OUC All-in-One Photovoltaic Sensor

Group 5

Department of Electrical and Computer Engineering

University of Central Florida

Christian Avalos - BSEE
Julian Gentry - BSEE
Ryan Kehrmeyer - BSEE
Zoran Kolega - BSCpE

Advised by Mark Steiner and Rubin York

Sponsored by the Orlando Utilities Commission

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Solar panels are becoming increasingly commonplace in both commercial and residential zones in the United States and around the world. These instances of renewable energy harvesting range from small arrays on a roof to large farms with a quarter million panels. Utility companies like the one in Orlando make up a large part of the total energy harvest through medium- or large-sized arrays. The Orlando Utilities Commission (OUC) has many all around Central Florida, and the farms are quite effective at producing useful power. These solar arrays have specific implementation constraints, and as a result require lots of testing prior to implementation. That testing is done at sites like OUC's research array in Pershing, Orlando.

Normally, a solar panel array can be monitored using voltage and current sensors, and often that data is available to the companies that install and monitor the health of panels. However, the same panel health data is not available to utility commissions. DC optimizers are commonly used to enhance performance operations, but these devices still lack the ability to translate the performance data to utility commissions. Rubin York of OUC explained that while OUC's solar arrays are already monitored, none of the performance data is readily available for analysis. To solve this issue, Rubin has requested the design of an affordable and easy to install sensor to monitor relevant metrics to solar power generation.

The goal of designing and implementing such a sensor is several fold. Foremost, it will allow utilities such as OUC access to important data regarding the efficiency that their solar panels are operating at. This data can support research efforts which aim to discern the most promising solar technologies. In the long term, the result of utilities having greater access to data relating to their solar installations is a more effective solar panel which better meets the requirements of the utilities and the populations which they provide electrical power to.

Another purpose for such a device is to provide indication of panel failure in a solar array. By taking readings for the voltage and current of a solar panel, a defect in a string of panels can be identified and corrected more rapidly. Improving the speed at which issues are fixed is in the interest of the utility commission, the solar panel provider, and the consumer alike.

Upon completion, this project seeks to have improved the way in which solar energy providers generate power by giving utilities the means to test and monitor their arrays more effectively, both on the level of individual panels as well as that of an entire string.

2. Project Narrative

All projects stem from a need or desire from a client or a community. Our project is built upon the world's need to improve solar power performance and our desire to improve the world's renewable energy generation. Here, we focused on the purpose of the project, along with what we expect to attain as an engineering team. Not only does our sponsor require results from us, but we also aim to reach goals ourselves by going the extra mile and providing supplemental features to our resulting device. We explored the requirements established by our sponsor, along with the constraints on our approach. Furthermore, we also noted the relationships between the different parameters imposed on the project, and how helpful or harmful each parameter is on another. While some constraints may encourage others, particular restrictions may manifest difficulty in fulfilling others. Nonetheless, our team ambitiously took on this project, and our expected roles and positions were also explained in this section.

2.1 Statement of Motivation

Every engineering project is started to serve a specific need. The greater purpose of engineering is to improve the lives of the communities that we live in. This project focuses on a seemingly small detail in a much larger picture. When it comes to solar power dependence, it is already an uphill battle. The current fight in the renewable energy industry is one of proving self-worth. With multiple generations already raised and established on the dependence on fossil fuels, advocates of renewable energy must be able to optimize the use of renewable energy. In fact, due to our current infrastructure and its foundation upon the use of fossil fuels, renewable energy must be presented as a spectacular substitute for fossil fuels. In order to do this, the renewable source of energy must be powerful, efficient, and easy to use. Therefore, we must be able to perfect and maintain the electronic devices that help convert and distribute renewable energy.

Solar farms are structured with strings of solar panels which share the same current and produce a shared voltage known as a string voltage. Similar to Christmas lights, this arrangement produces some issues when one or more solar panels in a string become defective in producing the desired voltage output. The overall string voltage drop is not proportional to the amount of solar panels which are defective in a string. For example, if a string of 100 solar panels were to have one panel become defective, the string voltage drop could be much greater than 1% of the overall string voltage (these numbers are simply illustrative, and are not meant to be taken as real measurements). Panels can become defective when they are exposed to external factors. For example, debris such as branches or hail can scratch the panel surface and even crack it; this damage will reduce the panel's effective surface area and lower the power generation. Moreover, other factors such as shade or animal droppings can also contribute to a loss in effective surface area. In addition, water damage can occur over a long period of time. If the panel's seal becomes dilapidated or aged over time, water can seep inside and corrupt the

encased technology. Even electrical components, specifically wiring, can go awry and negatively impact the solar panel's performance. As such, we are motivated to design a device which can affordably and reliably measure voltage, current, and possibly temperature and irradiance to be able to quickly identify defective panels in a string.

The sensors are uniformly distributed throughout a string of solar panels. This layout would allow a solar farm operator to determine the relative location of a faulty panel. As a result, on-site technicians would have a much easier time locating a defective panel, and could instead spend more time resolving the issue with the defective panel. Besides the benefits of easier repair, the sensor devices contribute in some other meaningful ways as well. The array's average performance should increase, providing more consistent power generation due to faster repairs or replacements. In addition, energy producers gain a more refined monitoring process and avoid surrendering fines to distribution partners when they cannot meet the agreed upon energy generation goals. Energy producers would also have access to more technical data regarding their solar panels, which would enable further insight as to the strengths and shortcomings of the manufacturer's product.

In addition to the ability to detect and diagnose faulty solar panels, the sensor device would also be useful for research purposes. OUC has a small scale research solar array, with an approximate output of 12 kW, as opposed to a full-scale solar farm with a power output of greater than 1 MW. This solar array is used to test many different designs of solar panels, including entirely different technologies. Each of the five racks has a different tilt angle, and a combination of several different panels. The panels include several monofacial panels with differing cell counts, as well as bifacial panels, which is what our device will be installed on. Solar cells are considered monofacial when just one side of the panel is metallized and doped, while the other side of the panel is baked in a white insulating film called ethylene vinyl acetate, or EVA. The purpose of the EVA backing is to increase the statistical likelihood of photons being properly absorbed initially and producing electricity, while maintaining a productive temperature in the solar panel. On the other hand, bifacial solar cells are solar cells that have metallized and doped regions on both the front and back side of the cell. Here, more photons are likely to pass through the cell without being absorbed. However, the back-side of the cell is now equipped with the ability to be able to collect the photon energy that is reflected by the ground or other objects. While this may seem redundant or even futile, it is actually a blossoming technology that increases the Wattage per square meter (more energy for the same amount of space!). In fact, the technology is likely to eventually match or overtake the dollars per Watt of traditional monofacial solar panels. This, of course, can be attributed to the bifacial panel's ability to collect what is known as *albedo* from the back side. Albedo is the proportion of incident light (light that is normal to the surface of a solar panel) that is reflected by the panel or its immediately surrounding environment (such as the aluminum rack that the panels are mounted on). Due to the design differences in the monofacial and bifacial panels, they have different configurations in which they are most efficient. With our sensor design, OUC will be able to compare both technologies with far more ease and reliability, and determine strategies

that can heighten the technologies' efficiencies. This is just one example of the type of research that is done on solar research arrays like OUC's.

The important takeaway from this is that while the ultimate goal of this device is to save solar farm operators money via reducing downtime due to faulty solar panels, it is also useful in collecting data for research purposes. The data that is collected from research arrays like OUC's is one of many driving forces for innovation in the industry. It provides solar farm managers with the knowledge they need to select the best technology to meet their goals; influencing their purchases and investments, thus providing an incentive for solar panel manufacturers to make the most competitive product they can. While it may seem rather indirect or disconnected, the end result of this data collection device - if implemented properly - could be a shorter path to a future with more sustainable renewable energy production.

2.2 Goals and Objectives

In order to successfully complete the project, our group established a set of goals and objectives. In fact, an engineering project is essentially a checklist of needs and concerns met. Originally, the team met informally in order to discuss what they envisioned the project could turn out like and achieve. Once we had our initial thoughts, we met with our project sponsor Rubin York, who provided clarification and direction for the project.

Our primary goal was to create several prototypes of a photovoltaic sensor usable in small- to medium-scale solar arrays that measure the voltage and amperage of the panels in order to determine panel health. The engineering and marketing requirement specifications will help to distill this goal into a workable design, but we must keep in mind the importance of successful and precise data measuring and data transmission.

Our secondary goal was to incorporate two supplementary measuring instruments to our original sensor design: a thermocouple capable of tracking internal panel temperature and a pyranometer that can detect incident light irradiance.

These additions separate our design from other iterations already in the market, and allow us to provide an extra layer of functionality and measurement to it in order to meet our sponsor's needs. Together, these devices will transmit data down the string to a local database on the collector node that is hardwired to the producer's network. The producer can then monitor panel health and performance, optimizing the longevity and consistency of the array's generation.

2.3 Function

The function of the project is ultimately the acknowledgement of the customers' needs. Moreover, it is tangible evidence that we not only acknowledged the customers' problems but also obtained a solution to resolve them. In the context of our project, the function of our device will help improve the state of renewable power generation. The sensor will provide a potential to aid utility companies all over the world in optimizing their renewable energy technology. We started with OUC, and designed a prototype for them that they will be able to improve and use in the coming years to best monitor their systems. The future plans involve a potential for mass-marketing, but at the current stage we have created a simple design that can measure panel health at a low cost.

The function of this 'all-in-one' sensor is to detect deviation in a panel's performance. Due to the string configuration of the solar array, sensors need to be implemented at calculated and uniform positions down the line. These positions are configurable, as to be able to uniformly distribute the sensors through a string of panels regardless of the number of available sensors or the number of panels in the string. Note that the sensors measure the sole performance of a single panel; this way we can track the health of a string of panels while minimizing the difficulty of the data collection.

Panel performance metrics are sampled from the sensors in regular intervals, and the sensors relay the information to a collector node which houses a database. From the database, the panel performance metrics can be analyzed and compared against either baseline numbers when the panels were installed, or comparatively against measurements from other panels in the string. Deviations in a panel's or set of panel's performance are traceable to a specific sensor, allowing the defective panel or panels to be located. While the exact location may not be explicitly revealed, technicians can determine the location via the location of the first poorly-performing panel. With this information, the technicians are then aware that the defective panel(s) exist somewhere up the river (up the solar panel string).

In addition to sensing defective panels in a string, the sensor is also useful in testing experimental arrays. With access to quality data regarding the voltage and current output of an individual panel, the process of finding the optimal angle for energy collection using a bifacial panel can be made much more simple.

2.4 Requirements & Specifications

The following is the table of requirements both in an engineering sense and a marketing sense that we determined with the help of our sponsor. These are a crucial step in determining the scope of any project, and they are ranked from highest to lowest priority.

The more important parts of the requirements are the demonstrable ones. They are the requirements we will demonstrate in our final prototype demonstration in Senior Design II. These requirements are not listed in any priority ranking; they are quantities that we ascribe to the engineering requirements in the first table section below, and are required for a successful demonstration anyway.

Engineering Specifications	**Priority**
Must be able to withstand 32 V [DC]	High
Must be able to carry 10 A [DC]	High
Must be rated for up to 10 W power	High
Must last more than a year in use	Moderate
Must be water resistant	Moderate
Must be under $20/sensor	Low
Demonstrable Specifications	**Deliverable**
Voltage measurement accuracy	Within 5% of real value
Current measurement accuracy	Within 5% of real value
Data transmission time	Data accessible in local database within 10 seconds

Table 1: Requirements Specifications

Table 1 lists the requirements specifications for this project. As can be seen in the table, the most important tasks are correct measurement of the voltage and current of the panel, while the most significant engineering specifications are related to the device's ability to survive being subjected to significant amounts of voltage, current, and power. Ability to withstand the environment is also significant, as the device will be used outdoors where solar panels are installed. While having a cost of $20 or less per device was an initial specification, it was quickly deemed impossible with the functional requirements for each sensor. With the costs of parts, PCB printing, external tools, and enclosures considered, this could have potentially been a reasonable goal if we bought everything in bulk, but we only had three prototypes to construct, so the cost requirement was not expected to be met and therefore a low priority. The requirement of lasting at least a year in use was chosen for a specific reason. We expect, with the cycle of students graduating from UCF, that many other projects in the future may involve

solar panels or solar-adjacent technology. Rubin York and his coworkers have determined that the scope of our project is the prototyping phase of what they expect to be a larger, multi-year project. Therefore, some group in the future may take our design and improve it as the topic of their Senior Design project, and since Rubin would like a year or so of successful data collection to be used in that future project, and also to have a prototype that doesn't fall apart right after we graduate and the team is split up, we chose a one-year longevity period. This was achieved on a two-front basis. The first one is protection from external threats. This is mainly accomplished by the NEMA rated enclosure. The second front is protection from internal threats. By this we mostly refer to the overall integrity of the design, and having it be robust enough so that the individual sections of the design do not interfere with one another and cause premature system failure.

For demonstration of the project, the metrics by which the device will be deemed successful are correct measurement of voltage and current as well as successful and timely transmission of this data to the database on the collector node. Additional objectives include the correct sensing of irradiance and temperature using a pyranometer and thermocouple, respectively. However, these are secondary to the voltage and current measurements. These are also well-defined in testing. We have demonstrated that the system behaves in the way we want given a certain voltage input, a certain current input, and the successful and timely transfer of the data to the local database on the collector node.

Table 2 lists marketing specifications for the project. The most important features are related to the ability to connect to a solar panel properly. The device uses MC4 connection as it is the standard used by OUC and many other solar panel operators around the world. Another important specification is NEMA certification of the housing enclosure. This is vital for both the ability of the device to survive the outdoors environment as well as for the safety of people who might end up near the solar array. Noise filtering is also of significance, as accuracy is highly important when designing a sensor.

The marketing specifications are important in that they help constrain the engineering specifications. While the system needs to meet our engineering requirements, it also needs to meet lots of logistical requirements like the ones in the following table. For example, our design takes an input voltage and transmits that data to the database using amplification circuitry and transmission hardware. That gets constrained by the requirement of needing capacitive loads to filter out noise. Then, the system has to fit into a transportable case, so that constrains the bounds in which we have worked to produce something that meets the engineering requirements.

In conjunction with this, the marketing standards include FCC standards and NEMA standards as other constraints. Now instead of just needing a portable case, we needed the case to withstand outside elements to a degree that satisfies NEMA 4X rating. We also needed to make sure that our wireless communication did not interfere with nearby wireless networks or

provide a security risk to anyone working with our system. This narrowed down our choices in wireless communication protocol and structure and helped produce a more robust system. If these requirements weren't included in the design process, then we could have had a system that, while meeting the engineering specifications, would be potentially cumbersome, unreproducible, non-diagnosable, or a potential security risk to OUC or anyone else who may choose to adopt our design for their own uses.

Marketing Specifications	**Priority**
Must use MC4 port connections to integrate into solar panel string	High
Must filter out noise with capacitive loads	Moderate
Must be removable and transportable	Moderate
Must connect completely between solar panels; connecting in series	Moderate
Must transfer data to database	High
Enclosure must be NEMA 4X or 4R certified	High
Must be reproducible	Moderate
Must comply with all relevant FCC standards	High
Must contain port connections for thermocouple and pyranometer instruments	Low

Table 2: Marketing Specifications

2.5 Hardware Block Diagram

The hardware block diagram was the start of brainstorming a design process. This shows not only the division of labor in the group, but also the steps by which the design fits together into a cohesive prototype. The pyranometer and thermocouple are simple inputs, while the VDC and ADC sensors are on-board components of the PCB. Then we have the filtration (and amplification) circuitry and the wireless transmission circuitry. This transmits to our node, and both are housed in separate NEMA 4X rated enclosures. The diagram has been updated to show the progress on each step of the design by each member. Much of the design came together toward the end of Senior Design 1, and required much research.

The following block diagram shows the distribution of labor among our group. As Zoran is the computer engineer of our group, and specializes in software over hardware, his hardware contributions were minimal. For the remaining members, their design and research assignments were decided upon after discussing everyone's hardware strengths and weaknesses. Julian has experience with op amps and control, so his assignment was that of actual measurement sensing and signal amplification.

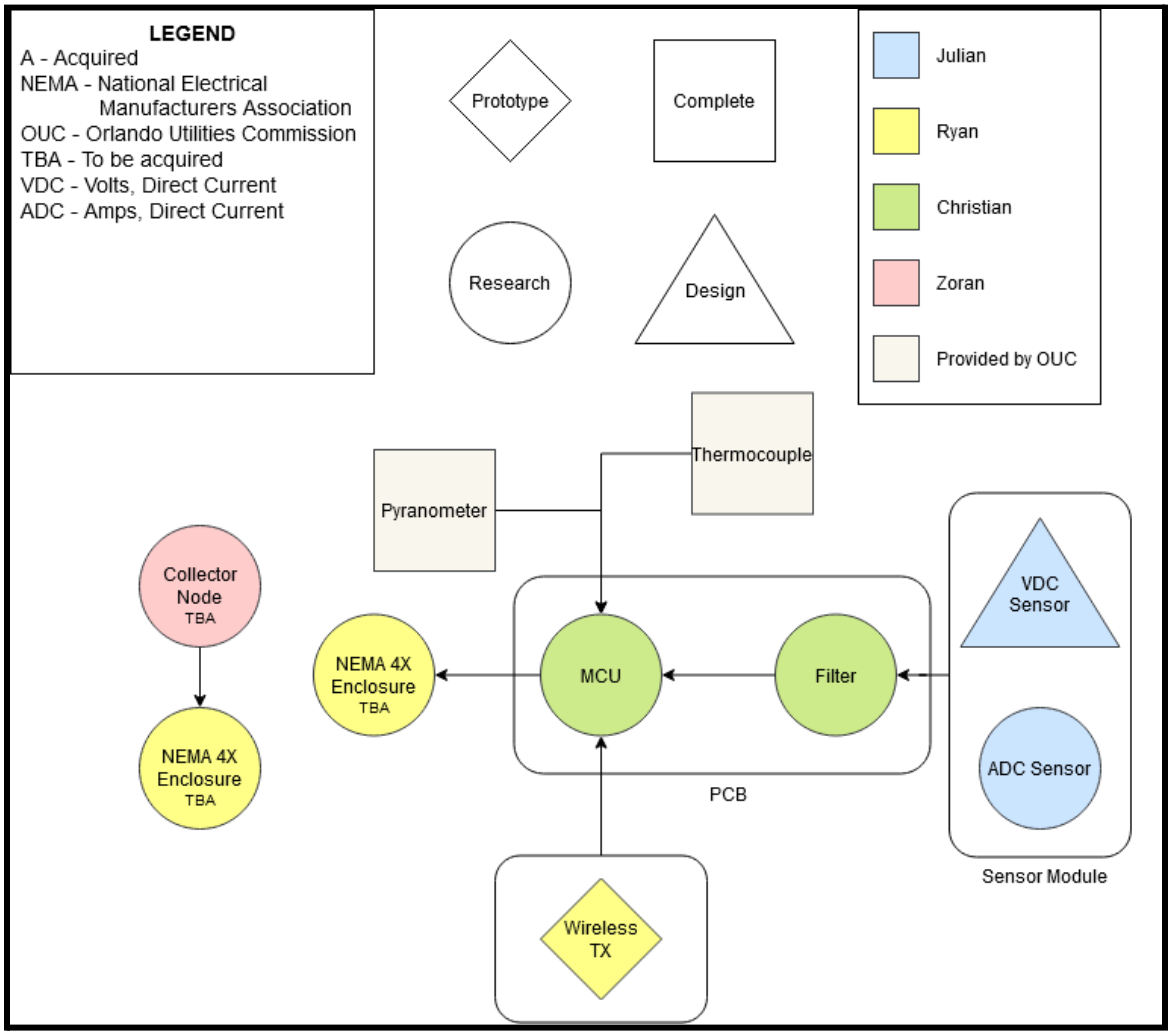

Figure 1: Hardware Block Diagram

Ryan was tasked with researching transmission modules, and determining the trade-off of features and cost, and also researching the enclosures for our design. Chris was tasked with designing portions of the PCB, including the signal filter design. Our application is expected to have a certain amount of signal noise which we need to remove. He was also tasked with structuring the MCU, as the interconnections of the various modules in our design will be complex, and are not likely to be able to be ordered in a single package.

2.6 Software Block Diagram

Figure 2 provides an overview of the software design of the project.

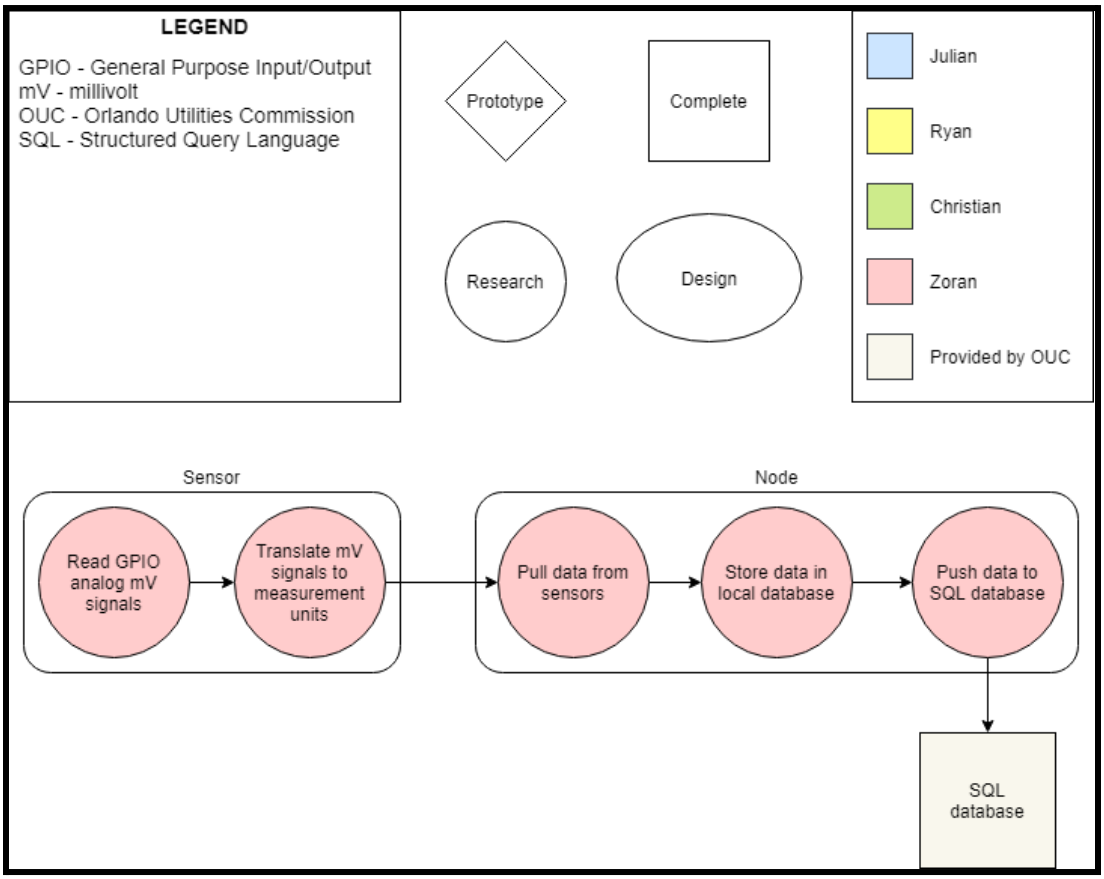

Figure 2: Software Block Diagram

Our design requires several sensors to send their data independently to the node to be collected and analyzed. This, broken down, requires reading signals, turning them into digital information, and transmitting the data. This application of Zoran's software knowledge allows for a more simple design, as much was done to transmit signals via software that would be much harder to implement via hardware.

2.7 Visualized Prototype

We put together a visualization of the final prototype layout with regard to the solar panel array. It is important to make a distinction between the two versions of it, because they illustrate the importance of the design process and how things change over time. In later sections, we show at a technical level why the parallel layout is superior to the series layout of the sensors. By this we mean it is better - for purposes of reducing power consumption - that the sensors be placed in parallel with individual panels rather than in series with the whole string.

Here, we simply want to delineate the difference between them. The first version is the series layout. Clearly, the sensors are in series with a string of panels, and all of them transmit to a central node that transmits data wirelessly to a database.

The second version is of the parallel layout. The string of panels is not broken at all, but the sensors are in parallel with individual panels. We decided to keep both of the visualizations as an example of the scientific method working out design issues and providing solutions.

The sensors shown in Figure 3 are in communication with one another in a WiFi daisy chain. The last sensor (i.e. the one topmost in the diagram) is the only one directly communicating with the node. Each sensor had the ability to contact the node however, allowing for the sensors to be arranged in any orientation. The sensor outermost from the node sent its measurements to the sensor closest to it; that sensor did the same in sending data to the topmost sensor, and finally the topmost sensor pushed its data through it along with the data from the other sensors to the node. The node is a microcontroller unit that has a direct connection to the database via ethernet. This ensures that once all the available data is collected in the node that it is transported in the easiest and most secure way. Once the data reaches the database it can be interpreted for analytical use by OUC.

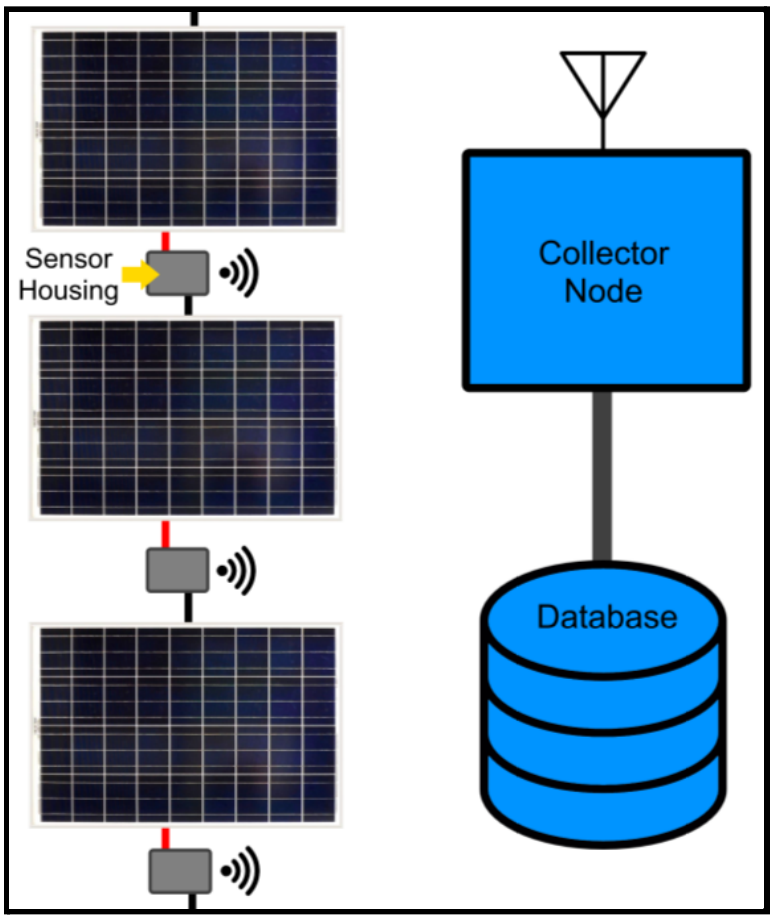

Figure 3: Version 1 of The Visualized Prototype Design

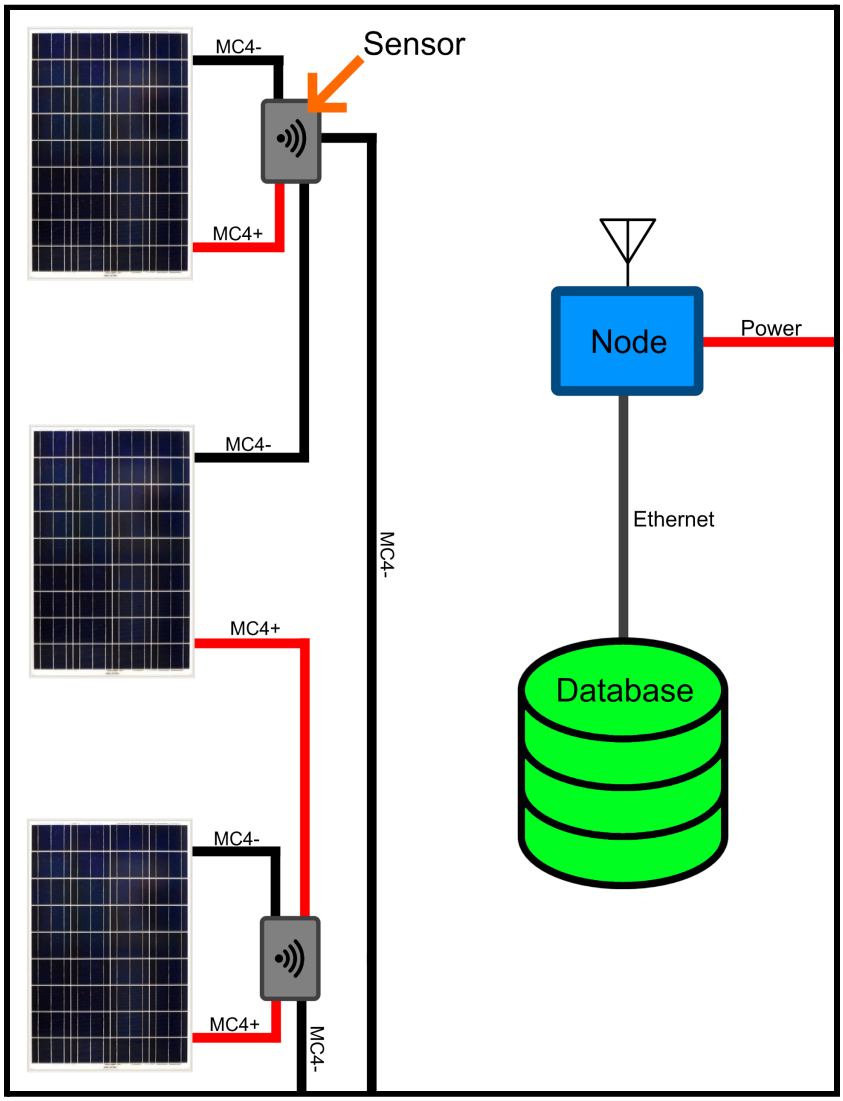

Figure 4: Version 2 of The Visualized Prototype Design

Figure 4 shows the second iteration of our visualized prototype design. It was decided in a meeting with the project sponsor, Rubin York, that gathering data from individual solar panels was a higher priority than measuring the aggregate string voltage and current values. Each sensor now had four connections, instead of the two connections present in the first version of the visualized prototype. Two are the connections to the string, as in the previous visualized prototype. The two new connections connect the sensor in parallel to an individual solar panel, to monitor its current and voltage while avoiding the high string voltage. The sensor and node communication layout remained identical.

The final prototype has some changes from the prototype design shown in Figure 4. Instead of WiFi, the sensors communicate to the node via Bluetooth. They also are not daisy chained, and instead each sensor communicates directly with the node. This simplifies the setup of the network, as each sensor only needs to know the Bluetooth MAC address of the node. The

database is also stored locally on the node with the use of its expandable storage. This still provides OUC with easy access to the data, as it is hardwired to their network via a power over ethernet connection, which also eliminates another cable run. All other aspects of the prototype remain the same.

2.8 House of Quality Matrix

In our house of quality matrix, we confront the practicality of our project. Specifications that are categorized as engineering requirements possess qualities that are much more technical and measurable. On the other hand, specifications that are categorized as marketing requirements serve a slightly different purpose. For example, the accuracy or ease of installation of the device contributes to how marketable and replicable our product is. In the house of quality matrix, we pin point correlations between each requirement. For example, the cost and weight of a product have a strong positive correlation. As the weight of a product increases, the cost of the product typically follows suit.

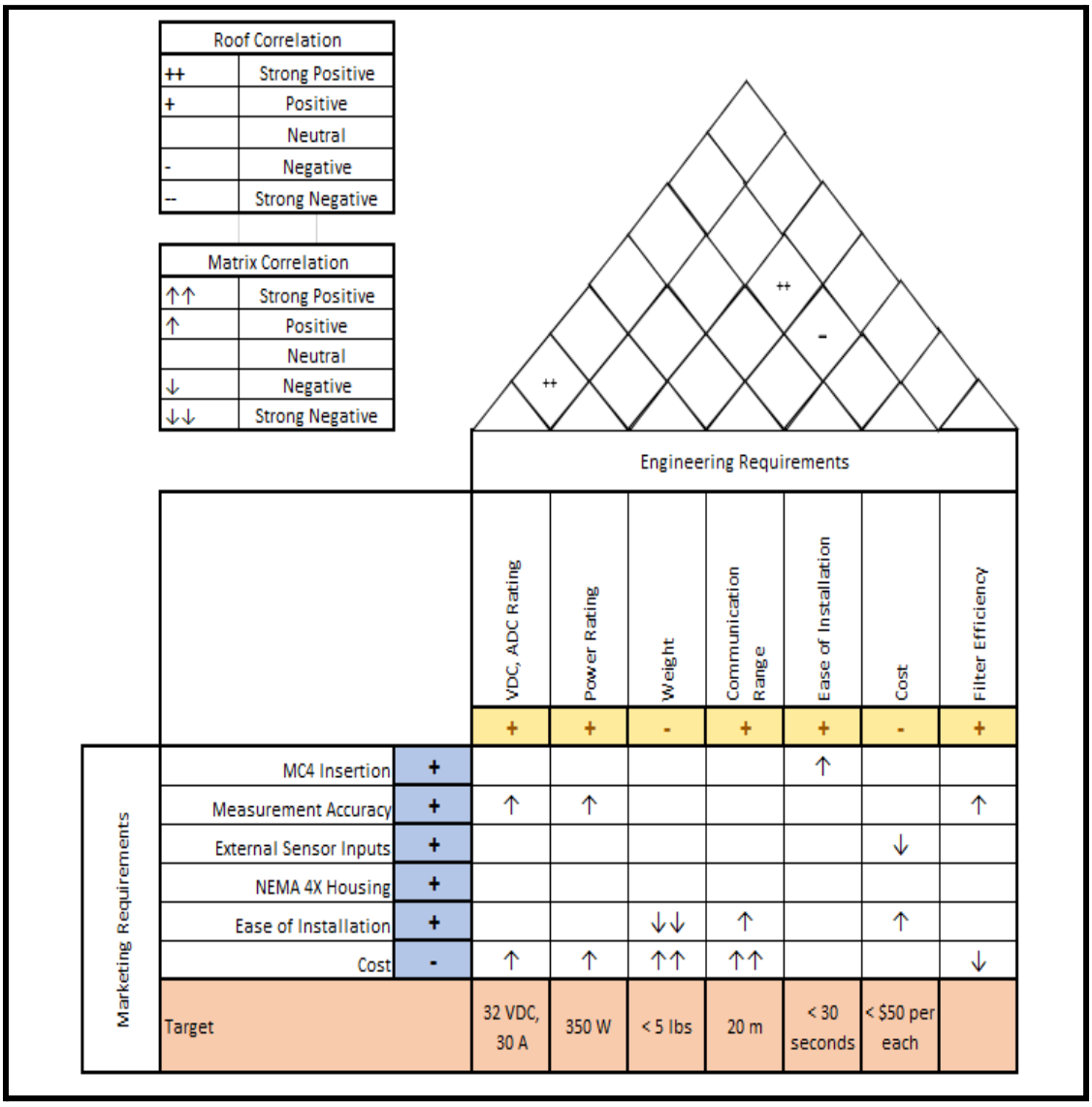

Figure 5: House of Quality Matrix

2.9 Past Efforts

Before we began working on our own project, it was beneficial to take a look at what had already been done in the past. In this section, we acknowledged how OUC used to deal with their problem in the past. Furthermore, we explored the methods they used, the procedures they followed, and the effectiveness of said methods. With this information, our group gathered more insight as to how the issue was tackled before, and how we could improve the past methods or create new ones.

In the past, OUC lacked a very efficient way of monitoring panel string health. In fact, they depended on the final product to determine if there was any issue in the string. Moreover, they would interpret the final production values of the string and dictate whether or not they were short of what they expected the output to be. From there, they would move on to investigate the input strings on the inverter to determine which solar panel string was underperforming. Upon this distinction, technicians from OUC would be sent out to analyze the health of each individual panel in the affected string, until the problem was found. Clearly, this method is tremendously exhaustive in terms of resources and time. Additionally, recognizing the mere existence of an underperforming device via the final output fails to provide any type of buffer for the utility company to resolve the issue before the power is distributed and others are affected.

In light of this shortcoming, Rubin York attempted to improve the procedure by devising a sensor device that could raise an alert the moment a particular panel began to underperform. In an attempt to establish a rudimentary proof of concept, Rubin designed a circuit using a voltage divider, a buck converter, and a Broadcom ACHS-7122-000E sensor. With this, he was able to capture some of the amperage flowing through the system. However, the amperage reading was accompanied by a moderate amount of noise in the output. Also, this method required Rubin to go to a panel and measure the specific panel's performance; this still lacks the ability to communicate the generation in advance. Once he had established that his idea may be feasible, he assembled our project team.

In our project, we designed a sensor that can read the voltage, current, temperature, and irradiance values at certain points in a solar panel string. With our implementation, OUC technicians will not only be notified that an issue exists, but the relative location of the faulty panel. In addition, the sensors' capability of monitoring temperature and irradiance helps provide context as to what is impeding the solar panel's performance. Similarly to Rubin's initial project, we used a variety of circuit components to support a circuit that captures the solar panel's performance. However, we utilized technology that can translate and communicate the data to a database accessible to the utility so that they can be notified of any issues.

3. Constraints and Standards

When designing any engineering product, it is beneficial to take note of standards developed by organizations such as NEMA (National Electronics Manufacturers Association) or IEEE (Institute of Electrical and Electronics Engineers) that serve to increase the transparency around products used in engineering design. By publishing documents regarding the standards that components can be expected to live up to, an engineer can have confidence that their purchase will be able to fit a specific need. Customers can also utilize constraints and standards to help engineers tailor their specific needs while maintaining safe and practical conditions.

3.1 NEMA Enclosure Standards

A NEMA Enclosure is an enclosure for an electrical device that has been reviewed and rated by NEMA to meet certain levels of protection from the environment. There are various rating levels pertaining to the protection offered by the enclosure which range from a Type 1 to a Type 13, with some levels having a few variants denoted by various letters. Most are rated simply to protect people from coming into contact with the electrical components within an enclosure, as well as to provide various degrees of protection for the internal devices as well. Further, some types are specialized for containing a possible explosion. Our device does not have such needs and as such we focused on picking from the more standard variants. As the sensor will be placed outdoors, we considered the possibility of not only dirt and dust trying to get into the enclosure but water as well. While the device is intended to be placed underneath the solar panel which should provide some protection from rainwater, there is no guarantee it will stop the box from getting wet. The most proper level of protection seems to be provided by a Type 4X enclosure, which is rated to provide protection for the internals from dust, water, corrosion, and external formation of ice.

NEMA 250 and IEC 60529

IEC 60529 is a standard from the International Electrotechnical Commission that proposes a system for classifying levels of protection provided by enclosures. They define an enclosure as “a part providing protection of equipment against certain external influences and in any direction protection against direct contact”. This shows the two importance of the two points we brought up previously, that the protections provided by enclosures are bi-directional. IEC 60529 does NOT cover anything other than the requirement of “degree of protection”. There are different degrees of protection laid out in the standard. The one with which we are concerned is IP 66. The first digit in the code refers to the level of solid object protection, with 6 being the highest level. It offers complete protection of the internal components from anything equal to or larger in size than dust particles. The second digit refers to the level of liquid protection. The sixth level of protection provides shelter from power water jets. This would include any type of power water hose, which could be encountered in the outdoors area

on OUC's property. This also means any dripping water, rain, splashing water, or spraying water will have no harmful effects to the sensors inside. These tests were done to a specification of 1 minute of direct water pressure, which far exceeds what we expect to need in terms of liquid protection. This level does not however protect against immersion in water. IP 66 is the IEC 60529 equivalent of the NEMA 250 4X enclosure standard.

