# Auto Sun Visor

## ECE 445 Design Document — Spring 2024

Project #4

Blair Huang, Xiaoyang Tian, Siying Wang

TA: Douglas Yu

**Table of Contents**

# 1. Introduction

## 1.1 Problem

As drivers go through urban areas or blocks on sunny days, it's common practice to use sun visors when sunlight interferes with visibility. However, the frequent turns encountered in such environments can make sunlight seem erratic. Attempting to adjust the sun visor while driving the vehicle can lead to brief periods of distraction and reduced visibility, which is not only unsafe but increases the risk of accidents, including potential collisions with pedestrians. According to data from the National Highway Traffic Safety Administration, sun glare contributes to approximately 9,000 traffic accidents annually in the United States [1]. This statistic shows the need for drivers to exercise caution and seek alternative solutions to manage sunshine interruption while driving, especially in densely populated areas where the risk of accidents is heightened. Polarized sunglasses can reduce sun glare in certain areas. However, drivers who already need prescription glasses would need either to customize prescription sunglasses or add another polarized filter on their own glasses, which can be inconvenient. Moreover, wearing sunglasses while driving could potentially lead to accidents due to reduced visibility under certain conditions. Also, this era's technology also allows electrochromic glasses, which could be the most ideal choice to respond to sun glare if it would not be that expensive.

## 1.2 Solution

An auto-adjustable sun visor that is powered by vehicle electricity and reacts to the direction of the sun relative to the vehicle can be a safe choice for drivers to avoid manually modifying the visor position. Our solution seeks to provide a compromise between the expense of photochromic glasses and the manual adjustment of sun visors or wearing sunglasses. This project has two hardware parts, the first part is the robotic arm holding the visor, and the second part is the sensor system. We will install sensors surrounding the vehicle body to determine sun position relative to the car, then take the sensors' input into our algorithm to decide the rotation of our robotic arm to let the visor block the sunlight accordingly under the reaction time that won't let people's eyes be exposed under sunlight.

### 1.3 Visual Aid

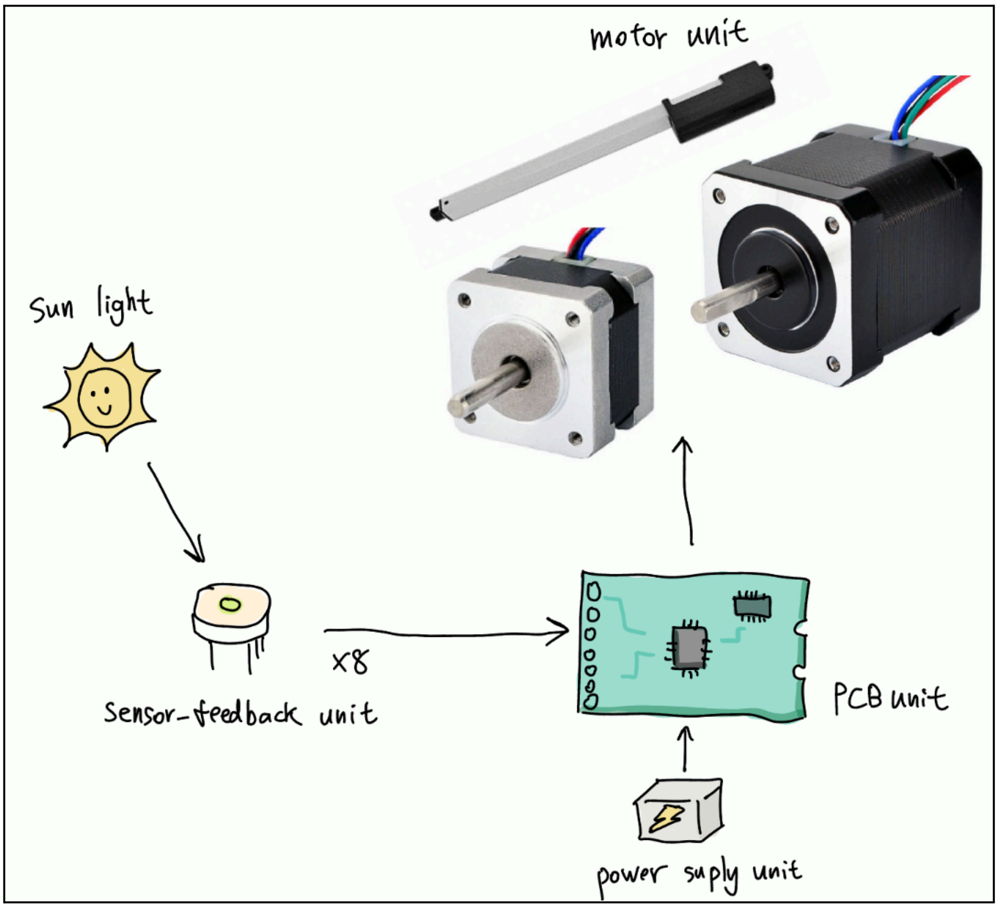



Figure 1.1: Breakdown Relation

The basic idea is the installed light sensors will always feed back the sun intensity measured to the PCB unit, and the PCB unit will calculate the position of the sun relative to the vehicle, and output the motion result to the motors, which will be assembled as a robotic arm holding a sun visor. As shown in figure 1.2, is the basic relationship of our product's principle of performance.

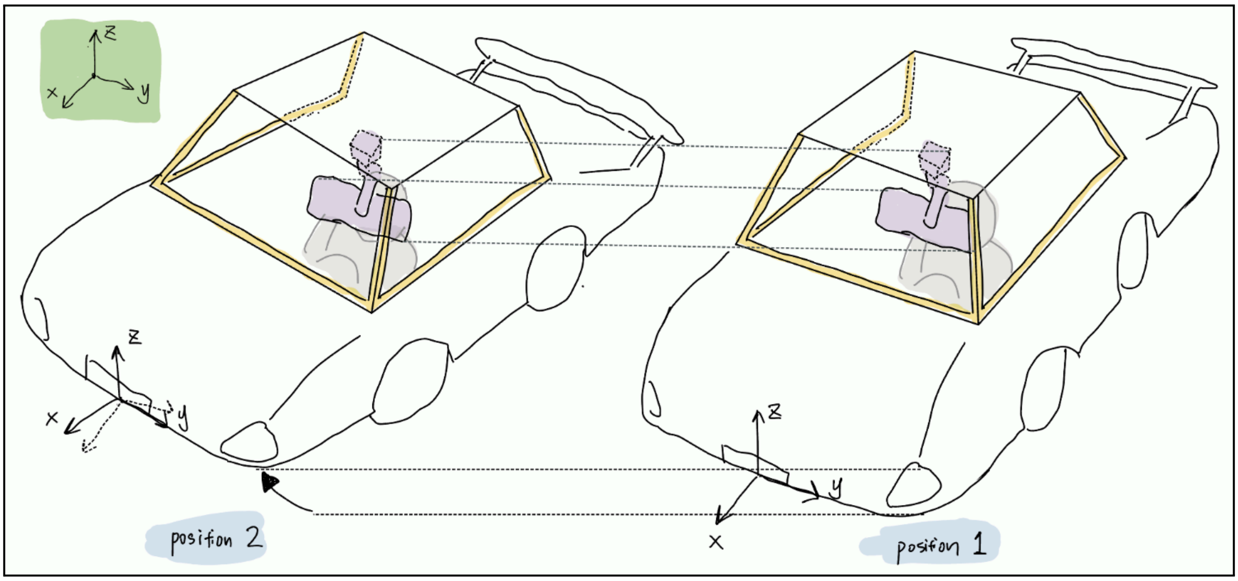


Figure 1.2 - Demo Frame

This product has not been applied into the market yet, so we cannot compare. Our designs are purely based on theory and machine design. As shown in figure 1.2, position 1, highlighted in purple, is where the ideal product prototype will be installed relative to a car and driver. Light sensors will be installed around the car frame highlighted yellow in figure 1.2. Instead of the classical manual adjusting the sun visor, we replace the place with our automatic sun visor, and the visor is blocking the sun light due to the observer's position overlapping with the sun 's light position. The highlighted green reference frame is showing the visor’s body frame. We can see from the dotted line, when the car adjust its angle relative to the sun, the visor’s position relative to the sun does not change, and this is the purpose of the product when the car make a turn, while the sun do not change position, the visor is relative in the same position as blocking the sun for the driver when the car make turns as shown from position 1 to position 2. The car’s frame is labeled in front of the car and we can see the visor rotates relative to the vehicle.

