Automatic Window and Blinds

By

Mahdi Almosa

Austin Chong

Marco Oyarzun

ECE 445 Final Report

Spring 2024

Group 72

TA: Douglas Yu

Abstract

This paper presents the processes involved in the designing and assembling of a system capable of analyzing user environment to control the degree to which both a window and its blinds open or close. This product aims to ensure a more user-tailored environment for daily comfort as well as sleep efficiency. This product has both commercial and residential applications due to the ubiquity of windows.

Table of Contents

1. Introduction

For our project, we are addressing the challenge of creating an optimal sleep and wake environment as well as a tailored temperature regulation according to individual preferences. We recognize that people have diverse preferences when it comes to their sleep environment, including variations in temperature, light, sound, and air quality. While some prefer a cooler room with natural sunlight in the morning, others may prefer a warmer setting with subdued lighting.

However, implementing these preferences encounters practical constraints, such as safety, security, and energy efficiency. Leaving windows open indefinitely raises concerns about safety, while unpredictable weather conditions and noise disturbances can disrupt sleep. To address these challenges, we propose the development of a self-service window and blind regulator system.

Our solution involves the creation of a comprehensive system that adjusts window openings and blind positions based on user preferences and real-time environmental conditions. By integrating intelligent sensors and responsive controls, we aim to provide users with a seamless and personalized control over their sleep environment while ensuring comfortability and energy efficiency.

Users will have the flexibility to observe temperature, humidity, and air quality and determine what the system should do accordingly. The system will continuously monitor these conditions and display them to the user’s device. Based on user preferences, the system has the capability to adjust window and blind positions to maintain the desired ambient conditions.

The project’s high-level requirements are as follows:

1. Windows/Blinds need to open/close upon request.
2. The windows/blinds should both move automatically or by remote control.
3. The windows and blinds must have sensors that analyze the surrounding environment.

This figure represents our initial block diagram. Throughout the semester we encountered issues and needed to change how the system interpreted the sensors as well as the automated approach.

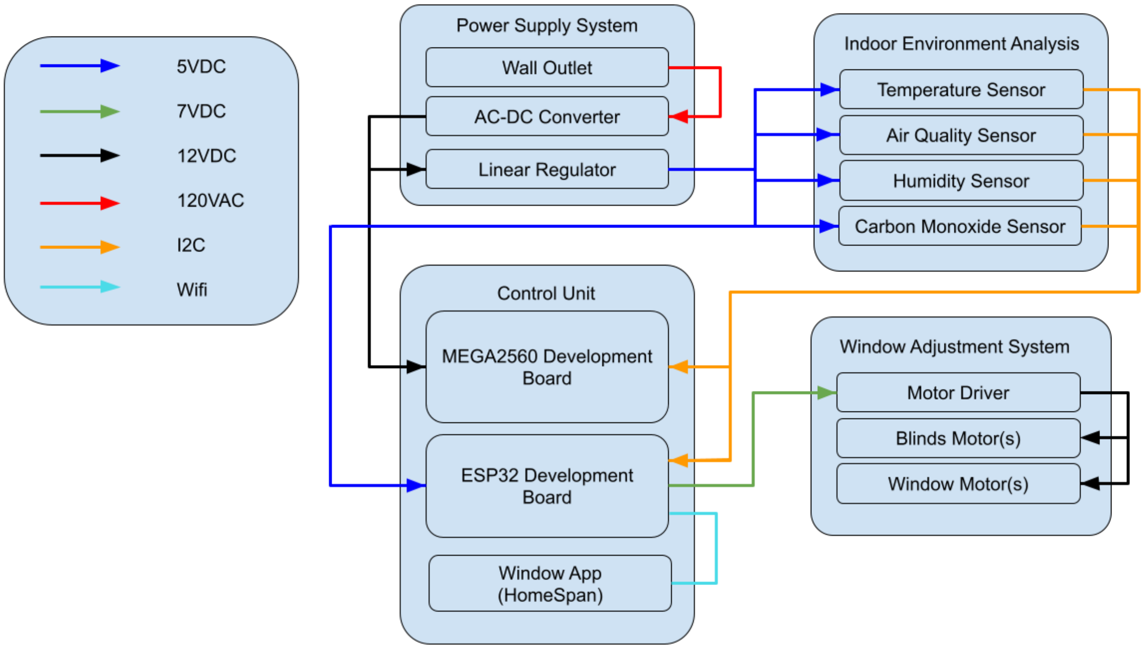

Figure 1. Block Diagram

2. Design

2.1 Power Supply Subsystem

For the Power Supply Subsystem, alternatives to consider included different power sources such as batteries or AC-to-DC adapters. Additionally, various voltage regulation methods, such as linear or switching regulators, could have been explored to ensure stable voltage supply.