NEMA 250 provides the standards of ingress protection tests for all of the enclosure types rated by NEMA, from Type 1 to Type 13. The higher the number, the more robust the protection, generally. For our project, a NEMA Type 4X enclosure is required by OUC, and the seven test requirements as stated in the NEMA 250 standard are as follows:

- Access to hazardous parts
- Ingress of solid foreign objects (falling dirt)
- Ingress of water (dripping and light splashing)
- Ingress of water (rain, snow and sleet)
- Ingress of solid foreign objects (windblown dust, lint, fibers and flyings)
- Ingress of water (hose-down and splashing water)
- Corrosive agents

How this Standard affects our design:

OUC expects the above testing standards to be field tested in several ways. By this we mean that each ingress protection of the 4X enclosure did have an explicit purpose once we used it in the solar array. We know the operating power of the system can be as high as 320 W, so the enclosure will protect anyone working on the array from electrical shock. Since the array is outside and in a grassy field, we expect plenty of dust, fibers, and dirt to be in the air around the enclosure. We also expect plenty of rain and the occasional splash of water. Being exposed to open air would mean that many of our parts would corrode easily, so there is protection from that as well. All of these protections mean that we can go about designing the PCB for our sensors as it would be for any indoor application and not have to worry about specializing the PCB for outdoor use. The protection comes from the enclosure, not the board itself. This goes as well for the collector node.

As a side note, we did have many inputs into this box. We required, whether we use a box with punch-out holes or not, the use of waterproofing seals. Any hole we punch in the box will effectively lower the level of protection.

3.2 MC4 Connection Standards

Solar panels can be connected in strings to produce more power, essentially acting like multiple batteries wired in series. As of today, the commonly used technology for these connections is

MC4, which stands for Multi-Contact (manufacturer name) and the 4 millimeter diameter of the contact pin. We are interested in the design of these connectors because we plan to connect our sensor using MC4 connection as well, placing it in parallel with the DC optimizers already present on the panel array. These connectors are notable for conforming to US Electrical Code requirements by allowing easy joining by hand but requiring a tool to be separated. The connectors are also UL Listed, which means that they have been tested and shown to meet nationally recognized safety standards. Additionally, they are free from any foreseeable risk of fire or electric shock, which means they are good for power applications.

IEEE 802.15

The standard IEEE 802.15 "defined the protocol and compatible interconnection for data communication devices using low-data-rate, low-power, and low-complexity short-range radio frequency (RF) transmissions in a wireless personal area network (WPAN)." [49]

3.3 Pyranometer Standards

A pyranometer is an electronic device that measures the solar irradiance from the space immediately surrounding the panel, and communicates the value via its millivolt or even digital output. The capture of solar irradiance is an important aspect in photovoltaic design. In fact, it is proportional to the amount of power a solar panel generates. Therefore, it is crucial that these devices are maintained and organized amongst manufacturers.

IEC 61724-1:2017

The standard IEC 61724-1 directly addresses all of the measuring and sensing equipment that is involved in photovoltaic system monitoring. This includes the installation, maintenance, and performance expectations of all of the equipment. In the update established in 2017, the International Electrotechnical Commission (IEC) directly addressed the measurement of solar irradiance and the fact that it was not as prioritized as other parameters. In doing so, the IEC created classifications for pyranometers, clearly defining when and where they should be used. In addition, the standard also defined new methods of upkeep for the devices, in an effort to improve their overall benefit.

Each class was defined so that the use of the pyranometer would be optimized depending on the system it would be used in. In Table 6, we summarized the different properties that the IEC designed for the use of pyranometers. This applies to our project because we implemented pyranometers in our design, which is applied to a research-scale array. Using IEC's standard helped us optimize the performance and effectiveness of the pyranometer by helping us choose which classification of the pyranometer to use.

Class	A	B	C
Precision	Greatest ability of precision	Advanced precision	Basic precision
Application	Utility-scale systems	Large commercial-scale	Residential and small-scale commercial or research
Quality-Assurance	Must be calibrated before use at least once a year	Must be calibrated before use at least once every two years	Must be calibrated according to the discretion of the manufacture
Preservation (Heating)	Must have access to heater	Must have access to heater	**N/A**
Preservation (Ventilation)	Must be designed with ventilation	Ventilation design is optional	**N/A**
Cleansing	Must be washed at least once per week	Device can be cleaned according to the discretion of the manufacturer	**N/A**

Table 6: IEC 61724-1:2017's Class Structure

3.4 Thermocouple Standards

A thermocouple is a simple device that is solely composed of two conductor ends made of different metallic alloys. While the first end has two split wires, the other end has two wires welded together. This welded junction generates a millivolt voltage in response to a temperature gradient. Therefore, it is crucial that these devices are maintained and organized amongst manufacturers.

IEC 60584-1:2013

The standard IEC 60584-1:2013 is the standard that provided the thermocouple technology with their internationally known classification system. Categorized as Type

T, J, E, K, N, R, S, or B, each type of thermocouple is designed with a particular set of properties. Each type is primarily focused on the thermocouple's temperature tolerance, which is inspired by reference polynomials created by the IEC. These mathematical functions help scientists determine the EMF capabilities of thermocouples in terms of temperature.

This standard is especially advantageous to our effort because it allows us to effectively track the temperature in our module with accuracy. In other words, IEC's classifications allow us to utilize the most appropriate thermocouple for the job. Note that different arrays in different weather conditions would require different thermocouples. For example, a panel in an array in Orlando, Florida, would reach much higher temperatures than a panel in an array in Manchester, New Hampshire. Therefore, different classes of thermocouples would have to be implemented.

3.5 Python Programming Language Standards

Programming languages are the interface which programmers use to convert their thoughts of how a computer should complete a task into instructions that the computer can understand and execute. As with most problem solving methods, even the simplest programs can be written in a plethora of ways while still accomplishing the same result.

While there is an abundance of reasons to write programs in specific ways to optimize performance, there are also worthwhile reasons to standardize aspects of programming that are not strictly performance related. Differences in programming style become evident to any programmer which has ever read another programmer's code. It is for this reason that programming standards are helpful in giving a uniform appearance to a codebase, such that a team of programmers are able to understand each other's code. This is accomplished by improving the code's readability, as well as its complexity. Adhering to programming standards can also provide a more efficient workflow and reduce development time.

The Python Software Foundation (PSF) is the non-profit organization which holds the intellectual property rights to the Python programming language. In the interest of a more consistent and cooperative development experience, many members of the PSF have created a set of standards for the Python programming language known as Python Enhancement Protocols, or PEPs. These PEPs are design documents which aim to provide information to the Python community, or describe new features of the programming language, among other purposes. The PSF separates the PEPs into three categories: Standards Track PEPs, Informational PEPs, and Process PEPs.

Standards Track PEPs describe new feature implementations for Python. These are primarily for compatibility and interoperability purposes, and are a prerequisite for features to be given standard library support.

Informational PEPs aim to inform the Python community of design issues or provide some general guidelines for implementations of the Python programming language.

For the purposes of our project, the most relevant of the three categories of PEPs are the Process PEPs. Process PEPs describe processes which surround the Python programming language, but are not concerned with the functionality or implementation of the language itself. Process PEPs may also include new proposals, but not directly to Python's codebase, as that would fall under a Standards Track PEP. Most of the PEP Index concerns itself with development of the Python programming language itself. For implementations using the Python programming language, the most relevant PEP is PEP 8.

PEP 8 -- Style Guide for Python Code

PEP 8 is the Python Enhancement Protocol detailing a style guide for Python code. For programmers, a style guide is a set of coding conventions. These conventions describe a plethora of subjects, including string formatting, whitespace usage, comments, and variable naming. The full PEP 8 style guide can be found in the index of PEPs [50], and navigating to PEP 8 in the Meta-PEP category.

While this PEP (and the rest of the PEPs) are only required for code which is to become part of the Python codebase, it is still a positive coding practice to follow a style guide, due to the reasons mentioned earlier. For the purposes of our project, following the PEP 8 style guide is primarily for the purpose of readability. This is important to ensure that the hardware and software are interfacing as intended, and all of the information is being collected from and sent to the correct sources.

4. Technology Research and Part Selection

In our project, we ended up designing a magnitude of circuit designs in an effort to best satisfy our engineering requirements. However, different designs involve different parts, which introduce brand new implications for every alternative design. This section dives into the details of the electronics that make up the hardware components of our project. Every prospective piece of technology that is a candidate to be employed in our designs was scrutinized. We introduced each device and explained its function, as well as its role within our project. This includes the technology's advantages and its shortcomings, both of which had a great impact on our design. While some technology is rudimentary and does not vary much, others have different variations amongst different brands. These differences were also compared and considered during our part selection.

While physically our device is made up of a variety of hardware components, we must also keep in mind the device's invisible inner structure of a layer of software components. Here, we explored many varieties of databases, languages, connectivities, and other software components to determine what best suits our implementation. The ability to store data and transmit it to other devices wirelessly is crucial to our project. Data transmission is an important part of a plethora of technology applications, and as such there are a myriad of ways to approach the transmission of data. Our focus for this section is a power system setting at a utility scale. The technology we use to transmit the data is also key to meeting our project requirements and constraints. The scale, advantages, and drawbacks of all the software components, as well as how they interconnect with the hardware components were dissected.

4.1 Voltage Divider

Voltage dividers are simple circuits that turn a large voltage into a smaller voltage. Given some larger voltage at the positive terminal of one resistor, and given another resistor (or string of resistors) in series with the first, the voltage across the second is equal to the ratio of the second resistance divided by the total resistance and multiplied by the source voltage. In other words:

$$V_{out} = \frac{V_{in} * R_2}{R_2 + R_1}$$

This is a simple way to ensure the safety of the PCB components in nearly any design. A voltage divider can take a large input voltage, and through multiple configurations, divide out voltage for an on-board power supply, a microcontroller, or any other IC. These can be arranged in a variable resistor to allow for precise testing, but since we know what input voltage our solar panels will provide, and also what voltage the components on our board needs, the resistor values can be fixed. This configuration is expressed in Figure 6.

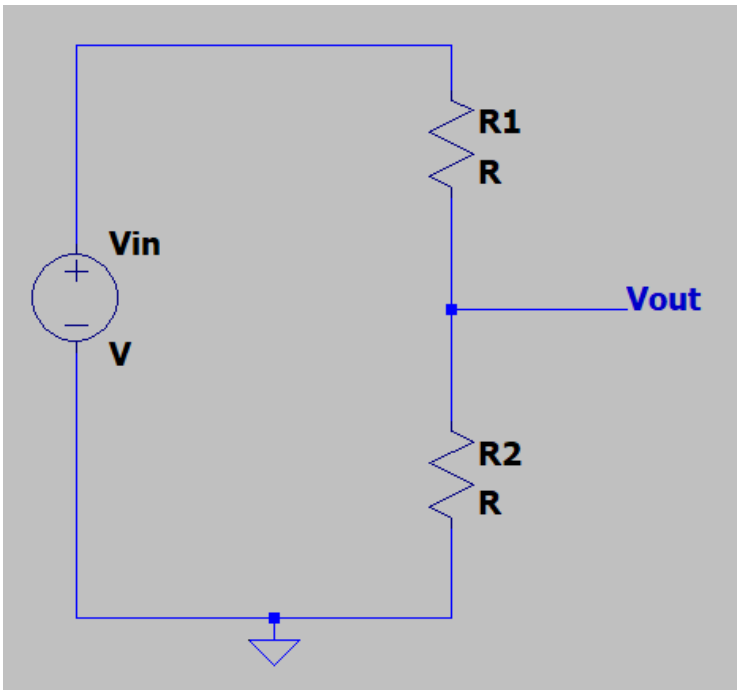

Figure 6: A sample voltage divider.

4.2 Shunt Resistor

Most measurement circuits take voltage inputs. This is especially common with operational amplifiers, as the input current is effectively zero due to the design of the IC. This then raises a problem of measuring current. Generally speaking, it can be very beneficial in certain applications to measure input current. However, this problem of not being able to accurately measure input current can easily be solved with shunt resistors. Given a resistor in series with the circuit branch from which a current measure is needed, the voltage across that resistor corresponds with the current through it, since the resistance is constant. This is useful to us as we need a current measurement from the solar panel string, and since we don't want to draw much power from the solar panel line itself, we are choosing a very small-value resistor.

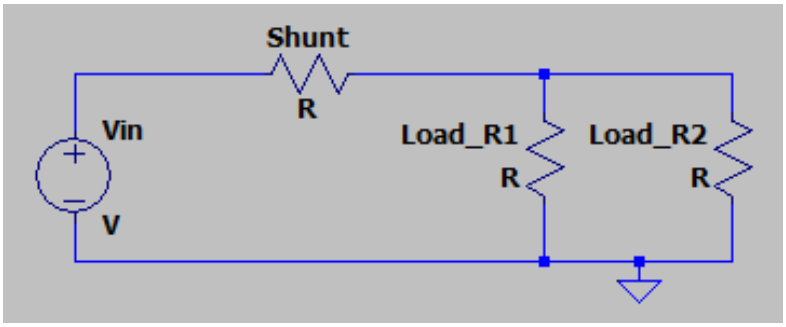

Figure 7: A sample shunt resistor connection.

4.3 Hall Effect

The Hall Effect is the production of a voltage across a conductor that is perpendicular to the current through the conductor and to the magnetic field around the conductor.

In Figure 8, we have a simple voltage source connected to a conductive plate, inducing a current that flows right through the conductor. However, if a magnetic field is introduced at a 90 degree angle from the current, then the carriers will be split in half and the normal flow of current will be disrupted. More specifically, the charge carriers such as holes and electrons will be split in two, separated from their usual pair. Some sensors are designed this way so that they can measure the current flowing through.

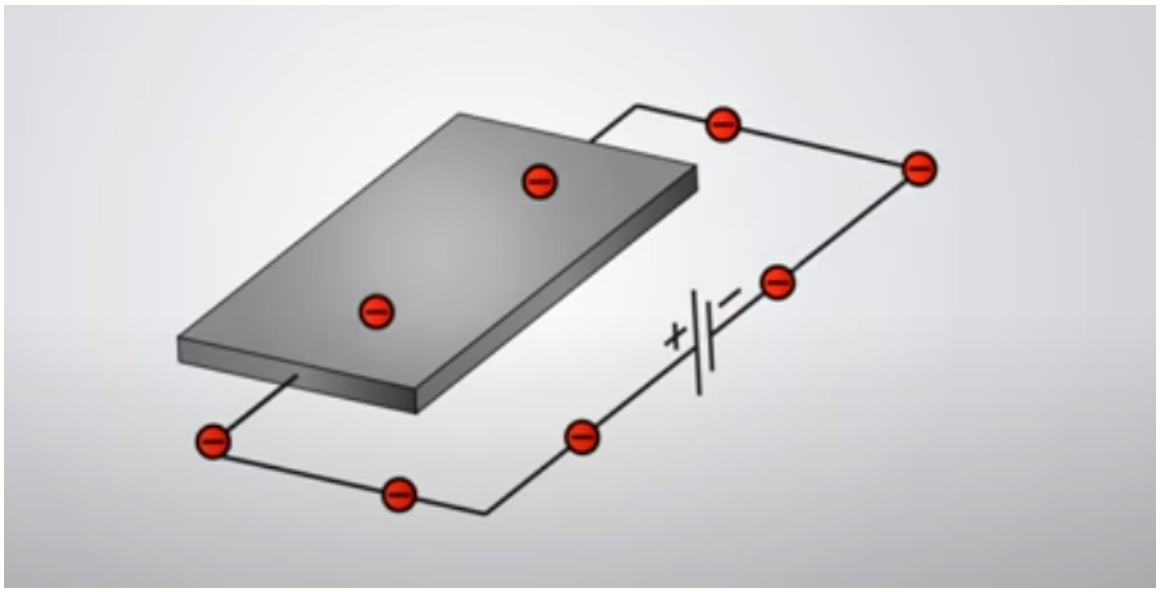

Figure 8: A simple current flow through a conductive plate.

In Figure 9, we can see how the perpendicular magnetic field separates the charge carrier pairs. Note that, similar to a capacitor, the separation of these charges induces a voltage across the conductive plate. Now, with the help of the magnetic field [disruption], we can determine a number of details about this current, such as:

- The sign of the carriers (are they holes or electrons?)
    - Which direction are the carriers flowing?
- The density of the carriers
    - This is crucial for solid-state device characterization.
        - Design of MOSFET, CMOS devices
- The strength of the electric field
    - This can be determined via the voltage induced due to the separation of charges.

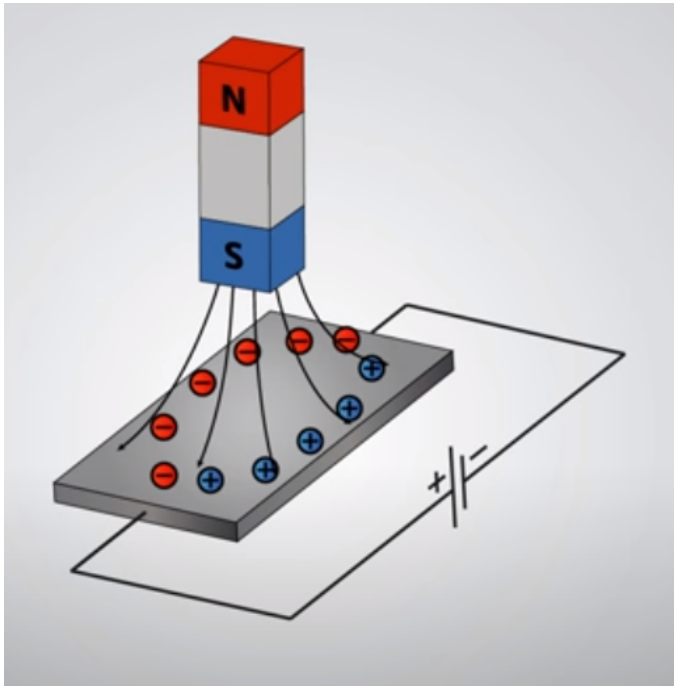

Figure 9: The Hall Effect in action.

The second design example uses a hall effect sensor to sense current. This is a robust and efficient design, but not the type of sensor we wish to use in our prototype. The Hall effect sensors tend to be a single package.

4.4 Amplifier

Amplifier circuits have many applications, but our focus is on using operational amplifiers to boost the voltage measurements to a range that can be read by the ADC. Therefore, we looked at simple unity gain buffers and negative feedback amplifiers. Unity gain buffers (see Figure 10) are very bare-bones circuits, consisting of just an op amp, although sometimes they have capacitors on the input or output. They take an input and, given the efficiency of the operational amplifiers, output the same value with a small range of error. They are useful in that they can allow for the coupling of two different loads without the interference of the Loading Effect.

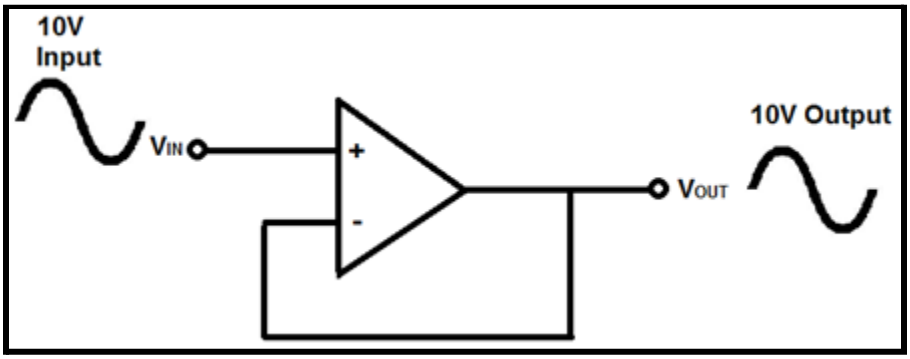

Figure 10: Unity Gain Buffer

For very small inputs we use non-inverting amplifiers (see Figure 11), that through the configuration of the resistors from the negative terminal to the output terminal and to ground provide an amplification or 'gain' equal to the ratio of the resistor values plus one. They usually take in millivolt-order readings and amplify them to volt-order readings.

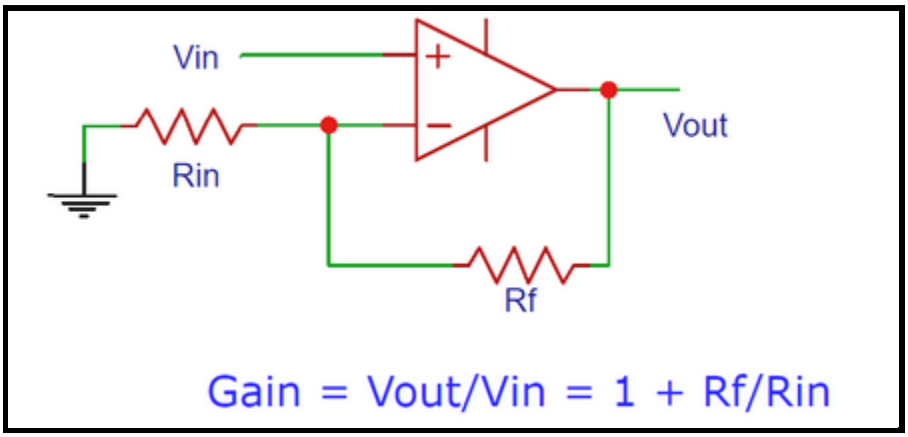

Figure 11: Non-Inverting Amplifier

4.5 Regulator

Our most pressing design considerations in sensor technology, with the exception of voltage and current maximum specifications, were:

- Minimizing percent error in voltage and current readings
- Optimal filtering out of signal noise
- Successful conversion and transmission of measurement data to be later stored and interpreted

There are several options available in the market for off-the-shelf (OTS) voltage conversion modules. Particular OTS conversion modules include linear and switching regulators.

Linear Regulator

The linear regulator (see Figure 12) is prized for its simple design. It is only used for stepping down voltages, and if this is enough for a design then a linear regulator is the preferred choice in most cases. However, a constraint of the linear regulator is that it cannot step-up voltages. The efficiency of a linear regulator is as follows:

$$\eta = \frac{Pout}{Pin} = \frac{Vout*Iout}{Vin*Iin} = \frac{Vout}{Vin}$$

Also, since the current in matches the current out, and since the voltage out is smaller than the voltage in, the efficiency is determined by the step-down ratio, and is therefore quite low. This leads to excess heating and is considered in our design, since we dealt with fluctuating external temperatures.

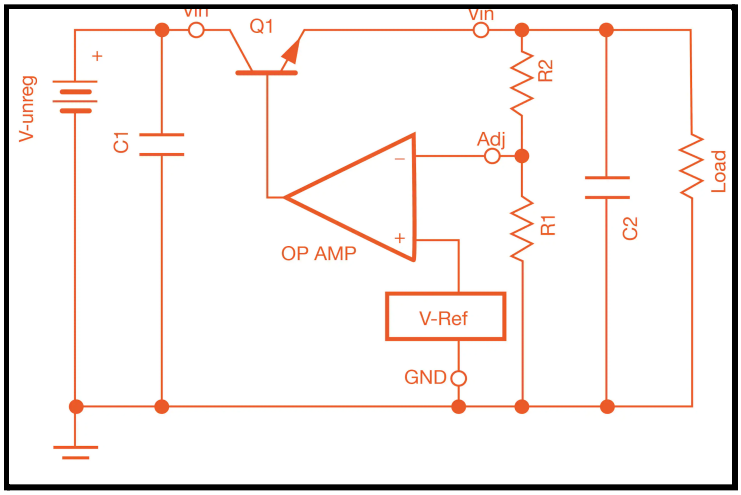

Figure 12: Typical Linear Regulator Design

Switching Regulator

Simply put, switching regulators (see Figure 13) take the DC voltage of one level and change it to a voltage of another level. The way that the voltage level is changed is through storage and release of voltage. This is either done using inductors/transformers or capacitors. This can either step-up or step-down the voltage, and is very energy efficient. Ideally, whether the transistor component inside the switching regulator is on or off, the efficiency is capped at 100%, as ideally the capacitive and inductive components are lossless. In practicality, they are not lossless, but switching regulators can reach efficiencies much higher than linear regulators, usually in the range of 75-98%. The efficiency is this high because unlike the linear regulator,

the current in a switching regulator is near zero, which results in zero or near-zero power dissipation. We would be trading the cost of a heatsink for the cost of more complex circuitry.

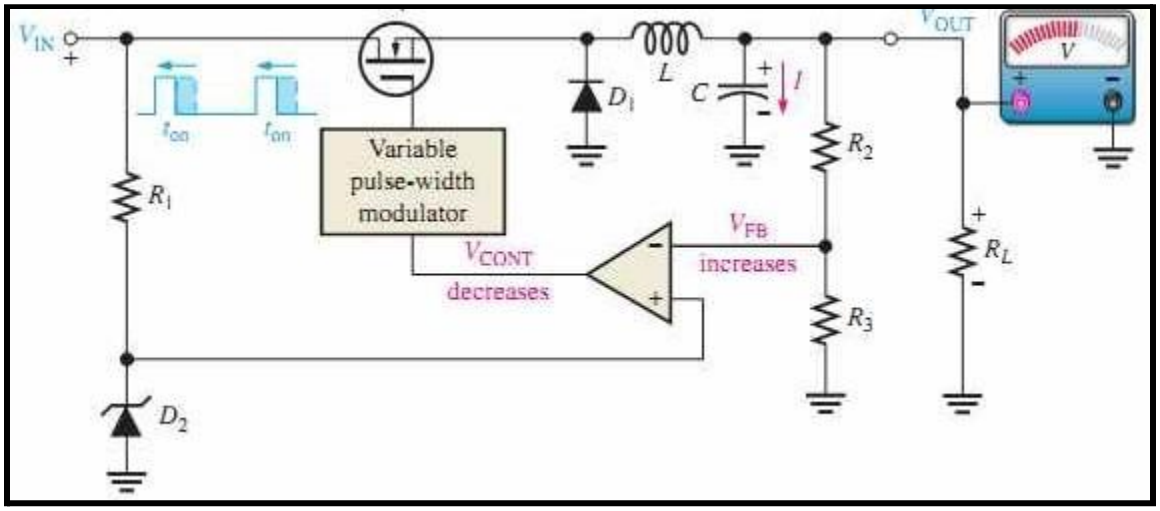

Figure 13: Typical Switching Regulator Design

Type of Regulator	Efficiency	Power Consumption	Complexity	Thermal Stress	Size
Linear	Low	Dissipates excess power	Low	High	Small-moderate. Large at higher power.
Switching	High	Stores excess power	High	Low	Small-moderate. Small at higher power.

Table 7: Regulator Comparison

4.6 Power Supply

For our design, the power supply must have a large input range. The maximum output voltage from the solar panel was about 32 VDC, and fluctuates at different times of the day; this is due to the intermittence of solar energy. Throughout the day, the sun gradually translates across the sky, filling different positions. The panel sits stationary, receiving solar energy at different angles. Therefore, the amount of energy produced by the panel differs throughout the day with the peak typically occurring around noon. Note that external factors can impact the power generated, such as cloud cover or animals. Because of this, we decided to maintain overhead

and establish the input voltage parameter [for the power supply] to be an upper limit of 40 VDC.

Note that the precision of the output voltage of the power supply is not as crucial as the precision of its input voltage: this is because the devices involved in the circuit do not require a very specific voltage range with which to operate. Therefore, an output voltage of 3.3 V was chosen, which is safely above the typical power supply lower-bound of 2.7 V. Needless to say, the output voltage is expected to swing. However, the maximum supply voltage for the typical op-amp hardly drops below 5 V, which preserves room for our power supply to adjust. Moreover, the ADC we selected shares the same upper and lower bound for its supply voltage.

The final aspect needed in a PCB power supply design is the supply current. Typically, manufacturers do not divulge much information in their datasheets about supply current limits for op amps. Here, the same stood true about both op amps and the ADCs. Consequently, we decided to focus on avoiding hazardous heat dissipation amongst our electronics and chose a supply current of 0.1 ADC.

4.7 MC4 Connectors

MC4 connectors are the most common electrical connector for strings of solar panels. They were originally designed such that they could be connected by hand but needed a tool to be disconnected in order to protect the electrical integrity of a string of panels. This is the connector used in OUC's array, and is such because they are rated for up to 1500 V which allows for longer strings of panels. The National Electric Code (NEC) limits user-installed, exposed wires to voltages of 50 V for safety reasons. Therefore, the MC4 connector was created so that the danger of high voltage would be circumvented. The plugs are inside plastic protective shells, and each looks like the opposite gender of the actual connector inside. The male pin is inside a female housing, and vice versa. The challenge proposed by these connectors is in implementing them into our design, which we expect to be a relatively small enclosure. This involved panel-mount ports for MC4 connectors, through which the rest of the PCB is connected via soldering. Our main design concern remains connecting them in a way such that the safety of their design is not compromised, and in a way that allows for easy disconnection using their special tools. These connectors were provided by OUC as they have an abundance of them for use with their various solar arrays.

4.8 Sensor Distribution Configuration

There are two sensor-system configurations that we have considered. The first involves placing the sensors in parallel with the panels, and the other involves placing them in series. We discussed why we picked the first option and why it serves a better fit in our project.

Both of the configurations allow for each sensor to draw power externally and therefore eliminate from the design the need for on-board power sources. In turn, this significantly reduces both the size and cost of our design. In addition, both configurations allow for the inclusion of the pyranometer and thermocouple measurements, which are secondary features. The inclusion of each of these for each PV sensor allows for full-string temperature and irradiance measurements, and the precision increases with the insertion of more sensors either in series or parallel.

Parallel

Placing the sensors in parallel with the solar panels involves connecting the sensor to each terminal of each chosen panel. This could be every panel in a string or every few panels in a string. This would make it so that there is no resistance in the solar panel string itself. However, this may be an irrelevant positive as the voltage that is being transmitted to the node, unless there exists a sensor for every panel, would be an incomplete representation of the voltage of the entire string. Several panels would have their voltages measured and recorded, but the comparison one one panel's voltage to another is almost useless without the perspective of the entire string. This is not necessarily a bad thing; there is a change in required maximum voltage which lowers the overall power requirements of the PCB. The incompleteness of the string voltage measurement can be mitigated by changing the positions of the sensors in the string, and since this is a small scale string that is a feasible option.

Series

Placing the sensors in series allows us the capability of measuring current easily through multiple parts of the string. This creates an opportunity for monitoring of the overall integrity of the string; if one panel were to malfunction, there could be a drop in line current, and that would be monitored through sensor reading comparison. Measurement of voltage, being one of our top concerns, is made easy through series configuration. Instead of measuring the voltage of select panels and comparing them without a reference, the voltage at the insertion point for each sensor can be measured with respect to ground, and therefore with respect to the rest of the string. For example, if each panel in a string should have a voltage drop of 50 V, and there are ten panels, the string voltage drop should be 500 V. If a sensor is placed at the beginning of the string, and another after the fourth panel, we know if the string is 'healthy' given respective readings of approximately 500 V and 300 V, and if either of those are outside a predetermined acceptable range, the health of one or more panels set between the two sensors could be compromised. Extrapolating this to a larger string means inserting more sensors, but with more sensors comes more precision in determining panel health.

In talks with OUC, we determined that, given that the individual panel voltage is the priority in measurement over the full string voltage, the parallel configuration is our choice. This will help reduce the degree of difficulty in data collection, while also maintaining the technicians'

ability to find the relative location of the defective panel. To reiterate, our sensor assists them in determining where in the string does the malperformance start.

Figure 14: OUC Solar Research Array in Pershing, Orlando

4.9 Design Example #1

As with many design projects, data and inspiration can be pulled from existing designs. Ours is no different, as we are working with photovoltaic technology and with sensor technology, and for either of which there is no short supply. Our design does not incorporate PV technology, but we are adjacent to it, and so therefore must consider the constraints and limitations of such technology. For example, in the array with which we worked, on any given panel there is a maximum voltage rating of 32 VDC and a maximum current rating of 30 ADC. This constitutes a large portion of our design considerations, as an incorrect or incomplete design could result in the destruction of components given the maximum power in a panel being 960 Watts. We researched similar designs that sense voltage and current in solar panels and then considered scaling them up for our applications.

One such design was that of the TIDA-00640 from Texas Instruments (see Figure 15). This is a similar PV adjacent technology, as it involves voltage and current sensing of MLPEs, which are Module-Level Power Electronics. Such electronics are used for optimization of solar modules (i.e. panels) and are placed in parallel with each module, thereby creating a string separate from the module string. The voltage and current sensing technology in particular was of great use in the design of this project, as we desired similar features like:

- Input voltage step-down (high range)
- Low power consumption
- Low-cost
- Data transmission

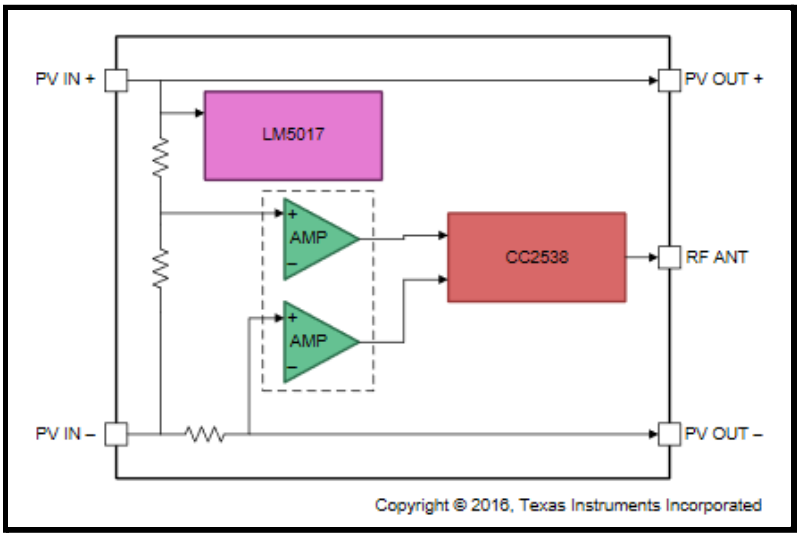

Figure 15: TIDA-00640 Block Diagram

Even though it is a design specific to MLPEs and not to panels themselves, the configuration of the inputs is preferable to our design because we are attaching sensors to solar panels instead of attaching MLPEs to panels. In fact, much of the end result in terms of the circuit structure and in terms of the data collected was the same. We needed to measure panel voltage and current, just like TI does, but we had extra features that distinguish our design from theirs. Despite this, there is always the concern for copyright claims and fair use of materials. We have found that because the TIDA-00640 is used in-house by TI for testing purposes and is not designed for sale on the open market that they do not have any copyrighted material in this design, not trade secrets or trademarks. Therefore, even a carbon copy of this design would not merit legal action. However, as stated above, we circumvent this altogether by including unique features and by using completely different individual components.

Explanation and Theory

What follows in this section is an explanation of the design theory and technology used by Texas Instruments in the TIDA-00640 design that informed and inspired us in the design of our prototype.

Power Input

The power input of the TIDA design was to be fully drawn from the panel array, so the cost of power draw needed to be as small as possible. At the time of writing, TI showed that the nominal voltage rating for most current modules was then 75 V and the power rating was 300 Watts. The values are only expected to increase, and that is in-line with the line of thinking from OUC, as they expect even small-scale arrays to have panels approaching 400 W in the not-too-distant future. They therefore had a wide input voltage range of up to 90 V. Our specifications only call for a max voltage of 32 V. For their needs, they used a step-down regulator like us, from the LM family of regulators. Another important consideration of the design was to have not only a high maximum voltage but also a wide input range, as the nature of PV modules is to have a high voltage swing on the input values. This is consistent with our design, though the change we made was that in considering an input voltage range, we only wish to transmit voltage data from the solar panels when they are at the normal operating range, so we have a minimum voltage requirement of 14 V on our PCB power supply.

Their final consideration, which we decided to adopt, was to choose ceramic capacitors, as the high capacitance numbers would require physically large electrolytic capacitors, and cost should be kept to a minimum.

Current Measurement

From the TIDA datasheet: *"The primary goals of the current measurement feature in the TIDA-00640 are to minimize impact on the solar string and to provide reasonable accuracy."*

Their chosen max current aligns with our chosen max current per OUC's specifications, which is 10 A. They thus chose a small-value shunt resistor for current sensing, of a value 1 mΩ. Due to the large amount of current, a shunt resistor could have problems with overheating, so they placed a limit on it of 0.1 W. Then, the power loss is negligible as shown:

$$\%\ power\ loss\ from\ shunt\ = \frac{shunt\ power\ dissipation}{Maximum\ module\ power}\ x\ 100\ =\ \frac{0.1\ W}{300\ W}\ x\ 100\ =\ 0.03\%$$

Since our max current is the same, we also chose to use a 0.001 Ω resistor. The tolerance of the resistor was lowered in their design to 1% to keep costs low; we did similarly. Now, given our specifications, the voltage output of the shunt resistor is determined by Ohm's Law as follows:

$$V = I \times R = 10\ A \times 0.001\ \Omega = 0.01\ V$$

Then the TIDA design used an amplification circuit to get this mV value to an acceptable input value for the ADC of 3 V. This op amp circuit has a gain of 260, using resistors of values 300k and 1.15k in a negative feedback configuration.

Voltage Measurement

The voltage divider on the TIDA-00640 is of a simple design. A low upper limit was set to give the analog circuitry plenty of space to account for any possible spike in input voltage. The input voltage maximum value for their design was 90 V, and in order to bring that down to a reasonable value of 1.77 V - which is in the range of the ADC - they needed a divider ratio of 50.8. The range and division was clearly different for our design, with a maximum value of 32 V. This value for the TIDA-00640 is fed into a simple unity gain buffer, which shares the same PCB package as the amplifier for the current measurement. In order to minimize power loss, high value resistors were used. A total drop in the divider of 510k Ohms results in only 177 uA of the current from the module being drawn. In our design, the maximum input voltage is 32 V, and the total drop is 105k Ohms, which resulted in 304 uA current draw from the input line, which is sufficiently low.

Wireless Capability

Another difference we had from the TIDA-00640 is in the wireless transmission medium chosen. What follows is a summary of their choices and why we are choosing a different path. With the wireless back end having multiple possibilities in terms of configuration (i.e. the computer they used in testing to receive data), the important consideration for them was the configuration of the wireless front end, as that would be the part of the wireless communication that would be permanently set into a PCB. They chose a PCB trace F-type-antenna designed for 2.4 GHz systems. This is fairly complex, so they did not go into the specifics in the datasheet. The antenna is a 50-Ohm load and so must use a matching filter network. The components of that network need to be carefully chosen and placed. This problem was reduced with the choice of a Murata Electronics integrated balun that replaced 6 of those specific components needed. The remaining three consisted of two capacitors being DNP (do-not-populate) and a resistor acting as a 0-Ohm shunt. This allows for precise testing which was the aim of their design.