In our actual demo and testing scenario, we will reduce the size of the visor and also replace the real vehicle with a reduced seized car frame as shown in figure 1.3. Additionally we will install a handler to hold the car frame and we can rotate our body to monitor the turn of vehicle.

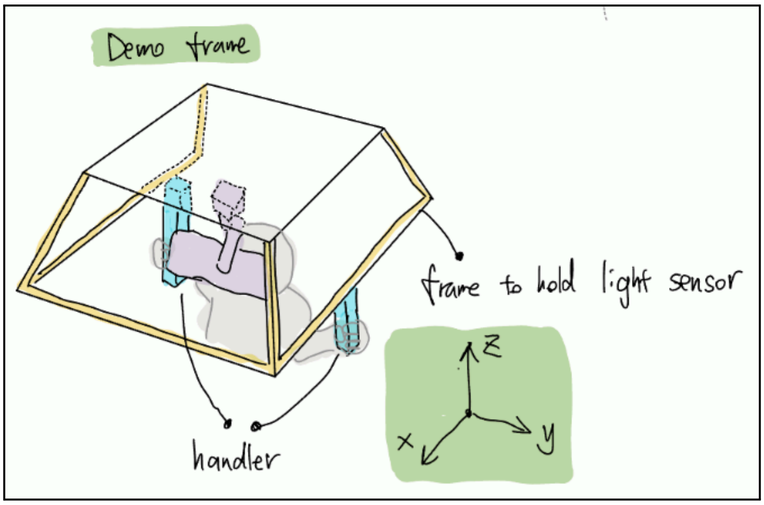


*Figure 1.3: Demo Frame*

### 1.4 High-level Requirement List

- A stabilized 12 V power supply for the whole system to simulate the power status when the car engine is started. Also a backup rechargeable lithium power supply should be able to connect to the remaining systems within 3 seconds in case the vehicle's power source fails to support the system. Also need to have enough power to let the visor return to original place when car engine cut off
- With ESP32-S3 as the microcontroller installed on the PCB board. The microcontroller needs to identify the different levels of light intensity and therefore provide a proper visor angle to cover the sunshine. The system should be able to correctly handle situations such as vehicles entering tunnels, the constant change of sunlight (on winding roads), and weather transitions (from sunny to overcast).
- The light sensor and microcontroller must be able to accept light intensity (lux) as data input and correctly input the data into the PCB unit within 1 seconds. As the statistic shows direct sunlight has luminosity between 32,000 lux to 100,000 lux [4], and ambient Daylight is 10,000 lux to 25,000 lux. Since the sensor is installed outside the car and supposed to directly face the sunlight, we do not consider ambient light, and we will use TSL2591 with range 188u lux to 88,000 lux range to detect.

According to  research, 6,000 lux is already a relatively high intensity for humans to see, so we will set the threshold as 6,000 lux.  The motor should be capable of moving the shading board to the correct position within 0.5 seconds after receiving a move command.

- The motor driver should effectively convert the lower voltage power provided to the board back to 12V to supply the three motors. Also, it should provide the correct microstep as programmed in the microcontroller.

## 2. Design

### 2.1 Physical Design

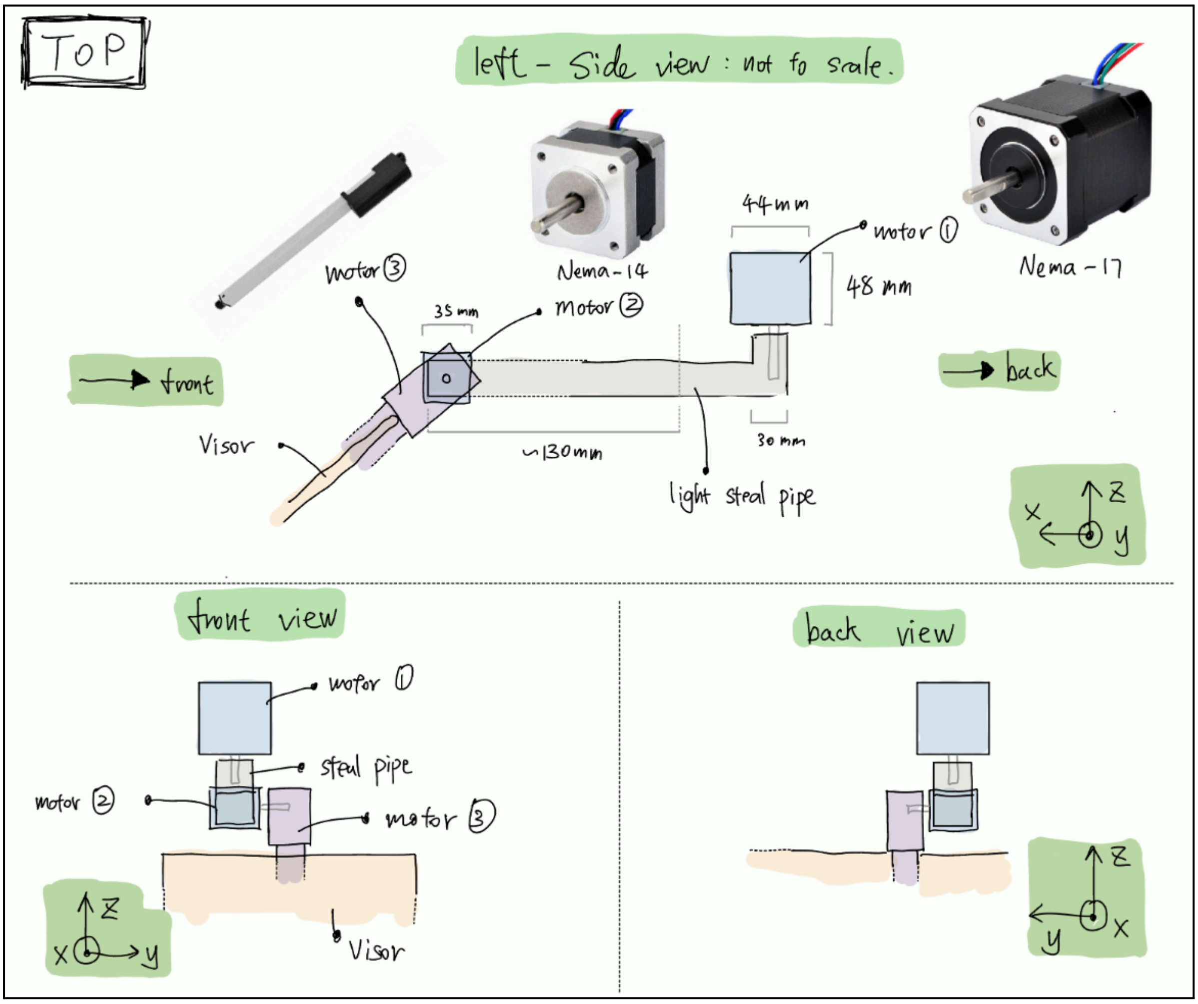



Figure 2.1: Visor Handler Mechanics

As in figure 2.1, these three views only show the mechanical part for the robotic arm that consisted with two stepper motors and one linear actuator to form two revolutional joints and one prismatic joint. Accordingly, this mechanism design lets the sun visor board contain enough degree of freedom that it can appear in any location the driver needs. For the left-side view in figure 2.1, we starting to have the 17mm stepper motor[1] connected to the vehicle ceiling, and a steel pipe connected with motor[1]'s shaft to link to the rear of another stepper motor[2] with 14mm dimension, and the motor's shaft directly connected with a converter linked to the linear actuator, then the linear actuator's extension rod is directly connected with a light weighted visor. The motor[1] is mainly in charge of rotating the visor left and right , and the motor[2] works to adjust the angle of the visor front and back, then the linear actuator helps the motor go up and down. The reason we do not use the classic design for vehicle sun visors is because flipping the visor could harm/bother the driver under unaware automotive situation, so our design is letting the visor "slide" down to help cover the sun glare that will not affect the head position of the driver.

