PLEASE Science Report

HASP 2013

Louisiana State University
12/13/2013

Table of Contents

1. Mission Overview

1.1 Mission Goal

The goal of PLEASE was to develop a daytime passive attitude determination system (PADS) that could determine orientation in three dimensions. The PADS consisted of a camera with wide angle lens, dual-axis tiltmeter, and GPS as well as an Arduino Mega for systems control. Images were taken of the Sun and analyzed at LSU. The goal was to locate the centroid of the sun in each image as well as correct for known tilt.

1.2 Objectives

The objectives for the flight were as follows:

1) Develop and fabricate a PADS capable of finding pointing direction accurate to within 20 arc minutes for every second of flight
   a) The project was adjusted to using every 2 seconds instead of every second. We were able to find pointing direction within the desired range.
2) Be able to correct for gondola motion post-flight
   a) We were able to correct for gondola motion as explained in section 3.2 Tilt Correction.
3) Output time-stamped pointing data for post-flight analysis.
   a) Our payload output time-stamped data that was able to be analyzed post-flight and converted into orientation.

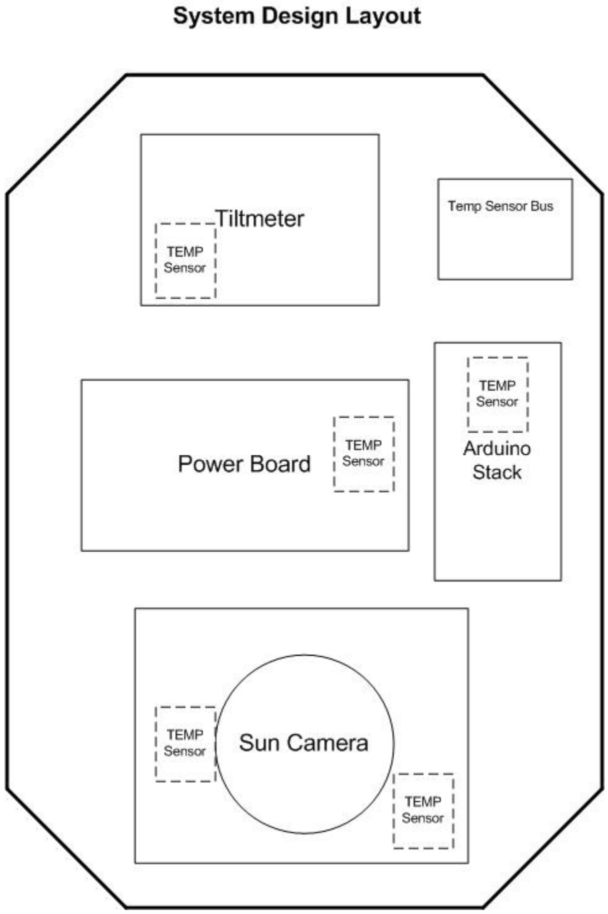
2. Payload Design

2.1 Operation Principles

The PLEASE payload needed to be able to function in temperatures ranging from ±80^{0}C. It also needed to be able to function at pressures as low as 1-5 mbars. The payload was required to run on 29V to 33V DC and a maximum of 2.5A and be a maximum of 20kg. The dimensional limits were 38cm by 30cm with a height limit of 30cm. The payload had a serial downlink running at 4800 baud.

During flight PLEASE would take pictures of the Sun and save them on the SD card of the camera. It would also collect, save, and transmit GPS data, tiltmeter data, and temperature. Each of the four tasks would be completed for each cycle which would occur every two seconds controlled by the Arduino.

2.2 Subsystems



2.2.1 Thermal System

Because the payload would be in an extreme environment, there were necessary steps taken to ensure proper functionality. In order to stay within the temperature bounds of our components a heating system was put in place to counter the cold extremes and insulation was used to counter the hot extremes. The temperature was measured using a system of digital temperature sensors as explained below.

2.2.1.1 Temperature Measurement

The temperature was measured using five digital temperature sensors. The sensors were placed in different locations on the most temperature sensitive components such as the tiltmeter. The sensors were chosen because they allowed the use of the OneWire Arduino library and wiring. OneWire allows the data from all of the sensors to be transmitted through a single data line. Each sensor had its own unique 8 byte ROM address which allowed us to differentiate each sensor from the others.

An OneWire bus was created to facilitate the temperature sensors. The V_{in} used for the sensors was 3.3V, which was provided by the 3.3V pin of the Arduino. The ground pin of the Arduino ground pin was also used. The output line was connected to a digital input pin of the

Arduino. The most important piece of using OneWire is to include a 4.7kΩ pullup resistor which runs from the V_{in} to the output line in parallel with the sensors.

