Groundstation: Requirements

Team #25
High-Altitude Rocketry Challenge
Natasha Anisimova
Terrance Lee
Albert Morgan

Abstract

The *Groundstation* software will collect telemetry from a rocket while it is in flight and graphically display the telemetry in real-time. Collecting the telemetry in real-time will reduce the chances of a failure-to-recover by allowing the ground team to track the rocket during its flight. The graphical display will make the telemetry instantly understandable without time-consuming analysis. This document will detail the specific requirements of the Groundstation software.

Contents

I. Introduction

A. Purpose

This document will specify the requirements for the Groundstation (GS) software. It is intended for use by the OSU High-Altitude Rocketry Team.

B. Scope

GS will gather telemetry from a rocket during flight. While the telemetry is being gathered, it will be logged and displayed graphically in real-time. This software will provide the following benefits to the users:

1) Real-time data will allow the rocket's location to be tracked during flight. Allowing this data to be accessed before recovery will aid in the recovery itself.
2) The graphical display will make the data easy to interpret.
3) Logging will allow the data to be analyzed at a later date.
4) Altitude data will give evidence of whether the objective of one hundred thousand feet was met.

C. Definitions

- **Accuracy:** The absence of errors in the telemetry GS receives, logs, and displays.
- **Binary data:** Telemetry that represents one of two states, for example, "stage 2 activated" and "stage 2 not activated."
- **Corruption:** The process by which data is altered or made unreadable.
- **Crash:** A software crash; the event in which a piece of software ceases operation unexpectedly.
- **Die:** A process dies when it ceases operation and is removed by the operating system. The difference between dying and crashing is that a crash may cause the program to become non-responsive, whereas dying causes the process to end.
- **GS:** Groundstation, the name of our software.
- **Graphical display:** Data that is displayed using a visualization.
- **Live:** Updated in real-time.
- **Non-volatile storage:** Storage that will not be erased when the system is powered down. For example, a hard drive or flash storage.
- **Page:** A web page that users of GS may connect to in order to view the telemetry.
- **Process:** A running program on a computer.
- **Raspberry Pi:** A small, inexpensive computing platform.
- **Real-time:** Each telemetry datum received from the rocket must be processed and displayed in under one second.
- **Redundant sensors:** Two or more sensors that provide the same type of data.
- **Reliability:** In the event of a software crash, the Groundstation software should automatically start and begin all normal functions in under five seconds.
- **Robustness:** In the event that GS receives data that is garbled or otherwise does adhere to the protocol, it must continue to receive and display data and not break the real-time requirement.
- **Storage:** A device where data is logged.
- **Telemetry:** Data received from the rocket while the rocket is in flight.
- **Telemetry packet:** The rocket will send a telemetry update once per second. Each one of these updates is a "telemetry packet."
- **Visualization:** Information or data, transformed into an visual context.

D. Overview

The rest of this document contains specific requirements about the functionality and constraints of GS. This includes the needs of the entire rocketry team, the physical constraints of the system, and the limitations of the launch site.

Overall description gives a high-level view of the functions of the software and describes any constraints. *Specific requirements* gives a detailed list of requirements that were proposed by the OSU High-Altitude Rocketry Team and describes specific requirements and constraints.

II. Overall description

A. Product perspective

GS will receive data from a serial interface, via hardware and a protocol provided by the avionics team. Due to the launch site not having any connection to the Internet or mobile service, GS will not interact with any outside systems. Additionally, GS will provide a software interface that allows users to view the telemetry, graphically, in real-time.

B. *Product functions*

GS provides two major functions while the rocket is in flight:

1) Telemetry logging to non-volatile storage.
2) Display of real-time graphical data.

C. *User characteristics*

The users of this software will be limited to engineering students and advisors who are part of the OSU High-Altitude Rocketry Team. These users will be expected to be familiar with the software and the rocket launch. Because the software team works closely with the rest of the rocketry team on regular basis, any training necessary will be conducted in-person.

D. *Constraints*

- Because of the remote nature of launch sites, the software must operate without an Internet connection.
- The software must be as reliable as is reasonably possible, and additional features must not compromise the reliability.
- Groundstation will receive telemetry via a protocol that is to-be-determined by the avionics team. GS will support most modern desktop web browsers, including:
  - Chrome version 54 or higher.
  - Edge version 14 or higher.
  - Firefox version 49 or higher.
  - Safari version 10 and or higher.

E. *Assumptions and dependencies*

Groundstation will run on a Raspberry Pi with no Internet connectivity.

F. *Apportioning of requirements*

Figure 1 shows a gantt chart outlining our schedule from now until the first test rocket launch. Although the final rocket launch is on 23 June, all core systems of GS should be finished by the test launch in mid-May.

III. Specific requirements

A. *External interface requirements*

1) User interfaces: Users will interact with GS by connecting via a web browser. The Raspberry Pi and other associated hardware, to be provided by the avionics team, will broadcast a wireless network that users will connect to and access GS through a web browser. From this page, users will be able to access the graphical display of the telemetry.

2) Hardware interfaces: GS will run on a Raspberry Pi broadcasting a wireless network. Users will connect to GS with their personal computers via the wireless network.

Data is received from a serial interface using a protocol that will be determined by the avionics team.

