# HASP Student Payload Application for 2021

Payload Title:
COMPASS (Calculating Orientation and Measuring Pointing Angle for Scientific Systems)

Institution: Louisiana State University

Payload Class (Enter SMALL, or LARGE): SMALL

Submit Date: January 8, 2021

Project Abstract:

Many types of detectors such as light detectors, telescopes, and cameras have a limited field of view that determines what the sensor sees. In the context of high-altitude ballooning, a major challenge in using these types of sensors is accounting for the rotation of the payload and knowing the sensor orientation. The main goal of COMPASS is to determine the orientation of a balloon payload in space. To accomplish this goal, COMPASS shall measure the azimuth and elevation of the payload’s pointing vector during flight and verify these measurements using the Sun as a reference. Once the azimuth and elevation have been found, the right ascension and declination shall be found. COMPASS shall be a small class payload that sits on HASP mounting plate. It will also require a boom that holds a magnetic compass away from the HASP gondola to take magnetic field readings. An array of cameras will be used to view the Sun and verify magnetometer measurements. Jeanne Garriz is the student team leader, Aaron Ryan is the faculty advisor, and the rest of the team will consist of undergraduate students.

Team Name: COMPASS Development Team

Team or Project Website:
N/A

| | | |
|---|---|---|
| | Student Leader Contact Information: | Faculty Advisor Contact Information: |
| Name: | Jeanne Garriz | Aaron Ryan |
| Department: | Physics and Astronomy | Physics and Astronomy |
| Mailing Address: | 1741 Brightside Drive | 202 Nicholson Hall |
| City, State, Zip code: | Baton Rouge, LA 70820 | Baton Rouge, LA 70803 |
| e-mail: | jgarri9@lsu.edu | aryan21@lsu.edu |
| Office Telephone: | N/A | N/A |
| Mobile Telephone: | (251) 391-2596 | (808) 783-9096 |

# Flight Hazard Certification Checklist

NASA has identified several classes of material as hazardous to personnel and/or flight systems. This checklist identifies these documented risks. Applying flight groups are required to acknowledge if the payload will include any of the hazards included on the list below. Simply place an (x) in the appropriate field for each hazard classification. **Note:** Certain classifications are explicitly banned from HASP (grey filled items on table below) and the remaining hazards will require additional paperwork and certifications. If you intend to include one of the hazards, you must include detailed documentation in section 3.8 of the application as required by the HASP Call for Payloads.

This certification must be signed by both the team faculty advisor and the student team lead and included in your application immediately following the cover sheet form.

## Hazardous Materials List

| Classification | Included on Payload | Not Included on Payload |
|---|---|---|
| RF transmitters | | X |
| High Voltage | | X |
| Pyrotechnics | | X |
| Lasers | | X |
| Intentionally Dropped Components | | X |
| Liquid Chemicals | | X |
| Cryogenic Materials | | X |
| Radioactive Material | | X |
| Pressure Vessels | | X |
| Magnets | | X |
| UV Light | | X |
| Biological Samples | | X |
| Li-ion Batteries | | X |
| High intensity light source | | X |

Student Team Leader Signature: ______________________________

Faculty Advisor Signature: __________________________________________

# Table of Contents

# 1. Payload Description

This section describes the payload goals, background, payload components, and the planned designs for mechanical, electrical, and thermal systems. Section 1.1 details the scientific and technical background that influences and shapes the planned payload design. Section 1.2 describes the highest-level system design plans and the main principle of operation of the entire payload. The diagrams and designs contained in sections 1.3-1.6 are very high level and will provide the foundations for prototypes and more detailed electrical, mechanical, and thermal designs.

## 1.1 Payload Scientific / Technical Background

Section 1.1 and its subsections contain a summary of the overall payload mission, objectives, and background regarding the payload design. Section 1.1.1 is a brief statement of the payload mission and reasons for this payload's flight. Section 1.1.2 details the scientific and technical background that informs the objectives, requirements, and accuracy of the payload and its proposed measurements. Section 1.1.3 lists the mission objectives and the requirements that go with each of these objectives. These objectives and requirements inform the design of the payload and help to fulfill COMPASS' main goal.

### 1.1.1 Mission Statement

The goal of COMPASS is to determine the orientation of a payload in space by measuring the payload's magnetic heading, acceleration, and verifying these measurements with an array of cameras using the Sun as a reference. These measurements will be taken at float altitude of around 120k feet. Determining the direction in which a payload points is important for making measurements using field of view dependent sensors when the sensor placement is not fixed, such as on a future balloon payload.

### 1.1.2 Mission Background and Justification

**1.1.2.1 Motivation**

Determining the direction in which a payload points is useful for using sensors with limited fields of view such as optical/light sensors, telescopes, and cameras. When the sensors are fixed in place this is not an issue, but the pointing direction of the sensors becomes important when the sensor is not fixed, such on a balloon payload. Finding an orientation method for balloon payloads would greatly simplify the use of field of view dependent sensors on future payloads and for applications such as UV measurements and locating the Sun during an eclipse.

**1.1.2.2 Previous Work**

The field of view and sensor orientation issue arose in a previous LaACES sounding balloon payload called Student Payload for UV Detection (SPUD), designed and built by undergraduates during the spring semester of 2020. SPUD used UV sensors to measure UV and ozone in the atmosphere. The UV sensor response was dependent on the angle at which the radiation from the Sun hit the sensor. A magnetometer was placed inside the payload with the intention of using the magnetic field readings to determine the payload's orientation and solar incidence angle at any given time and therefore correct the UV irradiance readings. However, this did not work as planned due to what the team believed to be unforeseen electromagnetic interference on the magnetometer from the payload electronics.

**SPUD Interior**

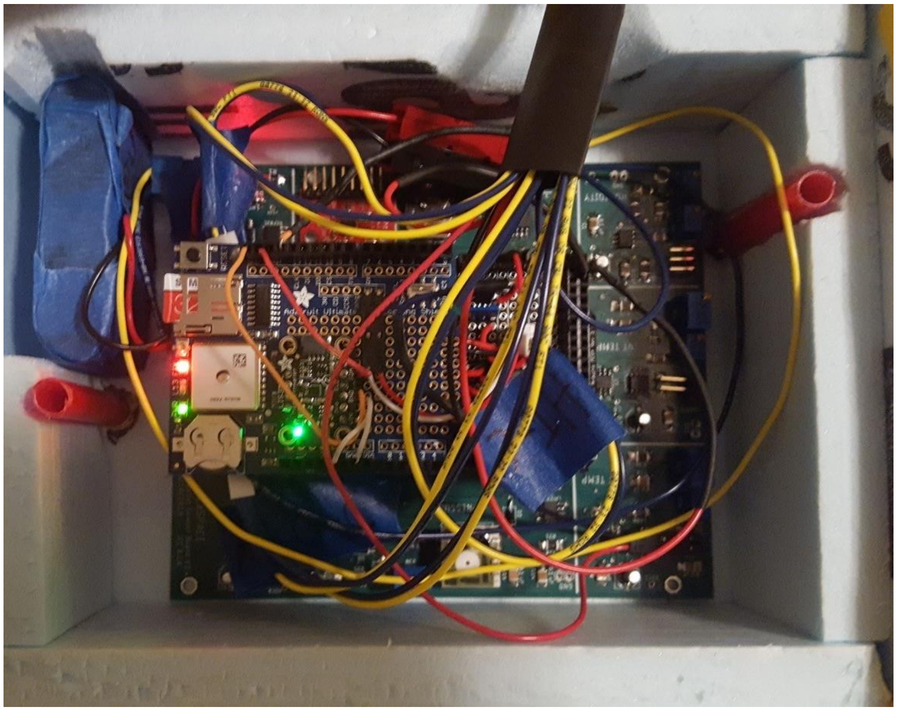


*Figure 1: Previous payload's interior. It's difficult to tell from the picture, but the magnetometer sat directly on top of the electronics stack of this payload. This is the theorized reason for the magnetometer malfunction on SPUD.*

COMPASS' planned use of an orientation system (magnetometer and accelerometer) placed on a boom away from the payload, as well as one inside the payload, will be able to provide insight into the functionality of a magnetometer when used for scientific ballooning applications. Specifically, COMPASS will provide information on the use of a magnetometer to determine the azimuth and elevation at which a payload points. The "front" of the payload will be defined to be the direction in which one of the cameras is pointing. To verify that these magnetometer readings are correct, an array of cameras placed around the payload will take snapshots during flight to orient the payload with respect to the Sun.

### 1.1.2.3 Science Background

The azimuth is defined as the angle between the magnetic north vector and the observed object's vector on the horizontal plane. In the case of this payload, it would be the angle between north and the direction in which the payload points. Azimuth can be measured with a magnetic compass (magnetometer) by relating the measured magnetic field readings to the Earth's magnetic field to find a relative angle. The elevation (or altitude) is the angle between an object (or in this case the payload pointing vector) and the observer's local horizon. Elevation can be measured with a tilt sensor/accelerometer, using the Earth's gravity as a reference. These two measurements will be able to describe where the payload is pointing in the Earth's reference frame. Right ascension and declination can also then be calculated using the orientation of the payload, latitude, longitude, and time of observation.

To verify whether the readings from the magnetometer are correct, an array of cameras placed on the outside of the payload will be used to orient the payload with respect to the Sun and the Earth's reference frame. These cameras will cover the total range of the Sun's motion relative to the payload from an early estimate of float time to sunset. This results in a total change of 190 degrees in azimuth and 50 degrees in elevation. Snapshots from these cameras and their timestamps will then be sorted, and pictures not containing the Sun will be disregarded. The rest will be run through image processing software to pinpoint the location of the Sun relative to the payload and relative to the Earth at different times and will be compared to the magnetometer readings at these times.

**1.1.2.4 Technical Background**

Because a future goal of such an orientation system would be to ascertain when detectors are pointed at the Sun, we would like to determine the azimuth and elevation of the payload's pointing vector within half a degree of the Sun's angular size (0.5 degrees). Therefore, our desired accuracy is 0.25 degrees. This is required because COMPASS' verification system uses the Sun as a reference. These magnetometer measurements and camera snapshots will be taken once HASP reaches float altitude and is stable. Ascent and descent will not be considered because the motion of the payload during these time periods is too erratic to take measurements within the specified accuracy. To take measurements at roughly every 1/8th (0.125) degree interval (half the required accuracy, for thoroughness and redundancy), the payload will need to sample data twice every second at float. This sampling rate was determined by estimating the rate of rotation of HASP from previous flight videos to be 0.23 degrees/second at float.

$$0.23\,\frac{degrees}{second} * \frac{1}{0.125\;degrees} = \frac{1.84}{second} \approx 2/second$$

Because of the importance of cameras in this payload, a Raspberry Pi is being considered as the microprocessor because of its many compatible camera options and full camera library. Raspberry Pis have also been used in previous HASP payloads, so they have been shown to survive the conditions present during flight.

## 1.1.3 Mission Objectives

COMPASS' mission objectives are listed below and underlined. The requirements are listed under the objectives.

1 <u>Measure azimuth and elevation of the payload's pointing vector</u>

1.1 Determine payload angle relative to north to within 1/4 degree using magnetic compass throughout flight

1.1.1 Sampling rate of 2x/sec during float (no data taken during ascent and descent)

1.1.2 Minimize electronic and magnetic interference with magnetic compass- determines placement of magnetometer and boom on payload

1.1.2.1 Length of boom so that interference is less than magnetometer error

1.1.3 Use Earth's magnetic field as reference- will not drift like a gyroscope

1.1.4 Error in x-y components needs to be ±100.6nT based on assumed horizontal field strength 23055.8nT obtained from an altitude 36km from WMM magnetic field model.

1.1.5 Shielded cables if communication between magnetometer and microprocessor is digital

1.1.6 Communicate over a few meters distance with microprocessor

1.2 Measure angle of payload relative to horizontal using tilt sensor

1.3 Correlate measurements with timestamp

1.3.1 Absolute timing requirement based on the movement of the Sun. For a desired accuracy of Sun position to a 0.25 degrees, absolute timestamp must be accurate within 100 seconds.

$$\frac{360\ degrees}{24\ hrs} * \frac{1hr}{3600\ sec} = 0.0025\ degrees/sec$$

1.3.1.1 Acquire timestamp data from GPS.

1.3.2 Need to know relative timing between images and data to within 0.5 seconds- use microprocessor timing.

2 Verify azimuth and elevation measurements using an array of cameras with the Sun as a reference

2.1 Multiple cameras to cover change of 190 degrees in azimuth and 50 degrees in elevation

2.2 Pixel resolution <1/4 degree based on desired accuracy

2.3 SD card to store telemetry data and approximately 288000 images (2 cameras*60min*20hrs = 2400)

2.4 Solar filters to protect camera sensors

2.5 Camera compatible microprocessor and library

3 Determine the RA and Dec of the payload's pointing vector

3.1 Correlate measurements with latitude, longitude, and altitude

4 Adhere to HASP requirements

4.1 Payload less than 3kg

4.2 Approval to place components outside designated area (boom for magnetometer)

4.3 Located on seat on opposite side of launch vehicle

5 Interface with HASP power system before and during flight

5.1 Handle voltage range of 29-33V at pins A, B, C, and D of the EDAC

5.2 Max current draw 0.5A for whole payload

6 Interface with HASP control/communication system before and during flight

6.1 Have specified data format for downlink data during flight

6.2 Use 1200 baud rate for serial communication

6.3 Have written commands to control payload during flight

7 Process sensor and camera data using a microprocessor

7.1 Voltage stepped down to power microprocessor

7.2 Use Raspberry Pi for camera compatibility

8 Store data on an SD for analysis after flight

8.1 Storage requirement will depend on number of cameras, frequency of pictures, and output format of camera- research in progress

8.2 Data record format to prevent corruption in stored and downlinked data

8.2.1 Downlink repetitive data
8.2.2 Include checksum
8.2.3 Close files after designated number of data points

9 <u>Withstand atmospheric conditions during flight</u>
9.1 Pass thermal/vacuum test
9.2 Insulate payload
9.3 Measure internal and external temperature during flight
9.4 Withstand 10G vertical and 5G horizontal shock

## 1.2 Payload Systems and Principle of Operation

Using a system of orientation sensors, COMPASS is designed to measure its elevation and azimuthal position in space. To verify that these measurements are accurate, located on the payload will be an array of cameras to take snapshots of the sun in real time and downlink them to HASP. These snapshots will be compared to the orientation data of which include magnetic field measurements that can be correlated with tracking the position of the sun.

COMPASS will consist mainly of a control system, camera-array system, two orientation systems, power regulation system, and environmental monitoring system. There will be one onboard orientation system and one mounted on a boom away from the payload. Each of these major systems will later be defined in the next section (1.3).

**System Design Diagram**

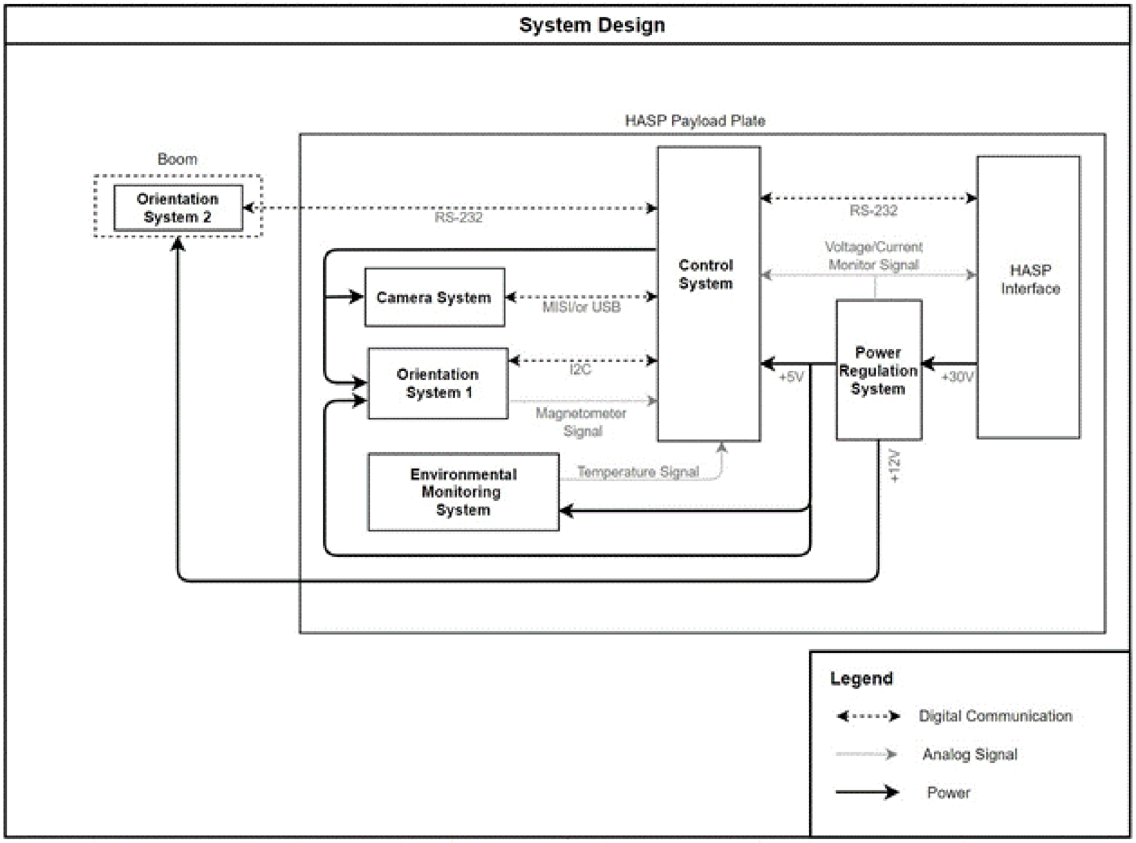


*Figure 2: COMPASS high level system design diagram*

## 1.3 Major System Components

### 1.3.1 Control System

The COMPASS control system will contain three major components: a microcontroller for receiving/transmitting data to HASP, a GPS system for recording positional data, and an SD card for data storage.

We will use a Raspberry Pi 4 Model B microcontroller which consists of a 40-pin GPIO pin header used to interface with each of the major system components, including the GPS system, and has an onboard Micro SD card slot for storing data. Communication with HASP will also require using multiple RS-232 level shifters to convert TTL level signals to RS-232 interface. One will be used between the microcontroller and HASP serial connector, and another will be used to communicate with signals coming from the electronics on the boom. In addition, we will need to add an analog to digital converter to the control system because the Raspberry Pi 4 is a “digital-only” microcontroller and does not have its own ADC chip. One such ADC we could use is the MCP3008, which has 10-bit resolution and 8-input channels.