The software of the temperature sensors is a modified form of the OneWire Arduino library. Each loop through the code reads one sensor and converts the data into degrees Celsius. The code will cycle through all connected sensors until it can't find one at which point it will restart at the first sensor.

2.2.1.2 Heating System

The heating system consisted of four 30V to 15V regulators and eight resistors. The regulators had heat sinks attached to them which could then in turn be connected to other components to provide heat through contact. The resistors were also large enough that copper could be safely wrapped around them and connected to other components to draw heat from the resistors. This was important because the near vacuum experienced at float altitude would limit the amount of heat transferred through radiation.

The heating system could be turned on or off using analog signals from the HASP platform through the use of dual-coil locking relays.

2.2.2 Sun Camera

The central focus of our payload was the camera. It would need to be able to take high quality pictures of the Sun. The camera and accessories used in our payload would have to be carefully selected because they would not be entirely within our outer shell and therefore partially exposed to the environment. We chose a 24 megapixel camera and a wide-angle lens with a FoV of 109^0. We used a neutral density filter over the lens to darken the images which prevented the sun from damaging the CCD, but also to prevent image washout. The filter did not affect the optics in any other fashion.

Because of the large number of pixels we chose a lower quality of image within the settings of the camera to ensure we would not run out of storage in the camera's SD card. The maximum size available was 128GB which, by our calculations, would fill in 1/3 of the total float time at maximum settings. A number of other settings were determined through testing including LOW ISO, manual shooting mode, manual camera focus with automatic lens focus. The last two settings prevent the camera from changing the focus during flight. The camera's display and "Photo Auto-Review" settings were turned off to reduce the normal current load of the camera to 0.3 amps.

The most important setting of the camera was the focus. Because of the distance of the image target, the Sun, the focus was set to "infinity." Once the focal length was set, the zoom was set to its widest setting to allow for the full 109^0 FoV.

The camera chosen was rated to run on 7.4V-9V. While taking pictures, the current draw of the camera spiked to 2.5amps, the limit allowed by the HASP platform. It was determined that the spike was too quick to trigger a shutoff by the HASP platform. To power the camera a variable output DC-DC converter was used to provide 8V to the camera. This was done in case the power from HASP fluctuated so that the camera would remain in a safe operating range.

The camera was housed in an aluminum structure which was secures to the payload. Between the camera and the frame was foam insulation to prevent damage to the camera from the aluminum.

To have more direct control over the camera trigger additional wiring and code were needed. A remote camera trigger was attached with its internal wires connected to the Arduino. The remote wire contained three wires: ground, focus, and shutter. The camera could be triggered by grounding all three wires. To do this the ground wire was connected to a ground pin of the Arduino and the focus and shutter pins connected together to a digital output pin of the Arduino. The focus/shutter pin was set to High which provided 5V to those wires. To trigger the camera the focus/shutter pin was set to Low for 1ms then raised back to High.

2.2.3 Tiltmeter

The tiltmeter used for our payload used an output of -2.5V to 2.5V at 4 degrees/volt and could show ±10 degrees of tilt. It was powered by 9V. For the tiltmeter to be compatible with the Arduino it needed to be adjusted to output 0V to 5V which was accomplished using a level shifting circuit. The circuit consists of the output from the circuit entering an AD820 op amp which halves the tiltmeter output. This then enters another op amp which has a gain of two and adds in a reference voltage in order to level shift the output up to 0V. Potentiometers were added into the circuit in order to tune the gain so that the final output is exactly 0V to 5V. This circuit was repeated for the X and Y outputs of the tiltmeter. The op amps required ±12V to function. . Using a REF-02 chip and a +12V input, the reference voltage of -5V to create the level shift was created. The reference voltage of -5 is added to the inverting input of the AD820 and in essence, level shifts a positive voltage.

2.2.4 Outer Shell

The outer shell of our payload was designed to protect the payload from the external environment. The considerations taken in its design were structural support, size, weight, and location of any parts that would need to protrude from the box. Parts that needed to be out of the box were the HASP power and serial wires, the GPS antenna, and the lens of the camera. The outside of the box was covered with “Appliance White” enamel to reflect some of the heat from the Sun and keep payload from overheating.

The inside of the shell contained a layer of insulating foam also meant to prevent the payload form overheating due the Sun.

2.2.5 GPS

A GPS was chosen that could function at the float altitude of approximate 125,000 ft. The LSU electronics shop designed a GPS shield for the Arduino Mega with the chosen GPS built in. A separate antenna was required for use with the GPS. The requirements for the antenna were that it be weatherproof and able to withstand a vast temperature range because it would be placed outside of the shell of our payload.

The code used allowed the Arduino to read the data from the GPS and parse the NMEA data. This was a requirement to make the GPS data easily readable.

