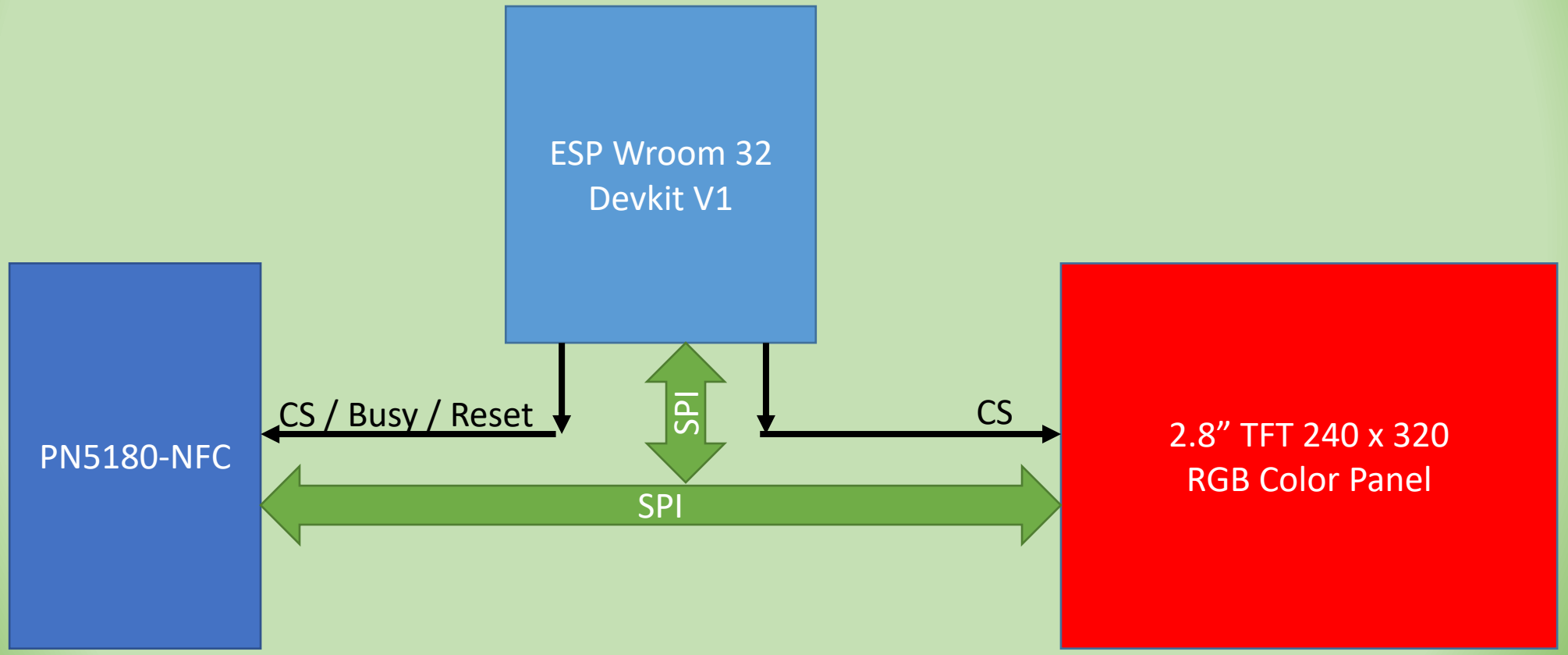
RFID Tag Scanner

Alan Lomax (M8640)

MERG RFID Special Interest Group

Zoom Meeting #29

Block Diagram



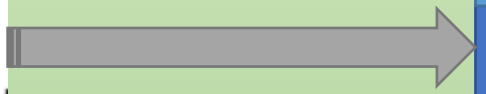
Block Diagram (Logical)

(1) Initially there will not be a connection to your WiFi. Device will set up its own Access Point. By connecting to this AP you can configure the WiFi Credentials for your own WiFi.

(2) Once on local WiFi unit will scan for RFID tags, update status numbers, and also update received MQTT messages. (You specify all the details).

