Preliminary Desing Document For RockSat-X Payload - Hephaestus

Helena Bales, Amber Horvath, and Michael Humphrey

CS461 - Fall 2016

November 17, 2016

Abstract

The Oregon State University (OSU) RockSat-X team shall be named Hephaestus. The preliminary design of our project 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

1.1 Document Overview

1.1.1 Helena Bales

1. Target Generation
2. Arm Movement
3. Arm Position Tracking

1.1.2 Amber Horvath

1. Emergency Payload Expulsion
2. Program Modes of Operation
3. Target Success Sensors

1.1.3 Michael Humphrey

1. Telemetry
2. Video Camera
3. Data Visualization and Processing

2 Technologies

2.1 Target Generation

2.1.1 Requirement Overview

The software shall generate points to be used in testing the Hephaestus arm. The points will constitute the total test of the arm, and should therefore include points representative of standard and edge cases. These points shall be used as targets for the arm body.

2.1.2 Solution Design

2.2 Arm Movement

2.2.1 Requirement Overview

The software shall control the movement of the arm body assembly. The position of the tip of the arm shall be tracked in the coordinate notation described in section 2.2 above. The software shall rotate the arm body assembly in a full 360 degrees. The software shall additionally control the movement the height of the arm body assembly. The arm should descend and touch the baseplate of the payload at any rotation.

2.2.2 Solution Design

2.3 Arm Position Tracking

2.3.1 Requirement Overview

The position of the arm shall be tracked using the same coordinate system described in the Target Generation requirement. The position of the arm shall be calculated using the known start position and the rotation of the motors.

2.3.2 Solution Design

2.4 Emergency Payload Expulsion

2.4.1 Requirement Overview

The software shall eject the arm upon system failure. System failure in this case is defined as the arm becoming lodged or stuck in a state where it is unable to retract. The software will enter Safety mode (defined in section 2.5.2) and attempt to retract the arm. If it is unable to complete this step, the system will continue attempting to eject the arm until ejection is completed

2.4.2 Solution Design

2.5 Program Modes of Operation

2.5.1 Requirement Overview

The software shall have the Modes of Operation necessary to insure the mission success. The software shall first deploy the payload, then the arm. Next the software shall activate the camera and perform a video sweep. The software shall then perform the science experiment. If the experiment fails, it shall return to observation mode. If observation mode fails, it shall return to idle. Once the experiment time has been exhausted, the payload shall shut down. If it shuts down correctly, everything will poweroff. If not, the payload shall attempt to retract again, or expel the payload from the rocket.

2.5.2 Solution Design

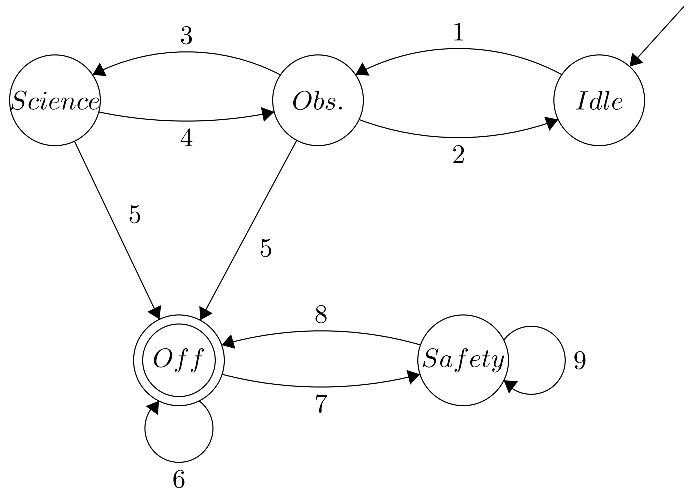

Diagram of software states of operation and transition between states [2].

Transitions between states occur as numbered:

1. **Appogee is reached.** The software shall activate when the power line goes to high at 28V. Observation mode shall be triggered when the OBC turns on.
2. **Error: Return to Idle.** If an error is encountered in entering Observation mode, the software shall fallback to Idle mode and retry. An error may occur if the payload fails to deploy correctly or if the camera fails to turn on.
3. **Payload Assembly and Camera have been deployed.** The software shall enter science mode once the payload assembly and arm have deployed and the camera has performed an observation sweep.
4. **Error: Return to Observation** The software shall return to observation mode if any error occurs in Science mode. An error may occur in Science mode if the arm fails to operate correctly and must return to default position. An error may also occur if the camera stops working.
5. **Timer switches to end appogee period.** Once the time period for observation has ended, the timer line will go to low and trigger to Shutdown state. This state can be reached from either Observation or Science mode.
6. **Accept: Shutdown correctly** If Shutdown occurs correctly, the arm should be closed, the Arm Assembly Body should be retracted, and the OBC should be powered off.
7. **Error: Shutdown not completed successfully.** If an error occurs in the shutdown sequence, the software shall enter Safety mode.
8. **Payload is Shutdown correctly.** If the payload is Shutdown through Safety mode, shutdown can be completed. In Safety mode the payload was either shut down correctly, retracted fully into the can with the arm open, or the arm was expelled safely from the rocket.
9. **Error: Shutdown not completed successfully.** If an error occurs in the shutdown sequence, the software shall enter Safety mode.

