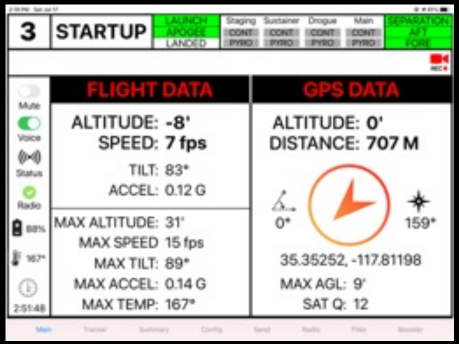
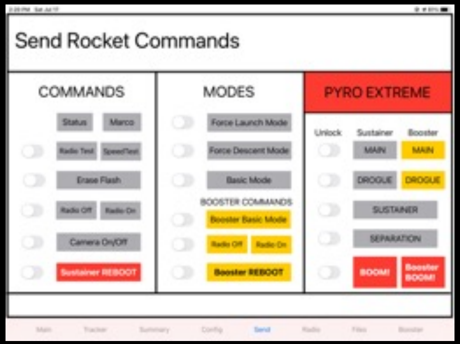
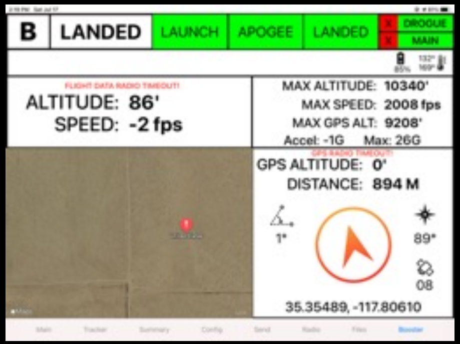
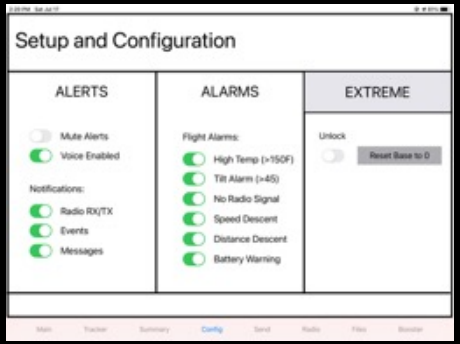
ROCKET TALK









Flight Telemetry, Tracking, and Deployment System

Version 5.1 – November 2022

Disclaimer:

We started this radio telemetry project in 2016 to better track our rockets in the skies over the Mojave desert. It has gone through a dozen hardware iterations and over fifty test flights. We always intended to share it and open source it, but projects like these are never complete. There is lots of room for improvement.

We've decided to take a snapshot and share our progress, so student groups and other amateurs can learn, borrow, or reuse some of our ideas and our sloppy code.

We owe a lot to the rocketry community, the TRF board, and student groups out at FAR that have shared their ideas, code, and practices to make our project and launches better and safer.

Please pay it forward.

Mike & Preston

Vehicle Board

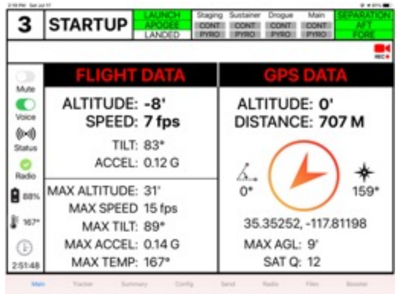
- Teensy 4.1 MCU (ARM M7)
- GPS
- Barometer
- Two Accelerometers
- Gyroscope
- Pyro drivers x 4
- Camera remote power control
- 64Mb of flash storage + SD transfer
- Temperature monitoring
- Voltage monitoring
- Bi-directional radio comms / 1W LoRa 433 Mhz

Optional Radio Repeater

Mesh network repeater
5W LoRa 433 Mhz

Base (handheld) iPad Mini

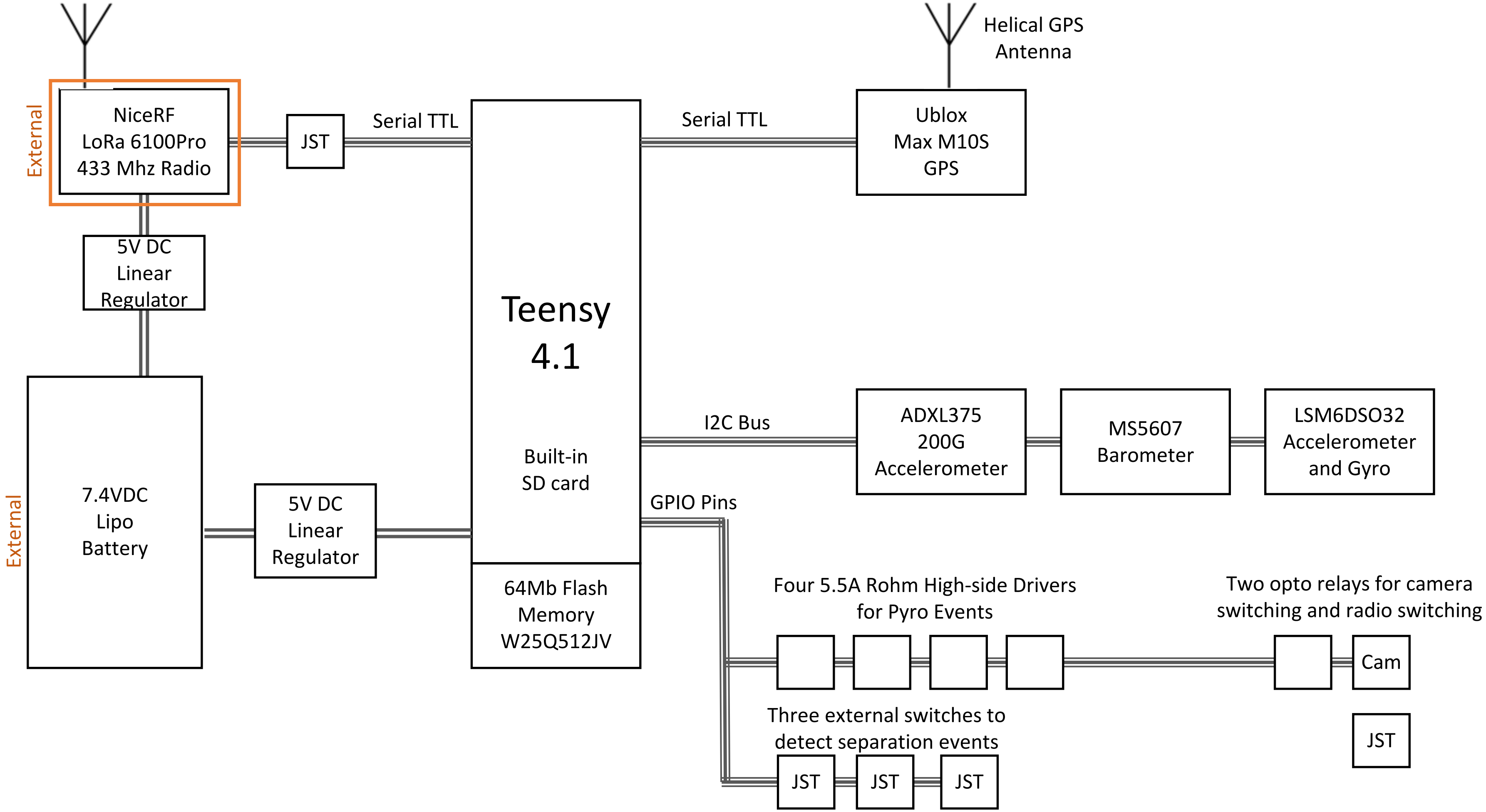
- Simple touch-based UI for data and tracking
- Voice event reporting
- Local base GPS
- Magnetometer / bearing
- Satellite mapping
- File storage
- Wifi/Internet
- Bi-directional radio comms
  1W LoRa 433 Mhz



RocketTalk

VEHICLE BOARD

Vehicle Board – Conceptual Architecture



Current Technology Choices – Vehicle Board

- **Teensy 4.1:** The T4.1 with its Arm Cortex-M7 is just insanely fast for an MCU and it is loaded with a lot of versatile GPIO. We started our project on other Arduino based MCUs. Our last one was the ESP32. In benchmarks, an ESP32 can increment an integer about 700K times a second. That seems fast until you see the T4.1 (or 4.0) increment an integer 242 million times a second (wowzers!). This provides for almost unlimited CPU cycles for anything you can dream up during flight. The real limitation becomes the I2C bus, communication speeds, serial speeds, etc. The T4.1 also has a built-in SD card for easy transfer of data. And it easily accommodates a 64mb flash chip (W25Q512JV). We log directly to the 64mb flash chip for more reliable 50-100hz logging and then copy to SD after flights for easy transfer to a computer.

