Solar Panel Cleaner

ECE 445 Final Report - Spring 2025

Team # 26

Cameron Little (clittle5), Thomas Cai (wcai10), Geoffrey Swisher (swisher5)

Professor: Arne Fliflet

TA: Maanas Agrawal

May 7th, 2025

Abstract

As global demand for renewable energy rises, solar power has become a cornerstone of sustainable energy solutions. However, the efficiency of solar panels is often compromised by dust, debris, and environmental contaminants. Regular and effective cleaning is essential to maintain optimal energy output. To address this need, we developed a cost-effective, rail-based solar panel cleaner featuring a rotating brush, cleaning liquid dispenser, microfiber cloth, and supporting components. The system offers reliable cleaning performance and user-friendly operation through accessible controls and both manual and autonomous modes.

Contents

1 Introduction

Solar panels are constantly exposed to environmental elements like dust, pollen, and even bird droppings, all of which can accumulate on the surface and block sunlight. Even partial shading or debris on the surface can reduce the panel's output power, especially if there is partial shading across individual cells due to a solar panel's properties. Manual cleaning of panels is labor-intensive and impractical for large arrays. In addition, improper cleaning can damage the delicate anti-reflective surface of solar panels, causing long-term performance degradation.

Our goal is to create an affordable, reliable, consistent, and adaptable solar panel cleaner to improve energy efficiency among solar panels on the roof of the ECEB. For demonstration purposes and proof of concept, we will demonstrate on a solar panel borrowed from the senior design lab from a previous year's project. This panel will be used due to space constraints in the lab as it is quite smaller, at 40.1 x 26.4 inches^2 in area. The panels on the roof are much larger, at 64.5 x 39.5 inches^2. With the help of the machine shop, we are designing a rail-based solar panel cleaner that have two degrees of motion: vertically and horizontally. The controller with the PCB housing and wiring connections will sit stationary at the top of the housing. The cleaner unit will be attached to one rail unit spanning horizontally, free to move both left and right. This rail will be attached to two vertically spanning rails that move simultaneously up and down. The cleaner unit will be connected to a stepper motor, pulley system, and belt to guide it along the rails. The cleaner unit has an interchangeable microfiber cloth, rotating brush, and cleaning solution dispenser.

2 Design

2.1 Overview

Our project is composed of four subsystems: Power, Drivetrain, Controller, and Cleaning Mechanisms. Each subsystem was chosen to simplify the problem and separate the amount of work evenly.

We also base the functionality of the design on a set of high level requirements we established at the beginning of the design process. The requirements are as such:

- The user can switch between manual and autonomous modes for the cleaning unit to run on without needing to restart the cycle.
- The cleaner must complete a full vertical sweep (top to bottom and back to top) in under 10 minutes.
- Have the cleaning system implementation work for its entirety without damaging the solar panel surface; the sprayer dispenses solution on the panel, the brush is able to spin on the cleaner unit, and the microfiber cloth picks up leftover debris.

The organization and functionality of the four subsystems has remained consistent throughout the entire semester. Minor changes in parts selection and implementation were done to ensure the high-level requirements were properly met.

2.2 Design Procedure

The design process prioritized functionality and completion of the high-level requirements while also being cost-effective. No major compromises were made in design, but given a larger budget, a better solution could have been developed. Any concerns were considered for each subsystem, and the impact of these concerns is described in each subsystem's associated section.

The overall subsystem breakdown and how they interact with one another is seen below in Figure 1.

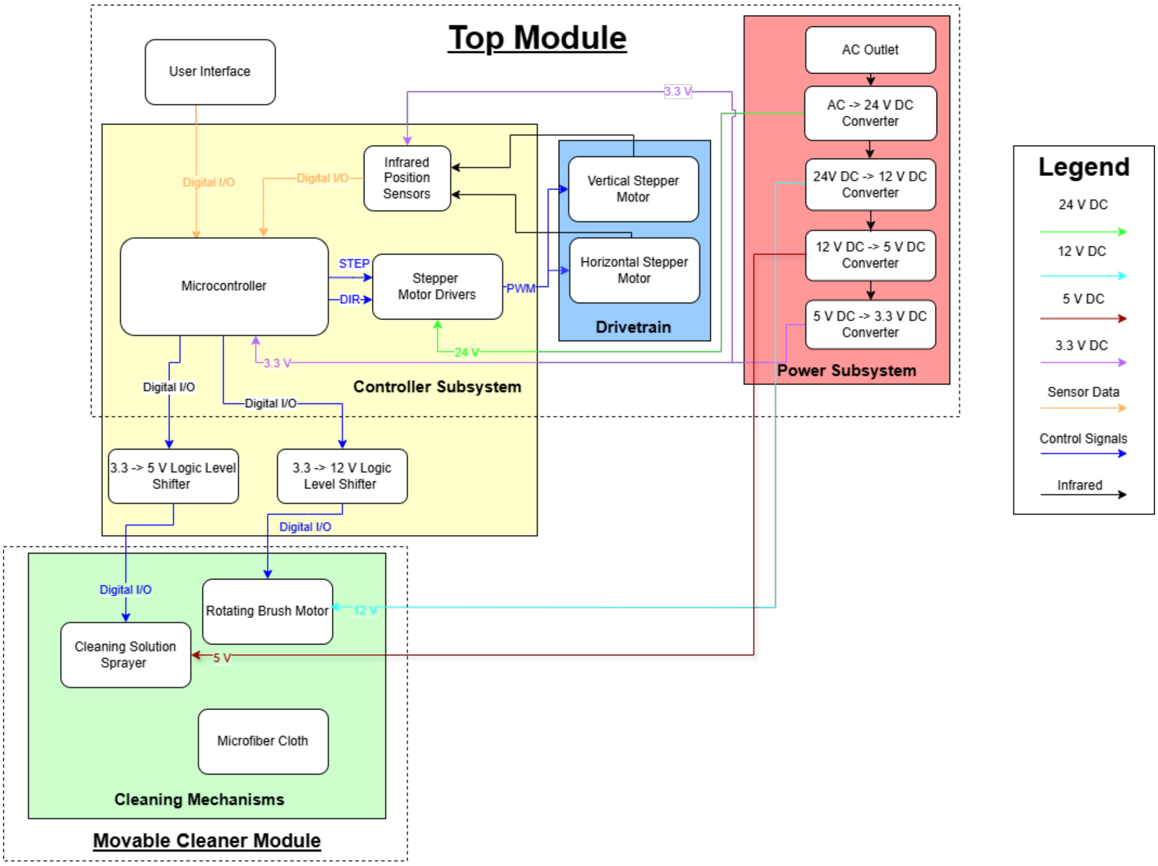


Figure 1: Block diagram of solar panel cleaner

The block Diagram from Figure 1 shows the Power, Drivetrain, Controller, and Cleaning Mechanism subsystems. The Power subsystem supplies reliable power to all the components of our cleaner. The Drivetrain subsystem moves our cleaning module vertically and horizontally across the solar panel. The Controller subsystem controls the Drivetrain and the Cleaning Mechanism subsystems, and is also responsible for autonomous cleaning given sensor inputs. The Cleaning Mechanism subsystem cleans the solar panel using the brush roller, with the sprayer dispensing cleaning solution, and a microfiber cloth picking up remaining debris. Next, we will dive into a deeper overview of these subsystems.

2.3 Subsystem Design

2.3.1 Power Subsystem

The power subsystem is responsible for supplying power to our cleaner. It draws power from a standard 120 V AC outlet and rectifies and converts its voltage into regulated DC power at multiple voltage levels: 24 V, 12 V, 5 V, and 3.3 V. These DC buses are then connected to other subsystems to power stepper motor drivers, DC brush motor, cleaning solution dispenser, infrared sensors and microcontroller. We decided on these voltage levels based on the operating voltage of the components we acquired.

The 120 V to 24 V converter used is the ALITOVE 24V 6A Power Supply Adapter Converter. The 24 V bus is used for stepper motor drivers, and it is further stepped down to 12 V using the N7812-2CH switching regulator from Mean Well. The 12 V bus is used to drive the brush motor, and it is stepped down to 5 V using the LC78_05-3.0 switching regulator from GAPTEC. The 5 V bus is used for the cleaning solution dispenser, and it is further stepped down to 3.3 V using the MIC5317-3.3YM5-TR low dropout regulator to supply power to the infrared sensors and the microcontroller. In addition, fuses, RKEF400, are implemented for overcurrent protection to protect the user, the motors, the solar panel, and the digital components. Figure 2 shows the schematic of the power subsystem of the PCB, as well as part of the cleaning mechanism subsystem, which is discussed later.

