April 29th, 2016

Dr. Angel Abbud-Madrid  
Director, Center for Space Resources  
1310 Maple Street  
Golden, Colorado 80401

Dear Dr. Abbud-Madrid,

For the EPICS II-Planetary class project, Dr. Angel Abbud-Madrid of the Center of Space Resources at Colorado School of Mines, and the Colorado Space Grant Consortium’s DemoSat program, requested that the team design and build a payload that will test how effective the insitu regolith on Lunar surfaces can shield cosmic radiation. Astronauts who land on the Lunar surfaces are exposed to radiation in quantities that are greater than what the human body can naturally intake. This poses a threat to the astronauts, and as a result, NASA is attempting to determine whether the regolith present on the Moon and Mars could potentially be used to create inhabitable stations on the Lunar surfaces. If the regolith could be used to provide shelter, this would greatly decrease transportation costs because astronauts would no longer have to carry housing supplies with them on their rocket. This would reduce the weight of the rocket, making for a more efficient flight. This experiment is important because it can help determine, if at all, how effective Lunar regolith is at shielding radiation.

Team Cosmic Shield was contracted with this project on January 27th, 2016. We were contacted by Mrs. Bernadette Garcia of the Colorado Space Grant Consortium’s DemoSat Program, who explained in detail the requirements and constraints of the project. The final product was delivered on April 9th, 2016, which was the launch date. After the launch, the data was collected, and the attached final report was created to explain in detail all aspects of the project.

This project was not without its challenges. But after much careful planning and trial and error, the final product we delivered turned out to be a success. Team Cosmic Shield thoroughly enjoyed designing this payload, and with that we would like to thank you for giving us the opportunity of working on this project and trusting that we would deliver a satisfactory end result.
FINAL

DESIGN REPORT

Submitted To: Dr. Angel Abbud-Madrid

Submitted By: Kevin Woods

April 29th, 2016

Team Members:

Marek Eddy, Sam Kozan, Trey Richard, Kevin Woods
# Table of Contents

Executive Summary ................................................................. 4

Introduction .............................................................................. 5

Background .............................................................................. 5

Colorado Space Grant Consortium .............................................. 6

Final Design .............................................................................. 7

Subsystems .............................................................................. 11

Housing .................................................................................. 11

Geiger Counters and Moon Dust Simulant .................................. 14

Arduino Assembly ..................................................................... 16

Heating .................................................................................... 18

Assembly and Operation .......................................................... 20

Interior Components ............................................................... 21

Cost Summary .......................................................................... 21

Testing .................................................................................... 24

Mission Launch ........................................................................ 25

Results .................................................................................... 26

Conclusion ............................................................................... 27

Application .............................................................................. 27

Improvements ......................................................................... 27

Bibliography ............................................................................ 28

Appendix .................................................................................. 29
Executive Summary

For the EPICS II-Planetary class, Dr. Angel Abbud-Madrid, of the Center of Space Resources at Colorado School of Mines, and Mrs. Bernadette Garcia, of the Colorado Space Grant Consortium’s Demo Sat program, requested that the team design and build a payload that can test how effective the in-situ regolith on the Lunar and Martian surfaces can shield cosmic radiation. The purpose of this report is to outline the design process and the experiment conducted by the team.

The final design of the teams payload was rectangular in shape. This was determined to be the most effective. The payload was constructed of foam board, silicone, and wooden boards used to mount the various components in place. These components include two Geiger Counters and two Arduino boards, each equipped with protoshield(s) and altimeters. This system was powered by a 12 volt battery. A separate circuit was utilized to heat the payload. This consisted of three 4 Ω resistors in series with three 9 volt batteries.

The total cost of the designed payload came to $465.21. The most expensive components were the Geiger counters. These were priced at $149.15 each. The team utilized two Geiger counters at a price of roughly $300.00. Several other components were utilized from the 2015 experiment. This included the 12 Volt batteries and one Arduino board equipped with a protoshield, OpenLog and altimeter. A second Arduino board and the attached components were purchased for roughly $60.00. These were originally purchased to run both Geiger counters before the team realized the additional components were needed to effectively run the entire system. The heating system was relatively cheap with the three 9 volt batteries being the most expensive at roughly $15.00

The designed payload was launched on April 9th, 2016. It was attached to a high altitude balloon, along with various other payloads. The balloon reached a height of roughly 17,600 meters before the balloon burst and the payloads returned to earth attached to a parachute. The Payload was recovered with minimal to no damage and was determined to have had a successful launch.

Although some error was present in the data logging, it is shown that the simulant shielded Geiger counter had on average lower radiation readings than the control Geiger counter. The team hopes that the data obtained can be further used for determining whether or not the simulant should be used for constructing habitable stations on the lunar and Martian surfaces. Further analysis of the data can be found in the body of the report.
Introduction

Background
Cosmic radiation consists of any particle that bombards the Earth from beyond its atmosphere [Citation: NASA]. Radiation is abundant outside the Earth’s atmosphere, and the Sun is the largest source for particles directed towards Earth. Highly energized protons are propelled towards Earth as a result of the hydrogen fuel that is consumed during fusion within the star. The Earth is initially shielded by the magnetic field surrounding it, exerted from the poles. This magnetosphere provides protection for 99.9% of harmful radiation [M:Space Mars]. The Earth’s atmosphere provides additional protection for the remaining 0.1%. This is why humans are able to safely inhabit the Earth’s surface without being exposed to harmful radiation.

Contrary to Earth, the surfaces of the Moon and Mars are not equipped with natural radiation shielding. The Moon has no atmosphere and a weak magnetic field, and Mars has no magnetic field and an atmosphere stripped by solar winds.

Because of this, astronauts on these cosmic bodies are exposed to radiation in quantities that are greater than what the human body can naturally intake, less than 6.3 cGy (cubic gray) per hour (Seed 2011). One possible solution, with various other benefits, would be to create habitable stations on these surfaces that can effectively shield radiation.

