Insights from Waves: Gathering Forensically-useful Insights from IoT Devices with Electromagnetic Side-Channel Analysis

DFRWS EU 2020, Oxford, UK.

UCD Forensics and Security Research Group

Outline

- Introduction (30 mins)
- GNURadio Companion Flowgraphs (45 mins)
- Software Defined Radio Programming (45 mins)
- Module Development for EMvidence Framework (45 mins)

Introduction (30 mins)

UCD Forensics and
Security Research Group

Digital Forensics

- Acquiring data from computing devices that can help to progress investigations.
- Increasing usage of computing devices adding up to more forensic data sources.
- More and more computing devices are employing encryption to store data.
- Most digital forensic literature assumes either that,
  - Cryptography is not employed
  - Cryptography is bypassed somehow as a part of the legal process.
- It is not possible to ignore the threat posed by encrypted devices.

Forensics of Internet of Things

- IoT opens up new evidence sources from unexpected places.

  ...health implants, sports wearables, smart burglar alarms, smart thermostats...

- Highly heterogeneous device designs.

- Application of encryption worsen the usability of IoT in forensics.

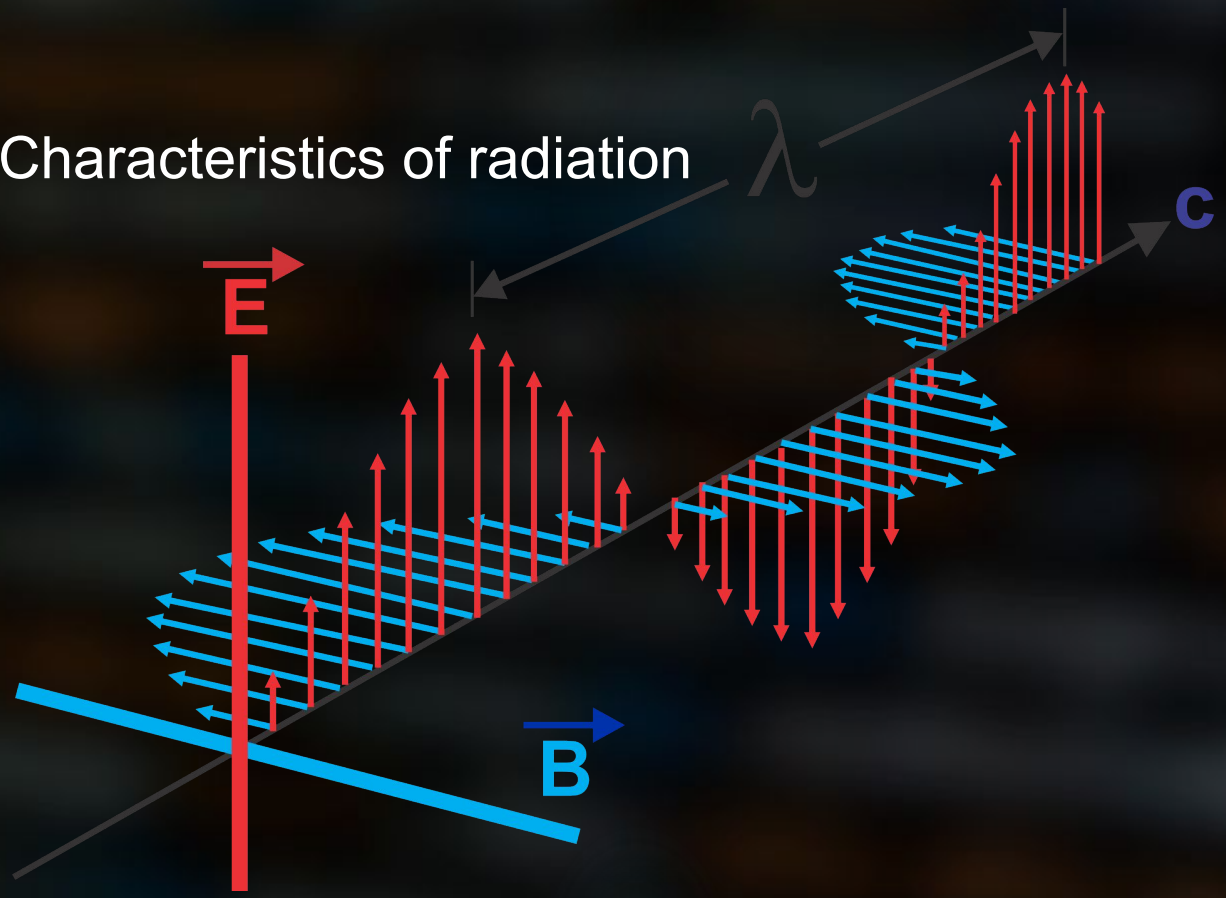
Electromagnetic Side-Channel Analysis

Time-varying electrical currents ⇒ Electromagnetic radiation

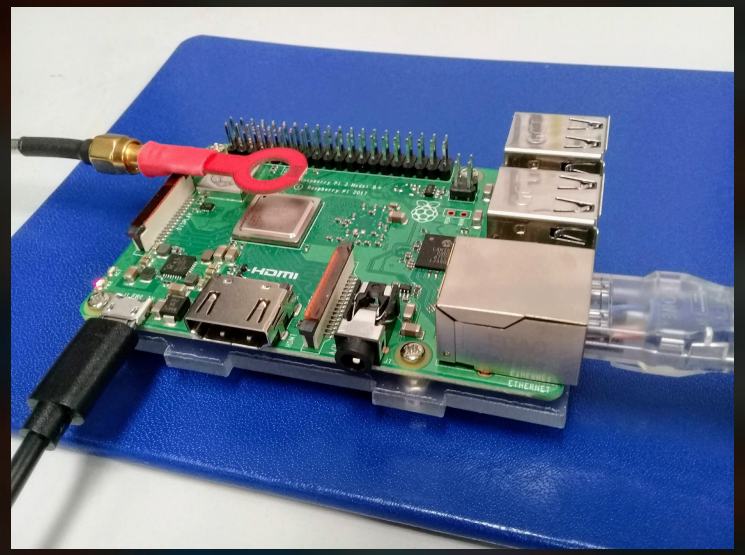
Nature of the time-varying current ⇐ Characteristics of radiation

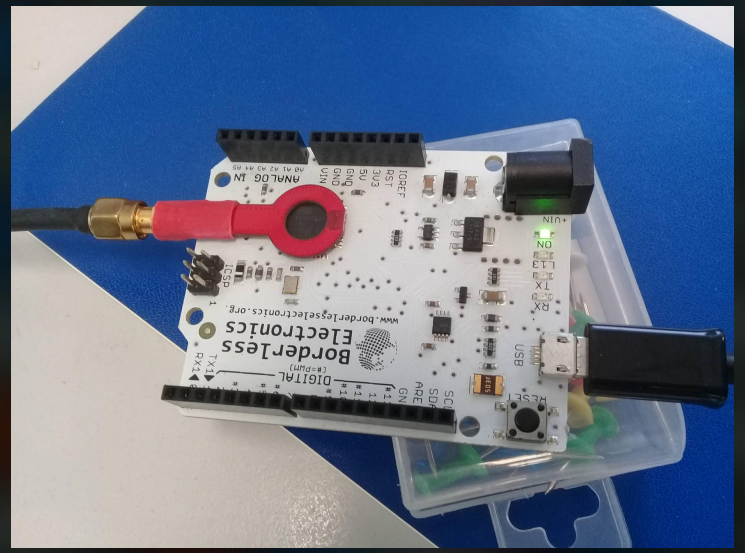
EM radiation from computer processors leak information

EM side-channel analysis (EM-SCA)

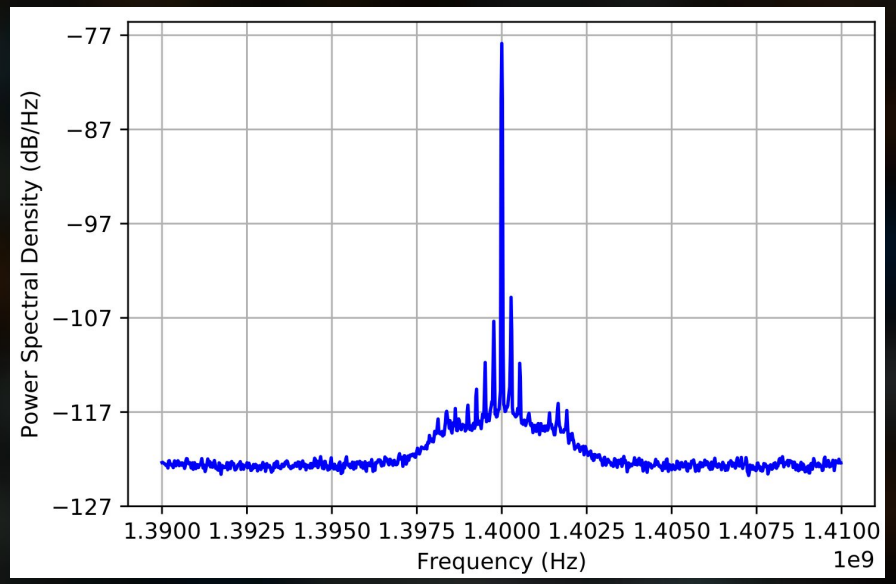


Observation of EM Side-channel





- CPU clock/oscillator is the main source of EM noise.
- EM emissions can be observed at clock frequency and its harmonics.
- Signal attenuates rapidly with distance from the CPU.
- H-loop antennas placed closer to the CPU can pick up strong signals.
- When the fundamental frequency is noisy due to external sources, harmonics can be used.



Electromagnetic Side-Channel Analysis

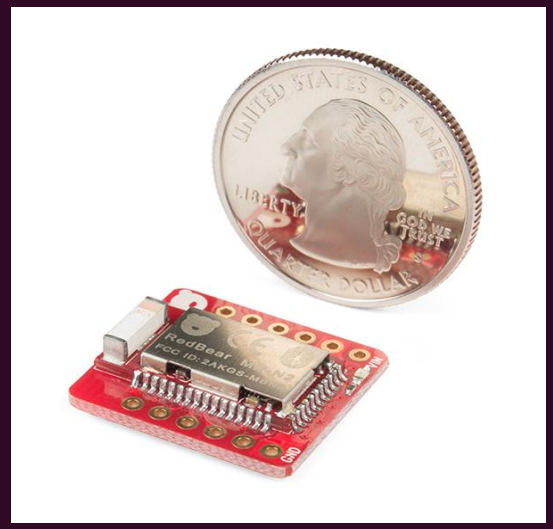
- EM-SCA has been applied to recover cryptographic keys,

  e.g., Camurati et al. (2018)
- Target device: BLE-Nano running AES-128 encryptions.
- ~ 8000 EM trace samples.
- Correlation electromagnetic

