Fan-Tastic RFID Thief

Revamping an old weaponised RFID reader tool

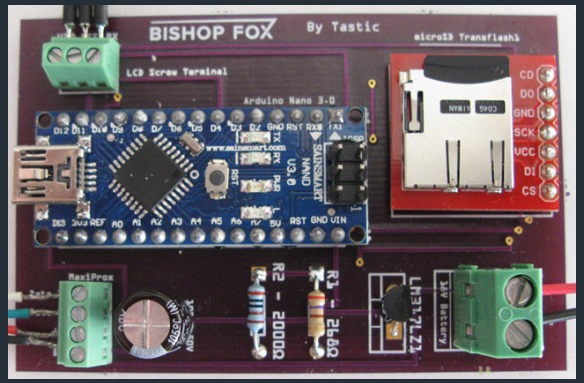
aura INFORMATION SECURITY | POWERED BY KORDIA

Whoami

- Daniel Underhay aka phish (@dunderhay on X / twitter)
- Principal Security Consultant at Aura Information Security (based in our Melbourne Office)
- Father

What is the Tastic RFID Thief

- Off the shelf card reader (HID MaxiProx 5375)
- Custom PCB + Arduino + SD Card

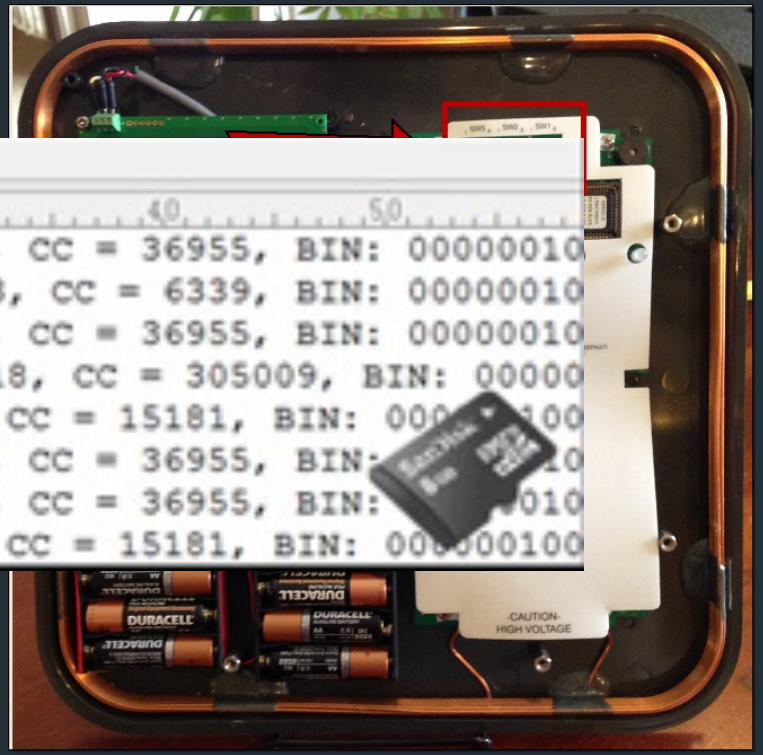


The Original Tastic RFID Thief

- Created by Francis Brown from
- First
- Used acces sever

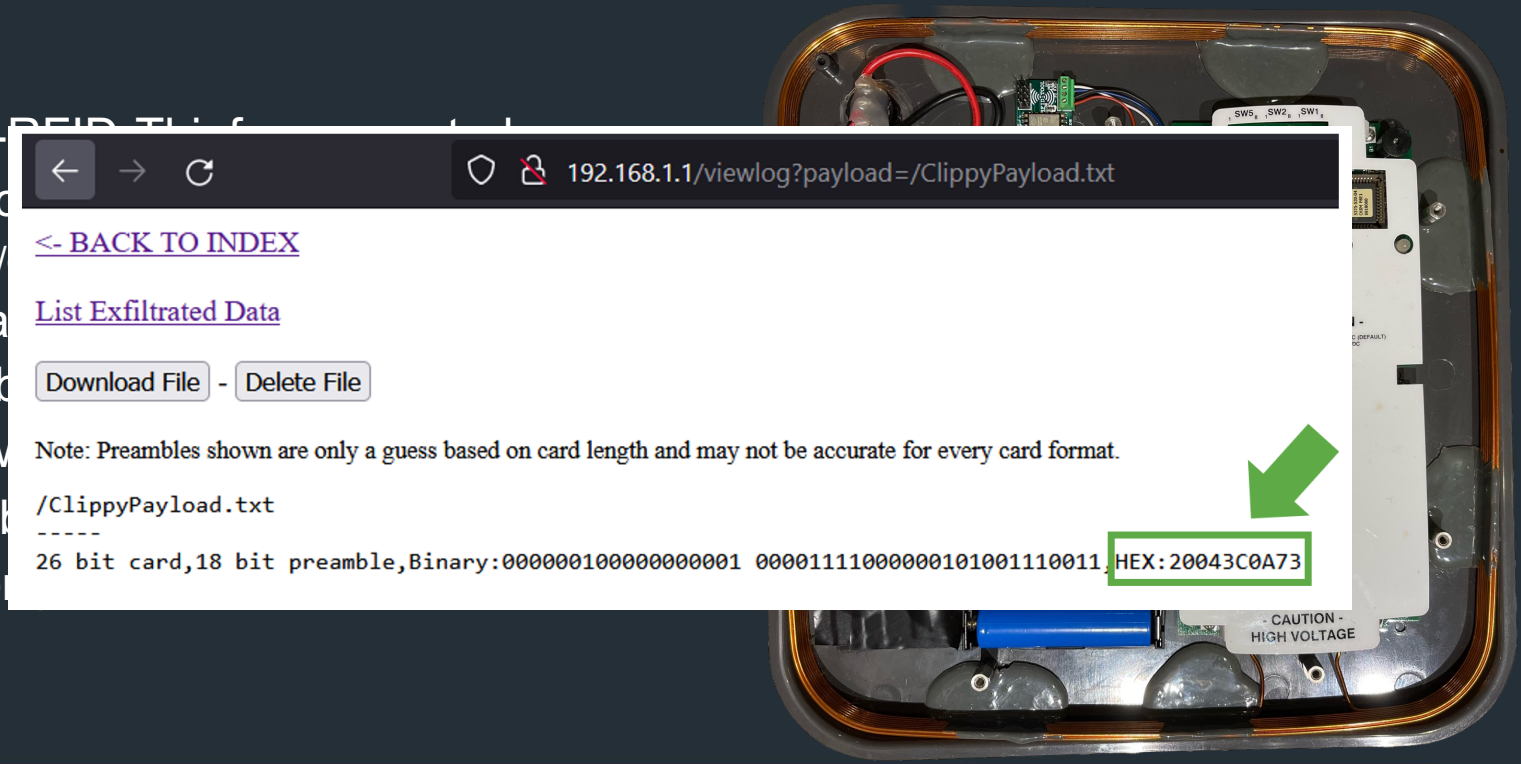
CARDS.TXT

```
34 bit card: 2400af20b6, FC = 87, CC = 36955, BIN: 00000010
26 bit card: 2006e23186, FC = 113, CC = 6339, BIN: 00000010
34 bit card: 2400af20b6, FC = 87, CC = 36955, BIN: 00000010
35 bit card: 2f85c94ee3, FC = 3118, CC = 305009, BIN: 00000
26 bit card: 200610769a, FC = 8, CC = 15181, BIN: 000
34 bit card: 2400af20b6, FC = 87, CC = 36955, BIN:
34 bit card: 2400af20b6, FC = 87, CC = 36955, BIN:
26 bit card: 200610769a, FC = 8, CC = 15181, BIN: 000000100
```



Tastic RFID Thief + ESP-RFID-Thief/Tool

- ESP-
  by Co
  https://
- Upda
  Has b
- Has v
- This
  for lo



Wiegotcha

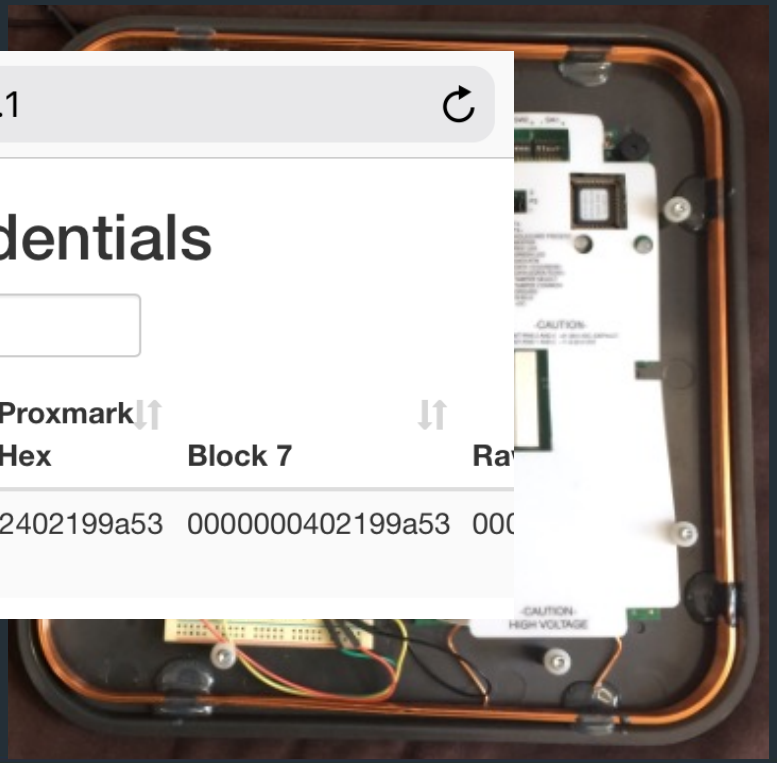
- Created
- Uses a R
- Has buil
- Has a pr
- This buil
  for longe

192.168.150.1

Wiegotcha Stolen Credentials

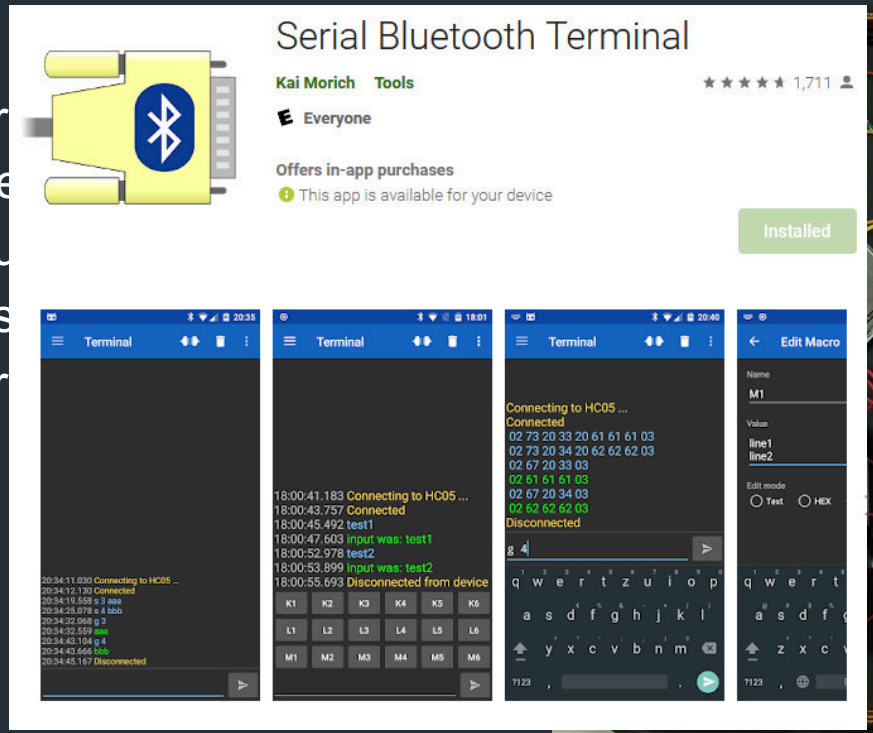
Search:

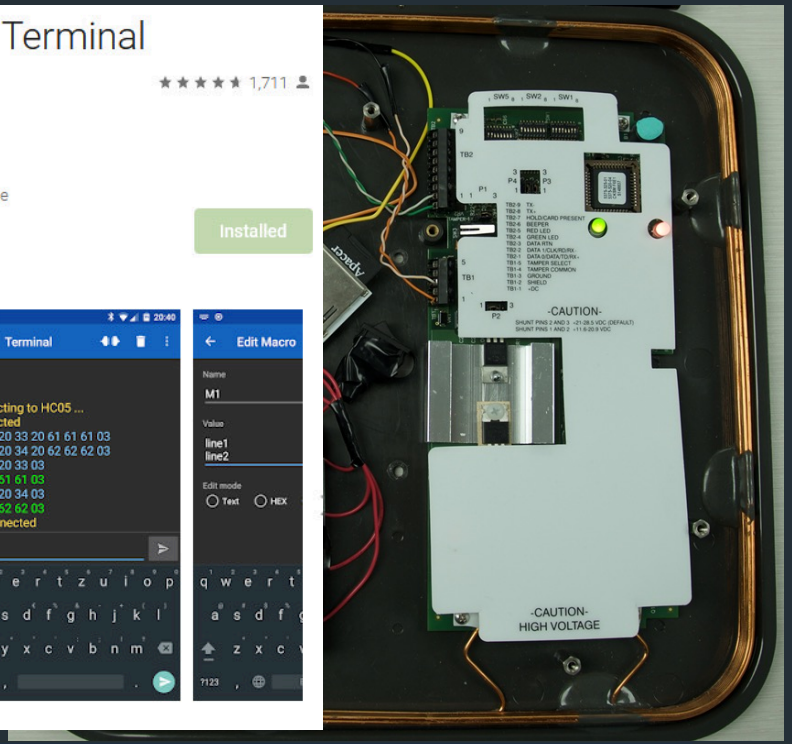
Time	Bit Length	Facility Code	ID Number	Proxmark Hex	Block 7	Ra
11/03/2016 17:18	34	268	52521	2402199a53	0000000402199a53	000



Tastic RFID Thief + HC06

- Created at Aur
- Modified by me
  which adds Blu
- This build uses
  for longer oper





LF vs HF in 20 seconds

Low Frequency RFID
125 kHz

Primitive protocol

No security at all*

High Frequency RFID (NFC)
13.56 MHz

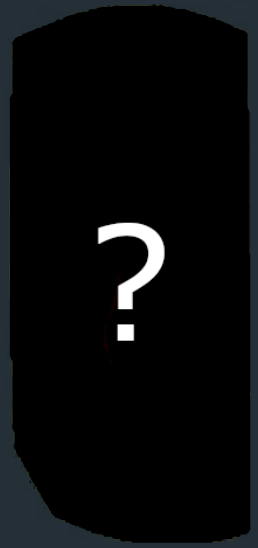
Advanced protocols

Can be secure*

- https://blog.flipper.net/rfid/

125kHz is Still Common


HID


?

