

Software Defined Radio (SDR) Demonstration - Ettus USRP E312

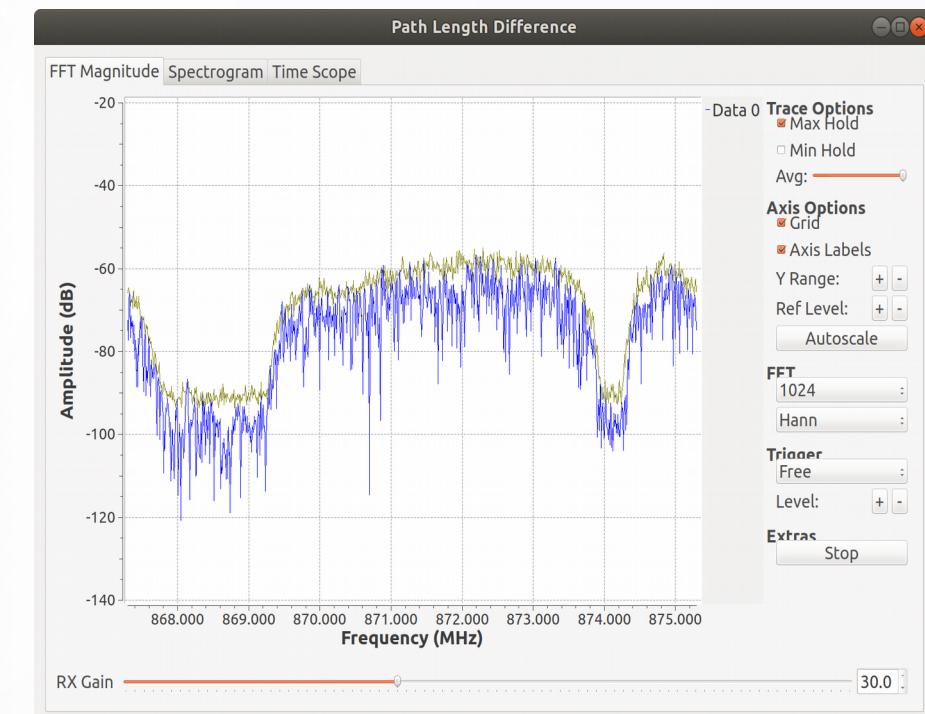
Spiro Sarris. 07 / 2018



```
rxstream_e3xx.py
```

```
4 # GNU Radio Python Flow Graph
5 # Title: Rxstream E3XX
6 # Generated: Sat Jul 14 18:09:10 2018
7 #####
8
9 from gnuradio import analog
10 from gnuradio import blocks
11 from gnuradio import eng_notation
12 from gnuradio import gr
13 from gnuradio import uhd
14 from gnuradio import zeromq
15 from gnuradio.eng_option import eng_option
16 from gnuradio.filter import firdes
17 from optparse import OptionParser
18 import SimpleXMLRPCServer
19 import threading
20 import time
21
22
23 class rxstream_e3xx(gr.top_block):
24
25     def __init__(self, freq=100e6, rx_gain=30, tx_gain=30):
26         gr.top_block.__init__(self, "Rxstream E3XX")
27
28         #####
29         # Parameters
30         #####
31         self.freq = freq
32         self.rx_gain = rx_gain
33         self.tx_gain = tx_gain
34
35         #####
36         # Variables
37         #####
38         self.tuning_lo_offset = tuning_lo_offset = 60e3
39         self.server_port = server_port = 30000
40         self.server_address = server_address = "192.168.10.184"
```

Line 1, Column 1 Spaces: 4 Python



Outline

- Equipment
- System Overview
- Example 1 – RX record, TX playback, post-process
- Example 2 – Measure length of RG316 coaxial cable