The decision was made to utilize an ESP32 and a MEGA 2560, requiring a 3.3V power supply. However, powering the motor for the windows posed challenges due to insufficient current. Initially, a power supply based on voltage compatibility was chosen, but it lacked the necessary current capacity for the motor.

Basic power calculations (P = IV) were used to determine motor power requirements. Equations governing voltage regulator operation were considered for selecting the appropriate regulator for stable voltage supply.

The power supply subsystem likely included voltage regulators, current-limiting resistors, and possibly capacitors for smoothing. The voltage regulator ensured stable 3.3V supply for the ESP32 and MEGA 2560, while additional circuitry may have been needed to provide sufficient current for the motor.

2.2 Environment Detection Subsystem

For the environment detection subsystem, alternative approaches to consider include different types of sensors and sensor configurations. Different sensors may offer varying levels of accuracy, precision, and sensitivity to environmental factors. Additionally, alternative sensor placement and integration methods may impact the system's performance and effectiveness.

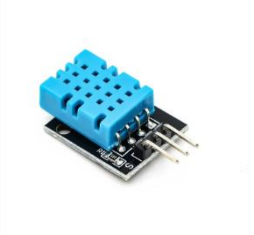

Figure 2. DHT11

Figure 3. MQ7

Figure 4. MQ3

Figure 5. MQ135

The chosen approach involved the use of multiple sensors, including the DHT11 temperature sensor, MQ7 carbon monoxide sensor, MQ3 alcohol sensor, and MQ135 air quality sensor. These sensors were selected based on their ability to detect key environmental parameters relevant to creating an optimal sleep environment. Figure 2 shows DHT11 sensor that measures temperature and humidity, which typically includes a built-in microcontroller interface for data communication and a pull-up resistor for signal stability. Figure 3 shows the MQ7 sensor, which detects carbon monoxide gas concentration, which includes a heating element and a sensing element. Figure 4 shows the MQ3 sensor, which detects alcohol vapor concentration. This also

includes a heating element and a sensing element. Lastly, Figure 5 is the MQ135 sensor. This sensor measures various air pollutants, including ammonia, benzene, and CO2.
All of these sensors are connected to the main control unit which is key in informing the user in order to determine how to adjust the window. This comprehensive data enables more informed decision-making regarding environmental adjustments for better sleep quality.

2.3 Window Subsystem

For the Window Subsystem, alternative approaches to motor control include different motor driver modules and control methods. Alternative motor driver modules such as H-bridge drivers or stepper motor drivers could be considered based on factors like motor type, power requirements, and control complexity.

The chosen approach involved the use of the L298N Motor Driver module. This module is commonly used for controlling DC motors due to its simplicity, versatility, and ability to handle relatively high currents. The high currents posed to be a problem in our implementation, making us utilize a larger power source as a result. The motor comes with a decoder that allows for our code to communicate with this motor.

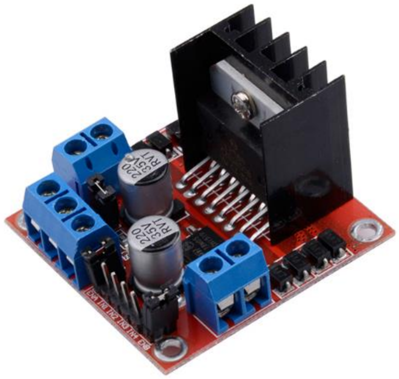

Figure 6. Motor Driver

Figure 7. Test Window

Figure 6 is the L298N Motor Driver module that offers a straightforward solution for controlling DC motors, making it suitable for applications like window control. Its dual H-bridge configuration allows bidirectional control of the motor, enabling both opening and closing actions. Additionally, the module's built-in protection features help ensure safe and reliable operation.

