

Software Defined Radio

How Luke Learned To Love the Source

Marcus Leech

Science Radio Laboratories

<http://www.science-radio-labs.com>

In The Beginning...

- A long time ago, in a laboratory not so far away
 - Marconi was messing with metal plates
 - Fessenden was futzing with FM
 - Armstrong was analyzing the Audion
 - Bardeen and Shockley had seized on semiconductors
 - **THERE WAS HARDWARE....**
 - Lots and lots and lots of hardware
 - Incalculable masses of coils and capacitors, and resistors and glass and plastic and ceramics and...

The Evolution of DSP

- Throughout 1980s and 1990s, more “ceramic and silicon” functions moved into the digital domain.
 - Many amateur receivers had “DSP” options, that allowed various types of IF tuning and audio processing
 - Commercial world used DSP techniques all over the place—radios, modems, TVs, VCRs, etc.
 - Payoff was in *flexibility*, and not always overall project costs.
 - BOM costs reduced
 - Software Development costs increased (sometimes infinitely!!!)

Sampling the Analog World

- Way back in 1924, Nyquist (and later Shannon and Whittaker) discovered that:
 - For a signal of F (Hz) bandwidth
 - You only need $2 \cdot F$ samples/sec to be able to reconstruct the signal
 - Led partially to the birth of *Information Theory*
 - Important result in digital systems using sampling techniques.

So, what does this mean?

- Signals can be processed in the digital domain
 - Signal processing in the analog domain uses functions that are ***analog approximations*** of precise mathematical functions
 - Some implementations are higher fidelity (better approximations) than others.
 - Difference between \$100.00 stereo and \$1000.00 stereo is in improving the quality of those *analog approximations*.
 - Converting to digital domain as early as possible gives you better control of mathematical fidelity.
 - Gives you *tremendous flexibility*, without having to break out the soldering iron!

So, what's SDR, then?

- Functionally identical to DSP that has been around for a couple of decades.
 - Different implementation
 - Use (as much as possible) general-purpose compute hardware, rather than custom/semi-custom DSP processors.
 - Modern desktop hardware has some pretty impressive capabilities for DSP work:
 - Even modest desktop systems are capable of several hundred MFLOPS (Million FLOating-point Operations Per Second).
 - A few Giga-FLOPs common on higher-performance general purpose computing gear.

Where do I plug in the antenna?

- Does that mean I can receive 1296Mhz FM on my computer directly? Where do I plug it in.
- Sadly, **some** hardware is still required!
- Need high-speed A/D and D/A I/O in and out of the machine
 - Need a way to downconvert/upconvert signals to bands of interest.
 - Typically use *Direct Conversion* techniques.
 - Signals represented in *baseband quadrature* form.
 - Signals are *in quadrature* (90 relative phase shift).

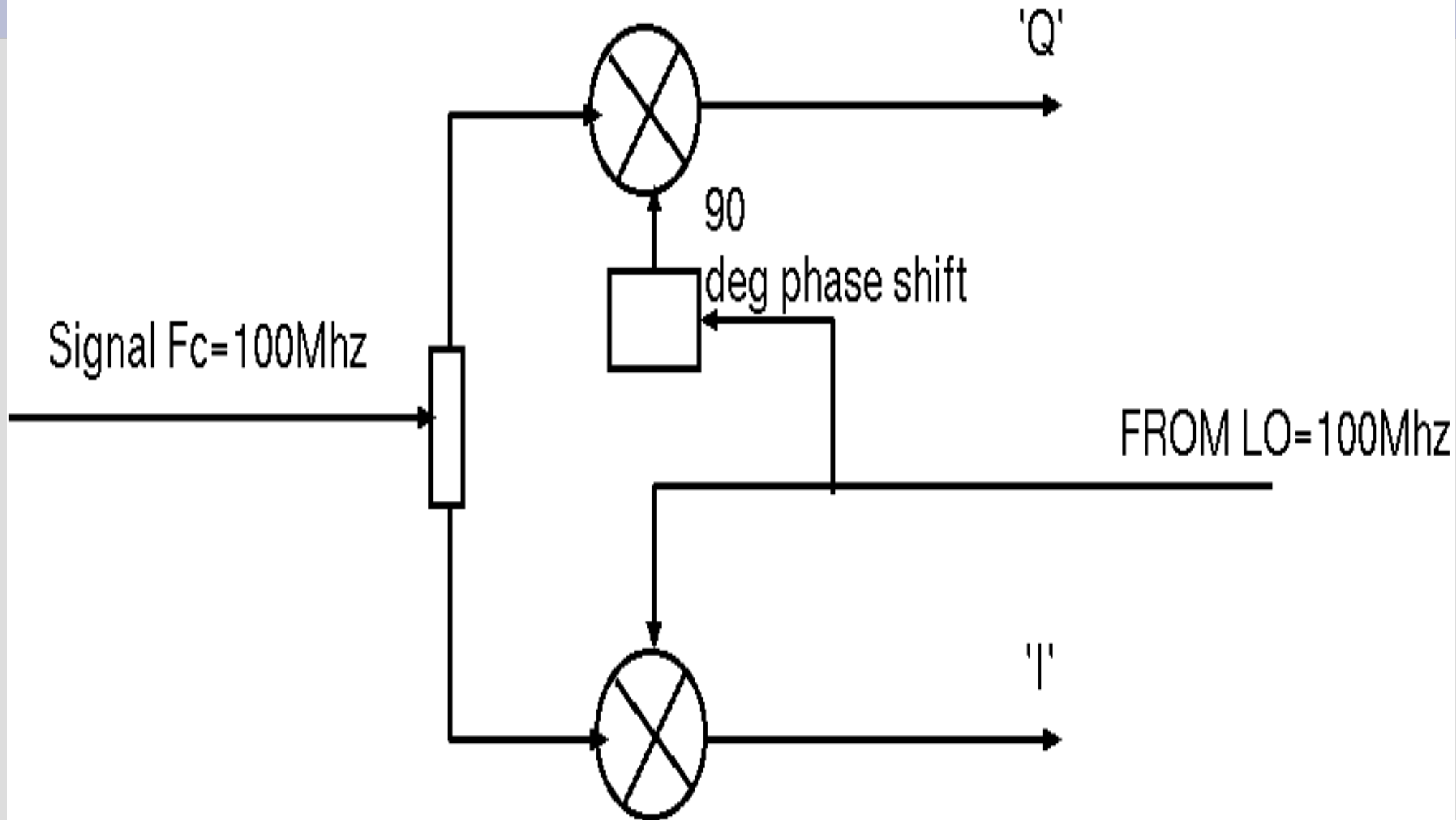
Watch your Is and Qs.

