Assembly Guide And User Manual

ResQ

Open Source Hardware

www.mkme.org

Revision Sheet

Release No.	Date	Revision Description
Rev. 0	07/01/2020	Initial Manual Assembly
Rev 1.0	08/25/2020	Sentry and Camera Modules
Rev 1.1	09/25/2020	Bluetooth/Radio Integration

USER'S MANUAL

TABLE OF CONTENTS

Page #

A. GENERAL INFORMATION ***A-1***

1.1 System Overview **A-1**

1.2 Project References **A-2**

1.3 Authorized Use **A-2**

1.4 Points of Contact **A-2**

1.4.1 Information A-2

1.4.2 Collaboration A-2

1.4.3 Help A-2

1.5 Organization of the Manual **A-2**

1.6 Acronyms and Abbreviations **A-3**

B. SYSTEM SUMMARY ***B-1***

2.1 System Configuration **B-1**

2.2 Schematic **B-1**

2.3 User Access **B-1**

C. GETTING STARTED ***C-1***

3.1 Assembly **C-1**

3.2 Systems **C-1**

3.2.1 Main Screen C-1

3.3 Serial Data Output **C-1**

3.4 System Shutdown **C-1**

D. USING the SYSTEM ***D-1***

4.1 Reviewing Data on Map (online) **D-1**

4.1.1 [x] D-1

4.2 Special Instructions **D-1**

4.3 Bluetooth Module **D-1**

4.4 Camera Module **D-1**

4.5 ResQ Sentry **D-1**

E. Appendix ***D-1***

1.0 GENERAL INFORMATION

NOTE: *highlighted, italicized text throughout this template is provided solely as background information to assist you as Tips and Tricks*

The User's Manual provides the information necessary for the user to effectively use the ResQ project

A. GENERAL INFORMATION

1.1 System Overview

- Detects all Beacons and their MAC addresses within range
- Detects All AP's in range (not needed but log them anyhow)
- Gets GPS Lat/Long positioning
- Gets GPS UTC Time
- Logs all the above to the SD card .csv file every time any device detected changes
- Displays the number of clients detected, APs detected, time and Lat/Long on the OLED screen live.
- Send Telemetry to ground station via LORA or locally via Bluetooth to a phone or tablet
- Trigger external camera module to capture image (useful for aircraft/drone based searches)

MODULES

RESQ AIR

ResQ air can be carried by aircraft drone or UAV to autonomously inspect any area for human presence. Targets, their GPS location, Time and signal strength are stored to the internal SD card and also transmitted over long-range (LORA) back to the base station or short-range via Bluetooth to a phone or tablet.

RESQ GROUND

ResQ ground is the exact same hardware but coupled to a high-gain directional antenna with internal power supply. This enables it to be carried by ground personnel to finely tune the search area and direct them to the target

RESQ CAM

ResQ Cam is a small and cheap ESP32 based module that connects directly to ResQ and captures still images to an SD card any time a new target is detected. This allows search teams to see the EXACT area that a target was detected in a full color image captured from the air.

RESQ SENTRY

ResQ Sentry is simply an Air or Ground unit coupled with a ResQ Camera module in a special enclosure. This is mounted in game preserves or other restricted areas. Sentry Captures an image any time a human enters the area but is completely immune to false triggers from animals etc. It captures all the same data as above and can relay it back to a base station via Long Range radio.

1.2 Project References

Initial design completed fall 2018. Testing and updates ongoing via Hackaday website:

https://hackaday.io/project/172090-resq-search-and-rescue-tools

Main repo for all files and updated instructions: https://github.com/MKme/ResQ

Buy parts/kits from our store here: http://store.mkme.org/

1.3 Authorized Use Permission

This work is licensed under a Creative Commons Attribution-Noncommercial 4.0 International License. https://creativecommons.org/licenses/by-nc/4.0/ See notes.h and comments in main code for detailed lineage/license info

1.4 Points of Contact

See below

1.4.1 Information

Support provided via the forum at http://mkme.org/forum as well as the GIT repository: https://github.com/MKme/ResQ

Support for the libraries used to support the code is provided by the original authors only. Restrictions for use can be found on their respective support pages (linked within the Arduino Source Code and reference materials

1.4.2 Collaboration

You are welcome to join our project! Submit your ideas, code, improvements corrections or anything else. Open source is for everyone to enjoy.

1.4.3 Help

ResQ Tool Forum Thread For Discussion - http://mkme.org/forum/viewtopic.php?f=28&t=1130

Live chat 24/7 on our dedicated Discord server: http://mkme.org/discord

1.5 Organization of the Manual

See index.

1.6 Acronyms and Abbreviations

I2C **I²C** (**Inter-Integrated Circuit**), pronounced *I-squared-C*, is a synchronous, multi-master, multi-slave, packet switched, single-ended, serial communication bus invented in 1982 by Philips Semiconductor (now NXP Semiconductors). It is widely used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication. Alternatively, I²C is spelled **I2C** (pronounced I-two-C) or **IIC** (pronounced I-I-C).

WIFI **Wi-Fi** (/ˈwaɪfaɪ/)[11] is a family of wireless network protocols, based on the IEEE 802.11 family of standards, which are commonly used for local area networking of devices and Internet access. *Wi-Fi* is a trademark of the non-profit Wi-Fi Alliance, which restricts the use of the term *Wi-Fi Certified* to products that successfully complete interoperability certification testing.[2][3][4] As of 2010, the Wi-Fi Alliance consisted of more than 375 companies from around the world.[5] As of 2009, Wi-Fi-integrated circuit chips shipped approximately 580 million units yearly.[6][needs update] Devices that can use Wi-Fi technologies include desktops and laptops, smartphones and tablets, smart TVs, printers, digital audio players, digital cameras, cars, and drones.

GPS The **Global Positioning System** (**GPS**), originally **NAVSTAR GPS**,[1] is a satellite-based radionavigation system owned by the United States government and operated by the United States Space Force.[2] It is one of the global navigation satellite systems (GNSS) that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.[3] Obstacles such as mountains and buildings block the relatively weak GPS signals.

Software Serial Arduino hardware has built-in support for serial communication on pins 0 and 1 (which also goes to the computer via the USB connection). The native serial support happens via a piece of hardware (built into the chip) called a UART. This hardware allows the Atmega chip to receive serial communication even while working on other tasks, as long as there room in the 64 byte serial buffer.

YAGI A **Yagi–Uda antenna**, commonly known as a **Yagi antenna**, is a directional antenna consisting of multiple parallel elements in a line,[1] usually half-wave dipoles made of metal rods.[2] Yagi–Uda antennas consist of a single driven element connected to the transmitter or receiver with a transmission line, and additional "parasitic elements" which are not connected to the transmitter or receiver: a so-called *reflector* and one or more *directors*.[2][3][4] It was invented in 1926 by Shintaro Uda of Tohoku Imperial University, Japan,[5] with a lesser role played by his colleague Hidetsugu Yagi.[5][6]

Dipole In radio and telecommunications a **dipole antenna** or **doublet**[1] is the simplest and most widely used class of antenna.[2][3] The dipole is any one of a class of antennas producing a radiation pattern approximating that of an elementary electric dipole with a radiating structure supporting a line current so energized that the current has only one node at each end.[4] A dipole antenna commonly consists of two identical conductive elements[5] such as metal wires or rods.[3][6][7] The driving current from the transmitter is applied, or for receiving antennas the output signal to the receiver is taken, between the two halves of the antenna. Each side of the feedline to the transmitter or receiver is connected to one of the conductors. This contrasts with a monopole antenna, which consists of a single rod or conductor with one side of the feedline connected to it, and the other side connected to some type of ground.[8] A common example of a dipole is the "rabbit ears" television antenna found on broadcast television sets.

SPI The **Serial Peripheral Interface** (**SPI**) is a synchronous serial communication interface specification used for short-distance communication, primarily in embedded systems. The interface was developed by Motorola in the mid-1980s and has become a *de facto* standard. Typical applications include Secure Digital cards and liquid crystal displays.

SD **Secure Digital**, officially abbreviated as **SD**, is a proprietary non-volatile memory card format developed by the SD Association (SDA) for use in portable devices.

2.0 SYSTEM SUMMARY

B. SYSTEM SUMMARY

2.1 System Configuration

This hardware can be embedded in to any very small drone or RC aircraft. It can also be attached to existing manned aircraft already participating in the search.

Using small drone aircraft (like the Nano Talon I use) gives all the advantages such as flight condition tolerance, height, terrain, time of day etc that drones have over manned aircraft. They

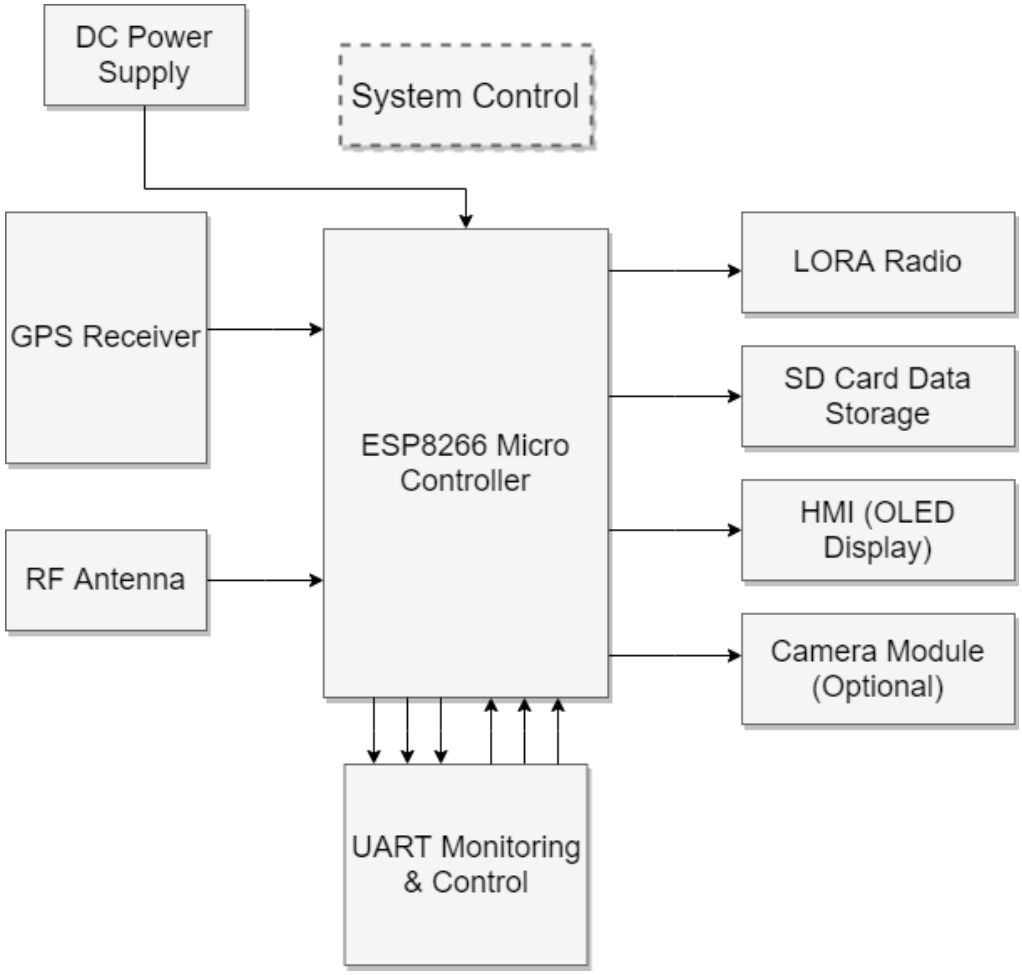
can also be sent on the full way-point mission without human intervention and tracked live from any ground station.

What it does (both air unit **and** ground unit):

- ***Detects all Beacons and their mac addresses within range***
- ***Detects All AP's in range (not needed but why not log them anyhow)***
- ***Gets GPS Lat/Long positioning***
- ***Gets GPS UTC Time***
- ***Logs all the above to the SD card .csv file every time any device detected changes***
- ***Displays the number of clients detected, APs detected, time and Lat/Long on the OLED screen live.***

Air unit will eventually send telemetry to ground control for live alerts (likely using existing 433MHz telemetry already sending flight data or LORA radio link)

Flow:



Build History:

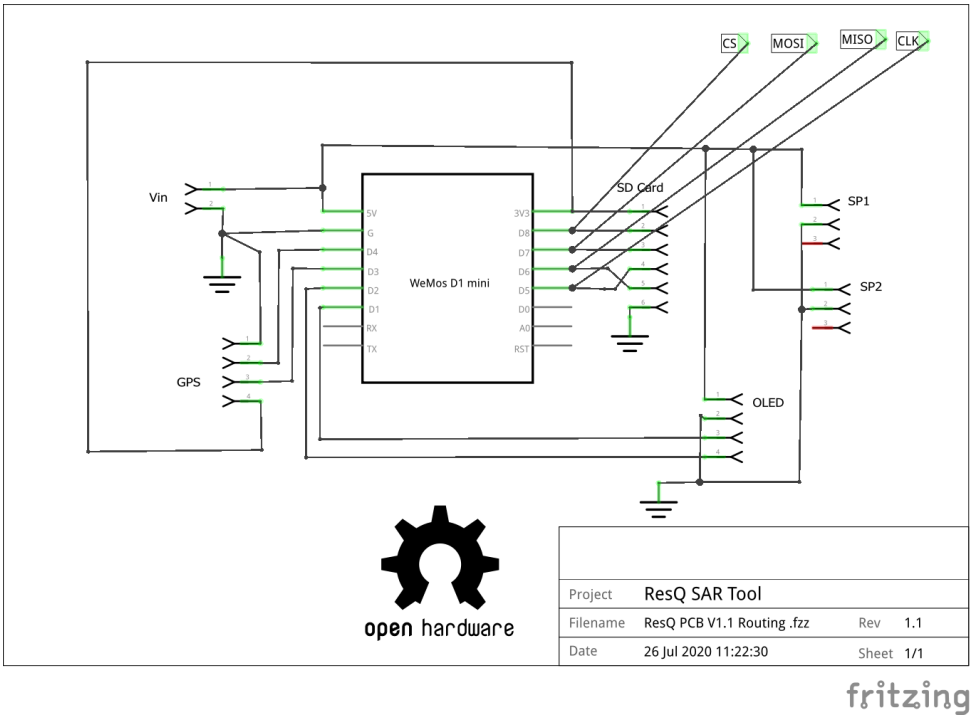
Using the Wemos D1 mini PRO with an external antenna (YAGI or Rubber duck) to receive the beacon frames, log them to SD Card with RSSI values (adds more precise direction finding if you fly a grid pattern) to review the MAC address list as well as plot with Google Earth/maps. I can easily transmit this via LORA or NRF module to the ground instantly as well. This will be logged as a separate version of the hardware.

Videos:

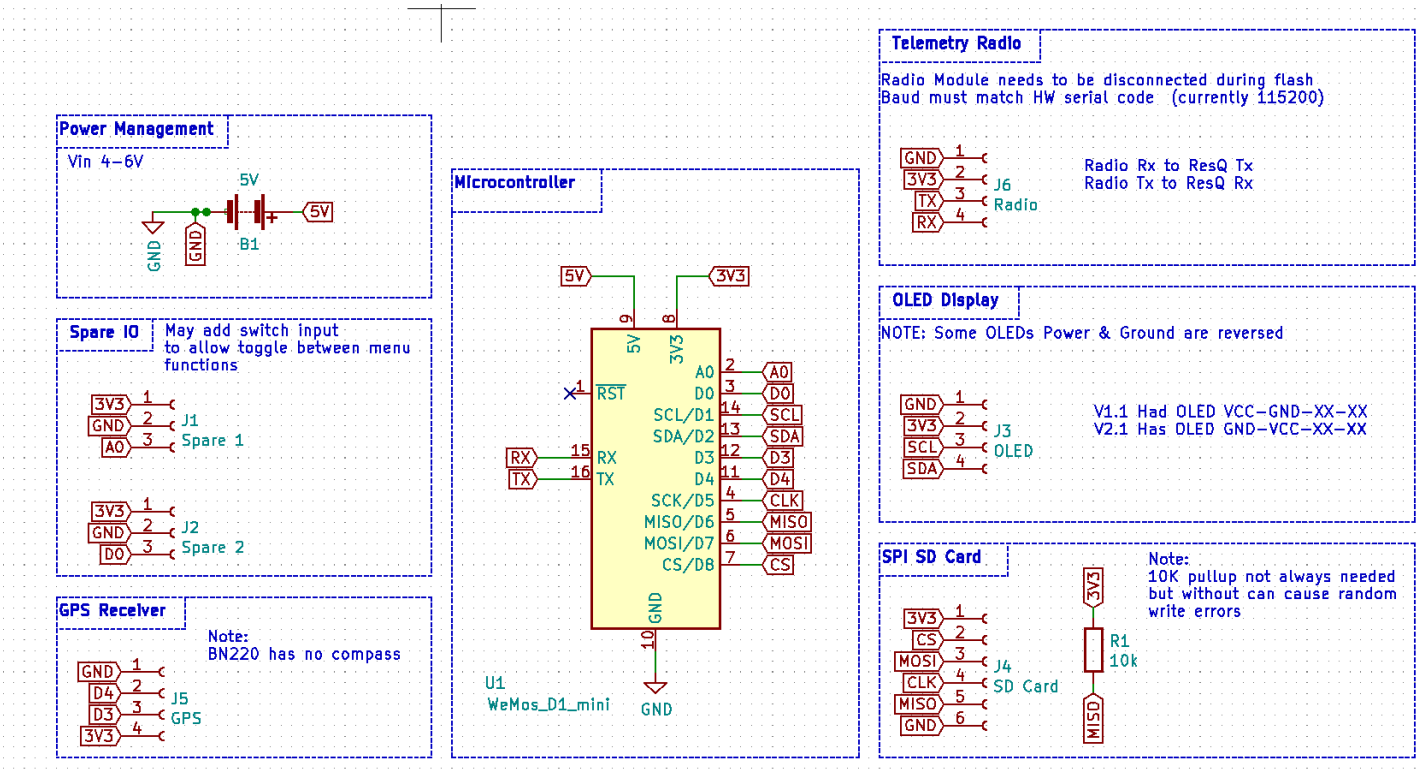
https://www.youtube.com/playlist?list=PLxyM2a_cfnzifVreG8aKurH-9nNYl8QLr