10. **Payload is Shutdown correctly.** If the payload is Shutdown through Safety mode, shutdown can be completed. In Safety mode the payload was either shut down correctly, retracted fully into the can with the arm open, or the arm was expelled safely from the rocket.
11. **Error: Payload is still deployed.** The software shall remain in Safety mode until the payload is either retracted correctly, retracted fully with the arm in the open position, or ejected safely from the rocket. Safety mode shall first try to correctly retract the arm, then retract with the arm open, then repeat attempting ejection until the payload is ejected.

2.6 Target Success Sensors

2.6.1 Requirement Overview

The software shall know whether or not the arm succeeded in touching the targets generated, as described in section 2.1. The sensors shall report back whether or not contact was made. This data can be used in post-mortem analysis to determine whether certain targets were faulty or whether the range of motion on the arm was faulty.

2.6.2 Solution Design

2.7 Telemetry

2.7.1 Requirement Overview

The software shall report via telemetry all sensor data.

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 software and hardware of the payload.

2.7.2 Solution Design

2.8 Video Handling

2.8.1 Requirement Overview

The software shall be responsible for controlling the camera output.

The criteria that these technologies will be evaluated on is:

- **Reliability.** The solution should guarantee that video footage is permanently recorded.
- **Ease of use.** The solution should be easy to implement and use.

2.8.2 Solution Design

2.9 Data Visualization and Processing

2.9.1 Requirement Overview

After the mission completes, the software shall provide visualizations for the collected data. The software shall be able to show whether the mission success criteria have been met or not. If the mission success criteria have not been met, the software shall show how and why they have not been met.

The criteria that these technologies will be evaluated on is:

- **Cross-platform compatibility.** The chosen solution should be able to run across any of the major computing platforms.
- **Range and variety of visualization methods.** The chosen solution should have a large variety of different visualization methods.
- **Documentation.** The chosen solution should be well documented. The developers should be able to quickly and easily locate supporting documentation for using the technology.
- **Developer proficiency.** The majority of developers should be able to comfortably develop the visualizations without needing to learn any new technologies.

2.9.2 Solution Design

3 Conclusion

4 Glossary

5 Appendix

5.1 Mission Patch

Figure 2: Mission Logo [1]

5.2 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.

5.2.1 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.

5.3 Software State Diagram

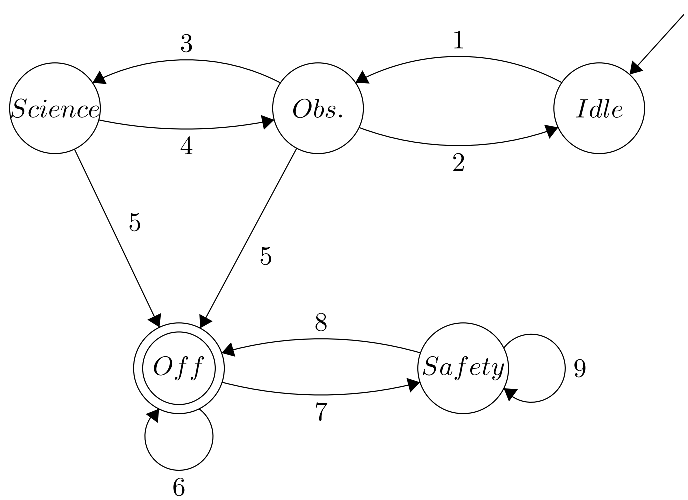

Figure 3 – Diagram of software states of operation and transition between states [2].

Transitions between states occur as numbered:

1. **Appogee is reached.** The software shall activate when the power line goes to high at 28V. Observation mode shall be triggered when the OBC turns on.

2. **Error: Return to Idle.** If an error is encountered in entering Observation mode, the software shall fallback to Idle mode and retry. An error may occur if the payload fails to deploy correctly or if the camera fails to turn on.

3. **Payload Assembly and Camera have been deployed.** The software shall enter science mode once the payload assembly and arm have deployed and the camera has performed an observation sweep.

4. **Error: Return to Observation** The software shall return to observation mode if any error occurs in Science mode. An error may occur in Science mode if the arm fails to operate correctly and must return to default position. An error may also occur if the camera stops working.

5. **Timer switches to end appogee period.** Once the time period for observation has ended, the timer line will go to low and trigger to Shutdown state. This state can be reached from either Observation or Science mode.

6. **Accept: Shutdown correctly** If Shutdown occurs correctly, the arm should be closed, the Arm Assembly Body should be retracted, and the OBC should be powered off.

7. **Error: Shutdown not completed successfully.** If an error occurs in the shutdown sequence, the software shall enter Safety mode.

8. **Payload is Shutdown correctly.** If the payload is Shutdown through Safety mode, shutdown can be completed. In Safety mode the payload was either shut down correctly, retracted fully into the can with the arm open, or the arm was expelled safely from the rocket.

9. **Error: Payload is still deployed.** The software shall remain in Safety mode until the payload is either retracted correctly, retracted fully with the arm in the open position, or ejected safely from the rocket. Safety mode shall first try to correctly retract the arm, then retract with the arm open, then repeat attempting ejection until the payload is ejected.

5.4 References

[1] Oregon State University RockSat-X Team, "Hephaestus Mission Patch," 2016. [Online]. Accessed: June 14, 2016.

[2] H. Bales and M. Humphrey, "Diagram of Software Modes of Operation," 2016. [Online]. Available: Hephaestus Requirements Document.