- Signals have an in-phase (I) and quadrature (Q) component.
- Quadrature signal of bandwidth BW:
 - Starts at $F_c - BW/2$
 - Ends at $F_c + BW/2$
- Use simultaneous-sampling A/D and D/A hardware to maintain phase relationship.
 - Very common these days, **because** of SDR!!
- Signal up/down conversion results in:
 - Signals centered around “DC”
 - Lowest component $-BW/2$, highest component $+BW/2$

Signal Quality

- Both amplitude and phase mis-match between I and Q can degrade system performance.
 - Controlled by that ugly-old **analog** world!
 - NCOs (Numerically-controlled Oscillators) used to provide well-phased down/upconversion LOs.
 - Amplitude imbalance is at the mercy of amplifier design.
 - Amplitude and Phase imbalance can be compensated for later digitally, at least partially.

Typical D-C receiver front-end



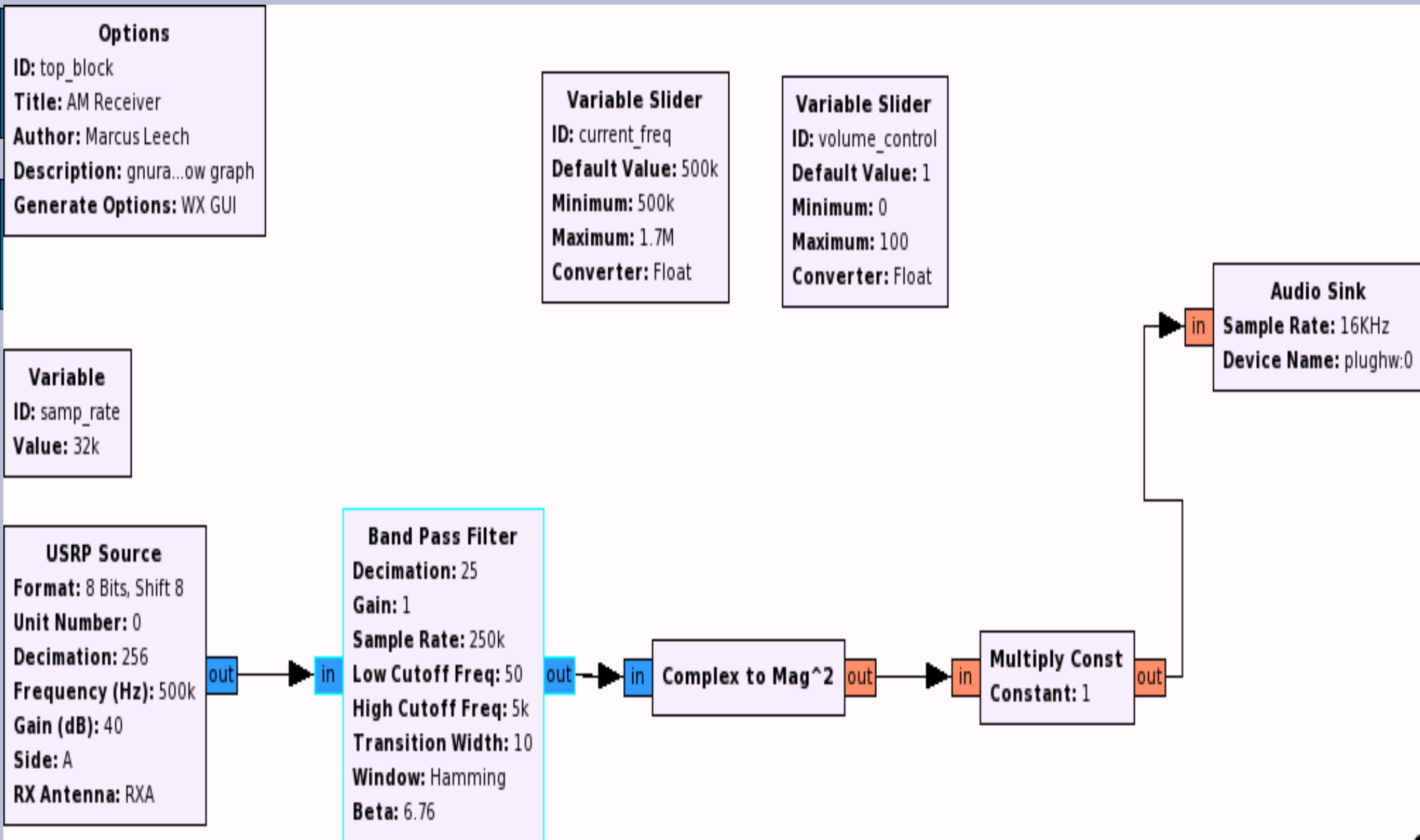