B. *System Features*

The stimulus for all specific requirements is receiving a telemetry packet. The specific responses to each telemetry packet are:

- GS will collect data from a serial interface using a yet-to-be-determined protocol designed by the High-Altitude Rocketry Team's avionics section.
- GS will log all telemetry in real-time to non-volatile storage so that it can be analyzed after the launch. The data from the serial port will be written to directly to non-volatile storage in order to limit the possibility of corruption.
- The graphical display shall be updated in real-time.
- Users will connect to GS and be able to view a visualzation of the telemetry in real-time.
- GS will support a live line graph visualization for altitude data.
- GS will have the ability to show the most recently recorded latitude and longitude in a numerical format.

C. *Performance requirements*

- When a new telemetry packet is received, GS will log the telemetry and update all visualizations in under one second.
- GS will support a minimum of 20 concurrent users.

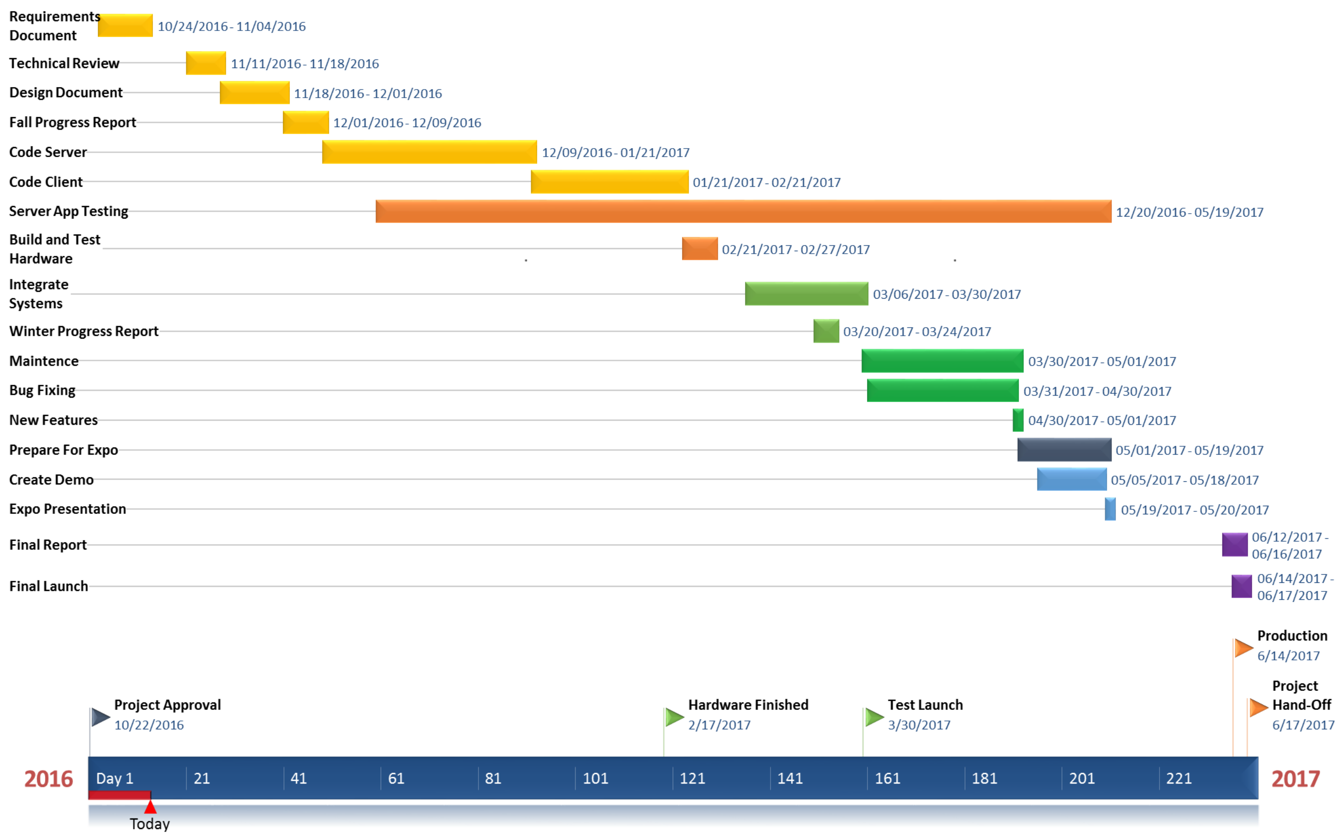

Fig. 1. A Gantt chart outlining the work to be done over the next nine months.

D. Design constraints

GS will run on the hardware chosen by the avionics team at the beginning of the project. At a minimum, this hardware will be a Raspberry Pi 3 model B, which has a 1.2 GHz quad-core 64-bit ARM Cortex-A53 processor with 1 GB of SDRAM memory.

E. Software system attributes

1) Reliability: In the event a GS process dies, a new process should begin and continue operation in under five seconds.

2) Robustness: If GS receives corrupted or otherwise incorrectly formatted telemetry, it should not crash. Additionally, the users should not be able to crash GS from the web interface.

3) Accuracy: GS should not introduce any errors in the telemetry it recieves. Data should be logged and displayed with 100% accuracy.

Index

Nancy Squires

Date

Natasha Anisimova

Date

Terrance Lee

Date

Albert Morgan

Date