- **GPS:** We are using the Ublox max M10S with serial communication. The performance of the M10S has been incredible. We often have 24+ satellites fixed while on the pad. We run the GPS serial ttl instead of I2C, so we can save our I2C bus for sensor sampling. These GPS units are still under ITAR regulations, so they have Cocom limits above 150K feet and above specific velocities. The v5.1 board has an SMA connector for the GPS, so we use a Maxtena Helicore M1575HCT-22P-SMA antenna instead of a narrowly focused patch antenna. The combination of M10S with an omnidirectional antenna has produced far more reliable results.

- **Barometer**: We are using the TE MS5607 barometer on I2C sampling at 40Hz. We were using the BME280, but found it very susceptible to RF interference. The MS5607 has been reliable (no RFI issues), tracks up to 110K feet in the baro chamber, but it has a noisier signal, especially on descent. We also have to do some code gymnastics to sample at 40 hz and allow the sampling thread (and I2C bus) to stay free.

- **Accelerometers:** We have an Analog Devices ADXL375 200G accelerometer and a ST LSM6DSO32 32G accelerometer. The 200G is good to catch all the really big stuff and we often exceed the 32G's of the LSM6DS, but the LSB resolution is not as good in a 200G setting, so the ST provides better resolution and accuracy below 32G's. We are sampling the accelerometers and the gyro at 400 hz on the I2C bus.

- **Gyro:** The ST LSM6DSO32 also has a built-in gyroscope. We are using that, along with quaternions to calculate the rocket attitude (tilt).

- **Radios:** We have tried a number of different radios, but we are very happy with our 1 watt LoRa radios on 433 Mhz (FCC amateur radio license required). You can get similar radios in 900 Mhz that don't require a license. But, you are in this deep, so take the time to get a license. We use the LoRa 6100Pro serial TTL radios from NiceRF in China. The Lora technology is great, as we can create a mesh network and add a third radio on a tower to act as a repeater. These have proven to be reliable up to 15 miles "peak to peak" on the ground (mountain test) and I'm certain they will go to 100K+ feet in the air with the right antennas.

- **Downlink Antenna:** The choice of downlink antennas on the vehicle is almost more important than the radio. We've tried dozens of antennas. The best antenna for us is a simple homemade dipole antenna running vertical in the airframe. We've tried whip, inverted V, fractal, and more, but the dipole performs the best. We used to make them by hand, but now we print our diploe antennas onto a long thin custom PCBs and trim to tune them (about $2/each).

Current Technology Choices – Vehicle Board

- **5V Switched Regulator:** We use two L4941BDT-TR 5v 1A regulators to power the Teensy and the Radio. We run a separate regulator for the radio to minimize RFI and RF blowback, as well, provide the radio with a full one amp of power. All of the pyro events are triggered directly off of the Lipo battery and do not go through the regulators.

- **Opto-Relays:** We are using the Toshiba TLP3553A SSR 4A opto relays to provide optoisolated switching for the radio power and to turn on-off the camera.

- **Terminals**: We use JST board connectors for the low voltage connections and screw terminals for the hazard connections (power, switch, pyros).

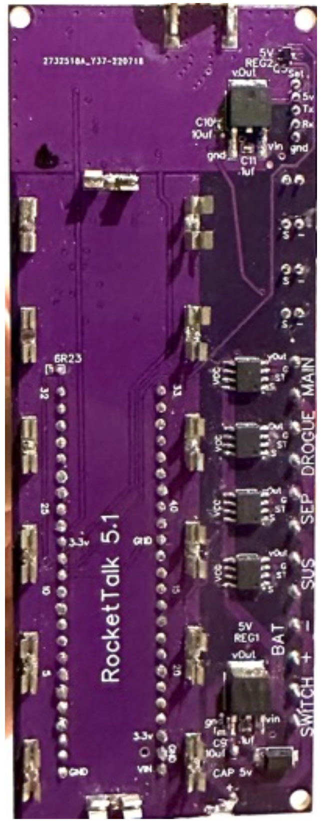
- **RFI Shielding:** A powerful one watt downlink transmitter comes with benefits and issues. On the positive, it can drive a data signal loud and far, but on the negative side it also showers all the sensors, the I2C lines, and the MCU with lots of disruptive radio frequency interference. In earlier versions we had sensor issues, code execution glitches, and full core dumps due to RFI. So, the current v5.1 board incorporates extensive shielding measures to protect the sensors and the MCU:

  - Full RF shielding cover top and bottom: Used Wurth Electronics DIY Shielding (360002) and Leader Tech TC-01 clips.
  - Ferrite beads inline with the I2C signals: Part BLM21SP102SH1D
  - Ferrite beads on the sensor VCC lines: TDK MPZ1608S101ATDH5
  - Small resistors on GPIO pins: 33 ohm resistors on GPIO pins
  - Added brown-out capacitor: 4F Vishay MAL219691104E3
  - Grounded unused GPIO pins
  - Significant copper ground infill under and across MCU and sensors
  - Note: The GPS chip is already shielded, so did not require special consideration

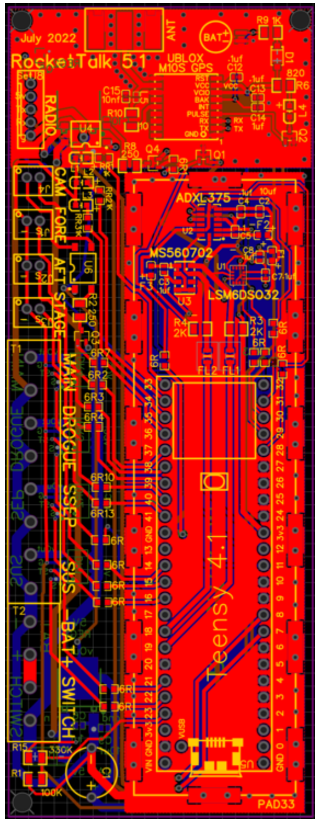
Vehicle Board – Custom PCB

Front

w/RFI Shielding



Back



PCB Design

The schematic and custom PCB are designed in EasyEDA and printed and ordered through JLCPCB (via EasyEDA). It really is easy.

Original versions were all through-hole design, but later versions have used SMD components to pack more on the board.

The PCB is a four layer board with one layer 100% filled for RFI protection.

Vehicle Board – Code Design

The Teensy code is built from a single C sketch in the Arduino (Teensyduino) IDE. It utilizes some external libraries for sensors, but is otherwise self-contained in a single code file.

The code is partitioned into three Teensy threads, mimicking three separate infinite looping cores. The three threads share global variables, but are otherwise completely autonomous. Given the CPU speed, there is very little noticeable blocking between threads. Individual timers are used to trigger checks and activity within each loop.

