Boeing Bluetooth Protocol Analytical Research : Final Demo

By: Caroline Terre, Matthew Irvin, Gianna Scarangelli, Jonah Rowell, Connal Grace

EMBRY-RIDDLE
Aeronautical University

Project Proposal

- Obtain IoT/Bluetooth devices
- Evaluate security of common IoT devices and protocols such as Bluetooth, Zigbee
- Produce Final White Paper presentation of findings

Brief Overview

- Main goal: discover what possible ways Bluetooth and other wireless communication may be interrupted or exploited within the realm of aviation
- This semester we have:
  - Researched functioning of Bluetooth and Zigbee protocols
  - Obtained majority of necessary tools and equipment
  - Developed test bed design
  - Built the functioning components of test bed

Purchases Overview

Current Possession

- 5 Raspberry Pi CanaKit
- 1 Zigbee Development Board
- 2 Hack RF Ones (software defined radio)
- 1 ESP32 Development Board
- 10 Ethernet cables
- 3 Flipper Zero
- 3 Raspberry Pi Keyboards
- 3 Raspberry Pi Displays
- 3 SSD 1TB

Ordered & Awaiting

- 1 Antenna for HackRF
- 3 Micro SD
- 1 Mini Hdmi to Mini HDMI Cable
- 10 Mini HDMI to USB A Cables

Requirements

- All devices establish proper and secure connection as defined
- All signals and data rates will operate/transmit between given minimum and maximum standards
- Controlled testing environment assuming no interruptions
- Other requirements for security, software interfaces

