Selective Pointing Apparatus for Research of Turbulence and Atmospheric Noise Variation Final Report

Bryan Barnhart, Brian Ibeling, Christopher Nie, and Sushia Rahimizadeh
University of Colorado at Boulder, Boulder, Colorado, 80309
Colorado Space Grant Consortium - Boulder

December 16th, 2011

Payload Class: Large Flight Number: 2011 Payload ID: 10

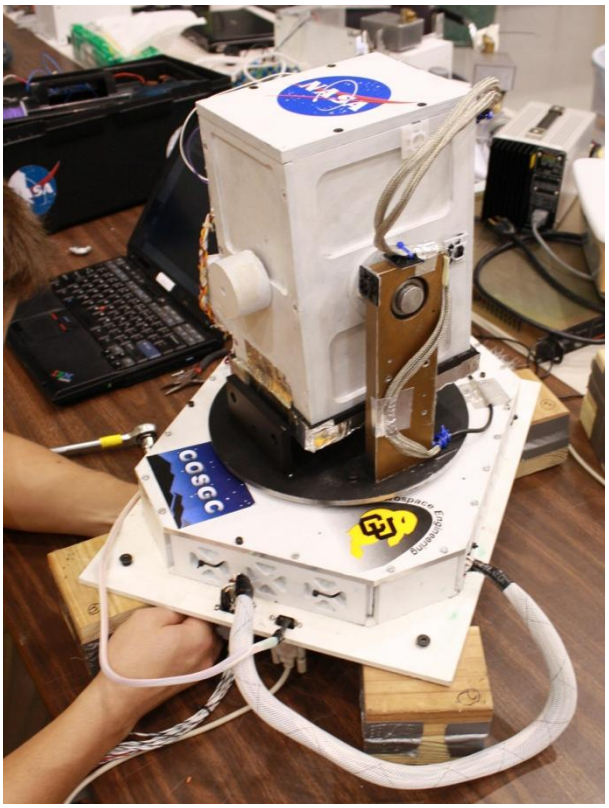

Table of Contents

I. Mission Overview

On March 7th, 2009, the Kepler Spacecraft was launched for a 3.5 year mission in the hopes of discovering habitable Earth-sized planets in surveying a portion of the Milky Way galaxy. Kepler will hold the same field of view for its entire flight duration and continually observe the photometric output of its collection of stars. A planet orbiting that star will demonstrate a periodic dip in a star's brightness as that planet passes between Kepler's observations and the output of the star, which is also known as the planetary transit detection method. The SPARTAN-V payload is entertaining that the same capabilities of the Kepler mission can be obtained with greater ease and a fraction of the cost with lighter than air balloon based observatories. Floating above 99% of Earth's distorting atmosphere, the near space environment may be a perfect viewpoint for capturing clear imaging of the cosmos.

The strongest motivation for this project has been the recent Kepler discovery of an exoplanet to be orbiting a star similar to the Sun, and residing within that star's habitable zone. The high cost of Kepler's place in orbit and the risk of failure from inaccessibility of repairs make balloon based platforms a highly feasible, cost efficient alternative. With its low cost, more than one "Kepler" can be placed at the edge of earth's atmosphere upon several balloons to view as many stars as possible. With this, the possibility of an army of balloon payloads to view the edges of space is not far from the future.

A. Planetary Transits and Earth's Atmospheric Light Scattering Effects

SPARTAN-V's primary mission is to determine the feasibility of detecting planetary transits from a high altitude balloon platform at approximately 120,000 feet (36km). This will be focusing on the level of interference from the remaining 1% of Earth's atmosphere at altitude, and if that interference is minimal enough to distinguish a planetary transit. Signal to Noise calculations of the images detail to what magnitude this distortion has on an optical system at this altitude.

B. Balloon Environment Characterization

SPARTAN-V's secondary mission was to help in characterizing the HASP balloon environment with the use of a 3-axis gyroscope, a 3-axis accelerometer, and a number of temperature sensors internal and external to the SPARTAN-V structure. All these sensors were sampled at a rate of 1 Hz for launch and duration of flight.

The balloon's spin rate during flight proved to be faster than anticipated, and a number of images had stars which were streaking across a number of pixels. Although this was not the data anticipated, it does provide another method to calculate the spin rate of the HASP balloon in the yaw rotation. Comparisons were also made between gyro data and image calculated spin rates, yielding promising results.

Figure 1, below, shows the anticipated flight concept of operations for the SPARTAN-V payload. After approximately four hours, float altitude is reached by the HASP platform, wherein the SPARTAN-V payload is configured such that the telescope locked (facing down to protect from the sun) and the electrical system running and collecting platform characterization data. After astronomical twilight, the telescope is unlocked and set into position to begin collecting image data. Sensor data collection is continued throughout the night. Upon flight termination, the telescope is locked and the system to powered down for HASP descent.

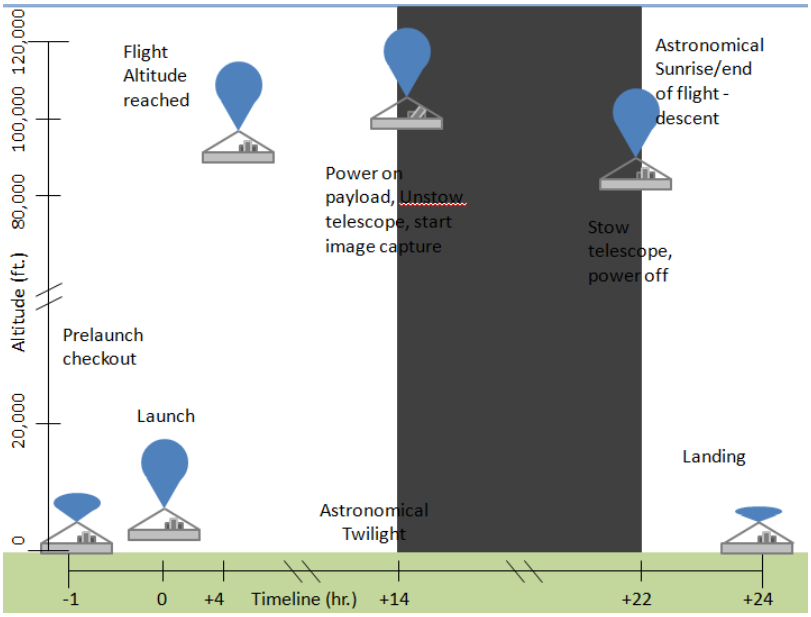

Figure 1. Flight Concept of Operations (Launch to Termination).

II. Mission Background and Theory

NASA's Kepler spacecraft was launched on March 7th of 2009, for a 3.5 year mission to survey and discover hundreds of Earth-size and smaller planets within our region of the Milky Way. Kepler will view the same field of view (10 degrees squared), shown in Fig. 2, for the duration of its flight, monitoring approximately 100,000 stars to measure periodic variances in their brightness. A periodic decrease in a star's brightness would be indicative of a planet orbiting that star. This planetary detection method is also known as a planetary transit, which is shown in Fig. 3 below.

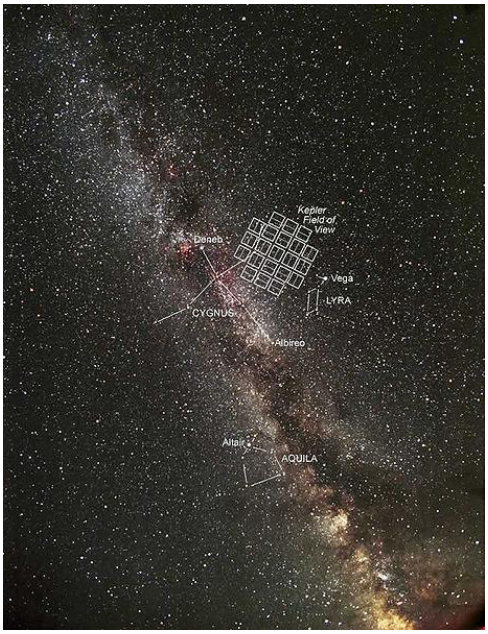

Figure 2. Kepler's Field of View of the Milky Way Galaxy.

As of December of 2011, Kepler has announced 30 confirmed planetary discoveries[1]. Being that these planets have short orbital periods, a max of 4.9 days, Kepler is able to verify that the periodic dip in brightness is due to a transiting planet. Even with 2,300 planetary candidates found in just two years, the majority of discoveries are not expected to be found until Kepler's 3rd year in flight.

For a 3.5 year flight, Kepler is estimated to cost $600 million. It has also been speculated that Kepler could be extended to a 4 year or even a 6 year flight, causing the $600 million cost to increase. The alternative use of a Balloon-Based Observatory (BBO) to view and measure the photometric output of stars could achieve Kepler's mission for a fraction of the cost. By comparison it costs around $600 per kilogram to launch a payload into the upper-stratosphere upon a lighter-than-air vehicle, while prices soar upwards of $20,000 to launch a single kilogram into Low Earth Orbit (LEO) on a rocket. BBOs ascend to an altitude of 120,000 feet, above 99% of Earth's atmosphere. At such an altitude, there is little to no influence from the weather, making the flight environment highly stable for optical devices. Also, since balloon vehicles are significantly cheaper, it is possible to station more than one in Earth's upper atmosphere. BBOs are also more easily accessible for maintenance or adjustments. Balloon Based Observatories may be the perfect alternative for discovering more of the universe we still know so little about.

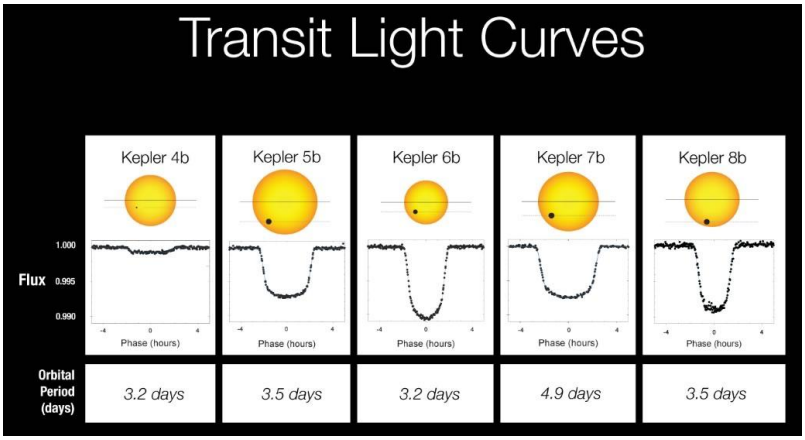

Figure 3. Transit light curves from 5 Kepler exoplanets discovered.

III. Engineering Application

In pursuit of fulfilling the science objectives of the project, a physical unit was engineered. The following sections provide a background of the payload that was built by the Colorado team during the course of the HASP 2010 lifetime.

A. Mechanical design

Based on the HASP platform requirements, a payload was design to fit within a 15'' x 12'' x 17'' volume envelope with a mass less than 20 kg. The payload consists of three main parts: the electrical casing, rotary table assembly, and telescope. The electrical casing serves as mounting for the majority of electronics including motherboard and memory storage devices. The rotary assembly is positioned at the top of the electrical housing and includes two pitch arms that support the telescope structure. The telescope consists of a custom built folding refractor telescope that allows for photometric data to be taken on flight. Figure 4, below, shows a comparison between the design and fabricated structure.

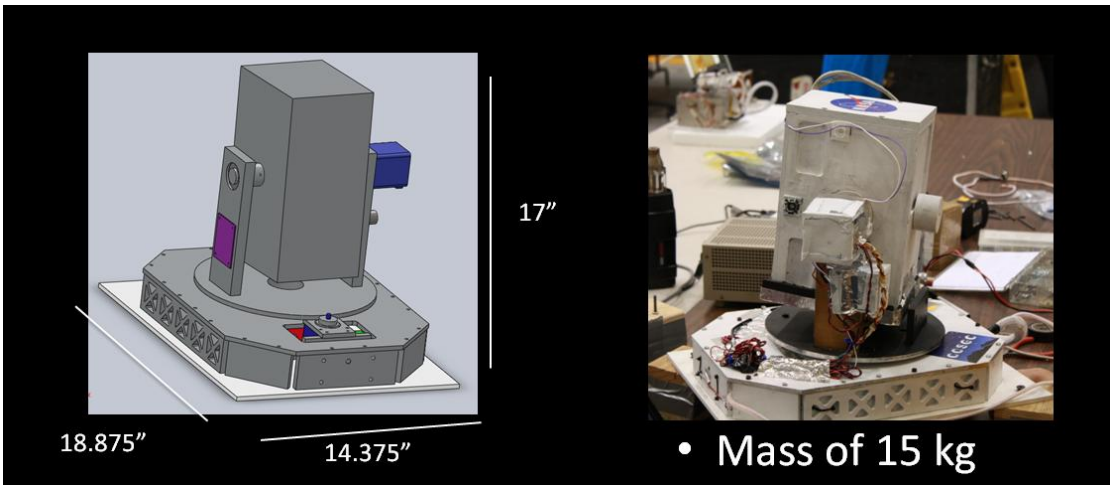

Figure 3. SPARTAN-V design and fabrication comparison.

The electrical casing was constructed with a primary skeleton of honey-combed 6061-T6 aluminum alloy for maximum strength, as well as insulated with foam core in order to maintain as stable a thermal environment as possible, and protect from solar radiation. The rotary table was made to support both the telescope's weight as well as its movement. It consists of a base plate that mates to two vertically parallel precision bars that house the bearing on which the telescope rotates. The telescope itself was manufactured separately to house the optical system, explained in the following section. Figure 5 displays the components of the design and flight structure.

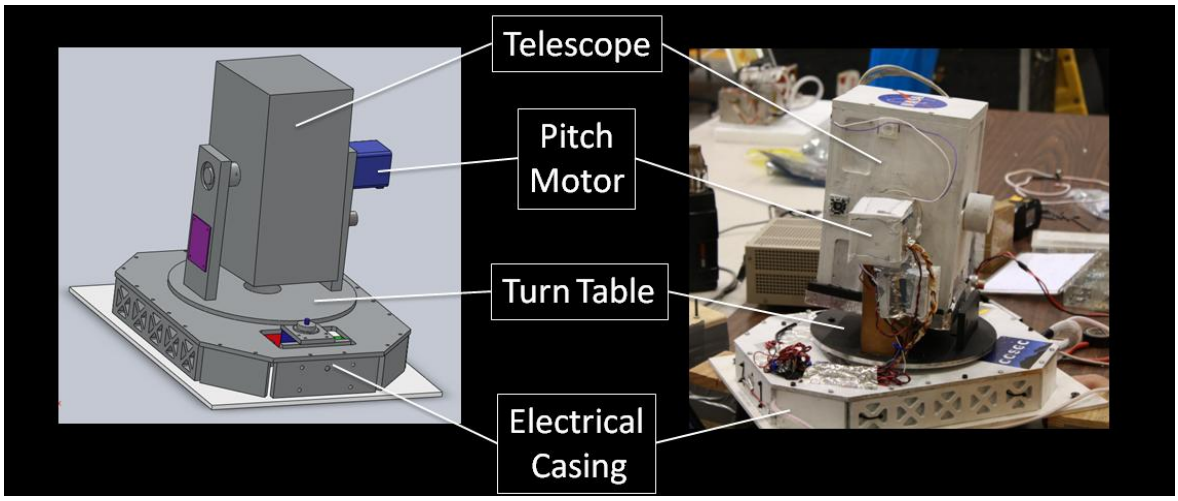

Figure 4. SPARTAN-V design and fabrication comparison.

B. Telescope Design

The mission of the SPARTAN-V team was heavily dependent on the optics chosen to complete the photometric imaging. The telescope used was a custom designed and fabricated folding refractor design displayed in Fig. 6 with a 70 mm aperture and 400 mm focal length. This design allowed for a light-sensitive telescope to be packaged in a small area. Without the folding design technique, the telescope would have to extend three times the length to achieve the same precision.