2.2.6 Control System

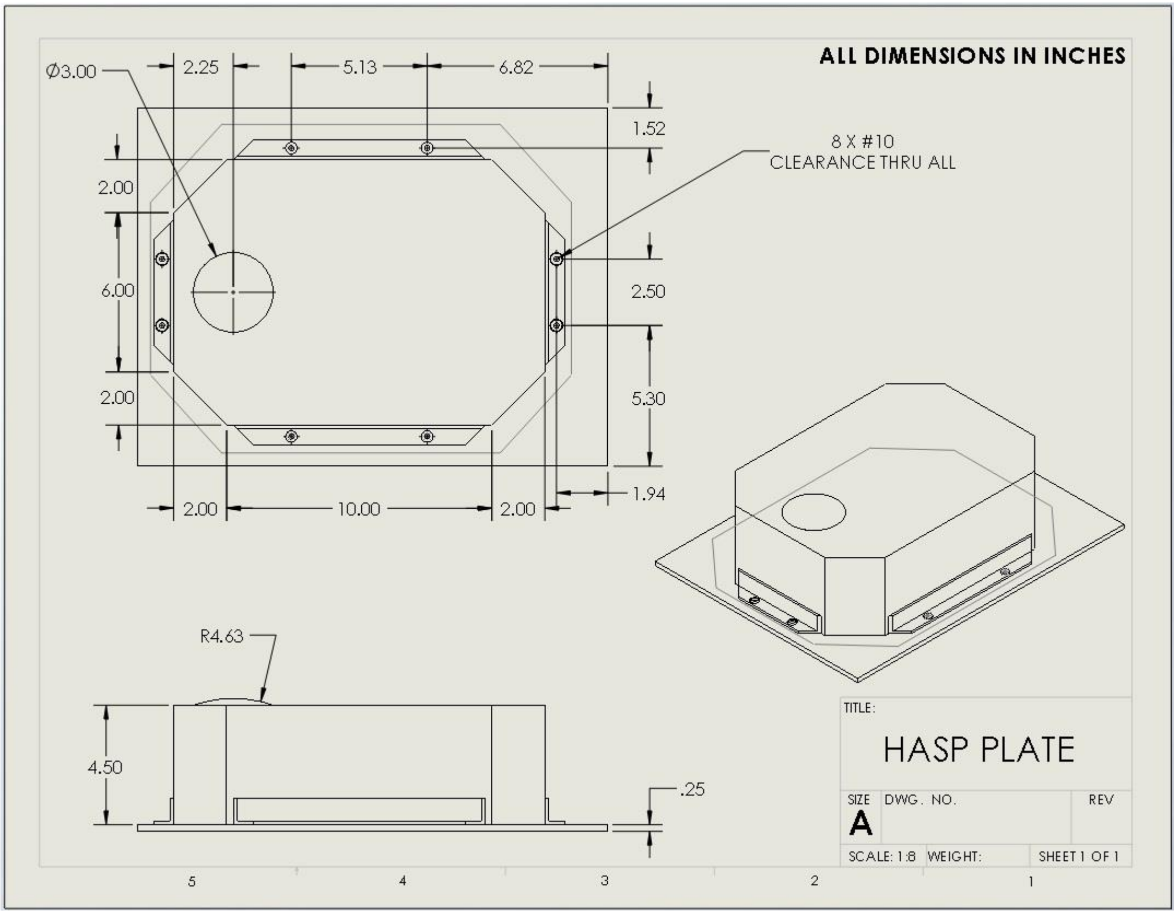
The control system for the payload was the Arduino. The primary loop of the code contained each of the systems and allowed for easily connecting data from the systems with each other. To be sure that data properly lined up the first input had to be the temperature sensors. Each loop through read the next sensor until it found no more, at which time it returned an error statement. To avoid having data from the other components match with the error statement, the loop was exited at the error statement.

The primary purpose of the loop was to be sure that the tiltmeter, GPS, and camera were triggered within a second of each other. This was to ensure that the tiltmeter and GPS data aligned properly with pictures taken. This was important because the images were stored on a separate SD card from the other data taken.

2.3 Mechanical Specifications

The following table shows the measured weight of the payload components.

