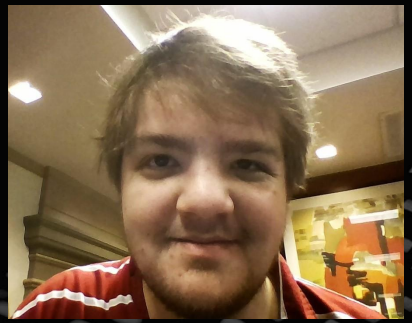
OSU AIAA

Oregon State University Chapter of the American Institute of Aeronautics and Astronautics

Team 41
Winter Progress Report



30k Spaceport America Cup 2017-18

Oregon State UNIVERSITY

Overview

Software and ground station components for the 2018 Spaceport America Cup 30K Challenge

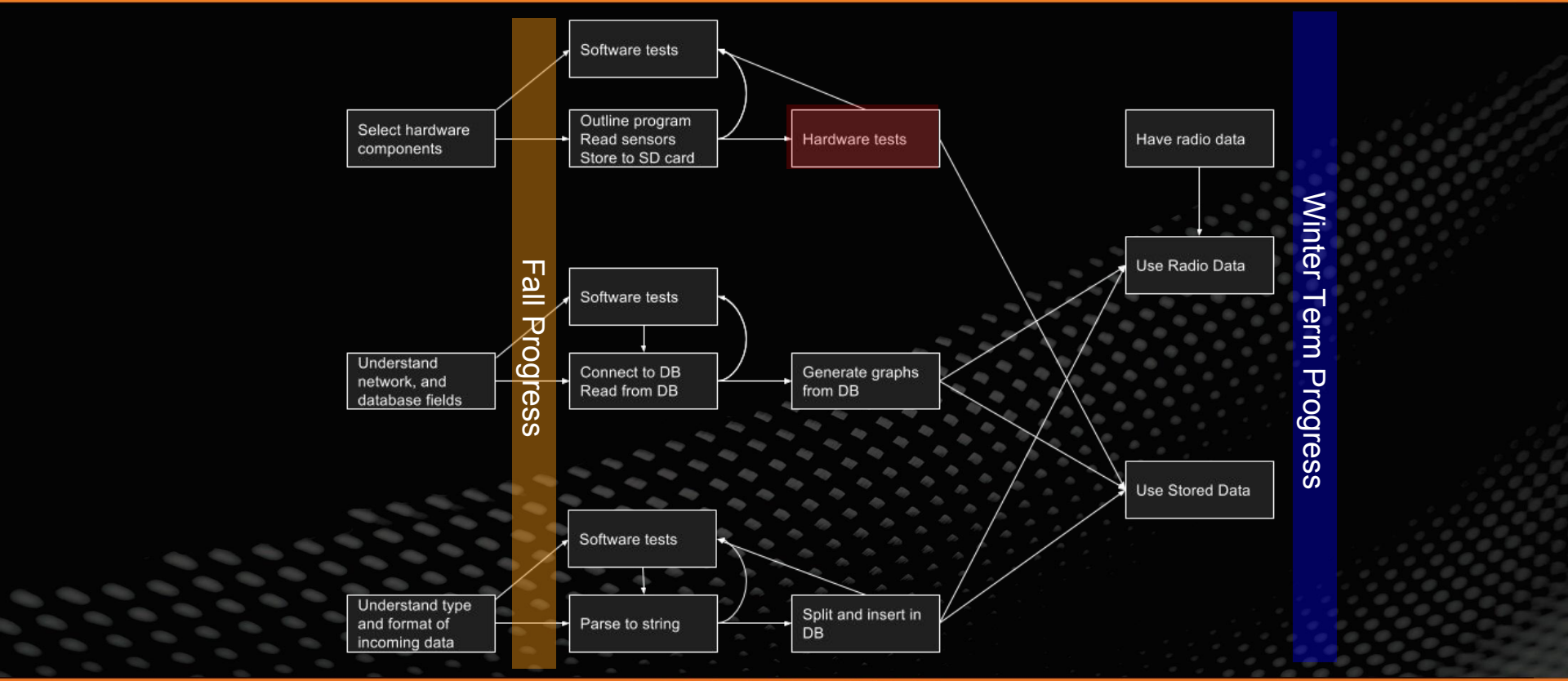
Our mission is to write flight avionics for both a rocket and scientific payload, as well as design a ground station capable of receiving and displaying live telemetry data from the rocket.

