RF Project Intro

SAGI Summer School 2024 (3S24)
Session RA-03b

Bang D. Nhan, Ph.D

(Nhan Định Bang)

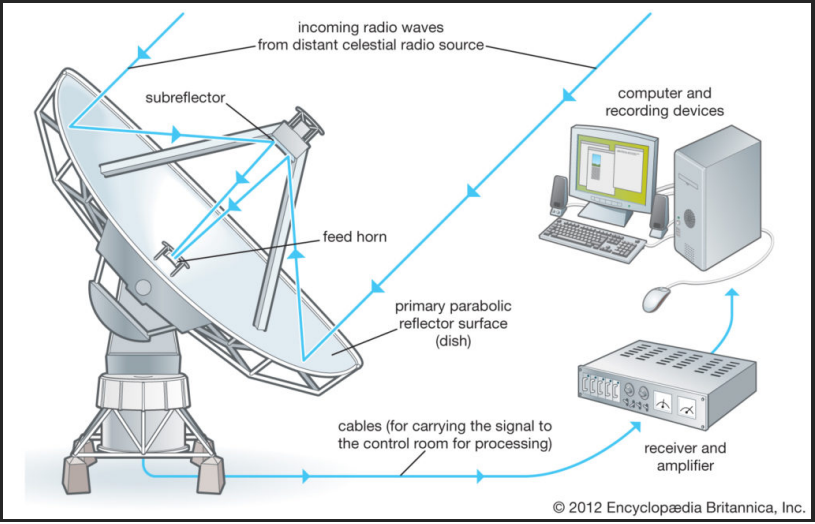
National Radio Astronomy Observatory (NRAO)
Charlottesville, Virginia, USA
Email: bnhan@nrao.edu

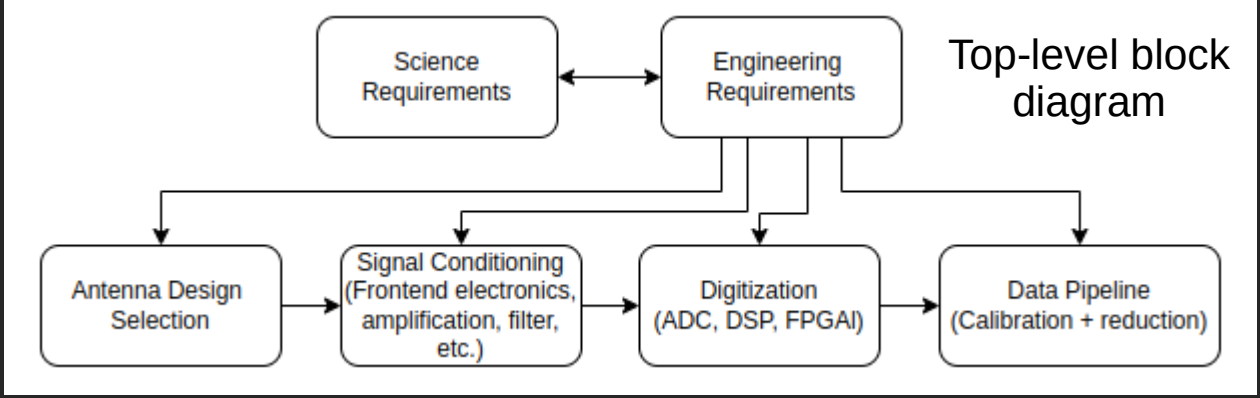
Outline

I. RF DAQ
   i. Analog-to-Digital Converter (ADC)
      - Quantum efficiency
   ii. Field Programmable Gate Array (FPGA)
II. Digital Signal Processing (DSP)
   i. Oversampling vs undersampling
   ii. Fast Fourier Transform (FFT) / FFTW3
   iii. Polyphase Filter Bank
III. Software Defined Radio (SDR)
IV. High-Performance Computing (HPC)