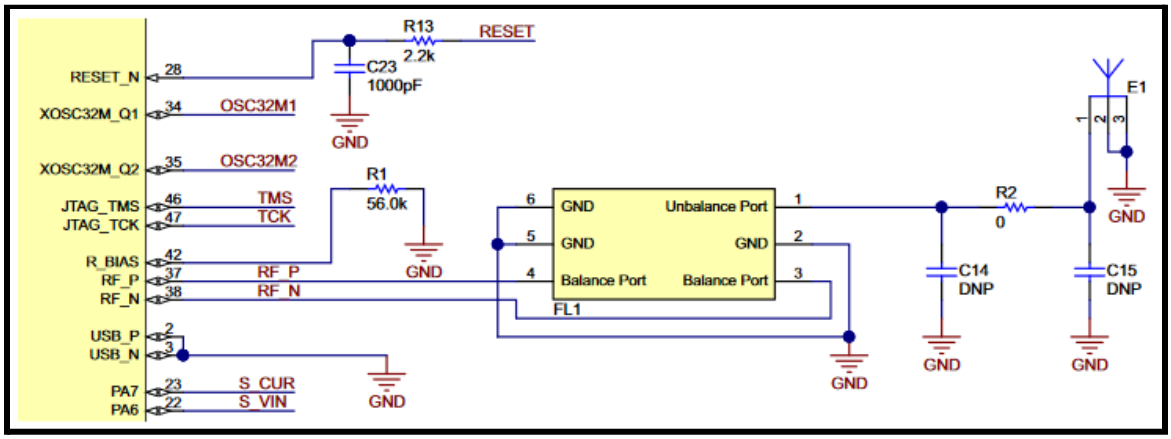

Figure 16 : TIDA-00640 Wireless Front-End

As shown above, this design is fairly complex, and even the design team at TI chose to reduce complexity by using integrated circuit design. We reduced complexity - and therefore the chance of issues arising - even further by outsourcing the entire wireless front end to a Raspberry Pi Zero. More on the specifics of transmission are shown in the latter sections of the Research chapter, but the important distinction between that choice and TI's choice can be made here. Our expertise in circuitry is in the field of amplification and data measuring. Since the expertise of our programmer is sufficient to handle the transmission through code, the hardware design issues that would surely arise from our inexperience in complex antenna design would be nullified through outsourcing. This even trumps the path by which we would take inspiration from this design as we have done with many other aspects of our prototype, because the complexity of the circuit design is quite a bit more than the rest.

Temperature Sensor

The temperature sensing capability of the TIDA-00640 comes from the innate temperature sensing capability of the CC2538, which is the IC they are using for transmission and for its internal ADC. The temperature sensing is therefore at the module level, and better helped Texas Instruments determine the disparity between solar panel expected performance and solar panel practical performance. The different path we are taking requires the thermocouple, which still takes the temperature at the module level, as it was attached to the front of one panel at a time; it simply fed into the enclosure to be transmitted from the Raspberry Pi Zero along with our other data. Therefore, even though the channel by which the temperature is monitored differs entirely in our design with respect to the example design, the end result is the same in helping precisely determine the health of individual solar panels.

4.10 Design Example #2

This example comes from the ACHS-712x Evaluation Board from Broadcom. This is a commercially available board that is used to measure current and return a value, proportional to the measurement, that can be transmitted and analyzed. In the section below we discussed the similarities and differences between this design and what we expect our design to be, and what takeaways we have from this design.

It should be noted that this is a much more simple design than ours. It is a simple current measuring device that can take up to 50 ADC as input and output a value that corresponds to the input value. There is no on-board power supply, no voltage measurement, and no transmission module. Therefore, we can simply compare it to our proposed current measurement and amplification circuit.

The ACHS-712x is a Hall effect-based current sensor. It can be used for either AC or DC current sensing. Clearly, there are two major differences here between this design and ours already. Our design has no need for AC current sensing, as the current in solar panels is always DC, and our sensing modules are not fully integrated. We have resistors adjacent to amplification modules that do the same function as this fully-integrated design.

Per the datasheet, pins 1 and 2 on the IC are linked, and pins 3 and 4 are linked, as they are the input and output, respectively. This is apparently to accommodate the large amount of current. There are several other characteristics they require for current accommodation, like large input terminals (the large plates in the Figure 17 below) and 4 oz copper.

We chose differently for specific reasons. While we do have large input terminals (like the terminal block we chose), we do not need 4 oz copper. Our board does not need to be as small as possible for cost-effectiveness, so we chose to use the standard - and the less expensive - 1 oz copper and offset the reduced heat dissipation by making the input signal planes very large in comparison to the ones on this board.

The board, as mentioned earlier, does not have an on-board power supply. They have an external 5V supply as they are simply using the design as an evaluation board. We had a power supply on-board that output 3.3V. The ACHS-712x does take input power like any of our ICs and does not have any power generation system of its own.

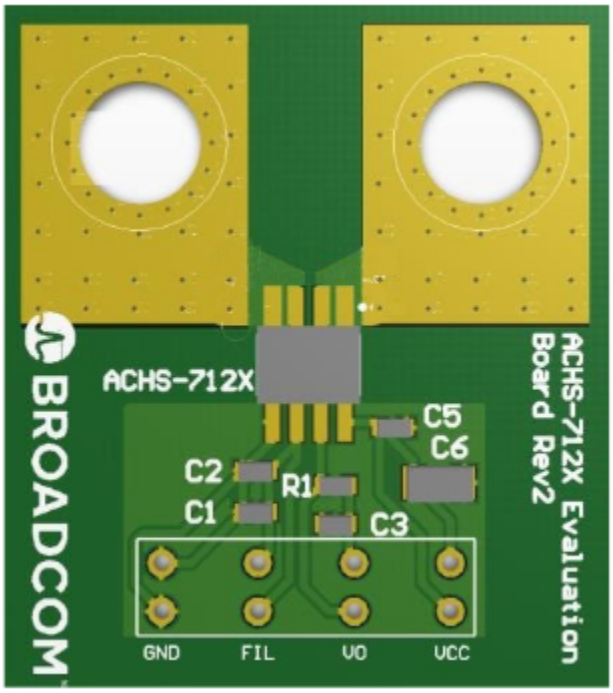

Figure 17 : Evalboard layout for ACHS-712x

The other features of this board include:

- Temp. Operation Range: –40ºC to 110ºC
- Sensing current range: ±10A, ±20A, and ±30A
- Output sensitivity: 66 to 185 mV/A
- Typical total output error of ±1.5%
- Small footprint, low-profile SO-8 package

Let us go through each of these and compare them to our proposed design specifications.

First, we have the operating temperature range. The evaluation board has specifications much in excess of what we expect in Orlando. We never experienced anything below 0°F even in the coldest times of the year, and we won't expect much over 100°F to be sustained in the hottest summers, and our temperature range, being much diminished in comparison, reflects that.

Second, the max ACHS-712x current sensing range is ±30A. We never experienced much over 10A in the solar panel array, as the output current is proportional to the energy from the sun, and at peak times we expect around 10A as a maximum value. Therefore, we did not need that level of current sensing range.

Third, they have designed the system to have a sensitivity range of 66 to 185 mV/A. The sensitivity of our design was proven in the testing phase of Senior Design 2, but we experienced

a similar level of granularity due to the robustness of the MCP-3008 ADC we chose, and that would be the case for either input voltage or input current.

Fourth, the output error of the ACHS-712x is typically ±1.5%. As per our demonstrable design specifications, confirmed through discussion with Rubin York, our error was typically ±5% as the importance of our design is to detect panel health and successful operation, not determine with precision the exact operating voltages or currents of the solar panels.

Lastly, the footprint of the IC is a small SO-8 package. They prioritized SMD components in their design, and cost-effectiveness of a small board footprint as well. We, on the other hand, have a lesser regard for the size of our ICs, and whether we have SMD or THT components is not a major concern. This is because we would prefer a lot of room to be left for heat dissipation, maintenance, or design tweaking in the future.

4.11 DC Optimizers

Our project objective has a very similar counterpart in the solar power industry. DC optimizers are a part of the wave of new, cutting-edge technology that has contributed significant gains made in the solar world. Although similar, our sensor technology provided services that DC optimizers do not. Whether or not both devices are needed in solar arrays was also discussed further.

Similar to a string of old-fashioned Christmas lights, the production of a solar panel string is limited by the performance of its worst performing member. With solar panels, this poor performance can be attributed to many factors, such as imperfect shading conditions or contrasting orientation compared to the other panels. In turn, these variations within the string bring about mismatch losses, which can dissipate the effective/maximized power generated by the healthy panels. Moreover, because solar panels are commonly connected in series, the current flowing through the string of solar panels cannot exceed that of the worst-performing panel. Because of the high power dissipation in the degraded panel, irreversible damage can be brought about amongst the entire string of solar panels.

DC optimizers were created to maintain a healthy performance along a string of solar panels. Optimizers account for the presence of differing physical conditions about the panels by utilizing maximum power point tracking technology (MPPT). This technology monitors the maximum power of each panel and conditions it so that the inverter at the end of the string of solar panels can handle the power more easily. The MPPT technology does this by maintaining a fixed string voltage, which makes life easier for the inverter at the end of the stream. The inverter can, in turn, process the voltage and power much more smoothly. This enhanced operation can increase solar string power outputs by almost 25%.

DC optimizers are a class of module-level power electronics (MLPE), meaning that they are designed to operate amongst various solar modules (composed of a square of solar cells). These electronics are designed to collect and feed solar module performance data, such as output voltage or peak efficiency, from each photovoltaic module to the power company's database. Similar to our intentions, the power company can then interpret these readings and determine if there exists a damaged [or just underperforming] solar panel in the string.

However, there are significant reasons as to why our solar panel sensor needs to exist. For starters, DC optimizers are designed to be installed on the back of every solar panel in a string, as shown in Figure 18. This hefty appendage project increases the [already high] cost of any solar installation. In our project, our sensors were installed every few panels down the string. Our configuration allowed us to *relatively* locate underperforming panels while still tracking panel performance and saving money.

In addition, inverters have been around since the early 1900's, and solar panels have been on the rise since the 1950's. On the other hand, DC optimizers have only become relevant in the last couple decades. Because of their novelty, DC optimizers are not widely compatible with various inverter technologies. If a utility commission wanted to install and depend on DC optimizers for their tracking services, the utility commission would have to ensure that their inverter technology is compatible with the DC optimizer technology. In the likely event that they are not compatible, this would incur another considerable cost in replacing the inverter.

Although the shortcomings of the DC optimizer provide reason for the existence of our solar panel sensors, the DC optimizer does provide unique services that are worth keeping around. For example, our solar panel sensors solely monitor solar panel performance and provide insight to the utility commission about the string's health. Integrating a complementary number of DC optimizers invokes the string to maintain a much higher standard of health and performance. This improvement in solar power performance is the primary objective of any engineer or scientist or researcher in the renewable energy field. Therefore, OUC incorporated both our sensor design and the DC optimizer design in their research solar arrays.

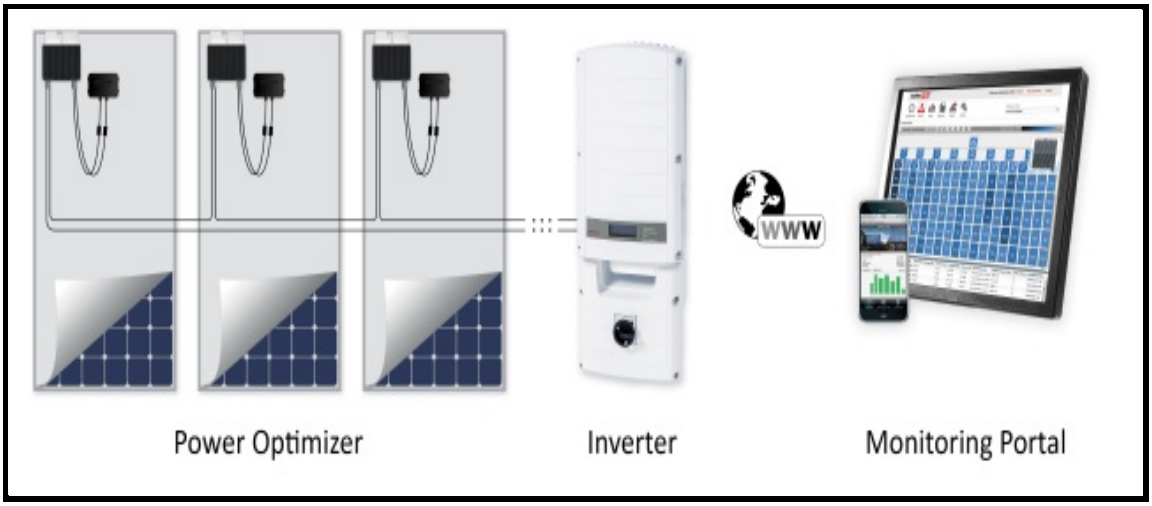

Figure 18: Typical DC Optimizer Configuration

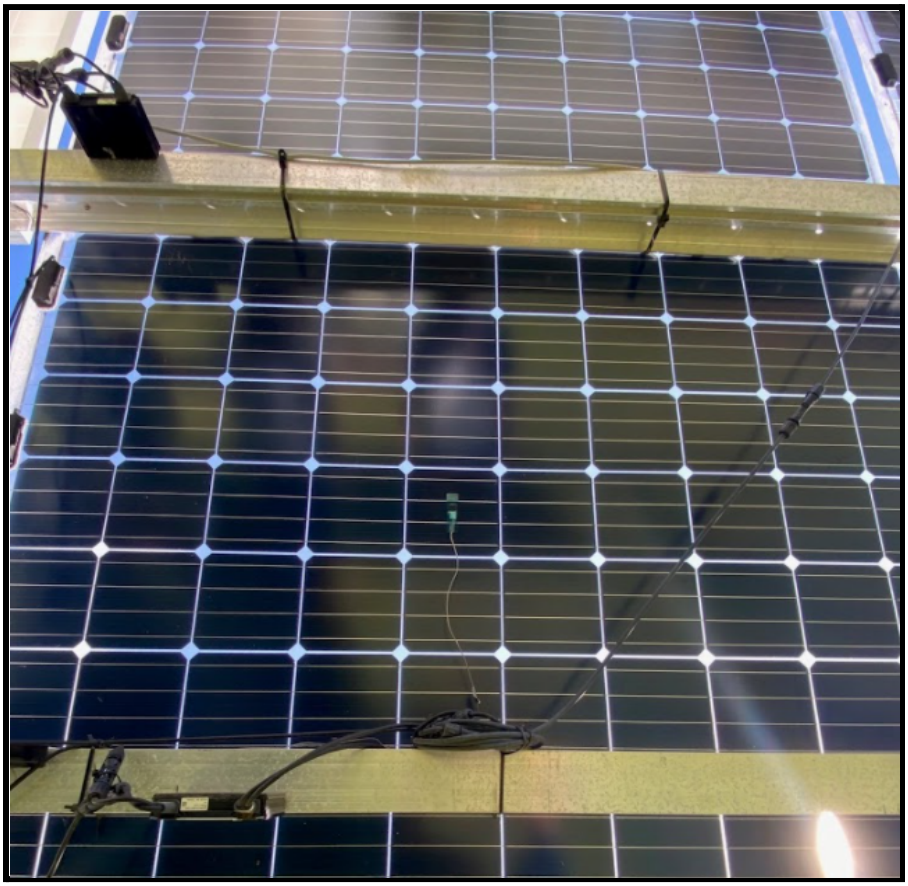

Figure 19: OUC's DC Optimizer Configuration

In Figure 19, you can see OUC's arrangement of DC optimizers in their research array. For every solar panel in the string there is a large metal bracket supporting the structure; this is where the DC optimizers are installed. Our sensors, on the other hand, would only be installed on every few metal brackets. The focus of OUC in using our sensors is to primarily measure the data from the bifacial panels. There are three panels placed close together that use bifacial technology, and as OUC has many types of panels on their test array for research purposes, they did use our sensors on the other types of panels - they measured only one type of panel at a time, as it is important to find out which panels have the best overall health while being used every day for years on end.

4.12 Converter

Designing the power source in our circuit board is crucial to our project. Because we have a variety of active elements, power is crucial to our design. However, simply incorporating a DC-DC converter is not enough to optimize our technology. To do so, we paid an incredible amount of attention to how we design the converter. This is so that we maintain a level of efficiency and high performance without introducing elements that may be detrimental to our overall design.

Instead of designing each component separately, we decided to utilize Texas Instruments' (TI) WEBENCH tool. TI's WEBENCH tool offers a variety of recommendations for power supply schematics based upon our detailed parameters. The WEBENCH tool offers its suggestions through the scope of four primary aspects: efficiency, BOM count, BOM cost, and ecological footprint. Note that in our design, the BOM count is negligible, given that it is not outrageously large. To minimize our footprint from the jump, we decided to specify that we would implement ceramic capacitors, as opposed to electrolytic capacitors.

The first option we reviewed was a schematic based upon an LMR36506RF3 synchronous buck converter. This schematic rendered an efficiency of 64% at maximum voltage. However, operating at a small voltage of 23 V would produce an efficiency of 70%. Despite its seeming lack of efficiency, this design would not dissipate much power because it is a switching regulator. The design also offered a low BOM cost of $1.29 and a relatively low footprint of 112 mm^2.

The second design we reviewed was a schematic based upon an LMR50410X buck converter. This design boasted a much higher efficiency of 81.6% for a fraction of the cost, which was $0.70 per BOM. Even so, the footprint was higher than option 1, at 142 mm^2.

The third design we reviewed was a schematic based upon an LMZM23600V3 buck converter. Similar to option 2, this design had an efficiency of about 79.5%. In addition, its footprint was 41 mm^2, significantly smaller than the footprints produced by options 1 and 2. This design's

efficient build did not come without a cost, as it was the most expensive option at $2.32 per BOM.

The fourth design we reviewed was a schematic based upon a TPSM265R1V3 buck converter. This design had an efficiency of 71.7%, a BOM cost of $1.98, and a footprint of 40 mm^2. Now, this design possesses many similarities with the aforementioned options. However, the fact that many of the IC's pins are left unused generally reduces the design's overall effectiveness.

The fifth and final design we reviewed was a schematic based upon an LM317 linear regulator. A linear regulator will consume a lot more voltage and power than a buck converter, as well as generate much more ambient heat [in an effort to dissipate extra current]. This is reflected in its design properties, as it possesses the lowest efficiency and highest footprint of all the designs at 9.8% and 357 mm^2, respectively. This design does cost the least at $0.40 per BOM.

In Table 8, you can see all of the prospective designs' properties listed, along with their respective rank. We determined the rank based upon the schematics' most tangible and practical properties, such as efficiency and BOM cost.

Option	**Part #**	**Efficiency (%)**	**BOM Cost ($)**	**Footprint (mm^2)**	**Rank**
1	LMR36506RF3	64	1.29	112	4
2	LMR50410X	81.6	0.70	142	1
3	LMZM23600V3	79.5	2.32	41	3
4	TPSM265R1V3	71.7	1.98	40	2
5	LM317	9.8	0.4	357	5

Table 8: Potential Power Supply Designs

In Figure 20, the WEBENCH design is displayed. This design provides the most balanced approach to designing the installation of the LMR50410X in a circuit. This design, of course, prioritizes our requirements in its design.

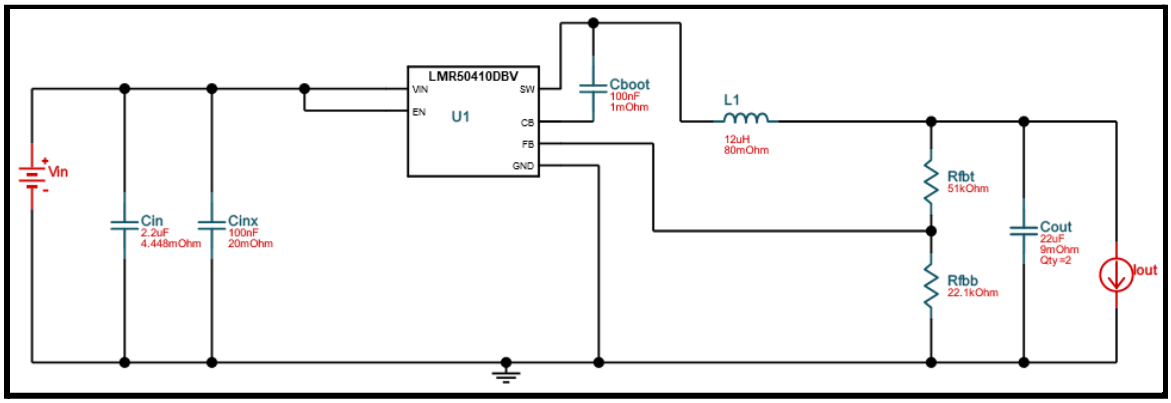

Figure 20: WEBENCH Generation Based On LMR50410X

The efficiency of the converter is illustrated in Figure 21, seen below. We experienced a maximum voltage of 32 V, and we still have an efficiency of conversion of 87 % at maximum output current. The good news for us is that, given the max output current, the efficiency of the converter begins to converge for each value of input voltage. Even though the efficiency increases with the lower the input voltage, our concern is at the high end of the range because that would be the range of meaningful array operation.

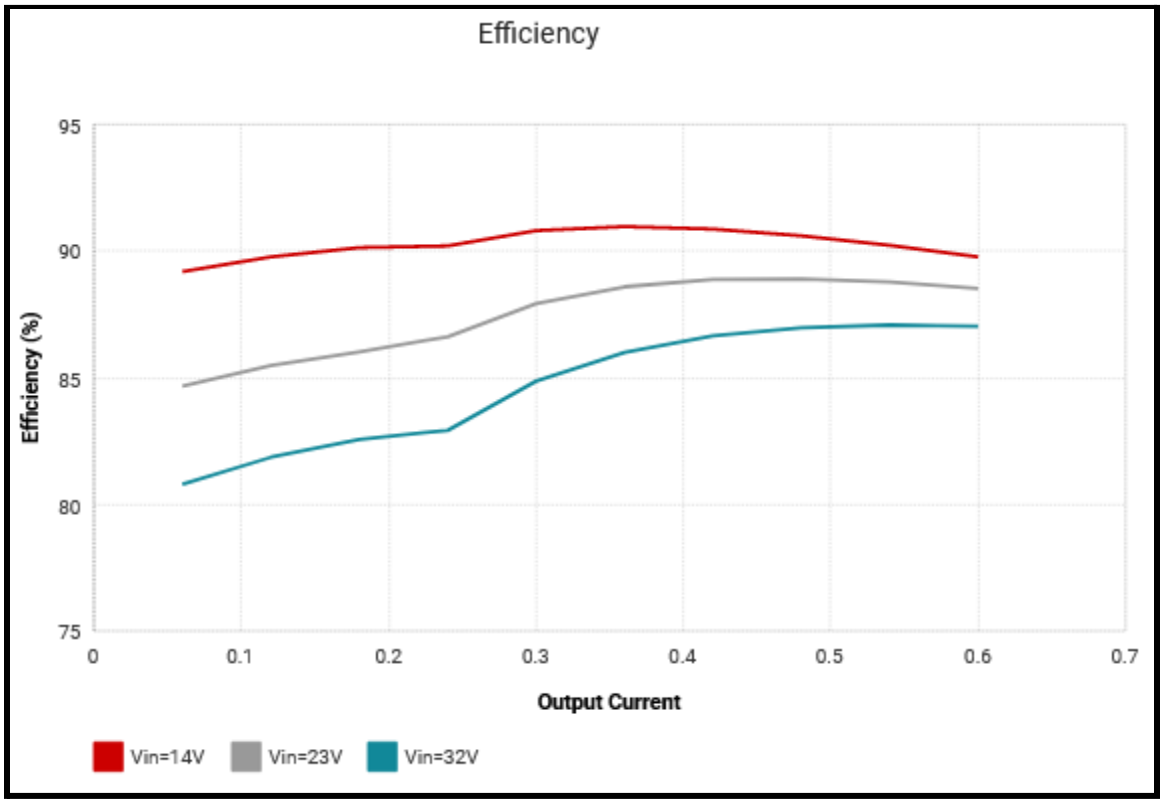

Figure 21: Converter Efficiency Curves

4.13 Operational Amplifier

When deciding on the specific op amps to use for this design, there were several considerations. While one amplification circuit requires resistor ratios, the other is simply a unity gain buffer, so it doesn't require the use of external resistance. This constrains our search criteria, as we don't need op amp packages with built-in resistors if they can't produce unity gain. Also, we only needed two op amps, so a dual package was our focus, though if the characteristics of a quad package fit our requirements better, we would have preferred those packages.

TL072

This package is a dual-channel low-noise general purpose op amp. It has been a widely used package for a while, and has specifications that are useful for many applications. Its Gain Bandwidth is 3 MHz, its max supply voltage is 30 V, its output current is 10 mA, and it has a normal operating range of 0-70 degrees Celsius. All of these values are sufficient in terms of meeting our design constraints, and this was the default choice as an IC for the amplification circuit unless there is a wholly better option discovered in our research.

NE5532

This package is a dual-channel low-noise audio op-amp. Its Gain Bandwidth is 10 MHz, as is needed for audio applications. However, despite its specified use it can still be applicable for us. Its max supply voltage is 30 V, its output current is 38 mA, and the normal operating temperature is also 0-70 degrees Celsius.

RC4558

This package is a dual-channel general-purpose op amp. Its GBW is 3 MHz, its max supply is 30 V, its output current is 10 mA, and its normal operating temperature is 0-70 degrees Celsius as well.

LM358B

This is an industry-standard dual-channel op amp with slightly more robust manufacturing than previously mentioned amplifiers. Its GBW is only 1.2 MHz, but its supply voltage has a larger range with a max of 36 V, the output current is similar to the audio amplifier at 30 mA, and the manufacturing specifies an operating range of -40-85 degrees Celsius.

LM741

This op amp is a single package amplifier that has been in production for over 30 years. It has the lowest GBW at 1 MHz, but the highest max supply voltage at 44 V. Its output current is 25 mA, and the normal operating temperature range is the standard 0-70 degrees Celsius.

Table 9 below summarizes our op amp research. After consideration, we decided the best package overall to be the TL072. All of the op amps meet our specifications, but each had demerits. The NE5532 and LM741 were much more expensive, and we are aiming to minimize cost, so they are not usable. The LM358B is the most robust, yet we simply do not need the upgraded specifications for our application. Lastly, the RC455B is very similar to the TL072, but the latter is a more widely used op amp and therefore is in greater supply and has more actual designs to use as reference.

Op Amp	Channels	GBW (MHz)	Max Supply	Output Current	Temperature Range (°C)	Cost (each)
TL072	2	3	30 V	10 mA	0-70	$ 0.075
NE5532	2	10	30 V	38 mA	0-70	$ 0.296
RC4558	2	3	30 V	10 mA	0-70	$ 0.055
LM358B	2	1.2	36 V	30 mA	-40-85	$ 0.051
LM741	1	1	44 V	25 mA	0-70	$ 0.236

Table 9: Operational Amplifier Comparison

4.14 Pyranometer

One of OUC's product requests was for our sensing system to include a pyranometer. A pyranometer is a device that measures the radiative energy flux from the ambient space, and communicates the value via its millivolt or even digital output. The capture of solar irradiance is an important aspect in photovoltaic design. In fact, it is proportional to the amount of power a solar panel generates.

Pyranometers are typically designed around the use of a thermopile. A thermopile harvests surrounding heat radiation and converts it to an electrical [power] signal. Some pyranometers have op-amps built into the internal design to amplify the millivolt output signal. Moreover, some pyranometer models even internally convert the output directly to digital output.

Other pyranometers are constructed with a dependence on a photodiode. This photodiode generates a small electric current in the presence of solar irradiance. It is typically less expensive and faster in response time. However, photodiode-based pyranometers are limited by their tendency to neglect certain wavelengths of visible light, particularly in cloudy conditions. This wouldn't be very significant in our design, however; OUC is more concerned about making sure that the panels are functioning properly when they need to, i.e in sunny conditions. Photodiode-based pyranometers are not the only technology constrained by their design: thermopile diodes depend on converting thermal energy to electrical energy. This focuses heavily on the sensitivity of the device material and on the conversion [and preservation] of ambient radiation, all of which are prone to error.

When shopping for pyranometers, there is a particular standard to keep in mind: International Electrotechnical Commission (IEC)'s 61724-1:2017. This standard addresses the lack of structure regarding irradiance monitoring technology, and establishes different classes for the devices. Each device falls into either Class A, Class B, or Class C. In Table 10, you can see the summarized characteristics for each device class. In our project, OUC only requires the implementation of a Class C pyranometer. This is because the device is not a crucial component in their plans and is only being used on their smaller research array.

After discussing with Rubin and digging around online, we narrowed it down to four potential pyranometers. Of these four options, two were photodiode-based while the other two were thermopile-based; each of which were categorized as Class C.

Class	Precision	Application	Preservation
A	Greatest ability of precision	Utility-scale systems	Heating and Ventilation
B	Advanced precision	Large commercial-scale	Heating
C	Basic precision	Residential and small-scale commercial or research	N/A

Table 10: IEC 61724-1:2017's Summarized Class Structure

The first pyranometer we reviewed was the SP-510-SS Upward-Looking Thermopile Pyranometer. This device is self-powered and has a calibrated output maximum of 100 mV. It can register data with a 180-degree field of view and has a detector response time of 0.5 seconds. In addition, each unit cost about $333. Now, this "upward-facing" sensor is functional alone. However, it is meant to be optimized by supplementing it with a respective "downward-facing" sensor, to form an albedometer; this is not ideal considering the design's complexity and high cost.

The second pyranometer we reviewed was the SP-110-SS: Self-Powered Pyranometer. This photodiode-based device is self-powered and has a calibrated output maximum of 400 mV. It can register data with a 180-degree field of view and has a detector response time of less than 1 milliseconds. Note that, in comparison with option 1, this pyranometer has a 500x faster response time and an ability to better amplify its millivolt readings. For option 2, each unit cost $223, which makes this the cheapest of all of our options.

The third pyranometer we reviewed was the SP-522-SS Modbus Digital Output Thermopile Pyranometer. This device is not self-powered and requires an input voltage of 5.5 V to 24 V. This means that we must incorporate a supplementary power converter circuit to power the device. Furthermore, this pyranometer can register data with a 180-degree field of view and internally converts the readings to a respective digital output. Each unit costs $433, making this the most expensive option.

The fourth pyranometer we reviewed was the SP-422-SS Modbus Digital Output Silicon Cell Pyranometer. Similar to the third option, this device is not self-powered and requires an input voltage of 5.5 V to 24 V. This means that we must incorporate a supplementary power converter circuit to power the device. Furthermore, this pyranometer also registers data with a 180-degree field of view and internally converts the readings to a respective digital output. This pyranometer, however, is a photodiode-based device, so each unit costs $323, making this much cheaper than the third option.

After reviewing all of the options (summarized in Table 11), we decided to go with option 2: the SP-110-SS: Self-Powered Pyranometer. This pyranometer boasted a lightning fast response time and does not warrant a supplemental powering circuit. In addition, this option was by far the cheapest. It's important to note that, while we chose a photodiode-based pyranometer, the thermopile pyranometers recognize a broader spectral range of 385 to 2105 nanometers, whereas the silicon cell pyranometers recognize a narrower spectral range of 360 to 1120 nanometers. This difference is not very significant in our project; both ranges cover the entirety of the visible light spectrum, extending just slightly into the infrared zone. Note that the visible light spectrum is where the majority of solar energy wavelengths transmit.

Option	Design	Self-powered?	Output	Price
1	Thermopile	Y	Calibrated output of 100 mV	$333
2	Silicon Cell	Y	Calibrated output of 400 mV	$233
3	Thermopile	N	Internally converts output to digital	$433
4	Silicon Cell	N	Internally converts output to digital	$323

Table 11: Summarized Properties of Prospective Pyranometers

4.15 Thermocouple

Solar cell efficiency is inversely proportional to temperature after a certain point. In fact, for every degree above 25°C, the maximum power rating of the solar panel falls by about 0.3%. In order to monitor the temperature levels within the solar panels, OUC has requested the integration of a thermocouple. A thermocouple is a minimalist device that consists of two wire ends made of different metallic alloys. While the first end has two split wires, the other end has two wires welded together. This welded junction generates a millivolt voltage whenever it experiences a change in temperature. The thermocouple was connected to the PCB, where the ADC can convert the thermocouple's millivolt output to digital output.

Thermocouples are commercially categorized based on their metallic alloy, temperature range, durability, vibration resistance, chemical resistance, and their compatibility. In Table 12, you can see the different classes that the thermocouples are classified under.

Type	Metallic Composition	Temp. Range (°F)	Margin of Error (%)
K	Nickel, Chromium, Aluminum, Manganese, Silicon	200 - 2300	0.4 - 0.75
J	Iron, Constantan	200 - 1400	0.4 - 0.75
T	Copper, Constantan	-330 - 600	0.4 - 0.75
E	Nickel, Chromium, Constantan	200 - 1650	0.4 - 0.5
N	Chromium, Silicon, Magnesium, Nickel	1200 - 2300	0.4 - 0.75
S	Platinum, Rhodium	1800 - 2640	0.1 -0.25
R	Platinum, Rhodium	1600 - 2640	0.1 - 0.25
B	Platinum, Rhodium	2500 - 3100	0.25 - 0.5

Table 12: Summarized Properties of Prospective Thermocouples

Unlike other technologies, such as the pyranometer, thermocouples do not vary much from one manufacturer to another. Moreover, Rubin has provided T-type thermocouples from the manufacturer Omega, so we used T-type thermocouples for the temperature monitoring in our sensor. When compared to the other thermocouple classifications, T-type does suit our circumstances the best. For instance, it has a suitable temperature range that spans temperatures well above what we expect to observe. With a margin of error of 0.4-0.75%, the millivolt output is not in danger of being very inaccurate. To reiterate, the temperatures we expect to observe amongst the panels should not exceed any more than approximately 200°F. Even if the solar panels reached 200°F, our readings could have an error of, at most, 1.5°F, which is negligible.

4.16 Analog to Digital Converter

In our circuit design, we depend on an analog-to-digital converter (ADC). This device converts whatever voltage or amperage value that enters our circuit to a digital value that can be read and processed by the single-board computer (SBC). The data was then communicated over to the node at the end of the string of solar panels, and then to the database. Because we have decided to employ the Raspberry Pi Zero W SBC, we explored three different ADCs that are seamlessly compatible with Raspberry Pi devices.

First, we reviewed the MCP3008, an 8-channel device with 10-bit resolution. The SPI interface is designed for very simple implementation, which fits our needs. The MCP3008 operates at 2.7 - 5 V, which can be easily obtained through basic circuit design. In addition, the controller can convert data at up to 200,000 samples per second using its low power CMOS technology. The controller can also handle up to 185 °F, which is convenient for our outdoor, high voltage system. The MCP3008 stood as the cheapest option at $3.75 per controller.

Next, we studied the ADS1015, which has 4 channels and achieves a higher resolution at 12-bits. With the ADSx controllers, the user can program the gain to be [up to] 16x, so the small signals can be amplified and read with high precision. Similar to the MCP3008, the controller operates at 2 - 5.5 V and consumes a small amount of current. However, the ADS1015 can only reach about 3300 samples per second, which is approximately 1/60th of the rate achieved by the MCP3008. The controller can handle a higher temperature of 257 °F, which easily surpasses any temperatures we expect to experience in our system. The ADS1015 is more expensive than the MCP3008, listed at $9.95 per controller.

Finally, we looked at the ADS1115. This controller is very similar to the ADS1015, but with some minor differences. With 4 channels and 16-bit resolution, this controller achieves the most precision of the three SBCs. However, it executes the lowest sample rate at 860 samples per second; this is 1/232th of the rate that the MCP3008. Like the ADS1015, the controller can handle a maximum temperature of 257 °F. Likely due to its immaculate resolution, the ADS1115 is listed at $14.95 per controller.

After considering all three ADCs, the most practical solution for our system is the MCP3008. It has the simplest design, which can be easily incorporated in our system with basic circuit design. Although the ADSx series ADCs both maintain higher resolutions, it is unnecessary for our scope and definitely not worth the significant increase in price. Moreover, the MCP3008 implements an SPI interface, whereas the ADSx ADCs utilize an I2C interface. For our project, SPI's ability to communicate with its peripherals quickly and effectively (via its full-duplex configuration) qualifies the MCP3008 as the preferred ADC for our project.

ADC Device	MCP3008	ADS1015	ADS1115
Channel Count	8	4	4
Resolution	10-bit	12-bit	16-bit
Sample Rate	200,000	3,300	860
Cost	$3.75	$9.95	$14.95
Interface	SPI	I2C	I2C

Table 13: The summarized characteristics of the prospective ADCs.

4.17 Enclosure

When installing our final printed circuit board, which is designed to read the voltage, current, temperature, and irradiance data and transmit it wirelessly, we ensured that the device can safely be installed outdoors. In the outdoors, a number of conditions can be expected: harsh weather, extreme temperatures, rain, and exposure to dust and dirt. All of these conditions contribute to the corrosion and overall degradation of electronic devices. To prevent this, we utilized an enclosure that satisfies the aforementioned NEMA standards. As previously stated, the NEMA 4X rated enclosures provide the most appropriate support for our sponsor's needs. The NEMA 4X rating certifies that the enclosure is designed to provide protection against dust, water, and ice, along with an extent of insulation.

Our enclosure must also provide more than enough room for our printed circuit board. Although our printed circuit board is about 3" by 3", and the pi zero is about 2" by 1", we needed ample room remaining in the enclosure for a variety of reasons. For starters, there needs to be enough room in the enclosure for installation. If the enclosure only provides an inch of room between the circuit board and the enclosure wall, then the installation would prove to be much more difficult. Moreover, the same logic applies to any maintenance the board may require. Another reason for leaving extra spacing in the enclosure is for ventilation purposes. Because of the relatively high levels of voltage and current pumping through each circuit board, there should be, for lack of a better word, *breathing room* for the circuit board to prevent overheating.

With our intentions in mind, we researched different NEMA rated enclosures that could fit our cause. First, we reviewed the polycarbonate corrosion-resistant washdown enclosures. These enclosures are made of polycarbonate and help fight against structural damage, such as denting, chipping, and cracking. In addition, not only is this enclosure NEMA 4X rated, but it is also NEMA 13 rated. A NEMA 13 rating indicates that the enclosures protect components from oil/coolant contamination. This enclosure is also equipped with quick-release latches,

which allow easy, frictionless access to the equipment inside without screws or clamps to unfasten the cover. Each enclosure starts off at $111.08 for its smallest size, which is 11” by 8” by 7”.

