ECE 445 Senior Design Lab Final Report

Smart Curtains

Team No. 41

Vinay Konda (konda3@illinois.edu)

Max Mauschbaugh (maxjm2@illinois.edu)

Jack Davenport (johndd2@illinois.edu)

TA: Nikhil Arora (na32@illinois.edu)

Professor: Olga Mironenko

Abstract

This report briefly describes an identified problem, and the solution we developed for ECE 445. In this report we will also provide a thorough description of our design choices, related data, and product cost.

Contents

1 Introduction

1.1 Problem

Having a blaring alarm be the first thing you hear in the morning, every morning, makes for a rough morning routine. A better alternative is to wake up naturally with the sun each morning. The first issue with this plan is that one would need to fall asleep with their curtains open in order to have this smooth wakeup, but they then run the risk of waking up earlier than expected with the sunrise.

1.2 Solution

Our solution to this problem was to create smart curtains, an Apple HomeKit enabled device that automatically opens your curtains alongside your alarm clock. Our device also features a light sensing and LED subsystem that compensates for a lack of sunlight on a cloudy morning by activating the attached lights when our device detects that there is not enough ambient light in your room.

1.3 Visual Aid

A visual depiction of our smart curtain system can be found below. The blue spot represents the photo sensor input for our ambient light detection system. The green button in the sensor represents a button that the user would be able to use to manually open or close the curtains independently from the WiFi connected system.

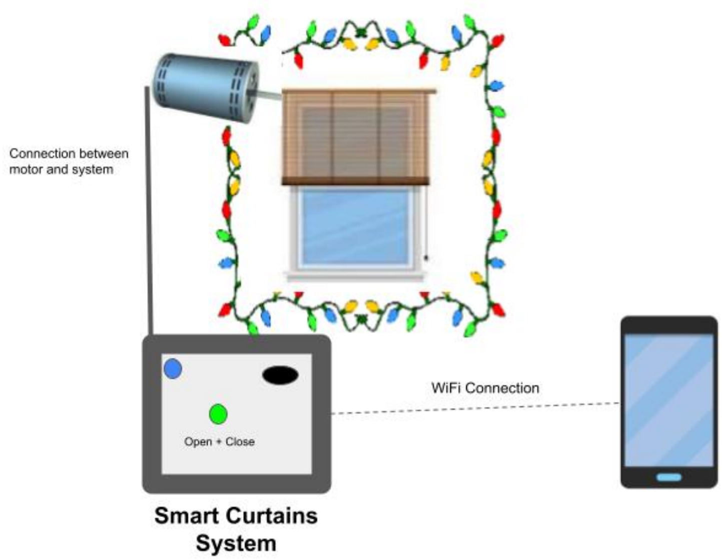

Figure 1: Visual representation of Smart Curtains.

1.4 High-Level Requirements

- Able to open and close curtains in less than 10 seconds using the manual button on the device.
- The curtain system works based on the user's settings set from their smartphone.
- Our LED strip reacts and illuminates a room above 650 lumens when the photoresistor level is lower than our decided threshold value.

2 Design

2.1 Physical Design

Our design utilizes a motor to turn the curtain shaft causing the curtains to raise and lower. Above the curtains, there will be an LED strip for providing additional light if needed in the room and to the bottom left of the curtains, we will use a push button for easy manual control of the curtain. We will use a plastic enclosure to house the majority of our electronic components. Furthermore, there will be a photoresistor placed on the enclosure facing the inside of the room to sense light levels. Additionally, inside of the enclosure we will house the PCB to avoid dangling wires and components.

