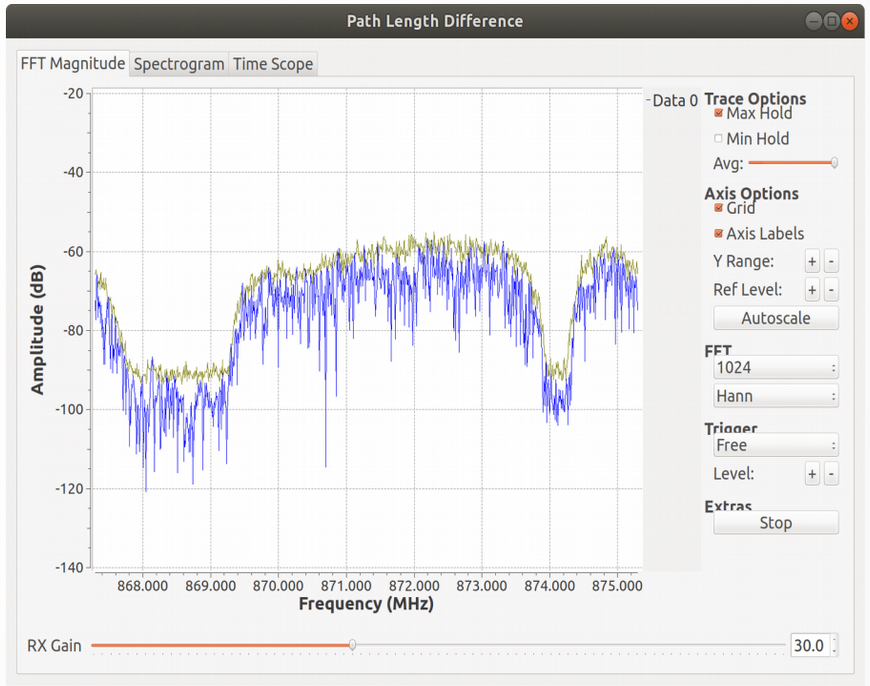
Software Defined Radio (SDR) Demonstration - Ettus USRP E312

Spiro Sarris. 07 / 2018

rxstream_e3xx.py — UNREGISTERED

```python
# GNU Radio Python Flow Graph
# Title: Rxstream E3Xx
# Generated: Sat Jul 14 18:09:10 2018
##################################################

from gnuradio import analog
from gnuradio import blocks
from gnuradio import eng_notation
from gnuradio import gr
from gnuradio import uhd
from gnuradio import zeromq
from gnuradio.eng_option import eng_option
from gnuradio.filter import firdes
from optparse import OptionParser
import SimpleXMLRPCServer
import threading
import time


class rxstream_e3xx(gr.top_block):

    def __init__(self, freq=100e6, rx_gain=30, tx_gain=30):
        gr.top_block.__init__(self, "Rxstream E3Xx")

        ##################################################
        # Parameters
        ##################################################
        self.freq = freq
        self.rx_gain = rx_gain
        self.tx_gain = tx_gain

        ##################################################
        # Variables
        ##################################################
        self.tuning_lo_offset = tuning_lo_offset = 60e3
        self.server_port = server_port = 30000
        self.server_address = server_address = "192.168.10.184"
```

Line 1, Column 1 Spaces: 4 Python



This presentation - https://github.com/spiro-sarris/pres/blob/master/sdrdemo.pdf

Download examples at https://github.com/spiro-sarris/sdr/

Outline

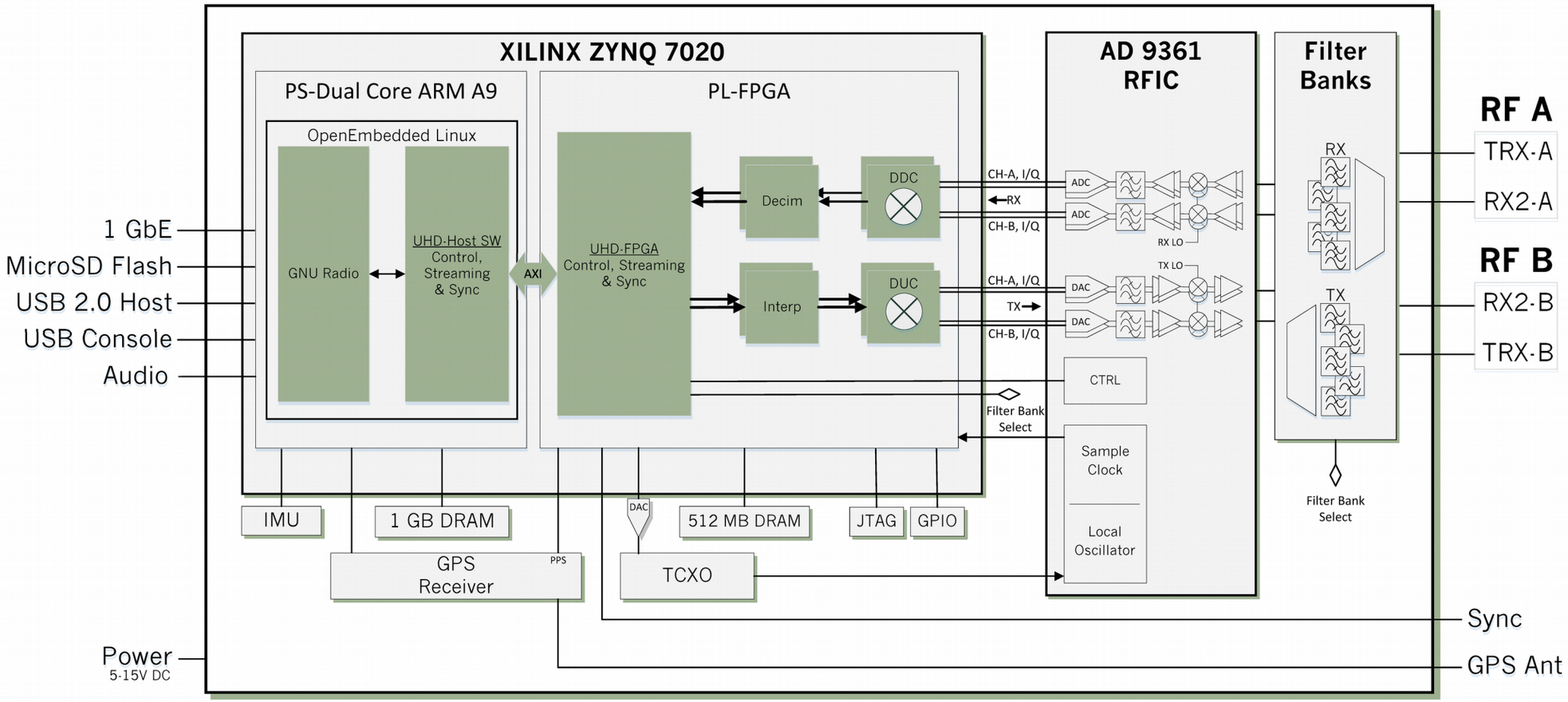
- Equipment
- System Overview
- Example 1 – RX record, TX playback, post-process
- Example 2 – Measure length of RG316 coaxial cable