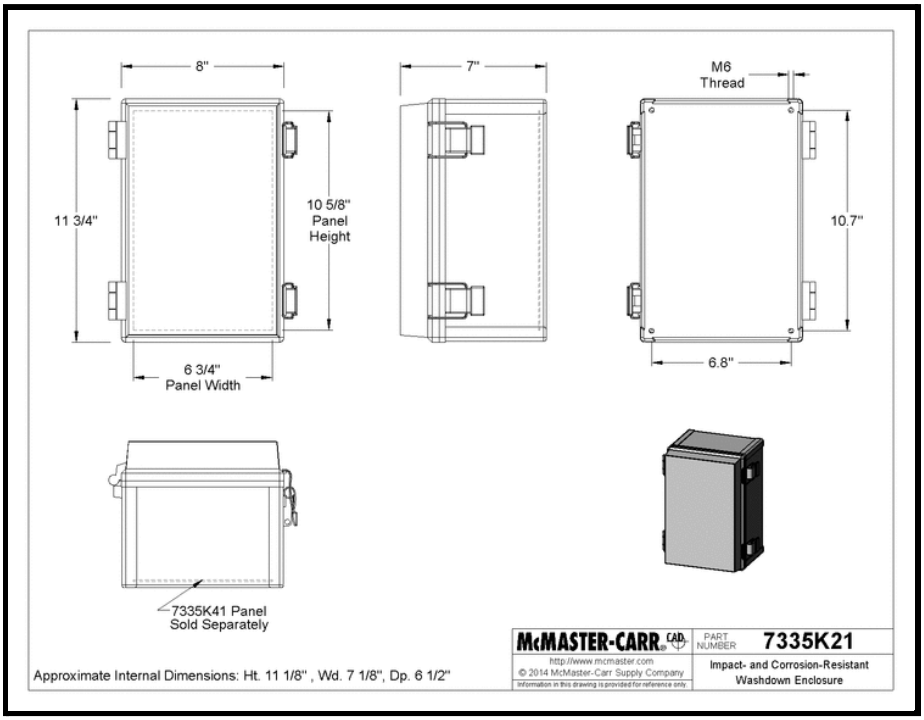

Figure 22: Layout of the washdown enclosure

Next, we reviewed the versa-mount polycarbonate washdown enclosures. These enclosures are also made out of polycarbonate, which helps fight against structural damage, such as denting, chipping, and cracking. These enclosures are rated IP66, NEMA 3S, 4X, and 13. Therefore, these enclosures are much more resistant against outdoor influences such as dust, oil/coolant contamination, water, dirt, and corrosion. Similar to the polycarbonate corrosion-resistant washdown enclosures, this apparatus also implements a quick-release latch. However, unlike the polycarbonate corrosion-resistant washdown enclosures, these enclosures are constructed with swing-out panels. Swing-out panels act as a false-front, and allow our team and future solar technicians to adjust the lighting seen in the enclosure, along with offering the ability to reconfigure the wiring behind the panel. These types of panels are particularly employed in applications that are exposed to outdoor conditions for long periods of time. This second option also comes with mounting brackets preinstalled. These brackets allow us to easily install the enclosures onto the steel beams that support the solar array. Additionally, the enclosure comes equipped with mounting pads, which allow us to orient our circuit board at various depths within the enclosure. The smallest unit is available as 6.5” by 6.5” by 5.5”, starting at $71.97.

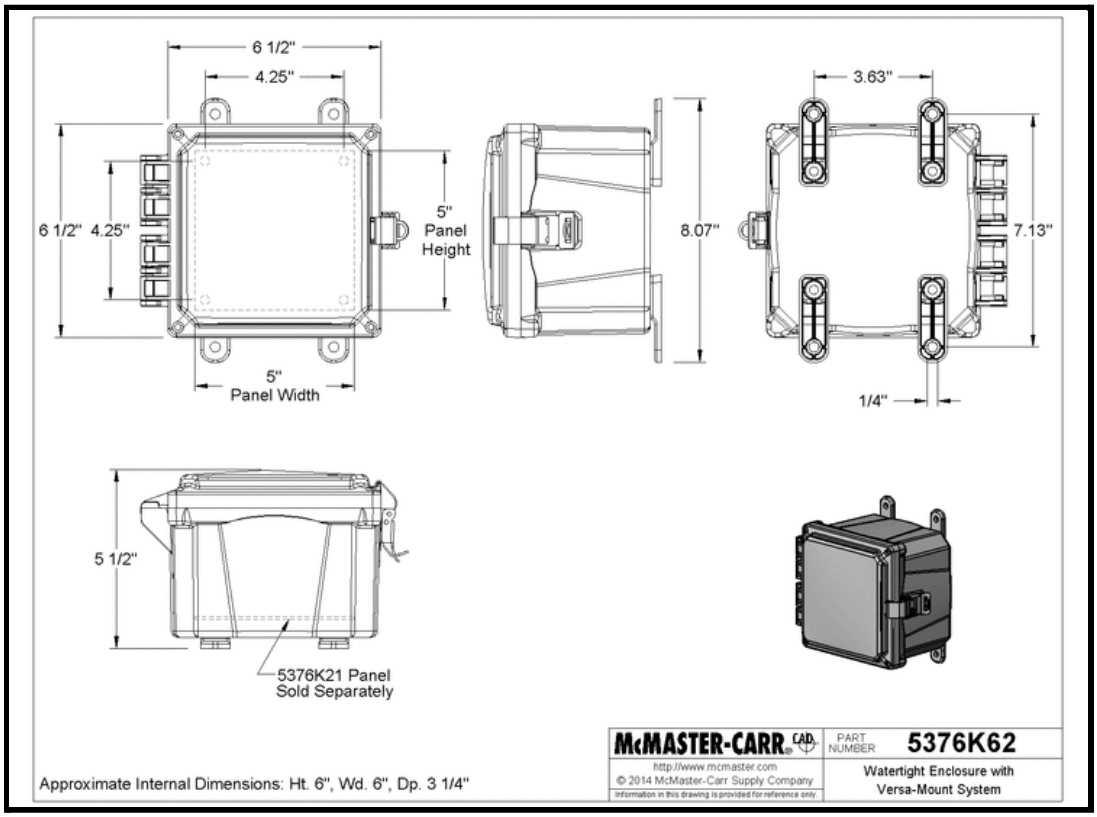

Figure 23: Layout of the versa-mount washdown enclosure

The final two units we reviewed were unique variations of the polycarbonate corrosion-resistant washdown enclosures and the versa-mount polycarbonate washdown enclosures. Here, the enclosures are designed with a see-through panel door. The transparent covers allow for our team and future solar technicians to visually inspect the components installed inside the enclosure without having to open the door. While the price of the polycarbonate corrosion-resistant washdown enclosure remains the same, this upgrade does increase the price of the versa-mount polycarbonate washdown enclosure to $77.07.

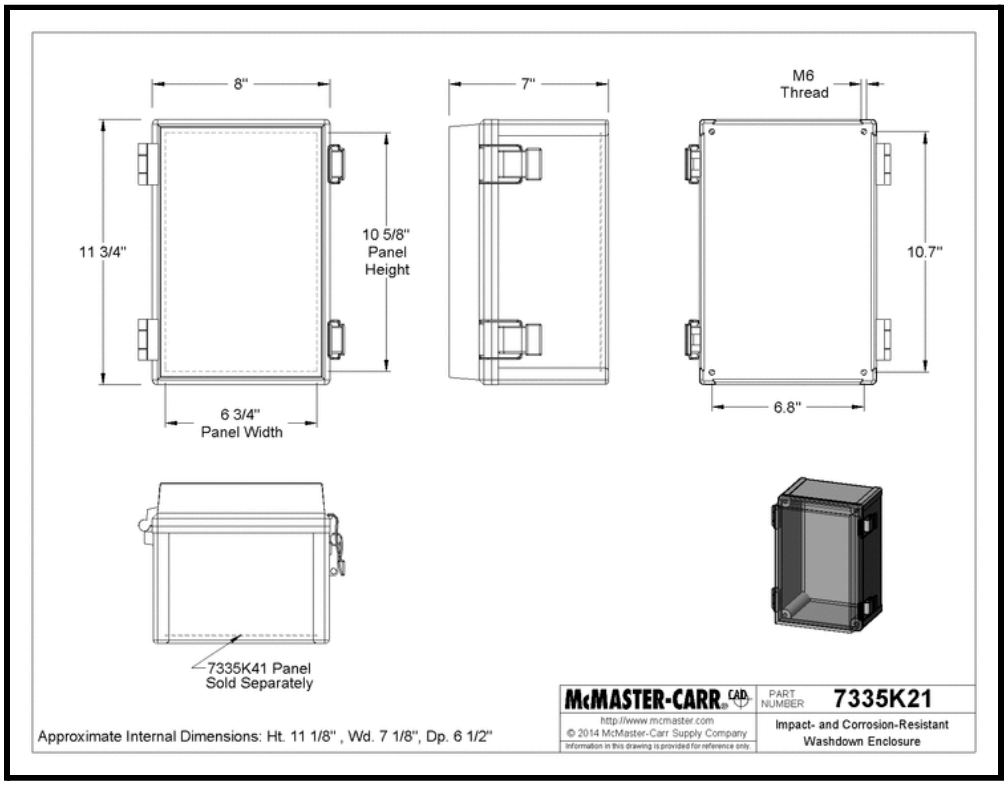

Figure 24: Layout of the washdown enclosure with transparent cover

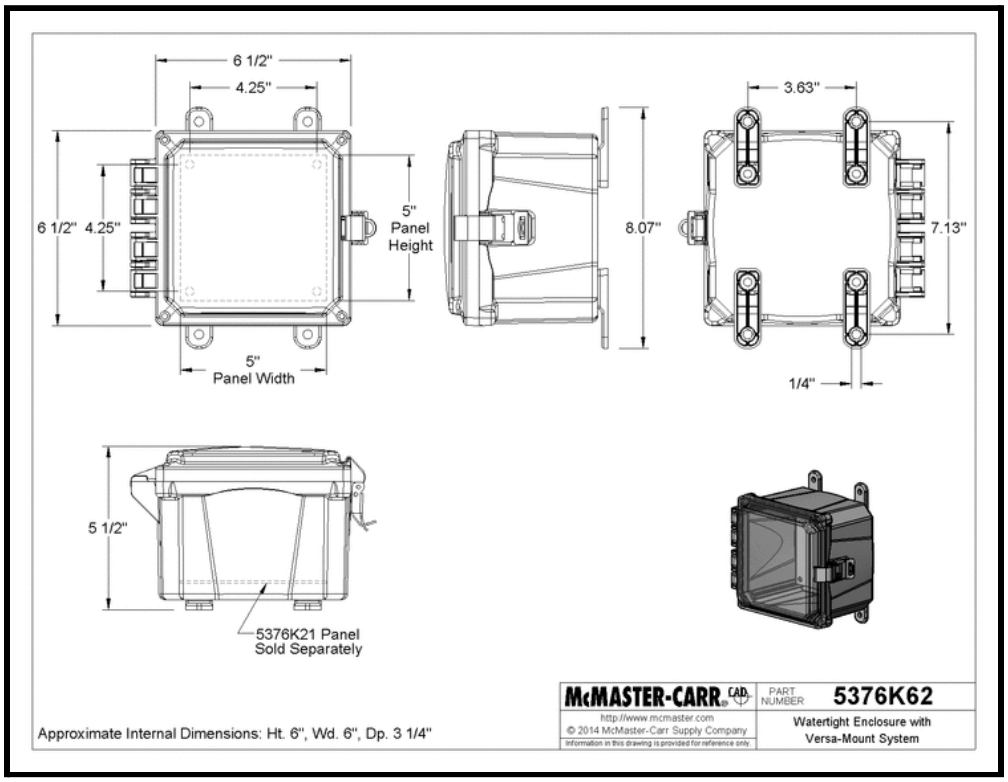

Figure 25: Layout of the versa-mount washdown enclosure with transparent cover

After reviewing all three options, our team decided to use the versa-mount polycarbonate washdown enclosure with the opaque panel-door. The polycarbonate corrosion-resistant washdown enclosure was a minimalist design, which was attractive. However, it's higher cost did not appropriately account for its lack of design features, compared to the second option. The versa-mount polycarbonate washdown enclosure is also designed to handle many more outdoor factors via its four different ratings. The size of the second option was also much more appropriate for our electronic devices. Considering the two devices combined fail to exceed 5" in length, it is not necessary to select an enclosure that is more than twice that size. As for the decision between an opaque door or translucent door, we decided that incorporating the see-through door would be a superfluous addition with an unnecessary increase in cost.

Enclosure	**Rating**	**Starting Size**	**Cost per Enclosure**	**Extra Features**
Polycarbonate Corrosion-Resistant Washdown Enclosures	NEMA 4X NEMA 13	11" x 8" x 7"	$111.08	N/A
Versa-Mount Polycarbonate Corrosion-Resistant Washdown Enclosures	IP66 NEMA 3S NEMA 4X NEMA 13	6.5" x 6.5" x 5.5"	$71.97	Swing-out panel. Preinstalled mount. Mounting pads.
Polycarbonate Corrosion-Resistant Washdown Enclosures with See-Through Cover	NEMA 4X NEMA 13	11" x 8" x 7"	$111.08	Transparent Cover.
Versa-Mount Polycarbonate Corrosion-Resistant Washdown Enclosures with See-Through Cover	IP66 NEMA 3S NEMA 4X NEMA 13	6.5" x 6.5" x 5.5"	$77.07	Swing-out panel. Preinstalled mount. Mounting pads. Transparent Cover.

Table 14: Characteristics of the prospective enclosures.

In Figure 23, we attached the dimensional layout of the Versa-Mount Polycarbonate Corrosion-Resistant Washdown Enclosure. This sketch provides us with a clearer insight on how the enclosure is constructed. In addition, the sketch gives us an idea on how we can approach designing the interior spatially. With a volume of 232.375 in^2, we should have ample amount of space to not only install our circuit board, but maintain it as well.

4.18 Extra Components

This section is a miscellaneous section comprising the remaining important design considerations for the PCB. None of these devices are integral for testing purposes, and each would not be considered a 'major component' of the final design, but each of them plays a specific and important role in facilitating a successful prototype. The terminal blocks are useful in that they connect the external features and inputs/outputs of the system to our board. The pin header connects our board to the Raspberry Pi Zero W to make a cohesive circuit board. The heat sink, regardless of its size, was useful in dissipating excess heat from our design to secure the integrity of the smaller components.

Terminal Block

In order to meet the high current requirement of our design, and to give us a simple way by which we can connect external features - and replace them if necessary - is by using terminal blocks. There are a few different choices in terms of terminal block placement from which we can select. First, we must consider what our inputs and outputs would be. We knew we needed two inputs (one for each the string and the panel) and two outputs for the same string and panel. The idea is that the circuit through the PCB will come from the string, into the panel, across it, back out of the panel, and back into the string. This was shown in our Visualized Prototype. Since each of these needs to be classified as its own signal, we needed four terminal block poles in total for the string/panel inputs and outputs. The poles are the individual bays into which a lead is screwed down.

For the external inputs (i.e. the thermocouple and pyranometer) we need one input each. Both of the inputs have two leads (unlike the previous string and panel inputs, which only have one input cord per MC4 connector, and so they each need two poles on a terminal block. In total, we need eight poles. Since the smallest mass-produced terminal blocks have two poles, we needed a maximum of six 2-pole terminal blocks. However, we could alternatively use four blocks for the string in/out, and a 4-pole block for the external inputs. This, we decided, was the less desirable choice as we would like plenty of space on the PCB to lay out signal planes for each of these inputs. Therefore, since each pole of a block is only a couple mm apart, they would be too close for our plans.

There are also several types of terminal blocks. There are screw, barrier, spring-cage, push-in, and tab connector terminal blocks. Due to their wide availability and ease of lead-removal, we have chosen to use screw terminal blocks. In particular, we used the 282834-2 terminal block from TE Connectivity as it has 2 poles, an acceptable wire gauge range of 16-30 AWG, and a max current of 10 A and voltage of 150 V.

Pin Header

The pin header choice is relatively simple in comparison to other components. We know, given that we would be outputting signals to a Raspberry Pi Zero W for transmission, and given that it uses a 40-pin connector, we would match it by using Adafruit's own 20x2 GPIO pin header.

Heat Sink

A way to reduce the risk of overheating within the enclosure that houses our electronic equipment would be to include a heat sink. Heat sinks absorb and disperse heat from electronics devices. They utilize various conductive materials that absorb the heat from the electronics. This heat is stored in conductive pins called fins, which then use convection to disperse the heat via ambient air flow. The ambient air flow is planned to originate from custom-made holes drilled into the enclosure. In turn, the degradation of our electronic devices could be slowed significantly if we incorporate a heat sink. Note that because we would implement them inside of our enclosure, we have to use passive heat sinks, which feature no mechanical components. Furthermore, having to fit and power a fan in the enclosure would be impractical and unnecessarily increase the cost.

Depending on how hot it may get within the enclosure, we could utilize aluminum heat sinks or copper heat sinks. In table 15, we can see that aluminum has a higher specific heat than copper. This allows for aluminum to absorb more heat without changing its internal temperature. However, it's copper's thermal conductivity that makes it such an effective option in the engineering industry. While copper can conduct heat more efficiently, it is also heavier and more costly.

Material	Aluminum	Copper
Specific Heat [J/g*°C]	0.9	0.385
Thermal Conductivity [W/m*K]	237	386
Cost	Cheap to produce	More expensive to produce
Weight	26.98 amu	63.55 amu

Table 15: Comparison between materials for heat sinks

Realistically speaking, the internal temperature of the enclosure should not get extremely hot; the enclosure's portfolio of NEMA and IP ratings, along with its PVC structure, should help the enclosure insulate pretty well. Moreover, we can expect the internal temperature to reach approximately the ambient air temperature plus the heat generated by the electric system. Given that our specific application takes place in Orlando, FL, we can expect ambient air temperatures to reach anywhere between 30 and 100 degrees over the course of the year. It is important to note that, unlike most other states, Florida tends to have very sunny weather even during the winter. In other words, the sensors are active year-round.

When it comes to determining the heat generated in the enclosure, we must inspect the devices that are housed within the enclosure. In particular, we must study the ambient temperature of the devices, the temperature the devices may reach during operation. While this information is explained in the devices' respective datasheets, another way to obtain a more accurate/practical answer is via experimentation. Upon the construction of our circuit, we can place the device within the enclosure, mimicking the actual future setup. Then, we can utilize a power supply to mimic the solar panel string voltage and current that we expect to face. As the 300 W courses through our circuit, each of the elements began to heat up. To reiterate, each of these elements should heat up to a measured ambient temperature, which is mentioned in table 16. After an appropriate amount of time, we can use a thermometer to measure the temperature within the enclosure and determine the circuit's collective ambient operating temperature. Note that because of the enclosure's NEMA and IP ratings, the PVC structure should insulate this heat fairly well; this could create an oven within the enclosure.

In addition to experimentation, we can also depend on the devices' datasheets for their ambient temperatures. In table 16, we highlighted the ambient temperatures for the components we are using inside of the enclosure. Their ambient temperatures correlate to how hot the component can get during use. However, we as engineers want our components to operate optimally. In order to do this, we want to make sure that the components can operate at the manufacturers' recommended temperature conditions. According to the information listed in this table, we can expect the heat radiating from the components to alter the ambient temperature to be

around 170 °C. Considering the rated operating temperatures are relatively lower than the expected ambient temperature, we considered a heat sink capable of dissipating a tremendous amount of heat.

In future testing, we decided to go with the empirical testing, where we measured the actual heat radiating off of operating elements. However, if this is not very effective, we can always fall back on the manufacturers' measured data and inspire our heat sink selection from there.

Component	**Operating Temperature Range** [°C]	**Maximum Ambient Temperature** [°C]	**Potential Power Dissipation [mW]**
TL072CDR Dual Low-Noise JFET-Input General-Purpose Operational Amplifier	-40 - 125	150	105
MCP3008 Analog-to-Digital Converter	-40 - 85	150	3
LMR50410xDBVR DC-DC Converter	-40 - 125	150	5150
Inductors	165	165	N/A
Capacitors	125	125	N/A
Resistors	155	155	N/A

Table 16: Components' temperature parameters

For the Raspberry Pi Zero, the datasheet explains that the device's maximum operating temperature is 70 °C (158 °F). In Orlando, Florida, this temperature can easily be reached within an enclosure housing utility-scale power equipment. In addition, heat sinks made for Raspberry Pi units are readily available and very cost-friendly. On account of the high temperature risk, we would accompany the Raspberry Pi Zero with its own heat sink. We reviewed two heat sinks designed for Raspberry Pi applications: an aluminum heat sink and a copper heat sink.

First, we reviewed the FIT0367 Raspberry Pi Copper HeatSink. This passive heat sink is a small appendage that depends on an adhesive strip to attach onto the Raspberry Pi. A unique

feature of this device is that it is made out of copper. Recall that copper is the most effective material for heat sinks. Because it is made out of copper, however, it is on the heavy side, weighing 10 grams. Copper's difficult production procedures drive the cost per unit to $0.99.

We also reviewed the Mini Aluminum HeatSink for Raspberry Pi. Similar to the copper heat sink, this device also depends on an adhesive strip to attach onto the Raspberry Pi. However, this device is made out of aluminum. While aluminum may not be as efficient as copper when it comes to heat conductivity, aluminum is naturally lighter and less expensive to manufacture. Therefore, this heat sink only weighs 0.7 grams, and costs $0.95.

After reviewing both heat sink options for the Raspberry Pi, we decided to implement the Raspberry Pi Copper HeatSink. This heat sink utilizes copper, which makes it much more impactful than the aluminum heat sink. In addition, considering the copper device only costs 4 cents more than the aluminum, we believe that utilizing the copper material would provide much more benefit in relation to the cost. Although this option is more than 14 times heavier than the aluminum heat sink, we do not believe that the added weight would contribute enough negative influence on the PCB to discard the copper's abilities.

Device	FIT0367 Raspberry Pi Copper HeatSink	Mini Aluminum HeatSink for Raspberry Pi
Weight	10 grams	0.7 grams
Cost	$0.99	$0.95
Material	Copper	Aluminum

Table 17: Prospective heat sinks for the Raspberry Pi

4.19 Database Selection

As part of our design, all the sensors in a given string of solar panels eventually relay all the data they collect to a node. This node housed a temporary database to store the data, which can then be sent in batches to OUC's main database. This allows a level of configurability as to how often the data is sent to the main database. It also allows the data to be traced back to the node that it came from, allowing the data indicating defective solar panels to be tied to a physical location.

This temporary data storage is an important middle-man to get the data to its final destination where it can be properly observed to locate and diagnose defective solar panels; it is essential that the right solution is chosen. As such, several key aspects of the database component were considered in making a choice of the available solutions.

4.20 SQL vs. NoSQL

Modern database solutions come in two main varieties: Structured Query Language (SQL) and not only SQL (NoSQL). The key difference is that SQL databases are relational, whereas NoSQL databases are non-relational. This distinction has several implications, the most relevant of which were considered in making a choice of a database technology. A visual representation of the relative strengths and weaknesses can be seen in Table 18.

Scalability

Early NoSQL databases were aimed at solving two problems with existing SQL databases. These are a lack of horizontal scalability, and the rigidity of the table design. Horizontal scalability is the ability to increase the capacity of the database by connecting multiple software or hardware entities together, such that they act as a single unit. Vertical scalability, on the other hand, is the ability to increase the capacity of the existing software or hardware by adding resources or processing power. In our proposed layout, each string of solar panels would have a single node which pulls data from every sensor in a string of solar panels. As such, horizontal scalability is desirable to be able to expand the network of nodes based on the size of a solar farm. As strings of solar panels are limited in size by the voltage ratings of the cables and other components, each node would only have to pull data from a relatively limited number of sensors. Because of this, horizontal scalability maintains a higher priority than vertical scalability.

Versatility

SQL databases are defined by the language used to define and manipulate the data within them. Though the querying language may be seen as antiquated or outdated, SQL has time on its side. It is still a popular schema for databases and is well understood through its decades of use since the 1970s. Queries in SQL are also generally flexible and able to handle a broad range of workloads. In contrast, NoSQL databases are more dynamic in their approach to storing and querying data. Data can be stored in a variety of ways, including documents, graphs, or key-value pairs, as opposed to SQL's rigidly defined table layout. As the data collected from the solar panels is relatively simple, and would lend itself well to a simple document or key-value storage schema, a NoSQL database seems to be a more fitting choice.

Performance

As a result of its adoption dating back over 50 years, SQL databases have seen many optimizations and improvements over time. SQL databases often have a reduced data storage footprint as compared to NoSQL databases as a result. However, these optimizations can come at the cost of careful up-front design to ensure the performance of the optimizations with SQL's rigid data models. NoSQL databases can also have high performance however, by limiting the range of their abilities to only the specific functions necessary for a given implementation.

Characteristic	SQL	NoSQL
Horizontal Scalability	✗	✓
Vertical Scalability	✓	✗
Querying Versatility	✓	–
Storage Versatility	✗	✓
Data Footprint	✓	–

Table 18: SQL vs. NoSQL Comparison

4.21 NoSQL Database Comparison

Once it was decided that a NoSQL database would be used, four of the most popular engines were chosen to be reviewed: MongoDB, Cassandra, Redis, and Firebase. While all four are NoSQL database engines, each have their own unique data storage schema and features.

MongoDB

MongoDB uses a document data storage schema, containing a JSON-like notation. The documents that make up the database contain text entries that share the same formatting as JSON. Each entry has a set of fields which correspond to a value, similar to a key-value schema. MongoDB's free tier accommodates up to 5 GB of cloud storage, and is open-source software.

Cassandra

Originally developed by Facebook, and now the Apache Software Foundation, Cassandra uses a wide-column storage schema. Due to its column-oriented design, its performance for queries which look up only a subset of the entire dataset are drastically faster than a typical

row-oriented database engine. However, this architecture also makes incrementally adding data to the database slower than usual, making it unsuited to our use case.

Redis

Redis is another open-source NoSQL database engine. It uses a key-value storage schema. Each value stored in the database is assigned a key, which is then used to access that value (similar to a hash map or a Python dictionary). One of Redis' unique features is that it holds the key-value pairs in main memory, at the cost of some data durability.

Firebase

Firebase is Google's entry into the NoSQL database engine market, having acquired it in 2014, only two years after its inception in 2012. It uses a fairly unique data storage schema; a JSON tree with a unique key for each node. Being one of the newer offerings, Firebase has plenty of unique features which make it stand out, including real-time synchronization and Firebase Authentication.

Conclusion

After considering some of the relevant characteristics of SQL and NoSQL databases, the choice was made to use a NoSQL database due to their horizontal scalability, versatile storage schemas, and high performance. Deciding between the various options of NoSQL databases was more down to preference and suitability to the project than performance or other characteristics. In the field of strong community support, MongoDB stands out, as it has existed as part of some popular full stack applications for over a decade. This makes MongoDB much more attractive from a programming standpoint than some of the newer offerings with less usage.

Because the Raspberry Pi 4 allows for expandable storage via a micro SD card, the database is designed to be stored locally on the node. However, it is always a good idea to have a solid backup plan, and MongoDB offers that with the largest free tier, at 5 GB of cloud storage. Additionally, our project sponsor, Rubin York, has another ongoing project with OUC that utilizes a MongoDB database to interface with OUC's main SQL database. To avoid compatibility issues and any additional overhead of introducing a new database, MongoDB was the right choice for our implementation.

4.22 Collector Node SBC Selection

As explained in the visualized prototype (Figures 3 and 4), each of the sensors in a string of solar panels relay the data they collect to a collector node. The collector node serves as the final connection to OUC's main database. The collector nodes have several other functions including providing a reliable ethernet connection for data transmission, housing a small database to enable batch processing of the data, and executing the software which pulls the data from each sensor in the string.

In order to meet these needs, the collector nodes were a Single Board Computer (SBC), of which three popular models were considered and compared against each other to find the best suited one for the job.

Performance

Performance is one of the most important considerations in nearly any embedded system. That being said, the performance requirements for each use case drastically differ based on the expected workload of the computer. In order to determine whether performance should be an important consideration in deciding which SBC to use for the collector nodes, some calculations were done.

What we know about the data being collected:

- 1 floating point variable in Python requires 8 Bytes of memory.
- Data will be sampled every 2 to 10 seconds.
    - Assume a 2 second sampling period for the worst-case scenario.
- Each sensor will collect 2 to 4 floating point values.
    - Assume 4 values will be collected for the worst-case scenario.

To calculate the amount of data each sensor would produce per time period:

$$\frac{bits}{second} = \frac{8\ Bytes}{1\ float\ value} \times \frac{8\ bits}{1\ Byte} \times \frac{1\ float\ value}{2\ seconds} = 32\ bps$$

- Most solar panels produce a voltage of either 12 V or 24 V.
    - Assume 12 V for the worst-case scenario.
- The requirement specification for the maximum voltage is 700 V.

To calculate the number of solar panels in a string, in the worst-case scenario:

$$\frac{solar\ panels}{string} = \frac{1\ solar\ panel}{12\ volts} \times \frac{700\ volts}{1\ string} \approx 59\ panels/string$$

To calculate the amount data each node would be pulling from the sensors in a string per time period, assuming there exists 1 sensor for each solar panel, in the worst-case scenario:

$$\frac{bits}{second} = \frac{32\ bps}{1\ sensor} \times 59\ sensors = 1,888\ bps \approx 1.9\ kbps$$

Assuming all the worst-case scenario conditions for the data transmission rate, the amount of data collected by a single node only amounts to 1.9 kbps, which only saturates 0.19% of the available bandwidth of a Gigabit ethernet connection, which all three SBCs have (see Table 19). This does not account for the possibility of batch processing. However, even if the batches of data were to be sent to the main database in periods of 24 hours, one day's worth of data for a node would still only amount to 164.16 Mb (20.52 MB), which is still a far cry from bandwidth restrictions. As such, performance is not something that was considered in deciding between the three SBCs.

Characteristic	Raspberry Pi 4 Model B	Banana Pi BPI-M5	NanoPi M4B
Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz	Amlogic S905X3 Quad-Core Cortex-A55 @ 2.0 GHz	Dual-Core Cortex-A72 @ up to 2.0GHz + Quad-Core Cortex-A53 @ up to 1.5GHz
RAM	1 GB - 8 GB	4 GB	2 GB
PoE Support	Yes - w/ PoE HAT ($20.00)	No	No
Wireless Capability	2.4 GHz and 5.0 GHz IEEE 802.11ac Wi-Fi, Bluetooth 5.0, BLE	Wi-Fi - with optional USB dongle	802.11a/b/g/b/ac, Bluetooth 5.0
OS Support	Raspberry Pi OS, LibreELEC, OSMC, Recalbox, Lakka, RISC OS, Screenly OSE, Windows 10 IoT Core, TLXOS	Android 9.0, CoreELEC (sample build), Ubuntu Mate Desktop 20.04, Ubuntu Server 20.04, Debian Buster	Android 8.1, Android 10, Lubuntu 16.04, FriendlyCore 18.04, FriendlyDesktop 18.04, FriendlyWrt 19.07.1
Power Requirement	5V/3A via USB Type-C or GPIO header	5V/3A via USB Type-C	5V/3A via USB Type-C or GPIO header
Operating Temp. Range	0°C to 70°C	N/A	-20°C to 70°C
Dimensions and Weight	88mm x 58mm 46g	85mm x 56mm 48g	85mm x 56mm 51g
Price	$30.00	$60.00	$70.00

Table 19: Comparison of Three SBCs for Collector Nodes

4.23 Sensor SBC Selection

Each sensor placed along a string of solar panels was responsible for collecting at least two and up to four data values. The hardware design is responsible for converting the relatively high string voltage and current to more reasonable values to be analyzed. These signals can then be collected by a single-board computer in each sensor housing. The SBC was responsible for converting these electrical signals into data which can be manipulated in the software, and later transmitted to the collector nodes at the end of each string. Several important features were considered in selecting a suitable device.

Processing Power

SBCs as small as the Raspberry Pi Zero W, Banana Pi BPI-M2 Zero, and Orange Pi i96 are built primarily to be small, power efficient, and inexpensive. Performance is secondary to these qualities, as for the typical use cases, the SBCs do a relatively small amount of work. As such, performance benchmarks for these SBCs are few and far between, and are mostly community projects performed by individuals who only have access to a few different SBC models. As a result, manufacturer core speed estimates and other numerical performance measures were used to evaluate their relative performance.

The Banana Pi BPI-M2 Zero is the obvious standout performer based on its technical specifications. It houses a quad-core Cortex-A7 CPU, where each core has a typical clock speed of 1.5 GHz. In contrast, the Raspberry Pi Zero W and Orange Pi i96 both have only single-core ARM1176JZF-S and Cortex-A5 CPUs, respectively. However, it must be considered that the Banana Pi BPI-M2 Zero's powerful processor would only have a significant advantage in highly multi-threaded workloads, of which our use case is not. Comparing the single-core performance of these ARM cores is more relevant to our use case.

To measure the performance of a single ARM core, ARM provides a DMIPS/MHz rating for each of their core architectures. DMIPS/MHz is simply a measurement of the number of calculations a CPU can do per time at a certain clock frequency. For example, a 1.5 DMIPS/MHz CPU can be expected to do 150 million instructions per second when running at 100MHz. Unfortunately, while a DMIPS/MHz rating may give a vague representation of a CPU core's speed, it lacks the ability to accurately represent many real-world applications of embedded processors. The DMIPS/MHz score is formulated from a pool of synthetic scientific computing benchmarks, which fails to take into account some important factors, such as long idle times. Nevertheless, if this performance measure is to be taken into account, the Banana Pi BPI-M2 Zero still has the fastest processor, with a score of 1.9 DMIPS/MHz, followed by the Orange Pi i96 with a score of 1.57 DMIPS/MHz, and the Raspberry Pi Zero W with a score of 1.25 DMIPS/MHz. Adjusting for clock speeds reveals that the Banana Pi BPI-M2 Zero should be about 88% faster than the Zero W.

In the worst-case, the data acquisition programs which were run on these SBCs would be on the order of tens of millions of instructions. Given that the Raspberry Pi Zero W's DMIPS/MHz score of 1.25 translates to its processor being able to do 1.25 billion instructions per second, this should leave plenty of headroom to achieve the desired scan rate between 2 to 10 seconds. Upon realizing that performance is far from being a limiting factor in achieving our goal, the other features were weighed more heavily in choosing the right SBC.

Caching Storage

The SBCs housed in the sensors won't have access to either OUC's database or the temporary database accessed by the nodes. Because of this, any data that needs to be stored had to be cached locally on the SBC's RAM or onboard storage. In our design, the sensors won't be responsible for storing any data, as the node is the device responsible for initiating data collection. The node's primary job is to "pull" data from each sensor in a string of solar panels at the preconfigured scan rate; it would handle any intermediate data storage before the data arrives at OUC's main database. Nevertheless, if the software design were to be changed, all of these SBCs have adequate memory capacity and expandable storage to allow for some batch processing from the sensors.

Power

Power requirements are another important consideration to be made to decide between these three SBCs. As power for the sensor device was siphoned from the power that the solar panels generate, it was in all parties' best interest that they be as power efficient as possible. Unfortunately, the only power requirements that are easily accessible for these devices are their recommended power input ratings. All three share similar ratings, with the Banana Pi BPI-M2 Zero and the Orange Pi i96 requiring 5V @ 2A, whereas the Raspberry Pi Zero W requires 5.1V @ 2.5A. These ratings don't tell us the full story however, as what is really crucial for our use case is their power consumption while idle and with WiFi or Bluetooth on.

There are some other features which may help to lower power draw however, such as the Raspberry Pi Zero W's Bluetooth Low Energy (BLE) mode. Conventional Bluetooth takes about 100ms to initiate a connection, and then remains connected for long periods of time. In contrast, BLE remains in sleep mode until it receives a connection request, where it can connect as fast as 1ms. As a powerful CPU is something that was deemed to be of lower priority, another consideration to be made is that a less powerful CPU may be desirable to reduce power consumption. With the limited knowledge we have of these device's power draw without testing each in our exact scenario, the Raspberry Pi Zero W seems like the right choice because of its BLE feature and its processor which is likely to draw less power than the alternatives.

Physical Limitations

As with most embedded systems, one of the primary considerations to be made are any physical limitations present in the design. In our scenario, these include mainly size and temperature restrictions. The SBC was housed in a NEMA housing, along with the other components and wiring that the sensor consists of. Luckily, all three of these SBCs are very small, and would likely fit alongside the other components in even the smallest commercially available NEMA housings. These devices are also extremely light, alleviating any weight concerns.

Temperature restrictions are however something to be worried about, due to the nature of a solar farm. While solar panels don't generate electricity via heat, ambient heat is always present in an environment which is directly exposed to sunlight for many hours. Computer electronics also produce heat during normal operations, but it is important to keep the components at reasonable temperatures to not accelerate the normal degradation of the electronics. Of the three SBCs, the Raspberry Pi Zero W's acceptable operating temperature range reaches the highest, at a scorching 70 °C (158 °F). While this may seem to be cutting it close, the sensors were mounted underneath the solar panels, limiting their heat exposure to just the ambient heat from the other electronics and the solar panel, rather than the direct sunlight. Concerningly, the Banana Pi did not list any operating temperature specification for the BPI-M2 Zero, making it a difficult choice for our project.

Price

One of the main motivations for this project is to reduce downtimes for solar farm operators, which is indirectly a profit motive. Whether it be by reducing the likelihood of fines from failing to meet power production quotas, or data collection for solar energy research, the usefulness of this project hinges a lot on its price. While prototypes are expected to be much more expensive than the envisioned mass-produced final product, it is still an important consideration to be made. Fortunately, the Raspberry Pi Zero W and Orange Pi i96 are widely available at inexpensive prices. In fact, the Raspberry Pi Zero W and Orange Pi i96 can be found for only $10.00 and $12.00, respectively (some price fluctuations are to be expected). This made the Banana Pi BPI-M2 Zero a hard sell for our use case, with its comparatively high price of $30.00.

Characteristic	Raspberry Pi Zero W	Banana Pi BPI-M2 Zero	Orange Pi i96
Processor	Single-core ARM1176JZF-S 1 GHz	H2+ Quad-core Cortex-A7	Single-core ARM Cortex-A5 32bit
Processor DMIPS/MHz	1.25	1.9	1.57
RAM	512 MB	512 MB	256 MB
Storage	MicroSD (variable)	MicroSD (variable)	512MB, TF card (variable)
Wireless Capability	2.4GHz 802.11n	Wi-Fi 802.11 b/g/n	802.11b/g/n 2.4GHz
Bluetooth Capability	Bluetooth Classic 4.1, BLE	Bluetooth 4.0	Bluetooth 2.1
OS Support	Raspberry Pi OS, LibreELEC, OSMC, Recalbox, Lakka, RISC OS, Screenly OSE, Windows 10 IoT Core, TLXOS	Android, Linux	Android, Ubuntu, Debian Image
Power Input	5.1V @ 2.5A	5V @ 2A	5V @ 2A
Operating Temperature Range	0 °C to 70 °C	N/A	-10 °C to 65 °C
Dimensions and Weight	65mm x 30mm 9g	65mm x 30mm 15g	60 mm x 30mm 30g
Price	$10.00	$30.00	$12.00

Table 20: Comparison of Three SBCs for Sensors

4.24 Comparison of WiFi and Bluetooth for Data Transmission

In the initial concepts of our design, one of the main topics for decision was whether to equip the sensors with wireless communication abilities, or to have more reliable, but harder to set up wired connections. With the decision to commit to a wireless setup for data transfer, we have since kept our options open as to which wireless technology to use and how to implement it when considering our choices of hardware. The Raspberry Pi 4 and Raspberry Pi Zero W both include a Wi-Fi antenna, Bluetooth, and a Bluetooth Low Energy mode. When deciding which of these technologies to use to send data from the Raspberry Pi Zero Ws in the sensors to the Raspberry Pi 4 nodes, several factors were considered (see Table 21 for a summary of the considered features).

Signal Range

Our design is meant to be adaptable to solar panel strings of different lengths, such that the sensors in a string can be spread out depending on the number of sensors and the number of panels in the string. As such, a longer range wireless technology is preferably to give the solar farm operator more flexibility with the physical distances between the sensors. In this department, WiFi is the clear winner. The Raspberry Pi Zero W's class 2 Bluetooth transmitter is rated for about 33 feet of range [26], with users reporting real ranges in a similar ballpark. In contrast, its WiFi transmission range can be expected to have many times greater range, with some users reporting ranges upwards of 150 feet. It is important to note that these range numbers are in open-air tests, with no walls or other obstacles to reduce range. Fortunately, our scenario is near open-air, as the sensors would only need to transmit signals along a string of solar panels. Visualized in Figure 26 is an approximation of the distances between the node and the sensors in the configuration they were tested in at OUC's solar research array in Pershing, Orlando.

