High altitude soil temperature/moisture monitoring

April 11, 2019

Team: Carson Fiechtner, Samuel Croft, and Tyler Vendetti

Mentor: Dr. Kevin Negus

Sponsor: Dr. Martha Apple

Contents

[1. Problem Statement 2](#_Toc37502756)

[1.1 Introduction 2](#_Toc37502757)

[1.2 Background 2](#_Toc37502758)

[2. Project Design 2](#_Toc37502759)

[2.1 Components 2](#_Toc37502760)

[2.1.1 SHT-10 Soil Temperature/Moisture Sensors 2](#_Toc37502761)

[2.1.2 Arduino Uno Rev 3 3](#_Toc37502762)

[2.1.3 RockBLOCK 4](#_Toc37502763)

[2.1.4 Rocketscream 5](#_Toc37502764)

[2.1.5 SD Card 5](#_Toc37502765)

[2.2 Scalability 5](#_Toc37502766)

[2.3 Power 6](#_Toc37502767)

[3. Constraints 7](#_Toc37502768)

[3.1 Product Constraints 7](#_Toc37502769)

[3.2 Project Constraints 7](#_Toc37502770)

[4. Project Management 7](#_Toc37502771)

[5. Next Steps 7](#_Toc37502772)

[6. Conclusion 7](#_Toc37502773)

[7. References 7](#_Toc37502774)

# Abstract

The biology department has one method for monitoring soil conditions in the Pintler Mountains, which is placing isolated sensor units underground and taking note of where they are buried for later retrieval. This technique is imperfect because the current sensors can’t provide humidity readings, data can only be collected by retrieving the sensors, and finding sensors can prove difficult. This method is especially time consuming because of the remote location. Each problem was evaluated individually, and a system was designed that would incorporate a solution for each. First, sensors capable of reading both temperature and humidity are used to capture all relevant data. Next, a host that can operate the sensors and is capable of long-term data storage coupled with a satellite modem that can transmit data without cellular coverage provide the core functionality we need. Finally, a separate controller that can manage power consumption is required for extended battery life.

# 1. Problem Statement

## 1.1 Introduction

Dr. Martha Apple is looking for ways to improve her high-altitude soil temperature monitoring setup at Goat Flat in the Pintler Mountains. Presently, Dr. Apple uses several HOBO Tidbit V2 Temperature Sensors [1], all of which log their own data individually. When Dr. Apple would like to retrieve that data, she must hike up to Goat Flat, locate each sensor through line-of-sight pictures and painted rocks, transfer the data to her ruggedized computer, and hike back down.

While there are several things that Dr. Apple would like to improve on, our team decided to focus on retrieving the sensor data remotely, removing the need for as many trips up to Goat Flat. Specifics on how we implemented remote communication can be found in the Project Design section below.

## 1.2 Background

As was mentioned earlier, Dr. Apple had multiple ideas for how to improve her high-altitude monitoring setup. Specifically, she had three requirements in mind, ranked in order of importance to Dr. Apple:

1. Add soil moisture monitoring capabilities

2. Remotely retrieve sensor data

3. Easier retrieval of temperature sensors

While we decided to focus on remote retrieval of sensor data, we did not overlook the other requirements. As you will see later in the report, we decided to use sensors that monitor both soil temperature and moisture, removing the need for separate sensors.

We also thought of a few ways to make finding the sensors less of a pain. Since our sensors send data remotely, there will be much less need to find the sensors on a regular basis. In the case of a sensor malfunction or dead battery, the sensors can be found by simply following the wire that leads from the data hub. Also, since everything connects to the data hub, it is no longer required to track each sensor individually.

# 2. Project Design

## 2.1 Components

In thinking about how we could create a great project that is still achievable, we strived to keep all aspects of the design as simple as possible. We’ve chosen wired sensors that are built to interface with a microcontroller [2]. We’ve chosen a power-efficient microcontroller [3] to ensure a long useful life. Perhaps most importantly, we’ve chosen a communication modem that is designed to work in the harshest, most remote locations and makes messaging through a microcontroller simple [4]. Details on each individual component of the system, as well as alternatives that were not chosen, can be found below.