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)

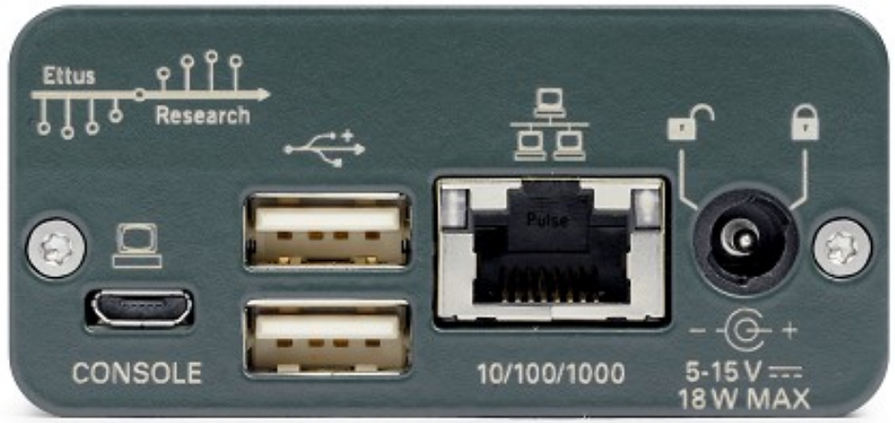


System Overview – RF

- 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 and 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.

