Solar Tracker

Cooper Hamlin
Osman Farook
Talha Zaheer
Alexandra Torres

Final Report

REVISION – 4
December 3rd, 2023

Table of Contents:

Solar Tracker: ECEN 403

Cooper Hamlin
Osman Farook
Talha Zaheer
Alexandra Torres

Concept of Operations

REVISION - 4
4 December 2023

Concept of Operations for Solar Tracker

Team <41>

Approved by:

Cooper Hamlin 12/04/2023
Project Leader Date

Prof. Kalafatis Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	[2/10]	Team 41		Draft Release
1	[2/20]	Cooper		Mid Term Report Changes
2	[4/26]	Cooper		403 Final Report Changes
3	[12/4]	Cooper		404 Final Report Changes

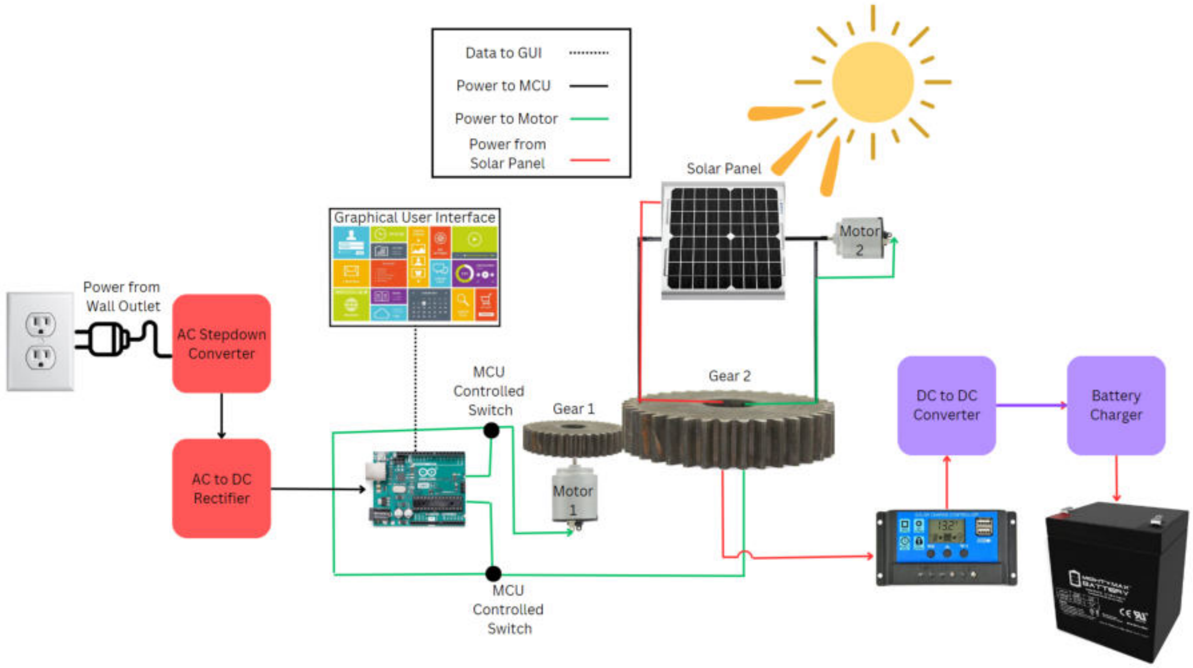
Table of Contents

List of Figures

1. Executive Summary

As green energy revolutionizes the modern world, solar panels have one primary issue that decreases efficiency: They don't always face the sun. Our team is working to build an autonomous solar tracker that increases the power output of a solar panel by most effectively facing the sun. By facing the sun perfectly, we can increase the efficiency of the solar panel by maximizing the duration of direct sunlight incident on the panel. We will achieve this with an autonomous mechanism designed to adjust the angle of the panels to follow the sun throughout the day. Unfortunately, the cost of tracking the sun is greater than buying additional panels, But this system will significantly increase the total output of solar energy conversion and will especially outweigh the cost issue in the scenario that it is not feasible to implement more panels due to space restrictions.

Figure 1: Full Design Concept



2. Introduction

This document is an introduction to the Dual-axis Solar Tracker, a device capable of following the movement of the sun to gain the maximum amount of sunlight. Compared to a fixed solar panel, the solar tracker will be based on a subsystem of microcontrollers and photo-sensors to precisely locate the sun's position and adjust its angle accordingly. The necessary information will also be displayed in a Graphical User interface to allow the user easy access to the information of the system.

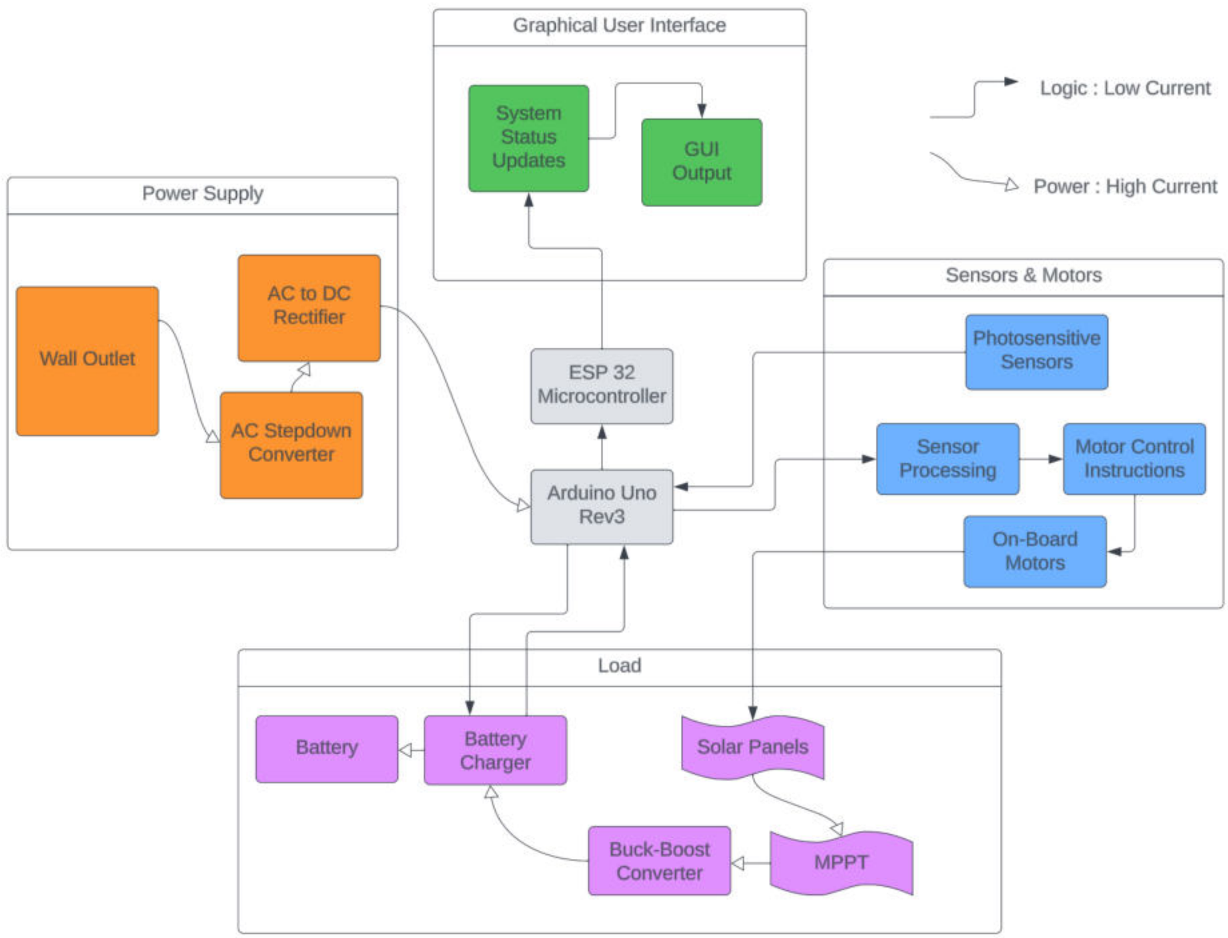
2.1. Background

Solar panels are gradually becoming more popular as the world shifts to renewables to keep up with growing energy consumption. According to statistics by NASA, the Sun gives off enough light to produce 44 quadrillion watts of power each year. The biggest obstacle preventing photovoltaics from taking the lead in the energy competition is efficiency. Fossil fuels have remained atop the pecking order of energy sources for their robust production of power. However, with the climate crisis worsening each year in large part due to fossil fuel use, the time for solar power is now. In order to replace the use of non-renewables and become worthy of more investment, solar energy will need to become a lot more efficient. Given that solar panels do not have a mechanism on their own that follows the movement of the sun, the energy gained at a given time may not be at the maximum potential.

The Solar Tracker improves on a traditional design by following the movement of the sun. The dual-axis nature of the device will allow the solar panel to always be at the optimum position to maximize efficiency. A study published in the European Scientific Journal found that dual-axis solar trackers increased energy output by 25.2% as compared to a fixed-tilt solar panel. Although solar trackers would be a big investment, the potential benefits in terms of power would be undeniable.

2.2. Overview

Figure 2: Solar Tracker Subsystem Block Diagram



Our system will be able to house a solar panel and use a motor based structure to move according to the position of the sun. The design will contain light based sensors that will be programmed to detect the optimal position for the solar panel to gain the most sunlight. The solar tracker will utilize 2 microcontrollers and 2 motors that will allow the panel to move in each cardinal direction. The first of the two microcontrollers(Arduino rev 3) will be able to read the data and the motors will move the device to that specific setting. The second Microcontroller will take gyroscope, current, and voltage readings to determine battery capacity and Solar Panel configuration. The device will be powered through a wall outlet and a Transformer will step down the AC voltage and then a diode bridge will convert the power to a DC 12 V voltage setting to be used by the microcontrollers, sensors, motors, etc. The energy produced by the solar tracker will then be transferred to a load in the form of a battery which will also allow for measurements to be done to differentiate between the energy produced between a fixed tilt, single axis and dual axis form of operation. The sensor data readings obtained from the second microcontroller will be transmitted wirelessly, via WiFi, to a database. A website will be developed to obtain the data received from the database and display that information in a user-friendly format, providing users with real-time, daily insights into the system’s load and the solar panel’s orientation.

Figure 3: Solar Panel

2.3. Referenced Documents and Standards

[1] Amazon.com: Acopower Solar Panel 100 Watt 12 Volt PV Module High https://www.amazon.com/ACOPOWER-Watts-Connectors-Battery-Charging/dp/B08GM2ZGXS.

[2] Dual-axis Solar Tracker vs Fixed Tilt Solar panel study: Axaopoulos, Petros J., and Emmanouil D. Fylladitakis. "Energy and Economic Comparative Study of a Tracking

vs. a Fixed Photovoltaic System.” *CORE*, Apr. 2013, https://core.ac.uk/display/236417964?utm_source=pdf&utm_medium=banner&utm_campaign=pdf-decoration-v1.

[3] IEEE Solar Performance Analysis N. Othman, M. I. A. Manan, Z. Othman and S. A. M. Al Junid, "Performance analysis of dual-axis solar tracking system," 2013 IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, 2013, pp. 370-375, doi: 10.1109/ICCSCE.2013.6719992.

[4] NASA statistic on Solar Energy https://www.nasa.gov/pdf/135642main_balance_trifold21.pdf.

[5] Standard for Wall Outlet Voltage https://energyeducation.ca/encyclopedia/Electrical_outlet#:~:text=Domestic%20electrical%20outlets%20supply%20120,one%20of%20those%20two%20values.

3. Operating Concept

3.1. Scope

We were tasked with designing and constructing a Dual-Axis solar tracker to help maximize solar panel output. The dual-axis mechanism allows us to achieve three-dimensional adjustments. In contrast, a single-axis construction will only achieve its best performance when the sun is in its apex position in the sky which is severely inefficient. Therefore we will use a dual-axis system to account for the sun’s changing path through the sky. In order to do this effectively we will update the system periodically at five minutes to find the new current location of the sun and adjust the solar panel to face the sun directly. Considering that we have a $300 budget to accomplish the entire project, we decided to downsize our project to accommodate a solar panel of dimensions 11.5 by 11.5 by 9 inches allowing us to achieve a cost effective proof of concept with reasonable technical merit.

3.2. Operational Description and Constraints

The solar tracker is designed to be a prototype version of a larger, more practical, mechanism. A smaller version is being constructed to keep the project within our $400 budget. The solar tracker helps to maximize the surface area and duration of incident sunlight on the solar panel to get the most productive solar panel possible.

The resulting constraints from this operational description are as follows:

- A $400 budget for the entirety of our project will limit the quality and size of the parts that are able to be used.
- The system will require the user to place it in an area with plenty of sunlight in order to maximize the efficiency of the solar panel.
- The system will require the user to place the Solar Tracker in close proximity to a wall outlet in order to power the device.
- The system will require the parts that make up the design to be durable and capable of working in most weather conditions.

3.3. System Description

Our project has 4 main subsystems consisting of an AC-DC Power supply subsystem, a Microcontroller & Photo-Sensor Processing subsystem, a Graphical USer Interface subsystem and a Solar Powered Buck Converter and Battery Charger subsystem.

- Power supply: The power supply for the system will be from a wall outlet. Given that there will be a constant AC power supply, a rectifier will be used to change the AC power to DC. However, to make it safer to use the input voltage, a transformer will be utilized in order to lower the voltage in AC before the rectification process. Once the voltage is in DC, a capacitor filter will be attached to filter out ripple and the voltage will feed into a regulator to output the constant voltage required for the system.
- Microcontroller & Photo-Sensor Processing: This subsystem is responsible for the acquisition of sensory data which is to be processed by microcontrollers, resulting in a generated decision to be directed to the motorized axis subsystem to make optimal angle adjustments. The subsystem takes its input from four photosensitive sensors placed in a voltage divider circuit in a “+” shape layout mounted at the top of the panel. The microcontroller will then analyze the signals and energy levels to determine which sensor is receiving the most sunlight, this information is the basis of all decisions. The code written will generate instructions to be sent to the two motor controllers (one custom PCB made and one commercial made) to move the on board Dual Axis stepper motors which upon successful execution will result in adjusting the position and angle for the panel to be in its most optimal orientation i.e directly facing the sun. The code’s instruction set includes but is not limited to edge case scenarios for rough weather conditions, cloudy days and the night time. Its notable default position will be a flat upward facing position at the end of each day.

- Solar Powered Buck-Boost Converter and Battery Charger: The now correctly placed solar panel will begin to provide both voltage and current to the system. A Maximum Power Point Charger (MPPT) will determine the ideal Power setting to input into the buck converter. This voltage will be taken to a DC-DC Buck converter to prepare it for battery charging. The battery will require an input voltage of 14.7 to charge and will be denoted as phase 1. Once the battery is close to fully charged the battery charger will disconnect the ground end of the battery effectively turning the battery charger off. This allows the battery to increase the efficiency at which it charges and keep the battery from suffering from internal damage due to overcharging.
- Graphical User Interface: Data from a current sensor, voltage sensor, and gyroscope will be collected and showcased on a dashboard-style website. This setup enables users to easily access and understand information regarding the solar panel's load and positioning. Additionally, a separate microcontroller will be set up to gather data from these three sensors. It will be programmed to transmit this collected data to a database through WiFi. Finally, a website will be developed to retrieve and present this data from the database in a user-friendly, interactive format.

3.4. Modes of Operations

The Solar Tracker will have two modes of operation which will be on and off. While on, the solar tracker will adjust to the current location of the sun to maximize energy production of the on board solar panel. While off the solar tracker will be stationary and not follow the sun’s path through the sky. Notably, the solar panel will still be producing electrical energy at whatever rate the default position allows for, usually much lesser than when it is in the “on mode”

3.5. Users

Our Solar Tracking system will be marketed towards people who have high energy consumption demands without the ability or space to add numerous panels. The device would allow for buildings such as skyscrapers to get the maximum output out of their solar panels ls.

The highlight of this system is that anybody possessing a solar panel could significantly increase its output efficiency through the use of our Dual-Axis Solar tracker. The extra initial investment may seem discouraging at first but we are confident that the energy gained from increased output will eventually be offset by the user getting the most out of their solar panels. The system is designed for easy installation, requiring minimal experience. The device may require basic maintenance from time to time based on weather conditions and excessive usage to ensure that it is performing optimally.

3.6. Support

Support for the Solar Tracker will be provided in the form of a detailed user manual providing information on system installation, maintenance, and recommended usage. This will also include a guide to solar tracking, as well as how the solar panel should be operated to assist the user in understanding the movements of the solar tracker. Additionally, a website will be developed to provide helpful information regarding the performance of the load of the system.

4. Scenarios

4.1. On the Move Research Trips

A dual axis solar tracker will yield a greater energy output than a traditional solar panel by about 40%. Oftentimes, researchers and recreational tourists will go on trips deep into wilderness zones detached from the electrical grid. An alternative to bringing gas powered generators, a solar panel with a solar tracker allows the user to generate and use energy on the go. Though the Solar tracker would have a certain weight, it would be more travel efficient than bringing multiple solar panels.

4.2. Skyscraper Roof Solar Systems

A dual axis solar tracker has the other benefit of making each solar panel more effective at harnessing the energy from the sun. This allows for greater energy production when space is limited and the user needs to maximize production of power for each solar panel. Many skyscrapers house large numbers of residential living spaces, office spaces, and shopping centers. This demands a large amount of energy to power the entirety of the building. By placing solar panels on the roof many structures have begun to help offset the energy consumption of their building. By using solar tracking in the limited horizontal space the user can maximize the space on top of their building to most effectively harness the sun’s rays.

5. Analysis

5.1. Summary of Proposed Improvements

- The improved system would be significantly larger. This would allow for increased power output which would lead to faster charging times. This would also grant the user the ability to charge a larger battery in a similar time frame.
- The improved system will store energy in the battery that will simultaneously run the solar tracker's dual axis motors. This would allow for a self-sufficient system instead of one that requires an outside power source.

5.2. Disadvantages and Limitations

- The improved system will be more expensive as the ability for the solar tracker to move will come at a premium. Due to its larger size potentially larger motors will be required to rotate the dual axis of the tracker.
- The system may require more frequent maintenance to ensure that the device is working optimally as the introduction of self sufficiency leads to an issue that if 1 issue occurs the entire system will fail.
- The system will constantly consume a fixed amount of energy to power the movement capability of the device. This leads to decreased available energy produced by the tracker for other uses.

5.3. Alternatives

A few alternatives to a dual-axis solar tracker would be:

- A passive dual-axis tracker that has its angular motion based on a low boiling point liquid that would cause the mechanism to tilt towards the sunlight as the liquid evaporates/cools.
- A dual axis solar tracker could use sun tracking data from websites such as Gaisma and follow a predicted path of the sun instead of the solar tracker tracking it itself.
- A tracker could also use the fact that the sun moves roughly three hundred and sixty degrees per 24 hours. This is equal to 15 degrees per hour. This would just require the tracker to rotate 15 degrees every hour and then reset at the end of each day.

5.4. Impact

- This system will significantly increase the efficiency of solar panels through the movement setting of the design.
- Allows for use of solar technology in more wilderness research cases that will decrease the use of gas powered generators.
- An increase in solar energy efficiency leads to an increase in use of solar panels. This will decrease the demand for fossil fuels and will have significant environmental benefits.
- Although the use of a solar tracker will have an initial cost it will be outweighed over time through the increased production of power from the solar panel.
- Improving solar production allows for an increase in use of green energy and a decrease in fossil fuels leading to greenhouse gas emissions.

- The use of a rechargeable battery will have some negative environmental effects in the short term but the benefits of solar panel use over time will outweigh the environmental production costs.

Solar Tracker

Cooper Hamlin
Osman Farook
Talha Zaheer
Alexandra Torres

Functional System Requirements

Functional System Requirements
for
Solar Tracker

Prepared by:

____
Author Date

Approved by:

________Cooper Hamlin__________
Project Leader Date

____
John Lusher, P.E. Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	2/14/23	Full Team		Draft Release
1	2/20/23	Cooper Hamlin		Midterm Report Updates
2	12/4/23	Cooper Hamlin		Final Report Updates

Table of Contents

List of Tables

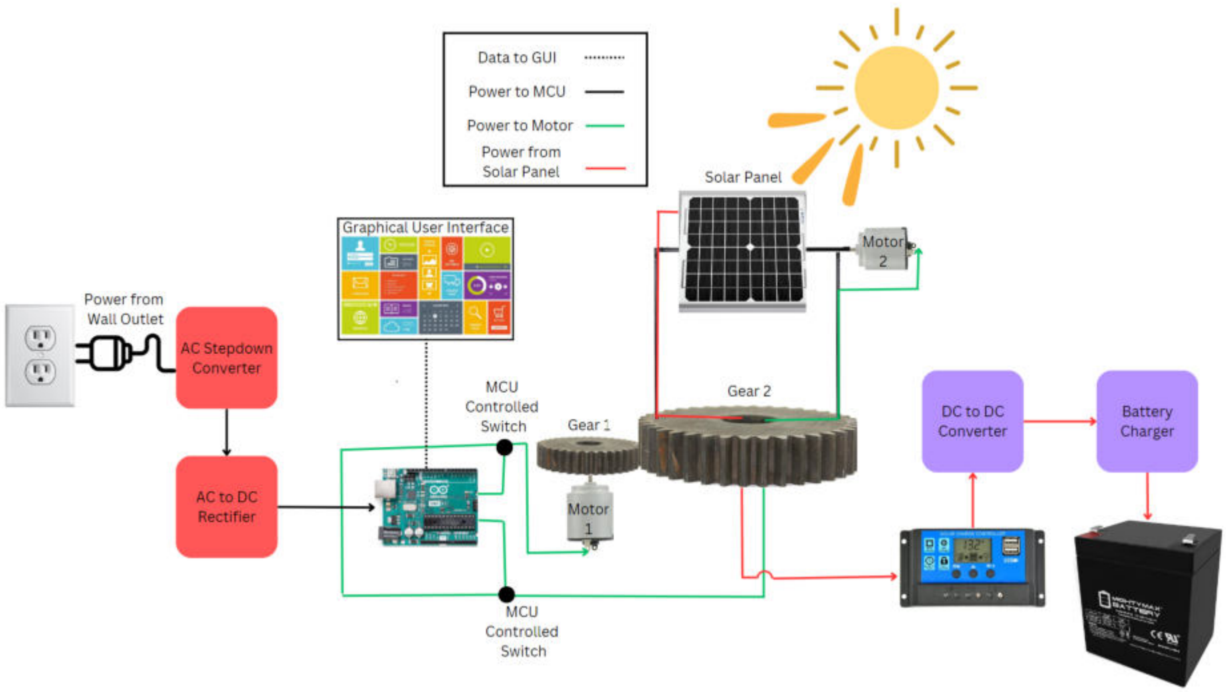
List of Figures

1. Introduction

1.1. Purpose and Scope

Solar Energy grows more popular by the day and the environmental concerns around non-renewable energy mean that the world will need to eventually replace this source of energy. The Solar Tracker is a dual-axis device that will house an existing solar panel, will be able to detect the optimal position for maximizing sunlight and will be able to move the panel to that specific setting. Our device will be able to rotate in each cardinal direction. It will be connected to a load in the form of a battery in order to test the energy outputted by the dual-axis nature of the solar panel. The power to the battery will go through a Maximum Power Point Tracker or MPPT to maximize the energy extraction from the Solar Panel. A Battery Charger will also be used to ensure that the battery does not over-charge. Figure 1 shows our proposed design with all the working subsystems. Our system's data readings will be displayed in a website, allowing the user to obtain information regarding the system on-the-go.

Figure 1. Solar Tracker Conceptual Diagram



The following definitions differentiate between requirements and other statements.

Shall:	This is the only verb used for the binding requirements.
Should/May:	These verbs are used for stating non-mandatory goals.
Will:	This verb is used for stating facts or declaration of purpose.

1.2. Responsibility and Change Authority

The team leader Cooper Hamlin will be responsible for verifying the project requirements' completion. Constant communication with the sponsor, Wonhyeok Jang will be needed to approve the goals of the project.

Table 1: Subsystem Leads.