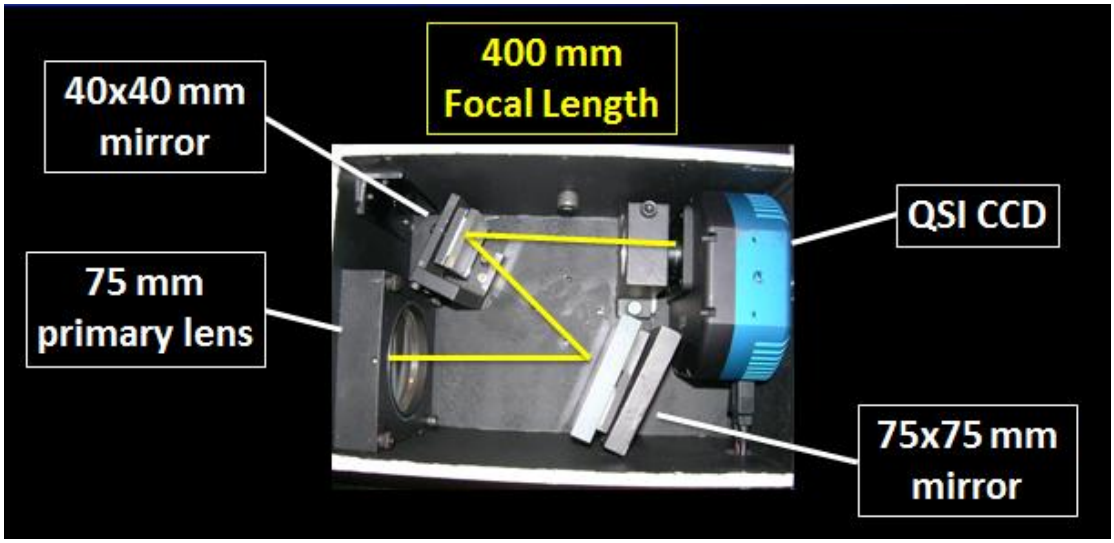

Figure 5. Detailed internal view of the SPARTAN-V telescope design.

In the lower left of the diagram is the 75 mm primary lens, where the light enters and then is reflected by the 75x75 mm and 40x40 mm mirrors into the charged-coupled device, or CCD. A close up of the CCD camera may be seen in Fig. 7, below.

Figure 6. QSI 504 CCD camera.

The telescope was machined locally with the help of team mentor, Russ Mellon of Equinox Interscience. It is comprised of aluminum 6061 and was made to fit within the "pitch arms" of the payload in order to rotate about the elevation axis.

Post fabrication, the telescope was aligned by laser and optical methods. The light path was aligned by pointing a laser into the lens and allowing it to reflect off the mirrors into the CCD mount, as seen in Fig. 8. Once the correct light path was achieved, the laser was focused to the estimated focal plane of the CCD camera.

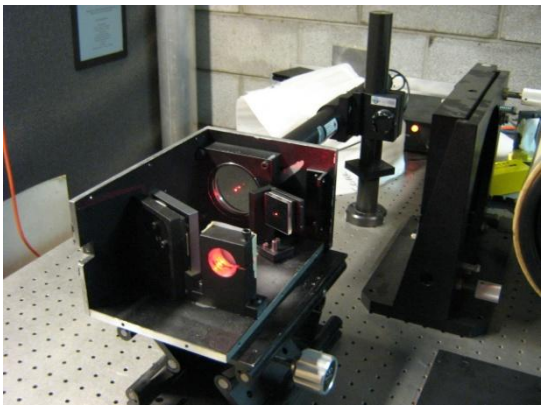

Figure 7. Laser Beam Mirror Alignment Set Up

The optical focus was then aligned using a USAF three-bar alignment chart, pictured in Fig. 9. This photo was taken by the SPARTAN-V telescope during alignment. This test allows for the camera to take a physical picture of a target, in this case, three bars; the smaller the set of bars that the camera can focus on, the greater the angular resolution.

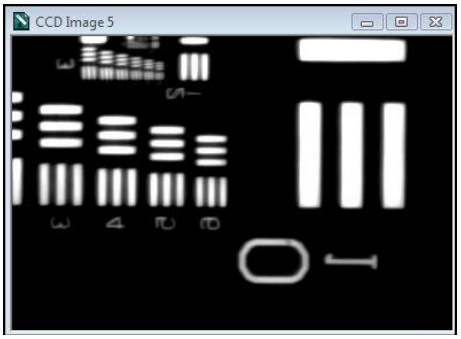

Figure 8. USAF three-bar alignment.

Using the testing set-up in Fig. 10, the telescope was focused to infinity through a collimating lens. This essentially makes the target seem very far away; this mimics looking through a telescope backwards.

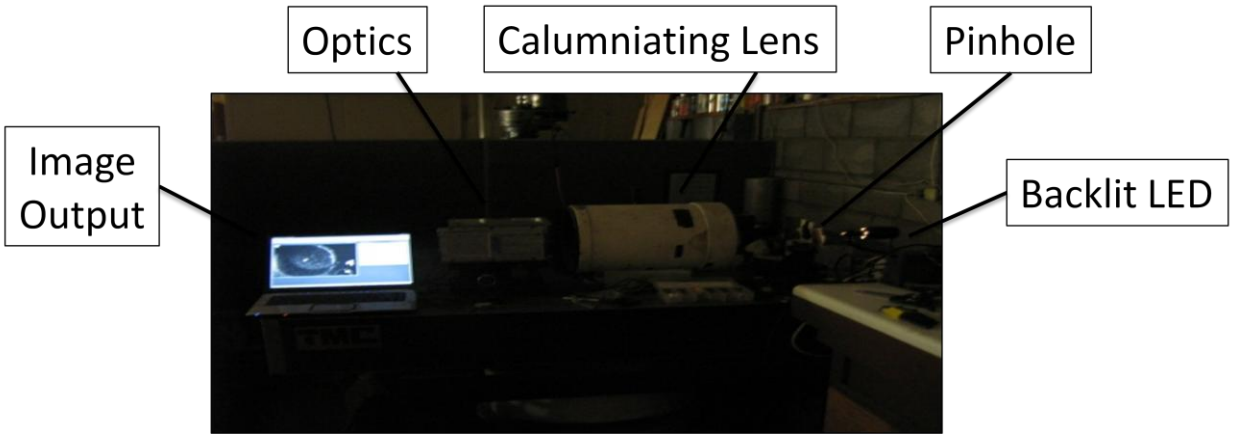

Figure 9. Optical bench set-up used during alignment.

This test was a good benchmark to align the focus of the telescope, but to further focus the camera a pinhole-LED set-up was used. By performing the same technique of focusing the target at infinity, the telescope focused on a 50 µm wide hole. This allowed for a more precise focusing of the optical system for flight. Figure 11 displays a photo taken during the pinhole-LED alignment.

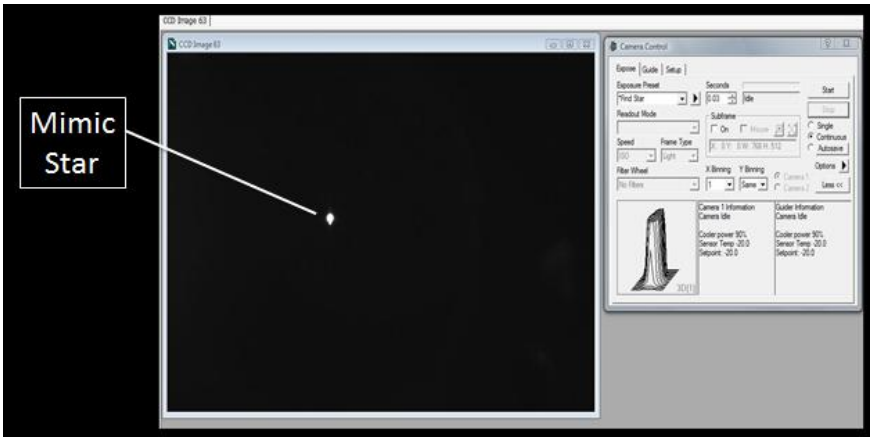

Figure 10. CCD image of 50 µm pinhole for alignment.

After alignment the telescope was used for several night observations near Boulder, CO, and confirmed to function properly. These observations verified the functionality of the telescope at ground-level conditions shown in Fig. 12, below. The telescope was later verified to operate in a thermal vacuum chamber located in Palestine, TX, granting the payload flight certification.

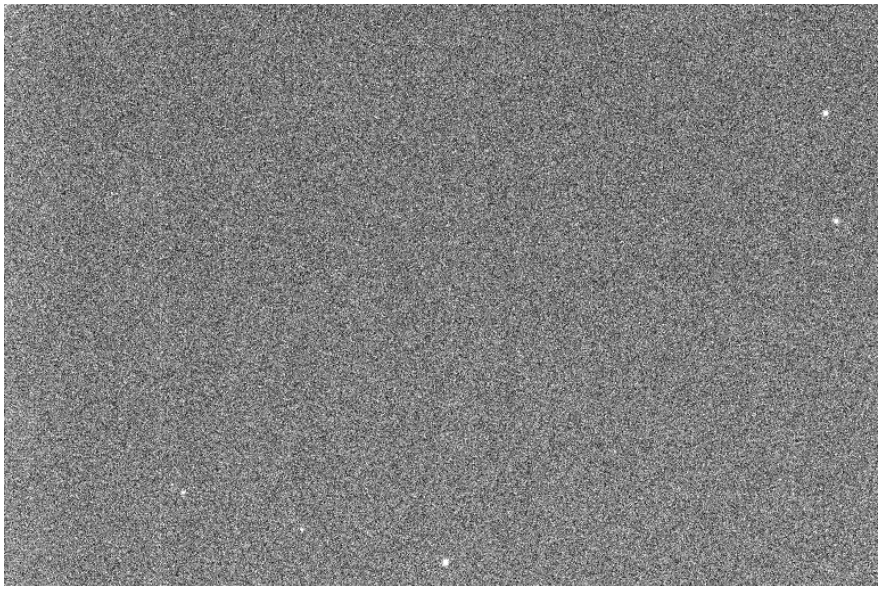

Figure 11. Ground-based star field image captured with aligned optical system.

The telescope field of view was angled at approximately 55˚ above the horizon (red line in Fig. 13), ensuring that viewing was not obscured by other payloads on flight, or the bottom of the balloon.

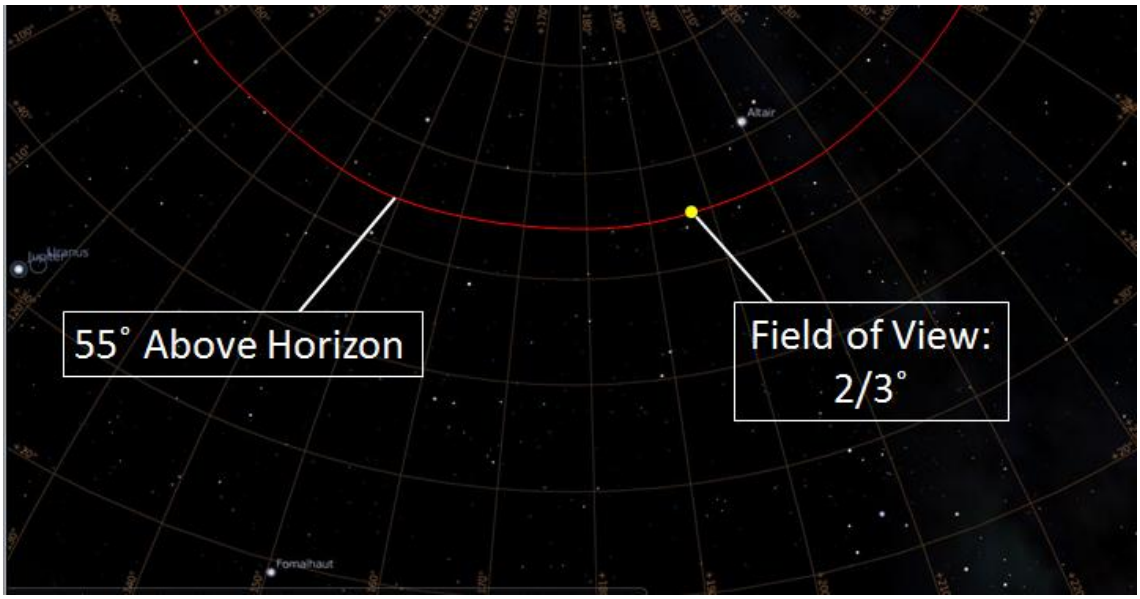

Figure 12. Telescope field of view and flight pointing angle.

C. Electrical Power and Hardware Design

The electrical hardware and power system is displayed in Fig. 14 below, which also presents a visual representation of the data flow for the SPARTAN-V system.

30V at 2.5A is provided by the HASP platform to the SPARTAN-V payload, which is stepped down through a series of linear regulators to the appropriate voltages needed for various hardware components. All components are powered to the shown voltages upon system start up, with sensor sampling initiating via the AVR microcontroller.

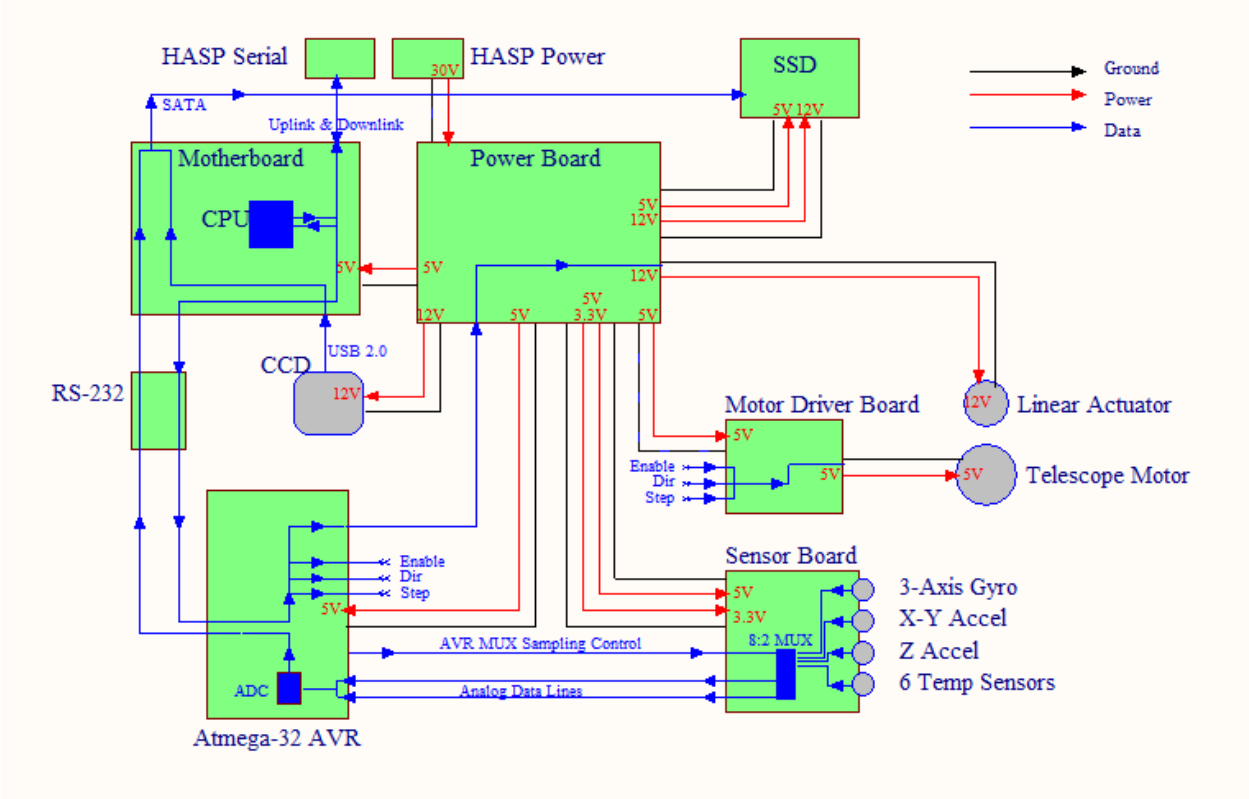

Figure 13. Electrical Hardware, Power, and Data Flow FBD

SPARTAN-V consisted of nine sensors total, and information is provided in Table 1, below. These sensors will help to characterize the HASP balloon environment, which will detail temperature, spin rates and pendulum motion of the platform at float altitude of 120,000 feet. The temperature sensors are also used to help detect thermal failures within the SPARAN-V system. All sensors were sampled at a rate of 10 Hz from the AVR microcontroller, which cycles through the sensors with an 8:2 multiplexer. The analog outputs of the sensors are sent to the 10-bit ADC of the AVR, which is then saved to the Solid State Drive (SSD) via RS-232 and SATA communication. Images captured during flight were also saved to the SSD via USB 2.0 and SATA communication.