Products for the Amateur

- SDR-1000, FLEX-5000, FLEX-3000
 - Transceiver based on SDR
- SDR-14, SDR-IQ, SDR-IP
 - Receive only
- USRP and USRP2
 - General-purpose receiver, transceive, transmit
 - Daughter-cards map into various bands
 - USRP uses 480Mbit USB-2.0 for PC I/O
 - USRP2 uses 1Gbit Ethernet for PC I/O
 - It's my favourite :-)

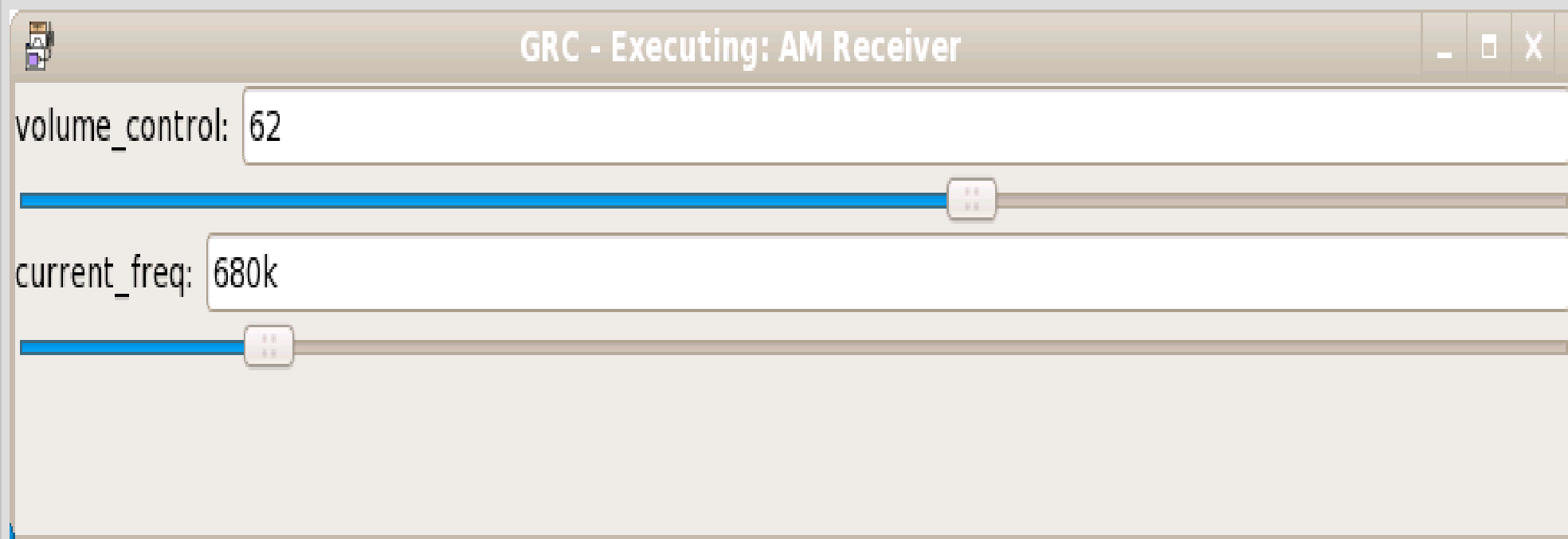
SDR development frameworks

- Many commercial frameworks
- Only ONE Open Source framework
 - Gnu Radio (<http://www.gnuradio.org>)
 - Has dozens of different signal-processing blocks
 - Most critical blocks have been ruthlessly optimized for various PC hardware.
 - Now has a GUI-based “constructor” to make life easier for running experiments.
 - A little bit like playing with LEGO
 - Somewhat like LABView or MATLAB
 - Needless to say, it's the only one I use!!