Final Design

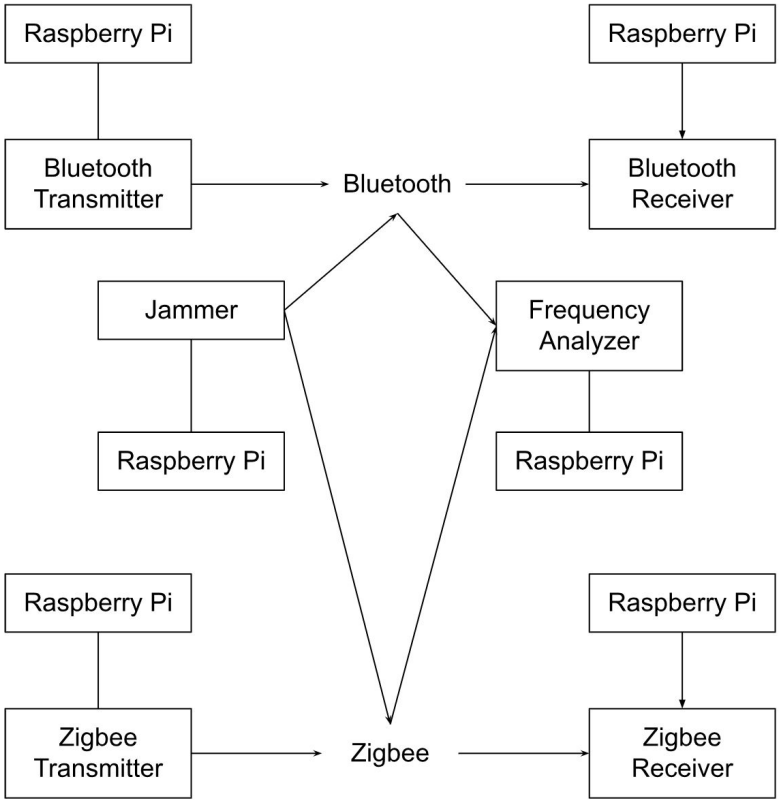
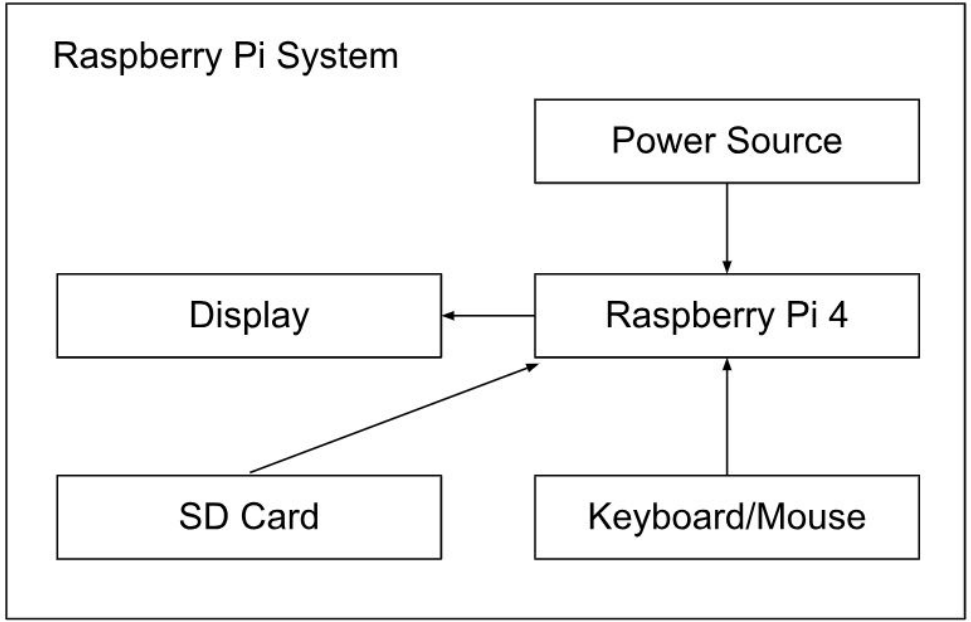
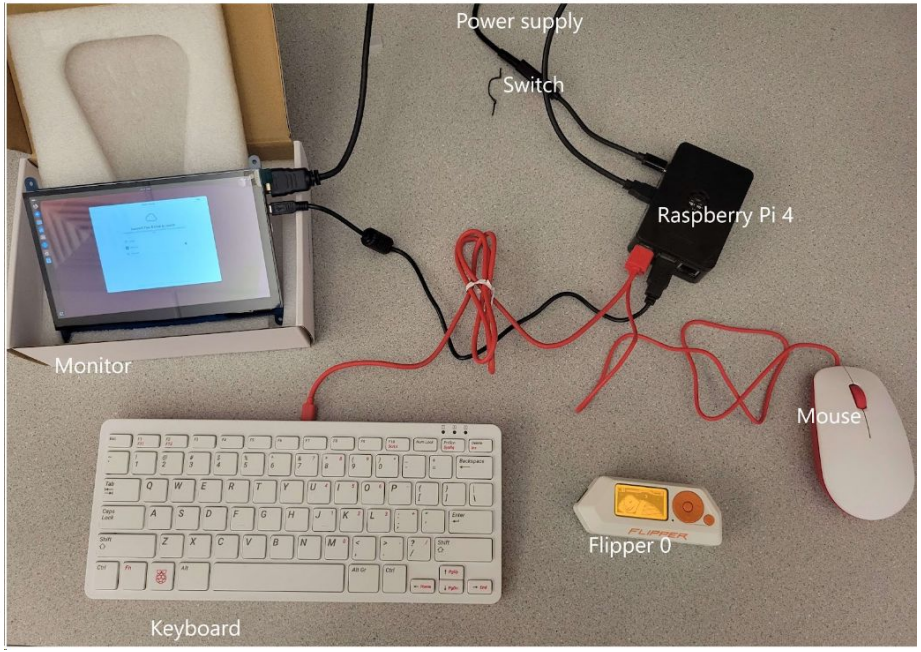


