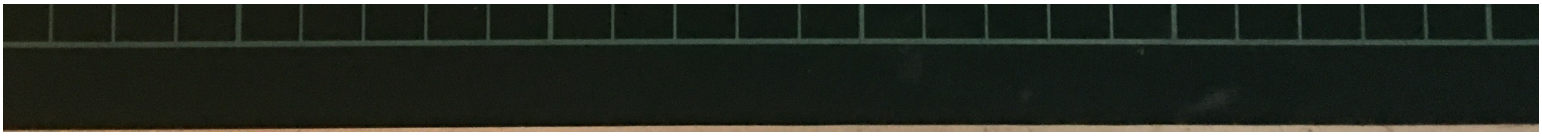
Feathering for SSIDs.

El Kentaro Follow

May 5 · 8 min read

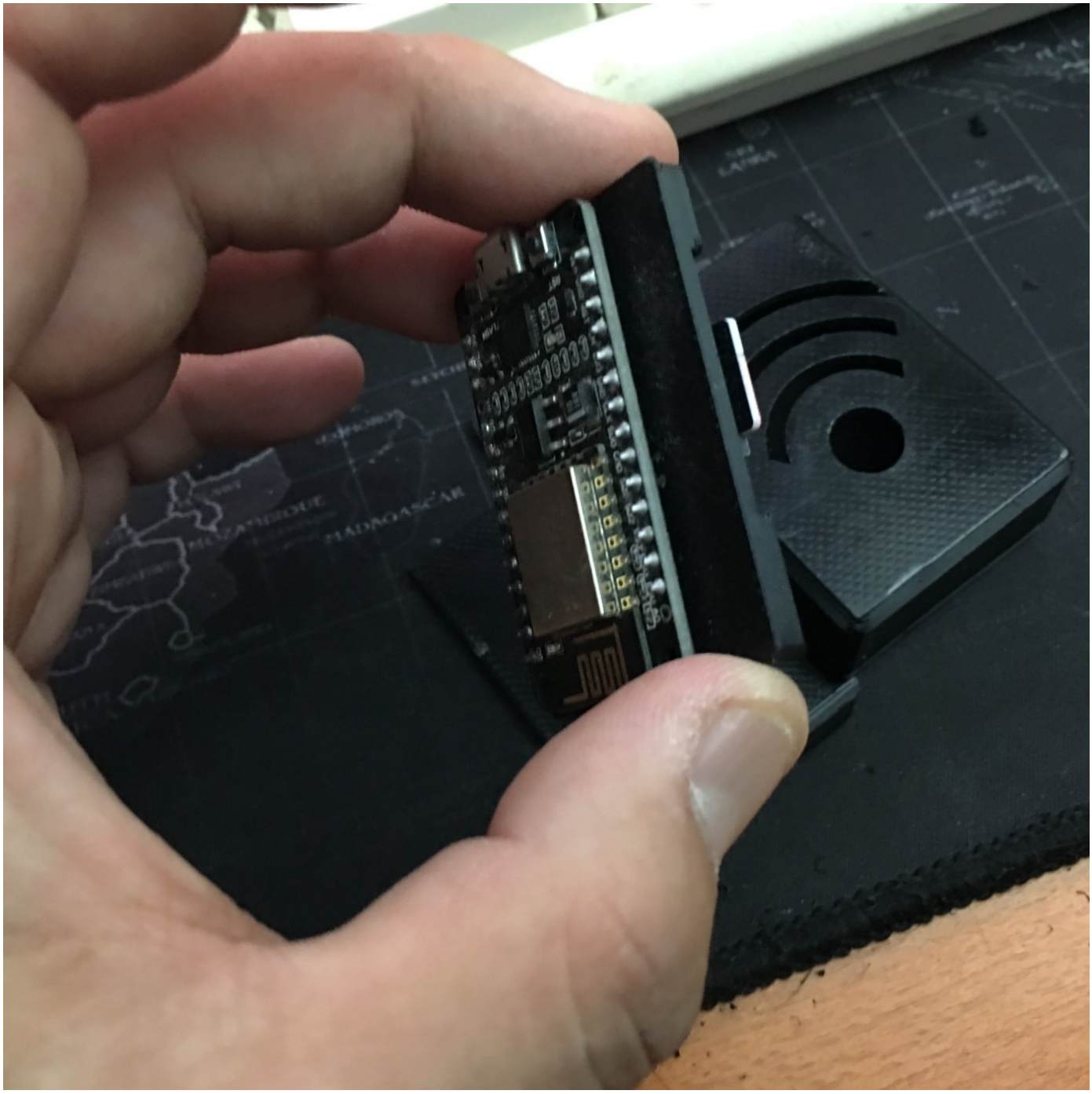


Prologue.