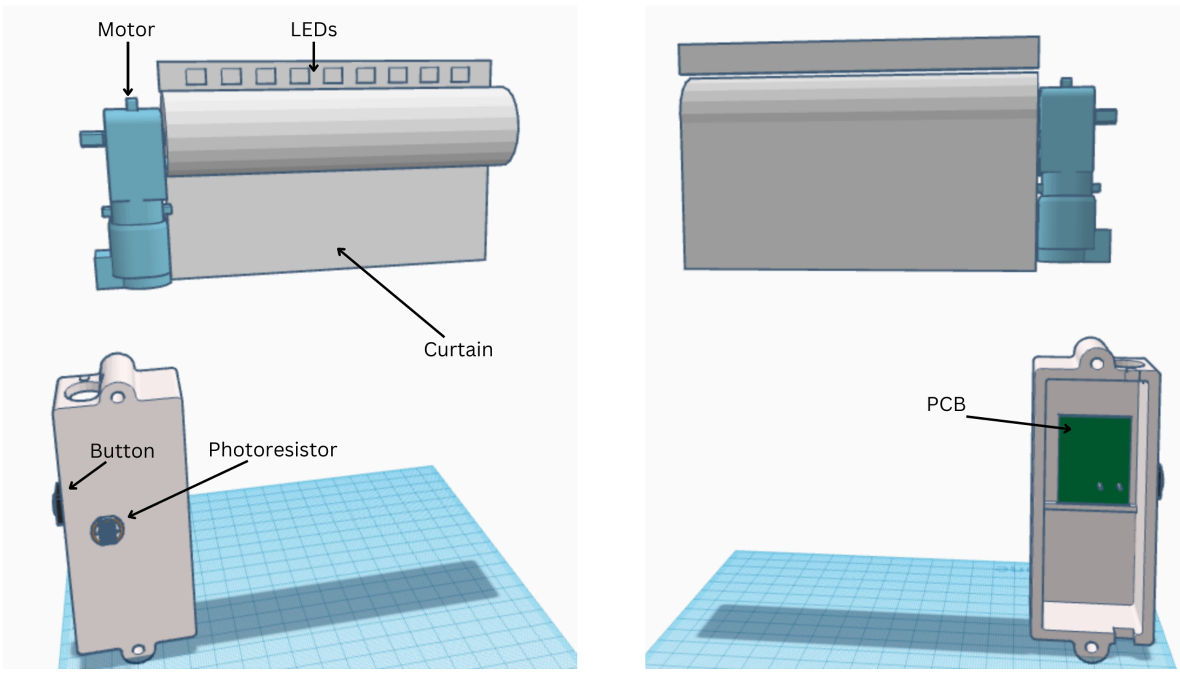

Figure 2: Front and Back view of CAD representation of physical design.

2.2 Block Diagram

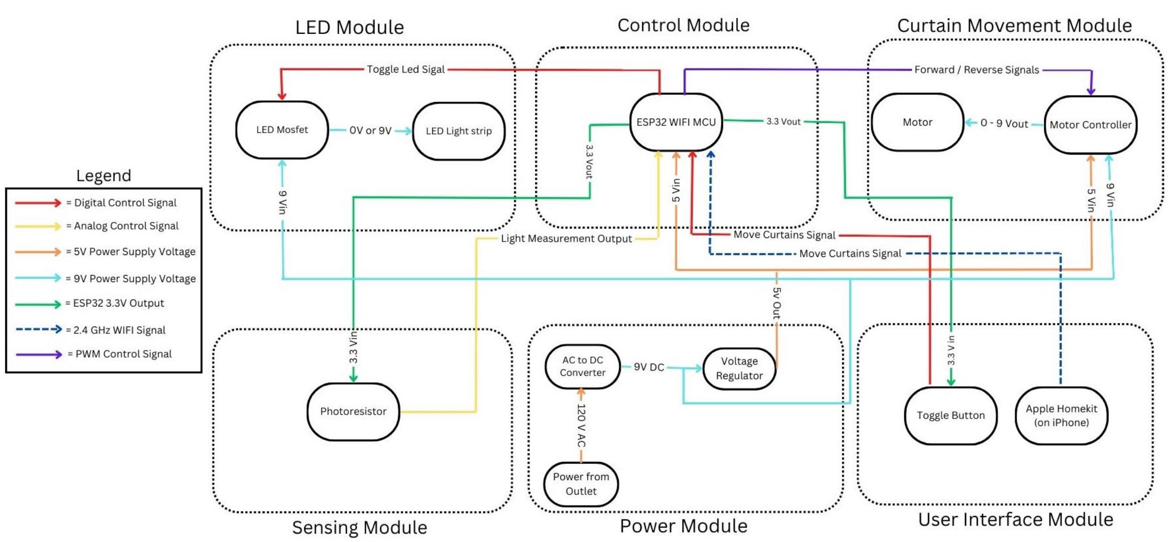

Figure 3: Block diagram representation of Smart Curtains.

2.3 Subsystem Overview

2.3.1 Curtain Movement Subsystem

Overview: The Curtain Movement module was the subsystem used to control the rolling and unrolling of the curtains. This system was composed of an L293D motor controller integrated circuit, a two-phase DC motor, and an encoder housed within the motor.

The L293D chip was used to drive the motor with 9V while only requiring 3.3 V digital signal to be provided from our control unit. The encoder within the motor allowed the control module to keep track of the current state of our curtains, and ensure that the curtains would always be fully opened or fully closed based on the user's desired input.

Design Choices: The main choices we made for our curtain movement subsystem include which motor controller to use, the desired strength of the motor, and whether or not to include an encoder in our system. We also decided to add an array of capacitors with varying values to ensure that the motor controller and encoder wouldn't brown out during operation.

We decided to use the L293D motor controller due to the safety features it provides as well as its high voltage and current tolerance. The device's documentation defines its operating voltage as being anywhere between 4.5 V to 36 V, and its maximum output current being 600 mA [8]. For safety features this device offers internal flyback diodes within the device, as opposed to this device's relative, the L293 driver.

Our final design included the use of an encoder alongside our hobby motor for a few main reasons. The most important reason is that this encoder allows the software to track how open or closed the curtains were, and to precisely fully change the state of the system when the user's alarm is triggered. Also we decided to include the encoder since we had unused pins in our control module, so this enhancement didn't require sacrificing another feature or a large quantity of board space.