It is thought that the basaltic regolith present on Lunar surfaces is a prime candidate for radiation shielding due to its ability to reflect certain radiation waves (Miller et al. 2009) As a result, NASA is attempting to determine whether lunar regolith could potentially be used to construct habitable stations on the Lunar and Martian surfaces. NASA wants to test the in-situ material of the planet because it is readily available on the bodies, and transporting shielding material from Earth would be costly and inefficient. On a large scale, many trips would need to be made to and from Earth to transport the materials needed. The additional material on the shuttles would increase the weight, thus decreasing flight efficiency and resulting in higher transportation costs, such as fuel consumption. For these reasons, it is more economical to utilize the readily available

Figure 1: Diagram of Earths Magnetosphere.
substitute present on Lunar surfaces. This experiment is important because it can help determine, if at all, how effective lunar regolith is at shielding radiation.

Colorado Space Grant Consortium
The COSGC is funded by NASA and is a statewide organization involving 17 colleges, universities and institutions around Colorado. The COSGC’s DemoSat Program sponsors an event on a quarterly basis, in which they launch high altitude balloons into the atmosphere. A multitude of payloads constructed by teams from participating institutions are attached to the balloons, which are equipped with a parachute and a GPS tracker.

![DemoSat Balloon Diagram](image)

Figure 2: DemoSat Balloon Diagram.
Each payload is uniquely constructed by its respective team and has flight objectives varying from the other teams. The balloons are to rise to an estimated elevation of 100,000 feet (30,480 meters) before they expand and pop. The entire system will then falls back to Earth, and the parachute deploys, allowing for a safe landing and recovery of payloads. Prior to launch, the DemoSat program provides each team with specific guidelines to be met regarding the overall design of the payloads. The guidelines are as follows:

- The total mass of the payload must not exceed 1.1 kg.
- It must withstand temperatures as low as -80°C.
- It must survive landing impact speeds of up to 100 mph.
- All electrical switches must be located on the exterior of the payload.
- The flight tubing must be plastic and have a ⅜-inch inner diameter.
Team Goal
Team Cosmic shield is committed to research, improve, and construct the Lunar soil simulant shielding experiment conducted from Team MEAPS (Spring 2015). The team aims to record relevant data to further understand the shielding properties of Lunar and Martian soil so that it may provide a cost efficient alternative to current shielding methods. The team is also committed to having a successful flight within the requirements set forth by the COSGC.

Final Design

![Final Design Isometric View (Exploded)](image)

Figure 3: Final Design Isometric View (Exploded).

Given the specifications laid out by the COSGC, the team designed a payload to measure both radiation and altitude as it ascends into the atmosphere. The payload housing was constructed of foam board. Ultimately, a square design was determined to be the most efficient housing design. Encased in the housing was two Geiger Counters, an altimeter, data logger two Arduino Boards, two proto shield(s), batteries, and a heating element. One of the Geiger counter sensors was cased in a Lunar dust simulant developed at Colorado School of Mines.
The simulant was used as a substitute because genuine moon dust is not readily available and access to it is cost prohibitive. The remaining Geiger counter was not encased with the moon dust simulant, and was used as the control portion of the test.

The shielded Geiger counter data was compared to the unshielded Geiger counter data to see if there was a difference in radiation readings. Two Arduino boards were used run each Geiger counter separately. The readings outputted from the Geiger counter were recorded in one second intervals. A protoshield was attached to each Arduino board to allow an interface between the Arduino board the rest of the components. Attached to the protoshield was a to measure the pressure, which is converted into the altitude (meters). The results also included the radiation readings (CPM), using an OpenLog data logger to record the measurements.
Figure 6: Arduino Assembly.

One 12-volt battery was used to power both Arduino boards and Geiger counters. A separate circuit was used to run the heating system.

Figure 7: Arduino Circuit Diagram.

The heating system consisted of three 4 Ω ceramic resistors in series with three 9 volt batteries powering the circuit. The resistors emitted heat when current was passed through them.
The following section will go into further detail on the components of the final design. The payload was divided up into four subsystems: housing, Geiger counter and moon dust simulant, Arduino assembly, and heating. The subsystems were determined by grouping all the components that worked most closely together, and breaking up the components that did not really affect one another.
Subsystems

Housing

Figure 9: Housing Isometric View

Function
The overall design of the housing serves as the shielding of the experiment. The housing has to ensure that all the DemoSat guidelines are met. The major function of the housing subsystem is to protect and guarantee survival of the experiment.

Design
For the housing, several designs were considered when trying to determine the best possible design. Four major requirements were used in evaluating the overall effectiveness of each design. They were balance, packing efficiency, center of gravity, and survivability. The balance of the payload was vital in determining how the payload will fall back to earth. The packing efficiency dealt with ability to pack all the components inside the payload while using the least possible amount of space. It was crucial for the payload to have a low center of gravity to ensure that it would not have the tendency to want to tip over during flight. Lastly, it was crucial that the payload survive the impact of the fall so that it could be recovered to have the data extracted. The four considered designs were based on the examination of past missions and determining which of those payloads experienced the most success. The four structures that were considered were a sphere, pyramid, cylinder, and a box.
Sphere
A spherical design was considered for the payload housing because of its uniformity. For this design, the components would be able to sit near the flight string, which would generate good balance. A spherical design would also have a high chance of surviving a potential impact. The sphere would deflect most of the energy and protect the components inside in case of parachute failure.

Pyramid
A pyramid design was considered because of its strength and low center of gravity. However, the pyramid design would result in excess space that would not be utilized. This would require more foam and would add unnecessary weight to the design.

Cylindrical
A cylindrical design was considered because it resembled the spherical design. This design would be more efficient than the spherical design, however, and also would minimize weight. However, because the components of the payload are rectangular shaped, there would be wasted space in this design.