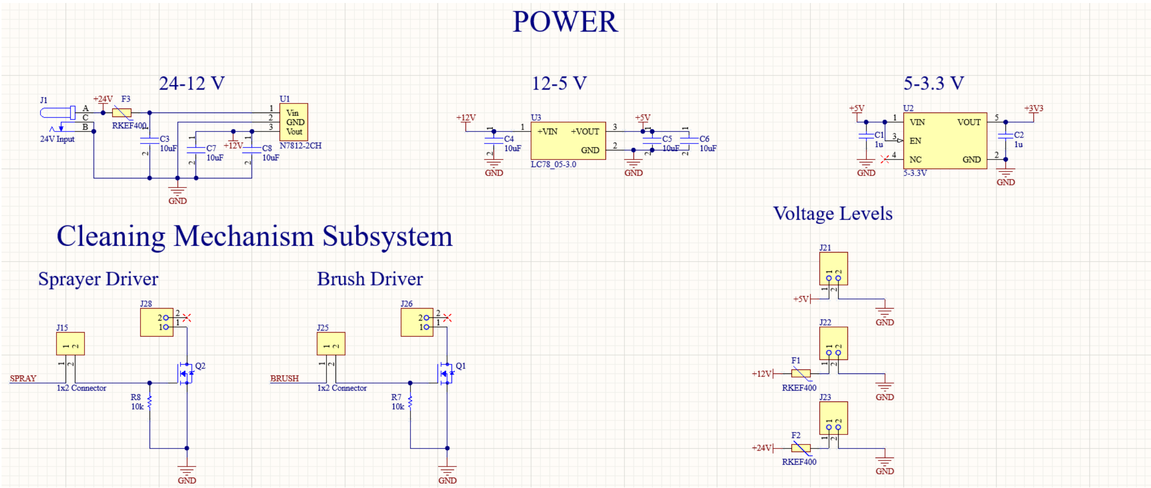


Figure 2: Schematic of power and cleaning mechanism subsystems on PCB

2.3.2 Drivetrain Subsystem

The drivetrain subsystem is mainly mechanical and is responsible for moving the cleaning module on the solar panel. The drivetrain subsystem consists of two vertical rails, a horizontal rail, and complementary motors, gears, pulleys, and timing belts, providing precise positioning along the length and width of the panel. Infrared sensors are also implemented, which is used to sense the position of the cleaner for the autonomous mode.

The two vertical rails are fixed on the press board platform, one on each side of the solar panel. The horizontal rail is attached to the two vertical rails, with its two attachments driven by two stepper motors, one on each rail, through a timing belt. The stepper motors on the vertical rails drive in synchrony and move the horizontal rail vertically. The cleaning module is attached onto the horizontal rail and can be driven horizontally by an additional stepper motor using a timing belt. We used three NEMA 23 stepper motors, with two that have the same speed running in synchrony to drive the cleaning module vertically and one to drive the cleaning module horizontally.

For infrared sensors, we used IR break beam sensors from Adafruit. We have two pairs on each side across the length of the solar panel, and two pairs on each side across the width of the solar panel. The length-oriented sensors are placed to aim for the sprayer, and the width-oriented sensors are placed to aim for a fin attached to the horizontal rail. When the IR beam of the length-oriented sensors are broken, it would indicate our cleaning module has reached the horizontal edge of the solar panel. Similarly, a break in width-oriented sensors would indicate a position on the vertical edge of the solar panel.

With these sensor outputs, we could set the states of our autonomous mode. Figure 3 shows the implementation of our project, containing the three rails and their corresponding stepper motors for the Drivetrain subsystem, and the IR sensors being taped on wood blocks around the solar panel.

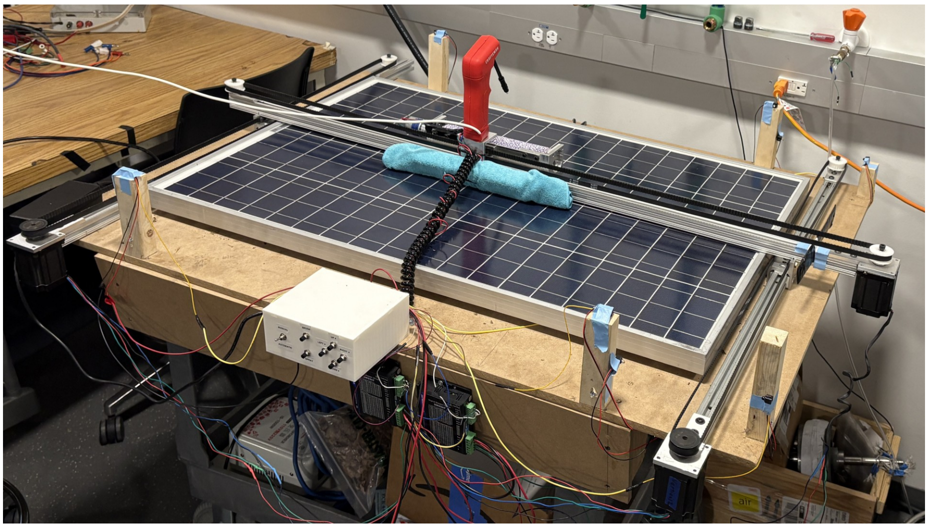


Figure 3: Drivetrain subsystem implementation on solar panel cleaner

2.3.3 Controller Subsystem

The controller subsystem consists primarily of a STM32 microcontroller, with its corresponding inputs and outputs. It also includes the user interface, which is used to control the movement and operation of the drivetrain and cleaning mechanisms, and to switch between manual and autonomous modes.

Looking at Figure 4, our user interface implementation on our custom 3D printed case is seen. The six buttons: up, down, left, right, brush, and spray control the corresponding movements of the cleaning module through the drivetrain subsystem and the operation of the cleaning module in manual mode. The toggle switch can switch between manual and automatic, which will set the operating state of the controller subsystem. For manual operation, the microcontroller will change its output signals to drive the stepper motors on the vertical rails correspondingly through PWM when an activation is detected with the up or down buttons. Similarly, when it senses an activation in the left or right button signal, the microcontroller will change its output signals to drive the stepper motor on the horizontal rail correspondingly to move the cleaning module across the panel.

Figure 4: User interface implementation on 3D printed case

For the autonomous mode, these output switches are performed automatically due to responses from the IR sensors detecting the panel edges. In order to achieve a cycle that would clean the whole panel and perform different actions based on the cleaner's position, a finite state machine was implemented in the microcontroller code. In state 0, the cleaner will reset itself to the bottom left corner, which is achieved by moving left to the edge of the panel, then moving down to the bottom of the panel. Next, the cleaner will spray for a time designated in the code, which is approximately 2.3 seconds by default, in state 6. The cleaner then moves to state 1, where the brush is spinning and the cleaner is moving up to the top of the panel. State 2 is next, where the cleaner moves right for approximately 2 seconds, then moves into state 3, where it moves back down the panel with the brush activated. Then, it will repeat the process from state 6. If the right edge of the panel is detected in state 2, the cleaner enters state 4, which performs the sequence of states 4, 7, and 5 (an edge case version of 3, 6, and 1), before entering state 0 again. These states are summarized in Table 1. The code used is in Appendix B.

State	Description	BRUSH	SPRAY
0	Reset to bottom left	LOW	LOW
1	Move up cleaning	HIGH	LOW
2	Move right 2s	LOW	LOW
3	Move down cleaning	HIGH	LOW
4	Move down cleaning (edge)	HIGH	LOW
5	Move up cleaning (edge)	HIGH	LOW
6	Spray for 2.3s	LOW	HIGH
7	Spray for 2.3s (edge)	LOW	HIGH

Table 1: Autonomous Mode States

2.3.4 Cleaning Mechanism Subsystem

The cleaning mechanism subsystem is responsible for physically removing dirt and debris from the solar panel surface. It consists of a rotating brush, driven by a DC motor, a cleaning solution gallon container and its associated sprayer, and a microfiber cloth. The cleaning solution sprayer would spray the solution autonomously or manually to moisten the resilient stains, while the rotating brush rotates and brushes them off. The microfiber cloth will do the final round of cleaning, wiping off the remaining residue. All of these cleaning mechanisms can be seen in Figure 5.

