

Technical Review And Implementation Plan For RockSat-X Payload - Hephaestus

Helena Bales, Amber Horvath, and Michael Humphrey

CS461 - Fall 2016

November 12, 2016

Abstract

The Oregon State University RockSat-X team shall be name Hephaestus. The possible methods for implementing our project requirements shall be outlined in this document. The mission requires that the payload, an autonomous robotic arm, perform a series of motions to locate predetermined targets. The hardware shall be capable of performing the motions to reach the targets. The software shall determine the targets and send the commands to the hardware to execute the motion. The combination of the hardware controlled by the software shall demonstrate Hephaestus's ability to construct small parts on orbit. This document will focus on the implementation of the software, but shall include necessary project context including hardware.

Approved By _____ Date _____

Approved By _____ Date _____

Approved By _____ Date _____

Approved By _____ Date _____

Contents

1	Introduction	3
1.1	Document Overview	3
1.2	Project Overview	3
1.2.1	Project Phases	3
2	Technologies	4
2.1	Tech 1	4
2.2	Tech 2	4
2.3	Tech 3	4
2.4	Tech 4	4
2.5	Tech 5	4
2.6	Tech 6	4
2.7	Telemetry	4
2.8	Video Handling	5
2.9	Data Visualization	5
3	Conclusion	5
4	Glossary	5
5	Appendix	5
5.1	Payload Modes of Operation	5
5.2	Model of Payload Hardware	5
5.3	Payload Wiring Diagram	5
5.4	References	5

1 Introduction

1.1 Document Overview

This is the Technical Review And Implementation Plan for the Hephaestus project. This document shall investigate possible methods of implementing our project software requirements. The nine general requirements investigated below were identified as project requirements in our Requirements document. This document will focus on the "how" of our requirements implementation.

1.2 Project Overview

The Hephaestus project is a Capstone Senior Design project for Oregon State University's 2016/2017 Senior Design class (CS461-CS463). The CS senior design project is one part of the overall Hephaestus project. In addition to the CS team, there is one team of Electrical Engineers and two teams of Mechanical Engineers working on this project through other senior design classes. The Hephaestus payload is a rocketry payload developed as part of the 2016/2017 RockSat-X program. The RockSat-X program is a year long program where groups of students develop rocketry payloads with the help of the Colorado Space Grant Consortium and Wallops Flight Facility. The term "rocketry payload" refers to an experiment inside a section of the rocket. Each section of the rocket is called a can, and is a standard space that we can fill with an experiment. The Hephaestus payload shall take up half a can and shall be mounted on a standard base plate provided by Wallops. We, as the Hephaestus team, will create the hardware and software for the payload, then integrate it into the rocket before launch.

1.2.1 Project Phases

The project shall include several phases. The first is the design phase. The design phase shall last all of Fall 2016 term at OSU. In the design phase, we shall design the robotics, electronics, materials, and software. The design phase shall include presentations to the RockSat-X program, where there will review our designs. Following the design phase will be the implementation phase. In the implementation phase we shall last through June 2017. This phase shall include testing of the payload. We will perform testing both at OSU and at Wallops. At OSU we will be testing the payload functionality. At Wallops, we will be testing the structural integrity of the payload, as well as its resistance to vibrations, heat, and cold. Following the implementation phase will be the integration phase. This phase will occur at Wallops in July. This is the point at which our base plate will be integrated into the rocket as a whole, along with the other participating teams. The final phase will be launch. Launch will occur in Summer of 2017. The rocket shall be launched from Wallops Flight Facility. During the flight we shall send telemetry to the ground station at Wallops. The payload shall perform the experiment once it reaches appogee. The payload will hopefully be recovered post-flight.

2 Technologies

2.1 Tech 1

2.2 Tech 2

2.3 Tech 3

2.4 Tech 4

2.5 Tech 5

2.6 Tech 6

2.7 Telemetry

The three options being considered for transmitting telemetry are 1. a custom-built solution for our own needs, 2. Open MCT developed by NASA for space-specific missions, and 3. PSAS Packet Serializer developed by Portland State Aerospace Society (PSAS). The goal for this technology is to let the developers quickly and easily relay data to the ground station.

The criteria that these technologies will be evaluated on is:

- Ease of use. The chosen solution should let the developers focus on writing code and not encoding data for telemetry transmission. Ideally, sending data through one of the telemetry ports should be no more than one line of code.
- Reliability. The chosen solution should be able to relay 100% of transmitted data to the ground station without corrupting or losing any of it.
- Documentation. The chosen solution should be well documented. The developers should be able to quickly and easily locate supporting documentation for using the technology.
- Compatibility. The chosen solution should be compatible with the architecture of the payload.

The custom-built solution is the least appealing. It would require the most amount of work to develop and maintain by the developers. The advantage of a custom-built solution is that it can be tailored to the requirements of our system, making it extremely to use. However, the benefit is offset by the huge amount of work upfront it would require to develop and test the solution. Since the developers would be coding up this solution themselves, it would require a lot of testing to ensure a reliable solution. The hand-written test cases cannot guarantee the reliability of the solution, especially given the relative inexperience of the developers with writing code for this platform. Therefore one can expect to have relatively unreliable code and encounter lots of bugs. Compatibility would not be a problem with this solution because the code would be custom-made for the hardware. However, documentation would non-existent because the developers would be writing the code themselves. The only documentation that would be relevant would be from other projects that have written telemetry code for spacecraft. However, most of that documentation would be internal to the organizations building the spacecraft, most likely wouldn't be helpful.

Open MCT

PSAS Packet Serializer is a student aerospace engineering project developed by PSAS at Portland State University (PSU). The project seeks to create a standard way to encode data for telemetry transmission between various components and the ground station. This solution would be very easy to use because of its simple interface. Only one line is required to both encode and decode data. This solution is also extremely reliable since it has been used in several flights by the PSU team. The solution is also well-documented. There is an entire website dedicated to documenting the simple API. However, the major problem with this solution is compatibility. The solution is implemented in Python, whereas the code for the payload is restricted to C. It is not feasible to run the Python implementation on the microcontroller in C, but it may be possible to port the code to C. This would require a lot of unpleasant work on the developers' part.

The best option for now appears to be adopting Open MCT for the purposes of this mission.

2.8 Video Handling

2.9 Data Visualization

3 Conclusion

4 Glossary

5 Appendix

5.1 Payload Modes of Operation

5.2 Model of Payload Hardware

5.3 Payload Wiring Diagram

5.4 References