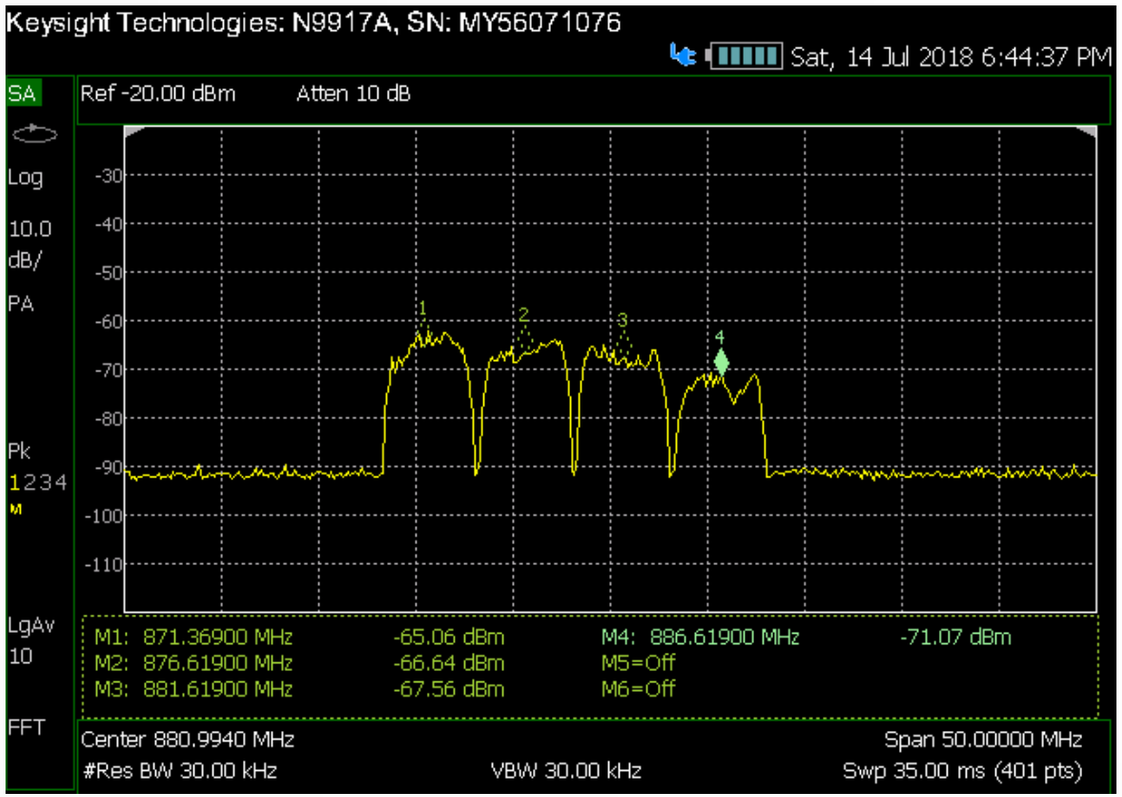


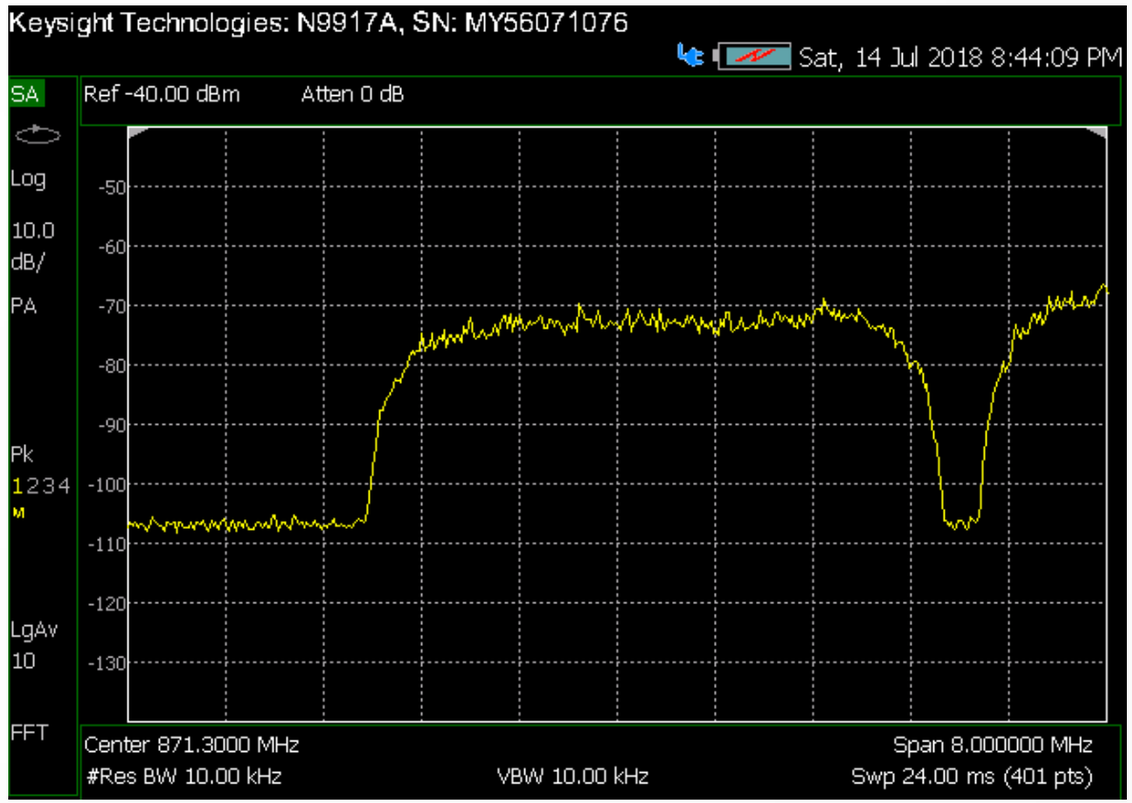
System Overview – Software

- 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

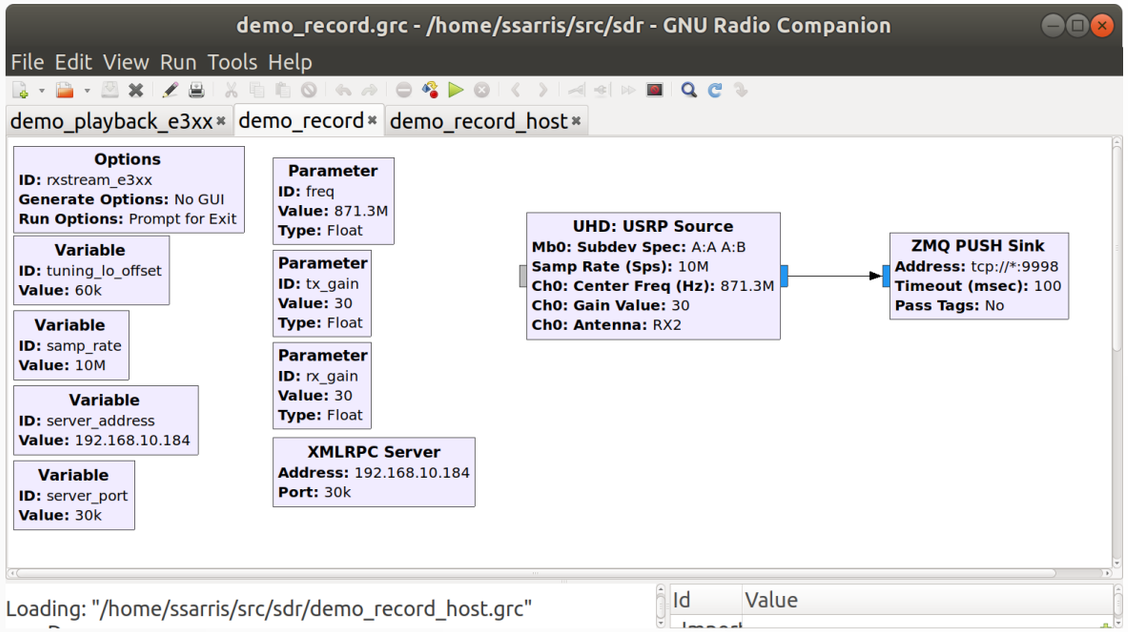
- Use spectrum analyzer with antenna to find something interesting near 900 MHz. Mobile wireless signal?