Configure

Monitor



ESP Wroom 32 Devkit V1

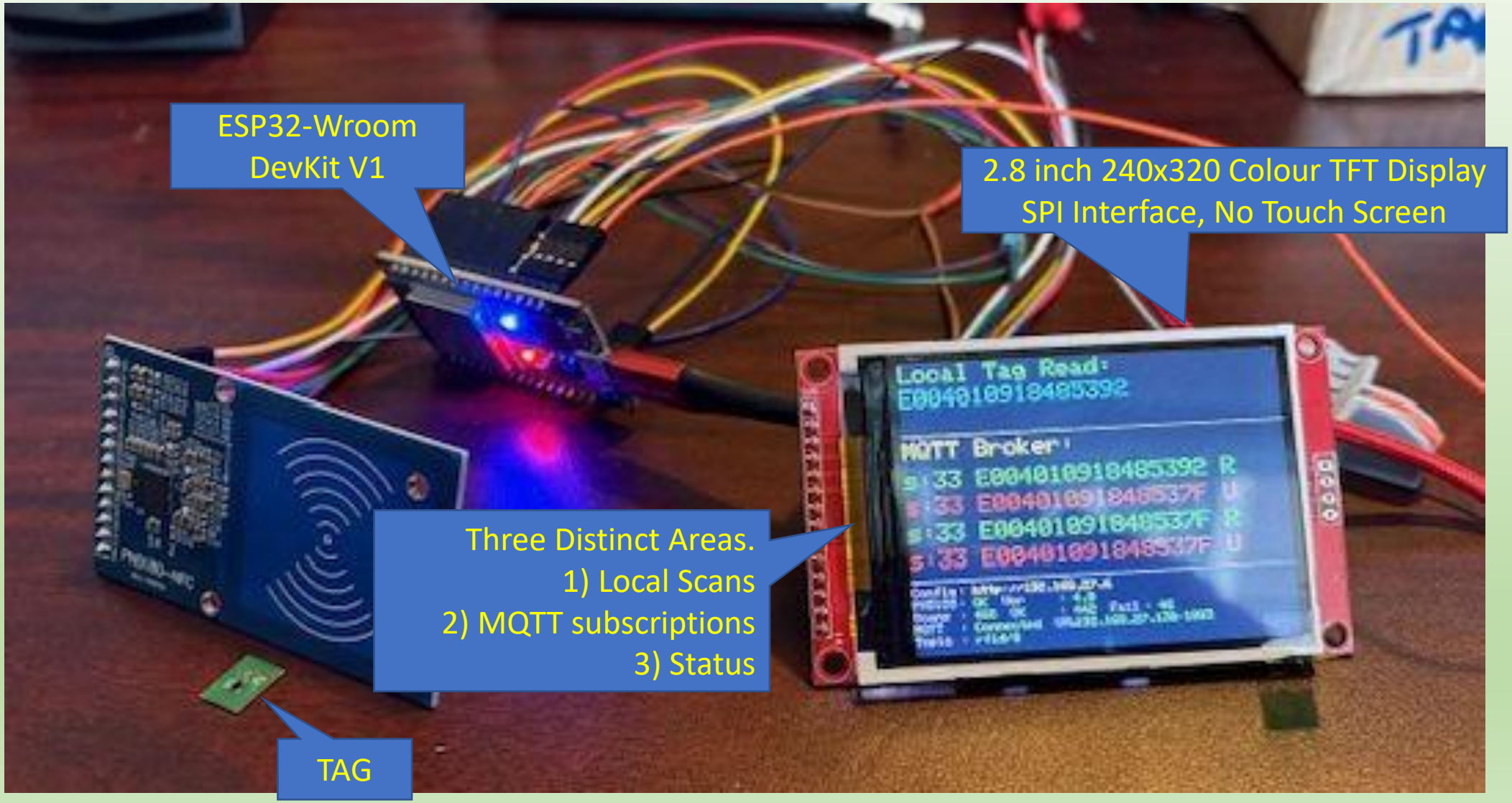
PN5180-NFC

2.8” TFT 240 x 320
RGB Color Panel

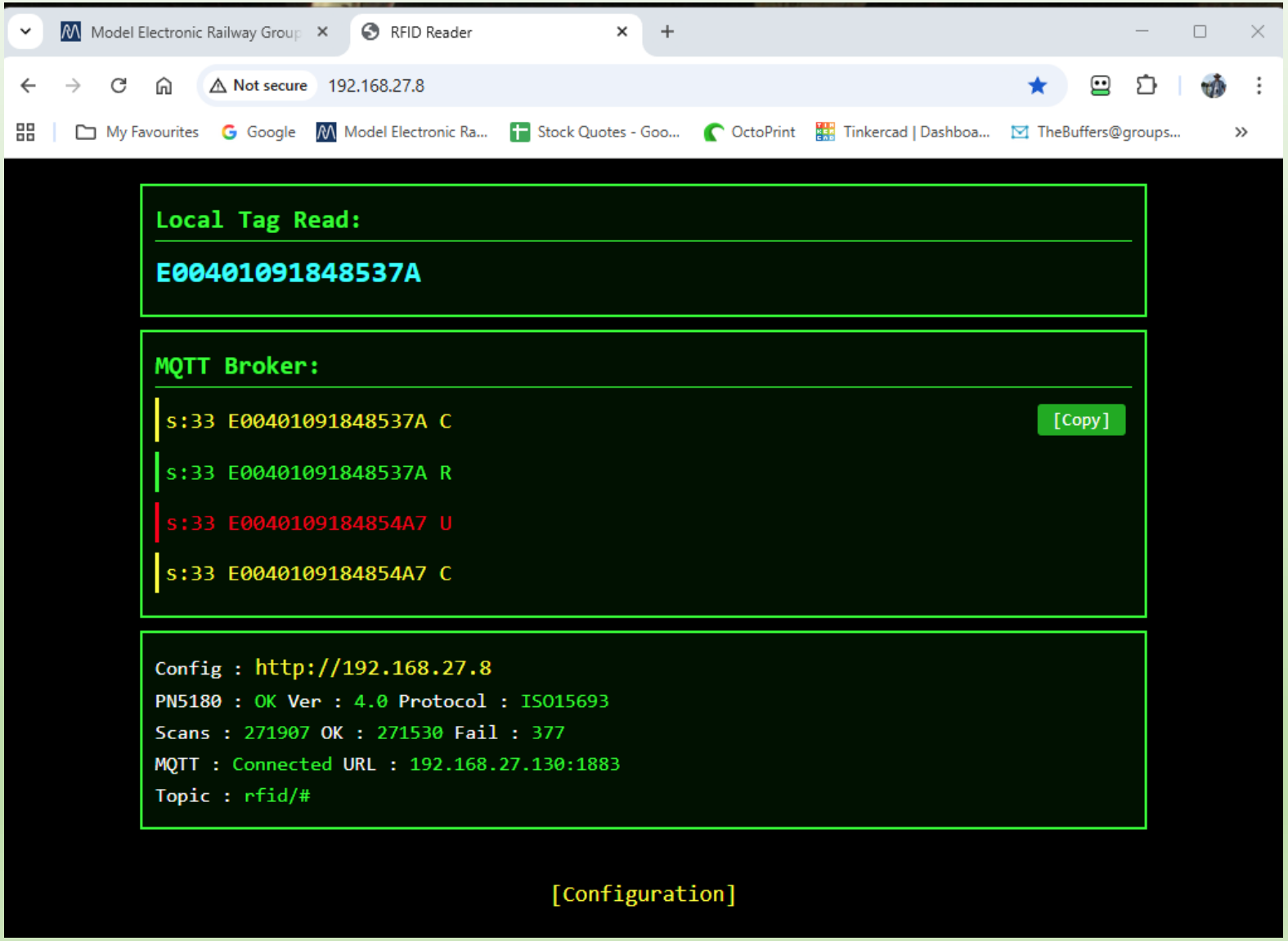
(3) Tag scanning occurs several times a second. When an RFID tag is in range the GUID is captured and published to MQTT.

(4) The Local display shows all the results. These are also shown on the web page

