# SYDE 361

Nicholas Barton (20877569), Jack Eshkenazi (20893647), Greg Fouzie (20889968), Curtis Franz (20831938), Azaan Wazir (20877312)

---

Prototype - Solar Panel Powered Fan
Components 2-4

## Component 2: Insights

**Insight #1:** When building the initial prototype, there was an issue of there not being enough space to place all electrical components securely within the product.

**Method:** The design team was able to find this issue by initially attempting to place all the components and seeing that they did not all fit.

**Application:** The team measured the dimensions of all the components to see if there was any feasible method to place all the parts within the container. Realizing that the electrical components did not all fit in the design, the team decided to redesign the electrical container. The container was bound horizontally such that it had to fit within an existing piece, so the piece was expanded vertically to allow more components to sit inside. This insight applies to the project as this forced a change to the aesthetics of the design. Previously, the component box was small and didn't stick out when being used, but now with the longer container the electrical container stuck out. The team could address this in later design iterations by making the bounding component larger horizontally or extending the container base vertically such that when tilted the electrical container doesn't stick out.

**Insight #2:** The initial fan cage that was designed was no longer required for the prototype.

**Method:** The team found the originally designed fan cage to be unnecessary, due to the fan being slow and therefore not harmful to the user. This was discovered by gradually increasing the fan speed through increasing current and testing the force exerted on the test user's hands. Once the upper bound of the producible fan speed was reached, the team was able to derive that it wouldn't cause any pain when interacting with the fan.

**Application:** This insight was useful for the project to help simplify the physical complexity of the design and allow for the product to be built in the limited time that was given. In future designs and iterations, the team will need to find out the theoretical max current output of the panel and the max rotations per minute of the fan in the final design. Using research and analysis the team can find out whether the fan will be damaging or dangerous to the user. The fan cage will then be implemented as planned should there be any danger.

**Insight #3:** The summer season tilt angle of the solar panel could not be reached using the device which did not properly allow the solar panel to have the correct tilt angle for all seasons.

**Method:** The team was able to find this issue during the initial testing. When the team attempted to test all the seasonal configurations (tilt angles) of the solar panel, the rotation of the solar panel was blocked due to its platform being too long. This led to the rotation being stopped and the summer tilt angle being unreachable.

**Application:** This insight applies to the project since it directly impacts the design of the prototype. This impacted the design by forcing the team to decrease the solar panel platform length. The redesign did not change the overall look of the design, but it did make it harder to attach pieces together. The surface area for the command strips was smaller which led to having less grip between the components. Overall, the prototype still functioned reliably, but for future iterations where parts could be scaled, all tilt angles should be simulated before being tested physically to ensure functionality without risking redesign.

## Component 3: Prototype Verification

The principal objective of data collecting was to gain insight in two particular areas. The first was to determine the degree to which different levels of cloud cover affect the maximum theoretical power output. The second was to determine how light-based tracking affects the maximum theoretical power output.

Three unique sets of data were collected (Table 3.1). All tests were performed at the same picnic bench near the north entrance to E5. In each test, the system was placed on top of the picnic bench such that the solar panel was exposed to the sky with no obstructions. The angle of elevation of the solar panel was set to 20° which corresponds to the 4th position of the adjustment wheel, optimal for June-September in Waterloo (see figure 4.1). Finally, a fine sampling period of 0.5 seconds was used to capture transient changes in the metrics being collected.

**Table 3.1** - Testing Conditions

| | **Test 1** | **Test 2** | **Test 3** |
|---|---|---|---|
| **Date. time** | 13/07/23, 2:30-3 pm | 16/07/23, 2:30-3 pm | 16/07/23, 3-3:30 pm |
| **Sun condition** | Cloudy | Partly Sunny/Hazy | Partly Sunny/Hazy |
| **Tracking System** | Single Axis Tracking | Single Axis Tracking | Fixed |

Figure 3.1 compares the maximum theoretical power yields of the solar panel in cloudy versus sunny conditions. From about 2:30 to 2:57 pm, there is a significant amount of variation in maximum theoretical power yields in both plots due to transient changes in the sun conditions due to clouds and cloud breaks. However, from 2:43 to 2:57 pm there is a clear disparity in maximum theoretical power yields. During this timeframe, the respective sun conductions held mostly steady. Here, test 2 (mostly sunny conditions) had theoretical power yields of approximately 250mW, around 5x, the theoretical power yields of test 1 (mostly cloudy conditions), which hovered around 50mW. This deviation shows how significantly power yields are affected by cloud conditions. This information would be useful to communicate to users in a user manual to inform them when and what sun conditions the system can realistically be used to power the fan. This result would also be valuable to consider in future design iterations in the development of power-saving features.