Subsystem	Responsibility
Microcontroller & Photosensor Processing	Osman Farook

DC-DC Buck Converter and Battery Charging	Cooper Hamlin
Power Supply	Talha Zaheer
Graphical User Interface	Alexandra Torres

2. Applicable and Reference Documents

2.1. *Applicable Documents*

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Table 2: Applicable Documents

Document Number	Revision/Release Date	Document Title
DOD-HDBK-791	3/17/1998	Maintainability Design Techniques Metric
MIL-HDBK-217	Revision F – 2/28/1995	Reliability Prediction of Electronic Equipment
MIL-HDBK-263	Revision B – 7/31/1994	Electrostatic Discharge Control Handbook for Protection of Electrical and Electronic Parts, Assemblies and Equipment
MIL-HDBK-338	Revision B – 10/1/1998	Electronic Reliability Design
IPC A-610E	Revision E – 4/1/2010	Acceptability of Electronic Assemblies
MIL-STD-461	Revision E – 8/20/1999	Requirements for the Control of Electromagnetic Interface Characteristics of Subsystems and Equipment
MIL-HDBK-1003/19	Revision C – 2/1/2010	DESIGN PROCEDURES FOR PASSIVE SOLAR BUILDINGS

2.2. *Reference Documents*

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Table 3: Reference Documents

Document Number	Revision/Release Date	Document Title
IEC 61427	1999	Secondary cells and batteries for solar

		photovoltaic energy systems - General requirements and methods of test
IEC 62548	2016	Photovoltaic (PV) arrays - Design requirements
IEC 61724-3	2016	Photovoltaic system performance - Part 3: Energy evaluation method
IEC 60904-1	2006	Photovoltaic devices-Part 1: Measurements of PV current-voltage characteristics

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings, or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

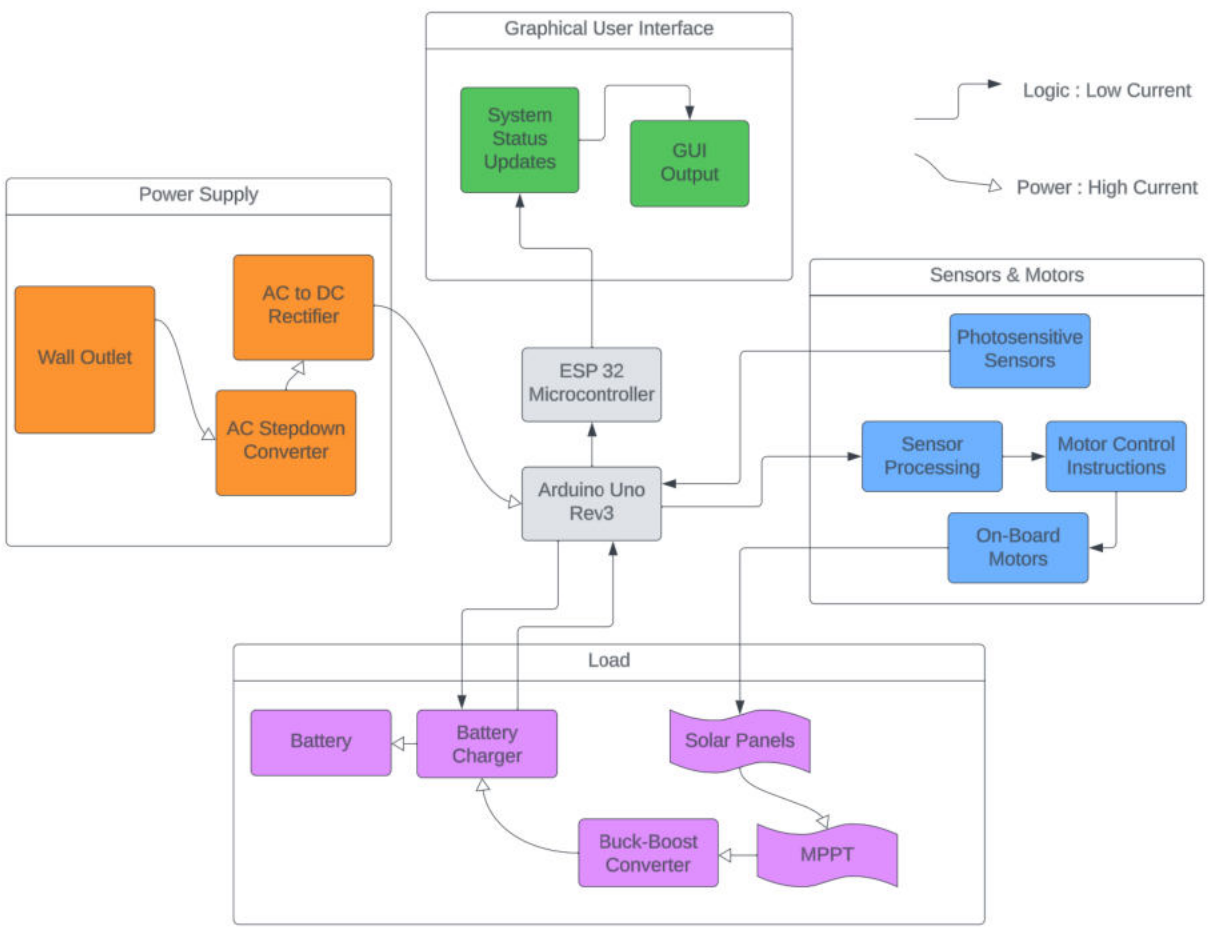
3. Requirements

In the following section, “Dual Axis Solar Tracker” will exclusively refer to the entire system for which the proof of concept is being developed in the scope of this project. This includes the Power supply containing the AC-AC Transformer, Diode bridge rectifier, Capacitor filter and voltage regulator. For the MCU Sensor subsystem this includes the microcontroller, photosensitive sensors, and Arduino IDE code. For the Load Subsystem this includes the Solar Panel, DC-DC Buck Converter, MPPT, Battery, Arduino IDE battery charger.

3.1. System Definition

The Solar Tracker is a reliable upgrade to a fixed Solar Panel design. It will allow users to improve the efficiency of their existing panels and will work autonomously. The Solar Tracker has 4 distinct subsystems: Power Supply, MCU Sensors, Mobility Plus Load, and the Graphical User Interface, which are shown in Figure 2.

Figure 2. Block Diagram of Dual Axis Solar Tracker System



The Power Subsystem will use a wall outlet to power the working parts of each of the subsystems. As shown in the figure above, the input voltage will be transformed down from AC to AC then using a rectifier and a DC voltage regulator that will be used to convert the voltage in order to power the Arduino Uno Microcontroller and 2 motors.

The MCU Sensor Subsystem will use 4 Photosensitive Sensors used that will provide the sunlight data to the microcontroller. There will be 2 motors within the design and the microcontroller will provide the instructions to each motor in terms of how much they need to rotate in order to get to the optimal position. The first motor will be at the base of the design and will be attached to 2 gears that will allow the panel to move east or west. The second motor will be attached to the housing on the panel itself and will allow it to move up and down thus creating the dual-axis nature of the design.

The Graphical User Interface (GUI) Subsystem will obtain readings from sensors measuring current, voltage, and angular motion, and present this information in a website. There will be a second microcontroller that will be configured with an INA219, voltage sensor, and an MPU6050 sensor to transmit the sensor data to a database. The website will retrieve the data from the database and display the data in a dashboard-like configured website. The second microcontroller will be powered through the arduino, utilizing its 3.3V power supply. The MPU6050 gyroscope will be mounted onto the surface of the solar panel, while the current and voltage sensors will be linked to the load subsystem's buck converter.

The Load Subsystem will use the power generated from the Solar Panel to recharge the battery that will act as the load. A DC-to-DC Buck converter will be used to tend to the specifications of the battery which requires around 14.7 Volts to charge optimally which is referred to as phase 1. An MPPT will be used to maximize the power output from the Solar Panel at an average of 17.5 V. A Battery Charger will use the output from the DC-DC converter to ensure that the Battery does not overcharge. It will do this by decreasing the output of the DC-DC Buck Converter to 13.7 when the battery is close to fully charging which is referred to as phase 3. Once the battery is fully charged the battery will disconnect from the charger and stop attempting to charge the battery. When the battery is disconnected it is referred to as phase 3.

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1. Efficiency

The Solar Tracker shall be at least 20% more efficient than a fixed tilt design.

3.2.2. Physical Characteristics

3.2.2.1. Structure

The mass of the Solar Tracker shall be less than or equal to 30 lbs. The Solar Tracker shall be elevated from the base using 2 metal rods. This structure shall have a motor at the top alongside the panel to move it up and down and a second motor near the base connected to 2 gears to allow it to move left and right. See Figure 1 for the complete design.

Rationale: This is a number calculated with the weight of the solar panel in mind alongside the weight of the supporting structure.

3.2.2.2. System Area

The entire Solar Tracker system will fit within a 20in X 20in horizontal plane with a maximum height of 30 in. This gives a total volume of just under 6 cubed feet.

3.2.2.3. Installation

The installation information for the Solar Tracker shall be provided to the customer through a user manual. All parts will be included and preloaded with necessary operating information.

3.2.3. Electrical Characteristics

3.2.3.1. Inputs

a. The presence or absence of any combination of the input signals in accordance with ICD specifications applied in any sequence shall not damage the Solar Tracker, reduce its life expectancy, or cause any malfunction, either when the unit is powered or when it is not.

b. No sequence of command shall damage the Solar Tracker, reduce its life expectancy, or cause any malfunction.

Rationale: By design, should limit the chance of damage or malfunction by user/technician error.

3.2.3.1.1 Power Consumption

a. The maximum peak power of the system shall not exceed 26.4 watts.
b. This comes from a maximum of 2.4 Watts for the Arduino Uno and 12 Watts for each of the 2 motors.

Rationale: This is a requirement specified to ensure that none of the components receive a current that is too large and will cause it to malfunction.

3.2.3.1.2 Input Voltage Level

- The Arduino Uno Microcontroller should have a voltage input of 7-12V.
- The Motors should both have an input voltage between 6-24 V.

- The Rechargeable Battery should have an input voltage between 14.2-14.6V.

Rationale: Based on manufacturer specifications.

3.2.3.1.3 External Commands

The Solar Tracker shall document all external commands in the appropriate ICD.

Rationale: The ICD will capture all interface details from the low-level electrical to the high-level packet format.

3.2.3.2. Outputs

3.2.3.2.1 Battery Load

- The Solar Tracker shall include a rechargeable battery to act as the load for the system.
- It will have an input voltage between 14.2 and 14.6 V which will need to be stepped up from the Solar Panel's output of 12 V.
- It will also require a maximum of 50 milliamps.

Rationale: The Solar Panel within the system will charge the battery.

3.2.3.3. Connectors

The Solar Tracker shall use external connectors in accordance with MIL-DTL-38999.

3.2.3.4. Wiring

The Solar Tracker shall follow the guidelines outlined in MIL-HDBK-5400 paragraph 4.3.35 Wire and cable.

3.2.4. Environmental Requirements

The Solar Panel shall be designed to withstand and operate in the environments and laboratory tests specified in the following section.

Rationale: This is a requirement specified by our customer due to constraints of the system into which the Solar Tracker is integrating.

3.2.4.1. Thermal

The Solar Tracker shall be designed to withstand temperatures up to 150℉.

Rationale: Solar Panels perform optimally in Standard Test Conditions of about 77 ℉.

3.2.4.2. External Contamination

The Solar Tracker will face external contamination from outside light sources besides the sun. However, photovoltaic sensors have built-in capabilities to differentiate between light from the Sun and other sources.

3.2.4.3. Rain

The Solar Tracker shall not be designed to work optimally in rainy conditions and will not be waterproof. This is due to difficulties around waterproofing motors.

3.2.4.4. Humidity

The Solar Tracker shall be designed to work optimally in humidity conditions of 0 to 50%.

3.2.5. Failure Propagation

The Solar Tracker shall not allow propagation of faults beyond the interface specified below.

3.2.5.1. Failure Detection, Isolation, and Recovery (FDIR)

The Solar Tracker shall use the built-in test daily in order to ensure the system is working optimally.

3.2.5.1.1 Built-In Test (BIT)

The Solar Tracker will have a built-in test that at the beginning of each day the Solar Tracker will rotate 45 degrees on both motors from the day starting position. This will show that the system is working as intended and the user will be able to see as the panel follows the sun throughout the day. In the case of a BIT failure, the user will need to reset the system.

3.2.5.1.1.1 BIT Critical Fault Detection

The BIT shall be able to detect a critical fault in the Solar Tracker by the user seeing the Solar Tracker not facing the sun. This will be very easy to notice if there is a critical fault. If there is only a small miscalculation the solar panel will be facing essentially toward the sun but not directly.

3.2.5.1.1.2 BIT False Alarms

The BIT shall have a false alarm rate for critical errors of less than 5 percent. Meaning the Solar Tracker will be facing the sun essentially all the time during peak sun hours from roughly 10 am - 4 pm.

4. Support Requirements

The Solar Tracker requires close proximity to a wall outlet in order to power the design. Users may choose to use a different load if they desire as the Solar Tracker defaults with a rechargeable battery. The system will work autonomously but users will be provided with a handbook that will provide instructions on troubleshooting common problems that may arise.

Appendix A: Acronyms and Abbreviation

BIT	Built-In Test
CCA	Circuit Card Assembly
GUI	Graphical User Interface
Hz	Hertz
ICD	Interface Control Document
kHz	Kilohertz (1,000 Hz)
LED	Light-emitting Diode
mA	Milliamp
MCU	Microcontroller
MHz	Megahertz (1,000,000 Hz)
W	Watt
mW	Milliwatt
PCB	Printed Circuit Board
RMS	Root Mean Square
TBD	To Be Determined
TTL	Transistor-Transistor Logic
USB	Universal Serial Bus
V	Voltage
I	Current
in	Inches
ft	Feet
lbs	Pounds

Appendix B: Definition of Terms

MPPT - A Maximum Power point Tracker that will output the ideal voltage and current to maximize power output.

Photosensitive Sensors - Sensors that detect light. Information from these sensors will be read by Arduino.

Solar Tracker

Osman Farook
Talha Zaheer
Cooper Hamlin
Alexandra Torres

Interface Control Document

REVISION – 4
12/04/2023

INTERFACE CONTROL DOCUMENT
FOR
Solar Tracker

PREPARED BY:

____
Author Date

APPROVED BY:

________Cooper_Hamlin_________
Project Leader Date

____
John Lusher II, P.E. Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	2/19/2023	Cooper Hamlin		Draft Release
1	04/28/2023	Cooper Hamlin		403 Final Report
2	12/03/2023	Cooper Hamlin		404 Final Report

Table of Contents

List of Tables

List of Figures

1. Overview

This document will provide details on how the 3 subsystems of the Solar Tracker design will mesh. It will list all the physical descriptions of the parts utilized within the device and the characteristics associated with each of them. This document will also reference the goals and requirements mentioned within the ConOps report and the FSR document and how the design will go about reaching those parameters.

2. References and Definitions

2.1. References

MIL-HDBK-1003-19
DESIGN PROCEDURES FOR PASSIVE SOLAR BUILDINGS

MIL-STD-454 - Electronic Equipment,Standard General Requirement

MIL-STD-461 - Electromagnetic Emission and Susceptibility Requirements for the Control of Electromagnetic Interference

MIL-P-24764 - Power Supplies,

2.2. Definitions

rpm	Rotations per Minute
MCU	Main Control Unit
V	Voltage (Volts)
I	Current (amps)
lbs	Pounds
in	Inches
mA	Milliamp
mW	Milliwatt
MHz	Megahertz
Kg/cm	Kilogram per Centimeter
MPPT	Maximum Power Point Tracker

DC	Direct Current
AC	Alternating Current

3. Physical Interface

3.1. Weight

The desired weight of the Solar Tracker shall be less than 30 lbs.

Component	Weight(lbs)
Solar Panel	2.6
Battery	2.2
Motor	0.35x2
Arduino MCU	0.055
Structure	20

Table 1: Weight of the Design

3.2. Dimensions

Component	**Dimension(inches)**
Solar Panel	11.5 x 11.5 x 0.9
Battery	5.94 x 2.56 x 3.7
Motor	4.29 x 2.76 x 2.2
Arduino MCU	2.7 x 2.1 x 0.4
Structure	20 x 20 x 25

Table 2: Dimensions of the Design

3.3. Mounting Locations

The Solar Tracker should be deployed anywhere with limited space or maximum output needed. This means locations suggested in our ConOps report Scenarios section would be perfect locations. The benefit of such a productive solar panel tracker allows for uses in a variety of scenarios and situations.

3.3.1: Placement of the Design:

The Solar Tracker will need to be placed on a flat section of the land with plenty of available sunlight to maximize the efficiency of the solar panel.

3.3.2: Mounting of the Panel:

The Solar Panel will be housed at an elevated position from the base and will be supported by 2 metal rods. The housing will allow a Solar Panel with the dimensions 20 x 20 x 25.

3.3.3: Mounting of the Photovoltaic Sensors:

The Sensors will be mounted onto the edges of the solar panel housing structure. This will allow for the tracker to get a full view of the sun from each direction and will allow for the tracker to position the panel such that each sensor is receiving similar photon levels.

4. Thermal Interface

4.1. Thermal Shielding for all electrical components

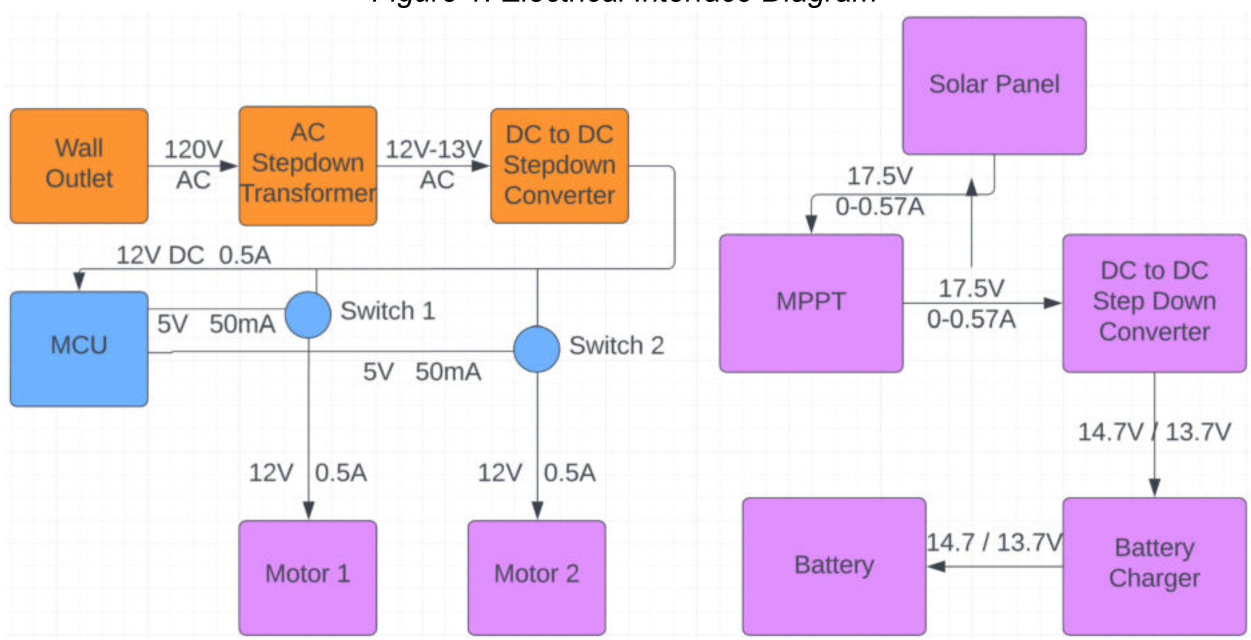
The electrical components will be covered by a housing unit to keep all electrical components out of the sun's rays and all components should be able to sustain temperature for sufficient time frames.

4.2. Mounting of the Panel:

The Battery will be air-cooled under the solar panel. The heat that is generated from the battery charging will be able to dissipate on its own accord.

5. Electrical Interface

Figure 1: Electrical Interface Diagram



5.1. Primary Input Power

5.1.1: Wall Outlet

The power for the system will be provided by a wall outlet. The voltage through a wall outlet is 120V AC at 60 Hz and thus will need to be converted to DC in order to power the microcontroller and the motors for the system. The rectifying circuit will have 3 key parts:

5.1.2: Transformer

In order to make it safer to handle the incoming voltage from the wall outlet, a transformer will be used. That will allow the 120V AC input to be stepped down to 12 V through the secondary windings. This voltage will then be used in the circuit to be rectified.

5.1.3: Diode Rectifier

The 12 V AC voltage will then be passed through a rectifier containing 4 diodes. 2 of the diodes will be forward biased allowing only the positive half of the AC voltage to go through while the other 2 diodes will be reverse biased which will prevent the other half from flowing through. This will allow the AC voltage to be converted into DC.

5.1.4: Filter and Voltage Regulator

The output voltage from the rectifier will then pass through a capacitor which will allow any ripple from the DC voltage to be smoothed out. This is important because excess ripple can damage the microcontroller which would be detrimental to the system. The smoothed-out voltage would then feed into a voltage regulator which would output a constant 12 V. That voltage would also pass through a resistor which would allow the necessary current of 0.5A to be achieved.

5.2. MCU Switch

We will have two switches controlled by the MCU that will allow and constrict current flow to the two motors. This will be how the MCU controls how much the Solar Tracker will spin horizontally and how far the solar panel will tilt.

5.3. Battery Load

5.3.1: Solar Panel and MPPT

The Maximum Power Point Tracker or MPPT will be used to control the output of the Solar Panel and maximize the energy being fed to the DC-DC converter at an average of 17.5 Volts and 57 milliAmps

5.3.2: DC-DC Step Down Converter

A DC-DC Buck converter will be used to tailor to the specifications of the battery. The 17.5V input from the solar panel will be stepped down to around 14.7 Volts(for phase 1) or 13.7 Volts(for phase 2) which will then be input into the battery.

6. Software & Microcontroller Interface

6.1. Photo-Sensors (MCU)

We will be using 4 strategically placed 220 ohm light dependant resistors around the perimeter of the solar panel to monitor light levels and direction of intensity. This information will be communicated to the Microcontroller which in our case is the Arduino REV 3.

6.2. Arduino UNO REV 3

6.2.1: Sensors (MCU)

There are 6 Analog to digital converters each having a 5 Volt limit. The data from the Photosensors will be converted to digital signals and the Arduino Uno will use this information to decide where there is more light intensity, instructing the motors to assume a more favorable position.

6.2.2: Battery Charger (Load)

The battery charger will take both the input voltage from the DC-DC converter and output voltage of the converter to provide feedback and update the output voltage to the targeted output voltage. This is necessary as the charger will prevent overcharging and will create a more efficient charging setup to improve longevity of the battery. This is accomplished by using dual stage constant voltage modes. Phase 1 of the battery charger uses an output of 14.7 Volts. Likewise, phase 2 will use 13.7 as the target voltage. Finally, phase 3 will disconnect the battery from the ground and the battery will end charging.

6.3. Graphical User Interface

6.3.1: Sensors (MCU)

There is 1 Analog to digital converter sensor, the voltage sensor,to be converted to digital signals and the ESP 32 WROOM will use this information to gather voltage readings from the load subsystem. There are two I2C sensors, an MPU6050 Gyroscope and INA219 Current sensor, that the ESP 32 will also be reading-in data from.

6.3.1: Website

A website will be developed to display readings obtained from the ESP 32 WROOM. The data will be transmitted from the ESP32 to a SQL database, where the website will retrieve the database information and output the data into a comprehensive website.