**Control System Diagram**

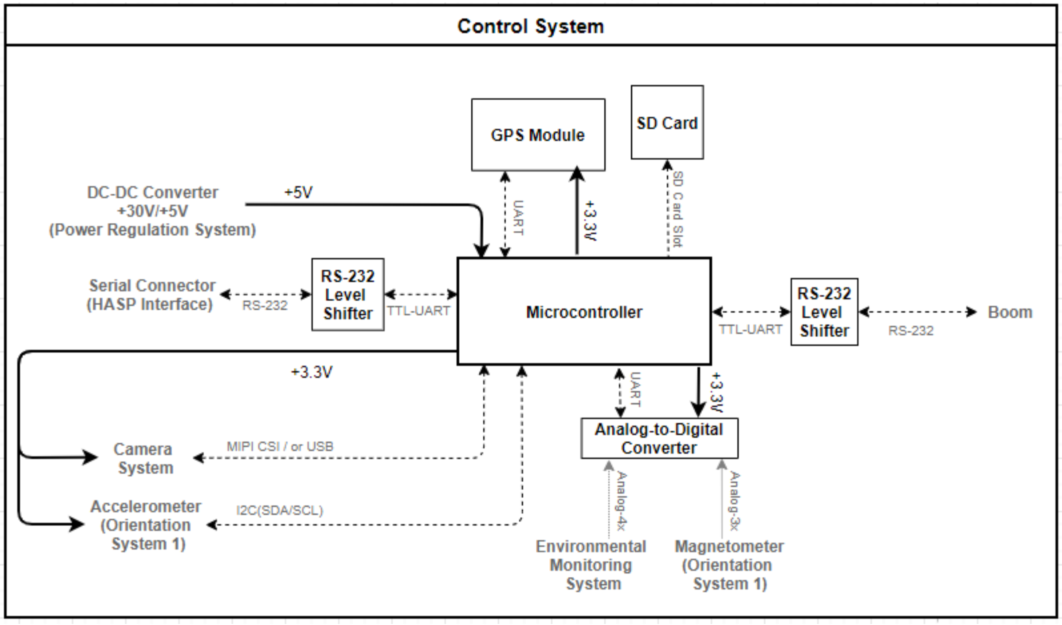


*Figure 3: High level diagram detailing the components of the control system.*

### 1.3.2 Orientation System 1

**Orientation System 1 Diagram**

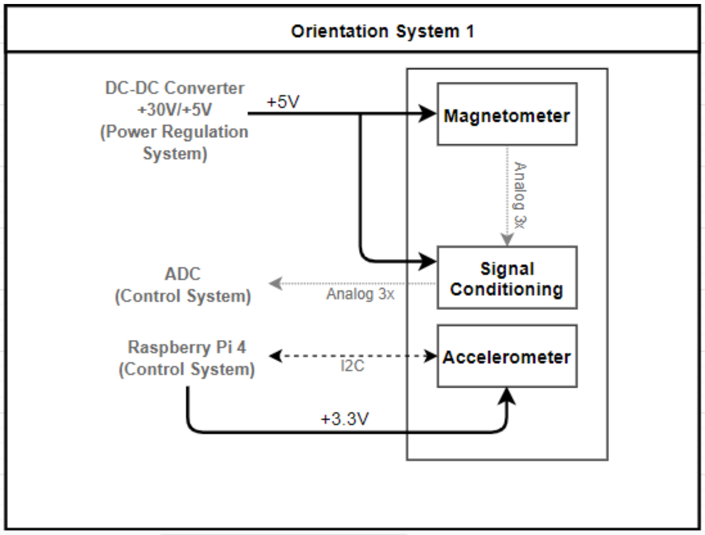


*Figure 4: High level diagram of Orientation System 1, which will be located on the payload.*

Measures azimuth and elevation of the payload. This orientation module will contain a 3-axis magnetometer that will be used to measure a magnetic field vector at each timestamp. Given that we know what Earth's magnetic field is, we can compare this to our magnetic field readings to determine the azimuthal direction of the payload in space. Based on the pointing accuracy requirement, we were able to determine that the magnetometer shall have an accuracy of about +/-100nT. This system will also contain a 3-axis accelerometer which will be used to measure the elevation of our payload. The accelerometer will allow us to detect which direction our payload is pointing down towards Earth which can be used to determine the angle of elevation.

### 1.3.3 Environmental Monitoring System

**Environmental Monitoring System**

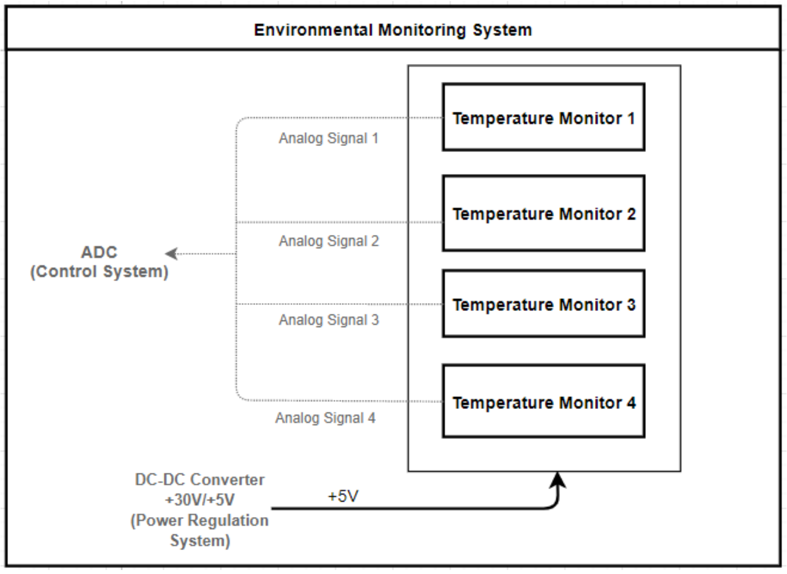


*Figure 5: High level diagram of the environmental monitoring system. This system includes multiple temperature sensors and this diagram shows the sensors that will go in the payload box on the HASP payload plate*

This system will include internal temperature sensors to help monitor the health of the onboard payload components. Four temperature sensors will be placed on the payload box on the HASP gondola and two will be placed on the payload on the boom with Orientation System 2.

### 1.3.4 Power Regulation System

The power regulation system will consist of two onboard DC-DC converters, one +5V and one +12V. It will also include two voltage and two current monitoring circuits to regulate the output voltage and current of each converter. These will distribute the 30V supplied by HASP to the rest of the payload.

**Power Regulation System Diagram**

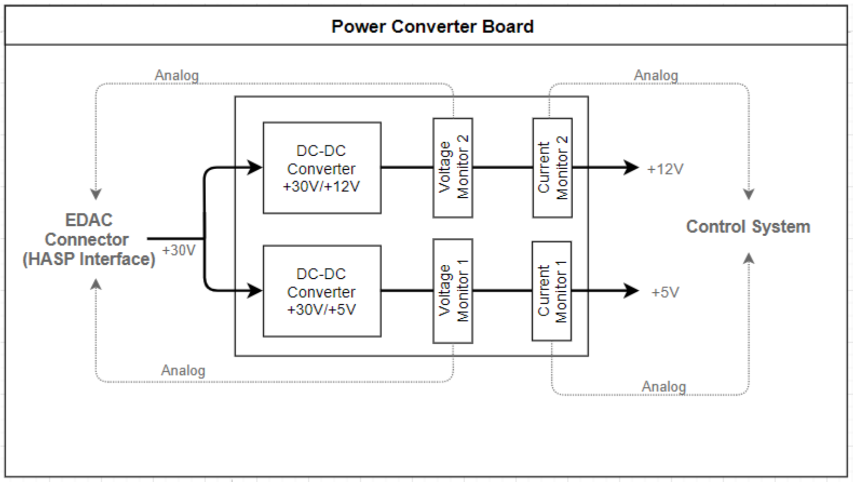


*Figure 6: Power converter high level diagram. This shows how the 30V supplied by HASP will be distributed to the payload.*

### 1.3.5 Camera System

An array of two cameras will take simultaneous snapshots of the Sun to verify the measurements taken by each of the payloads orientation systems.

These cameras will cover an angular area of at least 190 degrees in azimuth and 50 degrees in elevation, which is the range of motion of the Sun relative to the payload during float and have a max angular error of <1/4 degree. This camera array will take snapshots at a rate of around 1 picture per minute.

**Camera Array System Diagram**

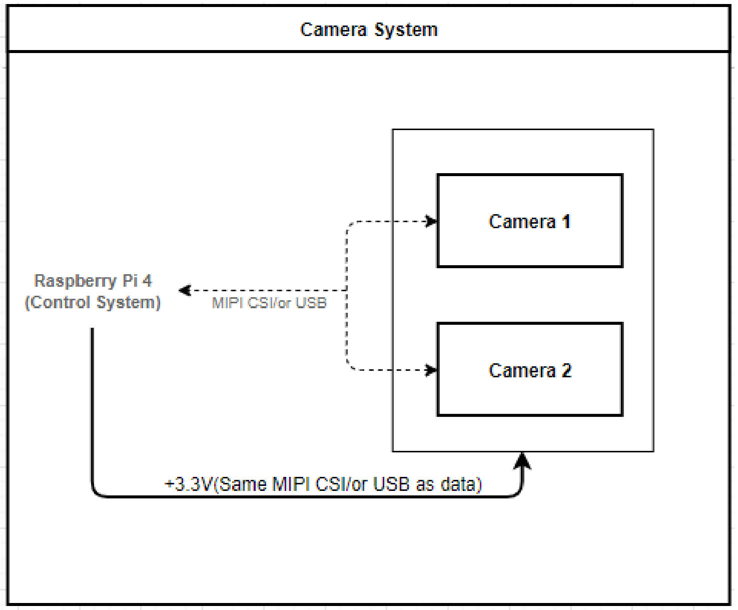

Camera System
Camera 1
Raspberry Pi 4 (Control System)
MIPI CSI/or USB
Camera 2
+3.3V(Same MIPI CSI/or USB as data)

*Figure 7: This diagram shows the basic components required for the camera array system*

Images will be stored on the SD card and processed post-flight to determine the location of the payload in the Earth’s reference frame, and then these results will be compared to magnetometer readings at the same timestamp to verify magnetometer measurements. We also plan on downlinking images periodically, which will require onboard image processing after the pictures are taken.

### 1.3.6 Orientation System 2

OS2 will be mounted on a fiberglass boom away from our main payload (and the HASP gondola) to avoid possible electronic interference. This orientation system will also include its own environmental monitoring system. For this system to communicate with the Raspberry Pi on the payload, it will require another RS-232 level shifter to interface with the designated one on the control system.

**Orientation System 2 Diagram**

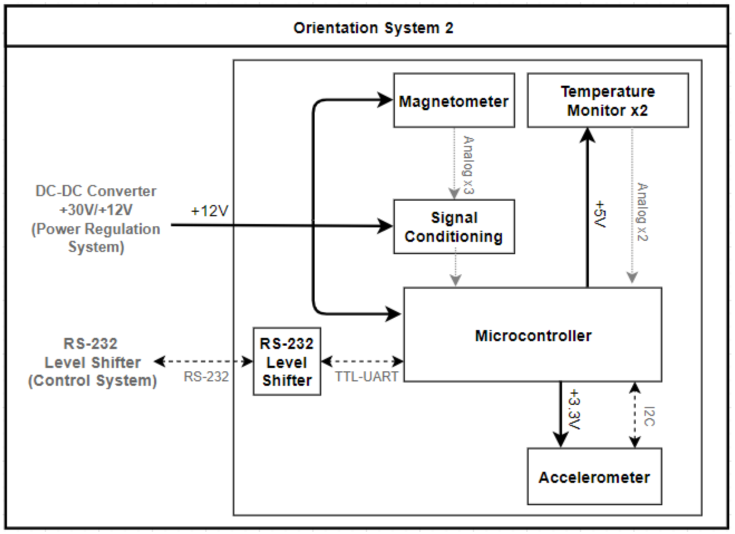


*Figure 8: Orientation System 2 high level diagram. This system will be on the boom away from the rest of HASP.*

Mounting the module far from the microcontroller can cause potential communication issues, because sensors cannot be directly read over such long distances. RS232 communication is used to communicate such long distances. However, because the accelerometer only communicates via the I2C bus, there would have to be some way of converting from I2C to TTL-UART, then to RS232 before it can communicate with the RPi4. Also, like the first orientation system, there would have to be a way to convert the magnetometer and environmental monitor analog signals to digital signals before it can communicate with the RPi4. Our solution to solve both these problems would be to add a second microcontroller and an RS232 level shifter to the boom that can read the detector with an ADC, and then convert the digital signal to RS232 before it can be transmitted to the control system on the HASP payload plate.

## 1.4 Mechanical and Structural Design

The payload will consist of 2 major structural components, the main payload mounted directly to HASP payload plate and a remote magnetometer boom consisting a magnetometer mounted at the end of a 12 ft long fiberglass boom.

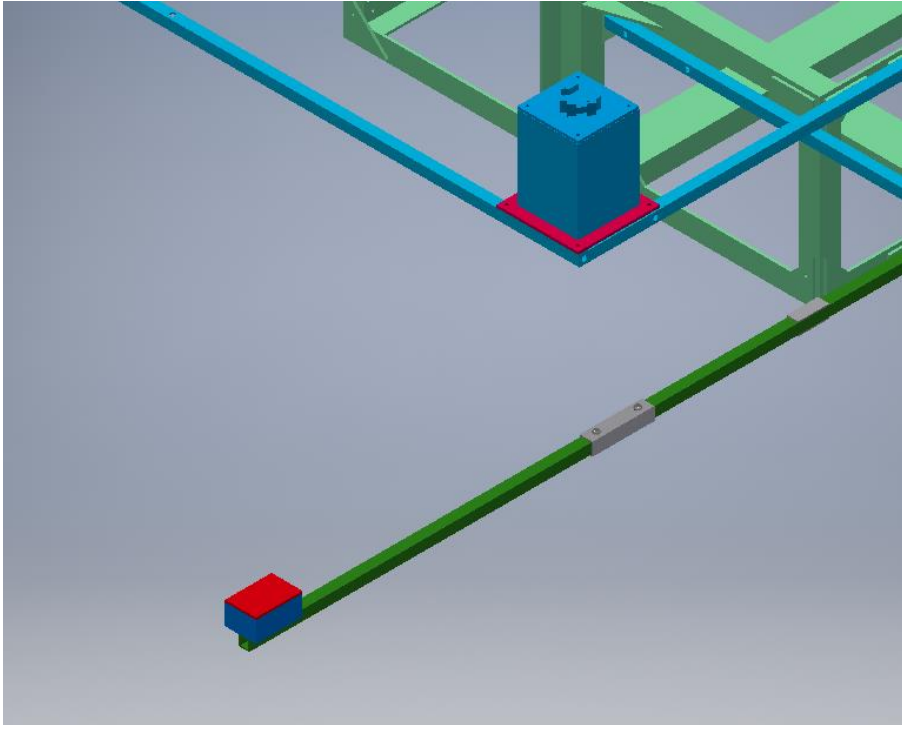


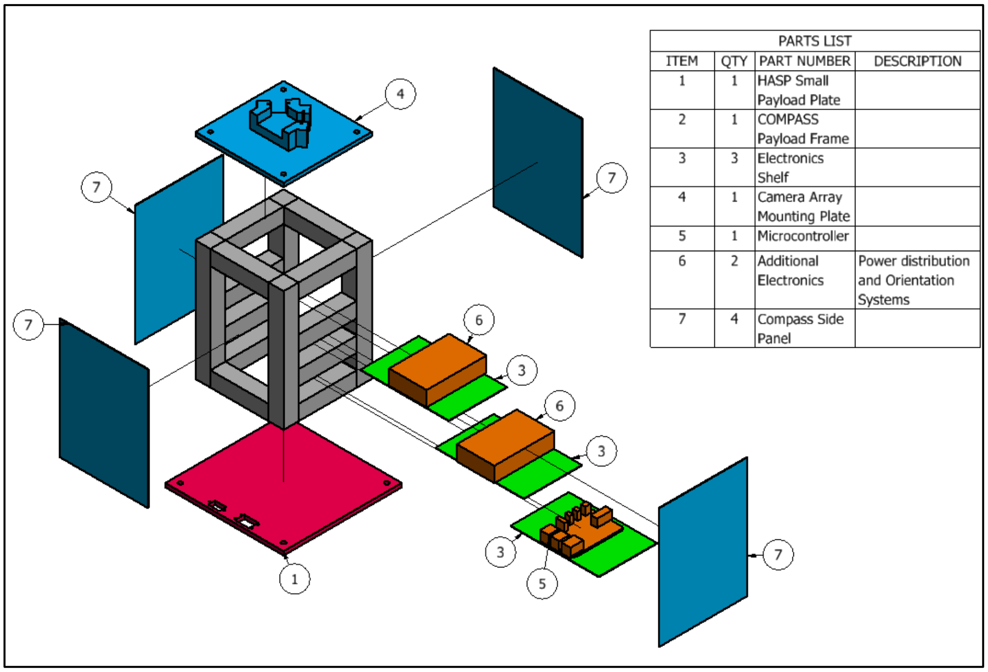
*Figure 9: An example of the arrange of the COMPASS payload and the remote magnetometer on the HASP frame. Note drawing does not reflect proposed payload seat.*

### 1.4.1 Main COMPASS Payload

The main COMPASS payload will be mounted directly on to a HASP small payload plate. It will consist of the payload frame, 3 electronics mounting shelves, a camera mounting plate, the main payload electronics, and the payload side panels.

The arrangement of the components is shown below in figure 10.

**COMPASS Interior**



| PARTS LIST | | | |
|---|---|---|---|
| ITEM | QTY | PART NUMBER | DESCRIPTION |
| 1 | 1 | HASP Small Payload Plate | |
| 2 | 1 | COMPASS Payload Frame | |
| 3 | 3 | Electronics Shelf | |
| 4 | 1 | Camera Array Mounting Plate | |
| 5 | 1 | Microcontroller | |
| 6 | 2 | Additional Electronics | Power distribution and Orientation Systems |
| 7 | 4 | Compass Side Panel | |

*Figure 10: An exploded view of the major components of the COMPASS main payload. Note no fasterneres are shown for simplicity and clarity.*

#### 1.4.1.1 HASP Plate Frame Interface

The payload frame will be mounted to the HASP payload plate via 4 ¼-20 x 1” bolts. Bolts will pass through the payload plate into 4 tapped holes in the center of the vertical pieces of the payload frame.

#### 1.4.1.2 COMPASS Payload Frame

The payload frame will be constructed out of 1” extruded aluminum T-slot railing. The frame will be a rectangular prism as shown below in figure 10 with dimensions of 5.875 x 5.875 x 8 inches.

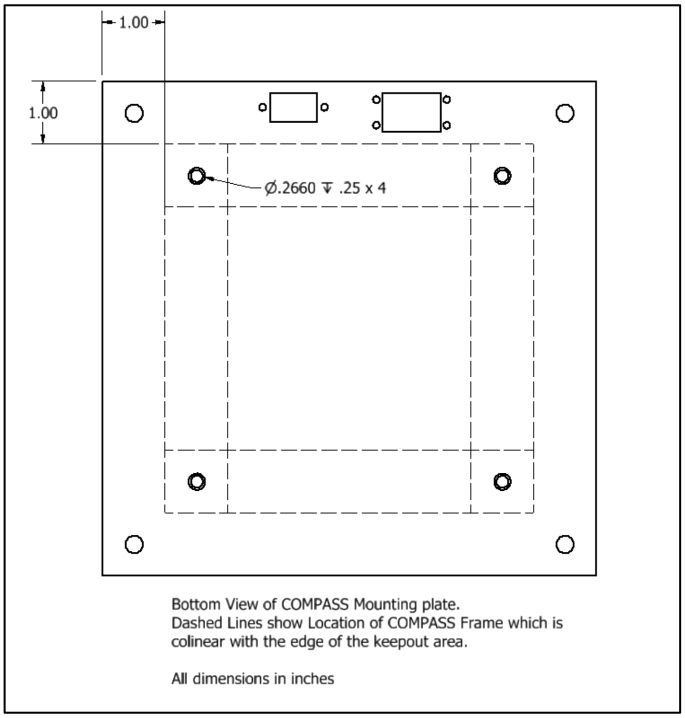
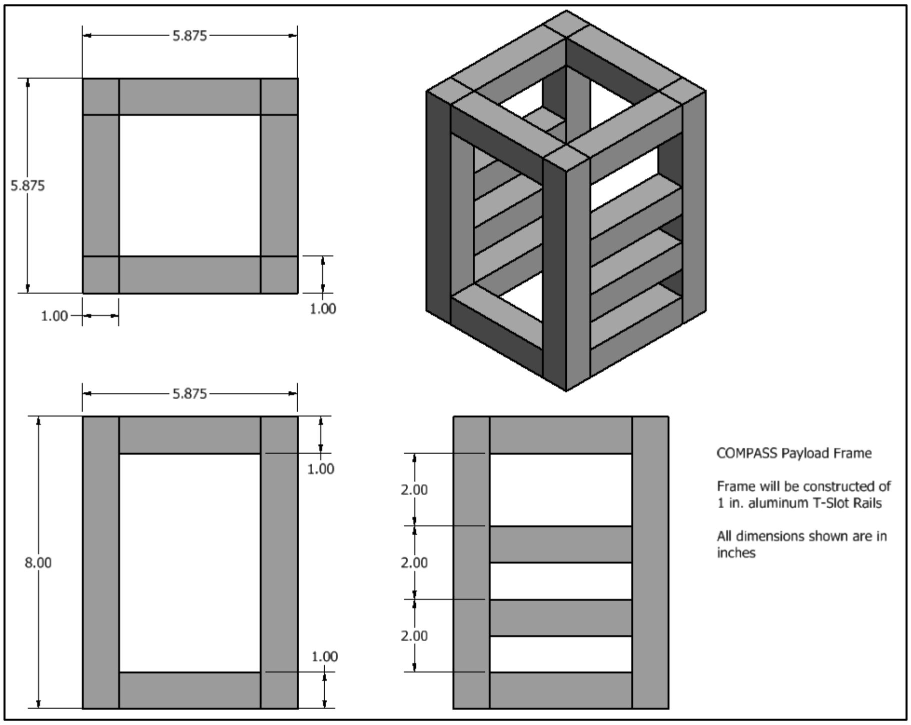

1.00
1.00
Ø.2660 ⊽ .25 x 4
Bottom View of COMPASS Mounting plate.
Dashed Lines show Location of COMPASS Frame which is colinear with the edge of the keepout area.
All dimensions in inches

*Figure 11: A bottom view the HASP mounting plate showing the proposed anchor points. Attachment will be via 1/4-20 bolts at each of the indicated through holes. The only portion of the payload extending below the payload play will be the bolt head.*

Two additional horizontal cross bars provide the mounting surface for the 2nd and 3rd electronics plates on 2 sides. The other 2 sides will remain open to allow for access to the electronics.

