Development of a Solar Array Deployment Mechanism for a CubeSat

a project presented to
The Faculty of the Department of Aerospace Engineering
San José State University

in partial fulfillment of the requirements for the degree

Master of Science in Aerospace Engineering

by

Christopher J. Hernandez

May 2025

approved by

Dr. Periklis Papadopoulos
Faculty Advisor

San José State
UNIVERSITY

ABSTRACT

Development of a Solar Array Deployment Mechanism for a CubeSat

Christopher J. Hernandez

CubeSats could be used as a low-cost platform to test experimental technologies and to conduct technology demonstrations. The underlying objective of this project was to design and develop a low-cost SADM that has sun-tracking capabilities and axis controllability by utilizing a system of four NEMA 11 motors. Initially, a two-motor configuration was considered, in which one motor would provide initial deployment about one axis and the other would provide controllability about the axis normal to the chassis wall surface. To achieve this, rotary disks, spur-gear, rail-block, and linkage-based deployment systems were considered, but were later disregarded due to volume limitations and lever arm restrictions. Instead, the SADM was downsized to permit controllability about the axis normal to each chassis wall surface. Rather than use a motor for initial solar array deployment, a set of torsion springs was used. The retention mechanism--fishing string stowing the solar panels--was burned by a burn-wire mechanism to permit initial deployment. By 3D printing the hinge and purchasing the torsion springs, single-set and double-set torsion designs were designed and developed. The single-set torsion spring design deflected each solar array by 45°, whereas the doublet-set torsion spring deflected each solar array by 90°. Blockers integrated on the hinge mechanism prevented the solar array from deflecting past 90°. The single-set torsion spring hinge was attached to the rotary disk of the motor, permitting rotation about the normal axis to the chassis surface. All structural components were 3D printed using PETG HF. By using a screw-based design, the CubeSat was assembled by using heat-set inserts and M2 screws. Component-level testing and integration-level testing were conducted to verify the functionality of all sensors, whereas the latter was used to verify the functionality of the ambient light sensor array and the NEMA 11 motors. System-level testing verified the functionality of the light sensor array, current sensors, and motors operating together. Solar panel testing has shown that a fully deployed solar array will generate more voltage than solar arrays in the partially deployed and stowed configurations. Numerically, the voltage generated in each respective generation for a particular array were 0.7 V, 0.75 V, and 1.05 V, respectively. Development costs of the SADM amounted to $625. Considering that existing COTS components and assembly could lead to CubeSat costs amounting to over $10,000, the underlying objective of this project was fulfilled. However, the developed CubeSat SADM could be further improved by transitioning from a breadboard prototype design to a PCB-based design with an integrated burn-resistor circuit. Furthermore, rather than using an external 12V power supply, the EPS could be further developed by incorporating energy storage capabilities through the use of boost converters, buck converters, and switches where appropriate.

Acknowledgements

Dedicated to my parents, who have been with me since the very beginning. Their undying support has inspired me to give everything in life my all.

Special thanks to Dr. Periklis Papadopoulos for his systems engineering insights and recommendations during the ADCS senior design project. That project ultimately led to my special interest in CubeSats. His recommendations and insights were implemented in this project to streamline the design and prototyping process. Dr. Papadopoulos's guidance and recommendations were invaluable.

I would also like to thank the entire Aerospace Engineering Department for their enthusiasm and extensive knowledge across all subjects.

Table of Contents

List of Figures

List of Tables

Symbols

Symbol	**Definition**	**Units (SI)**
Greek Symbols		
ε	Efficiency Change	-------
η	Efficiency	-------
λ	Temperature Coefficient	% K^{-1}
ω	Angular velocity	Rad s^{-1}
τ	Shear Stress	Pa
θ	Incidence Angle	Radians
Subscripts		
θ spring	Torsion Spring Deflection Angle	°
σ b	Torsion Spring Bending stress	Pa
σ y	Yield Strength	Pa
C F	Correction Factor	-------
D mean	Wire Mean Diameter	m
d inner	Inner Wire Diameter	m
d outer	Outer Wire Diameter	m
d wire	Wire Diameter	m
F n	Normal Force	N
G eff	Effective Solar Irradiance	W m^{-2}
I LSB	Current resolution	mA
In Max	Max Input	-------
K B	Bending Stress Correction Factor	-------
K Wahl	Wahl Correction Factor	-------
K Berg	Bergsträsser Correction Factor	-------
KE Panel	Rotational Kinetic Energy of the Solar Panel	J
L leg	Wire Leg Length	m
R shunt	Shunt resistor	Ω
T cell	Cell Temperature	K
T spring	Torque of the Torsion Spring	Nm
U spring	Potential Energy of the Torsion Spring	J
V In	Input Volage	V
V LSB	Voltage resolution	mV
V Mot	Motor Voltage	V
Acronyms		
A	Area	m^{2}
ACS	Attitude Control System	-------
ADCS	Attitude Determination Control System	-------
BLDC	Brushless DC	-------
C	Spring Index	-------
CAD	Computer-Aided Design	-------
COTS	Commercial Off-The-Shelf	-------
CSD	Canisterized Satellite Dispenser	-------
DH	Double-Hinge	-------
DIR	Direction	-------
E	Young’s Modulus	Pa
EMI	Electromagnetic Interference	-------
EPS	Electrical Power System	-------
FOV	Field of View	-------

G	Solar Irradiance	$\mathrm{W\ m^{-2}}$
GOES	Geostationary Operational Environmental Satellite	-------
GND	Ground	-------
H	Height	m
HF	High Flow	-------
I	Moment of Inertia	$\mathrm{kg\ m^2}$
$\mathrm{I^2C}$	Inter-Integrated Circuit	
ICA	Irradiated Cell Area	$\mathrm{m^2}$
IoT	Internet of Things	-------
k	Spring Constant	$\mathrm{N\ m\ rad^{-1}}$
L	Length	m
LS	Light Sensor	-------
LSA	Light Sensor Array	-------
LSB	Least Significant Bit	-------
MISP	Maximum Irradiated Solar Panels	-------
MCU	Microcontroller Unit	-------
MH	Multi-Hinge	-------
MOI	Moment of Inertia	$\mathrm{kg\ m^2}$
MPO	Maximum Power Output	W
n	Number of active coils	-------
NOAA	National Oceanic and Atmospheric Administration	-------
OTR	Operating Temperature Range	K
PCB	Printed Circuit Board	-------
PETG	Polyethylene Terephthalate Glycol	-------
PLA	Polylactic Acid	-------
PSU	Power Supply Unit	-------
PV	Photovoltaic	-------
PWM	Pulse Width Modulation	-------
R&R	Retention and Release	-------
RD	Rotary Disk	-------
S	Signal	-------
SADA	Solar Array Drive Assembly	-------
SADM	Solar Array Deployment Mechanism	-------
SCL	Serial Clock Line	-------
SDA	Serial Data Line	-------
SF	Safety Factor	-------
SH	Single-Hinge	-------
SMA	Shape Memory Alloy	-------
SP	Solar Panel	-------
STC	Standard Test Conditions	-------
SVR	Shunt Voltage Range	mV
T	Period	s
TCA	Total Cell Area	$\mathrm{m^2}$
U	Unit	-------
UNK	Unknown	-------
V	Voltage	V
W	Width	m

1. Introduction

1.1 Motivation

It is within humanity's nature to push the technological envelope to new heights for what was once inconceivable. It is within humanity's nature to discover the unknown. In 1903, the Wright brothers invented and flew the world's first airplane [1]. From this, the foundations of aerodynamics were established. In 1947, the first airplane broke the sound barrier, setting the stage for supersonic flight [1]. By 1957, Sputnik 1, humanity's first artificial satellite, was deployed in space [1]. 1961 marked the year when the first human entered space [1]. In 1969, Neil Armstrong ushered the sentence, "That's one small step for [a] man, one giant leap for mankind [2]." Despite all the technological limitations at the time, humankind persevered and made tremendous contributions to science.

Launched in 1977, the Voyager probes underwent a flyby around Jupiter, Saturn, Uranus, and Neptune before being sent into interstellar space [3]. Although the environment around the sun contains intense heat and radiation, the Parker Solar Probe continues to observe and measure the Sun [4]. Around the Earth, constellations of weather satellites are used to image Earth's surface and to collect atmospheric data. However, the consequence of obtaining scientific data was the expensive time and monetary costs of developing and maintaining these satellites. $865 million was allocated to create the Voyager probes [5]. The Parker Solar Probe was projected to cost $1.1 billion [6]. NOAA's GOES-R weather satellite constellation has a lifecycle budget of $10.8 billion [7]; the cost to launch the GOES-U satellite is ~$153 million [8].

CubeSats are nanosatellites that can be constructed with minimal time and monetary expenditures. CubeSats are typically developed within 18 – 24 months but could be developed within nine months [9,10]. Originally developed within academia, CubeSats provide a low-cost, quick-to-develop platform which can be used to conduct scientific missions and technology demonstrations [10]. CubeSats have begun incorporating larger form factors to support larger, more complex payloads. CubeSats have implemented deployable solar arrays to meet the increased power demand of these systems. As CubeSats incorporate increasingly complex instruments, the power requirement will only increase. Thus, this paper will develop and quantify the power generation capabilities of a sun-tracking SADA system.

1.2 Literature Review

1.2.1 CubeSats Background

CubeSats are nanosatellites originally developed by California Polytechnic State University and Stanford University in 1999 [11]. Adhering to a design standard, an individual CubeSat unit (1U) has a size and mass limit of 10 cm x 10 cm x 10 cm and 1.33 kg, respectively; [11] provides comprehensive CubeSat design specifications. CubeSats can be deployed individually or combined with other units to form a larger nanosatellite. Typically, CubeSats have a form factor between 1U and 12U, but smaller and larger units have been deployed or are in development [12].

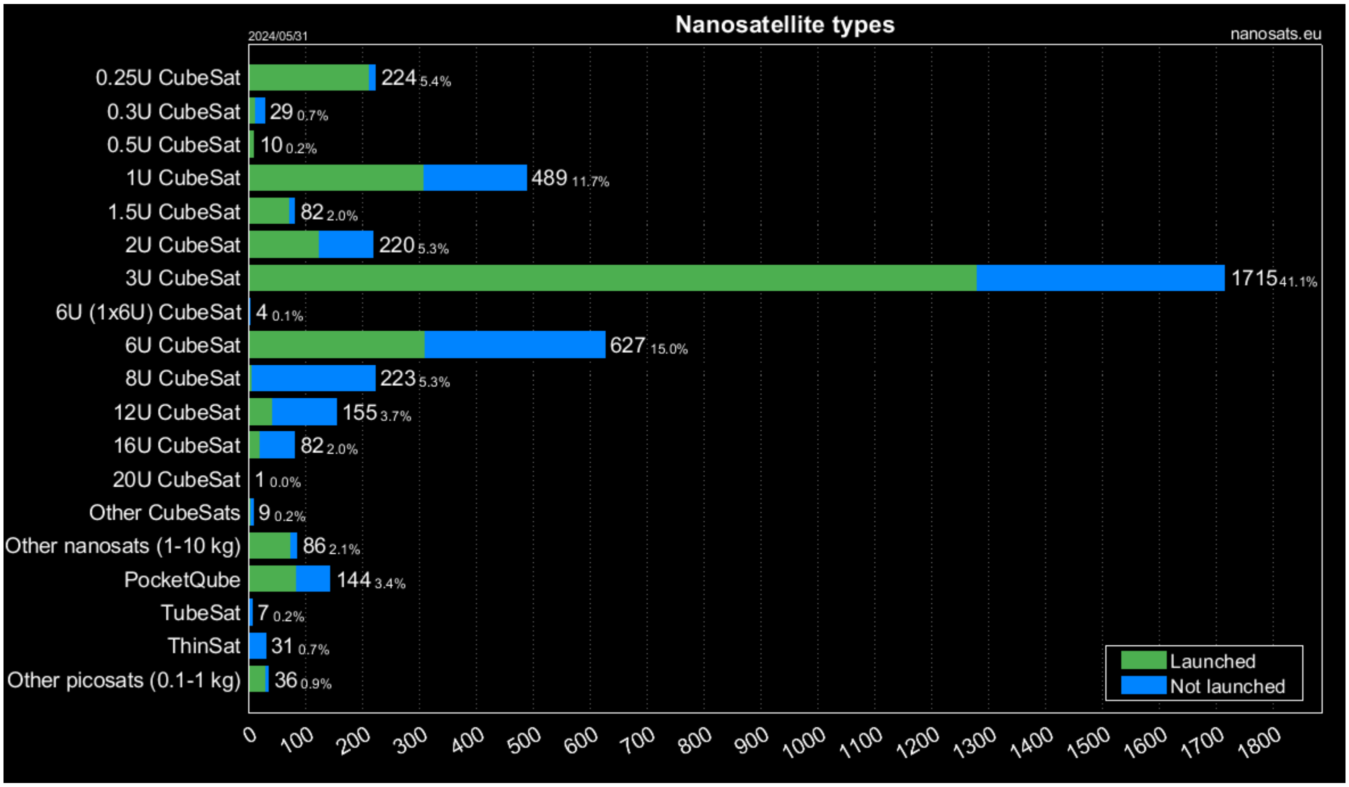


Figure 1.1. CubeSat size composition and launch status [12].

CubeSats were developed to provide a low-cost and time-investment development platform compared to conventional satellites. CubeSats have been used for educational research, conducting technology demonstrations or scientific missions. Table 1.1 provides a list of CubeSat missions performed by government agencies and academic institutions; [13] provides a list of CubeSat missions ranging from 1998 to the present.

Table 1.1. Previous, ongoing, and future CubeSat science missions.

Science Mission	Mission Objective
BurstCube	The study of gamma-ray bursts [14].
CeREs	The study of charged particle dynamics within Earth's radiation belts [15].
CuPID	The measurement of soft x-rays and the testing of magnetic reconnection models at Earth's magnetopause [16].
CuSP	The measurement of particles and magnetic fields in space [17].
E-TBEx	The study of how communication disruption bubbles are formed within Earth's ionosphere [18].
ELFIN	The investigation of how electrons escape the Van Allen radiation belts [19].
MarCo	The testing of deep space communication equipment [20].
PREFIRE	The measurement of Earth's radiation emissions to study Arctic heat emissions [21].
SunRISE	The study of solar activity to determine how the Sun generates solar particle storms [22].
TechEdSat-15	The testing of an exo-brake designed to operate within high-temperature environments [23].

1.2.2 Launch Vehicle Failures

CubeSats are typically the secondary payload of a launch vehicle and are deployed via a CSD. Before this, however, CubeSats must follow all design specifications in [11] and conduct testing and verification as specified in [24]. One of the design requirements specified in [24] is that the CubeSat must withstand the accelerative loads and vibrations within the launch environment. Considering that the CubeSat designers performed all the necessary testing to meet the requirements, there is a possibility that the launch vehicle will fail. [25] provides the number of launch vehicle missions and failure rates from 1957 – 2017; Figure 1.2 provides the number of launch vehicle missions, number of failures, and failure rates from 1957 – 2017, whereas Table 1.2 lists the average failure rate via 10-year intervals. This data shows that launch vehicles experienced a significant failure rate throughout 1957 – 1961. Although the number of launches increased from 1961 to 1965, the number of launch vehicle failures and failure rates decreased. Throughout the succeeding decades and up until now, the failure rate decreased before settling around 6%. Jacklin [25] suggests that the failure rate of the launch vehicles has not improved because when a new launch vehicle is introduced, a technological learning curve is introduced.

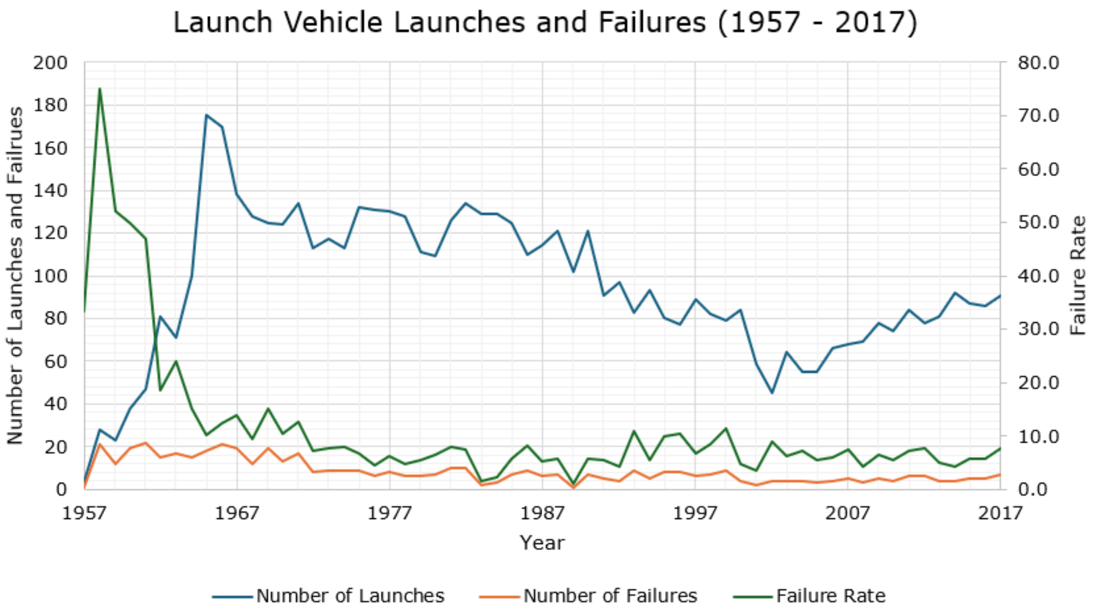


Figure 1.2. Number of launch vehicle launches, failures, and failure rate from 1957 – 2017 [25].

Table 1.2. The failure rate of launch vehicles based on a 10-year period from 1957 – 2017 [25]. *This data point was extended by an additional year.

10 Year Period	Average Failure Rate
1957 – 1966	33.7
1967 – 1976	9.6
1977 – 1986	5.6
1987 - 1996	6.0
1997 – 2006	6.9
2007 − 2017*	6.1

1.2.3 CubeSat Points of Failure

CubeSats are composed of multiple subsystems—mechanical, electrical, communications, etc—, critical to ensure mission success. If any components—reaction wheels, microcontroller, power delivery board, etc.—within these subsystems fail, the CubeSat mission will likely result in a partial, if not total, failure. From 2002 – 2016, CubeSats experienced a 48.1% failure rate; 139 of 270 have failed due to power, mechanical, communications, or system-level issues [26]. In a survey sent in a Cal Poly CubeSat workshop—the sample size is thirty-five individuals—48% of respondents experienced tool-related issues. Communication issues were based on radio, transceiver, and power failures [26]. In a research study [27], researchers interviewed academic, commercial, and government organizations to determine the causes of CubeSat failures. These researchers found that a CubeSat would be launched regardless of its readiness, selected components were not radiation hardened, and there was insufficient time to conduct subsystem-level testing [27]. The researchers found that the most common points of failure involved the communications system, power system, and deployables [27]. From these categories, CubeSat developers could not find a sufficient ground segment—ground station, control center, etc.—the purchased power system did not match specifications, and that burn-wire systems were not easily resettable, respectively [27]. Researchers from [28] examined 178 CubeSats until 2014 and categorized the points of failure. Generally, the electrical power system, on-board computer, and communications system were responsible for most known failures [28]. Conversely, CubeSats rarely failed due to the ADCS, payload, structure and deployables [28]; failures related to the deployables do not include the antennas.

Table 1.3. CubeSat categorized failures from 0 – 90 days [28].

	Dead on Arrival	30 Days	90 Days
Electrical Power System	28%	44%	36%
On-Board Computer	16%	20%	21%
Communication System	14%	16%	29%
Payload	3%	4%	0%
Structure and Deployables	3%	4%	0%
ADCS	3%	0%	0%
Unknown	33%	12%	14%

There are cases where the CSD itself will fail to deploy the CubeSat. This was the case for the 2014 Orb-1 and Orb-2 missions, in which the CSD failed to deploy its payload [29]. The deployer in the Orb-1 mission failed due to an electronics issue, whereas the deployer in the Orb-2 mission failed due to excessive loading of the ball-lock mechanism [29].

1.2.4 Solar Arrays

1.2.4.1 Body-Mounted Arrays

CubeSats generate power through PV cells, which can be placed on the CubeSat in various arrangements. The first arrangement is where the cells are body-mounted on the walls of the CubeSat (Figure 1.3). While this arrangement is the least complex, this arrangement generates the least amount of power. For a 1U CubeSat, this arrangement allows up to 1 W or 3 W of power to be generated depending on whether the cells are silicon or gallium arsenide-based [30]. For a 3U CubeSat, the maximum power generated using advanced triple junction cells is less than 10 W [31].

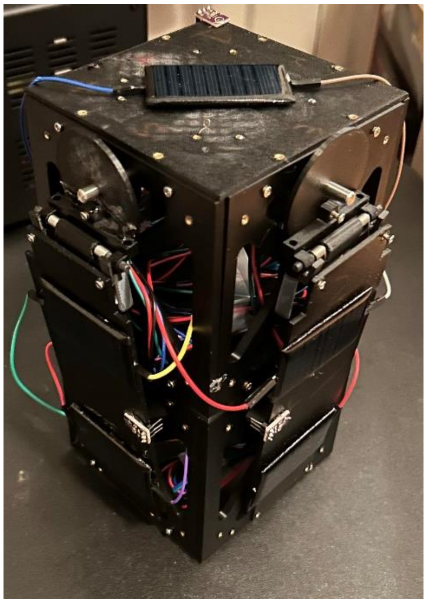


Figure 1.3. 2U CubeSat with body-mounted solar cells.

1.2.4.2 Deployable Solar Arrays

The other arrangement involves a deployable solar array. In this configuration, the solar array is initially stowed; the solar panels are initially attached to the CubeSat's walls. The solar array is then deployed/extended via a deployment mechanism (Figure 1.4). The deployable solar array arrangement offers more power generation capabilities than the previous arrangement because more solar panels are exposed to the sun. 3U CubeSats using such an arrangement have generated between 22W – 56W of power [31].

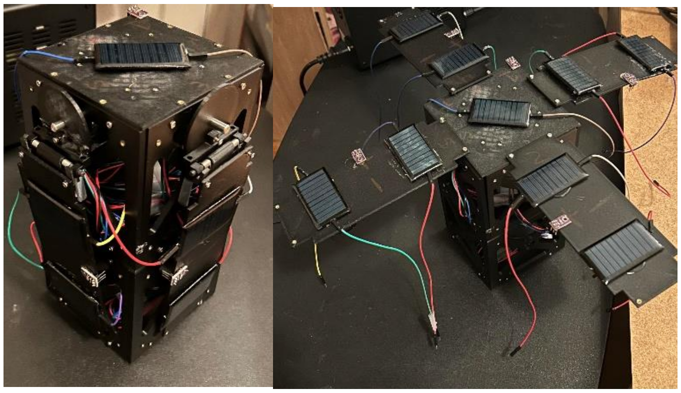


Figure 1.4. 2U CubeSat with a stowed solar array (left) and a deployed solar array (right).

Solar arrays can be deployed based on single or multi-hinge designs. In single-hinge designs, a singular hinge connects each solar array to the CubeSat's body. Multi-hinge designs follow the single-hinge design, except that the solar panels are stacked/folded together; secondary hinges connect the solar panels. These hinges then unfold the solar panels. The amount of solar panels that can be stacked is limited by the space available within the CSD. In the multi-hinge designs, the solar panels can be connected with respect to the longitudinal or lateral axis of the CubeSat [31]. Deployment with respect to the longitudinal axis compared to the lateral axis has the following advantages:

- Symmetry provides a lesser disturbance torque during deployment [31].
- The motion of the solar panels and SADA have a lesser disturbance impact [31].
- In a possible failure event during deployment, the solar panels cannot impact the CubeSat [31].
- More space for the installation of the secondary hinges [31].

However, the disadvantage of a longitudinal system is that the larger CubeSat's MOI could potentially impact attitude control [31]. [31] developed a multi-hinge longitudinal deployment system; two wires are cut to deploy the solar array. The primary hinge rotates 90°, whereas the secondary hinges rotate 180°.

1.2.4.3 Sun-Tracking Solar Arrays

Deployable solar arrays provide more power generation capabilities than body-mounted solar arrays. However, power performance can be further improved by adding sun-tracking capabilities to deployable solar arrays. [32] developed a prototype SADA externally placed at the ends of a CubeSat. Sun sensors were not yet implemented, but instead, a temporary model was used to determine the Sun's position. A stepper motor-gear system and motor drivers were used to rotate the output shaft, which is used to provide attitude control [32]. To prevent overwinding of the cable twists—these were used to transfer power from the solar array—the rotation of the output shaft was limited by mechanical hard stops [32]. This system uses one mW when idle and 500 mW when actively providing attitude control [32].

1.2.5 Solar Array Release Mechanisms

1.2.5.1 Burn Wire Release Mechanism

Deployable solar arrays can be deployed via a burn wire release mechanism. [33] developed a nichrome burn wire release mechanism that can be used to deploy antennas, solar arrays, and other deployables (Figure 1.5). In this mechanism, a heated nichrome burn wire cuts through a Vectran tie cable, which releases the deployable; a current is applied to heat the wire [33]. The optimal current supplied through the wire was determined to be 1.60 amps [33]. The developers chose to use a spring-loaded nichrome burn wire system because the electronics were inexpensive and could be powered by the CubeSat's bus [33]. To avoid heat loss issues, the minimum length of the wire should be at least 10 mm [33]. Conversely, [33] found that the maximum length of the wire should not exceed 32 mm, otherwise the structural integrity of the wire would be compromised. The burn wire mechanism operated fifty times before the nichrome wire had to be replaced; over four hundred tests were conducted without the mechanism failing [33]. Despite the high reliability of this system and relatively low cost, there are some design issues:

- Inadequate tensioning of the tie-down cable can lead to a spring preload loss [33].

- Improperly securing the nichrome wire could prevent the burn wire from cutting the tie-down cable [33].
- The burn wire could short out before the tether is cut [34].
- The tether can become tangled upon deployment [34].

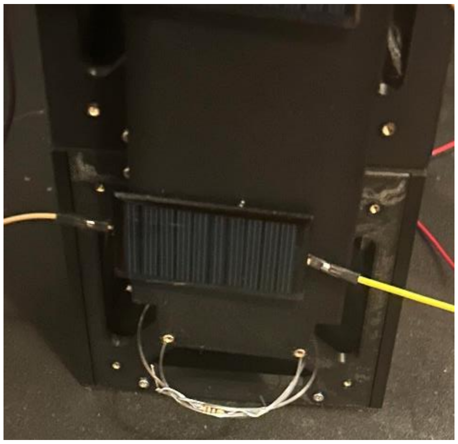


Figure 1.5. Burn wire resistor release mechanism.

1.2.5.2 Shape Memory Alloy Release Mechanism

SMAs have seen usage as a deployment mechanism. SMAs are a type of material that returns to its original shape when heated. In the case of deployment mechanisms, SMAs are advantageous compared to consumable mechanisms because SMAs are lightweight, produce low shocks, and can be designed to be resettable [34]. In [34], two forms of SMAs were used: thermally activated and mechanically activated. The former used SMAs for a two-stage R&R mechanism, whereas the latter used SMAs as deployment springs [34]. The first stage of the R&R mechanism uses a pin-puller device that is activated by a SMA actuator [34]. Once the SMA wire is heated, the wire contracts to pull a pin, which releases the second stage [34]. Once this occurs, four compression springs deploy the solar arrays, which would then be locked into place [34]. Before using the pin-puller system, the authors of [34] considered a piston-based and collet-based design. The previous design was not used because the vibrations experienced within the launch environment could cause a premature deployment [34]. The second design was not developed due to mass and time budget limitations [34]. There are some critical SMA potential failure points:

- The SMAs could prematurely activate when the ambient temperature exceeds the activation temperature [34]. The mechanism developer resolved this issue by selecting an SMA with an activation temperature of 100°C [34].
- The SMA hinge springs can fracture due to fatigue and overstraining [34]. This issue was resolved by adding a washer to prevent the formation of cracks [34].

1.3 Project Objective

Existing deployment mechanisms have used nichrome burn-wires and SMAs to deploy solar arrays. The limitation of these mechanisms is that once the solar arrays are deployed, there is no form of attitude control; the arrays remain fixed when deployed. The power generation capabilities of CubeSats can be further increased by adding sun-tracking and attitude control capabilities to a SADM. Thus, the underlying objective of this project is to design and develop a low-cost and reliable SADM that utilizes sun-tracking capabilities for the solar arrays of a CubeSat. The power generated with this new design will be compared with the other solar array configurations: body-mounted, single-hinge deployable, and multi-hinge deployable solar arrays. A vibrational and dynamics analysis will then be conducted based on the deployment of the solar arrays. Once these tasks have been completed, all project objectives will be extended to a 2U, 3U and 6U CubeSat.

1.4 Methodology

1.4.1 Project Scope

The purpose of this project is to develop a SADM that could be integrated within a CubeSat. The following will not be implemented within the CubeSat:

- An ADCS incorporating reaction wheels, magnetorquers, etc.
- A communications system involving antennas, ground stations, etc.

Although these systems will not be implemented, a design goal is to create a compact SADM.

1.4.2 CubeSat Unit Construction

A CubeSat chassis is required to house and provide structural support for all necessary components. Although COTS components can be purchased, the costs would exceed the project's budget, and the COTS components cannot fulfill the project's custom design needs. Instead, CubeSat chassis with sizes of 1U – 6U will be designed within SolidWorks and then 3D printed. The critical design parameters involve a lightweight chassis that minimizes vibration propagation while maximizing structural integrity.

Table 1.4. CubeSat chassis COTS costs [35–38].

CubeSat Chassis	Cost
1U	$995 – 1,395
2U	$2,339.90
3U	$6,574
6U	$15,487.94

1.4.3 SADM Design

1.4.3.1 Deployment Sequence

A 2-axis ACS for the solar array will be developed by incorporating a system of motors, hinges, and a rotary base. Figure 1.6 shows that the rotary base allows rotation about the Y-axis, whereas the hinges allow rotation about the X-axis. Initially, the solar panels will be stowed (Figure 1.7). During the deployment sequence, the primary hinge rotates the primary solar panel by 90° (Figure 1.8). Next, the secondary hinge rotates the secondary panel by 180°. A similar process would occur if a third solar panel were attached; the number of attachable solar panels depends on the space available within a CSD. Once deployed, the rotary base can rotate the solar array about the Y-axis. Similar degrees of rotation are provided among the other surfaces of the CubeSat, excluding the top and bottom surfaces.

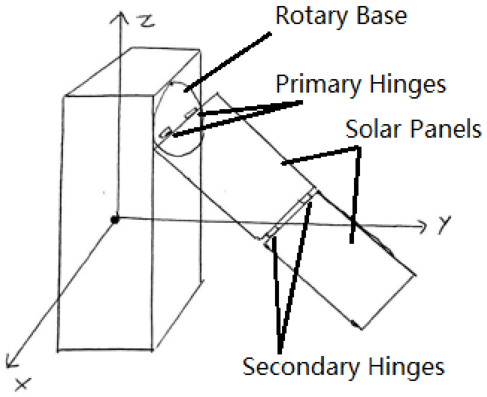


Figure 1.6. Deployed solar array for a 3U CubeSat.

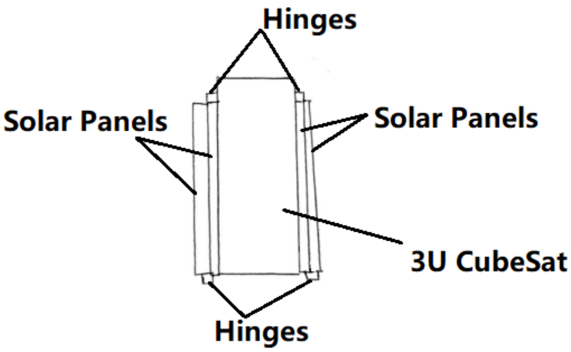


Figure 1.7. Stowed solar panels for a 3U CubeSat.

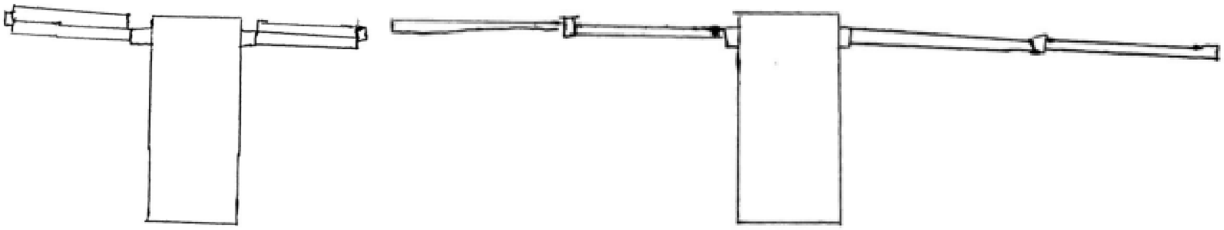


Figure 1.8. Deployment sequence for a 3U CubeSat.

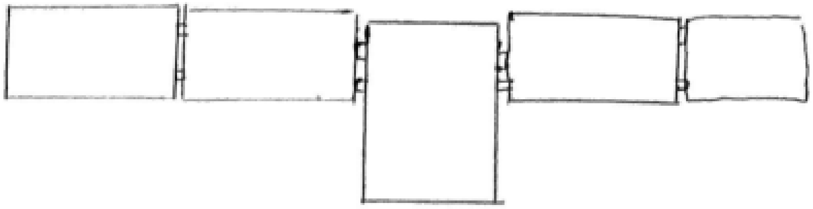


Figure 1.9. The rotary base allows rotation about the CubeSat's Y-axis.

1.4.4 SADM Electronics

Electronic components necessary to develop the mechanism and control system include an MCU, sun sensors, electric motors, solar panels, and a power supply; a PCB can be implemented to make the system more compact. To develop a flight-ready SADM for LEO deployment, all selected electrical components must withstand an operating temperature range of -65 to 125°C.

1.4.4.1 Motor Usage

In CubeSat missions, BLDC motors drive the reaction wheels within ADCS systems. BLDC motors generate less noise and have a longer lifetime than brushed DC and stepper motors. Stepper motors provide great point precision and holding torques but suffer from power waste and excessive thermal heating compared to BLDC motors [39]. Despite the issues of stepper motors, these motors will be used due to their attitude precision. Stepper motors will rotate the primary and secondary hinges by a fixed amount. Stepper motors provided by Applied Motion Products and FAULHABER were examined. However, the products examined in Table A.1 and Table A.2 do not operate within the specified temperature range. Stepper motors provided by Ducommun [40], phytron [41], and American Motion Systems [42] are suitable for operating within the desired temperature range. However, quotes are required to acquire the costs. To minimize project costs, stepper motors that do not operate within the space qualification temperature range will be used for the prototype.

1.4.4.2 Solar Panels

Solar panels could either be body-mounted to the CubeSat or attached to the deployment mechanism. The CubeSat marketplace, Pumpkin Inc., and EnduroSat offer COTS solar panels for CubeSats (Table 1.5). PV cells will be purchased instead of COTS solar panels and connected to the EPS to minimize project costs.

Table 1.5. COTS solar panels acquired from [43–45].

Product	Cost
CubeSat Solar panel DHV-CS-10 (1U)	$1,613.53
EXA DMSA – Deployable Multifunction Solar Array (1U – 6U)	$4,896.23 - $22,255.60
Solar Panel for AntS antenna (1U)	$4,730
CubeSat Kit™ Fixed Solar Panels (1U – 3U)	$2,500 – $5,650
CubeSat Kit™ Fixed Solar Panel: End Panel (1U)	$2,500
1U Solar Panel	$2,800 - $3,600
1.5U Solar Panel	$3,500 - $4,100
3U Solar Panel	$6,800 - $6,900
3U Deployable Solar Array	$21,000 - $35,000
6U Solar Panel	$17,100
6U Deployable Solar Array	$51,300
8U Solar Panel	$21,300 - $64,000

1.4.5 Power Performance Test

The following solar array configurations will be tested for each CubeSat unit:

- **Body-mounted:** The PV cells will be attached to the walls of the CubeSat.
- **Single-hinge deployable:** The PV cells will be attached to a single-hinge deployment mechanism.
- **Double-hinge deployable:** The PV cells will be attached to a two-stage multi-hinge deployment mechanism. Figure 1.8 shows the deployment sequence for the primary and secondary hinges.
- **Single-hinge deployable ACS:** Sun sensors will be implemented to provide attitude control for the single-hinge deployable configuration.
- **Double-hinge deployable ACS:** Sun sensors will be implemented to provide attitude control for the double-hinge deployable configuration.

The power generated by the PV cells and power usage by the EPS will be recorded across each test. Power losses are expected to come from the motors, sun sensors, MCU and other electrical components, which will draw power during the deployment and attitude control of the solar array. Testing will be conducted within an indoor and outdoor test environment. To ensure there is no variability between each test, the following actions will be taken:

- Testing will start and end at consistently set times. For example, all tests will be conducted between 11 AM – 3 PM and the duration of each test is the same.
- Testing will be conducted on days with similar ambient conditions. Since temperature affects the performance of PV cells, it is ideal to conduct tests on days with similar ambient conditions; ambient conditions will be recorded.
- Testing will undergo several trials. All outliers will be recorded but excluded from the analysis.

- Testing will be conducted under uniform weather conditions. For example, testing will only be conducted on sunny days. One test on a sunny day and another on a rainy day will not occur.
- A location will be selected so that the testing platform is consistently illuminated. For example, all solar panels will be initially placed in an area where they are illuminated. Additionally, throughout the day, shadows must not obscure the solar panel’s access to sunlight. Once a suitable location has been determined, all tests will occur there.
- Tests will be recorded by video to determine any failure points or anomalies.

2. SADM System Architecture

2.1 Structural Architecture

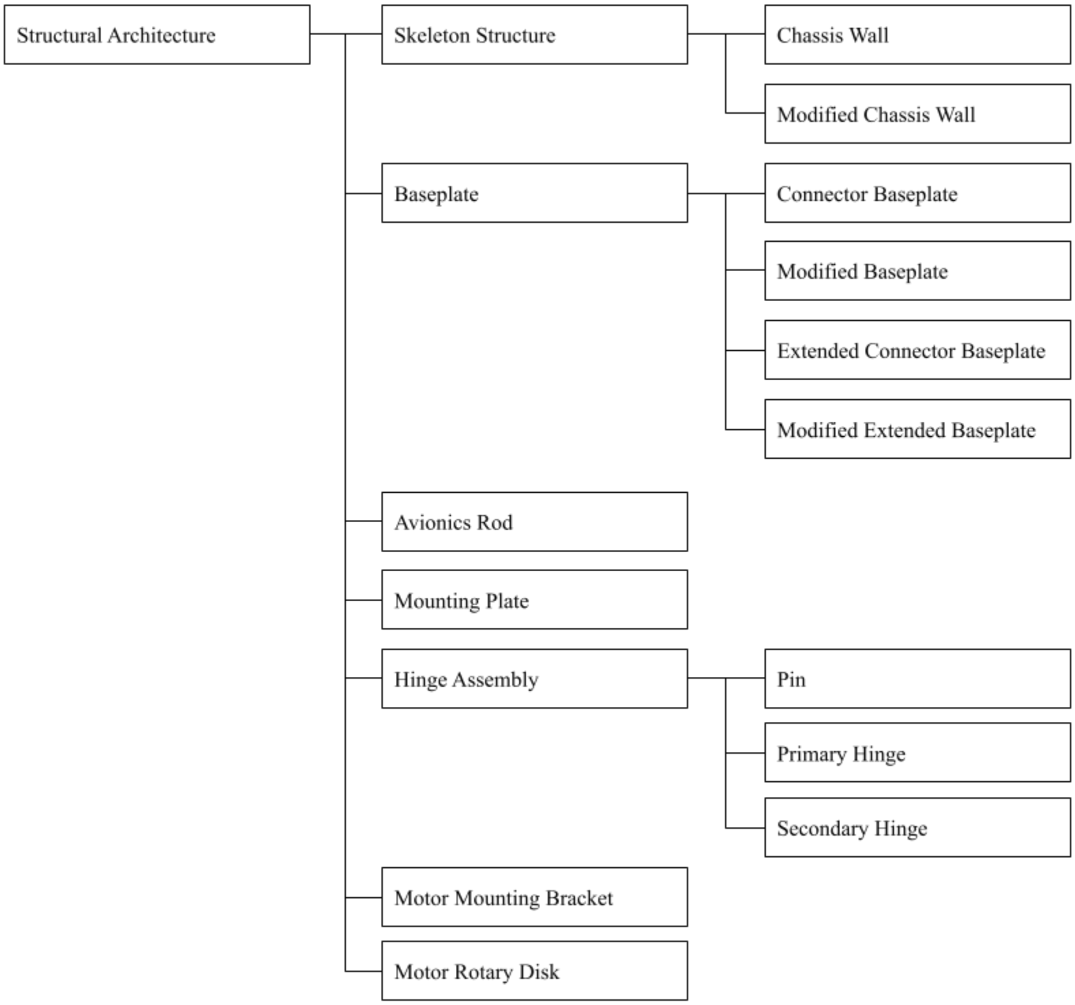


Figure 2.1. SADM structural architecture.

Structural components:

- **Chassis wall:** Provides structural support and allows the attachment of solar panels (Figure 2.8). The standard variation has dimensions of 100 x 100 mm (Figure C.9), whereas the modified variation has dimensions of 100 x 92 mm (Figure C.10). The reduced size of the modified variation is essential to ensure that each chassis wall could be secured to the baseplate via M2 screws.
- **Baseplate:** This component is used for the attachment of the chassis walls; two are used at the bottom and top of a CubeSat (Figure 2.3). An extended baseplate is used for 4U and 6U CubeSats (Figure 2.4).

    - **Connector baseplate:** Similar to the baseplate, this component is used to connect two different CubeSat units (Figure 2.5). An extended variation is used for 4U and 6U CubeSats (Figure 2.6).
- **Avionics rod:** A cylindrical 6.35 mm rod is directly inserted into the mounting pads to allow the direct installation of mounting plates and PCBs.
- **Mounting plates:** Allows the attachment of the rotary disk system and other instruments (Figure 2.7).
- **Hinge assembly:** Composed of a pin, secondary hinge, and primary hinge. The hinge is attached to the motor's rotary disk. When the fishing string retention mechanism is cut, a set of torsion springs deploy each solar array.
- **Motor mounting bracket:** Secures the NEMA 11 motor to the baseplate.
- **Motor rotary disk:** Attached to the motor, this disk allows each solar array to rotate about the axis that's normal to each chassis wall.

2.1.1 CubeSat Skeleton

The CubeSat skeleton was created based on the design specifications outlined in [11], such that the skeleton was limited to a volume of 100 mm^3 and a mass of 1.33 kg. The design objectives of the skeleton were to ensure that its components were interchangeable, and that installation was accessible. The two designs that were considered involved a component slide-based and slotted screw-based installation approach. In the former, grooves would be present on each of the launch rails. During installation, the chassis walls would be slid into place via these grooves. In the latter design, the chassis walls would be screwed onto a baseplate; the ends of these walls are a part of the launch rail. The selected design was the screw-based design since it appeared to be the most straightforward approach. This is because, with the former design, the location to attach the baseplate and the chassis walls without compromising space for the solar panels was unknown. The CubeSat screw-based skeleton design for 1U–6U CubeSats is provided in Figure 2.2.

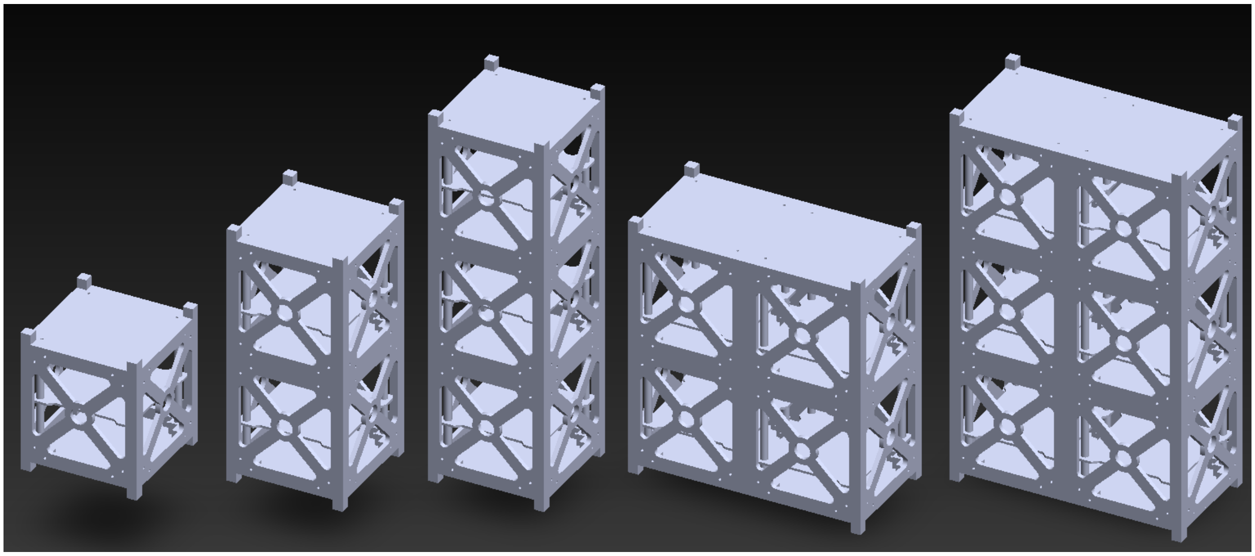


Figure 2.2. The CubeSat screw-based skeleton design for 1U, 2U, 3U, 4U, and 6U CubeSats.

2.1.2 Solar Panel Selection

The design of the CubeSat skeleton is dependent on the dimensions of the selected solar panel. The initial primary drivers during solar panel selection involved low-cost solar panels and their cell area. The efficiency of the solar panel was not a primary driver, because an objective of this research project was to determine the power output difference between each solar array configuration. Appendix B features a list of solar panel specifications from amazon.com, voltaicsystems.com, and digikey.com; the provided dimensions include the cell area and backplate. The solar panels below 65 mm and those exceeding 100 mm were discarded from the consolidated solar panel list (Table 2.1). From this consolidation, the operational environment was the final driving factor in selecting the solar panel. Based on the specified primary drivers, the 313070005 solar panel was considered as the initial prime candidate because it provided the most cell coverage. However, considering that a 6U CubeSat requires 22 solar panels, the total cost would be $271 (Table 2.2). The Sunnytech GP80*80-10A100 model was initially selected due to its lower total cost of $154 (Table 2.2). As all necessary electrical components and equipment were finalized, it became necessary to reduce project costs. Thus, selecting low-cost solar panels in bulk was the sole primary driver. The SUNYIMA A3D053 was selected because ten solar panels could be purchased for $15.99; for a 6U solar panel, the cost would be $44.97 (Table 2.2). Specifications of the A3D053 solar panel are listed in Table 2.3.

Table 2.1. Consolidated solar panel list.

Component Name	Cost ($)	L	W	H	Operational Environment
Sunnytech GP80*80-10A100	6.99	80.00	80.00	UNK	Outdoor
SC20036	17.49	83.82	83.82	5.08	Outdoor
LL200-2.4-75	4.53	88.90	73.70	UNK	Indoor
MPT4.8-75	10.41	94.00	72.90	0.22	Outdoor
313070005	12.3	100.00	80.00	UNK	Outdoor

Table 2.2. Total solar panel costs for each CubeSat.

		Total Cost ($)		
CubeSat Size (U)	Solar Panels	313070005	GP80*80-10A100	A3D053
-------	1	12.3	6.99	14.49
1	6	73.8	41.94	14.49
2	10	123	69.90	14.49
3	14	172.2	97.86	29.98
4	16	196.8	111.84	29.98
6	22	270.6	153.78	44.97

Table 2.3. A3D053 specifications [46]. *Mass was measured during the assembly process.

A3D053	
Material	Polycrystalline Silicon
Max Current (mA)	30
Max Voltage (V)	5
Max Power (W)	0.15
*Mass (g)	6.0
Area (m^2)	0.00159

2.1.3 Baseplate

The baseplate (Figure 2.3) was designed to support the attachment of screwable chassis walls via the use of M2 screws (10 mm). The external rails have equidistant dimensions of 8.55 mm. The baseplate has mounting holes to support the installation of a 6.35mm diameter avionics rod. For the 4U and 6U CubeSats, it was necessary to extend the baseplate. The baseplate and extended baseplate share similar design characteristics, except that the central mounting pads for the extended baseplate were joined together (Figure 2.4).

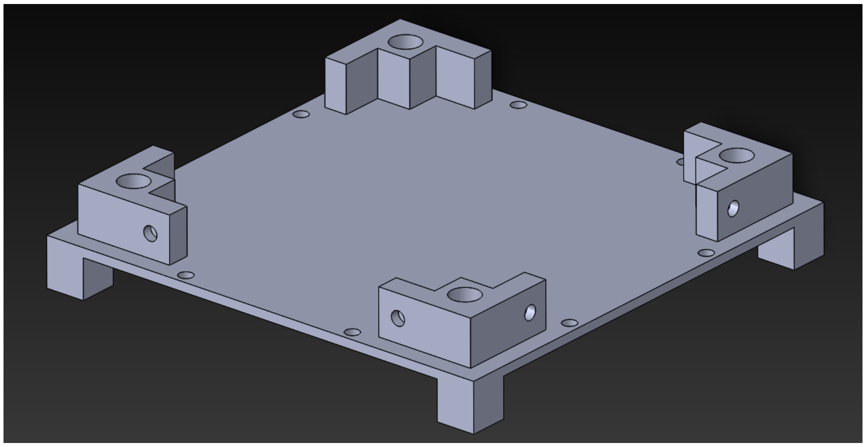


Figure 2.3. The baseplate supports the attachment of screwable chassis walls.

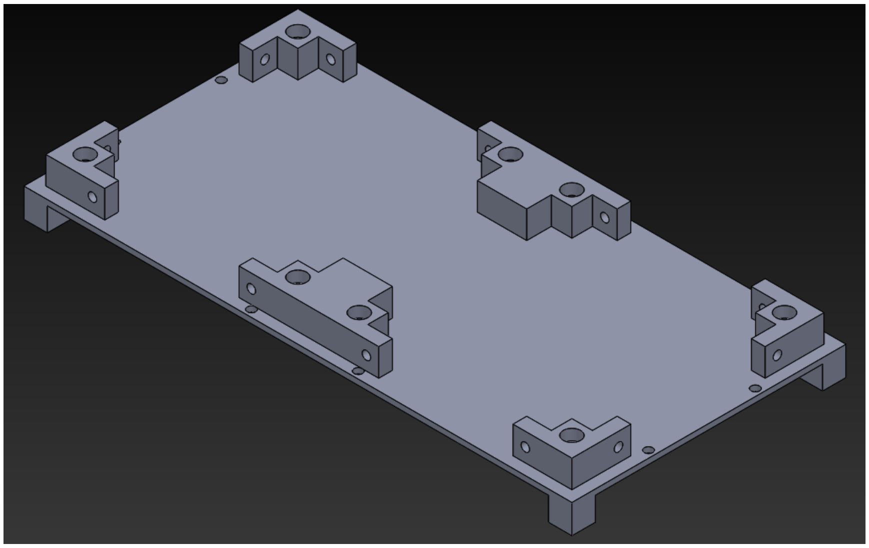


Figure 2.4. The extended baseplate supports the attachment of chassis walls for 4U and 6U.

2.1.4 Connector Baseplate

To support the vertical attachment of another CubeSat unit, the baseplate was modified as shown in Figure 2.5. The difference between the initial and modified baseplate is that the modified baseplate has no external rails, and it has mounting pads on both sides. The connector baseplate has an extended variation for 4U and 6U CubeSats (Figure 2.6). The difference between these designs is that the extended variation has one set of shifted mounting pads.

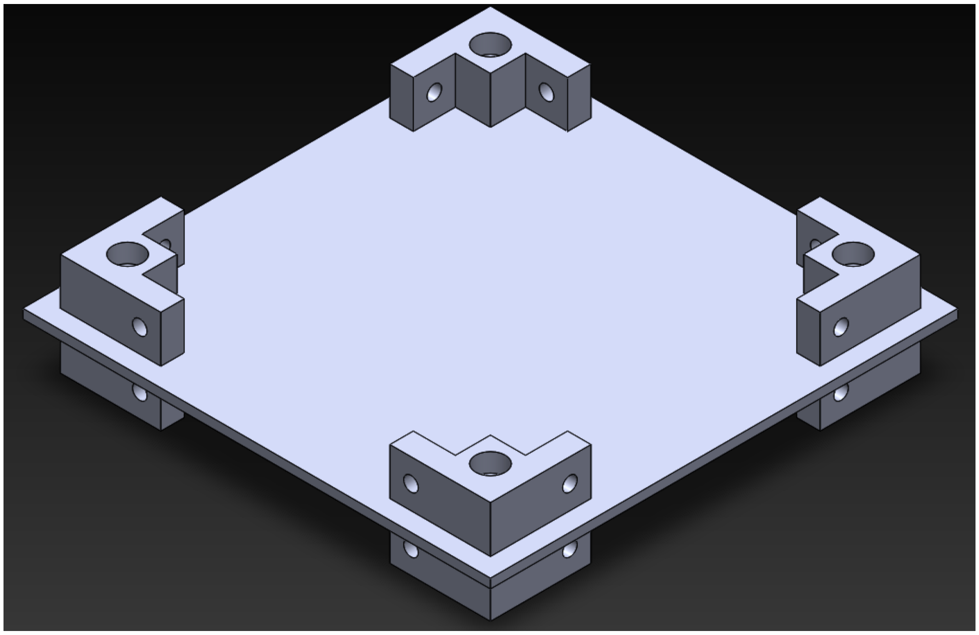


Figure 2.5. The connector baseplate supports the vertical attachment of another CubeSat unit.

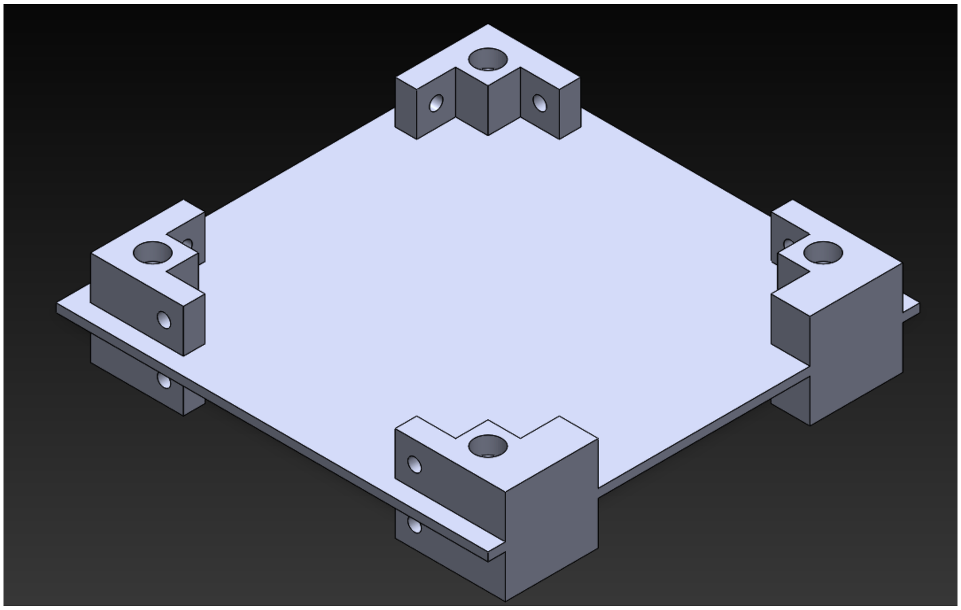


Figure 2.6. The extended connector baseplate is used to attach chassis walls for 4U and 6U.