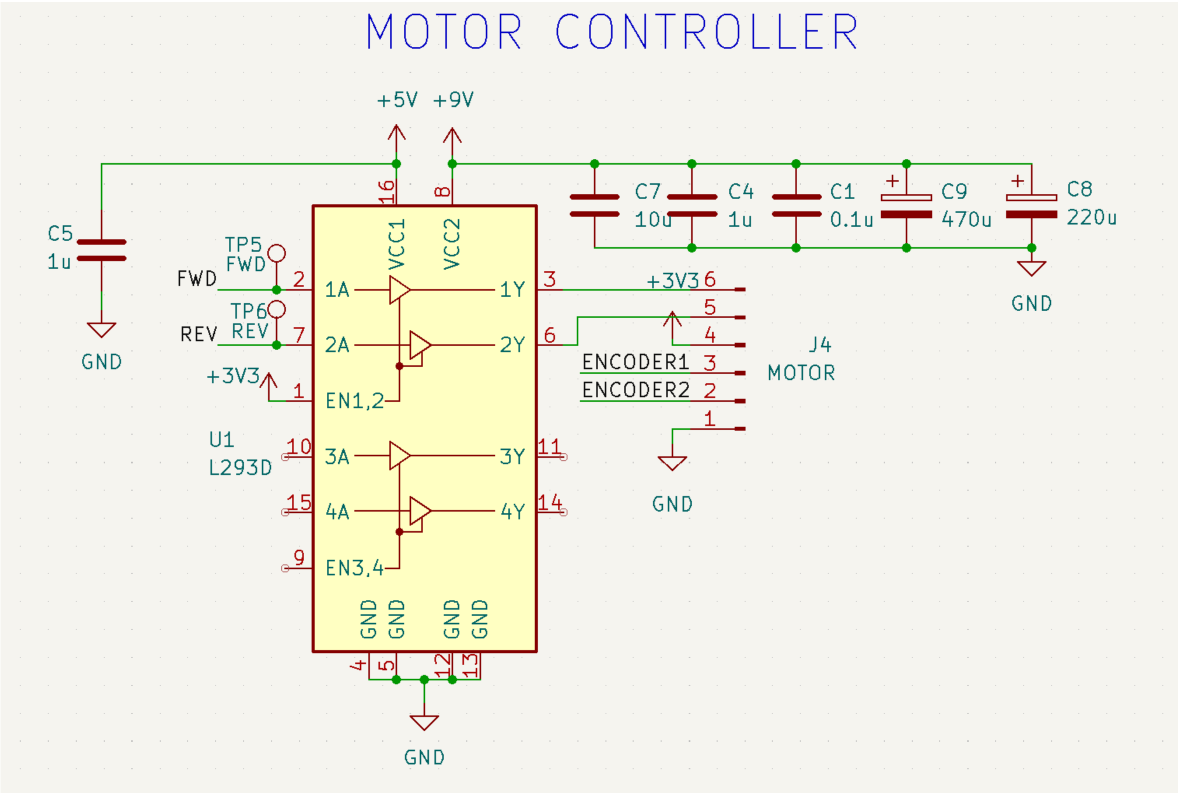

Figure 4: Circuit Diagram of Motor Controller.

Curtain Movement R&V Table:

Requirements	**Verification**
• The motor should open and close the curtain in ~ 10 secs.	• Connect the device to the power source and ensure the curtain shaft is attached to the motor. • Once the curtain is secured to the motor trigger the push button to begin open or close curtain movement. • Start a timer. • Measure the time it takes to roll/unroll the length of the curtain that covers the window and verify that it takes less than 10 seconds. • Perform 10 iterations of this procedure for robust testing.
• The motors should spin just the right amount to have the length of the curtain that covers the window to be rolled/unrolled within ± 0.5 inch of the window length	• Connect the device to the power source and ensure the curtain shaft is attached to the motor. • Trigger the push button to begin open or close curtain movement. • Observe the curtain and with a tape measure, verify whether the bottom of the curtain is within ± 0.5 inch of the window length when unrolled and in the other case check if there is ± 0.5 inch of curtain remaining when it is rolled. • Perform 15 iterations of this procedure for robust testing.

Requirement #1 Test Data:

Trials	**Unroll (secs)**	**Roll (secs)**
1	3.16	3.2
2	3.15	2.83
3	3.13	2.68
4	3.47	3.13
5	3.01	2.76
6	3.04	2.88
7	3.06	2.87
8	3.08	2.99
9	3.18	2.8
10	3.04	2.9

Table 1: Shows the time required for rolling and unrolling our curtain system.

Requirement #2 Test Data:

Trials	Roll Up (inches)	Roll Down (inches)
1	0.17	0.19
2	0.17	0.24
3	0.15	0.19
4	0.19	0.21
5	0.13	0.23
6	0.18	0.24
7	0.18	0.16
8	0.14	0.23
9	0.13	0.22
10	0.22	0.17
11	0.13	0.13
12	0.2	0.23
13	0.2	0.17
14	0.22	0.23
15	0.23	0.19

Table 2: Shows the distance of bottom of curtain from its original starting/ending position after every iteration.

2.3.2 Brightness Monitoring Subsystem

Overview: The explanations of the LED and light sensing modules are more straightforward if the two are grouped together. These two systems are used to measure the ambient light in the user's room, and depending on the decision made from the control unit about this measurement an LED is either turned on or off.

Our photosensing circuit works using a direct voltage divider across a photoresistor and a constant value resistor. As the light in the room increases, the resistance of the photoresistor decreases and therefore V_PHOTO decreases. In turn, V_PHOTO increases as the brightness of the room decreases. Our measured range for V_PHOTO is 1.48 V to 0.32 V.

Our LED design consists of a current limiting resistor, a connection for our LED strip, and a MOSFET used to block or allow the current based on the LED signal provided by the control module.