Project Goals

This project can be (greatly) simplified to three primary goals:

1. Design a ground station to receive and display flight data.
2. Write payload avionics to control a scientific experiment.
3. Write rocket avionics to log kinematics and detect apogee.

Development Roadmap



Ground Station

The physical build includes:

1 - 8”x10”x6” yellow case

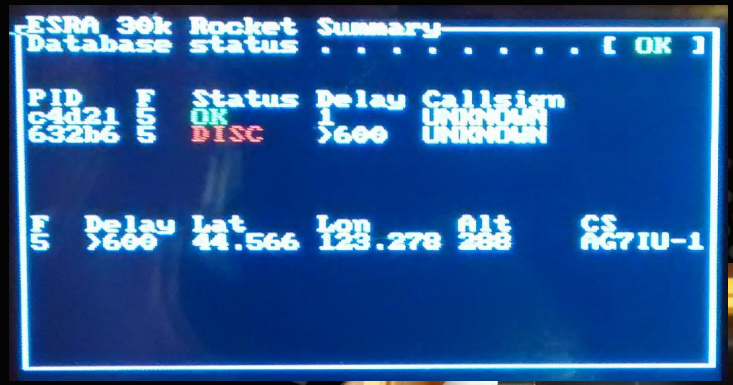
1 - Raspberry Pi B3

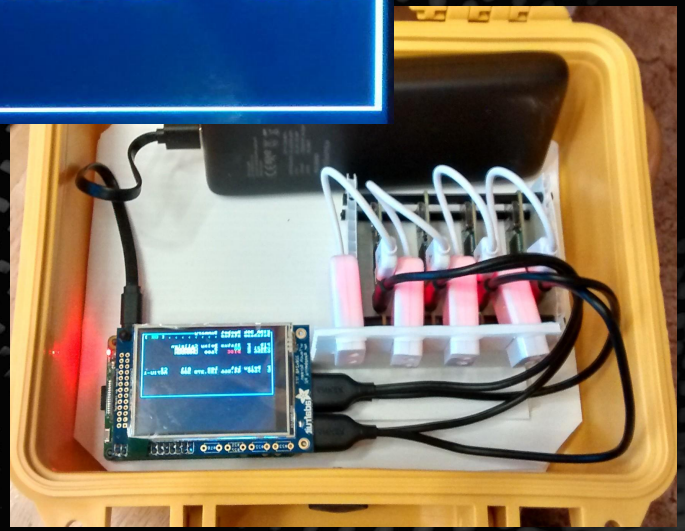
4 - Raspberry Pi Zero W

4 - USB sound cards

1 - 2.7” LCD Display

1 - 22,000 mAh rechargeable battery



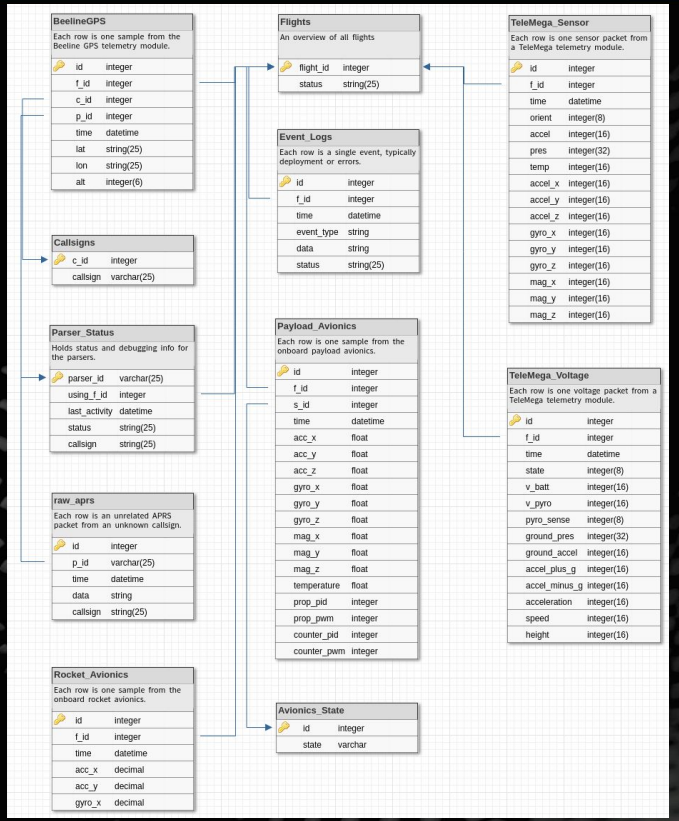


Ground Station

Ground Station - Database

Updated database schema:

- Calculating more, storing less
- Additional Avionics fields
- Using more M:M relationships
- Added TeleMega fields



Ground Station - Parsers

Overview:

- Each Zero listens to a single radio frequency
- Completed one end-to-end test using radio TX and RX
- Range tests performed using audio recorded by ECE subteam
- Performance comparable to COTS hardware decoders

Ground Station - Parsers

Neat Features:

- All Zeros use identical SD cards
- The cpu serial number uniquely identifies each Zero in the DB
- All Zeros run headless, the parsing program is a daemon
- Minimal text output, writes directly to database

Ground Station - Parsers

Parser-specific test results:

Name	Stmts	Miss	Cover
DWParser.py	33	1	97%
Mariadb.py	58	3	95%
parser.py	97	25	74%
TOTAL	188	29	85%

Ground Station - Networking

- Using USB OTG to connect the Pi zeros to the main Pi 3B
- Main Raspberry Pi serves a Wi-Fi Network
- Works for testing
- Stress test revealed issues that still need to be addressed

Ground Station - NodeJS

NodeJS website:

- Served by our main Raspberry pi 3
- A GUI for our graphs and flight data
- Implemented using express-handlebars
- Makes frequent queries to the database

Ground Station - Display

Graphs Overview

- Query database
- Use CanvasJS
- Update every second
- Can take inputs from numerous sources

Ground Station - Display

Line Graphs

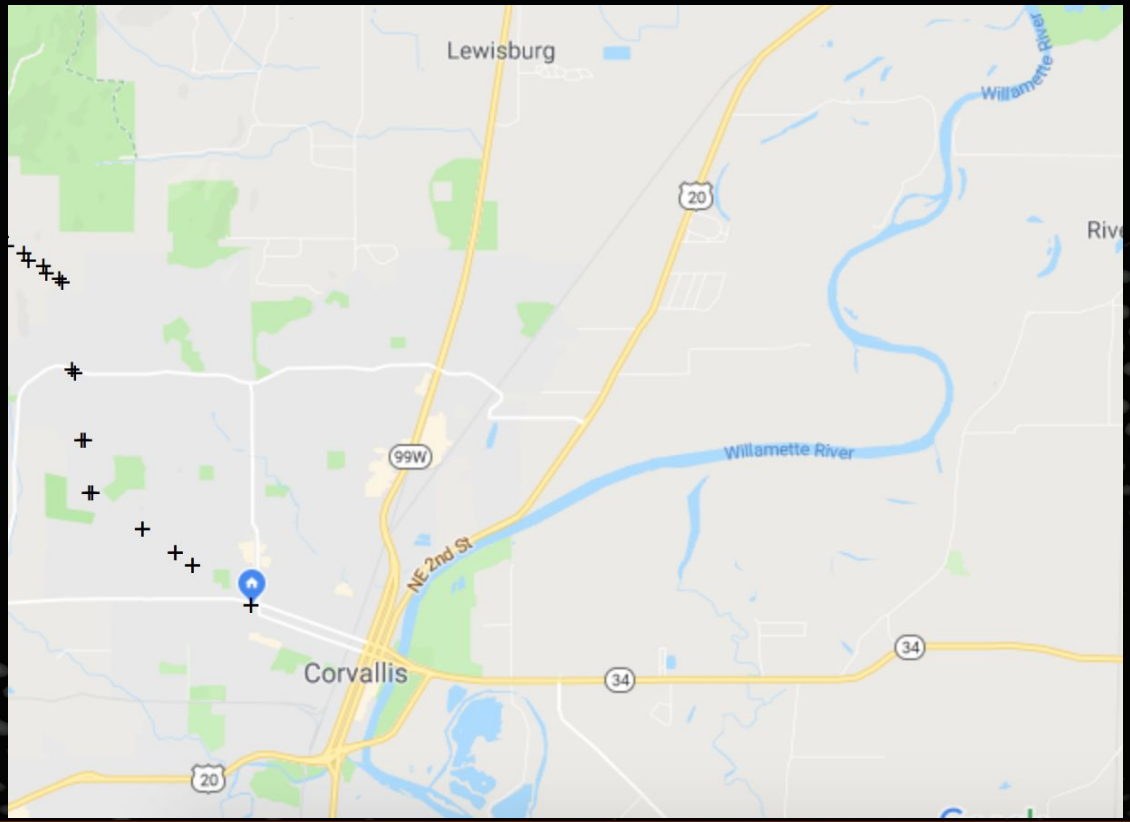
- Altitude vs Time and Vertical Velocity
- Use CanvasJS
- Use any number of sources
- Based on Flight ID

