CS Capstone Requirements Document

October 27, 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

Client requirements for the computer science portion of the Oregon State University's entry into the 30k Spaceport America Cup in 2018. The software components outlined in this requirements document control the rocket avionics, record and display live telemetry from the rocket, and record and display results from the scientific payload.

CONTENTS

Revision History

Name	Date	Reason For Changes	Version
J.Novak, A.Sladek, L.Willmeth	10/26/17	Initial document draft	0.1

1 Project Overview

1.1 Introduction

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 software components like flight avionics, recording and displaying telemetry, and later displaying the results from an scientific payload.

1.2 Purpose

This document outlines the software requirements for the Spaceport America Cup 30k rocket competition in 2018.

1.3 Scope

The software described by this document will 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.

1.4 Definitions, acronyms, and abbreviations

AIAA	American Institute of Aeronautics and Astronautics
CS	Computer Science
ECE	Electrical and Computer Engineering
GUI	Graphical User Interface
GPIO	General Purpose Input Output pin
GPS	Global Positioning System
OSU	Oregon State University
PCB	Printed Circuit Board
SD card	Secure Data card
TBD	To Be Determined

1.5 References

Initial Project Description

http://eecs.oregonstate.edu/capstone/cs/capstone.cgi?project=340

Client Requirements Document assignment

http://eecs.oregonstate.edu/capstone/cs/capstone.cgi?hw=reqs

CS capstone group final Problem Statement

https://github.com/OregonStateRocketry/30k2018-CS-Capstone

1.6 Overview

This document contains the complete software requirements and specifications for the computer science portion of the Spaceport America Cup 30k rocket competition. It is organized into groups of user stories and characteristics, specific product functions, constraints, and assumptions about program inputs and outputs.

2 Overall description

2.1 Product perspective

The tracking and ground station program is designed to receive data from a file, or an external source (telemetry from the rocket) in the form of a radio signal. The tracking program will parse the radio signal into a well-formatted string that includes several pre determined and delimited fields of data, as agreed on by both the CS and ECE teams. The program will graph or otherwise display the telemetry data within 5 seconds of receiving the radio signal.

The payload data analysis program will accept an input file from an SD card, in a format agreed on by both the CS and ECE teams. This data will be used to graph or otherwise display the data collected during the flight. This data may include sensors or other results from the zero gravity experiment.

The CS team will support the development of avionics software that will be dependant on the hardware selected or designed by the ECE team. This software will need to read various sensors to determine when to separate the rocket, eject the payload, and trigger the drogue and main parachutes. This software may also record data from the sensors to an SD card for later analysis.

2.1.1 System interfaces

The avionics software will interact with the rocket through the PCB designed by the ECE team.

2.1.2 User interfaces

The user interfaces for our software will be limited. Users will be able to use the GUI to view recent telemetry data. They will also be able to read the graphs generated from the payload data to draw conclusions about its flight and experimental results.

2.1.3 Hardware interfaces

The parsing software will interact with a radio device to collect telemetry data. The avionics software will interact with the rocket through a PCB designed by the ECE team, which will provide sensors that will probably include at least one accelerometer, gyroscope, and barometer, as well as multiple explosive charges to separate the rocket, and deploy drogue and main parachutes.

2.1.4 Software interfaces

We may use external libraries or software to create graphs, especially from the data recorded on the scientific payload. The parsing program will read and write to a local database.

(stretch goal) We will allow other computers or devices to connect to the computer running the parsing program, to view telemetry data and payload graphs via an intranet page. This will require configuring wifi networks and generating dynamic graphs from a database.

2.1.5 Communication interfaces

Telemetry for the rocket will be transmitted as a radio signal. The specific format and fields of the data will be determined by both the CS and ECE teams.

2.1.6 Memory constraints

Avionics software will need to execute on a computer system to be determined by the ECE team, which is likely to be a micro controller. All other software can be run on laptops with varied memory capacity.

2.1.7 Operations

The telemetry or analysis programs will allow the user to load data from a file, which will create static displays that do not change over time.

The telemetry program will also allow the user to monitor a radio signal as an input source, which will create a dynamic display as new data is received.

The avionics program does not offer a user interface because it operates solely based on sensor inputs.

2.2 Product functions

2.2.1 Telemetry Program

Import a data file from hard drive or SD card.

This can be used to either test a set of telemetry data, or display a previously completed flight.

Display one or more graphs or text fields of the telemetry data.

2.2.2 Payload analysis

Import a data file from hard drive or SD card.

This can be used to either test a set of telemetry data, or display a previously completed flight.

Display one or more graphs of the data recorded during flight.