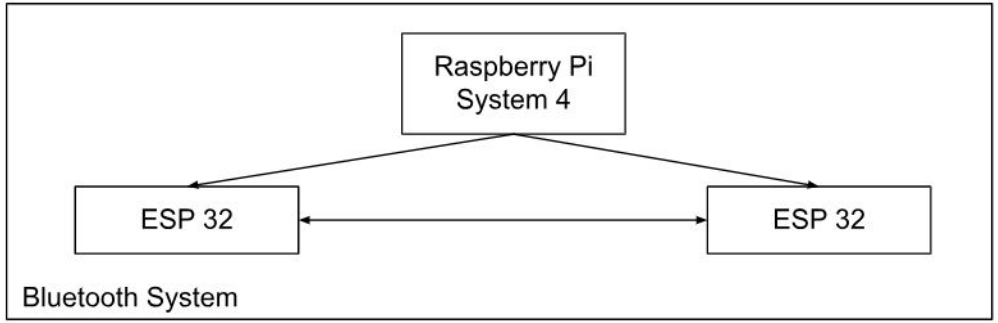
Figure 1: High-Level System Architecture Diagram

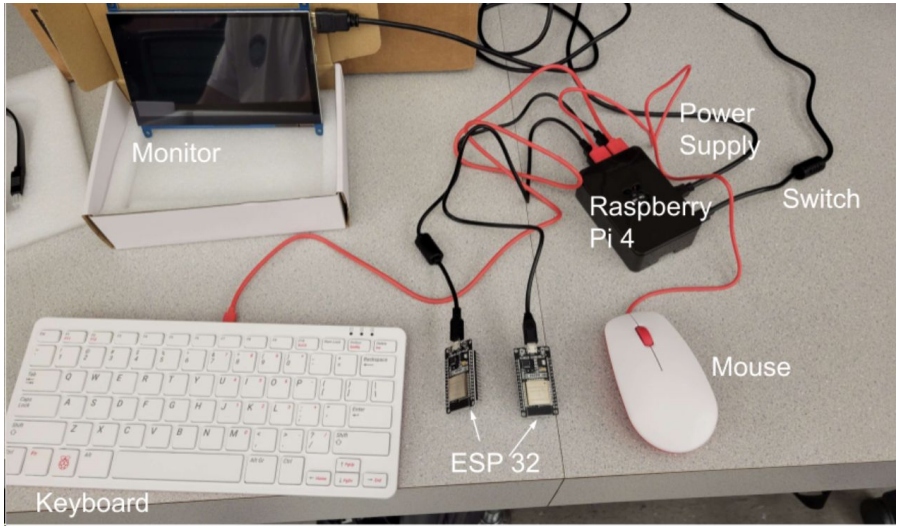
Raspberry Pi System





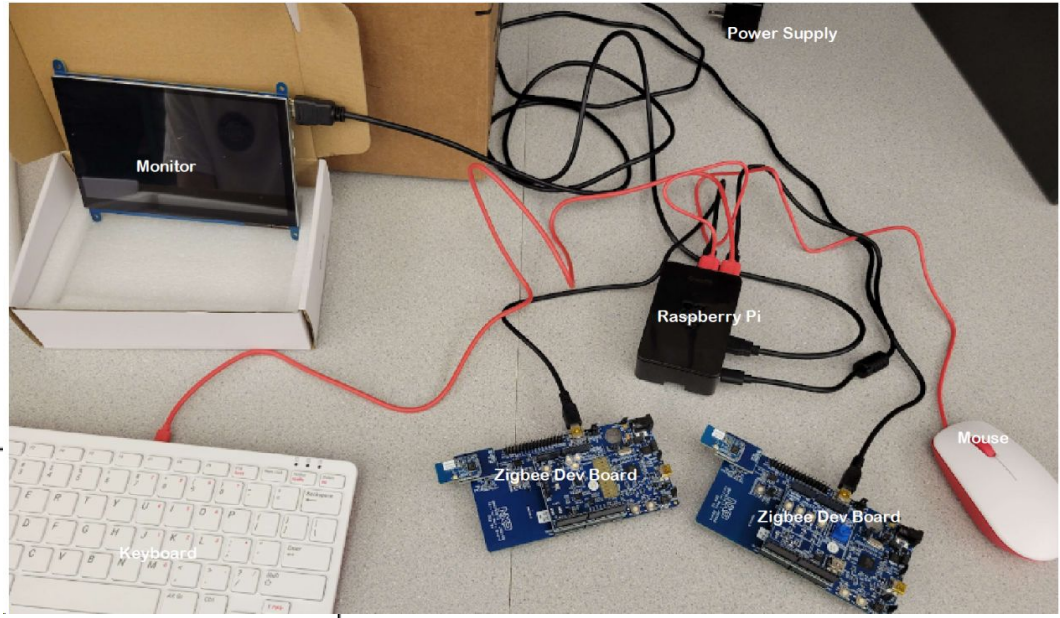
Bluetooth System

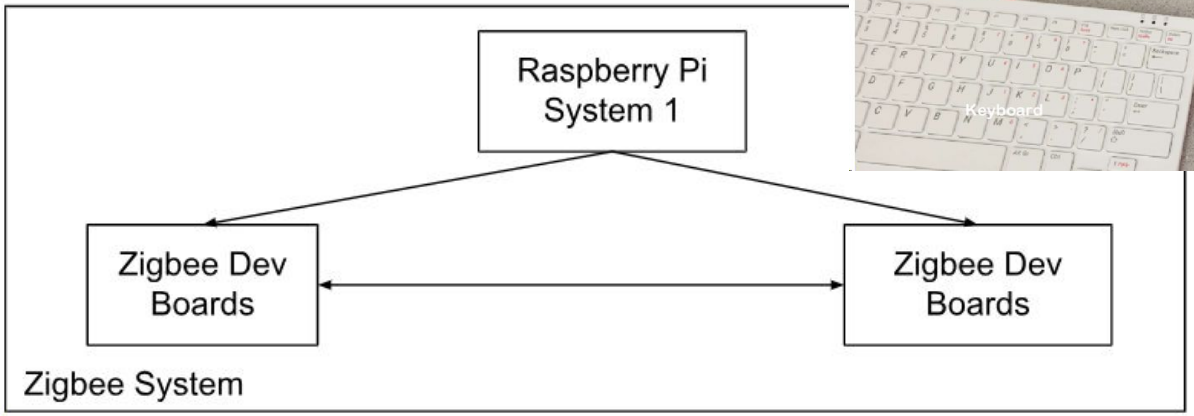




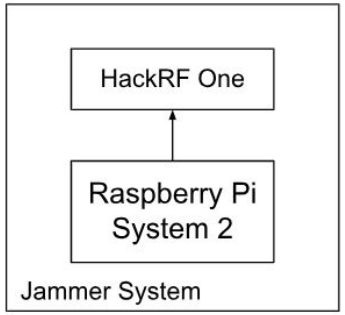
ESP32 Testing

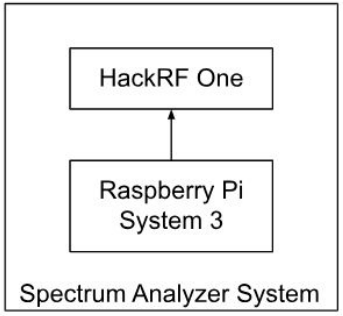
Zigbee System

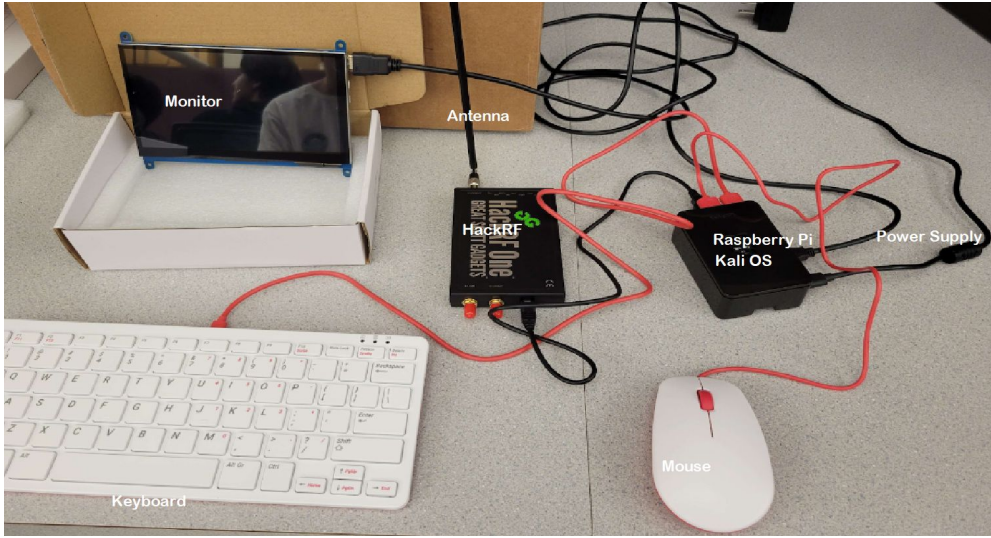




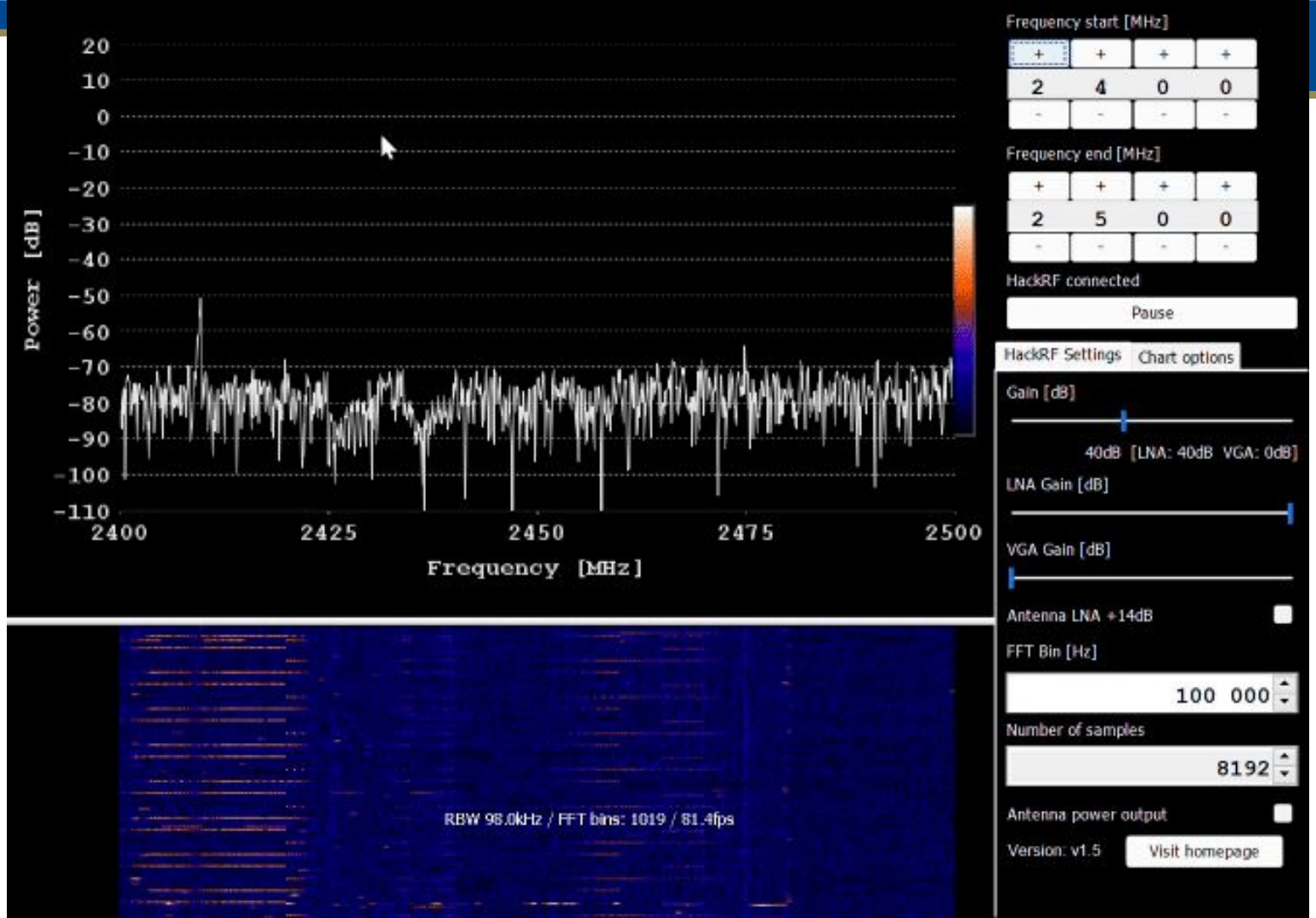
Software-Defined Radio System