Planning Upgrades for the Tastic RFID

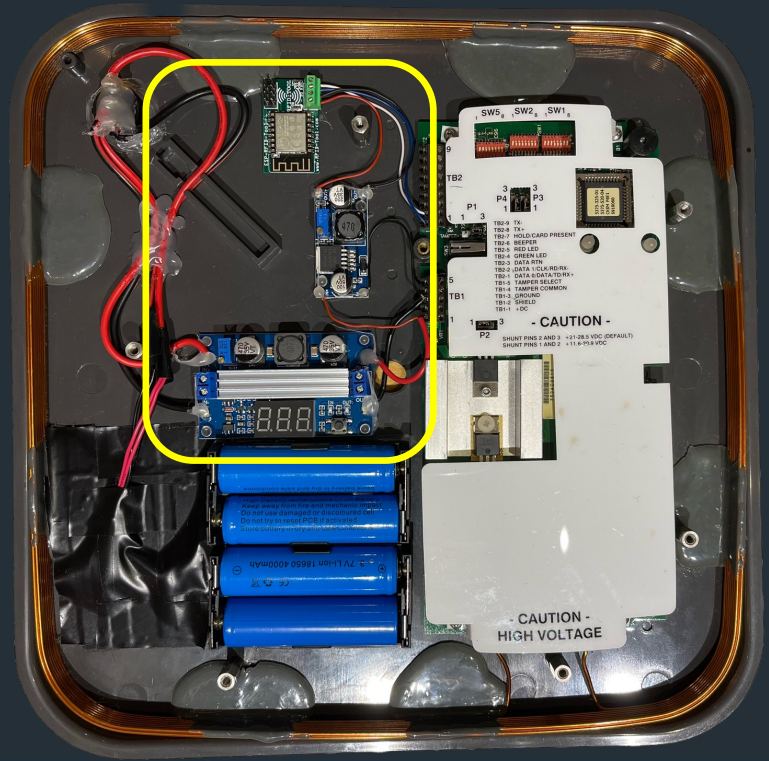
- Relocate noisy RF components
- Update batteries
- Update microcontroller
- Create new custom PCB
- Update web application
- Add some new capability

Radio Frequency (RF) Noise

Types of (RF) Noise

- Noise in Radio Frequency (RF) systems can generally be regarded as any RF energy that is not the desired signal
- Two terms commonly used to describe RF noise:
  - Electromagnetic Interference (EMI) = random, broadband noise
  - Radio Frequency Interference (RFI) = narrowband noise broadcast at specific frequencies

Relocating Noisy RF Components



Batteries

1.5v

1700 mAh to
2850 mAh

AA

4.2v

2700 mAh to
3600 mAh

18650

4.2v

3000 mAh to
5000 mAh

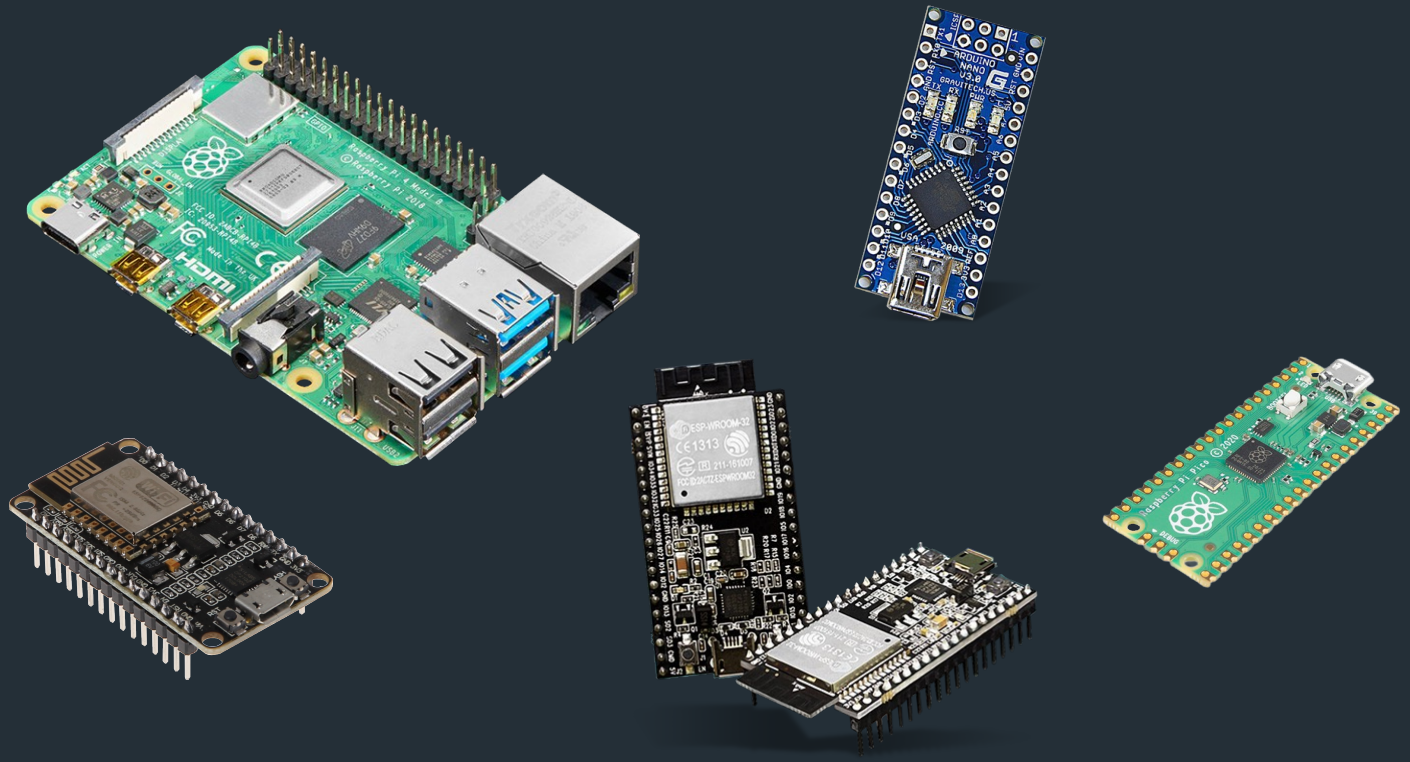
21700

Batteries

2 x 18650 for Microcontroller

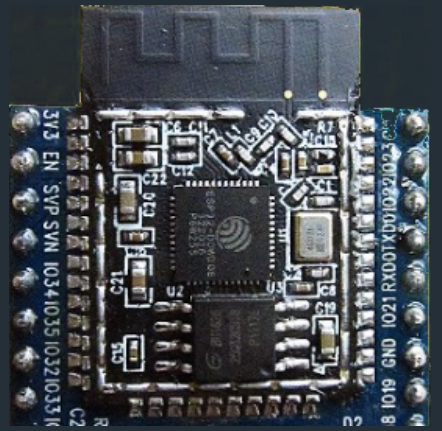
5 x 21700 for the reader

Microcontrollers

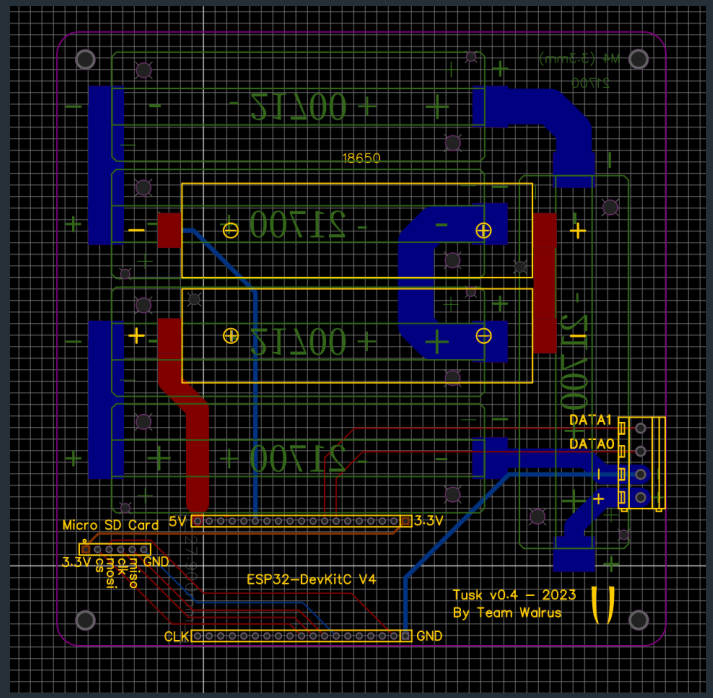


ESP32-WROOM-32D

- 32-bit Dual-Core Processor (Tensillica Xtensa LX6)
- Built-in Wi-Fi & Bluetooth
- Low Power
- Low Cost
- 4 MB SPI Flash Memory
- 448 KiB ROM
- 520 KiB SRAM

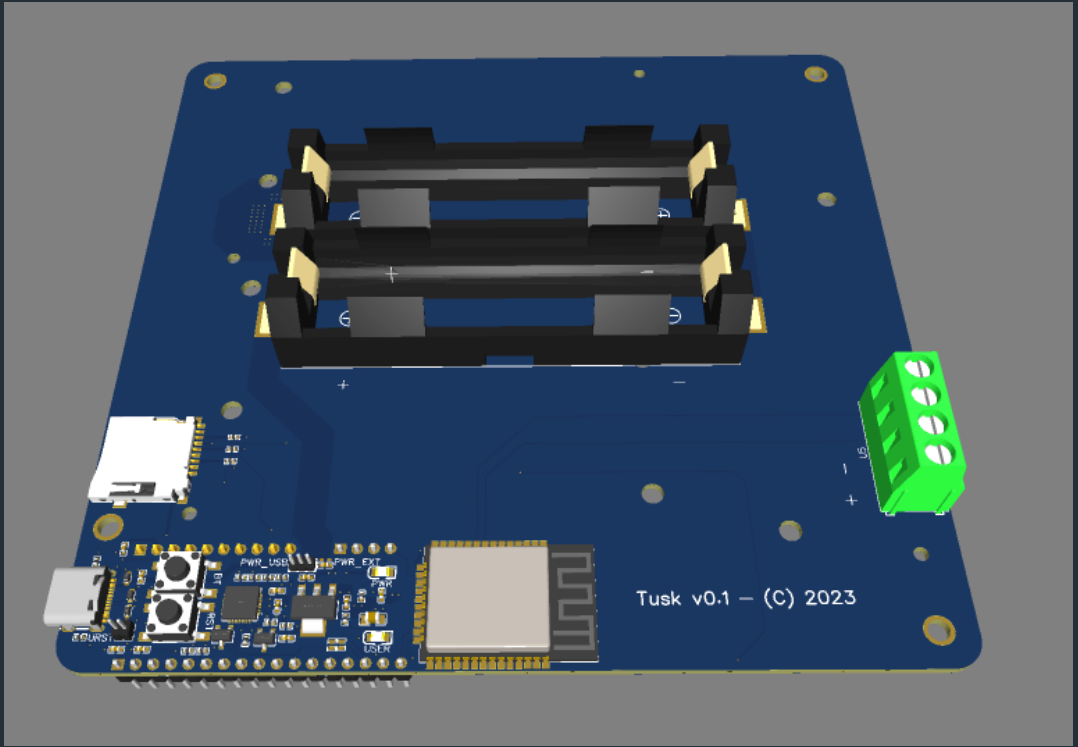


Design a new PCB



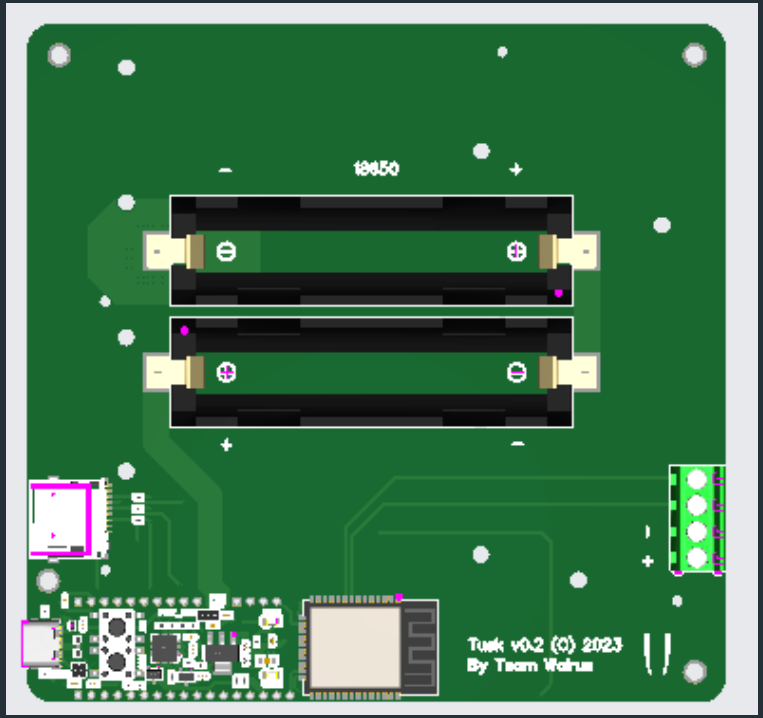
First Attempt

- ESP32
- SD Card slot
- 2 x 18650’s for ESP32
- 5 x 21700’s for reader
- Terminal block to connect reader