The AVR also provides hardware logic control for the telescope motor and linear actuator. Serial uplink commands are sent to the Motherboard, which is then communicated via RS-232 to the AVR. Depending on which byte the AVR receives, it will call the appropriate action for the corresponding hardware component.

Table 1. SPARTAN-V Sensor Details

Sensor	Voltage	Sensitivity	Placement	Functional Temp Range
ADXL325 3-Axis Accelerometer	3.3V	174mV/g	Sensor Board	-40 to + 85˚C
LPR503AL Pitch, Roll Gyro	3.3V	33.3mV/˚/s	Sensor Board	-40 to + 85˚C
LY503ALH Yaw Gyro	3.3V	33.3mV/˚/s	Sensor Board	-40 to + 85˚C
AD592 Temperature Sensors	5V	1uA/K	1 Sensor Board, 2 Motherboard, 1 External, 1 Motor, 1 Linear Actuator	-25 to + 105˚C

Figures 15 and 16 detail the orientation of the sensors relative to the HASP launch vehicle, and to the SPARTAN-V payload.

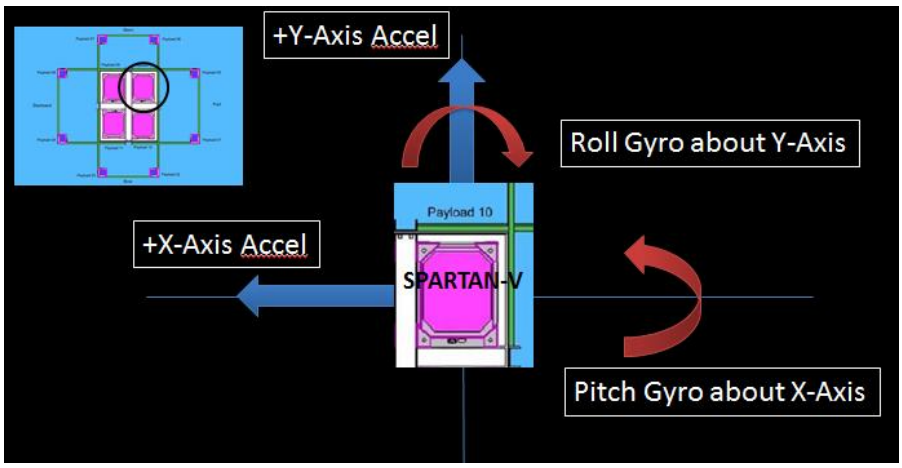

Figure 14. Orientation of X and Y accelerometers and yaw, roll gyroscope sensors relative to the HASP Platform.

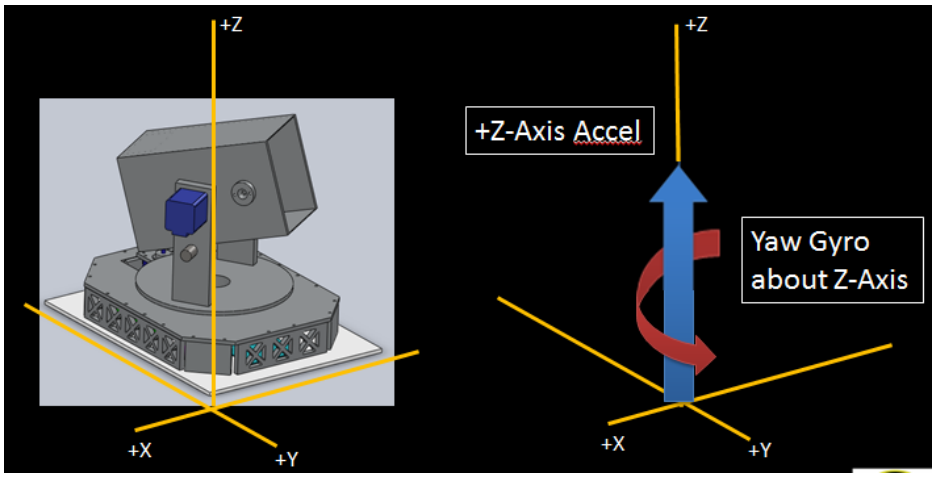

Figure 15. Z accelerometer and pitch gyroscope sensor orientation relative to the HASP Platform.

All uplink and downlink traffic is sent to the CPU of the Motherboard, which either sends health and status (H&S) packets down, or calls an action within the SPARTAN-V system. Details for each health and status packets are included in Tables 2 and 3, below. A full list of uplink and downlink commands is detailed in the Appendix (Table 7).

Table 2. Large Health and Status Packet

Large Health and Status Packet
Packet time stamp
Number of uplink commands successfully received
Number of process restarts from Watchdog
Last process restarted
Runtime since last restart
What restarted/initiated the software system

Table 3. Small Health and Status Packet

Small Health and Status Packet
Current CPU Usage
SSD Data Storage

D. Software Design

i. High Level Software Architecture

The software system is built around two major components: the motherboard and the AVR microcontroller. Figure 17 below presents a flow diagram for the SPARTAN-V software system, and primarily handled by the motherboard software. The motherboard includes a software Watchdog that monitors the status and operation of the entire software system upon the motherboard, and interfaces with the power board, CCD camera, and SSD devices. More details about the software architecture of the motherboard, microcontroller, and process watchdog are detailed in the sections below.

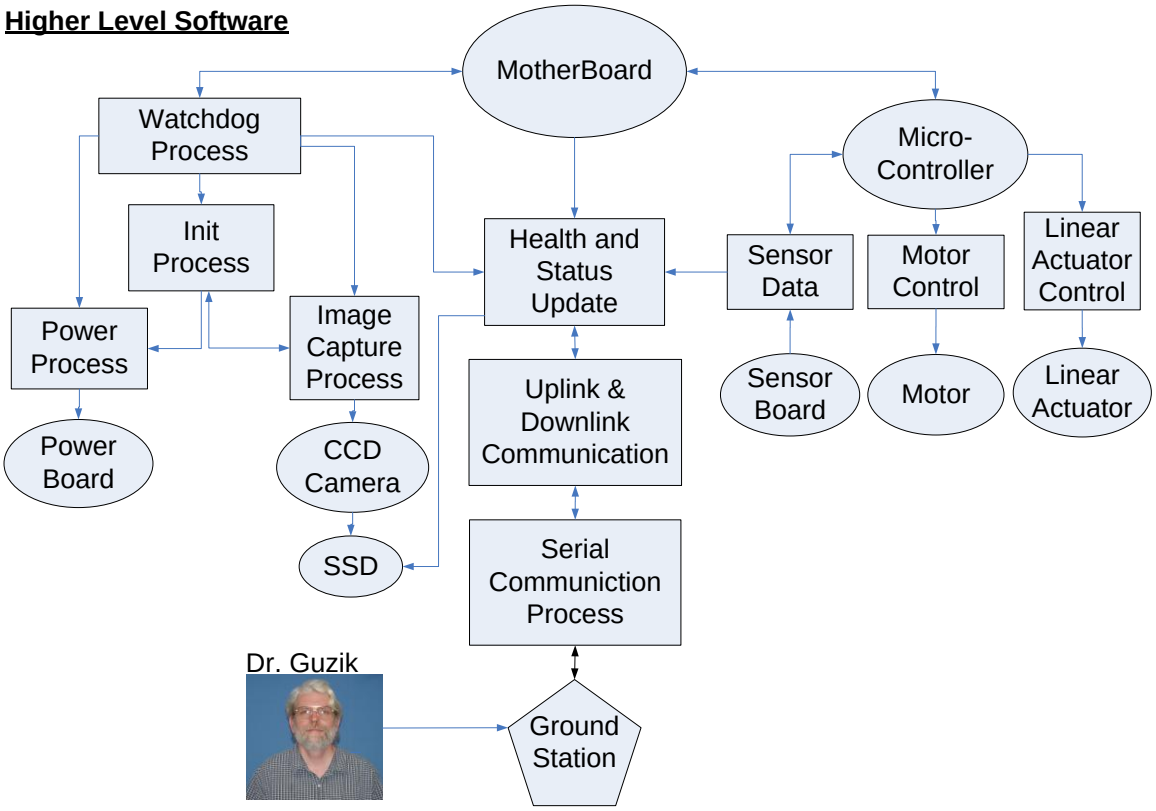

Figure 16. Higher Level Software Flow Diagram

ii. Motherboard Software

Serving as the central processing element on-board the payload, the motherboard is responsible for objectives necessary to the mission. The motherboard software system is designed to properly handle data and commands, support image capture functionality, and maintain reliable operation for the duration of the flight.

Command and data handling involves inter-system communication that spans the entirety of the physical payload. In order to provide this communication between processes, a certain structure was designed to enable parallel processing and the queuing of messages. Message queue wrappers were created for each process that contained awaiting messages (according to a first in, first out policy), where a message is either a command or data. This enhances parallel processing capabilities of the system by providing inter-thread communication, and providing more efficient command handling capabilities.

Uplink and downlink processes are essential to the command and data handling system, as they serve as the means of delivery of messages external to the motherboard through serial communication. On the motherboard side, one serial port was set to simply read data coming in, and another as the output serial port. Separating serial data in and data out on the motherboard allowed for ease with system communication with HASP and intersystem communication for hardware control. Figure 18 shows a block diagram of the motherboard's responsibilities.

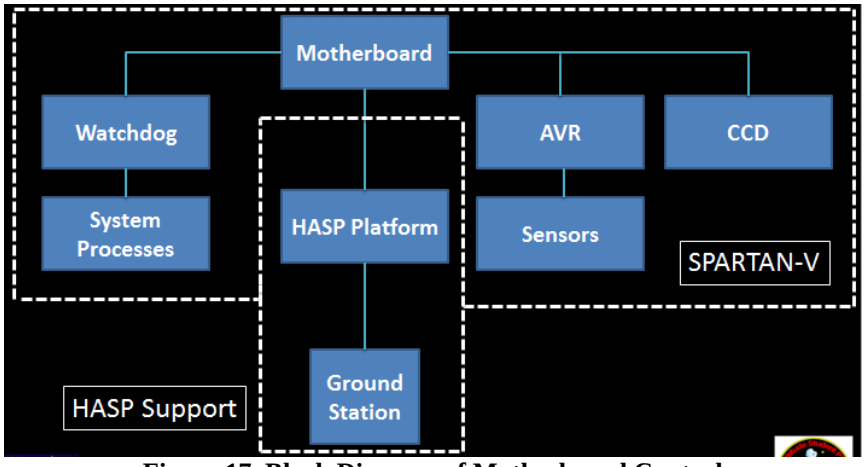

Figure 17. Block Diagram of Motherboard Control

Health and status data from the AVR is parsed, and sent to the SSD to be either stored or downlinked to the ground station upon request by the process local to the motherboard that generates health and status packets with this data and from calculated system statistics. Commands sent from the ground station are parsed and sent to appropriate message queues. If the command involves control of the linear actuator or telescope movement, the command byte is relayed to the AVR serially where it is processed by the microprocessor's control system.

Image capture functionality is defined as capturing an image using the CCD camera and storing the image to the SSD continuously throughout the flight. The image capture process controlled the camera exposure settings and timing of image capture as well as sending the

image, time-stamped, to the SSD for non-volatile memory storage. Code was written to interface the CCD camera that enables the control of its hardware. This CCD interface code was utilized also to disable undesirable hardware functions such as LED lighting, fan-controlled cooling, and optical filtering. The CCD proved difficult to interface, as the API had to be customized for compatibility with the Linux kernel and processer architecture residing on the motherboard.

Focus was directed towards designing a robust software system that would reliably function, through all of its processes, for the entirety of the flight. This was accomplished by establishing a failure recovery system, displayed in Fig. 19, that abstractly partitions the processes, where if a process within a partition fails, the higher level process will restart the inactive process.

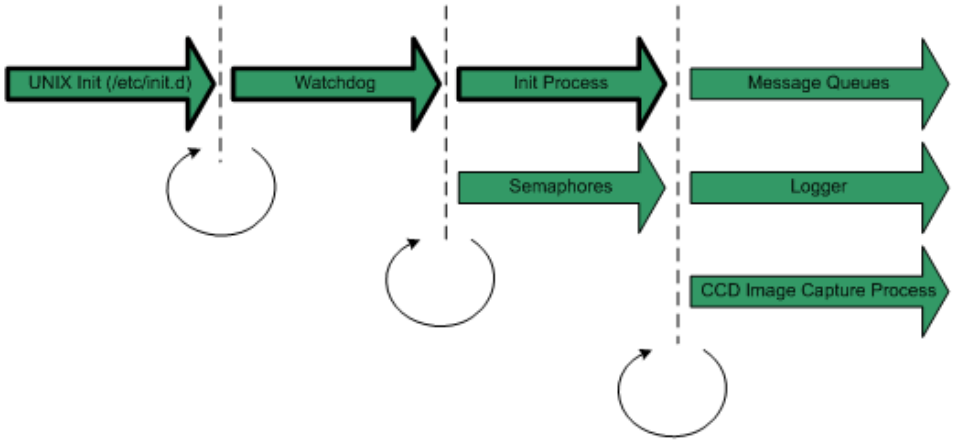

Figure 18. Process Initiation and Failure Analysis

The status of processes is monitored by a collection of semaphores which indicate whether a process is active or inactive. The process semaphores, in turn, are monitored by the Watchdog process. The Watchdog has the sole responsibility of ensuring that motherboard processes remain active. Detailed status information on active, inactive, and restarted processes are sent to the health and status process to be relayed to the ground station to also provide a human element to the monitoring of motherboard processes.

iii. Microcontroller Software

The AVR ATmega32 microcontroller provides processing parallel to the motherboard, enabling a level of multitasking necessary to achieve mission objectives. The low-level programming implemented on the microcontroller allows a greater level of interaction with hardware components that include the linear actuator, stepper motor, and sensor board multiplexors. Control of these components is accomplished general purpose input/output lines (GPIO's) and by code that manipulates them.

Sampling of sensors involves multiplexing between the analog-to-digital converter (ADC) and sensor board multiplexor. When sampling each individual sensor, the cascaded multiplexors required a specific sequence of logic HIGH GPIO's in order to receive a voltage from the proper sensor to the appropriate ADC channel, the timing and sequential order which

can be seen in Fig. 20. The temperature sensors, accelerometers, and gyroscopes are sampled at 1 Hz. Constant sampling of sensors assures more accurate characterization of flight and conditions during the capture of an image.