### 2.1.1 SHT-10 Soil Temperature/Moisture Sensors

We chose to use wired SHT-10 combined soil temperature and moisture sensors for several reasons. First, using these sensors allows us to add soil moisture monitoring [2], which was our highest-priority request from Dr. Apple. Second, because these sensors are wired and not wireless, we believe they will be more reliable and less likely to unexpectedly fail or lose connection to their host. Finally, they even have a code library built specifically for our microcontroller, the Arduino, that should make integration relatively painless [2].

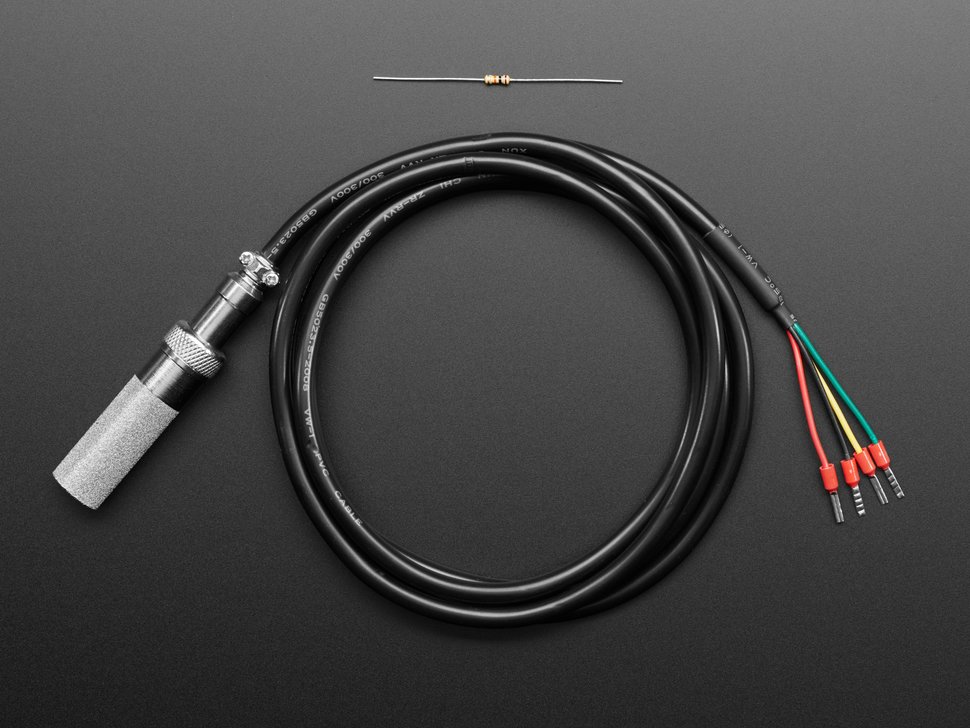


Figure 1. SHT-10 Soil Temperature/Moisture Sensor [2]

We considered numerous alternative sensors while making our decision. All the alternative sensors considered were wireless. The big benefit to using wireless sensors



Figure 2. ALTA Wireless Temperature Sensor

### 2.1.2 Arduino Uno Rev 3

Because we had a few frontrunners in the microcontroller space, we didn’t consider as many microcontrollers as we did soil sensors. Early in the design of our project, we had expected to use a Raspberry Pi as the microcontroller. However, after evaluating the Arduino and its libraries, it seemed to be the clear choice; it advertised intuitive interoperability with the chosen sensors, SD card module, and modem [3]. The Arduino IDE is easy to use, and Arduino has a number of open-source libraries that make most of the system interactions painless.