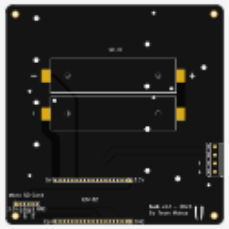
First Attempt

Total Price:	$154.45
Weight	1.42kg
Product Description	



Economy of Scale

Second Attempt



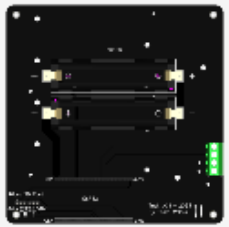
PCB Prototype

Order #: Y7-2682123A

Build Time: 2 days

5 pcs $11.10

Product Details



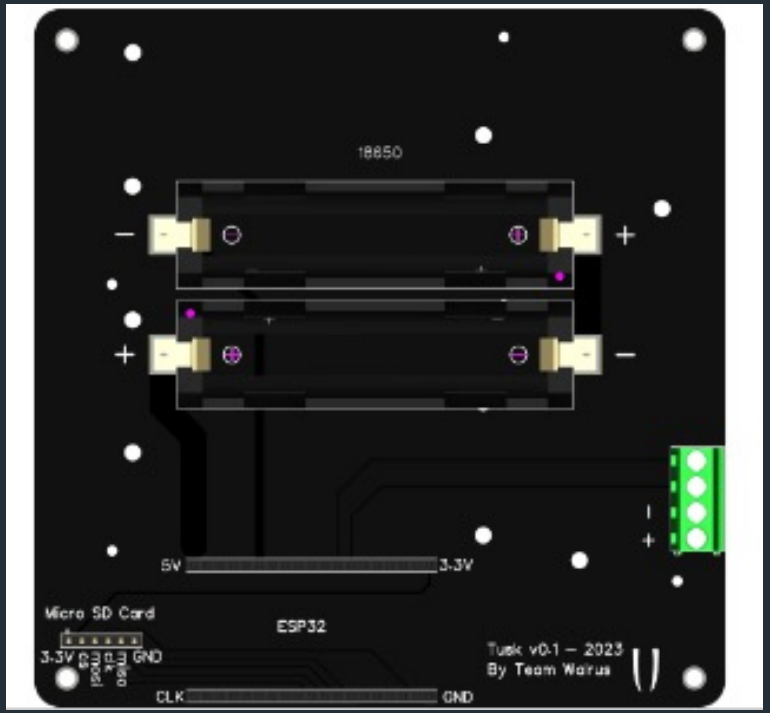
Economic PCBA

Order #: SMT023021711…

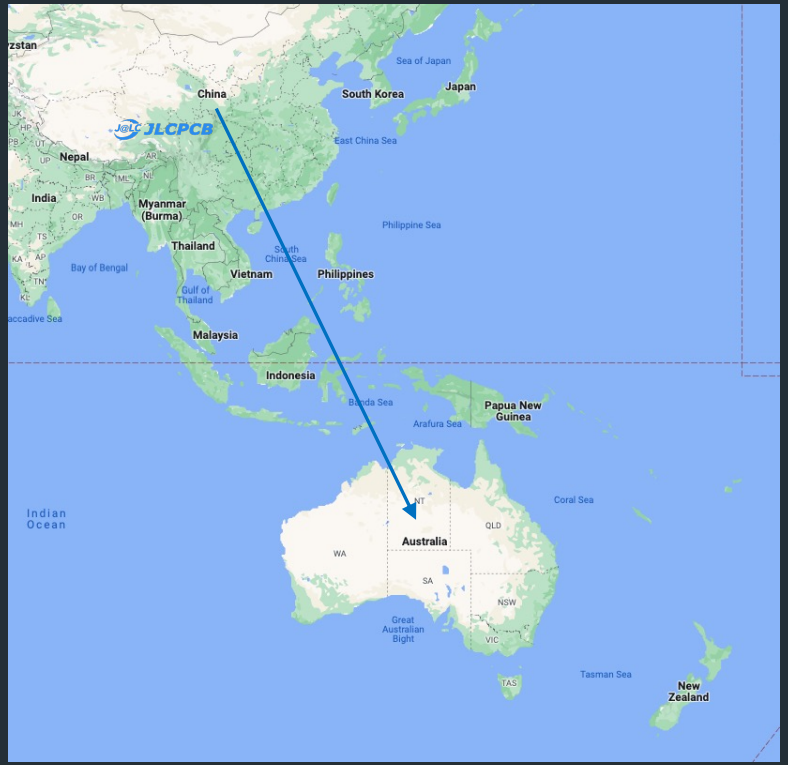
Build Time: 2-3 days

5 pcs $34.53

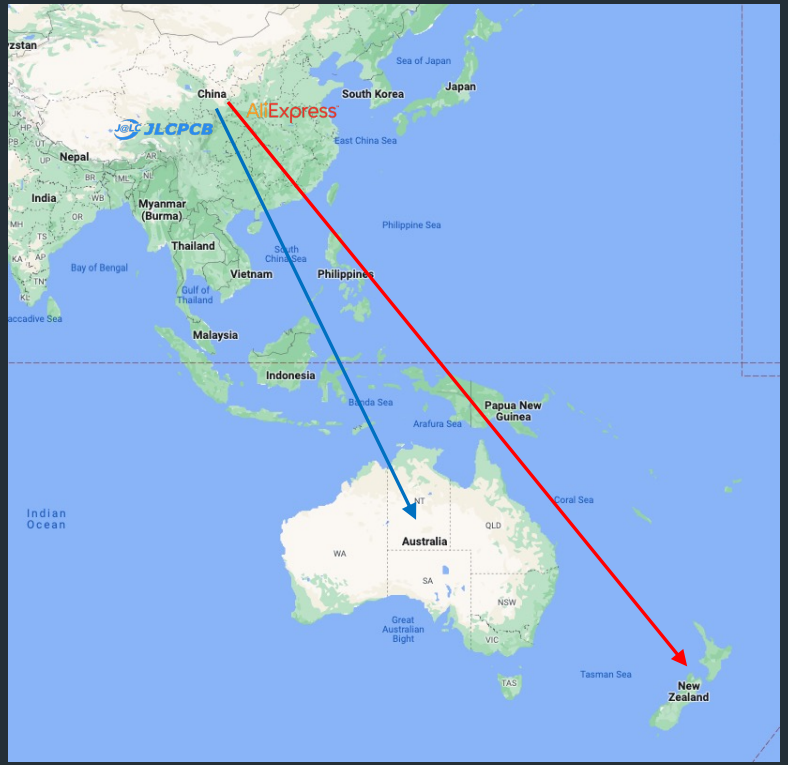
Product Details



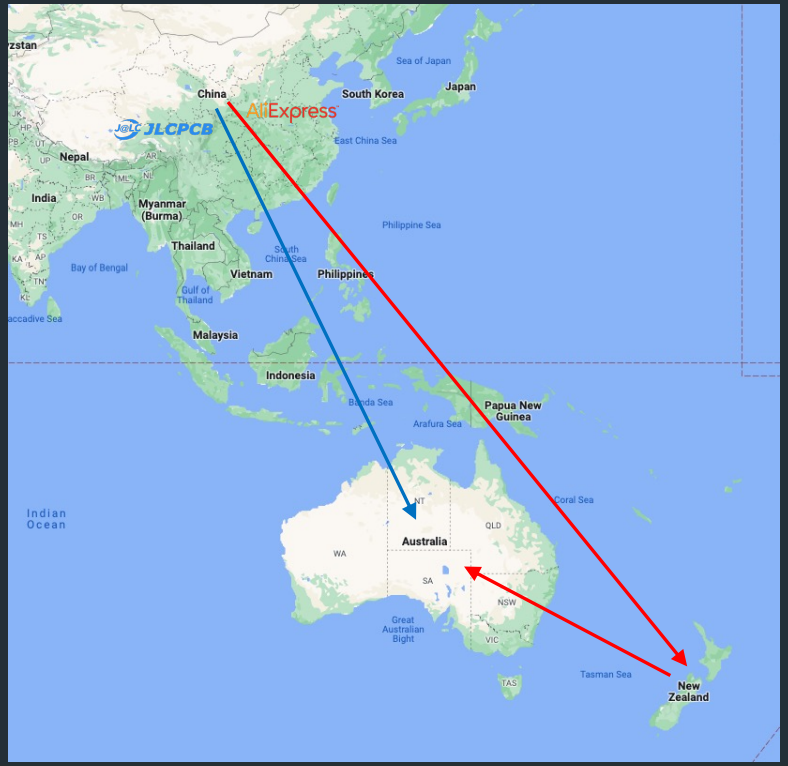
Ordering Custom PCB



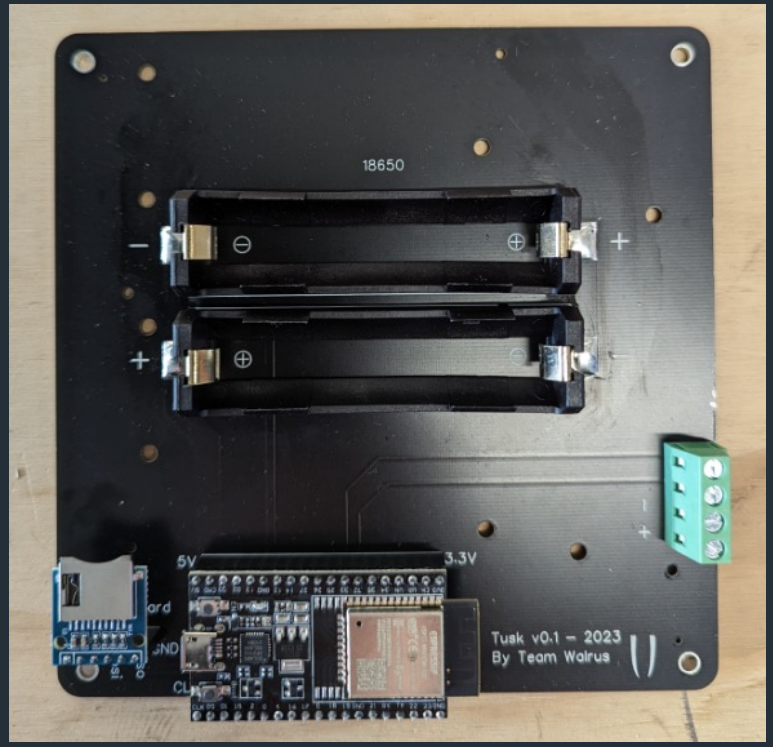
Ordering ESP32 and Other Parts 🤦

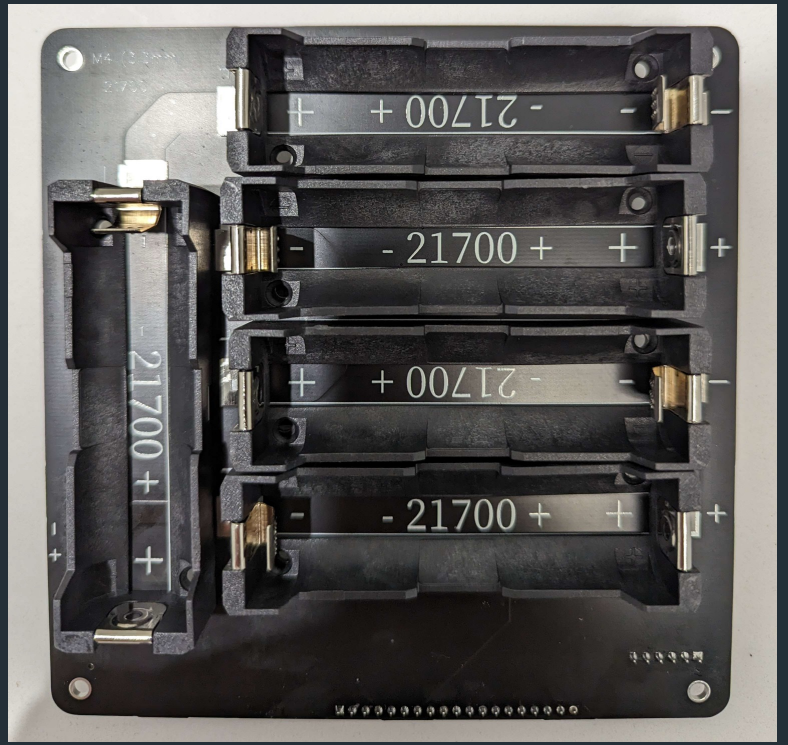


Redirect Parts

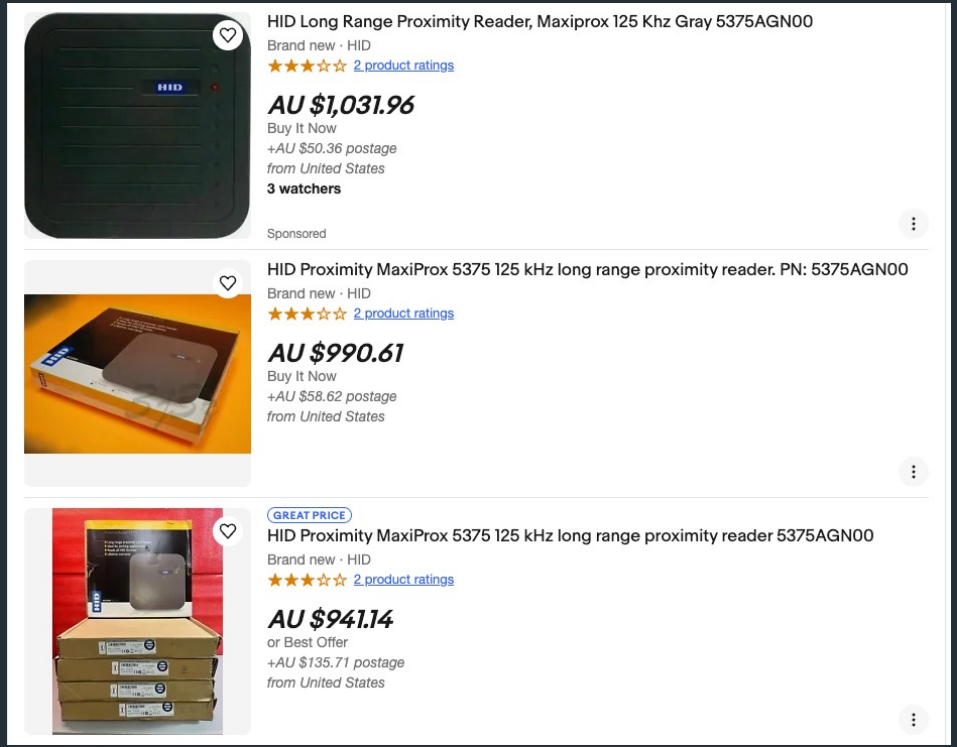


Assembled PCB (Tusk)

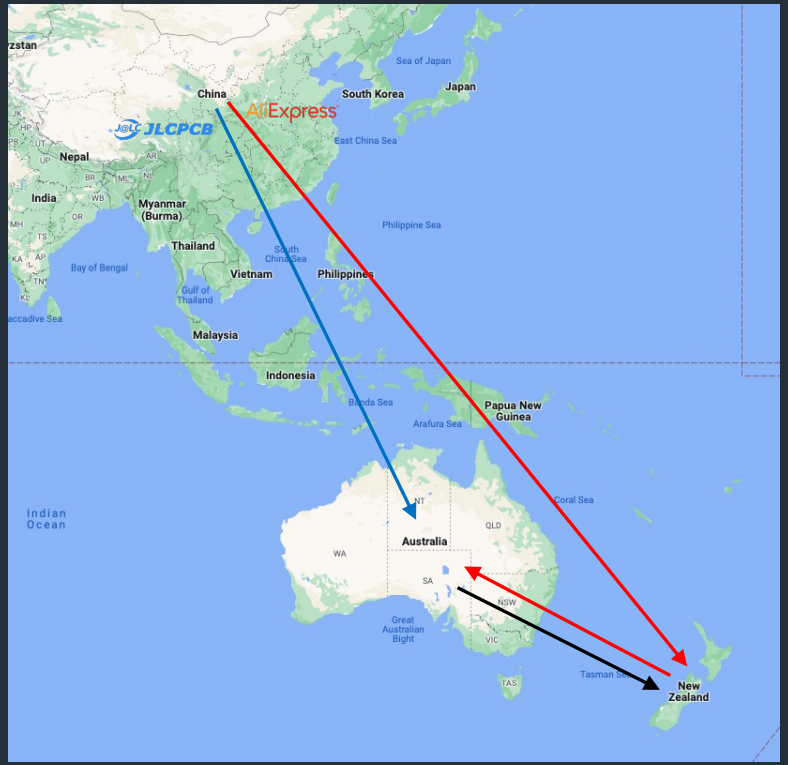




Testing?

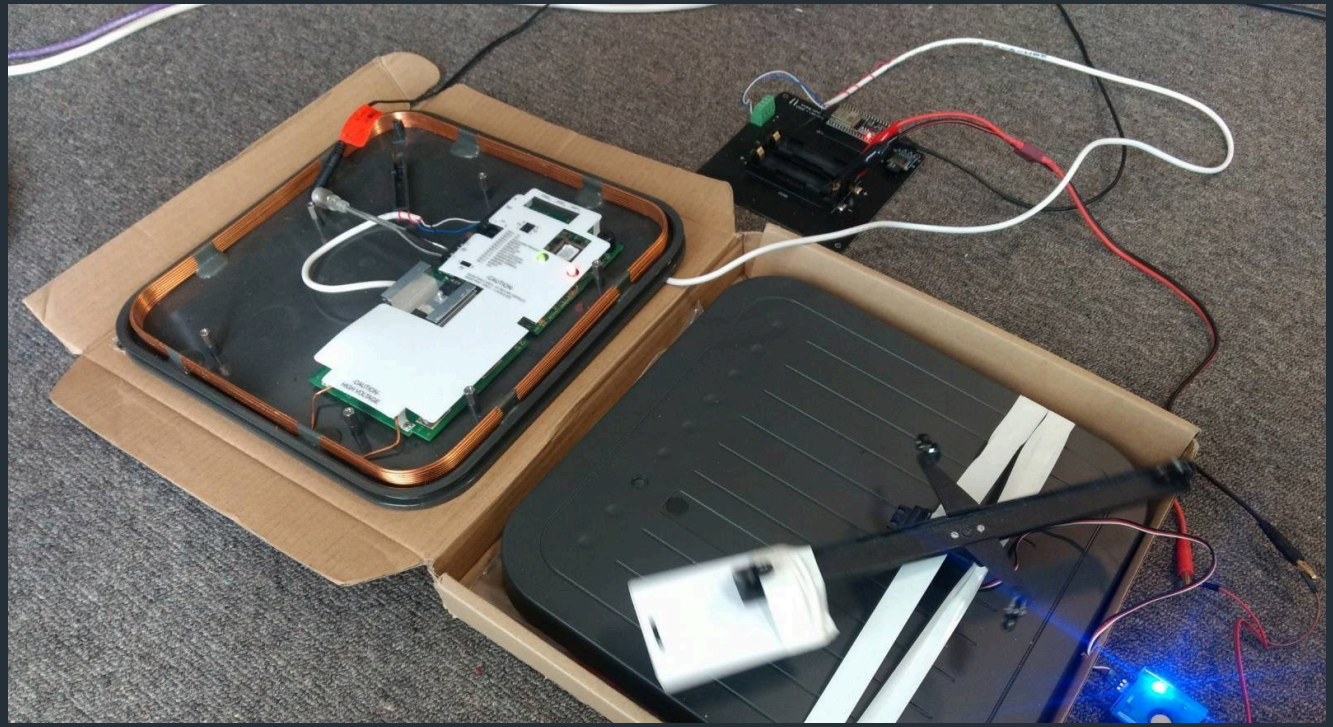

HID Long Range Proximity Reader, Maxiprox 125 Khz Gray 5375AGN00