2.1.5 Mounting Plate

The mounting plate allows the attachment of the rotary disk system and other electronics, such as a breadboard. These plates are secured in place via 1/4” x 5/8” x 3/8” spacers and 1/4” hex nuts; 6.35 mm x 15.875 mm x 9.525 mm and 0.25 mm, respectively.

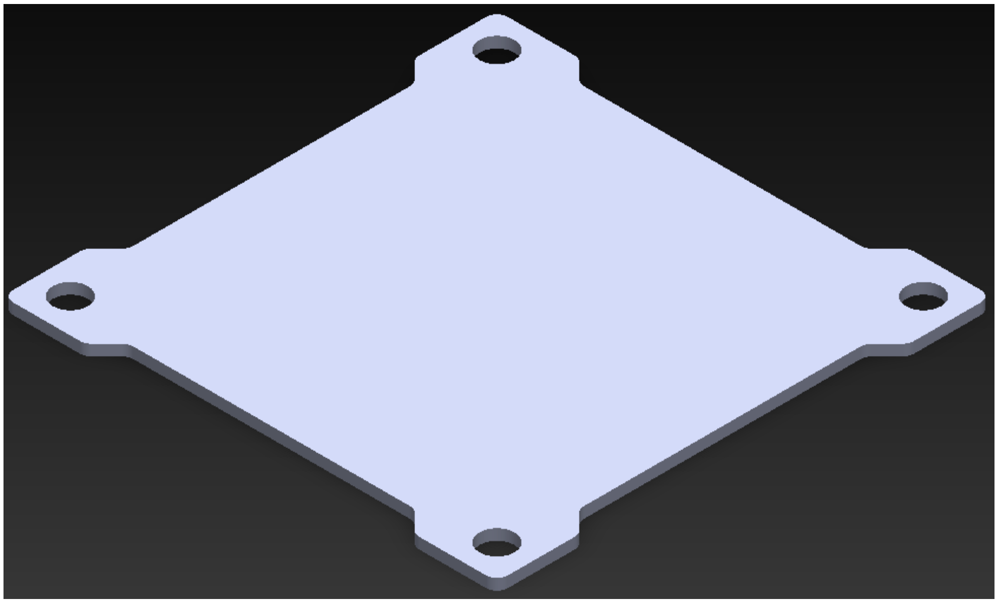


Figure 2.7. The mounting plate supports the attachment of various components.

2.1.6 Chassis Wall

The chassis walls are secured to the baseplate via M2 x 10 mm screws and were originally designed to support the attachment of a 80 x 80 mm solar panel. Since the ends of the chassis walls are a part of the launch rails, a 100 x 92 mm chassis wall variation was created. The internal structure of both chassis wall designs shares the same dimensions. These chassis walls are used in the 1U to 6U CubeSats.

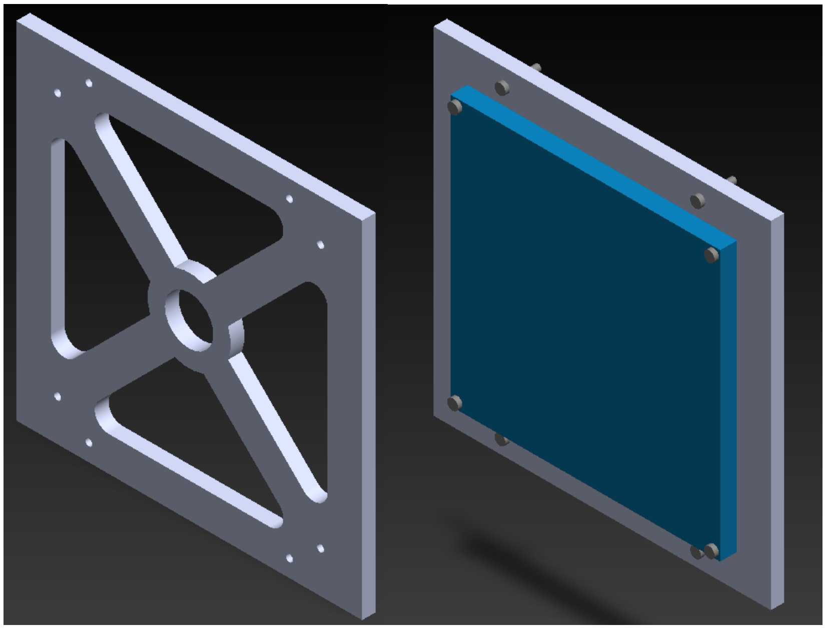


Figure 2.8. The chassis wall supports the installation of a 80 x 80 mm solar panel.

2.1.7 Deployable Solar Array

In the body-mounted solar array configuration, the solar panels were attached directly to a solar panel backplate retained in the stowed configuration (Figure 5.10). The solar panel is deployed by a burn-wire system, which incorporates a low-resistance resistor. As will be later discussed, the single-set and double-set torsion spring systems successfully deployed the solar array. Under Earth's gravity, however, the torsion springs did not have sufficient strength to fully keep the solar arrays deployed. During early phases of project development, attachable skeletonized baseplates were developed to keep a 80 x 80 mm solar panel fully deployed (Figure 2.9). Multiple baseplates could be attached for double-hinge and multi-hinge deployments.

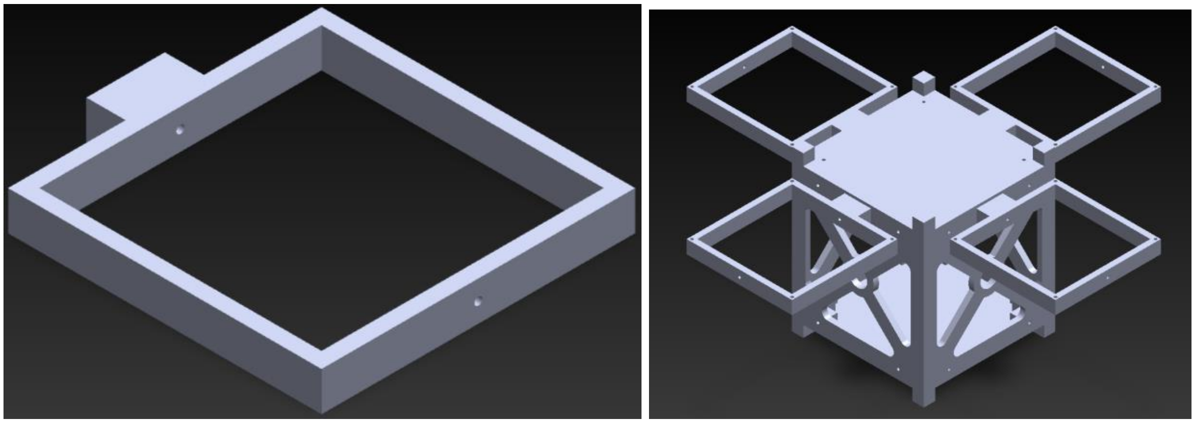


Figure 2.9. Skeleton baseplate (left) and deployed solar array state (right).

An alternative prototype design involved a linkage-based deployment mechanism (Figure 2.10). In this design, the solar panels would be attached together via a system of linkages. To deploy the solar array, a block driven by an electrical current or motor will be slid along a rail. As the block slides across this rail, the block will be pressing against the linkages, which causes the linkages to extend.

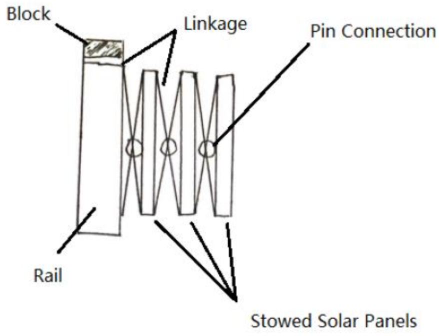


Figure 2.10. Linkage deployment mechanism. The space between the solar panels is enlarged for visual purposes.

The prototype designs were not used for a few reasons. The skeleton baseplate was not used because its dimensions were not compatible with the size of the A3D053 solar panel. The linkage-based deployment mechanism was not developed as it would introduce unnecessary complexity, costs, mass, and EMI issues. For example, this design would require the block to be magnetized and the rails to be electrified; the presence of the magnetic field would interfere with the onboard sensors. Instead of using the prototype designs, the solar array backplate was attached directly to the baseplate of the CubeSat (Figure 2.11).

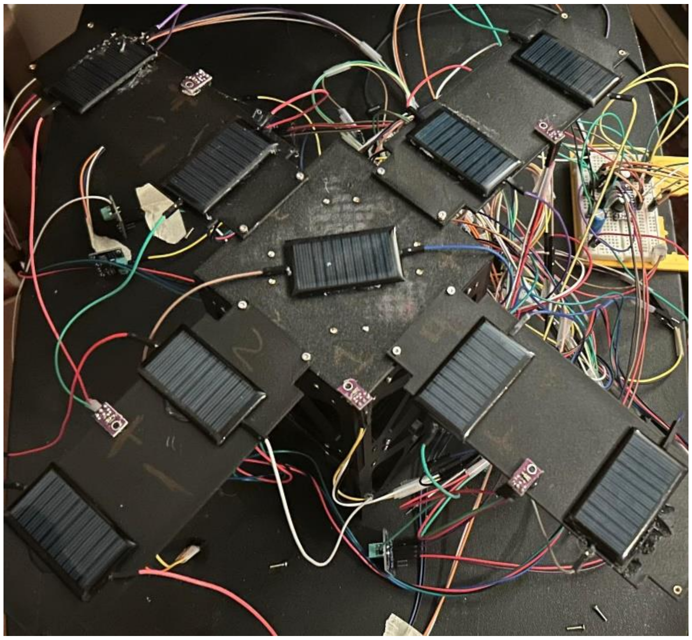


Figure 2.11. Fully deployed solar arrays.

2.2 Electrical Architecture

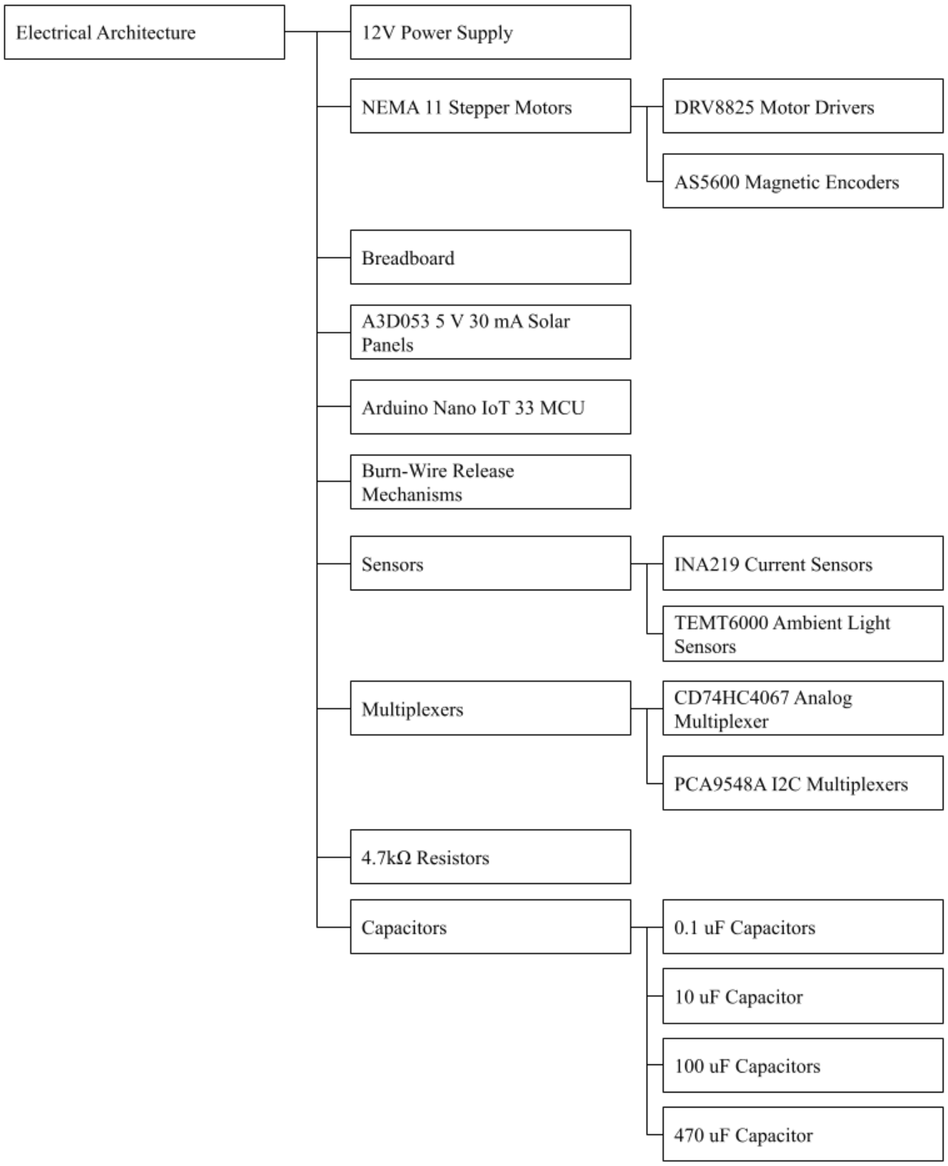


Figure 2.12. SADM electrical architecture.

2.2.1 Stepper Motor Configuration

To provide two-axis controllability, a collection of stepper motors is necessary to drive a system of rotary disks and hinges. The amount of necessary motors depends on the motor-rotary disk configuration and the hinge configuration. The motors can drive the rotary disks via two configurations: non-redundancy and redundancy. In the first configuration, one motor has the capability of driving two rotary disks that intersect on the same plane. However, if the motor were to fail, two solar arrays would fail to deploy. Whereas in the second configuration, only one motor drives one's rotary disk; thus, failure would be limited to one solar array. As previously described in 1.4.3, the rotary disk rotates the solar arrays about the axis parallel to the CubeSat's surface. To provide rotation about the axis that is parallel to the CubeSat's surface, a motor must be attached to each hinge. The solar arrays could utilize a single, double, and multi-hinge design. The amount of hinges and motors required for each hinge configuration is specified in Table 2.4.

Table 2.4. Amount of motors that are necessary depend on the rotary-disk redundancy configuration and hinge configuration.

Hinge Configuration	Amount of Hinges	Motors
Single	4	6 – 8
Double	8	10 – 12
Multi (3)	12	14 - 16

2.2.2 Stepper Motor Selection

The driving factors of the stepper motor selection process involved its cost, size, and whether the motor had an encoder. Selecting a motor with an encoder is an important consideration because the encoder provides information about the motor shaft's angular position. These elements are important in creating a closed-loop attitude control system. Based on the stepper motor table (Table A.1), the lowest-priced motor with an integrated encoder is the HT17-268D-FBA NEMA 17 motor with a cost of $102. Table 2.5 provides the costs of the motor based on the redundancy and hinge configuration. The configuration involving the deployment of one solar array with one solar array provides the lowest-cost configuration to fulfill the objectives of the project. In this configuration, project costs would vary from $204 to $408 depending on the hinge configuration. Whereas within the extreme case, the deployment of four solar arrays with full redundancy would result in project costs ranging from $816 to $1020.

Table 2.5. Motor configuration costs based on redundancy and hinge configuration. Note that one motor controls initial deployment (surface motor) and the other rotates the disk (RD motor). The DH and MH configurations adds one and two additional motors, respectively.

	Hinge Configuration		
Motor Configuration	SH Cost ($)	DH Cost ($)	MH (3) Cost ($)
1 Surface, 1 RD	204	306	408
2 Surfaces, 1 RD	306	408	510
2 Surfaces, 2 RDs	408	510	612
3 Surfaces, 2 RDs	510	612	714
3 Surfaces, 3 RDs	612	714	816
4 Surfaces, 2 RDs	612	714	816
4 Surfaces, 4 RDs	816	918	1020

Generally, stepper motors rotate by a fixed amount based on the number of steps. For example, a stepper motor with 200 steps offers 1.8° of precision between each step. Therefore, the angular position of the rotary disks and solar arrays could be determined at any given time. During operation, there is a risk that a step could be skipped under heavy loading. However, considering the solar arrays have a low mass, loading on the motor shaft will be minimal. Initially, a $14 NEMA 17 Bipolar motor was selected because of its lower cost and availability compared to the NEMA 14 and NEMA 11 motors. However, $18 NEMA 11 motors were purchased once they were restocked due to their lower size. Specifications of the NEMA 11 and NEMA 17 motor are listed in Table 2.6.

Table 2.6. NEMA 11 (28HD1411-02) and NEMA 17 (17HS19-2004S) bipolar stepper motor specifications [47,48]. *Obtained from multiplying amps/phase with resistance/phase.

	NEMA 11	NEMA 17
Step Angle (°)	1.8	
Amps/Phase (A)	0.8	2
Rated Voltage (V)	3.84	*2.8
Holding Torque (mNm)	65	59,000
Resistance/Phase (Ω)	4.8 ± 10%	1.40 ± 10%
Inductance/Phase (mH)	2.8 ± 20%	3.00 ± 20%
Step Accuracy (Non-Accum)	-------	± 5.00%
OTR (°C)	-------	-10 to 50
Rotor Inertia (g cm^2)	7.5	82
Motor Mass (g)	80	-------

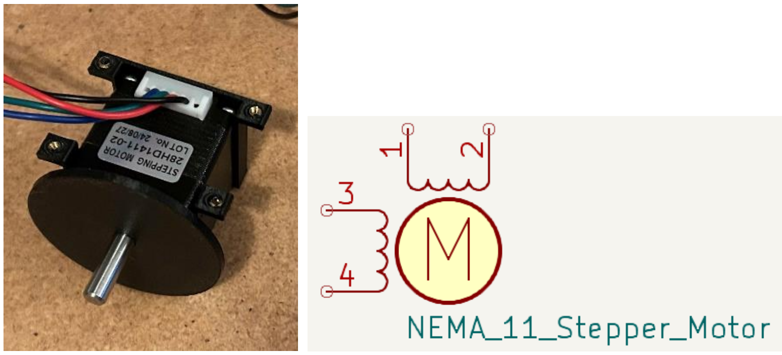


Figure 2.13. NEMA 11 motor with 3D printed attachments (left) and simple wiring diagram (right).

2.2.3 Stepper Motor Driver Selection

For a stepper motor to fully operate, a motor driver is necessary. The motor driver is used to adjust the voltage, current, and speed, and allow microstepping of the stepper motor. The downsides of microstepping is that the torque output for each step decreases and that heat generation increases. To maximize torque output of the motors, microstepping was not used.

The motor drivers that were considered involved the A4988, DRV8825, and TMC2208 drivers. The primary factors during the selection of a motor driver involved operating Amps and cost. Based on the specifications of the NEMA 17 motor, the motor driver should deliver 2A to the motor. This current supply requirement is fulfilled by the DRV8825 motor driver. The TMC2208 enables quiet stepper motor operations, but it does not meet the current supply requirement.

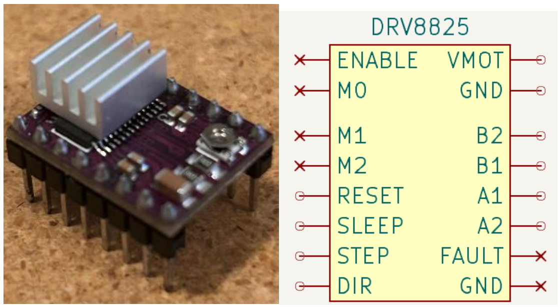


Figure 2.14. DDRV8825 motor driver (left) and simple wiring diagram (right).

Table 2.7. A4988, DRV8825, and TMC2208 motor driver specifications [49–51].

Motor Driver	Motor Supply Voltage (V)	Output Current (A)	Peak Current (A)	Average Cost ($)
A4988	8 - 35	1.2	2	8.32
DRV8825	8.2 - 45	1.5	2.5	10.49
TMC2208	4.75 - 36	1.4	2	18.44

2.2.4 Motor Encoder Selection

Although the possibility of the stepper motor skipping steps is unlikely, available motor encoders were examined. The types of encoders that were examined involved magnetic, optical, and mechanical encoders. Optical encoders were not considered to be optimal due to the low-light environment within the CubeSat, high costs, and the risk of outgassing effects. The AS5600 magnetic encoder was selected over the rotary encoder due to its greater step precision.

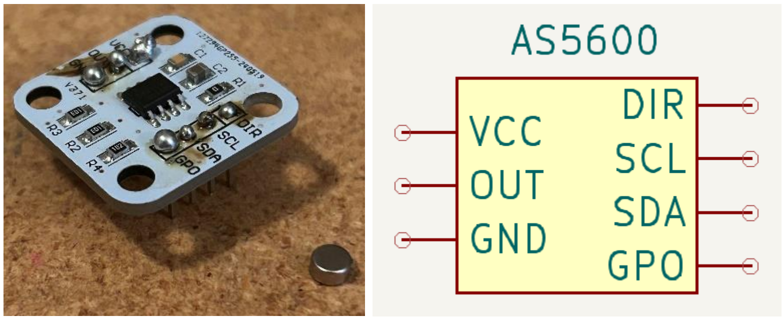


Figure 2.15. AS5600 magnetic encoder with magnet (left) and simple wiring diagram (right).

2.2.5 Microcontroller Unit Selection

The MCU is responsible for driving the motors by regulating the voltage sent to these motors. The MCU that was selected to fulfill this purpose is the Arduino Nano 33 IoT. This component has been selected due to its low-cost, small form factor, and that it has historically been used on previous CubeSat missions without suffering from radiation failure.

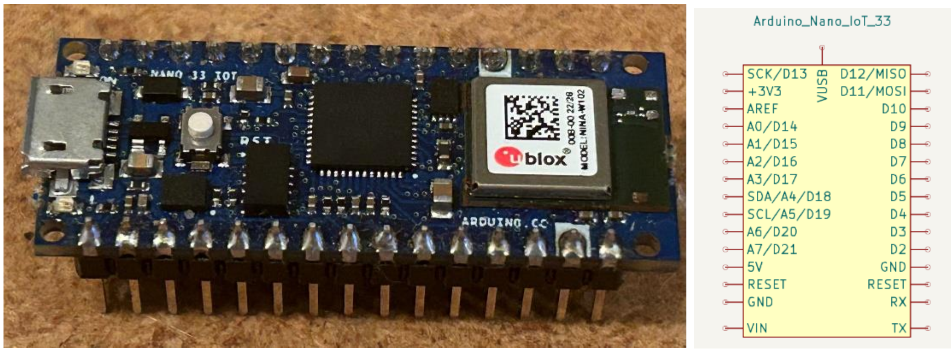


Figure 2.16. Arduino Nano IoT 33 MCU (left) and simple wiring diagram (right).

2.2.6 Sun Sensor Selection

Sun sensors are instruments that are used to determine the direction and position of the sun. This is achieved when photosensitive cells determine the angle of incidence of incoming light. The TEMT6000 ambient light sensor has been temporarily selected to serve as the sun sensor due to its low cost, 60° angle of half sensitivity, and compatibility with the Arduino MCU; specifications are provided in provided in Table 2.8. Once the prototype has been completed, a sun sensor that can be directly surfaced onto the PCB will be selected.

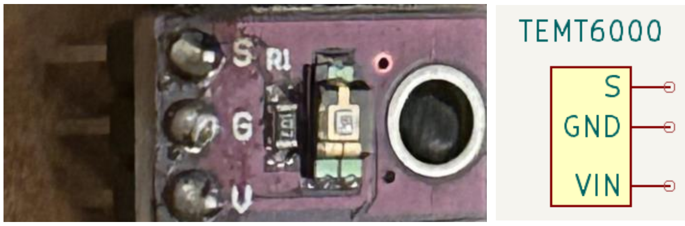


Figure 2.17. TEMT6000 light sensor with breakout board (left) and simple wiring diagram (right).

Table 2.8. TEMT6000 light sensor specifications [52].

Sensor Type	Analog
Working Voltage (V)	3.3 – 5.5 V
Illumination Range (Lux)	10 – 1000
OTR (°C)	-40 to 100
Wavelength of Peak Sensitivity	570
Range of Spectral Bandwidth	440 – 800

2.2.7 Current Sensor Selection

Voltage and current sensors are required to measure the respective quantities. To simplify the wiring process, a sensor that both measures voltage and current was selected. The two candidates considered were the INA219 and INA226 current sensors. The sole driving factor when selecting the current sensor was cost. The prices of the INA219 and INA226 were $7 and $38 for ten units, respectively; the INA219 was selected based on cost (Figure 2.21).

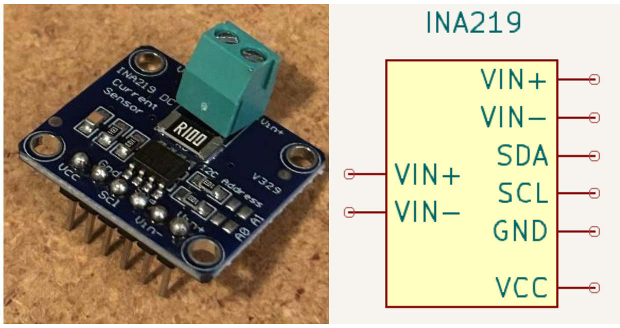


Figure 2.18. INA219 current sensor (left) and simple wiring diagram (right).

By examining the datasheets of the INA219 and INA226, and applying the following equations, Table 2.9 was formed. The INA219 current sensor has a I_{LSB} of 0.78 mA while the while the max current of the A3D053 solar panel is 30 mA; the INA219 sensor has sufficient resolution to measure current changes.

$$V_{LSB} = \frac{SVR}{2^N} \qquad 2.1$$

$$I_{LSB} = \frac{V_{LSB}}{R_{shunt}} \qquad 2.2$$

Table 2.9. Specifications of the INA219 [53] and INA226 [54] current sensors. *Based on available COTS breakout boards. †Calculated from the shunt resistor and ADC resolution value.

	INA219	INA226
Bus Voltage Range (V)	0 - 26	0 – 36
Operating Voltage (V)	3 – 5.5	2.7 – 5.5
N	12	16
OTR (°C)	-40 to 125	
SVR (mV)	±320	±81.92
*Shunt Resistor	100mΩ	
†Maximum Current (A)	3.2	0.819
V_{LSB} (mV)	0.078125	0.00125
† I_{LSB} (mA)	0.78125	0.0125

2.2.8 Digital Multiplexer Selection

The INA219 current sensor uses the SDA and SCL pins for data transmission and data synchronization via the I^2C protocol. The INA219 datasheet specifies that 16 INA219 current sensors could be connected to the SDA and SCL port of the MCU. Practically, however, there are not enough tie points on the breadboard to achieve this; a I^2C multiplexer is required. The two multiplexer boards that were examined are the PCA9548A and the TCA9548A. Aside from the newer processes of the former and the operating voltages of 2.3V – 5.5V [55] and 1.65V – 5.5V [56], respectively, there does not appear to be a substantial difference between these two multiplexers. The PCA9548A was purchased due to its lower cost of $10 compared to the $12 cost of the TCA9548A. Five INA219 current sensors were connected to the first multiplexer (0x70 address) and four were connected to the second multiplexer (0x71 address). The relevant connections are viewable in Figure D.2. The reset pins must be connected to the 3.3V rail for continuous operation. Unlike the DRV8825, the digital pins (A0-A2) must not be left floating as there are no internal resistors. The address pins are set by connecting the A0-A2 pins in a particular configuration. Functionally, the first multiplexer sequentially reads through channels SC0/SD0 to SC4/SD4 to read the current and voltage of each current sensor. This process is repeated for the second multiplexer. As will be later discussed in the results, this process takes 3.76 seconds to loop back to the initial current sensor. This time delay is inconsequential as a fast response is not required. 4.7kΩ resistors were placed between the SCL and SDA lines (Figure D.2) to the 3.3V rails to ensure that these lines remain at the 3.3V level when not actively pulled low; this maintains appropriate voltage levels for I^2C communication. During component-level testing, the INA219 sensors were initially undetected. However, when these resistors were connected, the INA219 sensors became fully functional.

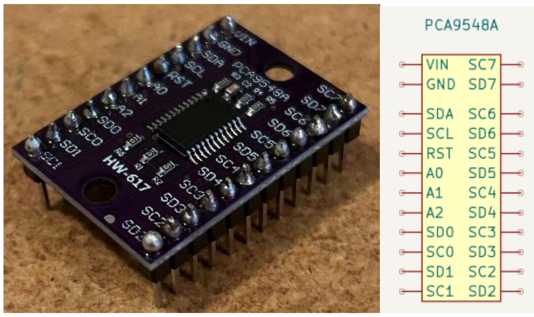


Figure 2.19. PCA9548A digital multiplexer (left) and simple wiring diagram (right).

2.2.9 Analog Multiplexer Selection

The CD74HC4067 was the only available analog Arduino multiplexer board. Although not strictly necessary—there are enough analog ports for the five light sensors—this multiplexer was purchased to allow additional light sensors to be integrated for redundancy purposes. The EN pin must be connected to GND for continuous operation. A particular channel is selected by setting the S0-S3 digital pins to low or high (Table 2.10). The unused digital pins must not be left floating as there are no internal pull resistors. Since there are only five light sensors, the S3 pin was connected to GND to conserve the amount of available digital pins; eight channels are still available.

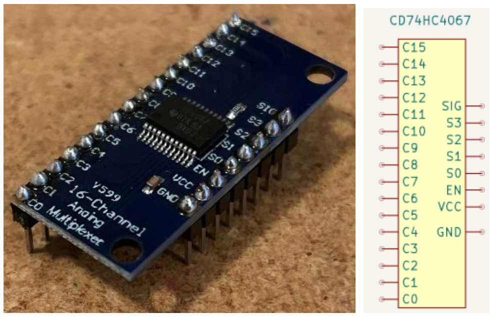


Figure 2.20. CD74HC4067 analog multiplexer (left) and simple wiring diagram (right).

Table 2.10. Channel selection guide [57].

Channel	S0	S1	S2	S3
C0	Low	Low	Low	Low
C1	High	Low	Low	Low
C2	Low	High	Low	Low
C3	Low	Low	High	Low
C4	High	High	Low	Low
C5	High	Low	High	Low
C6	Low	High	High	Low
C7	High	High	High	Low
C8	Low	Low	Low	High
C9	High	Low	Low	High
C10	Low	High	Low	High
C11	High	High	Low	High
C12	Low	Low	High	High
C13	High	Low	High	High
C14	Low	High	High	High
C15	High	High	High	High

2.2.10 Electrical Power System

The EPS is responsible for supplying power to the MCU, stepper motors, motor drivers, and the light sensors within the SADM. A breadboard will be used for prototyping purposes, and past the prototyping phase, the circuit architecture will be integrated into a PCB. A PCB EPS is advantageous over a breadboard EPS since the risk of a wire disconnecting is eliminated. Secondly, a PCB yields a more compact EPS, which is ideal due to the limited volume within a CubeSat. Figure 2.16 and Figure 2.17 show the schematics of the MCU and light sensors, respectively. For a light sensor to operate, the V_{In} pin must be connected to a 3.3V lane provided by the MCU, the GND pin must be connected to a GND lane, and the S pin must be connected to one of the analog pins of the MCU. Connecting all light sensors to the MCU accordingly yields Figure 2.21; the 3.3V lane originates from a voltage lane provided by the motor drivers. For each wiring diagram, red wires, back wires, magenta, and green wires indicate positive voltage, ground, digital signals, and analog signals, respectively.

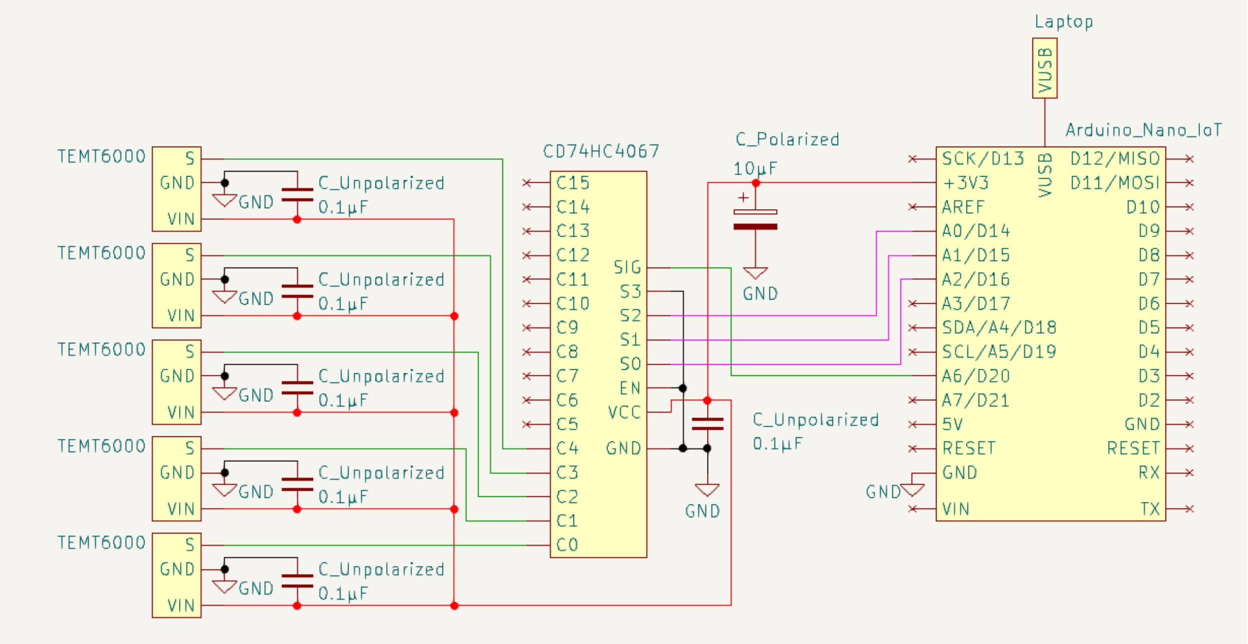


Figure 2.21. Light sensors, analog multiplexer, and MCU simple wiring diagram.

Figure 2.13 and Figure 2.14 show the simple wiring diagram of the stepper motor and motor driver, respectively. For the stepper motor to operate, each pair of phases must be connected to the corresponding A and B pin pairs. For example, pins 3 and 4 would be connected to A1 and A2. For the motor driver to function, an external power supply must provide voltage to the V_{MOT} and GND pins; a 12V power supply has been selected. Next, the STEP and DIR pins must be connected to two available digital MCU pins. The STEP and DIR pins relay the stepper motor's current step and rotation direction. A jumper connects the RESET and SLEEP pins because they will not be used. Lastly, GND is connected to GND and FAULT is connected to the 5V MCU pin to provide power. Figure 2.22 shows the resulting component connections.

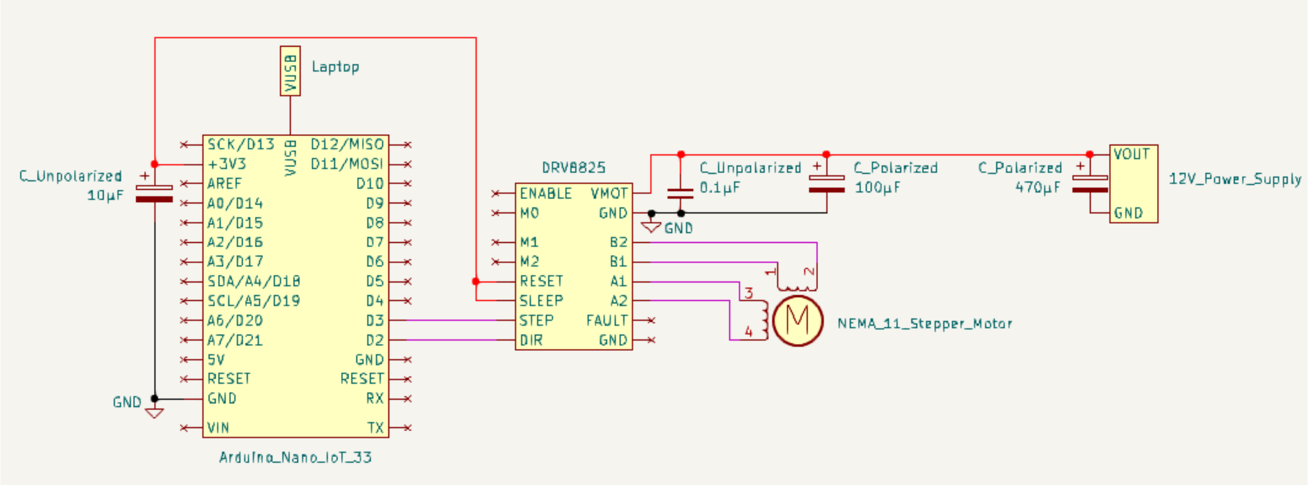


Figure 2.22. Stepper motor, motor driver, PSU, and MCU simple wiring diagram.

Figure D.1 shows the EPS system with four attached motors. However, such a configuration is not possible for a 1U CubeSat based on the motors that have been selected. As shown by Figure 4.13, only two NEMA 17 motors could fit within a 1U CubeSat.

2.3 Software Architecture

A PID control algorithm will be programmed within the Arduino IDE software. By incorporating data from the sun sensors and motors, the control system could deploy the solar array and provide attitude control. Once this program has been created, the program will be uploaded to the MCU via a USB connection. Figure D.1 shows that the EPS can support a maximum of six light sensors and seven motors based on project design objectives. Four motors are used to control the deployment of the SH solar array and two are used to control the rotary disks; the third motor is not used due to increased complexity. Thus, the current EPS cannot support additional light sensors and motors for the DH and MH configurations. This issue can be resolved by utilizing the MOSI and MISO pins, an additional MCU, or using a larger MCU such as the Arduino Mega 2560.

2.3.1 Light Sensor Placement

Two methods were considered for the deployment of the solar arrays for a CubeSat. In the first case, the solar arrays would fully deploy regardless of light-intensity values from the light sensors; the light sensors are attached to the ends of each solar array. Upon full deployment, only then does the PID control system reorient the solar arrays based on the light-intensity values. In the second case, the primary set of solar arrays deploy based on current light-intensity values; the primary light sensors are attached to the CubeSat. For the double-hinge configuration, the secondary set of light sensors is attached to each end of the primary solar arrays. The control system then reorients the secondary solar array based on light-intensity values from the secondary light sensors. The multi-hinge deployment process is similar to the double-hinge deployment process. Regarding the second case, the outermost set of light sensors could no longer be attached to the outermost solar arrays for the DH and MH configurations. This is because when the solar arrays are folded, the light sensors become obscured. Subsequently, these light sensors could either be attached to the top or bottom end of the CubeSat. If the light sensors were placed at the top, the light-sensors FOV would be partially obstructed when the solar panels are fully deployed. Secondly, there is a possibility that the reflected irradiance from the solar panels would interfere with the measurements of the light sensors. For these reasons, the ideal placement was considered to be at the bottom end of the CubeSat (Figure 2.23).

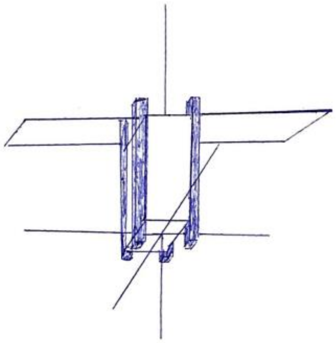


Figure 2.23. Approximate light sensor placement for double-hinge and multi-hinge configurations. The vertical and horizontal lines indicates the light sensors FOV direction.

Comparatively, the first method is more power inefficient but could potentially offer more attitude control precision. In this method, all of the light sensors are attached directly to the solar arrays rather than a set being attached to the CubeSat structure. Subsequently, the light-intensity value profile from end-to-end could be measured. However, considering that the outermost light sensor is located only 100 mm away from the outermost edge of the solar array, it is probable that any precision or power increase is negligible. For these reasons, the second deployment method will be used.

2.3.1.1 Single-Hinge

After a CubeSat has been deployed from a CSD, the CubeSat will have an unknown arbitrary orientation; for CubeSats that incorporate an ADCS, the initial orientation can be determined from a collection of accelerometers, gyroscopes, and magnetometers. To determine the sun's location, a collection of six light sensors is used; one light sensor is placed at the end of each solar array. The PID control algorithm will determine which light sensor has the greatest light intensity value and rotate each solar array accordingly. Although the sun's orientation could be determined based on a minimum of two light sensors, the use of six light sensors ensures that the algorithm does not mistake the Earth for the Sun due to reflected irradiation. Secondly, the additional sensors provide redundancy in the event a light sensor fails.

2.3.1.2 Double-Hinge and Multi-Hinge

Placing a light sensor at the end of each solar array is still possible in the double-hinge configuration as long as there are design changes to the solar array's backplate. As shown in Figure 2.24, the secondary backplate has its edges trimmed. Thus, the light sensor could be attached to the top or bottom edges of the backplate. Although a light sensor could be placed on the back surface of the backplate of each solar panel, such a placement invalidates the purpose of the light sensor within the PID algorithm. Upon deployment, the solar cells and light sensor would be antiparallel; attitude control would be reversed for what is intended. The trim design, however, will not be used due to project volumetric design constraints. This is why the first set of light sensors is attached to the CubeSat structure. The light sensor placement for the MH configuration is similar to the previous configuration.

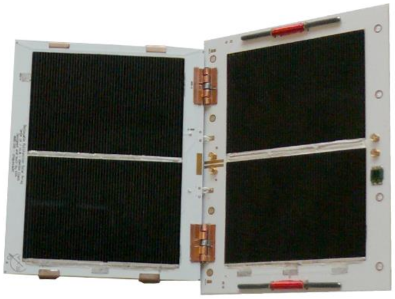


Figure 2.24. Double-hinge solar panel [43].

2.3.2 Control System Logic

When the CubeSat is first deployed in LEO, the first illumination case is where the sun only irradiates one surface of the CubeSat. Figure 2.25 and Figure 2.26 shows the four cases and which motors would be activated based on the surface with the most illumination. For clarity on the convention used, Figure 2.25 has been provided. Figure 2.28 - Figure 2.30 shows the control system logic when only one surface is illuminated. Initially, the CubeSat deploys the solar arrays from each surface of the CubeSat. The MCU then reads the sensor data of each light sensor. As will be later discussed in Chapter five, there is substantial noise in the TEMT6000 ADC value reading. After taking 50 sensor samples, the noise was reduced by removing the extremum values and taking the average of the trimmed data set. From there, the control algorithm determines which light sensor has the maximum ADC value. The appropriate motors are activated, and the new ADC value are compared with the previous one. If the new ADC value is greater than the previous, the motor will continue rotating. If otherwise, the motor will go back one step and will hold the array in that position. Lastly if the top surface has the maximum irradiation value, no action is necessary because it is assumed that the solar arrays are deployed to face the top surface.

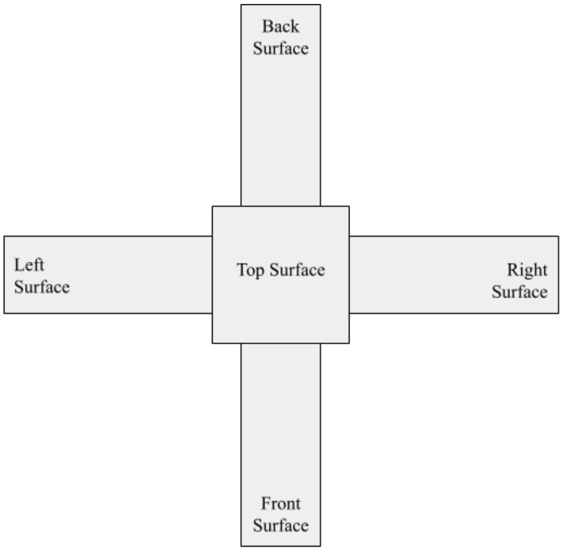


Figure 2.25. Surface classifications of the CubeSat.

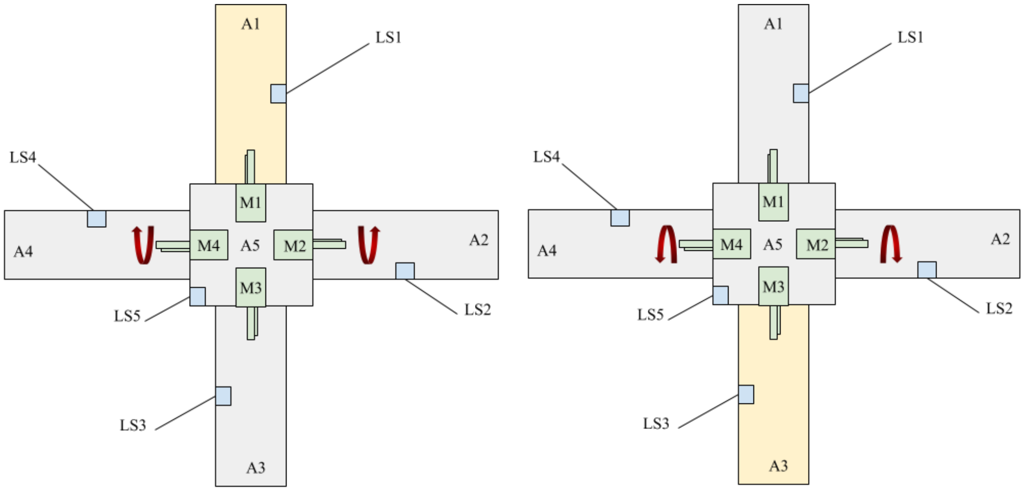


Figure 2.26. M2 rotates CCW and M4 rotates CW when the back surface has maximum illumination (left). M2 rotates CW and M4 rotates CCW when the front surface has maximum illumination (right).

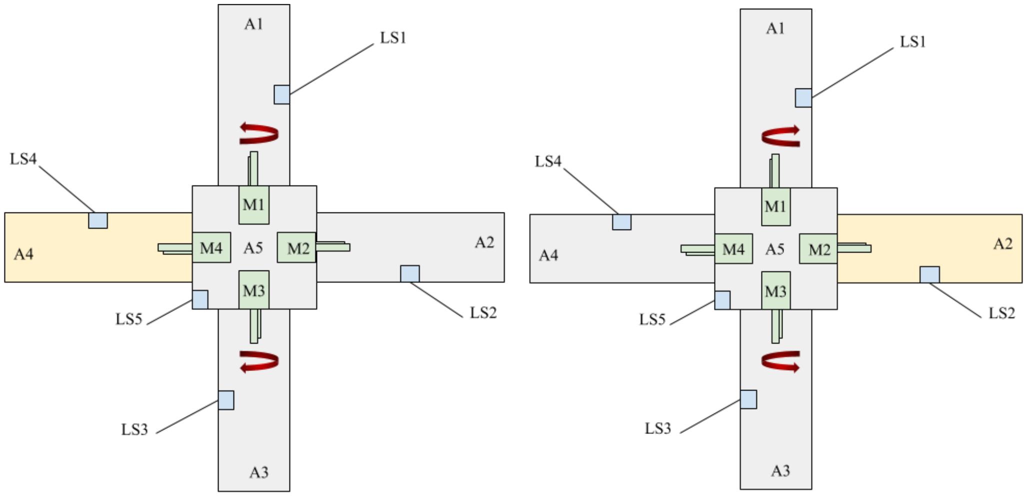


Figure 2.27. M1 rotates CW and M3 rotates CCW when the left surface has maximum illumination (left). M1 rotates CCW and M3 rotates CW when the right surface has maximum illumination (right).

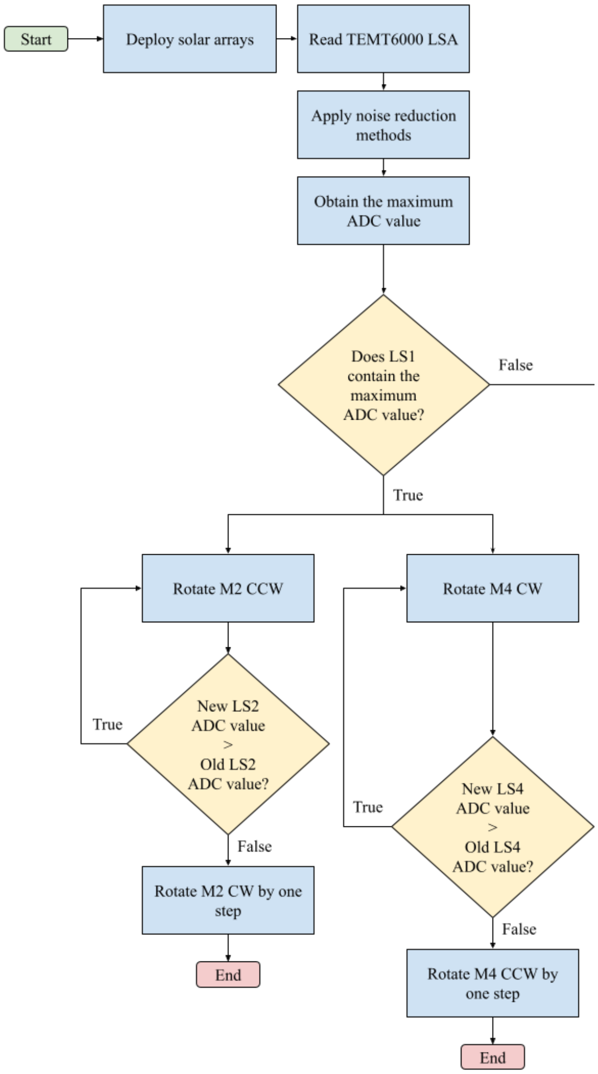

Start
Deploy solar arrays
Read TEMT6000 LSA
Apply noise reduction methods
Obtain the maximum ADC value
Does LS1 contain the maximum ADC value?
False
True
Rotate M2 CCW
Rotate M4 CW
New LS2 ADC value > Old LS2 ADC value?
True
False
Rotate M2 CW by one step
End
New LS4 ADC value > Old LS4 ADC value?
True
False
Rotate M4 CCW by one step
End

Figure 2.28. Control system logic involving solar panel deployment, sensor readings, and the first set of motors.

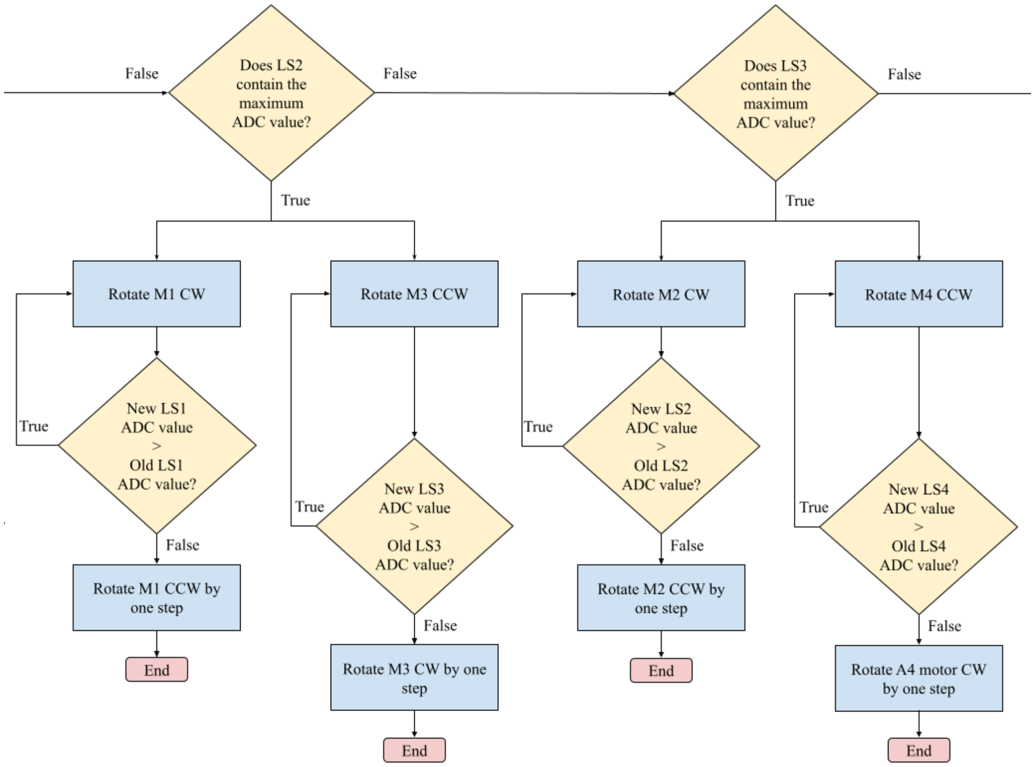

False
Does LS2 contain the maximum ADC value?
False
Does LS3 contain the maximum ADC value?
False
True
True
Rotate M1 CW
Rotate M3 CCW
Rotate M2 CW
Rotate M4 CCW
True
New LS1 ADC value > Old LS1 ADC value?
True
New LS3 ADC value > Old LS3 ADC value?
True
New LS2 ADC value > Old LS2 ADC value?
True
New LS4 ADC value > Old LS4 ADC value?
False
Rotate M1 CCW by one step
End
False
Rotate M3 CW by one step
End
False
Rotate M2 CCW by one step
End
False
Rotate A4 motor CW by one step
End

Figure 2.29. Control system logic involving two additional sets of motors.

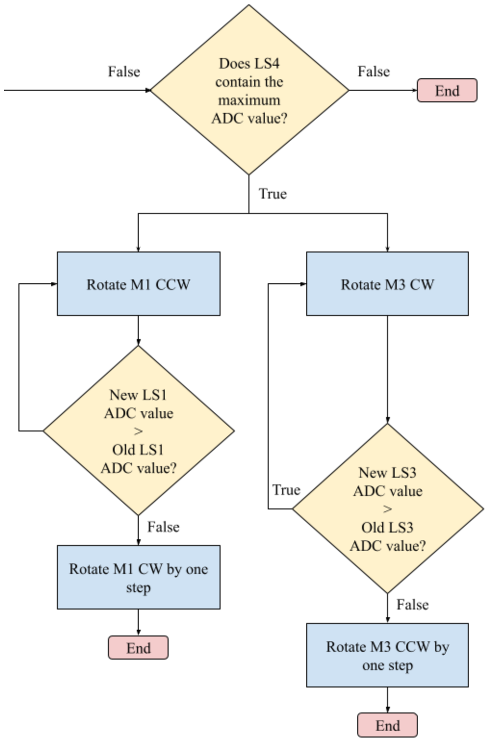

False
Does LS4 contain the maximum ADC value?
False
End
True
Rotate M1 CCW
Rotate M3 CW
New LS1 ADC value > Old LS1 ADC value?
True
New LS3 ADC value > Old LS3 ADC value?
False
Rotate M1 CW by one step
End
False
Rotate M3 CCW by one step
End

Figure 2.30. Control system logic involving the final set of motors.

