Installation and usage: Spectro-Radiometer

Marcus Leech

Canadian Centre for Experimental Radio Astronomy

Introduction

This document describes the installation, system requirements, and usage of the *Gnu Radio* application known as **Spectro-Radiometer**[1].

System Requirements

This document assumes that you have already installed *Ubuntu 18.04 LTS* onto your system. The instructions given here would need to be modified for other *Linux* distributions, and *Ubuntu* is one of the more popular distributions.

The system hardware should be a fairly-modern, multi-core system, preferably X86-based, but some of the higher-end ARM SBCs, like the *Odroid N2* and *XU4* will also work, albeit at lower sample rates.

There should be at least 2G of system memory, with a basic clock speed of at least 1.4GHz for X86 systems, and 1.8GHz for ARM systems.

Caveats

This document **DOES NOT** cover basic system installation procedures and requirements. If you aren't familiar with Ubuntu 18.04 installation procedures, and basic system management tasks, you should **BECOME FAMILIAR WITH THOSE FIRST**.

SDR requirements

The software supports many different SDR hardware platforms:

- USRP products: B2xx series, and N2xx, X3xx and N3xx series
- RTLSDR dongles
- HackRF
- LimeSDR and LimeSDR-mini
- AirSpy R2 and AirSpy mini

This document will **NOT** cover installation requirements for this hardware, as that information is

1 See: https://github.com/Radio-Source/Spectro-Radiometer

generally provided by the manufacturers.

Prerequisite Installation

There are certain prerequisites for the **Spectro-Radiometer** software that must be satisfied before installing the software.

Development Tools

```
sudo apt install build-essential
sudo apt install git
sudo apt install python3-pip
python3 -m  pip install ephem
```

Gnu Radio

```
sudo apt install gnuradio gnuradio*
```

Application: *Spectro-Radiometer*

In your home directory:

```
git clone https://github.com/Radio-Source/Spectro-Radiometer
cd Spectro-Radiometer

Spectro-Radiometer is a Python3 script run by

python3 spectro_radiometer.py (if you use RTL-SDR with default [args])
python3 spectro_radiometer.py [args*] --> see documentation for [args] (if you
use another type of SDR  and or using some [args] differents)
```

Invoking Spectro-Radiometer

The application takes many options and command line parameters:

```
 -h, --help            show this help message and exit

  --abw=ABW             Set Analog  bandwidth [default=4.0M]

  --antenna=ANTENNA     Set Antenna [default=RX2]

  --baseline=BASELINE   Set Baseline length [default=99.3]

  --bbgain=BBGAIN       Set Baseband Gain [default=5.0]

  --bw=BW               Set Bandwidth [default=-1.0M]

  --dcg=DCG             Set Detector DC Gain [default=100]

  --decln=DECLN         Set Observing Declination [default=0.0]

  --device=DEVICE       Set SDR Device Name [default=rtl=0
```

```
                        file=/dev/zero,rate=5e6]
  --fftsize=FFTSIZE     Set FFT size [default=2048]
  --frequency=FREQUENCY
                        Set Center Frequency [default=1.42041G]
  --gain=GAIN           Set RF Gain [default=30.0]
  --ifgain=IFGAIN       Set IF Gain [default=5.0]
  --latitude=LATITUDE   Set Local Latitude [default=44.9]
  --longitude=LONGITUDE
                        Set Local Longitude [default=-76.03]
  --mode=MODE           Set Operation Mode [default=total]
  --prefix=PREFIX       Set Data File Prefix [default=h1]
  --ra=RA               Set Target RA [default=12.0]
  --rfilist=RFILIST     Set RFI Frequency List [default=]
  --srate=SRATE         Set Sample rate [default=2.56M]
  --zerotime=ZEROTIME   Set SIdereal time for auto baseline set [default=99.3]
  --clock=CLOCK         Set Clock Source [default=default]
  --ppstime=PPSTIME     Set Time Source [default=default]
```

The *–abw* parameter controls the analog bandwidth, in Hz, on the hardware—this options does nothing on some hardware, and actually sets the pre-ADC bandwidth on others.

The *–antenna* option controls which antenna port is to be used. Again, this “does nothing” on some hardware, and on others selects which antenna port to use.