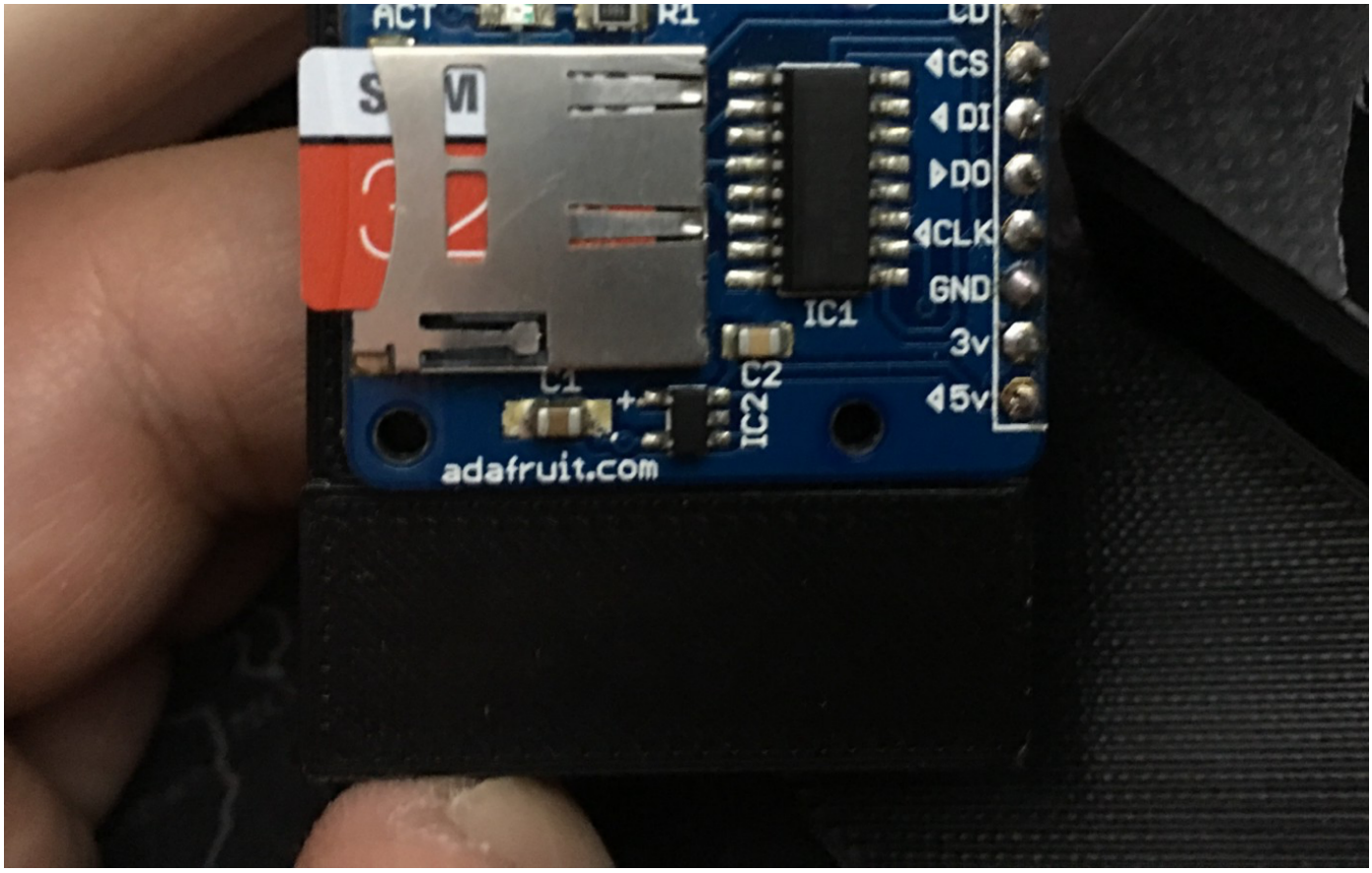
Its been a while since the last time I posted one of my projects. I've been on the road constantly for the last couple of month, every time I decided to sit down and make a new toy a 12 hour trip to another place would come up. As a hacker maker it was frustrating and my workshop was suffering. Stuff was just piling up. New parts laid in boxes delivered on the few times I was at my workshop. On the road I had kept a memo of all the stuff I wanted to build. Finally after traveling every month I finally had some time to clean the workshop, open up the boxes of parts and stuff and actually sit down an build a new toy. After cleaning up my workshop, I was ready for another build.