Box
Ultimately, the box was chosen as the final design for the housing. This design fit the components of the payload in a more efficient manner than any of the other options. Because the components fit more efficiently, the weight of the design would be reduced. The components would be aligned near the flight string, which would give the payload exceptional balance. One drawback associated with this design is its survivability. Landing on a corner in case of parachute failure could damage the payload severely.

Decision Matrix
A decision matrix was used to evaluate the major requirements and determine the best possible housing design (Table x). Each design requirement was scored out of 5, with 1 being the lowest and 5 being the highest. These criteria are important because each could potentially have a major impact on the outcome of the experiment.
Table 1: Design Matrix for Alternative Housing Design

<table>
<thead>
<tr>
<th>Designs Considered</th>
<th>Packing Efficiency</th>
<th>Balance</th>
<th>Low Center of Gravity</th>
<th>Survivability</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>Pyramid</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Cylinder</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>Box</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>16</td>
</tr>
</tbody>
</table>

*Note: Ratings are on a 1 to 5 scale with 5 being the highest.*

**Specifications**

As mentioned above, a box design was ultimately found to be the best design option. The housing subsystem has four major components: switches, plywood bases, flight tube with connections, and the overall housing protection. For the exterior of the housing, aluminium reflective material was used.

The housing serves as protection to the internal components. The dimensions of the components are as follows: (L x W x H) (cm)

- Power Supply – (13 x 4.2 x 2.2)
- Arduinos and Protoshields – (7.5 x 5.5 x 2.3)
- Geiger Counters – (12 x 4.75 x 2.5)
- Flight Tube – ⅜” ID, ⅝” OD
- Heating Element – (8 x 5)
- 9 Volt Batteries – (2.65 x 4.85 x 1.75)

It was determined that the interior of the payload must be (20cm x 20cm x 7.2cm) to allow all the components to fit inside. In order to ensure the survival of the components, it was determined that the walls of the payload must have a thickness of 1.4 cm. The size of the base and ceiling had dimensions of (22.8cm x 22.8cm), again with a depth of 1.4 cm. Each of the four walls had a length of 21.4cm and height of 7.2cm, this would allow the walls to have an interlocking system, each pushing on one another.

A major concern was the potential for the components to move around and make contact with each other. Therefore a mounting system inside the housing was required. As seen in Figure x, two pieces of plywood (8.5cm x 19.5cm x 1cm) (Figure x) were used to mount each component into place.
Another requirement of the payload was that the switches must be located on the exterior of the housing. As seen in Figure 4 each switch is 3cm x 1.4cm x 1.5cm. One hole (3cm x 2.8cm) was cut in the side wall of the foam board to allow both switches to fit through. The last major requirement of the housing subsystem is the flight tubing. DemoSat required that the flight tube must be plastic and have a ⅜″ inner diameter. A piece of PVC tubing, with a height of 15.24cm, was used in order to allow an excess of 1″ (2.54 cm) to stick out on either side of the closed payload. The exterior diameter of the flight tube was ⅝″. Therefore, a ⅝″ hole was drilled through the ceiling and base of the payload. In order to secure the flight tube, a locking system of a washer and paperclip (acting as a cotter pin) was used. The washer had a ⅝″ interior diameter, allowing the bottom of the PVC pipe to fit through. These pieces were then securely attached to the base of the housing (Figure x). Lastly, a paperclip was pushed through the tubing at a distance of about 1 cm below the washer. The paper clip was then bent backwards to act like a cotter pin (Figure x).

**Geiger Counters and Moon Dust Simulant**

![Figure 9: Isometric Sketch of Geiger Counter Assembly](image)

**Function**

The Geiger counters (Fig. x) and moon dust simulant (Fig. x) were the focal points of the entire system. A Geiger counter is a device that measures radiation levels and can generally detect alpha, beta, and gamma particles. By utilizing this device, the team was able to monitor the radiation levels as the payload ascends into the atmosphere. As previously mentioned, the payload consisted of two Geiger counters, one shielded by the moon dust simulant, and one open to the atmosphere. The moon dust simulant, developed at Colorado School of Mines, will block
the sensor of the Geiger counter from the atmosphere, which means the shielded Geiger should end up having lower radiation readings than the exposed Geiger at each altitude.

**Design**

For the initial design, the team had the option of choosing between unidirectional and omnidirectional Geiger counters. The main difference between these two types of Geiger counters is the amount of data they are able to sense. For the unidirectional Geiger counter, it is equipped with a sensor that can only detect radiation in one direction. On the other hand, the omnidirectional Geiger counter is capable of picking up radiation particles in all three hundred and sixty degrees. In the next paragraph, it will be explained in greater detail how the unidirectional Geiger counter was determined to be the best option.

Both of the Geiger counters weighed about the same, so that aspect did not factor into the decision of which to choose. They also both had about the same minimum operating temperature, so that was not a factor. Since the unidirectional Geiger counter was made by the same company, SparkFun, as the Arduino Board, it was determined it would be more compatible with the Arduino than the omnidirectional Geiger counter would be. Also, if the omnidirectional Geiger were to be used, there would have been an overwhelming amount of data to process, seeing as it records in every direction. With the unidirectional Geiger, there will still be a sufficient amount of data obtained needed to draw conclusions from the flight. Also, probably the most important factor in choosing which Geiger counter to use, was determining which one would be the easiest to shield with the moon dust simulant. Since the sensor on the unidirectional Geiger counter was much smaller than the one on the omnidirectional Geiger counter, it was decided that it would be best to use the unidirectional Geiger counter. By being able to use less moon dust simulant for shielding, the weight of the payload was minimized and the extra weight was utilized for increased structural integrity. The only con for using the unidirectional Geiger counter instead of the omnidirectional one was the price difference. The unidirectional one costs roughly forty dollars more than the omnidirectional one. But price was not an important factor in the decision because it was more important to be able to adequately shield the sensor, and log reliable data.

