OSU
Oregon State
UNIVERSITY

College of Engineering

CS CAPSTONE REQUIREMENTS DOCUMENT

OCTOBER 31, 2018

HIGH ALTITUDE ROCKETRY PROJECT

PREPARED FOR

OSU AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS (AIAA)

DR. NANCY SQUIRES
_____ _____
Signature   Date

PREPARED BY

GROUP 11
HART CS CAPSTONE

RICK MENZEL
_____ _____
Signature   Date

MATTHEW FORSLAND
_____ _____
Signature   Date

Abstract

ABSTRACT This document outlines the customer requirements for the 2018-2019 HART Computer Science capstone team. The overall system as described will consist of a frontend, responsible for the collection and processing of raw telemetry data, as well as a frontend, which is in turn responsible for the visualization of telemetry and stage location. Both of these components will be implemented on a hardware groundstation.

CONTENTS

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the 2018-2019 HART CS sub-team software engineering project, and will serve simultaneously as a roadmap for project work and as a rubric for eventual evaluation. It will describe the purpose and scope of the project and specific requirements of the completed system. It will also detail the interface and the system's interactions with the HART ECE avionics system.

1.2 Scope

The HART CS groundstation is a system which receives telemetry data transmitted wirelessly from two remote systems located on separate stages of a 2-stage rocket. The job of the groundstation is to interpret raw telemetry data, calculate derived values, and provide an application which visualizes this data, all in the service of allowing the ground team to track and recover the rocket stages and determine whether the target altitude is reached. These displays will take the form of both 2-dimensional graphs (for parameters including altitude, vertical and horizontal velocity, acceleration, barometric pressure and thrust) and also a 3-dimensional visualization of the rocket stage trajectories.

1.3 Glossary

- **Avionics** The guidance system for the rocket, which controls engine ignition and parachute deployment. The avionics system also transmits telemetry to a groundstation.
- **Backend** The non-user-facing components of the ultimate system; responsible for the collection, processing and storage of telemetry data in real-time.
- **Frontend** The user-facing components of the ultimate system; responsible for the real-time display of telemetry data, as well as the visualization of rocket stage locations.
- **HART** High Altitude Rocketry Team
- **Karman Line** An altitude of 100 km above sea level; commonly understood as the boundary between the Earth's atmosphere and outer space.
- **Non-Volatile Storage** A type of computer memory capable of preserving data without the need for constant power. Also known as persistent storage.
- **Sub-Team** A disciplinary or task-level subdivision of the larger multidisciplinary HART team.

1.4 Overview

The following two sections provide a description of the necessary system functions and of system interactions with external libraries and data sources. The immediately following section *Overall Description* discusses the system in broad terms and details a number of important design considerations. The final section *Specific Requirements* enumerates and breaks down specific requirements and expectations for the final system.

2 OVERALL DESCRIPTION

2.1 Product Perspective

This software will consist of a server backend which receives telemetry data broadcast by avionics systems located on both stages of a 2-stage rocket and interprets that data into additional derived values. The server backend will communicate with a frontend application which provides visualization for raw and interpreted data in 2-dimensional graphs, and generates a 3-dimensional model for the location and trajectory of both stages of the rocket throughout the duration of the flight.

2.2 Product Functions

The final deliverable can essentially be broken down into two main components, hereafter referred to as the backend and the frontend.

2.2.1 Backend

The first component, the backend, will be responsible for the receiving and subsequent processing of raw telemetry. This implies two primary functions for the backend: first, the processing of telemetry into a form usable by the frontend, and second, the storing of telemetry for later revisiting. The ability to store data is a critical consideration, as this year's launch effort is only a step toward the ultimate goal of launching a rocket built by OSU students to above the Karman Line. As such, all information gathered during this year's project should be preserved and made available to subsequent teams, a requirement which necessitates the transfer of data to non-volatile storage.

2.2.2 Frontend

The second component, the frontend, will in turn be responsible for displaying the processed telemetry in a meaningful and useful manner. This portion of the product can be further broken two into two sub-components. The first sub-component shall display the in-flight characteristics of the airframe including altitude, vertical and horizontal velocity, acceleration, barometric pressure and thrust in a series of continuously-updated 2-dimensional graphs or numerical readouts. Notably, this sub-component with feature little if any user interaction. The second sub-component then will be responsible for the 3-dimensional visualization of the two rocket stages and their respective trajectories. Given that a 3D flight path can proceed in any number of lateral directions, it will likely become necessary to adjust the eye-position and/or viewing angle to maintain the utility of this sub-component. Functionally, this means that a degree of user-interaction is called for.