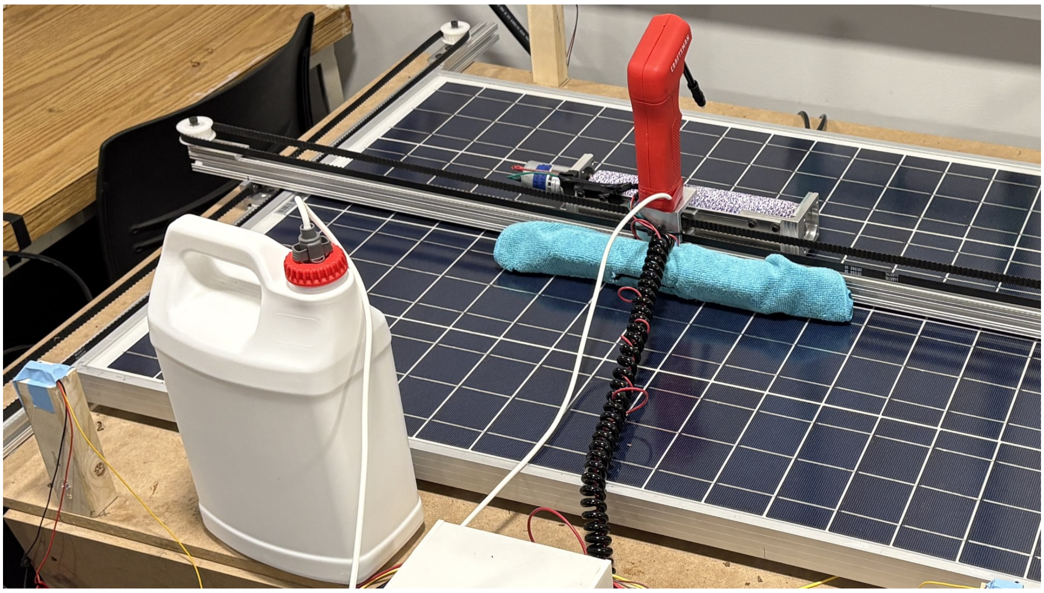


Figure 5: Cleaning Mechanism subsystem implementation

The brush motor is driven by 12 V, which is controlled by the 3.3 V BRUSH signal from the controller subsystem through a voltage logic shifter using a FQP30N06L MOSFET. The sprayer pump is driven by a similar fashion, but with the 5 V power bus. The voltage logic shifter circuit schematic is shown in Figure 6. The circuit is explained later in Section 3.2

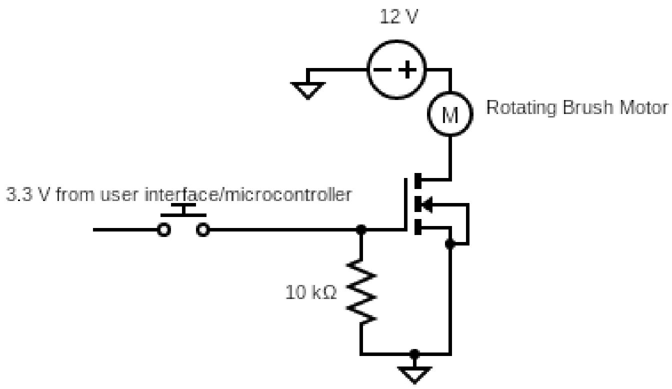


Figure 6: Logic level shifter circuit schematic

The sprayer we have is the Craftsman battery-operated weed sprayer, which functions on a physical trigger and a 4.5 V battery pack. In order to switch from mechanical activation to electrical activation, we had to disassemble the sprayer and connect our logic shifter directly to the pump.

We are able to activate the sprayer with control signal SPRAY using this setup. The inside of the sprayer is seen later in Figure 11. In addition to activating the sprayer, we also had to find a way to transfer power from our PCB to the cleaning module. We decided to use a coiled cable to achieve this. Figure 7 shows this coiled cable used.

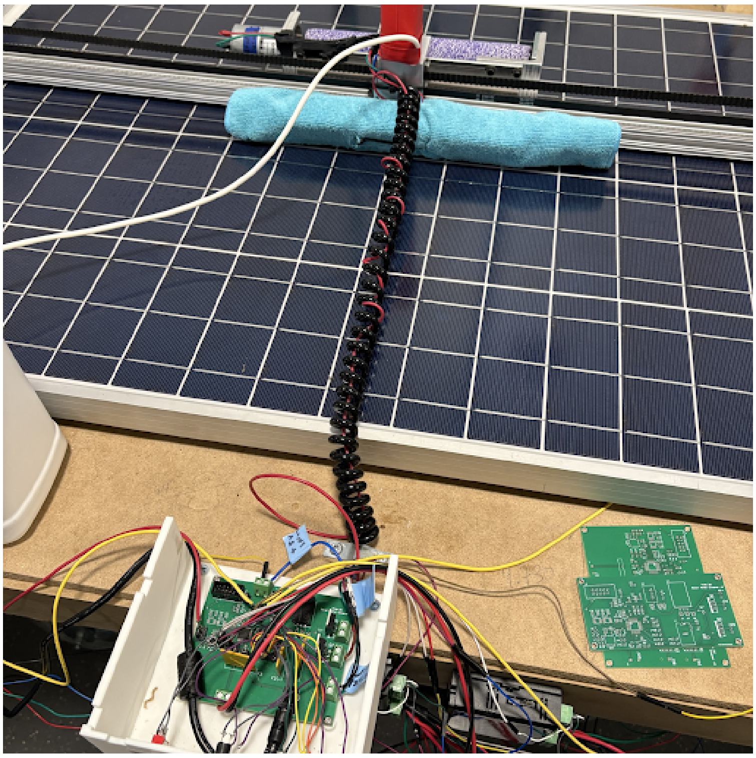


Figure 7: Coiled cable used to transfer power from PCB to moving cleaner module

2.4 Physical Design

The solar panel cleaner design, as seen in Figure 8, has the cleaning solution dispenser attached to the rotating brush head piece. The PCB is mounted to a stationary 3D printed case at the top of the panel, with wires lengthened and guided underneath the panel to prevent them from getting tangled with the moving bar and brush. This is to allow the spray to reach the panel section currently being cleaned by the cleaner unit. The user can manually press the button associated with the sprayer to dispense the cleaning solution. There are also buttons to turn on the rotating brush, as well as to move the panel in four directions. In addition, there is an autonomous mode which cleans the panel using infrared position sensors. There are also three stepper motors and their associated pulleys and timing belts. A coiled cable is connected from the controller PCB to the DC motor of the brush spinner, as well as the solution sprayer, to provide power for the brush to spin and clean the panel.

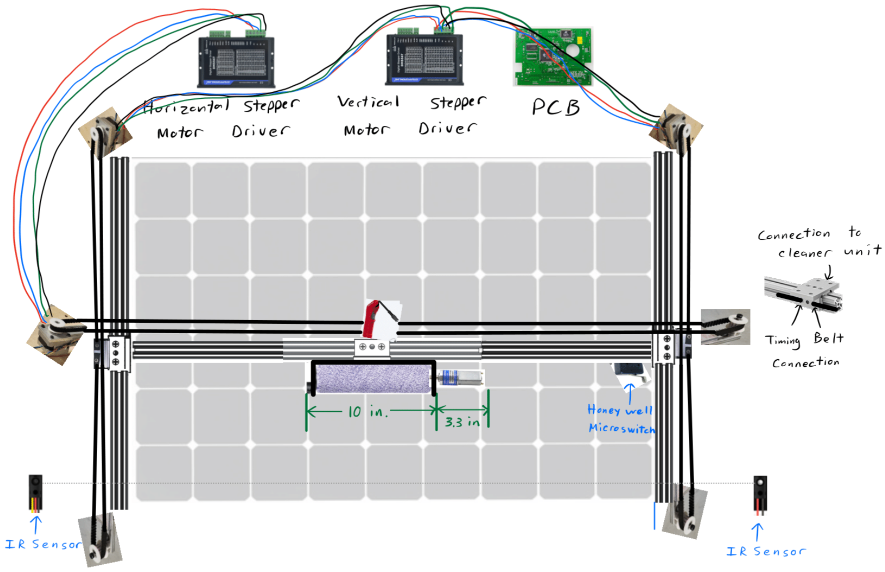


Figure 8: Solar Panel Final Design

The finished product after everything is done is seen below in Figure 9.