**Specifications**

The Geiger counter itself is approximately 12 cm long, 4.75 cm wide, and stands about 3.5 cm tall. Each weighs 0.048 kg. The shielded Geiger counter will have some moon dust attached to it, but the weight of it is so small that it is considered negligible. Though they do not weigh too much, they are some of the larger components in the payload, and will take up a large portion of space. Because of this, the Geiger counters were one of the main driving forces in the payload design. Thought was put into making sure the Geiger counters would fit efficiently in the payload without causing too much wasted space. Some other features of the unidirectional Geiger counter include five volt logic, a total current of thirty milliamps, an activation voltage of five hundred sixty volts, a CPM limit of 100 Hz, and a minimum operating temperature of negative forty degrees Celsius.
**Arduino Assembly**

![SparkFun RedBoard](image)

*Figure 10: SparkFun RedBoard*

**Function**
The motherboard must be capable of processing large amounts of data, as well house and integrate with the data logger and altimeter. The board must provide power to all of the components including the Geiger counters. Finally programming the board must be simple enough for a novice, as well capable enough to perform the desired functions.

During the flight, the balloon could experience strong crosswinds, so the motherboard must be able to withstand turbulence, as well remain connected to the other components. Another factor is that the Arduino must be able to operate in extremely cold temperatures.

The two possibilities considered for the motherboard were Raspberry PI and Arduino boards. Both are easy to program, and have a large support basis for any issues with custom programs. The altimeter, data logger, and Geiger counters can be interfaced with both boards. Ultimately, the Arduino was chosen to be the best option to be the brains of the payload.

**Design**
The Arduino assembly consists of an altimeter, OpenLog data logger, SparkFun protoshield, and SparkFun RedBoard. The purpose of the assembly is to read the data from the Geiger Counters, correlate the readings to the current time, pressure, altitude, and temperature in one second intervals. After the data is read, it is stored in a microSD card that is housed in the OpenLog.

Interfaces

- SparkFun RedBoard:
  - The “brain” of the circuit that stores the operating program.
  - Interfaces and operates the connected components.
  - Provides a consistent 5 volts or 3.3 volts to the circuit from a 11-12 input voltage.
- Protoshield:
  - Connects to the corresponding inputs through mounting pins.
  - Provides a solderable circuit board that connects to the Arduino Board.
- Open Log:
  - Houses a microSD card, varying in memory.
  - Records all serial inputs and converts or “prints” the data to a excel spreadsheet.
- Altimeter:
  - Measures the barometric pressure and through an algorithm calculates the altitude as well records the temperature.

Specifications

The design is based on limitations of the circuitry itself, as well as the required ports from the programming. The altimeter is soldered to the protoshield, and powered by the 3.3 V connection. The OpenLog was soldered to the protoshield and connected to the corresponding inputs and powered by a 5 V connection. The Geiger counter was connected to specific ports as well. The protoshield was then inserted onto the RedBoard, where mounting pins were lined up with the input ports on the RedBoard. Finally through a DC plug, the RedBoard was connected to the 12 V battery. This assembly is similar for the second Arduino board.

Dimensions

The weight of the protoshield, data logger, and altimeter is combined with the RedBoard due to the accuracy of the scales used. (W x L x H)(cm)

- RedBoard – (5.5 x 7.5 x 1.0)
- Protoshield – (5.5 x 6.5 x 1.3)
Heating

Figure 11: Heating Element

Function
The heating system of the payload was crucial for keeping the electronic components from freezing at high altitudes. When constructing the housing of the payload, the temperature of the surroundings was also considered. The payload was constructed of 1.4 cm thick foam board, wrapped in aluminum tape. All walls, including the lid and base, were siliconed together to ensure the internal components would not be exposed to the outside atmosphere. Because of this, the inside of the payload was be considered to be a closed system. However, at high altitudes, heat loss is still a considerable factor. Two circuits were utilized to ensure the inside of the housing stays sufficiently heated. The heating system was separate from the Arduino circuit, equipped with its own power supply. Below lays out how the heating system is integrated with the other components.

Interfaces

* Power Supply and Computational/Testing
  * One power source was provided for the data recording portion of the test.
* Power Supply and Heating
  * The heating system was powered by its own separate power source.
* Power Supply and Structure
○ The two power supplies used for the experiment were securely mounted to the payload structure and did not shift during flight.

- Heating and Structure
  ○ The heating system was securely mounted to the payload structure and did not shift during flight.
- Power Supply and Switch to Structure
  ○ Both power sources were connected to a switch secured to the outside of the payload. This allowed the systems to be turned on prior to launch.

**Design**

Several options were considered for heating the payload. Team MEAPS, who constructed a payload for the DemoSat Program in 2015, placed a heating pad inside their payload to keep their components warm. Reusing this heating pad was an option considered, but ultimately the pad did not produce enough heat to adequately insulate the electrical components from the cold temperatures in the atmosphere. Research was also done into PRC ceramic heaters, which seemed to be the most effective form of heating for the payload. Lastly, several resistors constructed in a series circuit was considered. As current is passed through the resistors, they radiate heat. To determine the best option, a design matrix was used (Table x). The factors considered in the design were price, availability, and effectiveness. The PRC heaters were the most expensive of the options considered, and the heating pad slightly behind it in price. The resistors were considered the most cost effective of the options, and they also were the most accessible. The heating pad that Team MEAPS used was tested, and it was determined that it was no longer functional. The PRC heaters proved to be the most difficult to obtain. All of the PRC heaters found were located in China, and had increasingly long shipping times. Lastly it was determined how effective each method would be for supplying heat to the payload. When considering this, heat distribution was taken largely into account. Because of the heating pad’s relatively small size (roughly 10 cm * 5 cm), it was determined that it would be the least effective at dispersing heat throughout the payload. The PRC heaters were considered to be the most effective and have operating temperatures of up to 200 °C. The resistors were considered slightly more effective than the heating pad because they could be placed freely inside the payload. Summing the categories in Table 1, it was determined that the resistors would be the most effective means of heating the payload.

