

Loyola Marymount University
Department of Mechanical Engineering

AUTOMATED SOLAR PANEL FOR LMU CAMPUS

by
Jack Michaelis
Jack Leon
Nick Aiello
Mike Hennessy

for: Dr. Mustafa Mozael, Dr. Xiangyi Cheng, Gustavo Castillo
Department of Mechanical Engineering, Loyola Marymount University, Los Angeles, CA 90045
May 12, 2025

Acknowledgements

Thank you to LMU's Office of Sustainability (Green LMU), especially Lucy Renfrow, for providing information and support on this capstone project. Thank you to the electrical engineering department, especially electrical lab manager Leah Ghebranious for equipment and assistance with the electrical aspects of the project. Thank you to all the capstone advisors and visiting industry professionals for feedback and support.

Abstract

This Final Design Review outlines the design and implementation of an automated sun-tracking solar panel system for Loyola Marymount University's campus. The project aims to improve the efficiency of existing static solar panels by designing a new system of sun-tracking panels at a low cost and high power efficiency. The report covers the project's background, including LMU's current solar energy infrastructure and sustainability goals. It analyzes the advantages of automated solar panels over static ones, presenting comparative studies that demonstrate significant increases in energy capture. The report details the calculations for solar angle tracking, along with the mechanics and electronics for how the tracking device works. The design specifications focus on using linear actuators for panel movement and a Raspberry Pi for system control. The report also addresses considerations for weather-dependent operation and integration with existing infrastructure. The average percent difference in instantaneous power produced by the dynamic panel vs. the static panel was 36.5%. This project aligns with LMU's sustainability initiatives and aims to increase clean energy production on campus while providing a cost-effective solution for upgrading existing solar installations.

Table of Contents

Section	Page
Introduction	1
Problem Statement	2
Background	3
Design Specifications	11
Design Alternatives and Evaluation	12
Description of Design	16
Testing	18
Conclusions	19
References	21
Appendix A: Engineering Analysis	23
Appendix B: Manufacturing Plan	39
Appendix C: Project Schedule	44
Appendix D: Purchased Parts List	45
Appendix E: Engineering Drawings	52
Appendix F: Project Budget	57
Appendix G: Project Academic Assessment	65

Introduction

Energy defines the climate crisis as the main contributor to greenhouse gas emissions, but also the key to the solution. Fossil fuels make up over 75 percent of global greenhouse gas emissions and nearly 90 percent of all carbon dioxide emissions [1]. To avoid the destruction from climate collapse, emissions must be reduced immediately. To do so, investments must be made in alternative, clean energy sources. Loyola Marymount University [LMU] recognizes that over 50 percent of their carbon footprint comes from energy use, and increasing clean energy production moves them closer to carbon neutrality. LMU's dedication to sustainability is clear, with its many programs for making campus more sustainable. Green LMU, the school's office of sustainability, has an Alternative Energy initiative with a goal of implementing more clean energy sources on-campus [2]. Historically, LMU has been a leader in campus renewable energy sources, but have fallen behind because of their lack of upgrades since 2005. Specifically, they have almost 100,000 square feet of solar panels throughout campus, with the majority atop University Hall. Despite LMU's success in these initiatives in the past, they can, and must, improve in many areas. Such a solution is improving the infrastructure already present on campus. This project aims to create low cost, energy efficient solar panels that track the sun. Such tracking will increase the amount of power each panel can harvest which would improve LMU's sustainability initiative in current and future projects.

Problem Statement

This capstone team will design an automated, sun-tracking solar panel using similar solar cells to those currently on the roof of University Hall at LMU. The system will be designed so that the current panels can be converted to sun-tracking panels at a low cost and high power efficiency. Testing will be done on the automated panels to find how much more energy they are producing than non-automated panels, comparing their energy and cost efficiencies. Further, the panel's operational energy must not surpass the energy produced because that would be a net negative output. The panel must be designed with ease of use and installation considered for better chances of LMU administration to adopt the design. The experimental results along with the completed design will be shared with LMU administration and the Office of Sustainability with hopes that LMU adopts the new devices to increase the clean energy production on campus.

Background

Loyola Marymount University Clean Energy

Greenhouse gasses that trap heat around the Earth are largely produced through energy consumption by burning fossil fuels like coal, oil, and gas. Science shows that these emissions must be halved by 2030 and net-zero by 2050 in order to avoid total climate catastrophe. To do so, clean energy production must overtake the use of fossil fuels as the planet's main energy source, and fossil fuels currently account for more than 80 percent of global energy production. Further, 80 percent of people on the globe live in countries that are net importers of fossil fuels, so they are dependent on energy sources that come from outside of their home nation. Renewable energy sources are abundant and replenishing, provided by natural resources like the sun, wind, and water. Currently, renewables generate about 29 percent of the world's electricity. They are becoming more cost efficient, like how solar power's cost reduced by 85 percent through the 2010s. Moreover, the air pollution from fossil fuels cost \$2.9 trillion in health and economic costs in 2018. Renewables are healthier because they have limited impacts on air quality, releasing significantly less fine particulate matter and nitrogen dioxide than fossil fuels [1].

Historically, LMU has been a leader in renewable energy, and in sunny southern California, the feasibility for solar is clear. Former California Governor Arnold Schwarzenegger even signed a renewable energy initiative on the top of LMU's University Hall in 2009, depicting LMU's system as a model for future development (Figure 1). Between 2003 and 2005, LMU upgraded their energy production by installing \$4.5 million worth of solar panels on the roofs of three buildings: Gersten Pavilion, University Hall, and the Von der Ahe Library. This spread 81,000 square feet over the rooftops and generated 1,100,000 kilowatt hours (kWh) annually. At the time, this marked the largest solar system in all of Southern California, and the largest of any university in the world. They deployed this system for cheap, approximately \$325,000 total, because of rebates from the city. 10 years later, in 2015, LMU deployed 7,982 square feet of solar panels to the rooftop of the Life Sciences Building, which generates 200,000 kWh annually. In total, including other small solar energy producing systems on-campus, LMU produces 1,335,000 kWh per year (Figure 2), and the rooftop systems simultaneously insulate and protect the buildings from UV radiation, saving heat and maintenance costs. It is estimated that the solar systems save the university more than \$210,000 per year [2].



Figure 1: Gov. Arnold Schwarzenegger and Secretary of the Interior Ken Salazar signing energy initiative atop LMU's University Hall in 2009 [2].

	LIFE SCIENCE	VON DER AHE	UNIVERSITY HALL	GERSTEN PAVILLION	ZON UMBRELLAS	CAMPUS TOTAL
INSTALL DATE	2/16/2016	4/1/2003	4/1/2003	5/1/2005	Fall 2014	N/A
SQUARE FOOTAGE	7,983	13,540	43,330	25,000	N/A	89,853
TYPE OF MODULE	Sun supreme SNPM GxB - 280 watt	Shell Solar SP-75 - 75 watt	Shell Solar SP-70 - 70 watt	Double Shell SP- 70 & SQ- 85 - 70 watt	12 Powersol Umbrella Off-Grid Charging Stations - 54 watts each	N/A
NUMBER OF PANELS	451	1,664	5,330	1,209	192	8,846
ANNUAL YIELD KWH (CALCULATED)	200,690	194,050	583,715	355,706	1,011	1,335,172

Figure 2: Solar production on LMU's campus [2].

LMU participates in a report called The Sustainability Tracking, Assessment & Rating System from AASHE that ranks colleges and universities on their sustainability performance. LMU currently ranks gold, the second highest ranking, scoring 81.82 points out of 100. They were rewarded 0.49 out of 4.00 points for the category titled Clean and Renewable Energy. Their percentage of total energy consumption from clean and renewable sources is listed as 12.24%, with their solar photovoltaic energy making up 14.10% of all electricity use [3]. The Office of Sustainability sees this ranking as a need to be improved upon. Moreover, implementing more clean energy systems could provide financial advantage through President Biden's Inflation Reduction Act (IRA). The law allows clean energy tax credits for higher education institutions that do not pay Federal income tax, which qualifies LMU, through a feature called Elective Pay. Universities can get up to \$27.50 per MWh "for qualified clean energy produced and sold if prevailing wage and apprenticeship requirements are met", and they can get between 6-30% credit for qualified investments [4]. The new programs incentivize institutions like LMU to improve the sustainability on campus, having both climate benefits and financial savings.

The solar panels that LMU currently houses on the roof of University Hall are Shell SP70 Photovoltaic Solar modules which contain 36 series connected 125 x 125 mm PowerMax® monocrystalline silicon solar cells (Figure 3). The frame is made of torsion and corrosion-resistant anodized aluminum. The surface is made with highly transparent tempered glass on a pyramidal textured surface, which protects well from weather. At its maximum, the panel generates a power of 70 watts at 16.5 volts. They are expected to last 25 years, and were designed for grid connection and industrial applications [5]. See Figure 4 for the exact dimensions of each panel, and Figure 5 for the panel's power data.

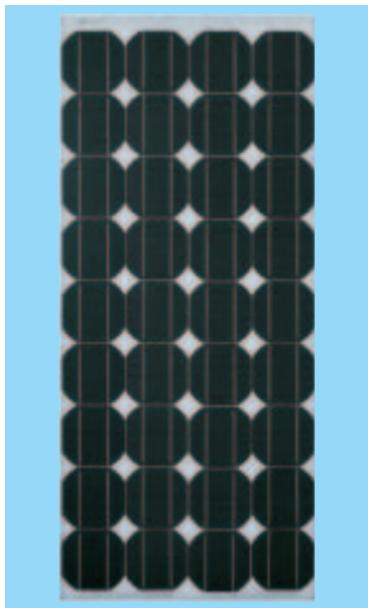
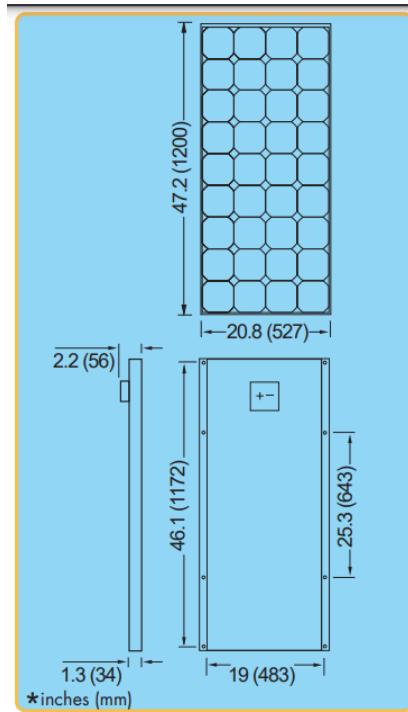


Figure 3: Shell SP70 Photovoltaic Solar module [5].



Outside dimensions (in)	47.2 x 20.8
Thickness (inc. junction box) (in)	2.2
Thickness (exc. junction box) (in)	1.3
Weight (lbs)	16.7

Figure 4: Dimensions of the Shell SP70 Photovoltaic Solar module [5].

Data at Standard Test Conditions (STC)		
STC: irradiance level 1000W/m ² , spectrum AM 1.5 and cell temperature 25°C		
Rated power	P _r	70W
Peak power	P _{mpp}	70W
Peak power voltage	V _{mpp}	8.25*/16.5V
Peak power current	I _{mpp}	8.50*/4.25A
Open circuit voltage	V _{oc}	10.7*/21.4V
Short circuit current	I _{sc}	9.4*/4.7A
Series fuse rating		15A
Minimum peak power	P _{mpp min}	65W
The abbreviation 'mpp' stands for Maximum Power Point.		

Typical data at Nominal Operating Cell Temperature (NOCT) conditions		
NOCT: 800W/m ² irradiance level, AM 1.5 spectrum, wind velocity 1m/s, T _{amb} 20°C		
Temperature	T _{NOCT}	45°C
Mpp power	P _{mpp}	51W
Mpp voltage	V _{mpp}	7.55*/15.1V
Open circuit voltage	V _{oc}	9.8*/19.6V
Short circuit current	I _{sc}	7.6*/3.8A

* The Shell SP70 may be reconfigured in the field for 6V operation

Figure 5: The power data of the Shell SP70 Photovoltaic Solar module in standard test conditions and typical data at nominal temperature [5].

Automatic vs. Static Solar Panels

Fixed solar panels are a popular choice for renewable energy due to their low maintenance, cost-effectiveness, and ease of installation. Common configurations include fixed-angle, vertical, and season-adjusted fixed-angle installations, each designed to optimize solar capture based on geographic and seasonal conditions [6]. They typically have a lifespan of 25-30 years with minimal degradation, making them a reliable long-term investment. However, fixed panels tend to be less efficient than tracking systems because they do not adjust their angle for optimal power generation. Their fixed angle also makes them more dependent on weather conditions, which can impact energy production. Studies comparing fixed panels to automated systems help users determine the best option for their specific needs and applications, balancing advantages and limitations based on individual circumstances.

A study by the Artificial Intelligence Engineering Department at Alayan University in Nasiriyah, Iraq, compares the energy output of a fixed-angle solar panel with that of a custom-designed automated dual-axis solar panel. Over a three-day period, data was collected continuously from both types of panels to monitor their voltage output, aiming to compare the results afterward. The automated solar tracking system showed impressive voltage stability, consistently maintaining a range of 18 V to 20 V throughout the day [7]. Figures 6 and 7 below display the data gathered from both the static and automated solar panels. As illustrated, the static solar panel struggled to maintain a steady voltage, fluctuating between 10 V and 20 V.

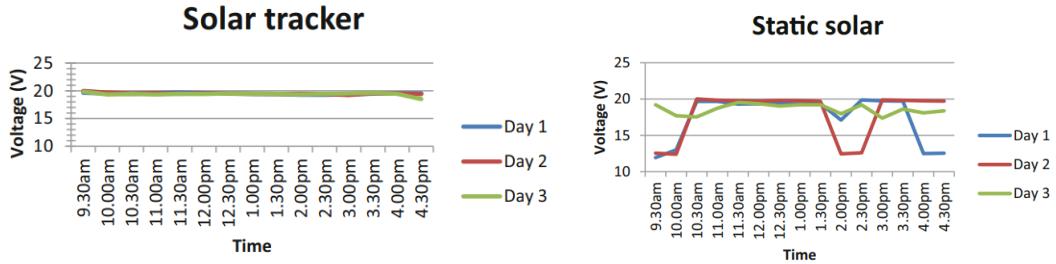


Figure 6: Automated solar panel data [7].

Figure 7: Fixed solar panel data [7].

The tracking system recorded a maximum irradiance of 1555 W/m^2 , compared to just 1460 W/m^2 for the static solar system. This substantial difference in irradiance levels is a testament to the effectiveness of our solar tracking system in maximizing solar energy absorption [7]. The empirical findings show a notable increase in irradiance levels (averaging 15%), consistent voltage output (18–20 V), and better temperature control (53.4°C compared to 59.5°C for static panels).

A similar study conducted in collaboration with The Hashemite University in Zarqa, Jordan, compared the performance and overall economic impact of fixed versus tracking photovoltaic systems. Both systems have a capacity of 7.98 kWp and were monitored for a full year, from February 9, 2014, to February 8, 2015. The results indicate that the annual production of the tracking system is 31.29% higher than that of the fixed system [8]. In Figure 8 below, the instantaneous irradiance for both systems on July 15, 2014, demonstrates that the automated system achieves a higher daily average irradiance compared to the fixed solar panel. The tracking system maintains a plateau for a significant portion of the day's sunlight, whereas the fixed system peaks only at noon. Figure 9 illustrates the daily average irradiances for both systems for each month [8].

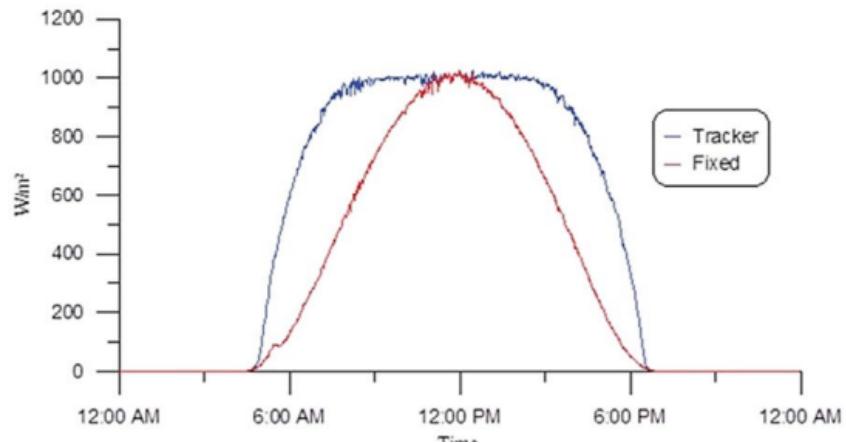


Figure 8: Instantaneous irradiance on fixed and tracking systems [8].

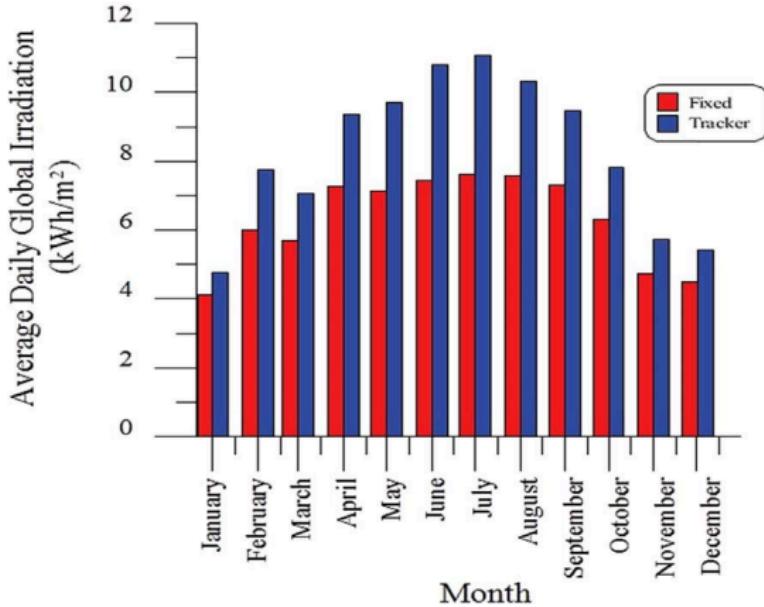


Figure 9: Daily average irradiance for each month on fixed and tracking systems in kWh/m² [8].

