CS CAPSTONE PROBLEM STATEMENT

OCTOBER 24, 2017

30K ROCKET SPACEPORT AMERICA

PREPARED FOR

DR. NANCY SQUIRES

Signature   Date

PREPARED BY

30K CS AVIONICS

JOSHUA NOVAK

Signature   Date

ALLISON SLADEK

Signature   Date

LEVI WILLMETH

Signature   Date

Abstract

The Spaceport America Cup is an international engineering competition to design, build, and fly a student-made rocket to 30,000 feet. The competition is scored on several criteria including the software components outlined by this problem statement. These software components control the rocket avionics, record and display near real time telemetry, and later display the results from an scientific payload.

CONTENTS

1 PROJECT OVERVIEW

This computer science (CS) capstone project will include writing software to support the Oregon State University (OSU) American Institute of Aeronautics and Astronautics (AIAA) team's entry for the Spaceport America Cup 30k competition in the summer of 2018. The competition involves designing, building, and launching a student-made rocket to 30,000 feet.

2 PROBLEM DEFINITION

Our goals for entering the Spaceport America Cup are to learn and experience working together as a team to build a rocket, and to also score highly during the competition. As such, the scoring criteria are very important to forming the needs of our project. The exact scoring metric has not been released yet, but we expect the software components will include avionics to control the rocket, receiving near real time telemetry on the ground during the flight, and visualizing the results of an onboard scientific payload after the flight. Our task will be to write the software necessary to accomplish those goals.

The AIAA team includes several sub teams that will all need to work together. In particular, we will need to work with the electrical and computer engineering (ECE) students who are responsible for developing the avionics hardware and telemetry radio systems. The ECE subteam is ultimately responsible for those systems, but we will provide support to write the necessary software. It is important to support the overall team because at the competition we will all succeed or fail together.

3 PROPOSED SOLUTION

3.1 Develop a ground station to receive telemetry

To track the rocket and payload, we propose writing a ground station program to parse and display telemetry data sent to the program as an audio source or computer file.

At a minimum, the telemetry data is expected to include a unique identifier for the rocket or payload, timestamp, and GPS latitude, longitude, and altitude, but it may also include additional fields. We are only responsible for receiving, not transmitting this telemetry data.

We will display the telemetry data in near real time in such a way that it can be used to locate the equipment if we have a successful flight, or help identify problems with the launch in the event the rocket or payload are not recoverable. We may display the data as a series of way points on a map of the local area, with accompanying graphs or figures to show useful information such as maximum altitude or calculated velocity.

The rocket and payload may both use primary and secondary telemetry signals, in which case the program may need to select a single source and run on more than one computer. This may be necessary if, for example, the audio signals are on different frequencies or overlap in timing.

3.2 Interpret the scientific payload data

The project is still developing and we are not yet certain what type of data the scientific payload will collect. However, we expect that it will include a GPS transmitter similar to the rocket, as well as some number of additional scientific measurements which will probably include several fields of acceleration. The rocket and payload may each record different sets of data and if so, we will be expected to analyze data from each vehicle independently.

This data will be stored on a micro SD card that we can copy after the flight. We will write a program to graph or otherwise display the recorded data based on the details of the experiment and Spaceport America competition scoring rubric.

3.3 Support development of avionics software

We will work closely with the ECE subteam to develop the software necessary for avionics. This system is responsible for triggering the separation of the rocket at apogee, deployment of the scientific payload and drogue parachute, and the later deployment of the main parachute. These are all mission-critical and time-sensitive tasks that MUST occur for a successful flight.

A suite of unit tests will be generated for this program, including the ability to read in a complete set of sample data to simulate a launch. We will use these tests to ensure that our program ONLY triggers the separation and deployment stages at the proper times.

This avionics software is mission-critical. Aerospace companies take testing very seriously, and so will we.

3.4 Support the AIAA team as a whole

We will attend regular team meetings and try to understand the problems and challenges faced by the other subteams. If we see an opportunity to help another subteam with something, we will do our best to do so. We will also attend the majority of team building and training exercises, as well as all practice rocket launches.

Sometimes, supporting the rest of the team means asking for help from them. In the event that we are the ones who need support, we will talk with our team members and ask for help. We will stay in communication with our primary sponsor Dr. Nancy Squires, and other team mentors. We will treat ourselves and other team members with respect, and do our best to create a good working environment both online and in person.

4 PERFORMANCE METRICS

Many aspects of this project could change between now and completion. As such, these performance metrics are based on current knowledge and may need to be updated at a later date. The performance metrics for the computer science subteam will not require the successful completion of components by any other subteam.

4.1 Tracking the rocket and payload

Demonstrate the ability to track a rocket and payload, given a set of viable inputs. These inputs may be in the form of properly formatted radio signals, or as a computer file of sensor values. The program should record and display the launch location, a series of way points, and the final location of the vehicle. We will not be expected to generate radio signals.

If there are multiple radio signals available, for example primary and secondary signals, the program may be applied selectively to only one or more of the signals with the expectation that a second copy of the program may be used to record the other signals. This may be necessary if, for example, we have four simultaneous radio signals on different frequencies. In that situation we may need to run the program on two or more computers to successfully process the multiple audio sources.

4.2 Displaying data recorded during flight

Demonstrate reading sensor data from an SD card, and display that data as a set of graphs. This could be combined with the above tracking program, or be written as a separate program. The program should handle invalid sensor inputs by warning the user and ignoring those inputs. The program should be able to display all fields of data generated by the avionics hardware during a normal flight.

4.3 Hardware simulation and suite of unit tests

Our team will produce a test suite that will cover at least 80% of the lines of code written for the ground station and avionics software. Depending on the language used, test cases may be a combination of manually written tests or tests generated using automated or mutation-based software.