Radio telescope instrumentation

Sensors → Measure Voltage → Digitize & Spectrum (FFT) → Data analysis

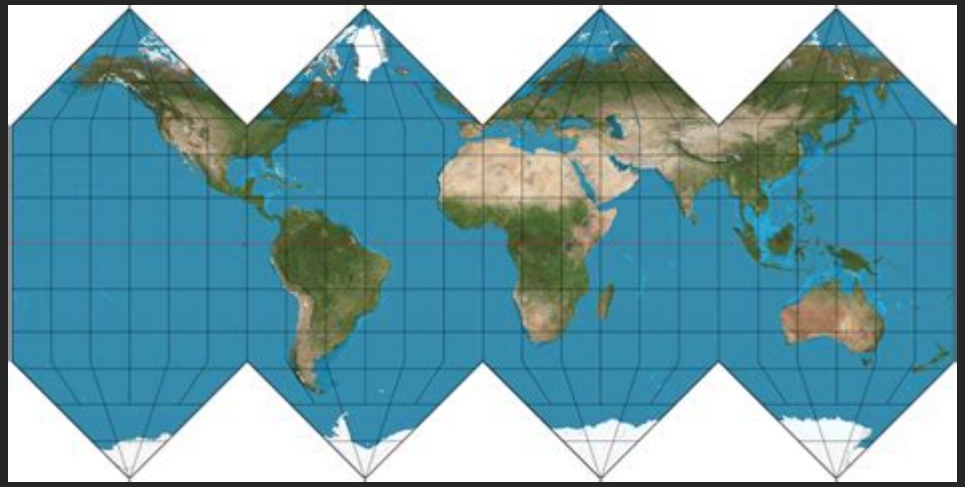


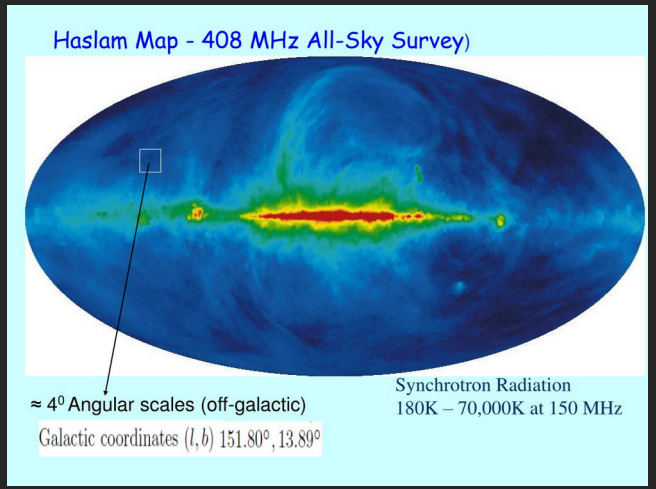


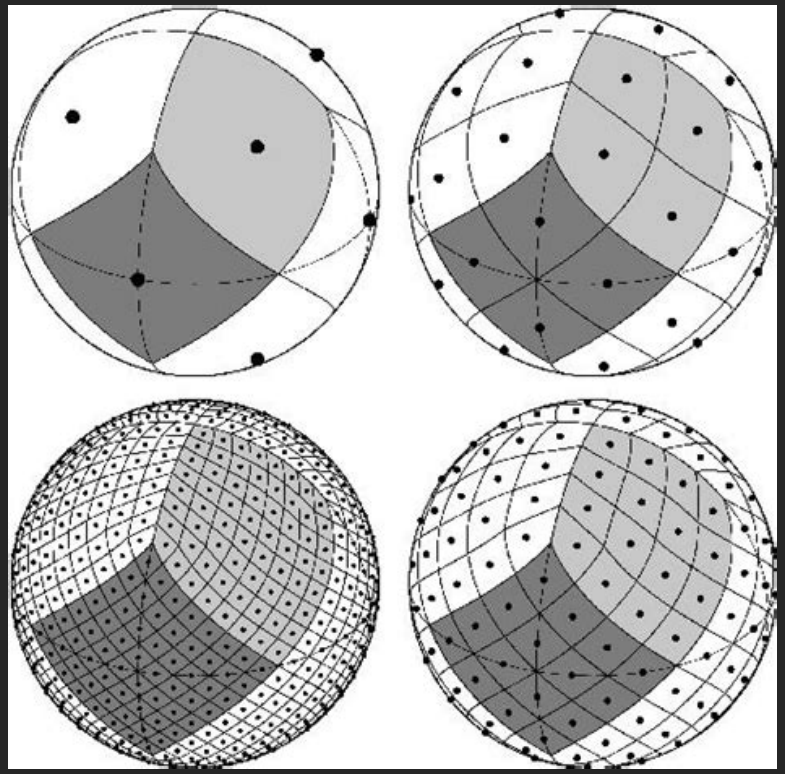
Top-level block diagram

Lab Part 1: Observation simulation (Python simulation with Astropy & Healpy)