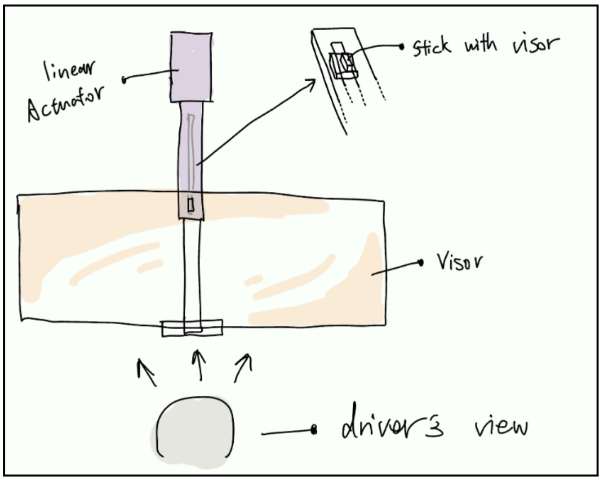



Figure 2.2: Linear actuator – Visor connector design

The figure 2.2 shows the head of linear actuator's extension rod is having an holder that holds the visor board, and the other side of the visor board is attached to a slider that can help the visor freely move horizontally correspond to the extension rod's motion but not fall from the arm.

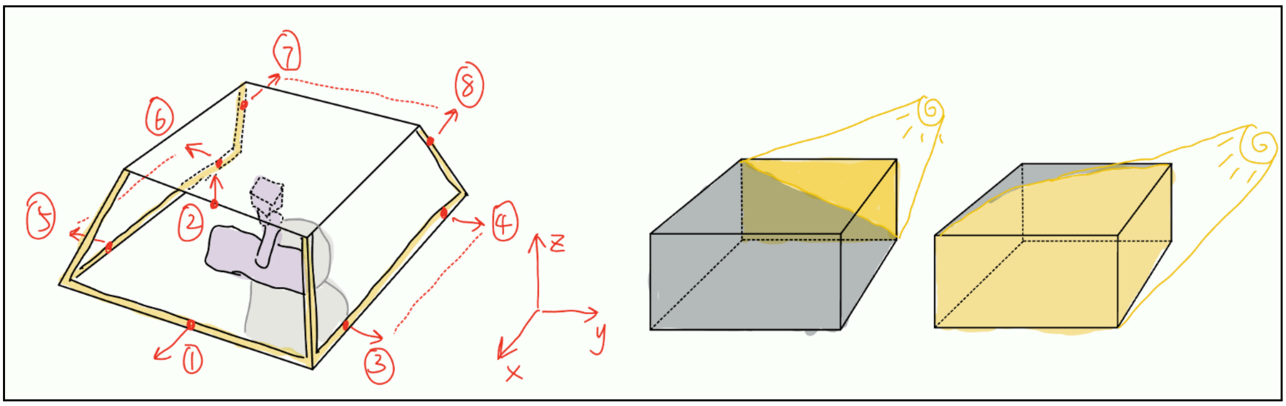



Figure 2.3 - Sensor installation design

For light sensor installation, we need to make sure to find the position of the sun, so first we can imagine the driver's cab as a rectangular box, as shown in figure 2.3, and the different angle's light rotation needs to be taken into account. So we need a sensor face on top of the car, and also sides of the car. Also because it is rectangular, the light sometimes will not cover all its longer side, as shown in the left light rectangular example, so we need to install two sensors for the longer side.

## 2.2 Block diagram

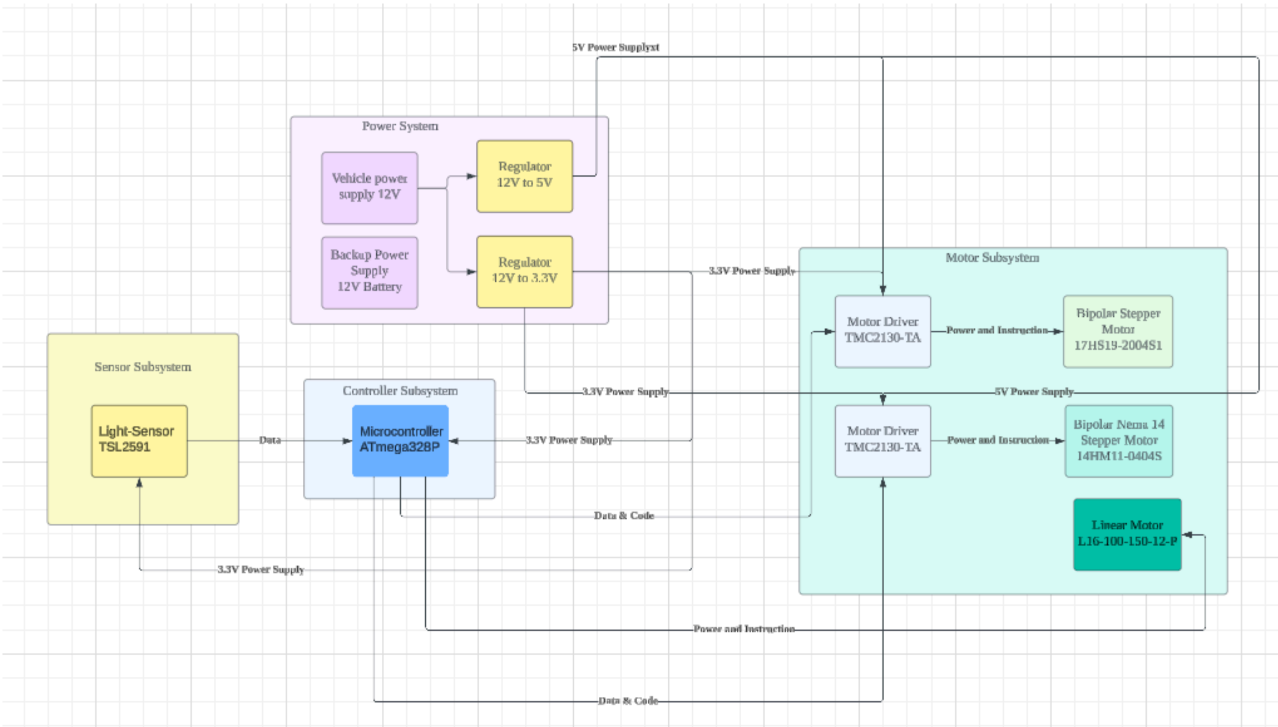



Figure 2.4: Block Diagram

## 2.3 Subsystem Overview

○ Sensor Subsystem: A total of 8 light sensors will receive light intensity information, enabling the control system to determine how the visor should rotate. These sensors will be placed at different positions at the front of the vehicle window to facilitate the determination of the visor's optimal position.

○ Controller Subsystem: The control system receives information from all the light sensors and converts it into a packet that is transmitted via a data cable to the motor, enabling the motor to adjust the control arm according to the instructions received.

○ Motor Subsystem: The motor system comprises three motors, a steel pipe, and a visor. The main motor is responsible for controlling the steel pipe, while the remaining two motors facilitate the rotation and movement of the visor.

○ Power Subsystem: This is a straightforward power system to supply all the rest of the subsystem.