The ambient light threshold that determines how light or dark a room should be before the LEDs are activated is subjective to each individual room and the user's preferences. A future upgrade to the user interface of our product would include the ability for the user to set this value themselves.

Design Choices: The two main design choices we had to consider for this design were the values of the resistor to be used in our photosensor circuit and how to configure the MOSFET for the LED. Initially we used our MOSFET in a common drain configuration, and ran into issues with using the MOSFET as a switch to the faulty gate-source voltage that arose from this configuration. Due to the results from this initial error, we corrected the circuit to use the MOSFET in a common source configuration.

The next decision we had to make was the value of R2 in the figure below (5). Initially we used a very high value of 20 KOhms as we believed the resistance of the photoresistor was around this value when under light conditions. Our experimentation showed otherwise, as V_PHOTO was consistently around 3 V, which suggests that R2 being 20 KOhms was too large of a value. During tests on a breadboard we found that R2 being 3.3 KOhms gave us the largest range of values for V_PHOTO when comparing the photoresistor under a flashlight versus being covered.

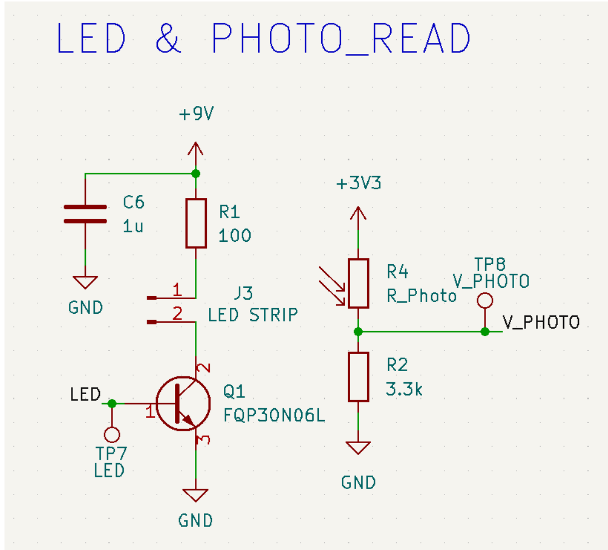

Figure 5: Circuit Diagram of LED Photoresistor Read.

Brightness Monitoring R&V Table:

Requirements	Verification
• The light sensor must be able to read the light intensity in the room and detect when the light threshold (≤ 500) is crossed. This should result in the LED strip turning on.	• Connect the device to power and make sure LED components and light sensor components are mounted to the window frame. • Use a dimmable flashlight and point at light sensor in a dark room. • Using flashlight, switch between medium light and very dim light/no light • Measure the amount of light being measured by the light sensor (0-1023) by reading the ESP32 serial log. • Record the state of the LED strip (ON, OFF) • Repeat procedure 10 times for robustness

• The LEDs must possess enough healthy lighting to make up for darkness inside the room. This light level should be greater than 650 lumens.	• Connect the device to power and make sure LED components and light sensor components are mounted to the window frame. • Make sure photoresistors indicate darkness inside of the room. • Use an external light sensor to measure LED brightness and verify whether it is greater than 650 lumens during night time.

Requirement #1 Test Data:

Trials	No Light	Low Light	Medium Light	High Light
1	200, ON	636, Off	784, Off	922, Off
2	231, ON	627, Off	768, Off	979, Off
3	241, ON	623, Off	764, Off	958, Off
4	213, ON	638, Off	787, Off	944, Off
5	244, ON	671, Off	760, Off	988, Off
6	200, ON	610, Off	762, Off	988, Off
7	210, ON	617, Off	793, Off	904, Off
8	243, ON	675, Off	764, Off	980, Off
9	246, ON	626, Off	771, Off	972, Off
10	203, ON	636, Off	761, Off	937, Off

Table 3: Photoresistor serial monitor log value and LED strip state in various light scenarios

Requirement #2 Test: To validate requirement 2, we checked the documentation of the LED strip included in our design [7]. This documentation indicates that the luminous flux of the LED strip at 12 V is 2400 lm. Our LED strip is measured to operate at 9 V, so our LED strip emits $\frac{3}{4}$ of 2400 lm, which is 1800 lm, satisfying our 650 lm requirement.

2.3.3 Processing Subsystem

Overview: We decided to use an ESP32 development board as the microprocessor for our control module. This processor serves as a hub for sending all logical outputs based on analog or digital inputs received from the other subsystems. As you can see from the block diagram (figure 3), the control module is the main mechanism through which the rest of the systems work together. This processor handles reading alarm information through the user's smartphone, receiving inputs from the toggle button or feedback from our sensing system, and sending digital signals to our motor controller and LED subsystem.

Design Choices: We decided to use the ESP32-WROOM microcontroller for its diverse capabilities and amount of documentation that can be found online [6].

We utilized a couple of the many capabilities offered by this chip. The most important of these features was the seamless WiFi connection capability, useful for our system to read information about the user's smartphone alarm clock. Another useful feature we utilized was the internal analog to digital converter that allowed us to convert our photo sensor input into a digital quantity. Finally the last feature that was helpful was the PWM function offered through a provided library within the Arduino framework.