The *–baseline* option is used in interferometer mode to calculate fringe-stopping parameters, and is in meters.

The *–bbgain* parameter is used to control the *baseband gain* some hardware supports it, some doesn't.

The *–bw* parameter controls the DSP filter bandwidth, in Hz, prior to detection.

The *–dcg* parameter controls a digital “gain” on the detected DC total power output.

The *–decln* parameter sets the astronomic declination associated with the observation.

The *–device* parameter specifies the device string to use with the *osmosdr* hardware source. *Further discussion of this later in the document.*

The *–fftsize* parameter sets the number of FFT bins used for spectral determinations. The default value is adequate for most use cases.

The *–frequency* parameter sets the tuned center frequency, in Hz.

The *–gain* parameter sets the RF gain of the hardware device, in dB, typically.

The *–ifgain* parameter sets the IF gian of the hardware device, in dB. This is only supported by some types of hardware.

The *–latitude* parameter sets the local geographic latitude of the observatory.

The *–longitude* parameter sets the local geographic longitude of the observatory.

The *–mode* parameter sets the operating mode. For a single receiver, choose "total". If you have two receiver channels, then choose either "differential" or "interferometer".

The *–prefix* parameter sets the file-name prefix used, including any directories.

The *–ra* parameter defines the observation RA – used mostly for documenting observations.

The *–rfilist* parameter specifies a list (separated by commas) of frequencies that contain RFI, and should not be counted when calculating total power across the spectrum.

The *–srate* parameter specifies the hardware sample-rate. Different hardware supports different rates, and generally higher rates require a faster/better computer host.

The *–zerotime* parameter can be used to specify an LMST during which no spectral components are expected to show up in the beam--used for (optional) **baseline-subtraction** in spectral data. This is only really useful for spectral line work, such as with the *hydrogen line*. If the value is invalid (which is the default), no such baseline-recording will be performed.

The *–clock* parameter specifies the reference clock source. Not all hardware supports this option.

The *–ppstime* parameter specifies the source of the 1PPS signal. Not all hardware supports this option.

Device options

The *Spectro-Radiometer* application uses the *gr-osmosdr* abstraction within Gnu Radio to provide support for many different types of SDR hardware.

The **general** form is:

```
--device "hwstring1 hwstring2"
```

Now, since *Spectro-Radiometer* supports **two** channels, in many cases, with only a single receiver, the 2nd *hwstring*, above should be: "file=/dev/zero,rate=20e6".

Single RTL-SDR

For a single *RTL-SDR* device, the –device option looks like:

```
--device "rtl=index-or-name file=/dev/zero,rate=20e6"
```

Recall that for RTL devices, one can either specify the device *index* which is simply the index from the order in which RTL devices were **enumerated** at boot time, or you can use the serial-number of the device, like “SKY1” or “ASTRON”, etc. See the *rtl_eeprom* tool for setting the serial number to allow unique device identification.

Dual RTL-SDR

With a dual-*RTLSDR* configuration, you can do differential measurements, with one device perhaps terminated in a load, or connected to a reference dish feed, or reference antenna, etc:

```
--device “rtl=index-or-name1 rtl=index-or-name2”
```

The key thing to understand is that the first device is always the notional **sky** channel and the 2nd device is always the notional **reference** channel**.**

You will also specify:

```
--mode differential
```

On the command line, to indicate that differential observations should be used.

Single AirSpy

To use an *AirSpy*:

```
--device “airspy=0 file=/dev/zero,rate=20e6”
```

You will need to specify a *–srate* option that is compatible with the hardware. For *AirSpy R2*, available rates are 10e6 and 2.5e6. For *AirSpy Mini* available rates are 3.0e6 and 6.0e6.

USRP B2xx

For a USRP B200, using the osmocom-UHD driver:

[single channel]

```
  --device “uhd,type=b200,num_recv_frames=128,subdev='A:A'
  file=/dev/zero,rate=20e6”
```

```
[dual channel]
   --device “uhd,type=b200,num_recv_frames=256,nchan=2,subdev='A:A A:B'”
```

In the 2nd, dual-channel case, this can be in support of either *differential* or *interferometer* mode. Use the *–mode* option appropriately.

The B2xx series is very *sample-rate agile*, so it's possible to pick a sample rate that's very close matching the compute hardware and local-RFI environment.