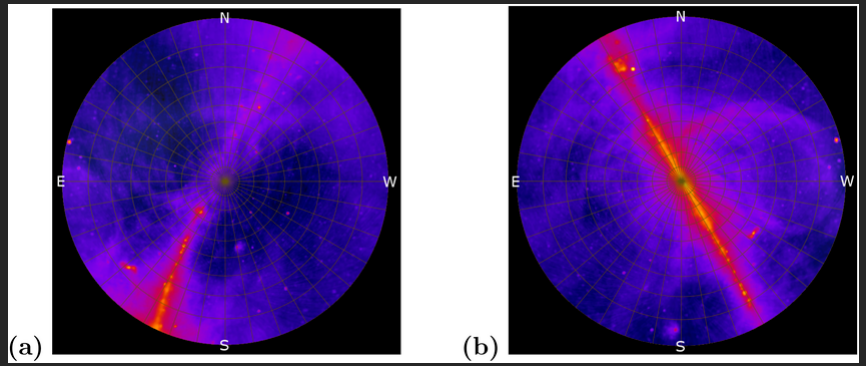
Healpix sky survey

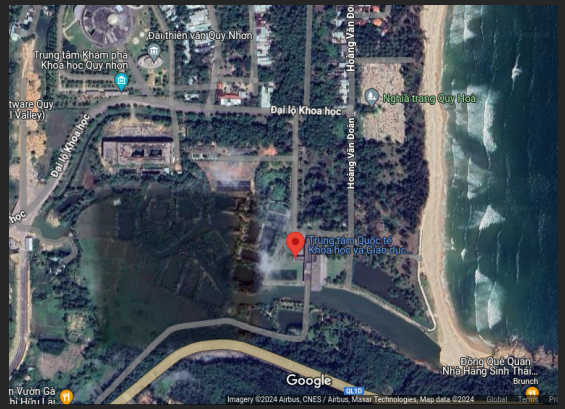


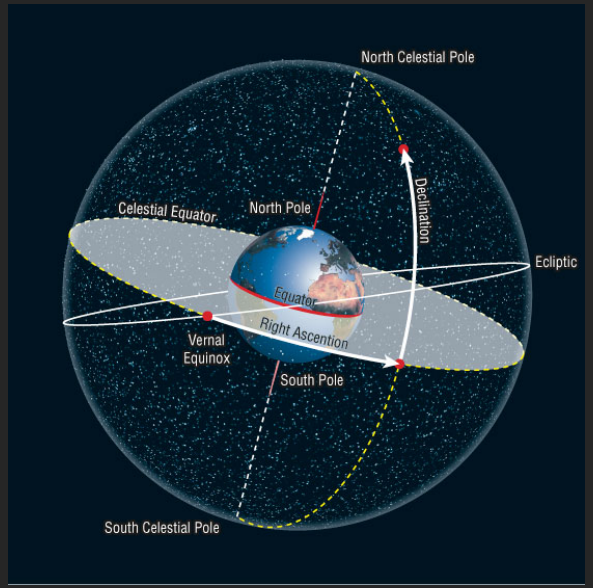




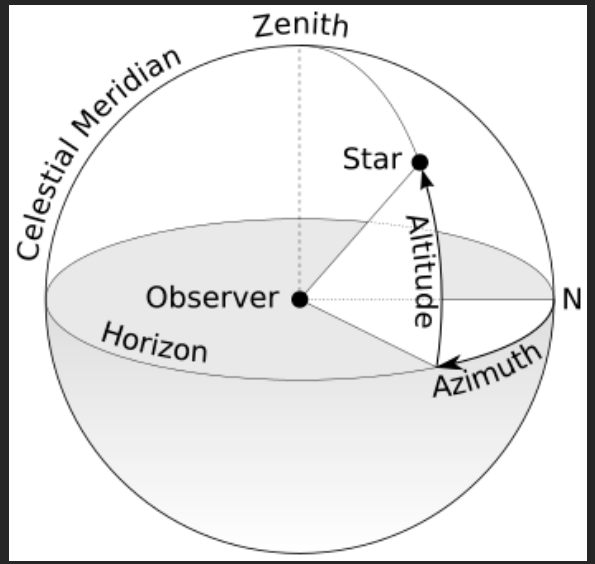
How the Galaxy look like for a given location Lon & Lat (Coordinate conversion with Astropy)