The final design choice we had to make for our final product was to utilize a predesigned development board instead of installing the chip and all necessary hardware directly into our design. This decision was made due to issues with soldering the chip onto our device with the tools provided in our lab. The ESP32 has a high thermal mass compared to smaller components such as hand solderable resistors and capacitors, so the heat guns that worked with other components could not make stable connections with our microprocessor.

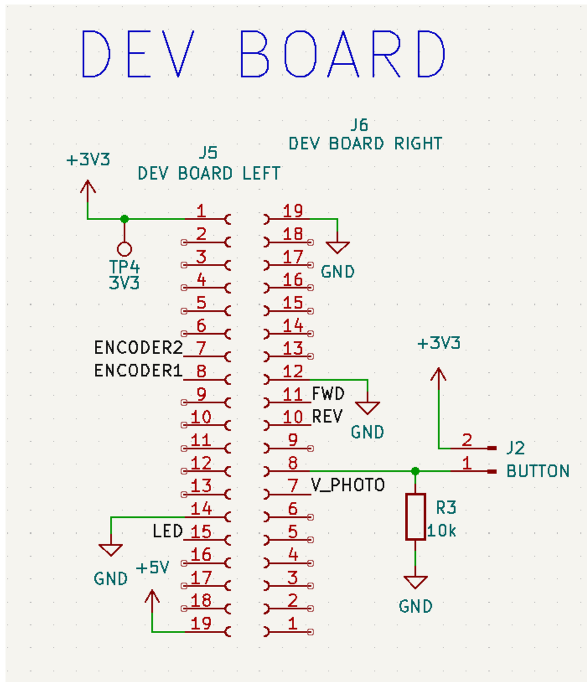

Figure 6: Control module schematic.

Processing R&V Table:

Requirements	Verification
• Even if the iPhone is in sleep mode during an alarm, the PCB WiFi-chip should still receive a trigger from the phone alarm within $\sim$ 1 sec and move its curtains accordingly.	• Connect the device to power and ensure all systems are attached to the window frame. • Wait until the iPhone is asleep/snoozed and leave it in this state during a wake up time. • Use a timer to verify that the PCB receives a signal and begins to open the curtains within $\sim$ 1 second after the wake up time. • Test feature at least 10 times to ensure speed in functionality.

• The PCB should handle manual button presses appropriately (i.e depending on the current state of the PCB, the button should trigger either a close command or open command).	• Connect the device to power and ensure all systems are attached to the window frame. • Set the device to an open curtains state. Toggle the button and verify it closes the curtains in that scenario. • Then set the device to a closed curtains state and toggle the button to verify it opens in that scenario. • Test button press handling at least 10 times to ensure PCB responds correctly with above procedure.
• For every significant change of sensed light levels from inside the room, the PCB should accurately receive data from the light sensors and dim/brighten the LED strip accordingly within ~ 1 sec.	• Connect the device to power and ensure all systems are attached to the window frame. • Make sure the room is completely dark and observe the LEDs at full brightness. • Start a timer. • Then using a dimmable flashlight, increase the amount of light observed by the photosensors. • Using the timer verify that the LED strip changes accordingly in ~ 1 second as the light levels pass the “darkness” threshold • Repeat with different changes in light in different environments to make sure the sensors work properly.

Requirement #1 Test Data:

Trials	**Time after Alarm (secs)**
1	0.89
2	0.85
3	0.8
4	0.9
5	0.85
6	0.86
7	0.83
8	0.81
9	0.91
10	0.8

Table 4: Measures how quickly the PCB responds to the iPhone when it is in sleep mode and the alarm triggers the curtain opening mechanism

Requirement #2 Test Data:

Trials	Start State (% open)	End State (% open)
1	0	100
2	25	100
3	50	100
4	75	0
5	100	0

Table 5: Shows curtain position before and after a manual button press with varying initial curtain positions.

Requirement #3 Test Data:

Trials	Time after light change (secs)
1	0.03
2	0.03
3	0.04
4	0.04
5	0.02
6	0.03
7	0.04
8	0.02
9	0.02
10	0.03

Table 6: Show response time of LEDs when the light levels go from high light to no light.

2.3.4 Power Subsystem

Overview: Our power system handles receiving the 9V barrel jack connector and converting this input to a 5V power supply. We use an L805 voltage regulator to handle our conversion in a safe and reliable way. The 9V power supply is used to drive our motor and LEDs, while the 5V supply drives all of our integrated circuits.

We placed two bypass capacitors on either end of the power supply in order to reduce noise at the input and output from affecting our system. The values of these two capacitors were chosen by the specifications recommended in the device's datasheet [9]. We also provided three test points around the power supply for our own testing convenience. These test points allow us to externally supply 9V, test the 9V to 5V conversion, and easily connect a ground wire to our power supply.

Design Choices: We chose to use the L7805CV device due to its high tolerance for input voltages and current draw. Another large factor in our decision to use the L7805CV was for the reliability of the output. This device also provides a large number of safety features while still maintaining its compact design. These safety features include thermal shutdown and internal overcurrent protection to ensure that the device is virtually indestructible from the power drawn by our circuit [9].

For our final product we chose to use wall mounted power instead of a battery system. Our reason for choosing this is that our product is a stationary wall mounted device, so connecting to wall power is feasible. This also provides us with more stable power and reduces the amount of user interaction required with our product since they won't have to deal with charging or replacing the batteries in our product.