Component	**Weight (g)**
Sun camera + lens	1020
Tiltmeter	555
Arduino + shields	145
Housing Structure	4800
Foam insulation	200
Electronics	500
Temp Sensor Array	10
Total	**7852**

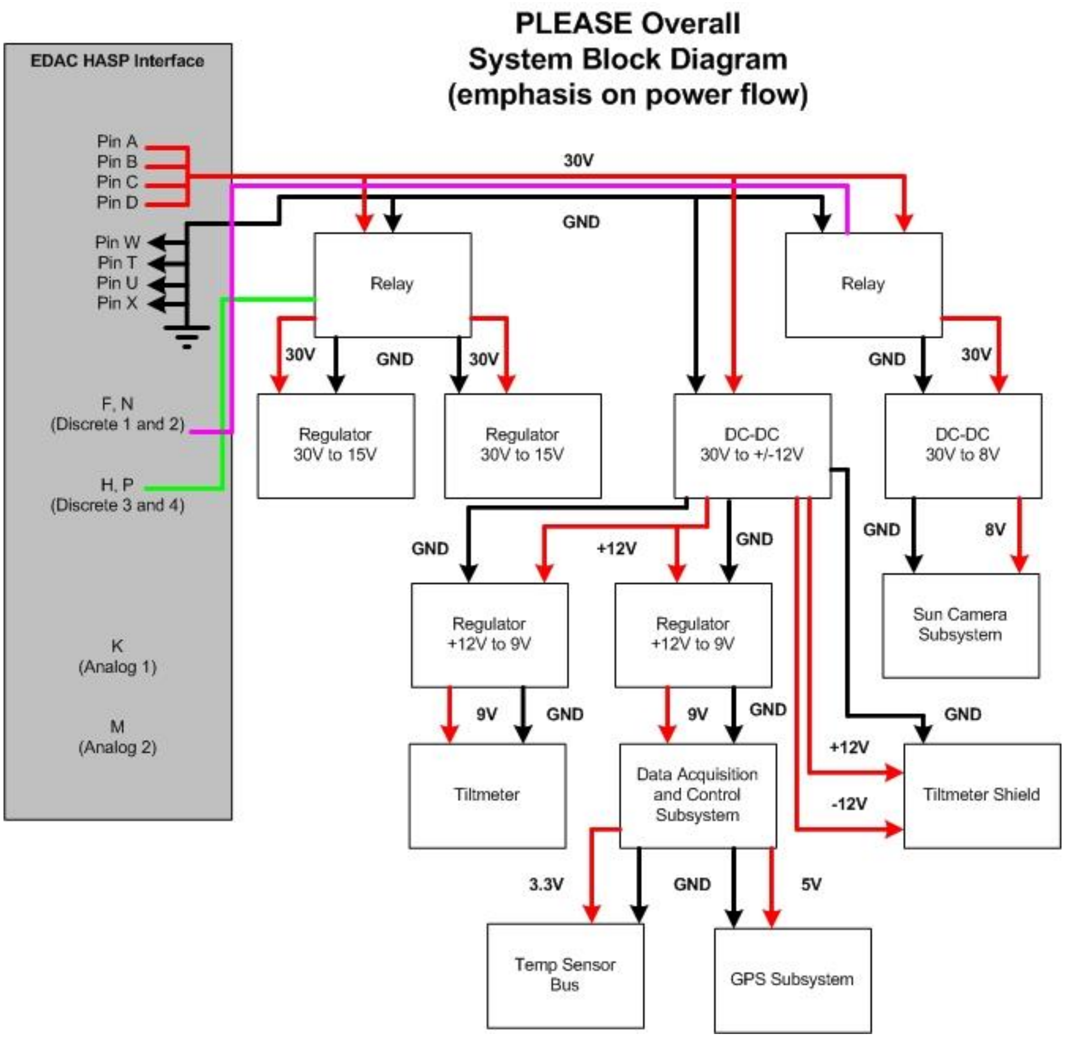


Above: The design specifications of the aluminum outer shell.

2.4 Power Specifications

The table below is of the power requirements for our components. The power drawn by the "Arduino Mega Stack" consisted of the Arduino, the tiltmeter conversion board, the GPS shield, the TTL to RS232, and the temperature sensors.

Component	**Part**	**Voltage (V)**	**Current Draw (mA)**	**Duty Cycle (%)**	**Power (W)**	**Power Consumed (Amp-hrs)**
Tiltmeter	A904-T	9	100	100	0.9	1.6
Sun Camera	Nikon D3200	8	100	99	0.792	1.584
Arduino Mega Stack	ATmega2560	9	100	100	0.9	1.6
Heater system		30	750	100	22.5	12
Minimum Current (30VDC)			1050			
Max Current (30VDC)			1550			
Total					24.6	16.8