Main Thread Loop

- Phase Logic (for each stage of flight)
- GPS check and transmit
- Transmit flight telemetry data
- Check voltage
- Check for separation events
- Check continuity
- Fire pyros
- Radio receive processing

Sample Thread Loop

- Sample ADXL375 at 400 Hz
- Sample LSM6DSO32 Accel at 400 Hz
- Sample LSM6DSO32 Gyro at 400 Hz
- Compute Quaternion Tilt at 400 Hz
- Sample Barometer at 40 Hz

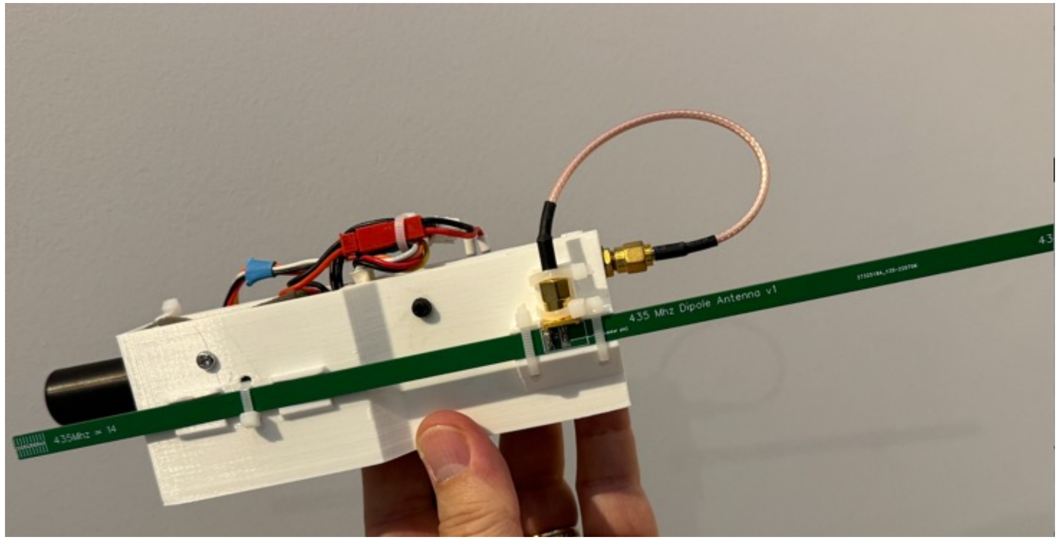
Logging Thread Loop

- Continuous logging to flash file system at various rates
- Logging flight and event data at 20 Hz
- Logging GPS data
- Logging radio TX/RX data
- Logging flight summary data

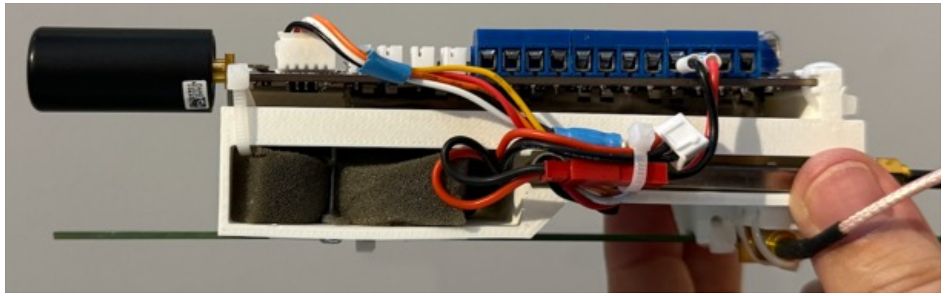
Vehicle Board – Code Design

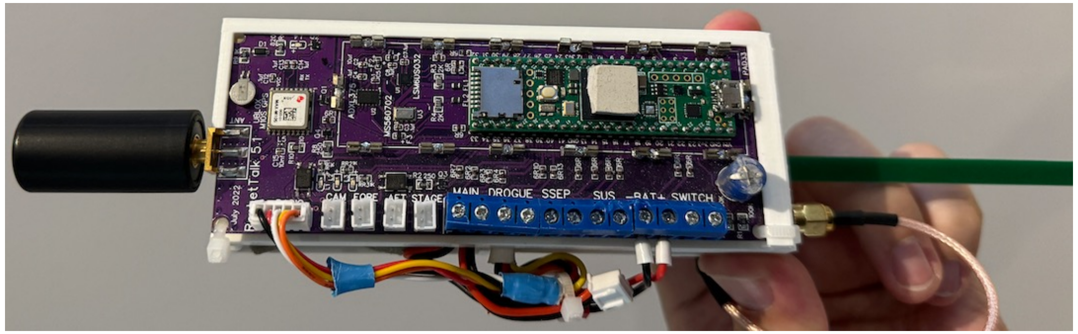
Flight Phase	
STARTUP	Power up sequence. Wait for GPS lock. Warm-up and characterize the Barometer. Bias adjust the Accelerometers.
WAITING	Waiting for launch. Calibrate the Barometer. Bias adjust the Accelerometers.
LAUNCH	Increase sample rate. Watch for and trigger staging pyro events. Watch for Apogee and fire drogue pyro.
DESCENT	Reduce logging after 20 seconds. Watch for main altitude and fire main pyro. Watch for landing using barometer.
LANDED	Log and transmit flight summary. Idle logging and transmitting.
BASIC	Exception mode (activated by base) with no event triggering, just sampling, logging, and transmitting flight/gps data.

Vehicle Board



A simple dipole antenna is the best antenna you can have for a rocket. Google “dipole antenna calculator” for precise measurement of the radials and use a cheap SWR meter to tune to size. This one was printed on a long and skinny PCB through EasyEDA and JCLPCB.

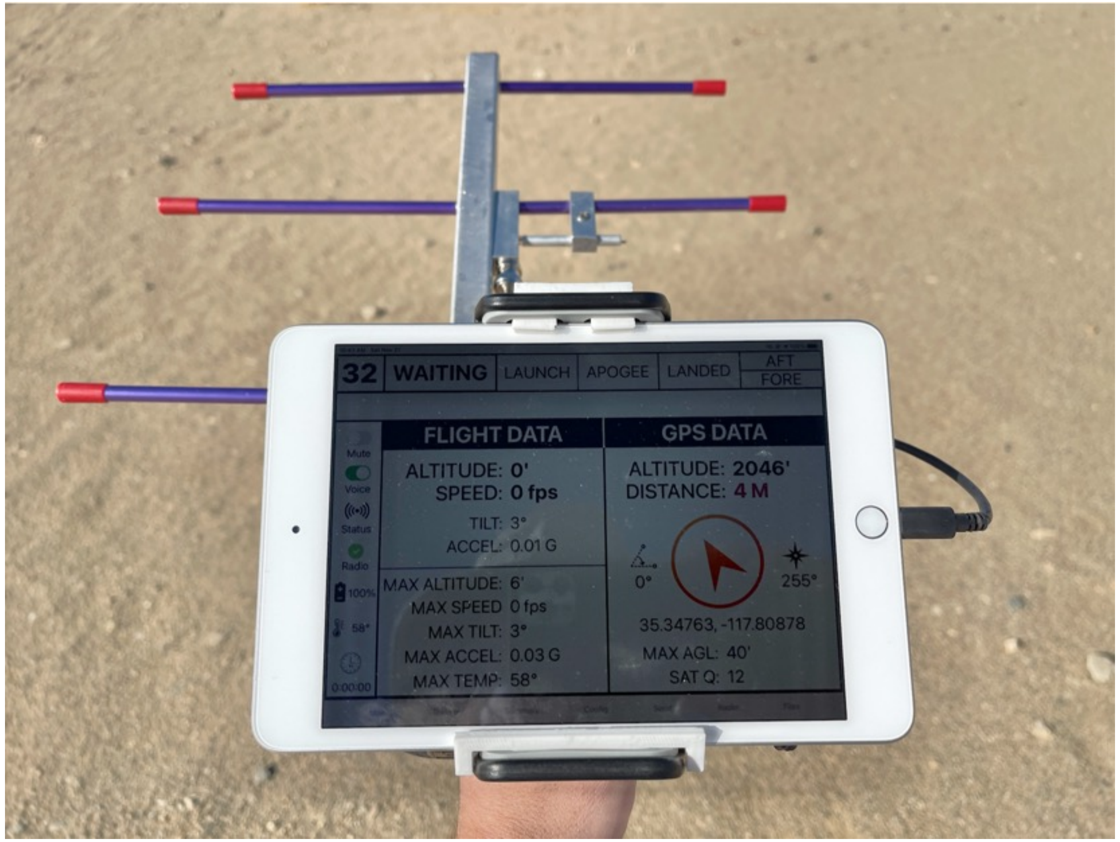




3D printed sled. Shown is our 3” rocket version with board, GPS, battery, and radio.