2.2 Schematics

V1.x Board:



V2.x Board:



2.3 User Access

No user access control is currently implemented

3.0 GETTING STARTED

C. GETTING STARTED

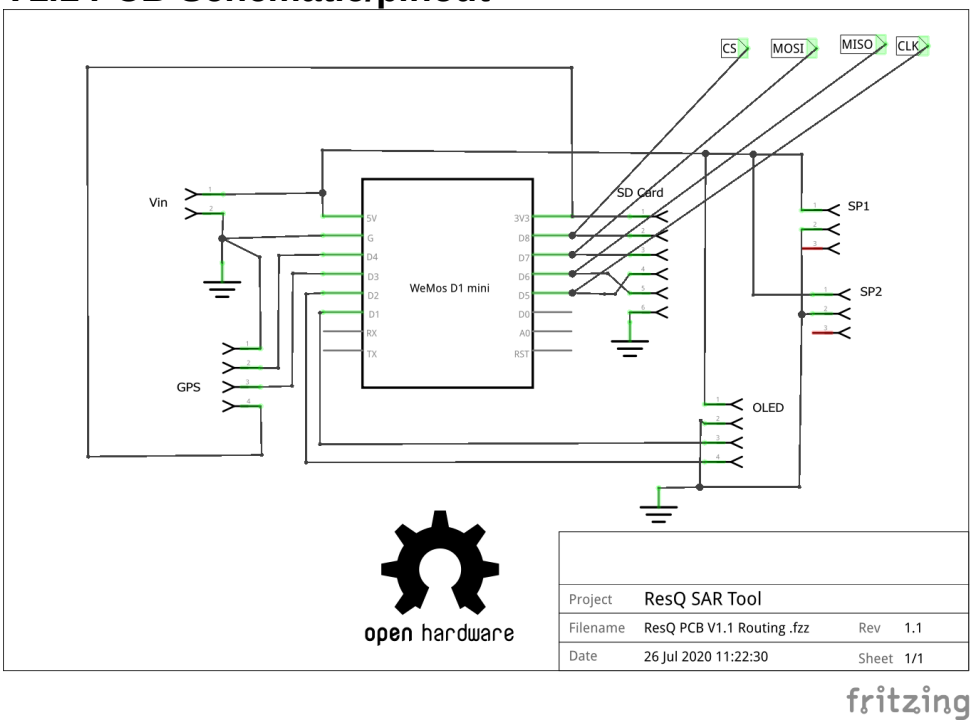
This section provides a general walkthrough from assembling the PCB and enclosures to usage of the system during a search/rescue operation.

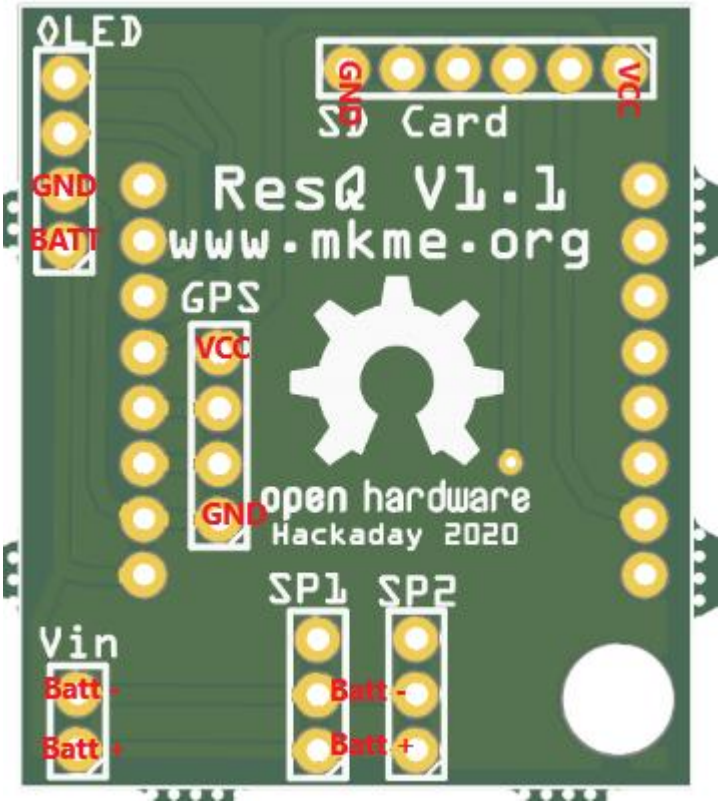
3.1 Assembly

i. If you haven't already got a PCB or components you can order a kit here: http://store.mkme.org/product-category/custom/

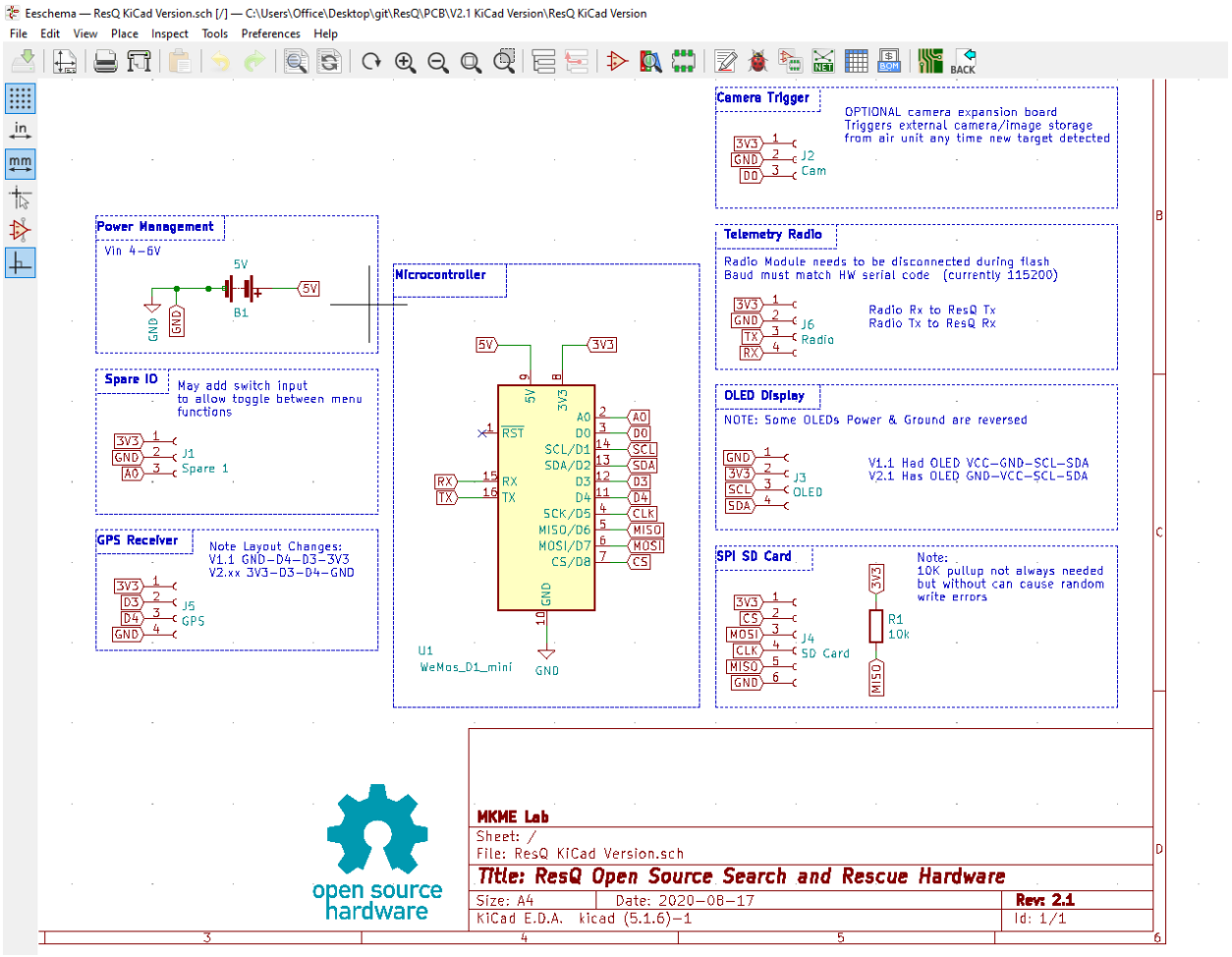
You can order the PCB by itself here as well (link updated as multiple versions available): http://store.mkme.org/product-category/custom/

V1.1 PCB Schematic/pinout

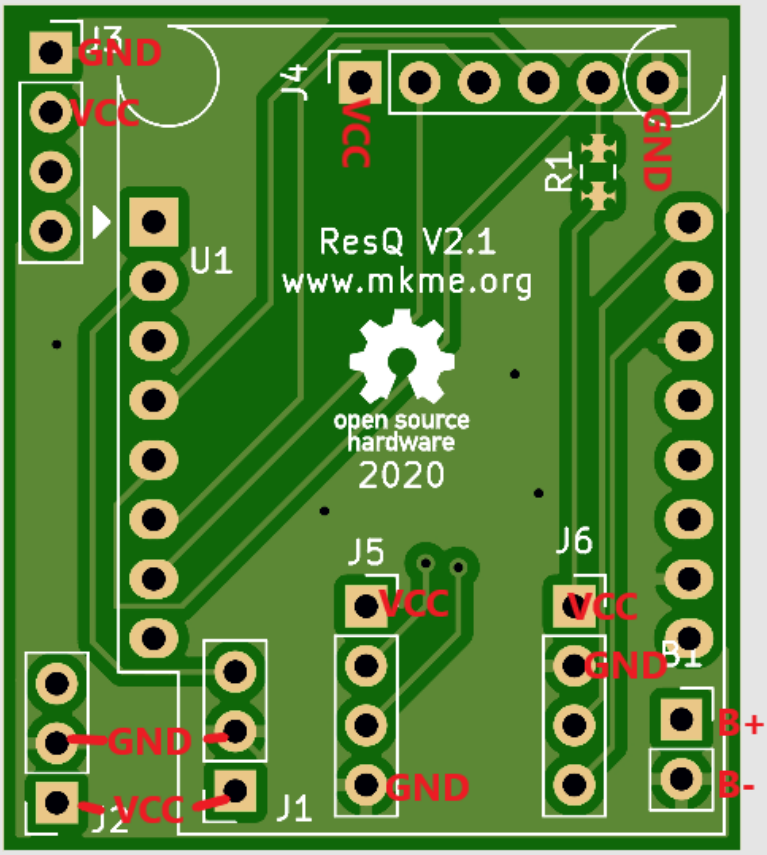




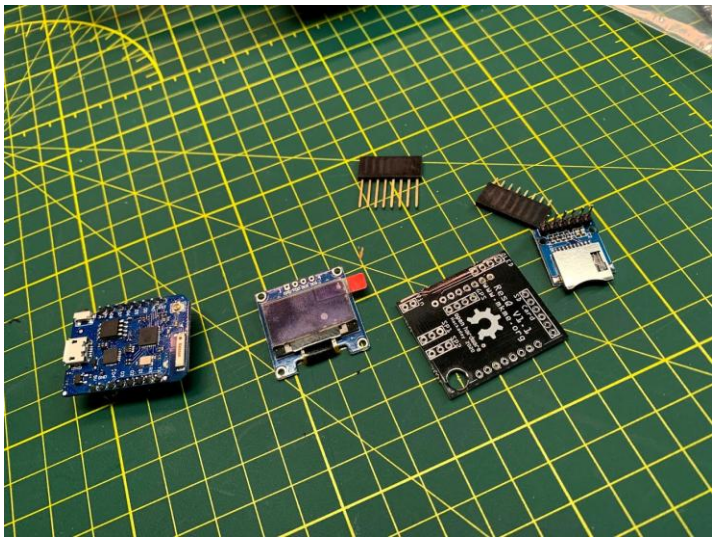
V2.1 PCB Schematic/pinout



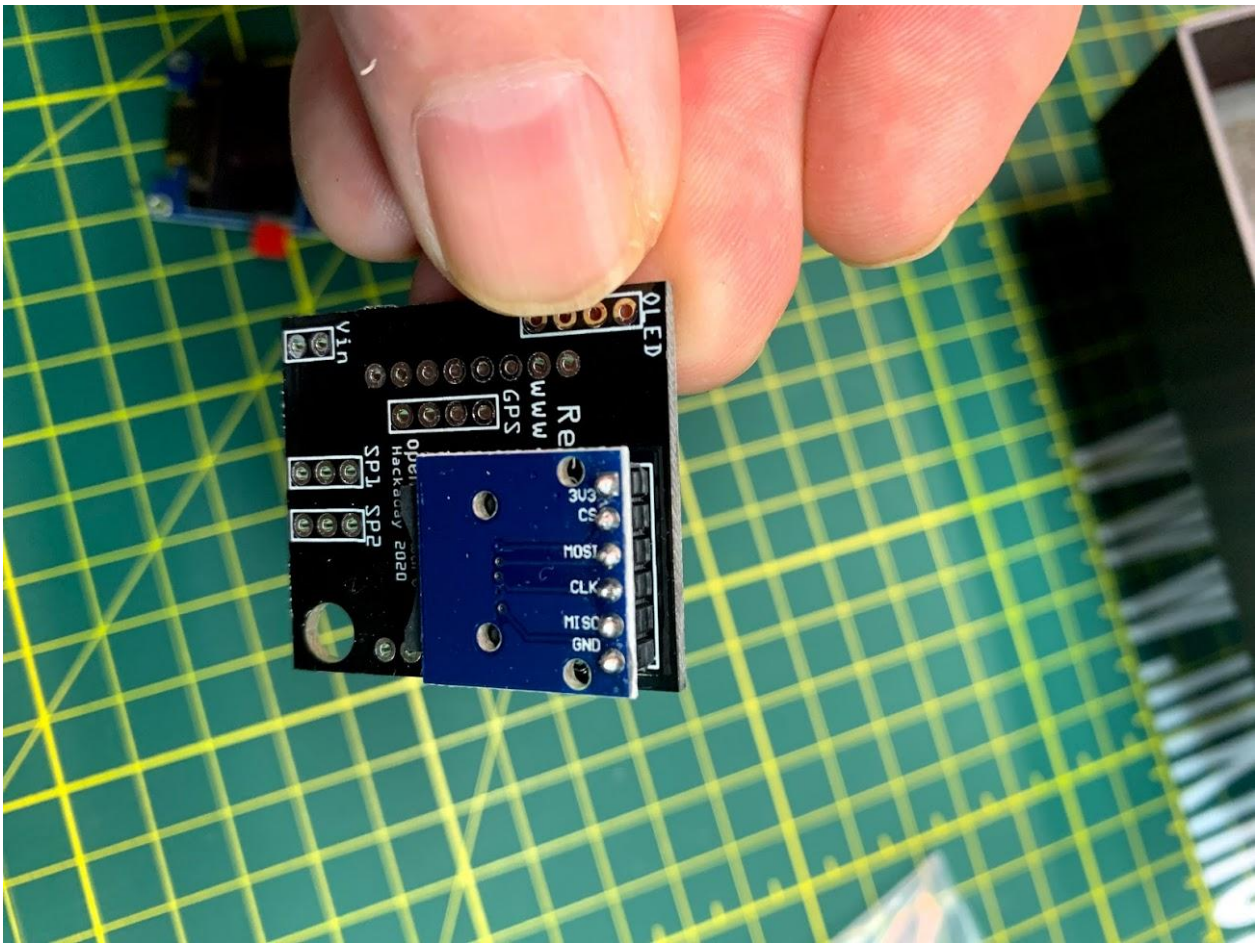
V2.x Power and ground locations



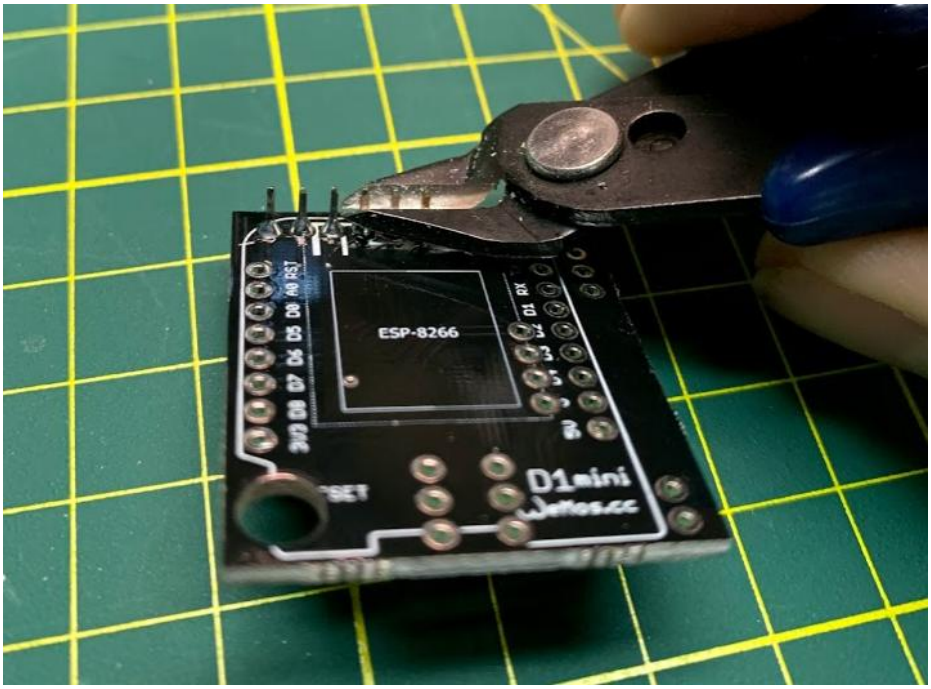
ii. Lay out your components and double check everything matches the schematic (pin order on the SD card and OLED)