<table>
<thead>
<tr>
<th>Designs Considered</th>
<th>Price</th>
<th>Availability</th>
<th>Effectiveness</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heat Pad</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>PRC Ceramic Heaters</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Resistors</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>

*Note: Ratings are on a 1 to 5 scale with 5 being the highest.*
Specifications

It was determined that a large power source be used for powering the Geiger counters and Arduino boards. The power source needed to last the entirety of the flight, and provide sufficient power to the various components. Team MEAPS had previously used a 12 Volt R/C battery to power their payload, which worked sufficiently for them. The same battery was then utilized for this experiment. The team determined that the Arduino boards (with the protoshield(s) and altimeters) weighed more than the two Geiger counters. The battery was easily the heaviest component in the payload, and was placed on the opposite side of the Arduinos. The battery, and testing and computational systems can be seen in Figure 7 and in the circuit diagram. The remaining 9 V batteries were placed on the same side as the Arduinos, opposite the 12 V.

The heating system consisted of three, 4 Ω resistors configured in series. The circuit diagram can be seen in Figure X. These resistors were placed onto a small breadboard, and mounted in the payload by placing screws through the various holes in the board. This system was powered by three 9 V batteries, and were connected to a switch located on the outside of the payload.

Assembly and Operation

Housing

After the exact dimensions of all the housing pieces were determined, assembly could begin. First, for the foam board exterior, silicone was determined to be the best construction sealant. All pieces were aligned in place and securely joined with silicon sealant. The four walls (21.4cm x 1.4cm x 7.2cm each) were attached, using the silicone, to the base of the payload (dimensions 22.8cm x 22.8cm x 1.4cm). This frame created an interior space of 20cm x 20cm x 7.2cm. Next, the two plywood pieces were secured with silicone into two opposing sides, leaving a small gap in between the two pieces. For the flight tube, two holes were drilled into the top and bottom of the housing, with a diameter of ⅝”, then secured with silicone, leaving approximately 1” (2.54
cm) on top and bottom. The flight tube was mounted with a washer that was pushed over the tubing, and siliconed to the bottom of the housing. Lastly, a paper clip was punched through the pipe and bent back to secure the washer. The roof of the payload was not attached until flight day in order to make sure everything was properly working. The roof was secured with aluminium tape with the PVC pipe threaded through the hole.

**Interior Components**

The Geiger counters and batteries themselves required no assembly, as they came fully functional. However, the moon dust simulant required some work to assemble onto the Geiger counter. Initially, the idea was to completely cover the entire Geiger counter with the moon dust simulant, but it was quickly realized that doing so would add unnecessary weight to the payload. It was determined that only the small sensor on the Geiger Counter needed to be covered in order to obtain efficient data from the flight. Travel size spray bottle caps were used to put the moon dust simulant in, and shield the sensor. The cap was filled with moon dust simulant to a depth of about 1 cm, and then taped onto the sensor to ensure that the cap would not fall off.

The altimeter and data logger was soldered to the protoshield, which connected to the RedBoard through mounting pins as seen in. The wires were connected to the required pins for each component defined by the programming. The components were powered in parallel to the 5 V, 3.3 V and ground outputs respectively. The Geiger counters were connected to the D4 and D5 inputs due to limitations of the libraries (device supported software). There were two RedBoards in the experiment to ensure the data was collected properly and efficiently.

After all of the components were wired to one another, they were mounted securely to the two pieces of plywood inside the payload. The Arduino boards and Geiger counters were screwed into place on either side of the payload. The 12 V battery was mounted using zip ties, and placed as close to the flight string as possible. The three 9 V batteries were secured into the middle of the payload with silicone calk. The batteries were all mounted in the middle of the payload to ensure the payload was balanced properly.

**Cost Summary**

A detailed summary of the various costs can be seen in appendices. The most expensive components of the payload were the Geiger counters. These had an individual cost of $149.00 and a total cost of roughly $300.00. Two Geiger counters were purchased instead of reusing the ones from the previous years to ensure quality data collection. The total cost of the designed payload was roughly $465.00. The arduino boards with the attached altimeters and protoshield(s) had a rough cost of $60.00.
### Table 3: Overall Cost Summary