Fishing for SSIDs.

I like collecting SSIDs, I don't do much with them other than look and enjoy what people name their Wifi access points. Many are generic default names based on the routers but once in a while you will find somebody with a sense of humor. So over the years I have build different SSID collecting devices. Out of them the most basic one I build using a NODE-MCU is the one I travel with the most. I do Wiggle when I travel but often I forget my Android home at my batcave. Unlike some of my friends I don't aim for the "most captured" on the Wiggle rankings or conduct a deep dive of the pcaps or captures, I just like to collect them. Call me a SSID hoarder. The NODE-MCU one satisfies this need, it just collects SSIDs and writes them to 2 files, one for all "open" access points and another one for everything it sees.



The NODE-MCU SSID Collector.



Back side of the NODE MCU SSID Collector

Open or Trap?

Now the NODE-MCU collector is like a fishing net it grabs everything it sees, but there are times I wanted to see whats around me and be able to check if the network was a true open network, meaning I could get out to the internet or not or if it was a captive portal. I could use my phone and connect to each AP and see if I could get out but I needed something simpler. So I decided to build another collector. This one had some different requirements, I wanted it to run off a battery, (the NODE-MCU one uses an external battery pack). It needed to have a screen and some way of interacting. So I picked up a M5 Stack. The M5 Stack is an Arduino compatible platform that has a screen with buttons and a battery inside. You can add different modules like GPS etc etc to it if needed. The downside of the M5 stack was that the documentation could use some

“improvement.” But I got one up and running and it works fine. You can read about this build here.

MOAR Gadgets.

So my hoarding needs were met, or so it seemed. I was traveling and it occurred to me that there would be times where I would want to trigger a collection log its findings and be able to find the AP with the strongest signal, like finding that river monster. The one down side of both the NODE-MCU build and the M5 Stack build was both of them lacked a good way to record “when” the scan was done. I needed an RTC (Real Time Clock) to keep track of my scans. I’ve always been a fan of the Adafruit Feather line of products ever since a friend of mine introduced me to them and showed me a packet monitor he build using one. They have a retro-tech nostalgic feel to it and the build quality is very nice. They are very modular and easy to swap in and out. So for this build I decided to use the Feather platform as my base

technology. The Feather has multiple components but the most basic and important concept is the Feather and FeatherWing concept. The Feather will be the main controller for your project and FeatherWings are modules you can add to it. So for the main controller I choose the HUZZAH with ESP8266.

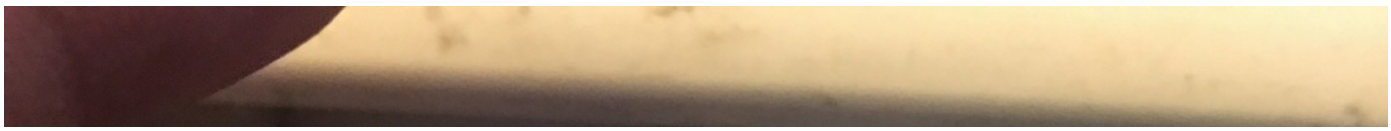
1.The requirements.

