Overview

- **How to install**
  - Fedora
  - Ubuntu

- **Understanding & Using GNU Radio**
  - Which signal blocks are provided by GNU Radio?
  - How to use python to create flow graphs and connect signal blocks?
  - How to write a new signal processing block in C++?

- **An example**: FM Radio receiver
Installation Guide for Fedora

1. Install all basic required package for building GNU radio
   - $ yum groupinstall "Engineering and Scientific" "Development Tools"
   - $ yum install fftw-devel cppunit-devel wxPython-devel libusb-devel guile boost-devel alsa-lib-devel numpy

2. Download and build GNU Radio
   - $ svn co http://gnuradio.org/svn/gnuradio/trunk gnuradio
   - $ ./bootstrap  # Do NOT perform this step if you are building from a tarball.
   - $ ./configure
   - $ make
   - $ make check
   - $ sudo make install

3. Install small device C compiler for USRP
   - $ yum install sdcc
   - $ export PATH=/usr/libexec/sdcc:$PATH

4. Export python environment parameter
   - $ export PYTHONPATH=/usr/local/lib/python2.4/site-packages

5. Test an USRP application
   - ./usrp_wfm_rcv.py -f 96.3
Installation Guide for Ubuntu

- Ubuntu already added GNU Radio packages to their repositories
  - System > Administration > Synaptic Package Manager
  - Search “gnuradio”
  - Select all related gnuradio libs and then apply to install

- To install from console by using command line:
  - Check: http://gnuradio.org/trac/wiki/UbuntuInstall
Understanding & using GNU Radio

- **GNU Radio provide a set of signal processing tools for the computer**
  - Hundreds of signal processing blocks
  - Graphical utilities
  - Can tie in with hardware such as the USRP and various ADC/DAC pci cards

- **Using GNU Radio**
  - Which signal blocks are provided by GNU Radio?
  - How to use python to create flow graphs and connect signal blocks?
  - How to write a new signal processing block in C++?
Signal Processing Blocks Provided by GNU Radio

- GNU Radio documentations generated by Doxygen
Python: Create flow graphs and connect signal blocks

- **Learn basic Python syntax**
  - Learn Python in 10 minutes: [http://www.poromenos.org/tutorials/python](http://www.poromenos.org/tutorials/python)

- **Learn how to use python to create flow graph and connect signal blocks**
  - Graph, Blocks & Connecting
    - [http://www.nd.edu/~jnl/sdr/docs/tutorials/6.html](http://www.nd.edu/~jnl/sdr/docs/tutorials/6.html)
  - Learn by examples
    - [http://www.nd.edu/~jnl/sdr/docs/tutorials/5.html](http://www.nd.edu/~jnl/sdr/docs/tutorials/5.html)
    - [http://www.joshknows.com/?key=gnuradio#example](http://www.joshknows.com/?key=gnuradio#example)
Graph, Blocks & Connecting

- **C++**
  - Performance-critical modules
- **Python**
  - *Glue* to connect modules
  - Non performance-critical modules
- Tutorial: [http://www.nd.edu/~jnl/sdr/docs/tutorials/6.html](http://www.nd.edu/~jnl/sdr/docs/tutorials/6.html)

© Chen Zhifeng, Electrical and Computer Engineering University of Florida
Hello world example

- Generates two sine waves and outputs them to the sound card

```python
#!/usr/bin/env python

from gnuradio import gr
from gnuradio import audio

def build_graph():
    sampling_freq = 48000
    ampl = 0.1

    fg = gr.flow_graph()
    src0 = gr.sig_source_f(sampling_freq, gr.GR_SIN_WAVE, 350, ampl)
    src1 = gr.sig_source_f(sampling_freq, gr.GR_SIN_WAVE, 440, ampl)
    dst = audio.sink(sampling_freq)
    fg.connect((src0, 0), (dst, 0))
    fg.connect((src1, 0), (dst, 1))

    return fg

if __name__ == '__main__':
    fg = build_graph()
    fg.start()
    raw_input('Press Enter to quit: ')
    fg.stop()
```

- Importing necessary module
- Generates two sine waves
- Writes sampling_freq input to the sound card
- Connect the blocks together
More complicated examples (demo)

- FM Receiver:
  - [http://www.nd.edu/~jnl/sdr/docs/tutorials/5.html](http://www.nd.edu/~jnl/sdr/docs/tutorials/5.html)
How to Write a Signal Processing Block

- Implement a class derived from gr_block in C++
- Use SWIG (Simplified Wrapper and Interface Generator) to generate the interface between Python and C++
- Provide a python module in gnuradio package, allowing us to access the new block in a simply way

References
- [http://www.nd.edu/~jnl/sdr/docs/tutorials/10.html](http://www.nd.edu/~jnl/sdr/docs/tutorials/10.html)
- [http://www.nd.edu/~jnl/sdr/docs/tutorials/11.html](http://www.nd.edu/~jnl/sdr/docs/tutorials/11.html)
References

- GNU Radio
  - http://www.gnu.org/software/gnuradio/
- SDR Documents:
  - http://www.nd.edu/~jnl/sdr/docs/
- GNU Radio 3.0svn Documentation