<table>
<thead>
<tr>
<th>Material</th>
<th>Source</th>
<th>Amount</th>
<th>Unit Cost</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geiger Counter</td>
<td>SparkFun</td>
<td>2</td>
<td>$149.45</td>
<td>$299.90</td>
</tr>
<tr>
<td>Protoshield kit</td>
<td>SparkFun</td>
<td>1</td>
<td>$9.95</td>
<td>$9.95</td>
</tr>
<tr>
<td>Arduino Board #1</td>
<td>SparkFun</td>
<td>1</td>
<td>$19.95</td>
<td>$19.95</td>
</tr>
<tr>
<td>Arduino Board #2</td>
<td>SparkFun</td>
<td>1</td>
<td>Reused</td>
<td>$0</td>
</tr>
<tr>
<td>Altimeter #1</td>
<td>SparkFun</td>
<td>1</td>
<td>$14.95</td>
<td>$14.95</td>
</tr>
<tr>
<td>Altimeter #2</td>
<td>SparkFun</td>
<td>1</td>
<td>Reused</td>
<td>$0</td>
</tr>
<tr>
<td>Date Logger #1</td>
<td>SparkFun</td>
<td>1</td>
<td>$24.95</td>
<td>$24.95</td>
</tr>
<tr>
<td>Data Logger #2</td>
<td>SparkFun</td>
<td>1</td>
<td>Reused</td>
<td>$0</td>
</tr>
<tr>
<td>12-Volt RC Battery</td>
<td>Team MEAPS</td>
<td>1</td>
<td>Reused</td>
<td>$0</td>
</tr>
<tr>
<td>Foam Board</td>
<td>Home Depot</td>
<td>1</td>
<td>$11.97</td>
<td>$11.97</td>
</tr>
<tr>
<td>Micro SD card (8 GB)</td>
<td>Team MEAPS</td>
<td>2</td>
<td>Reused</td>
<td>$0</td>
</tr>
<tr>
<td>9 Volt Battery</td>
<td>Radioshack</td>
<td>3</td>
<td>$4.99</td>
<td>$14.97</td>
</tr>
<tr>
<td>RC Charger</td>
<td>Action RC Aircraft Center</td>
<td>1</td>
<td>$29.99</td>
<td>$29.99</td>
</tr>
<tr>
<td>⅜” Tubing</td>
<td>Home Depot</td>
<td>1</td>
<td>$2.79</td>
<td>$2.79</td>
</tr>
<tr>
<td>Screws &amp; Washers</td>
<td>Home Depot</td>
<td>50</td>
<td>$.15</td>
<td>$7.49</td>
</tr>
<tr>
<td>Foil Tape</td>
<td>Home Depot</td>
<td>1</td>
<td>$15.32</td>
<td>$15.32</td>
</tr>
<tr>
<td>Silicone</td>
<td>Home Depot</td>
<td>1</td>
<td>$5.92</td>
<td>$5.92</td>
</tr>
<tr>
<td>Caulk Gun</td>
<td>Home Depot</td>
<td>1</td>
<td>$5.02</td>
<td>$5.02</td>
</tr>
<tr>
<td>4 Ω Ceramic Resistor</td>
<td>Parts Express</td>
<td>3</td>
<td>$0.68</td>
<td>$2.04</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>$465.21</strong></td>
</tr>
</tbody>
</table>
Testing

Rigorous tests were performed on the payload to ensure its integrity, and to provide the team with a better understanding on how to construct the payload to ensure its survival. The following paragraphs explain in detail the types of tests that were performed.

Drop Test
To ensure the housing of the payload would protect and withstand the impact of the landing, two tests were performed. The first of these being the drop test. To guarantee the survival of the electrical components, a 5.44 kg rock (much heavier than the weight of the components, which were a mere 0.612 kg) was placed inside the housing of the payload and dropped several times at different angles from a height of 16ft.

Whip Test
To ensure that the payload would not disconnect from the flight string, a whip test was performed. For this, a string was threaded through the PVC pipe attached to the payload and secured on its end. The payload was then whipped around in ways that would most closely simulate movements of the payload during the flight.

Cold Test
To test the functionality of the electrical components, a cold test was performed to simulate the extreme temperatures that would be encountered during the flight. To do this, the components were placed into the housing and the circuits turned on. The payload was then placed into a cooler filled with dry ice and the lid was then closed. After letting the payload sit in the cooler for about three hours, it was removed and assessed for malfunctions.

Sensor Test
The testing for the two Geiger Counters was quick and easy. Once the code was created, the Geiger Counters were tested by shining a UV light on Vaseline glass beads which emits beta and gamma particles. The beads were placed in front Geiger sensor to see if the sensor was able to pick up readings from the radiation being given off from the beads.

Mission Life Test
Since the flight would last approximately three hours, it was important to ensure that the batteries that powered the circuits would be able to provide sufficient power throughout the duration of the flight. To test this, the batteries were simply turned on, and hooked up to the circuit. The circuit was left running until the battery finally ran out of power.

Programming and Functionality Test
Testing for this subsystem required the Altimeter, data logger, Geiger Counters, and Protoshield fully assembled. Vaseline glass will to emit beta and gamma particles to test the programming
and functionality of the circuit as well. If the circuit is assembled correctly and there are no bugs in the programming, the emissions from the vaseline glass will be recorded by the Geiger Counters, transmitted to the motherboard and recorded by the data logger.

**Protoshield Assembly Test**
The Protoshield was tested on how well it could hold the other components and the ease with which it was able to mount onto the Arduino.

**Data Logging Test**
The data logger was tested on how well it could record the input data at different sample rates, as well different inputs.

**Altimeter Test**
The Altimeter was tested at different altitudes and pressures to ensure the accuracy of the readings and to verify the pressure readings matched weather station recorded pressures.

<table>
<thead>
<tr>
<th>Table 4: Testing Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Performed</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Drop Test</td>
</tr>
<tr>
<td>Whip Test</td>
</tr>
<tr>
<td>Cold Test</td>
</tr>
<tr>
<td>Sensor Test</td>
</tr>
<tr>
<td>Mission Life Test</td>
</tr>
<tr>
<td>Programming and Functionality Test</td>
</tr>
<tr>
<td>Protoshield Assembly Test</td>
</tr>
<tr>
<td>Data Logging Test</td>
</tr>
<tr>
<td>Altimeter Test</td>
</tr>
</tbody>
</table>
Mission Launch

On April 9th, 2016, the COSGC DemoSat sponsored flight occurred. The launch took place in Eaton, Colorado, at Eaton Middle School. On the launch date, the upon arrival to the site at 5:55 A.M., the payloads were checked in and given a final weighing. The payloads were then strung up to a high altitude weather balloon, which ascended to an elevation of approximately 89,000 feet (27127.2 meters). Team Cosmic Shield’s payload measured how well the Colorado School of Mines moon dust simulant would be able to shield cosmic radiation. The data collected from the flight is laid out in the proceeding section.