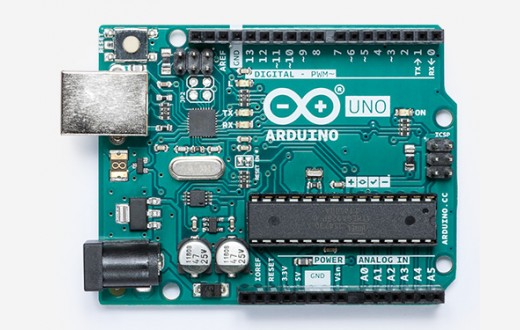


Figure 3. Arduino Uno Rev 3 [3]

### 2.1.3 RockBLOCK

The RockBLOCK was the standout choice in the decision for a communication modem. Built to make interfacing with a microcontroller easy, the RockBLOCK uses an Iridium modem to allow for satellite communication from anywhere in the world with a view of the sky [4]. It is also cheaper than the alternatives we had in mind [4][7][8], which made the choice that much easier.



Figure 4. RockBLOCK Iridium SatComm Module [4]

Before we discovered the RockBLOCK, we were considering modems from both Iridium and Globalstar. Iridium is designed to work world-wide and offers a starter kit for those new to satellite modems but was the most expensive choice and left communication setup with a microcontroller unclear [6]. Globalstar was less expensive but required a considerable amount of code before it could work and didn’t operate as efficiently in southern latitudes [8].

 Figure 5. Iridium Edge Demo Kit [7] Figure 6. Globalstar STX3 Satellite Transmitter [8]

### 2.1.4 Rocket Scream

Initially, the system was designed for the Arduino to go to sleep whenever it wasn’t reading from the sensors or transmitting data. However, during power analysis it was realized that the Arduino’s sleep mode power consumption was well over what we could handle to support a multi-year battery life. Instead, we decided to include a smaller, ultra-low-power controller that would manage power supplied to the Arduino.

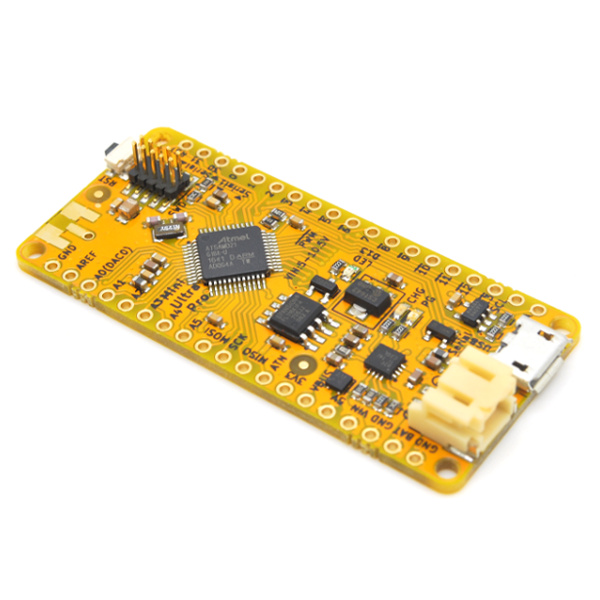


Figure 7. Rocket Scream Mini Ultra Pro V3 [11]

Operating on a real-time clock, the Rocket Scream will supply power to the Arduino on an adjustable time interval. Also, because the Rocket Scream’s sleep mode power usage is measured in terms of microamps, it still allows for the multi-year battery life required to make the system beneficial.

### 2.1.5 SD Card

Our initial system design didn’t include a storage system. Instead, it was expected that the Arduino’s internal EEPROM could serve as a temporary data storage between data transmissions. However, EEPROM on the Arduino Uno is only one kilobyte and cannot store text values, which would make storage unnecessarily tedious. Instead, we decided to integrate an SD card reader. The SD card provided an incredible amount of memory, while allowing for storing text. The SD reader and card used are shown below.