Ground Station - Display

Map

- GPS coordinate
- Calculate location based on map
- Any number of sources
- Based on Flight ID

Ground Station - Display



Ground Station - Display

Planned Features

- Use image for map
- Change color for map
- Acceleration graph
- True velocity/acceleration graphs
- Altitude or time in map

Sensor Avionics

Sensors Currently Implemented

- MPU - Accelerometer (Magnetometer functionality will not be used
- MPL - Altimeter
- PCF - Real Time Clock

Testing

Tests Currently Implemented

- Sensor
- Avionics
- Parser

Testing

Planned Tests

- Improvement in Avionics Tests
- Improvement in Parser Tests
- Simulated Launch w/ Randomized error

Payload Avionics

Our goal is to create 10-12 seconds of zero acceleration for a scientific experiment inside the payload.

Payload avionics record sensors on the payload as it descends.

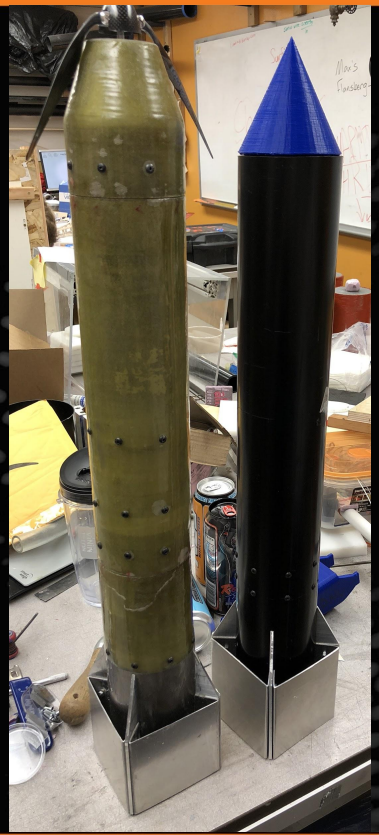
10” propeller and motor will reduce drag from air resistance.

Counterweight motor will reduce rotational forces.

Payload Avionics

The avionics loop currently:

- Reads 22 values from several sensors
- Calculates motor speed using PID loop
- Controls motor with a PWM output
- Logs data to CSV file



Rocket Avionics

Overview

- Very similar implementation to payload avionics
- Read and log sensor values
- Detect apogee

Rocket Avionics

Apogee Detection

- Mission critical task
- Can’t detect too early or too late
- Not being used for competition

Rocket Avionics

Apogee Sensors

- Altitude and Acceleration implemented
- Haven’t completed and end-to-end test
- Unit tests completed

Winter Term Progress