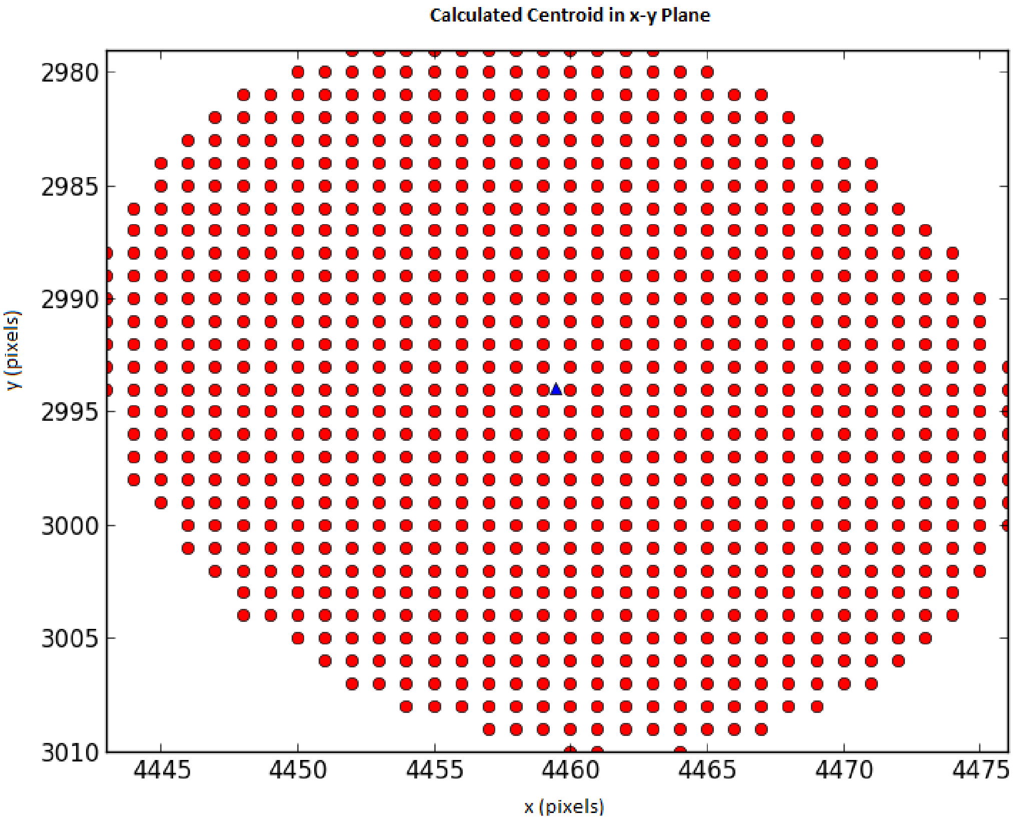
3. Image Analysis

3.1 Locating Centroid

The first step to finding the centroid of the Sun in an image was to separate each pixel into an x and y coordinate. Each pixel was scanned to see if it was above the chosen threshold. The Sun was located using luma and chrominance; which together can be used to represent brightness of an image. Each pixel of a given image is converted into RGB components and then used to compute luma using the following: $luma = (0.3 * R) + (0.59 * G) + (0.11 * B)$ where R, G, and B represent the red, green, and blue components respectively. There was the concern of reflection from surfaces on the HASP platform and from the balloon. These were seen, but were able to be found and removed from calculation. Reflections had higher concentrations of the blue component of the RGB spectrum and could be filtered out based on this.

To locate the centroid of the Sun in an image I used a center of mass calculation. For either direction the centroid was calculated by summing the number of pixels in the direction time the luma and dividing by the total luma. The centroid in the x-plane is shown below:

$$centroidX = \frac{\sum(luma * Xpixel)}{\sum luma}$$

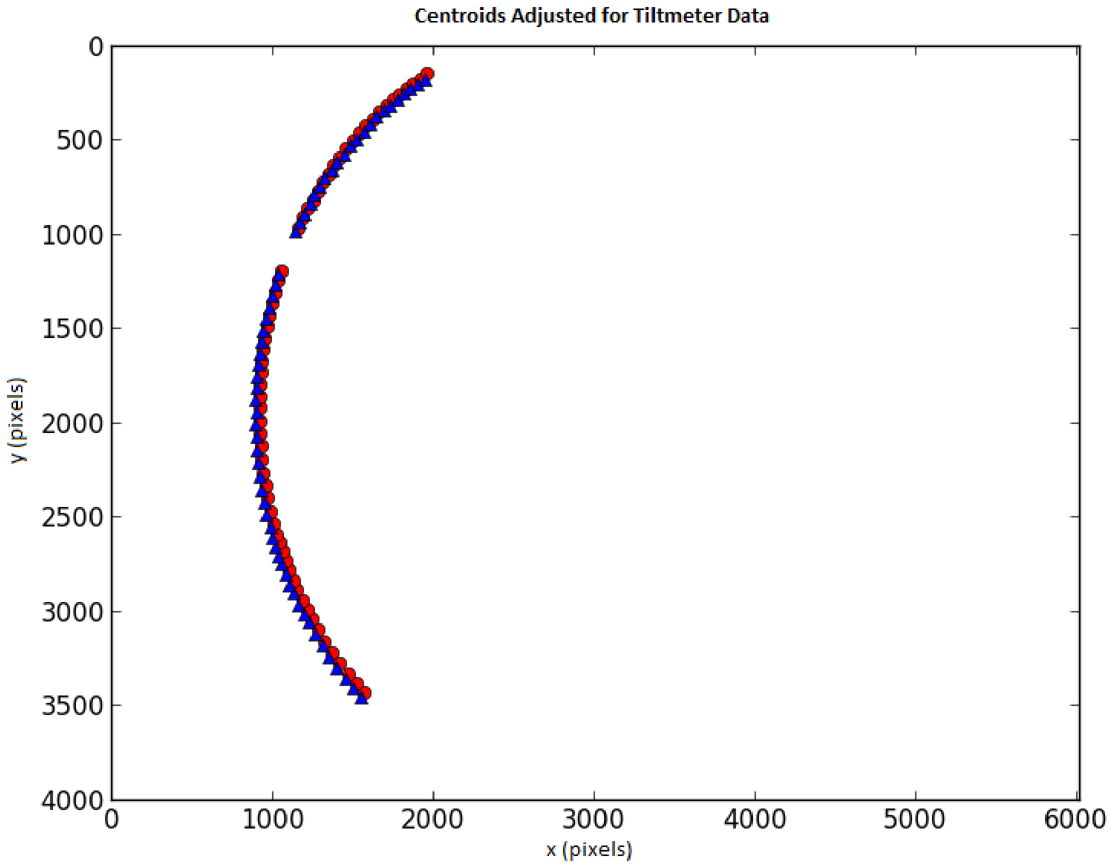


Above: An example of the centroid location method used. The red dots represent any pixels that meet the luma criteria. The blue triangle is the calculated centroid of those pixels.

3.2 Tilt Correction

After locating the centroid the next step was to correct for the tiltmeter measurements. The central formula for these corrections is: $\tan\theta = \frac{l}{\frac{l}{x}} * f$ where, for a given direction *l* is the length of the CCD plane; *x* is the distance of the centroid from the center of the image; and *f* is the focal length. To get *x* half the length of the CCD plane is subtracted from the centroid. This sets the center of the image as zero and follows standard x, y coordinates for each quadrant; up and right from center is positive and the quadrants proceed counter-clockwise. It was important to insure that images remained in the proper quadrant because, for example, without using negative or positive images on two centroids on opposite sides of the y-axis could share a θ. This would cause problems when trying to use θ to find the new centroid location.

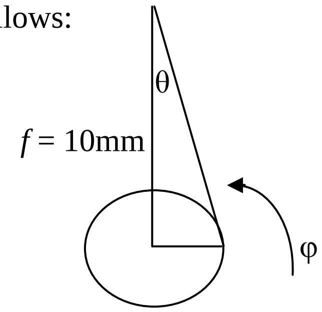
The angle offset found by the tiltmeter was then added to the θ found using the formula above. The formula was then manipulated using the tiltmeter+θ as the angle and finding a new *x*. Half the length of the CCD plane is added to the new *x* and this number is the corrected centroid.



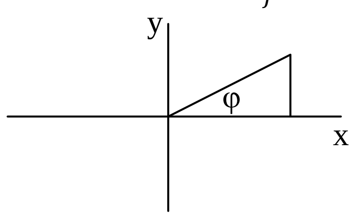
The above plot shows the tiltmeter correction for images over a two minute time span. The red dots are the originally calculated centroid and the blue triangles are the tilt-corrected centroid

3.3 Spherical Coordinate Calculation

The information found was also converted into polar coordinates of r, θ, and φ shown as follows:



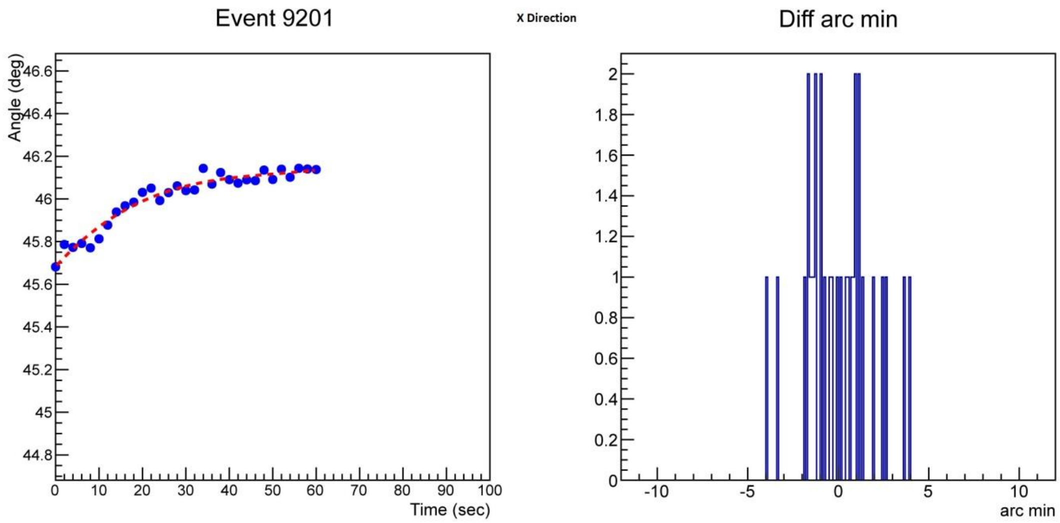
For this calculation each centroid was converted to millimeters by multiplying the centroid (in pixels) times the CCD length in $\frac{mm}{pixelsl}$ form. Pythagorean Theorem was then used to find r. To find θ I used $\theta = \arctan(\frac{r}{f})$. For φ I used $\varphi = \arctan(\frac{centroidY}{centroidX})$ based on the following:

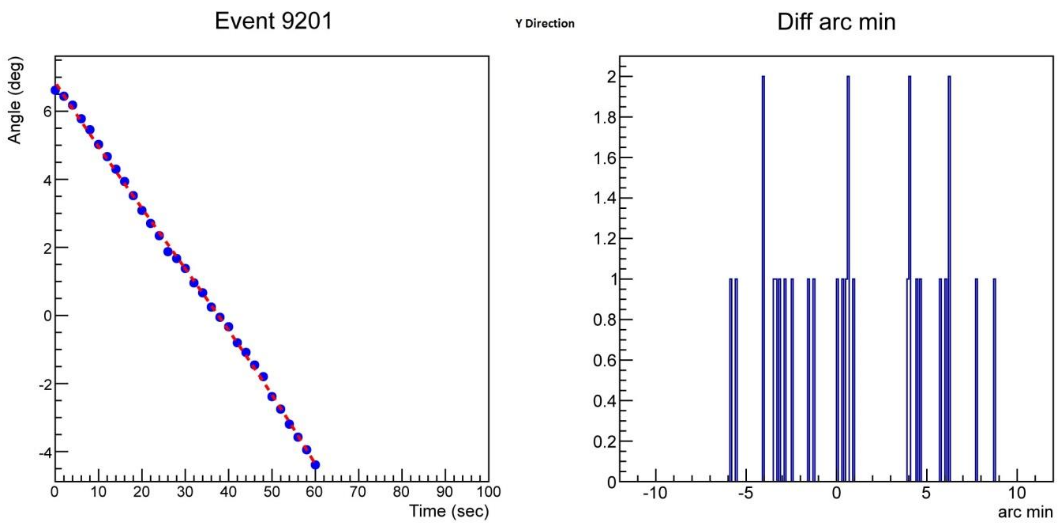


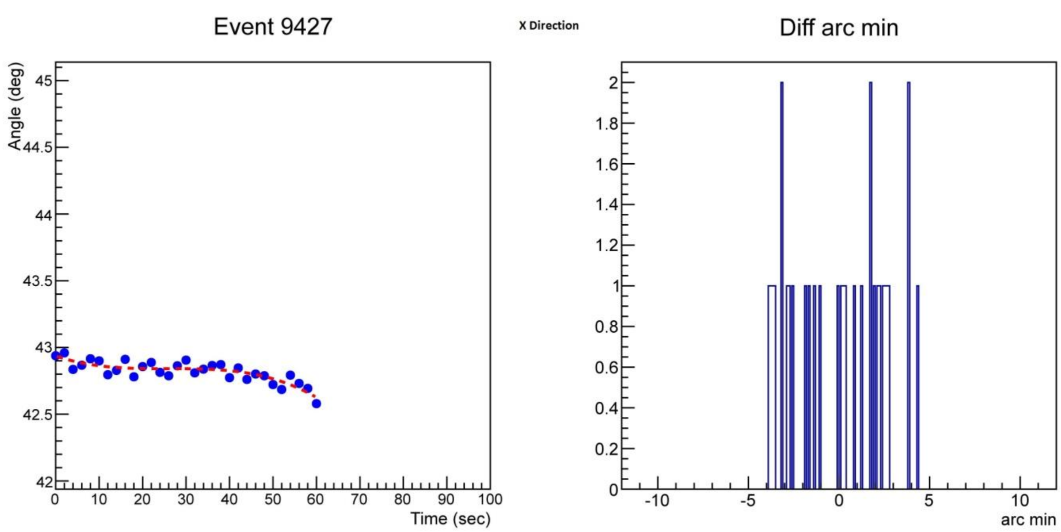
Depending on the location of the centroid 0, 90, 180, or 270 degrees was added to φ that was found. This placed φ in the correct quadrant with 0^0 as quadrant one with positive x and y and the quadrants continuing counter-clockwise.