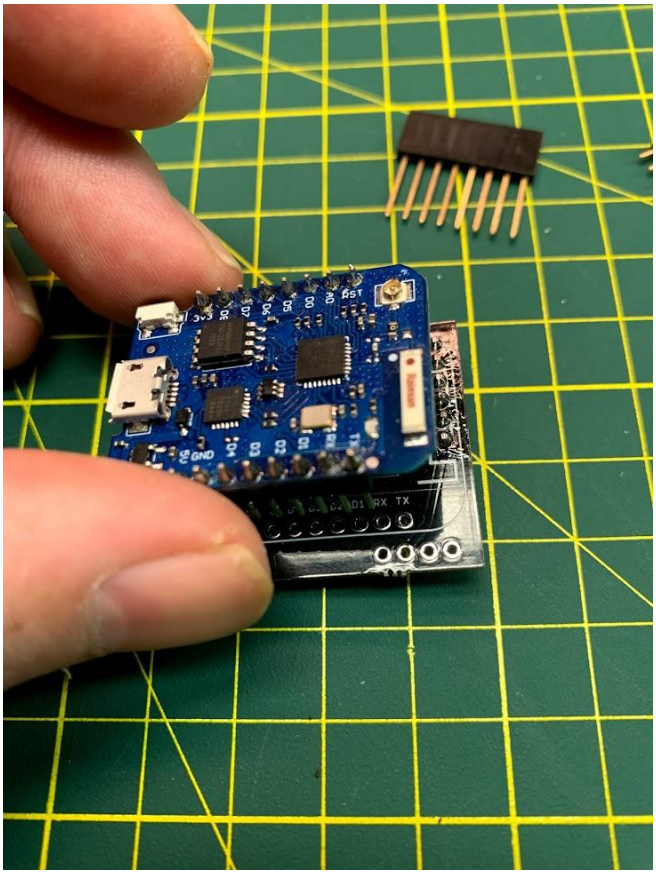
iii. Solder the SD card slot to the board as shown



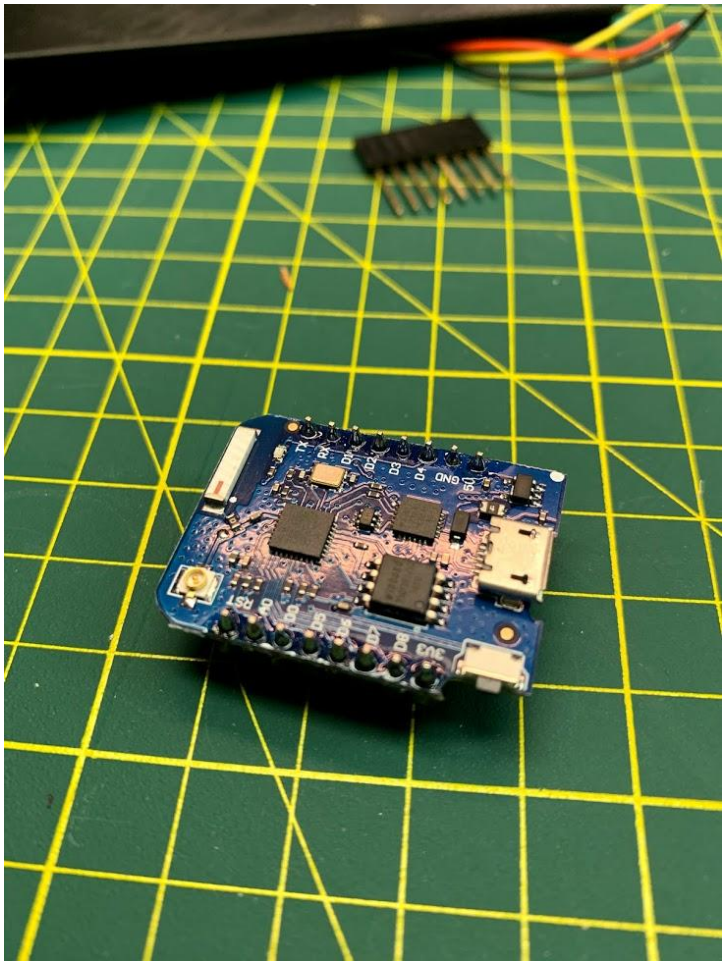
Cut the pins down as close to the board as you can to avoid clearance issues later



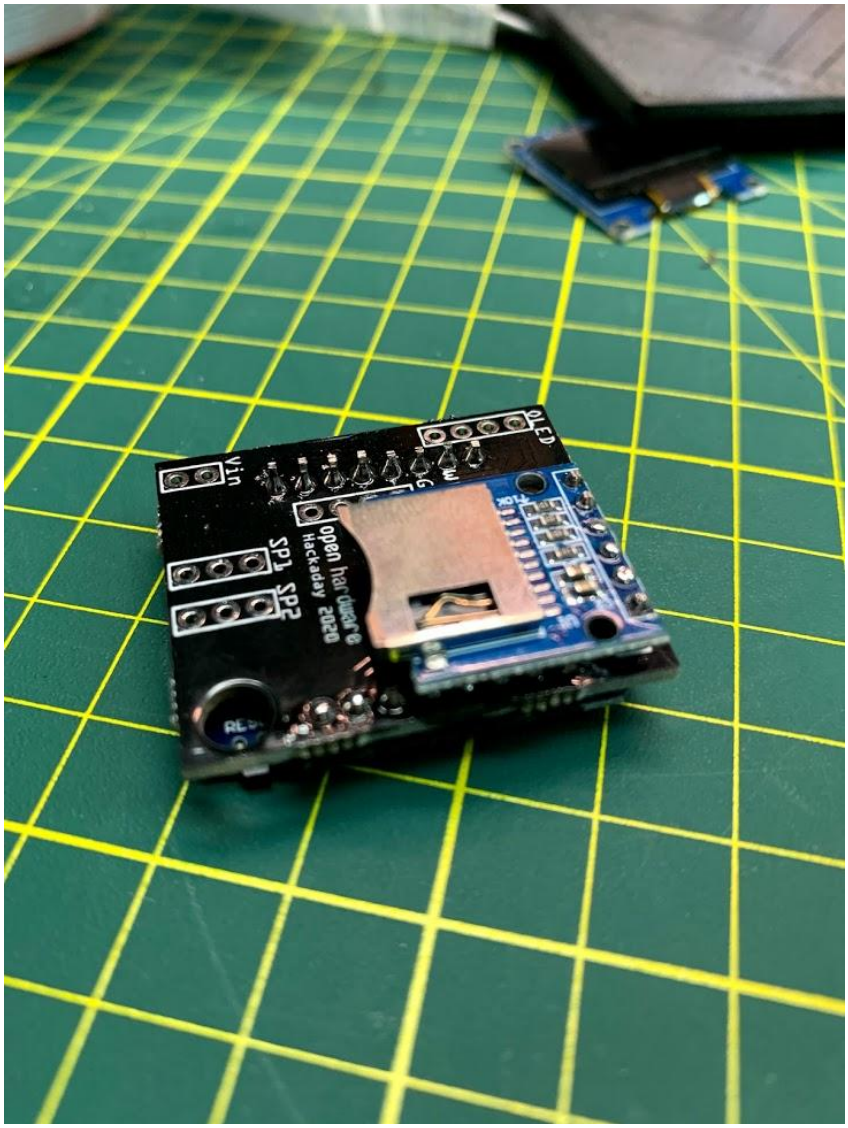
iv. Test fit your WEMOS D1 to the board to ensure fit



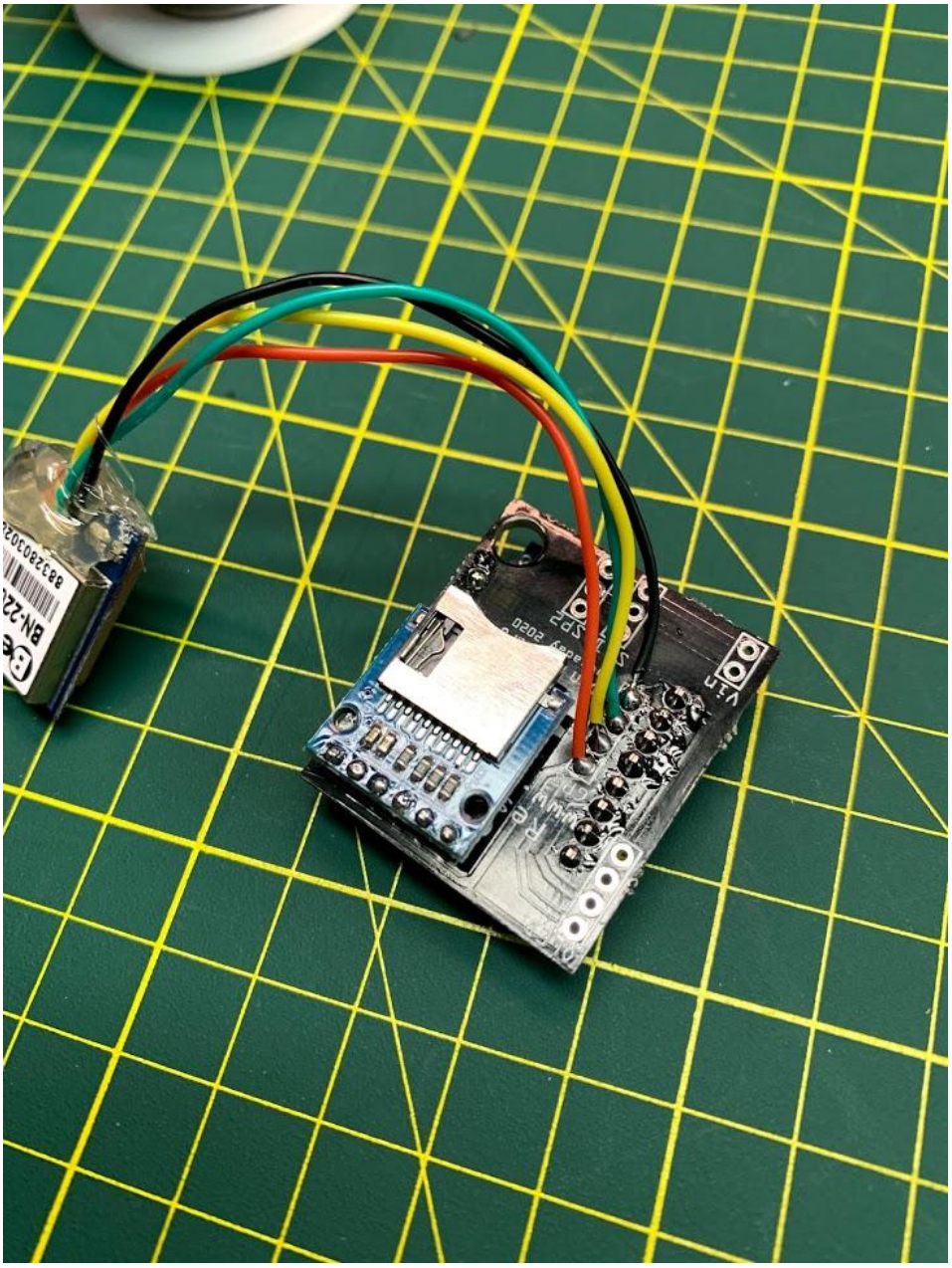
Then solder it in place:



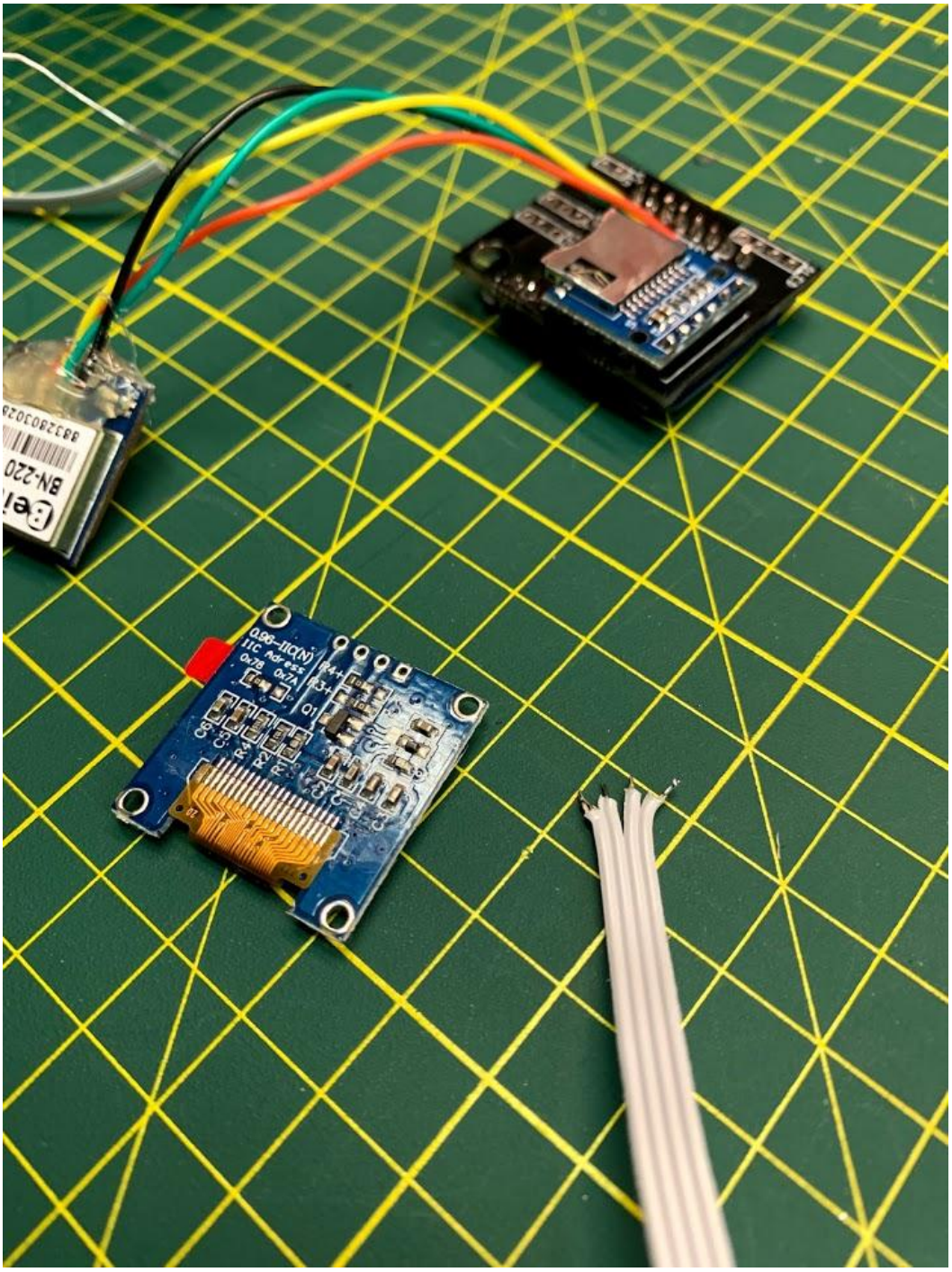
v. You should now have something like this:



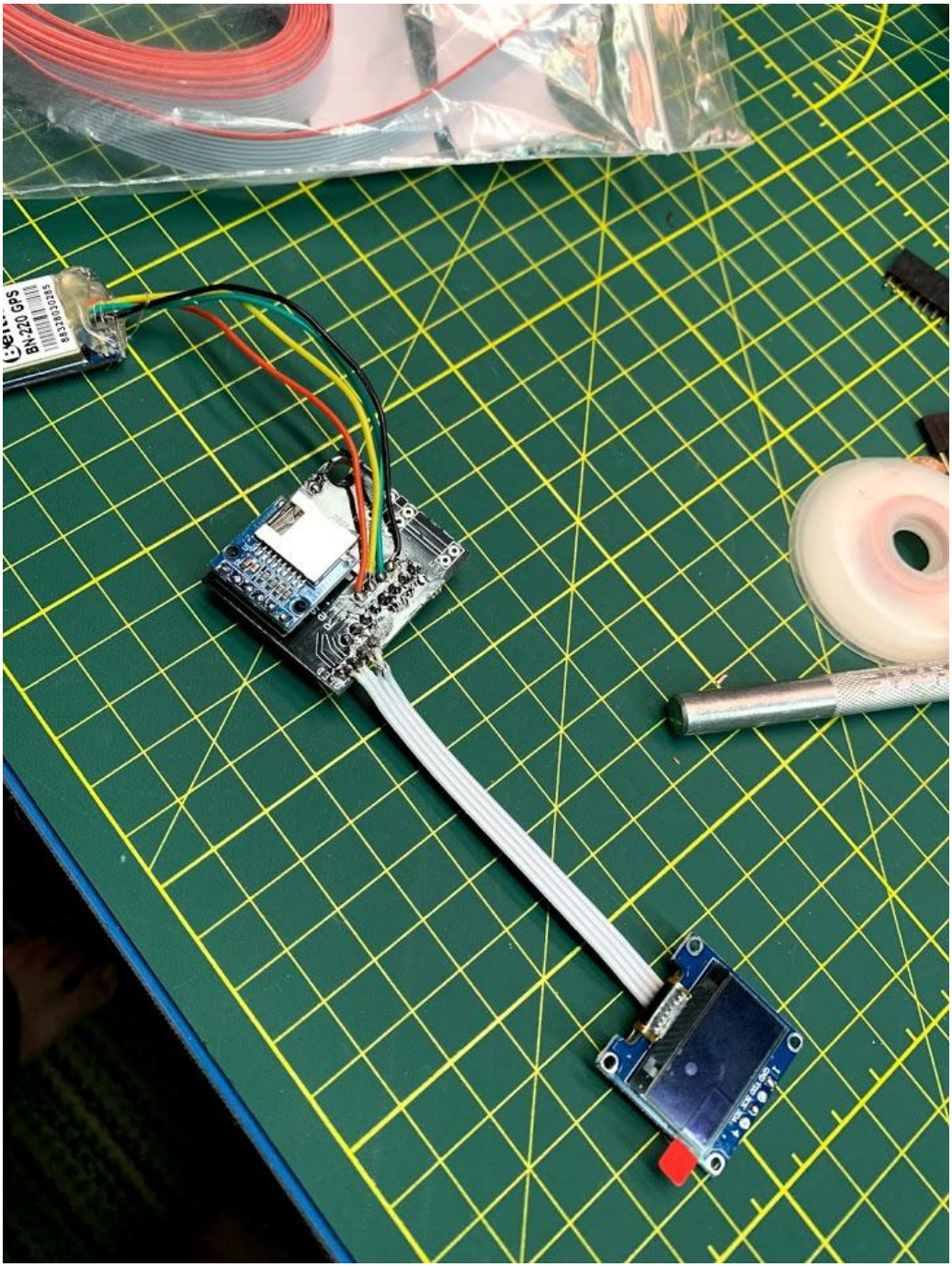
Now attach your GPS wires to the appropriate pads as laid out in the code and schematic:



vi. Prepare your OLED for assembly. You can use headers, wires or ribbon cable. Whatever you like. Here we will use simple ribbon. Note the pinout on V1 PCBs is different that V2.xx PCBs. Ensure you wire in the correct order as per the note on the schematic, One has VCC on pin one- the other has GROUND on pin 1. Very important you verify which OLED you have and which board you are using!



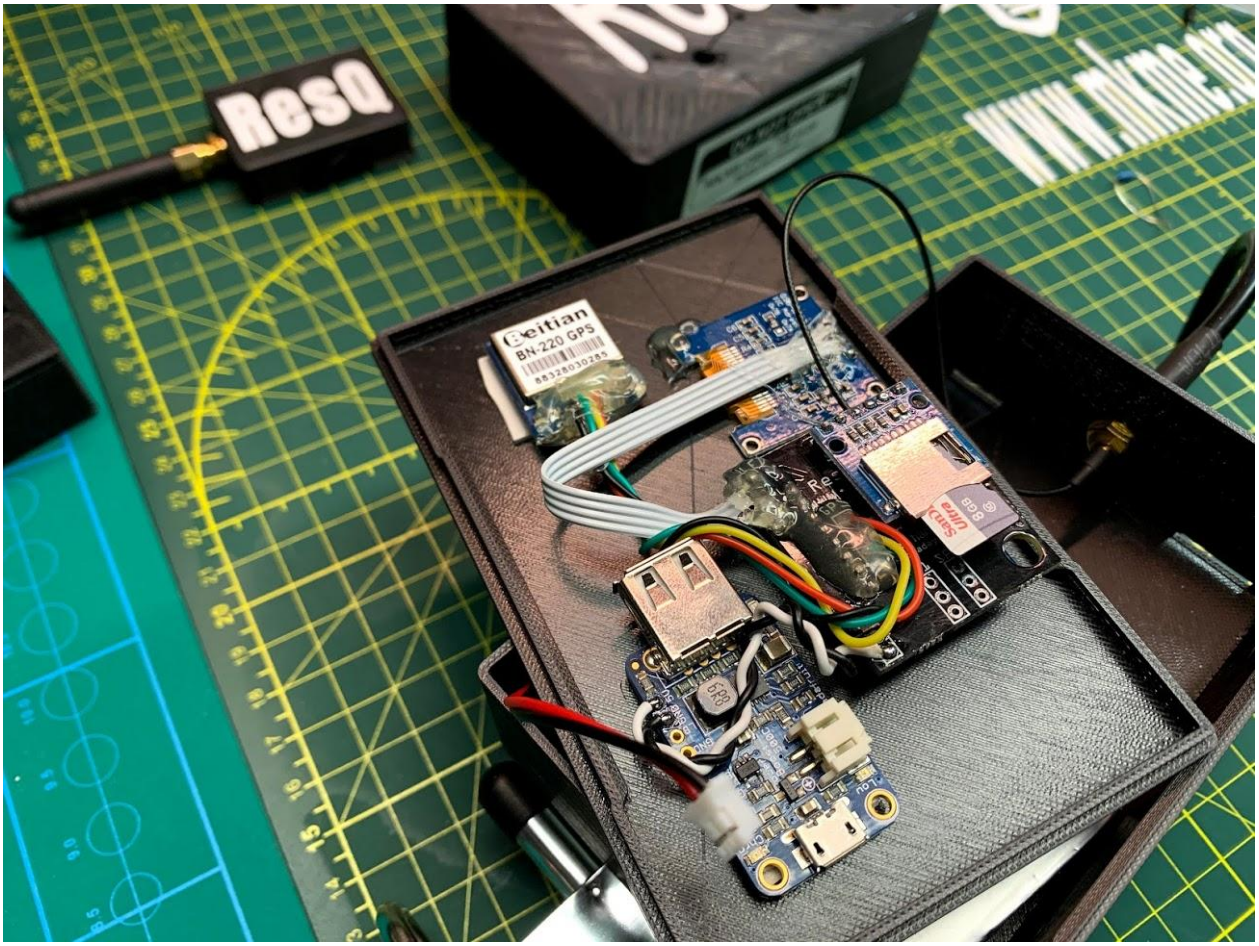
Solder the wires:



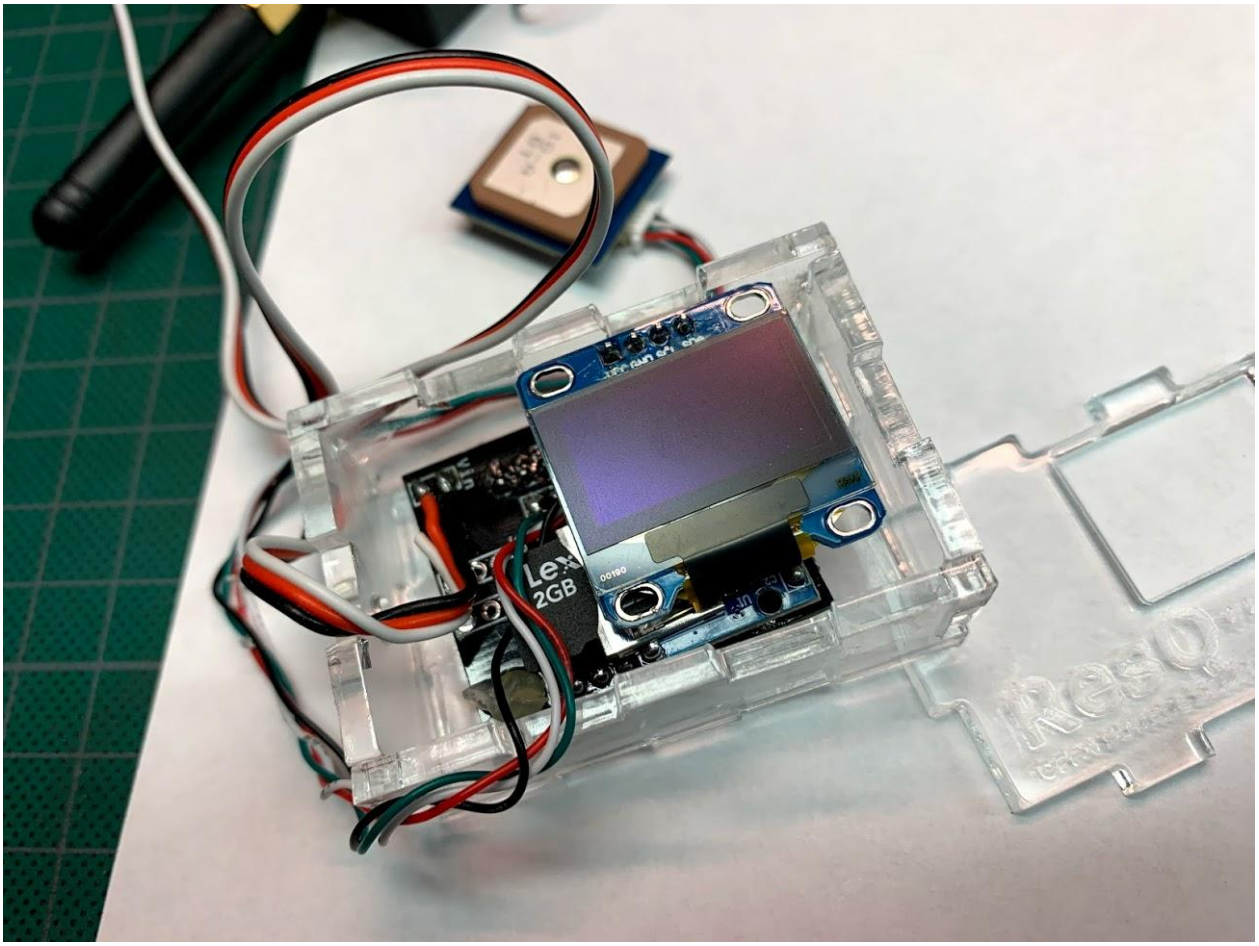
You now have something like this:

vii. Assemble in to the case of your choice. Add some hot glue to provide strain-relief on any of your wires and prevent damage.

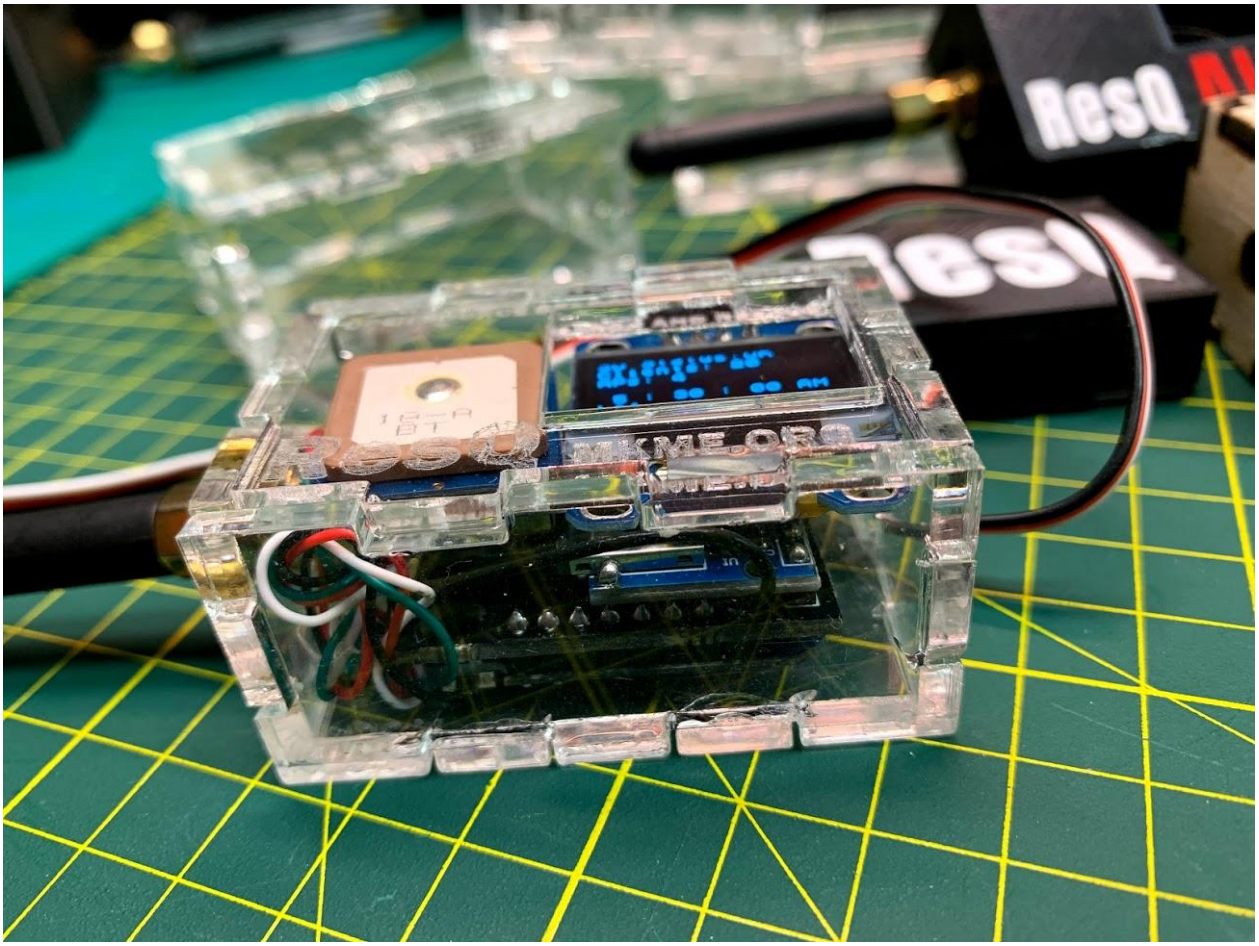
Or like this:



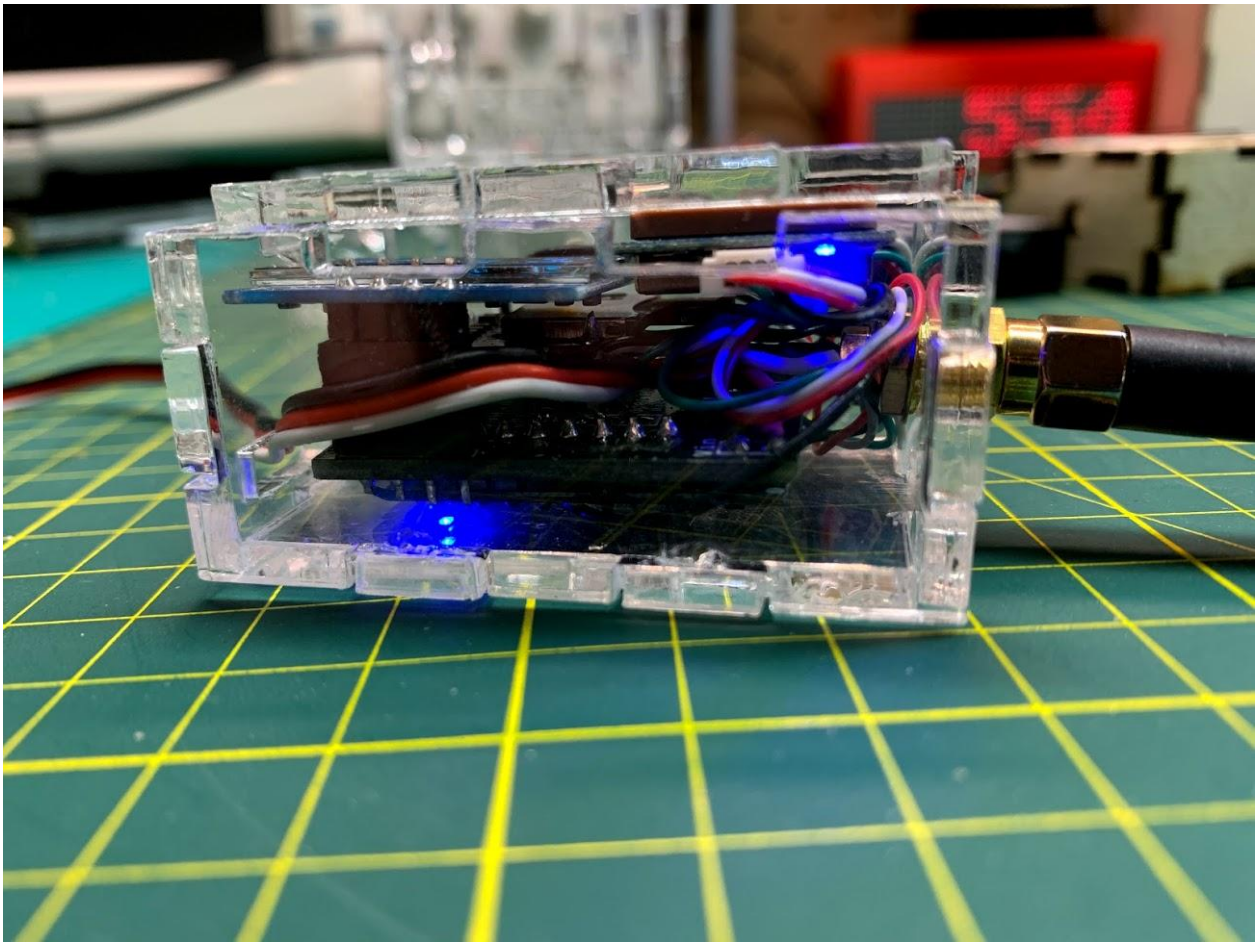
viii. Air unit is exactly the same assembly only a smaller case than shown above:



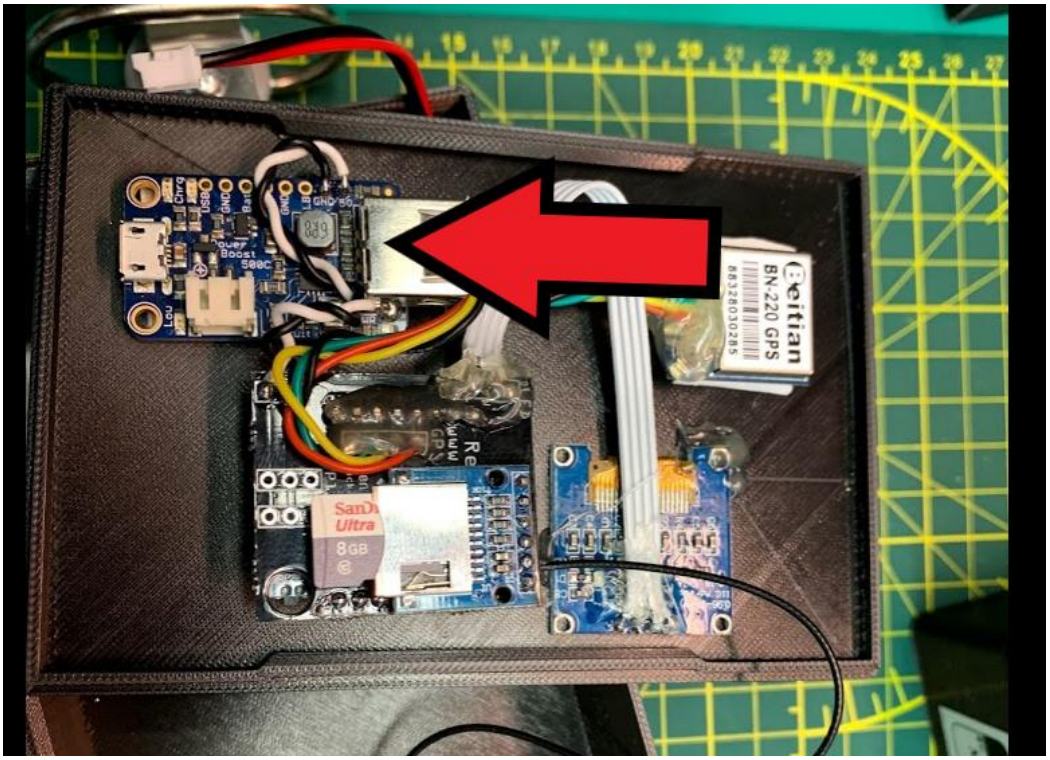
Or



Closer look:



ix. You need to supply 3-6VDC for power on the Batt pins. This can come from the BEC on your Drone/RC plane on the air unit (you can plug right in to your RC receiver) or a dedicated supply battery on the ground unit. Here I use an Adafruit PowerBoost 500:



x. Attach your antenna, insert an SD card and you are ready to program!

xi. To program the ESP8266 you need to install the Arduino IDE, ESP8266 board support within the IDE and the CH340 USB to Serial driver (if not already installed on your PC)

To do this follow the tutorial here:
https://www.youtube.com/watch?v=q2k3CzT5qE0&ab_channel=MakersMadLab

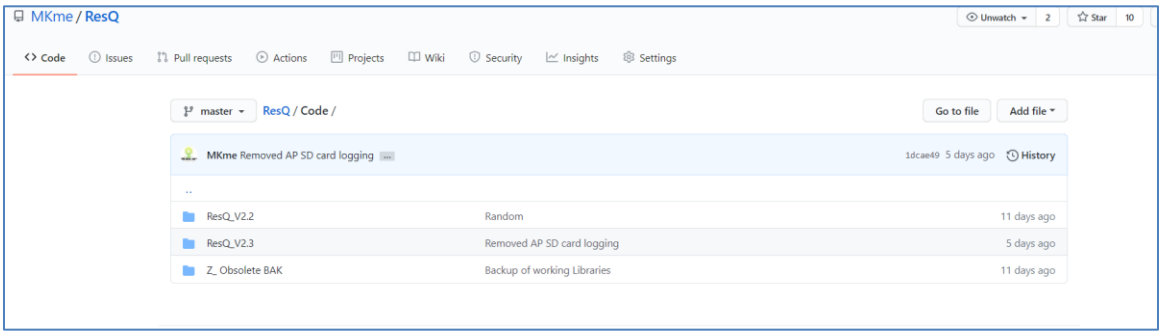
NOTE: ResQ kits come with the ESP8266 programmed with the current ResQ firmware at the time of manufacture.

xii. Install the supporting libraries called out in the sketch (they are also included as backups in the GitHub Repo for you as libraries can be broken by new development. Needed libraries are:

- **ArduinoJson**
- **TinyGPSPlus-1.0.2/TinyGPSPlus-1.0.2**
- **U8g2-2.27.6**
- **EspSoftwareSerial-6.8.1.zip**

For help installing libraries see this video:
https://www.youtube.com/watch?v=M6PZOqNHKxM&ab_channel=Brainy-Bits

xiii. Flash the most current ResQ firmware (sketch) found on GitHub here: https://github.com/MKme/ResQ/tree/master/Code



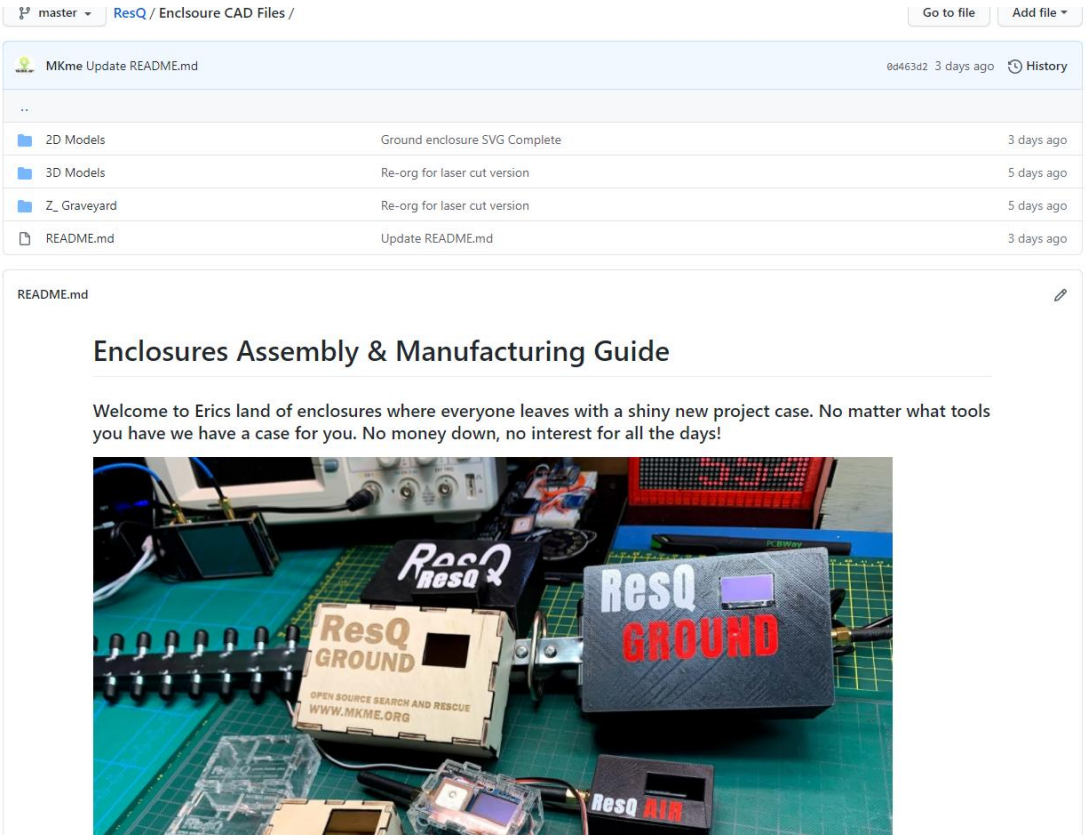
xiv. Your ResQ should boot up with the OLED screen active and code running. **You're done!**

Look what you have made!

xv. Case assembly Notes

If you are 3D printing your own case from the files provided you will have no special assembly. The lids just snap in to place and holes are already included.

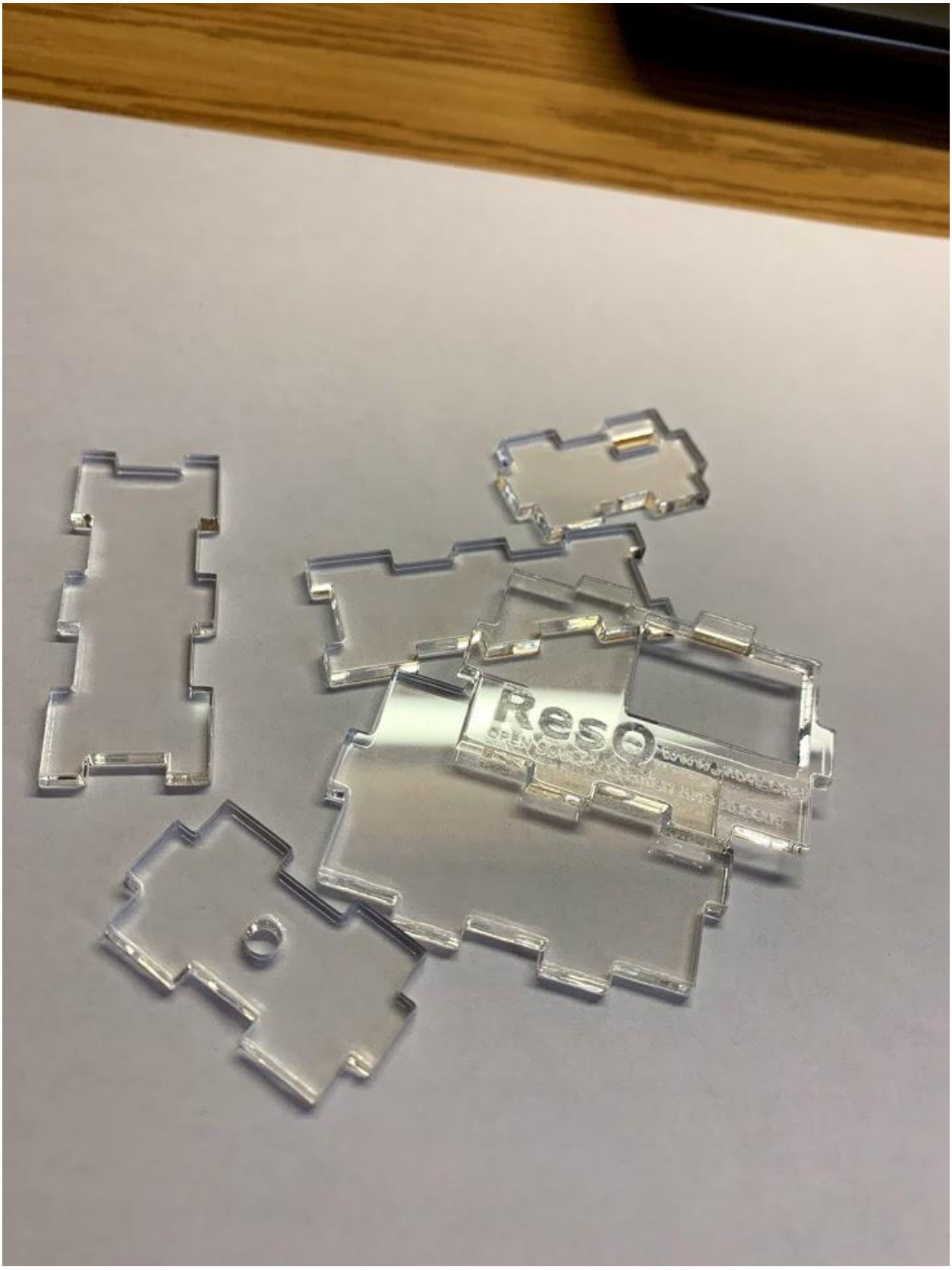
Get the files here: https://github.com/MKme/ResQ/tree/master/Enclsoure%20CAD%20Files



If you are using the 2D files to cut your own (or purchase the case from mkme.org) Laser or otherwise cut the material from materials. Materials with the best performance/cost ratio are:

- **3mm Birch plywood**
- **3mm clear acrylic**

You will be left with components like these:


ResQ

Assemble them with CA (super) glue or adhesive of your choice to resemble these:

Or the ground unit like this:

3.2 Systems

This section describes in general terms the system operation

3.2.1 Main Menu

The main screen contains some basic data/telemetry

1. **Number of Clients/Beacons detected**
2. **Number of AP's (Access points) detected**
3. **GPS UTC Time**
4. **GPS Latitude and Longitude**
5. **SD Card status**

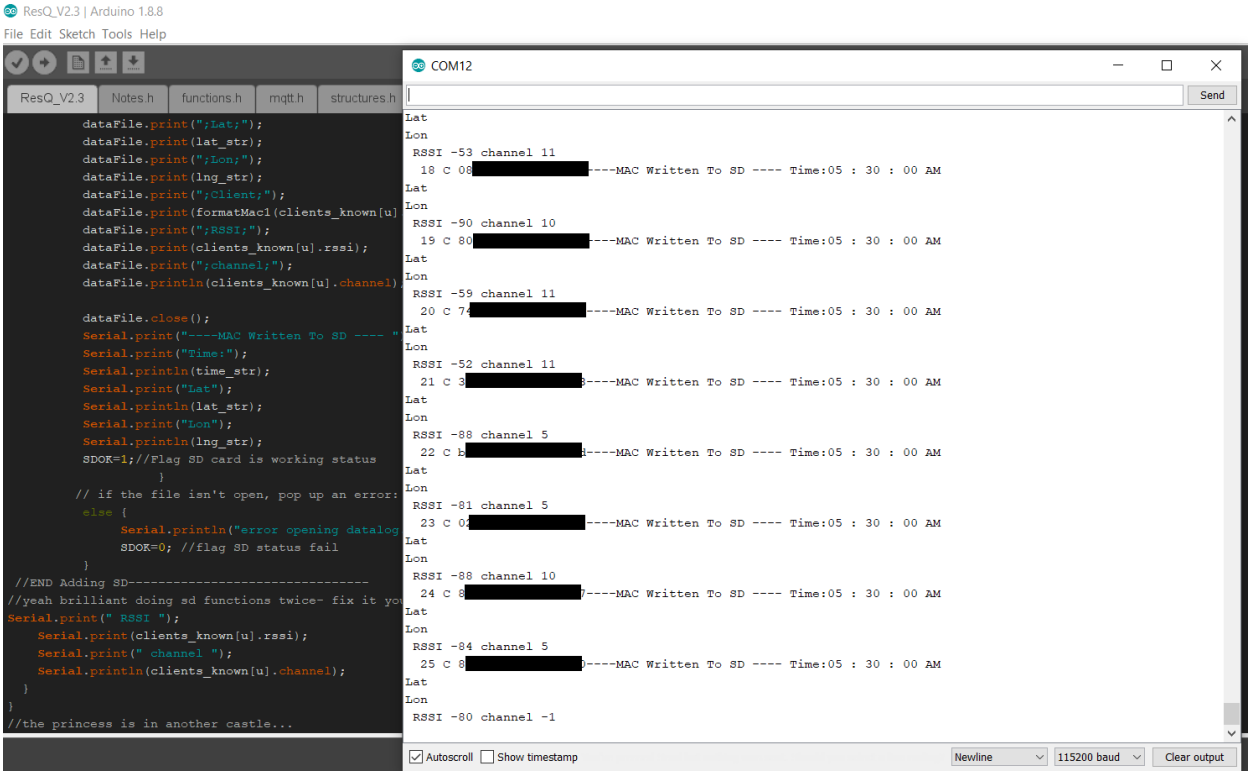
3.3 Serial Data

Serial output is possible from the USB connector or directly from UART pins RX & TX at **115200 baud**.

The pins are broken out on version 2.xx of the custom PCBs for use with LORA radio telemetry. This is not yet fully supported or documented here. To use it you will need 2 LORA radios- one connected to ResQ RX/TX and the other connected to your PC USB either through an Arduino or serial adapter etc. Just read the serial data on the PC and voila- telemetry received.

Serial data output is in plain text and provides all status updates, current AP and client information. RSSI, MAC addresses, GPS location and time, SD card status, SD card interfaces and any other debug info needed

Use serial interface of your choice. Some great options are the Arduino IDE or PUTTY. Both are free.