What's after the front-end?



Running AM Receiver



The image shows a screenshot of a software interface window titled "GRC - Executing: AM Receiver". The window contains two sliders. The first slider is labeled "volume_control:" and has a value of 62. The second slider is labeled "current_freq:" and has a value of 680k. Both sliders have a blue bar indicating the current value and a small square handle with two dots.

volume_control: 62

current_freq: 680k

Adding FFT Features

Options
ID: top_block
Title: AM Recei...us spectrum
Author: Marcus Leech
Description: gnura...ow graph
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32k

Variable Slider
ID: current_freq
Default Value: 1.42G
Minimum: 1.415G
Maximum: 1.425G
Converter: Float

Variable Slider
ID: volume_control
Default Value: 1
Minimum: 0
Maximum: 100
Converter: Float

Variable
ID: bw
Value: 500k

Band Pass Filter
Decimation: 15
Gain: 1
Sample Rate: 500k
Low Cutoff Freq: 40
High Cutoff Freq: 14k
Transition Width: 100
Window: Hamming
Beta: 6.76

USRP Source
Unit Number: 0
Decimation: 128
Frequency (Hz): 1.42G
Gain (dB): 40
Side: A
RX Antenna: RXA

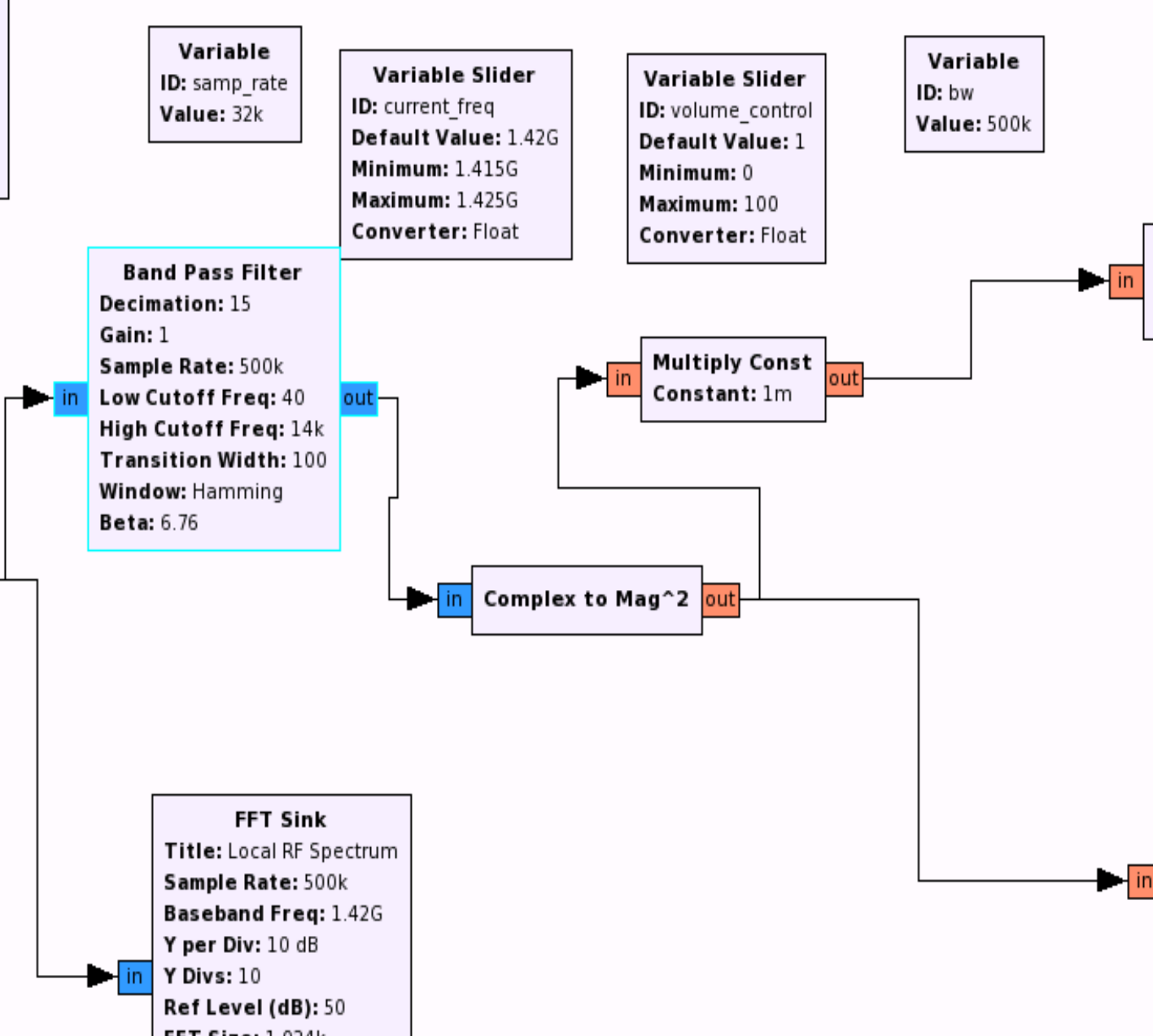
Complex to Mag²

Multiply Const
Constant: 1m

Audio Sink
Sample Rate: 32KHz
Device Name: plughw:0,0

FFT Sink
Title: Local RF Spectrum
Sample Rate: 500k
Baseband Freq: 1.42G
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 50
FFT Size: 1.024k
Refresh Rate: 2
Average: On
Average Alpha: 250m

FFT Sink
Title: Audio Spectrum
Sample Rate: 32k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 50
FFT Size: 1.024k
Refresh Rate: 10