A center hole in the 4 vertical rails will be tapped with ¼-20 threads for mounting the payload plate on the bottom and the camera mount plate on top.

**COMPASS Frame**



COMPASS Payload Frame

Frame will be constructed of 1 in. aluminum T-Slot Rails

All dimensions shown are in inches

*Figure 12: The COMPASS main payload frame.*

### 1.4.1.3 Electronic Shelves

Each electronics shelf will consist of single piece of .032 in. aluminum cut to 3.75 x 8.75 in. The top of each plate will be covered with a thin layer of neoprene to protect against inadvertent shorting and electronics shall be mounted each plate via standoffs.

The plates will be mounted to the horizontal pieces of T-slot rail by anchoring into the T-slot. The plates are size to allow 1 in. of open space on the open sides to allow for cables to be run between different levels.

### 1.4.1.4 Camera Mount

The top layer will consist of ¼ in. plate with a mounting structure for the cameras for the camera array.
Each mount for the camera will consist of slot sized for the camera PCB to slide into and clearance on the back for the camera cables. Once the cameras are inserted, they will be held in place via a set screw.

The camera mount and plate will consist of single 3D printed piece of ABS plastic. The camera mount will also form the top of the payload structure.

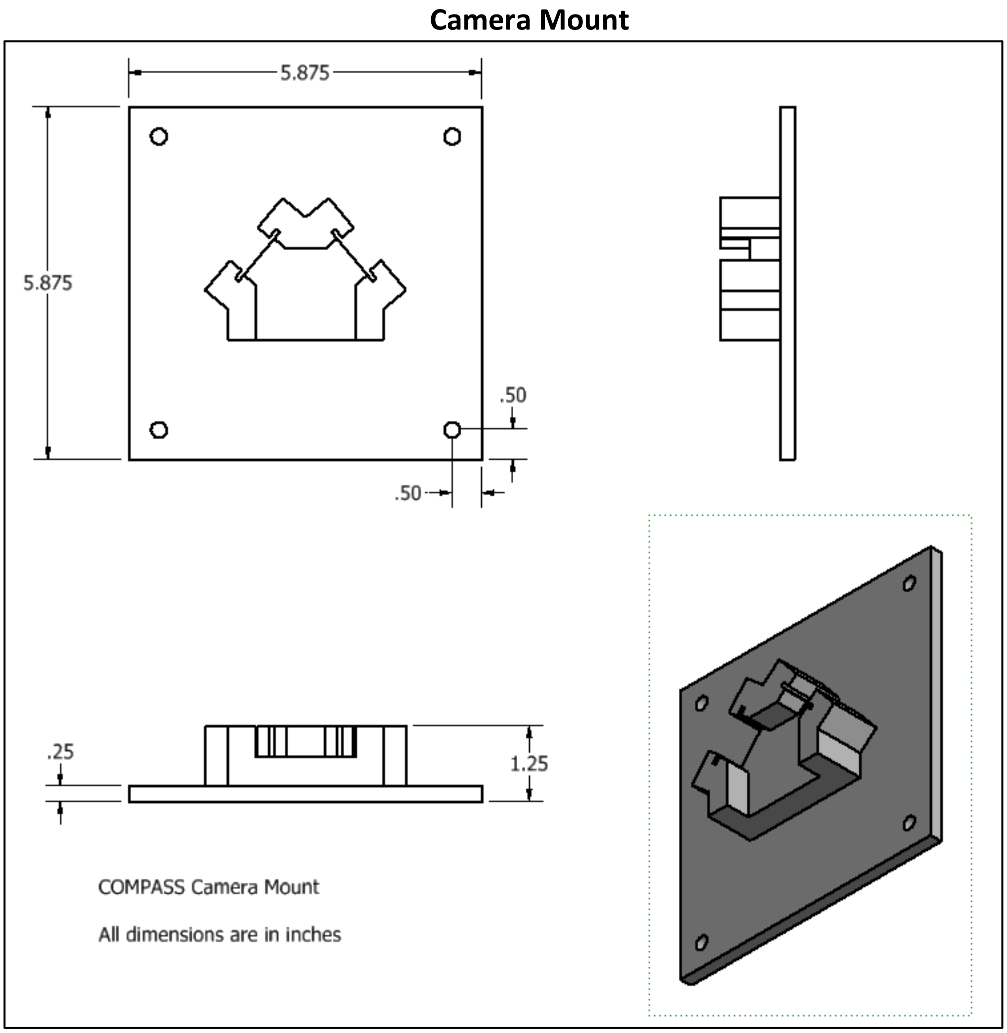


*Figure 13: Camera mounting plate*

#### 1.4.1.5 Side Panels

The side panels will be made of 0.032 in. aluminum with a polished interior surface and painted white exterior surface in order to provide a thermal regulation. The side panels will be bolted into the T-slots and had penetrations cut in them for the necessary electrical connections.

### 1.4.2 Remote Magnetometer Boom

The Remote Magnetometer will consist of a magnetometer in a plastic electronics box mounted at the end of a 96 in. long fiberglass square tube boom, identical to material used for the HASP small payload booms.

**Boom Structure**

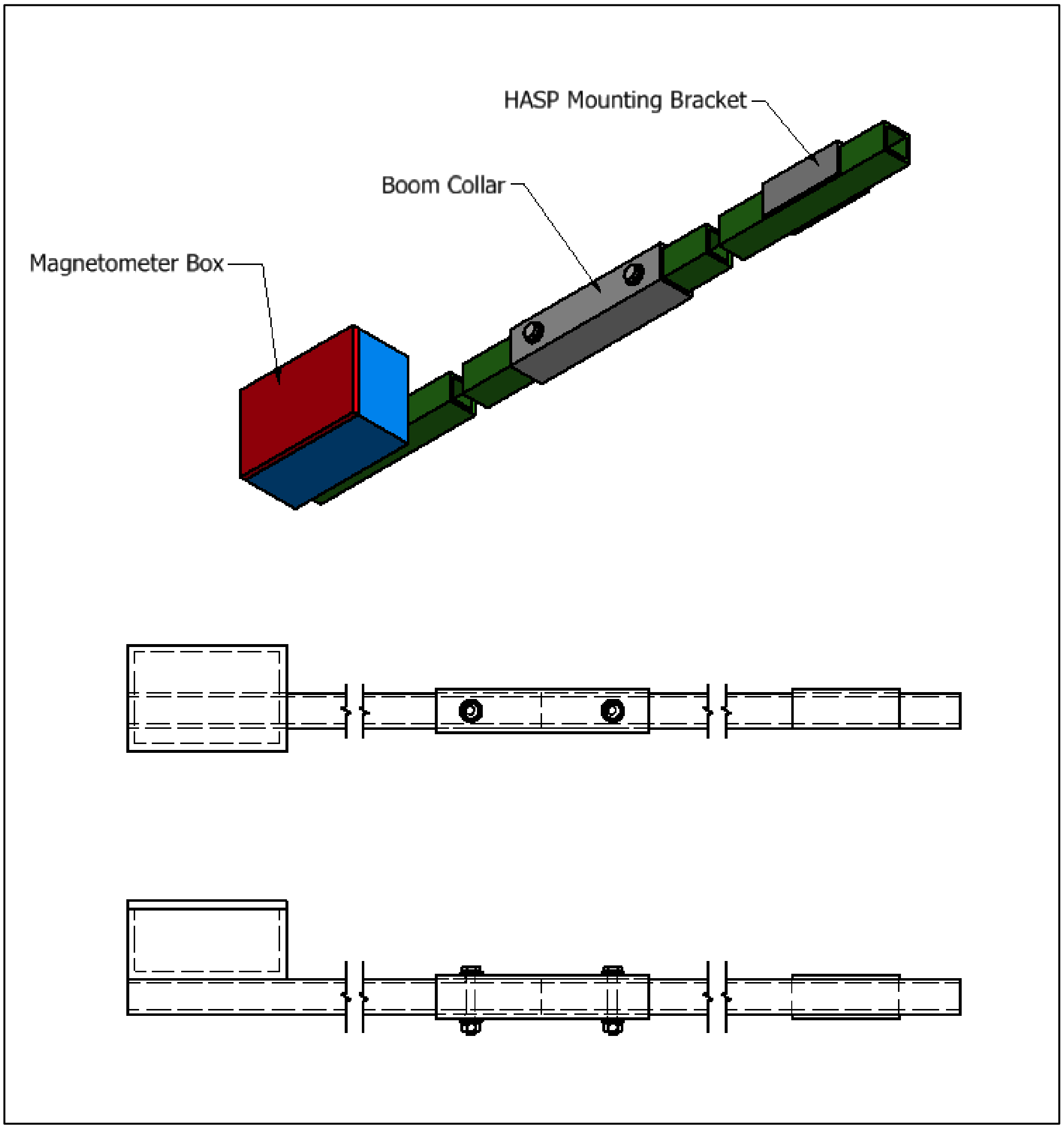


*Figure 14: The arrangement of the COMPASS magnetometer boom showing the arrangement of the 2 boom sections, the magnetometer box, the boom collar and 1 of the 2 HASP mounting brackets. Each boom section will be 48 in long with the total boom length being 96 in.*

See section 3.8 for waiver request and discussion of procedure for attaching the boom.

### 1.4.2.1 Fiberglass boom

The boom will be 1” x 1” square fiberglass tube with 1/8th thick walls.
The boom will be mounted to the HASP frame via the set of lower mount points and attached in a method identical to the HASP payload booms.

The total length of the boom will be 96 in. and will extend 76 in. past the out edge of the HASP frame. The boom will consist of 2 pieces of fiberglass tubing each 48 in. length. The first piece will be attached to HASP frame prior to installing the solar shields on HASP. The second section will be installed immediately prior to flight in order to not interfere with integration and rigging operations.

The sections of boom will be joined via a collar that over the sections boom.

### 1.4.2.2 Boom collar

The boom sections will be joined together via a collar made of steel square tube slightly larger ID (1.01") than the fiberglass boom OD (1.00").

Each end of the boom will be inserted 3 in. into the collar and fixed in place via ¼ 20 nut and bolt penetrating through the length of the collar on boom and fixed in place with a nut and lock washer.

### 1.4.2.3 Electronics box

The electronics at the end of the boom will consist of the magnetometer and support electronics. The electronics will be mounted in a standard off the shelf electronics project box of 6.5 x 5 x 3.5 inches.

The electronics box will be mounted to the boom via a pair of ¼ -20 nuts and bolts passing though the top and bottom of the electronics box and the through the fiberglass boom.

### 1.4.2.3 Cable Management

A single cable shall provide all electrical connections between the electronics stack and remote magnetometer. This cable shall be run along HASP payload boom. Then vertically outside the HASP solar shields. And finally down the length of the remote magnetometer boom.

The cable will be secured to the booms via zip ties wrapped around the cable and boom.

### 1.4.2.4 Boom Stress Analysis

We expect the boom to face a maximum stress at the point where the boom is joined to the HASP frame. Assuming a maximum expected 10g upward acceleration we expect to see a peak stress of ~6 KPSI compared to a flexural strength of 11 KPSI for the fiberglass tubing.

**Boom Stress Analysis**

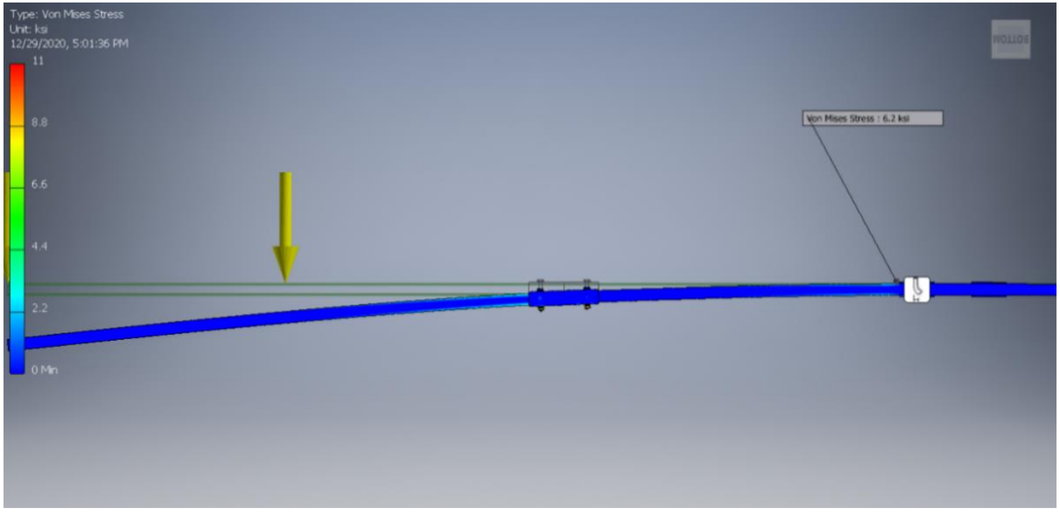


*Figure 15: The results of a finite element analysis of the boom structure. The boom was simulated under a 10g upward acceleration with an additional load of 55,000 N applied to the end of the boom to simulate a 563 g magnetometer box under a 10g acceleration. The text box shows the peak stress near the mating point between the boom and the HASP mounting bracket of 6.2 KPSI.*

Under a more typical 1g we expect to see a deflection of the end of the boom of ~0.7 in. which would result in an deflection of ~0.5 degrees.

## 1.5 Electrical Design

### 1.5.1 Control Electronics

The central component of the COMPASS control system will be a Raspberry Pi 4 Model B microcontroller. The RPi4 has various interface options responsible for communicating with each of the major system components to receive and transmit data. In addition, there will be another microcontroller, an Arduino Mega 2560, on the boom that is responsible for collecting data from the second orientation system and sending this data to the main control system using the necessary serial communication.

Because there are certain limitations to consider when determining what kind of GPS system to use, we must find a GPS system that is operational at least 120,000ft and that can receive satellite signals inside the metal HASP payload box. One solution could be the Copernicus II DIP GPS Module by Sparkfun which can operate at a maximum of 164,000ft, and can also mate with the Sparkfun SMA GPS antenna that includes an onboard low noise amplifier. If we were to use this option, we would have to consider drilling a hole through one of the payload walls so the antenna can be mounted outside of the payload box. The onboard GPS will be sampled at the same rate as the magnetometer to correlate GPS and magnetometer readings. In addition, we plan on requesting HASP GPS data periodically (at a slower rate than the onboard GPS) for redundancy.

**Control System Diagram**

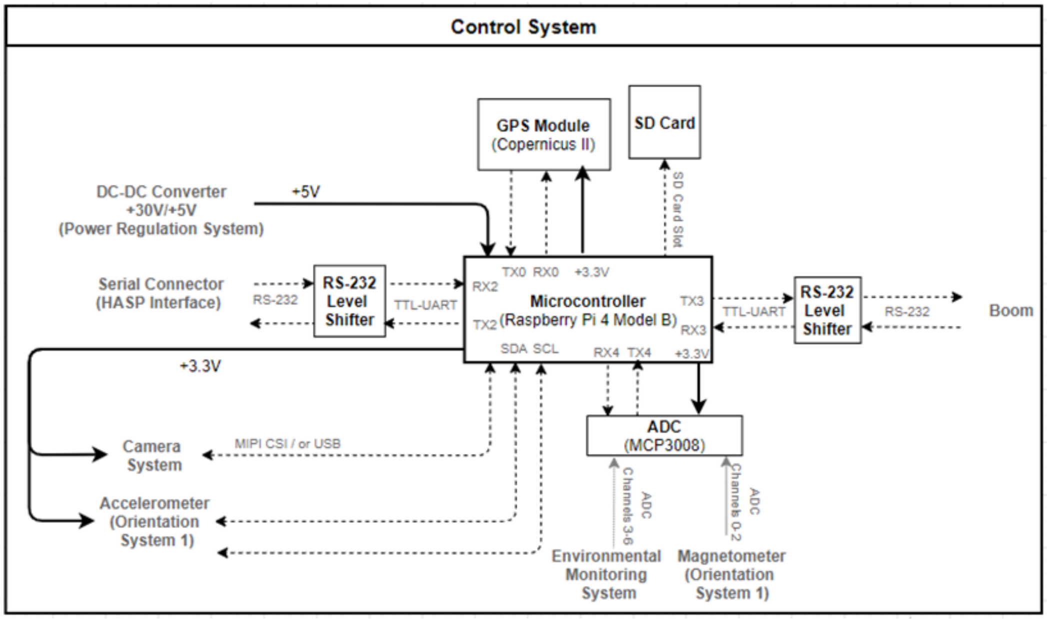


*Figure 16: Control electronics high level system diagram*

### 1.5.2 Orientation System 1

**Orientation System 1**

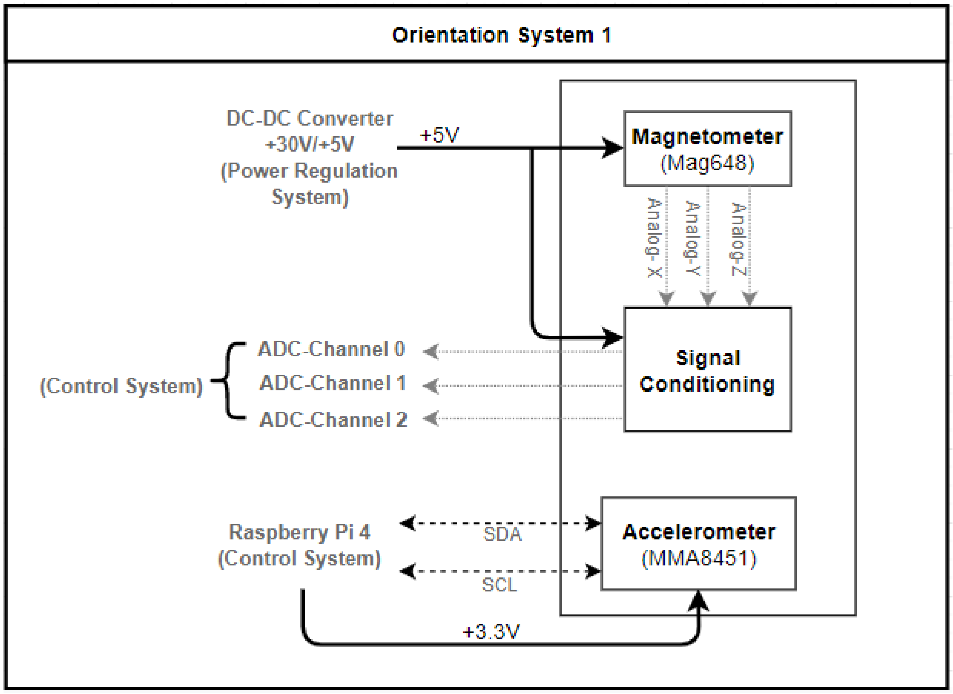


*Figure 17: Lower level diagram of the components and communication between components for OS1*

The components of Orientation System 1 will be located on a protoboard compatible with the RPi4 that will include a Bartington Mag648 3-axis magnetometer and an Adafruit MMA8451 3-axis accelerometer. The Mag648 is has a dual-selectable range of ± 60uT/or ±100uT that measures magnetic field readings in 3-axis and is an analog device that will have to connect to the ADC of the control system to communicate with the Raspberry Pi. The MMA8451 accelerometer has a usage range from +-2g to +-8g that measures acceleration in 3-axis and will communicate with the Arduino Mega via I2C communication.