Brand new · HID
2 product ratings
AU $1,031.96
Buy It Now
+AU $50.36 postage
from United States
3 watchers
Sponsored
HID Proximity MaxiProx 5375 125 kHz long range proximity reader. PN: 5375AGN00
Brand new · HID
2 product ratings
AU $990.61
Buy It Now
+AU $58.62 postage
from United States
GREAT PRICE
HID Proximity MaxiProx 5375 125 kHz long range proximity reader 5375AGN00
Brand new · HID
2 product ratings
AU $941.14
or Best Offer
+AU $135.71 postage
from United States

Back to New Zealand



Remote Testing Rig #1

Remote Testing Rig #1



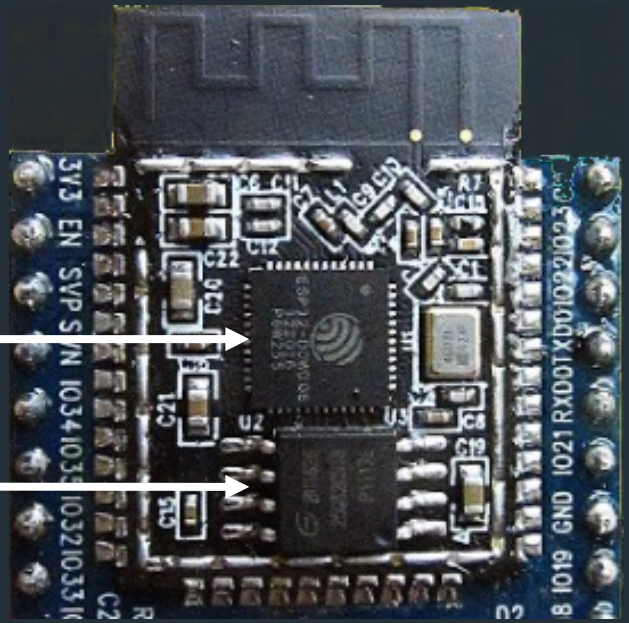
It Works and... First Bug! 🎉 🪲

Updating the Web Application

ESP32-WROOM-32D Overview

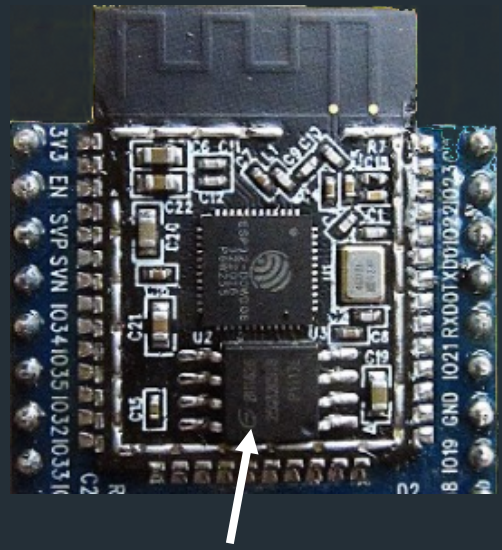
ESP32-D0WD SoC

4MB Flash memory



React Frameworks

Package	Build Size
Gatsby	1.01 MB
Next.js	6.34 MB
Create React App	504 KiB
Vite	136 KiB