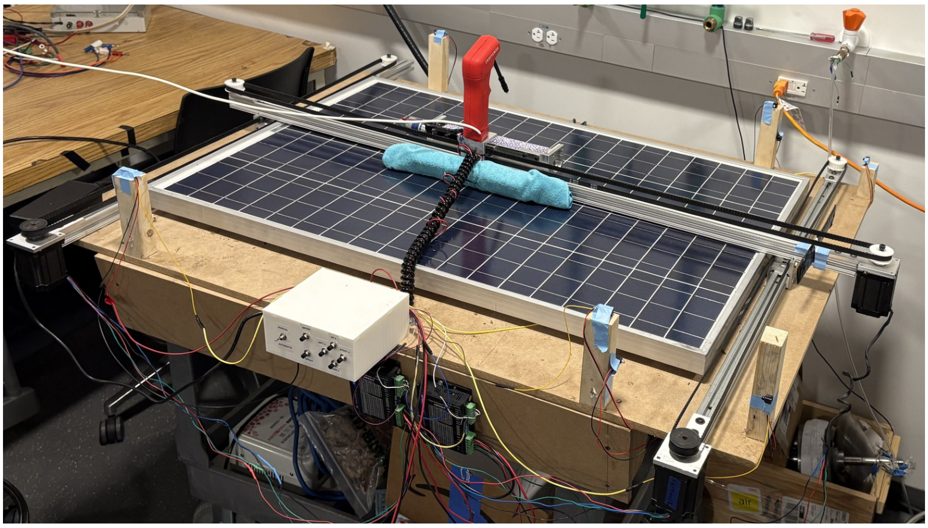


Figure 9: Solar Panel Finished Implementation

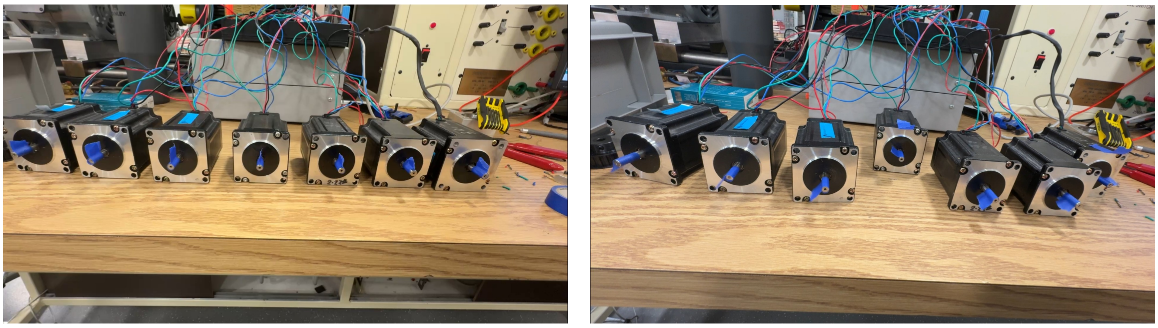
3 Verification

All the subsystem requirements and verifications, R&V, were met. For a complete list of R&V tables, please refer to the Appendix A. Below are some essential verification processes.

3.1 Stepper Motor Testing

One of the key challenges we encountered with the drivetrain subsystem is ensuring the synchronization of two stepper motors to drive the horizontal rail up and down. If the motors operate at slightly different speeds, the horizontal rail can become tilted, potentially causing it to jam and resulting in damage to both the project and the solar panel. To prevent this issue, we tested seven NEMA 23 stepper motors.

To see this more easily, we put a piece of tape on the shaft initially in the same upward position (12 o'clock on an analog clock). This is seen in Figure 10(a). We would then turn on the supply voltage for about 30 seconds. If the motors operated at the same speed, the tape remained aligned when power was turned off. Misalignment of the tape indicated speed differences. The end of the test is seen in Figure 10(b). This shows that the fourth and seventh stepper motors tape is not aligned, meaning they have a different speed compared to one, two, three, five, and six. This experiment was repeated three times with varying run times to confirm alignment. Two of those five same speed motors were then chosen to be implemented as vertical drivers for our drivetrain subsystem.



((a)) Initial shaft alignment at start of test ((b)) End shaft alignment at end of test

Figure 10: Stepper motor speed test

3.2 DC Brush Motor Testing

One issue we had with the 12 V DC brush motor was that it runs on 12 V voltage level, but the control signal we had was only 3.3 V. To overcome this issue, we decided to use a voltage logic shifter, where a MOSFET is used as the low-side switch to short the negative terminal of the motor to ground. The positive terminal of the motor is directly connected to our 12 V bus. The gate of the MOSFET is controlled by the signal coming from the controller unit, so when the signal is high, the MOSFET will turn on and short the negative terminal of the motor to ground. This will apply 12 V across the motor terminals and thus turning it on. Figure 6 shown earlier is the circuit schematic of this implementation. The sprayer is controlled in a similar fashion. To connect the motor to our PCB, we used one of the three wires inside the coiled cable to connect the negative terminal and the drain of the logic shifter, and we ran a separate wire along the coiled cable to connect the positive terminal to our 12 V bus.

3.3 Sprayer Testing

In order for the sprayer to function, we needed to reconfigure the spray to operate based on our electrical control signal. Initially, the spray operates based on a 4.5 V battery pack and dispenses based on a physical trigger and mechanical switch. After disassembling the case, we were able to operate the pump motor by supplying a 5 V across the pump terminals. In addition, we found that there also exists a spring solenoid valve that needs to pressed for water to flow. The spring is taken out keeping the solenoid valve permanently open. After finalizing our implementation, we soldered wires directly to the terminals of the pump. The wires are then soldered to the coiled cable to achieve power transfer between the sprayer and the voltage logic shifter on our PCB. Figure 11 below shows the inside of the sprayer, with the spring removed.

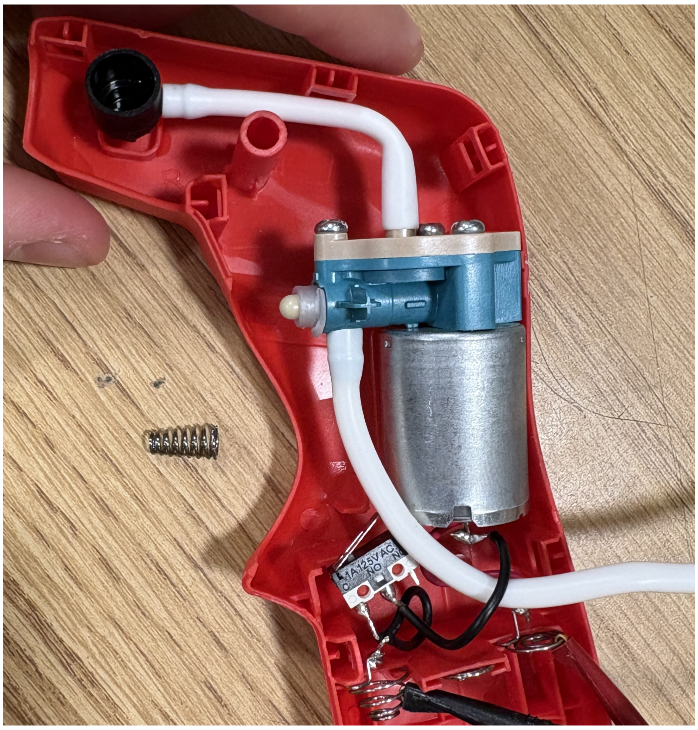


Figure 11: Inner structure of the sprayer

3.4 IR Sensor Testing

In order to be used within the proposed design for autonomous mode, the IR sensors needed to undergo testing for their usability. From the datasheet, the sensor is able to be operated at 3.3 V and 5 V, with the beam length increasing as the voltage increase. We tested both 3.3 V and 5 V, and found the sensors could be placed 48 inches apart and still retain contact between transmitter and receiver, which is a distance greater than the width of the solar panel system. We settled with 3.3 V in our final design for less power draw.