RocketTalk

BASE STATION iPAD


32
WAITING
LAUNCH
APOGEE
LANDED
AFT
FORE
FLIGHT DATA
GPS DATA
Mute
Voice
Status
Radio
100%
58°
0:00:00
ALTITUDE: 0'
SPEED: 0 fps
TILT: 3°
ACCEL: 0.01 G
MAX ALTITUDE: 6'
MAX SPEED: 0 fps
MAX TILT: 3°
MAX ACCEL: 0.03 G
MAX TEMP: 58°
ALTITUDE: 2046'
DISTANCE: 4 M
0°
255°
35.34763, -117.80878
MAX AGL: 40'
SAT Q: 12


6.9 – 8.3

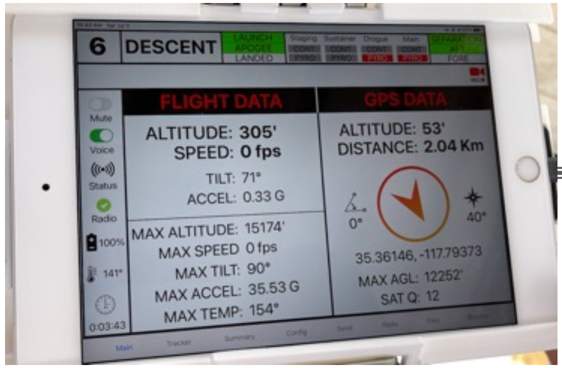
Current Technology Choices – Base Station

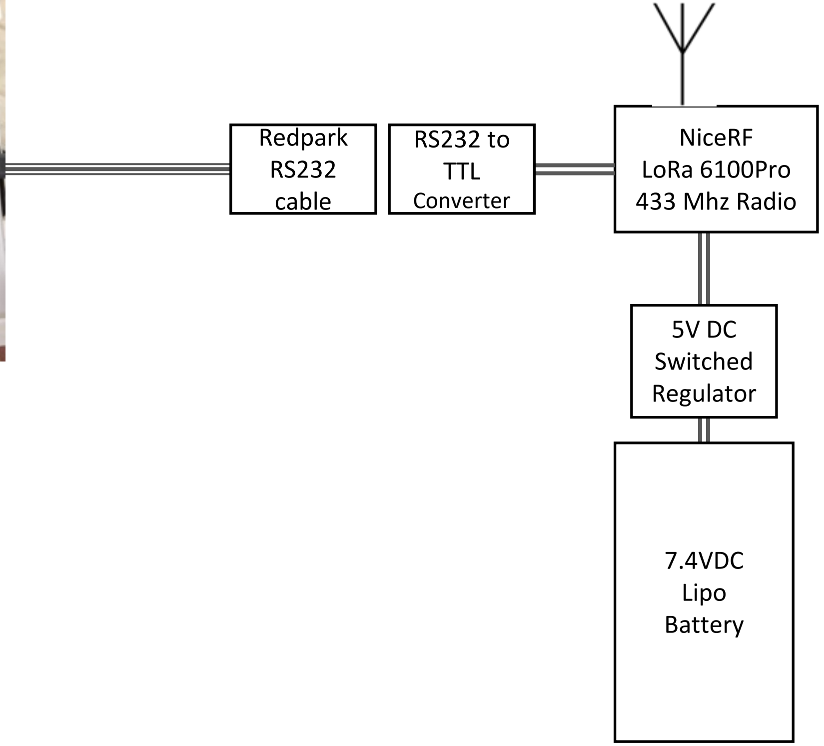
- **iPad Mini:** In earlier versions, we built the handheld base station from an Arduino and an LCD. This worked, but integrating the GPS and a magnetometer for tracking proved messy at best (field calibrating the magnetometer for an accurate bearing was not fun). So, we decided to switch over to an iPad mini, using Apple's Swift development code to build the application and UI. After a few days of "hello world" building in Swift was a dream, compared to building hardware. Pulling base coordinates to compare to vehicle coordinates only required one line of code. Magnetic bearing was only one line of code. Voice event announcements only one line of code. Maps drop right in. Everything was just so much easier. That said, this does require a cellular iPad mini (no plan required), as those are the only iPads with actual GPS hardware. Also, we needed to use a proprietary lightening to RS232 serial cable (Redpark serial cable) to interface with the radio. And the radio needs its own little companion box with a battery.
- **Redpark Serial Cable:** This cable and the iOS Swift libraries is required to interface the iPad to the LoRa serial radio. It is RS232 serial, so a small RS232 to TTL converter is also used before connecting to the radio. There are not many ways to interface an Apple iOS device to serial communications. This one works great. A note: You cannot distribute apps in the Apple App Store that use the RedPark cable. It can only be used with developer apps or enterprise app distribution.
- **Radios:** We use the exact same radio as the vehicle radio. The LoRa 6100Pro serial TTL radios from NiceRF in China.
- **Antenna**: We are using an Arrow II handheld Yagi antenna (440-3) for good directional reception.

- **Additional mounting components:** To make a great one-handed tracker, we mount the iPad onto a UZOPI Mavic Mini Air Pro Platinum Spark Accessories Tablet Holder and we bolted the antenna and a camera pistol grip to the holder. The radio, battery, and cable are in a small project box attached to the holder plate.

Base Station – Conceptual Architecture

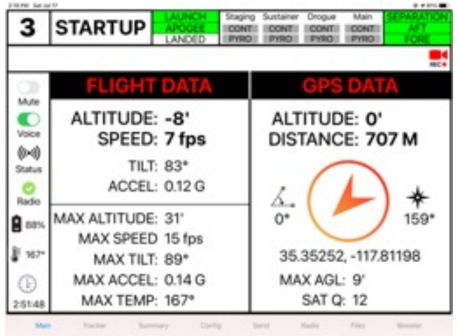
iPad Mini (cellular / no plan)





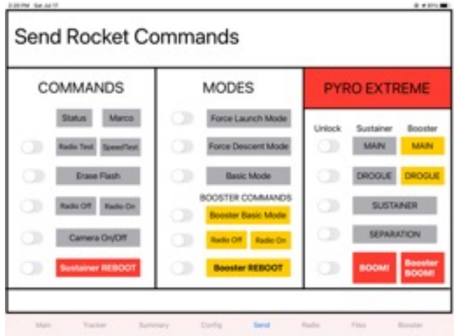
Base Station – Application Features

The native iOS application written in Swift uses a Tab View Controller with six tabs or screens that you can easily toggle between.



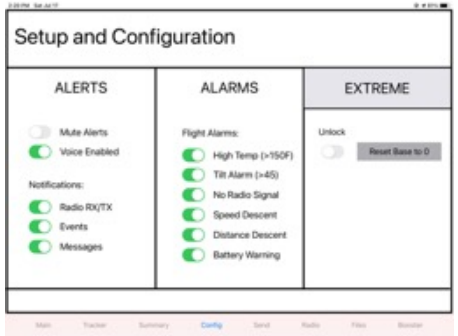
Main Screen (Sustainer)

Tracking Screen (Sustainer)



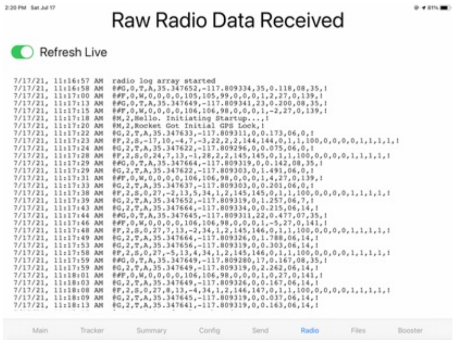
Transmit Screen

Booster Screen



Configuration

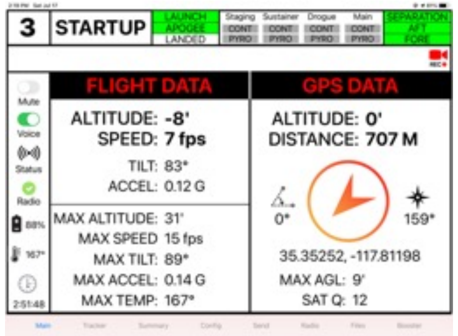
Log Files



Raw Radio Data

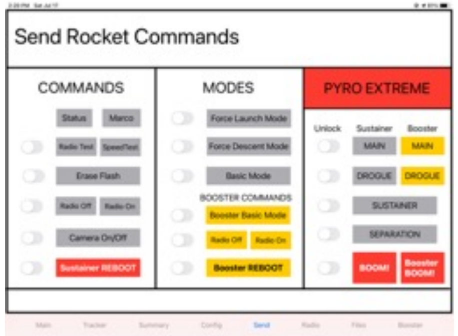
Base Station – Application Features

The native iOS application written in Swift uses a Tab View Controller with six tabs or screens that you can easily toggle between.