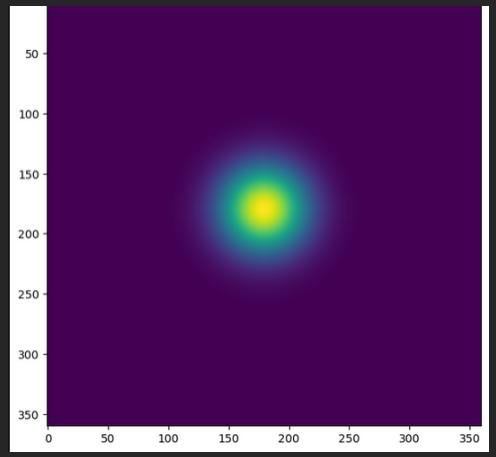


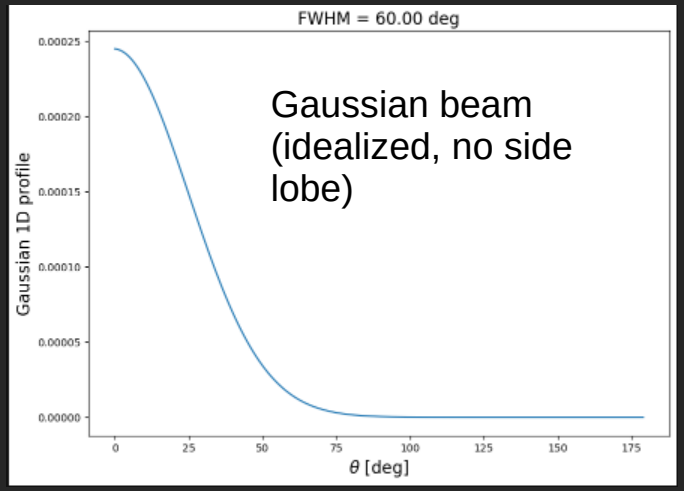
Equatorial coordinate



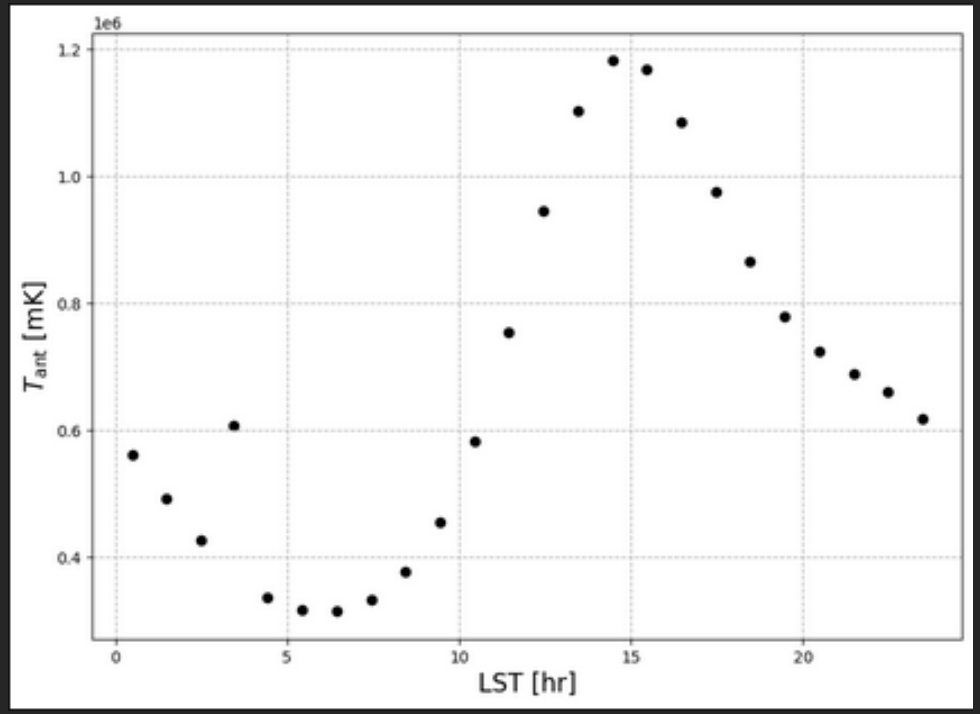