3.4 System Shutdown

No special shutdown is implemented. Remove power from the unit by disconnecting flight battery (on drone/aircraft version) or by removing battery power on the ground unit (slide switch on power boost enable pins in demo case)

NOTE: A future update really should add some soft shutdown function to avoid depower while SD card is being written. Feel free to contribute this

4.0 USING THE SYSTEM

This section provides a detailed description of the data collected on the SD card.

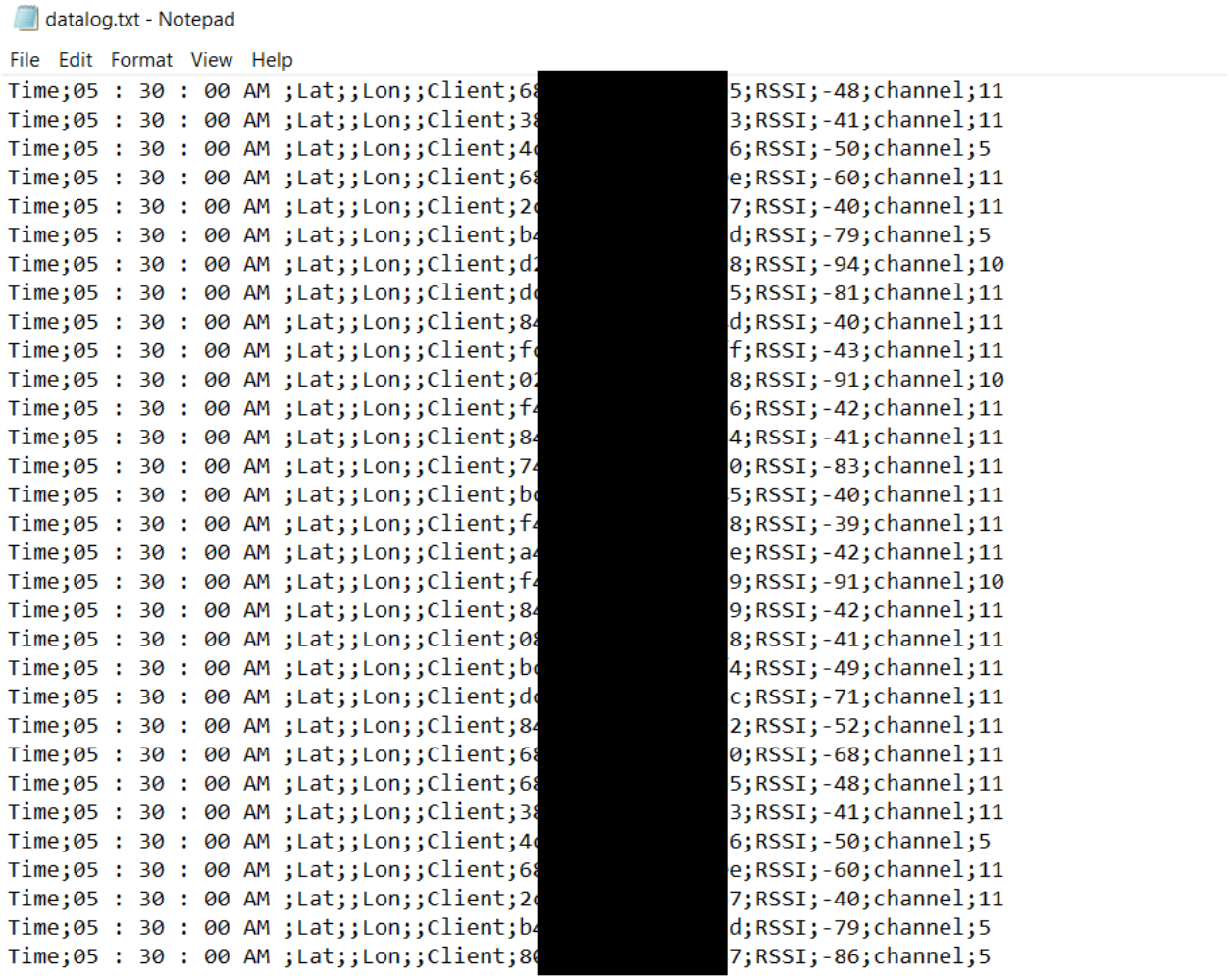
D. USING THE SYSTEM

Once data has been collected during a search event you can review it 2 ways:

1. **In a batch upload to Google Maps (max 2000 rows per file uploaded)**
2. **Manually by plotting the Lat/Long one by one**

4.1 Review Data on MAP (Online)

All data is stored to the SD card while the unit is powered on. Perform your search then afterward- remove the SD card and download the contents to your PC. The file is datalog.txt

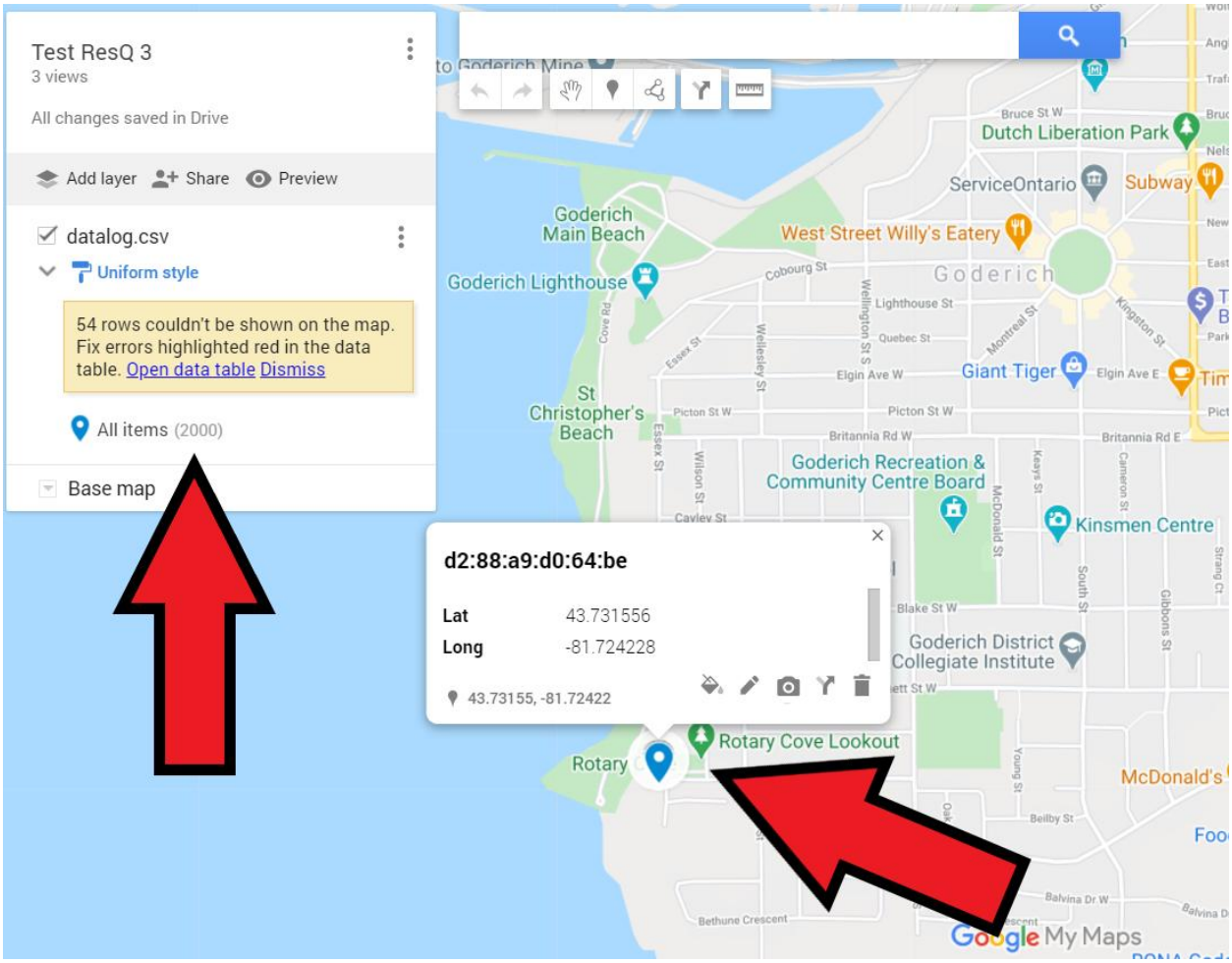


You can manually review the hits and plot them one by one in Google Maps or your preferred map software or you can do them all at once:

1. Download the datalog.txt file,
2. Rename from datalog.txt to datalog.csv
3. Open in excel - Add a row above the data and within that row add the text LAT, LONG and MAC above the corresponding columns for each (this will tell Google what each column is)
4. Delete all other columns as we won't need those
5. Upload to Google maps as done here https://www.youtube.com/watch?v=SLMzhOoA29M

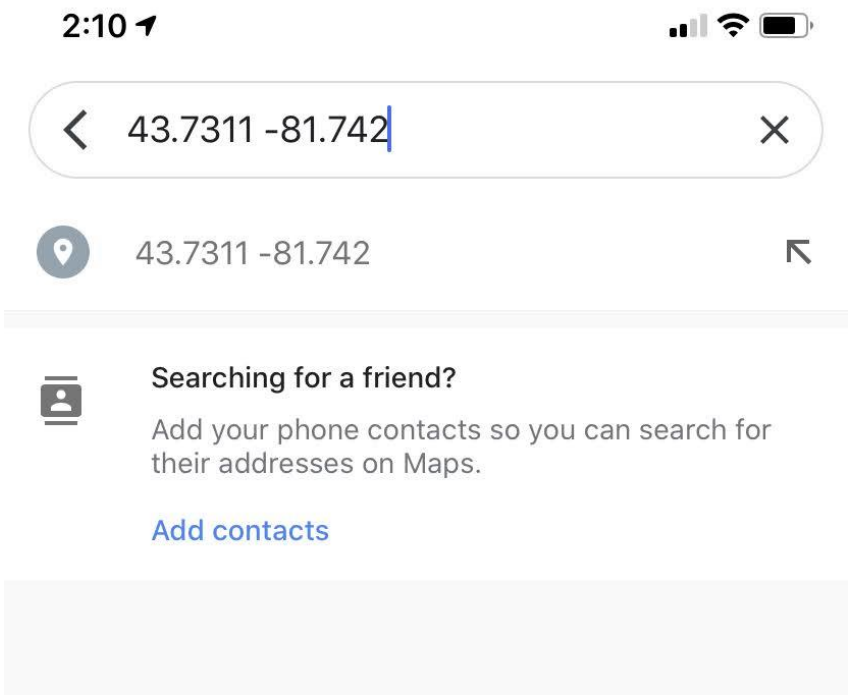
   NOTE: Google seems to limit the plots to 2000 entries so ensure your CSV file only has less than 2000 rows. If you have more- copy/paste them apart in to separate files for upload/plotting

Now you can then see all the MAC addresses plotted on a single map- HANDY!

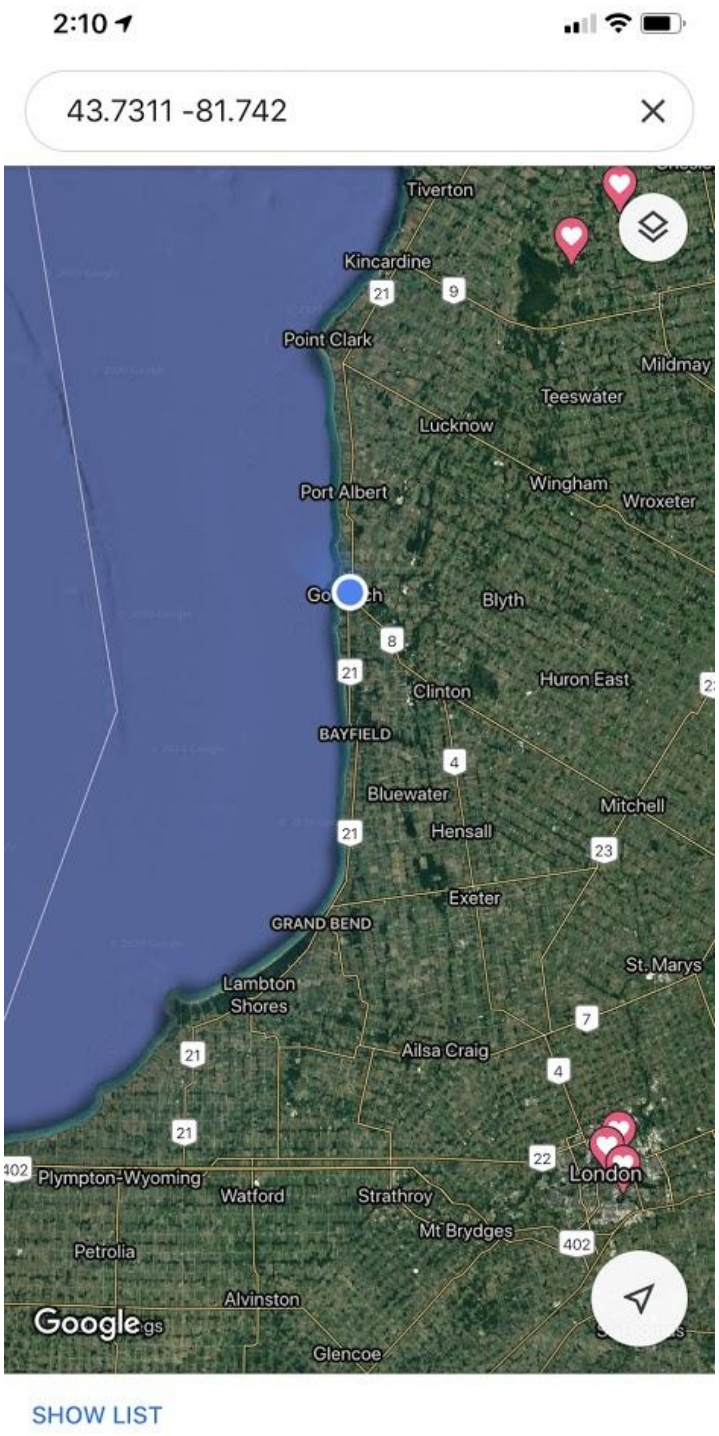


4.1.2 Manual single target plot

Enter the Lat/long directly in to your map program of choice such as directly on your cell phone. Hint- just take a picture of the ResQ display then enter those results in to your map app of choice.

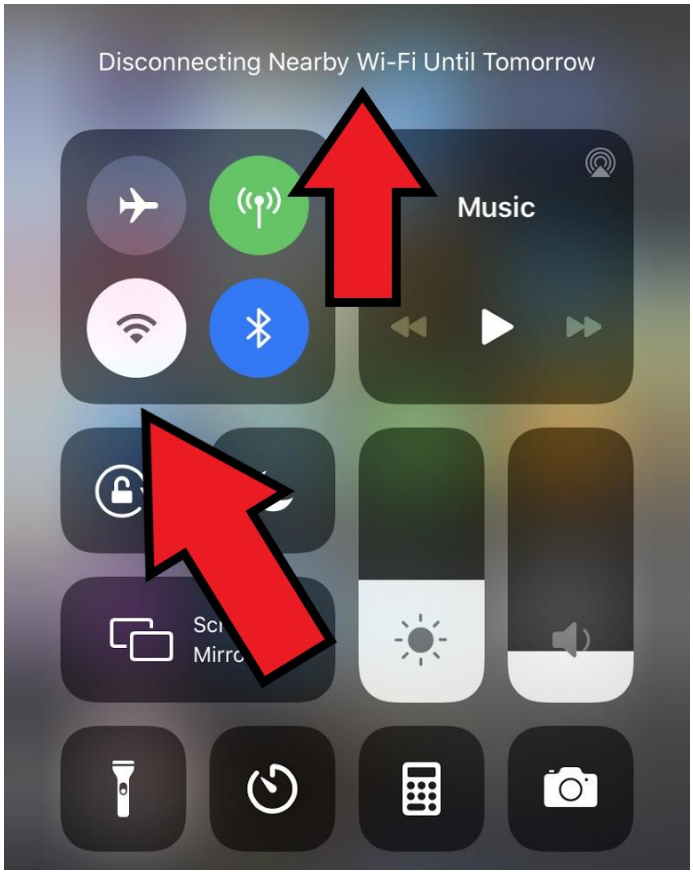


You will get this handy result right on your phone:



4.2 Special Instructions

If you are performing a search with other personnel you need to ensure they turn OFF WIFI on their phones. No need to do airplane mode but they need to turn off wifi to eliminate as many false targets as possible.



If you have (or can get) the MAC address for your search target you can filter easily for this or modify the Arduino code directly

IMPORTANT NOTE:

If you are doing a search/rescue and get a group of target MAC's in some remote area where there shouldn't be multiple people- THIS IS LIKELY ONE PHONE!

Apple phones spam fake MAC addresses so treat this as a single target if you get a bunch in the same GPS area.

Happy searching!