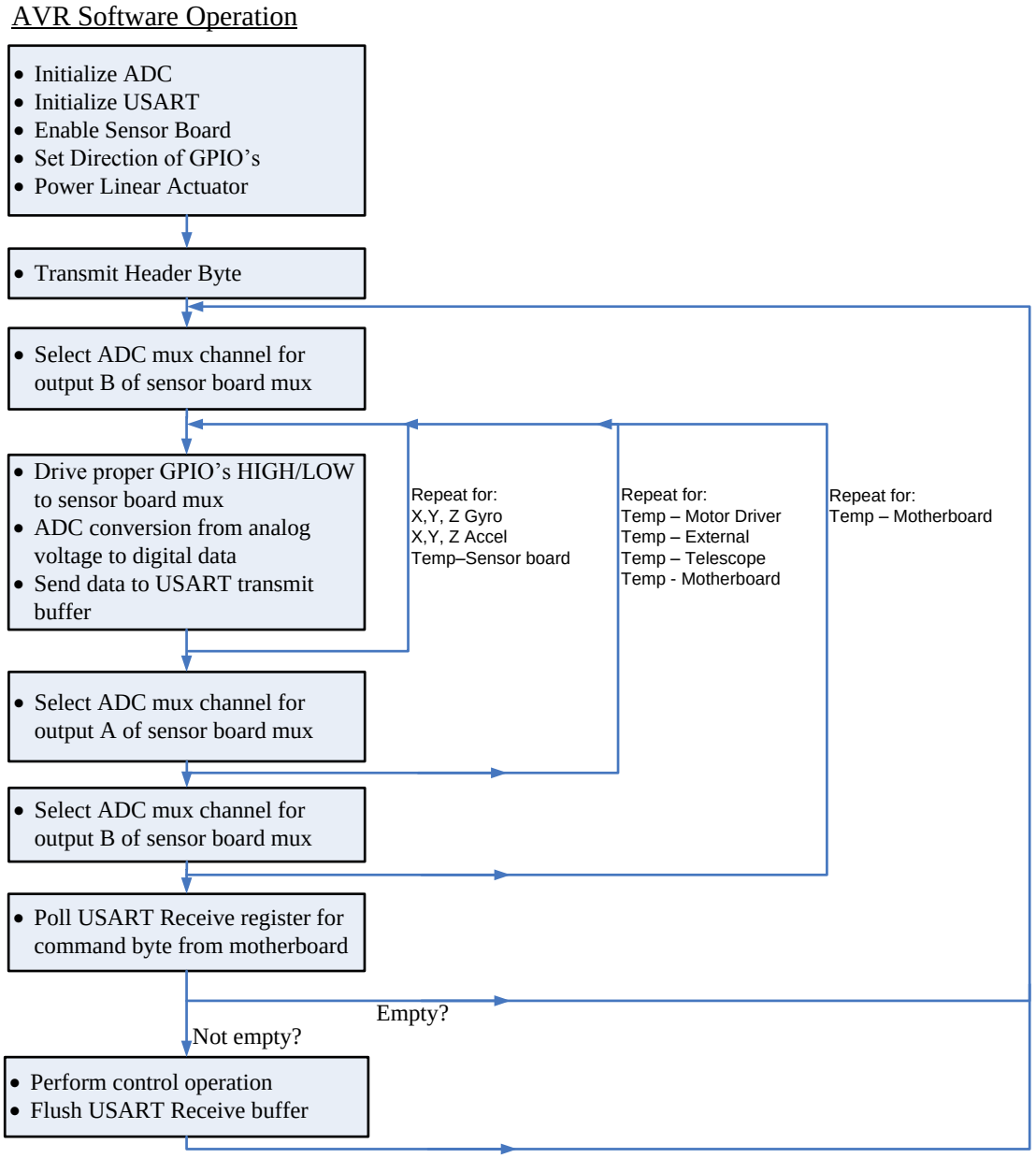

Figure 19. AVR Microcontroller Flow Diagram

Telemetry that is extracted from the sensors is placed into the transmitted serially via the bidirectional Universal Asynchronous Receiver/Transmitter (UART) of the AVR microcontroller. A single byte at a time is placed into the transmit buffer, due to the size limitation of the buffer. A header byte is transmitted at the beginning of a set of data for post-flight processing purposes. A footer byte, hex 0x0D, is transmitted at the end of a set of data in order for the motherboard to process a carriage return and to better organize the telemetry on the SSD.

Command of the linear actuator and the telescope pitch movement is originated external to the microcontroller and is to be received by the UART. The UART receive buffer is polled at 1 Hz in anticipation of a command byte from the motherboard. The baud rate of the UART was set to 9600 bits per second, with no parity and 1 stop bit.

Control of the telescope constitutes a clockwise and counterclockwise rotation in the pitch direction. The code supports 2 predetermined amount of steps equivalent to 30 and 90 degrees in either direction. It was observed that the microcontroller is required to generate a signal of precise frequency and duty cycle in order for the motor to rotate smoothly and accurately given the weight of the telescope and CCD forced upon the motor. This signal has a period of 40 milliseconds and a duty cycle of 87.5%.

IV. Failure Analysis

During the HASP 2011 flight, the SPARTAN-V team encountered no mission failures, but did experience several anomalies. The payload functioned as expected for all mission critical actions, but did not function as expected in several non-mission critical items.

A. Mechanical Anomalies

In order to protect the optical instrumentation from sunlight during the day, the telescope was in a "stowed" position until astronomical twilight. Upon command of the pitch motor - blue box in Fig. 21a and white box in Fig. 21b - the payload operator can rotate the telescope to a given elevation angle. During flight this command was sent set at an elevation angle of 55º. The motion was not confirmed real-time, as the CosmoCam was not operating by this time. Prior to termination of the payload the command to "stow" the telescope was sent, but once again, was not confirmed during flight.

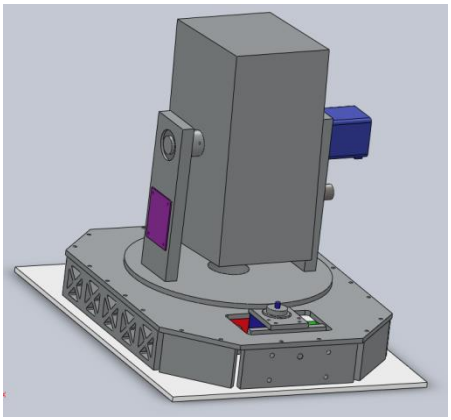

Figure 20a. Payload design, stowed.

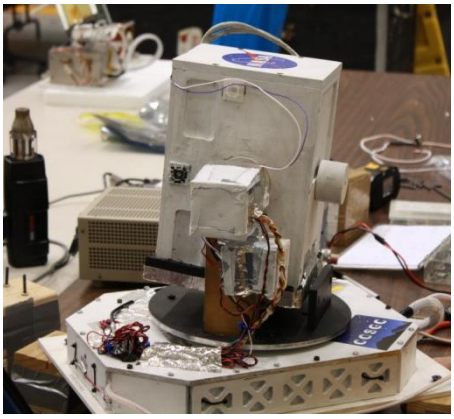

Figure 21b. Payload flight configuration, stowed.

Figures 21a and 21b display the telescope in the "stowed" position, while Fig. 22a and 22b display the "unstowed" or elevated telescope configuration.

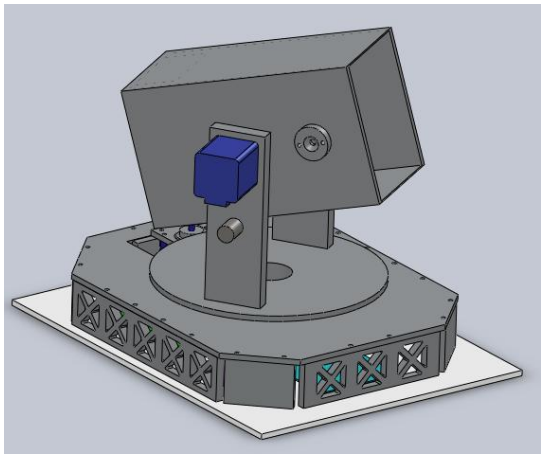

Figure 22a. Payload design, unstowed.

Figure 22b. Payload flight configuration, unstowed.

Upon post flight analysis, the commanded movement of the telescope was determined. The telescope successfully rotated to the unstowed position; as start images were taken during flight saved to the payload memory. However, the final image taken from flight, Fig. 23, includes stars. This suggests that the movement of the telescope prior to termination to stow failed. This does not hinder mission success, however, is recorded as a mission anomaly.

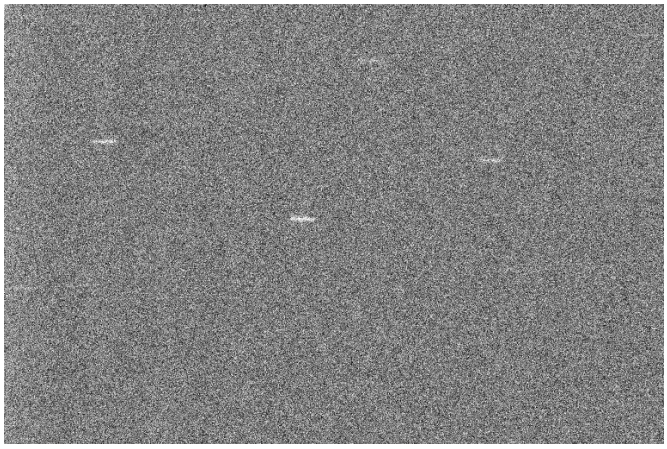

Figure 23. Final image captured during flight.

Extensive testing was performed on the functionality of the motor under extremely cold temperatures in the range of 0 to -60ºC. The bearings were found to lose performance after exposure to temperatures below -30ºC for extended periods of time on the order of several hours. It was also found that the wires connected to the telescope were freezing solid and also creating additional torque on the motor, causing it to fail. This failure to command the telescope movement was seen in TVAC and corrected for by changing to Teflon instead of PVC coated wires, which provide less torque on the motors under cold conditions. Ultimately, the stowing of the telescope was not critical to mission success, and this loss in functionality was accepted and anticipated during flight.

From the testing conduced, the SPARTAN-V team attributes this anomaly to bearing failure due to cold environmental conditions that provided excess torque on the motor hindering its movement.

B. CCD Anomaly:

SPARTAN-V experienced an anomaly in which the firmware for the CCD camera would discontinue capturing images, and would not restart upon receiving the proper uplink command. The only method to reinitiate the CCD to capture images again was a system power reset, and was effective the five occurrences during flight.

The anomaly occurred due to a "while(1)" loop within the CCD firmware code, which iterated a value from 0 to 10000, with each iteration capturing an image. Once the iterator reached 10000, the value should have reset to 0, and start the same process again. Monitoring the instance post-flight showed the CCD no longer responding after the value of 10000, and was unable to be restarted since the CCD firmware is separate from the Watchdog and motherboard CDH software. It appears the CCD firmware was unable to reset the value back to 0, and thus was stuck in an infinitely loop and would become non-responsive. Simply resetting the value to 0 at the end of the for loop proved to solve the problem. This could have been avoided with thorough system testing prior to launch.

C. Sensor Anomalies

i. Gyroscope drift

Upon post-flight processing of the gyroscopic sensors housed within the payload it was found that the accuracy of the data was significantly lower after approximately nine hours into flight. The data resembles noise, and in the pitch direction is incorrect, as we were not spinning about the pitch direction at 14 °/s for the last four hours of flight.

All three gyroscopes showed the same characteristic loss in accuracy at relatively the same time during flight in the data sets, with the yaw gyroscope being the least affected. The anomaly can be attributed to the cold temperatures experienced as night began, reaching values that exceeded what the sensors were rated to. The roll and pitch sensors are mounted through the same electrical interface as one sensor while the yaw gyroscope is its own sensor, explaining why it was not affected by the same magnitude the roll and pitch sensors were.

ii. Temp sensor failure

Also found during post-flight analysis was an anomaly in the temperature data set. These sensors were rated to -25ºC (Table 1), a value exceeded about one hour into flight. Until this point, the temperature sensors seemed to be recording accurate data – as compared with HASP temperature data. After this point the values do not correlate with HASP data and exceed the temperature range expected during flight. The sensors are believed to have broken at the low temperatures, explaining the loss in accuracy after this point.

V. Results and Discussion

A. Photometric Results

To calculate the brightness of a star, two different measurements were taken: one determined the number of background counts which were present, and the other determined the number of photons in the star.

First, the image was corrected by a flat field image. When dealing with CCDs, there tends to be a variable sensitivity within the detector. By exposing the CCD to a uniform light field (in this case a cloud, Fig. 24) and moving around the camera, an image is received that should have, for all practical purposes, exposed every part of the image to the same density of photons. The average of this image is taken, and divided by each pixel value. This results in an array that, when used to scale an image taken by the CCD, corrects for the local oversensitivity and low sensitivity of the CCD. For this experiment, a few hundred images at short exposures were taken of a cloud and combined, instead of a single long exposure.

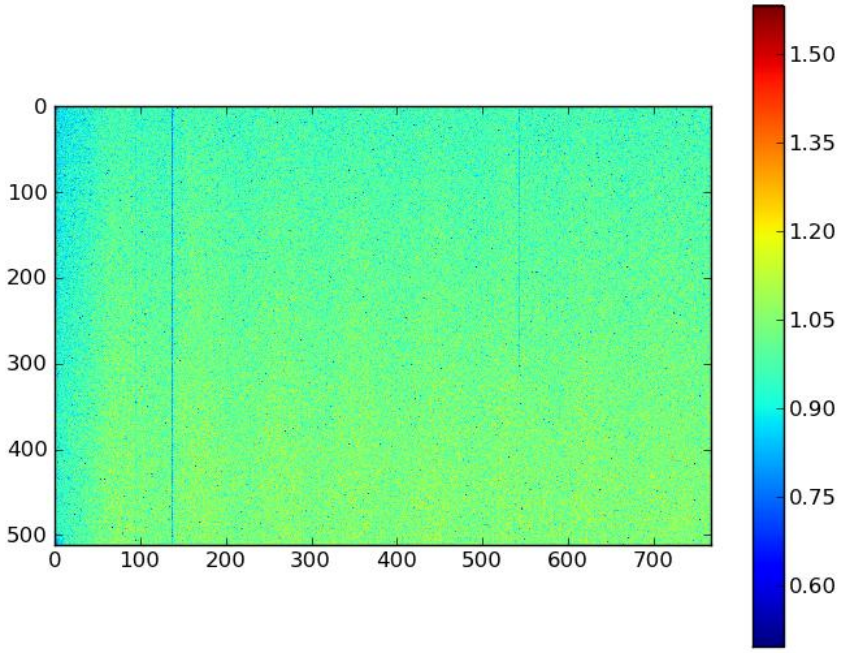

Figure 24. Flat field image.

A ring is then taken around the star which has an inner radius of 14 pixels and an outer radius of 21 pixels as shown in Fig. 25. The purpose of the ring was to provide a measurement of the background counts for the image. However, because of the likely inclusion of other stars in this radius, a robust measure of the mean must be taken. To do this, the median of the image, and the median difference of the image from the median were taken from this ring. By scaling the median difference of the image from the median by 1.4826 an approximation for the standard

deviation was found. Then, using these rough estimates, the points beyond four standard deviations away from the median were removed from the ring (due to the likelihood of them being stars). After this, the mean and standard deviation of the remaining pixels were calculated, result in robust measurements.

