Arapahoe Community College

**DEMO SAT DESIGN DOCUMENT**
Solar-Tracking Spectrometer

Written by:
Cameron Bain, David Colclazier, Elijah Mocabee, Jacob Pelster, Lawrence Perkins, Magdalena Franchois, Natalie Margaros, Nick Vail, Solomon Sidhu, Spencer Jackson, Tasha Estein, Tyler Ghafrari

4/14/2017
Revision I
## Revision Log

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Preliminary Design Review</td>
<td>02/10/17</td>
</tr>
<tr>
<td>B</td>
<td>Critical Design Review</td>
<td>03/10/17</td>
</tr>
<tr>
<td>C</td>
<td>Launch Readiness Review</td>
<td>04/07/17</td>
</tr>
<tr>
<td>D</td>
<td>Analysis and Final Report</td>
<td>See Website</td>
</tr>
</tbody>
</table>
# Table of Contents

## 1.0 Mission Overview

## 2.0 Requirements Flow Down

## 3.0 Design

3.1 Spectrometer and Structure Design
- Figure 3.1: General spectrometer design
- Figure 3.2: Concept design of payload
- Figure 3.3: Final design of spectrometer

3.2 Software & Flight Computer
- Figure 3.4: UML Diagram
- Figure 3.5: GraphIt! Software

3.3 Electronics
- Figure 3.6: Diagram of Electronics

## 4.0 Management

4.1 Schedule

4.2 Team Structure

## 5.0 Budget

- Figure 5.1: Budget

## 6.0 Test Plan and Results

- Figure 6.1: Temperature vs. Time for Heating Resistor
- Figure 6.2: Neon Spectra
- Figure 6.3: Hydrogen Spectra
- Figure 6.4: Sunlight Spectra

## 7.0 Expected Results

## 8.0 Launch and Recovery

- Figure 8.1: Payload at Recovery Site

## 9.0 Results, Analysis, and Conclusions

- Figure 9.1: Temperature of Heating Element and Payload vs. Time
- Figure 9.2: Magnetic Field Readings
- Figure 9.3: Battery Percentage vs. Time
DemoSat

10.0 Ready for Flight 22
11.0 Conclusion 23
12.0 Message to Next Year 24
1.0 Mission Overview

Our mission objective was to send a spectroscope up to approximately 100,000 ft while looking at the sun to measure changes in the light spectrum (dependent variable) with changes in altitude (independent variable). We also had the goal in mind of refining our positioning system to autonomously point the spectroscope toward the sun throughout the duration of the flight. In addition, collecting kinematic and environmental data was a side goal in order to better understand flight conditions and improve payload design accordingly for future research. We expected to see/prove that each specific element in the atmosphere at the instant light entered our spectroscope. In the scientific field it is fundamental to retest findings from others who conducted the same experiment. We hoped to verify the presence certain elements are in our atmosphere using our spectroscope.

2.0 Requirements Flow Down

Requirements

Structural integrity
- Under 1 kg (weight limit)
- Durable enough to fly repeatedly

Flight Computer
- Accurate control of kinematic elements
- Accurate and timely data collection
- Accurate data compression/storage
- Accurate parsing and decompression of flight data
- Concurrent operation

Sound logic
- Ability to point at the sun
- GoPro continuously take photos

Budget
- Maintain a healthy budget
3.0 Design

In order to complete the mission requirements, three systems had to be designed: the spectrometer, the structure, and electronics.

3.1 Spectrometer and Structure Design

For the spectrometer design, the general theory for observing absorption lines was used (See figure 3.1). The light in which the absorption spectra is to be observed is passed through a slit into a collimating lens, the collimated light passes through a diffraction grating, which breaks the light into individual spectra, which are then refocused by a lens and sensed in a detector array. In the built spectrometer a 6mm lens from a telescope was used to collimate the light. One major difference between our spectrometer and other spectrometer designs was the use of a mirrored 600 slit diffraction grating, which allowed the spectrometer to decrease in overall length, due to the redirection of light (See figure 3.3). The focusing lens and detector array were both part of the Go-Pro Camera, which was used to capture the spectra.