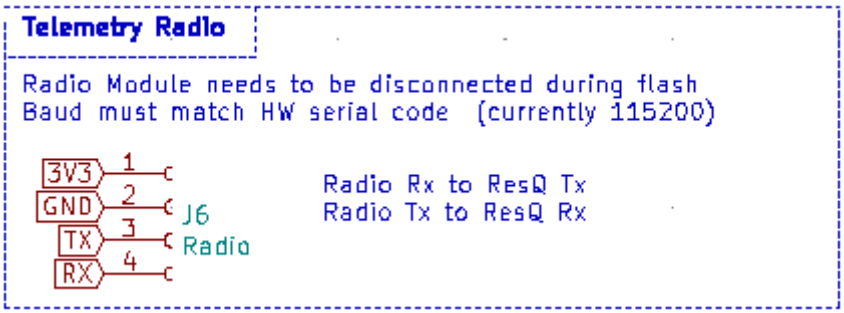
NOTE: A future code mod should add this functionality

If Clients detected ==1 (display RSSI and MAC ONLY on OLED screen)

This will make it possible to direction-find the target by varying the direction you point the YAGI antenna. Feel free to contribute this code if you are a dev ☺

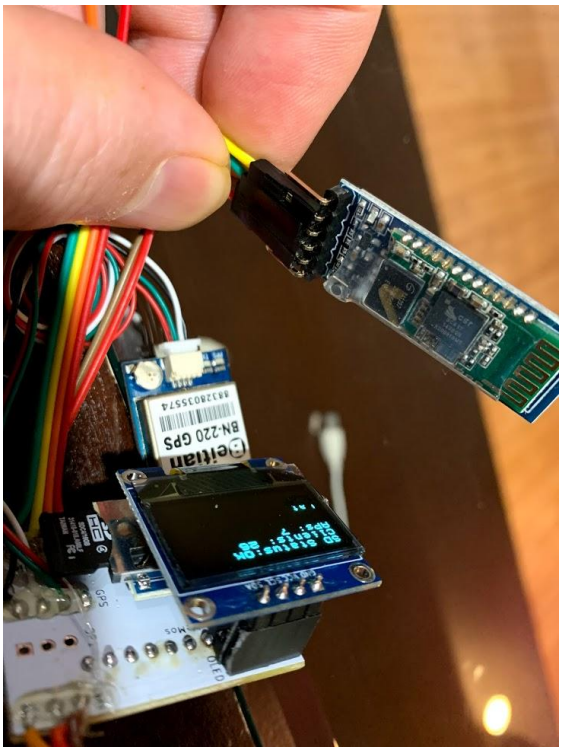
4.3 Bluetooth Module

V2.1 PCB and beyond supports the use of a radio module on the expansion port here:



You can use an HC05 Bluetooth module to view the ResQ status and current targets on your cell phone. This allows you to mount the ResQ unit to things like the exterior of an aircraft/helicopter while being able to monitor the status from the cockpit with only your cell phone. This eliminates the need of running wires and invasive installation on the aircraft.

For Bluetooth you need to use the HC05 Module or equivalent:



You need to program the HC05 using AT commands to change the baud rate and identifier (optional) so it can communicate at 115200 baud with the Resq. Instructions are in the code comments here:
https://github.com/MKme/ResQ/tree/master/Code

AT commands you need to send via serial adapter are:

"AT+UART=115200,1,0"
"AT+NAME=ResQ"

I am making pre-programmed HC-05 Bluetooth module available on my website here in case you do not wish to go through programming one yourself: http://store.mkme.org/product-category/custom/

Connect the Bluetooth module to the schematic connections shown.

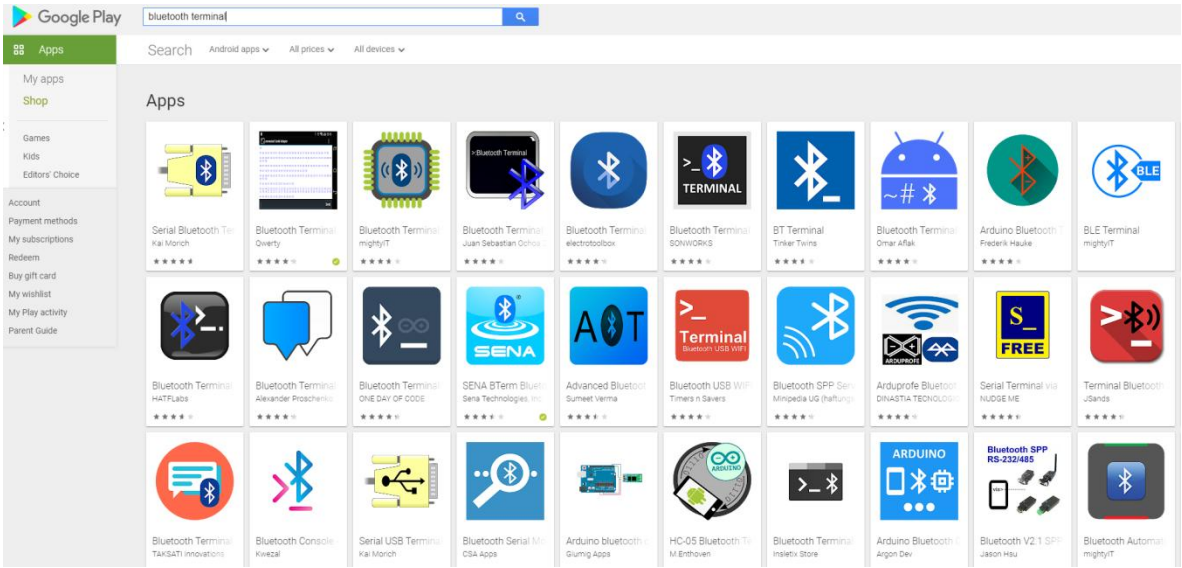
VCC must be 5 volts not 3.3V! (You must run your own wire/connection as the VCC on the radio port is 3.3V)
GND>GND
RX>TX
TX>RX

Pair the ResQ Bluetooth module to your phone:

PIN: 1234

I have no dedicated ResQ mobile application yet so you need to download any Bluetooth serial terminal application from the Play Store. I like Blueterm but you can use whatever one works with your phone and has good reviews at the time:

Here is how to find one:



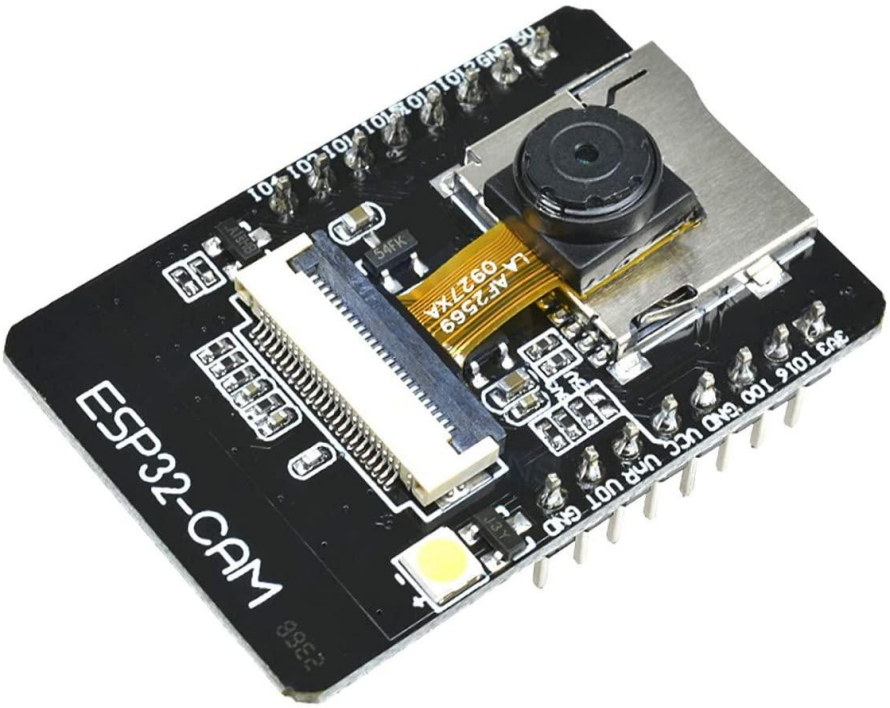
Open the ResQ data stream in your terminal and you will see the serial data in plain text! You can see the current targets and any changes as they scroll through. When we have a dedicated app made this will be a lot easier

If you are an app dev feel free to make oen which overlays the target on a map using the serial data which includes the targets, GPS coordinates and MAC ☺

Otherwise I will try to make an app somethime in the upcoming months. Check in on the Forum or Discord for updates.

4.4 Camera Module

Camera module consists of an ESP-Cam board interfaced to the ResQ board via 3V3, Ground and Signal wire.

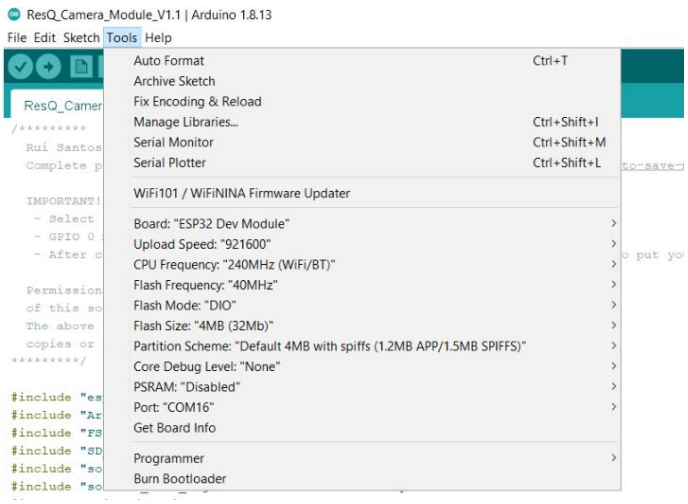


The signal wire is triggered by the ResQ any time a new MAC address target is detected

Camera module kits can be found here: http://store.mkme.org/product-category/custom/

Camera kits come preprogrammed. If you are supplying your own then you must program it with the code found here: https://github.com/MKme/ResQ/tree/master/Code

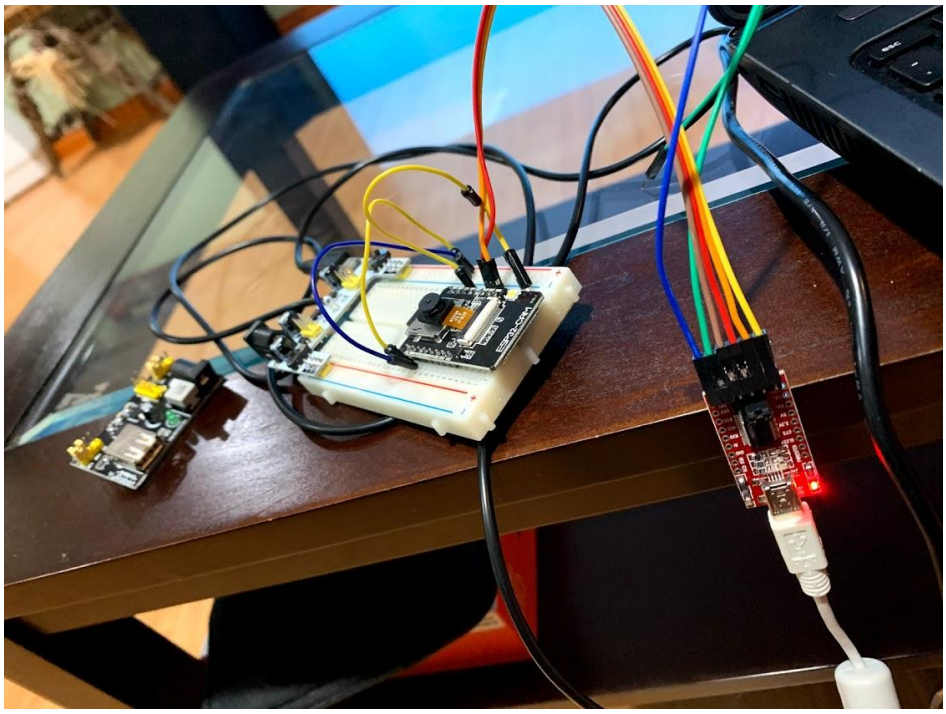
Use these settings in your IDE to program the board (These are VERY important and will save you hours of trying to figure out why your camera talks gibberish in serial after a seemingly successful upload) :



Instructions for how to do the code upload are found here:

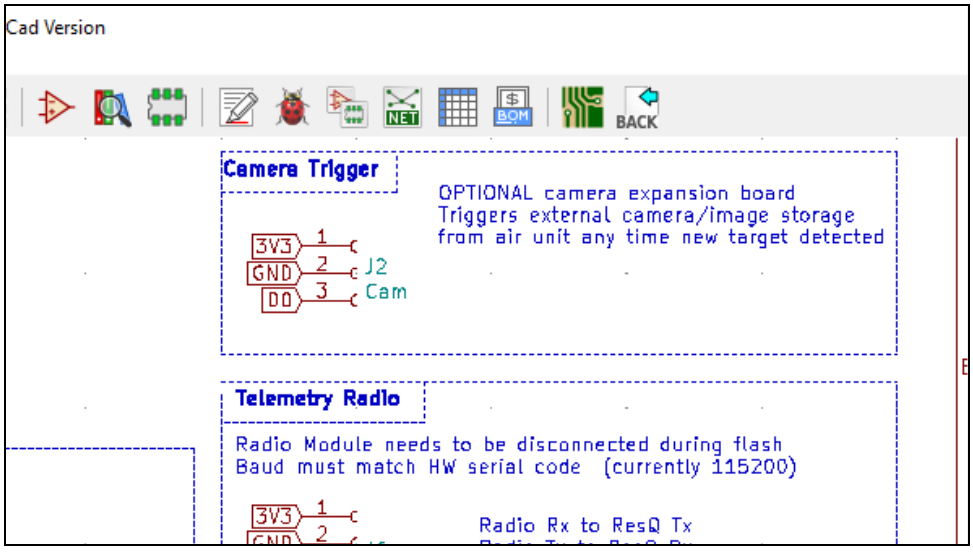
https://randomnerdtutorials.com/esp32-cam-take-photo-save-microsd-card/

You need to use a USB to serial programmer and connect it directly (or I will program them for you if ordered from my site)

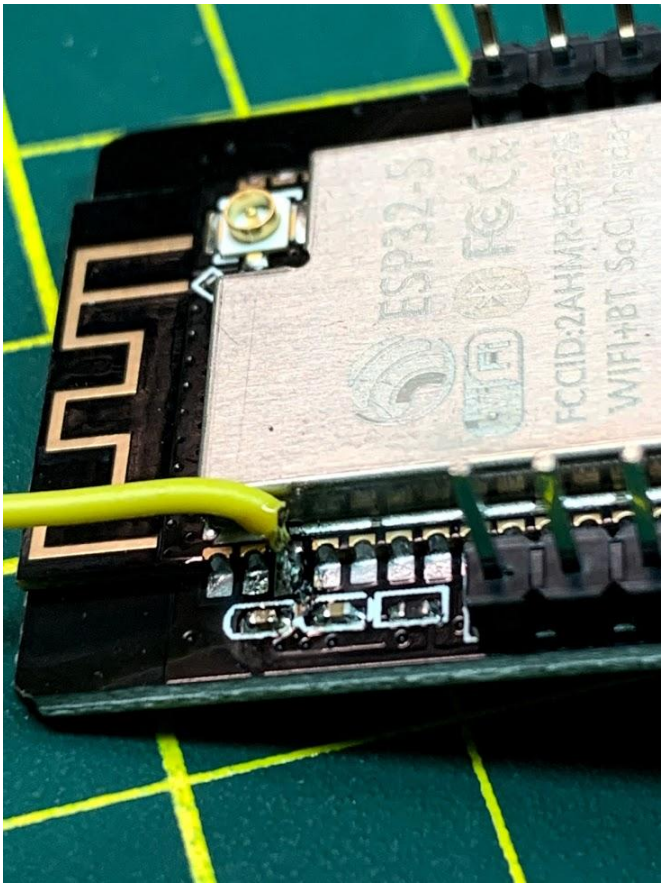


Attach the wires to the ESPCam pins directly. Almost all sources of these camera modules have the pins already soldered on so the case provided includes room for the pins. If you want to remove them and go with a thinner module you can find suitable enclosures on Thingiverse.

V1 of this design requires only the RST pin of the Cam module be connected to the ResQ Camera expansion port labeled D0 in the schematic:



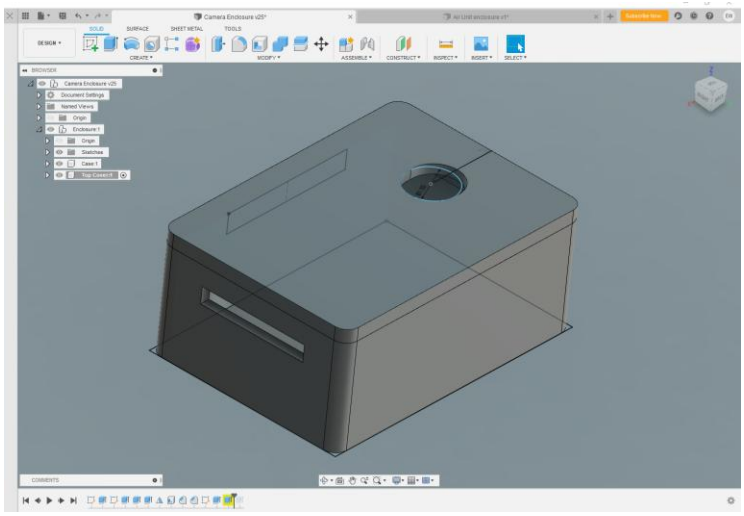
The pin on the camera module to reset is EN or RST which is this one:



You need to connect GND from the Cam module to GND on the ResQ as well.

