R.I.C.E

March 1, 2019

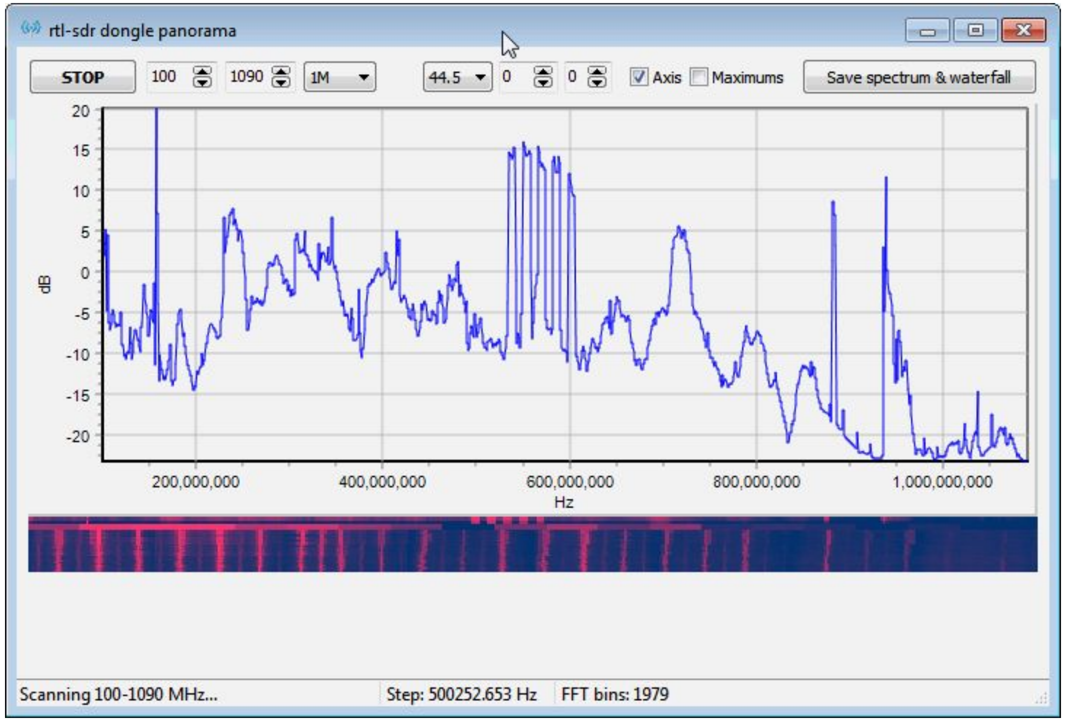
Jacob Fletcher, Emil Mathew, Matthew Sumpter

Components

- TX-2: Embedded computing device with optimizations for ML. Built around an NVIDIA Pascal-family GPU with 8GB Memory and 59.7GB/s memory bandwidth.
- RTL SDR and LIME SDR.
- Antenna
- GNURadio: Software that provides signal processing blocks to implement software radios.
- GR-OSMOSDR Block: Allows us to be hardware agnostic when it comes to the source block for GNURadio. Detects RTLSDR.
- GR-RDS: Used for the decoding of FM RDS data with the RTL-SDR.
- GR-RFTAP: This module implements the RFtap Encapsulation block, used to encapsulate Radio Frequency (RF) metadata about packets for Wireshark.
- Wireshark: Open source packet analyzer.
- RTLSDR Scanner: A cross platform Python frequency scanning GUI for USB TV dongles, using the OsmoSDR rtl-sdr library.
- Python Scripts utilized to automate and collect RF metadata.

RTL_SDR Scanner

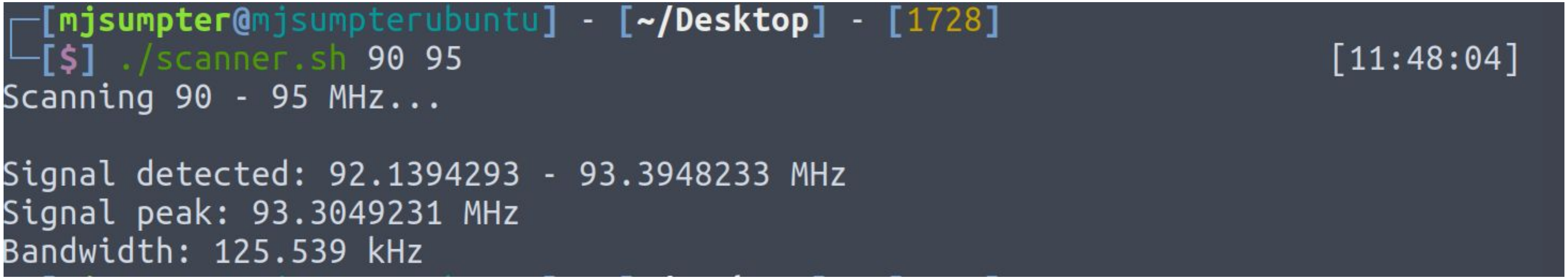
- Goal of Automating Signal Reception
- No tools for signal detection w/o GUI
- Analyze .csv output



Scanning 100-1090 MHz... Step: 500252.653 Hz FFT bins: 1979

RTL_SDR Scanner

- Analyze .csv with highdb_bandwidth.py
  - Average detected signal - “noise floor”
  - Set threshold based on difference between noise floor and maximum detected signal
  - When detected power exceeds threshold, triggers tracking variables

