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....**
    - <u>Lots and lots and lots of hardware</u>
    - <u>Incalculable masses of coils and capacitors, and resistors and glass and plastic and ceramics and...</u>

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*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).

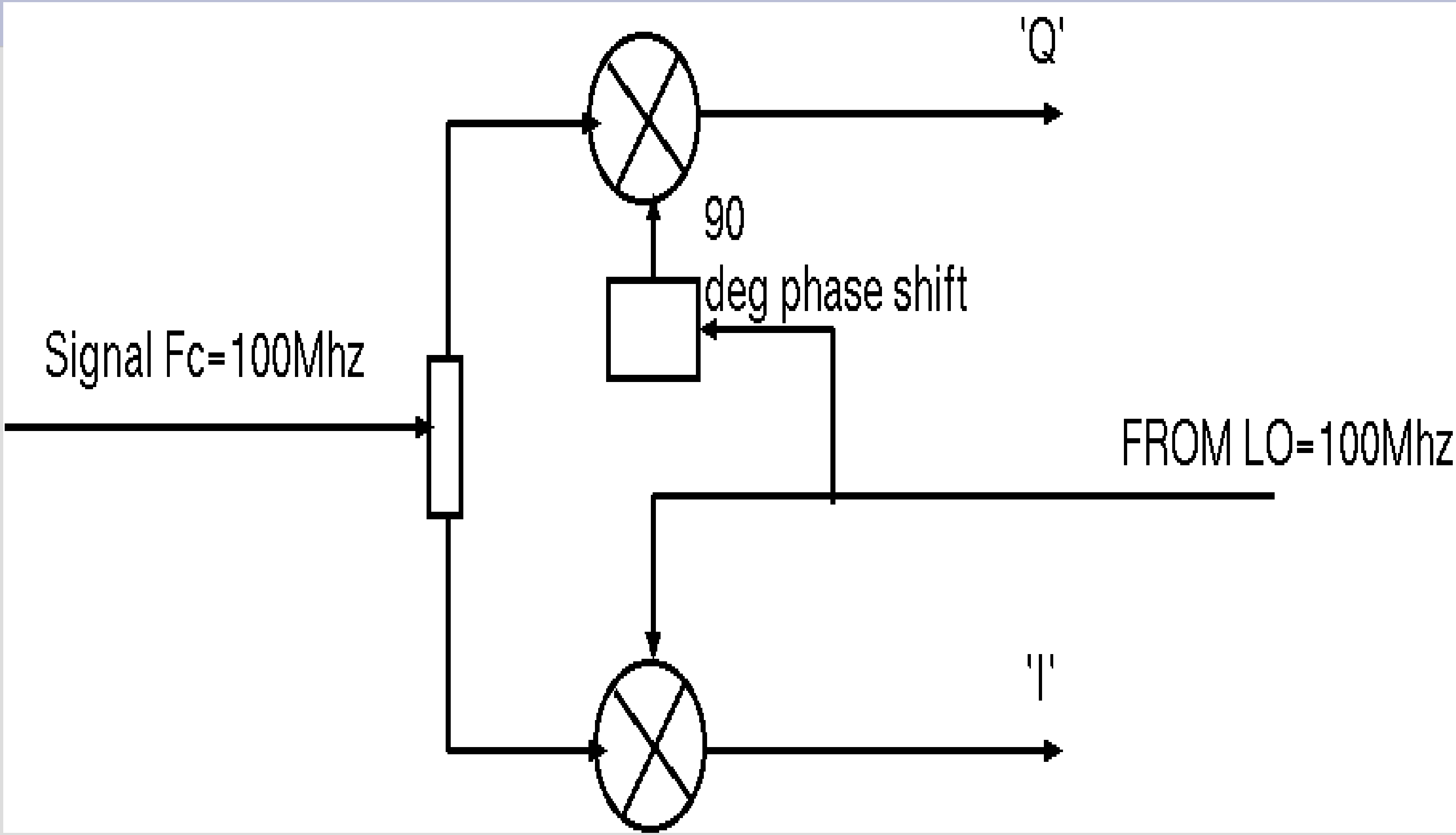
Watch your Is and Qs.