The sensors were also tested while mounted on the solar panel system. In this case, the sensors were unable to keep contact across the width of the panel (48 inches). The true cause of this was unknown as they worked in the lab. It might have been due to interference from other objects or misalignment between sensors. This discovery prompted a redesign of the sensor positions to detect the cleaner at the top and bottom of the panel. Instead of positioning the vertical detection sensors across the entire solar panel, we decided to add a fin to our horizontal rail, and place the sensors correspondingly such that the fin could break the IR beams when the cleaning module is moved to the top or the bottom of the solar panel. Figure 12 below shows the fin attached to the horizontal rail and the vertical IR sensors being mounted right across the vertical rail.

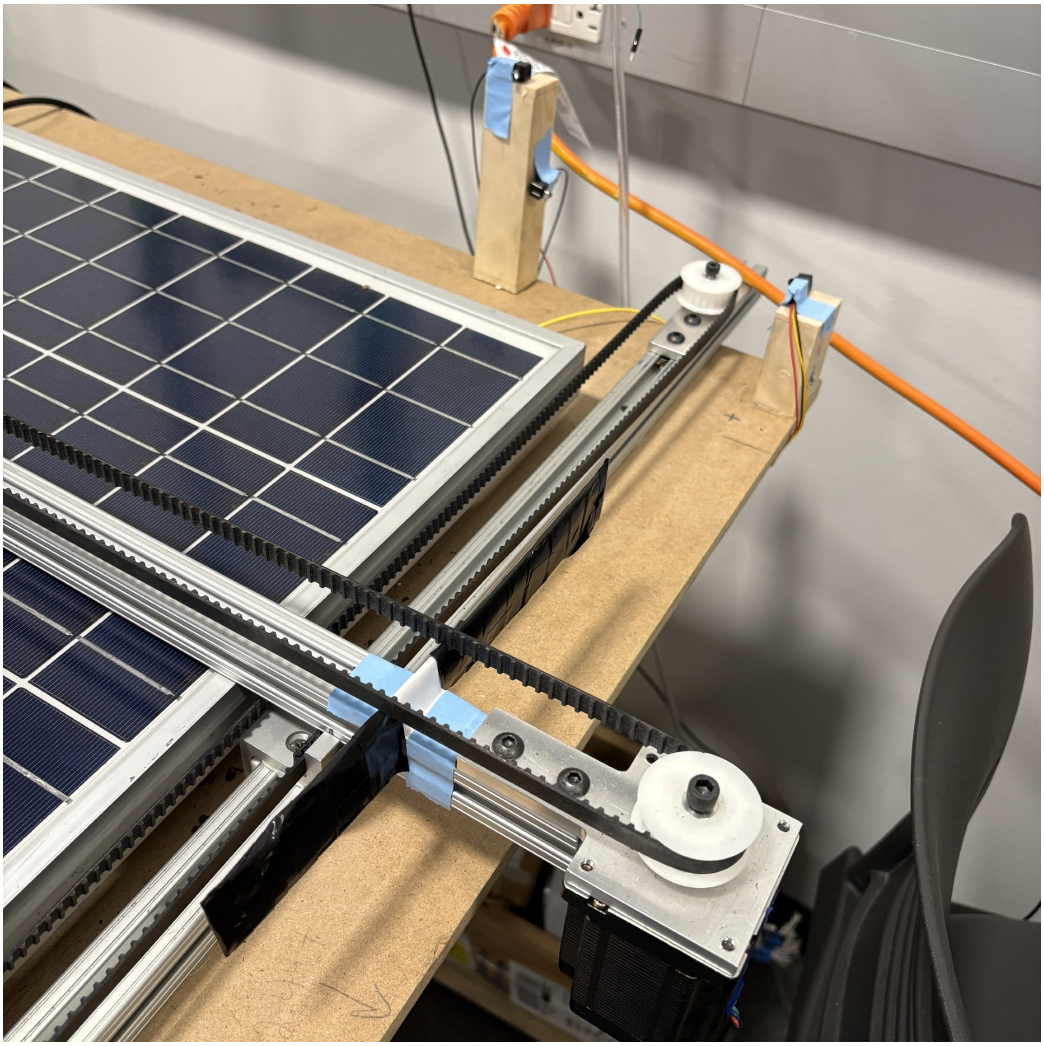


Figure 12: Vertical IR sensor and fin implementation

3.5 PCB Verification

The PCB design was verified in stages. First, the essential power conversion components were placed on the first round PCB. These components were then powered using the 120 V AC to 24 V DC converter. Then, the microcontroller was placed on the board and tested by attempting to program it. The device was successfully programmed, so the microcontroller and surrounding components were verified. Finally, in the third and fourth round PCBs, the logic level shifter circuits were tested on their own to ensure functionality before the addition of all the other components on the board. A final test was a full-load test, where all cleaner elements (brush, sprayer, and horizontal/vertical movement) were used and the voltage buses were measured. The resulting no-load and full-load voltages were recorded in Table 2 and verified with our tolerances.

Voltage Level	No-Load Voltage	Full-Load Voltage
3.3	3.270	2.261
5	5.032	4.972
12	12.061	11.894
24	24.147	23.982

Table 2: No-load and full-load voltage test

3.6 Combined System Verification

After verifying the operation of the individual components and subsystems, the entire system was wired together to be tested. First, the manual mode system was tested. The movement operation was observed, and the cleaner unit could successfully move up, down, left, and right without interruption. The cleaning mechanisms were then observed, and the brush spun and the sprayer dispensed liquid onto the panel. From this testing, it was observed that the cleaner movement speed should be slowed down from the initial testing value in order to clean the panel most effectively. Additionally, it was seen that the sprayer required a buffer time to build up pressure in the tube before liquid would dispense.

The autonomous mode was the last to be verified and configured. The sensors were aligned through a process of trial and error, and the fin was adjusted in order to perfectly place the limits of the brush movement. In testing, momentary glitches were observed in the sensor readings, which would change internal logic states in an undesired fashion. Therefore, a buffer time of approximately 100ms was added to remove the effect of momentary glitches. Finally, the time the sprayer was active was tuned such that a small amount of liquid would be dispensed each time the cleaner reached the bottom of the panel.

4 Costs

4.1 Cost Analysis

An critical factor in any project is the overall cost to the manufacturer to develop and produce the product. Below is a cost estimate of the prototype of the design of the solar panel cleaner, and the price of a company developing commercial-grade solar panel cleaners would cost.

4.1.1 Cost of Labor and Parts

According to the ECE department's statistics, the average starting salary for new graduates is $87,769 annually. To convert this into an hourly wage, it is approximately $42/hour. Therefore, the total cost of labor, including student labor and machine shop labor, can be calculated as follows:

Description	**Amount**
Hourly wage (students)	\$42/hour
Students	3
Labor hours per student per week	$2.5 \times 15 = 37.5$ hours/week
Weeks	11
Additional hours	20 hours
Student labor cost	\$42/hour $\times$ 37.5 hours/week $\times$ 11 weeks $\times$ 3 students + \$42/hour $\times$ 20 hours $=$ \$52,815
Machine shop estimate	60 hours $\times$ \$50/hour $=$ \$3,000
Total labor cost	\$52,815 $+$ \$3,000 $=$ \$55,815

4.1.2 Bill of Material

Our bill of material table is attached in Appendix C.

5 Conclusion

5.1 Summary and Future Work

We developed a cost-effective and user-friendly system for cleaning solar panels. The design successfully met all high-level requirements, and testing demonstrated a significant improvement in the solar panel's power output following cleaning. In addition, we completed the project within our $150 budget and gained valuable experience in Altium Designer, 3D printing, and STM microcontroller programming throughout the process.

During the implementation and testing phases, we encountered couple of unforeseen challenges. One issue involved inconsistent tube pressure in the sprayer, which led to unpredictable liquid dispensing. We resolved this by adjusting the reservoir height and adding control logic to fine-tune the dispensing duration. Additionally, we faced sensitivity problems with our IR sensors, which frequently produced false signals. Through repeated testing and repositioning, we mitigated these glitches by carefully realigning the sensors and introducing a buffer time, achieving reliable performance with minimal impact.