Although the algorithm successfully rotates each array to the appropriate orientation, an efficiency issue arises when two surfaces have similar ADC values. For example, if surface one has an ADC value of 1024 and surface two has an ADC value of 1023, surface two would still be rotated. This causes unnecessary power loss when the motor is actuated. Furthermore, the previous algorithm would not rotate surface three to face the second irradiated surface. Figure 2.31 and Figure 2.32 shows the cases of two irradiated surfaces with similar ADC values. The modified algorithm (Figure 2.33 - Figure 2.35) considers when two surfaces have similar irradiation values. After solar panel deployment and noise reduction methods, the algorithm checks whether any two surfaces have similar ADC values. If this is not true, the previous control algorithm is applied. However, if true, the surfaces are split into pairs and the average ADC values of each pair is taken. These ADC values are then compared to determine which pair experiences the greatest irradiation. Figure 2.31 and Figure 2.32 shows which motors would be activated based on this new control algorithm. As will be discussed in Chapter three, a CubeSat experiences maximum irradiance when three solar panels are irradiated at 45° angles. However, because solar array control is only possible with respect to the normal axis of each CubeSat surface, rotation with respect to the top surface is not possible. Subsequently, the control algorithm remains unchanged.

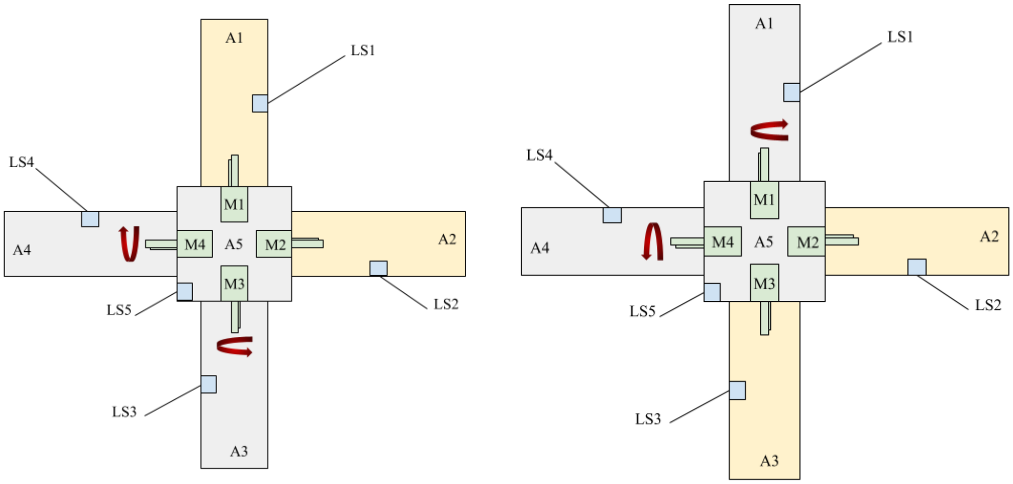


Figure 2.31. When the combined back and right surfaces have maximum illumination (left), M3 rotates CCW and M4 rotates CW. When the combined right and front surfaces have maximum illumination (right), M1 rotates CW and M4 rotates CCW.

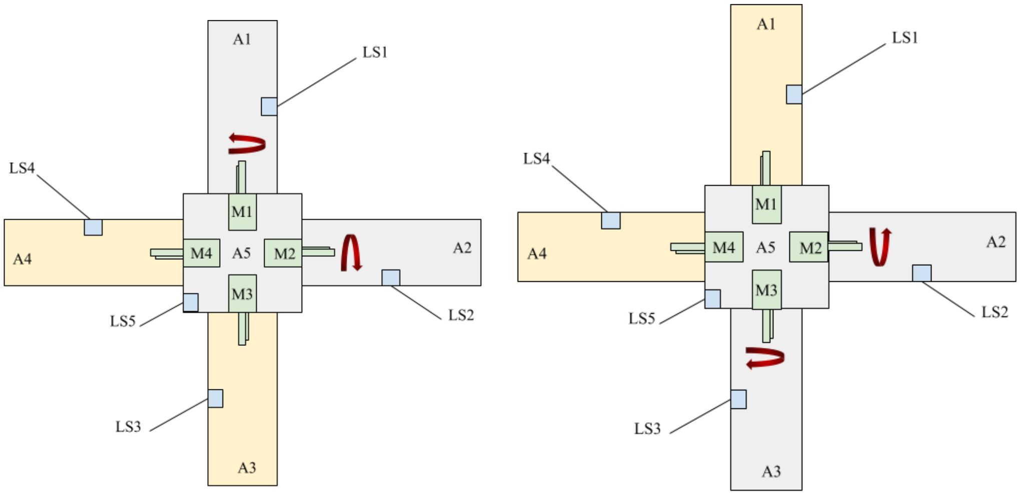


Figure 2.32. When the combined front and left surfaces have maximum illumination (left), M1 rotates CCW and M2 rotates CW. When the combined left and back surfaces have maximum illumination (right), M2 rotates CCW and M3 rotates CW.

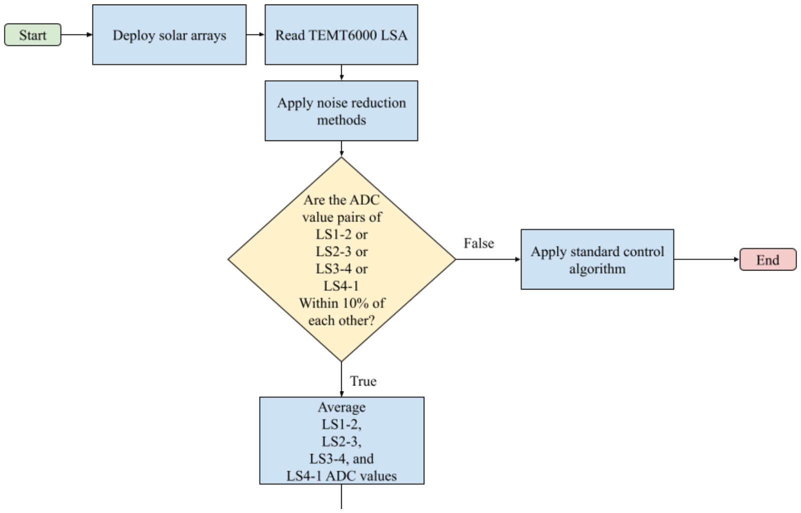

Start
Deploy solar arrays
Read TEMT6000 LSA
Apply noise reduction methods
Are the ADC value pairs of LS1-2 or LS2-3 or LS3-4 or LS4-1 Within 10% of each other?
False
Apply standard control algorithm
End
True
Average LS1-2, LS2-3, LS3-4, and LS4-1 ADC values

Figure 2.33. Control system logic involving the deployment of the solar arrays, sensor readings, application of the noise reduction method and determining whether two surfaces have similar irradiation values.

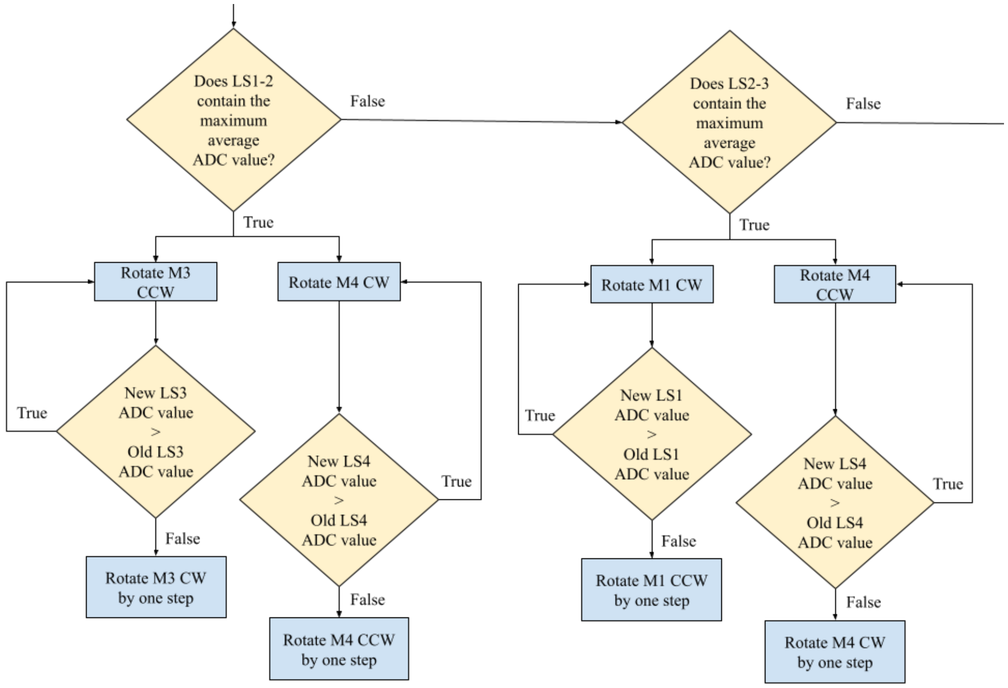

Does LS1-2 contain the maximum average ADC value?
False
Does LS2-3 contain the maximum average ADC value?
False
True
True
Rotate M3 CCW
Rotate M4 CW
Rotate M1 CW
Rotate M4 CCW
New LS3 ADC value > Old LS3 ADC value
True
False
Rotate M3 CW by one step
New LS4 ADC value > Old LS4 ADC value
True
False
Rotate M4 CCW by one step
New LS1 ADC value > Old LS1 ADC value
True
False
Rotate M1 CCW by one step
New LS4 ADC value > Old LS4 ADC value
True
False
Rotate M4 CW by one step

Figure 2.34. Control system logic of the motors for the back-right and right-front irradiated surface pairs.

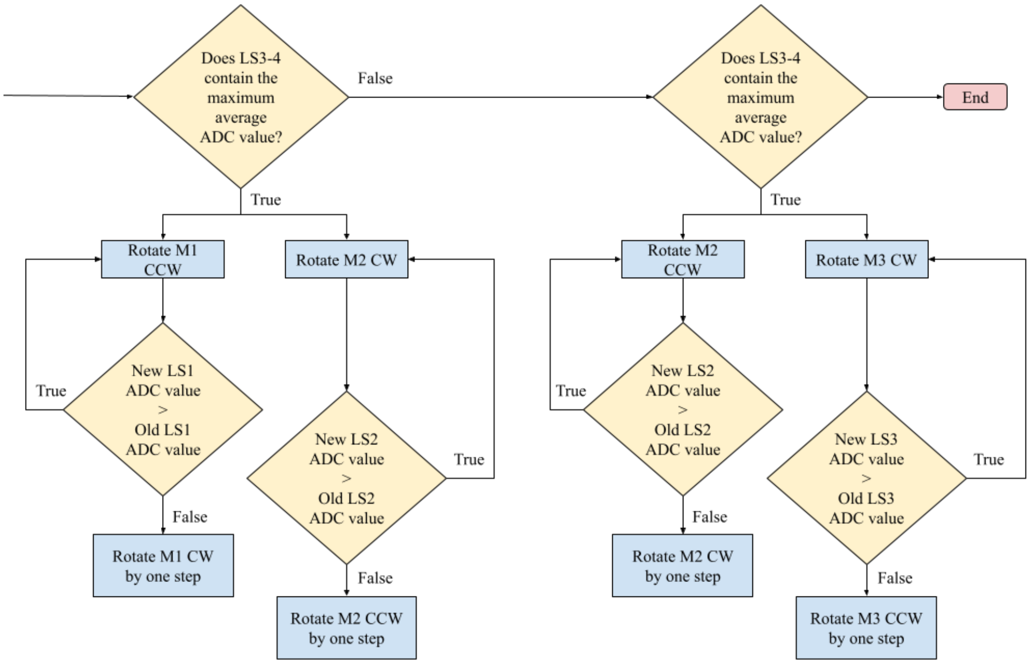

Does LS3-4 contain the maximum average ADC value?
False
Does LS3-4 contain the maximum average ADC value?
End
True
True
Rotate M1 CCW
Rotate M2 CW
Rotate M2 CCW
Rotate M3 CW
New LS1 ADC value > Old LS1 ADC value
True
False
Rotate M1 CW by one step
New LS2 ADC value > Old LS2 ADC value
True
False
Rotate M2 CCW by one step
New LS2 ADC value > Old LS2 ADC value
True
False
Rotate M2 CW by one step
New LS3 ADC value > Old LS3 ADC value
True
False
Rotate M3 CCW by one step

Figure 2.35. Control system logic of the motors involving the motors for the front-left and left-back irradiated surface pairs.

3. Theoretical Solar Array Power Output

3.1 Solar Panel Irradiance and Temperature

Solar panels generate electricity when their PV cells are exposed to sunlight. The amount that a solar panel could generate depends on the panel's size, efficiency, solar irradiance, cell material, cell temperature, humidity, and wind. Nominally, the wattage that a solar panel could output is based on STC. STC specifies that with a cell temperature of 25°C, a solar irradiance of 1000 W m^{-2}, and an air mass of 1.5, the solar panel is rated to output a particular wattage. Practically, solar panels do not operate at STC. By considering the factors that influence the power capabilities of a solar panel, its current power output could be determined. The power output of a solar panel is specified by equation 3.1 [57].

$$\mathrm{P = G \times A \times \eta} \tag{3.1}$$

Considering that solar irradiance is dependent on the solar incident angle (θ), the effective solar irradiance equation is formed.

$$\mathrm{G_{eff} = G \times \cos(\theta)} \tag{3.2}$$

Thus, by substituting G with $\mathrm{G_{eff}}$, equation 3.1 becomes the following equation.

$$\mathrm{P = G \times \cos(\theta) \times A \times \eta} \tag{3.3}$$

Manufacturers specify that for each 1°C change in temperature, the efficiency of their solar panel decreases by a percentage-based amount. [57] specifies the efficiency change of a solar panel by the following equation.

$$\varepsilon = \lambda(\mathrm{T_{cell}} - 25)\eta \tag{3.4}$$

By subtracting equation 3.4 from η in equation 3.3 and simplifying, the dependency of solar panel performance due to effective solar irradiance and temperature changes is listed by the following equation. It is important to note that the following equation considers power losses due to temperature as a linear relationship for a small temperature range. At extreme cell temperatures, the relationship becomes nonlinear, and the power output does not go negative.

$$\mathrm{P = G_{eff} \times A \times \eta\,[1 - \lambda \times (T_{cell} - 25)]} \tag{3.5}$$

3.2 Theoretical CubeSat Power Performance

As previously discussed, solar panels could be mounted on a CubeSat via the body-mounted, deployable, and sun-tracking deployable configurations. The theoretical power that each solar array could generate based on equation 3.5 will be shown throughout the following sections.

3.2.1 Body-Mounted Solar Array

3.2.1.1 Optimal Incidence Angle

From the previous section, the effective solar irradiance is dependent on the solar incidence angle, which is given by equation 3.2. For any CubeSat size that utilizes the solar array body-mounted configuration, the maximum number of faces that could be directly irradiated from the sun is three. Considering a solar panel with a cell area of 1 m^2, solar irradiance of 1000 W m^{-2}, and efficiency of 0.15, the maximum effective irradiance and subsequent power output of a solar panel based on the amount of irradiated faces is listed in Table 3.1. Table 3.1 was formulated based on equations 3.2 and 3.3. Figures 3.1 - 3.3 show the total effective irradiance and power generated by a solar panel per degree change in angle of incidence. It is important to observe that Figure 3.2 has two axes of the incidence angle because the additional axis is necessary to capture what occurs within the system. Initially, it is assumed that only one solar panel is irradiated; this indicates that the incidence angles present within the system are 0° and 90°. Past this initial condition, the incidence angles of both solar panels converge to 45° based on an angular change of 1°. For the case of three irradiated solar panels, the same previous assumption is made; the incidence angles of all solar panels converge to 45°. The results shown in Table 3.1 and Figures 3.1 - 3.3 were obtained based on the assumption that the CubeSat does not receive reflected solar irradiation from Earth's atmosphere and other bodies. These results show that body-mounted solar panels generate the maximum amount of power when three panels are irradiated at an incidence angle of 45°.

Table 3.1. Total maximum effective irradiance and maximum power output of one or multiple solar panels for a 1U CubeSat.

Irradiated Faces	Optimal Incidence Angle (°)	Maximum Effective Irradiance (W m^{-2})	MPO (W)
1	0	1000	150
2	45	1414.21	212.13
3	45	2121.32	318.20

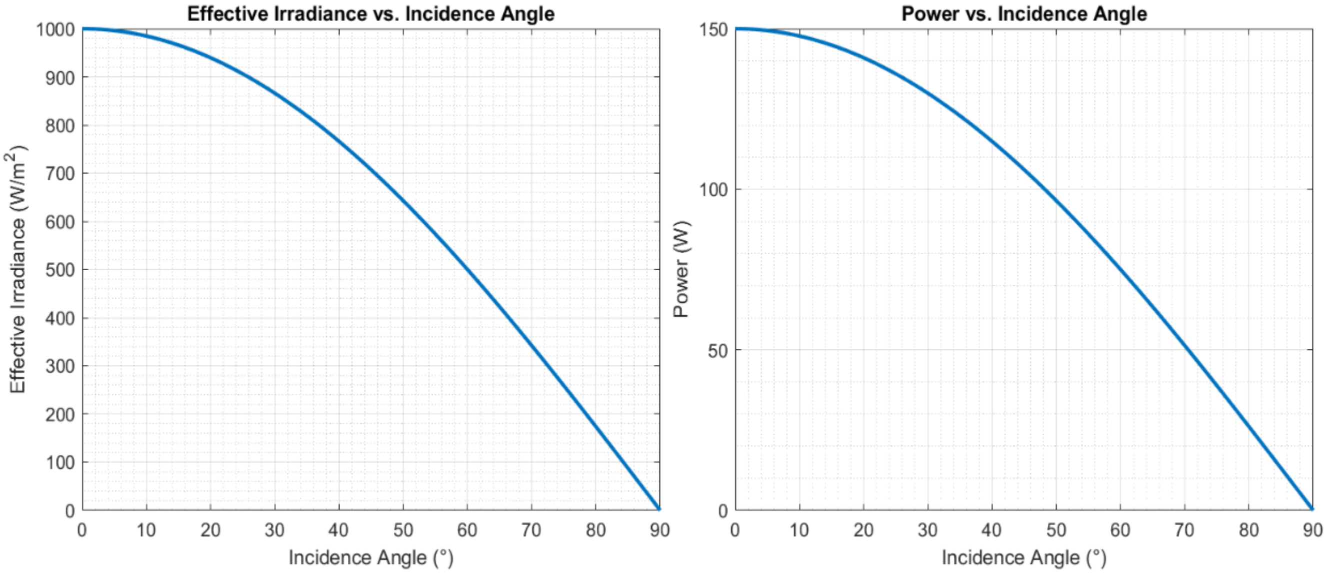


Figure 3.1. Effective irradiance and power generated from one irradiated solar panel.

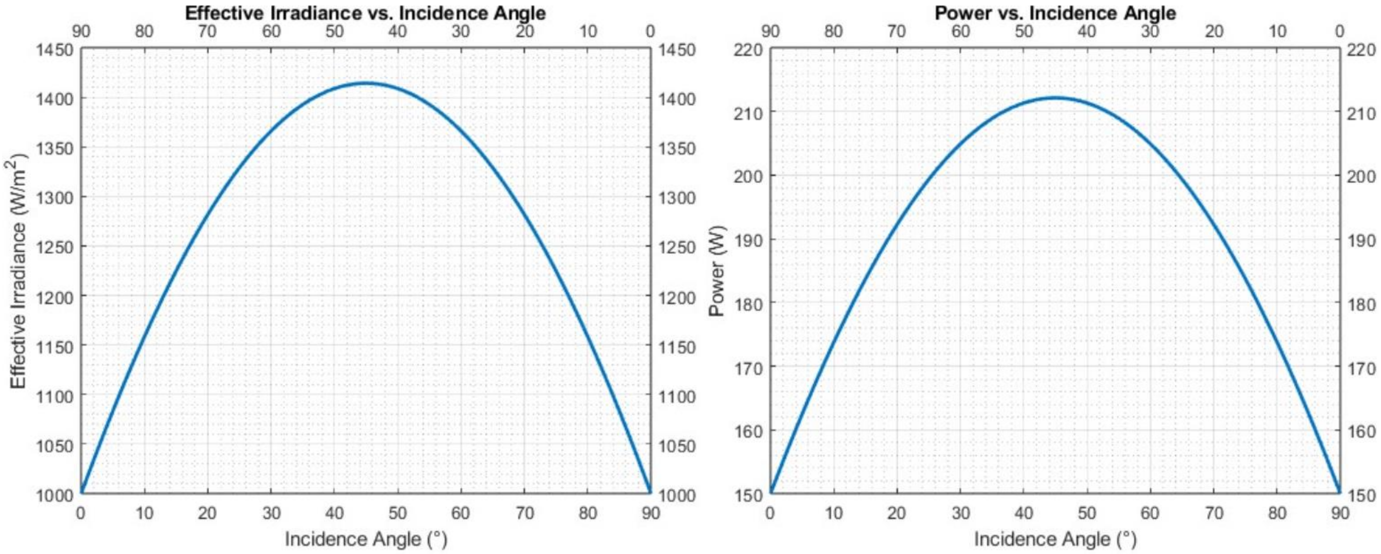


Figure 3.2. Effective irradiance and power generated from two irradiated solar panels.

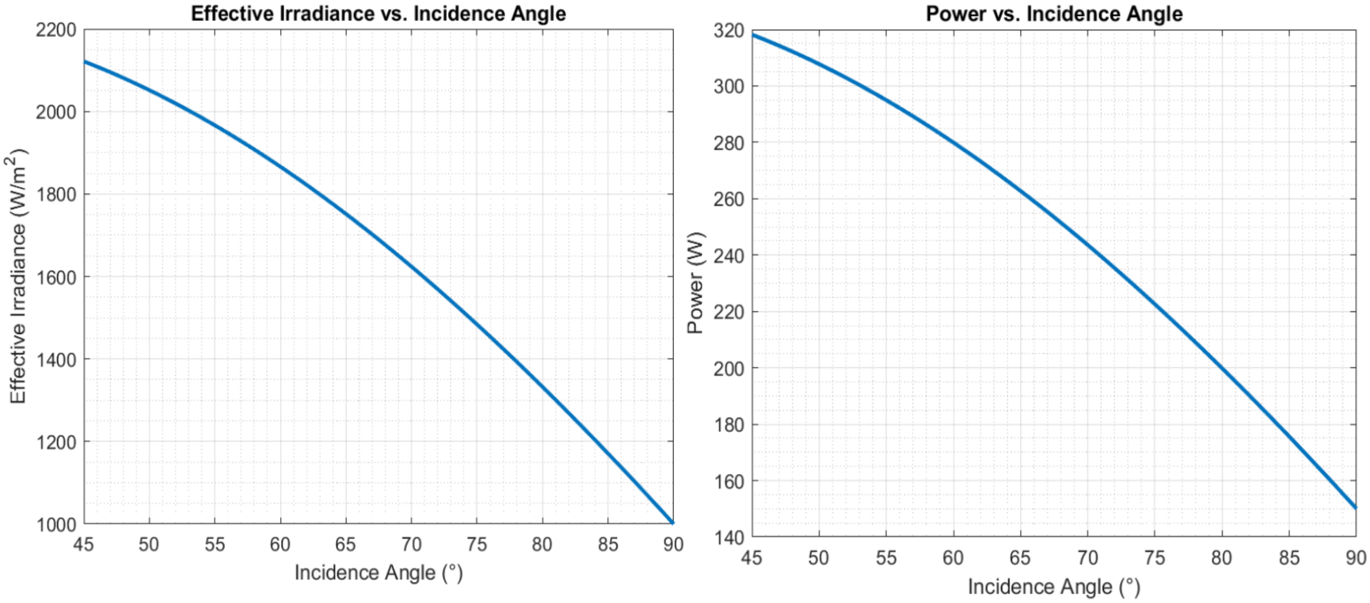


Figure 3.3. Effective irradiance and power generated from three irradiated solar panels.

3.2.1.2 STC Temperature Deviation

The previous subsection shows that body-mounted solar panels for a 1U CubeSat generate the maximum amount of power when three solar panels are irradiated with an incidence angle of 45°. Table 2.3 lists the specifications of the solar panel used in this project. Although the efficiency of the GP80*80-10A100 solar panel is not listed within the specifications, the efficiency of monocrystalline solar cells typically ranges between 15% - 24% [58]. [59] specifies that the solar irradiance at the top of Earth's atmosphere is 1,361 W m^{-2}. By applying the previous methodology that was used to form Table 3.1, the maximum power of the project's solar panel based on STC temperature is listed in the following table.

Table 3.2. Maximum solar array power output for a body-mounted 1U CubeSat.

		η = 0.15	η = 0.24
Irradiated Faces	Total Cell Area (m^2)	Power (W)	Power (W)
1	0.0064	1.307	2.090
2	0.0128	3.700	5.913
3	0.0192	8.315	13.304

Equation 3.5 was used to consider the impact of temperature deviation away from STC on the solar array's maximum power output. Since the GP80*80-10A100 monocrystalline silicon solar panel does not have its temperature coefficient listed, it will be assumed to be -0.45%/°C based on [60]. Table 3.3 shows the impact of the temperature deviation while the incidence angle is fixed at the optimal angle.

Table 3.3. Solar array power output based on temperature deviation away from STC.

	η = 0.15			η = 0.24		
	STC +5°C	STC +10°C	STC +15°C	STC +5°C	STC +10°C	STC +15°C
Irradiated Faces	Power (W)	Power (W)	Power (W)	Power (W)	Power (W)	Power (W)
1	1.013	0.719	0.425	1.620	1.150	0.680
2	2.864	2.033	1.201	4.582	3.252	1.922
3	6.444	4.573	2.702	10.310	7.317	4.324

3.2.1.3 Power Output by CubeSat Size

The same methodology used to obtain the values listed in Tables 3.2 and 3.3 can be applied to the other CubeSats. The MPO of a 1U body-mounted solar array along with losses due to cell temperature gains is listed in Table 3.4 and Table 3.5. For the body-mounted solar array configuration, MPO is reached when three solar panels are irradiated at an incidence angle of 45°.

Table 3.4. Maximum power output for a body-mounted solar array and power output losses due to cell temperature gains for 1U – 6U CubeSats (η = 0.15).

			STC +5°C	STC +10°C	STC +15°C
CubeSat Size (U)	Total Cell Area (m^2)	MPO (W)	Power (W)	Power (W)	Power (W)
1	0.0192	8.315	6.444	4.573	2.702
2	0.0320	13.858	10.740	7.622	4.504
3	0.0448	19.401	15.036	10.671	6.305
4	0.0512	22.173	17.184	12.195	7.206
6	0.0704	30.488	23.628	16.768	9.909

Table 3.5. Maximum power output for a body-mounted solar array and power output losses due to cell temperature gains for 1U – 6U CubeSats (η = 0.24).

			STC +5°C	STC +10°C	STC +15°C
CubeSat Size (U)	Total Cell Area (m^2)	MPO (W)	Power (W)	Power (W)	Power (W)
1	0.0192	13.304	10.310	7.317	4.324
2	0.0320	22.173	17.184	12.195	7.206
3	0.0448	31.042	24.057	17.073	10.089
4	0.0512	35.477	27.495	19.513	11.530
6	0.0704	48.781	37.805	26.830	15.854

3.2.2 Deployable Solar Array

For the body-mounted solar array configuration, it was discussed that the maximum amount of surfaces that could receive direct solar irradiation is three. For the deployable solar array configuration, the maximum amount of irradiated solar panels increases to five. The methodology that was used to determine the MPO and temperature relation for the body-mounted configuration could be applied to the deployable configuration. In doing so, Tables 3.6 and 3.7 were obtained. For the deployable solar array configuration, MPO is reached when five solar panels are irradiated at an incidence angle of 0°. The deployable sun-tracking solar array configuration shares the same MPO and temperature values/trends as the deployable solar array configuration. This is because the sun-tracking configuration continuously rotates the solar arrays to achieve an incidence angle of 0°. Practically, however, the non-tracking deployable configuration will not always achieve an incidence angle of 0. Thus, the tracking deployable configuration will match if not exceed the power performance of the former configuration as long as the power usage of the motors is not substantial.

Table 3.6. Maximum power output for a deployable solar array and power output losses due to temperature gains for 1U – 6U CubeSats (η = 0.15).

			STC +5°C	STC +10°C	STC +15°C
CubeSat Size (U)	Total Cell Area (m^2)	MPO (W)	Power (W)	Power (W)	Power (W)
1	0.0320	19.584	15.189	10.779	6.369
2	0.0576	35.277	27.340	19.402	11.465
3	0.0832	50.956	39.491	28.026	16.561
4	0.0896	54.876	42.529	30.181	17.835
6	0.1280	78.394	60.755	43.116	25.480

Table 3.7. Maximum power output for a deployable solar array and power output losses due to temperature gains for 1U – 6U CubeSats (η = 0.24).

			STC +5°C	STC +10°C	STC +15°C
CubeSat Size (U)	Total Cell Area (m^2)	MPO (W)	Power (W)	Power (W)	Power (W)
1	0.0320	31.357	24.302	17.247	10.191
2	0.0576	56.444	43.744	31.049	18.344
3	0.0832	81.529	63.185	44.841	26.497
4	0.0896	87.801	68.046	48.290	28.535
6	0.1280	125.430	97.208	69.986	40.764

3.2.3 Solar Panel Cost Comparison

The cost-effectiveness of each solar array configuration for CubeSats ranging from 1U to 6U is shown in Table 3.8. The cost-effectiveness values of each configuration were obtained by dividing the MPO by the total cost. For the body-mounted configuration, the cost-effectiveness of the solar panels does not vary across the various CubeSat sizes. This is because the maximum number of irradiated faces remains three. For the deployable configuration, the 3U CubeSat exhibited the most cost-effective solar array. As shown in Table 3.9, 92.9% of the cells are irradiated for the 3U deployable solar array. Thus, this result indicates that CubeSats arranged in a 1 x n form factor generate more power and are more cost-effective than CubeSats arranged in a m x n form factor. This expectation is verified by Table 3.10. Subsequently, it is important to determine whether the 21.43% and 25.00% wattage increase for 4U and 6U CubeSats is worth any subsequent design changes and risks.

Table 3.8. Cost-Effectiveness for each CubeSat size based on the solar array configuration (η = 0.15).

	Body-Mounted	Deployable
CubeSat Size (U)	Cost W^{-1}	Cost W^{-1}
1	5.04	2.14
2	5.04	1.98
3	5.04	1.92
4	5.04	2.04
6	5.04	1.96

Table 3.9. Cell area characteristics for solar panels within the body-mounted and deployable solar array configurations.

			Body-Mounted			Deployable		
CubeSat Size (U)	Solar Panels	TCA (m^2)	MISP	ICA (m^2)	Coverage	MISP	ICA (m^2)	Coverage
1	6	0.0384	3	0.0192	50%	5	0.032	83.3%
2	10	0.0640	5	0.0320	50%	9	0.0576	90.0%
3	14	0.0896	7	0.0448	50%	13	0.0832	92.9%
4	16	0.1024	8	0.0512	50%	14	0.0896	87.5%
6	22	0.1408	11	0.0704	50%	20	0.1280	90.9%

Table 3.10. Power performance comparison for the deployable solar arrays arranged in the m x n and 1 x n configurations (η = 0.15).

		m x n				1 x n			
CubeSat Size (U)	TCA	ICA (m^2)	Coverage	MPO (W)	Cost W^{-1}	ICA (m^2)	Coverage	MPO (W)	Cost W^{-1}
1	0.0384	0.0320	83.3%	19.584	2.14	0.0320	83.3%	19.584	2.14
2	0.0640	0.0576	90.0%	35.277	1.98	0.0576	90.0%	35.277	1.98
3	0.0896	0.0832	92.9%	50.956	1.92	0.0832	92.9%	50.956	1.92
4	0.1024, 0.1152	0.0896, 0.1088	87.5%	54.876	2.04	0.1088	94.4%	66.635	1.68
6	0.1408, 0.1664	0.1280, 0.1600	90.9%	78.394	1.96	0.1600	96.2%	97.992	1.57

4. Deployment Architecture

4.1 Deployment Mechanism

4.1.1 Retention and Release Mechanism

A burn wire system has been selected as the R&R mechanism due to its simplicity and space heritage. Table 4.1 shows the various materials that could retain the solar panels in the stowed state. Ideally, the selected thread material should have a low melting point and a high tensile strength. Initially, the Vectran, Dyneema, and Kevlar fibers were examined. However, these fibers were excluded because their equivalent thread data was unknown. Based on the available options, polypropylene and nylon 6 have the lowest melting points. For tensile strength, the best options are nylon 6/6, nylon 6, and polyester. By minimizing the melting point while maximizing the tensile strength, nylon 6 is the optimal option for the retention system. It is important to note that Table 4.1 lists the tensile strength of the fiber rather than the thread; this allows the comparison between each material to be consistent. The materials were not compared in thread form because not all of the tensile strength data for each material is available for a particular size.

Table 4.1. Melting point and general tensile strengths of various fibers [61–63].

Fiber	Melting Point (°C)	Tensile Strength (MPa)
Nylon 6 Monofilament	220	62
Nylon 6/6 Monofilament	260	77
Polyester	250	65.6
Polypropylene	164	33
PTFE (Teflon)	330	10 - 43

By applying Newton's 2nd law, the force exerted on the solar panel during the launch phase could be determined. Since g = 9.81 m s^{-2}, m = 0.02268 kg, and assuming a launch load of 15 g, the force exerted on the solar panel is 3.3374 N. With a safety factor of three applied, the selected nylon 6 thread should withstand at least 10.0121 N of force. A COTS nylon monofilament with a line diameter of 0.18 mm and break strength of 17.793 N was selected; COTS with smaller line diameters were considered the ideal option due to the volumetric constraints of the CubeSat. A knot is used to secure the primary solar panels to the CubeSat. The energy required to cut the nylon 6 monofilament thread was calculated by applying equation 4.1. The energy required to reach the melting point of the thread is given by equation 4.2 and the energy required to melt the thread at its melting point is given by equation 4.3.

$$Q_{total} = Q_{heat} + Q_{melt} \tag{4.2}$$

$$Q_{heat} = mC_p\Delta T \qquad (4.1)$$

$$Q_{melt} = mH_f \qquad (4.2)$$

However, since the mass of the thread is unknown and that experimental measurements would not be precise, Q_{heat} and Q_{melt} were expressed in terms of volume and density (equations 4.4 – 4.5). The values of ρ, H_f, and C_p were considered to be 1.15 $g\ cm^{-3}$ [64] and 230 $J\ g^{-1}$ [65], and 1.7 $J\ g^{-1}\ K^{-1}$ respectively. With a resistor diameter of 2.3 mm, L_{wire} was calculated to be 7.2257 mm. Assuming that the CubeSat is deployed a few minutes after LEO, the internal temperature is assumed to be 25°C; this results in a ΔT value of 195°C. By substituting all relevant values into equation (4.4, Q_{heat} and Q_{melt} was calculated to be 0.0701 J and 0.0486 J. Subsequently, 0.1187 J of energy is required to thermally cut the retention mechanism.

$$Q_{heat} = \rho AL_{wire}C_p\Delta T = \rho\,\pi r^2 L_{wire}C_p\Delta T \qquad (4.3)$$

$$Q_{melt} = \rho AL_{wire}C_p\Delta T = \rho\,\pi r^2 L_{wire}H_f \qquad (4.4)$$

Equation 4.6 shows the power dissipated by a resistor. For a burn-wire circuit, greater power dissipation causes the resistor to experience greater heating. Thus, it is evident that a low-resistance resistor should be selected. With a power supply of 12V and a resistor with 1Ω, the power dissipated by the resistor is 144W; the voltage is held constant. By applying equation 4.7, the time to burn the retention mechanism is 0.825 ms. During this process, 118.730 J of heat is generated by the resistor (equation 4.8). It is important to obtain an experimental burn time because each manufacturer uses different techniques during the construction of the nylon 6 monofilament wire. For this reason, the values of H_f and C_p are most likely to vary between each manufacturer.

$$P = VI = V^2 R^{-1} \qquad (4.5)$$

$$t_{burn} = \frac{Q_{melt}}{P} \qquad (4.6)$$

$$Q_{resistor} = I^2Rt = V^2R^{-1}t \qquad (4.7)$$

4.1.2 Hinge Mechanism and Torsion Spring

After the R&R mechanism has released the solar panel, a hinge is necessary to rotate the solar panel from the stowed state to the deployed state. Two torsion springs were selected to deploy the solar panel. The legs of the torsion spring are inserted within the solar panel supporting structure. Data for available COTS 90° torsion spring designs are aggregated in Table F.1. The primary driver during the selection process was that the torsion spring should have a minimal outer diameter and that its spring index should be within the interval of (4, 12) to ease the manufacturing process of the torsion spring. Specifications of the selected torsion spring are listed in the following table.

Table 4.2. Primary hinge torsion spring specifications. *Since the value of E is unknown, the common E value for stainless steel was used.

Material	Stainless steel
E (GPa)*	195
σ_y (MPa)	250
n	9
d_{wire} (mm)	0.5
d_{inner} (mm)	3.5
d_{outer} (mm)	4.5
D_{mean} (mm)	4
L_{leg} (mm)	34
θ_{spring} (°)	90
Width (mm)	5.5

The hinge system is composed of hinges driven by 90° and 180° torsion springs. The 90° torsion springs are used to deploy the primary solar panels, whereas the latter is used to deploy the tertiary solar panels. The 90° hinges are attached directly to the CubeSat, whereas the latter is connected to each subsequent solar panel. For 1U CubeSats, only 90° torsion springs are used. For 2U and larger CubeSats, 90° and 180° torsion springs are used. The hinge mechanism (Figure 4.1) was sized based on the specifications from Table 4.2. The material and mass of each part of the hinge mechanism are shown in Table 4.3; the mass of the hinge mechanism is 4.89 g. Although the 90° and 180° torsion springs are expected to deploy the solar panel at the corresponding deflection angle, there is a possibility that the torsion springs will overshoot. The blocking element prevents unintended rotation from occurring.

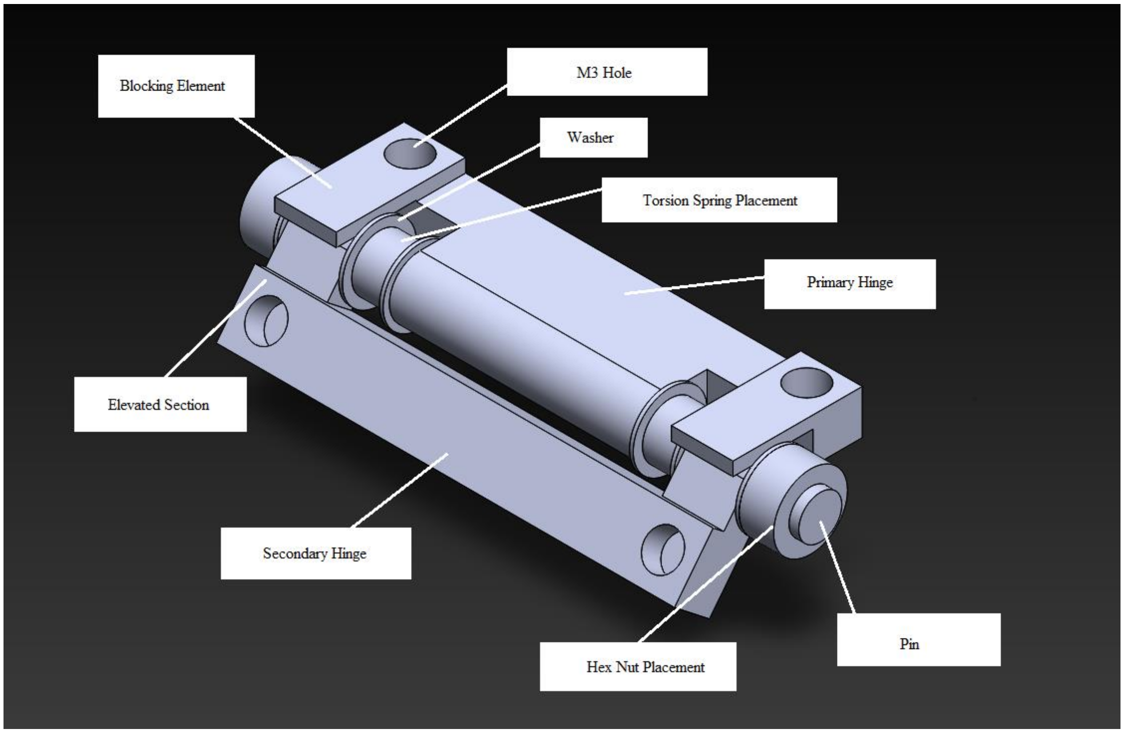


Figure 4.1. Hinge mechanism.

Table 4.3. Part material and mass characteristics of the hinge mechanism.

Part	Material	Total Mass (g)
Hex Nut	AISI 304	2.52
Pin	PA Type 6	0.48
Primary Hinge	PA Type 6	0.65
Secondary Hinge	PA Type 6	0.88
Torsion Spring	AISI 304	0.3
Washers	Nylon 6/10	0.06

A COTS 180° torsion spring with an inner diameter of 3.5 mm was selected to maintain capability with the primary hinge mechanism. The selected torsion spring has all the characteristics in Table 4.2 aside from the deflection angle. Since all the geometric characteristics are the same, only the starting position of the secondary hinge is affected (Figure 4.2). Although the starting positions are different, the final positions are the same (Figure 4.3). The elevated section ensures that the solar panels are deployed on the same plane.

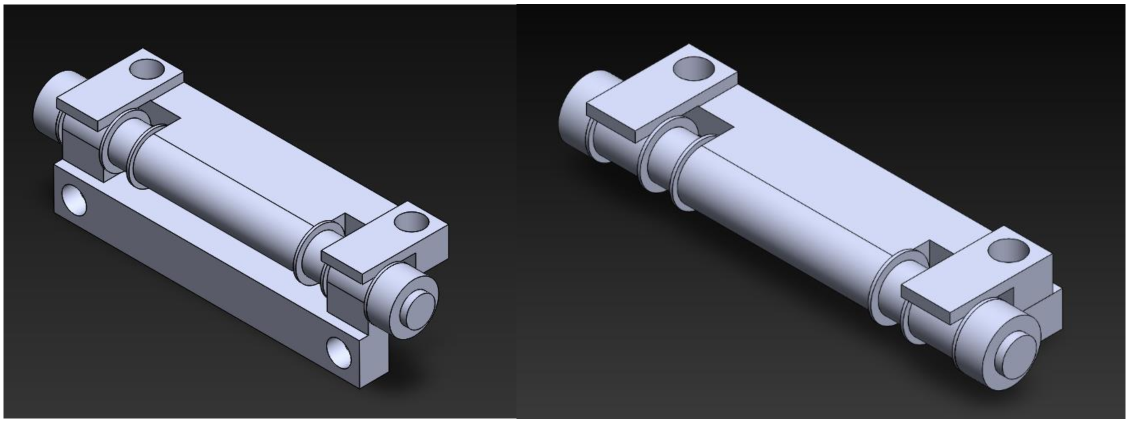


Figure 4.2. Hinge stowed position for the primary (left) and secondary solar panel (right).

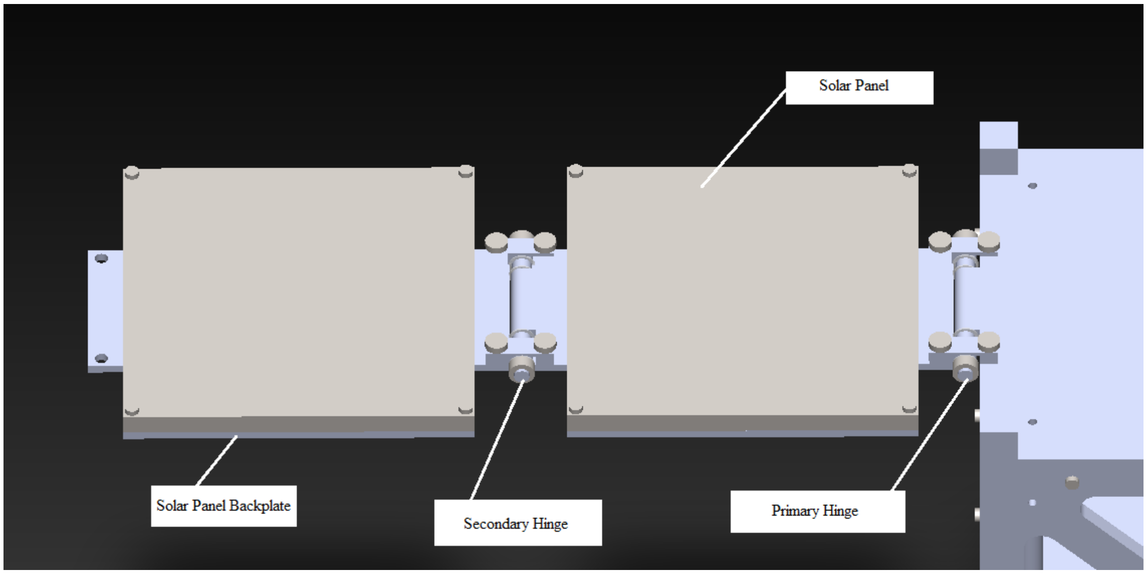


Figure 4.3. Hinge positions for a deployed solar panel.

To deploy a solar panel to 90°, and assuming friction is negligible in the hinge within LEO, the following torque equation was applied.

$$T_{spring} = k\theta_{spring} \tag{4.8}$$

For circular springs k is defined as follows.

$$k = \frac{E{d_{wire}}^4}{64D_{mean}n} \tag{4.9}$$

However, the former equation is applied in systems where the deflection of the spring is purely torsional. By considering friction, spring curvature, and spring bending, the helical spring equation provides a more accurate definition of k.

$$k = \frac{E{d_{wire}}^4}{10.8D_{mean}n} \tag{4.10}$$

Equations 4.12 – 4.14 are correction factors that consider the non-ideal curvature effect of a helical spring [66]. Equation 4.12 only considers curvature effects due to bending stress, whereas equations 4.13 and 4.14 consider bending stress and shear stress.

$$C = \frac{D_{mean}}{d_{wire}} \tag{4.11}$$

$$K_B = \frac{4C^2 - C - 1}{4C(C - 1)} \tag{4.12}$$

$$K_{Wahl} = \frac{4C - 1}{4C - 4} + \frac{0.615}{C} \tag{4.13}$$

$$K_{Berg} = \frac{4C + 2}{4C - 3} \tag{4.14}$$

By applying equations 4.8, 4.10, and 4.11, the respective values were calculated as T = 1.202 Nm, k = 0.7641 Nm rad^{-1} and C = 8.1667. The bending stress and shear stress of the torsion spring are given by the following equations.

$$\sigma_b = \frac{32TDC_F}{\pi {d_{wire}}^3} \tag{4.15}$$

$$\tau = \frac{16TDC_F}{\pi {d_{wire}}^3} \tag{4.16}$$

By substituting the values from Table 4.2 into equations 4.8 – 4.16, the bending stress, and shear stress for each method were calculated (Table 4.4). The ultimate shear strength of stainless steel was assumed to be 0.57 of its yield strength; this value was calculated to be 122.55 MPa. The safety factor for shear stress and bending stress of the material is given by the following respective equations.

$$SF_\tau = \frac{\tau_y}{\tau} \tag{4.17}$$

$$SF_\sigma = \frac{\sigma_y}{\sigma} \tag{4.18}$$

Table 4.4. Shear and bending stress of the torsion spring for each correction factor applied.

	K_B	K_{Wahl}	K_{Berg}
σ_b (Pa)	$2.730*10^7$	$2.850*10^7$	$2.823*10^7$
τ (Pa)	$2.730*10^7$	$1.425*10^7$	$1.412*10^7$
SF_σ	7.876	7.543	7.617
SF_τ	8.979	8.599	8.684

Based on Table 4.4 the torsion spring exhibits the greatest bending stress and shear values when the Wahl correction factor is applied. The respective safety factors for σ and τ are 7.543 and 8.599. Since the safety factors are greater than 1.5, this indicates that the selected torsion spring could deploy a solar panel to 90° without failing. By modeling the motion of the spring-panel system as simple harmonic motion, the governing equation, period, MOI of the solar panel, and natural frequency of the system are listed as follows.

$$\theta(t) = \theta_{spring}\sin(\omega t) \tag{4.19}$$

$$T = \frac{2\pi}{\omega} \tag{4.20}$$

$$\omega = \sqrt{\frac{k}{I}} \tag{4.21}$$

$$I = \frac{1}{3}mL^2 \tag{4.22}$$

$\theta(t) = \theta_{spring}$ = 90° once the solar panel is fully deployed. The statement $\sin(\omega t)$ = 1 is true when ωt = 0. By isolating for t, the deployment time equation is obtained.

$$t = \frac{\pi}{2\omega} \tag{4.23}$$

The time for the solar panel to reach any given angular position was determined by applying the energy conservation principle (equation (4.26). The potential energy of the spring and rotational kinetic energy of the solar panel is given by equations 4.24 and 4.25. By rearranging equation 4.26 and evaluating the integral with respect to dθ, equation 4.19 was obtained; in this instance θ_{spring} is a constant.

$$U_{Spring} = \frac{1}{2}k{\theta_{spring}}^2 \tag{4.24}$$

$$KE_{Panel} = \frac{1}{2}I\left(\frac{d\theta_{spring}}{dt}\right)^2 \tag{4.25}$$

$$k\theta_{spring}{}^{2} = k\theta^{2} + I\left(\frac{d\theta_{spring}}{dt}\right)^{2} \tag{4.26}$$

$$t = \sqrt{\frac{I}{k}}\sin^{-1}\left(\frac{\theta}{\theta_{spring}}\right) \tag{4.27}$$

By substituting the appropriate terms into equation 4.23 or 4.27, the deployment time was calculated to be 0.0630 seconds. This quick deployment time is problematic because the high angular velocity of the solar panels deploying could cause rotational instability with the CubeSat or cause strain to occur to the CubeSat's ADCS. An alternative torsion spring design—where d_{wire} = 0.5 mm, d_{outer} = 6, and n = 9—from Table F.1 allows the solar panel to be deployed in 0.090 s. To decrease the deployment time, the MOI of the solar panel could be increased, or a custom torsion spring could be manufactured by applying equations 4.21 – 4.23 and equations 4.10 – 4.18 for a specific target deployment time. Equation 4.10 could be expressed in terms of C as shown by equation 4.28. In this equation, n should be selected rather than solved because n could result in fractional values if otherwise. Due to manufacturing and monetary limitations, COTS torsion springs will be used, and the MOI of the solar panel will be increased by increasing its mass.

$$k = \frac{1}{n}\frac{Ed_{wire}{}^{3}}{10.8C} \tag{4.28}$$

4.2 2-Axis SADM Design

To create a deployable solar array with 2-axis controllability, this project uses a system of motors, rotary disks, yokes, and gears. Figure 4.4 shows a 1U CubeSat skeleton. A motor could be attached to a yoke to rotate a solar array about any axis of the CubeSat as shown in Figure 4.5. For example, to rotate a solar array about the x-axis, the motor-yoke configuration is aligned with the same axis; ideally, this configuration would be placed at the top or bottom external surfaces of the CubeSat to optimize space within the CubeSat.

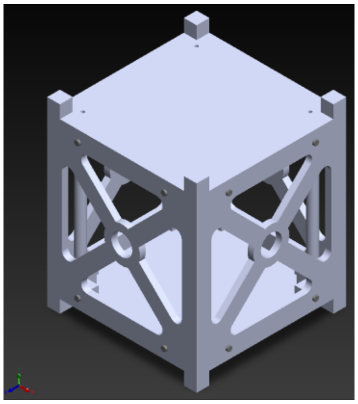


Figure 4.4. 1U CubeSat.

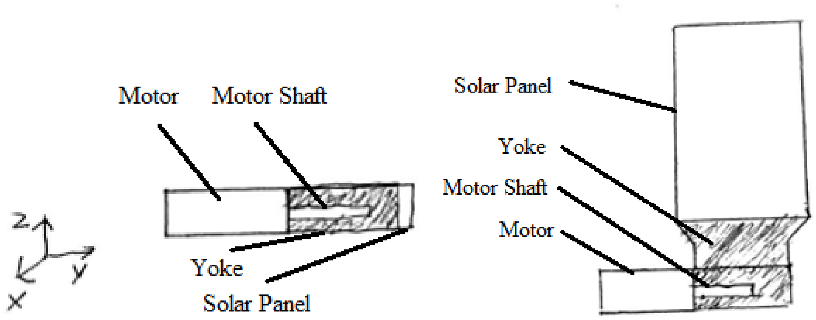


Figure 4.5. Motor-yoke configuration in the stowed position (left) and deployed position (right); rotation occurs about the y-axis.

A similar process is used for 2-axis controllability, except that a rotary disk is necessary. As shown in Figure 4.6, the motor-yoke configuration is mounted on top of a rotary disk. The yoke motor is used to deploy the solar array, and the rotary motor is used to rotate the solar array with respect to a normal axis of the CubeSat. For example, to deploy a solar array about the x-axis, the previous motor-yoke configuration is followed. To rotate the solar array about the y-axis, the rotary disk and motor are aligned with the same axis.

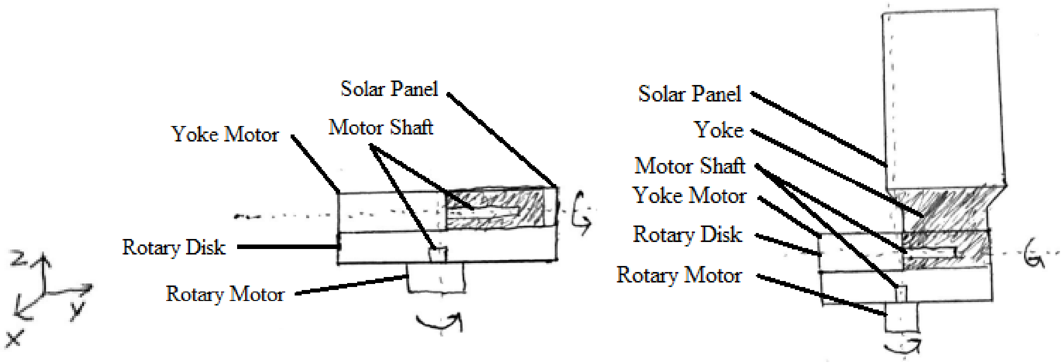


Figure 4.6. Motor-yoke-disk configuration in the stowed position (left) and deployed position (right)

4.1.1 Prototype SADM

4.1.1.1 Motor-Yoke Configuration

As previously discussed, NEMA 17 motors (42 mm x 42 mm x 23 mm) have been selected to drive the SADA. To implement the deployment process as shown in Figure 4.5, the motor must be mounted vertically within the CubeSat; it is not possible to place the motor outside the CubeSat due to its size. The motor is secured to CubeSat via mounting brackets and M3 x 6 mm screws (Figure 4.7).