2.4 Control Subsystem

For the control and communication block in our system, alternative approaches include different microcontroller options and communication protocols. Various microcontrollers, such as

Arduino boards, Raspberry Pi, or other IoT platforms, offer different capabilities in terms of processing power, connectivity, and ease of programming.

Communication protocols like Wi-Fi, Bluetooth, Zigbee, or LoRa could be considered for connecting the microcontroller to devices and the motor. Each protocol has its advantages and limitations in terms of range, data rate, and power consumption.

The chosen approach involved using an ESP32 and a Mega 2560 microcontroller. The ESP32 offers built-in Wi-Fi connectivity and sufficient processing power for IoT applications, while the Mega 2560 provides additional GPIO pins and compatibility with a wide range of Arduino libraries.

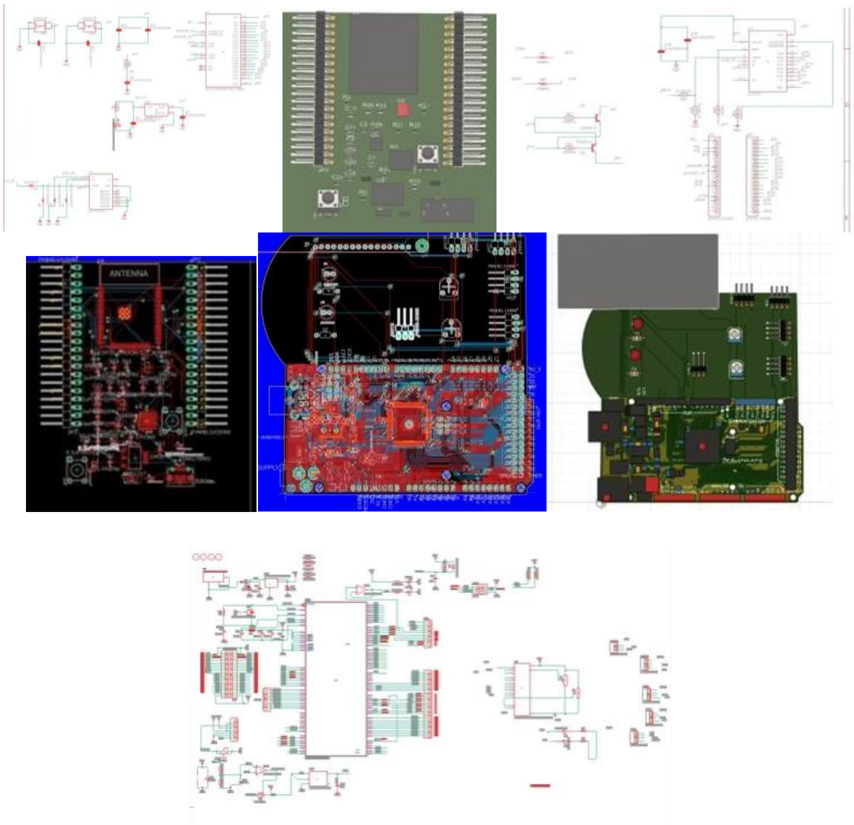

Figure 8. Schematics and PCB Designs

The combination of an ESP32 and Mega 2560 offers a balance of processing power, connectivity, and versatility. The ESP32's Wi-Fi capability enables wireless communication with devices and the motor, facilitating remote control and monitoring. The Mega 2560 provides ample GPIO pins for interfacing with sensors, motor drivers, and other peripheral devices, making it suitable for system integration and expansion.

The use of an ESP32 and Mega 2560 microcontroller combination provides a robust platform for control and communication in your system. Their respective features and capabilities enable efficient interfacing with devices, sensors, and the motor, facilitating the overall functionality of the system. This approach offers flexibility, scalability, and ease of integration, making it a desirable choice for your project requirements.