Interestingly, there are a couple of maximum theoretical power spikes in the test 1 plot where the sun broke out. At 2:58 pm the theoretical maximum power levels reached a peak of 551mW. This exceeded the maximum theoretical power observed over the entire duration of test 2 (mostly sunny conditions), due to some haziness during test 2.

Figure 3.2 compares the maximum theoretical power yields of the solar panel with and without single-axis tracking. In test 2, the panel utilizes a pair of photoresistors to orient the panel such that it receives optimal light exposure. While the first 13 minutes of test 2 depict varying sun exposure due to fleeting clouds, the maximum theoretical power peaks over the course of the entire interval remain fairly steady (around 250mW). In test 3, the panel was fixed at the optimal azimuth the moment the testing interval began. As time passed and the sun's azimuth changed, the panel's angle became increasingly divergent. The plot for test 3 initially shows a steady maximum theoretical power yield until around 3:18 pm at which point it steadily drops. This suggests that using a single-axis tracking system is indeed beneficial, as it results in steadier maximum theoretical power yields.

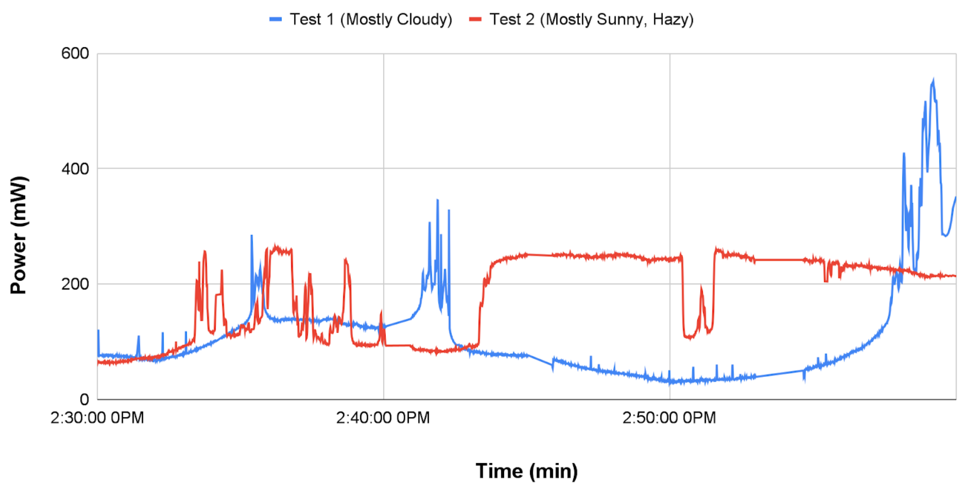



**Figure 3.1** - Theoretical maximum power yields of the solar panel in cloudy versus sunny conditions

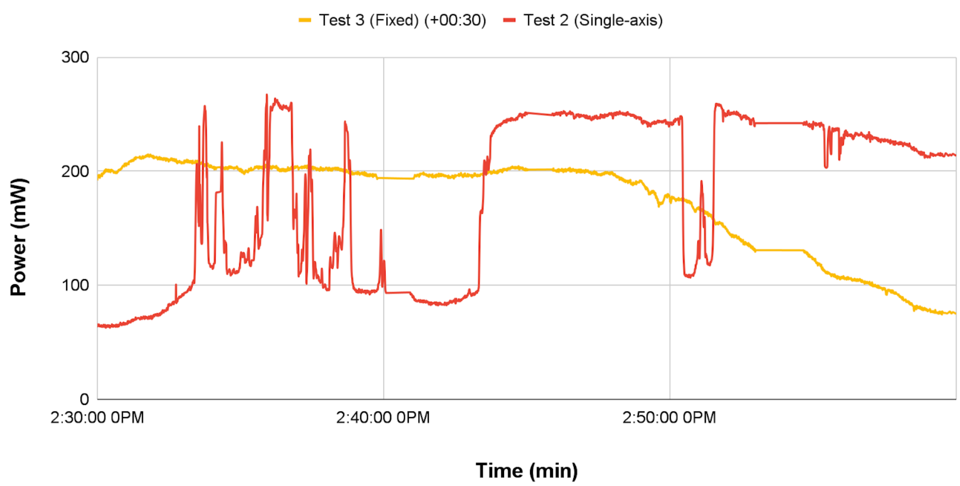



**Figure 3.2** - Theoretical maximum power yields of the solar panel with and without single-axis tracking

Unfortunately, there were several limitations in the testing methodology used for data collecting. Some of these include the uncontrollable variations in weather and sun conditions, having collected data on different days, testing periods limited to 30 minutes, testing limited to a single location and time period, etc.