Solar Tracker

Osman Farook
Talha Zaheer
Cooper Hamlin
Alexandra Torres

MILESTONES AND VALIDATION PLAN

REVISION – 2
12/04/2023

Milestones:

Work	End Date	Owner	Status
Validation of 403 PCB’s	8/30/2023	TALHA, COOPER	DONE
Finalize GUI plans	8/30/2023	Alexandra	DONE
Begin Mechanical Design	9/7/2023	TALHA, COOPER	DONE
Motor Controller PCB design	9/7/2023	OSMAN	DONE
Database design and server setup	9/14/2023	ALEXANDRA	DONE
Finalize Mechanical Design Idea	9/14/2023	TALHA, COOPER	DONE
Begin Enclosure Design	9/21/2023	TALHA, COOPER	DONE
Photosensor layout finalized	9/21/2023	OSMAN	DONE
API Development	9/28/2023	ALEXANDRA	DONE
Finish Enclosure Design	9/28/2023	TALHA, COOPER	DONE
Finish Solidworks Worm Gear and Screw Design	10/5/2023	TALHA, COOPER	DONE
Order Buck Boost PCB	10/5/2023	COOPER	DONE
3D Print Prototype design	10/12/2023	TALHA, COOPER	DONE
Backend integration	10/12/2023	COOPER	DONE
MCU-Power Integration	10/19/2023	OSMAN, TALHA	DONE
Solder and validate PCB	10/19/2023	COOPER	DONE
GUI and Load Integration	10/26/2023	ALEXANDRA, COOPER	DONE
Worm screw and gear	10/26/2023	TALHA, COOPER	DONE

3D print order submitted			
PCB Design Ordered	11/2/2023	OSMAN	DONE
Fabrication of design started	11/2/2023	COOPER, TALHA	DONE
RTK Query, Data fetching.	11/9/2023	ALEXANDRA	DONE
3D printing supporting pieces	11/9/2023	COOPER, TALHA	DONE
Machine Shop fabrication	11/16/2023	COOPER	DONE
Finalized GUI	11/16/2023	ALEXANDRA	
PCB Soldered	11/27/2023	OSMAN	DONE
Final Demo testing	11/28/2023	ALL	DONE
Final Demo	11/30/2023	ALL	DONE

Validation Plan:

Paragraph #	Test Name	Success Criteria	Methodology	Owner
3.3.2	Mounting Size	Fits within a 20x20x25in square volume	use tape measure to verify dimensions	ALL
3.3.3	Sensor Mounting	Sensors are securely mounted on solar panel peripheral and can effectively receive direct sunlight	Observe sensory data for reflecting change in physical conditions	Osman
4.4.1	PCB multiple output	Inputs 12 VDC to a perfboard allowing for multiple connections	use multimeter to validate input voltage	Talha
4.1.2	Power to MCU	Arduino is powered through 12 VDC	Arduino works as intended	Talha
4.1.3	Power to Motor Controller	Stepper motor controller is powered through the 12 VDC	Motor controller works as intended	Talha
4.2	MCU Switch	MCU successfully controls Motor driver for instructions	motors turn on when instructions are sent	Osman
4.3.2	Moving Solar	Steps down voltage from Solar Panel to	use multimeter to	Cooper

	Panel to Buck-Boost Converter	between 14.4 and 15V required for battery with sufficient current +/-5%	validate output voltage and changing input from panel movement	
4.3.3	Battery Charger to GUI	Successfully traverses between the two phases of battery charging	use multimeter to insure Battery shut of and GUI Input	Cooper
4.3.3	Battery Status Display	Send battery status information to GUI	Ensure observations in Arduino Display	Cooper
5.1	Photo-Sensors	Voltage change is communicated to Arduino through Circuitry	build test circuit in simulated environment	Osman
5.2	Arduino Uno REV 3	New code must be written for entire project since switch to stepper	write test code for validation data	Osman
6.1	GUI	Usability / Testing buttons, elements, drop-downs overall Website Functionality	Clicking through GUI, observing response of GUI	Alexandra
6.2	GUI	Data collection accuracy	comparing data in database to sensor readings	Alexandra
6.3	GUI	Graphs Update within website	comparing the data in website to database	Alexandra
N/A	Full System	System is at least 20% more efficient than a fixed tilt design	compare energy generated from tracker vs fixed tilt design	ALL

Solar Tracker

Osman Farook
Talha Zaheer
Cooper Hamlin
Alexandra Torres

SUBSYSTEM REPORT

REVISION – 2
12/04/2023

SUBSYSTEM REPORT
FOR
Solar Tracker

PREPARED BY:
Team <41>

____
Author Date

APPROVED BY:

__Cooper_Hamlin__________12/04/23
Project Leader Date

____
John Lusher II, P.E. Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	[4/26]	Team 41		403 Final Report
1	[12/04]	Cooper Hamlin		404 Final Report

Table of Contents

List of Tables

List of Figures

1. Introduction

The Solar Tracker system will take into account 4 major subsystems that will allow for its functionality. The system will take power from a wall outlet and rectify the voltage to power both the Arduino and the motors. The Arduino rev 3 will then take in information from the photosensors which will allow it to move the motors based on the position of the Sun. The power from the Solar Panel will then go through and charge the battery with efficiency through a buck converter and instructions from the microcontroller. The GUI will take in gyroscope, voltage, and current readings to determine battery levels, as well as solar panel orientation.

2. Power Subsystem Report

2.1. Introduction

The power subsystem consisted of multiple components that would work together in order to convert the 120V AC input from the wall outlet into 12 V DC at 0.5 Amps to power both the microcontroller and the motors within the system. The components included a Transformer, a diode bridge, a capacitor filter, a resistor for the desired current, and a voltage regulator.

2.2. Subsystem Details

The rectifying circuit used for the system consisted of multiple parts. Firstly, an AC transformer was used to step down the voltage from the wall outlet from 120V AC to 12 V AC. The transformer works by changing the voltage through primary and secondary windings. The use of coils within the transformer specifically allows the power to be stepped down. For the purposes of the system, a 10:1 ratio of coils was needed to convert to the desired voltage. The transformer would plug into the wall outlet through the wires on the primary end which were tied around a polarized plug while the wires on the secondary end would output the stepped-down voltage of 12V AC.

Once that was done, the voltage was input into a diode bridge consisting of 4 diodes. 2 of those diodes were forward-biased while the other 2 were reverse-biased. This would serve to rectify the AC voltage by only allowing the positive half of the sine wave to flow through which would allow for the creation of the DC voltage.

Once the voltage passed through the bridge and became DC, a 220 microFarad capacitor was attached to serve as a filter. This would allow the voltage to have less ripple and be more smooth when being used by either the microcontroller or the motors.

This was important as excess ripples can cause components to malfunction and potentially ruin the entire system.

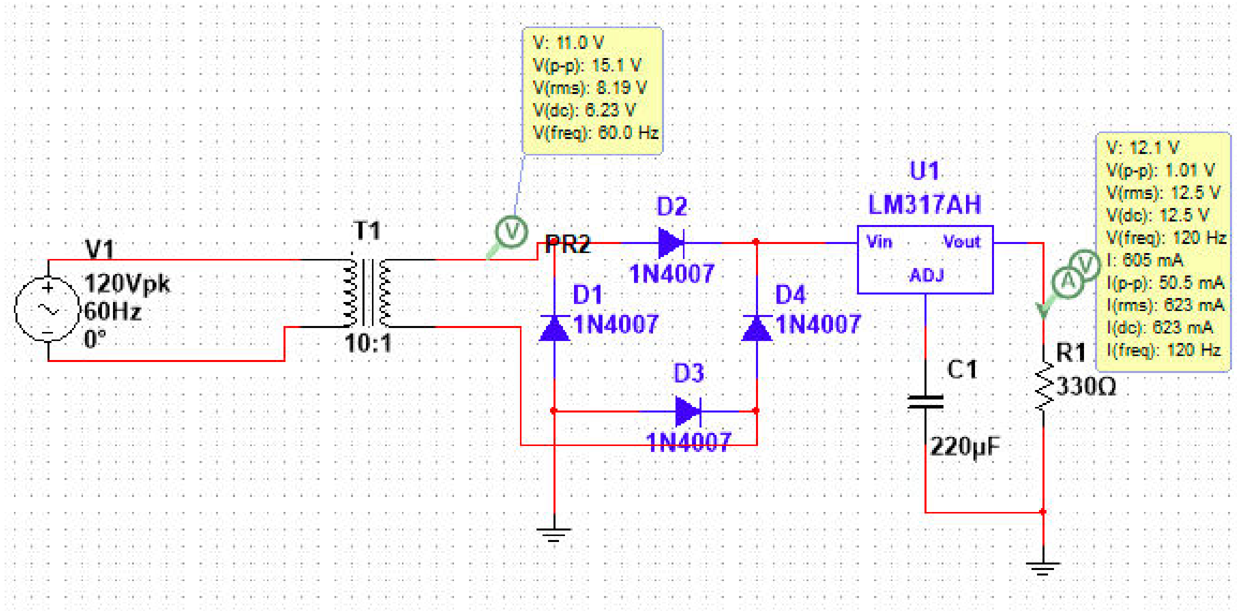
The smoothed-out voltage was then input into a 330-ohm resistor which was necessary to pull some of the currents in order to get to the ideal current of around 0.5 Amps. Finally, the voltage was input into a voltage regulator which allowed the circuit to output a constant voltage of 12 V which was in line with the requirements for both the Arduino Uno as well as the L298 stepper motor controllers being used to move the solar panel.

In 404, the majority of the time for the power subsystem was spent on the creation of the mechanical design of the system itself. As the pcb had been finished and the overall subsystem progress was ahead of the rest of the team members for a majority of the semester, the initiative to learn Solidworks was taken. It was a steep learning curve but one that was necessary for the sake of the team as the proposed design would require numerous parts. Over 50 total parts were printed over the course of the semester which included the final parts of the worm screw and the worm gear. There was a lot of testing done with printed prototypes in order to finalize the design. The files for some of the key parts can be found on the github page.

2.3. Subsystem Validation

Testing for this power subsystem began with Multisim and the creation of multiple designs that could possibly achieve the requirements. While keeping in mind that the budget for this project needed to be conserved, the circuit was designed as mentioned above and simulated with the required values. Figure 1 shows the Multisim simulation.

Figure 1: Power Supply Simulation



Once the simulations in Multisim were complete, the circuit was designed in Altium in order for a physical PCB to be designed. Figures 2 and 3 show the Altium designs. The schematic and design were completed and a number of changes had to be made throughout the entire process to ensure that the design met the criteria to be manufactured and passed all the different checks. Once this process was complete, the PCB was sent to be ordered.

Figure 2: Rectifier Altium Schematic

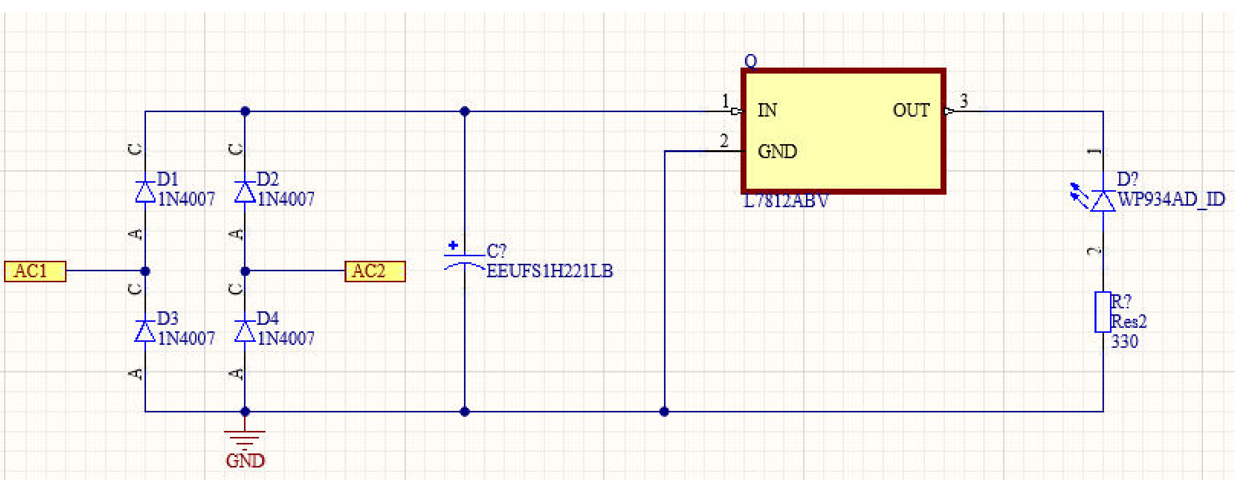
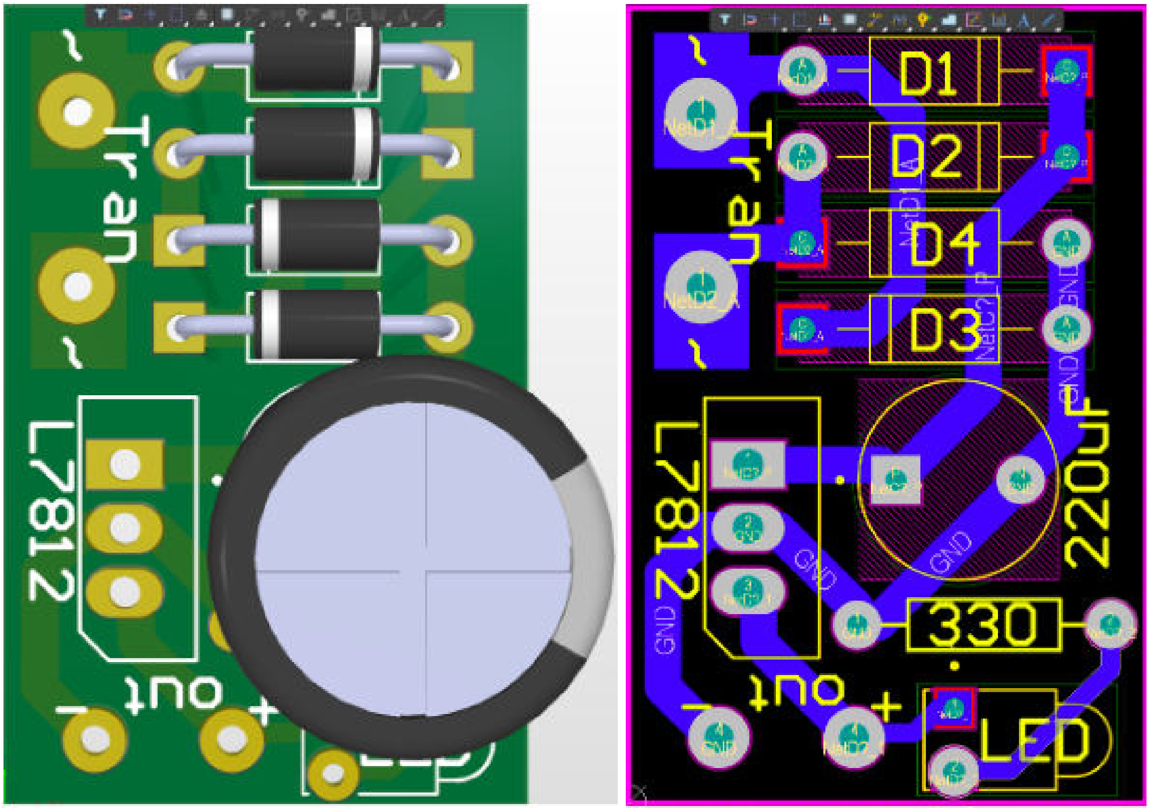
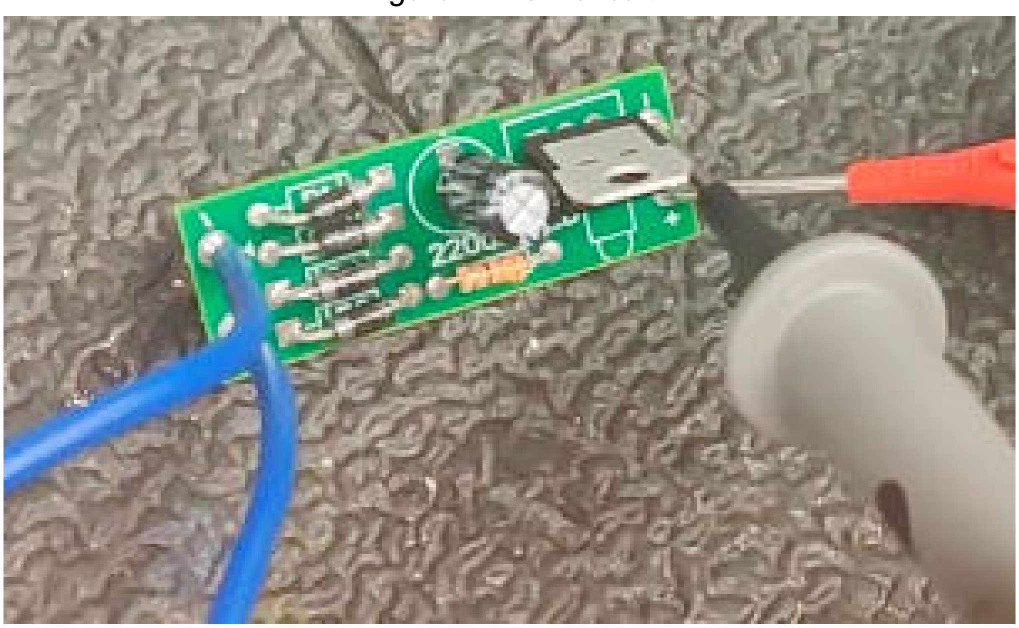


Figure 3: Rectifier Altium 3D and Design View



Once the PCB was ordered, all the parts were soldered which included the diodes as well as the capacitor, and the voltage regulator. Figure 4 shows the completed physical circuit being tested.

Figure 4: PCB circuit



The validation of the circuit began with making sure the wall outlet was inputting 120V AC to the transformer. This was an important step as the first few wall outlets were not inputting the correct voltage.

Once the wall outlet was connected, the next step was to ensure that the transformer was stepping down to 12 V. This was an essential step as the components within the PCB were not rated to work for more than 100V and required a lower voltage. The transformer was measured to output around 12-13V with a lot of ripple which would necessitate the use of a capacitor later within the circuit. The output from the transformer was fed into the actual circuit.

Then the final piece of validation was to check whether the circuit was outputting the 12V 0.5 Amps. The multimeter showed that the system was outputting 12.14 V at around 0.54 Amps which fit within the tolerance value of +/- 5%. The entire validation is shown in Table 1.

Table 1: Power Subsystem Validation

4.4.1	Wall Outlet Correct Output	Inputs 120V to the circuit	use multimeter to validate input voltage	117.5 V
4.1.2	AC Step Down	AC Voltage is stepped down to 12V +/-5%	use multimeter to validate output voltage	12.2 - 13.1 V
4.1.3	AC to DC Rectifier	Output from the rectifier is 12V +/-5%	use multimeter to validate output voltage	12.1429V

Figure 5 shows the voltage from the circuit over time without any load in order to make sure that the output is consistent. The graph shows the lack of ripple within the output from the circuit which was an important detail considering that the voltage would power both the microcontroller as well as the motor controllers within the system. Figure 6 shows the output of the PCB with both the motor controllers and the Arduino connected. There is a slight voltage drop but the output remains consistent which was the goal.

Figure 5: Circuit Output without Load

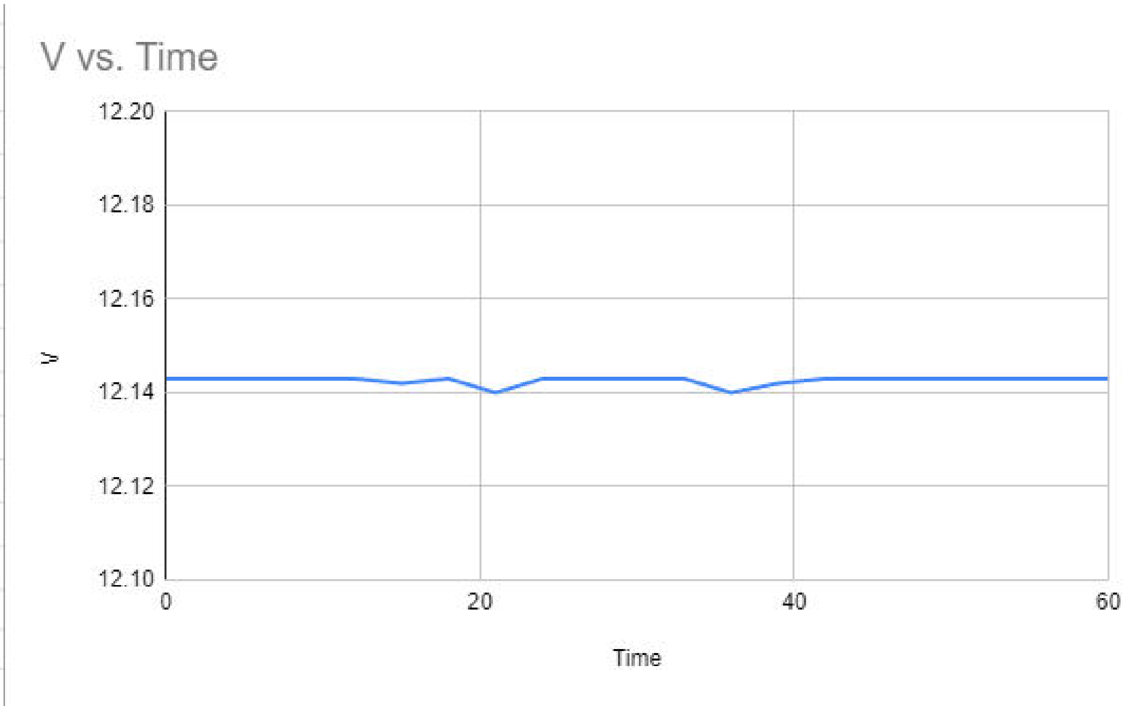
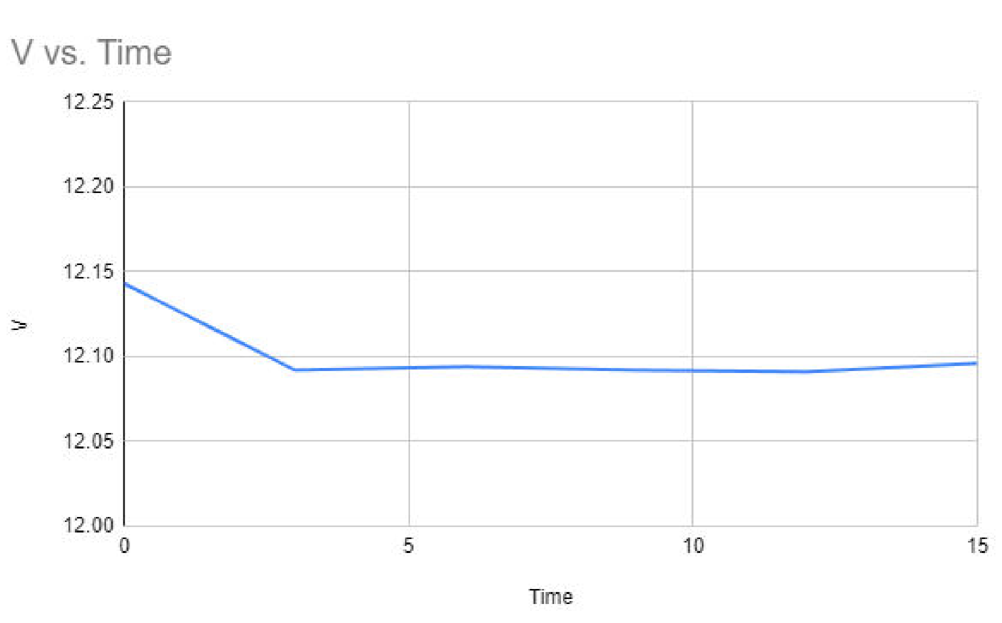


Figure 6: Circuit Output with Load



The validation at the time of integration came through testing the power system pcb with the Arduino uno and the L298 stepper motor controllers at the same time. The 12 VDC would be fed into the 3 components within the system. The Arduino would then provide the 5V signal to the motor controllers which would allow for the movement of the motors for a specific orientation based on the information captured by the light dependent resistors.

The overall subsystem was successfully able to power both components and the motors were able to move after running the Arduino code. That video was captured and put on the github link. Unfortunately, one of the ICs within the PCB burnt out two days before demo day. It was very frustrating to have this happen so close to the proximity of the demo as the IC that was burnt, which was the voltage regulator, was not able to be reordered. Given the nature of the testing that had picked up in intensity over the last few weeks before the demo, it is likely that the IC burnt out due to that fact. If this had happened even two weeks before the demo, it would have been possible to reorder the specific regulator and resolder the new part on. Thus the validation on demo day had to be proven through the numerous videos and graphs that had been acquired over the course of the semester.

2.4. Subsystem Conclusion

Overall, the power subsystem achieved all of the initial requirements and goals that were set when the design process started. There were a number of different pathways to achieving the end goal of outputting the required voltage of 12 V but this circuit was chosen with the budget and time frame in mind. Given that there were a number of mechanical parts to be purchased and printed, it was difficult to improve upon the efficiency of the circuit itself and considering the fact that it was working as intended and able to power the Arduino and the motor controllers, the power subsystem was integrated into the design. Unfortunately, the IC burning two days before the demo prevented the live demo from happening.

3. MCU Subsystem Report

3.1. Introduction

The MCU subsystem is responsible for the acquisition of photosensory data, which is fed in to a microcontroller (Arduino Uno Rev 3), and taken as input by the code to process and generate motor controller instructions to position the solar tracker mechanism in the optimum orientation for maximum sunlight coverage.

3.2. Subsystem Details

The subsystem can be broadly divided into three distinct phases. Acquisition of sensory data, data processing, and motor control movement.