The figures demonstrate the superior performance of tracking photovoltaic systems over fixed systems, with Figure 8 showing higher and more consistent daily irradiance levels for tracking systems, and Figure 9 extending this advantage to a monthly scale. The tracking system's ability to dynamically adjust to the sun's position results in significantly higher energy capture, particularly during summer months, as evidenced by the 31.29% increase in annual production. These visualizations effectively highlight the key advantage of tracking systems in maximizing solar energy capture, particularly in high solar potential locations.

To further expand upon the need to collect weather data and conditions, the design of the solar panels could lead to inefficiencies or damage to the panels themselves in certain common weather patterns. Since the panel design inherently needs energy to function, an overcast or rainy day could lead to the panel's tilting operations drawing more power than the panel produces. As shown in Figure 10, the power generated by the panels could decrease by ~50%. If such a decrease occurs leading to energy inefficiencies, the system would need to know if it is going to rain or be overcast. In addition, since the panels are raised off the ground, as opposed to traditional solar panels, high winds could lead to tipping. In such situations, a system to either lower the panel and lay it flat would need to be considered.

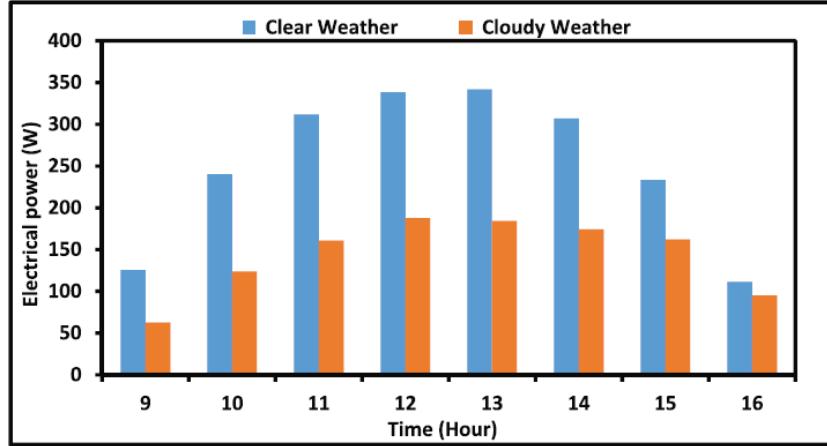


Figure 10: Electrical power generated in clear and cloudy weather [15].

Solar Angle Calculations

Two critical angles are used to orient solar panels, the azimuth angle and elevation angle. These angles are used to describe the sun's position in the sky in relation to time and location. Since these angles change throughout the day and depend on the specific location of the solar panel, these angles must be recalculated whenever a dual axis system moves. The elevation angle describes the elevation of the sun from the horizon line, as shown in Figure 11. The azimuth angle describes the compass direction of the sun, shown in Figure 12. The azimuth angle is what will be used to orient the E-W actuator and the elevation angle will be used for the N-S actuator. To find these angles throughout the day, the local solar time must be found, which is used to calculate the elevation and azimuth angles. To find the local solar time, the following equations are used [16]:

$$LSTM = 15^\circ \cdot \Delta T_{GMT} \quad [1]$$

$$EoT = 9.87\sin(2B) - 7.53\cos(B) - 1.5\sin(B) \quad [2]$$

$$B = \frac{360}{365}(d - 81) \quad [3]$$

$$TC = 4(Longitude - LSTM) + EoT \quad [4]$$

$$LST = LT + \frac{TC}{60} \quad [5]$$

Where LSTM is the Local Standard Time Meridian, ΔT_{GMT} is the difference of the Local Time from Greenwich Mean Time in hours, EoT is the equation of time, TC is the time correction factor, and LST is local solar time.

To find the elevation angle, the following equations are used:

$$HRA = 15^\circ(LST - 12) \quad [6]$$

$$\delta = 23.45^\circ\sin(B) \quad [7]$$

$$\alpha = \sin^{-1}[\sin \delta \sin \varphi + \cos \delta \cos \varphi \cos(HRA)] \quad [8]$$

Where HRA is the hour angle, δ is the declination angle, α is the altitude angle, and φ is latitude.

To find the azimuth angle, the following equation is used:

$$Azimuth = \cos^{-1}\left[\frac{\sin \delta \cos \varphi - \cos \delta \sin \varphi \cos(HRA)}{\cos \alpha}\right] \quad [9]$$

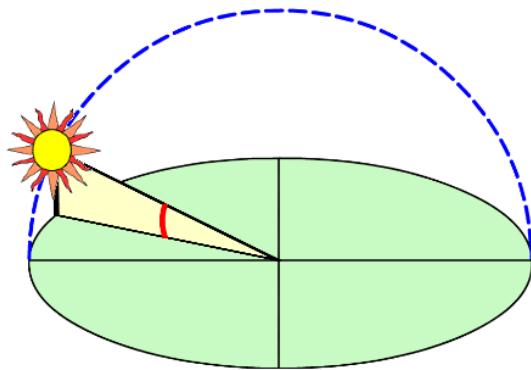


Figure 11: Visual Depiction of the Elevation Angle [16].

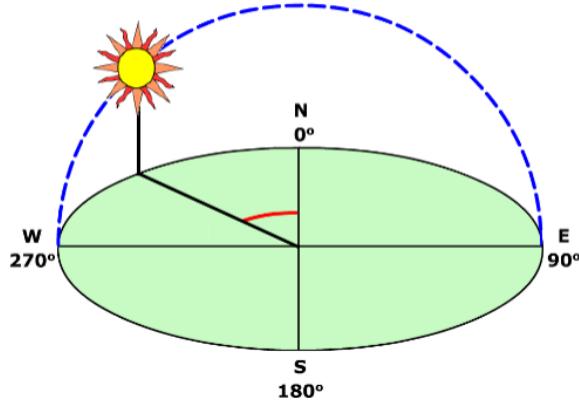


Figure 12: Visual Depiction of the Azimuth Angle [16].

Solar Panel Standards

Standards will play a crucial role throughout the design process for two main reasons. First, they ensure proper dimensioning and tolerancing, guaranteeing that all bolts, screws, and clearance holes meet the necessary specifications for fabrication. By utilizing ASME Y14.5 [17], we will create precise engineering drawings for each part according to these standards. This accuracy is essential, particularly for components that require machining, as it directly impacts the functionality of the final product. The second standard we will apply is Section 6.5.15 of ASCE 7 [18], which outlines design wind loads for various structures. This section will help us determine the wind load values for our solar tracking system. By considering factors such as panel size, geographic location, and height, we can calculate the theoretical forces acting on the panels, which will then inform our FEA simulations to evaluate the safety of components like hinges, posts, and frame members.

In the later stages of the project, we will also need to refer to IEEE 1547 and the National Electrical Code (NEC) to ensure our solar panel design can be effectively integrated into the electrical grid on the roof of University Hall. These standards will guide us in establishing safe and reliable interconnections, ensuring that our system complies with regulatory requirements and functions seamlessly within the existing infrastructure [19].

Design Specifications

1. The tracking-enabled solar panel system is projected to deliver positive daily net energy output.
2. The system is designed to achieve a return on investment (ROI) with cumulative energy savings projected to exceed the initial capital expenditure.
3. Solar panel design incorporates dual-axis tilting capability up to 45° for maximized solar exposure.
4. Actuator systems are engineered to withstand forces of up to 100 lbs, ensuring operational resilience and longevity.
5. The integrated control systems are programmed to operate at regular intervals, maintaining optimal panel alignment and maximizing energy yield.
6. A Raspberry Pi serves as the central processing unit for the system, managing core functionalities, and an Arduino is used for secondary functionalities, like controlling the actuators.
7. The system includes weather monitoring through a GPS and response capabilities to adapt to changing environmental conditions.
8. An energy-conserving sleep mode is implemented, which suspends non-essential operations while maintaining critical functions.
9. System should have easy installation in addition to quick, infrequent maintenance.
10. The system must abide by ASME Y-14.5 standards.

Design Alternatives and Evaluation

When designing a system to optimize the tilt of solar panels, three potential mechanisms are widely used - linear actuators, a gear system, and gyroscopes. All technologies offer distinct advantages, but after a detailed comparison shown in Table 1, linear actuators emerged as the superior choice.

Table 1: Movement system decision matrix.

Parameters	Linear Actuator (PA-14-18-150) [9]	Worm Gear System	Heavy Duty Gimbal Gyroscope [10] [11]	Comments
Load Capacity	150 lbs (667 N)	~500lbs - ~1000lbs (Depends on gear system)	~500 lbs	Although the heavy duty gimbal system can support higher loads, both mechanisms display more than enough load capacity for our project.
Stroke Length	18 inch (203 mm)	N/A	N/A	Stroke length is not applicable to the gyroscope as it only offers rotational movement. A stroke length of 18 inches is a tilt of 39.49 degrees. The Department of Energy recommends maximum tilt angle to be within 30 - 45 degrees. [12]
Precision	± 1.5 mm	Extremely Precise	± 0.01 angular accuracy	Both the gyroscope and the actuators allow for smooth, precise movements to adjust the panels tilt.
Energy Consumption	12V DC, 5A (Peak)	Motors: 12V - 48V (Would be on higher side)	Gyroscope: 3.3V DC, 0.9mA	The gimbal gyroscope system will consume more

			Motors: 12V - 48V	energy overall especially with the higher-powered arduino motors.
Durability	IP66 rating, resistant to dust and water	High durability (Especially in enclosed system)	Dependent on gimbal's material but often require extra protection	Linear actuators come with built-in environmental protection while gimbal and gear systems don't.
Cost	130\$	~\$200-\$600	1,500\$-2,000\$	The gimbal system is significantly more expensive than the alternatives.
Ease of Installation	Easy to install with mounting brackets	Medium to High (If enclosed much tougher)	Complex; requires calibration and multiple electrical and mechanical components	Linear actuators are more straightforward to install and easier to do maintenance.

Linear actuators were selected for tilting the solar panels due to their balance of precision, simplicity, and reliability. Unlike gyroscope-based systems, which require multiple components like motor controllers and high-torque motors, linear actuators offer a direct and efficient method for adjusting the tilt of solar panels. The PA-14-12-200 actuator provides the necessary load capacity (150 lbs), stroke length (200 mm), and high precision (± 1.5 mm), all while being easy to install and integrate with standard solar tracking control systems. Its IP66 rating ensures durability in outdoor environments, protecting it from dust and water, and its low maintenance requirements make it ideal for long-term operation with minimal upkeep [9]. Compared to the more complex and expensive alternatives, such as a gimbal gyroscope system or gear-driven setups, actuators offer a cost-effective and energy-efficient solution with reliable performance across a wide range of applications.

The PA-14-18-150 linear actuator remains a more efficient and cost-effective solution for tilting solar panels, especially when the focus is on durability, ease of installation, and low maintenance.

Next, two products are commonly used to control the panel: Arduino and Raspberry Pi. Shown in Table 2, Raspberry Pi is the superior choice for the central processing unit:

Table 2: Control system decision matrix.

Parameters	Arduino [13]	Raspberry Pi [14]	Comments
Energy Efficiency	7-12V, can be run off of a computer's USB.	Needs a dedicated power supply and uses more power when compared to the Arduino.	Raspberry Pi uses a lot more power but is much more powerful as a result.
Ease of Use	Restricted to specific IDE and coding language.	Supports Java, Python, and C++ and uses the Linux OS.	Arduino is much more simple but restricted. Raspberry Pi is a lot more open and would require less concessions to use.
Cost	\$28	\$40-80	Arduino is cheaper.
Tech Specs	8 bit microcontroller with 32 kB flash memory + 2 kB of SRAM @ 8-400 MHz	64 bit quad core processor w/ 1-8 GB of RAM @ 700 MHz-1.8 GHz	Raspberry Pi is faster and has better memory and processor.
Degree of Control	Single Core System	Multi Core System	The single core will limit the degree the system will be able to function.
Internet	Extra Component	Integrated	Raspberry Pi is the cheaper option.
GPS	Extra Component	Extra Component	
Database	Possible, but complicated and requires additional software.	Easily possible and commonly used.	Raspberry Pi will allow extra functions, like response to weather forecasts, easily possible.

While the Arduino does use less power and is cheaper, the product is much too restrictive for the additional features planned to integrate outside of simply moving the panel. The Raspberry Pi excels in software oriented uses, on account of its increased processing power, speed, and integrated OS system. While there are no plans for the panel to use large-scale software capabilities, the use of connecting to and analyzing publicly available weather databases is of high interest. Thus, the use of a GPS or IP locating API would be necessary to provide weather location. In addition, the lack of multiple cores on the Arduino means that it is

simply more limited in its processing power. Thus, the Raspberry Pi's swathe of features outweighs the additional cost and energy required.

Finally, there are two options for data collection techniques the control system will respond to. Using a photoresistor array to directly control the system or using solar equations throughout the year to find a tilting speed. The solar equations ultimately wins out:

Table 3: Data collection technique decision matrix.

Parameters	Photoresistor Array	Solar Equations	Comments
Cost	Would cost ~\$7	No upfront cost	Solar data is the cheapest option
Accuracy	Extremely accurate, vulnerable to obstructions.	Less accurate, more of an estimate. Not vulnerable to obstructions.	Photoresistors are more accurate but vulnerable to obstructions.
Ease of Implementation	Easy	Easy, all are equation based to provide accurate enough data.	Both are easy, however the solar data does not necessitate installation of a subsystem.
Accessibility of Data	Easily accessible	Easily accessible	Both are equally accessible.

While the photoresistor array is more straightforward and more accurate, the solar equations used to calculate the optimal tilt of the panel is ultimately cheaper and invulnerable to obstructions. This invulnerability is very important since this system is meant to be mostly autonomous, only needing regular maintenance to keep the system going. Using a program developed in python, the Raspberry Pi can easily calculate what angle it needs for each axis whenever it is time for it to adjust. This approach will save on cost and get rid of a large design risk that is prone to failure. Since any amount of obstruction of the photoresistor array would lead to inaccurate tilting, possibly leading to the array tilting away from the sun, avoiding the use of this approach leads to an overall simplification of the necessary failure mechanisms in place and less oversight when in prolonged use.

Description of Design

The panels will be placed in a 5 by 13 array, with 26 inches separation by row, and 31.5 inches separation by column. The calculations of separation between panels used an approach based on the cartesian sun path chart for West Los Angeles to measure shadow lengths, as seen in Figure 13 [20]. See appendix A for calculations.

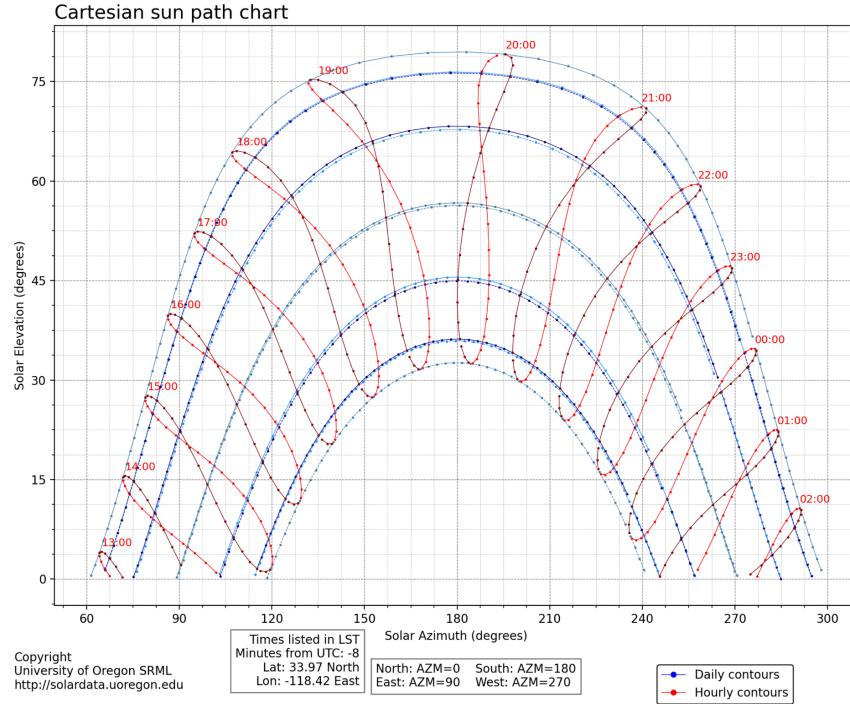


Figure 13: The cartesian sun path chart in Los Angeles, CA [21].

For the control system, a more expansive solution is needed than a single Raspberry Pi. Due to the Raspberry Pi's inherent limitation from the number of available I/O pins, use of this system in a solar panel array is extremely limited. Thus, for scalability and cost effectiveness, a mixed system combining Arduinos and a Raspberry Pi will be used. The mixed system is depicted in Figure 14, which takes advantage of the low cost and power draw of Arduinos by using them to control a number of actuators while the Raspberry Pi's superior processing power is used to control the entire system while dealing with interacting with a weather API, and outputting critical information to a LCD. This type of system also allows for the Raspberry Pi to control when the Arduinos are turned on or off, lessening the power draw of the system when it is not collecting power.