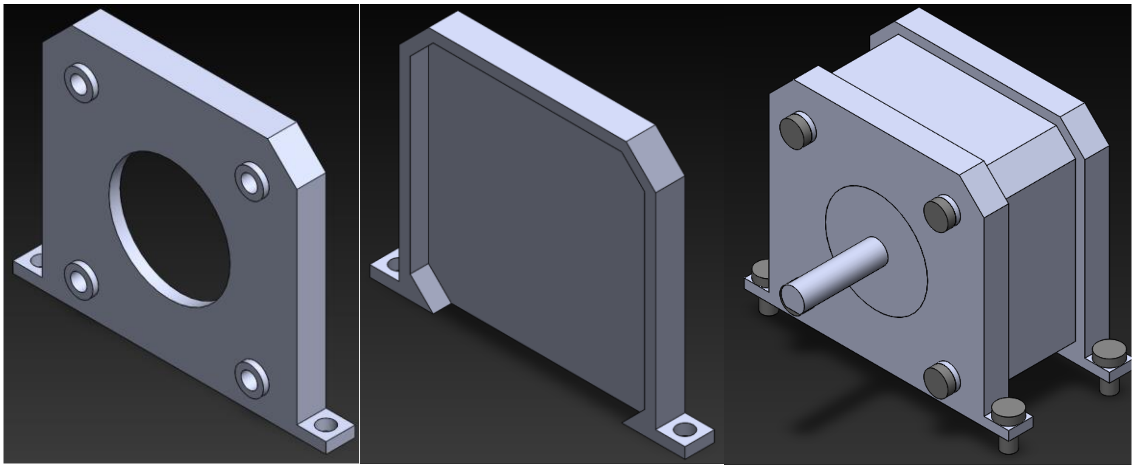


Figure 4.7. Vertical front mounting bracket (left), rear mounting bracket (middle), and motor-bracket assembly (right)

The yoke shaft is attached to the motor shaft via a transition fit and is used to support the installation of a solar panel (Figure 4.8); M2 x 6 mm screws are used. Originally the yoke shaft had a length of 42 mm, and the yoke arm had a width of 10 mm. The yoke shaft was set to this length to allow it to be connected to a bearing for additional support. However, due to volume limitations within the CubeSat, the yoke shaft and arm were reduced to a length and width of 10 mm, respectively. The yoke's arm has a length of 32.8 mm to ensure that the solar panel does not touch the motors. The motor-yoke configuration is shown in Figure 4.9.

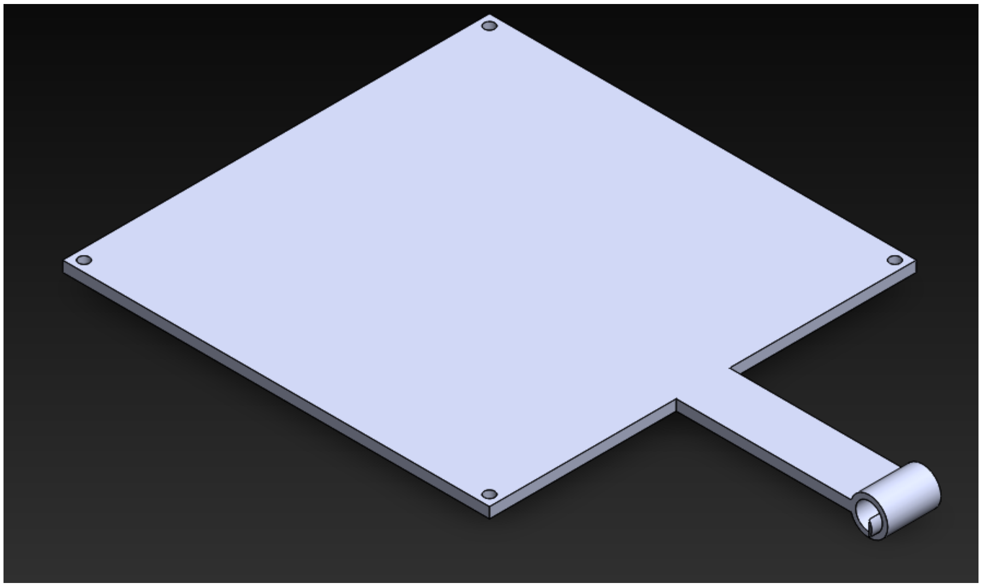


Figure 4.8. SADA yoke.

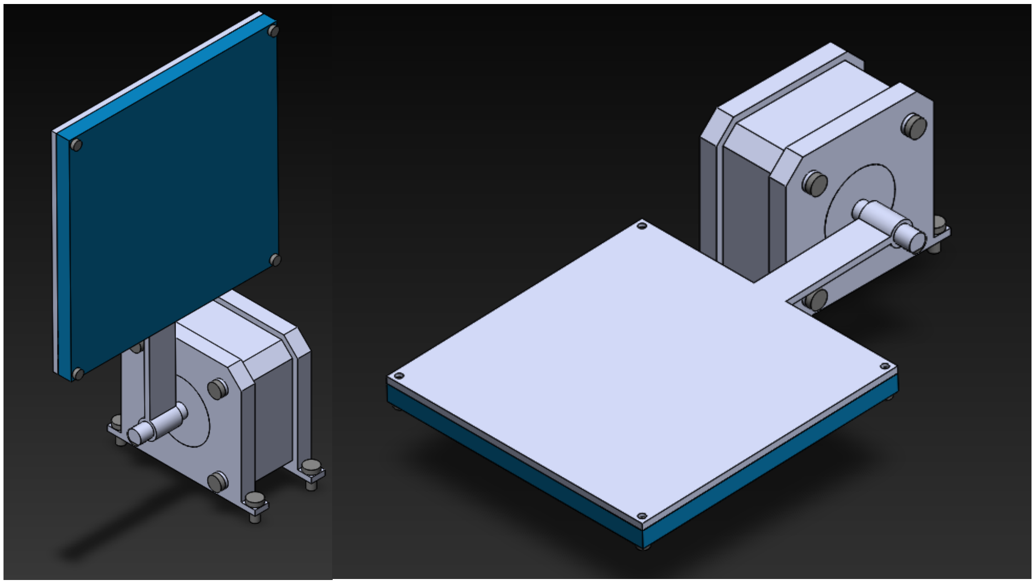


Figure 4.9. Stowed solar panel (left) and deployed solar panel (right) based on the motor-yoke configuration.

4.2.2 Motor-Yoke-Disk Configuration

The previous motor-yoke configuration could still be used to deploy the solar array. However, a rotary disk and horizontal mounting bracket are necessary to permit rotational controllability of the solar array; the motor shaft is aligned normal to each CubeSat surface. Figure 4.10 shows a horizontal mounting bracket, which could be used to secure the motor-yoke-disk configuration directly to the surface of a chassis wall. Although the vertical mounting bracket could be attached to the rotary disk to secure the system, the base of the mounting brackets would need to be enlarged to prevent the rotary disk from contacting the CubeSat surfaces. Such an adjustment would require an extrusion of at least 13 mm. This would require the use of M3 x 16 mm screws, which would increase the mass of the system. Alternatively, the vertical mounting bracket could be attached to the mounting plate. This mounting plate would have an extruded cut to permit clearance and would be secured in place via spacers and hex nuts. However, this process introduces unnecessary mass within the system. For these reasons, the horizontal mounting plate was determined to be optimal.

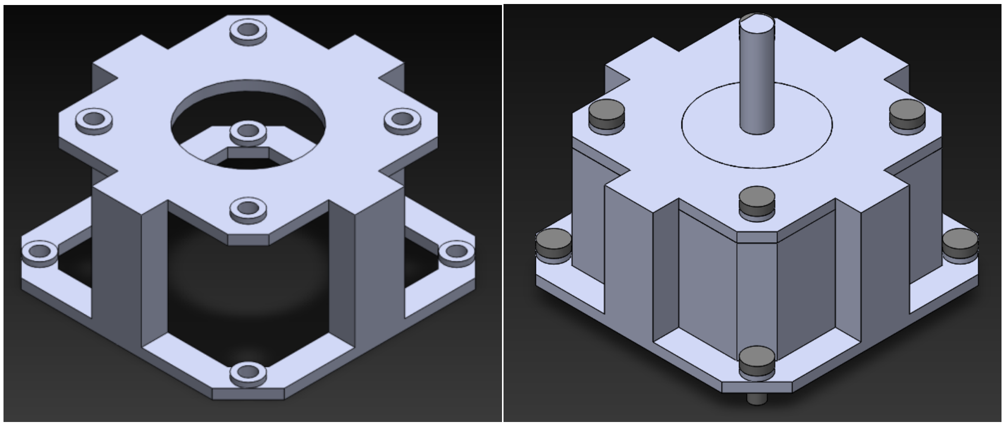


Figure 4.10. Horizontal mounting bracket (left) and motor bracket assembly (right).

The rotary disk is a cylinder secured to the rotary motor shaft via a transition fit. The top surface contains four M3 holes with a depth of 3 mm and contains a central hole for the motor shaft. On the bottom surface, an extrusion is present to permit clearance between the rotary disk and the M3 screws present on the horizontal mounting bracket.

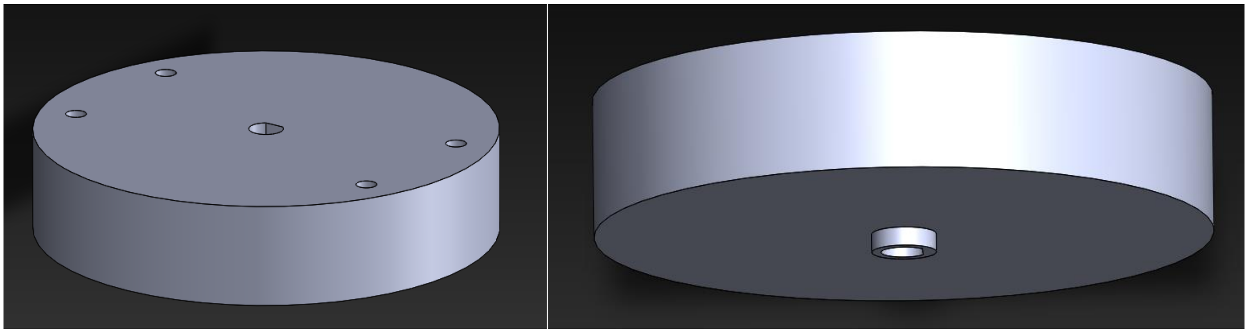


Figure 4.11. Rotary disk.

By integrating the rotary disk and the horizontal motors into the motor-yoke configuration, the motor-yoke-disk configuration is obtained in the following figure. This configuration allows the solar panels to deploy about each CubeSat surface and to rotate normally to each CubeSat surface.

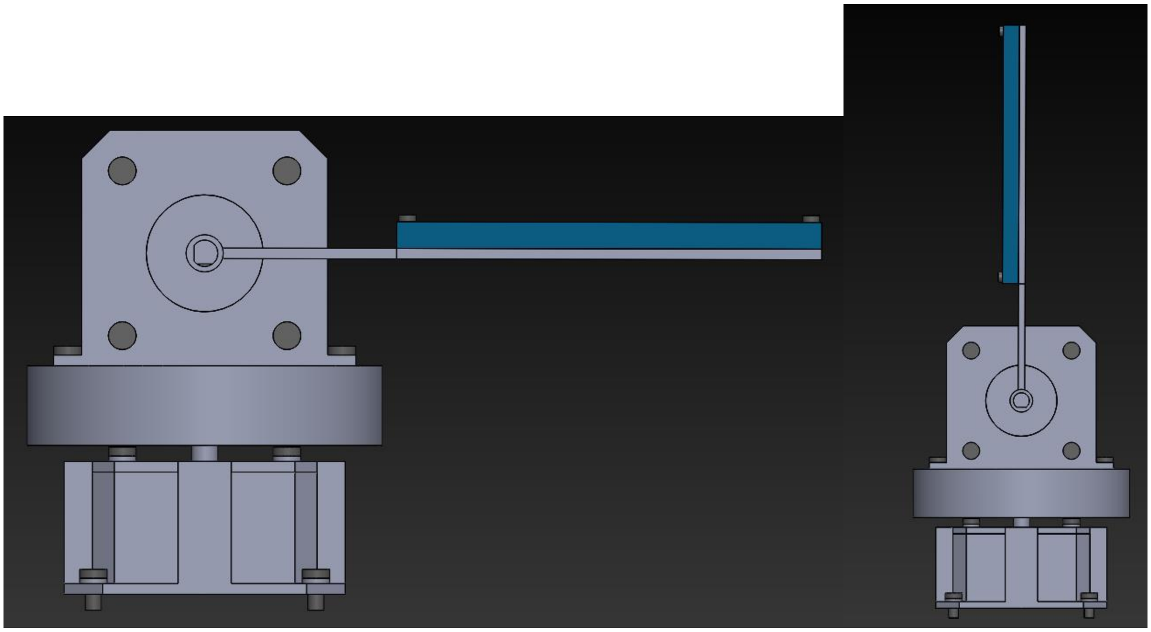


Figure 4.12. Stowed solar panel (left) and deployed solar panel (right).

4.1.2 2nd Iteration SADM

Figure 4.13 shows that the horizontal mount, rotary disk, vertical mount, and motors fit within the 1U CubeSat. However, due to volume constraints, it is evident that the yoke arm and yoke mounting plate do not fit within the CubeSat. Additionally, due to the size of the motor-yoke-disk configuration, only one configuration could be integrated within the CubeSat. Subsequently, an R&R mechanism will need to be developed for the deployment of the solar arrays.

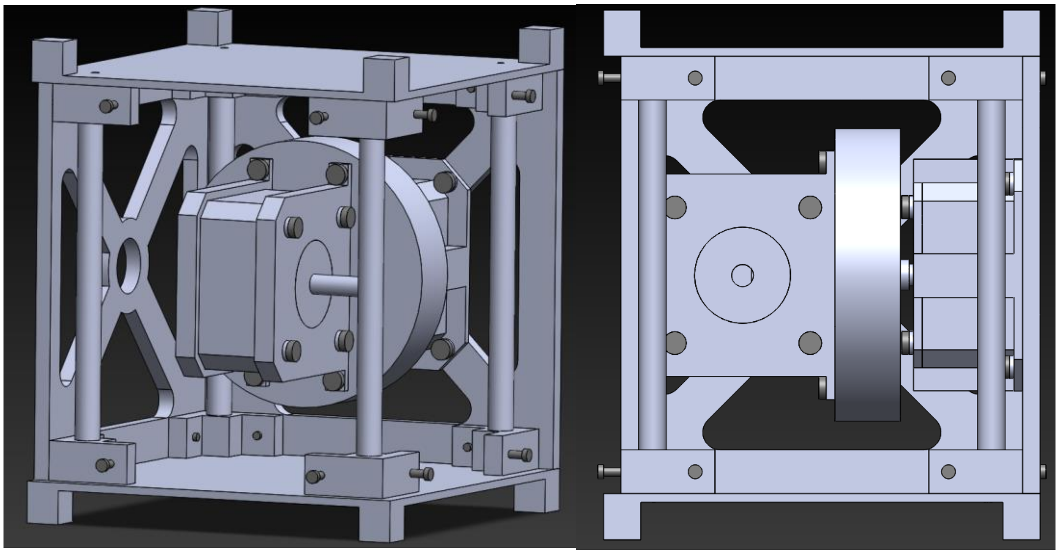


Figure 4.13. Yoke SADM isometric perspective (left) and right perspective (right).

A remedy to the volume issue is to use smaller stepper motors. Although smaller stepper motors do exist—these motors could be placed externally—, such motors are outside of the project's budget. Initially, spur gears were considered to deploy the solar array, in which they would be secured to the motor shaft and support shafts via transition fits. However, using spur gears would increase the mass of the system due to the presence of support shafts to keep the gears in place. Additionally, the presence of the avionics rod's mounting pads obstruct the placement of the gears.

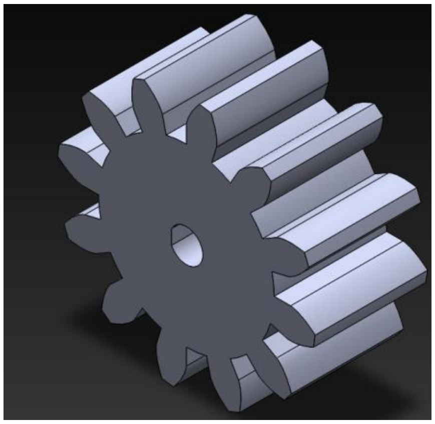


Figure 4.14. Spur gear.

A linkage-based rail-block system was then considered to deploy the solar panels. This system involves a rotary disk (Figure 4.15) attached to the deployment motor—which was formerly the yoke motor—, a rail-block system (Figure 4.16), and a linkage (Figure 4.17). This linkage forms a connection based on a pin on the rotary disk and a pin on the rail-block system. The linkage would then be secured to each pin via a transition fit. As the vertical motor rotates, the linkage rotates causing the rail's block to push the solar array into the deployed position. As shown in Figure 4.18, the linkage SADM system is compact and fits within the CubeSat.

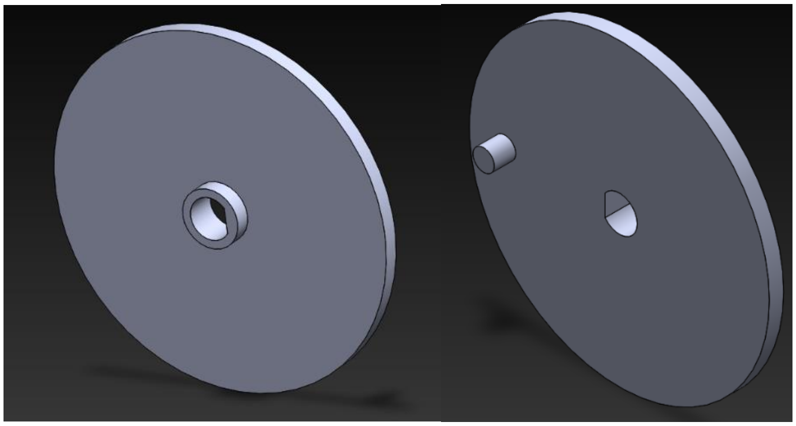


Figure 4.15. Rotary disk rear surface (left) and front surface (right).

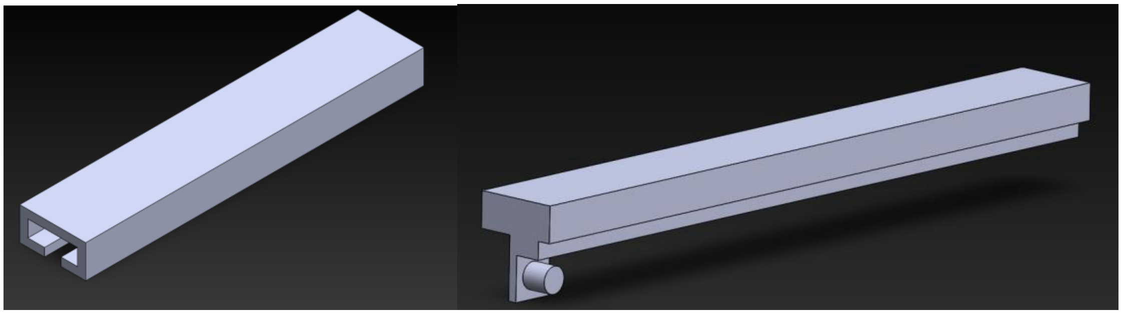


Figure 4.16. Rail housing (left) and rail block (right).

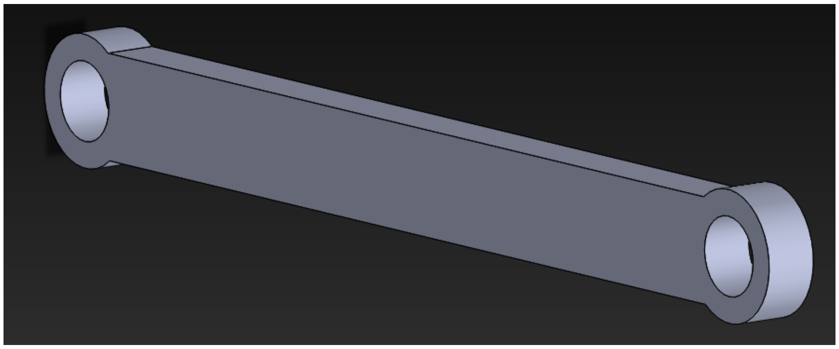


Figure 4.17. Linkage.

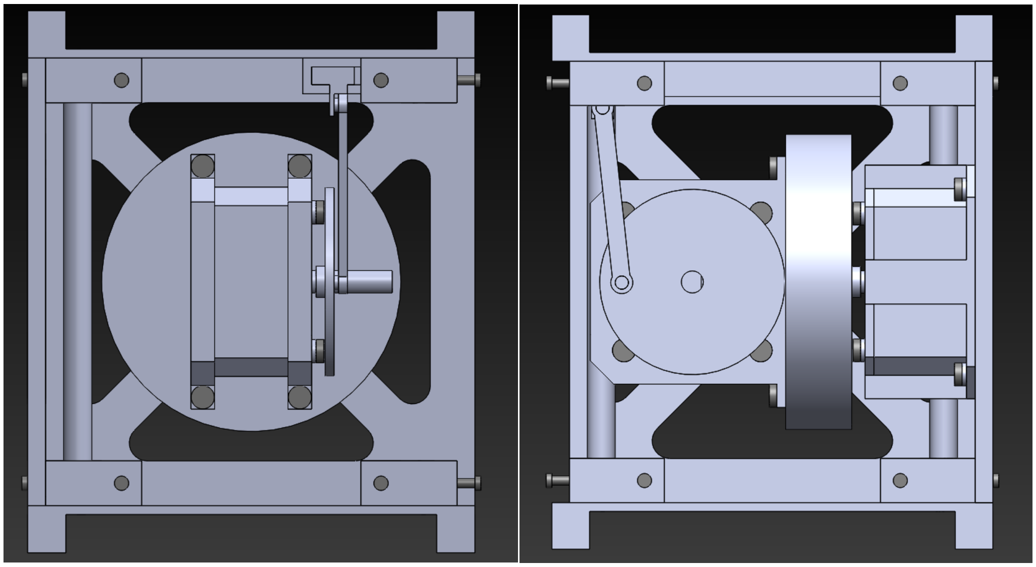


Figure 4.18. Linkage SADM front perspective (left) and right perspective).

The linkage rail-block permits rotation with respect to the axis parallel to each chassis wall. This resolve the volumetric issues of the initial prototype. However, considering that the linkage and rail system are fixed, rotation with respect to the solar panel longitudinal axis is not possible. This lead to

5. SADM Assembly

5.1 Structural Subsystem Assembly

5.1.1 CubeSat Assembly

A 2U CubeSat was chosen as the initial candidate for assembly. The structural components from Chapter 2 and Chapter 4 were 3D printed using PETG HF; all structural components with their 3D printing specifications are listed in Table 5.1. PETG HF was specifically chosen over PLA due to its greater strength, flexibility, and temperature capacity.

Table 5.1. Structural components involving filament usage and print time.

Structural Component	Amount	Unit Filament (g)	Total Filament (g)	Unit Print Time
Baseplate	4	27.93	55.86	0:45:47
Chassis Wall	4	16.43	65.72	0:30:55
Chassis Wall Modified	2	14.41	57.64	0:28:28
Solar Panel Backplate	4	19.34	77.36	0:32:19
Connector Baseplate	1	56.26	56.25	1:15:0
Pin	4	0.5	2	0:8:30
Secondary Hinge	4	1.12	4.48	0:13:13
Primary Hinge	4	1.25	5	0:13:54

As can be observed in Figure 2.3, the baseplate—the component the chassis walls are attached to—was designed with a rail at its base. However, the 3D printer slicer software indicated that the baseplate contained a floating cantilever beam. Although the baseplate could still be printed with support filament material, filament usage would have doubled, and its print time would have increased to over three hours. The rail was removed to conserve filament and expedite the prototyping process. The connector baseplate (Figure 2.6), which is necessary to connect each CubeSat unit, suffers from the cantilever beam issue. However, unlike before none of the baseplate sections could be removed. Initially, the baseplate was sent to the slicer with no support material. Under close observation, the 3D printer was unable to print the central plate; spaghettification occurred during this phase. Support material was then applied; this increased filament usage to 56.27g and the print time to 3:15:0. It is important to note that 3D printed objects should not be taken off the plate until it has cooled. The object could experience bending if otherwise; the hot material sticks to the plate and any applied moments could apply to bending deformation.

Generally, when holes are 3D printed in a material, there is a possibility that the hole would be over-extruded or thermally wrapped. Thus, heat-set inserts of M2 x 2 mm and M2 x 4 mm were thermally fitted for all relevant structural components. An alternative to heat-set inserts are press-fit inserts. However, heat-set inserts were selected due to the restrictive locations of the holes and small material thickness around the holes; there was a possibility that hammering in the inserts would crack the 3D printed components. Upon examination of the 3D printed components, the holes contained sections of over extrusion as expected. The M2 inserts were thermally inserted via a 60W soldering iron despite the over extrusion. M2 x 2 mm inserts were used for each chassis wall and the baseplate's outer region, whereas M2 x 4 mm inserts were used within the baseplate's inner region. The inserts were initially hand placed into position and then inserted via the soldering iron's tip. Although touching the inserts with the soldering iron's side would allow greater contact, this process resulted in lesser insertion guidance and greater time usage.

Table 5.2. Heat-set insert installation issues and mitigations/solutions

Issue	Mitigations/Solution
The soldering iron would stick to the soldering iron's tip during. This caused the insert to become misoriented and enlarge the hole.	The soldering iron was quickly removed to minimize damage. A different insert was hand-placed into position and the sides of the soldering iron were used.
The connector baseplate used through holes rather than blind holes. Subsequently when the insert was misoriented, there was no method to reorient the inserts.	Extra attention to ensuring that the inserts were perfectly aligned prior to fitting was taken. Revisions to the connector baseplate could incorporate blind holes.
The threaded inserts would bubble out of the hole if the soldering iron was left attached to the surface for too long. This caused the hole to be enlarged and melted plastic to fall into the hole.	The soldering iron was not left attached to a surface for more than two seconds. Melted plastic was removed by using a M2 Allen wrench.
The insert would become skewed despite prior guidance.	The sides of the insert were tapped until realignment was achieved.

The assembly process of the CubeSat was quick and simple. First, the walls were attached to the baseplate via M2 x 8 mm screws. For the second set, this process was repeated for the second baseplate. Lastly, the walls of the second set were screwed onto the connector baseplate. The modularity of the screw-based design allows any wall or baseplate for quick internal access. The issues encountered during the insert installation (Table 5.2) lead to a cascading issue during the assembly process. This issue was that the inserts between the baseplates and chassis walls were not aligned. Either the insert was misaligned too far up or down. More specifically, three faces were affected by the failures of five misaligned inserts. The misalignment issue was resolved for one of the faces by flipping the orientation of the wall; the inner surface was swapped to face outward. This solution did not work for the two misaligned inserts for the two affected faces. After the chassis wall was 3D printed, the wall was removed when the plate was still hot. Although substantial bending was present, this had caused no alignment issues. The connector baseplate experienced bending—removed when the plate was cold—and under extrusion in some areas. This caused no alignment issues.

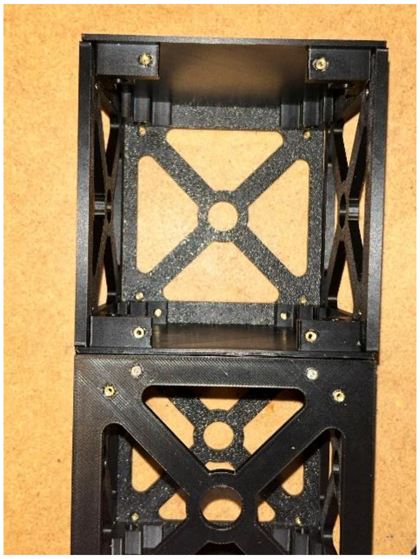


Figure 5.1. Chassis walls and baseplates thermal-screw assembly.

5.1.2 Hinge Assembly

Figure 4.1 shows each part of the hinge assembly that was designed around a 3.5 mm inner diameter 90° torsion spring. The washers and hex nut were excluded because there was no washer size that matched the M3.5 definition without its outer diameter being too large. Secondly, there were no available M3.5 hex nuts. Lastly, the M3 holes were downsized to M2 to allow streamline integration with other components. Initially the pin and hinge components were designed that they would be connected via an interference fit. When each hinge component was printed, the pin transition fit was insufficient. This is because the pin could not be forced into the hinges. The pin was sanded down and forcefully inserted via a twisting motion. When this pin was applied to other hinges during testing, the blocking element and central region of the primary hinge were fractured. This was remedied by thickening the central region and the blocker elements. Three additional pins were printed, but they all fractured during the sanding the process. Considering the tolerancing of the holes, three additional pins sizes were tested: 3.3 mm, 3.2 mm, and 3.15 mm. The 3.2 mm and 3.15 mm pins were able to fit in the holes without any sanding. The 3.2 mm pin required a moderate amount of force and twisting to be inserted; this eventually led to the fracturing of some of the hinges. The 3.15 mm pin required a minimal amount of force to be inserted and continued testing showed no structural pin or hinge failures. The 3.15 mm pin was sanded to allow easier fits. It is important to note that one of the pins was over sanded during testing. More specifically, when the pin's longitudinal axis was aligned with the ground, the pin would slide out of the hinge.

An issue during the insert installation was that one of the blocking elements thermally deformed downward. This was mitigated by quickly readjusting the blocking element while it was relatively still warm. During blocking element testing the deformed blocker still successfully prevented the overextension of the solar panel backplate. There were no misalignment issues for the hinge assembly.

Assembling the hinge was a straightforward process (Figure 5.2). After inserting the pin through one of the holes of the secondary hinge, the first torsion spring is inserted. The primary hinge must be inserted in a specific orientation in which one of the torsion spring legs rests on the primary hinge's central plate. The second leg is pointed downward such that it would rest upon the CubeSat's wall during integration. By extending the pin and slightly bending it, the final torsion spring was inserted.

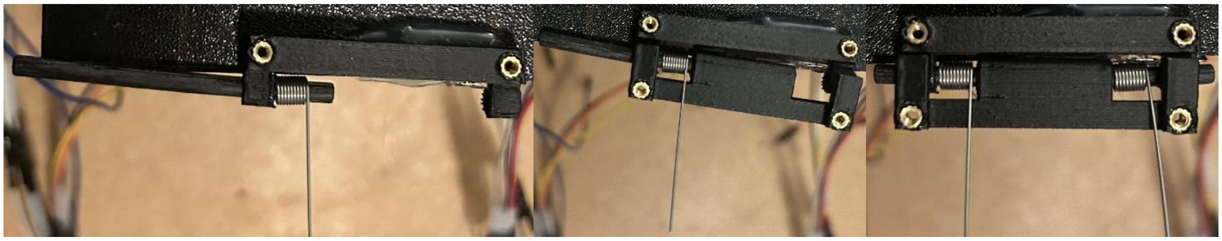


Figure 5.2. Hinge assembly process.

The solar panel backplate was attached to the holes of the primary hinge via M2 x 8 mm screws. In the stowed configuration, the torsion spring legs are compressed by the solar panel backplate (Figure 5.3). Once the fishing string is cut, the hinge will deploy the solar panels.

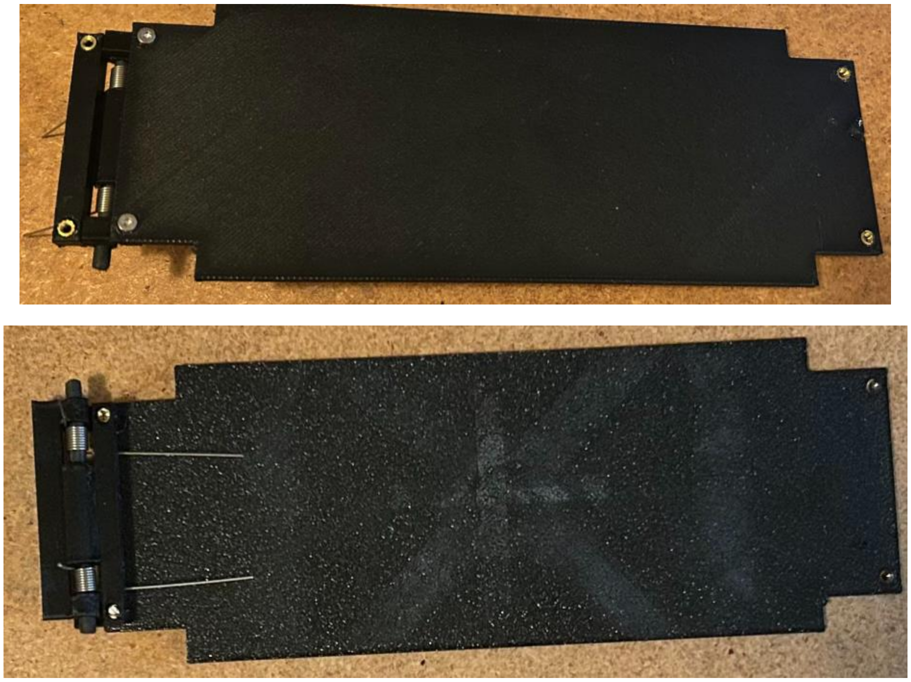


Figure 5.3. Single-set torsion spring hinge and solar panel assembly top and bottom surface, respectively.

The torsion spring hinge assembly successfully deflects the solar panel upwards when the applied load is released (Figure 5.5). The achieved deflection seemed to be low as if the solar array assembly's mass was too large. A double-set torsion spring hinge was created to improve performance. The original primary hinge CAD file was modified to accommodate these additional springs: the central plate was reduced by six mm in both directions. The assembly process (Figure 5.4) was identical to the single-set torsion spring design. Once assembled, the double-set torsion spring achieved maximum deflection; the blockers prevented the backplate from overextending. Figure 5.5 shows the maximum deflection and deflection at rest for both designs. The double-set design outperforms the single-set design in terms of deflection achieved and redundancy. In space torsion springs only experience friction forces as a counter moment. It is probable that a solar array with the single-set design will still reach full deployment. The double-set design would then nearly guarantee full deployment capabilities—maximum deflection was achieved under friction, air resistance, and gravity—in space. The only apparent design is the additional mass by the torsion springs and the possibility of the torsion springs breaking the blocking elements.

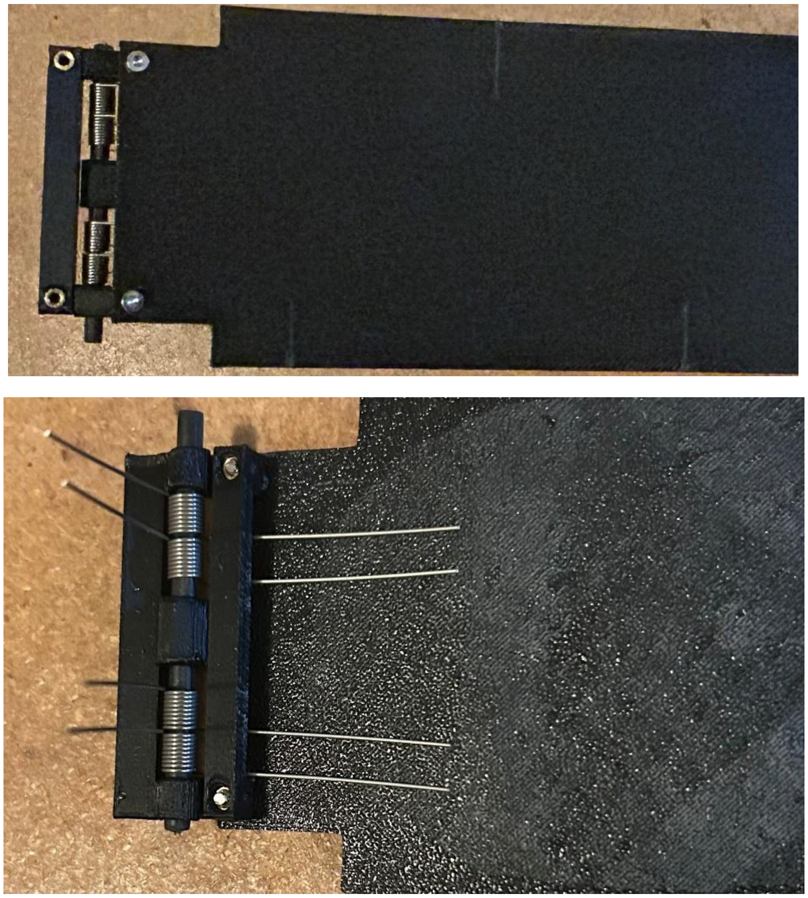


Figure 5.4. Double-set torsion spring hinge and solar panel assembly top and bottom surface, respectively.

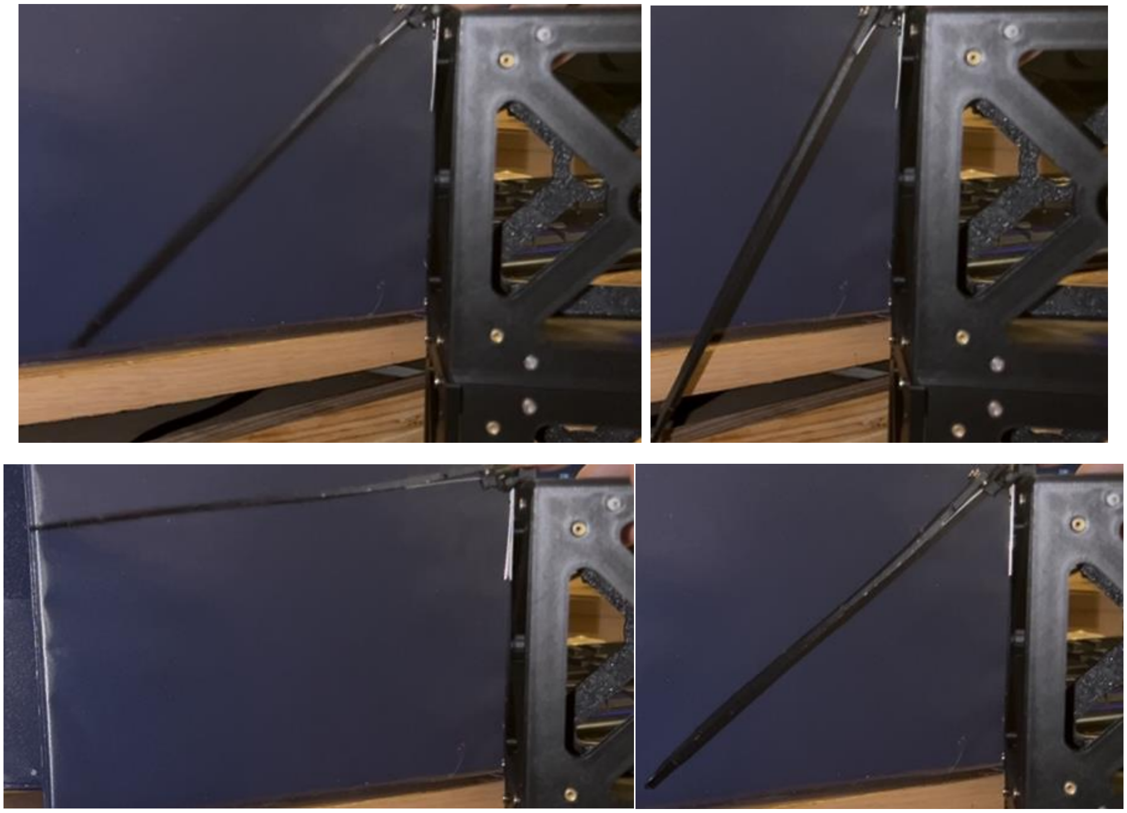


Figure 5.5. Single-set (top) and doublet-set (bottom) torsion springs. Maximum deflection (left) and deflection at rest (right).

With the torsion spring deployment test successful, the hinge was modified to allow attachment to a rotary disk via a M2 x 8 mm screw (Figure 5.6). Testing showed that the torsion spring legs would lose contact as the rotary disk spun. Subsequently, a fixed backplate was added to the secondary hinge to ensure that the torsion springs do not get stuck and that maximum torque would be generated (Figure 5.7). For prototyping purposes, the legs of the torsion spring were secured to the backplate with tape.

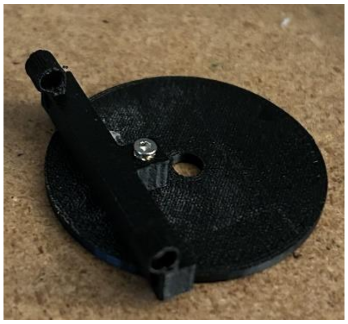


Figure 5.6. Rotary disk and secondary hinge part.

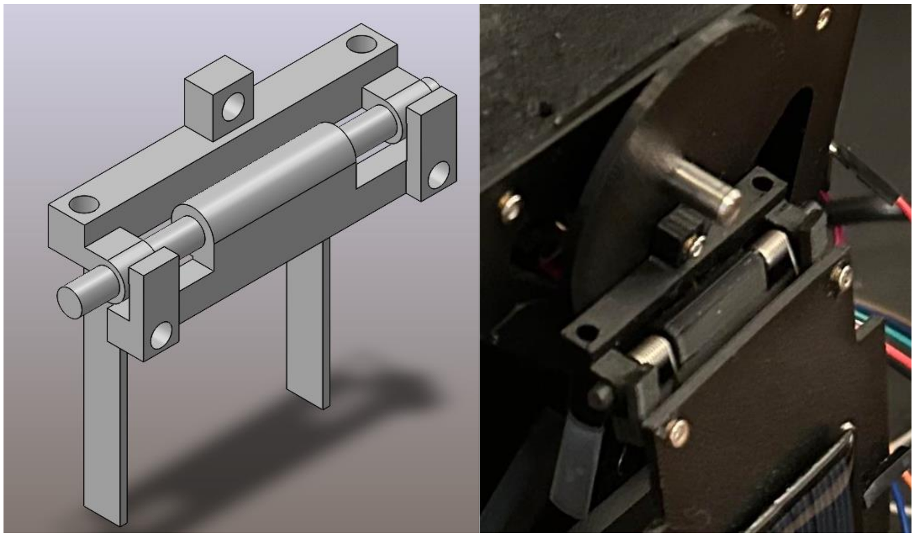


Figure 5.7. Finalized hinge design (left) and solar array attachment (right).

5.1.3 Motor Assembly

NEMA 17 motors were initially selected

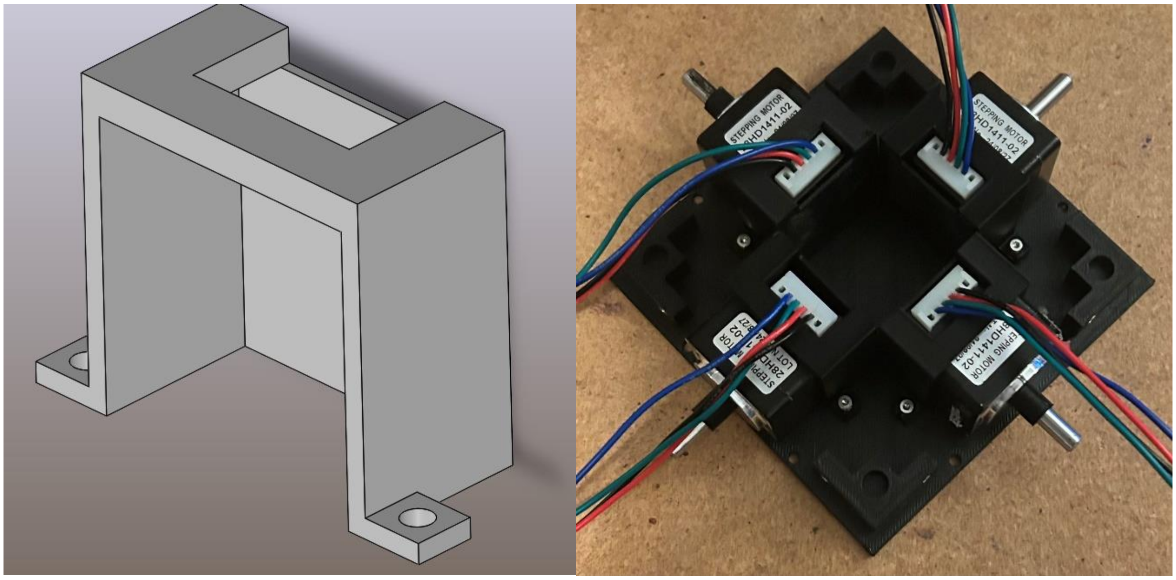


Figure 5.8. NEMA 11 stepper motor bracket.

5.1.4 Integration-Level Assembly

Figure 5.9 shows the entire structural assembly with no attached electronics. The baseplate was initially designed with no holes for screw insertion. The dimensions of the primary hinge were obtained and marked on tape attached to the baseplate’s surface. Dots were used to indicate the location of the hinge’s holes. A drill bit smaller than that of M2 was used to create eight through all holes for ease of complexity. Only one of the hinges perfectly aligned with the baseplate inserts. M2 x 10 mm screws were applied to the set aligned with the inserts. These screws were applied to one of the holes for each of the misaligned hinges. Since this is a prototype, hot glue was applied to the bottom surface of the misaligned primary hinges. This process ensured that the hinges were not slanted off of the baseplate. The solar panels were attached to the top baseplate and solar panel backplates via hot glue (Figure 5.10). All wires that were supplied with each solar panel disconnected during component-level and integration testing, which lead to additional wires being resoldered.

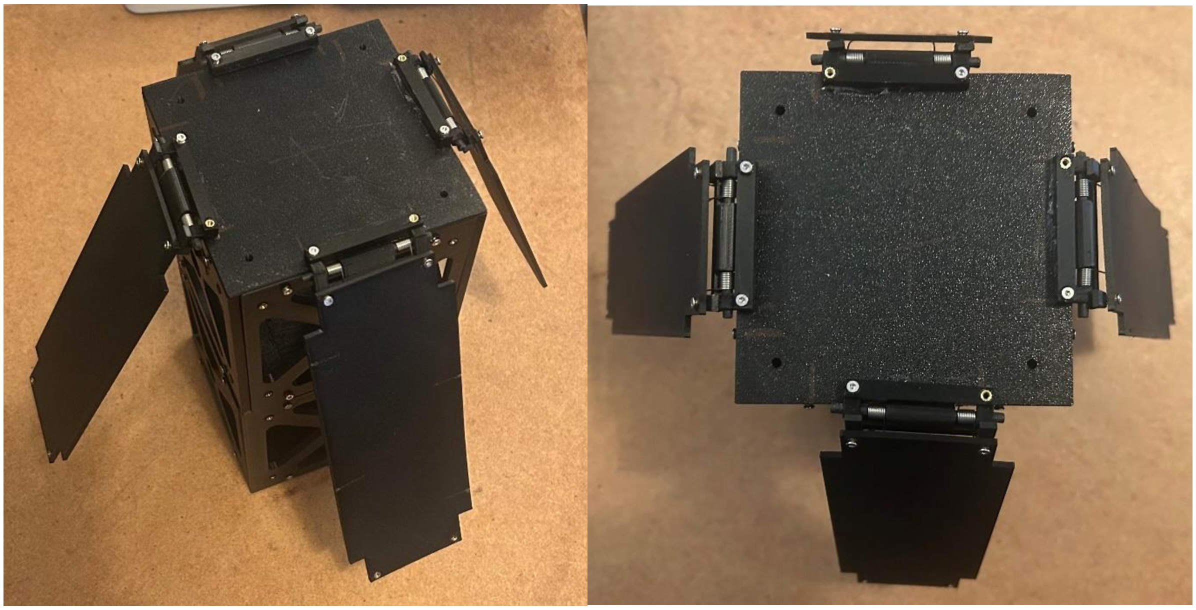


Figure 5.9. Deployed solar panel backplates.

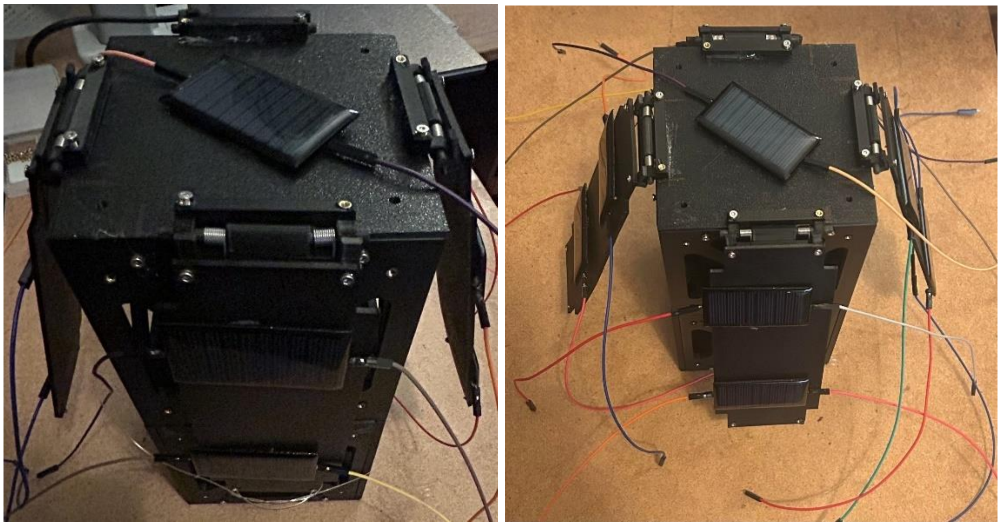


Figure 5.10. One stowed solar panel and deployed solar panels.

Fishing string was used to secure the solar panel array in the stowed state. Each end of the fishing string was inserted into the M2 hole of the solar panel backplate (Figure 5.11). On the other side of the backplate, each line was wrapped around the X-cross section of the chassis wall and back outside. The thread that started on the right is now resting on the left side of the solar panel backplate (Figure 5.12). After the fishing line is crossed (Figure 5.12), the line that is under the other line goes top bottom into the hole. Similarly, the line that is above the other line is brought bottom top. By pulling the lines in opposite directions, the knot shown in Figure 5.10 and Figure 5.12 was obtained. The burn resistor was wrapped the two bottom fishing string. It is important to note that only two of the three fishing string lines are responsible for retention. These two lines are the ones obtained at the point where the fishing string is crossed (Figure 5.12). The points past this are solely for tensioning the retention lines.

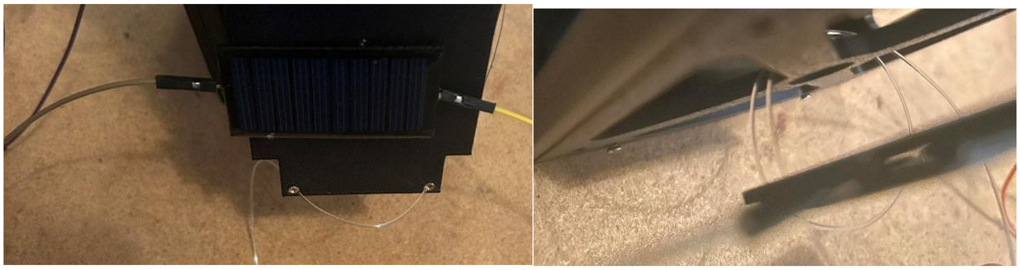


Figure 5.11. Initial fishing line insertion (left) and X-cross section wrapping (right).

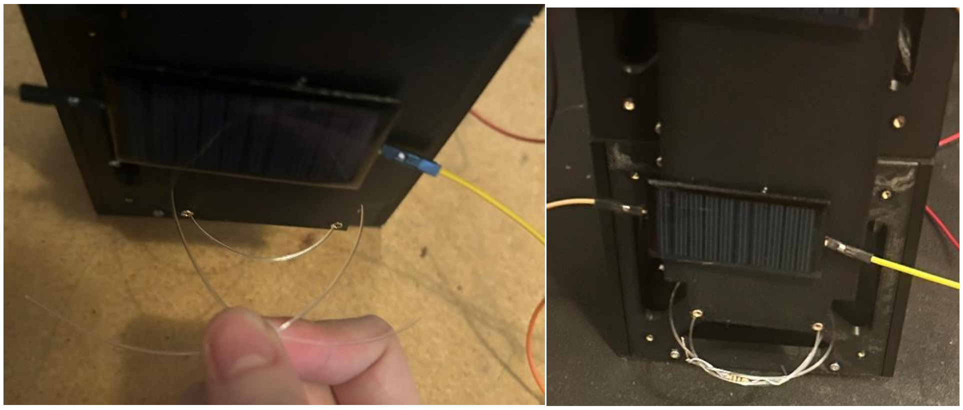


Figure 5.12. Fishing line crossing (left) and end result (right).

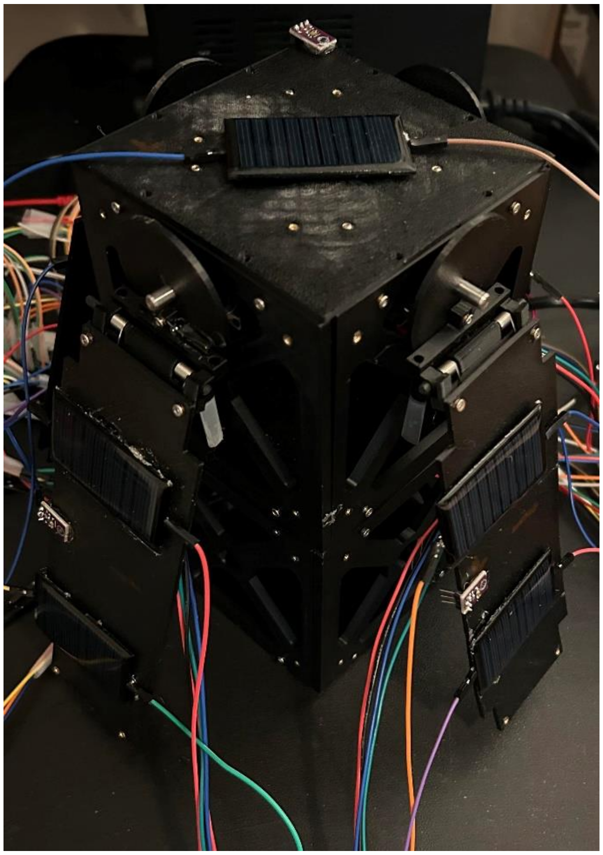


Figure 5.13. Rotatable solar array.