Figure 7: LDR schematic

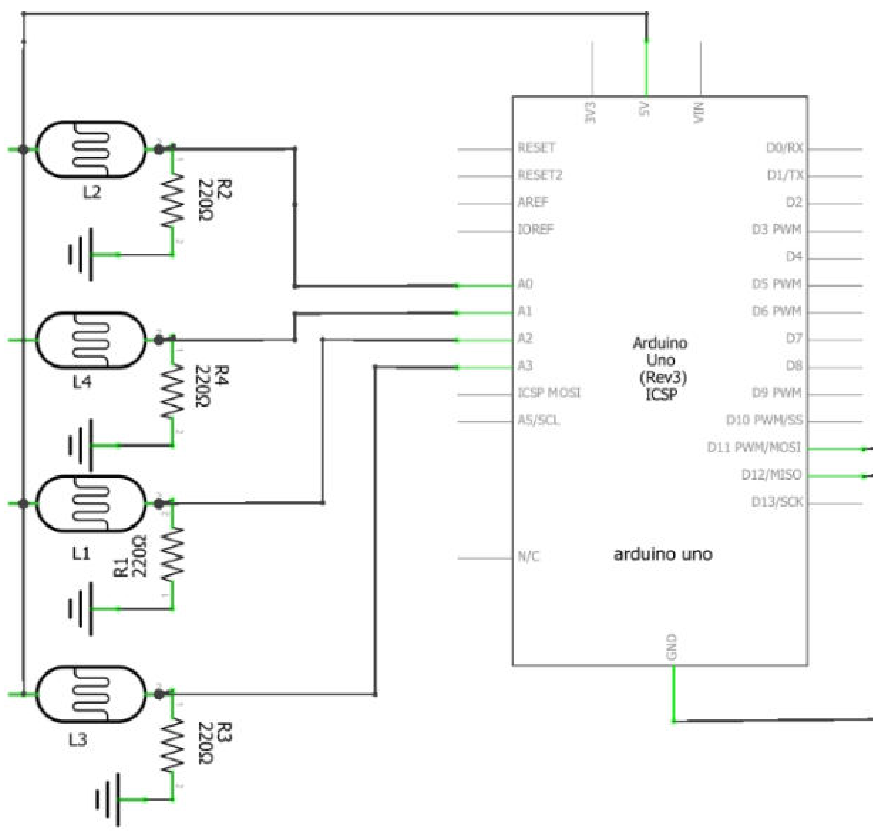


Figure 7 depicts a schematic of the LDR connections to the Arduino and Figure 8 shows the practical physicality of the circuit designed to be mounted on the mechanism.

3.2.1 Photo Sensor Acquisition

Circuit Configuration: In our solar tracking system, four Photoresistor GL5516 LDR (Light Dependent Resistor) sensors are integrated into a voltage divider circuit configuration, powered by a consistent 5V supply sourced from the Arduino Uno Rev3. Each LDR is connected in series with a fixed resistor. The resistors employed are of 10k ohms, a value meticulously selected to effectively complement the LDRs' response characteristics under standard lighting conditions.

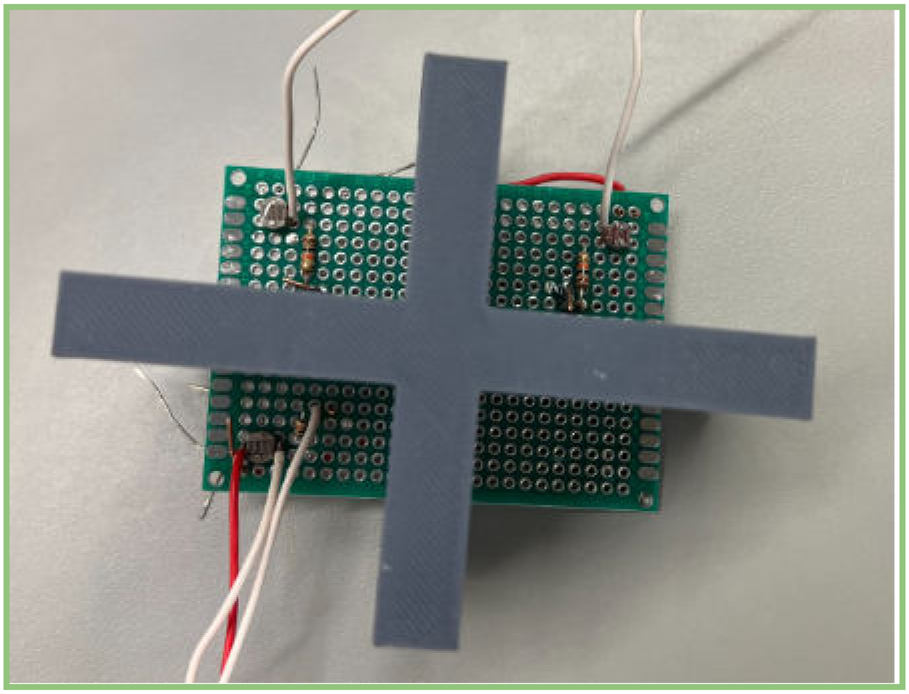
Selection Rationale: The 10k ohm resistors were chosen based on empirical tests conducted under typical ambient light conditions. This resistance value ensures an optimal voltage divider operation, where the voltage drop across the LDR varies proportionally and significantly with changes in light intensity. This setup allows for a more distinct and measurable response from the sensors, enhancing the accuracy of light intensity detection.

Noise reduction:.A significant enhancement to this setup is the addition of a '+' shaped barrier among the LDRs. This barrier is essential in isolating each sensor from extraneous light from other directions, addressing the issue of erroneous data due to cross-illumination. Initially, without the barrier, adjacent light sources impacted the LDRs, leading to inaccurate readings. This compromised the system's precision in determining the direction of the highest light intensity. The introduction of the barrier effectively shields each LDR, allowing it to respond only to light from its assigned direction. This adaptation has markedly improved the precision of the data, significantly reducing noise in the sensor readings.

Signal Acquisition: Between each LDR and its corresponding fixed resistor, I have positioned input channels that are linked to the analog input pins of the Arduino. These channels are critical for capturing the dynamic changes in electrical characteristics caused by varying light conditions. The resistance of each LDR inversely varies with light intensity – as light levels increase, the resistance decreases, and vice versa. This variation alters the voltage drop across each LDR, which is precisely measured by the Arduino's ADC (Analog-to-Digital Converter).

Data Processing Relevance: The voltage readings obtained from these points are pivotal for the data processing phase of our system. They provide real-time, quantitative insights into the light intensity at different angles relative to the solar panel. This data is fundamental for the algorithm that computes the optimal orientation of the solar panel to maximize solar energy acquisition. By continually adjusting the panel's position in response to the real-time sensor data, the system ensures maximum efficiency in solar energy collection throughout the day.

Figure 8: Physical LDR Circuit



3.2.1 Sensor Data Processing

Sensor Calculations: The solar tracking subsystem's core functionality hinges on processing the resistance data acquired from an array of Light Dependent Resistors (LDRs). These LDRs, integral in determining the solar panel's orientation, change their resistance based on the received light intensity. Each LDR's resistance and voltage drop are calculated by positioning them in a voltage divider setup, paired with a fixed 10k ohm resistor. The resistance is deduced from the voltage drop across the LDR, which is computed by scaling the analog reading (ranging from 0 to 1023) to the Arduino's 5V reference voltage.

Figure 9: LDR Data Processing

```
void loop() {
  // Read current LDR values
  int ldr_value_1 = analogRead(ldr1);
  int ldr_value_2 = analogRead(ldr2);
  int ldr_value_3 = analogRead(ldr3);
  int ldr_value_4 = analogRead(ldr4);

  // Calculate the resistance and voltage drop of each LDR
  float resistance_1 = (float)(1023 - ldr_value_1) * 10000 / ldr_value_1;
  float voltage_drop_1 = (float)ldr_value_1 * 5 / 1023;

  float resistance_2 = (float)(1023 - ldr_value_2) * 10000 / ldr_value_2;
  float voltage_drop_2 = (float)ldr_value_2 * 5 / 1023;

  float resistance_3 = (float)(1023 - ldr_value_3) * 10000 / ldr_value_3;
  float voltage_drop_3 = (float)ldr_value_3 * 5 / 1023;

  float resistance_4 = (float)(1023 - ldr_value_4) * 10000 / ldr_value_4;
  float voltage_drop_4 = (float)ldr_value_4 * 5 / 1023;
```

Figure 10: Analog to Digital Signal Code

```
void loop() {
  int ldr1_value = analogRead(LDR1_PIN);
  int ldr2_value = analogRead(LDR2_PIN);
  int ldr3_value = analogRead(LDR3_PIN);
  int ldr4_value = analogRead(LDR4_PIN);

  // convert analog values to voltage values
  float ldr1_voltage = ldr1_value * (5.0 / 1023.0);
  float ldr2_voltage = ldr2_value * (5.0 / 1023.0);
  float ldr3_voltage = ldr3_value * (5.0 / 1023.0);
  float ldr4_voltage = ldr4_value * (5.0 / 1023.0);

  // print voltage values to serial monitor
  Serial.print("LDR 1 Voltage: ");
  Serial.print(ldr1_voltage);
  Serial.print("V | Resistance: ");
  Serial.print((5.0-ldr1_voltage)/ldr1_voltage * 10e3);
  Serial.print("kOhm\n");
```

Resistance Change Comparison: The system's intelligence is exhibited in its ability to discern significant fluctuations in light intensity. These fluctuations are translated into resistance changes in the LDRs. A threshold of 300 ohms is established to differentiate meaningful changes in light conditions from nominal environmental noise. This threshold acts as a decision parameter to ensure the system's responsiveness is limited to substantial light intensity variations, thereby stabilizing its operation against minor perturbations.

Figure 11: Threshold Checks

```
  // Check if any LDR resistance has changed more than the threshold
  if (change_resistance_1 > threshold || change_resistance_2 > threshold || change_resistance_3
    // Resistance values have changed significantly; move motors
    // Define the logic for when and how to move the motors here
    // For now, let's rotate both motors forward as an example
    rotateMotor(motorXPin1, motorXPin2, motorXPin3, motorXPin4, true);
    rotateMotor(motorYPin1, motorYPin2, motorYPin3, motorYPin4, true);
    Serial.println("Motors moving forward");
  } else {
    // Resistance values have not changed significantly; do not move motors
    Serial.println("No significant change in resistance, motors are stopped.");
  }

  delay(1000); // Delay for a moment before next reading
}
```

Figure 12 : LDR Comparison

```
    // adjust motor direction based on LDR values
    if (ldr1_value < ldr4_value && ldr2_value < ldr4_value && ldr3_value < ldr4_value) {
      digitalWrite(MOTOR_PIN1, HIGH);
      digitalWrite(MOTOR_PIN2, LOW);
    } else if (ldr1_value > ldr4_value && ldr2_value < ldr4_value && ldr3_value < ldr4_value) {
      digitalWrite(MOTOR_PIN1, LOW);
      digitalWrite(MOTOR_PIN2, HIGH);
    } else if (ldr1_value < ldr4_value && ldr2_value > ldr4_value && ldr3_value < ldr4_value) {
      digitalWrite(MOTOR_PIN1, HIGH);
      digitalWrite(MOTOR_PIN2, LOW);
    } else if (ldr1_value < ldr4_value && ldr2_value < ldr4_value && ldr3_value > ldr4_value) {
      digitalWrite(MOTOR_PIN1, LOW);
      digitalWrite(MOTOR_PIN2, HIGH);
    }
  }
```

Motor Control Logic: Upon detecting resistance changes exceeding the set threshold, the system executes the rotateMotor function for each stepper motor. This function, pivotal in directing the motors, accepts parameters that define the rotation direction. It performs a stepping sequence, enabling precise, incremental rotation of the motors. The function iterates over a predefined number of steps, typically set to 200 for a full revolution, toggling the state of the digital pins to activate the motor coils. The duration of each step, controlled by a delay within the function, dictates the rotation speed.

Feedback and System Responsiveness: The subsystem provides real-time feedback by printing the LDRs' resistance and voltage values to the serial monitor. This feature not only aids in monitoring the system's current state but also in debugging and calibrating the sensors. If the resistance change is below the established threshold, the motors are instructed to remain stationary, preventing unnecessary adjustments. This decision is communicated through the serial monitor, ensuring the user is apprised of the system's status.

Figure 13: LDR feedback

```
// Print the resistance and voltage drop values to the serial monitor
Serial.print("LDR 1 - Resistance: "); Serial.print(resistance_1); Ser
Serial.print("LDR 2 - Resistance: "); Serial.print(resistance_2); Ser
Serial.print("LDR 3 - Resistance: "); Serial.print(resistance_3); Ser
Serial.print("LDR 4 - Resistance: "); Serial.print(resistance_4); Ser
```

In summary, the solar tracking subsystem's design capitalizes on the precision and reliability of stepper motors, controlled by a sophisticated algorithm that processes LDR data. This setup ensures that the solar panel is consistently aligned for optimal solar energy capture, effectively tracking the sun's trajectory with minimal energy expenditure. The system's adaptability and precision in response to changing light conditions underscore its efficacy in maximizing solar energy utilization.

3.2.3 Motor Control Implementation

In our dual-axis solar tracker system, precise and reliable motor control is crucial for the accurate positioning of the solar panel. This is why the most significant change from 403 to 404 was the switch from DC motors to stepper motors, enhancing control precision and directional accuracy. Stepper motors provide exact positioning without the need for feedback systems, a vital feature for aligning the solar panel optimally with the sun.

This motor control enhancement is achieved through two distinct mechanisms. The first involves the use of a commercially available Qunqi L298N Motor Drive Controller Board Module, interfacing with an Arduino to control a bipolar stepper motor for the X-axis movement. The second mechanism revolves around a custom-designed stepper motor driver PCB, developed using Altium Designer, to manage the Y-axis movement.

L298N Motor Driver for X-Axis Movement: For the X-axis movement, which facilitates the horizontal tracking of the solar panel, we employ the L298N Motor Drive Controller. This dual H-bridge motor driver is renowned for its efficiency in driving bipolar stepper motors. The motor in question, positioned at the base of the setup, is a Stepper Motor Nema 17 rated at 12V and 0.4A per phase, with a 1.8° step angle. This motor's specifications are ideal for the fine-grained movement required in tracking the sun's X-axis trajectory. The Arduino delivers control signals to the L298N driver, which in turn precisely maneuvers the bottom gear to ensure smooth and accurate horizontal positioning of the solar panel.

Motor Controller PCB for Y-Axis Movement: For the Y-axis movement, controlling the vertical orientation of the solar panel, a stepper motor of identical specifications is utilized. I took heavy inspiration from the Pre built stepper motor driver, and created my own schematic to be engineered on Altium Designer specifically for this application. This custom PCB offers tailored control features and is supposed to be fixed along the pillar mounting the solar panel, allowing for direct, efficient manipulation of the panel's vertical angle via an axle rod. The Arduino, through sophisticated algorithms, sends optimized movement commands to this driver, ensuring that the solar panel is always optimally aligned with the sun's elevation.

Figure 14: l298N PCB Motor Driver Schematic

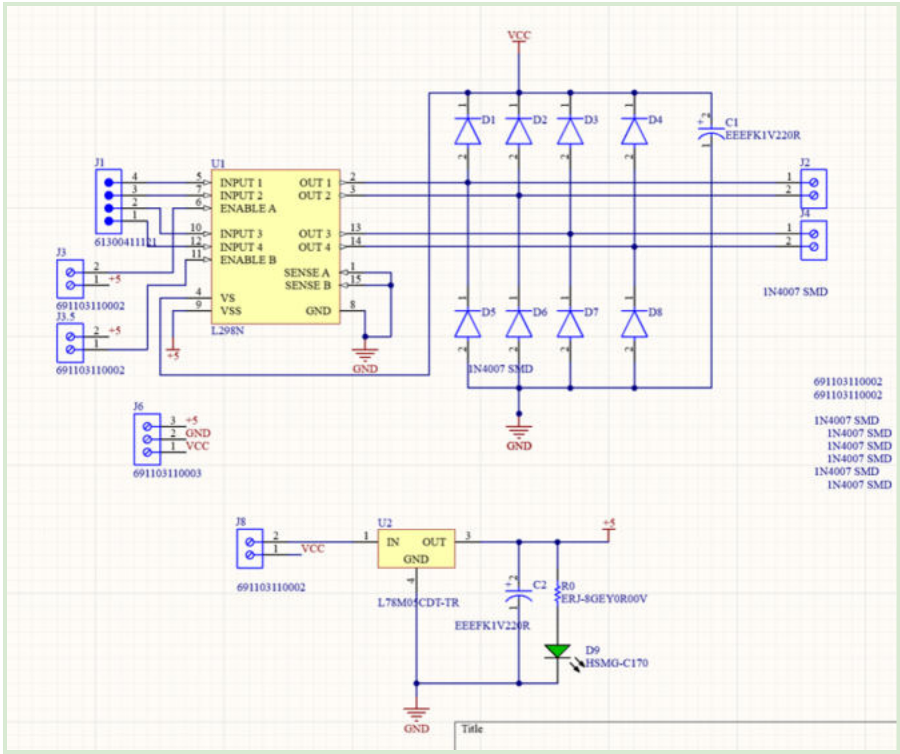


Figure 15: PCB 3D View

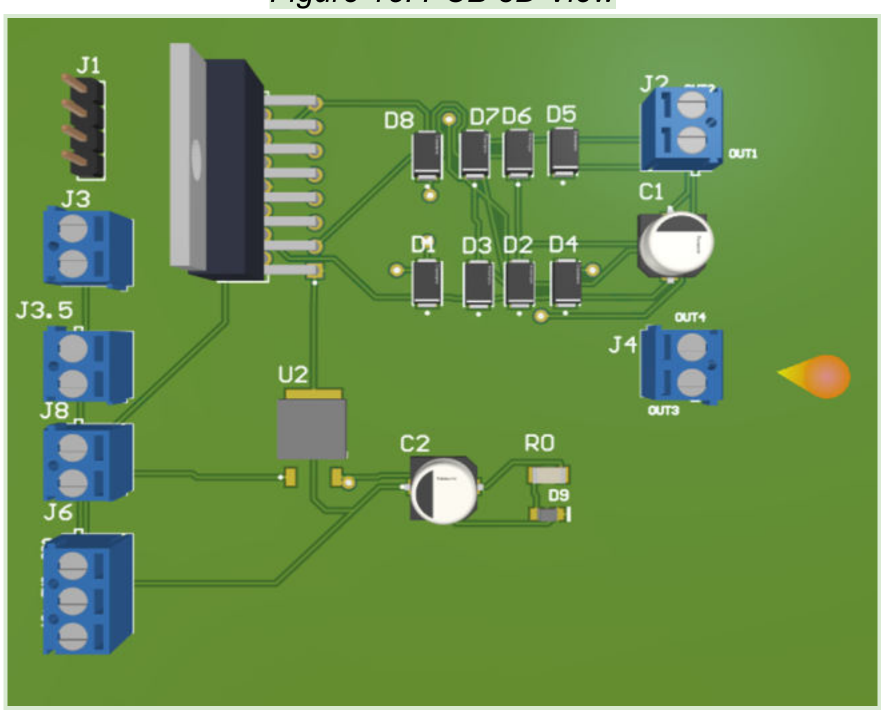


Figure 16: Physical PCB Motor Driver (Soldered)

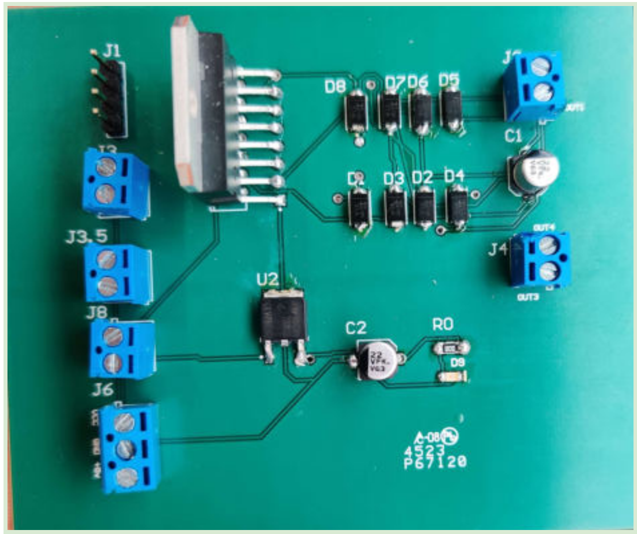
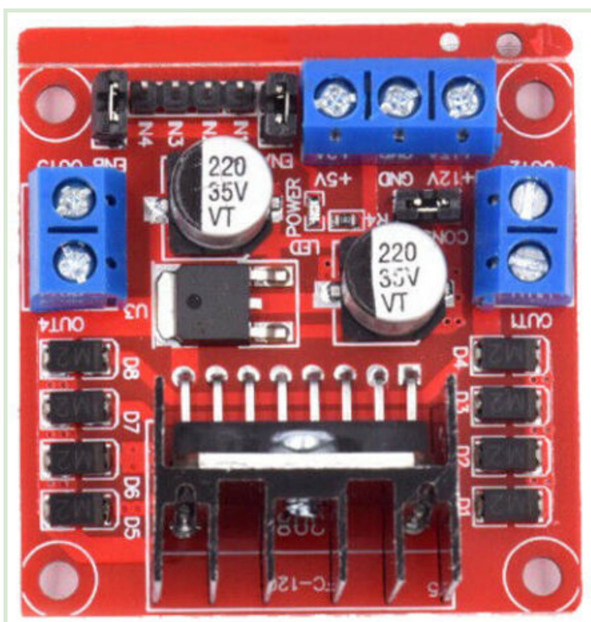


Figure 17: Commercial L298N motor driver



Synergy and Precision: Both motor controllers are integral to achieving the dual-axis tracking capability of the system. The L298N motor driver's robust and reliable performance, combined with the precision and customization offered by the Altium - designed PCB, ensures seamless and efficient operation. The motors, with their 0.4A current rating per phase and fine step angle, contribute to the system's overall precision, enabling minute adjustments in the panel's position for maximal solar energy capture.

This arrangement allows for the dynamic adjustment of the solar panel in both X and Y axes, responding accurately to the solar tracking algorithm's computations based on the data from the LDR sensors.

3.3. Subsystem Validation

Included in the GitHub repository is a video that shows successful LDR readings being printed to the console output and deliberate motor function which validates the motor control phase. Another video in the repository shows real-time code output. Data extracted from the live demo video will yield the following average results.

Table 2: Old LDR Results

Resistance in average light setting	Resistance in Max light intensity
R1 ~~ 1530 +- 50 kohms	R1 ~~ 720+- 50 kohms
R2 ~~ 2200 +- 100 kohms	R2 ~~ 950+- 100 kohms
R3 ~~ 1800 +- 100 kohms	R2 ~~ 900+- 100 kohms
R4 ~~ 1800 +- 100kohms	R2 ~~ 900+- 100 kohms

Table 3: New LDR circuit Results

Resistance in average light setting	Resistance in Max light intensity
R1 ~~ 1950 +- 50 kohms	R1 ~~ 950+- 100 kohms
R2 ~~ 1900 +- 75 kohms	R2 ~~ 940+- 100 kohms
R3 ~~ 1930 +- 100 kohms	R3 ~~ 970+- 100 kohms

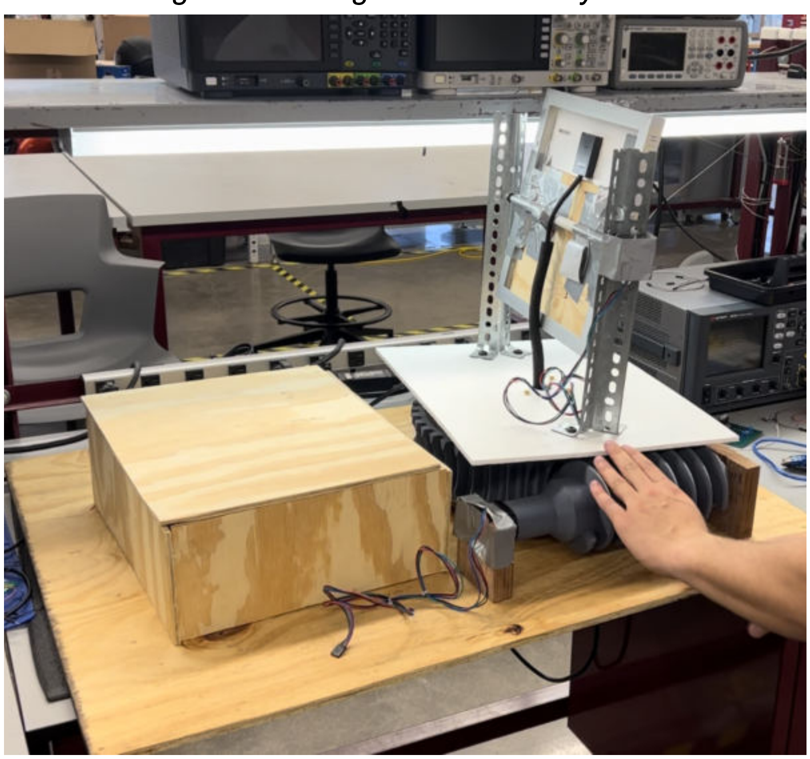
These results validate the relationship of change in Light intensity with the change in Resistance. Updated from 403, The new barrier layout from figure 8 has proven to be a successful improvement to the Photosensor Acquisition System. The resistance changes are more consistent with each other when protected from ambient light noise. Validation for the MCU subsystem is best shown by videos and therefore links have been embedded below. They have also been submitted to the github video folder:

Power integration with only Arduino
Power integration with Arduino & Motor Controllers
Gear Movement
LDR data

3.4. Subsystem Conclusion

Notably, The MCU subsystem went through heavy changes from 403. This entailed a new LDR circuit Layout, changing the motors from DC to Stepper and Creating a new PCB motor driver. It was able to succesfully collect Light intensity Data which was processed by the MCU code. However it wasnt able to integrate the new LDR circuit to the Mechanism on Technical Demo day. This was partly due to the lack of a power subsystem on that day. A lot of time was also spent on trying to power The MCU motor subsystem system by creating a perfboard power rail since bread boards are not allowed but unfortunately this was unsuccessful and a breadboard was used to connect to a DC supply. Another reason for missing the deadline was a lot of PCB testing, due to a lack of soldering experience a lot of time was spent troubleshooting the PCB connections. It is worth mentioning that there was successful intentional motor movement in both motors and the LDR’s when connected to the arduino did produce useful, meaningful data. But the final code wasn't able to be tested well by the demo day deadline, so that part of the system, while in theory works well, has not been completely validated for motor instructions based on LDR readings. Overall, Most of the system’s responsibility was validated.