4MB Flash Memory

https://youtu.be/R9n32nxrzug

Software Overview

C++ Backend
Web Server / API
Decode card data

React Frontend
Web App

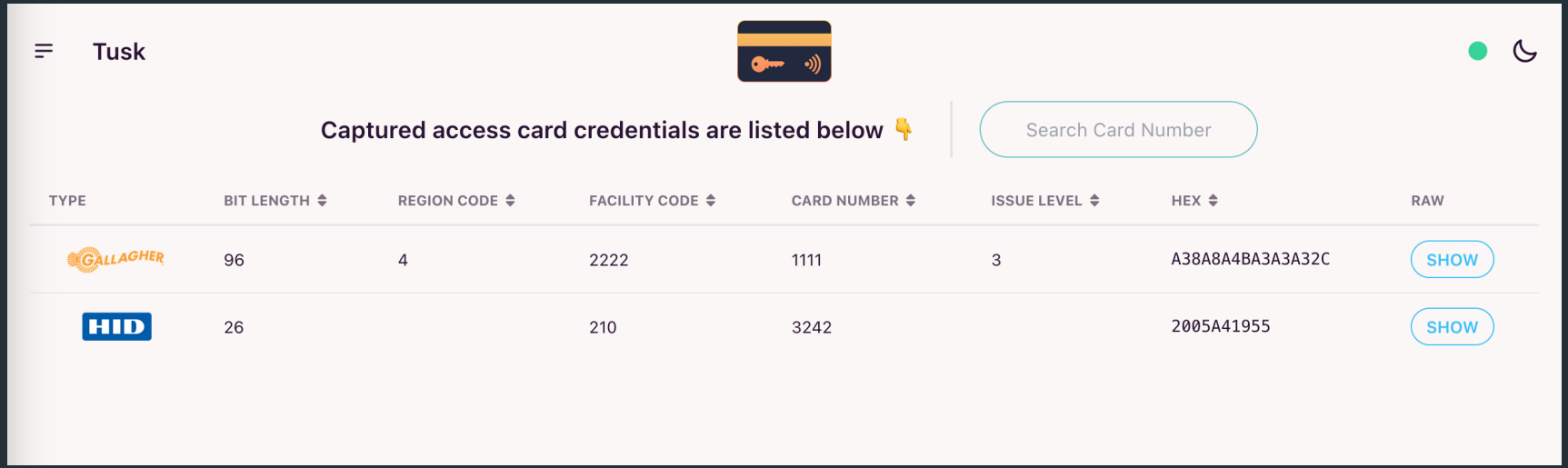
Tailwind CSS

daisyUI

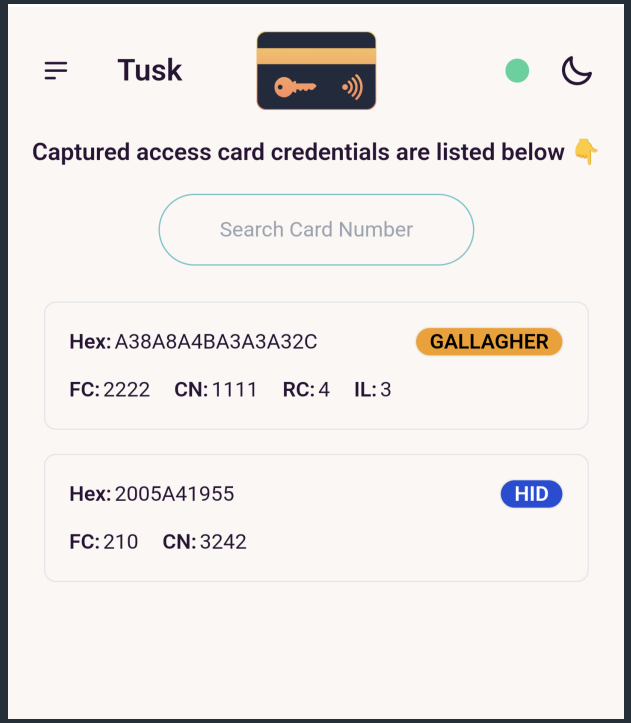
Plugin for TW

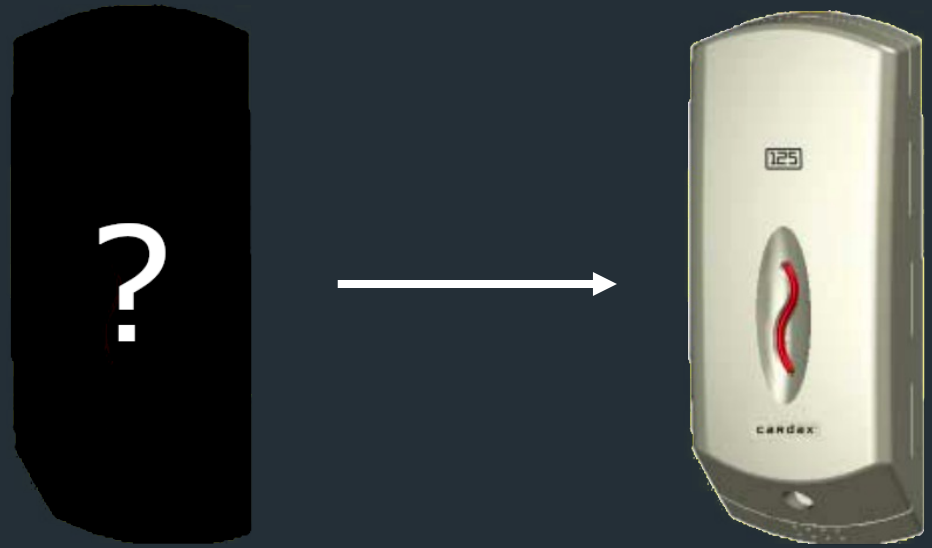
Vite

The New* Web User Interface



Mobile Friendly! 🦾




?
125
cardax

Add New Capability

- Decode Gallagher 125kHz cards

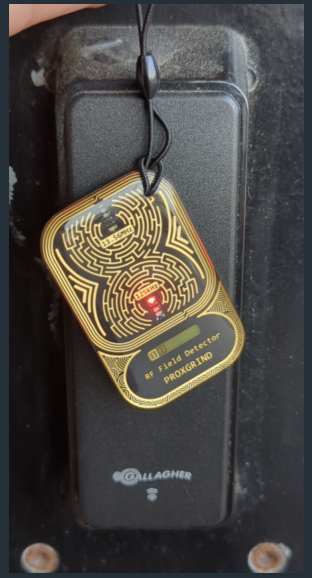
Cardax readers (125kHz only)

T-Series readers (HF + LF)

Backstory

T-Series Dual Mode

High Frequency
(13.56 MHz)



Low Frequency
(125kHz)

[redacted]

+

GALLAGHER

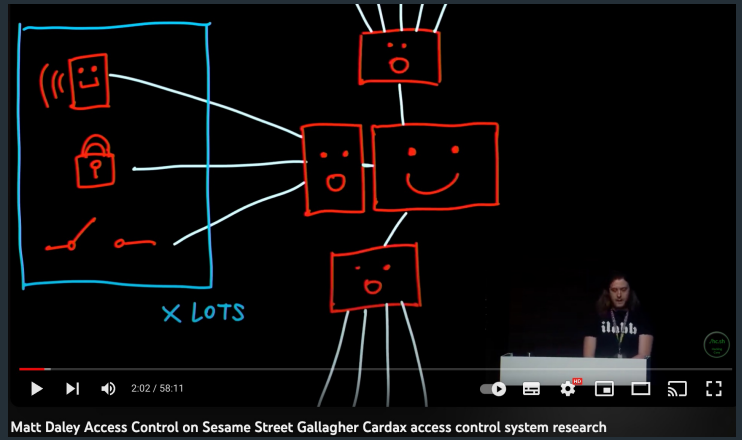
cardax

=

profit

Forward Engineering

- Matt Daley: Access Control on Sesame Street
- https://www.youtube.com/watch?v=MMB6x_QTz3E
- https://github.com/megabug/gallagher-research



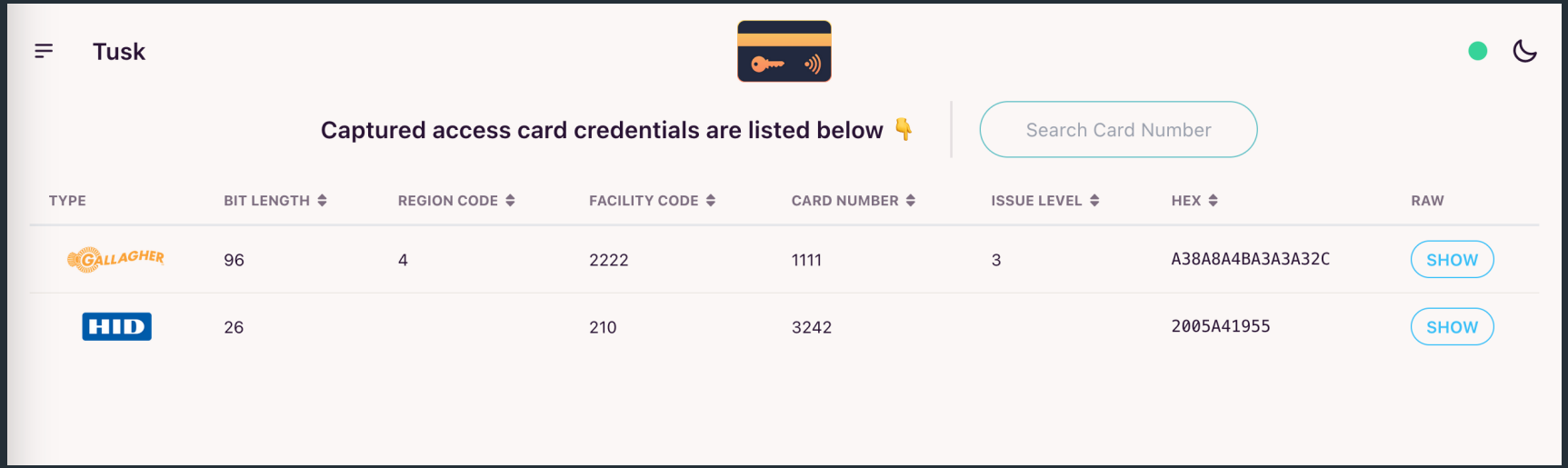
Low-frequency (125kHz) Cardax Card Format

- Manchester encoded data at a RF/32 clock speed (125kHz / 32 ~= 3.9kHz)
- First 16 bits is a fixed sequence: 0111111111101010
- Followed by the 8-byte cardholder credential data
- Each byte of cardholder credential data is followed by the inverse of the least significant bit
- The cardholder credential data consists of a tuple of items (Region Code, Facility Code, Card Number, Issue Level), which is obfuscated into an 8-byte format

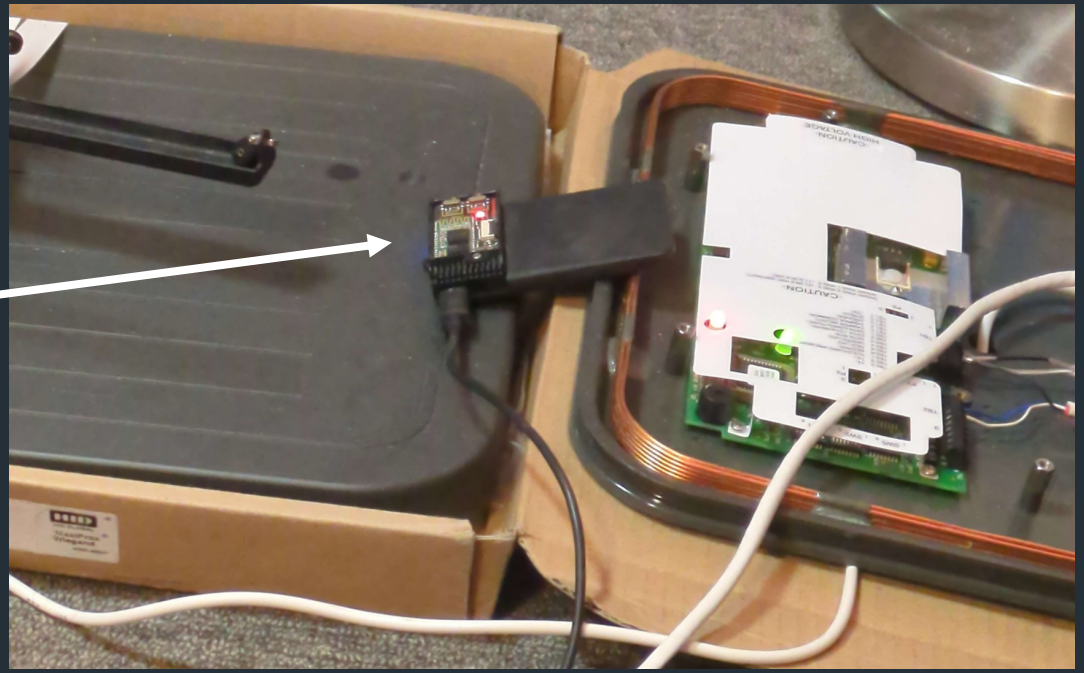
Faking a Reader


```
01111111111010101010001101000101011000101010100101101010001101010001101010001100010110
00111010010
```

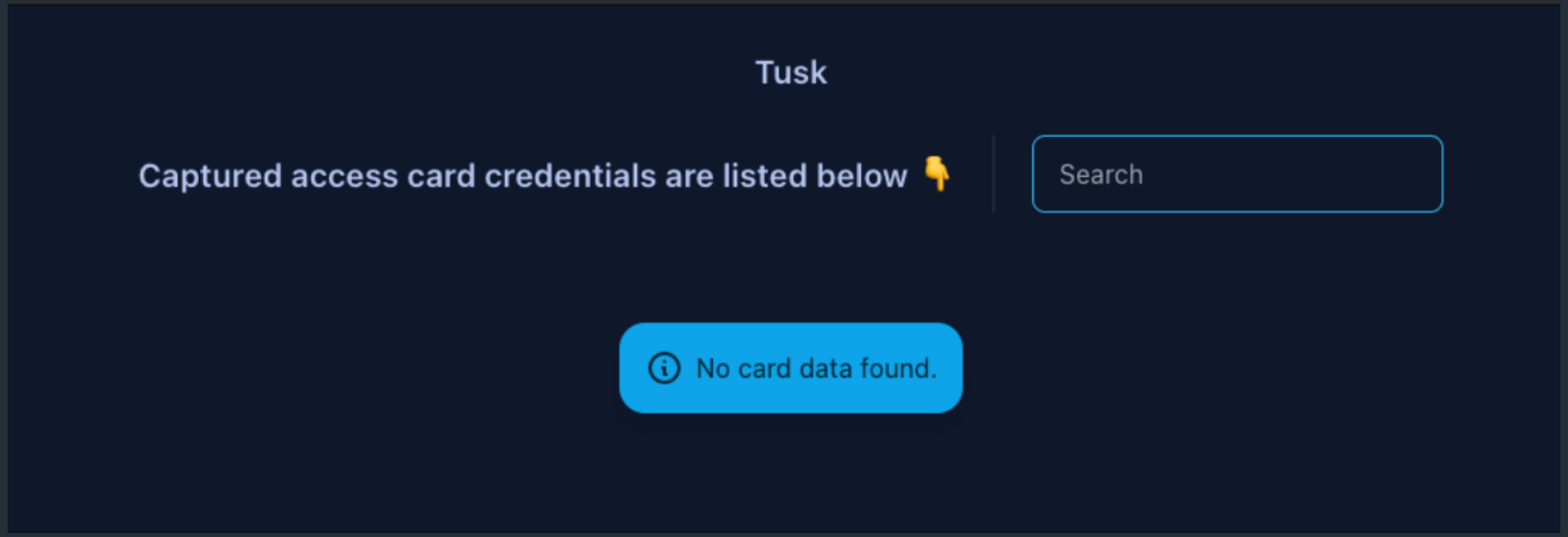
It works… I think?