- Signals have an in-phase (I) and quadrature (Q) component.
- Quadrature signal of bandwidth BW:
  - Starts at Fc – BW/2
  - Ends at Fc + 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



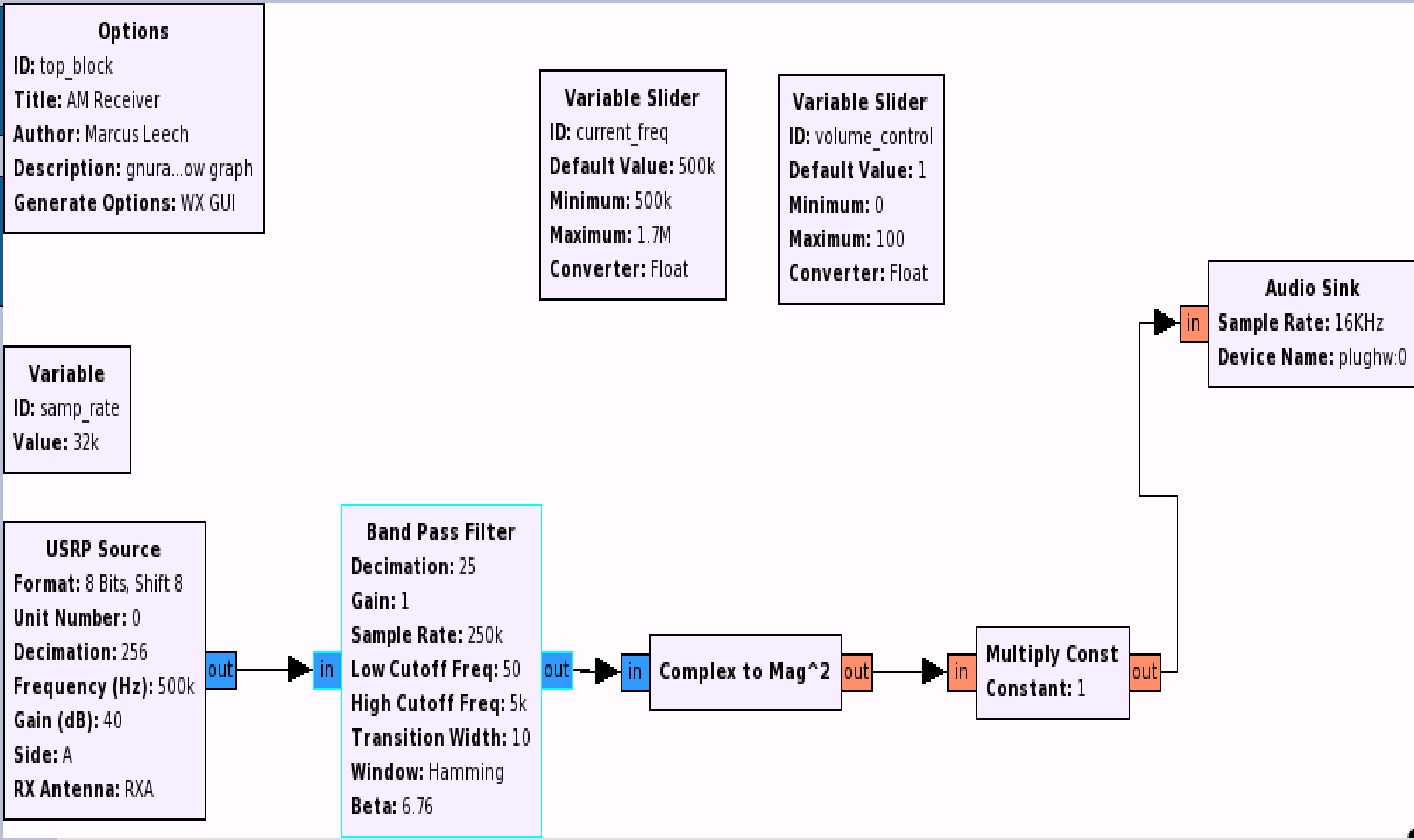
Products for the Amateur

- 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 :-)