3. Design Verification

3.1 High-Level Requirement 1

Windows/Blinds need to open/close upon request.
The windows and blinds were both controlled by the gear motors. One of the gear motors was connected to the rod that turns the blinds and simply spun to open or close the blinds. The other gear motor, responsible for the opening of the window, was responsible for rotating a threaded rod that screwed a plate holding the window upward. The final system fully completed both of these tasks.

3.2 High-Level Requirement 2

The windows/blinds should both move automatically or by remote control.
Via remote control, the system was controlled by the HomeSpan app connected to the ESP32 breakout board. The board connected with the app via a Wi-Fi signal and sent signals through the MEGA2560 board. While this was accomplished, the system was not fully autonomous and thus did not completely satisfy this requirement.

3.3 High-Level Requirement 3

The windows and blinds must have sensors that analyze the surrounding environment.
The system used multiple sensors to report data to the user, which included a temperature and humidity sensor as well as three distinct air quality sensors. While not as many sensors were used as originally hoped, this requirement was still a success given that sufficient data was provided to the user.

4. Costs

4.1 Parts

Part Type	Manufacturer	Model Number	Unit Cost	Quantity
ESP32 Development Board	HiLetGo	ESP-WROOM-32D	$9.99	1
MEGA2560 Development Board	ELEGOO	EL-CB-003	$20.99	1
Motor Driver	Qunqi	MK-050	$6.99	2
Temperature/Humiditiy Sensor	Universal Solder	DHT11	$2.23	1
Air Quality Sensor	Olimex	MQ135	$6.29	1
Alcohol Benzine Sensor	Olimex	MQ3	$5.15	1
Carbon Monoxide Sensor	Olimex	MQ7	$5.15	1
Gear Motor	Uxcell	a17092900ux0537	$20.79	2
TOTAL			**$105.36**	

A. Full list of parts with corresponding prices

4.2 Labor

The total cost of labor for all engineers involved can be calculated using the following formula:

($/hour) x 3 engineers x hours to complete x 2.5 multiplier = TOTAL

Assuming an hourly wage of $50/hour for each engineer involved and an average of 20 hours of work per week for 11 weeks the final value for labor costs is **$82,500.**

4.3 Total

Adding the costs of all the parts and labor, the total cost of this project is be estimated to be **$82,605.36.**

5. Conclusion

In conclusion, our project aimed to create an optimal sleep environment tailored to individual preferences through the integration of smart technology. We successfully developed a system utilizing the Apple Home app to display sensor data for humidity, air quality, and temperature, allowing users to make informed decisions about their indoor environment. However, challenges such as powering the motor for window control, having the position saved on the window, as well as the lack of automation were overcome through means of our own.

Moving forward, alternative power sources and voltage regulation methods will be explored to address the current capacity issue. Despite this challenge, our design demonstrates promise in enhancing the sleep experience through user-controlled environmental adjustments.

Our project focuses on creating a personalized sleep environment using smart technology. By integrating sensors and the Apple Home app, users can monitor and control indoor conditions for improved comfort. While successful in sensor integration, challenges for proper window control require further exploration of alternative solutions. Some of these alternative solutions would include utilizing a different motor for both the windows and the blinds. Since we need to correctly record the current position of the blinds and window, we need motors that have a designated start and stop point with a motor of less voltage necessity.

Our project adheres to ethical guidelines outlined by the IEEE Code of Ethics, ensuring integrity, honesty, and accountability in our design process. We prioritize user safety and privacy, implementing measures to protect sensitive data and ensure system reliability. With there being a concern for safety, our initial approach needed to detect incoming intruders. However, this will be a future endeavor as an extension of this project.

Our project's potential impacts extend beyond individual comfort to broader societal benefits. By promoting energy efficiency and personalized environmental control, our system contributes to sustainability efforts and enhances quality of life. Additionally, considerations of accessibility and inclusivity ensure that our technology benefits diverse user demographics, fostering a more inclusive society.

6. References

1. "The acceleration due to gravity (g) - Definition, Value, Formula." BYJU'S, n.d., https://byjus.com/physics/value-of-g/#:~:text=The%20acceleration%20due%20to%20gravity,due%20to%20the%20Earth's%20gravity.