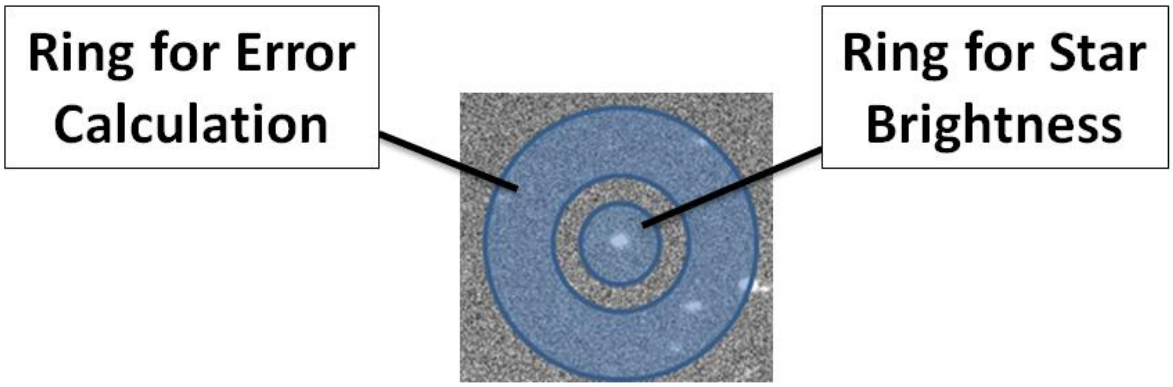

Figure 25. Star brightness and error calculations.

After this, a circle of radius nine was taken around the star to calculate the star brightness as shown in Fig. 25. The total number of counts within this ring were summed together to get a value for the counts of the star plus the sky. Because the detection of photons in this case is based on a Poisson distribution, the measurement $C_{star+sky}$ produces the associated error of $\sqrt{C_{star+sky}}$. To isolate the brightness of the star, the brightness of the sky was subtracted from the measurement. However, since the brightness of the background also has an associated error, the following result was retrieved:

$$C_{star+sky} \pm \left(\sqrt{C_{star+sky}}^{2} + \sqrt{C_{sky}}^{2}\right)^{\frac{1}{2}} = C_{star} \pm \sqrt{C_{star} + 2C_{sky}}$$

These are the values calculated in the measurement of the star signal and noise. The Signal to noise ratio is simply the ratio between C_{star} and $\sqrt{C_{star} + 2C_{sky}}$. [2].
These values were calculated from two sets of images: those from flight which showed little streaking, and those from the ground that we took to illustrate camera function ability. Both sets of images contained 0.2 second exposure times.

The measured signal to noise ratio for balloon images, however, when compared to the signal to noise ratio of the ground images, appears nearly identically in images containing stars of the same relative magnitude (for example, an image with a brightness of 14,223 photons on the ground had a signal to noise ratio of 34.5, whereas on the balloon a value of 14,157 had a signal to noise ratio of 33.4) which is contradictory to what we would expect. However, this calculation only took into account three sources of error: the background, the star, and the flat field offset. Read noise would increase both errors by roughly the same amount, and the

SPARTAN-V team did not have a good enough measure to determine whether dark current was a significant issue or not.

However, outside of these standard sources of error lies the biggest contributor: even the most stationary of our images still experienced some blur. Much of the written program relied on the assumption that the stars would be perfect circles. However, as can be seen by the images, very few of the stars resemble perfect circles. Therefore the images contain this large source of error which would be extremely difficult to account for.

Originally the SPARTAN-V team had the goal of achieving a signal to noise ratio on the order of 10^5, however, due to the lack of a control system the amount of images taken was drastically less than anticipated during the design phase. These factors resulted in the best signal to noise ratio during flight being 122.

B. Streaked Images and Spin Rate Calculations

Because of the lack of a control system, most of the images display streaking due to the spin of the balloon, as seen in Fig. 26. These streaks underwent analysis so to verify the gyroscopic data and characterize the balloon platform.

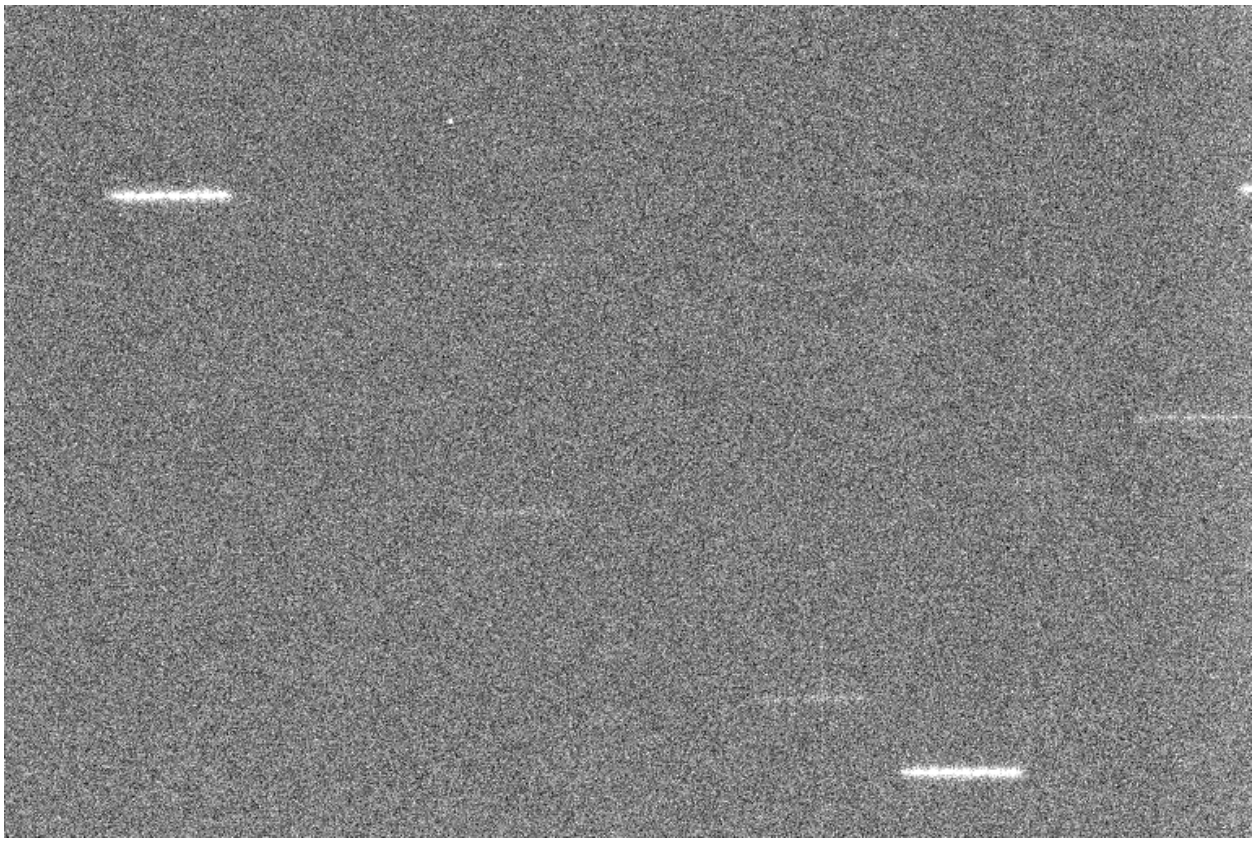

Figure 26. Streaked star field image.

To figure out the length of the streaks, the images were scaled by the flat field and passed the image through a median filter to normalize the image and make the smears which are the stars more uniform, displayed in Fig. 27. This also removed objects such as cosmic rays which would throw off programs looking for stars.

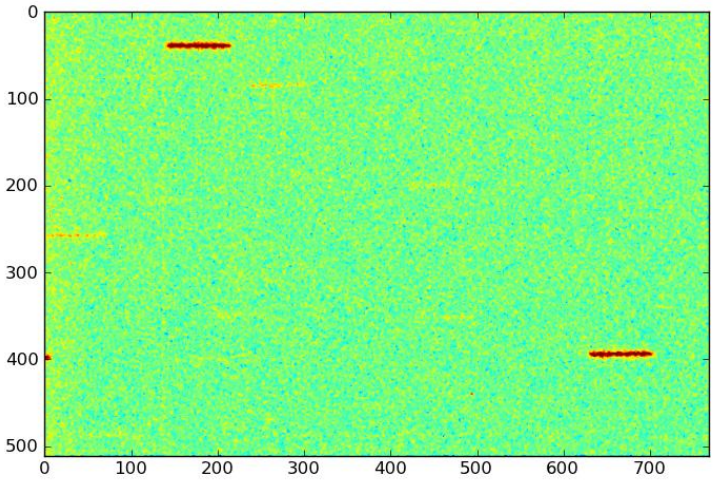

Figure 27. Median filter image.

Then the median value of the image was subtracted out, and any point less than three sigma away from zero was removed from the image. Then, the program determined took a subset of the column containing the point located, and measured the distance between the top of the star and the bottom of the star, as seen by the red lines in Fig. 28, below. After measuring this width, the row containing the located point was selected, and using the zeros which replaced the background values, a length was discovered. This total length then had the width of the star subtracted from it to account for the offset due to the spread of light over the CCD. The resulting value was then converted into degrees.

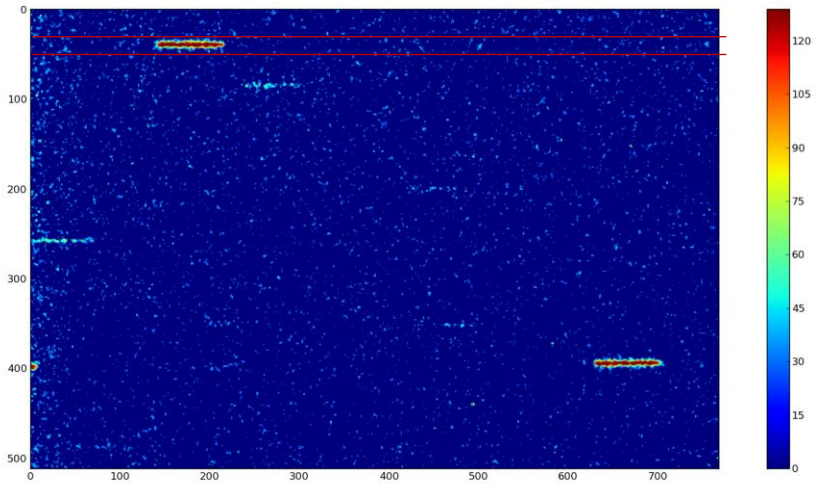

Figure 28. Selected streak method.

C. Error Analysis[3]

The goal of the project was to measure the brightness of a star and its error over several measurements. The sensor which records the images, the CCD, returns an array detailing the spread of photons which hit the sensor.

Calculating the brightness of an object is somewhat trivial: One simply counts the number of electrons which are detected by the CCD, multiply by a camera specific scalar which converts the counts into the number of photons, and then subtract out the background. Accounting for all the possible sources of error, however, is a much more complicated matter.

The following are standard sources of error when dealing with CCD images:

Dark Current: Dark current is the result of thermally excited electrons which build up in the pixels of a CCD. Dark current accumulates over time as a function of the CCD temperature, and at low temperatures and exposures is generally miniscule. To account for this, Dark frames can be taken, which are zero-light exposures over lengths of time equivalent to the length of the actual images. In this experiment, due to the mechanical setup, dark current was an unmeasureable source.

Pixel Non-Uniformity: Not all pixels are created equal. Some are more sensitive than others, leading to erroneous values. To account for this, a flat-field was taken, which is a long exposure over a uniform field. The pixels can then be scaled to account for this offset.

Read Noise (on-chip): Read noise is one of the more limiting factors when it comes to CCD images. It is the error build up from on-chip sources, and unique to the camera. Its presence can be overcome by combining large number of images.

Shot Noise: Shot noise is caused by the random arrival of photons. Since the measurement of randomly arriving photons is a counting problem, it can be modeled by a Poisson distribution, whose error can be reduced by combining large numbers of images.

Electronic Interference: This is the basic build up of error due to all the sensitive electronics within a camera, and their effect on the surrounding electronics.
CCD Camera Noise (off-chip): This is the error introduced by the camera when the image is converted from an analog signal into a digital value. This error is not necessarily random, and can sometimes appear periodic. In well-designed cameras, however, the noise should be random, and therefore decrease when multiple images are combined.

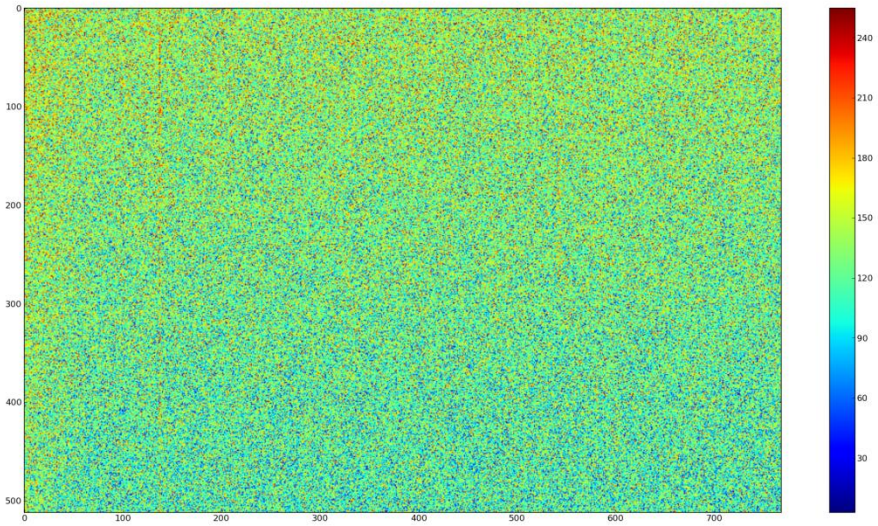

Figure 29. Bias image.

Looking at the above bias image (zero second exposure image), the buildup of dark noise over the readout of the image can be seen. Images are read from the CCD to the SSD row by row, causing the unbalance seen in this image. This indicates that dark current may have been a significant factor in the measurements.

D. Balloon Environment Characterization

As a secondary mission the Colorado team integrated a set of sensors into the payload in order to characterize the balloon environment of the HASP platform. The SPARTAN-V payload included a three axis gyroscope, three axis accelerometer, and six temperature sensor. The gyroscopes and accelerometers were placed within the electrical casing and the temperature sensors were spread throughout the payload, including one mounted externally. These sensors began recording data starting approximately 1 hour prior to launch and ending just before termination prior to descent, 14 hours after power on. Figures 30-38b present the results from the SPARTAN-V sensors during flight.

i. Gyroscopic analysis

Figures 30-33 display the spin rate of the platform during flight. Near one hour the gyroscopes show a large amount of activity; this is the time at which launch occurred. Once float altitude was reached, around 2 hours, the spin rate in all three sensors became much more uniform. At this point the spin rate in the roll and pitch directions stayed close to zero for the majority of float until the sensor failure described earlier at approximately 10 hours after power on.

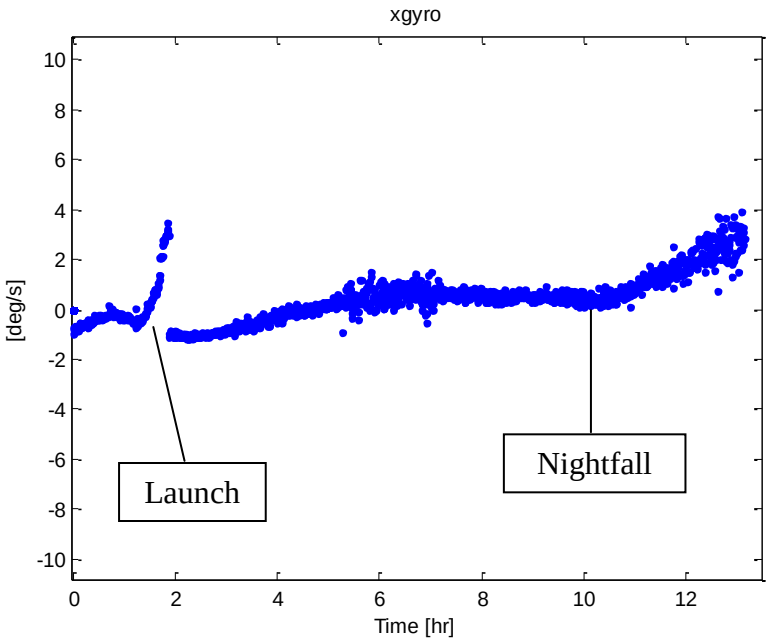

Figure 30. Roll gyroscope.