The intent of the structural design was to decrease the overall weight of the payload and provide the spectrometer box unimpeded rotation for the purposes of always pointing towards the sun. The design team settled on a “butterfly” design. Two tubes connected by a bracket with the spectrometer sitting on one tube (See figure 3.2). The two-tube design was picked to distribute weight evenly, offsetting the weight of the spectrometer box on one tube, with the electronics package in the opposite tube. For free rotation of the spectrometer box, the box was set on a Pet+ 3d printed platform, which was rotated with a servo motor. For the tubes, 4 inch diameter mail tubes were used, as they were both lightweight and sturdy. They were connected using a bracket made of PETG 3D printed plastic and metal bolts. On the inside, insulation was used to line the tubes, preventing heat escape. In order to keep the servo motor and batteries from overheating, an electric heating resistor was placed in the same tube as the servo motor and batteries, thus ensuring continued operation in lack of environmental heat.
Figure 3.1: General spectrometer design

Figure 3.2: Concept design of payload

Discussion on how requirements are being met with current design (Rev B)
3.2 Software & Flight Computer

For past projects we used microchips from the Arduino product line, due to its popularity, the myriad of sensor drivers it can use, and the usable software available online. However, with a 16MHz crystal and 2 kb SRAM, there is an obvious limit to the complexity of software that can run on it. To mitigate this for a project of any complexity, we chose to work with the .NET framework by using a Netduino 3 Wifi controller, which upgrades both the processing power and memory 168 MHz and 164 kb SRAM. Using this board, we developed a modular software suite takes advantage of .NET’s advanced task scheduling and concurrency capabilities to provide a lightning fast flight computer for us to integrate various features and sensors into.
Our solar tracking strategy was relatively simple in nature. The sun tracking algorithm used our BNO055 orientation sensor to extrapolate the euler heading of the payload when it was powered on, and to calculate its current offset from that position and compensate for it as time progresses. As long as the spectrometer was facing the sun when the payload was powered on, it would continue to face the sun moving forward.

We also included a myriad of sensors to collect various environmental data: pressure, altitude, temperature, and gravitational and magnetic field readings. We used existing C++ libraries for these sensors to create .NET micro drivers, all of which are open-source and available via the GitHub link in the appendix.

Modularity and concurrency for our flight computer was handled using a thread-pool. Any number of persistent threads running on the flight computer constantly monitor a queue of “work items”; A work item is a segment code which performs some task on the flight computer: logging debug statements, updating sensors, writing to the SD card, etc. When a work item needs to be executed, it is added to the list of pending actions the thread-pool monitors, to be processed by the next available thread. Some work items, i.e. sensor updates, are marked as repeatable such that they are added back to the thread-pool’s pending work items upon completion of its action. For sensor updates which require a much higher frequency than the thread-pool can offer (a repeatable action can only run once per every other repeatable action), we lower level c-style loops to force execution time to remain inside the high-frequency update until it finished collecting a particular number of updates sequentially. By assuming that the period in between each update was constant based on the generally constant load of the processor, we simply collected a timestamp before the update started and after it finished and used them to interpolate the times for each update in between.

We also had to get creative with the data-logging portion of the software, our goal being to ensure it was both as efficient and quick as possible. Each sensor-update related work item was designed to parse the data for the sensor into a binary packet to be more quickly and easily manageable by the data-logging system than the original UTF representation of the data. We were also able to achieve awesome compression for high-frequency update packets by removing the need for a 3-byte time stamp for each update within the packet. For an update packet containing 1200 updates of 1 byte each, this saves approximately 3.52 kb for each packet or around 210 kb/min, allowing more data to be stored on the SD card and thus a longer-lasting logging capability.
The structure for binary data packets is as follows:

<table>
<thead>
<tr>
<th>Start</th>
<th>Data Type</th>
<th>Data Size</th>
<th>Time-Stamp</th>
<th>Data</th>
<th>End Time-Stamp (for compression)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0xFF]</td>
<td>[0x00-0x04]</td>
<td>[1 byte]</td>
<td>[3 bytes]</td>
<td>[n bytes]</td>
<td>[3 bytes]</td>
</tr>
</tbody>
</table>

- Some packets did not require timestamps.

The flight computer logic uses the following logic after powering on:

- Upon initial power up, the payload enters a simple boot state, initializing communication channels and the status panel, as well the payload logger and internal timer.
- Once this is complete, the payload begins initializing and communicating with the different sensors attached to it.
- After all sensors are initialized, the boot sequence starts the associated work items for each sensor, causing them to be added to the flight computer’s execution queue.
- Once all sensors are updating and logging properly, the status panel switches to a runtime mode, displaying the calibration values on the orientation sensor, allowing us to calibrate it prior to launch.