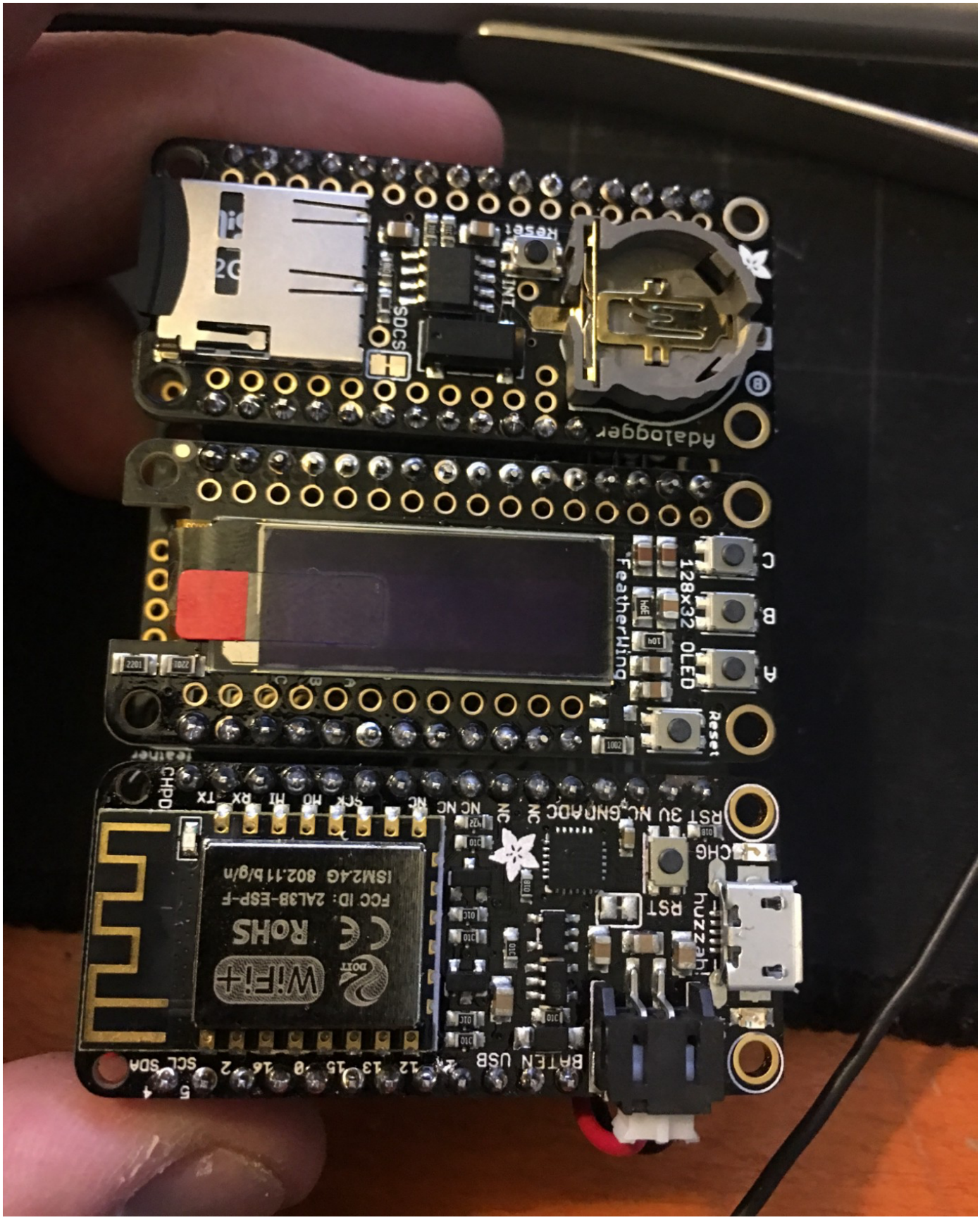
a) Collect and log all SSIDs it sees.
b) Write to 2 files, “Open” and “All”
c) Time stamp each collection for easier management.

2.Parts.

1. Adafruit Feather HUZZAH with ESP8266
   (https://www.adafruit.com/product/2821)
   2. Adafruit FeatherWing OLED — 128x32 OLED Add-on For Feather
   (https://www.adafruit.com/product/2900)
   3. Adalogger FeatherWing — RTC + SD Add-on For All Feather Boards
   (https://www.adafruit.com/product/2922)
   4. FeatherWing Tripler Mini Kit — Prototyping Add-on For Feathers
   https://www.adafruit.com/product/3417
   5. Switch
   6. Lipo battery.

The Assembly.