USRP1 with WBX/SBX/CBX/RFX cards

For a USRP1, using the osmocom-UHD driver:

```
--device “uhd,type=usrp1,num_recv_frames=128,subdev=A:0
file=/dev/zero,rate=20e6”
```

Dual channel support has not been tested on the USRP1, and requires different FPGA firmware.

Sample rates can be any proper integer fraction of 64MHz, but rates above 8Msps aren't really supported.

LimeSDR

For a LIMESDR-USB (NOT the mini):

[single channel]

```
--device “soapy,driver=lime”
```

```
[dual channel]
--device “soapy,driver=lime,nchan=2”
```

Both *differential* and *interferometer* modes are supported.

You will also have to specify the appropriate antenna port, using the –antenna option. My preference is to use the “LNAW” port:

```
--antenna LNAW
```

The *LimeSDR* is reasonably sample-rate agile, very much like the USRP B210 in this regard.

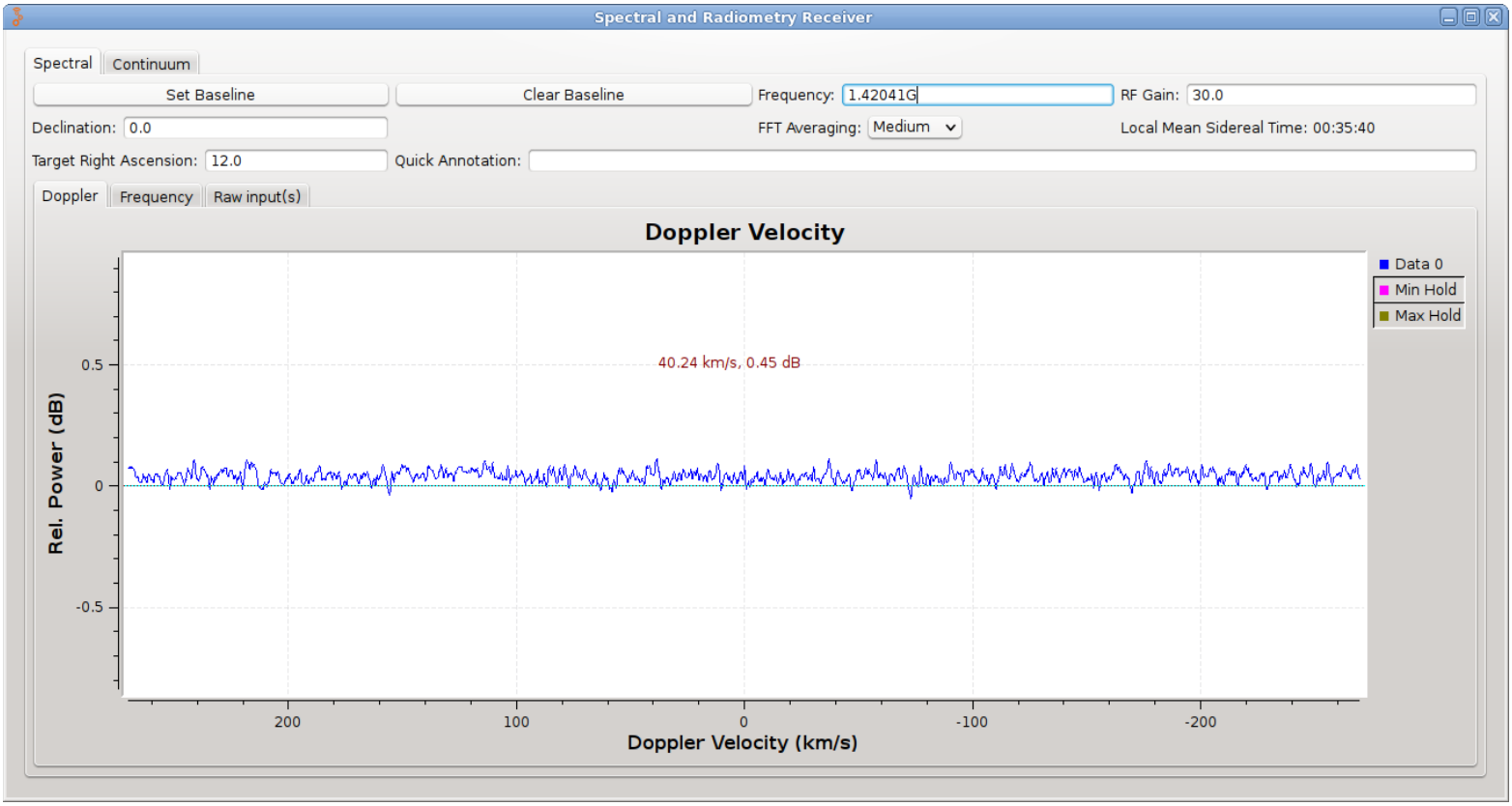
Other Hardware