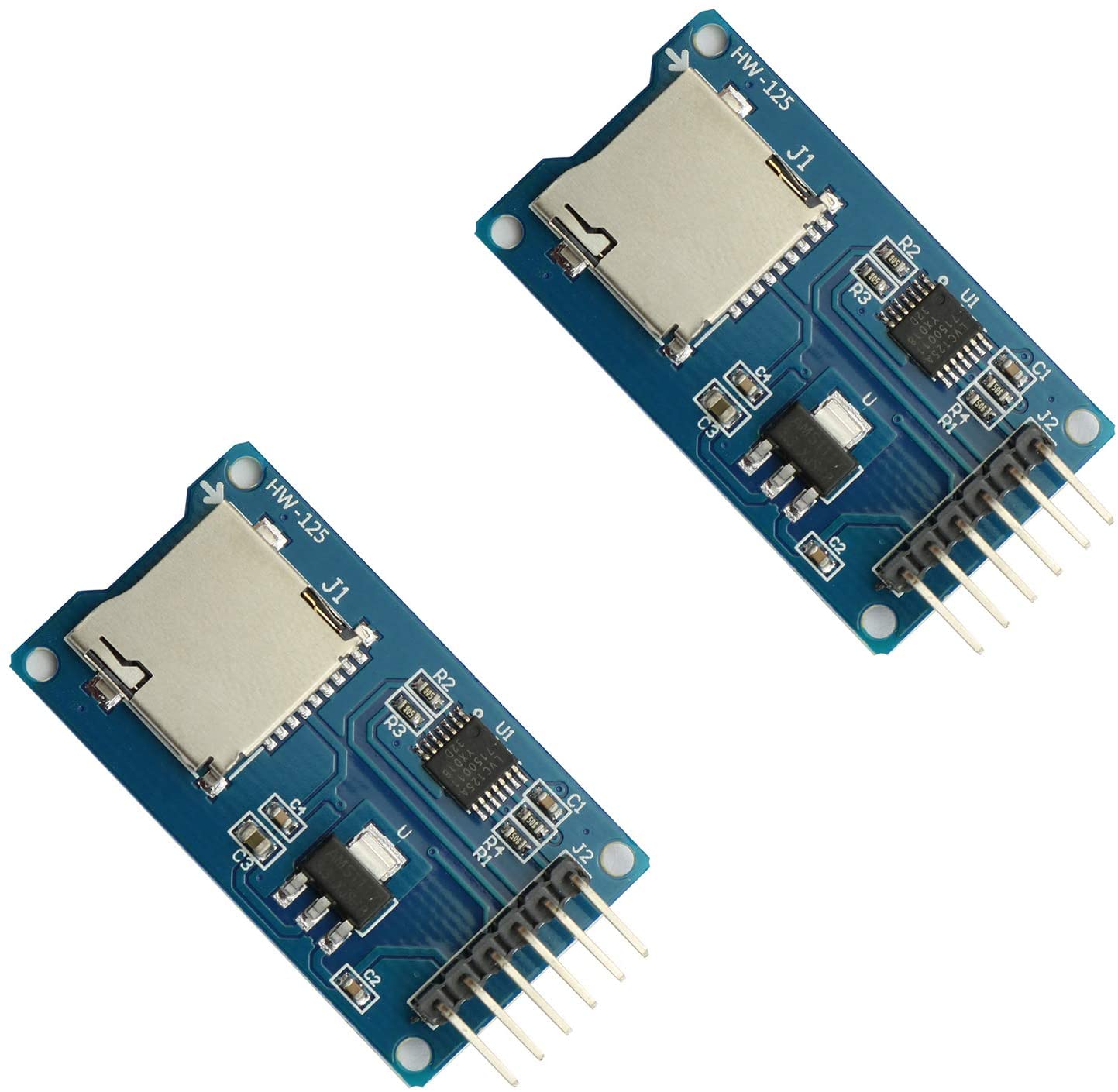
 

Figure 8. Arduino SD Card Reader [12] Figure 9. Low-Temp MicroSD Card [13]

The SD card and reader are both rated for as low as -40°C. Also, as the module enters sleep mode when inactive and reading from and writing to the card take well under a second, their impact on the system’s power consumption is negligible.

These components came together to complete our main requirement, enabling remote retrieval of sensor data.