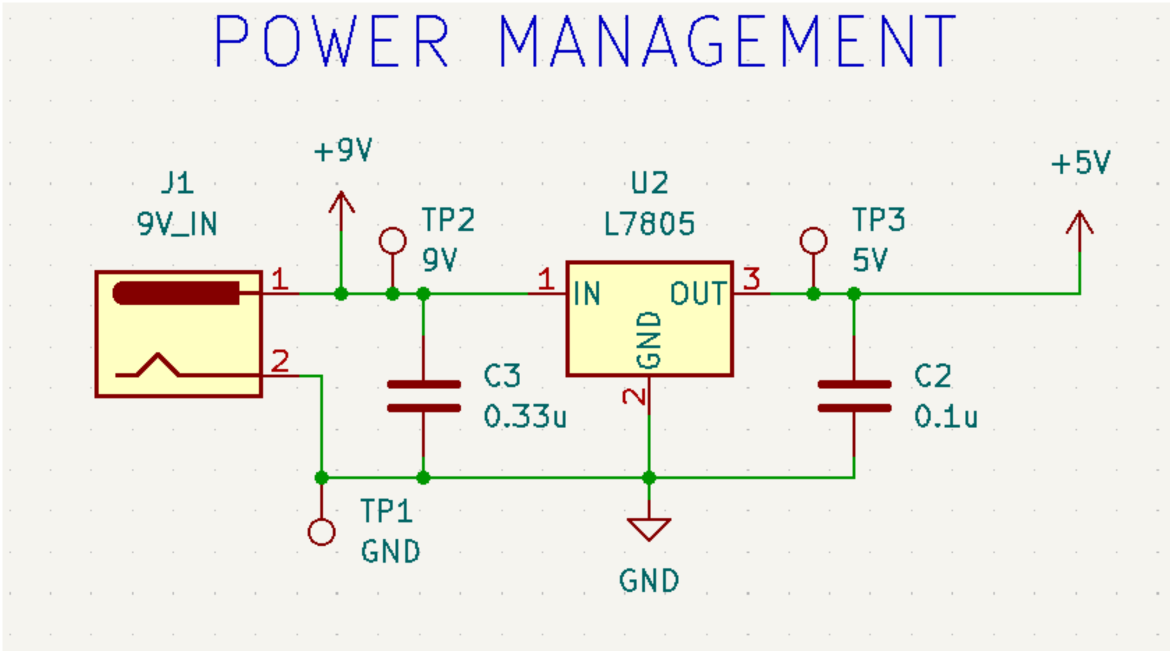

Figure 7: Circuit diagram of power management.

Power R&V Table:

Requirements	Verification
• The system should have protection against overvoltage, overcurrent, under-voltage, and undercurrent.	• Connect the device to power and ensure all systems are attached to the window frame. • Solder wires to the power rail and ground. • Set the multimeter to measure voltage and connect the probes to the power and ground wires. Measure the voltage and check that it is within the allowed range. • Set the multimeter to measure current and connect the probes to the power and ground wires. Measure the current and check that it is within the allowed range. • Set the multimeter to measure resistance and connect the probes to the power and ground wires.. Measure the resistance and check that it is within the allowed range. • Repeat steps 2-4 for overvoltage, over-current, undervoltage, and undercurrent. Make sure that the system is not allowing any of these values to be exceeded.

• The Power Subsystem must be able to supply at least 500mA to the rest of the system continuously at 9V $\pm$0.1V and at 5V $\pm$0.1V after the voltage step down.	• Connect the device to power and ensure all systems are attached to the window frame. • Measure the voltage of the device by soldering wires to the power rail and ground and connect it to the probes of a multimeter to ensure it is 9V $\pm$0.1V and 5V $\pm$0.1V after the step down. • Next connect a resistor to the device and measure the current flowing through the resistor with a multimeter. • If the current is greater than or equal to 500mA, then the device is able to supply at least 500mA to the rest of the system continuously at 5V $\pm$0.1V and 9V $\pm$0.1V.

Requirement #1 Test Data:

Device State	Voltage Reading	Current Reading
Motor Stationary + LED Off	5.0V	0.018A
Motor FWD	4.9V	0.026A
Motor REV	4.8V	0.031A
Processor	4.8V	0.396A

Table 7: Voltage and Current Readings for varying Device States

The peak current draw was:

$$0.396mA + 0.031mA = 0.437mA < 600mA$$

Using information from the L7805CV datasheet that the device can tolerate up to 1 A of current draw, the requirement is verified [9].

Requirement #2 Test: We applied a test point to the 5V output and watching the stability of the output as we varied the input using a DC power supply. Our output was a steady 5V while our circuit was fully powered once the input voltage became greater than 7.8V. The requirement was verified by the datasheet for the L7805CV voltage regulator.

2.3.5 Software User Interface Module

Overview: The software user interface involves a slider UI provided by Apple HomeKit. The slider sets the curtain position based on a percentage of openness (0% open indicating a fully closed state and 100% open indicating a fully open state). Once device is added the HomeKit software will remember the device and when on the same network, will show up as a home device option to connect to.