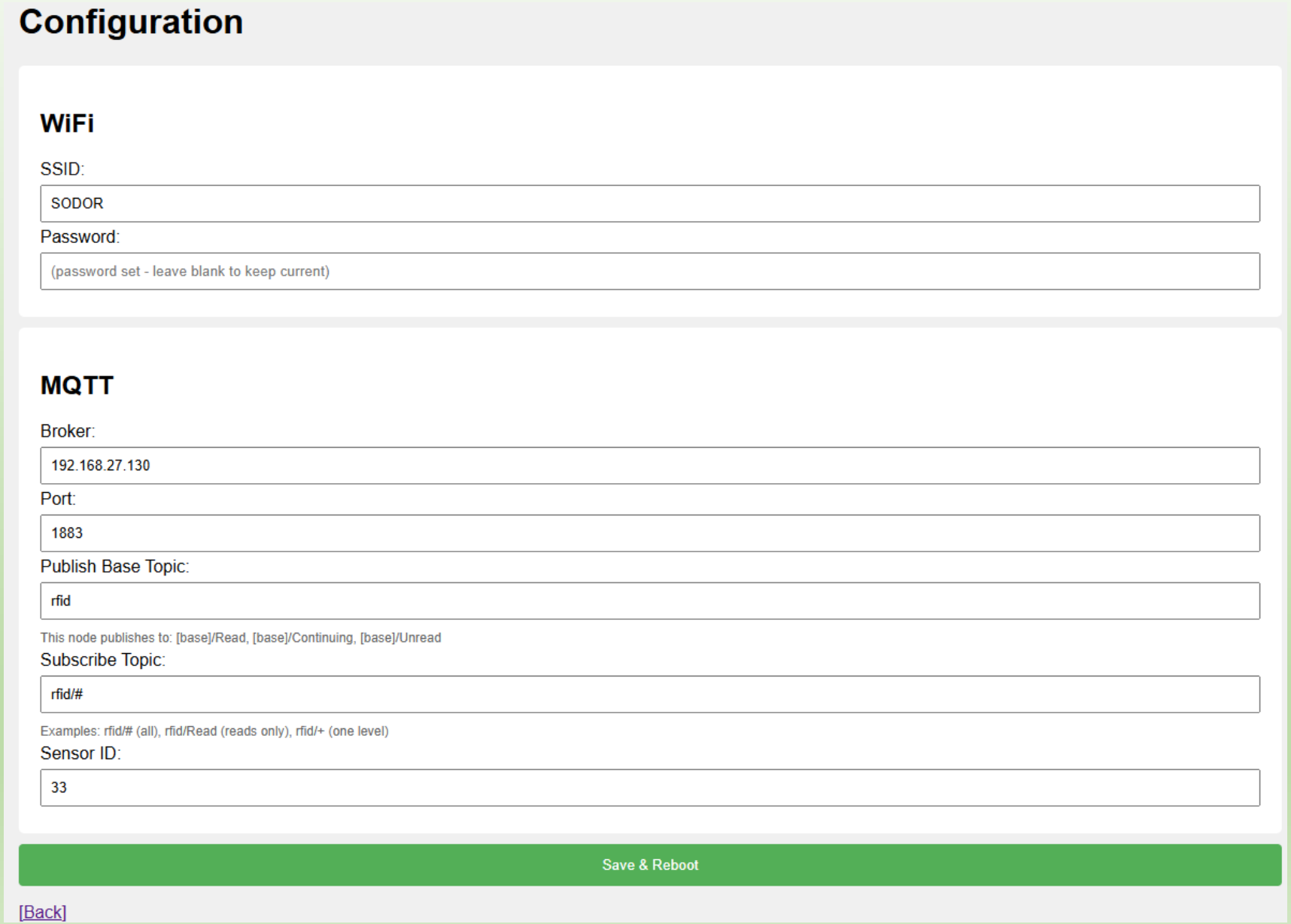
At this Moment



Web Interface (Main)



Web Interface (Configuration)



Code Development

Previously I had developed a “Stock Ticker” display as an XMAS gift for my son in law. Reusable elements from this effort were expected to be:

1) ESP8266 Libraries
2) Local Display Interface Library
3) Web Interface Library
4) Initial Configuration and Remote Monitoring of Local Display
5) Saving and restoring data to and from Flash

Working with Claude (AI)

Reusing code from this project (provided) help me develop a well documented program for reading RFID tags using a PN5180-nfc module.

- {provide my pin assignment for this module}
- {detail how I wanted the code broken out into several modules}
- All of this was written up ahead of time (in NP++) and pasted into Claude's window. Response was "Sure I can help you with that!"
- Needless to say it did not work first time. In fact it did not even compile
- Pasting the console output back into Claude quickly resolved compile issues. (Mostly variable scope and duplicate declarations)
- Then started working through the logic issues (Took several days).

Summary of Issues

Hardware:

- Many internet sources claim do NOT mix 5V and 3v3 supplies on this module. Turned out to be bad advice. 5V drives the final RF amplifier. Message should have been do not mix Logic Levels (3V3 logic only)
- ESP8266 does not have enough available GPIO. (Moved to ESP32)

Software

- Several libraries for this module are not fully implemented. Functions that should exist are merely stubs that return without doing anything.
- Local Display Update optimizations (flicker and update timing)
- Timing sequences when setting up scanning modes.