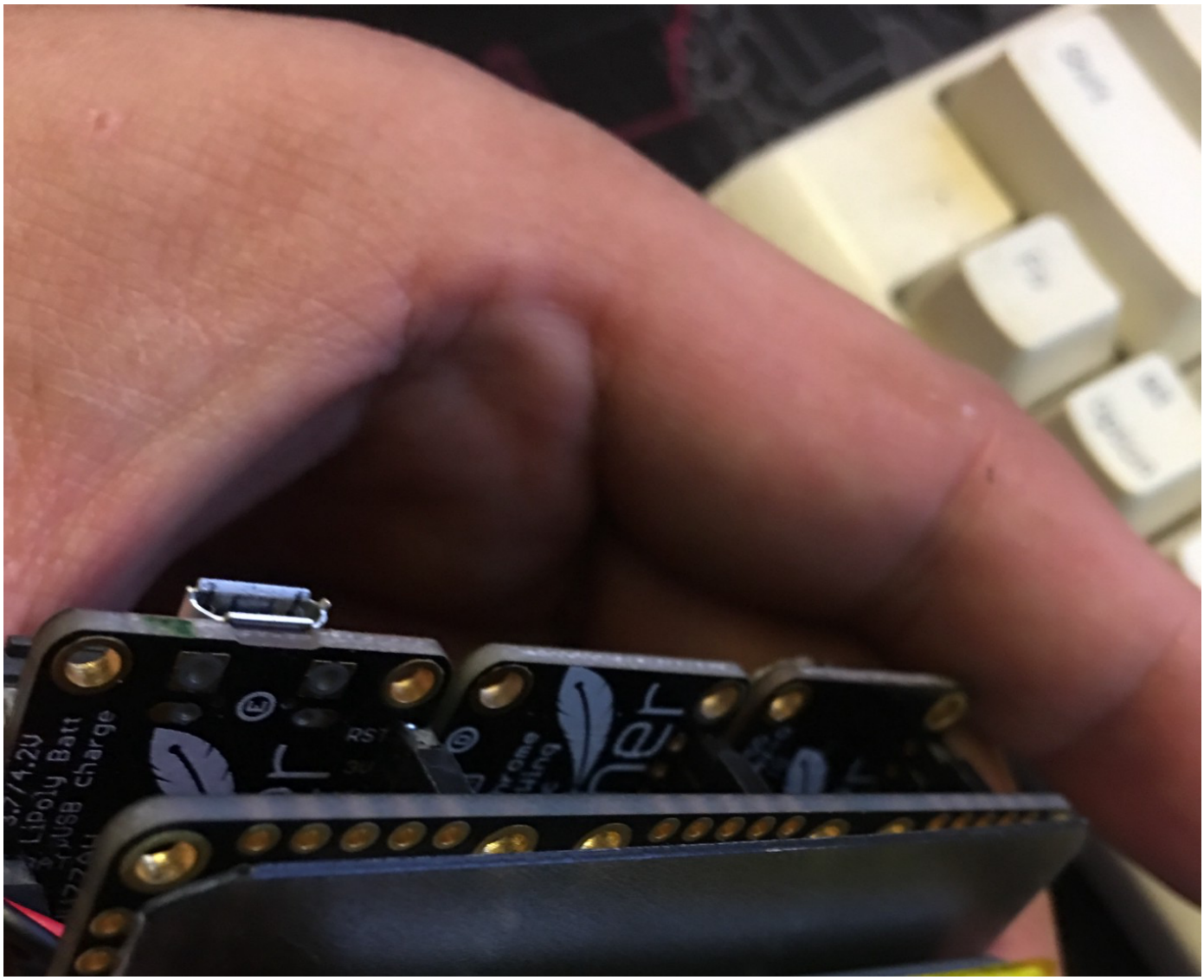




All hooked up.

