RESQ

[WWW.MKME.ORG](http://www.mkme.org)

OPEN SOURCE SEARCH AND RESCUE

SYSTEM DESIGN DOCUMENT V1.0

Overview

This open source project investigates using AP beacon packet sniffing to locate people such as lost hikers, campers and even poachers in game preserves. This works anywhere- as long as wifi is enabled on the targets phone. The hardware detects the beacon frames from their cellular phone, records the GPS coordinates for use later.

2 are made:

- ***Hand-held portable***
- ***Airborne carried by plane or drone***

Revision Sheet

Release No.	Date	Revision Description
Rev. 0	06/01/2020	Initial design doc

1 INTRODUCTION

1.1 Purpose and Scope

This document is a work in progress and will be updated OFTEN. Thanks for your understanding!

The ResQ tools will be able to deploy anywhere in the world cheaply and not dependant on any external support system. They can be made locally and deployed to search and rescue personnel, game wardens, law enforcement and many more.

1.2 Project Executive Summary

I [Eric] was inspired by the constant stream of search and rescue operations on the television show "North Woods Law". Each week they struggled to find lost people in the remote/rugged areas of Maine U.S.A. These areas often have no cellular service but in almost every case the target people had cellular phones on them and working. As I live in Canada- we have a surplus of these situations in our vast wilderness as well. I theorized I could use an external 2.4GHz antenna to receive the beacon frames from those phones ad capture that data.

This project tested that theory and proved that simple hardware could be used to detect the presence of a cellular phone in areas of limited visibility or access.

1.2.1 System Overview

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.

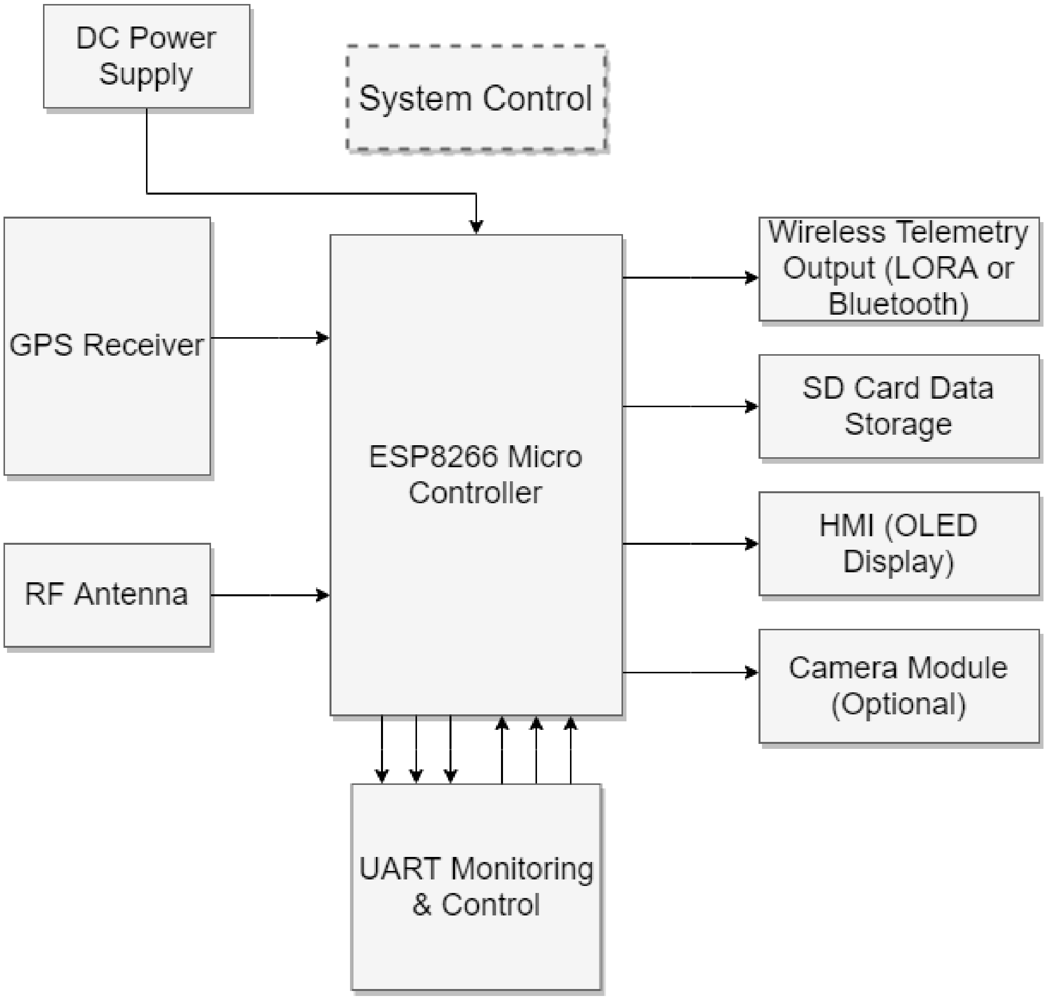
Using small drone aircraft (like the Nano Talon I use here) gives all the advantages such as flight condition tolerance, height, terrain, time of day etc that drones have over manned aircraft. They can also be sent on the full way-point mission without human intervention and tracked live from any ground 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:

I started this build in 2018 and have proven out the Nano Talon airrframe to be a very worthy commercial candidate with good payload and HUGE flight times while operating fully autonomously. The Nano Talon breaks down small for easy transport/storage.

So far I am using the Wedmos 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.

Video:

https://youtu.be/FxJkW-vr78s

1.2.2 Design Constraints

Space in/on any aircraft or drone is limited so care had to be taken to keep the size and weight of the air unit to a minimum. Current design can be drastically reduced further with a small PCB redesign (put all components on ONE board)

Handheld ground unit supply is limited to conserve weight but can easily be expanded if the situation requires. LiPo batteries can also be substituted for LiOn units to increase capacity by weight.

1.2.3 Future Contingencies

PCB redesign to place ESP8266, SD slot and OLED on single board with charge control circuit.

Amplification circuit with LNA can be added to increase target detection (not currently needed)

LORA/Wireless telemetry downlink from Air unit under final development

Camera trigger- Yes we can even trigger an external camera! Coming soon…

1.3 Document Organization

References/links will be provided throughout to external sources when applicable. Source files in GitHub repository will contain the most up to date information as it will accept community submissions as this project evolves.

1.4 Points of Contact

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

Forum Thread:
http://mkme.org/forum/viewtopic.php?f=28&p=1435#p1435

1.5 Project References

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

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

1.6 Glossary

Arduino- Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control the physical world.

The project is based on a family of microcontroller board designs manufactured primarily by SmartProjects in Italy, and also by several other vendors, using various 8-bit Atmel AVR microcontrollers or 32-bit Atmel ARM processors. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino platform provides an integrated development environment (IDE) based on the Processing project, which includes support for C,C++ and Java programming languages.

I2C
WIFI
GPS
Software Serial
YAGI
Dipole
SPI
SD

2 SYSTEM ARCHITECTURE

Arduino C++ code running on Expressif ESP8266 provides basic operating system uploaded via Arduino IDE

2.1 System Hardware Architecture

1. Power Supply- USB powered or ANY 3-6VDC source
2. Wemos D1 Mini Pro ESP8266
3. 0.96" OLED
4. SD Card
5. GPS Receiver
6. Antenna (dipole air unit, yagi ground unit)
7. VCC/Power (air unit)
8. Battery Power (ground unit)
9. Enclosures

BOM maintained in GitHub Repository here: https://github.com/MKme/ResQ/tree/master/BOM

2.2 System Software Architecture

- Arduino Sketch (C++) operating on ESP8266 hardware
- Code Releases: https://github.com/MKme/ResQ/releases
- Source Code: https://github.com/MKme/ResQ/tree/master/Code
- Full code walkthrough: TBD

2.3 Internal Communications Architecture

- I2C bus from ESP8266 to OLED display for HMI
- Serial data UART RX/TX at **115200 baud** rate for debugging/development
- Software Serial data at **9600 baud** UART RX/TX for GPS receiver communication
- SPI Interface to micro-SD card (storage)

3 FILE AND DATABASE DESIGN

All files and updates via GitHub: https://github.com/MKme/ResQ

3.1 Database Management System Files

- Datalog.txt on Secure Digital storage conatins all MAC addresses detected along with their GPS position and time detected in a comma separated values file (.csv)

3.2 Non-Database Management System Files

- Main code file and supporting header files. All can be found in the releases area on GitHub.