Results

The two geiger counters effectively read data throughout the flight. A summary of the flight data can be seen in Table 1. The initial altitude and temperature was 1459.8 meters and 53.8 ºF. The max altitude reached was 17644 meters with a temperature of 46.9 ºF. The lowest temperature reached was taken shortly after launch, and was 38.1 degF. Both Geiger Counters had an initial reading of 60 Counts Per Minute (CPM), or 0.1 MicroSieverts (µSv). The control Geiger counter had a final reading of 4503 µSv and the shielded Geiger counter had a final reading of 4400.3 (µSv). The average values of Geiger 1 and Geiger 2 readings were 1518 and 1459 µSv, respectively.

<table>
<thead>
<tr>
<th>Table 5: Summary of Geiger and Altimeter Readings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Initial Altitude</td>
</tr>
<tr>
<td>Max Altitude</td>
</tr>
<tr>
<td>Max</td>
</tr>
<tr>
<td>Min</td>
</tr>
<tr>
<td>Mean</td>
</tr>
</tbody>
</table>

Analysis and Error

As previously mentioned, the shielded Geiger counter had approximately 1 cm of simulant between the unidirectional sensor and the cap. The control Geiger counter was also capped so there were no unaccounted variables. It is likely that the cap could have shielded the sensor from undetermined amounts of radiation. The difference in readings between the two Geiger counters showed that the simulant effectively shielded the sensor from radiation. It can be seen in Figure 13 that the shielded Geiger counter (Blue) read lower levels of radiation than the unshielded one at high altitudes. The two Geiger counters started at the same initial radiation value, which was expected. As the balloon rose in altitude, the atmosphere provided smaller amounts of protection from the cosmic radiation. As such, the Geiger counters were exposed to larger quantities and the deviation in the two readings began to increase. Table 5 shows the average radiation level for Geiger 2 was lower than that of Geiger 1. There was some error involved with Geiger 1 during
the flight, in that the SD card become loose during the flight and the logged data restarted several times throughout the flight. This resulted in altitude and Geiger readings that were incorrect. The temperature values of Geiger 1 were also affected by the SD card, and as such, the altitude and temperature readings from Geiger 2 were used as the correct values. The Geiger counters could only read to a max value of roughly 32,000 CPM before the data experienced wrap around and reset near -32,000 CPM. This may have been caused by the data being logged in one second intervals resulting in a data overload. The wrap around was corrected by taking the absolute difference between the Geiger readings and summing them to reach the max values that were then converted into micro Sieverts. In regards to Geiger 2, the cap was not tightly packed with the moon dust simulant. Therefore, it is possible that the simulant could have shifted during flight, which would affect the Geiger readings. The data obtained during the descent of the flight was considered unnecessary, and as such was discarded. There may have also been errors in the equipment. Some of the components were reused from the 2015 DemoSat team. And as always, human error was also a contributing factor.
Conclusion

Team Cosmic Shield was contracted with this project on January 27th, 2016. Throughout the proceeding months, much planning, brainstorming, and fabricating was done. The project was culminated on April 9th, 2016, when the payload was ascended into the atmosphere, attached to a high altitude balloon. The data compiled from the flight was then analyzed, and conclusions were drawn from them. By analyzing the results, it can be determined that the moon dust simulant, developed at Colorado School of Mines, is capable of shielding cosmic radiation to some extent. To what extent is still somewhat unknown, because this project was conducted on such a small scale. To get a better sense of how effective its shielding properties are, the simulant should be tested on a large scale. Doing so would render much more accurate data than what was collected from this experiment.

Application

It is Team Cosmic Shield’s hope that this data can be further used by NASA in determining whether or not the in-situ materials on the Lunar and Martian surfaces could be used to create habitable stations that can effectively shield radiation. It is hard to determine how much regolith should be used to appropriately shield a said amount of radiation in space. However it can be assumed that it could be effectively used based on the level of thickness provided.

Improvements

The experiment could be improved in a variety of ways. One way to increase the accuracy of the findings would be to use omnidirectional sensors. This way the radiation levels can be measured in all directions. Using three Geiger counters could also prove to be effective. The third Geiger counter could measure the radiation without any cap protecting the sensor. This would provide data on how much radiation is shielded by the cap, and could further show how effective the Lunar dust is at shielding the sensors. It is also believed that a more powerful Arduino with more ports would work better in running the Geiger counters. This would result in all the Geigers running on the same code, and would have only one altimeter. If the Arduino were to malfunction, it would result in errors across all the Geiger counters. The heating circuit could also be improved. The initial design was centralized near the center of the payload. It could prove to be more effective if the resistors, or heating elements used, were spread out evenly throughout the payload.
Bibliography


Appendix

Figure 14: Schematic of Final Design
Figure 15: Arduino Assembly.
Figure 16: Geiger Counter Assembly with Shielding.
Arduino Code

//Libraries
#include <SoftwareSerial.h>
#include <Wire.h>
#include <SparkFunMPL3115A2.h>
#include <Time.h>

//Private Libraries
#include "OpenLog.h"
#include "Altimeter.h"
#include "geiger.h"

//Pin definitions
#define geigRX 3 // Geiger Counter Arduino TX pin (->RX on Geiger Board)
#define geigTX 2 // Geiger Counter Arduino RX pin (->TX on Geiger Board)
#define geig2RX 4 // Geiger Counter 2 Arduino TX pin (->RX on Geiger Board)
#define geig2TX 5 // Geiger Counter 2 Arduino RX pin (->TX on Geiger Board)
#define resetOpenLog 8 // OpenLog Reset Pin

//Create serial for geigers and openlog
SoftwareSerial geiger(geigTX, geigRX); //Arduino TX (->RX on Geiger) and RX (->TX on Geiger)
SoftwareSerial geiger2(geig2TX, geig2RX); //Arduino TX (->RX on Geiger) and RX (->TX on Geiger)