  analysis (CEMA)
- 18 minutes to recover the key

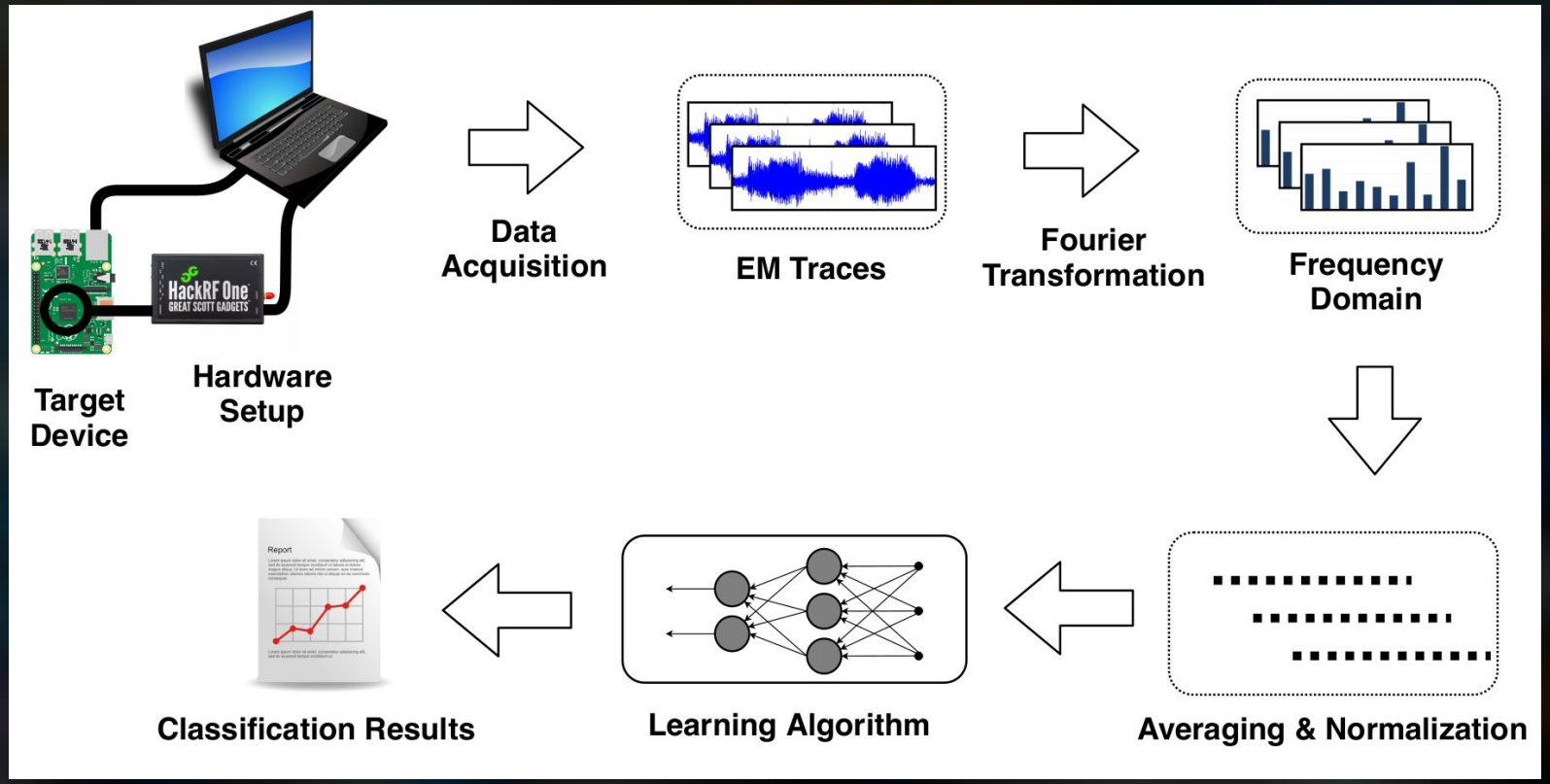
Current EM-SCA assumes that the attacker has physical access to the device to make any tampering.
In forensics, we don't!

```
Subkey 15, hyp = ed:  0.012530354407226786
Subkey 15, hyp = ee:  0.017266581888572823
Subkey 15, hyp = ef:  0.010556395126552152
Subkey 15, hyp = f0:  0.02396490987572155
Subkey 15, hyp = f1:  0.012949217484922705
Subkey 15, hyp = f2:  0.014502574032304778
Subkey 15, hyp = f3:  0.014963314285949247
Subkey 15, hyp = f4:  0.012954752080796888
Subkey 15, hyp = f5:  0.0130315561783500 3
Subkey 15, hyp = f6:  0.013772034631913068
Subkey 15, hyp = f7:  0.019364248397445407
Subkey 15, hyp = f8:  0.010932180008903168
Subkey 15, hyp = f9:  0.013027691526298332
Subkey 15, hyp = fa:  0.01665869128411864
Subkey 15, hyp = fb:  0.015427833690631214
Subkey 15, hyp = fc:  0.011935004419024819
Subkey 15, hyp = fd:  0.014665594979696694
Subkey 15, hyp = fe:  0.018540794601137632
Subkey 15, hyp = ff:  0.018274501184871804
Best Key Guess:   56   89   ed   be   4c   65   db   f2   aa   2c   bd   c9   26   42   e6   e1
Known Key:        56   89   ed   be   4c   65   db   f2   aa   2c   bd   c9   26   42   e6   e1
PGE:             000  000  000  000  000  000  000  000  000  000  000  000  000  000  000  000
SUCCESS:           1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
NUMBER OF CORRECT BYTES: 16
```

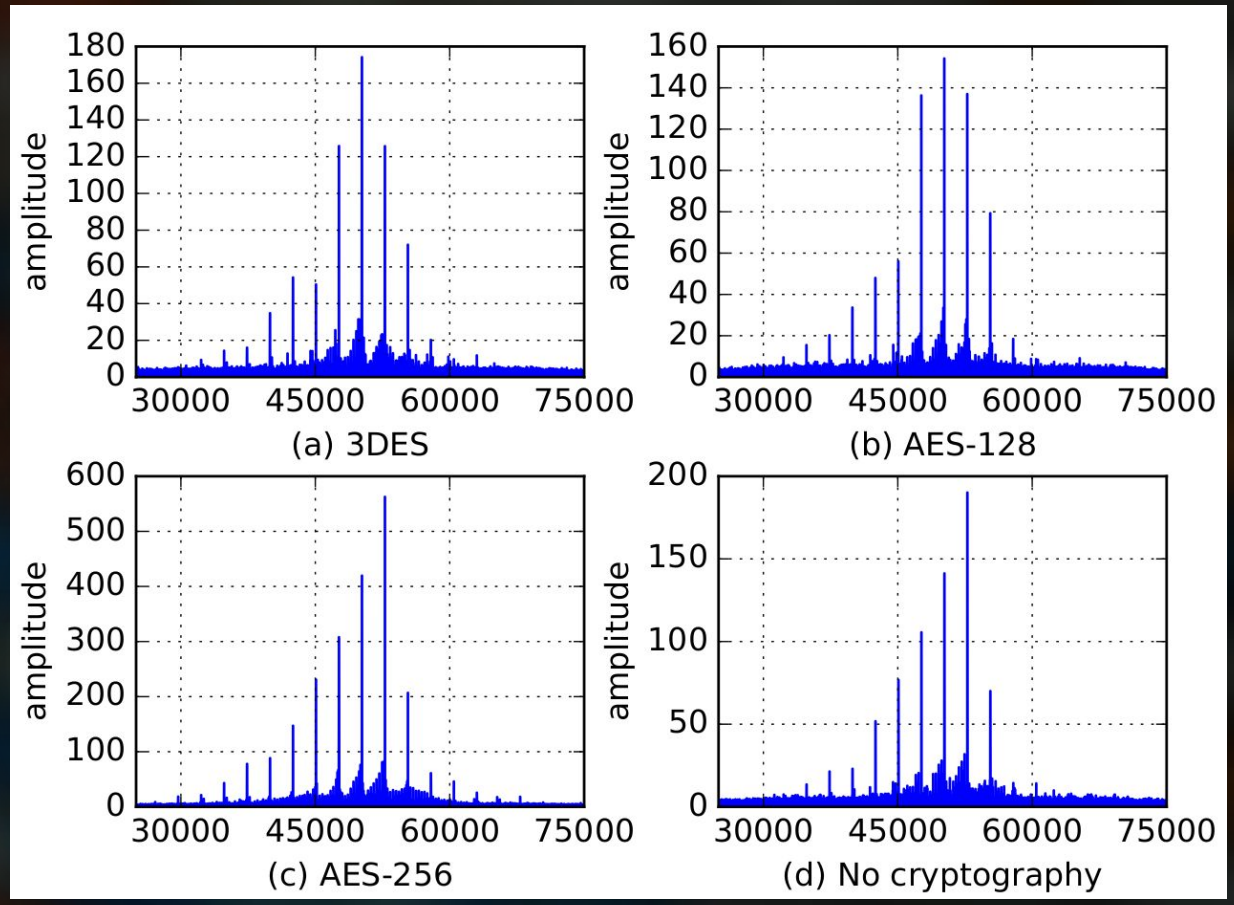


Some Experimental Results

1. Discriminating cryptographic activities
2. Detection of software behaviour
3. Detecting modifications to firmware



Discriminating Cryptographic Activities



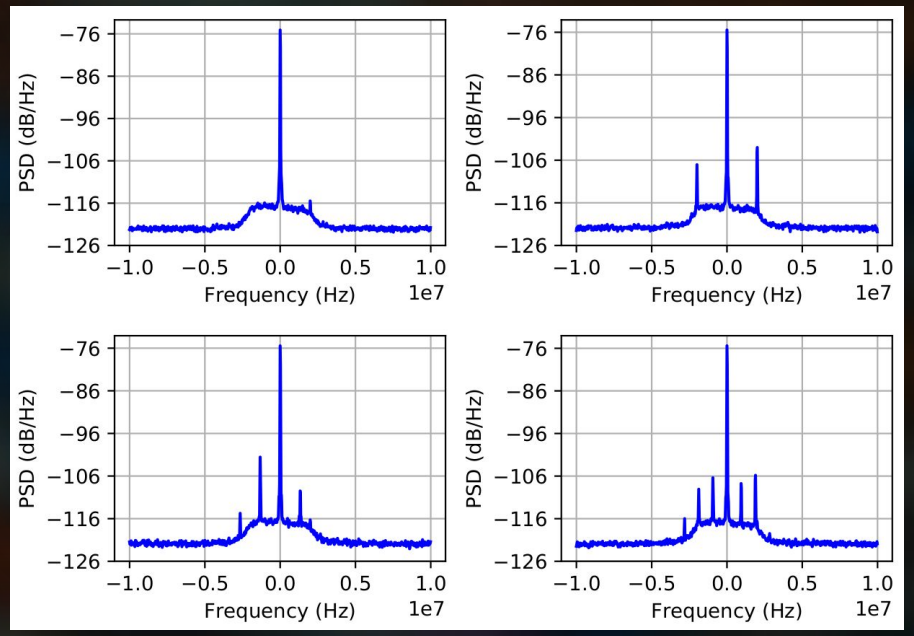
- Raspberry Pi as the target device.
- Three cryptographic classes and a ``no cryptography" class.
- From FFT to 500 features by averaging.
- 4 layer NN (2 hidden - 10x5)
- 600 samples per class.

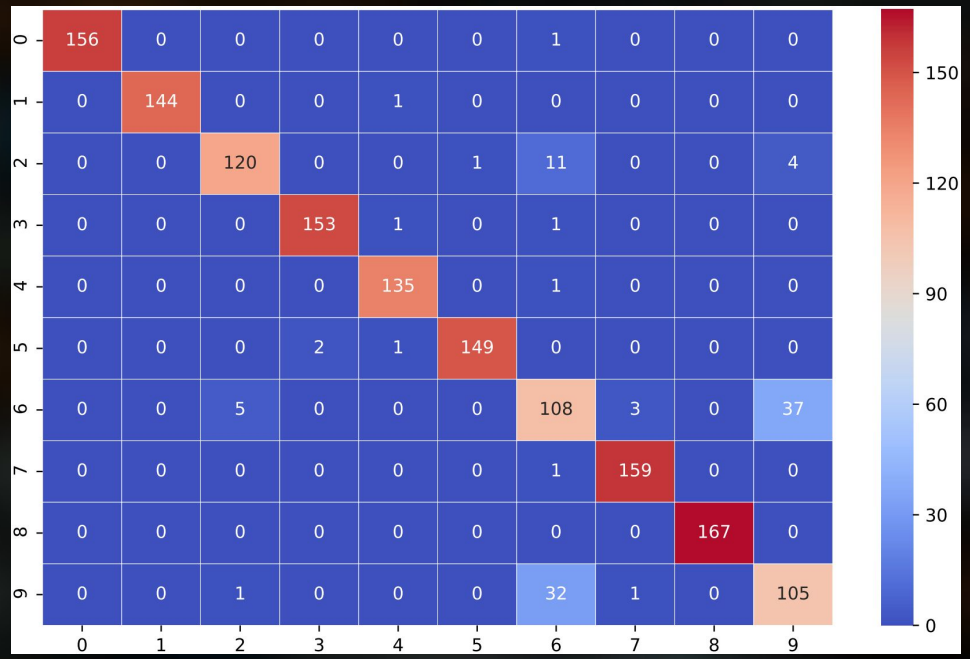
Activity	Precision	Recall	F1-Score
Other	0.93	0.85	0.89
AES-256	0.78	0.86	0.82
AES-128	0.99	0.92	0.95
3DES	0.81	0.85	0.83

Detection of Software Behaviour


```
/* Arduino test program */
void setup(){
}
void loop(){
    for(int i=0, i<20, i++) { delay(10); }
    for(int i=0, i<20, i++) { delay(10); }
    /* further loops */
}
```