### 1.5.3 Environmental Monitoring System

We are considering a few different types of temperature sensors. There are conventional temperature sensors such as diodes, which would require manual signal conditioning and calibration, however there are also analog temperature sensors with already build in signal conditioning that would be easier to set up. We would also have to consider using the ADC chip on the control system to convert the temperature signal from analog to digital so it can be read by the Raspberry Pi.

The four temperature sensors on the main payload would monitor the 30V to 12V DC-DC converter, the 30V to 5V converter, the magnetometer, and the camera system because these are the most important parts of the payload.

**Environmental Monitoring Detailed Diagram**

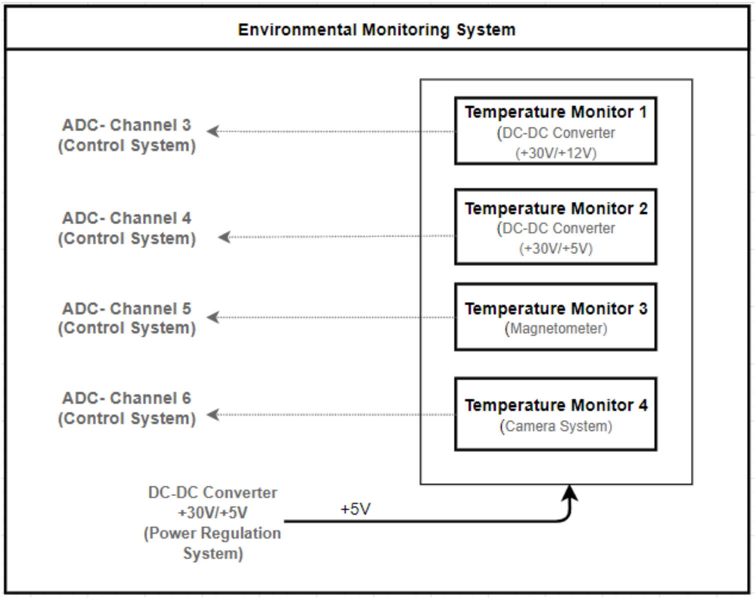

Environmental Monitoring System
ADC- Channel 3 (Control System)
ADC- Channel 4 (Control System)
ADC- Channel 5 (Control System)
ADC- Channel 6 (Control System)
Temperature Monitor 1 (DC-DC Converter (+30V/+12V)
Temperature Monitor 2 (DC-DC Converter (+30V/+5V)
Temperature Monitor 3 (Magnetometer)
Temperature Monitor 4 (Camera System)
DC-DC Converter +30V/+5V (Power Regulation System)
+5V

*Figure 18: This diagram shows a lower level view of the environmental monitoring components.*

### 1.5.4 Power Subsystem

COMPASS will be supplied ~+30V by the HASP interface system and will have its own power regulation system to regulate the required voltage going to each of its components. This power subsystem will consist of two DC-DC converters that will step down the +30V into two output voltages, one to +5V and the other to +12V. The RPi4 on the HASP payload plate has its own voltage regulation system in which it will provide power to the accelerometer on orientation system 1, camera system, and GPS system. The Arduino Mega on the boom also had its own +5V and +3.3V voltage regulators to power the second orientation system.

The P78xx-2000 series offer step-down switching regulators at +5V/+12V that can operate at a maximum input of 36Vdc and supply up to 2A of output current. We currently have access to a custom converter board, the “Murata DCDC,” that is designed for the Murata LM7805 and LM812 converters. However, they don’t supply enough current output so we would be able to switch these out for the P7805 and P7812 converters that meet out power requirements. This board also includes voltage monitors that communicate with the analog pins on the EDAC pin to make sure the converters are providing stable power. To further monitor the power output of these converters, we will add current monitors that will send current output signals to the

Raspberry Pi through the ADC on the control system. Overall, this power-converter module would be a feasible option for the COMPASS power system design.

**Power System Diagram**

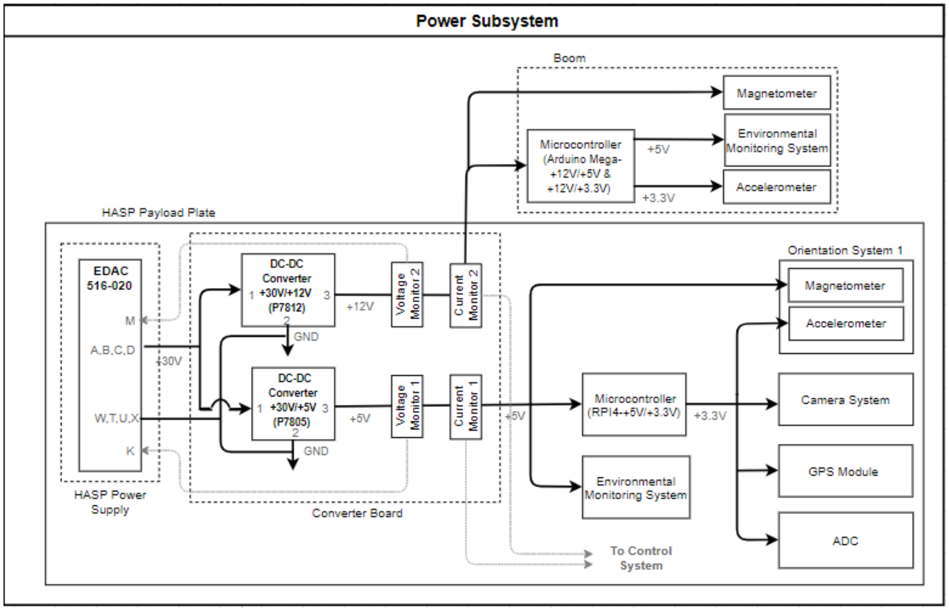


*Figure 19: Power subsystem high level diagram. This shows how each component on the payload is powered.*

### 1.5.5 Camera System

Currently we are considering two options for cameras:

**Option 1: Camarray – Arducam 1MP Stereoscopic Camera Bundle Kit**

- With M40160 M12 lens has 118 horizontal and 92 vertical FoV
- 1 MP
- Attaches directly to RPi camera module and communicates through serial communication
- 3.3 V power input from Rpi

**Option 2: Arducam 0.36MP MT9V022 Monochrome 1/3" CMOS Camera Module**

- With M30225 H10 M12 lens has 120 degree horizontal and 80 degree vertical FoV
- 0.36 MP
- Communicates through USB
- 3.3 V power input
- Also requires a USB3 camera shield and a stereo camera adapter board (also from Arducam)

## Camera Requirements Checklist

| Camera Setup/Requirement | Covers angular area of 190 degrees in azimuth and 50 degrees in elevation (2.1) | Max angular error of <1/4 degree (2.2) | Compatible with Raspberry Pi (2.5) |
|---|---|---|---|
| **Option 1** | Yes | Yes | Yes |
| **Option 2** | Yes | Yes | Yes |

*Table 1: Table showing the camera options and the requirements they satisfy. There are more verification system requirements detailed in the previous section but these are the ones pertinent to the camera itself.*

The two options are described by the images below:

### Camera System Option 1

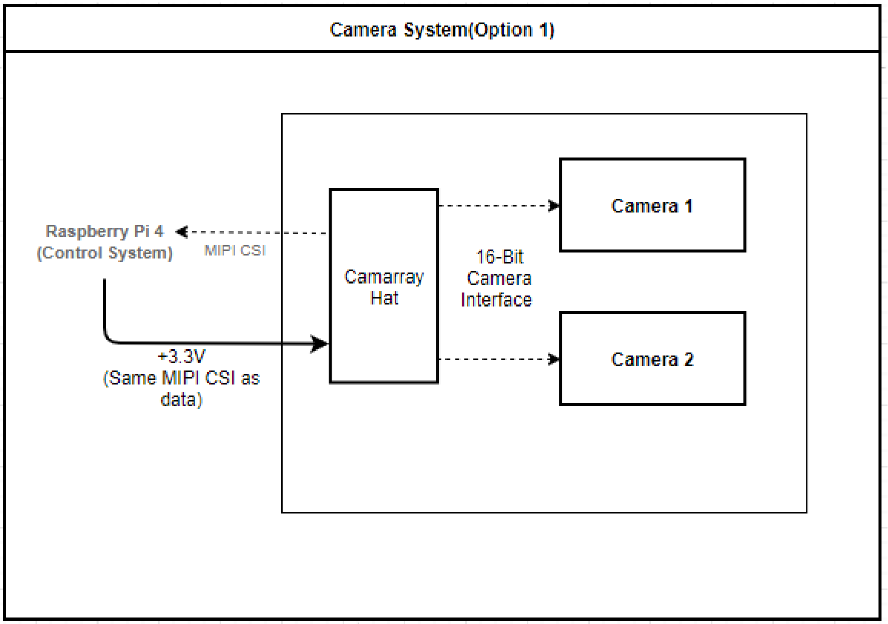


*Figure 20: Design diagram showing the components required for camera option 1.*

**Camera System Option 2**

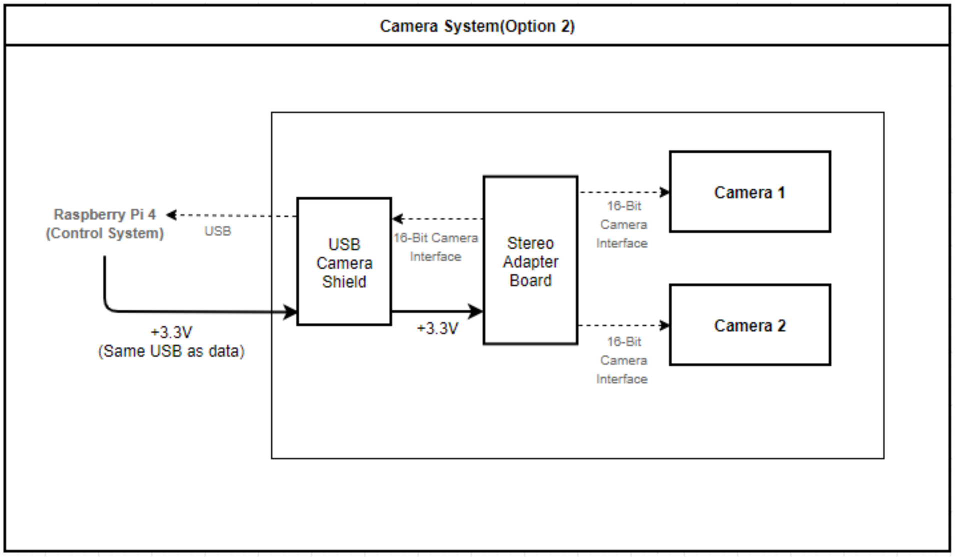


*Figure 21: Design diagram showing the components required for Camera option 2*

### 1.5.6 Orientation System 2

An Arduino Mega 2560 would satisfy the requirements for a microcontroller to read detector data and convert the analog signals to digital signals and to convert the accelerometer I2C signals to TTL-UART. It would also be able to supply power to the components on the boom with its onboard +3.3V/+5V linear regulators.

The same Mag648 magnetometer and MMA8451 accelerometer will be used in this orientation system as the first orientation system.

**Orientation System 2 Detailed Diagram**

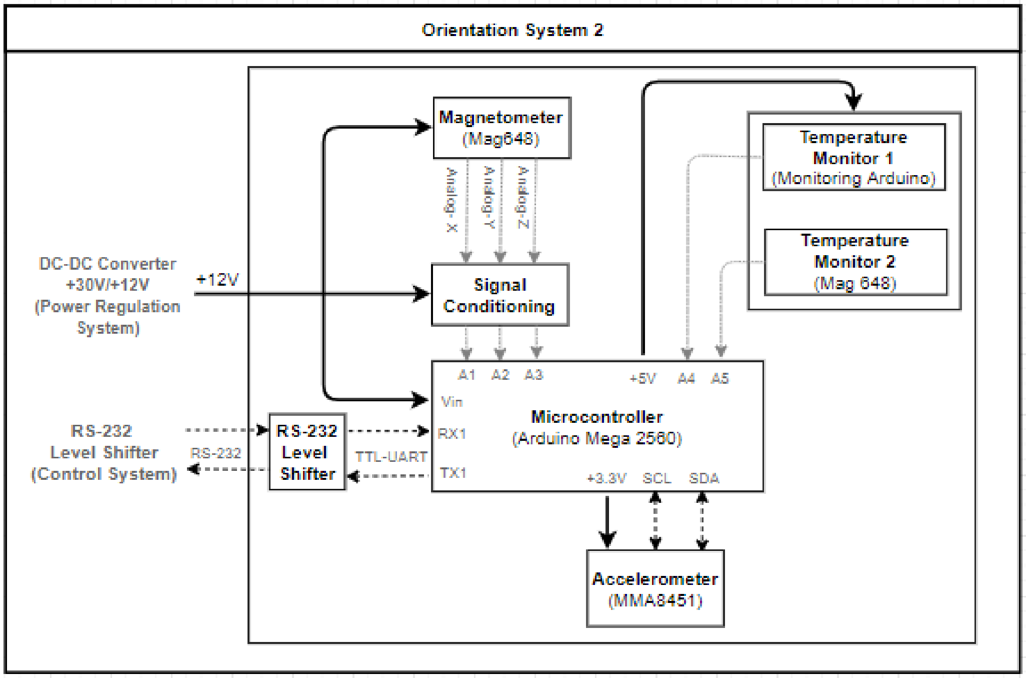


*Figure 22: This diagram shows all to components and communication protocols and pins in OS2.*

## 1.6 Thermal Control Plan

During flight, our payload should expect to encounter ambient temperatures from -30°C to 45°C according to the **HASP Call for Payloads** document. Based on the operating temperatures of the major system components in the table below, they should be able perform under the expected thermal conditions during flight. To ensure the performance of these components, the payload will be constructed with suitable insulation. For this, the payload will use aluminum plating as siding, with a polished inside to reflect infrared radiation. The outside of the aluminum siding will be painted "appliance white" to reflect visible light radiation.

Basic thermal computational models will be used later in the payload development process to determine if the temperature of critical components will exceed their operating or survival temperatures (given in Table 2 below). If it is found that these temperatures will be reached or exceeded, then actions will be taken to improve and adapt the thermal conditioning plan. To further test the performance of our payload, a thermal/vacuum test will be completed during the time designated for student payload integration prior to the launch day.

Below is a table indicating the operating temperature ranges of the COMPASS payload's components. "Storage/Survival Temperature" is the temperature that components should not exceed, even when they are not operating. Exceeding the storage/survival temperature may cause the components to not function properly once they return within operating temperature. As such, COMPASS will ensure that these temperatures will not be exceeded, using the means described above.

**Component Operating Temperatures**

| System Component | Operating temperature | Storage/Survival Temperature |
|---|---|---|
| RPi4 Model B | 0°C to 50°C | <-53°C to >84°C |
| Arduino Mega | -40°C to 85°C | -40°C to 150°C |
| Magnetometer | -40°C to 70°C | -40°C to 60°C |
| Accelerometer | -40°C to 85°C | -40°C to 125°C |
| Camera System | -30°C to 85°C | TBD |

*Table 2: The operating and survival temperature ranges of each critical component of the COMPASS payload. A storage/survival temperature could not be found for the camera system, so thermal testing will be conducted during the payload development to determine this max temperature range.*

Given the operating temperature range of the Raspberry Pi 4 Model B in Table 2, it initially appears that it would not be able to withstand the temperature range of -30°C to 45°C. However, because the Pi will generate a significant amount of heat from its operation, it is speculated that its temperature will not fall below the minimum operating temperature of 0°C. Many past HASP missions have also flown payloads using Raspberry Pis and did not face issues with temperature regulation of their microcontrollers (such as Sheffield University's Nova Balloon Lifted Telescope payload in 2019). As such, COMPASS does not plan on including any active form of heat management for the microcontroller.

Though there are not concerns for falling below the operating temperature, it is thought that the microprocessor may generate too much heat and overheat during its operation. If it is deemed necessary from the thermal modelling or thermal/vacuum testing, a heat sink will be added to the microcontroller by connecting the main processor to the aluminum siding.

# 2. Team Structure and Management

## 2.1 Team Organization and Roles

### 2.1.1 Team Roles

COMPASS will be designed and constructed by a student team. The team's structure and member contact information are given in Table 3. The team roles will be project lead, electrical lead, mechanical lead, software lead, and a mechanical and electrical secondary.

Jeanne Garriz, a physics undergraduate, will be the project lead. She will be responsible for team management, the compilation/submission of deliverables, and communication with LaSPACE and HASP management. She will also work on payload's mechanical design and fabrication.

Sabrina Huezo, also a physics undergraduate, will be the electrical lead. She will be responsible for the electrical design and implementation of the various electrical systems that make up COMPASS. She will be responsible for the design and ordering of any necessary PCBs and delegating tasks to secondary.

Jesse Frank, an electrical and computer engineering undergraduate, will be the software lead. He will be responsible for writing the in-flight software that controls the payload, formatting communication between HASP and the payload, and writing the ground software used to analyze the collected data.

Harrision Gietz, a physics undergraduate, will be the secondary for electrical and mechanical design. He will be responsible for the thermal control system and assisting the electrical and mechanical leads (Sabrina, Aaron) in larger tasks.

Aaron Ryan will be the faculty advisor for COMPASS and will provide advice, assistance, and guidance as needed. He will also be the designated mechanical lead and will work on the mechanical design and implementation of the payload.