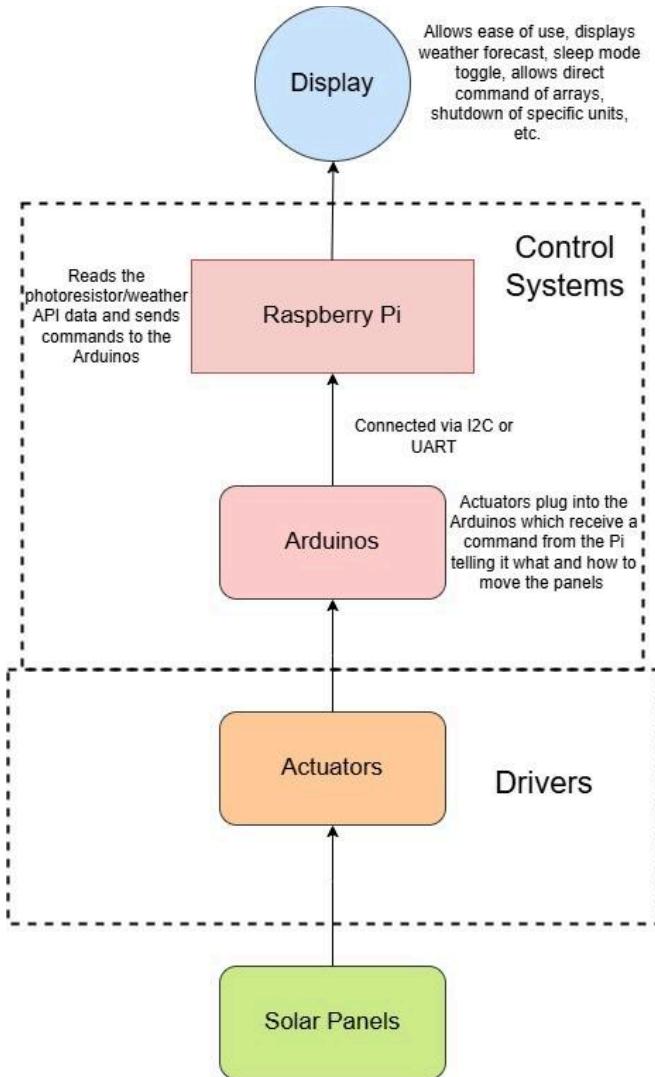


Figure 14: Mixed system layout.

The model consists of a rectangular main base that holds the system up. A bar is attached, which holds the panel frame and one actuator. Two actuators are used, one for the east-west axis and one for the north-south axis. The base is made from 1020 steel, and the panel frame is made from aluminum 6061. The drawings for the model can be found in Appendix D.

This project is designed around ethics. Specifically, the objective is to increase the production of renewable energy on LMU's campus. By doing so, the university will reduce carbon emissions from other energy sources. Thus, there are positive environmental impacts. The social impacts include educating students about the importance of solar energy and how it can be easily implemented at institutions like LMU. Other impacts include economically by supporting solar panel suppliers and global impacts by encouraging the use of solar panels at Universities and institutions around the world.

Testing

Testing will begin after the solar panel and electrical parts necessary to build the system are delivered and assembled. After returning from winter break, the team will split into two groups: mechanical and electrical. The mechanical group will complete the machining and assembly of the metal pieces and physical structure, while the electrical group will assemble the hardware and implement the software for the Raspberry Pi, Arduino, actuators, and complementary items. To keep as controlled of an environment as possible, the dynamic and static panels will be tested on simultaneous days. For example, data will be collected for the dynamic panel on the odd days of March (March 1, 3, 5, etc.), and data will be collected for the static panel on the even days of March (March 2, 4, 6, etc.). This will be done by replacing the software to match the day and gathering the data each day. The static and dual axis automated panels cannot be tested under the exact same conditions because there is only one panel available for use. Thus, the change in sunlight, clouds, amount of daylight, and weather conditions by the day will be considered when analyzing the data. If necessary, the data from one set will be multiplied by a factor in order to control the data analysis as best as possible. That factor will be determined after analyzing the weather patterns over the dates that testing took place for each panel. Variables that will determine this factor include, but are not limited to, length of day, position of sun in the sky, and cloudiness.

The apparatus will be set up on the sun roof of the Life Sciences Building, with access granted through the capstone advisor and Seaver College. No external power sources are necessary because the panel powers its own system. Caution tape and a warning sign will be placed around the apparatus for added safety measures. The apparatus will be checked regularly by one of four members of the capstone team. Data will be collected and stored directly using the Raspberry Pi and voltage and current sensors. Readings will be taken from directly after the solar panel, directly before the battery, and directly before the dual axis system.

Prior to the dual axis automated panel being launched for roof top testing, other tests will be conducted to test its strength and durability. Load tests will be done to simulate wind conditions at varying angles. This will be carried out by manually putting force on the panel at every 5 degrees between -45 and 45 degrees. The preliminary testing will also be used to verify the machine's control systems, ensuring that the actuators and other electrical mechanisms work fluidly.

Two major variables will be analyzed: the amount of energy produced, and the amount of energy used. The amount of energy used will be virtually zero for the static panel because it is not rotating, but it is expected to be significant (~6% of energy gained) for the dual axis automated panel. The energy produced by each hour of the day will be analyzed to see which hours have maximum production and should be prioritized when rotating the panel. Using these data, the efficiency of each panel will be calculated and compared.

Conclusions

Testing frequency occurred less than expected because of electrical component failures. Specifically, the Raspberry Pi broke three times. This likely occurred because the Raspberry Pi could not keep a constant voltage; rather, the voltage fluctuated ± 0.5 volts. After the third time it broke, it was decided to not use the Raspberry Pi and instead use the Arduino Uno as the central processing unit. This was possible because the system had just one panel. In a scaled system, the Raspberry Pi would regulate an array of panels, but this system was a subscale. After concluding that the Raspberry Pi is not fit for this array, it is recommended that a scaled system uses a stronger computer. Further, various sensors died multiple times. Two current sensors died, a motor driver died, and an analog to digital converter died. It is recommended that future projects buy more durable parts to withstand high voltage and current. This may increase costs, but they are necessary for the longevity of the system. Due to these malfunctions, extra parts were bought, and testing was delayed. Every time a part broke, there was a wait period for the replacement to be shipped, and that created time where testing could not occur. Specifically, only two days of testing were possible: one day with a static panel, and one day with a dynamic panel. It is recommended to order parts earlier, so that when parts break, there is leeway in the project schedule to replace them or find other solutions. Though this was less data than expected, there were still some results to analyze.

The completed, physical system can be seen in Figure 15. Further, figures 16 and 17 show the results from the two days of testing. The dynamic panel consistently produced more power than the static panel. However, the weather was cloudier on the day that the static panel was tested, which likely lowered the production. The average percent difference in instantaneous power produced by the dynamic panel vs. the static panel was 36.5%, a substantial margin and larger than the expected 30%. It was expected that the percent difference would be smaller towards the middle of the day and higher at the beginning and end of the day, and that hypothesis partially proved true. The percent difference was highest at 7:00 pm and second highest at 6:00 pm. Moreover, the total power produced by the dynamic and static panels in one day were 502.5 and 334.9 Whr, respectively, which is a percent increase of about 50%. The power that the actuators consume when they are turned on once every hour was 0.1 Whr, which is negligible compared to that which was produced. The difference of 167.6 Whr could be used to fully charge a smartphone about 14 times.



Figure 15: Completed system being tested on the sun roof of the Life Sciences Building.

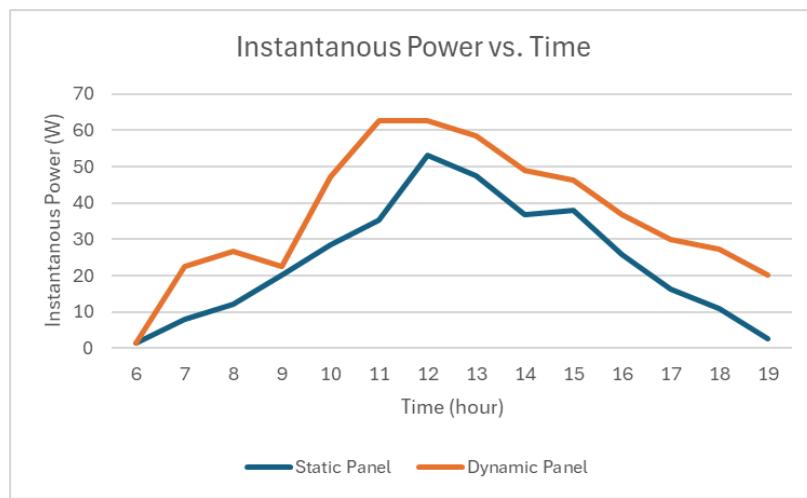


Figure 16: Graph of instantaneous power over time for the static and dynamic panels.

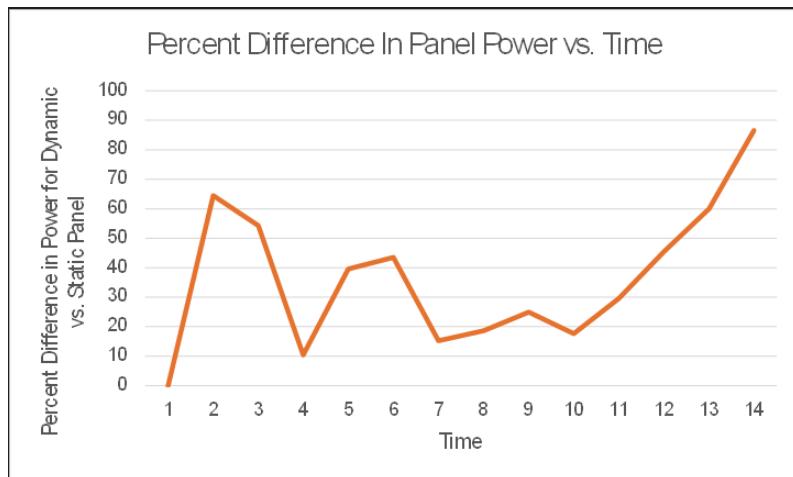


Figure 17: Graph of percent difference over time for the dynamic vs. static panels.

References

- [1] United Nations. (n.d.). *Renewable energy – powering a safer future*. United Nations Climate Action. <https://www.un.org/en/climatechange/raising-ambition/renewable-energy>
- [2] Loyola Marymount University. (n.d.). *Alternative Energy*. Green LMU. <https://admin.lmu.edu/greenlmu/initiatives/alternativeenergy/>
- [3] STARS: A Program of AASHE. (n.d.). *Loyola Marymount University*. The Sustainability Tracking, Assessment and Rating System. <https://reports.aashe.org/institutions/loyola-marymount-university-ca/report/2024-03-01/AC/curriculum/AC-2/>
- [4] *A Clean Energy Future for America's Colleges and Universities*. U.S. Department of the Treasury. (2024, February 20). <https://home.treasury.gov/news/featured-stories/a-clean-energy-future-for-americas-colleges-and-universities>
- [5] *Shell SP70 Photovoltaic Solar Module Product Information Sheet*. Shell Solar. https://reenergyhub.com/files/hersteller/Shell_Solar/pdf/Shell_Solar_SP170_EN.pdf
- [6] Shang, H.; Shen, W. Design and Implementation of a Dual-Axis Solar Tracking System. *Energies* 2023, 16, 6330. <https://www.mdpi.com/1996-1073/16/17/6330>.
- [7] Ahmed, Saadaldeen Rashid, et al. “Optimizing Solar Energy Efficiency Through Automatic Solar Tracking Systems.” *Springer Link*, https://link.springer.com/chapter/10.1007/978-3-031-62871-9_23. Accessed 15 Sept. 2024.
- [8] Hammad, Bashar, et al. “Performance and Economic Comparison of Fixed and Tracking Photovoltaic Systems in Jordan.” *Renewable and Sustainable Energy Reviews*, Pergamon, 1 June 2017, www.sciencedirect.com/science/article/pii/S1364032117308936.
- [9] “Miniature Electric Linear Actuator.” *Progressive Automations*, www.progressiveautomations.com/products/mini-linear-actuator?variant=18277278875715.
- [10] “Adjustable Gimbal Mount with Mounting Track.” *Traxstech Corporation*, traxstech.com/adjustable-gimbal-mount-with-mounting-track/.
- [11] *STEPPERONLINE High Torque NEMA 23 CNC Stepper Motor 114mm 425oz.in/3nm CNC Mill Lathe Router - Amazon.Com*, www.amazon.com/Torque-Stepper-Motor-425oz-Router/dp/B00PNEPW4C.
- [12] *Homeowner's Guide to Going Solar | Department of Energy*, www.energy.gov/eere/solar/homeowners-guide-going-solar.
- [13] *Arduino Uno REV3 [A000066]*, Amazon, https://www.amazon.com/Arduino-A000066-ARDUINO-UNO-R3/dp/B008GRTSV6/ref=asc_df_B008GRTSV6/?tag=hyprod-20&linkCode=df0&hvadid=693421862574&hvpos=&hvnetw=g&hvrand=15470078635019599228&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1013962&hvtargid=pla-457497319401&mcid=8d4415853f19330eb6cb8c1e7f18a8ed&th=1.
- [14] *Raspberry Pi 5, CanaKit*, <https://www.canakit.com/raspberry-pi-5-8gb.html>.
- [15] Merie, F. H., & Ahmed, O. K. (2024). Experimental assessment of the performance of the PV/solar chimney under the cloudy weather. *Results in Engineering*, 23, 102605. <https://doi.org/10.1016/j.rineng.2024.102605>.
- [16] “2.4 Terrestrial Solar Radiation,” *PVE Education*, <https://www.pveducation.org/pvcdrom/terrestrial-solar-radiation>. Accessed 21 November 2024.

- [17] “Y14.5 Dimensioning and Tolerancing.” *ASME*, www.asme.org/codes-standards/find-codes-standards/y14-5-dimensioning-tolerancing. Accessed 25 Sept. 2024.
- [18] “ASCE 7 Standard.” *ASCE American Society of Civil Engineers*, www.asce.org/publications-and-news/asce-7. Accessed 25 Sept. 2024.
- [19] *IEEE 1547 and 2030 Standards for Distributed Energy ...*, www.nrel.gov/docs/fy15osti/63157.pdf. Accessed 25 Sept. 2024.
- [20] Diehl, A. (2020, March 18). *Determining Module Inter-Row Spacing*. Greentech Renewables. <https://www.greentechrenewables.com/article/determining-module-inter-row-spacing>
- [21] *Create sun path chart*: University of Oregon Solar Radiation Monitoring Laboratory. (n.d.). <https://solardata.uoregon.edu/SunPathChart.html>

Appendix A: Engineering Analysis

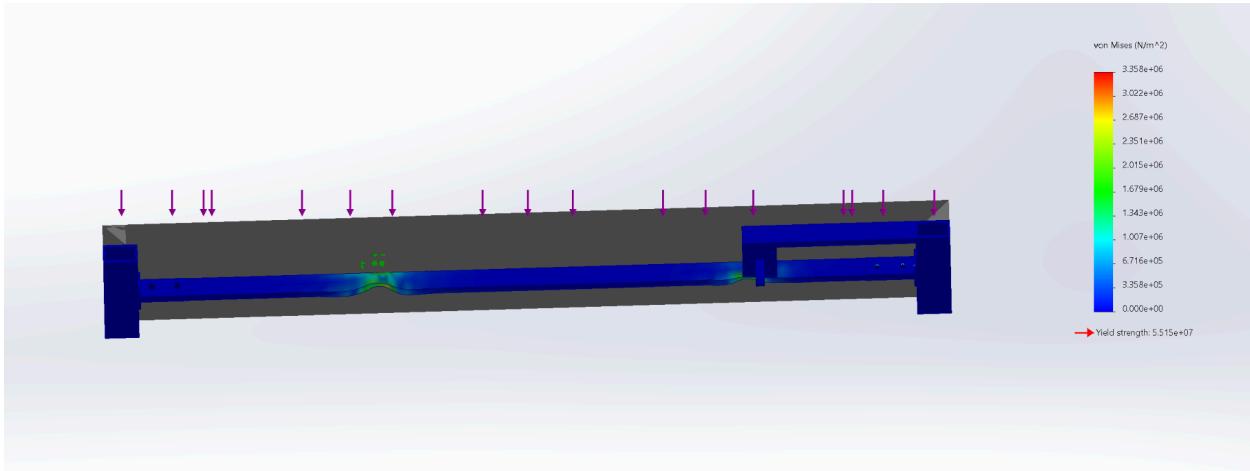


Figure 18: Finite Element Analysis of the Von Mises stress for the solar panel frame with AL 6061 and two fix points where the East-West hinges would attach.

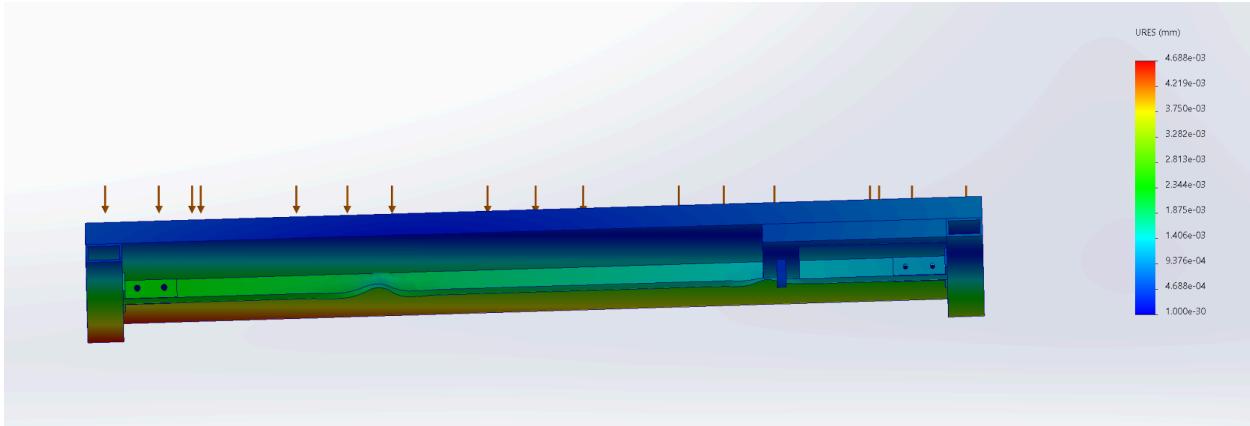


Figure 19: Finite Element Analysis of the URES for the solar panel frame with AL 6061 and two fix points where the East-West hinges would attach.