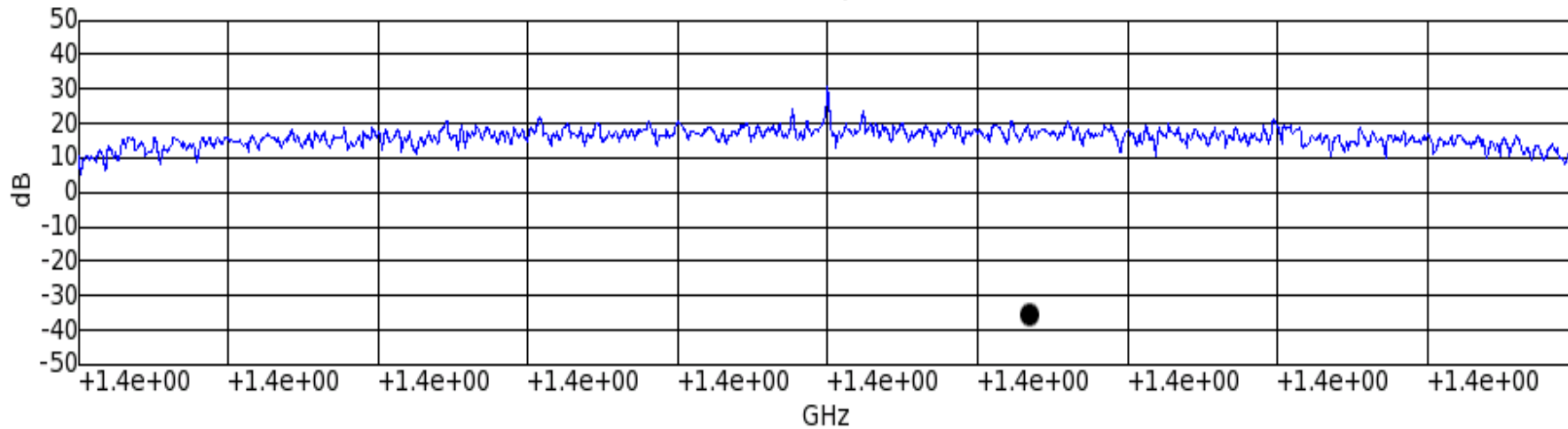
Running New Receiver

GRC - Executing: AM Receiver plus spectrum

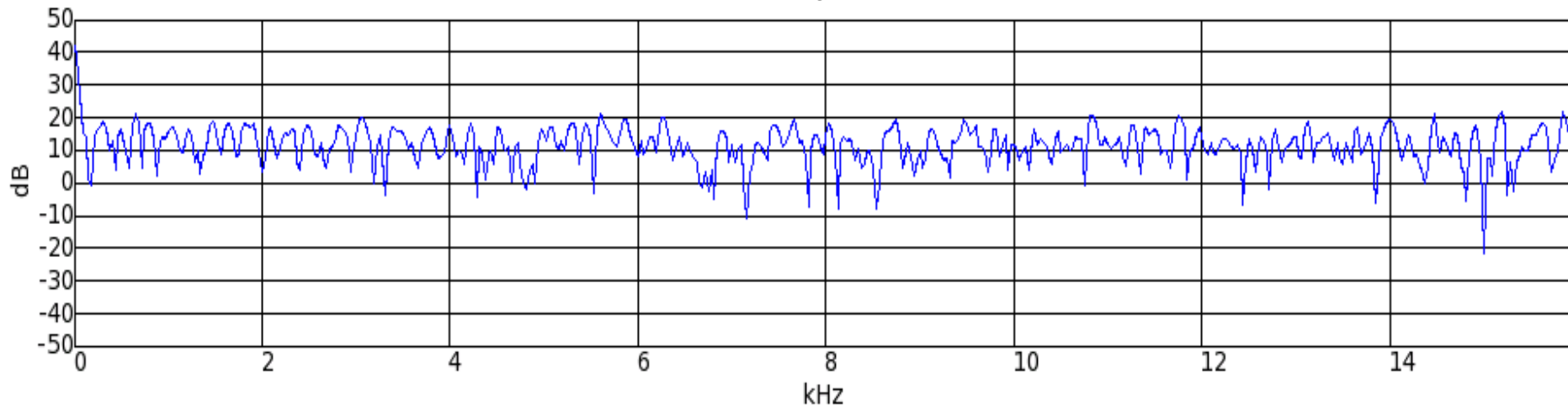
volume_control: 39

current_freq: 1.42G

Local RF Spectrum



Audio Spectrum



Options

Average

Peak Hold

Set dB/div

1 dB/div