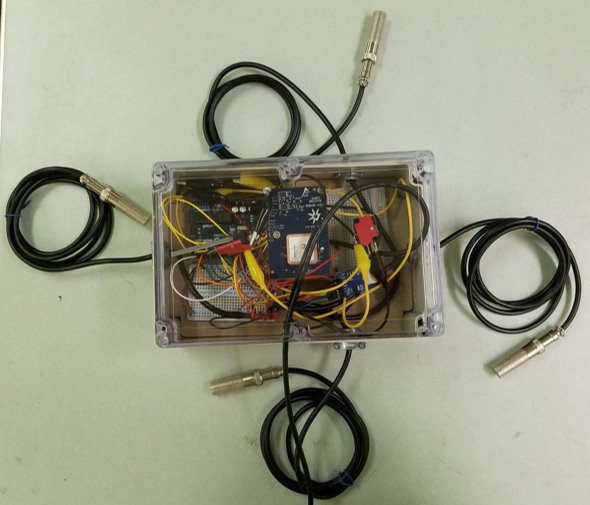
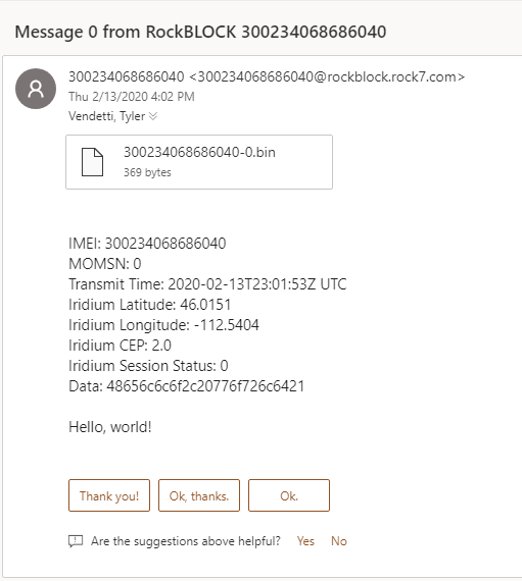


Figure 10. Completed Proof-of-Concept Figure 11. Message from RockBLOCK

## 2.2 Scalability

The sensors can also be multiplexed through a relatively inexpensive multiplexer. Below is a wiring diagram for a 16-pin multiplexer that could be used to increase the number of sensors that the system could utilize.

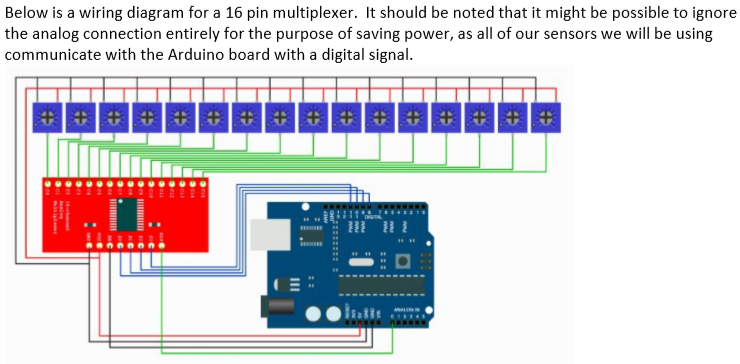


Figure 12. Wiring Diagram for Multiplexer

An Arduino Mega could also be substituted for the Uno to provide more inputs for sensors. The proof of concept developed did not require a multiplexer, but if more sensors are desired, either a multiplexer or an Arduino Mega could be integrated.

## 2.3 Power

Power was a focal point of the system, as it needs to operate for years on a single charge. This made power efficiency crucial to every aspect of the system. Below are the battery life calculations for the system.

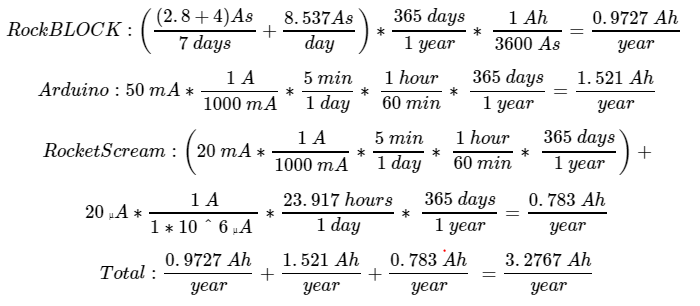


Figure 13. Power Usage Calculations for the System [3][4]

These calculations assume that the system is active for five minutes every day. This active time is heavily dependent on the time interval between readings and the frequency of transmission attempts. Active time can be adjusted according to user preferences.