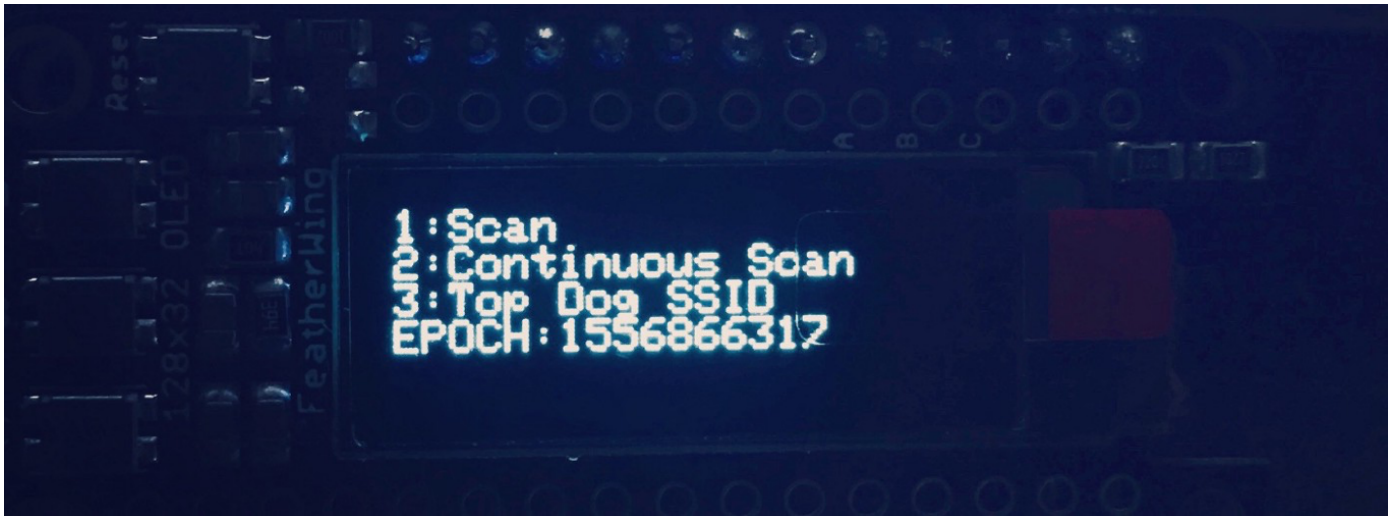
The assembly is pretty straight forward, solder on the headers to the

Feather and FeatherWing and the Tripler Mini Kit. I also soldered a switch on the EN(enable) pin and the GND on the FeatherWing OLED. (https://io.adafruit.com/blog/tip/2016/12/14/feather-power-switch/ .) This is the one of the nicest things of the Feather platform that you could use the FeatherWing GPIOs as you would the GPIO pins on the actual Feather.Also I wanted to add a Lipo battery so the device would have “internal”-ish power. Now I have a unhealthy fear for lipo batteries, go to youtube and search battery venting and you will too. So the pointy solder ends on the bottom side of the FeatherWing Tripler made want to be a bit more safer. So I cut out a thin sheet of plastic to sandwich between the bottom of the Tripler and the battery. MOAR SAFETY.



The Functionality.

So the Featherwing OLED has 3 buttons(technically 4 but one is the RESET button.) I wanted to use the buttons to , initiate a scan, scroll through the results and find the strongest AP signal. So I decided to make the menu as following:



1. Scan

2. Continuous Scan
3. Top Dog SSID

1.) Scan:
Pretty simple , perform a scan and log the findings to the 2 files (OPEN.txt and ALL.txt) and allow me to scroll through the findings.

2.) Continuous Scan
Perform a continuous scan and keep logging to the file untill the RESET button or power switch is triggered.

3.) Just find the strongest SSID (top dog SSID). No logging.

The CODE. (WARNING:I am a shitty coder)

code at : https://github.com/elkentaro/FeatheringSSIDs

The code is pretty straight forward, for the SCAN function just perform a basic wifi scan and write the output to the file. (Same as the NODE-MCU or M5Stack build) . The one change is that I wanted to be able to press a button to scroll through the findings. Since I decided I wanted to use all 3 buttons from the “top” menu, I needed a way to keep track of each button state. So 3 variables for defining each state were added. The CONTINOUS SCAN just scans continuously so pretty much an infinite loop. (Yea , I could have added a “start” , “stop” function but thats maybe for V2 of this gadget)

The new Top Dog function.
So for the new Top Dog function, the idea is to perform a scan and show the