2.2.3 Avionics

Read data from individual sensors

Toggle a GPIO pin to trigger rocket separation

Toggle a GPIO pin to trigger payload ejection

Toggle a GPIO pin to deploy drogue parachute (if needed)

Toggle a GPIO pin to deploy main parachute (if needed)

2.3 User characteristics

The intended users of this software will consist primarily of other team members and mentors. They will have technical knowledge related to operation and construction of the rocket and its sensors. These users are all seniors in college or beyond. The nature of the graphical data will be technical. The software team will also be present at launch to assist in the use of the software.

2.4 Constraints

a)Regulatory policies;

The project may be subject to competition restrictions.

The radio transmissions must comply with federal communications commission regulations.

b)Hardware limitations (e.g., signal timing requirements);

The computer hardware will be determined by the electrical engineering subteam of the 30k ESRA rocketry team.

The program requires an external radio to receive radio signals.

c)Interfaces to other applications;

The telemetry and analysis programs require an external data source.

d)Parallel operation

Multiple copies of the telemetry program can run simultaneously on different machines, without interfering with each other.

e)Audit functions

The telemetry and avionics programs will include a test suite covering at least 80% of the CS teams lines of code.

f)Control functions

Control functions are limited by the ECE and rocket hardware.

g)Higher-order language requirements

The telemetry software may be written in a high level language.

The avionics software will probably be written in C.

h)Signal handshake protocols (e.g., XON-XOFF, ACK-NACK)

i)Reliability requirements;

GPS tracking must be reliable to ensure recovery of the rocket. Computations for display and parsing of data must be accurate.

Separation must be triggered at the right time consistently.

j)Criticality of the application;

Displaying recent GPS data will be mission critical for recovery of the rocket. Interpretation and display of other data, including telemetry and payload is less critical, but still very important to the success of the launch. Separation at the right time is critical for recovery of the rocket.

k)Safety and security considerations.

Successfully tracking the rocket and payload are mission-critical.

Successfully timing the separation of the rocket, and deplying the payload and both parachutes are mission-critical.

2.5 Assumptions and dependencies

Rocket will launch far enough away from spectators not to pose a safety threat. The ECE team will provide the computer hardware necessary to run the avionics software and record data for use in the analysis program. The ECE team will create a radio signal containing accurate and well-formatted telemetry data.

3 Specific requirements

3.1 External interface requirements

The parsing program will need to be connected to an input source capable of receiving a properly formatted radio signal. Formatting may include audio signals in APRS format, as a serial string, or some other as-yet unknown but acceptable format.

3.1.1 User interfaces

The parsing and payload programs will include either text-based or graphical user interfaces.
The avionics program may or may not include a text-based interface.

3.1.2 Hardware interfaces

The avionics program will interface with a custom PCB designed and manufactured by the ECE team.

3.1.3 Software interfaces

The parsing and payload programs will be able to load a data file from an SD card or Linux operating system.

3.1.4 Communications interfaces

The parsing program will interface with the rocket through radio telemetry.

3.2 System features

3.2.1 Parser

The parser will interpret information received via radio signal. It will decode the message from an encoding to be determined by the ECE team and translated into a row to be added to a database. It will also check to ensure that the message received is actual telemetry data and possesses all of the required fields. If it appears to be invalid, not possessing the required fields, or inaccurate, falling vastly outside of the predicted scope in any field, it will be discarded. This would include a GPS location that falls outside the airspace of the test, a negative temperature, or an altitude that is several hundred feet below ground or over a hundred thousand feet above ground. If any field contains an unexpected character, the information will be discarded.

The parser will have a test suite with at least 80% line coverage for the code the CS team writes.

The parser will be expected to push information to the database at regular enough intervals to ensure a five second delay from receipt of packet to display.

3.2.2 Avionics

Avionics programs will be written depending on the needs of other teams working on the rocket. They will likely be written in C or shell script, but this is dependant on the processor and sensors chosen by the ECE team. Avionics programs will be written for both the rocket and the payload, though responsibility for these will be split between the ECE team and our team.

3.2.3 Graphs for Payload

Telemetry data and experimental data for the payload will be graphed by our team. Outlier data will be identified using mathematical software. Fit lines for experimental data will be generated on request, and for all recorded telemetry. Detailed graphs will be generated for telemetry data.

3.3 Performance requirements

The parsing program should process incoming radio packets within 5 seconds of receiving them. We expect to receive a GPS packet about once per second.

The payload program will generate graphs from an unknown amount of data, which makes a performance requirement difficult.

3.4 Design constraints

The parsing and payload programs will run on a linux operating system, requiring relatively inexpensive hardware that should cost no more than $1,000.

The parsing, payload, and avionics programs will not require an active internet connection.

The avionics software will run on a processor to be determined by the ECE team.

3.5 Software system attributes

The parsing and payload programs will run on a linux operating system.

3.6 Other requirements