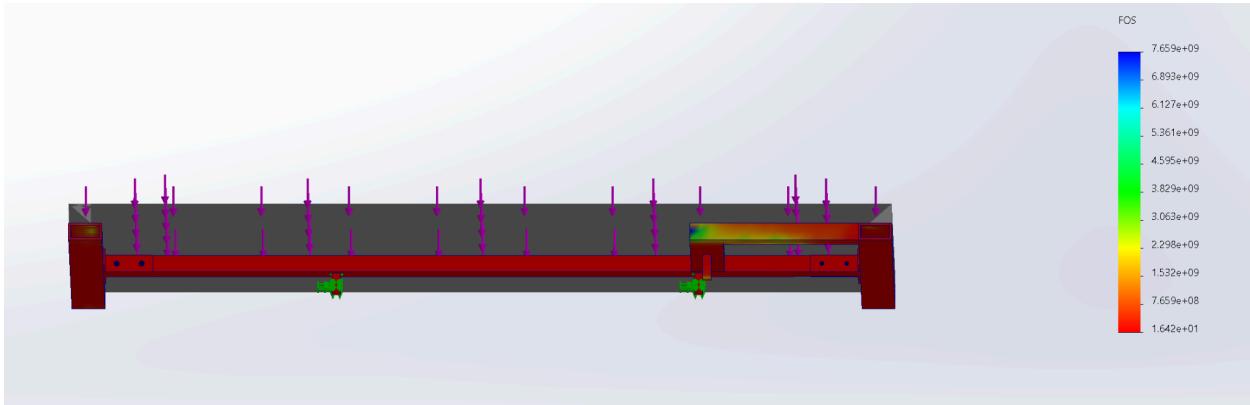


Figure 20: Finite Element Analysis of the Factor of Safety (FOS) for the solar panel frame with AL 6061 and two fix points where the East-West hinges would attach.

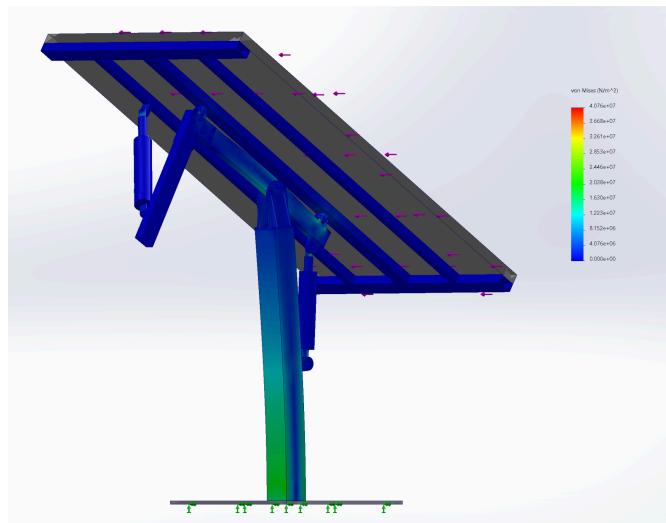


Figure 21: Finite Element Analysis of the Von Mises stress for the entire design with fixed points on the bottom of the base.

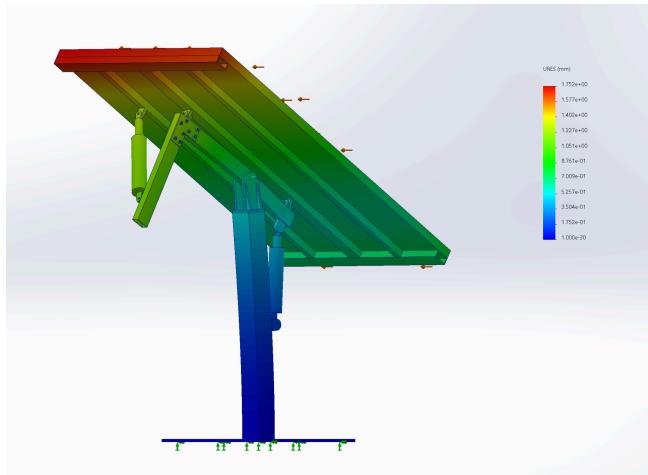


Figure 22: Finite Element Analysis of the URES for the entire design with fixed points on the bottom of the base.

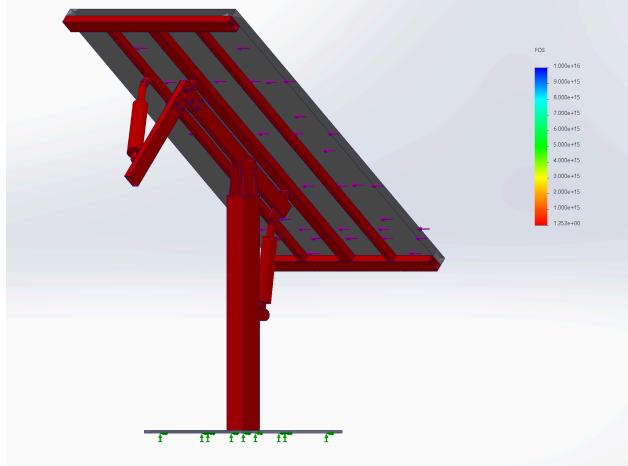


Figure 23: Finite Element Analysis of the Factor of Safety (FOS) for the entire design with fixed points on the bottom of the base.

The Finite Element Analyses use a force of 109.51 pounds, which is the estimated maximum wind force, as calculated using section 6.5.15 of ASCE 7 and shown in Figure 21.

Copstone FEA

$$q_z = 0.00256 \cdot k_z \cdot k_{zg} \cdot k_a \cdot V^2 \cdot I$$

k_z : Velocity pressure exposure coefficient = 0.7 [average for coast]

 k_{zg} : topographic factor = 1

 V : regional basic wind speed = 85 mph = 38 m/s [max 95 mph]

 I : importance factor = 0.87

 q_z : Velocity pressure evaluated @ centroid A_f

 k_a : Wind directability factor = 0.9

$$F = q_z \cdot G \cdot C_F \cdot A_f (1)$$

G : force coefficient = 1.3

 A_f : Surface area of object

 G : Gust factor = 0.85 [typical]

Given:

- Material: 1020 Steel
- assumed to be non-important building [$I = 0.87$]
- Wind load for western coast California calculated @ 1SF4

$$A_f = 1470.46 \text{ in}^2 = 9.774 \text{ ft}^2$$

$$q_z = 0.00256 [0.7] [1] [0.9] [85 \text{ mph}]^2 [0.87] = 10.14 \frac{\text{lb}}{\text{ft}^2}$$

$$q_{zg} = 0.00256 [0.7] [1] [0.9] [95 \text{ mph}]^2 [0.87] = 12.66 \frac{\text{lb}}{\text{ft}^2}$$

$$F_{min} = [10.14 \frac{\text{lb}}{\text{ft}^2}] [0.85] [1.3] [9.774 \text{ ft}^2] = 109.51 \text{ lbs}$$

$$F_{max} = [12.66 \frac{\text{lb}}{\text{ft}^2}] [0.85] [1.3] [9.774 \text{ ft}^2] = 136.73$$

Figure 24: The estimated maximum wind force calculation using section 6.5.15 of ASCE 7.

y : total length
 x : actuator extension
 $x = y - 16.0625$
 α : angle of sun
 θ : angle of actuator
 z : panel tilt angle
 $\text{alt} = 3\frac{3}{16} \text{ in}$
 $\text{alt} = 17.5 \text{ in}$

$$\tan^{-1}\left(\frac{21.5}{3.75}\right) = 80.1^\circ$$

$$y = \sqrt{16.0625^2 + (21\frac{5}{8})^2 - 2 \cdot 16.0625 \cdot 21\frac{5}{8} \cdot \cos(\theta)}$$

$$90 - \alpha = z$$

$$\theta = \alpha + 80.1 \rightarrow \theta = 90 - \alpha + 80.1 = 170.1 - \alpha = \Theta$$

$$90 - \Theta = \text{panel tilt angle}$$

$$y = \sqrt{483.64 - 173 \cdot \cos(170.1 - \alpha)}$$

$$x = \sqrt{483.64 - 173 \cdot \cos(170.1 - \alpha)} - 16.0625$$

$$y_2 = \sqrt{(3\frac{3}{16})^2 + (17.5)^2 - 2 \cdot (3\frac{3}{16}) \cdot (17.5) \cdot \cos(\theta)}$$

$$90 - \alpha_2 = z_2$$

$$\theta_2 = z_2 + 83.84^\circ \rightarrow \theta = 90 - \alpha_2 + 83.84 = 173.84 - \alpha_2$$

$$\text{angle} = \tan^{-1}\left(\frac{17\frac{5}{8}}{3\frac{3}{16}}\right) = 83.84^\circ$$

$$x_2 = \sqrt{318.0025} - 12.031 \cos(173.84 - \alpha_2) - 12.25$$

Figure 25: Calculations equating the actuator length to the angle of tilt.

The MATLAB script that calculates spacing between solar panels, both between rows and columns:

```
% declare measurements and specifications
strokeLength = 8; % inches
retractionSize = 15.87; % inches
extendedSize = retractionSize + strokeLength; % inches
rows = 5;
columns = 13;
panelLength = 49.69; % inches
panelWidth = 30.08; % inches
panelThickness = 1.18; % inches
solarElevationAngle = deg2rad(30); % radians, found from external software graph
azimuthCorrectionAngle = deg2rad(20); % degrees, found from external software graph
correctionFactor = 2;
actuatorDistance = 1;
maxAngleAttackLong = deg2rad(40); % radians, based on CAD
maxAngleAttackShort = deg2rad(40); % radians, based on CAD
% find height difference
heightLong = sin(maxAngleAttackLong)*panelLength; % inches
heightShort = sin(maxAngleAttackShort)*panelWidth; % inches
% find distance between rows
distanceLong = ((heightLong / tan(solarElevationAngle))*cos(azimuthCorrectionAngle)) /
correctionFactor % inches
distanceShort = (heightShort / tan(solarElevationAngle))*cos(azimuthCorrectionAngle) %
inches
```

The Python script that calculates the optimal N-S and E-W angle of tilts for the solar panel:

```
import math
import datetime
import requests
from geopy.geocoders import Nominatim
from timezonefinder import TimezoneFinder
import pytz

# Calculate solar declination angle (in radians)
def calculate_declination(day_of_year):
    return -23.45 * math.cos(math.radians(360/365 * (day_of_year+10)))

def get_utc_offset(lat, lon):
    tf = TimezoneFinder()
    timezone_name = tf.timezone_at(lat=lat, lng=lon)
    if timezone_name is None:
        return "Time zone not found"
```

```

timezone = pytz.timezone(timezone_name)
now = datetime.datetime.now(timezone)
utc_offset = now.utcoffset().total_seconds() / 3600
return utc_offset

# Calculate solar hour angle (in radians)
def calculate_hour_angle(hour, minute, day_of_year, lat, lon):
    LSTM = 15 * get_utc_offset(lat, lon)
    B = 360/365 * (day_of_year - 81)
    EoT = 9.87*math.sin(math.radians(2*B)) - 7.53*math.cos(math.radians(B)) -
    1.5*math.sin(math.radians(B))
    TC = 4*(lon - LSTM) + EoT
    LST = hour + (minute/60) + (TC/60)
    HRA = 15*(LST - 12)
    return HRA

# Calculate solar altitude angle (in radians)
def calculate_altitude_angle(lat, dec, HA):
    latitude = math.radians(lat)
    declination = math.radians(dec)
    hour_angle = math.radians(HA)
    return math.degrees(math.asin(math.sin(latitude) * math.sin(declination) +
        math.cos(latitude) * math.cos(declination) * math.cos(hour_angle)))

# Calculate solar azimuth angle (in radians)
def calculate_azimuth_angle(lat, dec, alt, ha):
    latitude = math.radians(lat)
    declination = math.radians(dec)
    altitude_angle = math.radians(alt)
    HRA = math.radians(ha)
    numerator = math.sin(declination) * math.cos(latitude) -
    math.cos(declination)*math.sin(latitude)*math.cos(HRA)
    denominator = math.cos(altitude_angle)
    azimuth = math.degrees(math.acos(numerator / denominator))
    # Adjust azimuth angle for hemisphere
    if lat > 0: # Northern Hemisphere
        if ha > 0: # Afternoon (sun is in the western sky)
            azimuth = 360 - azimuth # Reflect angle from north
        else: # Southern Hemisphere
            if ha < 0: # Afternoon (sun is in the western sky)
                azimuth = 360 - azimuth # Reflect angle from north to south
    return azimuth

# Get the day of the year from the current date
def get_day_of_year():

```

```

now = datetime.datetime.now()
return now.timetuple().tm_yday

# Get the Raspberry Pi's current date and time
def get_current_datetime():
    return datetime.datetime.now()

# Get the Raspberry Pi's latitude and longitude using the geopy library
def get_latitude_longitude():
    try:
        # Use an IP geolocation API to determine location
        response = requests.get("http://ip-api.com/json")
        data = response.json()
        if data["status"] == "success":
            return data["lat"], data["lon"]
        else:
            raise Exception("Could not determine location from IP.")
    except Exception as e:
        print(f'Error determining location: {e}')
    return None, None

# Main function
def main():
    # Get current date and time
    now = get_current_datetime()
    hour = now.hour
    minute = now.minute

    # Get the Raspberry Pi's latitude and longitude
    latitude, longitude = get_latitude_longitude()
    if latitude is None or longitude is None:
        print("Unable to determine location. Please provide latitude and longitude manually.")
        return

    # Calculate day of the year
    day_of_year = get_day_of_year()

    # Calculate solar angles
    declination = calculate_declination(day_of_year)
    hour_angle = calculate_hour_angle(hour, minute, day_of_year, latitude, longitude)
    altitude_angle = calculate_altitude_angle(latitude, declination, hour_angle)
    azimuth_angle = calculate_azimuth_angle(latitude, declination, altitude_angle, hour_angle)

    # Calculate optimal E-W and N-S tilts
    corrected_azimuth_angle = 180 - azimuth_angle
    ns_tilt = altitude_angle if -30 <= altitude_angle <= 30 else max(-30, min(30, altitude_angle))

```

```

# Clamp N-S tilt to a minimum of 0
ew_tilt = corrected_azimuth_angle if -40 <= corrected_azimuth_angle <= 40 else max(-40,
min(40, corrected_azimuth_angle))

# Output results
print(f"Optimal North-South tilt (from horizontal): {ns_tilt:.2f} degrees")
print(f"Optimal East-West tilt (from center): {ew_tilt:.2f} degrees")

if __name__ == "__main__":
    main()

```

The Python script that facilitates the raspberry pi and arduino communication:

```

import smbus
import struct
import time
import os

# I2C bus
bus = smbus.SMBus(1) # 1 for /dev/i2c-1

# Arduino I2C address
arduino_address = 0x08
# File to store actuator extensions
EXTENSION_FILE = "actuator_extensions.txt"

def read_extensions():
    """Reads the current actuator extensions from the file."""
    extensions = {1: 0.0, 2: 0.0} # Default values
    if os.path.exists(EXTENSION_FILE):
        try:
            with open(EXTENSION_FILE, "r") as f:
                lines = f.readlines()
                if len(lines) >= 2:
                    try:
                        extensions[1] = float(lines[0].strip())
                        extensions[2] = float(lines[1].strip())
                    except ValueError:
                        print("Error: Invalid data in extension file. Resetting to 0.")
                        extensions = {1: 0.0, 2: 0.0}
        except Exception as e:
            print(f"Error reading extension file: {e}")
    return extensions

def write_extensions(extensions):
    """Writes the current actuator extensions to the file."""