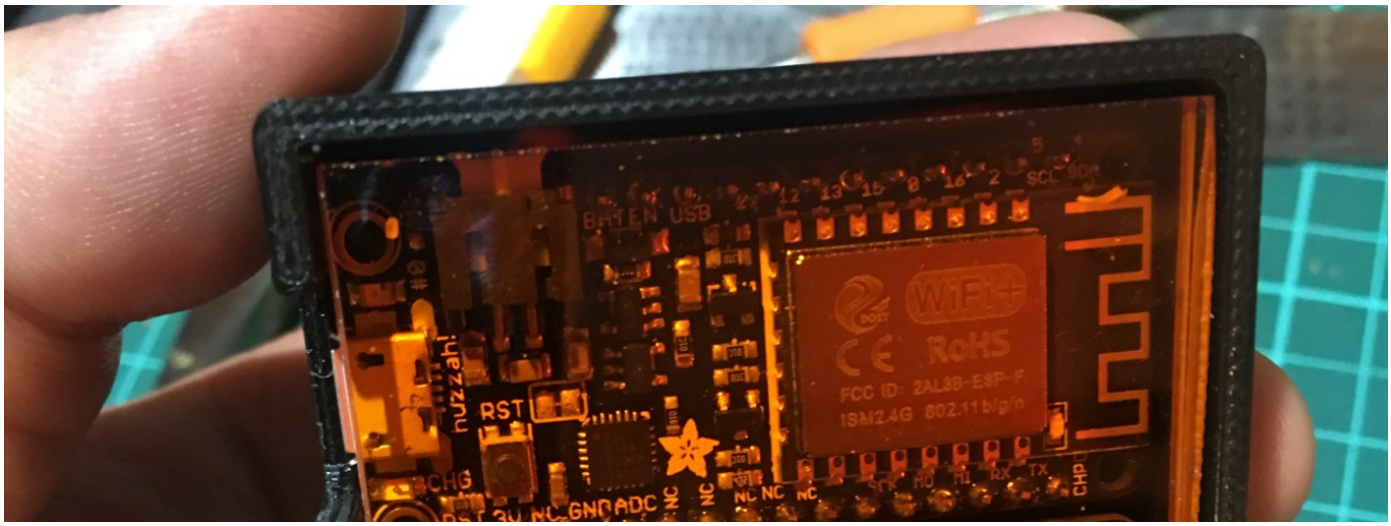
details of the strongest signal. Pretty simple code wise , perform a scan , throw every finding into an Struct Array and go over the Struct Array and compare RSSI signal strength.

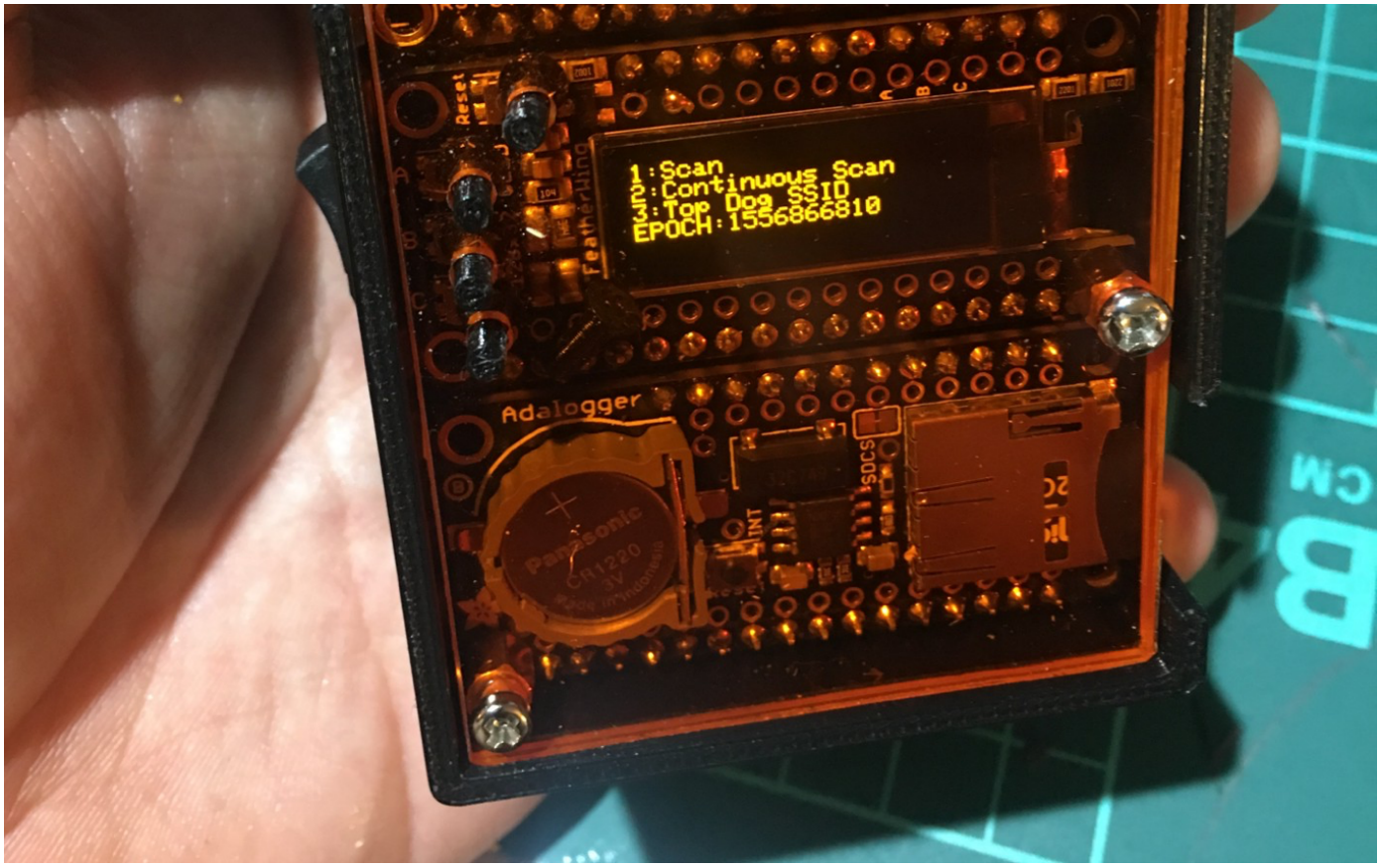
The RTC.