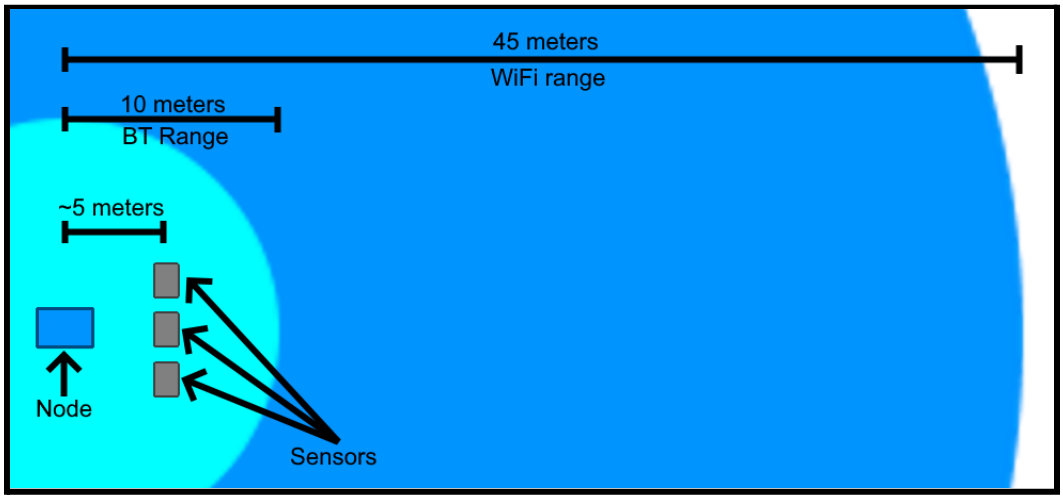

Figure 26: Visual Comparison of WiFi and Bluetooth Ranges

Power

As with all the hardware components in our sensor design, power consumption is an important consideration. Ideally, we should be drawing as little power from the solar panels as possible to minimize the observer effect. In this category is where the Raspberry Pi Zero W's Bluetooth Low Energy mode really shines. Prasad [27] states that when transmitting ten messages per day, a WiFi device required 500 μW of power, while BLE required only 50 μW to accomplish the same task. A ten times reduction in power consumption is certainly an enticing reason to go with BLE over WiFi. This also opens up other options, as BLE's lower power consumption could enable faster scan rates without consuming too much power, while also generating a faster and more up-to-date data stream.

While BLE's advantage over WiFi in power consumption may seem to make BLE an easy choice, it is important to consider that this is a direct consequence of its significantly reduced range. It may be best to be able to choose which one to use based on the specific layout of the sensors and nodes in a particular solar farm. Large solar farms which may only desire to have a few sensors per string could use WiFi, and smaller farms with lower range requirements could use BLE to reduce power consumption.

Data Transfer Rate

As discussed in the collector node SBC selection section, the amount of data being transferred is only on the order of kilobits per second in the worst-case scenario. Luckily, either choice's maximum data transfer rates far exceed our requirements, with Bluetooth reaching up to 3 Mbps and WiFi reaching speeds as high as 54 Mbps [26]. Arellano [26] even mentions that as a result of its lower maximum speed, "Bluetooth is typically used for transferring small chunks of data, such as the numerical values from IoT sensors", which perfectly suits our use case.

Additional Considerations

Some other features which may prove to be useful are location detection and security protocols. To be able to determine the physical location of defective solar panels, the data could include location information. While this would increase the amount of data to be transmitted, it could make installation easier, as sensors wouldn't have to be manually identified by inputting their physical location in the software. Both WiFi and Bluetooth can provide accurate location information, though Bluetooth may be more reliable in some scenarios [26].

For various reasons, a solar farm operator may want to protect the information they gather on their solar array. In OUC's case, the data they collect on their research array is important to make decisions for larger commercial implementations, and as such it is vital that the information not be tampered with. If Bluetooth's security is seen as insufficient, WiFi can add

an additional layer of security using one of several protocols: WEP, WPA, WPA2, and WPA3 [26].

Characteristic	WiFi	Bluetooth
Approximate Signal Range	45 m (150 ft)	10 m (33 ft)
Approximate Power Consumption per Message	50 μW	5 μW
Data Rate	54 Mbps	3 Mbps

Table 21: Comparison of WiFi and Bluetooth

4.25 Comparison of Scripting and Programming Languages

A programming language is a pivotal tool for any software developer. It acts as the bridge between the programmer's mind and the computer's abilities, translating what the programmer wants to happen into instructions that the computer can understand. Programming languages represent a very abstracted approach to instructing a computer to perform tasks, as there are many layers between the code that is typed in a text editor or integrated development environment, and the eventual binary machine code that is directly executable by a computer's processor. Though taking all the levels of abstraction in between out of the equation can be seen as a loss of control, it also enables far more complex development than would have ever been thought imaginable when programmers had to write assembly code, or even flip physical switches as a sort of pseudo-binary code. As with any software product, deciding which programming language to use is the first major step towards implementation, and likely one of the most important. Different languages are suited for different tasks, and some are even purpose built for a specific implementation. In relevance to this project, the first distinction that was made are the differences between programming languages and scripting languages.

The primary difference between programming languages and scripting languages are the pathway taken to make them understandable to the computer. The first step to making most programs executable is compilation. Typically, a compiler would translate a high-level programming language to assembly language, but it could also translate it directly to machine code. For example, gcc, a popular collection of compilers for C code, translates C code into machine code. Assembly language needs the additional step of an assembler to translate it into machine code before it is understandable by a processor. In contrast, scripting languages do not use a compiler to be translated into machine code, and instead are interpreted. Rather than translating high-level code into an executable file which can then be executed at a later time, the

interpreter does the translation to machine code during execution time. This has several consequences for the behavior of the program.

One of the many consequences of interpreted scripting languages is their inability to perform static code analysis. Static code analysis encompasses any method of debugging code by examining the source code before execution. Instead, interpreters must perform dynamic code analysis, and as a result are more likely to produce errors during runtime. This is mostly a developmental preference, as ideally the finished product should be bug-free, but some issues can be easier to detect with static code analysis. One such example could be conditional statements that rarely trigger. Compiled programs are often faster than interpreted ones. This is because the translation to machine code is completely finished before execution starts, as opposed to interpreted programs which have to be translated and then executed in real-time. Scripting languages have one major upside however; they are generally less code intensive than programming languages. This is evident in the upwards trend in languages like Python and JavaScript, for which the trade-off of execution speed for development time is favorable to many software companies.

Because of these differences in scripting languages and programming languages, there are some tasks which better suit one or the other. For example, scripting languages are purpose-built for automation. While many compiled languages require some kind of runtime environment to be installed, scripting languages' runtime environments often come pre-installed on the target system. Sometimes this can be advantageous however, such as in cases like the Java Virtual Machine (JVM). Since Java code is compiled into bytecode which executes on the JVM, any platform on which the JVM can be installed is capable of executing Java code. Fortunately for our use case, most Linux derived operating systems are capable of executing Python and Bash scripts out of the box. This is particularly advantageous in our use case, as the available storage on the Raspberry Pi 4 and Raspberry Pi Zero W relies on the size of the installed micro SD card, and for cost reasons the smaller the micro SD card can be, the better. Because scripting languages are targeted at purposes like the automation in our design, it is the obvious choice over any programming language.

4.26 Scripting Language Comparison

With the decision to use a scripting language made, the next step is to compare some of the available options. Within the world of internet of things (IoT) devices and SBC implementations, there are many popular options for scripting languages. The three that are to be compared here are selected mostly due to familiarity due to UCF's course curriculum or personal experience. They are Bash, JavaScript, and Python.

Bash

Bash is the command language that is used in most Linux shells, as a part of the GNU Project. The name Bash is an acronym referencing the shell that it replaced, the Bourne shell, where Bash means "Bourne Again Shell". As Bash comes pre-installed with mostly any Linux distribution (including Raspberry Pi OS), almost anyone who has used Linux before is at least familiar with some of its functionality. Though its most popular use is in command line interfaces (CLIs), Bash can also be used to write scripts. It is a fairly simple process to write scripts with Bash. Creating a new text file via a CLI or GUI in Linux and changing the file extension to .sh is the first step. The first line of any Bash script starts with a line that tells the interpreter the file location of the Bash binary file. This line is commonly referred to as the "shebang".

With the shebang line written, the possibilities of Bash scripting are unlocked. Writing the rest of the script is done just as entering commands into a CLI, only they are written as text in the script, and are executed in order as with mostly any programming or scripting language. This makes Bash have the least intensive setup of the three scripting languages compared here. Where Bash starts to lack for our implementation is the options for interfacing with the Firebase database. Unfortunately, there currently exist no tools to make queries to a Firebase database from a Bash script. There are some tools which can utilize Bash scripts through some secondary layer, such as the node package manager's (NPM) Firebase Shell which uses a SQL-like syntax, but none which work directly.

Though Bash won't be useful in writing the automated scripts that will be used to gather and transmit the data from the sensors, it can still be an important tool to set up Firebase on the Raspberry Pi 4 and Raspberry Pi Zero W. This can be done through the Firebase CLI, which can be used for a variety of commands such as fully or partially deploying a Firebase project to the device, or authenticating the CLI to a Firebase account.

JavaScript

JavaScript is a popular language for web development, being present in multiple layers of the MERN stack, as well as other frameworks and implementations. Though our project has no web development needs, JavaScript has many advantageous characteristics for IoT-type projects. The primary selling point of JavaScript is the Node.js framework. Node.js is an open-source JavaScript runtime environment. It is often used for data-centric solutions, which makes it well suited to our use case. Node.js comes with the NPM, which is compatible with the Raspberry Pi 4 and Raspberry Pi Zero W.

Aside from the Node.js framework, JavaScript's programming style also makes it unique. This is because JavaScript is aimed for use in what are referred to as event-driven applications.

Event-driven applications are those in which the program functions are meant to react in real-time to other program functions and outputs. An example of this are event loops in JavaScript, which allow multiple different devices to react to the same event. Aside from the real-time response benefits of the event-driven programming available in JavaScript, it also has the benefit of optimizing power consumption. This makes a compelling argument for using JavaScript in our sensor implementation, as the software design has the node pulling data from the sensors in series, one after the other. Using JavaScript could enable simultaneous data pulling, which could allow for more data to be pulled in the same amount of time, while not significantly impacting power consumption.

Python

Python's popularity is for much different reasons than most programming or scripting languages. It is known as a very beginner friendly language that is easy to pick up and learn for a variety of reasons. For one, Python programs are often shorter than their counterparts, with lots of the grunt work being done behind the scenes. It also has standardized whitespace usage, which makes reading other programmer's software less daunting. Its simple syntax and range of built-in function make it an ideal choice for many applications.

One of Python's largest drawbacks is its performance - or lack thereof. Python's ease of use comes at a cost, and that cost is its execution speed. This is a tradeoff that most are willing to make - paying a programmer for more working hours is often not worth the difference in execution time difference between a program written in Python and something faster but more code intensive, such as C. The extra time invested to create a faster program isn't just a concern for employers however, this is a relatively popular viewpoint. It is evidenced by the fact that Python is currently the number two most popular programming language on the near-ubiquitous code repository GitHub, just behind JavaScript. The real question to ask for hobbyist programmers and for projects like ours, is whether Python's performance would become a limiting factor in the implementation. After all, if there are components of the design which are slower than the program, or a faster execution simply isn't necessary, then the choice is clear.

For our project, computational power was not a limiting factor. This is detailed in several sections, including the node and sensor SBC selections. Our team is far from the first to come to this conclusion, as Python is the language of choice for the Raspberry Pi. Just as it is one of the strengths of the Raspberry Pi, Python also benefits from a large community of developers with different backgrounds and varying uses and implementations of the scripting language. This is a benefit that is not evident in many comparisons, as it isn't measurable. However, any programmer knows how valuable community resources can be to get a program working in the intended fashion. Online forums like StackOverflow come to mind, where it can sometimes be hard to find advice or information on lesser used or up-and-coming languages and software technologies. All in all, Python seems like the right choice for our project.

4.27 Communication Protocols for Raspberry Pi Zero W

Our circuit design produces millivolt readings which correspond to the various measurements of which we intend to collect data on. Once the circuit has produced these readings, its job is complete, and the readings are to be read by an ADC, converted to digital, and then handed off to the Raspberry Pi Zero W to transmit the data to a node. This last step in transmission is the focus in this section; the interface by which the Raspberry Pi Zero W reads the signals it receives on its GPIO pins through the ADC and makes them transmittable data via the software.

Microcontrollers (MCUs) and single-board computers (SBCs) have a few options when it comes to hardware communication interfaces. These interfaces are methods for the programmer to communicate with and make use of the hardware on an MCU or SBC through the software. They can also be used to communicate between different hardware components on the same board.

Apart from their ability to provide an interface for the programmer, these communication protocols also all have some unique features and architectures. These differences make some of them better suited to certain tasks, and as such some of the more popular choices were compared to determine which best suits our use case and is compatible with our hardware. To add another layer of complexity, these communication protocols often have multiple options in terms of implementation, which was discussed and weighed in our decision on which one to use for our implementation.

For the purposes of this project, three of the more popular hardware communication protocols that are available on the Raspberry Pi Zero W were compared. These included UART, SPI, and I2C.

UART

UART is the first of three communication protocols that was compared for use in the Raspberry Pi Zero W. It stands for Universal Asynchronous Reception and Transmission. UART works by having both of the communicating devices have a UART setup in which there are two transmission lines. One line is for transmission, referred to as the TX line, and the other is for receiving data, called the RX line (see Figure 27 below).

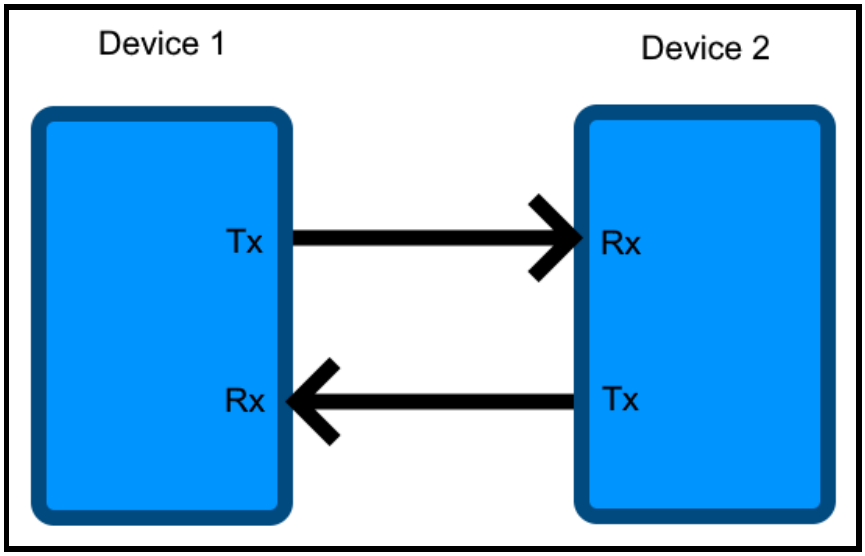

Figure 27: Example UART Interface

The main unique feature of UART is evident in its acronym; it is asynchronous. This differs from most hardware communication protocols in that it does not require a clock signal. These clock signals are used to synchronize data transmission, as well as provide the devices using UART a method of determining the beginning and end of individual messages made up of bits called a data packet. In the case of UART, each data packet has a start and stop bit appended to it before it is transferred, so that the receiving device can properly understand the data packet.

As UART does not use a clock signal for synchronization, it relies on a fixed transmission rate to resolve any synchronization conflicts. It also requires that both devices communicating with each other have the same transmission rate. UART's data transmission speed is measured as a BAUD rate. The BAUD rate is in units of bits per second, e.g. 9600 BAUD = 9600 bps. The default BAUD rate is 115,200 or roughly 115 kbps. Fortunately for our use case, this rate far exceeds the rate at which we would need to transmit the 10-bit data packets from the MCP3008 ADC. In this scenario, the BAUD rate can be configured to be much lower to reduce power consumption.

UART is capable of three different operation modes. These modes are simplex, half-duplex, and full-duplex. These modes control the direction and simultaneousness of the data transmission. Simplex transmits data in only one direction. Half-duplex enables bidirectional communication, but only from one device at a time, not both simultaneously. Full-duplex, while being the most complex, enables the most freedom and allows for both bidirectional and simultaneous communication from both devices. For our use case, only simplex communication would be required, as there is no need to transmit data to the ADC.

The primary advantage of UART is that it does not need a clock. This makes it simple to operate, and lots of online resources are available to aid in its setup. UART also has some

disadvantages. Namely for our use case, the size of the data packets is limited to only 9 bits. This unfortunately makes it unable to send the 10 bit signal from the MCP3008 in a single packet, requiring two packets.

I2C

I2C is another hardware communication protocol similar to UART. It stands for Inter-integrated-circuit. As its name suggests, I2C is used for communication between modules and sensors on a board, rather than communication between computers and peripheral devices. To facilitate data transmission, I2C uses a shared bus to connect all the communicating devices. Each device on the bus is assigned an address to enable the device transmitting data to specify the receiving device. This shared bus architecture makes for a simplified wiring, two wires are necessary for data transmission. These two lines are the serial clock line (SCL) and serial data line acceptance port (SDA) (see Figure 28 below).

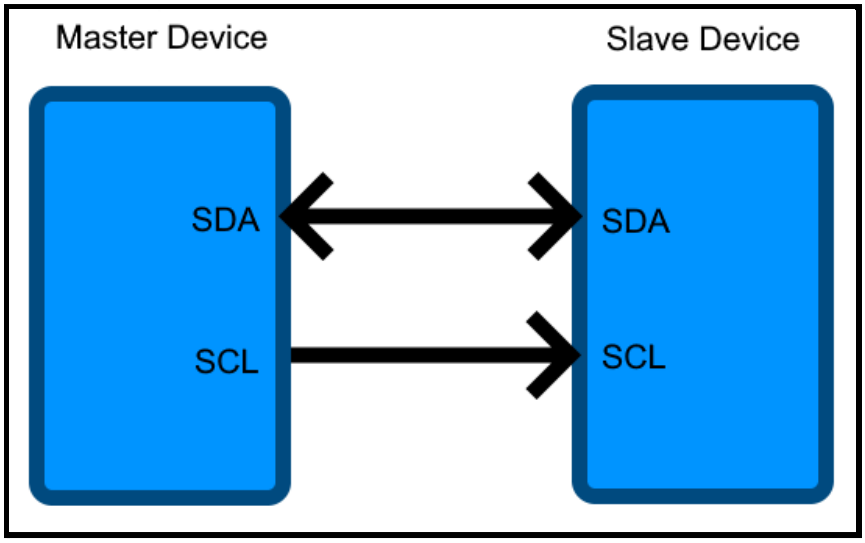

Figure 28: Example I2C Interface

Unlike UART, the SCL in I2C is used to synchronize transmissions between the communicating devices. I2C uses a master/slave system, where a single master device initiates all data transfers and generates the clock signal. Every time the master device wants to send data to a slave device, it first has to address it using its unique address. The master device is also in charge of terminating the data transfer. The relationship of the master and slave devices is flexible however, allowing for multiple master devices to allow bidirectional communication. This also removes the complexity of having different operational modes like UART has.

I2C's main advantages are a result of its simple architecture. It is very flexible in its configuration and can support numerous devices despite its low pin and signal requirements.

Its main downside is that it is generally slower than the next hardware communication protocol that was discussed, SPI.

SPI

SPI is the last of the hardware communication protocols that was compared. SPI stands for Serial Peripheral Interface. On the hardware side, SPI is the most complex of the three protocols. Like I2C, SPI uses a master/slave relationship between the communicating devices. There are four ports which the devices use to communicate in an SPI interface. The four ports are master data output, slave data input (MOSI), master data input, slave data output (MISO), a serial clock signal (SCLK), and chip select (CS). In a configuration where there is more than one slave device, each slave device has its own CS signal line. This is primarily what makes SPI more complex for the hardware than I2C or UART.

Like I2C, SPI utilizes a clock signal to synchronize data transmission. Rather than an addressing system, controlling which slave device a transmission is sent to from the master is simply a matter of sending a binary high signal on the slave device's CS line. Once the correct CS line is signaled, SPI uses two shift registers to transmit data. The data is transmitted bit-by-bit, rather than in separate packets or messages like UART and I2C, and the speed of the shift registers is controlled by the master device.

The main advantage of using SPI in our scenario is the simplicity on the software side. There exists no complex slave addressing system, which is ideal in a scenario where only one slave device exists. Having enough CS lines for slave devices is also a non-issue for this reason. SPI also happens to be the fastest of the three protocols, being able to reach data transmission speeds in the range of Mbits/s rather than kbits/s. Fortunately, the major disadvantages in the SPI protocol are present in the hardware. It uses comparatively more pins and lines than UART or I2C. However, the Raspberry Pi Zero W has support for the SPI protocol, making the hardware complexity a non-issue for our implementation. For an example of the required SPI connections to be made between the Raspberry Pi Zero W and MCP3008, see Figure 29 below. Based on the decision to use the MCP3008 ADC, SPI was used as the hardware communication interface in our implementation.

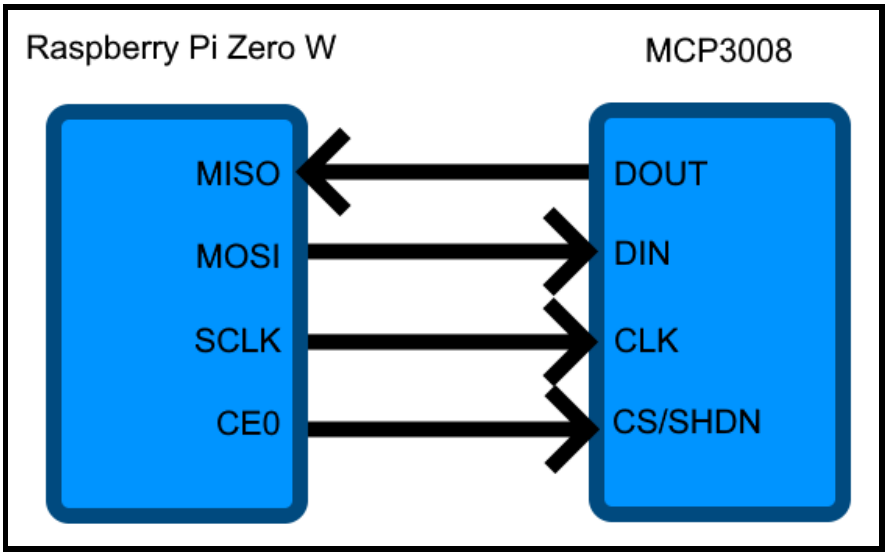

Figure 29: Example SPI Interface

Protocol	**UART**	**SPI**	**I2C**
Full Acronym	Universal Asynchronous Reception and Transmission	Serial Peripheral Interface	Inter-Integrated Circuit
Pin Configuration	Tx: transmit data Rx: receive data	MOSI: master output, slave input MISO: master input, slave output SCLK: serial clock CS: chip select	SDA: serial data SCL: serial clock
Approximate Maximum Transfer Rate	5 Mbps	60 Mbps	100 kbps
Addressing	None, only supports a one-to-one connection	CS line, one per slave device	7- to 10-bit address

Table 22: Comparison of UART, SPI, and I2C Protocols

4.28 Hardware SPI vs. Software SPI

The MCP3008 ADC requires eight connections to be made to interface with the Raspberry Pi Zero W using SPI. Of these eight, four are for power, and as such do not concern the SPI configuration. The other four connections are dependent on whether the implementation uses the Raspberry Pi Zero W's hardware SPI configuration, or the software SPI configuration.

Functionally, choosing either SPI connection method makes little difference to the resulting configuration. The primary difference between the two implementations is the configuration of the four SPI pins. Using the software SPI is the more flexible of the two options. This is because the four SPI pins required by MCP3008 ADC can be connected to any four of the Raspberry Pi Zero W's GPIO pins. Once the MCP3008's CLK, DOUT, DIN, and CS/SHDN pins are connected to four of the GPIO pins, the rest of the pin configuration is done in software. The GPIO pin connected to the CLK is designated as the SCLK pin, DOUT to MISO, DIN to MOSI, and CS/SHDN to CS.

To instead use the hardware SPI, it must first be enabled using the raspi-config tool that can be launched from the Bash CLI on the Raspberry Pi Zero W. Once the hardware SPI is enabled, the same four pins from the MCP3008 need to be connected to the correct pins on the GPIO header. This means the CLK from the MCP3008 needs to be connected to the Raspberry Pi Zero W's SCLK pin, DOUT to MISO, DIN to MOSI, and CS/SHDN to CE0. CE0 is simply the Raspberry Pi Zero's CS pin for the first SPI slave device, under a different name.

For our implementation, the software SPI was used. Any additional overhead of emulating an SPI configuration in software was insignificant, as our demands of the data transfer rate are many times smaller than SPI's capability. Setup is also easier than the hardware SPI, which is a notable advantage when our implementation is scalable to many sensors. Additionally, having a choice of which four GPIO pins to connect the MCP3008 for SPI could prove useful in a small NEMA enclosure which the Raspberry Pi Zero W was situated in.

4.29 Organization Schemas for MongoDB

With the decision to use MongoDB as the temporary database to store information on the node, there are some considerations to be made regarding the organization of the database. As most NoSQL databases are, MongoDB is relatively flexible in the way that the user decides to organize the database. That being said, it still uses a hierarchical structure. At the top of the hierarchy are collections. As the name implies, these store collections of information tied to a specific database. These collections are then further broken down into documents. Documents are where literal data are stored, using JavaScript Object Notation, or JSON. Within the documents is where the customizability of the database lies.

JSON objects that make up the content of documents are relatively simple. They are delimited at each end by a set of curly brackets. Each object then has a set of key-value pairs. Aside from being able to make these simple objects within a document, they can also be organized into arrays and matrices. The values in a JSON object can also be another JSON object, rather than just a piece of data. Being able to represent data in a matrix using JSON leaves a lot of room for optimizing the organizational scheme for the database implementation.

The data that we generate from our sensors is a type of data typically referred to as time series data. The type of information that we collect remains the same, but is collected over time, resulting in a series of data made up of numerical values in successive order. As such, the organization of the MongoDB database was based on this data format.

Document Per Event Format

With our sensor configuration, the simplest way of storing time series data in MongoDB would be to create a new document for each event. An event in this case would be the node requesting data from a sensor. Each document for each event would store the current, voltage, irradiance, and temperature data, as well as a timestamp. The timestamp is necessary to ensure the proper ordering of the documents, as there is no guarantee that the documents would stay in any particular order within their collection. The timestamps are a string that represent the year, month, day, hour, minute, and second of the data in the document. The timestamps could also be configured to have a millisecond field if a scan rate faster than one second is desired. This document format could then be expanded to collections, where each sensor would have its own collection.

This schema has the advantage of being the easiest to implement; Kiefer and Sewrathan [57] refer to it as “Mongo-naive”. It also has good write performance, yielding a significantly higher data ingest rate than the next configuration. In addition to its write performance, it also uses less disk space than the next configuration. This is relatively important, as the disk usage determines how much data the node is able to store, and how often the database stored locally on the Raspberry Pi 4’s would have to be transferred to OUC’s main SQL database. While these advantages are certainly appealing, they come with one major downside. This configuration suffers from poor querying performance, meaning that searching the database for data in a specific time period may be slower than other configurations.

Aggregate Readings Format

A popular alternative to a document per event format for time series in MongoDB is an aggregate schema. This format is meant to address the poor querying performance of a document per event schema.

Rather than creating a new document for every event, one document is created to represent a specified time period. Kiefer and Sewrathan [57] suggest what they call the "Mongo-recommended" configuration. The Mongo-recommended configuration is as follows: each device recording time series data would have a document created for each hour. The document contains a sixty by sixty matrix, for which each column represents each minute in the hour, and each row represents the seconds in each minute. Instead of creating new documents for each event, the existing document is updated by storing the data in the corresponding matrix indices. In this configuration, the aggregated document would have a timestamp which corresponds to the hour, and the minute and second of the reading would be inferred from the indices in the matrix that the data was stored at.

This configuration sacrifices some of the better aspects of the Mongo-naive configuration. It's write performance suffers somewhat, and the Mongo-naive schema outperforms it by more than 50%. Rather surprisingly, it also yields slightly higher disk usage, taking up approximately 16% more space than Mongo-naive.

While these drawbacks may seem to make Mongo-naive the obvious choice, Mongo-recommended does deliver on its promise of addressing Mongo-naive's poor querying performance. According to the querying benchmarks conducted in [57], Mongo-recommended outperformed Mongo-naive's querying performance by at least a whopping 5.5 times in every query type.

Whether to use Mongo-naive or Mongo-recommended is down to how often the database is to be queried. If queries are extremely sparse, and the database is mostly used to store information and rarely read it, then Mongo-naive could be the better option. For any database that is to be queried relatively often, Mongo-recommended is the obvious choice. In our implementation, the scan rates for the sensors are not going to be fast enough to really stress the write performance of the database. As such, the write performance advantages of the Mongo-naive schema are unnecessary, and the drastically better querying performance of Mongo-recommended would surely be appreciated by anyone tasked with analyzing the collected data. A summary of the pros and cons of Mongo-naive and Mongo-recommended can be seen in Figure 30 below.

Characteristic	**Mongo-naive**	**Mongo-recommended**
Write Performance	✔	–
Disk Usage	✔	–
Querying Performance	✘	✔

Figure 30: Comparison of Mongo-naive and Mongo-recommended Schemas

4.30 SD Card Selection for Raspberry Pi 4 and Zero W

The Raspberry Pi 4 Model B and Raspberry Pi Zero W both do not have any storage on-board. Instead, they rely on a micro SD card to store the operating system and any other data to be stored locally. This gives the end user the flexibility of deciding which kind of micro SD card best fits their use case, but there are of course some requirements to meet for compatibility.

Capacity

The first consideration to be made to select the right micro SD card for our application is capacity. The Raspberry Pi Documentation [58] states that the minimum required capacity for an installation of Raspberry Pi OS is 16 GB. This includes the full recommended software installation using the Raspberry Pi Imager. The Raspberry Pi Imager is a tool developed by the Raspberry Pi Foundation in order to make formatting an SD card with an operating system a streamlined process.

For the purposes of the Raspberry Pi Zero Ws in the sensors, no local storage is necessary for data, and so the minimum 16 GB capacity is all that is necessary. On the other hand, the Raspberry Pi 4 in the node had a database stored locally, and so it is important that the SD card is able to store a reasonable amount of data, while also keeping costs down. Luckily, unlike older Raspberry Pi models, the Raspberry Pi 4 does not have any partition size constraints. As such, the choice of capacity is just a trade off between the amount of data that can be stored on the node and the cost of the SD card. As a good middle ground that allowed some batch processing in transmitting the data from the temporary database in the node to OUC's main database, a 128 GB capacity SD card was used for the Raspberry Pi 4.

Speed

All SD cards have a class rating. While this rating measures sustained write speeds, it doesn't necessarily tell the whole story. Higher class SD cards can sometimes achieve their higher sustained write speeds at the cost of read speeds and seek times [58]. Fortunately for our use case, the read and write speeds of mostly any modern SD card did not present a bottleneck in the process of data transmission within our implementation. Because class 10 micro SD cards are readily available at relatively low prices, this is the class rating of choice for our implementation.

Operating System Installation

Installing an operating system (OS) on a micro SD card for a Raspberry Pi is a rather straightforward process. As previously mentioned, the Raspberry Pi Foundation has developed

a tool for this process; the Raspberry Pi Imager. The program works for any microSD card that is compatible with a Raspberry Pi product. All that is required is that the user selects the OS of their choice, locates the micro SD card in their file system, and then the Raspberry Pi Imager is able to configure the rest of the installation.

An important note for SD cards with capacities greater than 32 GB is that they must be formatted with the exFAT filesystem. This simply requires an extra step in the Raspberry Pi Imager, where instead of selecting an OS the user must reformat the card to the correct filesystem.

4.31 Operating System Selection for Raspberry Pi 4 and Zero W

An operating system is the lowest level of software which manages the hardware and software resources and provides basic user interface functionality. It is important for any software project that the operating system has all the necessary functionality to carry out the desired task.

Both the Raspberry Pi 4 and Raspberry Pi Zero W are flexible in terms of the operating systems that they support. In fact, one of the stand out features of the Raspberry Pi products is that the Raspberry Pi Foundation maintains an operating system specifically designed for them: Raspberry Pi OS. Raspberry Pi OS is designed to be a well-rounded OS that is approachable to new users. However, there are still compelling options that may be better suited for more narrow use cases like ours.

RISC OS

RISC OS is a freely available operating system that is compatible with any Raspberry Pi device. RISC OS is an extremely unique offering as far as most operating systems go, as it isn't Linux or Unix based like most modern operating systems. Instead, RISC OS has been under development since 1987 by the same team who designed the ARM processor. ARM is an instruction set architecture (ISA) which competes with the likes of Intel's x86. This is especially relevant to both the Raspberry Pi 4 and the Raspberry Pi Zero W as both use processors built on the ARM ISA. This makes RISC OS especially well suited to the devices, and the benefits can be seen in its smooth operation and fast boot times.

RISC OS's standalone architecture enables some unique features that are not present in other modern operating systems. Singular applications have the ability to take over the entirety of the OS's resources. This makes RISC OS attractive for embedded devices like our sensors which serve one single purpose continuously. It is also very lightweight in terms of storage, being able to fit on a 16 MB SD card. This could be useful to free up more space for data storage on the Raspberry Pi 4.

Uniqueness also tends to be the source of RISC OS's shortcomings. For one, the user interface is likely to be unfamiliar to new users, as it is unlike Windows, MacOS, or popular Linux distributions. There are a plethora of small software quirks as well. Based on user experiences on the Raspberry Pi Forums, these include USB support, 802.11 WiFi support, multitasking support, among other things.

In general, RISC OS is a fairly compelling option based on it being developed specifically for ARM devices. However, it doesn't seem that its compatibility with Raspberry Pi devices has fully matured yet, and that makes it a difficult sell for our project. Reliability is of utmost importance to ensure proper data collection, and the occasional software bugs present are concerning for this reason.

Windows IoT Core

Windows is the default OS of choice for many applications. It is the most commonly installed OS, and has been for over a decade. In an effort to keep the platform relevant, especially for programmers, Microsoft has developed a version of Windows called Windows IoT Core. Windows IoT Core is built and optimized for smaller internet of things devices such as the Raspberry Pi products. It puts an emphasis on security, connectivity, creativity, and cloud integration [60].

Windows IoT Core's headlining feature is its ability to run "headless" on a Raspberry Pi device [61]. Headless, meaning that it connects to another computer running Windows 10. The Windows 10 machine can then be used to program the Raspberry Pi device through Visual Studio Code. Though the setup is rather proprietary in that it requires a computer running Windows 10 and Visual Studio Code, it provides a compelling programming experience that places an emphasis on minimal setup overhead. The propriety of Windows IoT Core also has the advantage of the ubiquity of Windows 10, in that having a shared development platform with the deployment platform can often help to reduce the likelihood of bugs and errors in software implementations.

While Windows IoT Core makes a strong case for itself in its unique programming interface, it being closed source raises some concerns over long-term support and compatibility with other open source software projects. Unlike RISC OS, the development process for Windows IoT Core was more of a subtractive process, in that it is essentially a more stripped-down version of Windows, rather than built from the ground up for a specific purpose. The fact that Microsoft doesn't boast any performance numbers also leaves one questioning if the OS is really designed for anything more than prototyping.

Raspberry Pi OS

Raspberry Pi OS, formerly known as Raspbian, is the Raspberry Pi Foundation's officially supported operating system for all Raspberry Pi devices. It was built on Debian Linux, and so was relatively familiar to users who have experience with other popular Linux distributions such as Ubuntu. Unlike the previous two OS offerings, Raspberry Pi OS aims to be a jack-of-all-trades OS, supporting a variety of use cases. It comes pre-installed with office applications, a web browser, programming utilities, and other tools for DIY projects. It also has its own app store for easy installation and uninstallation. Because of it being specifically developed for Raspberry Pi devices, it has also seen widespread adoption. This makes finding community support easier than lesser used operating systems, much of which can be found on the Raspberry Pi Forums.

Even without features that are aimed specifically to use cases like this project, it is difficult to turn down the comfortability of using an OS specifically designed for the hardware used. Though other offerings like RISC OS may be marginally faster, for general use, and especially for our relatively light use case, Raspberry Pi OS is very unlikely to be a bottleneck for our data transmission tasks. Because of its widespread usage and extensive community support, it was used on both the Raspberry Pi Zero Ws in our sensors, as well as the Raspberry Pi 4 Model B in the node.

4.32 Data Analysis

Thus far, the design for our all-in-one photovoltaic sensor for OUC has been focused on properly collecting the desired data, and transmitting it to OUC's main database. While this encompasses the core functionality of what is necessary to meet our engineering and marketing requirements, it misses a key aspect of our true problem statement. This missing aspect is the ability to detect when a solar panel is behaving abnormally based off of the data we collect. Once the data has been properly obtained and stored, data analysis is key to be able to properly detect anomalies in the functionality of the solar panels.

There are several ways in which the voltage, current, irradiance, and temperature data that we collect could be compared in order to deduce whether a solar panel or panels are malfunctioning. Some options include comparing the numbers to a baseline that is derived from the manufacturer's specifications of the solar panel, or comparing the current numbers to a moving average that is calculated from a set time period. While these methods may seem rather straightforward, there is one key complication that must be considered when analyzing the data for dysfunctional solar panels; there are many other factors which can affect the voltage and current output of a solar panel that do not indicate any defect. Some of these include cloud cover, whether changes, and of course the day/night cycle.

Part of the solution to this problem is the temperature and irradiance readings. Having these numbers available for comparison alongside the voltage and current can help to rule out several of the confounding factors in determining whether to classify a panel as defective or not based on its voltage and current output. Just having these numbers is not enough however, what is key to properly using these numbers to identify the cause of a change in power output from a solar panel is their relationship to the solar panel's power output.

Shading and Cloud Cover

One of the most prevalent ways in which a solar panel's power output can be reduced without being defective is for the panel to be shaded. Most commonly this is due to cloud cover (soft shading), but other objects such as buildings or trees (hard shading) could cause shading as well. While shading may be obvious to the human observer who can simply see the shade being cast on the solar panels, understanding how it affects the data we are collecting is key to being able to understand when it is happening based only on the numbers we have available to us.

The first consideration that was made was that not all of the measurements would change in the case of a solar panel being shaded. Voltage output of solar panels does not change significantly with cloud cover [66]. While obstructions, which are much closer to the solar panel such as buildings and trees have a significant effect, cloud cover does not directly affect the voltage output of a solar panel. Instead, cloud cover has the effect of reducing irradiance by obstructing the sun's light rays. The lower light intensity is what is responsible for the decreased power output of solar panels when under cloud cover [67]. Rather than being reflected in the voltage output of the solar panel, it is instead the current that is reduced, causing a lower power output. This established one relationship that would help to determine a possible cause if a panel's current is to decrease. If voltage is observed to change under cloud cover, it can be said that it is due to some other factor, and not because of the cloud cover.

Temperature

Temperature has a measurable and consistent effect on the power output of a solar panel. It is for this reason that our sensor design includes the option to attach a thermocouple for temperature measurements. Though it may seem counterintuitive at first, temperature increases have a negative effect on solar panel efficiency. Fox [68] states that the photovoltaic modules that make up solar panels are tested at 25 oC (77 oF), but temperature increases above that can reduce the efficiency of the photovoltaic cells by as much as 25%. For purposes of data analysis, what is important is how this reduction in efficiency can be seen in both the voltage and current output of the solar panels. In the case of rising temperatures, current increases exponentially, but voltage decreases linearly, albeit at a much faster rate than the increase in current. The result is a net reduction in the power output of the solar panel. According to Fox [68], the linear voltage reduction is very consistent and predictable. This is important as it should have a very high correlation with the simultaneous decrease in temperature. Knowing

this relationship between temperature and voltage would allow us to identify simultaneous temperature and voltage drops as not an indicator of a defective solar panel.