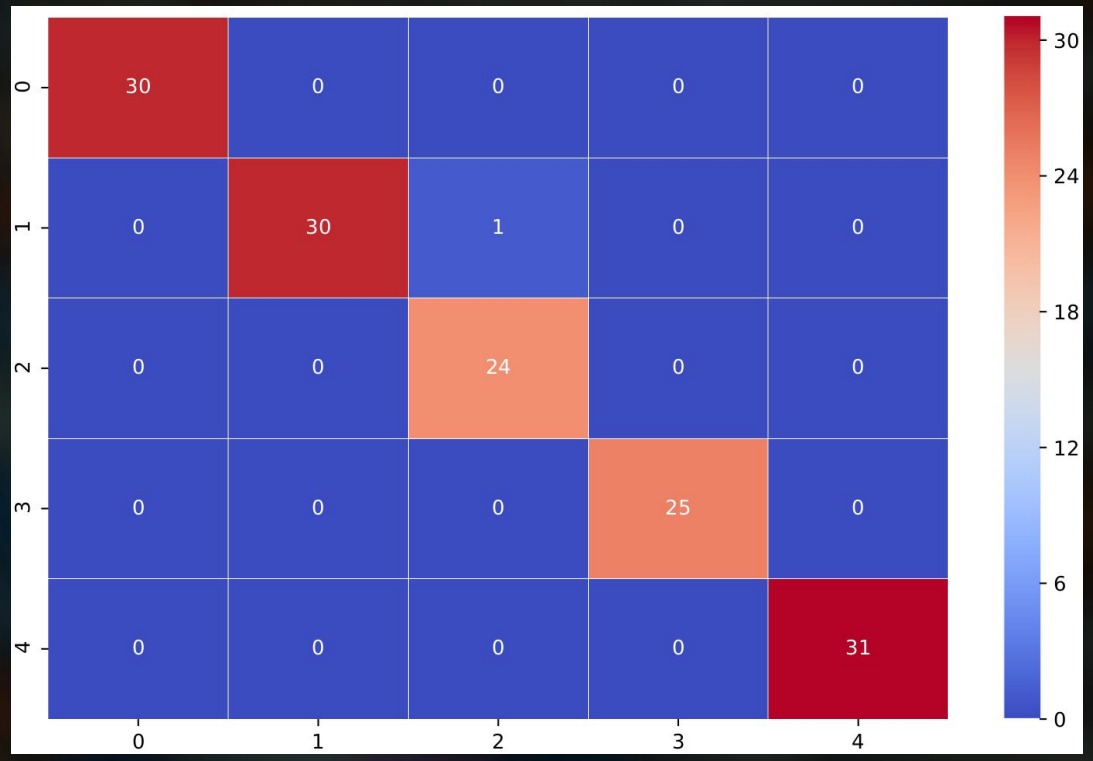
- Arduino Leonardo running 10 programs
- FFT (20,000,000) to a vector of 1000 features.
- 1000 buckets with max values.
- Over 90% classification accuracy

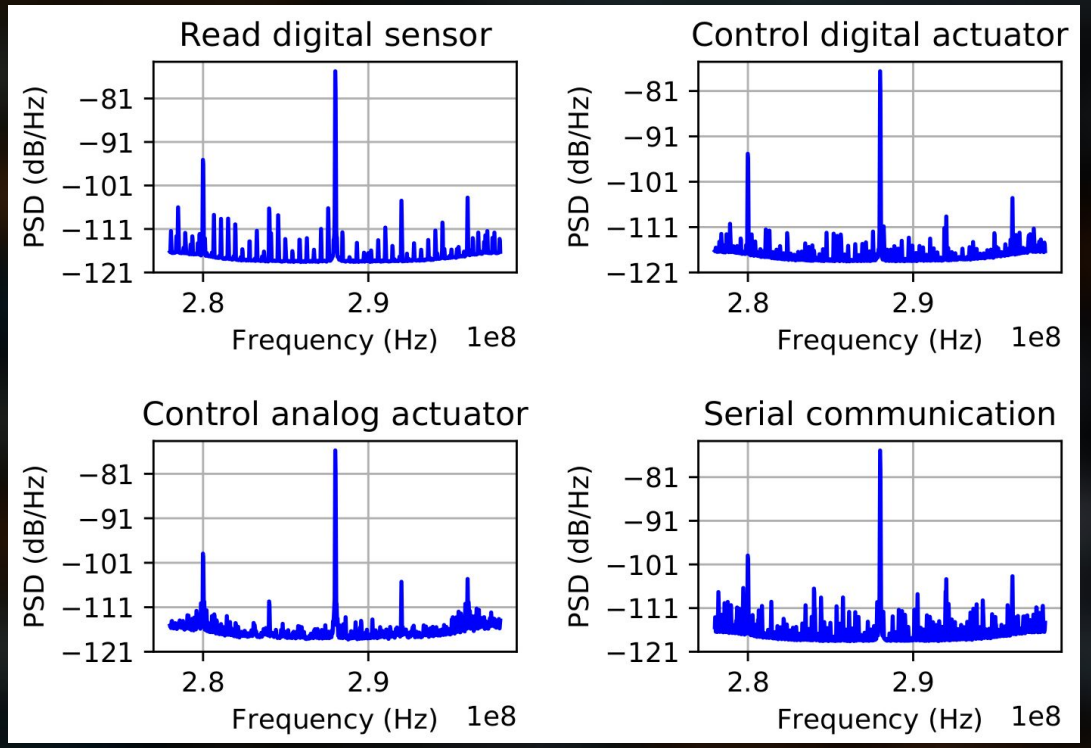




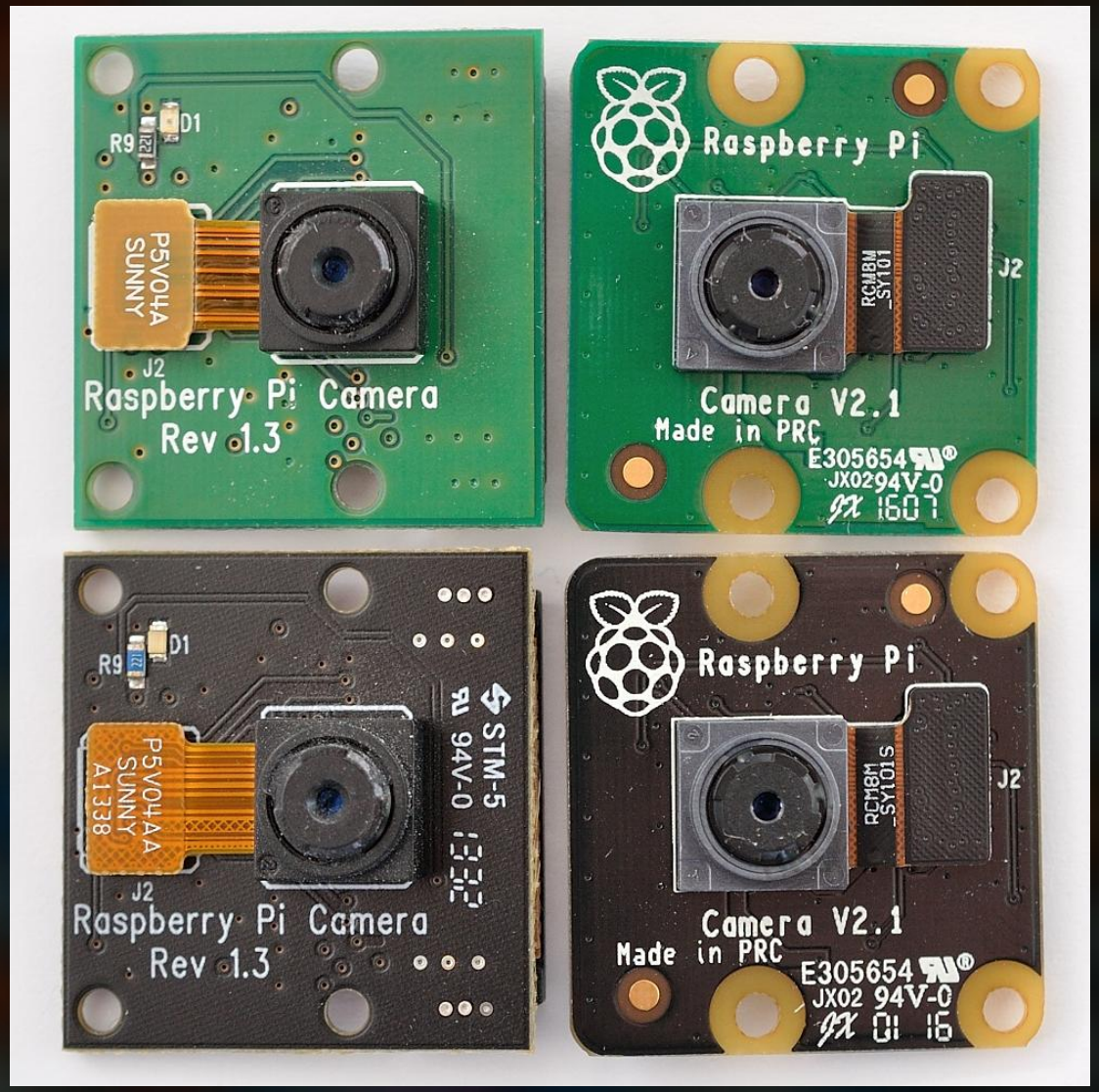
Detection of Software Behaviour (cont.)

- An IoT device with 5 internal states emulated using an Arduino.
- Similar neural network classifier like the previous case.
- Detection accuracy of over 99%





Detecting Modifications to Firmware



- Arduino Leonardo as the target device.
- FFT to 1000 features using max values.
- One-class SVM with a non-linear kernel (RBF).
- 1 legitimate program and 20 slightly modified programs for testing.
- 100% detection accuracy for all the tested programs.

What we've learned so far...

- We can identify known behaviors of software running on IoT devices.
- We can identify when known cryptographic algorithm implementations are running on an IoT device.
- Such insights can help inspecting an IoT device in a forensic investigative scenario.

- Ongoing work:

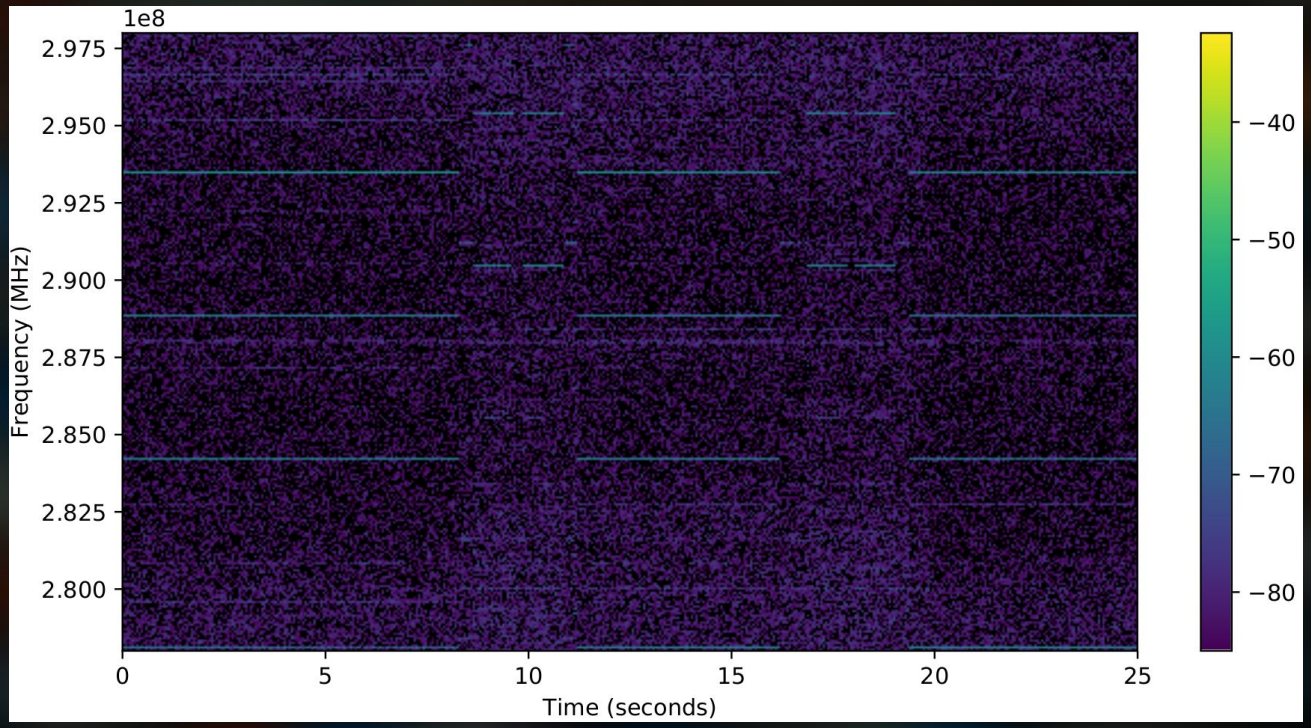
Cryptographic key retrieval on a forensic setting.