# Component 4: Design Documentation

## A) Electronic Hardware:

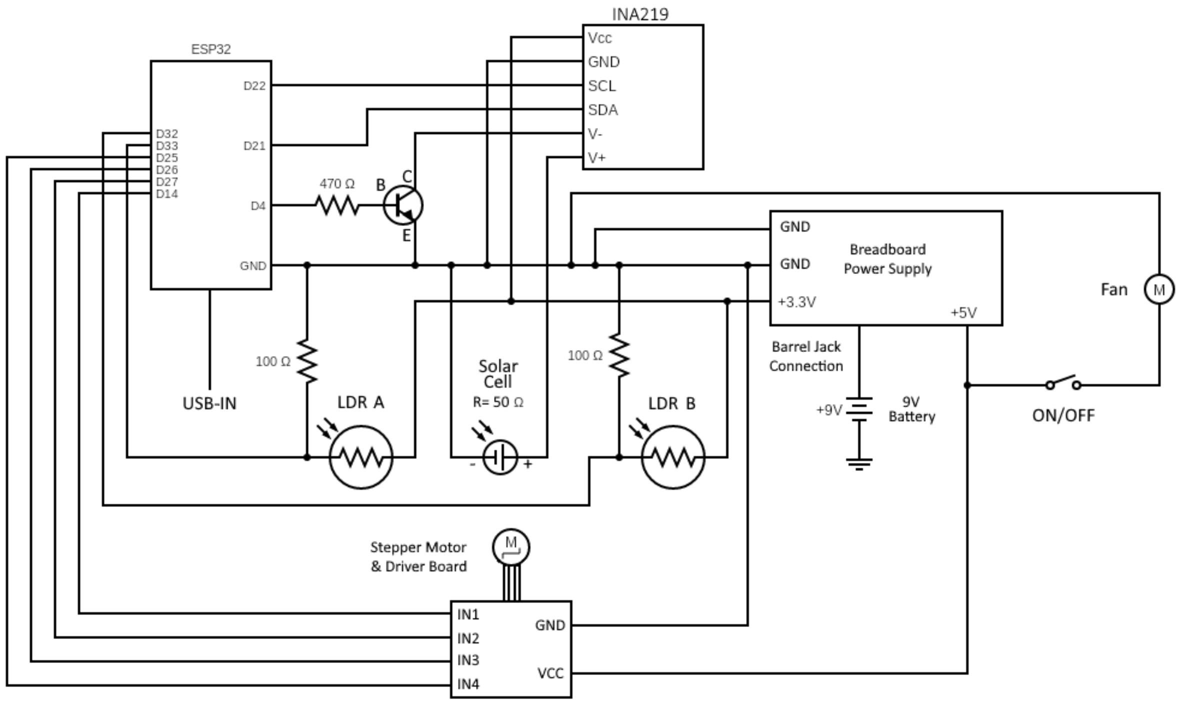



**Figure 4.1** - Fully labeled schematic including all components of the prototype

**Photoresistors/LDRs:** The photoresistors or light-dependent resistors (LDR) change their resistance depending on how much light they receive. This serves the purpose of calibrating the direction of the solar panel and tracking the sun throughout the day. Two LDRs are angled and pointed in opposite directions so they can detect the light from both sides of the solar panel. When one detects more light than the other, the code instructs the motor to move in the direction of the LDR that is detecting more light. This is how they are used to track the point of maximum power from the sun and get the maximum amount of power throughout the day.
**100Ω Pull-down Resistors:** One 100Ω pull-down resistor is placed in series with each LDR to allow the light data to be read by the microcontroller. To get the data from the LDRs, the voltage is measured across the 100Ω resistor. When it is bright, the resistance of the LDR drops which creates an increase in voltage across the pull-down resistor. This higher voltage reading indicates an increase in light detected by the LDR. This makes the 100Ω resistors essential for light detection in the prototype.
**9V Battery:** The 9V battery supplies power to the breadboard power supply which regulates the voltage and supplies both 3.3V and 5V to the components that need it. In later stages of the prototype, this battery would ideally be charged by the solar panel.
**5V DC Motor:** The 5V DC motor takes in 5V of power and turns a shaft that a fan blade is attached to. The motor rotates the fan blades quickly to cool the user on a hot and sunny day.
**ON/OFF Switch:** The ON/OFF switch is used to turn the fan on and off. If the user does not need the fan and wants to conserve power, they can turn the fan off and then back on when needed again with ease.