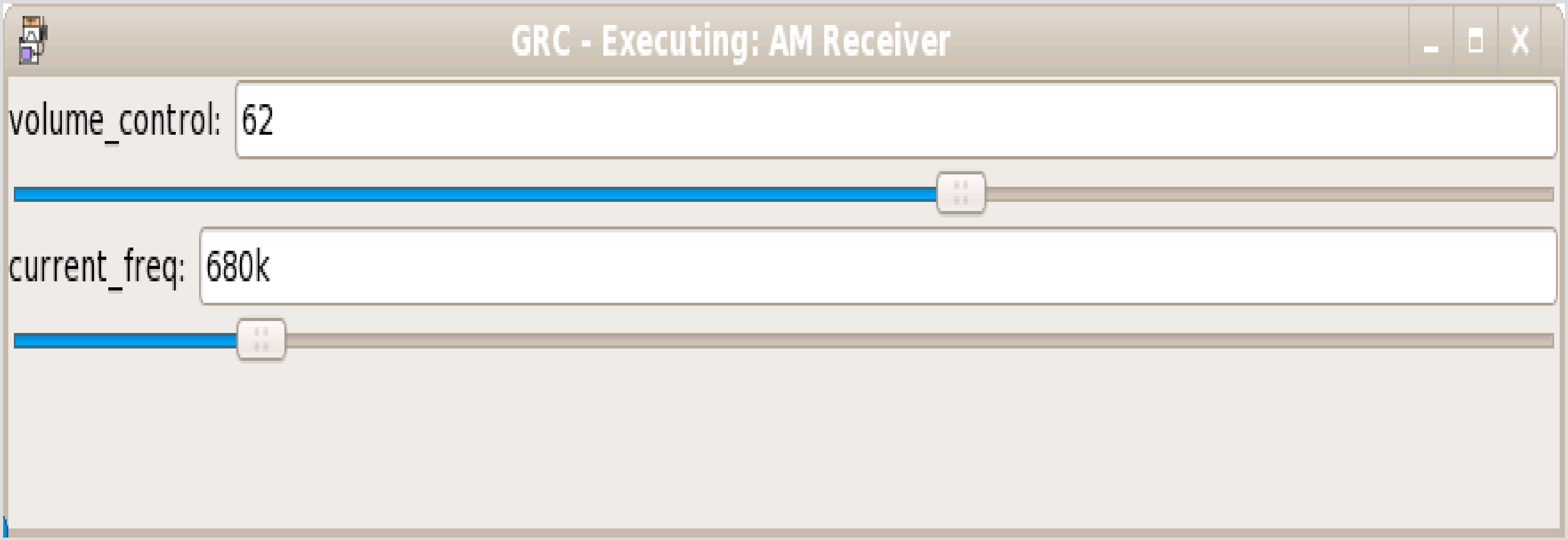
SDR development frameworks

- 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!!

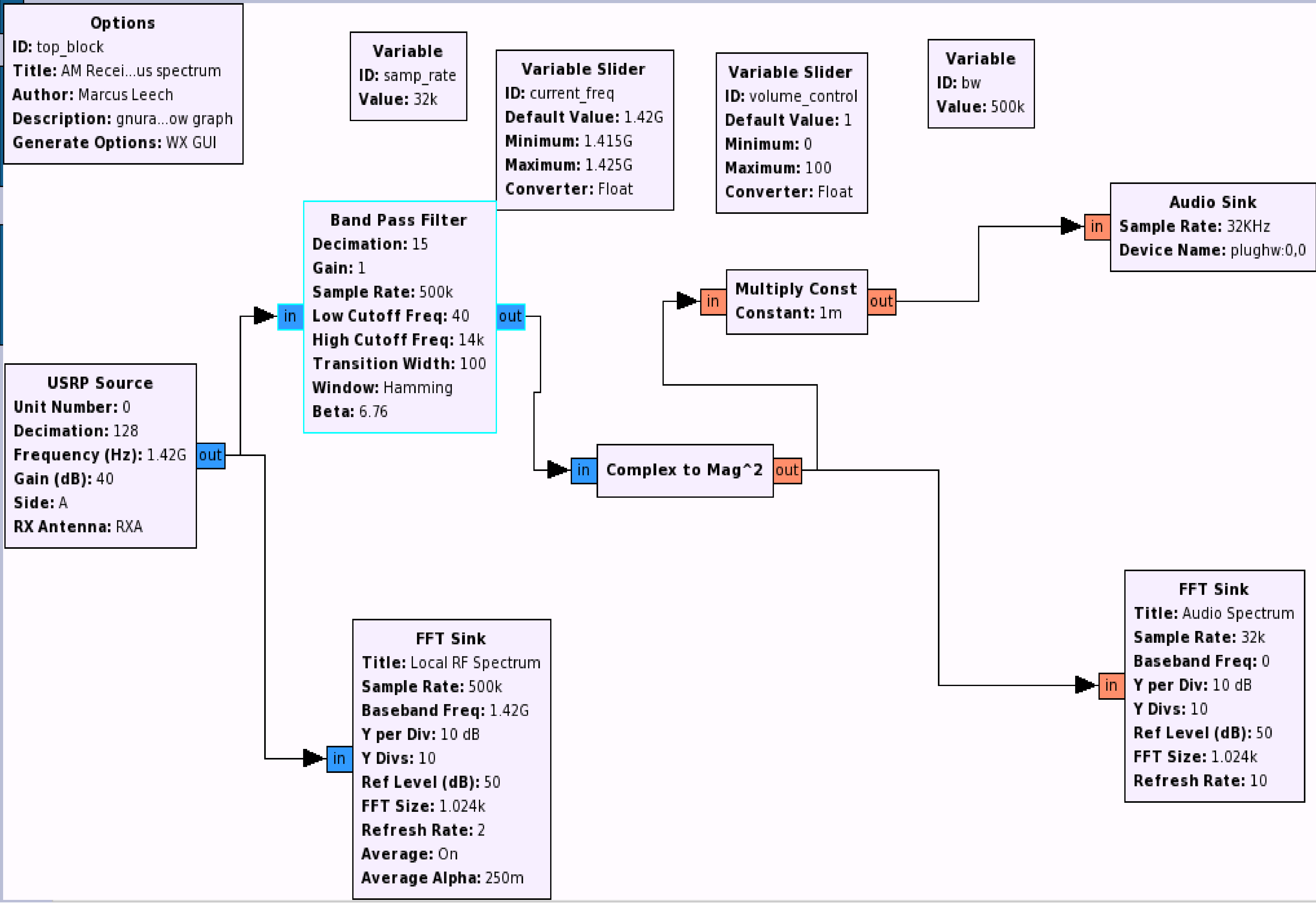
What's after the front-end?