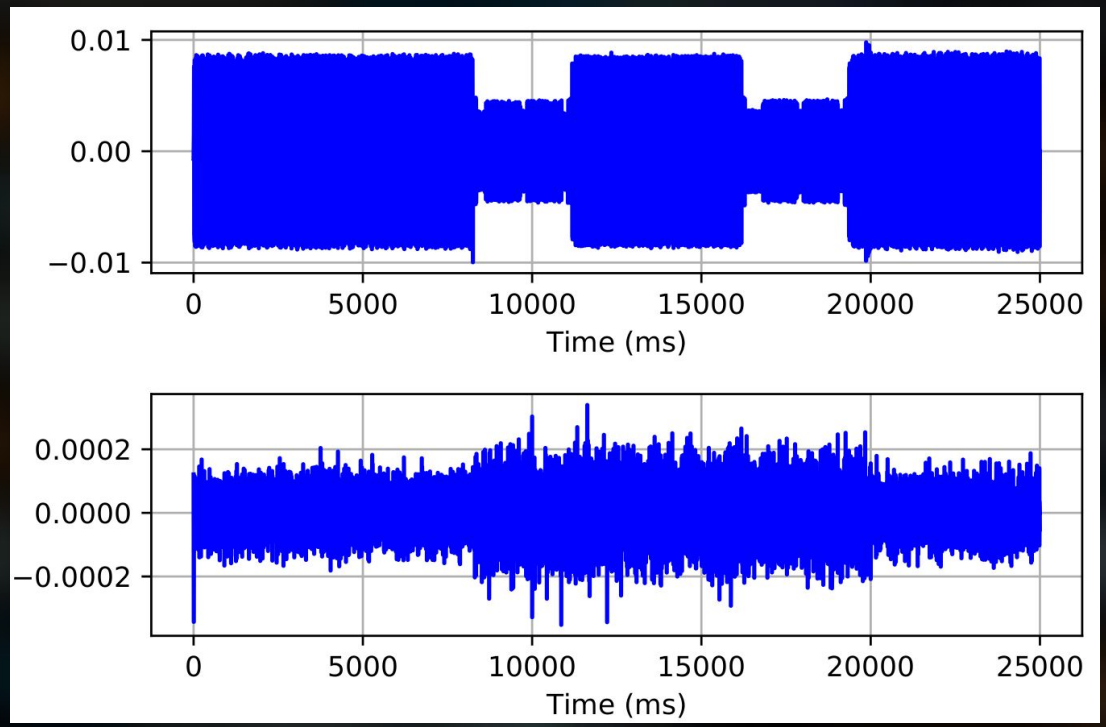
Cryptographic Key Retrieval on a Forensic Setting

- A key retrieval algorithm such as CEMA requires,
  a. a large number of EM trace files.
  b. each trace file to be enclosing a unique encryption/decryption operation using the same key.
  c. all the trace files to be precisely aligned, i.e., same time step in every trace should represent the same CPU activity.
- Currently, all these are achieved by instrumenting the target device in software or hardware means during data acquisition.
- When attacking a device in a digital forensic investigation, such tampering are not acceptable.

Cryptographic Key Retrieval on a Forensic Setting (cont.)

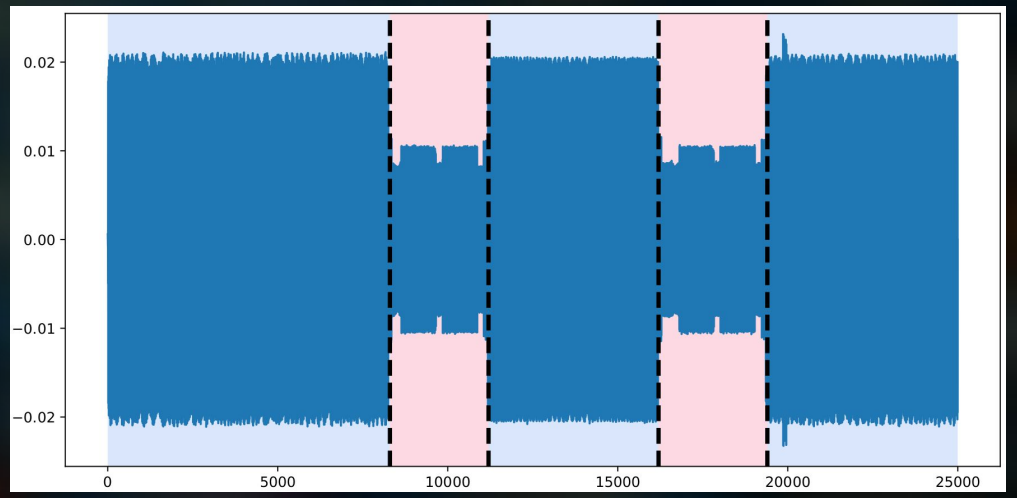
- Here's how signal looks like when running ECC encryption twice.
- Some channels clearly give away the starting and ending positions of encryption.

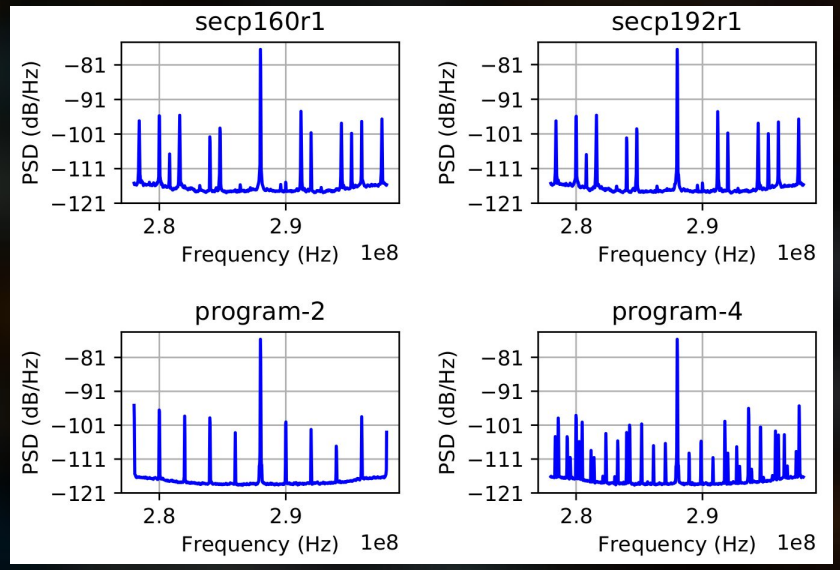




Cryptographic Key Retrieval on a Forensic Setting (cont.)

- A change-point detection algorithm, i.e., PELT, is used to segment the original signal into pieces.
- Cryptography-related pieces are identified using a classifier.
- However, key retrieval algorithm still doesn't work with such traces.
- Use of extremely small sliding windows is currently being explored.

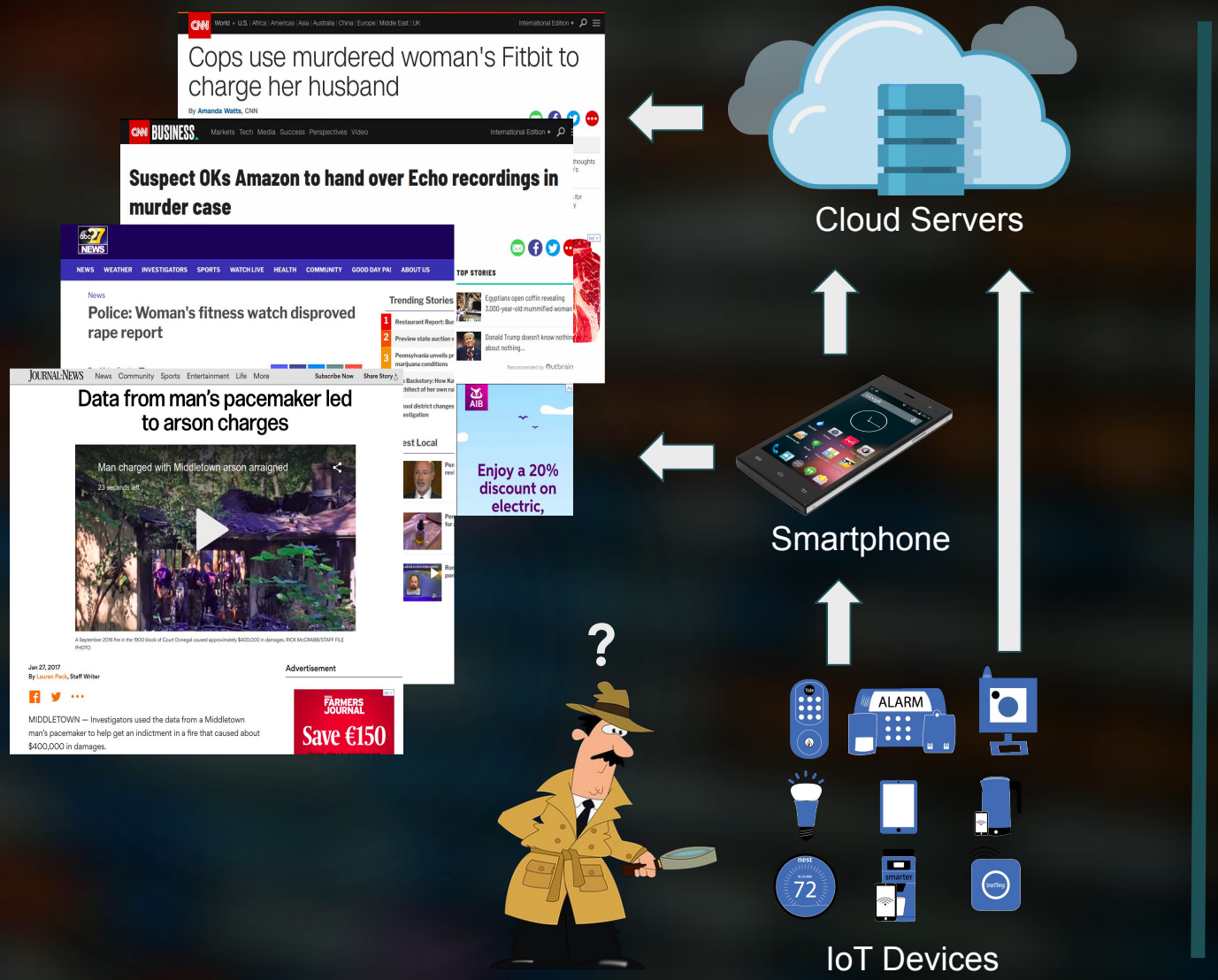




Limitations

- EM-SCA only applicable if the device is giving out sufficient EM radiation.
- Properly shielded devices are difficult to reach without high powered signal amplifiers.
- A firmware update can change the EM signature completely.
- Key recovery requires large number of traces.
  a. encryption should occur frequently
  b. need a sufficient time to observe as many encryption as possible
- Presence of multiple devices within the vicinity that produce EM radiation in similar frequencies can make the isolation of one device difficult.
- A huge variety of manufacturers/configurations for the same device - e.g., Amazon Echo.

Application of EM-SCA



- Smartphone apps and cloud servers are the window to most IoT devices.
- IoT devices are mostly black boxes due to,
  a. lack of standard forensic data gathering interfaces
  b. high heterogeneity of devices
  c. employment of encryption
- EM-SCA opens up a window to gather insights directly from the IoT devices.

Application of EM-SCA

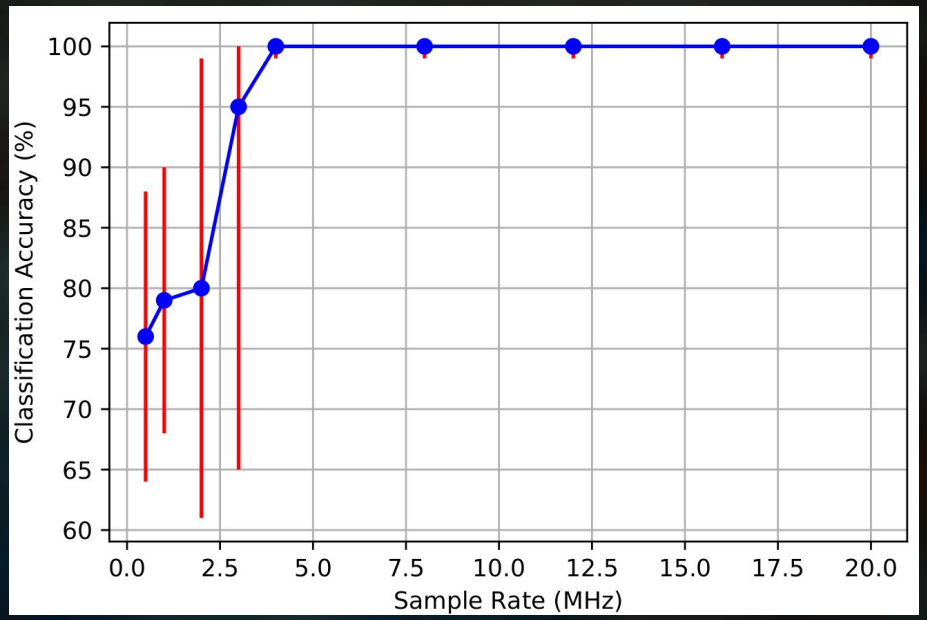
- An IP camera that takes photos when a motion is detected.
- Images are stored locally on an SD card with encryption.
- User can remotely initiate a video streaming which uses encryption as well.
- Cryptographic key is securely stored on the camera in a way not easily accessible to third parties.