**Power Estimation:**
The average power consumption used by the circuit is measured by summing the average current draw from each component in the entire system.

**ESP32:** When the ESP32 is in active mode, it draws an average of 240mA of current. Since the ESP32 is always in active mode in the system, it continuously draws 240mA of current.
**INA219:** From the INA219 datasheet, it takes a supply voltage of 3.3V and has a maximum current draw of 1mA. Since this value is low compared to the rest of the system the maximum current draw can be used in the estimation.
**LDRs & Pull-down Resistors:** Since the resistance and therefore current draw changes based on the light the LDRs receive, the current estimation will be measured in sunny conditions since that is when the system is most useful. In sunny conditions, the average current draw from both LDRs and both pull-down resistors was measured to be about 2.5mA each.
**DC Motor:** The DC motor, when powered on, was measured to consume an average of 417mA of current. However, it is hard to predict how often the user will have the fan on, since that is up to the user. Therefore, two power estimates will be made, one with it on and one with it off.
**Stepper Motor:** The 28BYJ-48 stepper motor typically draws about 240mA when powered on. In the system, outputs are disabled when it is not moving to save power. The ESP32 polls both LDRs every 15s but the motor does not necessarily move every time it polls. Therefore from observation during data collection, the motor activates about every 10 mins for around 5 seconds making the average power consumption 240*5/600 = 2mA.
**Transistor:** The transistor consumes very little current in the system, the base current was measured to be around 0.02mA which is very small and can therefore be neglected. The collector current was neglected as well.

After summing the current consumption from every component, the total power or current consumption of the system is 665mA when the fan is on and 248mA when it is off.

Possible approaches for reducing power consumption include putting the ESP32 in sleep mode when not polling or moving the motor. Since it is not necessary to get data from the current sensor at every second when not gathering data, the ESP32 can be put in Modem Sleep mode. In Modem Sleep mode, everything is active except for the Wi-fi, Bluetooth and the radio. When not gathering data, those three functionalities of the ESP32 are not necessary and the ESP32 can consume significantly less power. In Modem Sleep mode, the ESP32 only consumes 3mA at low speeds reducing about 237mA of current draw.

The other approach to reducing power consumption is setting a limit to how long the user can use the fan. The majority of the power consumption in the system is done by the DC motor that the fan is attached to. To reduce power consumption during future design iterations, the team can explore ways to limit the power used by the fan. This can be done by limiting the current and thus lowering the intensity of the fan when on for longer periods of time. This unfortunately does negatively impact the user’s experience and should be avoided unless completely necessary.

**B) Microcontroller Software**

The code that was run by the ES32 needed to accomplish two things: It needed to turn the stepper motor in a manner which tracked the sun, and it needed to monitor the current and voltage outputs from the solar panels. The code's secondary objective was to do all this using as little power as possible.