```

```

try:
    with open(EXTENSION_FILE, "w") as f:
        f.write(str(extensions[1]) + "\n")
        f.write(str(extensions[2]) + "\n")
except Exception as e:
    print(f"Error writing to extension file: {e}")

def send_actuator_data(actuator_num, mm_value):
    """
    Sends actuator number and a millimeter value to the Arduino via I2C.

    Args:
        actuator_num: The actuator number (1 or 2).
        mm_value: The millimeter value (float, between -20 and 20).
    """
    try:
        # Pack the data (Big Endian)
        data = struct.pack('>if', actuator_num, mm_value) # '>if' for int, float
        bus.write_i2c_block_data(arduino_address, 0, list(data))
        print(f"Sent: Actuator = {actuator_num}, MM = {mm_value}")
    except Exception as e:
        print(f"Error sending data: {e}")

if __name__ == "__main__":
    try:
        # Read the initial extensions from the file
        current_extensions = read_extensions()
        print(f"Initial extensions: Actuator 1 = {current_extensions[1]} mm, Actuator 2 = {current_extensions[2]} mm")

        # Example usage:
        move1_mm = 15.5
        send_actuator_data(1, move1_mm)
        time.sleep(1)
        current_extensions[1] += move1_mm
        write_extensions(current_extensions)
        print(f"New extensions: Actuator 1 = {current_extensions[1]} mm, Actuator 2 = {current_extensions[2]} mm")

        move2_mm = -10.2
        send_actuator_data(2, move2_mm)
        time.sleep(1)
        current_extensions[2] += move2_mm
        write_extensions(current_extensions)
    
```

```

    print(f'New extensions: Actuator 1 = {current_extensions[1]} mm, Actuator 2 =
{current_extensions[2]} mm')

    move3_mm = 0.0
    send_actuator_data(1, move3_mm)
    time.sleep(1)
    current_extensions[1] += move3_mm
    write_extensions(current_extensions)
    print(f'New extensions: Actuator 1 = {current_extensions[1]} mm, Actuator 2 =
{current_extensions[2]} mm')

    final_extensions = read_extensions()
    print(f'Final extensions from file: Actuator 1 = {final_extensions[1]} mm, Actuator 2 =
{final_extensions[2]} mm')

except KeyboardInterrupt:
    print("Script stopped by user")
finally:
    bus.close()

```

The Python script that records the sensor data:

```

import time
import spidev
import smbus
from datetime import datetime
import os # Import the os module for file operations

# I2C bus for INA219
bus = smbus.SMBus(1)

# TCA9548A address
TCA9548A_ADDRESS = 0x70

# INA219 address
INA219_ADDRESS = 0x40

# INA219 Registers
INA219_CONFIG_REGISTER = 0x00
INA219_BUS_VOLTAGE_REGISTER = 0x02

# SPI for MCP3008
spi = spidev.SpiDev()
spi.open(0, 0)
spi.max_speed_hz = 1000000
spi.mode = 0b01

```

```

# MCP3008 Channels
LOAD_CURRENT_CHANNEL = 0
SOLAR_CURRENT_CHANNEL = 2

# TCA9548A Channels for Voltage Sensors
LOAD_VOLTAGE_CHANNEL = 6
SOLAR_VOLTAGE_CHANNEL = 4

# ACS712 Sensitivity (adjust as needed for your sensors)
ACS712_LOAD_SENSITIVITY = 0.066 # For 30A ACS712 connected to load
ACS712_SOLAR_SENSITIVITY = 0.066 # For 30A ACS712 connected to solar
ACS712_VCC = 3.3 # Supply voltage
DATA_DIR = "sensor_data" # Directory to store data

def select_i2c_channel(channel):
    """Selects the I2C channel on the TCA9548A."""
    try:
        bus.write_byte_data(TCA9548A_ADDRESS, 0, 1 << channel)
        time.sleep(0.001)
        return True
    except Exception as e:
        print(f"Error selecting channel {channel}: {e}")
        return False

def read_ina219_register(channel, register):
    """Reads a 16-bit register from the INA219 on the specified TCA9548A channel."""
    if not select_i2c_channel(channel):
        print(f"Failed to select TCA9548A channel {channel}. Aborting INA219 read.")
        return None
    try:
        data = bus.read_i2c_block_data(INA219_ADDRESS, register, 2)
        value = (data[0] << 8) | data[1]
        return value
    except Exception as e:
        print(f"Error reading INA219 register 0x{register:02X} on channel {channel}: {e}")
        return None

def configure_ina219(channel):
    """Configures the INA219 for voltage reading on the specified TCA9548A channel."""
    if not select_i2c_channel(channel):
        print(f"Failed to select TCA9548A channel {channel}. Aborting INA219 config.")
        return False
    try:
        config_value = 0x1800 # Simplified for voltage-only
        bus.write_i2c_block_data(INA219_ADDRESS, INA219_CONFIG_REGISTER,

```

```

        [(config_value >> 8) & 0xFF, config_value & 0xFF])
time.sleep(0.001)
return True
except Exception as e:
    print(f"Error configuring INA219 on channel {channel}: {e}")
    return False

def get_bus_voltage(channel):
    """Reads the bus voltage from the INA219 on the specified TCA9548A channel in volts."""
    voltage_value = read_ina219_register(channel, INA219_BUS_VOLTAGE_REGISTER)
    if voltage_value is not None:
        voltage_mV = (voltage_value >> 3) * 4
        voltage_V = float(voltage_mV) / 1000.0
        return voltage_V
    else:
        return None

def read_mcp3008(channel):
    """Reads the analog value from the MCP3008 ADC on the specified channel."""
    if channel < 0 or channel > 7:
        raise ValueError("MCP3008 channel must be between 0 and 7")
    command = 0b00000001
    command |= (1 << 1)
    command |= (channel << 2)
    command_bytes = [command, 0b00000000, 0b00000000]
    response = spi.xfer2(command_bytes)
    adc_value = ((response[1] & 0x03) << 8) | response[2]
    return adc_value

def get_current(adc_value, sensitivity):
    """Converts the ADC value to current in Amperes using the specified sensitivity."""
    voltage = (adc_value * ACS712_VCC) / 1023.0
    zero_current_voltage = ACS712_VCC / 2.0
    current = (voltage - zero_current_voltage) / sensitivity
    return current

def get_filename():
    """Generates the filename based on the current date."""
    today = datetime.now().strftime("%Y-%m-%d")
    return os.path.join(DATA_DIR, f"sensor_data_{today}.txt")

if __name__ == "__main__":
    try:
        if not os.path.exists(DATA_DIR): #create directory if it does not exist
            os.makedirs(DATA_DIR)

```

```

if not configure_ina219(LOAD_VOLTAGE_CHANNEL) or not
configure_ina219(SOLAR_VOLTAGE_CHANNEL):
    print("INA219 configuration failed on one or more channels.")
    exit(1)

while True:
    # Get timestamp
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    # Read load voltage
    load_voltage_V = get_bus_voltage(LOAD_VOLTAGE_CHANNEL)
    load_voltage_str = f'{timestamp}, Load Voltage: {load_voltage_V:.3f} V' if
load_voltage_V is not None else f'{timestamp}, Failed to read Load Voltage'
    print(load_voltage_str)

    # Read solar voltage
    solar_voltage_V = get_bus_voltage(SOLAR_VOLTAGE_CHANNEL)
    solar_voltage_str = f'{timestamp}, Solar Voltage: {solar_voltage_V:.3f} V' if
solar_voltage_V is not None else f'{timestamp}, Failed to read Solar Voltage'
    print(solar_voltage_str)

    # Read load current
    load_current_adc = read_mcp3008(LOAD_CURRENT_CHANNEL)
    load_current_A = get_current(load_current_adc, ACS712_LOAD_SENSITIVITY)
    load_current_str = f'{timestamp}, Load Current: {load_current_A:.3f} A'
    print(load_current_str)

    # Read solar current
    solar_current_adc = read_mcp3008(SOLAR_CURRENT_CHANNEL)
    solar_current_A = get_current(solar_current_adc, ACS712_SOLAR_SENSITIVITY)
    solar_current_str = f'{timestamp}, Solar Current: {solar_current_A:.3f} A'
    print(solar_current_str)

    # Write to file
    filename = get_filename()
    try:
        with open(filename, "a") as f:
            f.write(load_voltage_str + "\n")
            f.write(solar_voltage_str + "\n")
            f.write(load_current_str + "\n")
            f.write(solar_current_str + "\n")
    except Exception as e:
        print(f"Error writing to file {filename}: {e}")

    time.sleep(1)

```

```

except KeyboardInterrupt:
    print("Script stopped by user")
finally:
    spi.close()
    bus.close()

```

The Arduino script that controls the actuators and receives commands from the raspberry pi:

```

#include <Wire.h>

// Pin definitions for actuators
#define R_EN_ACT1 7
#define L_EN_ACT1 6
#define RPWM_ACT1 8
#define LPWM_ACT1 9
#define ACTUATOR_ENABLE_ACT1 10
#define R_EN_ACT2 2
#define L_EN_ACT2 3
#define RPWM_ACT2 4
#define LPWM_ACT2 5
#define ACTUATOR_ENABLE_ACT2 11

// Constants
const float SPEED_INCHES_PER_SEC = 0.51;
const float MM_PER_INCH = 25.4;
const int TCA9548A_ADDRESS = 0x70; // I2C address of TCA9548A
const int TCA9548A_CHANNEL = 7; // Channel 7 (SCL/SDA7)
const int ARDUINO_I2C_ADDRESS = 0x09; // Arduino's I2C Address

// Global variables for actuator state
int currentActuator = 0; // 0: none, 1: actuator 1, 2: actuator 2
int currentDirection = 0; // 0: stopped, 1: forward, -1: backward
unsigned long moveStartTime = 0; // Time when movement started
unsigned long moveDuration = 0; // Duration of current movement
bool isMoving = false;
bool receivedCommand = false; // Flag to indicate a command was received

void setup() {
    Wire.begin(9); // Join I2C bus as slave with address 9
    Serial.begin(9600); // For debugging
    Wire.onReceive(receiveEvent);

    // pinMode declarations
    pinMode(R_EN_ACT1, OUTPUT);
    pinMode(L_EN_ACT1, OUTPUT);
    pinMode(RPWM_ACT1, OUTPUT);
}

```

```

pinMode(LPWM_ACT1, OUTPUT);
pinMode(ACTUATOR_ENABLE_ACT1, OUTPUT);
pinMode(ACTUATOR_ENABLE_ACT2, OUTPUT);

pinMode(R_EN_ACT2, OUTPUT);
pinMode(L_EN_ACT2, OUTPUT);
pinMode(RPWM_ACT2, OUTPUT);
pinMode(LPWM_ACT2, OUTPUT);

digitalWrite(R_EN_ACT1, HIGH);
digitalWrite(L_EN_ACT1, HIGH);
digitalWrite(ACTUATOR_ENABLE_ACT2, HIGH); // Enable Actuator 2
digitalWrite(R_EN_ACT2, HIGH);
digitalWrite(L_EN_ACT2, HIGH);
digitalWrite(ACTUATOR_ENABLE_ACT1, HIGH); // Enable Actuator 1
Serial.println("Arduino ready.");
}

void loop() {
    //this is needed for the arduino to receive commands
    manageActuators();      // Handle actuator movement
}

void moveActuator(int actuatorNum, int direction, unsigned long duration) {
    // Only start movement if not already moving
    if (!isMoving) {
        isMoving = true;
        currentActuator = actuatorNum;
        currentDirection = direction;
        moveStartTime = millis();
        moveDuration = duration;

        Serial.print("moveActuator: Actuator = ");
        Serial.print(actuatorNum);
        Serial.print(", Direction = ");
        Serial.print(direction);
        Serial.print(", Duration = ");
        Serial.println(duration);

        if (actuatorNum == 1) {
            if (direction > 0) {
                analogWrite(RPWM_ACT1, 255);
                analogWrite(LPWM_ACT1, 0);
            } else {
                analogWrite(RPWM_ACT1, 0);
                analogWrite(LPWM_ACT1, 255);
            }
        }
    }
}

```

```

        }
    } else if (actuatorNum == 2) {
        if (direction > 0) {
            analogWrite(RPWM_ACT2, 255);
            analogWrite(LPWM_ACT2, 0);
        } else {
            analogWrite(RPWM_ACT2, 0);
            analogWrite(LPWM_ACT2, 255);
        }
    }
} else {
    Serial.println("moveActuator: Already moving");
}
}

void manageActuators() {
if (isMoving) {
    if (millis() - moveStartTime >= moveDuration) {
        // Stop the actuator
        if (currentActuator == 1) {
            analogWrite(RPWM_ACT1, 0);
            analogWrite(LPWM_ACT1, 0);
            digitalWrite(R_EN_ACT1, HIGH); // Stop Actuator 1
            digitalWrite(L_EN_ACT1, HIGH);
        } else if (currentActuator == 2) {
            analogWrite(RPWM_ACT2, 0);
            analogWrite(LPWM_ACT2, 0);
            digitalWrite(R_EN_ACT2, HIGH); // Stop Actuator 2
            digitalWrite(L_EN_ACT2, HIGH);
        }
        isMoving = false;
        currentActuator = 0;
        currentDirection = 0;
        Serial.println("Target reached");
        // No need to send "OK" back to the Pi in this version.
    }
}
}

void receiveEvent(int howMany) {
    Serial.print("receiveEvent called. Received ");
    Serial.print(howMany);
    Serial.println(" bytes.");
}

```

```

if (howMany == 9) { // Expecting 9 bytes
    byte data[9];
    // Read the 9 bytes
    for (int i = 0; i < 9; i++) {
        data[i] = Wire.read();
        Serial.print("data[");
        Serial.print(i);
        Serial.print("] = ");
        Serial.print(data[i], HEX);
        Serial.print(" ");
    }
    Serial.println();
}

// Extract data from the received bytes (Big Endian)
int actuatorNum = data[4];
union {
    float f;
    byte b[4];
} u;
u.b[3] = data[5];
u.b[2] = data[6];
u.b[1] = data[7];
u.b[0] = data[8];
float targetDistanceMM = u.f;

Serial.print("Received: Actuator = ");
Serial.print(actuatorNum);
Serial.print(", MM = ");
Serial.println(targetDistanceMM, 4);

// Calculate the duration
unsigned long targetTimeMS = abs(long(targetDistanceMM / MM_PER_INCH /
SPEED_INCHES_PER_SEC * 1000));

// Move the actuator
moveActuator(actuatorNum, targetDistanceMM > 0 ? 1 : -1, targetTimeMS);
receivedCommand = true; // Set the flag
} else {
    Serial.print("Received ");
    Serial.print(howMany);
    Serial.println(" bytes. Expected 9.");
    while (Wire.available()) {
        Wire.read();
    }
}
}

```

Appendix B: Manufacturing Plan

This procedure details the manufacturing process for the most intricate and machining-intensive component in the entire design: the N-S Assembly. This assembly demands precision through CNC machining for radius cutouts, support shaping, and precise drilling of hinge pin holes, coupled with meticulous welding to ensure structural integrity. By comparison, all other parts of the design require only straightforward cutting of standard metal shapes and basic welding, making the N-S Assembly the centerpiece of the fabrication process. Refer to Figure 32 in the appendix for the individual parts of the N-S Hinge, which serve as guides for machining and material cutout instructions. Figure 33 illustrates the complete N-S Assembly along with accurate dimensions for welding.

All other design components consist of square tubing, which will be cut to length using a CNC machine. After cutting, each piece will be laid out and dimensioned with the aid of a 90° magnetic clamp, then welded together section by section until the assembly is complete.

Manufacturing Procedure for N-S Assembly

1. Material Preparation

- a. Procure a 10" x 10" sheet of 1020 steel, $\frac{1}{4}$ " thickness, as specified.

2. Initial Rough Cutout

- a. Utilize a laser cutter or CNC machine to create rough rectangular cutouts of all seven components required for the assembly. Ensure dimensions are within tolerances suitable for subsequent machining.
- b. The base can be cut to its final dimensions as it is a simple square.

3. Precision Shaping

- a. Employ a CNC machine to create the radius cutout (1.5" radius) on the specified components.
- b. Machine the 0.625" x 3.5" rectangles into triangular support shapes as per design specifications.

4. Preliminary Frame Assembly

- a. Temporarily align the vertical frames using clamps or adhesive tape to hold them in position for further processing. Ensure alignment accuracy.

5. Drilling Operations

- a. Drill pilot holes in both vertical frames at the specified hinge pin locations. Use the pilot holes as reference points to accurately machine the final hinge pin holes. Verify hole dimensions and positions against the design requirements.

6. Deburring

- a. Perform deburring on all edges and holes to remove burrs and sharp edges, ensuring safe handling and proper fit during assembly.

7. Welding Operations

- a. Weld the triangular support components to the vertical frames following specified weld sizes and positions.
- b. Assemble the vertical frames (with attached supports) to the square base and weld them in place. Maintain specified spacing and alignment during the welding process.

8. Final Assembly

- a. Secure the steel base to a wooden platform and add elbow supports to stabilize the main structure.
- b. Attach the N-S member to the base using a clevis pin.
- c. Mount the solar panel onto the N-S member, aligning the E-W hinges and securing them with clevis pins.
- d. Connect the actuators to their designated brackets using clevis pins to support and operate the installed solar panel.

9. Surface Treatment

- a. Apply a weather-resistant coating to the assembled structure to enhance corrosion resistance and ensure longevity in service.

For the electronic assembly, Figures 22-24 depict the full system. The electronics will be assembled exactly as depicted. As for the software, once the electronics are all assembled, the system will be coded with priority going to the actuator control and sensor response. Once those parts are working, a user interface (UI) for the LCD will be created that will output weather and tilt angle data and include manual override of tilt angles, emergency stops, and turning actuators and Arduinos on and off. Additionally, if time permits, the sensor data will also be outputted to the LCD to monitor the power consumption of the system in real time.

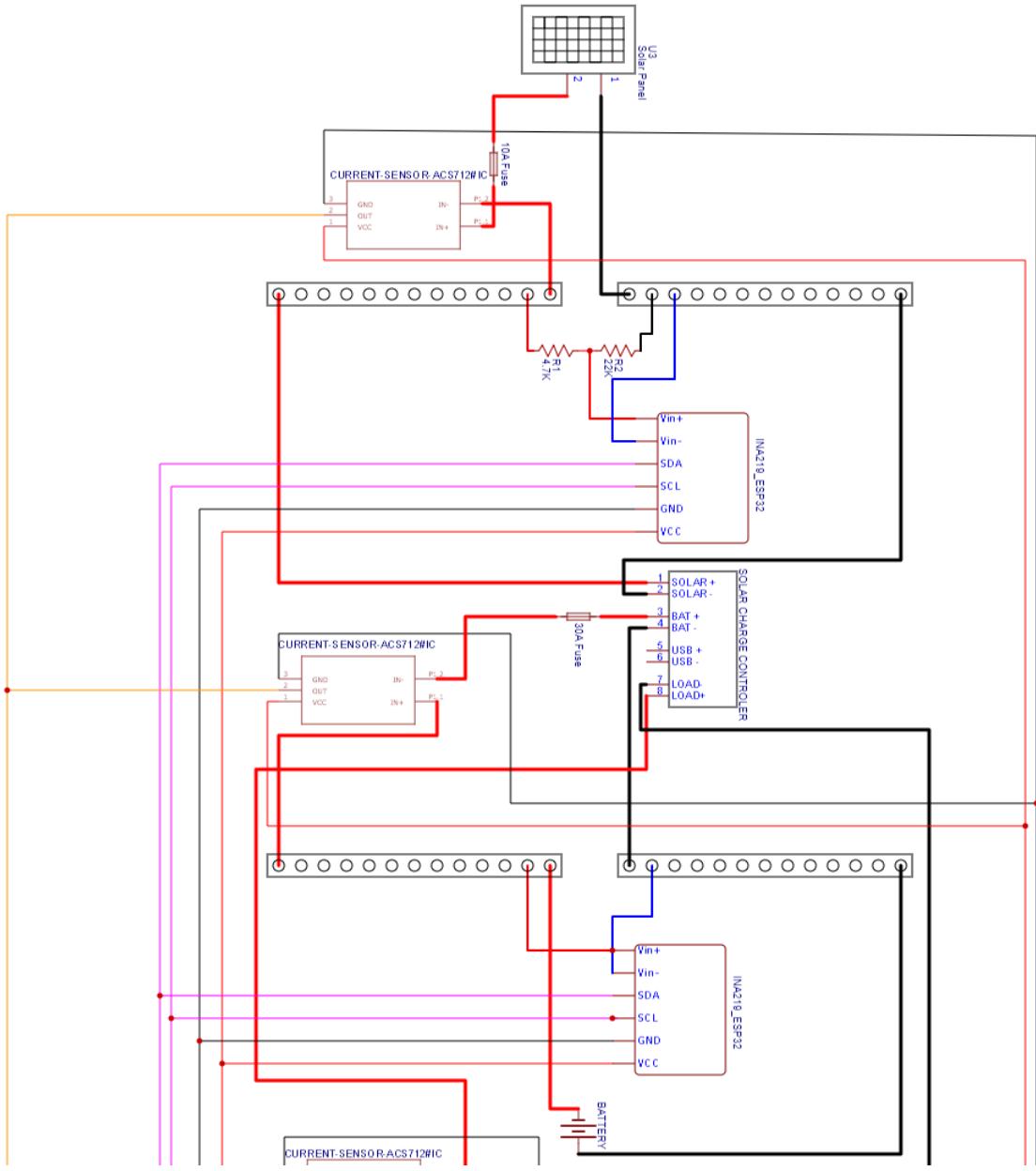


Figure 26: First Third of the Wiring Diagram.