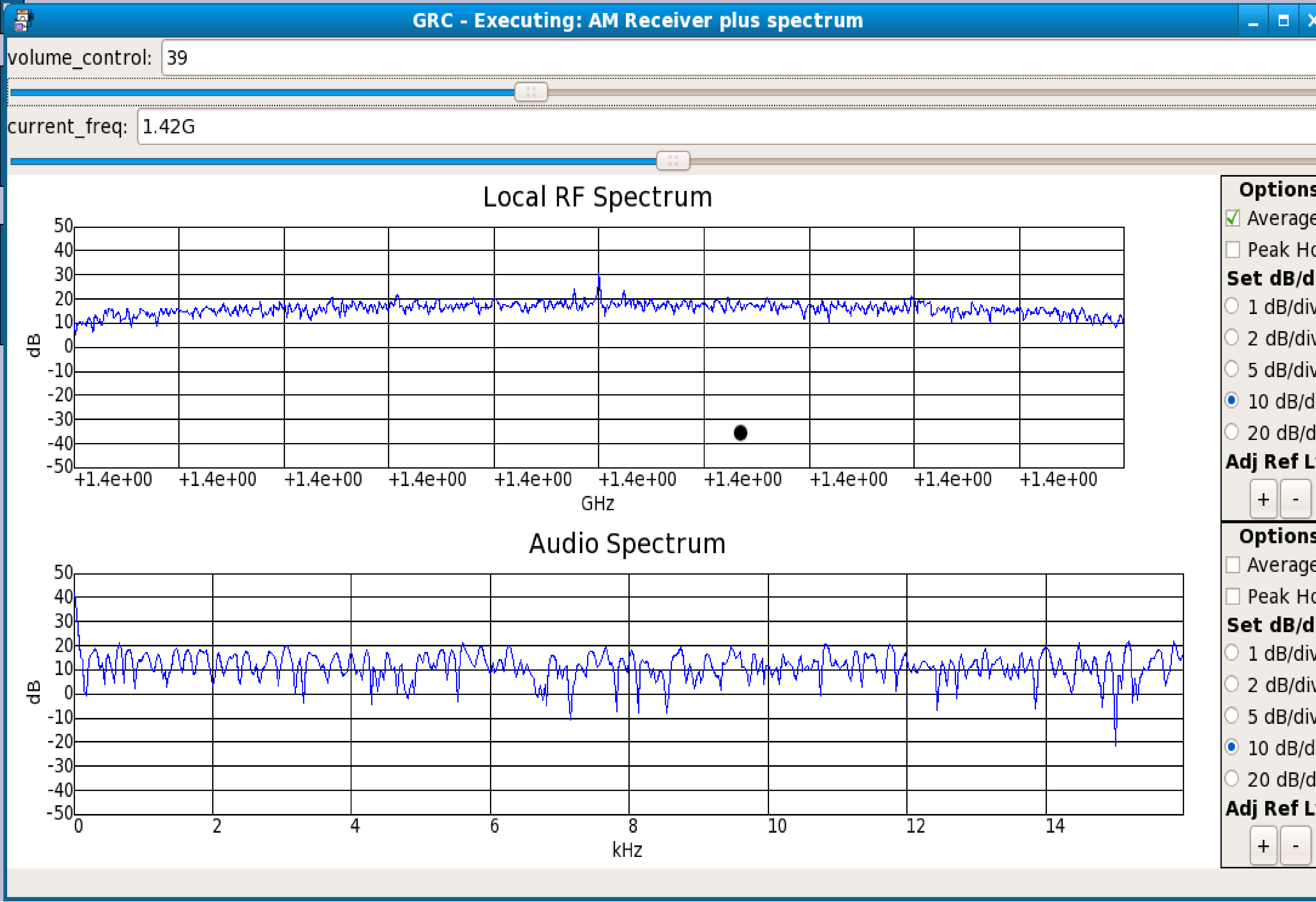
Running AM Receiver



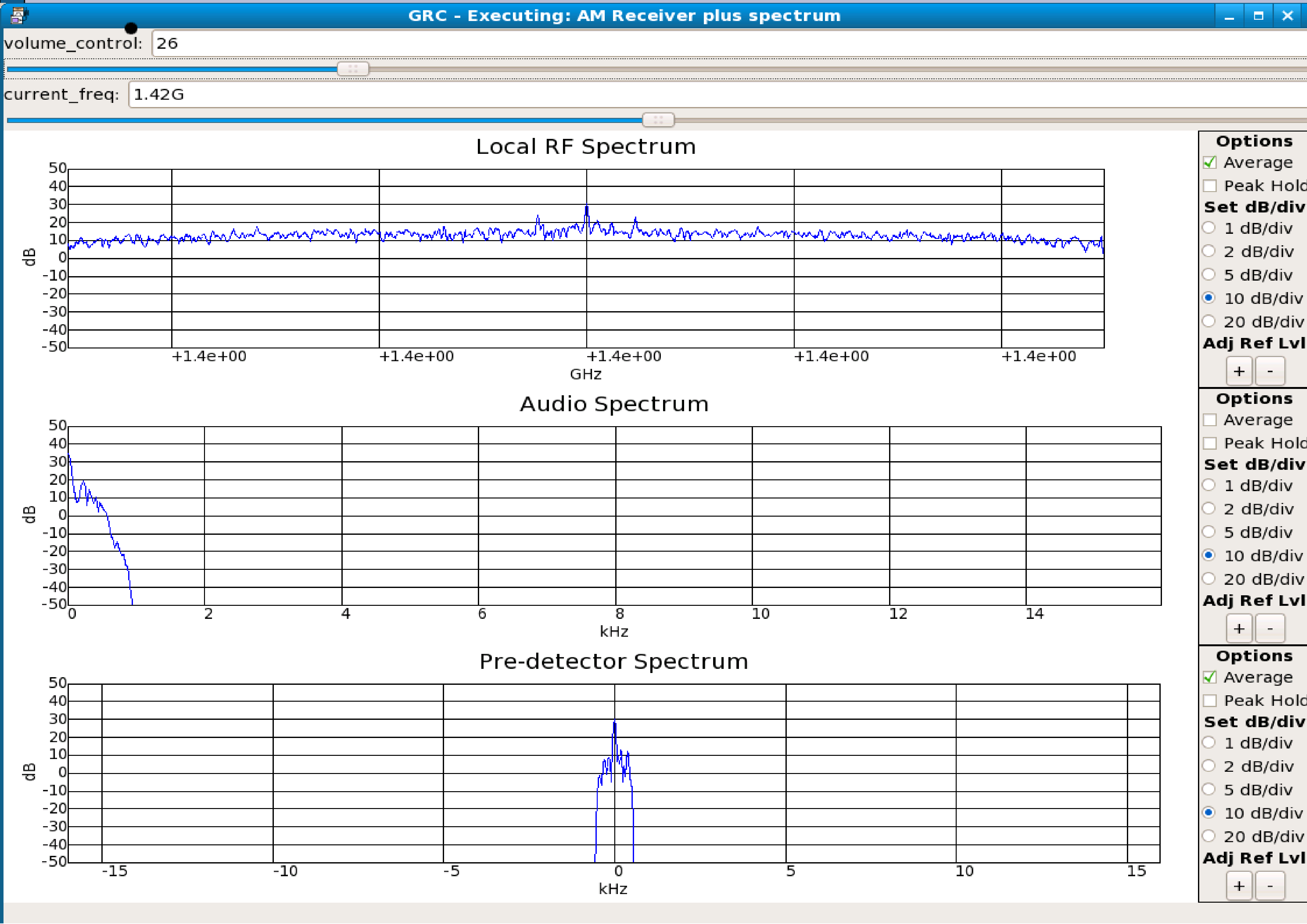
Adding FFT Features