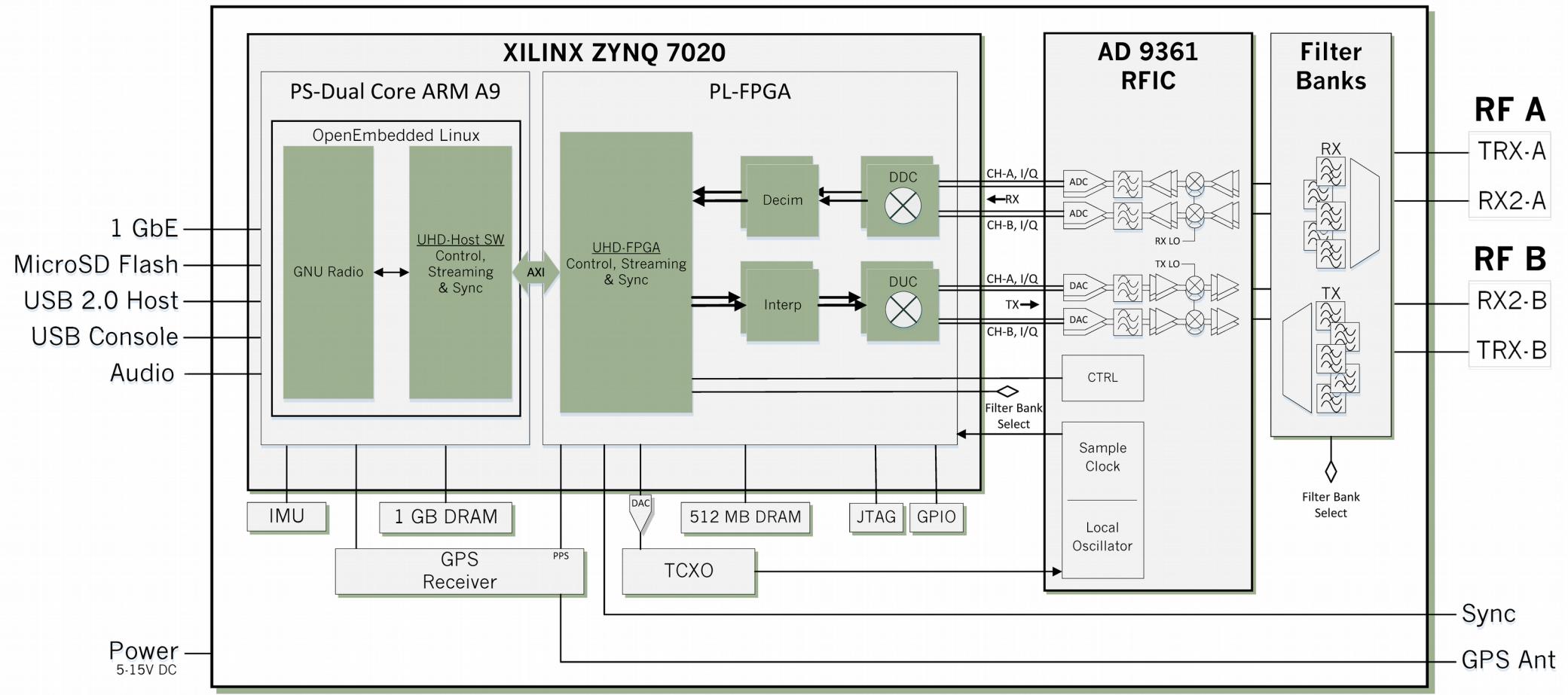
Equipment

- SDR – Ettus USRP E312
- Spectrum Analyzer
- Antenna – 900 MHz dipole
- Cables – coaxial with SMA
- Computer – OS Ubuntu Linux
- RF adapters



System Overview - E312

- Digital data interfaces on the left side (back of box)
- RF interfaces on the right side (front of box)



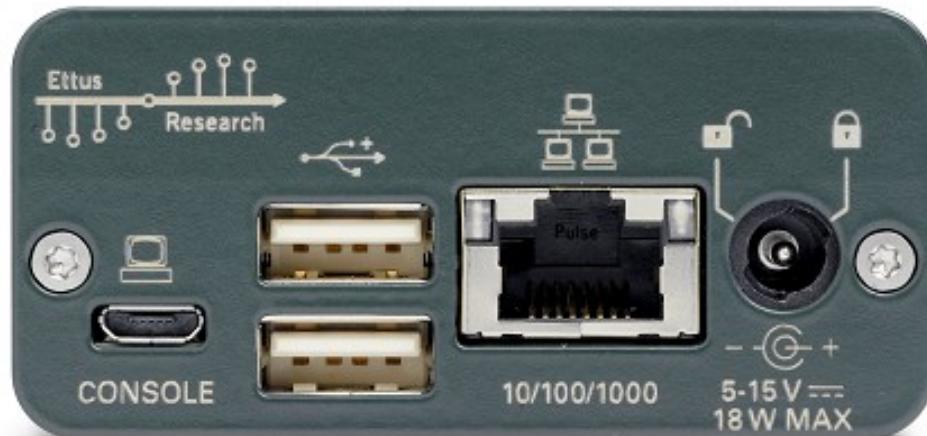
System Overview – RF

- 70 MHz - 6 GHz tuning range
- Analog bandwidth 56 MHz
- 2 TX, maximum output power > +10 dBm (depends on frequency)
- 2 RX, noise figure < 8 dB. IIP3 = -20 dBm
- Input and output filters automatic
- Direct-conversion transceiver – Analog Devices AD9361



System Overview – Digital

- Includes an Xilinx Zynq processor and Linux OS for stand-alone operation
- USB-serial interface for configuration (FTDI standard)
- Ethernet interface to connect to computer or network (SSH, TCP/UDP socket data)
- 10 Msamples/s data transfer on Zynq processor.