Figure 18: Integrated Motors System



4. Load Subsystem Report

4.1. Introduction

The Load subsystem consists of two main components, a DC-DC Buck Boost Converter, and a Mosfet Controlled Battery Switch to provide efficient charging to the battery. This subsystem gives the solar panel, being manipulated by the MCU and motors, a purpose to charge a battery.

4.2. Subsystem Details

A DC-DC Buck Boost converter was designed to convert the average 17.5 Volt output of the solar panel to a 14.7 volt requirement for battery charging. To accomplish this task the load subsystem attempted to use a traditional buck converter setup with a Mosfet, inductor, and capacitor and using the battery as a load. Additionally, the subsystem required using an IRF9540NLPBF Mosfet because it had a drain to source a maximum voltage of 100V more than double my maximum input which is the recommended rating. Needing a maximum Gate to Source threshold voltage of 5 volts as the maximum output for the Arduino Rev 3 is 5 volts the mosfet was ideal. This Mosfet has a gate to source a threshold voltage of 2 to 4 volts. A 100 microHenry inductor and a 20 microFarad Capacitor were chosen to achieve minimal output voltage ripple and output current ripple while maintaining a strong output current. Additionally, adding a diode to control the direction of the current before the MOSFET.

However, having such a high voltage output from the solar panel providing a voltage directly to the MOSFET's gate pin wouldn't lead to switching the MOSFET on or off as the drain-to-source voltage would still be greater than the gate-to-source voltage required to turn the MOSFET on. This issue was solved by adding a 2N3904 bipolar junction transistor that would help to decrease the required gate-to-source voltage to the MOSFET.

The subsystem then required a way to turn the current to the battery off. This was able to be accomplished by adding an IRF1010EPBF Mosfet that when powered will allow current to flow to the battery and charge it. When the MOSFET is off it doesn't allow current to flow to the battery. This mosfet was chosen for its drain-to-source breakdown voltage of 60 volts (more than double the input voltage) as well as a Gate to Source threshold voltage of 2 to 4 Volts. The system also included a 10kOhm resistor to decrease the current to the MOSFET so as not to fry the IRF1010EPBF.

Finally, to be able to read the voltages to an Arduino the load needed voltages to be less than 5 volts. This was done by creating voltage dividers at the necessary locations to take the voltage from somewhere between 12-20 volts to a tenth of that at roughly 1.2 to 2 volts. The final design of the buck converter can be seen in Figure 19 and the printed circuit board design can be seen in Figure 20.

Figure 19: Buck Converter Simulation Design

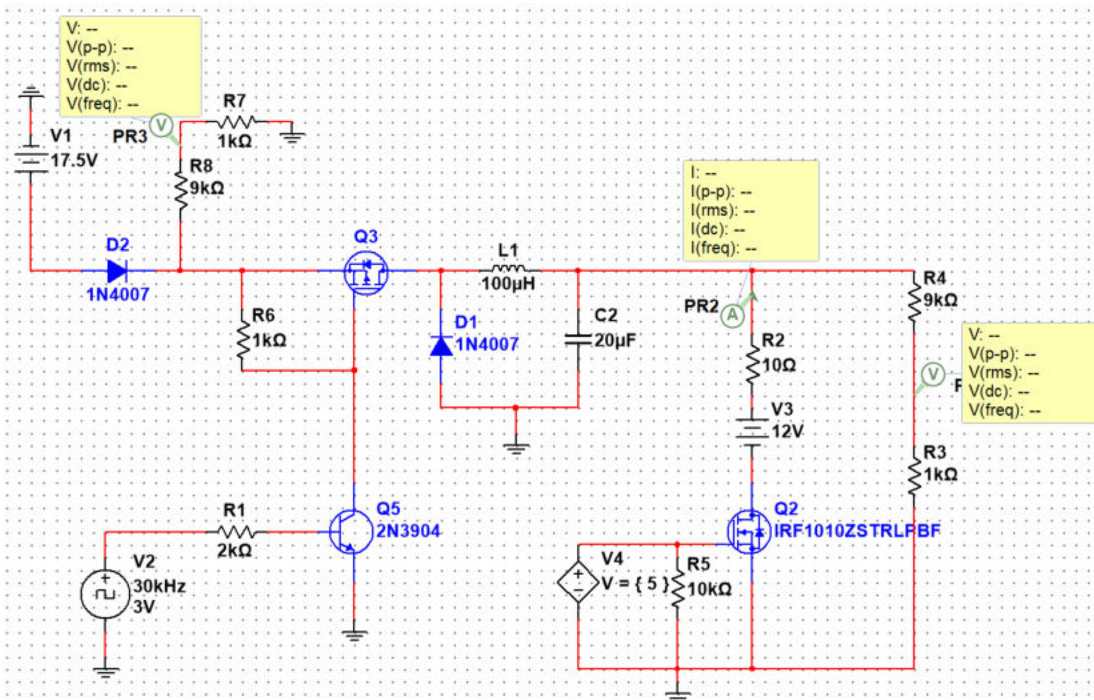
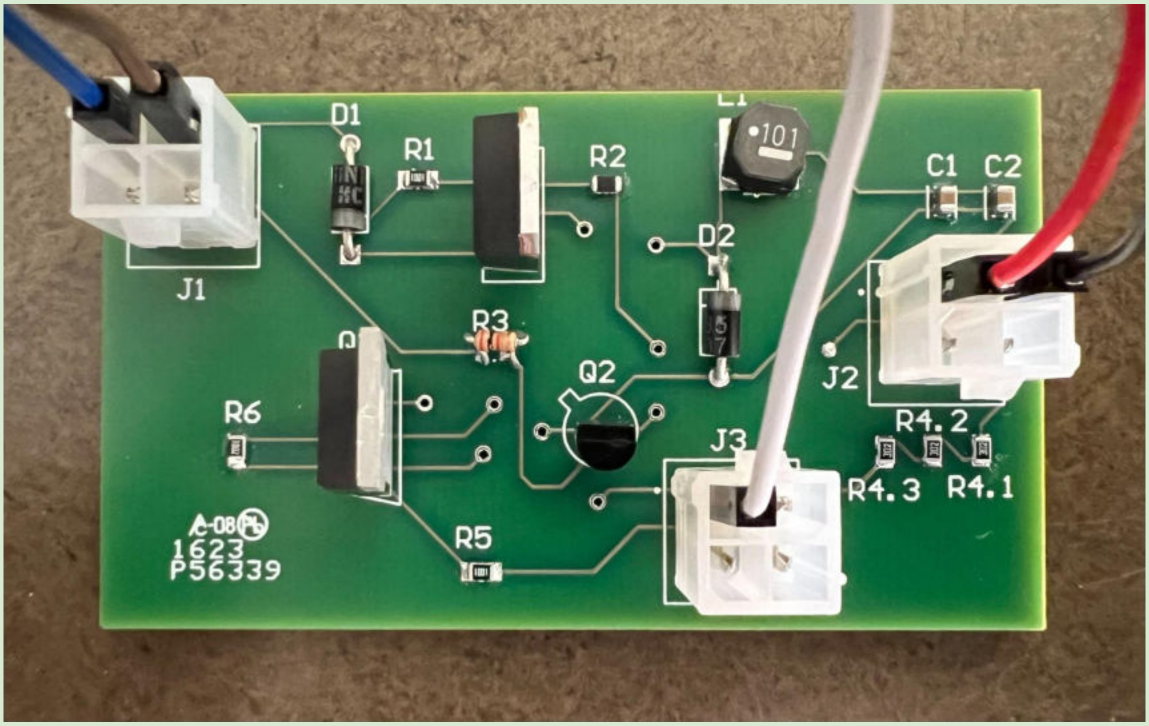


Figure 20: Buck Converter FInished PCB



In 404 the load subsystem updated the design to be a Buck-Boost converter. This was done because after testing, the Solar Panel did not always stay above the 14.7 V rated output. This required a boost component to achieve more efficient charging. After issues using discrete components in 403, the choice was made to switch to an IC to hopefully have more success in 404. This was beneficial because it would be more consistent and effective at converting the input voltage to the required 14.7 V. The design would also prove to be very energy effective when simulated. The design uses a combination of resistors, capacitors and inductors to set an output voltage that would be outputted regardless of the voltage inputted into the system. The choice to use a TPS55289 Buck-boost converter was made because of its impressive voltage capabilities of up to 30V input and a max output voltage of 22V significantly above the required 14.7V. The converter also allowed for up to 8A of current which is plenty for the system as we will max at just below 600mA. It is worth noting that the converter is not affected by this being so much lower than the max current.

Some negatives did come with this updated design. For instance because the output voltage is set at 14.7 due to the nature of the converter's discrete components the output voltage is no longer capable of switching down to 13.7 V on its own. An additional 2nd buck-boost converter or a voltage regulator could have been used however this would have many negative effects on the system as a whole. The main drawback to doing this would be a large increase in the wasted energy as much of the voltage would be lost due to heat. Because the design is attempting to improve the solar panels output it would be very foolish to waste so much energy with a regulator as it would severely decrease the efficiency of the system.

With the new design, it still needed to be able to turn the battery off. This was done by using the previous design from 403 to turn off the current flow through the IRF1010EPBF Mosfet. Unfortunately this wasn't able to work in 403. But validating simulations proved merit in the design; it was decided to implement this into 404. As an additional part of the battery charger the team learned having a resistance was necessary to pull current through the battery. The design needed a large enough resistor to be able to handle the large current and voltage through each of the resistors. However, the equivalent resistance needed to be small enough to not use the entirety of the voltage drop up. This was accomplished by using eight, two hundred and seventy ohm resistors in parallel with one another. This accomplished both the large resistors and the low equivalent resistance.

Figure 21: Buck-Boost Converter Simulation Design

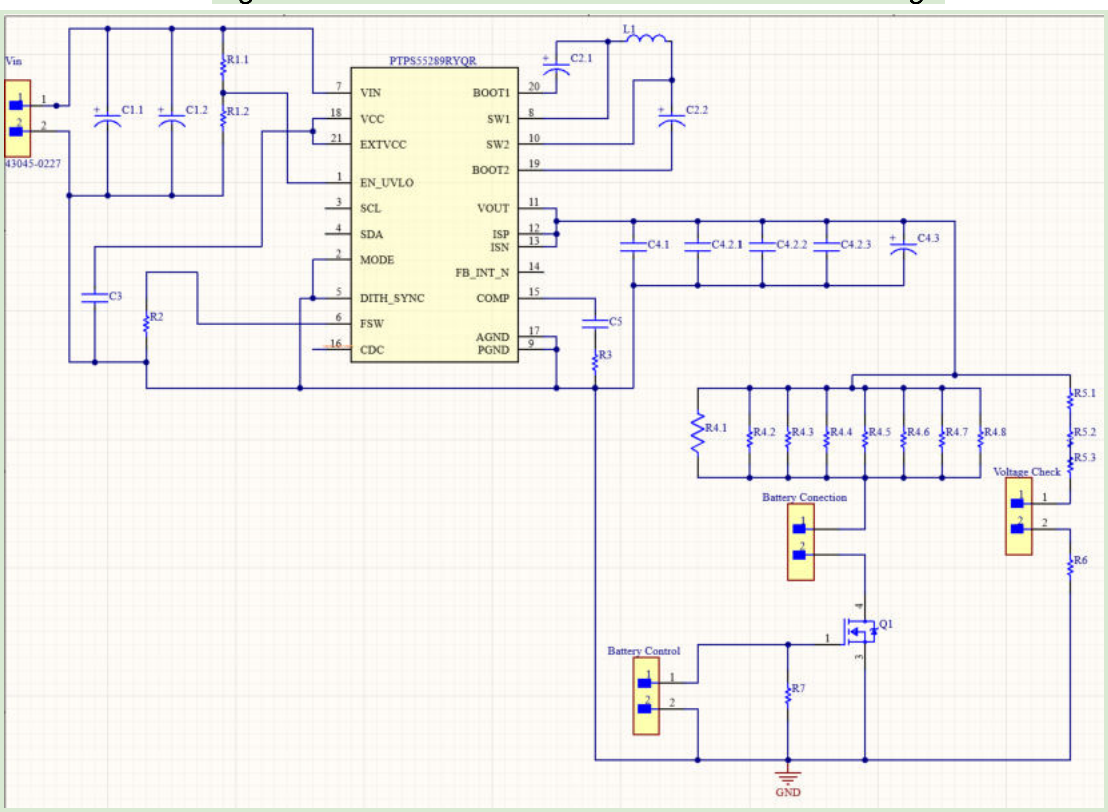


Figure 22: Buck-Boost Converter PCB

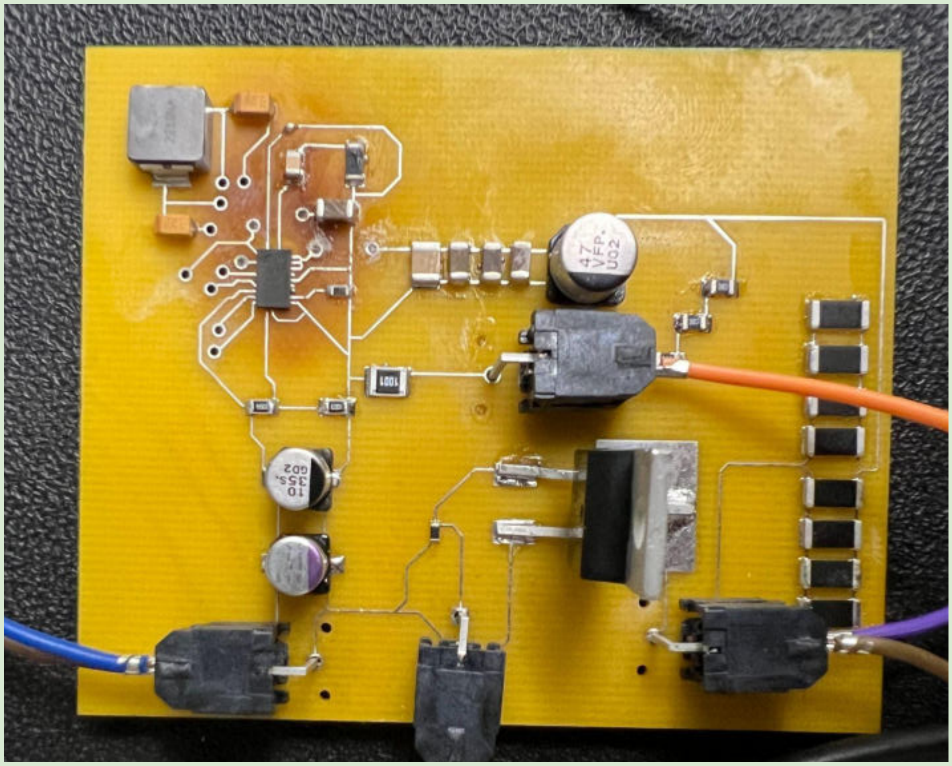
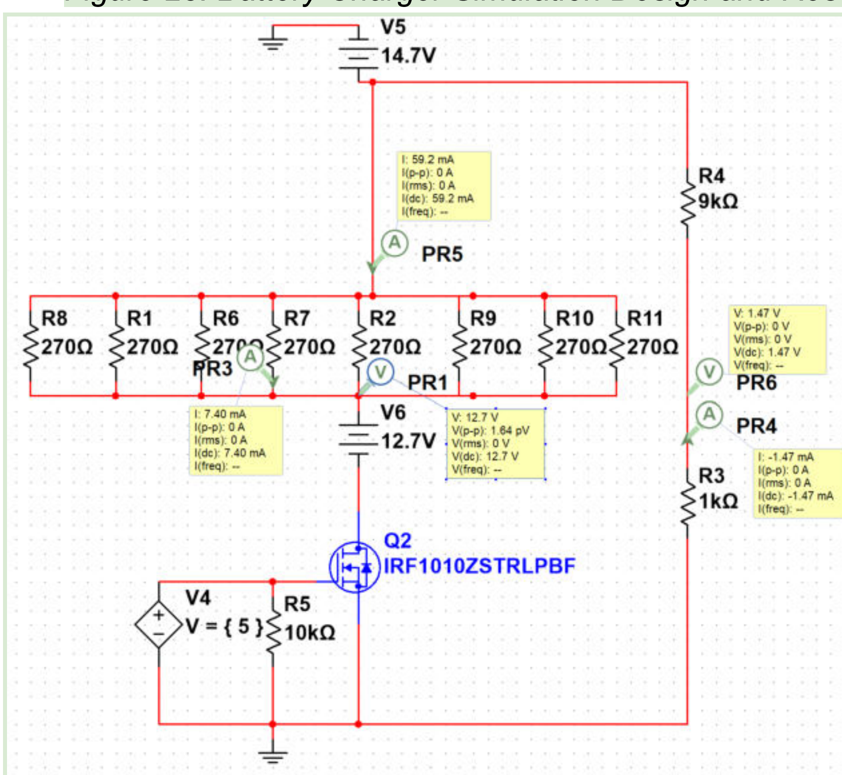


Figure 23: Battery Charger Simulation Design and Results

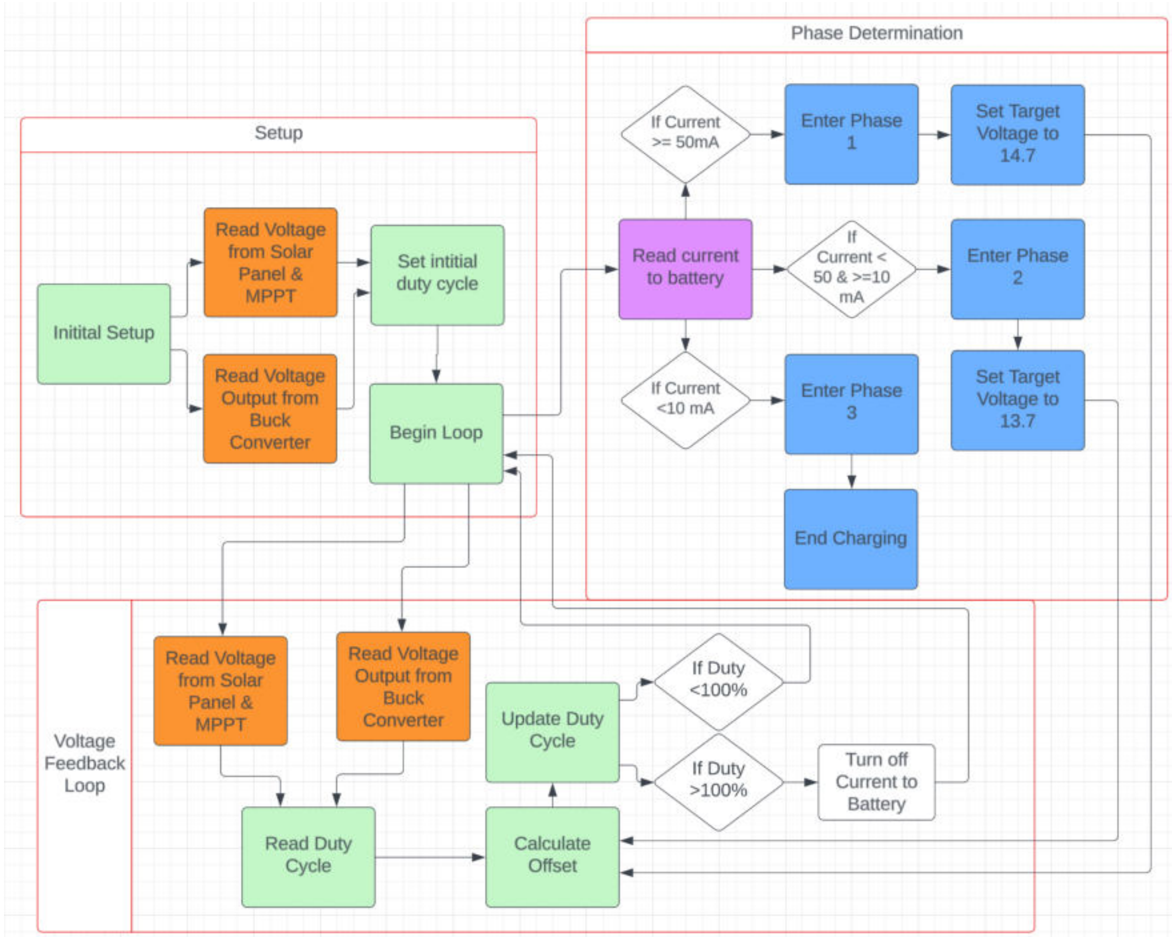


The second piece to the load subsystem is the battery charger code written in the Arduino IDE. Due to the fact that the MCU subsystem had already decided to use an Arduino Uno Rev 3 so it became most convenient to the project as a whole to also use the Arduino Uno Rev 3. We could then combine our code so we would only require a single microprocessor once we begin to combine subsystems.

The battery charger is designed to take the buck converter through 3 separate phases. Phase 1 is the initial phase where the buck converter has a target output voltage of 14.7 Volts and maximizes current to the battery. Phase 2 consists of setting the target output voltage to 13.7 volts and maximizing the current to the battery. Finally, phase 3 is equivalent to turning the battery charger off. Phase 3 accomplishes this by turning the IRF1010 Mosfet to an off position and disconnecting the battery's connection to ground so no current can flow through the connection.

The battery charger also uses a feedback loop to change the duty cycle to achieve the target voltage. The code reads in the DC-DC Buck Converter input and output voltage and adjusts the duty cycle slowly until the output of the buck converter is extremely close to the target voltage. It does this by calculating an offset of the target voltage divided by the output voltage and adjusting the duty cycle until the offset is within 2% of the offset difference. This entire process is shown in *Figure 24*.

Figure 24: Battery Charger Code Flow Diagram



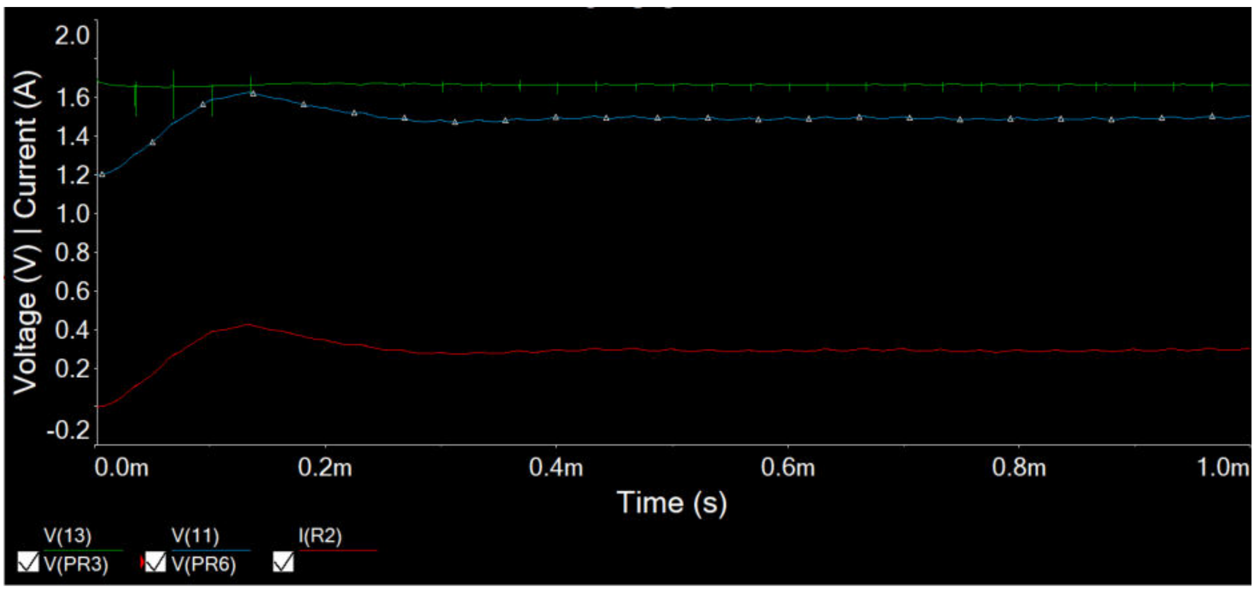
Due to the changes with the IC's new design the need for additional voltage and current sensors was required for the new design. To implement these, code was developed and integrated with Alexandra as part of the GUI Subsystem to read the current and voltages through different parts of the battery charger. These values would then be sent to the GUI to be displayed.