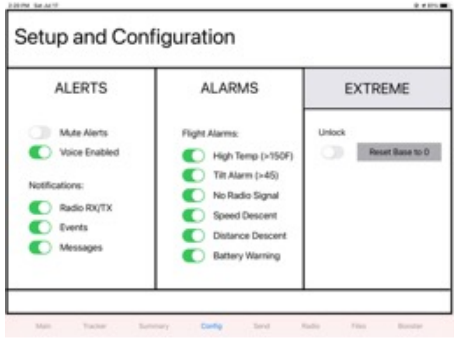
Main Screen (Sustainer)

Tracking Screen (Sustainer)



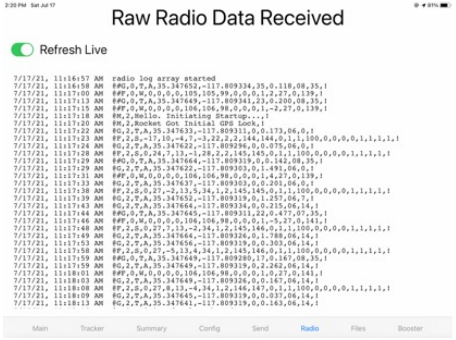
Transmit Screen

Booster Screen



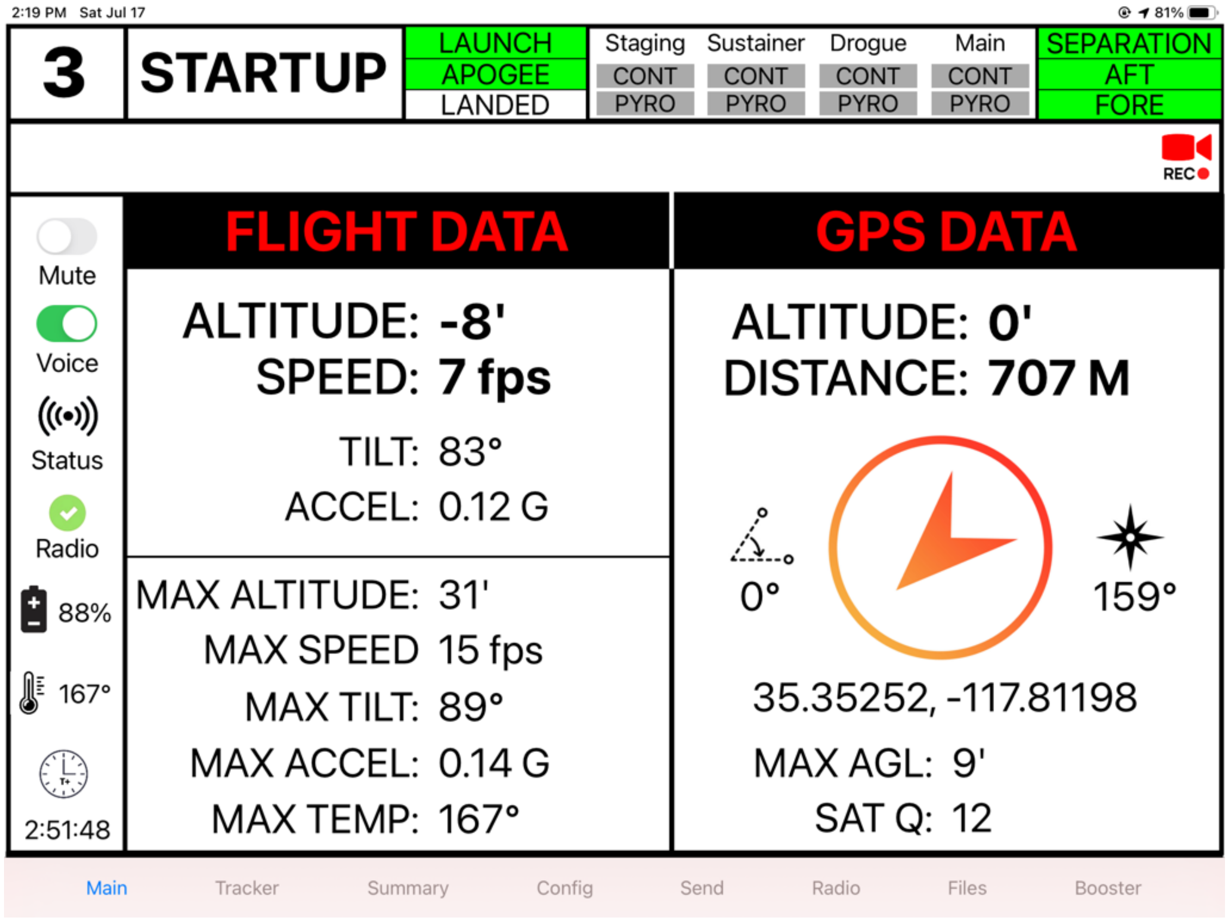
Configuration

Log Files



Raw Radio Data

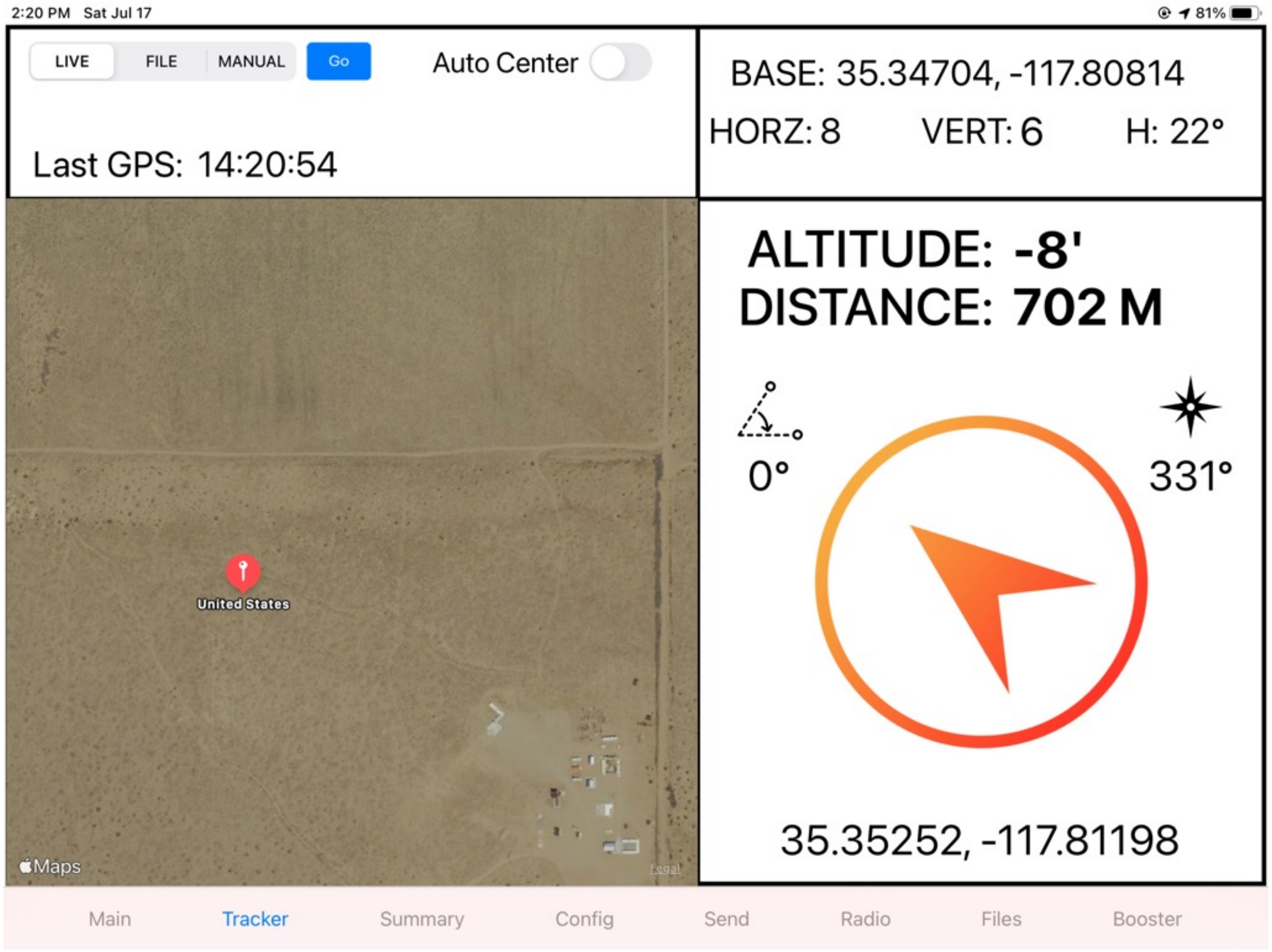
Base Station – Application Features



Phase event information, pyro continuity and fire confirmation, and separation events (using hall sensors)

GPS data provides altitude and distance, between the base and vehicle. It also provides bearing and azimuth angle and a large arrow direction indicator.