You must feed the Camera Module a reliable 5V capable of ~500mA to be completely safe. This will give you plenty of headroom if you decide to use the flash LED as supplied in the code etc (flash is enabled by default in the code). In deep sleep the camera module will draw only microamps.

Download the enclosure files to 3D print yourself here: https://github.com/MKme/ResQ



I have provided 3D printable enclosure files as well as 2D SVG files you can laser cut, CNC mill or even cut out by hand. Everyone should be covered regardless of your access to the machinery.

4.5 ResQ Sentry

Sentry is a ResQ variant which includes the existing hardware plus the camera module inside a case which then becomes a stationary camera for monitoring human traffic in any area. Any time a new cell phone MAC address is detected- the camera module is triggered and captures a still image of the area.

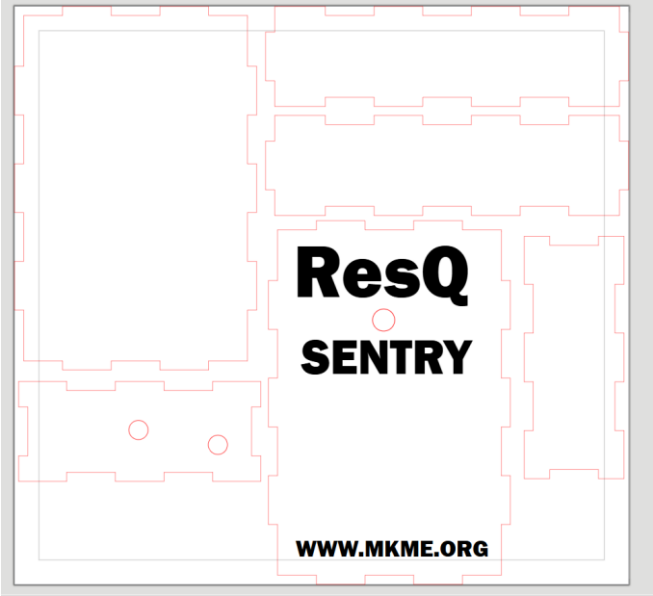
This can be used in game preserves to catch poachers, private property surveillance, restricted areas, and many more!

The GPS is still used to allow for accurate date/time stamps without the need for adding a real time clock. Time function will work without clear line of site to any satellites (it works indoors just fine).

2 different enclosures are provided:

1. 3D printable .stl version
2. Laser cut version (.svg files)

OPEN SOURCE SEARCH AND RESCUE
WWW.MKME.ORG
ResQ
CAM
ResQ
SENTRY



1. Install the ResQ hardware along with the camera in to the Sentry enclosure (camera through front hole)
2. Power the Sentry with an appropriate battery supply (5VDC to the ResQ)
3. Add an appropriate solar charging system if desired (I bought this panel for $21 CAD) https://amzn.to/3hOA5qX

Code update:

From Code version 2.5 ResQ supports Sentry mode by toggling one parameter!

From the code comments/notes:

```
V2.5
2020-09-24
Sentry Mode Integration
It works!!!!
Simply toggle the int in setup from SENTRYMODE=0 to 1 and
The OLED will show init right up untill clients are detected to be written to SD card (you can use this to test function by NOT disabling wifi on your cell phone)
This allows you to see if SD card and GPS are both OK then OLED will turn off to save power.  AWESOME!
Method used is powersave mode right in the OLED library- so cool
Done deal- Setry mode now works and is fully supported to capture poachers, theives, whatever you like. Enjoy!
```

You simply need to change this:

```
int SDOK = 1;//used to indicated if SD card is alive and still being written
int SENTRYMODE = 0; //Sentry Mode enable 1 disable 0 ------------------------------

// uint8_t channel = 1;
unsigned int channel = 1;
```

Change 0 to a 1 and upload the board. Your ResQ is now in Sentry mode and will turn off the OLED after it is initialized and writes the first target to the SD card. This will reduce power consumption significantly.

Alternatively you could disconnect the OLED via hardware disconnect but software sleep works nicely and is supported by the library used. Handy.

Notes:

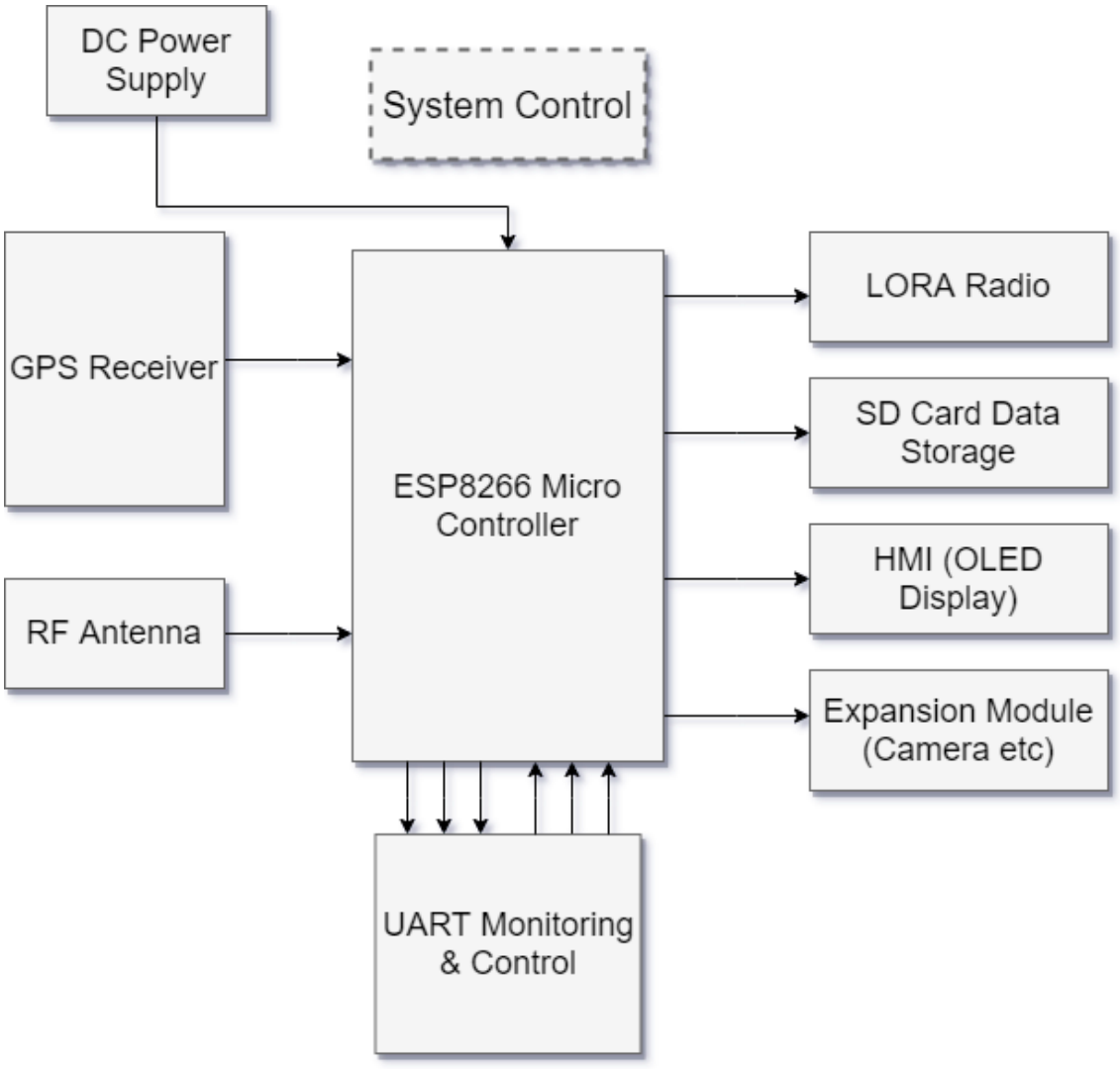
The ResQ cam module goes in to deep sleep whenever not taking a picture. The current consumption is VERY low in deep sleep. The ResQ unit itself will draw ~ 0.06A when passively sniffing for targets with an OLED attached. Disable/remove the OLED to reduce this current consumption much further.

10.0 APPENDIX

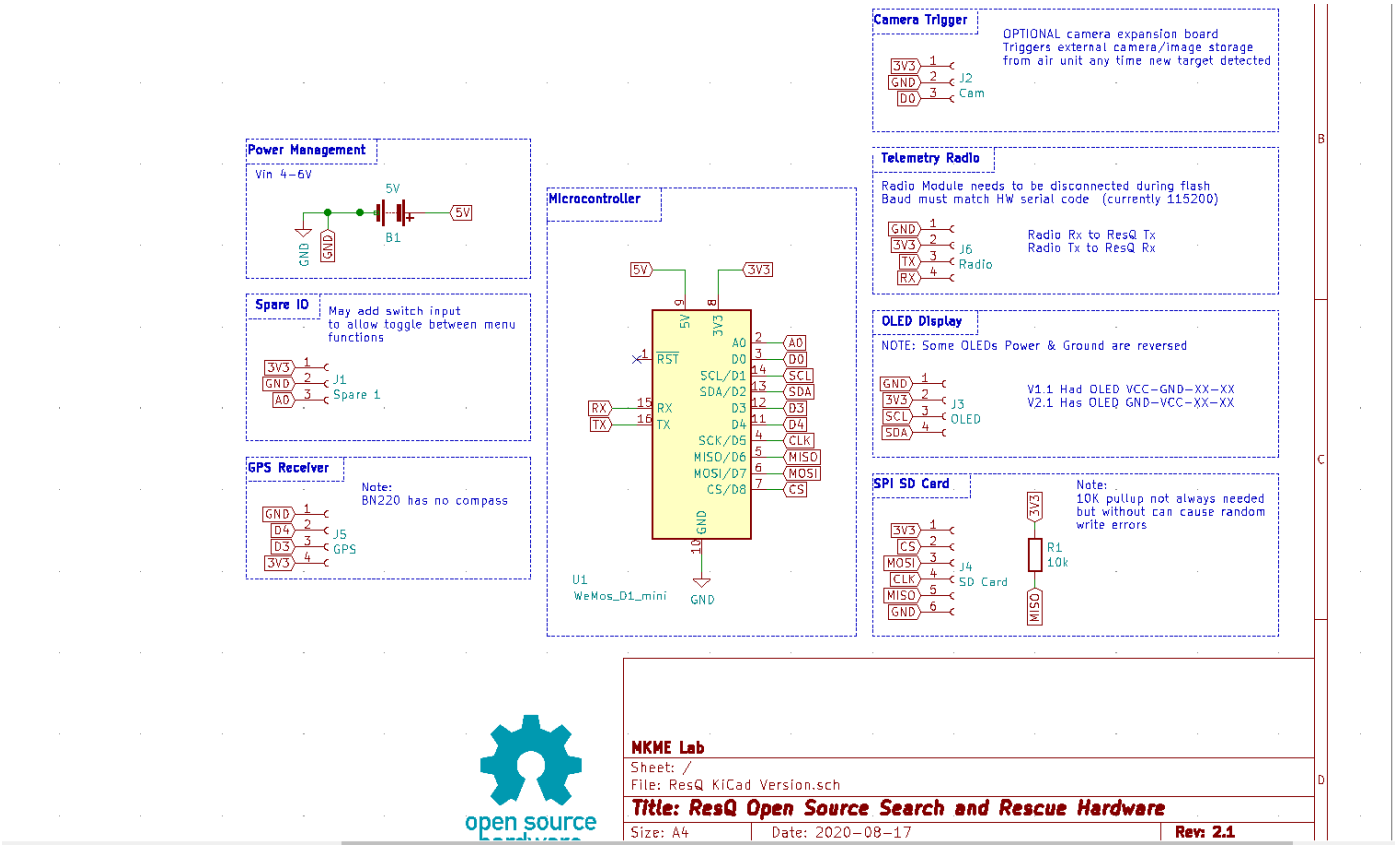
A. APPENDIX

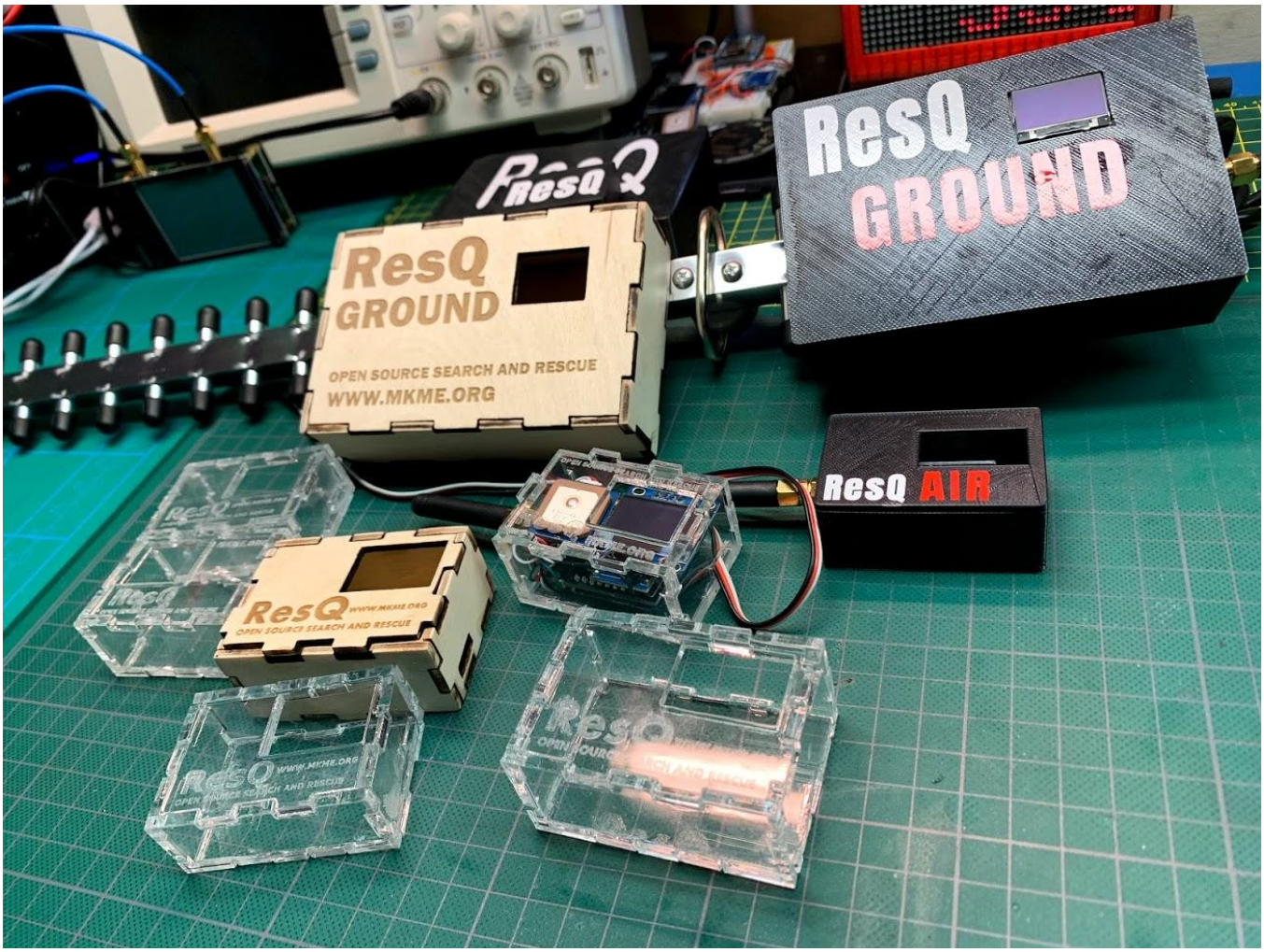
This section for all other handy/random info.

Flow:



ResQ
GROUND
ResQ AIR
ResQ ESP8266
DIY Kit
V1.1
RESQ ESP8266
PCB
V1.1
Eric @mkrnsorg
2020
PCBWay


Camera Trigger
OPTIONAL camera expansion board
Triggers external camera/image storage
from air unit any time new target detected
3V3
GND
D0
J2
Cam
Power Management
Vin 4-6V
5V
GND
B1
Telemetry Radio
Radio Module needs to be disconnected during flash
Baud must match HW serial code (currently 115200)
GND
3V3
TX
RX
J6
Radio
Radio Rx to ResQ Tx
Radio Tx to ResQ Rx
Microcontroller
5V
3V3
RST
RX
TX
A0
D0
SCL/D1
SDA/D2
D3
D4
SCK/D5
MISO/D6
MOSI/D7
CS/D8
GND
U1
WeMos_D1_mini
OLED Display
NOTE: Some OLEDs Power & Ground are reversed
GND
3V3
SCL
SDA
J3
OLED
V1.1 Had OLED VCC-GND-XX-XX
V2.1 Has OLED GND-VCC-XX-XX
Spare IO
May add switch input
to allow toggle between menu
functions
3V3
GND
A0
J1
Spare 1
SPI SD Card
Note:
10K pullup not always needed
but without can cause random
write errors
3V3
CS
MOSI
CLK
MISO
GND
J4
SD Card
R1
10k
GPS Receiver
Note:
BN220 has no compass
GND
D4
D3
3V3
J5
GPS
MKME Lab
Sheet: /
File: ResQ KiCad Version.sch
Title: ResQ Open Source Search and Rescue Hardware
Size: A4
Date: 2020-08-17
Rev: 2.1
open source


ResQ
GROUND
OPEN SOURCE SEARCH AND RESCUE
WWW.MKME.ORG
ResQ AIR