For future work, we aim to redesign the cleaning mechanism and drivetrain subsystem, as well as enhance the controller subsystem. While the current cleaning mechanism is effective against dust and leaves, it struggles with tougher debris such as accumulated pollen and bird droppings. To improve its performance, we plan to explore more effective cleaning solutions and modify the design to make the brush and microfiber cloth easily replaceable. Additionally, our drivetrain subsystem does not scale well for large solar panel farms, where building an extensive rail system would be impractical. A more flexible design that can be easily transferred across panels is essential. Furthermore, we intend to upgrade the controller subsystem by integrating additional sensors and potentially leveraging machine learning algorithms to detect dirty areas that require extra cleaning. Incorporating wireless controls will also be key to making the system more accessible and user-friendly.

5.2 Ethics

We recognize the importance of addressing potential ethical and safety issues related to our Solar Panel Cleaner. To ensure that the work performed is ethical and safe, we will closely follow the IEEE Code of Ethics.

5.2.1 Safety and Sustainability

Our top priority is the safety of our users, the solar panels, and the environment. Our goal is to design and assemble a cleaner that is safe to operate and gentle enough to avoid damaging the solar panels. We also plan to use environmentally friendly cleaning solutions to minimize harm to the environment.

5.2.2 Equality

We aim to design an inclusive and accessible product that can accommodate a wide range of users. We intend to create an easy-to-use cleaner at a low cost, enabling users from various backgrounds to benefit from it. The rail system of the cleaner is also scalable, making it adaptable to different setups.

5.2.3 Honesty and Transparency

Throughout the process, we will uphold the highest standards of integrity. We will present our product honestly and in accordance with the IEEE Code of Ethics.

5.3 Safety

5.3.1 Electrical Safety

On our PCB, we implemented some fuses to protect the solar panel and the circuits from over current, which can be caused by the rails getting stuck. Based on our measurements, we decided on using fuses rated for 4.5 A to balance headroom and sensitivity. In addition, we also put high quality electrical tape over all of our connections with exposed wires. This can protect our circuits from liquid and also prevent electrical shock.

5.3.2 Mechanical Safety

For mechanical safety, we deliberately set the cleaning speed to be very low. This can reduce the risk of causing physical harm to people and the solar panel. In addition, our brush motor makes a lot of noise when operating. This can serve as a warning for people to stay at a distance during cleaner's operation. However, these precautions cannot prevent all safety issues. We'd like to advise users to stay away from the cleaner when it is in operation.

6 References

References

[1] IEEE, "IEEE Code of Ethics," Jun. 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: May 7, 2025].

[2] *2A Non-Isolated DC-DC Switching Regulator*, datasheet, Mean Well, Inc., 2024. Available: https://www.meanwellaustralia.com.au/sites/default/files/2024-12/MEAN%20WELL%20N78-2%20Data%20Sheet.pdf.

[3] *LC78-3.0 series*, datasheet, Mean Well, Inc., 2024. Available: https://www.meanwellaustralia.com.au/sites/default/files/2024-12/MEAN%20WELL%20N78-2%20Data%20Sheet.pdf.

[4] *High Performance Single 150 mA LDO*, datasheet, Microchip Technology, Inc., 2019. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/MIC5317-High-Performance-Single-150mA-LDO-DS20006195B.pdf.

[5] *NTP30N06L, NTB30N06L*, datasheet, ON Semiconductor, Inc., 2006. Available: https://www.onsemi.com/pdf/datasheet/ntp30n06l-d.pdf.

[6] STM32 Example: LoRa Router, web page. Available at: https://courses.grainger.illinois.edu/ece445/wiki/#/stm32_example/index. [Accessed February 2025].

[7] Digi-Key Contributors, "Logic Level Shifting Basics," Digi-Key, Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/blog/logic-level-shifting-basics. [Accessed: May 7, 2025].

[8] Onefinity CNC Forum Contributors, "Two Stepper Motors on One Driver," Onefinity CNC Forum. [Online]. Available: https://forum.onefinitycnc.com/t/two-stepper-motors-on-one-driver/2140/9. [Accessed: May 7, 2025].

[9] OMC-StepperOnline, "DM556T Stepper Driver Manual," OMC-StepperOnline. [Online]. Available: https://www.omc-stepperonline.com/download/DM556T_V4.0.pdf. [Accessed: May 7, 2025].

[10] CRAFTSMAN, "1-Gallon Tank Sprayer," Lowe's. [Online]. Available: https://www.lowes.com/pd/CRAFTSMAN-1-Gallon-Tank-Sprayer/1001433038. [Accessed: May 7, 2025].

[11] Renogy, "RNG-100P Solar Panel Specification Sheet," Renogy. [Online]. Available: https://www.renogy.com/content/files/2015/06/RNG-100P-Specification.pdf. [Accessed: May 7, 2025].

[12] Adafruit, "IR Break Beam Sensor - 3mm LEDs Datasheet," Adafruit. [Online]. Available: https://cdn-shop.adafruit.com/product-files/2168/C2917.pdf. [Accessed: May 7, 2025].

A Appendix - Subsystem R&V Tables

A.1 Power Subsystem

Requirements	**Verifications**
AC wall voltage is converted to appropriate levels of DC voltage.	120VAC is converted to DC levels of 24V, 12V, 5V, and 3.3V. Each level has a tolerance of ±10% and can be measured using a digital multimeter.
The converters can handle full motor load without significant change in voltage.	The DC and stepper motors, as well as the internal motor and gear pump in the sprayer, are engaged at their designed speed and voltage. Each DC voltage level is monitored using an oscilloscope to ensure they remain within ±10% of their rated values.
The output of the converters falls to 0 when the input to the system fails.	The AC wall voltage is removed from the system at time $t = 0$. Each DC voltage level should drop to $< 10\%$ of its rated value within $t < 500ms$, measured using a DMM.
Motor power lines include overcurrent protection to prevent overloading of subsystems.	An intentional overcurrent event is triggered by a sudden load increase of 30%. A fuse blow or other shutdown is observed and recorded, and time to disconnect is measured with a stopwatch.
The power subsystem must operate reliably in ambient temperatures from 0°C to 45°C.	The entire system is placed outside on a warm, sunny day while running motors and sensors. Voltage rails are measured to confirm they remain within ±10% of nominal with the environmental stress

Table 3: Requirements and Verifications for the Power Subsystem

A.2 Drivetrain Subsystem

Requirements	Verifications
The drivetrain must support the cleaning module and the control module.	Perform static and dynamic load tests to simulate the operation of our cleaner and ensure the drivetrain is stable and remains operational while maintaining structural integrity. The cleaning module mount should also cover at least 80% of the solar panel, which can be verified by measuring the distances covered.
The speed of the two stepper motors controlling the vertical movement of our cleaning module must be similar. The mismatch should be within 1% between the two.	Use a tachometer to measure the actual speed of each motor under operation conditions and verify that the actual speed of the two motors are within 1% of each other.
The infrared sensor at different locations can be activated and output desirable signals to the microcontroller.	The infrared sensors can be powered at desired distance by our power subsystem using the +5V bus and can transfer the detection signal +5V back to the control module. The signal can be measured and verified by using an oscilloscope at operating distance away. Confirm motors stop or reverse within 100 ms when triggered by the IR sensor.
The drivetrain subsystem must be able to move the cleaning module across the entire solar panel without exceeding the ratings of all components.	When we are testing the drivetrain subsystem, we would first use a power meter to ensure the total power being drawn does not exceed our rectifier ratings. Then, we would use individual power meters to measure the power drawn by each stepper motor as they operate as desired. During the entire operation, all current being drawn should not exceed 95% of the corresponding maximum ratings. In addition, the motors and the belts must be able to drive the cleaning module horizontally and vertically across the panel.
The system must complete a full vertical sweep (top to bottom and back to top) in under 10 minutes.	A stopwatch can be used to time our system. The time it takes to complete a vertical sweep should be strictly less than 10 minutes.
Stepper motors must not exceed their rated temperature (e.g., 80°C case temperature) during operation.	After 30 minutes of continuous operation, measure the surface temperature of each motor using an infrared thermometer or thermocouple. Confirm measured temperature is below 80°C.