Cleanliness

Another factor which can affect the power output of solar panels is their cleanliness. As solar panels tend to be installed in wide open areas, they also tend to get lots of exposure to dust and other particles which settle on the surface of the panels over time. Though the effect that dust accumulation has on a solar panel's power output is not as significant as the previous factors, it still has a measurable impact. Based on some small scale tests, [67] found that power output could be reduced by as much as 10% after accumulating 23 days worth of dust, and [69] found that cleaning solar panels could increase their power output by as much as 3.5%. It should be noted that tests in [67] were conducted in Niamey, Niger, which has a notably more arid climate than Orlando, Florida.

The main difference in the relationship that cleanliness has with the power output of a solar panel versus the previous factors is that it is gradual. As more and more dust and debris are allowed to accumulate on the surface of a solar panel, its power output would slowly decrease. If or when the solar panels are washed, they would then experience a small, but quick shift in power output. This presents a rather unique challenge to the data analysis, though the change in power output would be expected by a solar farm operator and not be a cause for any concern. Using a running average comparison would negate the possibility of gradual dust accumulation registering as a faulty panel, but the sharp rise after washing the solar panels would likely have to be accounted for in a threshold that is slightly larger than the increase in power output. This shouldn't be cause for any issues however, as Florida's climate is relatively friendly as far as dust accumulation goes, and so the increases in power output from washing solar panels should be relatively small. According to [70], dust accumulation can be classified as hard shading, and is observable as a linear decrease in the voltage output of the solar panel.

Irradiance

Lastly, and most importantly, is the effect that irradiance has on the power output of solar panels. Though there are other factors which may influence the power output of a solar panel that are not discussed here, they are mostly insignificant, or can be accounted for in the data analysis techniques. Irradiance is simply a measure of the sun's light intensity at a specific location. As such, irradiance has a direct correlation with the power output of a solar panel, as sunlight is what the panels use to generate electricity. Aside from cloud cover and other forms of soft-shadinghard, irradiance changes most during the day/night cycle. This is an easily measurable change, and should cause no problems for data analysis, as all measurements had a timestamp to determine when the data was collected. What is important however, is that the decrease in power consumption that comes from lower irradiance is only reflected in the output current, while the voltage stays constant [66]. This decrease in current due to lower

irradiance is also a linear relationship. A visual summary of all the discussed relationships can be seen in Figure 31 below.

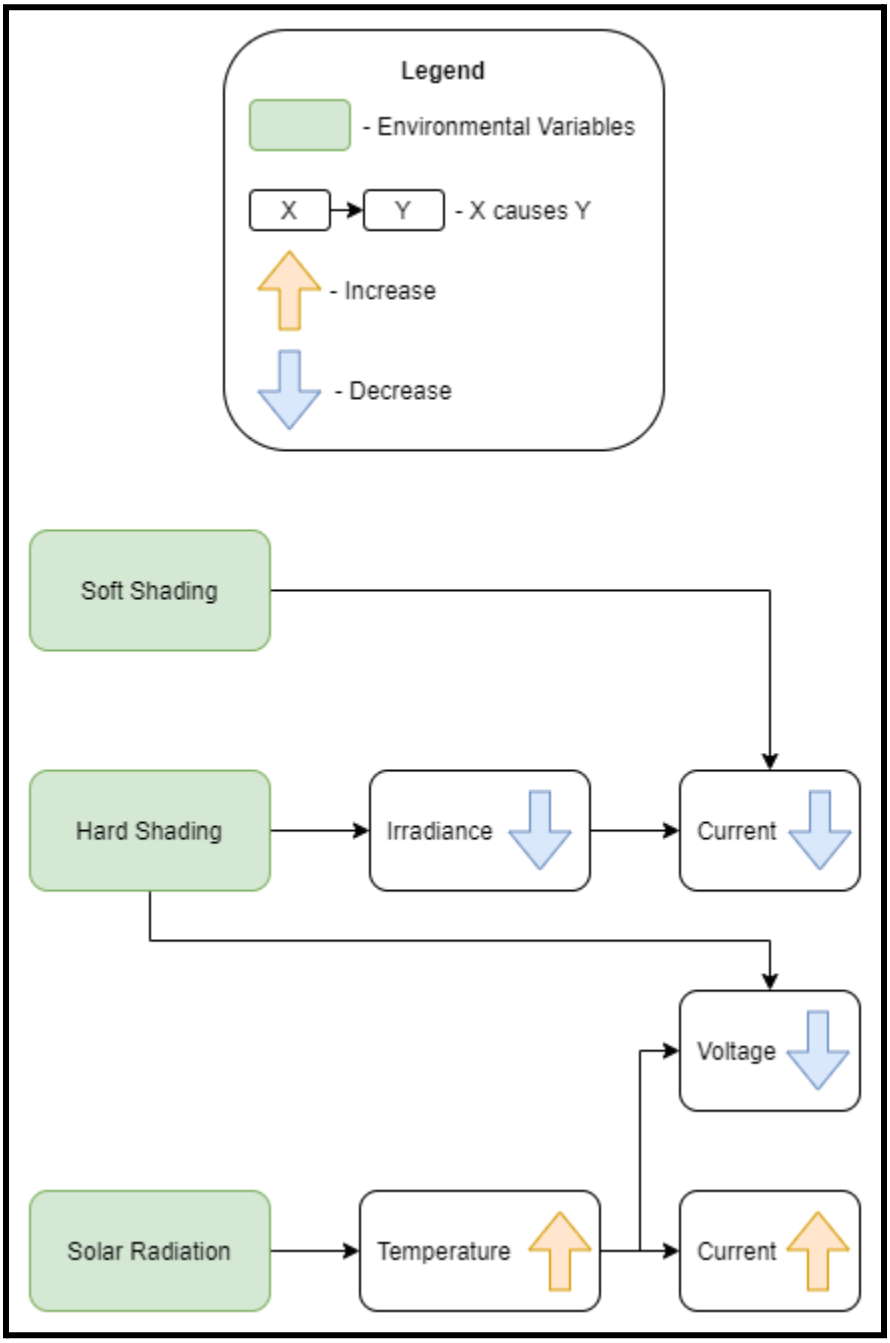

Figure 31: Solar Power Output Relationships

5. Design

To properly create our prototype, we decided it would behoove us to design and simulate schematics to avoid suffering a myriad of costly trial and error.

Rubin York provided a schematic for the PCB layout that OUC currently uses, depicted in Figure 32. He noted that while reading the voltage and current data, his readings were accompanied by a moderate amount of noise. Unfortunately, we had no ability to simulate the exact circuit Rubin used because we could not simulate the exact ACHS-7122-000E chip within LTSpice. However, the group learned that this would be a fact we would have to consider as we continue designing schematics for the device. Going forward, we explored how each of the technologies that we used were implemented in our circuit designs. Using LTSpice and Eagle, we were able to simulate some basic circuit designs, and eventually realize them in a more real and practical way (via Eagle).

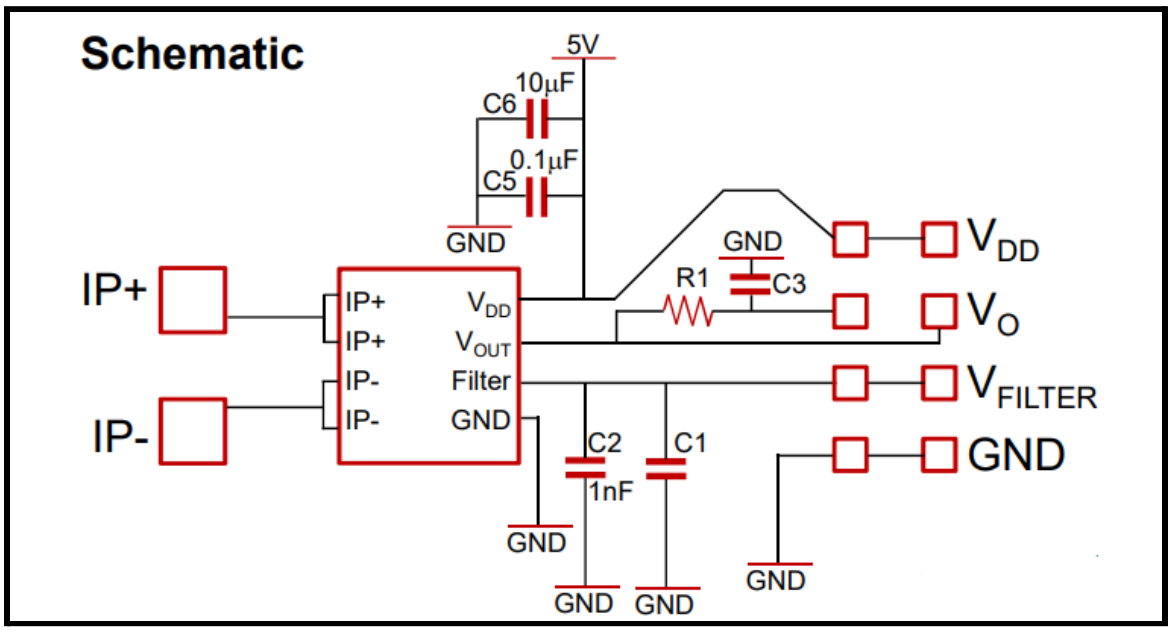

Figure 32: OUC's Current PCB Design

5.1 Conceptual Design

To maintain progress, we designed circuits that would serve as our current and voltage sensors. Instead of incorporating manufactured sensors, we decided to start with using rudimentary circuit technology. For example, we created a schematic in which the panel voltage would be measured by employing a voltage regulator and a comparator. Here, the regulator would generate some constant benchmark voltage that would be compared to the panel voltage by the

comparator. The comparator would then output a binary value, indicating whether or not the panel voltage is greater than or less than the benchmark voltage.

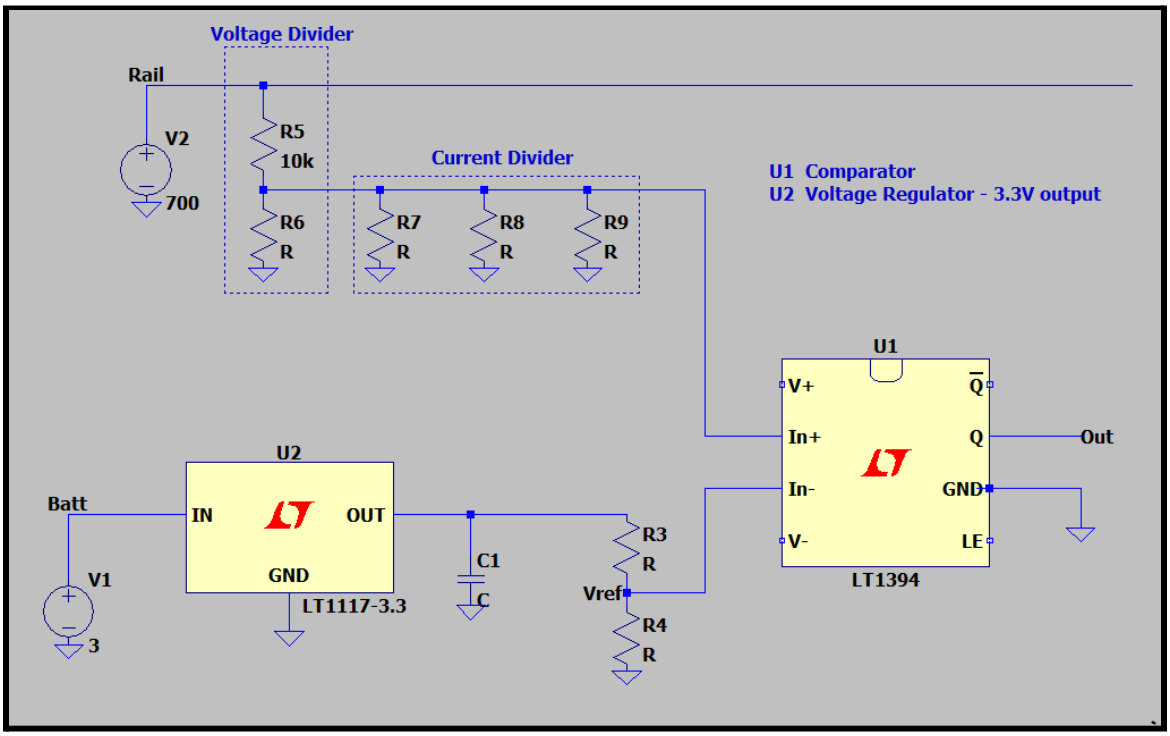

Figure 33: Voltage Sensor Design Using a Comparator and Voltage Regulator

The voltage dividers in Figure 33 serve to control the voltages, ensuring that they can remain at a magnitude in which our implemented devices can handle.

Similarly, we designed a current sensor circuit using the same logic: a voltage regulator would be manipulated into producing a constant current and a comparator would compare the constant current with the panel's current. The voltage regulator has a constant value for its R3 and its internal current source. Because of this, there is a constant voltage applied across R4, a resistor with a fixed resistance value. Therefore, there should be a resulting constant current flowing across R4.

An alternative approach to the current sensor design is to utilize the voltage sensor design through a different scope. In the design, the same direction for the voltage sensor in Figure 33 is followed. However, the idea here is to minimize R2's resistance value so that we can maximize the amount of [panel] current that flows through R2. If we expect a certain value of amps to flow from the panel, we could predict the voltage that would be applied across the fixed R2. Moreover, the predicted value of voltage can signify our desired current value. The comparator

can then compare the voltage across R2 with our benchmark voltage produced from the voltage regulator.

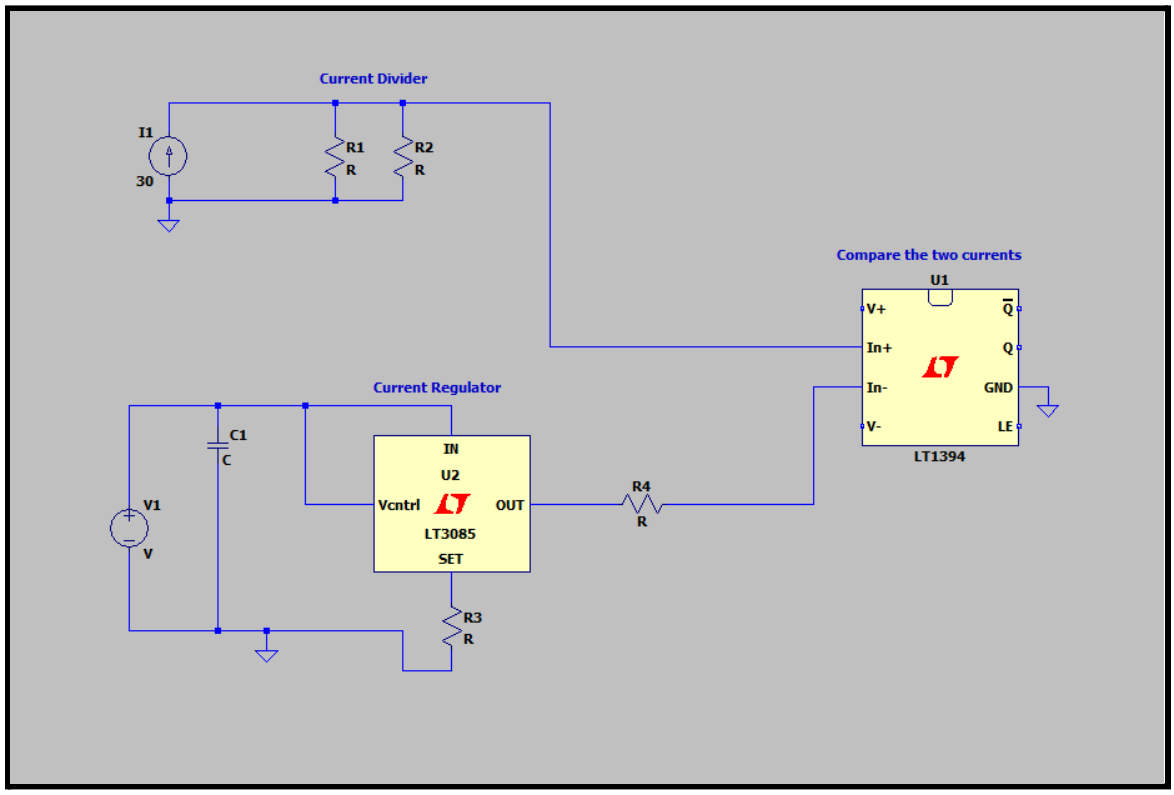

Figure 34: Current Sensor Design Using a Comparator and 'Current' Regulator

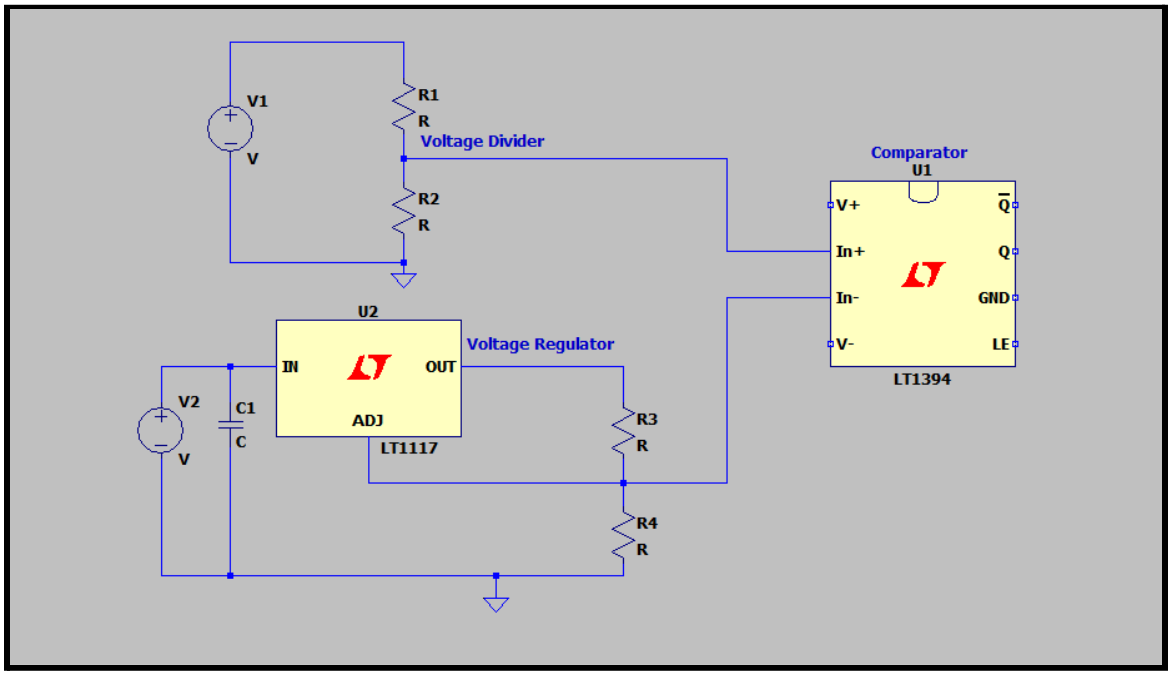

Figure 35: Current Sensor Design Using a Comparator and a Voltage Regulator

A design constraint we must keep in mind when designing our own sensor circuits is being able to handle the 32 V DC and 10 A that the solar panel produces, respectively. If these ratings are neglected, we could create a hazardous environment for the electronics involved. For instance, in Figure 34, if resistances R1 and R2 are both 10 kΩ, then the power dissipated across each resistor is about 12 W. If too much power is dissipated across a resistor, the component may heat up to dangerous levels and may be permanently damaged. This is caused by the amount of current flowing through the resistor- if the resistance increases/decreases, the current would decrease/increase. Due to Ohm's law, the power expended across the resistor will be maintained.

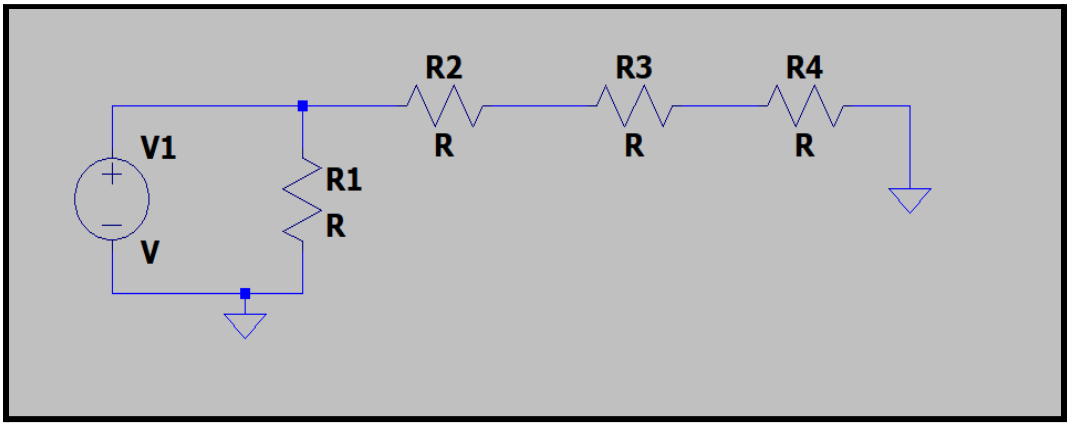

Figure 36: A Safe Method to Handle High Voltage DC

In Figure 33, we implemented a solution where we can prevent resistors from being overheated. In Figure 36, we added a resistor *in parallel* with the string of series resistors. R1 helps divert some of the current from the string, allowing for less power to be dissipated across the string of resistors. This small step was integral for our circuit designs going forward.

However, after simulating various circuit designs, we realized that we could reduce the amount of resistors. In order to do this, we would have to increase their individual resistances (or reduce their individual sizes). For example, if we were using 10 kΩ before, we could more than triple its resistance to 35 kΩ. This, of course, would not come without consequences; utilizing large value resistors can induce a large voltage across each resistor. This, in turn, would result in a large amount of power dissipation across each resistor. Fortunately, employing the aforementioned parallel resistor method helps remedy this issue.

Another issue that could arise from using large resistors is minimizing the current to levels too small for the electronics to use. The ADC, for instance, has a required minimum input value for voltage and current. If the current or voltage values get too low (drawn from the circuit), then the ADC may not be able to convert the value properly and therefore provide an inaccurate reading to the user. To rectify this situation, we turned to the use of operational amplifiers (op-amps). Op-amps amplify their input values without making any remarkable

effects in the circuit. In addition, they also prevent the loading effect that occurs in parts of the circuit. The first op-amp (shown by its connection to port 0 on the ADC) is a unity-gain buffer. A unity-gain buffer has a gain of 1, meaning that the output is identical to the input. In fact, the purpose of this amplifier is not to amplify at all. This amplifier serves as a circuit buffer. It moderates the amount of current drawn by the surrounding circuit, so that its presence does not alter the power dissipation in the circuit. The amplifier itself draws very little current, as to not disturb the original circuit. The second op-amp (shown by its connection to port 1 on the ADC) has a gain of 100. This amplification helps the ADC read the current values. This is essential to the circuit, given our use of the large resistor values.

For the ADC and op-amps to operate properly, we need to be able to supply an input voltage for these active elements. We can provide a healthy amount of power by incorporating a DC-DC converter. A DC-DC converter provides a DC voltage (Vcc) to the circuit's active elements. As mentioned earlier, we decided to implement the LMR50410X to power the circuit while radiating a negligible amount of noise.

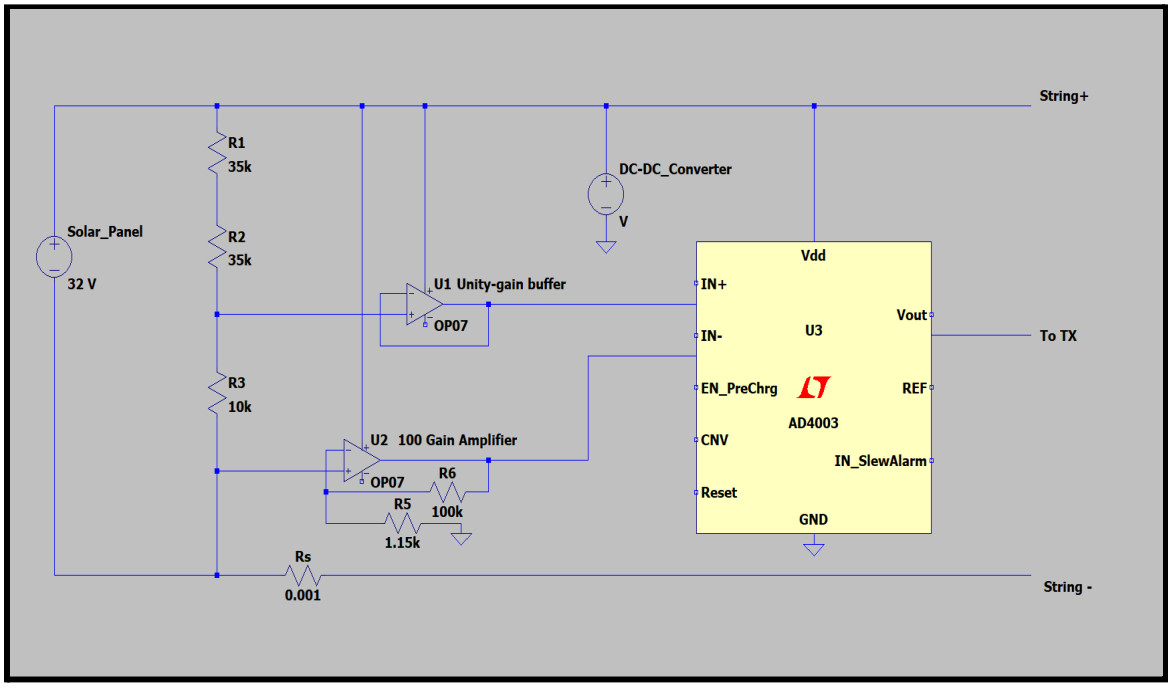

Figure 37: Current Simulation of The Sensor Circuit

In Figure 37, we had to replace the actual DC-DC converter technology with a voltage source. LTSpice did not have the LMR50410X converter, so we used a voltage source in its place as a representation. The exact ADC model or op-amp models are also not shown.

5.2 EAGLE Design

EAGLE is an electronic design automation (EDA) software that allows users to realize their printed circuit board designs. The software provides users with the ability to smoothly create and connect schematic diagrams, experiment with different component configurations, PCB routing, and also offers an expansive library of preinstalled content. In our project, we depended on Eagle to gain insight on what our printed circuit boards would look like.

Our schematic design and PCB footprints were realized in EAGLE design software. We chose EAGLE as it was familiar to us. There are many options for PCB design software, but our group had experience with EAGLE from Junior Design. All of the relevant component footprints were imported from Ultra Librarian, and then we began the design of our PCB layout. The connections are based on the above LTSpice schematics, but of course is laid out in more detail. Notable in this design is that we only used four input channels on the ADC. The others were not connected. Similarly, the IC used in the buck converter does not have all of its ports in use. The output of the converter is 3.3 VDC, a common value for voltage rails in PCB design. This was used to power the ADC and the op amps. Also notable in the design is our use of terminal blocks. They were used for multiple purposes, the first of which is connection in parallel to the solar panel. The second is in order to connect the external thermocouple and pyranometer to be transmitted along with the other measured data. These will have ramifications on the choice of enclosure, as they are much taller than the other components. Since the MC4 connectors are single strands of wire, we are using terminal blocks with only 2 poles; that is, we are using the smallest possible version of a terminal block. This ensures that we don't have excess unused terminal poles. Also, for the thermocouple and pyranometer, each of them has two input wires, so we will need four input poles. To save on costs, we are choosing to use the two-pole terminals for these as well, because we will not have to special order the four-pole blocks, and the bulk cost of the two-pole terminals means they have a lower cost-per-each.

We do not need to factor in the need to mitigate common problems seen in high-frequency designs, as we are working with DC voltage. We will therefore simply have a common ground across the board and will place the components strategically to avoid any noise that could happen in DC circuits. We may have to use thicker traces due to our higher max current.

Our limitations in testing and PCB construction are due to access to only hand-soldering tools. We do not have access to a reflow oven, so we must be careful in choosing the component sizes. And although PCB manufacturers offer component placement, their libraries do not carry our needed parts, and there is a significant upcharge in cost. Therefore we ended up hand-soldering all of our components onto the PCB, and of course we needed to hand-solder components together for testing. For resistors and capacitors, either 0603 or 0805 components were of sufficient size to both minimize PCB footprint and be hand-soldered. Anything smaller than that would require tools currently inaccessible to us. The inductor is naturally going to be

larger than either the resistors or the capacitors, so we chose one with a small footprint. Also to note, for the capacitors, there was a choice of electrolytic or ceramic capacitors. We chose ceramic due to our high power rating. Integrated circuits only have one size (per their complexity), and none of them are so complex that their footprint is sufficiently large where they reduce the efficiency of our design.

In order to connect board-to-board with the Raspberry Pi Zero W, we are using Raspberry Pi's standard 2x20 GPIO pin header. There are multiple footprint models available on EAGLE, but any will suffice so long as we designate the pins in the right orientation, and so long as the footprints we choose have the same dimensions as the parts we order.

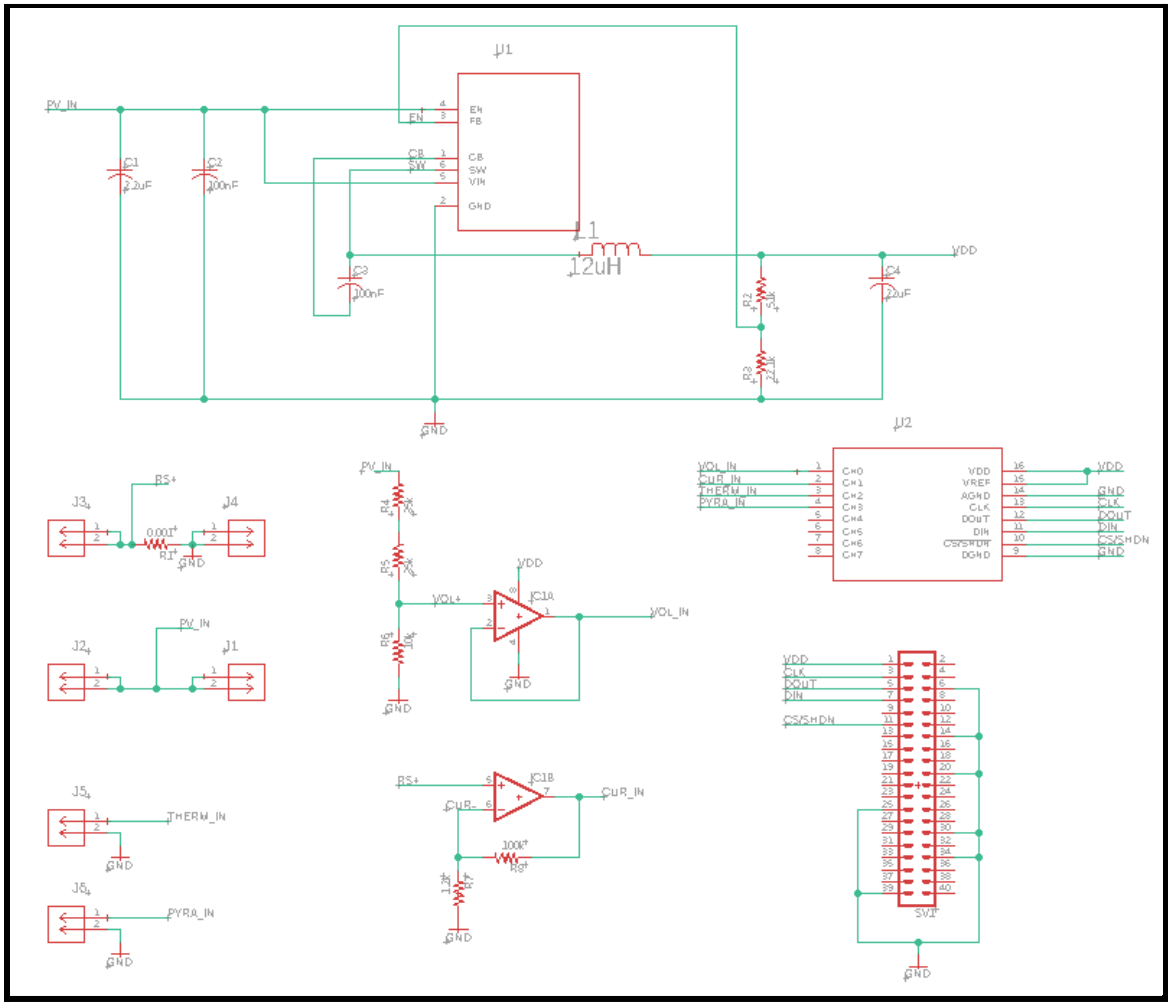

Figure 38: Final Schematic Design

Shown in Figure 38 is our final schematic design realized in EAGLE. It draws inspiration heavily from the previously discussed Texas Instruments design (TIDA-00640), yet with none of the final components actually being the same as what were used on that design. We have a few notable extra inclusions as well. Instead of the MCU they used which had an integrated

ADC, we have our own IC as a stand-alone ADC. This feeds directly into a standard Raspberry Pi 20x2 GPIO pin header that was used to interface with the Pi Zero for signal transmission. There are also 6 terminal blocks. The first two, labeled J1 and J2, are going to be the MC4 string input and panel input connectors, respectively. The second pair of blocks, J3 and J4, are the respective MC4 panel output and string output connectors. The last two, J5 and J6, are the input for the thermocouple and pyranometer, again respectively.

Since the MC4 connector was comprised of single wires, and each connector will not have two leads, we can just short the terminal blocks that each of them plug into (this applies to each block except J5 and J6, as they each have a voltage differential that needs to be measured with respect to GND). This is necessary because the smallest terminal blocks available are 2-pole, and we cannot get any smaller. We also have to be aware of the amount of space that they take up on the board. They are about the same size each as the inductor, so we will require a bit of clever positioning to make sure our board footprint is efficient, cost-effective, and of proper mechanical design so that our inputs can have enough space to plug in properly.

An important thing to note before we discuss the detailed construction of the schematic is the notion of the GND plane. Any reference to GND in this design is with respect to the perspective of the board itself, not with respect to the perspective of the overall solar panel system. That is to say, technically speaking, the GND node/plane does not mean 0 V as in most cases. It means the voltage that is on the negative side of whatever panel is currently being measured. This ensures that current is directed back into the string to minimize power loss and heat dissipation. In the case of the shunt resistor, the fact that the negative side of it is tied to GND does not mean that the current going through that flows into any actual ground. It simply, along with the other currents, flows back into the string.

Power Supply

Any PCB needs a power supply. In some cases, it is in the form of a battery, and sometimes, whether with or without a battery, an on-board power supply (regulator) is needed. As we have discussed, the power we needed was drawn from the string itself, and we do not need a battery, but our input voltage will swing drastically as the sun cycles and the solar panels reach both peak and minimum voltage. The overall health of the panels only matters when they are producing power, so that is why we don't need a battery to power the board when the string is 'off.' Again, as previously discussed, we have the LMR50410XDVBR dc/dc converter with which to work. In this section, we will discuss the overall design of the power supply.

The input terminal VIN, tied with input EN, takes the input voltage PV_IN across a range of 14 V to 32 V. The input is filtered with both a 2.2uF and 100nF capacitor in parallel. The GND terminal is shorted to the ground plane. The output FB takes the voltage across the 22.1k resistor, which, given that the output was 3.3 V, and that the voltage divider is a 51k resistor and the aforementioned one, that voltage was 0.997 V. The CB pin connects to a

capacitor, called the Cboot capacitor, that filters the noise out of the SW pin, which is the output voltage pin. The series inductor on the SW line similarly filters out noise, and the resulting output voltage is 3.3V across two other parallel filtering capacitors, Cout1 and 2, each of value 22uF. The resulting output current is 600mA.

Amplification

The amplification for our circuit is essential in getting the right proportional values passed to the ADC and then to the Pi Zero for transmission. Without knowledge of how the signals are being boosted, we won't know what corresponds to 'healthy' or 'unhealthy' panel voltages. The voltage divider of the panel input brings a maximum voltage of 32 V down to 4 V. This is passed to a unity buffer - requiring no extra resistors - to simply be passed to the ADC without fear of having the Loading Effect mess things up. The amplification of the other signal - given that it will come in as 0.01 V - needs to be significant to bring it to a readable level. We chose a gain of 85 to bring it to roughly 0.85 V. This distinguishes it from the other input voltage and puts it in the range of the ADC input ports. This is accomplished with a non-inverting amplifier with feedback resistor of value 100k and input resistor of value 1.2k.

PCB Trace/Drill Layout

The manufacturing layout shown below is generated by EAGLE. All the components are labeled for ease of soldering, and there is a clear distinction between the pads for SMD components and the holes for THT components. The 2.5mm holes on either side of the GPIO pin header and on the right side of two of the terminal blocks are for structural support. As discussed in the next section, they were the avenue by which we secure the board to the enclosure backsheet. They were also placed in those spots specifically for attachment of the Raspberry Pi Zero W. The holes on either side of the pin header are exactly the same size and placement as the ones that come drilled into the Pi Zero. This allows us the easiest way of both securing the Pi Zero to our board and the enclosure. The minimum number of labels were used for the pin header, as these are the most important connections for use with the Pi Zero.

Though faint, it is clear to see the separations between the different signal planes on the board. The space in between planes like the GND plane and the PV_IN plane are wide enough to be made by any of the printing houses we will later discuss. The ground plane was made to fit the entire surface area of the board on both the top and bottom sides as that allowed for the easiest way to ground any component that needed it. This alone got rid of many of the temporary, connecting airwires in the board design window of EAGLE. These planes were overridden in any spot that needed a plane or trace of a different signal; the size of any of the planes is wide enough to account for any kind of current spike in the system. The heat dissipation from those planes is enough that we may not need any kind of heat sink in our design.

The terminal blocks have distinct silkscreen outlines as shown on the J1-J6 components in the manufacturer's view. The orientation of these in the final build will of course matter, but for the purposes of the printing of the PCB they are less important. It was made certain that the drill holes are in the right positions for later soldering of the through hole components. The fact that they are THT components actually made things much more convenient in the PCB design. This allowed us to use the bottom side of the board both as a ground plane and as a space to route traces that would otherwise overlap on the top side of the board.

There is still considerable space left over after all of the pads are placed onto the board, and this is for several reasons. We wish to have a design that is relatively easy to hold, as there may be much work needed to be done to secure it to the enclosure, and the size of the board will affect the ease of that operation. Also, we expected that there was some room needed around the drill holes for the heads of the fasteners, and maybe some room in the enclosure for extra lengths of wire. Finally, we expect there to be a considerable amount of heat in the box regardless of placement in sun or shade, so the extra room on the board and the extra spacing between components would allow for maximum heat dissipation.