A UML class diagram for the flight computer follows. In addition, the repository containing all of the flight computer code is available in the appendix (section 3.3).
Because of the nature of the flight data, our development team developed another piece of software capable of parsing this data into readable comma-separated-value
tables for later analysis. The repository containing the source code for this software is available in Appendix section 3.5

In addition, our lead developer integrated new features into some software he’s developing called “Graphlt!” which allows for simple graphing and statistical analysis of sensor data. This began during the 2016 RockSat program and is still under active development, but supports automatic updates in case anyone wants to take advantage of the new features as they arrive. A link to download the software is available in Appendix 7.3.

![Graphlt! Software](image)

Figure 3.5: Graphlt! Software

3.3 Electronics

The electronics team aimed to change as little as possible when repeating this experiment. Since the goal of the experiment was to improve on the spectroscope design, we wanted to limit the differences between the two projects to better compare the data we collected. Our new electrical design goals were to:

- Design a logic circuit that would be able to detect if it was pointed at the sun.
- Store enough energy to power the payload for at least 3 hours.
- Interface with image sensor to allow for control.
- Heat critical components during the flight.
- Monitor condition of the main battery during flight.

The electrical system consisted of two isolated subsystems, the controller and the
spectrum detector. The controller consisted of the flight computer and the solar tracker which were interconnected between the two sides of the tubular payload design via 22 gauge color coordinated wiring. The primary tube housed the servo, heating control circuitry, and power system for the entire payload. To prevent any failures of these critical parts, a heater was implemented to prevent cold environment failures. Since the payload was divided into two sides, we prioritized the heating to one side. This was done to minimize the amount of power required to drive to heater. The spectrum detector subsystem was a modified GoPro Hero 3 plus silver edition. No electrical modifications were required to operate. Due to the amount of heat the camera produces during operation, the spectrum detector was left unheated.

![Diagram of Electronics](image)

**Figure 3.6: Diagram of Electronics**

### 4.0 Management

This year ACC shied away from the traditional flow model of electing individuals to be leaders of a specific teams. We have found in the past that this can be detrimentally stressful to these selected individuals. Instead, we formed three teams: Electronics, Structure, and Software. Each team had goals that all individuals were encouraged to and capable of completing and most aspects of management flowed to the responsibility of the individual.
4.1 Schedule

CDR - March 10, 2pm
To be complete: 1st prototype, working flight computer, fully testable unit, in testing phase prior to rebuild.

Final Prototype - March 24
To be complete: fully tested and working prototype, in final payload build phase

LRR - April 7, 2pm
To be complete: Final payload, fully working flight computer, fully tested unit.

Design Document- April 5, 5pm
To be complete: Revised payload, fixed brackets, camera, flight computer, fully retested unit.

4.2 Team Structure

<table>
<thead>
<tr>
<th>Teams</th>
<th>Cameron Bain</th>
<th>Cameron Bain</th>
<th>David Colclazier</th>
<th>David</th>
<th>Cameron Bain</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Electronics</strong></td>
<td>Cameron Bain</td>
<td>Cameron Bain</td>
<td>David Colclazier</td>
<td>David</td>
<td>Cameron Bain</td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td>Cameron Bain</td>
<td>Cameron Bain</td>
<td>David Colclazier</td>
<td>David</td>
<td>Cameron Bain</td>
</tr>
<tr>
<td><strong>Research and Development</strong></td>
<td>David Colclazier</td>
<td>Jacob Pelster</td>
<td>Magdalena Franchois</td>
<td>Tasha Estein</td>
<td>Magdalena Franchois</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td>David Colclazier</td>
<td>Jacob Pelster</td>
<td>Magdalena Franchois</td>
<td>Tasha Estein</td>
<td>Magdalena Franchois</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td>David Colclazier</td>
<td>Jacob Pelster</td>
<td>Magdalena Franchois</td>
<td>Tasha Estein</td>
<td>Magdalena Franchois</td>
</tr>
<tr>
<td><strong>Lawrence Perkins</strong></td>
<td>Magdalena Franchois</td>
<td>Nick Vail</td>
<td>Lawrence Perkins</td>
<td>Natalie Margaros</td>
<td></td>
</tr>
<tr>
<td><strong>Magdalena Franchois</strong></td>
<td>Nick Vail</td>
<td>Tasha Estein</td>
<td>Lawrence Perkins</td>
<td>Natalie Margaros</td>
<td></td>
</tr>
<tr>
<td><strong>Solomon Sidhu</strong></td>
<td>Spencer Jackson</td>
<td>Tyler Ghaffari</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Spencer Jackson</strong></td>
<td>Tasha Estein</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tyler Ghaffari</strong></td>
<td>Tasha Estein</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.0 Budget

Figure 5.1: Budget

6.0 Test Plan and Results

Cold Testing:

Heater operated for ~30 minutes, Servo stops at ~0°C

What was fixed:

Increased heater battery capacity, increased heater power output, and repositioned temperature sensor.
Stair pitch test results:

The payload was kicked down a flight of stairs and sustained no damage.