Horizontal coordinate

Calculated antenna beam weighted sky signal





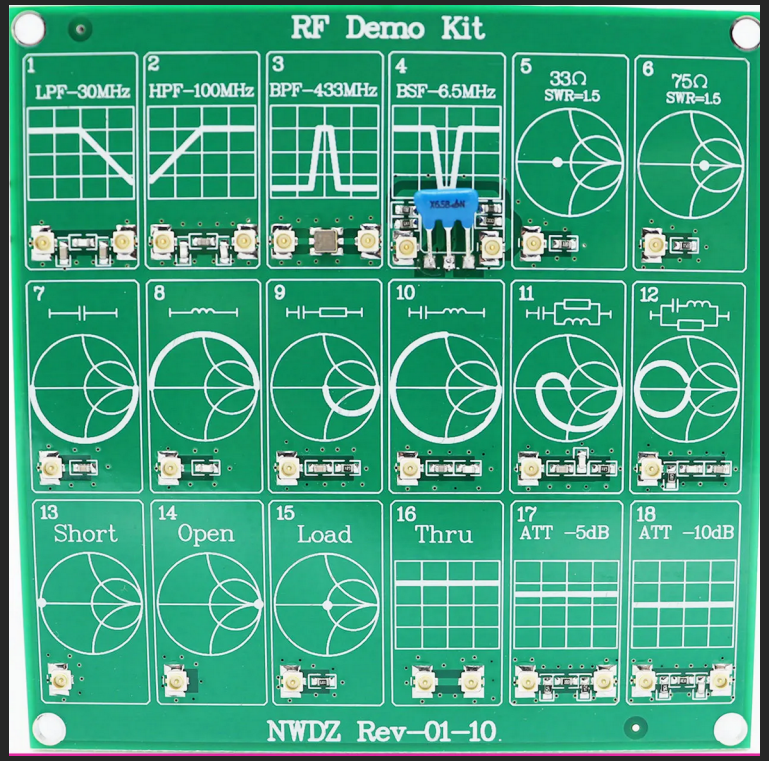
Gaussian beam (idealized, no side lobe)