As displayed in Fig. 30 and 31, the x and y gyroscopes measure the angular velocity about the roll and pitch axes. These values were expected to peak during launch and ascent and settle to around zero once float altitude was reached; however, unexpected measurements were recorded starting approximately nine hours into flight.

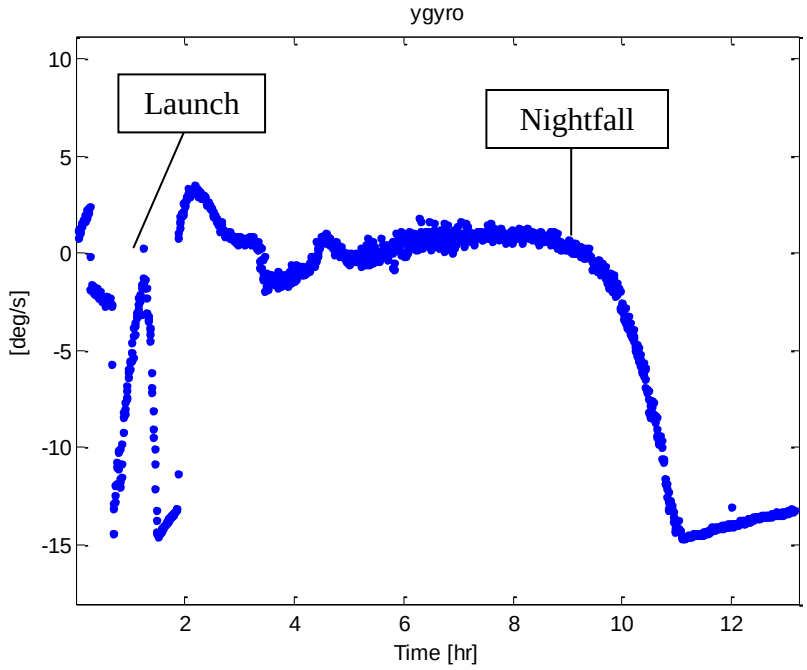

Figure 31. Pitch gyroscope.

The roll and pitch gyroscopes both tend away from zero after ten hours after the payload was first turned on. This data is erroneous, as the team has validated the gyroscopic measurements by calculated a spin rate of the balloon from images taken during the night. The drift in the gyroscopes is very possibly due to the cold temperatures seen during flight. These sensors were rated to -40°C and it is possible that the cold temperatures during flight negatively impacted the read out of these sensors. According to HASP measurements, the external temperature reached as low as -70°C. It is difficult to determine what the temperature was internally for the SPARTAN-V payload due to the failure of the temperature sensors earlier in flight.

The yaw axis, or z gyroscope shows the same drift around nine hours into flight, but is the least affected out of the three gyroscopic sensors. The spin rate about the yaw axis is by far the most active out of the pitch, roll, and yaw directions and can be viewed in Fig. 32. This is of great importance to optical payloads for two reasons: the length of exposure (integration time) for a photo is dependent on how long a telescope can remain steady on a target, and the signal to noise ratio is heavily dependent on the number of images of the same star field – if the spin rate is sufficiently slow, one could take a large amount of images of the same star field.

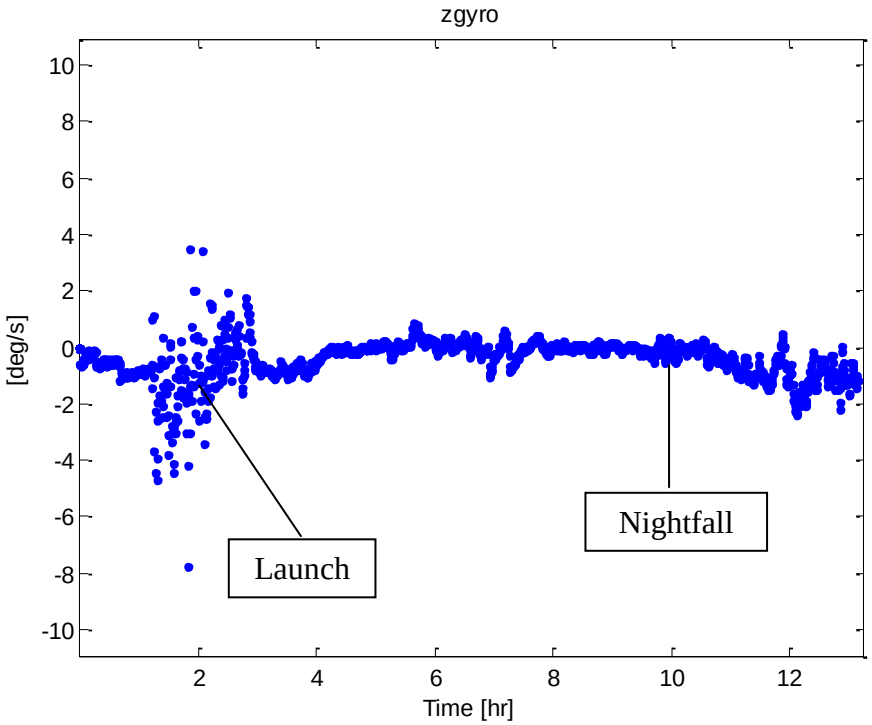

Figure 32. Yaw gyroscope.

Like the roll and pitch gyroscopes, the yaw shows a large amount of activity during flight; values range from 15 to -15 º/s. However, these sensors were sampled every second, making it difficult to accurately characterize the spin rates during launch and ascent. For the purposes of this project, the spin rate during float was the most important factor to characterize. During this time, before the sensors accuracy was lost due to gyroscopic drift, values remained within plus or minus 2 º/s as seen in Fig. 33, below.

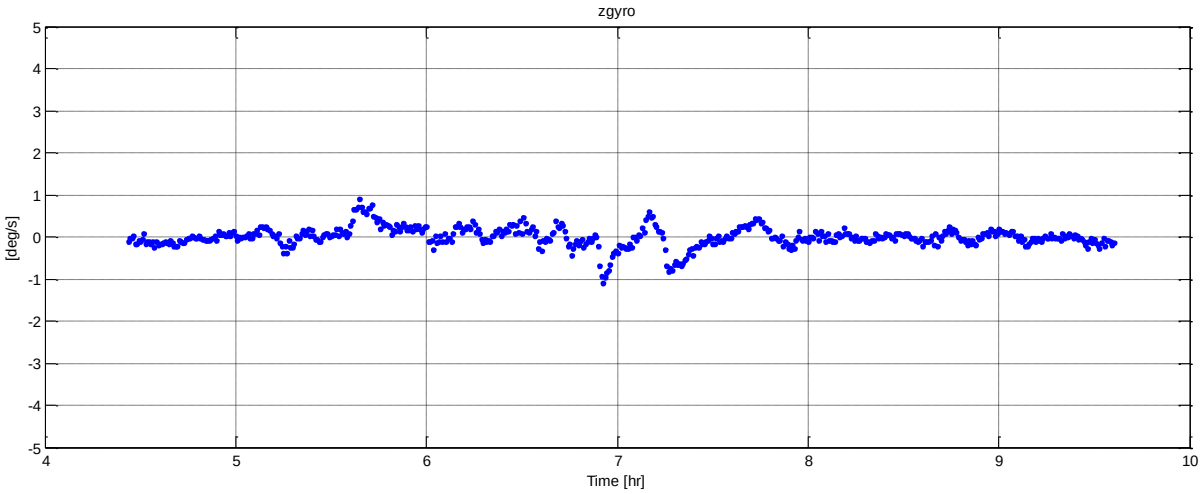

Figure 33. On-float yaw axis spin rates.

From previous years, 1.5 º/s was given as a max spin rate during float. This year's team has found the maximum spin rate in the yaw direction during float to be 2.28 º/s. The average of the yaw gyroscope values is 0 º/s, suggesting that for every counterclockwise rotation, the platform spun counterclockwise one rotation. The average spin rate during these oscillations was found to be 0.89 º/s. These values, as well as select statistics for the roll and pitch gyroscopes during float can be viewed in Table 4. The "instantaneous" values are values taken directly from the flight data set at one sample per second, and the "30 s time avg." values are averaged values of the flight data set for 30 second intervals, approximately one twentieth of the average oscillation of the balloon during flight.

Table 4. Gyroscopic data summary during float.

	Axis	Max CCW* [°/s]	Min CW** [°/s]	Average [°/s]
Float:	Roll	8.13	8.45	0.39
(instantaneous)	Pitch	7.46	8.01	0.43
	Yaw	2.14	2.28	-0.01
Float:	Roll	1.80	1.12	0.49
(30 sec time avg.)	Pitch	1.51	0.92	0.47
	Yaw	0.89	1.08	0.00

Initially, it was thought that the balloon rotated in a constant direction with various speeds of rotation. However, it was found that the balloon platform was seen to rotate in a somewhat oscillatory manner. Due to this, the spin rate of the platform changes from its maximum value of rotation and gently slows to zero, at which point the rotation changes direction and reaches a local maximum value in the other direction.

From analysis of the images taken during flight, the maximum spin rate in the yaw direction an optical system should be designed to correct for if a control system was implemented is 2.38 º/s, assuming that the system only plans on operating during float.

ii. Accelerometer analysis

The accelerometers are a measure of the perturbations during flight. The key measurements of the platform characterization that can be pulled from the accelerometer data are shock profiles. At a sampling rate of 1 second, and because vibrations on the platform were considered to be negligible, vibration profiles from the SPARTAN-V payload are not detailed enough to provide valuable conclusions. Assuming vibrations to be negligible was a valid assumption as during a low spin rate, images were not distorted by any perturbations. Figures 34-36 show the accelerations in the roll, pitch, and yaw axes.

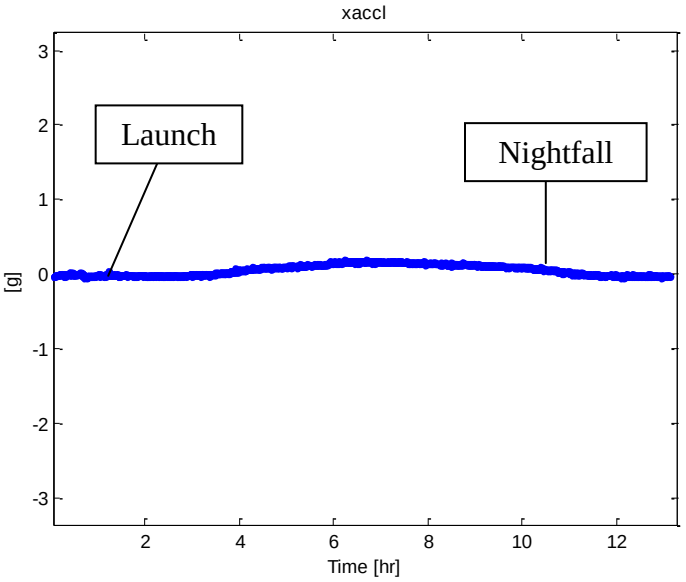

Figure 34. X axis accelerometer.

All three accelerometers show very little activity during flight, even during launch. In the roll and pitch directions this suggests that there is little to no forces acting on the payload in the horizontal direction. This was expected by the Colorado team, however, it is interesting to note that the force that causes the balloon to travel – the wind – was not sensed by the accelerometers on board. This may be due to the very small perturbations caused by the wind, and our sensors were not accurate enough to pick this up. The roll and pitch accelerometers do have a raise in measurement about the time that the platform would have been pushed by the winds in the upper atmosphere. However, the yaw accelerometer displays the same raise in values, suggesting that this was a systematic error in the system. Again, these sensors were rated to -40°C, and it is possible that the cold temperatures at altitude affected the read out of these sensors.

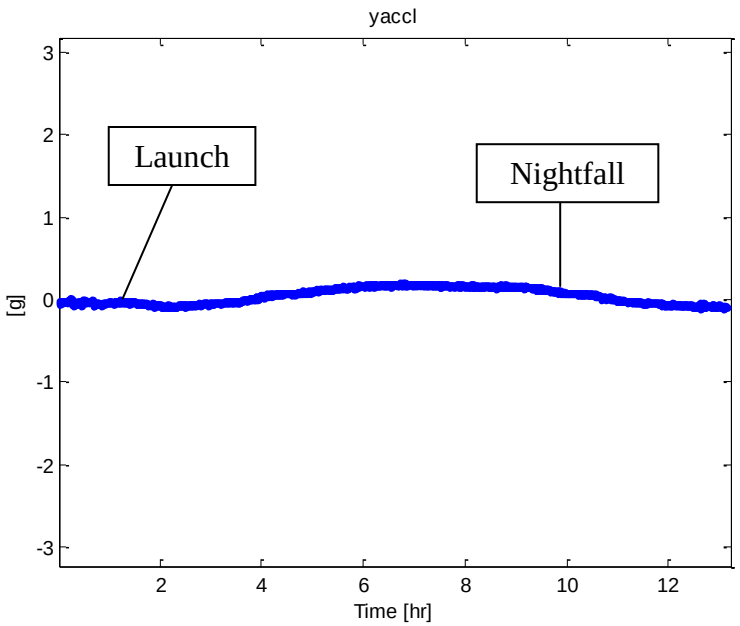

Figure 35. Y axis accelerometer.

Due to the slow sample rate of the accelerometers, quick changes in movement could not be detected. In the image analysis, two photos were found to show a large movement in the yaw direction. This measurement was not picked up by our sensors due to their constrained level of accuracy.

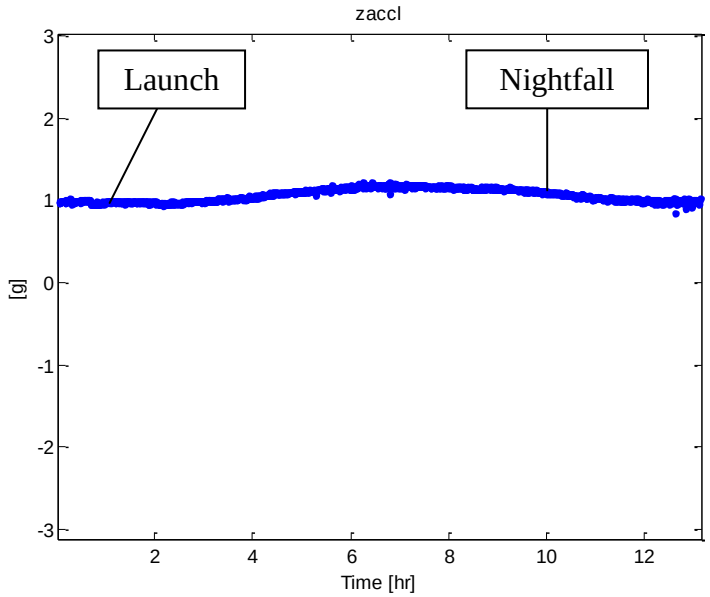

Figure 36. Z axis accelerometer.

Table 5 displays the average, maximum, and minimum values of the three previously mentioned accelerometers during the flight of the HASP platform.

Table 5. Accelerometer summary.