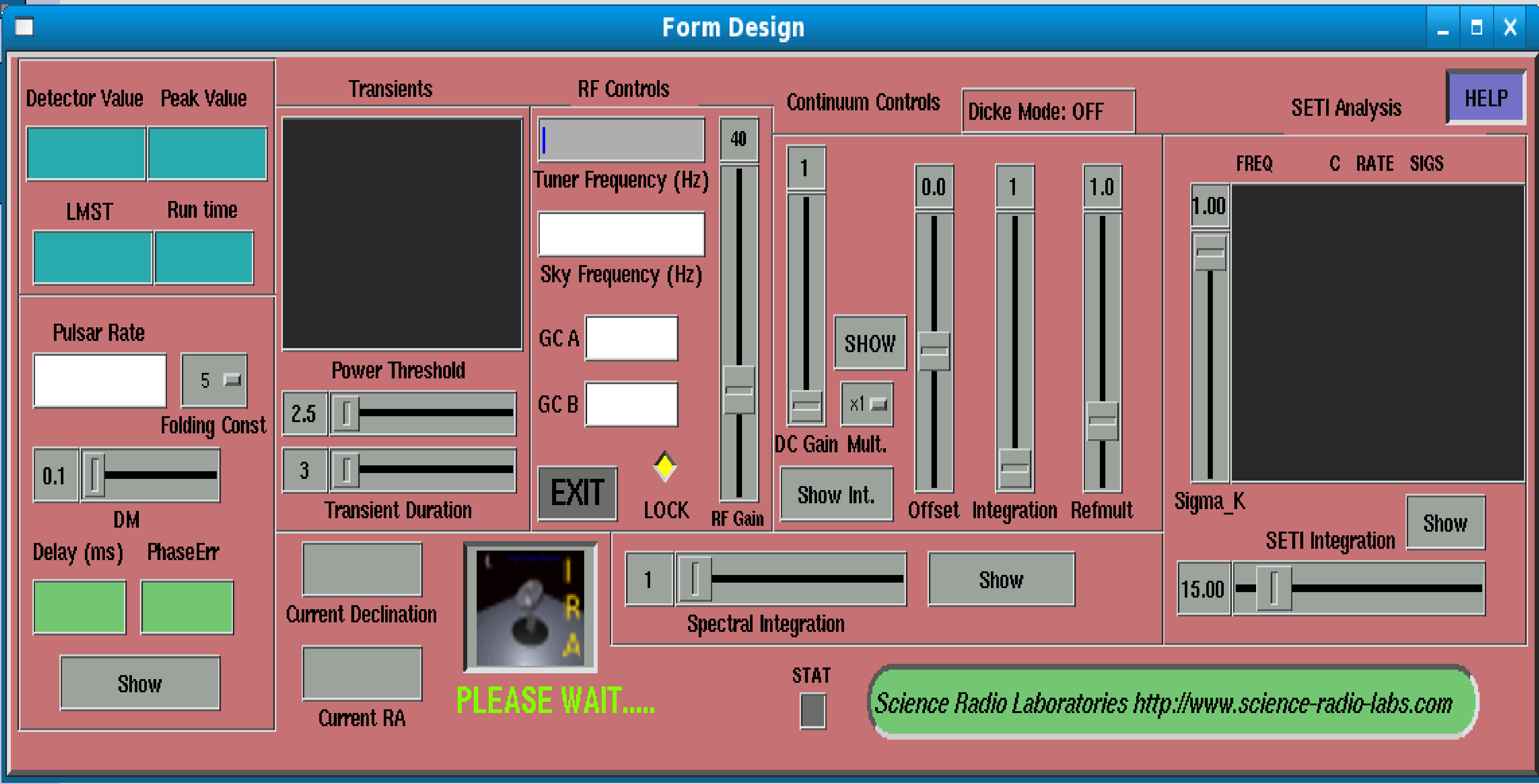
Running New Receiver



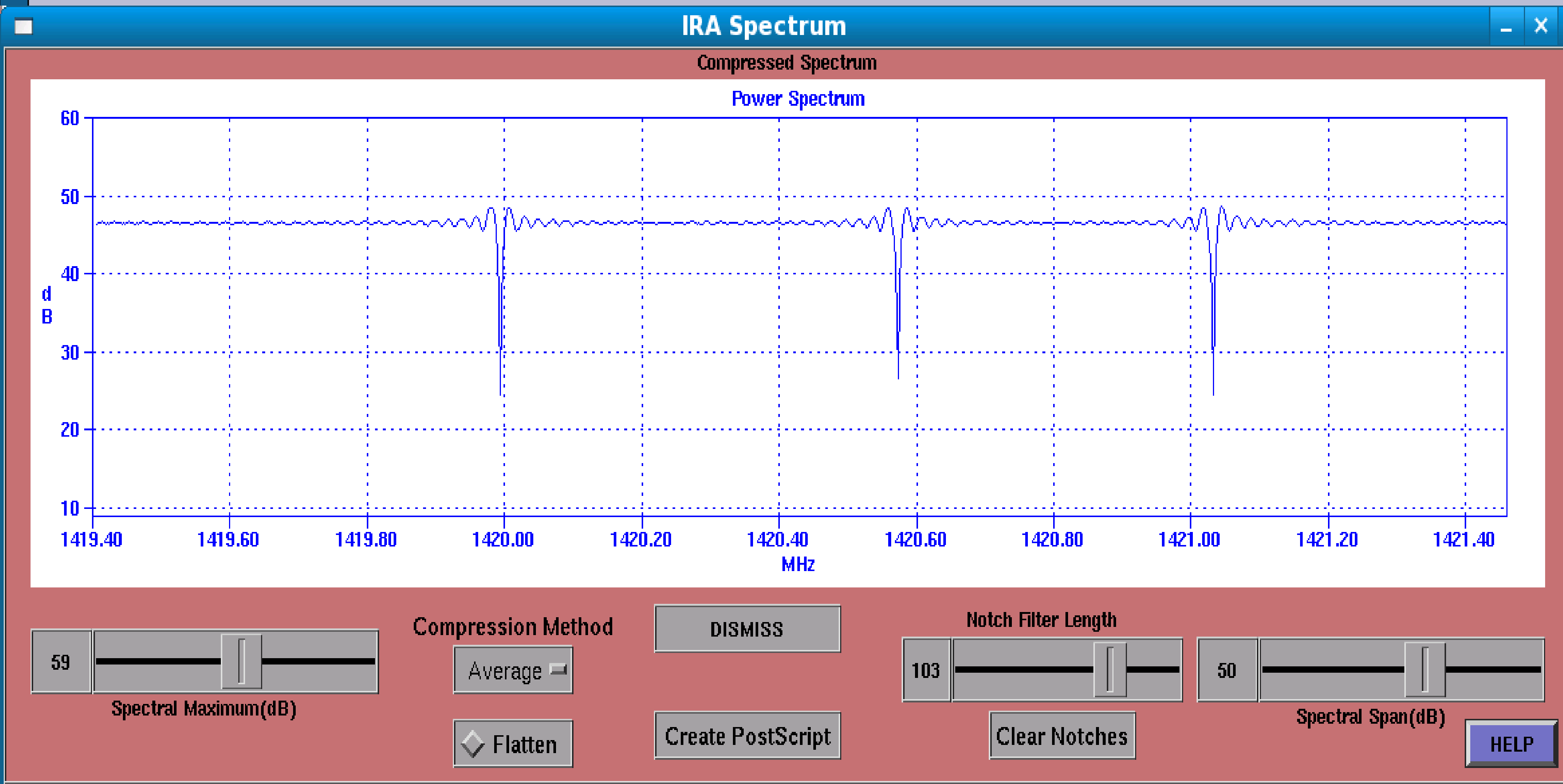
Narrow Filter