How can I extract and decrypt a particular encrypted image on the SD card?

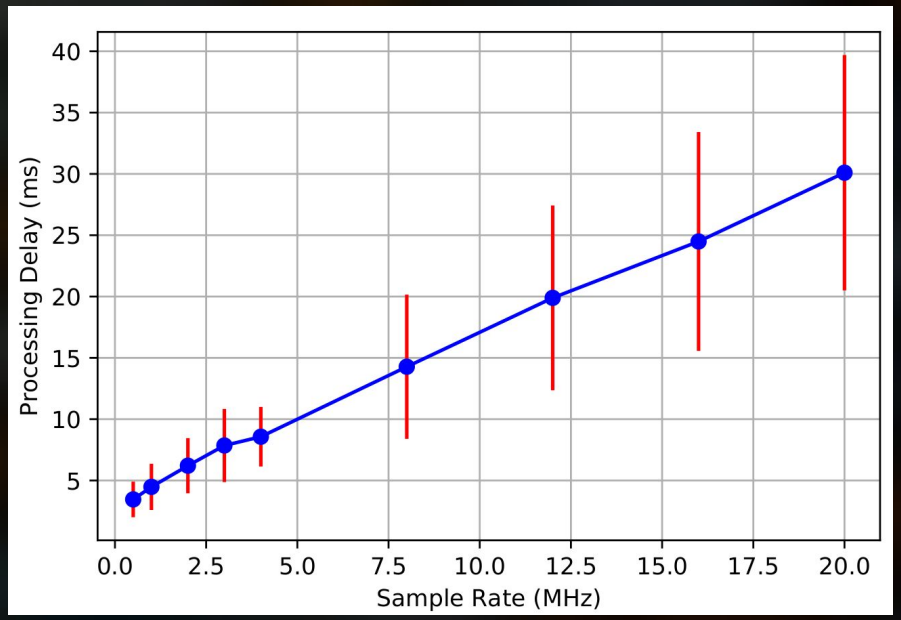
Storage and Real-time Requirements

- Each I-Q sample = 8 bytes
- Highest sampling rate = 20 MHz
- Size of the 1 minute signal capture ≈ 9 GB

  (8 bytes × 20 MHz × 60 seconds = 8.94 GB).



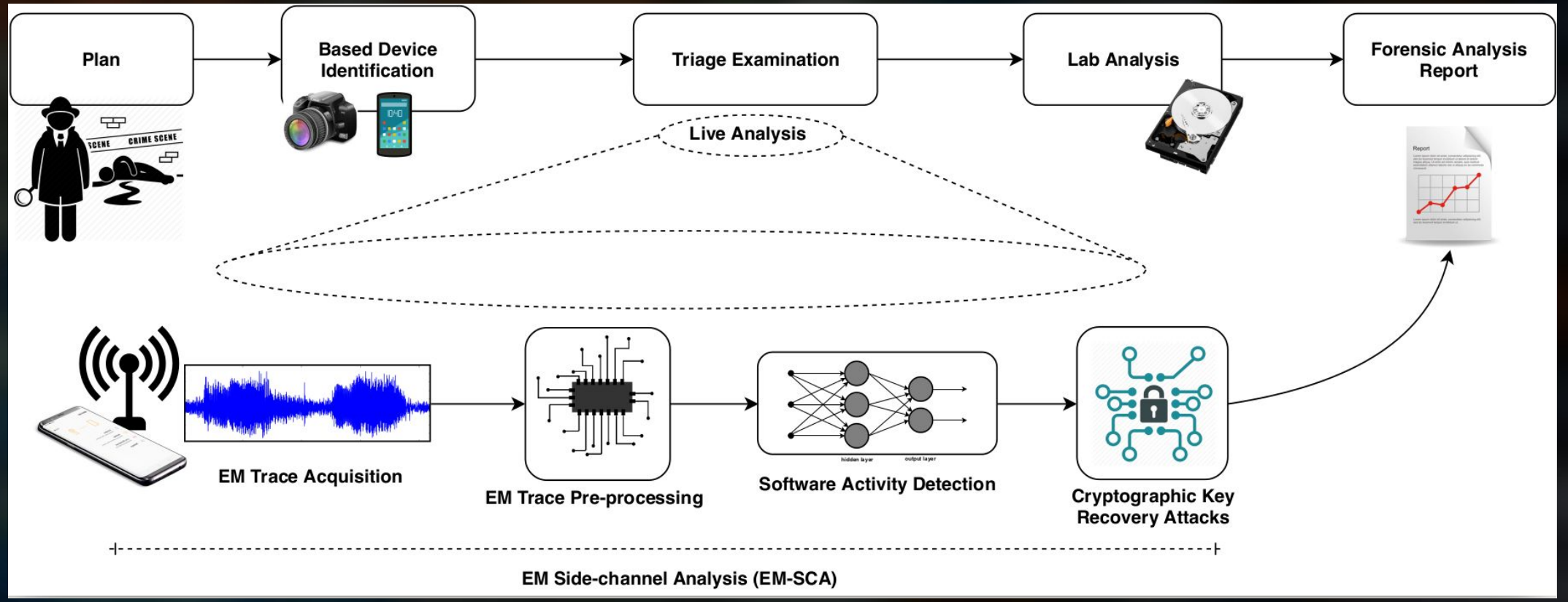
It's OK to have lower sampling rates to cope with storage requirements.



Even the highest sample rate does not exceed our capability to process data in real-time

Hence, live forensic analysis is possible!

Application of EM-SCA



Hardware for EM-SCA

Oscilloscopes / spectrum analyzers / traditional radio receivers

Difficult to handle in in digital forensic investigation settings.

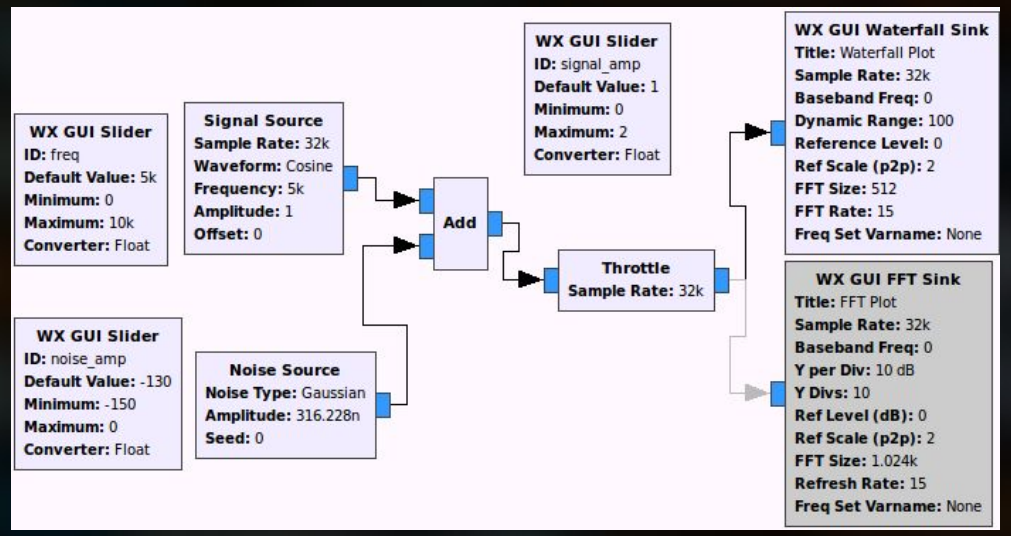
Software-defined radios (SDR)

Easily configurable with software.

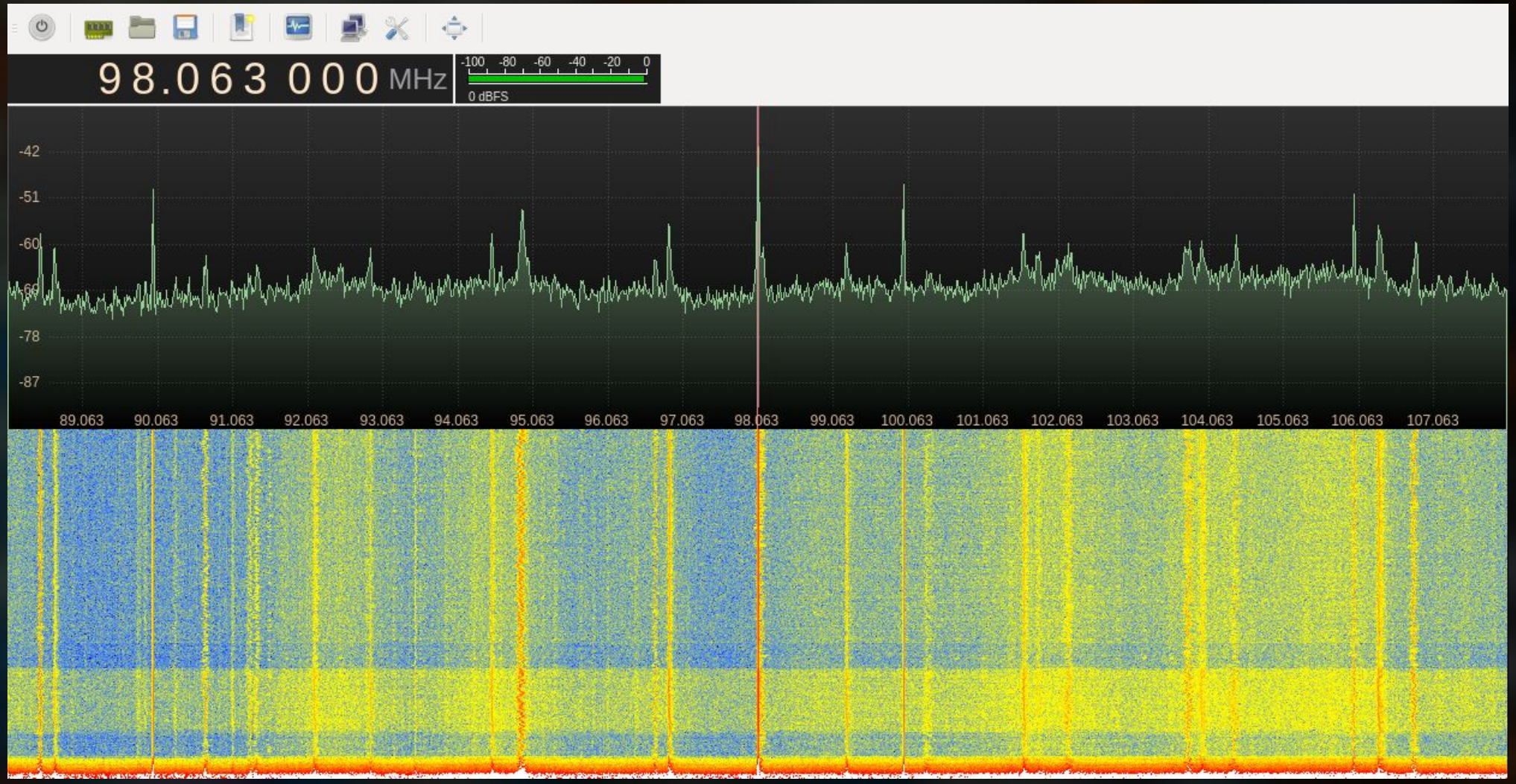
Software Defined Radio (SDR)

- A fast analog-to-digital converter (ADC).
- RTL-SDR, HackRF, USRP
- Generates digitized samples in Inphase-Quadrature (I-Q) format.
- Open source libraries to process streams of I-Q data samples.
- Can program for a task using Python or using a visual flowgraph editor, GRC.