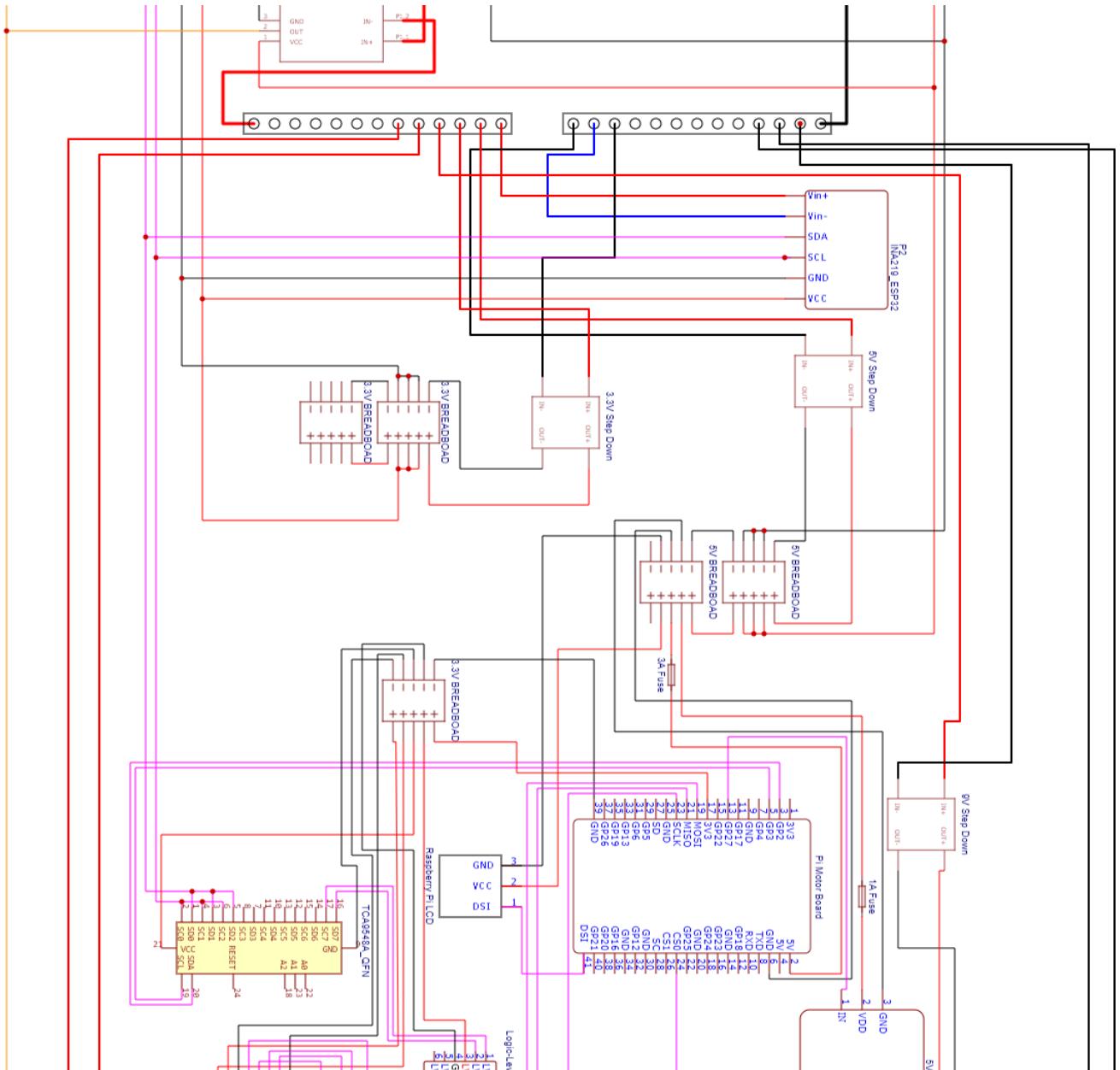


Figure 27: Second Third of the Wiring Diagram.

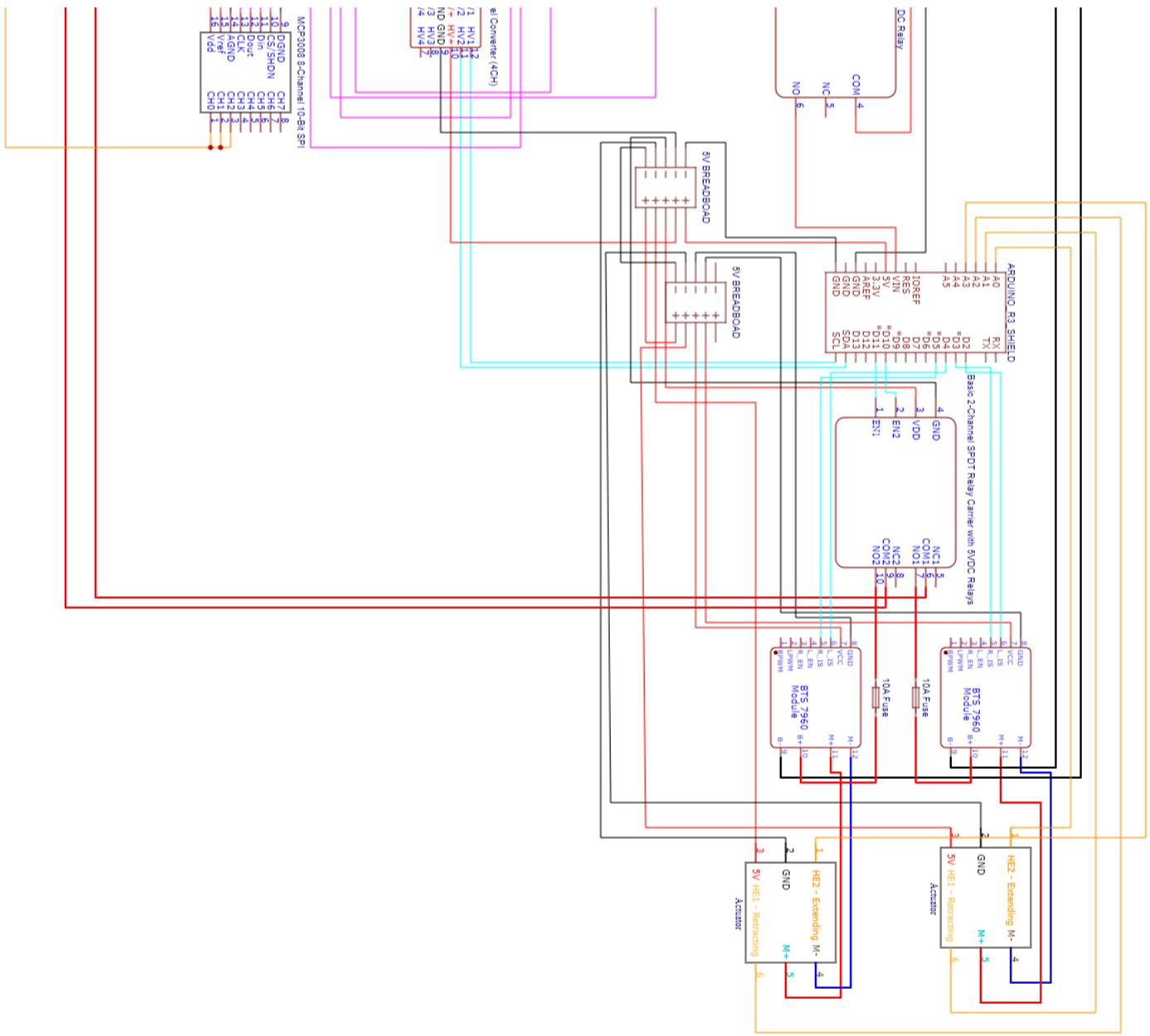


Figure 28: Final Third of the Wiring Diagram.

Appendix C: Project Schedule

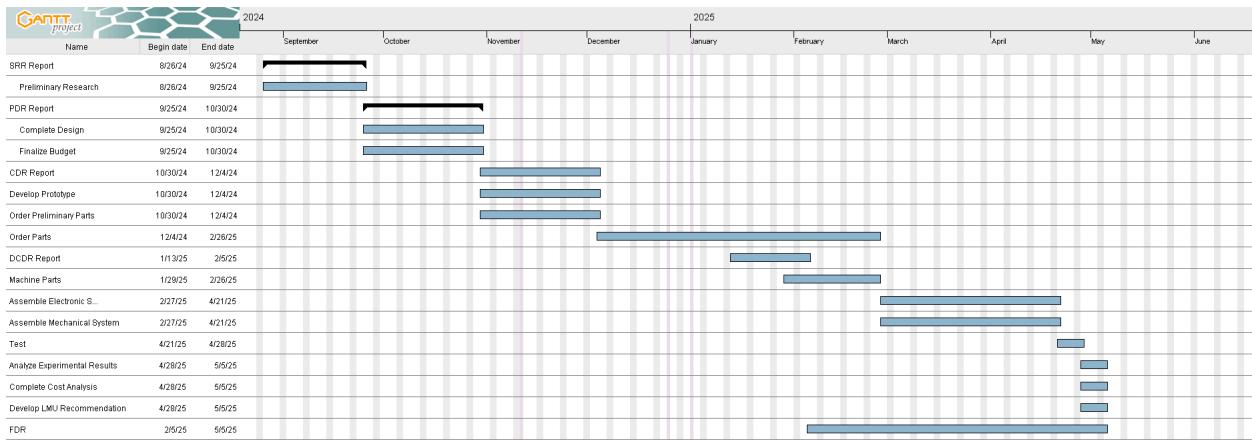


Figure 29: Gantt schedule from September, 2024 to May, 2025, including major milestones and deliverables.

Appendix D: Purchased Parts List

Table 4: Budgeted parts list for the design, including both mechanical and electrical parts. All parts have been purchased.

Part	Part #	Unit Price	Quantity	Total Price	Store/Link	Comments
Solar Panel	RSP200DC-A SR-US	\$239.99	1	\$239.99	Renogy	200 W Solar Panel
Arduino Uno REV3	ARD_A00006_6	\$29.00	1	\$29.00	Amazon	Arduino model
Raspberry Pi 4 Model B/4GB	4GB-9004	\$55.00	1	\$55.00	Raspberry Pi US	Raspberry Pi model
microSD Card with Raspberry Pi OS	1363-1	\$12.95	1	\$12.95	Raspberry Pi US	Raspberry Pi OS
USB-C Power Supply, 5.1V 3.0A	1203	\$7.95	1	\$7.95	Raspberry Pi US	Raspberry Pi power supply
7inch HDMI LCD	2131	\$77.95	1	\$77.95	Raspberry Pi US	LCD Screen
Logic Level Convert Bi-Directional	BOB-12009	\$2.95	5	\$14.75	electroMaker	Converts 5V signals to 3.3V signals and vice versa to allow raspberry pi to communicate with the arduino
5Pack LM2596 LM2596S DC-DC Step Down Variable Volt Regulator Input 3.0-40V Output 1.5-35V Adjustable Buck Converter Electronic Voltage Stabilizer Power Supply Module	B07PDGG84B	\$12.99	1	\$12.99	Amazon	Drops the voltage from the battery to 5V for Arduino+Pi
ECO-WORTHY 30A Solar Charger Controller Solar Panel Battery Intelligent Regulator with Dual USB Port Auto 12/24V PWM Positive Ground	B08JBLMC33	\$29.99	1	\$29.99	Amazon	Connect panel to battery/system

ECO-WORTHY 12V 30AH LiFePO4 Battery, Rechargeable Lithium Ion Phosphate Deep Cycle Battery for Trolling Motor, Golf Cart, Kids Scooters, Power Wheelchairs, Replacement of 12V 35AH SLA Battery	L13060202003 -1A	\$89.99	1	\$89.99	Amazon	System is connected to the battery
4pcs DC 5V Relay Module 1 Channel Relay Switch Relay Board with Optocoupler Isolation Support High or Low Level Trigger	B095YD3732	\$6.99	2	\$13.98	Amazon	Connect arduino to raspberry pi
2Pcs DC 5V 4 Channel Relay Module Board Shield with Optocoupler Isolation Suport High/Low Level Trigger Relay	B07PSGRFP7	\$11.99	1	\$11.99	Amazon	Connect actuators to arduino
10-12 AWG M5 Ring Terminals	B0D3TM9KQ D	\$9.98	1	\$9.98	Amazon	Adaptor for wire to connect to battery
10awg Silicone Electrical Wire 2 Core Wire 20ft [Black 10ft Red 10ft] 10 Gauge Soft and Flexible Hook Up Oxygen Free Strands Tinned Copper Wire	B0B9JBJ3LK	\$16.98	1	\$16.98	Amazon	Connect battery to solar charge controller
Electrical Wire 14 AWG 14 Gauge Silicone Wire Hook Up Wire Cable 20 Feet [10 ft Black and 10 ft Red] Soft and Flexible 400 Strands 0.08 mm of Tinned Copper Wire High Temperature Resistance (14AWG)	754047498411	\$9.88	1	\$9.88	Amazon	Connect actuators to bus bar

ECO-WORTHY 10FT 10AWG Solar Extension Cable with Female and Male Connector Solar Panel Wire Adaptor for Home, Ship and RV Solar Panels (10FT Red + 10FT Black)	B0CDZVMBP 8	\$17.99	1	\$17.99	Amazon	Connect solar panel to solar charge controller
PowMr 10A Waterproof in-Line Solar Fuse Holder w/Fuse, Black	B08C2PXBKN	\$12.99	1	\$12.99	Amazon	Fuse between panel and solar charge controller
30 Amp MRBF Terminal Fuse Block Kit (Fuse Included) – Single Stud 5/16" (M8) Compact High-Amp Marine Rated Battery Fuse – 58V DC Maximum, IP66 Waterproof (30A)	B0BDMWQS XZ	\$15.99	1	\$15.99	Amazon	
VANTRONIK Waterproof Inline Fuse Holder 6 Pack for Standard Car Fuse, 14 Gauge AWG Wire UL1015,10 Inches Long, Fit for ATO ATC Medium Blade Fuse	B081YDV8PS	\$6.79	1	\$6.79	Amazon	
(25Pcs) 10 Amp Standard Car Fuses, 10A Automotive Fuses(ATC/ATO), Blade Fuse for Car/RV/Truck/SUV/Motorcycle/Boat	B0CHZ3R265	\$3.99	1	\$3.99	Amazon	
uxcell Inline Screw Type Fuse Holder 22 Gauge AWG 5Pcs for 6mm x 30mm Tube Fuse	a19040700ux0 434	\$5.89	1	\$5.89	Amazon	
BOJACK 6x30mm 1 A 1 amp 250 V 0.24x1.18 Inch Volt F1AL250V Fast-Blow Glass Fuses(Pack of 18 Pcs)	B07V5HDVK 7	\$5.99	1	\$5.99	Amazon	

BOJACK 6x30mm 3 A 3 amp 250 V 0.24x1.18 Inch Volt F3AL250V Fast-Blow Glass Fuses(Pack of 18 Pcs)	B07SBH28ZW	\$5.99	1	\$5.99	Amazon	
Mofeez Bus Bar -3x1/4"Post,10x#8 Screw Terminal Power Distribution Block with Ring Terminals(Pair - Red & Black)	B07KVV7F5X	\$19.99	3	\$59.97	Amazon	
Female/Male Ext Jumper Wires 40x3	825	\$3.95	1	\$3.95	electroMaker	Connect everything connected to the Pi and Arduino
Female/Female Jumper Wires - 40 x 3"	794	\$3.95	1	\$3.95	electroMaker	Connect everything connected to the Pi and Arduino
Male/Male Jumper Wires - 40 x 3	759	\$3.95	1	\$3.95	electroMaker	Connect everything connected to the Pi and Arduino
Ferrule Crimping Tool Kit, LUNNEY Wire Ferrule Kit with 1250PCS Wire End Ferrules & Crimper Plier, Self-adjustable Ratchet Tool Set for AWG 23-7 Electrical Wire Connectors, 0.25-10mm ²	B0B23V7ZQJ	\$18.98	1	\$18.98	Amazon	
WWZMDiB ACS712 30A Compatible Current Sensor Module	WWZMDiB ACS712	\$8.99	1	\$8.99	Amazon	
6pcs INA219 I2C Bi-Directional DC Current Power Supply Sensor Breakout Module Compatible with Arduino Raspberry Pi (6pcs)	B08V4X5L5B	\$16.59	1	\$16.59	Amazon	
Adafruit MCP3008 8-Channel Analog-to-Digital Converter with SPI Interface	B00NAY3RB2	\$8.49	1	\$8.49	Amazon	