### 2.1.2 Team Organization

**Team Organization Chart**

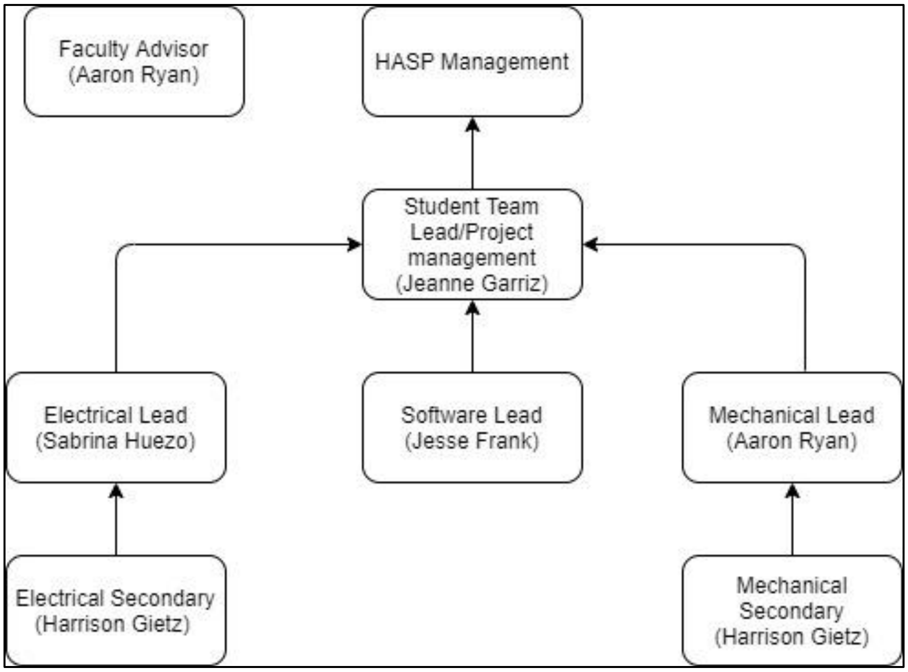

Faculty Advisor (Aaron Ryan)
HASP Management
Student Team Lead/Project management (Jeanne Garriz)
Electrical Lead (Sabrina Huezo)
Software Lead (Jesse Frank)
Mechanical Lead (Aaron Ryan)
Electrical Secondary (Harrison Gietz)
Mechanical Secondary (Harrison Gietz)

*Figure 23: Team organization chart showing the hierarchy and structure of the COMPASS development team.*

**Team Roles and Contact Information**

| Role | Name and Title | Contact Info |
|---|---|---|
| Faculty Advisor | Aaron Ryan<br>Staff | aryan21@lsu.edu<br>808-783-9096 |
| Project Lead/Mechanical Lead | Jeanne Garriz<br>Undergraduate | jgarri9@lsu.edu<br>251-391-2596 |
| Electrical Lead | Sabrina Huezo<br>Undergraduate | shuezo1@lsu.edu<br>818-621-5081 |
| Software Lead | Jesse Frank<br>Undergraduate | jfran82@lsu.edu<br>225-424-9385 |
| Mechanical/Electrical Secondary | Harrison Gietz<br>Undergraduate | hgietz2@lsu.edu<br>225-223-8358 |

*Table 3: This table contains the roles, names, and contact information of COMPASS development team members.*

### 2.1.3 Deliverables/Deadlines

The project lead shall be in charge of compiling and writing up deliverables. These include monthly status reports, the preliminary PSIP, final PSIP, FLOP, and security documents required for integration. In addition, the team will complete an internal PDR, CDR, and FRR to assess progress toward the deliverables. The project lead shall set internal deadlines for parts of deliverables or drafts from other team members. For a draft or design review (not a deliverable for HASP management), information shall be emailed to the project lead at least 1 day in advance of the update or review meeting for which the draft is being completed. For a deliverable, the information shall be emailed to the project lead 2 days in advance to give enough time for each team member to review the document and request minor changes if necessary, after the document is compiled. Each team member is responsible for adhering to these deadlines to the best of their abilities and notifying the project lead via text or email as soon as they think they will be unable to meet a deadline. If a team member is unable to meet a deadline, the rest of the team shall assist to complete the task.

### 2.1.4 Version Control/Documentation

All drafts of deliverables, code, mechanical and electrical schematics, flowcharts, schedules, and anything that will have a final version used in a deliverable or the final payload are considered working documents and shall be subject to version control. Working documents shall be given version numbers for major and minor changes.

Controlled documents are the final versions of deliverables that can only be edited by the project lead with the agreement of all team members. Once the final draft of a deliverable is approved by the rest of the team, compiled, and turned in, that working document becomes a controlled document. These controlled documents shall be assigned the last version number of the document while it was a working document.

### 2.1.5 COVID-19 Guidelines

Due to the ongoing COVID-19 pandemic, team meetings will be conducted online via Zoom, and work will be conducted remotely whenever possible. However, when the team gets to the prototyping and development stage and needs to work in the lab, all CDC guidelines will be followed. These guidelines will be followed until the COVID vaccine is more widely available and the CDC lifts their official guidelines.

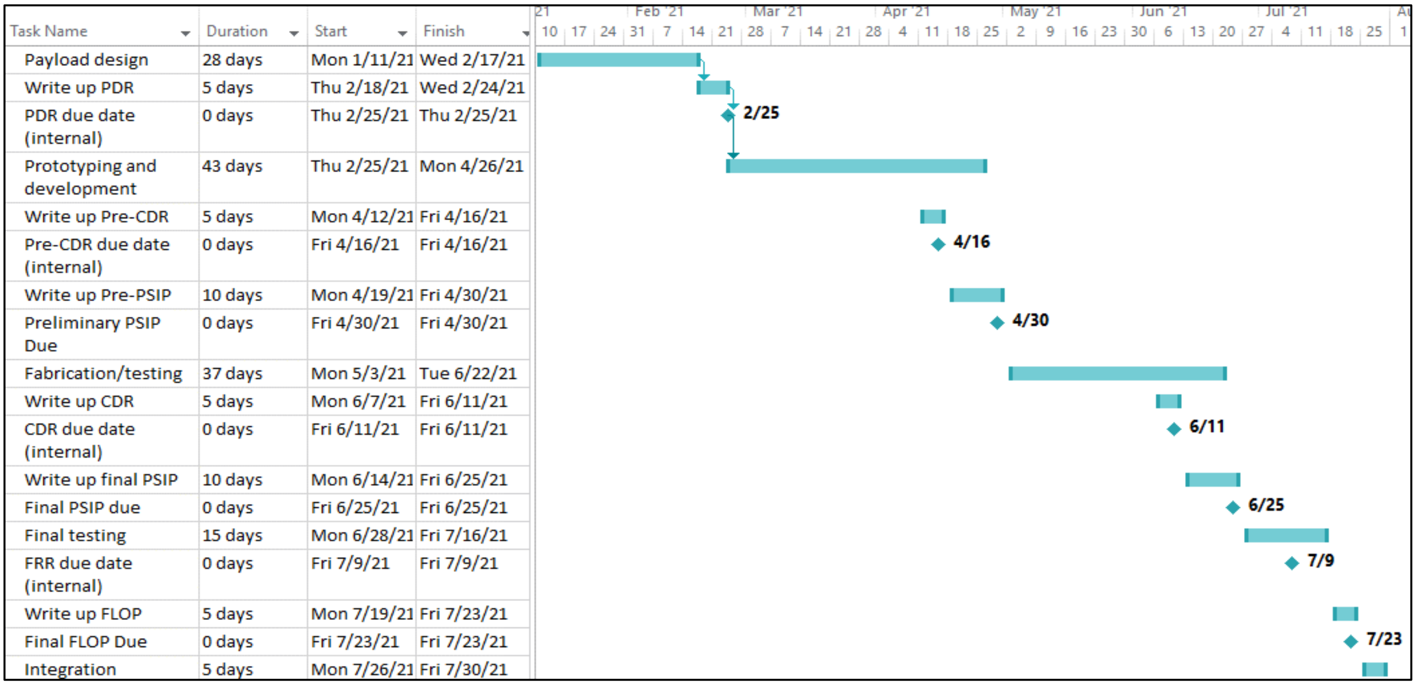
As of right now, the team is working under the assumption that things will be mostly back to normal by the time of integration and launch, so these events are being planned for as normal.

### 2.1.6 Meetings

Meetings between all team members shall be held weekly, at a time decided upon once schedules for next semester are more well known. Update meeting between the COMPASS Development Team and HASP management shall be conducted once every two weeks. Team members shall attend every meeting. If a team member is unable to attend, they shall notify the project manager as soon as they know they will be unable to attend. The project lead shall set the meeting agendas and shall take notes during meetings; however, it is recommended that other team members take notes as well.

## 2.2 Timeline and Milestones

**COMPASS High-Level Schedule**



| Task Name | Duration | Start | Finish |
|---|---|---|---|
| Payload design | 28 days | Mon 1/11/21 | Wed 2/17/21 |
| Write up PDR | 5 days | Thu 2/18/21 | Wed 2/24/21 |
| PDR due date (internal) | 0 days | Thu 2/25/21 | Thu 2/25/21 |
| Prototyping and development | 43 days | Thu 2/25/21 | Mon 4/26/21 |
| Write up Pre-CDR | 5 days | Mon 4/12/21 | Fri 4/16/21 |
| Pre-CDR due date (internal) | 0 days | Fri 4/16/21 | Fri 4/16/21 |
| Write up Pre-PSIP | 10 days | Mon 4/19/21 | Fri 4/30/21 |
| Preliminary PSIP Due | 0 days | Fri 4/30/21 | Fri 4/30/21 |
| Fabrication/testing | 37 days | Mon 5/3/21 | Tue 6/22/21 |
| Write up CDR | 5 days | Mon 6/7/21 | Fri 6/11/21 |
| CDR due date (internal) | 0 days | Fri 6/11/21 | Fri 6/11/21 |
| Write up final PSIP | 10 days | Mon 6/14/21 | Fri 6/25/21 |
| Final PSIP due | 0 days | Fri 6/25/21 | Fri 6/25/21 |
| Final testing | 15 days | Mon 6/28/21 | Fri 7/16/21 |
| FRR due date (internal) | 0 days | Fri 7/9/21 | Fri 7/9/21 |
| Write up FLOP | 5 days | Mon 7/19/21 | Fri 7/23/21 |
| Final FLOP Due | 0 days | Fri 7/23/21 | Fri 7/23/21 |
| Integration | 5 days | Mon 7/26/21 | Fri 7/30/21 |

*Figure 24: A high level Gantt chart detailing the tasks and milestones involved in developing COMPASS*

## 2.3 Anticipated Participation in Integration and Launch operations

The entire team anticipates participating in HASP integration activities and testing the week of July 26th-30th at CSBF to oversee the integration stage, as well as troubleshoot any issues that may arise. The integration team shall consist of Jeanne Garriz, Sabrina Huezo, Harrison Gietz, Jesse Frank, and Aaron Ryan. Two team members will be selected closer to launch and will oversee launch operations in Fort Sumner.

# 3. Payload Interface Specifications

## 3.1 Weight Budget

The estimated weight budget for the electrical components in the COMPASS mission is detailed in Table 4 below:

**Preliminary Electrical Component Weight Budget**

| Component | Mass per Unit (g) | Quantity | Mass (g) | Unc. (g) | Measurement Method |
|---|---|---|---|---|---|
| Accelerometer | 1.3 | 1 | 1.3 | ±0.1 | Estimate |
| Magnetometer | 120 | 1 | 120 | ±1 | Estimate |
| Raspberry Pi | 50 | 1 | 50 | ±5 | Weight Measurement |
| DC-DC Converters | 40 | 2 | 80 | ±10 | Estimate |
| Wiring | - | - | 15 | ±5 | Calculation |
| GPS System | 25 | 1 | 25 | ±5 | Weight Measurement |
| Cameras | 24 | 2 | 50 | ±10 | Estimate |
| Internal Temperature Sensor | 15 | 1 | 15 | ±5 | Estimate |
| **Total Electrical** | | | **356.3** | **±17.3** | |

*Table 4: Preliminary internal/electronics weight budget, not including the boom (see special requests, 3.8).*

The estimate weight budget for the structural components in the COMPASS mission is detailed in table 5 below:

**Preliminary Structural Weight Budget**

| Component | Mass per Unit (g) | Quantity | Mass (g) | Unc. (g) | Measurement Method |
|---|---|---|---|---|---|
| Aluminum Side Panels | 67.5 | 4 | 270 | ±14 | Calculation |
| Internal Electronics Shelves | 32.7 | 3 | 98 | ±5 | Calculation |
| Fasteners | - | - | 200 | ±20 | Estimate |
| Vertical T-frame | 154 | 4 | 617 | ±31 | Calculation |
| Horizontal T-frame | 74.7 | 12 | 896 | ±45 | Calculation |
| Camera Mount | 226 | 1 | 226 | ±11 | Calculation |
| Neoprene | 7 | 3 | 21 | ±1 | Calculation |
| **Total Structural** | | | **2328** | **±61** | |

*Table 5: Preliminary structural weight budget, not including the boom (see special requests, 3.8).*

The total preliminary weight budget for COMPASS is estimated below, using the results of Tables 4 and 5.

**Total Weight Budget**

| Section of Payload | Mass (g) | Unc. (g) |
|---|---|---|
| Structural | 2328 | ±61 |
| Electronics | 356.3 | ±17.3 |
| **Total** (cannot exceed 3000g) | **2684.3** | **±63.4** |

*Table 6: Total estimated weight budget for the COMPASS payload, not including the boom (see special requests, 3.8).*

As is shown, the estimated mass of the COMPASS payload falls below the limit of 3kg and meets the guidelines of the **Call for Payloads** document. The final weight estimate allows for a contingency of approximately 10% of the maximum possible weight.

## 3.2 Power Budget

The power system diagram in Section 1.5 gives a more detailed design of how our payload will utilize the power provided from HASP, but the power budget below gives an outline of how much power each of the major components will consume based on their operating voltage and estimated maximum current draw. The total system has a power limitation of 15W based on the power requirements outlined in the **HASP Interface Manual** and thus cannot exceed this limit.

Below are tables that show the power consumed by the 12V and 5V lines, used for the boom and main payload plate respectively. The power consumed by the Raspberry Pi and its loads is added in Table 9, prior to computing the total power consumption for the entire 5V line in Table 10.

**Arduino Power Consumption**

| COMPONENT | OPERATING VOLTAGE (V) | UPPER LIMIT CURRENT DRAW (mA) | POWER CONSUMED (W) | UNC. (W) |
|---|---|---|---|---|
| Accelerometer | 3.3 | ~2 | <1 | ±0.01 |
| Temperature sensor | 5 | ~2 | <1 | ±0.01 |
| Bare-board Arduino Mega | 12 | 200 | 2.4 | ±0.24 |
| TOTAL |  | 204 | 4.4 | ±0.24 |

*Table 7: This table shows the power consumption of the bare board Arduino Mega and also the parts that will be powered through the Mega*

**12V Power Budget**

| COMPONENT | OPERATING VOLTAGE (V) | UPPER LIMIT CURRENT DRAW (mA) | POWER CONSUMED (W) | UNC. (W) |
|---|---|---|---|---|
| Arduino Mega (includes load of accelerometer, temperature sensor, and Active Bare-Board Mega) | 12 | 204 | 2.5 | ±0.24 |
| Magnetometer | 12 | 6 | 0.07 | ±0.01 |
| Boom signal conditioning | 12 | 3.2 | 0.05 | ±0.01 |
| TOTAL |  | 213.2 | 2.62 | ±0.26 |

*Table 8: Power budget for all electronics connected to the 12V supply from the P7812-2000-S converter. These are the electronics that will be housed on the boom.*

### Raspberry Pi Power Consumption

| COMPONENT | OPERATING VOLTAGE (V) | UPPER LIMIT CURRENT DRAW (mA) | POWER CONSUMED (W) | UNC. (W) |
|---|---|---|---|---|
| Camera System* | 3.3 | 97 | 0.32 | ±0.03 |
| Rasp Pi Active Bare-Board | 5 | 600 | 3.0 | ±0.3 |
| GPS Module | 3.3 | 33 | 0.11 | ±0.01 |
| Accelerometer | 3.3 | ~2 | <1 | ± 0.01 |
| TOTAL | | 732 | 3.5 | ±0.3 |

*Table 9: The maximum power to be used from the Raspberry Pi and the loads it carries.*

*Power consumption could not be found for the intended camera system, so it was estimated using a similar single 0.36MP camera. It is likely that the actual current draw will be larger than this estimate, as two cameras will be used on COMPASS.

### 5V Power Budget

| COMPONENT | OPERATING VOLTAGE(V) | UPPER LIMIT CURRENT DRAW(mA) | POWER CONSUMED(W) | UNC. (W) |
|---|---|---|---|---|
| RPi4 Model B (includes loads of GPS, Camera System, Accelerometer, and Active Bare-Board RPi) | 5 | 732 | 3.7 | ±0.3 |
| Environmental sensors/Signal Conditioning | 5 | 50 | 0.25 | ±0.05 |
| Magnetometer | 5 | 6 | 0.03 | ±0.01 |
| TOTAL | | 788 | ~4 | ±0.3 |

*Table 10: Power budget for all electronics connected to the 5V supply from the P7805-2000-S converter. These are the electronics that will be housed on the main payload mounting plate*

Below the details of the power consumption and efficiency of the DC-DC converters is shown. Using the estimated maximum current output from the above tables, the required total input power was found with the efficiency of the converters. The total power consumed by the two converters is the total power consumption of COMPASS, given in the bottom right of the table.

### Power Consumption of DC-DC Converters

| COMPONENT | Voltage Out (V) | Current Out Max (A) | Power Out Max (W) | Efficiency (%) | Power In Max (W) | Power In Max Unc. (W) | Current in @30 V (A) |
|---|---|---|---|---|---|---|---|
| P7812 Converter | 12 | 0.2136 | 2.6 | 90 | 2.89 | ±0.29 | 0.1 |
| P7805 Converter | 5 | 0.788 | 3.94 | 85 | 4.64 | ±0.46 | 0.15 |
| **TOTAL (entire payload)** | | | | | **7.53** | **±0.54** | **0.25** |

*Table 11: Current and power consumption information for the DC-DC converters, and the total power consumption of the COMPASS payload (given in the bottom right).*

As shown, the total maximum power consumption does not exceed the upper limit given in the Call for Payloads document of 15W at 0.5A current draw and allows for a contingency of around 50% of the maximum available power consumption.

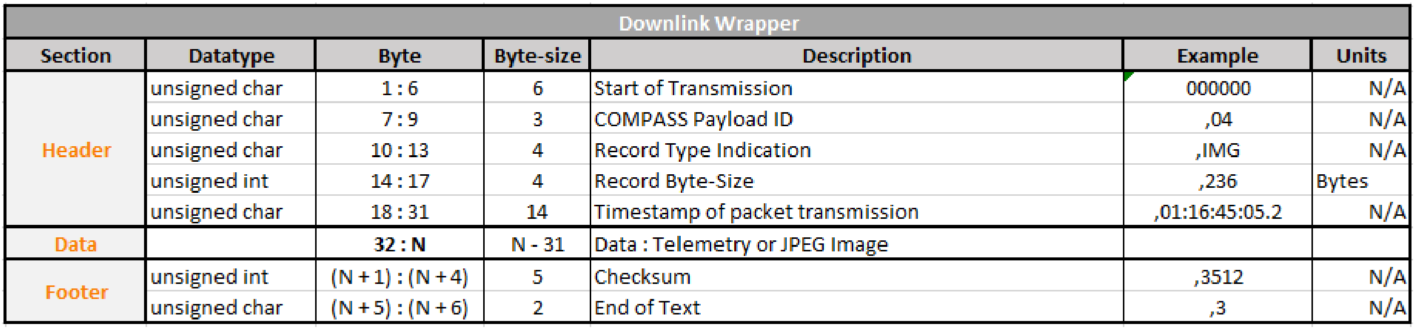
The max uncertainties in the above tables are estimated to be around 10% of the calculated power values.

## 3.3 Downlink Serial Data

COMPASS intends to downlink serial data from HASP. The total size of the data package will be dependent on the whether the data packet being sent is an image data packet, orientation/telemetry data packet, or process troubleshooting data packet. The orientation/telemetry data will be the default packet sent- images will only be sent on command, and process troubleshooting data will only be sent with the system is commanded into troubleshooting mode. All downlinked data will be transmitted as ASCII characters, while the datatype in the tables refer to the way the payload's computing system will store the values in memory. Research into camera types, image compression, and image downlink feasibility are still in progress.