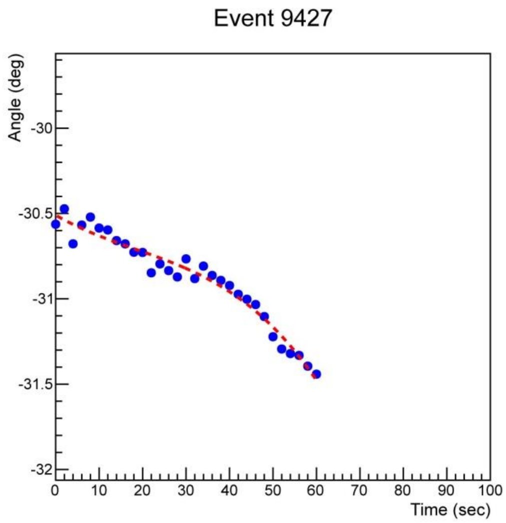
3.4 Conclusion

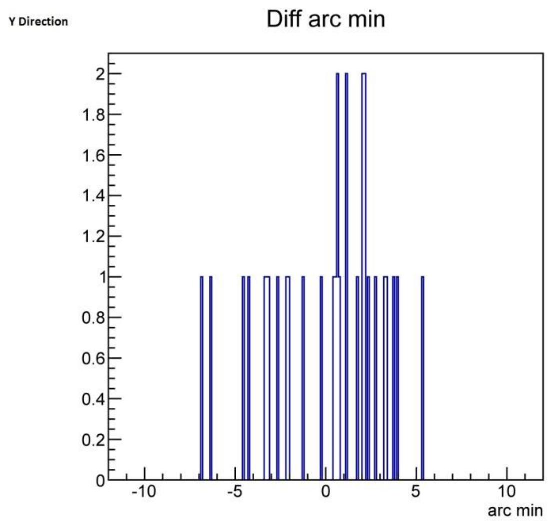
We tested precision using a 3rd degree polynomial line of best fit over one minute time intervals for five different times during float. As shown in the figures below, the difference between our points is well within the desired 20 arc minutes.


Event 9201
X Direction
Diff arc min
Angle (deg)
Time (sec)
arc min


Event 9201
Y Direction
Diff arc min
Angle (deg)
Time (sec)
arc min


Event 9427
X Direction
Diff arc min
Angle (deg)
Time (sec)
arc min





4. Team Demographics

Name	**Classification**	**Gender**	**Ethnicity**
Ryan Gueho	Undergraduate	Male	Caucasian
Nicholas Chason	Undergraduate	Male	Caucasian
Joel Taylor	Undergraduate	Male	Caucasian
Josh Frick	Undergraduate	Male	Caucasian

Josh Frick graduated from LSU in May 2013 with a BS in Mechanical Engineering and is currently employed with URSCorp as a mechanical engineer.

Joel Taylor completed his undergraduate degree at LSU and was accepted in graduate school at LSU for Experimental Condensed Matter and AMO. His work is with Ultrafast Laser Spectroscopy.

Faculty Advisors:

Dr. Michael Cherry

Dr. T. Gregory Guzik

Dr. Ching-Cheng Hsu

Brad Ellison

Michael Stewart

5. References

1. Bryan, Noelle. *HASP Student Payload Application for 2013: SMITH*. Rep. Baton Rouge: n.p., 2012. Print.
2. Budden, B., G. L. Case, and M. L. Cherry. *Design Concept for a High Altitude Rotating Modulator Gamma-Ray Imager*. Tech. N.p.: n.p., 2010. Print. Nuclear Science Symposium Conference Record.
3. Capps, Winston. *Orientation System*. 2012. Solid Model. Baton Rouge. *SolidWorks*. N.p.: n.p., n.d. N. pag. Print.
4. Guzik, T. G. *HASP 2012 Flight - September 1, 2012 - Total Rotation of Platform*. 2012. Chart. Baton Rouge. *Microsoft Excel*. N.p.: n.p., n.d. N. pag. Print.
5. Guzik, T. G. *HASP - Student Payload Interface Manual*. Issue brief. 02.17.09 ed. N.p.: n.p., n.d. Print.
6. Isbert, Joachim B., and Mike L. Cherry. *Conceptual Study of a Daytime Orientation System for High Altitude Balloon Platforms with Arc Second Accuracy*. Tech. Baton Rouge: n.p., 2011. Print.
7. Smart, W. M., and Robin M. Green. *Textbook on Spherical Astronomy*. 6th ed. Cambridge: Cambridge UP, 1977. Print.