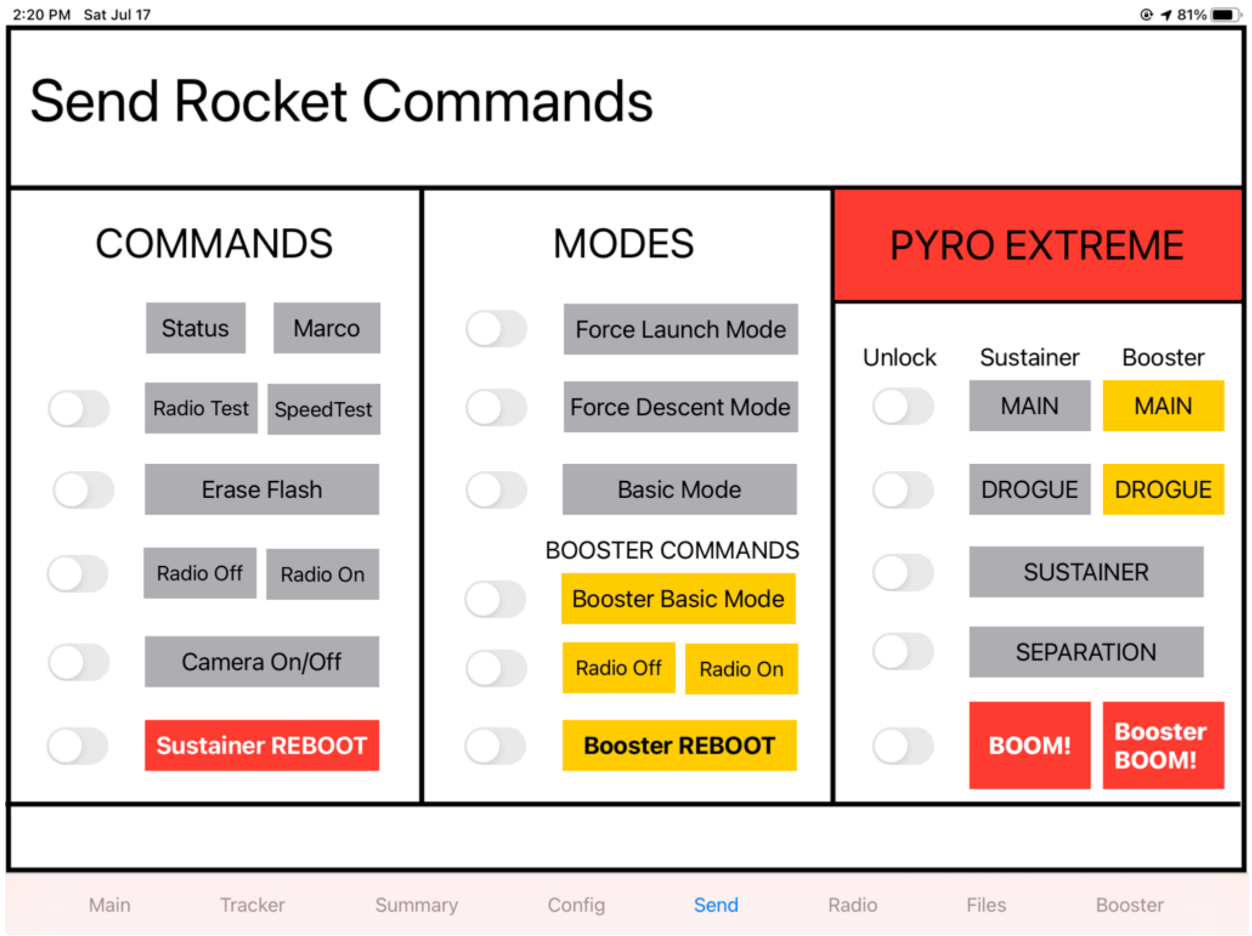
Base Station – Application Features



The tracking screen provides all the same GPS data, as well as satellite mapping.

Tracking also allows to pull location data live (from radio), from the log files, or by manually entering long/lat data.

Base Station – Application Features

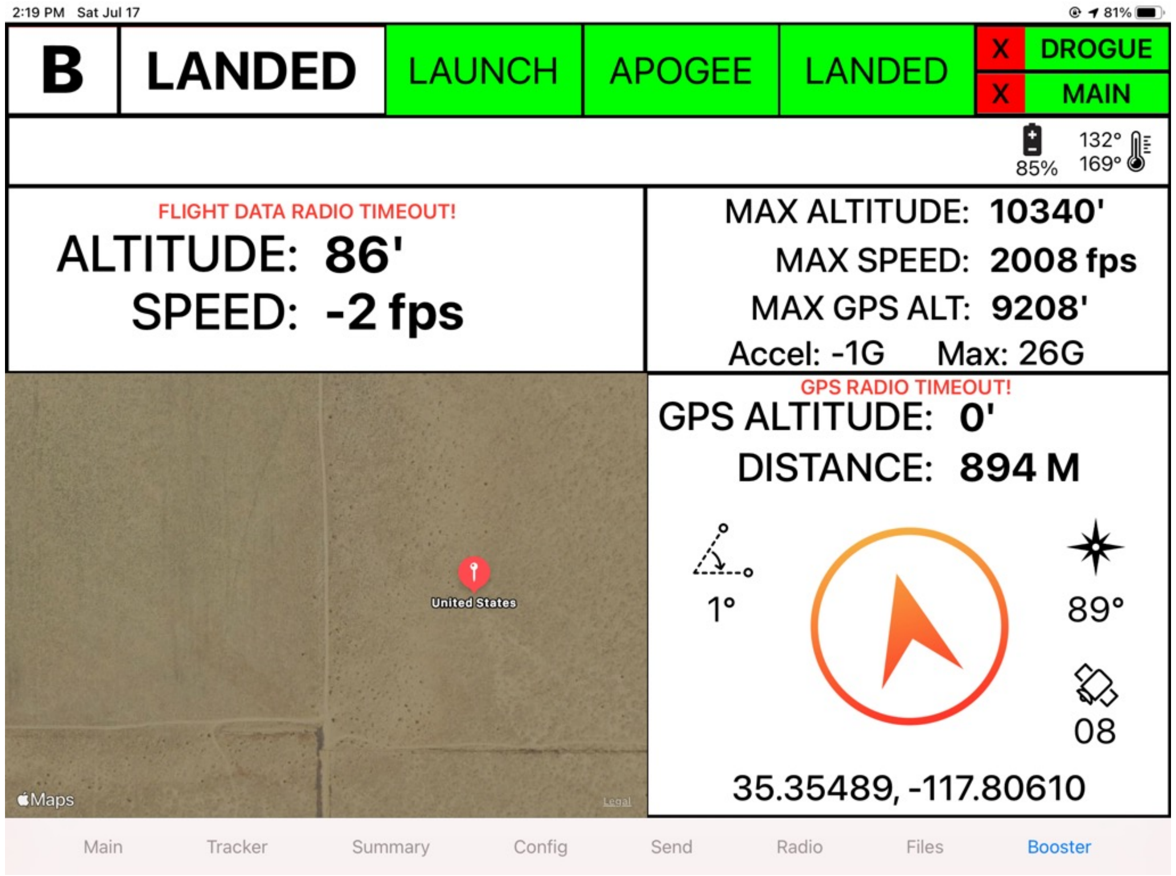


Commands can be sent to the sustainer or booster (separate vehicle board) for a number of different functions and modes.

Commands can be issued to manually fire the pyros or fire all the pyros (BOOM), as a poor mans FTS (flight termination system).

Flash can be erased, MCUs rebooted, radio TX can be turned off (good for speeding up GPS lock), etc.