**Downlink Wrapper:**

- Wrapper Length:  38 Bytes

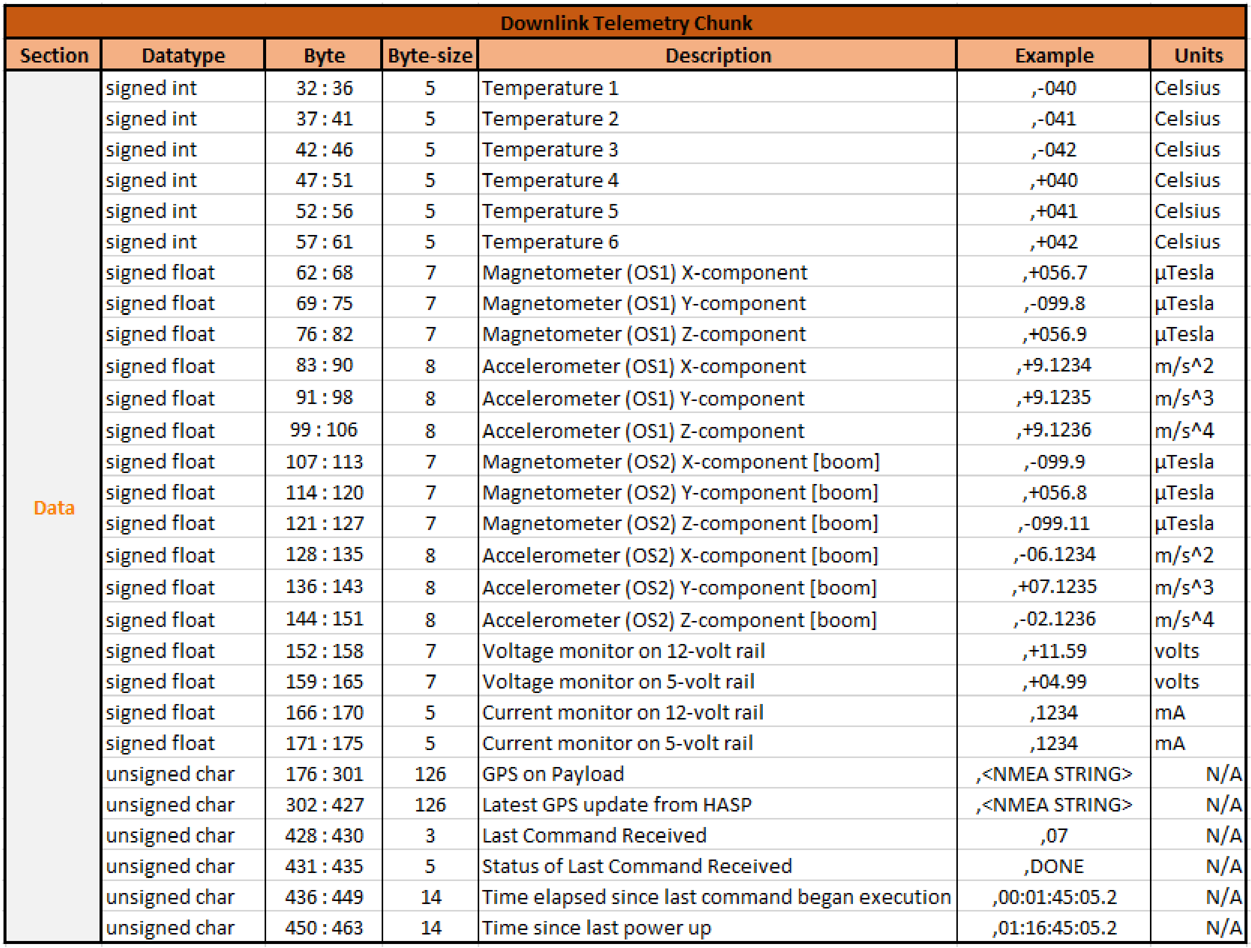


| Downlink Wrapper | | | | | | |
|---|---|---|---|---|---|---|
| **Section** | **Datatype** | **Byte** | **Byte-size** | **Description** | **Example** | **Units** |
| Header | unsigned char | 1 : 6 | 6 | Start of Transmission | 000000 | N/A |
| | unsigned char | 7 : 9 | 3 | COMPASS Payload ID | ,04 | N/A |
| | unsigned char | 10 : 13 | 4 | Record Type Indication | ,IMG | N/A |
| | unsigned int | 14 : 17 | 4 | Record Byte-Size | ,236 | Bytes |
| | unsigned char | 18 : 31 | 14 | Timestamp of packet transmission | ,01:16:45:05.2 | N/A |
| Data | | **32 : N** | N - 31 | Data : Telemetry or JPEG Image | | |
| Footer | unsigned int | (N + 1) : (N + 4) | 5 | Checksum | ,3512 | N/A |
| | unsigned char | (N + 5) : (N + 6) | 2 | End of Text | ,3 | N/A |

*Table 12: This table shows the preliminary Downlink Wrapper – all of the following data chunk are inserted into the Data section.*

**Telemetry Packet:**

- Packet length (with wrapper): 434 Bytes
- Transmission time (1200 baud): 3.3 Seconds
- Transmission frequency: 2x/minute

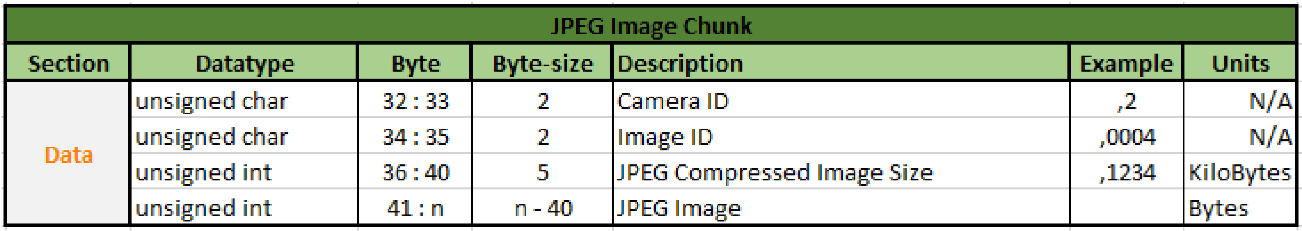


| Downlink Telemetry Chunk | | | | | | |
|---|---|---|---|---|---|---|
| Section | Datatype | Byte | Byte-size | Description | Example | Units |
| Data | signed int | 32 : 36 | 5 | Temperature 1 | ,-040 | Celsius |
| | signed int | 37 : 41 | 5 | Temperature 2 | ,-041 | Celsius |
| | signed int | 42 : 46 | 5 | Temperature 3 | ,-042 | Celsius |
| | signed int | 47 : 51 | 5 | Temperature 4 | ,+040 | Celsius |
| | signed int | 52 : 56 | 5 | Temperature 5 | ,+041 | Celsius |
| | signed int | 57 : 61 | 5 | Temperature 6 | ,+042 | Celsius |
| | signed float | 62 : 68 | 7 | Magnetometer (OS1) X-component | ,+056.7 | µTesla |
| | signed float | 69 : 75 | 7 | Magnetometer (OS1) Y-component | ,-099.8 | µTesla |
| | signed float | 76 : 82 | 7 | Magnetometer (OS1) Z-component | ,+056.9 | µTesla |
| | signed float | 83 : 90 | 8 | Accelerometer (OS1) X-component | ,+9.1234 | m/s^2 |
| | signed float | 91 : 98 | 8 | Accelerometer (OS1) Y-component | ,+9.1235 | m/s^3 |
| | signed float | 99 : 106 | 8 | Accelerometer (OS1) Z-component | ,+9.1236 | m/s^4 |
| | signed float | 107 : 113 | 7 | Magnetometer (OS2) X-component [boom] | ,-099.9 | µTesla |
| | signed float | 114 : 120 | 7 | Magnetometer (OS2) Y-component [boom] | ,+056.8 | µTesla |
| | signed float | 121 : 127 | 7 | Magnetometer (OS2) Z-component [boom] | ,-099.11 | µTesla |
| | signed float | 128 : 135 | 8 | Accelerometer (OS2) X-component [boom] | ,-06.1234 | m/s^2 |
| | signed float | 136 : 143 | 8 | Accelerometer (OS2) Y-component [boom] | ,+07.1235 | m/s^3 |
| | signed float | 144 : 151 | 8 | Accelerometer (OS2) Z-component [boom] | ,-02.1236 | m/s^4 |
| | signed float | 152 : 158 | 7 | Voltage monitor on 12-volt rail | ,+11.59 | volts |
| | signed float | 159 : 165 | 7 | Voltage monitor on 5-volt rail | ,+04.99 | volts |
| | signed float | 166 : 170 | 5 | Current monitor on 12-volt rail | ,1234 | mA |
| | signed float | 171 : 175 | 5 | Current monitor on 5-volt rail | ,1234 | mA |
| | unsigned char | 176 : 301 | 126 | GPS on Payload | ,<NMEA STRING> | N/A |
| | unsigned char | 302 : 427 | 126 | Latest GPS update from HASP | ,<NMEA STRING> | N/A |
| | unsigned char | 428 : 430 | 3 | Last Command Received | ,07 | N/A |
| | unsigned char | 431 : 435 | 5 | Status of Last Command Received | ,DONE | N/A |
| | unsigned char | 436 : 449 | 14 | Time elapsed since last command began execution | ,00:01:45:05.2 | N/A |
| | unsigned char | 450 : 463 | 14 | Time since last power up | ,01:16:45:05.2 | N/A |

*Table 13: This table shows the preliminary housekeeping/telemetry data chunk. This will be the default downlinked data packet.*

**JPEG Packet:**

- Assumed size of a single JPEG image (low-res telemetry JPEG approximately 50x50 pixels): 460 Bytes
- Expected packet length with wrapper: 507 Bytes
- Transmission time (1200 baud): 3.8 seconds
- Transmission Frequency: 1x/minute

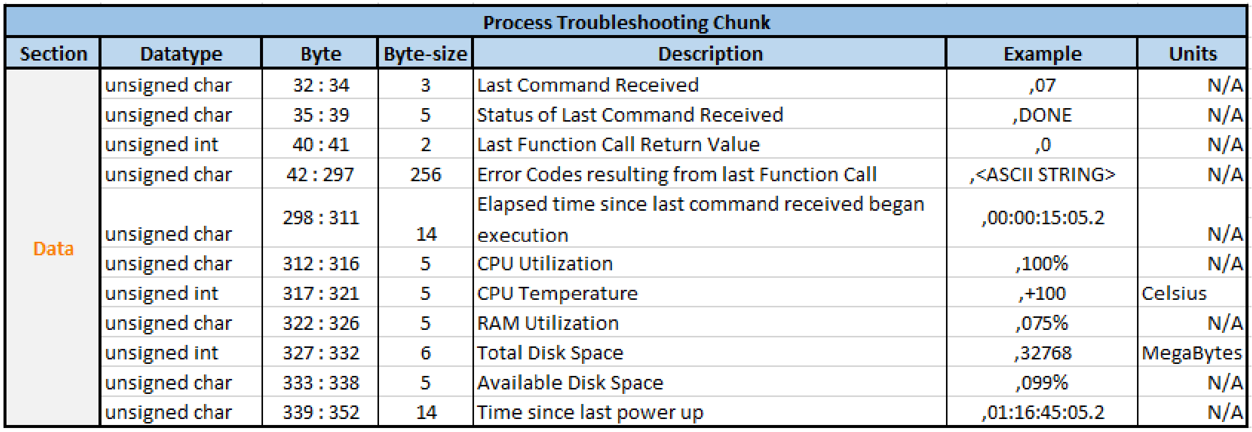


| JPEG Image Chunk | | | | | | |
|---|---|---|---|---|---|---|
| Section | Datatype | Byte | Byte-size | Description | Example | Units |
| Data | unsigned char | 32 : 33 | 2 | Camera ID | ,2 | N/A |
| | unsigned char | 34 : 35 | 2 | Image ID | ,0004 | N/A |
| | unsigned int | 36 : 40 | 5 | JPEG Compressed Image Size | ,1234 | KiloBytes |
| | unsigned int | 41 : n | n - 40 | JPEG Image | | Bytes |

*Table 14: This table shows the preliminary image data chunk. This will be send as requested by uplink commanding.*

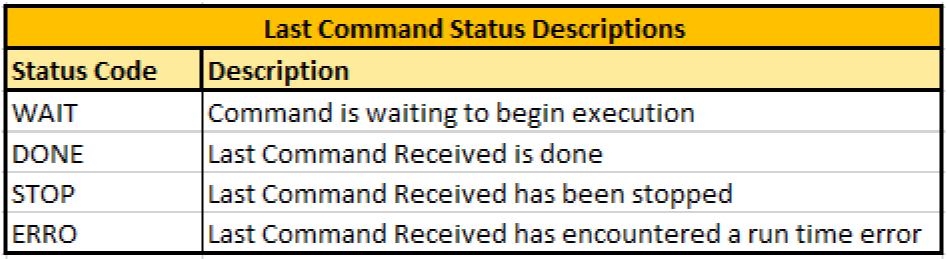
**Troubleshooting Packet:**

- Packet length with wrapper: 358 Bytes
- Transmission time (1200 baud): 2.7 seconds
- Transmission frequency: 1x/minute



| Process Troubleshooting Chunk | | | | | | |
|---|---|---|---|---|---|---|
| Section | Datatype | Byte | Byte-size | Description | Example | Units |
| Data | unsigned char | 32 : 34 | 3 | Last Command Received | ,07 | N/A |
| | unsigned char | 35 : 39 | 5 | Status of Last Command Received | ,DONE | N/A |
| | unsigned int | 40 : 41 | 2 | Last Function Call Return Value | ,0 | N/A |
| | unsigned char | 42 : 297 | 256 | Error Codes resulting from last Function Call | ,<ASCII STRING> | N/A |
| | unsigned char | 298 : 311 | 14 | Elapsed time since last command received began execution | ,00:00:15:05.2 | N/A |
| | unsigned char | 312 : 316 | 5 | CPU Utilization | ,100% | N/A |
| | unsigned int | 317 : 321 | 5 | CPU Temperature | ,+100 | Celsius |
| | unsigned char | 322 : 326 | 5 | RAM Utilization | ,075% | N/A |
| | unsigned int | 327 : 332 | 6 | Total Disk Space | ,32768 | MegaBytes |
| | unsigned char | 333 : 338 | 5 | Available Disk Space | ,099% | N/A |
| | unsigned char | 339 : 352 | 14 | Time since last power up | ,01:16:45:05.2 | N/A |

*Table 15: This table shows the preliminary process troubleshooting data chunk. This will be sent as requested by uplink commanding.*

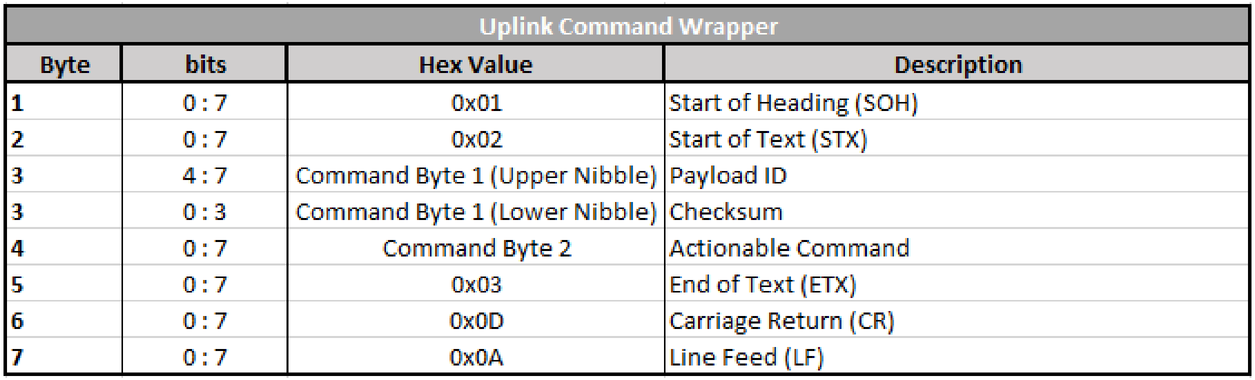


| Last Command Status Descriptions | |
|---|---|
| **Status Code** | **Description** |
| WAIT | Command is waiting to begin execution |
| DONE | Last Command Received is done |
| STOP | Last Command Received has been stopped |
| ERRO | Last Command Received has encountered a run time error |

*Table 16: This table describes each of the status codes associated with various states in the execution of the last command received via uplink.*

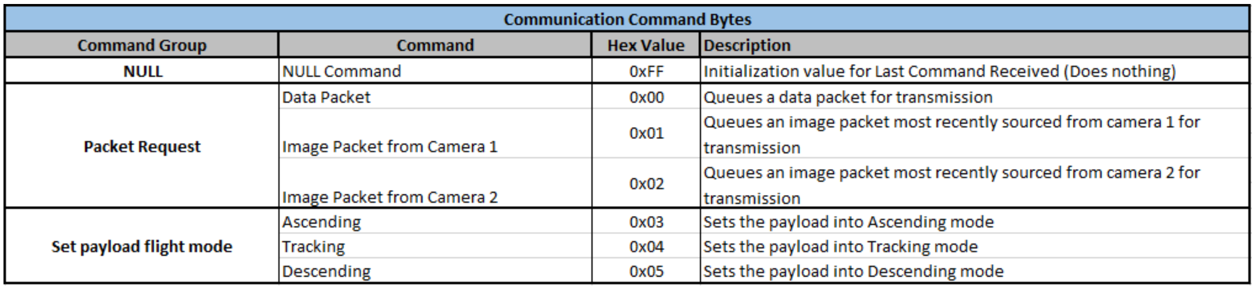
## 3.4 Uplink Serial Commanding

COMPASS intends to use the uplink command capabilities provided by HASP to monitor to payload as well as change the state of the payload if necessary. The proper packet wrapper and proposed commands are detailed in the tables below



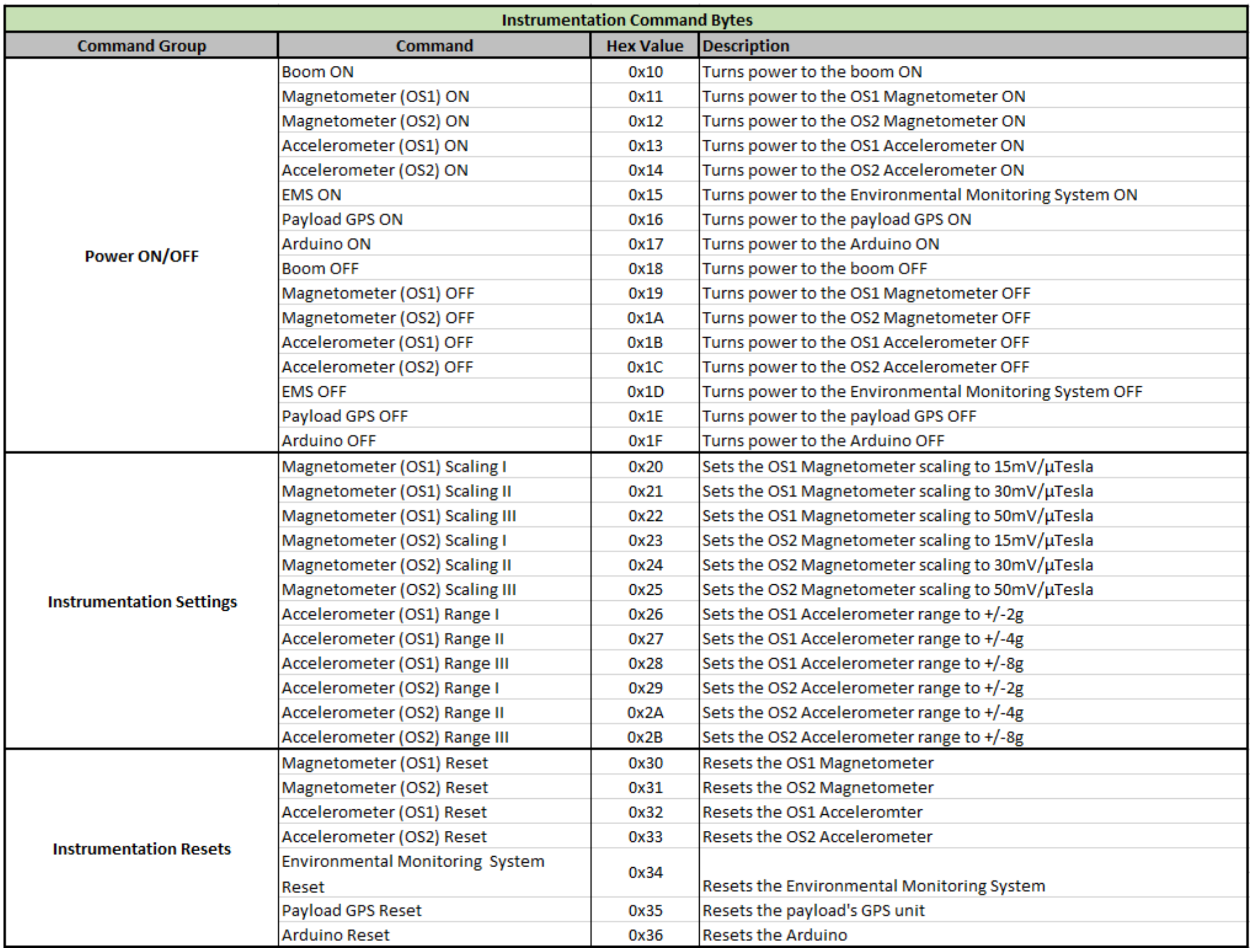
| Uplink Command Wrapper | | | |
|---|---|---|---|
| Byte | bits | Hex Value | Description |
| 1 | 0 : 7 | 0x01 | Start of Heading (SOH) |
| 2 | 0 : 7 | 0x02 | Start of Text (STX) |
| 3 | 4 : 7 | Command Byte 1 (Upper Nibble) | Payload ID |
| 3 | 0 : 3 | Command Byte 1 (Lower Nibble) | Checksum |
| 4 | 0 : 7 | Command Byte 2 | Actionable Command |
| 5 | 0 : 7 | 0x03 | End of Text (ETX) |
| 6 | 0 : 7 | 0x0D | Carriage Return (CR) |
| 7 | 0 : 7 | 0x0A | Line Feed (LF) |