Design Choices We have decided to utilize the curtain user interface module built-in with Apple HomeKit for our product. This interface has enabled us to achieve all the desired functionalities from a control perspective. The user can conveniently adjust the curtains to any percentage of openness or closure by utilizing the curtain slider feature, as demonstrated above. Furthermore, Apple HomeKit provides the capability for users to create customized home automation functions, such as automatically opening the curtains when their phone alarm goes off. To ensure flexible usage, we have incorporated a manual push-button to enable simplified use of the curtains. This feature allows the user to maintain control of the curtains even in situations where a working Wi-Fi connection or phone is unavailable. To accomplish the Apple HomeKit capabilities, our team used an open source library called ESPHap, which is an Arduino library which natively implements Apple's HomeKit protocol. [5]

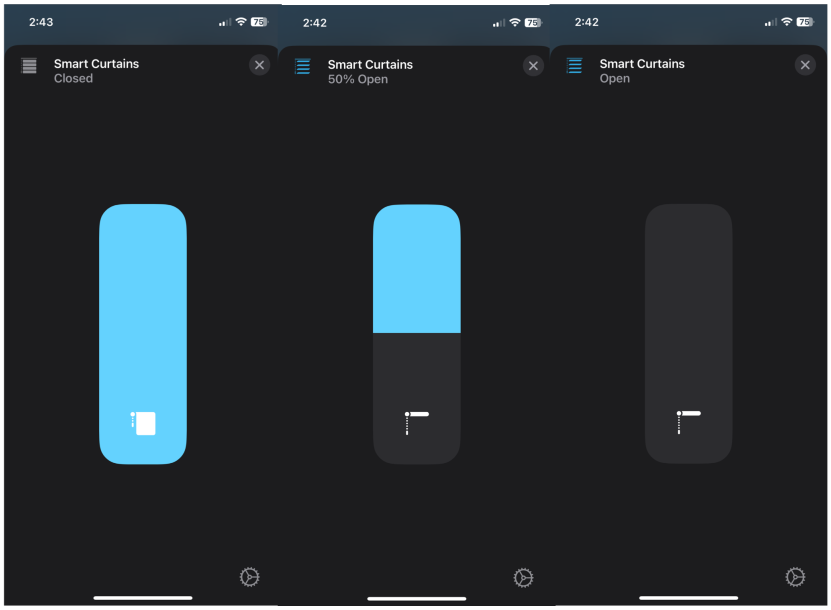

Figure 8: Images of HomeKit user interface when curtain is in different positions.

2.4 Cost Analysis

Here we will analyze the cost of parts and labor.

2.4.1 Labor Costs

According to the UIUC success reports for 2021, the average graduating salary is $80,296. In a 40 hour work week, this salary equates to $38.60/hour. We estimate the hours that we put into this project by week in the table below:

Week #	**Jack**	**Vinay**	**Max**
Week 1	15	6	6
Week 2	7	6	6
Week 3 (Spring B.)	0	0	0
Week 4	8	12	6
Week 5	16	0	8
Week 6	7	10	10
Week 7	15	17	18
Week 8	3	20	20
Total Hours	71	71	74

Table 8: Breakdown of hours worked by group members

$$(71 + 71 + 74) * 38.60\$/\text{hour} * 2.5 = \mathbf{\$20940.5} \quad \textbf{in labor costs}$$

2.4.2 Part Costs

Category	Part	Qty per Board	Cost per unit	Total cost	Spares Qty
Hardware	Blinds			14.08	0
	Push Button			1.48	0
	9V Chord			7.99	0
	LED strip			12.99	0
	Motor			6.62	0
	ESP32 Dev Board	1	6.33	18.99	3
	PCB (JLCPCB)			2.4	3
ICs	Motor Controller (L293D)	1		1.21	0
	9V to 5V (L7805)	1		0.44	0
	NMOS transistor	1		1.13	0
Resistors	100	1	0.1	0.1	0
	3.3k	1	0.1	0.1	0
	10k	1	0.1	0.2	1
	Photoresistor	1	1.3	1.3	0
Capacitors	.1u	2	0.1	0.2	5
	.33u	1	0.1	0.1	0
	1u	3	0.1	0.4	1
	10u	1	0.1	0.2	0
	220u	1	0.49	0.49	0
	470u	1	0.47	0.47	0
Misc	2-Pin Male	2	0.21	0.42	1
	2-Pin Female	2	0.22	0.44	0
	6-Pin Male	1	0.45	0.45	1
	6-Pin Female	1	0.46	0.46	0
	19-Pin Female	2	1.17	2.34	0
	Barrel Jack Input	1	1.61	1.61	0
GRAND TOTAL:		**76.61**			

Figure 9: Breakdown of all part costs.

The total cost for one of our systems amounts to \$68.62. Since we developed three individual prototypes of our project throughout the design process, our total parts cost was:

$$\$68.62/\text{board} * 3 \text{ boards} = \mathbf{\$205.86} \quad \textbf{in parts costs}$$

2.4.3 Grand Total Cost

Adding the parts and the labor, we find that the total cost to develop our our product was:

$$\$205.86 + \$20940.5 = \mathbf{\$21146.36} \quad \textbf{in total costs}$$

3 Conclusion

3.1 Accomplishments

We were able to successfully create, test, and produce our final design, which accomplished all of our high level requirements we had set out to complete. The Smart Curtains we created are able to quickly, conveniently, and seamlessly operate as an Apple HomeKit smart device that allows for personalized user customization for manipulating the curtains. The curtains also have functioning LEDs that turn on when the room is too dark, providing simulated natural light as if it were light outside. For basic use of the curtains, there is also a manual button provided in the design that successfully opens and closes the curtains manually.