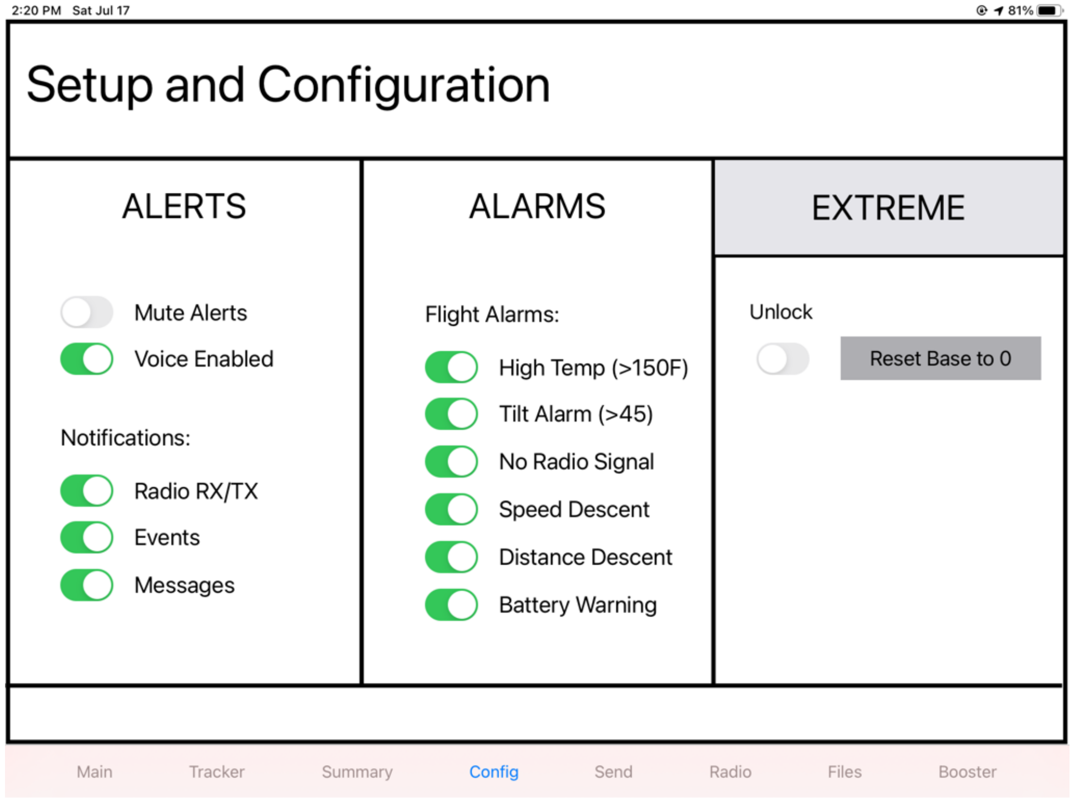
Base Station – Application Features



We have a smaller booster board that uses the same radio and frequency. So, we get both booster and sustainer reporting to the same iPad application.

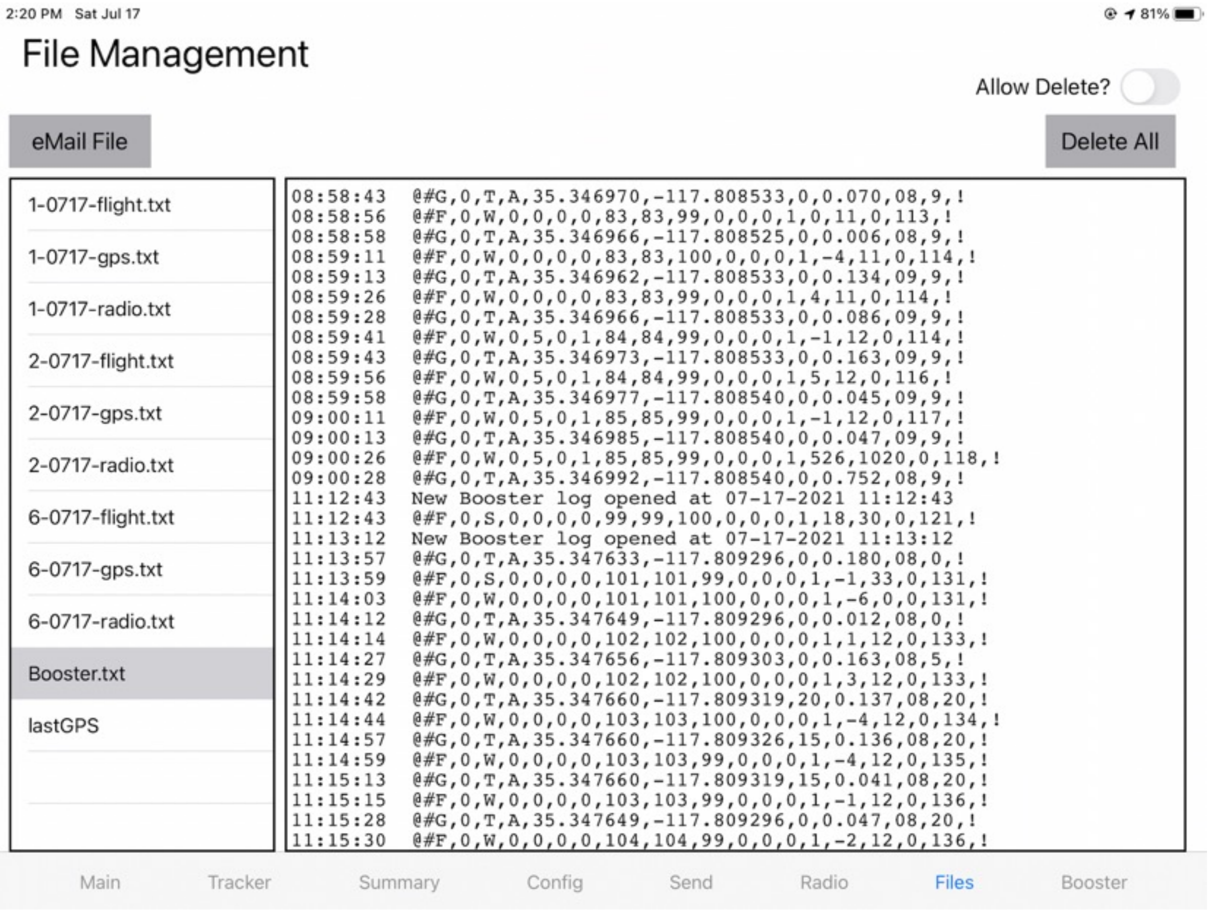
The booster screen has a lot of the same flight data and GPS tracking information, along with phase, event, and continuity confirmation on the top.

