Welcome!

Tom Rondeau
&
Johnathan Corgan
The GNU Radio New Users Day

- GNU Radio is growing as a project
- More 2\textsuperscript{nd} party project available
- New areas of use
- New models of use
What is this day for?

- Big project and big scope
- Lots within the runtime framework
- And the blocks both in-tree and out-of-tree
What is this day for?

Introduce opportunities and uses of GNU Radio

FOSDEM 15 .org
Brussels 31 Jan & 1 Feb

WHAT?
A FUTURE TECHNOLOGY FORUM
SEPTEMBER 9TH-11TH, 2015 • ST. LOUIS

DEF CON
Today's Outline

- Perspectives on SDR
- Concepts in building SDRs
- Introduction to:
  - New series on DSP and Analog concepts
  - Today: sampling theory and antennas
- Intros to other aspects of the project
  - Ways to get and build on GNU Radio
  - Using various pre-built projects
  - Hardware for GNU Radio
  - Debugging:
    - Exploring and viewing data
    - What to do when you run into problems?
- Tutorial side event
- Event social
Opening Day Social Event: Birch & Barley

- 1337 14th St NW, Washington, DC 20005
  - 7pm to 11pm
  - Light apps
  - (limited) open bar
Review of GNU Radio
GNU Radio/SDR System View

RF Feed System

- Antenna
- RX Filter/LNA
- TX PA
- Duplexer

Passband Analog RF

SDR Hardware

- AD/DA conversion
- Up/down frequency conversion
- Filtering/resampling

Baseband IQ Samples

GNU Radio Application

EVERYTHING ELSE!

PCI Enet USB...
GNU Radio Block Libraries

- DSP functions for a wide variety of analog and digital communications waveforms
- Accessible from GRC, Python, or C++ application
- Use processor SIMD extensions via the VOLK acceleration library
- `gr_modtool` generates framework for writing your own blocks
GNU Radio Ecosystem

- Commercial Developers
- InfoSec Researchers
- Hardware Vendors
- Professional Services Providers
- Students
- Hobbyists
- Academic Researchers
- Military Researchers
- Commercial Developers
- InfoSec Researchers
- Hardware Vendors
- Professional Services Providers
- Students
- Hobbyists
- Academic Researchers
- Military Researchers
DSP Block – C++ Work Function

```c
int gr_add_ff::work(int noutput_items,
                      gr_vector_const_void_star &input_items,
                      gr_vector_void_star &output_items)
{
    float *out = (float *) output_items[0];
    int noi = d_vlen*noutput_items;
    memcpy(out, input_items[0], noi*sizeof(float));
    volk_32f_x2_add_32f_a(out, out, (const float*)input_items[i], noi);
    return noutput_items;
}
```

GNU Radio Companion (optional)

- Blocks
  - Large library of existing IP -> Mod/demod, filters, USRP I/O, GUI features, etc.
  - Write custom blocks – C++ or Python
- GNU Radio Companion (optional)
  - Import blocks
  - Connect blocks
  - Generate python source code for flowgraph
- Python Flow-Graph
  - Generate from GRC and/or hand-write
  - Simplifies block connectivity

Python Flow-Graph

```python
import gr

tb = gr.top_block()
src1 = gr.sig_source_f(32000, gr.GR_SIN_WAVE, 350, .5, 0)
src2 = gr.sig_source_f(32000, gr.GR_SIN_WAVE, 440, .5, 0)
adder = gr.add_ff()
sink = audio.sink(32000)
tb.connect(src1, (adder, 0))
tb.connect(src2, (adder, 1))
tb.connect(adder, sink)
tb.run()
```