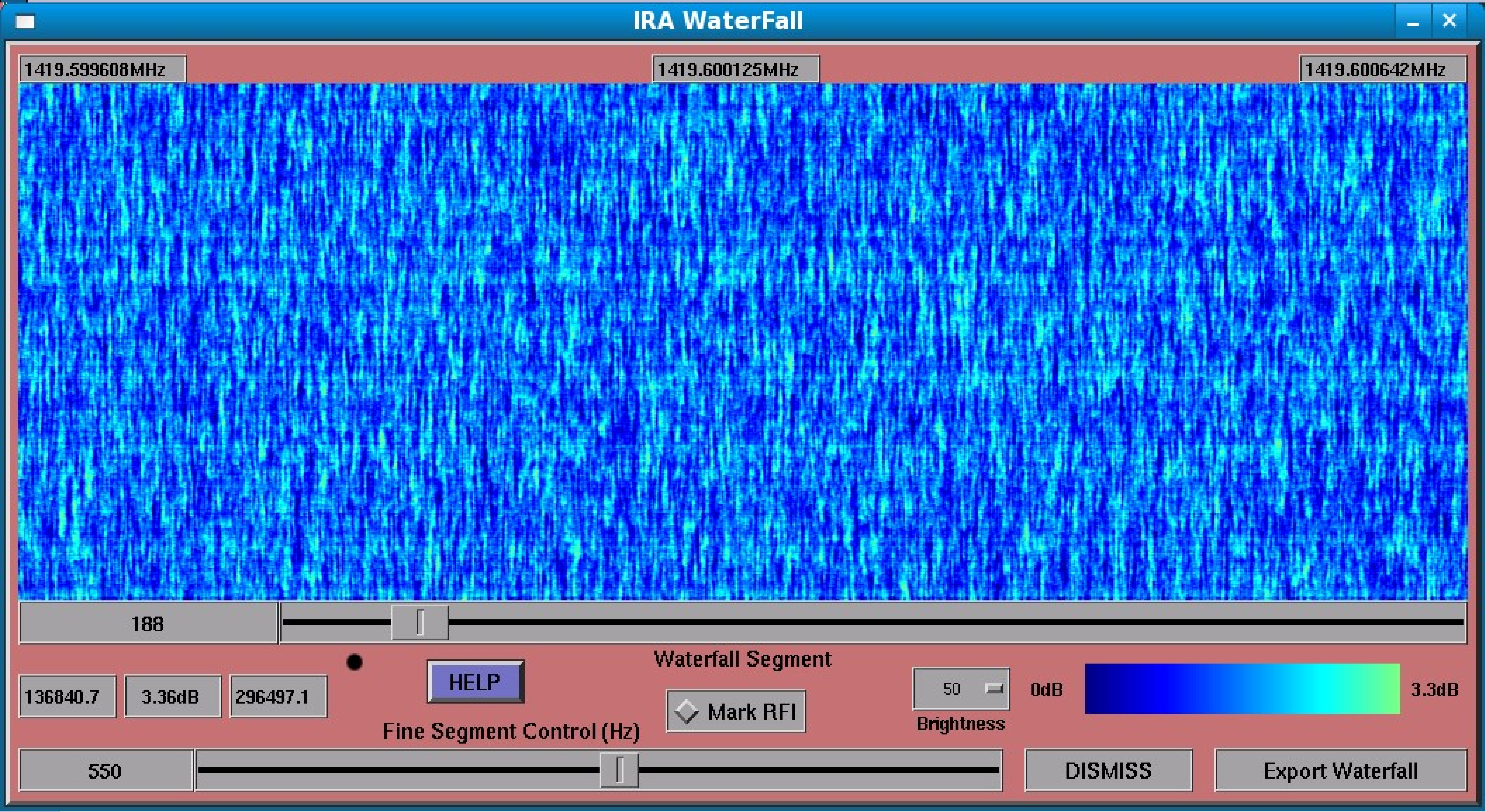
Flexibility: Different GUI



Sophistication limitless



...Really



IRA WaterFall
1419.599608MHz
1419.600125MHz
1419.600642MHz
188
136840.7
3.36dB
296497.1
HELP
Waterfall Segment
Mark RFI
50
Brightness
0dB
3.3dB
Fine Segment Control (Hz)
550
DISMISS
Export Waterfall

Resources

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