TCA9548A I2C Multiplexer	2717	\$6.95	1	\$6.95	Adafruit	
TERM BLK 2POS SIDE ENTRY 5MM PCB	0395930602-ND	\$1.04	1	\$1.04	Digikey	
RES 10K OHM 5% 1/4W AXIAL	CF14JT10K0	\$0.10	1	\$0.10	Digikey	
RES 3.3K OHM 5% 1/2W AXIAL	CFM12JT3K30	\$0.10	1	\$0.10	Digikey	
25kHz Semiconductor BTS7960B Motor Driver 43A H-Bridge Drive PWM For Arduino US	176693700253	\$6.79	2	\$13.58	Ebay	
PCB007 1X 12 Position Power Distribution Board 2 Inputs 2 x 13 Outputs for DC AC Voltage New	PCB007	\$6.99	4	\$27.96	Amazon	
Feedback Linear Actuator IP66	PA-04-6-400-HS	\$176.00	1	\$176.00	Actuator 1	Actuator for North-South Movement, 6 inch
Feedback Linear Actuator IP66	PA-04-8-400-HS	\$176.00	1	\$176.00	Actuator 2	Actuator for East-West Movement, 8 inch
Actuator U-Bracket	Model: PA-04	\$8.50	4	\$34.00	U-Bracket	Holds the Actuator
Low-Carbon Steel Sheet. 10" x 10" x 1/4"	6544K24	\$65.67	1	\$65.67	McMaster Hinge	Facilitates N-S movement
Multipurpose 6061 Aluminum Rectangular Tube. 1/16" Wall Thickness, 1" High x 2" Wide Outside	6546K2	\$37.58	1	\$37.58	Grid Support	Extra grid support
Low-Carbon Steel Rectangular Tube. 1/8" Wall Thickness, 3" x 3" Outside Size	6527K424	\$57.60	1	\$57.60	McMaster Base	Main Stem for Base
Multipurpose 6061 Aluminum Bar. 1/8" Thick x 4" Wide	8975K418	\$4.60	1	\$4.60	McMaster Bracket	N-S Bracket: Connects N-S members together
Tight-Tolerance Multipurpose 6061 Aluminum. Precision Ground on All Sides, 2" x	9057K413	\$50.52	1	\$50.52	E-W Hinge Block	E-W hinge. Manufacture hinges out of these. Extra material for

3" x 1-1/4" Thick						more hinges. Ability to extend hinges to be taller
Multipurpose 6061 Aluminum Rectangular Tube. 1/16" Wall Thickness, 1" High x 2" Wide Outside	6546K2	\$24.12	1	\$24.12	<u>N-S Member</u>	N-S member: Facilitate N-S movement. Extra material to extend NS member 2 for attachment of actuator.
Multipurpose 6061 Aluminum Rectangular Tube. 1/8" Wall Thickness, 1" High x 2" Wide Outside	6546K39	\$87.88	1	\$87.88	<u>Both Cross bars</u>	Crossbar: Top and Bottom Panel Mount. Extra material for support under crossbar.
Zinc-Plated 1004-1045 Carbon Steel Clevis Pin. 3/8" Diameter, 1-3/4" Usable Length	97245A179	\$8.38	1	\$8.38	<u>Clevis Pin EW</u>	E-W clevis pin. Zinc-Plated
Zinc-Plated 1004-1045 Carbon Steel Clevis Pin. 3/8" Diameter, 1-7/8" Usable Length	97245A347	\$9.36	1	\$9.36	<u>Clevis Pin NS</u>	N-S clevis pin. Zinc-Plated
Zinc Yellow-Chromate Plated Hex Head Screw. Grade 8 Steel, 1/4"-28 Thread Size, 1-3/4" Long	91257A567	\$20.29	1	\$20.29	<u>NS Screw</u>	N-S screws. Zinc Yellow Chromate Plated
High-Strength Steel Hex Nut. Grade 8, Zinc Yellow-Chromate Plated, 1/4"-28 Thread Size	94895A805	\$6.83	1	\$6.83	<u>NS Nut</u>	N-S nuts. Zinc Yellow Chromate Plated
Helping Third Hand Magnifier W/Magnifying Glass Tool - MZ101	291	\$6.95	1	\$6.95	<u>Magnifying Glass</u>	
Mini Leaded 60/40 Solder Spool - 0.3mm Diameter - 50 grams	5941	\$5.95	1	\$5.95	<u>Solder Spool</u>	
Square 60mm x 60mm Soldering Sponge – 3 Pack	3540	\$2.50	1	\$2.50	<u>Solder Sponge</u>	

TERM BLOCK 1POS SIDE ENTRY PCB	1849-1243-ND	\$1.17	6	\$7.02	<u>Terminal Block</u>	
Headers & Wire Housings MiniFit Jr Conn Kit Plug and Recept 6Ckt	538-76650-007 6	\$1.16	2	\$2.32	<u>Molex Connection S</u>	

Appendix E: Engineering Drawings

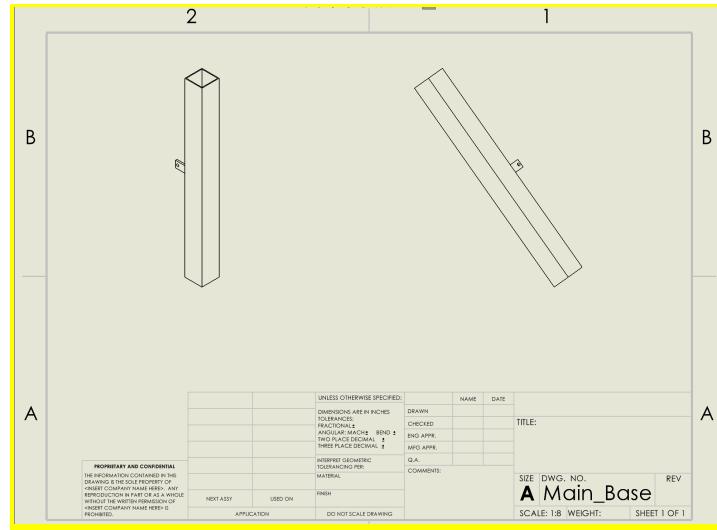


Figure 30: Main Base Drawing

The Main Base (Figure 27) serves as the foundation for the entire structure. It is a vertically oriented rectangular column, which provides stability and support for the other components. This base is constructed with a durable material to withstand environmental stresses and ensure long-term structural integrity.

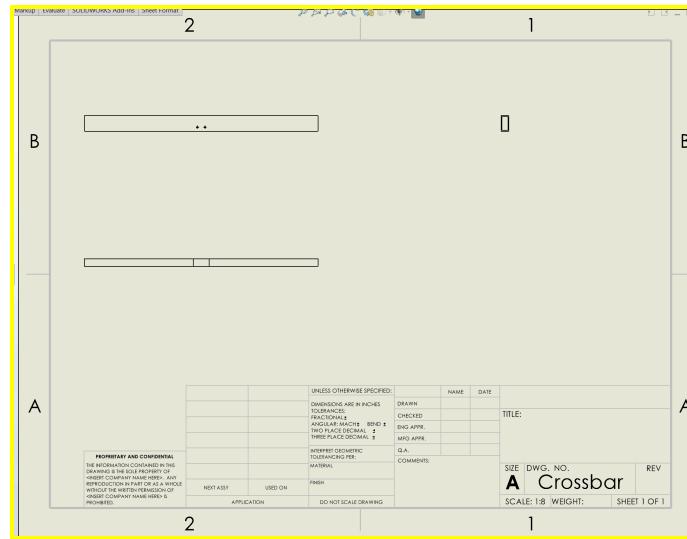


Figure 31: Crossbar Drawing

The Crossbar (Figure 28) is a horizontal component that connects two vertical elements of the structure, enhancing lateral stability. It is designed with precise holes for fasteners, allowing it to be securely attached to adjacent components. The crossbar's simple but sturdy construction makes it a crucial element in resisting external forces that may act on the structure.

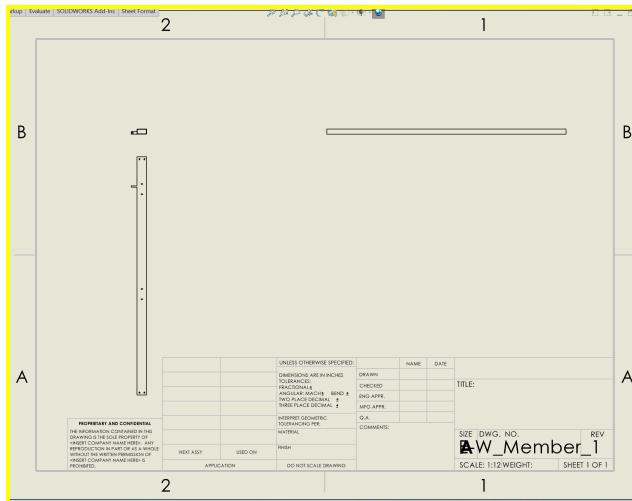


Figure 32: E-W Member 1 Drawing

The W_Member_1 (Figure 29) is a vertical support element that features several alignment holes for flexible attachment points. This member is positioned to reinforce the frame and ensure a balanced load distribution across the structure. The design allows for adjustable connections, making it versatile within the system.

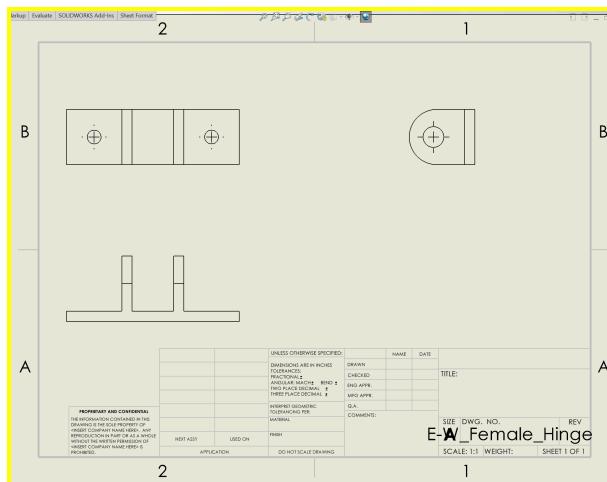


Figure 33: E-W Female Hinge Drawing

The W_Female_Hinge (Figure 30) is part of a hinge assembly that allows for rotational movement. It connects to a complementary male hinge component, enabling parts of the structure to pivot. This hinge system is integral for any application requiring adjustable angles or foldable elements within the structure.

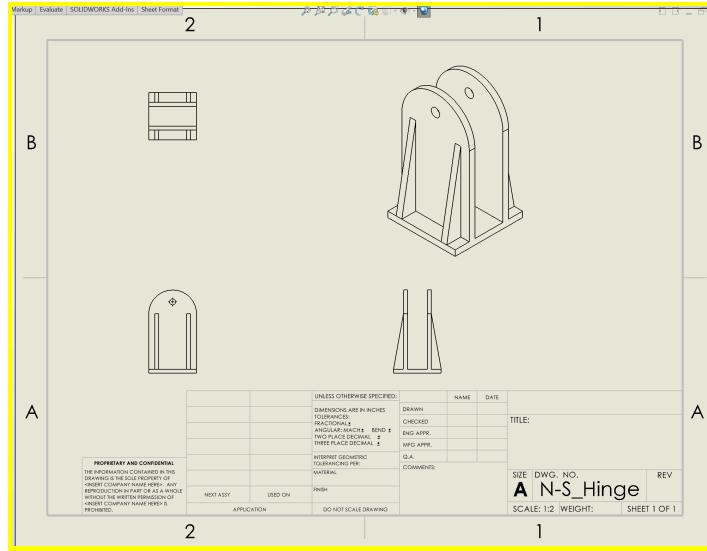


Figure 34: N-S Hinge Drawing

The N-S_Hinge (Figure 31) provides another pivoting joint within the system, oriented along a different axis compared to the W hinge components. This hinge adds a layer of flexibility in the North-South orientation, allowing for dynamic adjustments and adaptability to various operational configurations.

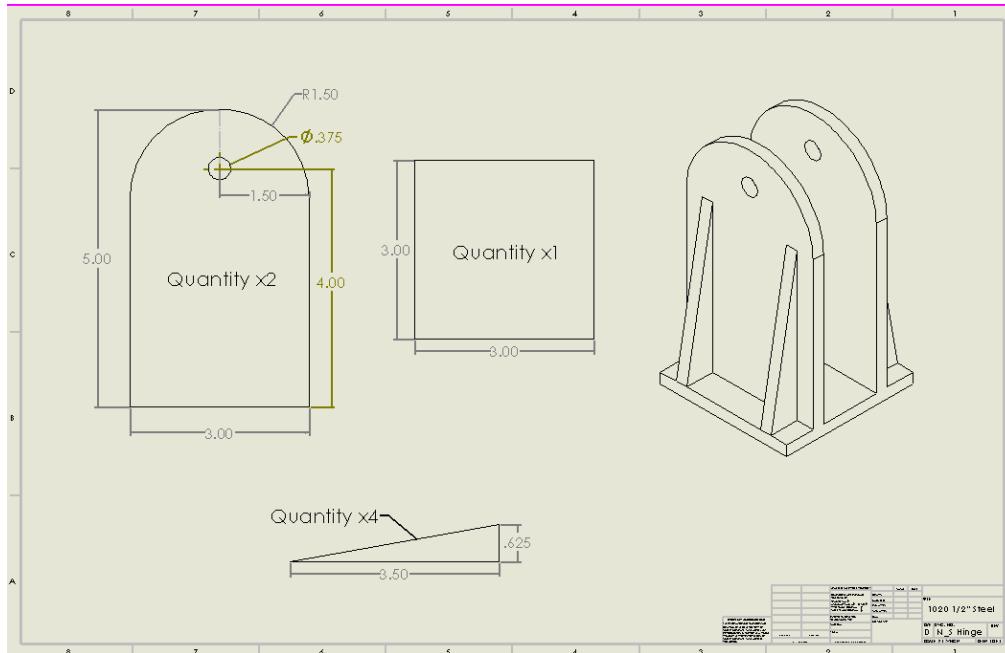


Figure 35: N-S Hinge Individual Part Drawing

The individual pieces manufactured to create the N-S hinge (Figure 32) provides the relevant dimensions for machining purposes. The individual parts will be cut from a 10" by 10" sheet of $\frac{1}{4}$ " 1020 Steel, utilizing a CNC machine.

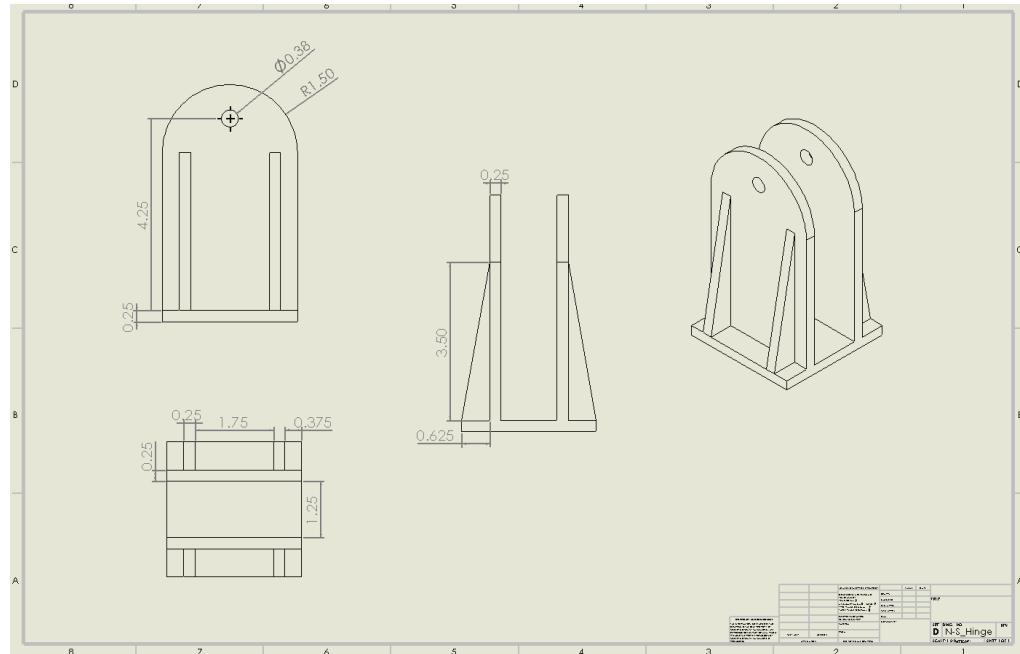


Figure 36: N-S Hinge Assembly Drawing and Dimensioning.

The N-S Hinge assembly (Figure 33) provides the relevant dimensions for welding and final assembly purposes.

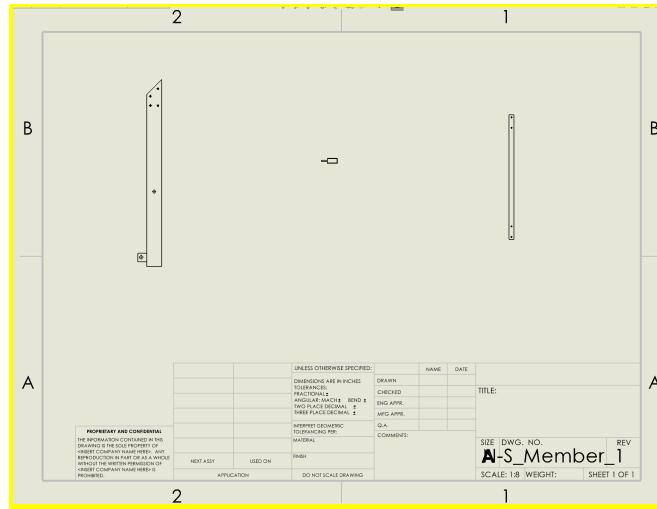


Figure 37: N-S Member 1 Drawing

The S_Member_1 (Figure 34) is a secondary support member, positioned at an angle to distribute forces and contribute to the rigidity of the overall frame. It contains strategically placed holes for secure fastening, ensuring a firm attachment to the main base and crossbar while allowing for modular assembly.