2.3 User Characteristics

The types of users who will use this software fall into two main categories. The first type of user includes personnel operating the rocket, who require the groundstation to accurately receive telemetry broadcasted by the rocket in real time, and provide an accurate location of the rocket once landed to facilitate quick recovery of the rocket. The operators will also need post-flight access to the received telemetry to analyze the performance of the rocket. The second type of user includes observers using the web application to monitor the condition of the rocket during flight.

2.4 Constraints

The software must receive telemetry data broadcast in real-time. The software must store all received and interpreted data for later use. The graphical user interface must be reliable and dependable. The entire system should be capable of operating on minimal hardware.

2.5 Assumptions and Dependencies

For this system to operate there are several key assumptions to consider:

2.5.1

First and foremost is the ability to actually launch the final rocket. Given the nature of rocket flight, there are a number of factors beyond our sub-team's control which have the potential to interfere with a full-scale launch, most notably the weather, as poor weather on the day of launch has the potential to scrub the entire mission. A further caveat to this is that a full-scale launch to our target altitude cannot be conducted locally. This restriction necessitates that HART travels to Spaceport America in New Mexico for full-scale, full-altitude launches, a necessity which is subject to budgetary and logistical considerations.

2.5.2

The second assumption is that an air-worthy rocket is ultimately produced. Our project is part of a multi-disciplinary effort and we are reliant upon several ME teams for the design and construction of the actual airframe. The failure to complete the airframe or a catastrophic failure upon launch has the potential to render telemetry capture impossible and irrelevant.

2.5.3

The third assumption is that the HART ECE sub-team's avionics system will be fully operational and able to broadcast telemetry for any flights during which the groundstation software is used. Even assuming that the avionics system is completed, a component failure while in-flight or high levels of electromagnetic interference have the potential to disrupt the groundstation's ability to reliably receive telemetry. Note that we are proceeding under the further assumption that the ECE sub-team will be installing a Telemega chip set in the ultimate rocket.

2.5.4

Finally, we are operating under the assumption that due to budgetary considerations we will be re-using the groundstation hardware from previous years, which runs a Linux distribution on a Raspberry Pi board.

2.6 Apportioning of Requirements

Currently our final launch is scheduled for June 2019, though all systems should be functionally complete in time for a potential full-scale test launch in April-May 2019. Sub-scale tests beginning in Winter term 2019 will also provide an opportunities to evaluate system components for proper function should they be ready at that time, In the case that the project is delayed, some requirements involving the frontend can be dismissed or delayed. Most notably, provided that telemetry data is stored by the groundstation, the 2-dimensional visualization can be performed by third-party software without the use of a specialized web-application. Analysis of captured data could also potentially be completed *after* the flight rather than in real-time. See Gantt Chart, Figure 1.

3 SPECIFIC REQUIREMENTS

3.1 Interfaces

As discussed above there will be two primary means of user interaction with this system, each having a varying degree of actual interaction. Both means will require the user to connect to the groundstation via a web browser or similar interface. The first point of user interaction will be strictly passive, and will involve the display of various flight characteristics via a series of 2D graphs and numerical readouts. The second will involve the 3D visualization of the flight path and location of stages, and will offer a more dynamic user experience allowing for the manipulation of eye-position and/or viewing angle.

3.2 Functional Requirements

This section includes the requirements that specify the fundamental actions of the software system. Furthermore, it describes the necessary outcomes that are expected by the client.

ID: FR1

TITLE: Acceleration Display

DESC: The software should display the current acceleration of the rocket.

RAT: The ability to display the current acceleration being experienced by the rocket will allow the ground team to monitor for emergent conditions which may threaten the safety of the airframe or ground personnel. Additionally, acceleration information will be useful for monitoring stage progression to include recovery deployment success.

DEPEND: None

ID: FR2

TITLE: Altitude Display

DESC: The software should display the altitude of the rocket.

RAT: Given that the goal of HART is to launch a rocket to a minimum altitude, the display of current and peak altitude will allow the team to monitor progress and determine success on the day.

DEPEND: None

ID: FR3

TITLE: Barometric Pressure Display

DESC: The software should display the current barometric pressure experienced by the rocket.

RAT: The ability to display the current barometric pressure being experienced by the rocket will allow the ground team to monitor for emergent conditions which may threaten the safety of the airframe or ground personnel.

DEPEND: None

ID: FR4

TITLE: Thrust Interpretation

DESC: The software should interpret the acceleration telemetry into the thrust provided by the rocket motors.

RAT: Providing the thrust curve produced by a rocket motor allows users to analyze the rocket motor's performance.