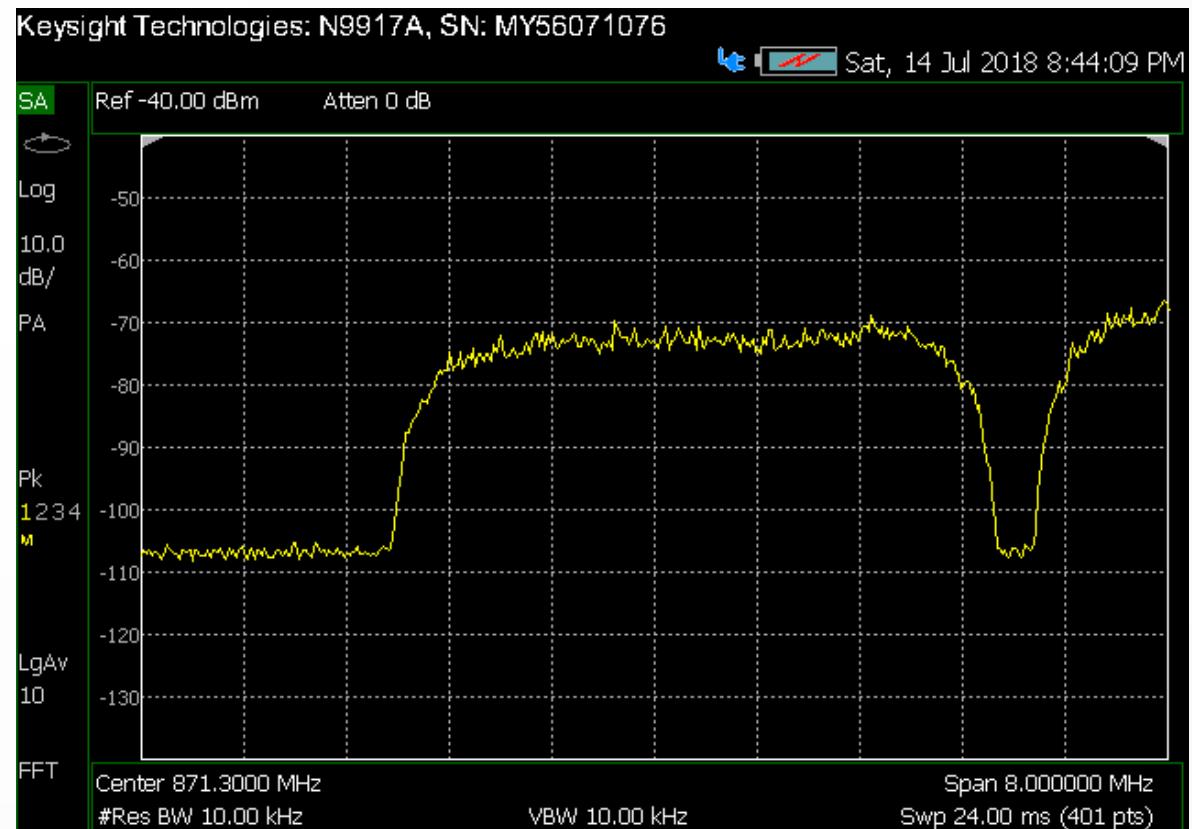
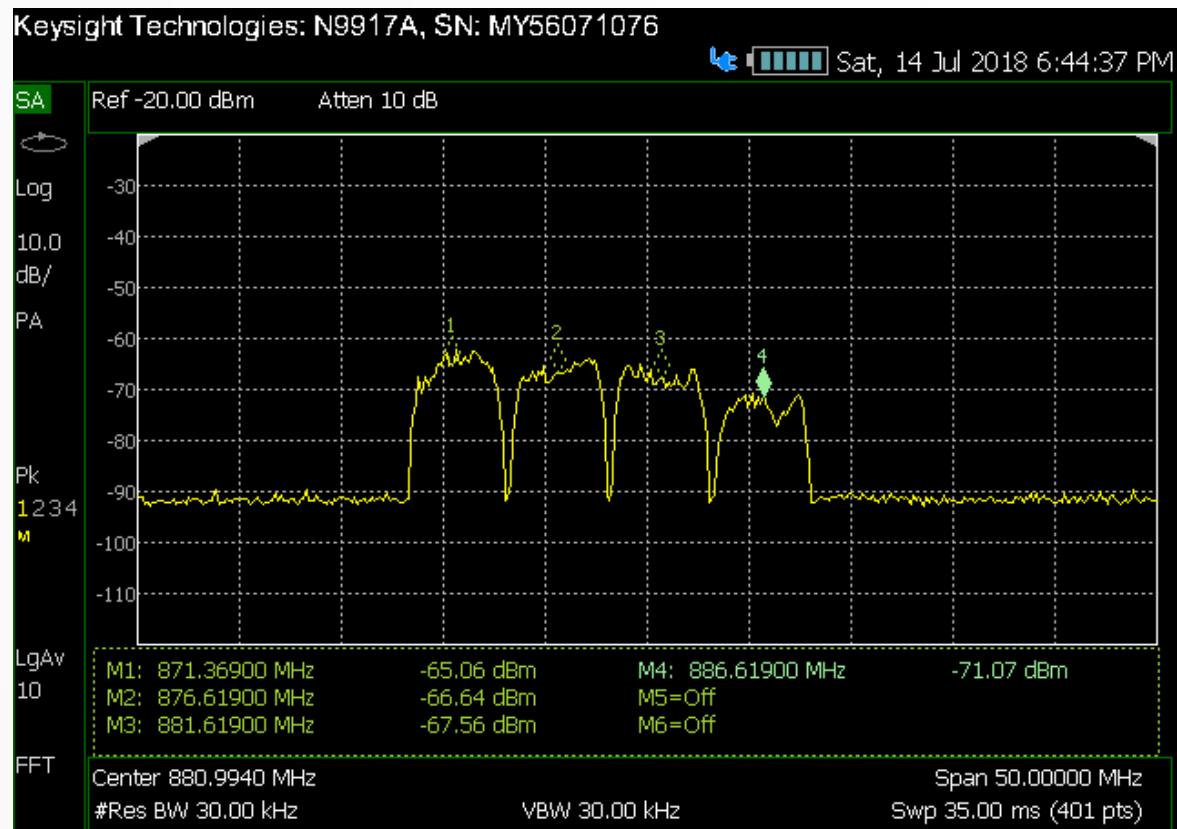


System Overview – Software

- UHD is the device driver software required Ettus to control USRP hardware. Source code available at <https://github.com/EttusResearch/uhd>
- GNURadio software provides many tools to create signal processing applications. Includes UHD classes to interface to Ettus USRP hardware. Source code available at <https://github.com/gnuradio>
- GRC (GNU Radio Companion) is the graphical development environment to create GNU Radio apps quickly.
- My example applications today are developed in GNURadio Companion and Python. Source code available at <https://github.com/spiro-sarris>

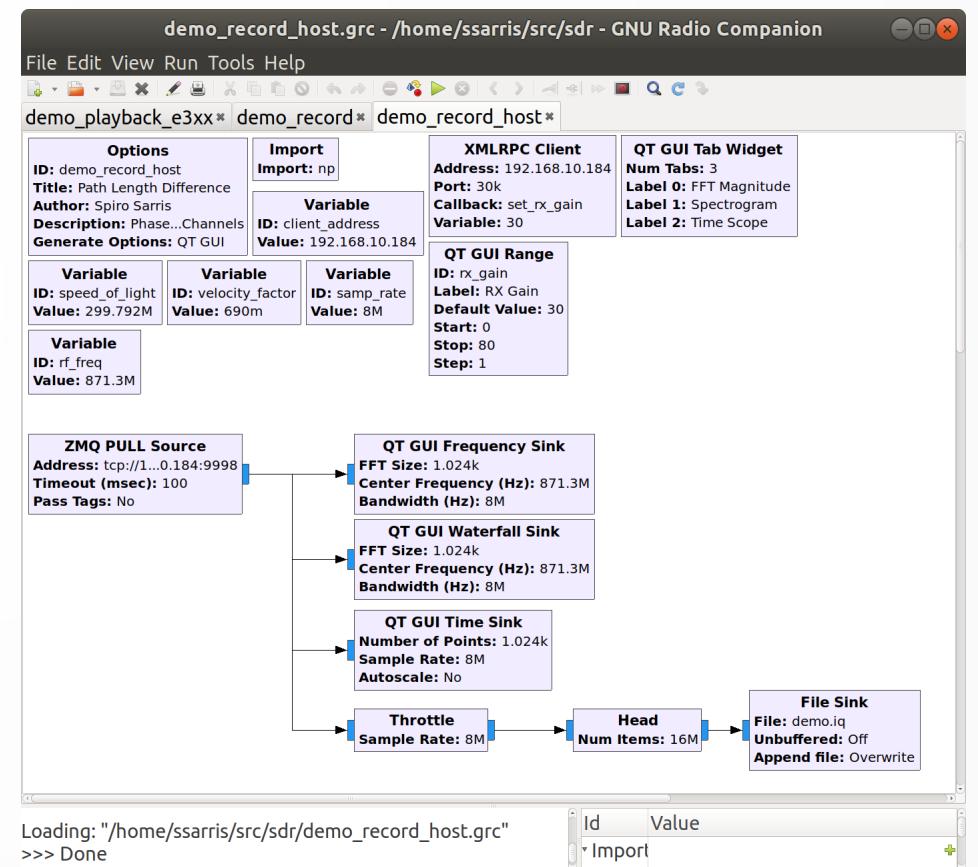
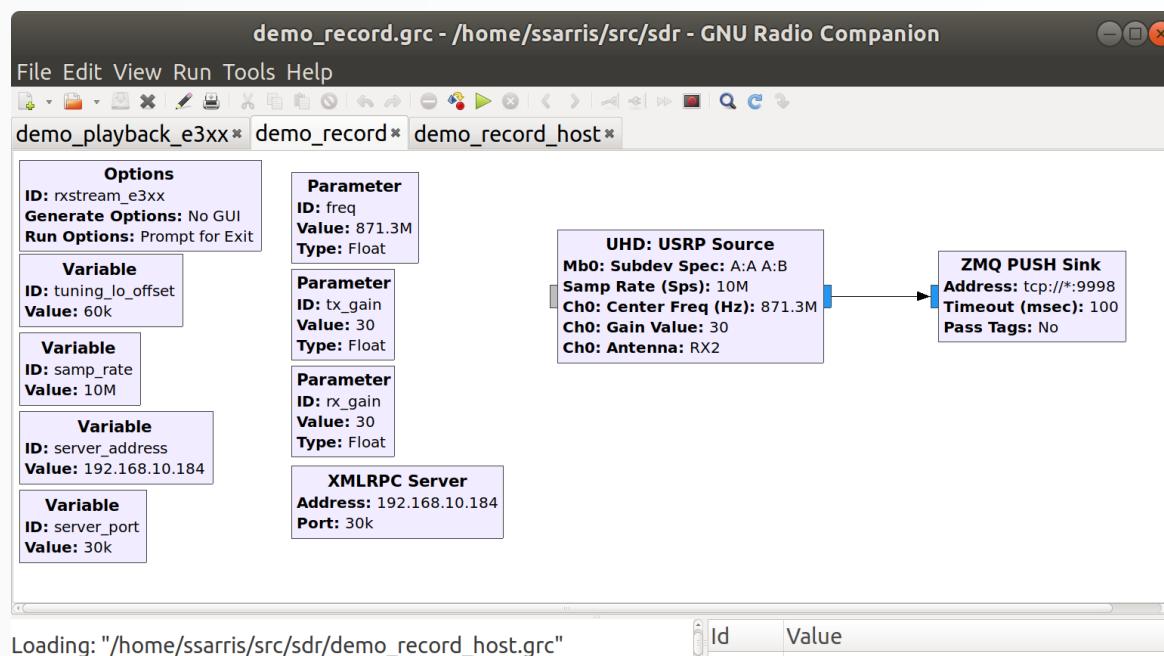
Example 1 - RX Record