### 2.3.1 Subsystem Requirement

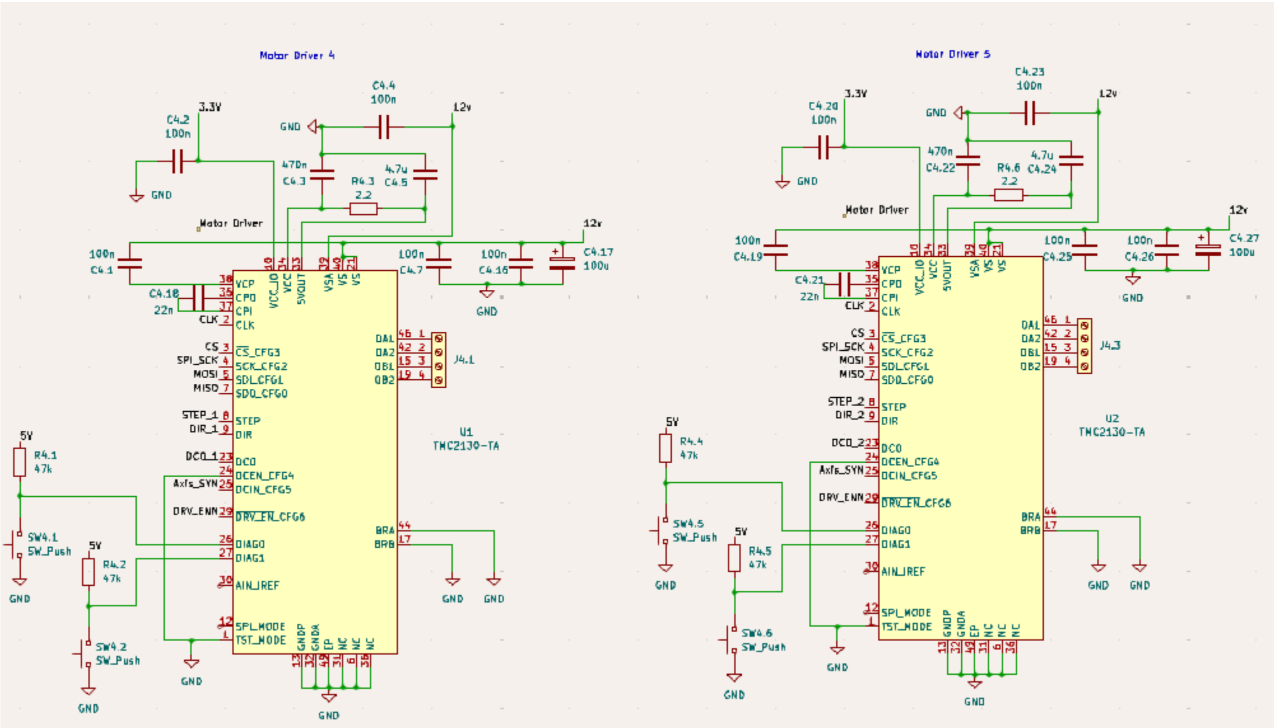
● **Motor Unit:** The initial motion should finish within 4 seconds, and the adjustment motion should react within 0.5 seconds.

○ **Motor[1]** Nema-17 stepper motor: should be solidly fixed on the ceiling and with enough torque be controlled to move in both directions (clockwise and counterclockwise) under a prepared threshold to move the light-weighted steel pipe. PID control system enabled to compromise the inertia effect with right/left turns (y-axis).

○ **Motor[2]** Nema-14 stepper motor: should be solidly attached to the end of the steel pipe and with enough torque be controlled to move in both directions (clockwise and counterclockwise) under a prepared threshold to move the Motor[3]. PID control system enabled to minimize the jitter caused by driving and stopping (x-axis), and car bouncing (z-axis).

○ **Motor[3]** L16-100-150-12-P linear actuator: is already a self-contained unit with integrated motor driver, and uses this to be solidly attached with the light-weighted visor and be programmed to move up/down accordingly.

○ **Motor driver** for stepper motors TMC2130: Motor driver is mainly used to do voltage regulation that microcontroller can not take such high voltage like 12V, so the driver needs to help the power out from the microcontroller back to 12V to supply to motors. Also to control the rotation of the stepper motor, the driver is very necessary. This selected driver can work with our selected stepper motors and provide 1/256 microstep under a smooth and silent operation condition, and work for both stepper motors. The selected linear actuator already has a built-in driver so we do not need to get another driver for it.




| Requirements | Verification |
|---|---|
| Be able to react to the order from control unit within 2s<br>Be able to change to different angle with given order from control system | - Power up the motors and ensure each motor works well (just the basic check whether motors work)<br>- Connect with the pcb board. Use the backend to send orders. Make sure the motor could move.<br>- Connect pcb with sensor and check our |

| | |
|---|---|
| | programming<br>- Measure the time took for motor to start to move after light source changed is within 2s |

- **Power Supply Unit:**
  - 12V DC main power supply (B[1]): to simulate the real vehicle electricity. It can charge the whole visor system with steady power.
  - 12V DC backup battery (B[2]): as the backup power supply. Should be able to connect to the remaining systems within 5 seconds in case the vehicle's power source fails to support the system when the car engine is cut off. Also recharging during the engine open process to ensure the sufficient power stored for next use.