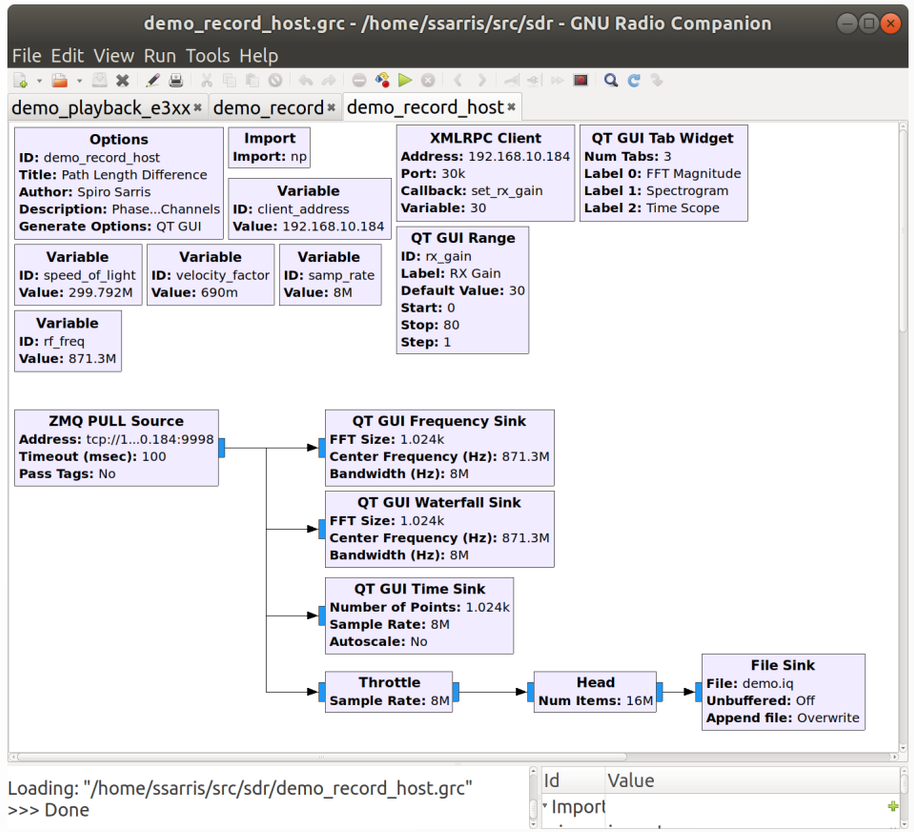




Example 1 – RX Record

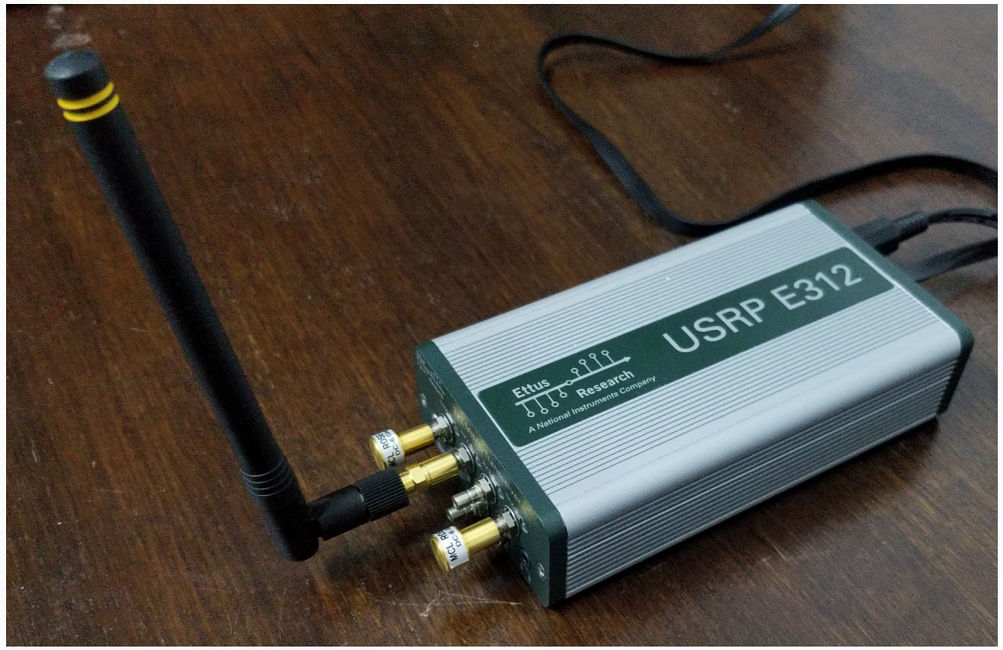
- Load software on SDR (stream data) and on host laptop (process and show GUI) TODO – add GUI of RXrecord