- Use spectrum analyzer with antenna to find something interesting near 900 MHz. Mobile wireless signal?



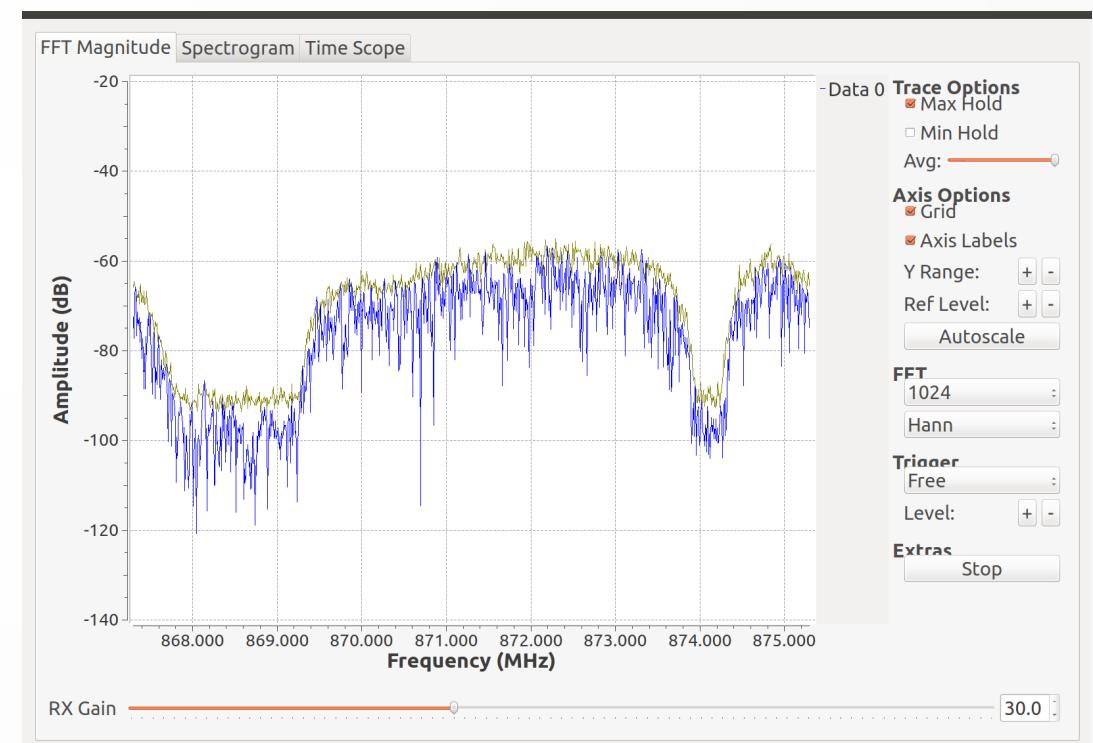
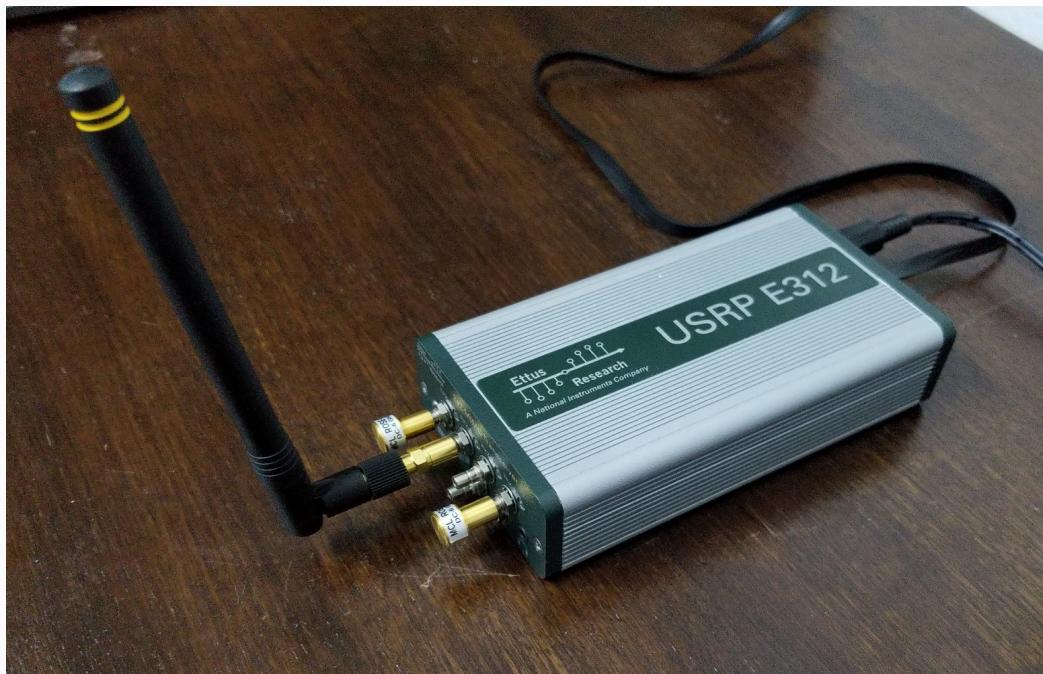
Example 1 - RX Record