4.3. Subsystem Validation

The DC-DC Buck Converter was simulated in Multisim, built on a perf board, and then constructed within Altium. After the Printed Circuit Board was shipped and physically constructed (Figure 22) the load subsystem ran into several problems. After realizing the buck converter output wasn't giving the correct output the need to begin to test the perf board circuit arose.

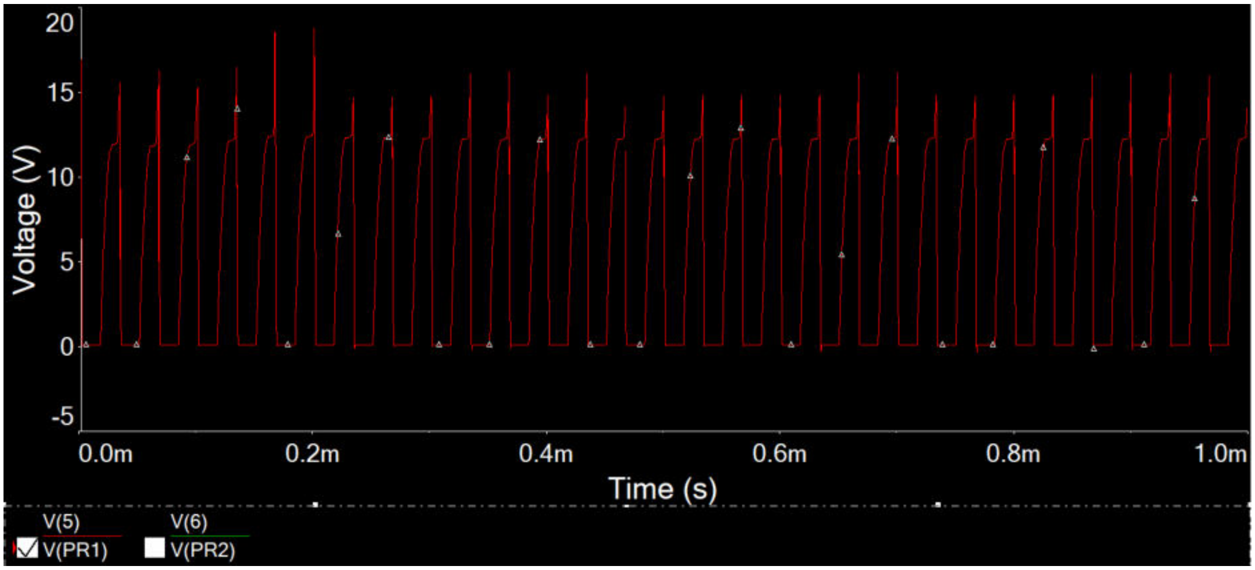
My voltage output was 16.8 Volts. This is from the initial 17.5 Volt input and then the voltage drop across the diode. This clued the load subsystem into the issue that the IRF9540 Mosfet wasn't switching on or off; it was exclusively in an on position. After further testing the drain to source voltage of the IRF9540 Mosfet was found to be equivalent, meaning the difference is 0. This didn't make sense because the input to the bipolar junction transistor was correct. This led back to the simulations where it was found that this was a fundamental problem with the setup. Confirmed by the simulations working properly as indicated in *Figure 25*.

Figure 25: Transient Simulation of Figure 1



However, the subsystem also learned how much glitching occurs with the voltage pulse wave entering the IRF9540 gate as indicated in Figure 26.

Figure 26: Transient Simulation of IRF9540 Mosfet Gate



This wave shows how poor of a square wave entering the MOSFET is. This leads to erratic switching which leads to no switching at all. This led me to fully understand why my setup wasn't working properly even if constructed and designed well. The circuit suffers from fundamental issues that arise from discrete circuit construction.

The new design for the Buck-Boost circuit was simulated, built in Altium and then physically constructed on a pcb which is shown in *Figure 22*. During initial testing and validation the pcb was quickly found that it wasn't working properly. This was noticed that the buck-boost section of the pcb wasn't consistent in its output and would get very hot quickly. The battery charger section of the pcb ran without issues and was able to properly validate, and demonstrate during the lab.

The voltage would drop once plugged in while the current increased. This led to the IC burning during testing leading me obviously frustrated and confused as the simulations had worked correctly. After speaking with Dr. Lusher about many possible issues it could be facing, and being able to determine the root of the issue was the IC and that the part was beyond repair. With the information gained it was decided to reorder parts and attempt to rebuild the pcb. Unfortunately, the team was incredibly far behind when it came to physical construction so this was placed on hold as the system needed to sacrifice part of the subsystem for the good of the team.

The load subsystem was able to analyze the battery charger section of my subsystem and was able to demonstrate the following validation. The system was able to integrate with the GUI subsystem to highlight the current and voltage sensors as we ran the current through the system. Due to the issues with the buck-boost section of the pcb the team used a store bought buck-boost converter to replace the first half of the subsystem and then used the battery charging section of the pcb together. *Figure 27* shows the consistent voltage output out of the buck-boost converter even once the mosfet is turned off. *Figure 28* shows the change when the mosfet is turned off and the current charging the battery is cut off and the charging is switched off. It is noticeable that the current of the battery charging system becomes negative. This is due to the fact that it is a battery and now the polarity of the system is reversed. The current may be seen showing a flow in the opposite direction but this does not drain the battery as the system is shown to be disconnected but the sensor will read the negative current.

Figure 27: Buck-Boost Output Voltage

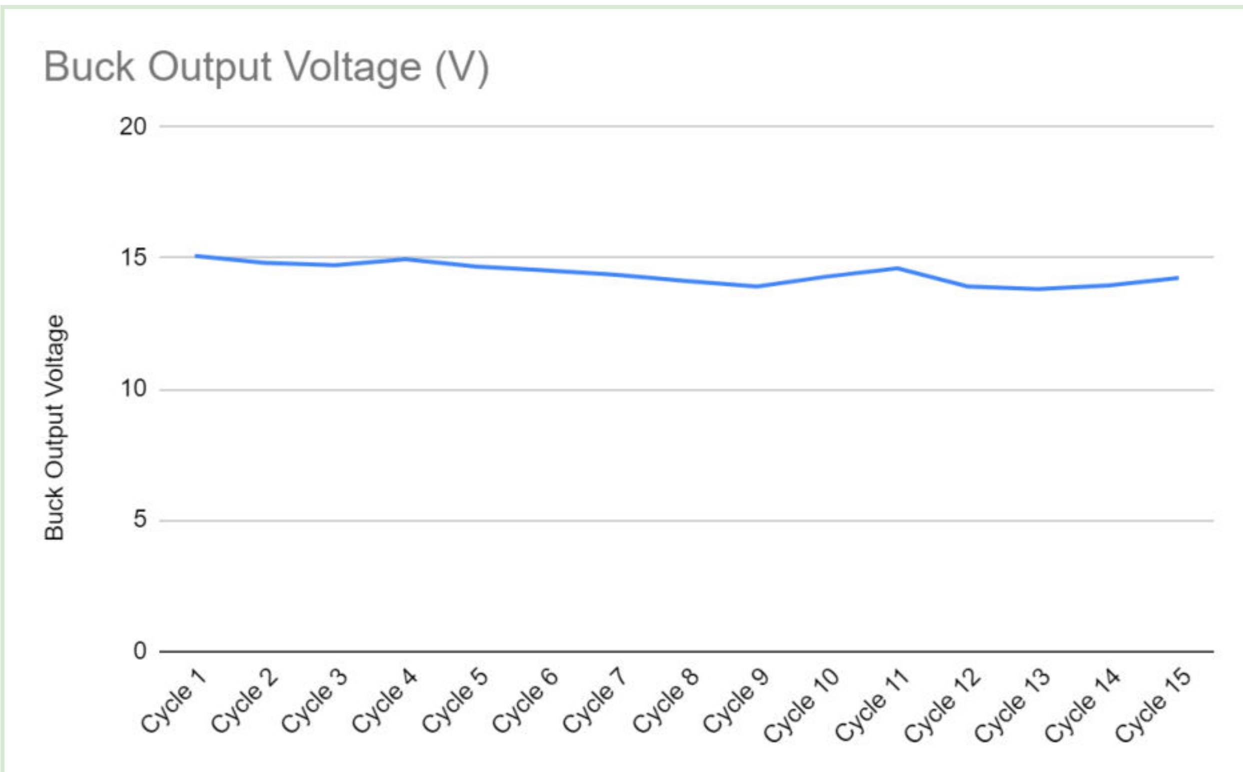
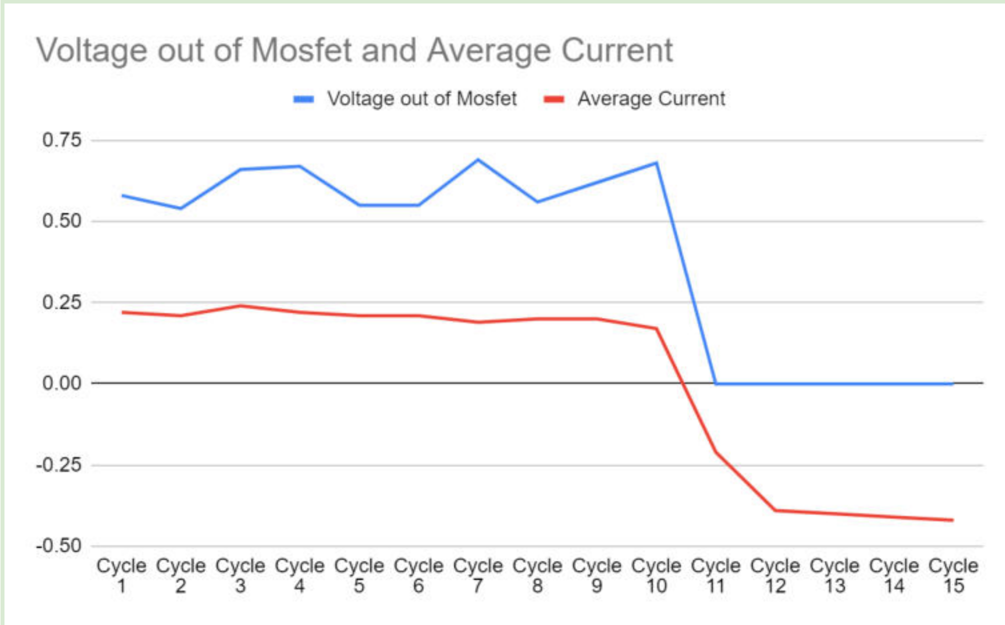


Figure 28: Voltage across Mosfet and Current through Battery Charger



In the Battery Charger code, the load subsystem was able to design a system for handling varying input cases and adjusting the battery charger to change as it goes through its three phases. Using the following pinout system as identified in Table 4.

Table 4: Arduino Uno Rev 3 Battery Charger Pinout

Arduino Uno Rev 3 Pin	Use
A0	Voltage Output from MPPT/Input to DC-DC Buck Converter (Input)
A1	Voltage Output of Buck Converter/ Battery to Ground Voltage (Input)
3	Bipolar Junction Transistor Pulse Width Modulation Voltage (Output)

The system was able to complete the code to the specification needed for switching between the phases and adjusting the voltage output of the buck converter to achieve the correct target voltage. Also cleaned up some of the code used for the demo to make it more readable. The logic and algorithms used within the code worked properly. For testing purposes, using a random variable to emulate the changing solar panel input, the buck converter output, and current setups that will decrease over time by a random amount. This is to show the logic works under varying conditions. Figure 29 is the output of the code showing the current decreasing over time resulting in changing phases. This leads to a change in the target voltage. The charger then loops through to find the correct duty cycle to get the output voltage to an equivalent level as the target voltage.

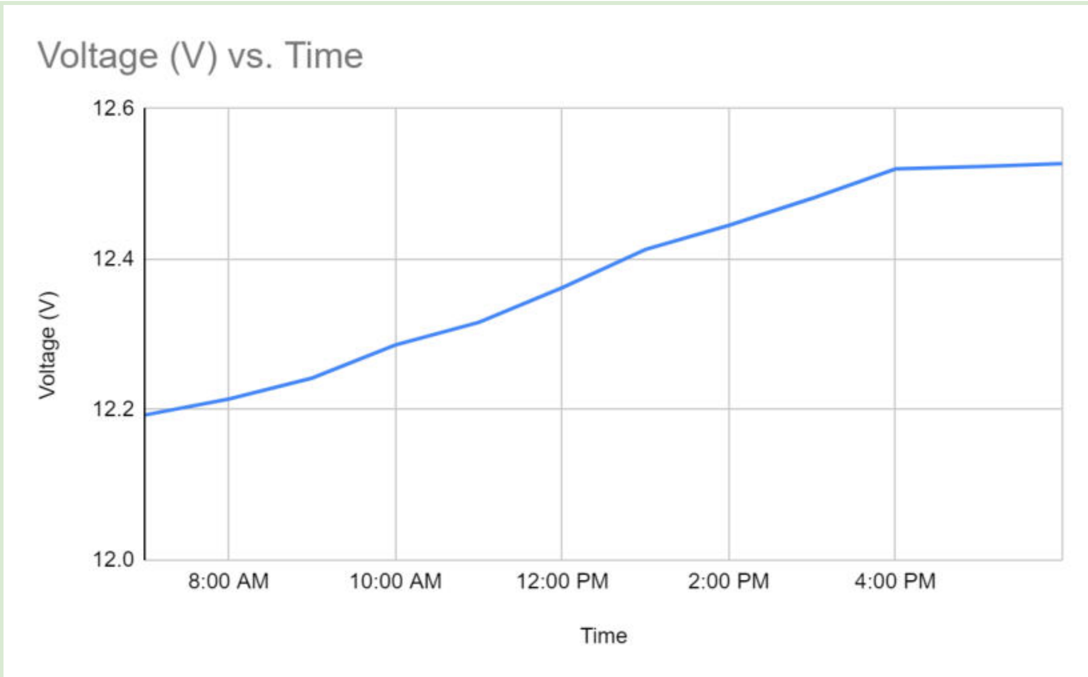
Figure 29: Arduino IDE Battery Charger Code Output

```
targetVoltage: 14.70 iBatt: 203.00 phase: 1 voltageFromMPPT: 17.80 BuckOutputVoltage: 14.50 duty: 82.58
targetVoltage: 14.70 iBatt: 122.00 phase: 1 voltageFromMPPT: 18.80 BuckOutputVoltage: 15.00 duty: 78.19
targetVoltage: 14.70 iBatt: 32.00 phase: 2 voltageFromMPPT: 18.80 BuckOutputVoltage: 13.90 duty: 72.87
targetVoltage: 13.70 iBatt: 10.00 phase: 2 voltageFromMPPT: 13.80 BuckOutputVoltage: 14.60 duty: 99.28
targetVoltage: 13.70 iBatt: 7.00 phase: 3
Battery is fully Charged
Reset
targetVoltage: 14.70 iBatt: 467.00 phase: 1 voltageFromMPPT: 15.80 BuckOutputVoltage: 14.50 duty: 93.04
targetVoltage: 14.70 iBatt: 446.00 phase: 1 voltageFromMPPT: 18.80 BuckOutputVoltage: 15.00 duty: 78.19
targetVoltage: 14.70 iBatt: 367.00 phase: 1 voltageFromMPPT: 20.80 BuckOutputVoltage: 14.50 duty: 70.67
targetVoltage: 14.70 iBatt: 310.00 phase: 1 requiered duty cycle is too high: 1.07
targetVoltage: 14.70 iBatt: 305.00 phase: 1 voltageFromMPPT: 17.80 BuckOutputVoltage: 14.50 duty: 82.58
targetVoltage: 14.70 iBatt: 349.00 phase: 1 voltageFromMPPT: 18.80 BuckOutputVoltage: 14.50 duty: 78.19
targetVoltage: 14.70 iBatt: 302.00 phase: 1 voltageFromMPPT: 17.80 BuckOutputVoltage: 14.50 duty: 82.58
targetVoltage: 14.70 iBatt: 320.00 phase: 1 voltageFromMPPT: 15.80 BuckOutputVoltage: 14.50 duty: 93.04
targetVoltage: 14.70 iBatt: 366.00 phase: 1 voltageFromMPPT: 15.80 BuckOutputVoltage: 14.50 duty: 93.04
targetVoltage: 14.70 iBatt: 288.00 phase: 1 voltageFromMPPT: 15.80 BuckOutputVoltage: 15.00 duty: 93.04
targetVoltage: 14.70 iBatt: 262.00 phase: 1 voltageFromMPPT: 14.80 BuckOutputVoltage: 12.80 duty: 99.32
targetVoltage: 14.70 iBatt: 189.00 phase: 1 requiered duty cycle is too high: 1.07
targetVoltage: 14.70 iBatt: 222.00 phase: 1 voltageFromMPPT: 15.80 BuckOutputVoltage: 14.50 duty: 93.04
targetVoltage: 14.70 iBatt: 207.00 phase: 1 voltageFromMPPT: 18.80 BuckOutputVoltage: 14.70 duty: 78.19
targetVoltage: 14.70 iBatt: 171.00 phase: 1 voltageFromMPPT: 20.80 BuckOutputVoltage: 15.00 duty: 70.67
targetVoltage: 14.70 iBatt: 196.00 phase: 1 voltageFromMPPT: 18.80 BuckOutputVoltage: 14.70 duty: 78.19
targetVoltage: 14.70 iBatt: 190.00 phase: 1 voltageFromMPPT: 18.80 BuckOutputVoltage: 14.50 duty: 78.19
targetVoltage: 14.70 iBatt: 107.00 phase: 1 voltageFromMPPT: 16.80 BuckOutputVoltage: 14.70 duty: 87.50
targetVoltage: 14.70 iBatt: 111.00 phase: 1 voltageFromMPPT: 16.80 BuckOutputVoltage: 15.00 duty: 87.50
targetVoltage: 14.70 iBatt: 149.00 phase: 1 voltageFromMPPT: 16.80 BuckOutputVoltage: 14.70 duty: 87.50
targetVoltage: 14.70 iBatt: 131.00 phase: 1 requiered duty cycle is too high: 1.07
targetVoltage: 14.70 iBatt: 47.00 phase: 2 voltageFromMPPT: 14.80 BuckOutputVoltage: 13.90 duty: 92.57
targetVoltage: 13.70 iBatt: 91.00 phase: 2 voltageFromMPPT: 18.80 BuckOutputVoltage: 13.90 duty: 72.87
targetVoltage: 13.70 iBatt: 62.00 phase: 2 voltageFromMPPT: 17.80 BuckOutputVoltage: 13.90 duty: 76.97
targetVoltage: 13.70 iBatt: -30.00 phase: 3
Battery is fully Charged
Reset
targetVoltage: 14.70 iBatt: 544.00 phase: 1 requiered duty cycle is too high: 1.07
targetVoltage: 14.70 iBatt: 575.00 phase: 1 voltageFromMPPT: 14.80 BuckOutputVoltage: 14.70 duty: 99.32
```

This shows the possibility of having a duty cycle of above 100% for this we would be unable to use a buck converter. The best solution would be a buck-boost converter that could bring the voltage to the required target voltage. This would eliminate the need for turning off the current to the battery when the duty cycle is greater than 100%. It is worth noting that the current to the battery would be reduced when the voltage has to be increased during the boosting of the input voltage.

The load subsystem was also able to charge the battery using the store bought buck-boost and the battery charging part of the pcb. Figure 30 shows how the voltage changes as the battery is charged throughout the day. The battery started at around 48% charged and rose to roughly 82% charged. When the battery is least charged you expect the voltage to increase the fastest. However, due to the rising level of the sun throughout the morning it led to a faster charge in the middle of the day even though the battery was more fully charged. Toward the end of the day when the battery was both well charged and the sun began to decline the rate at which the batteries voltage increased began to plateau. This is shown in *Figure 30.*

Figure 30: Battery Voltage throughout Charging



4.4. Subsystem Conclusion

The Load Subsystem currently contains most of the required functionality. Unfortunately, The Buck converter half of the subsystem resulted in unforeseen issues requiring the use of an Integrated Circuit. This however may come as a benefit because of the testing found through debugging the circuit having learned it may be more ideal to convert the buck converter to a buck-boost converter instead. This will help with over-current charging as the current to the battery will not need to be turned off when under the required voltage from the MPPT. The battery charger code half of the subsystem was able to work properly. It will require a small amount of additional work regarding integration with the buck-boost IC, but the overall function and logic work as intended.

After 404 The load subsystem was able to complete or at least integrate an alternative to each part of my subsystem with the rest of the team excluding the MCU subsystem. This is due to the MCU subsystem falling behind and being unable to be prepared by the time our team demod. Unfortunately, the PCB wasn't able to be reconstructed after the IC burned. The hope was to get it into a fully working state as Dr. Lusher had verified and approved the design and the simulations for it.

The load subsystem was able to get the majority of the necessary completion pieces ready for integration. Successfully implementing the battery charger that allows for the battery charging to be turned off once the battery reaches close to full capacity. Successfully validating and showing this part of the system working correctly during the demo. Additionally, Integrating the whole subsystem with the GUI subsystem to turn off/on the battery charging.

The load subsystem was able to successfully implement the voltage and current sensors required to read the correct voltage and current values of the battery charging system. This was also implemented and integrated with the GUI Subsystem.

A large portion of the work in 404 came from becoming the only person responsible for the physical construction of the project. Some assistance was received from Talha Zaheer as he was able to contribute his skills in CAD and 3D printing. However, when it came to the cutting, drilling, bolting, gluing, and sawing this was all placed into the responsibility of the load subsystem. The load subsystem has to place the pcb validation on pause to have anything built for demo. *Figure 31* highlights the physical construction of the project. This fabrication process was very intricate and time intensive and became a large portion of the load subsystems contribution to the team in 404. A document available in the github highlights the process and intricate nature of the fabrication process.

Figure 31:Full System Physical Construction

5. Graphical User Interface Subsystem Report

5.1. Introduction

The Graphical User Interface (GUI) subsystem consisted of both a website and an ESP32 WROOM module, which integrated with each other to enable users to analyze and monitor data from three sensors: an INA219 current sensor, an ADC voltage sensor, and an MPU6050 gyroscope sensor.

5.2. Subsystem Details

The development of the GUI subsystem combined web development, hardware interfacing, and data processing to project the system's readings for the user. Initially, a circuit was constructed to connect the INA219 sensor, a voltage sensor, and the MPU6050 sensor to the microcontroller (MCU). This setup ensured correct transmission of readings to the ESP-32 WROOM microcontroller. The ESP-32 WROOM, serving as the main communication module between the sensors and the website, worked by receiving data from the sensors, processing it, and then communicating the data to the website via WiFi.

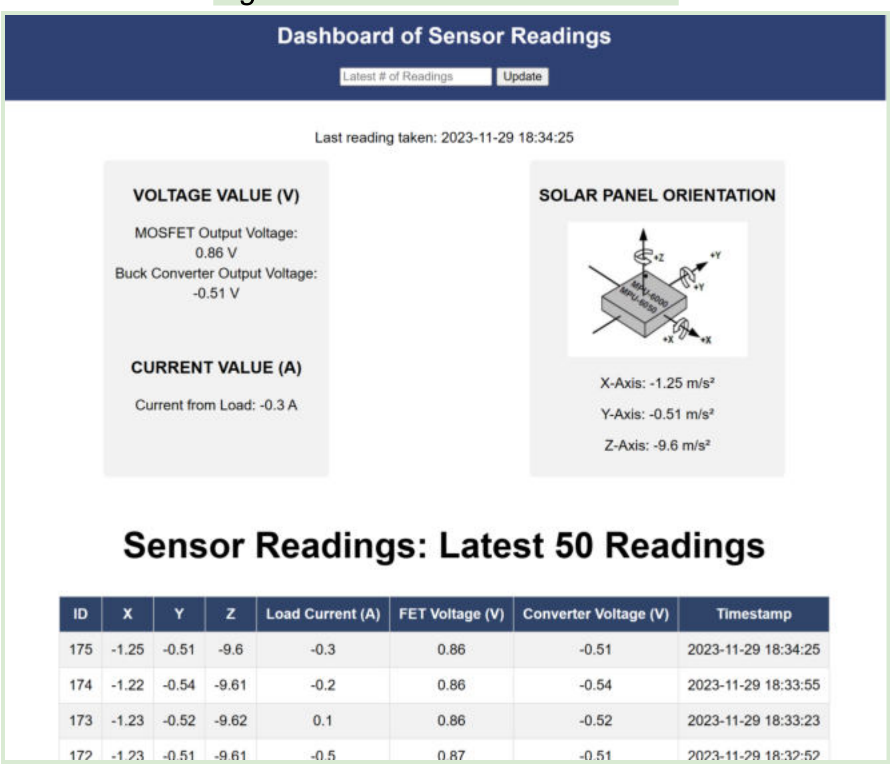
The INA219 and MPU6050 sensors, both I2C sensors, and the voltage sensor, an ADC sensor, required different considerations for the configuration of the circuit. I2C, a communication protocol, connects sensors to the MCU using two lines: the serial clock and serial data lines. The circuitry for these I2C sensors involved connecting them to the same I2C bus. The voltage sensor, communicating via an ADC connection, was connected to the MCU using the GPIO39 pin, which reads analog values.