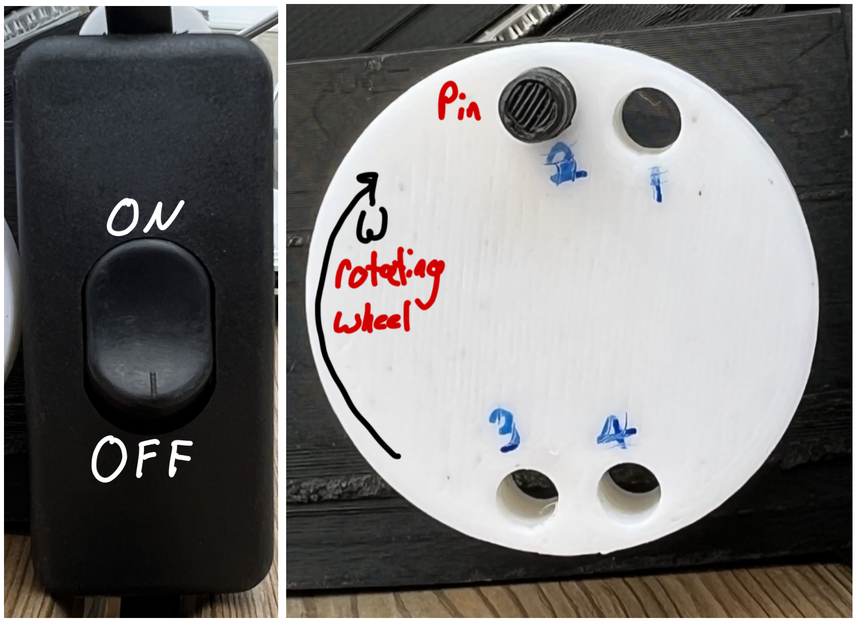
In order to track the movements of the sun, the code utilized a simple algorithm which used the photoresistor outputs as input. This algorithm was called during the "unmoved" mode of the prototype, and it worked in the following way: If the right photoresistor had a higher light intensity value than the left, then it turned right and if the left value was higher then it turned left. As soon as the code noticed that the left and right values were equal it stopped rotating. The photoresistor light intensity values were quantized into bigger buckets than the units they were measured in, meaning that there was a leniency of plus/minus 35 units. For example, if the right photoresistor's value is less than 35 units apart from the left photoresistor's value, they are considered equal. It was necessary to have the photoresistors facing different angles because otherwise, they would always have the same output from the sun since it is so far away that its rays are essentially parallel to both of them.

Upon initially booting up, the code starts off in the "unmoved" state, which pings the outputs of the photoresistors every 30 milliseconds. It takes an average of 20 samples of these outputs before deciding to make a movement with the motor, which adds up to a call to the stepper object every 0.6 seconds. Since we used the "moveTo" method however, the motor wouldn't appear to stop moving instantaneously, only accelerate and then decelerate quickly once it found the most favorable point. Once the system has found the most favourable point, it goes into a new state called "power_save" which only pings the photoresistor outputs every 15 seconds in order to save energy. Upon pinging, if the values are not equal, it would go back into an "unmoved" state and find the optimal positioning once again.

The reason that the ESP32 couldn't be put into a sleeping mode during our "power_save" state to save even more energy, is that we needed to constantly be tracking solar panel outputs for our testing. In order to measure both voltage and current, we followed closely to the design used in Lab 3, in which we utilized a transistor to change the INA-219 from being in parallel to the solar panel to measure the voltage, to being in series with it to measure its current. These power samples were taken every 30 milliseconds. In order to process this data into the charts shown above, the time stamp, voltage, current, and power (voltage times current) were placed into a comma-separated string sent via serial port to a terminal instance of an app called "CoolTerm" from which it was easy to record them and compile them into a .csv file, and then imported into our spreadsheet application of choice.

**C) Interface**

The interface for this prototype consists of manual adjustments, buttons, lights, and switches. There is a master power switch with a corresponding light that indicates if it's on or off, but that interface is relatively simple and should not be used after setting up the prototype. Rather, the focus for this section of the document will be on a manual adjustment component and a switch.




**Figures 5.0, 5.1 -** Power Switch and Manual Adjustment Wheel

Firstly, the power switch aims to allow users to control whether the fan is operating or not. If a user switches the interface in one direction, it will result in the fan beginning to move and therefore the system will cool down the user, and switching it in the other direction will turn the fan off. Users will know when the fan is on once they hear the motor and feel the wind from the blades and they will know when it turns off via the opposite indications.

Further, the manual adjustment wheel aims to allow users to adjust the elevation of the solar panel and therefore optimize the current output based on the month of the year. Theoretically, users would receive a manual when purchasing the fan and it would contain instructions for setting up the wheel. Different months of the year are approximated to have optimal elevation angles of 20° (June-September), 35° (May, October), 50° (April, November), and 65° (December-March) degrees which correspond to holes 4, 3, 2, and 1 respectively. Users can change the setting by removing the pin and then rotating the white wheel to the desired pin, and then sliding the pin back in. Users will know that they have successfully changed the elevation angle when the hole is correct according to the current month in the manual, and the pin is back in a hole such that the wheel no longer rotates.