2 dB/div

5 dB/div

10 dB/div

20 dB/div

Adj Ref L

+ -

Options

Average

Peak Hold

Set dB/div

1 dB/div

2 dB/div

5 dB/div

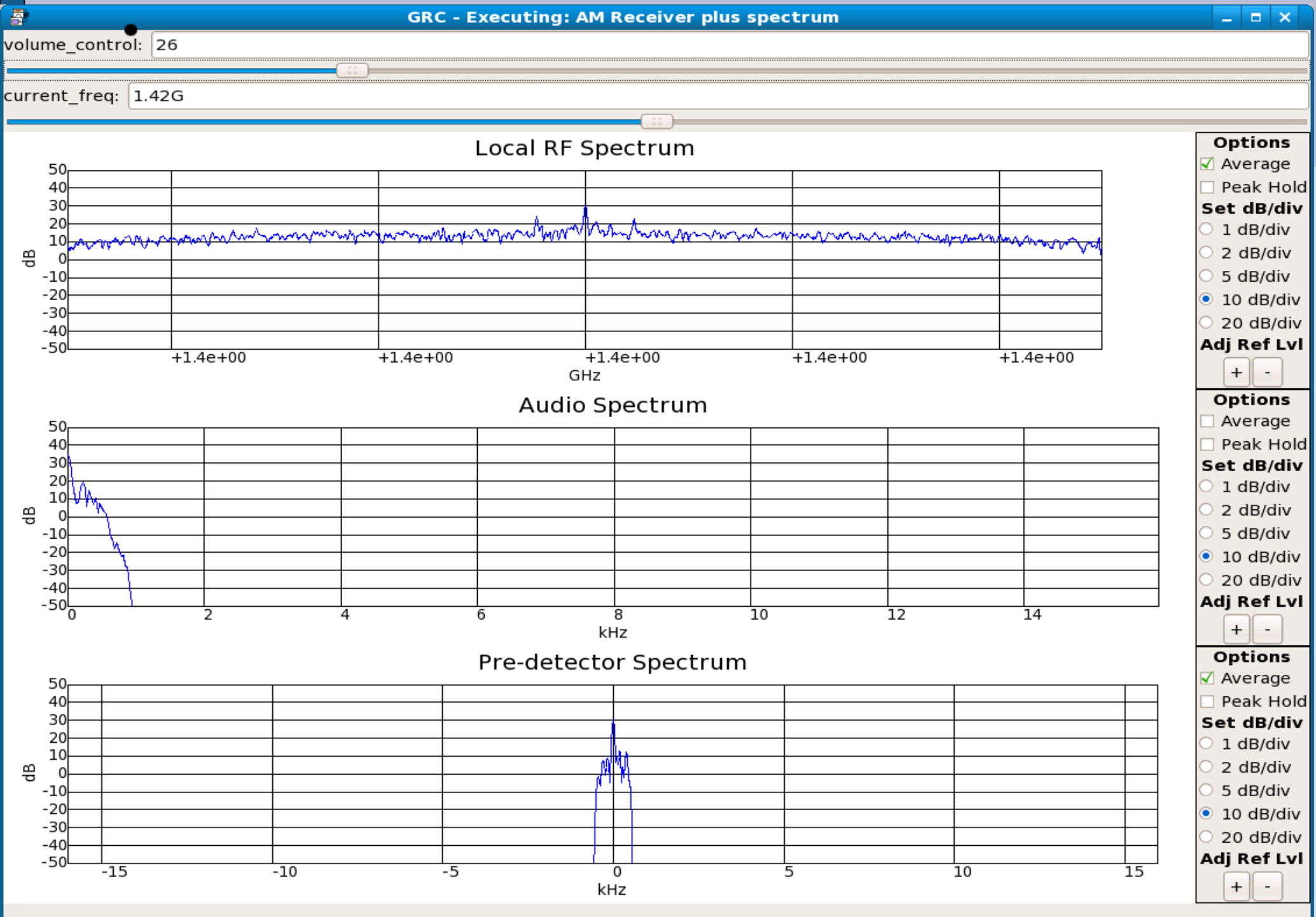
10 dB/div

20 dB/div

Adj Ref L

+ -

Narrow Filter



Flexibility: Different GUI

Form Design [-] [□] [X]

