
HART Avionics

Release 0.0.0

OSU HART

Nov 16, 2020

CONTENTS

1	Introduction	1
2	Project Overview	3
2.1	Goals	3
2.2	Basic Requirements	3
3	Sequence of Events	5
3.1	Pre-Launch	5
3.2	Launch	5
3.3	In Flight	5
3.4	Stage Separation	6
3.5	Upper-Stage Ignition	6
3.6	Parachute Deployment	6
3.7	Recovery	6
3.8	Analysis	6
4	System Structure	7
4.1	Power Systems	7
4.2	Avionics	7
4.2.1	Flight Computer	7
4.2.2	Payload	8
4.3	Launch System	9
4.3.1	Launch Box	9
4.3.2	Control Box	9
4.4	Ground Station	10
4.4.1	Transceiver	10
4.4.2	Ground Computer	10
4.4.3	WiFi Router	10
4.4.4	GUI Server	10
4.4.5	GUI Client	11
5	How to Contribute	13

INTRODUCTION



This is the documentation repo for the Oregon State University AIAA High Altitude Rocket Team's ECE & CS avionics sub-team (OSU HART: Avionics-EECS). This covers any system-wide and user-level documentation for the project. For more detailed documentation on project internals and developer documentation, see the corresponding project's repository.

PROJECT OVERVIEW

2.1 Goals

The OSU HART team's ultimate goal is to launch a rocket up to 150,000 feet above ground level. Any change to the system must be done with this goal in mind. As part of the Avionics-EECS subteam, our goal is to work on the rocket's avionics systems as well as any supporting systems that move us closer to our ultimate goal.

2.2 Basic Requirements

The most basic requirements of a rocket avionics system are *triggering events* and *tracking the rocket*. Tracking the rocket will be how we determine if the rocket reached its goal altitude or not. There are several ways in which this step could go wrong or produce incorrect results, so redundancy and rigorous testing should be preferred. In addition to this, the rocket's avionics system must be able to trigger events under the right conditions in order to maximize the chances of reaching higher altitudes and, more importantly, the chances of successfully recovering the rocket.

SEQUENCE OF EVENTS

The rocket avionics system requirements are structured around the following sequence of events:

3.1 Pre-Launch

- Hardware Safety Check
- Configure Flight Software
- Software System Check
- Calibration

3.2 Launch

- Send launch command
- Ignite booster
- Detect launch

3.3 In Flight

- Record raw sensor data
- Record flight data
- Record rocket vitals (engine temperature/pressure)
- Record video
- Estimate current state (position/velocity/etc.)

3.4 Stage Separation

- Detect booster depletion
- Ignite separation charges

3.5 Upper-Stage Ignition

- Delay to save fuel and increase peak altitude
- Ignite sustainer

3.6 Parachute Deployment

- Detect apogee
- Trigger chute deployment charges

3.7 Recovery

- Transmit signal for locating the rocket once it lands
- Download flight data for further analysis

3.8 Analysis

- Run more computationally intensive algorithms on the recorded raw sensor data to determine if the flight computer is functioning properly.
- Replay the flight in greater detail
- Use data analysis techniques to create a model of the flight
- Compare flight model to predicted/hypothesized model in order to inform future decisions & improvements

SYSTEM STRUCTURE

4.1 Power Systems

The power systems power the electronics. The avionics are Outputs 3.3V or 5V to the microcontrollers.

4.2 Avionics

Avionics refers to all the electronics onboard the rocket.

4.2.1 Flight Computer

The Flight Computer is the hardware & software that triggers the rocket events.

Interface Definitions & Properties

Inputs

- Environment (External)
 - Acceleration
 - * Max: 50 G
 - Altitude
 - Attitude (pose/orientation)
 - GPS data
 - Magnetic Field
 - Temperature
- Power Systems
 - Vmin: 3 V
 - Vnominal: 3.7 V
 - Vmax: 16 V

Outputs

- Pyro Charges (External)
 - Vmin: 4 V
 - Vnominal: 12 V
 - Vmax: 16 V
- Ground Station Transceiver

Block Properties

We are currently required to use a commercial solution. Both the booster and sustainer have an Altus Metrum TeleMega and EasyMega each. The EasyMega is just a TeleMega without any RF capabilities and will be serving as a backup flight computer. Both of these systems are running AltOS. For more information about Altus Metrum products, visit the [Altus Metrum website](#).

4.2.2 Payload

The payload refers to all the electronics onboard the rocket that are not responsible for triggering the rocket events.

HART Flight Computer

The HART Flight Computer is the student-developed flight computer currently under development.

Interface Definitions & Properties

Inputs

- Environment (External)
 - Acceleration
 - * Max: 50 G
 - Altitude
 - Attitude (pose/orientation)
 - GPS data
 - Magnetic Field
 - Temperature
- Power Systems
 - Vmin: 3.3 V
 - Vnominal: 3.7 V
 - Vmax: 16 V

Outputs

- Ground Station GUI Server
 - Flight data structure matches AltOS data structure

Block Properties

We are currently required to use commercially available products to control the rocket events, but the plan is to eventually replace the commercially-developed flight computers with a student-developed solution.

Tracking Beacon

A way to track the rocket for recovery in case the flight computer's telemetry fails

Rocket Vitals

Monitor rocket internals, mostly for post-flight debugging

- Examples include motor temperature & pressure, stage-separation detection, parachute deployment detection, and dedicated state estimation

Camera

Record video through a window in the side of the rocket

4.3 Launch System

The launch system is the system that launches the rocket. We are temporarily using the OSU AIAA wireless ignition system until we have one built and tested for OSU HART.

4.3.1 Launch Box

Ignites the booster when launch signal received from Control Box

4.3.2 Control Box

The enclosure with the big red button

4.4 Ground Station

The telemetry visualization system with simulation capabilities

4.4.1 Transceiver

Communicates with the Avionics

- Antennas
- TeleDongle

4.4.2 Ground Computer

The Ground Computer runs the GUI Server and may run the GUI Client as well.

Interface Definitions & Properties

Inputs

- Power Systems
- Transceiver
- User (External)

Outputs

- Environment (External)

Block Properties

AltOS Raspberry Pi?

4.4.3 WiFi Router

The WiFi Router serves as the access point for the WLAN.

4.4.4 GUI Server

The Ground Server saves telemetry for later use, processes the telemetry, and serves the processed data to the GUI Client.

4.4.5 GUI Client

The GUI Client displays data in a GUI. It can run either on the Ground Computer or a computer connected to the Ground Computer.

HOW TO CONTRIBUTE

Contributions to the project are primarily done through modifications to the system structure, improvements to documentation wording, and evaluation of potential solutions. This includes adding, modifying, or restructuring blocks and interfaces as well as correcting spelling/grammar mistakes and writing research reports on a new design.

If you want to suggest a few quick changes, feel free to fork the repository, make your changes, and create a pull request describing your changes. Otherwise, please use the `.github/ISSUE_TEMPLATE` to create a new issue.