5.2 Electrical Subsystem Assembly

5.2.1 Component-Level Testing

The purpose of component-level testing is to identify any sensors that are faulty due to improper soldering or software issues. Component-level testing was applied to every light sensor (TEMT6000), motor encoder (AS5600), and the voltage and current sensor (INA219). This process is crucial to prevent any single-point failures and to minimize performance loss in the stepper motor PID control system. Initially component-level testing was performed within the Arduino IDE. Once each component was verified, the Arduino code for each component was rewritten in MATLAB for data logging purposes.

5.2.1.1 Light Sensor (TEMT6000)

Each TEMT6000 sensor was verified to be functional by applying all the wire connections shown in Figure 5.14 and observing the light value outputted by the Arduino serial monitor. Under ambient light conditions, the light sensor outputted an average ADC value of 50. When the light sensor was covered the light sensor outputted an average ADC value of 10. Once the light sensor was uncovered, the light sensor outputted similar ADC values that were obtained initially. By shining a flashlight, an average ADC value of 1000 was obtained. Thus, this test indicates that each TEMT6000 sensor was verified to be fully functional under all light conditions.

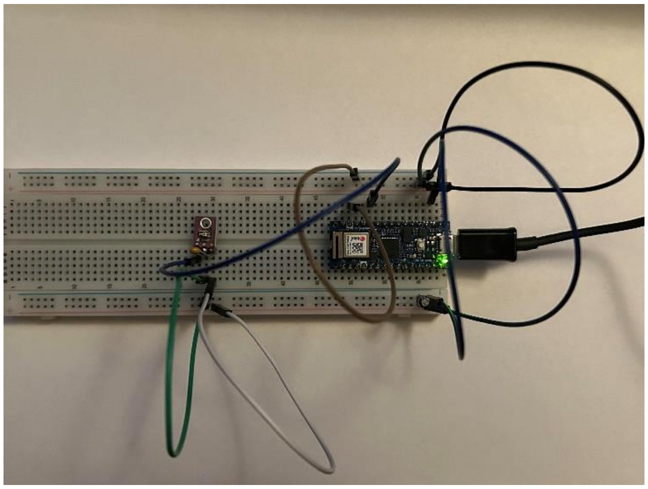


Figure 5.14. TEMT6000 component test.

The component test was recreated in MATLAB for further data analysis. Figure 5.15 shows the ADC readings—1920 datapoints were taken—of the TEMT6000 under ambient, covered, and light-exposed states. By examining the long-term behavior, it is evident that TEMT6000 experiences substantial noise and spikes in its readings; Figure 5.16 shows these observations more prominently.

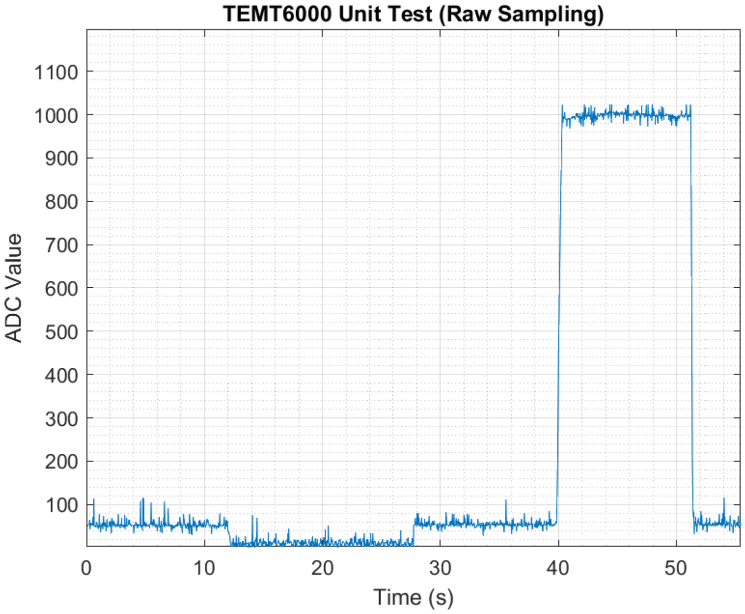


Figure 5.15. TEM6000 component test under ambient, covered, and light-exposed conditions.

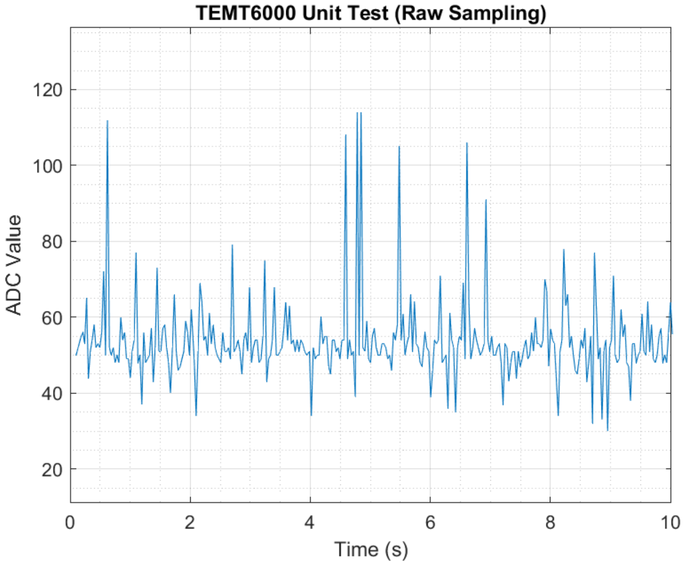


Figure 5.16. TEM6000 readings under ambient light conditions.

The TEMT6000's performance was improved by averaging 50 raw sampling values over a 1.9 second interval to reduce noise. Prior to averaging, the minimum and maximum values were omitted to mitigate the influence of data outliers. Based on Figure 5.18, it is evident that this methodology reduced and eliminated TEMT6000 ADC value noise and spikes, respectively. All averaged ADC values were within ± 5 of each other (Figure 5.17). Figure 5.19 shows the long-term TEMT6000 response under ambient, covered, and light-exposed conditions. At the transition phases—such as ambient to covered and covered to ambient—it takes the sensor a maximum of 1.9 seconds to update to the appropriate light environment (Figure 5.19). Considering that the position of the sun will not drastically change the time it takes to reach steady-state conditions is not of great importance. Subsequently, the sampling time interval could be increased to further reduce noise.

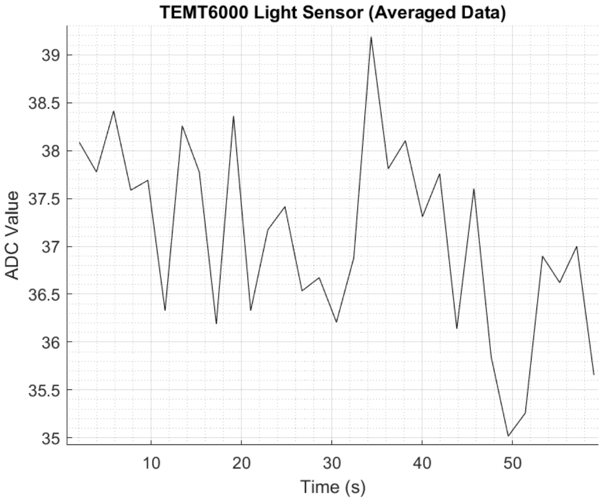


Figure 5.17. TEMT6000 averaged readings under ambient light conditions.

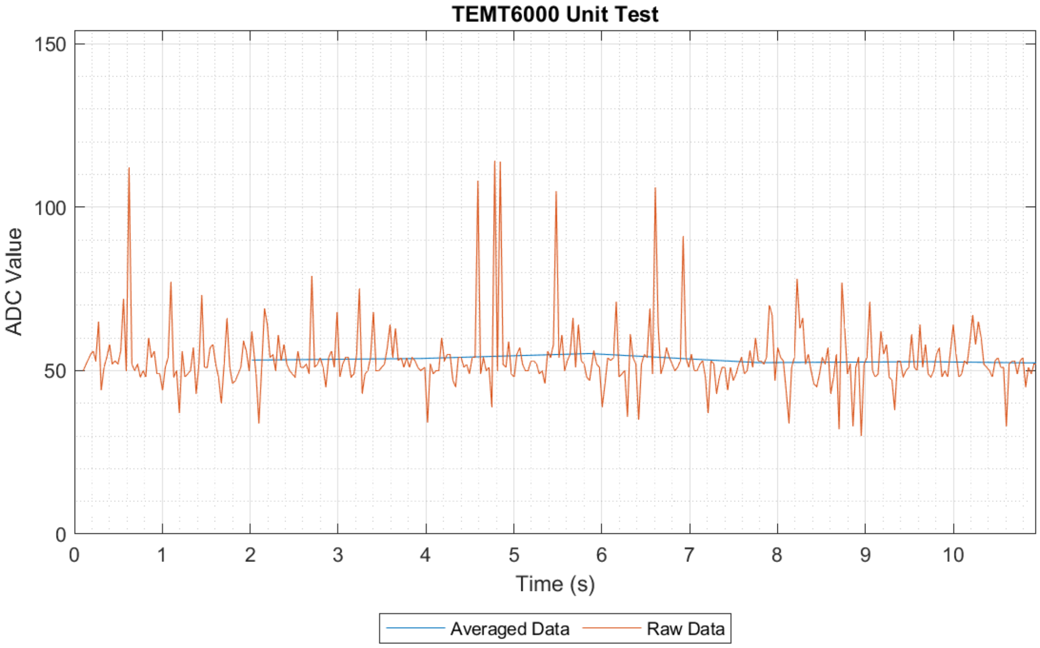


Figure 5.18. Raw and averaged TEMT6000 readings under ambient light conditions.

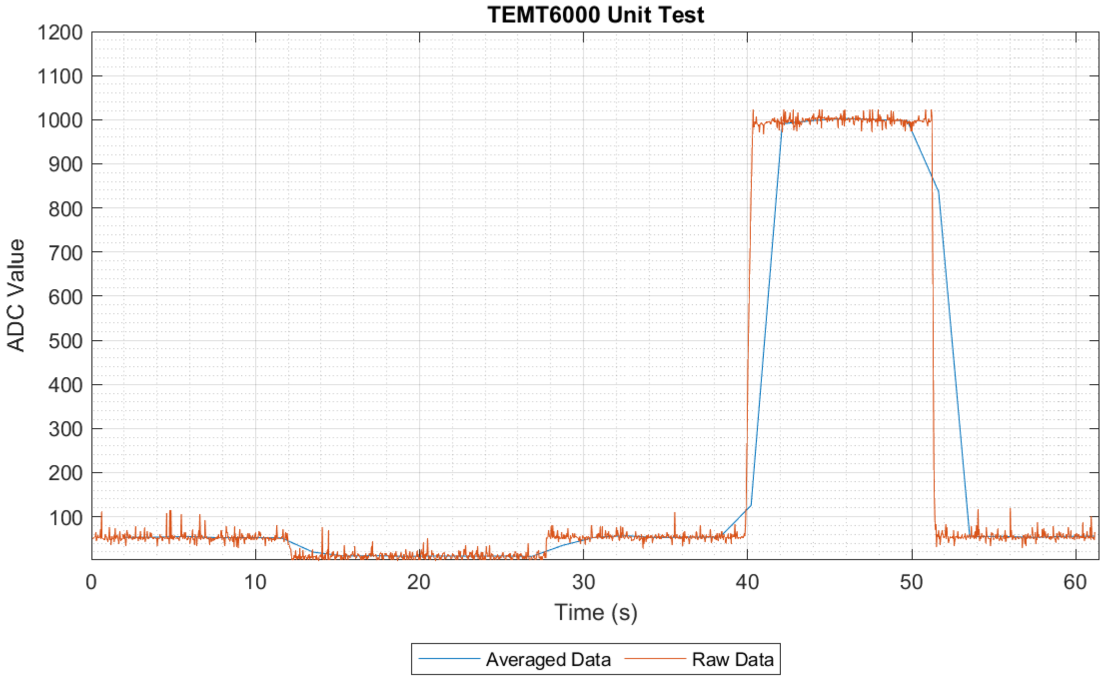


Figure 5.19. Raw and averaged TEMT6000 readings of the component test.

5.2.1.2 Motor Encoder (AS5600)

The motor encoder was verified by using an example Arduino library and rotating the magnet above the chip. Figure 5.21 shows the stepper motor position and angular position (°) of the

AS5600. By rotating the magnet clockwise, the angular position increased. The converse occurred when the magnet was rotated counterclockwise. Component testing successfully showed that the encoder could detect the presence of the magnet and its current angular position.

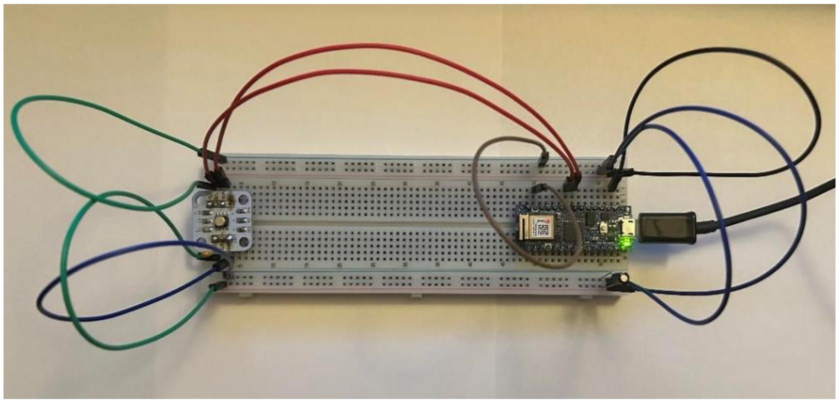
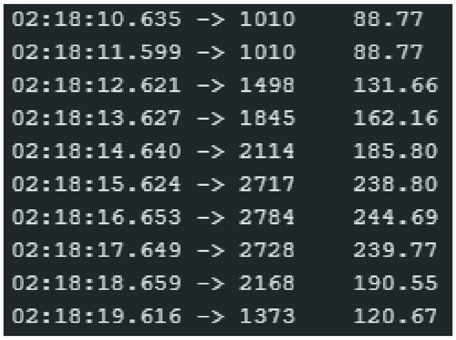


Figure 5.20. AS5600 component test.



```
02:18:10.635 -> 1010    88.77
02:18:11.599 -> 1010    88.77
02:18:12.621 -> 1498    131.66
02:18:13.627 -> 1845    162.16
02:18:14.640 -> 2114    185.80
02:18:15.624 -> 2717    238.80
02:18:16.653 -> 2784    244.69
02:18:17.649 -> 2728    239.77
02:18:18.659 -> 2168    190.55
02:18:19.616 -> 1373    120.67
```

Figure 5.21. Stepper motor position (left) and angular position (°) (right).

5.2.1.3 Voltage and Current Sensor (INA219)

The INA219 sensor was verified by using an example Arduino library and changing the ambient light delivered to the solar panel. The relevant wire connections are provided by Figure D.2. Initially, the sensor outputted 0V when a flashlight was shown on the solar panel. This should not be the case because the voltage measured by a multimeter was 1.321V. Decreasing the load had no influence on the sensors output. When the load resistor was increased to 2.3 kΩ, the sensor outputted a bus voltage of 40 mV. When increased to 10 kΩ, the bus voltage increased to 70 mV. In an open circuit—the load resistor was disconnected—the sensor outputted a voltage of 1.321 V as expected. When the flashlight was shone on the solar panel, the voltage reading matched the 2.942 V of the multimeter. In complete darkness, however, the bus voltage reading was 600 mV. This is greatly above the threshold for electrical noise or imprecise measurements; the cause for this is unknown. Figure 5.23 shows a snippet of the results from testing.

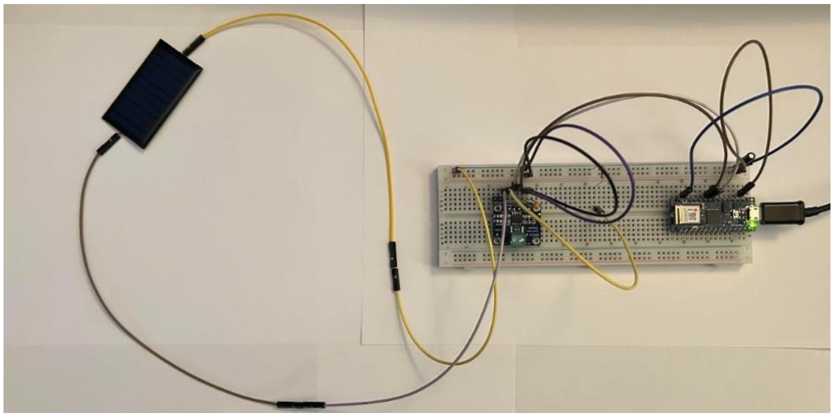
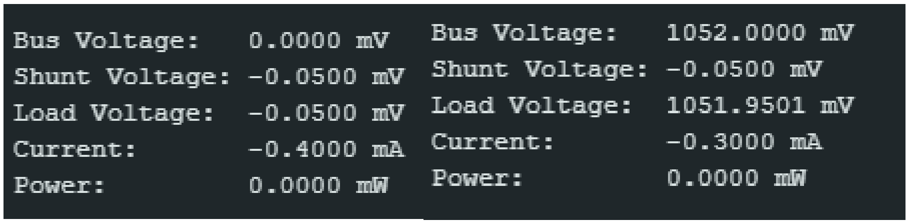


Figure 5.22. INA219 component test.



```
Bus Voltage:   0.0000 mV      Bus Voltage:   1052.0000 mV
Shunt Voltage: -0.0500 mV     Shunt Voltage: -0.0500 mV
Load Voltage:  -0.0500 mV     Load Voltage:  1051.9501 mV
Current:       -0.4000 mA     Current:       -0.3000 mA
Power:         0.0000 mW      Power:         0.0000 mW
```

Figure 5.23. Load resistor applied (left) and load resistor removed (right).

5.2.2 Integration-Level Testing

Integration-level testing is performed after each component was verified to operate successfully. This level ensures that each component can successfully function together and to identify any uncalibrated or skewed sensors.

5.2.2.1 Light Sensor Array

Five light sensors were connected to the analog multiplexer. By setting the bits of each channel of the multiplexer to high or low, the multiplexer could select a particular light sensor and read its value. This is achieved via a sequential process. Figure 5.25 shows the light sensor readings relative to V_{CC}. As shown by the results on the left side, there is no major discrepancy between each light sensor. The discrepancies could be explained by the jumper wires partially obscuring the light sensors. After continued monitoring, a few anomalies have appeared. For the results right of Figure 5.25, one of the light sensors have experience an intensity drop compared to the other sensors. Thus, it is necessary to sample the sensor hundreds of times—this is achieved within milliseconds—and to take the average of a particular sensor. By repeating the component-level methodology, the light sensor arrays outputted accurate results.

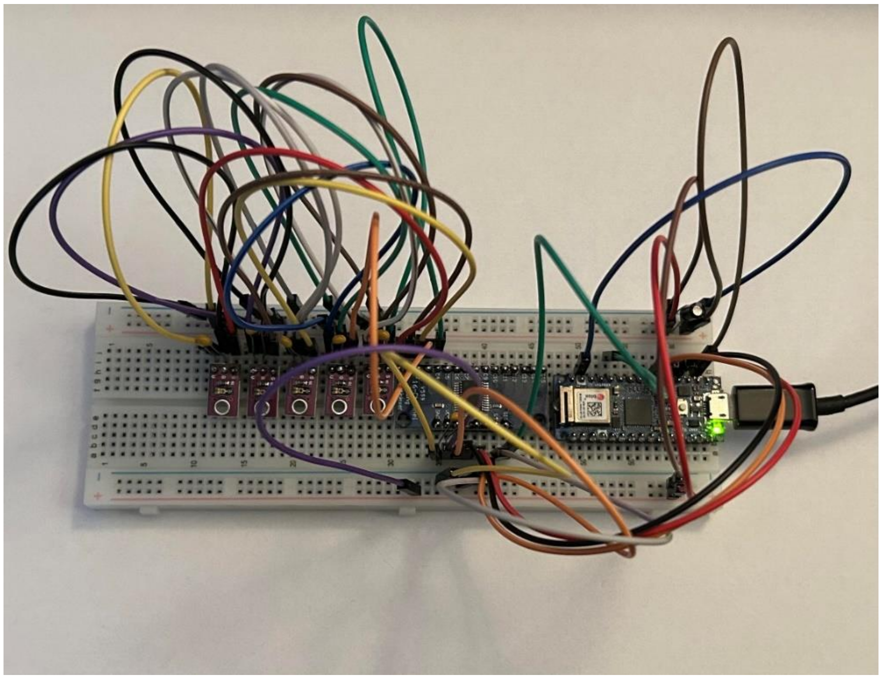
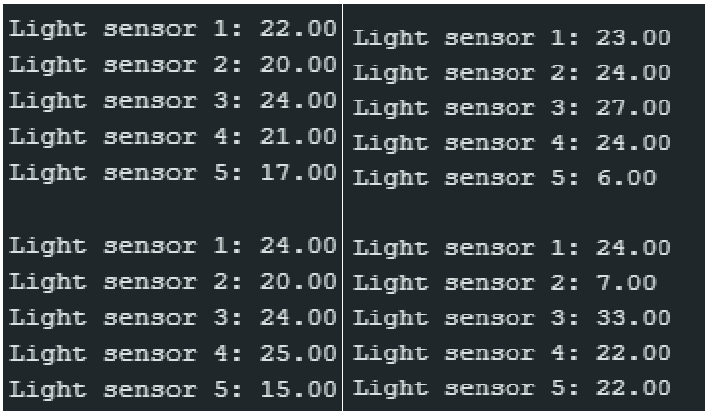


Figure 5.24. TEMT6000 array and CD74HC4067 integration test.



```
Light sensor 1: 22.00
Light sensor 2: 20.00
Light sensor 3: 24.00
Light sensor 4: 21.00
Light sensor 5: 17.00

Light sensor 1: 24.00
Light sensor 2: 20.00
Light sensor 3: 24.00
Light sensor 4: 25.00
Light sensor 5: 15.00
```

```
Light sensor 1: 23.00
Light sensor 2: 24.00
Light sensor 3: 27.00
Light sensor 4: 24.00
Light sensor 5: 6.00

Light sensor 1: 24.00
Light sensor 2: 7.00
Light sensor 3: 33.00
Light sensor 4: 22.00
Light sensor 5: 22.00
```

Figure 5.25. Nominal (left) and anomalies (right) light sensor readings relative to V_{cc}.

5.2.2.2 Motor and Motor Driver

The DRV8825 motor driver was tuned to ensure that the motor does not draw more than 0.8 A. This was done by setting the reference voltage of the DRV8825 to 0.4 V by varying the onboard potentiometer. The motor successfully moved clockwise and counterclockwise via Arduino commands.

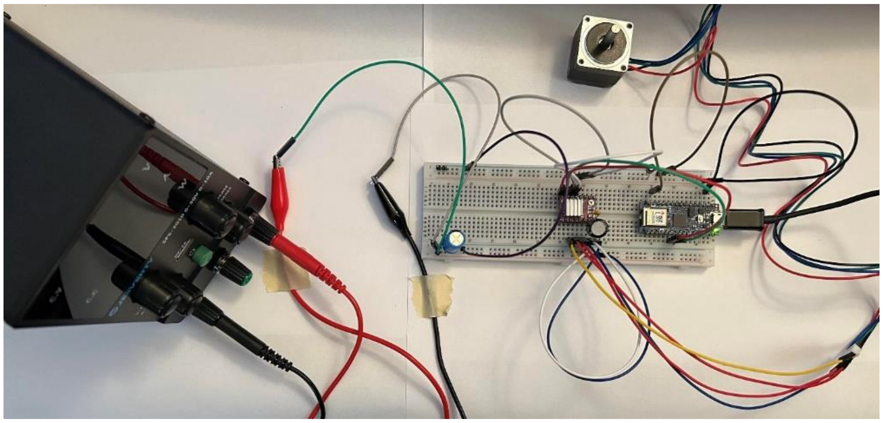


Figure 5.26. DRV8825 and NEMA 11 stepper motor integration test.

5.2.2.3 Voltage Sensor Array

Similar to that of the analog multiplexers, the digital multiplexers uses I2C to obtain the voltage readings of each sensor. Since there are two multiplexers, the first multiplexer was set to an I2C address of 0x70 and the second to 0x71 by varying the states of A0 – A2 of each multiplexer. A program was written to ensure that the Arduino could successfully select both of the multiplexers and that the multiplexers could reach each of its channels. The voltage sensor array was successful.

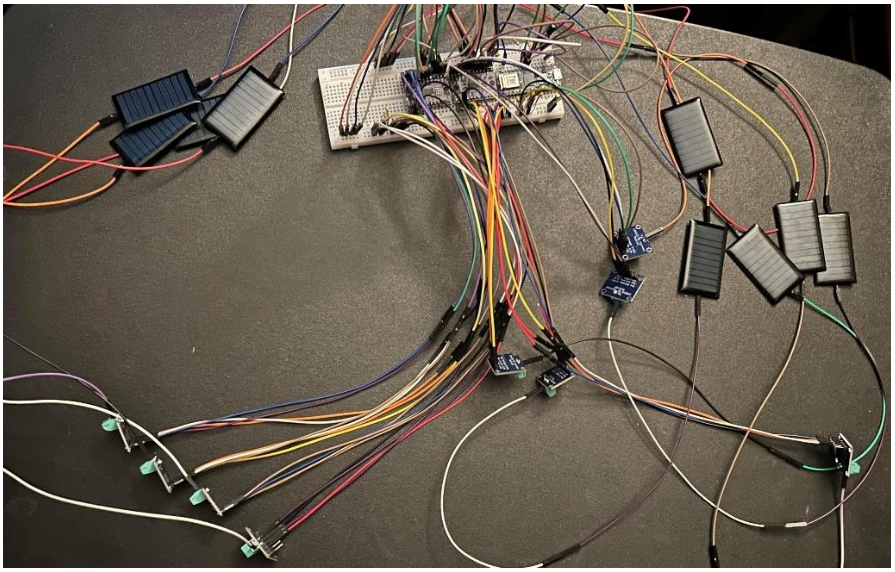


Figure 5.27. Solar panel array, INA 219, PCA9548A, and CD74HC4067 integration test.

5.2.3 System Level Test

System level testing of the light sensor array, NEMA 11 motors, and current sensors showed that each component successfully operated together.

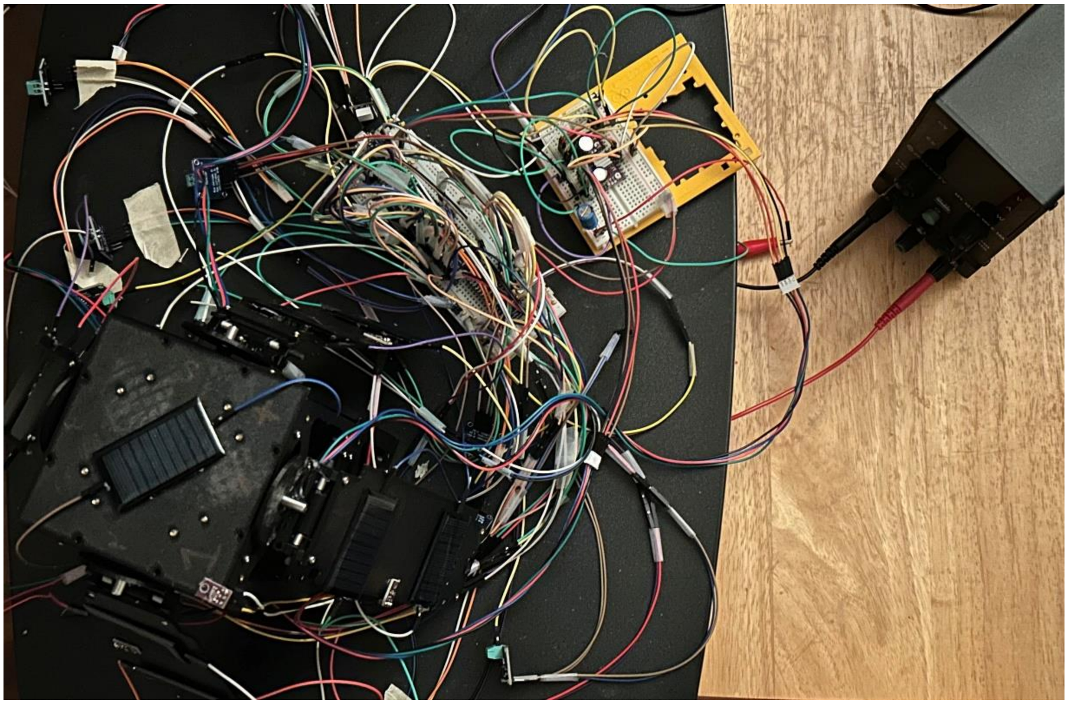


Figure 5.28. System level test wiring.

5.2.4 Burn-Wire Assembly

A burn-wire system was designed and built to cut through the fishing string to deploy the solar panels. A 10 V power and 10 Ω resistor would result in 1 A being delivered to the resistor; this is sufficient to melt through the fishing string. When designing a burn-wire system, the resistor should last long enough to cut through the wire before it thermally fails. Figure 5.29 shows the solar panel retained by the fishing string and the solar panel being deployed by the burn-wire system. Figure 5.29 - Figure 5.34 shows the burn wire process from burn-wire initiation to solar panel deployment and burn resistor blowing out. One second after power supply activation the resistor starts heating up (Figure 5.30). At 1.074 A, the first retention wire is burned (Figure 5.31). A moment later the support wire is burned (Figure 5.32). As previously discussed, the support wire provides tensioning to the retention lines to aid in the stowage of the solar panels. All retention wires are burned (Figure 5.32) before the burn resistor blows out a few seconds later (Figure 5.34). This process shows that the retention system is sufficient in keeping the solar panels stowed and that the burn-wire could sufficiently burn through the fishing string. In total, five burn tests were conducted. Two of these tests contained different designs than the one showed in Figure 5.12. Both of these tests failed—Figure 5.35 shows one of these failures—due the resistor unwrapping from the fishing string and the resistor burning one side of the support line. Three burn-wire tests were conducted based on the knot shown in Figure 5.12. Out of these three tests, one failed due to cutting the support line. By securing the burn resistor to the solar panel and using two burn resistors for redundancy, the success rate would increase.

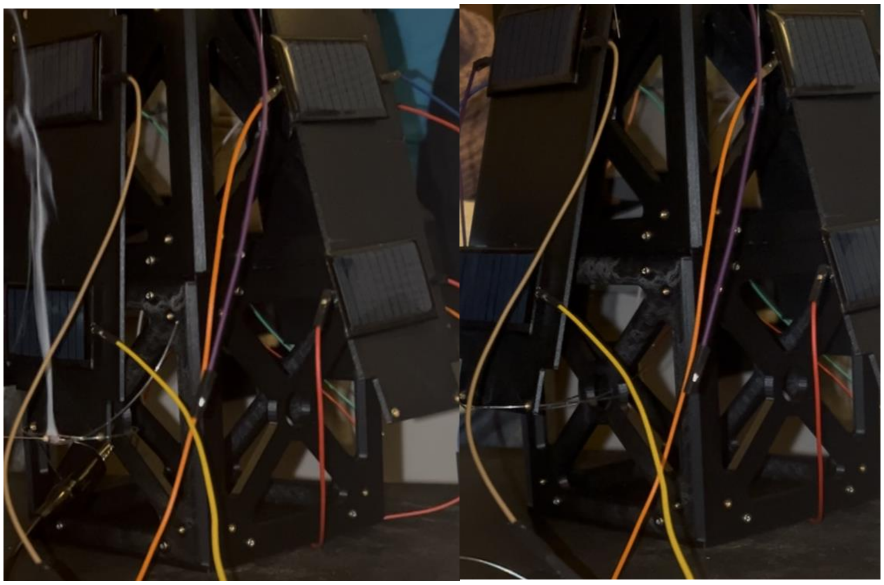


Figure 5.29. Stowed solar panels (left) and deployed solar panels (right)

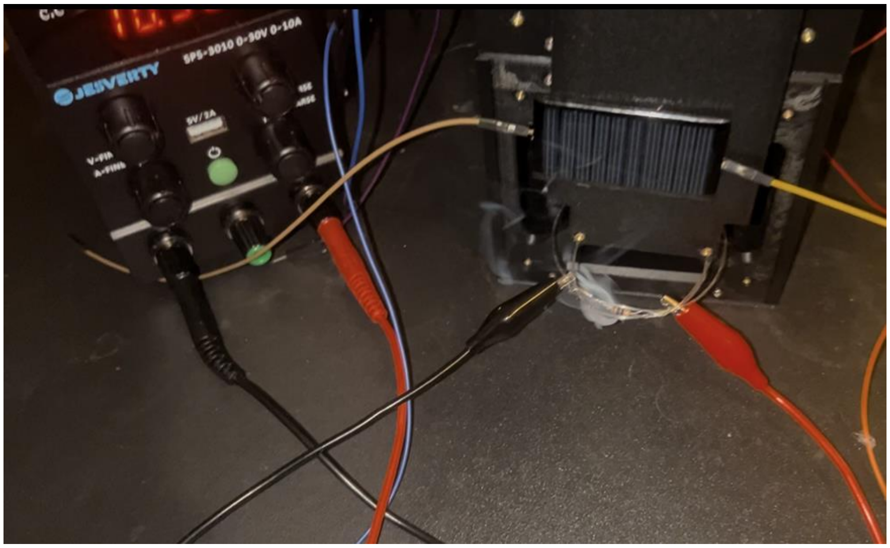


Figure 5.30. Burn wire process initiated.

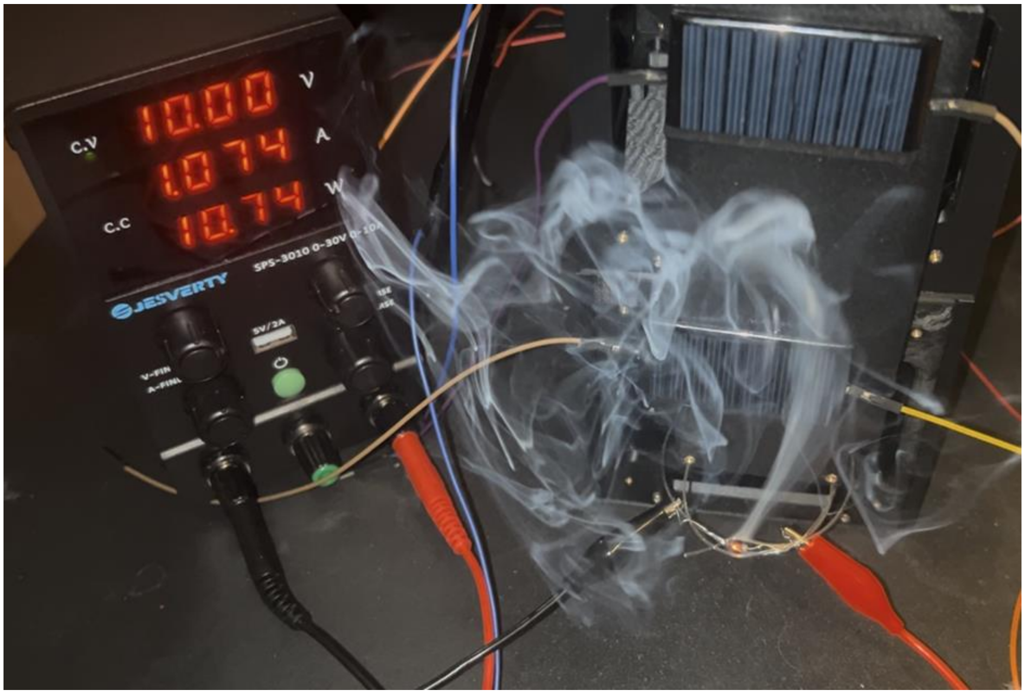


Figure 5.31. First retention wire burned.

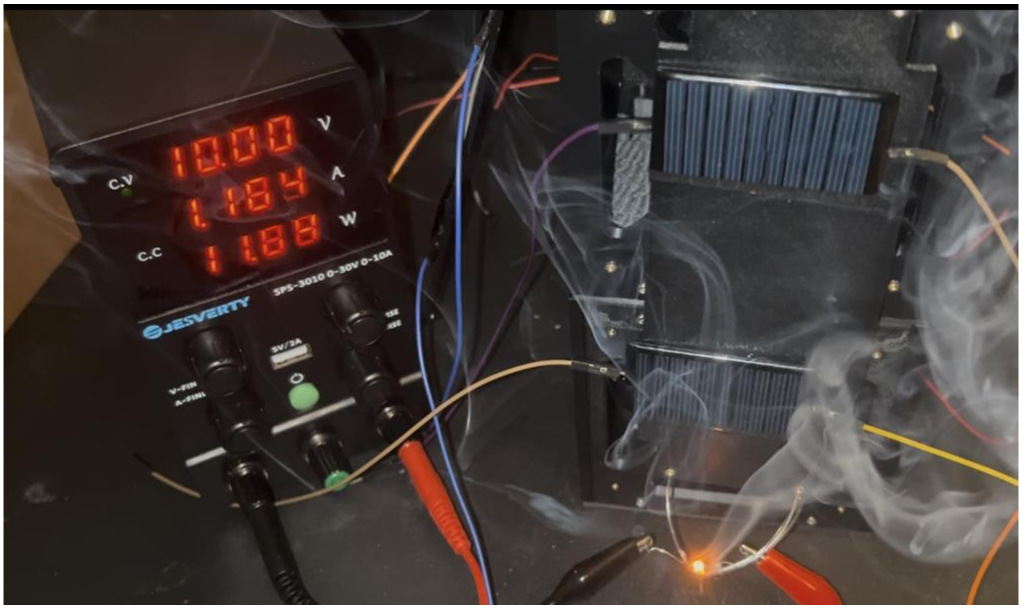


Figure 5.32. Support wire burned.

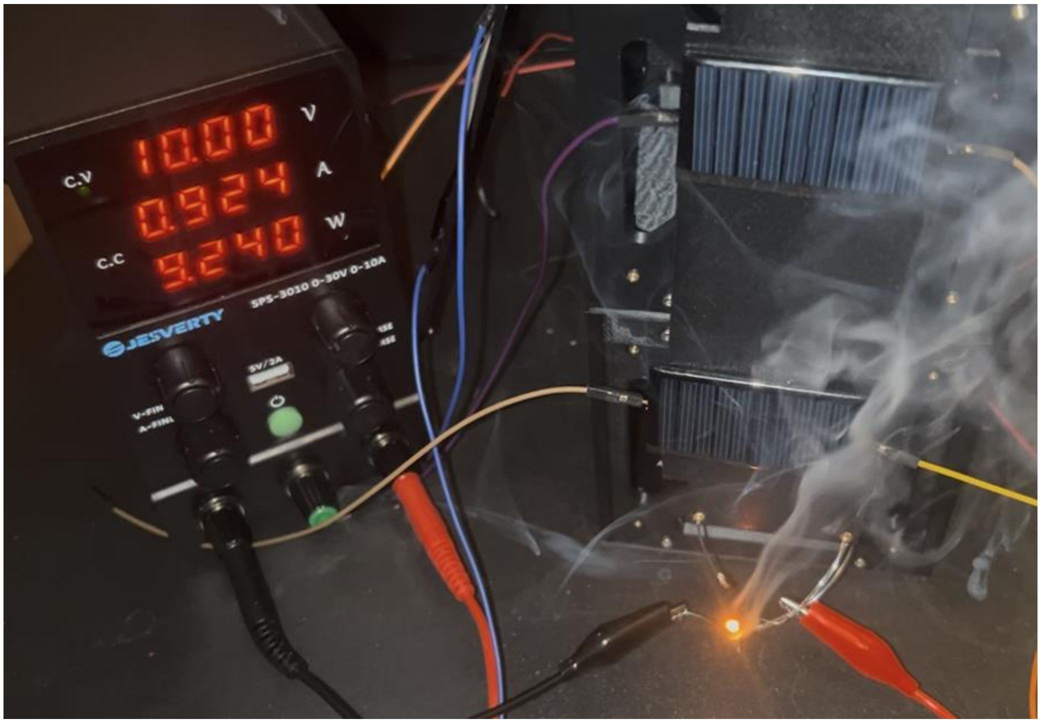


Figure 5.33. All retention wires burned (panel fully deployed).

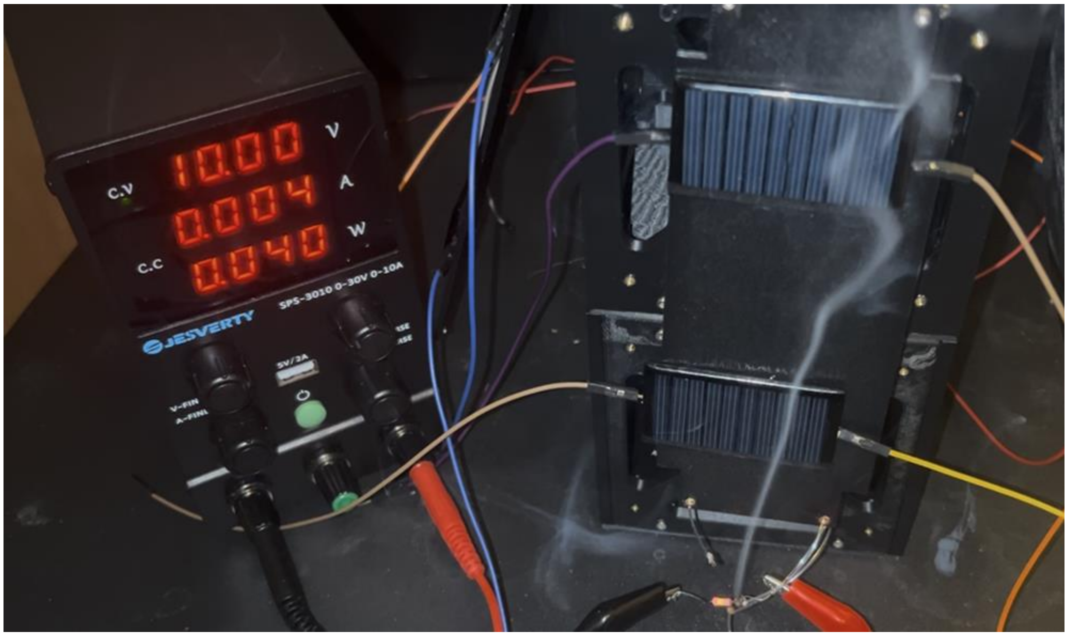


Figure 5.34. Blown resistor.

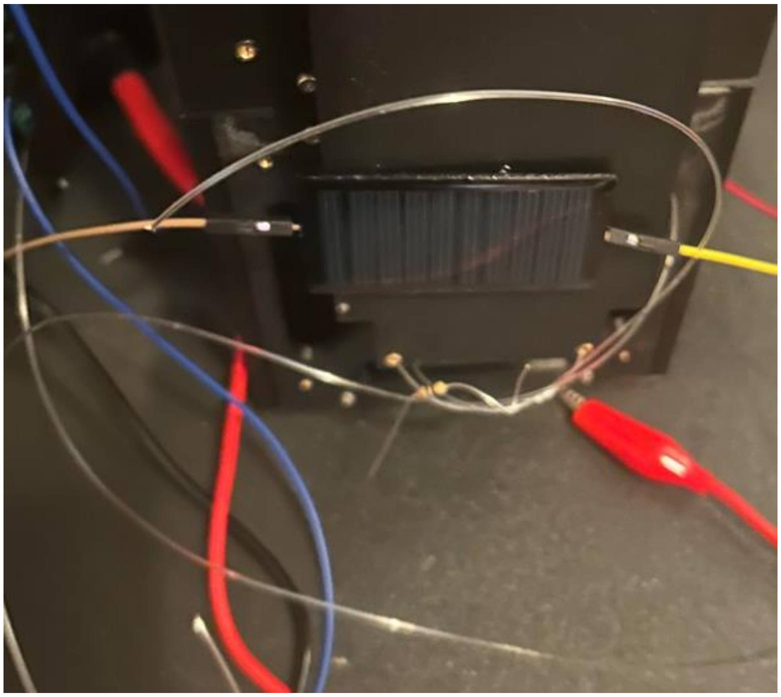


Figure 5.35. Failed burn-wire test.

6. Solar Panel Data

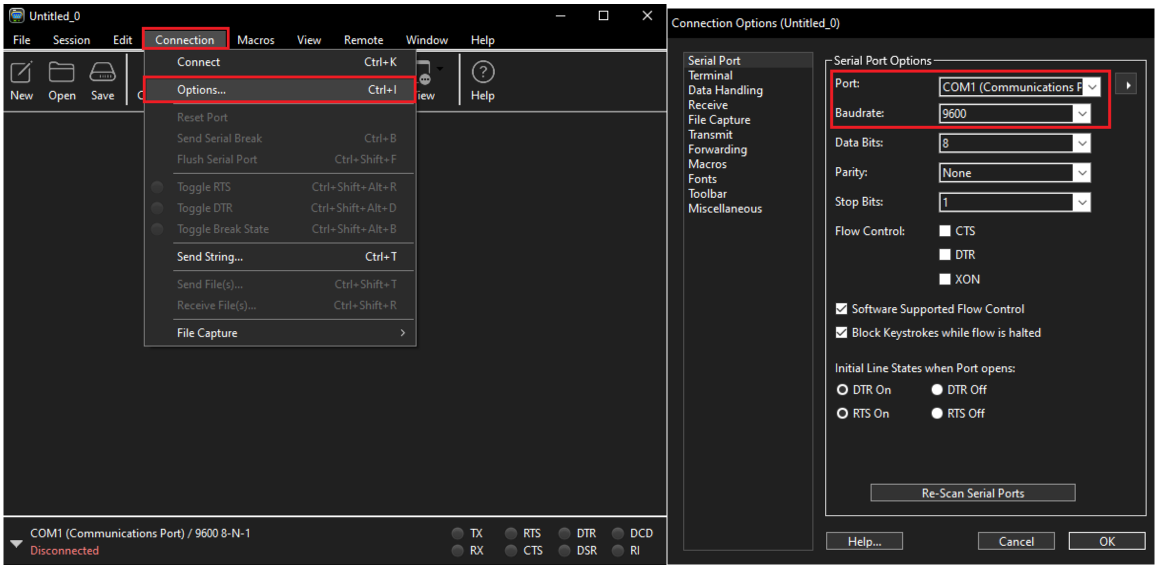
6.1 Methodology

6.1.1 Test Environment

Three solar panel configurations were tested: body-mounted, partially deployed, and fully deployed for the single-hinge configuration. The double-hinge and multi-hinge configurations were not tested due to limited project resources. Testing was conducted within an indoor test environment to ensure that the results remain consistent between each tested configuration. Testing was conducted at 10:00 PM to ensure that the ambient light level remains constant. For each test, the CubeSat remained in the same approximate position: 10 ft below and 2 ft away from a ceiling light source.

6.1.2 Data Collection

Although the Arduino IDE stores data in the serial monitor, this data is not persistent; old data will be erased after a certain amount of lines. Coolterm was used to capture and store the serial monitor data within a text file. Within the "Connection Options" window, the port was set to COM3 and the baudrate was set to 9600 (Figure 6.1). These selections reflect the port that the Arduino is connected to and its data transmission rate. Data was stored in a text file by selecting *Connection* > *File Capture* > *Start* (Figure 6.2). Selecting "Connect" begins the data collection process (Figure 6.3). After one hour of data collection, "Disconnect" was selected to end the data collection process. 8,731, 9,315, and 9,423 data points were collected for the stowed, partially deployed, and fully deployed solar arrays. The first 8,731 data points for the deployed configurations were compared with the stowed configuration data to maintain consistency. These data files were read and parsed in MATLAB. The data files were modified to ensure that the starting multiplexer and channel is M1C0. Additionally, the elapsed time was modified to ensure that each data set has the same start time.



Serial Port Options

Port: COM1 (Communications P

Baudrate: 9600

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: □ CTS □ DTR □ XON

☑ Software Supported Flow Control

☑ Block Keystrokes while flow is halted

Initial Line States when Port opens:

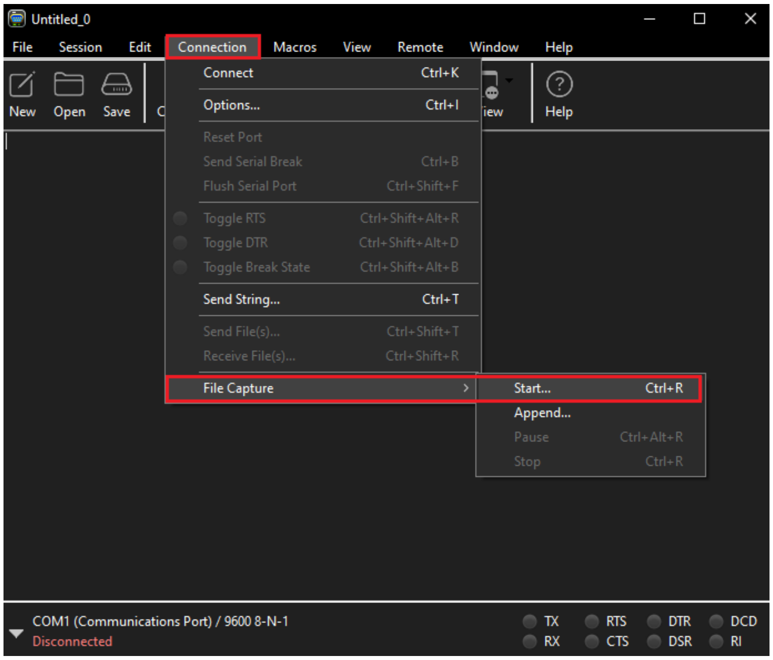
○ DTR On ● DTR Off

○ RTS On ● RTS Off

COM1 (Communications Port) / 9600 8-N-1

Disconnected

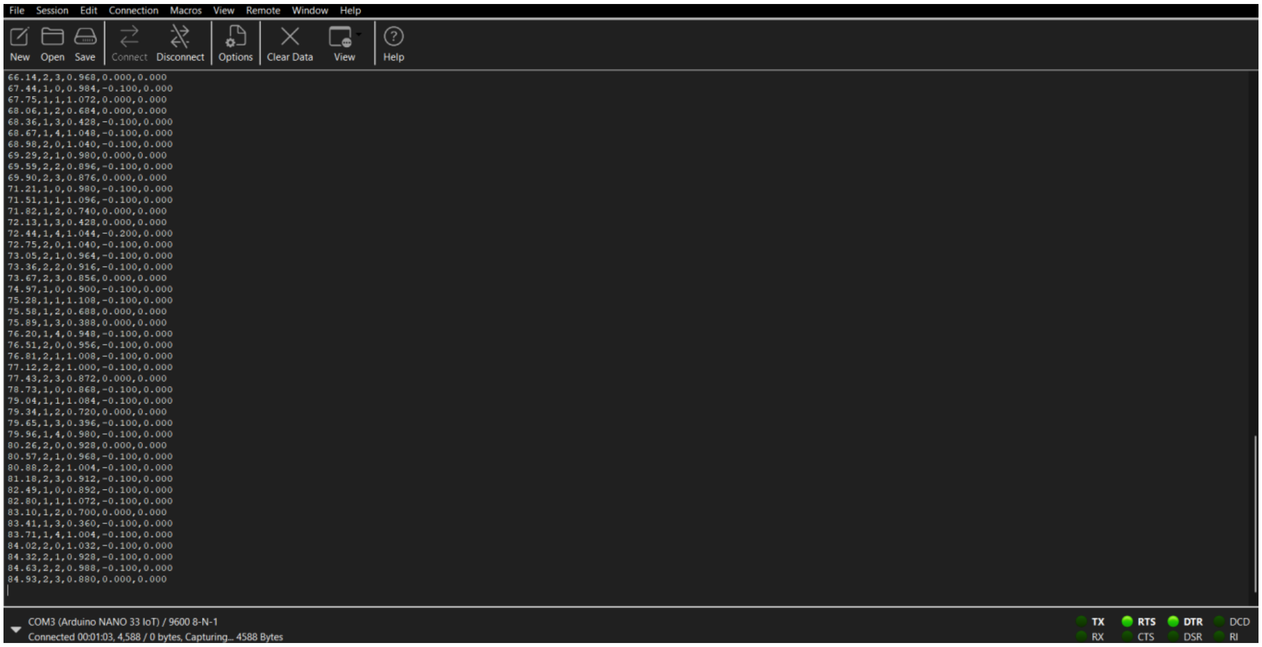
Figure 6.1. Coolterm port and baudrate selection.



COM1 (Communications Port) / 9600 8-N-1

Disconnected

Figure 6.2. Stores data in a text file.



```
66.14,2,3,0.968,0.000,0.000
67.44,1,0,0.984,-0.100,0.000
67.75,1,1,1.072,0.000,0.000
68.06,1,2,0.684,0.000,0.000
68.36,1,3,0.428,-0.100,0.000
68.67,1,4,1.048,-0.100,0.000
68.98,2,0,1.040,-0.100,0.000
69.29,2,1,0.980,0.000,0.000
69.59,2,2,0.896,-0.100,0.000
69.90,2,3,0.876,0.000,0.000
71.21,1,0,0.980,-0.100,0.000
71.51,1,1,1.096,-0.100,0.000
71.82,1,2,0.740,0.000,0.000
72.13,1,3,0.428,0.000,0.000
72.44,1,4,1.044,-0.200,0.000
72.75,2,0,1.040,-0.100,0.000
73.05,2,1,0.964,-0.100,0.000
73.36,2,2,0.916,-0.100,0.000
73.67,2,3,0.856,0.000,0.000
74.97,1,0,0.900,-0.100,0.000
75.28,1,1,1.108,-0.100,0.000
75.58,1,2,0.688,0.000,0.000
75.89,1,3,0.388,0.000,0.000
76.20,1,4,0.948,-0.100,0.000
76.51,2,0,0.956,-0.100,0.000
76.81,2,1,1.008,-0.100,0.000
77.12,2,2,1.000,-0.100,0.000
77.43,2,3,0.872,0.000,0.000
78.73,1,0,0.868,-0.100,0.000
79.04,1,1,1.084,-0.100,0.000
79.34,1,2,0.720,0.000,0.000
79.65,1,3,0.396,-0.100,0.000
79.96,1,4,0.980,-0.100,0.000
80.26,2,0,0.928,0.000,0.000
80.57,2,1,0.968,-0.100,0.000
80.88,2,2,1.004,-0.100,0.000
81.18,2,3,0.912,-0.100,0.000
82.49,1,0,0.892,-0.100,0.000
82.80,1,1,1.072,-0.100,0.000
83.10,1,2,0.700,0.000,0.000
83.41,1,3,0.360,-0.100,0.000
83.71,1,4,1.004,-0.100,0.000
84.02,2,0,1.032,-0.100,0.000
84.32,2,1,0.928,-0.100,0.000
84.63,2,2,0.988,-0.100,0.000
84.93,2,3,0.880,0.000,0.000
```

COM3 (Arduino NANO 33 IoT) / 9600 8-N-1
Connected 00:01:03, 4,588 / 0 bytes, Capturing... 4588 Bytes

Figure 6.3. Data monitoring window.