Status - Durability proven

Drop test results:

The payload was dropped from a 3-story building.

1st test: Nylon screws used to attach brackets were sheared off due to impact.

What was fixed - Replaced nylon screws with metal screws.
2nd test: The payload sustained no damage.

*Status*: Go!!

**Whip test results:**

*What was found* - The payload was spun for at least 15 full revolutions at high speed with no issues.

*Status* - Go!!

**Shake test results:**

*What was found* - All components stayed in place throughout the shake test.

*Status* - Strong, stable components ready for launch.

**Overall structural test results:**

*What was found* - Drop test, stair pitch test, whip test, and shake test completed.

*What was fixed* - In total, replaced nylon screws with metal screws. Now only one bracket is required and was tested again through all structural tests.

*Status* - Go!!

**Spectra test results:**

The discrete lines of a spectra of neon and hydrogen lights could be seen clearly. Because of this, we inferred that we could see absorption lines in the upper atmosphere.
GoPro camera needs to be precisely focused in order to see absorption lines.

**Power systems test results:**

*What was found* - Payload operated for ~3.5hrs Battery was at ~3.4V (dead)

*Status* - Go!!

### 7.0 Expected Results

Our expected results were dependent on the success of our sun-tracking technology. As long as our instrumentation tracks the sun, our spectrometer would collect and diffract the spectra; the images of which would be captured by the GoPro camera. After collecting our intact payload, we expected to see 5,000+ spectra images collected in the SD card.

From the testing of the spectrometer, it was shown that the spectrometer could pick up discrete emission lines from energized materials, and can pick up a full solar spectra. When in the atmosphere, it was expected that the absorption spectra of nitrogen will be the most prominent in the data, as nitrogen content increases in concentration as altitude is increased. By observing the missing spectra, and corresponding the spectra to the absorption spectra of different elements, it should be possible to gain an idea of which elements are present in the atmosphere at a given altitude, given the spectrometer was pointed at the sun.

We also expected to receive various other environmental data, such as payload orientation and kinematics, gravitational and magnetic field readings, and temperature
readings from both internal components and heating elements.

We had performed extensive tests to ensure complete software and hardware integrity in a variety of environments. We placed the spectrometer in a container filled with dry ice for three hours and performed flight tests as if it was in flight to ensure that once we launched the payload, it would run as expected in the freezing environment of the upper-atmosphere. We meticulously performed spectral tests to ascertain the precise orientation required for the spectral box to diffract the light at just the right angle for the GoPro to take images of the spectral readings. We placed the payload into a vacuum to ensure that the components work in this environment as well as to test for unwanted degassing. We dropped the payload off of a three story building, kicked it down a flight of stairs, and whirled it at a high rate of spin to ensure structural integrity.

8.0 Launch and Recovery

The payload was launched successfully at 7:15 a.m. on April 8th, 2017; there were no issues with the launch. We launched in Eaton, CO and went northwest of Sterling to recover the payload after landing at about 9:30 a.m. The payload landed in a field and was moved roadside before the teams arrived. The first thing we noticed after recovering the payload was that the center bracket had broken off from the payload at the joint holding the bolts to the(Fig. 8.1). We speculate that this happened when the payload hit the ground. All other pieces of the payload were structurally intact. Our GoPro was intact. However, it was not operating when we recovered the payload. Upon further investigation, we discovered that the GoPro operated for just over 2 minutes, taking 150 photos with the majority most likely having taken before the flight.
9.0 Results, Analysis, and Conclusions

Unfortunately, the camera only had 150 pictures in storage upon recovery. We thought the GoPro was taking a photo every second throughout the flight, however, it only took photos during the preflight readiness check. We realized we were incorrect when we recovered the payload and noticed that the SD card was nearly empty when it should have been completely full. We tried to find the source of the issue by turning the GoPro back on and waiting until it stopped taking photos which was after about 10 photos, or ten seconds. This led to the conclusion that an error in the script on the GoPro was not successfully restarting the camera after finishing a burst of 10 pictures @ 1Hz. The 150 pictures on the SD card included 10 from the flight, and 140 from various tests both before and after launch.