- Load software on SDR (stream data) and on host laptop (process and show GUI)



Example 1 - RX Record

- Connect antenna to SDR. Observe GUI and record file

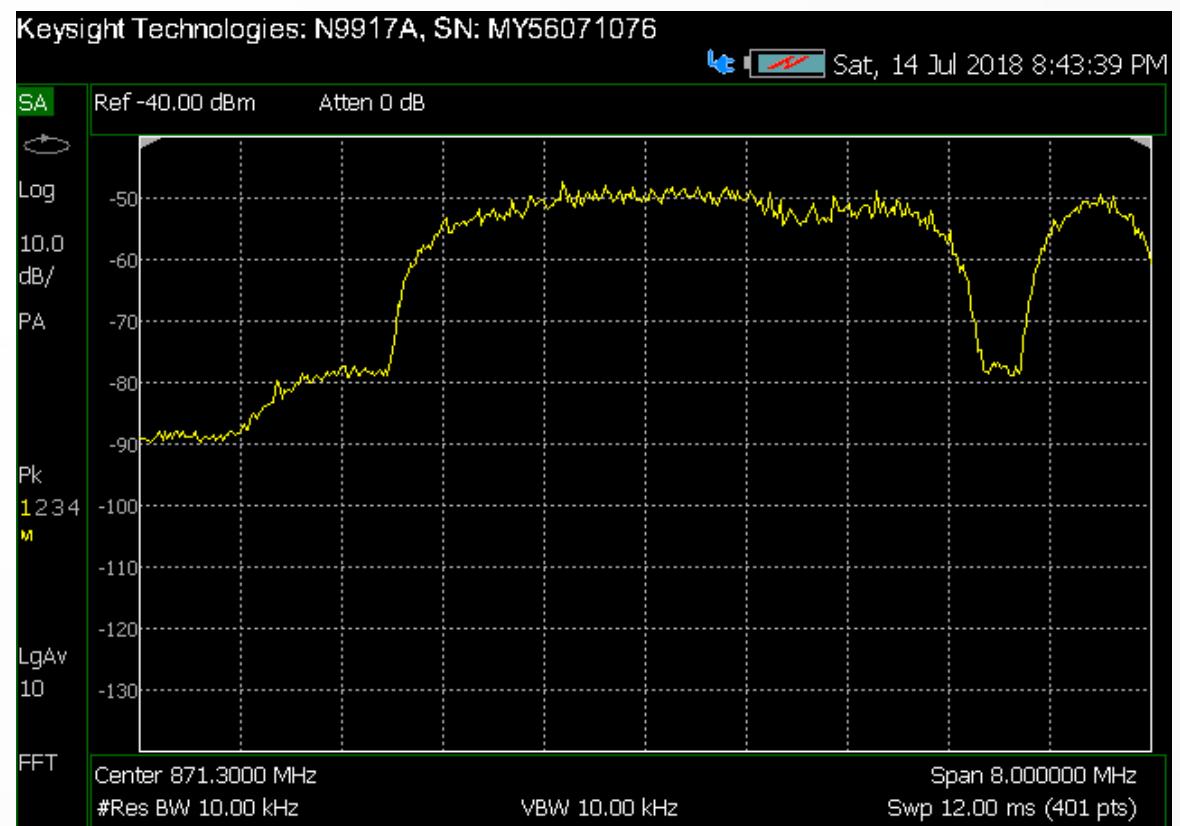
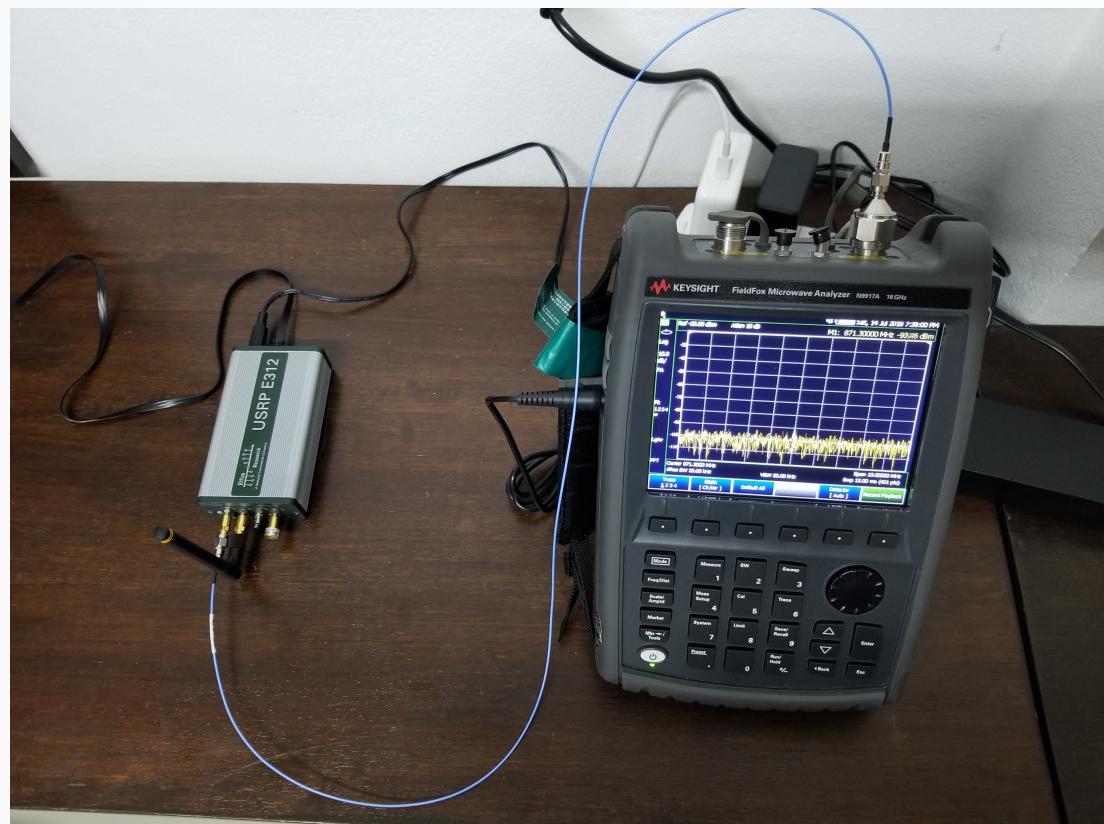


