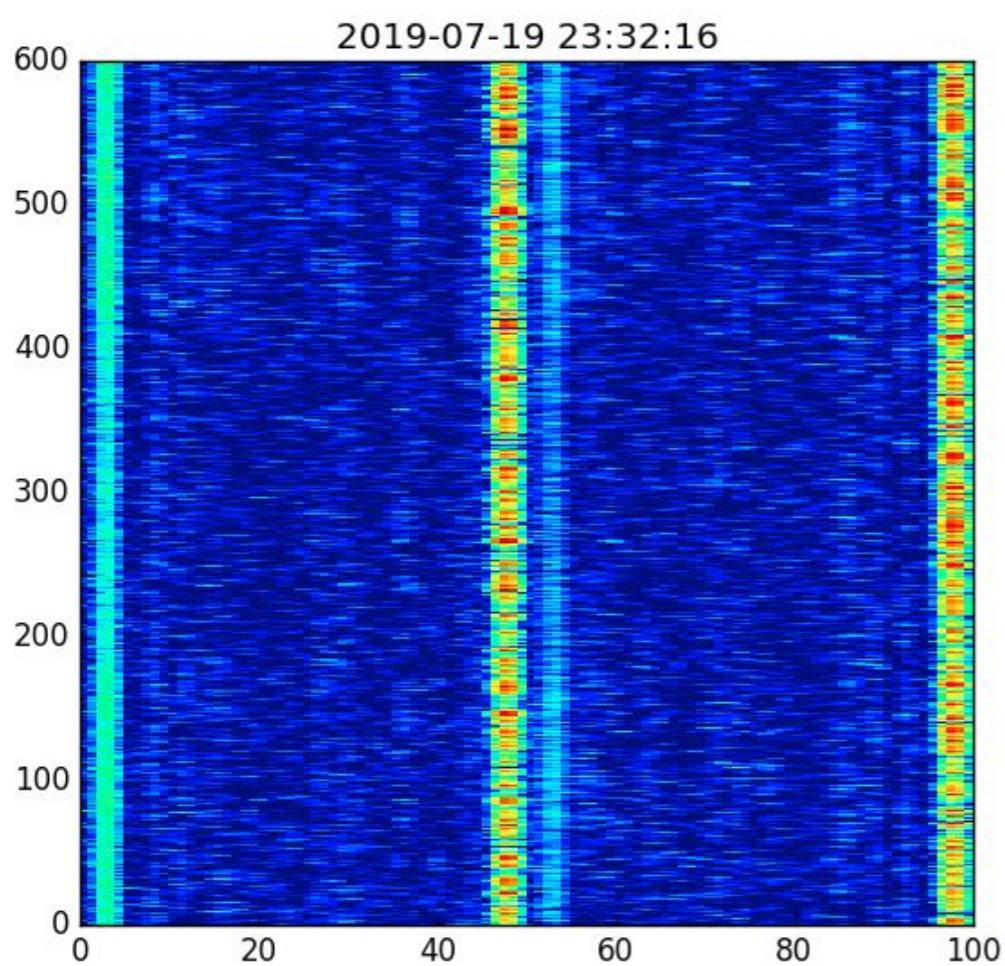
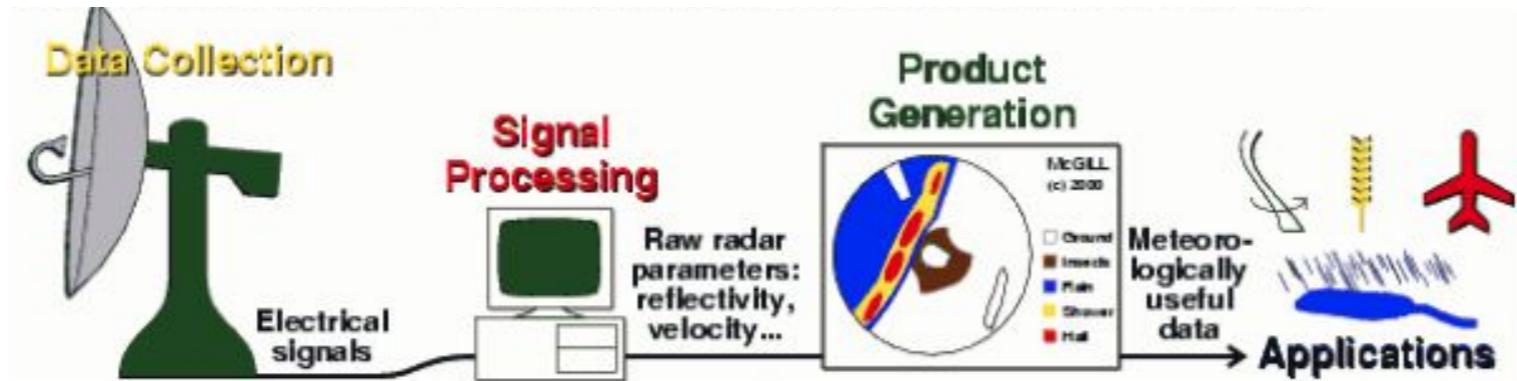