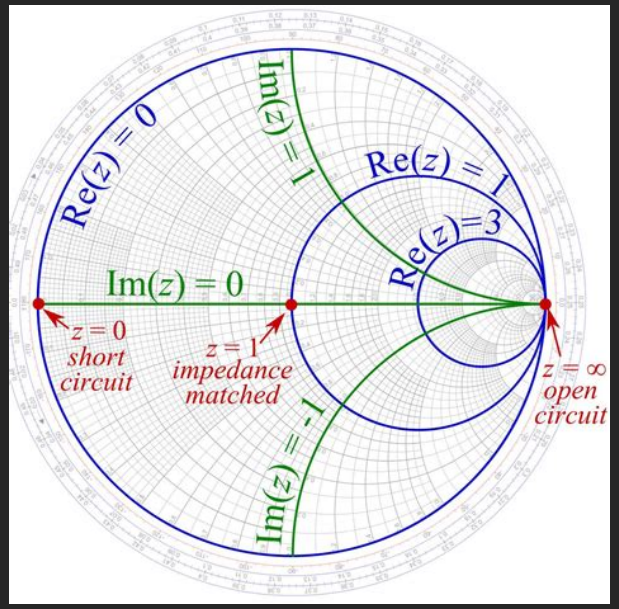


$$T_{\rm ant}(408{\rm MHz}) = \frac{\int_0^{2\pi}\int_0^{\pi} B(\theta,\phi)T_{\rm Haslam}(\theta,\phi)\sin\theta {\rm d}\theta {\rm d}\phi}{\int_0^{2\pi}\int_0^{\pi} B(\theta,\phi)\sin\theta {\rm d}\theta {\rm d}\phi} = \frac{\sum_i B(i)T_{\rm Haslam}(i)}{\sum_i B(i)}$$

Simulate observed spectrum at a given freq from the Milky Way

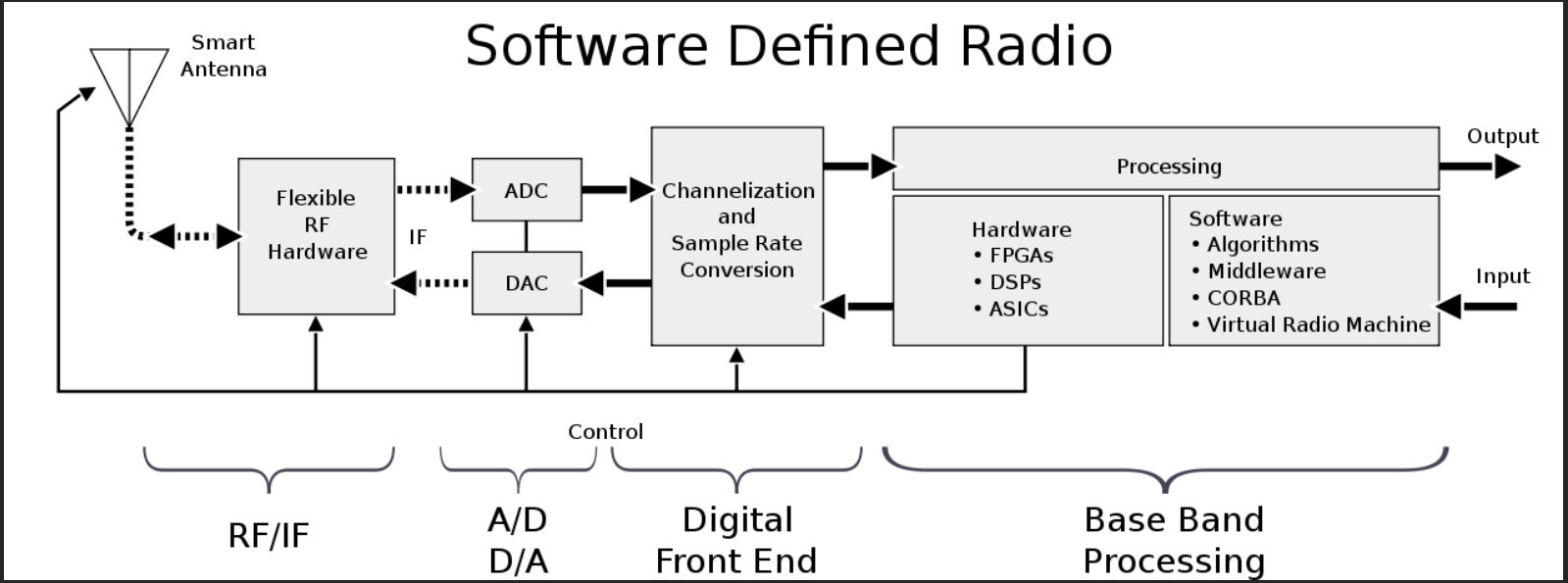
Lab Part 2: Intro with NanoVNA





- Learn basic VNA calibration (open, short, load)
- Measure S11 different RF components & antennas
- Plot with Pycraf