Other hardware that is supported by *gr-osmosdr* will probably also work, but hasn't been tested explicitly with *Spectro-Radiometer.*

User Interface

The application provides a *GUI*, based on the standard *Qt* widgets provided by *Gnu Radio*.

Main Window



This panel provides access to various spectral plots, showing (possibly base-lined) Doppler and Frequency views of the same spectrum, as well as “raw” spectral information from both devices.

In the case of the *Doppler* and *Frequency* spectral displays, only the notional “Sky1” channel spectrum is displayed (or the 1st channel in the case of interferometry).

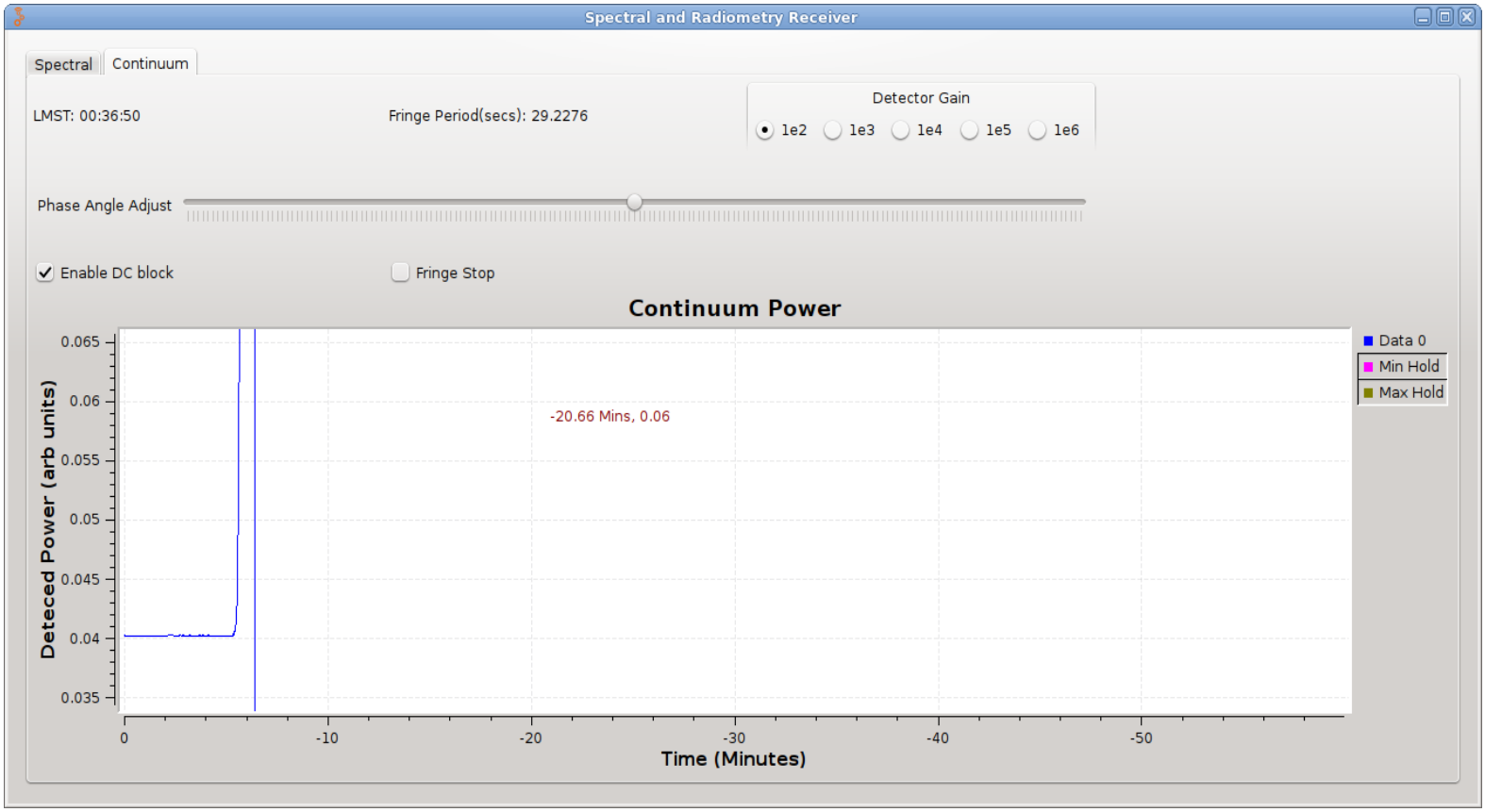
The display itself is governed by the behavior of a Qt plot widget. The middle mouse button pops up a menu to control such things as auto-scaling, axis limits, etc. More can be found in *Gnu Radio* documentation.