6.2 Results

As previously discussed, there were not enough SCL and SDA pins for the INA219 current and voltage sensors. This required the INA219 sensors to be connected to the PCA9548A digital multiplexer. The MCU sequentially read each INA219 sensor connected to the PCA9548A with a 0.3 second delay between each channel. It took the multiplexer 3.76 seconds to loop back to the starting channel. In MATLAB, the data was categorized into stowed, partially deployed, and fully deployed sets. These sets were further subcategorized into individual channels. These channels represent the data of each individual solar panel. Figure 6.4 shows the multiplexer and channel classification of each solar panel. Figure 6.6 - Figure 6.10 shows the voltage generated of each solar array. It is important to note that a moving average was applied every three points to smooth out the noise and eliminate voltage spikes. There was minimal difference between the raw and averaged data sets (Figure 6.5). Lastly, the difference between the "Deployed Panel" and "Fully Deployed Panel" is characterized by Figure 5.9 and Figure 2.11. In which the former refers to a partially deployed panel under Earth's gravity and the latter refers to a simulated fully deployed panel; the latter maintains the maximum deflection angle.

Figure 6.6 shows that the voltage generated by the stowed and deployed array are similar in magnitude; generally, the deployed array generated a greater voltage. Towards the end of the test the deployed array generated less voltage. This voltage drop should be ignored because the array was momentarily obscured during testing. The cause of the voltage drop between 1,000 – 1,500 seconds is unknown. Considering that solar array 1 was oriented away from the light source, these results meet expectations. Although the array was deflected by 20°, this deflection was not substantial enough for there to be a significant increase. However, once the solar array was deflected to 90° in the fully deployed configuration, the solar array generated an additional 0.4 V.

Figure 6.8 and Figure 6.7 shows that there is a substantial voltage difference between the stowed and deployed configurations. On average, the deployed configuration generated an additional 0.1 V. There were instances where the deployed configuration generated an additional 0.2 – 0.3 V. As for the fully deployed configuration, it is evident that this configuration generated 0.2 V less than the stowed configuration, which should not be the case. After the stowed and deployed tests were completed, the rotary disks and modified hinges were developed and assembled. During integration to the CubeSat and during system testing, a few of the solar panel wires were disconnected. Consequently, it is possible that too much heat was applied to the solar panel during the soldering process. Therefore, the fully deployed data set should be ignored.

Figure 6.9 fulfills all solar array expectations. The stowed, deployed, and fully deployed configurations of solar panel 5 generated 0.7 V, 0.75 V, and 1.05 V, respectively. Similarly, solar panel 6 generated 0.63 V, 0.69 V, and 0.95 V, respectively. Expectations were fulfilled because larger solar array deflections should generate more voltage. Furthermore, because solar panel 6's incidence angle was greater than solar panel 5, more voltage should be generated. Figure 6.7 fulfills these expectations for the first 3,000 seconds. The cause of the voltage drop experienced by solar panel 1 in the deployed configuration is unknown.

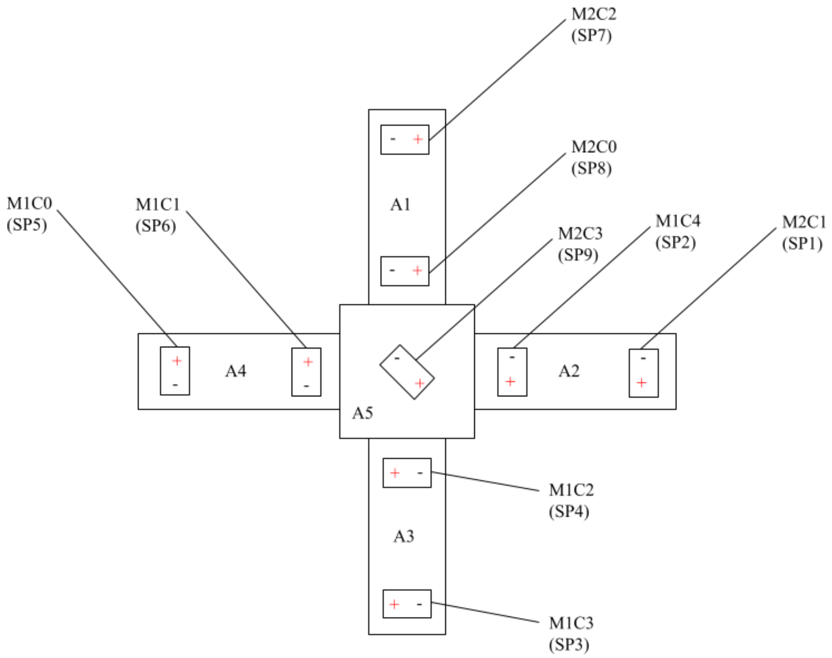


Figure 6.4. Multiplexer channel classifications for each solar panel.

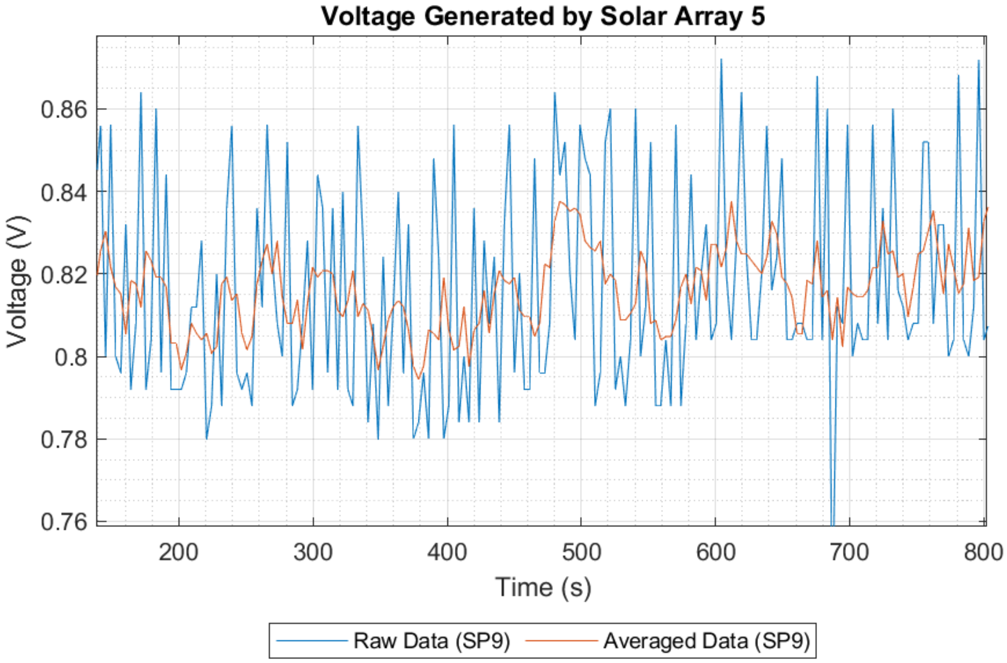


Figure 6.5. Raw and averaged data comparison for the voltage generated by solar array 5.

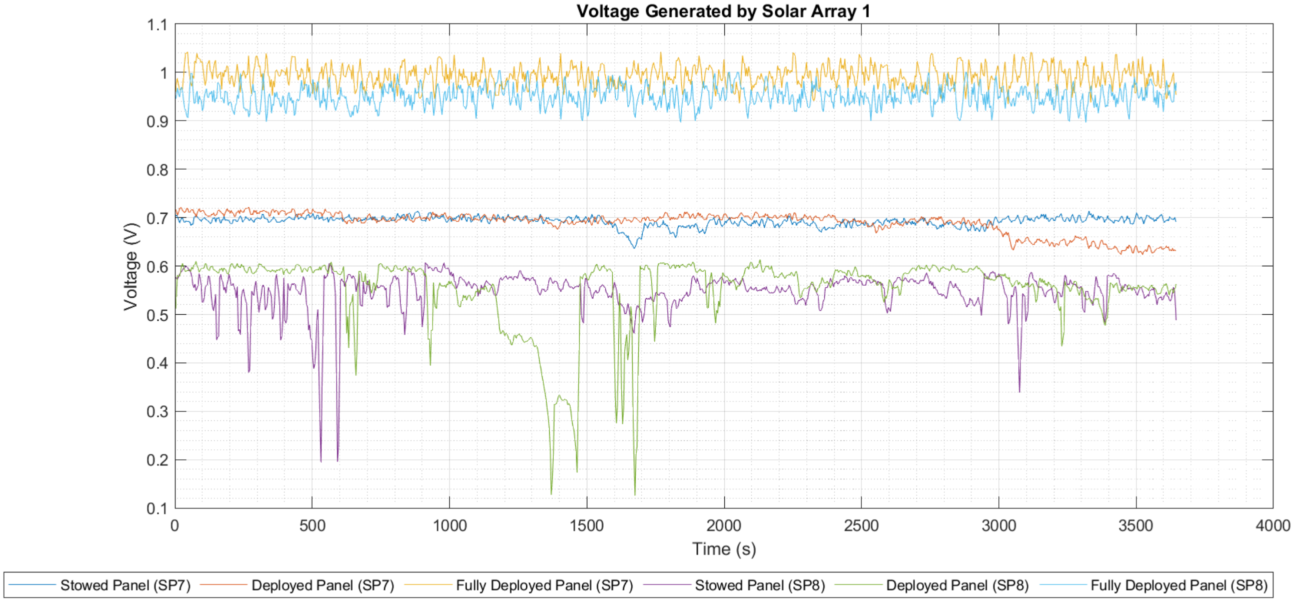


Figure 6.6. Voltage generated by solar array 1.

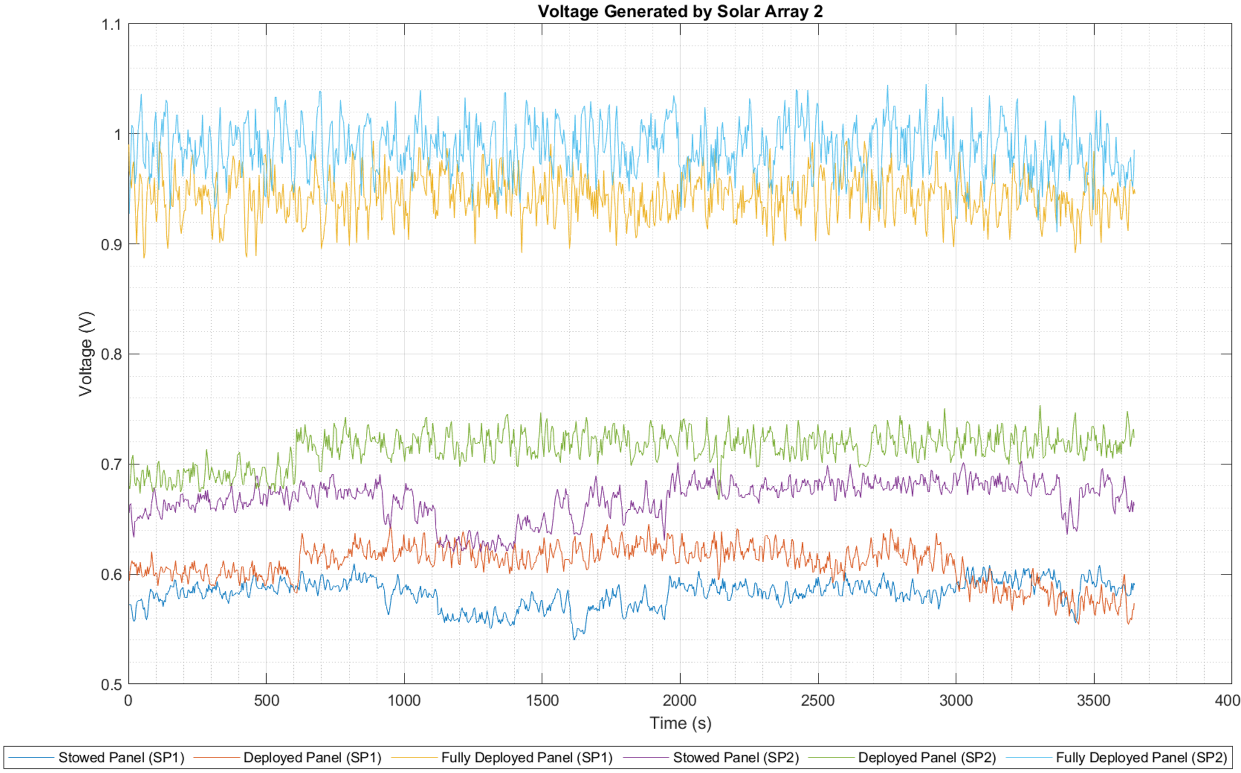


Figure 6.7. Voltage generated by solar array 2.

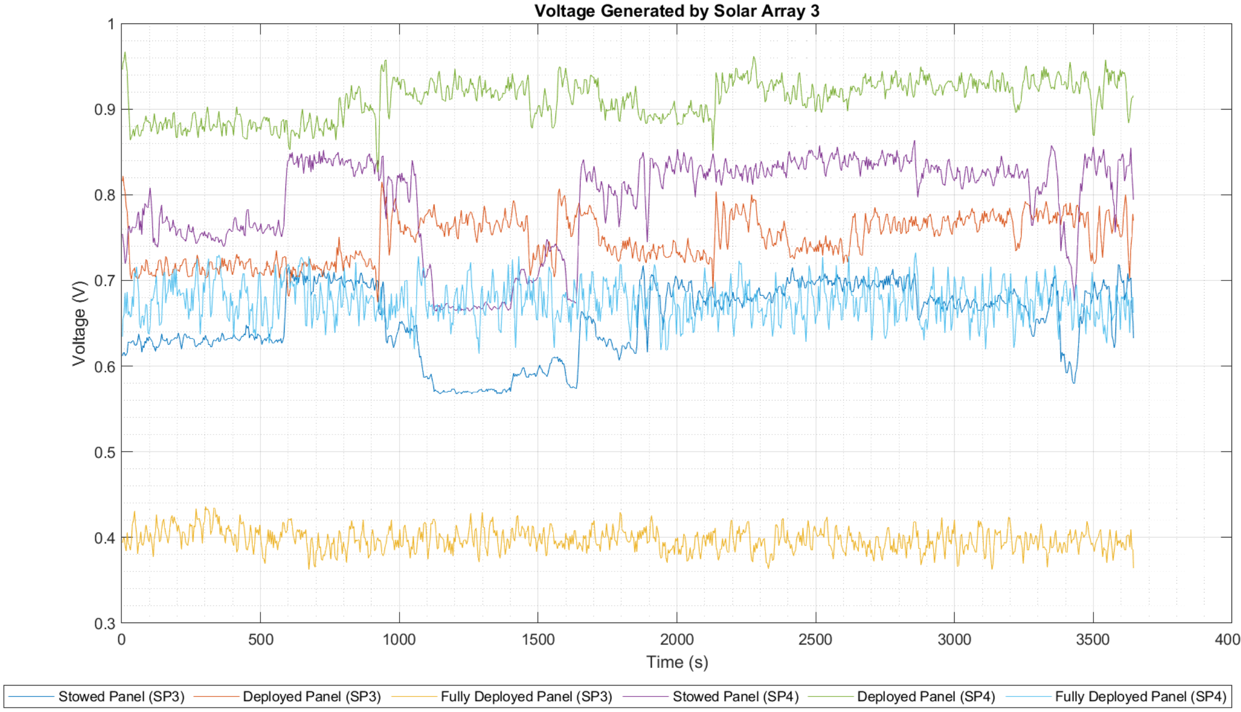


Figure 6.8. Voltage generated by solar array 3.

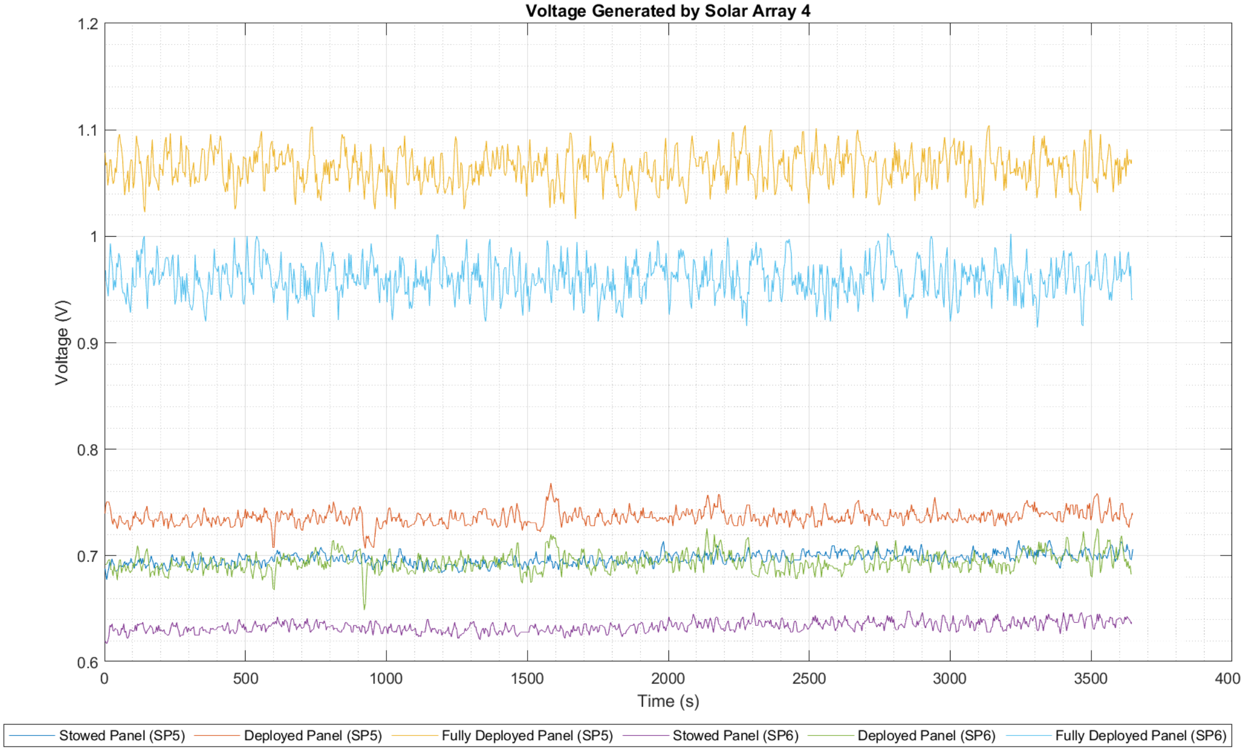


Figure 6.9. Voltage generated by solar array 4.

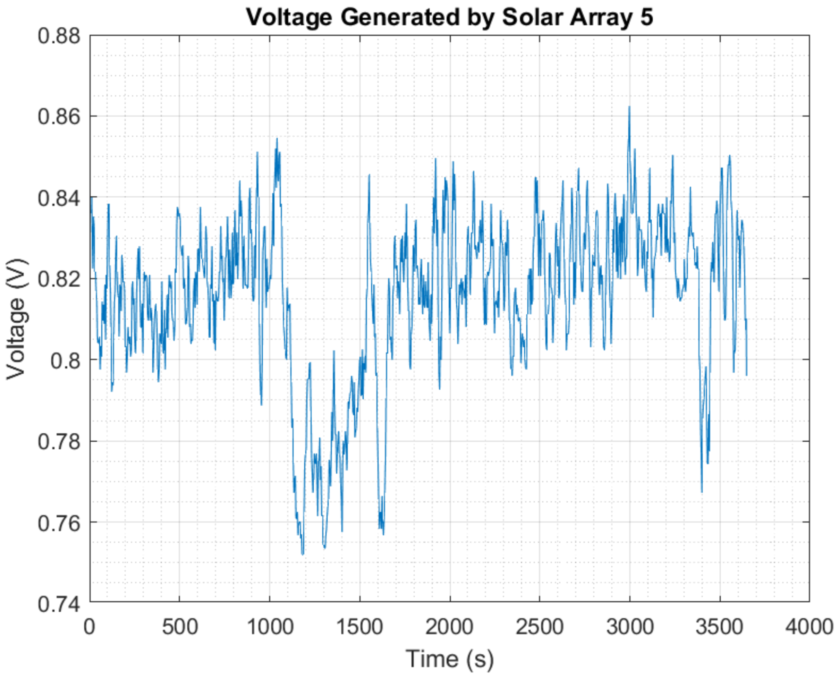


Figure 6.10. Voltage generated by solar array 5.

7. Conclusion and Recommendations

The underlying objective of this project was to design and develop a low-cost SADM with sun-tracking capabilities and axis controllability by utilizing a system of motors. An array of TEMT6000 ambient light sensors was used to enable light tracking capabilities. These sensors were placed on the side surfaces of the CubeSat, with an additional one being placed at the top surface. Since it was assumed that attitude control would be enabled by an external ADCS system, it was assumed that the bottom surface of the CubeSat faces Earth. During trade studies, NEMA 11, 14, and 17 stepper motors were considered. Although the NEMA 17 stepper motor offers the greatest torque capabilities, its downside is that it is the largest out of the three. To maximize the available internal volume of the CubeSat, four $18 NEMA 11 motors were selected. Implementing an encoder in the control system was deemed essential to create a closed-loop control system. However,

Initially, a system of two NEMA 17 stepper motors was considered, where one motor is responsible for the initial deployment of the solar array and the second is responsible for providing one-axis controllability. As depicted by the preliminary sketch of Figure 1.6 and the prototype SADM of 4.13, one motor is attached to a rotary disk. This motor controls solar array rotation about the axis normal to a chassis wall. The second motor, which is mounted on top of this rotary disk, controls initial deployment. No retention mechanism is necessary because the holding torque of the motors would keep each solar array in the desired orientation. An issue with this design, however, is that only one solar cell could be controlled due to the size of the motors, motor mounts, and rotary disks. Furthermore, the lever arm did not have sufficient volume to fully deploy the solar array. This eventually led to the consideration of a spur-gear and rail-block deployment system. However, these systems too suffered from volume and lever-arm limitations. This issue could be resolved by using smaller motors attached to each hinge. Although these motors exist, the availability of these motors is limited, and their costs are out of the project's budget. Consequently, a two-axis motor control SADM is not possible due to these limitations.

Instead of using an entirely motorized SADM, the SADM was downsized to permit controllability about the axis normal to each chassis wall surface. Rather than use motors for the initial solar array deployment, hinges driven by torsion springs were considered and developed. Hinges driven by a single and double set of torsion springs were tested; thermal analysis of torsion spring exposure within the environment should be conducted primarily on the possibility of thermal deformation or thermal failure points. The single set design deflected the solar arrays 45°, whereas the double set ensign deflected the solar arrays 90° under the influence of Earth's gravity and air resistance. The design of the hinge assembly prevented the solar array from deflecting past 90°. In the space environment, the single-set design will likely achieve full deflection because the dominating forces are primarily the friction forces between the pin and parts of the hinge. Although the blockers prevented the solar array from overextending, a future FEA analysis should be conducted to evaluate whether the blockers will fracture during contact. In this system, the solar array is stowed by fishing string and is released by a burn-resistor system. With burn-resistor tests successful, the rotary disk for the NEMA 11 motors was developed. By attaching the hinge and backplate to the rotary disk, the NEMA 11 motor permits rotation to occur about the axis normal to each chassis wall surface. However, during experimental testing, it was discovered that the NEMA 11 does not have sufficient torque to fully rotate the solar array past 15° either CW or CCW.

Ultimately, a low-cost SADM was developed for $625. Considering that CubeSats developed with COTS components and manufacturing could cost around $10,000, the project objective was fulfilled. A major issue during breadboard prototyping and testing was that wire disconnections were frequent. Furthermore, there were many instances where the pins of the wires themselves failed. Thus, although the SADM breadboard prototype was successful, the design should be shifted to a PCB-based design with an integrated burn-wire circuit. Furthermore, rather than using an external 12V power supply, the EPS could be redesigned to support energy storage via onboard batteries and incorporate boost converters, buck converters, and switches where appropriate.

References

[1] "Timeline: Flight and Space Travel," NASA, Oct 31 2008. Retrieved 25 September 2024. https://www.chipublib.org/timeline-flight-and-space-travel/

[2] Uri, J., "55 Years Ago: Apollo 11's One Small Step, One Giant Leap," NASA, Jul 16 2024. Retrieved 25 September 2024. https://www.nasa.gov/history/55-years-ago-apollo-11s-one-small-step-one-giant-leap/

[3] SMD Content Editors, "Voyager Mission Overview," NASA, Aug2024. Retrieved 25 September 2024. https://science.nasa.gov/mission/voyager/mission-overview/

[4] SMD Content Editors, "Parker Solar Probe," NASA, Oct2023. Retrieved 25 September 2024. https://science.nasa.gov/mission/parker-solar-probe/

[5] SMD Content Editors, "Voyager Mission Fact Sheet," NASA, Aug2024. Retrieved 25 September 2024. https://science.nasa.gov/mission/voyager/fact-sheet/

[6] Nicola, F. J., and Driesman, A. S., "Parker Solar Probe: A 60-Year Journey to Reach a Star," *The Bridge*, Vol. 51, No. 3, 2021, pp. 26–33.

[7] "GOES-R Series Frequently Asked Questions (FAQs)." Retrieved 25 September 2024. https://www.goes-r.gov/resources/faqs.html

[8] Costa, J., "NASA Awards Launch Services Contract for GOES-U Mission," NASA, Jul 26 2023. Retrieved 25 September 2024. https://www.nasa.gov/news-release/nasa-awards-launch-services-contract-for-goes-u-mission

[9] Chin, J., Coelho, R., Foley, J., Johnstone, A., Nugent, R., Pignatelli, D., Powell, N., Puig-Suari, J., Atkinson, W., Dorsey, J., Higginbotham, S., Krienke, M., Nelson, K., Poffenberger, B., Raffington, G., Skrobot, G., Treptow, J., Sweet, A., Crusan, J., Galica, C., Horne, W., Norton, C., Robinson, A., Aldred, M., Cooke, A., Hla, T., Ostovar, M., and Way, J., "CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers," October 2017.

[10] Poghosyan, A., and Golkar, A., "CubeSat Evolution: Analyzing CubeSat Capabilities for Conducting Science Missions," *Progress in Aerospace Sciences*, Vol. 88, 2017, pp. 59–83. https://doi.org/10.1016/j.paerosci.2016.11.002

[11] Lee, S., Hutputanasin, A., Toorian, A., Lan, W., Munakata, R., Carnahan, J., Pignatelli, D., Mehrparvar, A., and Johnstone, A., "CubeSat Design Specification," February 2022.

[12] Kulu, E., "Nanosats Database," May 31 2024. Retrieved 19 September 2024. https://www.nanosats.eu/#info

[13] Kulu, E., "Nanosatellite & CubeSat Database," May 31 2024. Retrieved 20 September 2024. https://www.nanosats.eu/database

[14] SMD Content Editors, "BurstCube," NASA, Sep 17 2024. Retrieved 20 September 2024. https://science.nasa.gov/mission/burstcube/

[15] SMD Content Editors, “CeREs,” NASA, 2023. Retrieved 20 September 2024. https://science.nasa.gov/mission/compact-radiation-belt-explorer/

[16] SMD Content Editors, “CuPID,” NASA, 2023. Retrieved 20 September 2024. https://science.nasa.gov/mission/cupid/

[17] SMD Content Editors, “CuSP,” NASA, 2023. Retrieved 20 September 2024. https://science.nasa.gov/mission/cusp/

[18] SMD Content Editors, “E-TBEx: Enhanced Tandem Beacon Experiment,” *NASA*, 2023. Retrieved 20 September 2024. https://science.nasa.gov/mission/e-tbex/

[19] SMD Content Editors, “ELFIN,” NASA, 2023. Retrieved 20 September 2024. https://science.nasa.gov/mission/elfin/

[20] SMD Content Editors, “Mars Cubesat One (MarC0),” NASA, Jun2024. Retrieved 20 September 2024. https://science.nasa.gov/mission/marco/

[21] SMD Content Editors, “PREFIRE,” NASA, Jun2024. Retrieved 20 September 2024. https://science.nasa.gov/mission/prefire/

[22] SMD Content Editors, “SunRISE,” NASA, 2023. Retrieved 20 September 2024. https://science.nasa.gov/mission/sunrise/

[23] Erik, K., “TechEdSat-15 Satellite,” May 29 2024. Retrieved 20 September 2024. https://www.nanosats.eu/sat/techedsat-15

[24] “General Environmental Verification Standard (GEVS) for GSFC Flight Programs and Projects,” NASA, Greenbelt, Apr 28 2021. Retrieved 21 September 2024. https://standards.nasa.gov/sites/default/files/standards/GSFC/B/0/gsfc-std-7000b_signature_cycle_04_28_2021_fixed_links.pdf

[25] Jacklin, S. A., “Small-Satellite Mission Failure Rates,” March 2019.

[26] Alanazi, A., and Straub, J., “Statistical Analysis of CubeSat Mission Failure,” 2018.

[27] Berg, G., Braun, B., Hinkley, D., and Venturini, C., “Improving Mission Success of CubeSats,” 2018.

[28] Langer, M., and Bouwmeester, J., “Reliability of CubeSats-Statistical Data, Developers’ Beliefs and the Way Forward,” 2016.

[29] Lewis, M., and Brown, C., “Failure of the Ball-Lock Mechanism on the NanoRacks CubeSat Deployer,” 2018.

[30] Vertat, I., and Vobornik, A., “Efficient and Reliable Solar Panels for Small Picosatellites,” *International Journal of Photoenergy*, Vol. 2014, No. 1, 2014. https://doi.org/10.1155/2014/537645

[31] Santoni, F., Piergentili, F., Donati, S., Perelli, M., Negri, A., and Marino, M., “An Innovative Deployable Solar Panel System for Cubesats,” *Acta Astronautica*, Vol. 95, No. 1, 2013, pp. 210–217. https://doi.org/10.1016/j.actaastro.2013.11.011

[32] Passaretti, M., and Hayes, R., “Development of a Solar Array Drive Assembly for CubeSat,” 2010.

[33] Thurn, A., Huynh, S., Koss, S., Oppenheimer, P., Butcher, S., Schlater, J., and Hagan, P., “A Nichrome Burn Wire Release Mechanism for CubeSats,” 2012.

[34] Guzik, A. T., and Benafan, O., “Design and Development of CubeSat Solar Array Deployment Mechanisms Using Shape Memory Alloys,” Cleveland, June 2018.

[35] “1U Cubesat Skeleton Chassis,” satsearch.co, Sep 08 2021. Retrieved 23 September 2024. https://satsearch.co/products/aphelion-1u-cubesat-skeleton-chassis

[36] “VERSE-02 2U Structure,” satsearch.co, Sep 09 2024. Retrieved 23 September 2024. https://satsearch.co/products/2ndspace-verse-02-2u-structure

[37] “VERSE-03 3U Cubesat Structure,” satsearch.co, Sep 09 2024. Retrieved 23 September 2024. https://satsearch.co/products/2ndspace-verse-03-3u-structure

[38] “VERSE-06 6U Structure,” satsearch.co, Sep 09 2024. Retrieved 23 September 2024. https://satsearch.co/products/2ndspace-verse-06-6u-structure

[39] Bhamra, C., “Comparative Study: Stepper Motors vs. Brushless DC Motors,” ezmotion.co. Retrieved 23 September 2024. https://www.ezmotion.co/comparative-study-stepper-motors-vs-brushless-dc-motors

[40] “Space-Qualified Stepper Motors,” ducommun.com. Retrieved 24 September 2024. https://www.ducommun.com/engineeredsolutions/MotorsResolvers/test/Steppers.aspx

[41] “PhySPACE,” phytron.eu. Retrieved 24 September 2024. https://www.phytron.eu/products/motors-actuators/physpace/

[42] “Hybrid Stepper Motors,” american-motion.com. Retrieved 24 September 2024. https://american-motion.com/space/products/hybrid-stepper-motors/

[43] “Solar Panels,” cubesatshop.com. Retrieved 24 September 2024. https://www.cubesatshop.com/product-category/nanosatellite-solar-panels/

[44] “Standard Solar Panels,” pumpkinspace.com. Retrieved 24 September 2024. https://www.pumpkinspace.com/store/c25/Standard_Solar_Panels.html

[45] “Solar Panels,” endurosat.com. Retrieved 24 September 2024. https://www.endurosat.com/products-category/solar-panels/

[46] “10Pcs 5V 30mA Mini Solar Panels for Mini Solar Cells DIY Electric Toy Materials Photovoltaic Cells Solar DIY System Kits 2.08"x1.18"(5V 30mA 53mmx30mm),” 2018. Retrieved 16 October 2024. https://www.amazon.com/dp/B07BMMHMSJ?ref=ppx_yo2ov_dt_b_fed_asin_title

[47] “Iverntech Nema 11 Stepper Motor 28mm Body 1.8 Stepper Angle 0.8A 2 Phase 4-Lead with 50CM Cable for 3D Printer, CNC Machine and Robotics,” Mar 16 2019. Retrieved 3 February 2025. https://www.amazon.com/Iverntech-Stepper-Printer-Machine-Robotics/dp/B07PNV7RBW/ref=sr_1_1_pp?dib=eyJ2IjoiMSJ9.3hRNEjd32MJppZsE32yhHtykSdFSfCsI0mKDNRkTu7xbfFcm94-8HjrECexIW7r5vBZPKcvBIEfiSNgAVtkv-

Qt0Gc9C2Lu-mlcqh08f0_oNMBdlMuaHoB95xh4itoDAd1aUlnwHUqfW2t6p6I2_Wmoln2Z_cjv59HrOUz6RlzI7IAXVvmjuz3hBO3itqIqkw4vuvhMJsIdUVaQNEauDReVFmYbfgjuJ0t9oXKAUK_8.pd1uYG3WJNJFLzu3dJKM6iHnUILNmDecgvkTALofUFQ&dib_tag=se&keywords=nema%2B11%2Bmotor&qid=1746596782&sr=8-1&th=1

[48] “NEMA 17 Stepper Motor Datasheet,” Oct 30 2020. Retrieved 17 November 2024. https://www.omc-stepperonline.com/download/17HS19-2004S1.pdf

[49] “DRV8833 Datasheet,” Jul2015. Retrieved 17 November 2024. https://www.ti.com/lit/ds/symlink/drv8833.pdf?ts=1733067055800&ref_url=https%253A%252F%252Fwww.google.com%252F

[50] “DRV8825 Datasheet,” Jul2014. Retrieved 17 November 2024. https://www.ti.com/lit/ds/symlink/drv8825.pdf

[51] “A4988 Datasheet,” May 07 2014. Retrieved 17 November 2024. https://www.pololu.com/file/0j450/a4988_dmos_microstepping_driver_with_translator.pdf

[52] “TEMT6000X01.” Retrieved 16 November 2024. www.vishay.com/doc?91000

[53] “INA219 Zero-Drift, Bidirectional Current/Power Monitor with I2C Interface Datasheet,” Dec2015. Retrieved 8 April 2025. https://www.ti.com/lit/ds/symlink/ina219.pdf?ts=1746722971494

[54] “INA226 36V, 16-Bit, Ultra-Precise I2C Output Current, Voltage, and Power Monitor With Alert Datasheet,” Sep2024. Retrieved 8 April 2025. https://www.ti.com/lit/ds/symlink/ina226.pdf?ts=1746719111981&ref_url=https%253A%252F%252Fwww.google.com%252F

[55] “PCA9548A Low Voltage 8-Channel I2C Switch with Reset Datasheet,” 2021.

[56] “TCA9548A Low-Voltage 8-Channel I2 C Switch with Reset Datasheet,” Sep2024. Retrieved 8 April 2025. https://www.ti.com/lit/ds/symlink/tca9548a.pdf

[57] Wei, C.-C., “Evaluation of Photovoltaic Power Generation by Using Deep Learning in Solar Panels Installed in Buildings,” *Energies*, Vol. 12, No. 18, 2019. https://doi.org/10.3390/en12183564

[58] Ameur, A., Berrada, A., Loudiyi, K., and Adomatis, R., “Performance and Energetic of Hybrid PV Systems with Battery Energy,” *Hybrid Energy System Models*, Elsevier, 2020, pp. 195–238. https://doi.org/10.1016/B978-0-12-821403-9.00008-1

[59] Coddington, O., Lean, J. L., Pilewskie, P., Snow, M., and Lindholm, D., “A Solar Irradiance Climate Data Record,” *Bulletin of the American Meteorological Society*, Vol. 97, No. 7, 2016, pp. 1265–1282. https://doi.org/10.1175/BAMS-D-14-00265.1

[60] Takyi, G., and Nyarko, F. K., “Investigation of the Effect of Temperature Coefficients on Mono-Crystalline Silicon PV Module Installed in Kumasi, Ghana,” *Journal of Power and Energy Engineering*, Vol. 08, No. 09, 2020, pp. 20–34. https://doi.org/10.4236/jpee.2020.89003

[61] "MX Polyamide MXD6 Nylon," Chiyoda. Retrieved 20 March 2025. https://www.fostercomp.com/wp-content/uploads/2018/11/MX-Nylon_properties.pdf

[62] "(PTFE) Polytetrafluoroethylene." Retrieved 20 March 2025. https://www.bearingworks.com/uploaded-assets/pdfs/retainers/ptfe-datasheet.pdf

[63] Maradini, G. da S., Oliveira, M. P., Carreira, L. G., Guimarães, D., Profeti, D., Dias Júnior, A. F., Boschetti, W. T. N., Oliveira, B. F. de, Pereira, A. C., and Monteiro, S. N., "Impact and Tensile Properties of Polyester Nanocomposites with Conifer Fiber Cellulose Nanocrystal: A Previous Study Extension," *Polymers*, Vol. 13, No. 11, 2021. https://doi.org/10.3390/polym13111878

[64] "Polyamide." Retrieved 20 March 2025. https://www.polyplasty.cz/material-polyamid.html?lang=2

[65] Blaine, R. L., "Polymer Heats of Fusion," New Castle. Retrieved 20 March 2025. https://www.tainstruments.com/pdf/literature/TN048.pdf

[66] Tan, P. S., Farid, A. A., Karimzadeh, A., Rahimian Koloor, S. S., and Petrů, M., "Investigation on the Curvature Correction Factor of Extension Spring," *Materials*, Vol. 13, No. 18, 2020. https://doi.org/10.3390/MA13184199

[67] "Stepper Motors," applied-motion.com. Retrieved 23 September 2024. https://www.applied-motion.com/s/category/products/stepper-products/stepper-motors/0ZG5i0000008sUpGAI?measurement=Imperial

[68] "Motion System Selector," shop.faulhaber.com. Retrieved 24 September 2024. https://shop.faulhaber.com/productselector

Appendix A: Stepper Motors

Table A.1 Stepper motor specifications acquired from [67].

Stepper Motor	OTR (°C)	Cost ($)
5014-020 NEMA 14 Stepper Motor	-20 to 50	31
5014-020D NEMA 14 Stepper Motor w/ double shaft	-20 to 50	41
5014-042 NEMA 14 Stepper Motor	-20 to 50	52
5014-042D NEMA 14 Stepper Motor w/ double shaft	-20 to 50	52
5014-897 NEMA 14 Stepper Motor, 0.9 Degree	-20 to 50	46
5014-897D NEMA 14 Stepper Motor, 0.9 Degree w/ double shaft	-20 to 50	26
HH17-101 Hollow Shaft, NEMA 17 High Torque Stepper Motor	-20 to 50	72
HH17-103 Hollow Shaft, NEMA 17 High Torque Stepper Motor	-20 to 50	79
HH17-105 Hollow Shaft, NEMA 17 High Torque Stepper Motor	-20 to 50	116
HH23-101 Hollow Shaft, NEMA 23 High Torque Stepper Motor	-20 to 50	107
HH23-103 Hollow Shaft, NEMA 23 High Torque Stepper Motor	-20 to 50	136
HT08-230 NEMA 8 High Torque Stepper Motor	-20 to 50	53
HT08-230D NEMA 8 High Torque Stepper Motor w/ double shaft	-20 to 50	53
HT08-231 NEMA 8 High Torque Stepper Motor	-20 to 50	67
HT08-231D NEMA 8 High Torque Stepper Motor w/ double shaft	-20 to 50	67
HT11-020 NEMA 11 High Torque Stepper Motor	-20 to 50	58
HT11-020D NEMA 11 High Torque Stepper Motor w/ double shaft	-20 to 50	58
HT11-021 NEMA 11 High Torque Stepper Motor	-20 to 50	73
HT11-021D NEMA 11 High Torque Stepper Motor w/ double shaft	-20 to 50	73
HT17-221 NEMA 17 High Torque Stepper Motor, 0.9 Degree	-20 to 50	51
HT17-221D NEMA 17 High Torque Stepper Motor, 0.9 Degree w/ double shaft	-20 to 50	51
HT17-222 NEMA 17 High Torque Stepper Motor, 0.9 Degree	-20 to 50	56
HT17-222D NEMA 17 High Torque Stepper Motor, 0.9 Degree w/ double shaft	-20 to 50	56
HT17-268 NEMA 17 High Torque Stepper Motor	-20 to 50	48
HT17-268D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	48
HT17-268D-FBA NEMA 17 High Torque Stepper Motor with Double Shaft and FBA encoder	-20 to 50	102
HT17-268D-WAA NEMA 17 High Torque Stepper Motor w/ WAA encoder	-20 to 50	228
HT17-269 NEMA 17 High Torque Stepper Motor	-20 to 50	58
HT17-269D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	58
HT17-270 NEMA 17 High Torque Stepper Motor	-20 to 50	58
HT17-270D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	58
HT17-271 NEMA 17 High Torque Stepper Motor	-20 to 50	53
HT17-271D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	53

HT17-271D-FBA NEMA 17 High Torque Stepper Motor with Double Shaft and FBA Encoder	-20 to 50	105
HT17-271D-WAA NEMA 17 High Torque Stepper Motor w/ WAA encoder	-20 to 50	233
HT17-272 NEMA 17 High Torque Stepper Motor	-20 to 50	65
HT17-272D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	65
HT17-273 NEMA 17 High Torque Stepper Motor	-20 to 50	65
HT17-273D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	65
HT17-274 NEMA 17 High Torque Stepper Motor	-20 to 50	65
HT17-274D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	65
HT17-275 NEMA 17 High Torque Stepper Motor	-20 to 50	70
HT17-275D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	70
HT17-275D-FBA NEMA 17 High Torque Stepper Motor with Double Shaft and FBA encoder	-20 to 50	112
HT17-275D-WAA NEMA 17 High Torque Stepper Motor w/ WAA encoder	-20 to 50	250
HT17-276 NEMA 17 High Torque Stepper Motor	-20 to 50	77
HT17-276D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	77
HT17-277 NEMA 17 High Torque Stepper Motor	-20 to 50	77
HT17-277D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	77
HT17-278 NEMA 17 High Torque Stepper Motor	-20 to 50	84
HT17-278D NEMA 17 High Torque Stepper Motor w/ double shaft	-20 to 50	84
HT17-278D-FBA NEMA 17 High Torque Stepper Motor with Double Shaft and FBA encoder	-20 to 50	133
HT17-278D-WAA NEMA 17 High Torque Stepper Motor w/ WAA encoder	-20 to 50	263
HT23-548 NEMA 23 High Torque Stepper Motor	-20 to 50	141
HT23-548D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	141
HT23-549 NEMA 23 High Torque Stepper Motor	-20 to 50	152
HT23-549D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	152
HT23-550 NEMA 23 High Torque Stepper Motor	-20 to 50	200
HT23-550D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	200
HT23-552 NEMA 23 High Torque Stepper Motor	-20 to 50	125
HT23-552D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	125
HT23-552D-ZAA NEMA 23 High Torque Stepper Motor w/ ZAA Encoder	-20 to 50	304
HT23-552D-ZAC NEMA 23 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	462
HT23-553 NEMA 23 High Torque Stepper Motor	-20 to 50	135
HT23-553D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	135
HT23-553D-ZAA NEMA 23 High Torque Stepper Motor w/ ZAA Encoder	-20 to 50	315
HT23-553D-ZAC NEMA 23 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	472
HT23-554 NEMA 23 High Torque Stepper Motor	-20 to 50	172
HT23-554D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	172
HT23-554D-ZAA NEMA 23 High Torque Stepper Motor w/ ZAA Encoder	-20 to 50	351

HT23-554D-ZAC NEMA 23 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	509
HT23-559 NEMA 23 High Torque Stepper Motor, 0.9 Degree	-20 to 50	70
HT23-559D NEMA 23 High Torque Stepper Motor, 0.9 Degree w/ double shaft	-20 to 50	70
HT23-560 NEMA 23 High Torque Stepper Motor, 0.9 Degree	-20 to 50	59
HT23-560D NEMA 23 High Torque Stepper Motor, 0.9 Degree w/ double shaft	-20 to 50	59
HT23-593 NEMA 23 High Torque Stepper Motor	-20 to 50	84
HT23-593D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	84
HT23-594 NEMA 23 High Torque Stepper Motor	-20 to 50	84
HT23-594B NEMA 23 High Torque Stepper Motor w/ Brake	-20 to 50	277
HT23-594D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	84
HT23-594D-GBA NEMA 23 High Torque Stepper Motor with Double Shaft and GBA Encoder	-20 to 50	118
HT23-594D-ZAA NEMA 23 High Torque Stepper Motor w/ ZAA Encoder	-20 to 50	263
HT23-595 NEMA 23 High Torque Stepper Motor	-20 to 50	64
HT23-595D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	64
HT23-596 NEMA 23 High Torque Stepper Motor	-20 to 50	91
HT23-596D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	91
HT23-597 NEMA 23 High Torque Stepper Motor	-20 to 50	91
HT23-597D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	91
HT23-598 NEMA 23 High Torque Stepper Motor	-20 to 50	81
HT23-598B NEMA 23 High Torque Stepper Motor w/ Brake	-20 to 50	295
HT23-598C NEMA 23 Stepper Motor with 10 Foot Shielded Cable	-20 to 50	133
HT23-598D NEMA 23 High Torque Stepper Motor w/ double shaft	-20 to 50	81
HT23-598DC NEMA 23 Stepper Motor with 10 Foot Shielded Cable w/ Double Shaft	-20 to 50	133
HT23-598DC-ZAA NEMA 23 Stepper Motor with 10 Foot Shielded Cable w/ ZAA Encoder	-20 to 50	314
HT23-598DC-ZAC NEMA 23 Stepper Motor with 10 Foot Shielded Cable w/ Encoder and Cover	-20 to 50	471
HT23-598D-GBA NEMA 23 High Torque Stepper Motor with Double Shaft and GBA Encoder	-20 to 50	124
HT23-598D-ZAA NEMA 23 High Torque Stepper Motor w/ ZAA Encoder	-20 to 50	262
HT23-599 NEMA 23 High Torque Stepper Motor	-20 to 50	151
HT23-599D NEMA 23 High Torque Stepper Motor w/ Double Shaft	-20 to 50	151
HT23-600 NEMA 23 High Torque Stepper Motor	-20 to 50	151
HT23-600D NEMA 23 High Torque Stepper Motor w/ Double Shaft	-20 to 50	151
HT23-601 NEMA 23 High Torque Stepper Motor	-20 to 50	107
HT23-601B NEMA 23 High Torque Stepper Motor w/ Brake	-20 to 50	320
HT23-601C NEMA 23 Stepper Motor with 10 Foot Shielded Cable	-20 to 50	184
HT23-601D NEMA 23 High Torque Stepper Motor w/ Double Shaft	-20 to 50	107
HT23-601DC NEMA 23 Stepper Motor with 10 Foot Shielded Cable w/ Double Shaft	-20 to 50	184

HT23-601DC-ZAA NEMA 23 Stepper Motor with 10 Foot Shielded Cable w/ ZAA Encoder	-20 to 50	363
HT23-601DC-ZAC NEMA 23 Stepper Motor with 10 Foot Shielded Cable w/ Encoder and Cover	-20 to 50	521
HT23-601D-GBA NEMA 23 High Torque Stepper Motor with Double Shaft and GBA encoder	-20 to 50	161
HT23-601D-ZAA NEMA 23 High Torque Stepper Motor w/ ZAA Encoder	-20 to 50	286
HT23-603 NEMA 23 High Torque Stepper Motor, 354 oz-in	-20 to 50	171
HT23-603B NEMA 23 High Torque Stepper Motor, 354 oz-in w/ Brake	-20 to 50	384
HT23-603D NEMA 23 High Torque Stepper Motor, 354 oz-in w/ Double Shaft	-20 to 50	171
HT23-603D-GBA NEMA 23 High Torque Stepper Motor with Double Shaft and GBA encoder	-20 to 50	224
HT23-603D-ZAA NEMA 23 High Torque Stepper Motor, 354 oz-in w/ ZAA Encoder	-20 to 50	350
HT24-100 NEMA 24 High Torque Stepper Motor	-20 to 50	102
HT24-100D NEMA 24 High Torque Stepper Motor w/ Double Shaft	-20 to 50	102
HT24-105 NEMA 24 High Torque Stepper Motor	-20 to 50	114
HT24-105D NEMA 24 High Torque Stepper Motor w/ Double Shaft	-20 to 50	114
HT24-108 NEMA 24 High Torque Stepper Motor	-20 to 50	205
HT24-108D NEMA 24 High Torque Stepper Motor w/ Double Shaft	-20 to 50	205
HT34-504 NEMA 34 High Torque Stepper Motor	-20 to 50	155
HT34-504D NEMA 34 High Torque Stepper Motor w/ double shaft	-20 to 50	155
HT34-504D-HBA NEMA 34 High Torque Stepper Motor with Double Shaft and HBA Encoder	-20 to 50	246
HT34-504D-YAA NEMA 34 High Torque Stepper Motor w/ YAA Encoder	-20 to 50	334
HT34-505 NEMA 34 High Torque Stepper Motor	-20 to 50	212
HT34-505D NEMA 34 High Torque Stepper Motor w/ double shaft	-20 to 50	212
HT34-505D-HBA NEMA 34 High Torque Stepper Motor with Double Shaft and HBA Encoder	-20 to 50	294
HT34-505D-YAA NEMA 34 High Torque Stepper Motor w/ YAA Encoder	-20 to 50	392
HT34-506 NEMA 34 High Torque Stepper Motor	-20 to 50	294
HT34-506C NEMA 34 Stepper Motor with 10 Foot Shielded Cable	-20 to 50	414
HT34-506D NEMA 34 High Torque Stepper Motor w/ double shaft	-20 to 50	294
HT34-506DC NEMA 34 Stepper Motor with 10 Foot Shielded Cable w/ double shaft	-20 to 50	414
HT34-506DC-YAA NEMA 34 Stepper Motor with 10 Foot Shielded Cable w/ YAA Encoder	-20 to 50	593
HT34-506DC-YAC NEMA 34 Stepper Motor with 10 Foot Shielded Cable w/ Encoder and Cover	-20 to 50	751
HT34-506D-HBA NEMA 34 High Torque Stepper Motor with Double Shaft and HBA Encoder	-20 to 50	350
HT34-506D-YAA NEMA 34 High Torque Stepper Motor w/ YAA Encoder	-20 to 50	473
HT34-687 NEMA 34 High Torque Stepper Motor	-20 to 50	403
HT34-687D NEMA 34 High Torque Stepper Motor w/ double shaft	-20 to 50	403

HT34-687D-HBA NEMA 34 High Torque Stepper Motor with Double Shaft and HBA encoder	-20 to 50	485
HT34-687D-YAA NEMA 34 High Torque Stepper Motor w/ YAA Encoder	-20 to 50	583
HT34-687D-YAC NEMA 34 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	740
HT34-695 NEMA 34 High Torque Stepper Motor	-20 to 50	398
HT34-695D NEMA 34 High Torque Stepper Motor w/ double shaft	-20 to 50	398
HT34-695D-YAA NEMA 34 High Torque Stepper Motor w/ YAA Encoder	-20 to 50	578
HT34-695D-YAC NEMA 34 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	735
HT34-696 NEMA 34 High Torque Stepper Motor	-20 to 50	407
HT34-696D NEMA 34 High Torque Stepper Motor w/ double shaft	-20 to 50	407
HT34-696D-YAA NEMA 34 High Torque Stepper Motor w/ YAA Encoder	-20 to 50	586
HT34-696D-YAC NEMA 34 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	745
HT34-697 NEMA 34 High Torque Stepper Motor	-20 to 50	415
HT34-697D NEMA 34 High Torque Stepper Motor w/ double shaft	-20 to 50	415
HT34-697D-HBA NEMA 34 High Torque Stepper Motor with Double Shaft and HBA encoder	-20 to 50	497
HT34-697D-YAA NEMA 34 High Torque Stepper Motor w/ YAA Encoder	-20 to 50	594
HT34-697D-YAC NEMA 34 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	754
HW23-598 IP65 Rated, NEMA 23 High Torque Stepper Motor	-20 to 50	256
HW23-598D-ZAC IP65 Rated, NEMA 23 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	602
HW23-601 IP65 Rated, NEMA 23 High Torque Stepper Motor	-20 to 50	331
HW23-601D-ZAC IP65 Rated, NEMA 23 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	679
HW23-753 IP65 Rated, NEMA 23 High Torque Stepper Motor	-20 to 50	256
HW23-753D-ZAC IP65 Rated, NEMA 23 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	602
HW23-754 IP65 Rated, NEMA 23 High Torque Stepper Motor	-20 to 50	331
HW23-754D-ZAC IP65 Rated, NEMA 23 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	679
HW24-108 IP65 Rated, NEMA 24 High Torque Stepper Motor	-20 to 50	350
HW24-108D-ZAC IP65 Rated, NEMA 24 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	696
HW34-506 IP65 Rated, NEMA 34 High Torque Stepper Motor	-20 to 50	590
HW34-506D-YAC IP65 Rated, NEMA 34 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	934
HW34-696 IP65 Rated, NEMA 34 High Torque Stepper Motor	-20 to 50	575
HW34-696D-YAC IP65 Rated, NEMA 34 High Torque Stepper Motor w/ Encoder and Cover	-20 to 50	920
HX56-100 - Stepper Motor for Hazardous Locations	UNK	UNK

Table A.2 Stepper motors acquired from [68].