*Table 17: HASP command wrapper. All command packets are Length: 7 Bytes; Transmission Time: 0.05 seconds' Expected frequency: Every 12-minutes*



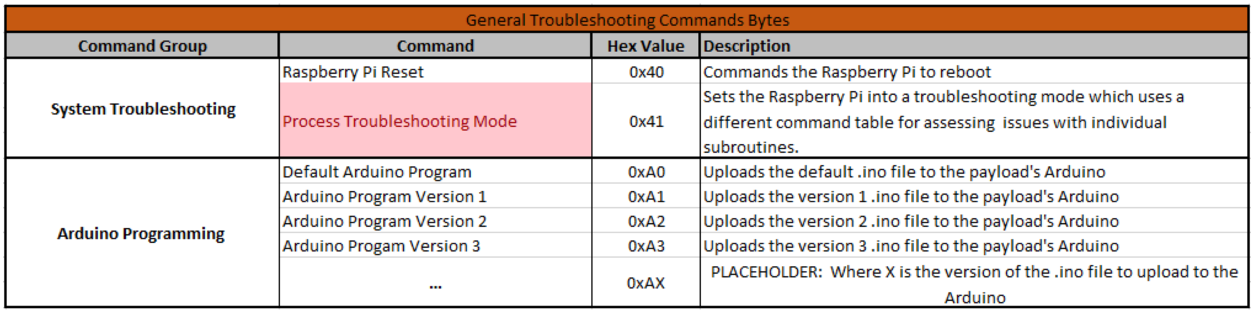
| Communication Command Bytes | | | |
|---|---|---|---|
| **Command Group** | **Command** | **Hex Value** | **Description** |
| **NULL** | NULL Command | 0xFF | Initialization value for Last Command Received (Does nothing) |
| **Packet Request** | Data Packet | 0x00 | Queues a data packet for transmission |
| | Image Packet from Camera 1 | 0x01 | Queues an image packet most recently sourced from camera 1 for transmission |
| | Image Packet from Camera 2 | 0x02 | Queues an image packet most recently sourced from camera 2 for transmission |
| **Set payload flight mode** | Ascending | 0x03 | Sets the payload into Ascending mode |
| | Tracking | 0x04 | Sets the payload into Tracking mode |
| | Descending | 0x05 | Sets the payload into Descending mode |

*Table 18: This table details the proposed communication commands for the payload during flight.*



| Instrumentation Command Bytes | | | |
|---|---|---|---|
| **Command Group** | **Command** | **Hex Value** | **Description** |
| **Power ON/OFF** | Boom ON | 0x10 | Turns power to the boom ON |
| | Magnetometer (OS1) ON | 0x11 | Turns power to the OS1 Magnetometer ON |
| | Magnetometer (OS2) ON | 0x12 | Turns power to the OS2 Magnetometer ON |
| | Accelerometer (OS1) ON | 0x13 | Turns power to the OS1 Accelerometer ON |
| | Accelerometer (OS2) ON | 0x14 | Turns power to the OS2 Accelerometer ON |
| | EMS ON | 0x15 | Turns power to the Environmental Monitoring System ON |
| | Payload GPS ON | 0x16 | Turns power to the payload GPS ON |
| | Arduino ON | 0x17 | Turns power to the Arduino ON |
| | Boom OFF | 0x18 | Turns power to the boom OFF |
| | Magnetometer (OS1) OFF | 0x19 | Turns power to the OS1 Magnetometer OFF |
| | Magnetometer (OS2) OFF | 0x1A | Turns power to the OS2 Magnetometer OFF |
| | Accelerometer (OS1) OFF | 0x1B | Turns power to the OS1 Accelerometer OFF |
| | Accelerometer (OS2) OFF | 0x1C | Turns power to the OS2 Accelerometer OFF |
| | EMS OFF | 0x1D | Turns power to the Environmental Monitoring System OFF |
| | Payload GPS OFF | 0x1E | Turns power to the payload GPS OFF |
| | Arduino OFF | 0x1F | Turns power to the Arduino OFF |
| **Instrumentation Settings** | Magnetometer (OS1) Scaling I | 0x20 | Sets the OS1 Magnetometer scaling to 15mV/µTesla |
| | Magnetometer (OS1) Scaling II | 0x21 | Sets the OS1 Magnetometer scaling to 30mV/µTesla |
| | Magnetometer (OS1) Scaling III | 0x22 | Sets the OS1 Magnetometer scaling to 50mV/µTesla |
| | Magnetometer (OS2) Scaling I | 0x23 | Sets the OS2 Magnetometer scaling to 15mV/µTesla |
| | Magnetometer (OS2) Scaling II | 0x24 | Sets the OS2 Magnetometer scaling to 30mV/µTesla |
| | Magnetometer (OS2) Scaling III | 0x25 | Sets the OS2 Magnetometer scaling to 50mV/µTesla |
| | Accelerometer (OS1) Range I | 0x26 | Sets the OS1 Accelerometer range to +/-2g |
| | Accelerometer (OS1) Range II | 0x27 | Sets the OS1 Accelerometer range to +/-4g |
| | Accelerometer (OS1) Range III | 0x28 | Sets the OS1 Accelerometer range to +/-8g |
| | Accelerometer (OS2) Range I | 0x29 | Sets the OS2 Accelerometer range to +/-2g |
| | Accelerometer (OS2) Range II | 0x2A | Sets the OS2 Accelerometer range to +/-4g |
| | Accelerometer (OS2) Range III | 0x2B | Sets the OS2 Accelerometer range to +/-8g |
| **Instrumentation Resets** | Magnetometer (OS1) Reset | 0x30 | Resets the OS1 Magnetometer |
| | Magnetometer (OS2) Reset | 0x31 | Resets the OS2 Magnetometer |
| | Accelerometer (OS1) Reset | 0x32 | Resets the OS1 Accelerومter |
| | Accelerometer (OS2) Reset | 0x33 | Resets the OS2 Accelerometer |
| | Environmental Monitoring System Reset | 0x34 | Resets the Environmental Monitoring System |
| | Payload GPS Reset | 0x35 | Resets the payload's GPS unit |
| | Arduino Reset | 0x36 | Resets the Arduino |

*Table 19: This table details the proposed instrumentation commands for the payload during flight. Instrument power, reset, and modes of operation.*



| General Troubleshooting Commands Bytes | | | |
|---|---|---|---|
| **Command Group** | **Command** | **Hex Value** | **Description** |
| **System Troubleshooting** | Raspberry Pi Reset | 0x40 | Commands the Raspberry Pi to reboot |
| | Process Troubleshooting Mode | 0x41 | Sets the Raspberry Pi into a troubleshooting mode which uses a different command table for assessing issues with individual subroutines. |
| **Arduino Programming** | Default Arduino Program | 0xA0 | Uploads the default .ino file to the payload's Arduino |
| | Arduino Program Version 1 | 0xA1 | Uploads the version 1 .ino file to the payload's Arduino |
| | Arduino Program Version 2 | 0xA2 | Uploads the version 2 .ino file to the payload's Arduino |
| | Arduino Progam Version 3 | 0xA3 | Uploads the version 3 .ino file to the payload's Arduino |
| | ... | 0xAX | PLACEHOLDER: Where X is the version of the .ino file to upload to the Arduino |

*Table 20: This table details the proposed general troubleshooting commands for the payload during flight.*

While we are not planning to use different versions of Arduino code during the float; the ability to do so might be useful in troubleshooting systems that are handled and interpreted by the Arduino. The Raspberry Pi will be able to upload pre-defined versions of code which focus on particular aspects of the Arduino's functions. Since there are still decisions to be made on what the Arduino will be doing, these version commands are temporary placeholders and will be replaced/expanded with more descriptive names later on.

Additionally, troubleshooting options for the subroutines running on the Raspberry Pi itself will be available. From the general troubleshooting commands found in Table 19, the "Process Troubleshooting Mode" can be accessed via command "0x41".  In this mode, the Process Troubleshooting data chunks will be the default transmission and the payload will interpret uplink commands as outlined below in table 20: *Process Troubleshooting Commands*. These commands are used to manually execute individual subroutines in order to determine which process(es) may be causing faulty behavior. If a subroutine is determined to be a source of faulty behavior, a new sequence of function calls can be created without the erroneous subroutine, tested, and saved – and the primary commanding mode can resume.

**Command Sequence Flowchart**

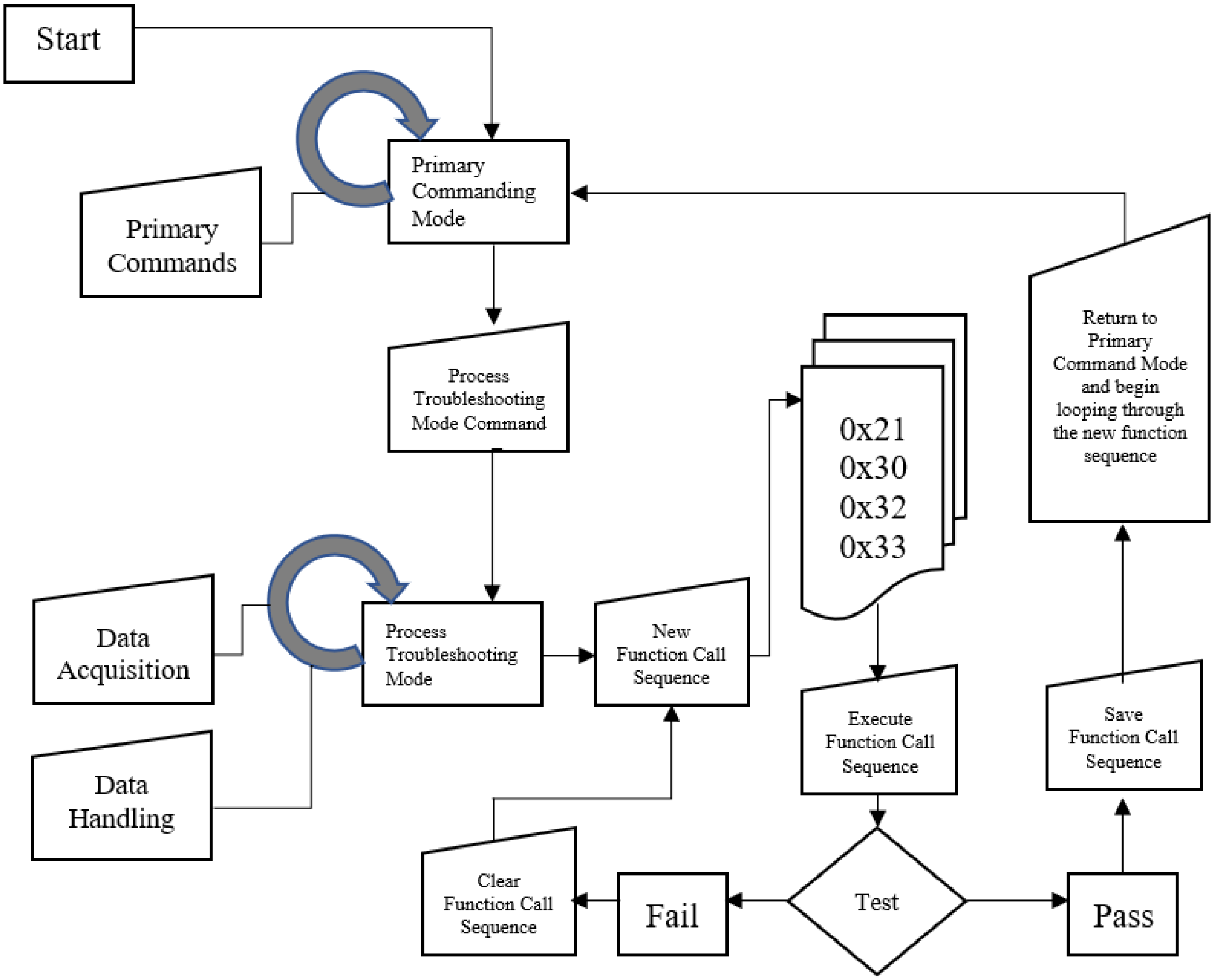
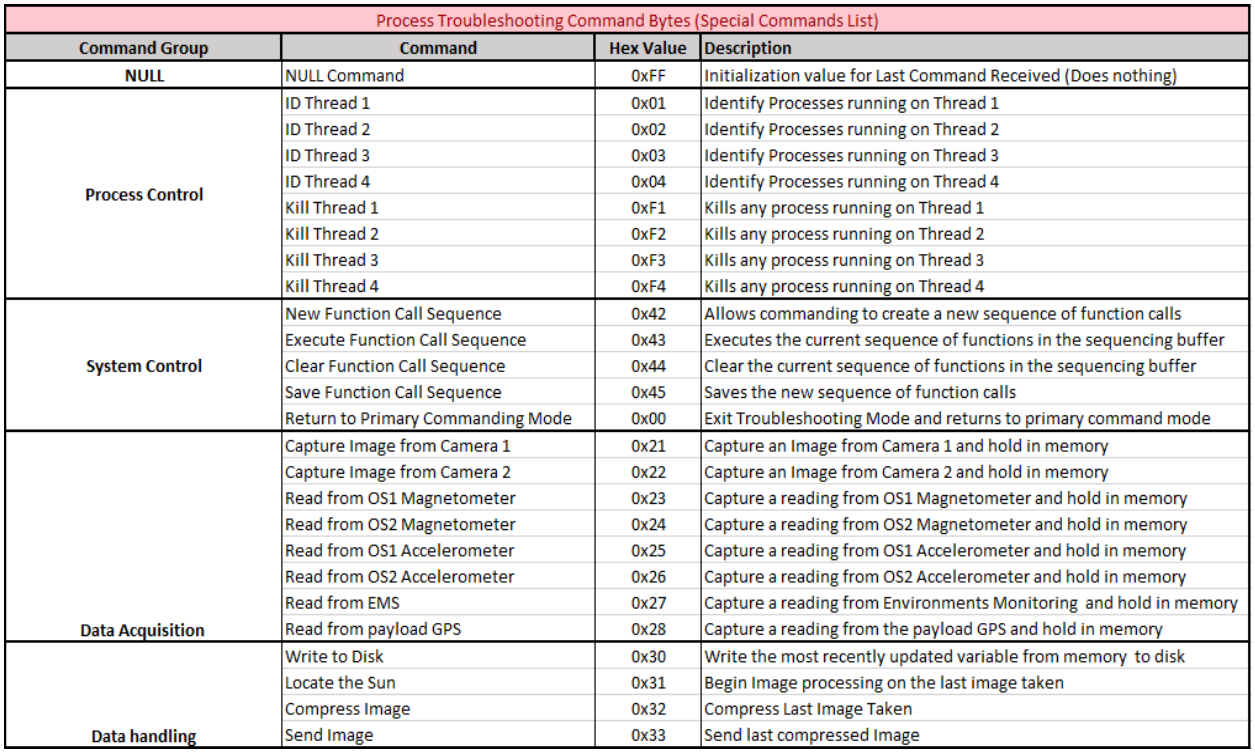

Start
Primary Commanding Mode
Primary Commands
Process Troubleshooting Mode Command
0x21
0x30
0x32
0x33
Return to Primary Command Mode and begin looping through the new function sequence
Data Acquisition
Process Troubleshooting Mode
New Function Call Sequence
Execute Function Call Sequence
Save Function Call Sequence
Data Handling
Clear Function Call Sequence
Fail
Test
Pass

*Figure 25: This flowchart illustrates the process by which a new command sequence is written and control is handed back to primary commanding*



| Process Troubleshooting Command Bytes (Special Commands List) | | | |
|---|---|---|---|
| Command Group | Command | Hex Value | Description |
| NULL | NULL Command | 0xFF | Initialization value for Last Command Received (Does nothing) |
| Process Control | ID Thread 1 | 0x01 | Identify Processes running on Thread 1 |
| | ID Thread 2 | 0x02 | Identify Processes running on Thread 2 |
| | ID Thread 3 | 0x03 | Identify Processes running on Thread 3 |
| | ID Thread 4 | 0x04 | Identify Processes running on Thread 4 |
| | Kill Thread 1 | 0xF1 | Kills any process running on Thread 1 |
| | Kill Thread 2 | 0xF2 | Kills any process running on Thread 2 |
| | Kill Thread 3 | 0xF3 | Kills any process running on Thread 3 |
| | Kill Thread 4 | 0xF4 | Kills any process running on Thread 4 |
| System Control | New Function Call Sequence | 0x42 | Allows commanding to create a new sequence of function calls |
| | Execute Function Call Sequence | 0x43 | Executes the current sequence of functions in the sequencing buffer |
| | Clear Function Call Sequence | 0x44 | Clear the current sequence of functions in the sequencing buffer |
| | Save Function Call Sequence | 0x45 | Saves the new sequence of function calls |
| | Return to Primary Commanding Mode | 0x00 | Exit Troubleshooting Mode and returns to primary command mode |
| Data Acquisition | Capture Image from Camera 1 | 0x21 | Capture an Image from Camera 1 and hold in memory |
| | Capture Image from Camera 2 | 0x22 | Capture an Image from Camera 2 and hold in memory |
| | Read from OS1 Magnetometer | 0x23 | Capture a reading from OS1 Magnetometer and hold in memory |
| | Read from OS2 Magnetometer | 0x24 | Capture a reading from OS2 Magnetometer and hold in memory |
| | Read from OS1 Accelerometer | 0x25 | Capture a reading from OS1 Accelerometer and hold in memory |
| | Read from OS2 Accelerometer | 0x26 | Capture a reading from OS2 Accelerometer and hold in memory |
| | Read from EMS | 0x27 | Capture a reading from Environments Monitoring and hold in memory |
| | Read from payload GPS | 0x28 | Capture a reading from the payload GPS and hold in memory |
| Data handling | Write to Disk | 0x30 | Write the most recently updated variable from memory to disk |
| | Locate the Sun | 0x31 | Begin Image processing on the last image taken |
| | Compress Image | 0x32 | Compress Last Image Taken |
| | Send Image | 0x33 | Send last compressed Image |

*Table 21: This table details the proposed process troubleshooting commands for the payload during flight.*

## 3.5 Analog Downlink

The analog downlink capability on the EDAC will be utilized to read (tentatively) the currents of the 5-volt and 12-volt power regulators. This is a very important step to monitor because the rest of the payload function hinges on correctly stepping down the power to be supplied to the rest of the payload. The other option for the analog downlink capabilities of the payload we are considering is monitoring currents in the serial connection to determine whether our payload in receiving commands or transmitting data to and from HASP.

## 3.6 Discrete Commanding

COMPASS will not require the ability to utilize extra discrete commanding.

## 3.7 Payload Location and Orientation Request

Our top two preferences for the location of COMPASS would be at payload positions 3 or 4. This is mostly because we want the boom to be the furthest it can from the launch vehicle to avoid any possible interference. The measurements given in section 1.4 assume placement in payload position 3, but placement in payload position 4 will also work.