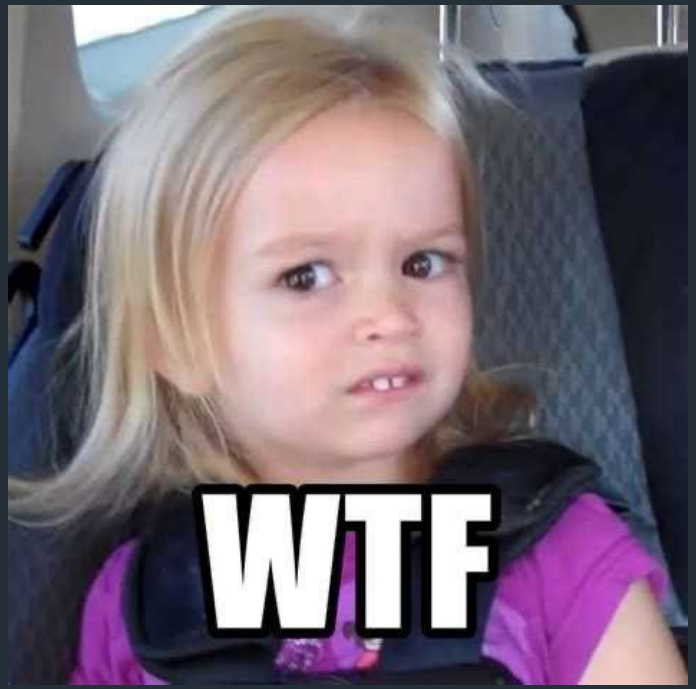
Revised Remote Testing Rig #2



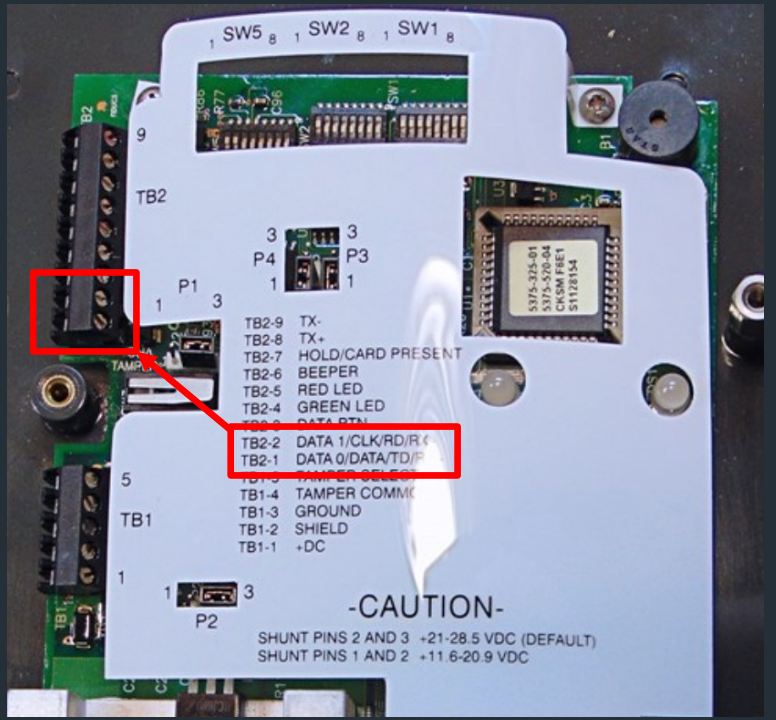
No Card Data?


Tusk
Captured access card credentials are listed below 👇
Search
No card data found.

CFP was Accepted



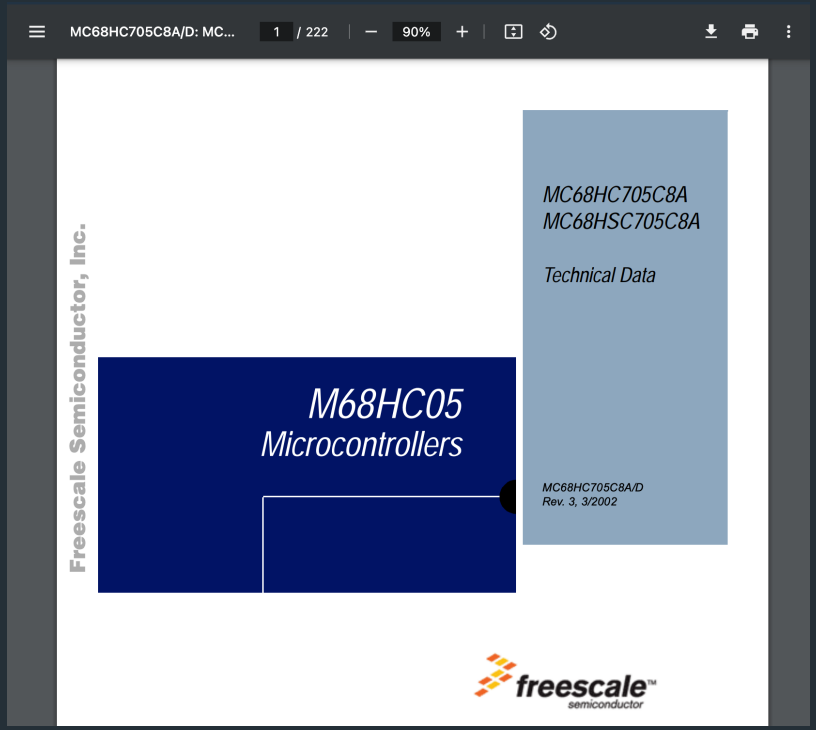
Digging Deeper



Identifying the Reader’s Microcontroller

- MC68HC705C8ACFNE
- 8-Bit Microcontroller Unit
- Uses M68CH05 CPU
- 8KB EPROM memory
- Number of Input/Output pins

Datasheet



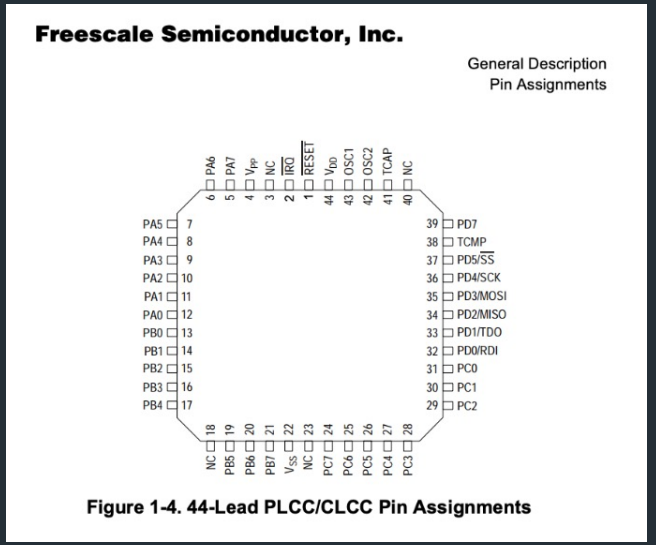
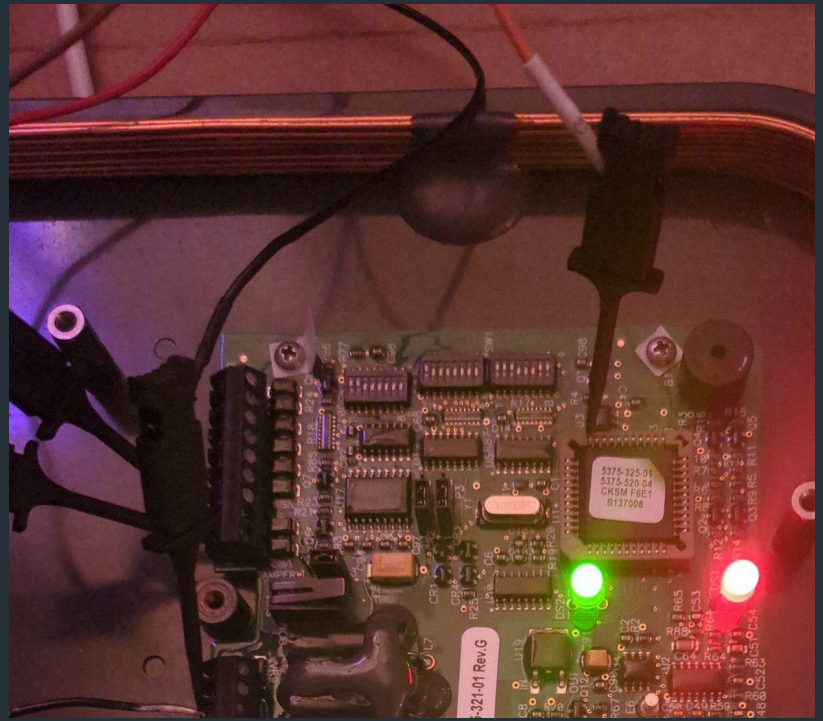


Figure 1-4. 44-Lead PLCC/CLCC Pin Assignments

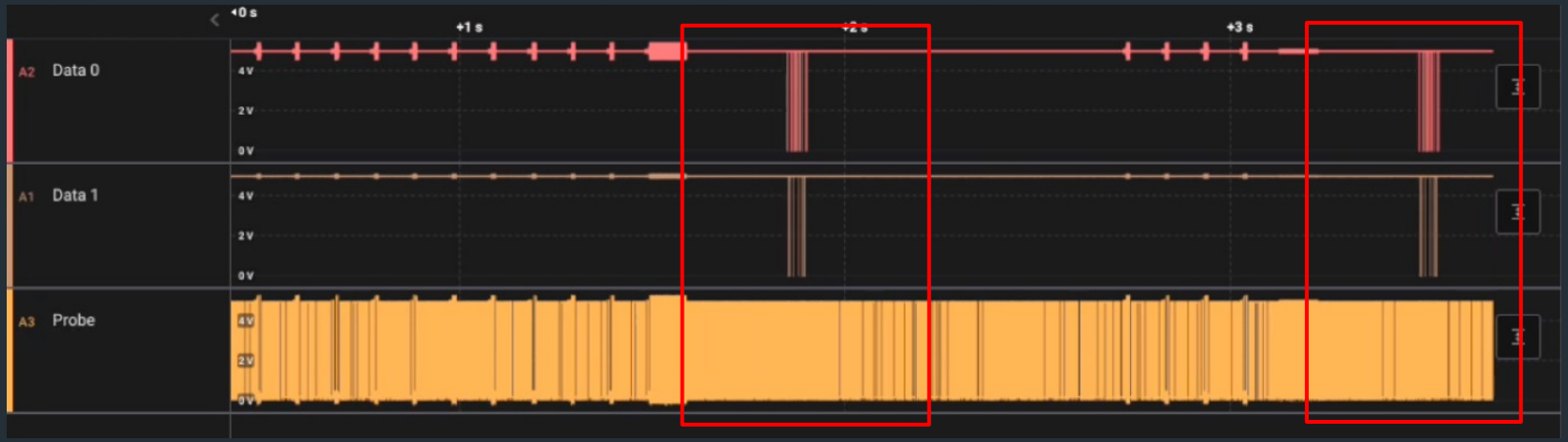
1.7.11 Port D I/O Pins (PD7 and PD5–PD0)

These seven lines comprise port D, a fixed input port. All special functions that are enabled (SPI and SCI) affect this port. See 7.6 Port D.