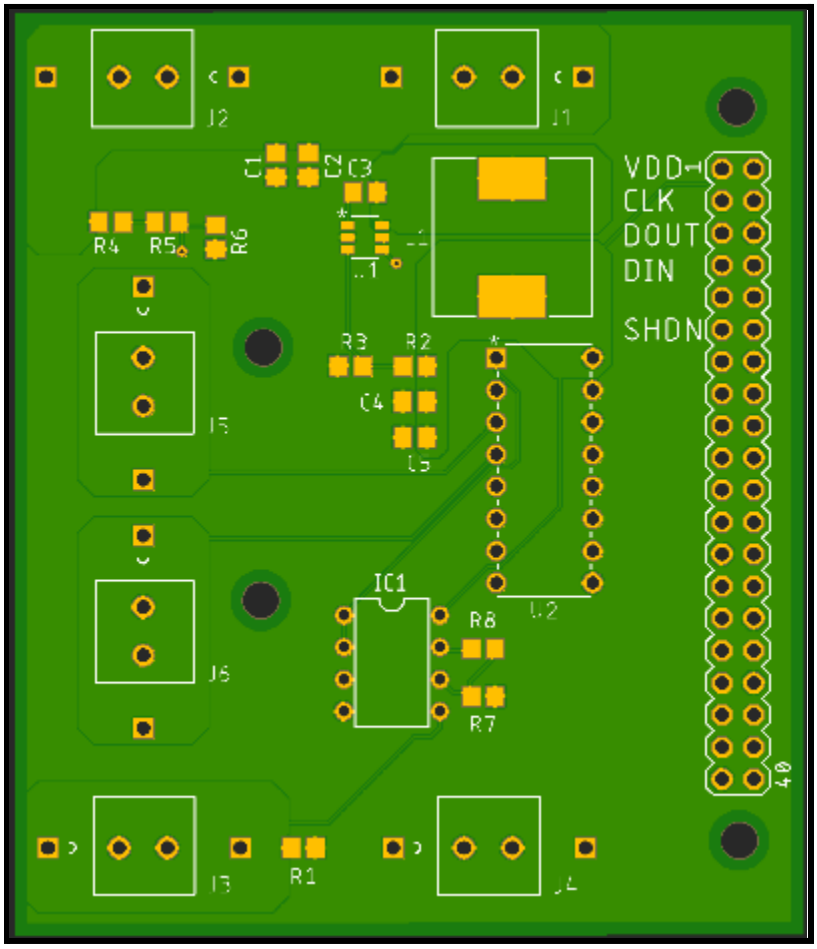

Figure 39: Manufacturer's view of PCB

5.3 Bill of Materials

Qty	Value	Device	Package	Parts	Description
1	--	MA20-2	MA20_2	SV1	Pin Header
1	--	1727010	CONN_1727010	J1, J2, J3, J4, J5, J6	Terminal Block
1	--	LMR50410YQDBVRQ1	SOT-23-6_DBV_TEX	U1	Buck Converter
1	--	MCP3008-X/P	PDIP16_300MC_MCH	U2	ADC
6	--	TL072P	DIL08	IC1	Op Amp
1	0.001	R-US_R0805	R0805	R1	Resistor
1	1.2k	R-US_R0805	R0805	R7	Resistor
1	10k	R-US_R0805	R0805	R6	Resistor
1	22.1k	R-US_R0805	R0805	R3	Resistor
2	35k	R-US_R0805	R0805	R4, R5	Resistor
1	51k	R-US_R0805	R0805	R2	Resistor
1	100k	R-US_R0805	R0805	R8	Resistor
2	100nF	C-USC0805	C0805	C2, C3	Capacitor
1	2.2uF	C-USC0805	C0805	C1	Capacitor
2	22uF	C-USC0805	C0805	C4, C5	Capacitor
1	12uH	MSS1278H-474KED	IND_MSS1278H_COC	L1	Inductor

Table 23 : Bill of Materials (per each board)

5.4 Enclosure Customization

We will also need to drill holes in the enclosure to accommodate all of the inputs and outputs of our sensor system. We expect to be having a maximum of 6 inputs/output holes needed. The weather rating can be preserved with the use of sealing glands. These are a simple and cost-effective way to weatherproof punch-throughs.

Within the enclosure, we plan on installing a backsheet. This will add an extra layer within the enclosure, allowing for ventilation and mobility amongst the devices. The backsheet can hold the actual devices by drilling them to the actual sheet. First, we would need to drill holes in the sheet to attach standoff. Then, we could attach the pi zero module and the printed circuit board to the standoffs.

The backsheet has its own screws to provide the ability to mount directly onto the polycarbonate enclosure. Ultimately, the planned stack up would look like the following:

1. Fastener [head]
2. Washer
3. PCB
4. The standoff female thread protruding the front of the backsheet
5. The standoff male thread penetrating the backsheet
6. Washer
7. Hex Nut
8. Enclosure back wall

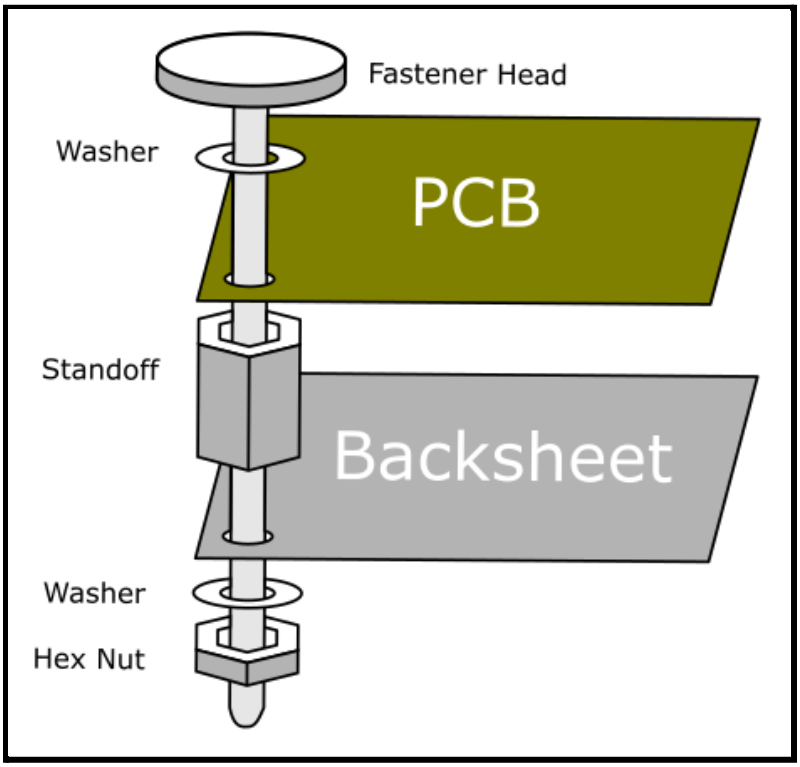

Figure 40: Board fastener expanded view

Note that it is crucial to utilize the backsheet in our project. Without the use of a backsheet, we would have to mount our printed circuit board directly onto the back surface of the enclosure. Of course, we could not practically drill through the circuit board to attach it to the wall. Instead, we would have to depend on adhering the board directly onto the enclosure via adhesives. In the engineering industry, however, this is a bad practice for the following reasons:

1. We introduce contamination via untested chemicals to the surface of the electrical components.
2. Adhering directly onto the wall subtracts surface area that can contribute to the free air flow convection, which now stops it from contributing to heat dissipation.
3. Access to the electronics within the enclosure is limited. This obstruction may prevent future technicians from maintaining or upgrading any of the devices within the enclosure.

5.5 Project Operation

Here, we will discuss the inner workings of the actual sensor device. It is important to understand how we connect all of the aforementioned components, and how they will come together and operate in harmony.

The finished prototype was installed as follows:

- Install the node (Raspberry Pi 4) to any on-site place that has a wired Ethernet connection.
- Choose the desired solar panels that need data acquisition.
- Place each prototype near its designated panel using the T-channels on the array structure.
- Plug in the sensors:
    - The first port (String+) goes to the incoming MC4 from the adjacent panel.
    - The second port (Panel+) connects to the positive side of the to-be-measured panel.
    - The third port (Panel-) connects to the negative side of the to-be-measured panel.
    - The fourth port (String-) connects to the outgoing MC4 to the next panel in the string.
    - The fifth (Thermocouple) connects to a thermocouple.
    - The last (Pyranometer) connects to the pyranometer.
- The sensors should be within a 10m range of the node.

The PCB was attached to the Pi Zero via a pin header. This will connect their signals as well as make them structurally connected. The enclosure will need to house both of these. Both the PCB and the Pi Zero will have through-holes for attachment to the enclosure. The enclosure will have a backsheet that allows for fixation of structures with bolts. The board was attached as follows:

- Put a washer on the bolt, followed by the through-hole of the PCB.
- Thread the stand-off onto the bolt.
- Push through the backsheet, and thread on a washer.
- Thread on one last piece - the hex nut - and attach the whole structure to the back of the enclosure.

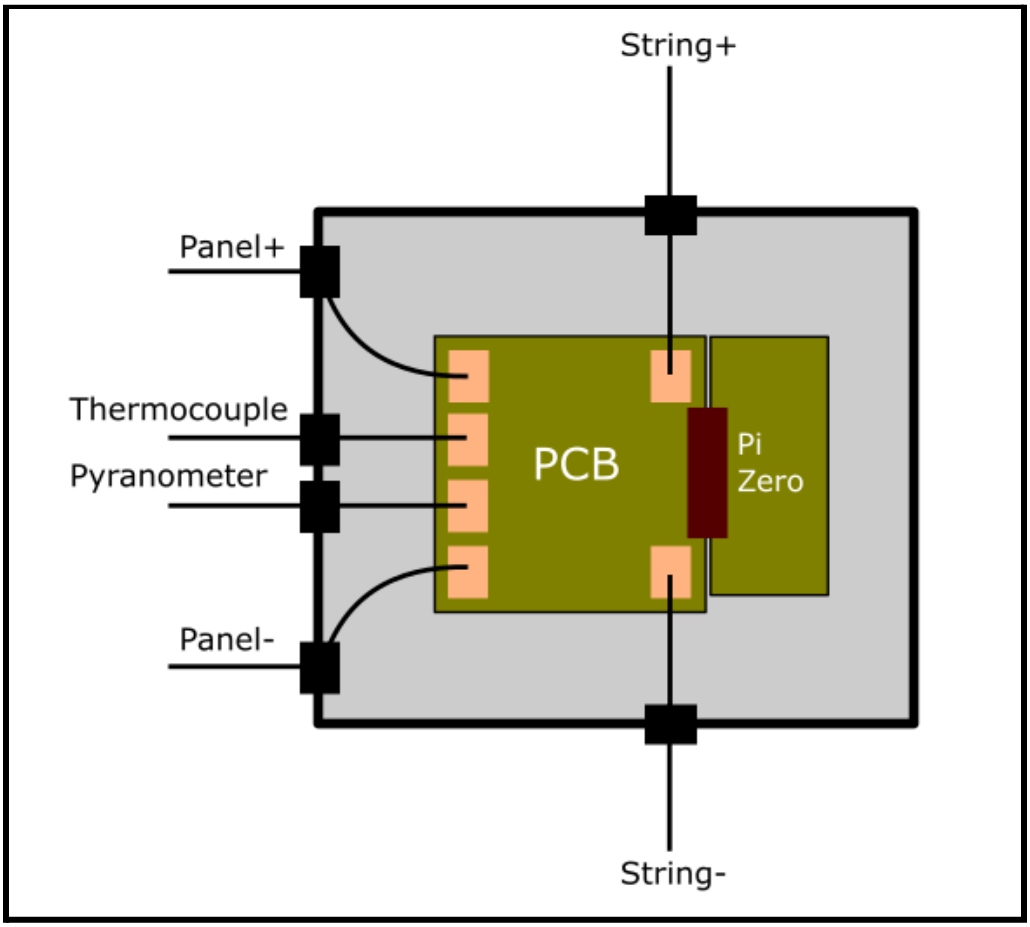

Figure 41: PCB/Enclosure Wiring Layout

Important considerations:

The space between the outer wall of the enclosure and the terminal blocks on the PCB need to be accounted for with respect to the wire gauge. Since the MC4 wires are 10-gauge as a standard, this shouldn't be a problem. The spacing, however, must be considered for reasons other than the space itself. If the space between the wall and input port is too small, the wire may end up bending in a way that will harm it over time. Given that these sensor units were implemented across utility scale projects, sustainability is of very high concern. In addition, the lack of space may even obstruct the wire so much that the wire may be prevented from being plugged in at all. Of course, if no power is reaching the sensor, then the health of the solar panel string can no longer be tracked.

The placement of the enclosure matters as well in the perspective of the array. By this we mean that the enclosures should be placed out of direct sunlight to avoid temperature spikes. The array provides much shade due to the installation angle of the panels, so the sensors should be able to be placed in the shade and thus avoid the high temperatures that would come by being out in the sun all day.

Therefore, the important part lays upon the consideration of where to place the node. The rack on which we expect to place the node lacks much shade, and is very much so exposed to the sun. Note that while our sensors are directly connected to the panels, the actual node was installed on its own unit, separate from the array. However, we do not expect for the sun exposure to be much of an issue. Moreover, the magnitude of temperature extremes were limited because the rack is doubled sided. As the sun tredges across the sky and changes position, the angle of which the sunlight is reaching the node will also change. In turn, the node won't be directly exposed to the sun's rays entirely throughout the day. Upon installation, we will attempt to aim for the side of the rack that has the most free space. Following talks with our OUC sponsor, it doesn't seem to matter on which side of the bracket to place the node [in regards to shading purposes].

6. Testing

Once we have got our circuit design in mind, we will put it to the test. First, however, we need to build the circuit. We will explore different ways to build the circuit, as different approaches have different benefits and drawbacks. After building the circuit, we will utilize an actual power supply to mimic the practical power inputs the circuit would experience in the field.

6.1 Testing Apparatus

There are several avenues by which we can go about testing the validity of our design, and its robustness with regard to our requirement specifications. The first is the most common method: breadboarding. The second is an older practice, now less popular, called 'dead bug circuitry', and the last is using protoboards. All of these were discussed before we continue with the testing procedure, as finding the best avenue of testing will ensure a higher quality in the final prototype.

Breadboarding

This method is great for testing, as it lends itself to easy modification. Any component can be plugged into any bus on the breadboard and be connected in parallel with anything else connected to that bus, and none of it requires soldering. This is a widely used testing method for this reason, but its limitation comes from the max current these boards are capable of handling. We expect our designs to have a maximum current of 10 Amps, and therefore a maximum power rating, given the maximum voltage rating of 30 V, is 300 W. The standard breadboard has a maximum power rating of 5 W. Clearly, this is not an option for us.

Dead Bug Testing

This is a circuitry method much used in the past, but is less robust than the other methods, as it is less structurally sound and can often be a less safe design, so is now less popular for permanent designs. Therefore, with the ease of construction, it makes a good testing method. The name 'dead bug' comes from the look of the integrated circuits when they are turned upside down, as is common in this method. Using some kind of untraced metallic board as ground, all of the components are connected directly with copper wire, usually. This makes for a potentially less safe design when power supplies are implemented. However, this is a good method for circuits that have too much power to be used for breadboards. The potentially heavy gauge wire used in these designs can carry a lot more current than the small buses in breadboards, so these designs are only limited by the individual components.

Due to the point-to-point nature of these circuits, it becomes increasingly more difficult to construct these circuits when the number of ICs is more than two or three. In our design, we have integrated circuits for the op amps, the ADC, and the power supply, but we should still be able to implement it well enough for testing.

We would not use this method in the final prototype, because it does not lend itself to withstanding large temperature gradients, as we will expect with the outside conditions of the solar array.

Protoboards

Protoboards are quite similar to breadboards in that they lend themselves very much to modification. They are still somewhere in-between breadboards and dead bug circuits on that spectrum, however, because removing parts requires desoldering them, rather than just removing them from a voltage bus like on a breadboard. They use thru-hole electronics as opposed to surface-mount electronics that would be seen on a PCB. Since our final prototype will use surface mount components, and since we are ordering extras of those specific parts for testing purposes, it would make no sense to use protoboards for testing.

Apparatus	**Power Limit**	**Modularity**	**Scope of Usable Parts (SMD, THT, etc.)**
Breadboard	Low	High	Moderate
'Dead Bug'	High	Low	High
Protoboard	Moderate	Moderate	High

Table 24 : Comparison of Testing Methods

6.2 Testing Procedure

Once the circuit is assembled using the chosen method, the actual testing of the design can commence. In order to make sure our design functions properly, we will test varying values of both voltage and current supplied by a DC power supply. Using a power supply gives us a known value that we can then compare with the reading given on the device. We will then be able to tune the device to make it display values in line with the ones we expect. Some error is expected regardless of how the circuit is designed due to the imperfection present in every electrical component. However, a well-placed potentiometer can be used to counteract this error, as it can be adjusted to different resistance values which will in turn slightly alter circuit

behavior. As both voltage and current are related to electrical resistance by Ohm's law, having an adjustable resistance should allow us to tune our sensor to the real values.

For our test, we will set various target voltages and currents in an effort to make sure the sensor functions properly. The expected values for voltage and current in a single commercial panel are about 32 volts and 10 amps, respectively. Therefore, we will center our test values around the typical commercial/practical values. A reasonable number of points would be five for each parameter. In turn, our spread of test values for voltage were 20, 24, 28, 30, and 32 volts, while our test values for current were 2, 4, 6, 8, and 10 amps.

Amplification

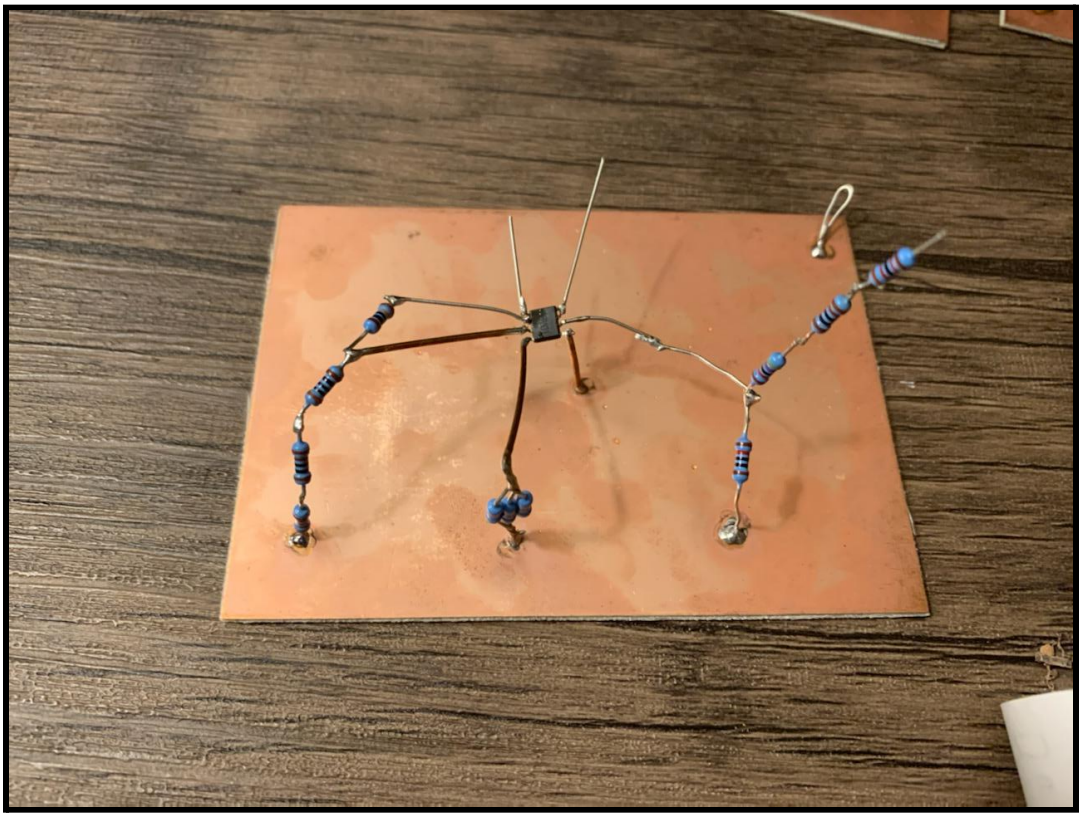

Figure 42: Op Amp Testing Setup

The circuit above is used for the testing of the amplification circuit to ensure the validity of the design. We cannot use the on-board parts for testing, as they are too small, so we obtained general-purpose resistors and capacitors for this test. The copper plate serves as a GND connection for the whole circuit, and because we have several copper plates, it allows us to easily isolate each part of the PCB for individual testing.

The testing procedure for determining the validity of the amplification circuit involves changing the supply voltage to determine the acceptable range of input voltage values. Given a maximum panel input voltage of 32 V and a nominal shunt resistor voltage of 0.01 V as test values, the rest of the testing procedure involved changing the supply voltage.

The voltage divider for the panel input voltage, at the aforementioned maximum voltage, would input a voltage of 4 V into the amplifier. As the supply voltage was increased, the voltage reached saturation at 4 V. There was of course a margin of error, and the closest we got was 3.96 V, but that is well within the working range we need. For the input voltage from the shunt resistor, given the gain of 84.3 from the op amp configuration, we should expect an output of 0.843 V.

VDD	3.3 V		4.3 V		5.3 V	
	Expected	Measured	Expected	Measured	Expected	Measured
IN+ 1	4 V	2.8 V	4 V	3.8 V	4 V	3.96 V
IN+ 2	0.84 V	2.67 V	0.84 V	3.67 V	0.84 V	4.67 V

Table 25: Test Results for the Op Amps

Power Supply

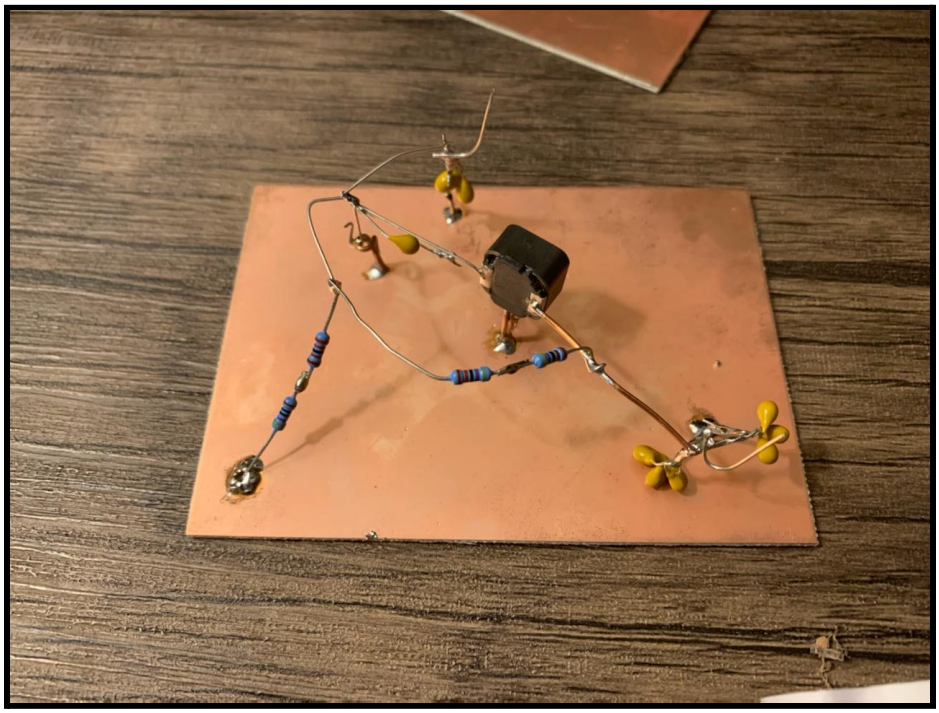

Figure 43: Power Supply Testing Setup

The circuit shown above is for the testing of the PCB power supply. This is one of the most important parts of the circuit to get right, and is also one of the easiest to validate. The criterion for successful operation are that the output current and voltage are correct and that the parts are not overheating. Since this design is from the WEBENCH online design tool, we expect the circuit to have a high level of validity and, should the part selection and soldering be sufficient, we expect this to test very well and require little or no tweaking.

The testing procedure for the power supply involved simply inputting a range of voltages to the input terminal and checking with the multimeter whether we have the desired output on the other end. Before we discuss the results of the first round of testing, we must discuss the assembly process, and why we think that affected the outcome of the testing.

The components that connect to the pins of the DC/DC converter that make up the power supply circuit are a mix of resistors, capacitors, and an inductor. The actual components to be soldered onto the PCB are, as previously discussed, 0805 packages, and are currently too small to use with our testing apparatus. They are made for surface mount applications, and not protoboards. In the place of them we will use the same type of general purpose components that we have used in other testing apparatus. The table that follows is a summary of the necessary component combinations used to get the exact values we need for the supply, as the exact values are unusual and precise in some cases. In the case of the inductor, the size of it is sufficiently large to be used in the testing apparatus.

Package Component	General Purpose Equivalent
100 nF	100 nF available
22 uF	Parallel: 2*(10 uF) + 2*(1 uF)
51 kΩ	Series: 47 kΩ + 4*(1 kΩ)
22.1 kΩ	Series: 22 kΩ + 100 Ω
12 uH	Same package as SMD

Table 26: Package Components and Equivalents

The importance of this chart is both to illustrate the need for general purpose components and to lead into the next benefit of these general purpose components, which is the power dissipation. They can each dissipate ¼ W and are therefore more capable than the SMD components for testing purposes, as we ended up testing maximum input criteria many times. Also, the parallel and series configurations listed above further help disperse the amount of heat generated by the circuit to ensure that heat dissipation isn't a problem.

The connections to the ground plane and from component-to-component are sufficiently soldered, as we had to double-check those. But because of the small size of the converter, we found it very difficult to solder the parts together, taking many hours to do so. The point of concern is the connection from the converter to ground, and this we believe is the cause of the frustrating results we got in the first round of testing.

The testing procedure started with an input of 14 V, as that is the lowest input acceptable on the range of the LMR50410X line of products. We incremented that by 2 V until we reached the maximum voltage of 32 V. For an unknown reason, though we thought the soldering we had done was sufficient and that the values of components were correct, we were getting an output of 4.36 V instead of the desired 3.3 V. This was not the only concern, though, as we believe some part of the circuit shorted because the output shot up to meet the input as we approached the maximum input value and never went back to the desired value, or even to 4.36 V. For example, we have an input of 18 V that was being met by an output of 18 V. Clearly, this is wrong, and more troubleshooting must be done in future testing sessions.

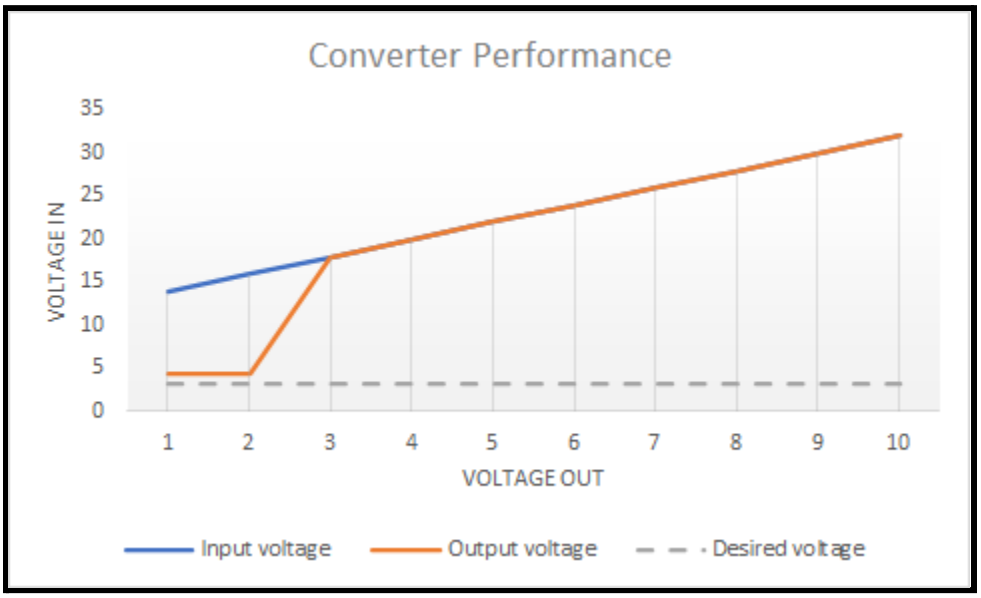

Figure 44: Converter Performance (First Round Test)

This does however leave us with another question: *Is the inaccurate output we got actually preferred?* By this, we mean that we should consider 4.3 V as an option or even just raise it to 5.0 V and use that as the power supply voltage. On the same day as the testing of the power supply, we were doing some testing of the amplification circuit, and decided to try out the 4.36 V result we were getting from the power supply as the supply voltage for the amplifier. The precision of the amplifier seemed to improve. For example, when we had 3.3 V as the power supply and were trying to send the voltage reading we got from the voltage divider through the unity gain buffer, we should've expected a value of 4 V, but got 2.8 V. When we used the rail voltage of 4.3 V, we got a buffer voltage of 3.8 V, and as discussed in the previous section, when

we used 5 V as the rail voltage, our buffer rounded out at 3.9 V, clearly close to the desired value.

This clearly shows an example of some part of our design that can be changed for the better. The choice of using 3.3 as rail voltage came from the first Design Example, as the prototype they used is structurally similar to ours and they used 3.3 V as the rail voltage. We think this could be another point where we differ from the test department at Texas Instruments, as we are not using all the same packages and will clearly have some divergence from their results, but this is not something we discovered in design, rather in testing.

ADC

The testing procedure for the Analog to Digital Converter (ADC) necessitates a few assumptions, namely that we assume the inputs to the ADC are constant and correct. This allows us to troubleshoot the ADC without worry that issues would arise from faulty design of any other aspect of the PCB. The inputs come from multiple sources. We know that both the pyranometer and the thermocouple from OUC are functional, as neither have been used before and were given to us in manufacturer's packaging. For the other two inputs, the voltage reading and current reading, we are assuming the amplification circuit works and are simulating those inputs using the Rohde & Schwarz NGE-100 DC power supply. As long as the inputs are within the readable range of the ADC, their instantaneous value is arbitrary.

Per the datasheet of the MCP-3008 ADC, the maximum rail (VDD) value is 7.0 V. The input ranges rely on that voltage, and are from -0.6 V to VDD + 0.6 V. We are using several values of VDD to check the functionality of the ADC. First, we will use 3.3 V and then 5 V as supply voltages. The table below clearly shows the effect changing VDD has on the allowed input voltages.

	VDD	Input Range
	3.3 V	-0.6 V to 3.9 V
	5.0 V	-0.6 V to 5.6 V
Max. Ratings	7.0 V	-0.6 V to 7.6 V

Table 27: ADC Input Voltage Ranges

Another crucial factor in this portion of testing is that we are able to read the outputs of the ADC. This requires the Raspberry Pi Zero W, so we connected it to an HDMI monitor and ran the *simpletest.py* program from Adafruit to read the output of the ADC. Since the MCP-3008 is from Adafruit as well, there is built-in compatibility between the ADC and the

Pi Zero, and the Python program from Adafruit is known to work, so that further lessens the number of things we have to keep track of that may present problems in testing.

We know that our design requires relatively high power at times, but the ADC will never have much current going into the device. It does in fact simply convert some voltages from analog to digital, and since those voltages come from operational amplifiers, a thermocouple, and a pyranometer, we will no expect the ADC to be taking much current at all, so for the purpose of testing we can leave the current value on the DC supply at minimum value to reduce the amount of power flowing through the circuit. This means we ended up using less than the 5 W rating of a typical breadboard, so we ended up using the breadboards we have on hand to ease the testing procedure. The benefit this has over the 'dead bug' method is two-fold: the MCP-3008 fits snugly into the holes of a typical breadboard, and the jumper wires we use are temporary as opposed to permanently soldered jumper wires, so the degree of modularity is significantly greater.

This test does not need any components beside the ADC, the Pi Zero, and jumper wires. As shown in the photo below, we have a color-coordinated approach to connecting the ADC and Pi Zero.

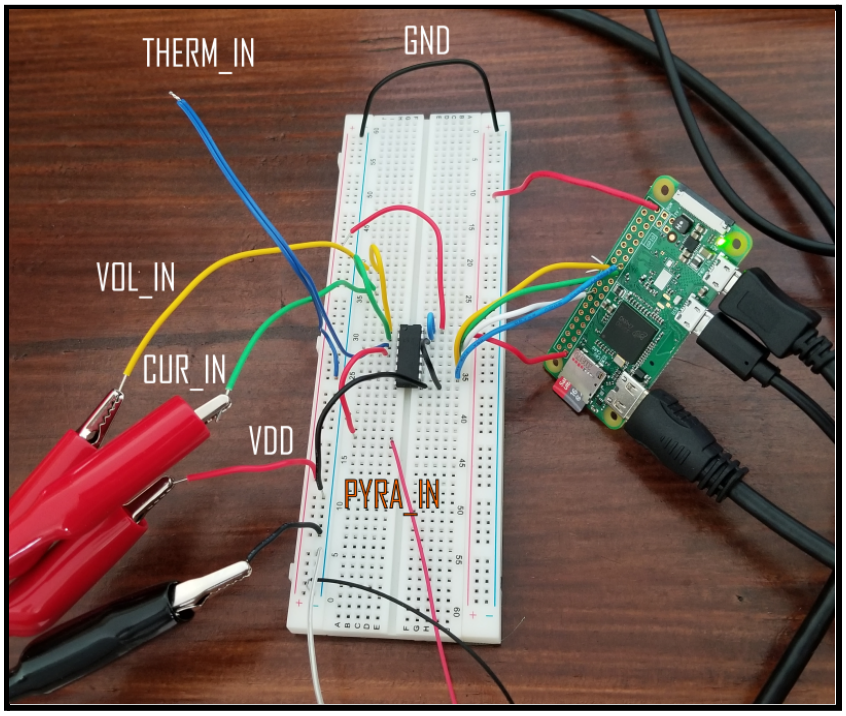

Figure 45: ADC Testing Setup

The Python script reads the output of the ADC a few times every second, so at first it displays all 0s. We noticed as well that when the Pi Zero was being moved around and the script was running, at some point a few of the pins on the Pi Zero shorted and the reading on the script

spiked to its max value of 1023 for a fraction of a second. This confirmed that the script was running properly and that it could read values from the ADC.

However, we could not read any other values from the ADC. In light of this, we began troubleshooting down the chain of command. First, we took a closer look at the status of the power supply. We made sure it was connected properly, and even tried restarting the operating system. We then set the voltage to 3V to power the ADC. From there, we used a multimeter to measure the output at the power supply leads. Fortunately, we obtained accurate output voltage values from the leads, confirming that the problem did not exist inside the power supply. Considering this is a power supply borrowed from UCF, we would not strongly suspect that this be the issue; the product is from a reputable company that UCF collaborates with often.

The next element in the chain of command would be the wires and breadboard configuration that carried the power from the power supply to the ADC. Using the multimeter, we measured the voltage at the input voltage Vdd pin of the ADC with respect to ground. We were able to obtain the same accurate readings at this pin, which once again reveals that there is no problem with the breadboard circuit. We made sure to rewire the breadboard multiple times, closely checking to make sure that the wires were connected to the correct busses. In addition, breadboards rarely break down without extreme physical trauma or carelessness regarding the power supply.

After confirming that power was reaching the ADC, the next possible area where the problem could exist would be the ADC itself. Specifically, the issue may be that the ADC is not translating and/or transmitting the analog voltage values properly. The first step we took was to make sure the ADC itself was functioning properly. We easily tested this by swapping the first ADC with another. However, our efforts were futile. The next step we took was to study the ADC's connection to the Raspberry Pi Zero. Upon looking at the manual, we had several episodes of doubt [and false realization] as to how the device should properly be wired. In light of the confusion upstream, we decided to continue downstream and make sure that the Raspberry Pi Zero's code was free of errors. After reading line by line and consulting with the manufacturer, it was agreed upon that the code was written correctly. In Figure 46, you can see the visualization of our testing troubleshooting.

In conclusion, we believe that the problem exists with how the Raspberry Pi Zero should be connected to the ADC. After checking all of the other areas involved that would influence the transmission of data, we believe that it could be the connection configuration of the pins. Note that, if true, this problem would be much more fortunate than if the other parts of the equipment were not working. For instance, obtaining another power supply would be an arduous process that would involve time-consuming communication between our group and UCF. In addition, purchasing another breadboard, while simple, would increase the costs of our experimentation. Going forward, if rewiring the connection between the ADC and the

Raspberry Pi Zero does not work for the hundredth time, we will try to consult our sponsor from OUC. Rubin York has had some experience with working with Raspberry Pis and may be able to diagnose the problem. If Rubin is unable to provide any feedback, communicating with Mouser (ADC's manufacturer) and further online research will hopefully hold the key.

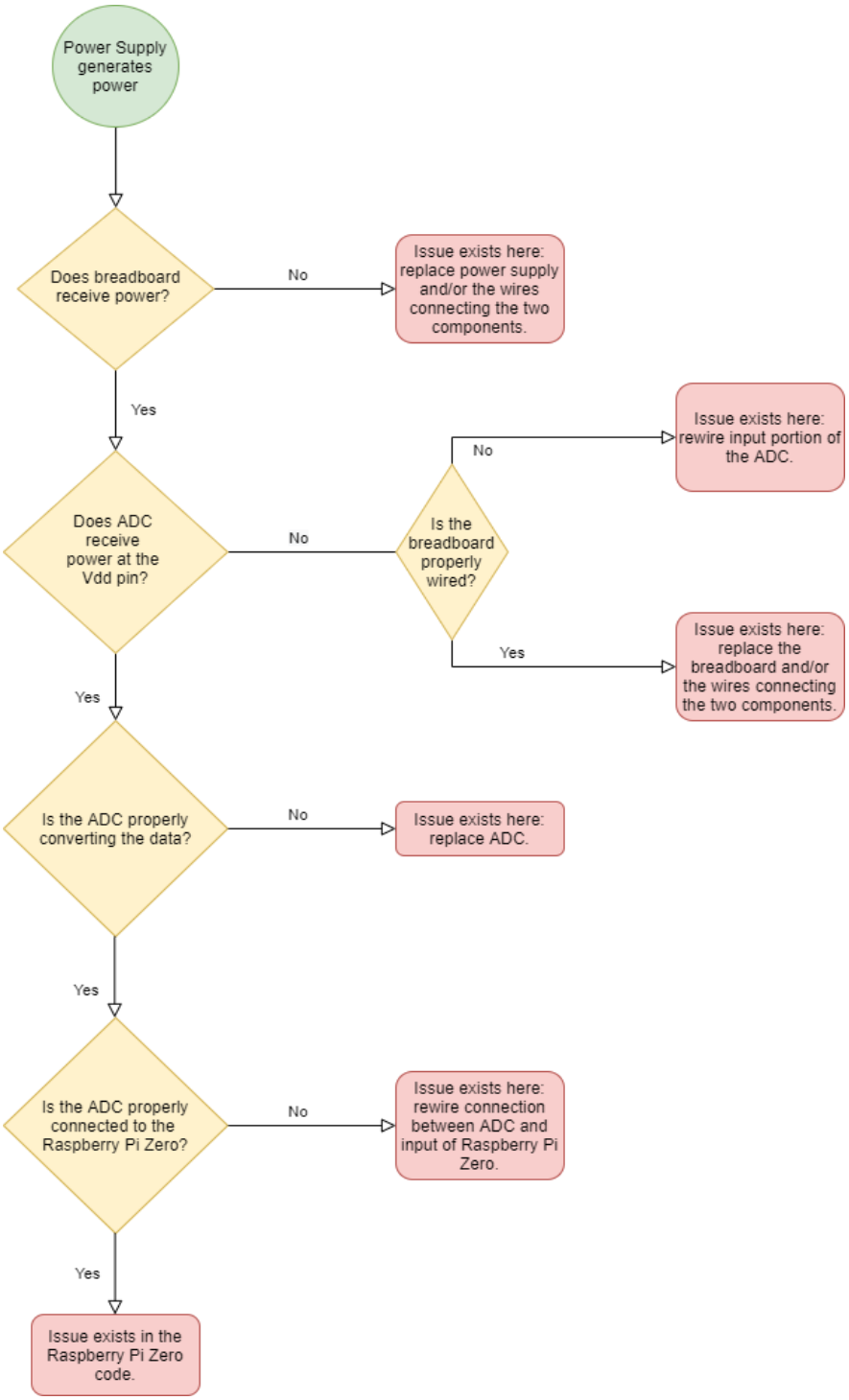

Figure 46: Troubleshooting our testing

After many attempts at rewiring the ADC to the Pi Zero, we could not get the values being read on the ADC to be transferred to the Pi Zero. As a recap, here is a summary of what we know works and what could be causing the problem:

Component	Description	Status
ADC	From Adafruit; not used before. Multiple copies.	✔
Pi Zero	From Adafruit, not used before. Multiple copies.	✔
Python Script	From Adafruit. We know it works.	✔
Power Supply	New model loaned by UCF. Worked previously.	✔
Breadboard	Used before, but we know it hasn't been broken.	✔
SPI Configuration	From Adafruit. Tried both software and hardware.	✔
Pin Connection	There is confusion here. Possibly the issue.	?