Table 4: Requirements and Verifications for the Drivetrain Subsystem

A.3 Controller Subsystem

Requirements	Verifications
The microcontroller is able to read from the position sensors.	A voltage of 4.3-5V on the signal wires (IN_IRX on the MCU) is detected as HIGH, and a voltage of 0-0.7V is detected as LOW.
The microcontroller sends a signal to the position stepper drivers to engage/disengage stepper motors in a given direction.	A voltage of 4.5-5.5V is detected at the input (STEP) of the positioning stepper motor driver when the ouput logic is HIGH to engage the motor, and a voltage of <1.0V is detected when the output logic is LOW. When direction needs to be switched, the direction input (DIR) of the stepper motor driver switches from 4.5-5.5V to <1.0V or vice versa.
The microcontroller sends a signal to the brush motor driver to engage/disengage the DC brush motor.	A voltage of 4.5-5.5V is detected at the input (STEP) of the brush DC motor driver when the ouput logic is HIGH, and a voltage of <1.0V is detected when the output logic is LOW.
The microcontroller sends a signal to the sprayer to turn on/off the spray.	A voltage of 4.5-5.5V is detected at the input of the sprayer control when the output logic is HIGH, and a voltage of <1.0V is detected when the output logic is LOW.
The manual control buttons create corresponding signals for each movement or action.	Each movement/action button on the user interface inputs either HIGH (∼3.3V) or LOW (∼0V) to the microcontroller. The microcontroller outputs HIGH or LOW in repsonse to these inputs on signals STEPX, DIR and SPRAY.
The manual and autonomous mode buttons switch the internal logic mode.	The manual and autonomous mode buttons on the user interface inputs either HIGH (∼3.3V) or LOW (∼0V) to the microcontroller. When the system is switched to autonomous mode, output signals will change from HIGH to LOW without user input. When manual mode is pressed, output signals will cease to switch state without user input.
The controller responds to all input changes within 100ms to avoid perceived lag.	Apply step changes to position sensors and manual buttons while measuring output pins using oscilloscope. Verify time between input change and output response is <100ms.
Microcontroller software must perform self-check on startup and detect invalid states.	Reset the microcontroller and monitor serial debug output or onboard LED status indicators to confirm self-check routine executes correctly. Intentionally apply an invalid state to verify detection.
All unused GPIO pins should be configured as inputs with internal pull-down resistors to avoid floating pins.	Read all unused pin states after power-up and confirm they remain LOW. Use multimeter if necessary to measure pin voltages.

Table 5: Requirements and Verifications for the Controller Subsystem

A.4 Cleaning Mechanism Subsystem

Requirements	Verifications
The sprayer and roller must operate when the control signals are HIGH and they must not operate when the signals are LOW.	Use an oscilloscope to ensure that when the control signals are LOW, the roller and sprayer will not draw power. In addition, use a power meter to assist in measuring the roller power while they are operating.
The cleaning brush and sprayer must engage/disengage within 200 ms of receiving control signals.	Monitor the motor and sprayer power lines using an oscilloscope while commanding on/off transitions. Confirm that actual activation/deactivation occurs within 200 ms of command signal changes.
The cleaning mechanism should clean up most of the debris and cover at least 80% of the solar panel's total surface area.	Prepare the solar panel by adding a thin layer of dust on its surface, then operate our cleaner to ensure most of the dust are cleaned off, and measure the remaining dust to ensure it covers less than 20% of the entire solar panel's surface area.
The cleaning brush must maintain consistent contact with the panel surface throughout operation.	Place the system on a mock panel surface with intentional unevenness (simulated dirt buildup). Observe brush contact visually to determine if there are any gaps between the brush and panel at multiple points.
The cleaning mechanism must not damage the solar panel.	An alternative glass or plastic surface should be used during initial trials. A thin, fragile layer could be added before operating the cleaning module to ensure it does not damage the solar panel surface.
The cleaning mechanism must operate reliably in ambient temperatures from 0°C to 45°C.	Place the system outside on a warm, sunny day. Operate the brush and sprayer continuously for 5 minutes. Monitor RPM, flow rate, and current draw to ensure stability.