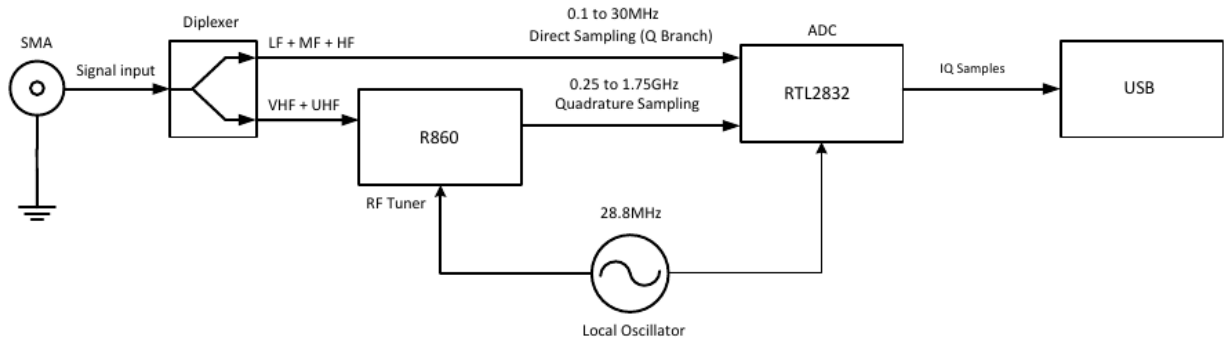
Lab Part 3: SDR & GNU-Radio Intro (DSP & DAQ)



Simply:
RF components traditionally implemented in hardware (**e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.**) are instead implemented via **software** on a PC or embedded systems (e.g., FPGA, SoC)

https://en.wikipedia.org/wiki/Software-defined_radio

Simplified Schematic



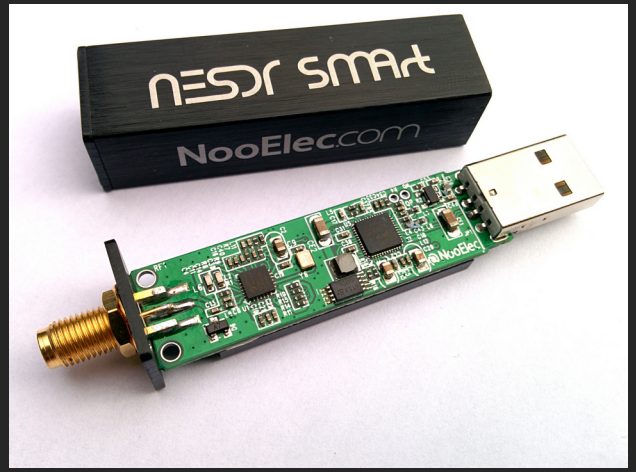
2022-10-04

1 of 3

https://www.nooelec.com/

Revision 1

© 2022 Nooelec Inc. All Rights Reserved.