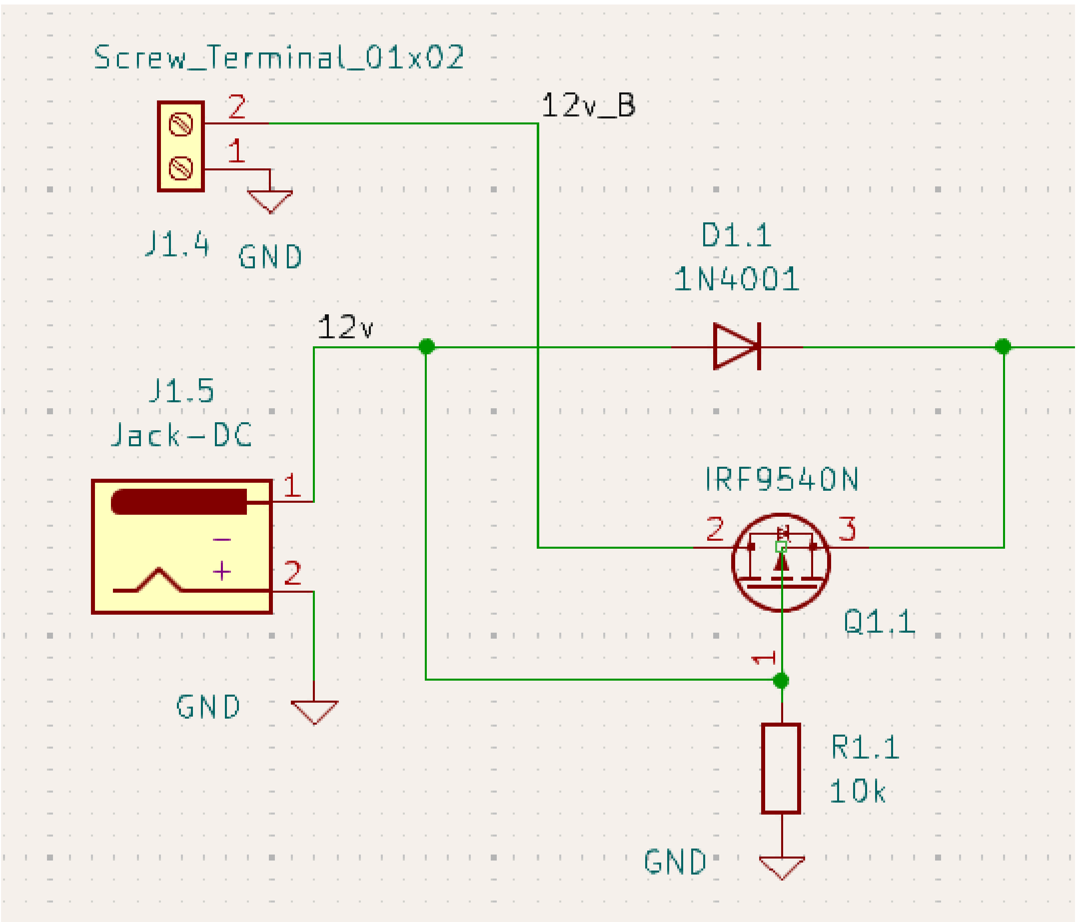


Figure 2.5: Automatic power replacement

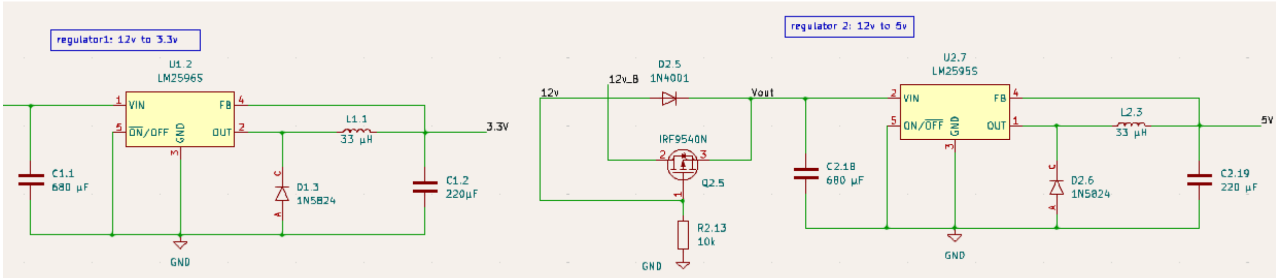


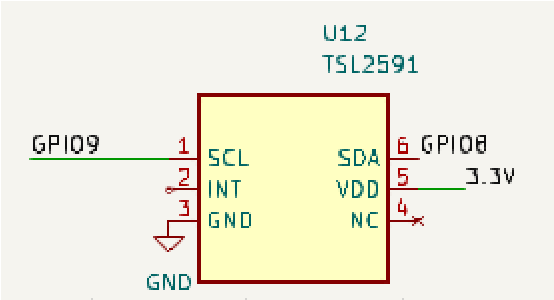

Figure 2.6: 12V to 3.3V Regulator and 12V to 5V Regulator

As we mentioned below, we need 3.3V and 5V to power the motor drivers and therefore we need two voltage regulators to obtain the proper power.

| Requirements | Verification |
|---|---|
| Be able to provide 11.5V to 12.5V power supply to make all the subsystems work<br><br>Be able to switch from car providing power to back up power if needed | -Power the whole system<br>-Measure the current and voltage of every branch and item to check whether they match with the rated currents and voltages<br>-Shut down the car providing power and measure the magnitude at every branch and item to check whether backup power works successfully |

- **Sensor-feedback Unit**: Use TSL2591 light sensors to get Luminosity (lux) feedback as input into the PCB algorithm. The range of Lux for this sensor is 188µ lux - 88,000 lux (*AMS*) which is sufficient since the sunlight Luminosity is 32,000 to 100,000 lux (*Lux, Lumens and Watts: Our Guide*). The data will be filtered by a threshold (lux) to determine the necessity (true/false) of generating the visor. Totally 8 light sensors.
    - Front sense (+x) will have one up and one down.
    - Left sense (+y) will have one up and one down.

- Right sense (-y) will have one up and one down.
- Back sense (-x) will have both installed on the side frame.




| Requirement | Verification |
| --- | --- |
| Be able to detect light intensity changes within 1s | - Power up the light sensor and connect to backend<br>- Try covering by hand or shining a lamp onto the sensor to experiment with the light levels<br>- Check the data from the backend and make sure the lux output is valid<br>- Repeat step 2 and measure response time. Ensure sensor takes less than 1s to give feedback |

- **PCB Unit**: The control board contains enough ports for wire connection, correct chips, and microcontroller ESP32-S3. This part should take the 8 outputs' data of the sense-feedback unit as input and transfer it into the microcontroller to generate three outputs back to the motor unit to control each motor. The output of the sensor chip should be lux thus our receiving microcontroller should be able to read in lux, and our programing algorithm should be also in lux units to reflect in the real-world simulation.  Enable pull-up resistor to stabilize power supply to motor. The algorithm used to get the position of the sun will be based on real-world light sensitivity experiments and data, and the actual installation of the demo frame. Edge cases like entering a tunnel and having different weather conditions will also be considered and tested.

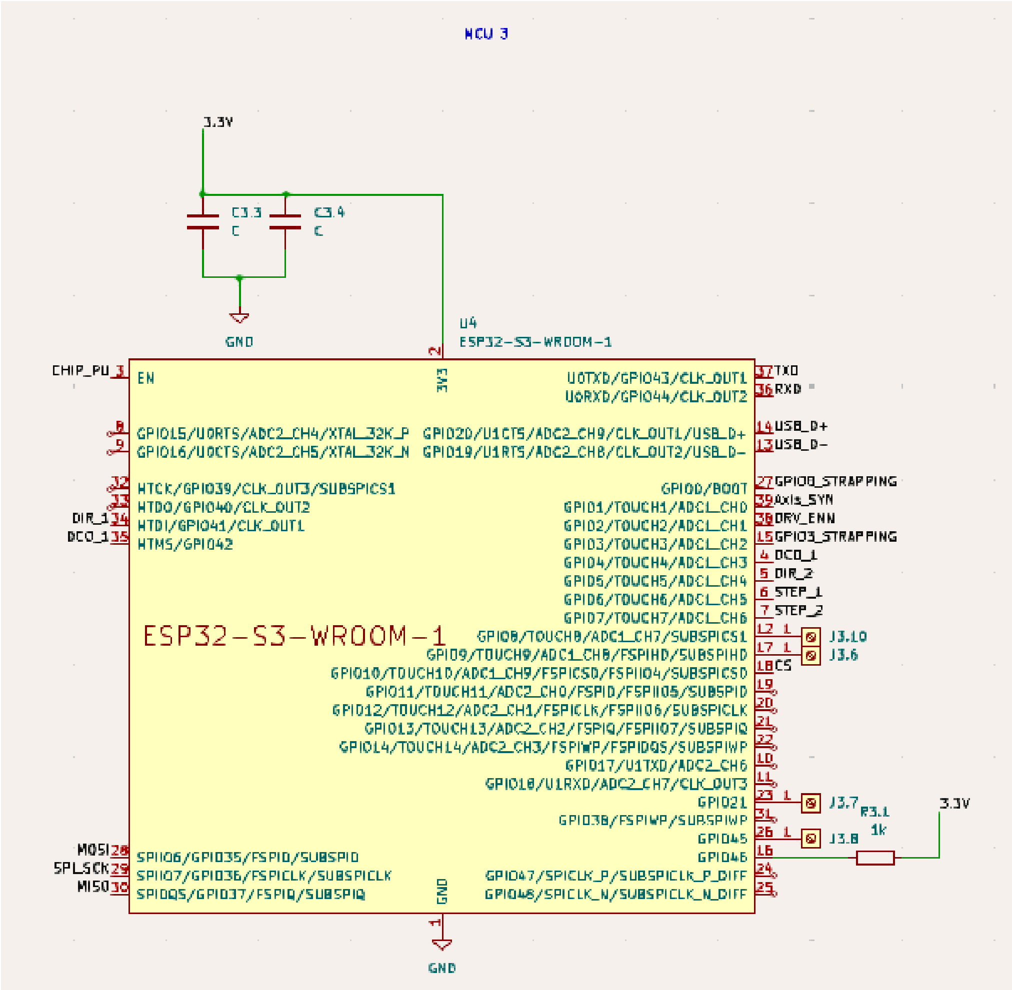
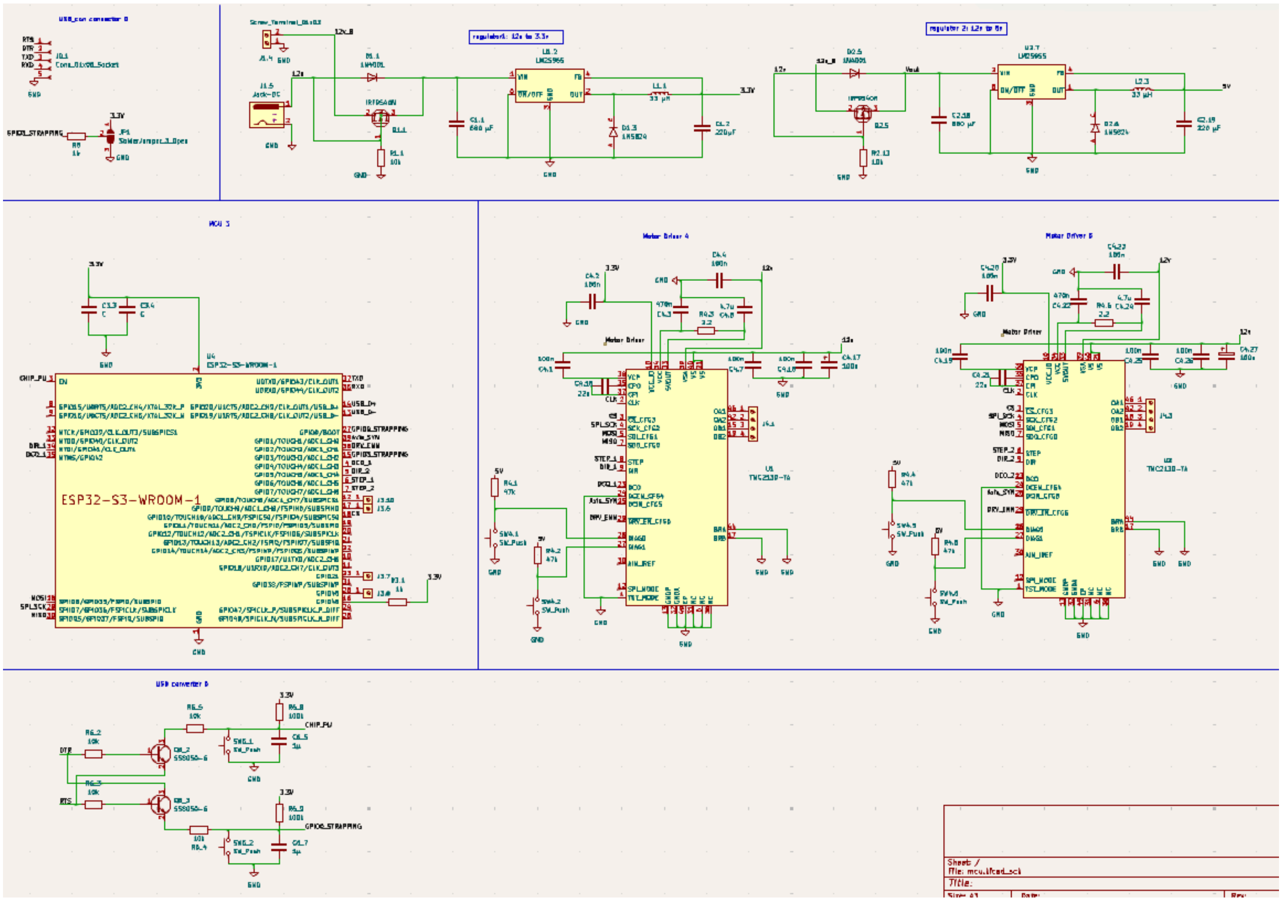