Table 6: Requirements and Verifications for the Cleaning Mechanism Subsystem

B Appendix - STM32 Code

```
  /* USER CODE BEGIN 2 */

// defaults for the motor rotation directions
GPIO_PinState up = 1;
GPIO_PinState down = 0;
GPIO_PinState left = 1;
GPIO_PinState right = 0;

// IR pin definition
uint16_t ir_top = IN_IR1_Pin;
uint16_t ir_bottom = IN_IR2_Pin;
uint16_t ir_right = IN_IR3_Pin;
uint16_t ir_left = IO1_Pin;
```

```
// sets PWM frequency
int stepRate = 100;
int delayVal = 500/stepRate;

// Internal states
int state = 0;
int substate = 0;
int record_time = 0;
int time_elapsed = 0;
int en_step1 = 0;
int en_step2 = 0;

// IR Sensors
int ir_buf = 0;
int ir_sense = 250/delayVal;

// Sprayer
int spray_time = 2300;
int prime_time = 5000;
int needs_priming = 0;

// Mode switching
int prev_mode = 0; // 0-manual, 1-auto
int mode_buf = 0;

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
while (1)
{
  GPIO_PinState upBut = HAL_GPIO_ReadPin(GPIOB, IN_UP_Pin);
  GPIO_PinState downBut = HAL_GPIO_ReadPin(GPIOB, IN_DOWN_Pin);
  GPIO_PinState leftBut = HAL_GPIO_ReadPin(GPIOB, IN_LEFT_Pin);
  GPIO_PinState rightBut = HAL_GPIO_ReadPin(GPIOB, IN_RIGHT_Pin);

  GPIO_PinState manualBut = HAL_GPIO_ReadPin(GPIOA, IN_MANUAL_Pin);
  GPIO_PinState sprayBut = HAL_GPIO_ReadPin(GPIOA, IN_SPRAY_Pin);
  GPIO_PinState brushBut = HAL_GPIO_ReadPin(GPIOA, IN_BRUSH_Pin);


  if (!manualBut) {

          // Detect mode switch
          if (prev_mode == 0) {
```

```
        if (mode_buf > 250/delayVal) {
                prev_mode = 1;
                mode_buf = 0;
        }
        else {
                mode_buf ++;
        }
}

// Detect sprayer button pressed
if (!sprayBut) {
        needs_priming = 1;
}

// reset state
else if (state == 0) {
        HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 0);
        en_step1 = 0;
        en_step2 = 0;
        record_time = 0;
        time_elapsed = 0;

        // move to bottom left
        if (substate == 0) {
                // move left
                HAL_GPIO_WritePin(GPIOA, DIR2_Pin, left);
                en_step2 = 1;
                // detect at left
                if (!HAL_GPIO_ReadPin(GPIOB, ir_left)) {
                        ir_buf ++;
                }

                if (ir_buf >= ir_sense) {
                        ir_buf = 0;
                        substate = 1;
                        en_step2 = 0;
                }

        }
        else if (substate == 1) {
                // move down
                HAL_GPIO_WritePin(GPIOA, DIR1_Pin, down);
                en_step1 = 1;
                // detect at left
                if (!HAL_GPIO_ReadPin(GPIOB, ir_bottom)) {
                        ir_buf ++;
                }
                if (HAL_GPIO_ReadPin(GPIOB, ir_bottom) && ir_buf) {
```

```
                                        ir_buf --;
                                    }
                        if (ir_buf >= ir_sense) {
                                ir_buf = 0;
                                substate = 0;
                                en_step1 = 0;
                                if (needs_priming) {
                                        needs_priming = 1;
                                }
                                state = 6;
                        }
                }
        }

        // Spray state
        else if (state == 6) {
                int t = spray_time;
                if (needs_priming) {
                        t=prime_time;
                }
                // spray nossle
                record_time = 1;
                HAL_GPIO_WritePin(GPIOA, SPRAY_Pin, 1);
                if (time_elapsed >= t) {
                        record_time = 0;
                        time_elapsed = 0;
                        HAL_GPIO_WritePin(GPIOA, SPRAY_Pin, 0);
                        state = 1;
                        needs_priming=0;
                }
        }

        // Move up cleaning
        else if (state == 1) {
                HAL_GPIO_WritePin(GPIOA, DIR1_Pin, up);
                en_step1 = 1;
                HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 1);
                // detect at top
                if (!HAL_GPIO_ReadPin(GPIOB, ir_top)) {
                        ir_buf ++;
                }
                if (HAL_GPIO_ReadPin(GPIOB, ir_top) && ir_buf) {
                                        ir_buf --;
                                }
                if (ir_buf >= ir_sense) {
                        ir_buf = 0;
                        en_step1 = 0;
                        HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 0);
```

```
                state = 2;
            }

    }

    // Move right
    else if (state == 2) {
            record_time = 1;
            // move right
            HAL_GPIO_WritePin(GPIOA, DIR2_Pin, right);
            en_step2 = 1;
            // detect at right
            if (!HAL_GPIO_ReadPin(GPIOB, ir_right)) {
                    ir_buf ++;
            }
            if (ir_buf >= ir_sense) {
                    ir_buf = 0;
                    en_step2 = 0;
                    record_time = 0;
                    time_elapsed = 0;
                    state = 4;
            }
            // detect time elapsed
            else if (time_elapsed >= 2500) {
                    en_step2 = 0;
                    record_time = 0;
                    time_elapsed = 0;
                    state = 3;
            }
    }

    // Move down cleaning
    else if (state == 3) {
            HAL_GPIO_WritePin(GPIOA, DIR1_Pin, down);
            en_step1 = 1;
            HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 1);
            // detect at bottom
            if (!HAL_GPIO_ReadPin(GPIOB, ir_bottom)) {
                    ir_buf ++;
            }
            if (HAL_GPIO_ReadPin(GPIOB, ir_bottom) && ir_buf) {
                                    ir_buf --;
                            }
            if (ir_buf >= ir_sense) {
                    ir_buf = 0;
                    en_step1 = 0;
                    HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 0);
                    state = 6;
```

```
        }
}

// Move down cleaning (edge)
else if (state == 4) {
        HAL_GPIO_WritePin(GPIOA, DIR1_Pin, down);
        en_step1 = 1;
        HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 1);
        // detect at bottom
        if (!HAL_GPIO_ReadPin(GPIOB, ir_bottom)) {
                ir_buf++;
        }
        if (HAL_GPIO_ReadPin(GPIOB, ir_bottom) && ir_buf) {
                                ir_buf--;
                        }
        if (ir_buf >= ir_sense) {
                ir_buf = 0;
                en_step1 = 0;
                HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 0);
                state = 7;
        }
}

// Spray state (edge)
else if (state == 7) {
        int t = spray_time;
        if (needs_priming) {
                t=prime_time;
        }
        // spray at bottom
        record_time = 1;
        HAL_GPIO_WritePin(GPIOA, SPRAY_Pin, 1);
        if (time_elapsed >= t) {
                record_time = 0;
                time_elapsed = 0;
                HAL_GPIO_WritePin(GPIOA, SPRAY_Pin, 0);
                state = 5;
                needs_priming = 0;
        }

}

// Move up cleaning (edge)
else if (state == 5) {
        HAL_GPIO_WritePin(GPIOA, DIR1_Pin, up);
        en_step1 = 1;
        HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 1);
        // detect at top
```

```
            if (!HAL_GPIO_ReadPin(GPIOB, ir_top)) {
                    ir_buf ++;
            }
            if (HAL_GPIO_ReadPin(GPIOB, ir_top) && ir_buf) {
                                    ir_buf --;
                            }
            if (ir_buf >= ir_sense) {
                    ir_buf = 0;
                    en_step1 = 0;
                    HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 0);

                    // back to reset state
                    state = 0;
            }
    }

}

if (manualBut) {

    // Detect mode switch
    if (prev_mode == 1) {
                    if (mode_buf > 250/delayVal) {
                            prev_mode = 0;
                            mode_buf = 0;
                    }
                    else {
                            mode_buf ++;
                    }
            }

    en_step1 = 0;
    en_step2 = 0;

    // UP/DOWN Buttons
    if (!upBut && !downBut) {}
    else if (!upBut) {
            // Set dir 1 up
            HAL_GPIO_WritePin(GPIOA, DIR1_Pin, up);
            dir1val = up;
            // Enable step 1
            en_step1 = 1;
    } else if (!downBut) {
            // Set dir 1 down
            HAL_GPIO_WritePin(GPIOA, DIR1_Pin, down);
            dir1val = down;
            // Enable step 1
            en_step1 = 1;
```

```
        } else {
                en_step1 = 0;
        }


        // LEFT/RIGHT Buttons
        if (!leftBut && !rightBut) {}
        else if (!leftBut) {
                // Set dir 2 up
                HAL_GPIO_WritePin(GPIOA, DIR2_Pin, left);
                dir2val = left;
                // Enable step 2
                en_step2 = 1;
        } else if (!rightBut) {
                // Set dir 1 down
                HAL_GPIO_WritePin(GPIOA, DIR2_Pin, right);
                dir2val = right;
                // Enable step 1
                en_step2 = 1;
        } else {
                en_step2 = 0;
        }


        // SPRAY Button
        if (!sprayBut) {
                HAL_GPIO_WritePin(GPIOA, SPRAY_Pin, 1);
        } else {
                HAL_GPIO_WritePin(GPIOA, SPRAY_Pin, 0);
        }


        // BRUSH Button
        if (!brushBut) {
                HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 1);
        } else {
                HAL_GPIO_WritePin(GPIOA, BRUSH_Pin, 0);
        }

}


if (en_step1) {
        HAL_GPIO_TogglePin(GPIOA, STEP1_Pin);
} else {
        HAL_GPIO_WritePin(GPIOA, STEP1_Pin, 0);
}
if (en_step2) {
```

```
            HAL_GPIO_TogglePin(GPIOA, STEP2_Pin);
    } else {
            HAL_GPIO_WritePin(GPIOA, STEP2_Pin, 0);
    }

    HAL_Delay(delayVal);

    if (record_time) {
            time_elapsed += delayVal;
    }
      /* USER CODE END WHILE */

      /* USER CODE BEGIN 3 */
}
```

C Appendix - Bill of Material Table

Table 7: Bill of Materials

Component	Part Number/Link	Quantity	Unit Cost	Total Cost
Control System and User Interface				
Momentary Push Buttons	Push button NO	6	1.74	10.64
Toggle Switch	Toggle Switch	1	1.94	1.94
1x2 Connector	1x2 header	20	0	0
2x3 Connector	2x3 header	1	0	0
AC-DC Jack	barrel jack	1	0.69	0.69
10 Pin Header	10 pin header	1	0	0
10nF Cap	10nF cap	3	0.12	0.36
100k Resistor	100k resistor	3	0	0
1uF Cap	1uF cap	4	0	0
4.7uF Cap	4.7uF cap	1	0	0
0.1uF Cap	0.1uF cap	1	0	0
10uF Cap	10uF cap	1	0	0
10k Resistor	10k resistor	3	0	0
ESP 32 Microcontroller	ESP32-S3-WROOM-1-N8	1	0	0
80/20 Rails	Railing	3	0	0
T-Slotted Track Rollers	2 Flange Slot	3	0	0
Timing Belt	Belt	3	0	0
Drivetrain				
NEMA 23 Motors	NEMA 23 motors	0	18.91	0
Infrared Laser	IR laser	4	5.95	23.80
Motor Drivers	DM542T	2	9.99	0
Cleaning Mechanisms				
Microfiber Cloth	Microfiber Cloth	1	4.99	4.99
Rotating Brush (Soft)	Amazon Brush	1	16.99	16.99
DC Motor	DC motor	1	0	0
Solution Container	Container	1	5.09	5.09
Sprayer	Sprayer	1	9.88	9.88
Energy Conversion				
AC to DC Converter	24V 6A Converter	1	22.59	22.59
24V to 12V Converter	N7812-2CH	2	5.99	11.98
12V to 5V Converter	LC78 05-3.0	1	13.4	13.4
5V to 3.3V Regulator	MIC5317-3.3YM5-TR	2	0.22	0.44
Coiled Cable	Coiled Power Cable	1	4.99	4.99
Total				**127.78**