DEPEND: None

ID: FR5

TITLE: Thrust Display

DESC: The software should display the current thrust level of the rocket.

RAT: The ability to display the current acceleration being experienced by the rocket will allow the ground team to monitor for emergent conditions which may threaten the safety of the airframe or ground personnel. Additionally, thrust information will be critical for monitoring stage progression.

DEPEND: FR4

ID: FR6

TITLE: Velocity Interpretation

DESC: The software should interpret the acceleration and altitude telemetry into velocity data.

RAT: The telemetry data will provide only information pertaining to the rocket's orientation, acceleration, altitude, and GPS position. Determining the velocity of the rocket allows users to correlate changes in performance with the passing of specific velocities, such as Mach one.

DEPEND: None

ID: FR7

TITLE: Velocity Display

DESC: The software should display the current vertical and horizontal acceleration of the rocket.

RAT: The ability to display the current vertical and horizontal acceleration of the rocket will allow the ground team to monitor for emergent conditions which may threaten the safety of the airframe or ground personnel.

DEPEND: FR6

ID: FR8

TITLE: Flight Path and Location Display

DESC: The software should display the flight path and current location of the rocket.

RAT: The ability to display the flight path and location of the rocket will allow the ground team to monitor for emergent conditions which may threaten the safety of the airframe or ground personnel. Additionally, stage location information will be critical to airframe recovery efforts.

DEPEND: FR1, FR6

ID: FR9

TITLE: Data Storage

DESC: The software should record all processed data onto non-volatile storage.

RAT: As this year's effort is only a stepping stone toward the ultimate goal of launching a student-built rocket to the Karman Line, it is critical to make all the data gathered available to future teams in order to inform their subsequent designs.

DEPEND: FR4, FR6

3.3 Performance Requirements

This section provides a detailed specification of user interaction with the software and performance metrics for the software.

ID: QR1 TITLE: Telemetry Reception

DESC: The software should receive telemetry in real-time.

RAT: The avionics systems will be broadcasting telemetry in real-time. If the groundstation software is not receiving data at the same rate, packets of telemetry data will be lost.

DEPEND: None

ID: QR2 TITLE: Real-time Visualization

DESC: The visualization application should update with telemetry data in real-time, as it is received by the groundstation.

RAT: Monitoring the rocket's status while it is in flight but not visually tracked is important in determining if the parachutes have deployed, as a failure at this stage poses a significant safety hazard to personnel on the ground.

DEPEND: QR1

ID: QR3

TITLE: Interactive Visualization

DESC: The 3-dimensional trajectory and location visualization should be interactive to allow the user to adjust the eye-position and/or viewing angle.

RAT: Given that a 3D flight path can proceed in any number of lateral directions, it will likely become necessary to adjust the eye-position and/or viewing angle to prevent display occlusion and aid in meaningful location association.

DEPEND: None

ID: QR4

TITLE: Data Accessibility

DESC: Data should be stored in a manner which is accessible to other users via standard text editor applications, and in a form which is conducive to meaningful processing via means such as scripts.

RAT: As this year's effort is only a stepping stone toward the ultimate goal of launching a student-built rocket to the Karman Line, it is critical to make all the data gathered available to future teams in order to inform their subsequent designs.

DEPEND: None

4 GANTT CHART

	2018				2019						
	Sep	October	November	December	January	February	March	April	May	June	

Task	Schedule
Problem Statement	Oct 15–22
Customer Requirements	Oct 22 – Nov 05
Tech Review, Draft 1	Nov 05
Tech Review	Nov 05–19
Team Standards	Nov 05
Design Document	Nov 19–26
Tech Research and Development	Nov 19 – Dec 03
FR1: Acceleration Display	Jan 07–21
FR2: Altitude Display	Jan 07–21
FR3: Barometric Pressure Display	Feb 04–18
FR5: Thrust Interpretation	Feb 04–18
FR5: Thrust Display	Feb 25 – Mar 11
FR6: Velocity Interpretation	Jan 07–21
FR7: Velocity Display	Jan 14–28
FR8: Flight Path and Location Display	Mar 04–18
FR9: Data Storage	Jan 07–21
QR1: Telemetry Reception	Feb 04–18
QR2: Real-time Visualization	Mar 04–18
QR3: Interactive Visualization	Feb 04–18
QR4: Data Accessibility	Jan 07 – Jun 03
Testing	Jan 07 – Jun 03
EXPO	May 27
Spaceport America Cup: Presentation	Jun 22
Spaceport America Cup: Flight	Jun 22