Logic

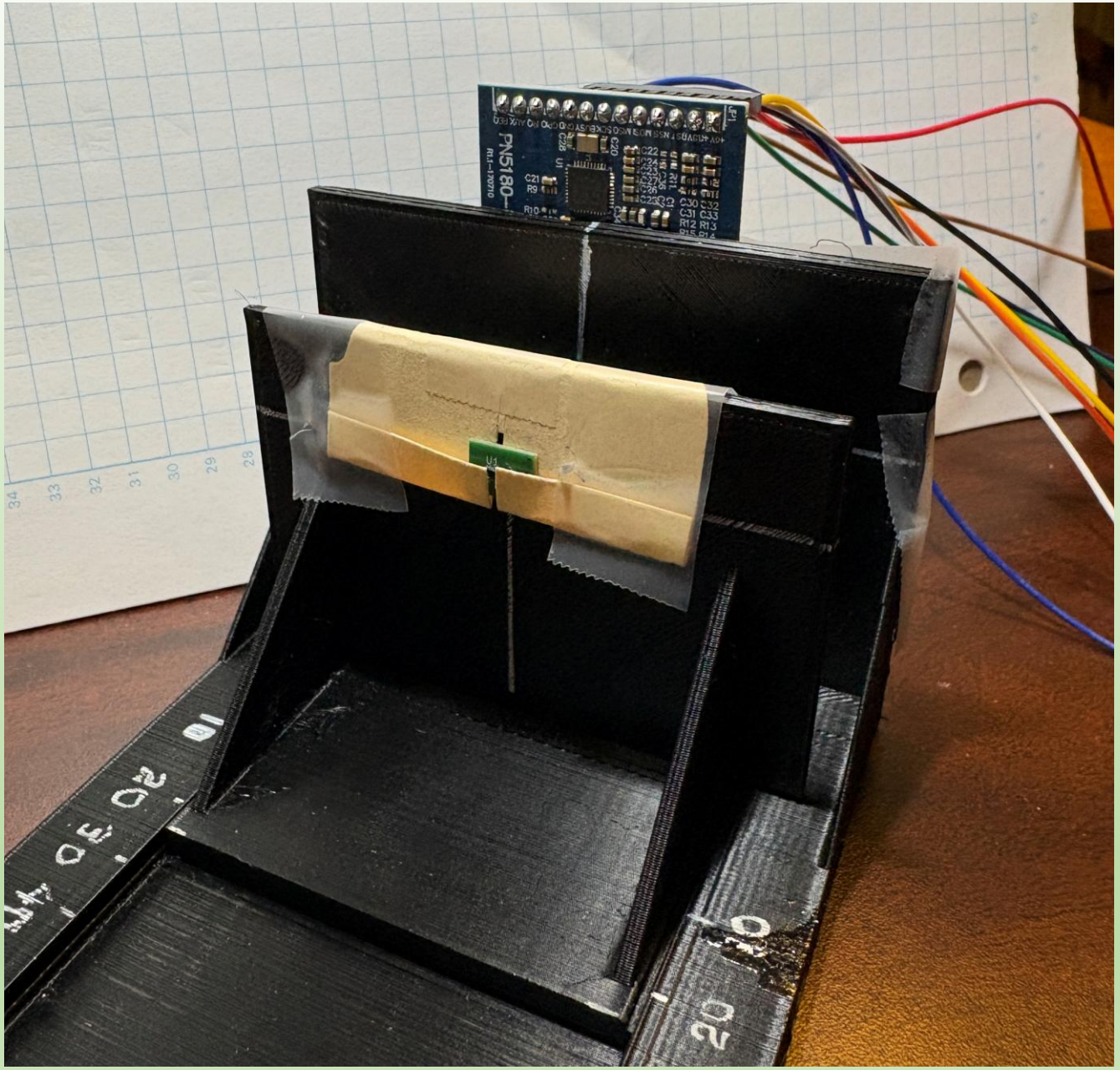
- “Ghost Tags” – UID’s being reported when no tags in range (cleanup).
- **Scope Creep** – adding features which while cool – distracted from goal

In Use
(Range Testing)

- Nick’s design of Test Jig
- Aerial is on fixed back plate
- Tag is on movable sled.(in pocket)
- Cross hairs for alignment
- Scale for distance