GNU Radio
THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM



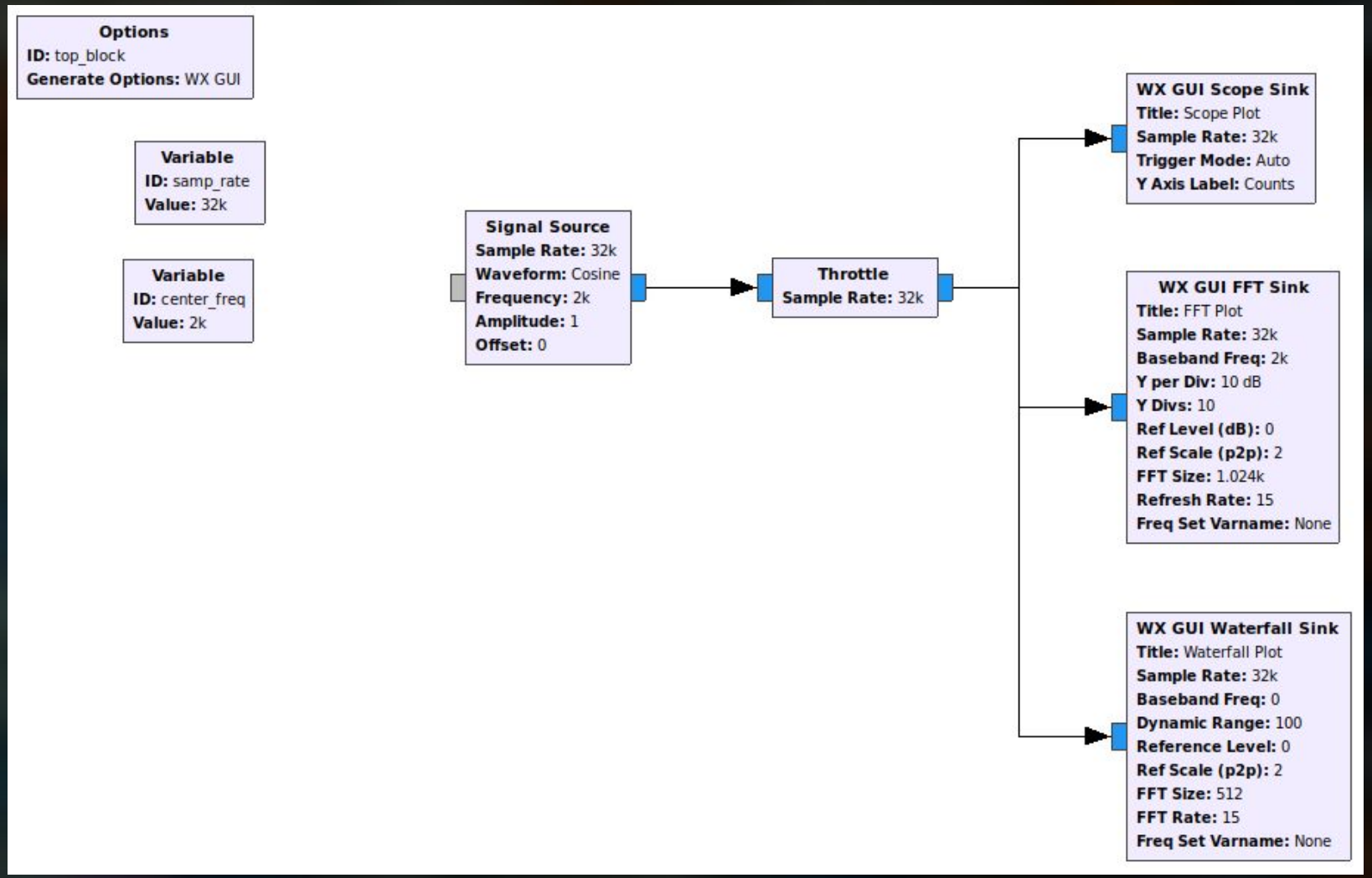
A walkthrough the spectrum with GQRX



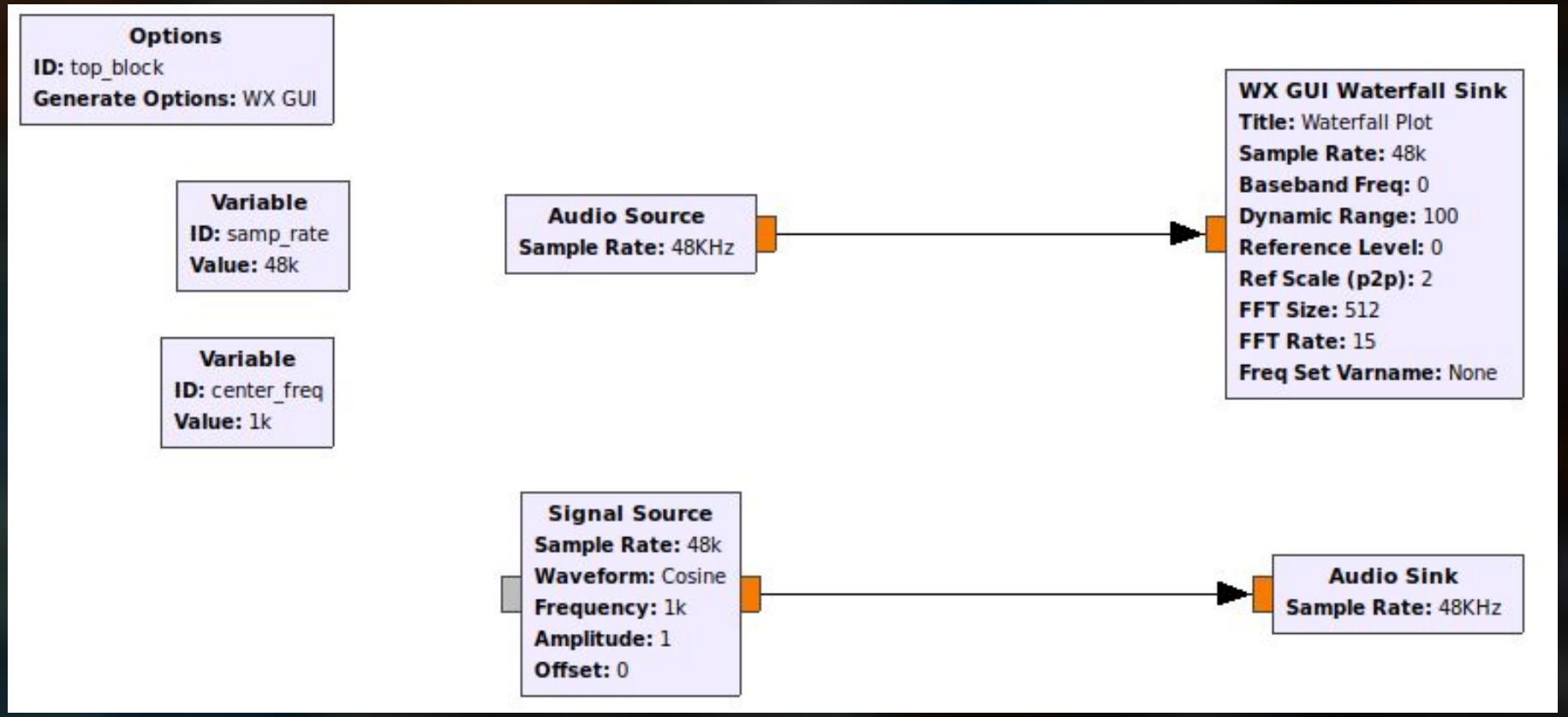
GNURadio Companion Flowgraphs (45 mins)