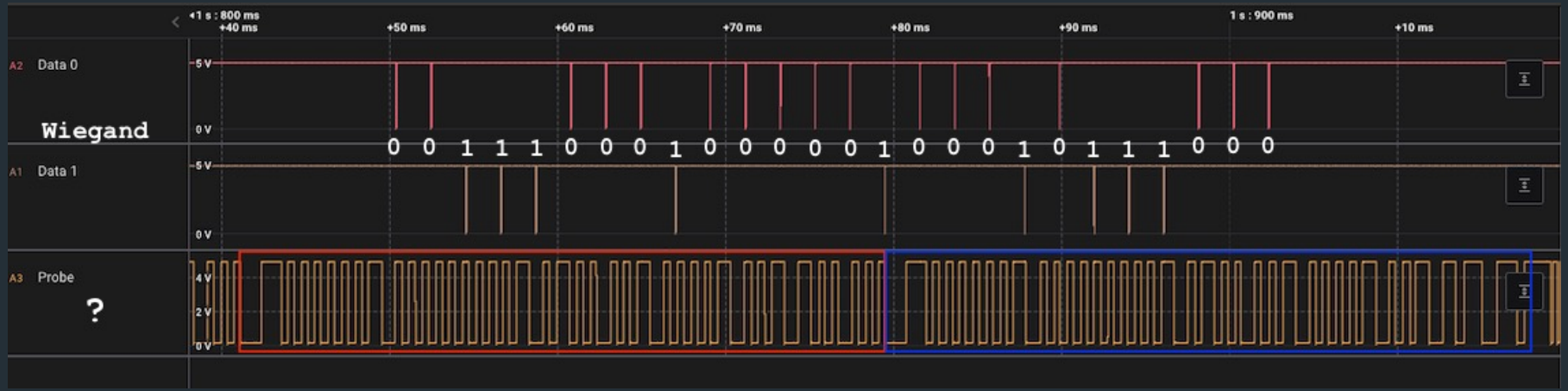
Probing the Reader’s Microcontroller



Logic Analyzer Output

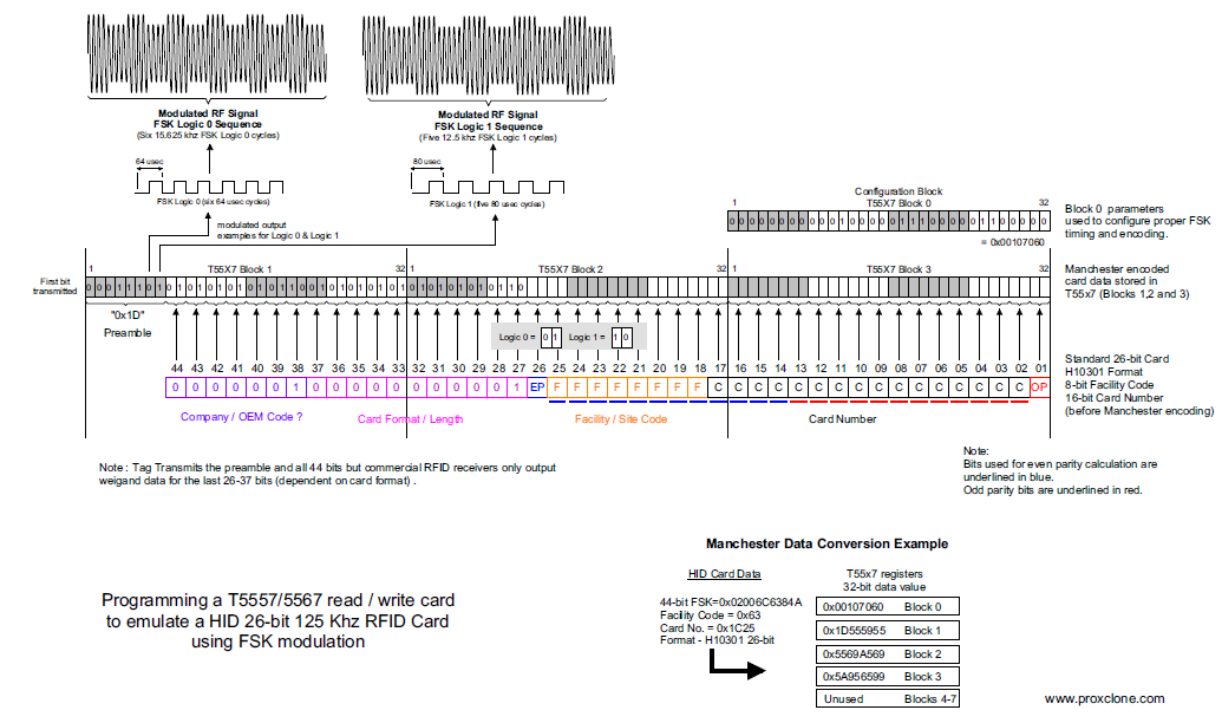


RF Signal for HID Card



RF Signal for HID Card

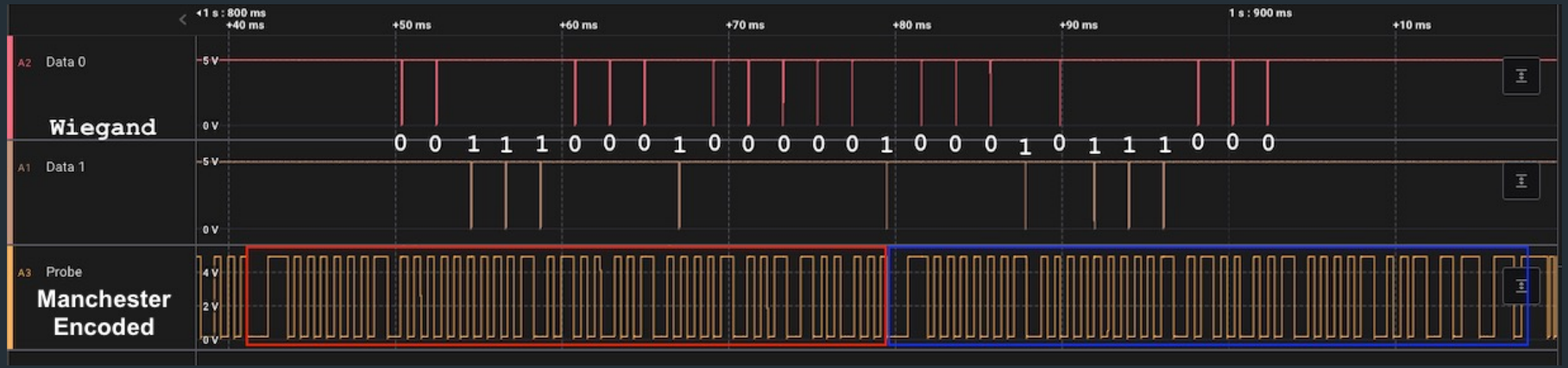
RF Signal for HID Card



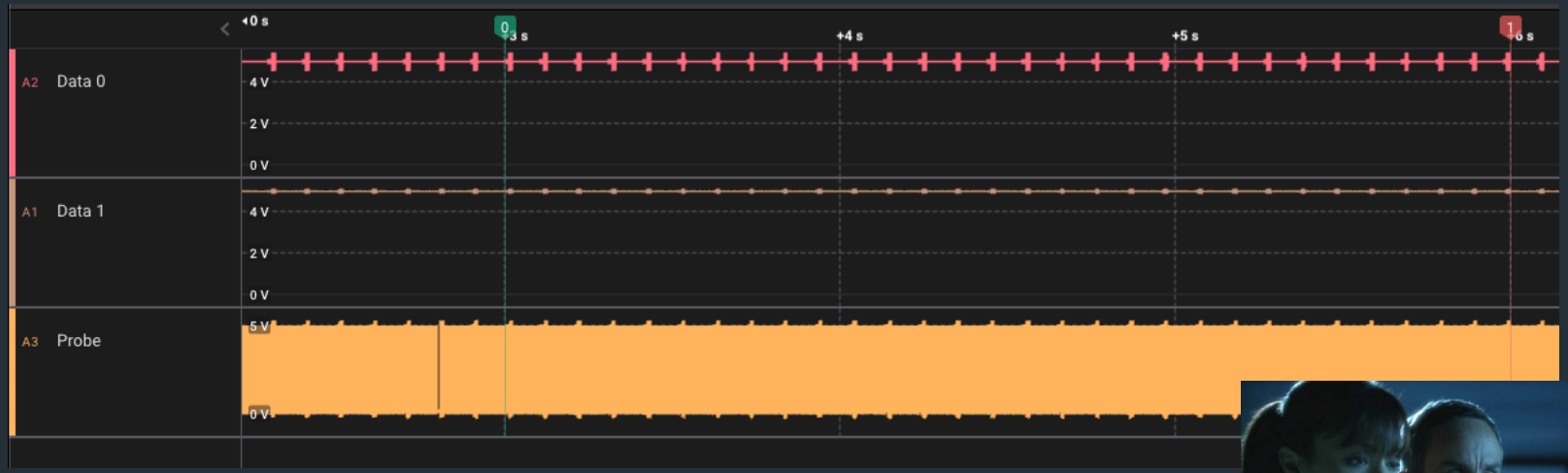
RF Signal for HID Card

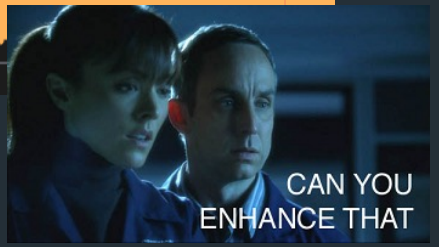
```
Reader Probe pin output:
00011101010101010101100101010101010101011001011010100101011001010101011001010110011010100101

Appears to be manchester encoding
Logic:
01 = 0
10 = 1

        44 Bits ->
00011101  01010101010110010101010101010101011001011010100101011001010101011001010110011010100101
        0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0
preamble

        0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  1  1  1  0  0  0  1  0  0  0  0  0  1  0  0  0  1  0  1  1  1  0  0  0
        44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9  8  7  6  5  4  3  2  1
                                                            EP F  F  F  F  F  F  F  F  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  OP

EP = Even Parity
OP =  Odd Partiy
F  = Facility Code
C  = Card Number

Manually decoded FC & CN bits: 00111000100000100010111000
```

RF Signal for HID Card



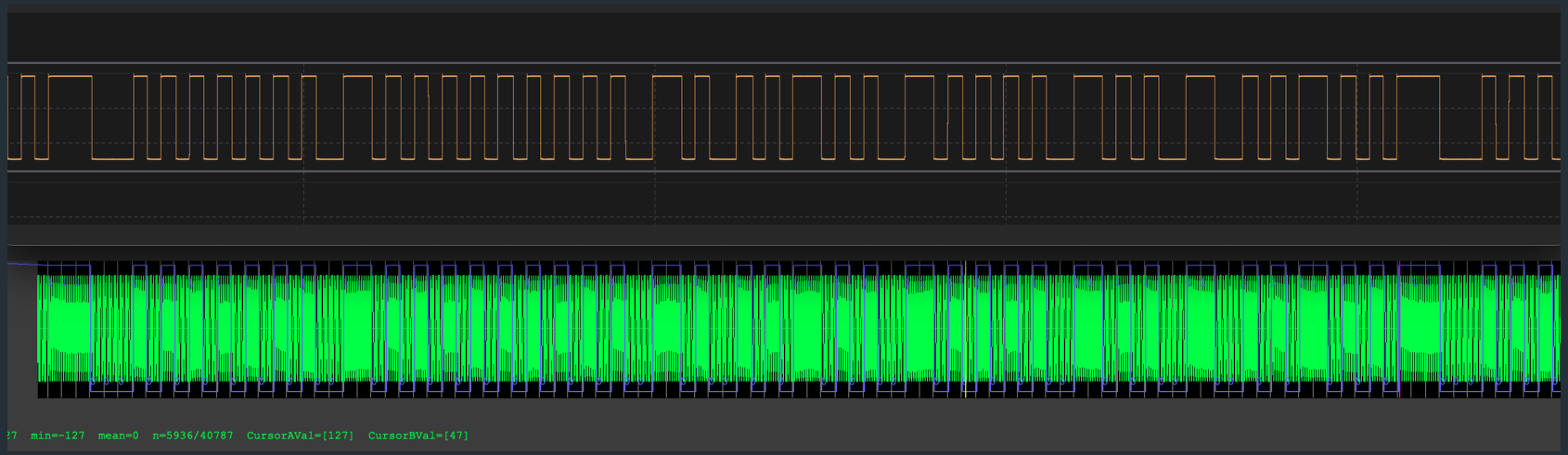
RF Signal for Gallagher Card





RF Signal for Gallagher Card

RF Signal for Gallagher Card



Progress

- The “raw” Gallagher card data is being picked up by the reader’s antenna / RF frontend and is being sent to the onboard microcontroller
- The firmware on the microcontroller is clearly doing some processing to determine ‘valid’ card data and is discarding the Gallagher card data as noise
- We need to bypass the logic of the microcontroller and sample data directly from the bitstream sent from the antenna / RF frontend

How to Sample Data from the Microcontroller

- Write some code which does the following.
  - Detect specific patterns in the input bit stream
  - Extract the relevant bits when pattern is seen (HID preamble or Gallagher fixed seq)
  - Filter out glitches (due to unstable input - perhaps noise)
  - Manchester decoding
  - Output to relevant functions to de-obfuscate Gallagher card data

Card Data Examples

- Generic 26-bit HID Card Data:
  00011101010101010110010101010101010101011001011010101011010100101100110011010101001011010

- Gallagher / Cardax 125kHz Card Data:
  01111111111010101010001101000101011000101010100101101010001101010001100010110011101001 0

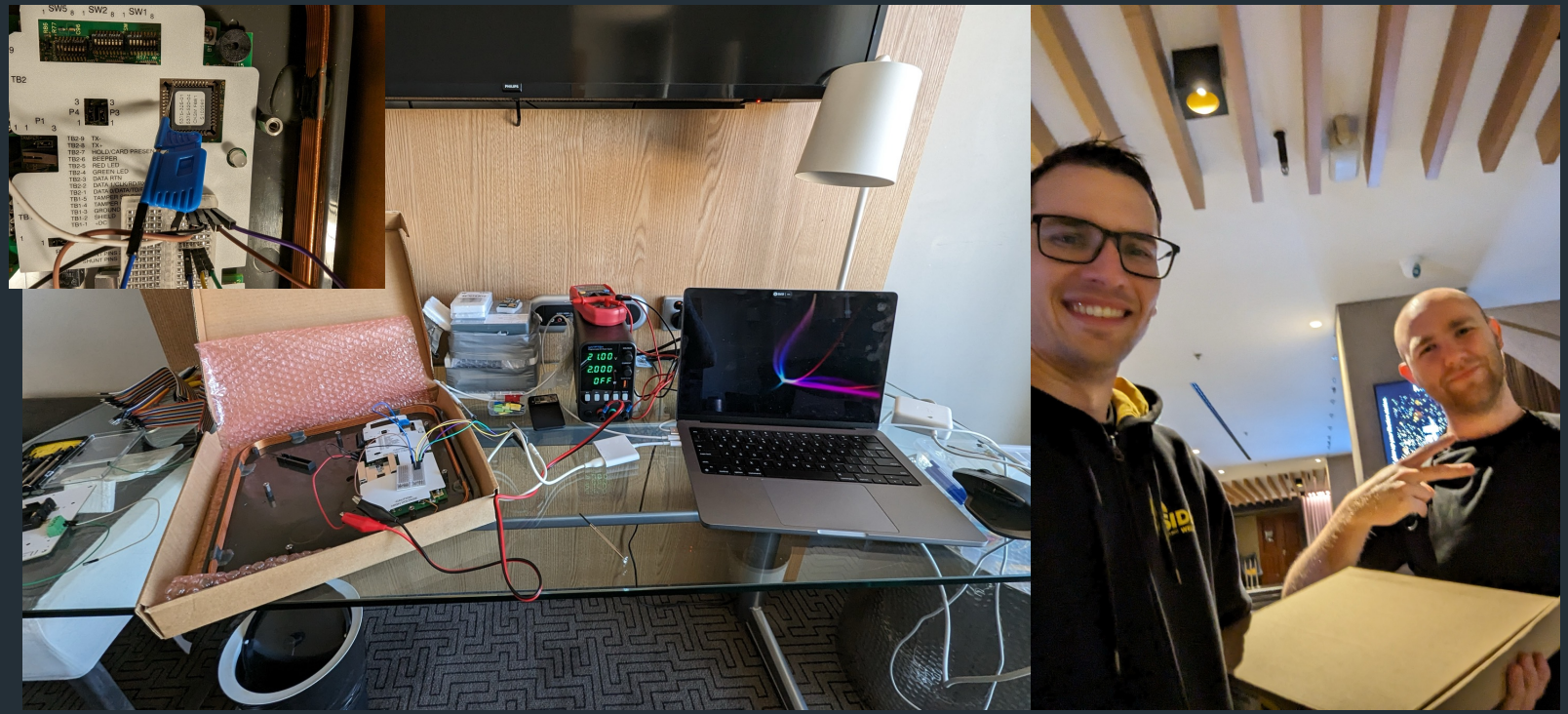
Sampling Card Data Example (HID)