3.2 Uncertainties

In our project, we encountered an issue with the Lumens measurement process. Instead of using a light sensor to measure the amount of light produced by the LED strip, we relied on the product spec sheet provided by the manufacturer. The product spec sheet provides an estimate of the Lumens produced by the LED strip, based on the strip's specifications and characteristics at a full 12 V. In our design, we only provide the LED strip with 9 V, so we used a ratio to calculate the estimated Lumens produced. This method may be faulty as it is assuming that the voltage to lumen ratio is linear. Using a light sensor to measure the Lumens produce would provide a more accurate measurement of the LED strip.

3.3 Future Work / Alternatives

There are always opportunities to improve existing designs and add new features that enhance their functionality. In the case of Smart Curtains, one area of potential improvement would be to add a more powerful motor that can handle heavier curtains. Currently, our Smart Curtains only work with lightweight curtains, which limits their application in spaces with heavier or thicker window treatments. Adding a more powerful motor to the smart curtain system would require changes to the existing design, including modifications to the motor mount and control systems. More voltage would likely be needed to operate a more powerful motor, so our circuit schematics would need to be modified to make that possible. In addition to the added functionality of a more powerful motor, another potential design improvement for a smart curtain system could be the ability to control the LED lights within the system using a button or using Apple HomeKit. Currently, the design does not allow you to manually control the LED lights or the threshold of light needed to turn the LED lights on. In a future design, adding a button or functionality through Apple HomeKit could allow the user to have more control and customization of how they want the LED lights to operate.

3.4 Ethical considerations

In considering the ethics and safety of our project, our group will look to adhere to the IEEE code of ethics during the design and creating process [1]. In reviewing our design, there are a few potential safety concerns that we will have to address. Firstly, automated window curtains systems may pose a risk of electric shock if not properly installed or maintained. Things such as exposed wire or improper grounding could lead to this safety hazard. Additionally, we recognize that using a wall outlet to power the device can also pose safety concerns, such as the risk of electrical fires or overloading the circuit. Therefore, we will ensure that our design includes proper insulation and circuit protection measures to prevent these hazards. Another potential risk is the risk of entrapment of fingers or other body parts getting caught in the moving parts of the automated window curtains. To mitigate these risks, we will work to make sure there are no exposed wires or improper wiring in our design, use plastic casing to make sure nothing can get stuck in the motor subsystem, and install the motor subsystem out of reach of everyday movement to avoid potential hazards. Furthermore, excessive motor noise would defeat the purpose of the quiet wake up system we are trying to implement here. In order to decrease this noise we will use insulating and noise cancelling materials that will encapsulate the motor. Finally, motor overheating poses as a safety hazard and even though we are using insulation for noise cancelling, we will ensure that the encapsulation has ventilation holes and the motor itself is far from away other sources of heat. Additionally, the motor will have its own thermal shutoff functionality.

In reviewing our design, there are also a few ethical concerns that we will have to address. Firstly, automated window curtains systems may raise privacy concerns for the users. If used non cautiously, the automated curtains could be opened at times that the user would not want leading to a potential privacy violation. Also, automated window curtain systems may pose security risks if they are vulnerable to hacking or other forms of cyber attack which could lead to unwarranted use. Our team will make these potential concerns transparent to the user and work to mitigate them as much as possible with our design and creation of the system.

As stated in the IEEE code of ethics, we will treat all people we engage with fairly and with respect no matter their race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression. We will be open to and seek honest criticism, and acknowledge and accept any mistakes made along the way, prioritizing the safety, health, and welfare of everyone involved.

References

[1] "IEEE Code of Ethics." IEEE, www.ieee.org/about/corporate/governance/p7-8.html.

[2] Maxim Integrated, "Compact 4.5V to 36V Full-Bridge DC Motor Drivers", MAX14870/MAX14872, Jun. 2017

[3] Nawazi, Farwah. "DC Motor Speed Control PWM Circuit." Circuits DIY, 24 Aug. 2022, www.circuits-diy.com/dc-motor-speed-control-pwm-circuit/.

[4] Texas Instruments, "xx555 Precision Timers", NE555D, Sep. 2014

[5] Yurik72. "Yurik72/Esphap: ESP32/esp8266 Arduino Library for Native Apple Homekit Accessory Protocol (HAP)." GitHub, https://github.com/Yurik72/ESPHap.

[6] ESP32 Series - Espressif. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[7] YUNBO. "YUNBO LED Strip Light Narrow Width 4mm Natural White 4000-4500K." Amazon, 2016, https://www.amazon.com/dp/B0928PTDT1?ref=ppx_yo2ov_dt_b_product_details&th=1.

[8] Texas Instruments. "L293x Quadruple Half-H Drivers." L293D ACTIVE Quadruple Half-H Drivers, Texas Instruments, https://www.ti.com/lit/ds/symlink/l293d.pdf?ts=1683013799855

[9] SparkFun Electronics. A7800 Series Positive-Voltage Regulators - SparkFun Electronics. https://www.sparkfun.com/datasheets/Components/LM7805.pdf.