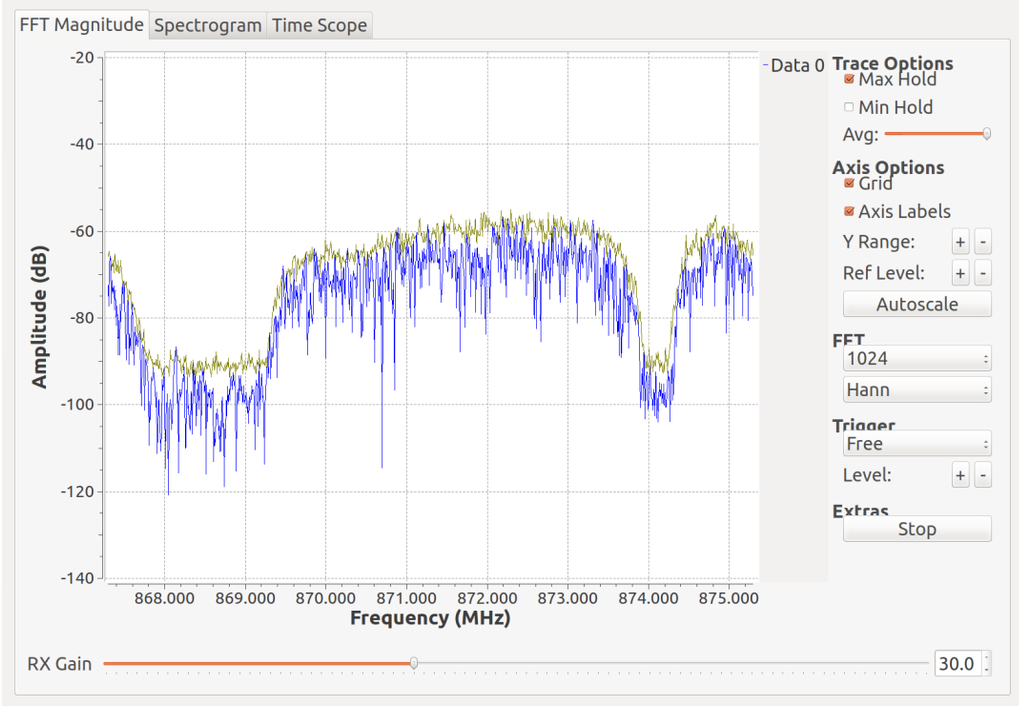




Example 1 – RX Record

- Connect antenna to SDR. Observe GUI and record file



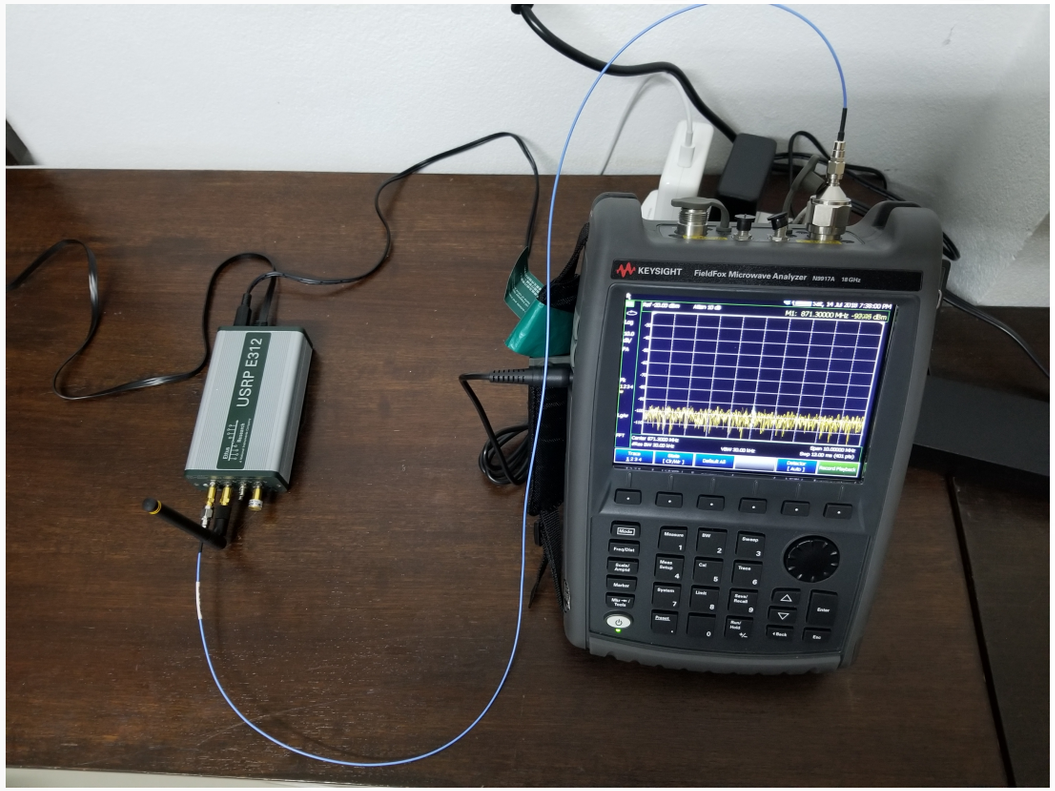


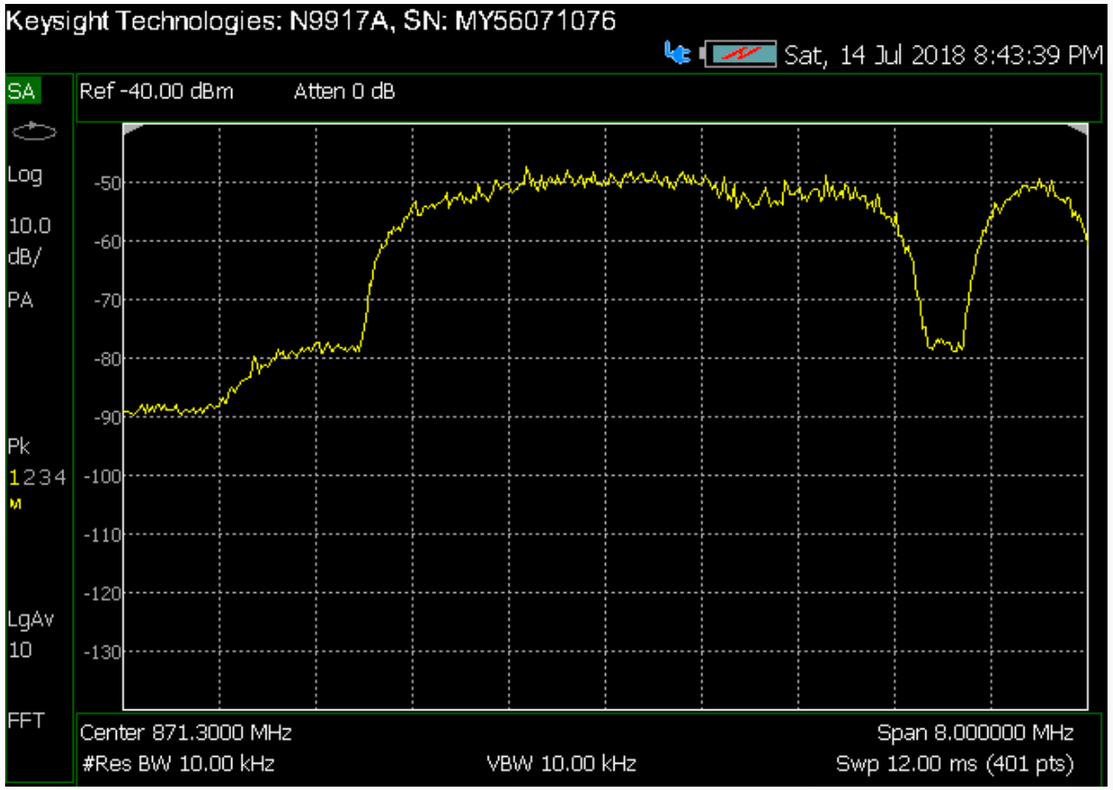
Example 1 – TX Playback