Detector Value [] **Peak Value** []

LMST [] **Run time** []

Pulsar Rate [] **Folding Const** [5]

DM [0.1]

Delay (ms) [] **PhaseErr** []

Show []

Transients

[]

Power Threshold

[2.5] []

[3] []

Transient Duration

[]

Current Declination []

Current RA []

RF Controls

Tuner Frequency (Hz) []

Sky Frequency (Hz) []

GC A []

GC B []

EXIT []

LOCK []

RF Gain [40]

Continuum Controls

Dicke Mode: OFF

[1] [] [] [] []

DC Gain Mult. [x1]

Show Int. []

Offset [0.0]

Integration [1]

Refmult [1.0]

Spectral Integration [1] [] **Show** []

STAT []

SETI Analysis **HELP** []

FREQ [1.00] **C RATE** [] **SIGS** []

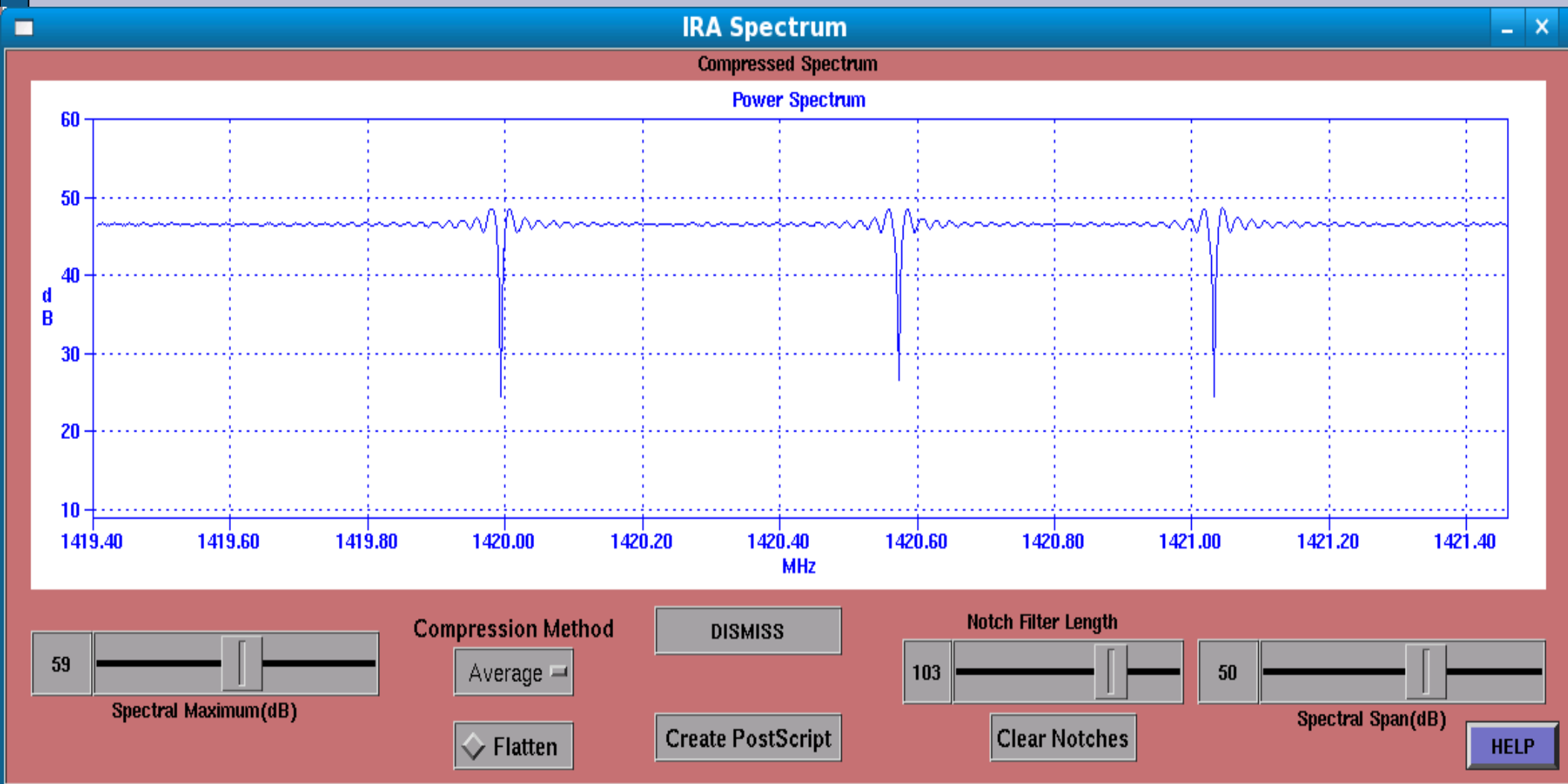
Sigma_K []