Stepper Motor	OTR (°C)	Cost ($)
AM1524SB045004	-35 to 70	156.96
DM1220SB033030	-35 to 70	157.88
DM12202R033030	-35 to 70	172.01
AM15242R045004	-35 to 70	174.48
AM0820SB022500	-30 to 70	181.78
AM10202R025000	-35 to 70	187.51
AM1524AE045004	-35 to 70	192.37
AM1524SB045006+16A 11.9:1+MG01	-35 to 70	192.79
AM08202R022500	-30 to 70	197.52
DM0620SB008011	-35 to 70	198.85
DM0620SB013011	-30 to 70	198.85
AM1524SB045006+16A 22:1+MG01	-35 to 70	199.55
AM1524SB045006+16A 41:1+MG01	-35 to 70	199.55
AM1524SB045006+16A 141:1+MG01	-35 to 70	201.41
AM1524SB045070+15A 5.33:1+MG03	-35 to 70	202.52
DM1220SB033038+12/4 4:1+MG09	-35 to 70	205.33
DM06202R008011	-35 to 70	206.36
AM1524SB045006+16A 485:1+MG01	-35 to 70	206.36
AM1524SB045006+16A 262:1+MG01	-35 to 70	206.75
AM1524SB045070+15A 19:1+MG03	-35 to 70	208.79
AM1524SB045070+15A 28:1+MG03	-35 to 70	209.79
AM1524SB045070+15A 14:1+MG03	-35 to 70	208.79
AM15242R045006+16A 11.9:1+MG01	-35 to 70	210.31
AM1524SB045006+16A 900:1+MG01	-35 to 70	210.97
AM1524SB045006+16A 1670:1+MG01	-35 to 70	210.97
AM1524SB045070+15A 52:1+MG03	-35 to 70	211.75
AM1524SB045070+15A 69:1+MG03	-35 to 70	211.75
AM1524SB045070+15A 52:1+MG03	-35 to 70	211.75
AM1524SB045070+15A 152:1+MG03	-35 to 70	211.75
AM1524SB045006+16A 5752:1+MG01	-35 to 70	215.63
AM1524SB045006+16A 3101:1+MG01	-35 to 70	215.63
AM1524SB045070+15A 369:1+MG03	-35 to 70	216.4
AM1524SB045070+15A 809:1+MG03	-35 to 70	216.4
AM1524SB045070+15A 249:1+MG03	-35 to 70	216.4
AM1524SB045070+15A 546:1+MG03	-35 to 70	216.4
DM1220SB033038+12/4 16:1+MG09	-35 to 70	216.49
AM1524SB045004+HEAM152412	-35 to 70	216.82

Appendix B: Solar Panels

The following tables show solar panel components candidates with the following text color indication:

- Red text: Components that had dimensions less than 65 mm or greater than 100 mm.
- Yellow text: Components that exceeded 65 mm.
- Green text: Components that satisfied all primary driving factors and had at least one dimension exceeded 80 mm.

Table B.1. Solar panel component specifications acquired from amazon.com.

Component Name	Cost ($)	L (mm)	W (mm)	H (mm)
Micro Solar Panels with Wire, 5PCS 5V 200mA Photovoltaic Solar Cells Kit 110mmx60mm / 4.33"x 2.36'	15.99	109.982	59.944	2.54
SUNYIMA 10pcs Mini Monocrystalline Solar Cells Solar System Kit 50mm X 50mm/1.96" X 1.96" 2V 160MA for DIY Charge Solar Panels	15.5	50.038	50.038	50.038
Retisee 30 Pcs Small Solar Panels Mini Polycrystalline Solar Cells 5v 60ma Solar Epoxy Plate Photovoltaic Cells Charger Module with 10 cm Cable for Solar Battery Charger DIY Solar System 68 x 37 mm	25.99	68.072	37.084	UNK
100pcs 0.5V 400mA Micro Mini Solar Cell for Solar Panels 52mmx 19mm/2‘’x0.75” Polycrystalline Silicon Photovoltaic Solar Cells Sun Power for DIY Cell Phone Charger	15.99	50.8	19.05	UNK
Sunnytech 0.5w 5v 100ma Mini Small Solar Panel Module DIY Polysilicon Solar Epoxy Cell Charger B016 (GP80*80-10A100)	6.99	80.00	80.00	UNK
SUNYIMA 100pcs Mini Solar Cells 0.5V 0.4W Micro Thin Polycrystalline Silicon Solar Panels DIY 52 x 52mm/2x2inch	21.99	52.07	52.07	52.07
FellDen Micro Solar Panels Photovoltaic Cells, 10PCS 5V 200mA Epoxy Panel Kit Polycrystalline Cells 110mmx60mm / 4.33"x 2.36" (5V200mA)	20.99	145.034	82.042	39.878
uxcell 2W 6V Small Solar Panel Module DIY Polysilicon with 250mm Wire	5.35	109.982	3.048	UNK
Newpowa 5W(Watts) 12V(Volts) 5W Monocrystalline Solar Panel Battery Maintainer High-Efficiency PV Module Power for Battery Charging of Boat RV Gate Opener Fences	20.49	225.044	195.072	18.034
SUNYIMA Mini Solar Panel, 5V 7W USB IBC Solar Panel Charger with Built-in Voltage	16.99	329.946	1859.28	12.954

Stabilization System for Smart Phone, Camping Lanterns, Small Fans Monitor				
SUNYIMA 10Pcs 5V 60mA Epoxy Solar Panel Polycrystalline Solar Cells for Solar Battery Charger DIY Solar Syatem Kits 68mmx37mm / 2.67"x1.45" 5V Solar Cells	16.99	67.818	36.83	UNK
uxcell 5Pcs 0.6W 5.5V Small Solar Panel Module DIY with 100mm Wire	18.99	66.04	66.04	UNK
MECCANIXITY Mini Solar Panel Cell 3V 90mA 0.27W 62mm x 36mm for DIY Electric Power Project Pack of 5	12.49	61.976	36.068	UNK
Mini Solar Panel, Pack of 3 DC 6V 1W Solar Panel Cell Power Module Polycrystalline Silicon Solar Panel with 30cm Cable for Making Solar Lawn Lights Solar Cell Phone Charger	13.71	109.982	59.944	29.972
Cell Power Module, 3pcs Mini Solar Panel Set DC 6V 1W Solar Cell Power Module, Polysilicon Solar Cell Charger Module Solar DIY System Kits with 11.8in Cable for Outdoor Camping Hiking Travel	12.69	109.22	60.96	2.54
Jameco Valuepro G/S Solar Panel with Alligator Clips, 6V, 150 mA, 0.9W	12.21	88.9	76.2	UNK
DMiotech 5 Pack 3V 110mA 60mm x 55mm Mini Solar Panel Cell for DIY Electric Power Project	8.49	59.944	55.118	UNK
DMiotech 5 Pack 0.5V 100mA 53mm x 18mm Mini Solar Panel Cell for DIY Electrical Power Project	7.99	53.086	18.034	UNK
DMiotech 5 Pack 2V 160mA 60mm x 60mm Mini Solar Panel Cell for DIY Electric Power Project	7.99	59.944	59.944	UNK
DMiotech 12V 1.5W 115mm x 90mm Mini Solar Panel Cell for DIY Electric Power Project	9.49	115.062	115.062	89.916
DMiotech 5 Pack 5.5V 60mA 60mm x 60mm Mini Solar Panel Cell for DIY Electric Power Project	7.49	59.944	59.944	UNK
DMiotech 5 Pack 0.5V 100mA 53mm x 18mm Mini Solar Panel Cell for DIY Electric Power Project	5.49	53.086	18.034	UNK
DMiotech 18V 2.5W 194mm x 120mm Mini Solar Panel Cell for DIY Electric Power Project	13.49	194.056	119.888	UNK
4Pcs 0.15W 3V Solar Panel Polysilicon Solar Charging Module Accessory 40x30mm 4Pcs 0.15W 3V Solar Panel for Family	9.83	130.048	109.982	9.906

DMiotech 6V 3W 145mm x 145mm Mini Solar Panel Cell for DIY Electric Power Project	13.99	145.034	145.034	UNK
2W 12V Solar Panel DIY Battery Solar Charger Board + Tiger Clip- Perfect for Outdoor Charging Batteries and Low-Power Electronic Products	9.99	110	13	UNK
20 Watt Solar Panel High Efficiency Monocrystalline Photovoltaic Modules	22.99	350.52	17.018	495.3
RANIT 65mm x 65mm 5.5V 110mA Poly Mini Solar Cell Panel Module with 30cm Cable DIY for Charger	11.65	65	65	UNK
3.5W 6V 600ma Mini Solar Panel Module Solar System Cell Outdoor Camping Battery Charger DIY Parts	7.59	165.1	134.874	2.032
2W 12V 160ma Mini Solar Panel Module Solar System Cell Outdoor Camping Battery Charger DIY Parts	6.99	135.89	109.982	2.032
3W 12V 250ma Mini Solar Panel Module Solar System Cell Outdoor Camping Battery Charger DIY Parts	7.99	145.034	145.034	2.032
Ejoyous Mini Solar Panel, 3Pcs DC 6V Solar Cells Charger Power Module Polycrystalline Silicon Solar Panel with 30cm Cable for Outdoor Camping Hiking Travel	13.56	109.22	60.96	UNK
Flexible Solar Panel Mini Thin Film Solar Cell Battery Charger Solar Energy Charging Home Science Projects 2V 0.5W	8.99	195.072	57.912	1.016
Flexible Solar Panel Mini Thin Film Solar Cell Battery Charger Solar Energy Charging Home Science DIY Projects 1.5V 0.7W | 270mm x 55mm	11.99	270.002	55.118	1.016
Velleman SOL3-VP Solar Cell, 1V, 200 mA, 1.8" x 3.0" x 0.2" Size, Red/Black	6.95	45.72	71.12	0.508

Table B.2. Solar panel specifications acquired from voltaicsystems.com

Component Name	Cost ($)	L (mm)	W (mm)	H (mm)
0.4 Watt 2.5 Volt Mini Solar Panel - ETFE	5.5	32.5	94	2.6
0.3 Watt 2 Volt Mini Solar Panel - ETFE	5.5	52	52	2.6
0.3 Watt 6 Volt Mini Solar Panel - ETFE	6	52	52	2.6
0.6 Watt 6 Volt Small Solar Panel - ETFE	9	66	66	2.6
1.2 Watt 6 Volt Small Solar Panel - ETFE	14	66	113	2.6
2 Watt 6 Volt Small Solar Panel - ETFE	21	112	136	2.7
5 Watt 6 Volt Solar Panel - ETFE	35	148	223	4
9 Watt 18 Volt Solar Panel - ETFE	45	223	274	4

9 Watt 18 Volt Solar Panel	79	221	257	5
10 Watt 6 Volt Solar Panel - ETFE	65	223	274	4
20 Watt 6 Volt Solar Panel - ETFE	99	436	274	4
25 Watt 18 Volt Solar Panel	59	507	307	17
50 Watt 18 Volt Solar Panel	89	586	507	30
100 Watt 18 Volt Solar Panel	139	980	586	30
1.2 Watt 6 Volt Small Solar Panel - Glass	14	66	113	2.6
1 Watt 6 Volt Solar Panel	21	89	113	5
2 Watt 6 Volt Solar Panel	29	136	112	5
3.5 Watt 6 Volt Solar Panel	39	210	113	5
6 Watt 6 Volt Solar Panel	59	175	221	5
9 Watt 6 Volt Solar Panel	79	221	257	5
17 Watt 18 Volt Solar Panel	129	274	39.3	5

Table B.3. Solar panel specifications acquired from digikey.com.

Component Name	Cost ($)	L (mm)	W (mm)	H (mm)
SM141K04LV	5	4.50088	15.0114	2.1082
BCS1714B6	1.86	16.9926	13.9954	0.1778
SM141K05TF	6.24	21.9964	35.0012	1.4986
SM101K07TF	6.85	21.9964	35.0012	1.4986
KXOB25-12X1F-TR	2.92	22.098	7.112	1.778
KXOB25-12X1F-TB	2.92	22.098	7.112	1.778
KXOB25-04X3F-TR	3.01	22.098	7.112	1.778
KXOB25-01X8F-TB	3.29	22.098	7.112	1.778
KXOB25-14X1F-TR	2.79	23.0124	8.001	1.8034
KXOB25-05X3F-TR	3.01	23.0124	8.001	1.8034
KXOB081K06TF-TR	5.97	23.0124	24.9936	1.4986
KXOB201K04F-TR	7.14	23.0124	42.0116	2.1082
KXOB141K06TF-TR	8.19	23.0124	4.20116	1.4986
KXOB25-05X3F-TB	2.9	23.0124	8.001	1.8034
KXOB25-14X1F-TB	2.92	23.0124	8.001	1.8034
KXOB25-03X4F-TR	3.08	23.0124	8.001	2.1082
KXOB25-14X1TF-TR	3.2	23.0124	8.001	1.1938
KXOB25-03X4TF-TR	3.22	23.0124	8.001	1.4986
KXOB25-02X8F-TR	3.29	23.0124	8.001	1.8034
KXOB25-02X8F-TB	3.29	23.0124	8.001	1.8034
KXOB25-02X8TF-TR	3.62	23.0124	8.001	1.1938
KXOB121K04F-TR	5.39	23.0124	24.9936	2.1082
KXOB081K06F-TR	5.5	23.0124	24.9936	2.1082
KXOB061K08F-TR	5.63	23.0124	24.9936	2.1082
KXOB121K04TF-TR	5.67	23.0124	24.9936	1.4986
KXOB061K08TF-TR	5.8	23.0124	24.9936	1.4986
KXOB141K06F-TR	7.3	23.0124	42.0116	2.1082
KXOB101K08F-TR	7.51	23.0124	42.0116	2.1082

KXOB201K04TF-TR	7.85	23.0124	42.0116	1.4986
KXOB101K08TF-TR	8.04	23.0124	42.0116	1.4986
SM700K10L	6.91	24.003	35.9918	1.8034
ONP1.2-12X24	3.91	24.003	11.9888	UNK
SM500K12L	6.11	24.003	32.004	1.8034
SM500K12TF	6.65	24.003	3.2004	1.4986
SM700K10TF	7.44	24.003	35.9918	1.4986
AM-1456CA-DGK-E	2.71	24.9936	10.0076	1.0922
AM-1801CA-DGK-E	4.31	24.9936	53.0098	1.0922
AM-5610CAR-DGK-T	8.12	24.9936	19.9898	2.0066
BCSC452B3	4.05	24.9936	18.9992	0.1778
SM141K04L	5	29.0068	23.0124	2.1082
SM141K04TF	5.5	29.0068	23.0124	38.1
SM141K04LS	5	29.0068	23.0124	2.1082
SM141K04TFS	5.5	29.0068	23.0124	1.4986
AM-1819CA-DGK-E	3.65	30.988	24.003	1.0922
SM710K12L	4.87	32.512	33.02	2.032
SM400K10L	4.65	32.9946	15.0114	1.8034
SM730K12L	5.56	32.9946	32.004	1.8034
SM400K10TF	5.11	32.9946	15.0114	1.4986
SM340K10L	3.27	34.036	16.002	2.032
AM-1417CA-DGK-E	2.88	35.0012	13.8938	1.0922
SM101K07L	4.95	35.0012	21.9964	2.1082
SM141K05L	5.67	35.0012	21.9964	2.1082
SLMD960H12L	9.9	35.0012	42.0116	2.0066
SM850K12L	8.28	38.5064	32.9946	2.0066
SM850K12TF	8.99	38.5064	32.9946	1.4986
AM-5904CAR-DGK-T	9.85	40.1066	33.0962	2.0066
AM-1454CA-DGK-E	4.05	41.6052	26.289	1.0922
SM111K10L	9.39	41.91	35.052	2.032
SM940K12L	8.59	41.91	35.052	2.032
SM141K06TF	4.87	42.0116	23.0124	1.4986
SM141K06L	6.3	42.0116	23.0124	2.1082
SM141K06LS	6.3	42.0116	23.0124	2.1082
SM141K06TFS	6.93	42.0116	23.0124	1.4986
SM111K04L	5	42.926	1.397	2.032
BCSC491B6	5.04	43.9928	29.9974	0.1778
BCS4430B6	5.7	43.9928	4.5974	0.1778
BCSC421B1	5.81	43.9928	29.9974	0.1778
BCS4430B5	5.81	43.9928	29.9974	0.1778
INP3.6-24X44T	7.3	43.9928	24.003	0.2286
SM940K09L	5.01	44.45	21.59	1.778
SM141K10LV	6.6	45.0088	35.9918	1.8034
SM141K04TFV	5.5	45.0088	15.0114	1.1938
SM141K06LV	6.3	45.0088	21.9964	2.1082

SM101K12L	6.82	45.0088	32.004	2.1082
SM141K08LV	7.98	45.0088	29.0068	2.1082
SM101K12TF	8.73	45.0088	32.004	1.4986
SM141K06TFV	6.93	45.0088	21.9964	1.4986
SM141K10TFV	9.96	45.0088	35.9918	1.4986
SM141K04LVS	5	45.0088	15.0114	2.1082
SM141K04TFVS	5.5	45.0088	15.0114	1.4986
SM141K06LVS	6.3	45.0088	21.9964	2.1082
SM141K06TFVS	6.93	45.0088	21.9964	1.4986
SM141K08LVS	7.98	45.0088	29.0068	2.1082
SM141K08TFVS	8.78	45.0088	29.0068	1.4986
SM141K10LVS	9.39	45.0088	35.9918	2.1082
SM141K10TFVS	9.96	45.0088	35.9918	1.4986
SM101K09L	5.61	46.99	23.0124	2.1082
SM141K07L	6.79	50.0126	23.0124	2.1082
SM141K07LS	6.3	50.0126	23.0124	2.1082
SM141K07TFS	7.26	50.0126	23.0124	1.4986
SM141K07TF	7.26	50.0126	23.0124	1.4986
AM-5412CAR-DGK-T	10.98	50.0888	33.0962	2.0066
P121	5.5	51.9938	51.9938	2.8956
P122	6	51.9938	51.9938	2.8956
PRT-18723	6.19	51.9938	51.9938	2.8956
PRT-18724	6.75	51.9938	5.19938	2.8956
5855	7.44	51.9938	51.9938	3.6068
5853	6.88	51.9938	51.9938	3.6068
SP-53X30-4-DK	22.53	52.9844	29.9974	2.0066
SP-53X30-1-DK	8.17	52.9844	29.9974	UNK
ONP1.2-37X54	5.88	53.975	36.4998	UNK
MPT2.4-21	4.88	54.0004	2.09042	UNK
LL200-2.4-37	4.03	54.0004	3.64998	0.2286
SM811K04L	16.94	56.007	62.992	2.1082
SM811K04TF	18.64	56.007	62.992	1.4986
SM351K09L	17.45	56.896	64.008	1.778
SM351K09TF	19.1	56.9976	64.008	1.4986
SM351K09TFS	19.1	56.9976	64.008	1.4986
SM351K09LS	17.36	56.9976	64236.6	2.1082
AM-8801CAR-DGK-T	11.16	57.7088	55.0926	2.0066
AM-8701CAR-DGK-T	11.26	57.7088	55.0926	2.0066
AM-8702CAR-DGK-T	14.25	57.7088	41.3004	2.0066
SM301K09L	17.36	57.912	57.912	2.032
AM-1815CA-DGK-E	5.14	58.0898	48.5902	1.0922
3604	8.7	59.9948	48.006	2.9972
AM-5608CAR	86.7	60.0964	41.3004	1.8796
SC10036	6.99	60.96	60.96	5.08
SM111K09L	6.88	61.976	2.1082	2.032

SM141K09L	8.67	62.992	23.0124	2.1082
SM141K09TF	9.53	62.992	23.0124	1.4986
SM141K09LS	8.67	62.992	23.0124	2.1082
SM141K09TFS	9.53	62.992	23.0124	1.4986
SP3-37	4.97	64.008	36.8046	0.2286
SM261K10TF	16.32	64.008	4.50088	1.4986
P123	9	65.9892	65.9892	2.8956
PRT-18725	10.12	65.9892	65.9892	2.8956
SM261K05TF	10.39	66.9544	24.9936	1.4986
SM261K05L	8.78	67.0052	24.9936	2.1082
SP-68X37-4-DK	27.69	67.9958	36.9824	2.0066
313070004	6.07	69.85	54.991	1.4986
AM-5706CAR-DGK-T	20.29	69.85	50.0126	2.0066
MIKROE-651	19.2	69.85	64.9986	3.2004
SM141K10L	9.39	70.0024	23.0124	2.1082
SM141K10LS	9.39	70.0024	23.0124	2.1082
SM141K10TFS	9.96	70.0024	23.0124	1.4986
SM141K10TF	9.96	70.231	23.0124	1.4986
FIT1002	21.9	71.501	59.9948	0.2032
MPT3.6-75	9.09	73.8886	72.898	0.2286
MPT3.6-150	15.29	73.8886	145.9992	0.2286
LL200-3.6-75	8.42	73.9902	72.9996	0.2286
AM-5907CAR-DGK-T	20.76	75.0062	54.991	2.0066
SC10050	7.99	76.2	60.96	5.08
SP-80X60-1-DK	11.88	79.9846	59.9948	2.0066
SP-80X60-4-DK	34.45	79.9846	59.9948	UNK
SC20036	10.99	83.82	83.82	5.08
SP4.2-37	4.98	84.0994	36.8046	0.2286
SM111K08L	7.98	87.122	13.97	2.032
SM141K08TF	8.78	88.011	15.0114	1.4986
SM141K08L	7.98	88.011	15.0114	2.1082
SM141K08LS	7.98	88.011	15.0114	2.1082
SM141K08TFS	8.78	88.011	15.0114	1.4986
SM451K10L	21.86	88.90	67.056	2.032
LL200-2.4-75	4.53	88.90	73.70	UNK
SM531K10L	21.86	89.0016	64.9986	1.8034
SM531K08L	19.6	89.0016	51.9938	1.8034
SM811K08L	28.16	89.0016	78.0034	2.1082
SM531K10TF	23.31	89.0016	64.9986	1.4986
SM811K08TF	30.96	89.0016	78.0034	1.4986
SM531K08LS	19.6	89.0016	51.9938	2.1082
SM531K08TFS	21.57	89.0016	51.9938	1.4986
SM531K10LS	21.86	89.0016	64.9986	2.1082
SM531K10TFS	24.02	89.0016	64.9986	0.14986
SLMD481H10L	31.21	89.0016	67.0052	2.0066

SM531K08TF	21.11	89.0016	51.9938	1.4986
SM451K12L	28.71	89.916	78.994	2.032
SM531K12L	28.71	89.9922	78.994	2.0066
SM531K12LS	28.71	89.9922	78.994	2.1082
SM531K12TFS	31.58	89.9922	78.994	1.4986
SLMD481H12L	35.07	89.9922	78.994	2.0066
SM401K08L	17.19	89.9922	40.005	2.1082
SM401K08TF	19.81	89.9922	40.005	1.4986
SM531K12TF	31.58	89.9922	78.994	1.4986
P134	5.5	94	32.5	2.2
MPT4.8-75	10.41	94.00	72.90	0.2286
ONP2.4-15X94	4.92	94.0054	14.605	UNK
LL200-4.8-37	7.54	94.0054	36.4998	0.2286
INP2.4-10X94T	9.18	94.0054	10.0076	0.2286
MPT4.8-150	18.04	94.0054	146.0246	0.2286
AM-1816CA-DGK-E	8.55	96.6978	56.6928	1.0922
313070005	12.3	100.00	80.00	UNK
3605	28.44	110.0074	110.0074	0.29972
SM641K10L	28.3	110.998	62.992	0.9652
SM641K10TF	31.14	110.998	62.992	1.4986
P124	14	113.0046	65.9892	2.8956
PRT-18726	15.75	113.0046	65.9892	2.8956
5368	18.69	113.0046	65.9892	2.8956
MP3-25	4.88	113.9952	24.003	0.2286
MP3-37	7.62	113.9952	36.8046	0.2286
LL200-3-37	5.33	113.9952	36.8046	UNK
MPT6-75	11	113.9952	72.898	0.2286
MPT6-150	20.75	113.9952	145.9992	0.2286
SC20050	12.99	119.38	71.12	5.08
SC10072	9.99	119.9896	59.9948	UNK
SM102K06L	26.92	132.9944	51.9938	2.1082
SM102K06TF	29.7	132.9944	51.9938	1.4986
P126	21	135.9916	111.9886	2.9972
5366	26.19	135.9916	111.9886	2.9972
313070002	18.53	137.16	81.0006	1.4986
SC20072	15.99	139.7	88.9	5.08
AM-5902CAR-DGK-T	27.57	150.114	37.4904	2.0066
313070001	37.21	160.02	137.9982	1.4986
PIS-0569	185.22	175.006	175.006	35.0012
313070003	24.76	180.0098	80.01	2.4892
SM881K04L	18.5	199.9996	19.9898	2.1082
SM881K04TF	20.23	199.9996	19.9898	1.4986
P105	35	221.996	137.0076	3.9878
5367	43.69	223.012	148.0058	3.5052
MPT15-75	19.27	253.0094	72.898	UNK

MP7.2-75F	22.46	253.0094	72.898	UNK
MPT15-150	39.79	253.0094	145.9992	UNK
MP7.2-150F	40.07	253.0094	145.9992	UNK
PT15-300	95.43	270.002	324.8914	1.1176
RC7.2-75F	30.32	270.002	89.8906	UNK
PT15-150	63.83	270.002	175.006	1.1176
RC7.2-75 PSAF	15.8	270.002	89.8906	UNK
PT15-75	41.94	270.002	100.1014	UNK
P7.2-150F	56.61	270.002	175.006	UNK
5369	81.19	273.9898	223.012	3.9878
SM182K01L	12.89	284.988	8.001	2.1082
PIS-0571	410.37	289.9918	189.992	40.005
SM281K07L	12.77	299.9994	7.493	2.1082
INP3.6-12X310	4.81	310.007	11.8618	0.2286
PIS-0572	152.61	340.0044	210.0072	54.991
114992711	65	350.012	250.0122	24.9936
TPS-12-15W	65.94	370	360	17
29250	199.99	520.7	50.8	927.1
TPS-24-30	2758.98	540.9946	510.0066	29.9974
TPS-12-35W	125.14	545.0078	489.9914	35.0012

Appendix C: Structural Parts Specifications

Table C.1.Structrual parts directory. All dimensions are in mm.

Part	Location
Baseplate	Figure C.1
Connector Baseplate	Figure C.2
Extended Baseplate	Figure C.3 - Figure C.6
Extended Connector Baseplate	Figure C.7
Mounting Plate	Figure C.8
Chassis Wall	Figure C.9
Modified Chassis Wall	Figure C.10
Secondary Hinge	Figure C.11
Modified Secondary Hinge	Figure C.12
Primary Hinge	Figure C.13
Rotary Disk	Figure C.14
NEMA 11 Mounting Bracket	Figure C.15

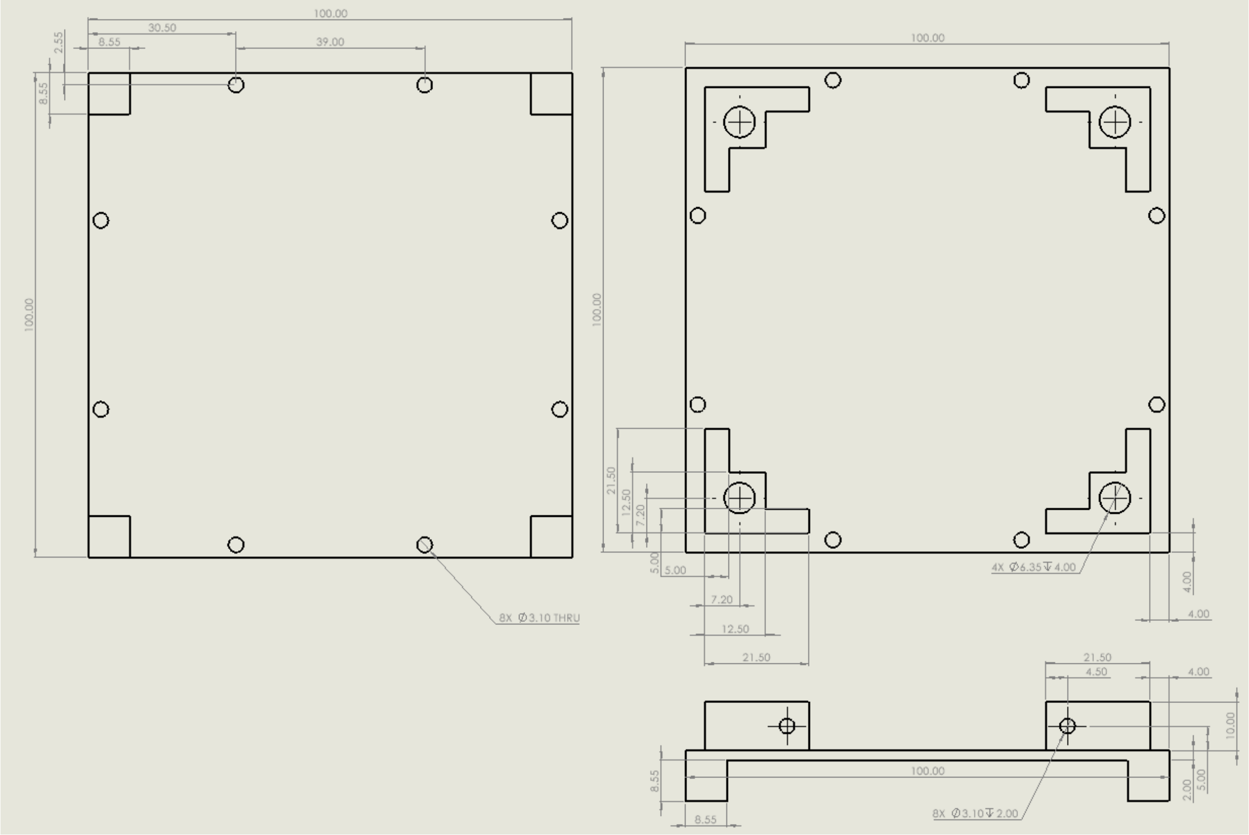


Figure C.1. Specifications of the CubeSat baseplate (bottom, top, and right view).

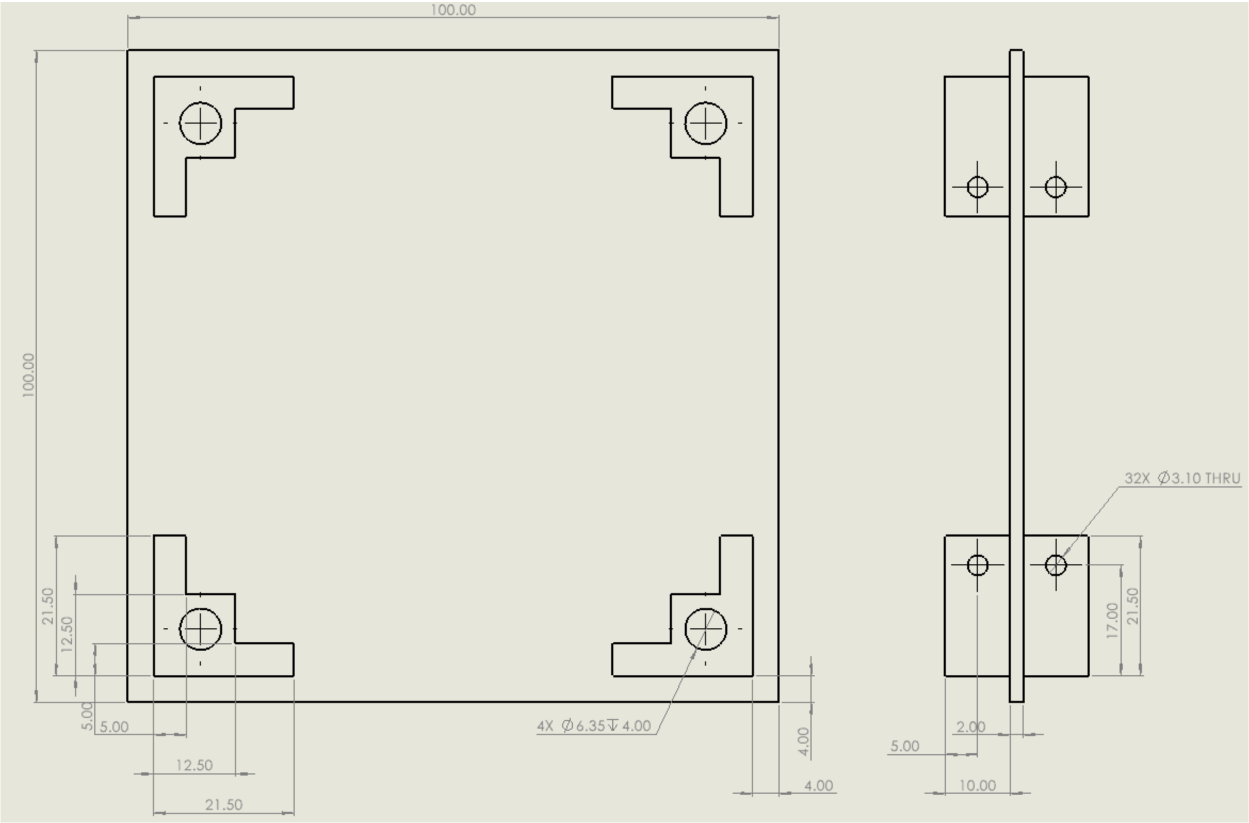


Figure C.2. Specifications of the CubeSat connector baseplate (top and right view).

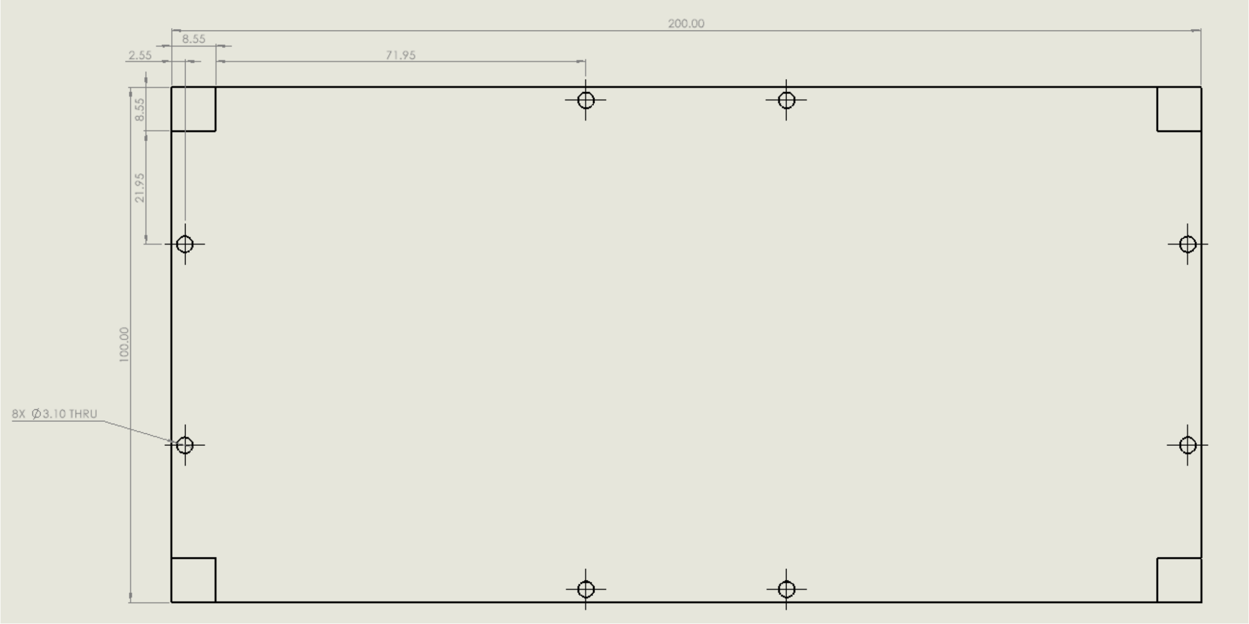


Figure C.3. Specifications of the CubeSat extended baseplate (bottom view).

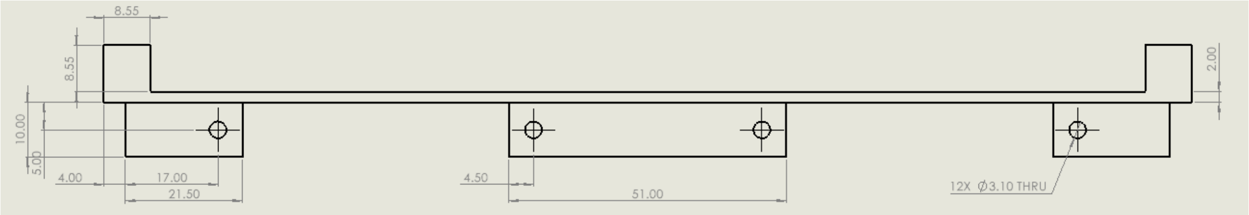


Figure C.4. Specifications of the CubeSat extended baseplate (front view).

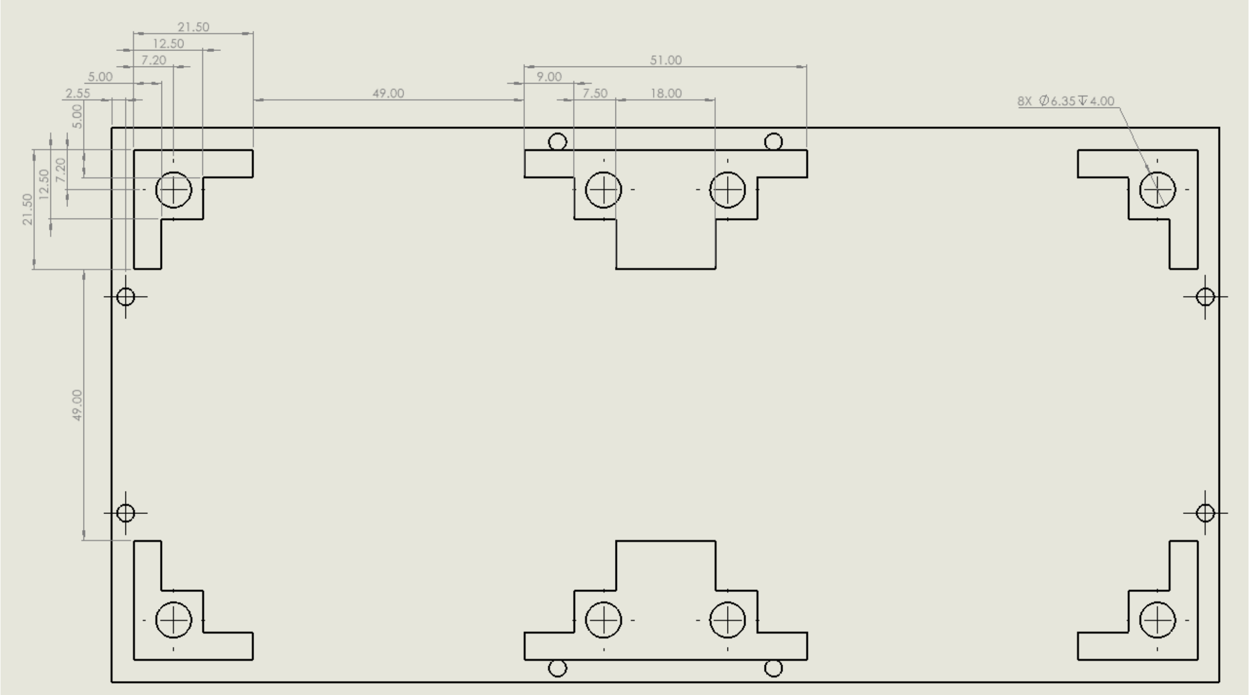


Figure C.5. Specifications of the CubeSat extended baseplate (top view).

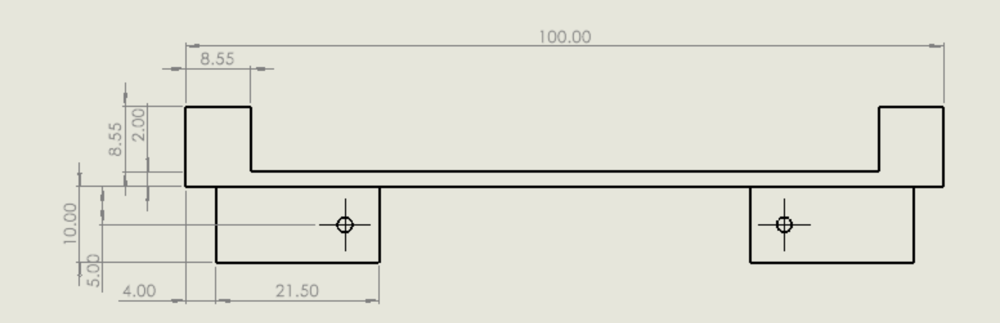


Figure C.6. Specifications of the CubeSat extended baseplate (right view).

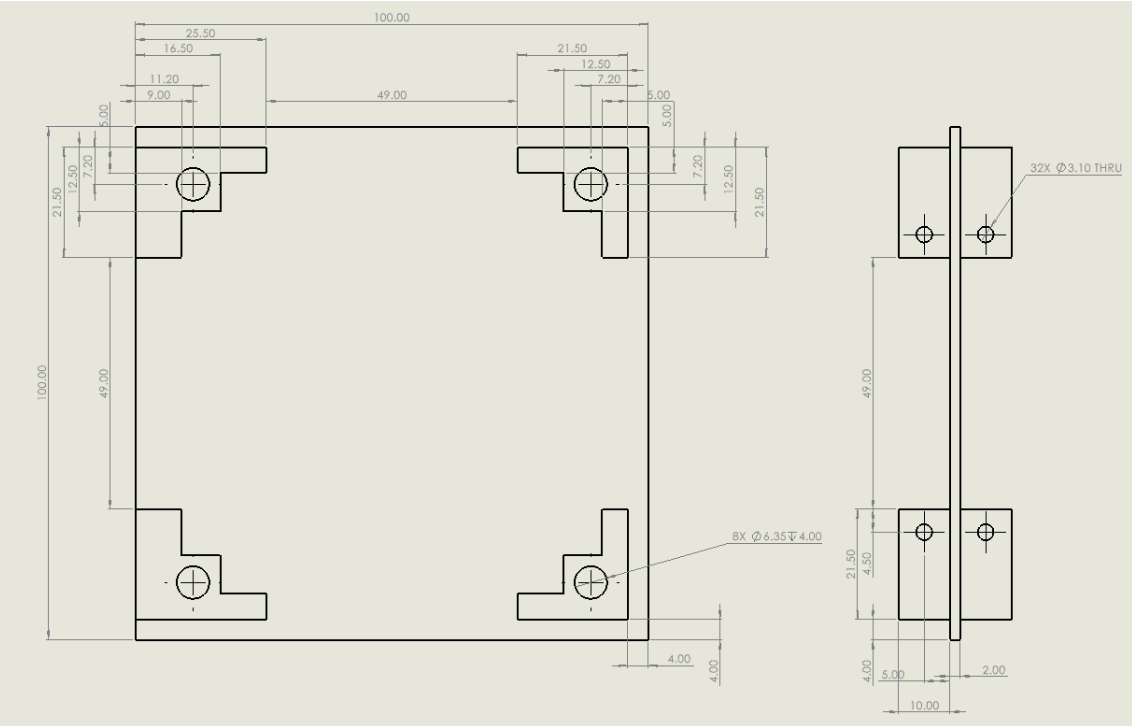


Figure C.7. Specifications of the CubeSat extended connector baseplate.

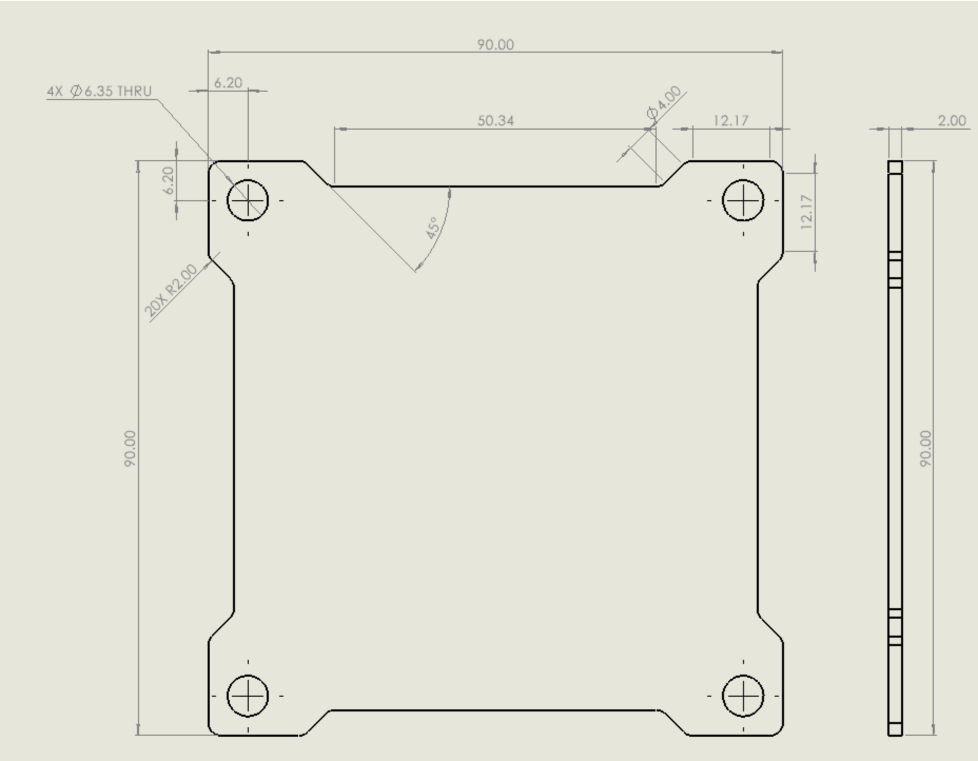


Figure C.8. Specifications of the mounting plate.

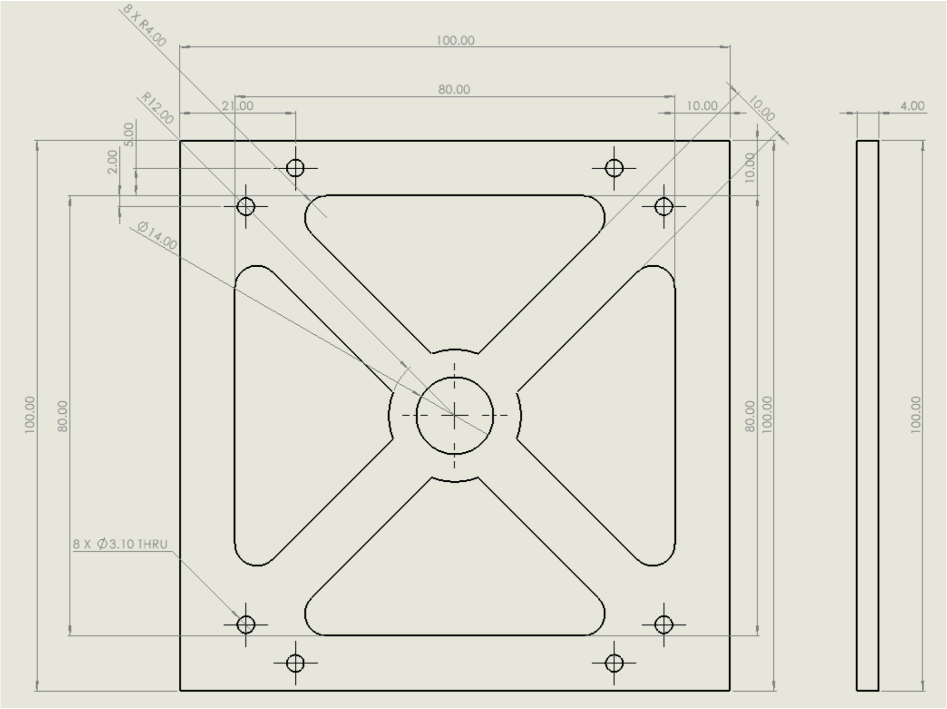


Figure C.9. Specifications of the chassis wall.

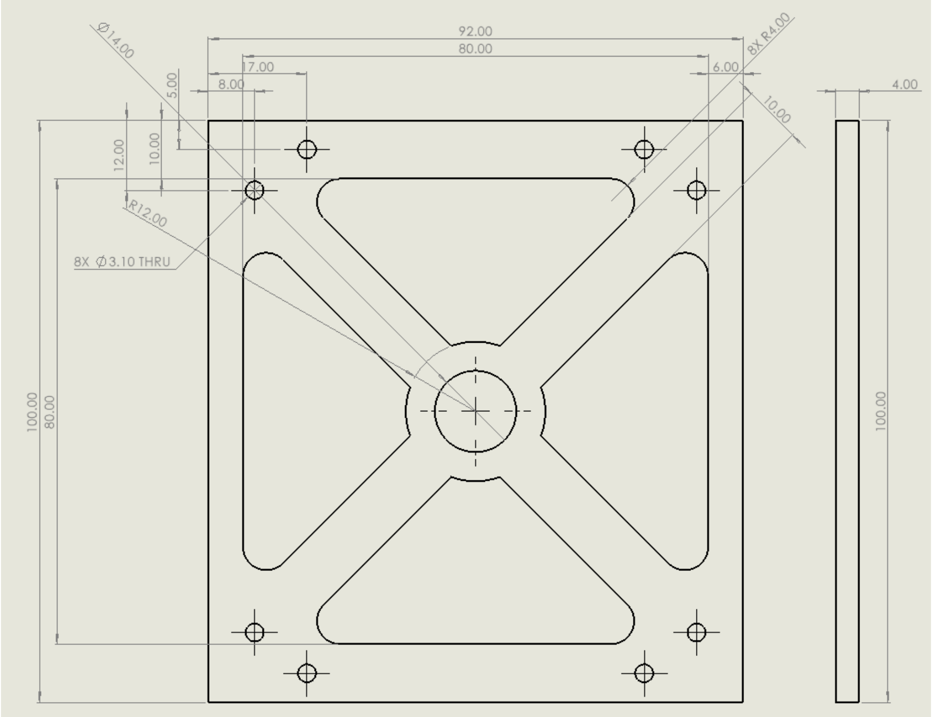


Figure C.10. Specifications of the modified chassis wall.

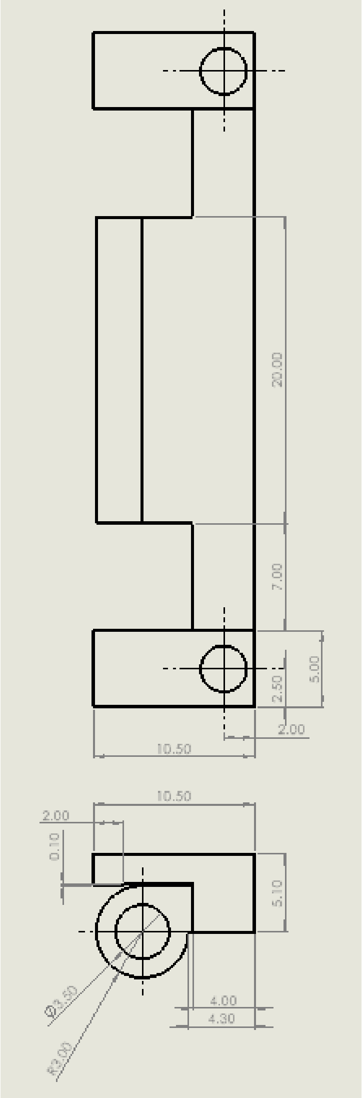


Figure C.11. Specifications of the primary hinge.

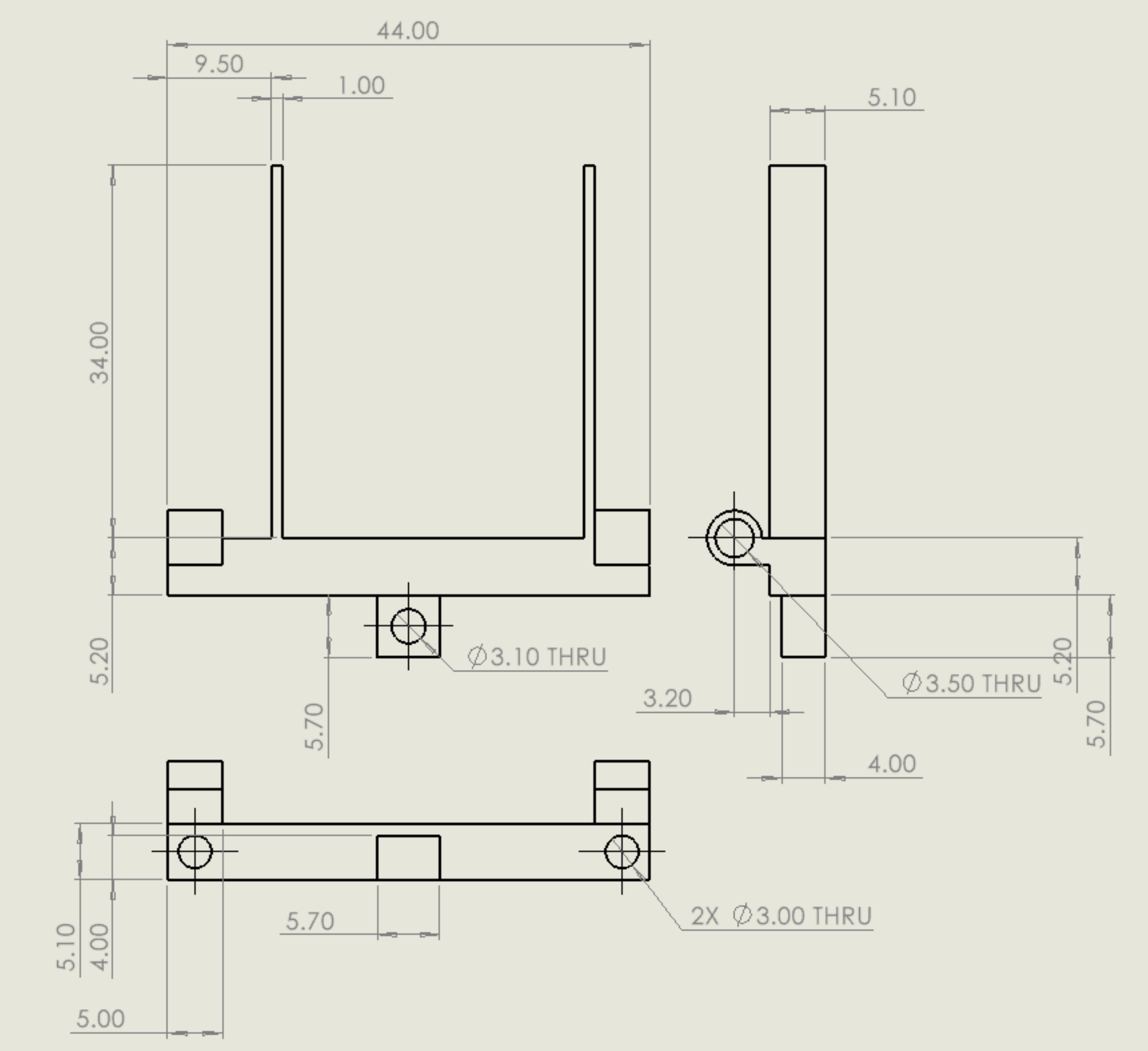


Figure C.12. Specifications of the modified secondary hinge.