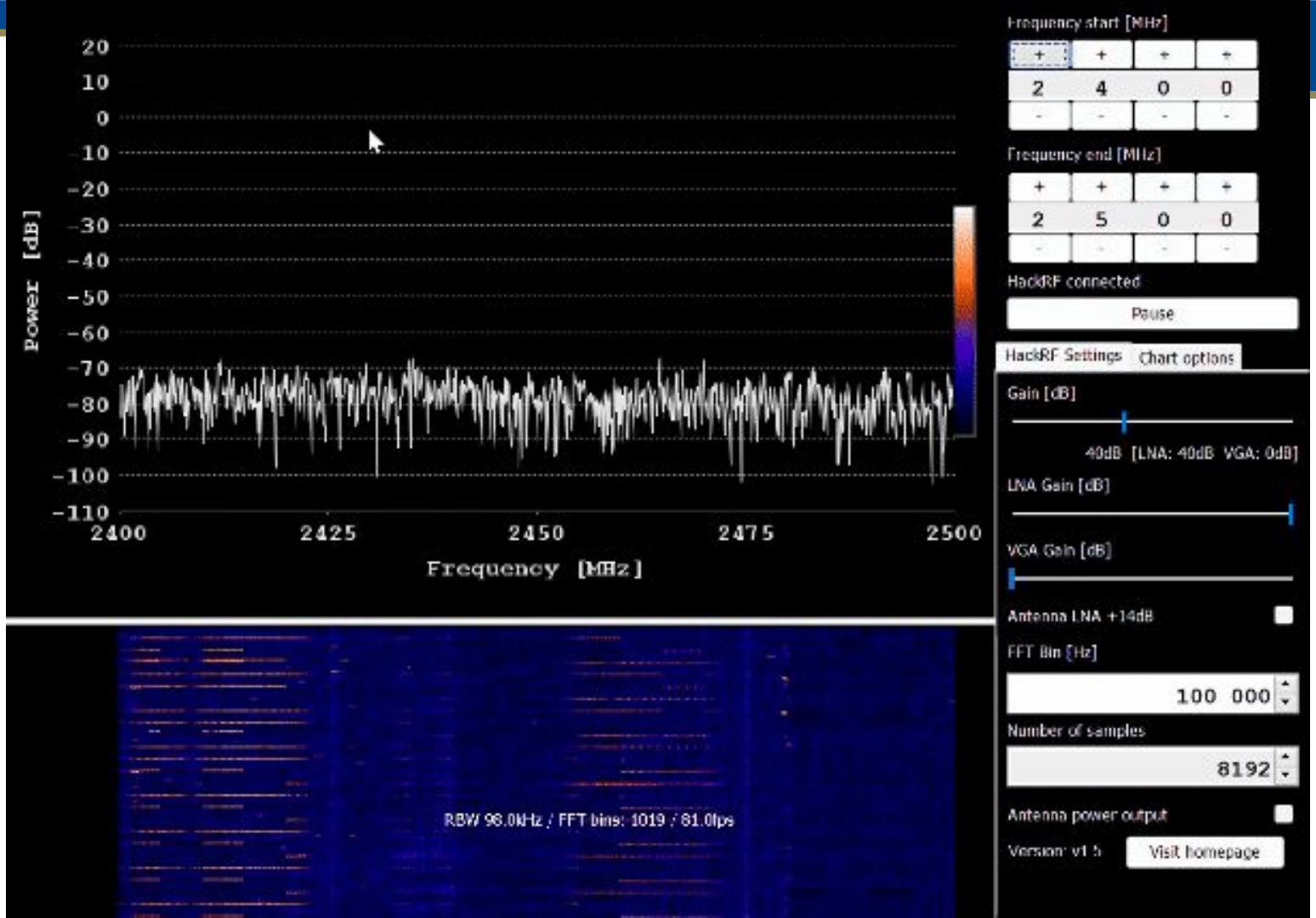




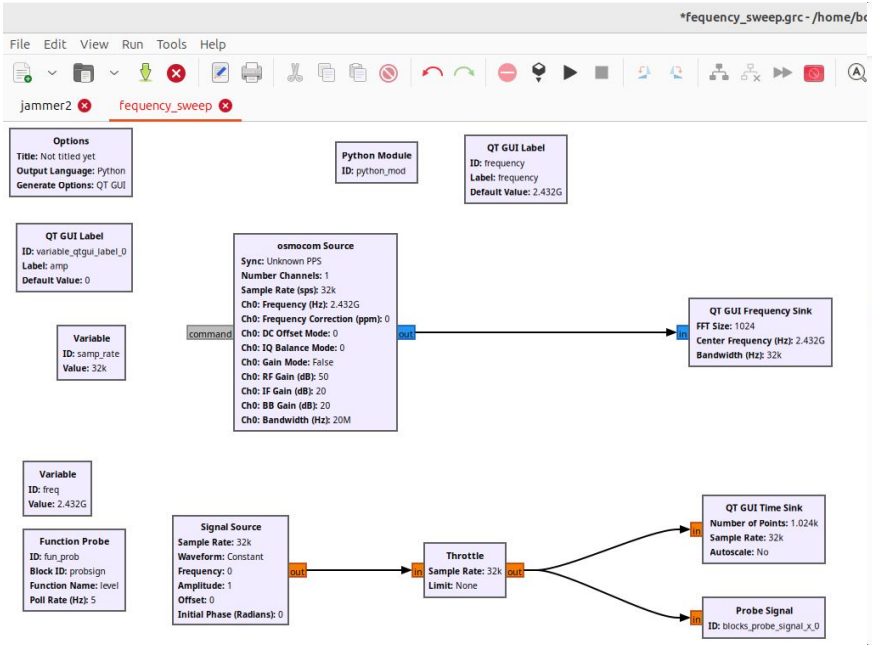
Hack RF Testing

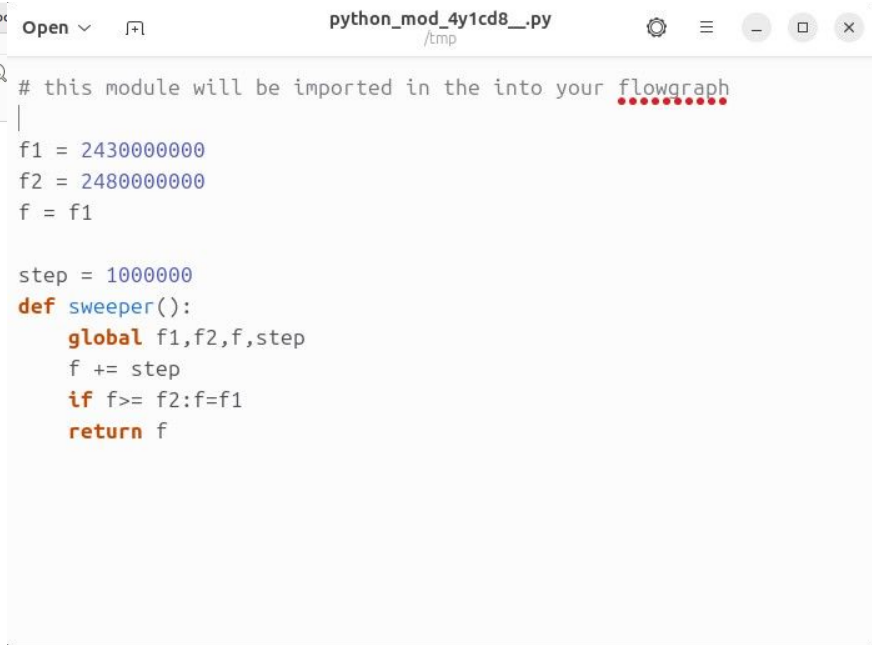


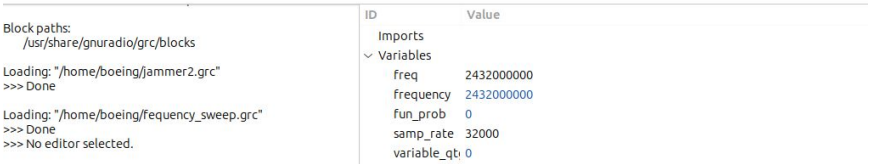
Hack RF Testing



GNU Radio


*fequency_sweep.grc - /home/bo
File Edit View Run Tools Help
jammer2
fequency_sweep
Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI
Python Module
ID: python_mod
QT GUI Label
ID: frequency
Label: frequency
Default Value: 2.432G
QT GUI Label
ID: variable_qtgui_label_0
Label: amp
Default Value: 0