SETI Integration [15.00] **Show** []

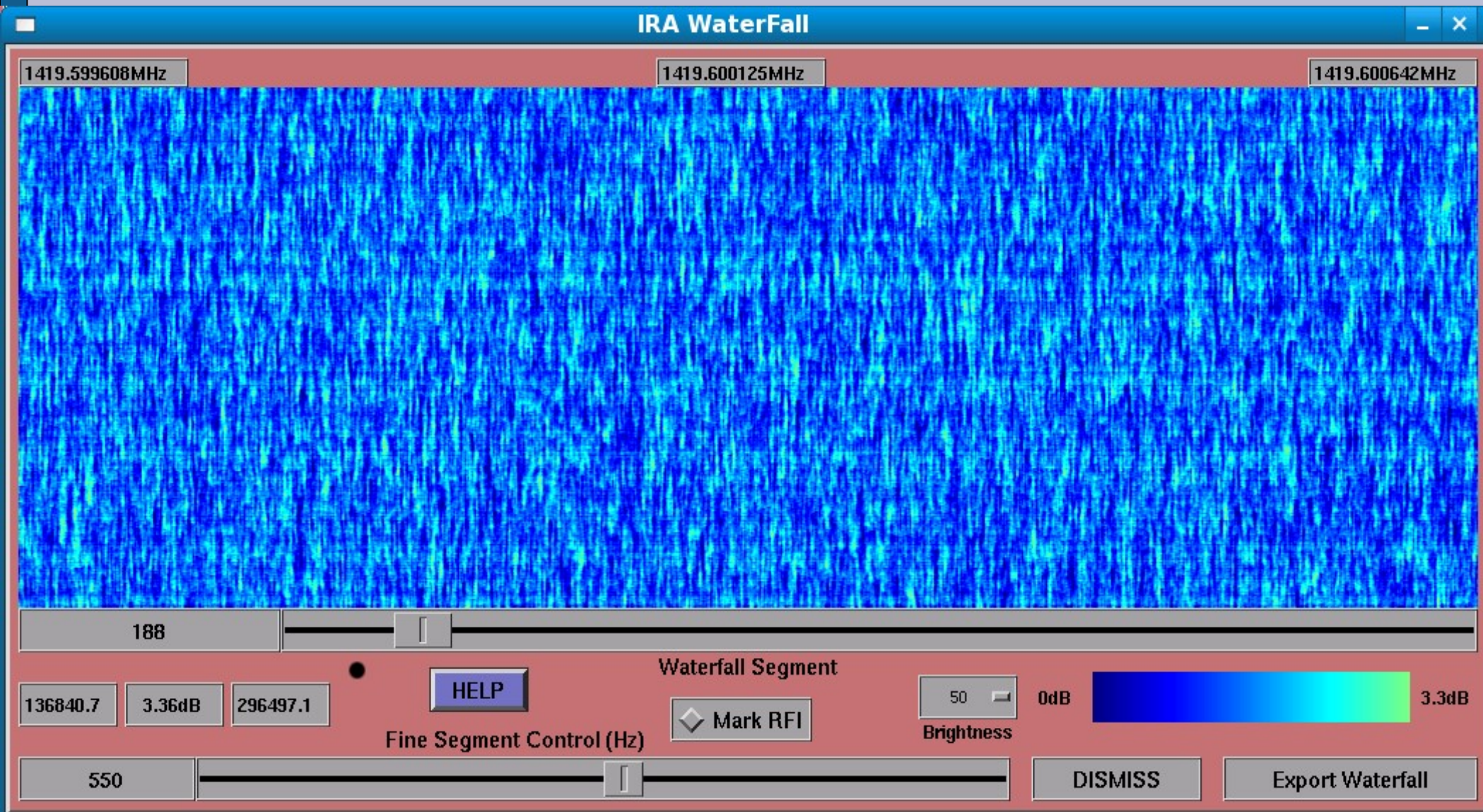
PLEASE WAIT.....

Science Radio Laboratories <http://www.science-radio-labs.com>

Sophistication limitless



...Really



Resources

- <http://www.gnuradio.org>
- <http://www.ettus.com>
- <http://www.science-radio-labs.com>
- http://www.tapr.org/kits_janus.html