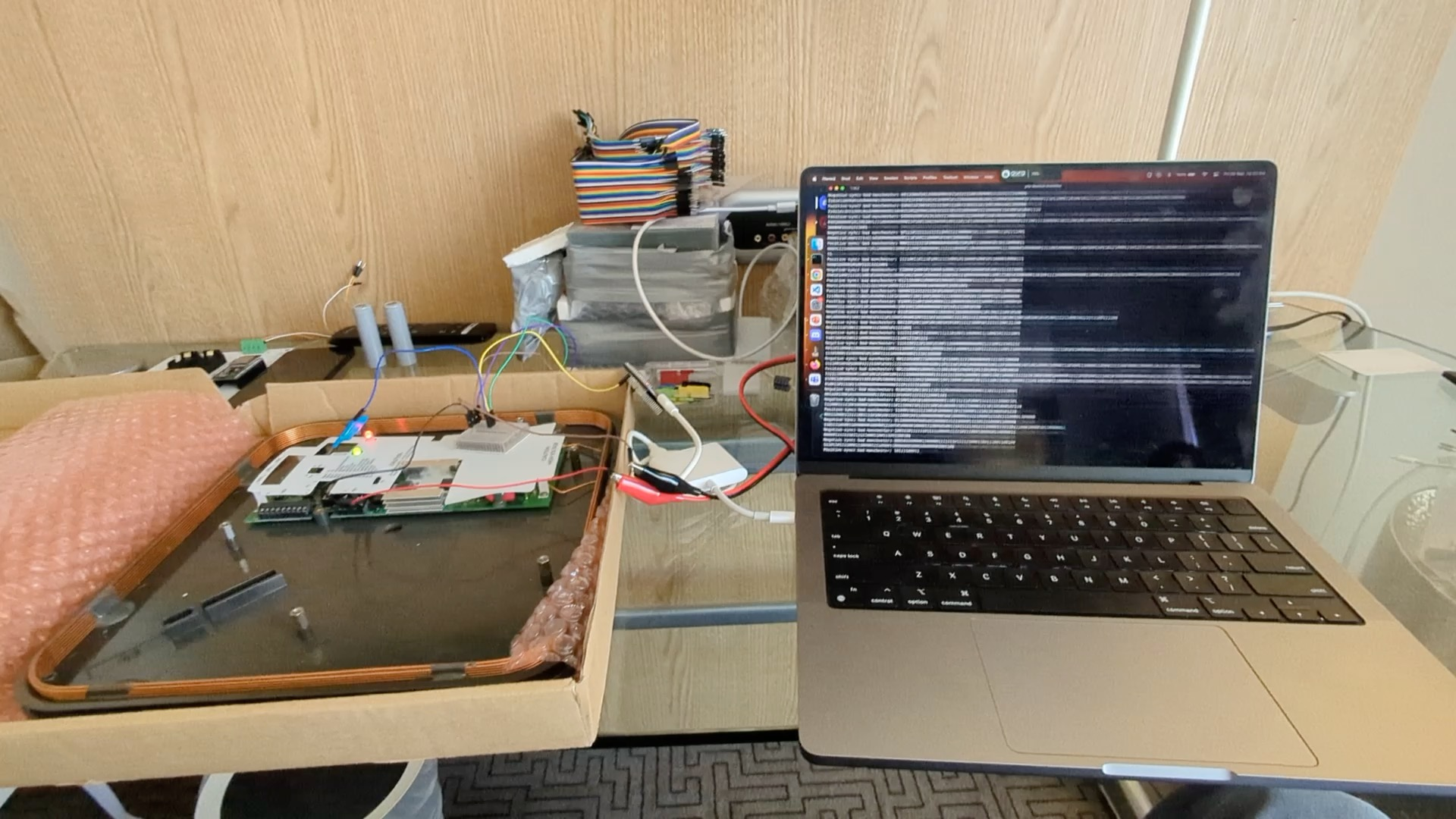
...01010001010000110010101100000101000000100010100100100
01001100000001000000010000010000001100000010110001001010
110000001010110000**00011101****010101010101100101010101010101**
01011001011010100101011010100101100110011010101001011010
1000110000000110000000101100001100101011000000101011 0...

Sampling Card Data Example (Gallagher)

```
...01010001010000110010101100000101000000100010100100100
01001100000001000000010000010000011000000010110001001010
11000000101011000001111111111101010101000110100010101100010101010010110101000110101010001100010110011101001001100000000110000000101100001100101011000000101011O...
```

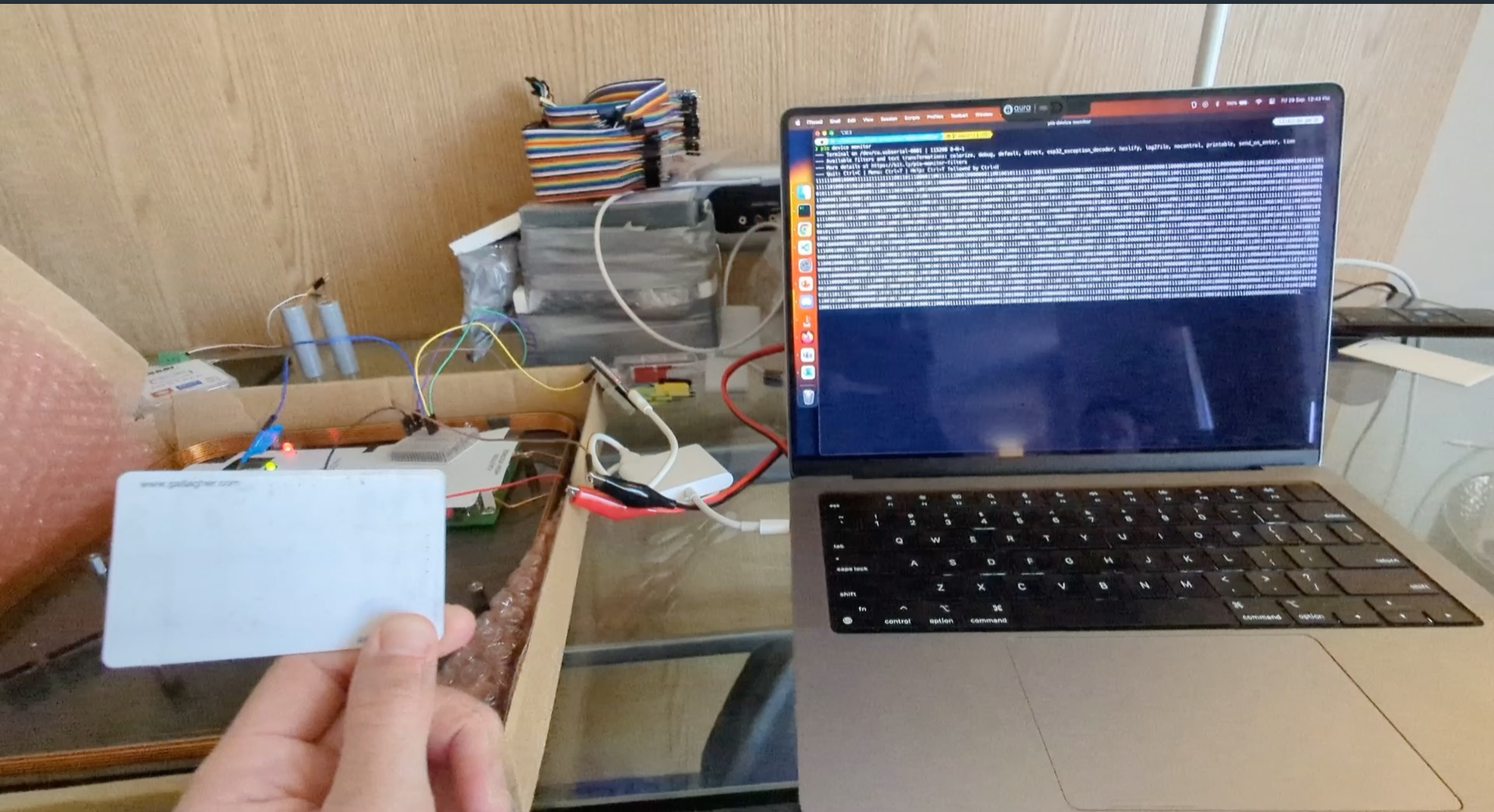
Finally Buying a Reader (thx evildaemond!)



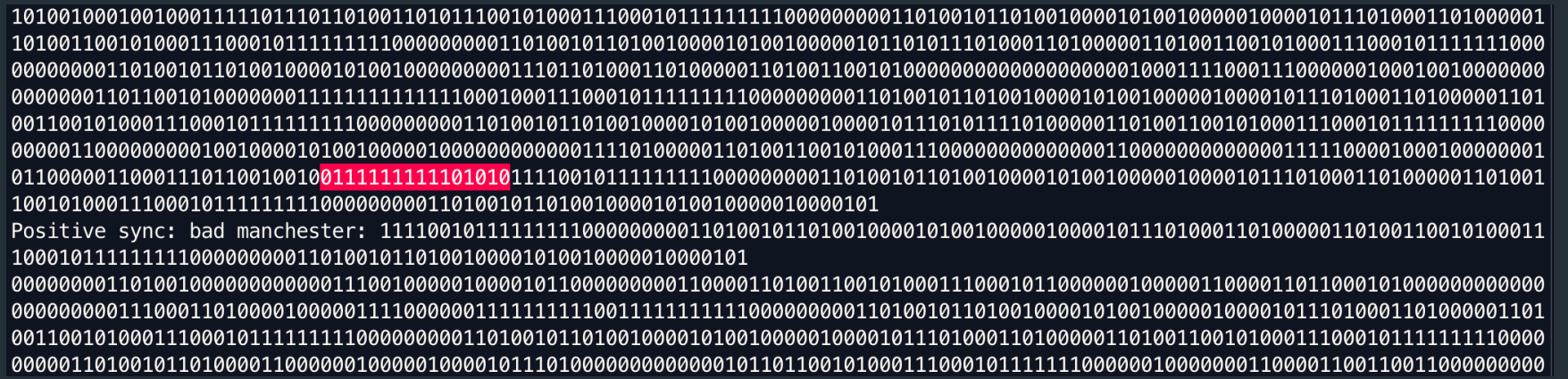


Demo (New Operation Mode) - HID

```
Positive sync: good manchester: 0000001000000000000100111000
001010010110011001101010100101101010000111010101010101011001010101010101010101100101101010010101
Positive sync: good manchester: 0000001000000000000100111000
001010010110011001101010100101101010000111010101010101011001010101010101010101100101101010010101
Positive sync: good manchester: 0000001000000000000100111000
001010010110011001101010100101101010000111010101010101011001010101010101010101100101101010010101
Positive sync: good manchester: 0000001000000000000100111000
001010010110011001101010100101101010000111010101010101011001010101010101010101100101101010010101
Positive sync: good manchester: 0000001000000000000100111000
001010010110011001101010100101101010000111010101010101011001010101010101010101100101101010010101
Positive sync: good manchester: 0000001000000000000100111000
001010010110011001101010100101101010000111010101010101011001010101010101010101100101101010010101
Positive sync: good manchester: 0000001000000000000100111000
001010010110011001101010100101101010000111010101010101011001010101010101010101100101101010010101
Positive sync: good manchester: 0000001000000000000100111000
```



Demo (New Operation Mode) - Gallagher



Status

- A legacy operating mode (using Data 0 / Data 1 lines)
- New operating mode (sampling directly from the reader microcontroller)
  - Code is mostly working to sample the card data
  - Functionality exists to decode and de-obfuscate Gallagher 125kHz card data
  - TODO: Combine the above and debug issues
  - Figure out how to get this ready for in-the-field use (PlayStation style mod?)

Going Forward

- HID
- Gallagher
- EM4100?
- Other low frequency card formats?

Bill of Materials for PCB

- 1 x PM254V-11-06-H85
- 2 x PM254-1-19-Z-8.5
- 1 x DB128V-5.0-4P
- 1 x BH-18650-B1BA002
- 5 x SMD 21700 Battery Holder
- 1 x ESP32-DevKitC V4
- 1 x Micro SD Card Reader Module (with level converter)

Thanks & Questions

- BSides for accepting my talk
- Aura Information Security for the time to do this research
- Matt Daley (megabug) as per usual for helping me with my endeavors
- Evildaemond for hooking up a reader at BSides
- Attendees for your time and listening to my talk
- Everything is open-source: **https://github.com/TeamWalrus/tusk/**
- Hit me up on X (twitter) @dunderhay
- Red team enquiries: redteam@aurainfosec.com