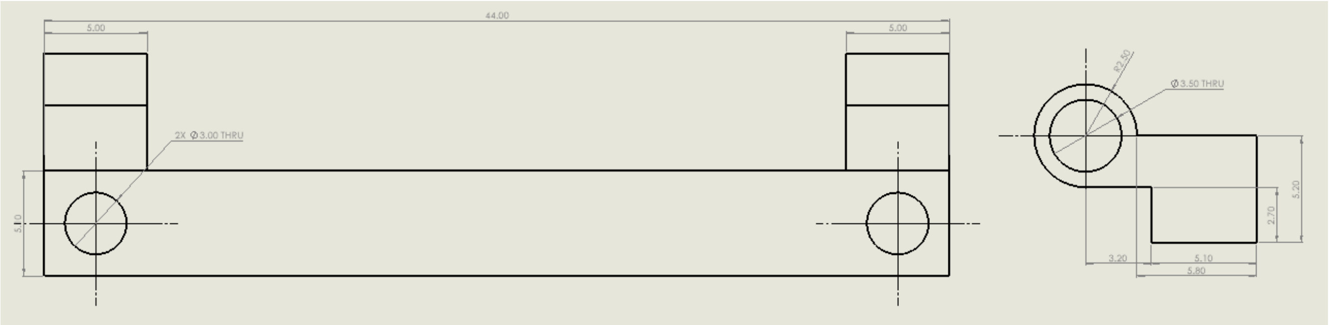


Figure C.13. Specifications of the secondary hinge.

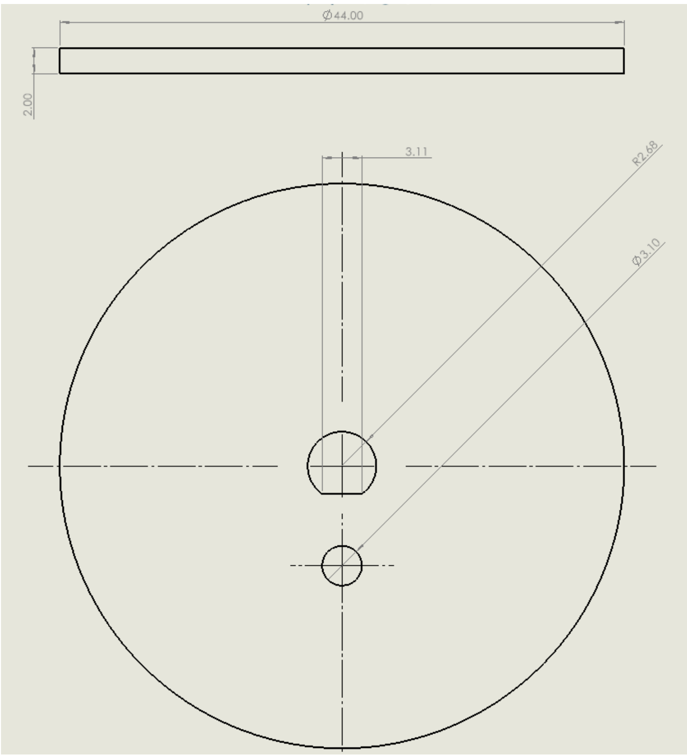


Figure C.14. Rotary disk specifications.

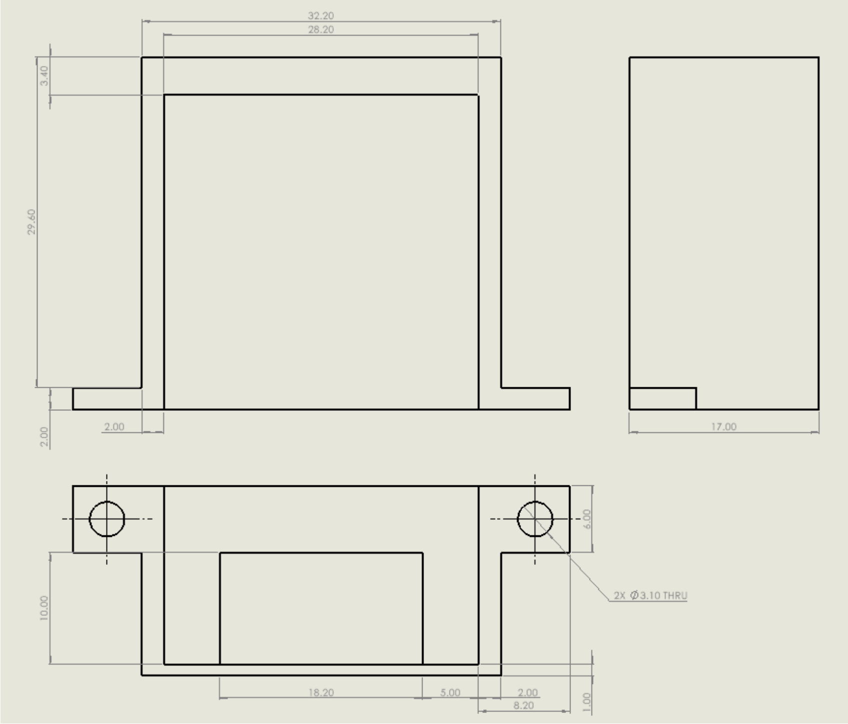


Figure C.15. NEMA 11 mounting bracket specifications.

Appendix D: EPS Schematic

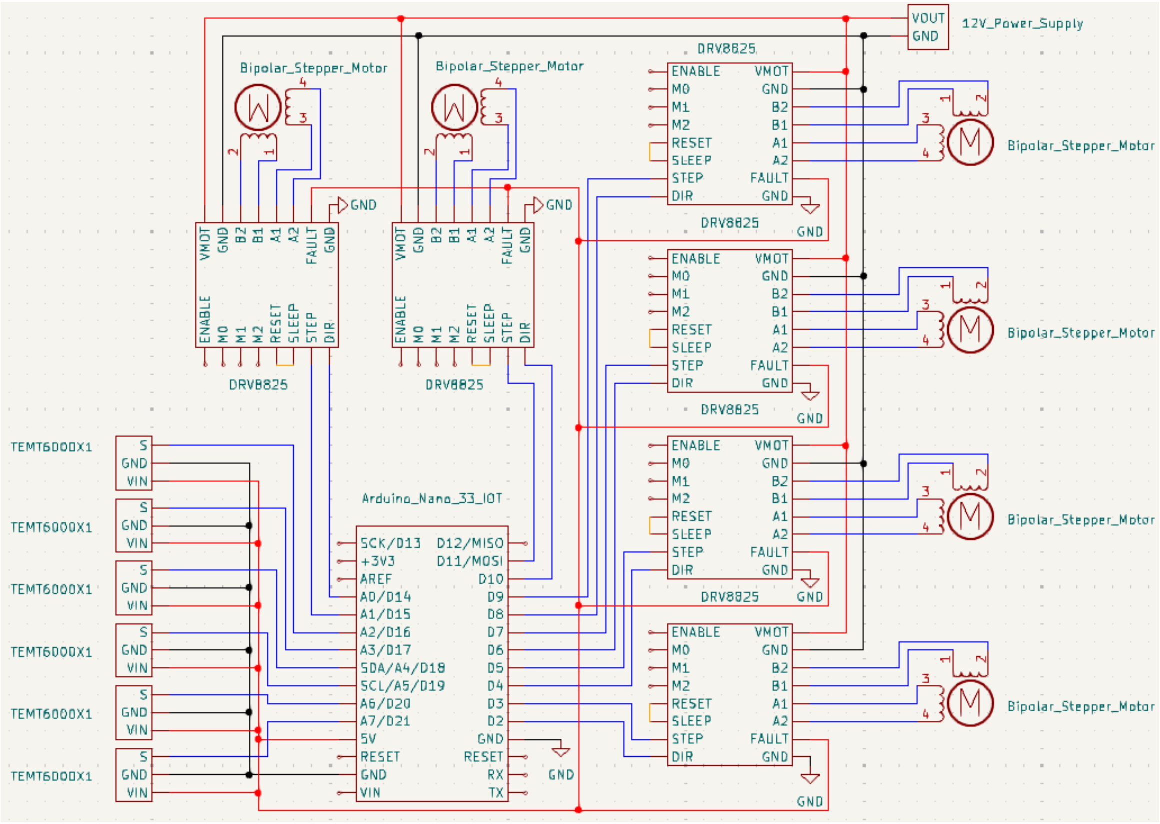


Figure D.1. Prototype schematic of the EPS.

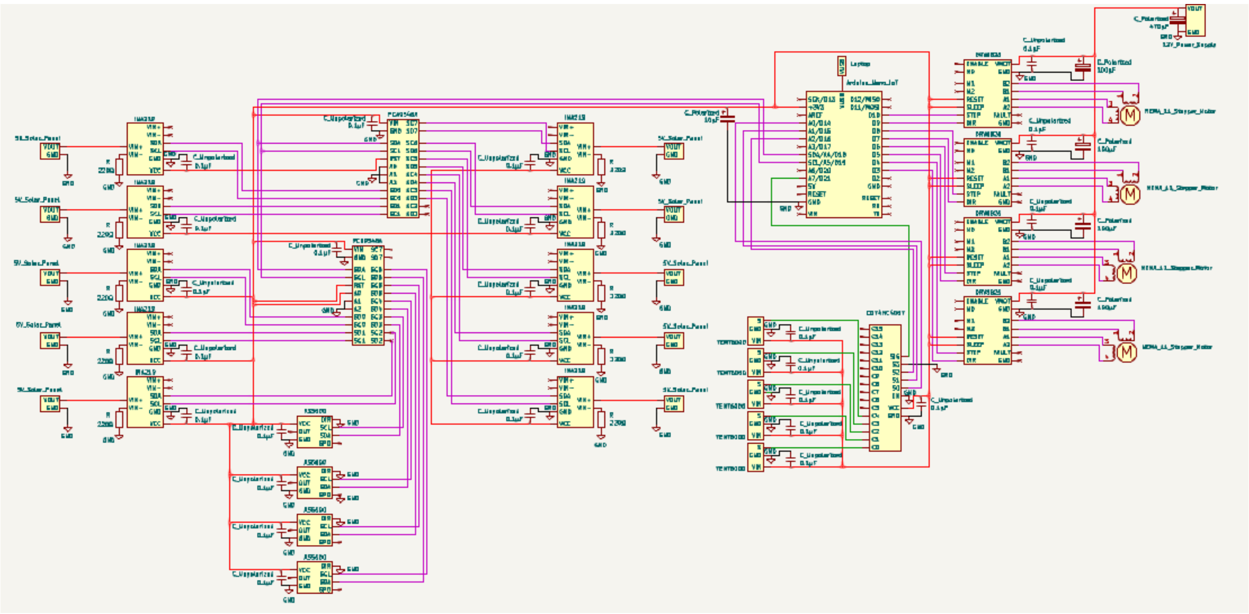


Figure D.2. Current schematic of the EPS.

Appendix E: Project Expenditures

Table E.1. Total expenditures during the course of this project: $625.11.

Item	Quantity	Unit Cost ($)	Cost (4)
3D Printer	1	239	239
PETG HF Filament	1	23	23
M2 Inserts (2L:120 - 4L:100)	1	9.98	9.98
M2 Flat Head Screws (30)	2	8.59	17.18
DC Power Supply	1	49.49	49.49
Multimeter	1	36.99	36.99
Resistors (40)	1	9.99	9.99
90 Degree Torsion Spring (10)	1	7.99	7.99
180 Degree Torsion Spring (10)	1	7.99	7.99
Precision Scale	1	8.39	8.39
M3.5 Hex Nut (100)	1	9.29	9.29
Solar Panel (10)	1	15.59	15.59
Voltage and Current Sensor (10) [INA219]	1	19.99	19.99
Digital (I2C) Multiplexer [TCA9548A] (Voltage and Current Sensors) [6]	1	9.99	9.99
Analog Multiplexer [CD74HC4067] (Light Sensors) [12]	1	9.99	9.99
NEMA 11 Stepper Motor	4	17.99	71.96
Motor Driver (DRV8825) [5]	1	14.49	14.49
Light Sensor (TEMT6000) [5]	1	9.99	9.99
Magnetic Encoder (AS5600) [3]	1	9.99	9.99
100 uF Capacitors (Polarized) [Radial Electrolytic] [100 ufd 35V, 10x13]	20	0.19	3.8
10 uF Capacitors (Polarized) [Radial Electrolytic] [10ufd 35V, 4x8, 105]	2	0.12	0.24
470 uF Capacitors (Polarized) [Radial Electrolytic]	2	0.35	0.7
0.1 uF Capacitors (Unpolarized) [Monolithic]	30	0.14	4.2
830 Tie Point Breadboard [165.1 x 53.975 x 9.144 mm]	1	7.95	7.95
1N5817 1A 20V Schottky Diode	10	0.3	3
Breadboard Jumper Wires	6	3.99	23.94

Appendix F: COTS Torsion Spring Data

Table F.1. COTS 90° torsion spring specifications acquired from amazon.com. The material for all torsion springs was stainless steel.

d_{wire} (mm)	d_{outer} (mm)	d_{inner} (mm)	D_{mean} (mm)	n	C
0.5	4.5	3.5	4	6	8
0.5	4.5	3.5	4	3	8
0.5	4.5	3.5	4	6	8
0.5	4.5	3.5	4	6	8
0.5	5	4	4.5	6	9
0.5	5	4	4.5	3	9
0.5	5	4	4.5	3	9
0.5	6	5	5.5	6	11
0.5	6	5	5.5	9	11
0.8	6	4.4	5.2	4	7
1.2	6	3.6	4.8	3	4
0.5	8	7	7.5	6	15
0.5	8	7	7.5	6	15
0.5	8	7	7.5	6	15
1	8	6	7	4	7
1	8	6	7	4	7
1	8	6	7	4	7
1	9	7	8	5	8
1	9	7	8	4	8
1.2	9	6.6	7.8	5	7
1.2	10	7.6	8.8	3	7
1.2	10	7.6	8.8	3	7
1.2	10	7.6	8.8	4	7
1.26	10	7.48	8.74	4	7
1.5	10	7	8.5	5	6
1.5	10	7	8.5	10	6
1.5	10	7	8.5	5	6
2	10	6	8	5	4
2	10	6	8	10	4
2	10	6	8	5	4
1	11	9	10	5	10
1.2	11	8.6	9.8	5	8
1.2	11	8.6	9.8	4	8
1.2	11	8.6	9.8	5	8
2.5	15	10	12.5	3	5
2.5	15	10	12.5	15	5
2.5	15	10	12.5	3	5
1.2	16	13.6	14.8	5	12

Appendix G: Solar Panel Test Data

The unprocessed solar panel test data is persevered within this Github repository for accessibility purposes. A direct Github link to each dataset is provided in the following table. Each data set is comprised of elapsed time, multiplexer, channel, voltage (V), current (mA), and power (W), respectively. As previously discussed, the open circuit voltage of each solar panel was used to examine the individual voltage generation capabilities of each solar panel. Subsequently, the current and power columns should be disregarded. Lastly, the initial start time does not start at zero because there was a delay connecting the USB to the MCU, connecting the serial data in Coolterm, and creating the text file. This is resolved by simply subtracting the time of the initial data point from every time data point.

Table G.1. Github links to each solar panel test configuration.

Dataset Link	Data points
Stowed Data	8,731
Partially Deployed Data	8,731
Fully Deployed Data	9,423

Appendix H: Arduino and MATLAB Code

For the preservation of formatting and ease of accessibility, all code used throughout this project is stored within this Github repository. A direct Github link and description for each program is provided in the following table. Alternatively, shortcuts to the raw code present within the document is also provided. During component-level testing, example sketches within the Arduino library was used to verify the functionality of the INA219 current sensor and AS5600 magnetic encoder; these sketches are not included.

Table H.1. MATLAB and Arduino program GitHub links, program raw code, and descriptions.

Program Name & Github link	Raw Code	Description
IncidenceCalculations.m	Table H.2	This program calculates G_{eff} based on the incident angle of a solar panel and calculates the power the solar panel generates.
TemperatureCalculations.m	Table H.3	This program improves the accuracy of the IncidenceCalculations.m solar panel power calculations by considering the impact of T_{cell}. G_{eff} plots are not included.
Torsion_Spring_Calculations.m	Table H.4	This program calculates the shear and bending stress of a torsion spring.
TEMT6000_Component_Test _With_Noise_Reduction.m	Table H.5	Removes the extremum values of a 50 point ADC value data set from a TEMT6000 sensor. The trimmed values are then averaged.
Solar_Panel_Data_Analysis.m	Table H.6	Parses through the stowed, partially deployed, and fully deployed data sets obtained during solar panel testing. A moving average was applied during the plot creation process for smoothing purposes.
TEMT6000_Sensor.ino	Table H.7	This program measures the ADC values of the TEMT6000 based on the ambient light level.
Analog_Multiplexer_Test.ino	Table H.8	This program tests the TEMT6000 LSA and CD74HC4067 analog multiplexer.
I2C_Multiplexer_Test.ino	Table H.9	This program tests the INA219 current sensors and the PCA9548A I2C multiplexer.

NEMA_11_Stepper_Motor.ino	Table H.10	This program tests whether the NEMA 11 motor could rotate CW and CCW.

Table H.2. IncidenceCalculations.m MATLAB program.

```
%{
 Christopher Hernandez
 Development of a Solar Array Deployment Mechanism for a CubeSat
 Effective Irradiance and Power Calculator
 November 2024
 This program calculates effective irradiance based on the incident angle
 and the subsequent power generated by a solar panel.
%}

clc, clear, close all

%% Variables
G = 1361; % w/m^2
Area = 0.0192; % m^2
Efficiency = 0.15;

% Booleans
closeFigures = 1;

%% One Face
% Calculations
iAngleVector1 = linspace(0,90,91);
GeffMatrix = zeros(2,91);
GeffVector = zeros(1,length(iAngleVector1));
for i=1:length(iAngleVector1)
    GeffMatrix(1,i) = G * cos(deg2rad(iAngleVector1(i))); % Face 1
    GeffMatrix(2,i) = iAngleVector1(i); % Incidence Angle 1
    GeffMatrix(3,i) = GeffMatrix(1,i) * Area * Efficiency; % Power Output
end

% Plots
% Effective irradiance plot
figure("Name","Effective Irradiance vs. Incidence Angle (One Face)")
plot(GeffMatrix(2,:),GeffMatrix(1,:),"LineWidth",2)
title("Effective Irradiance vs. Incidence Angle")
xlabel("Incidence Angle (°)")
ylabel("Effective Irradiance (W/m^2)")
grid on, grid minor

% Power Plot
```

```
figure("Name","Power vs. Incidence Angle (One Face)")
plot(GeffMatrix(2,:),GeffMatrix(3,:),"LineWidth",2)
title("Power vs. Incidence Angle")
xlabel("Incidence Angle (°)")
ylabel("Power (W)")
grid on, grid minor

fprintf("One Face\n")
fprintf("Maximum Irradiance (1 Face): %f (W/m^2) \n",max(GeffMatrix(1,:)) )
fprintf("Maximum Power (1 Face): %f (W) \n\n",max(GeffMatrix(3,:)) )

%% Two Faces
% Calculations
iAngleVector1 = linspace(0,90,91);
iAngleVector2 = linspace(90,0,91);
GeffMatrix = zeros(6,91);
for i=1:length(iAngleVector1)
   GeffMatrix(1,i) = G * cos(deg2rad(iAngleVector1(i))); % Face 1
   GeffMatrix(2,i) = G * cos(deg2rad(iAngleVector2(i))); % Face 2
   GeffMatrix(3,i) = GeffMatrix(1,i) + GeffMatrix(2,i); % Combined Irradiance
   GeffMatrix(4,i) = iAngleVector1(i); % Incidence Angle 1
   GeffMatrix(5,i) = iAngleVector2(i); % Incidence Angle 2
   GeffMatrix(6,i) = GeffMatrix(3,i) * Area * Efficiency; % Power Output
end

% Plots
% Effective irradiance plot
figure("Name","Effective Irradiance vs. Incidence Angle (Two Faces)")
t = tiledlayout(1,1);
ax1 = axes(t);
plot(ax1,GeffMatrix(4,:),GeffMatrix(3,:),"LineWidth",2)
title("Effective Irradiance vs. Incidence Angle","Units","Normalized","Position",[0.5, 1.05, 0])
xlabel("Incidence Angle (°)")
ylabel("Effective Irradiance (W/m^2)")

ax2 = axes(t);
plot(ax2,GeffMatrix(5,:),GeffMatrix(3,:),"LineWidth",2)
set(gca, "xdir","reverse")
ax2.XAxisLocation = "top";
ax2.YAxisLocation = "right";
ax2.Color = "none";
ax1.Box = "off";
ax2.Box = "off";
grid on, grid minor
```

```
% Power Plot
figure("Name","Power vs. Incidence Angle (Two Faces)")
t = tiledlayout(1,1);
ax1 = axes(t);
plot(ax1,GeffMatrix(4,:),GeffMatrix(6,:),"LineWidth",2)
title("Power vs. Incidence Angle","Units","Normalized","Position",[0.5, 1.05, 0])
xlabel("Incidence Angle (°)")
ylabel("Power (W)")

ax2 = axes(t);
plot(ax2,GeffMatrix(4,:),GeffMatrix(6,:),"LineWidth",2)
set(gca, "xdir","reverse")
ax2.XAxisLocation = "top";
ax2.YAxisLocation = "right";
ax2.Color = "none";
ax1.Box = "off";
ax2.Box = "off";
grid on, grid minor

fprintf("Two Faces\n")
fprintf("Maximum Irradiance (2 Faces): %f (W/m^2) \n",max(GeffMatrix(3,:)) )
fprintf("Maximum Power (2 Faces): %f (W) \n\n",max(GeffMatrix(6,:)) )

%% Three Faces
% Calculations
iAngleVector1 = linspace(0,45,91);
iAngleVector2 = linspace(90,45,91);
iAngleVector3 = linspace(90,45,91);
GeffMatrix = zeros(8,91);
matrixPosition = 0;
for i=1:length(iAngleVector1)
   GeffMatrix(1,i) = G * cos(deg2rad(iAngleVector1(i))); % Face 1
   GeffMatrix(2,i) = G * cos(deg2rad(iAngleVector2(i))); % Face 2
   GeffMatrix(3,i) = G * cos(deg2rad(iAngleVector3(i))); % Face 3
   GeffMatrix(4,i) = GeffMatrix(1,i) + GeffMatrix(2,i) + GeffMatrix(3,i); % Combined
Irradiance
   GeffMatrix(5,i) = iAngleVector1(i); % Incidence Angle 1
   GeffMatrix(6,i) = iAngleVector2(i); % Incidence Angle 2
   GeffMatrix(7,i) = iAngleVector3(i); % Incidence Angle 3
   GeffMatrix(8,i) = GeffMatrix(4,i) * Area * Efficiency; % Power Output
end

% Plots
% Effective Irradiance Plot
figure("Name","Effective Irradiance vs. Incidence Angle (Three Faces)")
plot(GeffMatrix(7,:),GeffMatrix(4,:),"LineWidth",2)
```

```
title("Effective Irradiance vs. Incidence Angle")
xlabel("Incidence Angle (°)")
ylabel("Effective Irradiance (W/m^2)")
grid on, grid minor

% Power Plot
figure("Name","Power vs. Incidence Angle (Three Faces)")
plot(GeffMatrix(7,:),GeffMatrix(8,:),"LineWidth",2)
title("Power vs. Incidence Angle")
xlabel("Incidence Angle (°)")
ylabel("Power (W)")
grid on, grid minor

fprintf("Three Faces\n")
fprintf("Maximum Irradiance (3 Faces): %f (W/m^2)\n",max(GeffMatrix(4,:)) )
fprintf("Maximum Power (3 Faces): %f (W)\n\n",max(GeffMatrix(8,:)) )

if closeFigures == 1
   close all
end
```

Table H.3. TemperatureCalculations.m MATLAB program.

```
%{
   Christopher Hernandez
   Development of a Solar Array Deployment Mechanism for a CubeSat
   Temperature Calculator
   November 2024
   This program calculates power losses at the optimal incident angle for
   each degree change in temperature.

   Cases
   - When simulating one irradiated face, the iAngle should be set to 0°
   and oneFace should be set to 1.
   - When simulating two irradiated faces, the iAngle should be set to 45°
   and twoFace should be set to 2.
   - When simulating three irradiated faces, the iAngle should be set to 45°
   and twoFace should be set to 3.

%}

clc, clear, close all
```

```
%% Variables
G = 1361; % w/m^2
Area = 0.1600; % m^2
Efficiency = 0.15;
lamda = 0.045;
iAngle = 0;

% Booleans
oneFace = 0;
twoFaces = 0;
threeFaces = 1;
closeFigures = 1;

% Calculations
tCellVector = linspace(25,85,61);
pVector = zeros(1,length(tCellVector));
epsilonVector = zeros(1,length(tCellVector));
Geff = G*cos(deg2rad(iAngle));

systemMatrix = zeros(3,length(tCellVector));
for i=1:length(tCellVector)
   epsilonVector(1,i) = lamda*(tCellVector(i) - 25)*Efficiency;
   if oneFace == 1
      pVector(1,i) = Geff * Area * Efficiency * (1 - lamda*(tCellVector(i) - 25));
   elseif twoFaces == 1
      pVector(1,i) = 2 * Geff * Area * Efficiency * (1 - lamda*(tCellVector(i) - 25));
   elseif threeFaces == 1
      pVector(1,i) = 3 * Geff * Area * Efficiency * (1 - lamda*(tCellVector(i) - 25));
   end
end
systemMatrix(1,:) = tCellVector;
systemMatrix(2,:) = pVector;
systemMatrix(3,:) = epsilonVector;

% Plots
figure("Name","Temperature vs. Power")
plot(tCellVector,pVector);
title("Temperature vs. Power")
xlabel("Cell Temperature (°C)")
ylabel("Power (W)")
ylim([0, max(pVector(1,:))])
grid on, grid minor

fprintf("Maximum Power: %f (W)\n\n",max(systemMatrix(2,:)) ) % STC
fprintf("Power +5°C: %f (W)\n\n",max(systemMatrix(2,6))) % 30°C
fprintf("Power +10°C: %f (W)\n\n",max(systemMatrix(2,11))) % 35°C
fprintf("Power +15°C: %f (W)\n\n",max(systemMatrix(2,16))) % 40°C

if closeFigures == 1
   close all
end
```

Table H.4. The torsion spring calculator calculates the shear and bending stress based on certain characteristics of a torsion spring.

```
%{
   Christopher Hernandez
   Development of a Solar Array Deployment Mechanism for a CubeSat
   Torsion Spring Calculator
   March 2025
   This program calculates the shear and bending stress of a torsion
   spring.
%}

clc, clear, close all

%% Constants
E = 195*10^9; % Young's Modulus (Pa)
d = 0.35/10^3; % Wire diameter (m)
n = 5; % Number of coils
outerDiameter = 11/10^3; % m
theta = 90; % Deployment angle (Degrees)
m = 0.02268; % Solar panel mass (kg)
L = 100/10^3; % Solar panel length (m)
yieldStrength = 215 * 10^6; % Pa
shearStrength = 0.57 * yieldStrength; % Pa

% Conversions
deg2rad = pi/180;
rad2deg = 180/pi;

theta = theta*deg2rad;


%% Calculations

% Spring Constant
innerDiameter = outerDiameter - 2*d; % m
D = (outerDiameter + innerDiameter)/2 % Mean coil diameter (m)
k = (E*d^4) / (10.8*n*D) % (Nm/rad)
I = (1/3)*m*L^2 % Moment of Inertia (kg/m^2)
```

```
w = sqrt(k/I) % rad/s
T = k*theta % Torque (N*m)
t = pi/2/w % s
t2 = sqrt(I/k)*asin(90*deg2rad/theta)

C = D/d;

% Correction factors
kB = (4*C^2 - 1 - 1) / (4*C * (C-1))
kWahl = ((4*C - 1) / (4*C - 4)) + 0.615/C % Wahl correction factor
kBerg = (4*C+2)/(4*C-3) % Bergsträsser correction factor

% Stress calculations
shearStress = 16*T*kWahl*D/(pi*d^3)
bendingStress = 32*T*kWahl*D/(pi*d^3)

% Safety factor calculations
bendingSafetyFactor = yieldStrength/bendingStress
shearSafetyFactor = shearStrength/shearStress
```

Table H.5. TEMT6000 component test with noise reduction methods.

```
%{
   Christopher Hernandez
   Development of a Solar Array Deployment Mechanism for a CubeSat
   TEMT6000 Component Test with Noise Reduction
   April 2025

   This program uses MATLAB to log and visualize the plotted ADC values of
   a TEMT6000 sensor. The unprocessed and noise reduced ADC values are
   plotted.
%}

clc, clear, close all

%% Arduino initialization
a = arduino("COM3", "Nano33IoT");
lightSensorPin = "A7";

% Variables
delayTime = 0;
testTime = 60;

% Constants
voltageReference = 3.3;
adcMax = 1023;

% Animated figure
figure("Name","TEMT6000 Unit Test (Averaged Data)");
line = animatedline;
title("TEMT6000 Light Sensor Unit Test");
xlabel("Time (s)");
ylabel("ADC Value");
ylim([0 adcMax]);
grid on, grid minor

% Tables
voltageValue = [];
adcValue = [];
timeValue = [];
averageADCValue = [];
averageTimeValue = [];

% Indexes
samplingIndex = 1;
averageIndex = 1;
sampleSize = 60;
```

```
trimIndex = 1;

tic % Start time

%% Main
while true
   % Raw sampling light sensor readings
   rawSamples = zeros(sampleSize, 1);
   for ii=1:sampleSize
      voltageValue(samplingIndex,1) = readVoltage(a, lightSensorPin); % Measured voltage
      adcValue(samplingIndex,1) = round((voltageValue(samplingIndex,1) / voltageReference)
* adcMax); % ADC value
      timeValue(samplingIndex,1) = toc;
      rawSamples(ii) = adcValue(samplingIndex,1);
      samplingIndex = samplingIndex + 1;
   end
   averageTimeValue(averageIndex,1) = toc; % Elapsed time

   % Averages the raw sampling light sensor readings
   startIndex = samplingIndex - sampleSize;
   indexRange = startIndex:samplingIndex-1;

   minADCValue = min(adcValue(indexRange));
   maxADCValue = max(adcValue(indexRange));

   % Trims minimum and maximum values
   sortedSamples = sort(rawSamples);
   trimmedSamples = sortedSamples(2:end-1);

   minADCValueTrimmed = min(trimmedSamples);
   maxADCValueTrimmed = max(trimmedSamples);

   averageADCValue(averageIndex, 1) = mean(trimmedSamples);
   ylim([0 max(averageADCValue) + 10])

   % Runs the unit test for a specified amount of time
   if averageTimeValue(averageIndex,1) >= testTime
      break
   end

   % Updates animated plot
   addpoints(line, averageTimeValue(averageIndex,1), averageADCValue(averageIndex,1));
   drawnow limitrate;

   fprintf("Time (s): %f\n", averageTimeValue(averageIndex,1))
   fprintf("Index Range: %f - %f\n", startIndex, samplingIndex-1)
```

```
    fprintf("Average ADC Value: %f\n", averageADCValue(averageIndex,1))
    fprintf("Min ADC value: %f\n", minADCValue)
    fprintf("Min ADC value (Trimmed): %f\n",minADCValueTrimmed)
    fprintf("Max ADC value: %f\n", maxADCValue)
    fprintf("Max ADC value: (Trimmed) %f\n\n", maxADCValueTrimmed)

    averageIndex = averageIndex + 1;
end

%% Plots
figure("Name","TEMT6000 Unit Test (Raw Data)")
plot(timeValue,adcValue)
title("TEMT6000 Unit Test (Raw Sampling)")
xlabel("Time (s)")
ylabel("ADC Value")
grid on, grid minor

figure("Name","TEMT6000 Unit Test (Averaged Data)")
plot(averageTimeValue,averageADCValue)
title("TEMT6000 Unit Test (Average Sampling)")
xlabel("Time (s)")
ylabel("ADC Value")
grid on, grid minor

figure("Name","TEMT6000 Unit Test")
plot(averageTimeValue,averageADCValue)
title("TEMT6000 Unit Test")
xlabel("Time (s)")
ylabel("ADC Value")
hold on
plot(timeValue,adcValue)
grid on, grid minor
legend("Averaged Data","Raw Data","Location","southoutside","Orientation","Horizontal")
```

Table H.6. Solar panel test data parser.

```
%{
   Christopher Hernandez
   Development of a Solar Array Deployment Mechanism for a CubeSat
   Solar Panel Test Data Parser
   April 2025

   This program parses through the data sets of the stowed, partially
   deployed, and fully deployed configurations. The program sets the start
   time to 0 seconds and parses the original data file into elapsed
   seconds, multiplexer, channel, and voltage. During solar array
   modification the channels were switched between the stowed/partially
   deployed configurations and the fully deployed configurations. This
   issue was resolved by adjusting the indices accordingly. A moving
   average is applied to the original data set for smoothing purposes.
%}

clc, clear, close all
partiallyDeployedData = readmatrix("Partially Deployed Data.txt"); % 9,315
stowedData = readmatrix("Stowed Data.txt"); % 8,731
fullyDeployedData = readmatrix("Fully Deployed Data.txt"); % 9,423


% Sets a common start point
for jj = 1:length(fullyDeployedData)
   fullyDeployedData(jj,1) = fullyDeployedData(jj,1) - 22.31;
end

for jj = 1:length(partiallyDeployedData)
   partiallyDeployedData(jj,1) = partiallyDeployedData(jj,1) - 3.79;
end

for jj = 1:length(stowedData)
   stowedData(jj,1) = stowedData(jj,1) - 3.79;
end


% Multiplexer 1 - Channel 0
index = 1;
for ii=1:9:length(stowedData)
   % Stowed Data
   sM1C0(index,1) = stowedData(ii,1); % Elapsed time
   sM1C0(index,2) = stowedData(ii,2); % Multiplexer
   sM1C0(index,3) = stowedData(ii,3); % Channel
   sM1C0(index,4) = stowedData(ii,4); % Voltage
```

```
    % Partially Deployed Data
    pdM1C0(index,1) = partiallyDeployedData(ii,1); % Elapsed time
    pdM1C0(index,2) = partiallyDeployedData(ii,2); % Multiplexer
    pdM1C0(index,3) = partiallyDeployedData(ii,3); % Channel
    pdM1C0(index,4) = partiallyDeployedData(ii,4); % Voltage

    % Fully Deployed Data
    fdM1C0(index,1) = fullyDeployedData(ii,1); % Elapsed time
    fdM1C0(index,2) = fullyDeployedData(ii,2); % Multiplexer
    fdM1C0(index,3) = fullyDeployedData(ii,3); % Channel
    fdM1C0(index,4) = fullyDeployedData(ii,4); % Voltage

    index = index + 1;
end

% Multiplexer 1 - Channel 1
index = 1;
for ii=2:9:length(stowedData)
    % Stowed Data
    sM1C1(index,1) = stowedData(ii,1); % Elapsed time
    sM1C1(index,2) = stowedData(ii,2); % Multiplexer
    sM1C1(index,3) = stowedData(ii,3); % Channel
    sM1C1(index,4) = stowedData(ii,4); % Voltage

    % Partially Deployed Data
    pdM1C1(index,1) = partiallyDeployedData(ii,1); % Elapsed time
    pdM1C1(index,2) = partiallyDeployedData(ii,2); % Multiplexer
    pdM1C1(index,3) = partiallyDeployedData(ii,3); % Channel
    pdM1C1(index,4) = partiallyDeployedData(ii,4); % Voltage

    % Fully Deployed Data
    fdM1C1(index,1) = fullyDeployedData(ii,1); % Elapsed time
    fdM1C1(index,2) = fullyDeployedData(ii,2); % Multiplexer
    fdM1C1(index,3) = fullyDeployedData(ii,3); % Channel
    fdM1C1(index,4) = fullyDeployedData(ii,4); % Voltage
    index = index + 1;
end

% Multiplexer 1 - Channel 2
index = 1;
for ii=3:9:length(stowedData)
    % Stowed Data
    sM1C2(index,1) = stowedData(ii,1); % Elapsed time
    sM1C2(index,2) = stowedData(ii,2); % Multiplexer
    sM1C2(index,3) = stowedData(ii,3); % Channel
    sM1C2(index,4) = stowedData(ii,4); % Voltage
```

```
   % Partially Deployed Data
   pdM1C2(index,1) = partiallyDeployedData(ii,1); % Elapsed time
   pdM1C2(index,2) = partiallyDeployedData(ii,2); % Multiplexer
   pdM1C2(index,3) = partiallyDeployedData(ii,3); % Channel
   pdM1C2(index,4) = partiallyDeployedData(ii,4); % Voltage

   % Fully Deployed Data
   fdM1C2(index,1) = fullyDeployedData(ii,1); % Elapsed time
   fdM1C2(index,2) = fullyDeployedData(ii,2); % Multiplexer
   fdM1C2(index,3) = fullyDeployedData(ii,3); % Channel
   fdM1C2(index,4) = fullyDeployedData(ii,4); % Voltage

   index = index + 1;
end

% Multiplexer 1 - Channel 3
index = 1;
for ii=4:9:length(stowedData)
   % Stowed Data
   sM1C3(index,1) = stowedData(ii,1); % Elapsed time
   sM1C3(index,2) = stowedData(ii,2); % Multiplexer
   sM1C3(index,3) = stowedData(ii,3); % Channel
   sM1C3(index,4) = stowedData(ii,4); % Voltage

   % Partially Deployed Data
   pdM1C3(index,1) = partiallyDeployedData(ii,1); % Elapsed time
   pdM1C3(index,2) = partiallyDeployedData(ii,2); % Multiplexer
   pdM1C3(index,3) = partiallyDeployedData(ii,3); % Channel
   pdM1C3(index,4) = partiallyDeployedData(ii,4); % Voltage

   % Fully Deployed Data
   fdM1C3(index,1) = fullyDeployedData(ii,1); % Elapsed time
   fdM1C3(index,2) = fullyDeployedData(ii,2); % Multiplexer
   fdM1C3(index,3) = fullyDeployedData(ii,3); % Channel
   fdM1C3(index,4) = fullyDeployedData(ii,4); % Voltage

   index = index + 1;
end

% Multiplexer 1 - Channel 4
index = 1;
for ii=5:9:length(stowedData)
   % Stowed Data
   sM1C4(index,1) = stowedData(ii,1); % Elapsed time
   sM1C4(index,2) = stowedData(ii,2); % Multiplexer
   sM1C4(index,3) = stowedData(ii,3); % Channel
   sM1C4(index,4) = stowedData(ii,4); % Voltage

   % Partially Deployed Data
   pdM1C4(index,1) = partiallyDeployedData(ii,1); % Elapsed time
   pdM1C4(index,2) = partiallyDeployedData(ii,2); % Multiplexer
   pdM1C4(index,3) = partiallyDeployedData(ii,3); % Channel
   pdM1C4(index,4) = partiallyDeployedData(ii,4); % Voltage

   % Fully Deployed Data
   fdM1C4(index,1) = fullyDeployedData(ii,1); % Elapsed time
   fdM1C4(index,2) = fullyDeployedData(ii,2); % Multiplexer
   fdM1C4(index,3) = fullyDeployedData(ii,3); % Channel
   fdM1C4(index,4) = fullyDeployedData(ii,4); % Voltage

   index = index + 1;
end

% Multiplexer 2 - Channel 0
index = 1;
for ii=6:9:length(stowedData)
   % Stowed Data
   sM2C0(index,1) = stowedData(ii,1); % Elapsed time
   sM2C0(index,2) = stowedData(ii,2); % Multiplexer
   sM2C0(index,3) = stowedData(ii,3); % Channel
   sM2C0(index,4) = stowedData(ii,4); % Voltage

   % Partially Deployed Data
   pdM2C0(index,1) = partiallyDeployedData(ii,1); % Elapsed time
   pdM2C0(index,2) = partiallyDeployedData(ii,2); % Multiplexer
   pdM2C0(index,3) = partiallyDeployedData(ii,3); % Channel
   pdM2C0(index,4) = partiallyDeployedData(ii,4); % Voltage


   % Fully Deployed Data
   fdM2C0(index,1) = fullyDeployedData(ii,1); % Elapsed time
   fdM2C0(index,2) = fullyDeployedData(ii,2); % Multiplexer
   fdM2C0(index,3) = fullyDeployedData(ii,3); % Channel
   fdM2C0(index,4) = fullyDeployedData(ii,4); % Voltage

   index = index + 1;
end

% Multiplexer 2 - Channel 1
index = 1;
for ii=7:9:length(stowedData)
```

```
    % Stowed Data
    sM2C1(index,1) = stowedData(ii,1); % Elapsed time
    sM2C1(index,2) = stowedData(ii,2); % Multiplexer
    sM2C1(index,3) = stowedData(ii,3); % Channel
    sM2C1(index,4) = stowedData(ii,4); % Voltage

    % Partially Deployed Data
    pdM2C1(index,1) = partiallyDeployedData(ii,1); % Elapsed time
    pdM2C1(index,2) = partiallyDeployedData(ii,2); % Multiplexer
    pdM2C1(index,3) = partiallyDeployedData(ii,3); % Channel
    pdM2C1(index,4) = partiallyDeployedData(ii,4); % Voltage

    % Fully Deployed Data
    fdM2C1(index,1) = fullyDeployedData(ii,1); % Elapsed time
    fdM2C1(index,2) = fullyDeployedData(ii,2); % Multiplexer
    fdM2C1(index,3) = fullyDeployedData(ii,3); % Channel
    fdM2C1(index,4) = fullyDeployedData(ii,4); % Voltage

    index = index + 1;
end

% Multiplexer 2 - Channel 2
index = 1;
for ii=8:9:length(stowedData)
    sM2C2(index,1) = stowedData(ii,1); % Elapsed time
    sM2C2(index,2) = stowedData(ii,2); % Multiplexer
    sM2C2(index,3) = stowedData(ii,3); % Channel
    sM2C2(index,4) = stowedData(ii,4); % Voltage

    % Partially Deployed Data
    pdM2C2(index,1) = partiallyDeployedData(ii,1); % Elapsed time
    pdM2C2(index,2) = partiallyDeployedData(ii,2); % Multiplexer
    pdM2C2(index,3) = partiallyDeployedData(ii,3); % Channel
    pdM2C2(index,4) = partiallyDeployedData(ii,4); % Voltage

    % Fully Deployed Data
    fdM2C2(index,1) = fullyDeployedData(ii,1); % Elapsed time
    fdM2C2(index,2) = fullyDeployedData(ii,2); % Multiplexer
    fdM2C2(index,3) = fullyDeployedData(ii,3); % Channel
    fdM2C2(index,4) = fullyDeployedData(ii,4); % Voltage

    index = index + 1;
end

% Multiplexer 2 - Channel 3
index = 1;
```

```
for ii=9:9:length(stowedData)
   sM2C3(index,1) = stowedData(ii,1); % Elapsed time
   sM2C3(index,2) = stowedData(ii,2); % Multiplexer
   sM2C3(index,3) = stowedData(ii,3); % Channel
   sM2C3(index,4) = stowedData(ii,4); % Voltage

   % Partially Deployed Data
   pdM2C3(index,1) = partiallyDeployedData(ii,1); % Elapsed time
   pdM2C3(index,2) = partiallyDeployedData(ii,2); % Multiplexer
   pdM2C3(index,3) = partiallyDeployedData(ii,3); % Channel
   pdM2C3(index,4) = partiallyDeployedData(ii,4); % Voltage

   % Fully Deployed Data
   fdM2C3(index,1) = fullyDeployedData(ii,1); % Elapsed time
   fdM2C3(index,2) = fullyDeployedData(ii,2); % Multiplexer
   fdM2C3(index,3) = fullyDeployedData(ii,3); % Channel
   fdM2C3(index,4) = fullyDeployedData(ii,4); % Voltage

   index = index + 1;
end

% Solar Array 1
figure("Name","Voltage Generated by Solar Array 1")
% Solar Panel 7
plot(movmean(sM2C2(:,1),3),movmean(sM2C2(:,4),3)); hold on, grid on, grid minor
plot(movmean(pdM2C2(:,1),3),movmean(pdM2C2(:,4),3))
plot(movmean(fdM1C4(:,1),3),movmean(fdM1C4(:,4),3))
% Solar Panel 8
plot(movmean(sM2C0(:,1),3),movmean(sM2C0(:,4),3))
plot(movmean(pdM2C0(:,1),3),movmean(pdM2C0(:,4),3))
plot(movmean(fdM2C2(:,1),3),movmean(fdM2C2(:,4),3))
title("Voltage Generated by Solar Array 1")
xlabel("Time (s)")
ylabel("Voltage (V)")
legend("Stowed Panel (SP7)","Deployed Panel (SP7)","Fully Deployed Panel (SP7)" ...
   ,"Stowed Panel (SP8)","Deployed Panel (SP8)","Fully Deployed Panel (SP8)", ...
   "Location","southoutside","Orientation","horizontal")



% Solar Array 2
figure("Name","Voltage Generated by Solar Array 2")
% Solar Panel 1
plot(movmean(sM2C1(:,1),3),movmean(sM2C1(:,4),3)); hold on, grid on, grid minor
plot(movmean(pdM2C1(:,1),3),movmean(pdM2C1(:,4),3))
plot(movmean(fdM1C0(:,1),3),movmean(fdM1C0(:,4),3))
% Solar Panel 2
```

```
plot(movmean(sM1C4(:,1),3),movmean(sM1C4(:,4),3))
plot(movmean(pdM1C4(:,1),3),movmean(pdM1C4(:,4),3))
plot(movmean(fdM2C0(:,1),3),movmean(fdM2C0(:,4),3))
title("Voltage Generated by Solar Array 2")
xlabel("Time (s)")
ylabel("Voltage (V)")
legend("Stowed Panel (SP1)","Deployed Panel (SP1)","Fully Deployed Panel (SP1)" ...
   ,"Stowed Panel (SP2)","Deployed Panel (SP2)","Fully Deployed Panel (SP2)", ...
   "Location","southoutside","Orientation","horizontal")

% Solar Array 3
figure("Name","Voltage Generated by Solar Array 3")
% Solar Panel 3
plot(movmean(sM1C3(:,1),3),movmean(sM1C3(:,4),3)); hold on, grid on, grid minor
plot(movmean(pdM1C3(:,1),3),movmean(pdM1C3(:,4),3))
plot(movmean(fdM1C3(:,1),3),movmean(fdM1C3(:,4),3))
% Solar Panel 4
plot(movmean(sM1C2(:,1),3),movmean(sM1C2(:,4),3))
plot(movmean(pdM1C2(:,1),3),movmean(pdM1C2(:,4),3))
plot(movmean(fdM1C2(:,1),3),movmean(fdM1C2(:,4),3))
title("Voltage Generated by Solar Array 3")
xlabel("Time (s)")
ylabel("Voltage (V)")
legend("Stowed Panel (SP3)","Deployed Panel (SP3)","Fully Deployed Panel (SP3)" ...
   ,"Stowed Panel (SP4)","Deployed Panel (SP4)","Fully Deployed Panel (SP4)", ...
   "Location","southoutside","Orientation","horizontal")

% Solar Array 4
figure("Name","Voltage Generated by Solar Array 4")
% Solar Panel 5
plot(movmean(sM1C0(:,1),3),movmean(sM1C0(:,4),3)); hold on, grid on, grid minor
plot(movmean(pdM1C0(:,1),3),movmean(pdM1C0(:,4),3))
plot(movmean(fdM1C1(:,1),3),movmean(fdM1C1(:,4),3))
% Solar Panel 6
plot(movmean(sM1C1(:,1),3),movmean(sM1C1(:,4),3))
plot(movmean(pdM1C1(:,1),3),movmean(pdM1C1(:,4),3))
plot(movmean(fdM2C1(:,1),3),movmean(fdM2C1(:,4),3))
title("Voltage Generated by Solar Array 4")
xlabel("Time (s)")
ylabel("Voltage (V)")
legend("Stowed Panel (SP5)","Deployed Panel (SP5)","Fully Deployed Panel (SP5)" ...
   ,"Stowed Panel (SP6)","Deployed Panel (SP6)","Fully Deployed Panel (SP6)", ...
   "Location","southoutside","Orientation","horizontal")

figure("Name","Voltage Generated by Solar Array 5")
% Solar Panel 5
```

```
plot(movmean(sM2C3(:,1),5),movmean(sM2C3(:,4),5)); hold on, grid on, grid minor
plot(movmean(fdM2C3(:,1),5),movmean(fdM2C3(:,4),5));
title("Voltage Generated by Solar Array 5")
xlabel("Time (s)")
ylabel("Voltage (V)")
legend("Stowed Panel (SP9)","Fully Deployed Panel (SP9)", ...
    "Location","southoutside","Orientation","horizontal")

figure("Name","Voltage Generated by Solar Array 5")
plot(movmean(sM2C3(:,1),5),movmean(sM2C3(:,4),5)); hold on, grid minor, grid on
title("Voltage Generated by Solar Array 5")
xlabel("Time (s)")
ylabel("Voltage (V)")
```

Table H.7. TEMT6000 component test code.

```
/*
  Christopher Hernandez
  Development of a Solar Array Deployment Mechanism for a CubeSat
  TEMT6000 Component Test
 April 2025
  This program measures the ADC values of the TEMT6000 based on the ambient light level.
*/

#define lightSensor1 A7
void setup() {
  pinMode(lightSensor1, INPUT);
  Serial.begin(9600);
}

void loop() {
  int lightValue = analogRead(lightSensor);
  Serial.println(lightSensor);
  delay(1000);
}
```

Table H.8. TEMT6000 LSA and CD74HC4067 analog multiplexer integration test code.

```
/*
    Christopher Hernandez
    Development of a Solar Array Deployment Mechanism for a CubeSat
    TEMT6000 Integration Test
    This program tests the TEMT6000 LSA and CD74HC4067 analog multiplexer.
*/

  const int S0 = A0;
  const int S1 = A1;
  const int S2 = A2;
  const int SIG = A7;

void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);

  digitalWrite(S0,0);
  digitalWrite(S1,0);
  digitalWrite(S2,0);


  Serial.begin(9600);
  while(!Serial);
}

void loop() {
  // Reads the first light sensor
  digitalWrite(S0,0);
  digitalWrite(S1,0);
  digitalWrite(S2,0);
  delayMicroseconds(10);
  float lightSensor1 = analogRead(SIG);
  Serial.print("Light sensor 1: ");
  Serial.println(lightSensor1);

  // Reads the second light sensor
  digitalWrite(S0,1);
  digitalWrite(S1,0);
  digitalWrite(S2,0);
  delayMicroseconds(10);
  float lightSensor2 = analogRead(SIG);
  Serial.print("Light sensor 2: ");
  Serial.println(lightSensor2);

  // Reads the third light sensor
  digitalWrite(S0,0);
  digitalWrite(S1,1);
  digitalWrite(S2,0);
  delayMicroseconds(10);
  float lightSensor3 = analogRead(SIG);
  Serial.print("Light sensor 3: ");
  Serial.println(lightSensor3);
  // Reads the fourth light sensor
  digitalWrite(S0,1);
  digitalWrite(S1,1);
  digitalWrite(S2,0);
  delayMicroseconds(10);
  float lightSensor4 = analogRead(SIG);
  Serial.print("Light sensor 4: ");
  Serial.println(lightSensor4);

  // Reads the fifth light sensor
  digitalWrite(S0,0);
  digitalWrite(S1,0);
  digitalWrite(S2,1);
  delayMicroseconds(10);
  float lightSensor5 = analogRead(SIG);
  Serial.print("Light sensor 5: ");
  Serial.println(lightSensor5);
  Serial.println();
}
```

Table H.9. INA219 and PCA9548A integration test code.

```
/*
    Christopher Hernandez
    Development of a Solar Array Deployment Mechanism for a CubeSat
    INA219 and PCA9548A Integration Test
    April 2025
    This program tests the INA219 current sensors and the PCA9548A I2C multiplexer. The
lowest predefined calibration range of the INA219
*/

#include <Wire.h>
#include <Adafruit_INA219.h>

// I2C addresses
#define PCA9548A_1 0x70
#define PCA9548A_2 0x71

Adafruit_INA219 ina219s[2][5];

// Channel selection
void selectMuxChannel(uint8_t multiplexerChannel, uint8_t multiplexerAddress) {
  if (multiplexerChannel > 7) return;
  Wire.beginTransmission(multiplexerAddress);
  Wire.write(1 << multiplexerChannel);
  Wire.endTransmission();
```

```
}

// Checks the connection of the multiplexer and the initialization of the INA219 sensors
void setup() {
  Wire.begin();
  Serial.begin(9600);
  while (!Serial);
  for (int mux = 0; mux < 2; mux++) { // Multiplexer cycling
    uint8_t muxAddr = (mux == 0) ? PCA9548A_1 : PCA9548A_2;
    for (int ch = 0; ch < 5; ch++) { // Channel cycling
      if (mux == 1 && ch == 4) continue; // Skips unused channel
      selectMuxChannel(ch, muxAddr);
      if (!ina219s[mux][ch].begin()) { // Checks if the INA219 was initialized
        Serial.print("INA219 not found on multiplexer ");
        Serial.print(mux + 1);
        Serial.print(" channel ");
        Serial.println(ch);
      } else {
        ina219s[mux][ch].setCalibration_16V_400mA(); // If INA219 is initialized,  the lowest
predefined calibration range is used
      }
    }
  }
}

// The MCU cycles through each channel of a multiplexer before cycling to the next
multiplexer
void loop() {
  for (int mux = 0; mux < 2; mux++) { // Multiplexer cycling
    uint8_t muxAddr = (mux == 0) ? PCA9548A_1 : PCA9548A_2;
    for (int ch = 0; ch < 5; ch++) { // Channel cycling
      if (mux == 1 && ch == 4) continue; // Skips unused channel
      selectMuxChannel(ch, muxAddr);
      unsigned long timeNow = millis();
      float elapsedSeconds = timeNow / 1000.0;
      float voltage = ina219s[mux][ch].getBusVoltage_V();
      float current = ina219s[mux][ch].getCurrent_mA();
      float power = ina219s[mux][ch].getPower_mW();

      // CSV format: Elapsed time, multiplexer, channel, voltage, current, and power
      Serial.print(elapsedSeconds, 2); Serial.print(",");
      Serial.print(mux + 1); Serial.print(",");
      Serial.print(ch); Serial.print(",");
      Serial.print(voltage, 3); Serial.print(",");
      Serial.print(current, 3); Serial.print(",");
      Serial.println(power, 3);
      delay(300);
    }
  }
  delay(1000);
}
```

Table H.10. NEMA 11 and DRV8825 integration test code.

```
/*
    Christopher Hernandez
    Development of a Solar Array Deployment Mechanism for a CubeSat
    NEMA 11 and DRV8825 Integration Test
    This program tests whether the NEMA 11 motor could rotate CW and CCW.
*/

const int stepPin = 10;
const int dirPin = 9;
const int stepsPerRevolution = 200;

/*
  Current draw test
           Step-dir pins
  Motor 4: 8-7, 0.275 amps
  Motor 3: 10-9, 0.340 amps
  Motor 2: 6-5, 0.330 amps
  Motor 1: 4-3, 0.355 amps
*/

void setup() {
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
}

void loop() {
  digitalWrite(dirPin, HIGH);
  for(int x=0; x < 4 * stepsPerRevolution; x++){
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
  delay(1000);

  digitalWrite(dirPin, LOW);
  for(int x=0; x < 4 * stepsPerRevolution; x++){
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
  delay(1000);
}
```