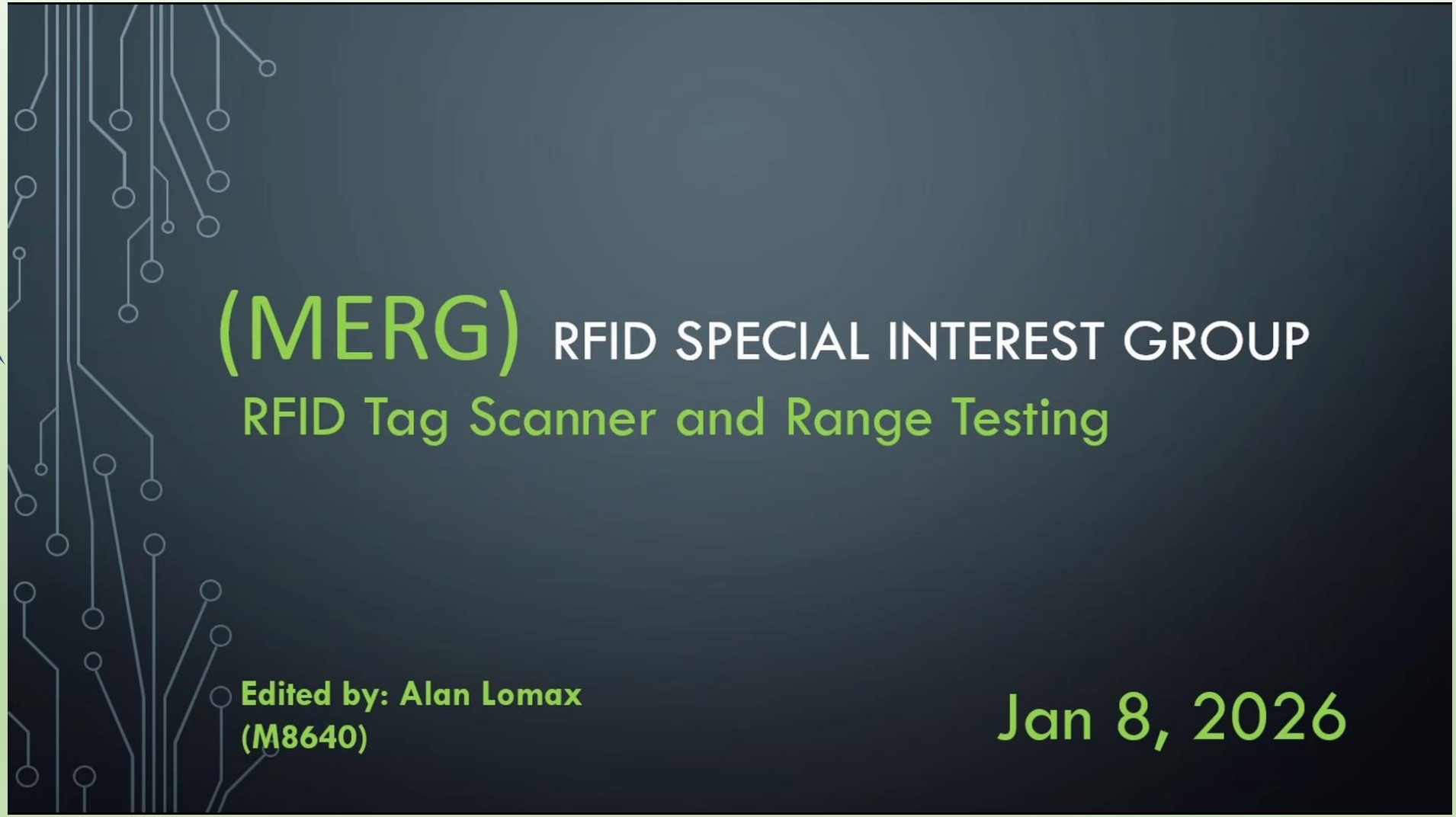
- Shown with PN5180-nfc
- Also tested SFR small aerial

- 55 tags .. Each tested twice



30 Second Video (Scanner In Use)

Play Me!