Figure 2.7: ESP32-S3 Pin assignment

| Requirements | Verification |
|---|---|
| Be able to receive signal from sensor unit continuously<br>Be able to send order to motor with given light intensity from sensor unit<br>Be able to process input lux and send order with 2s (motor should begin to move after light source change)<br>Be able to deal with the situation mentioned in the | - Power up sensor and pcb board<br>- Connect sensor and pcb together<br>- Waving hand on the sensor and print out the input of pcb at the backend<br>- connect motor with pcb board and send simple order to make the motor move<br>- change the light source and measure how long it takes to move the motor |

| | |
|---|---|
| requirements (eg. tunnel) | - To check whether the program is correct, use the model frame and specific light source. Then verify the board is at the correct position.<br>- Model the situation like a tunnel and power shut down to see whether the motor functions well. |




## 2.4 Hardware Design

Even though the visor is not required to work with wifi or bluetooth, the ESP32-S3 microcontroller chip is still the primary choice with its flexible library packers. Although this microcontroller has a small built-in voltage regulator capable of adjusting up to 5V, as shown in Figure 2.8, it requires 3.3V, and our power supply is 12V DC. Therefore, to provide this regulated voltage, we will use a DC-DC voltage regulator, specifically the LM2596, to create a voltage divider circuit. This chip is beneficial in minimizing heat generation and better efficiency for powering when the motor needs to work or stop and the continuous calculation of sun direction. Also this regulator is capable of providing enough current to support the microcontroller, and its output voltage can be adjusted to exactly 3.3V by co-work with correctly calculated resistors.

**Table 2-12. VDD_SPI Voltage Control**

| EFUSE_VDD_SPI_FORCE | GPIO45 | eFuse [1] | Voltage | VDD_SPI power source [2] |
|---|---|---|---|---|
| 0 | 0 | Ignored | 3.3 V | VDD3P3_RTC via $R_{SPI}$ |
| | 1 | | 1.8 V | Flash Voltage Regulator |
| 1 | Ignored | 0 | 1.8 V | Flash Voltage Regulator |
| | | 1 | 3.3 V | VDD3P3_RTC via $R_{SPI}$ |

[1] eFuse: EFUSE_VDD_SPI_TIEH

[2] See Section 2.5.2 Power Scheme

Figure 2.8: VDD_SPI voltage control table from datasheet *p.25 [2]*

By default, the SPI flash on ESP32-S3 can operate at a maximum clock frequency of 80 MHz, this is a generally relatively fast communication flow speed that is enough for our project. Since the ESP32-S3 features a built-in small voltage regulator, it can adjust the clock frequency according to the command requirement, which eliminates the need for manual regulation. Our power supply will remain steady because we will utilize the building's voltage through a voltage transformer. With the US voltage being 120V, we could apply a 120V to 12V voltage transformer. LM2596 will manage the reduction to 3.3V DC for the microcontroller chip.

To power the motors, which require 12V instead of 3.3V, we use the TMC2130 motor driver. This driver is designed to interface with the microcontroller's logic level while separately receiving a higher motor supply voltage to drive the stepper motors. The TMC2130 can regulate stepper motors from 5 to 46V, up to 1.2A with DC [3]. So the TMC2130 can directly connect from the microcontroller to the motors.