- 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`



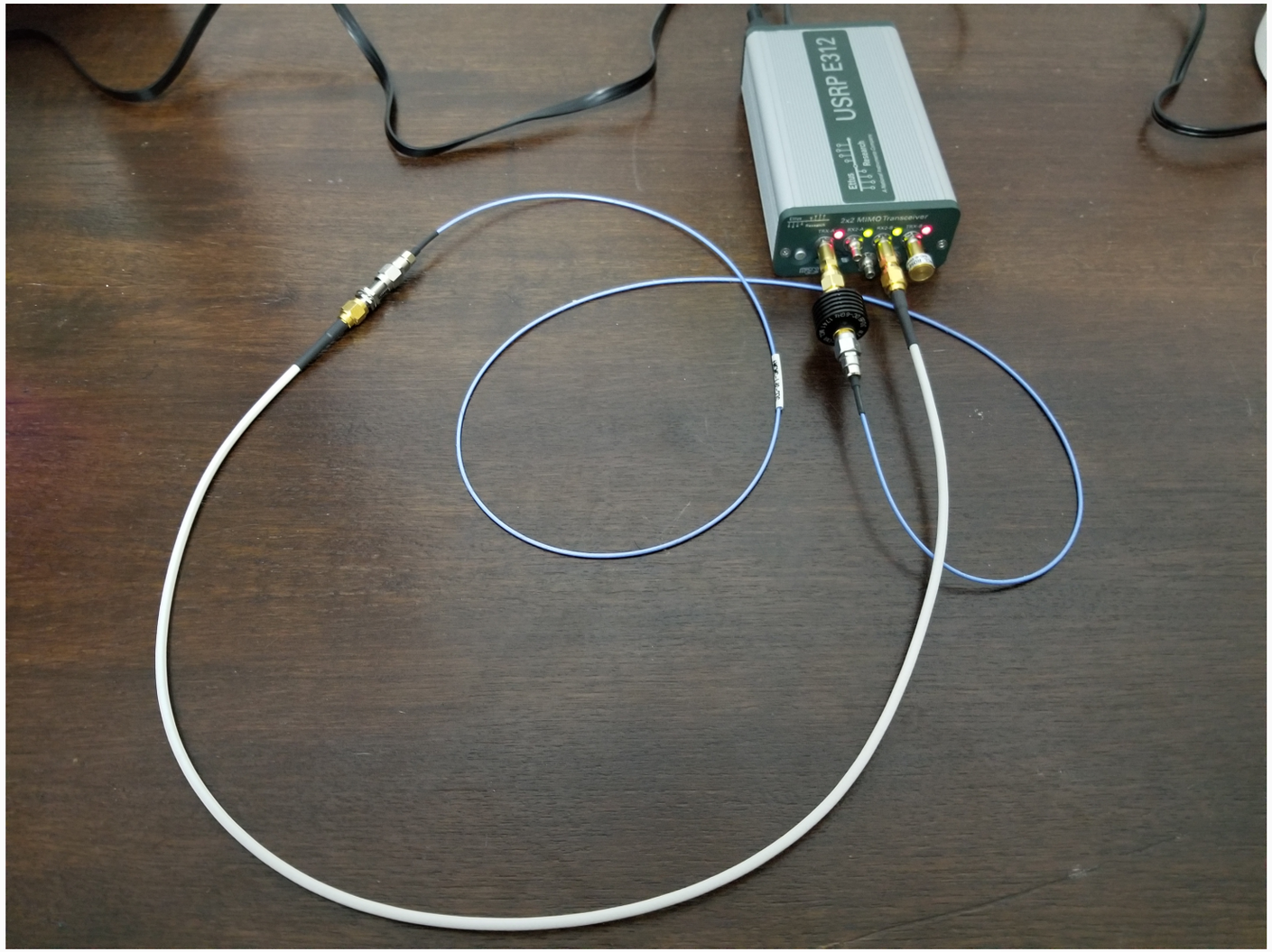


Example 1 – Post Process

- 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.
- Read file example Python code - https://github.com/spiro-sarris/utils/blob/master/iqstep.py