2. "Blind Fabric Material Weight Calculations." Rollertrol, n.d., https://rollertrol.com/blind-fabric-material-weight-calculations.

3. Flori, Robert E. "F=ma Particle Straight Line 1." Missouri University of Science and Technology, https://web.mst.edu/~reflori/be150/Dyn%20Lecture%20Videos/F=ma%20Particle%20Straight%20Line%201/F=ma%20Particle%20Straight%20Line%201.pdf.

4. "ECE Admissions | Why ECE | Salary Averages." Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, https://ece.illinois.edu/admissions/why-ece/salary-averages.
5.

7. Appendix

Requirements and Verification

Power Subsystem

Requirement	**Verification**
The system must provide stable 6-12V DC, 2 Amp power to motor system and DC-DC converter	• Connect the Converter to the wall and use Voltmeters and Ammeters in the ECE 445 lab to measure the output voltage of the Converter, verify the reading is between 11.5 and 12.5 V • After 5 minutes, measure the output voltage and check it is still in the range of 11.5 - 12.5 V

The system must provide the correct stable DC power to the sensors and microcontroller in this project	• Connect each sensor connector to a probe and ensure that:
	  • The ESP32 receives 3.3V
	  • The MEGA receives 12V
	  • The DHT11 receives 2V
	  • The MQ135 receives 1.7V

The system must provide stable -12 ± 0.5 V DC and -5 ± 0.5 V DC power to the motor system	• Connect the negative power converter to the two converters above, connect wires to positive and negative power rail and ground. • Connect the wires to Voltmeters and measure the output voltage of the converter • When connected to the 12V DC output power converter, verify the readings are +(11.5 to 12.5)V and -(12.5 to 11.5) V from the two power rails • When connecting to the 5V DC output power converter, verify the readings are +(4.5 to 5.5)V and -(5.5 to 4.5)V from the two power rail

	• After 5 minutes, measure the output voltage which should be the same compare to the reading 5 minutes ago

Control Subsystem

Requirement	**Verification**
Could read in I2C data passed in from brightness sensor, and 1-wire data passed in from carbon monoxide, rain, IR, humidity, and temp sensors. And analog data passed in from the P.M. 2.5 and temperature sensor.	• Simulate input signals for different pins: carbon monoxide, rain, IR, humidity, and temp sensors can generate high/low signals, thus we can create high/low input to see if Microcontrollers work. • Check the microcontroller can print out the I2C data input coming from the brightness sensor on the monitor. • Simulate analogy data input for P.M. 2.5 and temperature sensor.
Respond within 0.5 seconds to the different input signals, especially for IR sensors	• Simulate input signals and use the oscilloscope to measure the time for the Microcontrollers to respond • Provide the low/high signal for the assigned IR data reading pin measures the microcontroller response time, verify it is less than 0.5 seconds
Microcontrollers should detect the environment in various time frames (detailed in high-level requirements) and check whether we need to change the status	• Use an oscilloscope to check whether the Microcontollers could process the data every 30 seconds / 5 minutes / 30 minutes: check if there is output to the motor when we change the testing environment after 30 seconds / 5 minutes / 30 minutes • Check if the Microcontroller would constantly read data for 30 seconds in a 1-minute duration: we will keep providing high readings to the Microcontroller for 30 seconds check if the motor position changes

Motor Subsystem

Requirement	**Verification**
Window should be steadily opened within 30 seconds	• Attach a piece of paper to the motor and check if the motor spins with a constant Angular velocity: use a timer to see how many cycles the motor spins within 30 seconds. • Used an object with a similar mass as the window to see whether the motor can lift the object steadily • Use the timer to see whether the motor spins enough cycles within 30 seconds
Blinds should be steadily opened within 30 seconds	• Attach a piece of paper on the motor and check if the motor spins with a constant Angular velocity: use a timer to see how many cycles the motor spins within 30 seconds. • Attach the motor to the blinds rotation rod to see whether the motor can open the blinds steadily • Use the timer to see whether the motor performs enough rotations within 30 seconds
Encoder should precisely control the spin of the motor	• Attached a piece of paper on the motor and check if the motor spins the desired number of cycles • Used the timer to measure the time it takes to perform the rotation and check with the preset speed