Base Station – Application Features



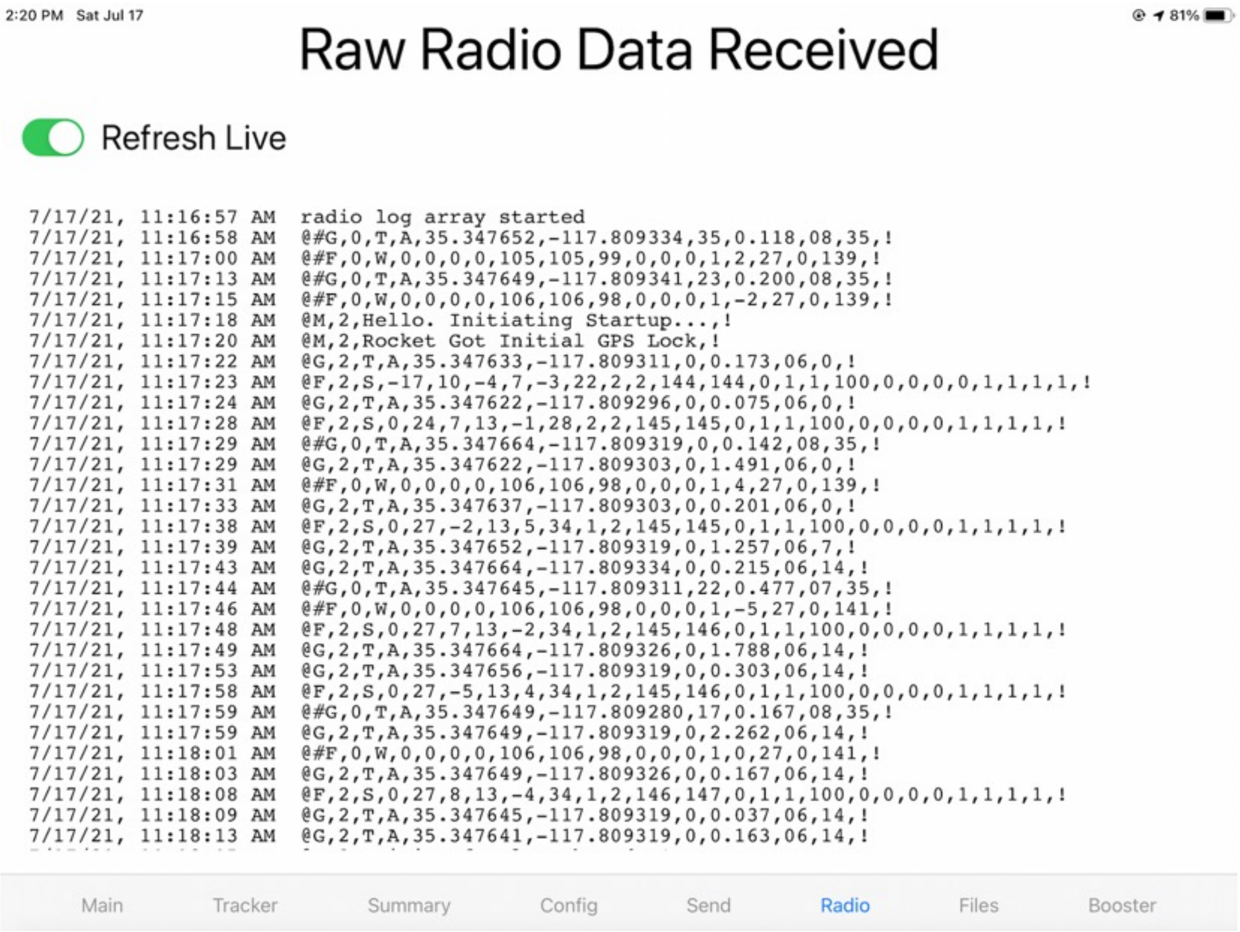
Basic settings for alerts and alarms

Base Station – Application Features



The base station logs all flight, GPS, and radio data. It logs Booster data separately. It also keeps a file for last know GPS locations.

Base Station – Application Features



Radio data is transmitted in comma delimited sentences with a stop and start character. The radio screen allows reading the radio data in real-time to confirm activity at a more technical level. Generally, this is only used for troubleshooting or diagnostics.

RocketTalk

OTHER

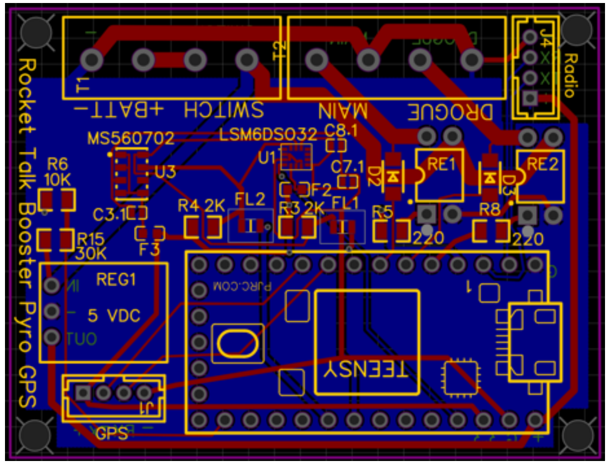
Radio Repeater

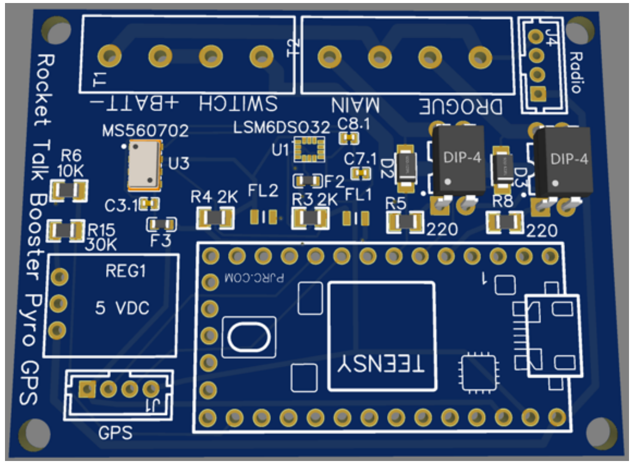
The beauty of LoRa radios is that they can be configured to work in “mesh” mode, as either nodes or repeaters. So, we have a NiceRF 5 watt LoRa6500Pro that we configure as a repeater on the same frequency and ID as our base and vehicle radios. Adding the repeater and putting an antenna high in a tower gives us exceptional range on the horizon (or in the dirt) with almost zero packet loss.

The repeater is passive, so the base and vehicle work just fine without it, but work even better when it is on.

We use a 4.5 foot Diamond X30A Dualband 2m/70cm Antenna with the repeater.

Booster Board





We have a mini version of the vehicle board that we use in the Booster. It is a subset of the main (sustainer) vehicle board, using a Teensy 4.0 (instead of 4.1), barometer, accelerometer/gyro, two pyro charge relays, a radio, and GPS.

It has no logging capability and no 200G ADXL375 accelerometer. It also is using only one 5V regulator.

This board uses the exact same radio, on the same frequency, so the handheld base station only needs one radio for both booster and sustainer communication.

The code for the booster is a subset of the sustainer, although radio commands/sentences are all prefixed with #, to differentiate between the booster and sustainer.

RocketTalk

PARTS REFERENCE

Vehicle Components

Teensy 4.1	https://www.pjrc.com/store/teensy41.html
“fully loaded” w/flash soldered	https://protosupplies.com/product/teensy-4-1-fully-loaded/
Ublox Max M10S GPS	https://www.digikey.com/en/products/detail/u-blox/MAX-M10S-00B/15712906
GPS Antenna M1575HCT-22P-SMA	https://www.mouser.com/ProductDetail/Maxtena/M1575HCT-22P-SMA
TE MS5607 barometer	https://www.digikey.com/catalog/en/partgroup/ms5607-02ba03/54690
ADXL 375 Accelerometer	https://www.digikey.com/en/products/detail/analog-devices-inc/ADXL375BCCZ/4376342
LSM6DS032 Accel/Gyro	https://www.digikey.com/en/products/detail/stmicroelectronics/LSM6DSO32TR/11694177?s=N4IgTCBcDaIDIGUCyA2AIgg8gZggXQF8g
NiceRF Lora 6100Pro Radio	https://www.nicerf.com/products/detail/1w-lora-wireless-data-transmission-module-lora6100pro.html
Rohm BV1HD090FJ-CE2 driver	https://www.digikey.com/en/products/detail/rohm-semiconductor/BV1HD090FJ-CE2/10249266
Toshiba TLP3553A opto-relay	https://www.digikey.com/en/products/detail/TLP3553A(F/TLP3553A(F-ND/9472359
LD1117 5v regulator	https://www.mouser.com/ProductDetail/STMicroelectronics/LD1117DT50CTR?qs=6MdFBD70OtGoLwD4bsdGtg%3D%3D
Lipo Battery 7.4v 850 Mah	https://www.amazon.com/gp/product/B07P2PDVM5/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
5v brown out cap (4F)	https://www.digikey.com/en/products/detail/vishay-beyschlag-draloric-bc-components/MAL219691104E3/5017510
Wurth DIY Shielding	https://www.digikey.com/en/products/detail/würth-elektronik/360002/10468233?utm_medium=email

Base Components

iPad Mini 64GB Wifi/Cellular	https://www.amazon.com/gp/product/B07PQ8WX75/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
Redpark Serial Cable	https://redpark.com/lightning-serial-cable-l2-db9v/
Serial RS232 to TTL	https://www.amazon.com/gp/product/B00OPTOKI0/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
NiceRF Lora 6100Pro Radio	https://www.nicerf.com/products/detail/1w-lora-wireless-data-transmission-module-lora6100pro.html
Arrow II Antenna 440-3	http://www.arrowantennas.com/arrowii/440-3ii.html
Tablet holder	https://www.amazon.com/gp/product/B071RK5FYY/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1
Camera Grip handle	https://www.amazon.com/gp/product/B07D3QDL1T/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

Questions?
@PAD33space on Twitter/X

or @AllDigital on TRF