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



Example 2 – Cable Length

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

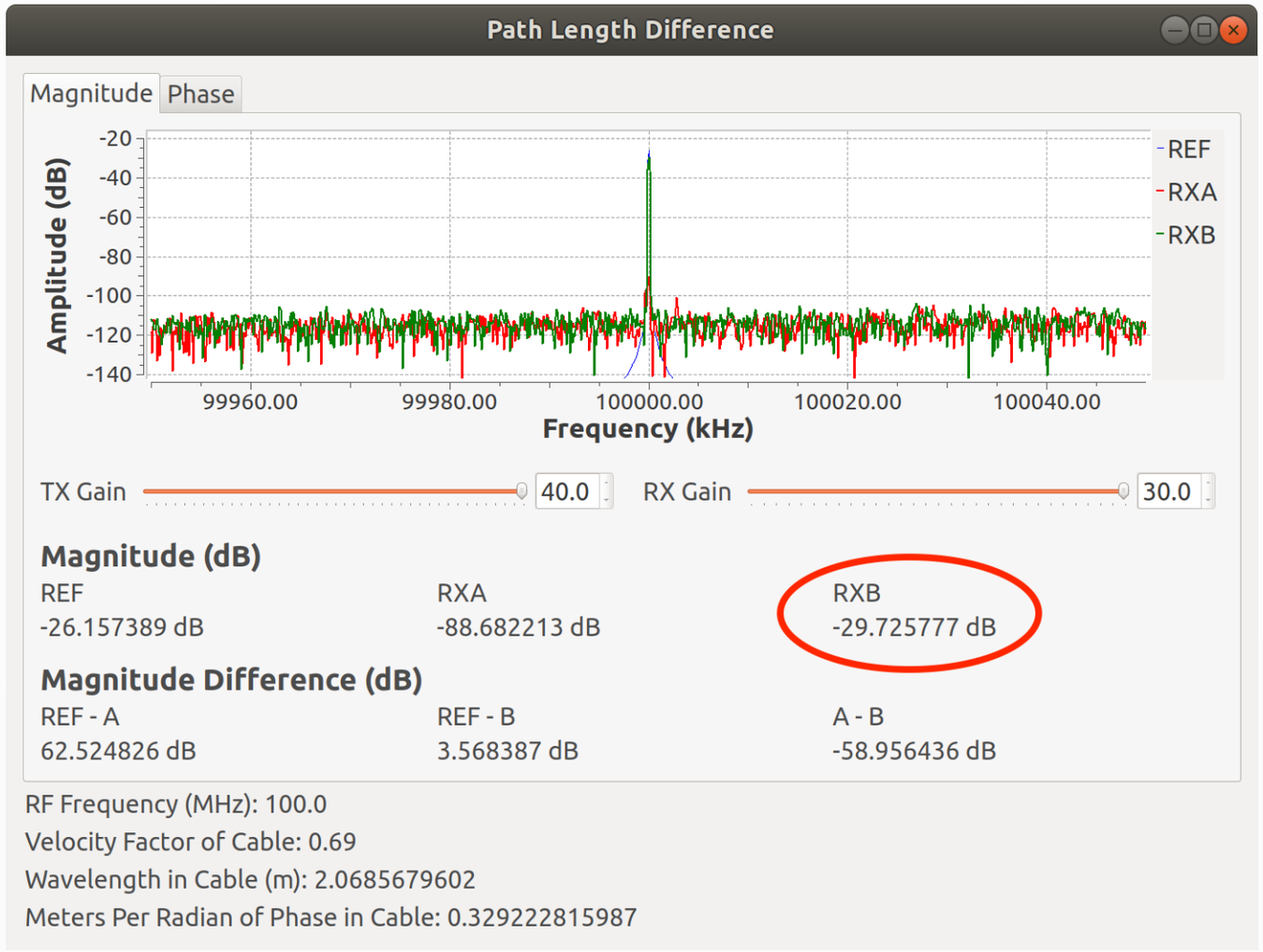
$$\lambda = \frac{c * vf}{f}$$

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

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

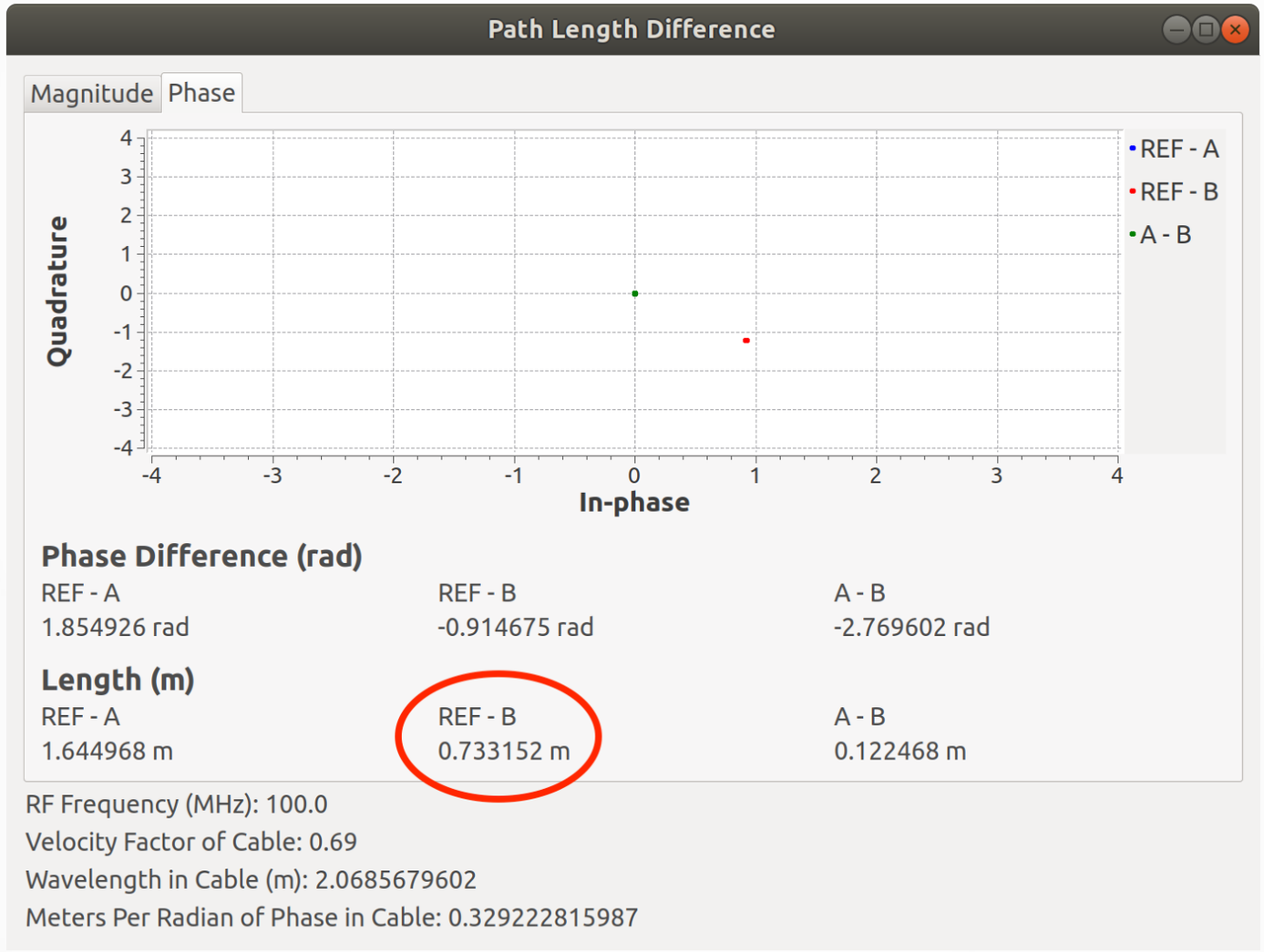
Example 2 – Cable Length

- Relative amplitude of calibration cable = -29.73 dB