Temperature inside the payload was regulated throughout the flight and reported to be within acceptable levels for the majority of the flight until the battery died. There appeared to be an error with data collection on one of the sensors as it stopped recording the temperature of the heating element approximately 35 minutes into the flight. We suspect that a piece of hardware came in contact with the microcontroller
causing the failure. Despite this, the payload did not reach a temperature below 0 degrees until around 1.5 hours in flight, and did not start decreasing at a non-heated rate until the battery died:

![Figure 9.1: Temperature of Heating Element and Payload vs. Time](image)

In looking at the other environmental data, it appears that there might have been a problem with the electronics after a certain point in the flight. We hypothesize this because as Earth’s magnetic field changes minorly throughout the flight path, our data should reflect similar values as well throughout the entirety of the flight. However, after about 95 minutes into flight the readings became highly erratic:

![Figure 9.2: Magnetic Field Readings](image)

Although this may have been caused by turbulence with the payload after burst, we believe it is more likely that power distribution to the sensor caused readings to fluctuate towards the end of the flight. Whether this was due to the dwindling battery capacity or some other hardware failure, we are not sure. It does seem likely that it is related to battery capacity, as the fluctuations seem to happen at the same time the
battery dropped to below 40% capacity:

![Battery Percentage vs. Altitude](image)

**Figure 9.3: Battery Percentage vs. Altitude**

The photos retrieved after the flight were not useable in our data analysis. The photos were taken during the preflight readiness check rather than throughout the flight. Post flight, we ran the script in the GoPro and determined that the program was taking 10 photos before shutting down, rather than taking sustained photos. This issue did not allow us to gather data for spectral analysis. We cannot confirm, deny or null our hypothesis or regarding spectral analysis. That said, we are launching this payload three more times this year in the hopes to correct our mistakes.

### 10.0 Ready for Flight

After redesigning and testing a new bracket, rewriting and uploading a new script and firmware to the GoPro, and replacing the netduino our payload is ready to fly again. The new bracket has been subjected to a series of stress tests proving its durability. Additionally, we redesigned the code for the GoPro so that it will take one picture every minute until the battery runs out. Furthermore, we replaced the netduino after failure of the adc, which will enable consistent temperature readings throughout
The payload will be stored securely in the physics lab. Our team will be meeting weekly until the Eclipse Launch to polish the payload and perform administrative tasks.

If not launched within 6 months, the only component that may not perform perfectly are the batteries, which are easily removed and recharged. There are no volatile parts in the payload, thus rendering an otherwise infinite shelf life to our spectrometer.

11.0 Conclusion

Revisiting the spectrometer and solar-tracking payload has been a worthwhile experience. We have had the opportunity to apply what we learned from our last experience which culminated in a fundamental redesign. This redesign brought forth a minimalist design with lighter and more effective instrumentation than the last spectrometer project. Creating a viable spectroscopy within the size and weight limits proved to be both challenging and fulfilling. Additionally, redesigning the solar tracker to track in only two planes as opposed to three planes proved to provide a much simpler and effective strategy. Furthermore, initializing the solar tracker while pointing at the sun also proved to produce more consistent results because there are less variables to account for.

11.1 Lessons Learned

Performing a full systems test should be done before launch to ensure all components work together for the expected duration of the flight. We have also learned that we should have tested our PET-G bracket more aggressively. Additionally, we should have tested different 3d printed filaments to determine which material would be best suited for the expected stresses. Regarding the hardware failure, we learned to compartmentalize our work on the components. We could have avoided the microcontroller ADC failure by limiting the type of work that was being performed on the payload at the same time. By changing the function of the solar tracker to point at the sun on X and Y axis, we had more even weight distribution and less complicated program code. Although we feel we have improved our design process, continued work and testing on this design is warranted.
12.0 Message to Next Year

For the next DemoSAT team, it is crucial to remember that there is no such thing as too much testing or testing too soon. One of the biggest difficulties we ran into was the fact that we ran consistently behind schedule, and ran out of time in the end to do important pre-flight testing. Making a detailed timeline as early as possible in the semester, punctuated by set deadlines provided by Colorado Space Grant, should assist with this process. Additionally, making sure to take detailed notes of what was worked on after the end of each session will help the team stay on track as to what has been accomplished. Checking our progress every two weeks will ensure that we will stay on track and enabled us to spot potential problems as they arise. Dividing our group into teams assured that all who participated could be involved in specific and detailed work.