# 3. Constraints

## 3.1 Product Constraints

The physical constraints of this project were set based on the location where our sponsor, Dr. Martha Apple conducts her research in the Pintler Mountains. This environment demands our electronics and enclosure to be able to endure harsh conditions in order to survive. These would be easiest to cover by discussing the electronics and enclosure separately.

Starting with the electronics, the main concern we had with each component was temperature. Everything needed to be able to operate at very low temperatures, and we decided that -40°C would be our baseline. Our sensors will have to exist outside the enclosure, and will be exposed directly to moisture, frost, and shifts in the soil. This meant that the sensors would have to be the most rugged of all the electronic components.

We also had to find a way to establish satellite communication where there was no cellular coverage. The only clear choice was using a satellite modem that could operate in the conditions mentioned, which left us with a limited number of options.

Finally, there is the constraint of battery life. We set a goal to give this device multiple years of operation on one charge, which meant every component had to be able to utilize a long-term low power consumption mode, or to be turned off entirely, and then able to wake up and function again. We would also need a rugged high capacity battery that could still fit within the enclosure and not suffer in performance due to the extreme low temperatures it may be exposed to.

Next, we have the enclosure and other non-electronic hardware. It is vital that the enclosure be completely watertight, and rugged enough to not break or rupture at any point. We also required a way to run the cables of our sensors through the box without compromising the integrity of the enclosure. Last, it had to be able to allow a signal to propagate through it when completely sealed, so metal as a material was not an option.

## 3.2 Project Constraints

The main constraints we faced were time and budget. At the beginning of the project we were given a budget of approximately $1,000.00 that was subject to change. We were also given a timeline of two semesters, or the equivalent of 8 months to complete the project, with smaller progress deadlines along the way.

# 4. Project Management

Project management tasks were handled mostly through Microsoft programs like OneDrive and Outlook. Team meetings were scheduled on Outlook, with invitations and reminders going out to anyone who needed them. Any relevant files or documentation were stored in a shared OneDrive folder to allow for asynchronous updates by team members.

On the topic of documentation, our project was primarily documented in an online notebook. This notebook includes meeting notes from each of our weekly meetings as well as notes from the research conducted throughout the project. The notebook is included with this report and expounds upon some of the design decisions mentioned earlier.

# 5. Next Steps

With this project, we ultimately want to create something that can be deployed in the field and produce useful data for years at a time. It needs to be a complete network of multiple large-scale versions of our current proof of concept device.

**5.1 Near Future**

To achieve this goal the first thing that needs to be done is to integrate the Rocket Scream board into the current device. The transistor switch will also need to be built within the device so that the power of the Arduino Uno can be regulated as intended. The power management of every component will rely upon the Rocket Scream, and so this must be done first.

Once the Rocket Scream has been integrated, the current consumption can be optimized, and a timestamp can be implemented in the message sent by the RockBLOCK. This will require a serial connection between the Rocket Scream and the Arduino Uno, which is further described in the HAST Rocket Scream Documentation document included with this report.

Once all is working and optimized, a battery or batteries can be sized and bought for remote operation.

**5.2 Far Future**

Once power consumption is ideal and everything is fully functional, the enclosure itself will need to be updated to be a fully functional prototype that meets the constraints of the project. The box itself will need to be upgraded to something much more rugged, such as a NEMA 4X box, and the gaskets will need to be changed as well. The current gasket solution was only added for proof of concept, which relieves strain on the cables, but does not in any way provide resistance to water or soil. Waterproof cable grip connectors seem promising and should be a good starting place.

Once the internals are working and the enclosure is properly ruggedized, it will be time to perform a field test. Once it is proven that the prototype can perform as required, the project can move to a larger scale.

The last few steps to reach the overall goal will ready the project to be deployed in the field. The first of these steps will be networking multiple versions of the prototype together so that the data collected can properly represent the scale that Dr. Apple’s research requires. It should be kept in mind that this will all need to be transported by foot and buried underground, so the physical size of this network will matter.