Example 1 - TX Playback

- Transfer file from host computer to SDR local storage using scp over Ethernet.
 - \$ scp ./demo.iq root@192.168.10.198:/home/root/sdr/
- SSH login to SDR. Locate playback python script.
 - \$ ssh root@192.168.10.198
 - \$ cd sdr/

Example 1 - TX Playback

- Connect TX-A to spectrum analyzer with 30 dB attenuator to guarantee safe power level at receiver.
 - \$ python ./demo_playback_e3xx.py



Example 1 – Post Process

- File recorded from GNURadio “file sink” object is a binary file without header / meta data. If you want meta data, open a second file and write data of interest.
- For each sample in time (example 8 MS/s), the complex-valued result is written sequentially as a pair of 32-bit (4-byte) float. “I” first, “Q” second. For each sample, the file size increases by 8 bytes. (write file 64 MB/s)
- You can choose a programming language to read this file into native complex-valued data format. Python and Matlab are convenient.

byte 0	byte 4	byte 8	byte 12	byte = 8*N	byte 8*N+4
I[0]	Q[0]	I[1]	Q[1]	I[N]	Q[N]

Example 2 - Cable Length

- c = speed of light in free space = 299792458 [meters/second]
- vf = velocity factor. Speed reduced by dielectric in cable [no unit]
- λ = wavelength in cable [meters]
- k = wave number [radians / meter]
- L = cable length [meters]