Many Tags Were Built

Tags

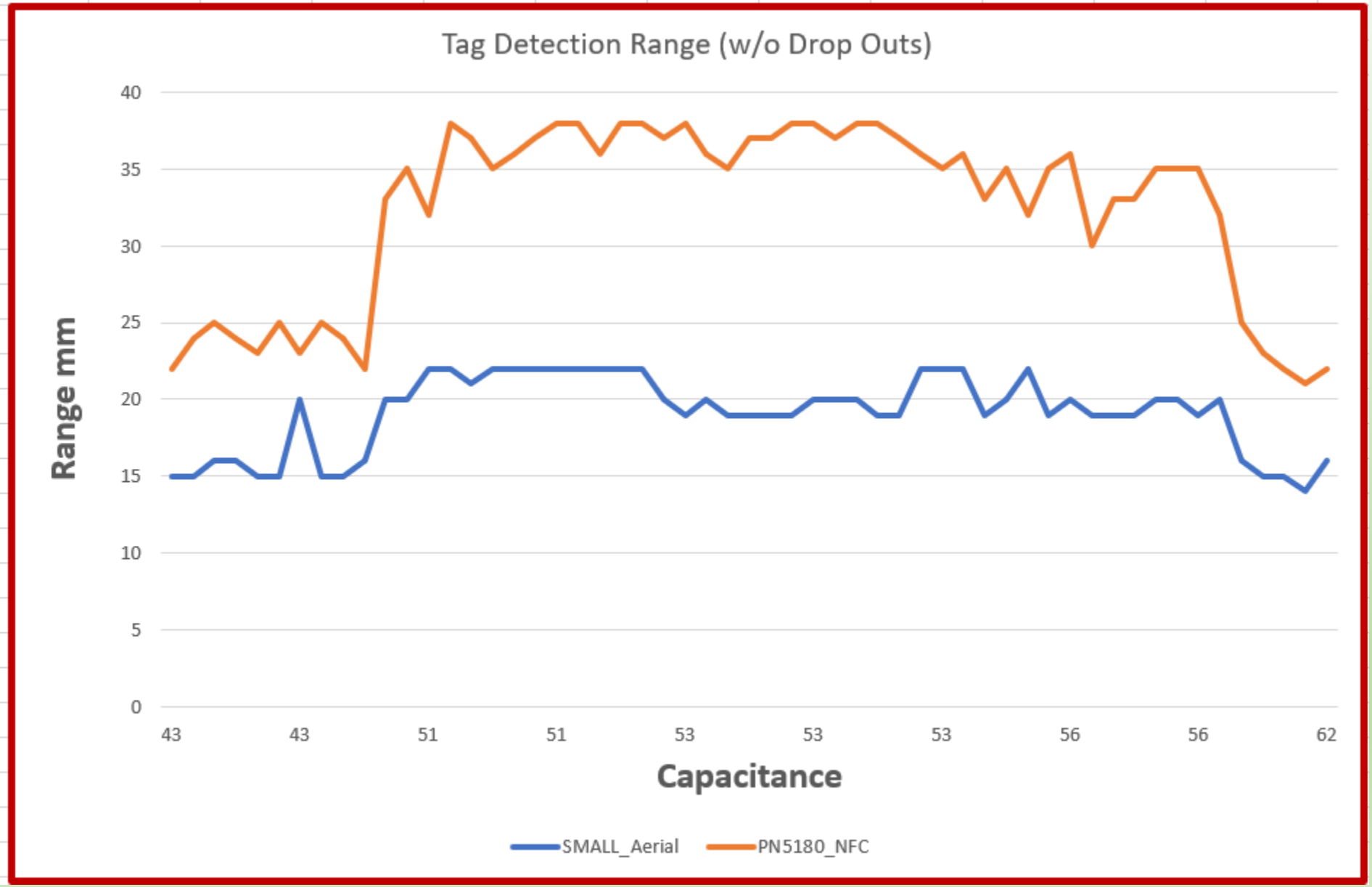
- Each PCB RFID Tag was numbered (before assembly)
- Various values of matching capacitors were used (43pF, 51pF, 53pF, 56pF, 62pF)
- Go / No-Go testing screened out defective tags (~5%) (build issues)
- Also 10 StaRFIshrail Rail ‘control’ tags were tested. (C unknown)

Testing

- Drop tag into pouch on the sled part of the test jig
- Side it into and out of range. Intermittent reads did not count.
- At point where no read failures occurred the distance was recorded (along with the GUID)

After using the PN5180-nfc reader replace with a standard SFR aerial and repeat.

Results


Tag Detection Range (w/o Drop Outs)
Range mm
40
35
30
25
20
15
10
5
0
43
43
51
51
53
53
53
56
56
62
Capacitance
SMALL_Aerial
PN5180_NFC

The End