Table 28: Comparison of testing components' relation to the issue

From this table, it is clear that one of the components could present a problem. There was confusion in the pin connection, as we had allocated it slightly differently from the article that we are now using to facilitate testing. Also, the pin header had been oriented upside down, as the reference photo we used for the Pi Zero pin connection was backwards. We will fix this on the final design, but for now that isn't so much of a concern, and was something we were able to fix in testing.

The pin connections from the ADC to the Pi Zero are one of the more configurable components. As we are using the software SPI configuration for the Pi Zero, the SPI pins CLK, MOSI, MISO, and CS are all configurable with a few lines of code in the Python testing script. As previously mentioned, the configuration in which our schematics show the pin header on the PCB that is meant to connect to the Pi Zero is in the opposite orientation as most pinout diagrams for the Pi Zero. This is to eliminate the possibility of interference with any other components on the PCB, since the Pi Zero would be raised above the PCB if it were in the more common orientation. However, this had the downside of causing some confusion with the pin configuration. Once this was discovered, we readjusted the connections from the ADC to the Pi Zero to match the existing software configuration in the test script provided by Adafruit.

To also eliminate the possibility of the software SPI configuration causing any issues, we tried running the same test using the hardware SPI configuration. Unfortunately the result remained the same, but at least eliminated the SPI configuration as a possible issue. As the

hardware SPI configuration requires different pin connections to the Pi Zero than the software SPI configuration, we elected to leave them in place when returning to the software SPI configuration. Because the SPI pins are configurable in the software, we reconfigured it to use the existing connections for the software SPI. This also did not fix our problem, but reduced the possibility of any faulty pin connections causing the problem.

6.3 Testing Facilities and Equipment

We sourced the needed supplies from our own connections and from the Senior Design Lab at UCF. The power supply is a Rohde & Schwarz NGE-100 power supply on loan from the UCF Senior Design Lab. The Soldering irons and multimeter are from our own supplies, as well as the general-purpose resistors and capacitors, and the other miscellaneous tools needed for successful testing.

Due to the high demand for lab space and the capacity restrictions currently in place from COVID-19 recommendations, we have decided to test off-campus. We are using at-home table space for our testing purposes.

Equipment	Model	Source
Power Supply	R&S NGE-100	UCF Senior Design Lab
Multimeter	RadioShack 22-812	Own supply
Soldering Iron	--	Own supply
Hand Tools	Stanley (general tools)	Own supply

Table 29 : List of Testing Equipment

6.4 Prototype Testing

After the next iteration of the device using the printed circuit board had been ordered and assembled, it was time to perform tests using the actual solar panel array. At this point the device was attached to a single solar panel using MC4 connection and output data was compared to the expected values for the panel. This is part of the plan for Senior Design 2, but we are carefully analyzing the parallels between what we are testing indoors and what we expect to test outdoors.

7. Administration

7.1 Senior Design I Milestones

Milestone	Tasked to:	Start Date	End Date	Status
Introductions				
Familiarize with the project	Group 5	1/11/2021	1/29/2021	Completed
Role Assignments	Group 5	1/11/2021	1/29/2021	Completed
Identify Parts	Group 5	1/18/2021	1/29/2021	Completed
Project Reports				
Initial Divide-and-Conquer	Group 5	1/11/2021	1/29/2021	Completed
Divide-and-Conquer 2.0	Group 5	1/29/2021	2/12/2021	Completed
Final Report (1st Draft)	Group 5	2/12/2021	4/2/2021	Completed
Final Report (2nd Draft)	Group 5	4/2/2021	4/16/2021	Completed
Final Report (Final)	Group 5	4/16/2021	4/27/2021	Pending
Research and Investigation				
PCB Configuration	Julian	1/29/2021	3/31/2021	In Progress
Processor	Zoran	1/29/2021	3/31/2021	In Progress
Filter Circuit	Christian	1/29/2021	3/30/2021	Completed
Communication Ability (wireless/wired)	Zoran	1/29/2021	3/30/2021	In Progress
Pyranometer	OUC	1/29/2021	3/31/2021	Completed
Thermocouple	OUC	1/29/2021	3/31/2021	Completed
Voltage Sensor	Julian	1/29/2021	3/30/2021	Completed
Current Sensor	Julian	1/29/2021	3/30/2021	In Progress
Enclosure Design	Ryan	1/29/2021	3/31/2021	In Progress

Table 30: The tentative timeline for our project during Senior Design 1.

The Senior Design 1 milestones are split into two groups: temporary and permanent. By this we mean that we have many milestones with 'soft boundaries' that can be changed as our workflow determines that they might need to change, but that some are determined by UCF and are immutable. These are the Project Report milestones, and require a submission before each deadline.

The moveable milestones are further categorized as follows: Introduction and Research. We used 'Research' as an all-encompassing term as we are not required to produce any physical prototype in Senior Design 1. These Research milestones are tied directly to the labor distributions assigned in the Hardware Block Diagram and are in themselves changeable. We allow for ourselves to have a lot of room to maneuver as we flesh out the tasks according to everyone's ability.

7.2 Senior Design II Milestones

Milestone	Tasked to:	Start Date	End Date	Status
Construct First Prototypes	Group 5	4/27/2021	5/30/2021	In Progress
Tinker	Group 5	5/30/2021	6/30/2021	Pending
Finish Indoor Testing	Group 5	--	6/15/2021	Pending
Field Test	Group 5	6/16/2021	6/30/2021	Pending
Establish Final Product	Group 5	6/30/2021	TBD	Pending
Final Peer Presentation	Group 5	TBD	TBD	Pending
Final Report	Group 5	TBD	TBD	Pending
Final Presentation	Group 5	TBD	TBD	Pending

Table 31: The tentative timeline for our project during Senior Design 2.

The Senior Design 2 milestones are less defined as we are not completely sure what will define those weeks of work, but we do know the outline per UCF's instruction. We expect to move these deadlines up a bit as we are ahead of schedule in the Senior Design 1 milestones.

After talks with Rubin York, we have set a milestone of field testing with him to be completed in the second half of June. We have already visited the site and have gathered preliminary information, but we will expect to have a near-completion prototype to show to Rubin. At that time we will decide how to conduct the final presentation as it will have to be done outside.

7.3 Project Budget

The budget has become one of the more flexible aspects of our design process, as we are sponsored by OUC and do not have to consider out-of-pocket costs. That being said, every time we have come across an obstacle that would increase the price per unit, we have discussed it with Rubin and determined the viability of each step.

If we consider the projected final cost to be accurate, then the cost per unit will be $147.00, which is much higher than the original specification of $20.00. However, along the way we have seen the cost of several of the line items below come out to be far less than expected, so the price of $441.00 total is what we expect to be the ceiling cost. In fact, we expect to come in under this significantly.

Component	Quantity	Cost per Each	Estimated Cost
NEMA 4X Enclosure	4	$71.00	$284.00
Pyranometer	3	$223.00	Provided by OUC
T-type Thermocouple	3	TBD	Provided by OUC
PCB	3	Varies	$7.00
Wireless TX/RX Module	3	$20.00	$60.00
Node	1	$40.00	$40.00
Extra Hardware	Varies	Varies	$40.00
ICs	3	Varies	$10.00
Total	3 prototypes		Approx. $441.00

Table 32: The proposed budget for our prototype solution.

The enclosures were the bulk of our expense, if we go by the proposed cost on McMaster-Carr. However, we know that other options are available, and ended up inspecting those options for cost effectiveness in Senior Design 2. The PCB cost comes from JLCPCB's website, and accounts for extra cost of which we may not currently know. The costs for the node and TX/RX modules comes from Adafruit's website, and includes shipping. Extra hardware includes a buffer for any special tools or materials we might need, and the IC line item gives us room to consider several ICs as they tend to be a costly part of PCB design.

PCB Vendor

When selecting a PCB printing house, there are three cardinal characteristics that we must search for in a company's products:

- A reputable standing in the power engineering industry
- High quality products that will stand the test of time, along with handling utility-scale power
- Affordability as to maintain replication

The three printing houses we will consider are JLCPCB, OSH Park, and Custom Circuit Boards. Each of the options belong to a long history of producing quality boards in reasonable times and for reasonable costs. Each of the vendors provide common specifications, such as multiple layers and 1-ounce copper traces. Each of these vendors can also offer a maximum size well above 3"x 3", which is the maximum size we could expect for our design.

The unique capabilities of the vendors are compared as follows:

	JLCPCB	OSH Park	Custom Circuit Boards
Min. Drill Hole Size	0.2 mm	0.254 mm	0.152 mm
Board Edge Keepout	0.2 mm	0.381 mm	--
Min. Annular Ring	0.13 mm	0.127 mm	--
Min. Trace Width	0.127 mm	0.152 mm	0.076 mm
Min. Trace Spacing	0.127 mm	0.152 mm	0.076 mm

Table 33: The prospective vendors' capabilities.

Vendor Cost

For each vendor, the cost can be calculated (as an estimation) by uploading our current gerber files to their websites. For JLCPCB, using the stock parameters, the cost per bundle of 5 boards is $2, plus $3.80 for shipping. There are many other cosmetic options that raise the price substantially. For OSH Park, their prototype option gives three copies of a board for $5/square inch. After uploading the gerber files, the cost comes out to $35 for three copies. For CCB, as they are a much more bespoke option, the cost is substantially higher. For 5 boards, the cost is $79.00 per board, so $395 total. Another option is $401 for 15 boards. Clearly, this is

out of our range. It seems that CCB is geared toward mass production, and prototypes are not the main product of their facility.

Printing House	No. of Boards	Total Cost
JLCPCB	5	$5.80
OSH Park	3	$35.65
Custom Circuit Boards	5	$395.00

Table 34: Print House Cost Comparison

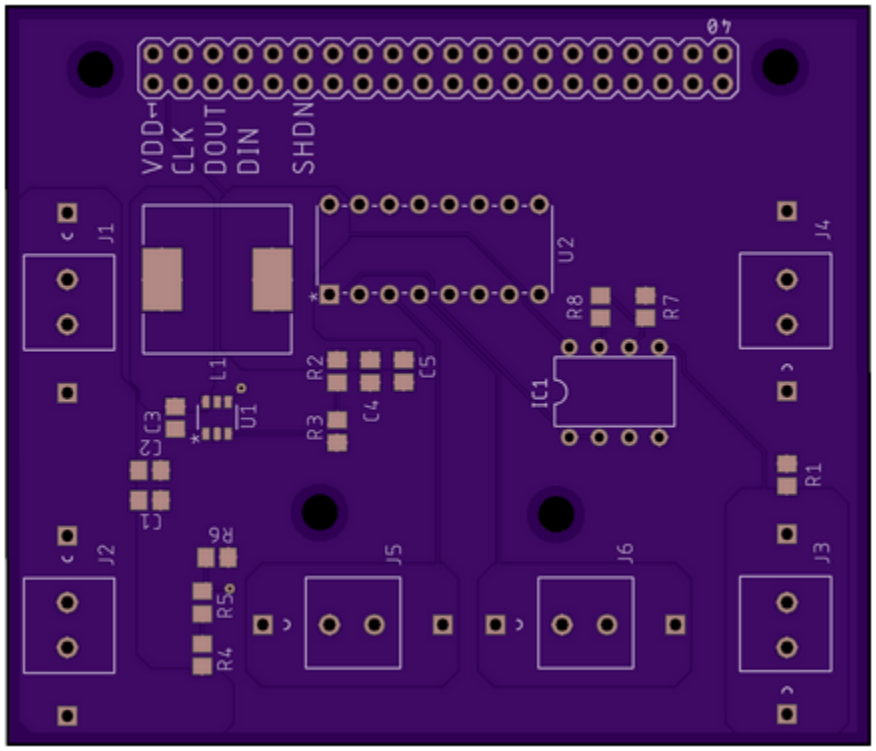

Figure 47: PCB generation from OSH Park

8. Conclusion

In today's age, renewable energy is on the rise. All around the world, scientists and engineers are working tirelessly to not only make renewables more efficient, but also make them more easy/practical to use. The purpose of this project is to contribute to that mission, by making the dependence on photovoltaics much more practical. We intend on doing so by manufacturing a sensor that will aid technicians in determining the relative location of a faulty panel in a string of solar panels. This way, technicians at utility companies can minimize their response time and swiftly repair/replace any faulty panel that is hindering the farm's potential power output.

So far, we have conducted a myriad of research as to what parts we will need and why. For this project, we need to reserve the ability to read the voltage, current, temperature, and irradiance of the respective panel, while maintaining a minimal presence in the solar power string circuit. Therefore, we have reviewed many active devices (voltage regulators, DC-DC converters, ADC) that consume low amounts of power. We explored the field of unique devices such as pyranometers, thermocouples, and heat sinks, and determined the most effective model to use. We also reviewed a variety of software interfaces, and decided that software SPI would be the most appropriate and the most effective for our project.

Early on in the project, the group realized that we would not be able to simulate every portion of the circuit accurately. To reiterate, LTSpice's library does not contain every viable device that we can implement in our project. Therefore, we had to try either using similar parts for those sections, or just depending on actual experimentation during testing. Going forward, this method will continue to be used; in the case where a portion of the circuit does not work as expected, we may have to consider redesigning. However, we do not believe that this will happen. In fact, we were able to modify the physical testing on the fly via physical calculations.

In Senior Design 2, we remained steadfast on prototype testing. We plan on working tirelessly to ensure that we can prove that the ADC can receive, translate, and transmit data to the Raspberry Pi Zero. Then, once we have confirmed that the prototype can function properly connected to a power supply, we can move forward towards implementing it in the field. Rubin York from OUC has assured us that we can arrange to install the device on OUC's research array in their facility. During field testing, we will have to keep an eye out for a variety of factors, such as overheating (due to ambient temperatures), burning out (due to the power of the solar string), and feasible installation. Note that we will have to make some sort of installation within the enclosure before we implement the sensor in the field. Considering the sensor will still be in its testing stages, we may consider making some sort of provisional installment [within the enclosure] so that we can make any necessary modifications if need be. Moreover, if any remarkable changes need to be made to the PCB, we will not want to have to deconstruct a finalized assembly.

9. Senior Design II Addendum

Testing began with breadboarding the major circuit elements: the amplifier circuit, the power supply circuit, and the ADC circuit. All three circuits were tested with a simulated photovoltaic input with a DC power supply. After successful tests with the DC power supply, the design was tested with a 40W power supply to more closely match the expected power of the solar array. Unfortunately, the 40W power supply far exceeded the breadboard's capabilities and subsequently fried it and the connected components. This led to a stronger medium for testing: protoboards.

Reimplementing the same circuits on the protoboards revealed some unexpected issues, which were likely due to reuse of components which had been damaged during the breadboard testing. Considering our prior success and lack of time, we decided to forego further protoboard testing and ordered printed circuit boards (PCBs). With the PCBs ordered, we also had to reorder the other circuit components. Unfortunately, the LMR50410X became backordered, leading us to order its sister components: LMR14010ADDCT and LV2862XLVDDCR. These two parts matched the original's function quite well. We did have to quickly revise our original PCB design and order a new batch (PCB V.2).

With the PCBs assembled, and using the DC power supply to test them, all of the major circuit elements were functioning properly, with the exception of the current side of the amplifier. This issue existed on both versions of the PCB, leading us to believe that the DC power supply couldn't output enough current for the amplifier to recognize. Another theory that was suggested was that the input offset voltage of the amplifier was too high, skewing the small voltage values that would result from the shunt resistor. As a result, we decided on getting the TLV342A, which had a minimum voltage input of 0 V. However, the input offset voltage was high enough that it still interfered with the amperage reading. Then, we got the TSZ122IDT, because of its input offset of only 8 uV. Field testing the current set of PCBs by plugging them into the solar array was the next step.

Amplifier	**Manufacturer**	**Offset Voltage**	**Cost**
TL072	Texas Instruments	6mV	$0.08
TLV342A	Texas Instruments	0.3mV	$0.33
TSZ122IDT	STMicroelectronics	8uV	$2.50

Table 35: New Amplifier Comparison

When implementing the boards with the solar array, a strange problem occurred: the board was only receiving about 1.6V instead of the expected 34V. With the help of our sponsor, we began troubleshooting the issue. We confirmed that the panel was outputting 34V and that the DC optimizer was outputting near 0V when unplugged, as expected. We tried plugging into a different panel on the string, but to no avail. With our options limited, we decided to try plugging a board directly to the panel, expecting to receive all 34V to the board without the regulation of the DC optimizer. Fortunately, our board was able to receive all 34V with no damage. However, this revealed that the issue lay somewhere between the interaction of our board and the DC optimizer, which left our team and our sponsor puzzled.

The final prototype contains a fully functional power supply unit (PSU). As we fed the PSU different input voltages, the PSU consistently produced a steady 5V output, seen in Table 35.

Voltage (V)	8.000	16.000	24.000	32.000
PSU Output (V)	5.000	5.000	5.000	5.000
Percent Error	0%	0%	0%	0%

Table 35: Power Supply Unit Voltage Output Accuracy

With a steady 5V power supply from the PSU, the amplifier functions properly as well. With an input of 8V, the voltage divider steps this down to 1V, and the amplifier amplifies the 1V with a unity gain, seen in Figure 48.

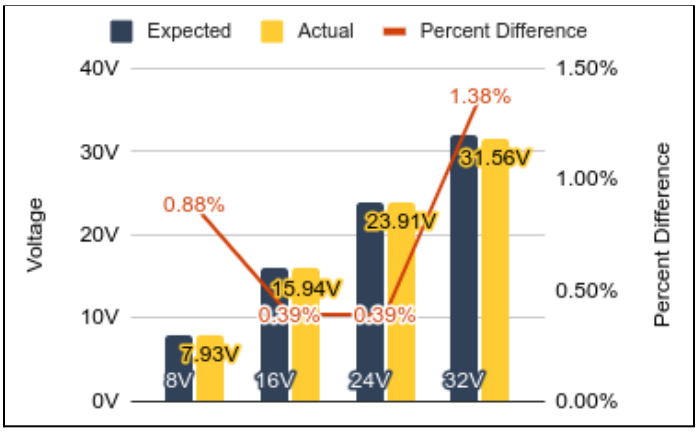

Figure 48: Voltage Amplifier Accuracy

It is crucial to note the integral bypass capacitor connected to the amplifier's V_{DD}. Without this capacitor, the 5V supply would experience ripples, which could increase the supply voltage above the amplifier's rated input. This could cause a short in the amplifier.

The amplifier was also responsible for measuring the current flowing through the panel. However, our amplifier could not get an accurate reading of the input current. This inaccuracy could be the result of a few possibilities. First, when using the DC power supply, we can see how much current is being drawn by the sensor. When simulating the PV input with the DC power supply, the sensor would only be drawing about 1mA. This low current consumption could be so small that the amplifier circuit could not register the value properly. Another possibility could be the input offset voltage of the amplifier. The amplifiers we were using (TL072, TLV342A) may have an input offset voltage that is too high, causing the amplifier to malfunction. Another way to help remedy the offset voltage and possibly any noise involved in the circuit is to add a low pass amplifier to the amplifier. This would help remove any mV noise that could interfere with our low voltage (amperage) reading.

Although we were able to obtain a functioning ADC circuit, we were also not able to transmit the temperature values. Even while using the thermocouple provided by OUC, the readings from the thermocouple were always inaccurate. In fact, the voltage measured between the two leads was inaccurate, which is likely evidence that the product was defective. Another theory suggests that the issue could be easily resolved by supplying a voltage to the thermocouple from the 5 VDC power converter, which would allow the ADC to detect changes in voltage input due to changes in the thermocouple resistance due to temperature. Regardless, this was a low priority for our sensor. In the future, we could definitely spend much more time studying the reference designs of thermocouples, and not underestimate their complexity. The same logic would apply to the DC optimizer as well.

The data transmission time is measured from when the node requests data from a sensor, to when that data is inserted into the local database on the node. Tracking this transmission time across 10,000 data transfers revealed an average of 50 milliseconds. The maximum transfer time was 672 milliseconds, though this only occurred on the first data transfer. All other transfer times were very near the average, leaving the transfer times within the 10 second specification.

As for the budget, we were unable to meet the $20/sensor specification.

Cost per Sensor

Component	Quantity	Cost
NEMA 4X Enclosure	1	$71.97
Mounting Materials	Varies	$4.96
RPi Zero W	1	$10.00
PCB	1	$3.75
MCP3008 ADC	1	$3.75
TSZ122IDT Amp	1	$2.50
LMR50410X PSU[1]	1	$1.47
LMR14010ADDCT PSU[1][2]	1	$1.08
LV2862XLVDDCR PSU[1][2]	1	$1.36
Various Components	14/15	$10.45/$11.20

Total Cost for One Sensor (w/o enclosure)
$34.91

Total Cost for One Sensor (w/ enclosure)
$111.84

[1] Only 1 of the three PSUs are required
[2] Requires diode (extra component)

Figure 49: Cost per Sensor

Ultimately, it seems that that number may have been impractical.

Using the TIDA-00640 reference design, we created a sensor that was able to measure the irradiance and the voltage across a panel with 5% accuracy and communicate the readings to the utility database. With minor modifications, the ambient temperature of the panel could also be communicated to the database. To reiterate, the current-measurement issue is something that can continue to be explored in the next iteration of the project. Furthermore, the future testing between the panels and microcontrollers should maintain some level of electrical isolation in case of voltage spikes. The team learned important lessons on always being aware of the power ratings of the relevant components and handling power electronics. Once the current-measurement is perfected, the sensor will be ready to be installed along solar panel strings and assist utility-based companies and other solar farms with optimizing their solar power output.

10. Resources

[1] "SP-110-SS: Self-Powered Pyranometer," Apogee Instruments, Inc. [Online]. Available: https://www.apogeeinstruments.com/sp-110-ss-self-powered-pyranometer/#product-tab-information. [Accessed: 28-Jan-2021].

[2] "Thermocouple types," https://www.omega.com/en-us/, 14-Oct-2020. [Online]. Available: https://www.omega.com/en-us/resources/thermocouple-types. [Accessed: 28-Jan-2021].

[3] "Raspberry Pi 1 Model B+," Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-1-model-b-plus/. [Accessed: 28-Jan-2021].

[4] "Raspberry Pi Zero W," Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-zero-w/. [Accessed: 28-Jan-2021].

[5] "MC4 Connector," Staubli. [Online]. Available: https://ec.staubli.com/AcroFiles/Technical%20Info/PV_SOL-LVDC_(en).pdf [Accessed: 28-Jan-2021].

[6] B. Anderson and B. Nicholson, "SQL vs. NoSQL Databases: What's the Difference?," IBM. [Online]. Available: https://www.ibm.com/cloud/blog/sql-vs-nosql. [Accessed: 07-Feb-2021].

[7] M. Smallcombe, "SQL vs NoSQL: 5 Critical Differences," Xplenty, 19-May-2020. [Online]. Available: https://www.xplenty.com/blog/the-sql-vs-nosql-difference/. [Accessed: 07-Feb-2021].

[8] M. Mayo, "Top NoSQL Database Engines," KDnuggets, Jun-2016. [Online]. Available: https://www.kdnuggets.com/2016/06/top-nosql-database-engines.html. [Accessed: 07-Feb-2021].

[9] "Raspberry Pi 4 Model B specifications," Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/. [Accessed: 09-Feb-2021].

[10] "BananaPi M5," Banana Pi. [Online]. Available: http://www.banana-pi.org/m5.html. [Accessed: 09-Feb-2021].

[11] "NanoPi M4B," FriendlyElec. [Online]. Available: https://www.friendlyarm.com/index.php?route=product%2Fproduct&path=69&product_id=275. [Accessed: 09-Feb-2021].

[12] "Need a Current Regulator? Use a Voltage Regulator! - Technical Articles," *All About Circuits*. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/need-a-current-regulator-use-a-voltage-regulator/. [Accessed: 11-Feb-2021].

[13] "ACHS-7122-000E," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/broadcom-limited/ACHS-7122-000E/9178255. [Accessed: 11-Feb-2021].

[14] Linear Technology, "800mA Low Dropout Positive Regulators Adjustable and Fixed 2.85V, 3.3V, 5V," 1117fd datasheet. [Accessed: 11-Feb-2021]

[15] Linear Technology, "7ns, Low Power, Single Supply, Ground-Sensing Comparator," 1394f datasheet. [Accessed: 11-Feb-2021]

[16] Linear Technology, "Adjustable 500mA Single Resistor Low Dropout Regulator," 3085fb datasheet. [Accessed: 11-Feb-2021]

[17] L. Hattersley, "Raspberry Pi 4, 3A+, Zero W - specs, benchmarks & thermal tests," The MagPi magazine, 2019. [Online]. Available: https://magpi.raspberrypi.org/articles/raspberry-pi-specs-benchmarks. [Accessed: 22-Feb-2021].

[18] "BPI-M2 Zero," Banana Pi. [Online]. Available: http://www.banana-pi.org/m2z.html. [Accessed: 22-Feb-2021].

[19] "Orange Pi i96," Orange Pi. [Online]. Available: http://www.orangepi.org/OrangePii96/. [Accessed: 22-Feb-2021].

[20] "List of ARM microarchitectures," Wikipedia, 06-Feb-2021. [Online]. Available: https://en.wikipedia.org/wiki/List_of_ARM_microarchitectures. [Accessed: 23-Feb-2021].

[21] Microchip, "2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface," MCP3008 datasheet. [Accessed: 24-Feb-2021]

[22] Texas Instruments, "Ultra-Small, Low-Power, 12-Bit Analog-to-Digital Converter with Internal Reference," ads1015 datasheet. [Accessed: 24-Feb-2021]

[23] Texas Instruments, "Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference," ads1115 datasheet. [Accessed: 24-Feb-2021]

[24] B. Ray, "Bluetooth Vs. Bluetooth Low Energy (BLE): What's The Difference?," Link Labs, 01-Nov-2015. [Online]. Available: https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy. [Accessed: 25-Feb-2021].

[25] E. F. Y. Team and P., "Comparison: Buck Converter vs. Linear Voltage Regulator," *Electronics For You*, 16-Jun-2018. [Online]. Available: https://www.electronicsforu.com/videos-slideshows/buck-converter-vs-linear-voltage-regulator-3#:~:text=Linear%20voltage%20converter%20heats%20up,and%20switches%20back%20on%20again.&text=Buck%20converter%20can%20provide%20variable,stuck%20at%20only%201%20output. [Accessed: 04-Mar-2021].

[26] J. Arellano, "Bluetooth vs. Wi-Fi for IoT: Which is Better?," Very, 09-Jul-2019. [Online]. Available: https://www.verypossible.com/insights/bluetooth-vs.-wi-fi-for-iot-which-is-better. [Accessed: 08-Mar-2021].

[27] P. R. Prasad, "BLE vs Wi-Fi: Which is Better for IoT Product Development?," Cabot, 01-Feb-2018. [Online]. Available: https://www.cabotsolutions.com/ble-vs-wi-fi-which-is-better-for-iot-product-development#:~:text=Bluetooth%20devices%20have%20lower%20power,Fi%20devices%20consume%20more%20power.&text=The%20signals%20edited%20by%20iBeacon,are%20performing%20the%20same%20tasks. [Accessed: 08-Mar-2021].

[28] A. Instruments, "Apogee Pyranometers," *Apogee Instruments*. [Online]. Available: https://www.apogeeinstruments.com/content/SP-100-200-spec-sheet.pdf. [Accessed: 10-Mar-2021].

[29] M. Gostein, "Update on Edition 2 of IEC 61724: PV System Performance Monitoring," *National Renewable Energy Lab*. [Online]. Available: https://www.nrel.gov/pv/assets/pdfs/2014_pvmrw_84_gostein.pdf. [Accessed: 10-Mar-2021].

[30] "SP-110-SS: Self-Powered Pyranometer," *Apogee Instruments, Inc.* [Online]. Available: https://www.apogeeinstruments.com/sp-110-ss-self-powered-pyranometer/#product-tab-information. [Accessed: 10-Mar-2021].

[31] "SP-510-SS Upward-Looking Thermopile Pyranometer," *Apogee Instruments, Inc.* [Online]. Available: https://www.apogeeinstruments.com/sp-510-ss-upward-looking-thermopile-pyranometer/#product-tab-information. [Accessed: 10-Mar-2021].

[32] "SP-422-SS Modbus Digital Output Silicon-cell Pyranometer," *Apogee Instruments, Inc.* [Online]. Available: https://www.apogeeinstruments.com/sp-422-ss-modbus-digital-output-silicon-cell-pyranometer/#product-tab-information. [Accessed: 10-Mar-2021].

[33] "SP-522-SS Modbus Digital Output Thermopile Pyranometer," *Apogee Instruments, Inc.* [Online]. Available: https://www.apogeeinstruments.com/sp-522-ss-modbus-digital-output-thermopile-pyranometer/#product-tab-information. [Accessed: 10-Mar-2021].

[34] "Thermocouples," *Thermocouple Info*. [Online]. Available: https://www.thermocoupleinfo.com/. [Accessed: 14-Mar-2021].

[35] Person, *Thermocouple Types and Thermocouple Range*. [Online]. Available: https://www.thomasnet.com/articles/automation-electronics/Thermocouples-Types/. [Accessed: 14-Mar-2021].

[36] Greentumble, "Effect of Temperature on Solar Panel Efficiency," *Greentumble*, 05-Dec-2020. [Online]. Available: https://greentumble.com/effect-of-temperature-on-solar-panel-efficiency/. [Accessed: 14-Mar-2021].

[37] Ultra Librarian PCB CAD Database [Online]. Available: https://www.ultralibrarian.com/ [Accessed: 18-Mar-2021]

[38] "What are DC power optimizers?," Solar Reviews, 21-Jan-2021. [Online]. Available: https://www.solarreviews.com/blog/complete-guide-to-power-optimizers. [Accessed: 21-Mar-2021].

[39] SolarEdge Power Optimizer Increases Energy Output. [Online]. Available: https://www.solaredge.com/us/products/power-optimizer. [Accessed: 21-Mar-2021].

[40] K. Zipp, K. Zipp, R. P. M. D. says, R. P. McDonald, B. says, Bieng, R. H. says, R. Harrisson, J. says, and Jones, "How do power optimizers help harvest more energy from solar projects?," Solar Power World, 07-Jan-2016. [Online]. Available: https://www.solarpowerworldonline.com/2015/11/23495/. [Accessed: 21-Mar-2021].

[41] "Mismatch for Cells Connected in series," PVEducation. [Online]. Available: https://www.pveducation.org/pvcdrom/modules-and-arrays/mismatch-for-cells-connected-in-series. [Accessed: 21-Mar-2021].

[42] "NEMA Ratings For Enclosures," NEMA Enclosures. [Online]. Available: https://www.nemaenclosures.com/enclosure-ratings/nema-rated-enclosures.html. [Accessed: 21-Mar-2021]

[43] "What's the difference between Scripting and Programming Languages?," *GeeksforGeeks*, 21-Jan-2019. [Online]. Available: https://www.geeksforgeeks.org/whats-the-difference-between-scripting-and-programming-languages/. [Accessed: 23-Mar-2021].

[44] A. Monus, "Top 13 Scripting Languages You Should Pay Attention to in 2021," *Kinsta*, 19-Mar-2021. [Online]. Available: https://kinsta.com/blog/scripting-languages/. [Accessed: 23-Mar-2021].

[45] A. Desai, "Top 5 Reasons to Choose JavaScript for Your IoT Project," *IoT For All*, 18-Feb-2020. [Online]. Available: https://www.iotforall.com/javascript-iot. [Accessed: 23-Mar-2021].

[46] P. Wayner, "The top 6 programming languages for IoT projects," *TechBeacon*. [Online]. Available: https://techbeacon.com/app-dev-testing/top-6-programming-languages-iot-projects#:~:text=In%20a%20survey%20of%20developers,are%20%E2%80%9Cbuilding%20IoT%20solutions.%E2%80%9D. [Accessed: 23-Mar-2021].

[47] "A Brief Comparison of NEMA 250 and IEC 60529," *NEMA*. [Online]. Available: https://www.nema.org/standards/view/a-brief-comparison-of-nema-250-and-iec-60529. [Accessed: 25-Mar-2021].

[48] "NEMA 250 Type 4X Enclosures," *Keystone Compliance*, 26-Jun-2020. [Online]. Available: https://keystonecompliance.com/nema-250-type-4x-enclosures/. [Accessed: 25-Mar-2021].

[49] "IEEE 802.15.4-2020 - IEEE Standard for Low-Rate Wireless Networks," *IEEE SA - The IEEE Standards Association - Home*. [Online]. Available: https://standards.ieee.org/standard/802_15_4-2020.html. [Accessed: 25-Mar-2021].

[50] "PEP 0 -- Index of Python Enhancement Proposals (PEPs)," *Python.org*, 13-Jul-2000. [Online]. Available: https://www.python.org/dev/peps/#introduction. [Accessed: 29-Mar-2021].

[51] S. Kumar Pal, "Coding Standards and Guidelines," *GeeksforGeeks*, 02-Jul-2019. [Online]. Available: https://www.geeksforgeeks.org/coding-standards-and-guidelines/#:~:text=Purpose%20of%20Having%20Coding%20Standards,helps%20to%20detect%20error%20easily. [Accessed: 29-Mar-2021].

[52] "Enclosures," *McMaster*. [Online]. Available: https://www.mcmaster.com/enclosures/material~plastic/material~polycarbonate-plastic/cover-attachment-style~hinged/cover-closure-method~quick-release-latch/. [Accessed: 07-Apr-2021].

[53] "Versa-Mount Polycarbonate Washdown Enclosure," *McMaster*. [Online]. Available: https://www.mcmaster.com/5376K621/. [Accessed: 07-Apr-2021].

[54] Gabrian, "6 Heat Sink Types: Which One is Best for Your Project?," *Gabrian*, 11-Oct-2019. [Online]. Available: https://www.gabrian.com/6-heat-sink-types/. [Accessed: 07-Apr-2021].

[55] "UART vs I2C vs SPI – Communication Protocols and Uses," Seeed Studio, 2019. [Online]. Available: https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses/. [Accessed: 08-Apr-2021].

[56] "UART vs SPI vs I2C | Difference between UART, SPI and I2C," RF Wireless World. [Online]. Available: https://www.rfwireless-world.com/Terminology/UART-vs-SPI-vs-I2C.html. [Accessed: 08-Apr-2021].

[57] R. Kiefer and A. Sewrathan, "MongoDB Time-Series - A NoSQL vs. SQL Database Comparison," Timescale, 17-Dec-2020. [Online]. Available: https://blog.timescale.com/blog/how-to-store-time-series-data-mongodb-vs-timescaledb-postgresql-a73939734016/. [Accessed: 08-Apr-2021].

[58] "SD cards," SD cards - Raspberry Pi Documentation. [Online]. Available: https://www.raspberrypi.org/documentation/installation/sd-cards.md. [Accessed: 08-Apr-2021].

[59] Shawn, "Raspberry PI Operating Systems (OS) - Which one to use in 2020?," Seeed Studio, 2020. [Online]. Available: https://www.seeedstudio.com/blog/2019/10/29/raspberry-pi-operating-systems-os-which-one-to-use/. [Accessed: 08-Apr-2021].

[60] "20 Best Operating Systems You Can Run on Raspberry Pi in 2021," FOSSMint, 25-Mar-2021. [Online]. Available: https://www.fossmint.com/operating-systems-for-raspberry-pi/. [Accessed: 08-Apr-2021].

[61] J. Hendrickson, "What Is Windows 10 IoT, and When Would You Want to Use It?," How-To Geek, 06-May-2019. [Online]. Available: https://www.howtogeek.com/413102/what-is-windows-10-iot-and-when-would-you-want-to-use-it/. [Accessed: 08-Apr-2021].

[62] R. Hartle, "How Heat Sinks Work," *HowStuffWorks*, 31-Aug-2010. [Online]. Available: https://computer.howstuffworks.com/heat-sink.htm#:~:text=Heat%20sinks%20work%20by%20redirecting,the%20surroundings%20to%20transfer%20heat. [Accessed: 09-Apr-2021].

[63] B. Loeffler, "Heat Sinks and Process Cooling," *North Slope Chillers*, 23-Mar-2020. [Online]. Available: https://northslopechillers.com/blog/heat-sinks-and-process-cooling/. [Accessed: 09-Apr-2021].

[64] "FIT0367," *DigiKey*. [Online]. Available: https://www.digikey.com/en/products/detail/dfrobot/FIT0367/6588428?utm_adgroup=Thermal+-+Heat+Sinks&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Product_Fans%2C+Thermal+Management_NEW&utm_term=&utm_content=Thermal+-+Heat+Sinks&gclid=Cj0KCQjwsLWDBhCmARIsAPSL3_0ptOmAUJJllodW4utapkGCSDyz25aifIE4IlIkm5TwUiRYRUHdx-4aAg7GEALw_wcB. [Accessed: 09-Apr-2021].

[65] A. Industries, "Mini Aluminum Heat Sink for Raspberry Pi - 13 x 13 x 3mm," *adafruit industries blog RSS*, 2020. [Online]. Available: https://www.adafruit.com/product/3084?gclid=CjwKCAjw07qDBhBxEiwA6pPbHlcoOsuCfawKqSTdV_yfu6NYvyly2UXvMYTswjmMNLeWAkZTPTP1sRoCPZMQAvD_BwE. [Accessed: 09-Apr-2021].

[66] Justin, "PV Panel output voltage - shadow effect?," Victron Energy, 20-Feb-2020. [Online]. Available: https://www.victronenergy.com/blog/2020/02/20/pv-panel-output-voltage-shadow-effect/. [Accessed: 14-Apr-2021].

[67] A. Bonkaney, S. Madougou, and R. Adamou, "Impacts of Cloud Cover and Dust on the Performance of Photovoltaic Module in Niamey," Journal of Renewable Energy, 07-Sep-2017. [Online]. Available: https://www.hindawi.com/journals/jre/2017/9107502/. [Accessed: 14-Apr-2021].

[68] S. Fox, "How Does Heat Affect Solar Panel Efficiencies?," CED Greentech, Nov-2017. [Online]. Available: https://www.cedgreentech.com/article/how-does-heat-affect-solar-panel-efficiencies#:~:text=As%20the%20temperature%20of%20the,solar%20panel's%20production%20of%20power. [Accessed: 14-Apr-2021].

[69] "Dirty Solar Panels - Should you clean them?," HESOLAR, 14-Aug-2017. [Online]. Available: https://www.hesolarllc.com/cleaning-dirty-solar-panels/. [Accessed: 14-Apr-2021].

[70] M. R. Maghami, H. Hizam, C. Gomes, M. A. Radzi, M. I. Rezadad, and S. Hajighorbani, "Power loss due to soiling on solar panel: A review," ScienceDirect, Jun-2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032116000745. [Accessed: 14-Apr-2021].

[71] *What is Hall Effect and How Hall Effect Sensors Work*. YouTube, 2015.