So the Adafruit RTC logger FeatherWing has a unique characteristic, it take the compile time for the start time of the RTC. You could manually set it but i decided to keep it as is since it would suit me better. Now, I travel a lot and and I like to keep the clocks in sync no matter where I was. So I decided to log the data based on UTC time. To do this , I set up my development machine to UTC time before I uploaded the code to my Feather.

The actual build.

Once all the components were hooked up and the code was uploaded, I designed a case for the whole setup. I do like the way the Feather boards look , it has a retro-tech gadget looks to it, I did not want to “hide” all the beauty of the PCB boards, but I did want to protect them from my rough handling and travels. So I decided to encase the setup in a box and give it an acrylic cover, but to enhance the nostalgic mono-chrome look of the OLED I decided to use an orange acrylic cover.





Epilogue.

The most common question I got for this build was , “Does it have GPS?” The answer is a solid “nope” and there is a good reason for this. Unless you have done actual GPS development when you think of GPS you think of the GPS on your phone. That GPS isn’t a “pure” GPS. It uses other sources like your wifi signal to pin point where you are. Pure GPS on the other hand needs a clear line of sight to a GPS satellite, this is fine if I was only scanning outside, but I’m not. Once I’m inside a building I would loose the line of sight and if I had used the GPS time for recording I would have to wait for the gadget to sync up outside prior to entering a facility. For collecting data and uploading it to Wiggle for example, I would need GPS data. But I have another tool for this type of need. My Android phone. This gadget is for me to purely collect SSIDs and log them to 2 files for my SSID hoarding needs.

You could use the collected SSIDs with the awesome work by the @sensepost team's research around host-apd and mana or any of the other cool wifi projects out there.

What's next?

So I do plan to "upgrade" the code so I can incorporate the "open or trap" test the M5 stack build does and some other functions (maybe a deauther or beacon flooder or some other shenanigans code) . However for now I need to clean up my workshop which is in its typical "Here goes Kentaro again, building weird shit trashing the shit out of the workshop" state.

So I ended up building a PoC version using the Esp32 Huzzah Feather from adafruit industries that has GPS cause people asked for it. Also for a more practical version I've build a vertically stacked version.

https://www.instagram.com/p/BxlZNcQDafi/?igshid=evkn6n0tkwr4

Happy fishing and stay legal.

Arduino Adafruit Wifi Hacking