PARAMETER OF EXPERIMENT

Parameter	SIMO -CW
Frequency	25.280 MHz (11.859 meters / 38.9074803 feet)
Transmitter	250 kHz (out of 500 kHz licensed band), target ERP is 1W with antenna. 1 Transmitter
Pulse Type	Coded CW
Code length	10000 (Oversample by a factor of 5)
Code Type	Pseudorandom
Rx	Digital RF target recording bandwidth is 250 kHz. For N310, sample at 25.0 MHz with 1.25 MHz of bandwidth, use thor to offset tune 25.280 MHz and produce 250 kHz output as sc16 integer data
Receiver	<ul style="list-style-type: none"> a. Haystack Array Pad site with LWA-R antenna, MIT LWA-R LNA, and N310 radio (dual polarization) b. Haystack Office Lab site with LWA-R antenna and N310 radio (dual polarization) c. Lind LWA-R site (N200 radio, 30 MHz LPF, dual polarization)

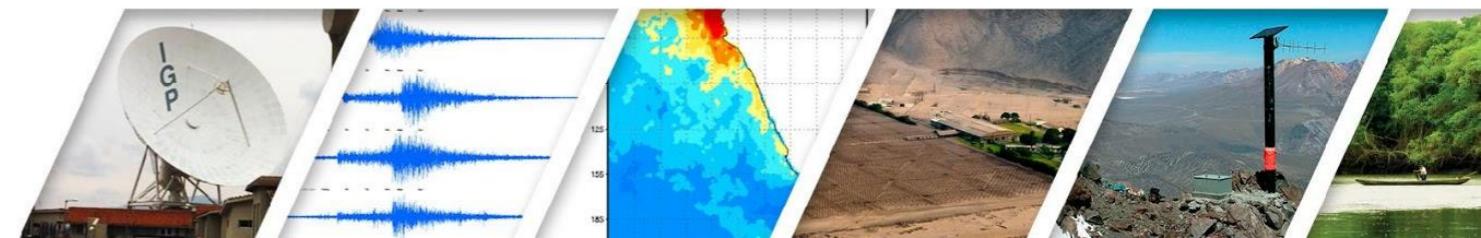


PROCESSING AND RESULT

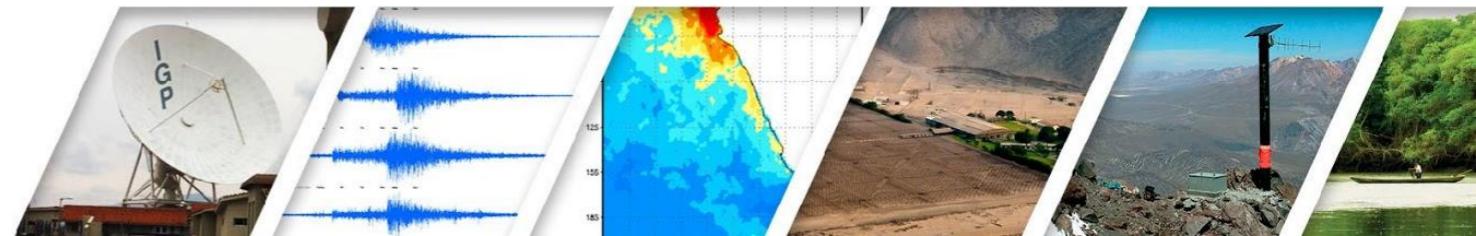


CONCLUSIONS AND RECOMMENDATION

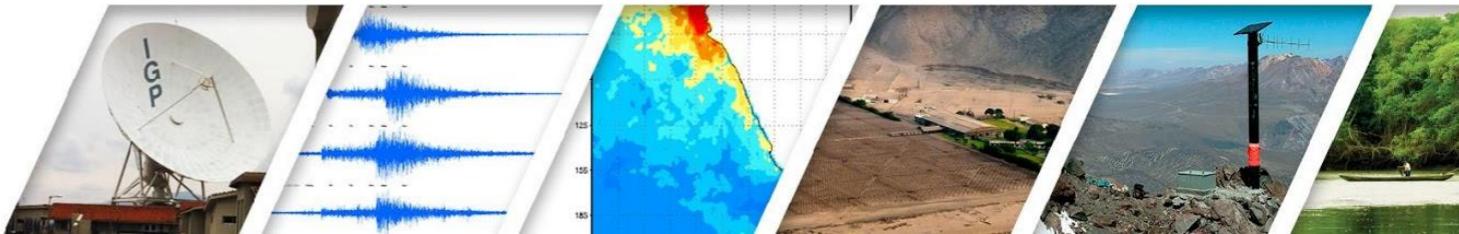
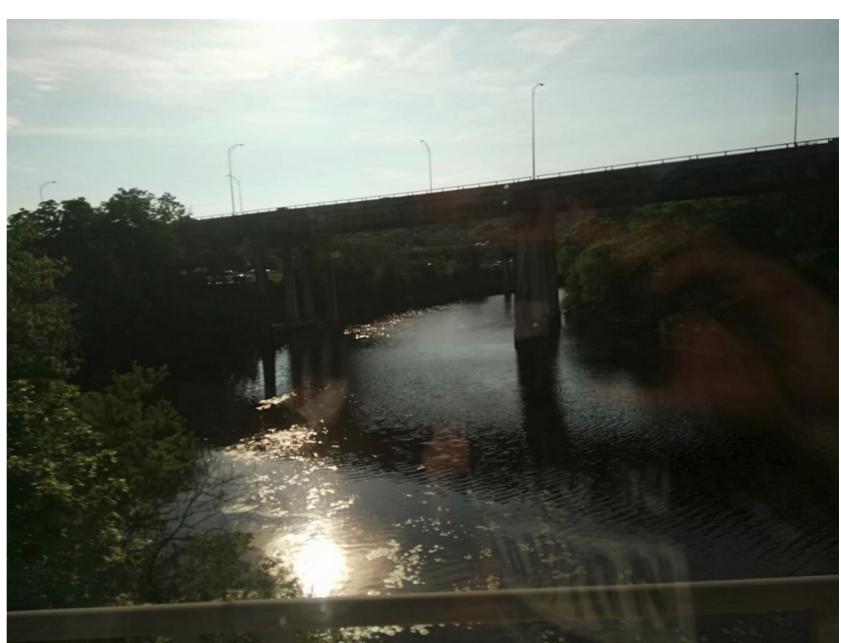
- Run a Meteor Campaign with all instrument available is possible we need to do more test.
- SDR and GNURADIO in general are a powerfull tool to rapid prototyping and implementation.
- Test in laboratory with the 25.280 Mhz and repeat the experiment with more Power.
- Learn more about code continuous wave meteor radar.
- Design a program to acelerate and setup the configuration in USRP N310.
- Design a Web Application for Monitoring SDR in general.



Happyness



WONDERFUL INFRASTRUCTURE



Thank you!

**Ciencia para protegernos,
ciencia para avanzar.**

www.igp.gob.pe