Once the circuit was configured, the ESP32 was programmed. The firmware code development included several steps, such as library inclusions, sensor initialization and calibration, network setup, data processing and transmission, timing, and error handling. The most notable aspects were the data processing and network setup, which integrated the MCU and the website. The firmware code created a query string for an HTTP post request, including parameters to identify the sensor, sensor readings, and location. This code was then utilized in the HTTP request to send data to the web server.

The second half of the subsystem's development focused on web development for the GUI. To store data sent from the MCU, an SQL table was created using a PHP script to insert a timestamp, sensor identity, and sensor readings into individual rows. A database.php file was created, serving as the interaction point for the database and retrieval of data for the website. A post-data.php file was then developed to handle POST requests from the MCU, inserting data into the website. This file included database.php as a reference. An API Key was verified in the post-data.php file to authenticate POST requests. Data extracted from the POST requests was inserted into the database using an “insert reading” function defined in database.php, adding the data to the previously created SQL table.

For the front-end of the website, two major files were utilized: styles.css and soltracker.php. The styles.css file contained CSS code to define the visual appearance of the website. The soltracker.php file was the primary visual file, arranging the visual elements on the website. It included PHP backend logic to interact with database.php, fetching sensor data to be displayed on the front-end. This file also incorporated HTML and CSS elements to structure and style the dashboard-like website.

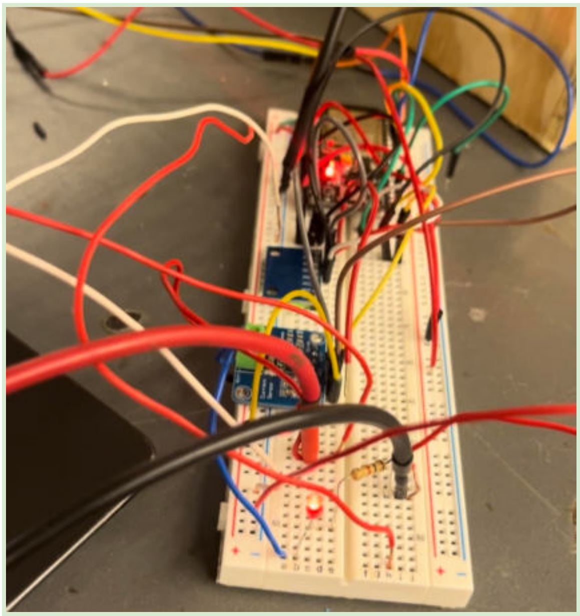
Figure 32: Final Website Product



5.3. Subsystem Validation

Several steps were involved in ensuring the performance and efficiency of the overall subsystem. Initially, the circuit of the MCU and sensors was tested on a breadboard to ensure the physical connections were correct. This was a crucial first step, as it not only validated the overall performance of the MCU but also served as the blueprint for the perfboard. Once the circuit had been constructed on the breadboard, the firmware code was imported into the MCU, and the circuit's performance was monitored via the serial monitor of the Arduino IDE.

Figure 33: Breadboard of the ESP32, INA219, and MPU6050 Sensors



After the breadboard circuit was constructed, the sensors were individually tested to ensure that each behaved as intended. This was accomplished by using separate test programs to initialize, calibrate, and read from the sensors individually. For the INA219 current sensor and the voltage sensor, the outputs were compared to readings obtained from a multimeter and the expected outputs from a DC power supply. Similarly, the MPU6050 was tested individually. To test this sensor, it was moved in different orientations to verify its responsiveness. This involved moving the sensor along the X, Y, and Z axes and monitoring the readings outputted by the serial monitor.

Once the sensors had been tested, the circuit was transferred to a perfboard for the final demonstration. The perfboard circuit was tested again to ensure that all parts were correctly soldered. The testing and validation of the perfboard were similar to the breadboard tests, where the perfboard's output was monitored through the Arduino IDE's serial monitor. The readings from the three sensors were compared to readings obtained from the power supply, the multimeter, and the movements along the X, Y, and Z axes.

Figure 34: Perfboard with MCU, UART Bridge, and Current and Voltage Sensors

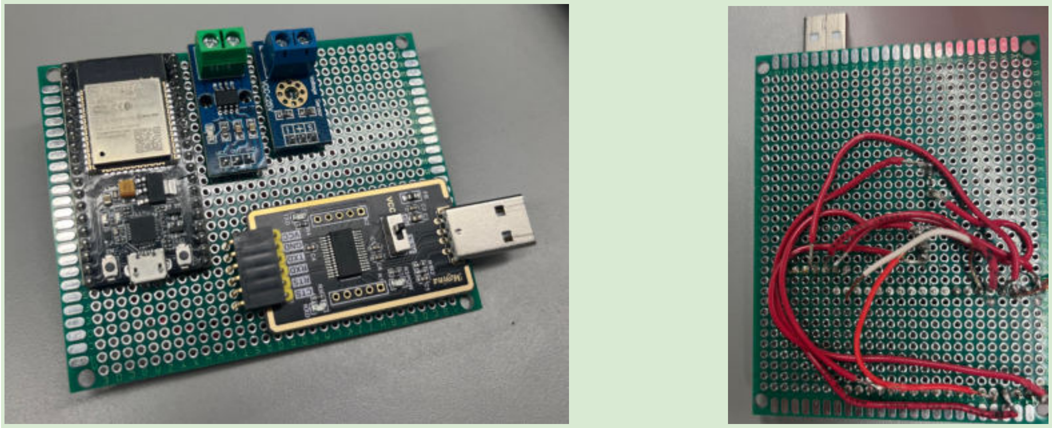


Table 5: Comparisons of Voltage and Current Sensor Readings to Supply

Expected Voltage (V)	Sensor Reading (V)	Error Percent (%)
3.3V	3.21	2.72
3.7V	3.61	2.43
3.9V	3.79	2.82
4.2V	4.07	3.095
4.6V	4.45	3.26
5V	4.845	3.1
	Average Error=	2.904166667

Expected Voltage (V)	Sensor Reading (V)	Error Percent (%)
13mA	12.5	3.85
17mA	16.49	3.01
19mA	18.45	2.89
22mA	21.38	2.81
26mA	25.25	3.02
30mA	29.19	2.7
	Average Error=	3.046666667

Table 6: Readings obtained from the MPU6050 on a Flat Surface

Orientation	X	Y	Z
Flat	0.21	0.01	9.97
Flat	0.2	0.01	9.98
Flat	0.23	0.01	9.98
Flat	0.2	0.01	9.97
Flat	0.22	0.01	9.99
Flat	0.21	0.01	9.95
Flat	0.2	0.01	9.95
Flat	0.2	0.01	9.99
Flat	0.21	0.01	9.95
Flat	0.21	0	9.95
Flat	0.2	0.03	9.97
Flat	0.21	0.02	9.96
Flat	0.21	0	9.91
Flat	0.22	0.01	9.97
Flat	0.22	0.02	9.96
Flat	0.2	0.01	9.98
Flat	0.2	0.01	10
Flat	0.22	0	9.97
Flat	0.21	0.02	9.99
Flat	0.21	0.03	9.98
Flat	0.2	0	9.99
Flat	0.2	0.01	9.99
Flat	0.2	0.02	9.97

The readings obtained from the MPU6050 were compared to the specifications listed in the MPU6050 Datasheet. Above is an example of a test performed, where data was obtained from a specific orientation (Flat surface orientation) and compared to an expected orientation value.

After validating the performance of the MCU component of this subsystem, the data processing was validated to ensure seamless communication of data from the MCU to the website. The database was tested to verify that data was being inserted and updated as expected. This involved reading the data obtained by the database and ensuring that it matched the data transmitted from the MCU, observable from the Arduino IDE's serial monitor. Additionally, the integrity of the website was tested and validated. The database was accessed through an API Key, so the integration of the website and database was tested to ensure proper communication between these components.

Figure 35: Output from the Serial Monitor of the Arduino IDE

```
11:23:21.093 -> HTTP Response code: 200
11:23:51.077 -> Bus Voltage:   0.88 V
11:23:51.077 -> Shunt Voltage: -0.04 mV
11:23:51.077 -> Load Voltage:  0.88 V
11:23:51.077 -> Current:       -0.30 mA
11:23:51.077 -> httpRequestData: api_key=tPmAT5Ab3j7F9&sensor=MPU6050_INA219&location=School&value1=-1.29&value2=-0.72&value3=-9.59&value4=0.88&value5=-0.30
11:23:51.926 -> HTTP Response code: 200
11:24:21.941 -> Bus Voltage:   0.88 V
11:24:21.941 -> Shunt Voltage: -0.04 mV
11:24:21.941 -> Load Voltage:  0.88 V
11:24:21.941 -> Current:       -0.20 mA
11:24:21.941 -> httpRequestData: api_key=tPmAT5Ab3j7F9&sensor=MPU6050_INA219&location=School&value1=-1.30&value2=-0.74&value3=-9.64&value4=0.88&value5=-0.20
11:24:22.957 -> HTTP Response code: 200
11:24:52.957 -> Bus Voltage:   0.88 V
11:24:52.957 -> Shunt Voltage: -0.01 mV
11:24:53.007 -> Load Voltage:  0.88 V
11:24:53.007 -> Current:       -0.30 mA
11:24:53.007 -> httpRequestData: api_key=tPmAT5Ab3j7F9&sensor=MPU6050_INA219&location=School&value1=-1.31&value2=-0.79&value3=-9.61&value4=0.88&value5=-0.30
11:24:53.841 -> HTTP Response code: 200
11:25:23.868 -> Bus Voltage:   0.88 V
11:25:23.868 -> Shunt Voltage: -0.04 mV
11:25:23.868 -> Load Voltage:  0.88 V
11:25:23.868 -> Current:       -0.50 mA
```

The output from the serial monitor gave important information regarding the status of the MCU, sensors, WiFi connection, and the HTTP response code. An example of the output obtained from the firmware code is shown above.

Figure 36: Result of the SQL Database

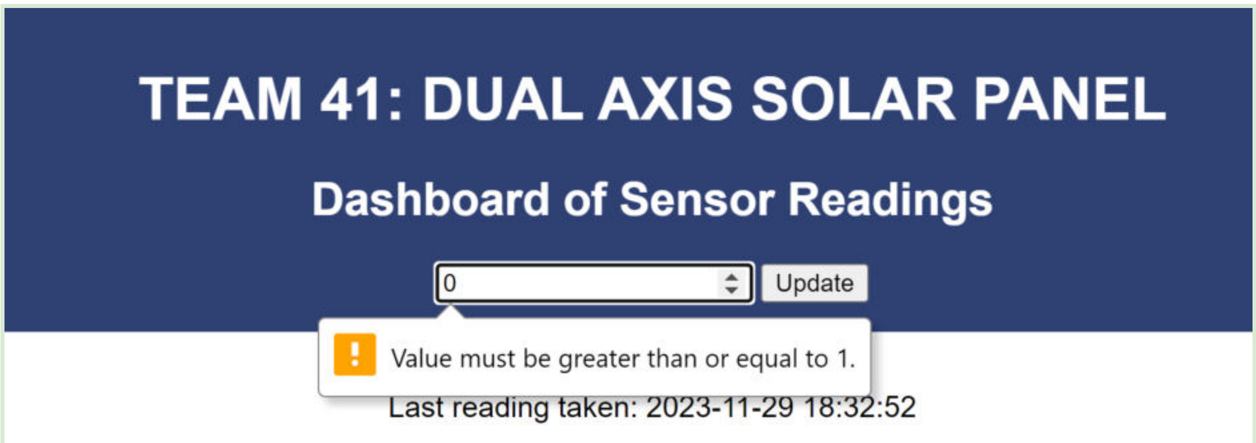
Click the drop-down arrow to toggle column's visibility.

←T→			id	sensor	location	value1	value2	value3	reading_time	bus_voltage	current_mA
Edit				PU6050_INA219	School	-0.21	-0.01	-9.97	2023-11-20 05:38:52	0.86	-0.4
Edit	Copy	Delete	14	MPU6050_INA219	School	-0.22	-0.01	-9.98	2023-11-20 05:39:26	0.86	-0.2
Edit	Copy	Delete	15	MPU6050_INA219	School	-0.23	-0.01	-9.98	2023-11-20 05:39:57	0.86	-0.5
Edit	Copy	Delete	16	MPU6050_INA219	School	-0.2	-0.01	-9.97	2023-11-20 05:40:28	0.86	-0.4
Edit	Copy	Delete	17	MPU6050_INA219	School	-0.22	-0.01	-9.99	2023-11-20 05:40:59	0.86	-0.4
Edit	Copy	Delete	18	MPU6050_INA219	School	-0.22	-0.01	-9.95	2023-11-20 05:41:31	0.86	-0.4
Edit	Copy	Delete	19	MPU6050_INA219	School	-0.21	-0.01	-9.95	2023-11-20 05:42:02	0.86	-0.4
Edit	Copy	Delete	20	MPU6050_INA219	School	-0.22	-0	-9.99	2023-11-20 05:42:33	0.86	-0.4
Edit	Copy	Delete	21	MPU6050_INA219	School	-0.2	-0.03	-9.97	2023-11-20 05:43:04	0.86	-0.4
Edit	Copy	Delete	22	MPU6050_INA219	School	-0.21	-0.02	-9.96	2023-11-20 05:43:35	0.86	-0.1
Edit	Copy	Delete	23	MPU6050_INA219	School	-0.21	-0	-9.91	2023-11-20 05:44:09	0.86	-0.4
Edit	Copy	Delete	24	MPU6050_INA219	School	-0.2	-0.01	-9.97	2023-11-20 05:44:40	0.86	-0.2
Edit	Copy	Delete	25	MPU6050_INA219	School	-0.2	-0.02	-9.96	2023-11-20 05:45:12	0.86	0
Edit	Copy	Delete	26	MPU6050_INA219	School	-0.21	-0.01	-9.95	2023-11-20 05:45:45	0.86	-0.3
Edit	Copy	Delete	27	MPU6050_INA219	School	-0.21	-0.01	-9.98	2023-11-20 05:46:17	0.86	-0.4
Edit	Copy	Delete	28	MPU6050_INA219	School	-0.21	0	-10	2023-11-20 05:46:48	0.86	-0.2
Edit	Copy	Delete	29	MPU6050_INA219	School	-0.22	-0.02	-9.97	2023-11-20 05:47:22	0.86	-0.2
Edit	Copy	Delete	30	MPU6050_INA219	School	-0.2	-0.03	-9.99	2023-11-20 05:47:54	0.86	-0.3

The outputs obtained from the serial monitor and SQL database were compared, as the SQL database was receiving the information from the MCU and sensors. This was a step mentioned in the validation process of this subsystem. The output of the database is shown above.

Lastly, the front end of the website was validated for its functionality, intuitive nature, and responsiveness. The components tested were its input fields, submit button, and the table that displayed the data received from the database. The functionality of the input field and submit button was tested by inputting values that should return an error, such as letters or the number 0, and observing the website's response. Additionally, the table was updated in accordance with the user's input. The results of this testing were documented as indicated:

Figure 37: Error Message - Entered “0” in Input Field



Link to video showing the output when a character, such as “A” , is entered:
https://youtube.com/shorts/aREHnAx3tMg?si=SRD07tPvU8KF2XXJ

5.4. Subsystem Conclusion

Joining this team in August and faced with the time-constraint of having one semester to create a website and incorporate the hardware setup and firmware development of the MCU, the website was designed in a straightforward manner to guarantee its functionality and reliability. Initially, the intent of the website was to incorporate more web components, such as a side-navigation bar with links to a device manual, contact information, etc. to create a more “commercial” website. However, due to time, the intent of the website shifted toward a dashboard-website format with sensor data readings. In result, the overall GUI subsystem did not successfully meet all the initial objectives that were established at the start of the design process. However, the GUI subsystem did result in a functional dashboard that presented accurate sensor readings obtained from the system.

Solar Tracker

Osman Farook
Talha Zaheer
Cooper Hamlin
Alexandra Torres

FULL SYSTEM REPORT

REVISION – 4
12/04/2023

Full System Report
for
Solar Tracker

Prepared by:
Team <41>

____
Author Date

Approved by:

__Cooper_Hamlin__________4/26/23
Project Leader Date

____
John Lusher II, P.E. Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	[12/03]	Team 41		Initial Draft
1	[12/04]	Cooper Hamlin		404 Final Report

Table of Contents

List of Figures

1. System Overview

The dual-axis solar tracker improves upon the traditional functionality of a solar panel by allowing for x and y axis movement that follows the sun in order to boost efficiency. The system consists of several subsystems including Power, MCU, Load and the GUI. All systems had to be integrated in order for the device to function as intended. There were a number of issues at the end of the semester which prevented the full completion of the system and the device did not meet all of the requirements and specifications that had been laid out initially.

2. Development Plan and Execution

As mentioned before, there were four subsystems in total that were built over the course of the two semesters. 3 of the subsystems were planned originally while a 4th subsystem of the GUI was introduced at the halfway point of the project. While the first semester was spent on finalizing the intricacies of each subsystem, the second semester was spent integrating and designing the device itself.

2.1.1 Physical Design Plan

The mechanical design of the project was planned and refined over the course of both semesters. There were a number of challenges that had to be overcome and a lot of ideas were explored in terms of the best way to go about the construction of the solar tracker. Cooper and Talha took the lead initiative in this part of the project and came to the idea of using a worm screw and gear at the bottom which would allow for a smooth transition within the bottom axis. Meanwhile, the top axis would include an axial rod which would mount the Solar Panel. The motors would be connected to both the worm screw and the axial rod respectively.

The 3D design was created in solidworks for many components in order to test how they would fit in the entire design. Figure 1 shows the design of the worm screw which would be attached to the motor in order to turn a gear. Figure 2 shows the final gear drawing while Figure 3 shows a quarter of the gear. Given that the gear would not be able to fit in the FEDC 3D printers due to its size, it had to be printed in 4 parts which would later be joined together.

Figure 1: Worm Screw Drawing

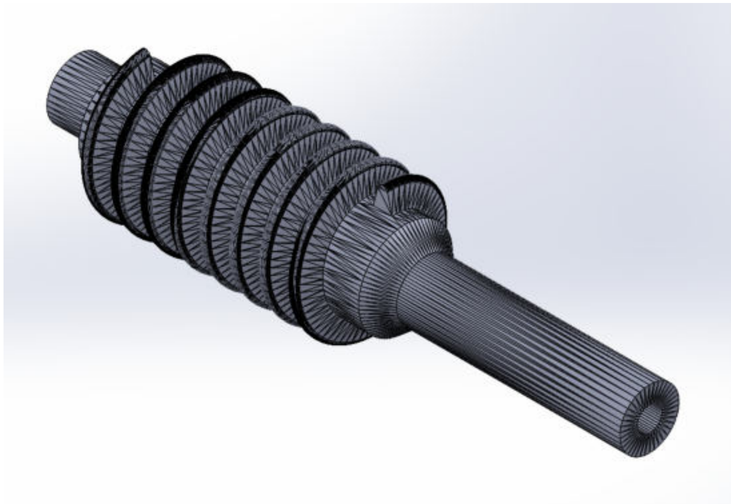


Figure 2: Gear Drawing

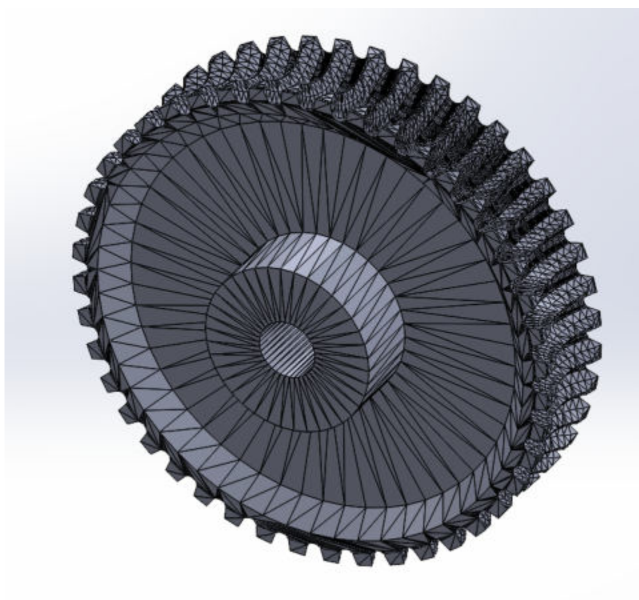
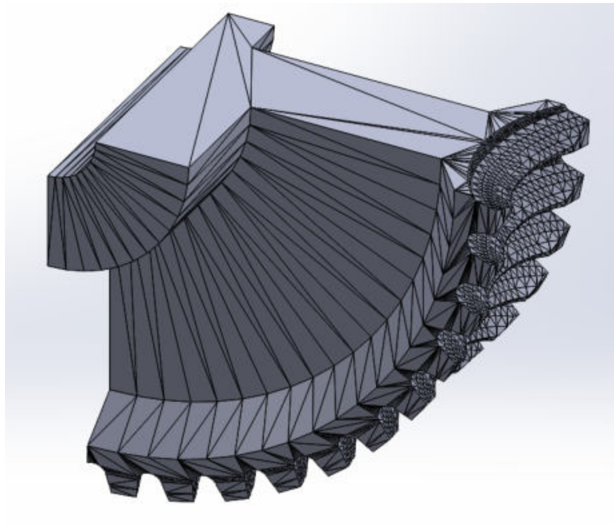


Figure 3: ¼ Gear Drawing



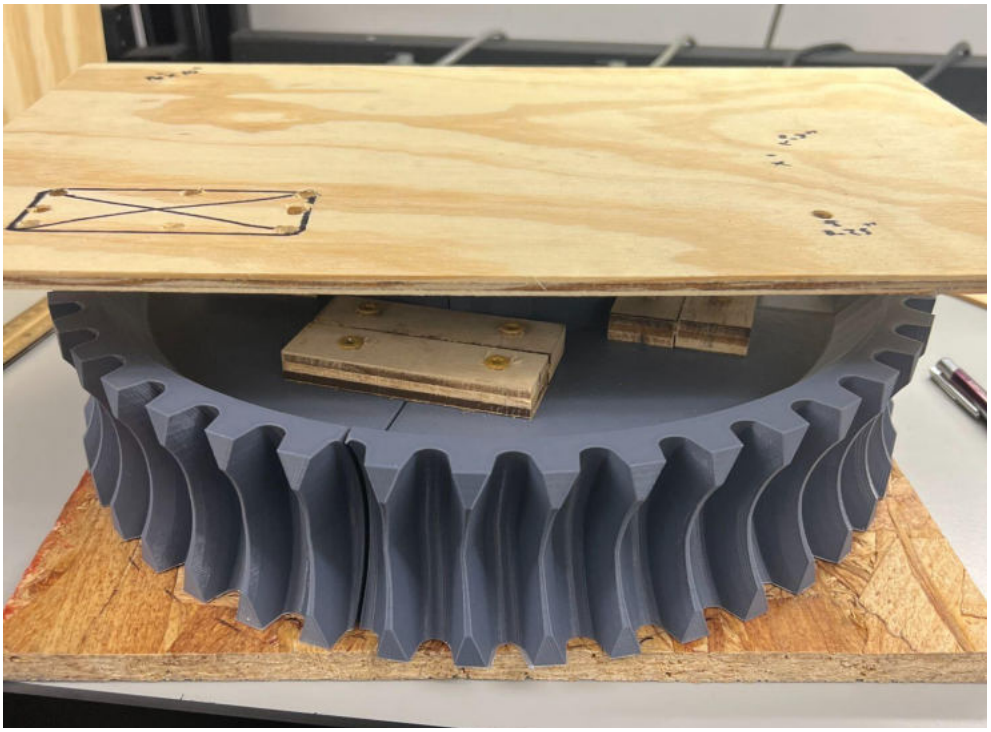
These components would serve as major building blocks for the final design as the top axis would be mounted on the gear itself. This meant that the design had to be robust and capable of withstanding wind and other elemental factors. Cooper took the initiative to incorporate the physically printed parts within the machine shop which included drilling the correct holes and cutting the right materials to finalize the device itself.