4 HUMAN-MACHINE INTERFACE

1. 0.96" OLED Display communicating via I2C datalines

4.1 Inputs

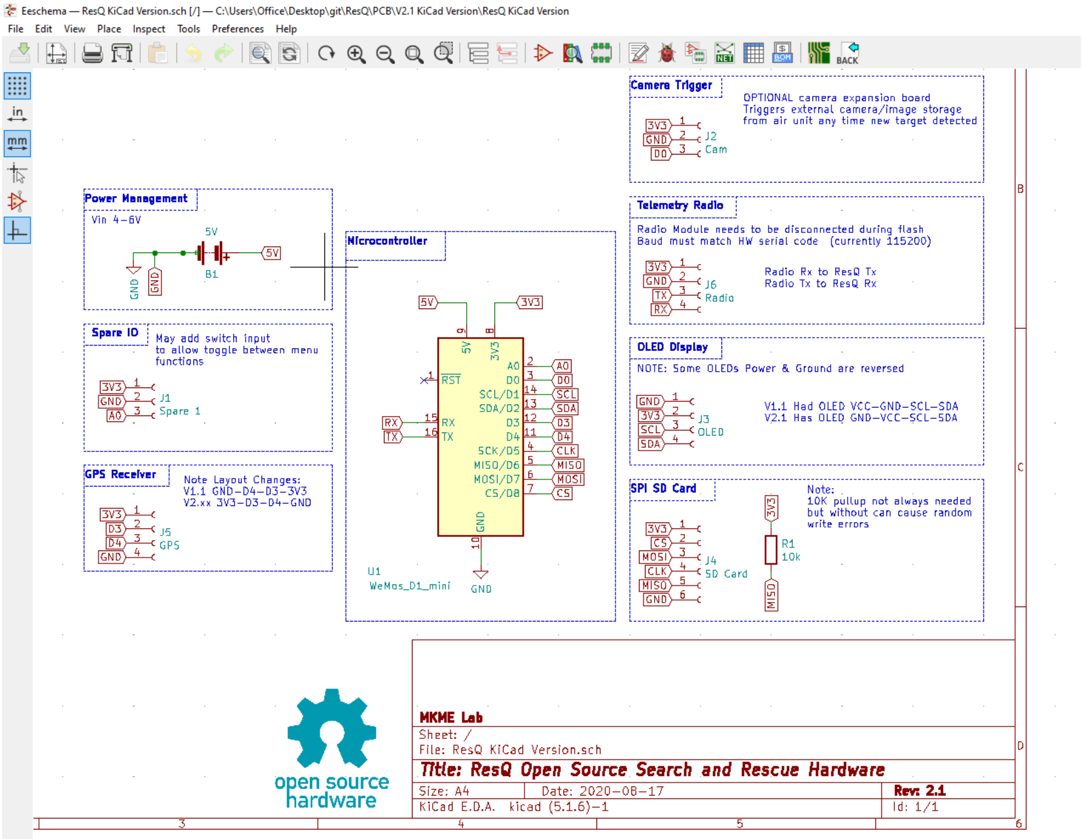
- Wireless 2.4Ghz Packet traffic captured/parsed by ESP8266 radio
- Datasheet:

4.2 Outputs

- Targets/Data displayed on HMI screen
- SPI Interface to Secured Digital Storage (SD card)

5 DETAILED DESIGN

5.1 Hardware Detailed Design



OPTIONAL camera expansion board
Triggers external camera/image storage
from air unit any time new target detected

Radio Module needs to be disconnected during flash
Baud must match HW serial code (currently 115200)

Radio Rx to ResQ Tx
Radio Tx to ResQ Rx

NOTE: Some OLEDs Power & Ground are reversed

V1.1 Had OLED VCC-GND-SCL-SDA
V2.1 Has OLED GND-VCC-SCL-SDA

Note:
10K pullup not always needed
but without can cause random
write errors

May add switch input
to allow toggle between menu
functions

Note Layout Changes:
V1.1 GND-D4-D3-3V3
V2.xx 3V3-D3-D4-GND

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
KiCad E.D.A. kicad (5.1.6)-1 Id: 1/1

Schematic Diagram

5.2 Software Detailed Design

- Setup
- Loop

5.3 Internal Communications Detailed Design

- 115200 baud UART Hardware Serial
- 9600 baud UART Software Serial (GPS communication)
- I2C bus for HMI (display)
- SPI bus for SD Card Storage interface

6 EXTERNAL INTERFACES

- Camera trigger (Digital I/O pin)
- Telemetry radio via 115200 baud UART shared with programming USB

6.1 Interface Architecture

- 2 terminal UART for telemetry on PCB
- 3 terminal camera trigger interface (VCC,GND,Trig)

6.2 Interface Detailed Design

- 128X128 pixel OLED

7 SYSTEM INTEGRITY CONTROLS

- Internal hardware/software watchdog native to ESP8266 is active and will initiate reboot of the processor in the event of a code execution stoppage.

- No checksums or manifest otherwise in place

- Automatic recovery is initiated from any SD file issue (removal) whereby a new file is automatically created if not present on every power cycle

- If datalog.txt is present on the SD card at power up the file is not removed- new data is added in sequence.