Axis	Max [g]	Min [g]	Average [g]
Roll	0.29	-0.34	0.05
Pitch	0.31	-0.21	0.06
Yaw	1.42	0.58	1.06

From the data taken from the accelerations during flight, the HASP platform was found to be a very "nice" flight. Due to its low shock and vibration profiles there is very little stress on the on-board payloads' structures. This makes it a valuable testing ground for space experiments, as well as a likely candidate for optical observations.

iii. Temperature sensor analysis

The SPARTAN-V payload included six temperature sensors. The position of these sensors is as follows: two adjacent to the mother board, one adjacent to the motor driver board, two internal in the structure, and one mounted externally. Figure 37 show the plots of these sensors during flight. These plots show the general trend of the temperature during flight, a more detailed plot will be provided for in-depth analysis.

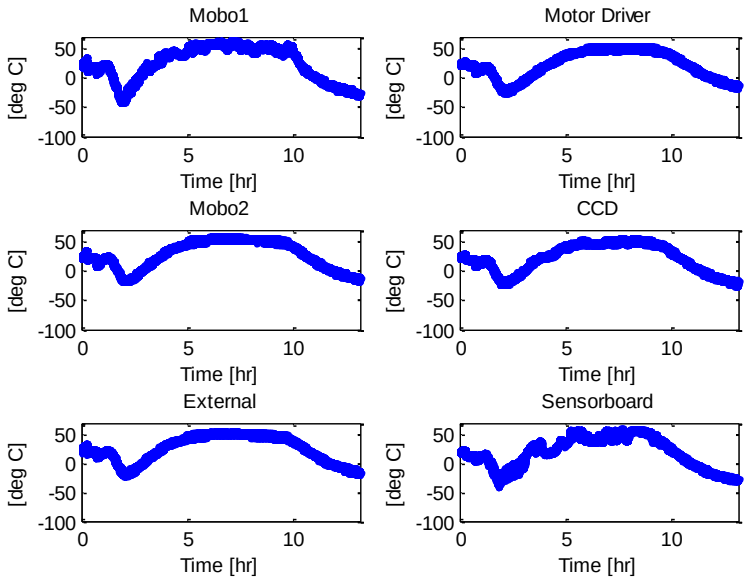

Figure 37. Temperature plots.

The maximum, minimum, and average values of each temperature sensor are displayed in Table 6, below.

Table 6. Temperature summary.

Temp Sensor	Max [°C]	Min [°C]	Average [°C]
Motherboard 1	64	-40	25
Motherboard 2	58	-18	29
Motor Driver	52	-25	24
Internal	55	-25	23
Sensorboard	55	-22	27
External	61	-40	20

As previously discussed, the temperature sensors were rated to a value of -25°C and this minimum was exceeded during flight. This led to inaccurate readings from the temperature sensors as seen in the plots above. The HASP platform recorded the minimum temperature to be -70°C, a value that was never reached by one of the sensors on the Colorado payload. Initially, the sensors recorded accurate values (Ft. Sumner at about 25°C on launch day). After the first drop in temperature the sensors bottom out at -40°C and from this point forward, about 2 hours after power on, the sensors values deviate from those measured by the HASP platform.

Although the values of the sensors are inaccurate, the trending of the temperature still gives an outline of the temperature profile during flight. Figure 38a displays the external temperature sensor and shows the drop in temperature after launch, the heating up of the payload during the day, and the sharp decrease in temperature during the night. Beside it, in Fig. 38b, is the recorded temperatures on the HASP platform. This comparison validates the SPARTAN-V temperature sensors for the time period starting at launch until approximately 3 hours into flight.

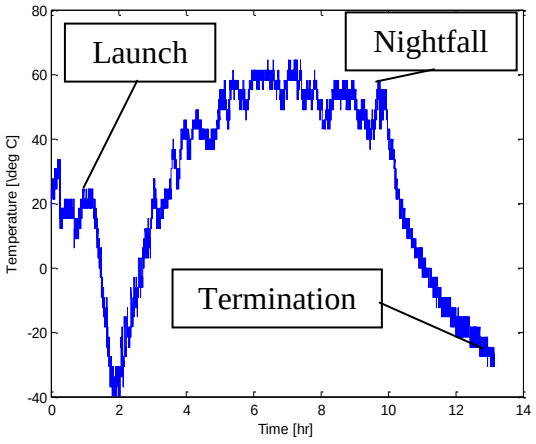

Figure 38a. SPARTAN-V external temperature.

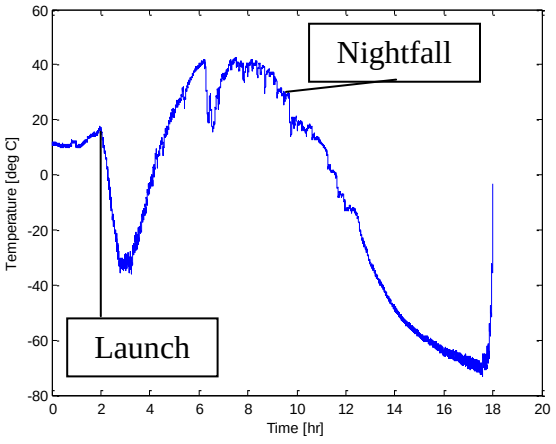

Figure 38b. SPARTAN-V external temperature.

The data taken to characterize the HASP platform and balloon environment suggests that the HASP platform is a viable, near-ideal, environment to conduct space based observations. The major hindrance to optical payloads in the future will be the rotation about the yaw axis. The next largest obstruction is the thermal environment which provides both hot and cold extremes. Speaking from the perspective of the aforementioned sensors on-board the Colorado payload, it is extremely feasible to conduct optical observations from a balloon-borne platform.

iv. Spin rate verification

The streaking in the images was initially thought to be detrimental to the data analysis, but under further analysis it was determined that the streaked images could be used to validate the gyroscopic measurements. Figure 39 is a chart that displays the absolute value of the spin rates of the HASP platform measured by analysis of the streak length of stars in images taken by the CCD camera. Compared with Fig. 33, the magnitudes of the rotation rate are very similar. This method of streak length analysis proved useful in validating the gyroscopic sensors during float.

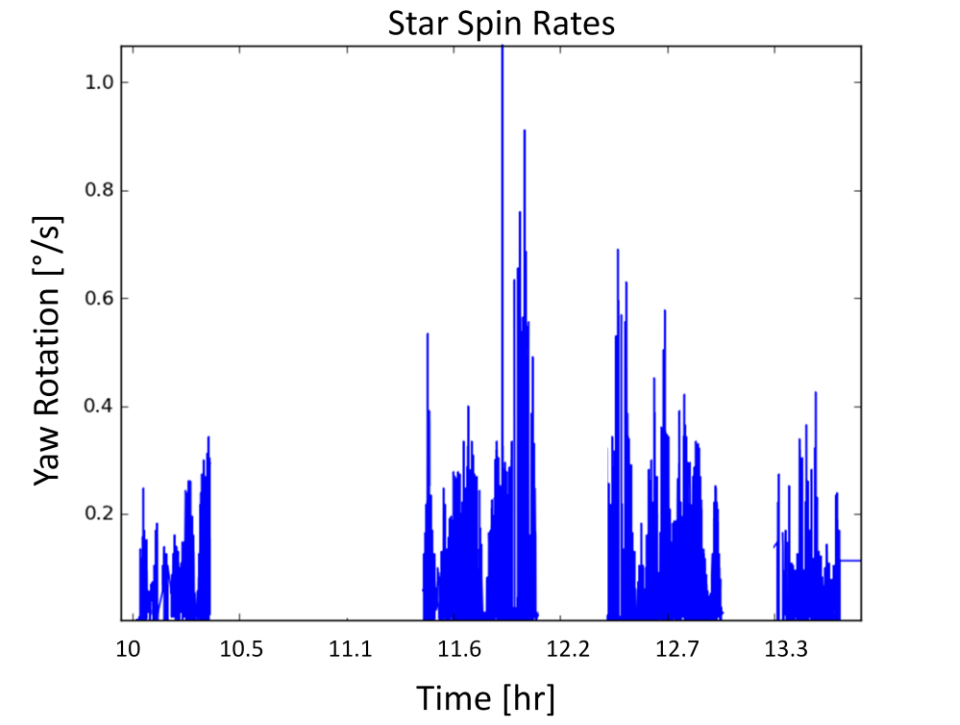

Figure 39. Yaw spin rates determined by images.

VI. Lessons Learned

A. Proposal Phase

System requirements – have them. During the beginning of the project it is sometimes difficult to comprise a set of requirements for a new project, but they will be the guiding light for the project. Each requirement should be achievable, affordable, verifiable, unambiguous, and expressed in terms of a need and not a solution. Whenever a decision needs to be made to make or change something to the system you will have the requirements to help you make your decision. When preparing for flight and you verify all of your requirements, you know you are ready.

Should not allow hardware selections develop your requirements – Your requirements should flow down to define the requirements that you need from your hardware, not the other way around. Every design choice should be backed up by your requirements.

Establish schedule for internal design reviews – Use the fall semester to develop a strong design, and internal Space Grant design reviews are a great way to do this. Really use these reviews as major deadlines and place as much time and effort into your design as possible. You don't want to be making design changes as you're developing systems and making component purchases.

Limit moving parts. If you are thinking about adding any moving parts to your system know that it will increase the complexity of the system tremendously; it does not matter if it is an off the shelf item. Try to limit the amount of moving parts used in the design – Keep It Simple.

B. Design Phase

Need requirements review – Something that the HASP 2010 team didn't have that would be been immensely helpful is a review devoted entirely to requirements. This is something that ended up hurting us later in the project when we had to descope the complexity of our project and it felt as if we didn't have a map that we should have had available to us. Going over this will help to clarify that all your design choices have a strong purpose within your project.

Develop clear mission off ramps and minimum success criteria, and build a schedule that establishes these goals first – Pick out pieces of your system that can be simplified (off ramps) and still achieve your main mission success goals. Having these plan B and C options will ease the burden of major design choices when you're running out of time or money and need to hit a major deadline (Thermal-Vacuum or Integration for launch for example). You also want to very clearly establish what will achieve your minimum success criteria. Building this first will ensure

you can have an effective T-VAC and be approved for flight, after which more complexity can be added with this strong base.

Team communication. One of the first things the project manager of a team should do is develop an easy way for the team to communicate with each other. We used Google Groups and it worked really well once people learned to use the "reply all" button.

Know team member backgrounds and expertise – Managers need to be sure they are assigning tasks to the personnel with the proper expertise, and know who is best fit for a job. If something is more challenging, assign multiple people you believe will effectively work together. Knowing this will also help to decide subsystem team leads.

Challenge your team members – Don't be afraid to assign tasks to members who have no prior experience in a subject. The purpose of Space Grant and the HASP program is to learn and develop work experience with topics you've never experienced before.

Delegate team member accountability and responsibility to individuals and team leads – As a project manager or team lead, don't take on all the responsibility of ensuring deadlines are met and work gets done; utilize the other leaders on the project. You also want to make it clear how each team member's work is affecting the team, and how important their work is.

Take pictures and video. In the moment taking the time to take pictures or video may seem like an unnecessary use of time, but in hindsight it is one of the most helpful things you can do. Pictures are great for documenting the status of the system before and after changes, e.i. electrical boards. Pictures can sum up an entire test, something that is very useful when documenting what happened during the test later. Videos are great to have as a memory of what you accomplished and are fun to show off.

Lessons learned from the beginning. In writing this document now, I wish I would have started it a long time ago. It is difficult to take the time to document what is happening when lots of things are going on, usually in the most stressful part of the project around integration, but keeping a log of lessons learned when they happen is invaluable. A lessons learned document should be the first file saved to the team folder.

Clearly establish weekly and long-term goals – In scheduling, develop long term goals and use the weekly team meetings to create the building blocks to get there. This will help to make major goals more manageable.

Schedule aggressively, have wiggle room. Staying on schedule is always a challenge for new and young projects, so anticipate this. It is easier said than done, but if a schedule can be laid out that gives more time than necessary for tasks to be completed there will be some wiggle room when something goes awry.

Inform launch operators about special requirements early – If your project has special requirements (needs more power, faster communication rate, breaks the physical limitations of the payload space), tell your launch personnel ASAP! They will appreciate being informed early and are going to be more willing to work with you to develop a solution. They want to help you succeed!

Keep people busy – Don't ever think you're ahead of schedule; setbacks will occur! There is always work to be done and a team member should never be saying they have nothing to do. You can always be thinking ahead on how to simplify a task in the future and working to catch potential problems early.

Think of how things will integrate during design. This is a classic lessons learned for mechanical systems, but is applicable to all subsystems. It is easy to design a hole for a bolt and later realize that the way it was designed, you cannot insert the bolt into the hole, or there is not enough room for a tool to get within a the gap to screw in the bolt (happened to us). When you are designing a part or component thing of how it will attach to the structure, and if it will need to be accessible during testing. Do not be afraid to make cheap cardboard mock-ups of the system to physically represent where things will go.

Design wiring harness for electronics within the structure – Something that caused the HASP 2010 team some trouble and was entirely overlooked was a proper wiring harness method. You're going to want to spend a lot of time simplifying this, and proper organization ease payload accessibility and electrical integration and deintegration.

C. Fabrication Phase

Minimum mission success is still success. When designing and fabricating a system keep in mind what is the minimum that you need to do to fulfill the mission's goal. Keep It Simple. If there are additional capabilities that the system *could* perform, but does not necessarily *have* to, make them a "stretch goal" and prioritize them lower than things that are *needed* for mission success.

Team member accountability, especially for team leads – Have a strong method to go over team member's work they declare complete. Team members can says they have a task complete but team members need to hold each other accountable for getting their work done. Demodays are good way to present proper functionality.

Centralize team files. Colorado Space Grant uses a system server as a central location of files for projects. This has been very helpful in keeping files accessible and easy to reference across subsystems. Having multiple file locations can get confusing, and should not be the cause of something going wrong on a project.

Demo-days. When subsystems are in the middle of developing their subsystems we found it beneficial to have the subteams present hardware that they are working on. Not only is it cool, but it provides mini-deadlines for tasks and shows the rest of the team that things are getting done which can improve accountability.

Very clearly lay out independent and dependent tasks for each subsystem – Distinguishing these will greatly help in scheduling and when specific subsystems need to have a task complete. This will help team wide communication and what work a subsystem can continue working if they are waiting for another team to complete a task.

Have backups ready at-hand. This is limited by budgetary constraints, but whenever possible have backups pre-purchased for things like electrical components, mechanical tools, etc. A major obstacle in some projects is the lead time associated with waiting for parts to ship.

D. Integration & System Testing

Document testing procedures. When testing the system it is easy to want to change something during the test because you find that something else is not working correctly, or the test was laid out like you thought it would be. These changes are alright, but make sure you document them. In the moment they might seem like major things that will be easily remembered, but months down the road it is helpful to have what happened during the test in writing so you can bring it up and use it as a reference.

Test early, test often. When building something, do not wait for the complete product to test its functionality. As much as you can, test incrementally. This way you know what works, and what was added that caused something to stop working. Document this along the way, too.

During integration, something will not work. Very rarely does something work with full capabilities the first time it is put together. Even if all the subsystems work independently, plan for something to go wrong during integration. If the project makes it through integration, something could break during testing. Have backups once you know it works if possible, and budget time for these things. It is a tradition of the Colorado HASP team to finish integration hours before thermal vacuum testing, but this is extremely stressful and is not recommended.

Ensure subsystem functionality before integration. This goes along with test early and test often and integration failures, but if you can ensure the functionality of a subsystem before it is combined with another piece of the project it will drastically reduce the time needed to conduct troubleshooting if something does not work when the payload is integrated.

Before someone leaves the project, make sure their work is well documented and knowledge is sufficiently transferred – This will become especially important if someone has

an expertise in something (usually software or electrical). The best way to maintain this is if you know a team member will no longer be available is to have someone shadow their work.