The *FFT Averaging* control controls the averaging applied to the **displayed** spectral information. The *Set Baseline* records a notional *baseline* spectrum, which is subtracted from the incoming spectrum to provide a flattening and RFI-excision function for display purposes. The *Clear Baseline* function clears this recorded baseline.

The *Frequency* and *RF Gain* controls set the appropriate parameters in the hardware. Like most such numerical inputs, exponential notation is accepted (1420.4058e6, for example).

The *Quick Annotation* input is used to add time-stamped annotation to an annotation file that is an adjunct to the observing data.

Continuum Window



This panel shows the Continuum *Total Power, Differential Power,* or *Correlation* depending on the operating mode of the software. For simplicity, only the *Cos* component of the complex correlation (interferometer mode) is displayed, but both *Cos* and *Sin* components are actually logged.

The *Enable DC Block* control applies only to *interferometer mode* and applies a high-pass (DC block) component to the correlation calculations. Since useful observations will have a decidedly-non-zero fringe rate, this helps to filter out long-term DC offset drift.

The *Phase Angle Adjust* control is used to adjust the relative phase angle between the two channels, and again is only applied in *interferometer* mode.

The *Fringe Stop* control is used to apply a fringe-stopping rotation to the relative phase-angle between the two channels in *interferometer* mode. It must have correct inputs for *baseline-length*, *frequency*, *declination* and *target RA*, and local *latitude* and *longitude* in order to operate correctly.

The *DC Gain* control is used to apply a simple numerical multiplier to the values displayed and logged in the data files for the various continuum data products: *total-power* for A and B, *Differential*, and the *complex correlations*.

The display is again a *Qt plot* display, and the same controls as in the spectral display can be used—middle mouse button pops up a control panel menu, etc.

Data Files

The application produces data files for both spectral and total-power/differential/interferometer data.

Files are written with whatever prefix was specified with the *–prefix* option, described earlier, and a generated name. A new file is begun at 00:00:00 UTC.

The **spectral** data are written into files:

YYYYMMDD-spec.csv

Each record in the file consists of a record header:

UTC(HH,MM,SS),LMST(HH,MM,SS),frequency-in-MHz,bandwidth-in-Hz,declination,

Followed by a large number (FFT size) of comma-separated values in dB.

These records are produced every 20 seconds.

The **total-power/differential/interferometer** records are produced in a file:

YYYYMMDD-tp.csv

Each record in the file consists of a record header:

UTC(HH,MM,SS),*LMST*(HH,MM,SS),frequency-in-MHz,bandwidth-in-Hz,declination,

Followed by various items of data:

tpa,tpb,diff,corr_cos,corr_sin

Where *tpa*,*tpb* is the total power calculated for the first and second channels, *diff* is the difference of these channels (a-b). When in single-receiver mode, the *tpb* channel will always be 0, and therefore the *diff* channel will be the same as the *tpa* channel. The *corr_cos* and *corr_sin* are the *cosine* and *sine* components of the complex correlation of the two channels—these will only be meaningful when in *interferometer* mode.

These records are produced every 2 seconds.