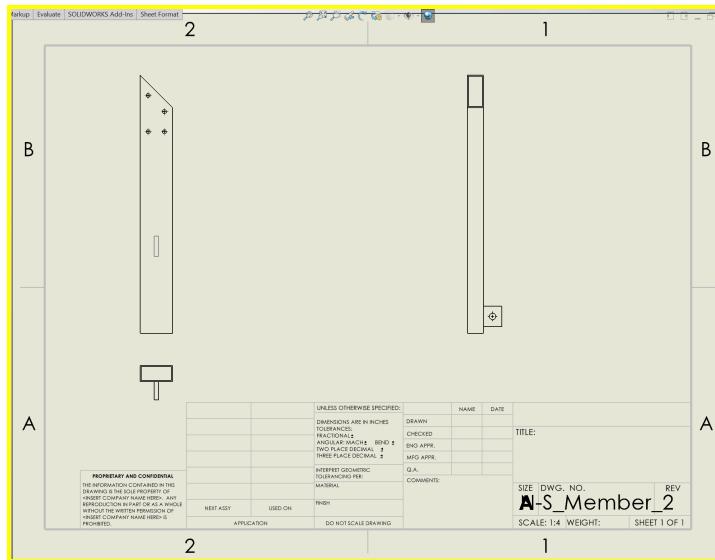


Figure 38: N-S Member 1 Drawing

The S_Member_2 (Figure 35) is a reinforced support element similar to S_Member_1, but with slightly different hole placements and dimensions for customized assembly options. This member provides additional structural support and is essential for stabilizing the assembly during operation.

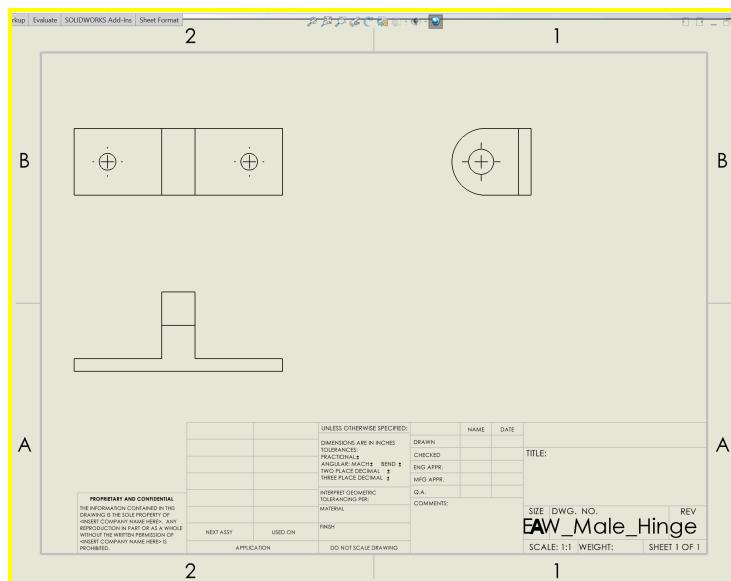


Figure 39: Male Hinge Drawing

The W_Male_Hinge (Figure 36) completes the hinge assembly, pairing with the W_Female_Hinge to form a rotatable joint. This male hinge component interlocks with the female hinge to allow rotational movement, ensuring that connected parts can pivot smoothly without compromising stability.

Appendix F: Project Budget

Table 5: Budgeted parts list for the design, including both mechanical and electrical parts.

Part	Part #	Unit Price	Quantity	Total Price	Store/Link	Comments
Solar Panel	RSP200DC-A SR-US	\$239.99	1	\$239.99	Renogy	200 W Solar Panel
Arduino Uno REV3	ARD_A00006_6	\$29.00	1	\$29.00	Amazon	Arduino model
Raspberry Pi 4 Model B/4GB	4GB-9004	\$55.00	1	\$55.00	Raspberry Pi US	Raspberry Pi model
microSD Card with Raspberry Pi OS	1363-1	\$12.95	1	\$12.95	Raspberry Pi US	Raspberry Pi OS
USB-C Power Supply, 5.1V 3.0A	1203	\$7.95	1	\$7.95	Raspberry Pi US	Raspberry Pi power supply
7inch HDMI LCD	2131	\$77.95	1	\$77.95	Raspberry Pi US	LCD Screen
Logic Level Convert Bi-Directional	BOB-12009	\$2.95	5	\$14.75	electroMaker	Converts 5V signals to 3.3V signals and vice versa to allow raspberry pi to communicate with the arduino
5Pack LM2596 LM2596S DC-DC Step Down Variable Volt Regulator Input 3.0-40V Output 1.5-35V Adjustable Buck Converter Electronic Voltage Stabilizer Power Supply Module	B07PDGG84B	\$12.99	1	\$12.99	Amazon	Drops the voltage from the battery to 5V for Arduino+Pi
ECO-WORTHY 30A Solar Charger Controller Solar Panel Battery Intelligent Regulator with Dual USB Port Auto 12/24V PWM Positive Ground	B08JBLMC33	\$29.99	1	\$29.99	Amazon	Connect panel to battery/system

ECO-WORTHY 12V 30AH LiFePO4 Battery, Rechargeable Lithium Ion Phosphate Deep Cycle Battery for Trolling Motor, Golf Cart, Kids Scooters, Power Wheelchairs, Replacement of 12V 35AH SLA Battery	L13060202003 -1A	\$89.99	1	\$89.99	Amazon	System is connected to the battery
4pcs DC 5V Relay Module 1 Channel Relay Switch Relay Board with Optocoupler Isolation Support High or Low Level Trigger	B095YD3732	\$6.99	2	\$13.98	Amazon	Connect arduino to raspberry pi
2Pcs DC 5V 4 Channel Relay Module Board Shield with Optocoupler Isolation Suport High/Low Level Trigger Relay	B07PSGRFP7	\$11.99	1	\$11.99	Amazon	Connect actuators to arduino
10-12 AWG M5 Ring Terminals	B0D3TM9KQ D	\$9.98	1	\$9.98	Amazon	Adaptor for wire to connect to battery
10awg Silicone Electrical Wire 2 Core Wire 20ft [Black 10ft Red 10ft] 10 Gauge Soft and Flexible Hook Up Oxygen Free Strands Tinned Copper Wire	B0B9JBJ3LK	\$16.98	1	\$16.98	Amazon	Connect battery to solar charge controller
Electrical Wire 14 AWG 14 Gauge Silicone Wire Hook Up Wire Cable 20 Feet [10 ft Black and 10 ft Red] Soft and Flexible 400 Strands 0.08 mm of Tinned Copper Wire High Temperature Resistance (14AWG)	754047498411	\$9.88	1	\$9.88	Amazon	Connect actuators to bus bar

ECO-WORTHY 10FT 10AWG Solar Extension Cable with Female and Male Connector Solar Panel Wire Adaptor for Home, Ship and RV Solar Panels (10FT Red + 10FT Black)	B0CDZVMBP 8	\$17.99	1	\$17.99	Amazon	Connect solar panel to solar charge controller
PowMr 10A Waterproof in-Line Solar Fuse Holder w/Fuse, Black	B08C2PXBKN	\$12.99	1	\$12.99	Amazon	Fuse between panel and solar charge controller
30 Amp MRBF Terminal Fuse Block Kit (Fuse Included) – Single Stud 5/16" (M8) Compact High-Amp Marine Rated Battery Fuse – 58V DC Maximum, IP66 Waterproof (30A)	B0BDMWQS XZ	\$15.99	1	\$15.99	Amazon	
VANTRONIK Waterproof Inline Fuse Holder 6 Pack for Standard Car Fuse, 14 Gauge AWG Wire UL1015,10 Inches Long, Fit for ATO ATC Medium Blade Fuse	B081YDV8PS	\$6.79	1	\$6.79	Amazon	
(25Pcs) 10 Amp Standard Car Fuses, 10A Automotive Fuses(ATC/ATO), Blade Fuse for Car/RV/Truck/SUV/Motorcycle/Boat	B0CHZ3R265	\$3.99	1	\$3.99	Amazon	
uxcell Inline Screw Type Fuse Holder 22 Gauge AWG 5Pcs for 6mm x 30mm Tube Fuse	a19040700ux0 434	\$5.89	1	\$5.89	Amazon	
BOJACK 6x30mm 1 A 1 amp 250 V 0.24x1.18 Inch Volt F1AL250V Fast-Blow Glass Fuses(Pack of 18 Pcs)	B07V5HDVK 7	\$5.99	1	\$5.99	Amazon	

BOJACK 6x30mm 3 A 3 amp 250 V 0.24x1.18 Inch Volt F3AL250V Fast-Blow Glass Fuses(Pack of 18 Pcs)	B07SBH28ZW	\$5.99	1	\$5.99	Amazon	
Mofeez Bus Bar -3x1/4"Post,10x#8 Screw Terminal Power Distribution Block with Ring Terminals(Pair - Red & Black)	B07KVV7F5X	\$19.99	3	\$59.97	Amazon	
Female/Male Ext Jumper Wires 40x3	825	\$3.95	1	\$3.95	electroMaker	Connect everything connected to the Pi and Arduino
Female/Female Jumper Wires - 40 x 3"	794	\$3.95	1	\$3.95	electroMaker	Connect everything connected to the Pi and Arduino
Male/Male Jumper Wires - 40 x 3	759	\$3.95	1	\$3.95	electroMaker	Connect everything connected to the Pi and Arduino
Ferrule Crimping Tool Kit, LUNNEY Wire Ferrule Kit with 1250PCS Wire End Ferrules & Crimper Plier, Self-adjustable Ratchet Tool Set for AWG 23-7 Electrical Wire Connectors, 0.25-10mm ²	B0B23V7ZQJ	\$18.98	1	\$18.98	Amazon	
WWZMDiB ACS712 30A Compatible Current Sensor Module	WWZMDiB ACS712	\$8.99	1	\$8.99	Amazon	
6pcs INA219 I2C Bi-Directional DC Current Power Supply Sensor Breakout Module Compatible with Arduino Raspberry Pi (6pcs)	B08V4X5L5B	\$16.59	1	\$16.59	Amazon	
Adafruit MCP3008 8-Channel Analog-to-Digital Converter with SPI Interface	B00NAY3RB2	\$8.49	1	\$8.49	Amazon	

TCA9548A I2C Multiplexer	2717	\$6.95	1	\$6.95	Adafruit	
TERM BLK 2POS SIDE ENTRY 5MM PCB	0395930602-ND	\$1.04	1	\$1.04	Digikey	
RES 10K OHM 5% 1/4W AXIAL	CF14JT10K0	\$0.10	1	\$0.10	Digikey	
RES 3.3K OHM 5% 1/2W AXIAL	CFM12JT3K30	\$0.10	1	\$0.10	Digikey	
25kHz Semiconductor BTS7960B Motor Driver 43A H-Bridge Drive PWM For Arduino US	176693700253	\$6.79	2	\$13.58	Ebay	
PCB007 1X 12 Position Power Distribution Board 2 Inputs 2 x 13 Outputs for DC AC Voltage New	PCB007	\$6.99	4	\$27.96	Amazon	
Feedback Linear Actuator IP66	PA-04-6-400-HS	\$176.00	1	\$176.00	Actuator 1	Actuator for North-South Movement, 6 inch
Feedback Linear Actuator IP66	PA-04-8-400-HS	\$176.00	1	\$176.00	Actuator 2	Actuator for East-West Movement, 8 inch
Actuator U-Bracket	Model: PA-04	\$8.50	4	\$34.00	U-Bracket	Holds the Actuator
Low-Carbon Steel Sheet. 10" x 10" x 1/4"	6544K24	\$65.67	1	\$65.67	McMaster Hinge	Facilitates N-S movement
Multipurpose 6061 Aluminum Rectangular Tube. 1/16" Wall Thickness, 1" High x 2" Wide Outside	6546K2	\$37.58	1	\$37.58	Grid Support	Extra grid support
Low-Carbon Steel Rectangular Tube. 1/8" Wall Thickness, 3" x 3" Outside Size	6527K424	\$57.60	1	\$57.60	McMaster Base	Main Stem for Base
Multipurpose 6061 Aluminum Bar. 1/8" Thick x 4" Wide	8975K418	\$4.60	1	\$4.60	McMaster Bracket	N-S Bracket: Connects N-S members together
Tight-Tolerance Multipurpose 6061 Aluminum. Precision Ground on All Sides, 2" x	9057K413	\$50.52	1	\$50.52	E-W Hinge Block	E-W hinge. Manufacture hinges out of these. Extra material for

3" x 1-1/4" Thick						more hinges. Ability to extend hinges to be taller
Multipurpose 6061 Aluminum Rectangular Tube. 1/16" Wall Thickness, 1" High x 2" Wide Outside	6546K2	\$24.12	1	\$24.12	<u>N-S Member</u>	N-S member: Facilitate N-S movement. Extra material to extend NS member 2 for attachment of actuator.
Multipurpose 6061 Aluminum Rectangular Tube. 1/8" Wall Thickness, 1" High x 2" Wide Outside	6546K39	\$87.88	1	\$87.88	<u>Both Cross bars</u>	Crossbar: Top and Bottom Panel Mount. Extra material for support under crossbar.
Zinc-Plated 1004-1045 Carbon Steel Clevis Pin. 3/8" Diameter, 1-3/4" Usable Length	97245A179	\$8.38	1	\$8.38	<u>Clevis Pin EW</u>	E-W clevis pin. Zinc-Plated
Zinc-Plated 1004-1045 Carbon Steel Clevis Pin. 3/8" Diameter, 1-7/8" Usable Length	97245A347	\$9.36	1	\$9.36	<u>Clevis Pin NS</u>	N-S clevis pin. Zinc-Plated
Zinc Yellow-Chromate Plated Hex Head Screw. Grade 8 Steel, 1/4"-28 Thread Size, 1-3/4" Long	91257A567	\$20.29	1	\$20.29	<u>NS Screw</u>	N-S screws. Zinc Yellow Chromate Plated
High-Strength Steel Hex Nut. Grade 8, Zinc Yellow-Chromate Plated, 1/4"-28 Thread Size	94895A805	\$6.83	1	\$6.83	<u>NS Nut</u>	N-S nuts. Zinc Yellow Chromate Plated
Helping Third Hand Magnifier W/Magnifying Glass Tool - MZ101	291	\$6.95	1	\$6.95	<u>Magnifying Glass</u>	
Mini Leaded 60/40 Solder Spool - 0.3mm Diameter - 50 grams	5941	\$5.95	1	\$5.95	<u>Solder Spool</u>	
Square 60mm x 60mm Soldering Sponge – 3 Pack	3540	\$2.50	1	\$2.50	<u>Solder Sponge</u>	

TERM BLOCK 1POS SIDE ENTRY PCB	1849-1243-ND	\$1.17	6	\$7.02	<u>Terminal Block</u>	
Headers & Wire Housings MiniFit Jr Conn Kit Plug and Recept 6Ckt	538-76650-007 6	\$1.16	2	\$2.32	<u>Molex Connection</u> S	

The electronics components in Table 4 consist of a power management and control system that integrates solar energy capture, data processing, and system operation. A Renogy solar panel generates power, which is regulated by a solar charger controller and stored in a lithium battery from ExpertPower. An Arduino Uno, supported by a Raspberry Pi and components like the MCP3008 ADC and logic level converters, manages data collection and processing. Additional accessories, such as the 7-inch LCD screen and relay modules, enable interaction and control over system operations, ensuring smooth and efficient functionality across the device.

The mechanical components in Table 4 provide the structural integrity and movement capabilities needed to optimize solar positioning. Actuators facilitate North-South and East-West adjustments, guided by robust steel and aluminum brackets, hinges, and members sourced from McMaster. The design integrates a main base for stability, complemented by crossbars and N-S and E-W members to secure and orient the solar panel accurately. These components ensure the durability and alignment necessary for precise tracking and maximize solar energy capture across different positions.

The summary of expenses captures all financial elements related to the project, including raw materials, purchased components, and labor costs. Each item is listed in a detailed table of expected costs, specifying quantity, unit price, and total cost. Additionally, components acquired through the university, vendor donations, or discounts are clearly marked as in-kind contributions. The table not only provides a clear picture of total project spending but also facilitates future budgeting for similar projects by clarifying actual costs incurred versus the theoretical value of in-kind support.

Given the precision required, the team decided to purchase certain mounting assemblies, which would be difficult to achieve within project deadlines if fabricated in-house. Although slightly more expensive than DIY fabrication, the purchased components saved valuable time and contributed to overall structural reliability.

The project's energy storage solution differs between experimental setup and full-scale deployment. Currently, a standalone battery system is used to store electricity generated from the solar panel, which allows for immediate energy access and efficient project testing. In a real-world application, however, this energy storage would integrate with LMU's power grid, enabling a more sustainable energy solution by routing the electricity directly into campus infrastructure rather than relying on individual battery storage.

All mechanical components, including brackets, hinges, and structural materials, were sourced directly from McMaster-Carr. McMaster was selected as the primary supplier due to its extensive catalog of high-quality industrial parts, fast shipping, and reliable availability. By sourcing standardized mechanical components from McMaster, the team ensured that all parts would meet stringent quality and durability standards necessary for the project's structural

integrity. This choice also streamlined ordering and reduced lead times, making McMaster an ideal partner for the project's mechanical needs.

To calculate a rough ROI for this project, we consider both the initial project costs and potential long-term savings or value generated. The total initial investment for the project, based on the provided costs, includes \$406.83 for mechanical parts sourced from McMaster, \$578.35 for electronic components, \$681.98 for major components including the actuators, solar panel, and battery, and \$574 for manufacturing, totaling \$2,241.16 in overall costs. At a larger scale and larger quantities of manufacturing, it is estimated that the total cost would drop significantly.

Appendix G: Project Academic Assessment

Course	Support
EECE 2110 Circuits I	Introduced the electrical team to wiring and wiring diagrams.
ENG 100 Intro to Engineering Analysis	Led our team members to pursue a Mechanical Engineering degree.
ENG 160 Algorithms and Applications	Introduced the electrical team to writing software for engineering applications.
ENG 200 Statics	Introduced the mechanical team to calculations involving static objects.
MECH 211 Computer Aided Design	Taught team members how to use SolidWorks.
MECH 212 Mechanics of Materials	Taught mechanical team members how to calculate stresses on a standing object.
MECH 214 Materials Science	Taught team members how to decide what materials to use in designing systems for strength and mobility.
MECH 310 Machine Design	Taught team members how to design machines with moving parts.
MECH 321 Energy Systems	Introduced team members to alternate energy systems, like solar energy.
MECH 412 Control Systems	Introduced team members to control system design, including how they work and function.