Lastly, it may be necessary to create an endpoint that will process and provide a useful display of the incoming data. It will be more reliable and convenient than email for viewing the collected data, and depending on how far this project goes, this may become a necessity.

# 6. Conclusion

Due to the semester ending the way it did, we were not able to complete everything we would have liked to. However, we did accomplish our main goal of ending with a proof of concept. We were also able to provide an abundance of documentation to pass over to the group who will continue this project. This documentation includes calculations, a project notebook (meeting notes), plans, research, code, and information regarding the devices used. Our research has shown that our plans are feasible, and we are confident that our work has paved the way for a prototype to be built. Design decisions thus far have been made with future improvements in mind. The only things remaining to complete are the transistor switch for an automated system, and improvements upon the current design (upgraded box, powering with a battery, weather proofing, multiplexing for more sensors, and so on).

# 7. References

[1] “HOBO UTBI-001 TidbiT v2 Water Temperature Data Logger,” GlobalTsetSupply.com. [Online]. Available: https://www.globaltestsupply.com/product/hobo-data-loggers-utbi-001-tidbit-v2-water-temperature-data-logger. [Accessed: 26-Oct-2019].

[2] Adafruit Industries, “SHT-10 Mesh-protected Weather-proof Temperature/Humidity Sensor,” adafruit industries blog RSS. [Online]. Available: https://www.adafruit.com/product/1298. [Accessed: 26-Oct-2019].

[3] Arduino, Arduino Uno Rev3. [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3. [Accessed: 26-Oct-2019].

[4] RockSeven, “RockBLOCK Mk2 - Iridium SatComm Module,” SparkFun Electronics. [Online]. Available: https://www.sparkfun.com/products/13745. [Accessed: 26-Oct-2019].

[5] Sleepy Pi, “Sleepy Pi FAQ,” Spell Foundry. [Online]. Available: https://spellfoundry.com/sleepy-pi/sleepy-pi-faq/#Does\_The\_Raspberry\_Pi\_Have\_A\_Sleep\_Mode. [Accessed: 26-Oct-2019].

[6] “Monnit,” Monnit Corp. [Online]. Available: https://www.monnit.com/Products/Sensor/Temperature/MNS-9-W1-TS-ST/. [Accessed: 11-Apr-2020].

[7] Iridium, “Iridium Edge,” Iridium Satellite Communications, 19-Feb-2019. [Online]. Available: https://www.iridium.com/products/iridium-edge/. [Accessed: 26-Oct-2019].

[8] Globalstar, “STX3,” Globalstar. [Online]. Available: https://www.globalstar.com/en-us/products/embedded-solutions/stx3. [Accessed: 26-Oct-2019].

[9] jonoxer, “practicalarduino/SHT1x,” GitHub. [Online]. Available: https://github.com/practicalarduino/SHT1x/blob/master/examples/ReadSHT1xValues/ReadSHT1xValues.pde. [Accessed: 26-Oct-2019].

[10] RockSeven, “What's RockBLOCK?,” What's RockBLOCK? [Online]. Available: https://docs.rockblock.rock7.com/docs. [Accessed: 26-Oct-2019].

[11] “Rocket Scream Mini Ultra Pro V3,” Rocket Scream. [Online]. Available: https://www.rocketscream.com/blog/product/mini-ultra-pro-v3-without-radio/. [Accessed: 11-Apr-2020].

[12] “Arduino SD Card Reader,” Amazon, 2011. [Online]. Available: https://www.amazon.com/gp/product/B07S3F4BLN/ref=ox\_sc\_act\_title\_2?smid=A8GRJU0V6WTPX&psc=1. [Accessed: 11-Apr-2020].

[13] “MicroSD Card,” Amazon, 2011. [Online]. Available: https://www.amazon.com/gp/product/B01DOFCPNW/ref=ox\_sc\_act\_title\_1?smid=ATVPDKIKX0DER&psc=1. [Accessed: 11-Apr-2020].