UCD Forensics and
Security Research Group

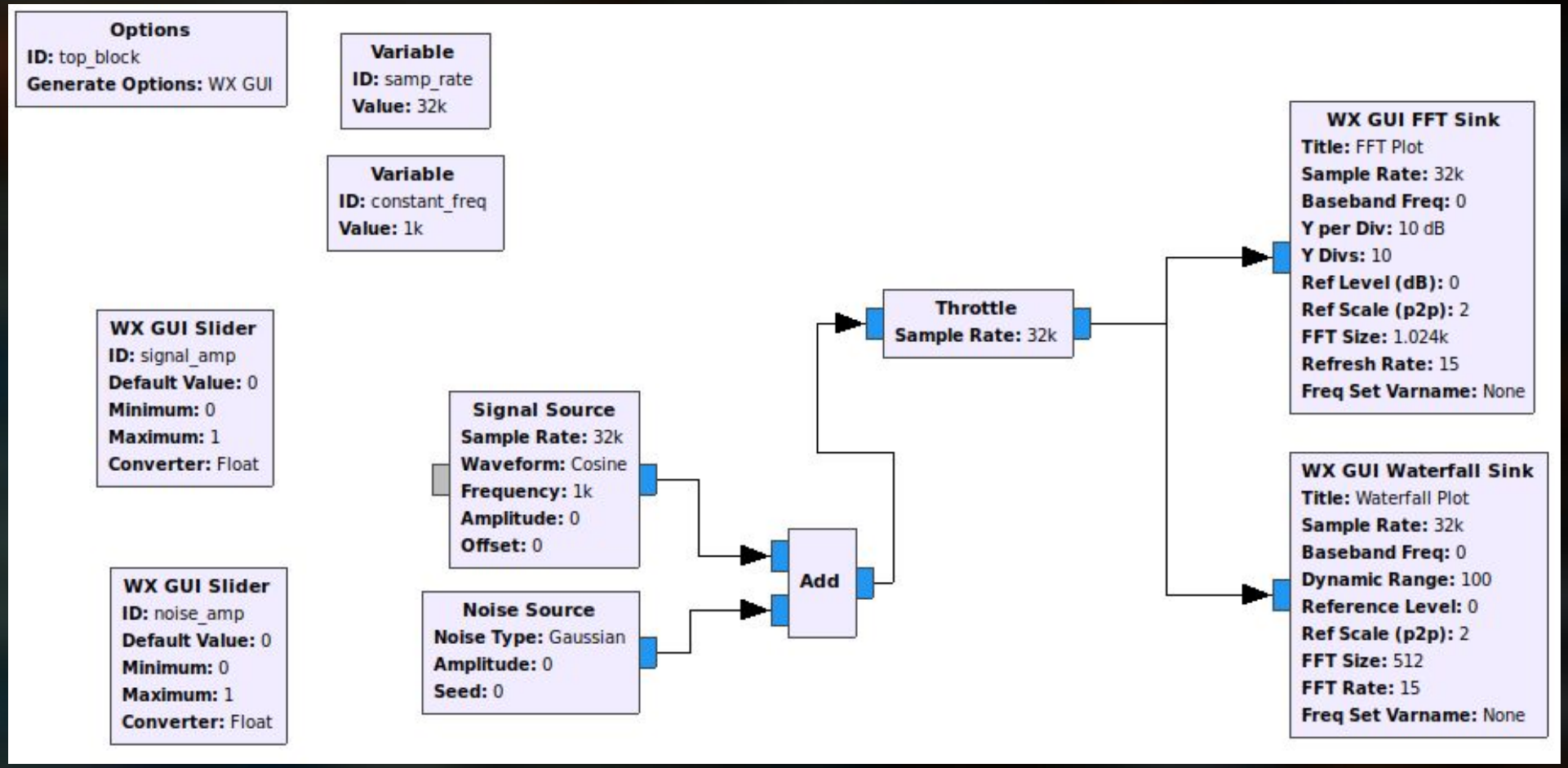
Generating Signals and Visualization



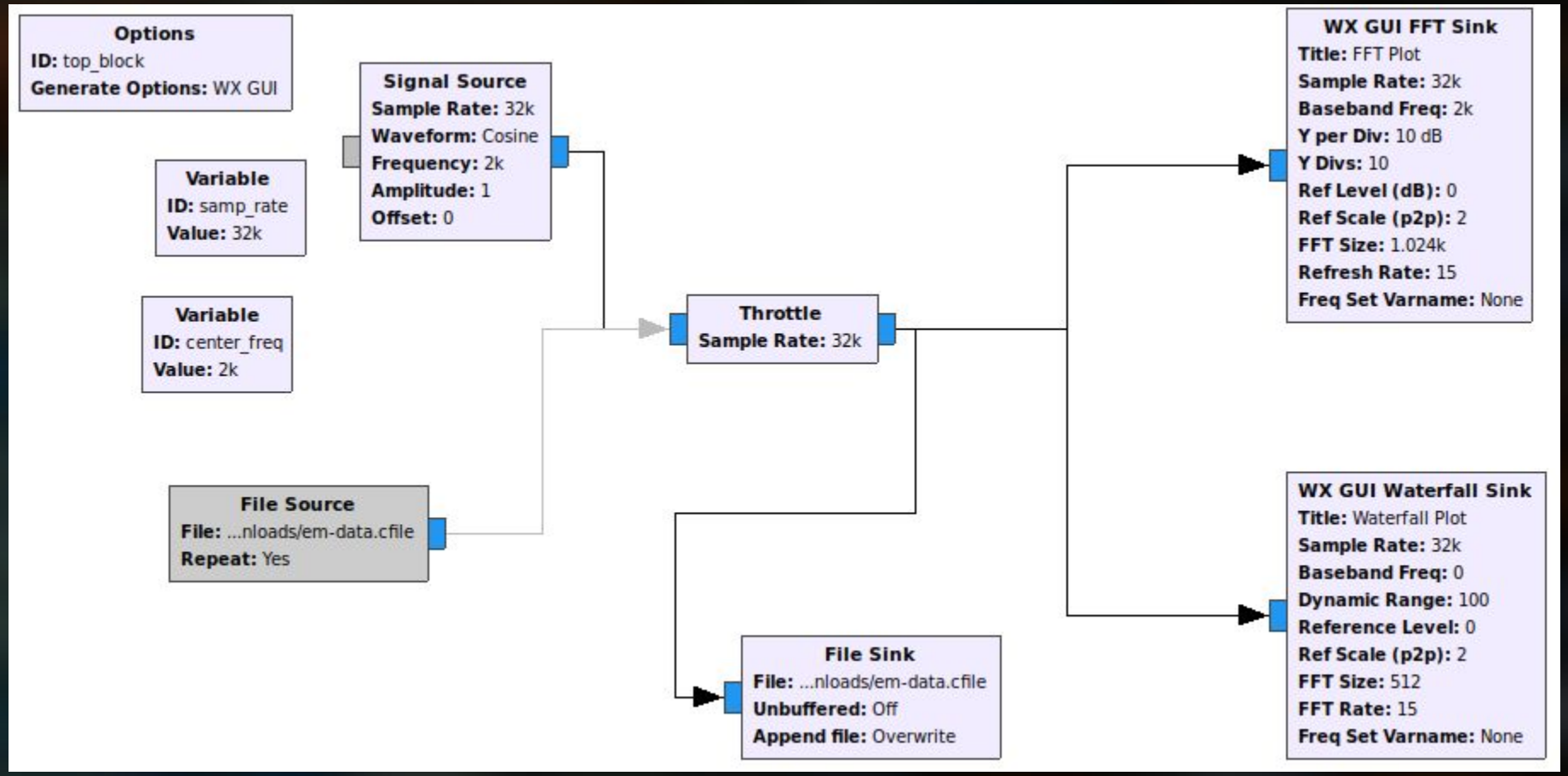
Audio Signals



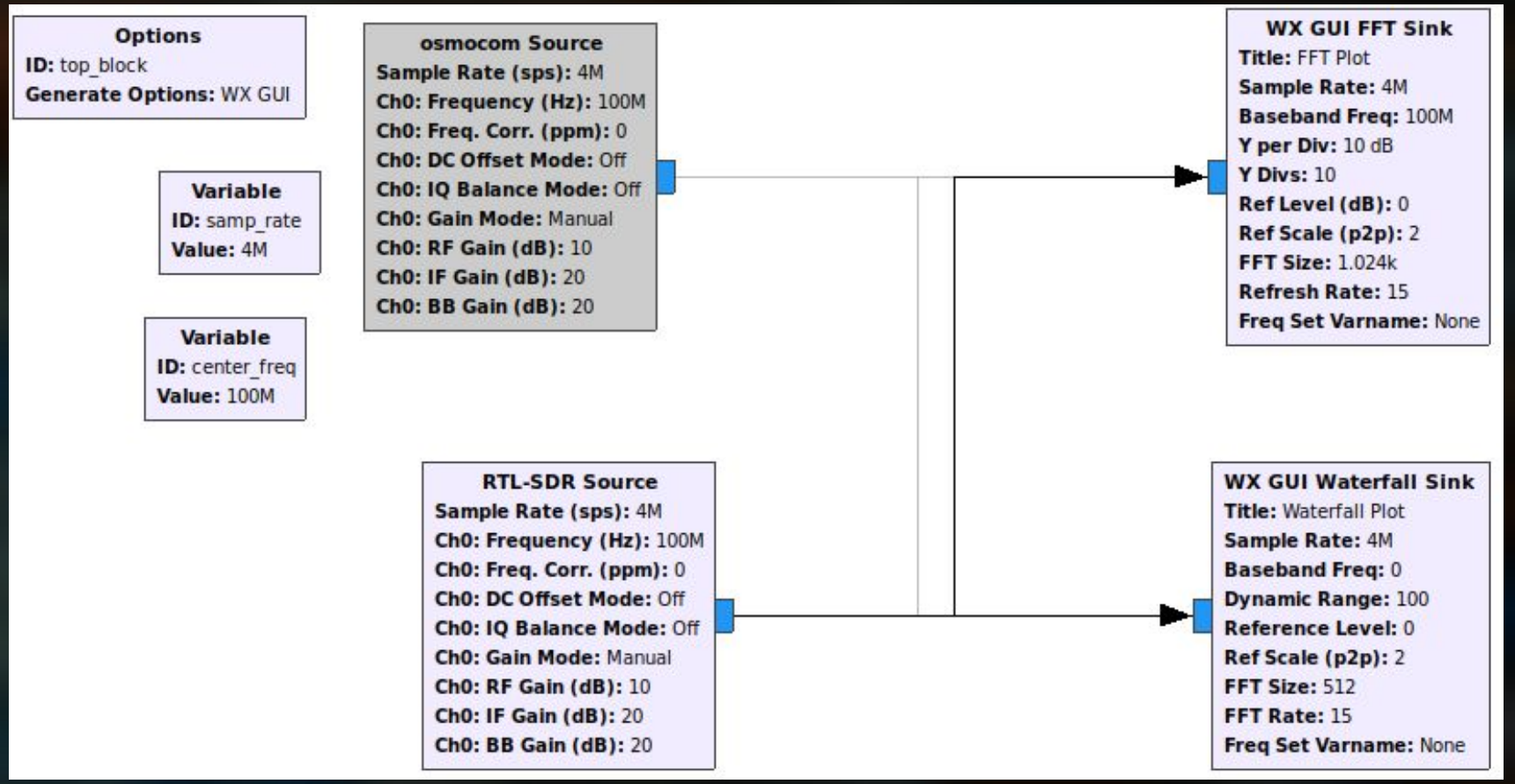
Signals vs Noise



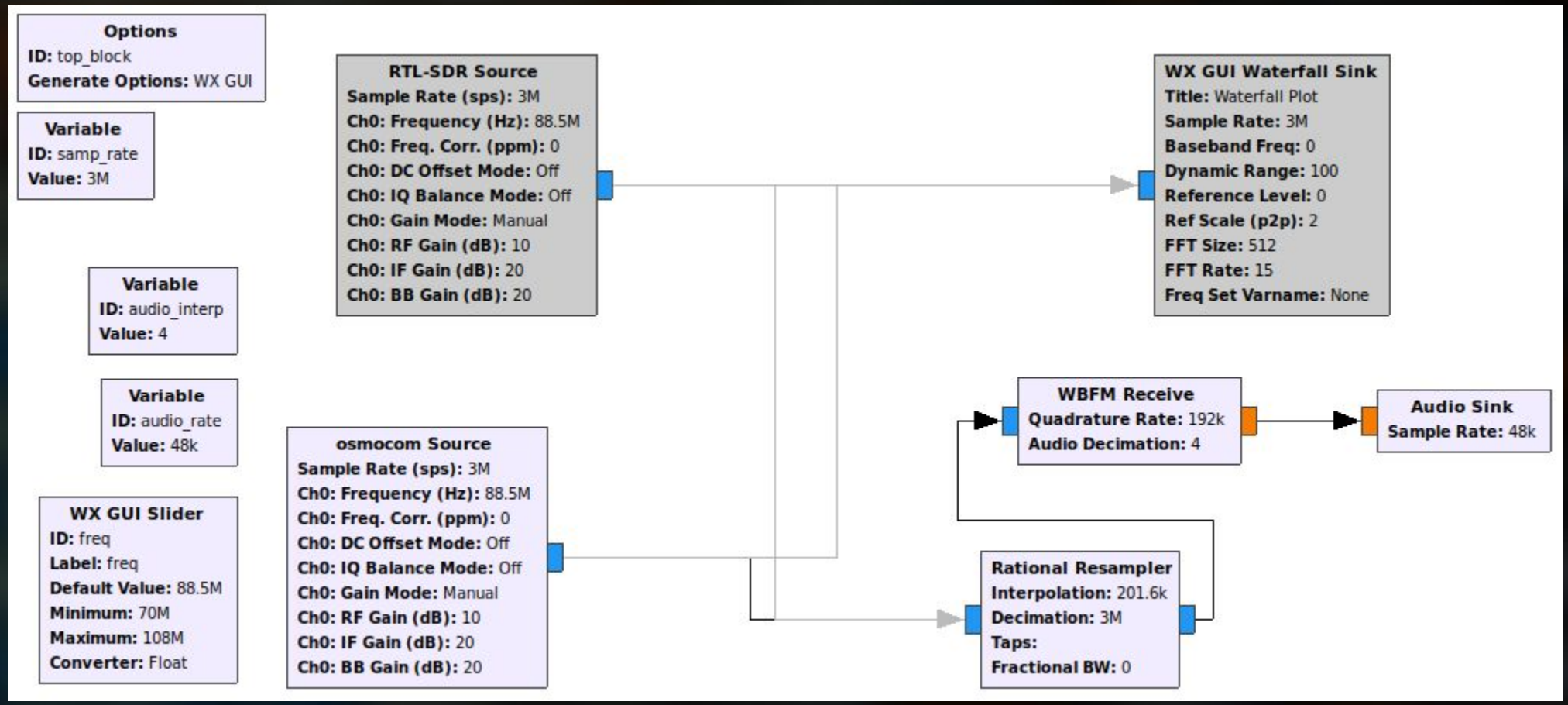
Saving EM data to files



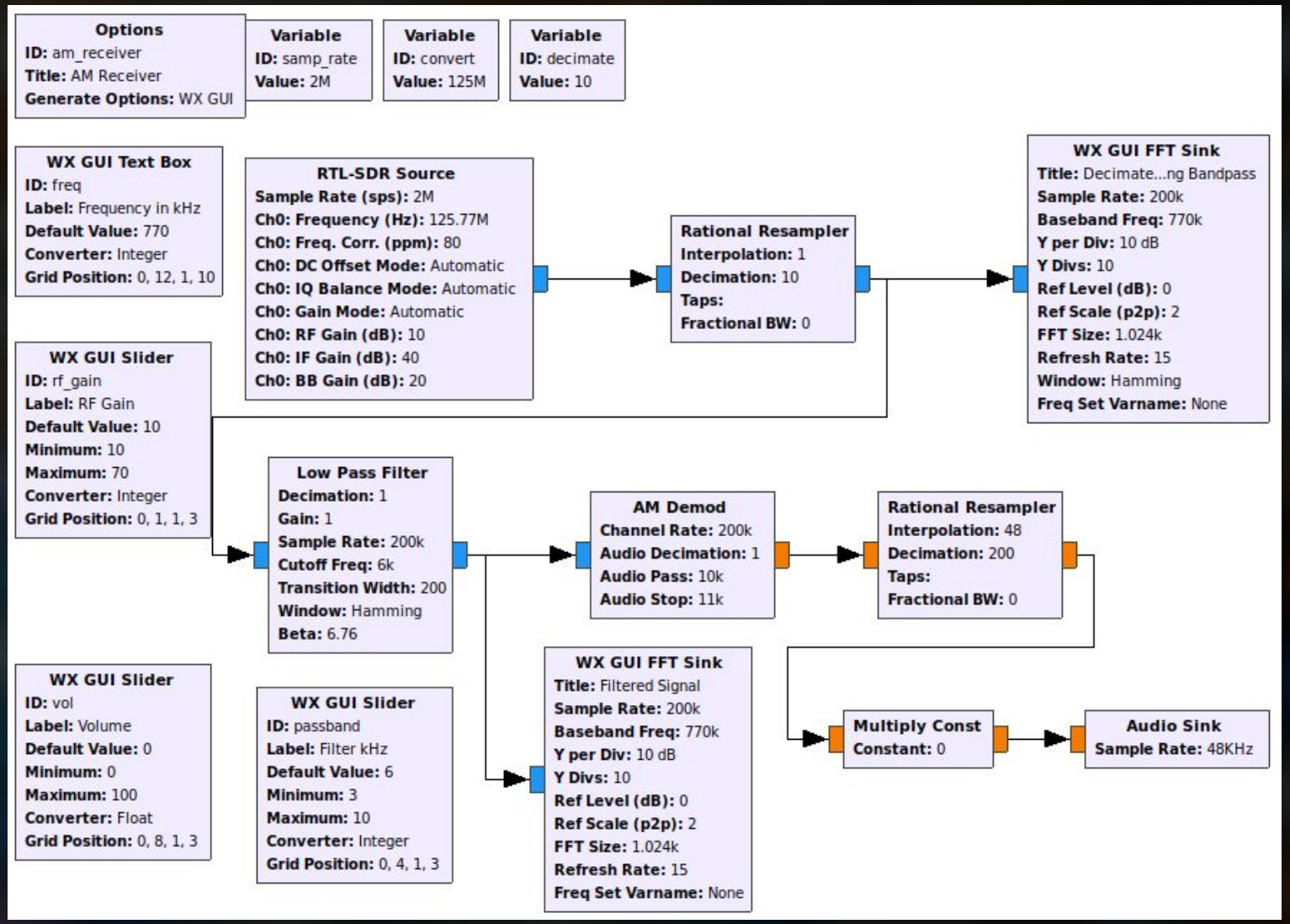
Reading data from SDR hardware



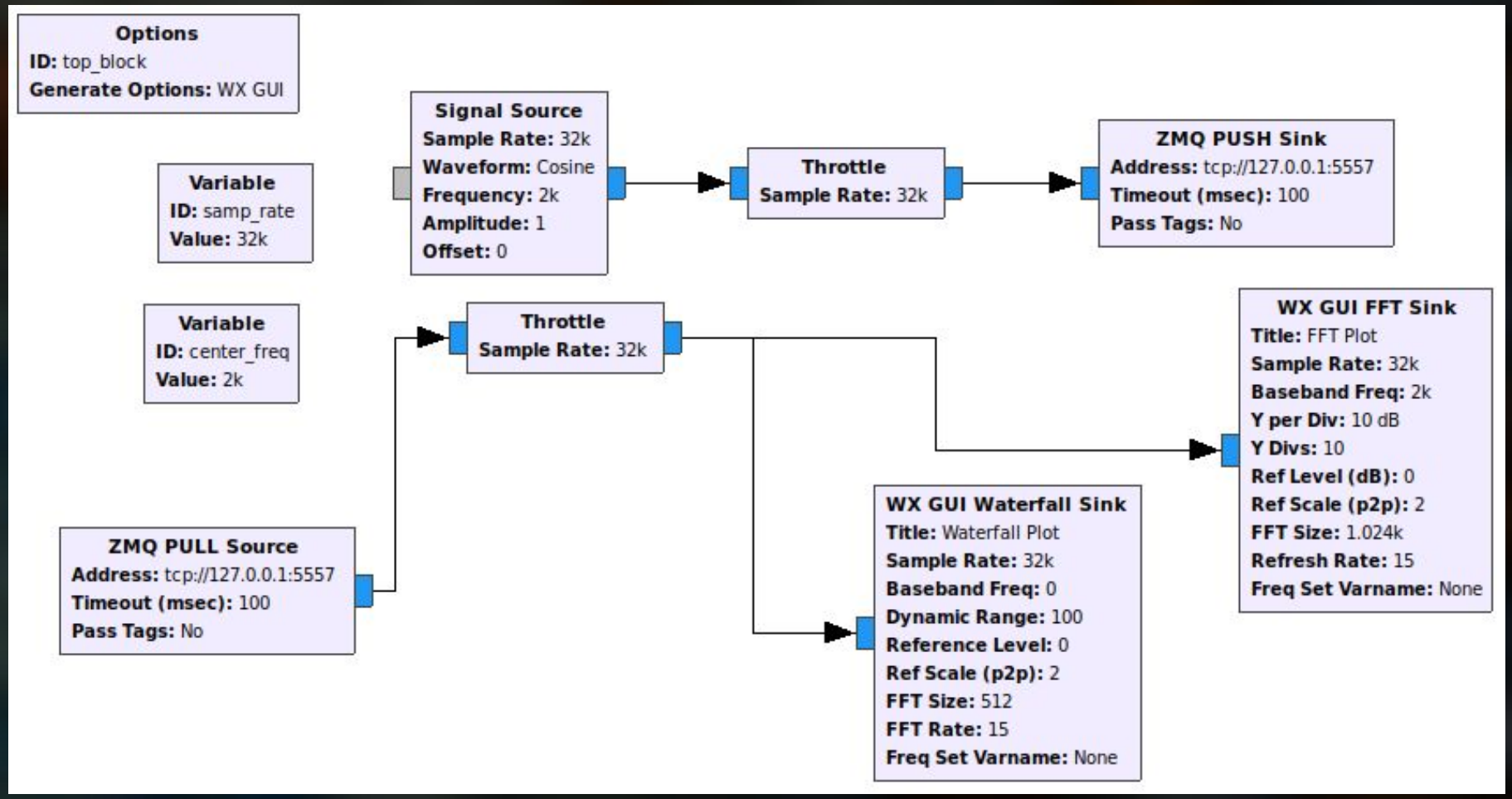
FM radio receiver



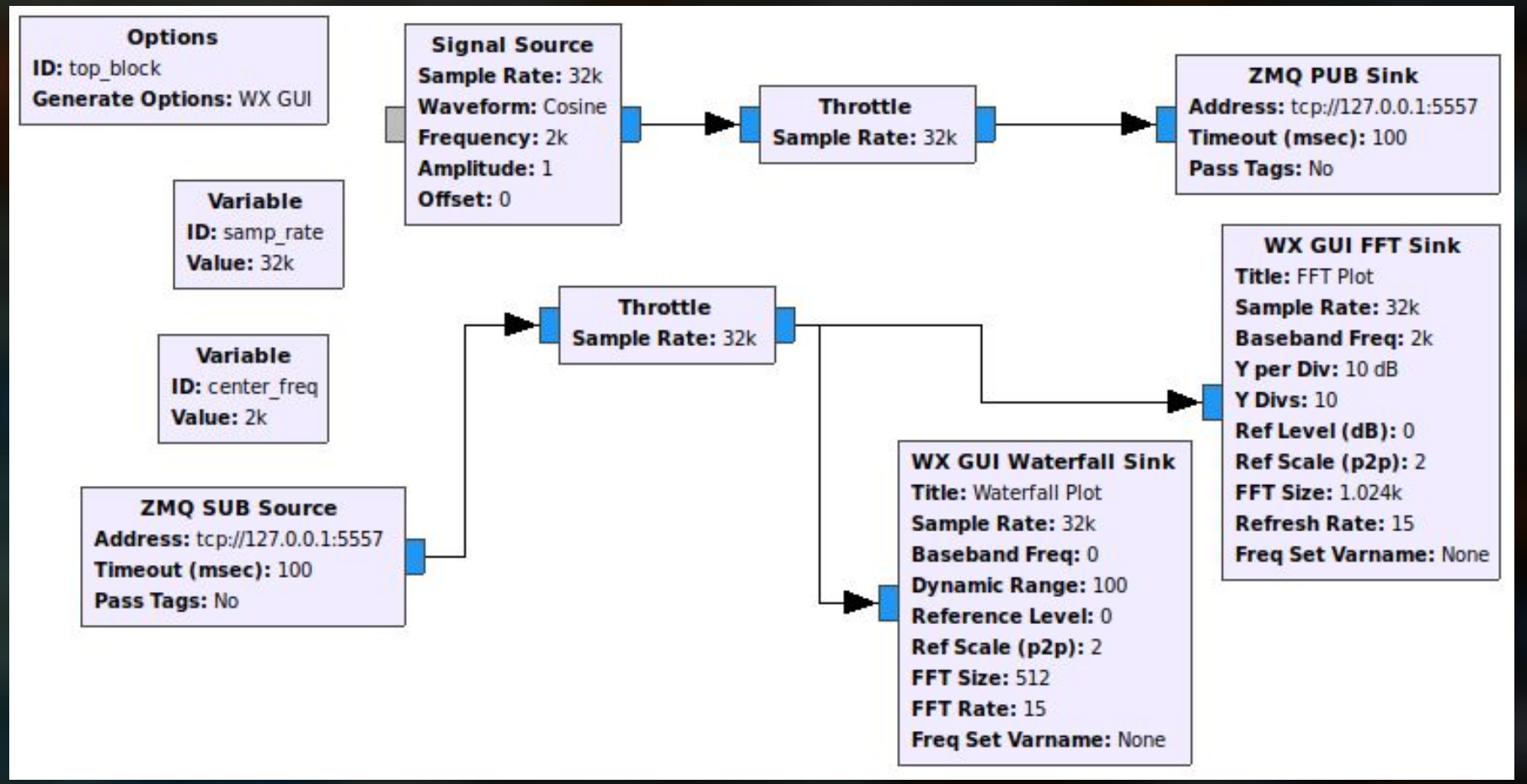
AM radio receiver



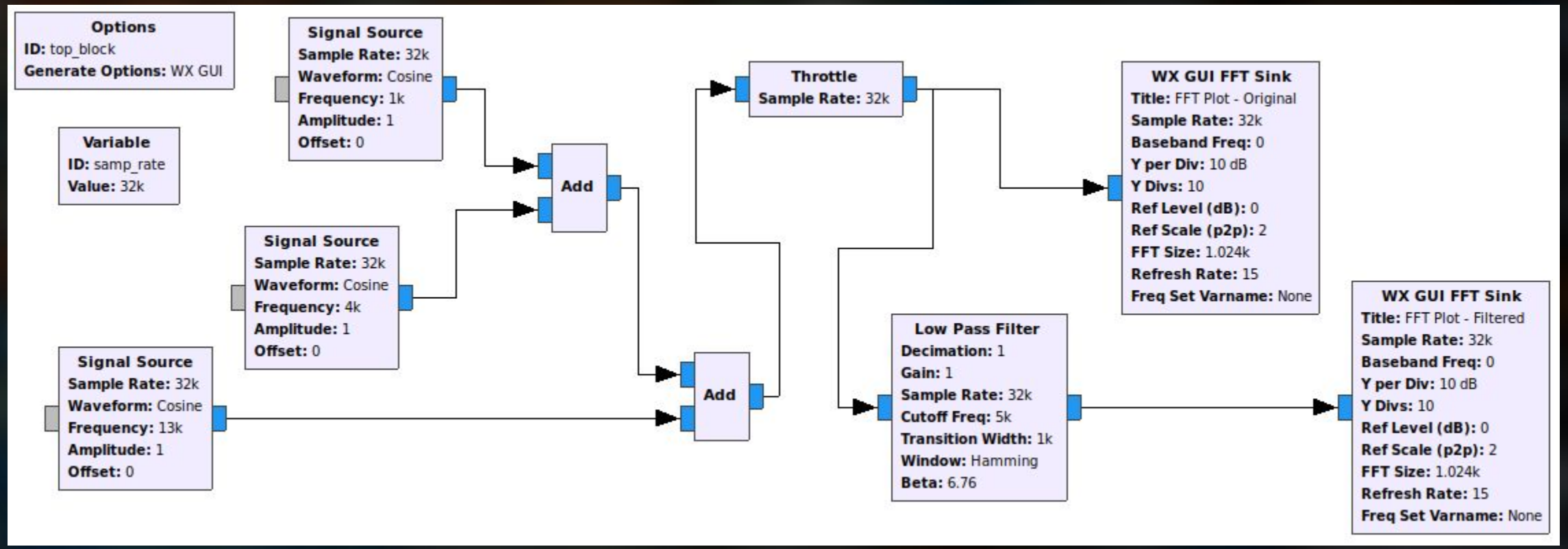
ZMQ PULL-PUSH Sockets



ZMQ PUB-SUB Sockets



Filters


Options
ID: top_block
Generate Options: WX GUI
Variable
ID: samp_rate
Value: 32k
Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0
Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 4k
Amplitude: 1
Offset: 0
Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 13k
Amplitude: 1
Offset: 0
Add
Add
Throttle
Sample Rate: 32k
Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 32k
Cutoff Freq: 5k
Transition Width: 1k
Window: Hamming
Beta: 6.76
WX GUI FFT Sink
Title: FFT Plot - Original
Sample Rate: 32k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None
WX GUI FFT Sink
Title: FFT Plot - Filtered