Document the flight configuration of the payload. As close to final integration as possible, document everything you have time for about the payload. Where are the sensors located? In what direction are they pointing? Finalize the list of uplink commands and make sure you have the most up-to-date copy. Are all bolts in place and secure? Save a copy of the flight code and make it read-only. Anything you can do to archive the status of the system before flight will help if questions arise during post-flight processing and may help future teams in design.

Keep a log of the "status of the system." When working with a team everyone is not always together when work is done on the experiment. If you can keep a log near the payload that documents who worked on what when and the status of the system before and after the work was done troubleshooting issues will be a lot easier. This becomes especially important the closer you get to launch.

Perform Day In The Life (DITL) testing. Once the payload is integrated and working in a room temperature, standard pressure environment, test it under flight conditions (or as close as you can get to flight conditions). We did a lot of cold testing with dry ice in a thermal chamber, and if you can vacuum test your payload then you will be in a great place. It is nice to finish integration early, because then there is more time to conduct DITL testing. DITL testing is great for highlighting unforeseen problems could arise and might need fixes, we found out that under flight temperatures some wires were freezing and preventing our telescope from rotating.

System test pre and post shipping (from SG to Integration or Thermal Vacuum testing) – Develop a checklist of all the functions of your system prior to any payload movement or travels. It is possible for components to become disconnected, and you'll want to know if any changes occurred.

Maintain the big picture, continually look at requirements – It can be easy to get lost in system testing, and it is always good to double check that your actions are making sense.

Know the testing schedule for thermal vacuum testing in Texas. Before going to Palestine for integration and testing, know what the test is going to be like. What temperatures will be tested? What will the pressure profile look like for the tests? Are there things that you want to test when it is hot or cold? Knowing the schedule for the thermal vacuum testing will help you get the most out of the test. Request this information from the HASP program early if it is not given to you.

E. Flight/Post-Flight

Make verification methods internal. It is a good idea to make sure that you can verify things about your system independently from the HASP platform. It is nice to use the CosmoCam as a visual verification for things such as movement, but what happens if the CosmoCam stops working? Depend on things that are not in your control as little as possible.

Develop a strong con-ops plan for flight and system control – Know exactly what needs to happen when, you don't want to waste any flight time developing a plan of action. Anticipate what can go wrong, and what you can do to fix it.

Develop post-flight analysis early – Something to ease the data analysis process is to have team members develop a post-flight analysis plan early. Good to keep team members busy, and relieve some work after the long hours from launch and final integration.

F. General

Don't be afraid to ask questions if you don't know something – Never be afraid to admit you don't know something or need help. Take advantage of the team members around you; time is extremely valuable and the faster you can become comfortable with your task the better.

Schedules vs. Lists. Having a schedule to outline the timeline of project is a great way to keep the big picture in mind. That being said, sometimes when a lot of things need to be done simultaneously it is more useful to use a burndown list to prioritize. Used a schedule throughout the project up until "crunch time" during integration and before launch (where we used many burndown lists).

Hold team member accountable through team-wide presentations – Have team members show off and present their hard work; it will motivate them to reach deadlines and give them an opportunity to be proud of what they've done.

Do not forget why you are doing this. There will be late nights and times where you have to sacrifice hanging out friends to finish the project, but remember how fun it is to be working with your hands and designing something that is going to space! Plus, launch makes it all worth it.

VII. Conclusion

In just under a three year orbit, Kepler has discovered approximately 2,300 star candidates which hold a high probability for supporting a planet orbiting that star, with 30 confirmed planets within its 10˚ by 10˚ field of view of the Milky Way. One of these 30 is the promising Kepler-22b, which is the first discovered exoplanet to orbit a sun-like star within its habitable zone. Kepler is truly changing the way space exploration is viewed, and could very well change the direction of the space program with the information it has been able to uncover. The high cost of placing Kepler in orbit is one of the major concerns for being able to reproduce this mission, but balloon based observatories offer a strong alternative to this challenge. Balloons offer the reproducibility of the Kepler spacecraft for a fraction of the cost, with minimal interference from the Earth's remaining 1% atmosphere at 120,000 feet.

The findings of the SPARTAN-V payload proved the balloon environment is ideal for night and day observations. The gentle float environment was characterized to have a maximum instantaneous yaw spin rate of 2.28 degrees per second, with an average yaw spin rate of 0.89 degrees per second verified by gyroscopic sensors and streaked star images. A control system would then need to compensate for spin rates of upwards to 2.5 degrees per second to maintain a star within its field of view of 2/3 of a degree. The streaking star images proved that pendulum motion of the HASP platform is negligible, and any controlled pointing system would primarily need to compensate for yaw rotations only. The HASP platform proved to spin in a single direction for a maximum of 30 minutes, and oscillations occurring on average every10 to 15 minutes. Analyzing approximately 65 individual stars presented the brightest star to have a signal to noise ratio of 122:1.

Future missions in this family of balloon-borne optical systems include the next Colorado HASP project, HELIOS, and the current University of Colorado Aerospace Senior Design Project, DayStar. HELIOS has the goal to design and implement a balloon-borne optical tracking sensor in order to improve the capabilities of payloads such as SPARTAN-V. An additional goal of the team is to improve upon the HASP platform characterization data collected by the SPARTAN-V team. DayStar is a project working towards creating a balloon-borne optical pointing system capable of tracking stars during the day. Both projects have the potential to greatly improve the capabilities of balloon-borne optical systems and increase the knowledge about these systems as the SPARTAN-V payload has done.

VIII. Appendix

Table 7: Serial Uplink Command List

ID	Command Action	Command Value	Description	Critical?	Confirmation Method	Contingency Plan
1	Restart system	01 ; 00	Kills all processes simultaneously	No	Downlink	Cycle power
2	Kill Init Process	01 ; 01	Kills CDH system	No	Downlink	Cycle power
3	Kill Ctl process	01 ; 02	Kills control processes	No	Downlink	Cycle power
4	Kill Pwr Process	01 ; 03	Kills power relate processes	No	Downlink	Cycle power
5	Storage Status	01 ; 13	Displays usage of solid state and flash memory	No	Downlink	Cycle power, set to default
6	Uplink & H&S status	01 ; 14	Displays last two uplink commands sent, # of successful uplink commands	No	Downlink	Cycle power, set to default
7	Decrease H&S read size	01 ; 15	Decreases amount of data transmitted in H&S	No	Downlink	Cycle power, set to default
8	Increase H&S read size	01 ; 16	Increases amount of data transmitted in H&S	No	Downlink	Cycle power, set to default
9	Serial attempts --	01 ; 17	Decreases # of attempts set to run uplink commands by 1, default is set to 7	No	Downlink	Cycle power, set to default
10	Serial attempts ++	01 ; 18	Increases # of attempts set to run uplink commands by 1, default is set to 7	No	Downlink	Cycle power, set to default
11	H&S decrease filesize	01 ; 19	Decreases data rate of read from other processes	No	Downlink	Cycle power, set to default
12	H&S increase filesize	01 ; 1a	Increases data rate of read from other processes	No	Downlink	Cycle power, set to default
13	Reset AVR	02 : 06	Resets AVR processes	No	Downlink	Cycle power
14	Initiate image capture	01 ; 1c	Begins image capture process	Yes	Downlink, data storage	Resend command; Cycle power
15	Rotate telescope CW	02 ; 01	Moves motor from stowed position to viewing angle	Yes	CosmoCam	Resend command; Cycle power
16	Rotate telescope CCW	02 ; 02	Stows telescope	Yes	CosmoCam	Resend command; Cycle power
17	Extend linear actuator	02 ; 03	Locks telescope	Yes	CosmoCam on descent	Resend command; Cycle power
18	Retract linear actuator	02 ; 04	Unlocks telescope	Yes	CosmoCam (if telescope moves)	Resend command; Cycle power
19	Turn off linear actuator	02 ; 05	Pauses movement of linear actuator between use	Yes	Decrease in current pull	Resend command; Cycle power

Table 8. Flight star values.

Balloon-Star Brightness (photons)	Associated Error (Photons)	Signal to Noise Ratio
3.90624513e+04	4.87134492e+02	8.01882271e+01
4.52178189e+04	4.89599095e+02	9.23568269e+01
3.90897415e+04	4.84216559e+02	8.07278082e+01
3.79416528e+04	4.74751927e+02	7.99189023e+01
4.07891124e+04	4.80956741e+02	8.48082768e+01
4.65128051e+04	4.83218052e+02	9.62563482e+01
5.08257073e+04	4.85429476e+02	1.04702557e+02
5.30688906e+04	4.88288325e+02	1.08683513e+02
4.16778919e+04	4.43877078e+02	9.38951209e+01
4.59616136e+04	4.47212615e+02	1.02773518e+02
1.41570227e+04	4.23234289e+02	3.34496119e+01
1.19627062e+04	4.19970386e+02	2.84846421e+01
1.39248663e+04	4.17675325e+02	3.33389727e+01
1.15288523e+04	4.16390107e+02	2.76876229e+01
1.24299337e+04	4.22397476e+02	2.94271022e+01
5.36051489e+04	4.67369915e+02	1.14695335e+02
5.05048467e+04	4.68995713e+02	1.07687225e+02
5.58613073e+04	4.67379096e+02	1.19520338e+02
5.77320643e+04	4.71953480e+02	1.22325752e+02
3.19935516e+04	4.44544379e+02	7.19693086e+01
3.24665688e+04	4.44825784e+02	7.29871557e+01
2.64700039e+04	4.42396934e+02	5.98331539e+01
2.75011373e+04	4.39343298e+02	6.25960097e+01
2.97390538e+04	4.43372835e+02	6.70745959e+01
3.08278887e+04	4.41998751e+02	6.97465516e+01
2.67574110e+04	4.39938722e+02	6.08207682e+01
2.68999818e+04	4.40002262e+02	6.11360079e+01
2.57690508e+04	4.39668486e+02	5.86101838e+01
2.62813784e+04	4.41598520e+02	5.95141905e+01
2.80038286e+04	4.41668006e+02	6.34047027e+01
3.03480043e+04	4.43961306e+02	6.83573184e+01
2.76856768e+04	4.41774307e+02	6.26692778e+01
2.37393159e+04	4.33893197e+02	5.47123488e+01
2.90961961e+04	4.42413553e+02	6.57669638e+01
3.22055759e+04	4.48304013e+02	7.18386965e+01
3.27354749e+04	4.46113602e+02	7.33792350e+01
3.21194484e+04	4.46383315e+02	7.19548588e+01
5.69059799e+04	4.71238851e+02	1.20758252e+02
4.87384275e+04	4.65499285e+02	1.04701401e+02
4.41318900e+04	4.60982066e+02	9.57345052e+01
4.25269725e+04	4.56149741e+02	9.32302897e+01
4.35107747e+04	4.58426497e+02	9.49133067e+01
4.21139122e+04	4.58709545e+02	9.18095396e+01
4.63822066e+04	4.63411958e+02	1.00088497e+02
3.24384655e+04	4.41237222e+02	7.35170649e+01
3.41135047e+04	4.41326016e+02	7.72977424e+01
3.46149180e+04	4.41656815e+02	7.83751475e+01
3.49223194e+04	4.45908542e+02	7.83172245e+01
3.38620679e+04	4.50154847e+02	7.52231551e+01
2.99057630e+04	4.47496079e+02	6.68291062e+01
3.03324607e+04	4.47694096e+02	6.77526484e+01
3.20215289e+04	4.46888795e+02	7.16543562e+01
3.21535426e+04	4.50561415e+02	7.13632848e+01

2.36875062e+04	4.31883565e+02	5.48469729e+01
2.06719558e+04	4.28070443e+02	4.82910142e+01
1.62186278e+04	4.28499168e+02	3.78498466e+01
2.09096321e+04	4.31329277e+02	4.84771918e+01
1.72948776e+04	4.31990203e+02	4.00353469e+01
1.96996272e+04	4.29558185e+02	4.58602067e+01
2.47055085e+04	4.31080561e+02	5.73106532e+01
2.40088505e+04	4.34099644e+02	5.53072337e+01
2.38423998e+04	4.35136583e+02	5.47929106e+01
2.32864977e+04	4.36008191e+02	5.34083950e+01
3.01899789e+04	4.46542873e+02	6.76082427e+01

Table 9. Ground star values.

Ground- Star Brightness (Photons)	Associated error	Signal to Noise ratio
1.19110669e+04	4.16823020e+02	2.85758374e+01
1.42232151e+04	4.12256963e+02	3.45008487e+01
1.24463497e+04	4.12369342e+02	3.01825291e+01
1.39077254e+04	4.06139160e+02	3.42437441e+01
7.68881605e+03	3.99086028e+02	1.92660617e+01
8.52110731e+03	3.98896696e+02	2.13616894e+01
1.35711026e+04	4.04619556e+02	3.35404022e+01
1.01975861e+04	3.99244806e+02	2.55421885e+01
8.72091862e+03	3.97025732e+02	2.19656257e+01
1.15986663e+04	3.91886812e+02	2.95969804e+01
1.05482632e+04	3.89739664e+02	2.70648953e+01
1.21441324e+04	3.87942301e+02	3.13039654e+01
6.51777098e+03	3.87007875e+02	1.68414428e+01
1.04036666e+04	3.89427685e+02	2.67152722e+01
7.72795296e+03	3.90113600e+02	1.98094939e+01
8.51726979e+03	3.89794823e+02	2.18506488e+01
4.33341463e+03	3.90629444e+02	1.10934153e+01

Table 10. Student team members.

Name	Standing	Major	Ethnicity	Age	Sex
Akash Agrawal	Grad	ECE	Asian	25	M
Alex Harvey	Undergrad	ASEN	Caucasian	21	M
Bear Sawicki	Undergrad	ECE	Caucasian	21	M
Brian Ibeling	Undergrad	ECE	Caucasian/Asian	20	M
Bryan Barnhart	Undergrad	MATH/PHYS	Caucasian	19	M
Chris Nie	Undergrad	ASEN	Caucasian	20	M
Eddy Scott	Undergrad	ASEN	Caucasian	19	M
Irene Chen	Grad	ECE	Asian	22	F
Jackie Myrose	Undergrad	CSCI	Caucasian	19	F
Josh Tiras	Undergrad	ASEN	Caucasian	19	M
Josh Yeaton	Undergrad	ASEN	Caucasian	20	M
Kevin Wong	Undergrad	ASEN	Asian	19	M
Maulik Kapuria	Grad	ECE	Asian	24	M
Nate Lapinski	Undergrad	CSCI	Caucasian	19	M
Sreyas Krishnan	Undergrad	MCEN	Asian	20	M
Sushia Rahimizadeh	Undergrad	ECE	Asian	21	M
Tyson Sparks	Undergrad	ASEN	Caucasian	20	M
Venkat Janakiraman	Grad	ECE	Asian	24	M
Vignesh Muralidharan	Undergrad	EE	Asian	18	M
Jeff Bryne	Undergrad	MCEN	Caucasian	20	M
Carly Smith	Undergrad	APPM	Caucasian	21	F
Anthony Cangelosi	Undergrad	ASTR	Caucasian	22	M

APPM – Applied Math
ASEN – Aerospace Engineering
ASTR - Astronomy
CSCI – Computer Science
ECE – Electrical and Computer Engineering
EE – Electrical Engineering
MATH – Mathematics
MCEN – Mechanical Engineering
PHYS - Physics

VIII. References

[1] "Kepler, A Search for Habitable Planets." NASA. Retrieved from: http://kepler.nasa.gov/Mission/discoveries/

[2]Romanishin, W. "An Introduction to Astronomical Photometry Using CCDs." University of Oklahoma, 2006

[3]"Understanding CCD Read Noise." Quantum Scientific Imaging. Retrieved from: http://www.qsimaging.com/ccd_noise.html