osmocom Source
Sync: Unknown PPS
Number Channels: 1
Sample Rate (sps): 32k
Ch0: Frequency (Hz): 2.432G
Ch0: Frequency Correction (ppm): 0
Ch0: DC Offset Mode: 0
Ch0: IQ Balance Mode: 0
Ch0: Gain Mode: False
Ch0: RF Gain (dB): 50
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20
Ch0: Bandwidth (Hz): 20M
command
out
QT GUI Frequency Sink
FFT Size: 1024
Center Frequency (Hz): 2.432G
Bandwidth (Hz): 32k
Variable
ID: samp_rate
Value: 32k
Variable
ID: freq
Value: 2.432G
Function Probe
ID: fun_prob
Block ID: probsign
Function Name: level
Poll Rate (Hz): 5
Signal Source
Sample Rate: 32k
Waveform: Constant
Frequency: 0
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0
out
Throttle
Sample Rate: 32k
Limit: None
in
out
QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 32k
Autoscale: No
Probe Signal
ID: blocks_probe_signal_x_0


Open
python_mod_4y1cd8__.py
/tmp
# this module will be imported in the into your flowgraph
f1 = 2430000000
f2 = 2480000000
f = f1
step = 1000000
def sweeper():
global f1,f2,f,step
f += step
if f>= f2:f=f1
return f


Block paths:
/usr/share/gnuradio/grc/blocks
Loading: "/home/boeing/jammer2.grc"
>>> Done
Loading: "/home/boeing/fequency_sweep.grc"
>>> Done
>>> No editor selected.
ID
Value
Imports
Variables
freq 2432000000
frequency 2432000000
fun_prob 0
samp_rate 32000
variable_qt 0

Difficulties

- Obtaining detailed information about Bluetooth and Zigbee vulnerabilities
- Timeline of ordering & receiving equipment
- Moving rooms in order to work & short class time
- Zigbee software is windows based; cannot run on Raspberry Pi
- Unable to get Zigbee to successfully connect
- One HackRF cannot jam entire frequency; will need to order more
- Could not get GNU Radio to sweep properly
- Could not get spectrum analyzer properly setup in Linux

Future Goals

- Finalize proper test bed functioning (particularly Zigbee)
- Begin with performing jammer attack
- Attempt to exploit and explore other vulnerabilities within these devices
- Write-up of our findings

Thank You Boeing!