//Variables
unsigned int count = 0;
unsigned int count2 = 0;
long unsigned int countStartTime = 0;
long unsigned int countStartTime2 = 0;
float elevation = 1475; // Local elevation
float pressure_offset; // Used for calibration
char buff[50];
int counter;
int countsPerMinute;
float coefficientOfConversion = 140.0 / 60000.0;
float sV = 0;
int countsPerMinute2;
void setup()
{
    myPressure.begin();
    Serial.begin(9600);
    Serial.println();
    setupOpenLog();
    Serial.println();
    geiger.begin(9600);
    geiger2.begin(9600);
    Serial.flush();
    Serial.println("Power up!");
    Wire.begin(); // Join i2c bus
    myPressure.setModeAltimeter();
    myPressure.setOversampleRate(7);
    myPressure.enableEventFlags();
    pressure_offset = calibAlt();
    Serial.print("pressure offset");
    Serial.println(pressure_offset);
    OpenLog.println("CPM, ALT(m),Pressure(kPa),Temp(F),CPM(2),TIME(H:M)");
    Serial.println("Setup Complete!");
}

void loop()
{
    geiger_CPM();
    //OpenLog.print(",,");
    geiger2_CPM();
}

void setupOpenLog()
{
    OpenLog.begin(9600);
    pinMode(resetOpenLog, OUTPUT);
    digitalWrite(resetOpenLog, LOW);
delay(100);
digitalWrite(resetOpenLog, HIGH);

//OpenLog File Setup
while (1)
{
  //pinMode(12, Output);
  //digitalWrite(12,TRUE);

  if (OpenLog.available())
    if (OpenLog.read() == '<') break;
}

//Send three control z to enter OpenLog command mode
//This is how Arduino v0022 used to do it. Doesn’t work with v1.0
//Serial.print(byte(26));
//Serial.print(byte(26));
//Serial.print(byte(26));
//Works with Arduino v1.0
OpenLog.write(26);
OpenLog.write(26);
OpenLog.write(26);

//Wait for OpenLog to respond with '>' to indicate we are in command mode
while (1)
{
  if (OpenLog.available())
    if (OpenLog.read() == '>') break;
}

Serial.println("Command Mode Entered");

/*
 * sprintf(buff, "new test % 03d.csv");
 * OpenLog.print(buff);
 * //regular println works with v2.51 and above
 * OpenLog.write(13);
 */

sprintf(buff, "append data.csv");
OpenLog.print(buff);
OpenLog.write(13);

//Wait for OpenLog to indicate file is open and ready for writing
while (1)
{
  if (OpenLog.available())
    if (OpenLog.read() == '<') break;
void geiger_LOG()
{
    OpenLog.println("Open Log Ready for Data");
    //digitalWrite(12,FALSE);
}

void geiger_LOG2()
{
    OpenLog.println(countsPerMinute2);
    OpenLog.println(""," ");
    OpenLog.println(myPressure.readAltitude(), 1);
    OpenLog.println(""," ");
    OpenLog.println(myPressure.readPressure(), 1);
    OpenLog.println(""," ");
    OpenLog.println(myPressure.readTemp(), 1);
    OpenLog.println(""," ");
    OpenLog.println(hour());
    OpenLog.println("":" ");
    OpenLog.println(minute());
    OpenLog.println("":" ");
    OpenLog.println(second());
}

float calibAlt()
{
    //Calibrate Altimeter
Serial.println("Calibrating Altimeter....Please wait.");
delay(3000); //Allow for a realitvely level measurement to be made
float pressure = myPressure.readPressure(); //Read pressure from Altimeter
float elevation_offset = pressure - 101325 * pow((1 - (2.2557e-5) * elevation), 5.25588); //Calibrate an intital offset for the day
Serial.println("Calibration Complete!");
return (elevation_offset);

//Environment

//Geiger Counter

void geiger_CPM()
{
    while (geiger.available() > 0)
    {
        int inChar = geiger.read();
        if (inChar == '0' || inChar == '1')
        {
            count++;
        }
    }

    unsigned int currentTime = millis();
    unsigned int elapsedTime = currentTime - countStartTime;
    if (elapsedTime >= 1000)
    {
        countsPerMinute = 60 * count;
        sV = (float) countsPerMinute * coefficientOfConversion;
        count = 0;
        countStartTime = now()*1000;
        float inpressure = myPressure.readPressure();
        delay(30);
        float intemp = myPressure.readTemp();
        Serial.print("CPM: ");
        Serial.print(countsPerMinute);
        Serial.print(" ");
        Serial.print("Time (H: M: S) : ");
        Serial.print(hour());
        Serial.print(":");
        Serial.print(":");
```cpp
Serial.print(minute());
Serial.print(" ");
Serial.println(second());
Serial.print("\u03BC\text{Sv} / \text{hr}: ");
Serial.println(sV, 5);
Serial.print("Pressure (kpa): ");
Serial.println(myPressure.readPressure(), 1);
Serial.print("Altitude (m): ");
Serial.println(myPressure.readAltitude(), 1);
Serial.print("Temperature (F): ");
Serial.println(myPressure.readTemp(), 1);
geiger_LOG();
}
}

//GeigerCounter2 int countsPerMinute2;
float sV2 = 0;
void geiger2_CPM()
{
    while (geiger2.available() > 0)
    {
        int inChar = geiger2.read();
        if ((inChar == '0' || inChar == '1'))
        {
            count2++;
        }
    }

    unsigned int currentTime2 = millis();
    unsigned int elapsedTime2 = currentTime2 - countStartTime2;
    if (elapsedTime2 >= 1000) {
        countsPerMinute2 = 60*count2;
        sV2 = (float) countsPerMinute2 * coefficientOfConversion;
        int count2 = 0;
        countStartTime2 = now()*1000;
        float inpressure = myPressure.readPressure();
        delay(30);
        float intemp = myPressure.readTemp();
        //delay(50); problems logging data from both counters;
        //delay possible solution
        Serial.print("CPM(2): ");
    }
```
Serial.println(countsPerMinute2);
Serial.print("uSv / hr(2): ");
Serial.println(sV2, 5);
geiger_LOG2();
}