Sample Rate: 32k
Baseband Freq: 0
Y per Div: 10 dB
Y Divs: 10
Ref Level (dB): 0
Ref Scale (p2p): 2
FFT Size: 1.024k
Refresh Rate: 15
Freq Set Varname: None

Exercises

1. Edit the **1.exercise.grc** flowgraph to only keep the 8 kHz by filtering out other signals with the help of suitable GRC blocks.

2. Create a GRC flowgraph called **2.exercise.grc** that reads the two given wav files (**siren-sound.wav** and **bikehorn-sound.wav**) and generate a new wav file called **mixed-signal.wav**. When this resulting file is played, we should hear the sounds of the two original files as the same time.

Software Defined Radio Programming (45 mins)

UCD Forensics and
Security Research Group

Reading data through ZMQ to Python

- ▶️ See Jupyter-Notebook 1

Reading data from SDR hardware

- ▶️ See Jupyter-Notebook 2

Pre-processing EM data

- ▶️ See Jupyter-Notebook 3

Machine learning for EM data classification

- ▶️ See Jupyter-Notebook 4

Arduino program classification

- ▶️ See Jupyter-Notebook 5

Module Development for EMvidence Framework (45 mins)

UCD Forensics and Security Research Group

ASANKA.SAYAKKARA@UCDCONNECT.IE

WWW.FORENSICSANDSECURITY.COM

@FORSECRESEARCH

UCD Forensics and
Security Research Group