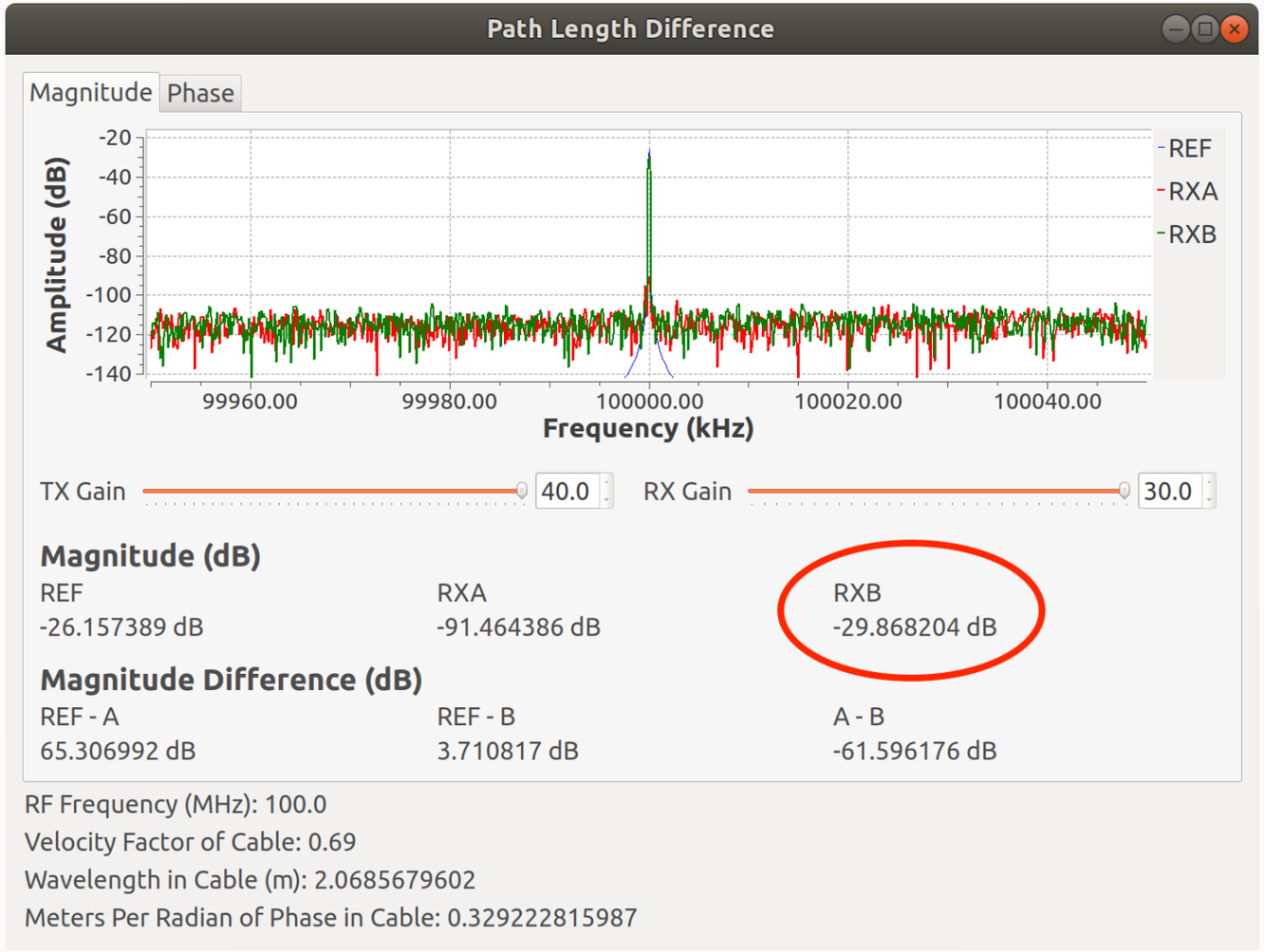
Example 2 – Cable Length

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



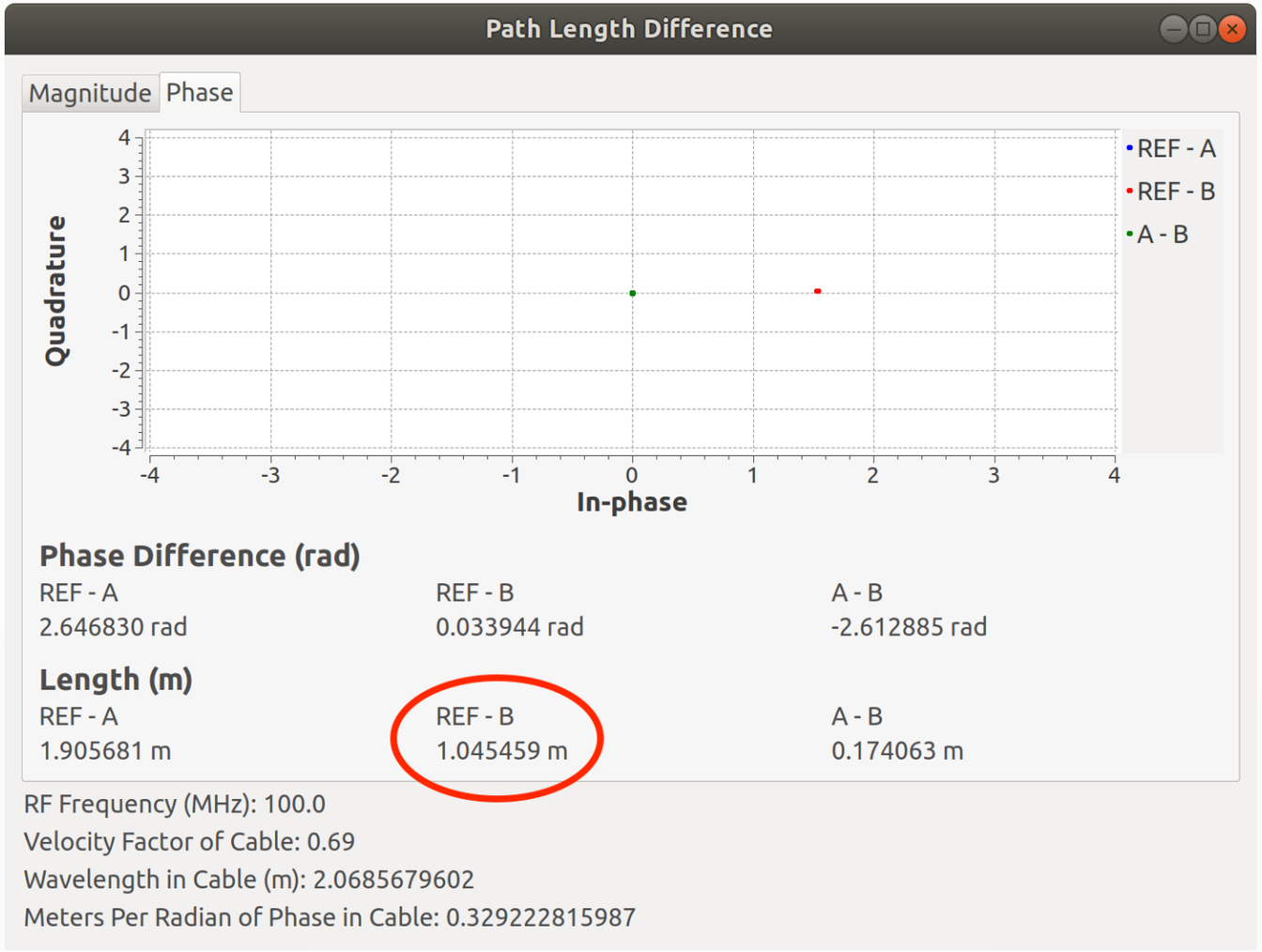
Example 2 – Cable Length

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



Example 2 – Cable Length

- Observe additional phase and length of DUT cable. L = 104.5 cm



Example 2 – Cable Length

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