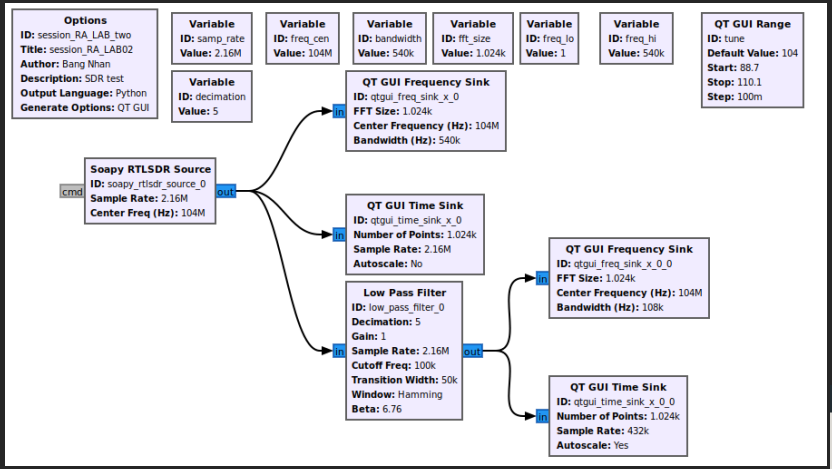
Features

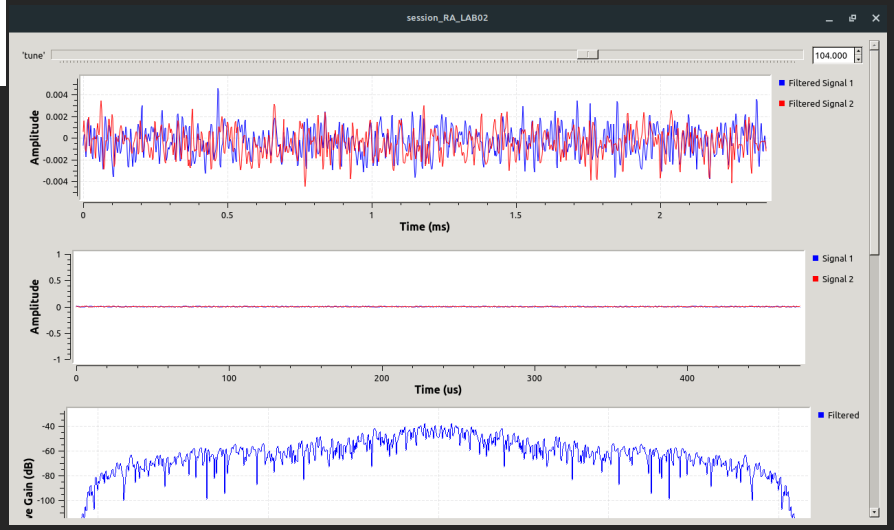
- 0.5PPM TCXO
- 0.1 to 25MHz direct sampling tuning range
- 25 to 1750MHz quadrature sampling tuning range
- 0 - 49.6dB tuner gain
- 7 Bit ADC
- Up to 3.2MSPS sample rate
- 50Ω input SMA connector
- USB type A output
- Lightweight
- FCC, CE and IC certified

Applications

- Software defined radio
- General purpose radio reception
- Low cost radio systems
- Spectrum analysis
- Wideband reception
- Embedded application
- Remote installations
- Lightweight systems

GNU-Radio


Options
ID: session_RA_LAB_two
Title: session_RA_LAB02
Author: Bang Nhan
Description: SDR test
Output Language: Python
Generate Options: QT GUI
Variable
ID: samp_rate
Value: 2.16M
Variable
ID: freq_cen
Value: 104M
Variable
ID: bandwidth
Value: 540k
Variable
ID: fft_size
Value: 1.024k
Variable
ID: freq_lo
Value: 1
Variable
ID: freq_hi
Value: 540k
QT GUI Range
ID: tune
Default Value: 104
Start: 88.7
Stop: 110.1
Step: 100m
Variable
ID: decimation
Value: 5
QT GUI Frequency Sink
ID: qtgui_freq_sink_x_0
FFT Size: 1.024k
Center Frequency (Hz): 104M
Bandwidth (Hz): 540k
Soapy RTLSDR Source
ID: soapy_rtlsdr_source_0
Sample Rate: 2.16M
Center Freq (Hz): 104M
cmd
out
in
QT GUI Time Sink
ID: qtgui_time_sink_x_0
Number of Points: 1.024k
Sample Rate: 2.16M
Autoscale: No
Low Pass Filter
ID: low_pass_filter_0
Decimation: 5
Gain: 1
Sample Rate: 2.16M
Cutoff Freq: 100k
Transition Width: 50k
Window: Hamming
Beta: 6.76
QT GUI Frequency Sink
ID: qtgui_freq_sink_x_0_0
FFT Size: 1.024k
Center Frequency (Hz): 104M
Bandwidth (Hz): 108k
QT GUI Time Sink
ID: qtgui_time_sink_x_0_0
Number of Points: 1.024k
Sample Rate: 432k
Autoscale: Yes


session_RA_LAB02
'tune'
104.000
Filtered Signal 1
Filtered Signal 2
Amplitude
Time (ms)
Signal 1
Signal 2
Amplitude
Time (us)
Filtered
re Gain (dB)

Lab Extra: Connect and acquire some data from the radio antenna at ICISE