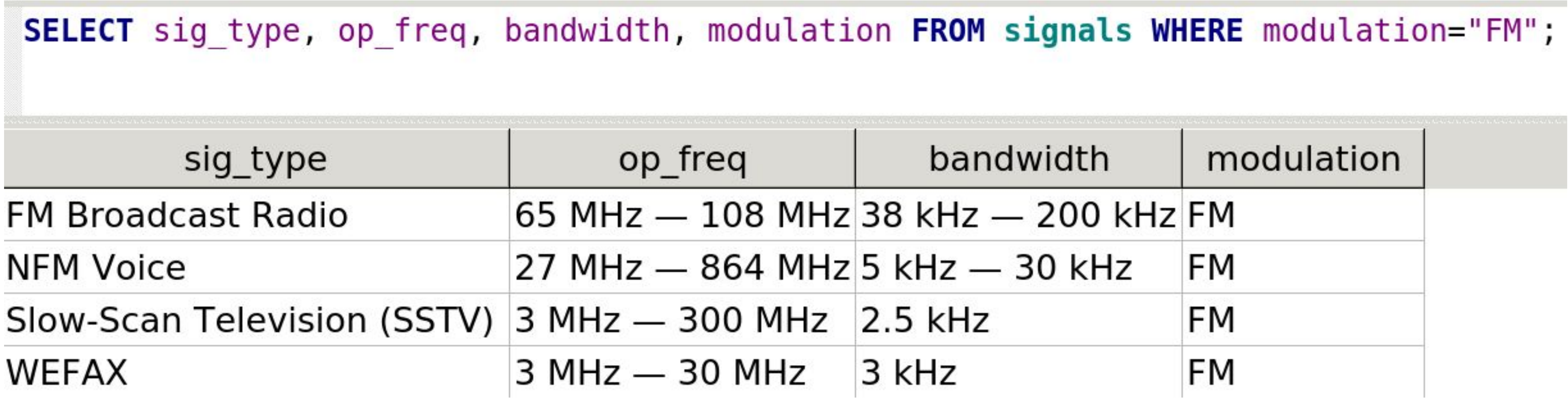


```
┌─[mjsumpter@mjsumpterubuntu] - [~/Desktop] - [1728]
└─[$] ./scanner.sh 90 95                                          [11:48:04]
Scanning 90 - 95 MHz...

Signal detected: 92.1394293 - 93.3948233 MHz
Signal peak: 93.3049231 MHz
Bandwidth: 125.539 kHz
```

RTL_SDR Scanner

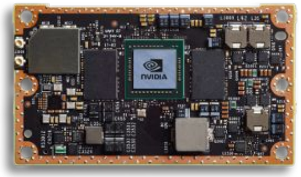
- Improving .csv analyzer
  - Currently “anecdotal” - works for expected behavior
  - Fourier Transform analysis
- Inject peak frequency to GNUradio
- Catalogue bandwidth data for classification/database query



```
SELECT sig_type, op_freq, bandwidth, modulation FROM signals WHERE modulation="FM";
```

sig_type	op_freq	bandwidth	modulation
FM Broadcast Radio	65 MHz — 108 MHz	38 kHz — 200 kHz	FM
NFM Voice	27 MHz — 864 MHz	5 kHz — 30 kHz	FM
Slow-Scan Television (SSTV)	3 MHz — 300 MHz	2.5 kHz	FM
WEFAX	3 MHz — 30 MHz	3 kHz	FM

TX2



- Problems Faced:
  - ○ TX-2 is hardware physically located in FL. Team was distributed around the country.
  - ○ Initial testing of configuration was successfully achieved on a VM with x64 processor. ARM processor posed challenges.
  - ○ Errors in building and missing dependencies meant TX2 had to be reflashed if the cause of failure was not detected.

TX2

- Solutions utilized:
  - TX-2 was configured with static IP Addr in order to allow for SSH and VNC connections.
  - Team had to breakdown install methodology into steps in order to identify which component was causing compile errors. Team was not able to identify a substitute for a VM that provided quick saves and reverts similar to VM snapshots.
  - Team was able to configure system with successful operation. Team continued on to clone the SD card of the successful build.

TX2

- Disadvantages of HW:
  - Inability to quickly revert during build tests. Architecture is difficult to emulate in Virtual Environments.
  - Unique components cause issues when it comes to open source libraries and software written for x86 and/or x64 systems.

TX2

- How to Improve:
  - Develop better documentation during install and debugging.
  - Explore emulation options if available.

Dependency Dependency Dependency Depends

- Programs that rely on other programs to rely on
- RTLSDR_Scanner relied on
  - Numpy,Matplotlib,pyserial,wxpython and 5 others
- Issues occur when the system cannot find the required issues
- Common causes:
  - Mixed Building from source and building via installer,packages etc…
  - Building with mixed permissions
  - Dependencies are the wrong version

This is out control….version

- noun..the task of keeping a software system consisting of many versions and configurations well organized.
- Issues
    - Team members will duplicate efforts to solve problems
    - All efforts can be lost on a project during a catastrophic failure
- Solutions
    - Software: Github, BitBucket, Docker
    - OS: VMs, Docker
- ALWAYS test your demo the day before ,and DO NOT make changes to it until after

The Double Cutting Edge of Technology

- Requirements/Issues
  - ○ Lack of Consensus
  - ○ Lack of Testing
  - ○ Resistance by establishment
- Tips
  - ○ Network and knowledge sharing
  - ○ Break problem into smaller obtainable pieces

Persistence pays off

Future Work – RF Dictionary/ML

- Utilizing SIG ID Wiki (see references) DB as local reference.
- Adapt current meta classifier for TX2 function
- Coordinate with DeepSIG for omniSIG on TX2

Resources

https://kb.ettus.com/Building_and_Installing_the_USRP_Open-Source_Toolchain_(UHD_and_GNU_Radio)_on_Linux

https://www.sigidwiki.com/wiki/Database