**HASP Payload Seats**

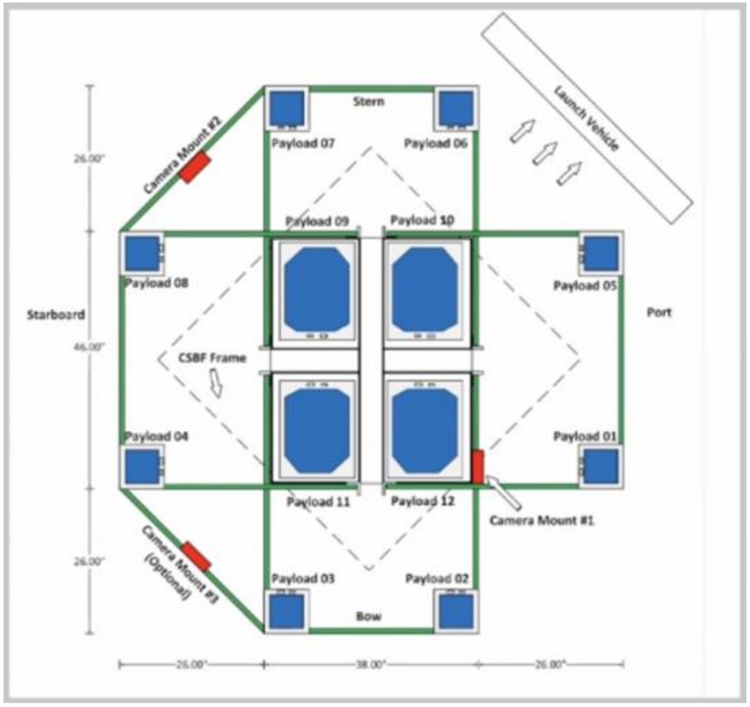


*Figure 26: HASP payload seats diagram showing the launch vehicle and the relative positions of the payload seats*

**Compass Requested Location**

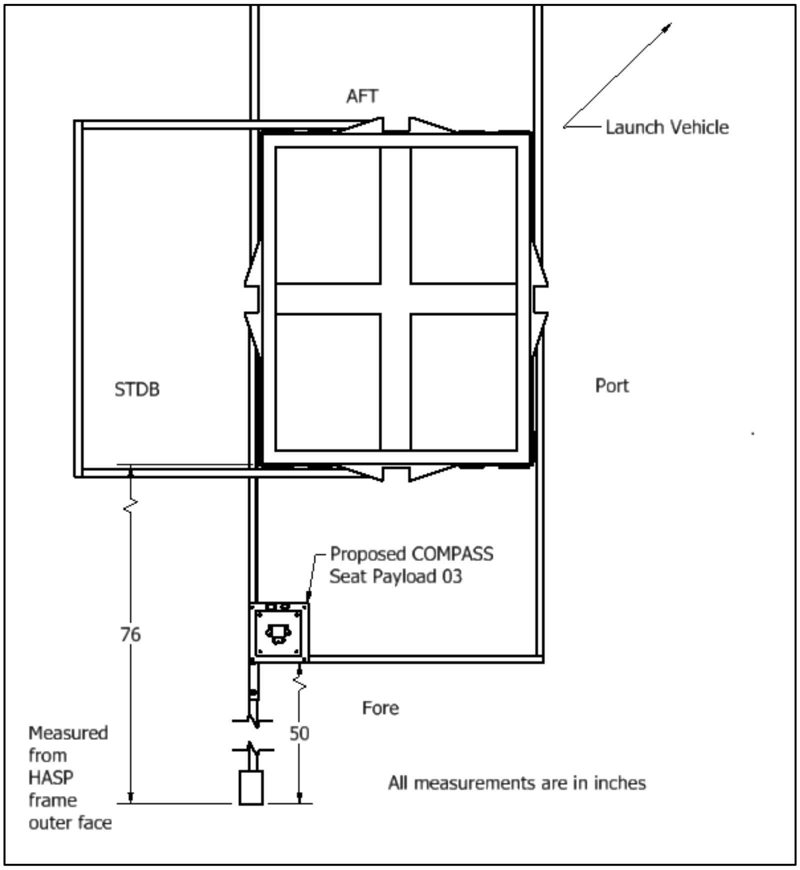


*Figure 27: Diagram of COMPASS payload and the boom in its requested location on HASP. Note that payload position 4 is also an acceptable position for COMPASS.*

## 3.8 Special Requests

It is critical for the COMPASS mission to ensure that the reading of the magnetometer is not affected by electronic interference. Without accurate data from the magnetometer, the orientation of the payload cannot be properly determined, as the azimuth angle would not be accurate, thus preventing science requirement 1.1 from being reached. This requirement must be met for the mission to be successful.

In order to ensure that the magnetometer reading is accurate, the error due to electrical interference cannot exceed the error in the magnetometer. For this reason, the COMPASS mission requests a weight waiver in order to attach a fiberglass boom to the HASP mechanical interface. A magnetometer and an Arduino Mega would be placed on the boom (as discussed ins section 1.3), away from the rest of the payload in order to avoid electrical interference. These parts would be contained in a plastic enclosure on the end of the boom estimated to have dimensions 156x122x52(mm). The total weight of the boom and its components are given below, and a stress calculation indicating that the boom can hold this payload is given in section 1.4.2.

In table 21 below the masses of the non-structural components of the boom are given, including the electronics housing box, the wiring necessary to communicate with the boom, and the electronics that will be operating on the end of the boom.

**Boom Payload Weight Budget**

| **Component** | **Mass per Unit (g)** | **Quantity** | **Total Mass (g)** | **Unc.** | **Measurement Method** |
|---|---|---|---|---|---|
| Wiring | - | - | 30 | ±10 | Estimate |
| Arduino Mega | 37 | 1 | 37 | ±1 | Estimate |
| Accelerometer Shield | 33 | 1 | 33 | ±1 | Estimate |
| Magnetometer | 120 | 1 | 120 | ±1 | Estimate |
| Zip ties (securing wiring) | 0.3 | 10 ± 5 | 3 | ±1.5 | Calculation |
| Electronics Housing Box | 340 | 1 | 340 | ±50 | Estimate |
| Environmental Sensor | 15 | 1 | 15 | ±5 | Estimate |
| **Total Non-Structural** | | | **578** | **±51** | |

*Table 22: Preliminary weight budget for the non-structural components of the boom*

Below is a table of the structural components of the boom, as discussed in section 1.4.2.

**Boom Structure Weight Budget**

| **Component** | **Mass per Unit (g)** | **Quantity** | **Total Mass (g)** | **Unc.** | **Measurement Method** |
|---|---|---|---|---|---|
| Fasteners | 23.5 | 4 | 94 | ±5 | Calculation |
| Steel Sleeve (to connect segments) | 416 | 1 | 416 | ±21 | Calculation |
| Fiberglass Segment (48in) | 666 | 2 | 1333 | ±67 | Calculation |
| **Total Structural** | | | **1843** | **±70** | |

*Table 23: Preliminary weight budget for the structural components of the boom.*

Table 23 displays the total estimated weight budget of the boom, using the results from table 21 and table 22. A weight waiver is requested in order to add this additional weight to the COMPASS payload.

**Total Weight Estimate for Boom**

| **Section** | **Mass (g)** | **Unc. (g)** |
|---|---|---|
| Non-Structural | 563 | ±51 |
| Structural | 1843 | ±70 |
| **Total** | **2406** | **±87** |

*Table 24: Total estimated weight budget for the boom and the components on it.*

## 4. Preliminary Drawings and Diagrams

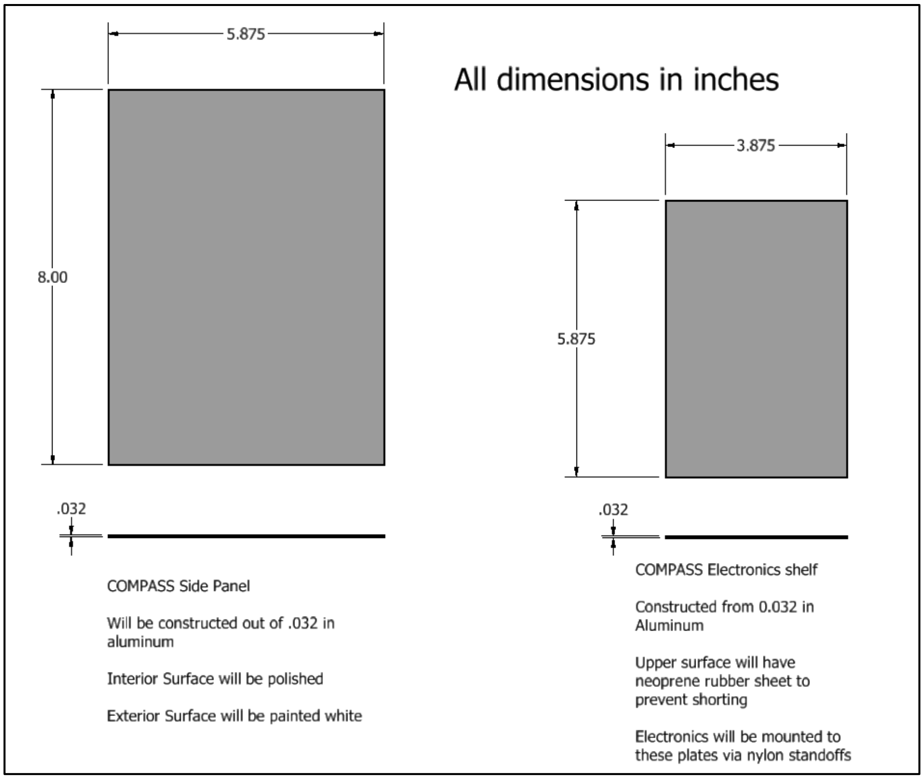

All dimensions in inches
5.875
8.00
.032
COMPASS Side Panel
Will be constructed out of .032 in aluminum
Interior Surface will be polished
Exterior Surface will be painted white
3.875
5.875
.032
COMPASS Electronics shelf
Constructed from 0.032 in Aluminum
Upper surface will have neoprene rubber sheet to prevent shorting
Electronics will be mounted to these plates via nylon standoffs

*Figure 28: COMPASS side panels*

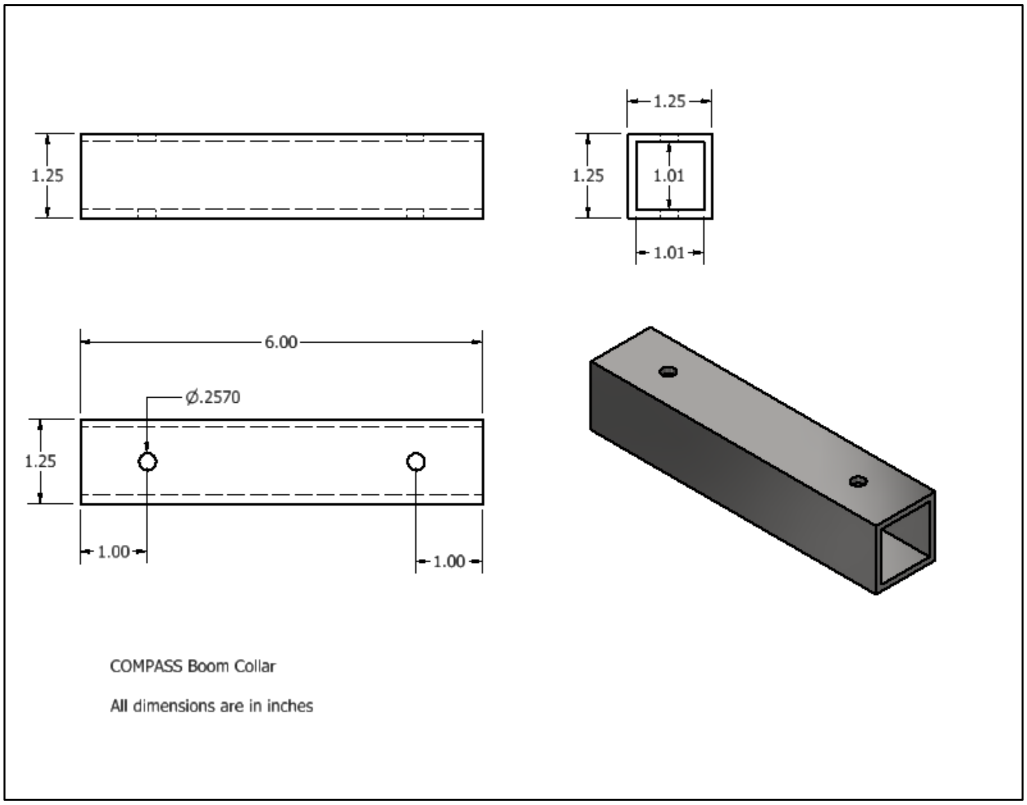


*Figure 29: COMPASS boom collars*

# 5. References

1. ‘Angle of View’ (2020) *Wikipedia.* Available at https://en.wikipedia.org/wiki/Angle_of_view
2. ‘Azimuth’ (2020) *Wikipedia.* Available at https://en.wikipedia.org/wiki/Angle_of_view
3. Angle/Path of the Sun calculator application, ‘Suncalc,’ Available at https://www.suncalc.org/
4. Adafruit, “LSM303 Accelerometer + Compass Breakout,” Last updated Jan 8, 2020, Available at https://cdn-learn.adafruit.com/downloads/pdf/lsm303-accelerometer-slash-compass-breakout.pdf?timestamp=1608269052
5. STMicroelectronics, “Ultra-compact high-performance eCompass module: ultra-low-power 3D accelerometer and 3D magnetometer,” November 2018, Available at https://www.st.com/resource/en/datasheet/lsm303agr.pdf
6. NXP, “Sensor Toolbox Development Boards for a 9-Axis Solution using MMA8652FC, FXAS21002C and MAG3110,” October 2015, Available at https://www.nxp.com/design/software/development-software/sensor-toolbox-sensor-development-ecosystem/evaluation-boards/sensor-toolbox-development-boards-for-a-9-axis-solution-using-mma8652fc-fxas21002c-and-mag3110:AGM04-SOLUTION
7. OKW Enclosures, “C7012011,” Available at https://www.okwenclosures.com/en/In-Box/C7012011.htm
8. McMaster-Carr, “Structural FRP Fiberglass Square Tube,” 2020, Available at https://www.mcmaster.com/8548K31/
9. LaACESProgram (2019), “HASP 2019 Downward Looking Camera #2 Stream Part 2,” Available at https://www.youtube.com/watch?v=Ler8qjh9bHw
10. Arducam, “Camarray – Arducam 1MP Stereoscopic Camera Bundle Kit,” December 7, 2020, Available at https://www.arducam.com/docs/cameras-for-raspberry-pi/synchronized-stereo-camera-hat/camarry-1mp2-stereoscopic-camera-hat/
11. Arducam, “Arducam 1/4″ M12 Mount 1.6mm Focal Length Camera Lens M40160M12,” March 20, 2019, Available at https://www.arducam.com/product/u105001-4-m12-mount-1-6mm-focal-length-camera-lens-ls-0002-for-raspberry-pi/
12. Arducam, “Introduction to Arducam USB Camera Shields,” August 13, 2020, Available at https://www.arducam.com/docs/usb-cameras/introduction-to-arducam-usb-camera-shields/
13. Arducam, “Arducam 0.36MP MT9V022 Monochrome 1/3” CMOS Camera Module,” March 22, 2019, Available at https://www.arducam.com/product/arducam-mt9v022-monochrome-1-3-inch-cmos-0-36mp-camera-module/
14. Arducam, “Arducam USB3.0 Camera Shield,” January 30, 3019, Available at https://www.arducam.com/product/usb3-0-camera-shield-2/
15. Uctronics, “Stereo Camera Adapter Board,” Available at https://www.uctronics.com/stereo-camera-adapter-board.html
16. Arduam, “Arducam 1/3″ M12 mount 2.25mm Focal Length Camera Lens M30225H10,” March 20, 2019, Available at https://www.arducam.com/product/m30225h10-m12-mount-2-25mm-focal-length-camera-lens-ls-30188-for-raspberry-pi/
17. Pultex, “The New and Improved Pultex Pultrusion Design Manual,” 2004, Available at https://www.creativepultrusions.com/default/assets/File/DMV5R11.pdf
18. Raspberry Pi, “FAQs,” Available at https://www.raspberrypi.org/documentation/faqs/#pi-power-gpioout
19. Freescale Semiconductor Inc, “Xtrinsic MMA8451Q 3-Axis, 14-bit/8-bit Digital Accelerometer,” 2013, Available at https://cdn-.adafruit.com/datasheets/MMA8451Q-1.pdf

20. Trimble, “Copernicus II GPS Receiver Reference Manual,” 2009, Available at http://cdn.sparkfun.com/datasheets/Sensors/GPS/63530-10_Rev-B_Manual_Copernicus-II.pdf
21. LECO Plastics, “Zip Ties- Cable Ties,” Available at https://lecoplastics.com/cable-ties.aspx
22. Arducam, “1MP OV9281 1/4” CMOS Global Shutter Standalone Camera UC9281M1 MIPI Interface,” Available at https://www.arducam.com/product/1-4-cmos-ov9281-global-shutter-standalone-camera-uc9281m1-mipi-interface/
23. Engineers Edge, “AWG Copper Wire Size and Data Table Chart @ 100 Degrees F,” Available at https://www.engineersedge.com/copper_wire.htm
24. McMaster-Carr, “Black-Oxide Alloy Steel Socket Head Screw,” Available at https://www.mcmaster.com/90044A125/
25. Microchip, “2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface,” 2008, Available at https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf
26. RobotShop, “Arduino Mega 2560 Datasheet,” Available at http://eprints.polsri.ac.id/4598/8/File%20VIII%20%28Lampiran%29.pdf
27. Murata Power Solutions, “78xxSR Series- 3.3V/5V/12V Outputs High-Efficiency Switching Regulators with LM78xx Pinouts,” 2015, Available at https://www.murata.com/products/productdata/8807028260894/dms-78xxsr.pdf?1583754811000
28. Bartington, “Mag648 & Mag649® Low Power Three-Axis Magnetic Field Sensors,” Available at https://www.bartington.com/wp content/uploads/pdfs/datasheets/Mag648_649_DS2298.pdf
29. Millan, R. et al, “The Balloon Array for RBSP Realistic Electron Losses (BARREL),” Space Science Reviews, DOI 10.1007/s11214-013-9971-z, available at http://www.dartmouth.edu/~barrel/documents/10.1007_s11214-013-9971-z.pdf
30. Raspberry Pi Trading Ltd, “Raspberry Pi 4 Computer- Model B,” 2020, Available at https://static.raspberrypi.org/files/product-briefs/200521+Raspberry+Pi+4+Product+Brief.pdf
31. Sparkfun, “RS-232 vs TTL Serial Communication,” 2010, Available at https://www.sparkfun.com/tutorials/215
32. Adafruit, “Raspberry Pi Analog to Digital Converters,” available at https://www.digikey.com/en/maker/projects/raspberry-pi-analog-to-digital-converters/72388f5f1a0843418130f56c53a1276c
33. Adafruit, “Analog Inputs for Raspberry Pi Using the MCP3008,” Available at https://learn.adafruit.com/reading-a-analog-in-and-controlling-audio-volume-with-the-raspberry-pi?view=all
34. Raspberry Pi, “Raspberry Pi Temperature Monitoring,” 2020, available at https://www.raspberrypi-spy.co.uk/2020/11/raspberry-pi-temperature-monitoring/
35. CUI INC, “P78-2000-S Non-Isolated Switching Regulator,” Feb 21, 2020, Available at https://www.mouser.com/datasheet/2/670/p78_2000_s-1729053.pdf
36. Cooperhill Technologies, “Raspberry Pi Proven to Withstand Extended Temperature Range,” 2016, Available at https://copperhilltech.com/blog/raspberry-pi-proven-to-withstand-extended-temperature-range/

# Appendix

None