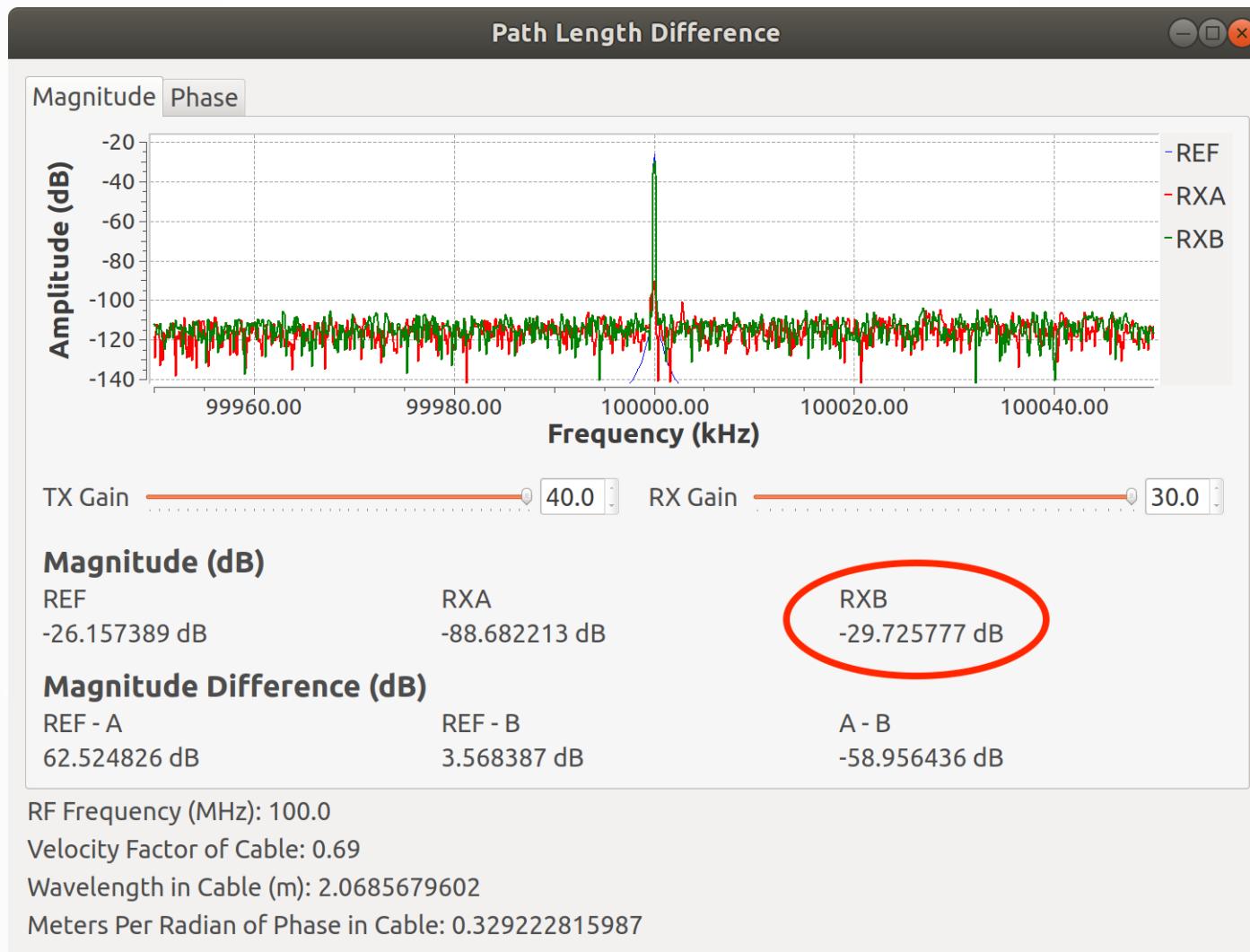
$$\lambda = \frac{c * vf}{f}$$

$$k = \frac{2\pi}{\lambda}$$

$$\Delta L = \frac{\Delta \phi}{k} = \Delta \phi * \frac{\lambda}{2\pi}$$

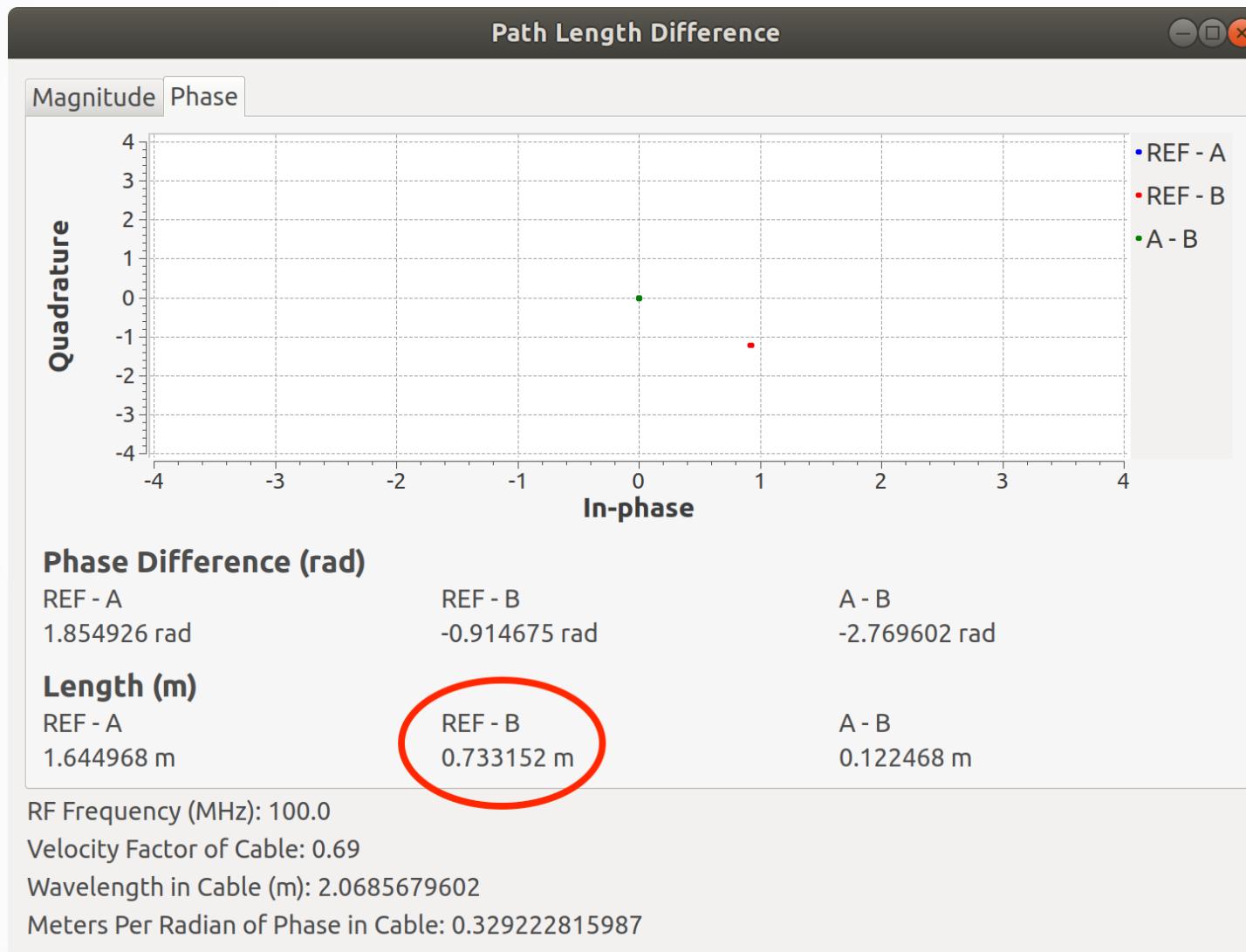
Example 2 - Cable Length

- Relative amplitude of calibration cable = -29.73 dB



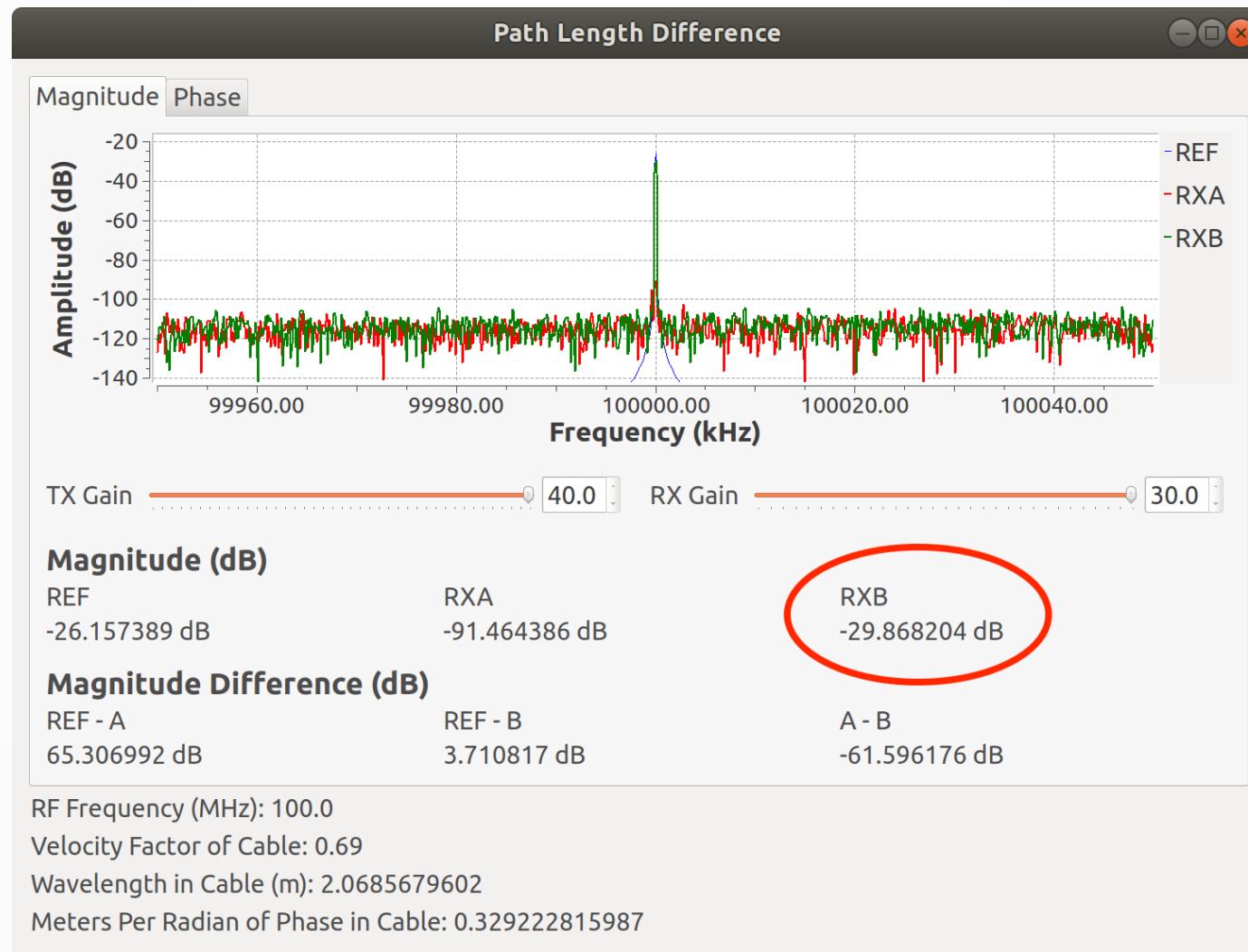
Example 2 - Cable Length

- Relative phase and length of calibration cable. $L = 73.3$ cm



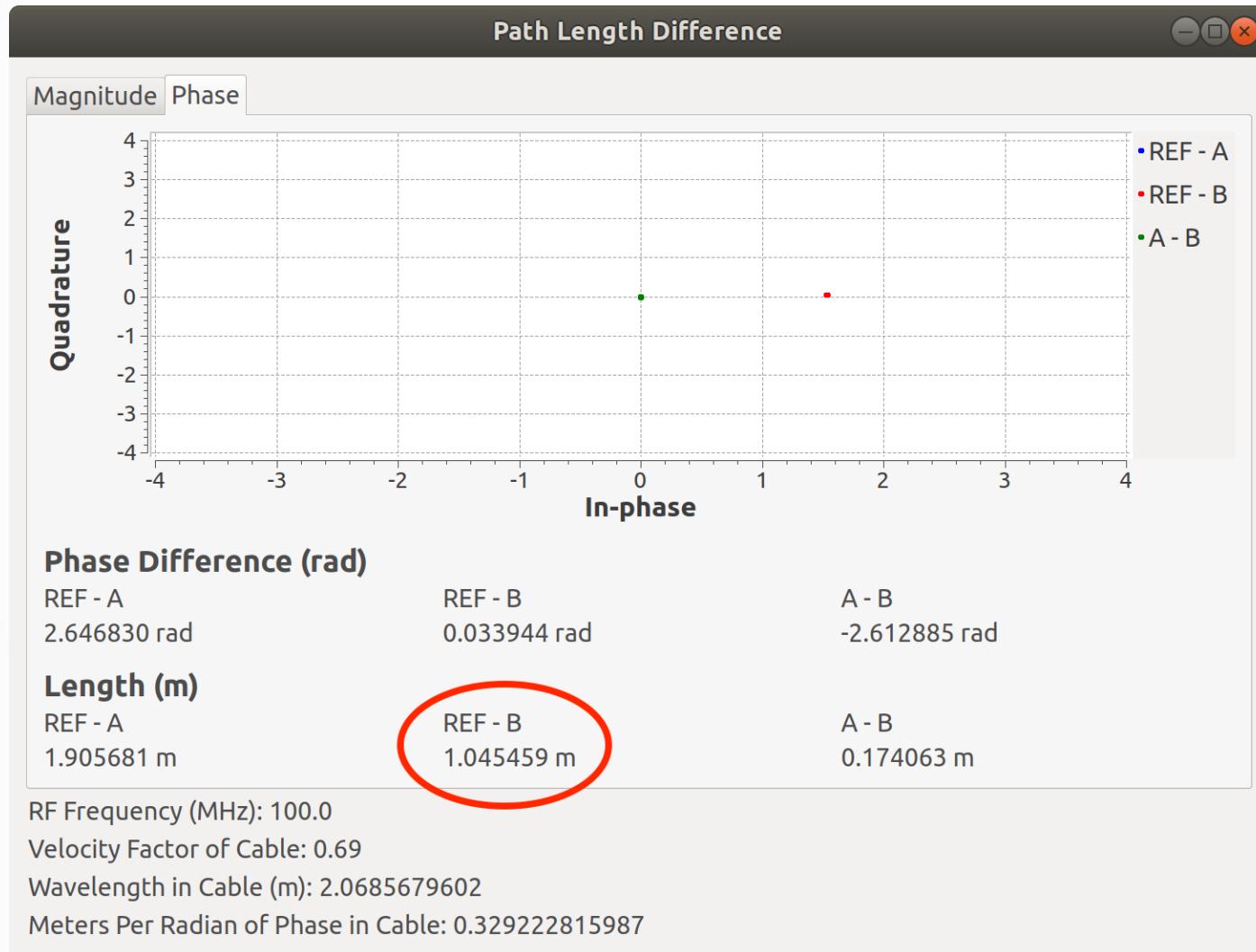
Example 2 - Cable Length

- Add DUT cable. Observe reduced amplitude = -29.87 dB



Example 2 - Cable Length

- Observe additional phase and length of DUT cable. $L = 104.5 \text{ cm}$



Example 2 - Cable Length

- Amplitude reduced by DUT cable : $(-29.87) - (-29.73) = -0.14 \text{ dB}$
- Length increased by DUT cable: $(104.5) - (73.3) = 31.2 \text{ cm}$
- Real cable length is $30.48 \text{ cm} + \text{adapter } (0.7 \text{ cm})?$