2.1.2. Enclosure Box

All of the electronic components would need to be housed in a constructed enclosure box that would have walls on all sides protecting the sensitive electrical equipment. The only access points to the enclosure box are two drilled holes that allow wire casings carrying all necessary wires to their respective locations throughout the system.

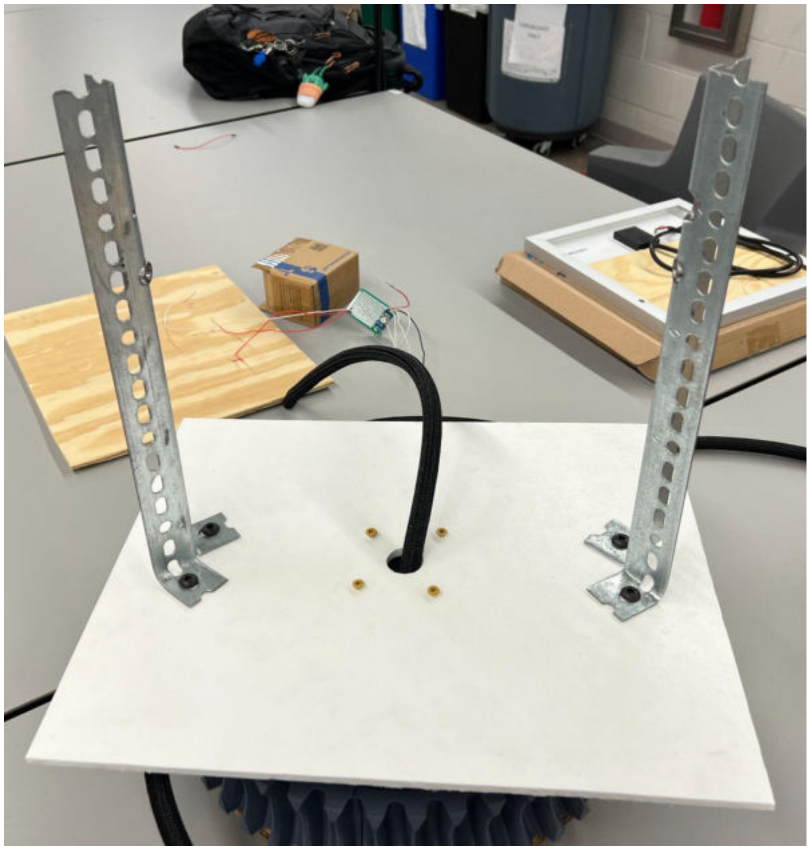
2.1.3. Fabrication of System

Figure 4: Securing the Worm Gear and Support Beam Platform



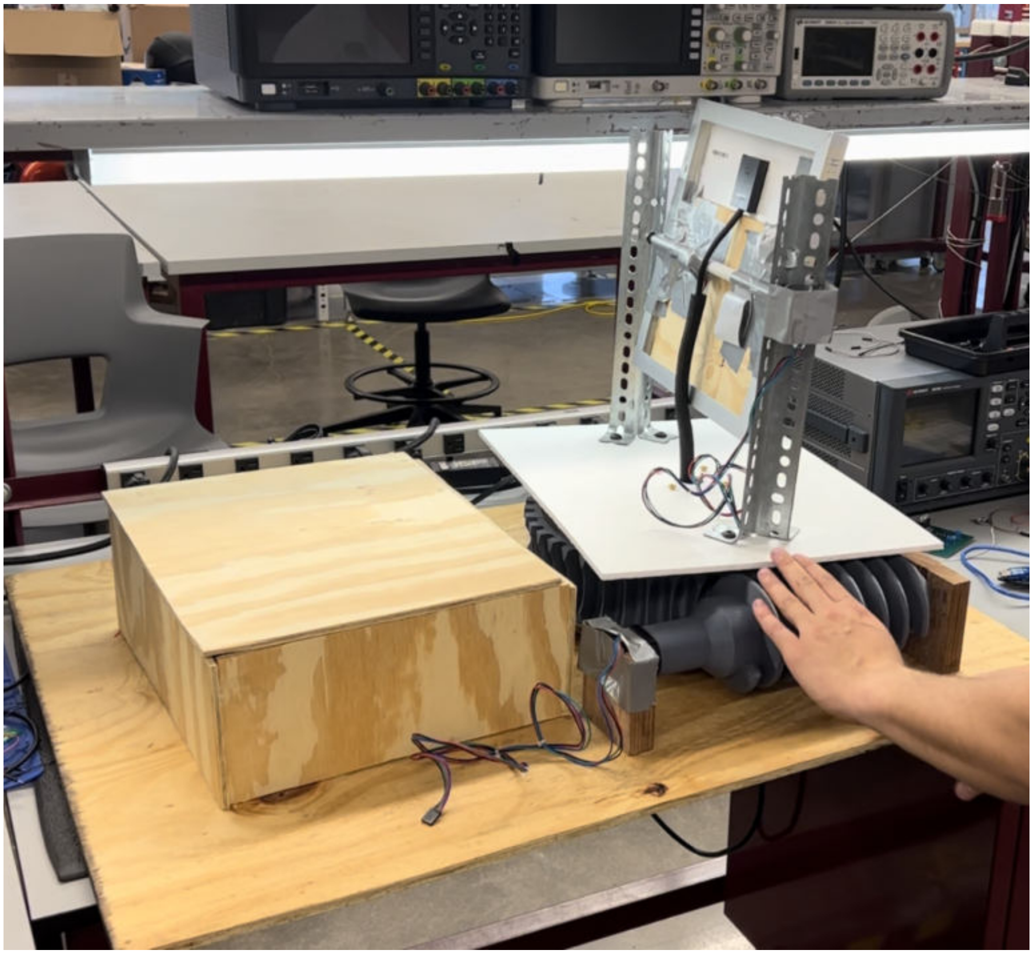
The main gear was too large for the 3d printers the team had available for this project as the extra large 3D printer was too expensive. To accommodate this issue, Talha and Cooper decided to print the main gear in 4 equal parts. This solved the printing problem but now the pieces had to be fastened together. The gear pieces were connected using small pieces of wood that had to be cut and fastened to the gear system using screws. Additionally in order to spin the main gear without friction with the ground a lazy susan was used and needed a smooth connection point. This was resolved using an additional piece of wood placed at the bottom of the contraption that was then cut to the proper circular shape as well fastened to the main gear. Lastly a platform was needed to connect the support beams of the solar panel to the main gear system. This was originally designed using wood but was later switched to a plastic board.

Figure 5: Solar Panel Support Beams



In order to connect the main gear system to the solar panel, two support beams were constructed to accomplish this. These beams were originally two long steel beams that needed to be cut to length. After the proper length was obtained, a slit was cut to fan out at the bottom so that the beam could be fastened to main gear. A hole was needed to be drilled into the vertex of the beams to allow for a ball bearing to be placed within the cut out. It was important not to drill the hole too big as well as to not drill the hole all the way through so the ball bearing can be secured without falling through the beam. This ball bearing setup was replicated on the other support beam as well.

Figure 6: Full Design Construction

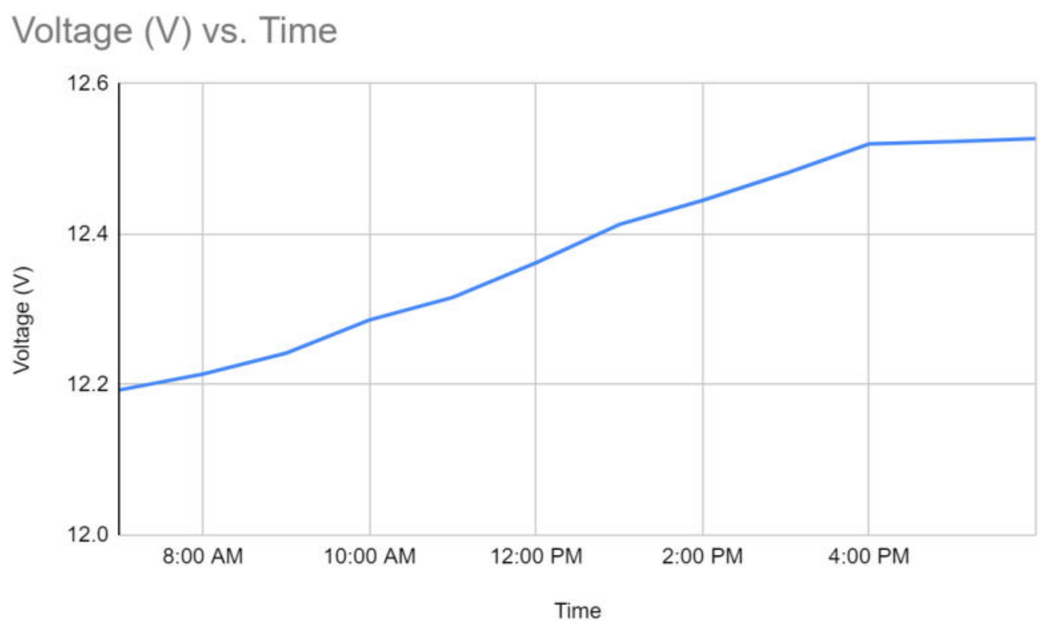


Our final built structure is presented here. It consists of the gear system, worm gear holds, main gear connecting platforms, solar panel support beams, axial rod that holds the solar panel, and the control box. The axial rod was the final part of our construction to be fabricated. It consisted of a cylindrical beam with holes drilled into the ends to allow for 3D printed components to attach to the ball bearings embedded into the support beams.

2.2. Data

The majority of our system was able to be integrated and the system was able to accomplish many of the fundamental goals of the project. The system was able to be implemented in a fixed tilt design to charge the battery according to *Figure 6.*

Figure 7: Solar Tracker Battery Charging Voltage



This graph shows how the voltage of the battery is increased as the day passes. The battery started at around 48% charged and rose to roughly 82% charged. When the battery is least charged it is expected that the voltage will increase the fastest. However, due to the rising level of the sun throughout the morning it led to a faster charge in the middle of the day even though the battery was more fully charged. Toward the end of the day when the battery was both well charged and the sun began to set, the rate at which the batteries voltage increased began to plateau.

Unfortunately, due to the lack of working motors for our system we were unable to verify the differences in a stationary vs a dual axis design.

2.2.1 Power and MCU Integration

The integration of the Power and the MCU subsystems was one of the most important aspects of the project.

The power subsystem designed and validated a working step down converter and Ac to DC rectifier. This means that the converted 120 volts AC was stepped down to 12 Volts DC which was fed into the Vinput of the Arduino Uno Rev 3. The same 12 Volts was also sent to the two L298N stepper motor controllers to power both 12 Volt rated stepper motors. The MCU subsystem had its motors mounted at the base of the mechanism. The stepper motor successfully moved the bottom gear (X axis movement) which validates our inter subsystem integration.The integration was successful and the video submitted to github shows the two subsystems being connected.

Figure 8: Power Output voltage with Load

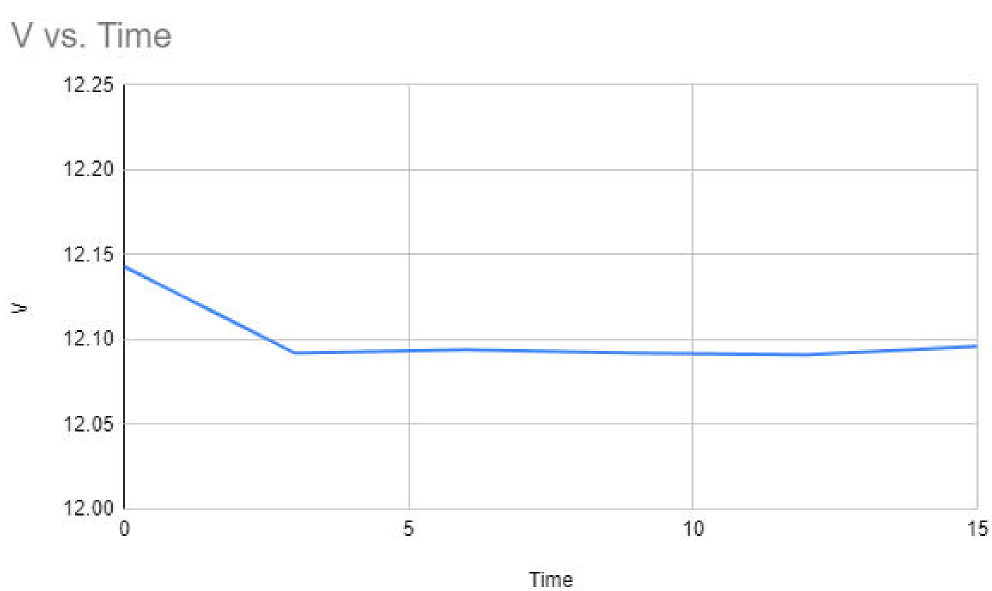


Figure 8 shows the voltage with the 3 loads connected and the stability of the output is shown. This was an important part as the lack of ripple would allow the system to have less risk of electrical failure in different environmental conditions.

2.2.2 Load and GUI Integration

Integration of the Load and GUI subsystems consisted of connecting the current and voltage sensors from the Load and GUI subsystem to different areas of the battery charger. The values obtained from the battery charger would later be displayed into the website. The load and GUI subsystems were able to successfully implement the battery charger and sensors, allowing for the battery charger to be turned off once the battery reaches close to full capacity. Below are graphs obtained from the sensor readings after integrating the battery charger system.

Figure 9: Buck-Boost Output Voltage

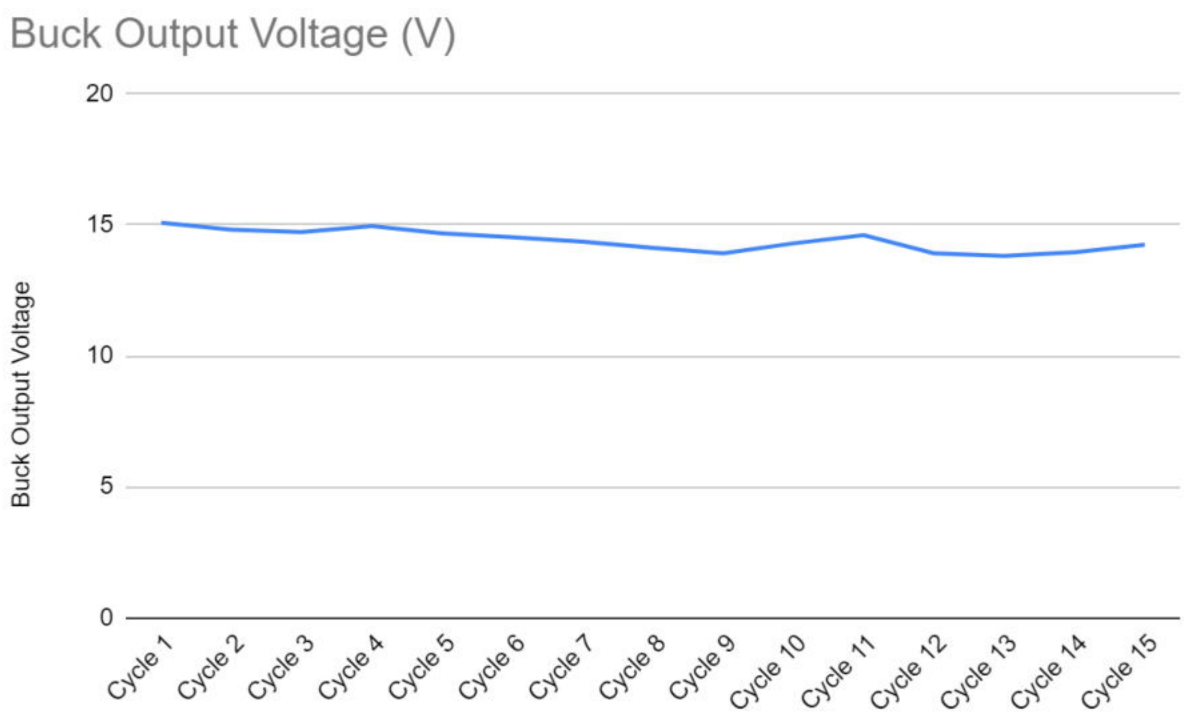


Figure 10: Voltage across Mosfet and Current through Battery Charger

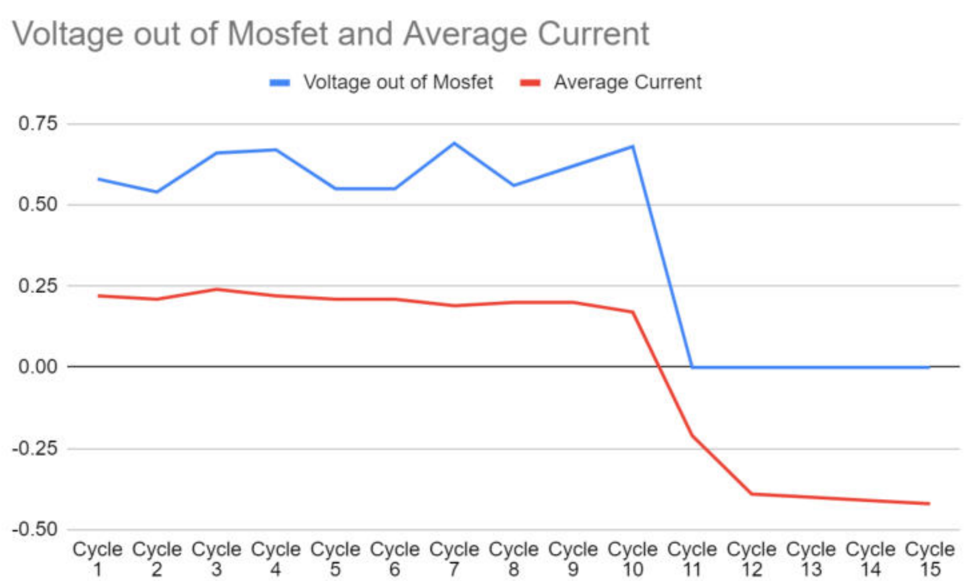
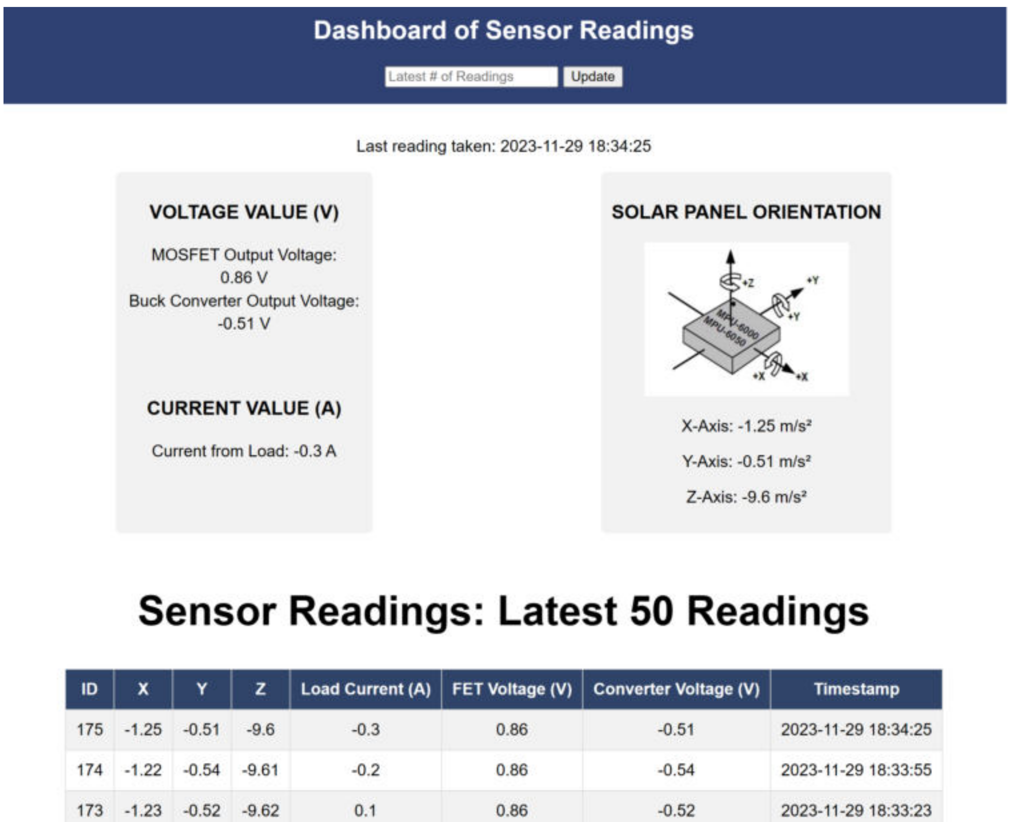


Figure 9 shows how the voltage of the buck-boost converter is unchanged when the battery charging is stopped at Cycle 10. The Mosfet is turned off and current is stopped from flowing through the pcb. Figure 10 shows how once the mosfet is turned off, the current through the battery charger is stopped from flowing in the positive direction charging the battery. The current actually shows a negative value because it is connected to a battery and the polarity is then flipped. This is also shown by the voltage across the mosfet decreasing to 0 V showing there is no longer a voltage drop across the mosfet which happens when no current flows through it.

Figure 11: Output of Data in Website



Dashboard of Sensor Readings

Latest # of Readings | Update

Last reading taken: 2023-11-29 18:34:25

VOLTAGE VALUE (V)

MOSFET Output Voltage:
0.86 V
Buck Converter Output Voltage:
-0.51 V

CURRENT VALUE (A)

Current from Load: -0.3 A

SOLAR PANEL ORIENTATION

X-Axis: -1.25 m/s²

Y-Axis: -0.51 m/s²

Z-Axis: -9.6 m/s²

Sensor Readings: Latest 50 Readings

ID	X	Y	Z	Load Current (A)	FET Voltage (V)	Converter Voltage (V)	Timestamp
175	-1.25	-0.51	-9.6	-0.3	0.86	-0.51	2023-11-29 18:34:25
174	-1.22	-0.54	-9.61	-0.2	0.86	-0.54	2023-11-29 18:33:55
173	-1.23	-0.52	-9.62	0.1	0.86	-0.52	2023-11-29 18:33:23

2.3. Conclusion

The Overall system performed fairly well. Unfortunately, we were not able to integrate every subsystem together fully. We dealt with quite a few unfortunate mishaps which included a change from using 2 DC motors to 2 stepper motors and PCB's not working as intended. Having to add the GUI subsystem at the beginning of the semester made all the subsystems fall behind as a plan had to be formed to effectively integrate that subsystem. Given that the MCU subsystem was behind schedule in terms of changing to the stepper motors, the power and load subsystems took the initiative on the mechanical design.

Over the course of the project many key improvements were made for the system to perform better. The MCU subsystem made the decision to switch to stepper motors. This allowed the motor to have better control of the axial rod so the solar panel would not need to constantly be powered as is the case with DC motors. Another improvement made was the switch of the load's buck converter to a buck-boost converter. This was made to allow for battery charging even when the voltage drops below 14.7V. The addition of a GUI subsystem was another large improvement implemented to allow for the user to have more understanding as to the functionality and usage of the solar tracker as a whole.

The physical system correctly operates and allows for movement of the solar panel. The dual axis nature of the design was able to be accomplished through a worm screw and worm gear design. This would accomplish the goal of having high torque at the expense of speed which was not necessary for our purposes as the sun would not be moving at a rapid pace. A prototype was 3D printed to be tested with the two stepper motors and was successful. The prototypes were then scaled by a factor of 4 in order to fit the needs of the dimensions within the project. After the worm screw and gear were printed, a lot of work was put in on the fabrication side to build the design robustly. The design was cut, screwed together and validated.

Some additional improvements we wish to make are the development of working stepper motors, a self powering system, and a more robust overall design to be better suited toward long run times to avoid the issues of PCB burninging. The self powering system would allow for the solar tracker to run off of the energy obtained through charging.

Overall, the Dual Axis Solar tracker performed fairly well. It was able to be powered by a wall outlet that would power a multitude of PCBs and sensors as well as the two motors. The solar panel was able to charge the battery and read the necessary sensor data into a graphical user interface for the user to use. Unfortunately, the motors were not able to properly control the dual axis nature of the solar tracker so the entire construction was unable to determine the benefits of the dual axis nature of the design.