## 2.5 Software Design

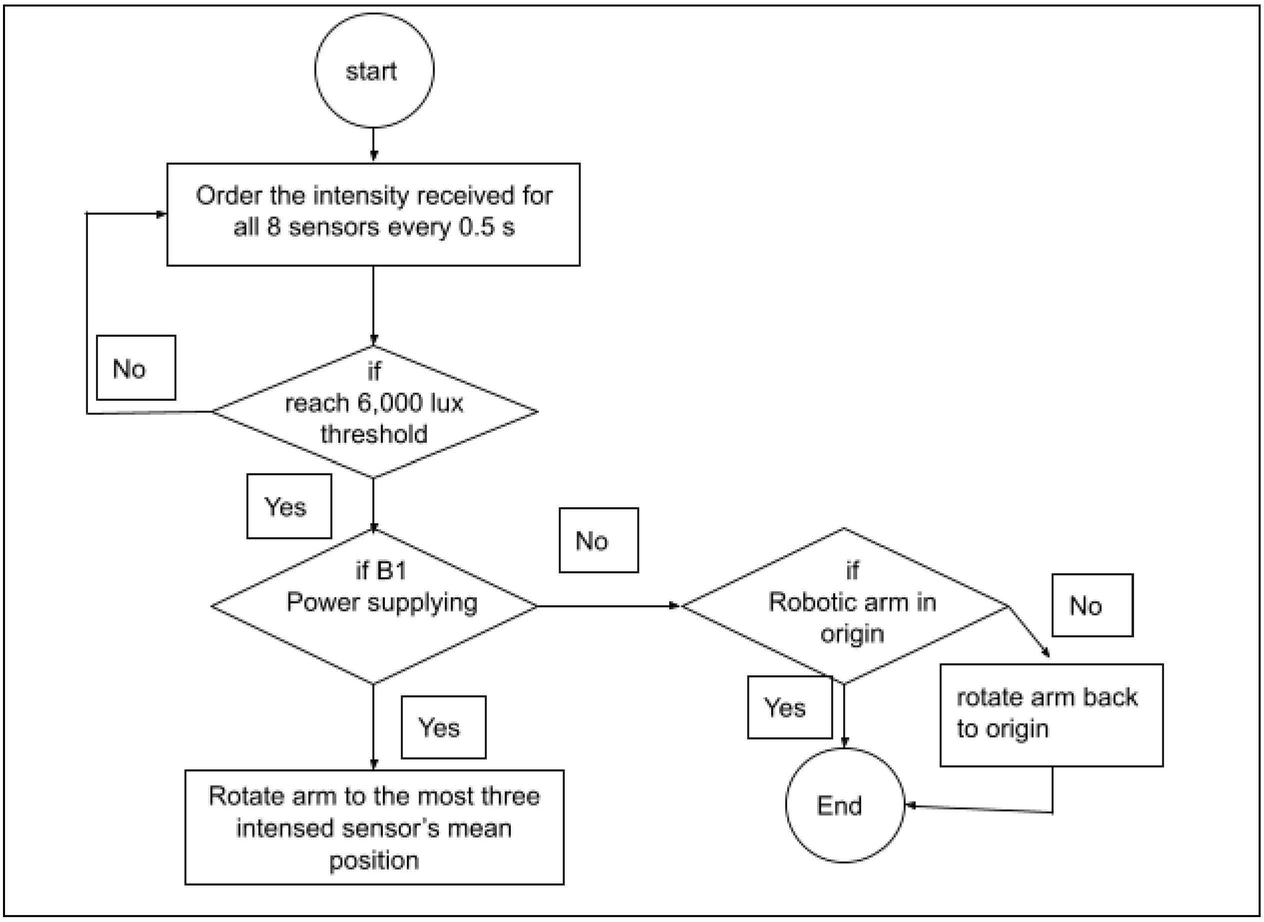



Figure 2.9: Sensor-motor logic flow chart

In our algorithm shown in the flowchart figure 2.9, to calculate the light sensor result to get corresponding output to the motor, we first set up a loop clock that updates the sensor's data every 0.5 second. This is to detect any changes in the sun's position. Then we detect whether the engine is not cut-off (if the person is still driving the car). If the sun position changes and the person is still driving, the motor will move to the corresponding position based on the given order from the control unit. If the sun position does not change or intensity is not as high as the preset-threshold, the visor will not move; if the power is cut off, the visor will go back to its original position by using the backup power.

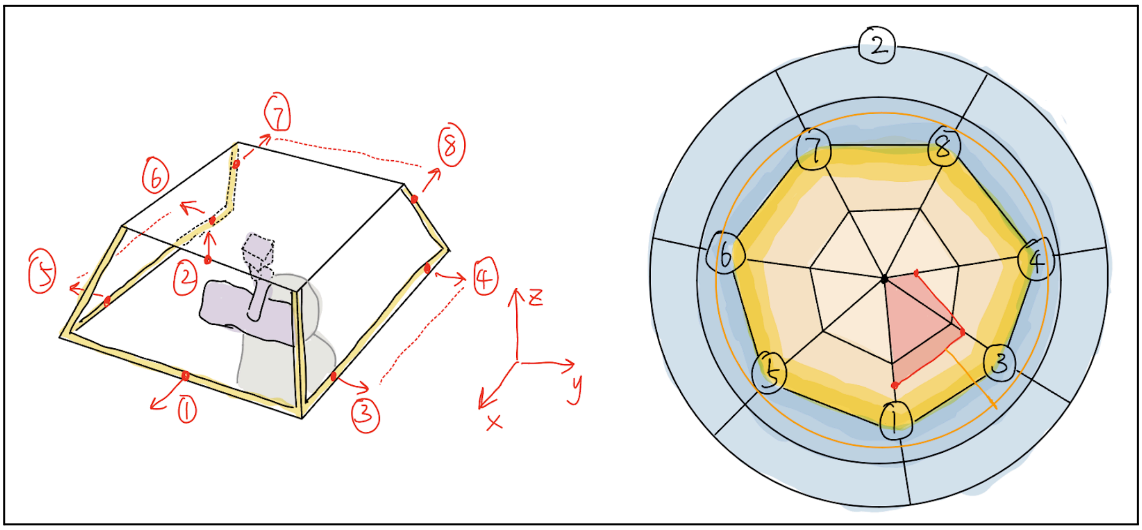



Figure 2.10: light intensity analysis algorithm logic

The algorithm indicates that the position of  motor light is measured by the intensity from each sensor and compared with other sensor relativity. In figure 2.10, the right circle graph shows the measure table. The left side of figure 2.10 shows the corresponding position of each sensor. Sensors 1 to 9 except 2 are side sensors, which are used to detect the sunlight on the side. The # 2 sensor is used to detect sunlight directly from the top. In the circle diagram, the outer circle is intensity for #2 sensor, and one circle inside it is the threshold for the #2 light sensor. The closer it approaches, the lower the intensity in lux that sensor #2 detects. If the intensity sensed is beyond the threshold, the visor will remain in its original position, because this result indicates that the sunlight is relatively above the car's ceiling. Each node of the inner heptagon represents a side sensor, numbered 1 and 3 to 8. The smaller heptagon within the outer heptagon means the intensity threshold line. The central dot indicates an intensity of 0; moving outward from this dot, the detected intensity increases.

As shown on the left side of Figure 2.10, sunlight is coming from the observer's direction. As the result, sensors 1 and 3 detect a similar strength of intensity, while sensor #4 detects less intense light. The red plot illustrates the intensity received by the side sensors relative to the other sensors. The outer orange circle indicate the light intensity received by the top #2 sensor, which is not beyond the threshold, which explains why the visor is lowered into that position blocking sunlight from front and left side.

**2.6 Tolerance Analysis**

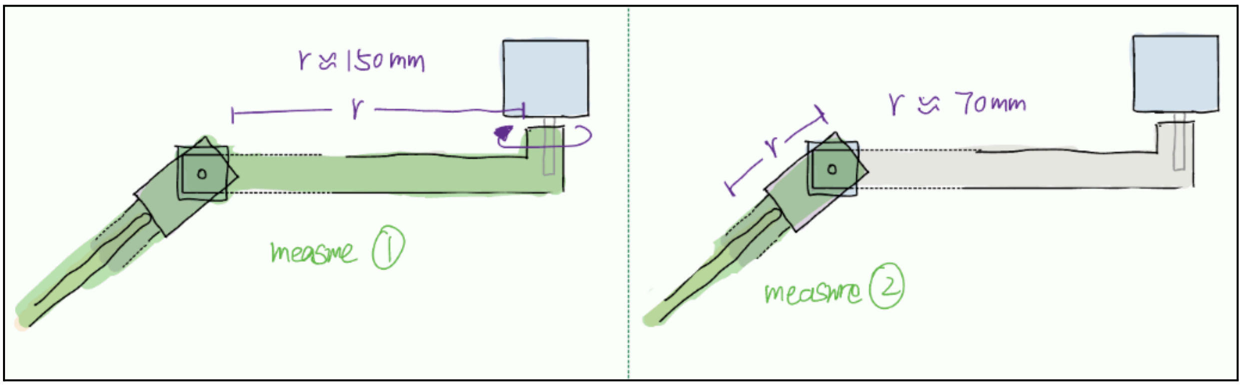



Figure 2.11: Estimate Measurement of tolerance

After discussing with the machine shop, we found a potential issue is that the motor may not have enough torque to be able to hold the weight and rotate. To measure tolerance, we can divide it into two separate measurements for the two main stepper motors. As shown in figure 2.11 measure 1, the motor$^1$ (torque = 59N·cm) is holding the rest of the whole system's weight: steel pipe (mass = 0.3N) , visor (mass = 0.1N), motor$^2$ (mass = 2N, torque = 11N·cm), motor3 (mass = 1.5N). The calculated measured torque will be defined as $T^1$. In measure 2 the motor$^2$ is only needed to hold motor$^3$ and visor, and we define this torque $T^2$. To calculator the torque resulted by the weight, use T = F*r*sinθ, for sinθ = 1 always since the initiative angle made by the arm and pivot is 90° in both case:

$$T^1 = (0.3+0.1+2+1.5) * 15 = 58.5N\cdot cm$$

$$T^2 = (0.1+1.5) *7 = 11.2N{\cdot}cm$$

As the calculated data shows $T^1$ is just at the edge of the tolerance of motor[1] and $T^2$ is 0.2 N·cm over the tolerance level, so to improve the tolerance under the controllable range, we can reduce the arm length r to reduce the torque, and reduce the arm will not impact our final demo since the demo model is a smaller version of the actual product.

## 3. Cost and Schedule

### 3.1 Cost Analysis

The total cost of the parts in the table below is \$349.071. The salary for 1 team member is \$40/hr×2.5×70hr = \$7000. The total labor cost is \$7000×3 = \$21000 for 3 team members. The sum of cost is \$21349.071.

| **Description** | **Manufacturer** | **Quantity** | **Price** | **Sale Tax & Shipping** | **Store** |
|---|---|---|---|---|---|
| Light Sensor TSL2591 | AMS | 8 | \$55.60 | \$5.00 +\$4.67 | Adafruit |
| Regulator LM2596 | Taxas Instruments | 1 | \$6.81 | \$0.681 +\$6.99 | Digikey |
| P-Channel MOSFET IRF9540N | Infineon Technologies | 1 | \$1.39 | \$0.52 +\$6.99 | Digikey |
| Nema-17 Stepper Motor 17HS19-2004S1 | STEPPER ONLINE | 1 | \$13.99 | \$1.40 | Amazon |
| Nema-14 Stepper Motor 14HM11-0404S | STEPPER ONLINE | 1 | \$21.65 | \$2.20 | Amazon |
| Linear Motor L16-100-150-12-P | Actuonix Motion Devices Inc | 1 | \$100 | \$10 | Digikey |
| Stepper Motor driver TMC2130-TA | Watterott Electronic GmbH | 2 | \$13.18 | \$1.32 + \$6.99 | Digikey |
| Microcontroller ESP32-S3-WROOM-1- | Espressif Systems | 3 | \$10.44 | \$1.04 +\$6.99 | Digikey |

| | | | | | |
|---|---|---|---|---|---|
| N16 | | | | | |
| Battery 12V | Duracell | 1 | $4.25 | $0.42 | Amazon |
| Resistor Kit | SparkFun | 1 | $8.95 | $0.90 +$11.62 | SparkFun |
| Steel Pipe 12”(300mm) | Tynulox | 1 | $21.99 | $2.20 | Amazon |
| Visor | 2M PLASTIC | 1 | $18.99 | $1.90 | Amazon |
| total price | $325.69 | | | | |

*Table 1: hardware part price*

### 3.2 Schedule

| **Week** | **Task** | |
|---|---|---|
| **2/19**<br>Design Document due 11:59p Thursday<br><br>Proposal Regrade due 11:59p Friday | Order parts | E |
| | Finished the first draft of schematic for design document | SW&XT |
| | Finalize proposal | E |
| | Start to design board | E |
| **2/26**<br>Design Review 2/26/2024 3:00p<br><br>PCB Review 3:00p - 5:00p ECEB 3081 | Prepare for design review | **E** |
| | Design Power unit | XT |
| | Design motor driver | SW |
| | Test basic function of parts | JH |
| **3/4**<br>First Round PCBway Orders due 4:45p Tuesday<br>Teamwork Evaluation due11:59pWednesday | Test motor driver | SW |
| | Test power on each part | XT |
| | Design Control unit | JH |
| | PCB first order | E |
| **3/11 spring break** | Establish connection on Power and sensor/motor | E |
| **3/18**<br>Second Round PCBWay Orders due | Board assembly | E |
| | Design Control unit | E |

| | | |
|---|---|---|
| 4:45p Tuesday | Test back up power | XT |
| **3/25**<br>Individual progress reports due 11:59p Wednesday | Finalize PCB design | E |
| | Establish connection on sensor and motor | JH |
| | Revision subsystems | SW |
| **4/1** | Finalize motor driver | SW |
| | Test the visor first version | E |
| | Add new parts(eg.switch) if we have time | XT |
| **4/8** | Test on possible bug | |
| | Test functionality | |
| | Finalize Assembly | |
| **4/15** | Mock demo | |
| | Fix any minor bugs | |
| **4/22** | Final demo | |
| | Mock Presentation | |
| **4/29** | Final Presentation | |
| | Final Paper | |

*Table 2: schedule*

## 4. Ethic & safety

● While the primary goal of the project is to improve driving safety by reducing sun glare, the system's reliability and response time are critical. If the system fails to operate correctly or introduces a delay, it could distract the driver or obscure their vision, leading to accidents. We should ensure the final system would not cause anything to interrupt the driving process.

- In critical situations where the vehicle's power supply is compromised, or the system's electronics fail, the ability to manually adjust the visor should remain straightforward and accessible to the driver to ensure visibility is maintained.

- Before engaging in tasks such as soldering, team members should receive proper training and follow proper safety protocols.

- For the final demonstration, we'll use a frame to model the light changing and car movement instead of using a real car. Also, we will make sure that everyone won't get hurt when we are managing the frame.

## Reference:

[1] Nowell Law Firm (2024) *The hidden dangers of car accidents caused by Sun Glare: Law blog, Nowell Law Firm*. Available at: https://www.nowelllawfirm.com/the-hidden-dangers-of-car-accidents-caused-by-sun-glare/#:~:text=According%20to%20the%20National%20Highway,the%20first%20being%20slick%20roads. (Accessed: 20 February 2024).

[2] (No date) *ESP32-S3 series Datasheet*. Available at: https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf (Accessed: 21 February 2024).

[3] *TMC2130* (no date) *TMC2130 | Analog Devices*. Available at: https://www.analog.com/en/products/tmc2130.html (Accessed: 21 February 2024).

[4] *Read our ultimate guide to lux vs lumens vs Watts for lighting installations: Warehouse & Factory Lighting* (no date) *Green Business Light*. Available at: https://greenbusinesslight.com/resources/lighting-lux-lumens-watts/ (Accessed: 21 February 2024).

[5] Wu, G. *et al.* (2019) *Retinal damage after exposure to white light emitting diode lights at different intensities in Sprague-Dawley rats*, *Annals of Eye Science*. Available at: https://aes.amegroups.org/article/view/4933#:~:text=The%20damage%20of%20the%20internal,group%2C%20and%2010%2C000%20lux%20group . (Accessed: 21 February 2024).

[6] *AMS TSL25911 ambient light sensor ambient light, color, spectral & Proximity Sensors - AMS-Osram* (n.d.) *ams*. Available at: https://ams-osram.com/products/sensors/ambient-light-color-spectral-proximity-sensors/ams-tsl25911-ambient-light-sensor

[7] *Read our ultimate guide to lux vs lumens vs Watts for lighting installations: Warehouse & Factory Lighting* (n.d.) *Green Business Light*. Available at: https://greenbusinesslight.com/resources/lighting-lux-lumens-watts/

[8] LM2596 (no date) LM2596 data sheet, product information and support | TI.com. Available at: https://www.ti.com/product/LM2596 (Accessed: 22 February 2024).